# APPENDIX 2 – TECHNICAL REQUIREMENTS

Version 1.0

# Table of Contents



# I. Introduction

This document constitutes an integral part of the agreement for IoT services entered by Telia and Customer (the "**Agreement**") and includes technical requirements for use of the Services. Defined terms and abbreviations used in this document has the meaning set out in the Agreement, unless otherwise is set out herein.

The document summarizes Telia Company IoT requirements originating in 3GPP, GSMA TS.34 version 7.0, *IoT Device Connection Efficiency Guidelines*[1] [1], use-cases or features not covered yet within GSMA TS.34, as well as lessons learned gained from IoT commercial deployments. Requirements including the words "MUST" or "MUST NOT" in their descriptions are mandatory; all guidelines with "SHOULD" or "SHOULD NOT" are recommended for best performance.

While a single or small number of misbehaving Devices typically won't impact other users, a large group of Devices operating in a negative pattern consistently or at the same time may have a very adverse impact on network resources and therefore on other users.

Standard mobile Devices on the Telia Network (including, when applicable, the mobile networks of Telia's partners), as well as the applications running on these Devices, must operate in a manner consistent with the requirements specified in this document (the "**Requirements**"). Customer is solely responsible for ensuring such compliance. If Customer is an IoT Service Provider (a provider of a solution and/or product that integrates the Services into its own products as an inseparable component and as an integral part of a complete solution, provided to its customers), the Requirements must be passed on to End-Users, where applicable. Customer's or an End-User's failure to comply with the Requirements, causing disturbances in the Telia Networks may *inter alia* lead to the suspension of the Service(s).

This document is divided into four sections, reflecting how IoT Service Providers and IoT Service users are required to implement the Requirements and best-practice design in the different IoT Solution Layers (refer to Figure 1).

- Definitions
- IoT Service Provider Requirements (IoT Server Application and Network Service Enablement Layer)
- IoT Device Requirements, covering:
  - **Monolithic IoT Device Applications** (refer to Figure 1)

    These are purpose-built IoT Device Applications that handle both business-specific logic as well as various Service Enablement functionalities (*e.g.* Device management, security, discovery, registration, location, application framework, etc.). These are historically referred to as "M2M Devices" as they are developed to manage assets in a specific, decoupled M2M silo.
  - **Tiered IoT Device Applications** (refer to Figure 2)

    On the IoT Device, the IoT Device Application may not be monolithic but focus instead only on processes specific to the customer's business. It is developed on top of a separate component which provides numerous generic Service Enablement functionalities (*e.g.* Device management, security, discovery, registration, location, application framework, etc.). This middleware is called the IoT Embedded Service Layer, or Service Enablement Layer. By supporting a common Service Enablement Layer, multiple M2M Devices can be interconnected to the same Cloud Platform enabling an ecosystem (either proprietary or standardized – such as LWM2M or oneM2M) of suppliers and data. This next generation of hardware can be referred to as true "IoT Devices."
- Communication Module Requirements

