SORACOM Technology Camp 2020

改めて双方向通信について考えよう！

～リモートアクセスのパターンとその実践～

株式会社ソラコム

2020/2/18

本日のハッシュタグ

SORACOM Technology Camp 2020

#soracom

@SORACOM_PR

https://www.facebook.com/soracom.jp/

セッションで扱うトピック

SORACOM Technology Camp 2020

- 本セッションでは以下のキーワードでお話をします。
  双方向通信/遠隔監視/遠隔制御/M2M連携

- 遠隔監視や遠隔制御、デバイス間通信といったサーバ – デバイス間、デバイス – デバイス間の通信を含めて双方向通信として扱います。

セッションのながれ

SORACOM Technology Camp 2020

- 双方向通信が必要となるシーン
- 双方向通信における考慮ポイント
- 双方向通信の実現デザインパターンと実践
- デザインパターンまとめ

自己紹介

SORACOM Technology Camp 2020

須田桂伍(すだ けいご - kei)

株式会社ソラコム

ソリューションアーキテクト

セッションのながれ

SORACOM Technology Camp 2020

- 双方向通信が必要となるシーン
- 双方向通信における考慮ポイント
- 双方向通信の実現デザインパターンと実践
- デザインパターンまとめ

双方向通信とは

SORACOM Technology Camp 2020

モノ　インターネット　クラウド

双方向通信とは

SORACOM Technology Camp 2020

モノ　　インターネット　　クラウド

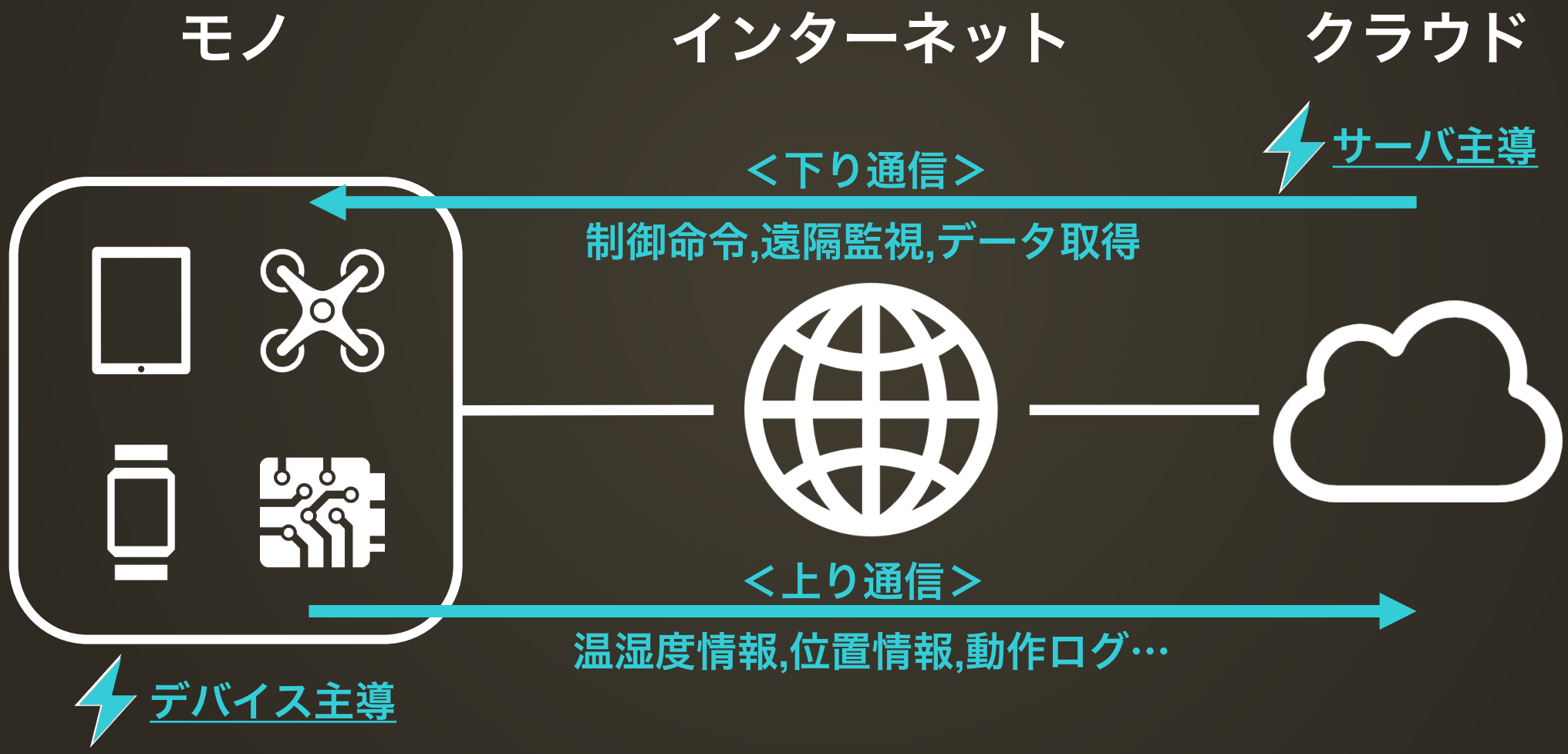
＜上り通信＞

温湿度情報,位置情報,動作ログ…

双方向通信とは

モノ　　インターネット　　クラウド

双方向通信とは


モノ
インターネット
クラウド
サーバ主導
＜下り通信＞
制御命令,遠隔監視,データ取得
＜上り通信＞
温湿度情報,位置情報,動作ログ…
デバイス主導

代表的な双方向通信

SORACOM Technology Camp 2020

- **遠隔制御**
  - 遠隔からデバイスに対して更新系の処理を実施
- **遠隔監視**
  - 遠隔からデバイスの状態や死活情報を取得/確認
- **M2M連携**
  - デバイス間,デバイス-サーバ間で自律的な連携をおこなう

セッションのながれ

SORACOM Technology Camp 2020

- 双方向通信が必要となるシーン
- **双方向通信における考慮ポイント**
- 双方向通信の実現デザインパターンと実践
- デザインパターンまとめ

本当にその通信必要ですか？

まず検討したいこと

SORACOM Technology Camp 2020

- 双方向通信は単純なデータ送信と比べ考慮することがとても多く実装も複雑になっていく

- **要件そのもの、設計そのものを簡素にできないか**をまずは検討することが重要
  - 通信要件自体を見直せないか
  - デバイス主導の通信として見直せないか

考慮ポイント – 物理面

SORACOM Technology Camp 2020

- デバイス
  - どういったデバイスが対象か(Linuxベース?RTOS?)
- 電源
  - 稼働時の電源確保(常時供給可能?バッテリー駆動?)
- 台数
  - 対象とするデバイスの台数(数台?数百~数千?数万!?)
- 制御内容
  - どういった制御命令を出す?(冪等?確実に1回?)
- セキュリティ
  - アクセス経路や制御情報は安全か？

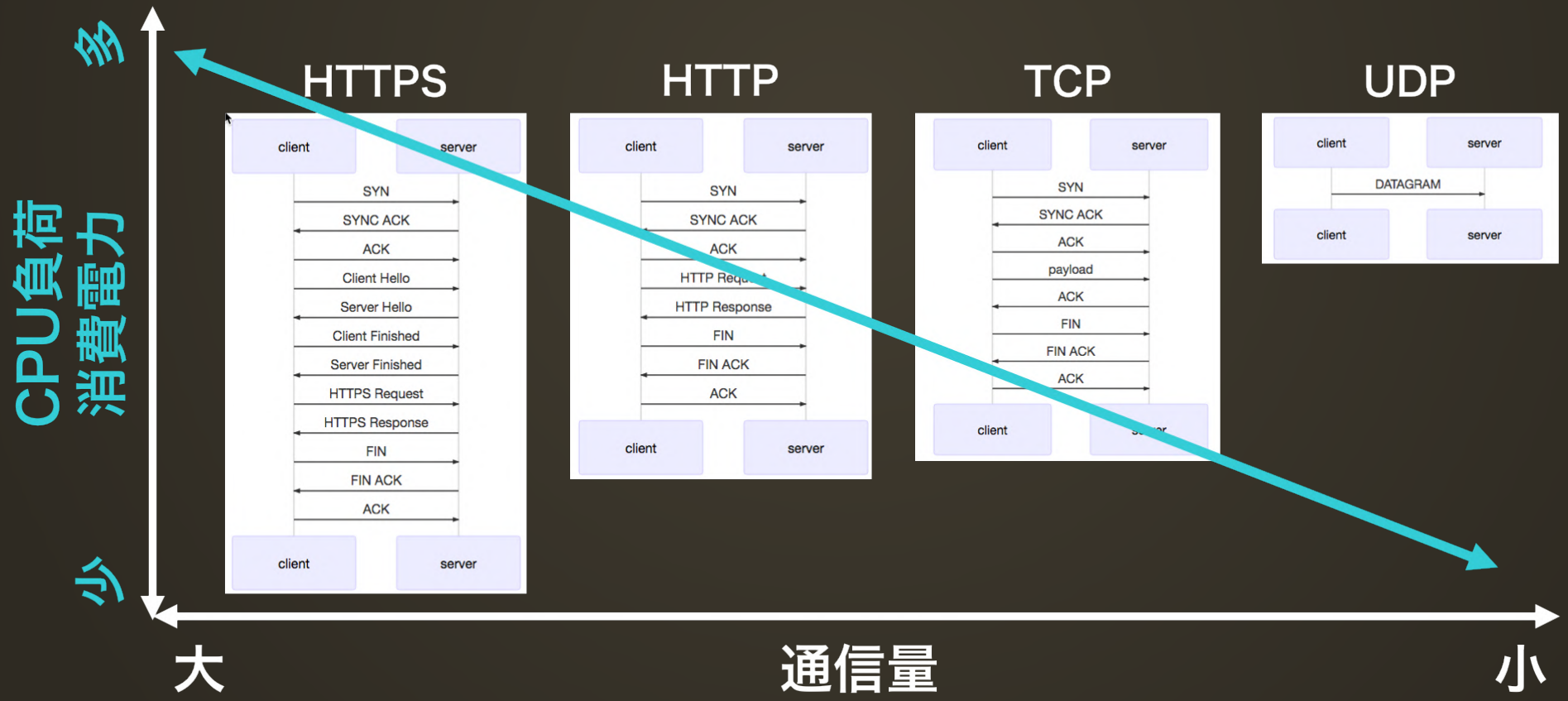
消費電力と双方向通信

SORACOM Technology Camp 2020

- デバイスがバッテリー駆動の場合、待機時及び処理時の消費電電力をいかに減らすかが重要
  - 各モジュールをディープスリープ(MCU/通信モジュール/接続センサ)
  - データ送信時に起動し処理後に再度スリープ
- スリープ中は外からのリクエストを受け付けられないため、即時制御できない
- 双方向通信が求められるシーンでは消費電力と双方向通信の要件はトレードオフまたはそもそも相性が悪い

消費電力と双方向通信

連携時に利用するプロトコル選択も影響が大きい。
アプリケーションプロトコルの機能が豊富なものほど実装の柔軟性は増すものの消費電力は大きくなる



考慮ポイント – アプリケーション

SORACOM Technology Camp 2020

- **データ到達保証**
  - データ到達をどこまで保証するか?
- **リトライ**
  - データ送信失敗時のリトライ設計をどうするか?
- **順序制御**
  - 連携データの順序制御が必要か?
- **重複排除**
  - 連携データの重複排除が必要か?
- **冪等処理**
  - 連携処理は冪等になっているのか?

