Design and Evaluation of a System to Extend Identity Management to Multiple Devices

Von der Fakultät für Informatik, Elektrotechnik und Informationstechnik
der Universität Stuttgart zur Erlangung der Würde
eines Doktor-Ingenieurs (Dr.-Ing.) genehmigte Abhandlung

vorgelegt von

Marc Andreas Barisch

geb. in Heilbronn-Neckargartach

Hauptberichter:	Prof. Dr.-Ing. Dr. h. c. mult. Paul J. Kühn
1. Mitberichter:	Prof. Dr.-Ing. Erwin Rathgeb (Universität Duisburg-Essen)
2. Mitberichter:	Prof. Dr.-Ing. Andreas Kirstädter
Tag der Einreichung:	28. März 2012
Tag der mündlichen Prüfung:	18. Dezember 2012

Institut für Kommunikationsnetze und Rechnersysteme
der Universität Stuttgart

2012

For Bettina and my parents

Summary

Identity Management (IdM) is a set of technologies and processes that enable the identification of users and the management of associated user data in Information and Communication Technology (ICT) systems. IdM is relevant for service providers (SP) that offer their services across the Internet or within intranets. Hereby, SPs want to restrict access to the offered service to identified and authorized users. Traditionally, SPs are independent of each other. Therefore, each SP forces the user to create an account, i.e. a digital identity. A digital identity consists at least of an identifier and corresponding credentials (e.g. a username/password combination) for authentication.

With an increasing number of used services, the number of digital identities per user increases. The increasing number of identities represents a usability and a security problem. Usability is decreased in two ways. First, the user has to authenticate against each SP manually, which results in increased effort. Second, the user has to memorize the credentials of many digital identities, which is difficult for most people. In consequence, users tend to reuse the same credentials for different SPs. This renders different attacks possible and results in a decreased security.

Federated IdM systems improve the usability and security with the introduction of so called Identity Providers (IdP). Users authenticate once against the IdP and are able to use services provided by all federated SPs without reauthentication and without the need to have an identity per SP. This concept is well-known as Single Sign-On (SSO). Federated IdM can reduce the number of digital identities. However, the user will still have more than one digital identity due to various reasons, e.g. different federations, privacy protection, separation of concerns. Therefore, identity selection is still required.

In recent years the number of devices per user, which are used to consume services, has increased. Users carry different mobile devices, like smartphones and notebooks with them and make use of fixed devices like TV sets. Existing IdM system are device centric, i.e. they focus on individual devices and do not care about users with several devices. Users are forced to authenticate on each device individually.

This thesis designs and evaluates an architecture that extends IdM systems with mechanisms to consider users with multiple devices. The designed mechanisms improve the security and usability regarding the usage of SSO on multiple user devices. The so-called multi-device IdM concept has three design goals: (1) The authentication should take place on the most secure device that is owned by the user. (2) It should be sufficient to authenticate on one of the user's

devices against the IdP. The remaining devices should benefit from that authentication. (3) The user should be guided regarding the selection of an identity with a filtered list of suitable identities. The filtering should consider the available user devices, but also additional information like the usage context (e.g. private or business).

A development methodology, which has been assembled and tailored from existing methodologies, has been used to design the architecture. The development methodology combines best-practices for system and software design. Based on usage scenarios and the elicitation of requirements, the architecture is stepwise refined. Since security is of uttermost importance, the development methodology considers security from the beginning by means of asset and threat identification. For the modeling of the overall architecture and the detailed mechanisms, the Unified Modeling Language (UML) has been used.

The evaluation of the designed system comprises three orthogonal evaluation approaches: (1) The functional evaluation showed that the requirements have been addressed and that the usage scenarios can be realized. A simple prototype serves as proof-of-concept regarding the interworking with an existing IdM system. (2) The security evaluation used attack trees to systematically identify vulnerabilities. No security weakness has been identified. (3) The performance evaluation established an analytical model to illustrate and quantify the consequences of multi-device IdM. The results showed that multi-device IdM significantly reduces the number of authentication procedures that a user has to perform leading to an improved usability.

Zusammenfassung

Unter dem Begriff Identitätsmanagement (IdM) versteht man eine Menge an Technologien und Prozessen, die die Identifikation von Nutzern und die Verwaltung derer Daten in Systemen der Informations- und Kommunikationstechnologie (IKT) ermöglichen. Für Dienstanbieter, welche ihre Dienste im Internet oder in Intranets anbieten, ist IdM von Interesse, um Nutzer zu identifizieren und zu authentisieren. In herkömmlichen Systemen sind die Dienstanbieter unabhängig voneinander. Ein Nutzer muss deshalb für jeden Dienstanbieter ein eigenes Konto, welches auch unter dem Begriff Digitale Identität bekannt ist, anlegen. Die Digitale Identität besteht mindestens aus einem Bezeichner und zugehörigen Berechtigungsnachweisen (bspw. eine Nutzername/Passwort Kombination), die zur Authentisierung eingesetzt werden.

Mit einer steigenden Zahl an genutzten Diensten, steigt die Anzahl an Digitalen Identitäten. Aus Nutzersicht stellt dies eine Erschwerung der Nutzbarkeit und eine entsprechende Reduktion der Sicherheit dar. Die Nutzbarkeit wird auf zwei Arten erschwert. Zum einen muss der Nutzer sich manuell gegenüber jedem Dienstanbieter authentisieren, was einen Zusatzaufwand darstellt. Zum anderen muss der Nutzer sich die einzelnen Berechtigungsnachweise merken. Als Folge tendieren Nutzer dazu, denselben Nutzernamen und dasselbe Passwort für verschiedene Dienstanbieter zu verwenden. Ein solches Verhalten ermöglicht verschiedene Angriffe und reduziert dadurch die Sicherheit.

Föderierte IdM-Systeme verbessern die Nutzbarkeit und erhöhen die Sicherheit durch die Einführung sogenannter Identitätsprovider (IdP). Nutzer authentisieren sich einmalig gegenüber dem IdP und sind dann in der Lage alle Dienste, die von föderierten Dienstanbietern zur Verfügung gestellt werden, ohne neue Authentisierung zu nutzen. Dieses Konzept ist als Single Sign-On (SSO) bekannt. Föderiertes IdM kann die Anzahl an Identitäten reduzieren. Es wird jedoch angenommen, dass ein Nutzer immer mehr als eine Digitale Identität haben wird, um beispielsweise verschiedene Föderationen zu unterstützen, seine Privatsphäre zu schützen oder unterschiedliche Angelegenheiten zu trennen. Dadurch ist es erforderlich, dass der Nutzer eine Identitätsauswahl durchführt.

In jüngster Vergangenheit ist der Trend zu beobachten, dass die Zahl der Geräte, die ein Nutzer einsetzt um Dienste zu konsumieren, ansteigt. Existierende IdM-Systeme sind jedoch gerätezentrisch, d.h. sie fokussieren sich auf das einzelne Gerät und bieten keine speziellen Mechanismen, um Nutzer mit mehreren Geräten zu unterstützen. Folglich müssen sich Nutzer auf jedem Gerät gegenüber dem IdP authentisieren.

In dieser Arbeit wird eine Architektur entworfen und bewertet, die IdM-Systeme um Mechanismen erweitert, Nutzer mit mehreren Geräten zu unterstützen. Die entworfenen Mechanismen verbessern hierbei die Nutzbarkeit und die Sicherheit bzgl. SSO über mehrere Geräte hinweg. Das entwickelte Mehrgeräte-Identitätsmanagementkonzept verfolgt drei Entwurfsziele: (1) Die Authentisierung soll auf dem sichersten Gerät stattfinden, welches der Nutzer mit sich führt. (2) Es soll ausreichend sein, dass ein Nutzer sich auf einem seiner Geräte gegenüber dem IdP authentisiert hat, die übrigen Geräte sollen in der Lage sein, von dieser Authentisierung zu profitieren. (3) Der Nutzer soll bzgl. der Identitätsauswahl unterstützt werden, indem nur tatsächlich nutzbare Identitäten zur Auswahl stehen. Die Filterung der Identitäten basiert hierbei nicht nur auf den vorhandenen Geräten, sondern auch auf Informationen bzgl. des Nutzungskontextes (privat oder geschäftlich).

Für den Entwurf der Architektur wurde eine zugeschnittene Entwicklungsmethodik verwendet. Die Entwicklungsmethodik kombiniert hierbei bewährte Praktiken für den System- und Softwareentwurf. Basierend auf definierten Nutzungsszenarien und der Ableitung von Anforderungen wurde die Architektur schrittweise verfeinert. Da die Sicherheit von höchster Bedeutung für den Systementwurf ist, wurde bei der Entwicklungsmethodik Sicherheit von Beginn an in Form einer Bedrohungsanalyse berücksichtigt. Für die Modellierung der Systemarchitektur wurde die Unified Modeling Language (UML) verwendet.

Das System wurde mittels dreier verschiedener Ansätze bewertet: (1) Die funktionale Bewertung zeigte, dass die definierten Nutzungsszenarien und die abgeleiteten Anforderungen mit dem entworfenen System erfüllt werden können. Eine prototypische Implementierung zeigte die Umsetzungbarkeit der Architektur in Verbindung mit einem bestehenden IdM-System. (2) Die Sicherheitsbewertung nutzt dabei Angriffsbäume, um systematisch Schwachstellen zu identifizieren. Es wurden keine Schwachstellen identifiziert. (3) Zur Leistungsbewertung wurde ein analytisches Modell entwickelt, um die Auswirkungen des Mehrgeräte-Identitätsmanagementkonzeptes zu bewerten. Die Ergebnisse zeigen, dass die Anzahl an Authentisierungsvorgängen, die ein Nutzer durchführen muss, reduziert werden kann, was zu einer Verbesserung der Gesamtsicherheit führt.

Contents

List of Figures

List of Tables

Abbreviations and Symbols

Abbreviations

AAI	Authentication and Authorization Infrastructure
ARP	Authentication Request Protocol
AuthN	Authentication
AuthNContext	Authentication Context
BLOB	Binary Large Object
BSI	Bundesamt für Sicherheit in der Informationstechnik
CA	Certificate Authority
CAPEX	Capital Expenditures
CAPTCHA	Completely Automated Public Turing test to tell Computers and Humans Apart
CC	Common Criteria
CLASP	Comprehensive, Lightweight, Application Security Process
DFN	Deutsches Forschungsnetz
DFN-AAI	DFN Authentication and Authorization Infrastructure
DoS	Denial of Service
EAL	Evaluation Assurance Level
EAP	Extensible Authentication Protocol
EAPOL	Extensible Authentication Protocol over LAN
GAA	Generic Authentication Architecture
GBA	Generic Bootstrapping Architecture
HCI	Human Computer Interface

HTTP	Hypertext Transfer Protocol
ICT	Information and Communication Technology
ID-FF	Identity Federation Framework
IdM	Identity Management
IdP	Identity Provider
IKEv2	Internet Key Exchange version 2
IMAP	Internet Message Access Protocol
IMSI	International Mobile Subscriber Identity
IP	Internet Protocol
ISO	International Standardization Organization
ITSEC	Information Technology Security Evaluation Criteria
ITU	International Telecommunication Union
KDC	Key Distribution Center
LA	Liberty Alliance
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LoA	Level of Assurance
MD-5	Message-Digest Algorithm 5
MitM	Man in the Middle
MS	Microsoft
OASIS	Organization for the Advancement of Structured Information Standards
OPEX	Operational Expenditures
OWASP	Open Web Application Security Project
P2P	Peer-to-Peer
PET	Privacy Enhancing Technology
PIN	Personal Identification Number
PKI	Public Key Instrustucture
RBAC	Role-Based Access Control

SAML	Security Assertion Markup Language
SASL	Simple Authentication and Security Layer
SDL	Security Development Lifecycle
SDP	Service Discovery Protocol
SIM	Subscriber Identity Module
SLO	Single Logout
SOAP	Simple Object Access Protocol
SP	Service Provider
SSH	Secure Shell
SSL	Secure Socket Layer
SSO	Single Sign-On
STRIDE	Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privileges
TC	Trusted Computing
TCP	Transmission Control Protocol
TCSEC	Trusted Computer System Evaluation Criteria
TLS	Transport Layer Security
TP	Trusted Party
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VPN	Virtual Private Network
WAYF	Where Are You From
WS	Web Service
WWW	World Wide Web
XML	Extensible Markup Language
XMPP	Extensible Messaging and Presence Protocol
XRDS	eXtensible Resource Descriptor Sequence
XRI	eXtensible Resource Identifier

Funtional Components and Interfaces

DM-DDS	DM Device Discovery Service
DM-LDM	DM Local Device Manager
DM-SCB	DM Secure Connection Broker
DM-UI	DM User Interface
DM-VDAM	DM Virtual Device Authorization Manager
DM-VDM	DM Virtual Device Manager
DM	Device Manager
If-App	Application Interface
If-AR	AssertionRequest Interface
If-DA	DeviceAvailability Interface
If-DD	DeviceDiscovery Interface
If-DI	DeviceInformation Interface
If-IA	IdentityActivation Interface
If-IDA	IdentityDataAccess Interface
If-IdM	IdentityManagement Interface
If-II	IdentityInformation Interface
If-L	Logging Interface
If-PR	PolicyRequest Interface
If-SC	SecureConnection Interface
If-SS	SecureStorage Interface
If-VDMod	VirtualDeviceModification Interface
If-VDM	VirtualDeviceManagement Interface
IM-C	IM Controller
IM-DM	IM Data Manager
IM-IFS	IM Filter and Selector
IM-UI	IM User Interface
IM	Identity Manager

ITE-AM	ITE Authorization Manager
ITE-C	ITE Controller
ITE-DM	ITE Data Manager
ITE	Identity Transfer Enabler
SSE-LS	SSE Logging Service
SSE-SSS	SSE Secure Storage Service
SSE	Secure Storage Enabler

Symbols

A	Load
$A_{Auth,C2,max}$	Maximum Authentication Load for several Identities
$A_{Auth,C2}$	Authentication Load for several Identities
$A_{Auth,C3,max}$	Maximum Authentication Load for several Devices
$A_{Auth,C3}$	Authentication Load for several Devices
A_{Auth}	Authentication Load
$A_{Id,i,Dev,j}$	Load for Identity i on Device j
$A_{Id,i}$	Load for Identity i
AR_{C2}	Authentication Rate for several Identities
AR_{C3}	Authentication Rate for several Devices
$E[N_{Act}]_{C2}$	Mean Number of Active Identities in case of several Identities
$E[N_{Act}]_{C3}$	Mean Number of Active Identities in case of several Devices
N_{Act}	Number of Active Identities
N_{Dev}	Number of Devices per User
N_{Id}	Number of Identities per User
$p(x)$	Probability to be in State x
$P_{C2}(N_{act} = k)$	Probability to be in any Macro State in which k different Identities are active for Case 2
$P_{C3}(N_{act} = k)$	Probability to be in any Macro State in which k different Identities are active for Case 3
$p_{k,Id,C2}(x)$	Probability to be in a dedicated Macro State in which k different Identities are active for Case 2
$p_{k,Id,C3}(x)$	Probability to be in a dedicated Macro State in which k different Identities are active for Case 3
$R_{C2,N_{Id}}$	Overhead without SSO
$R_{C3,N_{Dev},N_{Id}}$	Overhead without Virtual Device
$X(t)$	Vector indicating the Number of Service Sessions per Identity at Time t

λ	Mean Service Interarrival Rate
$\lambda_{Id,i}$	Mean Service Interarrival Rate for Identity i
$\lambda_{Id,i,Dev,j}$	Mean Service Interarrival Rate for Identity i on Device j
μ	Mean Service Termination Rate
ν_k	Transition Rate if k Identities are Active

1 Introduction

This chapter motivates the need for identity management systems in Section 1.1. Section 1.2 identifies shortcomings of identity management systems regarding users with several devices. Moreover, it points out the contributions of this thesis and enumerates the author's publications towards this thesis. Section 1.3 outlines the remaining thesis.

1.1 Identity Management - An Enabler for Security and Usability

The Internet consists of a set of service providers (SP) that offer their services, like e-commerce, online banking or webmail to users. Many SPs have the need to identify their users in order to deliver goods, personalize the service, or to restrict access. Hereby it does not matter, whether the service is subject to charges or for free. In order to identify a user, the SP forces the user to create an account, i.e. a digital identity. Such an identity consists at least of an identifier and a credential, e.g. a username and a password. From the perspective of the SP, identity management (IdM) represents a set of security techniques to identify the users and manage the corresponding identities. The corresponding systems are so-called IdM systems.

With an increasing number of SPs, the number of identities per user increases. This represents a usability and security challenge for users. For service consumption it is required to use the identity that corresponds to the SP. That means the user has to memorize or has to have available the corresponding credentials. In case of username password combinations, the users tend to use simple passwords and tend to reuse the same username password combination with different SPs in order to increase the usability [DMR10, KRC06]. The increase of usability results in a decrease of security and makes the user vulnerable to various attacks and amplifies their consequences. Among them are impersonification attacks by malicious SPs, phishing attacks and attacks against the SP's user database that often store user passwords in cleartext as recently exploited [Hun11, Gil11].

Federated IdM systems increase usability and therewidth mitigate security problems. Instead of authenticating against each SP individually, the authentication is performed against an Identity Provider (IdP). SPs that are in same federation as the IdP, trust the IdP regarding user authentication and provide access, i.e. Single Sign-On (SSO). Federated IdM systems increase usability for users because the number of identities could be reduced. If privacy is neglected and if one global federation would exist, the number of identities could be reduced to one per user. Such an identity can be protected by advanced security methods, e.g. strong passwords or dedicated security equipment.

1.2 Identity Management across Devices

From the user's perspective, federated IdM is restricted to one device. That means the authentication against the IdP is only valid and usable on the device on which the authentication has been performed. Today, users possess more than one device (e.g. smartphone, notebook, TV set) and even use them simultaneously in different usage contexts (e.g. private, business). This results in a decreased usability, because the authentication against the IdP has to be performed on each device individually.

This thesis proposes a multi-device IdM solution that increases usability and security by mechanisms that allow the collaboration of all devices that are owned by one user. The multi-device IdM solution has the following advantages:

- Seamless device change: SSO across devices becomes possible. That means if the user has successfully authenticated one device against the IdP, the other devices can consume services without explicit authentication. This brings the user closer to Weiser's vision of disappearing information technology [Wei99].
- Authentication on secure devices: Decoupling of authentication and service consumption becomes possible. In consequence, the more security-critical task of authentication can take place on the more secure device.
- Sharing of security capabilities: Authentication capabilities of another device that is owned by the same user can be used. This enables the fulfillment of SP's or IdP's requirements for particular authentication mechanisms, even if the device on which the service shall be used does not have the required capabilities.
- Usage context awareness: The usage context (e.g. private, business) of identities and devices is considered with respect to the used services. If the user has more than one identity, which is assumed due to privacy and federation aspects, support for identity selection is required. This enables to restrict the usage of identities according to the usage context, e.g. business identities might only be used on business devices.

Intermediate results towards this thesis have been published on various conferences and workshops [Bar11, BTL+10, Bar09, BKM09, BNP+08]. Aspects of this thesis have been part of the EU-funded projects SWIFT and DAIDALOS-II. Corresponding deliverables [B+09, MB+09, MB+10, AB+09, AB+10, GB+08] contain earlier versions of the architecture and related issues.

1.3 Overview of the Thesis

This thesis is structured into three main parts. The first part comprises the fundamental chapters on security (Chapter 2) and on identity management (Chapter 3). The second part introduces the architecture in Chapter 4 and provides details on the mechanisms, algorithms and protocols in Chapter 5. The third part consists of Chapter 6, which evaluates the architecture and the corresponding mechanisms, algorithms, and protocols.

Chapter 2 outlines the importance of security for information and communication technology. It introduces basic security terminology and provides an overview on security mechanisms. Cryptography and authentication mechanisms, which represent key issues for identity management, are discussed and classified. An overview on existing security protocols on different layers of the ISO/OSI stack complements the discussion of security mechanisms. In addition to security mechanisms itself, an appropriate design and evaluation methodology is required for the subsequent system design. Existing principles and methodologies are introduced and classified. They serve as input for the secure design and evaluation of the architecture in subsequent chapters.

Chapter 3 takes a detailed look on identity management and the corresponding identity management systems. It introduces basic terminology and defines the scope of identity management for this thesis. The defined reference architecture shows the roles and work flows of identity management systems. The introduced reference architecture allows the derivation of classification criteria for the subsequent classification of existing identity management systems. Existing identity management systems rely on two basic technologies. These are the Security Assertion Markup Language and the Web Service Federation framework, which are both introduced. Since security is of uttermost importance for identity management systems, a dedicated subsection discusses corresponding security issues. Finally, related work on the application of identity management systems with several devices is presented, categorized and compared.

Chapter 4 designs the functional architecture to realize multi-device identity management. A set of usage scenarios illustrates the benefits of such an extension to identity management systems. For better comprehensibility, three key concepts provide guidance for the subsequent requirement analysis and the architecture design. With a requirements engineering approach functional and non-functional requirements are derived. Security is considered from the beginning of the design phase by early identification of assets and threats and the definition of corresponding security requirements. The functional architecture to enable multi-device identity management specifies the functional blocks and their relationship. It uses the Unified Modeling Language for description. Finally, the design space is evaluated and restricted for the detailed design in Chapter 5.

Chapter 5 details the functional architecture by introducing algorithms, mechanisms and protocols. This chapter has four contributions. First, it specifies a solution to support the user regarding identity selection. Existing identities are filtered and ranked according to their security, authentication and usage context requirements. Based on the filtering, identity usage is restricted to a subset of devices. Second, protocols are specified in order to enable the interworking of devices regarding identity management. This includes the exchange of information on identities, the activation of identities on remote devices and the exchange of credentials to make use of identities. Third, the filtering and interworking of devices for identity management requires a coupling of devices. This is achieved by the virtual device concept that provides security associations among the devices and enables the device discovery. The virtual device concept is the foundation of all security features.

Chapter 6 evaluates the architecture with respect to three different issues. First, the functional evaluation checks, whether the usage scenarios can be realized by the architecture and whether all requirements have been addressed. Moreover, it introduces the prototype that serves as a proof-of-concept. Second, the security evaluation validates the security from three different perspectives. The internal security evaluation has a complete system view, the external security

evaluation takes the attacker's view and the evaluation of misuse cases looks at usage of multi-device IdM in situations that are note covered by the specification. Third, the performance evaluation assesses the potential benefits of the architecture from the user and identity provider perspective by means of an analytical model.

Chapter 7 summarizes this thesis and presents an outlook on future work.

2 Fundamentals of Security

Security in the area of information and communication technology (ICT) is becoming more important for private persons as well as for companies in recent years. With the increased penetration of ICT equipment and a much higher degree of connectivity, not only new possibilities appear, but also the dependency on ICT increases. Examples for these trends are the smart grid initiatives or cloud computing. Also private persons depend more and more on social networks for the organization of contacts or the exchange of information.

In consequence, security of ICT gets and requires more attention. Recent reports [Lag11, F+11] show that the number of security incidents and in particular their sophistication increases. This is subjectively confirmed by various attacks on prominent victims like Sony [Hun11], Google [Dig11], Lookhead Martin [Dre11], Citigroup [Kin11, Jul11] or the Iranian nuclear program [Lan11, CAN11].

Since security is very important for this thesis, this chapter introduces fundamental security concepts in the area of ICT. Figure 2.1 illustrates the structure of this chapter. Section 2.1 introduces security terminology used within this thesis. Based on this terminology, three sections are dedicated to well-established security mechanisms and protocols. Section 2.2 covers basic concepts of cryptography, followed by mechanisms for authentication in Section 2.3 and Section 2.4 introduces selected protocols that provide security functionality on different layers of the TCP/IP network stack [TW10]. Section 2.5 concludes this chapter with the introduction of different methods to design secure systems and corresponding evaluation.

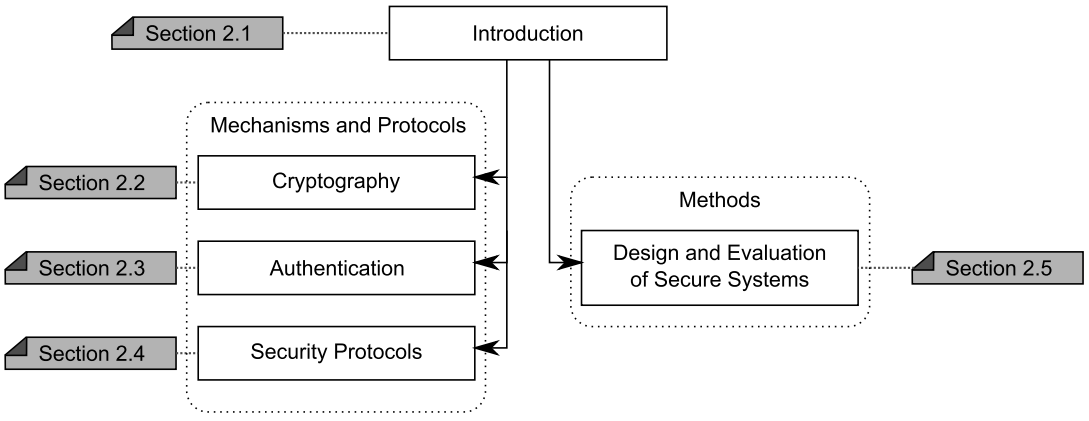


Figure 2.1: Chapter Outline

2.1 Introduction

In literature no unique definition of the term security is available [Chi04]. Two more general, non-recursive definitions are the following:

- ISO 7498-2 [ISOa, MPS+93]: "Security is used in the sense of minimizing the vulnerabilities of assets and resources."
- ISO 24765 [ISOe]: Security is "the protection of system items from accidental or malicious access, use, modification, destruction, or disclosure".

Section 2.1.1 elaborates this definition by introducing additional terms, like asset and vulnerability, and putting them into context to each other. Afterwards, Section 2.1.2 details the goals of providing security in a system and Section 2.1.3 introduces appropriate mechanisms to implement these goals.

2.1.1 Assets, Threats and Vulnerabilities

For an organization or an individual it is essential to protect its assets. Hereby, "an asset is anything of value that should be protected from harm" [Fir05]. An *asset* can be tangible or non-tangible. Examples for tangible assets are people, buildings, or IT hardware. Non-tangible assets comprise among others software, information, knowledge and reputation [Fir05, BSI05].

Assets may have *vulnerabilities*. A vulnerability of an asset represents a "flaw or weakness of the system security procedures, design, implementation or internal controls" [NIS02]. A vulnerability becomes a *threat* if there is a *potential attacker* that might exploit the vulnerability intentionally or unintentionally. A threat is the potential cause of undesirable effects on assets that result in *harm* to the asset owner [Amo94]. Three different kinds of threats can be distinguished [NIS02]: Natural Threats, Human Threats and Environmental Threats. Examples for natural threats are earthquakes of floods. Human threats are caused by intentional (e.g. an attacker) or unintentional (e.g. editing of a database) actions. Examples for environmental threats are power outages or leaks.

In case of the existence of an actual *attacker* a threat might result in an attack. "An attack is some action taken by a malicious intruder that involves the exploitation of certain vulnerabilities in order to cause an existing threat to occur" [Amo94]. Figure 2.2 illustrates the relationships of the above introduced terms.

2.1.2 Security Goals

The term security goal defines a property that has to be provided in order to protect assets. Most authors [Eck09, Bis09, IT91, BBC+05] differentiate between at least three different security goals:

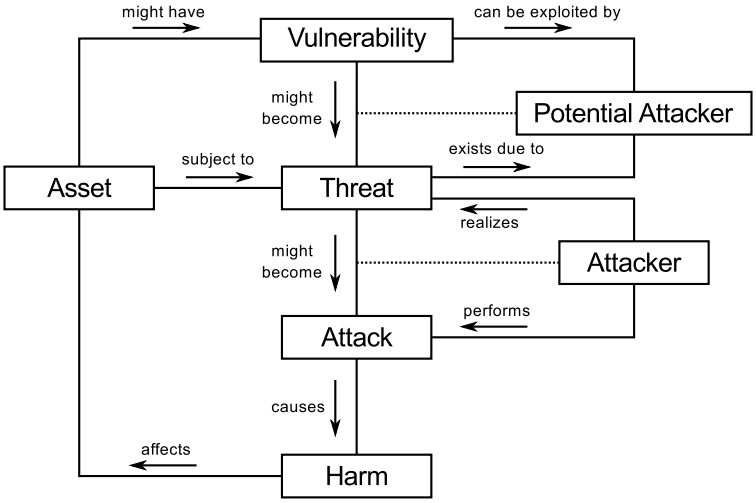


Figure 2.2: Relation of Assets, Threats, Vulnerabilities and Attackers (Adapted from [Fir05])

- Confidentiality: For an unauthorized entity, it is not possible to gain any information. In case of message transmission between a sender and a receiver, it must not be possible for an unauthorized entity to obtain information about the content of the message.

- Integrity: For an unauthorized entity, it is not possible to modify an asset. In case of message transmission between a sender and a receiver, it must not be possible to modify the content of the message for an unauthorized party. [K+08] differentiates between strong integrity protection and weak integrity protection. Strong integrity protection means that it is not possible to modify a message at all, whereas weak integrity protection means that modification of messages is possible but detectable by the receiver.

- Availability: For an unauthorized entity, it is not possible to influence the performance of an asset. In case of stored data, availability means that access to the data is always possible for authorized entities.

In more recent literature, the following security goals have been added:

- Non-Repudiation: For a responsible party, it is not possible to deny the exercise of an action. In case of message transmission between a sender and a receiver, it is not possible for the sender to deny the sending of the message, i.e. sender non-repudiation.

- Authenticity: Authenticity can be considered as a special case of integrity protection for sender and receiver information. For example the authenticity of the sender means that the information about the actual sender is correct [Bis09].

- Anonymity: It is not possible to identify entities within a set of entities [PH10]. For example, an anonymous message does not allow deduce the identity of the sender.

Different sources use alternative terms for the term security goal. [IT91] uses the term security service in order to put it in relation to an actual security mechanism. [Fir04, Fir05] uses the term quality factor, because security is considered as a non-functional quality factor of software. The

term security interest is used by [Bis09] in order to highlight the existence of a party that has an interest in providing such a security property and is therefore closer related to the term requirement.

2.1.3 Security Mechanisms

The above introduced security goals are realized by security mechanisms. Depending on the characterstics of the assets different security mechanisms and the corresponding realization can be distinguished. The following enumeration does not claim to be complete.

- Authentication: A mechanism to verify that an entity is the one it pretends to be. Authentication mechanisms are introduced in more detail in Section 2.3.

- Authorization and Access Control: Access control is the mechanism to prevent unauthorized entities from accessing an asset. The access control decision is based on previously granted authorizations.

- Accounting: A set of mechanisms to create, collect and maintain information about the usage of assets by entities[1,2].

- Auditing: The mechanism to evaluate accounting and other system information with respect to the correctness of the applied security mechanisms. For example, the auditing process could reveal that an unauthorized party has access to the system due to malfunctioning authentication mechanisms.

- Charging: The mechanism of aggregating accounting information and transforming the aggregated information into monetary values[1].

The security mechanisms authentication, authorization, and accounting are abbreviated as AAA mechanisms and often realized in corresponding subsystems. If auditing and charging are added, so-called A4C systems are in place. Cryptography is an additional method to fulfill the above introduced security goals and to realize the security mechanisms (→ Section 2.2).

2.2 Cryptography

Cryptography is one method amongst others to isolate information [Bis09] and realize the security goals of confidentiality, authenticity and integrity. Basically, one can differentiate between symmetric (→ Section 2.2.1) and asymmetric cryptography (→ Section 2.2.2).

[1]The IETF and the 3GPP differ with respect to the meaning of accounting and charging. The differences are elaborated in [BBB+10].

[2]In contrast accountability describes a system property "that ensures that the actions of a system entity may be traced uniquely to that entity, which can then be held responsible for its actions." [RFC4949]

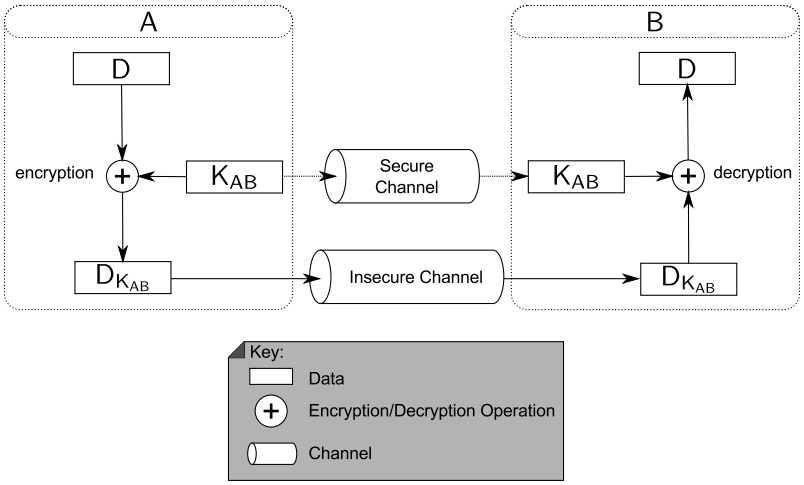


Figure 2.3: Principle of Symmetric Cryptography

2.2.1 Symmetric Cryptography

Symmetric cryptography [Sch96] means that the same key is used for encryption and decryption of data[3]. The sender A encrypts data D with the shared key K_{AB} and sends the encrypted data $D_{K_{AB}}$ to B. The receiver uses the shared key K_{AB} to decrypt the encrypted data $D_{K_{AB}}$ and obtain D as illustrated in Figure 2.3. The main drawback of symmetric cryptography is key distribution. The shared key K_{AB} has to be transferred between A and B across a secure channel. This is challenging and often achieved by an offline channel, like regular mail. In addition, key distribution has a scalability problem. For communication between N parties, $\frac{N(N-1)}{2}$ keys are required. However, symmetric cryptography is compared to asymmetric cryptography less computation-intensive [Sch96] and can be supported by dedicated hardware [PMDW05].

2.2.2 Asymmetric Cryptography

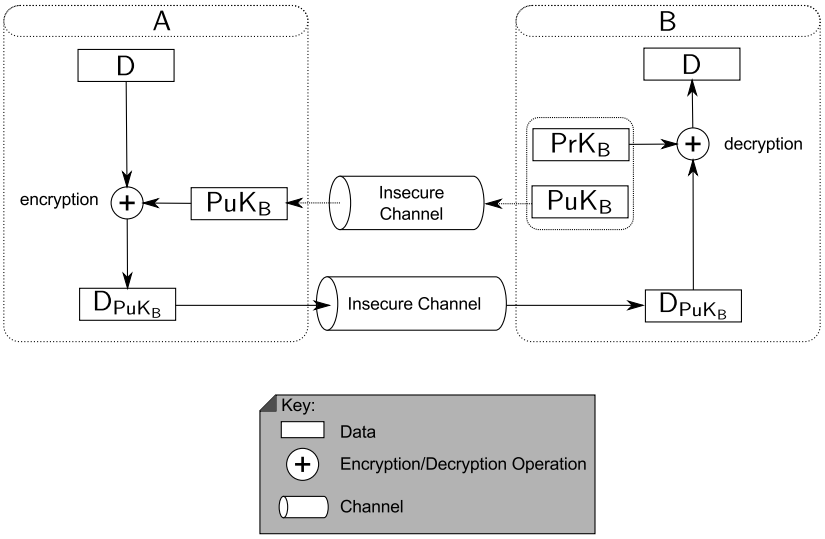
To overcome the key distribution problem asymmetric cryptography has been invented [RSA78, MH78, SPS11][4]. The following subsections introduce the principles and solutions to make asymmetric cryptography useable.

2.2.2.1 Principle

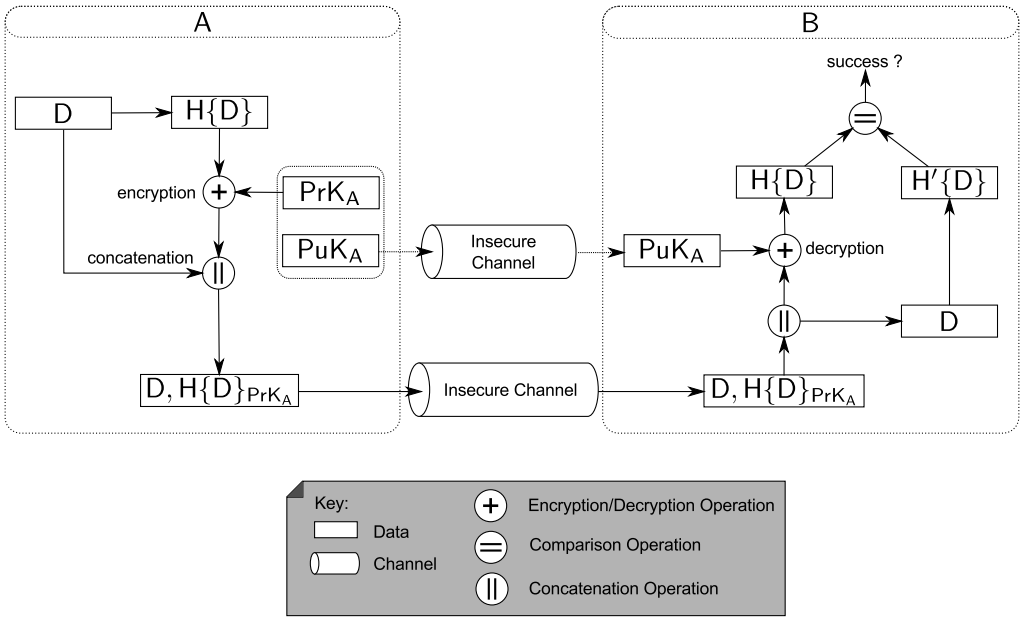
With asymmetric cryptography each party, i.e. A and B, has two keys: A private key PrK and a public key PuK. The public key can be distributed to the world without any security concerns, whereas the private key has to be kept confidential by the owner. Both keys are mathematically related to each other.

[3] In addition, the same cryptosystem is assumed to be used.

[4] In 2010, the IEEE appointed the milestone "Invention of Public Key Cryptography" to James Ellis, Clifford Cocks and Malcolm Williamson. They invented public key cryptography in 1975 at the British Government Communications Headquarters. Unfortunately, they have not been able to publish their invention due to confidentiality reasons [IEEb].



(a) Encryption



(b) Authenticity

Figure 2.4: Principle of Asymmetric Cryptography

For encryption, the sender A uses the public key PuK_B of B to encrypt data D and obtain D_{PuK_B} as shown in Figure 2.4(a). The receiver B can decrypt the encrypted data D_{PuK_B} with its private key PrK_B and obtains D. Since, only B possesses PrK_B no other party is able to decrypt the data.

Asymmetric cryptography can be used not only for encryption of data ($\rightarrow$ Figure 2.4(b)), but also for signing data. Signing data allows verifying the authenticity of data, i.e. whether A is actually the sender of the information or not. To sign data, A uses his private key PrK_A to encrypt an hash value $H\{D\}$ of D and obtain $H\{D\}_{PrK_A}$. B uses the public key PuK_A of A to decrypt the encrypted hash value $H\{D\}_{PrK_A}$ and compare this value against self-calculated hash value $H'\{D\}$. If $H'\{D\} = H\{D\}$ the transmitted data D originates from A.

2.2.2.2 Certificates

Even if the public key needs not to be transported via a secure channel, it is required to verify its authenticity. That means it has to be verified that PuK_A actually belongs to A. If the authenticity of a public key is not correctly verified, man in the middle (MitM) attacks render possible [SGSC+08, Bur02]. The authenticity can be achieved by certificates. A trusted party TP signs PuK_A with its private key PrK_{TP} and the designated name of A ($\rightarrow$ Figure 2.5). It thus certifies that PuK_A belongs to A. This allows the verification of the authenticity of PuK_A with the public key PuK_{TP} of TP and in turn requires to verify the authenticity of PuK_{TP}[5].

The most common format for certificates is the ITU-T specified X.509 format [IT00]. Among the public key of the subject (i.e. PuK_A) and information about the subject (i.e. the distinguished name of A), details on the issuer of the certificate and its validity are stored.

2.2.2.3 Public Key Infrastructures

Certificates and public keys are managed by so-called Public Key Infrastructure (PKI). A PKI consists of the technical infrastructure to

- enable users to verify the authenticity of presented certificates and the contained public keys.
- create certificates.
- verify the validity of certificates, i.e. whether the public key has been revoked.

Certificate Authorities (CA) issue certificates with their private keys. The corresponding public key of a CA represents the trust anchor and is used to verify all issued certificates. Therefore, the authenticity of public keys is of uttermost importance. This is for example achieved by built-in public keys in browsers and operating systems or by public keys distributed on smart cards. Recently, [LL11] proposed additional mechanisms to deal with compromised CAs and thus with forged certificates [Dig11].

2.3 Authentication

Authentication is an essential security mechanism that is of particular importance for identity management systems. The aim of authentication mechanisms is to verify the identity of an entity, i.e. to verify whether an entity is the one it pretends to be. Basically, three different means to authenticate exist ($\rightarrow$ Section 2.3.1). Section 2.3.2 details the usage of authentication means by authentication mechanisms. Authentication mechanisms abstract from actual authentication protocols that define the format and the order of exchanged messages to apply one or several authentication means. In turn, an application can support several authentication protocols. In such

[5]The problem of verification of the certificates of trusted parties is achieved by the built-in of certificates into software (e.g. Firefox) or operating systems (e.g. MS Windows)

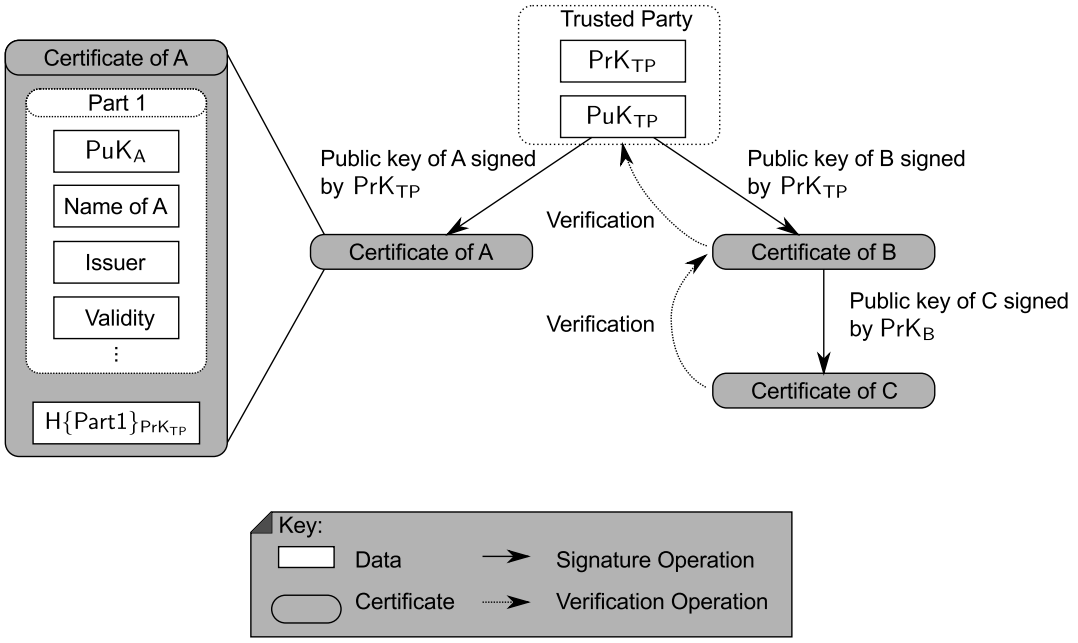


Figure 2.5: Certificates and Public Key Infrastructures

a case, mechanisms select one of the supported protocols (→ Section 2.3.3). If a combination of several mechanisms is applied, it is called multifactor authentication (→ Section 2.3.4).

2.3.1 Means to Authenticate

In literature [O'G03, And07] it is common sense to distinguish between three different means for authentication (→ Figure 2.6).

- Something you know
- Something you have
- Something you are

Independent of the authentication mean two entities participate in the authentication process. The supplicant is the entity that wants to prove his identity and the authenticator is the entity that verifies the claimed identity of the supplicant.

Something you know utilizes a shared secret that is known by both entities. The supplicant presents the shared secret (directly or indirectly) to the authenticator in order to convince him. Examples for shared secrets are username-password combinations. The username is an identifier for the identity and the password is the shared secret used for authentication.

Something you have refers to "physical" credentials that an entity has. Prominent examples are SIM cards used in mobile phones or one-time password generators[6]. Even if easy to copy, X.509 certificates are in this category, too.

[6] An example for a one-time password generator is SecurID produced by RSA[RSA10].

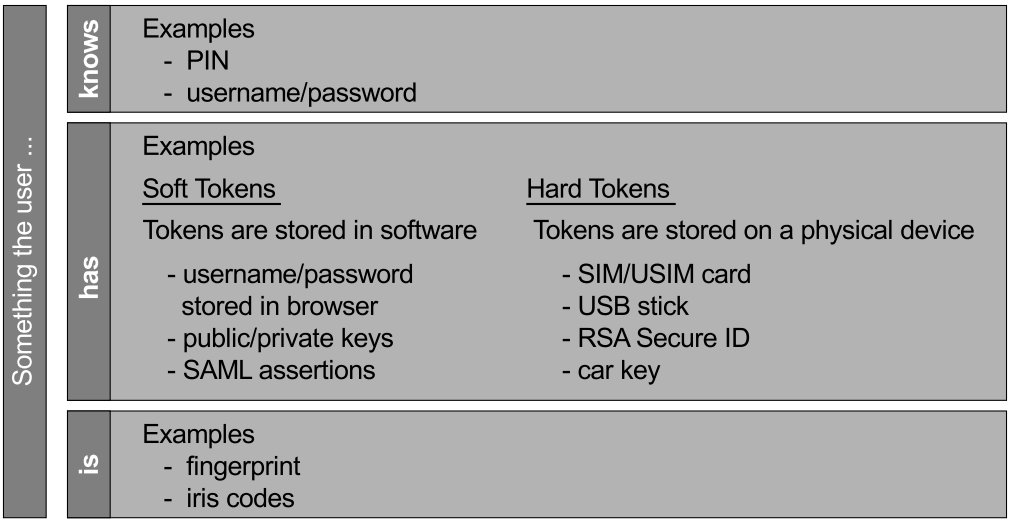


Figure 2.6: Overview on Credential Types

Something you are utilizes characteristics of the supplicant. In this case the supplicant is human and characteristics are biometric properties, like iris codes or fingerprints. For more details on biometric authentication it is referred to [And07, Cla11].

In addition, a fourth mean supporting authentication exists. *Something you can do* is required to distinguish between humans and robots that perform actions on behalf of somebody else. The human user is required to solve some kind of puzzle that is hard to solve for a machine but easy solvable for a human. Such puzzles are called CAPTCHAs[7] [HCR10, KZ09, BMM11]. Different kinds of CAPTCHAs protect web forms from the automatic submission, which might cause a complex computation and potentially might result in a denial of service (DoS) attack. CAPTCHAs have various weaknesses and do not represent an authentication mean on their own. Since it is not possible to identify an entity with CAPTCHAs, they should only be used in combination with other authentication means.

2.3.2 Authentication Mechanisms

A lot of different authentication mechanisms are in place. This section discusses (1) criteria to distinguish the application of authentication mechanisms and (2) introduces three authentication mechanisms in more detail: Password-based authentication, certificate-based authentication and authentication based on SIM cards. Different authentication mechanisms have advantages and disadvantages that make their application reasonable in dedicated contexts [O'G03, AW11]:

- OPEX and CAPEX: Each authentication method has costs. Hereby, operational expenditures (OPEX) and capital expenditures (CAPEX) can be distinguished. Some authentication methods require dedicated hardware, e.g. for fingerprint recognition appropriate readers are required, that has to be purchased (i.e. capital expenditures). On the other hand operation expenditures incur to deal for example with lost passwords.

[7] CAPTCHA is the abbreviation for Completely Automated Public Turing test to tell Computers and Humans Apart.

- Time and Effort for Authentication: If the authentication mechanism is complex and requires a lot of interaction with the user a lot of time and effort is required, resulting in a low usability. This has to be considered for biometric authentication [GMMJ11] and multifactor authentication (→ Section 2.3.4).

- Security: Different authentication methods have different relative security characteristics. This includes resistance against attacks, replacement of passwords/physical tokens and others.

- Compatibility: New security mechanisms might impose requirements on the existing infrastructure. This might result in comprehensive changes and high cost. (E.g. if existing printers do not support 802.1X, additional work flows have to deal with those printers or investments in new printers are required.)

2.3.2.1 Password Based Mechanisms

Authentication based on a username-password combination is the most applied authentication method [O'G03] from the perspective of the end user. Behind the scenes there are lots of differences with respect to the implementation of the authentication mechanism.

Strength of Password: The selection of appropriate passwords is a challenge [DMR10, KRC06]. From the system perspective passwords should to be unguessable, as long as possible and nowhere written down. Unguessable means that dictionary-attacks [PS02, Sei10] or similar attacks [KRC06] must not be possible. With increasing length of passwords the keyspace is increasing and thus the resistance against brute-force attacks. Large and unguessable passwords are often hard to memorize and in consequence written down, which is of course contradicting. If the user has to memorize several username-password combinations, this gets more challenging and led to the introduction of identity management systems (→ Section 3).

Transmission of Passwords: An authentication mechanism must not transport passwords in cleartext over an unencrypted transmission channel. An eavesdropper can easily intercept the password and impersonate the user. Passwords should only be transported across secure channels[8] (→ Section 2.4.4.2) or not at all. Alternative mechanisms for example rely on challenge response protocols (e.g. [RFC2195]).

Storage of Passwords: Storage of passwords with the authenticator represents another security threat. If possible, passwords should not be stored in cleartext. Depending on the authentication method it might be sufficient to store hash values of the password (e.g. in case of the Linux operating system).

2.3.2.2 Certificate Based Mechanisms

Certificates and corresponding public/private keys are in use for security-critical infrastructures and for example employed for the authentication against company virtual private networks

[8] Secure channels use mutual authentication, encryption and integrity protection.

(VPN) [Che01]. Various protocols (e.g. Transport Layer Security (TLS) [RFC5246], Secure Shell (SSH) [RFC4252]) support authentication based on certificates.

2.3.2.3 SIM-card Based Mechanisms

Each mobile phone contains a Subscriber Identity Module (SIM) for authentication against the network of the mobile operator. The SIM card is an electronic component that contains a shared secret that is only known by the mobile operator. Based on a challenge response mechanism the mobile phone authenticates with the shared secret against the mobile operator. In networks based on GSM technology, only the mobile phone authenticates against the mobile operator. With the introduction of UMTS technology, mutual authentication [Koi04, BHHN02] takes place to avoid International Mobile Subscriber Identity (IMSI) catching attacks [Mit01, MW04].

Even if the GSM authentication contains several security flaws [Lor03], the authentication infrastructure is outstanding. By the principle of roaming it is almost worldwide possible to authenticate against the mobile operator. In addition, the deployed SIM cards provide opportunities towards extensions. For example with the Generic Authentication Architecture (GAA) / Generic Bootstrapping Architecture (GBA) [3gpa] it is possible to support additional authentication methods towards 3rd parties, e.g. certificates [3gpb].

2.3.3 Selection of Authentication Mechanism

Selection of authentication mechanism is required in two different cases. First, if more than one authentication mechanism is available, the supplicant and the authenticator have to agree on one authentication method, which both support. Many existing protocols support more than one authentication method (e.g. TLS [RFC5246], SSH [RFC4252]) and provide a negotiation mechanism to select one of the supported methods. In a similar way many protocols (e.g. Internet Message Access Protocol (IMAP) [RFC3501], Lightweight Directory Access Protocol (LDAP) [RFC4511], eXtensible Message and Presence Protocol (XMPP) [RFC6120]) rely on the Simple Authentication and Security Layer (SASL) [RFC4422], which provides an abstraction layer for authentication itself and for the negotiation of authentication methods. If the authenticator and the supplicant have equal rights during the negotiation, downgrading to the weakest authentication mechanism has to be considered. For example, if both, the authenticator and the supplicant, support a certificate based mechanism and a username/password based mechanism, the supplicant might downgrade the authentication to the username/password based mechanism.

Second, if a service provider (SP) relies on a specific method the client has to use the requested authentication method. Reasons for a SP to impose specific authentication methods are driven by security needs. In particular if the SP does not perform the authentication on its own and relies on a 3rd party[9] the quality of the performed authentication is important. A mechanism for a SP to specify the requested authentication method is called Authentication Context [Aut05] (→ Figure 2.7). The SP maps such an Authentication Context (→ Section 2.7) with its own

[9]The 3rd party is the Identity Provider in Section 3.

domain-specific knowledge of the Level of Assurance (LoA) [SCLGS08] that gives confidence in the value of the performed authentication procedure.

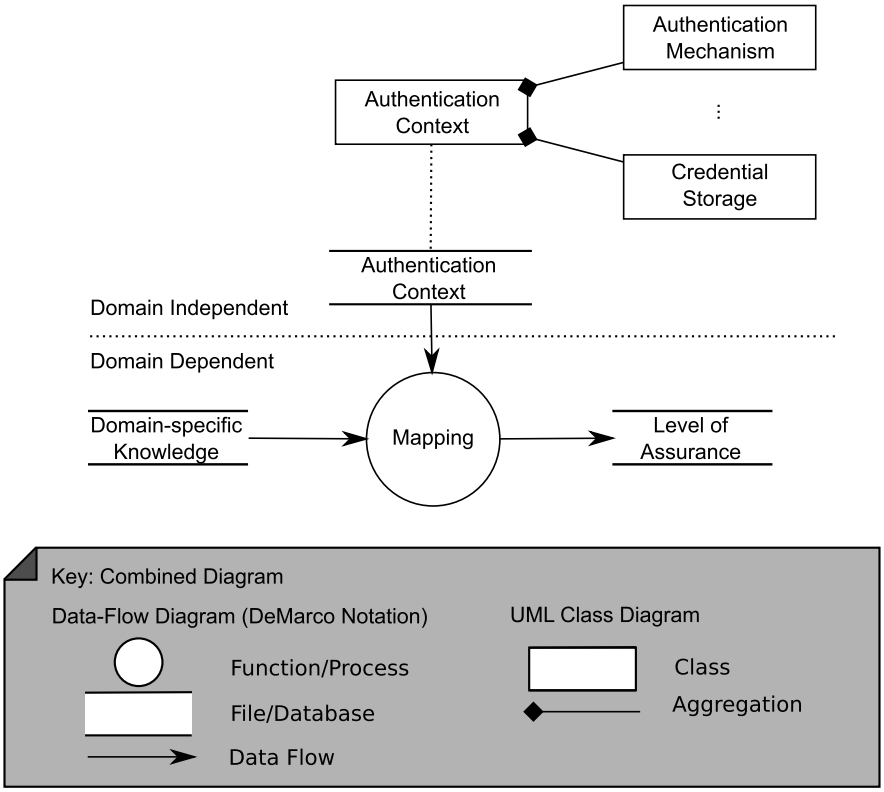


Figure 2.7: Authentication Context and Level of Assurance

2.3.4 Multifactor Authentication

Multifactor authentication means that the authenticator uses more than one authentication method [O'G03, Bea06] to authenticate the supplicant. Multifactor authentication provides the following degrees of freedom:

(1) Types of Authentication Methods: If the authenticator uses at least two authentication methods, the authentication methods can be identical or different ones. For example, if the supplicant requires the input of two different passwords, the same authentication method is used twice. Usage of different authentication methods, e.g. authentication with a smart card and a password, benefits from the individual advantages of each method and compensates the disadvantages.

(2) Order of Application: The authentication methods can be either applied in sequence or in parallel with respect to feedback to the supplicant. In case of sequential application, the supplicant gets immediately feedback whether the authentication was successful or not and proceeds with the next authentication method. In case of parallel application no feedback is provided, which of the authentication methods failed. The parallel application is more secure, because an attacker does not gain any knowledge about which one failed. From the perspective

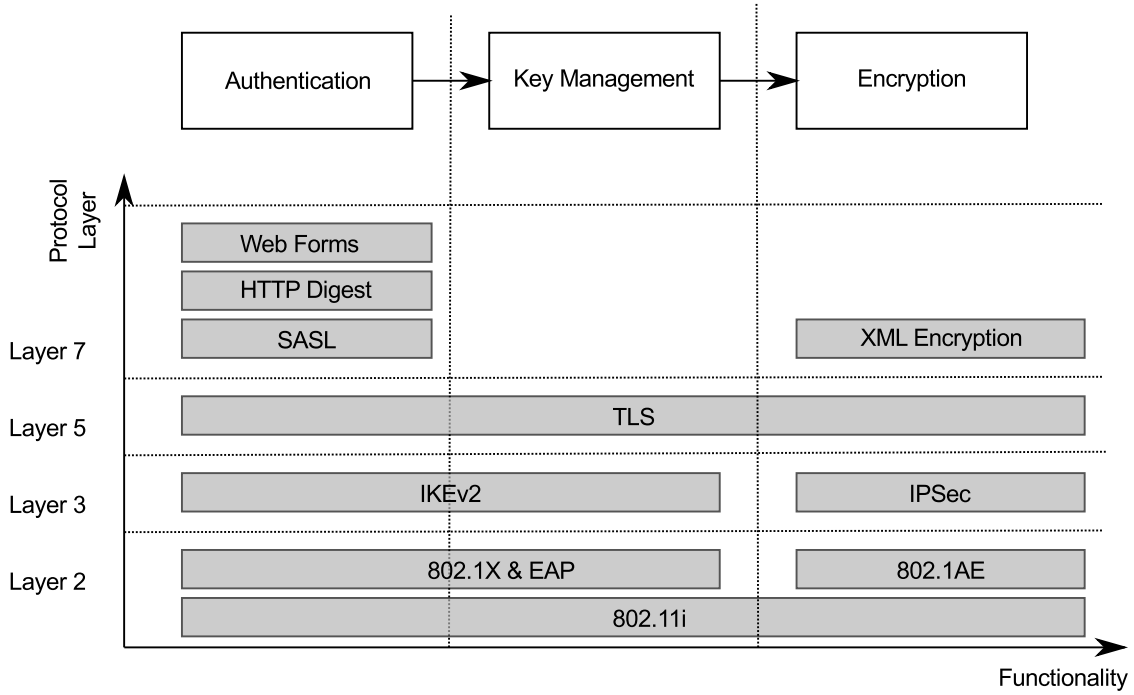


Figure 2.8: Security Functionality on Different Layers

of the authorized user, the sequential application provides a higher degree of usability, because he can identify the failed method.

If properly implemented, it is commonly agreed that multifactor authentication increases the overall security [Hen06] with the drawback of decreased usability [GMMJ11, AW11].

2.4 Security Protocols on Different Layers

Security protocols are the concrete realization of security mechanisms. Since each layer of the ISO/OSI stack [TW10] requires security mechanisms, various protocols have been specified. [SRC84] examined the motivation to have security protocols on different layers. In the following, the classification in Section 2.4.1 breaks down existing security protocols according to the provided functionality. Afterwards, Section 2.4.2 to Section 2.4.4 introduce security protocols that are relevant for the remaining thesis.

2.4.1 Classification

Figure 2.8 separates the functionality of existing security protocols that are applied on different layers of the ISO/OSI stack into three function classes: Authentication, Key Management, and Encryption. Often real world protocols integrate the functionality of different function classes. For example TLS [RFC5246] and 802.11i [IEEc, Ben10] integrate all three classes into single protocols. Figure 2.8 indicates the layer for which the security protocol provides the corresponding functionality. It is not mandatory that the security protocol operates on the same

layer. For example the IKEv2 [RFC4306] protocol works on top of the UDP protocol, i.e. on the transport layer.

2.4.1.1 Authentication

Authentication protocols make it possible to establish confidence into the identity of a user. As already introduced above various authentication methods exist. An authentication protocol defines the information and the order of the exchanged information for (mutual) authentication. In addition to pure authentication protocols, authentication frameworks exist that support more than one authentication protocol. Table 2.1 gives an overview of security protocols that provide authentication functionality.

Table 2.1: Examples for Authentication Protocols

Protocol	Explanation
802.11i	Integrates various authentication mechanisms, e.g. shared keys, and provides functionality for key management for the application in Wireless LANs [IEE07]
SASL [RFC4422]	Provides a framework for various authentication protocols
IKEv2 [RFC4306]	The Internet Key Exchange version 2 (IKEv2) provides authentication and key management functionality. It is used in conjunction with IP Security (IPSec) [RFC4301]
802.1X [IEEa] and EAP [RFC3748]	→ Section 2.4.2
TLS	→ Section 2.4.3
Web Forms	→ Section 2.4.4.2
HTTP Auth	→ Section 2.4.4.1

2.4.1.2 Key Management

A key management protocol is responsible for the negotiation of the applied encryption algorithms as well as for the establishment of a shared symmetric key, e.g. by means of the Diffie-Hellmann algorithm [DH76]. Table 2.2 shows examples for security protocols that provide key management functionality.

Table 2.2: Examples for Key Management Protocols

Protocol	Explanation
802.11i	→ Table 2.1
EAP	→ Section 2.4.2
IKEv2	→ Table 2.1
TLS	→ Section 2.4.3

2.4.1.3 Encryption

For the exchange of encrypted messages, it is required to identify the corresponding security association. This is typically achieved by additional header fields. Table 2.3 enumerates examples for protocols that apply encryption.

Table 2.3: Examples for Protocols that use Encryption

Protocol	Explanation
802.11i	→ Table 2.1
802.1AE	IEEE standard for the encryption of Ethernet frames
IPSec	IETF standard for the encryption of IP packets
TLS	→ Section 2.4.3

2.4.2 802.1X and EAP

The combination of 802.1X and the Extensible Authentication Protocol (EAP) is one example for a security solution on the data link layer. It is provides network access control of user devices in Ethernet networks.

802.1X defines the transport of EAP message as well as the concept of port based access control. First, the switch port to which the user device is connected works with restricted connectivity. All frames are discarded with exception of frames containing EAP messages. After successful authentication by one of the EAP authentication methods, full network access is granted.

Figure 2.9 shows a typical scenario for the application of 802.1X and EAP. The user's device, which has the role of the supplicant, is connected via Ethernet to a switch, which has the role of the authenticator. Both use 802.1X EAP over LAN (EAPOL) for the exchange of EAP messages. The authenticator decapsulates the EAP messages, encapsulates them in Diameter [RFC3588] or Radius [RFC2865] protocol messages and forwards them to an authentication server. Typically, the authentication server connects with a directory containing user accounts to verify the presented authentication credentials. Often an LDAP server realizes the user directory.

2.4.3 Transport Layer Security

The Transport Layer Security (TLS) [RFC5246] protocol provides authenticity, integrity protection as well as confidentiality on top of the transport layer. TLS is the successor of the Secure Socket Layer (SSL) protocol [FKK96] that has been developed and introduced by Netscape with its browser Netscape Navigator. With the rise of the Web for the exchange of information between different parties the need for security on the transport layer became obvious. In particular to enable e-commerce it was necessary to enable secure communication between customer and retailer.

TLS is a flexible protocol on top of TCP. It has the following key characteristics:

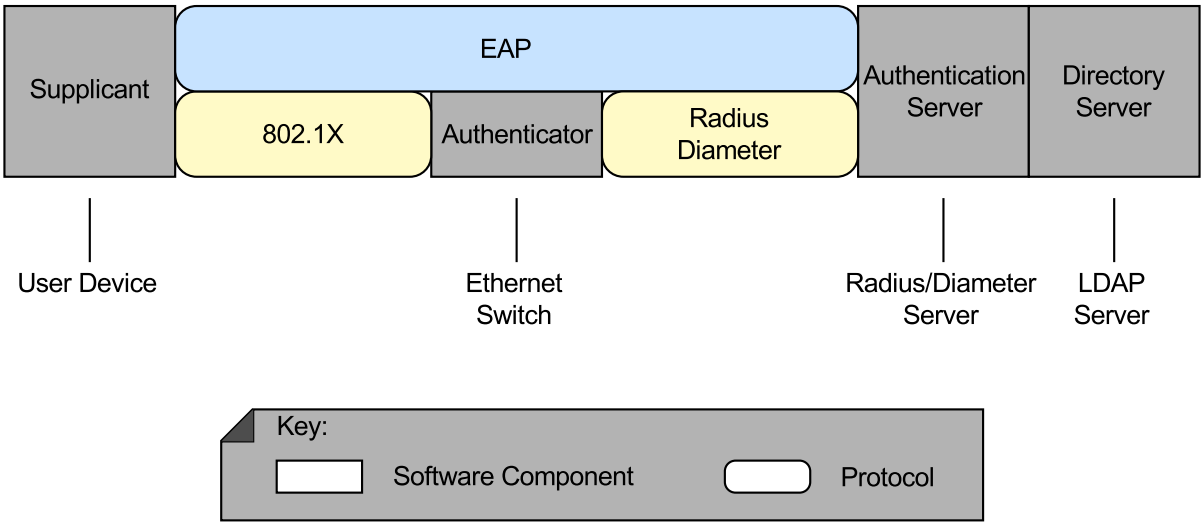


Figure 2.9: Application Scenario for 802.1X and EAP

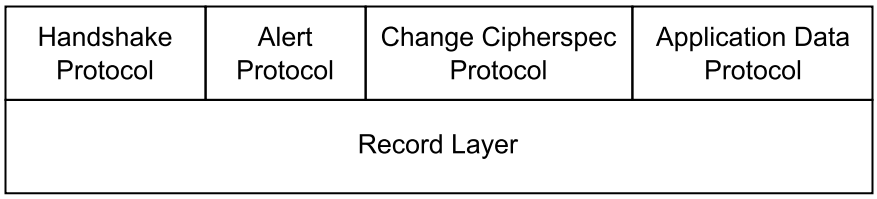


Figure 2.10: Structure of TLS Protocol

- Establishment of secure communication channels
- Authentication of communication partners (server only or mutual)
- Negotiation and selection of encryption algorithms
- Resumption of secure sessions

TLS is a layered protocol (→ Figure 2.10). It consists of the so-called Record Layer that is responsible for demultiplexing four different subprotocols on top:

- TLS Handshake protocol: Establishes or resumes the secure channel, i.e. authentication of communication partners and selection of encryption algorithms.
- TLS Alert protocol: Informs the communication partner about erroneous conditions. In some cases the secure channel is closed immediately after sending the message.
- TLS Change Cipherspec protocol: Informs the communication partner that all following messages will be encrypted using the specified cipher algorithm.
- TLS Application Data protocol: Transports the application data. This protocol is only needed to distinguish application data from the control protocol data units (PDU) of the TLS Handshake, Alert, and Change Cipherspec protocol.

Figure 2.11 illustrates the basic message exchange for the establishment of a secure channel and shows the interworking of the introduced subprotocols. Upon successful establishment of a TCP

connection, the client sends a *CLIENT_HELLO* message to the server and informs the server about the supported encryption algorithms (→ *CIPHER_SPEC*). The server responds with a corresponding *SERVER_HELLO* message including the supported encryption algorithms and a *SERVER_CERTIFICATE* message containing the X.509 certificate[10] of the server. If the client verifies the certificate successfully, it responds with a *CHANGE_CIPHER_SPEC* message to inform the server about the selected encryption algorithm and provides required keying material (→ *CLIENT_KEY_EXCHANGE*). Finally, the server confirms the encryption algorithm and terminates the handshake with a *FINISHED* message. Eventually, the client and the server have established an authenticated and encrypted connection.

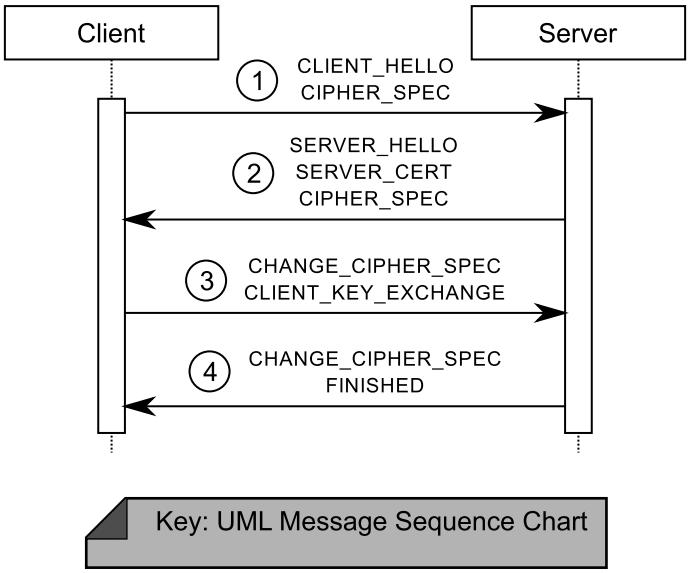


Figure 2.11: Establishment of a TLS Connection

Today, TLS is the de facto standard for securing the exchange of information between different parties in the World Wide Web. The security TLS provides strongly depends on the verification of the X.509 certificate that the server provides during the initial handshake. The client has to successfully verify the provided certificate in order to prevent MitM attacks.

2.4.4 Application Layer Authentication

On the application layer three authentication mechanisms are prevalent. In the WWW Hypertext Transfer Protocol (HTTP) authentication (→ Section 2.4.4.1) and the authentication by web forms (→ Section 2.4.4.2) are dominating. Other services on the application layer, like email (IMAP [RFC3501], SMTP [RFC5321]) or instant messaging (XMPP [RFC6120]), use SASL [RFC4422].

[10] Including the complete certificate chain

2.4.4.1 Hypertext Transfer Protocol Authentication

HTTP authentication [RFC2617] extends HTTP with mechanisms to identify and authenticate users. It specifies two authentication protocols: HTTP Basic Authentication and HTTP Digest Authentication.

HTTP Basic Authentication (→ Figure 2.12(a)) represents the most simple authentication protocol. If the user requests a protected resource, the web server responds with a "401 Authorization required" message. This message forces the client (i.e. the web browser) to display an authentication dialog to enter the username and the corresponding password. The client attaches the username and the password cleartext[11] to all following messages to the web server. This is a severe security weakness, because the username and the password can be easily intercepted. In addition, HTTP Basic Authentication is vulnerable to MitM attacks. In consequence, HTTP Basic Authentication should only be used on top of encrypted and authenticated communication channels, e.g. on top of TLS. By adding the username and the password to every client request, the client and server implicitly create a session.

HTTP Digest Authentication avoids the transfer of cleartext passwords. It creates a hash value[12] of the concatenation of the username, the password and a couple of other pieces of data (e.g. nonces, to prevent replay attacks) and attaches the hash value to all messages. (→ Figure 2.12(b)). The client recalculates the hash value for every request with a nonce provided by the server. Thus replay attacks can be prevented. However, HTTP Digest Authentication is still vulnerable to MitM attacks and should only be used on trusted networks or on top of a protocol that authenticates the web server, e.g. TLS.

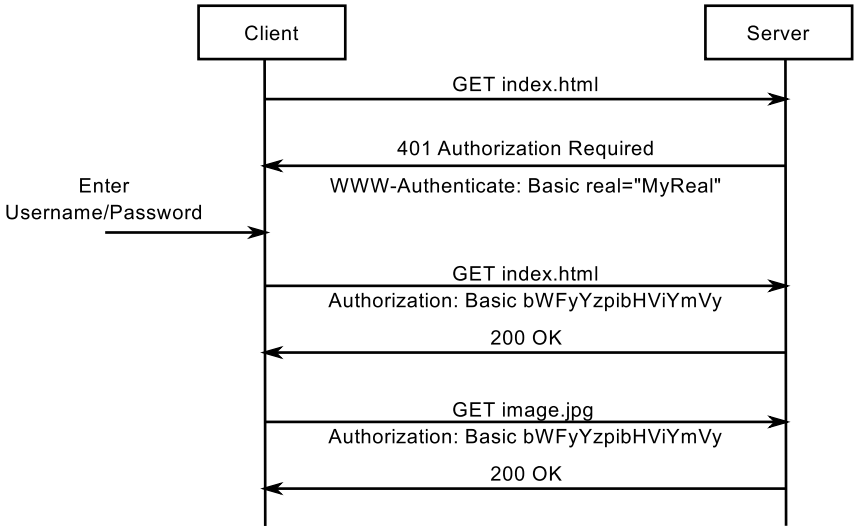
2.4.4.2 Authentication by Web Forms

Web forms are part of Hypertext Markup Language (HTML) [w3c99] based web pages and allow the user to enter information and submit them to the web server. This mechanism also allows authenticating users by means of username and password. Since the information entered in web forms is not encrypted, it is required to use TLS as the underlying protocol. TLS provides the additional advantage that the web browser can authenticate the web server before submitting information. If the client successfully verifies the presented certificate, it is possible to prevent MitM middle attacks.

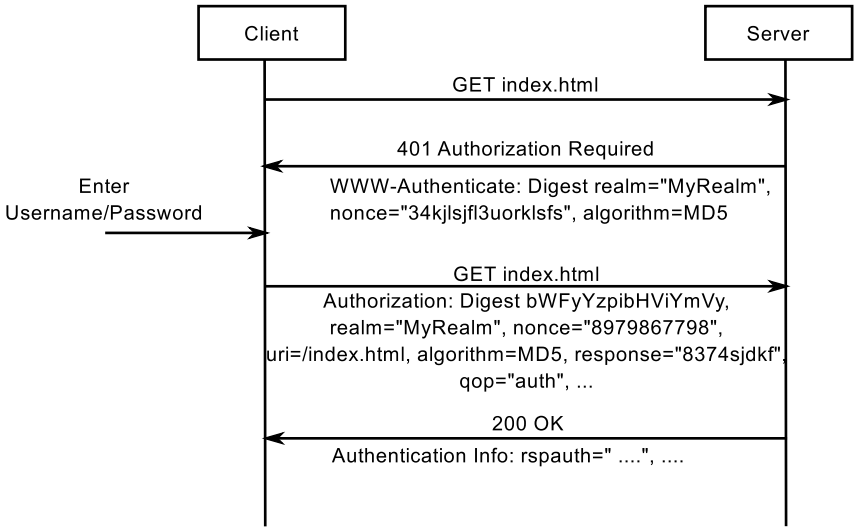
After successful authentication, the web server issues a so-called cookie [RFC2965], which the web browser stores either persistently or volatilely. The cookie is attached to all future requests to the same web server and allows the setup of a session between a client and server. The transmission of cookies has to be protected to avoid potential attacks by cookie stealing [WW10]. TLS provides adequate protection of cookies. With a stolen cookie an attacker is in the position to hijack a web session and consume services on behalf of the original user.

[11] The username and the password are concatenated and Base 64 [RFC4648] encoded, i.e. the username and password can be decoded without additional knowledge.

[12] HTTP Digest applies the Message Digest algorithm number 5 (MD5).



(a) HTTP Basic Authentication



(b) HTTP Digest Authentication

Figure 2.12: HTTP Authentication

2.4.4.3 Simple Authentication and Security Layer

To avoid that every application layer protocol that depends on authentication implements its own authentication mechanisms, [RFC4422] defines SASL. Various application layer protocols (e.g. IMAP [RFC3501]) integrate SASL. Using SASL has the advantage that it provides well examined authentication mechanisms and thus avoids protocol design and implementation errors.

2.5 Design and Evaluation of Secure Systems

Industry, standardization bodies and research proposed many methods to incorporate security into the system design. Moreover various organizations proposed methods and best practices to evaluate the security of existing systems. Section 2.5.1 provides a classification of these methods and Section 2.5.2 continues with selected methods and standards for secure system design. Section 2.5.3 and Section 2.5.4 introduce approaches for security evaluation from the standardization and from the research perspective. Finally, Section 2.5.5 introduces the approach that is applied within this thesis for the design and evaluation of system security.

2.5.1 Classification

Existing methods can be classified according to the following criteria:

- Category: To which category does the appplied method for the design and evaluation of the system belong?
- Phase of Development Lifecycle: During which phase of the development lifecycle can the method be applied?
- Audience: Who are the users of the method?
- Degree of Formalism: How much formalism is applied?
- Manageability: How much effort is required to apply or to reapply the methods?
- Usefulness: How usable is the method?

Category: [Bas93, Sip05] categorize methods for security evaluation into three and five generations, respectively. Since the term generation implies that the $(x+1)^{th}$ generation supersedes the x^{th} generation, the term category replaces in the following the term generation to indicate that all generations of methods are still in use. Figure 2.13 shows an adapted version of the three category model introduced by Baskerville [Bas93]. The categories partially reflect the historical evolution, i.e. at least one method of the 1st category has been known before any method of the 2nd category.

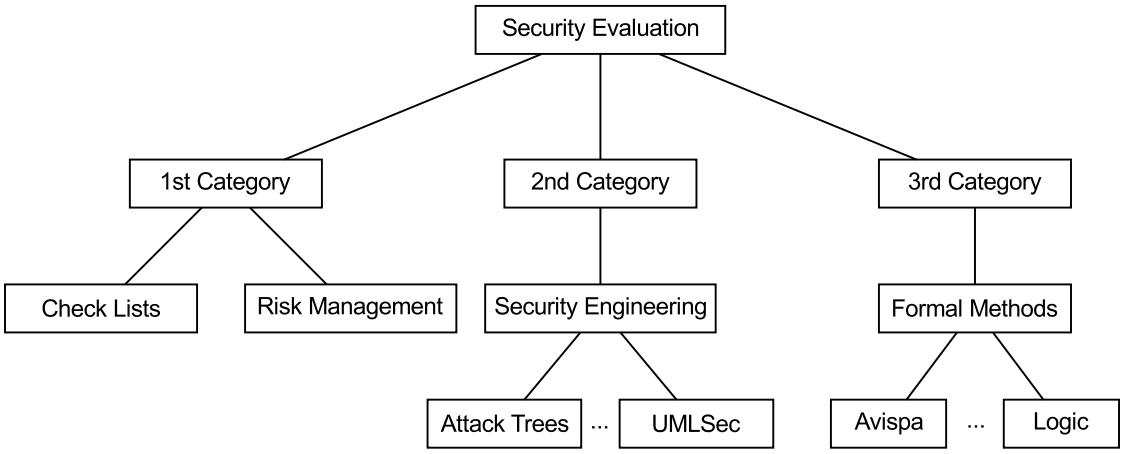


Figure 2.13: Overview of Security Methods

- 1st Category: Contains rather strict process models. Among them are simple check lists that consider for example the pure existence of certain security functionality.
- 2nd Category: Represents mechanistic engineering methods that have among others been inspired by methods of requirements engineering. Examples for such methods are Schneier's attack trees [Sch99] or the UMLSec approach [Jür04].
- 3rd Category: Formal methods allow the verification of security properties with the help of formal models that represent an abstract view on the original system.

Phase of Development Lifecycle: Evaluation methods can be applied in various phases of the development lifecycle, which distinguishes the following phases:

- Requirement Specification Phase: During the requirement specification phase, security requirements are defined. The latter evaluation method can guide the definition of security requirements. For example, it might be necessary to document security requirements in an appropriate form or to define requirements on the evaluation itself.
- Design and Implementation Phase: During the design and implementation phase, it is has to be verified that the specified security requirements are correctly considered. Moreover, the design and implementation process directly impacts the evaluation. For example for the evaluation it might be necessary to document the implementation appropriately or to have early prototypes.
- Test and Evaluation Phase: In addition to the fulfillment of functional and security requirements, additional tests, e.g. penetration tests, are the prerequisite of successful evaluation of productive systems.

Audience: During the different phases of the development lifecycle, different stakeholders exist that have different interests. Is it is possible to distinguish three stakeholders:

- User (Customer): The user of the system or the client that ordered the system influences the development as well as the later evaluation. The user is involved in the requirement specification as well as in the setup of evaluation goals.

- Developer: The developer realizes the security requirements by an appropriate system design. Moreover, he creates the required documentation and complies with specified development processes to fulfill the prerequisites of a successful evaluation.

- Evaluator: The system evaluator is involved after the system has been finished and takes the system as well as all processes that have been used to create the system and the corresponding documentation into account. After successful evaluation the product might become certified according to well-defined criteria (→ Section 2.5.3).

Formalism: The design and evaluation methods differ in the applied formalism. Basically, it holds that the more formal the method the more effort is required.

- informal: The requirements, the design as well as the evaluation is not well documented and specified.

- semi-formal: During all phases of the system development, specifications and documentation are created which allows the reconstruction of all steps. In most cases process models guide the creation of all specifications. An example for a semi-formal notation used during the design of systems is the Unified Modeling Language (UML) [UML09].

- formal: Formal methods use mathematical methods to represent the system or parts of it. [BH06] distinguishes three different levels of formalization according to their usage within the development lifecycle. From the lowest level of formal specification to the highest level of machine-checkable proofing, the effort to establish the formal model and the possible proofs increases. That means if more proofs should be possible, a more formal and more comprehensive description is required. The formality and the comprehensiveness of the model directly results in more effort and thus cost. Moreover, the creation of a formal model represents a challenge on its own regarding correctness. [PLÓCGS11] showed that formal methods and the corresponding proofs are only feasible with abstraction. And even with abstraction it was only possible to prove simple statements. [PW04] pointed out additional problems regarding the modularity and layering of security mechanisms and the resulting dependencies between the layers.

Manageability: The manageability represents the effort to apply a method. If it is not easy to integrate the method into the development process, the manageability would be very low. For example, if the method requires a completely decoupled model of the system, a lot of effort would be required to keep the implementation and the model consistent.

Usefulness: The usefulness of a method depends on the achievable results in relation to the required effort. For example some formal methods have a lack of usefulness, because the abstracted model is too far from the original system and is thus simplified and restricted.

2.5.2 Approaches for Secure System Design

System design has to consider security from the beginning of the system design [DS00, MGM03]. Considering security as an afterthought is not an appropriate design decision due

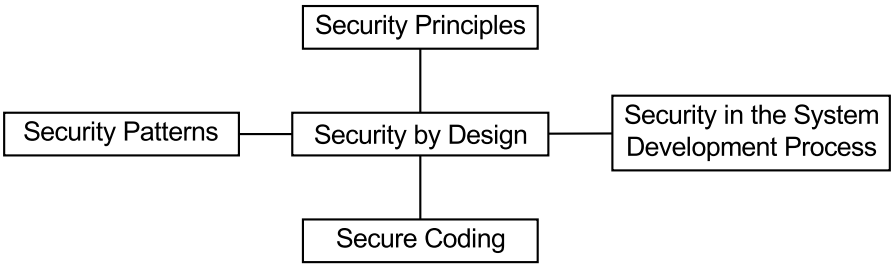


Figure 2.14: Secure System Design

to several reasons. First, security and security functionality leads to additional functional and non-functional requirements. Consideration of additional and changed requirements is known to be a critical factor of the success of software projects [Cha05]. Second, considering security from the beginning allows the consideration of design decisions with respect to their security impact. It enables the early identification and mitigation of threats. Finally, the interconnection of systems (e.g. by the Internet) made it essential to consider security from the beginning. Systems are not running independently from each other and not in isolated environments anymore. Recent attacks on industrial communication systems that got interconnected substantiate the need for security [CAN11, Lan11].

Figure 2.14 shows four essential concepts to consider security during the system design: Security Principles, System Development Processes, Security Patterns and Secure Coding. The following subsections introduce each of the concepts and provide references on related work.

2.5.2.1 Security Principles

Security principles are the foundation for all decisions within the design process. [SS75] described as one of the first design principles for the protection of computer systems and thus for secure system engineering. The eight postulated design principles are still valid today [Mei08, BG05, Bis04] and should underpin the design of ICT systems.

- Economy of Mechanism: Designed security functionality should be as simple as possible. This simplifies the validation of the correctness of the provided functionality and thus the robustness against attacks.
- Fail Safe Defaults: By default, a secure system should not permit anything. An explicit action must be necessary to grant the required permissions.
- Complete Mediation: There must be no possibility to access resources without appropriate permission checking.
- Open Design: It must be assumed that an attacker has access to the system design and architecture, and to the corresponding source code. This knowledge must not be exploitable by an attacker. In other words, the security of a system must not depend on the confidentiality of the system or mechanism itself, i.e. security by obscurity.
- Separation of Privilege: For security critical actions, separation of privileges is an appropriate mechanism. Privileges can be separated across different persons or mechanisms.

For example it may be necessary to enter a password to trigger an action and confirm the process in an additional step.

- Least Privilege: A user should only have the permissions that are actually required. Additional and in particular not required permissions create additional security threats.

- Least Common Mechanism: Mechanisms should be designed in a way that implicit interactions between different parties are only required if absolutely necessary. An implicit interaction between different parties enables hidden channels.

- Psychological Acceptance: User must accept security mechanisms. For successful security mechanisms it is essential that these mechanisms provide a high degree of usability.

[Mei08] suggests extending these principles with:

- Don't Trust User Input: Sanity checks are required on all entered user data. This becomes in particular obvious and painful with web applications and SQL injection attacks [WW10].

- Assume External Systems are Insecure: If external systems or communication partners are not known one has to assume that they are insecure and not trustworthy.

- Reduce Surface Area: The number of interfaces that are exposed to the outside should be restricted to a minimum. Every additional interface is subject to attacks. This includes disabling features that are not required in order to keep the surface to a minimum.

2.5.2.2 Security Patterns

Based on the well known approach of software patterns [GHJV94], various authors [S$^+$06, SNL05] defined so-called security patterns. Security patterns reflect architectural principles to realize security functionality. Security patterns help to avoid common design flaws by reapplying well-known practices to design problems. Therefore, security patterns can be considered as the realization of security principles.

2.5.2.3 System Development Process

Software development is a complex process. To handle this complexity software development processes, like the waterfall model or SCRUM, emerged [Bal09, Som10]. These software development processes do not consider dedicated phases or tasks concerning security. Various proposals [HL06, NIS, C$^+$06, McG06, Dau10] address this shortcoming by extended software development processes that take security related activities into account.

Microsoft proposed the Security Development Lifecyle (SDL) [HL06]. It is a well-defined process for the development of secure software. It has been used for the development of several Microsoft operating systems. It covers various security activities. Among them is threat modeling using the STRIDE methodology [HLOS06]. STRIDE is the abbreviation for Spoofing,

Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privileges. STRIDE identifies the assets of a system with data flow diagrams [Bal09] and checks for all assets whether a threat exists. Hereby, threats according to the STRIDE abbreviation are systematically identified for all assets.

The Open Web Application Security Project (OWASP) has also identified the need to extend the software development process. It defined the Comprehensive, Lightweight, Application Security Process (CLASP) [C+06]. In addition to SDL it considers security metrics and provides an overview on common mistakes in software design.

McGraw defined the Touchpoint process [McG06]. It originates from a set of best-practices and integrates a risk management approach. From a vendor and community neutral perspective NIST specifies security considerations in the software development for federal agencies [NIS]. For a detailed comparison of the above introduced development processes, it is referred to [DWSB+09]. Moreover various contributions from the research community take secure software design into account [Jür04].

2.5.2.4 Secure Coding

One particular activity within the software development process is implementation. The designed architecture is realized by implementing source code in a dedicated programming language. Depending on the programming language it is important to consider secure programming techniques. In particular languages like C and C++ are subject to various attacks (e.g. buffer overflows [One96]). With the selection of another programming language (e.g. Java, C#) many vulnerabilities can be avoided.

If insecure programming languages are required, the usage of programming guidelines (e.g. [Sea09]) allows the prevention of the most prominent programming flaws. Moreover, code reviews or static code analysis tools [CM04] support the detection of existing vulnerabilities.

