

デバイス-クラウドの双方向通信 デザインパターンとその実践

Keigo Suda
Solutions Architect, SORACOM, INC.

2018/11/22

本日のハッシュタグ

#soracom



@SORACOM_PR



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

セッションで扱うトピック

- 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ー-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。

セッションで扱うトピック

- 双方向通信の方式をデザインパターンとしてお話しします。
- 双方向通信が必要となった際にリファレンスとしてご活用ください。
- 質問はセッション後の技術コーナー、懇親会にてお願いします。

セッションのながれ

- 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

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

エンタープライズなものが好きです



セッションのながれ

- 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

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



双方向通信とは

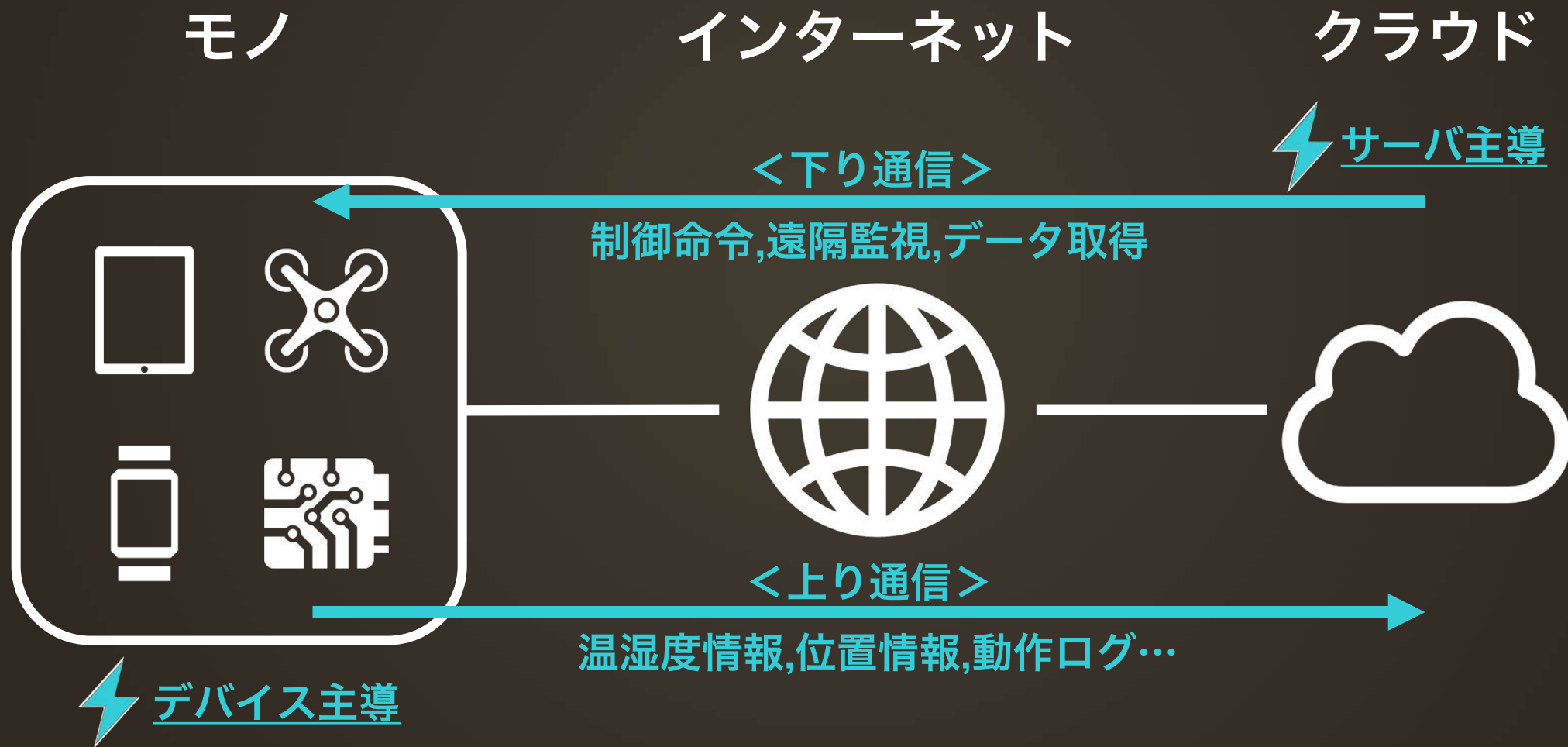
モノ

インターネット

クラウド



双方向通信とは



代表的な双方向通信

- 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

お客様事例: コマツ様

KOMATSU

SMART CONSTRUCTION



「ICT建機との高速でセキュアな通信を実現」

SORACOM Gateにより
ICT建機とクラウドシステムを
シームレスに接続
施工現場のIoT活用に貢献

お客様事例: アマノ様

AMANO



ロボット掃除機RcDCの
遠隔サポートに
SORACOM Gate

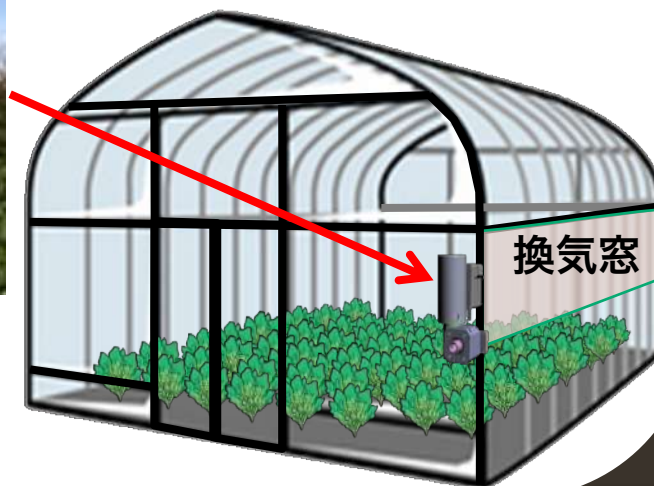
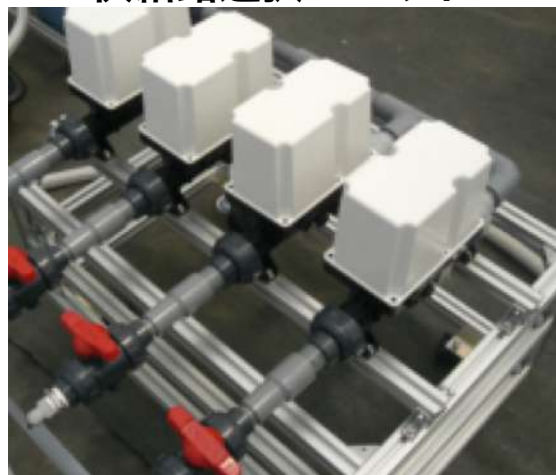
夜間施錠中のお客様先で
稼働するロボット掃除機
とセキユアに通信し、運
用をサポート

お客様事例: CKD様

換気窓開閉モータ



供給路選択ユニット



ビニールハウスや畜舎の
環境情報を基に
温湿度管理や流体制御

SORACOM Inventoryで
環境情報を基にバルブ・制御機
器を自動制御し省力化を実現

セッションのながれ

- 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？

本当にその通信必要ですか？

まず検討したいこと

- 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

- 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?(幂等?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？

考慮ポイント – 物理面

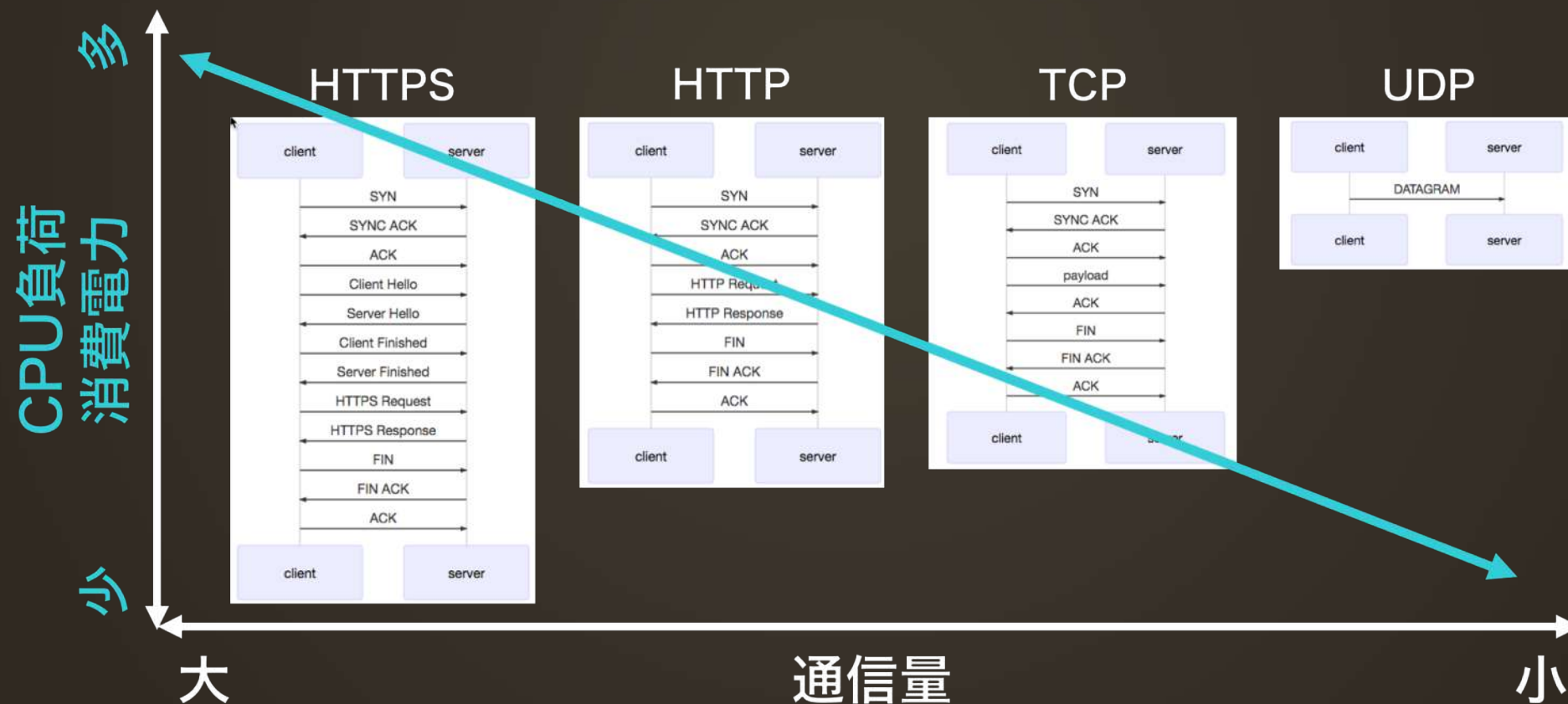
- デバイス
 - どういったデバイスが対象か(Linuxベース?RTOS?)
- 電源
 - 稼働時の電源確保(常時供給可能?バッテリー駆動?)
- 台数
 - 対象とするデバイスの台数(数台?数百~数千?)
- 制御内容
 - どういった制御命令を出す?(幂等?確実に1回?)
- セキュリティ
 - アクセス経路や制御情報は安全か？