双方向通信とエラー処理

SORACOM Technology Camp 2020

- IoTシステムでは多くの接続ポイントが存在
  - ネットワーク障害、デバイス故障、サーバ障害、プロセス障害・・
- リトライ設計やデータ到達保証レベル設計は通常のシステム設計と同様にとても重要
- 特にデバイスへの制御を伴う双方向通信において、単純にリトライ処理をおこなうと多重起動のリスクも伴うため設計時の考慮は必須！
- 作り込み自体に制約があるケースも多い
  - 既成品の場合、デバイス側の追加での設計/実装には限界がある(そもそもできないことも多い)

双方向通信とエラー処理

SORACOM Technology Camp 2020

- ロジックはサーバサイドへ配置しデバイスへ制御が到達する前に制御内容をハンドリング

- あえて同期通信/密結合な連携
  - 同期通信で連携させることでリトライをやりやすくする

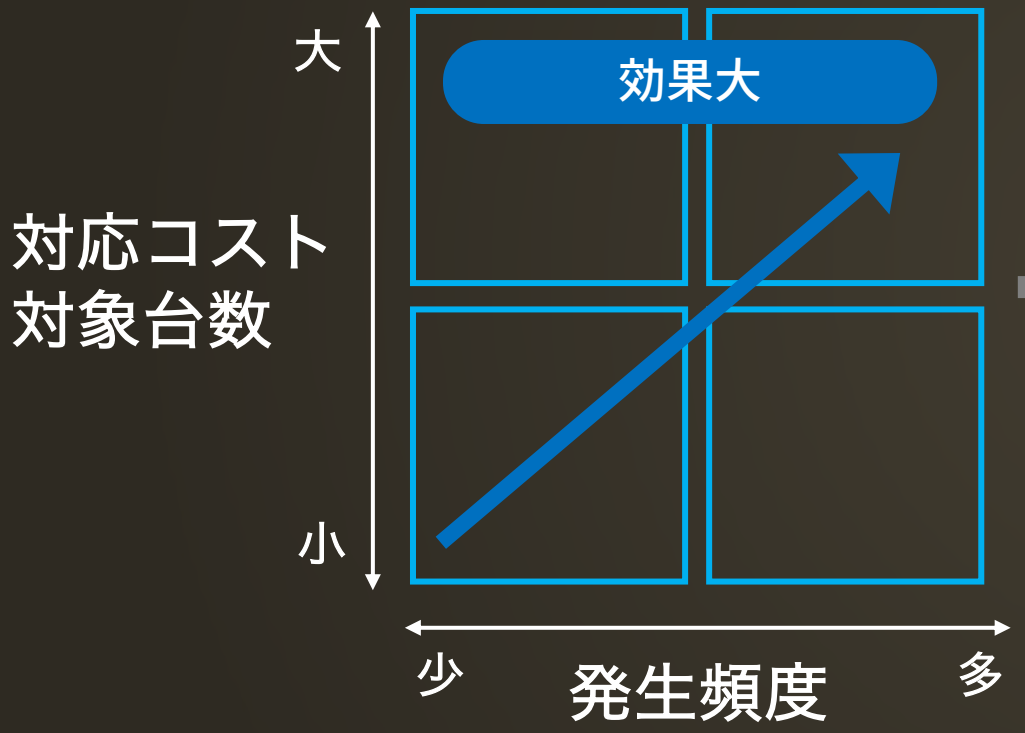
- リクエストを識別し処理済みリクエストを管理
  - リクエスト毎に一意のIDを付与し個別にステータスを管理

- 制御処理自体を冪等な実装にする
  - デバイス側での処理を冪等に設計しておく

最適なスポットをさぐる

SORACOM Technology Camp 2020

オンサイト対応

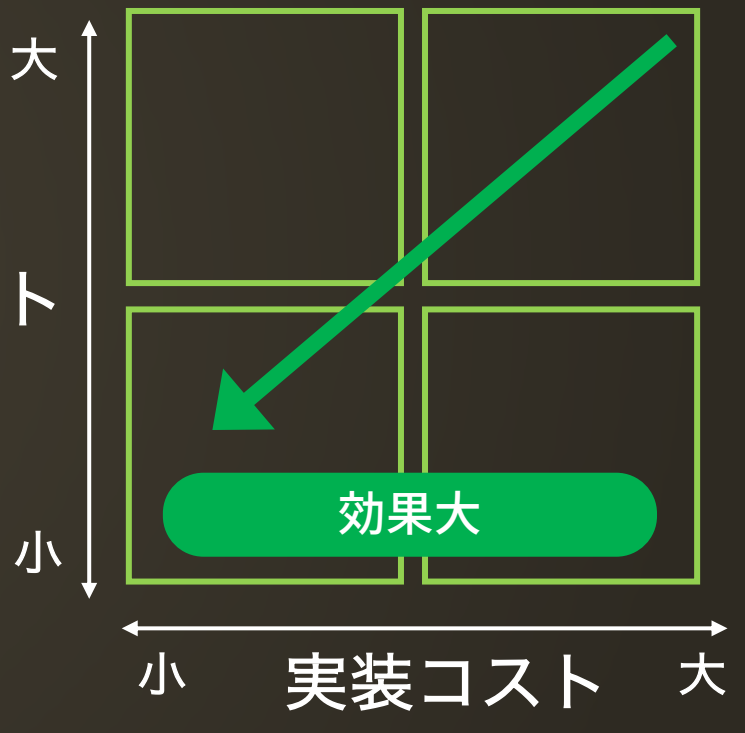


一回当たりの対応コスト × 対象 × 発生確率

例) 出張費用　設置先　年間回数

さらに

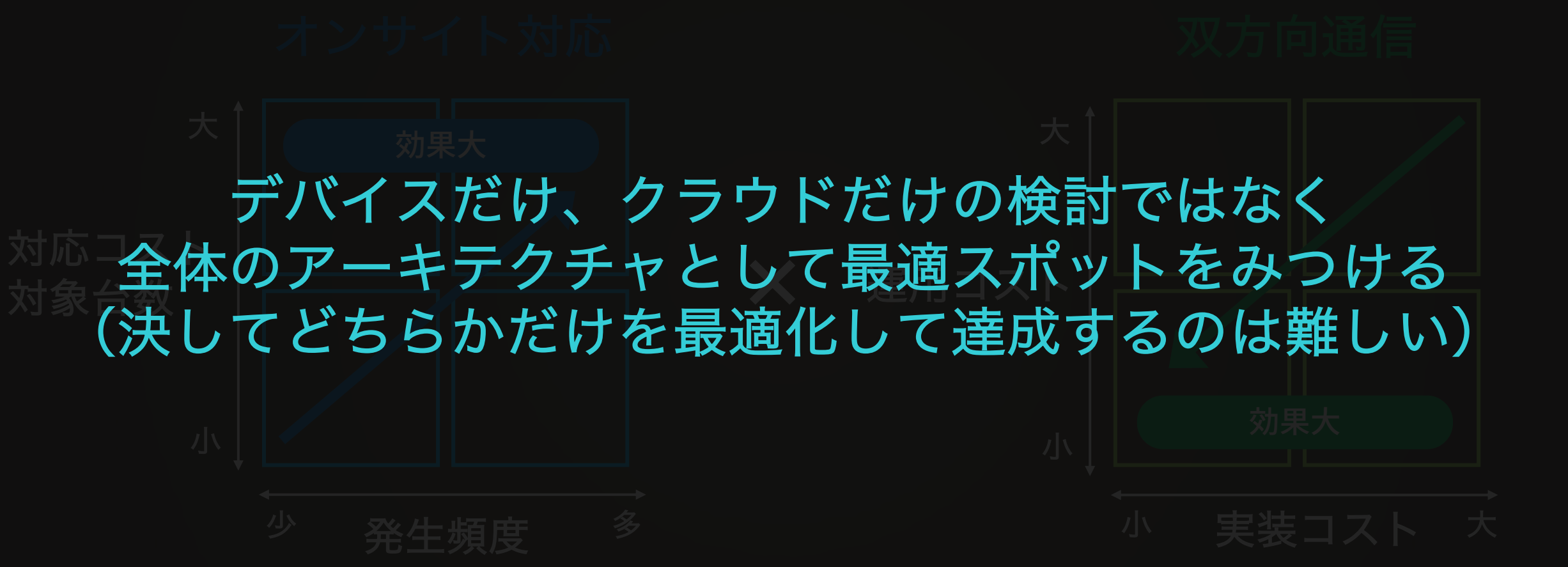
双方向通信



実装コスト + 運用コスト

例) 難易度　維持管理

スイートスポットをさぐる



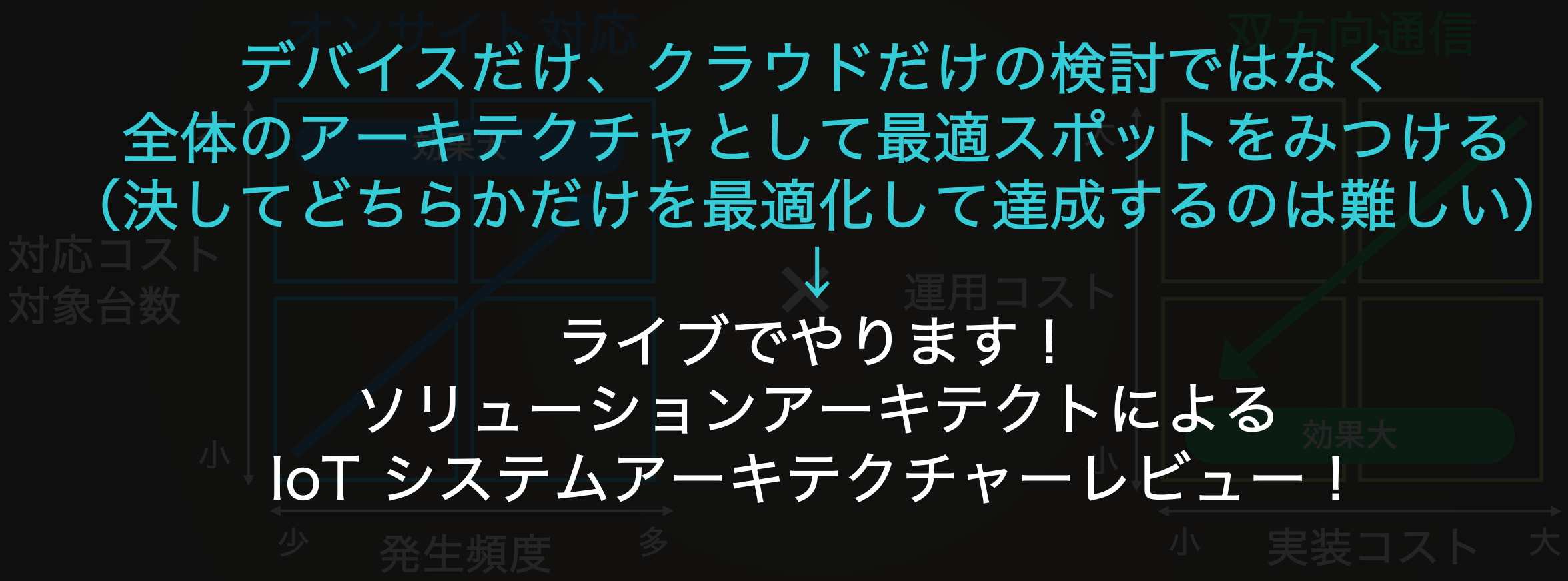
デバイスだけ、クラウドだけの検討ではなく
全体のアーキテクチャとして最適スポットをみつける
（決してどちらかだけを最適化して達成するのは難しい）

スイートスポットをさぐる

デバイスだけ、クラウドだけの検討ではなく
全体のアーキテクチャとして最適スポットをみつける
(決してどちらかだけを最適化して達成するのは難しい)

↓

ライブでやります！
ソリューションアーキテクトによる
IoT システムアーキテクチャーレビュー！



ここまでのまとめ

SORACOM Technology Camp 2020

双方向通信とは

- ヒト-デバイス/デバイス-サーバ/デバイス-デバイスでの相互通信
- 双方向通信にはデバイス主導/サーバ主導がある
- 遠隔制御/遠隔監視/M2M連携が代表的なユースケース

双方向通信への考慮ポイント

- 双方向通信の設計/実装は常にシンプルになるよう心がける
- 消費電力やエラーハンドリングのしやすさなどを考慮し、ユースケースに即した処理方式の検討が必要

セッションのながれ

SORACOM Technology Camp 2020

- 双方向通信が必要となるシーン
- 双方向通信における考慮ポイント
- 双方向通信の実現デザインパターンと実践
- デザインパターンまとめ

デザインパターンの考え方

SORACOM Technology Camp 2020

- 前述の通り双方向通信は考慮ポイントも多く、実装をいくらでも複雑にできる
- これまでの事例や経験をもとにデザインパターンという形で方式を整理したのが本セッション
- ユースケースや実現したい目的に応じて適切な方式案やソラコムサービスを選択できるように整理