2.5.3 Standardized Frameworks for Security Evaluation

For evaluation of IT infrastructures and communication networks several standards exist. In 1983 US government published the Trusted Computer System Evaluation Criteria (TCSEC)[13]. It was the first standard for the evaluation of IT security and had the drawback that only the existence of security mechanisms was checked. It did not consider the strength of the mechanisms (e.g. whether the key length is sufficient). As Figure 2.15 illustrates, it was the ancestor for the development of IT security standards in particular in Europe. This heterogeneity of standards from different countries represented a major drawback for equipment and system manufacturers, because the security evaluation and the corresponding certification were necessary according to each standard individually. This led first to a homogenization within Europe by the publication of the Information Technology Security Evaluation Criteria (ITSEC) standard in 1991. Since 1998, it is globally superseded by the Common Criteria (CC) standard.

[13] It is also known as the Orange Book

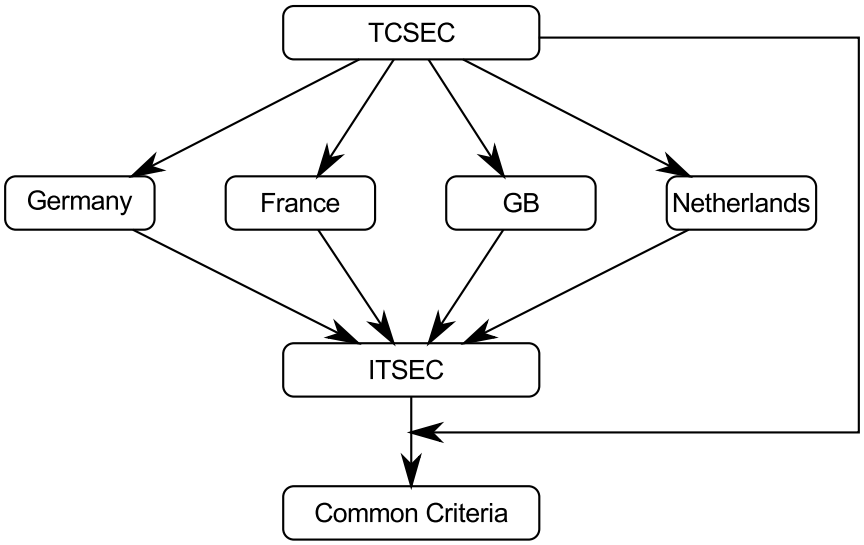


Figure 2.15: Historical Evolution of IT Security Standards

2.5.3.1 Common Criteria

The Common Criteria standard [CCPa, CCPb, CCPc, CCC] is the most important standard for security evaluation. The latest version, i.e. Version 3.1, represents the groundwork for the evaluation of IT products by certified organizations. In Germany, the German Federal Office for Information Security[14] (BSI) is responsible for CC evaluation. Due to harmonization reasons, Common Criteria is published as a set of ISO standards [ISOb, ISOc, ISOd].

For the evaluation of IT products, it considers not only the product itself but also the methods used during the development as well the produced documentation. It distinguishes seven Evaluation Assurance Levels (EAL). EAL 1 has the lowest evaluation requirements in contrast to EAL 7, which has the highest requirements. Table 2.4 provides an overview on the different EAL.

2.5.3.2 BSI IT-Grundschutz Catalogues

The German Federal Office for Information Security has published guidelines for the protection of information and communication systems [BSI05][15][BSI11][16]. These guidelines serve indirectly for a security evaluation. In combination with [BSI08a, BSI08b, BSI08c, BSI08d] it is possible to evaluate the security of an IT system.

2.5.4 Research Approaches

In research, various approaches for the evaluation of the system security exist, which can be applied at different stages of the system development.

[14] Bundesamt für Sicherheit in der Informationstechnik (BSI)
[15] Only an older version of the IT-Grundschutz Catalogues is available in English
[16] The latest version of the IT-Grundschutz-Kataloge is only available in German.

Table 2.4: Overview on Common Criteria Evaluation Assurance Level

EAL	Name	Requirements
1	Functionally tested	· Statement about security functionality
2	Structurally tested	· Security requirements derived from threat analysis · Functional and interface specification · Systematic testing
3	Methodically tested and checked	· Sound development practices · No substantial re-engineering
4	Methodically designed, tested and reviewed	· Modular design with separation of functionality · Security architecture description · Partial checks of implementation · Independent vulnerability assessment
5	Semiformally designed and tested	· Semiformal representation of design (UML models, ...) · Evaluation of covert channels
6	Semiformally verified design and tested	· Systematic evaluation of design
7	Formally verified design and tested	· Functional verification on low level

2.5.4.1 Model-Based Security Engineering

Model-driven software engineering is an approach to develop software by means of models. An abstract model is created that is stepwise refined towards running code. UML [UML09] is one technique to specify software systems by means of visual models.

UML does not allow the specification of aspects like performance and security. Therefore, such aspects cannot be considered during the development process. Several proposals exist that exploit the extensibility of UML to specify performance and security requirements and characteristics. Smith et al. [SW01, MS10] have developed an approach to annotate UML diagrams with performance requirements. Such annotations enable the evaluation of the system performance in all stages of the development process.

UMLSec [Jür04] extends UML to consider security requirements during the design phase and evaluates security properties of the design before implementation. Thus, security flaws and weaknesses can be detected and corrected in an early design stage. UMLSec specifies additional stereotypes, tags and constraints to annotate class diagrams, activity diagrams, deployment diagrams as well as message sequence charts. With these annotations it is possible to formally verify specified security properties with the support of tools. UMLSec is a promising method for systems that are developed from scratch or of limited size. In particular, it is well suited for the evaluation of cryptographic protocols [GHJW03]. In case of complex systems, i.e. many distributed components that make use of different protocols, using UMLSec would require to model the behavior of all components. For already existing systems, the effort to create a corresponding model is quite high.

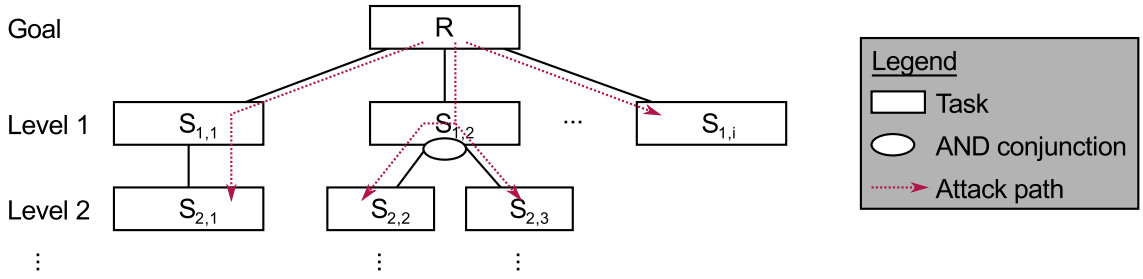


Figure 2.16: Exemplary Attack Tree

2.5.4.2 Attack Trees

The attack tree [Sch99] methodology has its foundation in the fault tree formalism [Eri99] developed at Bell laboratories. Fault trees allow the modeling of the reliability of systems or parts of a system. In contrast attack trees can describe the different ways to attack a system.

Attack trees are a hierarchical approach to model all steps required to render an attack possible. The root node R of an attack tree specifies the overall goal of a potential attack. Each subnode $S_{1,i}$ of the root node (i.e. level 1) represents one possible step to achieve the overall goal. For example, to achieve the goal R given in Figure 2.16, the attacker could perform any step $S_{1,x}$ with $x \in 1, 2, \cdots, i$. In turn, subnodes of level 2 detail necessary steps to achieve the corresponding step on level 1. With attack trees it is also possible to model AND conjunctions. To achieve step $S_{1,2}$, it is necessary to perform step $S_{2,2}$ and $S_{2,3}$.

Any path through an attack tree from the root node to the leaf nodes describes one possible attack. Each path requires a different effort to be exploited by an attacker and thus has a different exploitation probability. The effort per path can only be quantified by estimating the cost to accomplish the steps described by the leaf nodes. Based on the cost of the leaf nodes, the overall path costs can be calculated. The path with the minimum cost is the most likely attack path and dominates the potential risk to be attacked.

The creation of attack trees is difficult. In particular the completeness cannot be proven and strongly depends on the experience of the creator. As well, the quantification of the exploitation effort can only be estimated. Recent approaches have extended attack trees towards to attack graphs and their evaluation [OB06, HZO+11].

2.5.5 Design and Evaluation Approach

Figure 2.17 defines the process, which is applied within in this thesis, for the design of security features and the corresponding evaluation of the multi-device IdM architecture. This process is based on various best practice documents and adapted to our needs. The functional requirements derived in Section 4.3.3 and the usage and application scenarios introduced in Section 4.1 provide the first input for the elicitation and specification of security requirements in Section 4.4.5. This process starts with the identification of assets (→ Section 4.4.2). Based on the asset identification, the STRIDE methodology identifies threats (→ Section 4.4.3). Af-

terwards a prioritization of the threats according to defined criteria takes place. Based on the prioritized threads, corresponding security requirements will be identified in Section 4.4.5.

The design and implementation phase addresses all requirements. In case of security requirements, appropriate security mechanisms have to be identified, classified, and appropriate ones have to be selected.

The security evaluation consists of three different subphases that have to be passed. First, the established security requirements will be verified. Second, the existing assets are reconciled and extended if the designed architecture results in new assets. If new assets will be identified, a corresponding threat analysis needs to be performed. In the second subphase, the process identifies attack scenarios and analyzes them regarding existing security mechanisms. The third subphase evaluates the security of the designed system with respect to use cases that have not been in scope of the design. This is useful to identify potential limits of the system.

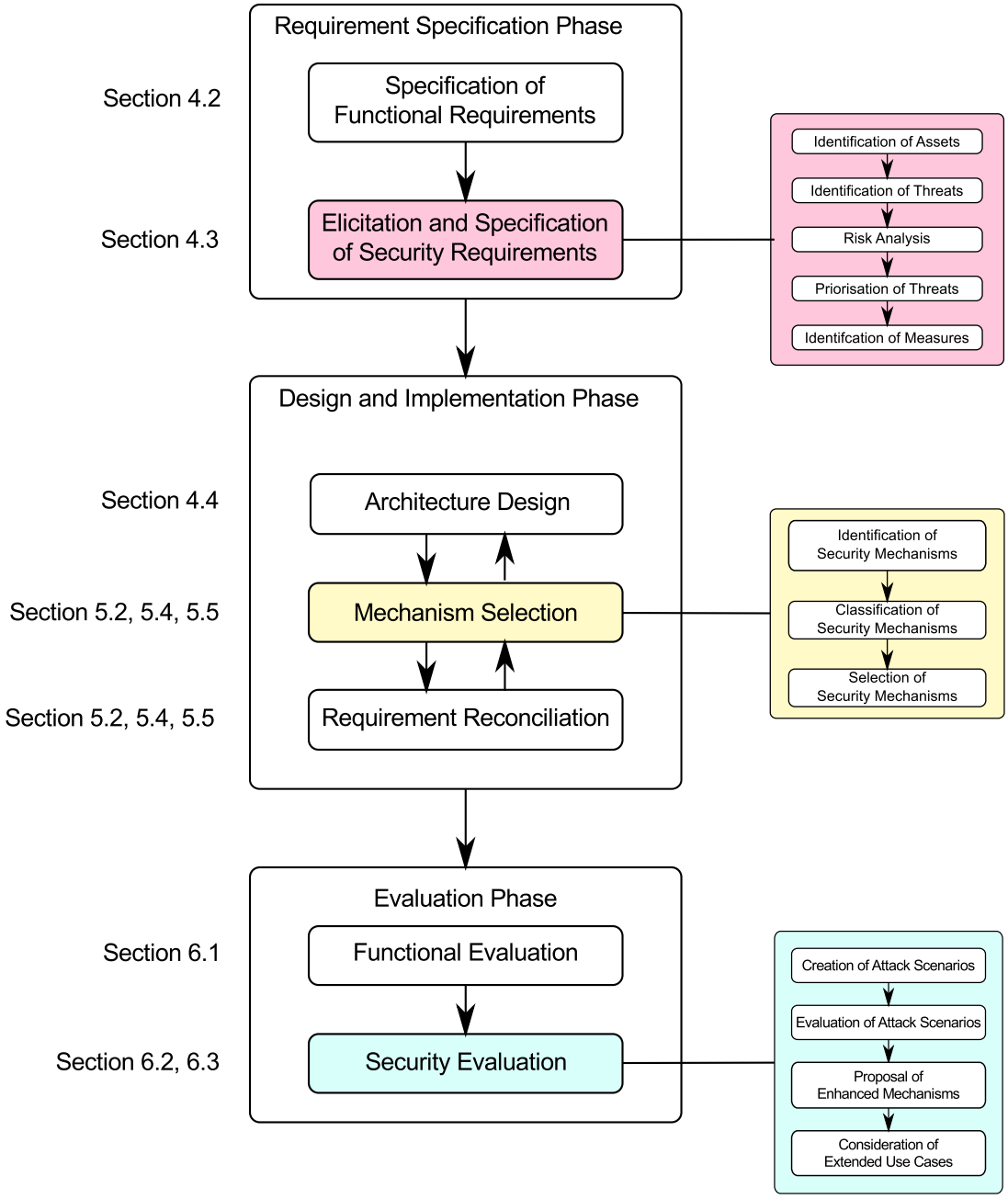


Figure 2.17: Phases of Security Engineering and Evaluation

3 Fundamentals of Identity Management

In the strict sense, identity management (IdM) is a set of security techniques that deals with the identification of identities in systems. Identification is required to grant access to resources. The identification is called authentication, if methods are applied to verify the identity (→ Section 2.3). In a wider sense, this chapter shows that IdM comprises additional aspects beyond technical security mechanisms.

IdM is important in different application areas. For instance, Internet SPs use IdM to identify customers. Companies use IdM to make services in a controlled way available to employees and partners. Recently, IdM got attention with the introduction of cloud systems [Old11] and the smart grid initiatives [ME10, DRHH10, FB11]. Hereby, IdM and the introduction of corresponding IdM systems have a high priority in the IT sector [Mes10] and among the various software vendors [Fol11].

Figure 3.1 illustrates the structure of this chapter. Section 3.1 introduces basic terminology, motivates the introduction of IdM systems and provides an overview on the different facets of IdM including non-technical aspects. The reference architecture in Section 3.2 focuses on the technical aspects of IdM systems and shows the work flows conducted between the different IdM roles that are relevant for this thesis. Section 3.3 provides an overview on the basic technologies that IdM systems rely on. These are the Security Assertion Markup Language (SAML) and the Web Service (WS) Federation framework. Based on this, Section 3.4 introduces existing IdM systems and Section 3.5 provides an overview on existing work regarding security and vulnera-

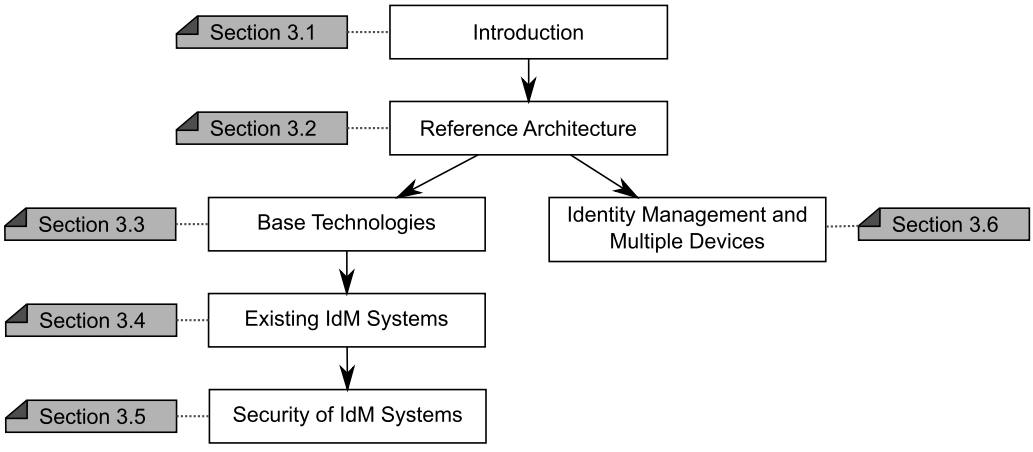


Figure 3.1: Chapter Outline

bilities of IdM systems. Based on the reference architecture, Section 3.6 discusses related work regarding the extension of IdM systems to multiple devices, which is the core of this thesis.

3.1 Introduction

After introducing IdM terminology in Section 3.1.1, Section 3.1.2 details the motivation for the introduction of IdM systems. Finally, Section 3.1.3 addresses facets beyond technology of IdM.

3.1.1 Terminology

As introduced above *IdM* deals with the identification of identities in technical systems. An *IdM system* represents the technical realization of IdM and provides concrete technical mechanisms for the identification of identities. This includes authentication protocols, identifier resolution, et cetera.

The term *identity* has different meanings depending on the considered area of research [Cam04, GV08, PH10]. The focus of this thesis is on technical aspects of IdM. Therefore, Figure 3.2 illustrates the used terminology with respect to identities. Entities request resources within technical systems. Entities are natural persons, machines, or software programs. In the following the focus is put on natural persons, i.e. users that request resources by means of devices (e.g. notebooks, smartphones). IdM systems represent each entity by digital identities, in the remainder called identities. An identity has an identifier and associated attributes [Win05] that reflect the characteristics of the corresponding entity. In case of natural persons, associated attributes can be the age or the postal address. Since private information is associated with identities, privacy has to be considered. Therefore, the term virtual identity [S+08] or partial identity [CK01, JKZ02] has been introduced to reflect the possibility to select an identity that fits most appropriately to the context (→ Section 3.2.2.5).

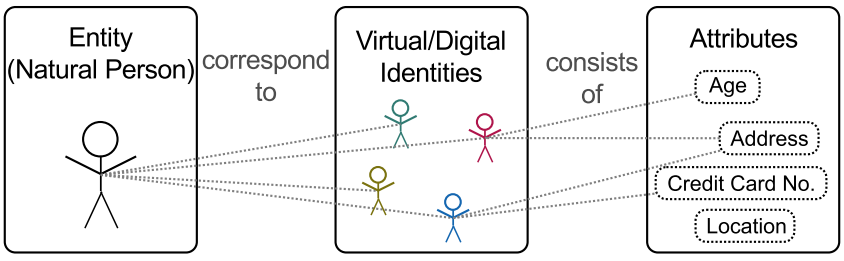


Figure 3.2: Natural and Digital Identities

3.1.2 Motivation

IdM systems are relevant in three different areas (→ Figure 3.3). In the Internet, users are faced with different services provided by different service providers (SP). Many SPs force users to create accounts, consisting of an identifier, a password and user attributes. The identifier, which is the username, and the password are used for authentication. According to the definition

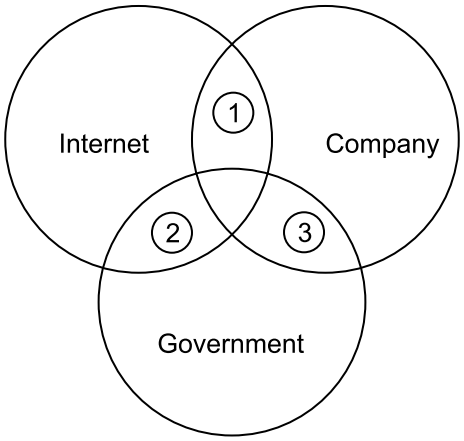


Figure 3.3: Different Application Areas of Identity Management

above, each account represents an identity. The SP's system to manage the identities is the corresponding IdM system. Without additional mechanisms, IdM systems of different SPs are isolated [J+05]. With an increasing number of SPs the user has to memorize an increasing number of identities. This results in two problems:

- *Usability*: Beside the memorization problem itself, the user has to individually authenticate for each service by typing the username/password combination or executing other authentication methods. This represents a decrease of convenience and thus a decrease of usability.

- *Security*: As a consequence of the memorization problem, users tend to reuse the same username/password combination with different SPs. This makes the user vulnerable to various attacks. Among them are impersonification attacks by malicious SPs, phishing[1] attacks and attacks against the SP's user database that often stores user passwords in cleartext, as recently exploited [Hun11, Gil11].

Federated IdM systems [Cha09] introduce a so-called identity provider (IdP). SPs trust the IdP, which performs the authentication on behalf of the SPs. This increases usability and security. There is an increase of usability, because several services can be used based on one authentication with the IdP and because the user does not have to memorize different accounts. Security is increased, because strong authentication mechanisms that are used with the IdP can protect the identity of the user. However, federated IdM introduces a privacy problem. The IdP obtains information about all SPs that are used by the user. Usage of several identities with the IdP can protect the privacy of the user [BNP+08].

Similar motivations hold for companies that provide several services and make them internally available to their employees. Within companies various departments are responsible for the operation of services. In consequence different authentication credentials are required for each service. Thus the usability and the security problem also hold for companies. In addition, the introduction of IdM systems can reduce the *administration effort* to manage different accounts for

[1] Pishing is a so-called portmanteau word consisting of "password" and "fishing"

the employees. Another focus of companies with respect to IdM are directory services and access management. Directory services like LDAP [RFC4150] or Active Directory [DRALN08] allow the maintenance of employee identities. Access management enables the management of authorizations based on concepts like role based access control (RBAC) [SCFY96]. If employees of other companies have to access the service, IdM systems represent a solution to reduce the administration effort.

Finally, IdM is important for governments. The issuance of passports is equivalent to the creation of identities, which comprise an identifier and associated attributes that are human-readable noted or electronically stored on the document itself [Sid08]. Identities that the government creates have additional value with respect to attributes, which have been validated by the government. Thus *assurance* is created that the attributes are correct. Depending on the country, 3rd parties (e.g. SPs) can access these attributes. Figure 3.3 shows that the application areas of Internet and Government (indicated as (2)), and Company and Government (indicated as (3)) get interconnected. For example, the new German Identity card [Pas10] provides such a feature. In addition to these three areas, IdM is the key for cloud computing [Old11]. Cloud computing[2] requires that companies connect their internal structures to the cloud, which is running somewhere in the Internet (indicated as (1) in Figure 3.3). Therefore, the cloud has to identify the employees in order to grant access.

3.1.3 Different Facets

IdM has more facets than the pure technical aspects of IdM systems itself. Figure 3.4 shows four additional facets that are not in focus of this thesis. (1) Society: Using identities and providing identity attributes to other parties has a direct impact on how other parties perceive ourselves. (2) Jurisdiction: Strongly related to IdM is privacy and thus data protection laws. Moreover, consequences of identity theft, i.e. the unauthorized usage of identities, are subject to jurisdiction [Bir07]. (3) Business: With IdM new business opportunities exist. The role of the IdP is valuable, because of the tight costumer relation. For an analysis of business models it is referred to [S$^+$10]. (4) Organization: With IdM it is possible to create new organization structures and distribute existing functionality in various new ways. [RGS$^+$10] has examined new structures.

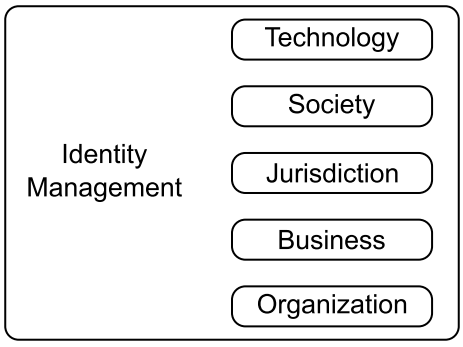
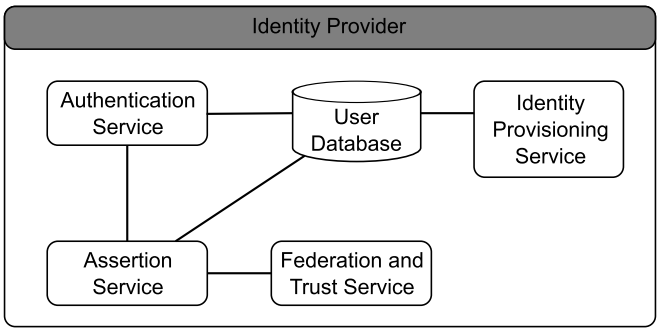


Figure 3.4: Different Facets of Identity Management

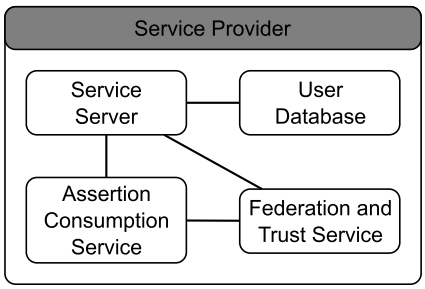
[2] Public clouds are assumed.

3.2 Reference Architecture

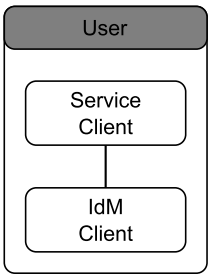
The reference architecture in Figure 3.5 defines an IdM system from a technical perspective. It abstracts from actual implementations by introducing functional blocks that highlight the most important IdM functionality. In contrast to the work from [Win05, Rad07], which served as input, the presented reference architecture focuses on the functional blocks rather than on the technologies used to realize those. The reference architecture distinguishes three roles that exist in an IdM system (→ Section 3.2.1). Between these roles different work flows take place (→ Section 3.2.2).



(a) Identity Provider



(b) Service Provider



(c) User

Figure 3.5: IdM Reference Architecture

3.2.1 Roles

IdM systems distinguish at least three different roles: User, Service Provider (SP), Identity Provider (IdP). The user has different identities and consumes services. The SPs provide services and rely upon authentication by IdPs. The IdP is responsible for the authentication of users and provides attribute services to the SPs.

IdP: The IdP authenticates the user's identities by means of the *Authentication Service*. The authentication service performs the authentication protocol and verifies the correctness of the exchanged information against the *User Database*. The user database contains the identities of its users, i.e. an identity identifier, corresponding user attributes, and meta-data about the user (e.g. date of last identity usage). The *Identity Provisioning Service* fills the user database. Upon creation of identities, it adds new entries to the user database. The IdP exchanges security-critical information, like the authentication status of identities and attribute data with SPs. Therefore,

preestablished trust relationships between IdPs and SPs protect the information exchange. This is subject of the *Federation and Trust Service*. Based on a trust relationship, a SP can request assertions about identities using the *Assertion Service*. An assertion is a statement about a user identity that is certified by the IdP. It may contain statements about the authentication status of the identity, i.e. whether an IdP has authenticated the identity and if yes by which authentication mechanisms, or statements about identity attributes, e.g. identity x is born on the 29.12.1940.

Basically it is possible to subdivide the role of the IdP into several subroles. [BNP+08] describes a model that splits the IdP into the role of the Attribute Provider, the Authentication Provider and the Identity Aggregator that bundles the two aforementioned roles. This subdivision is not relevant for this thesis and thus not further detailed.

SP: The SP provides services to its customers, i.e. to identities. The *Service Server* makes the actual service available. Regarding authentication, it relies on the *Assertion Consumption Service*, which consumes and verifies assertions created by a trusted IdP. A SP trusts an IdP if a preestablished trust relationship exists, which the *Federation and Trust Service* can verify. Since each SP has own requirements regarding identity attributes, a *User Database* augments the IdP's user database with the possibility to manage individual entries.

User: The user consumes services provided by the SP by making use of one of his identities. The authentication takes place against the IdP by means of the *IdM Client*. Upon successful authentication the IdP provides assertions to the SP and the user is in the position to consume services with its Service Client.

3.2.2 Workflows

Workflows detail the sequences of activities and message exchanges between the above introduced roles and put the motivation for IdM systems into practice. This thesis describes the following work flows:

- Identity Creation
- Single Sign-On
- Attribute Retrieval
- Single Logout
- Identity Selection
- Federation Establishment

3.2.2.1 Identity Creation

The identity creation is a workflow that takes place between the user and the IdP. The user provides the required attributes and obtains as a result an identity and everything needed to make use of the identity.

This process has many degrees of freedom and is realized in reality in very different ways. Depending on the realization of the identity creation process the identity has a different value, regarding reputation and assurance, i.e. how sure is the IdP to know the actual user behind the

created identity and the correctness of the provided information. It is important to distinguish between the identity creation process itself and the later authentication [Cha09]. If the identity creation process is weak it is questionable to have strong authentication mechanisms in place and vice versa. For example it is possible to create an identity with Google, which serves as IdP for 3rd party services [Goo11], without any verification of the user behind. On the other hand it is not possible in legal ways to obtain an identity from a cell phone operator, i.e. a mobile phone number, in Germany without showing an official identity card[3].

[NIS06] provides guidelines for the different ways of identity creation and distinguishes four different levels of assurance. The lowest level of assurance does not specify any rules, whereas the highest level requires the user to personally appear and show two different identity documents, one issued by the government.

The identity destruction workflow, which is the counterpart to the identity creation workflow, takes place upon destruction of an identity. Identity destruction is a challenging task [Win05, MZK+05] to guarantee the overall security. Existing identities that are not used anymore represent a security risk, because they can still be used to get access to services without that the original owner notices anything. In addition, information associated with identities has to be removed by IdPs upon identity destruction [Ber07].

3.2.2.2 Single Sign-On

Single Sign-On (SSO) allows the user to consume several services provided by different SPs without the need to manually authenticate against each SP. Figure 3.6 illustrates the principles of SSO. The user successfully authenticates with one of his identities against the IdP and establishes an IdP session[4]. The maintenance of the IdP session is subject to the actual implementation of the IdM system (e.g. by means of cookies, see Section 2.4.4.2). Based on an established IdP session, the user can request SP assertions that contain an IdP statement on the user's authentication status dedicated to the SP. The IdP uses the preestablished trust relation to the SP to dedicate the SP assertion to the SP. With the SP assertion the user establishes a SP session.

If the user wants to establish a second SP session with a different SP the existing IdP session is reused. In Figure 3.6, the user establishes two SP sessions based on one IdP session. Section 3.4 introduces selected IdM systems that provide SSO. SSO mechanisms have been around for more than twenty years. One of the first systems that provides SSO functionality was Kerberos [SNS88]. [Par95] provides an overview on single sign-on products available beginning of 1990s. SSO is still subject to research as this thesis and others show [LCGSG09, HJK08, CL12]. [Bar09] examines the performance implications of operating different identities.

[3] Article 111 of the "Telekommunikationsgesetz" [Tel04] regulates that companies offering telecommunication services have to collect amongst other information the name and the address of the user. This is typically achieved by the verification of the identity card. [Man10] describes illegal ways to circumvent this law. [G+06] provides a survey on the situation in 31 different OECD member states.

[4] It is assumed that the user has to authenticate each identity individually due to two reasons. First, it is possible that several IdPs exist. Second, the IdP should not be in the position to inherently link different identities of the user.

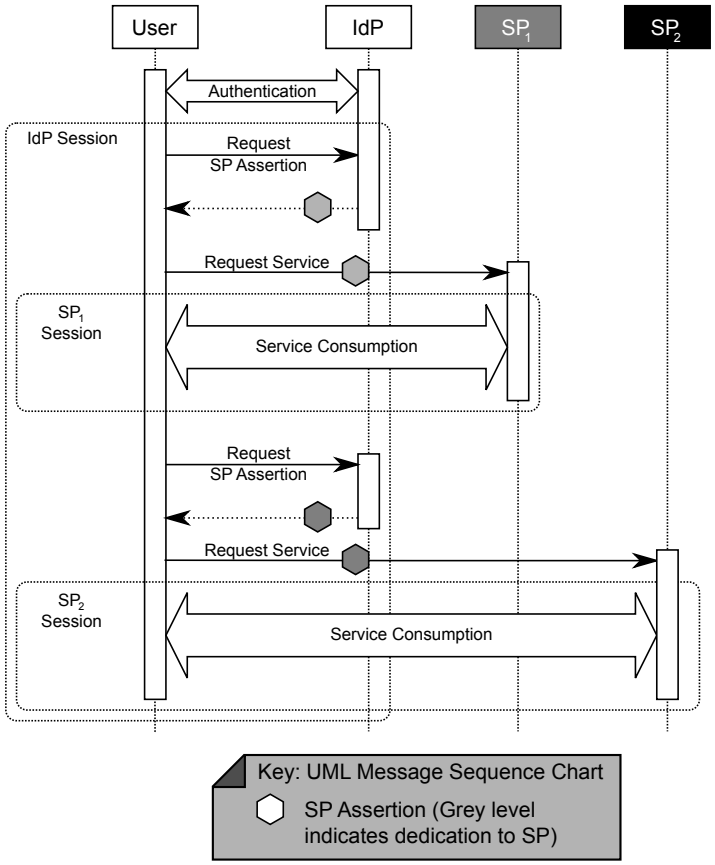


Figure 3.6: Principles of Single Sign-On

3.2.2.3 Attribute Retrieval

The IdP has a central role in IdM systems not only regarding authentication (c.f. SSO) but also regarding user attributes. It can centrally manage attributes associated with identities and make these attributes available to SPs. Thus the user does not have to individually provide attributes to each SP, which often means manually entering, but authorizes the SP to access the attributes with the IdP. Figure 3.7 shows that a SP can request identity attributes if a corresponding SP session exists. Current research on attribute retrieval targets on the aggregation of identity attributes. [CI09, TPL+10] propose mechanisms to provide access to attributes that are scattered across different IdPs and different storage locations.

3.2.2.4 Single Logout

For SSO the IdP creates a SP Assertion and thus knows which SP sessions the user intended to establish. The IdP uses this knowledge to terminate all SP sessions with a single user-initiated action, i.e. Single Logout (SLO). Figure 3.8 illustrates the principle of SLO. The user triggers the SLO for one identity with the IdP, which then performs the log-out with all SPs and finally terminates the IdP session. Many existing IdM systems provide SLO functionality. From the research perspective SLO is less interesting than SSO.

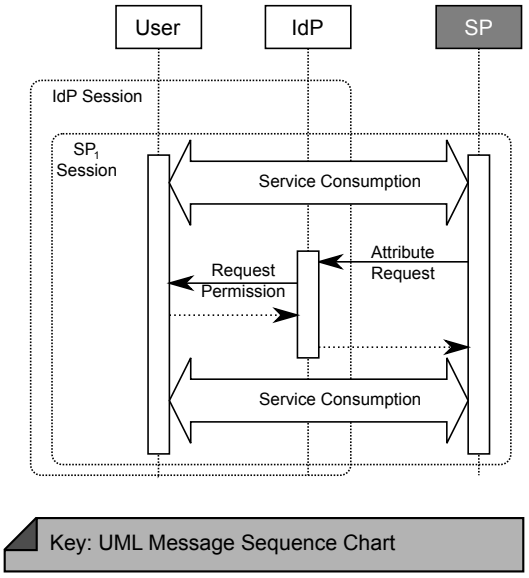


Figure 3.7: Principles of Attribute Retrieval

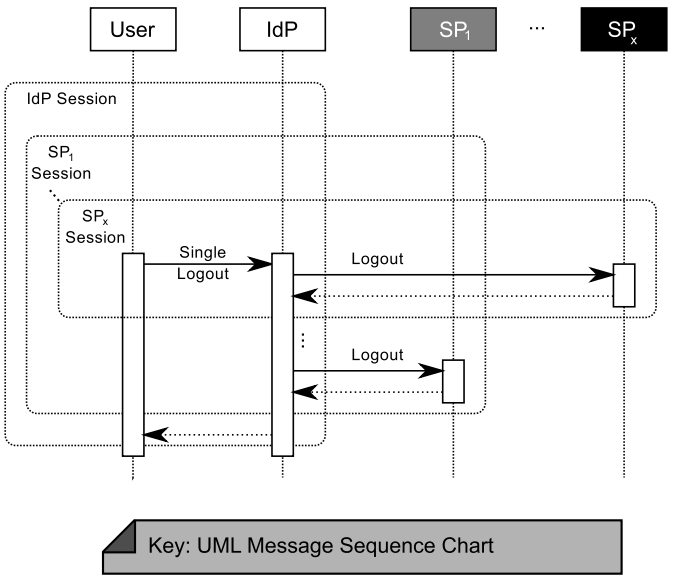


Figure 3.8: Principles of Single Logout

3.2.2.5 Identity Selection

A user can have more than one identity with one IdP. It is also possible to have identities with different IdPs. The concept of having more than one identity and selecting one identity is known as partial identities [CK01, JKZ02] or virtual identities [S+08]. Different motivations to have several identities exist. (1) Privacy Protection: In [S+08, CK01, JKZ02] a user has several identities to limit the view that a SP can obtain by observation of the identity attributes including the network characteristics [Hau08]. (2) Usage Context and Task Distribution: The usage context defines the circumstances in which an identity is used, e.g. private purposes, business purposes. Several identities render it possible to separate these usage contexts [GC07].

If a user has several identities, he has to select one identity that should be used to consume the intended service. Identity selection requires user interfaces that show relevant information for the selection decision. The Microsoft Cardspace IdM system provides a concept for identity selection (→ Section 3.4.3). Filtering and ranking algorithms allow adapting the choice a user has. Identities that cannot be used are not shown. For example, an identity cannot be used if the associated attributes do not match the requirements imposed by a SP. The remaining identities are ranked according to various criteria. If identity selection is applied as a privacy enhancing technology (PET), the ranking has to adhere to privacy metrics. For details on appropriate metrics it is referred to [Neu09].

Anonymous credential systems are a related concept to identity selection. They provide a set of techniques for users to use services without revealing their actual identity. With the usage of pseudonyms and the usage of anonymous credentials, users are in the position to convince SPs that they possess specific properties without revealing them. [Cha85] proposed such a system at first and various researchers (e.g. [CVH02]) refined it. Anonymous credential systems have a high complexity and are thus not used in practice. For further details on PETs it is referred to [Fri07, SP11, AGK03].

3.2.2.6 Federation Establishment

A federation is defined as "a group of organizations [...] that have joined together to form a larger organization [...]" [CBU08]. In case of IdM, the SPs and IdPs shape a federation that allows the exchange of identity information like the authentication status and identity attributes. Since the exchange of identity information is sensitive with respect to the privacy of users and the security in general (e.g. unauthorized service consumption) the parties within the federation must have a trust relationship.

Different types of federations for IdM can be distinguished and it is possible to define corresponding patterns [Win05, DFLP07]. The patterns differ with respect to the point of time at which the federation is established, the number of participants and the party that establishes the federation. (1) Point of time: An IdP and a SP can establish a federation at two points of time. Either they establish it independently of user activities or adhoc. Adhoc means that the IdP and SP establish the federation as soon as one user requires it. (2) Number of participants: A reasonable federation requires at least one IdP and two SPs[5]. With respect to the number

[5] In case of one IdP and one SP, the required effort to establish a separate IdP is not reasonable.

of IdPs within a federation two different patterns can be distinguished. In a “hub and spoke” federation one IdP exists and several SPs trust the IdP. In case of a circle of trust, more than one IdP can exist to render it possible that one entity can simultaneously carry out the role of an IdP and SP. (3) Establishing party: Either the IdP and the SP or the user triggers the federation establishment. The latter case is called user-centric federation. However, the dominating case today is that federation establishment takes place between IdPs and SPs.

A successful established federation results in the creation of metadata. The metadata contains certificates for the establishment of secure channels and information about connection points to perform SSO, SLO or attribute retrieval [C+05b].

The most successful federation of SPs and IdPs is the GSM network [GSM09]. In 2009 about 4 billion subscribers have been able to roam between different networks operated by different mobile network operators. Other examples for federations are eduroam that allows students to use WIFI services provided by universities around the globe [edu] or the DFN Authentication and Authorization Infrastructure (DFN-AAI) [Käh06].

3.2.3 Classification Criteria for Identity Management Systems

The application areas and the motivations for the introduction of IdM are manifold as introduced above. This results in different requirements on IdM systems and in consequence different realizations. The following classification criteria point out the degrees of freedom for IdM systems:

- Client Installation: Some IdM systems require that the user installs corresponding IdM software on his devices (e.g. Microsoft Cardspace [B+08]). Other systems exploit existing software like web browsers to perform all IdM workflows (e.g. Liberty Alliance [T+], Shibboleth [S+05]).

- Service Layer: IdM systems apply different protocols for the identification of users. These protocols are either targeted on network services (Layer 2/3 of the ISO/OSI protocol stack) or on the application layer (Layer 7 of the ISO/OSI protocol stack). Section 2.4 provided examples for authentication protocols on the different layers. Different initiatives [HHch, BSG08, LCGSG09] address the integration of IdM systems for the network and application layer.

- Openness: Some IdM systems and the corresponding protocols adhere to open standards (e.g. SAML [C+05a]) and allow the interoperability between products of different vendors. Others are proprietary and do not allow any kind of extensions.

- Privacy Features: Depending on the IdM system, the user has different possibilities to protect his privacy. For example, [B+08] allows the user to select an identity based on the contained identity attributes. [S+05] provides the user with the opportunity to restrict access to identity attributes by means of access control policies.

- Application Area: Existing IdM systems target different applications areas. Section 3.1.2 introduced the following application areas: Internet, Companies, and Government.

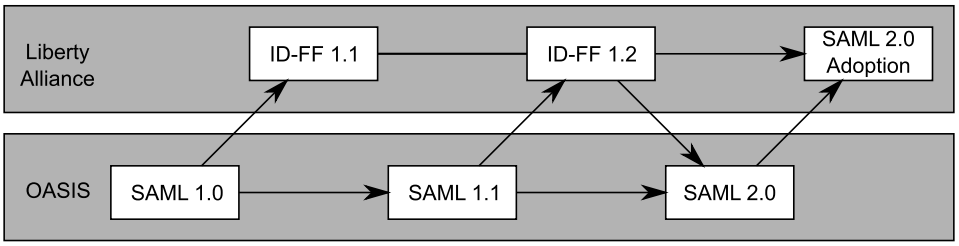


Figure 3.9: History of the Security Assertion Markup Language

- Usage Numbers: Depending on the application area and depending on the properties of the IdM system, the corresponding success and thus the usage numbers of the different IdM systems differ [HRZ10]. To the best of our knowledge no comparative study on the usage numbers of all IdM systems exist. Indicators for usage of IdM systems are login possibilities on web pages (e.g. OpenId [Bui11]) and products supporting the IdM systems (e.g. [LAI09]).

3.3 Base Technologies for Identity Management

Beyond the technologies introduced in Section 2.4, a couple of base technologies for IdM systems emerged. These are the Security Assertion Markup Language (SAML) (→ Section 3.3.1) and Web Services (WS) (→ Section 3.3.2).

3.3.1 Security Assertion Markup Language

SAML is an Organization for the Advancement of Structured Information Standards (OASIS) standard [C+05a] for the exchange of XML-encoded authentication, authorization and identity attribute statements. OASIS initiated SAML in 2001, because no standardized XML-based exchange format for security assertions was available. Figure 3.9 illustrates the SAML history, which is tightly coupled with the evolution of the Liberty Alliance (LA) Identity Federation Framework (ID-FF). LA is an industry consortium that started around the same time as SAML. LA and SAML standardization had a lot of mutual influences and resulted in version 2.0 of the SAML standard. The ITU-T adopted SAML as recommendation X.1141.

SAML 2.0 specifies XML schemas [HRF+07] for authentication, authorization and attribute assertions, and a set of simple protocols. The protocols define the exchange of assertions, SSO, SLO, name resolution and the establishment of federations. SAML has a modular structure that defines a core part, protocol bindings and extensions (→ Figure 3.10). The core part comprises the assertions and protocols. The bindings define how protocols like SOAP [W3C07] or HTTP transport the XML fragments containing assertions and protocol messages. Profiles specify coherent parts of assertions and protocols to realize specific use cases.

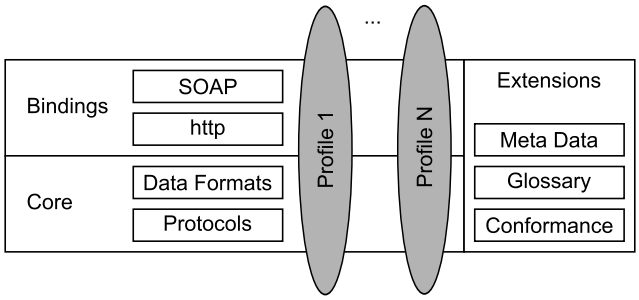


Figure 3.10: Structure of the SAML Protocol Suite

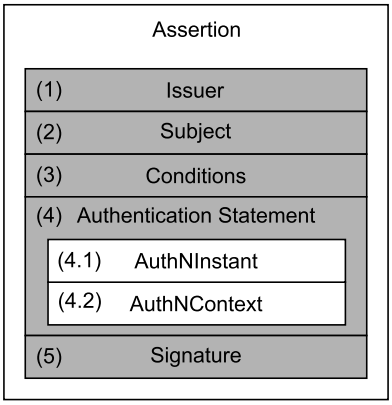


Figure 3.11: Structure of a SAML Authentication Assertion

Figure 3.11 shows the structure of a SAML Authentication Assertion[6]. The assertion container contains the following fragments: (1) Issuer: Contains details about the issuer of the assertion, which is typically the IdP. (2) Subject: Describes for whom the assertion is certified, i.e. one of the user's identities. (3) Conditions: Restrict the validity of the assertion regarding time, audience (e.g. only SP_x and reusability (e.g. only once)). (4) Authentication Statement: Gives the actual meaning to the assertion. [C$^+$05a] differentiates three different kinds of statements: authentication statements, authorization statement and attribute statements. (4.1) Authentication Instant: Specifies the point of time of the identity's authentication against the IdP. (4.2) Authentication Context: Gives details on the performed authentication procedure ($\rightarrow$ Section 2.3.3). (5) Signature: Signs the complete assertion. With the signature the issuing IdP enables integrity and authenticity checks of the assertion. For the signature XML Signature is applied [W3C08].

3.3.2 Web Service - Federation

A web service (WS) is a service that is identified with a Uniform Resource Identifier (URI) [RFC1630] and whose interface is described by means of XML. WSs are an enabler technology for the realization of the service oriented architecture (SOA) design principle. A set of web services has been specified for improved interoperability between different providers. Unfortunately, there is no unique standardization authority for web services (WS). Various standardization authorities (e.g. OASIS) or industry consortia have created the so called WS-*

[6]Figure 3.11 does not distinguish between required and optional parts. For a detailed description it is referred to [C$^+$05a].

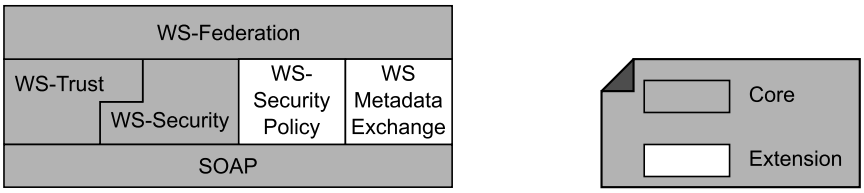


Figure 3.12: Structure of WS-Federation Protocol Suite

specifications. WS-Federation defines data formats and protocols for the in Section 3.2 introduced roles and workflows. According to Figure 3.12, WS-Federation relies on a couple of other standards and only coordinates the interworking of those. WS-Security [WS-02] specifies how the Simple Object Access Protocol (SOAP) [W3C07] messages are integrity protected and encrypted. Moreover WS-Security defines the encapsulation of security tokens, e.g. SP assertions. Based on WS-Security, WS-Trust [WS-09] specifies the exchange of tokens between IdPs, SPs, and users. Among the supported token formats are SAML and Kerberos. Finally, WS-Federation [WSF] puts WS-Security and WS-Trust together to realize for example SSO.

3.4 Existing Identity Management Systems

This section presents a selection of existing IdM systems. The selection is based on the prevalence of the IdM system and based on the existence of mentionable technical mechanisms. For each IdM system, technological aspects as well as the application area are introduced.

3.4.1 Shibboleth

Shibboleth [S$^+$05] is a successful IdM system that is used in academia. It bases on SAML version 2.0 and works with web browsers, i.e. it does not require the installation of software on clients. Shibboleth achieves this by means of HTTP redirections between the web pages of the SP and the IdP. Shibboleth allows that one SP federates with several IdPs. This requires an additional service that was not part of the reference architecture. The Where Are You From (WAYF) service allows the user to select the appropriate IdP.

[Shi11a] provides an overview of existing federations in academia that mostly rely on Shibboleth. Academic institutions make use of Shibboleth due to several reasons: (1) Heterogeneous services like access to library services (e.g. electronic journals) and network services are feasible. (2) No client installation is required and thus a heterogeneity of platforms can be supported with a simple web browser.

3.4.2 Liberty Alliance

Liberty Alliance (LA) [T$^+$] is a consortium of more than 30 companies that targeted on the harmonization of existing IdM systems. LA merged with other IdM projects into the Kantara initiative [Kan]. As Figure 3.9 shows, LA contributed to the evolution of SAML and is therefore in its core equal to SAML. Moreover, LA specifies additional techniques to enhance IdM

[LASb]. Many products targeting the application area of companies implement the LA specifications. For an overview of products and the implemented specifications, it is referred to [LAI09].

3.4.3 Microsoft CardSpace

Even if the development of Microsoft Cardspace [B+08] stopped in 2011 [Car11][7], it represents a valuable technology due to several reasons. First, Cardspace had a strong focus on usability. Identity cards, that reflect the identities of users and the associated attributes, create an analogy to business cards that is easy to understand from the user's perspective. A set of several cards creates a so called wallet, i.e. the second analogy, which is stored on the user's device. Second, the user selects an appropriate identity card for the authentication against IdPs or SPs and thus implicitly determines its appearance against the SP. This represents an easy to use privacy feature.

From a technology point of view, Cardspace bases on the WS-Federation stack (→ Section 3.3.2). For the storage and the selection of identity cards, the user has to install dedicated software[8]. Reasons for the termination of the Cardspace development by Microsoft are the limited usage (e.g. the number of web sites that supported Cardspace was always limited), the complex software development for IdPs and SPs, and the skepticism against Microsoft after the introduction of the centralized predecessor Passport[9] [Pas].

3.4.4 OpenId

OpenId [R+07, Reh07] is the most successful IdM system that works on top of the Internet with respect to the number of supporting web sites. [Bui11]. The OpenId design had the goal to provide a lightweight alternative to SAML-based systems. It has the following key characteristics:

(1) URI-based user identifier: A user identifier is either a Uniform Resource Identifier (URI) (e.g. http://barisch.com/marc) or an eXtensible Resource Identifier (XRI) [OAS08][10]. This identifier is used to retrieve information (XRDS document) that allows the identification of the corresponding OpenId provider. Thus the IdP discovery problem is solved without the need of additional services or mechanisms (c.f. WAYF service of Shibboleth).

(2) Decentralization: Relying parties and OpenId providers do not have to be part of a federation a priori. Both establish a trust relationship on demand. This is reasonable, since the SP only federates with the IdP on an identity-basis (→ Section 3.2.2.6).

Figure 3.13 shows the typical message exchange to authenticate users and enable SSO. If the user wants access to protected resources, the SP requires the user to enter an OpenId identifier.

[7] Development continues with an alternative technology called UProve [UPr].

[8] With the operating systems Microsoft Vista and Microsoft 7, Microsoft Cardspace is already installed. For other operating systems, open source implementations are available, e.g. DigitalMe [Dig], Higgins [Hig].

[9] Microsoft Passport is still in operation, but strongly coupled to Microsoft services.

[10] XRI has been developed as a generalization of uniform resource identifiers (URI).

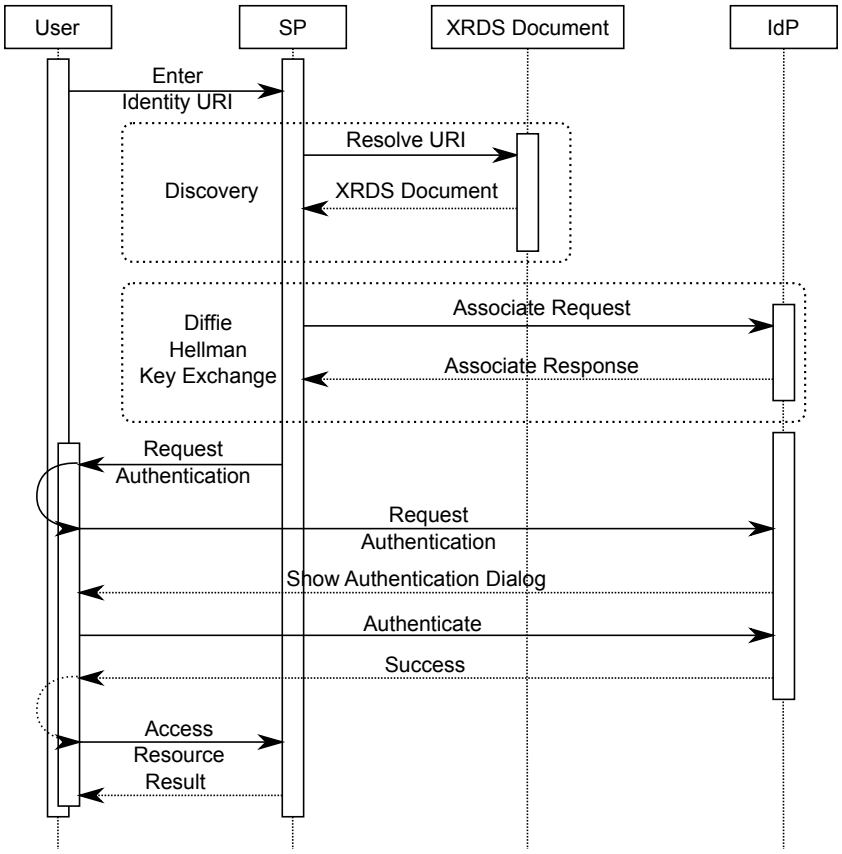


Figure 3.13: OpenId Message Sequence Chart

The SP uses the OpenId identifier to discover information about the user, i.e. the corresponding identity provider. This is achieved by retrieving an XRDS document that contains information about supported services, among them OpenId authentication. The SP contacts the OpenId provider and establishes a security association with it based on Diffie Hellman key exchange. Afterwards the SP redirects the user's browser to the IdP. The IdP is responsible for the user authentication. OpenId excludes the specification of supported authentication methods. In practice, the OpenId provider authenticates the user by means of username/password combinations. After successful authentication, the OpenId provider redirects the user back to the SP with a corresponding security assertion. The SP evaluates the authenticity and integrity of the security assertion and eventually grants access to the protected resource.

3.4.5 Kerberos

Kerberos [RFC4120] is a SSO authentication service for network services (e.g. HTTP, SSH). Initially developed at Massachusetts Institute of Technology (MIT) in the 80s [SNS88], it evolved to a technology that is available for many operating systems (e.g. Windows or Linux) for user authentication.

Kerberos combines the local authentication against the user's device with the authentication against the Authentication Service (→ Figure 3.14). The authentication service is part of the Key Distribution Center (KDC) that maintains trust relationships with the services. Based on

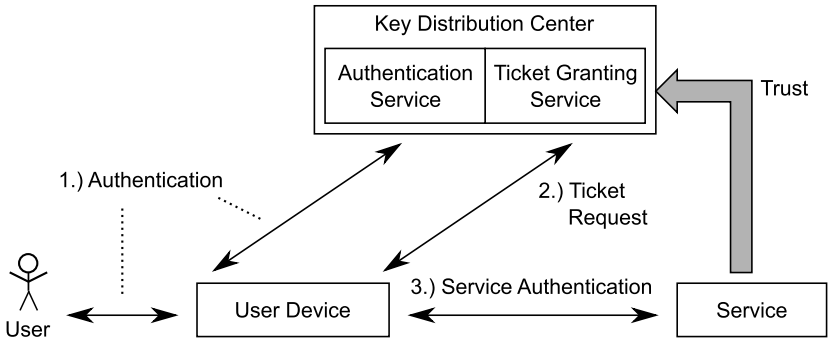


Figure 3.14: Kerberos System Structure

the authentication against the KDC, the user obtains SP assertions (called tickets with Kerberos) for the authentication against services.

Kerberos represents a sound technology for authentication within companies if the following drawbacks can be neglected. (1) Password-based: The Kerberos authentication only supports password-based authentication[11]. (2) No support for multiple identities: Most Kerberos implementations do not allow that the user authenticates with several identities at the same time, because of the coupling of local (device) authentication and authentication against the KDC. [Pal11] proposes extensions for multi-identity authentication. (3) Client Support: Kerberos requires clients to run the Kerberos client software. Applications that use Kerberos have to realize interfaces with the Kerberos client. For example, access to a Kerberos protected web page requires that the web browser supports Kerberos [Prö11] and has implemented an interface to the Kerberos client. (4) No Access Management: Kerberos specifies a basic mechanism for providing authorization information to services. The implementation of this feature is proprietary. In most cases an additional LDAP directory has to be inquired regarding authorization.

Even if Kerberos is applied in most cases in single domain scenarios, it supports multi-domain scenarios. A multi-domain scenario means that several KDCs exist and that corresponding trust relations exist between the KDCs.

3.5 Security of Identity Management Systems

The security of IdM systems is of uttermost importance for the following reasons: (1) Identity Aggregation: An IdM system enables a user to aggregate x identities with several SPs into y identities with different IdPs. An IdM system is only useful, if it aggregates identities, i.e. $y < x$. In consequence an IdP becomes a favored target for attacks and the identity with the IdP requires more secure authentication mechanisms, e.g. the password with the IdP has to adhere to stricter password policies [FH10]. (2) Additional Mechanisms and Protocols: IdM systems introduce additional protocols to enable SSO, SLO, et cetera. Additional protocols result in a larger attack surface and thus it has to be ensured that no additional vulnerabilities are created. That means security of IdM systems can only be guaranteed, if the following three

[11] [RFC4120] specifies an optional hardware supported authentication method. Open implementations do not support this option.

aspects are fulfilled: Security of the IdM protocols, security of the IdM system implementation and usability.

Security of Identity Management Protocols: The applied IdM protocols have to be secure regarding their design. Several security evaluations for SAML, OpenId and MS Cardspace have been conducted. In the early stages of the SAML and LA evolution several vulnerabilities have been identified. [PW03] identified a possible MitM attack in the protocol design and proposed countermeasures that have been taken up by LA. In contrast, [Gro03] pointed out deficiencies in the protocol specification of SAML that might lead to vulnerable implementations. A recent analysis of SAML 2.0 [ACC+08] has not identified vulnerabilities in SAML itself, but in a corresponding implementation. [H+05] provides a detailed security assessment of SAML.

OpenId is a weak IdM system. [Ope] provides an overview on the identified vulnerabilities. [TT07] identified vulnerabilities in OpenId 1.0. Among the vulnerabilities are MitM attacks on the Diffie-Hellmann key exchange (→ Figure 3.13) and phishing attacks. [BJM08] describes a cross-site request forgery attack on OpenId 2.0 that is caused by a lack in the specification. In addition, [SKS10] showed that identity information can be manipulated. Also for MS Cardspace design vulnerabilities have been identified [GSSX09].

Security of Identity Management System Implementations: Vulnerabilities in IdM system implementations can have a severe impact. Therefore, it is mandatory that IdM implementations do not contain vulnerabilities caused by improper implementation of the IdM protocols. A suitable mechanism for vulnerability detection is static code analysis (→ Section 2.5.2).

Security and Usability: The user represents the weakest chain link and consideration of the security of an IdM system cannot neglect usability [JZS07]. For example, monetary incentives can seduce users to ignore security mechanisms and recommendations [CEVG11]. IdM systems improve the usability by reducing the number of identities. Fewer identities justify to improve the authentication procedure with respect to the security level, e.g. by having stricter password policies to overcome improper passwords [FH07]. [DD08] gives an overview on additional flaws of IdM systems regarding usability.

3.6 Identity Management and Multiple Devices

Extending IdM to several devices is the core of this thesis. Existing IdM systems do not sufficiently address this issue. Several solutions exist that provide the possibility to users to consume services from different devices. We can classify existing solutions according to the distribution of credentials into three categories:

- Personal Authentication Devices: All credentials remain on one dedicated device that is used to perform the authentication (→ Section 3.6.1).
- Distribution of Credentials: All devices obtain all credentials (→ Section 3.6.2).
- On-demand Provisioning of Credentials: Devices can request credentials from another device upon demand (→ Section 3.6.3).

3.6.1 Personal Authentication Devices

Several solutions introduce dedicated authentication devices [W+85, PPSW97]. Wong et.al. [W+85] have been the first that introduced the concept of Personal Authentication Devices, which are used to authenticate against services independent of the actually used device. Even if they take multiple user-devices into account, no solution regarding SSO and federation is provided. Moreover, they do not provide sufficient usability, because the user has to manually enter a PIN. [J+05] discusses consequences of the resulting trust model. [Cor] defines an extension of the Personal Authentication Device based on Microsoft Cardspace. The developed solution stores all identity cards of a user on a mobile device and makes these cards available to other devices.

3.6.2 Distribution of Credentials

Instead of having one central authentication device, it is possible to distribute all credentials to all devices owned by the same user. Such a solution is typically applied to passwords. Table 3.1 shows a selected overview of existing solutions. All existing solutions have in common that they store passwords in an encrypted keystore, which is put to a central server. The central server allows the retrieval of the keystore by all devices of a user. Decryption of the keystore is only possible with an adequate key. Table 3.1 identifies for each solution the applied encryption algorithm and the key length used with the encryption algorithm. The storage location determines, whether it is possible to store the passwords on a server that the user operates or on servers provided by the solution provider. All identified mechanisms allow a per password synchronization, which enables a flexible password usage across all devices. If the source code is available, it is possible to conduct further security evaluation. The platform dependence informs about the flexibility of the solution regarding different devices (e.g. desktops or mobiles).

Password synchronization has disadvantages. If an attacker gets access to the encrypted keystore, an offline brute-force attack is possible to break the encryption. Moreover it is not possible to store credentials different than passwords. [Hüb08] proposed an extended version for a distributed key store, which allows the storage of any kind of credentials, i.e. no limitation on passwords exists.

3.6.3 On-demand Provisioning of Credentials

In contrast to personal authentication devices and the distribution of credentials to all devices belonging to a user, on-demand provisioning solutions provide devices with the necessary credentials on demand. That means one device can request the required credentials or assertions from another device a soon as they are needed for authentication against a SP.

With session mobility in mind, LA proposed a solution to transfer credentials on demand between devices [Mad08]. Combined with the transfer of the application context, which is required to enable session mobility, so called endpoint references can be transferred. Endpoint references allow the creation of an additional SAML assertion for service authentication. This

solution assumes a preestablished trust relationship between the participating devices, but specifies no additional details. To the best of our knowledge, further consideration of these initial concepts has stopped. Recently, [RG10] proposed an extension to OpenAuth. The proposal addresses authentication on devices with limited input devices (e.g. no keyboard to enter password). Hereby, the user authenticates on a more powerful device and authorization tokens are transferred to the restricted device. Trust between the devices is manually established on demand, i.e. an identifier is displayed on the limited device and has to be manually entered on the more powerful device.

Table 3.1: Comparison of Password Synchronization Mechanisms

Mechanism	Encryption	Encryption Algorithm	Key Length	Storage Location	Synchronization	Licence	Source Code Availabe	Plattform Dependence
SyncPlaces [Syn]	X	TEA, AES [12]	-	User defined [13]	Yes	LGPL	Yes [14]	No [15]
LastPass [Lasa]	X	AES	256 Bit	Last Pass Server	Yes	Prop.	Yes [14]	Yes
Opera Link [Vel11]	X[16]	AES	128 Bit	Opera Server	Yes	Prop.	Partially	No [17]
Firefox Sync [Fir]	X	AES	256 Bit	Firefox Server	Yes	MPL/ GPL/ LGPL	Yes	No [18]

The solution proposed and evaluated in the subsequent chapters falls into this category. In contrast to [RG10], it considers preestablished trust relationships between devices. Moreover it considers different identities as well as security levels of devices to authorize the usage of identities across user devices.

[12] TEA is used as default. Should be changed to AES. For AES no computationally feasible attacks are known, in contrast to TEA.

[13] Support for WebDav[RFC4918] and FTP[RFC959]

[14] Firefox Addon → Javascript code available

[15] LastPass supports many different operating systems.

[16] Protected with same password as Opera account, i.e. Opera is in the position to reveal passwords. The password is only used for the encryption of a strong encrpytion key[Vel11], i.e. brute force attacks are difficult.

[17] Opera supports many different operating systems.

[18] Firefox is running on different platforms.

4 Architecture Design for Multi-device Identity Management

This chapter describes the architecture and the design process to realize a system, which enables multi-device identity management. Figure 4.1 illustrates the steps of this process that are detailed in the corresponding subsections.

The usage scenarios in Section 4.1 illustrate and motivate the need for multi-device IdM. They shape the foundation for all other steps by formulation of challenges and high-level requirements that have to be addressed in order to realize the system. Based on the challenges and high-level requirements, the three key concepts – Virtual Device, IdP and SP Session Split and Multi-device IdM – have been defined and serve as overarching guidelines for the further derivation of requirements and for the design of the functional architecture. Section 4.3 elaborates functional and non-functional requirements based on a requirements engineering methodology and describes selected requirements in detail. Security has to be considered from the beginning of system design. Therefore, an initial security analysis is performed in Section 4.4. The security analysis identifies assets and threats based on the usage scenarios, the key concepts and the requirements. The identified threats are countered by corresponding security requirements. Based on the previous steps, the functional architecture is designed in Section 4.5. It comprises functional blocks and defines interfaces between those.

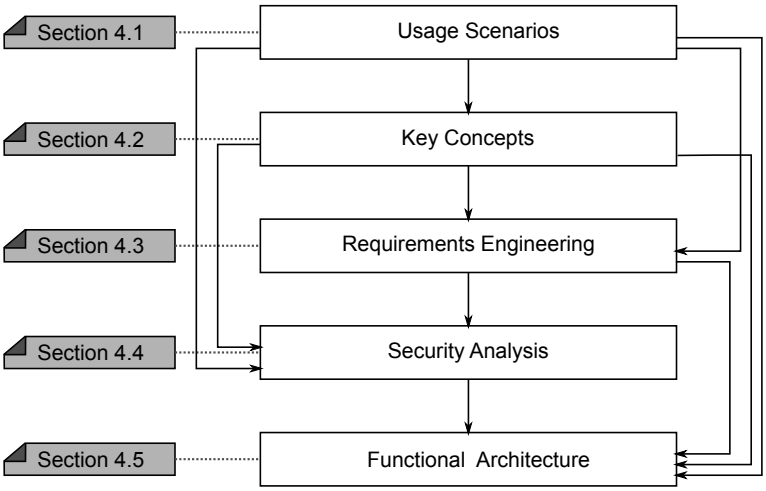


Figure 4.1: Chapter Outline

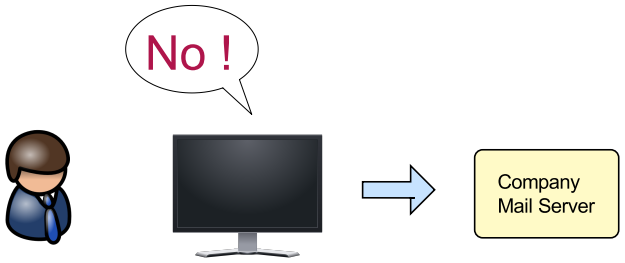


Figure 4.2: Usage Scenario 1 – Private and Business Usage of Devices

4.1 Usage Scenarios

This section introduces five usage scenarios to elaborate the vision of the planned system and to illustrate the potential benefits. Each usage scenario describes a particular situation of one user that wants to access services that require authentication. It is assumed that a user has several devices that are owned by him. Examples for such devices are notebooks, smartphones, or even TV sets. Currently, these devices are independent of each other, i.e. they do not collaborate. The usage scenarios highlight challenges that result from the independence of devices with respect to authentication and IdM. These challenges are elaborated by subsequent sections and finally addressed by the designed architecture that enables cooperation of devices with respect to IdM. Each usage scenario describes a situation of users and highlights a challenge regarding the interworking of devices to improve IdM. After the introduction of the usage scenarios, Section 4.1.6 provides a summary of the identified challenges.

- Scenario 1: Business and Private Devices (c.f. Section 4.1.1)
- Scenario 2: Fast “Device Change” (c.f. Section 4.1.2)
- Scenario 3: Identity Usage on Insecure Devices (c.f. Section 4.1.3)
- Scenario 4: Insufficient Security Features (c.f. Section 4.1.4)
- Scenario 5: Insufficient Input Methods (c.f. Section 4.1.5)

4.1.1 Scenario 1: Business and Private Devices

Many employees use notebooks, smartphones and other devices that are provided by their employer. In addition, every employee may have his own private devices. In many jobs the border between private life and business activities is blurred, i.e. one can work at home or use time on business trips for private purposes. A consequence of such nomadic behavior is that there is often no differentiation between private use and business use of communication devices. Figure 4.2 illustrates a scenario in which the user checks business mails on a private device via web interfaces. Vice versa, a user might use private Facebook accounts on the business smartphone. Hereby, every usage context has different security requirements [JGtM00], which has to be considered resulting in potential restrictions.

Challenge: The usage context of identities and devices has to be considered. If the usage context does not fit, identity usage must be restricted.

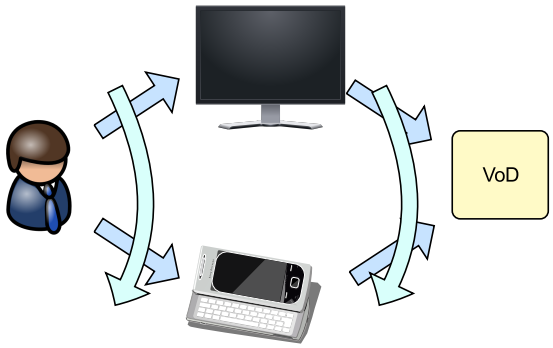


Figure 4.3: Usage Scenario 2 – Fast Device Change

4.1.2 Scenario 2: Fast "Device Change"

Mobility is becoming commodity. Various activities are performed at different locations with different devices. One visionary scenario, which is often used in research [L+08, J+08], is the change of devices due to mobility as illustrated in Figure 4.3. At home, the user begins watching a movie provided by a video on demand (VoD) provider on the 50" TV screen. That means the user has to authenticate against the VoD provider on the TV screen. During the movie, the user leaves his home and wants to continue watching on the smart phone. Today, the user has to reauthenticate, select the movie again and trigger a fast forward to the position where the movie was stopped.

Challenge: It shall be possible to continue an existing service session on a second device without the need for reauthentication against the SP on the second device.

4.1.3 Scenario 3: Identity Usage on Insecure Devices

Often users have computers or other communication devices at home that are not that good maintained from a security perspective. Either necessary security patches are not applied leading to vulnerabilities or malicious programs are installed. Therefore, using such kind of devices might have serious consequences. In particular, authenticating on such devices might lead to intercepted credentials (e.g. username/password combinations), resulting in identity theft and impersonification. Figure 4.4 illustrates an example scenario. A user wants to read his emails provided by a webmail provider on an insecure machine. Since the machine has generous hardware (large display, ...) it is attractive for the user, even if the machine does not provide adequate security. The smartphone, which is assumed to be more secure, is not used at all.

Challenge: For authentication the most secure device shall be used. Only short-time credentials shall be made available to insecure devices.

4.1.4 Scenario 4: Insufficient Security Features

Some services need more trust into the user identity than others. This can be achieved by the usage of dedicated security equipment (e.g. card readers or one-time password generators).

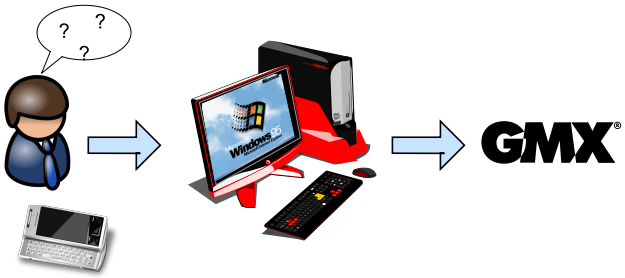


Figure 4.4: Usage Scenario 3 – Identity Usage on Insecure Devices

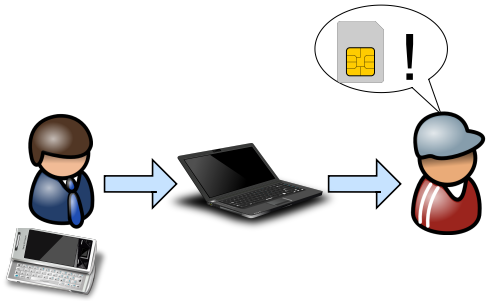


Figure 4.5: Usage Scenario 4 – Insufficient Security Features on Device

Figure 4.5 shows a scenario, in which the SP requests that authentication should be based on a SIM card, since the operator of the mobile network is trustworthy and has verified the identity of its customers by out-of-band means (e.g. verification of passport). Since the notebook of the user has no means for SIM card based authentication, the user cannot make use of the service.

Challenge: It shall be possible to share the authentication capabilities across all devices of a user.

4.1.5 Scenario 5: Insufficient Input Methods

More and more devices get network access without sophisticated input capabilities, like keyboards. A user should be able to authenticate on a device with appropriate input capabilities. Figure 4.6 illustrates a scenario, in which a user wants to access his private images on a game console. The game console has limited input capabilities and thus it should be possible to use the notebook for authentication [RG10] and the game console with the TV screen for the presentation of the images.

Challenge: It shall be possible to relay the authentication to a more powerful and trusted device in case of limited input capabilities.

4.1.6 Summary of Challenges

The previous sections outlined five different visions that are summarized in the following. The challenges resulting from Scenario 4 and 5 have been unified.

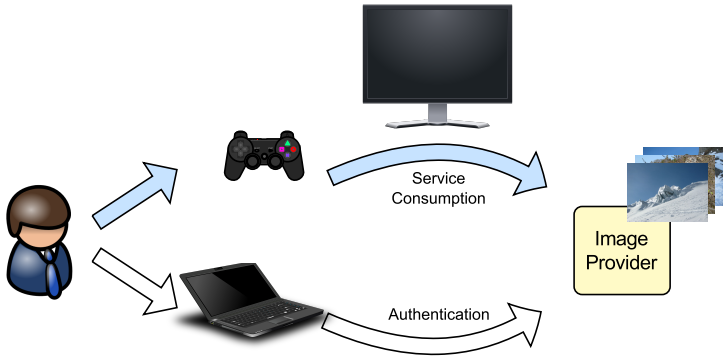


Figure 4.6: Usage Scenario 5 – Insufficient Input Capabilities

Table 4.1: Summary of Challenges

Restriction of Identity Usage	The usage context of identities and devices should be considered and potentially restricted.
Seamless Device Change	It should be possible to continue an existing service session on another device without the need for reauthentication.
Secure Authentication	For authentication, i.e. the usage of credentials, the most secure device should be used.
Sharing of Authentication Capabilities	It must be possible to make use of the authentication capabilities of another device by relaying the authentication process.

These challenges exemplify the goal behind the multi-device IdM concept: Improve the security and usability for users that have several devices and make use of IdM systems.

4.1.7 High-Level Requirements

The scenarios and identified challenges describe particular characteristics that a multi-device IdM system has to provide. These characteristics are defined in the following as so called high-level requirements that have to be addressed in the following by adequate concepts.

R1 - Secure exchange: The devices of the user exchange information, e.g. authentication assertions for SPs. The information exchange has to be secured.

R2 - Session distribution: Different devices of a user are responsible for different tasks. It must be possible that authentication against an IdP takes place on one device, whereas the service is consumed on another device.

R3 - Remote Activation: If one device cannot fulfill the requirements, e.g. by a SP regarding authentication, it must be possible to trigger another device to perform the authentication.

R4 - Discovery of user devices: It is required to discover all devices belonging to the same user. Only devices in the proximity of the user can be used for authentication against IdPs.

R5 - Capture of device characteristics: The properties of user devices have to be captured and exchanged among each other. Supported authentication methods are of interest in particular. In addition it is required to capture relevant device properties (e.g. operating system, installed software) to determine the security level of a device. The security level of a device is a metric that allows the quantification of security.

R6 - Establishment of security associations: It is necessary that devices authenticate each other and establish a confidentiality and integrity protected channel.

R7 - Determination of usage context: The usage context of devices as well as of identities has to be declared by the user.

R8 - Distributed data handling: Every device captures and stores information regarding the device itself as well as the user and his identities. Since this data is required for decision making, it has to be exchanged between the devices.

This thesis does not claim that the high-level requirements are complete and disjoint. The high-level requirements are derived from the usage scenarios in a intuitive and logical way. The same holds for the stepwise refinement of these high-level requirements towards functional and non-functional requirements.

4.2 Key Concepts

A key concept groups coherent aspects and requirements of the system under design. Hereby, a key concept provides overarching guidelines for the elicitation of requirements and separation of concerns. The defined functional architecture reflects a key concept by appropriate technical mechanisms. The usage scenarios, the challenges and the high-level requirements defined in Section 4.1 resulted in the definition of three key concepts:

- Virtual Device Concept
- Session Split Concept
- Multi-Device IdM Concept

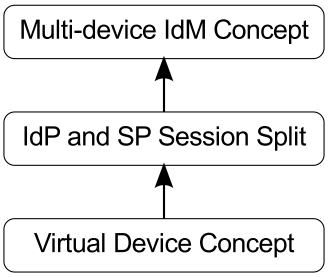


Figure 4.7: Hierarchy of Key Concepts

Figure 4.7 illustrates the dependencies of the key concepts. The Virtual Device Concept is the basic concept that provides a unified view on user's devices. The virtual device provides the opportunity to split user and service session across different devices. This in turn represents the enabler for IdM across multiple devices, i.e. the Multi-Device IdM Concept.

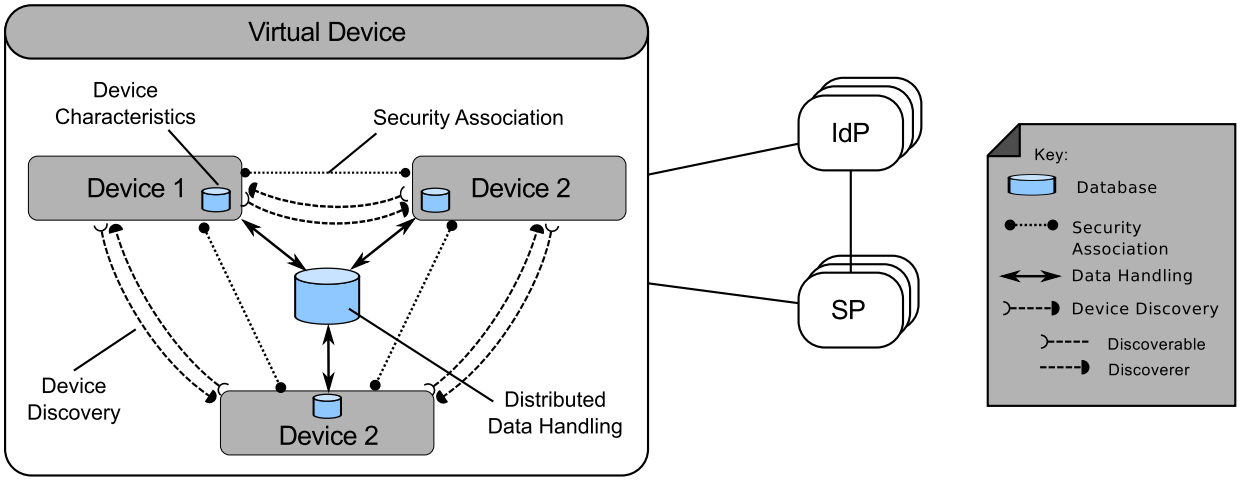


Figure 4.8: Conceptual View on Virtual Device Concept

4.2.1 Virtual Device Concept

The virtual device concept is the basic key concept that allows the realization of all usage scenarios. A Virtual Device (VD) provides an umbrella for all devices belonging to one user as illustrated in Figure 4.8. Devices that have been previously independent of each other are now integrated and provide functionality in cooperation. This concept renders it possible that several user devices appear to third parties as one device. With a VD, a user is in the position to benefit from the resources and characteristics that are individually provided by each device, i.e. the disadvantages of one device might be compensated by the advantages of another device.

In order to appear as one device, the VD inherently provides a couple of mechanisms that are illustrated in Figure 4.8:

- Device Discovery: Devices belonging to the same user and to the same VD are in the position to discover each other. Device discovery is the prerequisite for any other exchange of data and control between the devices.

- Security Association: Belonging to a virtual device requires a mechanism for mutual authentication between devices. This is achieved by a preestablished security association between all devices belonging to a virtual device. A security association is the prerequisite for the establishment of secure channels between devices.

- Device Characteristics: Since the VD is the basis for other mechanisms, it provides the functionality to capture device characteristics. Device characteristics enable the optimization of service delivery according to the capabilities of individual devices. This represents a prerequisite for the session split concept.

- Distributed Data Handling: A VD includes several devices owned by one user. In order to coordinate the devices and to benefit from individual capabilities, it is required to exchange data among the devices of a VD. Therefore, a mechanism is required that allows the exchange and the distributed handling of data.

Addressed High-Level Requirements: The Virtual Device Concept addresses the following high-level requirements of Section 4.1.7:

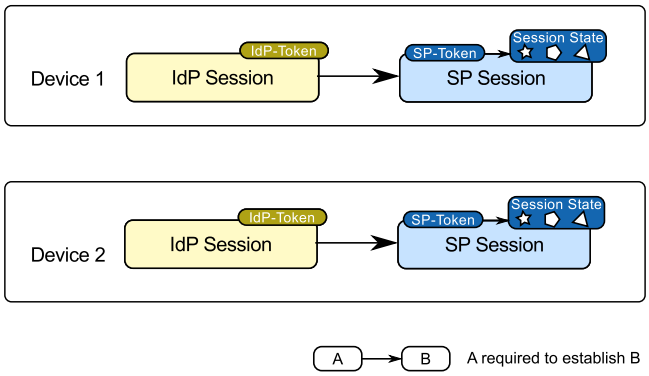
- R1 and R6: Each two devices of a VD have a trust relationship and can establish a secure channel for the exchange of data.
- R4: Devices belonging to a VD can discover each other.
- R5: Devices belonging to a VD are described by device properties. Therefore, a mechanism is provided to capture the relevant properties.
- R8: Devices belonging to a VD can exchange information among each other.

4.2.2 IdP and SP Session Split Concept

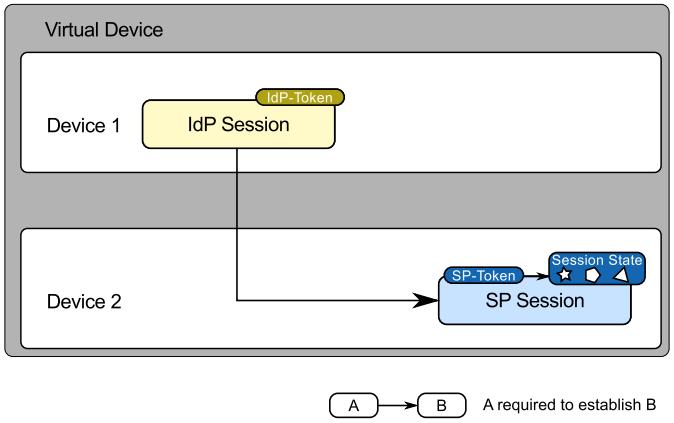
To realize the vision of secure authentication and sharing of authentication capabilities it is required that the device for authentication can be a different device than the one used for service consumption. Compared to traditional scenarios where the IdP session resides on the same device as the SP session (c.f. Figure 4.9(a)), the Session Split Concept splits the sessions. Splitting means the IdP session, which is established by authenticating against the IdP, resides on another device than the SP session (c.f. Figure 4.9(b)).

Based on the VD concept, devices trust each other up to a certain level, which is determined by the device characteristics. Therefore, it is basically possible to distribute IdP and SP sessions across devices. This enables a new degree of freedom, because additional selection criteria for the distribution of sessions can be considered. In line with the vision, the following criteria for the establishment of the IdP session are considered:

- Security level of device: Each device has a security level, which is a metric that allows the quantification of the security of the device. This metric can be used for example to select the most secure device for authentication.
- Authentication capabilities: Each device supports only a limited set of authentication methods. The supported authentication methods are restricted by the available hardware as well as by the installed software. The most prominent example for an authentication method that is only supported by a subset of devices is SIM card based authentication. Other examples are support for biometric authentication. The authentication capabilities describe the authentication methods supported by a device.
- Input devices: Each device has only limited input capabilities. Entering a password without a keyboard is tedious. Thus, a device with appropriate input capabilities can be selected.
- Usage context: The usage context designates the context in which the IdP session is used. We can differentiate on a high level between private and business context. Both usage contexts can be refined.



(a) Session distribution in traditional scenario



(b) Session distribution with Session Split Concept

Figure 4.9: Comparison of Session Distribution Approaches

In contrast to the IdP session, the SP session may be established on devices that are more powerful or dedicated regarding resources (e.g. display size, computing power).

Addressed High-Level Requirements: The IdP and SP Session Split Concept addresses the following high-level requirements of Section 4.1.7:

- R2: The IdP and SP Session Split concept allows the distribution of task across several devices owned by the same user.

4.2.3 Multi-device IdM Concept

The Multi-device IdM concept relies on the Virtual Device concept and the Session Split concept. It comprises the secure exchange of assertions to establish SP sessions, the remote activation of identities on other user devices, the prefiltering of useable identities as well as the acquisition of information about available and active identities (→ Figure 4.10).

Assertion Exchange: A device without an IdP session needs to transfer a SP assertion from another device to establish a SP session. This is realized by the Assertion Exchange protocol (AEP) (see Section 4.5.3) that allows requesting and obtaining assertions for a particular identity. The providing device checks the request against the configured policies that limit the usage

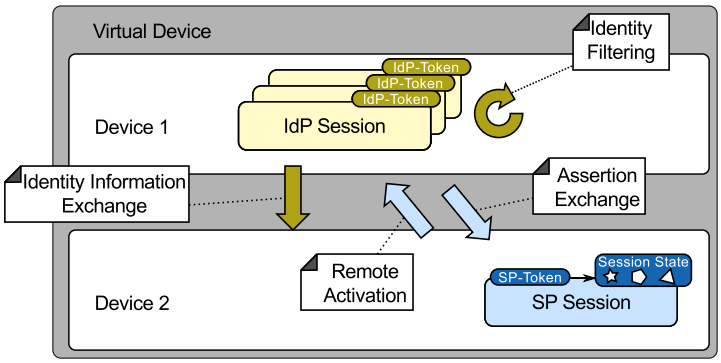


Figure 4.10: Multi-Device IdM Concept

of identities across devices. As an optional security mechanism, the user has to confirm the request on the providing device.

Remote Activation: If an IdP session for a dedicated identity cannot be established on one device, it has to be possible to activate this identity on another device. This is realized by the Identity Activation Protocol (IAP) (see Section 4.5.3). Activation of identities on other devices requires that the requested device is in the proximity of the user, because the user is involved in the authentication procedure, and that the request is authorized.

Filtering of Identities: It is assumed that not each identity can be used on every device. First, specific credentials or specific authentication methods, which are not available on all devices, might be required to activate an identity. Second, the usage context of an identity might not be appropriate for a device (see Section 4.1.1). Third, an identity should not be activatable on some devices due to security constraints, e.g. the security level is too low to use a dedicated identity.

Active Identities: The information on activated identities, i.e. an IdP session exists, should be available on other devices. With this information the number of active IdP sessions can be reduced and the burden for users to reauthenticate on another device is decreased. The Identity Information Exchange Protocol (IIEP) is applied for the exchange of information related to identities.

Addressed High-Level Requirements: The Multi-device IdM Concept addresses the following high-level-requirements (c.f. Section 4.1.7):

- R2: The Assertion Exchange and the Remote Activation possibilities allow distribution of sessions across devices.

- R3: With the Remote Activation, it is possible to perform the authentication on a remote device.

- R7: The Filtering of Identities considers the usage context of devices and identities.

