ISG - No Harm to Network for IoT Device Applications

Version 2.6

I. Introduction

This document summarizes No Harm to Network requirements from Deutsche Telekom originating in GSMA TS.34 version 6.0, *IoT Device Connection Efficiency Guidelines*[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 "SHALL" or "SHALL NOT" in their descriptions are mandatory; all guidelines with "SHOULD" or "SHOULD NOT" are recommended.

This document is divided into three sections, reflecting how IoT Service Providers are required to implement No Harm to Network considerations and best-practice design in the different IoT Solution Layers (refer to Figure 1).

- Definitions
- IoT Service Provider Guidelines (IoT Server Application and Network Service Enablement Layer)
- IoT Device Guidelines, 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.

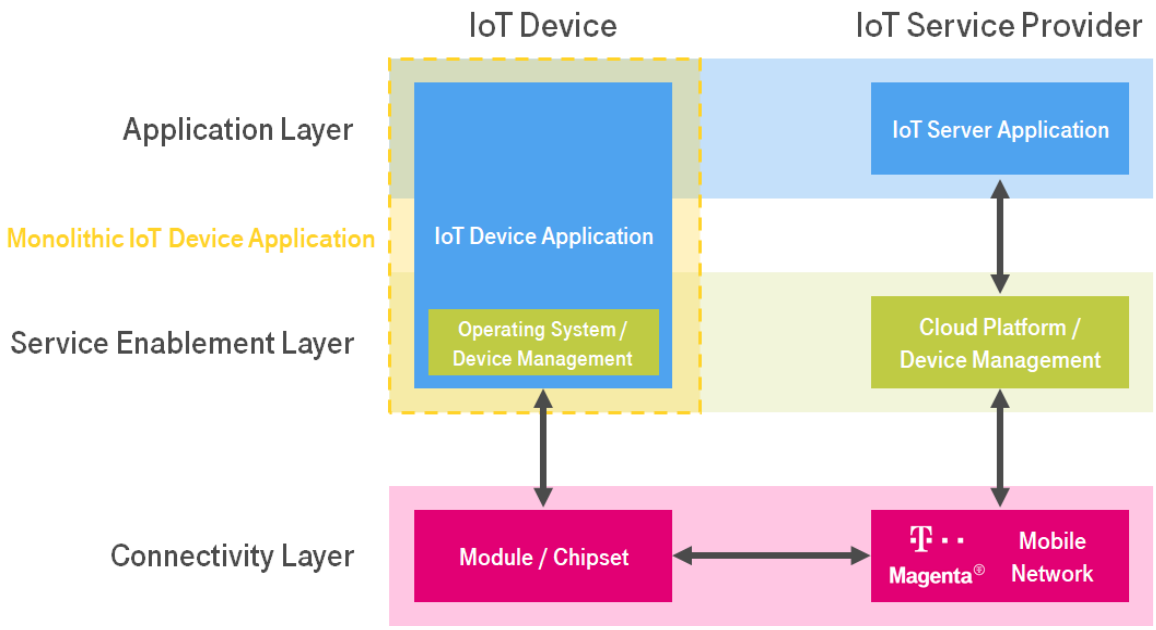

Figure 1: IoT Solution Layers with **Monolithic IoT Device Application**

  - **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

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

Enablement Layer, multiple M2M Devices can be interconnected to the same Cloud Platform enabling an ecosystem (either proprietary or standardized – such as oneM2M) of suppliers and data. This next generation of hardware can be referred to as true "IoT Devices."

