

# Richtlinien für IoT-Lösungen – Kommunikationseffizienz, Version 2.6

## I. Einleitung

Dieses Dokument fasst die Anforderungen zu „No Harm to Network“ [Vermeidung von Netzbeeinträchtigungen bzw. Kommunikationseffizienz] der Deutschen Telekom, die ihren Ursprung in den GSMA TS.34 v6.0, *IoT Device Connection Efficiency Guidelines*<sup>1</sup> [Richtlinien für die Verbindungseffizienz von IoT-Geräten] haben, Anwendungsfälle und Funktionen, die noch nicht vom GSMA TS.34 umfasst sind, sowie aus kommerziellen IoT-Einsätzen gewonnene Erfahrungen zusammen. Anforderungen/Bedingungen, die in ihren Beschreibungen die Worte „MUSS“ oder „DARF NICHT“ enthalten, sind obligatorisch; alle Richtlinien mit den Worten „SOLLTE“ oder „SOLLTE NICHT“ sind Empfehlungen.

Dieses Dokument ist in drei Abschnitte gegliedert, die aufzeigen, wie IoT-Serviceprovider Überlegungen zu „No Harm to Network“ und Best-Practice-Design in den verschiedenen Schichten von IoT-Lösungen (siehe Abbildung 1) umsetzen müssen.

- Begriffsbestimmungen
- Richtlinien für IoT-Serviceprovider (IoT-Server-Anwendung und Network Service Enablement Layer [Netzwerk-dienste-Ermöglichungs-Schicht])
- Richtlinien für IoT-Geräte, umfassen:
  - **Monolithische IoT-Geräteanwendungen** (siehe Abbildung 1)  
Dabei handelt es sich um speziell entwickelte IoT-Geräteanwendungen, die sowohl geschäftsspezifische Logik als auch verschiedene Service-Enablement- [Dienst-Ermöglichung] Funktionalitäten (z.B. Geräteverwaltung, Sicherheit, Discovery, Registrierung, Standort, Anwendungs-Framework, usw.). Diese werden historisch als "M2M-Geräte" bezeichnet, da sie zur Verwaltung von Anlagen in einem spezifischen, entkoppelten M2M-Silo entwickelt wurden.

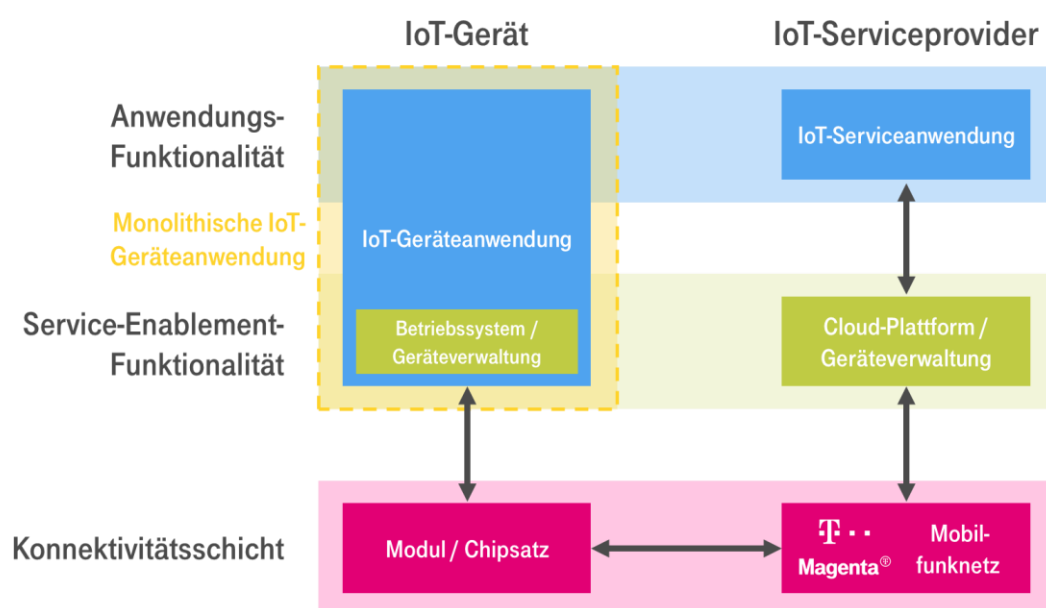


Figure 1: IoT-Lösungsschichten mit **monolithischer IoT-Geräteanwendung**

<sup>1</sup> <https://www.gsma.com/iot/gsma-iot-device-connection-efficiency-guidelines/>

- Mehrstufige IoT-Geräteanwendungen** (siehe Abbildung 2)
 

Auf dem IoT Gerät muss die IoT-Geräteanwendung nicht monolithisch sein, sondern kann sich stattdessen nur auf Prozesse konzentrieren, die für das Geschäft des Kunden spezifisch sind. Sie wird auf einer separaten Komponente entwickelt, die zahlreiche generische Service-Enablement- [Dienst-Ermöglichung] Funktionalitäten bietet (z.B. Geräteverwaltung, Sicherheit, Discovery, Registrierung, Standort, Anwendungs-Framework usw.). Diese Middleware wird als „IoT Embedded Service Layer“ oder „Service Enablement Layer“ bezeichnet. Durch die Unterstützung eines gemeinsamen Service Enablement Layer können mehrere M2M-Geräte mit derselben Cloud-Plattform verbunden werden, wodurch ein Ökosystem (entweder proprietär oder standardisiert - wie z.B. oneM2M) von Anbietern und Daten ermöglicht wird. Diese nächste Generation von Hardware kann als echtes "IoT Geräte" bezeichnet werden.

