

NB-IoTの仕組み ～脱インターネット NIDDとは～

ソフトバンク株式会社

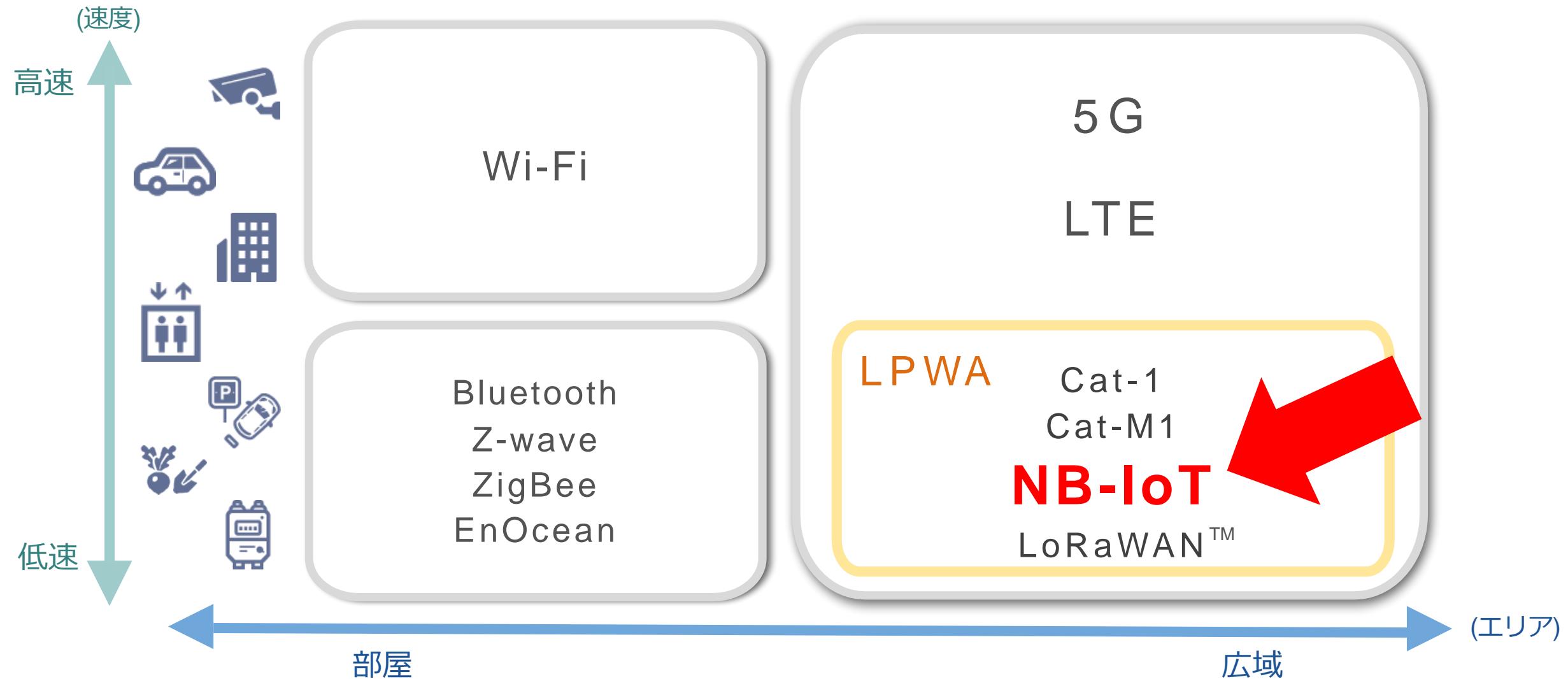
Agenda

- ✓ NB-IoTの特長
- ✓ NB-IoTの提供方法
- ✓ NIDD SCEF Based Delivery動作の概要
- ✓ NIDD SCEF Based Delivery 適用例

Agenda

- ✓ NB-IoTの特長
- ✓ NB-IoTの提供方法
- ✓ NIDD SCEF Based Delivery動作の概要
- ✓ NIDD SCEF Based Delivery 適用例

NB-IoTの適用領域

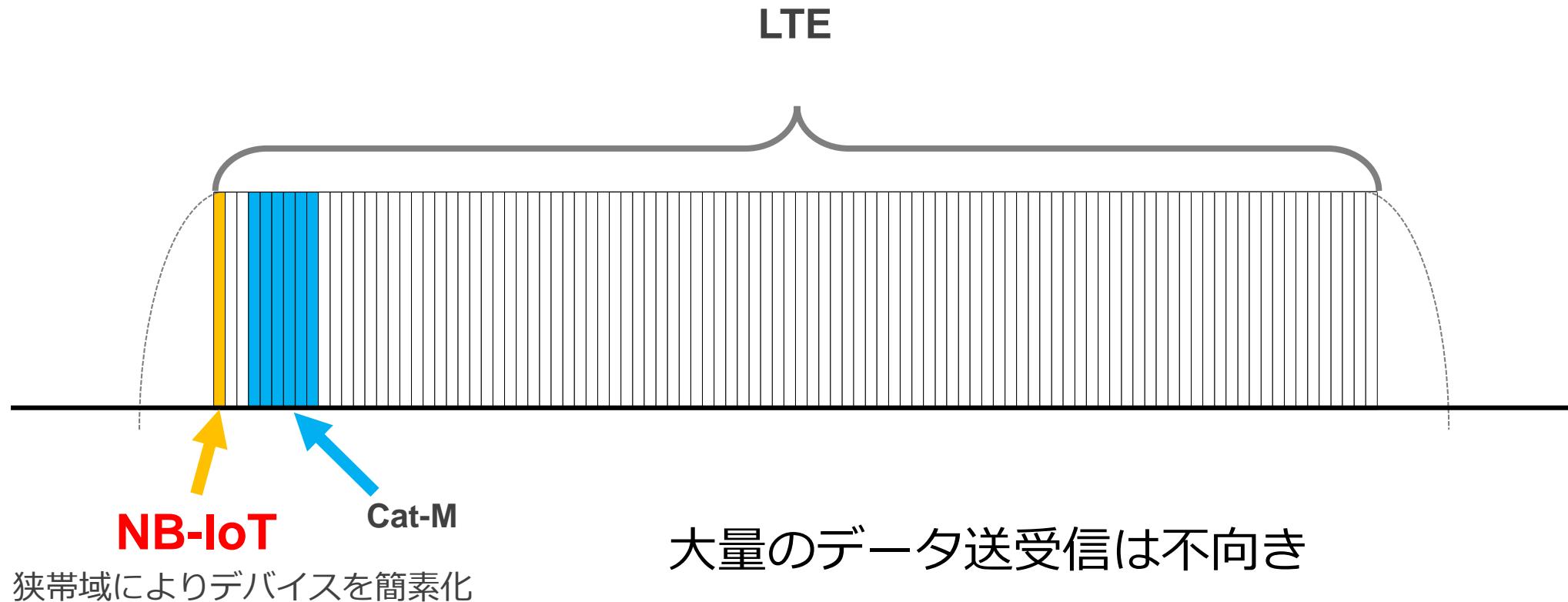


NB-IoTが目指すもの

- 簡素化によるデバイスコストの大幅な低減
- 電池の長寿命化
- カバレッジの広域化
- 大量のデバイスを効率よく収容

少量データの送受信をターゲットに
デバイスの簡素化を重視した仕様

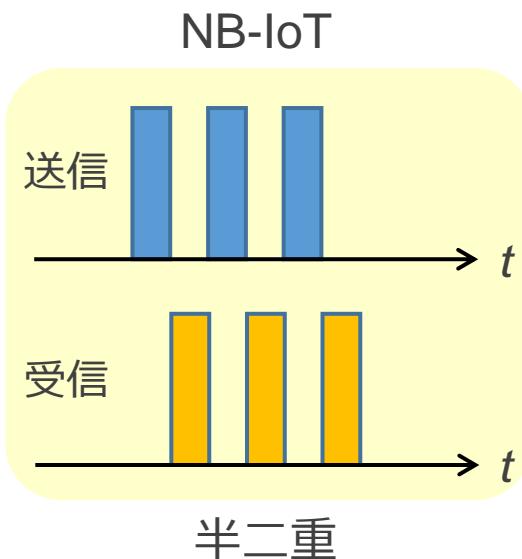
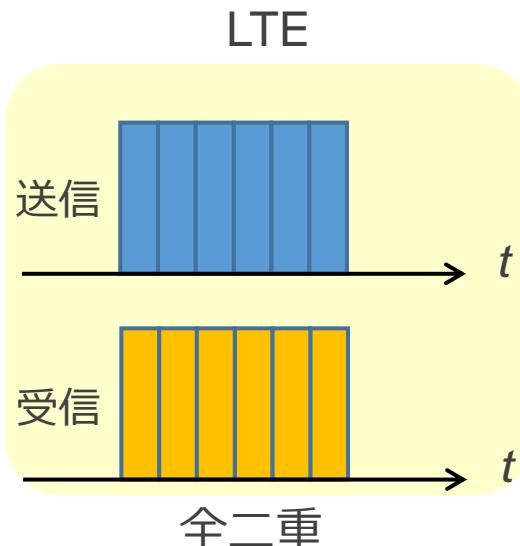
隙間を使う