[1] https://www.gsma.com/iot/gsma-iot-device-connection-efficiency-guidelines/

- Customer's Obligations and Requirements

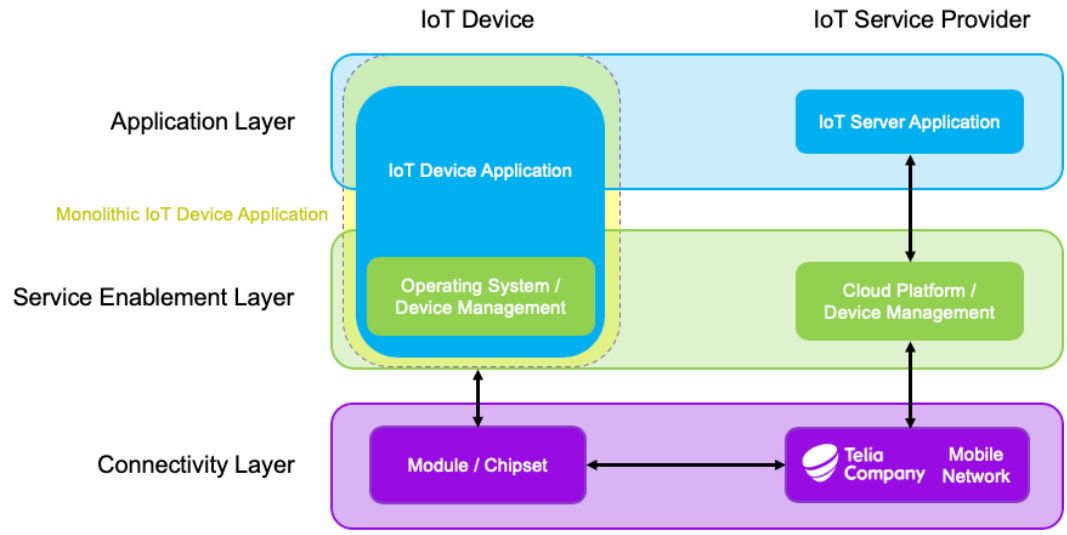


*Figure 1: IoT Solution Layers with **Monolithic IoT Device Application***

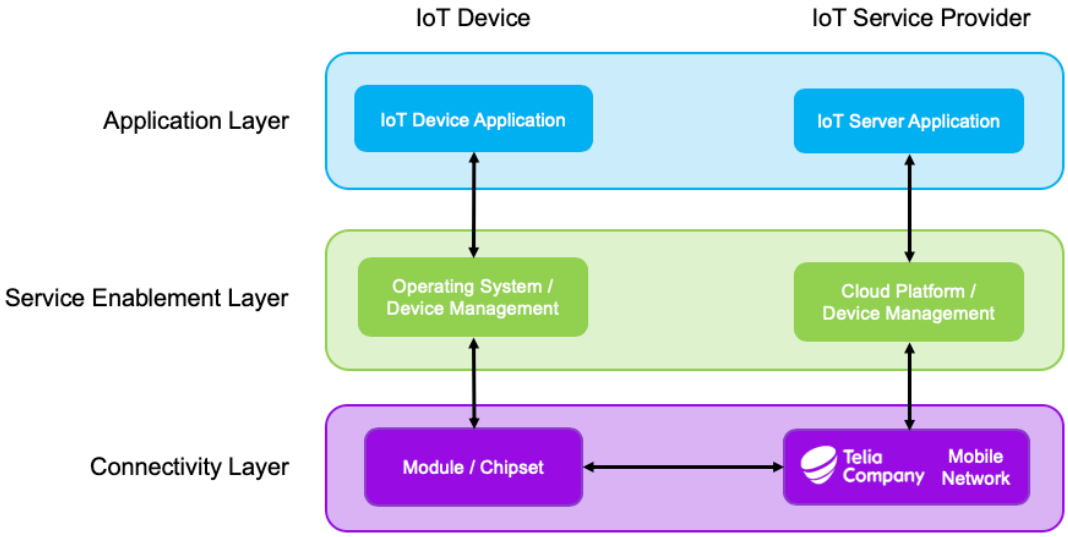


*Figure 2: IoT Solution Layers with **Tiered IoT Device Application***

## II. Definitions

"**Cloud Platform**" means the infrastructure used by the IoT Service Provider to host IoT Services, manage IoT Devices and exchange data with their IoT Devices over Telia Company's Connectivity Layer. It may host the IoT Server Application logic and includes Service Enablement functions. Generally, this is referred to as the "**IoT Service Platform**" in this document.

"**Communication Module**" means the communications component which provides wide area (2G, 3G, 4G, LPWA and 5G) radio connectivity, comprising of Communications Module Firmware, Radio Baseband Chipset and UICC.

"**Communication Module Firmware**" means the functionality within the Communications Module that provides an API to the IoT Device Application and controls the Radio Baseband Chipset.

"**Incremental Upgrade**" means an OTA Upgrade mechanism where only the delta between the new and the old firmware\software is downloaded and flashed.

"**IoT Device**" means sensors, actuators, or other deployed Machine to Machine (M2M) hardware exchanging data bidirectionally and managed by the IoT Service Provider over the Application, Service Enablement and Connectivity Layers.

"**IoT Device Application**" means the application logic running on the IoT Device's microcontroller (MCU) and exchanging data with the IoT Service Platform. It sends AT commands to the IoT Device's integrated communication module/chipset in order to access Telia Company's Connectivity Layer. The term "monolithic" refers to the handling of both business logic and Service Enablement functionalities within the same IoT Device Application, *i.e.* without a middleware. Alternatively, in a tiered device, the Service Enablement Layer is decoupled to provide Service Enablement separately from the IoT Device Application.

"**IoT Service**" means the communication between IoT Device and IoT Service Provider.

"**IoT Service Application**" means business application logic of the IoT Service which processes the data collected from Devices. The IoT Service Provider hosts its IoT Service Application on a server or Cloud Platform provided by Telia or another third party.

"**IoT Service Provider**" means a *Customer* offering IoT Services to End-Users or enterprises via Telia's Connectivity Layer (3GPP™ mobile networks), *i.e.* the Customer is not consuming the Service for itself, but is integrating it into its combined product/service. Note: Telia is not regarded as the "IoT Service Provider" in this context.

"**Mobile IoT**" is a terminology that is mainly used to refer to Low Power Wide-Area Networks (LPWA).

"**Over the Air (OTA) Upgrade**" means both Firmware Over the Air (FOTA) and Software Over the Air (SOTA) activities.

"**Pull**" means an operation through which IoT Service Providers or an IoT Service Application establishes a communication channel towards IoT Devices requesting them (pulling them) to send data to IoT Service Platform. This approach provides better control over critical data sessions like OTA Upgrades as an example.

"**Push**" means an operation through which IoT Device Application implemented on IoT Device(s) autonomously sends data (Pushes it) toward IoT Service Platform because it is programmed to do so without the need of any further instructions. This approach provides an efficient implementation when data needs to be sent periodically as an example.

"**Service Enablement**" means core service functions such as Device management, discovery, registration, group management, application and service management, communication management, data management, service charging and accounting, as well as subscription and notification, all of which are common needs across the wide spectrum of IoT solutions. These aspects are typically coordinated between IoT Devices and the IoT Service Platform on this logical layer. Server-side, service enablement

may be handled by an independent orchestrator or connector acting as an endpoint for all communication to/from the IoT Devices. Such a connector may be placed in front of one or several clouds hosting IoT Server Applications. The provider of the Service Enablement may be a Mobile (Virtual) Network Operator, or the developer of the IoT Device and Server Applications.

"**Mobile IoT**" is a terminology that is mainly used to refer to Low Power Wide-Area Networks (LPWA).

## III. IoT Service Provider Functional Requirements

A Customer being an IoT Service Provider must ensure that its products and services comply with the following Requirements:

**Avoidance of Synchronized Behavior**
Any IoT Service Platform or IoT Server Application which communicates with multiple IoT Devices MUST avoid synchronized behavior and employ a randomized pattern for accessing IoT Devices registered to the platform's domain. The triggering of data transmissions, the rebooting of the IoT Device hardware or sub-components (such as the communication module/chipset), or the issuing Device management commands (including, but not limited to (re-) registrations and firmware updates) MUST NOT be synchronized. Ref: TS.34_6.0_REQ_001.

**Implementation of Randomization**
Any IoT Service Platform or IoT Server Application which communicate to multiple IoT Devices MUST be designed to strive for asynchronous traffic behavior. When IoT Service Platform or IoT Server Application sends mobile terminated traffic to IoT Devices (data and/or SMS and/or voice) or pulls data from them, attempts MUST be randomized across IoT Devices of the same application both temporally and spatially.

In other words, if IoT Service Platform or IoT Server Application needs to send data to or pull it from IoT Devices at a fixed interval, attempts SHOULD be randomized within at least fifteen (15) percent of the transmission frequency. For example, if the frequency is 60min, IoT Server Application SHOULD use an approx. 10-minutes randomization window.