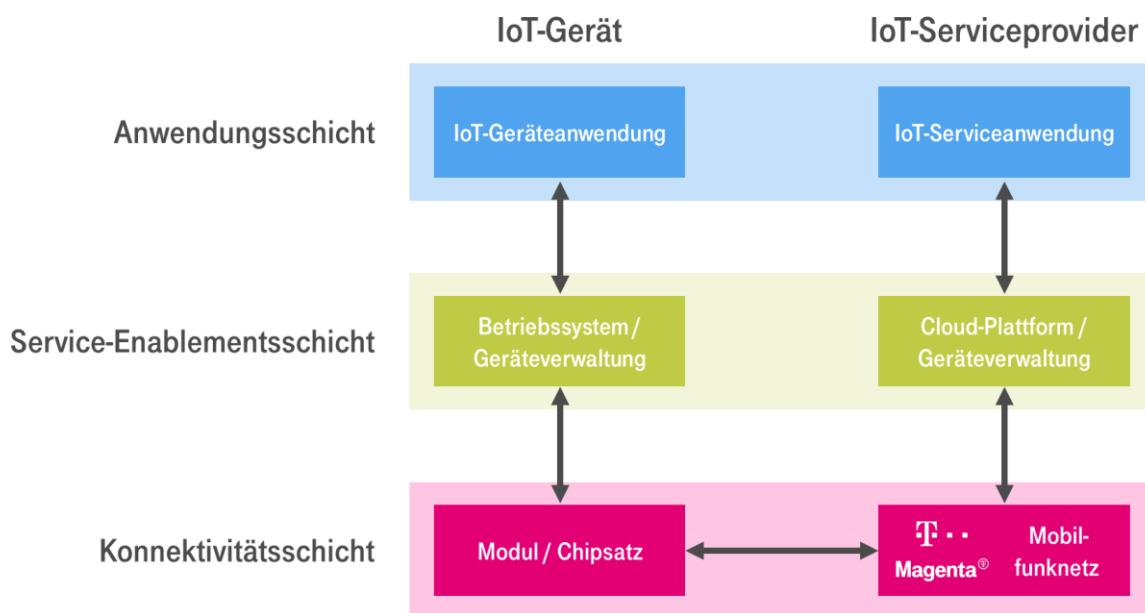


Abbildung 2: IoT-Lösungsschichten mit **mehrstufiger IoT-Geräteanwendung**

## II. Begriffsbestimmungen

---

### IoT-Serviceprovider

Unternehmen, die über die Konnektivitätsschicht der Deutschen Telekom (3GPP-Mobilfunknetze), IoT-Services für Endabnehmer oder Unternehmen anbieten. Der IoT-Serviceprovider hostet seine IoT-Server-Anwendung auf einem eigenen Anwendungsserver oder einer Cloud-Plattform. Alternativ kann er sich aber auch dafür entscheiden, seine Anwendung in die IoT-Plattform der Deutschen Telekom zu integrieren, welche Endgerätemanagement, Speicherplatz, Analytik, Web-Apps und weitere Funktionen bietet. Beim IoT-Serviceprovider kann es sich um einen (virtuellen) Mobilfunkanbieter, einen Anbieter von Service-Enablement-Funktionen oder einen Entwickler von IoT-Endgeräte- und/oder Serviceanwendungen handeln.

### IoT-Server-Anwendung

Geschäftsanwendungslogik des IoT-Service, der die von den Assets gesammelten Daten verarbeitet. Der IoT-Serviceprovider hostet seine IoT-Server-Anwendung auf einem Server oder einer Cloud-Plattform, die von der Deutschen Telekom oder einem anderen Drittanbieter bereitgestellt wird.

## Cloud-Plattform

Infrastruktur, die vom IoT-Serviceprovider verwendet wird, um IoT-Services zu hosten, IoT-Geräte zu verwalten und Daten mit ihren IoT-Geräten über die Konnektivitätsschicht der Deutschen Telekom auszutauschen. Sie kann die IoT-Server-Anwendungslogik hosten und beinhaltet Service-Enablement-Funktionen. Im Allgemeinen wird dies in diesem Dokument als die "IoT-Service-Plattform" bezeichnet.

## Service Enablement

Zentrale Servicenfunktionen wie Gerätemanagement, Discovery, Registrierung, Gruppenmanagement, Anwendungs- und Servicemanagement, Kommunikationsmanagement, Datenmanagement, Leistungsverrechnung, sowie Abonnement und Benachrichtigung sind allesamt gemeinsame Anforderungen in dem breiten Spektrum von IoT-Lösungen. Diese Aspekte werden in der Regel zwischen IoT-Geräten und der IoT-Service-Plattform auf dieser logischen Schicht koordiniert. Serverseitig kann das Service Enablement durch einen unabhängigen Orchestrator oder Konnektor erfolgen, der als Endpunkt für die gesamte Kommunikation zu/von den IoT-Geräten fungiert. Ein solcher Konnektor kann vor einer oder mehreren Clouds platziert werden, die IoT-Server-Anwendungen hosten. Der Anbieter des Service Enablements kann ein Mobilfunkprovider oder der Entwickler des IoT-Gerätes und der IoT-Serveranwendungen sein.

## IoT-Gerät

Sensoren, Aktoren und sonstige eingesetzte Machine-to-Machine (M2M)-Hardware, die Daten wechselseitig austauschen und vom IoT-Serviceprovider über die Anwendung, das Service-Enablement und Konnektivitätsschichten verwaltet werden. Die Kommunikation zwischen dem IoT-Gerät und dem IoT-Serviceprovider wird als IoT-Service bezeichnet.

## IoT-Geräteanwendung

Die Anwendungslogik, die auf dem Mikrocontroller (MCU) des IoT-Geräts läuft und Daten mit der IoT-Serviceplattform austauscht. Sie sendet AT-Kommandos auf das integrierte Kommunikationsmodul/den Chipsatz des IoT-Geräts, um auf die Konnektivitätsschicht der Deutschen Telekom zugreifen zu können. Der Begriff "monolithisch" bezieht sich auf die Handhabung sowohl der Geschäftslogik als auch der Service-Enablement-Funktionalitäten innerhalb derselben IoT-Geräteanwendung, d.h. ohne Middleware. Alternativ dazu wird in einem mehrstufigen Gerät die Service-Enablement-Schicht entkoppelt, um Service Enablement getrennt von der IoT-Geräteanwendung bereitzustellen.

## III. Richtlinien für IoT-Serviceprovider

---

### Vermeidung von synchronisiertem Verhalten

Jede IoT-Service-Plattform oder IoT-Server-Anwendung, die mit mehreren IoT-Geräten kommuniziert, **MUSS** synchronisiertes Verhalten vermeiden und für den Zugriff auf IoT-Geräte, die für die Domain der Plattform registriert sind, ein zufälliges Muster verwenden. Das Auslösen von Datenübertragungen, das Neustarten der Hardware des IoT-Geräts oder seiner Subkomponenten (wie beispielsweise das Kommunikationsmodul/der Chipsatz) und die Erteilung von Geräteverwaltungsbefehlen (insbesondere (Neu-)Anmeldungen und Firmware-Updates) **DÜRFEN NICHT** synchronisiert werden. Ref.: [TS.34\\_6.0\\_REQ\\_001](#).

### Implementierung einer Randomisierungszeit

Der IoT-Serviceprovider **SOLLTE** eine Randomisierungszeit von mindestens 40 Sekunden für das Einbuchen in das 3GPP™ NB-IoT-Netzwerk implementieren. Dadurch werden Übertragungszeit-überschreitungen und Roundtrip-

Verzögerungen reduziert. Zur Optimierung der Leistung in verschiedenen CE-Stufen (Coverage Enhancement) sollten unterschiedliche Randomisierungszeiten berücksichtigt werden.

### **Kommunikation von Geräten in unmittelbarer Nähe zueinander**

IoT-Serviceprovider **DÜRFEN NICHT** – weder manuell noch aus der IoT Plattform – veranlassen, dass eine signifikante Anzahl von IoT-Geräten (z. B. >100 Einheiten), die sich in unmittelbarer Nähe zueinander (in derselben Netzwerkzelle) befinden, innerhalb einer Stunde über das 3GPP™ NB-IoT-Netzwerk kommunizieren.

### **Anzahl kommunizierender Geräte in unmittelbarer Nähe zueinander**

Der IoT-Serviceprovider **MUSS** Folgendes beachten:

- Maximal 60 IoT-Geräte die sich in gute Abdeckung (draußen) und in unmittelbarer Nähe zueinander (in derselben Netzwerkzelle) befinden, **DÜRFEN** innerhalb einer Sekunde über das 3GPP™ NB-IoT-Netzwerk Anwendungsdaten versenden.
- Maximal 40 IoT-Geräte, die sich in moderate oder schwache Abdeckung (Installationen im Innen- oder tiefen Innenbereich und in unmittelbarer Nähe zueinander (in derselben Netzwerkzelle)) befinden, **DÜRFEN** sich innerhalb einer Sekunde über das 3GPP™ NB-IoT-Netzwerk Anwendungsdaten versenden.