4.3 Requirements Engineering

The usage scenarios with the corresponding challenges and high-level requirements (→ Section 4.1), and the key concepts (→ Section 4.2) serve as input for the requirements engineering process. Requirements engineering is the process to discover and maintain requirements [Som10]. The process includes activities to discover, analyze, document and maintain requirements [SWE10]. Requirements are defined as externally observable and explicitly desired characteristics of a system [Mar01]. The requirements identified in this phase detail and augment the high-level requirements and could be considered as low-level requirements. They are simply addressed as requirements in the following.

Since requirement engineering is a complex process [Par10, Som10, NE00] and not in the core focus of this thesis, it is restricted to the pure application as a method. Section 4.3.1 outlines the applied requirements engineering process. The methodology first identifies the stakeholders of the system in Section 4.3.2.1. Their interests and the in Section 4.1 introduced scenarios, serve as basis for the specification of functional and non-functional requirements in Section 4.3.3 and Section 4.3.4, respectively.

4.3.1 Methodology

The applied requirements engineering process consists of three steps:

1. Requirement Elicitation
2. Requirement Analysis
3. Requirement Specification

4.3.1.1 Requirement Elicitation

In literature, several approaches for the elicitation of requirements are known (e.g. interview-based, viewpoints [Som10]). The selected approach is scenario-based and augmented with the identification of stakeholders [GW07] and their view on the system. Section 4.1 already described the scenarios and Section 4.3.2.1 identifies the stakeholders.

4.3.1.2 Requirement Analysis

The requirement analysis phase categorizes requirements and identifies relations, dependencies and conflicts between different requirements. Basically, requirements can be divided into two categories [Par10, AM$^+$04] as depicted in Figure 4.11: Functional and Nonfunctional Requirements.

Functional requirements describe the required system functionality. This includes system input and system output. System input consists of data and events that lead to state changes. State changes are reflected by corresponding system output. Section 4.3.3 describes the functional

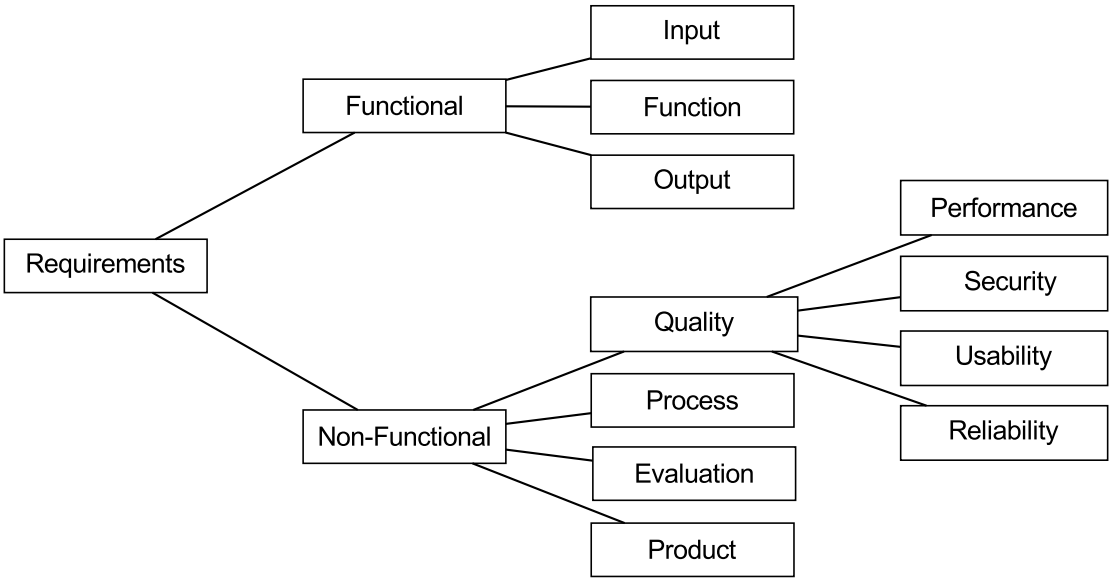


Figure 4.11: Overview on Requirement Types

requirements of the system and makes use of the already identified key concepts with their corresponding functionality.

In contrast, non-functional requirements specify additional constraints and conditions that the system, the development process or the product itself has to fulfill. Various taxonomies for the categorization of non-functional requirements exist [Boe08, Par10, NE00, Som10]. The taxonomy by Partsch [Par10], which is applied in the following, categorizes non-functional requirements into four categories:

- Quality Requirements: Quality requirements specify the circumstances under which the functionality has to be provided. Quality requirements differentiate between performance, security, usability and reliability requirements. Section 4.3.4 and Section 4.4[1] specify quality requirements.

- Development Process: Requirements on the development process itself are not considered. Development process requirements are of importance if several developers are involved and if the product has to comply with dedicated regulations that are established by the client. For example, in case of software for government authorities, the V-model [Bal09] might be prescribed as system development model. The V-model in turn requires the creation of detailed documentation of all development steps.

- Product Requirements: Requirements on the overall product (e.g. system documentation) are not considered. Product requirements restrict the design space regarding of the system. For example, the restriction that the system has to work on a dedicated operating system restricts the design space. Since such restrictions shall not be considered, product requirements will not be established.

[1] Since security is very important for the system, a specific methodology to gather security requirements has been applied. Section A.1.4 contains the specification of the security requirements.

- Evaluation Requirements: Requirement for the evaluation, which Section 6 details, are not considered. For example, an evaluation requirement might mandate the need to test every functional component. Since it is not intended to provide a complete prototype, it is not reasonable to establish evaluation requirements.

4.3.1.3 Requirement Documentation

Several possibilities exist for the documentation and specification of requirements. Each possibility has a different degree of formalism. Normal text provides almost no formalism, structured text or textual templates can be considered as a semi-formal approach and logical or graphical models provide the highest degree of formalism.

The Volere template [Mai06], which has been tailored to the needs of this thesis, is a semi-formal approach that documents requirements according to the criteria of good requirements [Par10]. After adaptation, the template contains the following parts:

- Requirement Identifier:	Required for identification of the requirement and cross references.
- Requirement Type:	Based on the classification in Figure 4.11, each requirement is assigned to one requirement type.
- Short Description:	Provides a human readable name of the requirement.
- Long Description:	Gives additional details on the requirements.
- Rationale:	Describes reasons and motivation for a requirement.
- Stakeholder:	Names the primary stakeholder of the requirement.
- Dependencies:	Provides references on dependant requirements.

4.3.2 Requirements Elicitation

As introduced above, the requirements elicitation focuses on a scenario-driven approach. This includes the definition of the relevant stakeholders and their relation among each other. Therefore, the following consideration excludes a couple of potential stakeholders and restricts itself to the primary stakeholders and their interaction.

4.3.2.1 Stakeholders

[GW07] defines a stakeholder as "a person or organization who influences a system's requirements or who is impacted by that system"[2]. According to this definition, all persons or organizations involved in the scenarios are stakeholders. Stakeholders can be categorized into two groups: Primary Stakeholders and Secondary Stakeholders. A primary stakeholder is directly involved in the usage and operation of the system. Secondary stakeholders are not involved in the usage or operation, but define additional rules regarding the usage and operation. The main focus of the following activities is on the primary stakeholders that are directly involved. These

[2]Stakeholder identification and analysis is not limited to system and software engineering. It originates from business management [Bry03].

are the user, the SP, and the IdP. The interests and responsibilities of the primary stakeholders have been already introduced in previous sections (see Chapter 3).

Stakeholders can also be classified in correspondence to the phases of the software lifecycle [PW01]. [Pow10] provides an extensive overview of potential stakeholders. In particular, many stakeholders are involved in the development phase of complex systems. Due to the nature of this thesis, it does not consider stakeholders in the development phase.

Additionally, the following stakeholders have been identified.

Legislator The legislator defines laws for the operation of ICT systems. Regarding IdM, data protection regulations are important and have to be considered if different parties collect information about each other. It is assumed that the IdP and the SP are compliant to legislation and that the devices belong to the same user. In consequence, no additional data is collected or exchanged. Therefore, there is no need to consider the legislator as a stakeholder.

Device Owner An inherent assumption of scenario 3 (c.f. Section 4.1.1) is that business devices are owned by the company. That means the owner of the device is not necessarily the user itself. Therefore, it must be considered that the device owner has an interest in influencing the operation of the device. The device owner is defined as secondary stakeholder.

4.3.2.2 Primary Stakeholder View

Figure 4.12 elaborates the relationships of the primary stakeholders, which are in focus. Central element is the federation. The federation is a virtual element[3] that puts SPs, IdPs and identities of users together (c.f. Section 3.2.2.6).

The user has one or more devices and one or more identities. An identity is only valid within a federation. To activate an identity it is necessary to authenticate against the IdP of this federation. For the authentication process, credentials are required that are associated to an identity. Each authentication method requires different kinds of credentials, e.g. a username/password combination or certificates.

A device is associated to a user, which does not mean that the user owns the device. For simplicity Figure 4.12 does not contain the device owner. A device has security properties that are used to determine a security level of the device. On the other hand a device supports different security mechanisms. Examples for security mechanisms are authentication algorithms and encryption algorithms.

Devices as well as identities are used in a certain context, i.e. a usage context. Identities and devices should support at least one usage context. Only if the usage contexts of devices and identities are not mutually exclusive, it is possible to use an identity on a device.

A SP offers services to users. Depending on the service, it is possible to request the use of specific security mechanisms. The request of security mechanisms is reasonable for example to obtain confidence in the authentication performed by the IdP.

[3]The term virtual element is used to classify objects that have no actual representation in the real world.

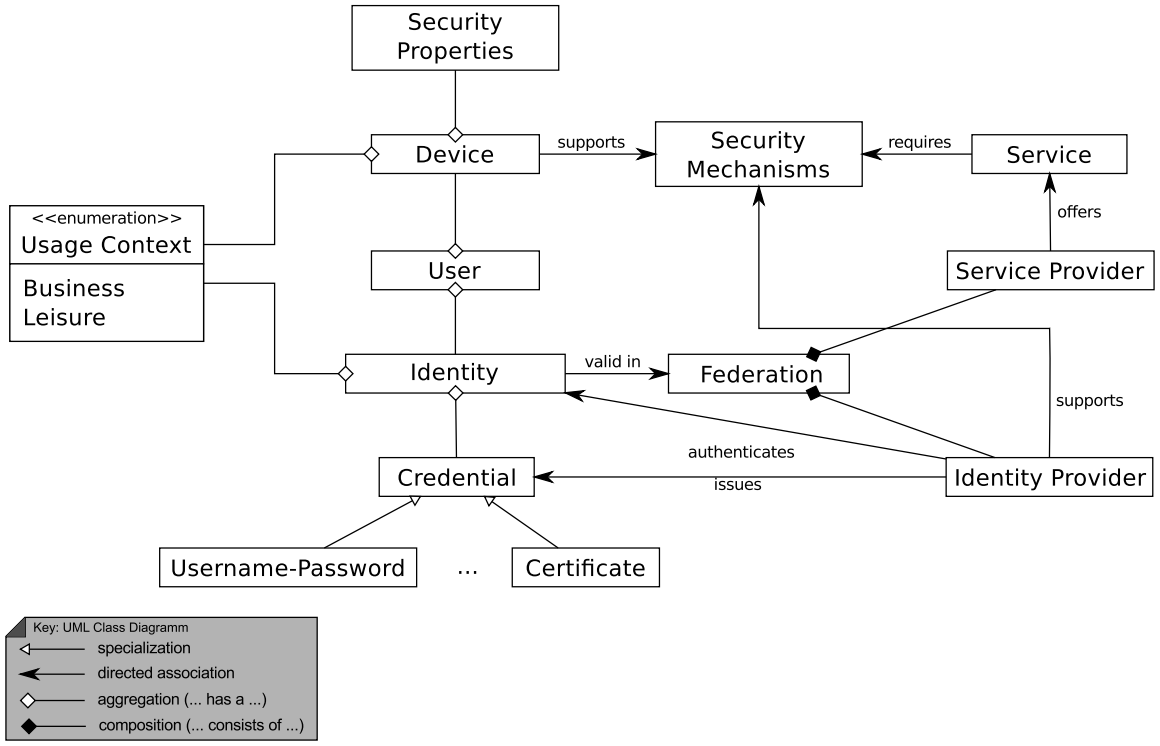


Figure 4.12: Class Diagram – User, Devices, Services

4.3.2.3 Session and Token View

In order to consume a service various steps have to be performed that lead to the establishment of sessions and the exchange of data. Figure 4.13 illustrates the dependencies. The arrows indicate the order of the data flow required to consume a service.

Starting with the authentication credentials that are assigned to an identity, it is possible to establish an IdP session. The IdP session exists between the identity, i.e. the user, and the IdP. It is characterized by an IdP token. The IdP can create an assertion that expresses that the IdP has authenticated the user successfully. Based on a trust relationship between the SP and IdP, the SP establishes the SP session with the user. The SP session is characterized by an SP Token.

4.3.3 Functional Requirements

The functional requirements that have to be fulfilled by the system are elaborated in the following. First, an overview that categorizes the requirements is provided. Afterwards, two functional requirements are discussed in more details. For a complete overview on all requirements it is referred to Section A.1.1.

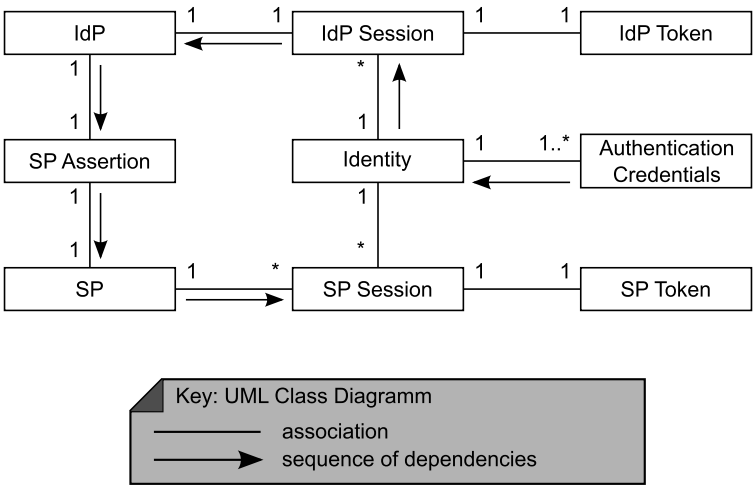


Figure 4.13: Session and Token View

4.3.3.1 Overview on Functional Requirements

Table 4.2 provides an overview on the elicited functional requirements and categorizes the requirements into eight different categories. The requirement number (No) in Table 4.2 is used to uniquely identify the requirement, whereas the short title (short) provides a semantic meaning.

4.3.3.2 Selected Functional Requirement

All functional requirements, which are itemized in Table 4.2, are elaborated in Annex A.1.1. In the following one functional requirement is exemplified:

Table 4.3 shows requirement DM-VDM-1, which is the first requirement in the category of device management (DM) that is concerned with the management of the virtual device (VDM) itself. The short title of the requirement illustrates that it must be possible to add devices to an existing virtual device. The full description details the mechanism. The only stakeholder for DM-VDM-1 is the user and the rationale provides reasons and motivations for the requirement itself. In addition, DM-VDM-1 depends on DM-SA-1, i.e. mutual authentication of devices.

Table 4.2: Overview on Functional Requirements

Category	Identifier	Short Name
Device Discovery	DM-DD-1	Device discovery in general (c.f. R4)
	DM-DD-2	Proximity detection
	DM-DD-3	Device Discovery should not reveal any information
Security Management	DM-SA-1	Mutual authentication of devices
	DM-SA-2	Secure storage of credentials
Device Management	DM-VDM-1	Adding a Device to VD
	DM-VDM-2	Removing a Device from VD
	DM-VDM-3	Device Identity
Device Characteristics	DM-DC-1	Capturing of device characteristics
	DM-DC-2	Manual editing of device characteristics
	DM-DC-3	Selection of usage context
	DM-DC-4	Data format for device characteristics
Data Management	DM-DA-1	Mechanism for data exchange
Assertion Exchange	IdM-AE-1	Protocol for the request of authentication assertions
	IdM-AE-2	Authorization based on device ID
	IdM-AE-3	Information about usage purpose
	IdM-AE-4	Manual confirmation
	IdM-AE-5	Feedback if manual confirmation is required
Identity Activation	IdM-IA-1	Protocol for identity activation (c.f. R3)
	IdM-IA-2	Authorization based on device ID
	IdM-IA-3	Feedback on device triggered for identity activation
Identity Management	IdM-IM-1	Capture existing identities
	IdM-IM-2	Store metadata on identities
	IdM-IM-3	Automatic capturing of existing identities
	IdM-IM-4	Manual adding and removal of user identities
	IdM-IM-5	Manual modification of identity information
	IdM-IM-6	Graphical user interface of identity selection
	IdM-IM-7	List of selectable identities
	IdM-IM-8	Priorities for selectable identities

Table 4.3: Requirement DM-VDM-1: Adding a Device to VD

Device Management	DM-VDM-1
Short Description	Adding a Device to VD
Full Description	A mechanism is required to add a device to the virtual device. The mechanism must be manually triggered and confirmed by the user. Adding a device to a Virtual Device must set the prerequisites to enable mutual authentication of devices.
Stakeholder	User
Rational	Users obtain new devices. It must be possible to add these devices to the virtual device. The manual trigger is required to avoid the unintentional adding of devices.
Dependencies	DM-SA-1

4.3.4 Non-Functional Requirements

In addition, to the functional requirements specified in Section 4.3.3, several non-functional requirements exist. First an overview of nun-functional requirements is provided (c.f. Table 4.4). Afterwards, one non-functional requirement is exemplified.

4.3.4.1 Overview on NonFunctional Requirements

Table 4.4 provides an overview on the elicited non-functional requirements. The columns are labeled in adherence to Table 4.2. One can distinguish between the categories of security, usability, performance and availability.

Table 4.4: Overview on Nonfunctional Requirements

Category	Identifier	Short Description
Security	NF-1	Data minimization principle
Usability	NF-2	High usability
Performance	NF-3	No performance penalties
Security	NF-4	No degradation of security
Availability	NF-5	No dependency on single devices

4.3.4.2 Selected Nonfunctional Requirement

All non-functional requirements, which are itemized in Table 4.4, are elaborated in Annex A.1.2. In the following one functional requirement is exemplified here.

Table 4.5 shows non-functional (NF) requirement NF-4 with the short title “No degradation of security”. The full description gives details on the requirement. An interest in NF-4 has the user, the IdP and the SP. The requirement does not depend on any other requirement.

Table 4.5: Requirement NF-4: No Degradation of Security

Security	NF-4
Short Description	No degradation of security
Full Description	The security of the overall solution must not be less than having individual devices.
Stakeholder	IdP, SP, User
Rational	The level of security is essential for the acceptance of the solution.
Dependencies	

4.4 Security Analysis

The term security requirement is used to address requirements related to the security of systems. Often security requirements are derived from security objectives that only manifest the need to

obey protection goals on a dedicated asset. Security requirements are categorized into four different categories:

- Functional Security Requirements: Describe the need for functionality and mechanisms that are required to guarantee the security of a system. Examples for functional security requirements are the need for authentication mechanisms or the need for encryption of data.
- Security Constraints: Constrain the design space of functional requirements. An example for a security constraint is that no identifiers that might be used to derive the user identity must be used in a protocol.
- Organizational Security Requirements: Specify additional aspects to provide security by non-technical mechanisms. Examples are the delegation of responsibilities to persons or an appropriate documentation of the system.
- Secure Coding Requirements: Define guidelines for the creation of the source code in order to implement the architecture. Examples are the avoidance of certain functions or the checking of all input data.

In the following, the focus is on functional security requirements and security constraints. Organizational security requirements and secure coding requirements are considered as an additional step that is needed to put the system into production.

4.4.1 Methodology

The applied methodology for the elicitation of security requirements consists of five steps.

- Identification of Assets: Assets have been identified based on usage scenarios, key concepts and functional requirements. Section 4.4.2 provides an overview of the identified assets.
- Threat Identification: The STRIDE methodology (see Section 2.5.2.3) is applied to identify threats. Section 4.4.3 provides an overview on the identified threats.
- Threat Description: The threat description provides additional details on the threats. Section 4.4.4 details two example threats. For the description of all threats, it is referred to Section A.1.3.
- Threat Prioritization: Since not all threats are equal, it is required to prioritize them. The used principles are elaborated below.
- Description of Security Requirements: Finally the threats are transposed to security requirements. More details on the description itself are provided below. Section 4.4.5 contains an overview on the security requirements with two sample descriptions. For the complete description of all security requirements it is referred to Section A.1.4.

4.4.1.1 Threat Description

To describe the threats, a semiformal representation is used. It consists of the following elements:

- Threat Identifier: Each threat is identified by an identifier in the format Ax_Ty. x identifies the asset, which is endangered, and y is a consecutive number.
- Threat Title: Each threat has a title for human interpretation
- Description: The threat description consists of two parts. First, the objective, elaborates the goal of a potential attacker. Second, the precondition describes what is required that the threat might materialize as attack.
- Attacker: Enumerates potential attackers that might transform the threat into an attack.
- Impact: The impact considers damage that might occur due to the threat with respect to monetary loss and loss of reputation. To quantify the impact, a discrete grading consisting of three grades has been selected:

$$\text{Impact} = \{High, Medium, Low\}$$

  - High: The impact is high, if an attacker is the position to create significant damage that cannot be recovered.
  - Medium: The impact is medium, if the damage is detectable, but recoverable. Moreover the obtained knowledge cannot be used to launch other attacks.
  - Low: The impact is low if the damage is detectable, but negligible.

- Probability of Precondition: Quantifies the probability that an attacker might fulfill the preconditions to transform a threat into an attack. Also for the quantification of the probability of preconditions, a discrete grading consisting of three grades has been selected:

$$\text{Probability of Precondition} = \{High, Medium, Low\}$$

  - High: The preconditions are often realized or it is easy to realize the preconditions.
  - Medium: The preconditions are realized from time to time.
  - Low: The preconditions are only rarely met or it is difficult to realize the preconditions.

- Priority: The priority defines the importance to consider the threat. The priority depends on the probability of preconditions and on the impact of the threat. The determination of priority values is detailed in the subsection Threat Prioritization.
- Scope: The scope defines whether the threat has to be considered by appropriate security requirements. A grading consisting of three grades has been defined:

$$\text{Scope} = \{In, Partially, Out\}$$

- In: The threat is relevant and has to be considered by the definition of security requirements.
- Partially: The threat is partially relevant has to be considered in dependence of the impact and the probability of preconditions.
- Out: The threat is not in scope and it is not required to define security requirements.

4.4.1.2 Threat Prioritization

The values of the impact of a threat as well as for the probability of preconditions are determined by hand. Combination of impact and probability allows the derivation of the priority according to the following table. The threat priority is hereby determined by the rounded-up mean value of the impact and the probability of preconditions. For example, if the impact is medium and the probability of preconditions is high, the threat priority will be high.

Table 4.6: Threat Prioritization

		Probability of Preconditions		
		High	Medium	Low
Impact	High	High	High	Medium
	Medium	High	Medium	Medium
	Low	Medium	Medium	Low

The threat priority is combined with the scope of a threat to decide whether security requirements have to be defined, i.e. the impact on security requirements. If a threat is out of scope the threat is not considered for the elicitation of security requirements. If the threat is partially in scope, the threat priority is downgraded by one level, i.e. a threat with high priority is downgraded to medium if the threat is only partially in scope. If the priority has been low and the threat is only partially in scope, the threat is not considered anymore.

$$\begin{aligned} \text{Impact on Security Requirements} &= (\text{Consideration}, \text{Security Requirement Priority}) \\ \text{Consideration} &= \{Yes, No\} \\ \text{Security Requirement Priority} &= \{High, Medium, Low\} \end{aligned}$$

Table 4.7: Threat Scoping

		Threat Priority		
		High	Medium	Low
Scope	In	(Yes, High)	(Yes, Medium)	(Yes, Low)
	Partially	(Yes, Medium)	(Yes, Low)	No
	Out	No	No	No

4.4.1.3 Description of Security Requirements

The in Section 4.4.5 enumerated security requirements are described by

- Identifier: Identifies the security requirement uniquely
- Short Name: Provides a human readable identifier for the security requirement
- Long Description: Gives a comprehensive description
- Stakeholder: Defines the stakeholders that have an interest in the fulfillment of the security requirement
- Addressed Threats: References the threats that are addressed and countered
- Security Requirement Priority: Defines the priority of the requirement (see Section 4.4.1.2)

4.4.2 Identification of Assets

This section provides an overview on the assets that can be identified on the base of the usage scenarios (Section 4.1), the key concepts (Section 4.2) and on top of the functional requirements (Section 4.3). Table 4.9 shows the identified assets. It distinguishes two different kinds of assets: physical assets and virtual assets (column Character). A physical asset is tangible. Whereas a virtual asset is non-tangible, i.e. it can be easily copied. The column "Source" gives details on the source for the given requirement.

Table 4.9: Overview on Assets

Asset	Character	Description	Source
User Device	Physical	The user owns several devices. Some of them are static others are mobile.	Figure 4.12
Virtual Device	Virtual	The virtual device is represented as composition of user devices.	Section 4.2.1
IdP Token	Virtual	Represents the state of an IdP session.	Figure 4.13
SP Token	Virtual	Represents the state of an SP session	Figure 4.13
Authentication Credentials	Virtual or Physical	Used to establish an IdP Session.	Figure 4.13
SP Assertion	Virtual	Used to establish an SP Session.	Figure 4.13
Security Properties	Virtual	Describes the security properties of a device.	Figure 4.12
Device Discovery Information	Virtual	For the discovery of other devices belonging to a virtual device, some kind of information has to be exchanged	DM-DD-3
Device Identifier	Virtual	Used for the communication/discovery with/of other devices.	DM-DM-3
Credential Store	Virtual or Physical	Stores sensitive information on a user device.	DM-SA-2
Device Characteristics	Virtual	Used for the determination of security properties.	DM-DC-2
Usage Context	Virtual	Information about the usage context of identities and devices.	DM-DC-3
Assertion Authorization	Virtual	Policies that describe who/what is allowed to request an assertion.	IdM-AE-2
Activation Authorization	Virtual	Policies that describe who/what is allowed to activate an identity.	IdM-IA-3

4.4.3 Threat Identification

Based on the identified assets in Table 4.9, the STRIDE methodology (c.f. Section 2.5.2) has been applied to identify threats. That means that every asset is considered under each of the six STRIDE threats. If an asset is subject to a threat, Table 4.10 contains a threat identifier. Each threat identifier (e.g. A1_T1) consists of the asset number (e.g. A1) and a threat identifier (T1) that is unique per asset.

Table 4.10: Identified Threats

No	Asset Description	Spoofing of User Identity	Tampering	Repudiation	Information Disclosure	Denial of Service	Elevation of Privileges
A1	User Device	A1_T1	A1_T2	A1_T3	A1_T4	A1_T5 A1_T6	A1_T1
A2	Virtual Device		A2_T1	A2_T2	A2_T3	A2_T1	
A3	IdP Token	A3_T1			A3_T1	A3_T2	
A4	SP Token	A4_T1			A4_T1	A4_T2	
A5	Authentication Credentials	A5_T1		A5_T2	A5_T1	A5_T3	
A6	SP Assertion	A6_T1			A6_T2		
A7	Security Properties		A7_T1		A7_T2	A7_T3	
A8	Device Discovery Information				A8_T1	A8_T2, A8_T3	
A9	Device Identifier				A9_T1		
A10	Credential Store				A10_T1		
A11	Device Characteristics		A11_T1		A11_T2		
A12	Usage Context		A12_T1		A12_T2		
A13	Assertion Exchange Authorization		A13_T1		A13_T2		A13_T1
A14	Identity Activation Authorization Policies		A14_T1		A14_T2		A14_T1

4.4.4 Threat Description

All threats, which are itemized in Table 4.10, are elaborated in Annex A.1.3. In the following one threat is exemplified in detail.

Threat A5-T1 in Table 4.11 describes the illegal service consumption by exploiting asset A5, i.e. authentication credentials. The objective of an attacker is to consume services on behalf of the user or to get access to user data. The preconditions specify that the attacker has to obtain the authentication credentials of the user. Since the authentication credentials are not specified in more detail, the effort to fulfill the preconditions depends on the kind of authentication credentials. An attacker might be an external attacker or the SP itself. The victims of the attack are the user and the IdP. If the authentication credentials are misused the IdP asserts wrongly a successful authentication towards other SPs and is potentially liable.

The impact is considered as being high, because an attacker that impersonates the user can consume all services that are bound to the identity. The probability of preconditions depends on the kind of credentials. Here, the probability of preconditions is high, because a username/password combination is considered to be easily obtainable. This results in a high priority of the threat. The threat is considered in scope of the system. In the discussion section some aspects are excluded from the consideration of the system. Finally, the resulting security requirements are defined.

Table 4.11: Threat A5-T1 – Illegal Service Consumption

Illegal Service Consumption	A5-T1
Objective: An attacker uses user's authentication credentials to consume service on behalf of the user and to get access to user data bound to user's identity. Prerequisite: The attacker must obtain the authentication credentials. Depending on the kind of authentication credentials, the difficulty to obtain them is different. The authentication credentials cannot only be obtained from the user, but also from the IdP.	
Attacker: External Attacker, SP	Victim: User, IdP
Impact: High	Precondition: High
Priority: High	Scope: In
Discussion: Social Engineering is out of scope. We do not consider attacks based on password phishing or similar vulnerabilities.	
Security Requirements: SR_14, SR_15	

4.4.5 Overview of Security Requirements

Table 4.12 provides an overview on the derived security requirements. A security requirement specifies either a concrete security mechanism or specifies constraints on dedicated functionality. In addition, security requirements SR-100 to SR-103 specify additional requirements regarding session management functionality. Such functionality seems to be reasonable to increase usability and thus security. A detailed discussion of one example security requirements follows in Section 4.4.6.

4.4.6 Selected Security Requirements

All security requirements, which are itemized in Table 4.12, are elaborated in Annex A.1.4. In the following one security requirement is exemplified.

Table 4.13 specifies security requirement SR-6, i.e. the need for a mechanism to limit the rate of identity activation and assertion exchange requests. The goal of a mechanism that addresses SR-6 is to prevent DoS attacks. The requirement addresses threat A1-T6 and A1-T5.

Table 4.12: Identified Security Requirements

No	Short Name
SR-1	Mechanism for prevention of unauthorized device usage
SR-2	Mechanism for removing devices from virtual device
SR-3	Logging of service consumption
SR-4	No dependency on single device
SR-5	Authorization mechanism for modifying a virtual device
SR-6	Rate Limiting for identity activation and assertion exchange
SR-7	Mechanism to obtain information about virtual device composition
SR-8	Logging for virtual device modification
SR-9	Confidentiality of information about virtual device composition
SR-10	Secure transmission of IdP token
SR-11	Secure storage of IdP token
SR-12	Secure transmission of SP token
SR-13	Secure storage of SP token
SR-14	Secure storage of authentication credentials
SR-15	Secure transmission of authentication credentials
SR-16	Logging mechanism for authentication credential usage
SR-17	Secure transmission of SP assertions
SR-18	Secure storage of security properties
SR-19	Logging mechanism for changed security properties
SR-20	storage of all security properties only on secure devices
SR-21	Secure device discovery
SR-22	Availability of device discovery
SR-23	Encrypt device identifiers or avoid unique device identifiers
SR-24	Access control on credential store
SR-25	Secure storage of device characteristics
SR-26	Logging mechanism for changed device characteristics
SR-27	Secure storage of usage context
SR-28	Logging mechanism for usage context change
SR-29	Secure storage of assertion exchange authorization policies
SR-30	Logging mechanism for assertion exchange authorization policy change
SR-31	Secure storage of identity activation authorization policies
SR-32	Logging mechanism for identity Activation Authorization Policy Change
SR_100	Mechanism to stop all IdP sessions
SR_101	Mechanism to stop IdP sessions on one device
SR_102	Mechanism to stop all SP sessions
SR_103	Mechanism to stop all SP sessions on one device

Table 4.13: Security Requirement SR-6 – Rate Limiting for Identity Activation and Assertion Exchange

SR-6	Rate Limiting for Identity Activation and Assertion Exchange
The rate of identity activation requests and Assertion Exchange requests should be limited to avoid DoS attacks and prevent malicious behavior. The rate must be adjustable by the owner of the virtual device.	
Stakeholder	0
Addressed Threats	A1-T6, A1-T5
Priority	Medium

4.5 Functional Architecture

The functional architecture addresses the elicited requirements, partitions the system into different blocks and puts them into relation to each other. This allows the fulfillment of the challenges exposed in the usage scenarios.

The functional architecture consists of the following parts:

- Functional Blocks: Realize the functionality within a device and within the virtual device composition. Functional blocks are composed of functional subblocks.
- Inter-device Interfaces: Inter-device interfaces are used for the communication and collaboration of devices that are part of the virtual device.
- Intra-device Interfaces: Intra-device interfaces interconnect the functional blocks residing within a device.

The semi-formal description of the functional architecture uses UML component diagrams. Component diagrams allow a step-by-step refinement of the architecture description. Functional blocks are modeled as components, i.e. the terms functional block and component are used synonymously in the following. The functional blocks operate on data models that have been specified using class diagrams. Component diagrams and class diagrams have been slightly adapted for simplification and improved readability. Section B provides details on the used modeling techniques and highlights modifications of the applied UML diagrams.

Section 4.5.1 gives an overview on the functional architecture. It introduces the virtual device view that highlights the inter-device interfaces and gives an overview on a single device with its functional blocks that are interconnected by intra-device interfaces. Afterwards the identified functional blocks are refined ($\rightarrow$ Section 4.5.2 - Section 4.5.5) regarding their internal structure and the used data models. The provided and required intra-device interfaces and the addressed requirements are also described.

4.5.1 Overview on Building Blocks

4.5.1.1 Virtual Device View

Figure 4.14 indicates the interfaces between the devices that are part of a virtual device. Each interface groups dedicated functionality and is detailed in Chapter 5. It is possible to distinguish between interfaces that rely on the existence of a secure channel[4] between the devices and those that do not require a secure channel.

- *IdentityManagement Interface (If-IdM)*: The *If-IdM* provides functionality that relates to the realization of the Multi-device IdM concept. It operates on top of a secure channel.

[4]This thesis assumes a mutually authenticated channel that provides confidentiality and integrity protection.

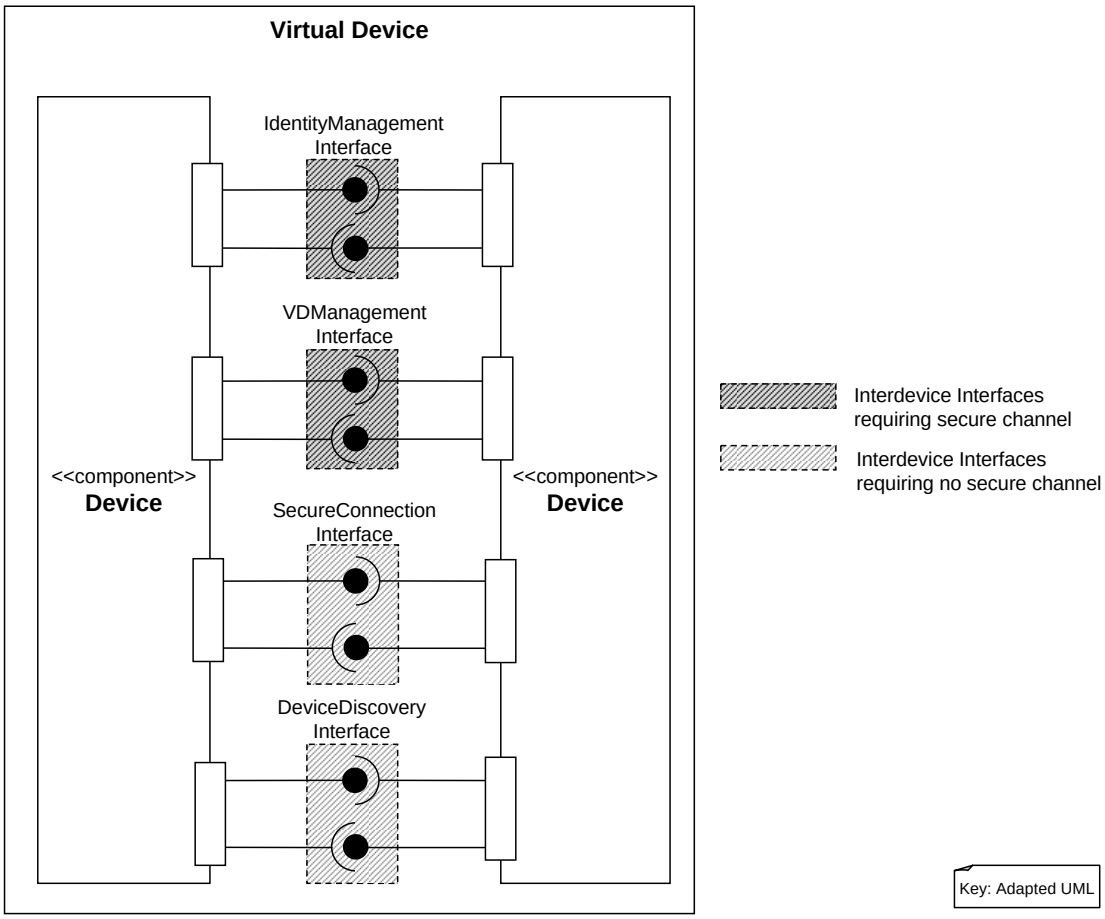


Figure 4.14: Component Diagram of Virtual Device – Virtual Device View

- *VirtualDeviceManagement Interface (If-VDM)*: The *If-VDM* provides functionality for the management of the virtual device. It operates on top of a secure channel.

- *SecureConnection Interface (If-SC)*: The *If-SC* provides functionality for the establishment of a mutually authenticated, confidentiality-protected and integrity-protected channel between devices that are part of a virtual device. It requires the prior establishment of a security association between the devices.

- *DeviceDiscovery Interface (If-DD)*: The *If-DD* provides functionality for the mutual discovery of devices that belong to the virtual device. Since it is required to bootstrap the secure channel required for the *If-IdM* and for the *If-DM*, it does not require the existence of a secure channel.

Figure 4.15 illustrates the dependencies of the interface, i.e. which interface depends on the functionality provided by another interface. The upper layer depends on the functionality provided by the lower layer. The *If-SC* depends on the *If-DD*, whereas *If-IdM* and *If-DM* depend on the existence of secure channels provided by the *If-SC*.

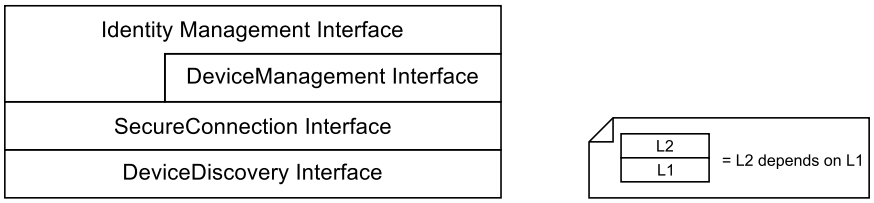


Figure 4.15: Dependencies of Interfaces

4.5.1.2 Intra-Device View

Within a device, which is part of the virtual device, four functional blocks have been identified based on the requirements. Each functional block is composed of several functional subblocks.

- *Identity Manager (IM)*: The *IM* is responsible for the management of user's identities. This comprises functionality for the interworking with applications that request the activation of identities, interfaces for the configuration of identities, and filtering and selection algorithms for identities.
- *Identity Transfer Enabler (ITE)*: The *ITE* complements the *IM* regarding the virtual device view. It realizes the protocols for the exchange of information about identities between devices, the activation of identities on remote devices and the retrieval of assertions from remote devices to consume services.
- *Device Manager (DM)*: The *DM* realizes the core functionality required to operate a virtual device. This includes device discovery, management of the virtual device composition, establishment of secure channels between devices, and collection and exchange of data describing the device characteristics.
- *Secure Storage Enabler (SSE)*: The *SSE* is a utility component to securely store data on devices. In addition it is responsible for recording logging data. The provided functionality is used by the *IM* and by the *DM*.

These functional blocks are connected by inter-device and intra-device interfaces as depicted in Figure 4.16. Details on the interfaces are provided below. Each functional block is described in the following with its internal structures, i.e. detailed view, the used data model, the provided and required interfaces, and by an explanation of the addressed requirements.

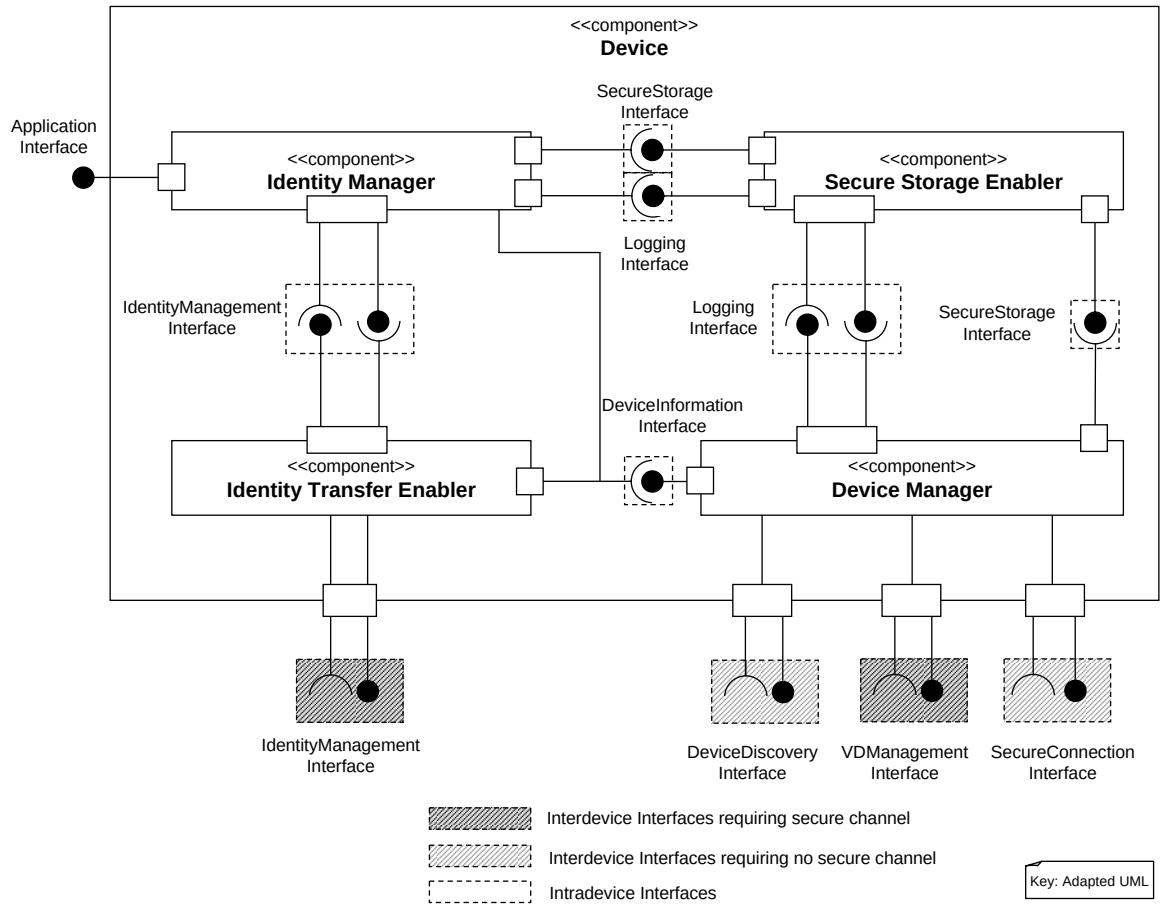


Figure 4.16: Component Diagram of Individual Device – Intra-Device View

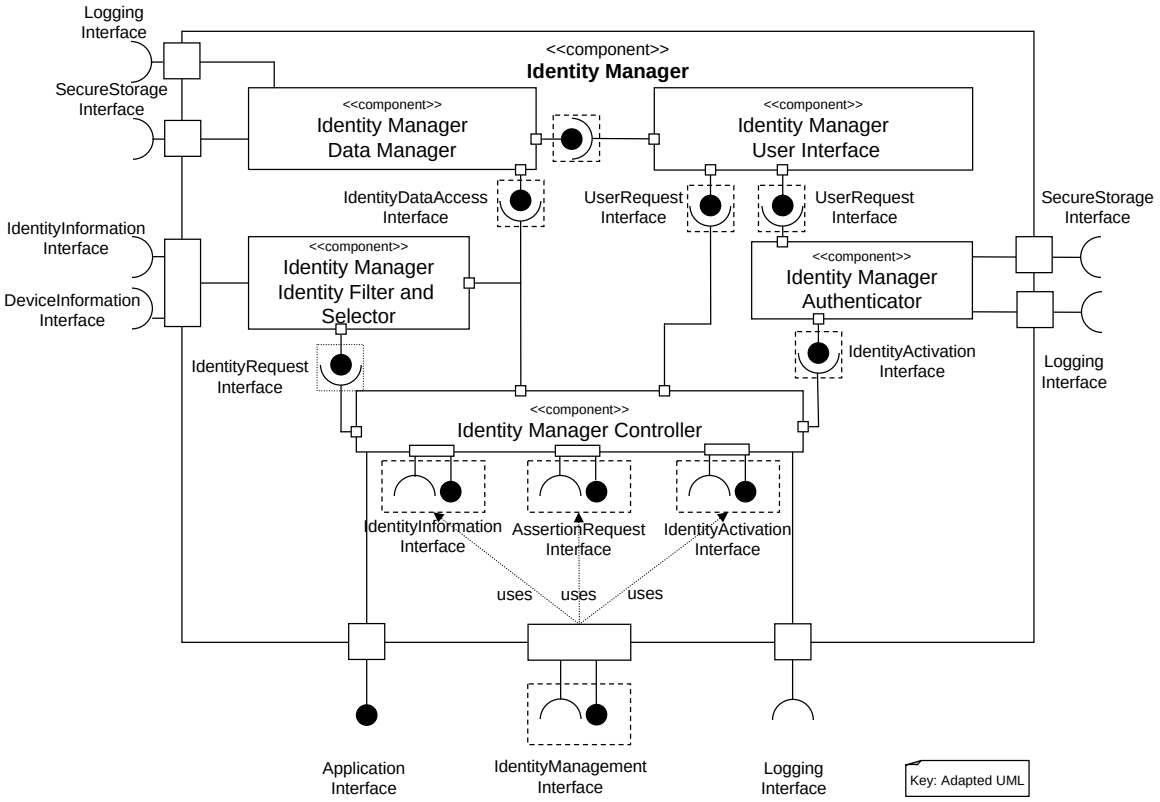


Figure 4.17: Component Diagram of Identity Manager

4.5.2 Identity Manager

4.5.2.1 Detailed View

The *Identity Manager (IM)* consists of five functional subblocks (→ Figure 4.17).

- *Identity Manager Controller (IM-C)*: The *IM-C* is the central component of the IM. It receives requests for identity activation and creation of SP assertions from applications via the *Application Interface (If-App)* or from other devices that are part of the virtual device via the *If-IdM* from the *ITE* (→ Section 4.5.3). The *IM-C* has the possibility to log events for auditing purposes using the *Logging Interface (If-Log)*. For interaction with the user, the *UserRequestInterface (If-UR)* is used.

- *Identity Manager User Interface (IM-UI)*: The *IM-UI* provides functionality to interact with the user. The provided functionality is used by the *Identity Manager Authenticator* and by the *IM-C*. In addition the user can modify identity data stored within the *IM-DM*.

- *Identity Manager Filter and Selector (IM-IFS)*: The *IM-IFS* contains filtering algorithms to support the user with the selection of possible identities. The filtering process is detailed in Section 5.2. It uses two information sources about available identities. First, information about locally available and potentially active identities is obtained from the Identity Data Manager by using the *IdentityDataAccess Interface (If-IDA)*. Second, information about identities that are available or active on other available devices that are part

of the VD is obtained from the *ITE* via the *IdentityInformation Interface (If-II)* in combination with the *DeviceInformation Interface (If-DI)* that provides information about the available devices.

- *Identity Manager Authenticator (IM-A)*: The *IM-A* implements actual authentication algorithms. Based on an extendable concept, existing authentication algorithms can be integrated. Required credentials can be securely stored using the *SecureStorage Interface (If-SS)*. For interactions with the user the interface to the *IM-UI* is used.

- *Identity Manager Data Manager (IM-DM)*: The *IM-DM* is a support component. It provides structured access to identity data, which is securely stored by using the *If-SS*. Identity data can be manipulated via a graphical user interface provided by the *IM-UI*.

4.5.2.2 Data Model

The *IM* requires a data model for its operation. Figure 4.18 specifies the used data model. An identity (→ Figure 4.12) consists of attributes that are associated with an identity identifier.

An identity has an associated state. If an IdP Session exists (→ Figure 4.13), the identity is considered to be active. If no IdP Session exists, but the identity could be enabled on the given device or on another device that is part of the virtual device, the identity is considered to be activatable. In all other cases, an identity is considered as not useable. The beginning and the supposed end of an IdP session are stored as time instances.

The IdP determines the authentication methods that are supported and that might by used for authentication. Moreover, the IdP issues appropriate authentication credentials that are bound to the identity identifier.

The usage context and the security level complement the meta data on identities. Both provide the possibility to specify additional constraints on the usage of identities on devices. Hereby, the security level describes the minimum level of security that a device has to fulfill in order that the identity can be activated on that device.

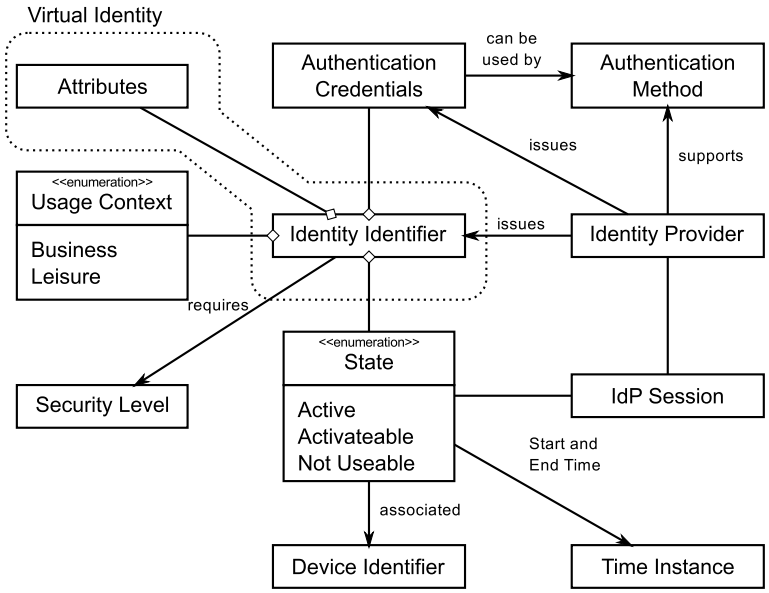


Figure 4.18: Identity Data Model for Internal Decisions

4.5.2.3 Intra-Device Interfaces

The *IM* requires or provides the following interfaces to other components (→ Table 4.14):

Table 4.14: Intradevice Interfaces provided/required by *IM*

Functional Subblock	Interface	Required/ Provided	Explanation
IM-C	If-App	Provided	Used by applications to make use of the identity services (c.f. for specification)
	If-IdM	Provided	Used by the ITE to request usage of identities on local devices.
		Required	Uses the ITE to request usage of identities on remote devices.
	If-Log	Required	Used to log activation/deactivation of identities.
IM-IFS	If-DI	Required	Used to obtain information about available devices and their characteristics.
	If-II	Required	Used to obtain information about identities on other devices that are part of the virtual device
IM-DM	If-SS	Required	Used to securely store and retrieve sensitive identity information on the local device. Uses the data model from Figure 4.18.
	If-Log	Required	Used to log changes of identity data.
IM-A	If-SS	Required	Used to securely store and retrieve authentication credentials on the local device.
	If-Log	Required	Used to log events regarding the usage of authentication credentials.

4.5.2.4 Addressed Requirements

The *IM* addresses the following requirements:

Table 4.15: Requirements addressed by the Identity Manager

Requirement	Addr.	Explanation
R2: Task distribution	Yes	The *IM-C* is able to trigger the activation of identities via the *ITE*. The *IM-IFS* is responsible for deciding about the task distribution.
R7: Determination of usage context	Yes	The *IM-IFS* takes the usage context into account, which the user can configure for identities via the *IM-UI*.
IdM-IM-1: Capture existing identities	Yes	The user has the possibility to add his identities to the *IM-ID* via the corresponding user interface provided by the *IM-UI*.
IdM-IM-2: Store metadata on identities	Yes	The *IM-ID* with the corresponding data model (c.f. Figure 4.18) provides the possibility to store identity information.
IdM-IM-3: Automatic capture of existing identities	Partially	An automatic capturing of existing identities, requires the integration into the application interface. This requirement is considered as optional in the following.
IdM-IM-4: Manual adding and removal of user identities	Yes	The user has the possibility to add his identities to the *IM-ID* via the corresponding user interface provided by the *IM-UI*.
IdM-IM-5: Manual modification of identity information	Yes	The *IM-UI* provides together with the *Identity Data Access Interface*the possibility to modify meta data associated to identities.
IdM-IM-6: Graphical user interface for identity selection	Yes	The *IM-UI* provides a user interface for identity selection. In addition it is possible for an application for implement a corresponding functionality by using the application interface.
IdM-IM-7: List of selectable identities	Yes	The *IM-IFS* filters user's identities and provides a sorted list of identities.
IdM-IM-8: Priorities of selectable identities	Yes	The *IM-IFS* provides the possibility to sort the list according to various priorities, e.g. prefer active identities, prefer locally useable identities.
SR-3: Logging of service consumption	Yes	The *IM-IMC* logs events regarding the activation and deactivation of identities as well as assertion and identity activation requests from local or remote.

Continued on next page

Table 4.15: Requirements addressed by the Identity Manager

Requirement	Addr.	Explanation
SR-10: Secure transmission of IdP token	No	The *IM-A* relies for the secure transmission of the IdP-Token on secure channel between the device and the IdP. The establishment of such a secure channel is important but not considered in more detail in the following.
SR-11: Secure storage of IdP token	Yes	The *IM-A* uses the *If-SS* to store IdP Tokens.
SR-12: Secure transmission of SP token	No	The *IM-A* relies for the secure transmission of SP-Tokens on a secure channel between the device and the IdP as well as between the device and the SP. The establishment of such a channel is important but not considered in more detail in the following.
SR-13: Secure storage of SP token	Yes	The *IM-A* uses the *If-SS* to store SP Tokens.
SR-14: Secure storage of authentication credentials	Yes	The *IM-A* uses the *If-SS* to store authentication credentials.
SR-15: Secure transmission of authentication credentials	No	The *IM-A* relies for the secure transmission of authentication credentials on a secure channel between the device and the IdP.
SR-16: Logging mechanism for authentication credential usage	Yes	The *IM-A* logs the usage of authentication credentials. Sensitive information (i.e. passwords) is not stored.
SR-18: Secure storage of security properties	Yes	The *IM-DM* stores all information about identities by using the *If-SS*.

4.5.3 Identity Transfer Enabler

4.5.3.1 Detailed View

The *Identity Transfer Enabler (ITE)* extends the *IM* with regard to multiple devices and enables the usage of identities across devices. It consists of three components (→ Figure 4.19):

- *Identity Transfer Enabler Controller (ITE-C)*: The *ITE-C* is the central component of the *ITE*. It interacts with the other devices that are part of the virtual device in order to exchange information about the available identities, activate identities on remote devices or retrieve SP assertions from remote devices. This is achieved by the *IdentityManagement Interface* that connects the *ITE* on one hand to the local *IM* and on the other hand to remote devices via a secure channel. Information and details about the available devices are obtained by using the *DeviceInformation Interface (If-DI)*.

- *Identity Transfer Enabler Data Manager (ITE-DM)*: The *ITE-DM* uses the same data model as the *IM-DM* in order to store meta data on identities. The temporarily stored information is accessible by the *IM-IFS* through the If-II. Since the information regarding identities on remote devices is of temporal nature, there is no need for permanent storage. Therefore, the *ITE-DM* does not implement the SecureStorage Interface.

- *Identity Transfer Enabler Authorization Manager (ITE-AM)*: The *ITE-AM* authorizes requests from other devices regarding identity activation and SP assertion retrieval. Authorization decisions are based on policies that are stored using the *If-SS*.

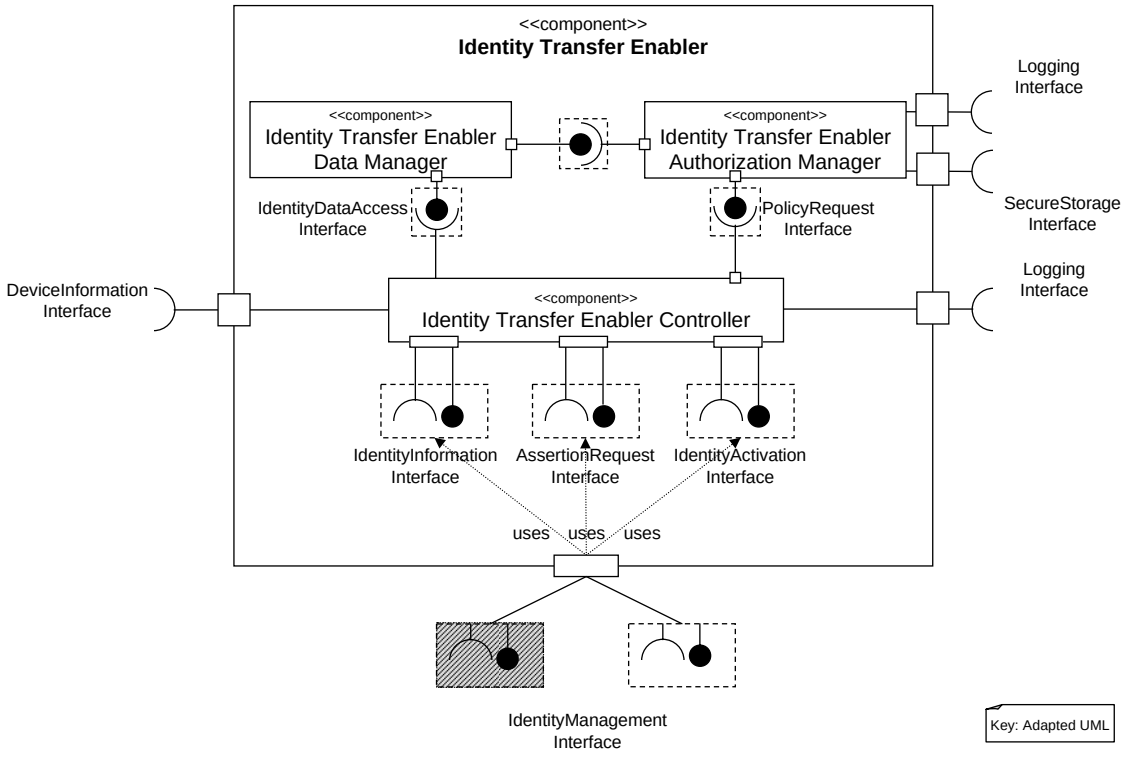


Figure 4.19: Component Diagram of Identity Transfer Enabler

4.5.3.2 Data Model

The *ITE* uses the same data model as the *IM* (→ Figure 4.18).

4.5.3.3 Intra-Device Interfaces

The *ITE* requires or provides the following interfaces:

Table 4.16: Interfaces provided/required by Identity Transfer Enabler

Functional Subblock	Interface	Required/ Provided	Explanation
ITE-C	If-IdM[5]	Provided	Used by the local IM to request usage of identities on remote devices.
		Provided	Used by ITE-C on remote devices to request usage of identities on local device
		Required	Uses the ITE-C of remote devices to request usage of identities on remote devices.
		Required	Uses the local IM to realize usage requests for identities from remote devices
	If-Log	Required	Used to log request for usage of identities from remote devices.
	If-DI	Required	Used to obtain information about available devices.
ITE-AM	If-SS	Required	Used to store access control policies.

[5]The *If-IdM* exists in two instantiations. One instantiation is for intra-device usage, whereas the other version is for inter-device usage, i.e. a secure channel is required.

4.5.3.4 Addressed Requirements

The *ITE* addresses the following requirements:

Table 4.17: Requirements addressed by the Identity Transfer Enabler

Requirement	Addr.	Explanation
R3: Remote activation	Yes	Remote identity activation is possible with the *If-IdM*.
R8: Distributed data handling	Partially	The *If-IdM* is used to transfer identity data between different devices.
IdM-AE-1: Protocol for the request of authentication assertions	Yes	Realized by the *AssertionRequest Interface (If-AR)*, which is part of the *If-IdM* and realizes the Assertion Request Protocol. The *If-AR* allows the retrieval of assertions from a remote device. For details on the protocol it is referred to subsection 5.3.3.
IdM-AE-2: Authorization based on device identity	Yes	Upon reception of an assertion request from a remote device, the *ITE-AM* has to authorize the request. As part of the assertion request protocol, the device identity has to be provided. Depending on the decision, the assertion request is allowed or denied.
IdM-AE-3: Information about usage purpose	Yes	As part of an assertion request or identity activation request, the usage context has to be provided. This is realized as part of the AssertionRequest Interface and within the IdentityActivation Interface.
IdM-AE-4: Manual confirmation	Yes	Realized within the IdentityInformation Interface that connects the *ITE* to the *IDM*. Allows the *ITE* to request user confirmation via the user interface provided by the *IM*.
IdM-AE-5: Feedback if manual confirmation is required	Yes	Has to be realized within the IdentityInformation Interface that connect the ITE to the IDM, which provides a user interface.
IdM-IA-1: Protocol for identity activation	Yes	Realized by the *Identity Activation Interface (If-IA)*, which is part of the *If-IdM* and realizes the Identity Activation Protocol. The *If-IA* allows the activation of identities from remote devices. For details on the protocol it is referred to subsection 5.3.2.

Continued on next page

Table 4.17: Requirements addressed by the Identity Transfer Enabler

Requirement	Addr.	Explanation
IdM-IA-2: Authorization based on device identity	Yes	Upon reception of an identity activation request from a remote device, the *ITE-AM* has to authorize the request. As part of the identity activation protocol, device identity has to be provided. Depending on the decision, the identity activation request is allowed or denied.
DM-DA-1: Mechanism for data exchange	Yes	Via the *Identity Information interface (If-II)* devices exchange properties of identities, e.g. the State of the identity, the configured usage context, etc.
SR-3: Logging of service consumption	Yes	All assertion and identity activation requests are logged by the *ITE-ITC* using *If-Log*.
SR-4: No dependency on single device	Yes	If a previously discovered device disappears, the Assertion Request Protocol as well as the Identity Activation Protocol provide time out mechanism.
SR-6: Rate Limiting for identity activation and assertion exchange	Yes	The *ITE-ITC* maintains an internal database that counts the number of assertion and identity activation requests from remote device. If a previously configured threshold is exceeded further requests are blocked.
SR-17: Secure transmission of SP assertions	Yes	The *If-AR* relies on a secure channel for the transmission of SP assertions between devices belonging to the same virtual device.
SR-29: Secure storage of assertion exchange authorization policies	Yes	The *ITE-AM* uses the *If-SS* to securely store policies
SR-30: Logging mechanism for assertion exchange authorization policy change	Yes	The *ITM-AM* uses the *If-Log* to log authorization policy changes.
SR-31: Secure storage of identity activation authorization policies	Yes	c.f. SR-29
SR-32: Logging mechanism for identity activation policy change	Yes	c.f. SR-30

4.5.4 Secure Storage Enabler

4.5.4.1 Detailed View

The *Secure Storage Enabler (SSE)* provides functionality for the secure storage of information and collection of logging functionality. This functionality is essential for the other three building blocks. It consists of two functional subblocks (→ Figure 4.20):

- *Secure Storage Service (SSE-SSS)*: The *SSE-SSS* offers the SecureStorage Interface, which is used to securely store data. If available the secure storage uses encrypted storage or dedicated hardware.
- *Logging Service (SSE-LS)*: The *SSE-LS* is used by other functional blocks to log events. This is essential for the auditing of incidents within the virtual device. It is possible to chain the *SSE-LS* running on different devices.

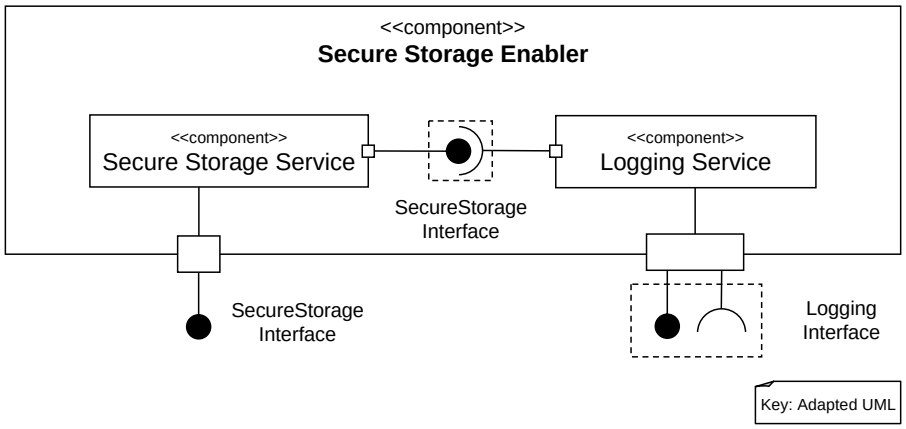


Figure 4.20: Component Diagram of Identity Manager

4.5.4.2 Data Model

The data model of the *SSE-SSS* does not need to be specified. The SecureStorage Interface provides access to a document oriented database system. A document oriented database system provides primitives for adding, deleting and editing documents. In consequence each functional block can maintain its data model and the *SSE-SSS* is reduced to storing simple documents with the corresponding metadata.

The *SSE-LS* requires a data model to store log entries. Figure 4.21 illustrates the data model. It allows the registration of the device identifier, the functional block and the functional subblock, respectively, as well as the time instance and the log message itself.

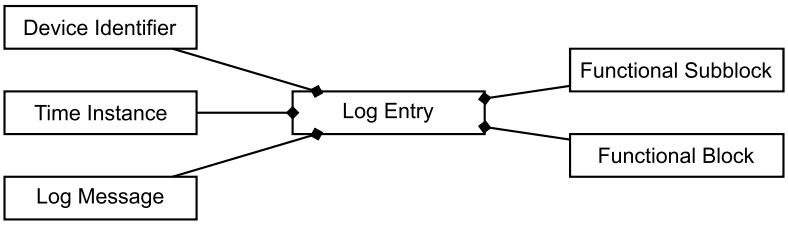


Figure 4.21: Data Model for Log Entries

4.5.4.3 Intra-Device Interfaces

The *SSE* requires or provides the following interfaces:

Table 4.18: Interfaces provided/required by Secure Storage Enabler

Functional Subblock	Interface	Required/ Provided	Explanation
SSE-SSS	If-SS	Provided	Document-oriented database interface to securely store sensitive data
SSE-LS	If-L	Provided	Used by other functional block to store logging entries.
		Required	Allows to forward all logging data for another Logging Service

4.5.4.4 Addressed Requirements

The *SSE* addresses the following requirements:

Table 4.19: Requirements addressed by the Secure Storage Enabler

Requirement	Addr.	Explanation
DM-SA-2: Secure storage of Credentials SR-9: Confidentiality of virtual device composition SR-11: Secure storage of IdP token SR-13: Secure storage of SP token SR-14: Secure storage of authentication credentials SR-18: Secure storage of security properties SR-20: Storage of all security properties only on a secure device SR-24: Access control for secure storage SR-25: Secure storage of device characteristics SR-27: Secure storage of usage context SR-29: Secure storage of assertion exchange authorization policies SR-31: Secure storage of identity activation authorization policies	Yes	The *SSE-SSS* addresses the need for secure storage of various kinds of information.

Continued on next page

Table 4.19: Requirements addressed by the Secure Storage Enabler

Requirement	Addr.	Explanation
SR-3: Logging of service consumption SR-8: Logging for virtual device modification SR-16: Logging mechanism for authentication credential usage SR-19: Logging mechanism for changed security properties SR-26: Logging mechanism for changed device characteristics SR-28: Logging mechanism for usage context change SR-30: Logging mechanism for assertion exchange authorization policy change SR-32: Logging mechanism for identity activation authorization policy change	Yes	The *SSE-LS* addresses the need to collect logging information about the various events that took place.

4.5.5 Device Manager

4.5.5.1 Detailed View

The *Device Manager (DM)* consists of six components (→ Figure 4.22):

- *Virtual Device Manager (DM-VDM)*: The *DM-VDM* is responsible for the management of the virtual device. It has knowledge about the individual devices that are part of the virtual device and stores this information using the *If-SS* and the data model introduced below. For the communication with other devices the *DeviceManagment Interface (If-DM)* is used. It consists of the following parts:

  The *VDModification Interface (If-VDMod)* is used to modify the virtual device, i.e. adding or removing individual devices to the virtual device composition. This interface is required by the *Device Manager User Interface (DM-UI)* to provide a graphical user interface for the modification of the virtual device. Via the *DeviceInformation Interface (If-DI)* it is possible to exchange characteristics (→ data model below) of devices among each other, e.g. if device properties have changed. Via *If-Log* it is possible that one device obtains logging information from all other devices. For security critical actions (i.e. modification of the virtual device and exchange of information between the devices belonging to the virtual device), the *DM-VDM* depends on the *Virtual Device Authorization Manager*. Via the *PolicyRequest Interface (If-PR)* authorization for security critical actions can be requested.

- *Secure Connection Broker (DM-SCB)*: The *DM-SCB* is an enabler component to establish secure channels between devices. A secure channel is established after mutual authentication of devices and provides message confidentiality and integrity protection. An appropriate technology for the establishment of secure channels is TLS. For the establishment of secure channels appropriate credentials (e.g. certificates) are required.

- *Local Device Manager (DM-LDM)*: The *DM-LDM* captures local device characteristics and makes the device properties accessible to the *DM-VDM* via the *If-DI*. Device characteristics are stored using the *If-SS*. In addition the *DM-LDM* is responsible for the management of the usage context of a device. All information can be accessed and modified by the *DM-UI*.

- *Device Discovery Service (DM-DDS)*: The *DM-DDS* discovers devices of the virtual device composition that are available. In addition it is able to determine whether the device is in the proximity. The realized *DeviceDiscovery Interface (If-DD)* is subject to special security requirements, since device discovery might reveal information. Internally, it provides the *DeviceAvailability Interface (If-DA)* which is used to obtain information about available devices.

- *Device Manager User Interface (DM-UI)*: The *DM-UI* is used for the interaction with the user. The user can access and modify information regarding the virtual device (i.e. changing device properties, removing, adding devices) and regarding the local device.

- *Virtual Device Authorization Manager (DM-VDAM)*: The *DM-VDAM* represents a policy decision point. Based on policies that are stored using the *If-SS*, the *DM-VDAM* decides

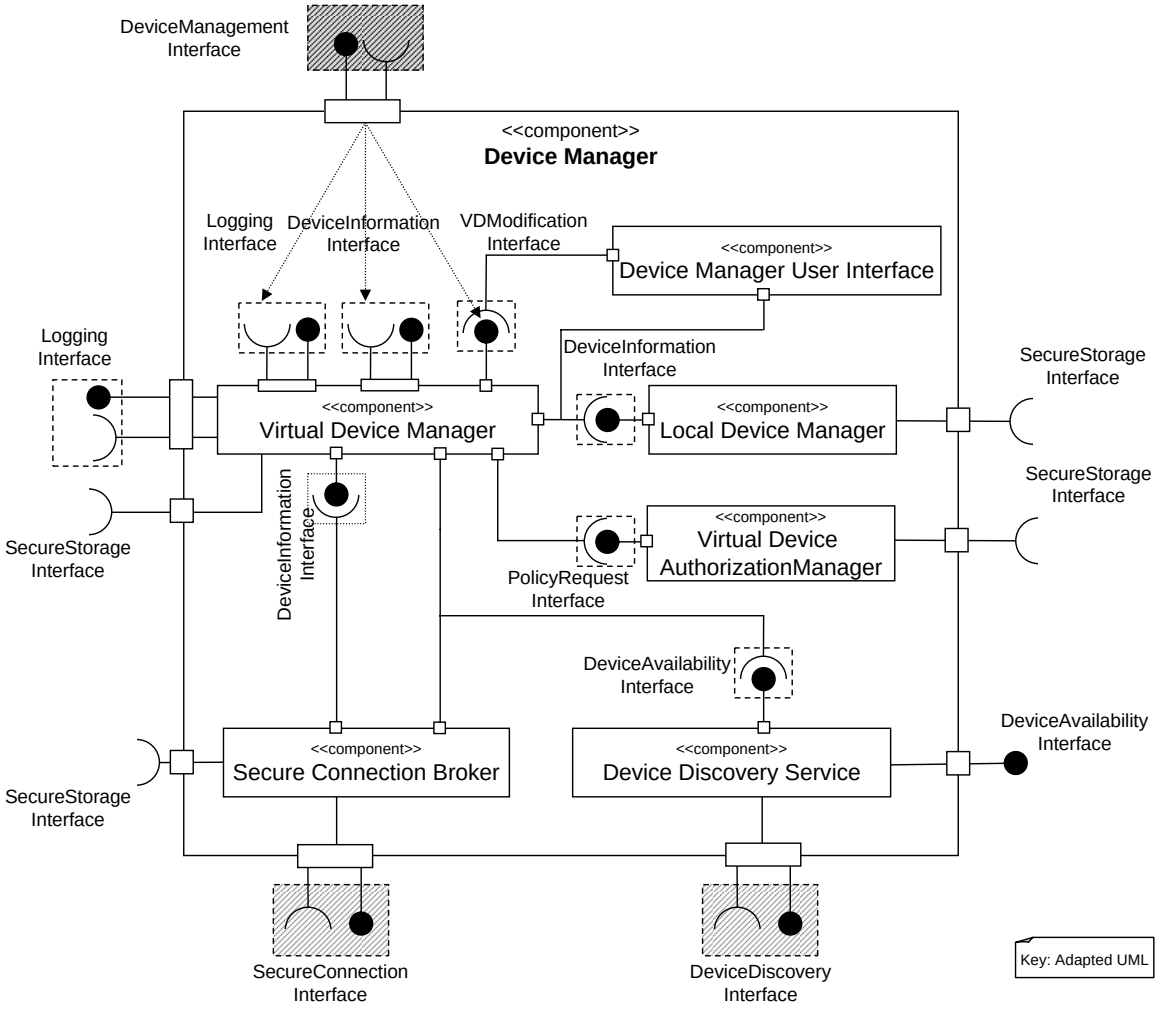


Figure 4.22: Component Diagram of Device Manager

whether authorization should be granted or not. The requesting component enforces the decision.

4.5.5.2 Data Model

For the description of device characteristics, the data model depicted in Figure 4.23 has been specified. A virtual device is considered as an association between devices, whereas devices are identified by device identifiers.

Various kinds of information are bound to the device identifier. One or several usage contexts determine the context in which the device can be used regarding service usage. The security characteristics describe the device regarding its security capabilities. This comprises supported authentication methods, supported security protocols and the availability of secure storage.

In contrast, the security properties are used to determine a security level for a device. Security properties describe the installed software on a device. This is based on the assumption that

the installed software is significant for potential vulnerabilities. Specializations of software descriptors detail the installed operating system with the corresponding patch level or descriptors that describe the existence of anti virus software.

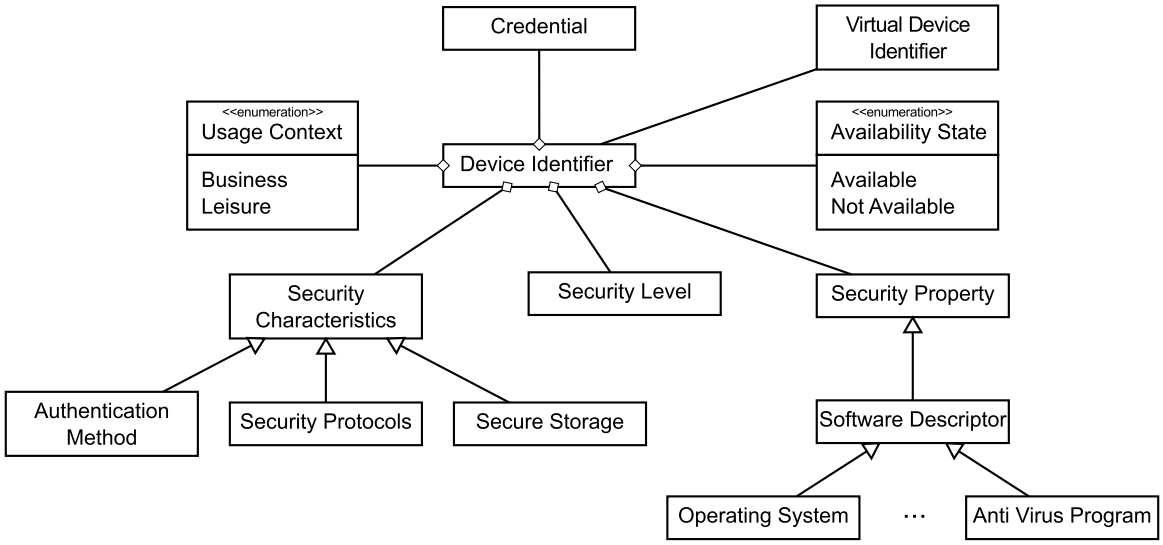


Figure 4.23: Device Data Model for Internal Decisions

4.5.5.3 Intra-Device Interfaces

The *DM* requires or provides the following interfaces:

Table 4.20: Interfaces provided/required by the Device Manager

Functional Subblock	Interface	Required/ Provided	Explanation
DM-VDM	If-Log	Provided	Allows the SSE-LS to transfer log entries to another device.
		Required	Allows a remote device to add log entries to the local SSE-LS.
	If-SS	Required	Information about other devices is stored using If-SS.
	If-DI	Required	Used to retrieve information about the local device from the DM-LDM
		Provided	Used to provide information about the local device or devices that are part of the virtual device to other devices, which are also part of the virtual device.
	If-DA	Required	Used to obtain information about available devices that are part of the virtual device
	If-VDMod	Provided	Used to modify the virtual device, i.e. adding or removing a device.
DM-LDM	If-DI	Provided	Used to provide information about the local device.

Continued on next page

Table 4.20: Interfaces provided/required by the Device Manager

Functional Subblock	Interface	Required/ Provided	Explanation
	If-SS	Required	Used to store information about the local device. Required in particular for manual modification of device properties, if automatic collection is not feasible.
DM-SCB	If-SS	Required	Used to store credentials that are required to establish a secure channel.
	If-SC	Required	Used to establish secure channels with other devices.
		Provided	Used to establish secure channels with other devices.
DM-DD	If-DA	Provided	Provides information about the availability of devices.

4.5.5.4 Addressed Requirements

The *DM* addresses the following requirements:

Table 4.21: Requirements addressed by the Device Manager

Requirement	Addr.	Explanation
R1: Secure exchange	Yes	The *DM-SCB* is responsible for the establishment of secure channels between devices.
R2: Task distribution	No	The distribution of tasks is not considered.
R4: Discovery of user devices	Yes	The *DM-DDS* addresses the discovery of user devices.
R5: Capture of device characteristics	Yes	The *DM-LDM* captures device properties.
R6: Establishment of security associations	Yes	The *DM-VDM* creates and maintains security associations with other devices of the virtual device.
R7: Determination of usage context	Yes	The *DM-UI* is used to configure the usage context of a device. It is stored with the *DM-LDM*.
R8: Distributed data handling	Partially	The *DM-VDM* provides a mechanism for the exchange of data characteristics.
DM-DD-1: Device discovery in general	Yes	c.f. R4
DM-DD-2: proximity detection	Yes	In addition to the discovery of devices, the *DM-DDS* is able to provide information about the proximity of devices belonging to a virtual device.

Continued on next page

Table 4.21: Requirements addressed by the Device Manager

Requirement	Addr.	Explanation
DM-DD-3: Device discovery should not reveal any information	Yes	It is assumed that the *DM-DDS* does reveal only as much information as necessary during the discovery process.
DM-SA-1: Mutual authentication of devices	Yes	The *DM-SCB* allows the establishment of secure channels.
DM-DM-1: Adding a device to virtual device	Yes	The *DM-VDM* provides the *If-VDM* to modify the virtual device composition. This can be either triggered via the local *DM-UI* or from another device that is part of the VD.
DM-DM-2: Removing a device from virtual device	Yes	c.f. DM-DM-1
DM-DM-3: Device identity	Yes	Each device within a virtual device can be identified by an device identifier (c.f. Figure 4.23.
DM-DC-1: Capturing of device characteristics	Yes	c.f. R5
DM-DC-2: Manual editing of device characteristics	Yes	The *DM-UI* allows to modify the device characteristics manually. It is possible to add device characteristics that cannot by collected automatically.
DM-DC-3: Selection of usage context	Yes	The *DM-UI* allows to modify the usage context of device by using the *DM-DI*.
DM-DC-4: Data Format for device characteristics	Yes	Figure 4.23 specifies a data model that captures required device characteristics.
DM-DA-1: Mechanism for data exchange	Yes	The *DM-VDM* exchanges information about devices and the virtual device with *DM-VDM* on remote devices. The *DM-VDAM* decides which information is provided to other devices.
SR-1: Mechanism for prevention of unauthorized device usage	Partially	It is assumed that every device provides a locking mechanism that prevents unauthorized usage of the device itself.
SR-2: Mechanism for removing devices from VD	Yes	c.f. DM-DM-2

Continued on next page

Table 4.21: Requirements addressed by the Device Manager

Requirement	Addr.	Explanation
SR-4: No dependency on single device	Yes	Each device that is part of a virtual device can operate on its own. Parts of the functionality depend on a single device. Assertion request and identity activation might depend on a single device, i.e. this functionality is not available, if the corresponding device is not available. If a previously discovered device becomes unavailable adequate time mechanisms are provided by the Device Management protocol.
SR-5: Authorization mechanism for modifying a VD	Yes	The *DM-VDAM* allows the specification of policies that restrict the modification of the virtual device. The *DM-VDM* requests policy decisions before modification of the virtual device take place.
SR-7: Mechanism to obtain information about VD composition	Yes	The *If-DI* allows to access information about the virtual device. The *DM-UI* visualizes the corresponding information.
SR-8: Logging of virtual device modification	Yes	The *DM-VDM* logs modifications of the virtual device.
SR-9: Confidentially of information about virtual device composition	Yes	The *DM-VDM* uses the *If-SS* to store virtual device information.
SR-18: Secure storage of security properties	Yes	The *DM-VDM* and the *DM-LDM* use the *If-SS* to securely store device characteristics.
SR-19: Logging of changed security properties	Yes	The *DM-VDM* and the *DM-LDM* use the *If-Log* to log changes regarding device characteristics.
SR-20: Storage of all security properties only on secure devices	Yes	The *DM-AM* decides which information about device characteristics is exchanged between devices by the *If-DM*.
SR-21: Secure device discovery	Yes	c.f. DM-DD-3
SR-22: Availability of device discovery	Yes	It is assumed that the device discovery does not depend on a single device belonging to the virtual device. It must be assumed that an individual device might instantly disappear.

Continued on next page

Table 4.21: Requirements addressed by the Device Manager

Requirement	Addr.	Explanation
SR-23: Encrypt device identifiers or avoid unique device identifiers	Yes	c.f. DM-DD-3
SR-25: Secure storage of device characteristics	Yes	c.f. SR-18
SR-27: Secure storage of usage context	Yes	c.f. SR-18
SR-28: Logging of usage context change	Yes	c.f. SR-19

4.5.6 Deployment Aspects

The functional architecture does not make assumptions on the actual deployment. Basically, it allows the realization of different deployment scenarios. For the deployment of functionality the following non-functional requirements have to be considered:

NF-5 – No dependence on a single device: The deployed system must not constrain the user in a way that he depends on a single device. On one hand, devices must remain independent within the bounds of their capabilities (e.g. a device that does not support SIM card based authentication, cannot use corresponding services). On the other hand, if several devices, which are part of a virtual device, are available they have to collaborate to enable the benefits resulting from multi-device IdM.

NF-4 – No degradation of security: The devices belonging to a virtual device collaborate to enable multi-device IdM. The collaboration has to be organized and coordinated by the most secure device which is currently available.

NF-1 – Data minimization principle: The devices belonging to a virtual device must not have the same view on data that is required to enable multi-device IdM. That means different devices of a virtual device must possess not more information about devices and identities than necessary. This increases the security and privacy of the user. An insecure device that gets easier compromised possesses less information, which results in less damage.

NF-2 – High usability: The virtual device provides a high usability, if the system limits the number of device switches for the user to perform activities, like authentication. That means it is assumed that it is beneficial, if there are devices with which the user interacts more often than with others. More details on this aspect with respect to multi-device IdM are elaborated in Section 5.5.

The non-functional requirement on performance (NF-3) is less important for the deployment, if the following assumptions hold: (1) Sufficient network transmission capacity for collaboration between the devices. (2) Powerful devices to perform activities regarding virtual device organization and multi-device IdM.

Non-functional requirements exclude both extreme deployment scenarios. A centralized deployment that depends on the existence of a single device is excluded by NF-5. A fully decentralized deployment, which means that all devices are equal with respect to functionality and responsibilities, is excluded by NF-1 and NF-4.

A deployment scenario that considers NF1, NF-2, NF4, and NF5 is a flexible master-device concept. That means one device out of the available devices that belongs to the virtual device is selected as the master device. The master device has a complete view on the user's identities and on the complete device characteristics of individual devices. Devices that are not selected as a master device depend on the coordination by the master device. The device that has the role of the master device changes according to the availability of devices. For example, if the master device becomes unavailable, e.g. if it is switched off, another device takes over the role of the master device. This fulfills NF-5, i.e. the virtual device does not depend on the existence of a single device. Moreover, NF-1 is fulfilled, because different devices have a different view on the device characteristics and on the user's identities. If the master device selection considers the security level of the devices as an aspect, NF-4 would be fulfilled. Moreover, the master device serves as main device for user activities and fulfills NF-2.

5 Algorithms, Mechanisms and Protocols for Multi-device Identity Management

This chapter details the functional architecture designed in Chapter 4 by providing different views on the behavior of the system. Figure 5.1 outlines the structure of this chapter.

Section 5.1 makes up the foundation of the chapter with the introduction of the virtual device lifecycle and of the identity lifecycle. The lifecycles provide an overview on the relationships of the subsequent sections. A state diagram and an activity diagram glue together the algorithms, mechanisms and protocols that are introduced in Section 5.2, Section 5.3, and Section 5.4. For improved security and usability it is required to restrict the usage of identities and provide a corresponding ranking of identities for the user to perform identity selection. Section 5.2 details the specified identity filtering mechanisms. Section 5.3 introduces the designed protocols to make identities usable across user devices, i.e. the multi-device IdM key concept. Section 5.2 and Section 5.3 rely on the virtual device concept. Section 5.4 specifies this concept with focus on the required mechanisms to provide multi-device IdM. The virtual device concept and the multi-device IdM concept provide various degrees of freedom with respect to the placement of functionality. Section 5.5 elaborates these degrees of freedom and provides an algorithm for a reasonable placement.