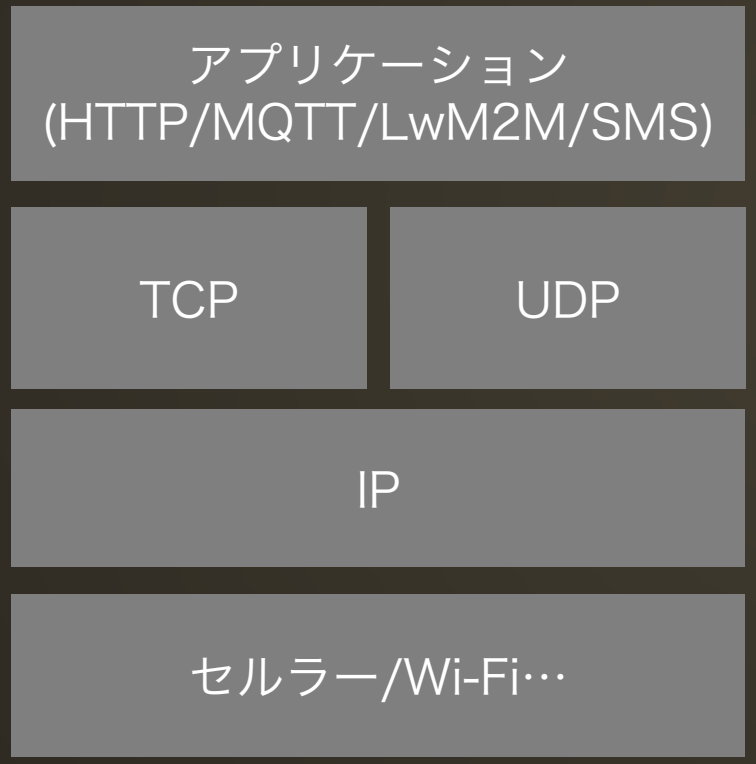
双方向通信デザインパターン

SORACOM Technology Camp 2020

- IPアクセスパターン
  - IPを指定してデバイスへ直接アクセスする方式
- アプリケーションパターン
  - MQTTやLwM2Mといった双方向通信を実現できるアプリケーションプロトコルを利用する方式
- デバイスリードパターン
  - デバイス側からの上り通信を利用する方式

プロトコルスタックでみるパターン

SORACOM Technology Camp 2020



アプリケーションパターン

デバイスリードパターン

IPアクセスパターン

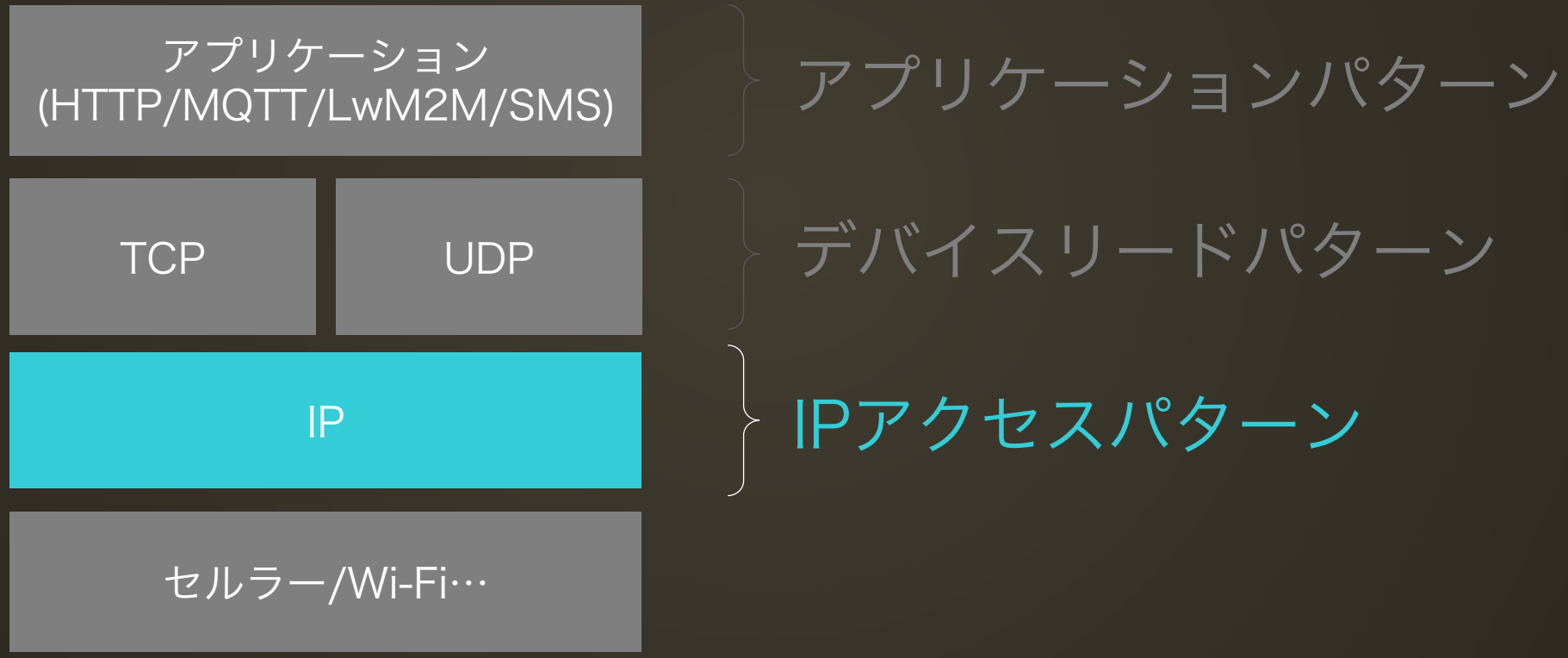
SORACOM Technology Camp 2020

双方向通信デザインパターンと実践

IPアクセスパターン

プロトコルスタックでみるパターン

SORACOM Technology Camp 2020



IPアクセスパターン

特徴

- デバイスに付与されたIPアドレスを指定した通信
- 通常のサーバ間通信と同じ要領で処理できるため方式としてはシンプル

課題

- いかにセキュアにエンドツーエンドをつなぐか

IPアクセスパターン x SORACOM

NEW

閉域接続

SORACOM Canal　SORACOM Door　SORACOM Direct

デバイスLAN接続

SORACOM Gate

オンデマンドリモート接続

SORACOM Napter

IPアクセスパターン x SORACOM

NEW

閉域接続

SORACOM Canal

SORACOM Door

SORACOM Direct

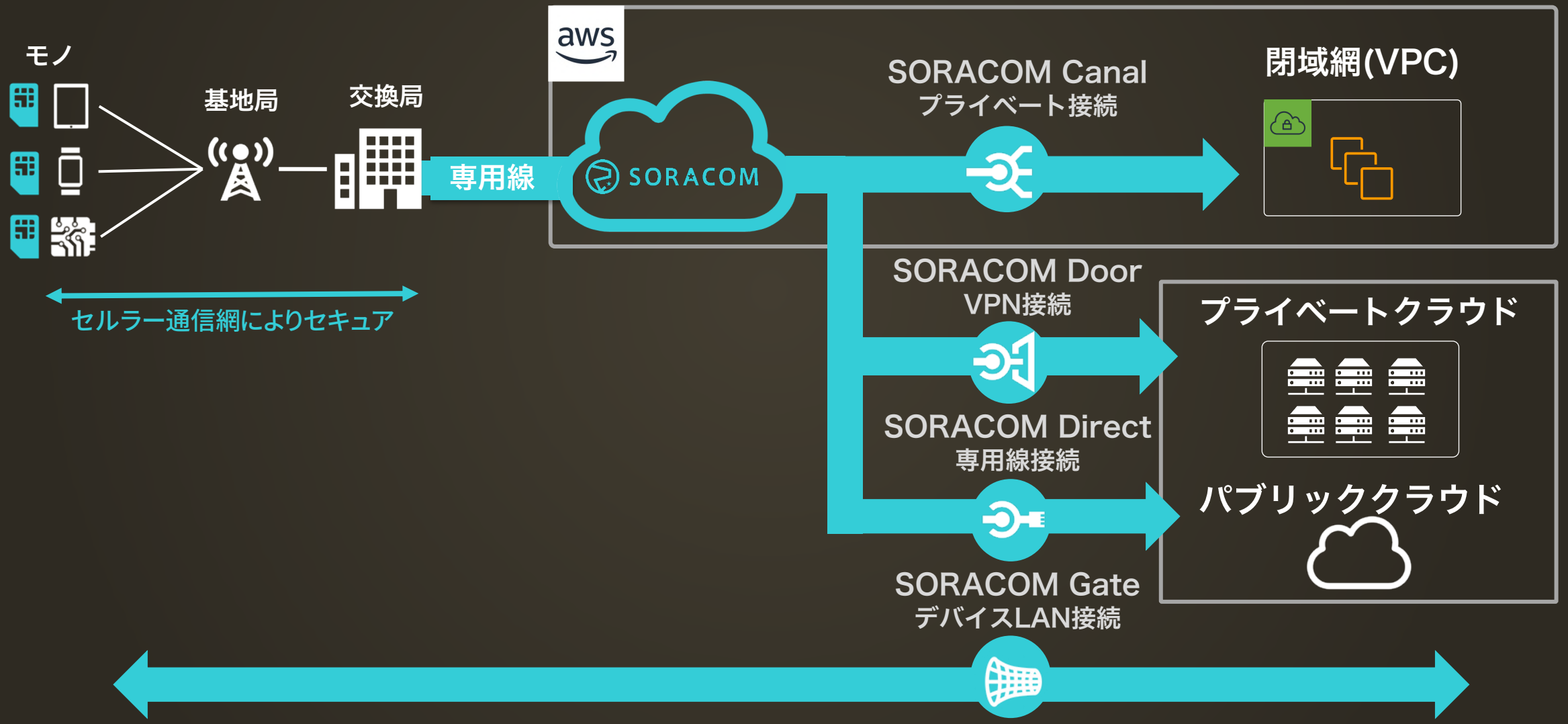
デバイスLAN接続

SORACOM Gate

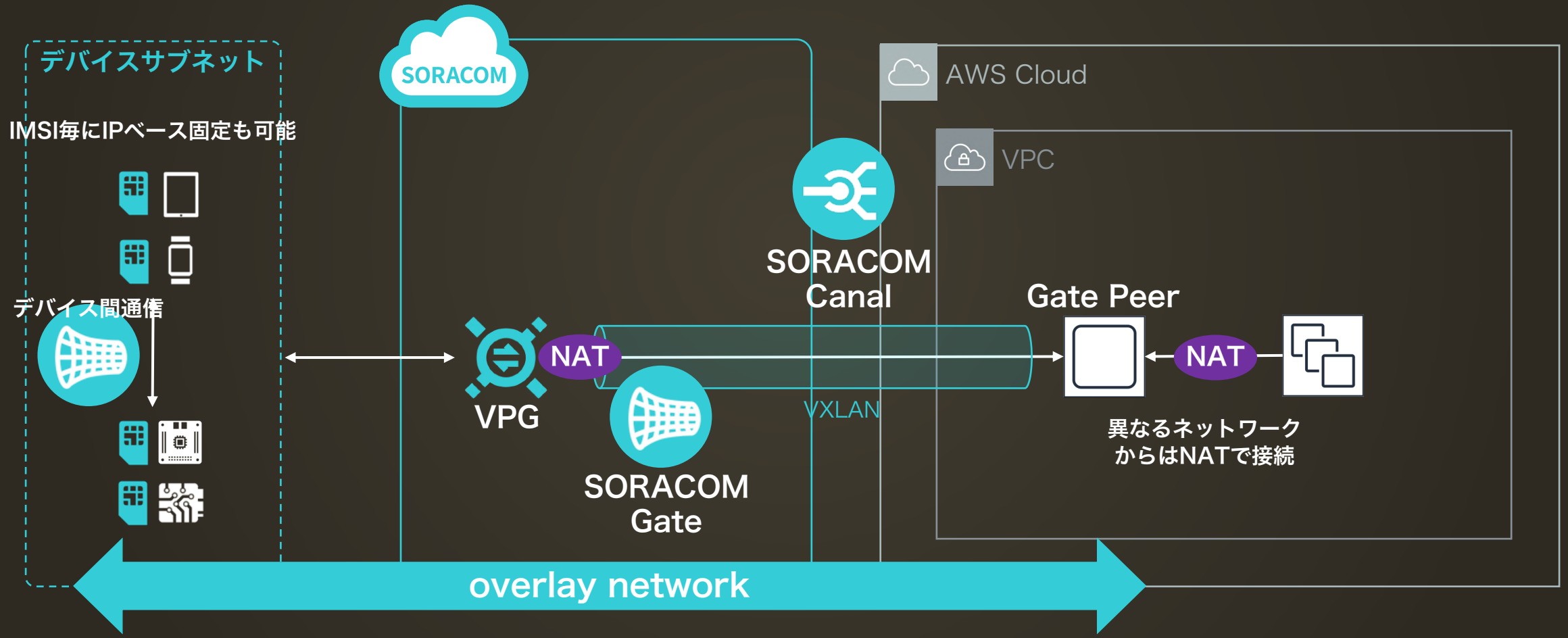
オンデマンドリモート接続

SORACOM Napter

システム全体概要


モノ
基地局
交換局
専用線
aws
SORACOM
SORACOM Canal
プライベート接続
閉域網(VPC)
SORACOM Door
VPN接続
SORACOM Direct
専用線接続
SORACOM Gate
デバイスLAN接続
プライベートクラウド
パブリッククラウド
セルラー通信網によりセキュア