消費電力と双方向通信

- 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

- 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

- IoTシステムでは多くの障害ポイントが存在
 - 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っても重要
- 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)
 - E2Eで物理的に距離があるため、総じてトータルでのレイテンシが高く即時でのロールバック的な処理を反映しにくい

双方向通信とエラー処理

- 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的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

ここまでのまとめ

双方向通信とは

- 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



デザインパターンの考え方

- 前述の通り双方向通信は考慮ポイントも多く、実装をいくらかでも複雑にできる
- 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

セッションのながれ

- 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

双方向通信デザインパターン

- 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

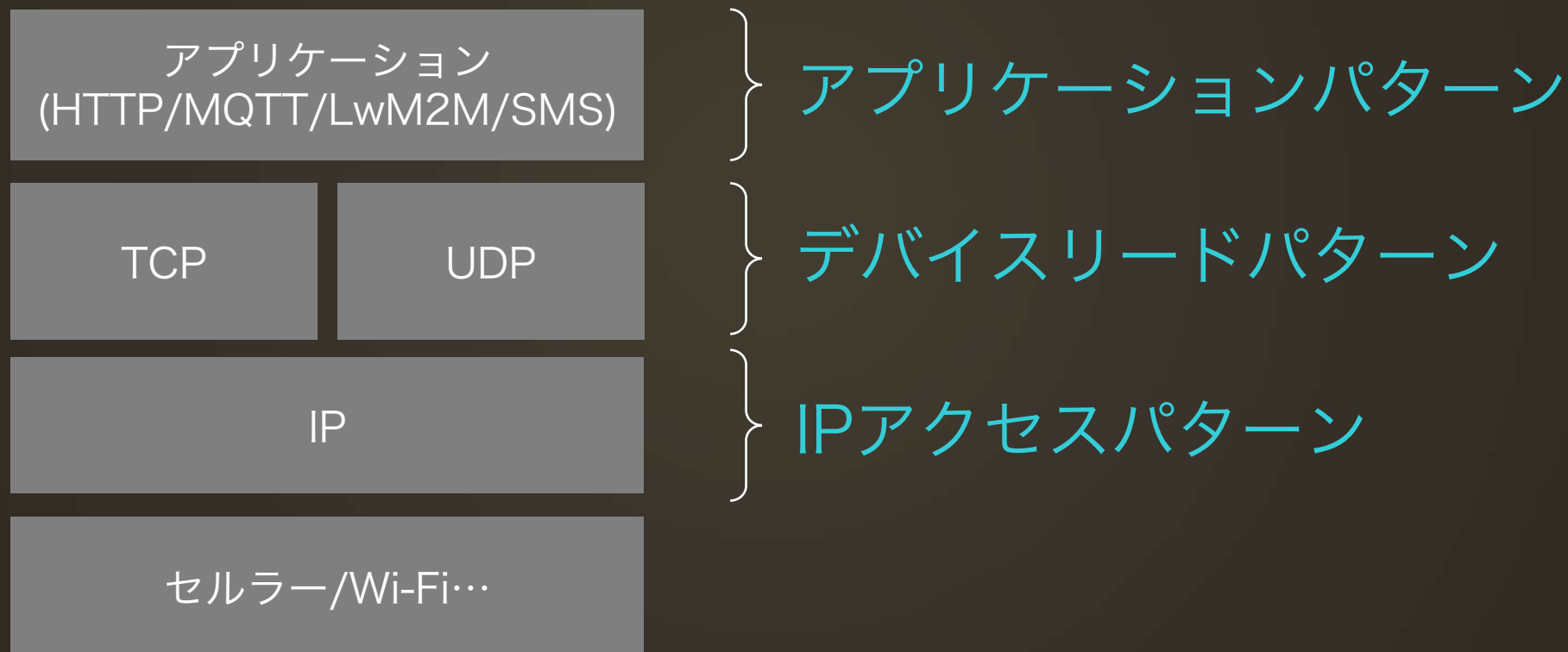
- アプリケーションパターン

- 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

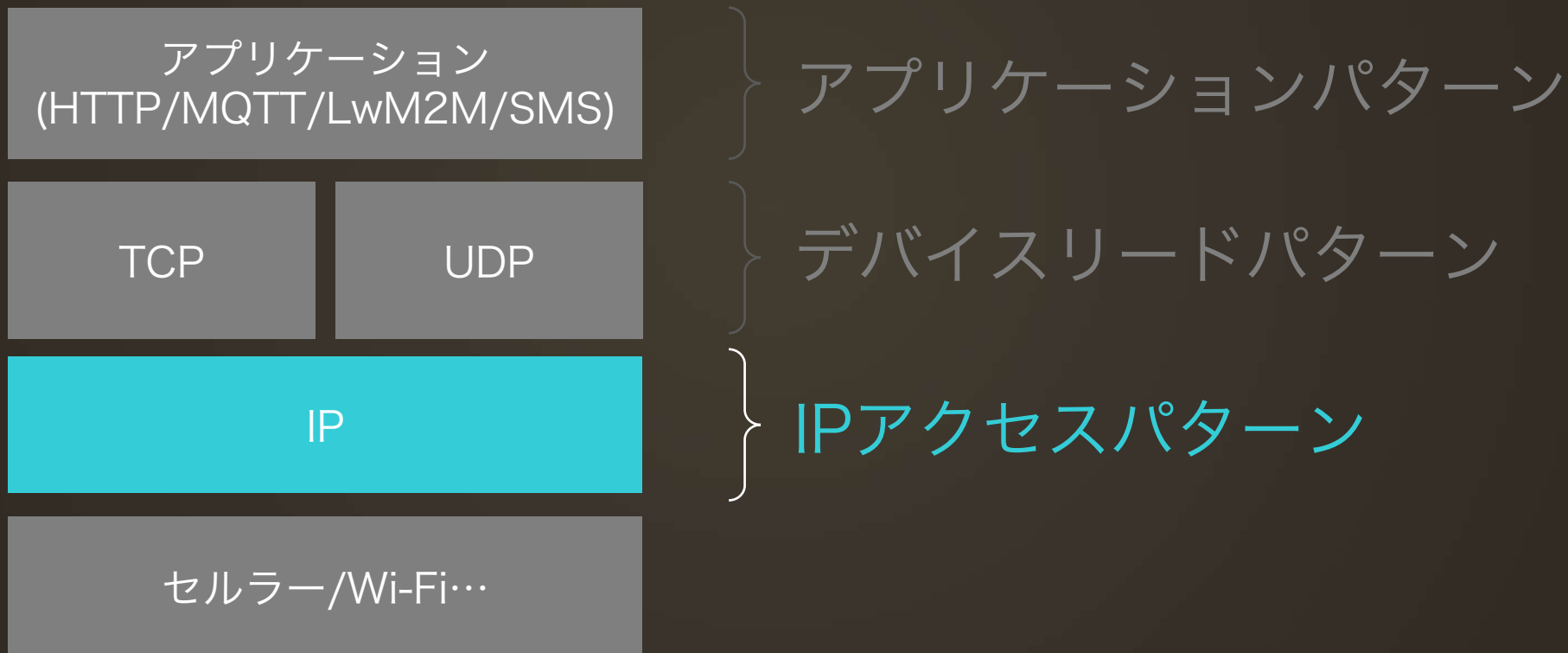
- デバイス側からの上り通信を利用する方式

プロトコルスタックでみるパターン



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



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

閉域接続



SORACOM
Canal



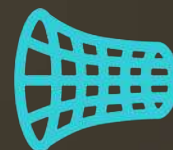
SORACOM
Door



SORACOM
Direct

+

デバイスLAN接続



SORACOM
Gate

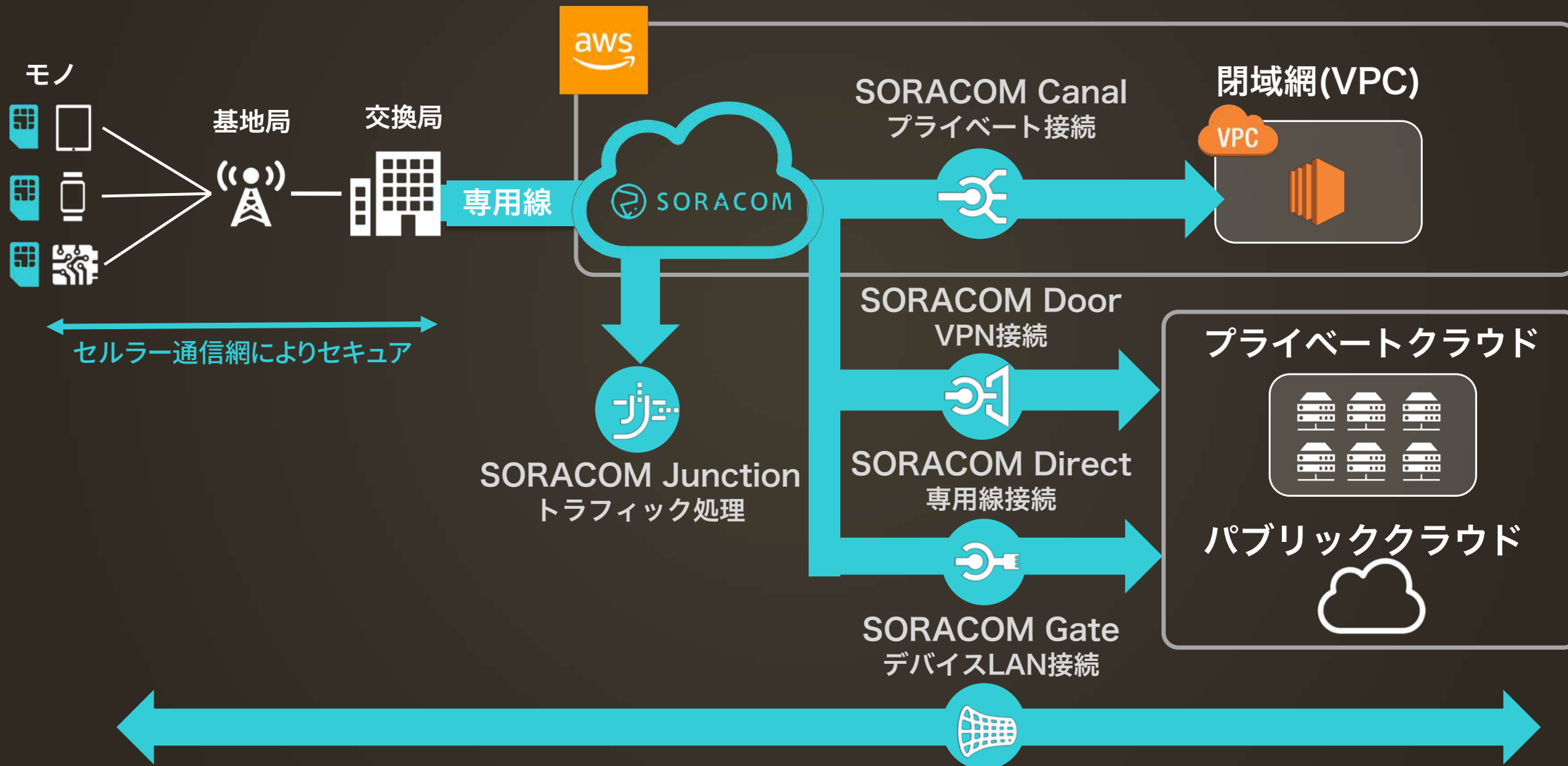
+

トラフィック処理



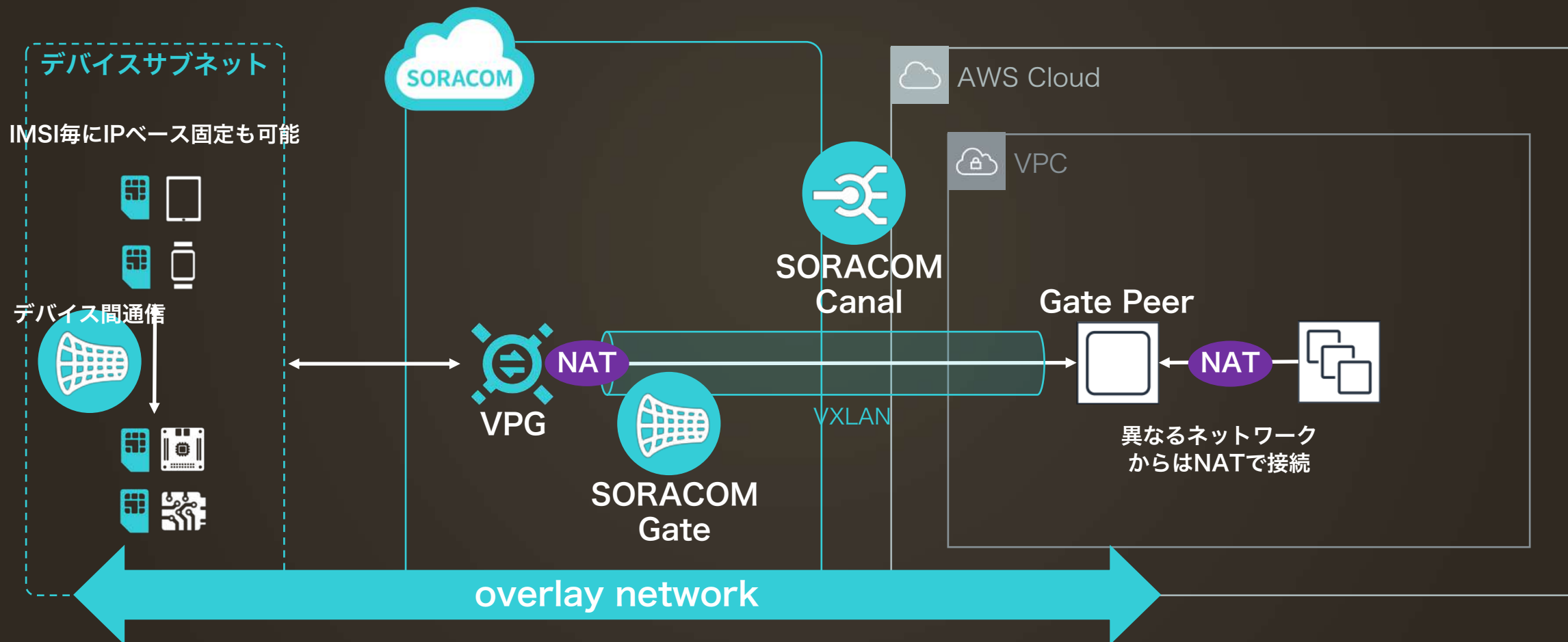
SORACOM
Junction

システム全体概要





基本構成(SORACOM Canalでの例)



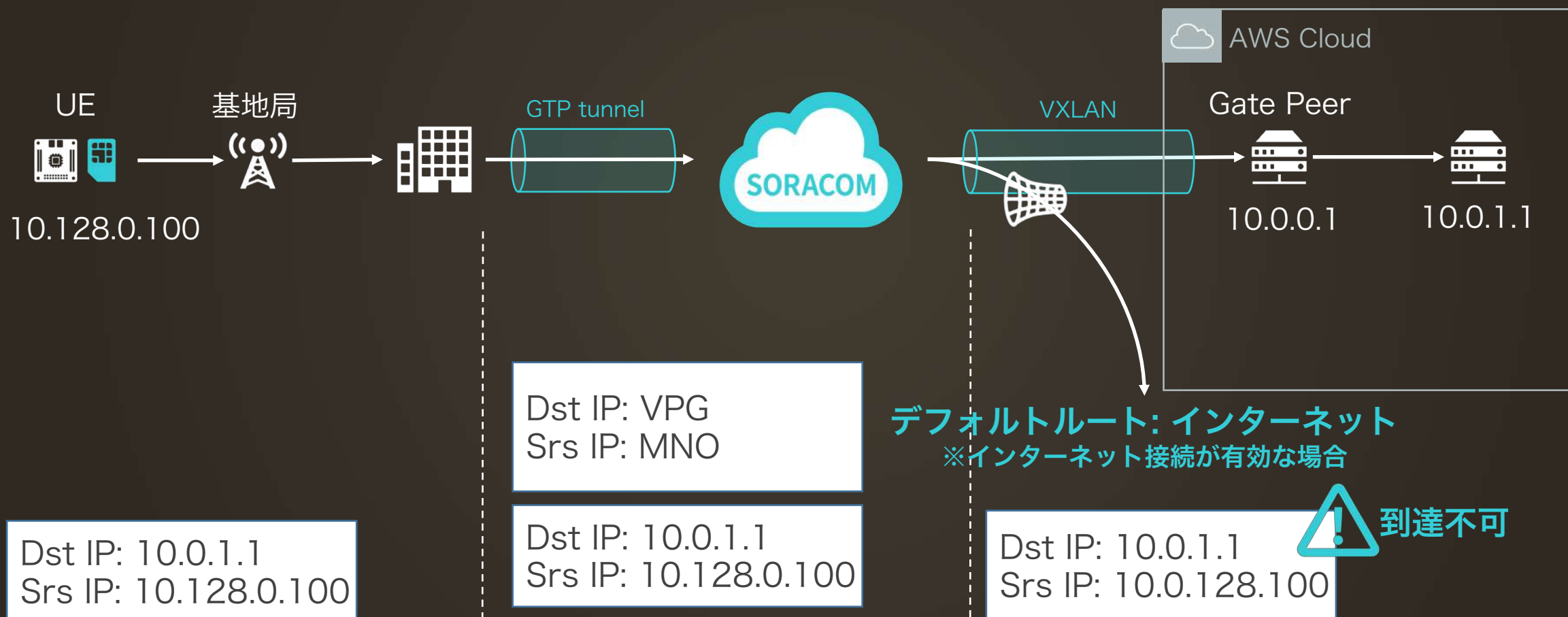
閉域接続+Gate構成の特徴

- 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をでNATされるため、デバイスのソースIPを特定できない
- Gate Peerとは異なるサブネットへのルーティングができない(後述)