### **Verhalten, wenn die IoT-Service Plattform oder IoT-Server-Anwendung vorübergehend offline ist**

Falls die IoT-Service-Plattform bzw. IoT-Server-Anwendung vorübergehend offline ist, **DARF** sie, wenn sie wieder online ist, **NICHT** alle IoT-Geräte auffordern, sich auf einmal zu synchronisieren. Ref.: [TS.34\\_4.0\\_REQ\\_011](#), [TS.34\\_4.0\\_REQ\\_029](#).

### **Auslösen von Geräten nur bei Verbindung**

Die IoT-Service-Plattform bzw. IoT-Server-Anwendung **MUSS** den Status des IoT-Geräts kennen, und **DARF** nur dann „Aufweck“-Signale an das IoT-Gerät schicken, wenn feststeht, dass das IoT-Gerät mit dem Mobilfunknetz verbunden ist. Ref.: [TS.34\\_4.0\\_REQ\\_004](#).

### **Verhalten, wenn das IoT-Gerät nicht auf SMS-Trigger antwortet**

Falls die IoT-Service-Plattform bzw. IoT-Server-Anwendung SMS-Trigger verwendet, um IoT-Geräte „aufzuwecken“, **DARF** die IoT-Serviceplattform **NICHT** mehrere SMS-Trigger schicken, wenn innerhalb eines bestimmten Zeitraumes keine Antwort erhalten wird.

Bei der Kommunikation über einen 3GPP-NB-IoT-Zugangsträger **DÜRFEN** im Mobilfunknetz der Deutschen Telekom **KEINE** SMS verwendet werden. Ref.: [TS.34\\_6.0\\_REQ\\_003](#).

### **Verhalten, wenn die SIM-Subskription inaktiv ist**

Falls die mit einem IoT-Gerät im Zusammenhang stehende SIM-Subskription vorübergehend (d.h. für einen bestimmten Zeitraum) stillgelegt werden soll, **MUSS** der IoT-Serviceprovider zunächst sicherstellen, dass das Kommunikationsmodul/der Chipsatz des IoT-Geräts vorübergehend stillgelegt ist, damit es nicht versuchen kann, sich im Mobilfunknetz anzumelden, nachdem die SIM stillgelegt wurde. Ref.: [TS.34\\_6.0\\_REQ\\_002](#).

### **Verhalten, wenn die SIM-Subskription dauerhaft stillgelegt ist**

Bevor die mit einem IoT-Gerät im Zusammenhang stehende SIM-Subskription in einen dauerhaft beendeten Zustand versetzt wird, **MUSS** der IoT-Serviceprovider zunächst sicherstellen, dass das Kommunikationsmodul/der Chipsatz des IoT-Geräts dauerhaft stillgelegt ist, damit es nicht versuchen kann, sich im Mobilfunknetz anzumelden, sobald die SIM stillgelegt ist.

Der IoT-Service-Provider **SOLLTE** zur dauerhaften Stilllegung von IoT-Geräten versuchen, Mechanismen zu vermeiden, die nicht leicht wartungsfähig sind, da ein manuelles Eingreifen (d.h. ein Service-Call) zwecks erneuter Aktivierung der IoT-Geräte erforderlich sein könnte. [Ref.: TS.34\\_6.0\\_REQ\\_002](#).

### Frequenz und Priorisierung von Datenübertragungen

Immer dann, wenn Daten über das Mobilfunknetz übertragen werden müssen, **SOLLTE** die IoT-Service-Plattform bzw. IoT-Server-Anwendung die Priorität jeder Kommunikation festlegen. Die IoT-Service-Plattform bzw. IoT-Server-Anwendung unterscheidet zwischen Daten mit hoher Priorität, die eine sofortige Übertragung erfordern, und verzögerungstoleranten Daten bzw. Daten mit niedrigerer Priorität, die gesammelt und außerhalb der Spitzenzeiten gesendet werden können. [Ref.: TS.34\\_4.0\\_REQ\\_018](#).

IoT-Server-Anwendungen, die über 3GPP-Low-Power-Wide-Area-(LPWA)-Mobilfunktechnik mit IoT-Geräten kommunizieren, wie beispielsweise NB-IoT und LTE-M, **MÜSSEN** die Häufigkeit der Datenübertragung so optimieren, dass die maximale Anzahl an Nachrichten pro Tag im Tarif der Konzerngesellschaft der Deutschen Telekom niemals überschritten wird.

### Datensammlung, Kompression und Transkodierung

Die IoT-Server-Anwendung **MUSS** die Anzahl an parallelen Mobilfunknetzverbindungen und die Gesamtfrequenz von Verbindungen zu IoT-Geräten über das Mobilfunknetz minimieren. Die Daten werden von der IoT-Server-Anwendung in einem Anwendungsbericht gesammelt, bevor sie komprimiert und über das Mobilfunknetz versendet werden. Gemäß der vorgesehenen Servicequalität des IoT-Service werden Datentranskodierungs- und Datenkompressionstechniken verwendet, um Verbindungsversuche und Datenvolumina zu reduzieren. [Ref.: TS.34\\_4.0\\_REQ\\_002](#).

### Häufigkeit der Übertragungsvorgänge

IoT-Server-Anwendungen, die 3GPP-Low-Power-Wide-Area-(LPWA)-Mobilfunktechnik verwenden, (wie beispielsweise NB-IoT und LTE-M), **MÜSSEN** die Häufigkeit ihrer Datenübertragungsvorgänge so optimieren, dass die von der jeweiligen Konzerngesellschaft der Deutschen Telekom vorgegebene maximale Anzahl täglicher Übertragungsvorgängen nicht überschritten wird.

### Monatliches Übertragungsvolumen

IoT-Server-Anwendungen, die 3GPP-Low-Power-Wide-Area-(LPWA)-Mobilfunktechnik verwenden (wie beispielsweise NB-IoT und LTE-M) **MÜSSEN** ihre Nutzlastgrößen so optimieren, dass die von der jeweiligen Konzerngesellschaft der Deutschen Telekom vorgegebene monatliche Volumengrenze nicht überschritten wird.