基本構成(SORACOM Canalでの例)



閉域接続+Gate構成の特徴

- 閉域接続サービス+Gateによりクラウドからのデイバイスへの下り通信が可能
- デバイスからサーバへの通信はVPGでNATされる
- Gate Peerとは異なるサブネットへのルーティングが必要な場合はJunctionを組み合わせる

SORACOMで拡張する企業ネットワークの構築例
https://blog.soracom.jp/blog/2018/08/28/why-soracom-junction-redirection/

IPアクセスパターン x SORACOM

NEW

閉域接続

SORACOM Canal　SORACOM Door　SORACOM Direct

デバイスLAN接続

SORACOM Gate

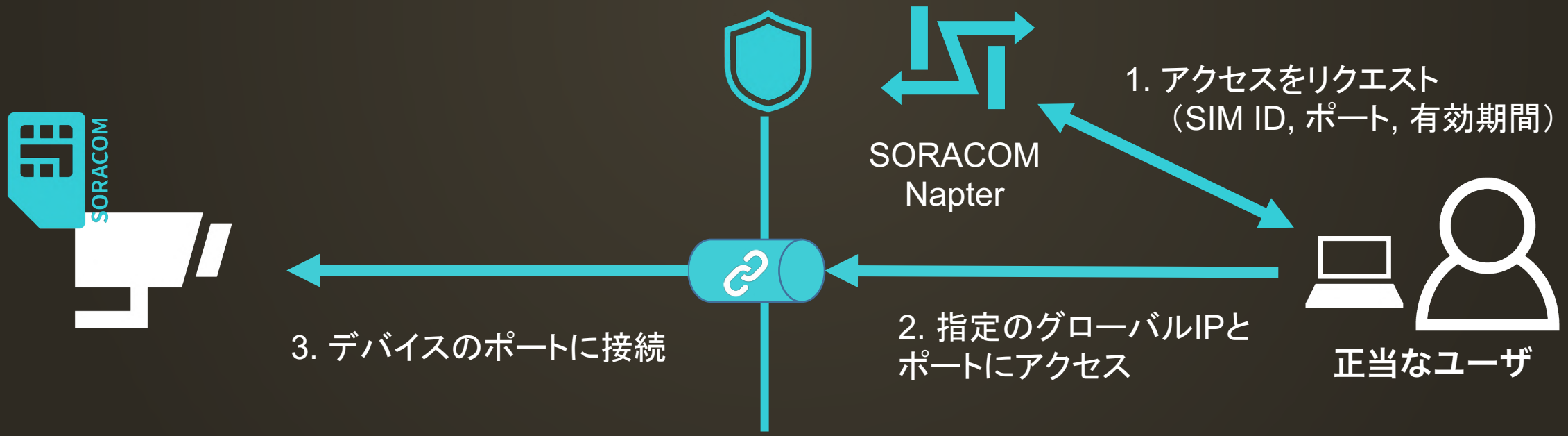
オンデマンドリモート接続

SORACOM Napter

SORACOM Napter

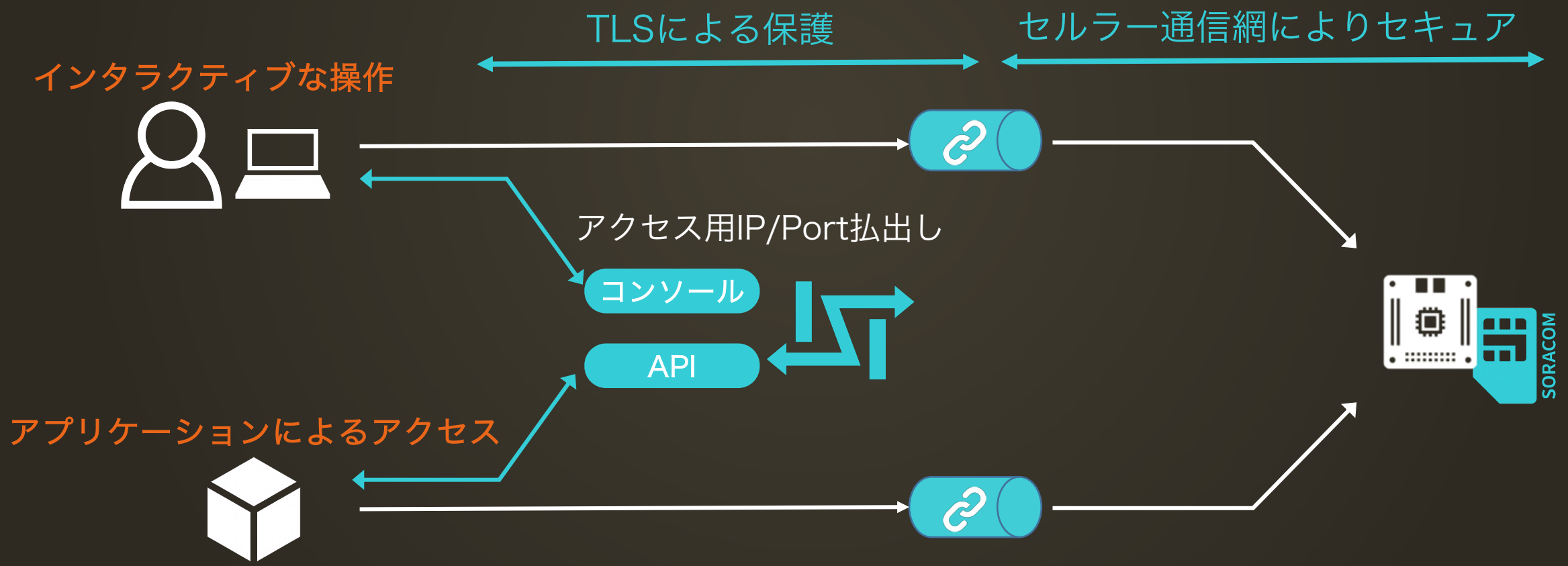
オンデマンドなリモートアクセス

SORACOM Napterで制限付きのWebアクセスやSSHアクセス可能

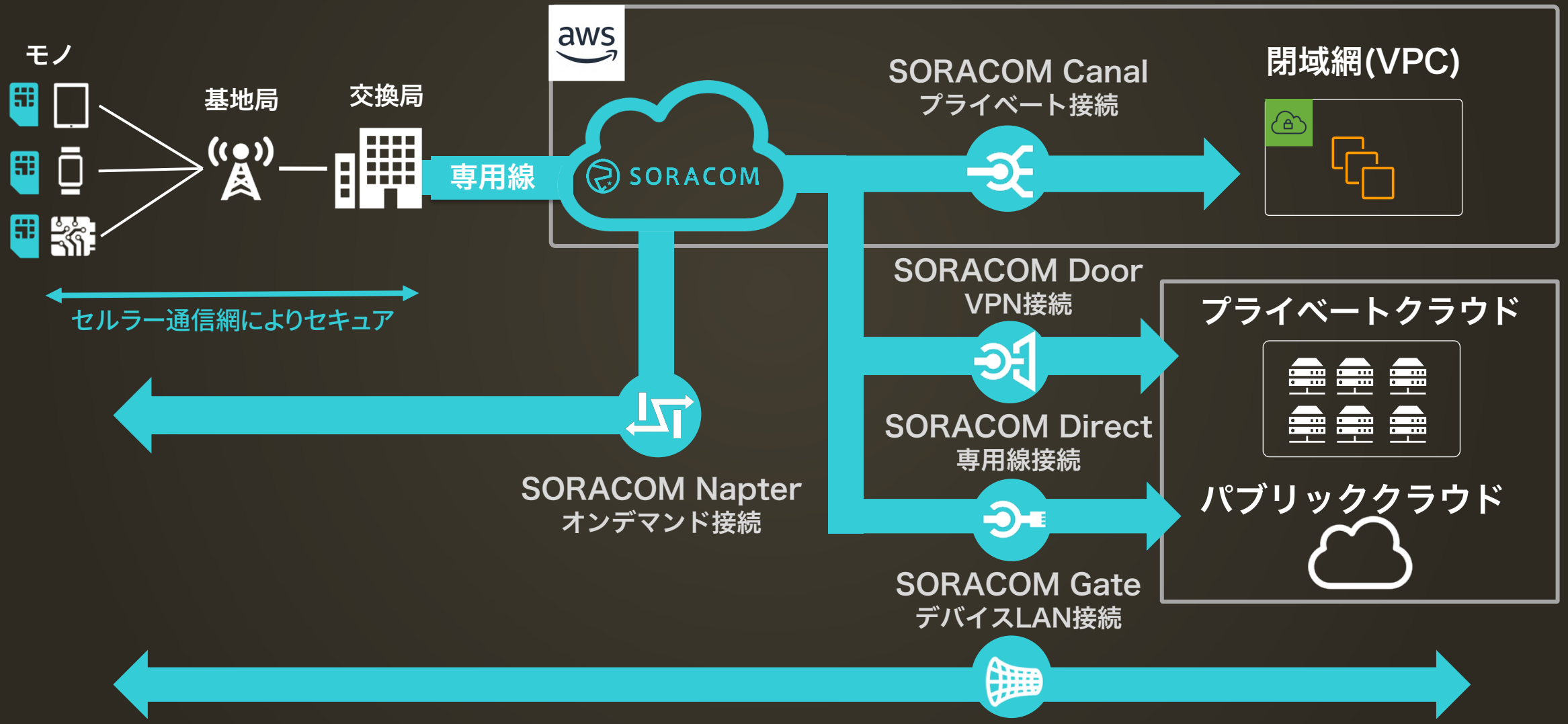


Napterによるデバイスアクセス

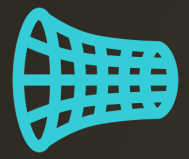
NapterによるアクセスエンドポイントはAPIからも設定が可能
そのためアプリケーションからのシステマティックな連携も可能



システム全体概要


モノ
基地局
交換局
専用線
aws
SORACOM
SORACOM Canal
プライベート接続
閉域網(VPC)
SORACOM Door
VPN接続
SORACOM Direct
専用線接続
SORACOM Gate
デバイスLAN接続
プライベートクラウド
パブリッククラウド
SORACOM Napter
オンデマンド接続
セルラー通信網によりセキュア

