

SORACOMで実現するデバイス管理

株式会社ソラコム ソリューションアーキテクト
今井雄太(factory)

2018/4/26

本日のハッシュタグ

#soracom



@SORACOM_PR



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

アジェンダ

Topic1: デバイス管理って？ どうしてデバイス管理が必要になるの？

Topic2: SORACOMで実現するデバイス管理のケーススタディ

Topic3: まとめと結論

Topic1:

デバイス管理って？

どうしてデバイス管理が必要になるの？

例えば GPSによる車両管理

例えば GPSによる車両管理



アプリケーションの本分は
「車両に搭載したGPSトラッカーから
1分おきに緯度経度を送信、地図上にマッピ
ングする」

「GPSによる車両管理」におけるデバイス管理

SIMをデバイスに挿して設置したら勝手に動き始めるようにしたい

SIMと車両のヒモ付はどうすれば？

そもそもの話として、新規車両の発注とワークフローを統合したい



クラウド



可視化/監視&通知



いまオンラインの車両は？

それぞれの車両が最後にオンラインになっていた時間を知りたい

SIMが盗まれても別の用途で使われたりしたくない

例えば
スマートメーター

例えば
スマートメーター



アプリケーションの本分は
「1日に1回、ガスの使用量をクラウドに送信。ガス利用者自身が自分の利用量を把握できるようにする。」

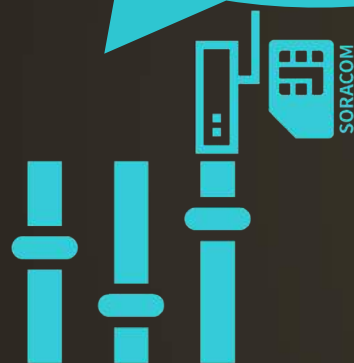
「スマートメーター」におけるデバイス管理

GPS車両管理での要件はほぼそのまま当てはまる。更に。

在庫としてストックしていたデバイスが利用開始されたのタイミングを知りたい

設置先固有の情報を、起動時に初期設定したい

メーターの設定値を変更したい



クラウド



可視化/監視&通知



メーターのオンライン/オフラインのタイミングを把握したい

1週間以上オンラインになっていないデバイスの検出をしたい

通信機器ごと盗まれたらどうする？

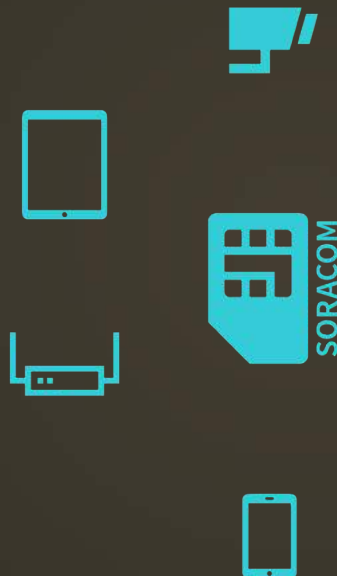
それ以外にも . . .

他にも「デバイス管理」といえば・・・

デバイスごとの通
信量を把握したい

デバイスの通信先
を制限したい

デバイスの利用可
能なプロトコルを
制限したい



デバイスにクライ
アント証明書や鍵
を安全に配布する
には？

(できれば) そう
いった情報をデバ
イスに持たせたく
ない

というかそもそも、
デバイスのIDはど
うやって管理すれ
ばいい？

「デバイス管理」

言い換えると

「デバイスが本来の仕事をできるようにする
ために必要な準備やメンテナンス、監視」

IoTデバイス管理あるある

- 使い始める前

- 発注
- デバイスとSIMのヒモ付
- デバイスの初期設定

- 使い始めた後：正常系

- 通信量やデバイスの状態の把握
- 設定値や挙動の管理

- 使い始めた後：異常系

- 通信利用用途の制約付け
- 異常な動作の検知

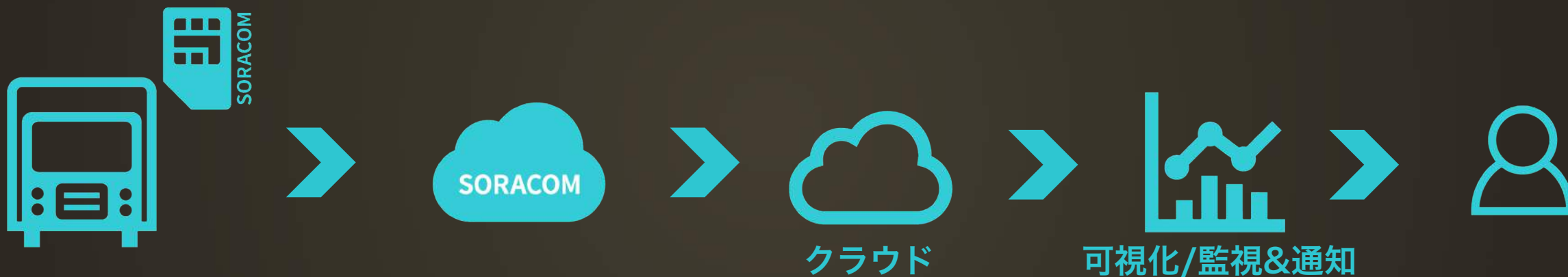
デバイスの数が少なければ手動でも対応可能
ですが、数百/数万になったら・・・？

デバイスの数が少なければ手動でも対応可能
ですが、数百/数万になったら・・・？

自動化しないとまわりませんよね？
そう、SORACOMならできるんです。

Topic2:
SORACOMで実現する
デバイス管理のケーススタディ

ケーススタディ 「GPSによる車両管理」



ワークフローを考えてみる

新しく車両にGPSトラッカー（デバイス）を設置する
には・・・

1. デバイスを発注
2. SIMを発注
3. デバイスとSIMのヒモ付
4. デバイスの初期設定&動作確認
5. (在庫としてストック)
6. 発送&設置
7. 利用開始

ワークフローを考えてみる

新しく車両にGPSトラッカー（デバイス）を設置する には・・・

1. デバイスを発注
2. SIMを発注
3. デバイスとSIMのヒモ付
4. デバイスの初期設定&動作確認
5. (在庫としてストック)
6. 発送&設置
7. 利用開始

SIMを発注する

これまでは・・・

デバイスとSIM、別々に発注するのは手間だし、ミスが紛れこむ可能性がありました。

SORACOMなら・・・

APIで発注できるので、みなさんの業務システムから発注が可能です。

SIMを発注する 見積もりの作成

\$ soracom orders create ¥

--body ¥

```
'{"orderItemList":[{"productCode":"4573326590013","quantity":1}],"shippingAddressId":"2d5f168b-7d60-480f-8ef7-xxxxxxxxxxxxx"}'
```



```
{  
  "orderId": "20180424062105_xxxxxxxx",  
  "totalAmount": 1555,  
  ...  
}
```

SIMを発注する

発注

```
$ soracom orders confirm ¥  
--order-id 20180424062105_xxxxxxxxxx
```

SIMを発注する

発注IDからSIMのIMSIやシリアルを取得

```
$ soracom orders list-subscribers --order-id 20180424062105_xxxxxxx  
{  
  "orderedSubscriberList": [  
    {  
      "iccid": "8981100024400433712",  
      "imsi": "440103167209666",  
      "msisdn": "07043389150",  
      "serialNumber": "AX0504400433710"  
    },  
    ...  
  ]  
}
```

SIMを発注する

さらに、前項の作業で取得したSIMを・・・

- 社内システムのDBに登録したり
- SORACOM上でタグ付けしたり
- SORACOM上でグルーピングしたり

ワークフローを考えてみる

新しく車両にGPSトラッカー（デバイス）を設置する には・・・

1. デバイスを発注
2. SIMを発注
3. デバイスとSIMのヒモ付
4. デバイスの初期設定&動作確認
5. (在庫としてストック)
6. 発送&設置
7. 利用開始

デバイスとSIMのヒモ付

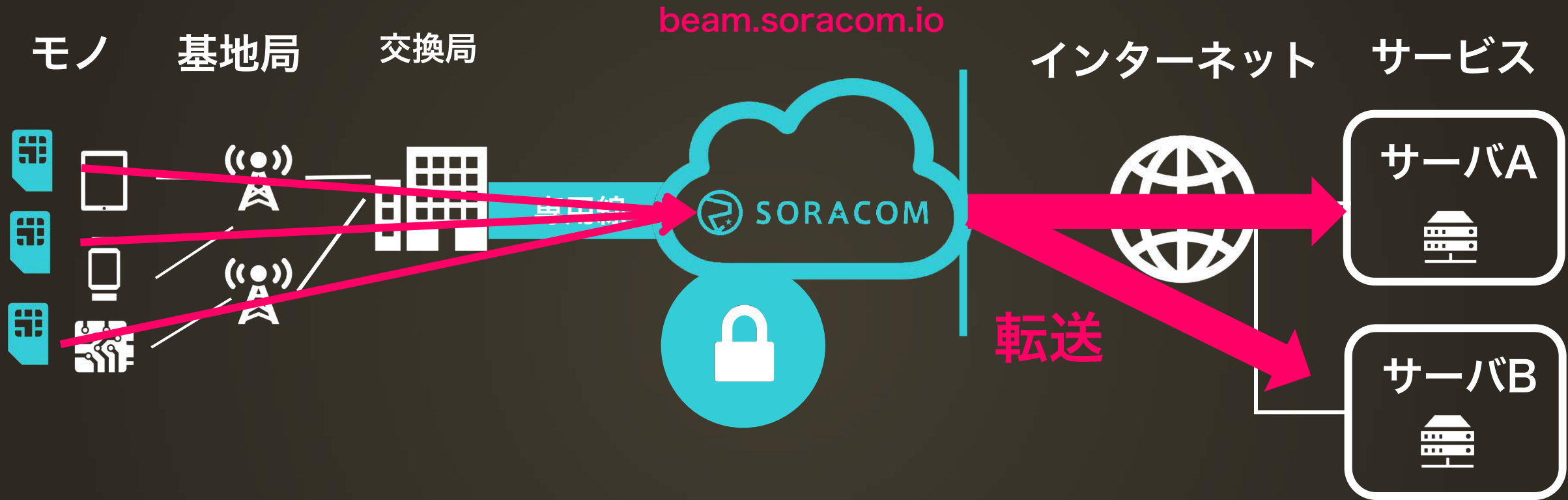
これまでは・・・

（極端ですが）大量のデバイスと大量のSIM（小さい）を、管理簿に基づいて手作業でマッチング&SIM挿入。

SORACOMなら・・・

それ、SORACOM Beamで自動化できます！

SORACOM Beam – データ転送支援



クラウドの潤沢な
リソースで暗号化・変換

HTTP->HTTPS
MQTT->MQTTS
TCP -> TCPS/HTTP/HTTPS
UDP -> HTTP/HTTPS
...