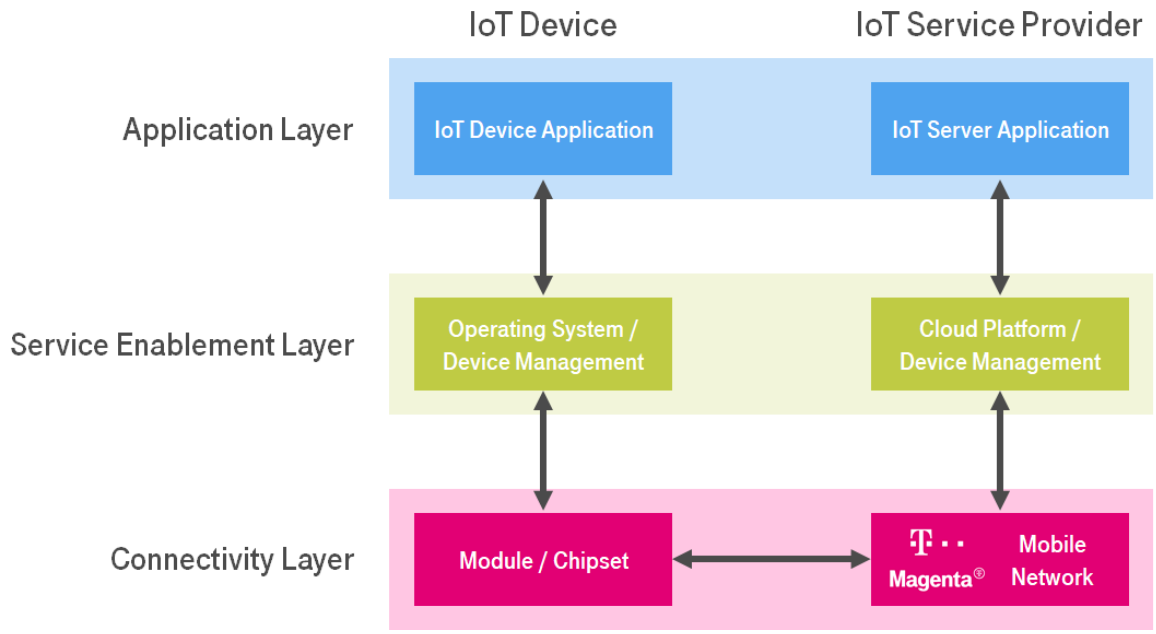

Figure 2: IoT Solution Layers with **Tiered IoT Device Application**

II. Definitions

IoT Service Provider

Companies offering IoT Services to end consumers or enterprises via Deutsche Telekom's Connectivity Layer (3GPP™ mobile networks). The IoT Service Provider hosts their IoT Server Application on their own application server or cloud platform. Alternatively, they may choose to integrate their application with Deutsche Telekom's IoT platform, which offers device management, storage, analytics, web apps, etc. The provider may be a Mobile (Virtual) Network Operator, the provider of Service Enablement, or the developer of the IoT Device Application and/or IoT Server Application.

IoT Server Application

Business application logic of the IoT Service which processes the data collected from assets. The IoT Service Provider hosts their IoT Server Application on a server or Cloud Platform provided by Deutsche Telekom or another third party.

Cloud Platform

Infrastructure used by the IoT Service Provider to host IoT Services, manage IoT Devices and exchange data with their IoT Devices over Deutsche Telekom'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.

Service Enablement

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.

IoT Device

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. The communication between IoT Device and IoT Service Provider is referred to as the IoT Service.

IoT Device Application

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 to access Deutsche Telekom'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.

III. IoT Service Provider Functional Requirements

Avoidance of Synchronized Behavior

Any IoT Service Platform or IoT Server Application which communicate to multiple IoT Devices SHALL 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) SHALL NOT be synchronized. Ref: TS.34_6.0_REQ_001.

Implementation of Randomization Timer

IoT Service Providers SHALL implement a randomization time of at least 40 seconds for the procedure of IoT Device Attach to the 3GPP™ NB-IoT network. This will reduce transmission time-outs and round-trip delays. Different randomization timer durations should be considered for optimizing performance in different Coverage Enhancement (CE) Levels.

Communication of Devices in Near Proximity

IoT Service Providers SHALL NOT trigger – neither manually nor from the IoT service platform - significant numbers of IoT Devices (e.g., a batch of >100 devices) to communicate over the 3GPP™ NB-IoT network within one hour in near proximity to each other (i.e., on the same network cell).

Number of Communicating Devices in Near Proximity

IoT Service Providers SHALL adhere to the following:

- No more than 60 devices in good coverage (outdoor installations) may initiate application traffic over the 3GPP™ NB-IoT network within one second in near proximity to each other (i.e., on the same network cell).
- No more than 40 devices in moderate or poor coverage (indoor or deep indoor installations) initiate application traffic over the 3GPP™ NB-IoT network within one second in near proximity to each other (i.e., on the same network cell).

Behavior when IoT Service Platform or IoT Server Application Temporarily Offline

If the IoT Service Platform or IoT Server Application are temporarily offline, they SHALL NOT request all IoT Devices to synchronize all at once when it comes 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 SHALL 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 SHALL avoid sending multiple SMS triggers when no response is received within a certain period.