どう使い分けるか



SORACOM Gate

デバイス-サーバー間でのLAN構築

デバイス-サーバー間をE2Eで閉域接続が必要
定常的にサーバー間での通信が発生

頻 度

用 途

SORACOM Napter

オンデマンドにデバイスと連携

ヒトによるインタラクティブな操作が多い
作業が必要な際にだけ通信が必要

IPアクセスパターンのまとめ

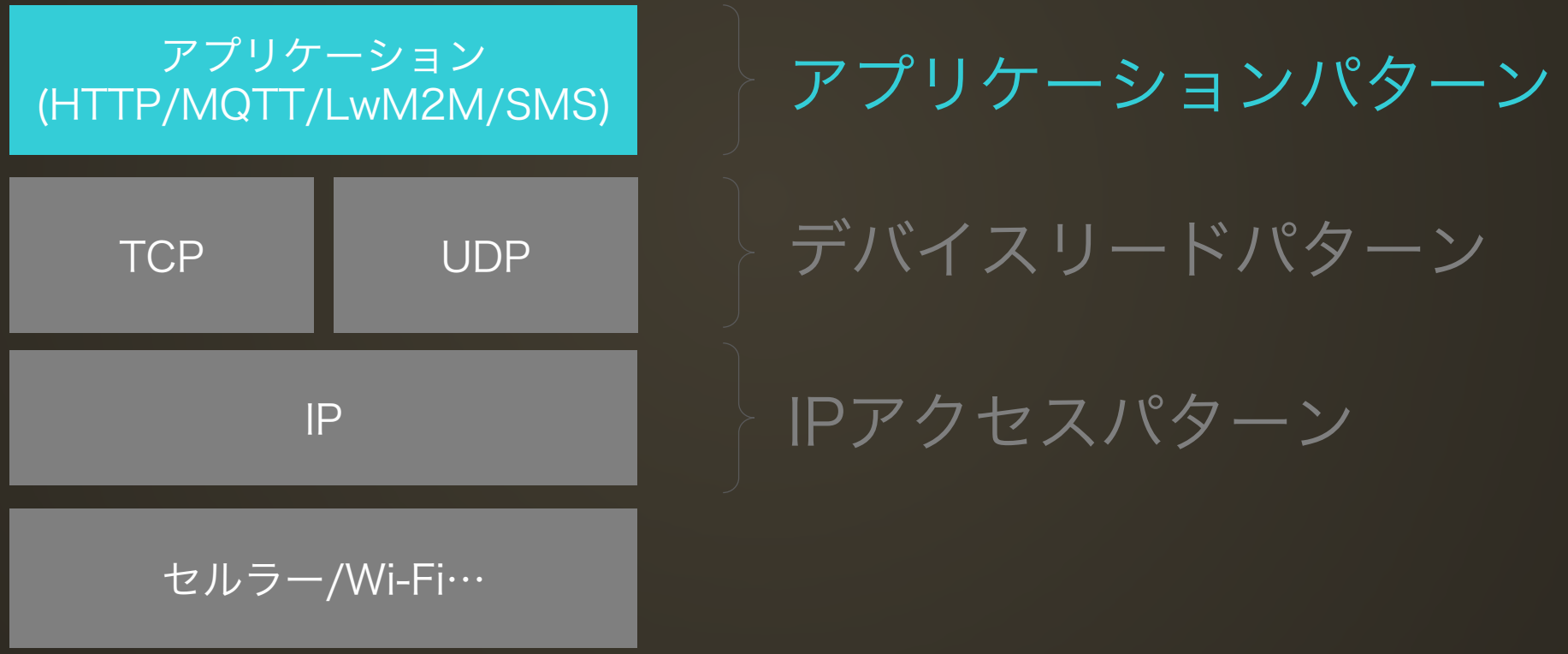
- Canal/Door/Direct + Gate + Junctionにより柔軟なネットワーク構成でのIPベース双方向通信が可能
- オンデマンドでの接続要件が必要であればNapterを利用
- インタラクティブな制御などに向いている
- 複数ネットワークを相互に接続する際はサーバ側のネットワーク設計も重要(1VPGに複数クラウドを接続したい場合など)
- 常時リクエストに応えられるようにしておくシーンと相性がよい。そのため電源が常に確保できているシーン向き。

SORACOM Technology Camp 2020

双方向通信デザインパターンと実践

アプリケーションパターン

プロトコルスタックでみるパターン



アプリケーションパターン

特徴

- MQTT/LwM2Mといった双方向通信を実現できるアプリケーションプロトコルやSMSを利用する方式
- 疎結合構成やイベント駆動といった様々な方式を実現可能

課題

- APIキーや認証情報をどう管理するか
- M2M連携のような自律連携をどうシンプルに実現するか

アプリケーションパターン x SORACOM

データ転送支援

SORACOM Beam

LwM2M

SORACOM Inventory

SMS API

アプリケーションパターン x SORACOM

データ転送支援

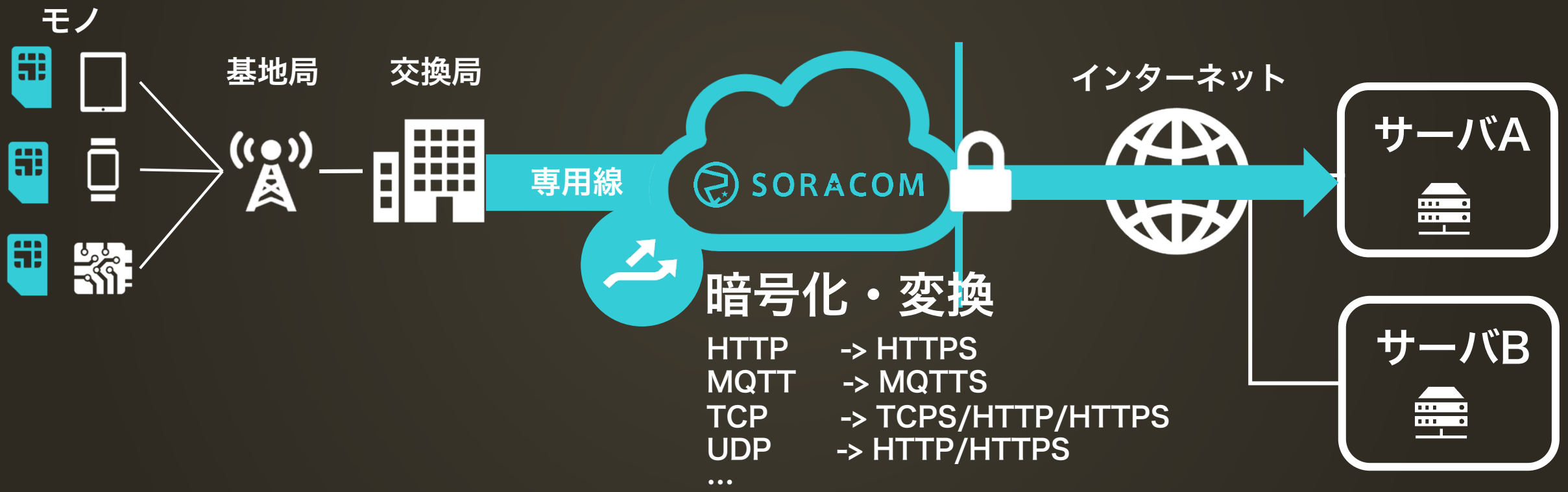
SORACOM Beam

LwM2M

SORACOM Inventory

SMS API

SORACOM Beam データ転送支援

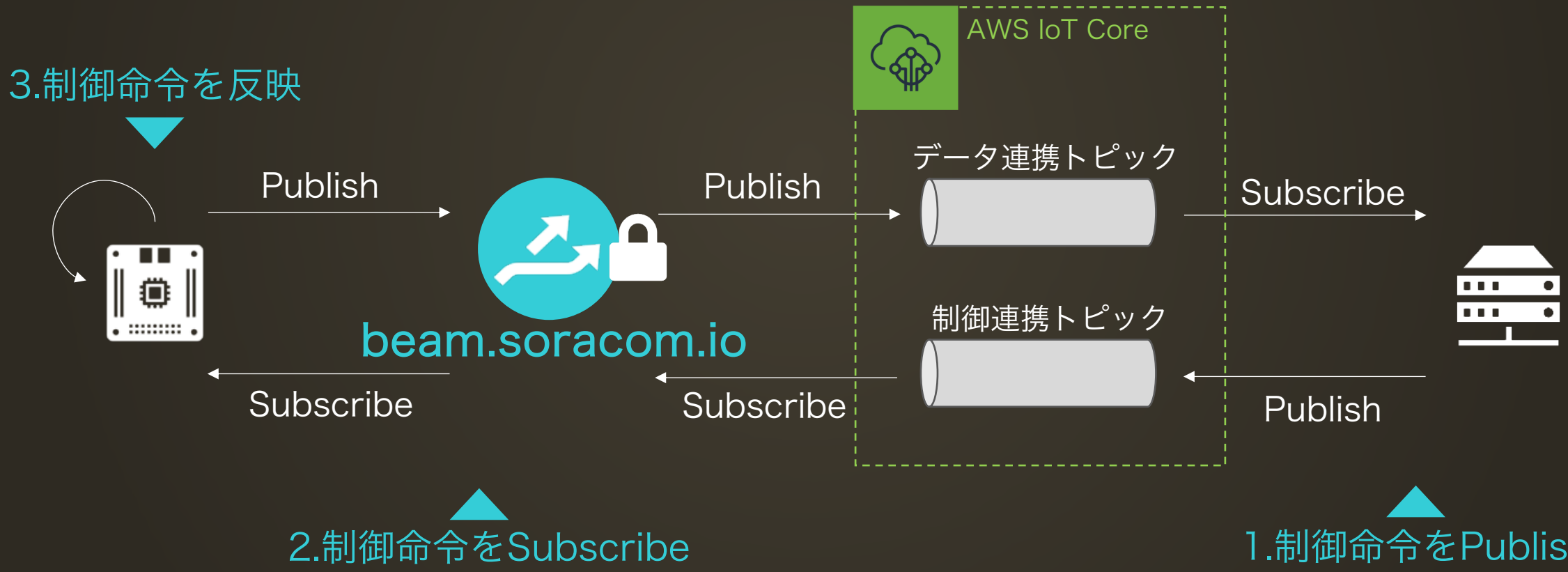


SORACOM Beam x MQTT

SORACOM Technology Camp 2020

UPDATES 1
MQTT QoS=1サポート

UPDATES 2
Willサポート



AWS IoT Shadowによる状態管理

デバイスの状態を管理するためにdevice shadowを利用したいがソラコムサービスとどう組み合わせるか

パターン1

SORACOM Beam経由でDevice Shadowで利用されているMQTT TopicをPub/Subする

＜参考＞デバイスシャドウ/デバイスツインを活用した実装のベストプラクティス (2018年末)
https://qiita.com/ma2shita/items/a330f75297c919a69f7b

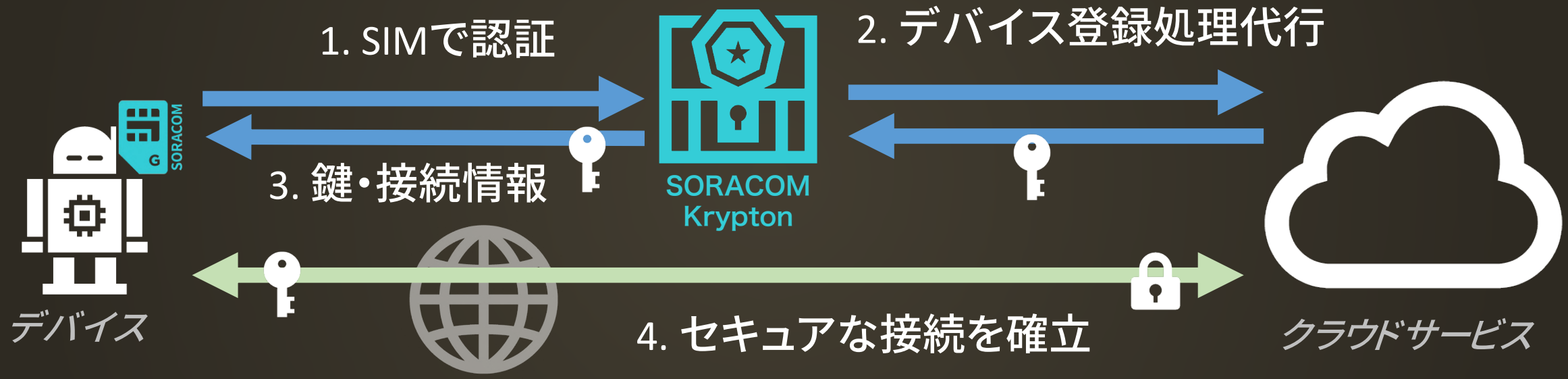
※なおAWS IoT Device SDKを利用する場合SDKによってはBeamをエンドポイントに指定できないケースがあるため事前の動作確認を推奨します