5.1 Overview on Lifecycles

This chapter shows the relationship between the virtual device concept and the multi-device IdM concept. The lifecycles of virtual devices and the lifecycles of identities illustrate the

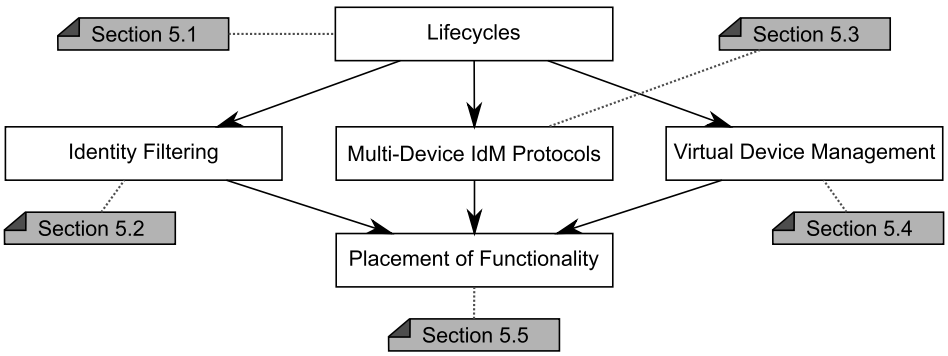


Figure 5.1: Chapter Outline

dependencies and serve as basis for the introduction of the detailed mechanisms in subsequent sections. The focus resides on the novelties of the multi-device IdM concept in relation to the virtual device concept. For aspects that are not in focus, corresponding references are provided.

5.1.1 Lifecycle of Virtual Device

The lifecycle of a virtual device consists of four states as shown in Figure 5.2. This section briefly introduces each state and provides references to sections that contain details.

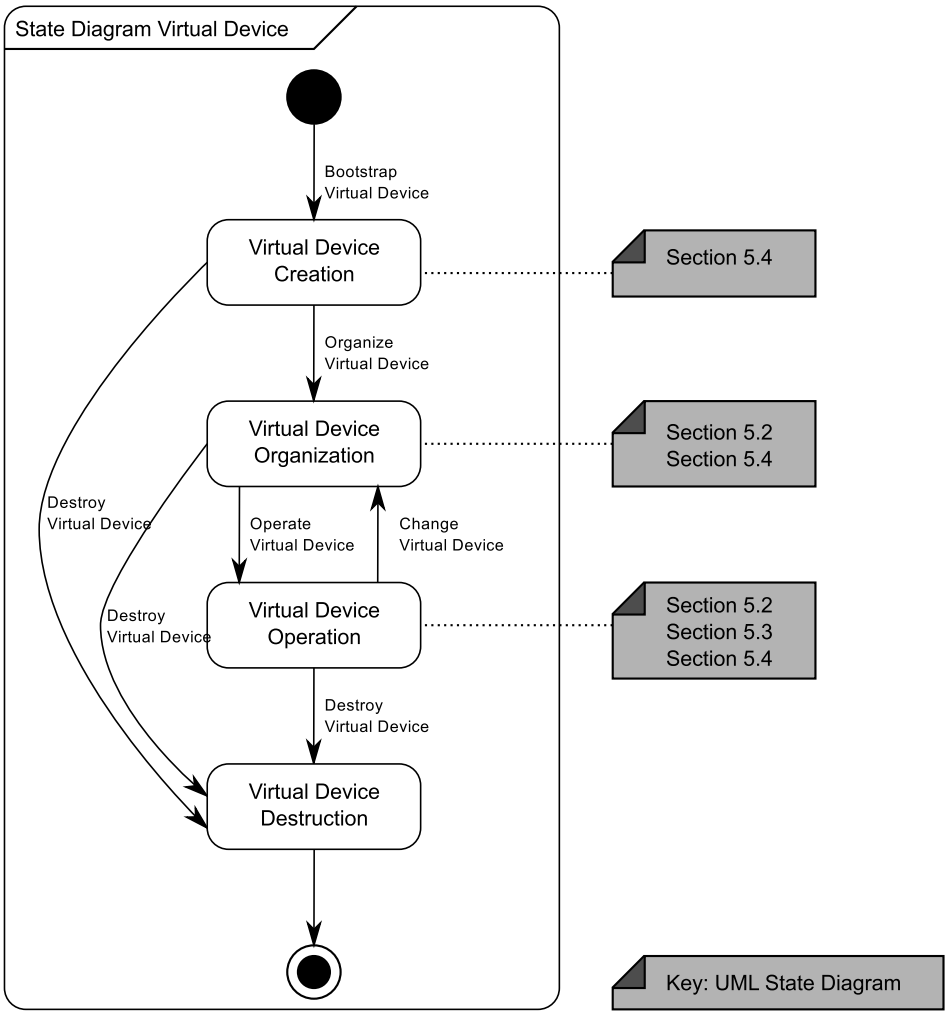


Figure 5.2: Virtual Device Lifecycle

Virtual Device Creation: The creation of a virtual device requires at least one device and includes the installation of the required middleware and the bootstrapping of security functionality. Since the virtual device creation is not the core focus of the thesis it is not covered. The only exception is made for the bootstrapping of the security infrastructure that is covered in Section 5.4.2.

Virtual Device Organization: A virtual device (re)organizes itself for two reasons. First, devices are added or removed from the virtual device. Since adding and removing has an

impact on the security associations between devices, the process of adding a device to the virtual device is detailed in Section 5.4. Second, devices that already belong to the virtual device might suddenly appear or disappear (e.g. one device might run out of battery power). In both cases a series of activities related to the management of identities as well as related to the organization of the virtual device have to be triggered. Section 5.2 and Section 5.3 detail the corresponding activities regarding multi-device IdM. These activities comprise the first phase of the two-phase identity filtering process and the protocols for the exchange of identity information.

Section 5.4.1 specifies activities regarding the virtual device organization itself. Among the activities is the process of Master Device selection, which is required to fulfill non-functional requirements (→ Section 4.5.6). The master device is one device within the virtual device composition that is responsible for the management of the virtual device itself (→ Section 5.4.1 for more details). All activities that exchange information depend on the establishment of secure channels between devices (→ Section 5.4.2 for more details). After all activities have finished, the transition to the Virtual Device Operation state is triggered.

Virtual Device Operation: In this state, the virtual device is stable. That means available devices do not appear or disappear. Based on a stable virtual device, the user consumes services with selected identities. Section 5.1.2 provides additional details on the identity selection process. If the make-up of the virtual devices changes, the Virtual Device Organization state is entered.

Virtual Device Destruction: In this state, all activities to destroy the virtual device take place. This includes the deletion of all data that has been collected or created during the other states of the virtual device lifecycle. This state can be reached from all other states. This thesis does not provide additional details on this state and the corresponding activities.

5.1.2 Identity Lifecycle

As introduced in Section 3.2.2.1 and shown in Figure 5.3, the lifecycle of an identity distinguishes three different states: Identity Creation, Identity Existence and Identity Destruction. Details on the creation and destruction phase are not in scope of this thesis. It is assumed that a user already has at least one identity.

Within the Identity Existence state different activities take place. At first, the user selects a service and requests this service from the SP. The SP provides information about the requirements of using this service with respect to the required attributes and the required authentication method. This information serves as input for the second phase of the identity filtering process (→ Section 5.2) that creates a list of ranked identities of which the user selects one. Section 5.3.1 introduces a set of protocols that exchange different kinds of information on user's identities. Based on the exchanged information it is possible to check whether an identity is active, i.e. whether an IdP session exists for that identity, or not. If no IdP session exists, the user can activate such a session within the virtual device by means of the Identity Activation Protocol (→ Section 5.3.2). Afterwards, the Assertion Request Protocol (→ Section 5.3.3) allows the retrieval of the required SP assertions to authenticate against the SP and start with the service consumption.

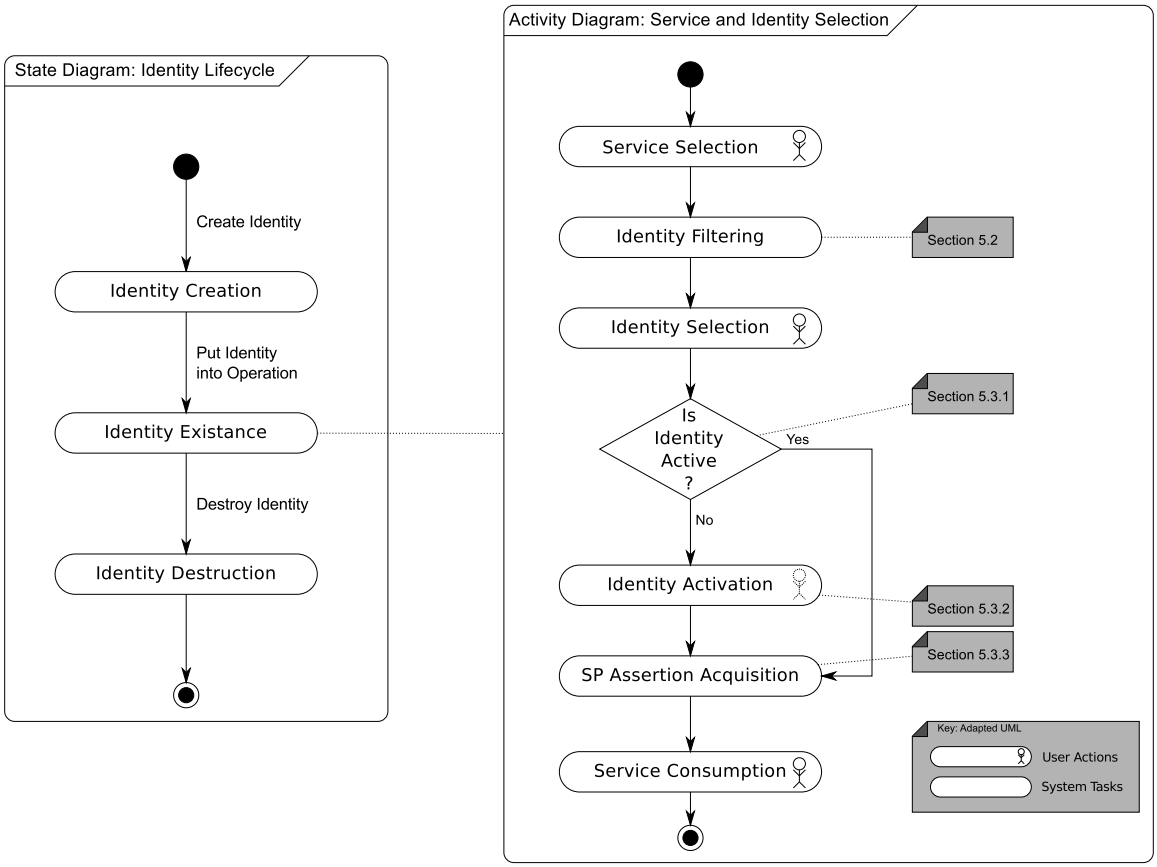


Figure 5.3: Identity Lifecycle [1]

5.2 Identity Filtering

The goal of the identity filtering process is to support the user with the identity selection as introduced in Section 3.2.2.5. That means to consume a service the user has to select one of his identities. The process of identity selection is considered to be difficult for the user. Therefore, the identity filtering process restricts the choice the user has and ranks identities according to defined policies. For example, the ranking prioritizes identities for which already an IdP session exists.

The identity filtering process consists of two phases that take place at different points in time as shown in Figure 5.4. Phase 1 is the Prefiltering phase, which takes place whenever the virtual device is reorganized (i.e. Virtual Device Organization). Section 5.2.1 details phase 1. Phase 2, the Final Filtering phase, takes place when the user selects an actual service, i.e. the requirements of the service provider are taken into account. Section 5.2.2 details phase 2. The following considerations do not take the actual deployment into account, i.e. where the identity filtering process is executed and how the described data is gathered.

[1] The activity “Identity Filtering” is actually the step “Final Filtering” in Figure 5.4

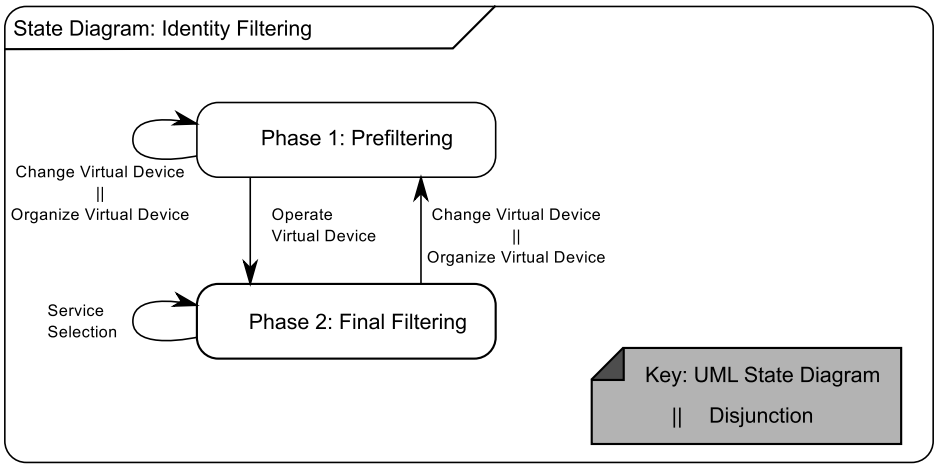


Figure 5.4: Overview on Identity Filtering

5.2.1 Phase 1: Prefiltering

The Prefiltering phase combines metadata about devices and metadata about identities with policies (→ Figure 5.5) and classifies user's identities into three categories:

- Directly usable identities (DI): This category comprises identities that can basically be used on a device without interaction with other devices. That means the identity and the device fulfill all requirements that are imposed by the policies.
- Indirectly usable identities (II): Identities that can only be used with the support of another device that takes over the authentication against the IdP and provides the required assertions.
- Unusable identities (UI): Identities that cannot be used on a device are in this category. Reasons why an identity cannot be used on a particular device are among others an insufficient security level or an inappropriate usage context.

The set of indirectly usable identities is a superset of the directly usable identities. That means if an identity is directly usable on a device, it is also indirectly usable with the support of another device. However, if an identity is indirectly usable, this does not mean that it is directly usable on a device.

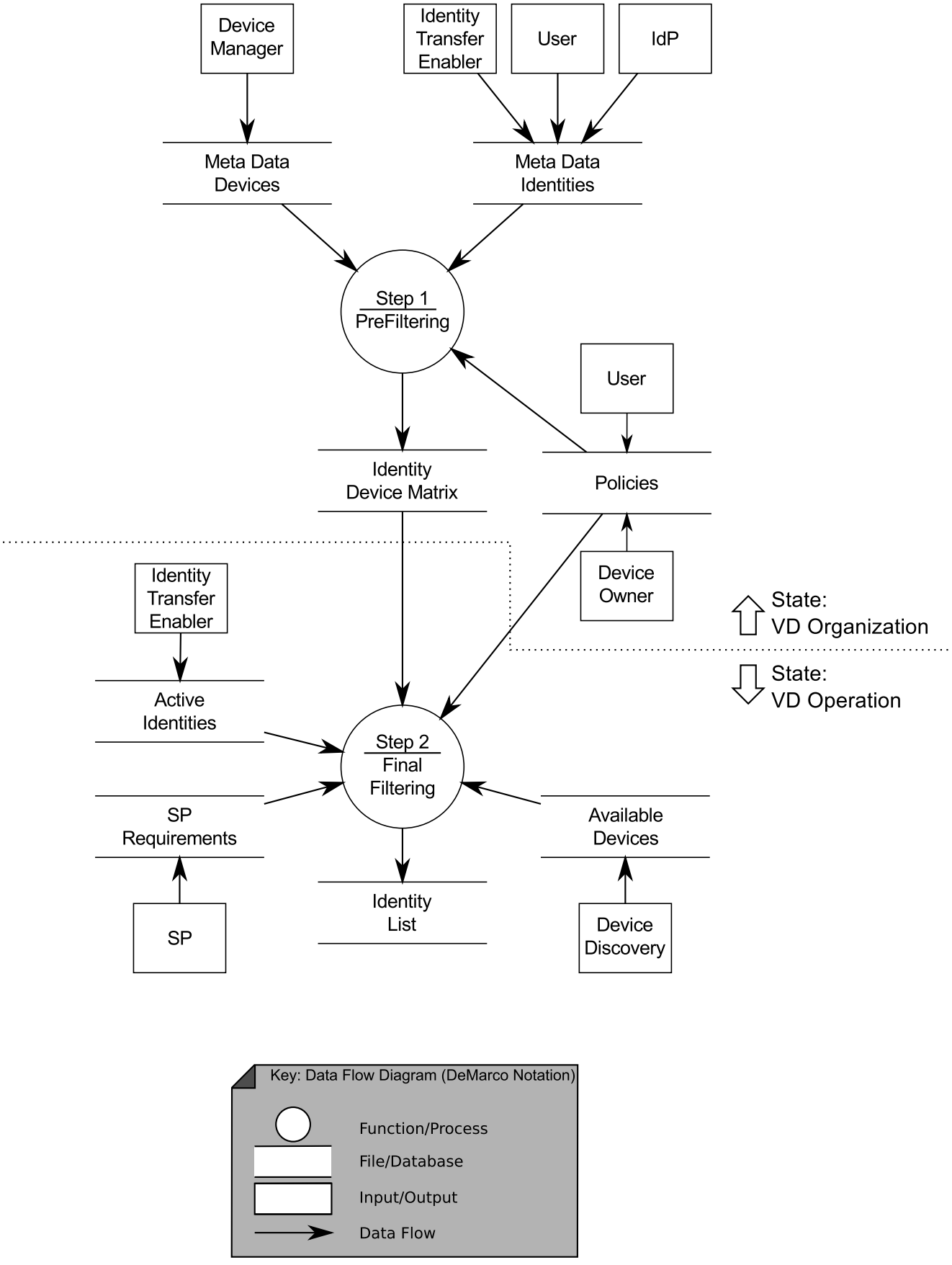


Figure 5.5: Data Flow Diagram of the Identity Filtering Process

The result of the Prefiltering phase is, as shown in Figure 5.5, the so called identity device matrix M. An entry $m_{i,j}$ describes the possibility to use identity i on device j.

$$M = \begin{pmatrix} m_{1,1} & m_{1,2} & \dots \\ m_{2,1} & \ddots & \\ \vdots & & \end{pmatrix} \tag{5.1}$$

$m_{i,j}$ is hereby a set of values $x_y^{i,j}$, e.g. $m_{i,j} = \{x_1^{i,j}, \dots, x_{k_{i,j}}^{i,j}\}$, with the following meaning. For example $m_{3,5} = \{DI, II_1, II_4\}$ describes the possibilities to use identity 3 on device 5. Identity 3 can be either used, if an IdP session is established on device 5 or by corresponding activation on device 1 or 4.

$$x_y^{i,j} = \begin{cases} DI & \rightarrow \text{Identity } i \text{ directly usable on device } j \\ II_z & \rightarrow \text{Identity } i \text{ indirectly usable on device } j \\ & \quad \text{with the support of device z} \end{cases} \tag{5.2}$$

Algorithm 1 (→ Page 115)[2] creates M by applying policies. Figure 5.6 specifies the data model for the policies in terms of filter rules. Two categories of filtering rules are distinguished:

- Activation Rules: Activation rules determine whether an identity can be activated on a device or not. If an identity can be activated an IdP session can be established on a device.
- Usability Rules: Usability rules determine whether an identity can be used on a device or not. If an identity can be used on a device, a SP session can be established. The corresponding IdP session does not have to reside on same device.

A *Filter Rule* (→ Figure 5.6) provides a method *isFilterMatching* that returns *true*, if the identity and the device match the criteria specified by the filter rule. Otherwise, the identity and device combination does not match, i.e. the identity is not usable/activatable on the given device for the specified usage contexts.

Figure 5.6 distinguishes three different specializations of filter rules.

- Usage Context Filter: A usage context filter compares the usage contexts of identities and devices. If the usage context is equal, the identity can be used/activated on the given device. An identity as well as a device can have more than one usage context, i.e. usage context sets S_{UC}. If the intersection of both usage context sets is not empty (→ Eq. (5.3)), the identity can be used/activated on the given device.

$$S_{UC,\text{DeviceId}} \cap S_{UC,\text{IdentityId}} \neq \{\} \tag{5.3}$$

[2] Algorithm 1 uses a pseudo code notation. It provides a procedure "Establish Device Identity Matrix" that requires six parameters. The ▷ symbol indicates a comment.

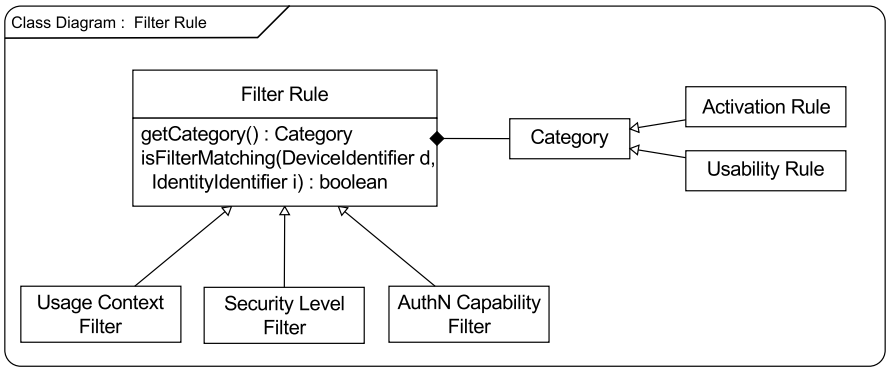


Figure 5.6: Data Model for Filtering Rules

- Security Level Filter: A security level filter checks whether the security level of the device is greater or equal as the security level specified by the identity metadata. Section 5.4.2.4 provides details on the determination of the security level.

- Authentication Capability Filter: An authentication capability filter checks whether the provided device supports at least one authentication method that is supported by the identity, i.e. the IdP supports the authentication method. S_{AuthN} is the set of authentication capabilities. If Eq. (5.4) holds, the identity can be used on the given device.

$$S_{AuthN,\text{DeviceId}} \cap S_{AuthN,\text{IdentityId}} \neq \{\} \tag{5.4}$$

Algorithm 1 creates M in a three-step procedure. In step 1, the necessary data structures are created. In step 2, Algorithm 1 iterates over all devices D and over all identities I of the user. For each device/identity combination, the two categories of filtering rules are evaluated. Depending of the matching result, the device/identity combination is added either to the list of activatable identities AI or to the list of usable identities UI. Finally, step 3 creates M by combining the intermediate results contained in AI and UI.

5.2.2 Phase 2: Final Filtering

Whenever the user selects a service, a list of identities has to be presented to the user containing identities that can be used with the service. From this list, which is ordered according to specified ranking rules, the user selects one identity to be used with the service.

The Final Filtering phase creates such a list based on the information depicted in Figure 5.5:

- Identity-Device Matrix M

- SP Requirements: The SP specifies requirements regarding the required user attributes and the required authentication method.

- Available Devices: The available devices limit the number of usable/activatable identities. If a device is not available and the usage/activation of an identity depends on that device, the identity cannot be used.

Algorithm 1 Prefiltering Algorithm

procedure ESTABLISH DEVICE IDENTITY MATRIX(D, I, A, U, AI, UI)
▷ D: Set of devices belonging to Virtual Device
▷ I: Set of user's identities
▷ A: Set of activation rules
▷ U: Set of usability rules
▷ Phase 1: Initialization
$init(AI)$ ▷ AI: Activatable identities per Device
$init(UI)$ ▷ UI: Usable identities per Device
▷ Phase 2: Matching
for all $d \in D$ **do** ▷ Iterate over devices
for all $i \in I$ **do** ▷ Iterate over identities
▷ Check activatable identities
$isMatching \leftarrow true$
for all $a \in A$ **do**
if $isFilterMatching(d, i, a) \neq match$ **then**
$isMatching \leftarrow false$
end if
end for
if $isMatching = true$ **then**
$add(d, i, AI)$
end if
▷ Check usable identities
$isMatching \leftarrow true$
for all $u \in U$ **do**
if $isFilterMatching(d, i, u) \neq match$ **then**
$isMatching \leftarrow false$
end if
end for
if $isMatching = true$ **then**
$add(d, i, UI)$
end if
end for
end for
▷ Phase 3: Combination
$constructDeviceIdentityMatrix(AI, UI)$ ▷ Creates M
end procedure

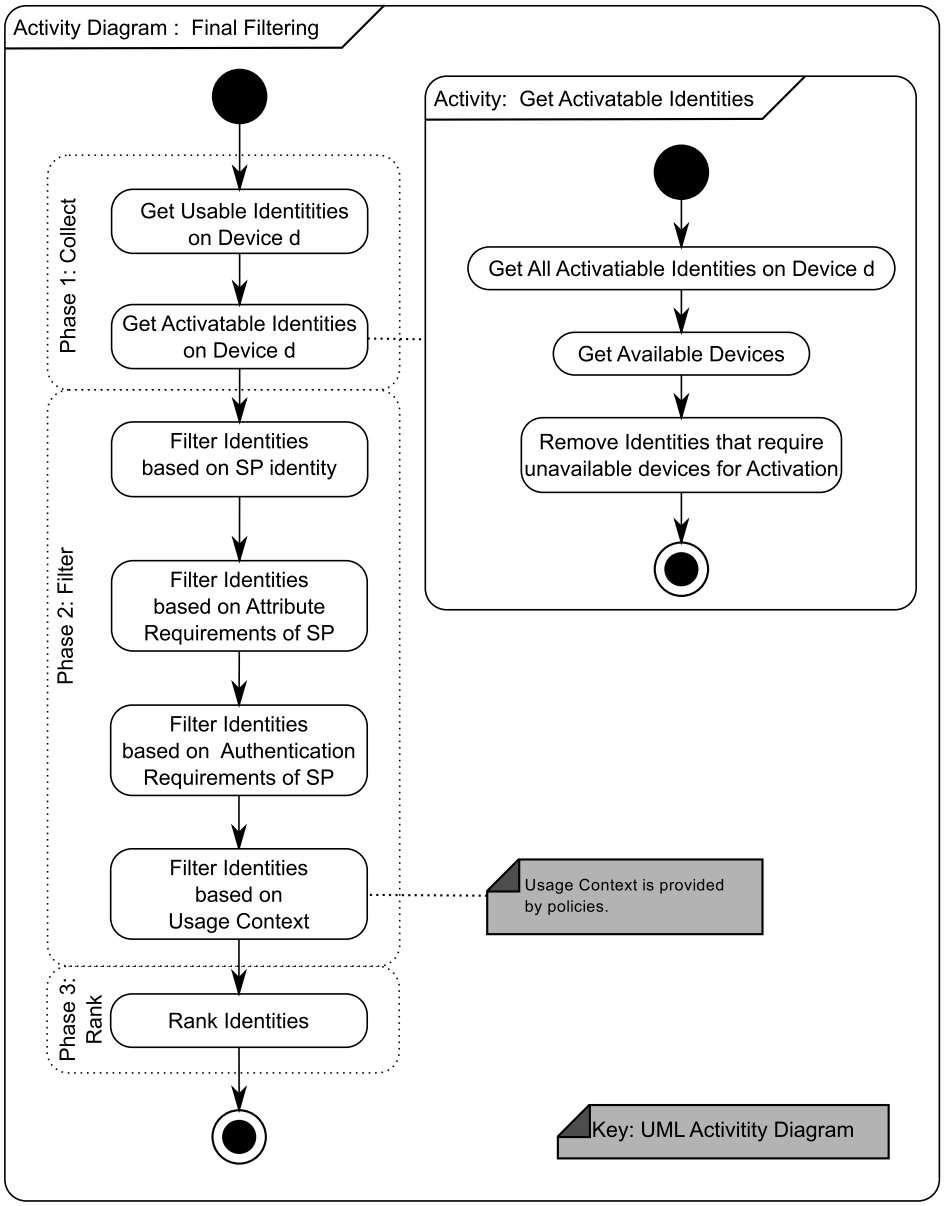


Figure 5.7: Activity Diagram for Final Filtering Algorithm

- Policies: The user or the device owner[3] might specify additional rules regarding the usability of an identity with respect to the identity of the SP. For example a rule might specify that services provided by a VoD X have always a usage context of “Private”.
- Active Identities: If an identity is already active, there is no need to re-authenticate against an IdP resulting in a higher usability. Therefore, an already active identity should be ranked higher in the list of selectable identities.

Figure 5.7 shows an activity diagram of the Final Filtering phase. It consists of three steps. The first step collects data on usable and activatable identities on the given device. This includes checking the availability of devices.

[3]The device owner might be for example the employer of the user, which wants to restrict the services the user can consume.

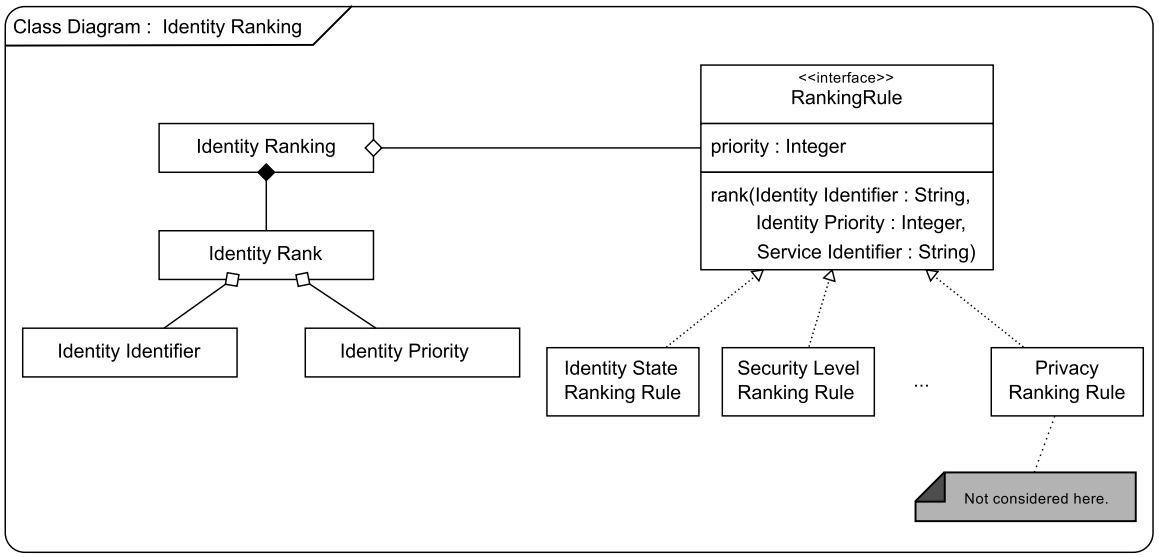


Figure 5.8: Class Diagram of Identity Ranking

The second step filters the identities based on the identity of the SP, the requirements of the SP and on the usage context, which is specified by means of policies. As a result, an unordered list of selectable identities exists. Each identity from the list can basically be used to consume the requested service.

Step three ranks the unordered list of identities according to ranking rules. Figure 5.8 specifies the corresponding data model. Each identity has a rank within the identity ranking list that specifies the priority to use this identity for the given service. The priority of an identity is modified by ranking rules ($\rightarrow$ *RankingRule*), which increase or decrease the priority. Increasing the priority value means that an identity becomes more prior. Examples for ranking rules are the *IdentityStateRankingRule* and the *SecurityLevelRankingRule*. The *IdentityStateRankingRule* evaluates the current state of an identity. If an identity is already active, the priority is increased. The *SecurityLevelRankingRule* takes the device on which the identity depends for authentication into account. The higher the security level, the higher the priority. The given ranking framework can be easily extended with ranking rules that prioritize identities according to potential privacy violations. Privacy considerations are not in scope of this thesis, a potential approach is outlined in [Neu09].

5.2.3 Addressed Requirements

The identity filtering process addresses the requirements[4] enumerated in Table 5.1.

Table 5.1: Requirements addressed by Identity Filtering Mechanism

Requirement	Addressed	Explanation
R2: Task distribution	Yes	The Identity Filtering process decides which identities can be used on which devices and which devices are able to perform the authentication.
R3: Remote activation	Yes	The Prefiltering as well as the Final Filtering take the security level and the available and required authentication methods into account to decide which identity can be used on which device with which service.
IdM-IM-7: List of selectable identities	Yes	The result of the Final Filtering step is a list of selectable identities (→ Figure 5.4).
IdM-IM-8: Priorities for selectable identities	Yes	The identity list, which is returned by the Final Filtering, is ranked according to specified criteria (→ Figure 5.8).
SR-4: No dependency on single device	Yes	The identity filtering itself can run on all devices. In addition, the process considers all devices that can be used for identity activation and assertion retrieval.

5.3 Protocols for Multi-Device Identity Management

This section specifies the protocols to enable Multi-device IdM. It complements the previous section on identity filtering with protocols to exchange metadata about the user's identities and protocols to activate and use the identity. The previous section on identity filtering did not specify details how information about identities is exchanged and how the selected identities can be used. Identity information is exchanged by means of the Identity Information Exchange Protocol (IIEP) (→ Section 5.3.1). Afterwards, the Identity Activation Protocol (IAP) (→ Section 5.3.2) specifies how identities can be activated on a remote device in order to retrieve the required SP assertions by means of the Assertion Request Protocol (ARP) (→ Section 5.3.3). Section 5.3.4 concludes this section with example scenarios that define the collaboration of the introduced protocols.

[4]For the detailed description of requirements, it is referred to Chapter 4.

5.3.1 Identity Information Exchange Protocol

The IIEP is responsible for the exchange of information about identities. This includes metadata on identities and the current state of the identity (e.g. Active). The IIEP realizes the *If-II* part of the *If-IdM* (→ Section 4.5.2). Figure 4.18 specified the underlying data model.

5.3.1.1 Message Flow

Figure 5.9 shows the protocol primitives for the exchange of identity metadata. The master device, which is one dedicated device within the virtual device composition with coordination responsibility (for more details regarding the master device, it is referred to Section 5.4) triggers the protocol primitives, whenever there are changes in the virtual device. That means the state Virtual Device Organization defined in Figure 5.2 is reached and the master collects information from all other devices. Moreover, the IIEP can be triggered upon changes regarding identities.

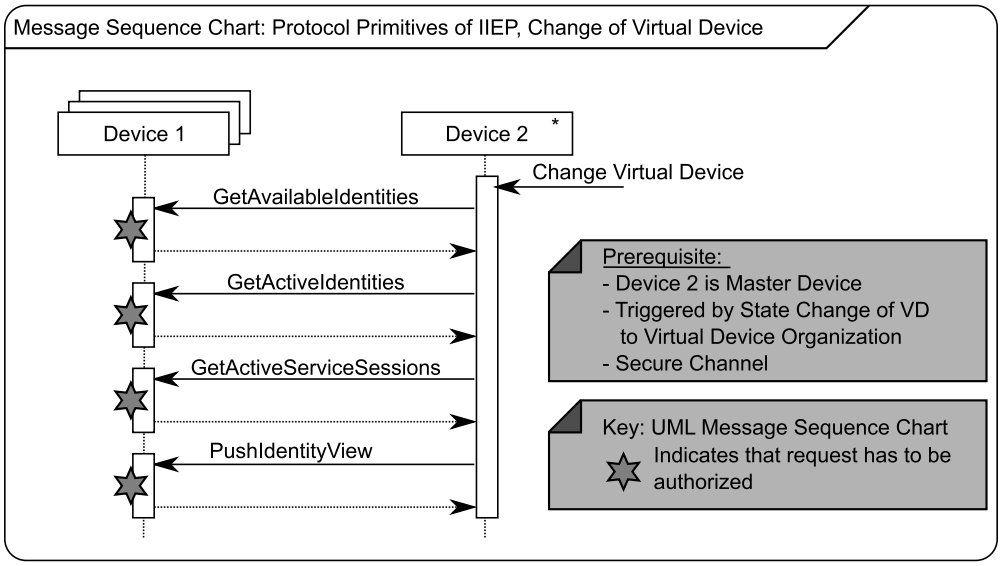
Figure 5.9(a) distinguishes four message calls:

- *GetAvailableIdentities()*: Returns all identities that are known on the other device. This allows the master device to obtain a unified view on all identities that the user has and provides the possibility to bootstrap[5] new identities on every device.
- *GetActiveIdentities()*: Returns all active identities, i. e. whether an IdP session for an identity exists or not. This primitive is a prerequisite to enable the retrieval of SP assertions without the need to reestablish an IdP session. In addition, it enables the directed tear down of an IdP session to counter security problems.
- *GetActiveServiceSession()*: Returns all active service sessions. This primitive enables the directed tear down of a service session.
- *PushIdentityView()*: Pushes the identity view that the device should have onto the device. The identity view corresponds to one row within the identity-device matrix M.

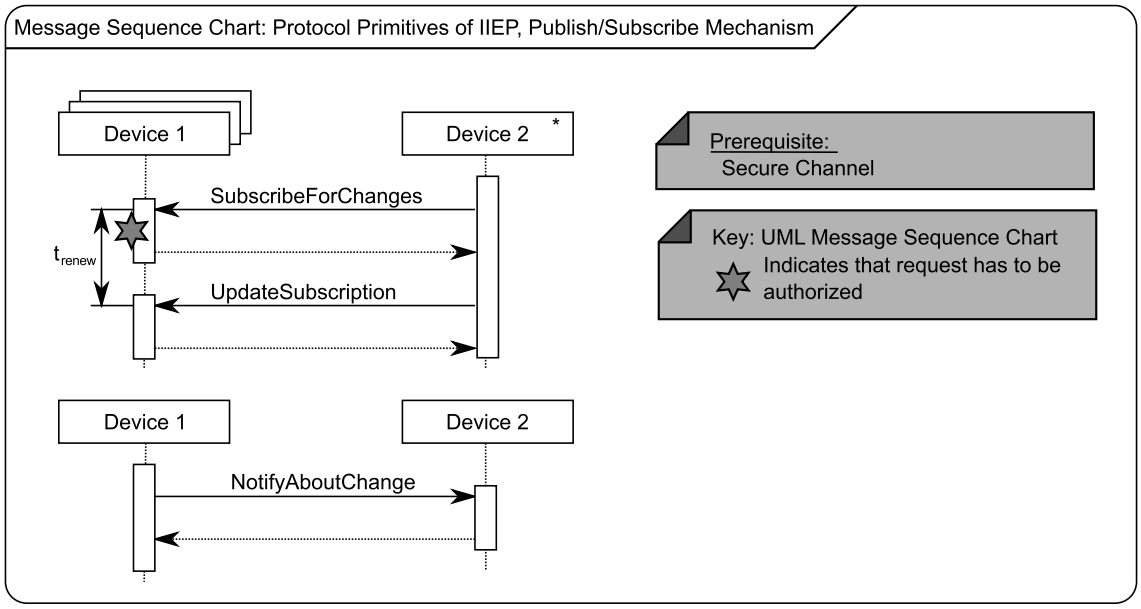
Since the requested information is security-sensitive it has to be transferred on top of a secure channel. In addition, the providing device has to authorize the request. The authorization procedure is indicated in Figure 5.9(a) with the star symbol and performed by the *ITE-AM* (→ Section 4.5.3).

Since metadata on identities and the state of identities can be influenced on every device of the virtual device composition, a notification mechanism based on the publish-subscribe pattern [GHJV94] is required. One device can subscribe with another device providing a list of interests as shown in Figure 5.9(b). Such a subscription has to be renewed after t_{renew} to deal with disappearing devices. Upon a change of the data corresponding to the registered interest, the subscribing device is notified. The following primitives exist:

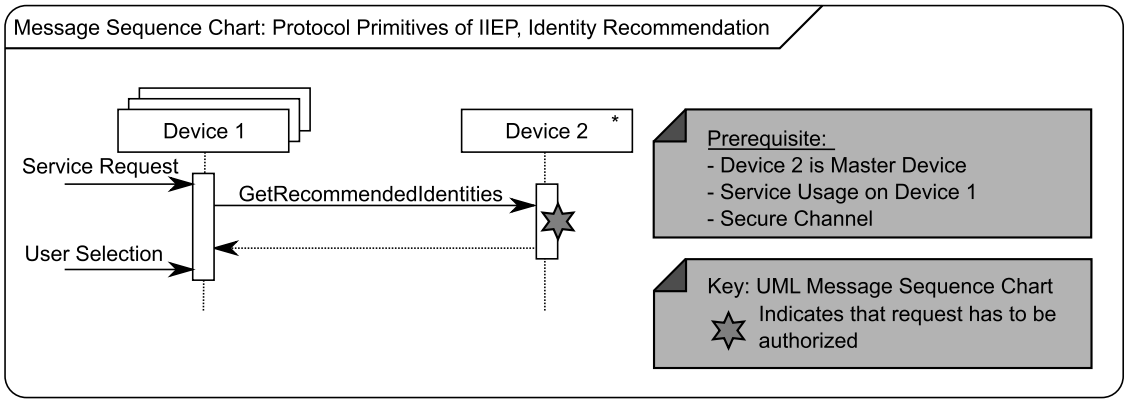
[5] Bootstrapping means the creation and first usage of an identity.



(a) Change of Virtual Device



(b) Publish Subscribe



(c) Identity Recommendation

Figure 5.9: Message Sequence Charts: Identity Information Exchange Protocol

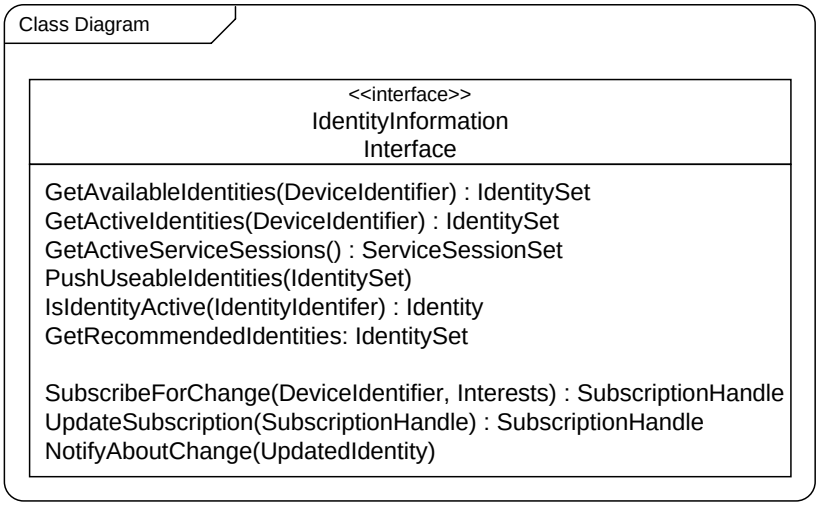


Figure 5.10: Interface Description of IIEP

- *SubscribeForChanges()*: Allows to subscribe with another device to get notified when changes regarding the subscribed topics occur. A subscription is critical and has to be authorized.

- *UpdateSubscription()*: Since a subscription adheres to the soft state principle [RM99], it is required to update the subscription periodically.

- *NotifyAboutChange()*: Used by the subscription-issuing device to inform the subscribers about changes.

Finally, the IIEP provides a primitive *GetRecommendedIdentities* to request identity recommendations from the master device (→ Figure 5.9(c)). This primitive is a potential consequence of a service request that is triggered by the user. Identity recommendations correspond to the ranked identity list that is created by phase 2 of the identity filtering process.

5.3.1.2 Interface Description

Figure 5.10 details the protocol primitives of the above introduced message flows. The *If-II* is part of the *If-IdM* and is used as inter-device as well as intra-device interface (→ Figure 4.14 and Figure 4.16). The class diagram provides details on the methods provided by the *IdentityInformationInterface*. The interface description is transformed into actual messages by the process described in Section B.5.

5.3.2 Identity Activation Protocol

The IAP is responsible for the activation and deactivation of identities and for the forced teardown of IdP sessions on remote devices. The IAP realizes the *Identity Activation Interface (If-IA)* (→ Section 4.5.2) and based on the data model specified in Figure 4.18.

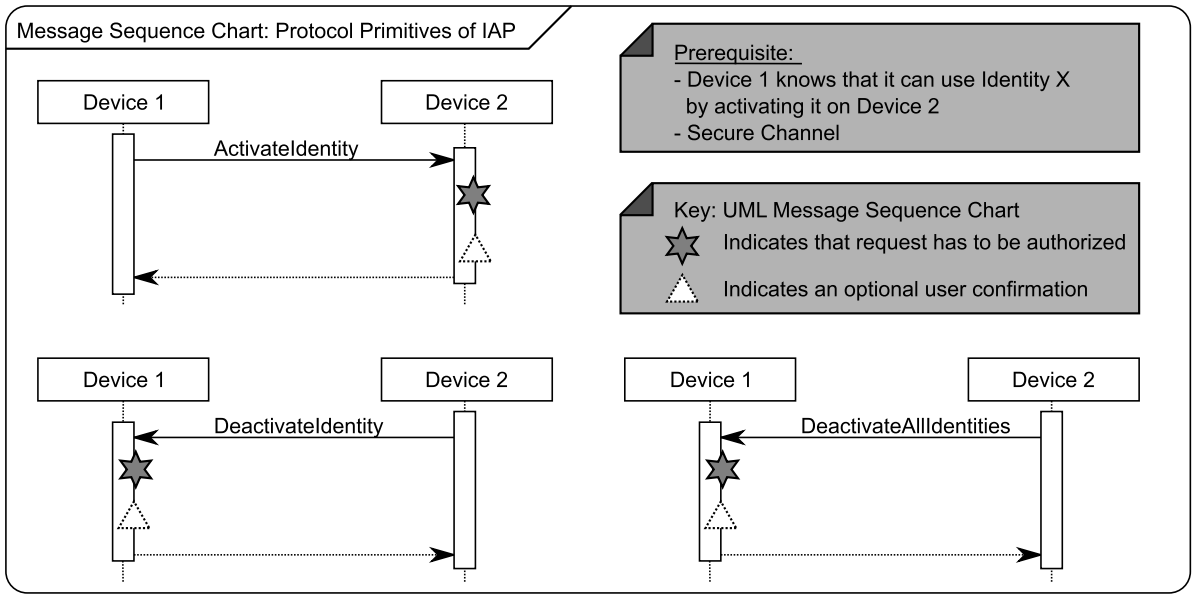


Figure 5.11: Message Sequence Chart: Identity Activation Protocol

5.3.2.1 Message Flow

Figure 5.11 shows the protocol primitives of the IAP. It distinguishes the following primitives:

- *ActivateIdentity()*: Activates the given identity on a remote device and allows one device to make use of the authentication capabilities of another device.

- *DeactivateIdentity()*: Deactivates the indicated IdP session on the remote device. The boolean argument *SingleLogOut* determines whether all service sessions on top of the IdP sessions should be torn down.

- *DeactivateAllIdentities()*: Deactivates all IdP sessions on the remote device. The purpose of this method is to provide an additional security mechanism that allows the explicit tear down of all IdP session. Afterwards it is required to establish a new IdP session with the corresponding authentication to consume a service.

Since all messages are security-sensitive, several measures are required. First, all messages are transported on top of a secure channel, which provides mutual authentication, confidentiality and integrity protection. Second, the requested device has to authorize the request in order to avoid requests from unauthorized devices[6]. The requesting device, the requested identity, the intended usage context and the number of already received identity activation requests serve as basis for the authorization decision. Third, the user might manually confirm the request. User confirmation is optional and subject to corresponding configuration.

[6]The authorization is realized by the *ITE-AM* in Section 4.5.3.

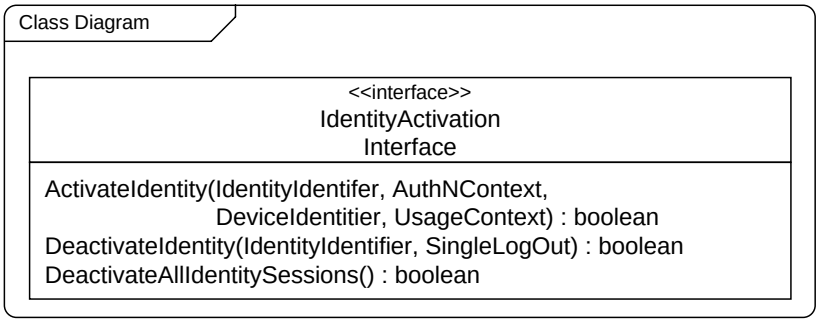


Figure 5.12: Interface Description of IAP

5.3.2.2 Interface Description

Figure 5.12 details the protocol primitives of the above introduced message flows. The *Identity-Activation Interface* (*If-IA*) is part of the *If-IdM* and is used as inter-device as well as intra-device interface (→ Figure 4.14 and Figure 4.16). The class diagram provides details on the methods provided by the *IdentityActivationInterface*.

5.3.3 Assertion Request Protocol

The ARP is responsible for the retrieval of assertions from remote devices. Given the assumption that the corresponding identity is already active on the remote device, another device can request an SP assertion to consume an intended service. In addition, it allows the tear down of an existing service sessions.

5.3.3.1 Message Flow

Figure 5.13 shows the protocol primitives for the retrieval of assertions and for the tear down of service sessions.

- *RequestAssertion()*: Used to request a SP assertion from a remote device. The requester provides an *AuthNContext* that describes the required authentication methods and allows the requestee to decide whether the existing IdP session fits the requirements.
- *DeactivateServiceSession()*: Used to tear down a particular service session. Allows the user to selectively tear down service sessions on remote devices. It is designed to enhance security and usability, because a directed tear down of a service session is possible, e.g. a forgotten service session can be closed from remote.
- *DeactivateAllServiceSessions()*: Used to tear down all service sessions on a remote device. This method is designed as security and as usability feature (c.f. *DeactiveService-Session()*).
- *RequestUserConfirmation()*: Allows to request the user confirmation for various actions, e.g. the *RequestAssertion()*. The device from which the user confirmation is requested is

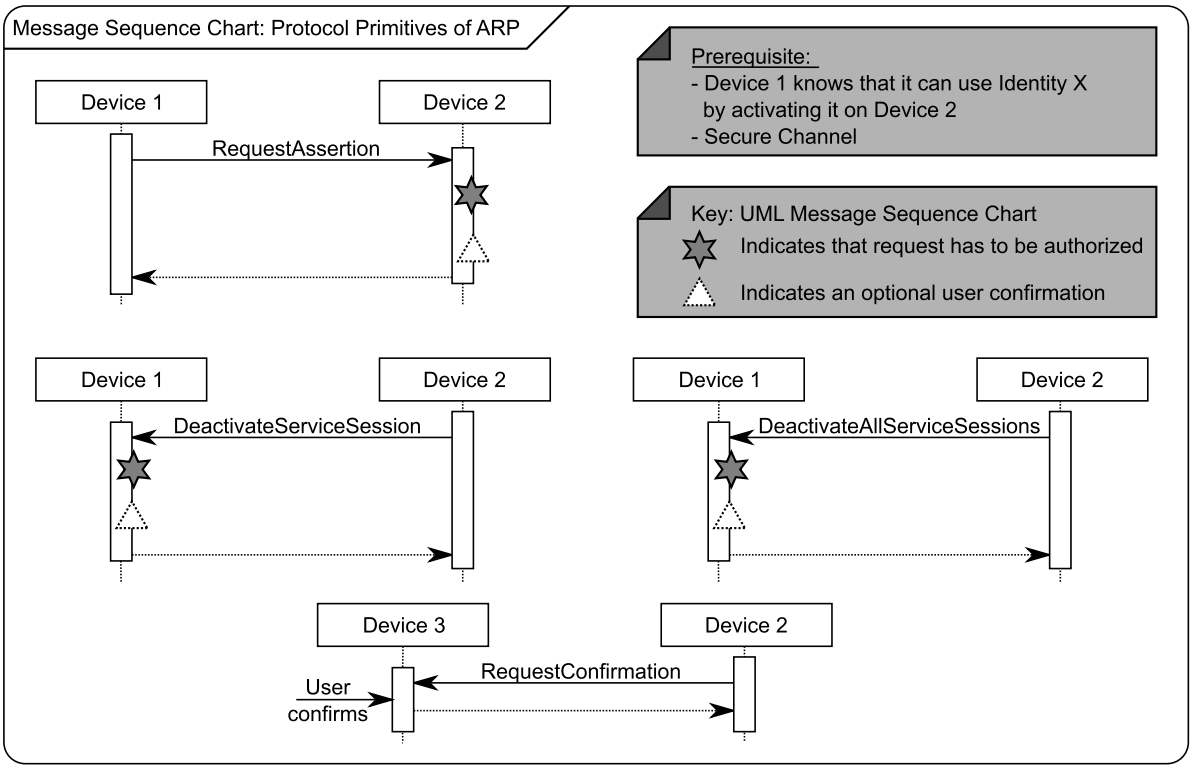


Figure 5.13: Message Sequence Chart: Assertion Request Protocol

not necessarily the requesting or requested device of *RequestAssertion()*, it can be a 3rd device. This method represents a security feature, because it allows to explicity involve the user in actions that would otherwise automatically take place.

For the messages of the ARP, the same security requirements as for the IAP hold (→ Section 5.3.1).

5.3.3.2 Interface Description

Figure 5.14 details the protocol primitives of the above introduced message flows. The *AssertionRequest Interface (If-AR)* is part of the *If-IdM* and is used as inter-device as well as intra-device interface (→ Figure 4.14 and Figure 4.16). The class diagram provides details on the methods provided by the *AssertionRequestInterface*.

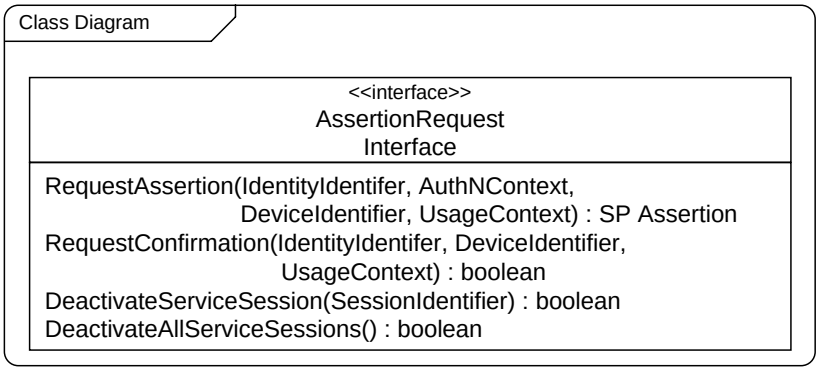


Figure 5.14: Interface Description of ARP

5.3.4 Scenarios

This section illustrates the collaboration of the three protocols introduced above. Two scenarios serve as examples for the collaboration of the user, the user's devices, i.e. the virtual device, the SP and the IdP. Both scenarios assume that the user has two devices that are part of a virtual device.

5.3.4.1 Scenario 1: Assertion Retrieval

Goal: Scenario 1 (→ Figure 5.15) focuses on the exchange of identity metadata and state information between devices and on the retrieval of an assertion from a remote device.

Prerequisites:

- Successful discovery: It is assumed that Device 1 and Device 2 have just discovered each other and that Device 2 became the master device.
- Functionality distribution: A dedicated distribution of functionality among Device 1 and Device 2 is assumed. That means the Prefiltering takes place on Device 2, whereas Device 1 is able to perform the Final Filtering.
- Secure Channels: For all interactions between the different parties, secure channels between the parties are assumed.
- No Failures: It is assumed that all interactions are successful. Therefore, there is no need within these example scenarios to check for errors and react correspondingly.

Stepwise explanation:

- Step (1): The discovery triggers Device 2 to request identity information from Device 1 by means of the IIEP.
- Step (2): After Device 2 has obtained all identity information, it triggers the Prefiltering phase (→ Section 5.2) and updates the identity device matrix M. Afterwards, the IIEP pushes the relevant view for Device 1 from Device 2 to Device 1.
- Step (3): The user establishes an IdP session on Device 2 by authenticating against the IdP for identity 'X'. The reason for authenticating against the IdP is neglected here.

- Step (4): The information about the established IdP session on Device 2 is pushed to Device 1. With this information Device 1 can prioritize identity 'X' during the Final Filtering phase.
- Step (5): The user intends to establish a service session. The SP responds with the request to authenticate and delivers his requirements regarding authentication methods and required user attributes.
- Step (6): The Final Filtering phase evaluates the requirements of the SP and creates a ranked list of possible identities. The user selects identity 'X'.
- Step (7): With the knowledge that identity 'X' is already active on Device 2, the *RequestAssertion()* method of the ARP is called in order to obtain a SP assertion. Before Device 2 triggers the creation of the SP Assertion with the IdP, Device 2 checks whether Device 1 is authorized. In addition, the user has to confirm the request by acknowledging a dialog displayed on Device 2.
- Step (8): Device 1 sends the obtained SP assertion to the SP, which checks the assertion (not shown) and provides the service. Finally, a SP session has been established.

5.3.4.2 Scenario 2: Identity Activation

Goal: Scenario 2 (→ Figure 5.16) shows the activation of an identity on Device 2. Device 1 triggers the activation.

Prerequisites:

- Successful discovery: It is assumed that Device 1 and Device 2 have just discovered each other and that Device 2 became the master device.
- Functionality distribution: A dedicated distribution of functionality among Device 1 and Device 2 is assumed. Device 2 is responsible for Prefiltering and Final Filtering. The Prefiltering phase, which is not shown, took place in advance.
- Secure Channels: For all interactions between the different parties, secure channels between the parties are assumed.
- No Failures: It is assumed that interactions are successful. Therefore, there is no need within these example scenarios to check for errors and react correspondingly.

Stepwise explanation:

- Step (1): The user requests a service on Device 1. The SP provides information about the supported authentication methods and the required user attributes.
- Step (2): Since device is not able to perform the Final Filtering, it requests a list of recommended identities from Device 2, which is the master device. Based on the requirements of the SP, the Final Filtering takes place on Device 2, which afterwards provides a list of possible identities to Device 1. The user selects one identity on Device 1.
- Step (3): Device 1 triggers the activation of the selected identity. Device 2 authorizes the request and displays an optional user confirmation dialog. Eventually, the authentication against the IdP takes place resulting in an IdP Session.

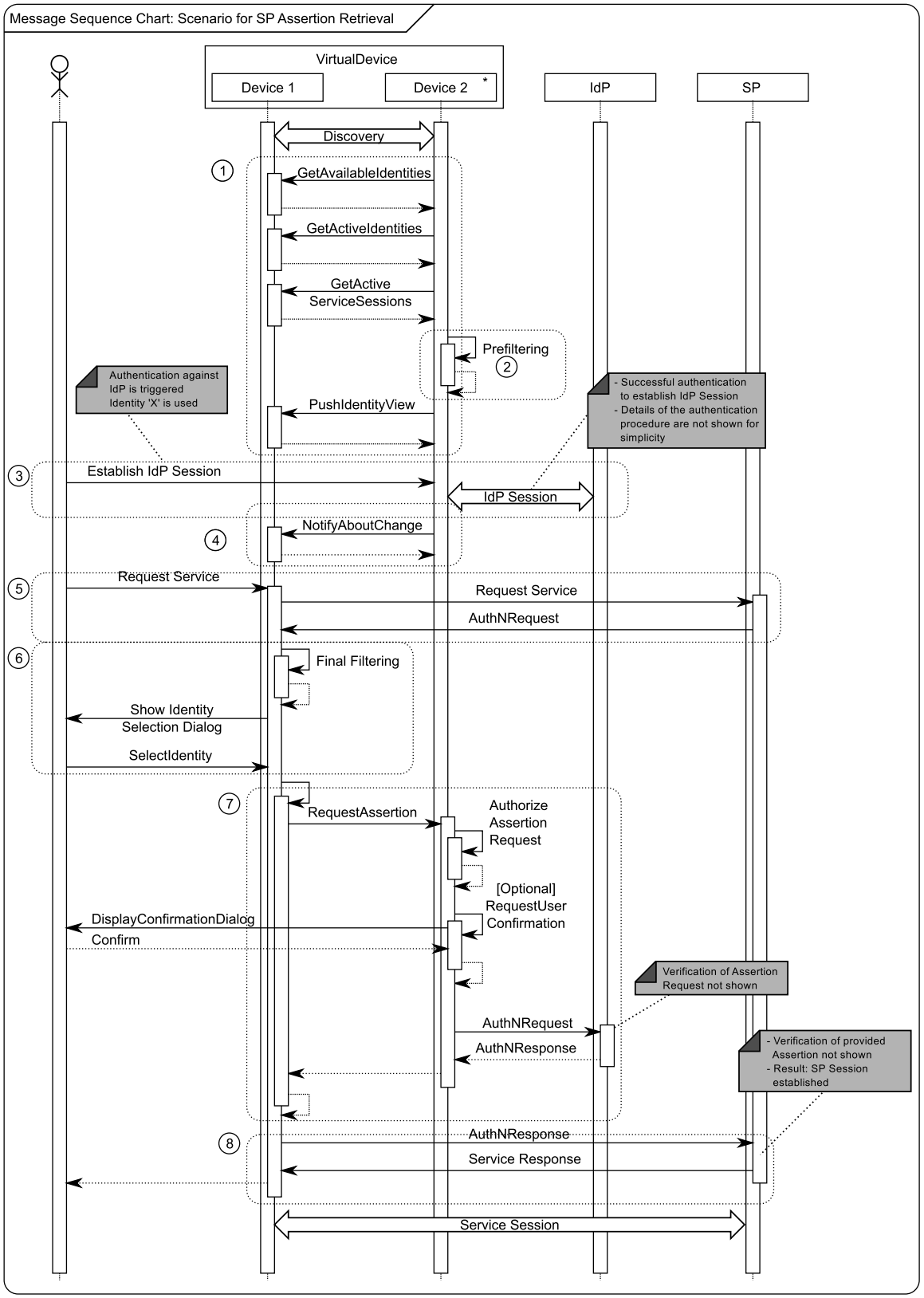


Figure 5.15: Message Sequence Chart: Example Scenario for the Retrieval of a SP Assertion

- Step (4): Based on the successfully activated identity, Device 1 is able to request a SP Assertion by triggering the *RequestAssertion* method. For more details on this step it is referred to the previous scenario.
- Step (5): Finally, Device 1 establishes the SP session.

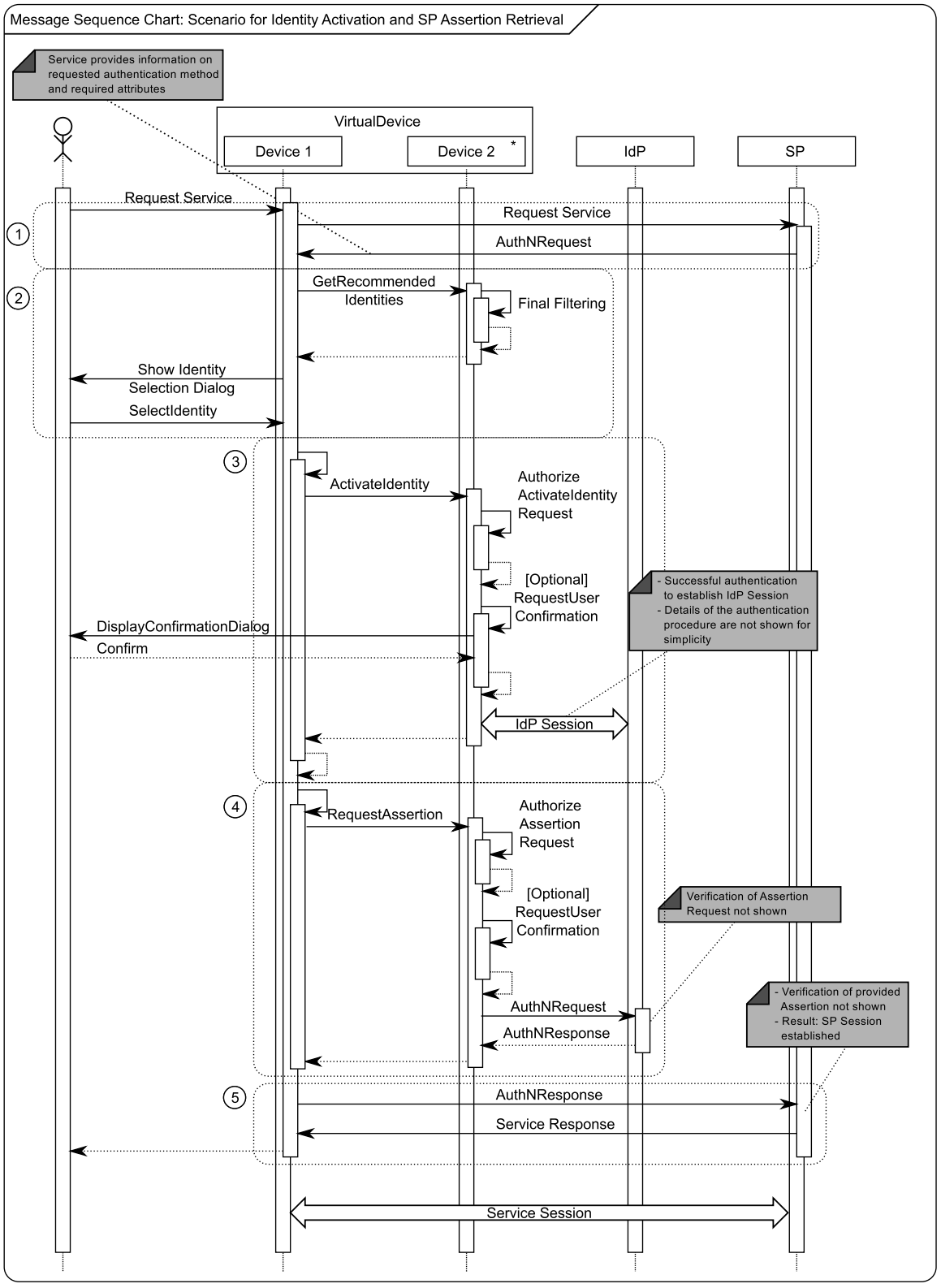


Figure 5.16: Message Sequence Chart: Example Scenario for the Activation of an Identity and subsequent SP Assertion Retrieval

5.3.5 Addressed Requirements

The multi-device IdM protocols address the requirements enumerated in Table 5.2.

Table 5.2: Requirements addressed by Multi-Device IdM protocols

Requirement	Addr.	Explanation
R2: Task distribution	Yes	The protocols provide the possibility to distribute authentication across devices.
R3: Remote activation IdM-IA-1: Protocol for identity activation	Yes Yes	ARP allows to trigger the authentication on a remote device (→ Section 5.3.3).
R8: Distributed data handling	Yes	IIEP exchanges identity information between devices.
IdM-AE-1: Protocol for the request of authentication assertions	Yes	Section 5.3.3 provides details on the designed protocol.
IdM-AE-2: Authorization based on device Id IdM-IA-2: Authorization based on device Id	Yes	IIEP, ARP and IAP consider mechanisms to perform authorization for requests (→ Section 5.3.1, Section 5.3.3 and Section 5.3.2).
IdM-AE-3: Information about usage context	Yes	The ARP and the IAP consider the transmission of usage context information to support access control decisions (→ Figure 5.14 and Figure 5.12).
IdM-AE-4: Manual confirmation	Yes	The ARP provides a primitive to request user confirmation upon requests. These primitives can also be used by the IAP. (→ Figure 5.13)
IdM-IM-6: Graphical user interface for identity selection	Partially	The existence of a graphical user interface for identity selection is imposed in Figure 5.15 and Figure 5.16.
SR-100: Mechanism to stop all IdP sessions	Yes	The IAP provides a method to stop all IdP sessions on a device (→ Figure 5.12). If this method is called on all devices, all IdP sessions are stopped.
SR-101: Mechanism to stop IdP sessions on one device	Yes	The IAP provides a method to stop all IdP sessions on a device (→ Figure 5.12).
SR-102: Mechanism to stop all SP sessions	Yes	The ARP provides a method to stop all SP sessions on a device (→ Figure 5.14). If this method is called on all devices, all SP sessions are stopped.

Continued on next page

Table 5.2: Requirements addressed by Multi-Device IdM protocols

Requirement	Addr.	Explanation
SR-103: Mechanism to stop SP session on one device	Yes	The ARP provides a method to stop a selected SP sessions on a device (→ Figure 5.14).

5.4 Virtual Device Management

The virtual device key concept realizes the foundation for the previously introduced multi-device IdM. From the IdM perspective, the security of a virtual device is the key for extending IdM to multiple devices. Therefore, this section introduces the concepts to create and organize a virtual device with respect to the necessary security mechanisms. These concepts are required to evaluate the security of the overall system in Section 6.2.

This section is structured as follows. Section 5.4.1 introduces basic concepts for the organization of a virtual device. This includes the various roles that a device is engaged in and the possible transitions between the roles. Section 5.4.2 describes the security architecture of the virtual device. The security architecture comprises the creation of security associations between the devices, the adding and removing of devices from the virtual device and the establishment of secure channels between the devices that are part of a virtual device. The description of an algorithm to determine the security level of an individual device concludes the subsection on the security architecture. Section 5.4.3 enumerates the addressed requirements. Finally, Section 5.4.4 details the impact of the virtual device concept on the multi-device IdM concept.

5.4.1 Organization

Figure 5.2 introduced the various states of the virtual device as a whole. In each of the states different activities take place. One activity is the master device selection that reorganizes a virtual device. The remainder of this section introduces the different roles of a virtual device and the activities to organize a virtual device.

5.4.1.1 Roles of Devices

A device can take different roles within a virtual device. This thesis distinguishes the roles of an independent device, a master device, and of a slave device as introduced in the following.

Independent Device: A device that has no contact to another device of the virtual device is considered as an independent device. It cannot rely on any other device for the provisioning of the multi-device IdM concept. If another device appears, an independent device can only become a slave device as follows.

Master Device: Within the virtual device composition, a master device takes over responsibility for the overall coordination. The overall coordination comprises the collection of information about devices and identities, and the execution of algorithms for the management of devices and identities ($\rightarrow$ Identity Filtering in Section 5.2). The role of the master device is conducted by only one device. If several devices pursuit for becoming a master device, the master device negotiation takes place and determines the master device. Different partitions of a virtual device do not exist, because the usage of services[7] requires connectivity and thus reachability of the individual devices. The operation of a disconnected partition is not useful in this context and therefore not considered.

[7] This thesis assumes that the services, which are offered by a SP, are based on Internet connectivity.

In order to become a master device, an independent device must fulfill the following requirements:

- Security Level: A master device must be a secure device, because it collects and manages sensitive information regarding devices and identities. That means the security level of the master device must be higher than the security level of any other device that is part of the virtual device and that is currently available.
- Human Computer Interface (HCI): A master device must have a HCI for the interaction with the user due to several reasons. First, it is often used to establish IdP sessions, i.e. the user has to perform the authentication on the master device. Second, for the management of the virtual device a user interface is necessary (→ Section 5.4.2.3).
- High availability: High availability means that the master device is locally available to the user, i.e. the master device is in the proximity of the user, and always in operation. A high availability is not a mandatory requirement, but rather helpful with respect to usability and performance. On one hand it improves the usability of the virtual device. A user gets accustomed to use the master device for example for authentication and management of the virtual device. On the other hand, it reduces the overhead of virtual device operation and makes IdM more efficient (→ Section 6.3).
- Master Device Capability: A potential master device has to have the master device capability, which can be considered as a flag that enables the device to become a master device. This restricts the set of potential master device candidates and avoids unintended decisions of the master device negotiation.

Slave Devices: If a device within the virtual device composition does not become a master device during the master device negotiation, it becomes a slave device.

Figure 5.17 shows the possible state transitions for one device. When the device is powered up, it takes the role of an independent device. This state might be left as soon as another device belonging to the virtual device is discovered. If one of the devices has the master device capability, it becomes the master device and the other device becomes the slave device. If none of the devices is able to become a master device, both devices remain independent of each other. If two devices compete for becoming a master device, the master device negotiation resolves the conflict and one device is becoming the master device. The role of a slave device or master device is left if a new device appears or disappears and the master device negotiation has to be triggered once again. The master device negotiation takes place between two devices. If more than two devices have the master device capability, master device negotiation is repeated until one of the devices emerged as master device. More efficient master device negotiation processes are subject to further research.

In addition to the devices that are part of a virtual device, the existence of a secure storage *SecStore* that is always available is assumed. Such a secure storage simplifies many virtual device management tasks. The secure storage is considered in the following as a device that is part of the virtual device composition which is always available and which provides secure storage. Therefore, the same mechanisms of the latter specified security architecture apply. The role of the *SecStore* can be provided by any device that is part of the virtual device.

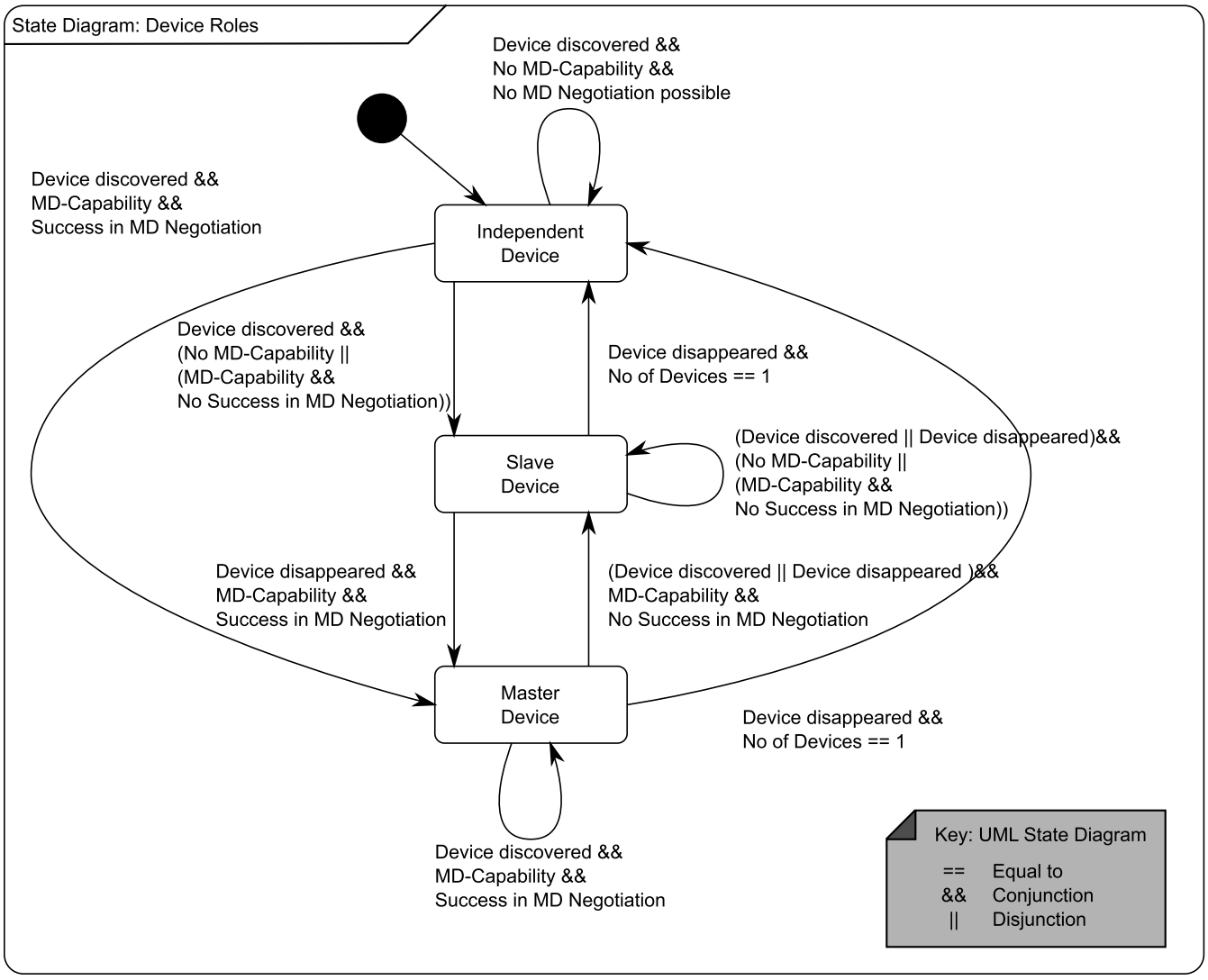


Figure 5.17: State Diagram for Role Transitions of Devices within Virtual Device

5.4.1.2 Virtual Device Organization

Figure 5.18 specifies the activities that take place when two devices discover each other. It is assumed that both devices belong to the same virtual device. The activities might lead to a change of the roles ($\rightarrow$ Figure 5.17) and thus to a reorganization of the virtual device. Whenever a new device is discovered, it has to be checked whether the master device negotiation is necessary[8]. The actual protocol and the algorithms for the master device negotiation are out of scope of this thesis.

If the device is not becoming the master device, it just registers the device identifier of the new master device. If the device is becoming the master device it triggers the information exchange with the slave devices. The information exchange is realized by a flexible framework that allows the registration of various handlers that are called when the information exchange is triggered. One of those handlers triggers the identity synchronization. The identity synchronization is realized by the IIEP and has been introduced in Figure 5.15. In addition, the exchange of device characteristics is triggered to deal with new or changed device properties. Changed device properties, e.g. caused by a system update, are of interest for the determination of the security level. Changed security levels have consequences with respect to master device negotiation and identity filtering.

Figure 5.19(a) and Figure 5.19(b) provide an additional view on the virtual device formation. Both figures assume that all devices have the master device capability, which is indicated by the '$(*)$' symbol. In Figure 5.19(a), D_1 and D_2 discover each other and enter the process of the master device negotiation. D_1 succeeds and becomes the master device indicated by the '$*$' symbol without brackets[9]. At t_3, D_3 is discovered and renegotiation between D_1 and D_3 takes place. D_3 succeeds and is responsible for the further organization of the virtual device.

A similar procedure takes place when one of the devices disappears. Figure 5.19(b) illustrates what happens when D_3 disappears. D_3 becomes an independent device and the two remaining devices have to renegotiate the role of the master device. D_1 succeeds in the master device negotiation.

[8]The master device negotiation is only necessary, if more than one available device has the master device capability.

[9]The security level of the device is a dominating reason to succeed in the master device negotiation. Section 5.4.2.4 details how the security level is determined.

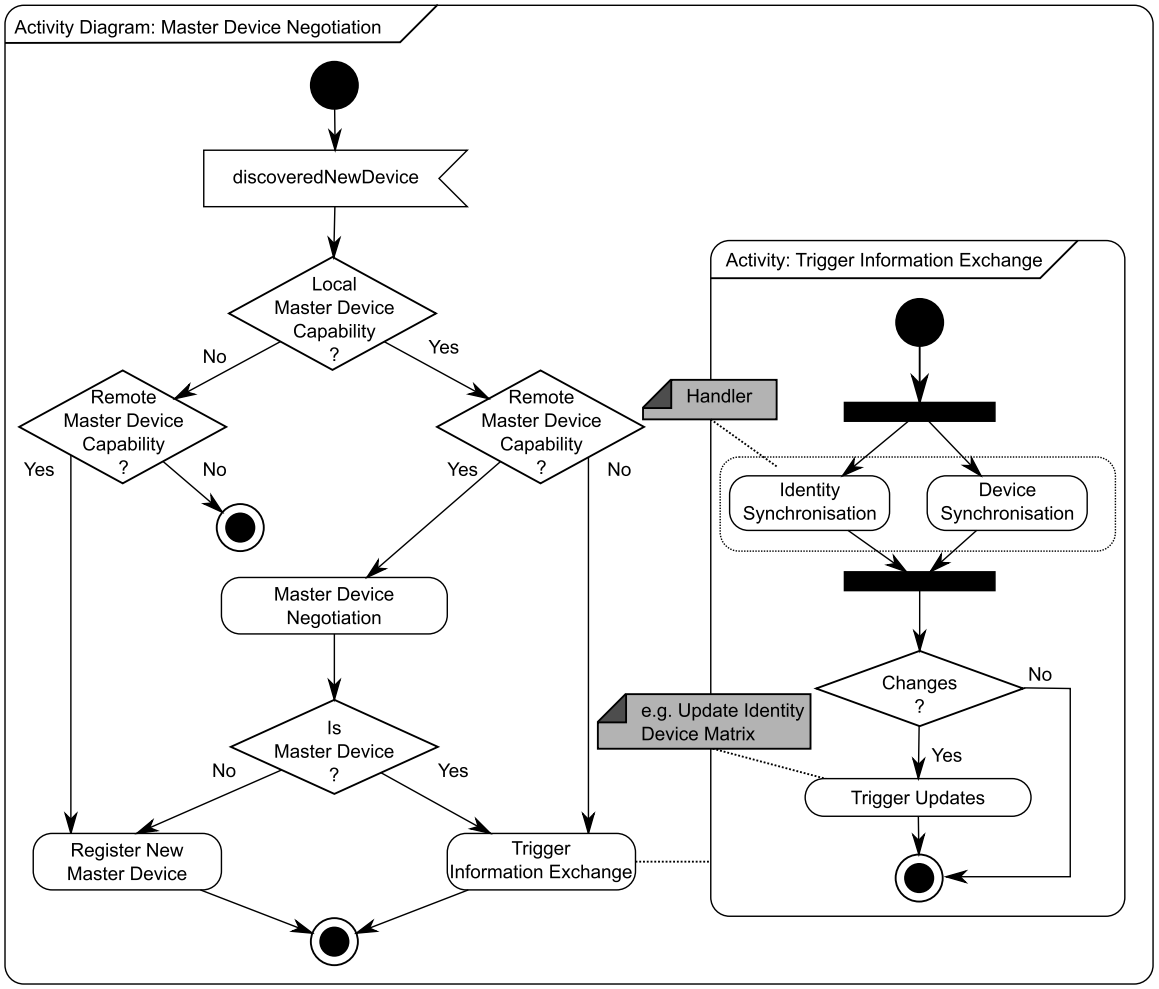
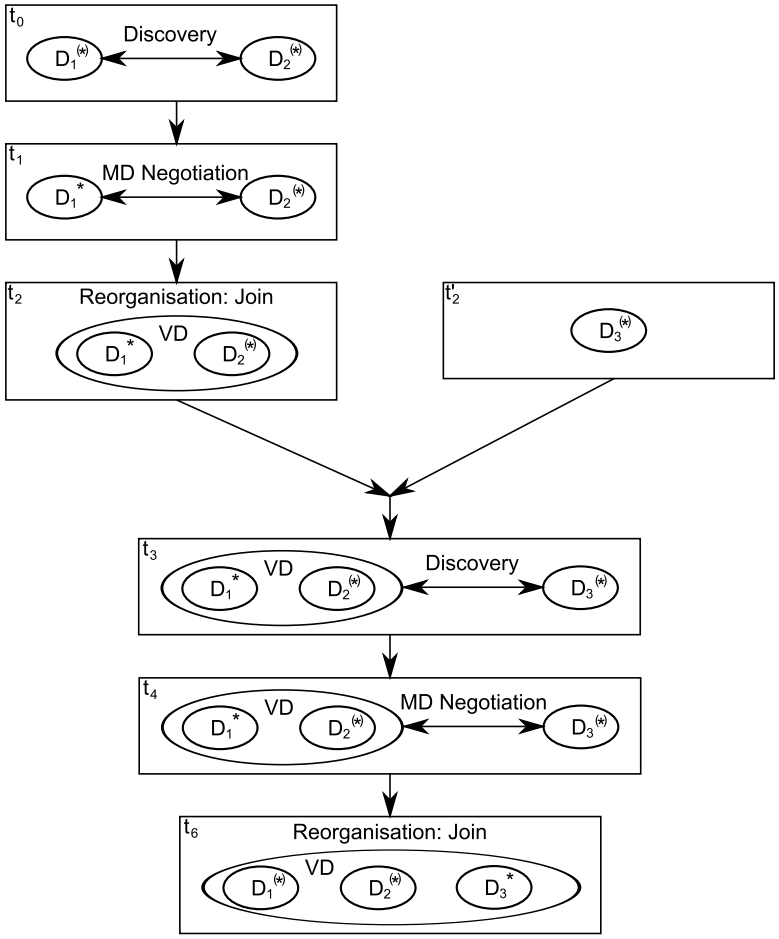
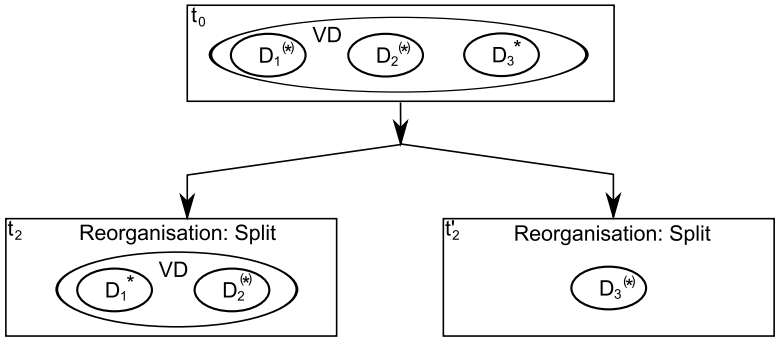


Figure 5.18: Activity Diagram on Master Device Management



(a) Virtual Device Formation



(b) Virtual Device Splitting

Figure 5.19: Structural View on Virtual Device Reorganization

5.4.2 Security Architecture

The security of the virtual device is of uttermost importance. If the security is compromised, various attacks may succeed (→ Section 6.2). This section details the security architecture of the virtual device. It addresses the following problems:

- Virtual Device Membership: Each device must be able to verify whether another device is a member of the virtual device or not. This is achieved by having security associations between all devices (→ Section 5.4.2.1).

- Establishment of Secure Channels: For the secure communication between devices that are part of a virtual device secure channels are required. The security associations between devices represent the prerequisite for the establishment of secure channels (→ Section 5.4.2.2).

- Modification of Virtual Device: It must be possible to modify the virtual device, i.e. adding or removing devices (→ Section 5.4.2.3).

- Security Level Determination: For many decisions (e.g. master device negotiation, identity filtering) the security level of a device is important. An algorithm is required to determine the security level (→ Section 5.4.2.4).

5.4.2.1 Virtual Device Membership – Security Associations

All devices that are part of a virtual device must be able to verify whether another device is part of the virtual device. One way to achieve membership verification is the establishment of security associations between all devices. A security association is defined as “a relationship between two or more entities to enable them to protect data they exchange” [RFC4949].

In the following the requirements regarding virtual device membership are summarized.

- Membership Verification: Any two devices of a virtual device must be able to mutually verify the virtual device membership (→ DM-SA-1 in Section 4.3.3).

- Adding of Devices: It must be possible to add devices to the virtual device. An added device must be able to verify the virtual device membership of any other device. Hereby, it is not needed that every device is in the position to add other devices to the virtual device (→ DM-DM-1 in Section 4.3.3).

- Removing of Devices: It must be possible to remove any device from the virtual device. A removed device must not be able to establish a secure channel with any device that is still part of the virtual device (→ SR-2 in Section 4.4.5).

- List of Devices: It must be possible that the user obtains a list of all devices that are part of the virtual device (→ SR-7 in Section 4.4.5).

Different solutions are possible to address these requirements. One solution is presented in the following. The solution for the management of security associations relies on asymmetric encryption, certificates and membership lists.

With asymmetric encryption the possessor of a private key can prove the possession of the key without revealing it. If the possessor is able to decrypt a message that has been encrypted with the corresponding public key, he must possess the private key.

However, possessing a private key does not provide any information about the device identity of the possessor. Private and public keys can be easily created or transferred between different parties. Therefore, it is required that the public key is bound to the device itself and to the virtual device. Such a binding can be achieved with certificates. Since certification is only relevant within the virtual device, there is no need for a globally existing 3rd party that serves as a trust anchor. It is sufficient that the scope of the certification is limited to one virtual device.

Finally, a mechanism is required to handle lost private keys. Lost keys have to be considered due to the following reasons:

- Removal of Device from Virtual Device: If a device is removed from the virtual device, it must be guaranteed that all other devices do not consider this device as being part of the virtual device anymore. Even if the device possesses the private key, a mechanism is required to invalidate the key.
- Compromised Device: If one device that is part of the virtual device becomes compromised, it must be possible to revoke the corresponding key and remove the device from the virtual device.

Two approaches exist to deal with lost keys. In the Internet so called revocation lists are used. Revocation lists enumerate lost keys. Such lists have to be consulted to verify the validity of keys. The other approach, which has been selected, are membership lists. The membership list enumerates all devices that are part of the virtual device. In order to add a device to the virtual device a corresponding entry in such a list has to be established. This approach allows the verification of the membership of devices within a virtual device. As a side effect, the user can be provided with an overview on all devices of a virtual device. Since the number of devices per user is assumed to be limited, management of explicit membership lists is not a problem regarding scalability.