**SORACOM Beamが提供するのは、
データの暗号化&プロキシではありません。**

SORACOM Beamの署名付与機能

- Beamは、転送するリクエストに対して事前共有鍵を使った署名を付与してくれる
- サーバーサイドアプリケーションは、この署名を検証することでリクエストが自分の管理するデバイスからであることが判別できる
- 事前共有鍵はデバイスに配布する必要がないので安全に管理/交換可能



署名の検証方法

IMSI、IMEI、TIMESTAMP、SIGNATUREがそれぞれx-soracom-imsi, x-soracom-imei, x-soracom-timestamp, x-soracom-signature, x-soracom-signature-versionという形でサーバーへのリクエストへ付与される

sha256(

事前共有鍵 +

["x-soracom-imei=IMEI" +]

"x-soracom-imsi=IMSI" +

"x-soracom-timestamp=TS"

)

ヘッダ操作

IMSI ヘッダ付与

ON

HTTP リクエストヘッダーに x-soracom-imsi: \${IMSI} を追加します

IMEI ヘッダ付与

ON

HTTP リクエストヘッダーに x-soracom-imei: \${IMEI} を追加します

署名ヘッダ付与

ON

リクエストに x-soracom-signature: を追加します。少なくとも IMSI か IMEI ヘッダ付与どちらか一方が有効になっている必要があります。

事前共有鍵*

factory-adeven-dev (adeven-senda-dev)

+

カスタムヘッダ

アクション	ヘッダ名	値
データがありません		

+

-

デバイスとSIMのヒモ付



SORACOM Beamを使ったデバイスとSIMのヒモ付

まずはデバイスにSIMを挿して起動



```
$ curl POST http://beam.soracom.io/${YOUR_DEVICE_ID}/register
```



SORACOM Beam



```
POST api.example.com/YOUR_DEVICE_ID/register
```

```
X-Soracom-IMSI: 123456789012345
```



YOUR_DEVICE_IDとIMSI: 123456789012345をヒモ付！

ワークフローを考えてみる

新しく車両にGPSトラッカー（デバイス）を設置する
には・・・

1. デバイスを発注
2. SIMを発注
3. デバイスとSIMのヒモ付
4. デバイスの初期設定&動作確認
5. (在庫としてストック)
6. 発送&設置
7. 利用開始

デバイスの初期設定&動作確認

これまでは・・・

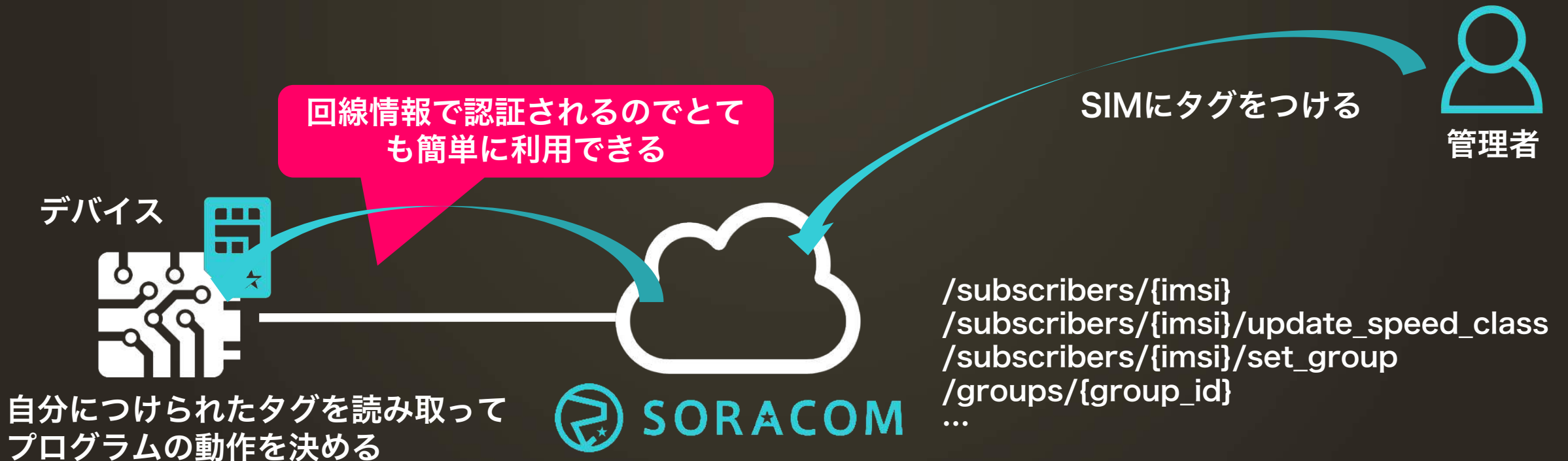
（極端ですが）デバイス一つ一つの初期設定や動作確認を手作業で行って、管理簿を更新。

SORACOMなら・・・

それ、メタデータサービスで（かなりの部分）自動化できます！

SORACOM Air: メタデータサービス

Air SIM を使用するデバイス自身から、Air SIM の情報を HTTP 経由で取得したり、更新のためにAPIを簡単に呼び出すためのサービス。



デバイスの初期設定&動作確認

メタデータサービスを使った初期設定: SIMのIMSIやタグ、名前等を簡単に取得

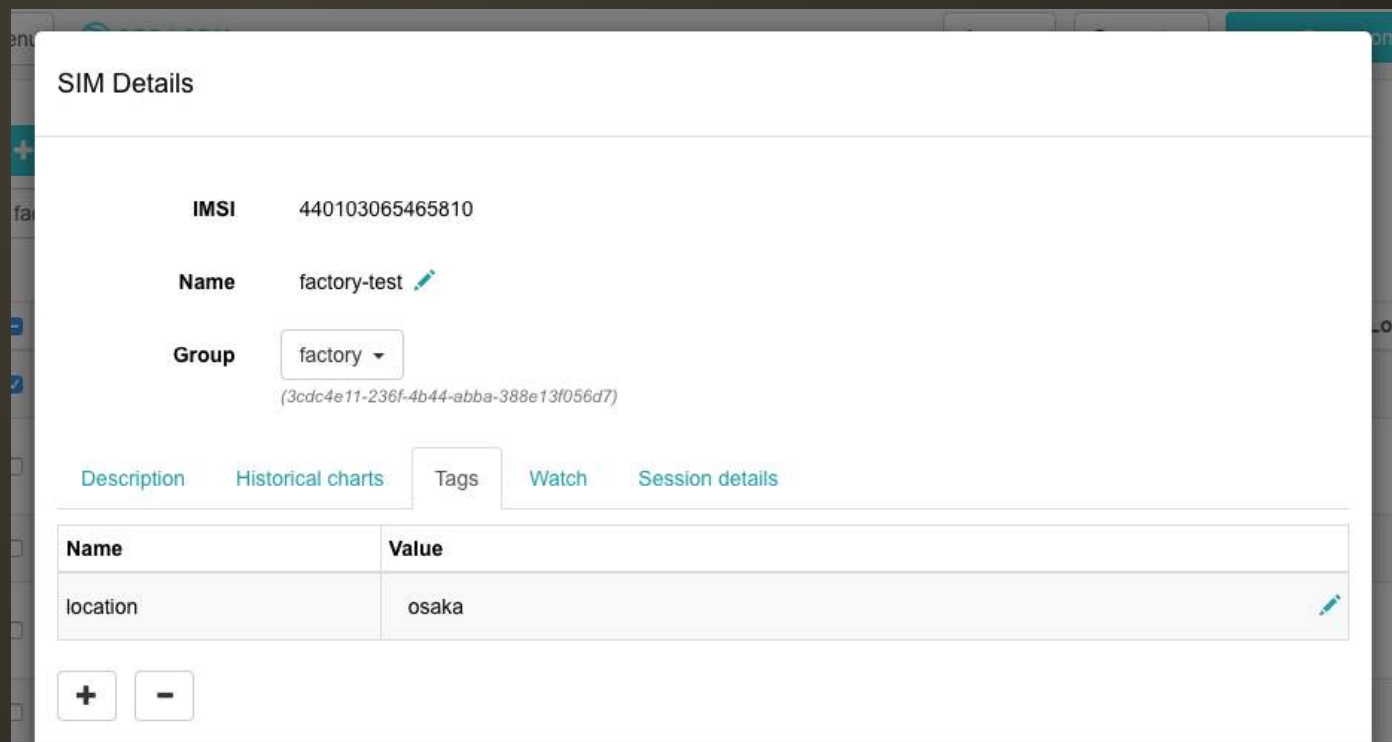
```
$ curl metadata.soracom.io/v1/subscriber
```

```
{  
  "groupId": "270dcdbc-4048-4298-9749-35ee3886ba50",  
  "iccid": "8981100005815107199",  
  "imsi": "440103198521342",  
  "ipAddress": "10.244.46.2",  
  "tags": {    "name": "imaiSIM"  },  
  "type": "s1.fast",  
  ...  
}
```