## IV. Richtlinien für IoT-Geräte

---

### Vermeidung von synchronisiertem Verhalten

Die IoT-Geräteanwendung **MUSS** synchronisiertes Verhalten mit anderen IoT-Geräten oder Ereignissen vermeiden und für die Anforderung der Verbindung mit dem Mobilfunknetz über die Konnektivitätsschicht ein zufällig ausgewähltes Muster (z.B. über einen Zeitraum von Minuten bis zu mehreren Stunden oder Tagen) verwenden. Das Auslösen von Datenübertragungen, das Neustarten der Hardware des IoT-Geräts oder seiner Subkomponenten (wie beispielsweise des Kommunikationsmoduls/des Chipsatzes) und das Ausführen von Geräteverwaltungsbefehlen (insbesondere (Neu-)Anmeldungen und Firmware-Updates) **DÜRFEN NICHT** synchronisiert werden. In mehrstufigen IoT-Geräten **MUSS** die eingebettete Service-Enablement-Schicht diese Anforderungen auf die gleiche Weise implementieren, wie bei monolithischen IoT-Geräteanwendungen. [Ref.: TS34 4.0 REQ\\_003, TS.34\\_4.2\\_REQ\\_003](#)

## Verwendung von „Always-on“ [ständiger] Konnektivität

Falls die monolithische IoT-Geräte-Anwendung sehr häufig (d.h. mit Inaktivitätszeiträumen unter zwei Stunden) Daten versendet, **MUSS** sie eine ständige PDP/PDN-Verbindung mit dem Mobilfunknetz verwenden, anstatt die genannte Verbindung zu aktivieren und zu deaktivieren. In mehrstufigen IoT-Geräten **MUSS** die eingebettete Service-Enablement-Schicht diese Anforderung erfüllen. Ref.: [TS.34\\_4.0\\_REQ\\_001](#), [TS.34\\_4.2\\_REQ\\_001](#)

## Handhabung von „Keep-Alive“ Nachrichten im Heimnetzwerk

Falls die Kommunikation zwischen dem IoT-Gerät und dem Mobilfunknetz IP-basiert ist, müssen unter Umständen TCP/UDP-Keep-Alive-Nachrichten verwendet werden. In diesen Fällen **MUSS** die monolithische IoT-Geräteanwendung die serverspezifischen Timer bzw. die Timer der Mobilfunknetz-Firewall und den TCP\_IDLE-Wert bzw. den UDP\_IDLE-Wert (NAT-Timer gemäß der Deutschen Telekom für Verbraucher-APNs [Zugangspunkte] oder für selbstverwaltete NAT bei Unternehmen, im Falle von privaten APNs) automatisch erkennen, wenn Push-Services verwendet werden. Dazu wird das Abfrageintervall dynamisch erhöht, bis ein Mobilfunknetz-Timeout stattfindet; danach wird knapp unter dem Timeout-Wert gearbeitet. Festgelegte Abfrageintervalle **DÜRFEN** von der monolithischen IoT-Geräteanwendung **NICHT** verwendet werden, da die Konzerngesellschaften der Deutschen Telekom unterschiedliche Abfrageintervallwerte haben, die sich unter Umständen dynamisch an die Mobilfunknetzbelastung anpassen. In mehrstufigen IoT-Geräten **MUSS** die eingebettete Service-Enablement-Schicht diese Anforderungen auf dieselbe Weise implementieren wie bei monolithischen IoT-Geräteanwendungen. Diese Anforderung gilt nicht für IoT-Geräteanwendungen, die mit der IoT-Server-Anwendung über 3GPP-Low-Power-Wide-Area-(LPWA)-Mobilfunktechnik, wie beispielsweise NB-IoT und LTE-M kommunizieren. 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](#).

Monolithische IoT-Geräte-Anwendungen, die über 3GPP-Low-Power-Wide-Area-(LPWA)-Mobilfunktechnik, wie beispielsweise NB-IoT und LTE-M, mit der IoT-Server-Anwendung kommunizieren, **SOLLTEN** im Heimnetzwerk **KEINE** TCP/UDP „Keep-Alive“ Nachrichten einsetzen. In mehrstufigen IoT-Geräten **MUSS** die eingebettete Service-Enablement-Schicht diese Anforderungen auf dieselbe Weise implementieren wie bei monolithischen IoT-Geräteanwendungen.

## Handhabung von „Keep-Alive“ Nachrichten im Roaming-Netzwerk

Um die Möglichkeit zu minimieren, dass die IP-Konnektivität verloren geht – z.B. aufgrund des Ablaufs von Firewall-Timern – wenn das IoT-Gerät über einen längeren Zeitraum (länger als zwei Stunden) in einem Roaming-Netzwerk eingebucht ist, **MUSS** die IoT-Geräteanwendung regelmäßig (in Abständen von weniger als zwei Stunden) kleine Datenmengen über das besuchte Netz auf die IoT-Serviceplattform übertragen. Ein Zufallstimer löst diesen Mechanismus aus und stellt dabei sicher, dass die simultane Datenübertragung durch eine große Anzahl an IoT-Geräten über das besuchte Netz vermieden wird. In mehrstufigen IoT-Geräten **MUSS** die eingebettete Service-Enablement-Schicht diese Anforderung erfüllen. Diese Anforderung gilt nicht für IoT-Geräteanwendungen, die mit der IoT-Server-Anwendung über 3GPP-Low-Power-Wide-Area-(LPWA)-Mobilfunktechnik, wie beispielsweise NB-IoT und LTE-M kommunizieren.

Monolithische IoT-Geräteanwendungen, die über 3GPP-Low-Power-Wide-Area-(LPWA)-Mobilfunktechniken, wie beispielsweise NB-IoT und LTE-M, mit der IoT-Server-Anwendung kommunizieren, **SOLLTEN** im Roaming-Netz **KEINE** TCP/UDP „Keep-Alive“ Nachrichten einsetzen. In mehrstufigen IoT-Geräten **MUSS** die eingebettete Service-Enablement-Schicht diese Anforderungen auf dieselbe Weise implementieren wie bei monolithischen IoT-Geräteanwendungen.

## IoT-Servicekoordination

Falls die monolithische IoT-Geräteanwendung unter Verwendung des gleichen Kommunikationsmoduls/Chipsatzes mit mehreren IoT-Server-Anwendungen kommuniziert, **SOLLTE** die IoT-Geräteanwendung die Nutzlast-Übertragung jedes IoT-Service so koordinieren, dass das Mobilfunknetz effizient genutzt wird. In mehrstufigen IoT-Geräten **MUSS** die eingebettete Service-Enablement-Schicht dieser Anforderung entsprechen.

## Datensammlung, Kompression und Transkodierung

Die monolithische IoT-Geräteanwendung **MUSS** die Anzahl der parallelen Mobilfunknetzverbindungen und die Gesamtfrequenz der Verbindungen zwischen dem IoT-Gerät und dem Mobilfunknetz minimieren. Die Daten werden von der IoT-Geräteanwendung in einem Anwendungsbericht gesammelt, bevor sie komprimiert und über das Mobilfunknetz versendet werden. Gemäß der vorgesehenen Servicequalität des IoT-Service werden Datentranskodierungs- und Datenkompressionstechniken verwendet, um Verbindungsversuche und Datenvolumina zu reduzieren. In mehrstufigen IoT-Geräten **MUSS** die eingebettete Service-Enablement-Schicht dieser Anforderung entsprechen.

Die monolithische IoT-Geräteanwendung **MUSS** das Datenvolumen, das sie über einen bestimmten Zeitraum sendet und empfängt, überwachen. Falls das Datenvolumen in Kürze einen vom IoT-Serviceprovider (siehe **Anhang: Mobile IoT „No Harm to Network“ Policies**) festgelegten Höchstwert überschreiten wird, sendet die IoT-Geräte-Anwendung einen Bericht an die IoT-Service-Plattform und stoppt das reguläre Senden von Daten, bis die erforderliche Frist verstrichen ist. In mehrstufigen IoT-Geräten **SOLLTE** die eingebettete Service-Enablement-Schicht dieser Anforderung entsprechen. 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](#).