Figure 5.20 shows the introduced certificate hierarchy for the management of the virtual device membership. It distinguishes three different levels of certificates in order to distinguish the different roles of devices: Virtual-Device Level, Master Device Level, and Non-Master Device Level.

1st Level – Virtual Device Level: For each virtual device a private and a public key exists. This key pair is only required for the certification of public keys on the next level. It is very security critical and has to be stored in a safe place ($\rightarrow$ SSE in Section 4.5.4). Since the private key is only needed to add a new master device it could be stored offline or protected by means of a Trusted Computing Platform (TCP)[GM09].

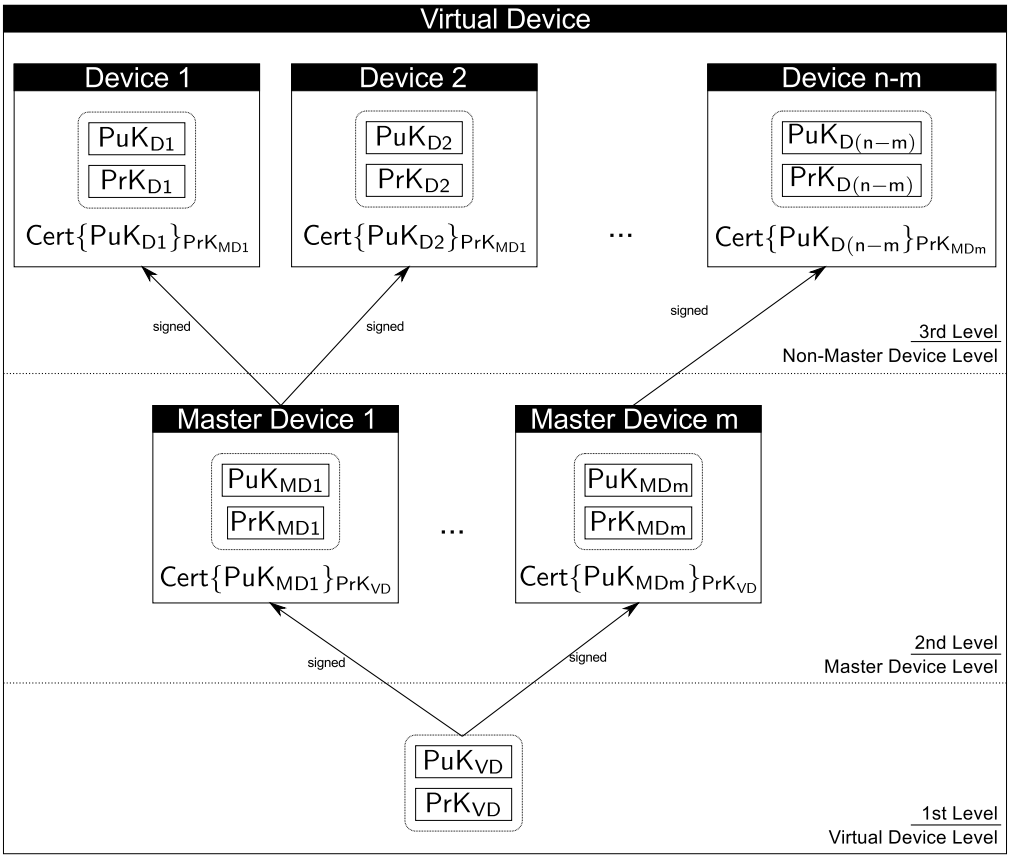


Figure 5.20: Certificate based Security of Virtual Device

2nd Level – Master Device Level: Devices with the master device capability have a key pair on the master device level. The public key of the master device is certified by the private key of the virtual device and binds the virtual device identifier, the device identifier and the public key of the master device together. With a key pair of the master device level, it is possible to certify public keys of the non-master device level.

3rd Level – Non-Master Device Level: Devices on the non-master device level are ordinary devices, i.e. devices that are not in the position to sign any other keys. Ordinary devices possess a certificate that binds the virtual device identifier, the device identifier and the public key of the device together, signed with the private key of a master device.

Basically it is possible to decouple the certificate hierarchy from the master device concept. The coupling of the certificate hierarchy and the master device concept is reasonable due to the following reasons. First, to become a master device a certain security level has to be reached, i.e. master devices are secure. Second, the master device is assumed to be a device that is essential for the user, i.e. often used and close to the user. Therefore, it is reasonable that master devices take over the responsibility to manage the virtual device. These reasons substantiate the 2nd level of the certificate hierarchy and the corresponding alignment to master devices.

Each device has a list of devices that are members of the virtual device. The membership list contains an entry for each device that is part of the virtual device. To verify its authenticity it is signed by one of the master devices. Since the membership lists can change over time, it

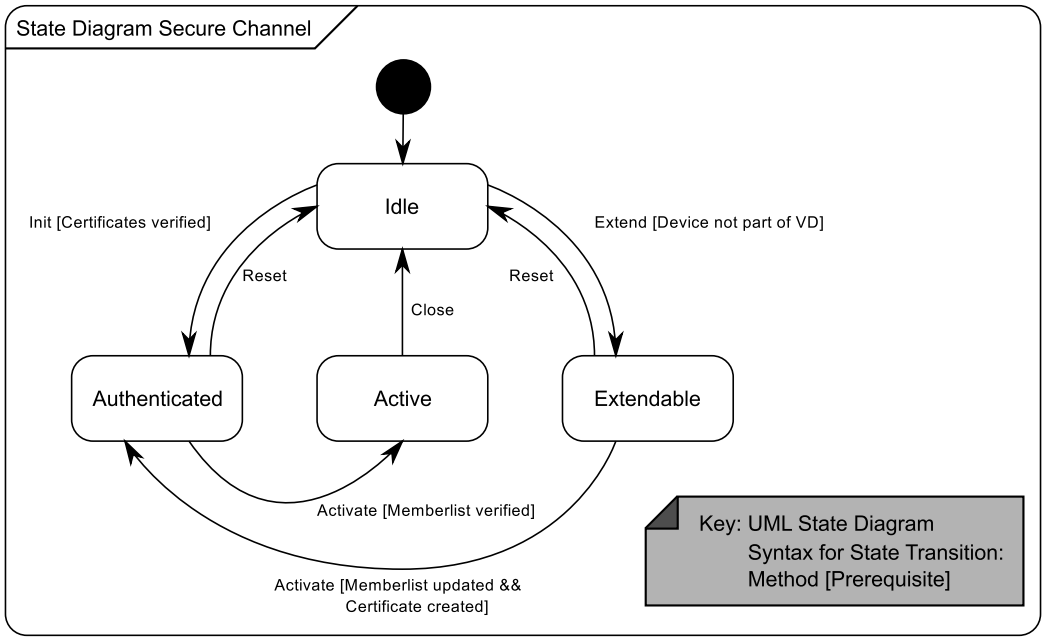


Figure 5.21: State Diagram for the Establishment of a Secure Channel

contains a version number that increases with each change of the membership. This number allows a device to verify whether the own list is older or newer compared to another list. For simplicity, the in Section 5.4.1 introduced secure storage stores the membership list and makes it accessible to all devices.

5.4.2.2 Establishment of Secure Channels

Secure channels are the foundation for all protocols that are used to manage the virtual device and for those on top of the virtual device (e.g. multi-device IdM). They enable secure communication between two devices of the virtual device. Secure channels have the following properties:

- Mutual Authentication: Both devices at the ends of a secure channel mutually authenticate each other. That means both devices are sure that the other device is the one it pretends to be. In addition, subsequent verification of certificates and membership lists guarantees that the other device is member of the same virtual device.
- Confidentiality: The channel does not leak any information to potential eavesdroppers. Confidentiality is achieved by usage of cryptography.
- Integrity: Information that is modified during the transmission across the channel is detectable. Integrity is achieved as a side-effect of applying cryptography.

Figure 5.21 shows the four different states of a secure channel. In the state *Idle* the secure channel is not established, but ready for establishment. An establishment request from a remote device or a corresponding trigger from the local device leads to the initialization of the secure

channel. During the initialization the two devices exchange certificates and verify the certificates. The verification checks whether the certificate belongs to the virtual device and whether it is signed by a master device or with the public key of the virtual device key pair. If the certificate verification is successful, the secure channel enters the state *Authenticated*. This state is left and the state *Active* is entered upon successful verification of the virtual device membership list. How the virtual device membership list is updated and verified is detailed in Figure A.1. In the state Active, the secure channel is ready to be used by protocols on top. When the protocol on top closes the secure channel, it enters the state Idle again. All secure channels to a device, which is going to be removed from the virtual device, are closed. This prevents persisting secure channels to non-member devices. For the extension of the virtual device, i.e. adding a device, the extra state *Extendable* exists. Section 5.4.2.3 provides more details on the extension of a virtual device.

The TLS protocol (→ Section 2.4.3) provides an adequate basis for the realization of the secure channel. It supports the transmission of certificates between two devices and provides adequate cryptographic functions. The verification of certificates and of the membership lists can be achieved by wrapping an existing TLS implementation. Such a wrapper extends the state space of TLS.

5.4.2.3 Modification of Virtual Device

For the security evaluation in Section 6.2 it is important to specify how a device is added to the virtual device. According to the design goals of usability and security, it is important that adding a device is easy but secure at the same time. Secure means that it must not be possible to add a device without the consent of the user. Therefore, the user has to be included in the procedure of adding a device by explicitly triggering dedicated actions.

Figure 5.22 details the process to extend a virtual device. The user wants to add Device 1 to the virtual device with the support of Device 2, which is a master device. The following steps take place:

- Step (1): The user installs the required middleware for the virtual device.
- Step (2): The user triggers the preparation procedure to include the device into the virtual device. In this step, Device 1 creates a device identifier[10] and a corresponding certificate. In addition, a secure channel is put into the Extendable state.
- Step (3): On Device 2, the user requests the admission of Device 1 to the virtual device by providing the previously created device identifier. With the device identifier, Device 2 discovers Device 1 and establishes a secure channel. Afterwards, Device 2 displays a PIN on the display.
- Step (4): The user enters the PIN on Device 1. Device 1 encrypts the PIN with the public key of Device 2 and sends it via the secure channel to Device 2. This serves as proof for Device 2 that Device 1 should be added to the virtual device. By manually

[10] Depending on the device type, the user is included in this step.

entering the PIN, the user is actively involved. This limits the risk that the secure channel is established with another device.

- Step (5): After all previous steps have been successfully passed, Device 2, which is a master device, creates a certificate for Device 1 and adds Device 1 to the membership list of the virtual device. Finally, it provides the updated list to Device 1, which displays a success message. The overall procedure is finished with the display of a success message on Device 2.

The removal of a device is at least as important as the addition of a device to the virtual device. The removal of a device must not depend on the device that should be removed. Two cases are distinguished: Removal of devices with master device level certificate and removal of devices with non-master device level certificate.

- Removal of devices on the Non-Master Device Level: A device that is not a master device level should be removed. Any master device is able to remove such a device simply by removing the device from the membership list. An updated list is provided to all devices of the virtual device.
- Removal of devices on Master-Device Level: If a device on the master-device level should be removed, two different cases have to be distinguished depending on the availability of a second master device.
  - Case 1: Removal by another device on Master-Device Level: One master device can remove another master device from the virtual device by removing the device from the membership list. The removal of a device on master-device level results in the removal of all devices that have been added by the corresponding master device[11].
  - Case 2: Complete Reset of Virtual Device: If no other master device is available and if the virtual device key pair is not available, the virtual device has to be reset. Such a mechanism is subject to further research[12].

The membership list is security critical with respect to the disclosure of information about the virtual device composition. To avoid unauthorized modification, all master devices sign the list after successful modification. Such a signature can be verified by all devices. The secure storage has to protect the list against eavesdropping. This can be achieved by an authorization mechanism based on the available device certificates. Details of such a mechanism are out of scope of this thesis.

5.4.2.4 Security Level Determination

The security level of a device is a security metric that allows the quantification of the security of a device. It is a servant for risk management [Jaq07], which is required for decisions regarding

[11] To avoid the removal of all dependent devices, more complex mechanisms that consider timestamps are feasible. Such mechanisms are out of scope of this thesis.

[12] A quorum based approach could be considered for example that resets the virtual device if a certain number of member devices votes for the reset.

[13] For the sake of a simple representation logging activities have been neglected.

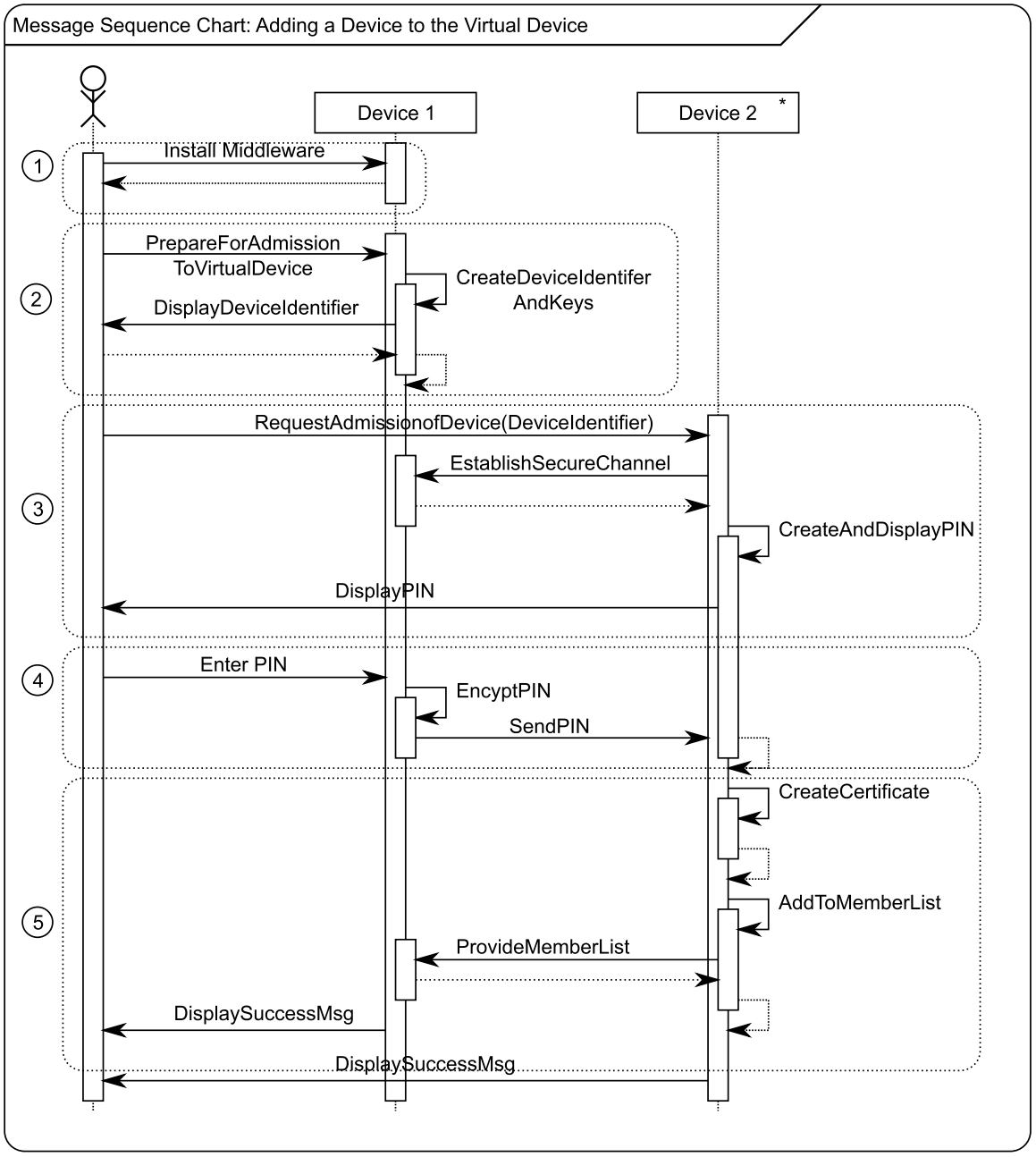


Figure 5.22: Message Sequence Chart: Extending a Virtual Device[13]

the function placement on devices. In particular authentication should take place on the most secure device.

The following aspects are important for the determination of the security level:

- Operating System (OS): The kind of OS (e.g. Windows, Android, Linux) on its own does not matter [Koe04]. The patch level, which determines whether discovered vulnerabilities have been covered by appropriate updates, is more important. In particular for some platforms the time to reach the latest patch level is high [Sve11].
- Configuration of System: The configuration of the system and the corresponding OS determines whether vulnerabilities due to misconfiguration exist. Misconfiguration is the reason for many security incidents [WW10] and difficult to detect. Therefore, policies regarding the configuration are required. For example a policy might require the encryption of persistent storage.
- Modification of System: Malware enables attackers to maliciously use the system and to intercept data. Proposals like Trusted Computing (TC) [GM09, GCB+08] and extensions [SJZvD04] allow the identification of system and configuration modification. Additional tools like anti-virus programs or personal firewalls can prevent or detect the modification of the system.

The absolute quantification of security is not possible. Therefore, this thesis proposes a metric that evaluates the security of devices in relation to each other. The security level is expressed by a value S which is a continuous number between zero and ∞. The higher S, the less secure the device. If S is zero, no vulnerability exists and the configuration of the device is considered as secure. The process to determine S consists of five phases.

(1) Data Capturing: In the first phase the *DM-LDM* captures device characteristics. These are the installed software, its patch level and the configuration of the device ($\rightarrow$ Figure 5.23). [OES11, Ser10] provide a certified framework to capture device characteristics, which is used for endpoint assessment. A suitable candidate for the exchange and in particular for the request of device characteristics is the Network Endpoint Assessment standard [RFC5209].

(2) Evaluation of Configuration: The second phase evaluates the captured configuration against defined policies. The defined policies demand the existence or absence of dedicated configuration items. For example a policy dictates the existence of encrypted persistent storage. If a device does not meet a policy, S is increased by a policy specific value c_x that is weighted by a factor C_C. The policy specific value reflects the importance of the corresponding configuration item. C_C determines the weight of the configuration in the overall determination of S.

(3) Evaluation of Installed Software: The third phase evaluates the existence or absence of software in addition to the operating system. If a policy specifies a dedicated piece of software (e.g. an anti-virus program) as mandatory, the absence results in an increase of S by a policy specific value s_x that is weighted by C_S.

(4) Evaluation of Software Version: The fourth version evaluates the installed software version regarding vulnerabilities. Existing vulnerability databases, like the NIST National Vulnerability

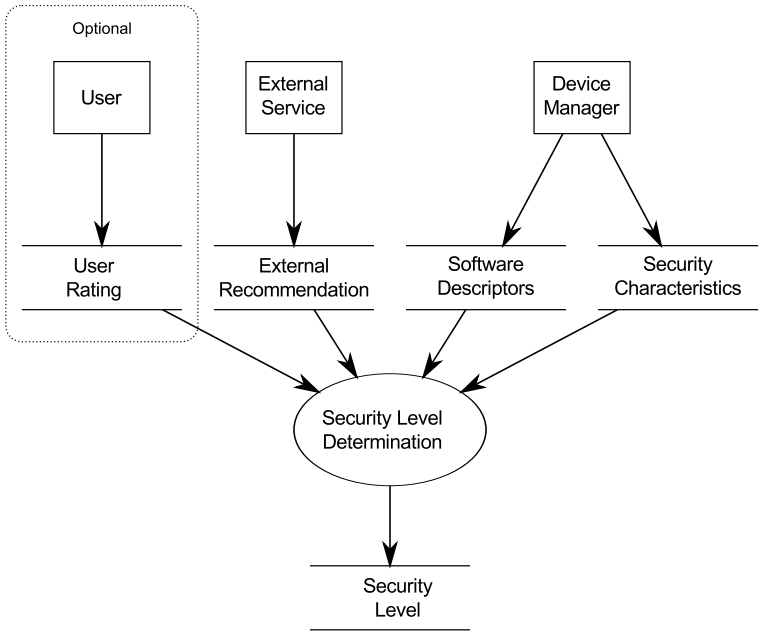


Figure 5.23: Data Flow Diagram for Determination of Security Level

Database (NVS)[14] or the Open Source Vulnerability Database (OSVDB)[15], provide details on known vulnerabilities of software products including OSs (indicated as External Service in Figure 5.23). The installed software versions are compared against identified vulnerabilities. If it identifies a vulnerability it decreases S with a by C_V weighted value of the severity of the vulnerability v_x. The severity of the vulnerability is expressed by a CVSSv2 score. For details on the determination of the CVSSv2 score it is referred to [MSR07].

(5) Evaluation of Modification: Depending on the device and the existence of TC mechanisms it is possible to identify modifications of the installed software. If a modification is detected, S is increased by C_M

Eq. (5.5) sums up the formula to determine S.

$$S = C_C \sum_{i=1}^{N_C} c_i + C_S \sum_{i=1}^{N_S} s_i + C_V \sum_{i=1}^{N_V} v_i + C_M \tag{5.5}$$

With the determination of S for each device, the security levels of different devices can be compared. It is not possible to make absolute statements regarding the overall security of a device. However, it is possible to perform the authentication on a device that is considered as more secure than another one. The determination of maximum values for S that should not be exceeded to perform authentication is subject to further study.

[14] http://nvd.nist.gov/
[15] http://osvdb.org/

5.4.3 Addressed Requirements

The described virtual device management concept addresses the in Table 5.3 enumerated requirements.

Table 5.3: Requirements addressed by the Virtual Device

Requirement	Addr.	Explanation
R1: Secure exchange	Yes	The secure channel allows the establishment of a mutually authenticated, confidentiality-protected and integrity-protected communication channel between the devices of a virtual device.
R4: Discovery of user devices DM-DD-1: Device discovery in general DM-DD-2: Proximity detection DM-DD-3: Device discovery should not reveal any information	No	The discovery of user devices has not been addressed in more detail. The discussion of related work provides an overview on existing mechanisms to perform device discovery (→ Section 5.4.5.4)
R5: Capture of device characteristics	Partially	No details on the mechanisms how device characteristics are captured are provided. However, the determination of the security level considers device characteristics in more detail.
R6: Establishment of security associations	Yes	The virtual device management details the process to establish security associations between different devices of a virtual device.
DM-SA-1: Mutual authentication	Yes	The security architecture of the virtual device addresses this requirement (→ Figure 5.21 and Figure A.1).
DM-VDM-1: Adding a device to VD	Yes	Figure 5.22 details the process to add a device to the virtual device composition.
DM-VDM-2: Removing a device from VD SR-2: Mechanism for removing devices from virtual device	Yes	Removing a device is detailed in Section 5.4.2.
SR-4: No dependency on single device	Yes	The security architecture does not depend on a single device (→ Section 5.4.2). The only exception is made for the single storage that has to be available for the management of membership lists. If no up-to-date memberlist exists, a fallback is considered that requires the user's intervention (→ Figure A.1).

Continued on next page

Table 5.3: Requirements addressed by the Virtual Device

Requirement	Addr.	Explanation
SR-5: Authorization mechanism for modifying a virtual device	Yes	Figure 5.22 contains a mechanism to avoid unauthorized addition of devices to the virtual device by explicitly including the user into the process.
SR-7: Mechanism to obtain information about virtual device composition	Yes	The user can easily obtain a list of all devices that are part of the virtual device by the membership lists.
SR-9: Confidentiality of information about virtual device composition	Partially	This aspect is partially considered by the security architecture in Section 5.4.2.
SR-10: Secure transmission of IdP token	No	This aspect is not directly related to the virtual device and is also subject of the IdP.
SR-12: Secure transmission of SP token	No	This aspect is not directly related to the virtual device and is also subject of the SP.
SR-15: Secure transmission of authentication credentials	No	This aspect is not directly related to the virtual device and is also subject of the IdP.
SR-17: Secure transmission of SP-assertions	Yes	The secure channel Section 5.4.2 addresses this issue.
SR-20: Storage of all security properties only on secure devices	Yes	The process of master device selection guarantees that only master devices obtain a comprehensive view on the virtual device including security properties.

5.4.4 Impact of Virtual Device Concept on Multi-Device IdM Concept

The consequences of the reorganization of the virtual device with respect to the multi-device IdM concept have not been discussed so far. Reorganization of the virtual device has an impact on existing IdP and SP sessions as well as on future sessions.

5.4.4.1 IdP and SP Session View

The consequences of the reorganization of the virtual device on existing IdP and SP session adheres to the following principles:

- Persistence of IdP Sessions: If an IdP session has been established on a device that had formerly the role of the master device, the IdP session remains active on this device until

the device becomes unavailable or until the IdP session times out. This holds, if the role of the master device is carried out by another device belonging to the virtual device. If the device has been able to establish the IdP session, there is no reason to tear down and reestablish the IdP session on another device. The security level of the former master device has been sufficient to establish the IdP session.

- Persistence of SP Session: If an SP session has been established on a device, the SP session remains active until the SP session is torn down by the user, the SP session times out, or the device becomes unavailable. Tearing down a session upon virtual device changes would decrease the usability, because the user would have to change devices to continue the service consumption. From a security perspective no increase is expected if the session would be torn down. If the device has been able to establish the SP session, the security level of the device has been sufficient. If the security level of a device changes, ARP provides primitives to tear down SP sessions.

- No Session Migration: If a device disappears, the IdP session and the SP session on the device will be torn down. No session migration mechanisms are considered. If an IdP session or an SP session is required on one of the remaining devices it has to be reestablished.

- Network Connectivity: All available devices have network connectivity and can reach each other. This is justified by the fact that the overall system is only useful if the IdP and SP can be reached via the network. The principle of network connectivity has the following consequences: (1) If a device is available it becomes immediately an active part of the virtual device. (2) If a device becomes unavailable, it is not part of the active virtual device anymore. (3) All devices that are part of the active virtual device have the same view on the active virtual device, i.e. all devices recognize the same master device.

Figure 5.24 exemplifies these principles by means of four base cases. The base cases can be used to construct more complex scenarios in which more than two devices are involved. The four different cases are arranged in matrix structure. Case 1 and Case 2 deal with appearing devices. Case 3 and Case 4 deal with disappearing devices[16]. Case 1 and Case 3 assume two devices with master device capability (i.e. Device 1 (D_1) and Device 2 (D_2)). It is assumed that D_1 is more secure than D_2. Case 2 and Case 4 assume one device with master device capability and one device without master device capability (i.e. Device 1 and Device 3). The case of two devices without master device capablity is not considered, because it does not fulfill the requirements for the formation of a virtual device.

Each base case consists of two subcases that differ with respect to the order of the appearance/disappearance of devices. For example with Case 1.1, D_1 is available at t_0 and D_2 appears at t_1. For Case 1.2 the order is the other way around. All cases assume that the device, which is available at t_0 establishes an IdP session and a SP session.

[16]Disappearing means that a device becomes unavailable, e.g. no network connectivity. The removal of a device from the virtual device is not considered as a reason for disappearance. All IdP sessions and all SP sessions have to be explicitly torn down before the device is finally removed. This is only feasible if the device, which should be removed, is available. If it is not available, it has no network connectivity and thus existing SP and IdP sessions time out

With Case 1.1 and 1.2, the existing IdP sessions reside on each device after t_1. New IdP sessions are established on D_1, because D_1 is more secure and succeeds in the master device negotiation[17]. SP sessions can take place on D_1 and D_2 according to the user's intention.

With Case 2.1 the IdP session and SP session resides on D_1. D_2 might be used for SP sessions. With Case 2.2, the IdP session has been established on D_3, which has no master device capability. This session has been established, because each device is able to fall back to independent operation. After establishment of the active virtual device at t_2, the IdP session is torn down if D_3 does not provide an adequate level of security. If the IdP session has to be established, it is established on D_1.

With Case 3.1 and Case 3.2, one of the master devices disappears. The IdP sessions and the SP sessions on the disappearing device terminate. The same holds for Case 4.1 and for Case 4.2.

Concepts to enable session migration of the IdP session are subject to future work. Migration of SP sessions to enable session mobility has been examined in [Mei08].

5.4.4.2 Virtual Device Organization and Session Establishment

Figure 5.2 showed the different states of a virtual device. Within the state "Virtual Device Organization" all necessary procedures to reorganize the virtual device takes place. Since these procedures, e.g. master device negotiation, impact the future of the virtual device, all activities that require an operational virtual device cannot take place. Figure 5.25 shows that a discovered device puts the virtual device into the state "Virtual Device Organization". The user that wants to establish a service session, which is indicated by the "Service Request", has to wait until the state "Virtual Device Operation" is entered. The restriction of certain activities to the operational state is required to avoid race conditions and to operate on settled data.

5.4.5 Related Work

This section puts the mechanisms and concepts introduced with the virtual device management in relation to existing work.

5.4.5.1 Virtual Device and Related Concepts

The concept of virtual devices or personal networks is well known in literature [AIDV07, CMCP06, FSF+04, NHdG02]. Several devices belonging either to one user or to trusted parties are cooperating to achieve a common goal. Among the goals are: Network access, data provisioning, capability sharing, and context management.

[17] If D_1 does not have sufficient authentication capabilities, it can delegate the establishment of an IdP session to any device that adheres to the corresponding requirements (e.g. security level). This case is not reflected in Figure 5.24 for simplicity reasons.

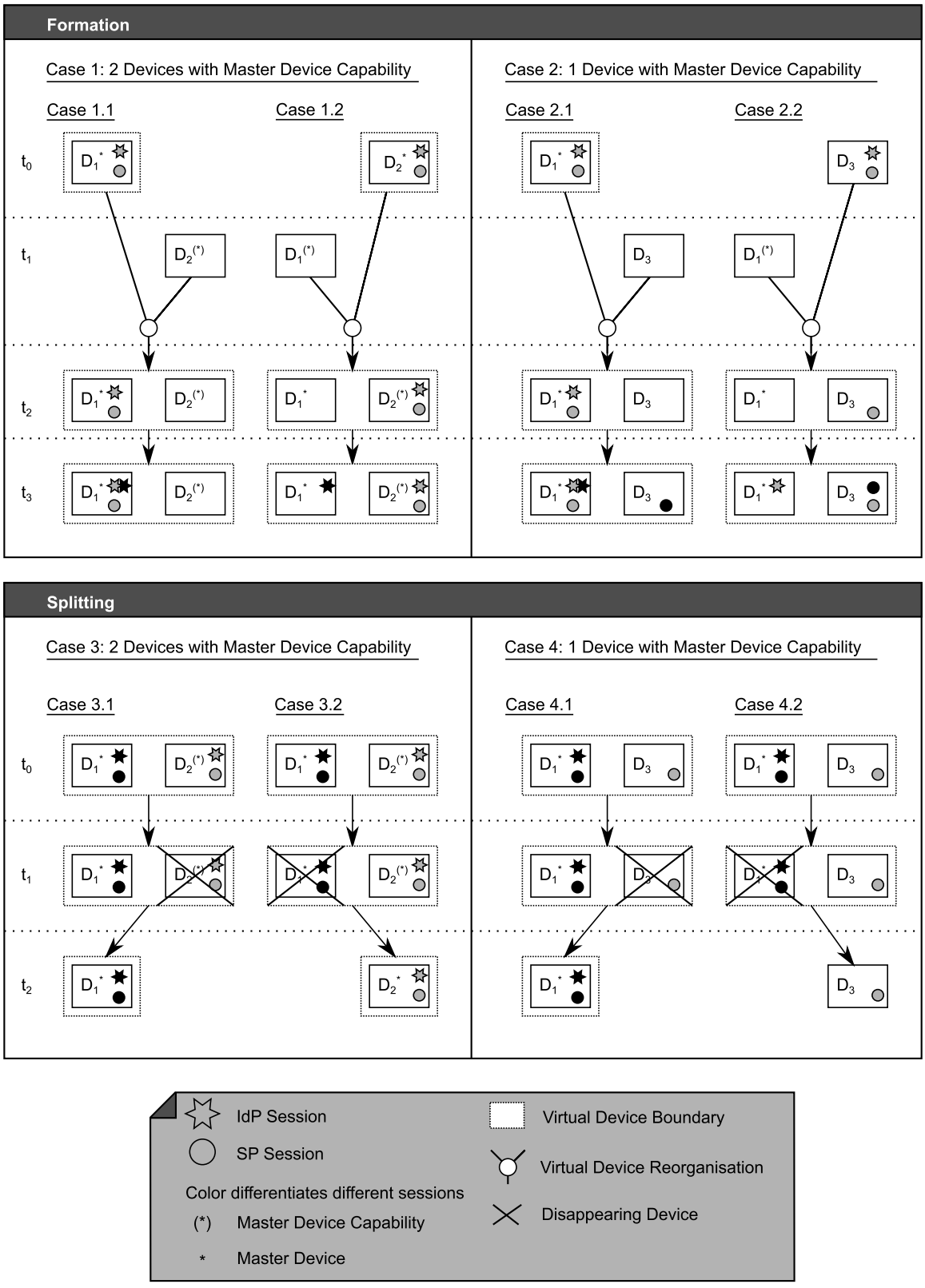


Figure 5.24: Impact of Virtual Device Reorganization on Existing Sessions

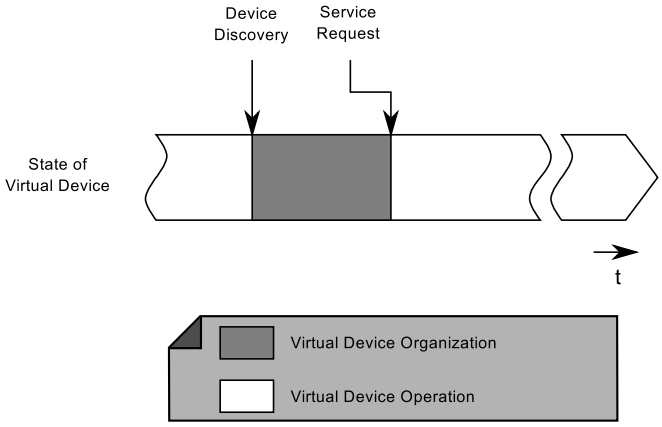


Figure 5.25: Virtual Device Organization and Session Consumption

- Network access: One device acts as a gateway and provides global connectivity to other devices. Hereby, the selection of the gateway and the corresponding interface is challenging [AIDV07, JMRA08].

- Data provisioning: Users want access to their personal data from all devices. In particular for copyright protected content, content owners want to limit access by digital rights management systems. Several solutions have been standardized [OMA11] or proposed to enable access on all user devices [SAN03, KMKI07].

- Capability sharing: The devices being part of a virtual device are heterogeneous regarding their capabilities (e.g. display size, computing power). From the user perspective the most benefit can be obtained, if the devices share their capabilities [FSF+04, SDV+07]. This includes the distribution of multimedia sessions across several user devices [TB08, SSTK07] but also the relaying of computation intensive tasks to more powerful machines [YOC08].

- Context Management: All devices belonging to a virtual device can cooperate to capture and process context information. Reasoned by the heterogeneity of devices and, thus, the availability of different sensors, more context information can be gathered [BOJ+06].

To achieve these goals all solutions have to address common challenges. These challenges are device discovery, trust establishment between devices and the exchange of device capabilities. This thesis considers device and service discovery as problem that has been comprehensively addressed in literature. Section 5.4.5.4 provides an overview on existing approaches. Most solutions do not address the establishment of trust, i.e. security associations, between devices. Therefore, this thesis presented a solution for this issue. Regarding the establishment of security associations, Section 5.4.5.2 discusses related work. With respect to the description to device capabilities Section 5.4.5.3 provides additional details. The application of the virtual device concept with respect to multi-device IdM is novel and has not been addressed in literature.

5.4.5.2 Security Associations

Section 5.4.2 introduced a concept to establish security associations between devices. The established security associations are valid within the virtual device, i.e. any two devices of the virtual device are able to establish security associations. Two categories of related work are relevant here: Device pairing and the set/group membership problem.

- Device Pairing: Device pairing describes methods and protocols to establish security associations between two devices. It does not address the establishment of security associations between groups of devices. Figure 5.22 provided a solution to pair devices and integrate them into the virtual device. This device pairing solution can be extended with more sophisticated device pairing methods. [KSTU09, SVA09] provide surveys of device pairing methods.

- Membership Problem: Closely related problems are the set membership problem [ST06] and group membership problem [FB01, Ric92]. Both problems are relevant for distributed computing and closely related to each other. The set membership problem addresses the following: A set of processes maintain and agree on the content of a dynamically changing set of arbitrary elements. Whenever the set changes, all processes get notified and have a consistent view on the set. With the group membership problem a set of processes maintains a list of processes running on a distributed system, which is the major difference to the set membership problem. The group membership problem is considered as not solvable, because of the consensus problem [FB01]. Not solvable means that there is no algorithm that always provides a solution. Solutions to the group membership problem only exist under randomization or probability assumptions [FB01]. The virtual device concept proposed in this thesis does not need such a strict view as the one imposed by the group membership problem. In addition, it is possible to include the user in case of uncertainties.

5.4.5.3 Device Description

For the description of device capabilities, several standards exist. CC/PP [w3c04] as well as OMA User Agent Profile [OMA06] focuses on the adaptation of content for optimal user experience on user devices. An alternative initiative is WURFL [WUR11], which provides open access to device descriptions. All proposals are not adequate for our scenarios, since they neither describe the usage context of a device nor its security capabilities.

5.4.5.4 Device Discovery

Academia and industry proposed and implemented various device discovery protocols. In the following there is no differentiation between device discovery and service discovery protocols made. Figure 5.26[18] classifies existing protocols into three categories.

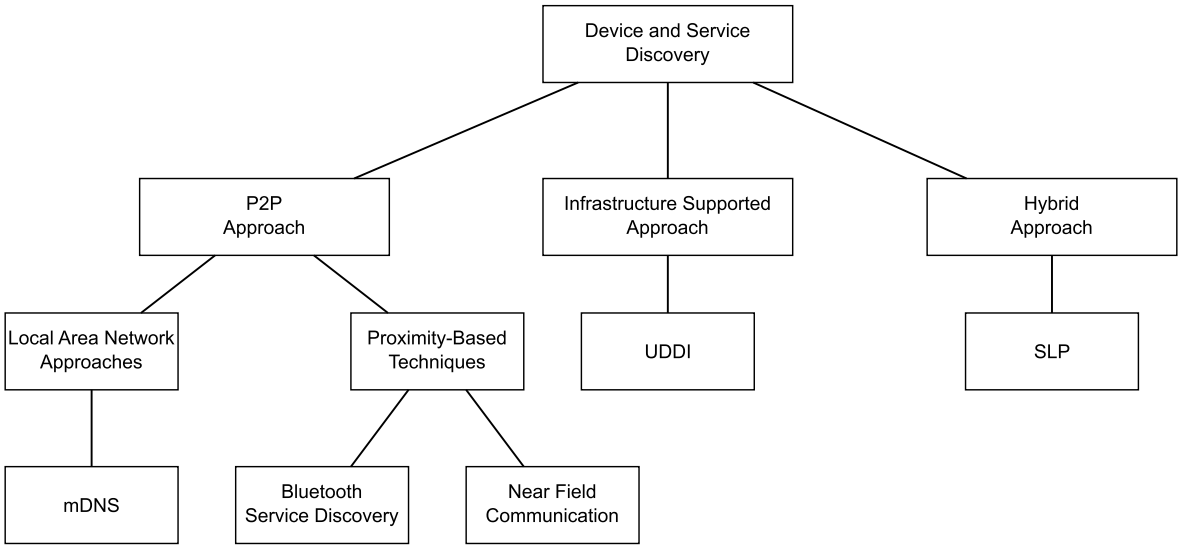


Figure 5.26: Classification of Device Discovery Approaches

- Infrastructure Supported Approach: Protocols in this category rely on the existence of a directory service that provides information about the available services and how the services can be accessed (e.g. [UDD04]).

- P2P Approach: Protocols in this category do not rely on the existence of a directory [CJYF06]. Each device is able to discover every other device in a peer to peer fashion. One candidate that is destined for the usage in Local Area Networks (LAN) is Zeroconf and mDNS [CK11] protocol. For devices that are close to each other there are a couple of protocols that are based on proximity detection. For example the Bluetooth Service Discovery Protocol (SDP) [SH00] is only able to detect devices that are in the surrounding.

- Hybrid Approach: A couple of protocols are able to use directories and P2P discovery either mutually exclusive or combined at the same time (e.g. Service Location Protocol (SLP) [RFC2608]).

For a detailed comparison of existing device discovery algorithms it is referred to [ZMN05, VP08]. This thesis assumes that a device discovery protocol is in place that can discover all devices belonging to a virtual device. In addition it can distinguish between devices that are in the proximity of each other and devices that are available but not in reach.

5.4.5.5 Secure Storage

Various techniques exist for the secure storage of data. Basically one can distinguish between hardware and software solutions.

- Hardware Solutions: Hardware solutions provide dedicated hardware for the secure storage of data. The most prominent example are smart cards. Smart cards provide a limited

[18]The figure makes no claim to be complete with respect to the mentioned protocols.

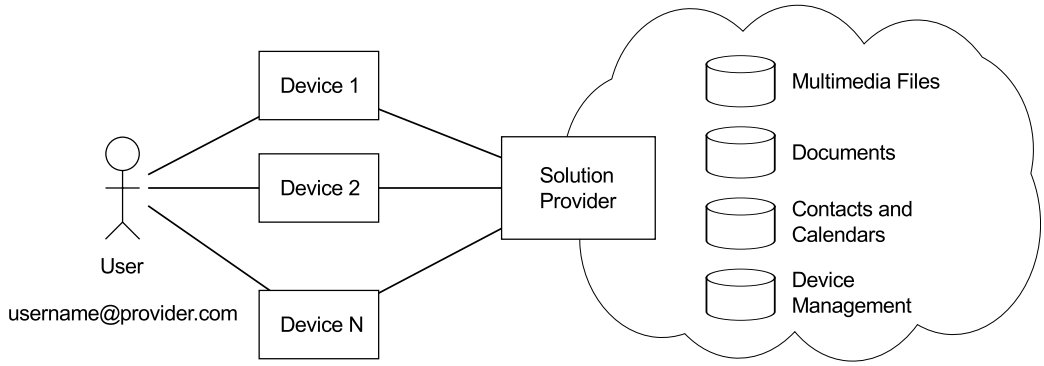


Figure 5.27: Device Management in the Context of Cloud Computing

interface to access the information only after successful authentication (e.g. by means of a personal identification number (PIN)) [KN07, LSW10]. Depending on the actual realization the effort to break hardware solutions is high [NPSQ03].

- Software Solutions: Software solutions rely on the encryption of data to protect sensitive information. For the decryption of information adequate keys are required. One can differentiate between the encryption of the data itself or the usage of encrypted storage.

5.4.5.6 Cloud Services and Device Management

Recently, commercial solutions [iCl, Fun10, Zha11, Ama11] appeared that integrate user devices by exploiting cloud technology. The overarching goal of the solution providers is the establishment of new ecosystems in order to bind customers. Users benefit by having the same user experience on all devices, i.e. having the same content and the same services available.

Figure 5.27 illustrates a typical scenario. The user has registered all his devices with the solution provider using one account. This enables him to synchronize purchased multimedia files (audio and video), personal documents, contacts, and calendars. In addition, the solution provider provides functionality for device management. This might include the installation of applications (also known as apps), remote locking and deletion of data (remote wipe) as well as modification of the device configuration.

Existing solutions can be classified according to the following criteria:

(1) Supported Hardware: The main focus of all solutions is on mobile devices, i.e. in particular smart phones, and their cooperation with fully-fledged computers (notebooks/computers). In addition, tablets and music players might be considered.

(2) Degree of Completeness: Existing solutions have various degrees of completeness depending on the solution provider. One can differentiate between the following three models: 1) Application: The solution is based on the installation of an additional client that has to run on the device. Examples for this approach are Amazon or Funambol. 2) Operating System: The solution is more complete, if the solution provider designs the operating systems. Examples are Google with its Android platform and Microsoft with Windows. 3) Hardware: The most

comprehensive solution is achieved by having everything including the hardware design under control. Currently only Apple and partially Google, have their hardware under full control.

(3) Closeness of Ecosystem: Almost all ecosystems have in common that they are based on a walled garden model. This is in particular achieved by proprietary protocols that are used to integrate devices. Therefore, it is required to have all devices from the same brand (i.e. Apple), having the same operating system (i.e. Google) or having proprietary client software.

Table 5.4 compares four different commercial solutions according to the above introduced categories. Regarding the terminology introduced before, the solution provider represents a SP. The offered service is the management of user's devices and user's data. Solution providers do not offer services (e.g. Single Sign On functionality or attribute provisioning) that qualify them as IdPs. The authentication towards the solution provider is based on a user account. Authentication against the solution provider cannot be reused for other SPs. In contrast to the above introduced solutions, the virtual device concept does not depend on any solution provider. It is open with respect to the system architecture and thus open with respect to different system platforms. Moreover, it does not depend on a centralized instance like the cloud.

Table 5.4: Comparison of Cloud-based Multi-Device Solutions

Provider and Product	Supported Hardware	Degree of completeness	Openness of ecosystem	Offered Services
Apple iCloud [iCl]	Computer, Tablet, Smartphone, Music Player	Hardware, Operating System, Application	Closed	Storage of personal documents, Management of multimedia files, Synchronization of calendars and contacts, Device management
Google Android [Zha11]	(Computer), Tablet, Smartphone, (everything on which Android OS runs)	Operating System, Application	Open with respect to hardware and competitors (e.g. alternative market applications possible) Standardized protocols for contact and calendar synchronization (ActiveSync)	Management of multimedia files, Synchronization of calendars and contacts, Device management
Amazon Cloud Drive [Ama11]	Computer, Tablet, Smartphones (Android and IOS)	Application	Open, purchased multimedia files can be exported to other devices and applications	Management of multimedia files

Continued on next page

Table 5.4: Comparison of Cloud-based Multi-Device Solutions

Provider and Product	Supported Hardware	Degree of completeness	Openness of ecosystem	Offered Services
Funambol [Fun10]	Computer, Tablet, Smartphones	Application	Standardized protocols for address and calendar synchronization	Storage of personal documents, Management of multimedia files, Synchronization of calendars and contacts, Device management

5.5 Placement of Multi-device IdM Functionality

Figure 5.3 showed the different activities that take place in the state VD Operation in order to consume a service. Some of the activities are bound to one device. The activities Service Selection, SP Assertion Acquisition and Service Consumption are bound to the device on which the service is supposed to be consumed. This device is called service consuming device in the following.

The other activities, i.e. Identity Filtering, Identity Selection and Identity Activation, provide degrees of freedom regarding their placement. In the following, this thesis provides an overview on the different placement possibilities in Section 5.5.1. Finally, Section 5.5.2 introduces an algorithm for the selection of an appropriate placement of functionality.

5.5.1 Overview Placement Possibilities

Table 5.5 gives an overview on prerequisites for the activities and derives corresponding placement possibilities. It is assumed that three devices belong to the virtual device. One of the devices is the master device. The other devices are slave devices. One of the slave devices is the service consuming device. In such a setting, the various placement possibilities can be enumerated reasonably.

Table 5.6 enumerates all combinations based on the placement possibilities in Table 5.5. It provides for each combination an explanation that discusses advantages, disadvantages and resulting requirements. The following notation is used:

- 0: The activity takes place on the master device.
- 1: The activity takes place on the service consuming device, which represents a slave device.
- 2: The activity takes place on any other slave device.

Table 5.5: Overview on Placement Possibilities

Activity	Prerequisite and Possible Placements
Final Filtering	Prerequisite: Requirements of SP regarding needed attributes and requested authentication methods are needed. Possible Placement: - Master Device: The master device performs phase 2 of the identity filtering process, i.e. Final Filtering, based on the information contained in the previously established Identity-Device-Matrix. For the Final Filtering is has to obtain the SP's requirements regarding attributes and authentication methods from the service consuming device. - Service Consuming Device: The service consuming device has obtained a restricted view on the Identity-Device-Matrix and is able to perform the Final Filtering on its own. - Slave Device: A 3rd slave device is not an option for the Final Filtering.
Identity Selection	Prerequisite: A prioritized list of identities has been created that can be used with the selected service on the service consuming device. The device on which the identity selection takes place has to have a user interface that is sufficient for selecting an identity. That means an adequate HCI is needed. Possible Placement: - Master Device: The user selects the identity on the master device. The master device is assumed to have in all cases an adequate HCI. - Service Consuming Device: The user selects the identity on the device on which the service is going to be consumed. This is only possible if the service consuming device has an adequate HCI. - Slave Device: A 3rd slave device is not an option for Identity Selection.
Identity Activation	Prerequisite: The user has selected an identity that should be used with the selected service. Identity activation requires that appropriate authentication mechanisms are available on the given device. In addition, the security level of the devices has to be taken in to account. Possible Placement: - Master Device: The user authenticates against the IdP on the master device. The master device is assumed to provide an adequate security level. - Service Consuming Device: The user authenticates against the IdP on the service consuming device. - Slave Device: In addition to the master device and to the service consuming device, it is possible to use any other slave device that is part of the virtual device. Such a decision is useful to meet requirements on authentication methods and security levels.

Table 5.6: Placement Combinations

Case Number	Final Filtering	Identity Selection	Identity Activation	Explanation
1	0	0	0	All activities take place on the master device. That means the requirements of the service provider have to be transported to the master device.
2	0	0	1	This combination is only useful, if the service consuming device supports authentication methods that are not implemented by the master device. In addition the same as for case 1 holds.
3	0	0	2	The authentication takes place on another slave device that supports the requested authentication methods. In addition the same as for case 1 holds.
4	0	1	0	An already filtered and prioritized list of useable identities is delivered to the service consuming device, which performs the identity selection. The authentication takes place on the master device. This combination is reasonable, because authentication takes places on the master device, which is more secure than the service consuming device.
5	0	1	1	In contrast to case 4, the authentication and the identity selection takes place on the service consuming device. This case is considered to provide a high degree of usability, since the authentication and the identity selection takes places on the same device.
6	0	1	2	In contrast to case 4 and 5, a third slave device is responsible for the identity activation. This is useful, if a dedicated authentication method is required that is neither supported by the master device nor by the service consuming device.
7	1	0	0	This combination is not useful, because the master device possesses the knowledge on the identity-device matrix, provides an adequate HCI and supports the required authentication method. Thus, the final filtering can also take place on the master device.
8	1	0	1	This combination is not considered to be useful. Compare case 7.
9	1	0	2	This combination is not considered to be useful. Compare case 7.
10	1	1	0	The identity filtering and the identity selection takes place on the service consuming device. The identity activation takes place on the master device, which is reasonable from a security point of view.
11	1	1	1	All activities take place on the service consuming device. From the usability perspective this case is considered as reasonable. This case is only valid, if the service consuming device provides the same security level as the master device.
12	1	1	2	The filtering and the identity selection takes place on the service consuming device. The service consuming device does not support the required authentication mechanism and relies for the identity activation on a 3rd slave device. This case is reasonable.

5.5.2 Selection of Placement

Table 5.6 excludes case 7, 8, and 9. These cases are not reasonable because there is no need to perform the Final Filtering on the service consuming device, whereas the Identity Selection takes place on the Master Device.

For the remaining reasonable cases, an algorithm has to be designed that orders the cases according to usability and security criteria. Three different aspects can be distinguished that make one case more usable than others:

- Human Computer Interface: A device has to provide an adequate HCI to perform identity selection and depending on the required authentication mechanism the corresponding identity activation.
- Number of Devices: The number of involved devices and the number of device switches should be limited. Identity selection and authentication should take place on the same device. Switching devices to perform activities belonging to the same context is considered as less usable.
- Understandable Patterns: It is preferable, if the number of cases is limited to simplify the usage and make the used patterns obvious to the user.

Security means that data processing in general and authentication in particular should only take place on secure devices. Therefore, it is preferred to perform Final Filtering, Identity Selection and Identity Activation on the most secure device. Since the master device is assumed to be one of the most secure devices that are part of the virtual device, it should be preferred to perform all operations on the master device.

Figure 5.28 extends Figure 5.3 by considering the placement of the above discussed activities according to the usability and security criteria. Three aspects have to be fulfilled by the service consuming device in order to not relay the final filtering, the identity selection and the authentication to the master device.

First, the service consuming device has to have an appropriate HCI. Second, an appropriate security level is required. Hereby, it is assumed that the user of the device can define a threshold that has to be reached by the service consuming device. Third, the required authentication mechanism, which is specified by the service itself, must be available. If one of these three conditions is not fulfilled, everything is relayed to the master device.

If all conditions are fulfilled, the steps of identity filtering and identity selection take place on the service consuming device. After all depicted activities have been finished, the control flow continues as in Figure 5.3 specified.

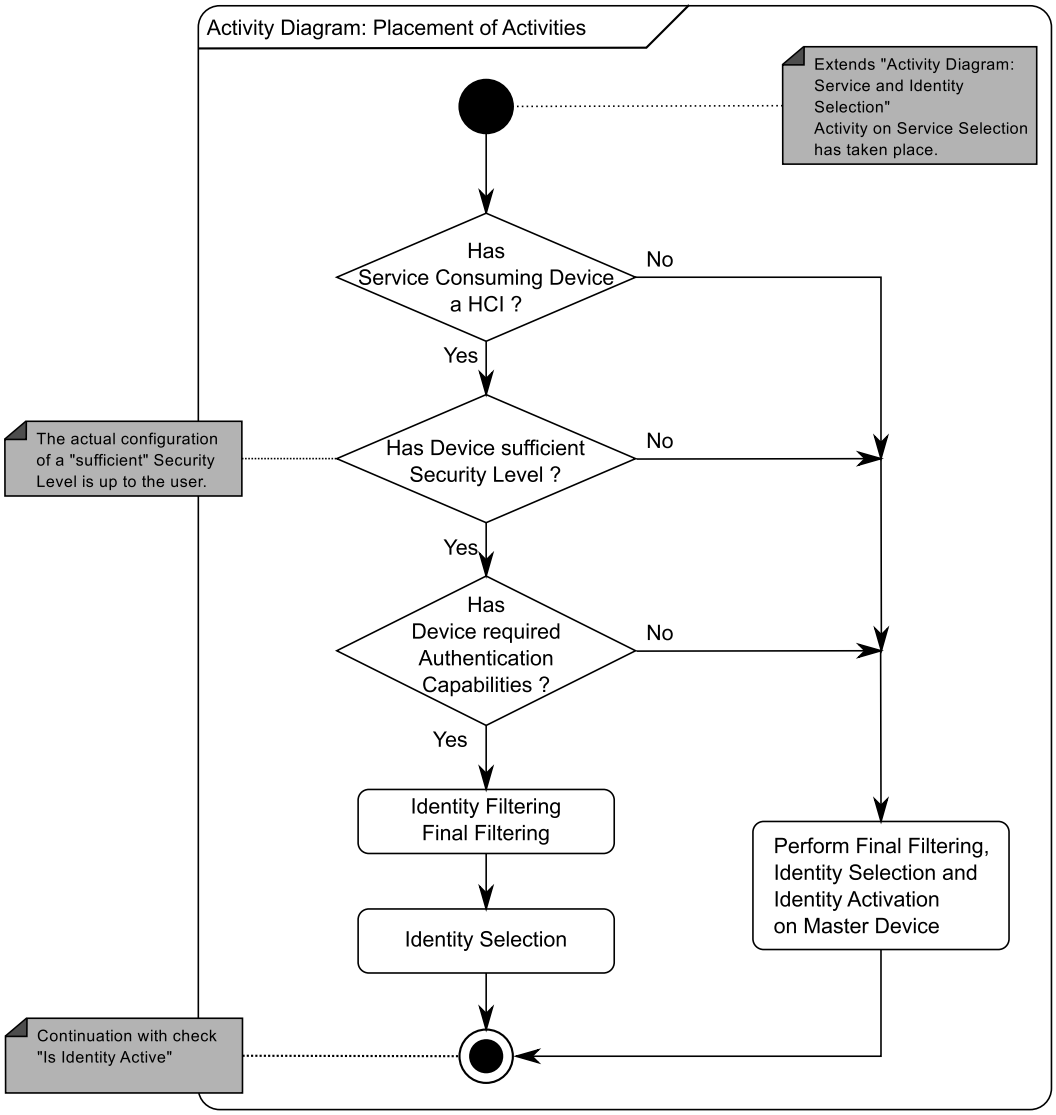


Figure 5.28: Placement Algorithm

6 Evaluation

Chapter 4 outlined the functional architecture for multi-device IdM and Chapter 5 detailed algorithms, mechanisms, and protocols. This chapter evaluates the proposed concepts from three different perspectives as illustrated in Figure 6.1. The perspectives are functionality, security, and performance.

In Section 6.1 the functional evaluation examines whether the required functionality is provided. This is achieved in three steps. First, the scenarios are considered regarding their feasibility. Second, the high-level, the functional, the non-functional and the security requirements are systematically checked. Third, the interworking of the architecture with an existing IdM system is evaluated by means of a prototype.

Since security has been one of the major design goals, Section 6.2 performs a security evaluation. In addition to the checking of security requirements that took place in Section 6.1, three different views are considered. First, an internal security evaluation is performed with full system knowledge applying the same methodology as in Section 4.4. Second, an external security evaluation takes the view of an attacker and evaluates attacks with different objectives. Finally, misuse cases consider use cases beyond the initial design goals.

Section 6.3 evaluates the architecture regarding the authentication effort for the user and regarding the signaling effort between the virtual device and the IdP. The performance analysis is conducted by an analytical model based on Markov chains. Finally, Section 6.4 summarizes the results of the three different evaluation perspectives and puts them into relation to each other.

6.1 Functional Evaluation

The functional evaluation consists of three parts. In Section 6.1.1 an example setting is defined to review whether the usage scenarios from Section 4.1 can be realized. Even if various sections in Chapter 4 and Chapter 5 have individually outlined the addressed requirements, it is required to systematically check all requirements regarding completeness in Section 6.1.2. Finally, it is necessary to show that the designed architecture is usable with an existing IdM system. Therefore, Section 6.1.3 introduces a prototype that is able to interact with the Shibboleth IdM system.

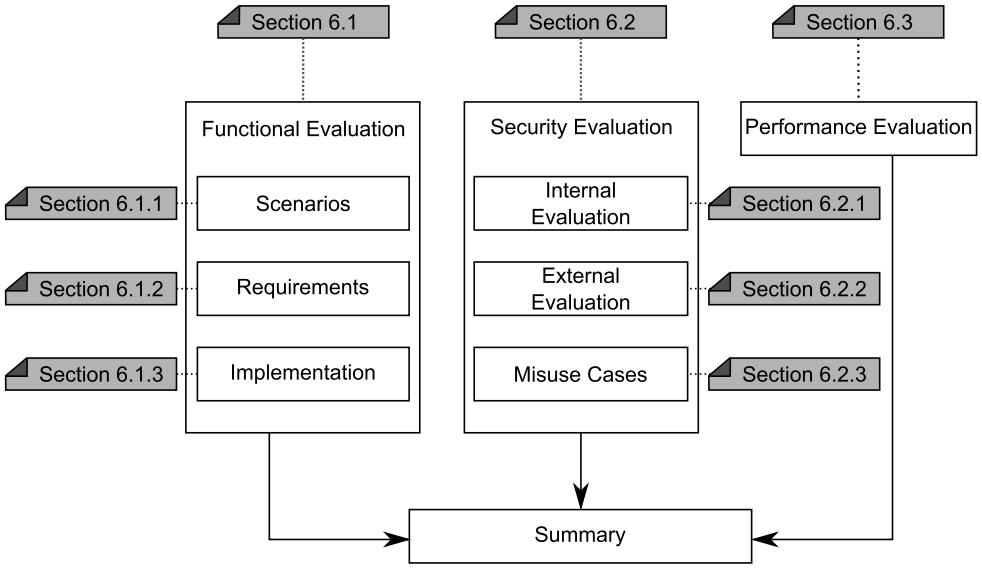


Figure 6.1: Chapter Outline

6.1.1 Evaluation of Scenarios

In Section 4.1 five different usage scenarios have been described to motivate the use cases that the designed system should address. To evaluate the functionality of the architecture with respect to the usage scenarios a concrete setting of the virtual device is assumed. Section 6.1.1.1 describes this concrete setting, which is defined in a way to cover a broad range of situations. Based on this setting, the five usage scenarios are evaluated regarding their basic feasibility, the applied mechanisms, and potential weaknesses. For one scenario, Section 6.1.1.2 presents the evaluation in more detail. The evaluation results for the other four scenarios are summarized in Section 6.1.1.3.

6.1.1.1 Evaluation Setting

Figure 6.2 illustrates a sample setting for the evaluation of the usage scenarios. It has the following characteristics:

- One user with one virtual device. The virtual device consists of four devices:
  - Smartphone: The smartphone is assumed to be a secure device that is always with the user. It is owned by the employer of the user, who allows private usage. It has master device capabilities.
  - TV Set: The TV set has only limited input capabilities, but a huge display to consume services. The TV set is owned by the user. It has no master device capabilities.
  - Notebook: The notebook is assumed to be a secure device, without master device capabilities. It is owned by the company and must not be used for private services.
  - Desktop PC: The Desktop PC is old and not sufficiently administrated. That means it has a very low security level. It is owned by the user and has no master device capabilities.

It is assumed that all devices have installed the designed middleware and that the virtual device is bootstrapped, i.e. security associations exist. Moreover, all devices are available and have discovered each other.

- Two federations exist:
  - Company Federation: The employer acts as an IdP called IdP_C for its employees. Within this federation are internal SPs[1]and external SPs. The internal SPs, e.g. the company wiki, have security requirements regarding authentication whereas the external SPs, like the Virtual Class Room Provider that offers professional training, have less security requirements.
  - Public Federation: A public IdP IdP_P has established a federation with different SPs. Among the SPs are a Webmail Provider that has very high security requirements[2], a Digital Notepad Provider, and the aforementioned Virtual Class Room Provider.
- The user has two identities:
  - Company identity Id_C: Usable within company federation, i.e. issued by IdP_C. The associated usage context is "Business".
  - Private identity Id_P: Usable within public federation, i.e. issued by IdP_P. The associated usage context is "Private".

6.1.1.2 Evaluation of Scenario 3: Identity Usage on Insecure Devices

Scenario Mapping: The scenario points out (→ Section 4.1.3) a situation in which the user intends to use an insecure device for authentication against a webmail provider. Such a situation is avoided by the virtual device. The Desktop PC is assumed to be insecure. Therefore, the smart phone should be used for authentication. The private identity Id_P is not active, i.e. no IdP session exists on any device that is part of the virtual device.

Scenario Work Flow: The user decides to check his emails with the Desktop PC and selects the service provided by the Webmail Provider. The Webmail Provider requests a SP Assertion for authentication. Since the Desktop PC has a low security level, the *IM* on the Desktop PC forwards the request by means of the ARP to the smartphone. The user is informed by a message on the display of the Desktop PC that the process is continued on the smartphone. The smartphone evaluates the Webmail Provider's request and provides a list of useable identities. In this case, the list has the length one and only identity Id_P is shown. The user selects this identity and performs the authentication with IdP_P. After successful authentication, the smartphone obtains an SP Assertion for the Webmail Provider and forwards it within the ARP reply to the

[1]The term SP is used to highlight that the services offered by the company might be under different administration, e.g. administrated by different departments.

[2]In reality, webmail providers often have very low security requirements regarding the user authentication and the attestation of user identities. As a consequence, many webmail accounts got compromised resulting in the potential compromisation of further accounts. The compromisation of further accounts is enforced by the circumstance that the email account can be used to reset other accounts.

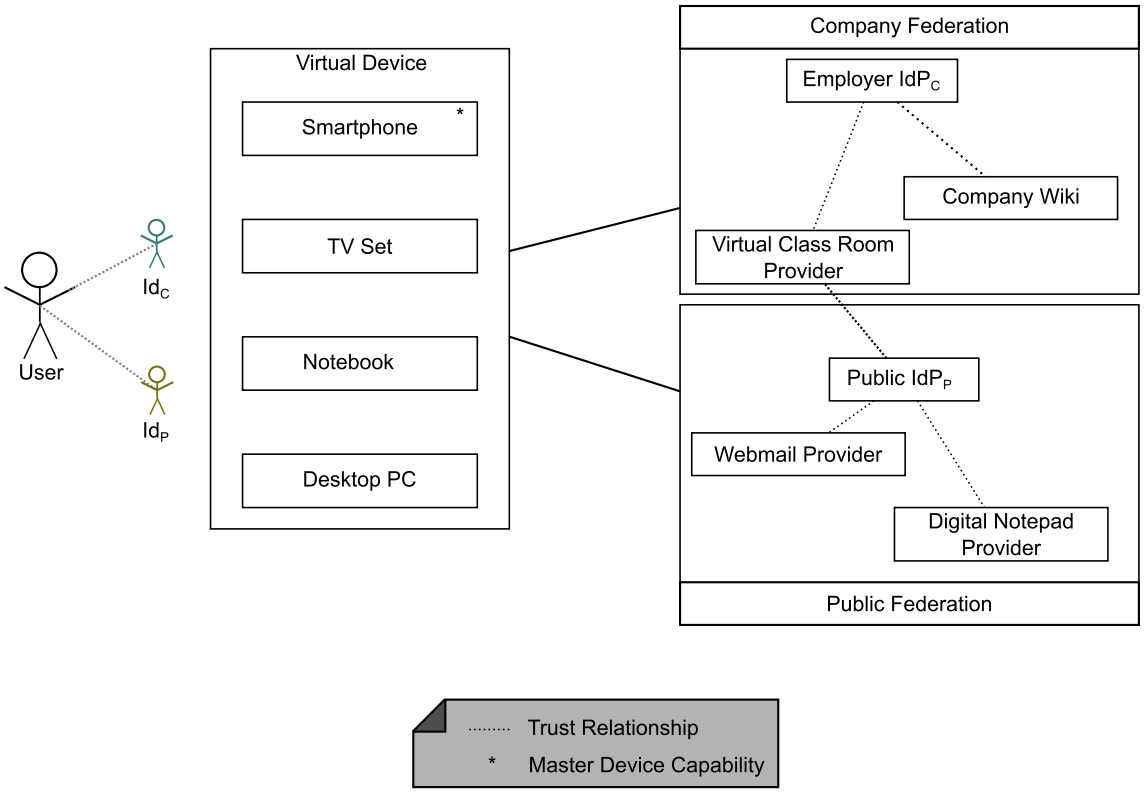


Figure 6.2: Base Scenario for the Usage Scenario Evaluation

Desktop PC. The Desktop PC uses the obtained SP Assertion for authentication against the Webmail Provider and starts a service session.

Involved Activities: The following activities take place in the scenario work flow:

- Master Device Selection: Upon mutual discovery of the four devices, the smartphone succeeded with the master device negotiation and became the master device. It triggers the information exchange process (→ Figure 5.18). For this process secure channels are established between the master device and the three other devices.
- Identity Filtering: Identity Filtering runs on the smartphone and is performed in two steps. After the smartphone became the master device, it gathered information from all other devices regarding usable identities and device characteristics. This information is used for the Prefiltering step of the identity filtering. Upon reception of the SP request for a SP Assertion, the Final Filtering step is triggered. The Final Filtering evaluates the information provided in the request of the SP. Based on this information IdP_C is excluded.
- ARP: The Desktop PC forwards all requests regarding authentication to the master device independently of the request itself. ARP is used by the Desktop PC to request the SP Assertion and used to provide the corresponding assertion. ARP makes use of a secure channel.
- Request Authorization: The Smartphone has to authorize the ARP request. This is based on the presented device identity, which has to adhere to the device identity used to es-

tablish the secure channel, based on the requested identity and based on the requested service.

Conclusion: The challenge imposed by Scenario 3, i.e. the authentication should take place on the most secure device, can be fulfilled.

6.1.1.3 Summary of Remaining Scenarios

Each of the following section elaborates how the challenges imposed by Scenario 1, 2, 4, and 5 are addressed.

Scenario 1 – Business and Private Devices: The challenge resulting from scenario 1, i.e. the consideration of the usage context, can be realized. The Final Filter step of the filtering process considers the usage context of identities and devices. In addition, the filtering process takes the SP identity into account. Therefore, it is possible to restrict the usage of identities, devices and services according to the defined policies.

Scenario 2 – Fast "Device Change": The challenge to use services without the need to re-authenticate against an IdP is covered. ARP provides the required mechanism to transport the necessary authentication. Not covered is the transport of the service state to enable session mobility. Appropriate mechanisms are proposed in [BKM09].

Scenario 4 – Insufficient Security Features: The challenge to relay the authentication to a device that supports the required authentication methods is met. With the Prefiltering step of the filtering process, device properties and identity properties are matched. IAP and ARP allow the handover of the authentication and the request of corresponding assertions to a capable device.

Scenario 5 – Insufficient Input Methods: The challenge to relay the authentication to a device with adequate input methods is addressed by the same set of mechanisms that are required to fulfill Scenario 4.

6.1.1.4 Summary

In conclusion, all 5 scenarios are feasible with the functional architecture and the corresponding mechanism, algorithms, and protocols.

6.1.2 Evaluation of Requirements

Chapter 4 set up high-level requirements (→ Section 4.1.7), functional requirements (→ Section 4.3.3), non-functional requirements (→ Section 4.3.4), and security requirements (→ Section 4.4.5). These requirements have been covered by the functional architecture in Section 4.5 and by the mechanisms, algorithms and protocols in Chapter 5.

The sections enumerated in Table 6.1 and the corresponding tables out of Section 4.5 and Chapter 5 elaborated how these requirements have been met:

Table 6.1: Requirements Addressing Sources

Section	Table	Functional Component or Mechanism
Section 4.5.2.4	Table 4.15	*Identity Manager*
Section 4.5.3.4	Table 4.17	*Identity Transfer Enabler*
Section 4.5.4.4	Table 4.19	*Secure Storage Enabler*
Section 4.5.5.4	Table 4.21	*Device Manager*
Section 5.2.3	Table 5.1	Identity Filtering
Section 5.3.5	Table 5.2	Protocols for Multi-Device IdM
Section 5.4.3	Table 5.3	Virtual Device Management

The functional architecture in Section 4.5 addresses requirements in a different way than the mechanisms, algorithms and protocols specified in Chapter 5. Chapter 5 provides concrete solutions for the requirements, whereas the functional architecture rather assigns requirements to functional blocks. Nevertheless it is required to systematically verify whether all requirements have been sufficiently covered.

6.1.2.1 Evaluation Process

The evaluation process, which has been defined within this thesis, consists of four phases: Compilation, Categorization, Consistency Check, and Discussion. The compilation phase creates a compilation table for each requirement type, i.e. high-level requirements, functional requirements, non-functional requirements, and security requirements. A compilation table combines the tables that elaborate the covered requirements referenced in Table 6.1 with the complete lists of requirements contained in Section 4.3. The compilation tables (e.g. Table 6.2) show if the requirements have been covered. If a requirement is covered, a reference is provided to one of the tables that detail how the requirement has been covered. The tables distinguish whether the requirement has been covered by the functional architecture (in the following called functional solution level), by a concrete concept in Chapter 5 (in the following termed as concrete solution level), or both.

The categorization phase assigns each requirement to one category. Four different categories are distinguished based on the following three factors: (1) Requirements that have not been covered at all. (2) Requirements that have been covered by the functional architecture, i.e. functional solution level. (3) Requirements that have been covered by concrete concepts in Chapter 5, i.e. concrete solution level. These three factors lead to the following four categories, to which a requirement is assigned to:

- Category 1 – Only (1): The requirement is not covered. Reasons have to be provided why the requirement has not been covered.

- Category 2 – Only (2): The requirement is only covered on the functional solution level. It has to be checked, whether it was intentionally not covered on the concrete solution level.

- Category 3 – Only (3): The requirement is only covered on the concrete solution level. This case is not valid and means that the requirement has been implicitly covered by the functional architecture.

- Category 4 – (2) and (3): The requirement has been considered on the functional solution level as well as on the concrete solution level.

The consistency check phase verifies, whether it is reasonable that the requirement has been covered by the corresponding functional block or by the concrete concept. Functional blocks and concrete concepts can be distinguished based on references to previous sections. Additional consistency checks are applied for security requirements (→ Section 6.1.2.5).

In the discussion phase, requirements are elaborated that require clarification. Clarification is required, if the categorization leads to unreasonable results or if consistency checks fail.

This process is applied for the high-level requirements in Section 6.1.2.2, for the functional requirements in Section 6.1.2.3 and with slight extensions for the security requirements in Section 6.1.2.5. Since the non-functional requirements have not been explicitly addressed before, Section 6.1.2.4 evaluates them one-by-one.

6.1.2.2 High-Level Requirements

Table 6.2 enumerates the high-level requirements from Section 4.1.7 and shows if the requirement has been covered on the functional solution level or on a concrete solution level. If it has been covered, a reference to one of the previous sections is provided.

Table 6.2: Evaluation of High-Level Requirements

		Functional Solution Level		Concrete Solution Level	
No	Short Name	Cov.	Reference	Cov.	Reference
R1	Secure Exchange	Yes	Table 4.21	Yes	Table 5.3
R2	Task Distribution	Yes	Table 4.15, Table 4.21	Yes	Table 5.1, Table 5.2
R3	Remote Activation	Yes	Table 4.17	Yes	Table 5.1, Table 5.2,
R4	Discovery of User Devices	Yes	Table 4.21	No	
R5	Capture of Device Characteristics	Yes	Table 4.21	Partially	Table 5.3
R6	Establishment of Security Assocations	Yes	Table 4.21	Yes	Table 5.3
R7	Determination of Usage Context	Yes	Table 4.15, Table 4.21	No	
R8	Distributed Data Handlung	Partially	Table 4.15, Table 4.21	Yes	Table 5.2, Table 5.3

Table 6.2 shows that R4 and R7 have only been covered by the functional architecture. The discovery of devices (R4) is not considered as part of this thesis[3]. The determination of the usage context (R7) has not been concretized, because it is strongly related to the provisioning of corresponding graphical user interfaces. Also no concrete mechanism for the capturing of device characteristics is provided to detail R5. It is only touched by Section 5.4.2.4. The statements regarding R8 are inconsistent. On the level of the functional architecture it is considered as only partially covered, whereas on the concrete solution level two references are provided. This is reasoned by the fact that the originally planning favored a central data handling service. For simplicity reasons and due to the heterogeneity of device information and identity information two different mechanisms have been realized.

In conclusion, it can be stated that all high-level requirements are sufficiently covered.

6.1.2.3 Functional Requirements

The same process as for the high-level requirements is applied for the functional requirements. Table A.1 shows the results.

For the requirements related to device discovery (DM-DD-1, DM-DD-2, DM-DD-3), the same holds as already stated in Section 6.1.2.2. This thesis does not provide a solution for secure storage (DM-SA-2), potential candidates are enumerated in Section 5.4.5.5. Details on the realization of device identifiers (DM-VDM-3) are not required. A simple, unique string is sufficient ($\rightarrow$ Figure 4.23). Requirements (DM-DC-1, DM-DC-2, DM-DC-3, DM-DC-4) regarding device characteristics and corresponding data models are not substantiated beyond the functional architecture. With the exception of Section 5.4.2.4 that covers aspects on the security level determination and Section 5.4.5.3 that introduces related work on devices description methodologies no more details are necessary. For the data exchange mechanism (DM-DA-1) the same holds as for R8 in Section 6.1.2.2.

IdM-IM-1, IdM-IM-2, IdM-IM-3, IdM-IM-4, and IdM-IM-5 target the capturing and modification of information about identities. This is not detailed due to two reasons. First, modifying information is coupled to the provisioning of user interfaces. Second, the capturing of identity information requires application interfaces. Both aspects are not in the core of this thesis and therefore excluded. IdM-IA-3 and IdM-AE-5 can be easily realized by a triggering function that acts on events caused by the IAP. Therefore, a detailed consideration was not necessary.

It can be stated that all functional requirements have been sufficiently covered.

6.1.2.4 Non-Functional Requirements

The non-functional requirements are not verified by the same process as the other requirements, because they have not been covered by concrete functional mechanisms. In the following, each non-functional requirement is discussed one-by-one.

[3] A comprehensive overview on related work is provided in Section 5.4.5.4.

NF-1: The devices within the virtual device composition are not equal, i.e. the devices obtain different information. Only master devices obtain a complete view on identities and device characteristics. Devices without master device capability have only a limited view on identities and devices. Thus, the data minimization principle is addressed. Additionally, the privacy ranking rules[4] within the Final Filtering step can be further detailed to propose identities that reveal only as much information as actually needed. Moreover, the coupling with additional access control mechanisms is feasible [BNP+08].

NF-2: Usability was a major design goal and has been covered in different ways. First, the system has to support the user with the management of identities and devices and not restrict him. That means that the user is able to influence the behavior of the system by corresponding configuration mechanisms and the possibility to override the system's proposals. Second, the system has to be comprehensible for the user. That means the user must be able to understand what happens why and when. This is achieved with the master device concept that assigns additional tasks to master devices and puts them more into the focus than other devices. In addition, the placement of functionality considers the usability (→ Section 5.5). Third, the number of authentication procedures that have to be performed should be limited. This is achieved by distributing the knowledge of existing IdP sessions and the retrieval of SP Assertions from remote devices. A detailed performance analysis is provided in Section 6.3.

NF-3: During the design of the system, performance was not considered as an issue that requires additional mechanisms due to the following reasons. First, the amount of information that is exchanged between the devices is limited. After an initial exchange of device and identity characteristics, the further exchange can take place incrementally. The amount of data exchanged by the IAP and the ARP is low. Second, the effort to process the exchanged information is also low. The complexity of the Prefiltering step of the Identity Filtering Process increases linearly with the number of devices, the number of identities, and with the number of rules. The Final Filtering algorithm is also of linear complexity. The complexity of all other algorithms and the effort to realize the security functionality is not considered to have a significant influence on the performance.

NF-4: Security was a major design goal and various technical and non-technical (e.g. manual user confirmation) mechanisms have been designed to make the multi-device IdM architecture secure. For further information about the overall system security it is referred to the evaluation of the security requirements in Section 6.1.2.5 and to the security evaluation in Section 6.2.

NF-5: The system does not depend on a single device. Even if the master device bundles different responsibilities, the system does not depend on it. First, it is possible to have several master devices. Second, the system is always in the position to fall back to single operation, i.e. each device operates on its own. With an additional mechanism that replicates the *SecStore*, e.g. on all master devices, it is possible to fulfill this requirement.

At this point it is not possible to fully confirm the fulfillment of all non-functional requirements. Therefore, it is required to consider the fulfillment again after the security evaluation and after the performance evaluation in Section 6.2 and Section 6.3, respectively.

[4]The actual realization of the privacy ranking rules have not been addressed by this thesis. For more details it is referred to [BNP+08]

6.1.2.5 Security Requirements

The evaluation of security requirements can be considered as part of the security evaluation. Since the same process as for high-level and functional requirements is applied, it is part of this section.

For the evaluation of security requirements an additional consistency check is applied based on the nature of the requirement. One can distinguish between two kinds of security requirements: Self-contained requirements and client/service requirements. A self-contained requirement is fulfilled, if the required characteristic is provided by one component. A client/server requirement is fulfilled, if the server provides the functionality and the client uses the functionality. For example in case of logging functionality, the server must provide the possibility to log data and a corresponding client must use the logging functionality.

Table A.2 shows how the security requirements have been addressed. In the following all security requirements are discussed that have not been adequately covered.

SR-1 requests a mechanism that prevents the unauthorized usage of devices. Such a mechanism is assumed to be available within at least master devices. If a device does not have such a mechanism, additional restrictions apply. For example, the device must not become a master device and manual confirmation is required upon ARP requests.

SR-10, SR-12, and SR-15 cover the transmission of Tokens and Credentials between the user devices and other parties. Therefore, it is not the sole responsibility of the user to take care of the application of appropriate mechanisms like encryption. A potential extension of the *IM-A* might restrict the transmission of tokens and credentials to secure channels.

SR-11, SR-13, SR-14, SR-18, SR-25, SR-27, SR-29, and SR-31 can be fulfilled with the provisioning of a secure storage. As stated above, the actual realization of the secure storage is out of scope of this thesis. SR-24 provides an additional requirement on the secure storage, which has to be considered with the actual realization of the secure storage.

SR-3, SR-8, SR-16, SR-19, SR-26, SR-28, and SR-30 are not considered in more detail. Logging functionality can be added to the corresponding message sequence charts. Such functionality is not contained in the diagrams of Chapter 5 for simplicity reasons.

For the requirements related to device discovery (SR-21 and SR-22) the same holds as with device discovery for the high-level requirements.

The requirements SR-100 to SR-103 are not explicitly addressed by the functional architecture. Nevertheless, they have been addressed by the ARP and IAP and are, thus, covered.

SR-23 demands encrypted device identifiers or the avoidance of unique device identifiers. This requirement has not been covered by the security architecture of the virtual device. Based on the assumption that only devices belonging to the same user can discover each other, only devices that belong to the same virtual device are in the position to establish a secure channel. The establishment of secure channels includes the exchange of certificates, which contain the device identifier. If a device cannot discover another device, no certificates are exchanged. Therefore, it is not required to fulfill SR-23.

The virtual device composition should be kept confidential (SR-9). Since information about the composition is not only stored within the virtual device, but also on *SecStore*, the fulfillment of SR-9 depends on the realization of the *SecStore*. Based on the assumption that the *SecStore* is part of the virtual device and based on the assumption that *SecStore* is secure, this requirement is considered to be fulfilled.

Requirement SR-6 has not been explicitly addressed by the defined multi-device IdM protocols. The requirement can be easily covered by the authorization mechanisms that are part of the IAP and ARP.

It can be stated that all security requirements have been covered. This does not imply any statement on the strength of the proposed mechanisms or the completeness. Security is considered in more detail by the security evaluation in Section 6.2.

6.1.2.6 Summary

The high-level and the functional requirements have been covered by adequate mechanisms. For the non-functional requirements and for the security requirements it is additionally required to consider the results of the performance and/or security evaluation, respectively.

6.1.3 Implementation

This section describes the motivation and implementation of the prototype. Basically it is possible to implement the specified architecture. However, the prototype serves as a proof-of-concept for the interworking with an existing IdM system. If the interworking is feasible, it would be guaranteed that the proposed architecture can be introduced without breaking mechanisms of existing IdM systems.

With respect to the interworking with existing IdM systems the following issues have to be clarified:

- SP Assertion Transfer:	Is it possible to transfer SP Assertions that are obtained from the IdP between devices?
- Requested Authentication Method:	How is it possible to obtain knowledge about the requested authentication methods?
- Supported Federations:	How is it possible to obtain knowledge about the supported federations, i.e. the supported IdPs?
- Requested Attributes:	How is it possible to obtain knowledge about the required user attributes?

The Shibboleth IdM system [Shi11b] has been selected to show the interworking out of two reasons. First, it is freely available and its source code is accessible. This is advantageous with respect to the clarification of the above introduced issues that cannot be necessarily derived from the corresponding specification. Second, the user community is large due to its prevalence in academia [Käh06, Shi11a]. Thus, a lot of experience is accessible to setup the Shibboleth IdM system.

Section 6.1.3.1 details the prototype structure and Section 6.1.3.2 illustrates the work flow of test scenario that has been realized. Finally, Section 6.1.3.3 summarizes the key findings of the implementation.

6.1.3.1 Prototype Structure

The prototype is restricted to the functional blocks that are required to clarify the issues raised above. The implementation of the individual functional blocks is restricted to the required functionality. The required functional blocks are:

- *IM*: Required to authenticate against the IdP and for the coordination with the *ITE*. The implementation of the *IM* comprises the *IM-C* and the *IM-A* together with a simple user interface. The *IM-IFS* and the *IM-DM* are not needed in this context.
- *ITE*: Required for the transfer of SP Assertions between the devices. The implementation of the *ITE* is restricted to the *ITE-C*. The *ITE-AM* and the *ITE-DM* are not needed in this context.
- *DM*: Required to establish secure channels. The implementation of the *DM* is limited to the *DM-VDM* and to the *DM-SCB*. Master device selection and device discovery is not needed in this context.

The *SSE* has not been implemented, because neither logging nor security storage functionality are required to show the interworking with an existing IdM system.

Figure 6.3 depicts the realized prototype. It consists of four virtual machines running Ubuntu 10.04 inside a virtualized environment [STM10]. Two machines depicted on the left realize the virtual device that is operated by the user. The IdP as well as the SP are running the unmodified Shibboleth IdM software [Shi11b]. The functionality of the IdP is realized as a servlet that runs inside a Tomcat application server [BD07]. The SP offers a service that is restricted to authorized users. The service is realized as a webpage by means of an Apache webserver [BC08] that runs the Shibboleth module *mod_shib* [Shi11b]. The SP and IdP have already established a security association by the exchange of X.509 certificates. The certificate information as well as the supported protocols that are applied between the IdP and SP are recorded as metadata with the IdP and SP.

6.1.3.2 Test Scenario

The user authenticates on Device 1 against the IdP and establishes an IdP session with the *IM*. The information about the existing session is transferred to Device 2 by the *ITE*. On Device 2 the user selects a restricted web page as a service with the local Web Renderer[5]. The SP requests the user to authenticate with an AuthNRequest that redirects the user to the IdP. Instead of establishing an IdP session on Device 2, Device 2 intercepts the redirection request of the SP

[5]The Web Renderer can be considered as a simplified version of a web browser and is provided as part of the JAVA SDK.

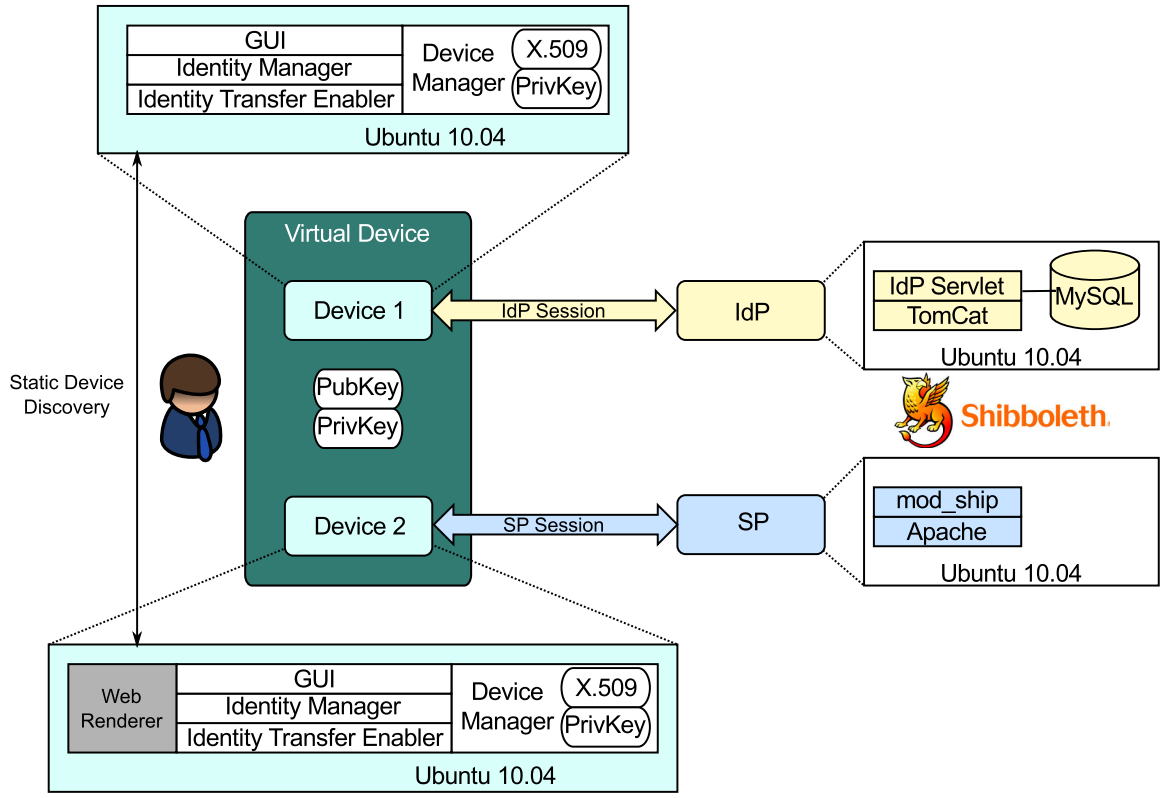


Figure 6.3: Conceptual View on Prototype

and requests an SP Assertion from Device 1. Device 1 retrieves the SP Assertion based on the already existing session with the IdP and delivers it to Device 2. Device 2 uses the SP Assertion for successful authentication and eventually consumes the service.