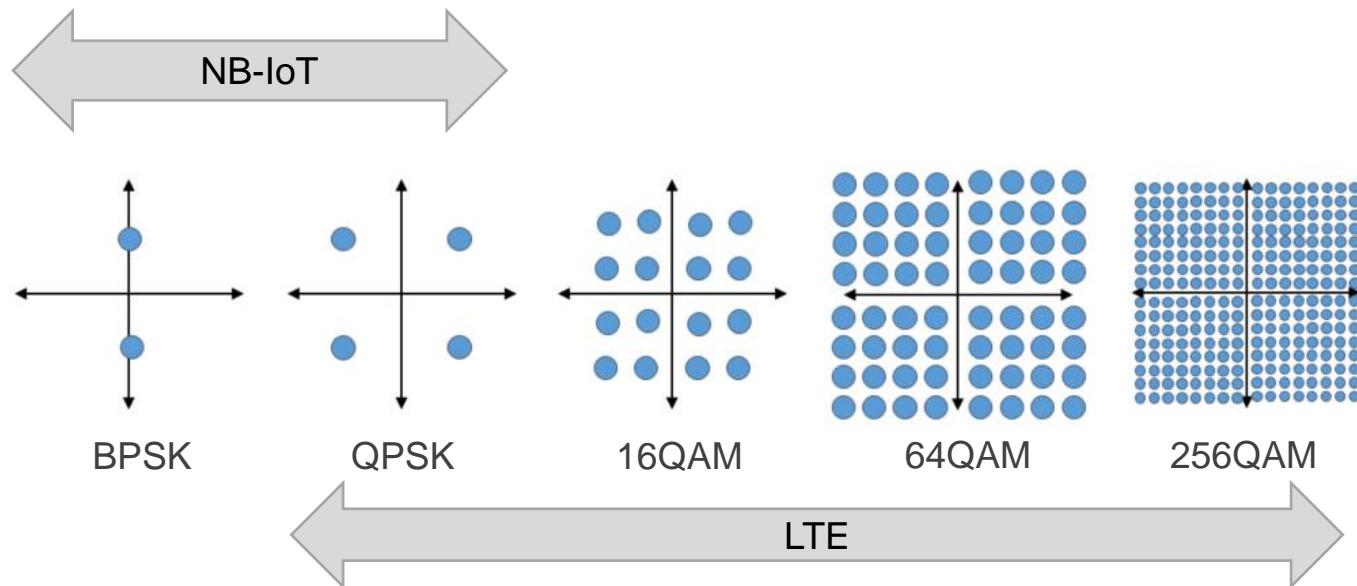
データの少量化と手順の簡素化により狭帯域で提供

無線区間の特長

- 復調方式

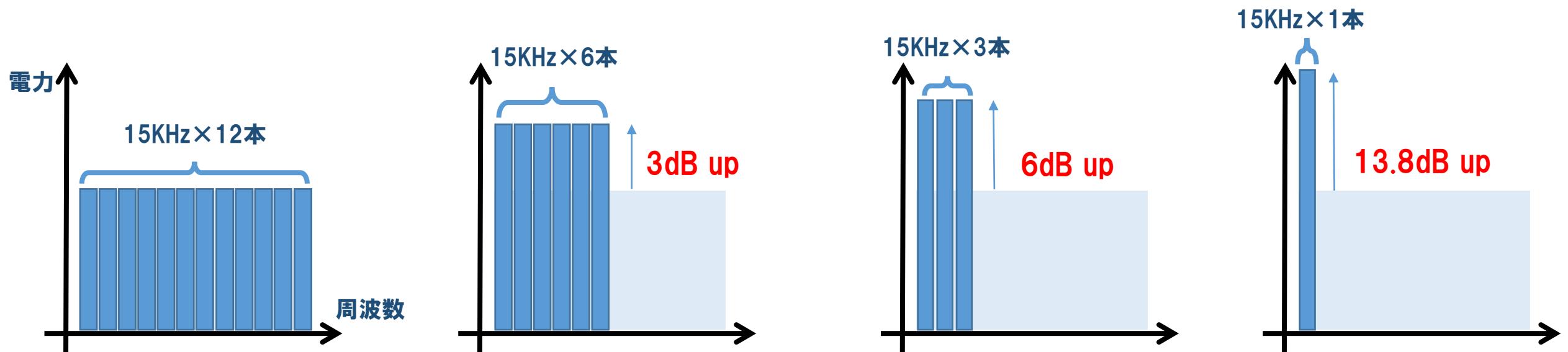


- 変調方式



データ送受信の効率化よりも
デバイスの簡易性を重視

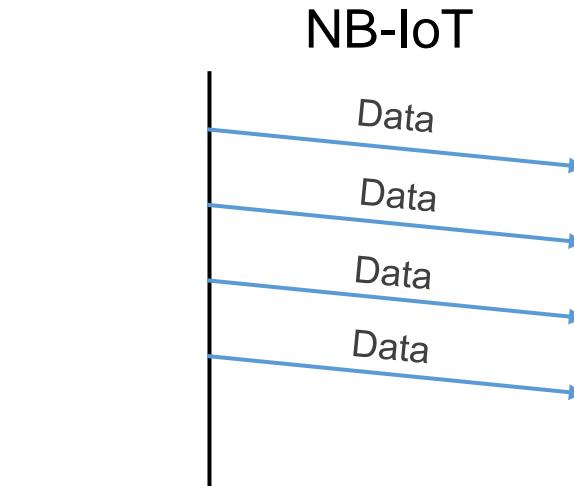
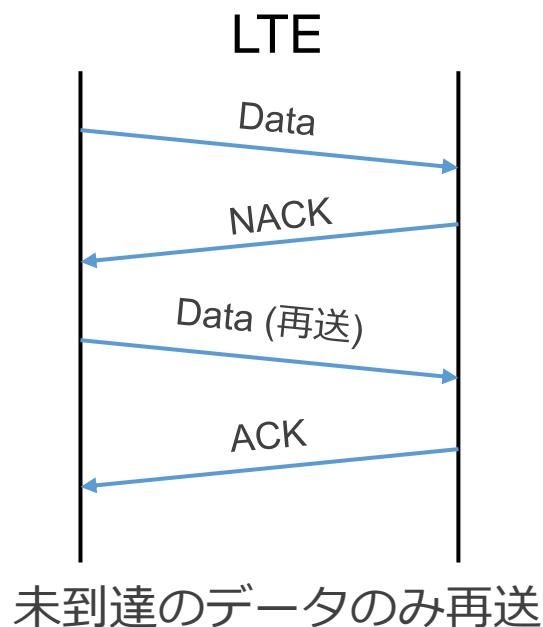
カバレッジ拡張(Multi/Single-tone)



少量データを前提に上りカバレッジを拡大

到達率の改善 (Repetition)