Monolithische IoT-Geräteanwendungen, die über 3GPP-Low-Power-Wide-Area-(LPWA)-Mobilfunktechnik, wie beispielsweise NB-IoT und LTE-M, mit IoT-Server-Anwendungen kommunizieren, **MÜSSEN** ihre Nutzlastgrößen so optimieren, dass sie die monatlichen oder lebenslangen Volumengrenzen der Konzerngesellschaft der Deutschen Telekom einhalten. In mehrstufigen IoT-Geräten **MUSS** die eingebettete Service-Enablement-Schicht dieser Anforderung entsprechen.

## Monatliches Übertragungsvolumen – Vom IoT Serviceprovider festgesetztes Limit

Die IoT-Geräteanwendung **MUSS** das Datenvolumen, das sie über einen bestimmten Zeitraum sendet und empfängt, überwachen. Falls das Datenvolumen in Kürze einen vom IoT-Serviceprovider oder vom (prepaid) SIM-Tarif festgelegten Höchstwert überschreiten wird, sendet die IoT-Geräte-Anwendung einen Bericht an die IoT-Service-Plattform und stoppt das reguläre Senden von Daten, bis die erforderliche Frist verstrichen ist. Ref: [TS.34\\_4.0\\_REQ\\_013](#), [TS.34\\_4.2\\_REQ\\_013](#).

## Priorisierung und Häufigkeit von Datenübertragungen

Immer dann, wenn Daten über ein Mobilfunknetz übertragen werden müssen, **SOLLTE** die IoT-Geräteanwendung die Priorität jeder Kommunikation festlegen. Die IoT-Geräteanwendung unterscheidet zwischen Daten mit hoher Priorität, die eine sofortige Übertragung erfordern, und verzögerungstoleranten Daten bzw. Daten mit niedriger Priorität, die gesammelt und außerhalb der Spitzenzeiten gesendet werden können. In mehrstufigen IoT-Geräten sollte die eingebettete Service-Enablement-Schicht die von der IoT-Geräteanwendung übermittelten Informationen über die Wichtigkeit und Dringlichkeit der Daten zur Bereitstellung des IoT-Services ohne negative Auswirkungen auf das Netzwerk berücksichtigen. Ref.: [TS.34\\_4.0\\_REQ\\_018](#), [TS.34\\_4.2\\_REQ\\_018](#), [TS.34\\_4.1\\_REQ\\_003](#).

Die monolithische IoT-Geräteanwendung **MUSS** die Anzahl der Netzwerkverbindungen, die es über einen bestimmten Zeitraum aufzubauen versucht, überwachen. Falls die Anzahl an Verbindungsversuchen einen vom IoT-



Serviceprovider (siehe [Anhang: Mobile IoT „No Harm to Network“ Policies](#)) festgelegten Höchstwert überschreitet, sendet die IoT-Geräteanwendung einen Bericht an die IoT-Service-Plattform und stoppt die Mobilfunknetz-Konnektivitätsanfrage, bis die erforderliche Frist vorbei ist. In mehrstufigen IoT-Geräten **MUSS** die eingebettete Service-Enablement-Schicht dieser Anforderung entsprechen. Ref: [TS.34\\_4.0\\_REQ\\_012](#), [TS.34\\_4.2\\_REQ\\_012](#).

Monolithische IoT-Geräteanwendungen, die über 3GPP-Low-Power-Wide-Area-(LPWA)-Mobilfunktechnik, wie beispielsweise NB-IoT und LTE-M, mit IoT-Server-Anwendungen kommunizieren, **MÜSSEN** die Häufigkeit der Datenübertragung so optimieren, dass die maximale Anzahl an Nachrichten pro Tag im Tarif der Konzerngesellschaft der Deutschen Telekom niemals überschritten wird (siehe [Anhang: Mobile IoT „No Harm to Network“ Policies](#)). In mehrstufigen IoT-Geräten **MUSS** die eingebettete Service-Enablement-Schicht dieser Anforderung entsprechen.

### **Vermeidung von Datenübertragung als „letzter Atemzug“**

Die IoT-Geräteanwendungen **MUSS** dafür sorgen, dass es im Falle einer systemweiten Störung (z.B. regionaler Stromausfall) nicht dazu kommt, dass eine signifikante Anzahl von IoT-Geräten (z.B. > 100 Geräte) gleichzeitig versuchen, über 3GPP NB-IoT oder LTE-M zu kommunizieren.

### **Kommunikation außerhalb der Spitzenzeiten**

Falls es der IoT-Service zulässt, **SOLLTE** die monolithische IoT-Geräteanwendung es vermeiden, die Kommunikation über das Mobilfunknetz in Phasen starker Nutzung zu konzentrieren (d.h. Übertragungen in den frühen Morgenstunden sind zu bevorzugen). In mehrstufigen IoT-Geräten **SOLLTE** die eingebettete Service-Enablement-Schicht dieser Anforderung entsprechen. Ref: [TS.34\\_4.0\\_REQ\\_016](#), [TS.34\\_4.2\\_REQ\\_016](#).

### **Lokalisierte Kommunikation**

Die monolithische IoT-Geräteanwendung **MUSS** alle geographischen Netzwerkbelastungsprobleme minimieren. Es **MUSS** keine Koordination aller IoT-Geräte in einer bestimmten Region des IoT-Service geben, um ähnliche Operationen durchzuführen, die zu Netzwerkbelastung führen, z.B. Firmware-Updates. In mehrstufigen IoT-Geräten **MUSS** die eingebettete Service-Enablement-Schicht diese Anforderung erfüllen. Ref: [TS.34\\_4.0\\_REQ\\_017](#), [TS.34\\_4.2\\_REQ\\_017](#).

### **Anpassung an Mobilfunknetzfähigkeiten, Übertragungsgeschwindigkeit und Latenz**

Die monolithische IoT-Geräteanwendung **MUSS** sich an Änderungen in der Funktionsfähigkeit und Auslastung des Mobilfunknetzes anpassen können. Des Weiteren ist sie so konzipiert, dass sie mit Schwankungen in der Übertragungsgeschwindigkeit und der Latenz des Mobilfunknetzes umgehen kann, unter Berücksichtigung von Unterschieden in der Durchlaufzeit, der Übertragungsgeschwindigkeit und der Latenz, wenn sie zwischen den verschiedenen 3GPP-Mobilfunktechniken (d.h. 2G, 3G, LTE and LPWA) wechselt. In mehrstufigen IoT-Geräten **MUSS** die eingebettete Service-Enablement-Schicht dieser Anforderung entsprechen; die IoT-Geräteanwendung **MUSS** in der Lage sein, von der Service-Enablement-Schicht Informationen über die Geschwindigkeit und Verbindungsqualität des Mobilfunknetzes abzurufen, um die entsprechende Qualität der Inhalte von der Cloud-Plattform anzufordern. 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](#).

Wenn Datengeschwindigkeit und Latenz kritisch für den IoT-Service sind, **SOLLTE** die monolithische IoT-Geräteanwendung die Geschwindigkeit und Verbindungsqualität des Mobilfunknetzes ständig überwachen, um die entsprechende Qualität der Inhalte von der Infrastruktur des IoT-Diensteanbieters anzufordern. In mehrstufigen IoT-Geräten **SOLLTE** die eingebettete IoT-Service-Enablement-Schicht die Geschwindigkeit und Verbindungsqualität des Mobilfunknetzes ständig überwachen, um die entsprechende Qualität der Inhalte von der Cloud-Plattform



anzufordern. Die IoT-Geräteanwendung ruft Informationen zur Geschwindigkeit und Verbindungsqualität des Mobilfunknetzes von der IoT-Service-Enablement-Schicht ab. Ref: [TS.34\\_4.0\\_REQ\\_010](#), [TS.34\\_4.2\\_REQ\\_010](#).