When it comes to spatial randomization, IoT Server Application MUST keep track of global cell identities (Cell-ID) where Devices are located/connected and apply spatial randomization across these cells whenever data needs to be Pushed or Pulled.

If IoT Service Platform or IoT Server Application fails to Push or Pull data, it MUST introduce a random exponentially increasing back-off timer before it attempts again. The purpose of this is mainly to prevent saturating paging and traffic channels on the radio interface.

**Behavior when IoT Service Platform or IoT Server Application Temporarily Offline**
If the IoT Service Platform or IoT Server Application are temporarily offline, they MUST NOT request all IoT Devices to synchronize all at once when they come back online. Ref: TS.34_4.0_REQ_011, TS.34_4.0_REQ_029.

**Triggering Devices only when Attached**
The IoT Service Platform or IoT Server Application MUST be aware of the IoT Device's state and only send "wake up" triggers whenever the IoT Device is known to be attached to the mobile network. Ref: TS.34_6.0_REQ_004.

**Behavior when IoT Device does not Respond to SMS Triggers**
If the IoT Service Platform or IoT Server Application uses SMS triggers to "wake up" IoT Devices, it MUST avoid sending multiple SMS triggers when no response is received within a certain period.

Communication over a 3GPP™ NB-IoT access bearer MUST NOT use SMS on the mobile network. Ref: TS.34_6.0_REQ_003.

**Behavior when SIM Subscription is Inactive**
If the SIM subscription associated with an IoT Device is to be placed in a temporarily inactive state (i.e. for a fixed period of time), the IoT Service Provider MUST first ensure that the IoT Device's communication module/chipset is temporarily disabled to restrict it from trying to register to the mobile network once the SIM is disabled. Ref: TS.34_6.0_REQ_002.

**Behavior when SIM Subscription is Permanently Disabled**
Before the SIM subscription associated with an IoT Device is to be placed in a permanently terminated state, the IoT Service Provider MUST first ensure that the IoT Device's communication module/chipset is permanently disabled to restrict it from trying to register to the mobile network once the SIM is disabled.

The IoT Service Provider SHOULD consider avoiding mechanisms for the permanent termination of IoT Devices that are not easily serviceable, as it may require manual intervention (*i.e.* a service call) to re-enable the IoT Devices. Ref: TS.34_6.0_REQ_002.

**Frequency and Prioritization of Data Transmissions**
Whenever there is a need to transmit data over the mobile network, the IoT Service Platform or IoT Server Application SHOULD classify the priority of each communication. The IoT Service Platform or IoT Server Application distinguishes between high-priority data requiring instant transmission, versus delay-tolerant or lower-priority data which can be aggregated and sent during non-peak hours. Ref: TS.34_4.0_REQ_018.

IoT Server Applications communicating with IoT Devices over 3GPP™ Mobile IoT access bearers, such as NB-IoT and LTE-M, MUST optimize their application reporting period to never exceed Telia's daily maximum number of messages as specified in Annex B.

**Data Aggregation, Compression and Transcoding**
The IoT Server Application MUST minimize the number of parallel mobile network connections and overall frequency of connections to IoT Devices over the mobile network. Data is aggregated by the IoT Server Application into an application report before being compressed and sent over the mobile network. Data transcoding and compression techniques are used, as per the IoT Service's intended Quality of Service, to reduce connection attempts and data volumes. Ref: TS.34_4.0_REQ_002, TS.34_4.2_REQ_002, TS.34_4.0_REQ_015.

**Frequency of Data Transmissions**
IoT Server Application using 3GPP™ Mobile IoT access bearers, such as NB-IoT and LTE-M, MUST optimize their application reporting period to never exceed Telia's and its partners' daily maximum number of messages as specified in Annex B.

**Monthly Data Volume**
IoT Server Application using 3GPP™ Mobile IoT access bearers, such as NB-IoT and LTE-M, MUST optimize their payload sizes to comply with Telia's and its partners' monthly volume limits as specified in Annex B.

**Device Initialization**
IoT Service Providers MUST NOT initialize significant numbers of IoT Devices (e.g. >100 units) communicating over 3GPP™ NB-IoT within one hour at the same location.

**Over the Air (OTA) Upgrade**
The IoT Service Platform or IoT Server Application MUST support OTA Upgrades for updating the embedded firmware\software of IoT Devices. A recommended file size not greater than 1 MB

SHOULD be used and randomized approach for file download MUST be implemented. Moreover, OTA Upgrades for IoT Devices communicating over 3GPP™ LPWA radio access technologies, such as NB-IoT and LTE-M, need to consider the following points:

- File compression SHOULD be used where possible
- OTA Upgrades MUST be performed over LTE-M when file size exceeds that predefined threshold of 1 MB
- Resume functionality MUST be supported
- Configurable number of downloads in a specific area SHOULD be supported
- Incremental upgrade MUST be supported
- OTA Upgrade over UDP MUST be used when IoT Devices are communicating over 3GPP™ NB-IoT radio access technology

## IV. IoT Device Functional Requirements

Customer must ensure that its Devices comply with the following Requirements:

**Avoidance of Synchronized Behavior**
The monolithic IoT Device Application MUST avoid synchronized behavior with other IoT Devices or events, employing a randomized pattern (*e.g.* over period ranging from a few minutes to several hours, or days) to request a mobile network connection over the Connectivity Layer. The triggering of data transmissions, the rebooting of the IoT Device hardware or sub-components (such as the communication module/chipset), or execution of device management commands (including, but not limited to (re-) registrations and firmware updates) MUST NOT be synchronized. In tiered IoT Devices, the embedded Service Enablement Layer MUST implement these requirements in the same way as for monolithic IoT Device Applications. Ref: TS.34_4.0_REQ_003, TS.34_4.2_REQ_003.

**Implementation of Randomization**
The monolithic IoT Device Application MUST be designed to strive for asynchronous traffic behaviour. When a Device connects to the network to access a service (data and/or SMS and/or voice), attempts MUST be randomized between Devices of the same application both temporally and spatially. In other words, if Devices need to upload data to the application server at a fixed interval, attempts SHOULD be randomized within at least fifteen (15) percent of the transmission frequency. For example, if the report frequency is 60min, the Device SHOULD use an approx. 10-minutes randomization window.