無線強度・品質に応じて、(応答有無に関係なく)同じデータを繰り返し送信



到達確率を高めエリアを拡大

少量データを前提に到達率を向上しカバレッジを拡大

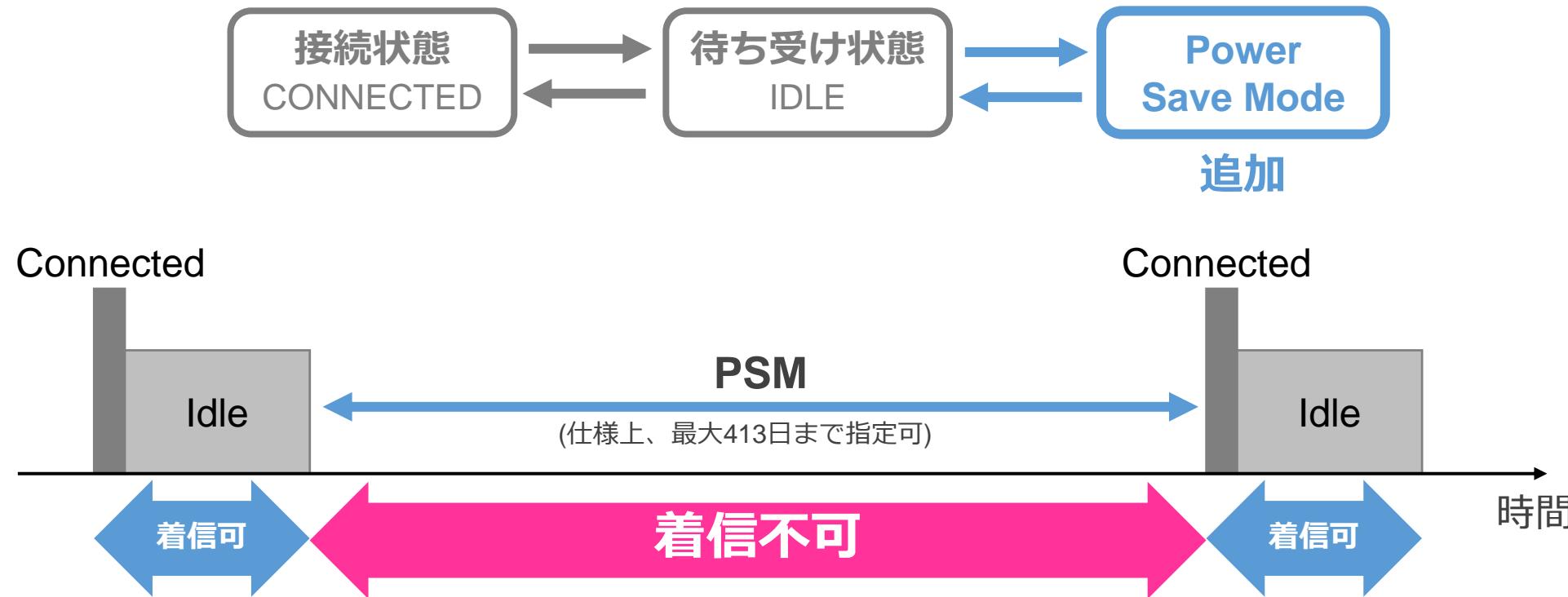
手順の簡素化



制御信号送受信の中に、データも入れて通信
少量データの特性を活かし手順を削減

省電力モード(PSM)の導入

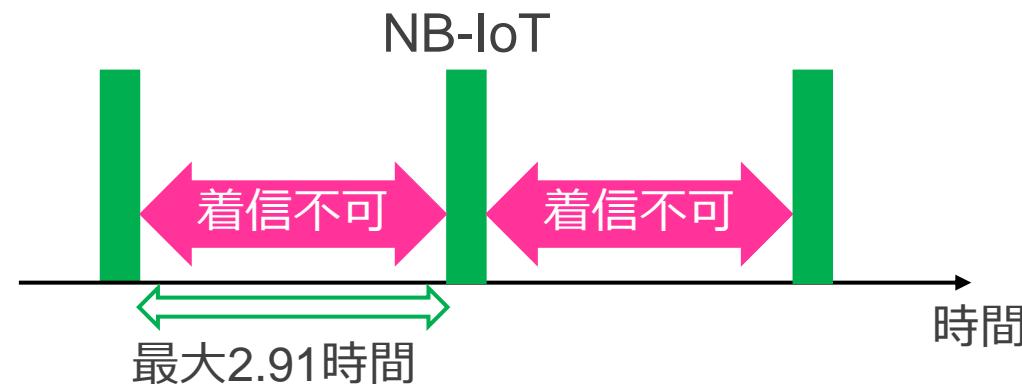
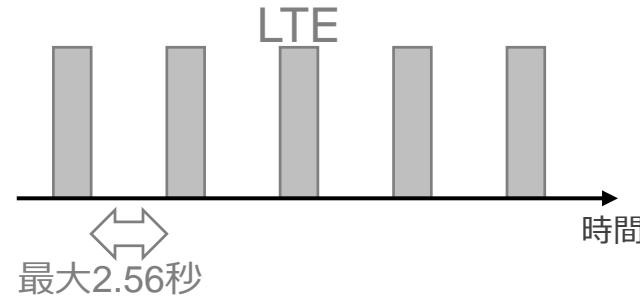
デバイスは網に登録(Registered)のまま、電源OFFと同じ状態を維持



着信不可の時間長と省電力はトレードオフ

間欠的信号受信の拡大 (eDRX)

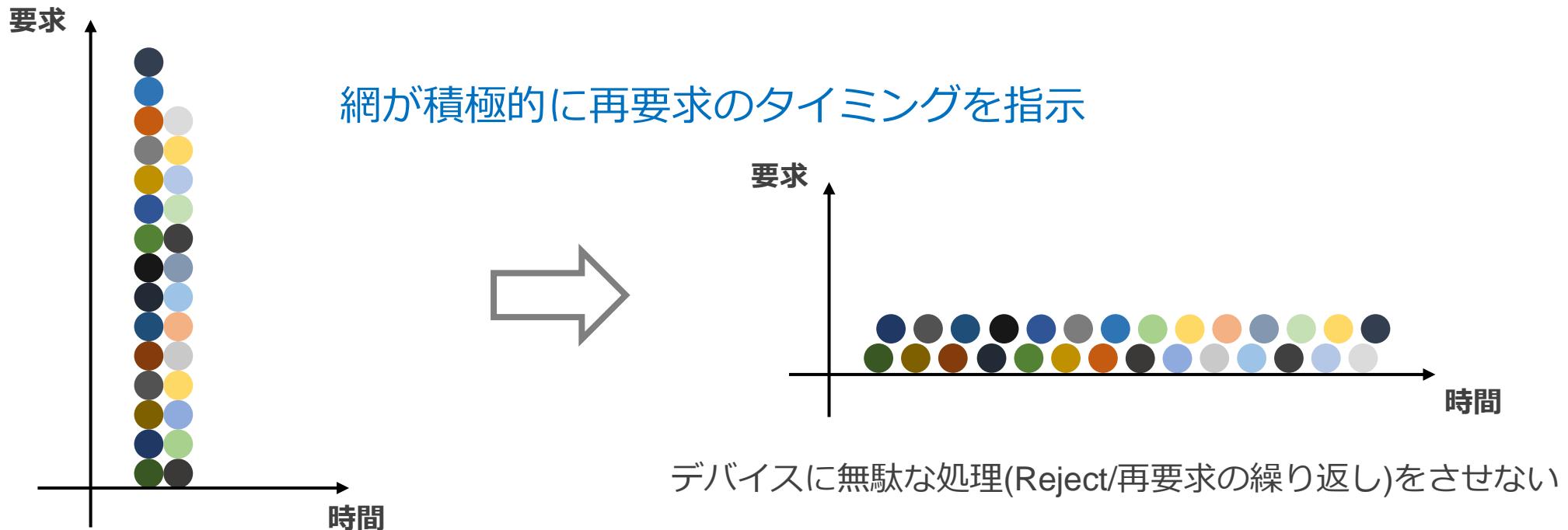
待ち受け状態(IDLE)において、信号受信確認を間欠的に行い、消費電力を抑制



eDRXによる着信不可の時間長も省電力はトレードオフ

過負荷保護機能 (back off)

同時に「何か」の要求があった時、負荷を平準化



大量のデバイスを効率よく収容
リアルタイム性/実時間性を求めるアプリケーションは注意が必要

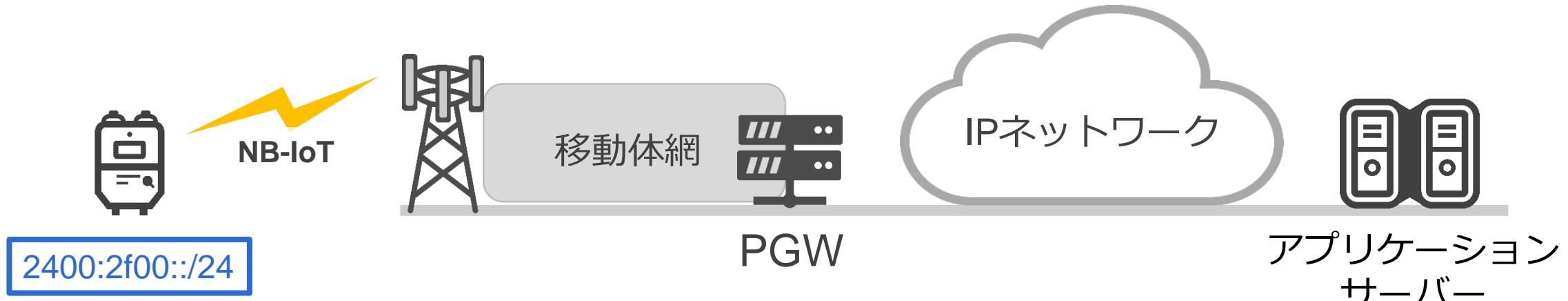
Agenda

- ✓ NB-IoTの特長
- ✓ NB-IoTの提供方法
- ✓ NIDD SCEF Based Delivery動作の概要
- ✓ NIDD SCEF Based Delivery 適用例