パターン2

SORACOM Kryptonによりデバイスのセキュアに証明書を配布し、デバイス-AWS IoT Coreと直接連携させる

SORACOM Krypton

― SIMで認証、接続情報をセキュアにプロビジョニング

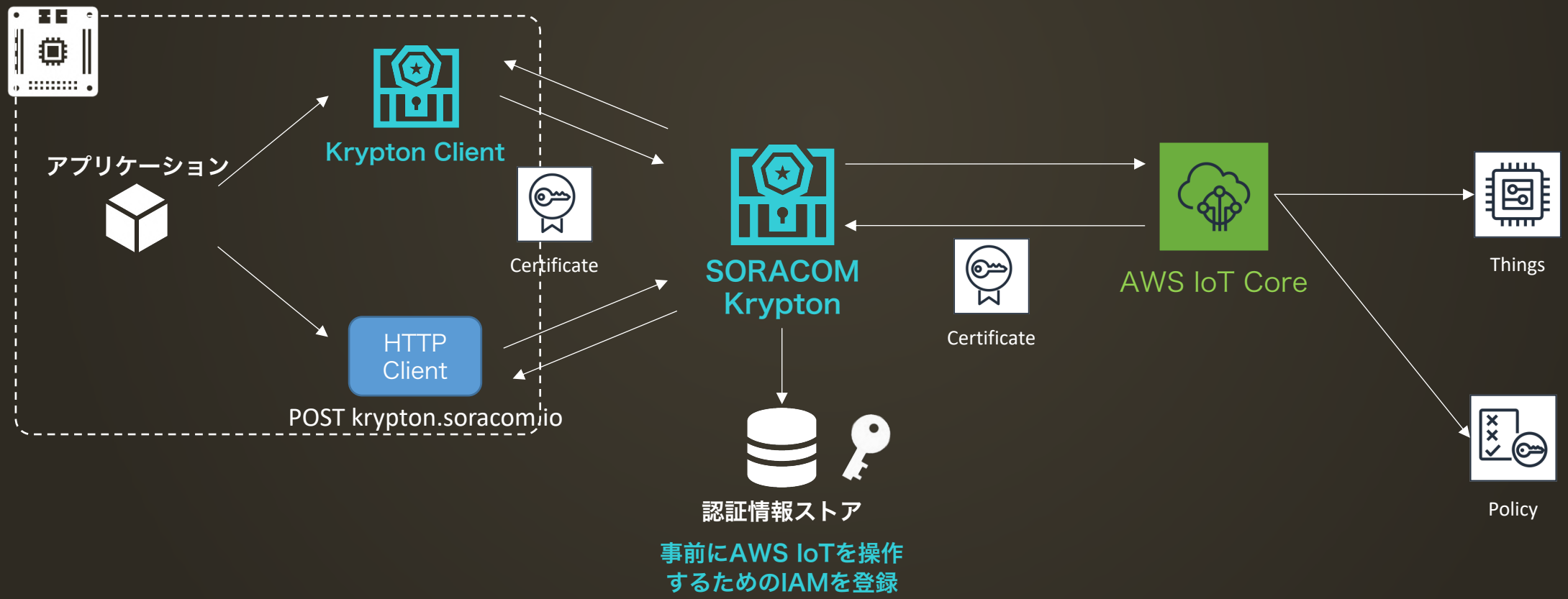


認証には強化された SORACOM Endorse を活用した2つの方法

- Option 1: セルラー回線経由で専用エンドポイントに接続
- Option 2: Wi-Fi, 有線を含む任意のアクセス回線上で SIM AKA 認証（plan01s のみ）

SORACOM Krypton x AWS IoT

デバイスシャドウを直接利用する場合、SORACOM Kryptonを利用しデバイスへ直接証明書を配布可能



アプリケーションパターン x SORACOM

データ転送支援

SORACOM Beam

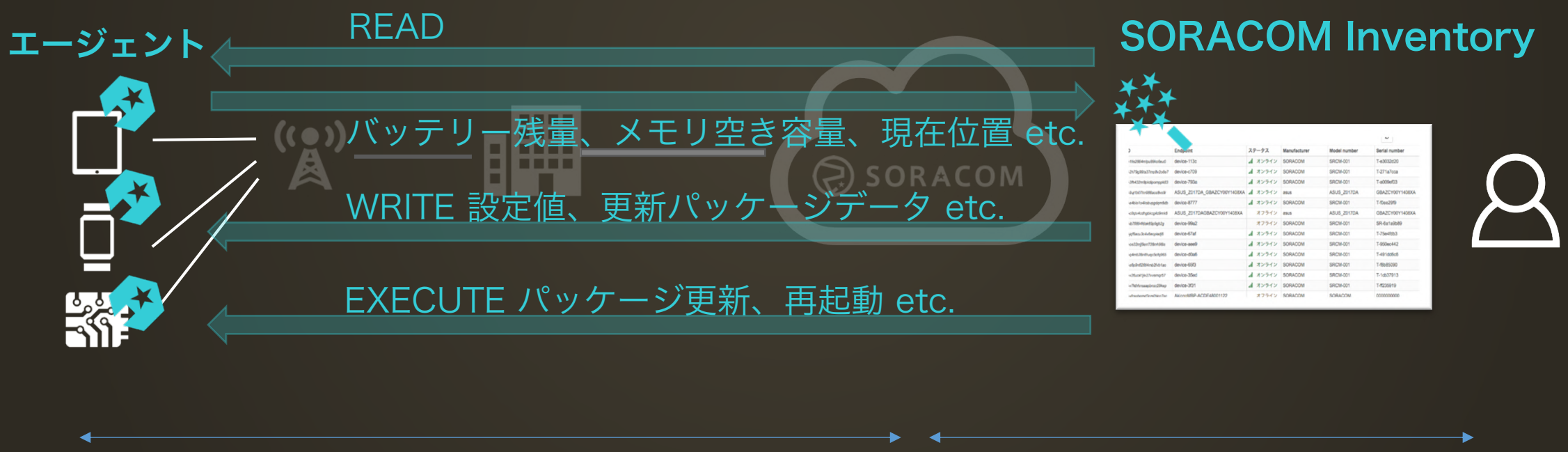
LwM2M

SORACOM Inventory

SMS API

SORACOM Inventory LwM2M

専用エージェントを通じて
デバイスの状態管理・設定更新・コマンド実行を可能に



オブジェクトモデル

Object definition

Name	Object ID	Object Version	LWM2M Version
Device	3		
Object URN		Instances	Mandatory
urn:oma:lwm2m:oma:3		Single	Mandatory

Resource Definitions

ID	Name	Operations	Instances	Mandatory	Type	Range or Enumeration	Units	Description
0	Manufacturer	R	Single	Optional	String			Human readable manufacturer name
1	Model Number	R	Single	Optional	String			A model identifier (manufacturer specified string)
2	Serial Number	R	Single	Optional	String			Serial Number
3	Firmware Version	R	Single	Optional	String			Current firmware version of the Device.The Firmware Management function could rely on this resource.
4	Reboot	E	Single	Mandatory				Reboot the LwM2M Device to restore the Device from unexpected firmware failure.
5	Factory Reset	E	Single	Optional				Perform factory reset of the LwM2M Device to make the LwM2M Device to go through initial deployment sequence where

http://devtoolkit.openmobilealliance.org/OEditor/LWMOView

READ

/3/0/3 -> Firmware Version

EXECUTE

/3/0/4 -> Reboot

LwM2M利用時のポイント

- LwM2Mで規定しているのはリソースIDとそれに対するリクエスト及びレスポンスのフォーマットであるため、実際にどういった方法で値を返すのかはLwM2Mエージェントの内部実装次第
- LwM2Mエージェントの起動やプロセス管理はOS側で設定が必要
- 定義済みリソースで要件を満たせないかをまずは考える
  - 本当に必要な場合のみカスタムオブジェクトを定義
  - リソースIDのメンテナンスや通信量の増加など
- 必要なモデルのみを吟味すること
  - どのシーンでも利用できるように汎用的に作ろうとすると実装が肥大化する恐れがありメンテナンスも大変

アプリケーションパターン x SORACOM

データ転送支援

SORACOM Beam

LwM2M

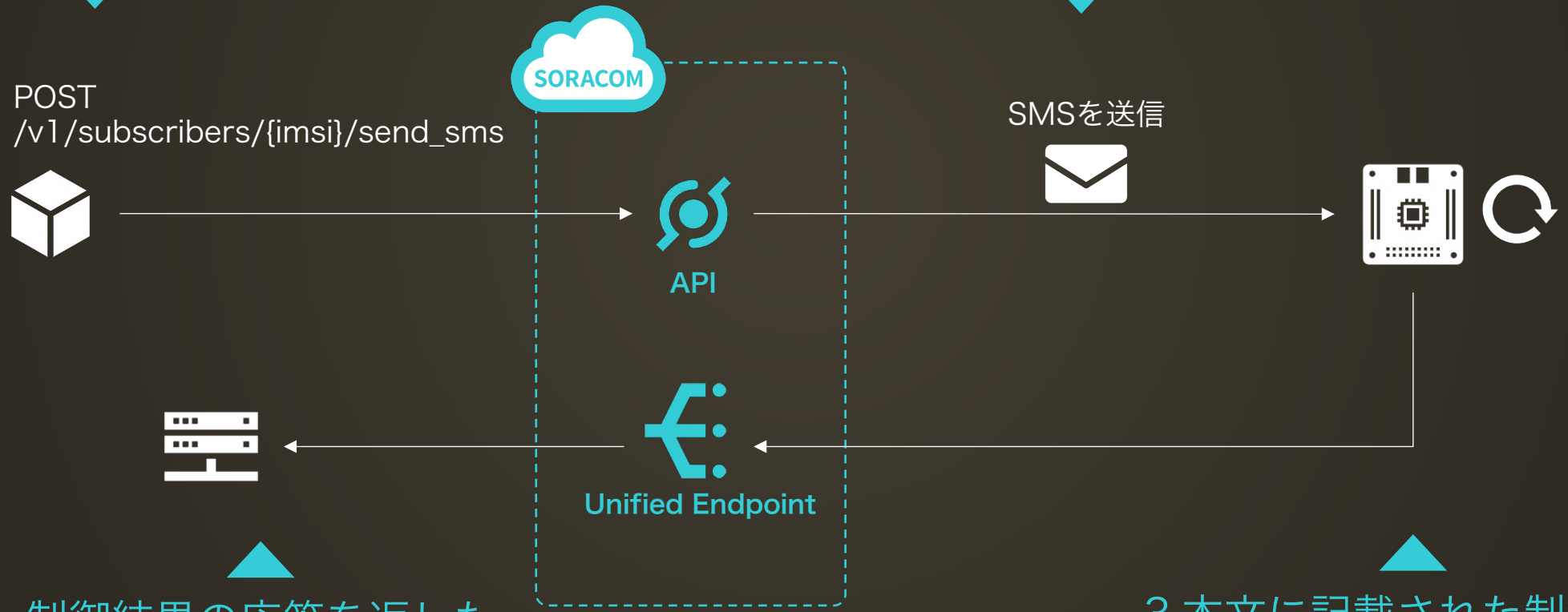
SORACOM Inventory

SMS API

SMSによる方式

1. SMS送信APIを実行

2. SMSをデバイスへ送信



4. 制御結果の応答を返したい際はBeamへ送信

3.本文に記載された制御命令を読み取り＆実行

SMSでのポイント

- モデムがドーマントモードでも利用可能
  - 基地局からのページングシグナルで受信
- セルラーネットワークのシグナリングのみを利用
  - IPデータ通信のセッション確立が不要
- APIによるSMS送信
  - SIMのオペレータやSAMユーザからのみAPIをコールできる
- SMSから各種ソラコムサービスへの連携も可能
  - Beam/Funnel/Harvest/FunkへSMS経由で送信

アプリケーションパターンまとめ

- MQTTやLwM2M、SMSといったプロトコルの機能を活用した柔軟な連携が可能
- 多くのデバイスとの連携や値を基にした疎結合で自律的なシステム連携といった用途に向いている
- 一方でプロトコルの機能にどこまで頼るかは検討が必要
  - デバッグ/切り分けが難しくなっていく
  - 通信量の増加