### **Energiesparmodus**

Wenn die monolithische IoT-Geräteanwendung während eines Zeitraums von über 24 Stunden keine Daten mit der IoT-Service-Plattform austauschen muss und der IoT-Service eine gewisse Latenz tolerieren kann, **SOLLTE** das IoT-Gerät einen Energiesparmodus aktivieren, in dem das Kommunikationsmodul/der Chipsatz des Geräts zwischen einzelnen Datenübertragungen effektiv herabgesetzt wird. Das senkt den Energieverbrauch des IoT-Geräts und reduziert die Signalisierungslast im Mobilfunknetz. Ref: [TS.34\\_4.0\\_REQ\\_020](#), [TS.34\\_4.2\\_REQ\\_020](#), [TS.34\\_4.1\\_REQ\\_004](#).

Monolithische IoT-Geräteanwendungen, die über 3GPP-Low-Power-Wide-Area-(LPWA)-Zugangsträger wie beispielsweise NB-IoT und LTE-M kommunizieren, **SOLLTEN** ihr Kommunikationsmodul/ihren Chipsatz **NICHT** vollständig abschalten. Stattdessen **SOLLTEN** die 3GPP-Energiesparfunktionen verwendet werden, um so eine energieintensive Zellsuche zu vermeiden. In mehrstufigen IoT-Geräten sollte die eingebettete Service-Enablement-Schicht diese Anforderung auf die gleiche Weise implementieren wie bei monolithischen IoT-Geräteanwendungen.

### **Verhalten, wenn die IoT-Service-Plattform vorübergehend nicht erreichbar oder offline ist**

Falls die monolithische IoT-Service-Plattform vorübergehend offline ist, **MUSS** die IoT-Geräteanwendung zuerst eine Diagnose erstellen, ob die Kommunikationsprobleme mit dem Server durch die Kommunikation auf höheren Schichten (TCP/IP, UDP, ATM...) verursacht werden. Die Mechanismen der höheren Schichten **MÜSSEN** dann versuchen, die Verbindung mit dem Server wiederherzustellen. Dies erfolgt durch die Bewertung der Konnektivität in einem schrittweisen Ansatz von oben nach unten (und erforderlichenfalls dem Versuch, diese wiederherzustellen). In mehrstufigen IoT-Geräten **MUSS** die eingebettete Service-Enablement-Schicht dieser Anforderung entsprechen. Ref: [TS.34\\_4.0\\_REQ\\_011](#), [TS.34\\_4.2\\_REQ\\_011](#), [TS.34\\_4.0\\_REQ\\_029](#).

Die monolithische IoT-Geräteanwendung **DARF NICHT** wiederholt einen anwendungsgesteuerten Neustart des Kommunikationsmoduls/Chipsatzes initiieren. Die IoT-Geräte **MÜSSEN** erneut versuchen, eine Verbindung mit der IoT-Service-Plattform herzustellen, wobei der Back-Off-Zeitraum zunehmend zu verlängern ist. In mehrstufigen IoT-Geräten **MUSS** die eingebettete Service-Enablement-Schicht diese Anforderungen auf die gleiche Weise implementieren wie bei monolithischen IoT-Geräteanwendungen.

Wenn das monolithische IoT-Gerät entdeckt, dass die IoT-Service-Plattform wieder online ist, **MUSS** es einen Zufallstimer verwenden, bevor Kommunikationsanfragen an das Mobilfunknetz ausgelöst werden. In mehrstufigen IoT-Geräten **MUSS** die eingebettete Service-Enablement-Schicht dieser Anforderung entsprechen.

### **Verhalten bei verlorengegangener Netzabdeckung (GPS, GLONASS, LAN, WAN)**

Wenn die GNSS-Abdeckung (GPS, GLONASS, BeiDou, Galileo) verloren geht, **DARF** die monolithische IoT-Geräteanwendung das Kommunikationsmodul/den Chipsatz **NICHT** neu starten. Die IoT-Geräteanwendung **SOLLTE** eine Diagnose durchführen, das betroffene Hardwareelement neu starten und eine Warnmeldung an die IoT-Server-Anwendung schicken. In mehrstufigen IoT-Geräten **MUSS** die eingebettete Service-Enablement-Schicht diese Anforderungen auf die gleiche Weise implementieren wie bei monolithischen IoT-Geräteanwendungen.

Wenn die LAN- bzw. WAN-Netzabdeckung verloren geht, **DARF** das monolithische IoT-Gerät das IoT-Gerät bzw. das Kommunikationsmodul/den Chipsatz **NICHT** neu starten. Die IoT-Geräteanwendung **MUSS** einen erneuten Scanversuch unternehmen, um die Konnektivität mit dem Mobilfunknetz herzustellen, wobei der Back-Off-Zeitraum zunehmend zu verlängern ist. In mehrstufigen IoT-Geräten **MUSS** die eingebettete Service-Enablement-Schicht dieser Anforderung entsprechen.

### **Verhalten bei einer Fehlfunktion der Sensoren / Aktoren**

Wenn eingebaute Sensoren bzw. Aktoren nicht richtig funktionieren, **DARF** die monolithische IoT-Geräteanwendung das Kommunikationsmodul/den Chipsatz **NICHT** neu starten. Die IoT-Geräteanwendung **SOLLTE** eine Diagnose durchführen, das betroffene Hardwareelement neu starten und eine Warnmeldung an die IoT-Server-Anwendung schicken. In mehrstufigen IoT-Geräten **MUSS** die eingebettete Service-Enablement-Schicht diese Anforderungen auf die gleiche Weise implementieren wie bei monolithischen IoT-Geräteanwendungen.

### **Verhalten bei einer Auslösung der Sensoralarme/Aktoren**

Wenn eingebaute Sensoren bzw. Aktoren ausgelöst werden, **DARF** die monolithische IoT-Geräteanwendung das Kommunikationsmodul/den Chipsatz **NICHT** neu starten. Stattdessen **SOLLTE** die IoT-Geräteanwendung eine Warnmeldung an die IoT-Server-Anwendung schicken. In mehrstufigen IoT-Geräten **MUSS** die eingebettete Service-Enablement-Schicht diese Anforderungen auf die gleiche Weise implementieren wie bei monolithischen IoT-Geräteanwendungen

### **Verhalten bei schwachem Ladezustand oder Stromausfall**

Die monolithische IoT-Geräteanwendung **SOLLTE** eine Benachrichtigung mit relevanten Informationen an die IoT-Service-Plattform senden, wenn ein unerwarteter Stromausfall oder ein Batterieproblem auftritt. In mehrstufigen IoT-Geräten **SOLLTE** die eingebettete Service-Enablement-Schicht diese Anforderung umsetzen. Ref: [TS.34\\_4.0\\_REQ\\_014](#), [TS.34\\_4.2\\_REQ\\_014](#).