デバイスの初期設定&動作確認

タグに必要な情報を書き込んでおいて、デバイス起動時に情報を取得させてブートストラップ。スクリプトのURLを書いておくのもあり。

※デバイスからの書き込みもできるので、結果をタグに書いておくのもあり。



SIM Details

IMSI 440103065465810

Name factory-test

Group factory
(3cdc4e11-236f-4b44-abba-388e13f056d7)

Description Historical charts Tags Watch Session details

Name	Value
location	osaka

+ -

デバイスの初期設定&動作確認

以下のようなフローで、SIMとデバイスのヒモ付から初期設定、動作確認まで自動化できます。

1. デバイスは起動したらメタデータから初期設定スクリプトをダウンロード
2. 初期設定スクリプトに沿って、BeamによるSIMとデバイスのヒモ付を行う
3. 更に初期設定スクリプトに沿って、動作確認を実施、結果をクラウドへ送信して記録しておく
4. (デバイスのLEDを光らせたりすると現場での確認作業が楽になってベター)
5. デバイスのバーコードをスキャンすると、上記の状況を確認できる

ワークフローを考えてみる

新しく車両にGPSトラッカー（デバイス）を設置する には・・・

1. デバイスを発注
2. SIMを発注
3. デバイスとSIMのヒモ付
4. デバイスの初期設定&動作確認
5. (在庫としてストック)
6. 発送&設置
7. 利用開始

在庫としてストック

これまでは・・・

在庫期間中のSIMにランニングコストが発生してしまっていましたよね。もしくは、通信会社と個別に課金調整を行っていたと思います。

SORACOMなら・・・

SORACOMのグローバルSIMなら、通信まで含めて動作確認後、ランニングコストがかからない状態にして在庫にしておくことができます。

在庫としてストック

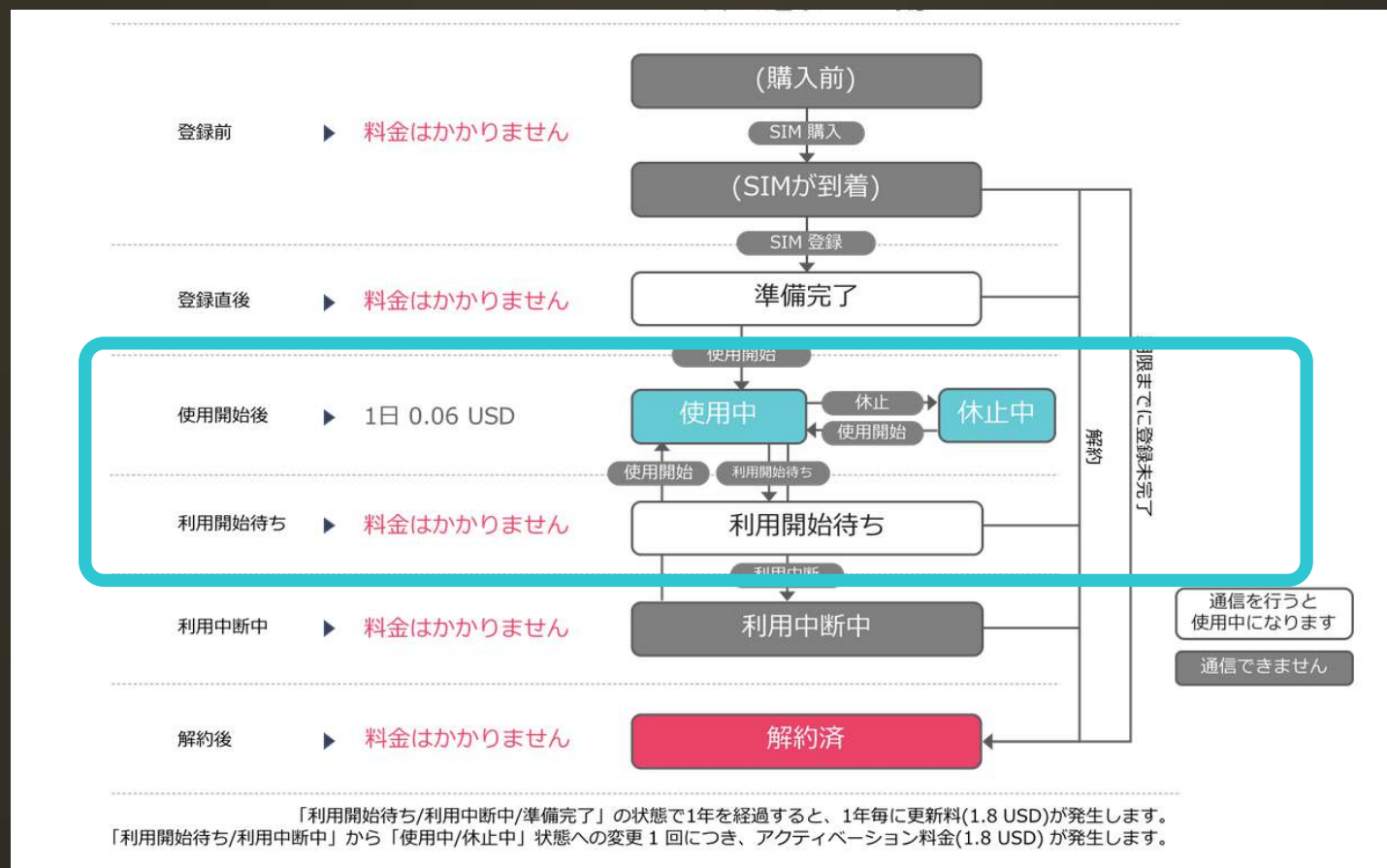
SORACOMのグローバルSIMなら動作確認の後、「利用開始待ち」状態にすることで利用料をゼロに。

基本料金

	plan01s	plan01s - Low Data Volume
状態	準備完了	使用中 / 休止中
基本料金	0 USD / 日	0.06 USD / 日
	利用開始待ち	利用中断中
	0 USD / 日	0 USD / 日

在庫としてストック

SORACOMのグローバルSIMなら動作確認の後、「利用開始待ち」状態にすることで利用料をゼロに。



ワークフローを考えてみる

新しく車両にGPSトラッカー（デバイス）を設置する には・・・

1. デバイスを発注
2. SIMを発注
3. デバイスとSIMのヒモ付
4. デバイスの初期設定&動作確認
5. (在庫としてストック)
6. 発送&設置
7. **利用開始**

利用開始

これまでは・・・

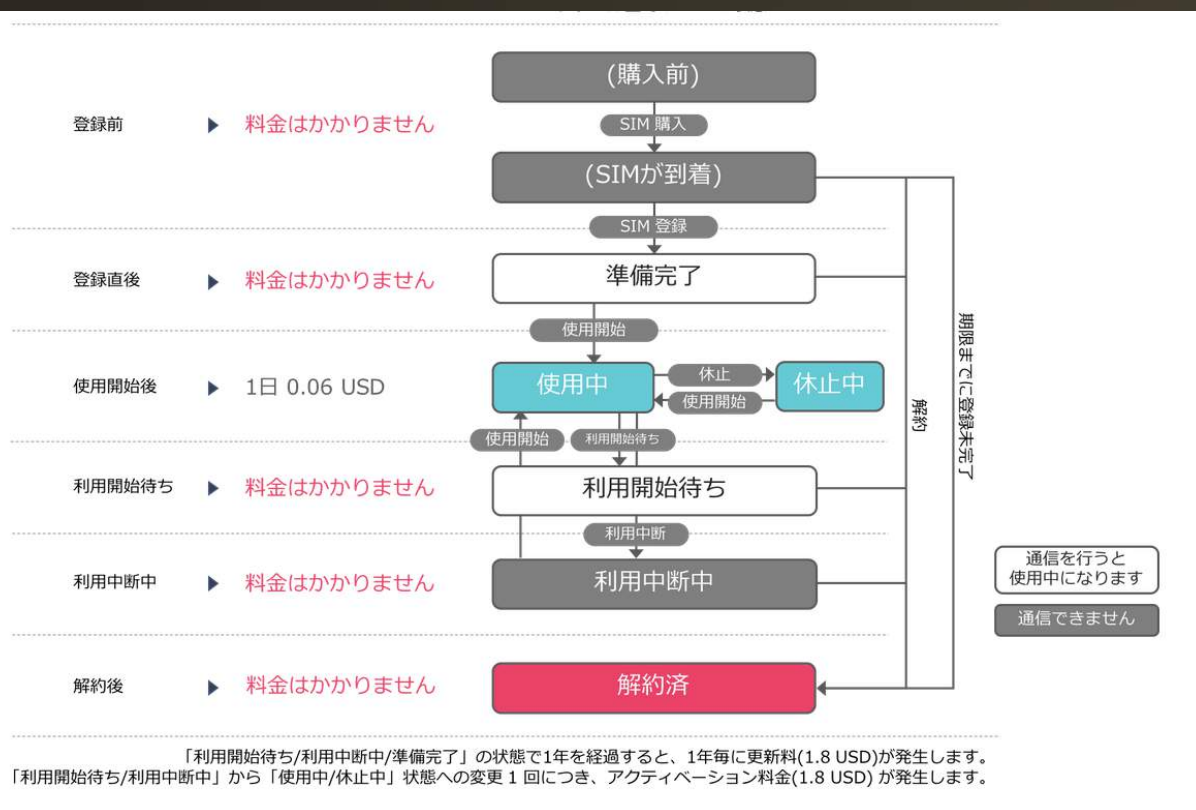
（これも極端ですが） 発送されるデバイスを管理簿でチェックして「利用中」とマークしたりしていたと思います。