If an application fails to establish a packet data connection to the network or to its application server, then it MUST introduce a random exponentially increasing back-off timer before it attempts to connect again. The purpose of this is to prevent a population of Devices all attempting to establish a data connection at the exact same time. The re-attempt MUST be randomized across a group of Devices.

An example of where this may occur is if Customer's application server fails. If multiple Devices need to communicate with that server and they all make a repeat attempt to access the data service at the same time, they may all be unsuccessful, as the demand for resources may be exceeded, with the result that they then drop into a retry pattern with an effective self-created Denial of Service.

**Zero-Byte Session Principle**
The monolithic IoT Device Application MUST only establish data sessions for the purpose of sending and receiving data. An application which establishes a data session and then terminates it without sending data (a "zero-byte session") on a regular basis is not compliant with the Requirements.

**Use of “Always-on” Connectivity**
If the monolithic IoT Device Application sends data very frequently (*i.e.* inactivity periods shorter than two hours), it MUST use a persistent PDP/PDN connection with the mobile network instead of activating and deactivating said connectivity. In tiered IoT Devices, the embedded Service Enablement Layer MUST comply with this requirement. Ref: TS.34_4.0_REQ_001, TS.34_4.2_REQ_001.

**Handling of "Keep Alive" Messages on Home Network**
If the communication between the IoT Devices and mobile network is IP-based, it may require the use of TCP / UDP “keep alive” messages. In such cases, the monolithic IoT Device Application MUST automatically detect the server-specific timers and/or mobile network firewall timers, such TCP_IDLE value or UDP_IDLE value (NAT timers as defined by Telia for consumer APN, or by business enterprise for own-administered NAT, in the case of private APN), when using push services. This is achieved by increasing the polling interval dynamically until a mobile network timeout occurs, and then operating just below the timeout value.

Fixed polling intervals MUST NOT be used by the monolithic IoT Device Application, as polling interval values change between Telia and its partners and may dynamically adapt with mobile network loading. In tiered IoT Devices, the embedded Service Enablement Layer MUST implement these requirements in the same way as for monolithic IoT Device Applications.
This requirement does not apply to IoT Device Applications communicating with the IoT Server Application over 3GPP™ Mobile IoT access bearers, such as NB-IoT and LTE-M. Ref: TS.34_4.0_REQ_006, TS.34_4.2_REQ_006, TS.34_4.0_REQ_007, TS.34_4.2_REQ_007.

Monolithic IoT Device Applications communicating with the IoT Server Application over 3GPP™ Mobile IoT access bearers, such as NB-IoT and LTE-M, SHOULD NOT implement TCP / UDP “keep alive” messages on the home network. In tiered IoT Devices, the embedded Service Enablement Layer SHOULD implement this requirement in the same way as for monolithic IoT Device Applications.

**Handling of "Keep Alive" Messages on Roaming Network**
To minimize the risk of IP connectivity being lost when camping for extended periods (two hours, or more) on a roaming network – *i.e.* due to expiration of firewall timers, the monolithic IoT Device Application MUST periodically (period less than two hours) transmit small amounts of data to the IoT Service Platform via the visited network. A randomized timer triggers this mechanism, ensuring that the simultaneous transmission of data from many IoT Devices via the visited network is avoided. In tiered IoT Devices, the embedded Service Enablement Layer MUST comply with this requirement. This requirement does not apply to IoT Device Applications communicating with the IoT Server Application over 3GPP™ Mobile IoT access bearers, such as NB-IoT and LTE-M.

Monolithic IoT Device Applications communicating with the IoT Server Application over 3GPP™ Mobile IoT access bearers, such as NB-IoT and LTE-M, SHOULD NOT implement TCP / UDP “keep alive” messages on the roaming network. In tiered IoT Devices, the embedded Service Enablement Layer SHOULD implement this requirement in the same way as for monolithic IoT Device Applications.

This requirement also applies to the Global Connectivity offering.

**IoT Service Coordination**
If the monolithic IoT Device Application communicates with several IoT Server Applications using the same communication module/chipset, the IoT Device Application SHOULD coordinate the payload transmission of each IoT Service in a way which makes efficient use of the mobile network. In tiered IoT Devices, the embedded Service Enablement Layer SHOULD comply with this requirement.

**Data Aggregation, Compression and Transcoding**
A monolithic IoT Device Application MUST minimize the number of parallel mobile network connections and overall frequency of connections between the IoT Device and the mobile network. Data is aggregated by the IoT Device Application into an application report before being compressed and sent over the mobile

network. Data transcoding and compression techniques are used, as per the IoT Service's intended Quality of Service, to reduce connection attempts and data volumes. In tiered IoT Devices, the embedded Service Enablement Layer MUST comply with this requirement. Ref: TS.34_4.0_REQ_002, TS.34_4.2_REQ_002, TS.34_4.0_REQ_015, TS.34_4.2_REQ_015.

Monolithic IoT Devices Applications communicating with IoT Server Applications over 3GPP™ Mobile IoT access bearers, such as NB-IoT and LTE-M, MUST optimize their payload sizes to comply with Telia Company monthly or lifetime volume limits as specified in Annex B. In tiered IoT Devices, the embedded Service Enablement Layer MUST comply with this requirement.

**Monthly Data Volume - IoT Service Provider Defined Limit**
The IoT Device Application MUST monitor the volume of data it sends and receives over a set period. If the volume of data will soon exceed a maximum value defined by the IoT Service Provider or the (prepaid) SIM tariff, the IoT Device Application sends a report to the IoT Service Platform and stops the regular sending of data until the necessary period has expired. Ref: TS.34_4.0_REQ_013, TS.34_4.2_REQ_013.