The AuthNRequest serves as a transport medium for the SP to inform the IdP about its requirements. Without modification of the configuration, the AuthNRequest is encrypted by the SP with the public key of the IdP. The interception and the subsequent processing of the AuthNRequest on Device 2, is only reasonable if the AuthNRequest is not encrypted. Therefore, the configuration was modified to transmit the AuthNRequest unencrypted, but integrity and authenticity protected. This does not represent a security threat, because TLS protects the confidentiality and the integrity of the AuthNRequest between the SP and the User, and between the User and the IdP. An unencrypted AuthNRequest allows the user to process the content of the AuthNRequest in order to obtain two pieces of information. First, it is possible to select an appropriate IdP, i.e. selecting an appropriate federation and thus limiting the set of potential identities. Second, the authentication context (→ Section 2.3.3) allows selecting an adequate authentication method that is appropriately supported by one of the devices out of the virtual device composition.

An extension of Shibboleth is the SAML Enhanced Client or Proxy profile (ECP). ECP explicitly supports the identity selection by the user. Thus, it provides an alternative solution to plaintext AuthNRequest. However, it requires explicit client support, which breaks the assumption of Shibboleth to operate with ordinary web browsers.

Shibboleth provides a mechanism to obtain meta-data about SPs and about IdPs. This meta-data can be used to detail the knowledge about the structure of the federation. Knowledge about the federations serves as input for the identity filtering process. Three basic approaches to obtain this knowledge can be differentiated. First, the SP supplies information on an endpoint to retrieve meta-data (e.g. an URL) with the AuthNRequest. The virtual device can retrieve the meta-data and identify suitable IdPs. Second, the virtual device can obtain meta-data from the IdP about all SPs that are part of the federation. This requires information about the endpoint. Shibboleth does not provide a mechanism how this could be achieved. That means extensions are required. Third, the WAYF service may serve as a source to obtain knowledge about the federations. All three approaches are basically feasible to support the identity filtering mechanisms. Approach 1 and 3 are considered as feasible without extensions to the Shibboleth system.

The test scenario did not address the need of the SP for user attributes. Even if Shibboleth supports the retrieval of user attributes by the SP directly from the IdP, it does not allow the expression of SPs requirements. However, knowledge of the required attributes improves the identity selection. Therefore, additional mechanisms are required to enable the selection of identities according to SP requirements. This is not further considered here. Solutions [BNP+08] are proposed based on privacy policy languages [KCLC07].

6.1.3.3 Summary

The implementation successfully confirmed the feasibility of the functional architecture in general and ARP and IIEP in particular. It was possible to show the interworking with an existing IdM system. However, extensions to the Shibboleth IdM systems regarding the provisioning of information on the required attributes and with respect to federations are required to take full advantage of the proposed architecture.

6.1.4 Summary

The functional evaluation consisted of three parts. The evaluation of the scenarios verified that the scenarios are feasible with the specified architecture. All requirements are sufficiently covered by the functional architecture. The implementation confirmed the basic feasibility, but pointed out necessary extensions to take full advantage of the multi-device IdM concept.

6.2 Security Evaluation

The security evaluation consists of four parts. The first part of the security evaluation is the verification, whether all security requirements have been covered (→ Section 6.1.2.5). The second part (→ Section 6.2.1) evaluates the security from an internal perspective. That means with the complete knowledge of the system design, (→ Chapter 4 and Chapter 5) assets beyond the ones in Section 4.4.2 are identified, potential threats are derived and if necessary additional mechanisms are proposed. The internal security evaluation is complemented by an external security evaluation in Section 6.2.2. The external evaluation is goal-oriented and evaluates the security of the designed system with particular attacks in mind. Eventually, the fourth part

studies so called misuse cases that regard the system behavior under use cases that are outside the system specification (→ Section 6.2.3). Section 6.2.4 summarizes the four parts.

Neither the internal nor the external security evaluation considers threats that are a consequence of the following causes:

- Availability of communication infrastructure: An attack on the availability of the communication infrastructure that is used for the organization of the virtual device is not considered.
- Software vulnerabilities: Vulnerabilities due to programming errors in the operating system, used libraries or the middleware designed in Chapter 4 and Chapter 5 are not considered. In particular it is assumed that the used security functionality (e.g. TLS) is flawless.
- Security of device discovery: Since device discovery is not the focus of this thesis, it is assumed that device discovery does not introduce security threats. Privacy threats on the physical layer, i.e. the detectability of a device without identification, are neglected.

6.2.1 Internal Evaluation

The internal evaluation identifies assets that have been introduced by the refinement of the functional architecture and the corresponding mechanisms. Based on the assets threats are identified by means of the STRIDE methodology. The threats are checked against existing security functionality. If necessary, additional security mechanisms are outlined.

6.2.1.1 Asset Identification

The refinement of the architecture has introduced additional assets or substantiated existing assets. These assets have been identified by systematic examination of the functional architecture (→ Section 4.5) and of the mechanisms, protocols and algorithms (→ Chapter 5). Table 6.3 provides an overview on the newly identified assets.

Table 6.3: Newly Identified Assets

No	Asset	Description	Source
A20	Logging Information	Logging information contains information about system activities.	Section 4.5.4
A21	Identity Information	Identity information provides details on the user's identities. Devices capture, store and exchange identity information (e.g. identity identifier, security requirements) as input for identity filtering.	Section 4.5.2
A22	Identity Device Matrix	The identity device matrix describes, which identity can be used on which device. As a result of the filter process the device identity matrix is created.	Section 5.2

Continued on next page

Table 6.3: Newly Identified Assets

No	Asset	Description	Source
A23	Device Information	Device information describes characteristics of devices. Devices capture, store and exchange device information among each other.	Section 4.5.5
A24	Virtual Device Composition	List of devices that are part of virtual device.	Section 5.4.2
A25	Virtual Device Key pair	Key pair of the virtual device, which is used to establish master devices.	Section 5.4.2
A26	Master Device Key Pair	Key pair of a master device. Several master device key pairs might exist to add/remove devices from the virtual device and to manage the virtual device list.	Section 5.4.2
A27	Non-Master Device Key Pair	Key pair of a non-master device. Each device belonging to a virtual device has a key pair to prove its membership.	Section 5.4.2
A28	Secure Storage	Contains security crititial data on the virtual device, e.g. used to store the device membership list.	Section 5.4.2
A29	Filter Rules	Used to control the Prefiltering step of the Identity Filtering Process.	Figure 5.6
A30	Ranking Rules	Used to rank identities in the Final Filtering step of the Identity Filtering Process.	Figure 5.8
A31	Identity Recommendations	List of identities that is recommended for use with a selected service.	Section 5.3.1

6.2.1.2 Threat Identification and Countermeasures

The application of the STRIDE methodology leads to Table 6.4 that provides an overview on the new threats. For more details on the individual threats it is referred to Section A.4.1. Table A.3 provides a short description for each threat and a corresponding assessment. If the threat is covered by the designed architecture or by defined security mechanisms, it is not necessary to identify additional security mechanisms. However, if the threat is not (i.e. “No” in column “Covered”) or only partially (i.e. “Partially” in column “Covered”) addressed, a detailed discussion and additional measures are required.

Table 6.4: Identified Threats based on Section 4.5 and Section 5

No	Asset	Spoofing of User Identity	Tampering	Repudiation	Information Disclosure	Denial of Service	Elevation of Privileges
A20	Logging Information		A20_T1	A20_T2	A20_T3		
A21	Identity Information		A21_T1		A21_T2		A21_T1
A22	Identity Device Matrix		A22_T1		A22_T2		A22_T1
A23	Device Information		A23_T1		A23_T2		A23_T1
A24	Virtual Device Membership List		A24_T1		A24_T2		A24_T1
A25	Virtual Device Key Pair		A25_T1		A25_T2	A25_T3	A25_T1
A26	Master Device Key Pair		A26_T1		A26_T2	A26_T3	A26_T1
A27	Non-Master Device Key Pair		A27_T1		A27_T2	A27_T3	
A28	Secure Storage		A28_T1		A28_T2	A28_T3	
A29	Filter Rules		A29_T1		A29_T2		
A30	Ranking Rules		A30_T1		A30_T1		
A31	Identity Recommendations		A31_T1		A31_T2		

As a result of Table A.3, all threats are covered at least partially. Partially means that security mechanisms to address these threats are in place, which are basically suited but do not cover a threat completely. The partially covered threats can be classified into four different categories in order to discuss the potential damage and suitable countermeasures: (1) Influence of User, (2) Storage of Virtual Device Key Pair, (3) Realization of *SecStore* and (4) Secure Storage.

(1) Influence of User: Threats A21_T1, A23_T1, and A29_T1 are only partially addressed, because the user is able to modify identity information, device information, and filter rules. Giving the user the possibility to influence the system behavior was a design choice, which follows the usability principle to rather assist the user than restrict him. Therefore, the user has comprehensive possibilities to influence the operation of the system by modifying various kinds of settings. Modification of this information results in different and potentially insecure results of the identity filtering process and the master device selection. As a countermeasure, the modifications of the user have to be evaluated. Based on the evaluation, an assistant should inform the user about the potential impact of the modifications and provide alternatives. That the stored information is modified by an external attacker is not considered as a threat, because the *SSE* provides secure storage.

(2) Storage of Virtual Device Key Pair: Threats A25_T1, A25_T2, and A25_T3 target the virtual device key pair that makes up the trust anchor of the virtual device. Existing security mechanisms address these threats only partially. The *SSE* provides the possibility to securely store the key pair on a selected device within the virtual device. However, the virtual device key pair is only rarely required, e.g. if a new master device should be added. Therefore, it is reasonable to consider additional security mechanisms. One solution is to store the key pair not within the virtual device, but somewhere else. Thus, it is not directly useable by any device of the virtual device composition. For example, an external storage medium that is not accessible via any network is appropriate, if it is stored at a secure place. Such a solution has a low usability. Therefore, additional mechanisms are required to provide an adequate level of usability. Another solution is the encryption of the private key. In both cases an explicit user action is required to make use of the virtual device key pair. This is a reasonable measure to protect the key pair and could be easily added.

(3) Realization of *SecStore*: The *SecStore* was added to host the virtual device membership list. In Section 5.4.2 it was assumed that *SecStore* is realized as a regular device within the virtual device according to the principle to limit the number of different security mechanisms. In consequence, the threats A28_T1, A28_T2, and A28_T3 can be reduced to threats that also hold for the *SSE*. However, alternative mechanisms should be considered, which do not follow the principle introduced before, but rather reduce the complexity of *SecStore*. One solution is the reduction to pure storage that is world readable, but restricted regarding modification. If the information is encrypted in such a way that it can be decrypted by every device that is part of the virtual device, it does not matter whether anybody can access the storage. However, modifications should be limited to authorized parties, i.e. master devices. This solution can be reduced to the group key problem [RH03].

(4) Secure Storage *SSE*: The remaining threats [6]are addressed by the *SSE* and the corresponding security mechanisms. Therefore, the security of the *SSE* is essential for the operation of the virtual device and all concepts that rely on it. In addition to the refinement of the *SSE*, addi-

tional security mechanisms have to be considered. In the following a couple of mechanisms are identified:

- Device Binding: The security of the secure channels depends on the security of the device key pair. Since the device key pair can be removed from one device and put to another device if the *SSE* is compromised, it requires mechanisms to bind the key pair to the device. Trusted computing can provide such a mechanism. But also the binding to an unmodifiable characteristic (e.g. MAC address) is considered as appropriate.
- History Mechanism: A history mechanism manages old versions of data and allows tracking of modifications. Therefore it is possible to detect malicious modifications. For example, decisions of the Final Filtering step of the identity filtering process depend on the device identity matrix. If the input data to create this matrix is modified, the matrix and thus the results of the Final Filtering step are influenced. If a history of the device identity matrix exists, it becomes possible to detect malicious activities.

6.2.1.3 Summary

The internal evaluation has identified a couple of threats that are not sufficiently addressed by the functional architecture and the concrete mechanisms. None of these threats is considered as severe, because for each threat an adequate mechanism could be proposed.

6.2.2 External Evaluation

For the external evaluation no standardized methodology is known. Therefore, an attacker model is defined that identifies different attackers and different locations for attacks. The attacker model is applied to different attacks that are reasonable with respect to the defined architecture. The attacks comprise active attacks, i.e. the attacker directly influences the system as well as passive attacks, in which the attacker only passively obtains knowledge about the system by observation. For each attack, the following issues are discussed:

- Attack Description: Outlines the overall attack.
- Attack Motivation and Potential Attackers: Details the motivation and objectives of performing an attack and identifies the potential attackers based on the attack model.
- Affected Parties: Elaborates how the different parties are affected by the attack.
- Attack Analysis: Describes the required steps to perform the attack with the support of attack trees (→ Section 2.5.4). The attack trees highlight attack paths that have been introduced by the virtual device and the multi-device IdM concept. Some subpaths are common to various attack paths. These subpaths are contained in Section 6.2.2.9.
- Attack Evaluation: Discusses the impact of the attack.

[6]These are all threats that are contained in Table A.3 and which have not been mentioned by one of the previous categories.

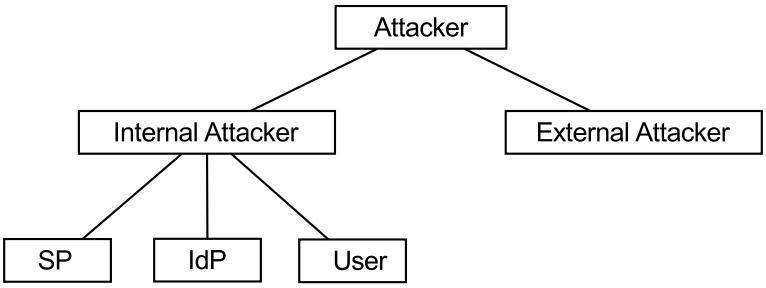


Figure 6.4: Classification of Attackers

The attack model (→ Section 6.2.2.1) defines the attackers and the potential attack locations. Based on the attack model, Section 6.2.2.2 motivates the considered attacks and provides a corresponding overview. The remainder of this subsection evaluates the considered attacks in detail. Finally, Section 6.2.2.10 sums up the results of the external evaluation.

6.2.2.1 Attack Model

The attack model defines the attackers and the potential locations of attacks.

Attacker: Figure 6.4 differentiates between two kinds of attackers: Internal attackers and external attackers. An *internal attacker* is any party that regularly participates in the system. That means each of the roles - user, IdP, and SP - can act as an attacker within the system. This includes attacking any other role as well as exploiting its capabilities to misuse the system. It is assumed that no party behaves maliciously against itself. Network access providers are not considered as attackers. For a detailed discussion on privacy threats caused by network operators, it is referred to [Hau08]. An *external attacker* is not an inherent part of the system. Examples for external attackers are fraudsters that want to consume services on behalf of the user. Depending on the strength of the attacker, he can conduct various attacks on the system and against all participating parties.

Attack Location: Figure 6.5 identifies seven different locations for potential attacks.

- Location (1) – Device of Virtual Device: An attacker might perform attacks on one of the member devices. Potential attacks include the usage of existing interfaces between the functional blocks as well as modification of the middleware.

- Location (2) – Communication between Devices of a Virtual Device: An attacker might perform attacks by misusing the communication channel between the member devices of a virtual device. Potential attacks include the eavesdropping of communication activities and the usage of exposed interfaces amongst others.

- Location (3) – System of Service Provider: An attacker might attack the systems of the SP. This attack location is not considered in detail, because the system of the SP is assumed to be secure.

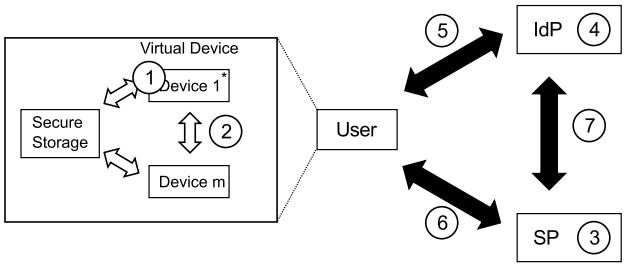


Figure 6.5: Points of Attack

- Location (4) – System of Identity Provider: An attacker might attack the systems of the IdP. This attack location is not considered in detail, because the system of the IdP is assumed to be secure.
- Location (5) – Communication channel between User and IdP: An attacker might get access to the communication channel between user and IdP. This might be used for eavesdropping as well as for message injection.
- Location (6) – Communication channel between User and SP: An attacker might get access to the communication channel between the user and the SP. This might be used for eavesdropping as well as for message injection.
- Location (7) – Communication channel between IdP and SP: An attacker might get access to the communication channel between the SP and IdP. Since this kind of communication is not relevant for the security evaluation, it is not considered in the following.

6.2.2.2 Attack Selection

For the external evaluation it is important to highlight attacks that result from the introduction of the virtual device and the multi-device IdM key concepts. These key concepts lead to additional interfaces, additional communication between the devices and additional data that is stored. Based on this, three categories of attacks are considered: (1) Unauthorized usage of services, (2) the observation of user characteristics and (3) the disturbance of the user. Table 6.5 provides an overview on the attacks.

(1) Unauthorized usage of services: Based on the ARP and the IAP, it is possible that services can be used on devices without performing the authentication on that device. Thus, the attack of unauthorized usage of services has to be assessed (Section 6.2.2.3) in detail. Category (1) consists only of this attack.

(2) Observation of user characteristics: With the virtual device concept, a lot of information on devices and on identities is exchanged between the devices. Information on devices and on identities is considered as information that describes the user's characteristics and thus inhibits the user's privacy if revealed. In consequence additional targets for attacks to breach user's privacy exist. In Section 6.2.2.4 an attacker is interested in the virtual device composition and wants to obtain information on the devices. The identities of the user are of interest in

Table 6.5: Overview of Considered Attacks

No	Attack Name	Category
1	Unauthorized Service Consumption	1
2	Obtain Information on Virtual Device Composition	2
3	Observation of User Identities	2
4	Observation of IdP and SP Sessions	2
5	Disturb Virtual Device Operation	3
6	Enforced Termination of IdP Session	3

Section 6.2.2.5 and the existing SP and IdP session are subject to the observation of an attacker in Section 6.2.2.6.

(3) Disturbance of User: The benefits of the virtual device and the multi-device IdM concept are provided by the collaboration of individual devices. If an attacker can disturb the collaboration of the devices, the devices fall back to independent operations after some time and the benefits are lost. Therefore, Section 6.2.2.7 assesses an attack to interrupt the virtual device operation. An attack to shut down an IdP session is evaluated within Section 6.2.2.8.

6.2.2.3 Attack 1: Unauthorized Service Consumption

Attack Description: An attacker consumes a service on behalf of a user without being authorized.

Attack Motivation and Potential Attackers:

(1) Direct monetary objective: If the service is not for free, consuming the service itself represents a valid goal. The user gets charged for the service. Depending on the charging model [BBB+10], the attacker creates additional costs (e.g. in case of pay per use) or only exploits the user's already paid access. Example: A charged online gaming service might be a worthwhile service to be consumed from the perspective of an attacker.

(2) Indirect monetary objective: If the service itself is for free, it might be used to trigger other events for which the user gets charged. For example, ordering goods on behalf of somebody else at an online shop.

(3) Destruction of reputation: Using a service on behalf of the user might trigger actions to destroy the reputation of one of the user's identities. Example: Modification of the user's profile at an online social network.

(4) Access to private data: Getting access to a service is often combined with getting access to data stored with the service. The attacker might be interested in this data. Examples for such data are the order history in case of online shopping, the user's profile with online social networks, or the account balance in case of online banking.

(5) Decrease of availability: Getting access to a service and consuming a service might put the attacker into the position to disable the account by explicitly quitting of the service contract or by malicious behavior leading to a blocking of the account by the SP.

From the attacker classification above, only an external attacker is considered. The user has no interest to consume a service on behalf of himself. Even if the SP has a potential interest to consume another service on behalf of the user, we do not consider this here due to the trust assumption within a federation (→ Section 3.2.2.6).

Affected Parties: The user, the SP and the IdP, are affected by illegal service consumption and have strong interests to avoid that.

- User: The user is directly affected by illegal service consumption from an unauthorized attacker. The maliciously created cost, the potential of the destroyed reputation, disclosure of private data and the non-availability of the service are not in the interest of the user. In addition to the user, the employer of the user is affected, if the service is running on behalf of the company or if sensitive company information gets accessible.
- SP: The SP is the other directly affected party of illegal service consumption. Even, if the user has to cover the resulting damage, the reputation of the SP is decreasing the more occurrences of illegal service consumption take place. In addition, the SP has to deal with users that repudiate the usage of the service.
- IdP: Since the IdP has created the SP Assertion for authentication and authorization, it is also affected by the illegal service consumption. In particular the reputation of the IdP is decreasing if too many illegal service consumption events take place. In consequence SP could release the federation and put the IdP out of business. Therefore, it is in the interest of the IdP to avoid illegal service consumption.

Attack Analysis: The attack tree in Figure 6.6 illustrates potential attack paths. The attack paths that result from the introduction of the virtual device concept and from the multi-device concept are indicated by $AP_i i \in \{1..6\}$. To consume services an attacker can either capture an existing service session or establish a new service session. Since it is not possible to elaborate all potential paths two example paths are detailed. Both attack paths have in common that a new service session should be established based on a SP Assertion.

Attack Path 3: The SP Assertion is actively requested from another device. In order to actively request a SP Assertion, either one of the user's devices must be compromised or the attacker must integrate a device into the virtual device. The latter approach is considered. The steps to add a device to the user's virtual device are detailed in Figure 6.15. If the attacker has added a device to the virtual device, two additional security mechanisms have to be circumvented. First, the requested device must have a policy to authorize the ARP request and second, the user has to confirm the release of the SP Assertion. The latter step is considered as optionally.

Attack Path 5: Instead of actively requesting the assertion, the attacker intercepts the SP Assertion during its transmission between two devices of the virtual device. That means the attacker has to break the secure channel between the devices. Figure 6.13 details the required steps to break the secure channel.

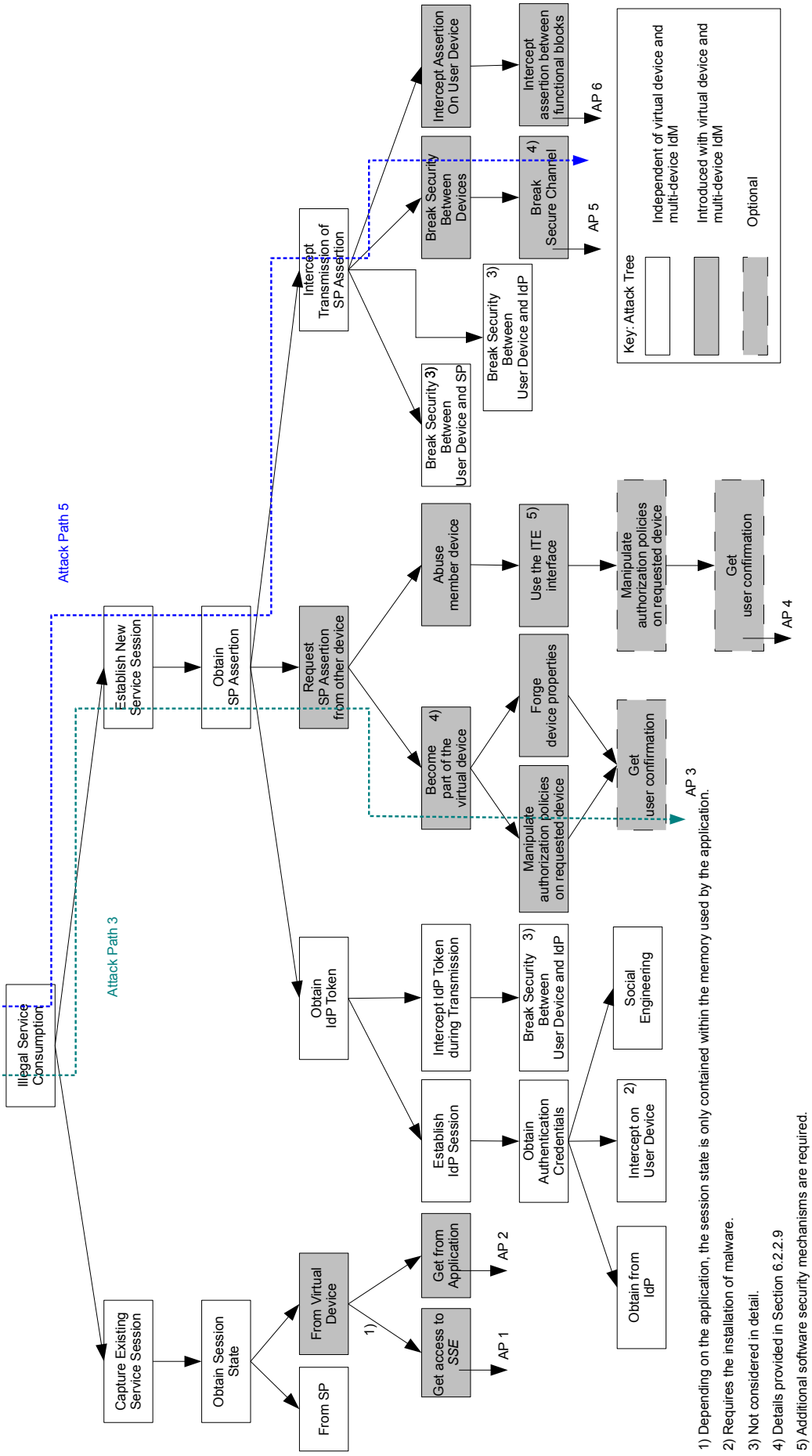


Figure 6.6: Attack Tree for Attack 1: Unauthorized Service Consumption

Attack Evaluation: Five different attack paths can be distinguished to consume a service on behalf of a user. Attack path 1 and Attack path 2 provide the possibility to overtake an existing session. This seems not to be attractive from an attacker's point of view, because he cannot decide on the service on its own. Moreover, security mechanisms of the *SSE*, which are considered to be strong (→ Evaluation of security requirements), or of the application have to be circumvented. Attack path 3 requires the attacker to make his device part of the virtual device. The analysis in Figure 6.15 showed that this is difficult and requires user involvement. In addition, at least two different actions are required to obtain the SP Assertion. Attack path 4 requires that the attacker has one member device under his control and abuse existing interfaces. Moreover, the attacker has to manipulate the requested device, which creates additional effort, i.e. manipulating a second device. In case of attack path 5 and 6, interception of the SP Assertion is not attractive. The attacker depends on the service selection of the user and is only in the position to use these services. Also the attention of the user is raised, since the intercepted SP Assertion cannot be used anymore for the service intended by the user.

6.2.2.4 Attack 2: Obtain Information on Virtual Device Composition

Attack Description: An attacker obtains information about the devices that are part of the virtual device.

Attack Motivation and Potential Attackers:

- IdP: IdM systems and the designed system do not provide any explicit information on devices to the IdP (e.g. no device identifiers are provided), i.e. the virtual device concept is transparent to the IdP. The motivation for an IdP to obtain information about the virtual device is the following: First, if the IdP would know which devices the user has and which devices should be used for authentication, it becomes possible to identify malicious behavior of attackers. Second, the IdP wants to prevent the transparent usage of the virtual device and therefore wants to identify the individual devices. However, it must be stated that the IdP knows a lot of information about the user. Therefore, the motivation of gathering information about the virtual device is not considered as relevant.
- External Attacker: There are two rationales for an external attacker to obtain information on the virtual device. First, the information on the virtual device composition can be used to launch other attacks. For example an attacker could directly attack a device with known weaknesses (e.g. zero-day exploits). Second, an external attacker might simply be interested in composition of the virtual device to breach the user's privacy and use the obtained knowledge for example for marketing purposes. Hereby, the number and the types of devices represent indicators for the wealth of the user. Moreover, the device types allow to derive additional information on the employer of the user (e.g. if the user has a "Blueberry" device, it is probably a business device).
- SP: The SP has the same interest as the external attacker with respect to user's privacy. The more information the SP obtains the better the service can be customized. On the one hand this improves the service experience of the user and is, thus, advantageous. On the other hand the customization can be harmful. For example depending on the user's devices different product prices might be presented by the SP. Moreover, the SP is interested

in the fact the user uses different devices with respect to digital rights management. For example, if digital content should only be accessible from one device at the same time, the SP wants to have mechanism to identify different devices.

. The SP might be interested to identify, whether the authentication is performed on another device than the device on which the service is used. Such information is useful with respect to digital rights management to avoid the service usage on different devices.

Affected Parties: In the first instance, the user of the virtual device is affected by this attack. The user's privacy is violated, if an attacker obtains knowledge on the devices the user possesses. In the second instance, the employer of the user is affected, if an attacker is able to categorize a device as a business device in order to prepare and launch future attacks on the device.

Attack Analysis: Figure 6.7 shows the attack tree to obtain information on the virtual device composition. Two different attack approaches are distinguished. Either an attack actively obtains the virtual device membership list or passively observes the virtual device. One way to observe the virtual device is to observe the network communication, which is detailed in Figure 6.8.

Figure 6.8 exemplifies different constellations for the communication flows in case of a virtual device consisting of two devices. In Figure 6.8(a) and Figure 6.8(b) the devices communicate by a public network which consists of one or two access networks. If an attacker wants to observe the communication between the devices, he must get access to the traffic within the access networks. In contrast, in Figure 6.8(c) the devices communicate directly, e.g. by using private network or a personal area network (PAN) technology like Bluetooth. Therefore, an attacker must be close to the communication devices. If the *SecStore* ($\rightarrow$ Figure 6.8(d)) is used by all devices, a central point for the observation of the virtual device exists. The knowledge that can be obtained there is restricted to the network addresses of the devices. If these network addresses are not unique, no information for information derivation is provided [Hau08]. Finally, Figure 6.8(e) considers the case where one device enables network access to the other device. Therefore, both devices appear from the network perspective as one device.

Attack Evaluation: Figure 6.7 shows that six different attack paths exist to obtain information on the virtual device composition. Attack path 1 and 2 rely on compromisation of one of the devices that are part of the virtual device. If this can be achieved a list of member devices and depending on the compromised device (e.g. a master device) more information is accessible. Attack path 3 is based on the breaking of the secure channel which is considered as not relevant. The security of the *SecStore*, if not realized as part of the virtual device, is not considered. Since device discovery is assumed to be safe, attack path 4 is not considered. Figure 6.8 showed that there is a high effort to obtain information on the virtual device composition based on network observation and thus attack path 5 is considered as very unlikely. Finally, attack path 6 describes the possibility that the IdP and SP obtain information on the virtual device. As stated above, information collection by the IdP and by the SP is not considered as relevant.

6.2.2.5 Attack 3: Obtain Information on User's Identities

Attack Description: An attacker obtains information about the user's identities.

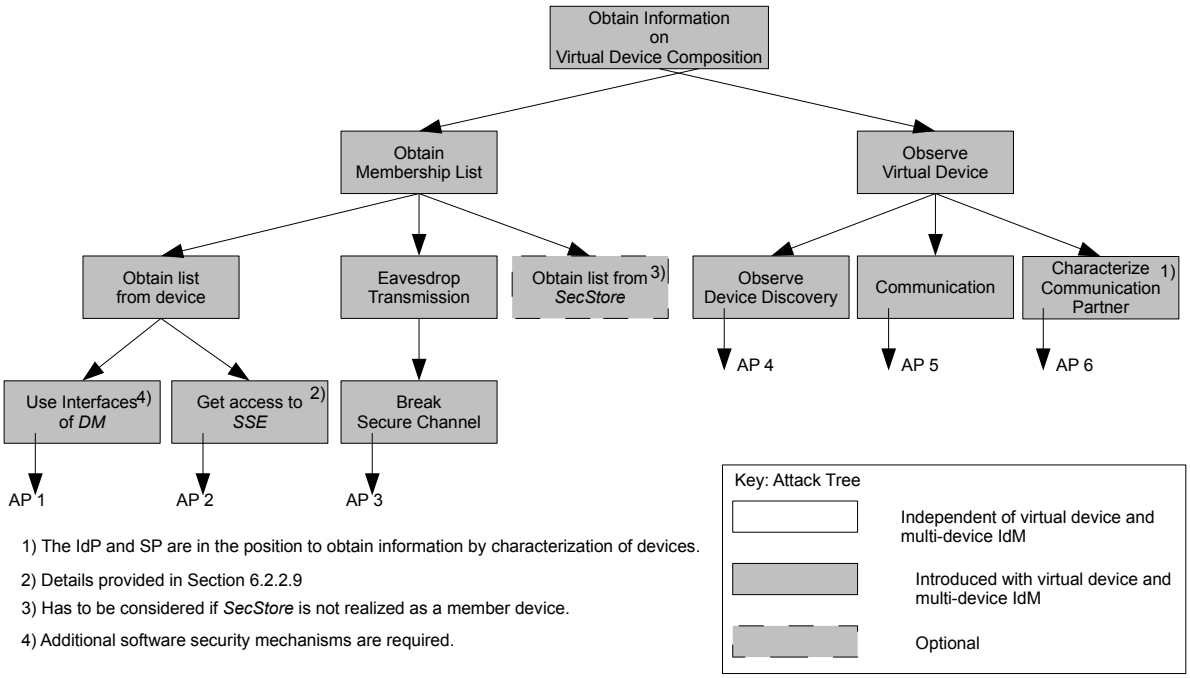


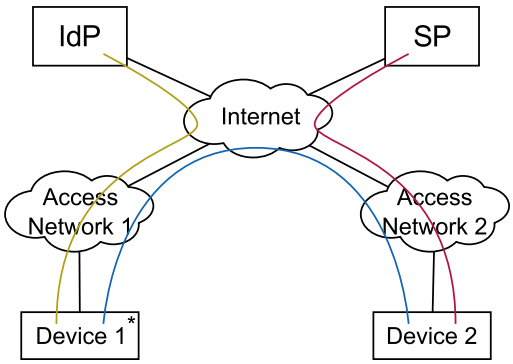
Figure 6.7: Attack Tree for Attack 2: Obtain Information on Virtual Device Composition

Attack Motivation and Potential Attackers: Identity information comprises all aspects that are relevant to the definition of the identity, i.e. the attributes (→ Figure 4.18) and its usage (i.e. the history of the services used). The IdP knows the attributes associated to an identity as well as the usage of the identity, because whenever a service is used, an appropriate SP Assertion is required. Therefore, it has to be assumed that the user trusts his IdPs and thus IdPs are not considered as attackers. Since not all SPs are trustworthy, a SP represents a potential attacker. Mechanisms to restrict the information that an SP might obtain have been examined in [Hau08, BNP+08, BDP07] and are not considered here. In the following, the focus is put on external attackers that target on breaching the user's privacy.

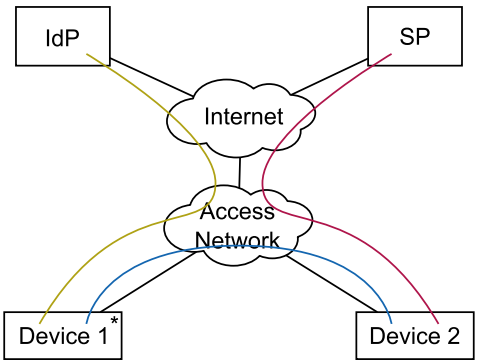
Affected Parties: Due to the usage of the virtual device, information on the user's identities is collected and stored. If this information is leaked, the user's privacy is breached. Therefore, the user is the only affected party.

Attack Analysis: Figure 6.9 shows the attack tree and identifies potential attack paths to obtain information about the user's identities. Attacks based on social engineering or attacks on the infrastructure of the IdP are not considered in more detail. The focus of the analysis is put on attack paths resulting from the virtual device concept. Hereby it must be noted that Figure 6.9 does not distinguish the different roles of the devices. Master devices have a complete view on the user's identities, whereas ordinary devices have only a restricted view.

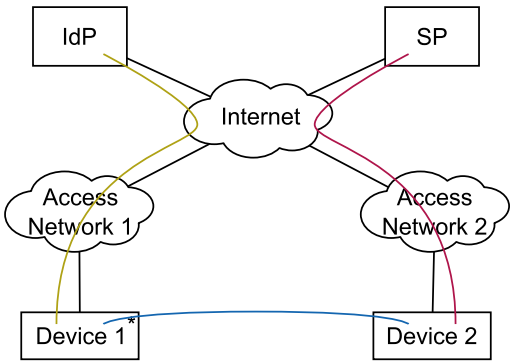
Attack Evaluation: All attack paths indicated in Figure 6.9 are based on attacking devices that are part of the virtual device. Attack path 1 targets the secure storage. Since master devices are assumed to be more secure than other devices, it is considered as difficult to break the corresponding secure storage. The same holds for attack path 2 that breaks the secure storage. Attack path 3 and 4 rely on the compromisation of a user device.



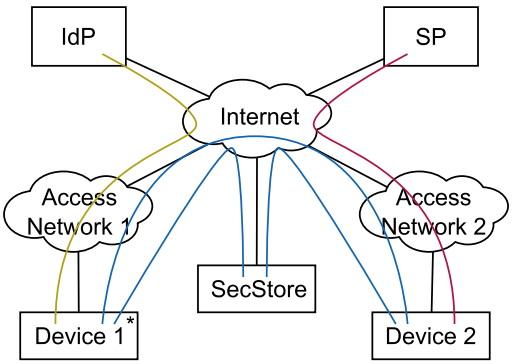
(a) Independent Access Networks



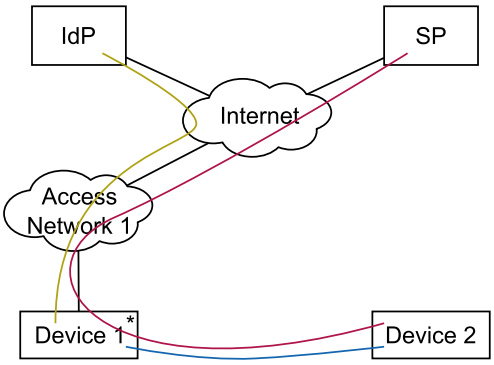
(b) Same Access Network



(c) Direct Communication



(d) SecStore



(e) Indirect Access

Figure 6.8: Communication Flows within Virtual Device

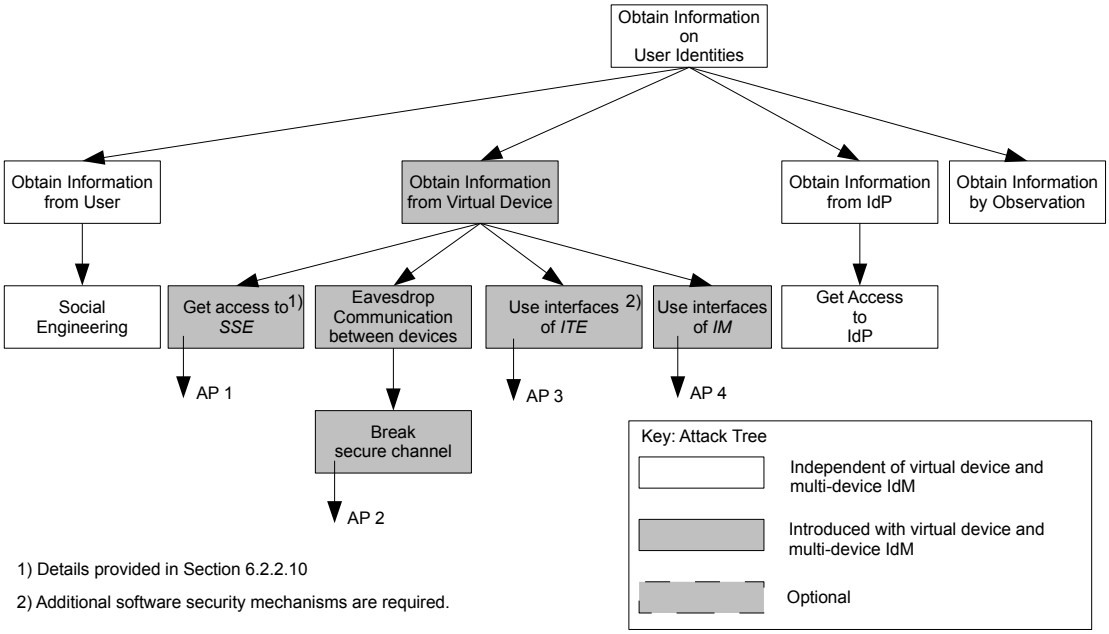


Figure 6.9: Attack Tree for Attack 3: Obtain Information on User's Identities

6.2.2.6 Attack 4: Observation of IdP and SP Sessions

Attack Description: An attacker obtains information about the active IdP and SP sessions of a user.

Attack Motivation and Potential Attackers: Only an external attacker has a motivation to obtain information about the user's sessions. IdPs and SPs inherently know about the sessions that exist with the user. An attacker might obtain this information to breach the user's privacy and learn about the user's behavior. This information is also useful to launch other attacks.

Affected Parties: The same as for attack 3 holds, the user is the only affected party of such an attack.

Attack Analysis: Figure 6.10 illustrates the attack tree to observe the active IdP and SP sessions. An attacker could observe the communication with the SP and the IdP. This possibility exists without the introduction of the Virtual Device and is not considered in more detail. The attack paths that exist due to the virtual device and due to the multi-device IdM concept are analyzed in the following.

Attack Evaluation: Three different attack paths can be distinguished. With attack path 1, the attacker observes the exchanged IIEP messages, i.e. it is required to break the secure channel. This is not considered as possible. In case of attack path 2, an attacker actively requests information about the existing IdP and SP sessions from one of the devices. This means that the attacker has to become part of the virtual device, which requires a lot of effort. If the attacker would get access to one of the devices, it would be possible to use the interfaces of the *ITE* or the *IM* to obtain information about the existing sessions (attack path 3). This can be prevented with adequate software security mechanisms. Since the *ITE* and the *IM* might store information

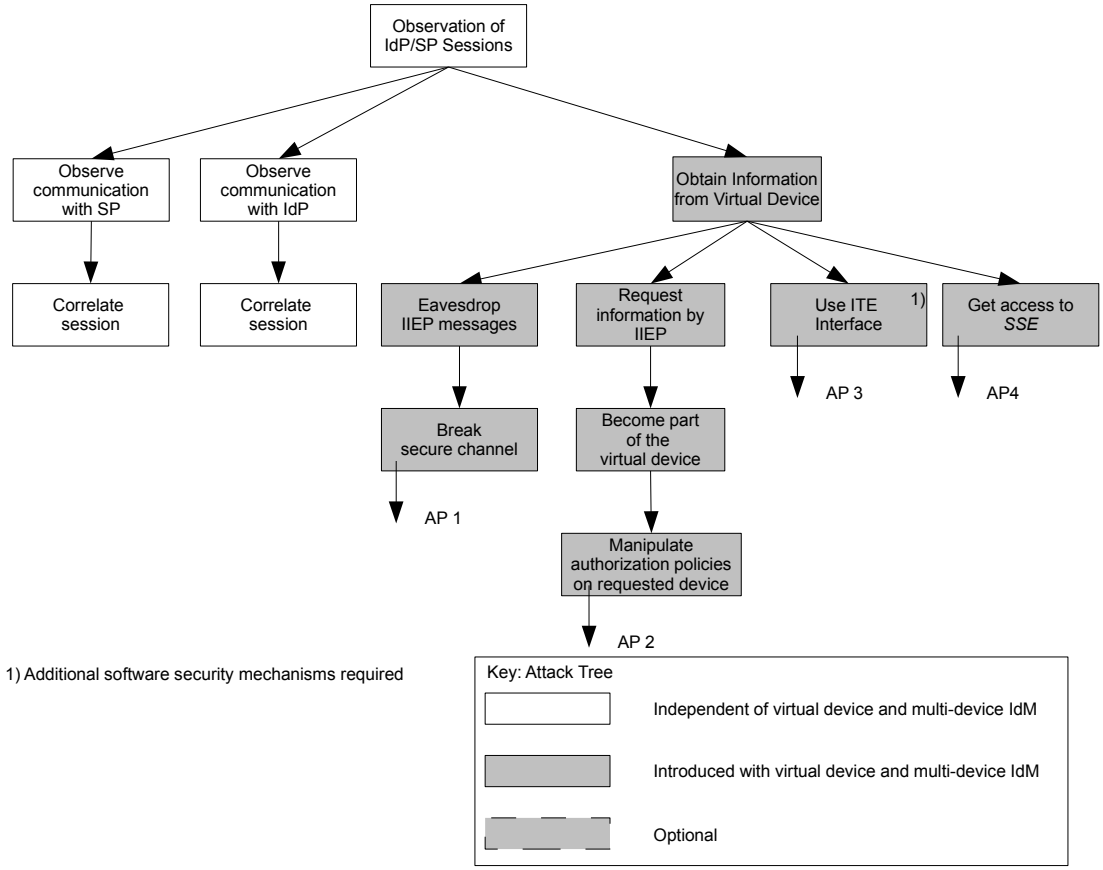


Figure 6.10: Attack Tree for Attack 4: Obtain Information on User's active IdP and SP sessions

about existing sessions with the *SSE*, it becomes subject to attack path 4. Getting access to the *SSE* is difficult.

6.2.2.7 Attack 5: Disturb Virtual Device Operation

Attack Description: An attacker disturbs the operation of the virtual device and makes the benefits provided by the virtual device and the multi-device IdM concept unavailable.

Attack Motivation and Potential Attackers: The user itself, the SP and the IdP have no interests in making the virtual device unavailable. Only an external attacker might have an interest in making the virtual device inoperable. There is no direct motivation for the external attacker, except the disturbance of the user.

Affected Parties: If an attacker disturbs the operation of the virtual device, the user is directly affected. The SP and the IdP are indirectly affected. The user cannot use the provided services and, thus, there is a loss of revenue for SPs.

Attack Analysis: Figure 6.11 identifies six different approaches to disturb the operation of the virtual device. Attacks on the communication infrastructure that make a device unreachable are not considered in detail, because each device is subject to such attacks anyway. Device discovery – as previously stated – is assumed to be secure. Attacks on the *SecStore* are not considered. The other attacks are based on the establishment of secure channels, the removal of devices and activities that require user actions.

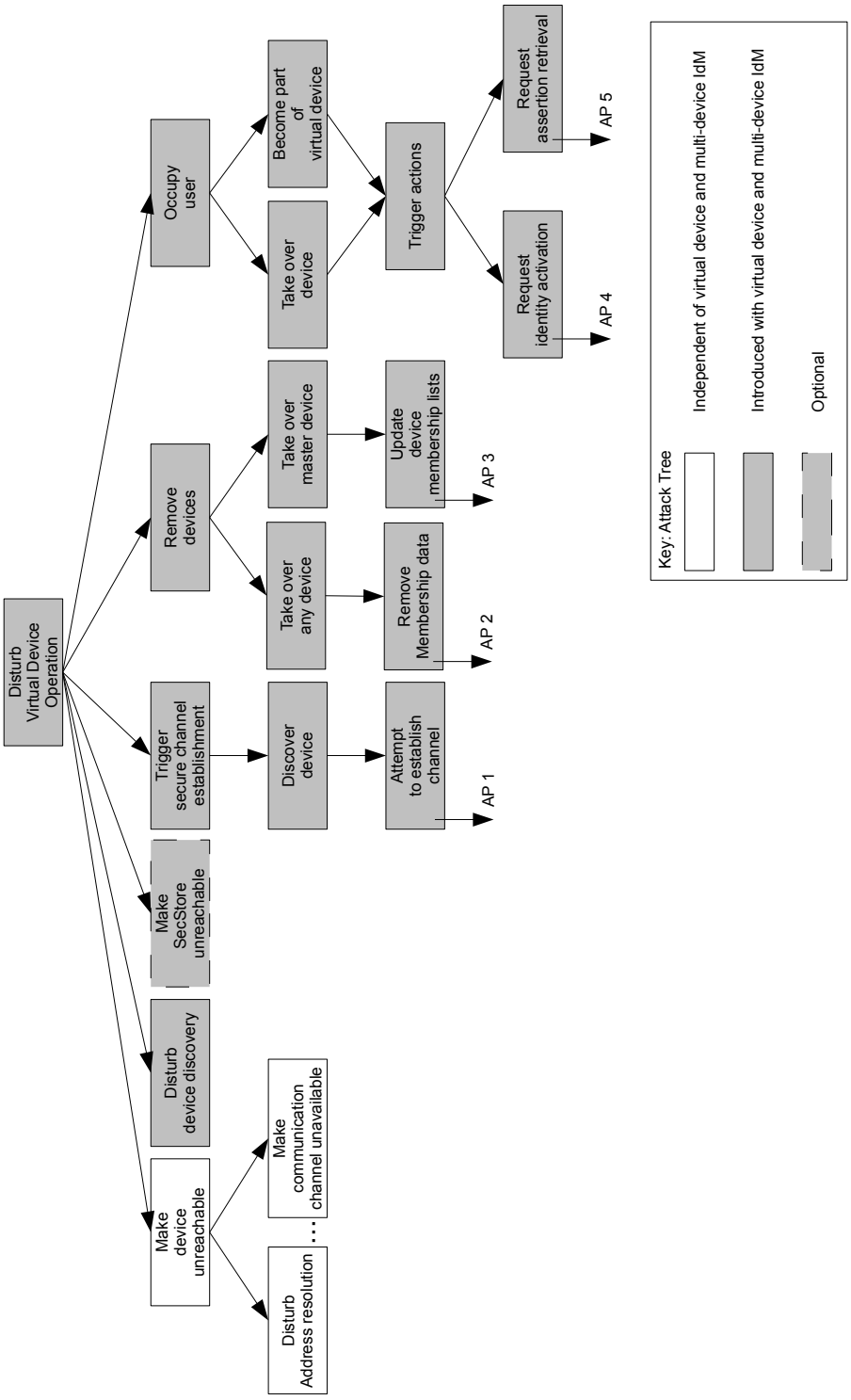


Figure 6.11: Attack Tree for Attack 5: Disturb Virtual Device Operation

Attack Evaluation: Attack path 1 represents a denial of service attack on one device by the request for secure channels. The establishment of secure channels fails, because the requesting device does not have appropriate certificates. Additional mechanisms on lower level (e.g. IP packet filters.) can prevent recurring requests for secure channels and thus prevent DoS attacks. Therefore, attack path 1 is considered as preventable with existing technology. With attack path 2 and 3, an attacker removes a device from the virtual device. In case of attack path 2, any device is taken over by the attacker and all data that is required to connect to the other devices of the virtual device is removed (e.g. the device certificates). Thus, the device is not part of the virtual device anymore. In case of attack path 3, a master device is captured. Since the master device is able to remove devices from the virtual device, an attacker could remove any device from the virtual device composition. Removing a device, which an attacker took over, from the virtual device represents a relevant attack to disturb the user. Attack path 4 and 5 occupy and bother the user with actions that require his attention. Either the attacker takes over one of the devices or becomes part of the virtual device to trigger actions like identity activation or the assertion retrieval. In case of attack path 3, 4, and 5 additional assistants could be provided that detect the malicious behavior and inform the user about potential measures like isolating a captured device from the virtual device composition.

6.2.2.8 Attack 6: Enforced Termination of IdP Session

Attack Description: An attacker terminates an existing IdP session and requires the user to re-authenticate, if the concerned identity should be used again.

Attack Motivation and Potential Attackers: The SP has no interest to terminate an existing IdP session. If a new IdP session for a given identity with different authentication methods is required, it could be signaled by means of an authentication context. Termination by the IdP is not considered as an attack but as a security feature. The IdP is always in the position to invalidate an existing IdP session in case of incidents. The user has also no motivation to terminate the IdP session without direct intention. Thus, only an external attacker has a motivation to terminate existing IdP sessions. First, terminating existing IdP sessions bothers the user, since he is required to reestablish the IdP session, i.e. to re-authenticate. Second, the external attacker could exploit this situation by launching other attacks, e.g. the interception of authentication credentials that are used in such a situation. Third, the coordinated termination of IdP sessions might result in a DoS attack on the IdP.

Affected Parties: The user and the IdP are affected by the termination of IdP sessions. On the first hand, the reestablishment of the IdP session represents an attack on the usability of the system from the perspective of the user. On the other hand, an external attacker could influence the availability of the IdP, if a coordinated attack on several users is possible. The re-authentication of each user would result in a potential DoS attack on the system of the IdP.

Attack Analysis: Figure 6.12 shows the attack tree for the termination of an IdP session. Basically it is possible for an IdP to terminate an IdP session. This is not considered in the following.

Attack Evaluation: Attack path 1 is based on the compromisation of one of the user devices and usage of existing interface to shutdown IdP sessions. If adequate software security mechanisms

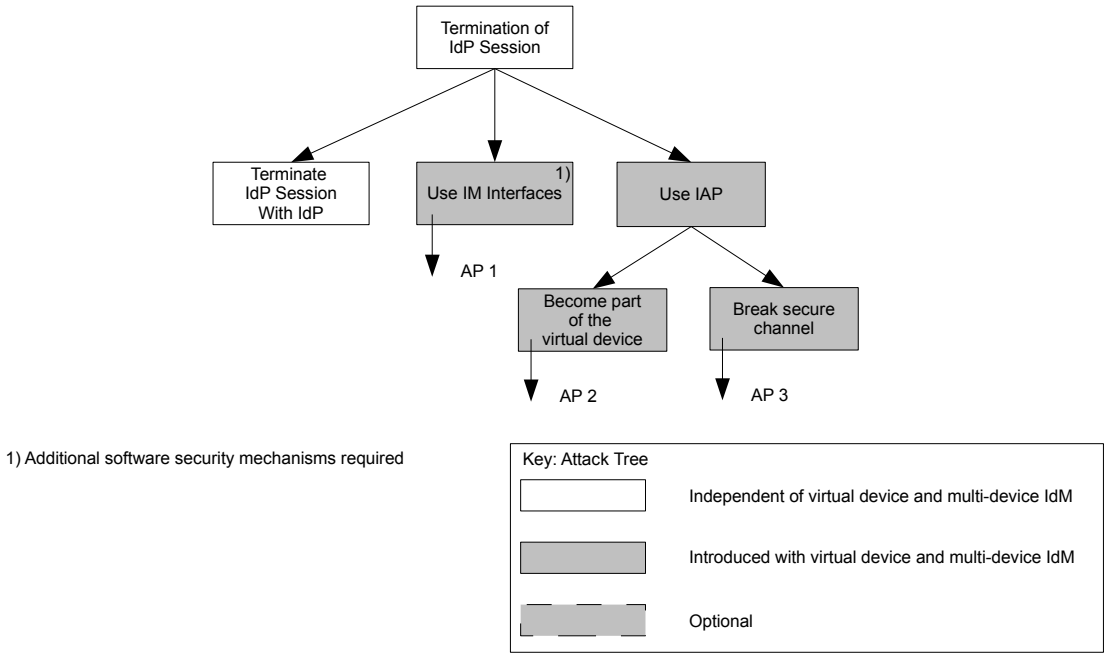


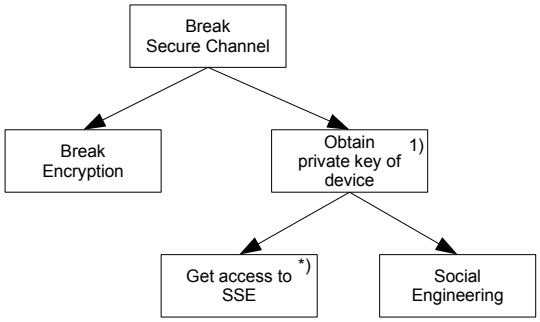
Figure 6.12: Attack Tree for Attack 6: Enforced Termination of IdP Session

are in place, this attack is not feasible. Attack path 2 and 3 rely on the usage of the IAP protocol that is used between the devices. This is only possible if the attacker becomes part of the virtual device or if he is able to break a secure channel. Both are considered to be difficult.

6.2.2.9 Common Subpaths of Attack Trees

Figure 6.13, Figure 6.14, and Figure 6.15 show common subpaths of the attack trees introduced in the previous subsections. Figure 6.13 illustrates the required steps to break a secure channel. Breaking the encryption of a secure channel is assumed to be not possible. However, if the attacker could obtain the private key that is used to establish the secure channel, it is possible to launch MitM attacks. Two ways exist to obtain the private key, either the security of the SSE is broken or the attacker uses social engineering methods to obtain the keys directly from the user. In particular social engineering represents a weakness that could not be prevented.

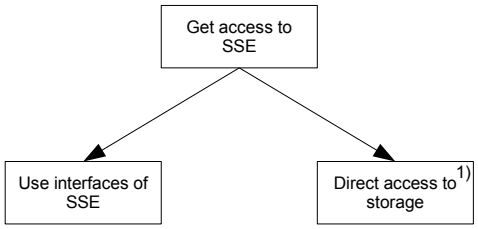
Two ways exist to get access to the *SSE* as illustrated in Figure 6.14. Either the attacker uses the existing interfaces for the access or he gets access to the storage medium that is used by the *SSE*. Usage of existing interfaces can be prevented by using software frameworks that have inherent security mechanisms like OSGI [OSG11]. Direct access to the storage medium can be either prevented by applying encryption or dedicated hardware. Basically two ways exist that can be exploited by an attacker to make a device part of the virtual device. Either social engineering is used to trick the user, which is considered as difficult due to the various steps that have to be performed, or by circumventing the existing security mechanisms. If an attacker takes over one master device, he can add any device to the virtual device. This is considered as easier than manipulating the virtual device membership list and signing the list with the public key of one of the master devices, which actually means to overtake a master device.



1) Used to launch a man in the middle attack.

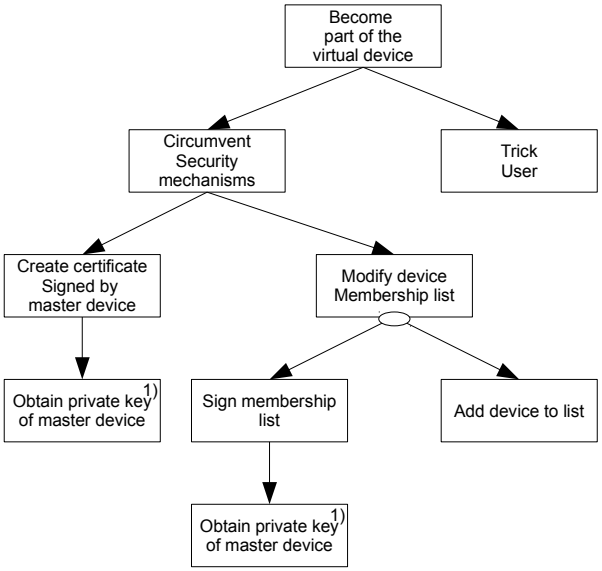
*) Detailed in Subsection 6.2.2.9

Figure 6.13: Common Subpath 1: Breaking the Secure Channel between Devices



1) How direct access to storage is realized is implementation dependent.

Figure 6.14: Common Subpath 2: Getting Access to SSE



1) Not considered in more detail.

Figure 6.15: Common Subpath 3: Adding a Device to the Virtual Device

6.2.2.10 Summary

The analysis of the six different attacks and the common subpaths showed that adequate security mechanisms are in place. The overall security depends on the strength and robustness of the implemented software security mechanisms. The unauthorized usage of the interfaces represents an issue that has not been adequately covered. However, existing solutions like OSGI [OSG11] or the security frameworks provided by recent mobile operating systems like Android [EOM09, BVO11] or IOS [BVO11] represent valid solutions that could be used.

6.2.3 Misuse Cases

[SO05] defines a misuse case as "a sequence of actions, including variants, that a system or other entity can perform interacting with misusers of the entity causing harm to some stakeholder if the sequence is allowed to complete". Typically misuse cases are applied for the elicitation of security requirements [SO05, Ale03, SO01]. Here, misuse cases consider use cases of the system that are not in scope of the design. This thesis assesses the potential impact of the misuse cases.

Two obvious misuse cases are elaborated in the following. Both misuse cases cannot be prevented by the designed technical mechanisms. However, extensions are outlined to cope with the misuse cases.

6.2.3.1 Integrating Foreign Devices

A user is in the position to add foreign devices, i.e. devices that are not under his control, to the virtual device. From the user's perspective this is reasonable, because the authentication does not have to take place on the foreign device and thus the authentication credentials are not revealed to a foreign party. The foreign device is only in the position to request the required assertions to consume a service.

With the integration of the foreign device into the virtual device, the foreign device obtains information about all devices that are part of the virtual device. The exchange of identity information and detailed information on device characteristics can be prevented with appropriate policies. In a similar way the consequences of the IAP and the ARP can be limited. Thus, potential threats can be limited, if the foreign device is removed from the virtual device composition immediately after retrieval of the SP Assertions. This means the exposure time t_{exp} in Figure 6.16 should be as short as possible. If the user does not remove the foreign device from the virtual device composition, any other party using the foreign device is in the position to request SP Assertions for one of the user's identities. If the user has disabled the manual confirmation (considered as optional in Figure 5.13), there is not even a way to notice the misuse.

Since the user uses the foreign device for the service consumption, the foreign device is able to exploit the service ($\rightarrow$ Section 6.2.2.3) and get access to corresponding content (e.g. user attributes, ...). Therefore, the trustworthiness of the foreign device determines the potential harm. The security level determination ($\rightarrow$ Section 5.4.2.4) does not reason on the trustworthiness of a

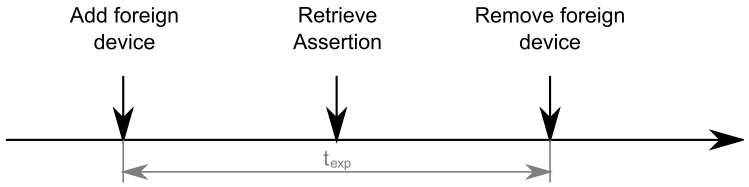


Figure 6.16: Misuse Case: Exposure Time

device. Thus, additional mechanisms (e.g. [YW09, GCB+08, OBDS04]) are required to reason about the trustworthiness or to protect the content.

If the trustworthiness of a foreign device can be assumed, the proposed architecture could be extended to add foreign devices for restricted actions. For example a foreign device can be added to request exactly one SP Assertion for a specific service. In addition, the view that the foreign device obtains on the virtual device could also be limited to exactly the device that fulfills the request.

6.2.3.2 Sharing of Virtual Device

The virtual device design is based on the assumption that one user has one or several virtual devices. It does not consider that several users share one virtual device. Sharing a virtual device means that several users integrate their devices into the same virtual device. The motivation to share a virtual device is coupled to the sharing of identities and thus to the sharing of services. In particular for expensive service (e.g. video on demand) several users are in the position to collaborate to use the same service with only one payment. Such a behavior is not in the interest of SPs. In most cases it is excluded by service agreements and prevented by additional mechanisms.

From the perspective of the users, all users are in the position to identify and discover all devices. With the knowledge on the availability of the other devices, it is possible to conclude on the behavior of the corresponding users. Information about the device characteristics and on the user's identities can be restricted by means of policies (→ Section 6.2.3.1). The effort for the configuration of the corresponding policies is high, resulting in a decreased usability. Therefore, sharing of a virtual device among several users is only useful if a strong trust relationship exists between the users. For example, within the same family trust relationships exist and sharing a virtual device without the effort to configure fine granular policies is reasonable. Moreover, it is possible to configure policies to restrict children regarding the usage of their identities and the corresponding policies. Such extensions to the functional architecture are feasible, but not detailed.

6.2.3.3 Summary

Even if the discussed misuse cases have not been considered in the design of the architecture, both can be supported with additional mechanisms. For the sharing of the virtual device a new use case has been identified.

Table 6.6: Summary of Security Evaluation

Part	Ref.	Title	Summary
1	6.1.2.5	Security Requirements	All security requirements have been sufficiently covered.
2	6.2.1	Internal Evaluation	Some assets are exposed to threats. No threat contradicts the architecture. Additional mechanisms are outlined.
3	6.2.2	External Evaluation	Attacks are countered by designed security mechanisms. Additional software security mechanisms are required.
4	6.2.3	Misuse Cases	Can be countered with additional security mechanisms. New use case identified.

6.2.4 Summary

Table 6.6 summarizes the results of the security evaluation. The functional architecture considers security inherently. The concrete mechanisms, algorithms and protocols in Chapter 5 address the core problems to provide multi-device IdM. None of the four parts of the security evaluation has revealed major security issues. The virtual device concept that enabled the collaboration of the user's devices and the multi-device IdM concept have increased the attack surface, i.e. the number of attack points. The security evaluation has shown that the increased attack surface does not result in additional vulnerabilities.

6.3 Performance Evaluation

This section evaluates the performance implications of the virtual device and the multi-device IdM concept. The goal is to create a basic analytical model that shows the relations between the user, the IdP and the SP in order to provide a better understanding of the implications of the architecture. Based on this model it is possible to quantify the benefits of the designed architecture. The architecture provides advantages from the performance viewpoint, if the following holds:

- Less Authentication Procedures: If the user has to authenticate less often, the usability of the architecture is increased in relation to conventional concepts.

- Decreased Number of IdP Sessions: If the IdP has to maintain a decreased number of IdP sessions for one user, it is advantageous from the perspective of the system load.

With the introduction of the virtual device concept and the multi-device IdM concept it is expected that the number of authentication procedures and that the number of IdP sessions is decreased. It is not subject of this section to provide numbers for capacity planning. That means the required server resources to manage IdP sessions or the required signaling capacity

for the establishment of IdP or SP sessions is out of scope. Moreover, it is not in scope to provide an accurate model for the user behavior regarding the service consumption.

This section is structured as follows. Section 6.3.1 introduces the applied methodology, which outlines three different cases for the evaluation and highlights the corresponding evaluation goals. The system model in Section 6.3.2 describes the assumptions for the subsequent analytical model. Section 6.3.3 introduces the metrics that are considered by the analytical model in Section 6.3.4. The evaluation results in Section 6.3.5 use the defined metrics to provide results on the consequences of the different cases. Finally Section 6.3.6 summarizes the key findings of this section.

6.3.1 Evaluation Methodology

The evaluation consists of the examination and the subsequent comparison of three different cases. Figure 6.17 illustrates the evaluation methodology and the corresponding cases. In Case 1, no SSO IdM system exists. That means the user has to authenticate against each SP individually. Case 2 introduces a centralized IdM system, which enables SSO. It has two subcases: Case 2.1 and Case 2.2. In Case 2.1 the user has only one identity and uses this identity with one device. Case 2.2 models the consequences of N_{Id} identities per user for which individual authentication is required. Case 2.1 and Case 2.2 are considered together, because Case 2.1 represents a special case of Case 2.2. Case 2.2 collapses to Case 2.1 for $N_{Id} = 1$. In the following it is not distinguished between Case 2.1 and Case 2.2. Case 3 extends Case 2 with respect to the number of devices. Instead of one device, the user has in Case 3 N_{Dev} devices.

The comparison of Case 1 and Case 2 allows the quantification of the impact of the SSO mechanism. Figure 6.17 highlights this as evaluation goal “Consequences of SSO”. The modification of N_{Id} provides details of the consequences of several identities. With the comparison of Case 2 and Case 3, it is possible to quantify the impact of the virtual device. Case 2 represents hereby the virtual device, which is considered as one device from the perspective of the IdP. Case 3 reflects the operation of N_{Dev} devices that are independent from each other.

6.3.2 System Model

The system model consists of three parties: User, IdP, and SP. The user consumes services with one of his identities, i.e. he has to select an appropriate identity. The required assumptions for a simple analytical model are the following:

- Single federation: It is assumed that there is only one federation. That means all SPs federate with one IdP, which is responsible for the management of user’s identities. Thus the identity selection is simplified.
- One user: It is assumed that there is only one user. This is sufficient for the evaluation of the number of authentication procedures and the number of IdP sessions. The consequences of several users can be evaluated by multiplying the results obtained for one user with the total number of users.

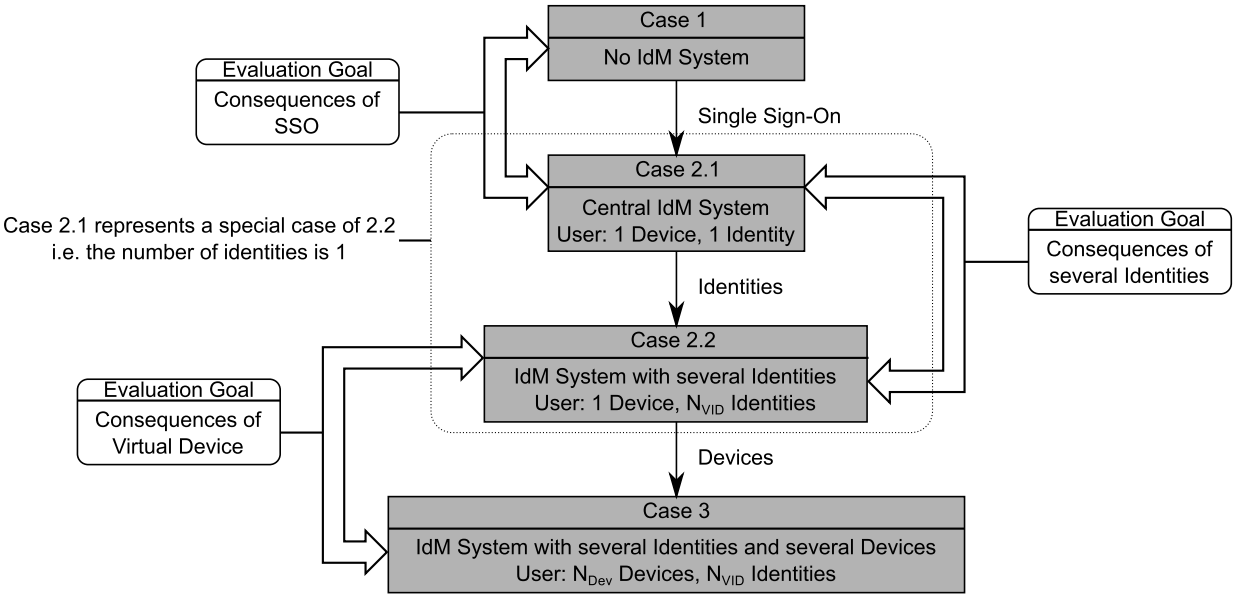


Figure 6.17: Performance Evaluation Strategy

- No identity restrictions: It is assumed that every identity can be used with every SP. Thus identity selection is simplified.
- No influence of multiple identities: It is assumed that users with several identities do not change their service consumption behavior in comparison to having only one identity.
- Duration of IdP session: It is assumed that an IdP session lasts until the last SP session that is based on this session terminates.
- No device restrictions: It is assumed that all devices are equal. That means IdP and SP sessions can be established on every device.

Assumptions concerning the service consumption model are detailed in Section 6.3.2.1. Section 6.3.2.1 creates the basis for the system models targeting on Case 1, 2, and 3 in the corresponding subsections (Section 6.3.2.2 to Section 6.3.2.4).

6.3.2.1 Service Consumption Model

Figure 6.18 illustrates the service consumption model. The user consumes different services, identified in Figure 6.18 as “Service 1” to “Service x”. If a user consumes a service, a SP session is established. It is possible to have more than one SP session at the same time. The duration of a SP session is called the holding time h_i. The time between two SP session starts is called the interarrival time d_i.

In the following it is assumed that the holding time and the interarrival time are negative-exponentially distributed. This assumption is based on the characteristics of HTTP sessions. According to [RLGPC+99, CGS01], HTTP sessions are negative-exponentially distributed. This confirms the well-known assumption regarding the call distribution and holding time in telecommunication networks [FM94, Gue87].

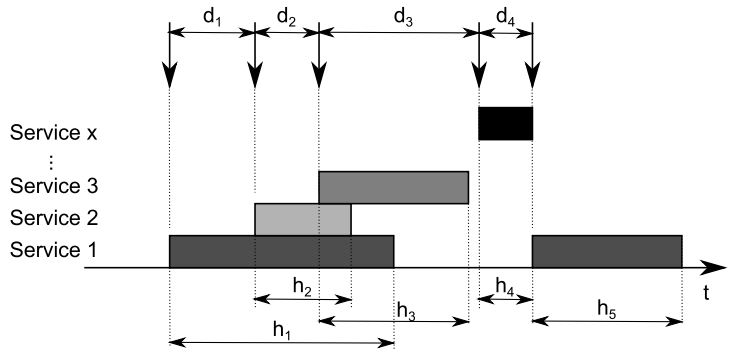


Figure 6.18: Service Consumption

6.3.2.2 Case 1: No IdM System

Figure 6.19 shows the system model for the case of independent SPs, i.e. no IdM system is in place. The Service Request Generator creates service requests with negative-exponentially distributed interarrival and holding times, which are indicated by $d = 1/\lambda$ and $h = 1/\mu$.

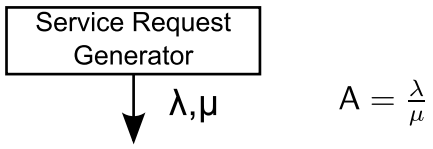


Figure 6.19: Service Consumption Model without IdM System

6.3.2.3 Case 2: User with Several Identities

Case 2 extends Case 1 with respect to the introduction of an IdM system and the support for N_{Id} different identities. In that way, Figure 6.20 extends Figure 6.19 by an Identity Selector component. The Identity Selector distributes the service requests across N_{Id} identities in a random way. Based on the random selection strategy and according to [Küh79], Eq. (6.1) defines the interarrival time of service requests for each identity. The value $A_{Id,i} = \frac{\lambda_{Id,i}}{\mu}$ defines the average number of SP sessions per identity.

$$\lambda_{Id,i} = \frac{\lambda}{N_{Id}} \tag{6.1}$$

6.3.2.4 Case 3: User with Several Identities and Several Devices

Case 3 extends Case 2 by the aspect of multiple devices as shown in Figure 6.21. A user has N_{Dev} devices to consume services. The selection of the device for a SP session is modeled

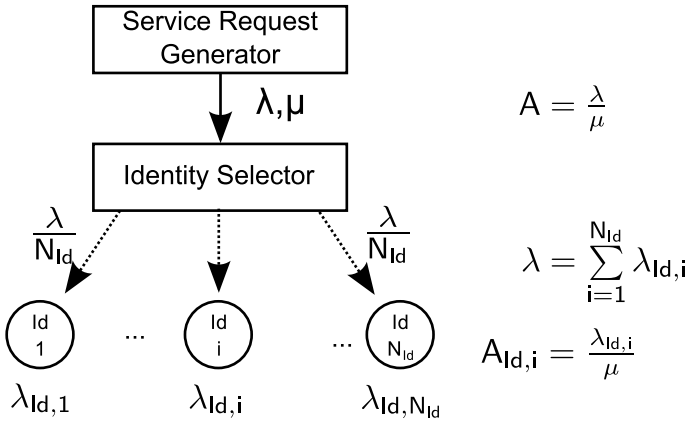


Figure 6.20: Service Consumption Model for one user device

by the Device Selector. The Device Selector randomly selects a device for each SP session and thus for the corresponding IdP sessions, i.e. the devices are independent of each other. In consequence SP sessions are distributed across identities and devices. Eq. (6.2) describes the arrival rate for SP sessions on device j for identity i. The corresponding average number of SP sessions per device and identities is defined by $A_{Id,i,Dev,j} = \lambda_{Id,i,Dev,j}/\mu$.

$$\lambda_{Id,i,Dev,j} = \frac{\lambda}{N_{Dev} N_{Id}} \tag{6.2}$$

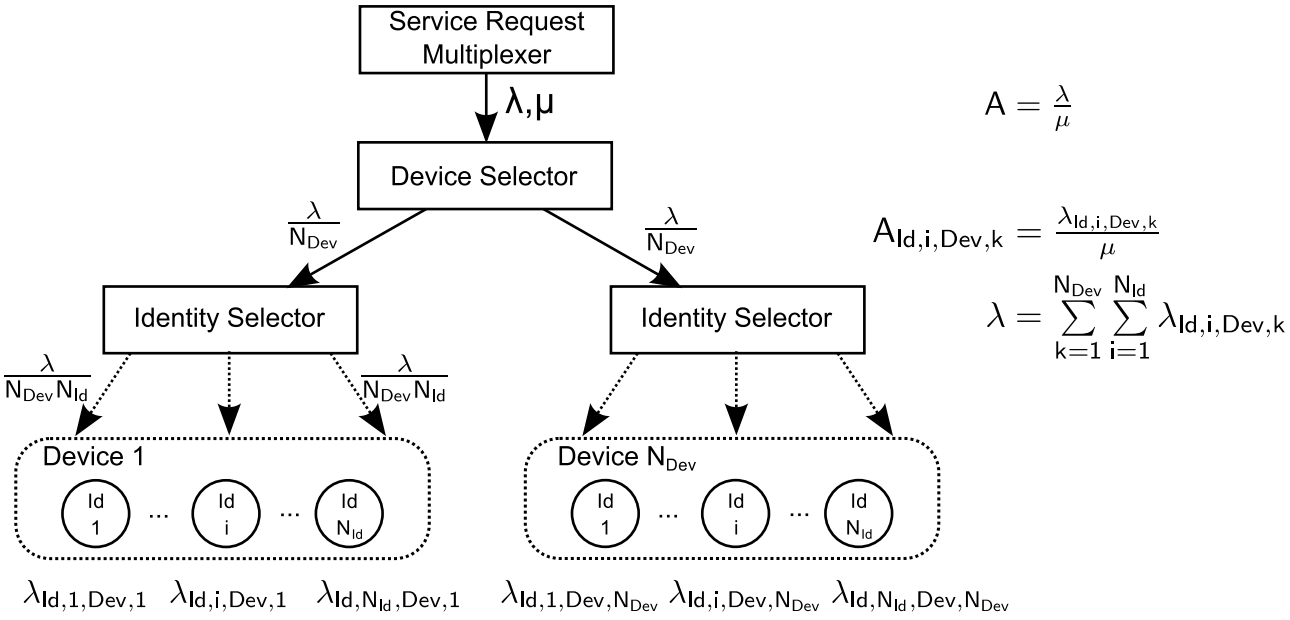


Figure 6.21: Service Consumption Model for Several Independent Devices

.

6.3.3 Metrics

In this thesis five key metrics for the quantification of the performance benefit of the multi-device IdM concept are applied:

- *Identity activation rate AR:* The mean identity activation rate AR quantifies how often identities are activated per time unit. It is used to determine the number of authentication procedures that have to be performed by the user. In accordance to AR, one can define an identity deactivation rate. Based on the assumption of a stationary system, which means that the results are independent of the particular instant of observation, the mean identity deactivation rate has to be the same as AR.

- *Authentication load A_{Auth}:* A_{Auth} is a normalized value of AR. It makes AR independent of the SP session holding time h and independent of the SP session interarrival time λ. A_{Auth} can be interpreted as the mean number of authentication requests per mean service duration $1/\mu$.

- *Mean number of active IdP sessions $E[N_{act}]$:* The mean number of active IdP sessions $E[N_{act}]$ gives the average number of identities that are simultaneously active. $E[N_{act}]$ can be used to quantify the additional state that has to be managed by the IdP.

- *Overhead without SSO $R_{C2,N_{Id}}$:* If no SSO mechanism is in place, the number of authentication procedures that have to be performed by the user is higher. $R_{C2,N_{Id}}$ captures the effort, i.e. overhead, of the user, if no SSO is in place. $C2$ is a reference to Case 2, which introduces SSO and multiple identities. Hereby, N_{Id} indicates the number of identities. Later, Eq. (6.18) defines $R_{C2,N_{Id}}$.

- *Overhead without Virtual Device $R_{C3,N_{Dev},N_{Id}}$:* If no virtual device concept is in place, SSO is performed on each device individually. This results in additional effort, i.e. overhead, for the user. $R_{C3,N_{Dev},N_{Id}}$ allows to capture this effort in dependency of N_{Id} identities and in dependency of N_{Dev} devices, which are not associated to a virtual device. Later, Eq. (6.27) defines $R_{C3,N_{Dev},N_{Id}}$.

6.3.4 Analytical Model

The analytical model implements the system model (→ Section 6.3.2) by means of Markov chains [Küh02]. A Markov chain is a stochastic process with discrete states that fulfills the Markov property. The Markov property defines that the future state only depends on the current state. The analytical model based on Markov chains allows the evaluation of the system model with the proposed metrics (→ Section 6.3.3). For Case 1, it is not required to detail the analytical model (→ Section 6.3.4.1). Section 6.3.4.2 introduces the analytical model for Case 2 and Section 6.3.4.3 extends the analytical model with respect to multiple devices. The analytical model extends existing work of the author [Bar09].

6.3.4.1 Case 1: No central IdM

If no central IdM is in place, no SSO will be available. That means the user has to authenticate for each SP session individually. Thus, AR is equal to λ ($\rightarrow$ Eq. (6.3)) and $A_{C1,Auth}$ is equal to A ($\rightarrow$ Eq. (6.4)). The average number of IdP sessions is equivalent to the average number of SP sessions, i.e. Eq. (6.5)[7].

$$AR = \lambda \tag{6.3}$$

$$A_{C1,Auth} = A = \frac{\lambda}{\mu} \tag{6.4}$$

$$E\left[N_{act}\right] = A \tag{6.5}$$

6.3.4.2 Case 2: User with Several Identities

N-dimensional Markov chains model the user with several identities. Each dimension models one identity of the user, i.e. $N = N_{Id}$ for a user with N_{Id} identities. Figure 6.22 shows for example the case for $N_{Id} = 2$.

Each state $X(t) = (x_1, \ldots, x_i, \ldots, x_{N_{Id}})$ describes the number of active SP sessions x_i per identity i at time t. The system state transits during an infinitesimal small time interval dt into a state $X(t+dt) = (x_1, \ldots, x_i+1, \ldots, x_{N_{Id}})$ with a rate $\lambda_{Id,i} = \lambda/N_{Id}$, if the user initiates a new SP session with identity i. A transition into state $X(t+dt) = (x_1, \ldots, x_i - 1, \ldots, x_{N_{Id}})$ takes place with rate $x_i \cdot \mu$, if one of the x_i SP sessions using identity i terminates. The combination of the Markov chain model and the Jackson theorem leads to $p(x_1, \ldots, x_i, \ldots, x_{N_{Id}})$ in Eq. (6.6)[8], which quantifies the probability to be in state $X(t) = (x_1, \ldots, x_i, \ldots, x_{N_{Id}})$.

$$p(x_1, x_2, \ldots, x_{N_{Id}}) = p(0, \ldots, 0) \frac{\left(\frac{A}{N_{Id}}\right)^{x_1+x_2+\ldots+x_{N_{Id}}}}{x_1! \cdot x_2! \cdot \ldots \cdot x_{N_{Id}}!} \tag{6.6}$$

$$p(0, \ldots, 0) = \frac{1}{e^A} \tag{6.7}$$

An identity is only activated if it is not used in a prior SP session. That means only transitions from $X(t) = (x_1, \ldots, 0, \ldots, x_{N_{Id}})$ to $X(t+dt) = (x_1, \ldots, 1, \ldots, x_{N_{Id}})$ are of interest for the i-th identity, $i = 1, 2, \ldots, N_{Id}$. In consequence, each state is assigned to one of $2^{N_{Id}}$ different macro states. A macro state is defined as set of states $X(t) = (x_1, \ldots, x_{N_{Id}})$ for which the same set of identities is active. An identity is active, if $x_i > 0, i = 1, \ldots, N_{Id}$.

$$x_i = \begin{cases} 0 & \text{that means identity } i \text{ is not active} \\ > 0 & \text{that means identity } i \text{ is active} \end{cases} \tag{6.8}$$

Hence, the complementary set of identities is inactive. It does not matter, if the actual number of SP sessions, which is indicated by x_i, is greater or equal to 1.

[7] There are no session request losses due to the idealized assumptions of this model

[8] Section A.5.1 provides details on the derivation.

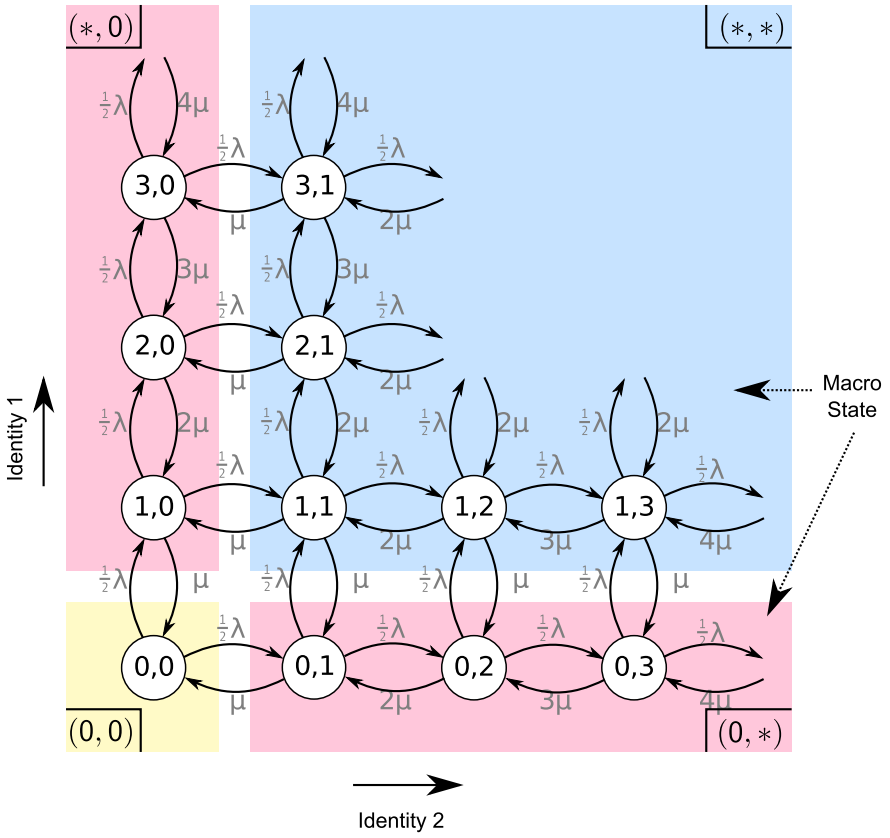


Figure 6.22: Markov State Transition Diagram for two Identities

Figure 6.23 illustrates the macro states for the case of $N_{Id} = 3$. The $*$ symbol indicates an active identity, i.e. $x_i > 0$. The transitions between the macro states, which are indicated by edges, represent identity activation or deactivation events. Activation events in combination with the probability to be in one of the macro states represent the basis for the calculation of AR.

Due to the symmetry of the macro states, only the number of active identities N_{act} matters for the calculation of the probability to reside in a macro state. It does not matter which identities are active. The probability $p_{1,id}$ determines the probability to be in a macro state in which one identity is active. Eq. (6.9) determines the corresponding probability.

$$\begin{aligned} p_{1,Id,C2} &= p(*,0,\ldots,0) = p(0,\ldots,*,\ldots,0) \\ &= \frac{1}{e^A} \sum_{i=1}^{\infty} \frac{\frac{A}{N_{Id}}^i}{i!} \\ &= \frac{1}{e^A} \left(e^{\frac{A}{N_{Id}}} - 1 \right) \end{aligned} \tag{6.9}$$

An identity is activated if a transition from a state in which $N_{act} = a, a \in \mathbb{N}$ to $N_{act} = a + 1$ takes place. Therefore, all states in which $N_{act} = a$ have to be considered. In case of $a = 1$, the probability to be in any state in which $N_{act} = 1$ is $P_{C2}(N_{act} = 1)$ and defined by Eq. (6.10).