NB-IoTを提供する3つの方式

1. IP Data Delivery
2. Non-IP SGi based Data Delivery
3. Non-IP SCEF based Data Delivery

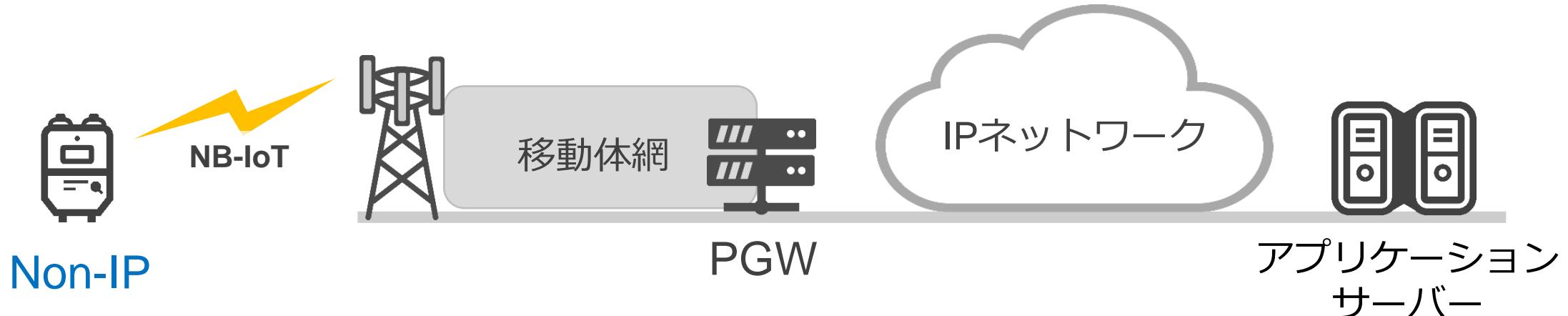
1. IP Data Delivery



- 1対多の通信が可能
- 従来のデータ端末と同じ構成

少量データを前提にしたNB-IoTでは、TCP(UDP)/IPのオーバーヘッドですら大きい
NB-IoTの特性を活かすためには、IPが無い方が望ましい

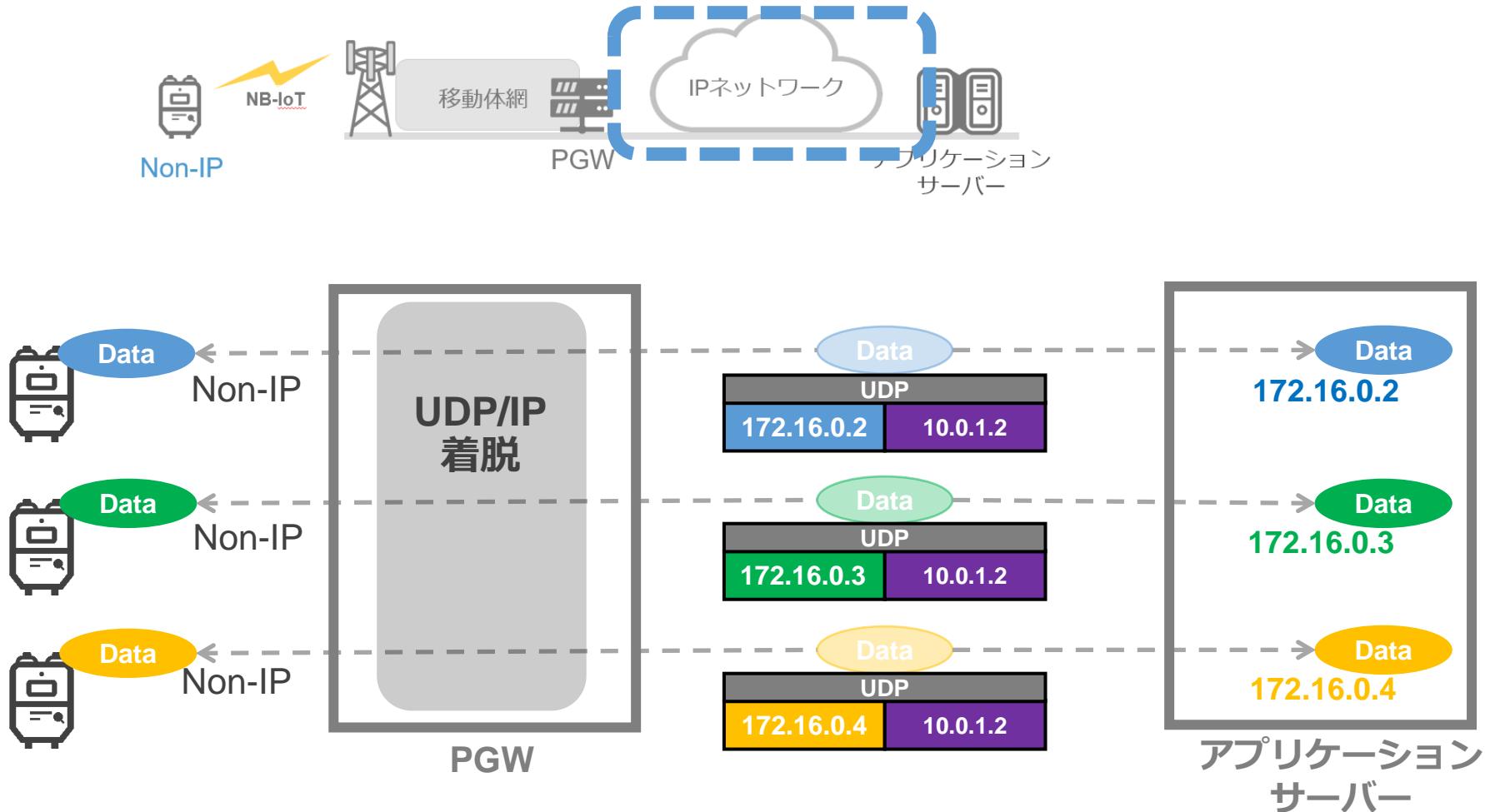
2. Non-IP SGi Based Data Delivery



- デバイスとアプリケーションサーバーの1対1の通信
通信先は、移動体網にあらかじめ設定
- デバイス毎にPGWがIPアドレス^(注)を割り当て
アプリケーションサーバーは、IPアドレスでデバイスを識別
- 移動体網内もIPネットワークもUDPでデータ転送

(注) GREのTunnel IDなど、トンネリングプロトコルでも可能

2. Non-IP SGi Based Data Delivery



3. Non-IP SCEF Based Data Delivery

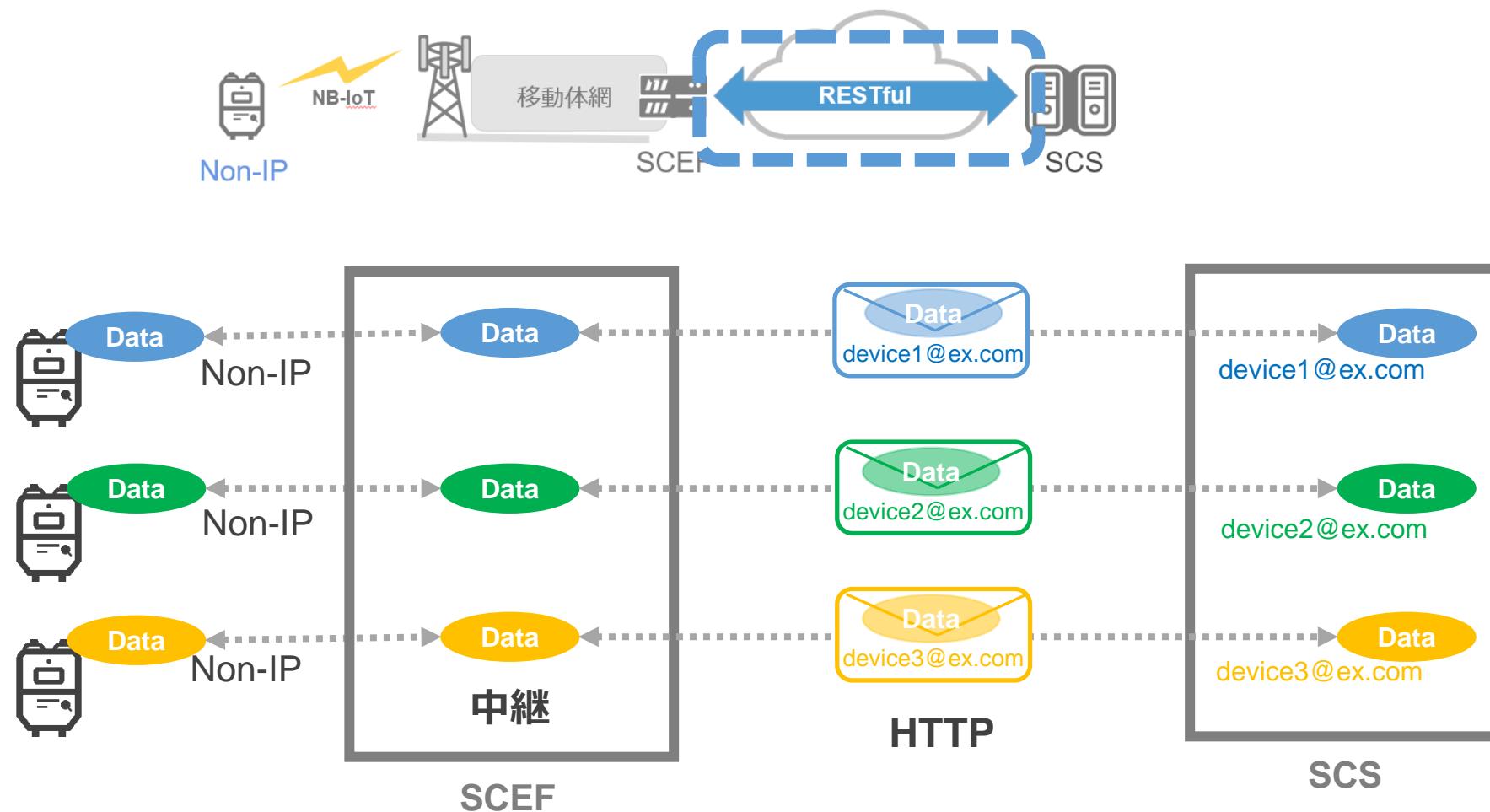


- デバイスとアプリケーションサーバーの1対1の通信
通信先は、移動体網にあらかじめ設定
- デバイス毎にNAI形式の識別子(External-ID)^(注)をあらかじめ設定
アプリケーションサーバーは、External-IDでデバイスを識別
- 移動体網内はSCTPで、IPネットワークはTCPで、データ転送