Gate利用時のパケット遷移

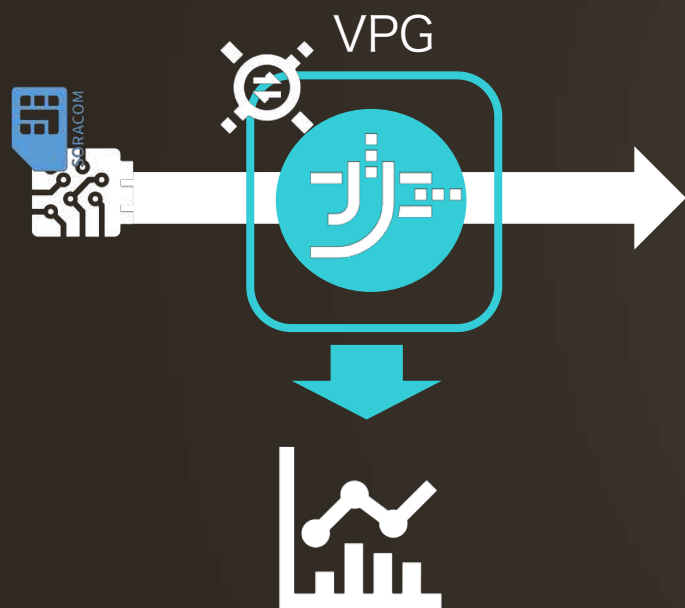
SORACOM
Technology Camp 2018

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/>

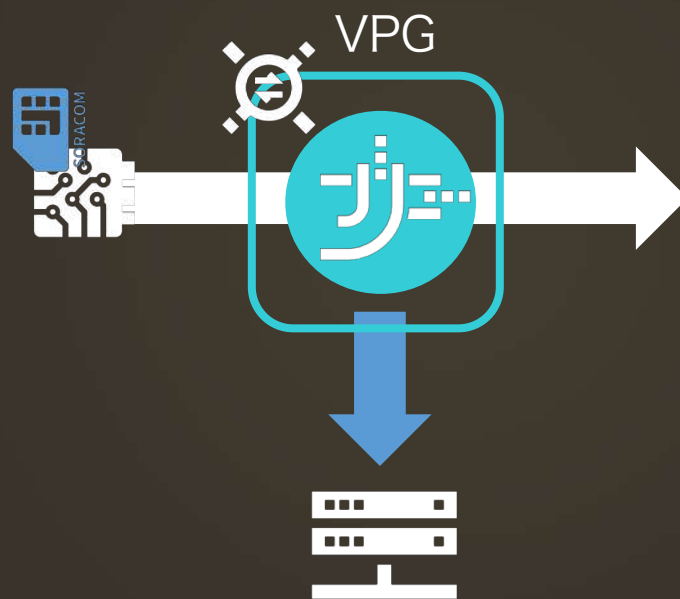
SORACOM Junction トラフィック処理

Inspection



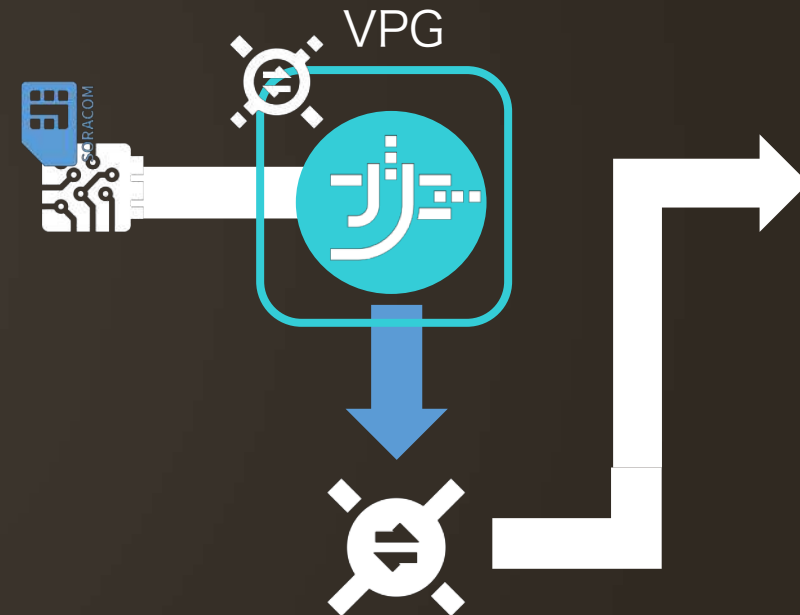
パケットフローを解析して統計情報を出力

Mirroring



パケットのコピーを指定の宛先に転送

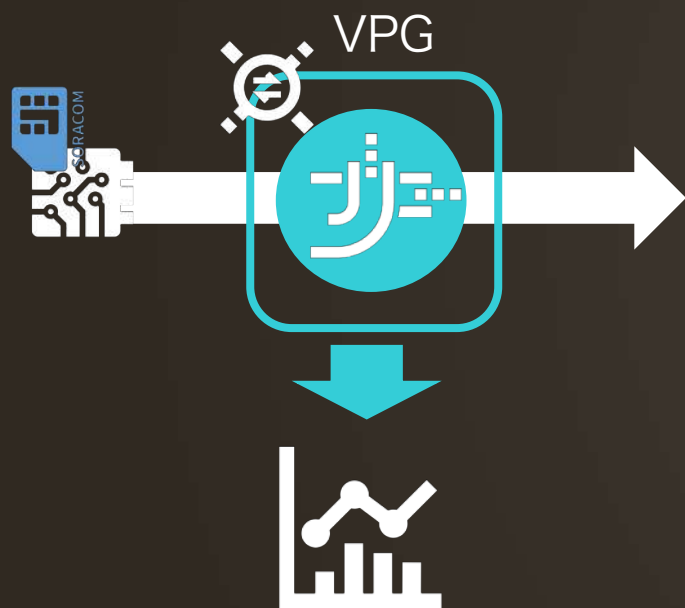
Redirection



パケットを指定のゲートウェイ経由で転送

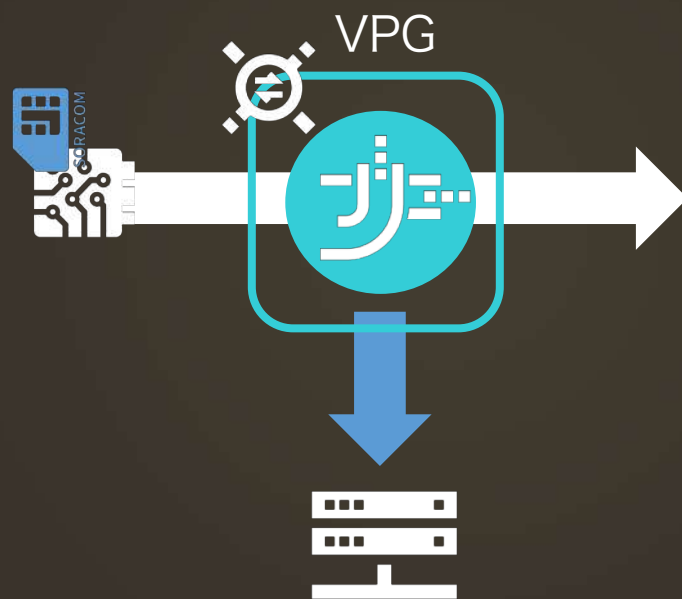
SORACOM Junction トラフィック処理

Inspection



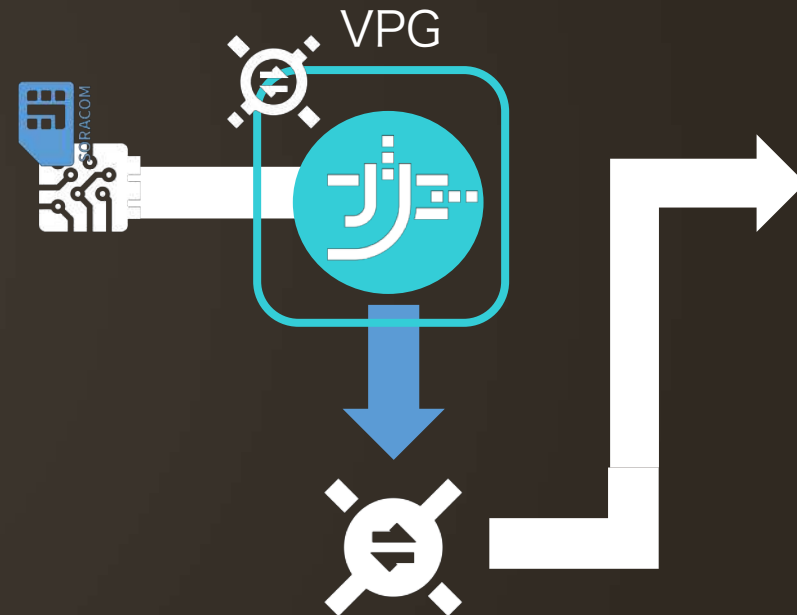
パケットフローを解析して統計情報を出力

Mirroring



パケットのコピーを指定の宛先に転送

Redirection



パケットを指定のゲートウェイ経由で転送



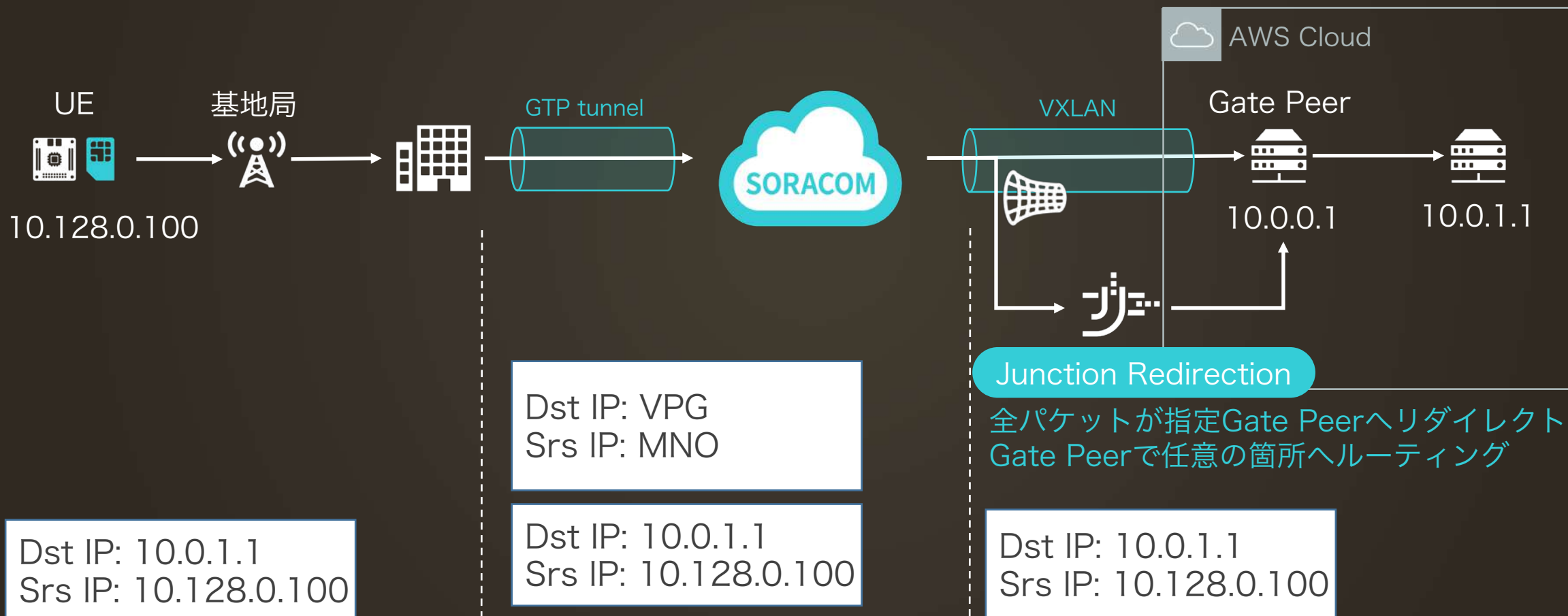
閉域接続+Gate+Junction

- SORACOM Junction - Redirectionを利用することで、全てのパケットを任意のエンドポイントへリダイレクト
- リダイレクト先でパケットをルーティングさせることでGate Peerとは異なるネットワークへのルーティングが可能

Gate利用時のパケット遷移

SORACOM
Technology Camp 2018

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アクセスパターン実装時のポイント

- ネットワーク設計

- 閉域接続先のネットワークアドレス重複などがないようにアドレス設計は入念に！
- Junction Redirectionを利用すると全てのパケットがGate Peerを経由するため、インターネットへの出口も自社ネットワーク内に持つ必要がある

- デバイス/アプリケーション設計

- IPでアクセスできる必要がある
- 常時リクエストに答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。
- 一度に大量のデバイスを制御する際は場合によっては不向き

IPアクセスパターンのまとめ

- ソラコムサービスを組み合わせることで、最初からプライベートなネットワーク空間を利用可能
- 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
- デバイスを指定した通信や、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)