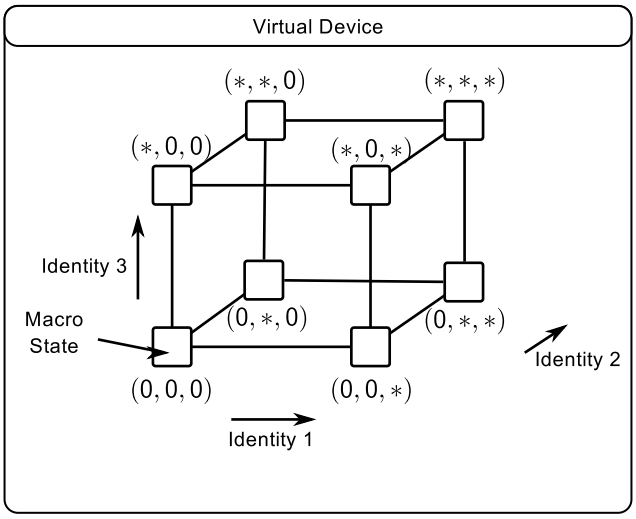


Figure 6.23: Macro States for three Identities and one Virtual Device

$$P_{C2}\left(N_{act}=1\right)=\binom{N_{Id}}{1}p_{1,Id,C2} \tag{6.10}$$

Eq. (6.9) and Eq. (6.10) are generalized by Eq. (6.11) and Eq. (6.12), respectively. Eq. (6.11) defines the probability to be in a state in which the same set of k identities is active. Eq. (6.12) sums up all states in which any set of k identities is active.

$$p_{k,Id,C2} \;=\; p(\underbrace{*,\ldots,*}_{k},0,\ldots,0)=\frac{1}{e^{A}}\cdot(e^{\frac{A}{N_{Id}}}-1)^{k} \tag{6.11}$$

$$\begin{aligned} P_{C2}(N_{act}=k) &= \binom{N_{Id}}{k}p_{k,Id} \\ &= \binom{N_{Id}}{k}\frac{1}{e^{A}}(e^{\frac{A}{N_{Id}}}-1)^{k} \end{aligned} \tag{6.12}$$

Eq. (6.12) is the basis for the calculation of AR, which takes all transitions between the different macro states into account. The transition rate ν_k from a state with k active identities to a state with $k+1$ active identities is given by Eq. (6.13).

$$\nu_k=\frac{N_{Id}-k}{N_{Id}}\lambda \tag{6.13}$$

Eq. (6.13) expresses that the more identities are active the less probable is the activation of an inactive identity. Eq. (6.14) calculates AR_{C2} for Case 2 by summing up the product of all aggregated probabilities $P(N_{act}=k)$ with the corresponding transition rate ν_k.

$$\begin{aligned} AR_{C2} &= \sum_{i=0}^{N_{Id}-1} \nu_i \cdot P_{C2}(N_{act} = i) \\ &= \sum_{i=0}^{N_{Id}-1} \frac{N_{Id} - i}{N_{Id}} \lambda \cdot P_{C2}(N_{act} = i) \\ &= \frac{A \cdot \mu}{e^{\frac{A}{N_{Id}}}} \end{aligned} \tag{6.14}$$

AR_{C2} depends on the load A and the mean service termination rate μ. The normalization of Eq. (6.14) with $1/\mu$ results in the authentication load $A_{Auth,C2}$. Eq. (6.15) defines $A_{Auth,C2}$.

$$A_{Auth,C2} = \frac{AR}{\mu} = \frac{A}{e^{\frac{A}{N_{Id}}}} \tag{6.15}$$

From Eq. (6.15) the maximum authentication load $A_{Auth,C2,max}$ can be derived. $A_{Auth,C2,max}$ represents the worst case mean authentication load in depencency of A. Eq. (6.16) defines $A_{Auth,C2,max}$.

$$A_{Auth,C2,max} = N_{Id} \cdot e^{-1} \tag{6.16}$$

Eq. (6.17) gives the mean number of active identities $E_{C2}[N_{act}, C2]$, i.e. the mean number of active IdP sessions[9].

$$\begin{aligned} E_{C2}[N_{act}] &= \sum_{k=0}^{N_{Id}} k \cdot P_{C2}(N_{act} = k) \\ &= N_{Id}\left(1 - e^{-\frac{A}{N_{Id}}}\right) \end{aligned} \tag{6.17}$$

Eq. (6.18) defines the overhead $R_{C2,N_{Dev}N_{Id}}$ caused by manual authentication of a system without SSO and a system with SSO.

$$R_{C2,N_{Id}} = \frac{A}{A_{Auth,C2}} - 1 \tag{6.18}$$

$$= e^{\frac{A}{N_{Id}}} - 1 \tag{6.19}$$

[9]Details on the calculation are contained in Section A.5.2

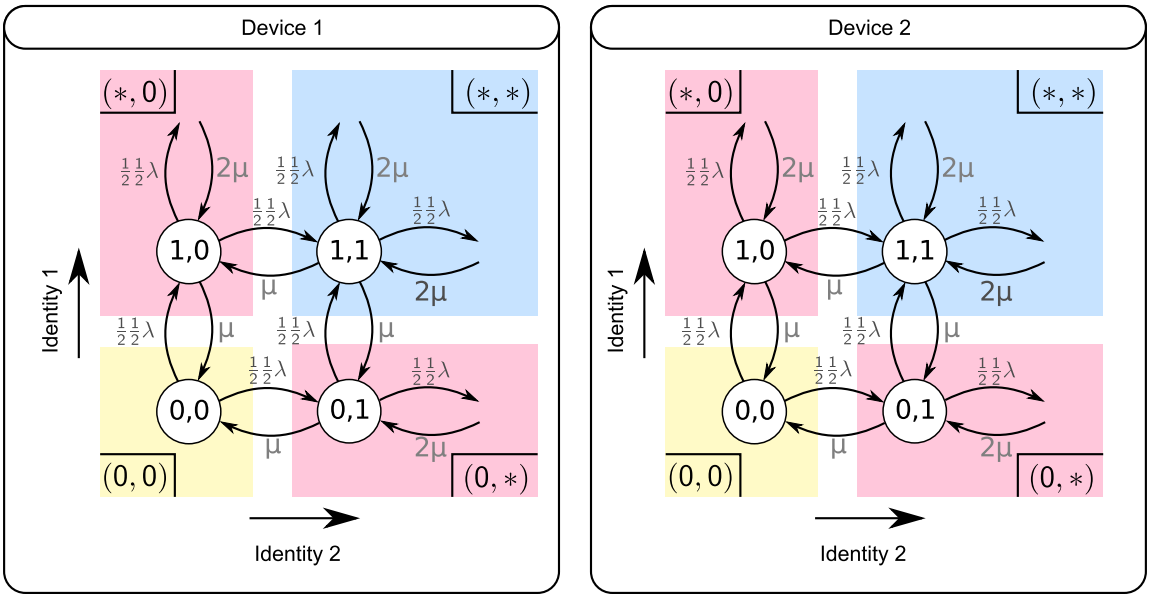


Figure 6.24: Markov State Transition Diagram for two Identities and two Devices

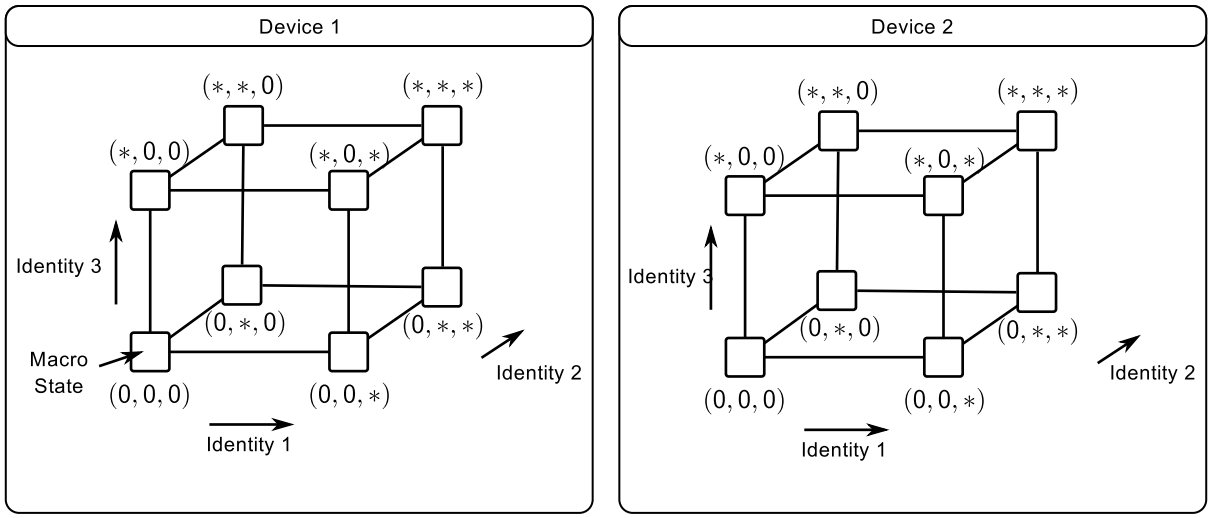


Figure 6.25: Macro States for three Identities and two Independent Devices

6.3.4.3 Case 3: User with Several Identities and Several Devices

This section extends the model introduced in Section 6.3.4.2 with respect to multiple devices. Consideration of multiple devices enlarges the state space. Figure 6.24 shows the extended state space. Each device has to be considered separately. Therefore, the state space is enlarged by a factor N_{Dev}.

As Figure 6.21 indicates, the factor N_{Dev} has to be considered with respect to the interarrival rate λ. Figure 6.24 shows the case for two devices and two identities. Each device has its individual Markov chain. Within each Markov chain the transition rates reflect the number of devices. $N_{Dev} = 2$ results in a reduced transition rate between the different states. Figure 6.25 gives the macro states for the case of two devices and three identities.

Eq. (6.20) gives the probability that k identities are active on a single device. If the interarrival rate λ in P_{C2} is replaced by λ', P_{C3} for an individual device is obtained. Hereby λ' is equal to $\frac{\lambda}{N_{Dev}}$, resulting in Eq. (6.20).

$$\begin{aligned} P_{C3}(N_{act} = k) &= \binom{N_{Id}}{k} p_{k,Id} \\ &= \binom{N_{Id}}{k} \frac{1}{e^{\frac{A}{N_{Dev}}}} \left(e^{\frac{A}{N_{Dev}N_{Id}}} - 1\right)^k \end{aligned} \tag{6.20}$$

This results in $AR_{D,C3}$ in Eq. (6.21) for an individual device.

$$AR_{D,C3} = \frac{1}{N_{Dev}} \frac{A \cdot \mu}{e^{\frac{A}{N_{Dev}N_{Id}}}} \tag{6.21}$$

Eq. (6.22) and Eq. (6.23) give the overall activation rate AR_{C3} and the authentication load $A_{Auth,C3}$ for Case 3, which considers N_{Dev} devices. Eq. (6.24) gives the maximum authentication load.

$$AR_{C3} = \frac{A\mu}{e^{\frac{A}{N_{Dev}N_{Id}}}} \tag{6.22}$$

$$A_{Auth,C3} = \frac{A}{e^{\frac{A}{N_{Dev}N_{Id}}}} \tag{6.23}$$

$$A_{Auth,max,C3} = N_{Dev}N_{Id}e^{-1} \tag{6.24}$$

For $N_{Dev} = 1$, Eq. (6.22) is equal to Eq. (6.14). This confirms the plausibility of Eq. (6.22).

Eq. (6.25) gives the mean number of active IdP session across all devices.

$$\begin{aligned} E_{C3}[N_{act}] &= N_{Dev} \sum_{k=0}^{N_{Id}} k \cdot P_{C3}(N_{act} = k) \\ &= N_{Dev}N_{Id} \left(1 - e^{-\frac{A}{N_{Dev}N_{Id}}}\right) \end{aligned} \tag{6.25}$$

Eq. (6.26) determines the ratio $R_{C3,N_{Dev,1},N_{Id,1},N_{Dev,2},N_{Id,2}}$ of two different settings, i.e. setting 1 and setting 2, for the authentication load $A_{Auth,C3}$. The settings differ with respect to the number of identities and devices, i.e. the user has $N_{Dev,x}$ devices and $N_{Id,x}$ identities for setting $x, x \in \{1, 2\}$. If the virtual device is in place, Eq. (6.27) simplifies Eq. (6.26) with $N_{Dev,2} = 1$ and $N_{Id,2} = N_{Id,1} = N_{Id}$ and allows the evaluation of the virtual device concept.

$$\begin{aligned} R_{C3,N_{Dev,1},N_{Id,1},N_{Dev,2},N_{Id,2}} &= \frac{A_{Auth,C3}(N_{Dev,2}, N_{Id,2})}{A_{Auth,C3}(N_{Dev,1}, N_{Id,1})} - 1 \\ &= e^{A\left(\frac{1}{N_{Dev,1}N_{Id,1}} - \frac{1}{N_{Dev,2}N_{Id,2}}\right)} - 1 \end{aligned} \tag{6.26}$$

$$R_{C3,N_{Dev},N_{Id}} = e^{A\left(\frac{1}{N_{Dev}N_{Id}} - \frac{1}{N_{Id}}\right)} - 1 \tag{6.27}$$

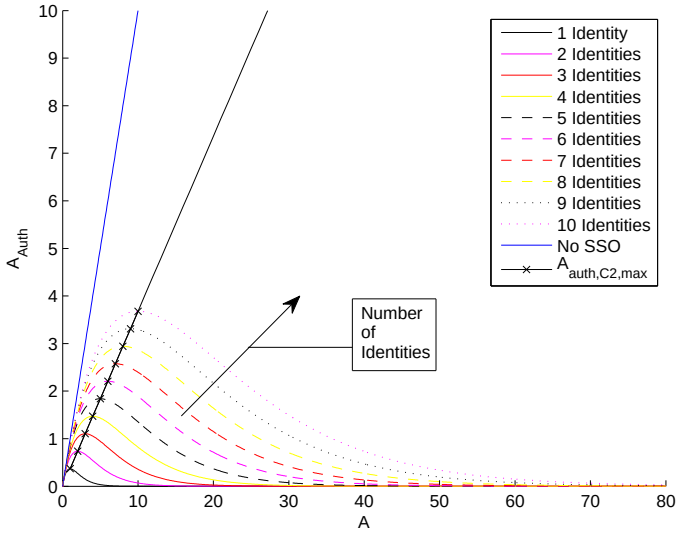


Figure 6.26: Authentication Load A_{Auth} in Dependency of SSO and Number of Identities

6.3.5 Evaluation Results

This section examines the evaluation goals outlined in Figure 6.17. Section 6.3.5.1 evaluates the benefits of SSO, Section 6.3.5.2 regards the impact of several identities and Section 6.3.5.3 examines the consequences of several devices.

6.3.5.1 Consequences of SSO

Figure 6.26 shows A_{Auth} in dependence of the number of identities and dependent on the existence of a SSO mechanism. If no SSO is in place, A_{Auth} increases linearly with the number of SP sessions, which is indicated by A. For low values of A, i.e. a low number of SP sessions, SSO does not provide benefits. It is obvious that the higher A is, the higher are the benefits of SSO mechanisms. For SSO in low load situations, there is a high probability that a new SP session triggers the activation of the identity. That means no other SP session exists that has triggered the establishment of the IdP session. The higher the load situation, the higher is the probability that a SP session already exists resulting in no need to activate the identity.

Figure 6.27, which shows the overhead according to Eq. (6.18), confirms these results. With an increasing A, the overhead without SSO increases significantly. In addition, Figure 6.28 shows that if SSO is applied, the number of active identities converges to the number of available identities. For example, if the user has only one identity, $E[N_{act}]$ converges to one.

6.3.5.2 Consequences of Several Identities

In addition to the benefits of SSO, Figure 6.26 shows the impact of splitting SP sessions across several identities. The more identities the user has, the higher $A_{Auth,C2}$. That means the advantages of having multiple identities result in a higher authentication effort for the user. Fig-

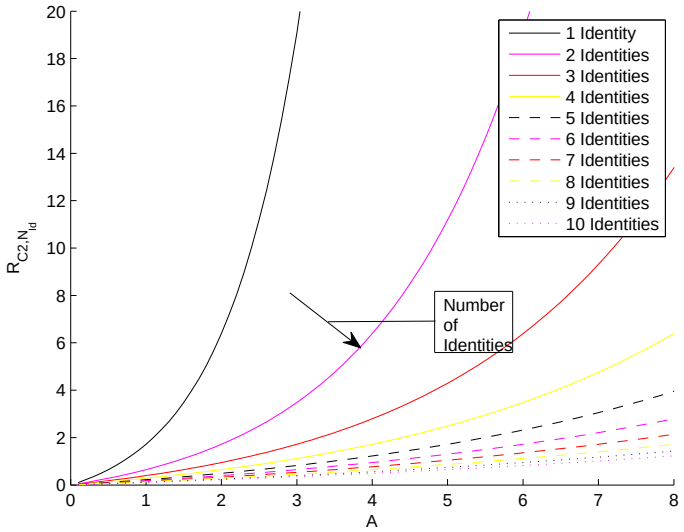


Figure 6.27: Overhead $R_{C2,N_{Id}}$ in Dependency of Number of Identities

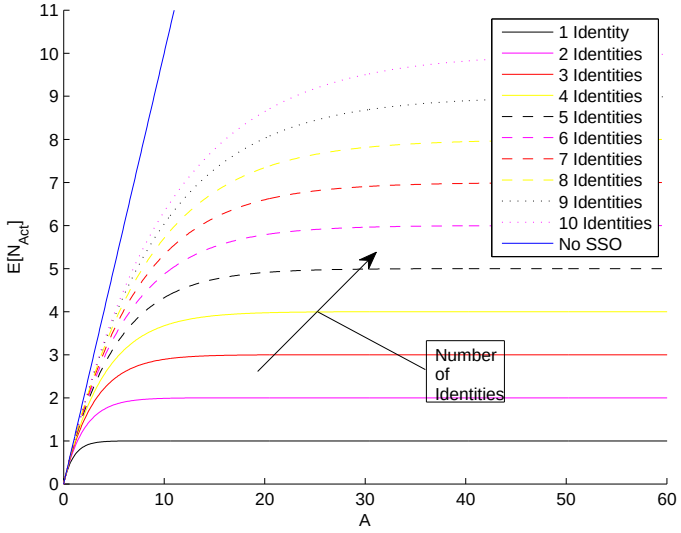
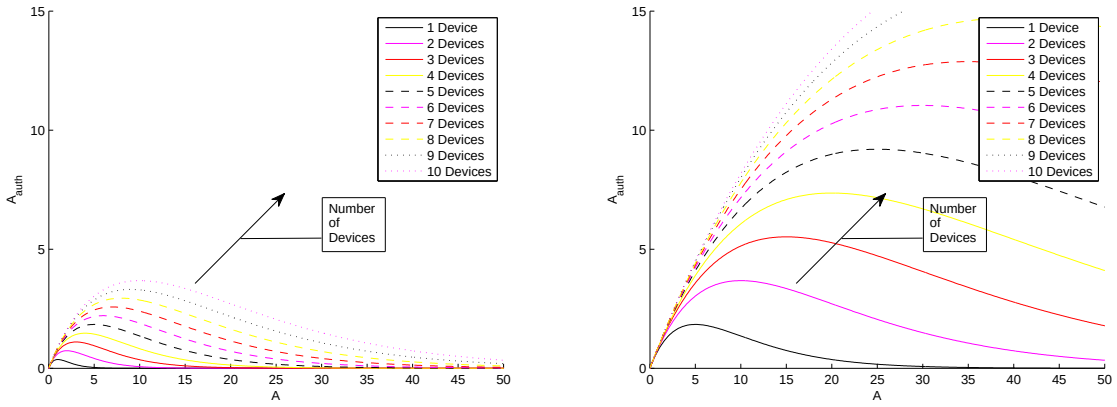


Figure 6.28: Mean Number of Active Identities $E_{C2}[N_{act}]$ in Dependency of SSO and Number of Identities



(a) Authentication Load $A_{Auth,C3}$ in Dependency of Number of Devices for 1 Identity

(b) Authentication Load $A_{Auth,C3}$ in Dependency of Number of Devices for 5 Identities

Figure 6.29: Authentication Load $A_{Auth,C3}$

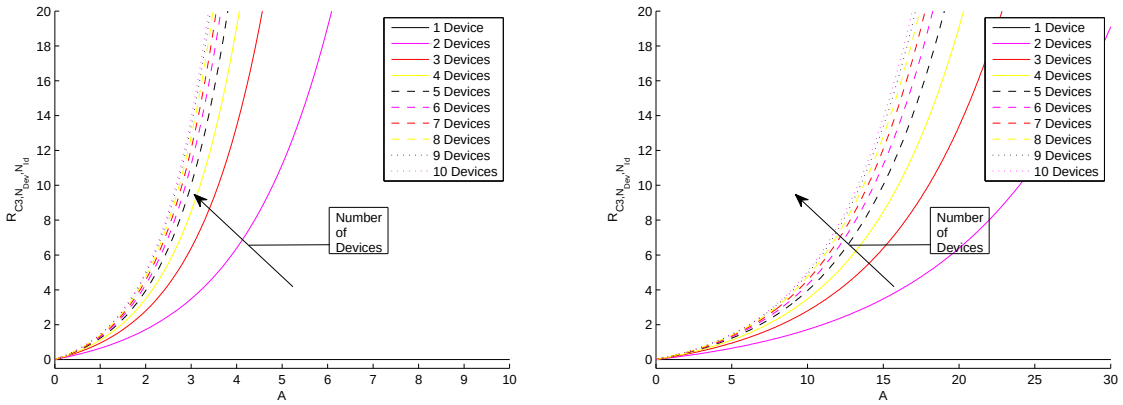
ure 6.27 illustrates that the overhead $R_{C2,N_{Id}}$ decreases with increasing number of identities. In consequence the decision to have multiple identities must be balanced against the increased authentication effort. From the perspective of the IdP, the number of existing IdP sessions increases (→ Figure 6.28). That means the decision to offer several independent identities to users must be considered with respect to capacity planning. The issue of capacity planning is not detailed here.

6.3.5.3 Consequences of Several Devices

Section 6.3.5.2 already assumed the existence of a virtual device. In the following the benefits of the virtual device are evaluated. This is achieved by comparison of SSO within the virtual device against SSO of independent devices. Figure 6.29(a) and Figure 6.29(b) show $A_{Auth,C3}$ for a user with one identity and for a user with five identities, respectively. The curve "1 Device" represents the case of the virtual device, i.e. all IdP sessions are established on the same device. An increasing number of devices results in the establishment of IdP sessions for the given identity on all devices. In consequence, $A_{Auth,C3}$ increases (→ Figure 6.29). The limit of $E[N_{Act}]$ converges to $N_{Dev} \cdot N_{Id}$ as Eq. (6.25) defines. Figure 6.30 shows the overhead of operating the devices independently of each other. The curve for "1 Device" is the baseline and does not result in overhead. The more devices a user has, the larger the overhead to use these devices.

6.3.6 Summary

This section evaluated the consequences of having several identities and several devices. It introduced a simple analytical model, which is based on Markov chains. The analytical model allows the quantification of the benefits provided by the virtual devices. The analytical model confirmed the advantages with respect to SSO across multiple devices. Even if the analytical model is simple, it allowed showing the relationships between the different degrees of freedom,



(a) Overhead $R_{C3,N_{Dev},1}$ in Dependency of Number of Devices for 1 Identity

(b) Overhead $R_{C3,N_{Dev},5}$ in Dependency of Number of Devices for 5 Identities

Figure 6.30: Overhead $R_{C3,N_{Dev},N_{Id}}$

like the number of devices or the number of identities. It can be stated that the introduction of the virtual device provides significant advantages with respect to the number of required authentication procedures. Thus it increases the usability.

6.4 Summary

The evaluation of the designed architecture and the corresponding mechanism consisted of three parts. (1) The functional evaluation showed that the architecture is able to fulfill the projected scenarios from Section 4.1 and that it covers all established functional and non-functional requirements. The prototype successfully confirmed the feasibility of the architecture and the interworking with existing IdM systems. (2) The security evaluation identified threats and analyzed the corresponding attack trees. The seven examined attacks are countered with appropriate security mechanisms. (3) The performance evaluation complemented the overall evaluation with an analytical model. The analytical model allows the quantification of authentication procedures that have to be completed by the user. Moreover, it showed the basic relationships between the IdP, the user, the user's devices and the user's identities. All three evaluation parts have been successfully passed. No shortcomings have been revealed.

7 Conclusions and Outlook

This thesis presented the design of an architecture for multi-device identity management. Multi-device identity management is motivated by the need to increase usability and security for users consuming services with several devices. The designed architecture extends existing identity management systems with mechanisms that enable collaboration between the user's devices.

Key concepts enable a modular design to counter the complexity. Hereby, a key concept has been a set of self-contained functionality to address a couple of coherent aspects. The introduced key concepts depended on each other to realize the architecture. Moreover, key concepts served as guidelines for the further refinement of the architecture. Three key concepts have been introduced: Virtual Device concept, IdP and SP Session Split concept, and the Multi-Device IdM concept. The Virtual Device concept provided basic functionality to enable the collaboration of devices. The IdP and SP Session Split concept enabled the distribution of functionality between the different devices. Finally, the Multi-device IdM concept made use of the other two key concepts in order to extend IdM across multiple devices.

The collaboration of user's devices requires a high level of security to provide distributed multi-device IdM. Only if sufficient security mechanisms are in place the overall security with respect to identity management can be guaranteed. Therefore, this thesis proposed a methodology for the design of secure systems. An iterative process consisting of different phases has been applied. First, security threats have been identified during the design phase to counter potential attacks on assets in an early stage. Second, a security architecture was designed consisting of access control mechanisms, security associations between the devices and a mechanism for the determination of the security level of each device. Finally, the security was subject to evaluation from different perspectives.

Chapter 2 introduced the basic security terminology in order to provide the background for the subsequent sections and chapters. It gave an overview on existing security mechanisms and classified existing technologies. The main focus regarding security mechanisms was on authentication. A detailed introduction of authentication mechanisms and protocols formed the basis for the understanding of identity management systems. With the introduction and classification of design methods for secure system design, the basics for the system design have been laid. Moreover, the introduction of different evaluation methods provided the background for the security evaluation in Chapter 6.

Chapter 3 provided the fundamentals on identity management. A reference architecture has been drafted to show the relationships of the different participating players – the user, the identity provider, and the service provider – and to show the functional blocks of each player. Based

on the reference model, existing base technologies for identity management and existing identity management systems have been introduced. Related work on the usage of identity management systems with multiple devices has been introduced and categorized. An outlook on the designed solution in comparison to related work was given.

Chapter 4 developed the architecture to enable identity management across multiple devices. Based on five usage scenarios, challenges have been defined, which have to be covered by the architecture. The definition of the usage scenarios and the key concepts guided the requirements definition process. Functional, non-functional and security requirements have been identified in a semi-formal way. The security requirements have been identified based on a threat analysis. The UML model of the architecture defined and detailed four functional blocks. Each functional block realized coherent functionality to address the key concepts.

Chapter 5 detailed selected functional blocks, which are either required to provide multi-device identity management or to determine the security of the system. Regarding multi-device identity management two subsections detailed the identity filtering mechanism and the protocols to make use of identities across devices. The identity filtering mechanism proposed an algorithm consisting of two phases. The first phase filters identities in dependency of the device characteristics. The second phase filters identities based on the requirements of the service provider and the availability of devices. Moreover, a ranking process was proposed to rank identities according to usability criteria. Three different protocols have been specified for the exchange of identity information, for the activation of identities on remote devices and for the retrieval of assertions from remote devices. Regarding the virtual device concept, this thesis only specified the overall framework and detailed the security architecture. The security architecture consists of mechanisms to setup secure channels between the devices belonging to a virtual device and mechanisms to add or remove individual devices from the virtual device.

Chapter 6 evaluated the proposed architecture and the developed mechanisms and protocols from three different perspectives.

- Functional Evaluation: The functional evaluation in Section 6.1 validated the feasibility of the usage scenarios with the proposed architecture. Moreover, the functional evaluation verified the fulfillment of all requirements that have been defined. Finally, the prototypical implementation served as proof-of-concept and showed the feasibility to implement the proposed architecture in correspondence with existing identity management systems.

- Security Evaluation: In Section 6.2 the security of the designed architecture was evaluated. Three different evaluation steps showed that the security requirements have been sufficiently addressed. The first step extended the threat analysis of Chapter 4 with the details of the defined architecture and the concretized mechanisms and protocols of Chapter 5. The second step took the perspective of an attacker and identified valuable attack goals. Attack trees identified the necessary actions to achieve the attack goal. The analysis of the attack trees showed that all actions are either countered by appropriate security mechanisms or are considered as practically infeasible. The third security evaluation step identified misuse cases. For each misuse case additional security mechanisms have been outlined.

- In Section 6.3, the benefits of the multi-device identity management solution have been evaluated with an analytical performance model. The performance model has been established with simplified assumptions regarding the service consumption behavior and regarding the devices. The obtained equations have been used to quantify the impact of having several identities per user and the impact of having a virtual device. The results showed that a virtual device reduces the authentication effort for users and thus reduces the number of identity provider sessions that have to be managed simultaneously by the identity provider. This improves usability and therewith security..

Concluding, the designed architecture fulfills all functional and non-functional requirements. Moreover, the designed security mechanisms counter all identified threats. The virtual device concept provides significant advantages regarding the usability and security.

The novelty of this thesis is the design of an architecture to enable multi-device IdM combined with a corresponding design and evaluation methodology. The architecture is based on the explicit collaboration of a user's devices and the support for multiple identities. The collaboration of user's devices and the Multi-device IdM concept enables SSO from multiple devices. This approach differs from existing work in the following ways: (1) The taken approach does not synchronize credentials between the devices. Instead, every device obtains only the view and the credentials that are required to consume services at a certain point of time. (2) In contrast to approaches that are based on personal authentication devices, the designed architecture provides more degrees of freedom. (3) The provided degree of flexibility with respect to the collaboration of devices is higher than with solutions that target to handle inappropriate input capabilities. The support for multiple identities extends existing work with respect to multiple devices. In addition to the consideration of the usage context, the proposed architecture takes the device into account for the decision whether an identity can be used or not. The tailored design and evaluation methodology combines best-practices from software and security engineering. It has been applied to design and evaluate the proposed architecture with respect to functionality and security.

Further work can take different directions. First, the basic identity filtering mechanisms could be extended to consider privacy in an adequate way. Second, the virtual device concept could not only be used to provide multi-device identity management. It could also be applied to provide session transfer mechanisms, which are also security critical and require detailed knowledge of the participating devices in order to appropriately adapt the service sessions. This would complement the architecture proposed in [BKM09]. Third, the virtual device concept could be extended with respect to the heterogeneity of devices. In particular for low-performance devices, the proposed security architecture might be too complex. Fourth, the consequences of device virtualization might be evaluated. In particular recent technologies allow the splitting of devices, into several virtual ones. For example it is possible to split a smartphone into several virtual ones. Finally, the proposed performance model could be extended with respect to more realistic service consumption models and with respect to different characteristics.

A Details

This chapter consists of five parts. Appendix A.1 provides additional details on the requirements. Appendix A.2 details the security architecture of the virtual devices. Appendix A.3 enumerates how the requirements have been addressed. Appendix A.4 provides additional details on the security evaluation. Finally, Appendix A.5 elaborates some of the equations derived in the performance evaluation.

A.1 Requirements

A.1.1 Functional Requirements

Device Discovery	DM-DD-1
Short Description	Device Discovery in General (c.f. R4)
Full Description	Devices belonging to one user should be able discover each other.
Stakeholder	User
Rational	For the later protocols it is required that it is known which devices of one user are available.
Dependencies	

Device Discovery	DM-DD-2
Short Description	Proximity Detection
Full Description	It is required that a device can detect other devices in close proximity.
Stakeholder	User
Rational	If the authentication is relayed from one device to another device, the device has to be close by in order to allow the user to manually authenticate.
Dependencies	DM-DD-2

Device Discovery	DM-DD-3
Short Description	Device Discovery should not reveal any information
Full Description	For the device discovery any mechanism that reveals information to unauthorized devices has to be avoided.
Stakeholder	User
Rational	With information about devices belonging to the virtual device, an attacker could lounge several attacks.
Dependencies	DM-VDM-3

Security Management	DM-SA-1
Short Description	Mutual Authentication of Devices
Full Description	A mechanism is required to mutually authenticate two devices. It is required that devices that successfully authenticate each other are part of a virtual device.
Stakeholder	
Rational	Mutual authentication is one prerequisite for the secure exchange of data between devices. Only after mutual authentication, sufficient assurance exists that the other device is the one it pretends to be.
Dependencies	DM-VDM-1

Security Management	DM-SA-2
Short Description	Secure Storage of Credentials
Full Description	A mechanism is required to store credentials in a secure way. Hereby, secure means that the credentials can not be obtained from a device by an unauthorized user.
Stakeholder	
Rational	For the mutual authentication of devices, some kind of credential is required. This credential must not be accessed in unauthorized way. In addition, the secure storage can be used for other security-critical information.
Dependencies	

Device Management	DM-VDM-1
Short Description	Adding a Device to VD
Full Description	A mechanism is required to add a device to the virtual device. The mechanism must be manually triggered and confirmed by the user. Adding a device to a Virtual Device must set the prerequisites to enable mutual authentication of devices.
Stakeholder	User
Rational	Users obtain new devices. It must be possible to add these devices to the virtual device. The manual trigger is required to avoid the unintentional adding of devices.
Dependencies	DM-SA-1

Device Management	DM-VDM-2
Short Description	Removing a Device from VD
Full Description	A mechanism is required to remove a device from the virtual device. The mechanism must be manually triggered. After removing a device from the virtual device, the mutual authentication of devices must fail.
Stakeholder	User
Rational	Users loss devices, devices break or got discarded because of technological progress.
Dependencies	DM-SA-1

Device Management	DM-VDM-3
Short Description	Device Identity
Full Description	For the usage within the virtual device it is required that devices can be identified. Identification requires a - within a virtual device - unique identifier.
Stakeholder	
Rational	
Dependencies	DM-SA-1

Device Characteristics	DM-DC-1
Short Description	Capturing of Device Characteristics
Full Description	A module is required that captures device characteristics. Device characteristics describe the software and hardware capabilities. Including aspects that are relevant for security. It must be differentiated between data that is subject to frequent changes and data that is subject to almost no changes.
Stakeholder	User
Rational	The characteristics of a device serve as input for the determination, which authentication methods are supported and for the determination of the security level of a device.
Dependencies	DM-DC-4

Device Characteristics	DM-DC-2
Short Description	Manual Editing of Device Characteristics
Full Description	An interface is required that allows a user to enter device characteristics manually.
Stakeholder	User
Rational	Since, we assume that we cannot capture all information automatically an interface is required that allows a user to add information about devices. In addition, it should be possible to manually modify automatically captured device information.
Dependencies	DM-DC-4

Device Characteristics	DM-DC-3
Short Description	Selection of Usage Context
Full Description	The user should be able to configure the usage context of a device by using an appropriate interface. It should be possible that a device has several usage contexts in parallel.
Stakeholder	User
Rational	The usage context cannot be captured automatically. Therefore, it is required to have an interface that allows the configuration and selection of the usage context. Since, a device can be used in several usage context (private and business) in parallel for different services, it is required to support more than one usage context.
Dependencies	

Device Characteristics	DM-DC-4
Short Description	Data Format for Device Characteristics
Full Description	Information between devices have to be exchanged. Thus it is necessary to encode information in a to be specified format. Hereby, the exchange has to take place in a secure way (confidentiality and integrity protected)
Stakeholder	User
Rational	For distributed decision making it is required to have information about the capabilities of other devices.
Dependencies	DM-DC-1

Data Management	DM-DA-1
Short Description	Mechanism for Data Exchange
Full Description	A mechanism is required that enables the data exchange between devices. Hereby it is required that devices have all data regarding identities and other devices available in order to draw appropriate decisions.
Stakeholder	User
Rational	For distributed decision making it is required to have information about the capabilities of other devices.
Dependencies	

Assertion Exchange	IdM-AE-1
Short Description	Protocol for the request of authentication assertions
Full Description	A protocol is needed that allows to request authentication assertions from another device that has an established IdP Session.
Stakeholder	User
Rational	There is no need to have an IdP session on each device. This increases usability and security by avoiding the need to enter authentication credentials on every device.
Dependencies	

Assertion Exchange	IdM-AE-2
Short Description	Authorization based on Device ID
Full Description	If another device requests an assertion, the providing device must check whether the requesting device is authorized to obtain the token. In the decision the following information should be considered: Usage Context, Security Level.
Stakeholder	User
Rational	It has to be avoided that every identity can be used on every device. In particular, it must not be possible that a device without a sufficient security level obtains more privileges than permitted.
Dependencies	DM-VDM-3

Assertion Exchange	IdM-AE-3
Short Description	Information about Usage Purpose
Full Description	The requester must provide information on the purpose for which the token is required. This information might be used to Authorization decisions.
Stakeholder	User
Rational	Even if the the result of the identity filtering process allows the usage of the identity on the given device. It is required that the requested device gets additional information to authorize the decision.
Dependencies	

Assertion Exchange	IdM-AE-4
Short Description	Manual Confirmation
Full Description	The user should have the possibility to enable an optional manual confirmation on the requested device. The manual confirmation should be lightweight, i.e. pressing a button or clicking a confirmation dialog.
Stakeholder	User
Rational	This feature requirement is considered as an additional security mechanism. It makes the authentication process more transparent for the user. It is assumed that the usability is not degraded, because the confirmation procedure is very lightweight.
Dependencies	

Assertion Exchange	IdM-AE-5
Short Description	Feedback if Manual Confirmation is required
Full Description	If manual confirmation is activated on the requested device, the user must obtain feedback on which device the manual confirmation is required.
Stakeholder	User
Rational	In case of several devices it might become difficult to identify the device on which the confirmation should take place.
Dependencies	

Identity Activation	IdM-IA-1
Short Description	Protocol for identity activation (c.f. R3)
Full Description	It must be possible to trigger the activation of an identity on another device.
Stakeholder	User
Rational	In order to use the most secure device for authentication a mechanism must exist to activate identities on a remote device.
Dependencies	

Identity Activation	IdM-IA-2
Short Description	Authorization based on Device ID
Full Description	If identity activation is requested, the requested device must check whether the requesting device is authorized to activate the identity.
Stakeholder	User
Rational	The requested device must not trust the requesting device regarding the authorization.
Dependencies	DM-VDM-3

Identity Activation	IdM-IA-3
Short Description	Feedback on Device triggered for Identity Activation
Full Description	The user that triggers the identity activation on another device must obtain feedback about the device on which the activation, i.e. authentication takes place.
Stakeholder	User
Rational	In case of several devices it might become difficult to identify the device on which the activation should take place.
Dependencies	

Identity Management	IdM-IM-1
Short Description	Capture existing identities
Full Description	The system has to know the identities an user has and store this information. The system has to support different ways to obtain information about user's identities
Stakeholder	User
Rational	For the identity filtering it is required to know all identities the user has in usage.
Dependencies	IdM-IM-3, IdM-IM-4

Identity Management	IdM-IM-2
Short Description	Store meta data on identities
Full Description	The system must support the storage of meta data on identities. Among the meta data is the usage context of the identity, the required methods for authentication, the security requirements.
Stakeholder	User
Rational	This information is required for the identity filtering process. Based on this information the usage of identities is restricted to a certain set of devices.
Dependencies	IdM-IM-5

Identity Management	IdM-IM-3
Short Description	Automatic capturing of existing identities
Full Description	The system should have a plug-in, that observes the user to obtain information about identities the user is using. In particular if the user is using a new identity, it should be automatically detected.
Stakeholder	User
Rational	To keep the system as simple as possible from the user perspective, a mechanism is required to automatically detect the user identities.
Dependencies	IdM-IM-1, IdM-IM-4

Identity Management	IdM-IM-4
Short Description	Manual adding and removal of user identities
Full Description	It should be possible that the user adds a new identity to the existing identities. Removal of identities should also be possible.
Stakeholder	User
Rational	To support cases in which the automatic capturing of existing identities is not possible, functionality to add identities is required. The removal of identities is required to remove identities that should not be usable anymore. The reasons for removal are for example: Compromised identities, job change, ?
Dependencies	IdM-IM-1, IdM-IM-3

Identity Management	IdM-IM-5
Short Description	Manual modification of identity information
Full Description	It should be possible to manually modify the stored identity information. An appropriate interface to list all identities the user has together with the corresponding properties is required. The provided information should be modifiable.
Stakeholder	User
Rational	It is assumed that not all information on identities can be automatically captured. It is therefore required to provide an interface to modify this information.
Dependencies	IdM-IM-2

Identity Management	IdM-IM-6
Short Description	Graphical User Interface of Identity Selection
Full Description	When the SP requests authentication, the user should select one identity out of a list of suitable identities by using a graphical interface.
Stakeholder	User
Rational	To make identity selection usable, a graphical interface is required.
Dependencies	IdM-IM-7, IdM-IM-8

Identity Management	IdM-IM-7
Short Description	List of selectable identities
Full Description	The user should only be able to select usable identities. Usable means identities that match the SP requirements regarding attributes and that match the usage context, security level, and provide the required authentication capabilities.
Stakeholder	User
Rational	Providing an overview of usable identities simplifies the selection of an appropriate identity.
Dependencies	IdM-IM-6

Identity Management	IdM-IM-8
Short Description	Priorities of selectable identities
Full Description	The list of selectable identities should be sortable according to different criteria. Among the criteria are: - Avoiding the need to manually reauthenticate. If identity activation is needed, the authentication should take place on the most secure device. Privacy is not considered as a criteria.
Stakeholder	User
Rational	The order of the list of selectable identities should increase the usability. That is the user gets identities presented for which the authentication overhead is low. Privacy is not considered in a first place due to the complexity of privacy decisions.
Dependencies	IdM-IM-6, IdM-IM-7

A.1.2 Nonfunctional Requirements

Security	NF-1
Short Description	Data Minimization Principle
Full Description	Reveal only as much information to others as really necessary.
Stakeholder	Legislative
Rational	Data minimization restricts the view that an individual device can obtain. Depending on the device, only the actually needed information is stored.
Dependencies	

Usability	NF-2
Short Description	High Usability
Full Description	The solution must have a high degree of usability ([1]. This includes that the user can trace what happens in the system. For example on which device authentication should take place.
Stakeholder	User
Rational	Usability is essential for the system design.
Dependencies	

Performance	NF-3
Short Description	No Performance Penalties
Full Description	The cooperation of devices for authentication must not have a significant performance penalty.
Stakeholder	User
Rational	If time required for the exchange of messaging and corresponding processing times is higher than manual authentication, the benefit of the proposed solution is low.
Dependencies	

Security	NF-4
Short Description	No Degradation of Security
Full Description	The security of the overall solution must be higher than having individual devices.
Stakeholder	IdP, SP, User
Rational	A level of security is essential for the acceptance of the solution.
Dependencies	

Availability	NF-5
Short Description	No Dependency on Single Devices
Full Description	The system must not fail, if one device is not available. Hereby failing means that the user cannot use any service. Some services might depend on a single device (e.g. because of unique authentication methods)
Stakeholder	User
Rational	Devices can appear and disappear, therefore it must be possible that the system works in case of unavailability of single devices.
Dependencies	

A.1.3 Detailed Threat Description

A1-T1 Illegal Service Consumption	
Objective: An attacker uses a user's device to consume services on behalf of the user. Prerequisite: An attacker steals a device or the user loses one of his devices. In addition, the following conditions have to be fulfilled. 1) IdP session active on stolen device 2) IdP session active on any other device that is part of the virtual device and that can request the required permission to request the service.	
Attacker: External Attacker	Victim: User, IdP, SP
Impact: High	Precondition: High
Priority: High	Scope: In
Security Requirements: SR-1, SR-2	

A1-T2 Data Modification	
Objective: An attacker modifies data stored on a user device. Modifying data can be exploited to prepare other attacks. Examples for modification of data are meta-data about identities and meta-data about devices. Prerequisite: An attacker might steal a device or the user might lose one of his devices. Such a device might be used to modify the data stored on a device.	
Attacker: External Attacker	Victim: User
Impact: High	Precondition: High
Priority: Low	Scope: Out
Security Requirements: SR_1, SR_2	

A1-T3 Repudiation of Service Usage	
Objective: An attacker consumes services on behalf of the user (c.f. A1_T1). The user repudiates the service consumption. Prerequisite: Realization of A1_T1	
Attacker: External Attacker	Victim: User
Impact: Low	Precondition: High
Priority: Medium	Scope: In
Security Requirements: SR_3	

A1-T4 Disclosure of User Data	
Objective: An attacker discloses data stored on user devices Prerequisite: An attacker steals a device or the user loses one of his devices.	
Attacker: External Attacker	Victim: User
Impact: High	Precondition: High
Priority: Low	Scope: Out
Security Requirements: SR_1, SR_2	

A1-T5 DoS on Virtual Device Operation I	
Objective: An disturbs the regular operation of the virtual device by the inherently provided mechanisms. For example an attacker might request activate identities on another user device, which bothers the user with the request for authentication. Prerequisite: An attacker steals a device or the user loses one of his devices.	
Attacker: External Attacker	Victim: User, IdP
Impact: Low	Precondition: High
Priority: Medium	Scope: In
Security Requirements: SR_1, SR_2, SR_4, SR_6	

A1-T6 DoS on Virtual Device Operation II	
Objective: An attacker disable one or several user identities by repetitively requesting authentication and supplying wrong credentials. Prerequisite: An attacker steals a device or the user loses one of his devices.	
Attacker: External Attacker	Victim: User, IdP
Impact: Low	Precondition: High
Priority: Medium	Scope: In
Security Requirements: SR_6	

A2-T1 Modification of Virtual Device Composition	
Objective: An attacker modifies the virtual device composition. This includes a adding a foreign device to the virtual device composition, which compromises the security, but also removing a device from the virtual device composition, which decreases the availability. Prerequisite: An attacker gets access to the data structures that are required to manage the virtual device composition.	
Attacker: External Attacker	Victim: User
Impact: High	Precondition: Medium
Priority: High	Scope: In
Security Requirements: SR_5, SR_7	

A2-T2 Repudiation of Virtual Device Modification	
Objective: An attacker modifies the virtual devices and the user of the virtual device repudiates the modifications. Prerequisite: An attacker gets access to the data structures that are required to manage the virtual device composition.	
Attacker: External Attacker, User	Victim: User
Impact: Medium	Precondition: Medium
Priority: Medium	Scope: In
Security Requirements: SR_8	

A2-T3 Disclosure of Information Virtual Device Composition	
Objective: An attacker discloses information about the virtual device composition.This information can serve as enabler for other threats (e.g. A2_T1) and represents a privacy threat. Prerequisite: An attacker gets access to the data structures that are required to manage the virtual device composition.	
Attacker: External Attacker	Victim: User
Impact: Medium	Precondition: High
Priority: High	Scope: In
Security Requirements: SR_9	

A3-T1 IdP Token Interception	
Objective: An attacker intercepts the IdP Token to spoof the user identity and consume services on behalf of the user. Prerequisite: The attacker must know between which parties, devices and functional components the IdP Token is transferred and stored, respectively.	
Attacker: External Attacker, User	Victim: User, IdP
Impact: High	Precondition: Medium
Priority: High	Scope: In
Security Requirements: SR_10, SR_11	

A3-T2 IdP Token Modification	
Objective: An attacker modifies the IdP Token to disturb the regular operation. Modification of the IdP Token might lead to invalidation of the IdP Session. Prerequisite: The attacker must know between which parties, devices and functional components the IdP Token is transferred and stored, respectively.	
Attacker: External Attacker	Victim:
Impact: Low	Precondition: Medium
Priority: Medium	Scope: In
Security Requirements: SR_10, SR_11	

A4-T1 SP Token Interception	
Objective: An attacker intercepts the SP Token to spoof the user identity and consume a service on behalf of the user. Prerequisite: The attacker must know between which parties, devices and functional components the SP Token is transferred and stored, respectively.	
Attacker: External Attacker, User	Victim: User, SP
Impact: Medium	Precondition: Medium
Priority: Medium	Scope: In
Security Requirements: SR_12, SR_13	

A4-T2 SP Token Modification	
Objective: An attacker modifies the SP Token to disturb the regular operation. Modification of SP Token leads to invalidation of the SP Session Prerequisite: The attacker must know between which parties, devices and fucntional components the SP Token is transferred and stored.	
Attacker: External Attacker, User	Victim: User, SP
Impact: Low	Precondition: Medium
Priority: Medium	Scope: In
Security Requirements: SR_12, SR_13	

A5-T1 Illegal Service Consumption	
Objective: An attacker uses user's authentication credentials to consume service on behalf of the user and to get access to user data bound to user's identity. Prerequisite: The attacker must obtain the authentication credentials. Depending on the kind of authentication credentials, the difficulty to obtain them is different. The authentication credentials cannot only be obtained from the user, but also from the IdP.	
Attacker: External Attacker, SP	Victim: User, IdP
Impact: High	Precondition: High
Priority: High	Scope: In
Security Requirements: SR_14, SR_15	

A5-T2 Repudiation of Authentication	
Objective: An attacker uses services on behalf of the user. The user repudiates the usage of authentication credentials. Prerequisite: The attacker must obtain the authentication credentials.	
Attacker: External Attacker, User	Victim: User, IdP
Impact: High	Precondition: High
Priority: High	Scope: In
Security Requirements: SR_16	

A5-T3 Usage of wrong Authentication Credentials	
Objective: An attacker uses intentionally the wrong credentials to reduce the availability of services provided by the IdP. Prerequisite: In case of an untargeted attack against the identities provided by the IdP, only knowledge about the entry point of the authentication process is required. In case of an targeted attack, i.e. against a specific identity, information about the identity is required, e.g. the username.	
Attacker: External Attacker	Victim: User, IdP
Impact: Low	Precondition: Low
Priority: Low	Scope: Out
Security Requirements:A6-T1	

A6-T2 Derive Information from SP Assertion	
Objective: An attacker derives information about the user, the service, the IdP, etc. by exploiting the information contained in an SP Assertion. Prerequisite: The attacker must know between which parties, devices and functional components the SP Assertion is transferred.	
Attacker: External Attacker, IdP, SP	Victim: User
Impact: Medium	Precondition: Medium
Priority: Medium	Scope: In
Security Requirements: SR_17	

A7-T1 Modification of Security Properties	
Objective: An attacker modifies the security properties that used for the determination of the security level of the device. With the modification, various decisions regarding identity activation and assertion transfer can be influenced. Prerequisite: The attacker must get access to the security properties and the possibility to modify them.	
Attacker: External Attacker	Victim: User, IdP, SP
Impact: High	Precondition: Medium
Priority: Medium	Scope: In
Security Requirements: SR_18, SR_19	

A7-T2 Disclosure of Information about Security Properties	
Objective: An attacker discloses information about the security properties of devices. The obtained information might serve as input to foster other attacks. In addition it represents a privacy threat. Prerequisite: The attacker must get access to the security properties.	
Attacker: External Attacker	Victim: User
Impact: Medium	Precondition: Medium
Priority: Medium	Scope: In
Security Requirements: SR_18, SR_19, SR_20	

A7-T3 Inaccessibility of Security Properties	
Objective: An attacker makes the access to security properties of the devices of a virtual device impossible. In consequence all processes that work on security properties are unavailable. Prerequisite: The attacker must know how security properties are accessed on an individual device and how they are exchanged within a virtual device.	
Attacker: External Attacker	Victim: User
Impact: Low	Precondition: Medium
Priority: Low	Scope: In
Security Requirements: SR_4	

A8-T1 Unintended Device Discovery	
Objective: An attacker obtains knowledge about the devices in the surrounding and knowledge about the relationships of devices, i.e. their belonging to a virtual device. Prerequisite: The attacker is able to eavesdrop the communication between devices.	
Attacker: External Attacker	Victim: User
Impact: Low	Precondition: Medium
Priority: Low	Scope: In
Security Requirements: SR_21	

A8-T2 Forged Device Discovery	
Objective: An attacker forges information exchanged during the device discovery process in order provide false information about available devices. Prerequisite: The attacker is able to eavesdrop the communication channel and inject messages.	
Attacker: External Attacker	Victim: User
Impact: Medium	Precondition: Medium
Priority: Medium	Scope: In
Security Requirements: SR_22, SR_21	

A8-T3 Denial of Service of Device Discovery	
Objective: An attacker forges information with the intent make regular device discovery unavailable. Prerequisite: The attacker is able to inject falsified messages.	
Attacker: External Attacker	Victim: User
Impact: Medium	Precondition: Medium
Priority: High	Scope: In
Security Requirements: SR_22	

A9-T1 Obtainment of Device Identifier	
Objective: An attacker obtains information to identify devices and potentially derive information about the user. Prerequisite: An attacker is able to eavesdrop exchanges messages.	
Attacker: External Attacker, IdP, SP	Victim: User
Impact: Low	Precondition: Low
Priority: Low	Scope: In
Security Requirements: SR_21, SR_23	

A10-T1 ř	
Objective: An attacker access the credential store and obtains sensitive information. Prerequisite: An attacker has access to the interfaces exposed by the credential store. This can be achieved by having access to the locality where the credential store resides or by potentially using interfaces exposed to the outside world.	
Attacker: External Attacker	Victim: User, IdP, SP
Impact: High	Precondition: High
Priority: High	Scope: Partially
Security Requirements: SR_24	

A11-T1 Modification of Device Characteristics	
Objective: An attacker modifies the device characteristics to influence the behavior of algorithms regarding device and identity selection. Prerequisite: An attacker must have access to the interfaces to store device characteristics.	
Attacker: External Attacker	Victim: User
Impact: High	Precondition: Medium
Priority: High	Scope: In
Security Requirements: SR_25, SR_26	

A11-T2 Disclosure of Information about Device Characteristics	
Objective: An attacker discloses information about the characteristics of deices. Prerequisite: An attacker must be able to access the interfaces, used to store device characteristics	
Attacker: External Attacker	Victim: User
Impact: Low	Precondition: Medium
Priority: Medium	Scope: In
Security Requirements: SR_25	

A12-T1 Modification of Usage Context	
Objective: An attacker modifies the usage context in order to influence the behavior of algorithms regarding device and identity selection. Prerequisite: An attacker must be able to have access to the interfaces, used to modify the usage context.	
Attacker: External Attacker	Victim: User
Impact: High	Precondition: Medium
Priority: High	Scope: In
Security Requirements: SR_27, SR_28	

A12-T2 Disclosure of Information about Usage Context	
Objective: An attacker discloses the usage context text of devices and identities. This represents a potential privacy threat and might be input for potential other attacks. Prerequisite: An attacker must be able to have access to the interfaces used to retrieve the usage context.	
Attacker: External Attacker	Victim: User
Impact: Low	Precondition: Medium
Priority: Low	Scope: In
Security Requirements: SR_27	

A13-T1 Modification of Assertion Exchange Authorization Policies	
Objective: An attacker modifies the policies that determine which devices of the virtual device are allowed to retrieve SP Assertions. This potentially serves as an enabler for other attacks. Prerequisite: An attacker must be able to have access to the interfaces used to modify the policies.	
Attacker: External Attacker, User	Victim: User
Impact: High	Precondition: Medium
Priority: High	Scope: In
Security Requirements: SR_29, SR_30	

A13-T2 Disclosure of Assertion Exchange Authorization Policies	
Objective: An attacker discloses the the policies. This represents a privacy threat. Prerequisite: An attacker must be able to have access to the interfaces used to retrieve the policies.	
Attacker: External Attacker	Victim: User
Impact: Low	Precondition: Medium
Priority: Medium	Scope: In
Security Requirements: SR_29	

A14-T1 Modification of Identity Activation Authorization Policies	
Objective: An attacker modifies the policies that determine which devices of the virtual device are allowed to activate an identity. This potentially serves as an enabler for other attacks. Prerequisite: An attacker must be able to have access to the interfaces used to retrieve and modify the policies.	
Attacker: External Attacker, User	Victim: User
Impact: High	Precondition: Medium
Priority: High	Scope: In
Security Requirements: SR_31, SR_32	

A14-T2 Disclosure of Identity Activation Authorization Policies	
Objective: An attacker discloses the policies that determine which devices of the virtual device are allowed to activate an identity. This represents a privacy threat. Prerequisite: An attacker must be able to have access to the interfaces used to retrieve the policies.	
Attacker: External Attacker	Victim: User
Impact: Low	Precondition: Medium
Priority: Medium	Scope: In
Security Requirements: SR_31	

A.1.4 Security Requirements

SR-1	Mechanism for prevention of unauthorized device usage
A mechanism is required that prevents device usage by unauthorized users. Unauthorized users are potential attackers that got physical access to the device owned by another user.	
Stakeholder	User
Addressed Threats	A1-T1, A1-T2, A1-T4, A1-T5
Priority	High

SR-2	Mechanism for removing devices from Virtual Device
A mechanism is required that allows a user to remove a device from the virtual device composition. The removal of a device from the virtual device composition must not depend on the to be removed device. If a device is removed from the virtual device composition, all data regarding the virtual device, regarding other devices that are part of the virtual device composition and regarding identities of the user have to be removed.	
Stakeholder	User
Addressed Threats	A1-T1, A1-T2, A1-T4, A1-T5
Priority	High

SR-3	Logging of service consumption
A logging mechanism is required that records events regarding service sessions. Relevant events are the start of the service session, the end of the service session, which device has been used for authentication and which device has been used for the service session itself.	
Stakeholder	User
Addressed Threats	A1-T3
Priority	Medium

SR-4	No dependency on single device
The operation of the virtual device must not depend on a single device. Service session should not depend on a single device. However, it is acceptable that specific service session depend on the security capabilities of one device.	
Stakeholder	User
Addressed Threats	A1-T6, A1-T5, A7-T3
Priority	Medium

SR-5	Authorization mechanism for modifing a virtual device
An authorization mechanism is required to modify the virtual device. It must not be possible to add or delete a device from the virtual device composition without successive rights.	
Stakeholder	User
Addressed Threats	A2-T1
Priority	High

SR-6	Rate Limiting for Identity Activation and Assertion Exchange
The rate of identity activation requests and Assertion Exchange requests should be limited to avoid DoS attacks and prevent malicious behaviour. The rate must be adjustable by the owner of the virtual device.	
Stakeholder	
Addressed Threats	A1-T6, A1-T5
Priority	Medium

SR-7	Mechanism to obtain Information about Virtual Device Composition
A functionality is required to identify, which devices are part of a virtual device. This can be realized for example by visualization of a list containing all devices that are part of a virtual device.	
Stakeholder	
Addressed Threats	A2-T1
Priority	High

SR-8	Logging for Virtual Device modification
A logging mechanism is required that records changes in the virtual device composition. This includes permanent changes, i.e. adding or removing a virtual device, and temporary changes due to the unavailability of some devices	
Stakeholder	
Addressed Threats	A2-T2
Priority	Medium

SR-9	Confidentiality of Information about Virtual Device Composition
Information that describes the virtual device composition, e.g. devices that are part of the virtual device must be kept confidential	
Stakeholder	
Addressed Threats	A2-T3
Priority	High

SR-10	Secure Transmission of IdP Token
The IdP Token must be securely transferred between the IdP and user. That means the communication channel must be mutually authenticated and provide protection of integrity and confidentiality.	
Stakeholder	
Addressed Threats	A3-T1, A3-T2
Priority	High

SR-11	Secure Storage of IdP Token
The IdP Token must be securely stored on the user device. Access to the token must be controlled.	
Stakeholder	
Addressed Threats	A3-T1, A3-T2
Priority	High

SR-12	Secure Transmission of SP Token
The SP Token must be securely transferred between the SP and user. That means the communication channel must be mutually authenticated and provide protection of integrity and confidentiality.	
Stakeholder	
Addressed Threats	A4-T1, A4-T2
Priority	Medium

SR-13	Secure Storage of SP Token
The SP Token must be securely stored on the user device. Access to the token must be controlled.	
Stakeholder	
Addressed Threats	A4-T1, A4-T2
Priority	Medium

SR-14	Secure Storage of Authentication Credentials
If authentication credentials are stored on of the devices that are part of the virtual device, the must be stored securely. Secure storage prevents unauthorized retrieval by unauthorized users	
Stakeholder	
Addressed Threats	A5-T1
Priority	High

SR-15	Secure Transmission of Authentication Credentials
If authentication credentials have to be transmitted, e.g. because it is required by the authentication mechanism, the transmission has to be secure. That means mutually authenticated, confidental and integrity protected.	
Stakeholder	
Addressed Threats	A5-T1
Priority	High

SR-16	Logging Mechanism for Authentication Credential Usage
A mechanism is required to logs the usage of authentication credentials in order to avoid repudiation of authentication credentials. This mechanism should be implemented within the virtual device and potentially with the IdP.	
Stakeholder	
Addressed Threats	A5-T2
Priority	High

SR-17	Secure Transmission of SP Assertions
The SP Assertion must be securely transferred between the IdP and the user as well as between the user and the SP. That means the communciation channel must be mutually authenticated and must provide protection of integrity and confidentiality.	
Stakeholder	
Addressed Threats	A6-T1, A6-T2
Priority	Medium

SR-18	Secure Storage of Security Properties
A secure storage is required to store the security properties. Access to the secure storage must be controlled.	
Stakeholder	
Addressed Threats	A7-T1, A7-T2
Priority	Medium

SR-19	Logging Mechanism for Changed Security Properties
Changes of the security properties must be recorded with the reason that caused the change (e.g. software upgrade, hardware modification)	
Stakeholder	
Addressed Threats	A7-T1, A7-T2
Priority	Medium

SR-20	Storage of all Security Properties only on Secure Devices
The security level of a device must be considered when sensitive information (e.g. about other devices) is stored. Sensitive information should only be stored, if device is above a to be defined threshold.	
Stakeholder	
Addressed Threats	A7-T2
Priority	Medium

SR-21	Secure Device Discovery
Discovery of devices belonging to a Virtual Device must be secure. That means the authenticity of the discovered device must be guaranteed and that the discovery process does not reveal information about the participating devices or about the owner of th device.	
Stakeholder	
Addressed Threats	A8-T1, A8-T2, A9-T1
Priority	Medium

SR-22	Availability of Device Discovery
It should not be possible disable the correct discovery of devices belonging to a virtual device.	
Stakeholder	
Addressed Threats	A8-T2, A8-T3
Priority	Medium

SR-23	Encrypt device identifiers or avoid unique device identifiers
Unique device identifiers should be avoided in communication processes	
Stakeholder	
Addressed Threats	A9-T1
Priority	Low

SR-24	Access control on credential store
The credential store must have sufficient access control mechanisms in place.	
Stakeholder	
Addressed Threats	A10-T1
Priority	High

SR-25	Secure Storage of Device Characteristics
A secure storage is required to store the device characteristics. Access to the secure storage must be controlled.	
Stakeholder	
Addressed Threats	A11-T1, A11-T2
Priority	High

SR-26	Logging Mechanism for Changed Device Characteristics
Changes of the device characteristics must be recorded with the reason that caused the change.	
Stakeholder	
Addressed Threats	A11-T1
Priority	High

SR-27	Secure Storage of Usage Context
A secure storage is required to store the usage context. Access to the secure storage must be controlled	
Stakeholder	
Addressed Threats	A12-T1, A12-T2
Priority	High

SR-28	Logging Mechanism for Usage Context Change
Changes of the device usage context must be recorded with the reason that caused the change.	
Stakeholder	
Addressed Threats	A12-T1
Priority	High

SR-29	Secure Storage of Assertion Exchange Authorization Policies
A secure storage is required to store the Assertion Exchange Authorization Policies. Access to the secure storage must be controlled	
Stakeholder	
Addressed Threats	A13-T1, A13-T2
Priority	High

SR-30	Logging Mechanism for Assertion Exchange Authorization Policy Change
Changes of the Assertion Exchange Authorization Policies must be recorded with the reason that caused the change.	
Stakeholder	
Addressed Threats	A13-T1
Priority	High

SR-31	Secure Storage of Identity Activation Authorization Policies
A secure storage is required to store the Identity Activation Authorization Policies. Access to the secure storage must be controlled	
Stakeholder	
Addressed Threats	A14-T1, A14-T2
Priority	High

SR-32	Logging Mechanism for Identity Activation Authorization Policy Change
Changes of the Identity Activation Authorization Policies must be recorded with the reason that caused the change.	
Stakeholder	
Addressed Threats	A14-T1
Priority	Medium

SR-100	Mechanism to stop all IdP Sessions
Several IdP Sessions might exist on different devices. A mechanism is needed to stop all IdP sessions immediately.	
Stakeholder	
Addressed Threats	
Priority	

SR-101	Mechanism to stop IdP Sessions on one device
Several IdP Sessions might exist on one device. A mechanism is required to stop these IdP Sessions immediately.	
Stakeholder	
Addressed Threats	
Priority	

SR-102	Mechanism to stop all SP Sessions
Several SP Sessions might exist on different devices. A mechanism is required to stop all SP sessions immediately	
Stakeholder	
Addressed Threats	
Priority	

SR-103	Mechanism to stop all SP Sessions on one device
Several SP Sessions might exist on one device. A mechanism is required to remotely stop these SP sessions immediately.	
Stakeholder	
Addressed Threats	
Priority	

A.2 Details of Virtual Device Management

Section A.2.1 contains additional details on the security architecture.

A.2.1 Security Architecture

This section details the establishment of secure channels and the extension of the virtual device.

A.2.1.1 Establishment of Secure Channel

Figure A.1 details the establishment of a secure channel. Figure A.2 shows an activity diagram, which elaborates the retrieval of the membership list.

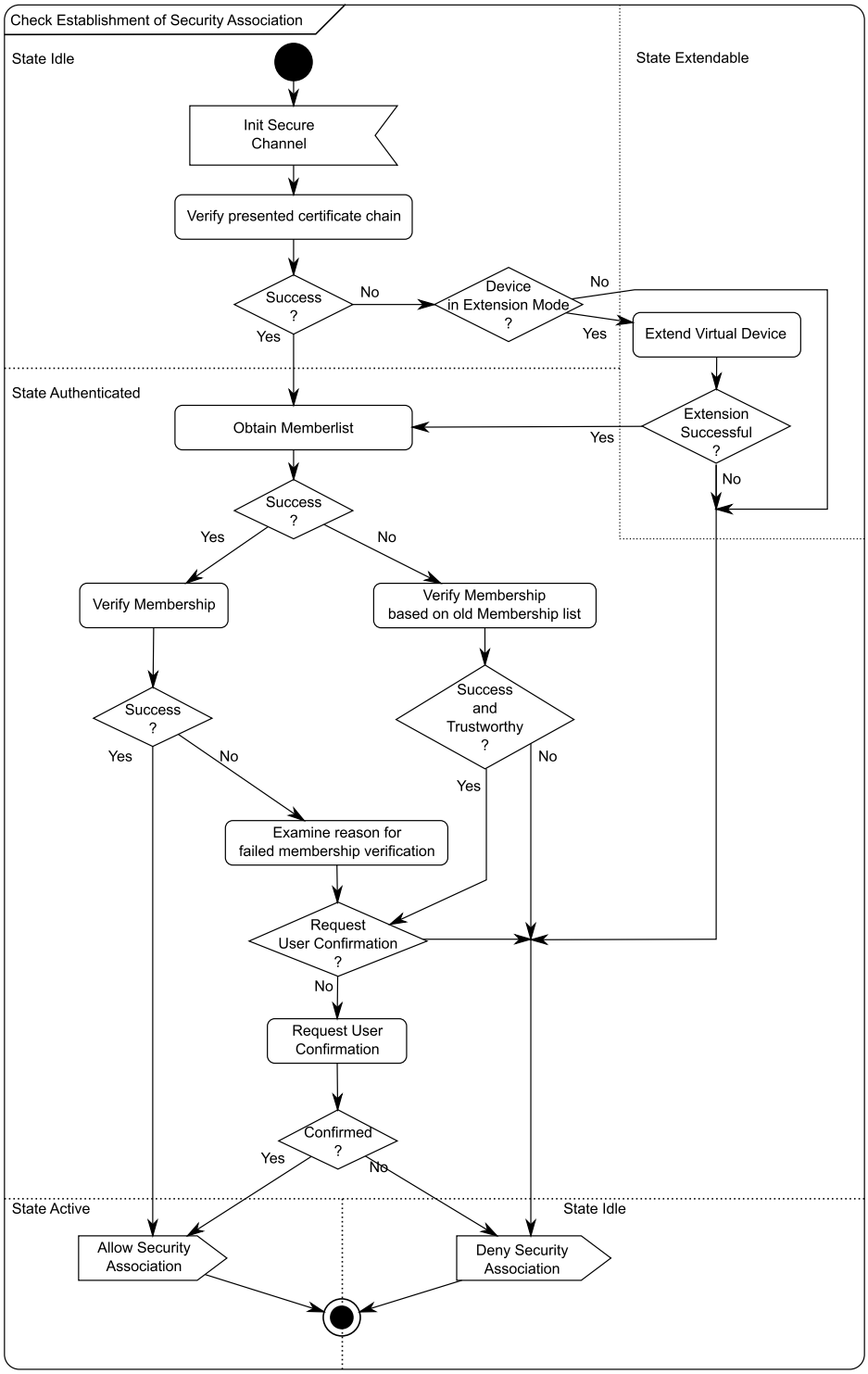


Figure A.1: Activity Diagram: Establishment of a Secure Channel.

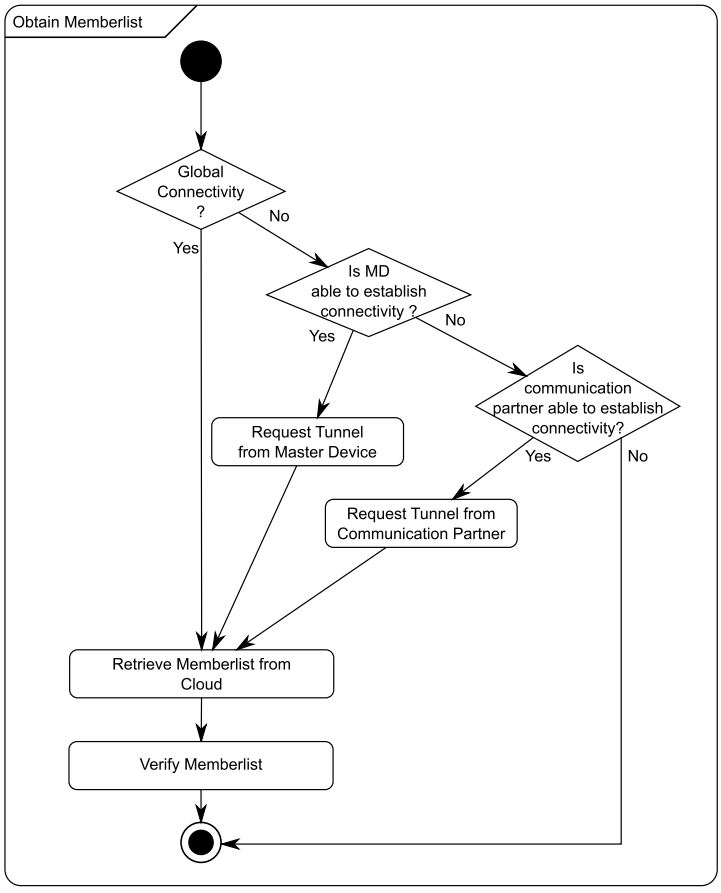


Figure A.2: Activity Diagram: Obtainment of Membership List

A.2.1.2 Extension of Virtual Device

Figure A.3 shows the necessary activities to extend a virtual device.

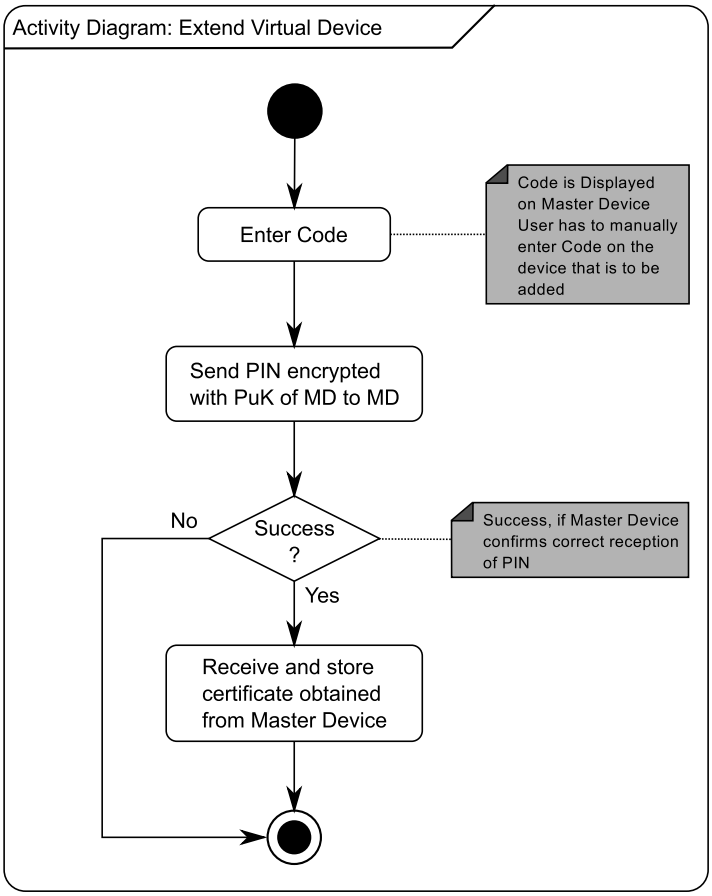


Figure A.3: Activity Diagram: Extending a Virtual Device

A.3 Addressed Requirements

A.3.1 Functional Requirements

Section 6.1.2.3 provides the detailed discussion of functional requirements that have not been sufficiently addressed. Table A.1 indicates how each functional requirement has been addressed.

Table A.1: Evaluation of Functional Requirements

		Functional Solution Level		Concrete Solution Level	
No	Short Name	Cov.	Reference	Cov.	Reference
DM-DD-1	Device Discovery in General (c.f. R4)	Yes	Table 4.21	No	
DM-DD-2	Proximity Detection	Yes	Table 4.21	No	

Continued on next page

Table A.1: Evaluation of Functional Requirements

		Functional Solution Level		Concrete Solution Level	
No	Short Name	Cov.	Reference	Cov.	Reference
DM-DD-3	Device Discovery should not reveal any information	Yes	Table 4.21	No	
DM-SA-1	Mutual Authentication of Devices	Yes	Table 4.21	Yes	Table 5.3
DM-SA-2	Secure Storage of Credentials	Yes	Table 4.19	No	
DM-VDM-1	Adding a Device to VD	Yes	Table 4.21	Yes	Table 5.3
DM-VDM-2	Removing a Device from VD	Yes	Table 4.21	Yes	Table 5.3
DM-VDM-3	Device Identity	Yes	Table 4.21	No	
DM-DC-1	Capturing of Device Characteristics	Yes	Table 4.21	No	
DM-DC-2	Manual Editing of Device Characteristics	Yes	Table 4.21	No	
DM-DC-3	Selection of Usage Context	Yes	Table 4.21	No	
DM-DC-4	Data Format for Device Characteristics	Yes	Table 4.21	No	
DM-DA-1	Mechanism for Data Exchange	Yes	Table 4.21	No	
IdM-AE-1	Protocol for the request of authentication assertions	Yes	Table 4.17	Yes	Table 5.2
IdM-AE-2	Authorization based on Device ID	Yes	Table 4.17	Yes	Table 5.2
IdM-AE-3	Information about Usage Purpose	Yes	Table 4.17	Yes	Table 5.2
IdM-AE-4	Manual Confirmation	Yes	Table 4.17	Yes	Table 5.2
IdM-AE-5	Feedback if Manual Confirmation is required	Yes	Table 4.17	No	
IdM-IA-1	Protocol for identity activation (c.f. R3)	Yes	Table 4.17	Yes	Table 5.2
IdM-IA-2	Authorization based on Device ID	Yes	Table 4.17	Yes	Table 5.2
IdM-IA-3	Feedback on Device triggered for Identity Activation	No		No	
IdM-IM-1	Capture existing identities	Yes	Table 4.15	No	
IdM-IM-2	Store metadata on identities	Yes	Table 4.15	No	
IdM-IM-3	Automatic capturing of existing identities	Partially	Table 4.15	No	

Continued on next page

Table A.1: Evaluation of Functional Requirements

		Functional Solution Level		Concrete Solution Level	
No	Short Name	Cov.	Reference	Cov.	Reference
IdM-IM-4	Manual adding and removal of user identities	Yes	Table 4.15	No	
IdM-IM-5	Manual modification of identity information	Yes	Table 4.15,	No	
IdM-IM-6	Graphical User Interface of Identity Selection	Yes	Table 4.15,	Yes	Table 5.1
IdM-IM-7	List of selectable identities	Yes	Table 4.15,	Yes	Table 5.1
IdM-IM-8	Priorities of selectable identities	Yes	Table 4.15,	Yes	Table 5.1

A.3.2 Security Requirements

Section 6.1.2.5 provides the detailed discussion of the security requirements that have not been sufficiently addressed. Table A.2 indicates how each security requirement has been addressed.

Table A.2: Evaluation of Security Requirements

		Functional Solution Level		Concrete Solution Level	
No	Short Name	Cov.	Reference	Cov.	Reference
SR-1	Mechanism for prevention of unauthorized device usage	Partially	Table 4.21	No	
SR-2	Mechanism for removing devices from Virtual Device	Yes	Table 4.21	Yes	Table 5.3
SR-3	Logging of service consumption	Yes	Table 4.15, Table 4.17	No	
SR-4	No dependency on single device	Yes	Table 4.17, Table 4.21	Yes	Table 5.1, Table 5.3
SR-5	Authorization mechanism for modifing a virtual device	Yes	Table 4.21	Yes	Table 5.3
SR-6	Rate Limiting for Identity Activation and Assertion Exchange	Yes	Table 4.17	No	
SR-7	Mechanism to obtain Information about Virtual Device Composition	Yes	Table 4.21	Yes	Table 5.3
SR-8	Logging for Virtual Device modification	Yes	Table 4.19, Table 4.21	No	

Continued on next page

Table A.2: Evaluation of Security Requirements

		Functional Solution Level		Concrete Solution Level	
No	Short Name	Cov.	Reference	Cov.	Reference
SR-9	Confidentiality of Information about Virtual Device Composition	Yes	Table 4.19, Table 4.21	Partially	Table 5.3
SR-10	Secure Transmission of IdP Token	No		No	
SR-11	Secure Storage of IdP Token	Yes	Table 4.15, Table 4.19	No	
SR-12	Secure Transmission of SP Token	No		No	
SR-13	Secure Storage of SP Token	Yes	Table 4.19	No	
SR-14	Secure Storage of Authentication Credentials	Yes	Table 4.15, Table 4.19	No	
SR-15	Secure Transmission of Authentication Credentials	No		No	
SR-16	Logging Mechanism for Authentication Credential Usage	Yes	Table 4.15, Table 4.19	No	
SR-17	Secure Transmission of SP Assertions	Yes	Table 4.15, Table 4.17	Yes	Table 5.3
SR-18	Secure Storage of Security Properties	Yes	Table 4.19, Table 4.21	No	
SR-19	Logging Mechanism for Changed Security Properties	No	Table 4.19, Table 4.21	No	
SR-20	Storage of all Security Properties only on Secure Devices	Yes	Table 4.19, Table 4.21	Yes	Table 5.3
SR-21	Secure Device Discovery	No	Table 4.21	No	
SR-22	Availability of Device Discovery	No	Table 4.21	No	
SR-23	Encrypt device identifiers or avoid unique device identifiers	Yes	Table 4.21	No	
SR-24	Access control on credential store	Yes	Table 4.19	No	
SR-25	Secure Storage of Device Characteristics	Yes	Table 4.19, Table 4.21	No	
SR-26	Logging Mechanism for Changed Device Characteristics	Yes	Table 4.19	No	
SR-27	Secure Storage of Usage Context	Yes	Table 4.19, Table 4.21	No	
SR-28	Logging Mechanism for Usage Context Change	Yes	Table 4.19, Table 4.21	No	

Continued on next page

Table A.2: Evaluation of Security Requirements

		Functional Solution Level		Concrete Solution Level	
No	Short Name	Cov.	Reference	Cov.	Reference
SR-29	Secure Storage of Assertion Exchange Authorization Policies	Yes	Table 4.17, Table 4.19	No	
SR-30	Logging Mechanism for Assertion Exchange Authorization Policy Change	Yes	Table 4.17, Table 4.19	No	
SR-31	Secure Storage of Identity Activation Authorization Policies	Yes	Table 4.17, Table 4.19	No	
SR-32	Logging Mechanism for Identity Activation Authorization Policy Change	Yes	Table 4.17, Table 4.19	No	
SR-100	Mechanism to stop all IdP Sessions	Yes		Yes	Table 5.2
SR-101	Mechanism to stop IdP Sessions on one device	Yes		Yes	Table 5.2
SR-102	Mechanism to stop all SP Sessions	Yes		Yes	Table 5.2
SR-103	Mechanism to stop all SP Sessions on one device	Yes		Yes	Table 5.2

A.4 Security Evaluation

A.4.1 Detailed Threat Description

Table A.3 details the threats that have been identified during the security evaluation.

Table A.3: Overview on newly Identified Threats

Identifier	ShortName	Covered	Covered By
A20_T1	Tampering of Logging Information	Yes	*SSE-SSS* (→ Section 4.5.4)
A20_T2	Repudiation of Logging Information	Yes	*SSE-SSS* (→ Section 4.5.4)
A20_T3	Disclosure of Logging Information	Yes	*SSE-SSS* (→ Section 4.5.4)
A21_T2	Disclosure of Identity Information	Yes	*SSE-SSS* (→ Section 4.5.4), Access Control Mechanism for IIEP (→ Figure 5.9)
A22_T2	Disclosure of Identity Device Matrix	Yes	Access Control Mechanism for IIEP (→ Figure 5.9)

Continued on next page

Table A.3: Overview on Newly Identified Threats

Identifier	ShortName	Covered	Covered By
A23_T1	Tampering of Device Information	Partially	*SSE-SSS* (→ Section 4.5.4), *SSE-LS* (→ Section 4.5.4)
A23_T2	Disclosure of Device Information	Yes	*SSE-SSS* (→ Section 4.5.4), *SSE-LS* (→ Section 4.5.4)
A24_T2	Disclosure of Virtual Device Membership List	Yes	*SSE-SSS* (→ Section 4.5.4)
A25_T1	Tampering of Virtual Device Key Pair	Partially	*SSE-SSS* (→ Section 4.5.4 and Section 5.4.2)
A25_T2	Disclosure of Virtual Device Key Pair	Partially	*SSE-SSS* (→ Section 4.5.4 and Section 5.4.2)
A25_T3	Denial of Service by modification of Virtual Device Key Pair	Partially	*SSE-SSS* (→ Section 4.5.4 and Section 5.4.2)
A26_T1	Tampering of Master Device Key Pair	Yes	Key Hierarchy (→ Section 5.4.2), Logging Mechanism (→ Section 4.5.4)
A26_T2	Disclosure of Master Device Key Pair	Yes	*SSE-SSS* (→ Section 4.5.4), Logging Mechanism (→ Section 4.5.4)
A26_T3	Denial of Service by modification of Master Device Key Pair	Yes	*SSE-SSS* (→ Section 4.5.4), Logging Mechanism (→ Section 4.5.4)
A27_T1	Tampering of Non Master Device Key Pair	Yes	Secure Storage (→ Section 4.5.4), Logging Mechanism (→ Section 4.5.4)
A27_T2	Disclosure of Non Master Device Key Pair	Yes	*SSE-SSS* (→ Section 4.5.4), Logging Mechanism (→ Section 4.5.4)
A27_T3	Denial of Service of Non Master Device Key Pair	Yes	*SSE-SSS* (→ Section 4.5.4), Logging Mechanism (→ Section 4.5.4)
A28_T1	Tampering of Secure Storage	Partially	*SSE-SSS* (→ Section 4.5.4)
A28_T2	Disclosure of data stored in Secure Storage	Partially	*SSE-SSS* (→ Section 4.5.4)
A28_T3	Denial of Service of Secure Storage	Partially	It is not always required to update the membership list, → Figure A.1
A29_T1	Tampering of Filter Rules	Yes	*SSE-SSS* (→ Section 4.5.4), *SSE-LS* (→ Section 4.5.4)
A29_T2	Disclosure of Identity Filtering Rules	Yes	*SSE-SSS* (→ Section 4.5.4), *SSE-LS* (→ Section 4.5.4)
A30_T1	Tampering of Ranking Rules	Yes	*SSE-SSS* (→ Section 4.5.4), *SSE-LS* (→ Section 4.5.4)

Continued on next page

Table A.3: Overview on Newly Identified Threats

Identifier	ShortName	Covered	Covered By
A30_T2	Disclosure of Ranking Rules	Yes	*SSE-SSS* (→ Section 4.5.4), Logging Mechanism (→ Section 4.5.4)
A31_T1	Tampering of Identity Recommendations	Yes	Secure Channel (→ Section 5.4.2)
A31_T2	Disclosure of Identity Recommendations	Yes	Secure Channel (→ Section 5.4.2)

A.5 Performance Evaluation

Section 6.3 established an analytical model to evaluate the performance of the overall system. The following subsections provide additional details how the equations have been derived.