**Prioritization and Frequency of Data Transmissions**
Whenever there is a need to transmit data over the mobile network, the monolithic IoT Device Application SHOULD classify the priority of each communication. The IoT Device Application distinguishes between high-priority data requiring instantaneous transmission, versus delay-tolerant or lower-priority data which can be aggregated and sent during non-peak hours. In tiered IoT Devices, the embedded Service Enablement Layer SHOULD consider the information communicated by the IoT Device Application about the importance and urgency of the data to deliver the IoT Service without negatively impacting the network. Ref: TS.34_4.0_REQ_018, TS.34_4.2_REQ_018, TS.34_4.1_REQ_003.

The monolithic IoT Device Application MUST monitor the number of network connections it attempts over a set period. If the number of connection attempts exceeds a maximum value set by the IoT Service Provider (refer to the Annex: Mobile IoT "No Harm to Network" Policies), the IoT Device Application sends a report to the IoT Service Platform and stops requesting mobile network connectivity until the necessary time period has expired. In tiered IoT Devices, the embedded Service Enablement Layer MUST comply with this requirement. Ref: TS.34_4.0_REQ_012, TS.34_4.2_REQ_012.

Monolithic IoT Devices Applications communicating with IoT Server Applications over 3GPP™ Mobile IoT access bearers, such as NB-IoT and LTE-M, MUST optimize their application reporting period to never exceed the IoT Service Provider's daily maximum number of messages (refer to Annex B: No Harm to Network Policies (LPWA)). In tiered IoT Devices, the embedded Service Enablement Layer MUST comply with this requirement.

**Avoidance of IoT Device "Last-Gasp" Messages**
The IoT Device Application MUST NOT trigger a significant number of IoT Devices (>100 units) to communicate over the 3GPP™ NB-IoT or LTE-M network simultaneously because of a system-wide failure situation (*e.g.* regional power outage).

**Off-Peak Communication**
If allowed by the IoT Service, the monolithic IoT Device Application SHOULD avoid concentrating communication over the mobile network during periods of high utilization. Periods to avoid will be communicated by Telia. In tiered IoT Devices, the embedded Service Enablement Layer SHOULD comply with this requirement. Ref: TS.34_4.0_REQ_016, TS.34_4.2_REQ_016.

**Localized Communication**
The monolithic IoT Device Application MUST minimize any geographical network loading problems. There MUST be no coordination of all IoT Devices in a given region of the IoT Service to undergo like-operations producing network loading, *e.g.* firmware updates. In tiered IoT Devices, the embedded Service Enablement Layer MUST comply with this requirement. Ref: TS.34_4.0_REQ_017, TS.34_4.2_REQ_017.

**Adaption to Mobile Network Capabilities, Data Speed and Latency**
The monolithic IoT Device Application MUST be capable of adapting to changes in mobile network feature capability and service exposure. Furthermore, it is designed to cope with variations in mobile network data speed and latency, considering the differences in available throughput, data speed and latency when switching between different 3GPP™ access bearers (*i.e.* 2G, 3G, LTE and Mobile IoT).

In tiered IoT Devices, the embedded Service Enablement Layer MUST comply with this requirement. the IoT Device Application MUST be able to retrieve mobile network speed and connection quality information from the Service Enablement Layer in order to request the appropriate quality of content from the IoT Service Platform. Ref: TS.34_4.0_REQ_008, TS.34_4.2_REQ_008, TS.34_4.0_REQ_009, TS.34_4.2_REQ_009.

If data speed and latency is critical to the IoT Service, the monolithic IoT Device Application SHOULD constantly monitor mobile network speed and connection quality to request the appropriate quality of content from the IoT Service Provider's infrastructure. In tiered IoT Devices, the embedded Service Enablement Layer SHOULD constantly monitor mobile network speed and connection quality in order to request the appropriate quality of content from the Cloud Platform. The IoT Device Application retrieves mobile network speed and connection quality information from the IoT Service Enablement Layer. Ref: TS.34_4.0_REQ_010, TS.34_4.2_REQ_010.

**Low Power Mode**
If the monolithic IoT Device Application does not need to exchange any data with the IoT Service Platform for a period greater than 24 hours, and the IoT Service can tolerate some latency, the IoT Device SHOULD implement a power-saving mode where the Device's communication module/chipset is effectively powered down between data transmissions. This will reduce the IoT Device's power consumption and reduce mobile network signaling. In tiered IoT Devices, the IoT Device Application SHOULD inform its embedded Service Enablement Layer that it does not need to exchange any data with the IoT Service Platform for a period greater than 24 hours, so that the latter can use this information in its interactions with the network. Ref: TS.34_4.0_REQ_020, TS.34_4.2_REQ_020, TS.34_4.1_REQ_004.

Monolithic IoT Device Applications communicating over 3GPP™ Mobile IoT access bearers, such as NB-IoT and LTE-M, MUST NOT power down their communication module/chipset. The 3GPP™ power saving features MUST be used instead, thus avoiding power-draining, system selection scanning procedures. PSM T3412 timer values shorter than 60 minutes are not supported. In tiered IoT Devices, the embedded Service Enablement Layer MUST implement this requirement in the same way as for monolithic IoT Device Applications.

**Behavior when IoT Service Platform is temporarily Unreachable or Offline**
If the monolithic IoT Service Platform is temporarily offline, the IoT Device Application MUST first diagnose if the communication issues to the server are caused by higher layer communications (TCP/IP, UDP, ATM…). Higher layers mechanisms MUST then try to re-establish the connection with the server. This is done by assessing (and if necessary, attempting to re-establish) connectivity in a stepwise approach, top-down. In tiered IoT Devices, the embedded Service Enablement Layer MUST comply with this requirement. Ref: TS.34_4.0_REQ_011, TS.34_4.2_REQ_011, TS.34_4.0_REQ_029.

The monolithic IoT Device Application MUST NOT frequently initiate an application-driven reboot of the communication module/chipset. The IoT Devices MUST retry connection requests to the IoT Service Platform with an exponentially increasing back-off period. In tiered IoT Devices, the embedded Service Enablement Layer MUST implement these requirements in the same way as for monolithic IoT Device Applications.

If the monolithic IoT Device detects that the IoT Service Platform is back online, it MUST employ a randomized timer to trigger communication requests to the mobile network. In tiered IoT Devices, the embedded Service Enablement Layer MUST comply with this requirement.