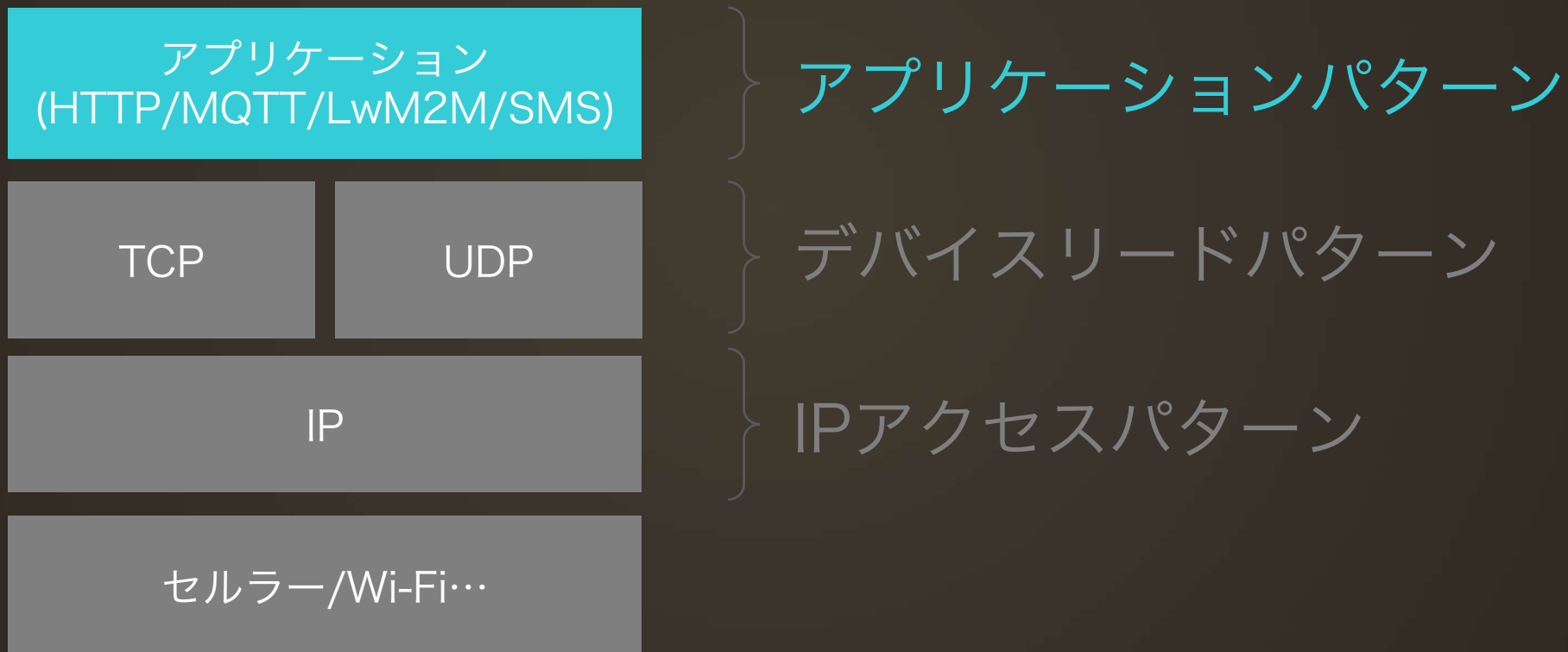
ちなみに

- ポートフォワーディングやエンドツーエンドでのVPNトンネリングも可能

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



セキュアプロビジョニング



SORACOM
Krypton

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



セキュアプロビジョニング



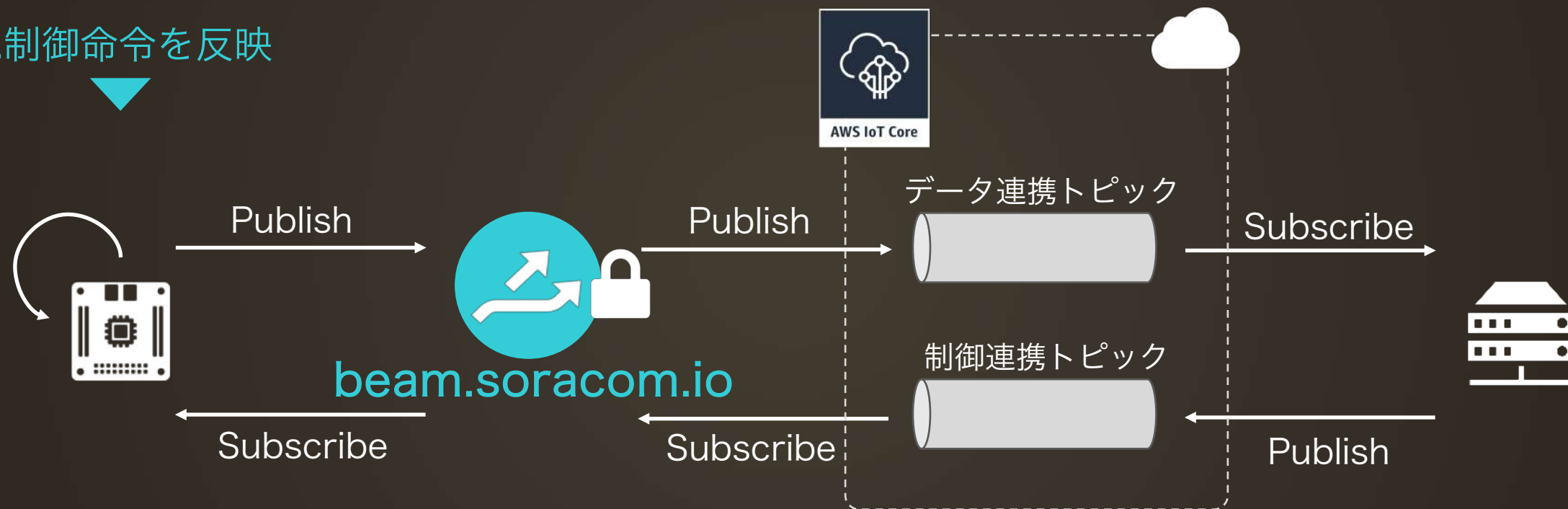
SORACOM
Krypton

SORACOM Beam データ転送支援



SORACOM Beam x MQTT

3.制御命令を反映



2.制御命令をSubscribe

1.制御命令をPublish



MQTT全般で気をつけたいこと

- 消費電力の観点で必ずしもMQTTが有利とは限らない
 - ペイロードが大きいデータをやり取りしようとしていないか
 - MQTTのヘッダサイズは2byte(固定値)だが、そもそもペイロードが大きくなっていないか
 - 通信量が想定以上に大きくなっていないか
 - QoSによるデータ到達保証を高めようとするほど通信回数は増えていくため、電力消費量への影響あり
 - KeepAliveによるPINGREQ/PINGRESPも考慮にいれているか

MQTTにしても勝手に省電力にならない!!

MQTT KeepAliveと通信量

MQTT Client



MQTT Broker



CONNECT

CONNACK

PUBLISH

SUBSCRIBE

SUBACK

PINGREQ

PINGRESP

KeepAliveタイマー

PINGREQ

PINGRESP

...

KeepAliveを有効にした場合

KeepAliveタイマー値を設定するとコネクション確立後にセッション監視のための通信が発生(PINGREQ/PINGRESP)



KeepAliveのための通信が発生するためバッテリー消費や通信量の増加につながる

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



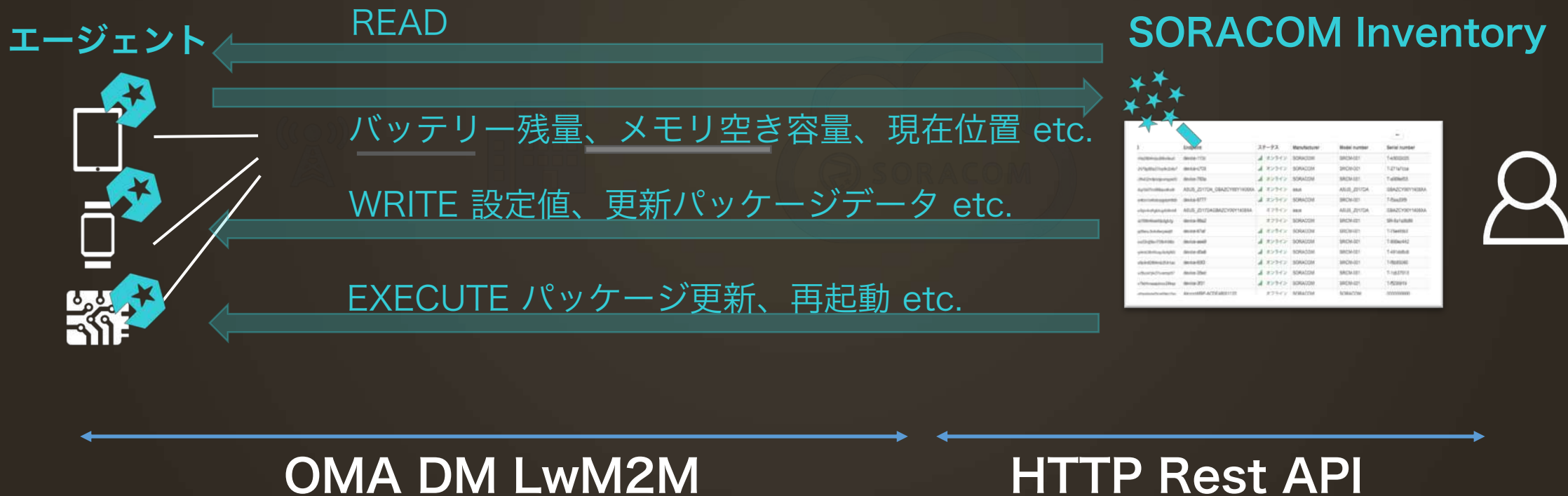
セキュアプロビジョニング



SORACOM
Krypton

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



OMA DM LwM2M



• OMA DMとは？

- Open Mobile Alliance (OMA)で定められた
デバイス管理の業界標準
- 通信プロトコルとデバイス管理オブジェクトを規定

• OMA DM LwM2Mとは？

- OMAが定めたIoT/M2M向けのデバイス管理標準
- 制約のあるデバイスでも適用可能なプロトコルを採用

プロトコルスタック

アプリケーション	オブジェクト
プレゼンテーション	LwM2M
セッション /暗号化	CoAP
	DTLS
トランスポート	UDP
ネットワーク	IP
データリンク/物理	LTE/Wi-Fi///

モデル定義をもとにしたデバイス管理

Constrained Application Protocol, UDP
ベースの軽量なデータ転送プロトコル

TLSのUDP向け実装

オブジェクトモデル

READ

/3303/0/5700



Sensor Value

Device

ObjectID

ObjectInstanceID

ResourceID

Temperature(3303)

0

1

2

...

Sensor Value
(5700)Units
(5701)Min measured value
(5601)

...