SORACOM Technology Camp 2020

双方向通信デザインパターンと実践

デバイスリードパターン

デバイスリードパターン

特徴

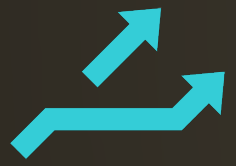
- デバイスからのリクエストに対するレスポンスに設定情報をつめて、デバイス側でハンドリングさせるパターン
- デバイスからのポーリングもこのパターンに含まれる

課題

- デバイスどのようにレスポンスを返すか
- デバイスの設定情報をどのように管理するか

デバイスリードパターン x SORACOM

データ転送支援



SORACOM Beam

NEW

FaaS連携

SORACOM Funk

メタデータ API

NEW

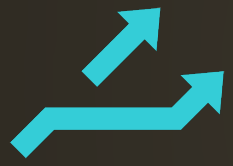
ファイル配信

SORACOM Harvest Files

デバイスリードパターン x SORACOM

リクエスト/リプライ方式

データ転送支援



SORACOM Beam

FaaS連携

SORACOM Funk

- デバイスからのデータ送信に対するリプライとして制御データをかえす
- デバイスからのデータに応じて制御情報を切り替えたいユースケース

ポーリング方式

メタデータ API

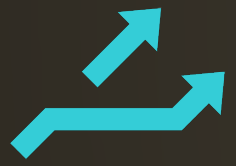
ファイル配信

SORACOM Harvest Files

- デバイスから定期的に問い合わせをおこない反映データを取得する
- 定期的に静的データを取得するユースケース

デバイスリードパターン x SORACOM

データ転送支援



SORACOM Beam

NEW

FaaS連携

SORACOM Funk

メタデータ API

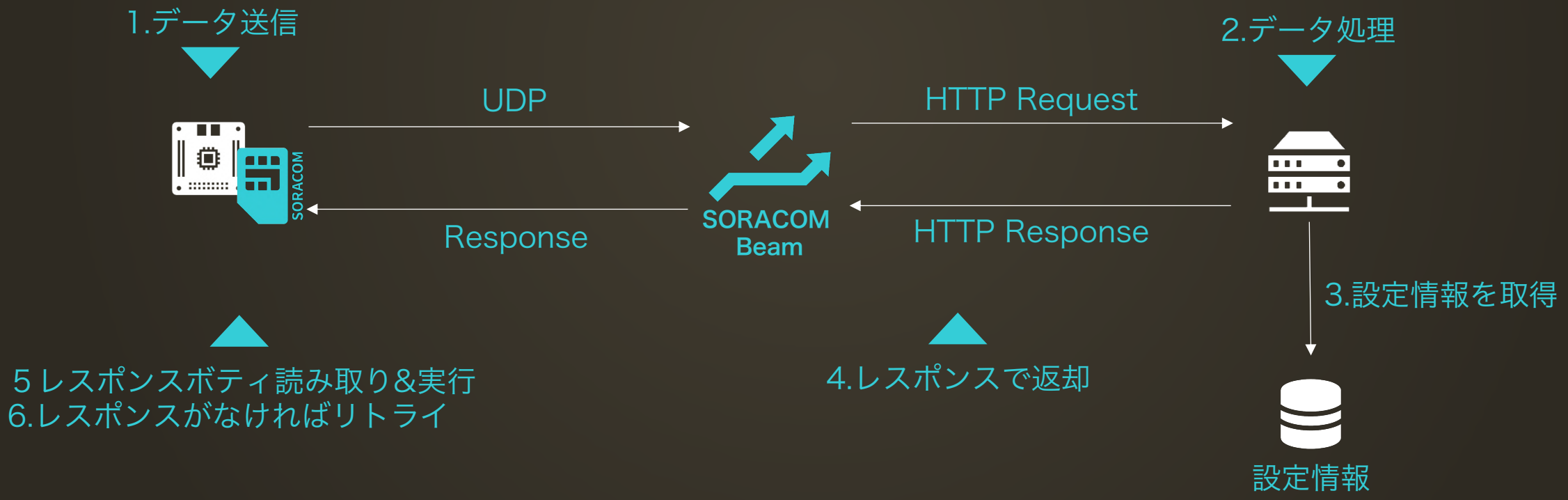
NEW

ファイル配信

SORACOM Harvest Files

SORACOM BeamにReq/Res方式

デバイスからの上り通信時のリクエストに対するレスポンスに
制御情報をつめて返す方式(ポーリングでも同様の方式が可能)



SORACOM BeamにReq/Res方式


```
char sendData[100];
char receiveData[1024];

//SORACOM Beam経由でUDPでデータ送信
int connectId;
connectId = Wio.SocketOpen("beam.soracom.io", 23080, WIOLTE_UDP);
if (connectId < 0) {
  SerialUSB.println("********** ERROR **********");
  goto err;
}

sprintf(data, "{\"uptime\":%lu}", millis() / 1000);
if (!Wio.SocketSend(connectId, data)) {
  SerialUSB.println("### ERROR! ###");
  goto err_close;
}

//SORACOM Beam経由でレスポンスデータを取得
int length;
length = Wio.SocketReceive(connectId, receiveData, sizeof (data), RECEIVE_TIMEOUT);
if (length < 0) {
  SerialUSB.println("********** ERROR **********");
  goto err_close;
}

//取得データを使った処理をおこなう　Ex) JSONの場合 {"key01": "value01", "key02": "value02"}
StaticJsonBuffer<1024> jsonBuffer;
JsonObject &json_root = jsonBuffer.parseObject(receiveData);
value01 = json_root["key01"];
value02 = json_root["key02"];
```

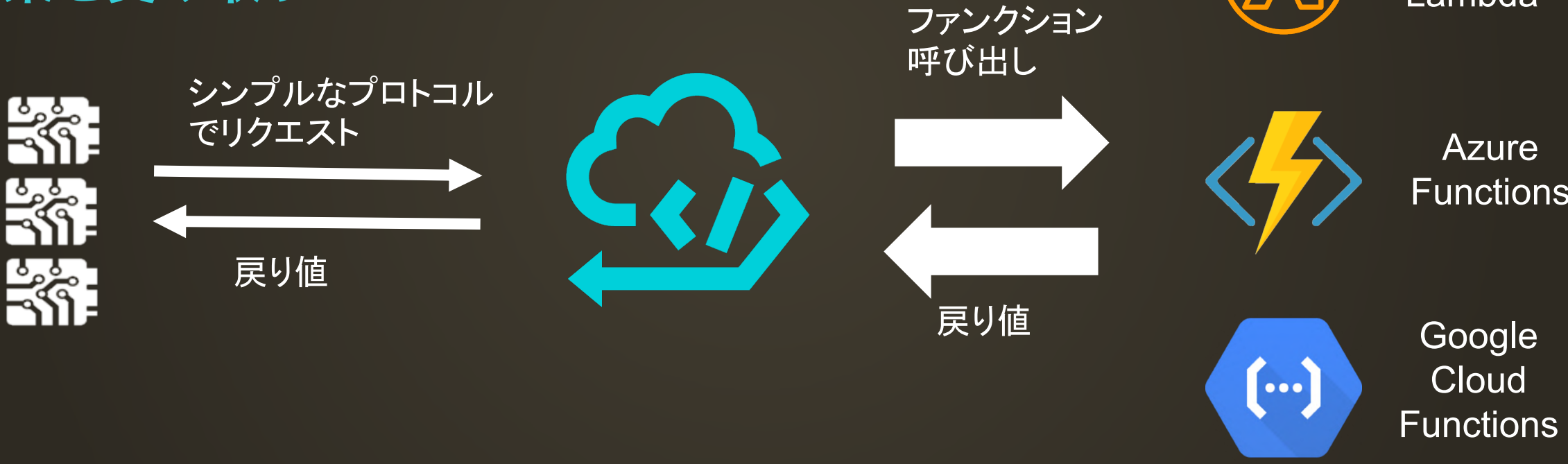
データ送信

レスポンス受信

取得データ反映

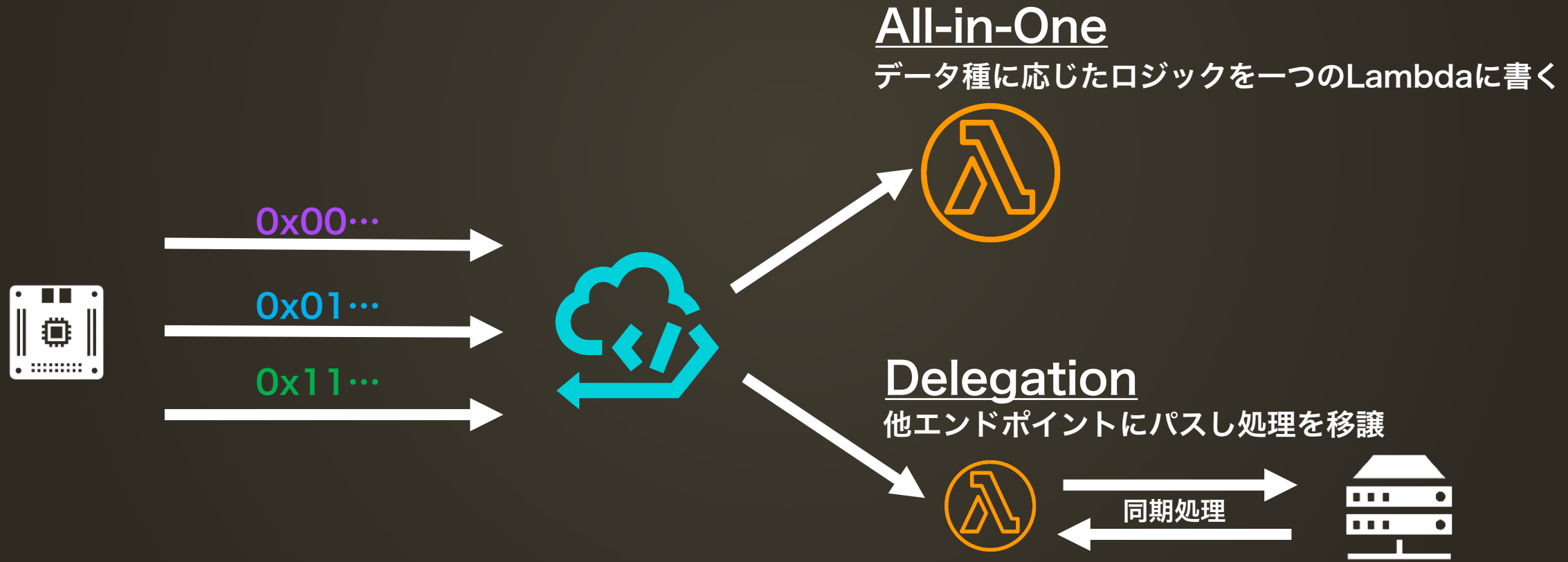
SORACOM Funk

サーバレスでクラウド上のコードを実行
結果を受け取り



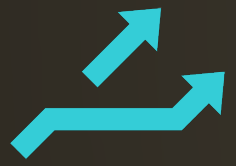
SORACOM Funk

複数のデータ種の連携が必要な場合、それらに応じた処理をファンクション側で吸収する必要がある。



デバイスリードパターン x SORACOM

データ転送支援



SORACOM Beam

NEW

FaaS連携

SORACOM Funk

メタデータ API

NEW

ファイル配信

SORACOM Harvest Files

メタデータサービスとポーリング方式

メタデータに設定情報を含め、デバイスはメタデータサービス
をポーリングし設定情報を反映する方式
ユーザデータへ任意の設定情報やスクリプトも格納可能

メタデータサービスへのリクエスト

メタデータサービスへの問い合わせにクエリを利用し、連携するデータ量を最低限に抑えられまたTCPでの問い合わせも可能

```
GET metadata.soracom.io/v1/subscriber
GET metadata.soracom.io/v1/subscriber.tags
GET metadata.soracom.io/v1/userdata
```

→ JSON で返却される

メタデータサービスへのリクエスト

詳細情報　通信量履歴　タグ　監視　セッション詳細

名前	値
location	tokyo 削除
debug_flag	true 削除

```
~/d/tech-camp curl -s http://metadata.soracom.io/v1/subscriber.tags | jq .
{
  "debug_flag": "true",
  "name": "kei-tech-camp-demo",
  "location": "tokyo"
}
```