SORACOMなら・・・

現場に到着して利用開始（通信開始）されたら、そのデバイスを自動的にマークすることができます。

利用開始

SORACOMのSIMは利用開始にあたって手動作業作業は不要。(国内専用SIMも同様)



通信を行うと
使用中になります

通信を行うと
使用中になります

通信できません

利用開始

勝手に利用開始できるだけでなく、それをシステムに記録しておきたいですね！

利用開始

イベントハンドラを使えば、SIMが「利用中」になったことをトリガにWebhookやAWS Lambdaを呼び出せます！



イベントハンドラとは？

イベントハンドラとは？

こんなトリガーで・・・

DailyTrafficRule	SIMの日のデータ通信量が一定を超えたらアクションを実行するルール
MonthlyTrafficRule	SIMの月のデータ通信量が一定を超えたらアクションを実行するルール
CumulativeTrafficRule	SIMの利用開始からの累積データ通信量が一定を超えたらアクションを実行するルール
DailyTotalTrafficRule	Operatorにひもづく全SIMの日のデータ通信量合計が一定を超えたらアクションを実行するルール
MonthlyTotalTrafficRule	Operatorにひもづく全SIMの月のデータ通信量合計が一定を超えたらアクションを実行するルール
SubscriberStatusAttributeRule	SIMのステータスが変更されたらアクションを実行するルール
SubscriberSpeedClassAttributeRule	SIMの速度クラスが変更されたらアクションを実行するルール
SubscriberExpiredRule	SIMの有効期限が切れた時にアクションを実行するルール

イベントハンドラとは？

こんな処理を呼び出せる機能です！

type

ChangeSpeedClassAction	SIMのSpeedClassを変更するアクション
SendMailAction	メールを送るアクション
SendMailToOperatorAction	オペレーターの登録メールアドレスにメールを送るアクション
ActivationAction	SIMの状態をActive（使用中）にするアクション
DeactivationAction	SIMの状態をInactive（休止中）にするアクション
ExecuteWebRequestAction	指定のURLにリクエストをするアクション
InvokeAWSLambdaAction	AWS Lambdaを呼び出すアクション

イベントハンドラとは？

変数にはこんなものが使えます。

<code>\${imsi}</code>	対象SIMのIMSI
<code>\${operatorId}</code>	対象SIMを持つOperatorのID
<code>\${date}</code>	送信日付(yyyy/m/d)
<code>\${year}</code>	送信年 (yyyy)
<code>\${month}</code>	送信月 (m)
<code>\${date}</code>	送信日 (d)

<code>\${newStatus}</code>	変更後のステータス
<code>\${oldStatus}</code>	変更前のステータス

ワークフローを考えてみる

通常業務

1. 証明書や鍵などの安全な交換
2. 日々のオンライン状態や通信量のレポート作成
3. 設定値やファームウェアの更新

ワークフローを考えてみる

通常業務

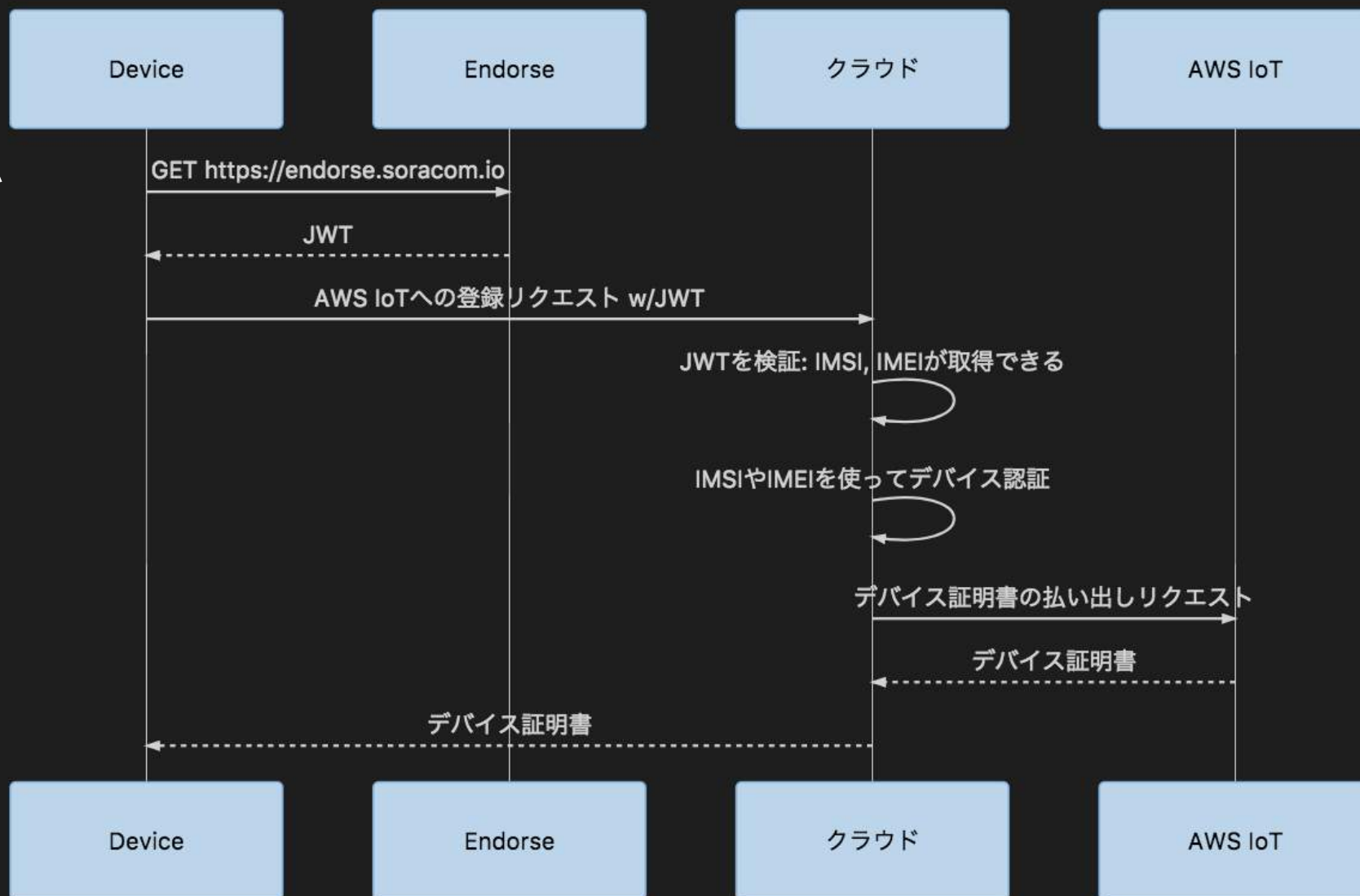
1. 証明書や鍵などの安全な交換
2. 日々のオンライン状態や通信量のレポート作成
3. 設定値やファームウェアの更新

証明書や鍵などの安全な交換

- 証明書や鍵は何に使う？
 - デバイスのIDを証明したり、安全な暗号化経路を作成するためにつかう
- 鍵交換のむずかしさ？
 - 正しく鍵交換するためには、デバイスIDでの認証や、暗号化された経路が必要。コロンブスの卵・・・
- これまでは・・・
 - （厳密にやるなら）工場出荷時にオフラインで鍵を配布
- SORACOMなら・・・
 - SIMをデバイス証明のための鍵として利用可能です！