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

READ

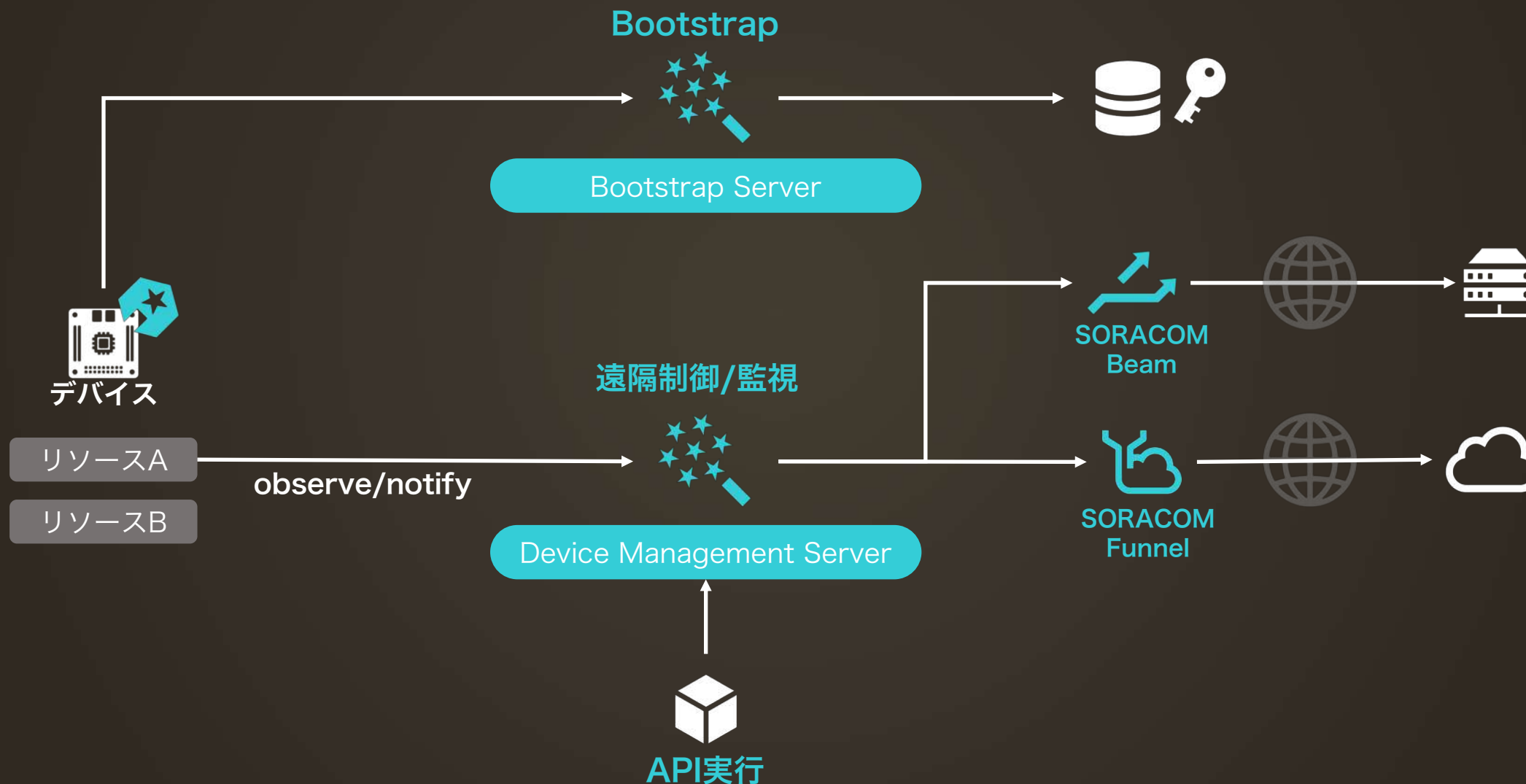
/3/0/3 -> Firmware Version

EXECUTE

/3/0/4 -> Reboot

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

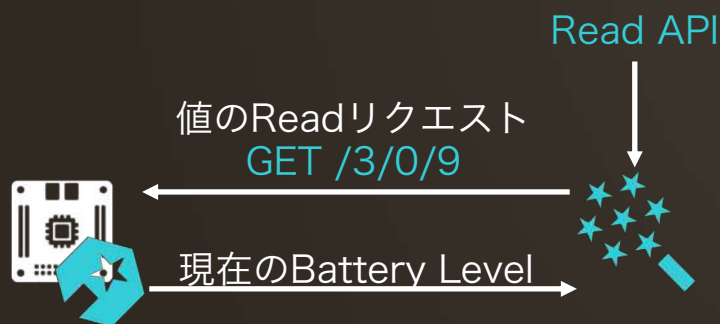
Inventory + Beam/Funnel/Krypton



Inventoryによる基本操作

- デバイスへの命令はInventory APIにて実行する
- デバイスへはLwM2Mによるリクエストが実行される
`/ObjectID/ObjectInstanceID/ResourceID`でリソースを指定しリクエスト実行

デバイス情報の取得



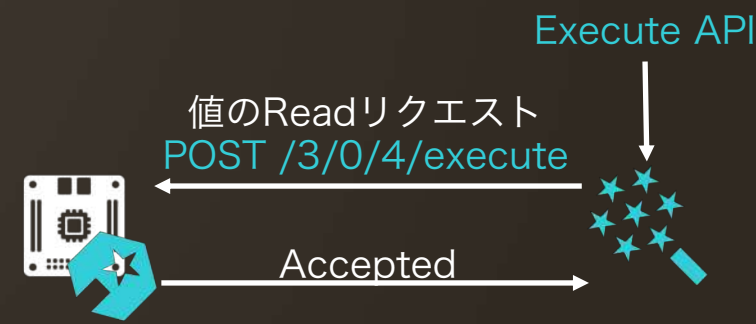
例) Battery Level(/3/0/9)を取得

デバイス設定の変更



例) UTC Offset(/3/0/14)を更新

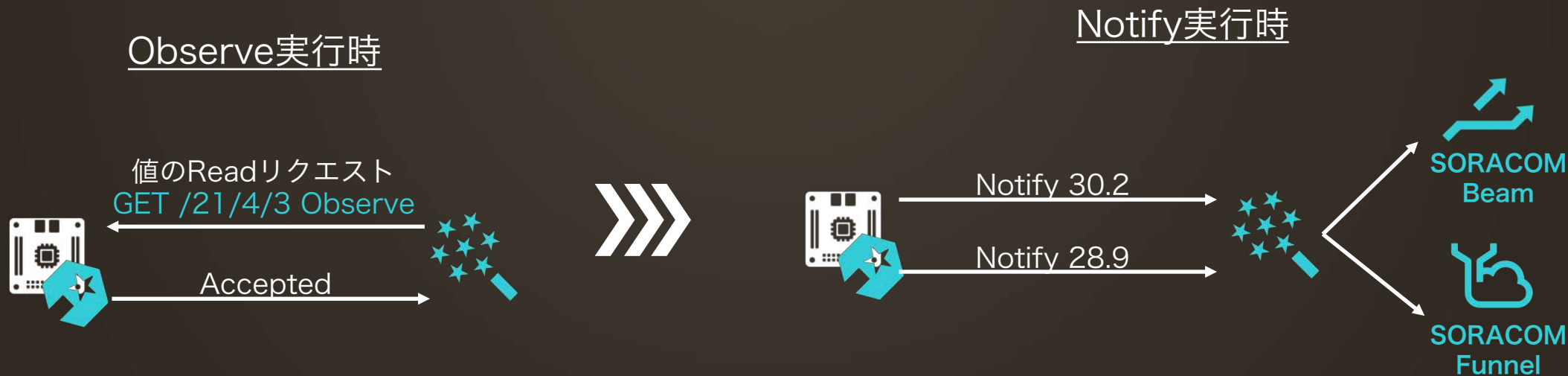
コマンドの実行



例) Reboot(/3/0/4)を実行

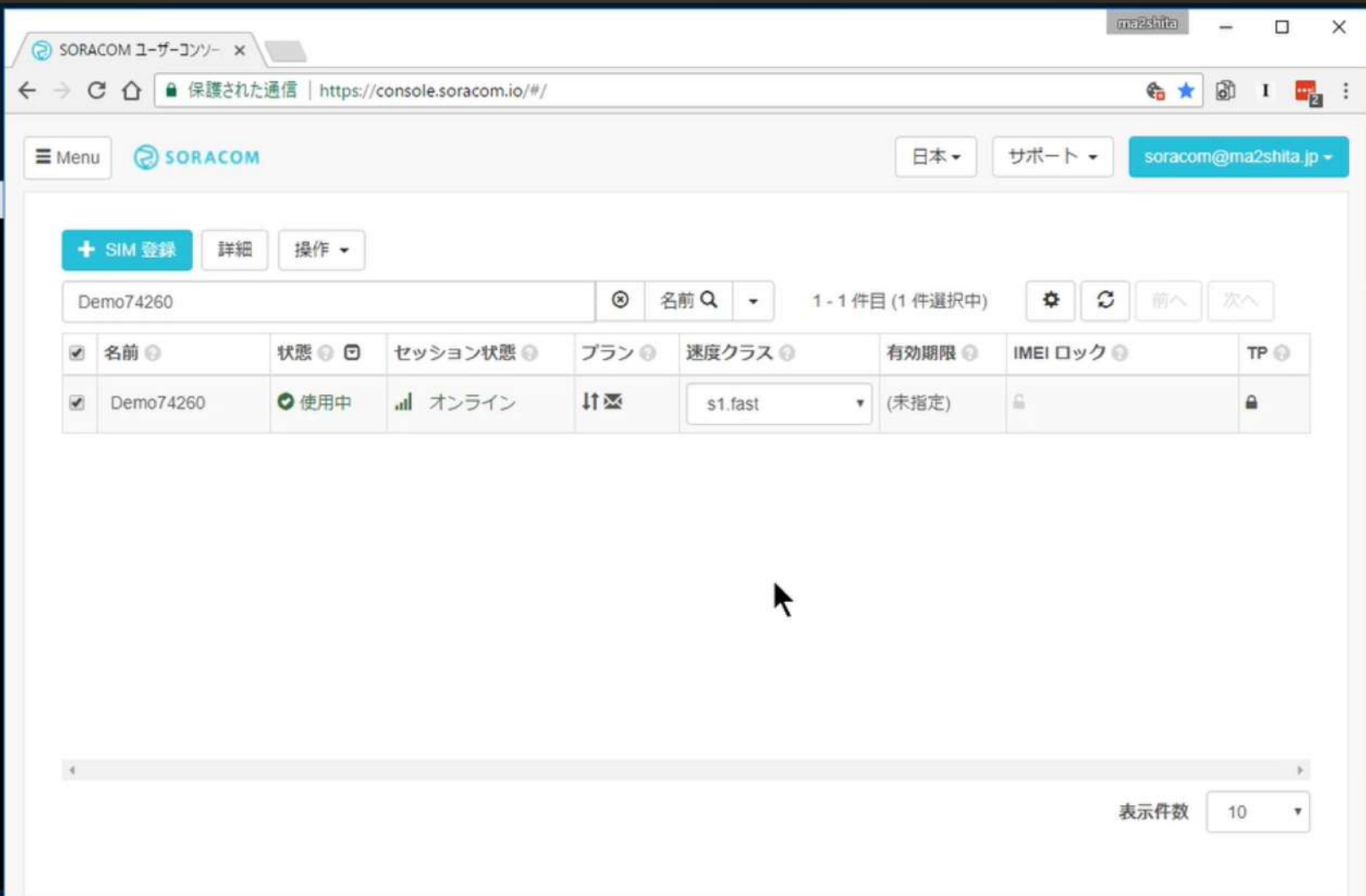
Observe/Notify

- 特定のリソースをObserveすることで、値の変更を起因にしてデータをサーバサイドへプッシュ通知(Notify)
- SORACOM InventoryではNotifyされたデータをSORACOM Beam/Funnel/Harvestへ連携可能