(注) 仕様上はMSISDNでも可

SCEF: Service Capability Exposure Function
SCS: Service Capability Server

3. Non-IP SCEF Based Data Delivery



Non-IP Data Delivery: 2方式の比較

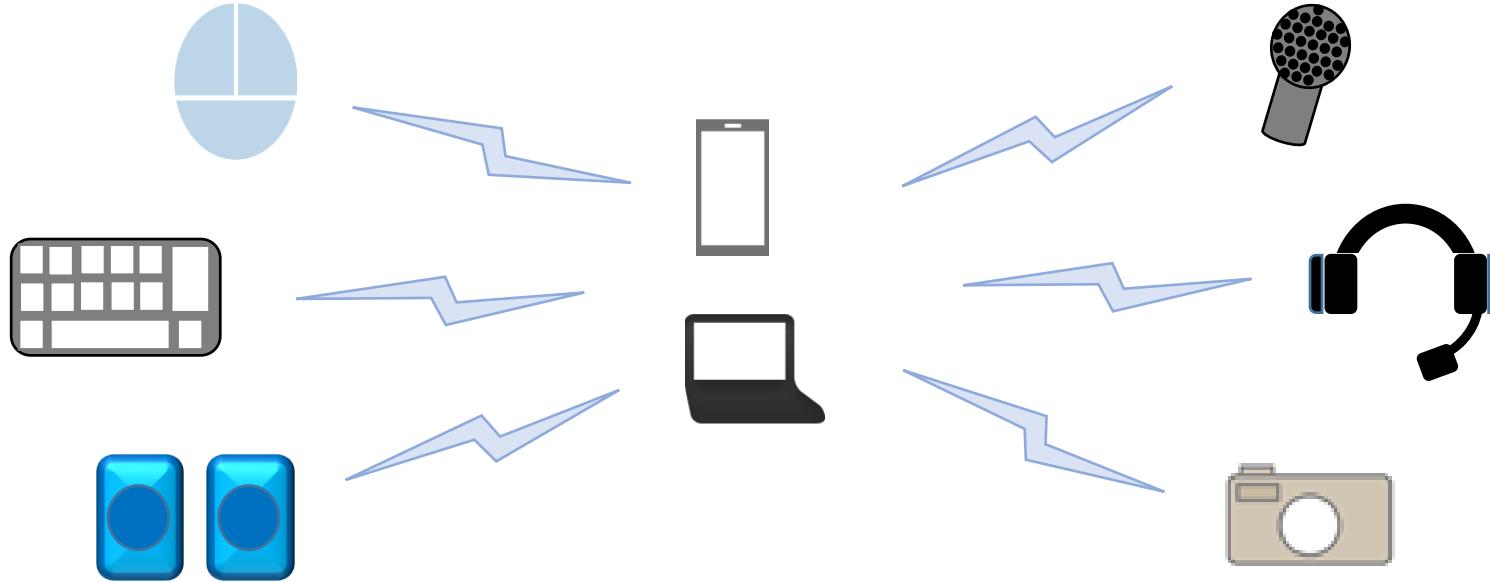
方式	SGi Based	SCEF Based
アプリケーションサーバーにおけるデバイスの識別	IPアドレス	External-ID (NAI形式)
データ転送	UDP 送達はアプリで確認	TCP/SCTP SCEFのAPIで確認
デバイスのセキュリティの確保	IPレベル	HTTPレベル

- SGi Based
従来の移動体データ通信に近い
- SCEF Based
ITとの親和性が高い



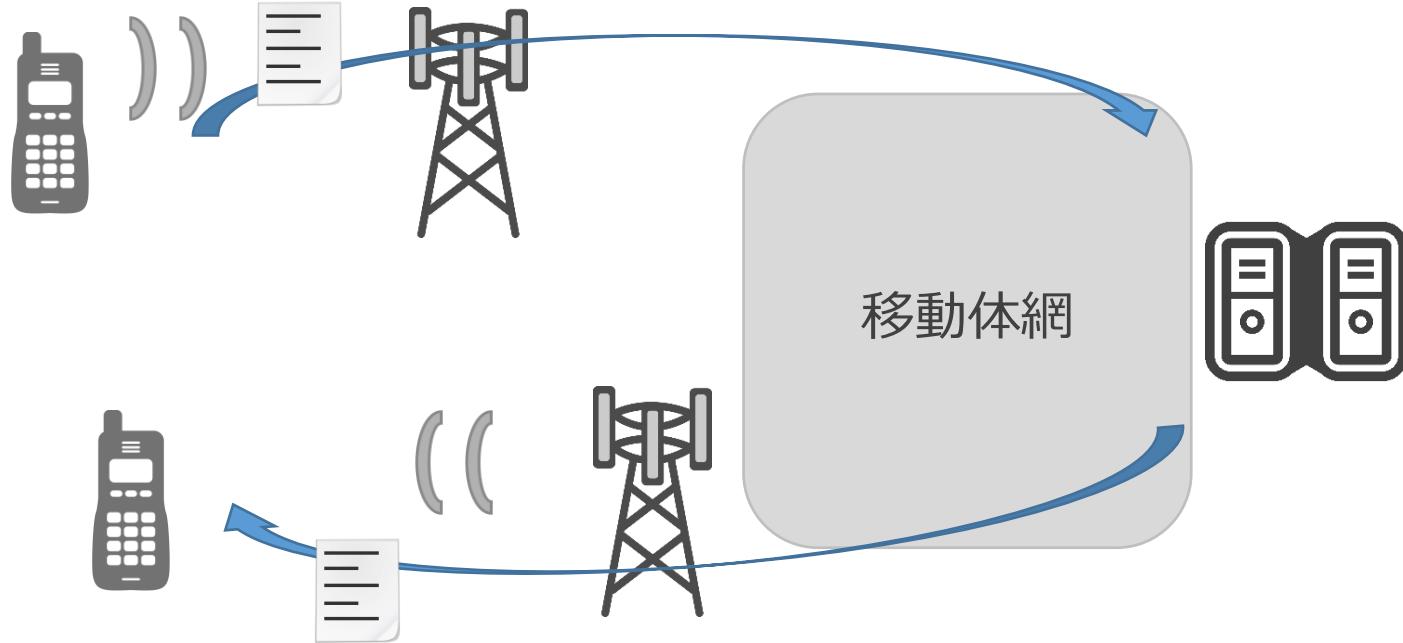
Non-IPって特別なこと？

IPを持たない通信



1対1の通信であれば、IPはいらない
用途を限定すれば、構造はシンプル

温故知新？ SMS

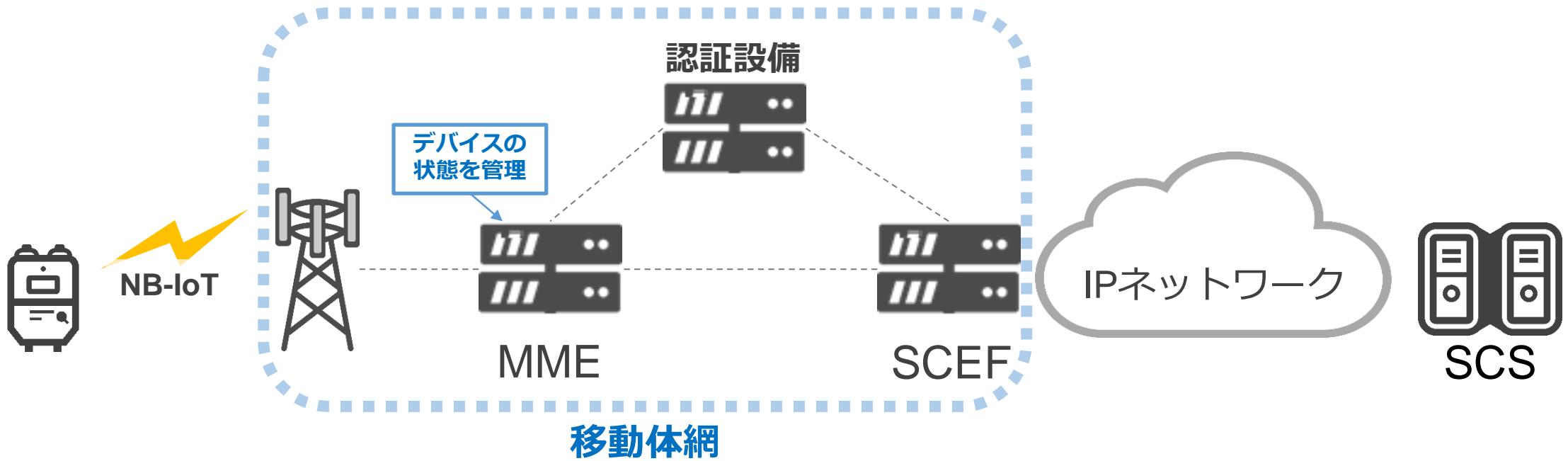


『誰から』 『誰へ』 がわかっていれば、
IPはいらない

Agenda

- ✓ NB-IoTの特長
- ✓ NB-IoTの提供方法
- ✓ NIDD SCEF Based Delivery動作の概要
- ✓ NIDD SCEF Based Delivery 適用例