A.5.1 State Probability

The modell is based on the assumption that the user randomly selects one of his identities. That means the usage of one identity is independent from the usage of another identity. Thus the number of SP sessions x_i for identity i is independent of the number of SP sessions x_{i+1} for identity $i+1$ and can be modelled independent of each other. In consequence the conditional probability is equal to the product of the individual propabilities, i.e. Eq. (A.1) holds.

$$p(x_1, x_2, \ldots, x_N) = \prod_{i=1}^{N} p(x_i) \tag{A.1}$$

In case of Figure 6.22, the two dimensons can be considered independent of each other. Figure A.4 shows the decoupled processes. Therefore, the propability p_{x_1} to be in a state in which x_1 SP session are active for identity 1 can be easily calculated. It is

$$p(x_1) = p(0)\frac{A'^{x_1}}{x_1!} \tag{A.2}$$

$$p(0) = \frac{1}{\sum_{i=0}^{\infty} \frac{A'^i}{i!}} = \frac{1}{e^{A'}} \tag{A.3}$$

In the general case A' is equal to

$$A' = \frac{\lambda}{N_{Id}\mu} = \frac{A}{N_{Id}} \tag{A.4}$$

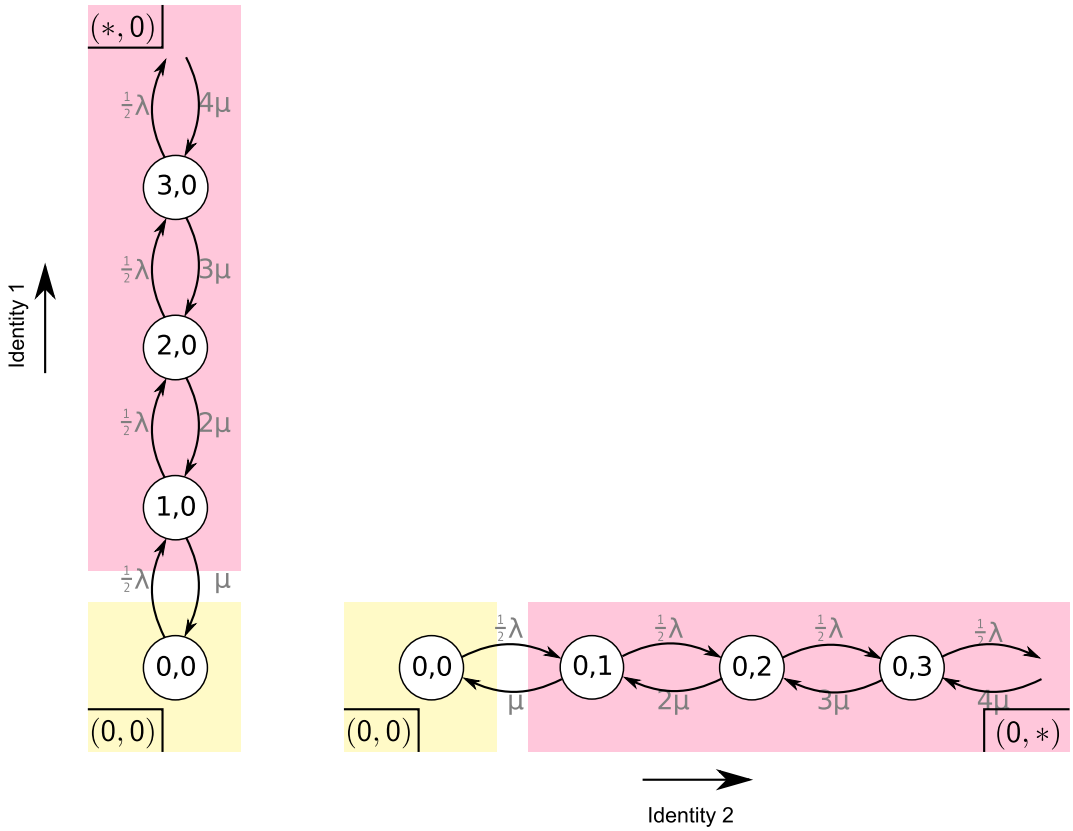


Figure A.4: Decoupled Birth and Death Processes

The combination of Eq. (A.2) and Eq. (A.1) results in

$$p(x_1, x_2, \ldots, x_{N_{Id}}) = p(x_1) \cdot p(x_2) \cdot \ldots \cdot p(x_{N_{Id}}) \tag{A.5}$$

$$\tag{A.6}$$

$$= p(0)^{N_{Id}} \frac{A'^{x_1}}{x_1!} \cdot \frac{A'^{x_2}}{x_2!} \cdot \ldots \cdot \frac{A'^{x_{N_{Id}}}}{x_{N_{Id}}!} \tag{A.7}$$

$$= \left(\frac{1}{e^{\frac{A}{N_{Id}}}}\right)^{N_{Id}} \frac{\left(\frac{A}{N_{Id}}\right)^{x_1+x_2+\ldots+x_{N_{Id}}}}{x_1! \cdot x_2! \cdot \ldots \cdot x_{N_{Id}}!} \tag{A.8}$$

$$= \frac{1}{e^A} \frac{\left(\frac{A}{N_{Id}}\right)^{x_1+x_2+\ldots+x_{N_{Id}}}}{x_1! \cdot x_2! \cdot \ldots \cdot x_{N_{Id}}!} \tag{A.9}$$

The same principle applies for the case of multiple devices. If one of the devices is considered a modified value for A' is required:

$$A' = \frac{A}{N_{Dev} N_{Id}} \tag{A.10}$$

This leads to

$$p(0) = \frac{1}{e^{\frac{A}{N_{Dev} N_{Id}}}} \tag{A.11}$$

$$p(x_1, x_2, \ldots, x_{N_{Id}}) = \frac{1}{e^{\frac{A}{N_{dev}}}} \frac{\left(\frac{A}{N_{Dev}N_{Id}}\right)^{x_1+x_2+\ldots+x_{N_{Id}}}}{x_1! \cdot x_2! \cdot \ldots \cdot x_{N_{Id}}!} \tag{A.12}$$

For the macro states:

$$p_{1,Id} = \sum_{i=1}^{\infty} \frac{1}{e^{\frac{A}{N_{Dev}}}} \frac{\left(\frac{A}{N_{Dev}N_{Id}}\right)^{i}}{i!} \tag{A.13}$$

with $\sum_{n=0}^{\infty} \frac{x^n}{n!} = e^x$

$$p_{1,Id} = \frac{1}{e^{\frac{A}{N_{Dev}}}} \left[\left(\sum_{i=0}^{\infty} \frac{\left(\frac{A}{N_{Dev}N_{Id}}\right)^{i}}{i!} \right) - 1 \right] \tag{A.14}$$

$$= \frac{1}{e^{\frac{A}{N_{Dev}}}} \left(e^{\frac{A}{N_{Dev}N_{Id}}} - 1 \right) \tag{A.15}$$

Generalized:

$$p_{k,Id} = p(\underbrace{*, \ldots, *}_{k}, 0, \ldots, 0) = \frac{1}{e^{\frac{A}{N_{Dev}}}} \left(e^{\frac{A}{N_{Dev}N_{Id}}} - 1 \right)^k \tag{A.16}$$

Considering all combination possibilites:

$$P(N_{act} = k) = \binom{N_{Id}}{k} p_{k,Id} \tag{A.17}$$

$$= \binom{N_{Id}}{k} \frac{1}{e^{\frac{A}{N_{Dev}}}} \left(e^{\frac{A}{N_{Dev}N_{Id}}} - 1 \right)^k \tag{A.18}$$

A.5.2 Mean Number of Active Identities

The derivation of Eq. (6.17) is detailed here.

$$E[N_{act,C2}] = \sum_{k=0}^{N_{id}} k \cdot P(N_{act} = k)$$

$$= \sum_{k=0}^{N_{id}} k \binom{N_{Id}}{k} \frac{1}{e^A} (e^{\frac{A}{N_{Id}}} - 1)^k \tag{A.19}$$

$$\tag{A.20}$$

with $\binom{N_{id}}{k} = \frac{N_{Id}}{k} \binom{N_{Id}-1}{k-1}$

$$E[N_{act,C2}] = \sum_{k=1}^{N_{Id}} N_{Id} \binom{N_{Id}-1}{k-1} \frac{1}{e^A} (e^{\frac{A}{N_{Id}}} - 1)^{k-1} (e^{\frac{A}{N_{Id}}} - 1) \tag{A.21}$$

with $k' = k - 1, k = k' + 1$

$$E[N_{act,C2}] = \sum_{k'=0}^{N_{Id}-1} N_{Id} \binom{N_{Id}-1}{k'} \frac{1}{e^A} (e^{\frac{A}{N_{Id}}} - 1)^{k'} (e^{\frac{A}{N_{Id}}} - 1) \quad \text{(A.22)}$$

$$= N_{Id} \frac{1}{e^A} (e^{\frac{A}{N_{Id}}} - 1) \sum_{k'=0}^{N_{Id}-1} \binom{N_{Id}-1}{k'} (e^{\frac{A}{N_{Id}}} - 1)^{k'} \quad \text{(A.23)}$$

with $\sum_{k=0}^{n} \binom{n}{k} x^k = (1+x)^n$

$$E[N_{act,C2}] = N_{Id} \frac{1}{e^A} (e^{\frac{A}{N_{Id}}} - 1)(1 + e^{\frac{A}{N_{Id}}} - 1)^{N_{Id}-1} \quad \text{(A.24)}$$

$$= N_{Id} (e^{\frac{A}{N_{Id}}} - 1) e^{\frac{A}{N_{Id}(N_{Id}-1)-A}} \quad \text{(A.25)}$$

$$= N_{Id} (e^{\frac{A}{N_{Id}}} - 1) e^{-\frac{A}{N_{Id}}} \quad \text{(A.26)}$$

$$= N_{Id} (1 - e^{-\frac{A}{N_{Id}}}) \quad \text{(A.27)}$$

B UML Modeling Methodology

For the modeling and for the description of the architecture, a semi-formal notation based on UML diagrams has been used. UML is considered as appropriate specification language for the description of software and system architectures [BK03, CBB+10]. This chapter gives an overview on UML and introduces afterwards the used diagram types in more detail. It highlights slight modifications compared to the UML standard [UML10]. The modifications have been made for simplification. Finally, Section B.5 introduces an approach for the modeling of communication protocols.

B.1 Overview

UML is a major standard in software engineering. It originated at Rational Software Corporation in the mid 90ies by the contributions of James Rumbaugh, Grady Booch and Ivar Jacobsen. All three are responsible for major contributions to the object-oriented analysis and design of software systems. Version 2 of UML adopted concepts from the Specification and Description Language (SDL).

UML distinguishes two different categories of diagrams to describe software architectures: Structure diagrams and behavior diagrams. (1) Structure diagrams describe the static view of all entities and their relation to each other. Table B.1 gives an overview on the different diagram types to describe the system structure. (2) Behavior diagrams model the work flows within entities and their interaction. Table B.2 gives an overview on the different diagram types to describe the behavior of a system.

All diagram types provide the possibility to add comments. An example for a comment is shown in Figure B.1. Comments are used to add additional information to the diagram. For example, comments allow the adding of semantic meaning. A comment can be associated with any element of a diagram.

Closely related to UML is the System Modeling Language (SysML) [sys10, DHJ+10]. SysML shares a common set of concepts with UML. Regarding these concepts SysML depends on

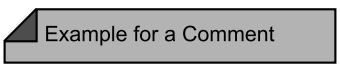


Figure B.1: Example for UML Comment

Diagram Type	Description
Class Diagram	A class diagram shows the fine-granular structure of system (→ Section B.2).
Component Diagram	A component diagram shows the instantiation of a system part from more coarse-granular point of view (→ Section B.3)
Composite Structure Diagram	A composite structure diagram details a class and shows its internal make-up.
Deployment Diagram	A deployment diagram shows the hardware that is used to deploy the system components.
Object Diagram	An object diagram shows the instantiation of a part of the system at a certain point of time.
Package Diagram	A package diagram shows the bundling of classes to packages.
Profile Diagram	A profile diagram is required to describe the meta-model of an UML diagram.

Table B.1: Overview on UML Structure Diagrams

Table B.2: Overview on UML Behavior Diagrams

Diagram Type	Description
Activity Diagram	An activity diagram shows the steps of a work flow that is executed within a dedicated system part (→ Section B.4).
Message Sequence Diagram	A message sequence chart shows the interaction between different parts of a structure diagram, e.g. between different objects.
State Machine Diagram	A state machine diagram models the states and the transitions between states of a dedicated system part.
Interaction Overview Diagram	An interaction overview diagram shows the work flows of a system that are conducted between different parties.
Use Case Diagram	A use case diagram models the goals and interactions of users with the system.
Communication Diagram	A communication diagram is a combination of a class diagram and an activity diagram.
Timing Diagram	A timing diagram is a special message sequence diagram that points out timing constraints.

UML. In addition, SysML extends UML regarding the collection and specification of requirements and test cases, which are necessary for the design of complex systems. SysML is for example used for the specification of large systems like cars or airplanes. For this thesis UML is sufficient. For more information on SysML it is referred to [sys10].

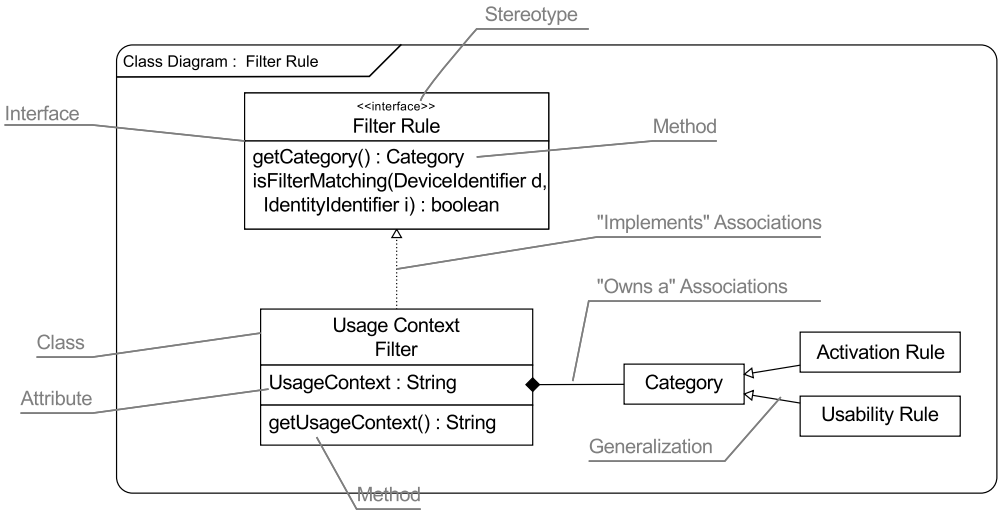


Figure B.2: Example for a Class Diagram

B.2 Class Diagrams

B.2.1 Usage

A class diagram shows a fine-granular model of a system part. It uses classes, interfaces and associations to describe the system part. A class represents a blueprint of an actual instantiation of a dedicated part that provides self-contained functionality. It describes the provided attributes and methods. An interface consists of a set of methods to describe functionality that is required or provided by a class or component. It serves as a kind of contract between a client, which requires the functionality, and a server, which provides the functionality. Figure B.2 shows an example for a class diagram. The class diagram specifies an interface *Filter Rule* that specifies two methods. The interface is implemented by the class *UsageContext Filter*, which has one attribute of type *String* called *UsageContext*. In addition it provides a method *GetUsageContext*. The class *UsageContext Filter* aggregates an object of class *Category*. Two specializations of the class *Category* exist: *Activation Rule* and *Usability Rule*.

B.2.2 Differences to UML

Within this thesis, class diagrams are used to specify data models and to illustrate the composition of components. The main focus resides on the understandability of the diagrams. Therefore, the complexity is reduced with respect to the number and types of the attributes and with respect to the provided methods. There is no claim for completeness regarding the class attributes and methods.

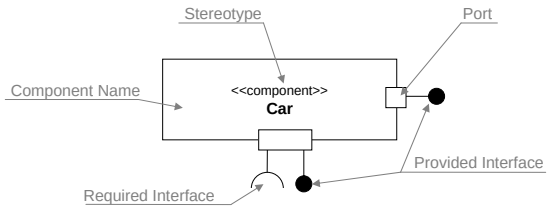


Figure B.3: Example for a Simple Component Diagram

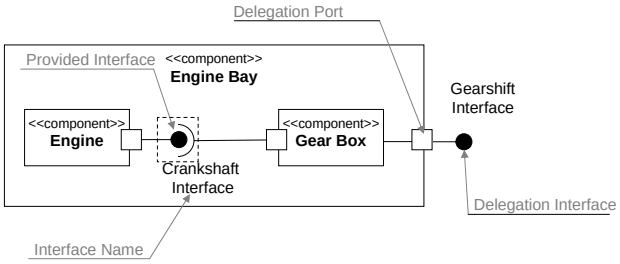


Figure B.4: Example for a Refined Component Diagram

B.3 Component Diagrams

B.3.1 Usage

Component diagrams are used to model complex software architectures. A component is a type that consists of other components or of a set of classes. With component diagrams a hierarchical refinement of the system can be achieved.

A component has a component name for identification and is qualified as component by the stereotype "«component»" as shown in Figure B.3. It provides interfaces to other components (indicated by the "ball" symbol) and requires interfaces provided by other components (indicated by the "socket" symbol). The port construct allows the grouping of several interfaces according to various criteria (e.g. stakeholder, categories) [BK03].

Figure B.4 shows that components are composed of internal components. The internal components are as well connected by interfaces. Each interface is identified by an interface name. External interfaces, which are realized by the internal components, are indicated by so called delegation ports. A delegation port connects the port of the internal component to the outside. The port concept allows to group interfaces that are exposed by internal components. The *Engine Interface* in Figure B.5 exposes functionality by using four internal interfaces.

B.3.2 Differences to UML

This thesis uses component diagrams slightly different. Originally, component diagrams are used to illustrate the runtime state of components [CBB+10]. That means a component of a specific type consists of instances of other components or classes, i.e. objects. With us, component diagrams consist of other types of components. This thesis neglects the UML highlighting of UML instances (ComponentName : ComponentType), because it is assumed that there is exactly one instantiation of each subcomponent.

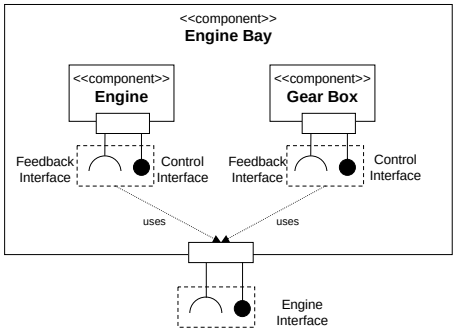


Figure B.5: Example for Composition Diagram with Complex Interfaces

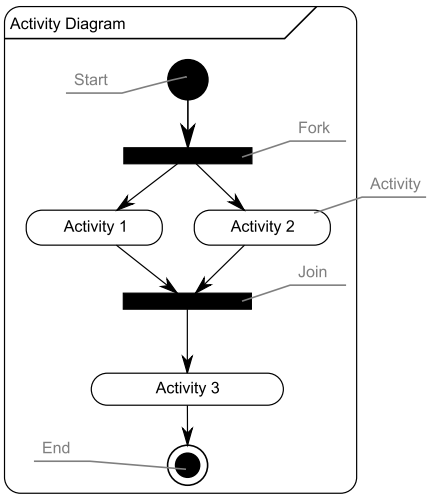


Figure B.6: Example for an Activity Diagram

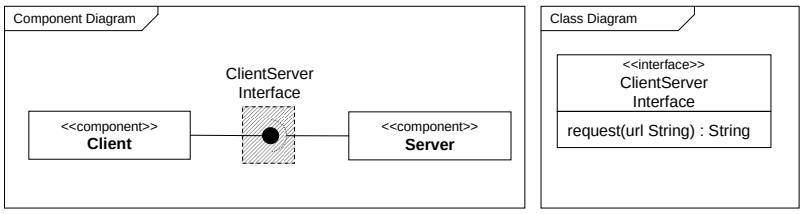
Moreover, dotted rectangles, around sets of interfaces, indicate interfaces that belong together. Interfaces belong together in two cases: First, a matching combination of provided and required interfaces is provided. Second, a component is providing and requiring the same interface.

B.4 Activity Diagrams

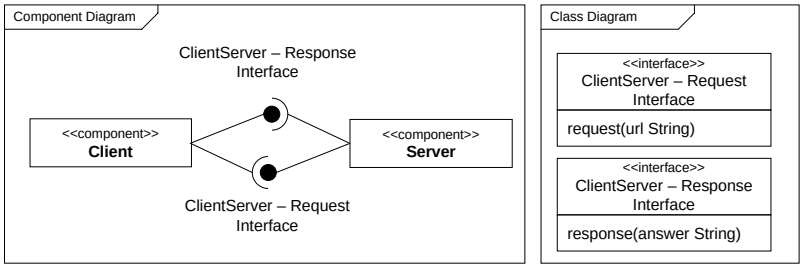
Figure B.6 shows an example for an activity diagram. An activity diagram starts with the start node and terminates with the end node. All activities on the path from the start node to the end node have to be carried out and represent the control flow. With activity diagrams, parallelism can be modeled. The fork node splits the control flow and executes two different activities: *Activity 1* and *Activity 2*. The join node merges the two control flows upon completion of the activities. An activity can be refined and may consist of several subactivities.

B.5 Modeling of Communication Protocols

This thesis modeled the structural view of communication protocols by means of component diagrams with corresponding interface definitions. The behavior is specified by means of message



(a) Abstract View



(b) Refined View

Figure B.7: Component and Class Diagram for Protocol Modeling Example

sequence diagrams and activity diagrams. In the following, this section illustrates the applied refinement procedure to deal with the characteristics of distributed systems that are different from local procedure calls. The addressed characteristics of distributed systems are:

- Message Loss: A message might be lost on the communication channel or by the sender itself.
- Delay: A message might be delayed on the communicaton channel or the response of the sender might delayed.

In consequence, it is necessary to model communication protocols as the asynchronous exchange of messages. That means each method call is equivalent to the sending of a message.

Figure B.7 shows the applied process to refine an interface in order to cope with asynchronity [BD11]. The component diagram of Figure B.7(a) shows the components that communicate by using the *Client Server Interface* across a network. The *Server* offers the interface, which is required by the *Client*. Details on the interface are contained in the corresponding class diagram.

Figure B.7(b) refines Figure B.7(a) by splitting the *Client Server Interface* into two separate interfaces. The *Client Server Request Interface* is used by the client to submit the request. In contrast, the *Client Server Response Interface*, which is provided by the client, is used by the server to respond to the request.

The diagrams of Figure B.7 do not impose any constraints on the order of method calls. Therefore, the behavioral view has to specify the correct order for the exchange of messages. Figure B.8 uses activity diagrams to specify the order of the exchanged messages. Exchanged messages are modeled by so called signals that interconnect the client with the server. In contrast to conventional SDL diagrams, only activities exist. That means states are not explicitly modeled. For example the server waits until the signal *request(String url)* is received.

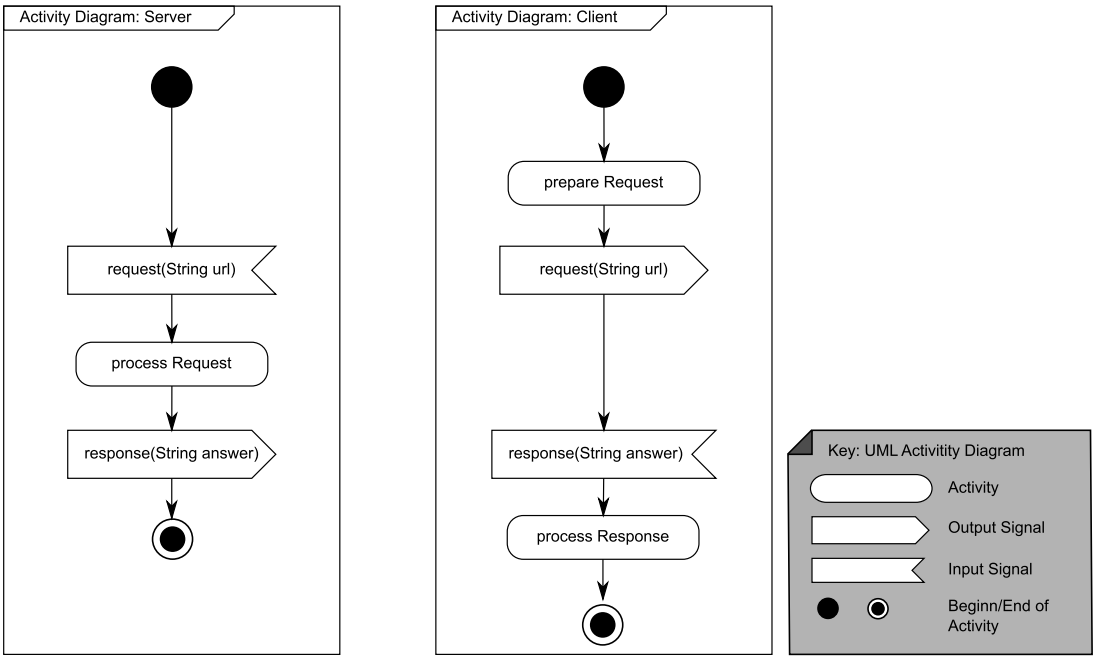


Figure B.8: Activity Diagram for Protocol Modeling Example

Bibliography

[3gpa] Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (GBA) – 3GPP 33.220, 3GPP.

[3gpb] Generic Authentication Architecture (GAA); Support for Subscriber Certificates – 3GPP TS 33.221, 3GPP.

[AB+09] R. Azevedo, M. Barisch, et al. SWIFT Deliverable D403 – SWIFT Mobility Architecture, June 2009.

[AB+10] R. Azevedo, M. Barisch, et al. SWIFT Deliverable D207b – Final SWIFT Architecture, April 2010.

[ACC+08] A. Armando, R. Carbone, J. Cuellar, L. Tobarra, and L. Compagna. Formal Analysis of SAML 2.0 Web Browser Single Sign-On: Breaking the SAML-based Single Sign-On for Google Apps. In *Proceedings of FMSE 2008*. ACM Press, 2008.

[AGK03] J. Argyrakis, S. Gritzalis, and C. Kioulafas. Privacy Enhancing Technologies: A Review. In *Electronic Government*, volume 2739 of *Lecture Notes in Computer Science*, pages 1079–1079. Springer Berlin / Heidelberg, 2003. 10.1007/10929179_51.

[AIDV07] R. Atkinson, J. Irvine, J. Dunlop, and S. Vadgama. The Personal Distributed Environment. *Wireless Communications, IEEE*, 14:62 –69, 2007.

[Ale03] I. Alexander. Misuse Cases: Use Cases with Hostile Intent. *Software, IEEE*, 20:58 – 66, 2003.

[AM+04] A. Abran, J. W. Moore, et al., editors. *Guide to the Software Engineering Body of Knowledge - SWEBOK*. IEEE, 2004.

[Ama11] Amazon Cloud Drive. https://www.amazon.com/clouddrive/learnmore, 2011. [Retrieved: March, 28th 2012].

[Amo94] E. Amoroso. *Fundamentals of Computer Security Technology*. Prentice Hall, 1994.

[And07] R. J. Anderson. *Security Engineering: A Guide to Building Dependable Systems*. Wiley, 2007.

[Aut05] Authentication Context for the OASIS Security Assertion Markup Language (SAML) V2.0, OASIS, 2005.

[AW11] K. Altinkemer and T. Wang. Cost and Benefit Analysis of Authentication Systems. *Decision Support Systems*, 51(3):394 – 404, 2011.

[B+08] V. Bertocci et al. *Understanding Windows CardSpace: An Introduction to the Concepts and Challenges of Digital Identities*. Addison-Wesley Longman, 2008.

[B+09] M. Barisch et al. SWIFT Deliverable D303 – Specification of Identity-centric Security Modules and Cross-layer Interfaces, October 2009.

[Bal09] H. Balzert. *Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering*. Spektrum Akademischer Verlag, 2009.

[Bar09] M. Barisch. Modelling the Impact of Virtual Identities on Communication Infrastructures. In *Proceedings of the 5th ACM workshop on Digital Identity Management*, pages 45–52, New York, NY, USA, 2009. ACM.

[Bar11] M. Barisch. Design and Evaluation for Ubiquitous User Authentication based on Identity Management Systems. In *IEEE International Workshop on Trust and Identity in Mobile Internet, Computing and Communications*, November 2011.

[Bas93] R. Baskerville. Information Systems Security Design Methods: Implications for Information Systems Development. *ACM Comput. Surv.*, 25:375–414, December 1993.

[BBB+10] M. Barisch, S. Berg, A. Brodt, L. Geiger, T. Gerpott, A. Gutscher, C. Hubig, M. Kühlewind, P. Kühn, and O. Siemoneit. Aspekte der Abrechenbarkeit und Bepreisung kontextbezogener Systeme. Technical report, SFB 627 Bericht Nr. 2010/03, February 2010.

[BBC+05] R. Bless, E.-O. BlaSS, M. Conrad, H.-J. Hof, K. Kutzner, S. Mink, and M. Schöller. *Sichere Netzwerkkommunikation: Grundlagen, Protokolle und Architekturen*. Springer, Berlin, 2005.

[BC08] R. Bowen and K. Coar. *Apache Cookbook*. O'Reilly, 2008.

[BD07] J. Brittain and I. Darwin. *Tomcat: The Definitive Guide*. O'Reilly, 2007.

[BD11] D. G. Barrera and M. Diaz. *Communicating Systems with UML 2: Modeling and Analysis of Network Protocols*. Wiley-ISTE, 2011.

[BDP07] A. J. Blazic, K. Dolinar, and J. Porekar. Enabling Privacy in Pervasive Computing Using Fusion of Privacy Negotiation, Identity Management and Trust Management Techniques. In *Proceedings of the First International Conference on the Digital Society*, Washington, DC, USA, 2007. IEEE Computer Society.

[Bea06] C. M. Beaumier. Multifactor Authentication: A Blow to Identity Theft. *Bank Accounting & Finance*, 2:33–37, 2006.

[Ben10] K. Benton. The Evolution of 802.11 Wireless Security. Technical report, University of Nevada Las Vegas Informatics, 2010.

[Ber07] R. Bernard. Information Lifecycle Security Risk Assessment: A Tool for Closing Security Gaps. *Computers & Security*, 26(1):26 – 30, 2007.

[BG05] S. Barnum and M. Gegick. Design Principles. https://buildsecurityin.us-cert.gov/bsi/articles/knowledge/principles/358-BSI.html, September 2005. [Retrieved: March, 28th 2012].

[BH06] J. Bowen and M. Hinchey. Ten Commandments of Formal Methods... Ten Years later. *Computer*, 39(1):40 – 48, 2006.

[BHHN02] K. Boman, G. Horn, P. Howard, and V. Niemi. UMTS security. *Electronics Communication Engineering Journal*, 14:191 – 204, 2002.

[Bir07] D. G. Birch, editor. *Digital Identity Management: Perspectives On The Technological, Business and Social Implications*. Gower Publishing Ltd, 2007.

[Bis04] M. Bishop. *Introduction to Computer Security*. Addison-Wesley Professional, 2004.

[Bis09] J. Biskup. *Security in Computing Systems*. Springer-Verlag, 2009.

[BJM08] A. Barth, C. Jackson, and J. C. Mitchell. Robust Defenses for Cross-site Request Forgery. In *Proceedings of the 15th ACM conference on Computer and communications security*, CCS '08, pages 75–88, New York, NY, USA, 2008. ACM.

[BK03] M. Bjerkander and C. Kobryn. Architecting Systems with UML 2.0. *Software, IEEE*, 20(4):57 – 61, july-aug. 2003.

[BKM09] M. Barisch, J. Kögel, and S. Meier. A Flexible Framework for Complete Session Mobility and its Implementation. In *Proceedings of the 15th Open European Summer School (EUNICE 2009)*, September 2009.

[BMM11] E. Bursztein, M. Martin, and J. Mitchell. Text-based CAPTCHA Strengths and Weaknesses. In *Proceedings of the 18th ACM Conference on Computer and Communications Security*, CCS '11, pages 125–138, New York, NY, USA, 2011. ACM.

[BNP+08] M. Barisch, M. Neubauer, J. Pagaime, J. Girao, and R. Aguiar. Privacy and Identity Management in a Layered Pervasive Service Platform. In *Proceedings of the ICT Mobile and Wireless Communications Summit 2008 (ICT-MobileSummit 2008)*, 2008.

[Boe08] J. Boegh. A New Standard for Quality Requirements. *Software, IEEE*, 25(2):57 –63, March 2008.

[BOJ+06] M. Bauer, R. L. Olsen, M. Jacobsson, L. Sanchez, J. Lanza, M. Imine, and N. Prasad. Context Management Framework for MAGNET Beyond. In *Open International Workshop on Capturing Context and Context Aware Systems and Platforms*, 2006.

[Bry03] J. M. Bryson. What To Do When Stakeholders Matter : A Guide to Stakeholder Identification and Analysis Techniques By What To Do When Stakeholders Matter : A Guide to Stakeholder Identification and Analysis Techniques. *Business*, 6(February):1–40, 2003.

[BSG08] M. Barisch, A. Sarma, and J. Girao. SWIFT - Integriertes Identitätsmanagement für Netzbetreiber und Dienstanbieter. Essener Workshop zur Netzsicherheit 2008 (EWNS08), 2008.

[BSI05] IT-Grundschutz Catalogues: Version 2005, Federal Office for Information Security, Germany, 2005.

[BSI08a] BSI-Standard 100-1 Managementsysteme für Informationssicherheit (ISMS) – Version 1.5, Bundesamt für Sicherheit in der Informationstechnik (BSI), 2008.

[BSI08b] BSI-Standard 100-2 IT-Grundschutz-Vorgehensweise – Version 2.0, Bundesamt für Sicherheit in der Informationstechnik (BSI), 2008.

[BSI08c] BSI-Standard 100-3 Risikoanalyse auf der Basis von IT-Grundschutz, Bundesamt für Sicherheit in der Informationstechnik (BSI), 2008.

[BSI08d] BSI-Standard 100-4 Notfallmanagement – Version 1.0, Bundesamt für Sicherheit in der Informationstechnik (BSI), 2008.

[BSI11] IT-Grundschutz Kataloge – 12. Ergänzungslieferung - September 2011, Federal Office for Information Security, Germany, 2011.

[BTL+10] M. Barisch, E. Torroglosa, M. Lischka, R. Marques, R. Marx, A. Matos, A. Perez, and D. Scheuermann. Security and Privacy Enablers for Future Identity Management Systems. In *Future Network & Mobile Summit 2010*, 2010.

[Bui11] BuiltWith.com – OpenId Usage Trends. http://trends.builtwith.com/docinfo/OpenID, Dezember 2011. [Retrieved: March, 28th 2012].

[Bur02] P. Burkeholder. SSL Man-in-the-Middle Attacks. SANS Institute, February 2002.

[BVO11] D. Barrera and P. Van Oorschot. Secure Software Installation on Smartphones. *Security Privacy, IEEE*, 9:42 –48, 2011.

[C+05a] S. Cantor et al. Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS standard, March 2005.

[C+05b] S. Cantor et al. Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS standard, March 2005.

[C+06] P. Chandra et al. OWASP - Comprehensive, Lightweight Application Security Process (CLASP). `https://www.owasp.org/index.php/Category:OWASP_CLASP_Project`, 2006.

[Cam04] J. Camp. Digital Identity. *Technology and Society Magazine, IEEE*, 23(3):34 – 41, fall 2004.

[CAN11] T. Chen and S. Abu-Nimeh. Lessons from Stuxnet. *Computer*, 44(4):91 –93, April 2011.

[Car11] Beyond Windows CardSpace. http://blogs.msdn.com/b/card/archive/2011/02/15/beyond-windows-cardspace.aspx, February 2011. [Retrieved: March, 28th 2012].

[CBB+10] P. Clements, F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, P. Merson, R. Nord, and J. Stafford. *Documenting Software Architectures*, volume 2nd edition. Addison-Wesley, 2010.

[CBU08] *Cambridge Advanced Learner's Dictionary*. Cambridge Univ Elt, 2008.

[CCC] Common Methodology for Information Technology Security Evaluation, Common Criteria.

[CCPa] Common Criteria for Information Technology Security Evaluation - Part 1: Introduction and general model – Version 3.1, Common Criteria.

[CCPb] Common Criteria for Information Technology Security Evaluation - Part 2: Security functional components – Version 3.1, Common Criteria.

[CCPc] Common Criteria for Information Technology Security Evaluation - Part 3: Security Assurance Components – Version 3.1, Common Criteria.

[CEVG11] N. Christin, S. Egelman, T. Vidas, and J. Grossklags. It's All About the Benjamins: An Empirical Study on Incentivizing Users to Ignore Security Advice. In *Proceedings Financial Crypto 2011*, February 2011.

[CGS01] E. Casilari, F. Gonzblez, and F. Sandoval. Modeling of HTTP Traffic. *Communications Letters, IEEE*, 5(6):272 –274, jun 2001.

[Cha85] D. Chaum. Security without Identification: Transaction Systems to Make Big Brother Obsolete. *Commun. ACM*, 28:1030–1044, October 1985.

[Cha05] R. N. Charette. Why Software Fails. *IEEE Spectrum*, 2005.

[Cha09] D. Chadwick. Federated Identity Management. In A. Aldini, G. Barthe, and R. Gorrieri, editors, *Foundations of Security Analysis and Design V*, volume 5705 of *Lecture Notes in Computer Science*, pages 96–120. Springer Berlin / Heidelberg, 2009. 10.1007/978-3-642-03829-7_3.

[Che01] P.-C. Cheng. An architecture for the Internet Key Exchange Protocol. *IBM Systems Journal*, 40(3):721 –746, 2001.

[Chi04] H. Chivers. Security and Systems Engineering. Technical report, Department of Computer Science, University of York, 2004.

[CI09] D. Chadwick and G. Inman. Attribute Aggregation in Federated Identity Management. *Computer*, 42:33 –40, 2009.

[CJYF06] D. Chakraborty, A. Joshi, Y. Yesha, and T. Finin. Toward Distributed Service Discovery in Pervasive Computing Environments. *Mobile Computing, IEEE Transactions on*, 5(2):97–112, 2006.

[CK01] S. Claußand M. Köhntopp. Identity Management and its Support of Multilateral Security. *Comput. Netw.*, 37:205–219, August 2001.

[CK11] S. Cheshire and M. Krochmal. DNS-Based Service Discovery – IETF Draft draft-cheshire-dnsext-dns-sd-10.txt. http://files.dns-sd.org/draft-cheshire-dnsext-dns-sd.txt, February 2011.

[CL12] C.-C. Chang and C.-Y. Lee. A Secure Single Sign-On Mechanism for Distributed Computer Networks. *Industrial Electronics, IEEE Transactions on*, 59(1):629 – 637, jan. 2012.

[Cla11] N. Clarke. *Transparent User Authentication: Biometrics, RFID and Behavioural Profiling*. Springer, 2011.

[CM04] B. Chess and G. McGraw. Static Analysis for Security. *Security Privacy, IEEE*, 2:76 – 79, 2004.

[CMCP06] D. Calin, A. R. McGee, U. Chandrashekhar, and R. Prasad. MAGNET: An Approach for Secure Personal Networking in Beyond 3G Wireless Networks. *Bell Labs Technical Journal*, 11(1):79–98, 2006.

[Cor] Corisecio. Mobile Cardspace. http://www.corisecio.de/. [Retrieved: March, 28th 2012].

[CVH02] J. Camenisch and E. Van Herreweghen. Design and Implementation of the Idemix Anonymous Credential System. In *Proceedings of the 9th ACM conference on Computer and communications security*, CCS '02, pages 21–30, New York, NY, USA, 2002. ACM.

[Dau10] M. I. Daud. Secure Software Development Model: A Guide for Secure Software Life Cylce. In *Proceedings of the International MultiConference of Engineers and Computer Scientists 2010*, 2010.

[DD08] R. Dhamija and L. Dusseault. The Seven Flaws of Identity Management: Usability and Security Challenges. *IEEE Security and Privacy*, 6:24–29, March 2008.

[DFLP07] N. Delessy, E. Fernandez, and M. Larrondo-Petrie. A Pattern Language for Identity Management. In *Computing in the Global Information Technology, 2007. ICCGI 2007. International Multi-Conference on*, 2007.

[DH76] W. Diffie and M. Hellman. New directions in cryptography. *Information Theory, IEEE Transactions on*, 22(6):644 – 654, nov 1976.

[DHJ+10] M. Debbabi, F. Hassaïne, Y. Jarraya, A. Soeanu, and L. Alawneh. *Verification and Validation in Systems Engineering: Assessing UML/SysML Design Models*. Springer, 1st edition, 2010.

[Dig] DigitalMe Project. http://code.bandit-project.org/trac/wiki/DigitalMe. [Retrieved: March, 28th 2012].

[Dig11] Fraudulently issued security certificate discovered. http://www.govcert.nl/english/service-provision/knowledge-and-publications/factsheets/factsheet-fraudulently-issued-security-certificate-discovered.html, August 2011. [Retrieved: March, 28th 2012].

[DMR10] M. Dell'Amico, P. Michiardi, and Y. Roudier. Password Strength: An Empirical Analysis. In *Proceedings of the 29th conference on Information communications*, INFOCOM'10, pages 983–991, Piscataway, NJ, USA, 2010. IEEE Press.

[DRALN08] B. Desmond, J. Richards, R. Allen, and A. G. Lowe-Norris. *Active Directory: Designing, Deploying, and Running Active Directory, Fourth Edition*. O'Reilly Media, 4th edition, 2008.

[Dre11] C. Drew. Data Breach at Security Firm Linked to Attack on Lockheed. http://www.nytimes.com/2011/05/28/business/28hack.html?_r=2&nl=todaysheadlines&emc=tha25, May 2011.

[DRHH10] P. Datta Ray, R. Harnoor, and M. Hentea. Smart Power Grid Security: A Unified Risk Management Approach. In *Security Technology (ICCST), 2010 IEEE International Carnahan Conference on*, 2010.

[DS00] P. T. Devanbu and S. Stubblebine. Software Engineering for Security: A Roadmap. In *Proceedings of the Conference on The Future of Software Engineering*, ICSE '00, pages 227–239, New York, NY, USA, 2000. ACM.

[DWSB+09] B. De Win, R. Scandariato, K. Buyens, J. Grégoire, and W. Joosen. On the Secure Software Development Process: CLASP, SDL and Touchpoints compared. *Inf. Softw. Technol.*, 51:1152–1171, July 2009.

[Eck09] C. Eckert. *IT-Sicherheit – Konzepte - Verfahren - Protokolle*. Oldenbourg, 2009.

[edu] eduroam. http://www.eduroam.org. [Retrieved: March, 28th 2012].

[EOM09] W. Enck, M. Ongtang, and P. McDaniel. Understanding Android Security. *Security Privacy, IEEE*, 7(1):50 –57, jan.-feb. 2009.

[Eri99] C. A. Ericson. Fault Tree Analyis - A History. In *Proceedings of the 17th International System Safety Conference*, 1999.

[F+11] M. Fossi et al. Symantec Internet Security Threat Report – Trends for 2010. Technical report, Symantec Cooperation, April 2011.

[FB01] M. Franceschetti and J. Bruck. A Group Membership Algorithm with a Practical Specification. *IEEE Trans. Parallel Distrib. Syst.*, 12:1190–1200, November 2001.

[FB11] H. S. Fhom and K. M. Bayarou. Towards a Holistic Privacy Engineering Approach for Smart Grid Systems. In *in Proceedings of IEEE TrustCom 2011*, 2011.

[FH07] D. Florencio and C. Herley. A large-scale Study of Web Password Habits. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 657–666, New York, NY, USA, 2007. ACM.

[FH10] D. Florêncio and C. Herley. Where do Security Policies come from? In *Proceedings of the Sixth Symposium on Usable Privacy and Security*, SOUPS '10, pages 10:1–10:14, New York, NY, USA, 2010. ACM.

[Fir] Firefox Sync Client Documentation. http://docs.services.mozilla.com/sync/index.html. [Retrieved: March, 28th 2012].

[Fir04] D. Firesmith. Specifying Reusable Security Requirements. *Journal of Object Technology*, 3(1):61–75, 2004.

[Fir05] D. G. Firesmith. Analyzing the Security Significance of System Requirements. In *The 13th IEEE Requirements Engineering Conference 2005*, 2005.

[FKK96] A. O. Freier, P. Karlton, and P. C. Kocher. Internet Draft - The SSL Protocol Version 3.0. http://www.mozilla.org/projects/security/pki/nss/ssl/draft302.txt, Internet Engineering Task Force, 1996.

[FM94] V. Frost and B. Melamed. Traffic Modeling for Telecommunications Networks. *Communications Magazine, IEEE*, 32(3):70 –81, mar 1994.

[Fol11] M. J. Foley. The 10 sexiest Microsoft Business Teases for 2012. http://www.zdnet.com/blog/microsoft/the-10-sexiest-microsoft-business-teases-for-2012/11492, December 2011.

[Fri07] L. Fritsch. State of the Art of Privacy-enhancing Technology (PET) – Deliverable D2.1 of the PETweb project. http://publications.nr.no/PETs_privacy_LotharFritsch_ID-tyveri2010.pdf, November 2007. [Retrieved: March, 28th 2012].

[FSF+04] R. Y. Fu, H. Su, J. C. Fletcher, W. Li, X. X. Liu, S. W. Zhao, and C. Y. Chi. A Framework for Device Capability on Demand and Virtual Device User Experience. *IBM Journal of Research and Development*, 48(5):635–648, 2004.

[Fun10] Funambol Open Source Device Management for 4G and 3G Networks and Devices. http://funambol.com/documents/Funambol_DMdatasheet_July2010.pdf, July 2010.

[G+06] G. Gow et al. Privacy Rights and Prepaid Communication Services – A Survey of Prepaid Mobile Phone Regulation and Registration Policies among OECD Member States. Technical report, Centre for Policy Research on Science and Technology – Simon Fraser University Vancouver, 2006.

[GB+08] Y. Grossman, M. Barisch, et al. Daidalos Deliverable D531 – Daidalos II Testing and Evaluation Report, December 2008.

[GC07] B. M. Gross and E. F. Churchill. Addressing Constraints: Multiple Usernames Task Spillage and Notions of Identity. In *CHI '07 Extended Abstracts on Human factors in Computing Systems*, CHI EA '07, pages 2393–2398, New York, NY, USA, 2007. ACM.

[GCB+08] S. Garriss, R. Cáceres, S. Berger, R. Sailer, L. van Doorn, and X. Zhang. Trustworthy and Personalized Computing on Public Kiosks. In *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services*, MobiSys '08, pages 199–210, New York, NY, USA, 2008. ACM.

[GHJV94] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1994.

[GHJW03] J. Grünbauer, H. Hollmann, J. Jürjens, and G. Wimmel. Modelling and Verification of Layered Security Protocols: A Bank Application. In *In Computer Safety, Reliability, and Security (SAFECOMP 2003), volume 2788 of LNCS*, pages 116–129. Springer, 2003.

[Gil11] C. Gillespie. Character Occurence in Passwords. http://csgillespie.wordpress.com/2011/06/16/character-occurrence-in-passwords/, June 2011. [Retrieved: March, 28th 2012].

[GM09] E. Gallery and C. J. Mitchell. Trusted Computing: Security and Applications. *Cryptologia*, 33:217–245, 2009.

[GMMJ11] N. Gunson, D. Marshall, H. Morton, and M. Jack. User Perceptions of Security and Usability of Single-factor and Two-factor Authentication in Automated Telephone Banking. *Computers and Security*, 30(4):208 – 220, 2011.

[Goo11] Federated Login for Google Account Users. http://code.google.com/intl/de/apis/accounts/docs/OpenID.html, 2011. [Retrieved: March, 28th 2012].

[Gro03] T. Groß. Security Analysis of the SAML Single Sign-on Browser/Artifact Profile. In *Proceedings of the 19th Annual Computer Security Applications Conference*, ACSAC '03, pages 298–, Washington, DC, USA, 2003. IEEE Computer Society.

[GSM09] GSM Association – Market Data Summary. http://www.gsmworld.com/newsroom/market-data/market_data_summary.htm, 2009. [Retrieved: November, 2nd 2012].

[GSSX09] S. Gajek, J. Schwenk, M. Steiner, and C. Xuan. Risks of the CardSpace Protocol. In P. Samarati, M. Yung, F. Martinelli, and C. Ardagna, editors, *Information Security*, volume 5735 of *Lecture Notes in Computer Science*, pages 278–293. Springer Berlin / Heidelberg, 2009. 10.1007/978-3-642-04474-8_23.

[Gue87] R. Guerin. Channel Occupancy Time Distribution in a Cellular Radio System. *Vehicular Technology, IEEE Transactions on*, 36(3):89 – 99, aug 1987.

[GV08] U. Glasser and M. Vajihollahi. Identity management architecture. In *Intelligence and Security Informatics, 2008. ISI 2008. IEEE International Conference on*, pages 137 –144, june 2008.

[GW07] M. Glinz and R. Wieringa. Guest Editors' Introduction: Stakeholders in Requirements Engineering. *Software, IEEE*, 24:18 –20, 2007.

[H+05] F. Hirsch et al. Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V2.0, 2005.

[Hau08] C. Hauser. *Protecting Virtual Identities in Mobile IP-based Communication - Communication Networks and Computer Engineering Report No. 98*. PhD thesis, Universität Stuttgart, 2008.

[HCR10] C. J. Hernandez-Castro and A. Ribagorda. Pitfalls in CAPTCHA Design and Implementation: The Math CAPTCHA, a Case Study. *Computers and Security*, 29(1):141 – 157, 2010.

[Hen06] P. A. Henry. Two-Factor Authentication – a look behind the headlines. *Network Security*, 2006(4):18 – 19, 2006.

[HHch] J. Hewlett and S. Hartman. Project Moonshot - Briefing paper for IETF 77, Anaheim, 2010 March.

[Hig] Higgins Project. http://eclipse.org/higgins/. [Retrieved: March, 28th 2012].

[HJK08] P. Harding, L. Johansson, and N. Klingenstein. Dynamic Security Assertion Markup Language: Simplifying Single Sign-On. *Security Privacy, IEEE*, 6(2):83 –85, march-april 2008.

[HL06] M. Howard and S. Lipner. *The Security Development Lifecycle*. Microsoft Press, 2006.

[HLOS06] S. Hernan, S. Lambert, T. Ostwald, and A. Shostack. Uncover Security Design Flaws Using the STRIDE Approach. *MSDN Magazine*, November 2006.

[HRF+07] D. Hunter, J. Rafter, J. Fawcett, E. van der Vlist, D. Ayers, J. Duckett, A. Watt, and L. McKinnon. *Beginning XML, 4th Edition (Programmer to Programmer)*. Wrox, 4 edition, 5 2007.

[HRZ10] D. Hühnlein, H. Roßnagel, and J. Zibuschka. Diffusion of Federated Identity Management. In *Sicherheit*, pages 25–36, 2010.

[Hun11] T. Hunt. A brief Sony Password Analysis. http://www.troyhunt.com/2011/06/brief-sony-password-analysis.html, June 2011. [Retrieved: March, 28th 2012].

[HZO+11] H. Huang, S. Zhang, X. Ou, A. Prakash, and K. Sakallah. Distilling Critical Attack Graph Surface iteratively through minimum-cost SAT Solving. In *Proceedings of the 27th Annual Computer Security Applications Conference*, ACSAC '11, pages 31–40, New York, NY, USA, 2011. ACM.

[Hüb08] M. Hübler. Untersuchung und Realisierung eines verteilten Passwortmanagers. Master's thesis, University of Stuttgart, Institute for Communication Networks and Computer Engineering, 2008.

[iCl] Apple iCloud. http://www.apple.com/icloud/, 2011. [Retrieved: March, 28th 2012].

[IEEa] 802.1X IEEE Standard for Local and Metropolitan Area Networks - Port-Based Network Access Control, IEEE.

[IEEb] IEEE – List of Milestones. http://www.ieeeghn.org/wiki/index.php/Milestones:List_of_IEEE_Milestones, year = 2011, note = [Retrieved: March, 28th 2012], owner = barisch, timestamp = 2011.12.03.

[IEEc] IEEE 802.11i Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications - Amendment 6: Medium Access Control (MAC) Security Enhancements, IEEE.

[IEE07] IEEE 802.11 Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, IEEE, 2007.

[ISOa] ISO 7498-2:1989 Information Processing Systems – Open Systems Interconnection – Basic Reference Model – Part 2: Security Architecture, International Organization for Standardization.

[ISOb] ISO/IEC 15408-1:2009 Common Criteria for Information Technology Security Evaluation - Part 1: Introduction and general model – Version 3.1, Internation Organisation for Standardisation.

[ISOc] ISO/IEC 15408-2:2009 Common Criteria for Information Technology Security Evaluation - Part 2: Security functional components – Version 3.1, Internation Organisation for Standardisation.

[ISOd] ISO/IEC 15408-3:2009 Common Criteria for Information Technology Security Evaluation - Part 3: Security Assurance Components – Version 3.1, Internation Organisation for Standardisation.

[ISOe] ISO/IEC/IEEE 24765 Systems and Software Engineering – Vocabulary, International Organization for Standardisation.

[IT91] ITU-T. Security architecture for Open Systems Interconnection for CCITT applications. Rec. X.800, ITU-T, March 1991.

[IT00] ITU-T. Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks. Rec. X.509, ITU-T, March 2000.

[J+05] A. Josang et al. Trust Requirements in Identity Management. In *Proceedings of the 2005 Australasian workshop on Grid computing and e-research*, pages 99–108, Darlinghurst, Australia, 2005.

[J+08] J. Jähnert et al. Description of DAIDALOS II Scenario Design Report. www.ict-daidalos.org, September 2008.

[Jaq07] A. Jaquith. *Security Metrics: Replacing Fear, Uncertainty, and Doubt*. Addison-Wesley Professional, 1st edition, 2007.

[JGtM00] U. Jendricke and D. Gerd tom Markotten. Usability meets Security - the Identity-Manager as your Personal Security Assistant for the Internet. In *Computer Security Applications, 2000. ACSAC '00. 16th Annual Conference*, 2000.

[JKZ02] U. Jendricke, M. Kreutzer, and A. Zugenmaier. Mobile Identity Management. In *Inproceedings of Ubicomp workshop*, 2002.

[JMRA08] U. Javaid, D.-E. Meddour, T. Rasheed, and T. Ahmed. Mobility Management Architecture for Personal Ubiquitous Environments. In *Personal, Indoor and Mobile Radio Communications, 2008. PIMRC 2008. IEEE 19th International Symposium on*, pages 1 –5, sept. 2008.

[Jul11] Julian. 10 Most Costly Cyber Attacks in History. http://www.businesspundit.com/10-most-costly-cyber-attacks-in-history/, August 2011. [Retrieved: March, 28th 2012].

[JZS07] A. Jøsang, M. A. Zomai, and S. Suriadi. Usability and privacy in identity management architectures. In *Proceedings of the fifth Australasian symposium on ACSW frontiers - Volume 68*, ACSW '07, pages 143–152, Darlinghurst, Australia, Australia, 2007. Australian Computer Society, Inc.

[Jür04] J. Jürjens. *Secure Systems Development with UML*. Springer, 2004.

[K+08] P. J. Kühn et al. Lecture Notes on Communication Networks II – Edition 2008, 2008.

[Kan] Kantara Initiative. http://kantarainitiative.org/.

[KCLC07] P. Kumaraguru, L. F. Cranor, J. Lobo, and S. B. Calo. A Survey of Privacy Policy Languages. In *Inproceedings of Symposium On Usable Privacy and Security*, 2007.

[Kin11] R. King. Hackers attack Citi, Access Data of over 200,000 Bank Accounts. http://www.zdnet.com/blog/btl/hackers-attack-citi-access-data-of-over-200000-bank-accounts/50267, June 2011. [Retrieved: March, 28th 2012].

[KMKI07] P. Koster, J. Montaner, N. Koraichi, and S. Iacob. Introduction of the Domain Issuer in OMA DRM. In *Consumer Communications and Networking Conference, 2007. CCNC 2007. 4th IEEE*, 2007.

[KN07] G. Kalman and J. Noll. SIM as Secure Key Storage in Communication Networks. In *Wireless and Mobile Communications, 2007. ICWMC '07. Third International Conference on*, page 55, march 2007.

[Koe04] L. Koetzle. Is Linux More Secure Than Windows? – Forrester Market Overview, March 2004.

[Koi04] G. Koien. An Introduction to Access Security in UMTS. *Wireless Communications, IEEE*, 11(1):8 – 18, feb 2004.

[KRC06] C. Kuo, S. Romanosky, and L. F. Cranor. Human Selection of Mnemonic Phrase-based Passwords. In *Proceedings of the second symposium on Usable privacy and security*, SOUPS '06, pages 67–78, New York, NY, USA, 2006. ACM.

[KSTU09] A. Kumar, N. Saxena, G. Tsudik, and E. Uzun. A Ccomparative Study of Secure Device Pairing Methods. *Pervasive and Mobile Computing*, 5(6):734 – 749, 2009. PerCom 2009.

[KZ09] K. A. Kluever and R. Zanibbi. Balancing Usability and Security in a Video CAPTCHA. In *Proceedings of the 5th Symposium on Usable Privacy and Security*, SOUPS '09, pages 14:1–14:11, New York, NY, USA, 2009. ACM.

[Käh06] U. Kähler. Sesam Öffne Dich – DFN-AAI bietet kontrollierten Zugang zu geschützten Ressourcen. *DFN Mitteilungen*, 70:10–13, 2006.

[Küh79] P. J. Kühn. Approximate Analysis of General Queuing Networks by Decomposition. *IEEE Transactions on Communications*, 27:113–126, 1979.

[Küh02] P. J. Kühn. Lecture Notes on Teletraffic Theory and Engineering – Edition 2002/2003. Institute of Communication Networks and Computer Engineering, 2002.

[L+08] M. Lischka et al. SWIFT Deliverable D502 - SWIFT Scenarios, Use Cases and Business Models. www.swift-ict.org, December 2008.

[Lag11] Die Lage der IT-Sicherheit in Deutschland 2011. Bundesamt für Sicherheit in der Informationstechnik – BSI, Mai 2011.

[LAI09] Liberty Alliance – Interoperability Matrix. http://www.projectliberty.org/liberty/liberty_interoperable/implementations/saml_2_0_test_procedure_v3_2_2_full_matrix_implementation_table_q309/, 2009. [Retrieved: November, 26th 2011].

[Lan11] R. Langner. Stuxnet: Dissecting a Cyberwarfare Weapon. *Security Privacy, IEEE*, 9:49 –51, 2011.

[Lasa] Last Pass Technology. https://lastpass.com/technology. [Retrieved: November, 13th 2011].

[LASb] Liberty Alliance Specifications. http://www.projectliberty.org/liberty/ resource_center/specifications. [Retrieved: November, 26th 2011].

[LCGSG09] G. Lopez, O. Canovas, A. Gomez-Skarmeta, and J. Girao. A SWIFT Take on Identity Management. *Computer*, 42(5):58 –65, may 2009.

[LL11] B. Laurie and A. Langley. Certificate Authority Transparency and Auditability. http://www.links.org/files/CertificateAuthorityTransparencyandAuditability.pdf, November 2011. [Retrieved: March, 28th 2012].

[Lor03] S. Lord. Trouble at the Telco: When GSM Goes Bad. *Network Security*, 2003(1):10 – 12, 2003.

[LSW10] H. Lohr, A.-R. Sadeghi, and M. Winandy. Patterns for Secure Boot and Secure Storage in Computer Systems. In *Availability, Reliability, and Security, 2010. ARES '10 International Conference on*, 2010.

[Mad08] P. Madsen. Liberty ID-WSF Multi-Device SSO Deployment Guide. http://www.projectliberty.org/liberty/content/download/4473/30604/file/draft-liberty-idwsf-mdsso-deployguide-v1.0-02.pdf, 10 2008.

[Mai06] N. Maiden. Improve Your Requirements: Quantify Them. *Software, IEEE*, 23:68 –69, 2006.

[Man10] U. Mansmann. Inkognito – Lebensmittel-Discounter schlampen bei der Identitätsprüfung für SIM-Karten. *c't magazin*, 05:82–84, 2010.

[Mar01] J. J. Marciniak, editor. *Encyclopedia of Software Engineering*. Wiley-Interscience, 2001.

[MB+09] R. Marx, M. Barisch, et al. SWIFT Deliverable D302 – Specification of General Identity-centric Security Model that supports user control of privacy, January 2009.

[MB+10] A. P. Méndez, M. Barisch, et al. SWIFT Deliverable D207a – Final SWIFT Architecture, March 2010.

[McG06] G. McGraw. *Software Security: Building Security In*. Addison-Wesley Longman, 2006.

[ME10] A. Metke and R. Ekl. Smart Grid security technology. In *Innovative Smart Grid Technologies (ISGT), 2010*, 2010.

[Mei08] J. Meier. Security Principles. http://blogs.msdn.com/b/jmeier/archive/2008/04/07/security-principles.aspx, April 2008.

[Mes10] E. Messmer. Identity Management Top Security Priority in Gartner Survey. http://www.networkworld.com/news/2010/061010-gartner-security-identity-management.html, June 2010. [Retrieved: March, 28th 2012].

[MGM03] H. Mouratidis, P. Giorgini, and G. Manson. Integrating Security and Systems Engineering: Towards the Modelling of Secure Information Systems. In *In Proceedings of the 15th Conference On Advanced Information Systems Engineering (CAiSE)*, pages 63–78. Springer-Verlag, 2003.

[MH78] R. Merkle and M. Hellman. Hiding Information and Signatures in Trapdoor Knapsacks. *Information Theory, IEEE Transactions on*, 24:525 – 530, 1978.

[Mit01] C. J. Mitchell. The Security of the GSM Air Interface Protocol. Technical report, Royal Holloway University London, 2001.

[MPS+93] S. Muftic, A. Patel, P. Sanders, R. Colon, J. Heijjnsdijk, and U. Pulkkinen. *Security Architecture for Open Distributed Systems*. Wiley, 1993.

[MS10] G. A. Moreno and C. U. Smith. Performance Analysis of Real-time Component Architectures: An enhanced Model Interchange Approach. *Perform. Eval.*, 67:612–633, 2010.

[MSR07] P. Mell, K. Scarfone, and S. Romanosky. A Complete Guide to the Common Vulnerability Scoring System Version 2.0, 2007.

[MW04] U. Meyer and S. Wetzel. A man-in-the-middle attack on UMTS. In *Proceedings of the 3rd ACM workshop on Wireless security*, WiSe '04, pages 90–97, New York, NY, USA, 2004. ACM.

[MZK+05] J. Mallery, J. Zann, P. Kelly, W. Noonan, E. S. Seagren, P. Love, and P. Love. *Hardening Network Security*. McGraw-Hill Osborne Media, 2005.

[NE00] B. Nuseibeh and S. Easterbrook. Requirements Engineering: A Roadmap. In *Proceedings of the Conference on The Future of Software Engineering*, ICSE '00, pages 35–46, New York, NY, USA, 2000. ACM.

[Neu09] M. Neubauer. Modelling of Pseudonymity under Probabilistic Linkability Attacks. In *International Symposium on Secure Computing (SecureCom09)*, August 2009.

[NHdG02] I. Niemegeers and S. Heemstra de Groot. From Personal Area Networks to Personal Networks: A User Oriented Approach. *Wireless Personal Communications*, 22:175–186, 2002. 10.1023/A:1019912421877.

[NIS] Security Considerations in the System Development Life Cycle – NIST Special Publication 800-64, NIST – National Institute of Standards and Technology.

[NIS02] Risk Management Guide for Information Technology Systems - NIST Special Publication 800-30, NIST – Nationale Institute of Standards and Technology, July 2002.

[NIS06] Electronic Authentication Guideline - NIST Special Publication 800-63, NIST – Nationale Institute of Standards and Technology, April 2006.

[NPSQ03] M. Neve, E. Peeters, D. Samyde, and J.-J. Quisquater. Memories: A Survey of Their Secure Uses in Smart Cards. In *Security in Storage Workshop, 2003. SISW '03. Proceedings of the Second IEEE International*, 2003.

[OAS08] Extensible Resource Identifier (XRI) Resolution Version 2.0, OASIS, February 2008.

[OB06] X. Ou and W. F. Boyer. A Scalable Approach to Attack Graph Generation. In *13th ACM Conference on Computer and Communications Security (CCS*, pages 336–345. ACM Press, 2006.

[OBDS04] A. Oprea, D. Balfanz, G. Durfee, and D. Smetters. Securing a remote terminal application with a mobile trusted device. In *Computer Security Applications Conference, 2004. 20th Annual*, pages 438 – 447, dec. 2004.

[OES11] OESIS Framework – OPSWAT Endpoint Security Integration SDK, 2011.

[O'G03] L. O'Gorman. Comparing Passwords, Tokens, and Biometrics for User Authentication. *Proceedings of the IEEE*, 91:2021 – 2040, 2003.

[Old11] E. Olden. Architecting a Cloud-Scale Identity Fabric. *Computer*, 44(3):52 –59, march 2011.

[OMA06] Open Mobile Alliance – User Agent Profile 2.0, February 2006.

[OMA11] Open Mobile Alliance – DRM Architecture Version 2.2, March 2011.

[One96] A. One. Smashing The Stack For Fun And Profit. *Phrack*, 49:14, 1996.

[Ope] OpenID Review. https://sites.google.com/site/openidreview/home. [Retrieved: March, 28th 2012].

[OSG11] OSGi Service Platform Release 4. http://www.osgi.org/Specifications/HomePage, April 2011.

[Pal11] D. Pal. MIT Kerberos & Red Hat – Past, Present and Future. In *Proceedings of 2011 Kerberos Conference*, 2011.

[Par95] T. Parker. Single Sign-On Systems – The Technologies and the Products. In *Security and Detection, 1995., European Convention on*, 1995.

[Par10] H. Partsch. *Requirements Engineering systematisch*. Springer, 2010.

[Pas] Microsoft Passport. https://accountservices.passport.net/ppnetworkhome.srf?lc=1033&mkt=EN-US. [Retrieved: March, 28th 2012].

[Pas10] Facts about the new Identity Card – Bundesministerium des Innern. http://www.personalausweisportal.de/SharedDocs/Downloads/DE/Flyer-und-Broschueren/PersonalausweisbroschuereA6_englisch.pdf?__blob=publicationFile, 2010.

[PH10] A. Pfitzmann and M. Hansen. A Terminology for Talking about Privacy by Data Minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management. http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf, August 2010.

[PLÓCGS11] A. Pérez, G. López, Óscar Cánovas, and A. F. Gómez-Skarmeta. Formal Description of the SWIFT Identity Management Framework. *Future Generation Computer Systems*, 27(8):1113 – 1123, 2011.

[PMDW05] N. Pramstaller, S. Mangard, S. Dominikus, and J. Wolkerstorfer. Efficient AES Implementations on ASICs and FPGAs. In H. Dobbertin, V. Rijmen, and A. Sowa, editors, *Advanced Encryption Standard – AES*, volume 3373 of *Lecture Notes in Computer Science*, pages 571–571. Springer Berlin / Heidelberg, 2005. 10.1007/11506447_9.

[Pow10] K. Power. Stakeholder Identification in Agile Software Product Development Organizations: A Model for Understanding Who and What Really Counts. In *AGILE Conference, 2010*, pages 87 –94, aug. 2010.

[PPSW97] A. Pfitzmann, B. Pfitzmann, M. Schunter, and M. Waidner. Trusting Mobile User Devices and Security Modules. *Computer*, 30:61 –68, 1997.

[Prö11] M. Pröhl. *Kerberos – Single Sign-On in gemischten Windows/Linux-Umgebungen*. dpunkt.verlag, 2011.

[PS02] B. Pinkas and T. Sander. Securing Passwords against Dictionary Attacks. In *Proceedings of the 9th ACM conference on Computer and communications security*, CCS '02, pages 161–170, New York, NY, USA, 2002. ACM.

[PW01] O. Preiss and A. Wegmann. Stakeholder Discovery and Classification Based on Systems Science Principles. In *Proceedings of the Second Asia-Pacific Conference on Quality Software*, Washington, DC, USA, 2001. IEEE Computer Society.

[PW03] B. Pfitzmann and M. Waidner. Analysis of Liberty Single-Sign-On with Enabled Clients. *Internet Computing, IEEE*, 7:38 – 44, 2003.

[PW04] B. Pfitzmann and M. Waidner. *Federated Identity-Management Protocols — Where User Authentication Protocols May Go*. Springer-Verlag, Berlin Germany, 2004.

[R+07] D. Recordon et al. OpenID Authentication 2.0 - Final, December 2007.

[Rad07] R. Radhakrishnan. *Identity and Security*. Futuretext, 2007.

[Reh07] R. U. Rehman. *OpenId*. Conformix Books, 2007.

[RFC1630] T. Berners-Lee. Universal Resource Identifiers in WWW: A Unifying Syntax for the Expression of Names and Addresses of Objects on the Network as used in the World-Wide Web. RFC 1630 (Informational), Internet Engineering Task Force, June 1994.

[RFC2195] J. Klensin, R. Catoe, and P. Krumviede. IMAP/POP AUTHorize Extension for Simple Challenge/Response. RFC 2195 (Proposed Standard), Internet Engineering Task Force, September 1997.

[RFC2608] E. Guttman, C. Perkins, J. Veizades, and M. Day. Service Location Protocol, Version 2. RFC 2608 (Proposed Standard), Internet Engineering Task Force, June 1999. Updated by RFC 3224.

[RFC2617] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, and L. Stewart. HTTP Authentication: Basic and Digest Access Authentication. RFC 2617 (Draft Standard), Internet Engineering Task Force, June 1999.

[RFC2865] C. Rigney, S. Willens, A. Rubens, and W. Simpson. Remote Authentication Dial In User Service (RADIUS). RFC 2865 (Draft Standard), Internet Engineering Task Force, June 2000. Updated by RFCs 2868, 3575, 5080.

[RFC2965] D. Kristol and L. Montulli. HTTP State Management Mechanism. RFC 2965 (Historic), Internet Engineering Task Force, October 2000. Obsoleted by RFC 6265.

[RFC3501] M. Crispin. INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1. RFC 3501 (Proposed Standard), Internet Engineering Task Force, March 2003. Updated by RFCs 4466, 4469, 4551, 5032, 5182, 5738, 6186.

[RFC3588] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, and J. Arkko. Diameter Base Protocol. RFC 3588 (Proposed Standard), Internet Engineering Task Force, September 2003. Updated by RFCs 5729, 5719, 6408.

[RFC3748] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz. Extensible Authentication Protocol (EAP). RFC 3748 (Proposed Standard), Internet Engineering Task Force, June 2004. Updated by RFC 5247.

[RFC4120] C. Neuman, T. Yu, S. Hartman, and K. Raeburn. The Kerberos Network Authentication Service (V5). RFC 4120 (Proposed Standard), Internet Engineering Task Force, July 2005. Updated by RFCs 4537, 5021, 5896, 6111, 6112, 6113.

[RFC4150] R. Dietz and R. Cole. Transport Performance Metrics MIB. RFC 4150 (Proposed Standard), Internet Engineering Task Force, August 2005.

[RFC4252] T. Ylonen and C. Lonvick. The Secure Shell (SSH) Authentication Protocol. RFC 4252 (Proposed Standard), Internet Engineering Task Force, January 2006.

[RFC4301] S. Kent and K. Seo. Security Architecture for the Internet Protocol. RFC 4301 (Proposed Standard), Internet Engineering Task Force, December 2005. Updated by RFC 6040.

[RFC4306] C. Kaufman. Internet Key Exchange (IKEv2) Protocol. RFC 4306 (Proposed Standard), Internet Engineering Task Force, December 2005. Obsoleted by RFC 5996, updated by RFC 5282.

[RFC4422] A. Melnikov and K. Zeilenga. Simple Authentication and Security Layer (SASL). RFC 4422 (Proposed Standard), Internet Engineering Task Force, June 2006.

[RFC4511] J. Sermersheim. Lightweight Directory Access Protocol (LDAP): The Protocol. RFC 4511 (Proposed Standard), Internet Engineering Task Force, June 2006.

[RFC4648] S. Josefsson. The Base16, Base32, and Base64 Data Encodings. RFC 4648 (Proposed Standard), Internet Engineering Task Force, October 2006.

[RFC4918] L. Dusseault. HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV). RFC 4918 (Proposed Standard), Internet Engineering Task Force, June 2007. Updated by RFC 5689.

[RFC4949] R. Shirey. Internet Security Glossary, Version 2. RFC 4949 (Informational), Internet Engineering Task Force, August 2007.

[RFC5209] P. Sangster, H. Khosravi, M. Mani, K. Narayan, and J. Tardo. Network Endpoint Assessment (NEA): Overview and Requirements. RFC 5209 (Informational), Internet Engineering Task Force, June 2008.

[RFC5246] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246 (Proposed Standard), Internet Engineering Task Force, August 2008. Updated by RFCs 5746, 5878, 6176.

[RFC5321] J. Klensin. Simple Mail Transfer Protocol. RFC 5321 (Draft Standard), Internet Engineering Task Force, October 2008.

[RFC6120] P. Saint-Andre. Extensible Messaging and Presence Protocol (XMPP): Core. RFC 6120 (Proposed Standard), Internet Engineering Task Force, March 2011.

[RFC959] J. Postel and J. Reynolds. File Transfer Protocol. RFC 959 (Standard), Internet Engineering Task Force, October 1985. Updated by RFCs 2228, 2640, 2773, 3659, 5797.

[RG10] D. Recordon and B. Goldman. OAuth 2.0 Device Profile – IETF Internet Draft draft-recordon-oauth-v2-device00.txt. IETF, 07 2010.

[RGS+10] H. Rajasekaran, J. Girao, H. Santos, M. Lischka, N. Gruschka, M. Barisch, et al. SWIFT – White Paper – SWIFT Identity Architecture. http://www.ist-swift.org/component/option,com_docman/task,doc_download/gid,23/Itemid,37/, January 2010.

[RH03] S. Rafaeli and D. Hutchison. A Survey of Key Management for Secure Group Communication. *ACM Comput. Surv.*, 35:309–329, September 2003.

[Ric92] A. M. Ricciardi. The Group Membership Problem in Asynchronous Systems. Technical report, Cornell University, 1992.

[RLGPC+99] A. Reyes-Lecuona, E. Gonzalez-Parada, E. Casilari, C. Casasola, and A. Diaz-Estrella. A Page-oriented WWW Traffic Model for Wireless System Simulations. In *Proceedings of 16th International Teletraffic Congress (ITC), Edinburgh*, 1999.

[RM99] S. Raman and S. McCanne. A Model, Analysis, and Protocol Framework for Soft State-based Communication. *SIGCOMM Comput. Commun. Rev.*, 29:15–25, August 1999.

[RSA78] R. L. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public-key Cryptosystems. *Commun. ACM*, 21:120–126, February 1978.

[RSA10] RSA SecurID Two-factor Authentication – RSA Solution Brief. http://www.rsa.com/products/securid/sb/10695_SIDTFA_SB_0210.pdf, 2010. [Retrieved: March, 28th 2012].

[S+05] T. Scavo et al. Shibboleth Architecture Technical Overview, Working Draft 02, June 2005.

[S+06] M. Schumacher et al. *Security Patterns – Integrating Security and Systems Engineering*. Wiley, 2006.

[S+08] A. Sarma et al. Virtual Identity Framework for Telecom Infrastructures. In *Wireless Personal Communications*, Netherlands, February 2008. Springer.

[S+10] P. Scholta et al. SWIFT Deliverable 505 – Refined SWIFT Scenarios, Use Cases and Business Modells. http://www.ist-swift.org/, March 2010.

[SAN03] S. Sovio, N. Asokan, and K. Nyberg. Defining Authorization Domains Using Virtual Devices. In *SAINT-W '03: Proceedings of the 2003 Symposium on Applications and the Internet Workshops (SAINT'03 Workshops)*, page 331, Washington, DC, USA, 2003. IEEE Computer Society.

[SCFY96] R. Sandhu, E. Coyne, H. Feinstein, and C. Youman. Role-based access control models. *Computer*, 29:38 –47, 1996.

[Sch96] B. Schneier. *Applied Cryptography*. Wiley, 1996.

[Sch99] B. Schneier. Attack Trees. *Dr Dobb's Journal*, 24(12):1, December 1999.

[SCLGS08] M. Sánchez, O. Cánovas, G. López, and A. F. Gómez-Skarmeta. Levels of Assurance and Reauthentication in Federated Environments. In *Proceedings of the 5th European PKI workshop on Public Key Infrastructure: Theory and Practice*, EuroPKI '08, pages 89–103, Berlin, Heidelberg, 2008. Springer-Verlag.

[SDV+07] M. Schuster, A. Domene, R. Vaidya, S. Arbanowski, S. M. Kim, J. W. Lee, and H. Lim. Virtual Device Composition. In *Proc. Eighth Int. Symp. Autonomous Decentralized Systems ISADS '07*, pages 270–278, 2007.

[Sea09] R. C. Seacord. *The CERT C Secure Coding Standard*. Addison-Wesley Professional, 2009.

[Sei10] C. Seifert. Analyzing Malicious SSH Login Attempts. http://www.symantec.com/connect/articles/analyzing-malicious-ssh-login-attempts, November 2010. [Retrieved: March, 28th 2012].

[Ser10] G. Serrao. Network Access Control (NAC): An Open Source Analysis of Architectures and Requirements. In *Security Technology (ICCST), 2010 IEEE International Carnahan Conference on*, 2010.

[SGSC+08] B. Stone-Gross, D. Sigal, R. Cohn, J. Morse, K. Almeroth, and C. Kruegel. VeriKey: A Dynamic Certificate Verification System for Public Key Exchanges. In *Proceedings of the 5th international conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, DIMVA '08, pages 44–63, Berlin, Heidelberg, 2008. Springer-Verlag.

[SH00] C. Schwingenschlögl and A. Heigl. Development of a Service Discovery Architecture for the Bluetooth Radio System. In *In Proceedings EUNICE 2000, Sixth EUNICE Open European Summer School*, 2000.

[Shi11a] REFEDS Federation Survey. https://refeds.terena.org/index.php/Federations, September 2011. [Retrieved: March, 28th 2012].

[Shi11b] Shibboleth IdM Software Version 2.2. http://shibboleth.internet2.edu/, 2011. [Retrieved, March 28th, 2012].

[Sid08] A. Siddhartha. National e-ID card schemes: A European overview. *Information Security Technical Report*, 13(2):46 – 53, 2008.

[Sip05] M. T. Siponen. Analysis of Modern IS Security Development Approaches: Towards the next Generation of Social and Adaptable ISS Methods. *Inf. Organ.*, 15:339–375, October 2005.

[SJZvD04] R. Sailer, T. Jaeger, X. Zhang, and L. van Doorn. Attestation-based Policy Enforcement for Remote Access. In *CCS '04: Proceedings of the 11th ACM conference on Computer and communications security*, pages 308–317, New York, NY, USA, 2004. ACM.

[SKS10] P. Sovis, F. Kohlar, and J. Schwenk. Security Analysis of OpenID. In *Sicherheit*, pages 329–340, 2010.

[SNL05] C. Steel, R. Nagappan, and R. Lai. *Core Security Patterns: Best Practices and Strategies for J2EE, Web Services, and Identity Management*. Prentice Hall, 1 edition, 10 2005.

[SNS88] J. G. Steiner, C. Neuman, and J. I. Schiller. Kerberos: An Authentication Service for Open Network Systems. In *in Usenix Conference Proceedings*, pages 191–202, 1988.

[SO01] G. Sindre and A. L. Opdahl. Templates for Misuse Case Description. In *Proceedings of the 7th International Workshop on Requirements Engineering, Foundation for Software Quality (REFSQ'2001*, pages 4–5, 2001.

[SO05] G. Sindre and A. L. Opdahl. Eliciting Security Requirements with Misuse Cases. *Requir. Eng.*, 10:34–44, January 2005.

[Som10] I. Sommerville. *Software Engineering*. Addison-Wesley Longman, 2010.

[SP11] Y. Shen and S. Pearson. Privacy Enhancing Technologies: A Review. Technical report, Hewlett Packard Labs, 2011.

[SPS11] S. Spitz, M. Pramateftakis, and J. Swoboda. *Kryptographie und IT-Sicherheit*. Vieweg und Teubner, 2011.

[SRC84] J. H. Saltzer, D. P. Reed, and D. D. Clark. End-to-end Arguments in System Design. *ACM Trans. Comput. Syst.*, 2:277–288, 1984.

[SS75] J. Saltzer and M. Schroeder. The Protection of Information in Computer Systems. *Proceedings of the IEEE*, 63:1278 – 1308, 1975.

[SSTK07] R. Shacham, H. Schulzrinne, S. Thakolsri, and W. Kellerer. Ubiquitous Device Personalization and Use: The next Generation of IP Multimedia Communications. *ACM Trans. Multimedia Comput. Commun. Appl.*, 3(2):12, 2007.

[ST06] A. Schiper and S. Toueg. From Set Membership to Group Membership: A Separation of Concerns. *IEEE Trans. Dependable Secur. Comput.*, 3, 2006.

[STM10] L. M. Surhone, M. T. Timpledon, and S. F. Marseken, editors. *VirtualBox: X86 Virtualization, Sun Microsystems, Sun xVM, Comparison of Platform Virtual Machines, Hardware Virtualisation*. Betascript Publishing, 2010.

[SVA09] J. Suomalainen, J. Valkonen, and N. Asokan. Standards for Security Associations in Personal Networks – A Comparative Analysis. *Int. J. Secur. Netw.*, 4(1/2):87–100, 2009.

[Sve11] H. Sverdlove. The Most Vulnerable Smartphones of 2011 – Bit9 Report, November 2011.

[SW01] C. U. Smith and L. G. Williams. *Performance Solutions: A Practical Guide to Creating Responsive, Scalable Software*. Addison-Wesley Professional, 2001.

[SWE10] Software Security Specialized Knowledge Area, Software Engineering Body of Knowledge, 2010.

[Syn] SyncPlaces. http://www.andyhalford.com/syncplaces.

[sys10] OMG Systems Modeling Language (OMG SysML) Version 1.2, Object Management Group, June 2010.

[T$^+$] J. Tourzan et al. Liberty ID-WSF Web Services Framework Overview, Version 1.1. Liberty Alliance document, liberty-idwsf-overview-v1.1.pdf.

[TB08] J. A. Tuijn and D. Bijwaard. Spanning a Multimedia Session across Multiple Devices. *Bell Labs Technical Journal*, 12(4):179–193, 2008.

[Tel04] Telekommunikationsgesetz. http://www.gesetze-im-internet.de/tkg_2004/, 2004. [Retrieved: March, 28th 2012].

[TPL$^+$10] E. Torroglosa, A. Perez, G. Lopez, A. Gomez-Skarmeta, and O. Canovas. SWIFT: Advanced Identity Management. In *Communications and Networking in China (CHINACOM), 2010 5th International ICST Conference on*, 2010.

[TT07] E. Tsyrklevich and V. Tsyrklevich. Single Sign-On for the Internet. In *Proceedings of Blackhat*, 2007.

[TW10] A. S. Tanenbaum and D. J. Wetherall. *Computer Networks (5th Edition)*. Prentice Hall, 2010.

[UDD04] UDDI Version 3.0.2, OASIS, 2004.

[UML09] OMG Unified Modeling Language (OMG UML), Infrastructure, Version 2.2, Object Management Group, 2009.

[UML10] Unified Modeling Language 2.3 – Superstructure Specification, Object Management Group, June 2010.

[UPr] Microsoft U-Prove. https://connect.microsoft.com/site1188. [Retrieved: March, 28th 2012].

[Vel11] E. M. Velázquez. Security of Synchronized Passwords with Opera Link. http://my.opera.com/operalink/blog/2011/05/03/security-of-synchronized-passwords-with-opera-link, May 2011. [Retrieved: March, 28th 2012].

[VP08] C. N. Ververidis and G. C. Polyzos. Service Discovery for Mobile Ad Hoc Networks: A Survey of Issues and Techniques. *IEEE Communications Surveys & Tutorials*, 10:30–45, 2008.

[W+85] R. Wong et al. Polonius: An Identity Authentication System. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 101–107, 1985.

[w3c99] HTML 4.01 Specification - W3C Recommendation. http://www.w3.org/TR/html4, World Wide Web Consortium (W3C), 1999.

[w3c04] Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0 – W3C Recommendation, W3C, January 2004.

[W3C07] SOAP Version 1.2 – W3C Recommendation, W3C, 2007.

[W3C08] XML Signature Syntax and Processing (Second Edition) – W3C Recommendation, W3C, 2008.

[Wei99] M. Weiser. The Computer for the 21st Century. *SIGMOBILE Mob. Comput. Commun. Rev.*, 3:3–11, 1999.

[Win05] P. J. Windley. *Digital Identity*. O'Reilly, 2005.

[WS-02] Web Services Security (WS-Security), IBM, Microsoft and Verisign, April 2002.

[WS-09] WS-Trust 1.4 – OASIS specification, OASIS, 2009.

[WSF] Web Services Federation Language (WS-Federation), BEA Systems, BMC Software, CA, IBM, Layer 7 Technologies, Microsoft, Novell, Ping Identity, and VeriSign.

[WUR11] Wireless Universal Resource FiLe (WURFL) – http://wurfl.sourceforge.net/, 2011. [Retrieved: November, 12th 2011].

[WW10] J. Williams and D. Wichers. The Open Web Application Security Project Top 10 - 2010 – The Ten Most Critical Web Application Security Risks. http://owasptop10.googlecode.com/files/OWASP

[YOC08] K. Yang, S. Ou, and H.-H. Chen. On Effective Offloading Services for Resource-constrained Mobile Devices Running Heavier Mobile Internet Applications. *Communications Magazine, IEEE*, 46:56 –63, 2008.

[YW09] C. Yue and H. Wang. SessionMagnifier: A Simple Approach to Secure and Convenient Kiosk Browsing. In *Proceedings of the 11th international conference on Ubiquitous computing*, Ubicomp '09, pages 125–134, New York, NY, USA, 2009. ACM.

[Zha11] H. Zhang. More Flexible Mobile Device Management with Google Apps. http://googleenterprise.blogspot.com/2011/11/more-flexible-mobile-device-management.html, November 2011. [Retrieved: March, 28th 2012].

[ZMN05] F. Zhu, M. Mutka, and L. Ni. Service Discovery in Pervasive Computing Environments. *Pervasive Computing, IEEE*, 4(4):81–90, 2005.

Acknowledgments

The past six years at the Institute of Communication Networks and Computer Engineering (IKR) have been a very formative time for myself. I enjoyed the high standards and the continuous intention for improvement in combination with many degrees of freedom. By my work in national and international projects, the collaboration with students, and my teaching responsibilities, I had the privilege to meet and learn from numerous interesting persons. I cherish the time at the institute as a time, which was sometimes hard but often it was fun. Fun with colleagues, project partners, and students in meetings, on conferences and at social activities.

Foremost, I would like to thank Prof. Dr.-Ing. Dr. h.c. Paul Kühn for the opportunity to join the IKR. His confidence in me and his scientific advice made up the basis for this thesis. Likewise, I would like to thank his deputy Ulrich Gemkow for the freedom and chances I got at the IKR. Moreover, I would like to thank Prof. Dr.-Ing. Erwin Rathgeb and Prof. Dr.-Ing. Andreas Kirstädter for the evaluation of this thesis.

I am very grateful to all former and active colleagues of the IKR. The colleagues created an open-minded, friendly, congenial, social and cozy atmosphere at the institute that has been one of the success factors for this thesis. Thanks go to the members of the former service and security group and its successor the fixed network group: Christian Hauser, Jochen Kögel, Martin Neubauer, Andreas Reifert, Joachim Scharf, David Wagner, Sebastian Meier, Frank Feller, Mirja Kühlewind, Domenic Teuchert, and Ulrich Gemkow. I greatly appreciate the support of Sebastian Meier, Joachim Scharf, Frank Feller and Ulrich Gemkow by reviewing drafts of this thesis. I would also like to thank Christian Hauser for mentoring me during my first years. The time at the institute has been the nucleus for many good friendships. Thanks to my friends Joachim Scharf and Jochen Kögel for their support in difficult phases of this thesis.

The work with diploma, bachelor and master students supported the origin of this thesis by many fruitful discussions and early prototypes. Thanks to all those students that contributed to this thesis without knowing that.

The topic of this thesis originated in the Daidalos project and got concretized within the SWIFT project. It was a pleasure for me to meet many brilliant people during my project work in Daidalos and SWIFT, but also in G-Lab, Nexus, and DynFire. It is impossible to enumerate all the people I would like thank, not to mention the reasons. The following list of people does not have any order: Alfredo Matos, Joao Girao, Antonio Gomez Skarmeta, Alejandro Perez, Steffen Drüsedow, Korbinian Frank, Kajetan Dolinar, Ricardo Azevedo, Joeri Van Cleynenbreugel,

Peter Scholta, Wolfgang Steigerwald, Mario Lischka, Amardeo Sarma, Jürgen Jähnert, David Lutz, Paul Christ, Sebastian Kiesel, Rui Aguiar.

Last but not least, I would like to thank my family. Without the support of my parents and my sister it would not have been possible to complete my studies and this thesis. My mum and my dad did everything to enable me the way I took. Many thanks. Finally, I would like to emphasize the person that motivated me and encouraged me to finish this work: My beloved girlfriend Bettina.