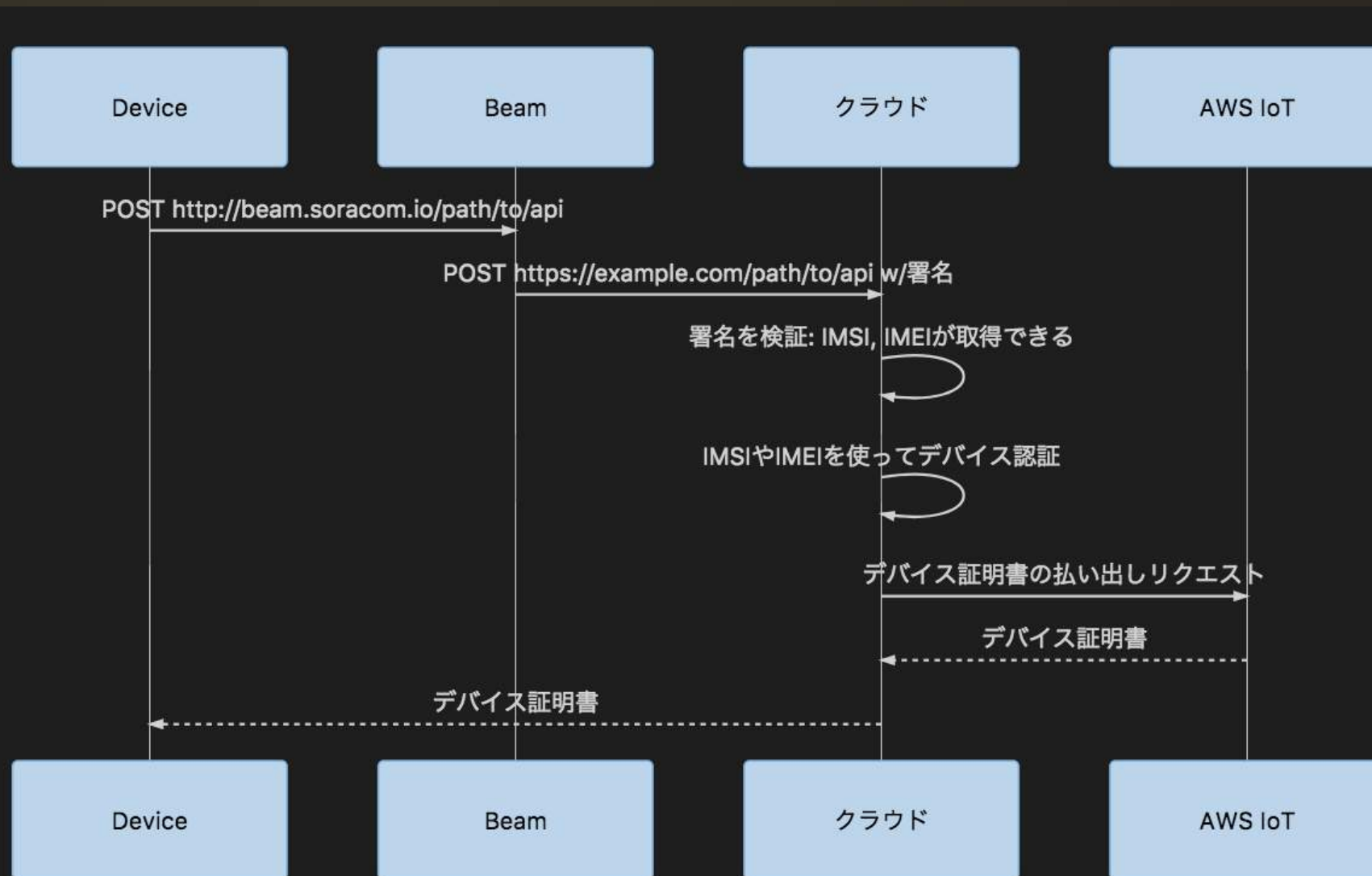
証明書や鍵などの安全な交換

例えばSORACOM
Endorseを使って
AWS IoTの証明書を
デバイスに安全に配
布する



証明書や鍵などの安全な交換

例えばSORACOM
Beamを使ってAWS
IoTの証明書をデバイ
スに安全に配布する



証明書や鍵などの安全な交換

SORACOM BeamやSORACOM Endorseを使うと、みなさん自身のサーバーアプリケーションでSIMベースの認証が利用可能です。

SIMとデバイスのIDがヒモ付されていれば・・・

SIMによるデバイスの認証が可能に！

ワークフローを考えてみる

通常業務

1. 証明書や鍵などの安全な交換
2. 日々のオンライン状態や通信量のレポート作成
3. 設定値やファームウェアの更新

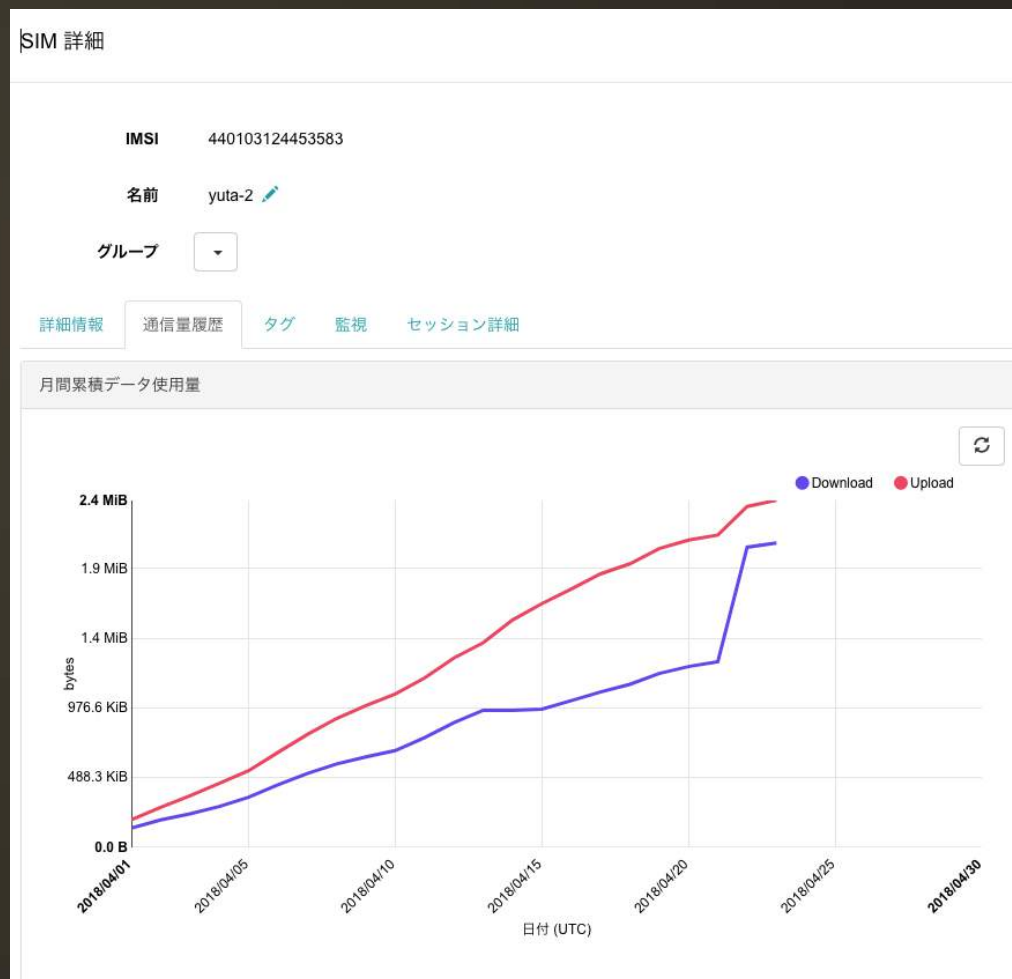
日々のオンライン状態や通信量のレポート作成

- SIM:デバイス=1:1という関係性が成り立っていれば・・・
 - 通信量履歴から、時系列のデバイスのアクティビティが確認できる。
 - セッション履歴から、デバイスのヘルス状態が確認できる。