構成概要

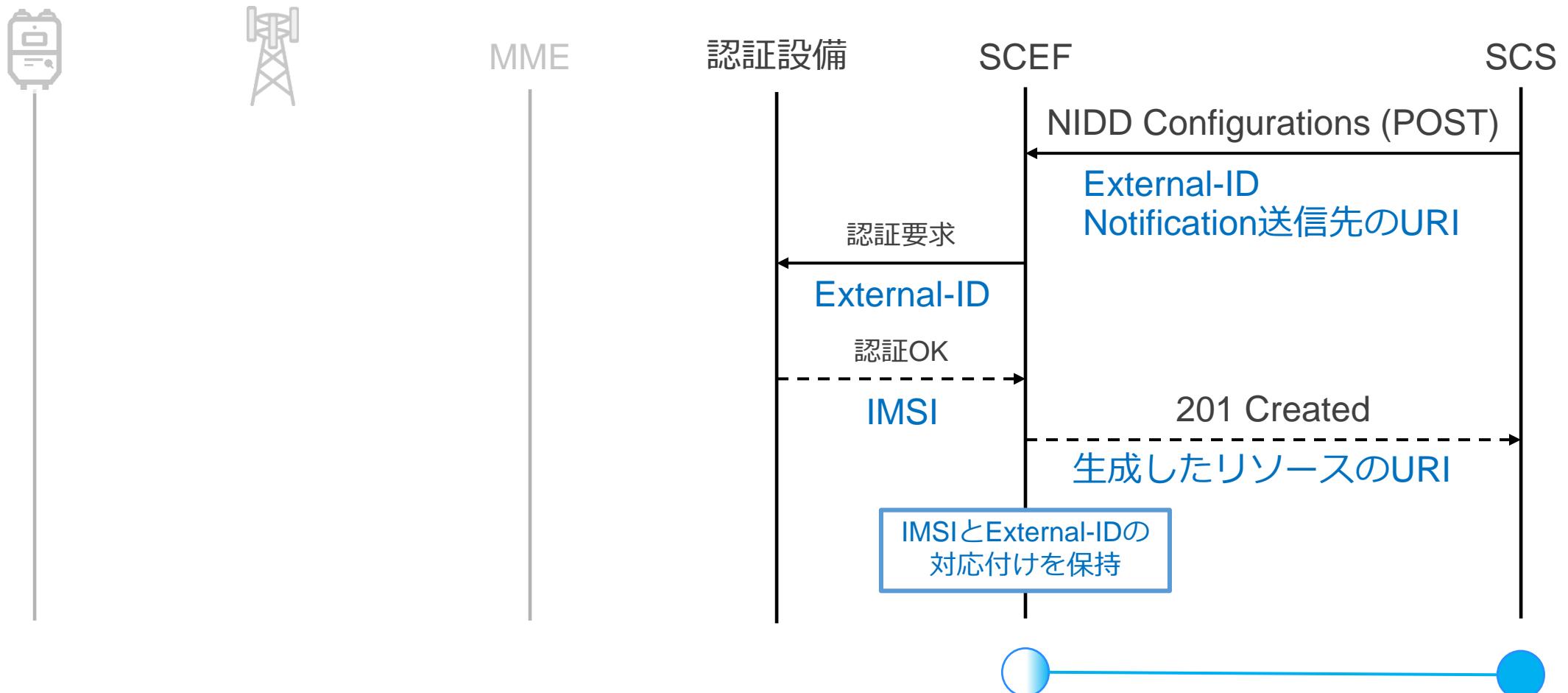


- 移動体網内は、C-Plane (通信を制御するための信号)のみを使用
従来の通信では、U-Plane (利用者が送受信するデータ)を利用

MME: Mobility Management Entity

通信手順: リソース生成

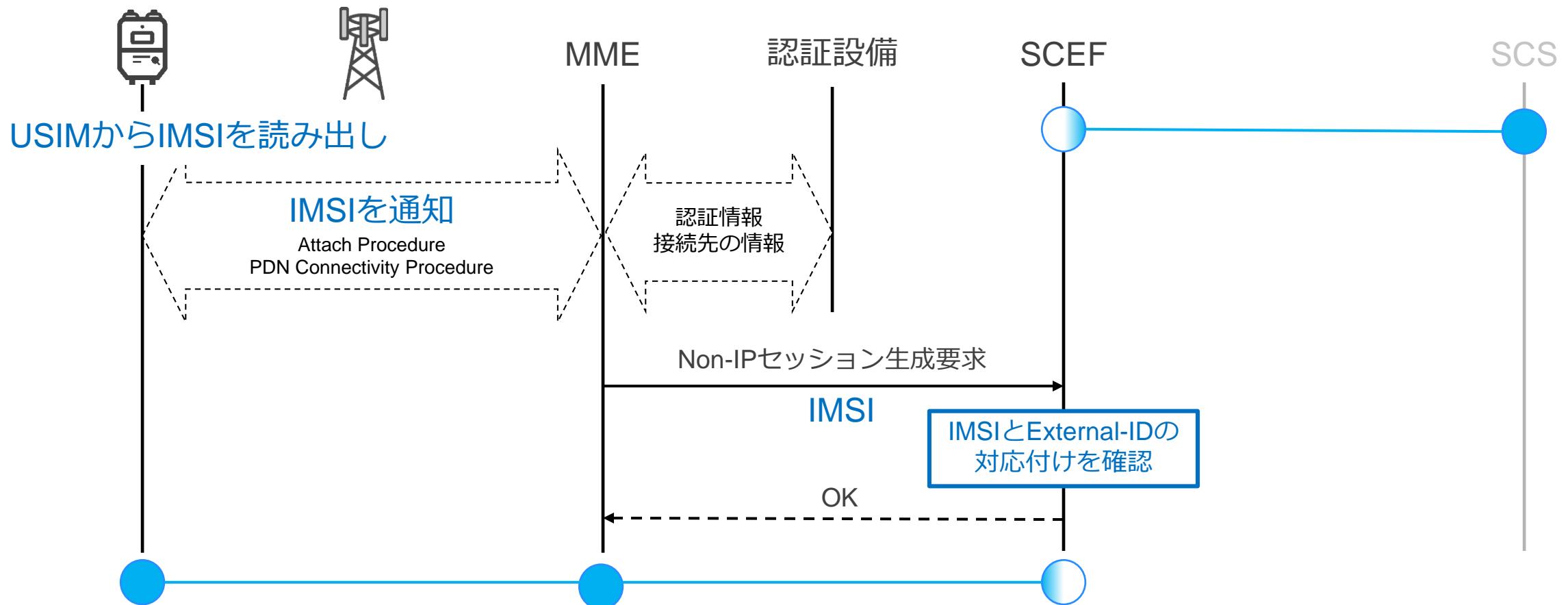
SCSはデバイス単位にリソースを生成



IMSI: International Mobile Subscriber Identity

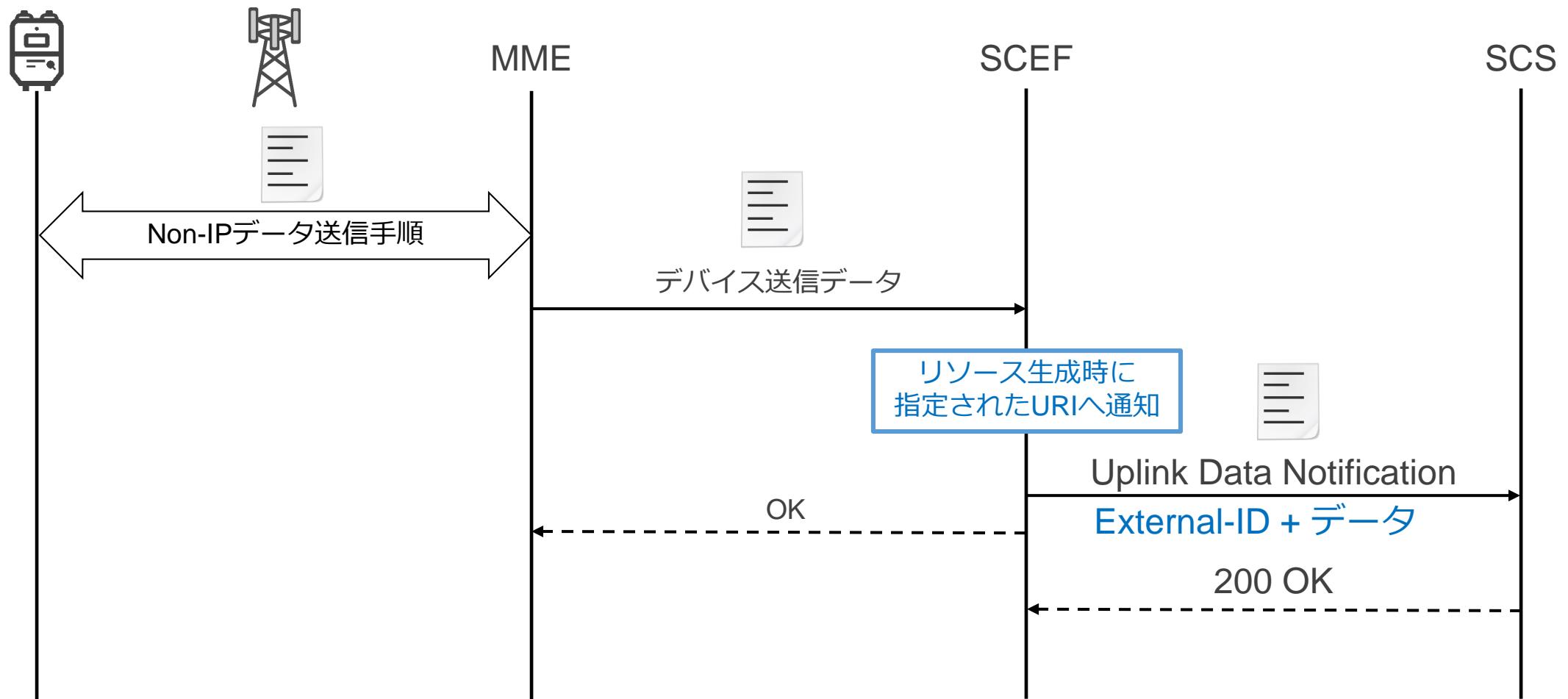
通信手順: デバイスの接続

移動体網内はIMSIによりデバイスを管理