### **Verhalten, wenn der Gerätespeicher voll ist**

Wenn der Speicher des monolithischen IoT-Geräts voll ist, beispielsweise aufgrund der Menge der gesammelten Daten oder aufgrund eines ungewollten Speicherlecks, **DARF** die IoT-Geräteanwendung das Kommunikationsmodul/den Chipsatz **NICHT** neu starten. Die IoT-Geräteanwendung **SOLLTE** eine Diagnose durchführen, das betroffene Hardwareelement neu starten und eine Warnmeldung an die IoT-Server-Anwendung schicken. In mehrstufigen IoT-Geräten **MUSS** die eingebettete Service-Enablement-Schicht diese Anforderung auf dieselbe Weise implementieren wie bei monolithischen IoT-Geräteanwendungen.

### **Verhalten, wenn Kommunikationsanfragen fehlschlagen**

Die monolithische IoT-Geräteanwendung **MUSS** Situationen, in denen Kommunikationsanfragen fehlschlagen, immer so handhaben, dass es zu keiner Beeinträchtigung des Mobilfunknetzes kommt. Das Mobilfunknetz kann Kommunikationsanfragen eines IoT-Geräts mit einem 3GPP™-error Cause Code [Fehlercode] (siehe GSMA TS.34) ablehnen. Wenn die IoT-Geräteanwendung feststellt, dass ihre Anfragen abgelehnt werden, **MUSS** sie erneut versuchen, Verbindungsanfragen an das Mobilfunknetz zu senden, wobei der Back-Off-Zeitraum zunehmend zu verlängern ist. Die IoT-Geräteanwendung **DARF KEINEN** anwendungsgesteuerten Neustart des Kommunikationsmoduls / Chipsatzes veranlassen, um zu versuchen, die Entscheidung des Mobilfunknetzes zu ignorieren bzw. sich darüber hinwegzusetzen. In mehrstufigen IoT-Geräten **MUSS** die eingebettete Service-Enablement-Schicht diese Anforderung auf die gleiche Weise implementieren wie bei monolithischen IoT-Geräteanwendungen. Darüber hinaus **MUSS** die IoT-Geräteanwendung immer darauf vorbereitet sein, Situationen zu handhaben, in denen Kommunikationsanforderungen fehlschlagen, wenn ein solcher Ausfall von der eingebetteten Dienst-Enablement-Schicht gemeldet wird.

Kommunikationsanfragen der IoT-Geräteanwendung **DÜRFEN NICHT** unbegrenzt wiederholt werden - alle Anfragen müssen irgendwann abgebrochen und von der IoT-Geräteanwendung beendet werden. In mehrstufigen IoT-Geräten **MUSS** die eingebettete Service-Enablement-Schicht dieser Anforderung entsprechen. Ref: [TS.34\\_4.0\\_REQ\\_011](#), [TS.34\\_4.2\\_REQ\\_011](#), [TS.34\\_4.1\\_REQ\\_002](#).

### Verhalten, wenn vom Gerät stammende SMS gesperrt sind

Wenn die monolithische IoT-Geräteanwendung feststellt, dass ihre Subskription für MO-SMS vom Mobilfunknetz gesperrt ist, **MUSS** die IoT-Geräteanwendung erneut versuchen, Verbindungsanfragen an das Mobilfunknetz zu senden, wobei der Back-Off-Zeitraum zunehmend zu verlängern ist. Die IoT-Geräteanwendung **DARF KEINEN** anwendungsgesteuerten Neustart des Kommunikationsmoduls/Chipsatzes veranlassen. In mehrstufigen IoT-Geräten **MUSS** die eingebettete Service-Enablement-Schicht diese Anforderung auf dieselbe Weise implementieren wie bei monolithischen IoT-Geräteanwendungen.

### Verhalten, wenn ein Attach Reject mit Session Management Cause-Code 26 empfangen wird

Für den Fall dass die monolithische IoT-Geräteanwendung einen PDN Connectivity Reject mit Session Management Cause-Code #26 erhält, **MUSS** eine angemessene Behandlungsroutine implementiert sein, um einen dauerhaften Dienstaussfall zu vermeiden. Zu berücksichtigen sind dabei die Konfiguration des Backoff-Timers T3396 durch den Netzbetreiber, sowie die Vorgaben der Spezifikation 3GPPTM TS 24.008:

- Wenn der Backoff-Timer T3396 vom Netzwerk nicht übermittelt wird, darf das IoT-Gerät erneut versuchen, sich mit dem Netzwerk zu verbinden;
- Wenn der Backoff-Timer T3396 vom Netzwerk übermittelt wird, aber als deaktiviert markiert ist, darf das IoT-Gerät versuchen, sich erneut mit dem Netzwerk zu verbinden, allerdings erst nach einem Neustart des Kommunikationsmoduls durch die Geräteapplikation;
- Wenn der Backoff-Timer T3396 vom Netzwerk mit einem Wert von „0“ übermittelt wird, darf das IoT-Gerät erneut versuchen sich mit dem Netzwerk zu verbinden;
- Wenn der Backoff-Timer T3396 vom Netzwerk mit einem Wert zwischen „1“ und „1116000“ Sekunden (310 Stunden) übermittelt wird, dann **DARF** das IoT-Gerät das Kommunikationsmodul/den Chipsatz **NICHT** abschalten, da dann der Backoff-Timer nicht weitergezählt wird. Soweit das Kommunikationsmodul/der Chipsatz den NB-IoT / LTE-M Energiesparmodus unterstützt, **SOLLTE** es in diesen Modus versetzt werden damit der Backoff-Timer weitergezählt wird und schlussendlich ablaufen kann.

### Neuauswahl von Funkzugangstechnologie (RAT) -Trägern

Im Falle von Massenbereitstellungen von IoT-Geräten (z.B. > 10.000 Geräte innerhalb des gleichen Mobilfunknetzwerks) und falls das monolithische IoT-Gerät mehr als eine Zugangstechnologie-Familie unterstützt (z.B. 3GPP™, WLAN) **MUSS** die IoT-Geräteanwendung eine randomisierte Verzögerungszeit abwarten, bevor sie zu einer anderen Zugangstechnologiefamilie wechselt. In mehrstufigen IoT-Geräten **MUSS** die eingebettete Service-Enablement-Schicht dieser Anforderung entsprechen. Ref: [TS.34\\_4.0\\_REQ\\_027](#), [TS.34\\_4.2\\_REQ\\_027](#).

Die monolithische IoT-Geräteanwendung **MUSS** einen Schutzmechanismus anwenden, um ein häufiges Hin- und Herwechseln zwischen diesen verschiedenen Technologien zu verhindern. Dies erfolgt durch die Begrenzung der Häufigkeit der Neuauswahl mit geeigneten Hysterese-Mechanismen. In mehrstufigen IoT-Geräten **MUSS** die eingebettete Service-Enablement-Schicht dieser Anforderung entsprechen. Ref: [TS.34\\_4.0\\_REQ\\_026](#), [TS.34\\_4.2\\_REQ\\_026](#).