日々のオンライン状態や通信量のレポート作成

- SORACOMではSIM単位/アカウント単位で簡単に利用量の確認ができます。

SIM単位で



日々のオンライン状態や通信量のレポート作成

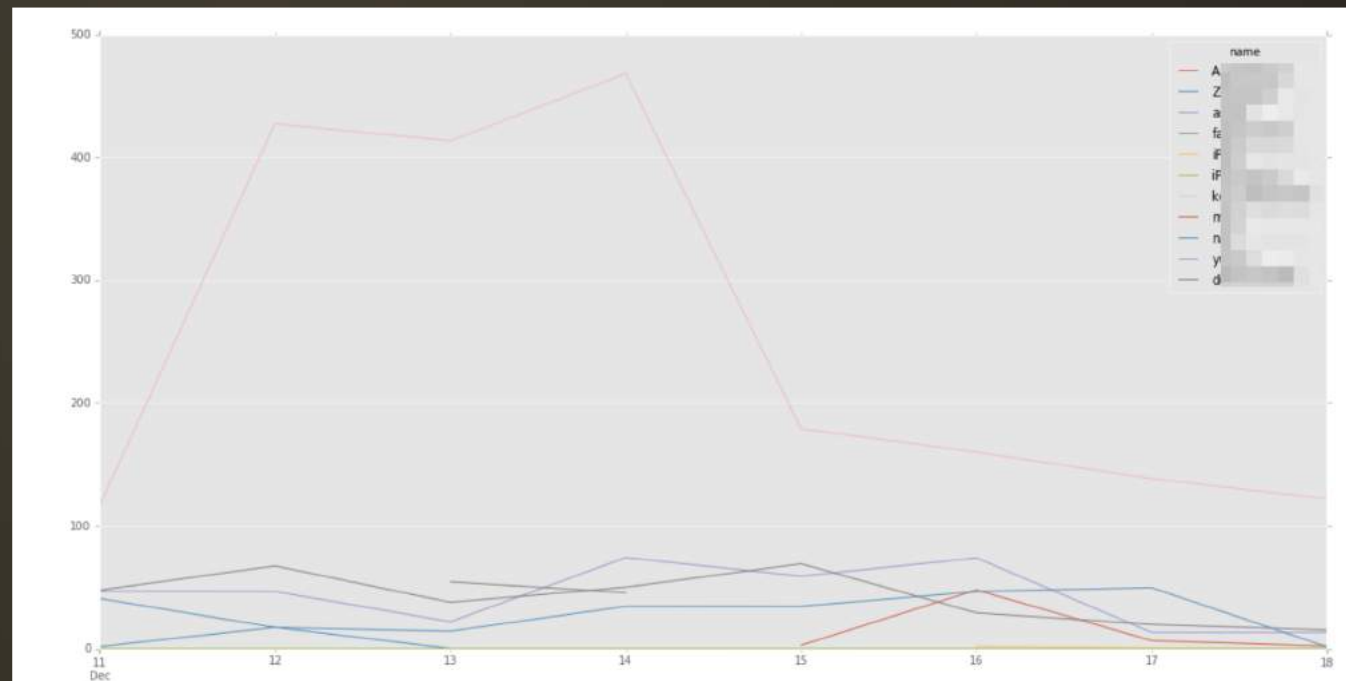
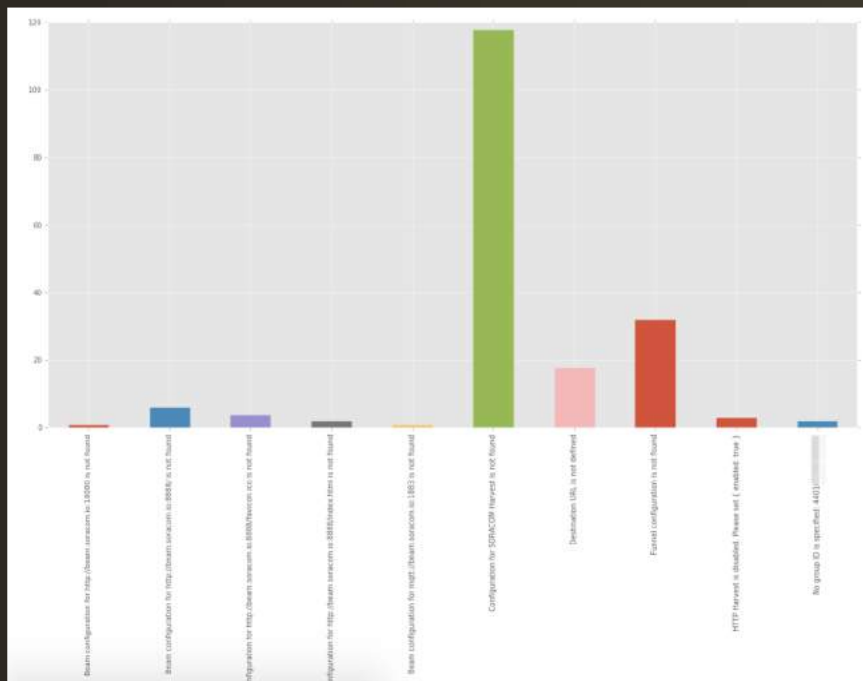
- SORACOMではSIM単位/アカウント単位で簡単に利用量の確認ができます。

アカウント単位で (CSV)

	A	B	C	D	E	F	G
1	imsi	date	type	uploadByteSizeTotal	uploadPacketCountTotal	downloadByteSizeTotal	downloadPacketCountTotal
2	440103056157506	20180123	s1.fast	754	7	658	8
3	440103056157506	20180124	s1.fast	1508	14	1036	12
4	440103056157506	20180125	s1.fast	1508	14	518	6
5	440103056157506	20180126	s1.fast	754	7	518	6
6	440103056157506	20180127	s1.fast	1751	16	1351	16
7	440103056157506	20180128	s1.fast	3942	37	3669	45
8	440103056157506	20180129	s1.fast	5882	50	6976	45
9	440103056157506	20180130	s1.fast	1508	14	5347	34
10	440103056157506	20180131	s1.fast	754	7	518	6

日々のオンライン状態や通信量のレポート作成

- もちろん、APIでデータ取得できるので、例えばJupyter Notebookで集計/分析もできます。



詳細とサンプルコードはこちら: <http://bit.ly/2JhueJN>

ワークフローを考えてみる

通常業務

1. 証明書や鍵などの安全な交換
2. 日々のオンライン状態や通信量のレポート作成
3. 設定値やファームウェアの更新

設定値やファームウェアの更新

- デバイスへの書き込みにはやり方がたくさん
 - デバイス → サーバーにHTTPでポーリング
 - SORACOM Beam経由でも実現可能です。
 - デバイス <-> サーバーでMQTTセッションでやりとり
 - こちらもSORACOM Beam経由でも実現可能です。
 - サーバーからデバイスへPush
 - SORACOM Gateを使えばデバイスのIPアドレスへサーバーから通信が可能。
 - IMSIからデバイスのIPアドレスを検索することも可能です。
 - Lightweight M2Mによるやりとり
 - SORACOM InventoryはLightweight M2Mというデバイス管理プロトコルを採用したデバイス管理サービス。
 - 通信だけではなくて、各種設定値のスキーマ、FWアップデートのプロシージャまで定義されているのが特徴で、それらを実装済みのエージェントもオープンソースで見つかります。
 - SORACOM Air経由で払い出した鍵を使って、WiFiや優先経由でのやり取りも可能。

設定値やファームウェアの更新

- デバイスへの書き込みにはやり方がたくさん

- デバイス → サーバーにHTTPでポーリング

- SORACOM Beam経由でも実現可能です。

- デバイス <-> サーバーでMQTTセッションでやりとり

- こちらもSORACOM Beam経由でも実現可能です。

- サーバーからデバイスへPush

- SORACOM Gateを使うとデバイスのIPアドレスへサーバーから通信が可能。

- IMSIからデバイスのIPアドレスを検索することも可能です。

- Lightweight M2Mによるやりとり

- SORACOM InventoryはLightweight M2Mというデバイス管理プロトコルを採用したデバイス管理サービス。

- 通信だけではなくて、各種設定値のスキーマ、FWアップデートのプロシージャまで定義されているのが特徴で、それらを実装済みのエージェントもオープンソースで見つかります。

- SORACOM Air経由で払い出した鍵を使って、WiFiや優先経由でのやり取りも可能。

で、どれを選べばいいの？

設定値やファームウェアの更新



SORACOM
Technology Camp 2018

• デバイスへの書き込みにはやり方がたくさん

• デバイス → サーバーにHTTPでポーリング

- SORACOM Beam経由でも実現可能です。

• デバイス <-> サーバーにMQTTでメッセージングでやりとり

- こちらもSORACOM Beam経由でも実現可能です。

• サーバーからデバイスへPush

SORACOM Gateを使えばデバイスのIPアドレスへサーバーから通信が可能。

- IMSIからデバイスのIPアドレスを検索することも可能です。

• Lightweight M2Mによるやりとり

- SORACOM InventoryはLightweight M2Mというデバイス管理プロトコルを採用したデバイス管理サービス。
- 通信だけではなくて、各種設定値のスキーマ、FWアップデートのプロシージャまで定義されているのが特徴で、それらを実装済みのエージェントもオープンソースで見つかります。
- SORACOM Air経由で払い出した鍵を使って、WiFiや優先経由でのやり取りも可能。

で、どれを選べばいいの？

デバイス次第

設定値やファームウェアの更新

まとめるとこんな感じです。

方式	SORACOMの提供する価値	コメント
HTTPポーリング	SORACOM Beamによるセキュアで、（デバイスにとって）負荷の低いやりとりが実装可能に。	- アプリケーションも含めてエージェントをゼロから実装するなら選択肢に入る。 - メッセージのスキーマやポーリングインターバル、メッセージを受け取った際の処理など、いろいろスクラッチで作る必要がある。が、言い換えれば自由にやれる。
MQTT	SORACOM Beamによるセキュアで、（デバイスにとって）負荷の低いやりとりが実装可能に。	少し古いアプリケーションだと、この方式でプロトコルが定義されていることが多いので、その際に採用すればよい。
サーバーからのPush （プロトコル問わず）	- SORACOM Gateによるサーバーから直接デバイスのIPアドレスへのアクセス - IMSIをベースにIPアドレスや、デバイスのオンライン状態を確認可能。	- アプリケーションとは別にデバイス管理エージェントとして採用することができる。 - 市販のデバイスの中には、LwM2M対応のデバイスもある。
LwM2M	SORACOM InventoryはLwM2Mサーバーをホストしているサービスなので、エージェント側だけを用意することでLwM2Mが利用可能。	

ワークフローを考えてみる

異常をふせぐために

1. 用途以外の利用を制約する
2. 用途以外の利用を検知する

異常をふせぐために

- デバイスが故障していたら・・・
 - 通信量やセッション履歴の変化から問題を発見できる。
(前述)
- デバイスからSIMが抜き取られても・・・
 - (任意の) 通信ができなければOK
- デバイスごと盗まれる/乗っ取られても・・・
 - (任意の) 通信ができなければOK
 - 通信ができてしまう場合でも、可能な限り早く発見をする
仕組みによって問題を最小化する