SORACOM Inventory 基本操作

```
pi@raspberrypi: ~/inventory/c/build  
pi@raspberrypi: ~/inventory/c/build $  
pi@raspberrypi: ~/inventory/c/build $
```



SORACOM ユーザーコンソール

保護された通信 | https://console.soracom.io/#/

Menu SORACOM 日本 サポート soracom@ma2shita.jp

+ SIM 登録 詳細 操作

Demo74260 名前 1 - 1 件目 (1 件選択中)

名前	状態	セッション状態	プラン	速度クラス	有効期限	IMEI ロック	TP
Demo74260	使用中	オンライン		s1.fast	(未指定)		

表示件数 10



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



セキュアプロビジョニング



SORACOM
Krypton

SMSによる方式

1. SMS送信APIを実行

POST
/v1/subscribers/{imsi}/send_sms



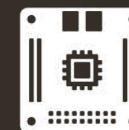
API



SORACOM

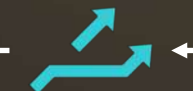
2. SMSをデバイスへ送信

SMSを送信



3. 本文に記載された制御命令
を読み取り & 実行

4. 制御結果の応答を返した
い際はBeamへ送信



SORACOM
Beam



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へSMS経由で送信

SMSでのポイント

- **連携頻度が低いユースケースに最適**
 - SMSの送受信費用は通常のセルラー通信量より割高なため、連携頻度が少ないケースで効果を発揮

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



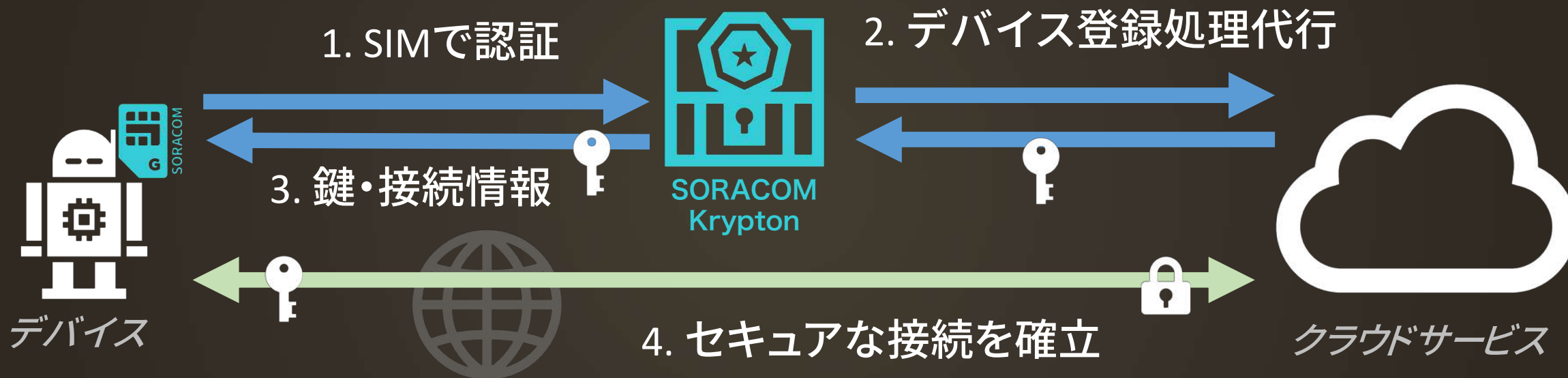
セキュアプロビジョニング



SORACOM
Krypton

SORACOM Krypton

— SIMで認証、接続情報をセキュアにプロビジョニング

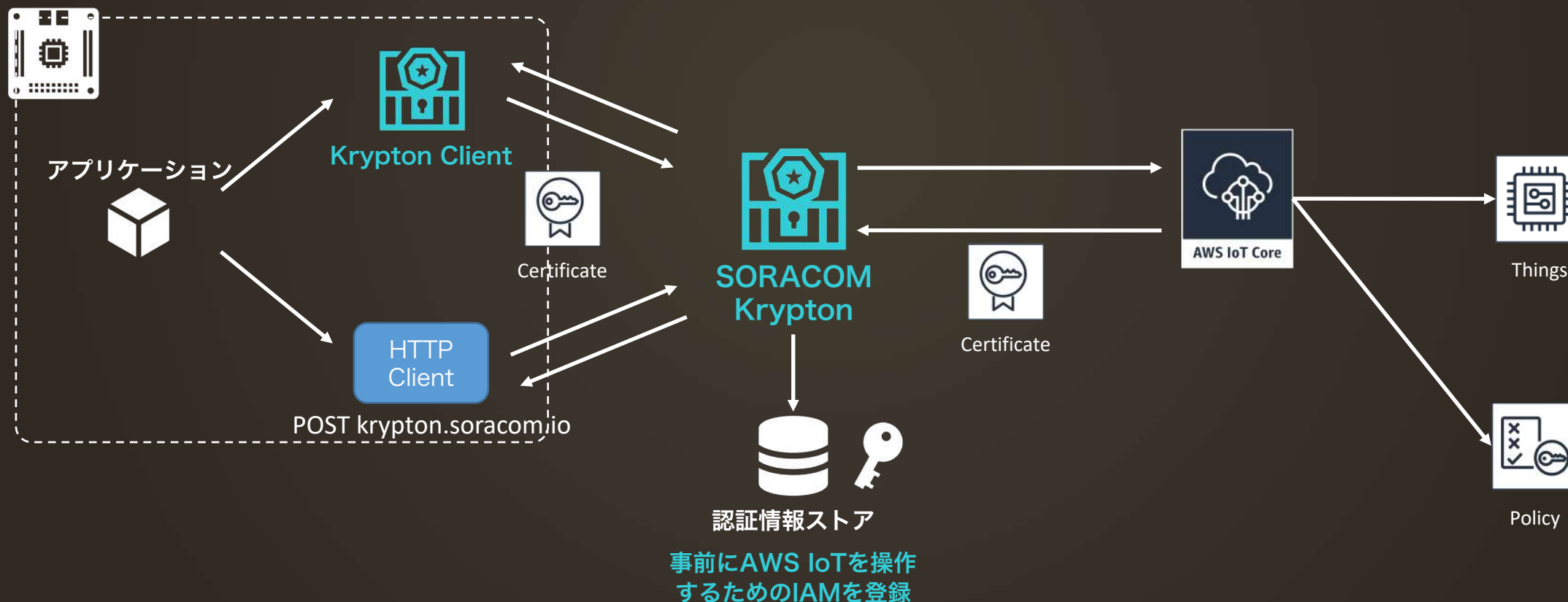


認証には強化された SORACOM Endorse を活用した2つの方法

- 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まとめ

- MQTT,LwM2MとBeam/Inventoryを組み合わせることにより柔軟な連携方式を実現可能
- 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
- 作り込みによってはいくらでも複雑にできてしまうので、シンプルな設計/実装に常に気を付ける

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



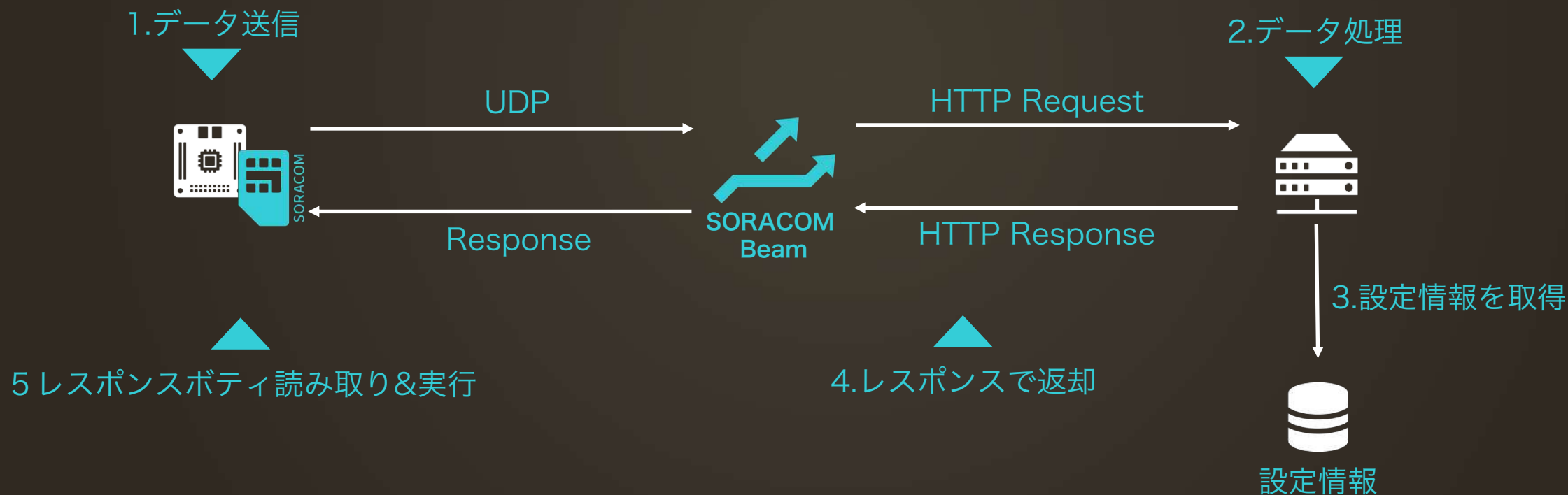
SORACOM
Beam

メタデータ API



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 &jjson_root = jsonBuffer.parseObject(receiveData);
value01 = jjson_root["key01"];
value02 = jjson_root["key02"];
```

データ送信

レスポンス受信

取得データ反映

デバイスリードパターン x SORACOM

データ転送支援



SORACOM
Beam

メタデータ API



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
```