Communication over a 3GPP™ NB-IoT access bearer SHALL NOT use SMS on Deutsche Telekom's 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 SHALL 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 SHALL 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 instantaneous 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™ 5G Massive IoT access bearers, such as NB-IoT and LTE-M, SHALL optimize their application reporting period to never exceed Deutsche Telekom affiliate tariff's daily maximum number of messages.

Data Aggregation, Compression and Transcoding

The IoT Server Application SHALL 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 Applications 5G Massive IoT access bearers (such as NB-IoT and LTE-M) SHALL optimize their application reporting period to never exceed Deutsche Telekom affiliate tariff's daily maximum number of messages.

Monthly Data Volume

IoT Server Application using 3GPP™ 5G Massive IoT access bearers (such as NB-IoT and LTE-M) SHALL optimize their payload sizes to comply with Deutsche Telekom affiliate monthly volume limits.

IV. IoT Device Functional Requirements

Avoidance of Synchronized Behavior

The monolithic IoT Device Application SHALL 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) SHALL NOT be synchronized. In tiered IoT Devices, the embedded Service Enablement Layer SHALL 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.

Use of "Always-on" Connectivity

If the monolithic IoT Device Application sends data very frequently (i.e., inactivity periods shorter than two hours), it SHALL 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 SHALL comply to 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 SHALL 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 Deutsche Telekom 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 SHALL NOT be used by the monolithic IoT Device Application, as polling interval values change between Deutsche Telekom affiliates, and may dynamically adapt with mobile network loading. In tiered IoT Devices, the embedded Service Enablement Layer SHALL 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™ 5G Massive 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™ 5G Massive 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 possibility 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 SHALL 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 SHALL comply to this requirement. This requirement does not apply to IoT Device Applications communicating with the IoT Server Application over 3GPP™ 5G Massive IoT access bearers, such as NB-IoT and LTE-M.

Monolithic IoT Device Applications communicating with the IoT Server Application over 3GPP™ 5G Massive 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.

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 to this requirement.

Data Aggregation, Compression and Transcoding

The monolithic IoT Device Application SHALL 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 SHALL comply to this requirement.

The monolithic IoT Device Application SHALL monitor the volume of data it sends and receives over a defined period. If the volume of data will soon exceed a maximum value defined by the IoT Service Provider (refer to the Annex: 5G Massive IoT "No Harm to Network" Policies), the IoT Device Application sends a report to the IoT Service Platform and stops the regular sending of data until the necessary time period has expired. In tiered IoT Devices, the embedded Service Enablement Layer SHOULD comply to 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 Device Applications communicating with IoT Server Applications over 3GPP™ 5G Massive IoT access bearers, such as NB-IoT and LTE-M, SHALL optimize their payload sizes to comply with Deutsche Telekom-tariff monthly or lifetime volume limits. In tiered IoT Devices, the embedded Service Enablement Layer SHALL comply to this requirement.

Monthly Data Volume - IoT Service Provider Defined Limit

The IoT Device Application SHALL 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 SHALL 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: 5G Massive 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 SHALL comply to 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™ 5G Massive IoT access bearers, such as NB-IoT and LTE-M, SHALL optimize their application reporting period to never exceed the IoT Service Provider's daily maximum number of messages (refer to the Annex: 5G Massive IoT "No Harm to Network" Policies). In tiered IoT Devices, the embedded Service Enablement Layer SHALL comply to this requirement.

Avoidance of IoT Device "Last-Gasp" Messages

The IoT Device Application SHALL NOT trigger a significant number of IoT Devices (e.g., >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 (i.e., transmissions during early morning hours are preferred). In tiered IoT Devices, the embedded Service Enablement Layer SHOULD comply to this requirement. Ref: TS.34_4.0_REQ_016, TS.34_4.2_REQ_016.