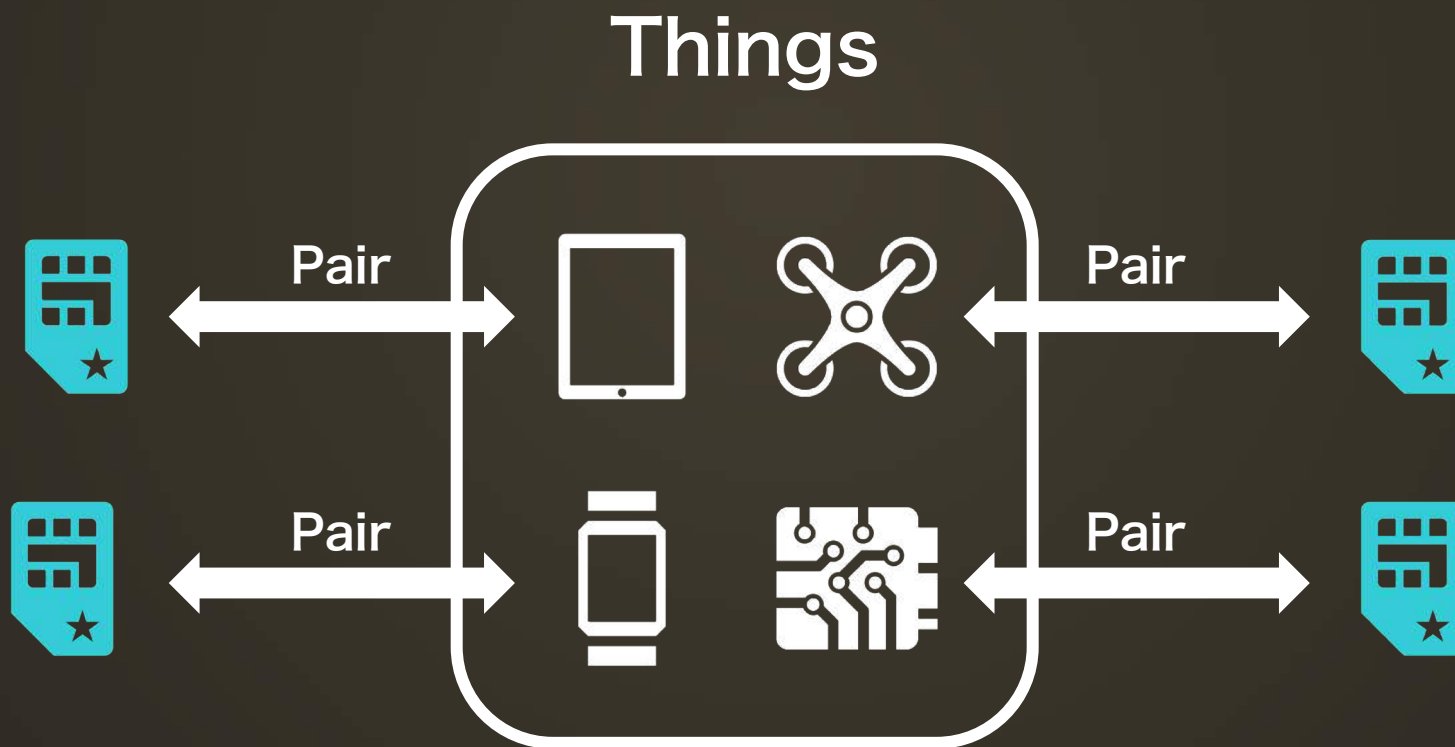
ワークフローを考えてみる

異常をふせぐために

1. 用途以外の利用を制約する
2. 用途以外の利用を検知する

用途以外の利用を制約する

SIMとIMEI(モデムのユニークなID)が一致していない
と通信できないようにする「IMEIロック」



用途以外の利用を制約する

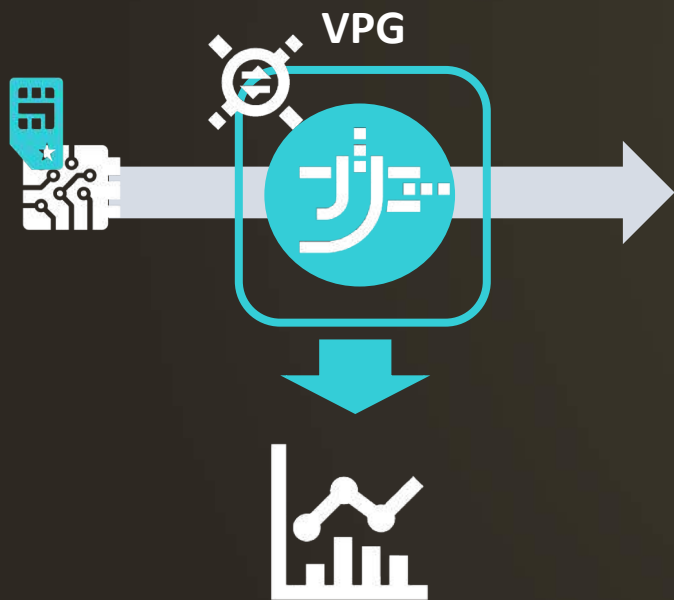
SORACOM Canal/Direct/DoorはInternet Gatewayを閉じることにも可能。結果、意図した社内システムとの閉域接続しかできない、という構成ができる。



用途以外の利用を制約する

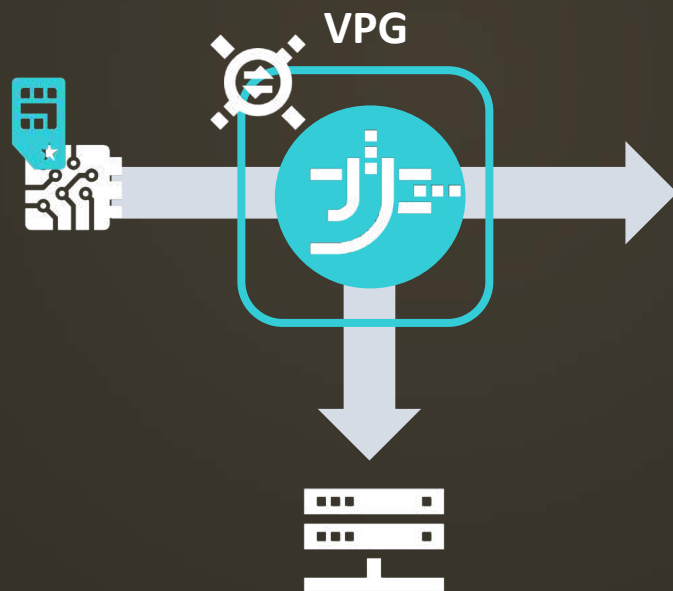
SORACOM Junction(Redirection)を使えばインラインでのパケットの監査やフィルタリングが可能に。

Inspection



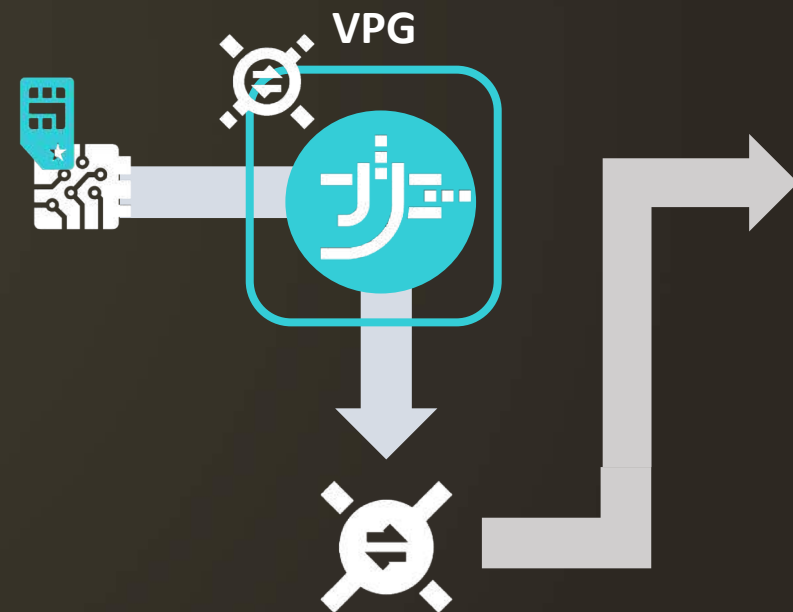
パケットフローを解析して
統計情報を出力

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

ワークフローを考えてみる

異常をふせぐために

1. 用途以外の利用を制約する
2. 用途以外の利用を検知する

用途以外の利用を検知する

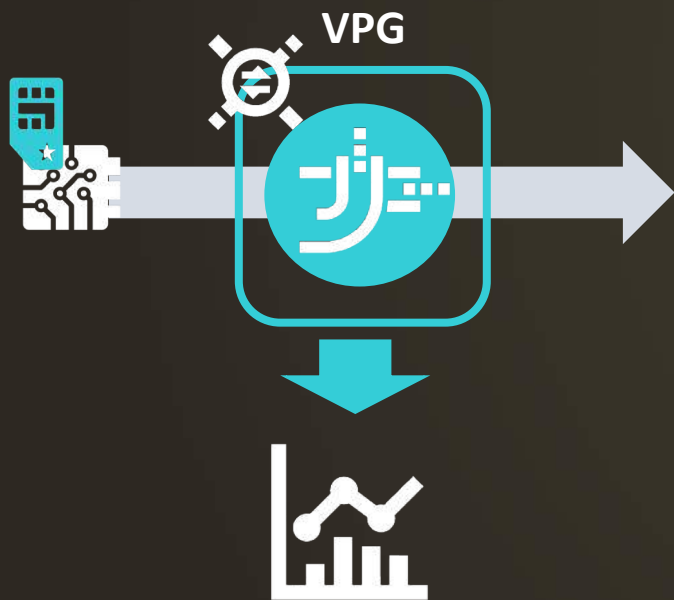
前述のイベントハンドラを使えば・・・

- 1日に100MB以上の通信を行ったSIM(デバイス)のIMSIをメールで通知 & 速度制限
- 1日に100MB以上の通信を行ったSIM(デバイス)をAWS Lambdaに通知、LambdaからSORACOM APIを呼び出して、SIMを利用休止状態に変更、その上で自社のシステムに通知。