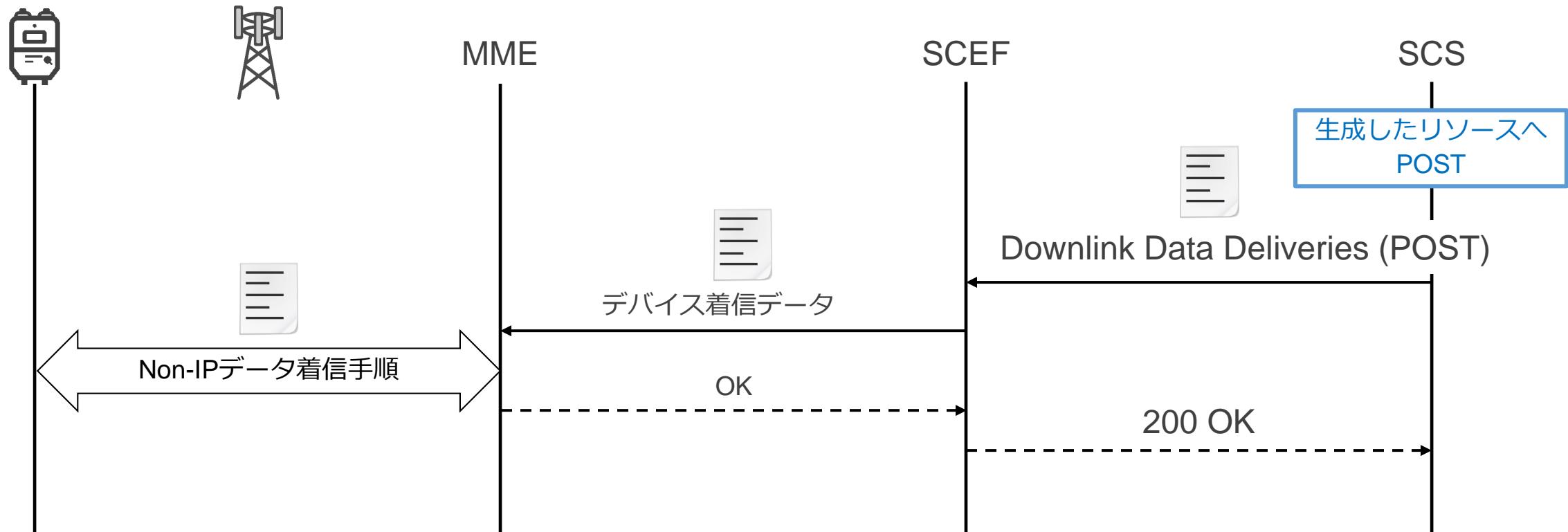
- デバイスはExternal-IDを知らない。知る必要もない。
USIMから読み出す情報のみで、通信することも可能

通信手順: デバイスからのデータ送信



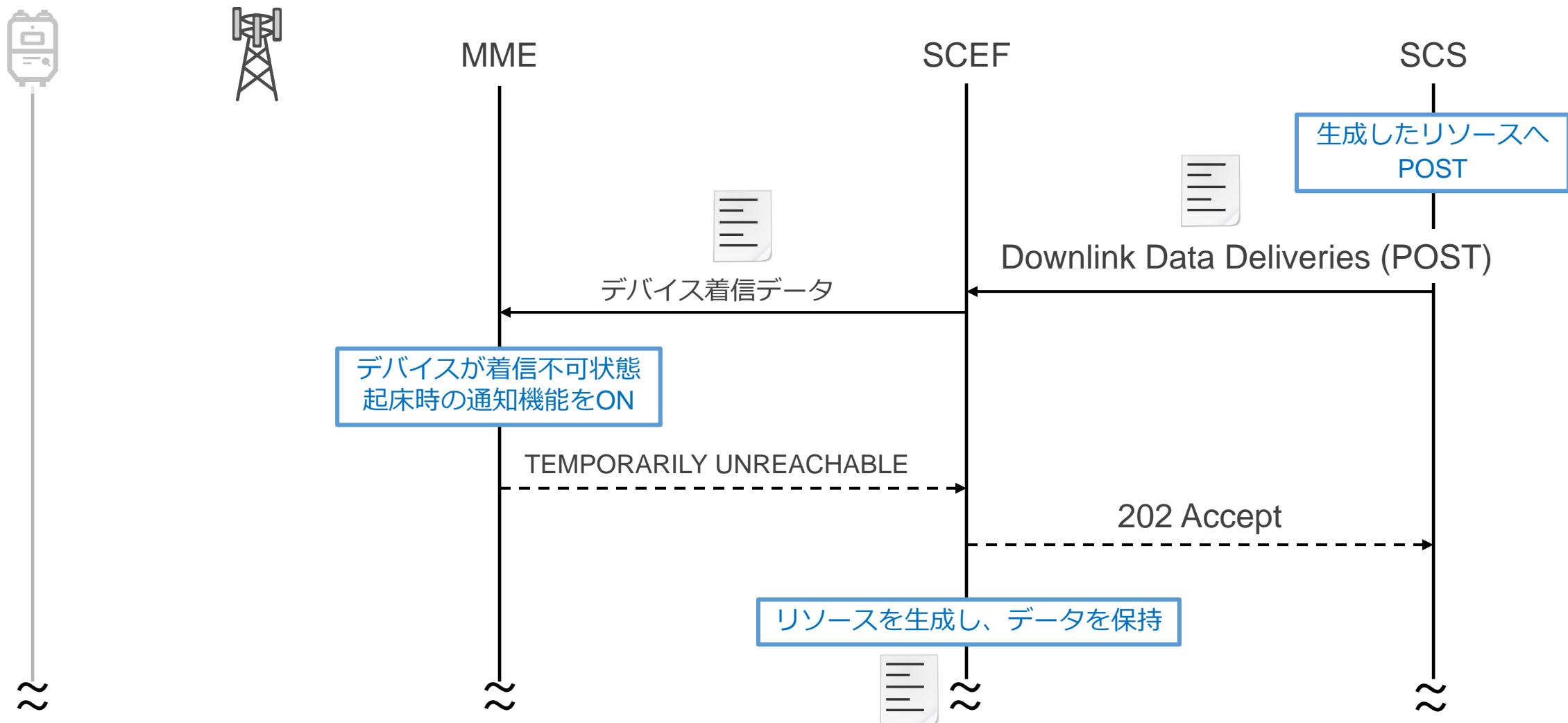
通信手順: デバイスへのデータ送信

デバイスがCONNECTEDの場合



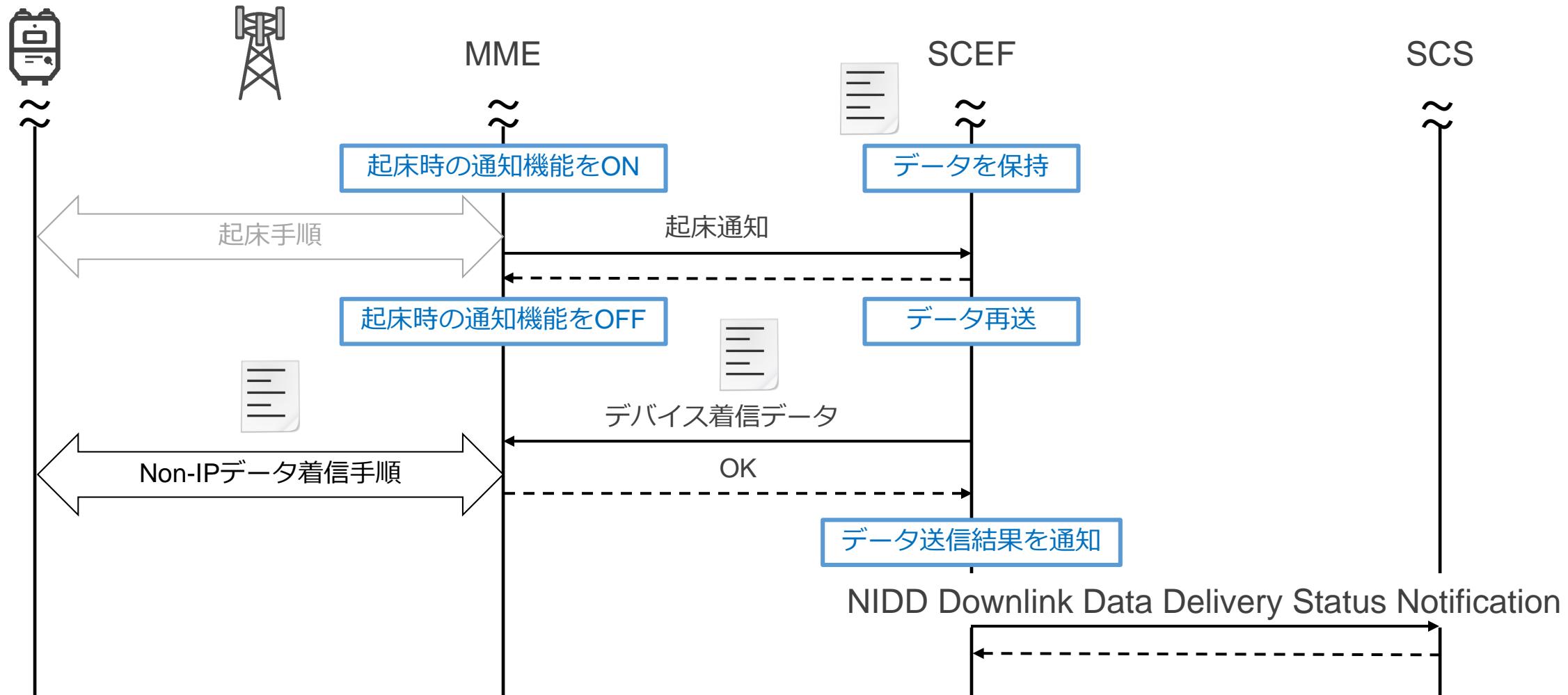
通信手順: デバイスへのデータ送信

デバイスがすぐに着信できない場合



通信手順: デバイスへのデータ送信

デバイスが着信不可から着信可に遷移



SCEF-SCSの仕様 (T8参照点)