### Handhabung eines Service-Verlusts im Roamingnetz

Die monolithische IoT-Geräteanwendung **MUSS** immer darauf vorbereitet sein, bei einem vorübergehenden Aufenthalt im Roamingnetz eine eventuell verlorengegangene Ende-zu-Ende-Verbindung wiederherzustellen. Dies wird durch einen stufenweisen Top-Down-Wiederherstellungs-Algorithmus umgesetzt, der für jede Protokollschicht eine Diagnose erstellt. Falls die Wiederherstellung einer Schicht fehlschlägt, initiiert der Algorithmus das Wiederherstellungsverfahren auf der nächstniedrigeren Protokollschicht. Dies kann zum Beispiel wie folgt geschehen:

- Schritt 1 Wiederherstellung der Verbindung auf einer höheren Schicht, z.B. VPN-Tunnel, SSH-Sitzungen etc.
- Schritt 2 Wiederherstellung der PDN-Verbindung oder des PDP-Kontexts,
- Schritt 3 Re-Attach (Neuanmeldung) an das Netz,
- Schritt 4 Erneutes Auslösen einer einfachen Netzauswahl,
- Schritt 5 Kompletter Neustart des Geräts.

Alle Wiederherstellungsverfahren **MÜSSEN**, zur Vermeidung eines exzessiven Versendens von Signalen an das Netz, ordnungsgemäß implementiert werden. Dazu gehört möglicherweise die Verwendung von Zufallstriggern und schrittweisen Back-Off-Wiederholungsmechanismen. Grenzen und Timer-Werte können dabei von den Anforderungen des IoT-Service abhängen. In mehrstufigen IoT-Geräten **MUSS** die eingebettete Service-Enablement-Schicht dieser Anforderung entsprechen. Ref: [TS.34\\_4.0\\_REQ\\_029](#), [TS.34\\_4.2\\_REQ\\_029](#).

### Überwachungsmechanismus

Die monolithische IoT-Geräteanwendung **SOLLTE** nach Ablauf eines konfigurierbaren Timers mit zufälligem Offset regelmäßig eine Nachricht an die IoT-Serveranwendung senden, und zwar mit einer Häufigkeit, die die Leistung des IoT-Geräts (z. B. Akkulaufzeit) nicht beeinträchtigt. Solche seltenen Nachrichten dienen dazu, die Verfügbarkeit und ordnungsgemäße Funktion des IoT-Geräts zu bestätigen (z. B. als Diagnosenachrichten). Der IoT-Dienstanbieter verwendet solche Nachrichten als Mechanismus, um regelmäßig den Zustand der eingesetzten IoT-Geräte zu überwachen.

### IPv4/v6-Dual-Stack-Unterstützung

Die monolithische IoT-Geräteanwendung **MUSS** für ordnungsgemäßes Roaming in Mobilfunknetzen, die entweder nur IPv4 oder nur IPv6 oder nur Dual-Stack unterstützen, IPv4/v6-Dual-Stack (PDN-Art = IPv4v6) unterstützen.

### Gerät auf Werkseinstellungen zurücksetzen

Die monolithische IoT-Geräteanwendung **SOLLTE** ein "Zurücksetzen auf Werkseinstellungen" über eine Remote- und lokale Verbindung unterstützen. In mehrstufigen IoT-Geräten **SOLLTE** die eingebettete Service-Enablement-Schicht diese Anforderung entsprechen. Ref: [TS.34\\_4.0\\_REQ\\_024](#), [TS.34\\_4.2\\_REQ\\_024](#).

### Gerätezeit Resynchronisierung

Die monolithische IoT-Geräteanwendung **SOLLTE** die Zeitsynchronisierung über Remote- und lokale Verbindungen unterstützen. In mehrstufigen IoT-Geräten **SOLLTE** die eingebettete Service-Enablement-Schicht diese Anforderung entsprechen. Ref: [TS.34\\_4.0\\_REQ\\_025](#), [TS.34\\_4.2\\_REQ\\_025](#).

### Sprachdienste, 5G/4G Campus

Sowohl monolithische als auch mehrstufige IoT-Geräteanwendungen **MÜSSEN**, soweit anwendbar, die in den separaten Richtlinien für IoT-Lösungen für die Konnektivität festgelegten obligatorischen Anforderungen erfüllen:

- Richtlinien für IoT-Lösungen – 5G für IoT-Konnektivität;
- Richtlinien für IoT-Lösungen – Sprachdienste (inkl. VoLTE) für IoT-Konnektivität.

## V. Anhang: No Harm to Network Policies

---

### Maximale Anzahl von Verbindungsanfragen

- **NB-IoT:** 24 Verbindungsanfragen (Network Attach) / Tag / Gerät (d.h. im Durchschnitt einmal pro Stunde)
- **LTE-M:** 144 Verbindungsanfragen (Network Attach) / Tag / Gerät (d.h. im Durchschnitt sechsmal pro Stunde)
- **LTE (alle Kategorien):** 144 Verbindungsanfragen (Network Attach) / Tag / Gerät (d.h. im Durchschnitt sechsmal pro Stunde)
- **5G-New Radio NSA:** 144 Verbindungsanfragen (Network Attach) / Tag / Gerät (d.h. im Durchschnitt sechsmal pro Stunde)

### Maximale Anzahl täglicher Nachrichten

- **NB-IoT:** 120 Anwendungsnachrichten / Tag / Gerät (d.h. im Durchschnitt 5 Nachrichten pro Stunde); keine Volumenbeschränkungen pro Nachricht
- **LTE-M:** 720 Anwendungsnachrichten / Tag / Gerät (d.h. im Durchschnitt 30 Nachrichten pro Stunde); keine Volumenbeschränkungen pro Nachricht
- **LTE (alle Kategorien):** 720 Anwendungsnachrichten / Tag / Gerät (d.h. im Durchschnitt 30 Nachrichten pro Stunde); keine Volumenbeschränkungen pro Nachricht
- **5G-New Radio NSA:** 720 Anwendungsnachrichten / Tag / Gerät (d.h. im Durchschnitt 30 Nachrichten pro Stunde); keine Volumenbeschränkungen pro Nachricht

### Maximales Datenvolumen pro Einzelgerät

Bitte beachten: Es können Einschränkungen bezüglich des Lebenszeit-Volumens oder Pooling-Einschränkungen bestehen, die das durchschnittliche monatliche Datenvolumen des Kunden begrenzen. Bei Verwendung eines Multimode-Tarifs stellen die folgenden Zahlen die maximal zulässige Grenze pro Technologie dar:

- **NB-IoT:** Durchschnittlich **1 Mbyte\*** / Monat / Gerät; Tarifspezifische Einschränkungen können bestehen
- **LTE-M:** Durchschnittlich **500 Mbyte\*** / Monat / Gerät; Tarifspezifische Einschränkungen können bestehen
- **LTE (alle Kategorien):** **Keine Einschränkungen**, sofern nicht durch entsprechende Tarifbedingungen definiert
- **5G-New Radio NSA:** **Keine Einschränkungen**, sofern nicht durch entsprechende Tarifbedingungen definiert

\* Datenverkehr für Firmware-Updates ausgeschlossen.

## Freigabehistorie

---

<b>Veröffentlicht am</b>	<b><u>Version</u></b>	<b><u>Verfasser</u></b>
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)
07.11.2022	2.7	Miguel Rodriguez (DT IOT)
12.12.2022	2.8	Miguel Rodriguez (DT IOT)