用途以外の利用を検知する

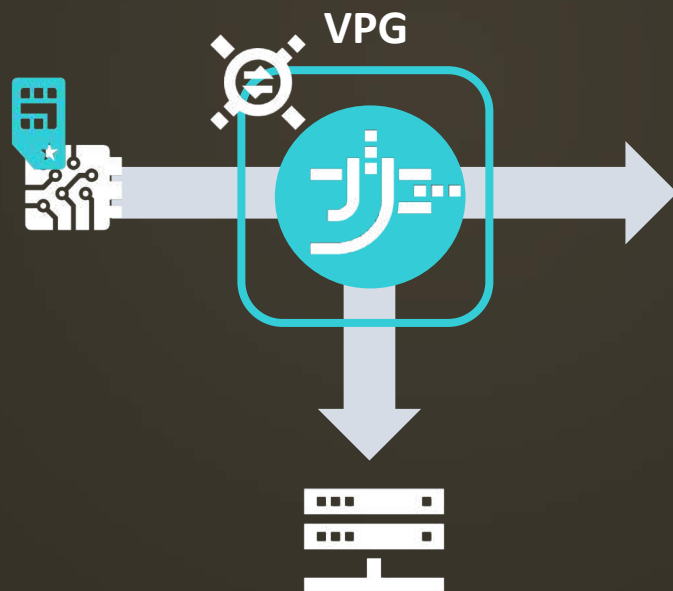
SORACOM Junction(Inspection, Mirroring)を使えばパケットを監査可能。

Inspection



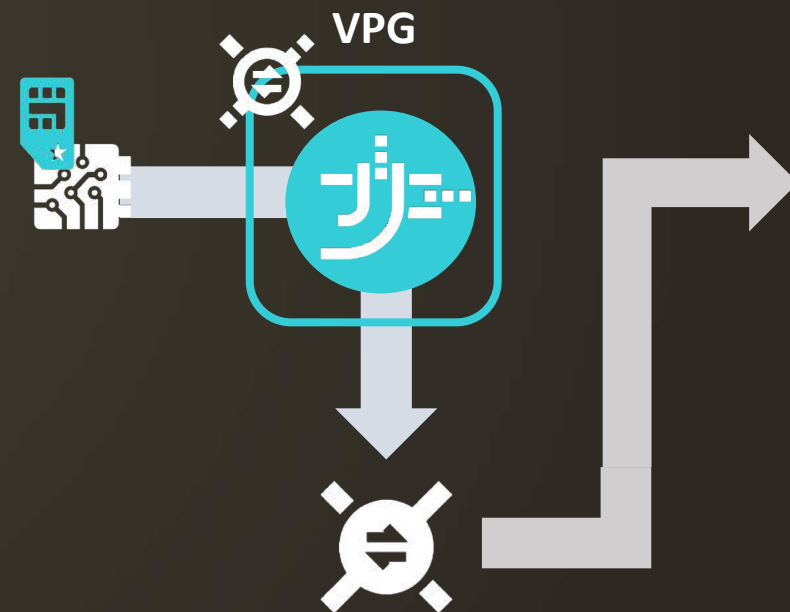
パケットフローを解析して
統計情報を出力

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サービスでは「SIM」を鍵/
証明書としてデバイスを識別することが
でき、それを活かすための便利なサービ
スがたくさん。

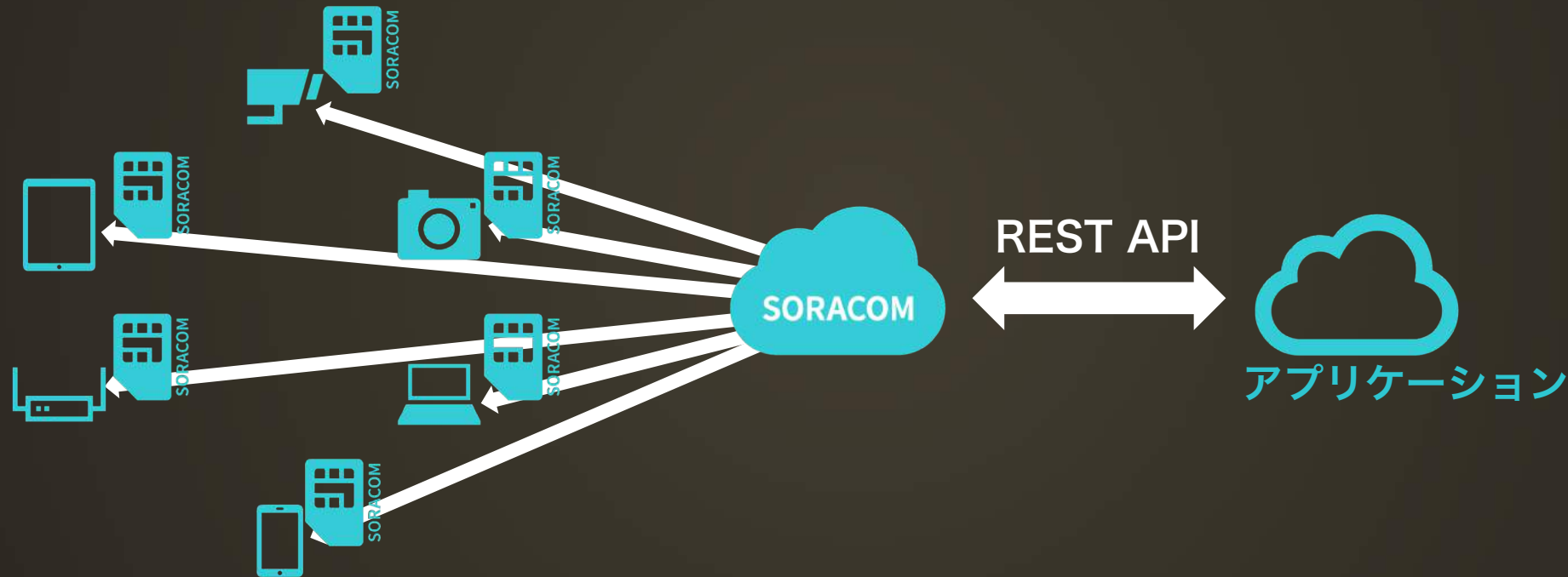
Topic3:

まとめと結論

SORACOMはただの通信サービスではなく、通信やデバイスを管理することのできる通信プラットフォームです！

ポイントは「SIM」を鍵/証明書として
デバイスを識別すること。

ポイントは「SIM」を鍵/証明書として
デバイスを識別すること。



SIM経由でSORACOMのさまざまなAPIにより、
デバイスの管理が可能になります！

すべてをSORACOMの機能に頼る必要
はありません。

アプリケーションの本分でないところを
SORACOMにオフロードしてください。

ご注意ください

アプリケーションから切り離して、**純粋な回線サービス**として比較検討してしまうと、あまりSORACOMの旨味がでないことが多いです。

アプリケーション設計の段階から
SORACOMの機能をひとつおり検討頂
いて、うまく利用できるところをピック
アップしていただくのがベストです。

まとめ

- SORACOMにはさまざまなサービスがあり、データ転送だけでなく、デバイス管理にも大きなメリットが。
- SORACOMのSIMをデバイスのIDとして利用することで、REST API経由でこれらの機能をフルに活用可能です。
- すべての機能を使い倒すのではなく、アプリケーションの本分でないところをSORACOMにオフロードを。
- アプリケーションから切り離して回線サービスとして見るとうまく活用しきれないことが多いのでご注意ください。

SORACOM User Group 本日開催



- 時間 18:00 – 20:00
- 参加費 500円

多数のライトニングトークや特別企画をご紹介します！



Try! SORACOM チャレンジキャンペーン

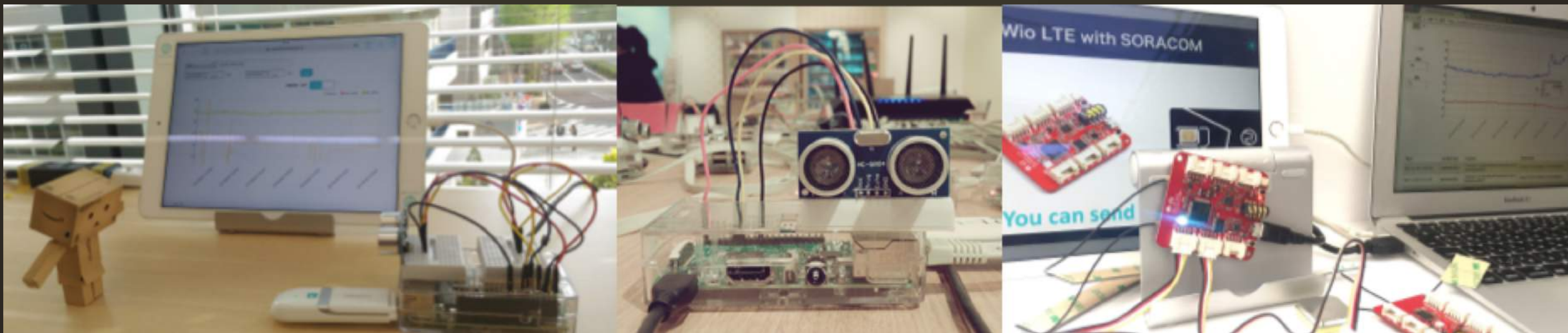


～記事を書いてノベルティをゲットしよう～

期間：4月26日～6月30日

対象：企業・個人問わずどなたでも参加可能

内容：SORACOMを使った電子工作や、Deep Diveな記事を書いた方にクールなノベルティをプレゼントします



今日から IoT を始めよう！ 体験キット本日販売中



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

(税抜価格：15,980円)



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

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