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

- 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

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。

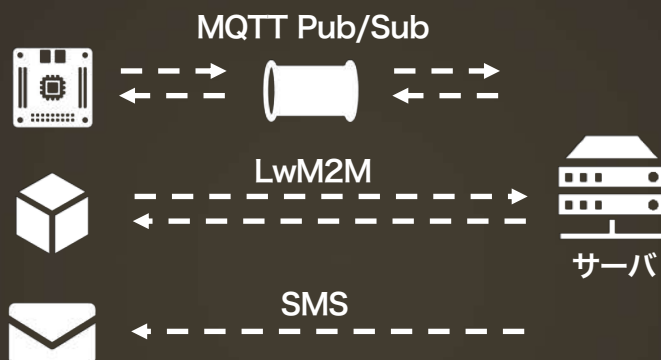
メリット

- 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

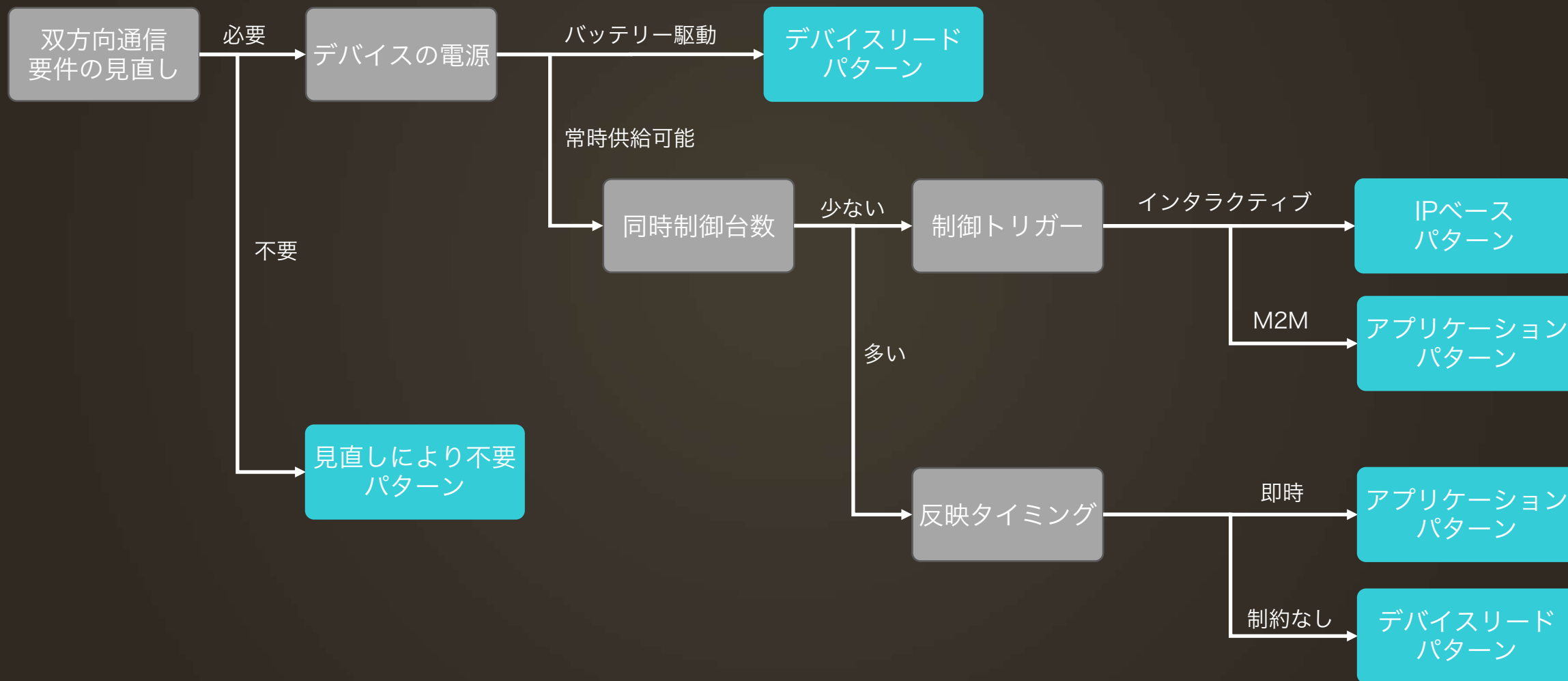
メリット

- 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

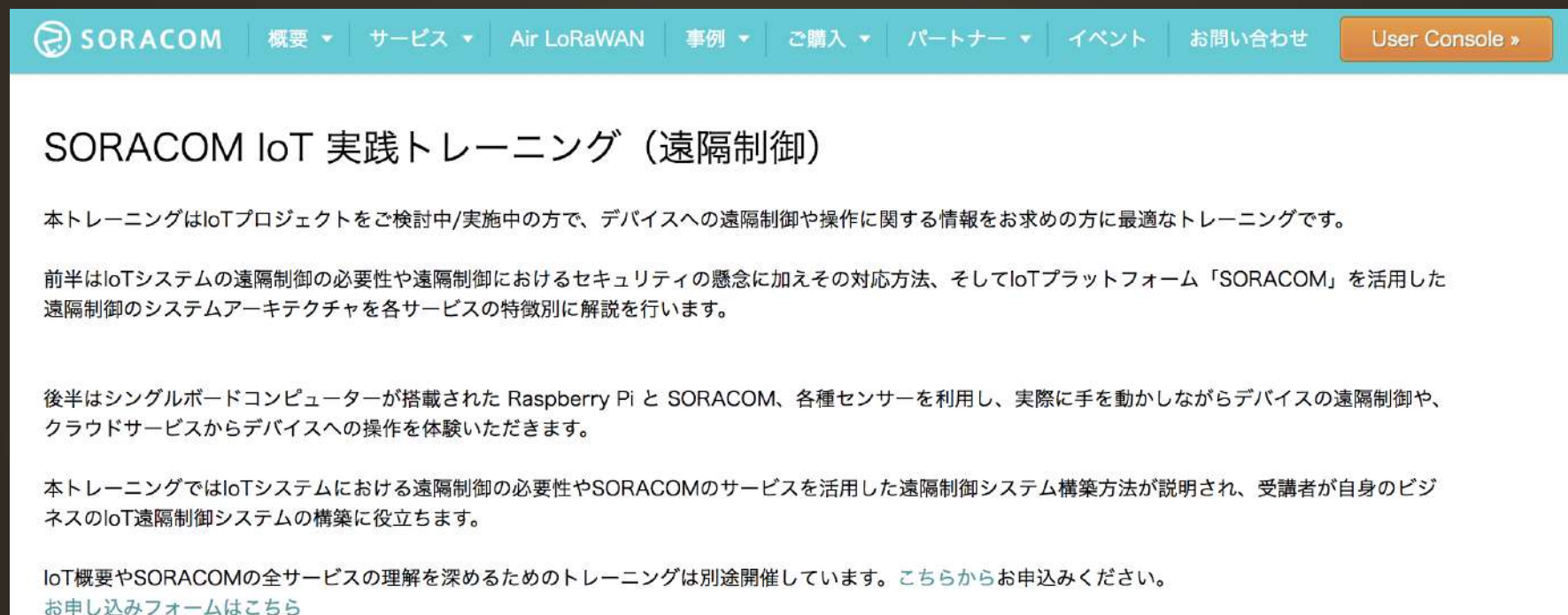
- その双方向通信が本当に必要かを再度検討する
 - 検討ポイント、実装が複雑になるのを避ける
- 目的に対して適切な方式を選択する必要がある
 - 対象台数や制御内容、対象デバイスなどを考慮
- 双方向通信とトレードオフになる要件を把握
 - 消費電力、設計や実装の難易度
- 双方向通信は必ずしも下り通信を必要としない。
 - デバイス主導で実現する方式案も検討する

総まとめ

- 双方向通信によりデバイス-サーバ/デバイス-デバイスでのシステムチックな連携が可能
- 一方で双方向通信ならではの難しさもあり、設計/実装はシンプルになるように心がける
- ソラコムサービスを活用していただくことで、双方向通信をよりシンプルに！

実践トレーニング

双方向通信(遠隔制御)のトレーニングもぜひご活用ください！



The screenshot shows the SORACOM website header with navigation links: SORACOM, 概要, サービス, Air LoRaWAN, 事例, ご購入, パートナー, イベント, お問い合わせ, and a User Console button. The main heading is "SORACOM IoT 実践トレーニング (遠隔制御)". The text describes the training content, which includes understanding the necessity of remote control, security, and the use of SORACOM's IoT platform. It also mentions a hands-on session using Raspberry Pi and various sensors. The page concludes with a link to the training registration form.

SORACOM IoT 実践トレーニング (遠隔制御)

本トレーニングはIoTプロジェクトをご検討中/実施中の方で、デバイスへの遠隔制御や操作に関する情報をお求めの方に最適なトレーニングです。

前半はIoTシステムの遠隔制御の必要性や遠隔制御におけるセキュリティの懸念に加えその対応方法、そしてIoTプラットフォーム「SORACOM」を活用した遠隔制御のシステムアーキテクチャを各サービスの特徴別に解説を行います。

後半はシングルボードコンピューターが搭載された Raspberry Pi と SORACOM、各種センサーを利用し、実際に手を動かしながらデバイスの遠隔制御や、クラウドサービスからデバイスへの操作を体験いただきます。

本トレーニングではIoTシステムにおける遠隔制御の必要性やSORACOMのサービスを活用した遠隔制御システム構築方法が説明され、受講者が自身のビジネスのIoT遠隔制御システムの構築に役立ちます。

IoT概要やSORACOMの全サービスの理解を深めるためのトレーニングは別途開催しています。[こちらからお申込みください。](#)
[お申し込みフォームはこちら](#)

https://pages.soracom.jp/20190129_training_remote_control.html

開発者サイト・ブログのご紹介

開発者サイト



<https://dev.soracom.io/jp/>

各サービスのGetting Started
を用意しています

SORACOM ブログ



<https://blog.soracom.jp/>

最新の技術情報アップデートを
いち早くお届けします

Try! SORACOM チャレンジキャンペーン



～記事を書いてノベルティをゲットしよう～

期間：11月22日～12月25日

対象：企業・個人問わずどなたでもご参加いただけます

内容：SORACOM を使った電子工作やDeep Diveな記事を書いてくださった方に素敵なノベルティをプレゼントします



SORACOM Advent Calendar 2018



今年は2種類！のアドベントカレンダーを開催

1. SORACOM LTE-M Button powered by AWS
2. SORACOM に関する内容ならなんでもOK

Qiita フォローしてね！

執筆者も絶賛募集中



今日からIoTを始めよう！本日ホワイエで販売中



Grove IoT スターターキット for SORACOM

7種類のGroveセンサーとSIMが搭載可能なデバイスで
素早くプロトタイピングが可能

(本日販売価格：15,980円) ※送料分お得



SORACOM LTE-M Button powered by AWS

セルラー通信のLTE-Mを搭載し
購入後すぐに使い始めることが可能

(本日販売価格：3,980円) ※送料分お得

ナイトイベントはボタンLT大会

時間：18:15– 19:30

ソラコム社員も多数参加！ぜひ交流を深めましょう！



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

世界中のヒトとモノをつなげ
共鳴する社会へ



SORACOM