メタデータサービスへのリクエスト

メタデータサービス設定

ON

☑ 読み取り専用

許可するオリジン

この値が Access-Control-Allow-Origin ヘッダーに指定されます

ユーザーデータ

```
#!/bin/bash
echo THIS IS USERDATA.
echo My name is $(curl -s http://metadata.soracom.io/v1/subscriber | jq -r .tags.name )
```

```
~/d/tech-camp curl -s http://metadata.soracom.io/v1/userdata | bash
THIS IS USERDATA.
My name is kei-tech-camp-demo
```

Harvest Files

サーバレスでファイルを収集・保存・取得

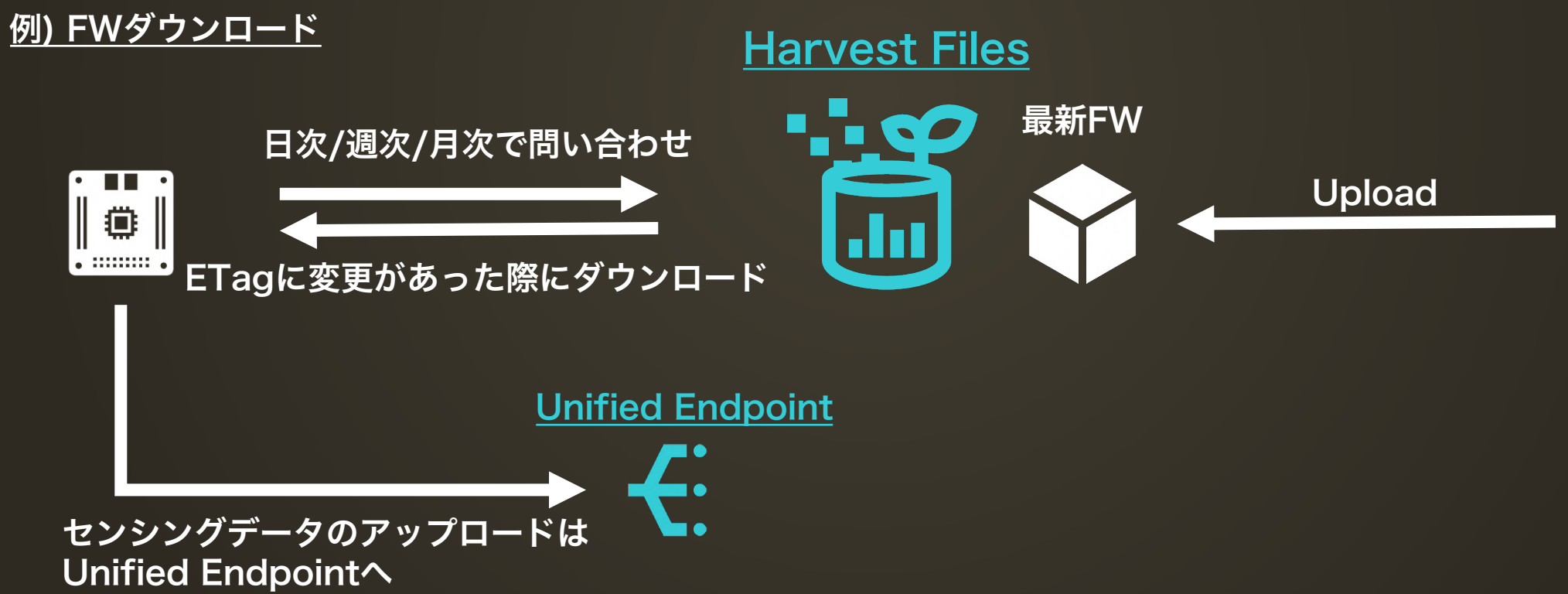
パスパラメータを
メタデータで置き換え
- SIM ID
- タイムスタンプ

PUT
/images/:imsi/:time.png

GET
/images/295101234567890/

Harvest Filesによるデータ配信

Harvest Filesではファイルのメタ情報としてETagをサポートしているため変更が同じパスに問い合わせをして変更があった際にのみダウンロードが可能



デバイスリードパターンのポイント

- デバイス主導での通信のため、バッテリー駆動のようなスリープを伴うケースでも利用可能
- 特に対象デバイス数が多い場合、デバイスからのアクセスが集中しすぎないようにリクエストを散らすことが望ましい
- メタデータサービスはDBではないので、シンプルなデータ構造を保つこと！
  - データ構造が複雑になるほどデバイス側の処理も複雑になっていく

デバイスリードパターンまとめ

- 上り通信のみを利用して双方向通信を実現可能
- 即時反映が不要なケースや展開するデバイス数が多いようなシーンに適している
- デバイス側でレスポンスに応じた処理を施す必要があるため、複雑なロジックやデータ構造はなるべく避けた方が無難

セッションのながれ

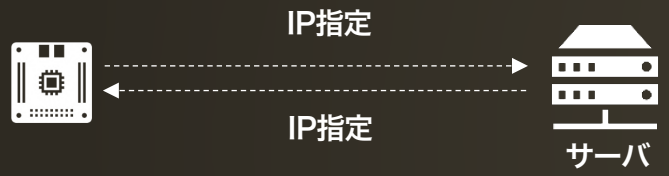
SORACOM Technology Camp 2020

- 双方向通信が必要となるシーン
- 双方向通信における考慮ポイント
- 双方向通信の実現デザインパターンと実践
- デザインパターンまとめ

デザインパターンまとめ

SORACOM Technology Camp 2020

IPアクセスパターン

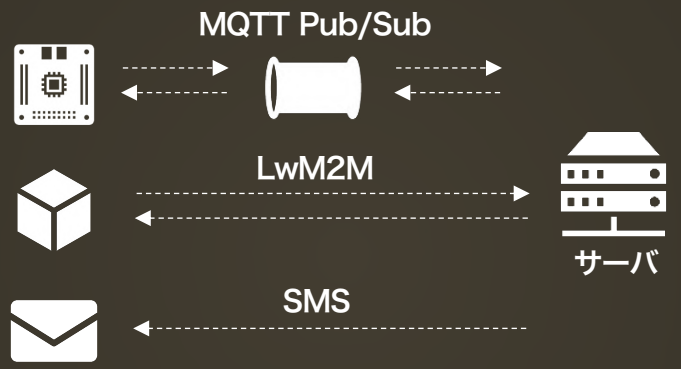


IPを指定して直接デバイス-サーバ間で接続が可能で、サーバ操作と同様のオペレーションが実現可能。

メリット
- インタラクティブな操作をおこないやすい

考慮ポイント
- デバイス-サーバ間をいかにセキュアに接続するか
- デバイスが常時稼働している用途向け

アプリケーションパターン

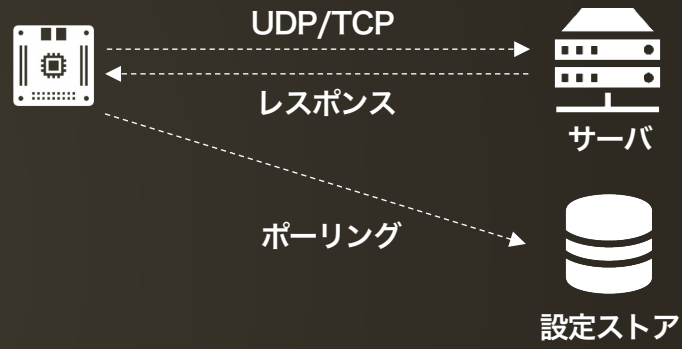


MQTTやLwM2Mといった双方向通信が可能なアプリケーションプロトコルを利用。疎結合な構成や、イベント駆動といった様々な方式を実現可能

メリット
- プロトコルで様々な機能をサポートしているため柔軟なM2M連携システムを構築可能。

考慮ポイント
- デバイス仕様で利用できるプロトコルに制限がある
- バッテリー駆動の際は通信量や使い方に注意

デバイスリードパターン

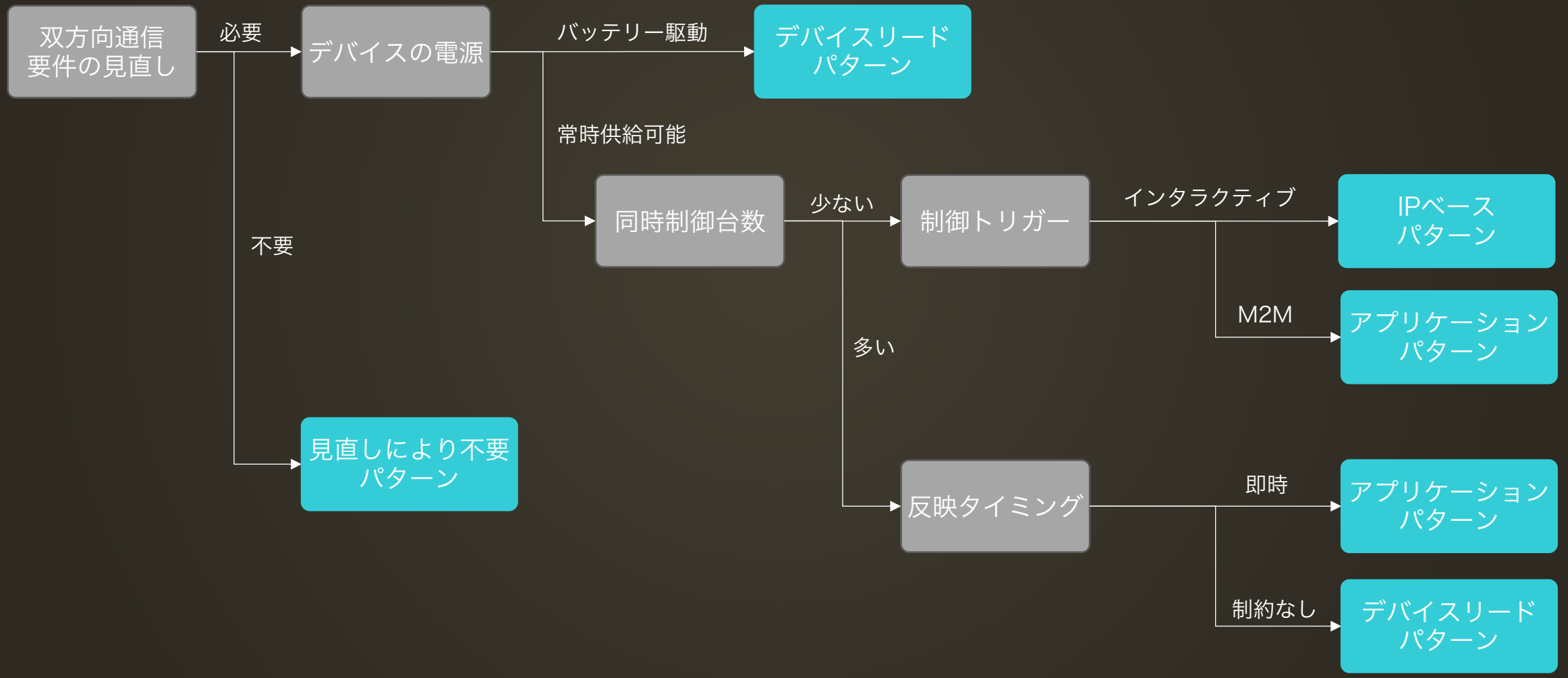


デバイスからのリクエストに対するレスポンスに設定情報や制御情報をつめて返す。デバイスからのポーリングもこのパターンに分類

メリット
- デバイス駆動のためバッテリー駆動デバイスのようなスリープを伴うケースでも利用可能

考慮ポイント
- 即時反映が必要なシーンでは不向き

デザインパターン選択表



改めて双方向通信について考える

SORACOM Technology Camp 2020

- その双方向通信が本当に必要かを再度検討する
  - 検討ポイント、実装が複雑になるのを避ける
- 目的に対して適切な方式を選択する必要がある
  - 対象台数や制御内容、対象デバイスなどを考慮
- 双方向通信とトレードオフになる要件を把握
  - 消費電力、設計や実装の難易度
- 双方向通信は必ずしも下り通信を必要としない。
  - デバイス主導で実現する方式案も検討する

SORACOMの願い

SORACOM Technology Camp 2020

クラウド ⇒ 多くのビジネス、Webサービス

SORACOM ⇒ 多くのIoTビジネス、システム

たくさんの
IoTプレイヤーが生まれますように

SORACOM Technology Camp 2020

世界中のヒトとモノをつなげ
共鳴する社会へ

SORACOM