**Behavior when Coverage Lost (GPS, GLONASS, LAN, WAN)**
When GNSS (GPS, GLONASS, BeiDou, Galileo) coverage is lost, the monolithic IoT Device Application MUST NOT reboot the communication module/chipset supporting GNSS. The IoT Device Application SHOULD perform diagnostics, reboot the affected hardware element and send an alert to the IoT Server Application. In tiered IoT Devices, the embedded Service Enablement Layer MUST implement these requirements in the same way as for monolithic IoT Device Applications.

When LAN or WAN coverage is lost, the monolithic IoT Device MUST NOT reboot the IoT Device or communication module/chipset. The IoT Device Application MUST retry scanning to acquire mobile network connectivity with an exponentially increasing back-off period. In tiered IoT Devices, the embedded Service Enablement Layer MUST comply with this requirement.

**Behavior when Sensors / Actuators Malfunction**
When in-built sensors or actuators malfunction, the monolithic IoT Device Application MUST NOT reboot the communication module/chipset. The IoT Device Application SHOULD perform diagnostics, reboot the affected hardware element and send an alert to the IoT Server Application. In tiered IoT Devices, the embedded Service Enablement Layer MUST implement these requirements in the same way as for monolithic IoT Device Applications.

**Behavior when Sensor Alarms / Actuators Triggered**
When in-built sensors or actuators are triggered, the monolithic IoT Device Application MUST NOT reboot the communication module/chipset. The IoT Device Application SHOULD instead send an alert to the IoT Server Application. In tiered IoT Devices, the embedded Service Enablement Layer MUST implement these requirements in the same way as for monolithic IoT Device Applications.

**Behavior when Battery Power is Low or Power Failure Occurs**
The monolithic IoT Device Application SHOULD send a notification to the IoT Service Platform with relevant information when there is an unexpected power outage or battery problem. In tiered IoT Devices, the embedded Service Enablement Layer SHOULD comply with this requirement. Ref: TS.34_4.0_REQ_014, TS.34_4.2_REQ_014.

**Behavior when Device Memory Full**
When the monolithic IoT Device's memory is full, for example due to the amount of collected data or an unwanted memory leak, the IoT Device Application MUST NOT reboot the communication module/chipset. The IoT Device Application SHOULD perform diagnostics, reboot the affected hardware element and send an alert to the IoT Server Application. In tiered IoT Devices, the embedded Service Enablement Layer MUST implement this requirement in the same way as for monolithic IoT Device Applications.

**Behavior when Communication Requests Fail**
The monolithic IoT Device Application MUST always handle situations when communication requests fail in a way that does not harm the mobile network. The mobile network may reject communication requests from the IoT Device with a 3GPP™ error cause code (refer to GSMA TS.34). When the IoT Device Application detects that its requests are rejected, it MUST retry connection requests to the mobile network with an exponentially increasing back-off period. The IoT Device Application MUST NOT start an application-driven reboot of the communication module/chipset, attempting to ignore or override the mobile network's decision. In tiered IoT Devices, the embedded Service Enablement Layer MUST implement this requirement in the same way as for monolithic IoT Device Applications. Additionally, the IoT Device Application MUST always be prepared to handle situations when communication requests fail, when such failure is reported by the embedded Service Enablement Layer.

Communication requests from the monolithic IoT Device Application MUST NOT be retried indefinitely – all requests must eventually time-out and be abandoned by the IoT Device Application. In tiered IoT Devices, the embedded Service Enablement Layer MUST comply with this requirement. Ref: TS.34_4.0_REQ_011, TS.34_4.2_REQ_011, TS.34_4.1_REQ_002.

**Behavior when Device-Originated SMS are Barred**
When the monolithic IoT Device Application detects that its subscription for MO-SMS is barred by the mobile network, the IoT Device Application MUST retry connection requests to the mobile network with an exponentially increasing back-off period. The IoT Device Application MUST NOT start an application-driven reboot of the communication module/chipset. In tiered IoT Devices, the embedded Service Enablement Layer MUST implement this requirement in the same way as for monolithic IoT Device Applications.

**Reselection of Radio Access Technology Bearers**
For mass deployments of IoT Devices (>10,000 units within one country), if the monolithic IoT Device supports more than one family of access technology (for example 3GPP™, WLAN) the IoT Device Application MUST employ a randomized delay before switching to a different family of access technology. In tiered IoT Devices, the embedded Service Enablement Layer MUST comply with this requirement. Ref: TS.34_4.0_REQ_027, TS.34_4.2_REQ_027.

The monolithic IoT Device Application MUST implement a protection mechanism to prevent frequent “ping-pong” between these different technologies. This is done by limiting the frequency of reselection actions, with appropriate hysteresis mechanisms. In tiered IoT Devices, the embedded Service Enablement Layer MUST comply with this requirement. Ref: TS.34_4.0_REQ_026, TS.34_4.2_REQ_026.

**Handling Loss of Service on Roaming Network**
The monolithic IoT Device Application MUST always be prepared to recover lost end-to-end connectivity while camping on a roaming network. This is implemented with a top-down, staged recovery algorithm diagnosing each protocol layer. In case of failing to re-establish one layer, the algorithm initiates the recovery procedure on the following protocol level below. This may be done, for example, as follows:

- Step 1. Re-establishment of higher layer connectivity, e.g. VPN tunnels, SSH sessions, etc.,
- Step 2. Re-establishment of the PDN connectivity or PDP context,
- Step 3. Re-attach (data) to the network,
- Step 4. Re-triggering of a plain network selection,
- Step 5. Complete reboot of the Device.

All re-establishment procedures MUST be implemented in a reasonable way avoid excessive signaling to the mobile network. This MUST include usage of randomized triggers and incremental back-off retry mechanisms. Threshold and timer values may depend on the IoT Service’s requirements. In tiered IoT Devices, the embedded Service Enablement Layer MUST comply with this requirement. Ref: TS.34_4.0_REQ_029, TS.34_4.2_REQ_029.

This requirement also applies to the Global Connectivity offering.