Localized Communication

The monolithic IoT Device Application SHALL minimize any geographical network loading problems. There SHALL be no localized coordination of all IoT Devices in any deployment 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 SHALL comply to 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 SHALL 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 5G Massive IoT). In tiered IoT Devices, the embedded Service Enablement Layer SHALL comply to this requirement; the IoT Device Application SHALL be able to retrieve mobile network speed and connection quality information from the Service Enablement Layer 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 to request the appropriate quality of content from the Cloud Platform. The IoT Device Application its 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™ 5G Massive IoT access bearers, such as NB-IoT and LTE-M, SHOULD NOT power down their communication module/chipset. The 3GPP™ power saving features SHOULD be used instead, thus avoiding power-draining, system selection scanning procedures. In tiered IoT Devices, the embedded Service Enablement Layer SHOULD 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 SHALL first diagnose if the communication issues to the server are caused by higher layer communications (TCP/IP, UDP, ATM...). Higher layers mechanisms SHALL 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 SHALL comply to 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 SHALL NOT frequently initiate an application-driven reboot of the communication module/chipset. The IoT Devices SHALL retry connection requests to the IoT Service Platform with an increasing back-off period. In tiered IoT Devices, the embedded Service Enablement Layer SHALL 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 SHALL employ a randomized timer to trigger communication requests to the mobile network. In tiered IoT Devices, the embedded Service Enablement Layer SHALL comply to this requirement.

Behavior when Coverage Lost (GPS, GLONASS, LAN, WAN)

When GNSS (GPS, GLONASS, BeiDou, Galileo) coverage is lost, the monolithic IoT Device Application SHALL 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 SHALL implement these requirements in the same way as for monolithic IoT Device Applications.

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

Behavior when Sensors / Actuators Malfunction

When in-built sensors or actuators malfunction, the monolithic IoT Device Application SHALL 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 SHALL 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 SHALL 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 SHALL 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 to 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 SHALL 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 SHALL implement this requirement in the same way as for monolithic IoT Device Applications.

Behavior when Communication Requests Fail

The monolithic IoT Device Application SHALL 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 SHALL retry connection requests to the mobile network with an increasing back-off period. The IoT Device Application SHALL 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 SHALL implement this requirement in the same way as for monolithic IoT Device Applications. Additionally, the IoT Device Application SHALL 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 SHALL 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 SHALL comply to 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 SHALL retry connection requests to the mobile network with an increasing back-off period. The IoT Device Application SHALL NOT start an application-driven reboot of the communication module/chipset. In tiered IoT Devices, the embedded Service Enablement Layer SHALL 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 (e.g., >10,000 units within the same mobile network), if the monolithic IoT Device supports more than one family of access technology (for example 3GPP™, WLAN) the IoT Device Application SHALL employ a randomized delay before switching to a different family of access technology. In tiered IoT Devices, the embedded Service Enablement Layer SHALL comply to this requirement. Ref: TS.34_4.0_REQ_027, TS.34_4.2_REQ_027.

The monolithic IoT Device Application SHALL 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 SHALL comply to 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 SHALL 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 SHALL be implemented in a reasonable way avoid excessive signaling to the network. This may 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 SHALL comply to this requirement. Ref: TS.34_4.0_REQ_029, TS.34_4.2_REQ_029.

IPv4/v6 Dual Stack Support

The monolithic IoT Device Application SHALL 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 SHALL comply to 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 to 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 to this requirement. Ref: TS.34_4.0_REQ_025, TS.34_4.2_REQ_025.

V. Annex: 5G Massive IoT "No Harm to Network" Policies

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).
- LTE (all Categories): **144** connection requests (Network Attach) / day / device (i.e., on average six times per hour).
- 5G-New Radio NSA: **144** connection requests (Network Attach) / day / device (i.e., on average six times per hour).

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.
- LTE (all Categories): **720** application messages / day / device (i.e., on average 30 messages per hour); no volume restrictions per message.
- 5G-New Radio NSA: **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. In case a multimode tariff is used, the figures below represent the maximum limit allowed per radio access technology.

- NB-IoT: **1 Mbyte*** average / month / device; tariff-specific restrictions may occur.
- LTE-M: **500 Mbyte*** average / month / device; tariff-specific restrictions may occur.
- LTE (all Categories): **No restrictions**, unless defined under the specific tariff conditions.
- 5G-New Radio NSA: **No restrictions**, unless defined under the specific tariff conditions.

* Data traffic for firmware updates excluded.

VI. Release History

Publication Date	Version	Author
17.08.2018	2.0	Miguel Rodriguez (ITS-IVA)
02.05.2019	2.1	Miguel Rodriguez (ITS-IVA)
17.03.2020	2.2	Miguel Rodriguez (ITS-IVA)
29.03.2021	2.3	Miguel Rodriguez (ITS-IVA)
30.06.2021	2.4	Miguel Rodriguez (ITS-IVA)
16.11.2021	2.5	Miguel Rodriguez (ITS-IVA)
07.04.2022	2.6	Dominik Thomas, Sanju Thoppil (DT IOT)