2018年6月

3GPP標準 初版がリリース (TS29.122)

OpenAPI3.0準拠

yamlは3GPPのサイトで公開



Swagger Editor / swagger-editor-3.5.5/

(+) Swagger Editor File → Edit → Generate Server → Generate Client →

```
1: openapi: 3.0.0
2: info:
3:   title: 3gpp-nidd
4:   version: v1
5:   externalDocs:
6:     description: Documentation
7:     url: 'http://www.3gpp.org/ftp/Specs/archive/29_series/29.122/'
8: servers:
9:   - url: "https://{{apiRoot}}/3gpp-nidd/v1"
10:  variables:
11:    apiloot:
12:      default: demohost.com
13:      description: 'apiloot as defined in subclause 5.2.4 of 3GPP TS 29.122, excluding the https:// part.'
14:      type: string
15: paths:
16:   '/[scsAsId]/configurations':
17:     get:
18:       parameters:
19:         - name: scsAsId
20:           description: String identifying the SCS/AS.
21:           in: path
22:           required: true
23:           schema:
24:             type: string
25:       responses:
26:         '200':
27:           description: all NIDD configurations.
28:           content:
29:             application/json:
30:               schema:
31:                 type: array
32:                 items:
33:                   $ref: '#/components/schemas/NiddConfiguration'
34:                   minItems: 0
35:               description: Individual NIDD configuration.
36:         default:
37:           description: Unexpected error
38:           content:
39:             application/problem+json:
40:               schema:
41:                 $ref: '#/components/schemas/ProblemDetails'
42:     post:
43:       parameters:
44:         - name: scsAsId
45:           description: String identifying the SCS/AS.
46:           in: path
47:           required: true
48:           schema:
49:             type: string
```

3gpp-nidd v1 (660)

Documentation

Server: https://{{apiRoot}}/3gpp-nidd/v1

Computed URL: https://demohost.com/3gpp-nidd/v1

Server variables: apiloot demohost.com

default

GET /[scsAsId]/configurations

POST /[scsAsId]/configurations

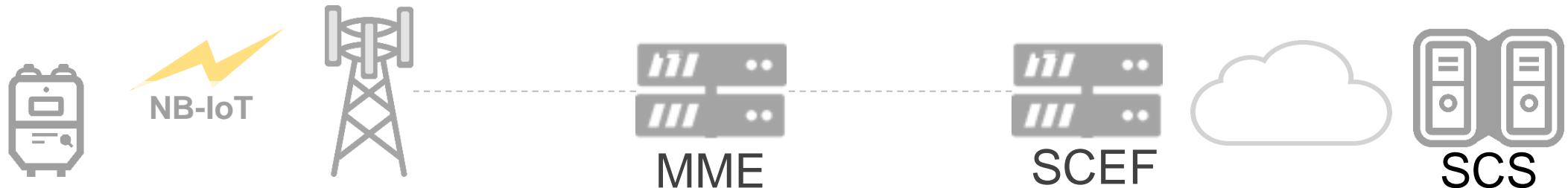
GET /[scsAsId]/configurations/[configurationId]

PATCH /[scsAsId]/configurations/[configurationId]

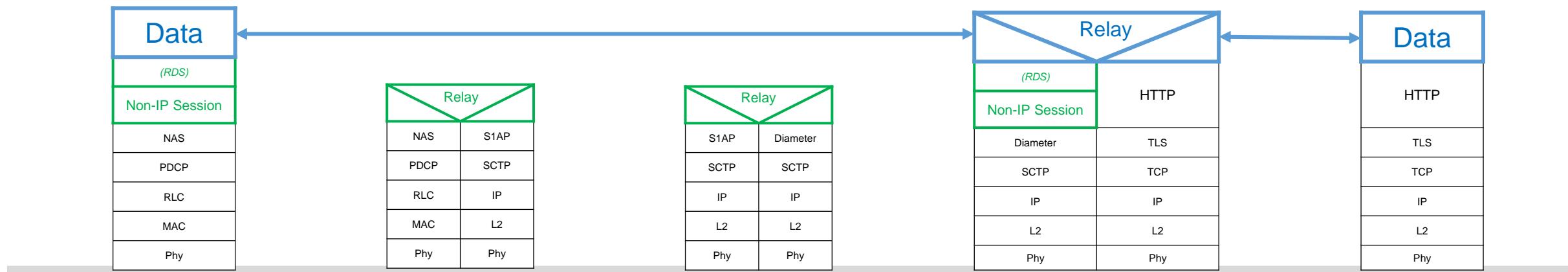
Agenda

- ✓ NB-IoTの特長
- ✓ NB-IoTの提供方法
- ✓ NIDD SCEF Based Delivery動作の概要
- ✓ NIDD SCEF Based Delivery 適用例
-SoftBank実証実験の構成を例に-

プロトコルスタック



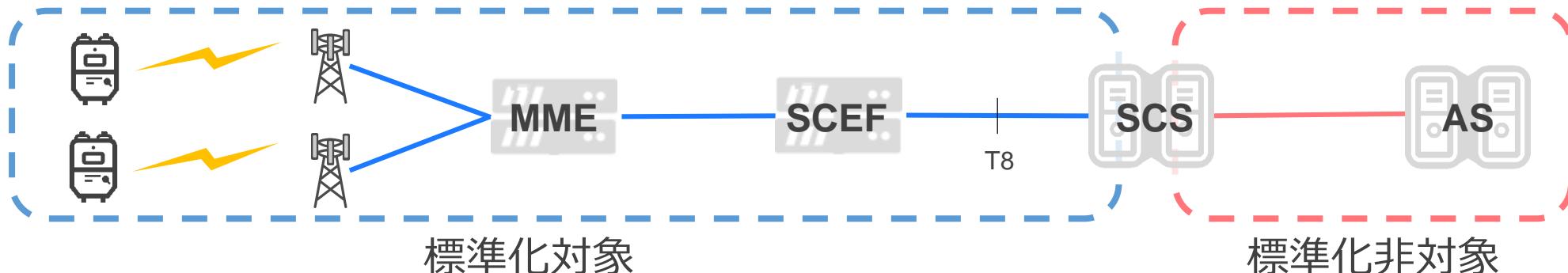
バイナリデータを透過に転送



- デバイスとSCSでプロトコルの『決め』が必要

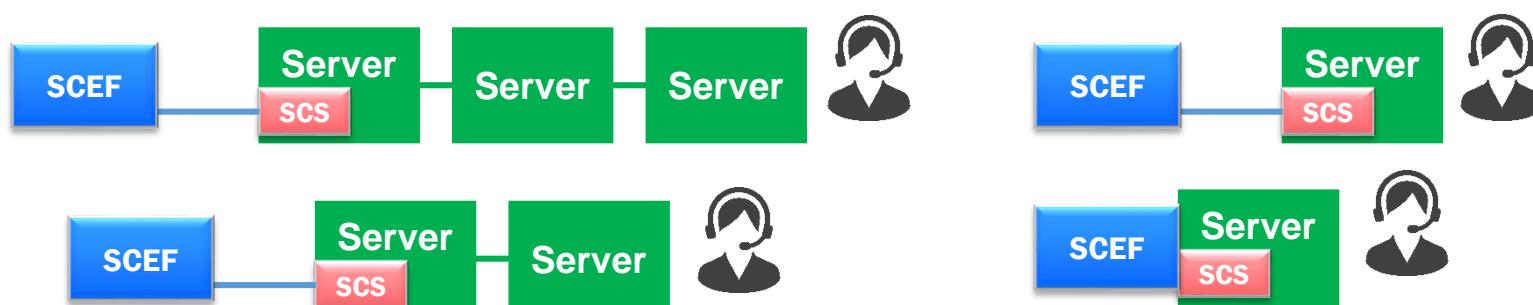
SCSとは

3GPPで定義された構成



SCSとASの違いは？

SCSはT8によりSCEFと通信する機能
それ以外、SCSとASを区別する定義はない(区別する必要もない)



実装依存で、様々な構成が可能

AS: Application Server

ご清聴、ありがとうございました。