**IPv4/v6 Dual Stack Support**
The monolithic IoT Device Application MUST support IPv4/v6 dual stack (PDN Type = IPv4v6) so that it can properly roam onto mobile networks having support for either IPv4 only or IPv6 only or dual stack only. In tiered IoT Devices, the embedded Service Enablement Layer MUST comply with this requirement.

**Device Reset to Factory Settings**
The monolithic IoT Device Application SHOULD support a “reset to factory settings” via remote and local connection. In tiered IoT Devices, the embedded Service Enablement Layer SHOULD comply with this requirement. Ref: TS.34_4.0_REQ_024, TS.34_4.2_REQ_024.

**Device Time Resynchronization**
The monolithic IoT Device Application SHOULD support time resynchronization via remote and local connection. In tiered IoT Devices, the embedded Service Enablement Layer SHOULD comply with this requirement. Ref: TS.34_4.0_REQ_025, TS.34_4.2_REQ_025.

**PLMN Selection**
The monolithic IoT Device Applications communicating over 3GPP™ Mobile IoT access bearers, such as NB-IoT and LTE-M, MUST avoid manual PLMN selection when possible. It's also highly recommended to avoid using AT+COPS command when roaming and use AT+CFUN instead.

**Avoidance of Spurious Behaviors**
The monolithic IoT Device Applications SHOULD be engineered to not exhibit spurious (not-authentic) behavior on the network. As an example, the Device SHOULD NOT try to establish a data session on the network and acquire a new IP address after a data session has been already established for the same Device. If the Device can handle two or more APNs, this applies per APN.

**Device Recovery**
The monolithic IoT Device Applications MUST implement a recovery method in case of unexpected behaviors such as a firmware failure or any similar event. An example of a recovery method is a watchdog timer that resets the Device when a specific condition is encountered. (Refer to Annex A: Device Recovery Guidelines)

**Over the Air (OTA) Upgrade**
The monolithic IoT Device Application MUST support OTA Upgrades to update the embedded firmware\software. A randomized approach for file download MUST be implemented. OTA Upgrades for IoT Devices communicating over 3GPP™ LPWA radio access technologies, such as NB-IoT and LTE-M, need to consider the following points:

- Resume functionality MUST be supported
- Incremental upgrades MUST be supported
- OTA Upgrades over UDP MUST be used when IoT Devices are communicating over 3GPP™ NB-IoT radio access technology

**Disabling Redundant Bands**
The monolithic IoT Device Applications communicating with the IoT Server Application over 3GPP™ LPWA Mobile IoT access bearers, such as NB-IoT and LTE-M, MUST disable redundant supported bands on the communication module that are not used by Telia. This helps to speed up PLMN and Cell Selection.

**Auto Corrected APN**
When communication module requests a wrong or a blank APN, Telia Network auto-corrects it and assigns it back to the Device. The monolithic IoT Device Application SHOULD accept the auto-corrected APN without trying to detach and attach the communication module again to get the requested APN assigned.

**Usage of the Embedded IP Stack**
The monolithic IoT Device Application communicating with the IoT Server Application over 3GPP™ LPWA Mobile IoT access bearers, such as NB-IoT and LTE-M, MUST use the embedded IP stack of the communication module. This guarantees a better compatible IoT implementation in terms of TCP timers and MTU size.

**Report Essential Health Information**
The monolithic IoT Device Application SHOULD support reporting some essential device and network health related information to IoT Service Platform. Some of the most important measurements would be

battery level, signal strength and quality (*e.g.* in LTE: RSRP, RSSI, RSRQ, and SINR\SNR) and enhanced coverage level (EC) for LPWA.

## V. Communication Module Requirements

Customer must ensure that its Devices comply with the following Requirements:

**GCF Certification**
All communication modules implemented in IoT Devices MUST be GCF certified. Non-certified communication modules will not be allowed in the Telia Network.

**RAT Support**
Communication module MUST support at least one of the following Radio Access Technologies (RATs): LTE, LTE-M or NB-IoT

**Multi-Operator Core Networks (MOCN)**
Communication module MUST follow MOCN network specification standards of the GERAN (GSM and GPRS) / WCDM (UMTS) / LTE / NB-IoT / LTE-M mobile network communication standards as specified in 3GPP.

**Power Down Requirements**
Communication module MUST adhere to power-down procedures and ensures a complete detach and IMS de-registration, if applicable, from the network when it's powered down.

**Firmware Over the Air (FOTA)**
Communication module MUST support FOTA. Firmware upgrades over LTE-M and NB-IoT SHOULD be performed as some sort of incremental FOTA where only the delta between the new and old firmware is downloaded and flashed.

**PLMN Selection and Idle Mode Procedures**
Communication module MUST implement network search and cell selection\re-selection procedures as specified in 3GPP TS 23.122.

**Data Over NAS (DoNAS)**
NB-IoT Communication modules MUST support DoNAS and attach to network using CP-CIoT.

**Extended T3412 (3GPP Rel-10)**
LTE-M and NB-IoT Communication modules SHOULD support Extended T3412 and indicate it in MS Network Feature Support IE. Supporting this timer helps in reducing signalling load and extending device battery life.

**EF-NASCONFIG (3GPP Rel-13)**
LTE-M and NB-IoT Communication modules SHOULD support EF-NASCONFIG file located on the SIM card under the path ADF(USIM)/6FE8. Communication modules MUST use the parameters in EF-NASCONFIG file to enable\disable NAS features as specified in 3GPP TS 24.368 in a way that takes precedence over NAS setting stored in device's non-volatile memory (NVM). If the file is not present, NAS settings stored in NVM MUST be used.

**Timer T3245 (3GPP Rel-10)**
LTE-M and NB-IoT Communication modules MUST support T3245 as specified in 3GPP TS 24.008 and 3GPP TS 24.301. When T3245 is enabled, communication module will clear FPLMN lists and sets USIM

to valid for non-EPS and EPS service upon the expiry of the timer. Devices MUST enable\disable T3245 timer according to the settings of *Timer_T3245_Behaviour* leaf in EF-NASCONFIG file.

**SIM Form Factor**
Communication module MUST support one of the standardized ETSI SIM form factors according to ETSI TS 102 221 (2FF, 3FF or 4FF) or ETSI TS 102 671 (MFF2).

**USIM Application Toolkit**
Communication module MUST support the USIM application toolkit commands.

**SIM-OTA**
Communication module MUST support SIM-OTA. The purpose is to secure that IoT Devices can relay incoming SIM-OTA messages in a correct way to the SIM according to the current standard.

**eSIM**
eUICC profile download functionality prerequisites that eSIM capable communication modules MUST support BIP over HTTPS and Short Message Service (SMS). SMS is used to trigger the Device to set up a BIP session for a profile download.

**LTE-M Requirements**
LTE-M communication modules MUST support the following features:

- Bands 3 and 20
- Combined attach for the support of SMS
- Intra and Inter Frequency Connected Mode Mobility
- Power Class 3 or 5 (Power Class 6 is not allowed)

**NB-IoT Requirements**
NB-IoT communication modules MUST support the following features:

- Bands 3 and 20
- Combined attach for the support of SMS
- Intra and Inter Frequency Cell Reselection
- Non-Anchor Carrier
- Release Assistance Indication (RAI)
- Power Class 3 or 5 (Power Class 6 is not allowed)

## VI. Customer's Obligations and Requirements

**System Related Requirements**

- If any authentication system resides in Customer's network and it is out of Telia's control, Customer MUST ensure that requests are handled and responded to in a timely manner. This MUST apply to systems receiving payload from any Device(s) as well.
- Customer MUST ensure that timeouts due to unacknowledged network requests and/or retransmissions are kept below one (1) percent of the total volume of traffic.
- Customer is not entitled to override the preferred roaming list (PLMN) on SIM Cards without Telia's prior written consent
- Customer is responsible for restarting any of its own equipment that was powered down or off as contemplated by this Agreement, for instance after a change of Service or troubleshooting.
- Customer MUST secure the following between the remote Customer Product and Customer Central System:

- appropriate testing (including all future releases) of the Device to comply with the requirements stated above, as well as
- correct communication functionality (voice and/or SMS and/or data).

**APN Configuration**
Customers are responsible for the configuration of the APN in their Device:

- APN MUST be configured in the Device by the Customer
- Customer MUST be able to re-configure the APN in the Device

**Testing of Devices and Applications**
Customer MUST at any time, upon Telia's request, provide Telia with Engineering Traceable Devices and applications with the relevant configuration for testing and troubleshooting purposes. Once the testing is over, Telia sends the Device back to Customer. Telia reserves the right to send a reference test device to site in cases where it becomes necessary to troubleshoot and establish a good network connectivity.

**Security Requirements**
It is the responsibility of Customer to ensure the following:

- Devices MUST conform to industrial security requirements and best practices, *e.g.* GSMA's IoT Security Guidelines and the IoT Project recommendations from OWASP.
- Industrial security requirements and best practices MUST be continuously monitored, and Devices MUST be adapted in a timely manner.
- Devices MUST be able to receive software and firmware upgrades in a safe manner as described in GSMA IoT Security Guidelines.
- Devices and their services MUST incorporate protection against impersonation attacks and replay attacks. Devices MUST be also able to support protection mechanisms against misuse, cloning, replacement or security credentials theft. Further protection against Denial of Service (DoS) attacks MUST be part of the solution.
- Devices that are directly accessible over the Internet MUST be designed with that in mind. Typically, this includes reduced exposure to only needed network services/ports and making sure that all access is authenticated. Furthermore, the input received MUST be validated to prevent buffer overrun attacks and the like.
- Customer MUST be able to nominate a point of contact in case of a security related question needs to be addressed by Telia.

Customer's failure to comply with Security Requirements listed above constitutes ground for suspension of Service.

**Regulatory and legal compliance**
Customer MUST ensure that its Devices always conform to current regulatory requirements of the countries in which the Device will be operational and to acquire any necessary certifications required by the regulatory bodies in the host countries, such as EMC, RED, SAR, FCC and other certifications.
Customer MUST also ensure that its Devices and applications comply with all applicable laws, including data privacy laws, advertising regulations (such as rules about cookies or "no-spam" laws) and the like.

**Over the Air (OTA) Upgrades**
Customer MUST contact Telia's IoT Support before performing any OTA Upgrades for more than 10,000 Devices. In the instances where an OTA Upgrade is required, a per customer approach and analysis will need to be carried out by Telia. This is to ensure a smooth upgrade for the Customer and a manageable impact to the Telia Network.

# VII. Annex A: Device Recovery Guidelines

This section sheds some light on how a recovery algorithm may be implemented in IoT Devices. It shouldn't be considered as a mandatory implementation requirement per se but more as a guideline and a recommendation. Customers may implement their recovery algorithms as they see fit for their IoT applications.

Download Device recovery guidelines here:
https://business.teliacompany.com/login/iot-connectivity-checklist

# VIII. Annex B: No Harm to Network Policies (LPWA)

**Maximum Number of Connection Requests**

- NB-IoT: 24 connection requests (Network Attach) / day / Device (*i.e.* on average once per hour).
- LTE-M: 144 connection requests (Network Attach) / day / Device (*i.e.* on average six times per hour).

**Minimum T3412 Timer**

Values shorter than 60mins will not be accepted.

**Maximum Number of Daily Messages**

- NB-IoT: 120 application messages / day / Device (*i.e.* on average 5 messages per hour); no volume restrictions per message.
- LTE-M: 720 application messages / day / Device (*i.e.* on average 30 messages per hour); no volume restrictions per message.

**Maximum Volume of Data per Single Device**

Please note: Maximum lifetime volume or pooling restrictions may be in place, limiting the Customer's average monthly data volume.

- NB-IoT: 1 MB average / month / Device; tariff-specific restrictions may occur.
- LTE-M: 500 MB average / month / Device; tariff-specific restrictions may occur.

# IX. Annex C: Document History

| Version | Date | Brief Description of Change |
|---|---|---|
| **1.0** | 20-Oct-2021 | |