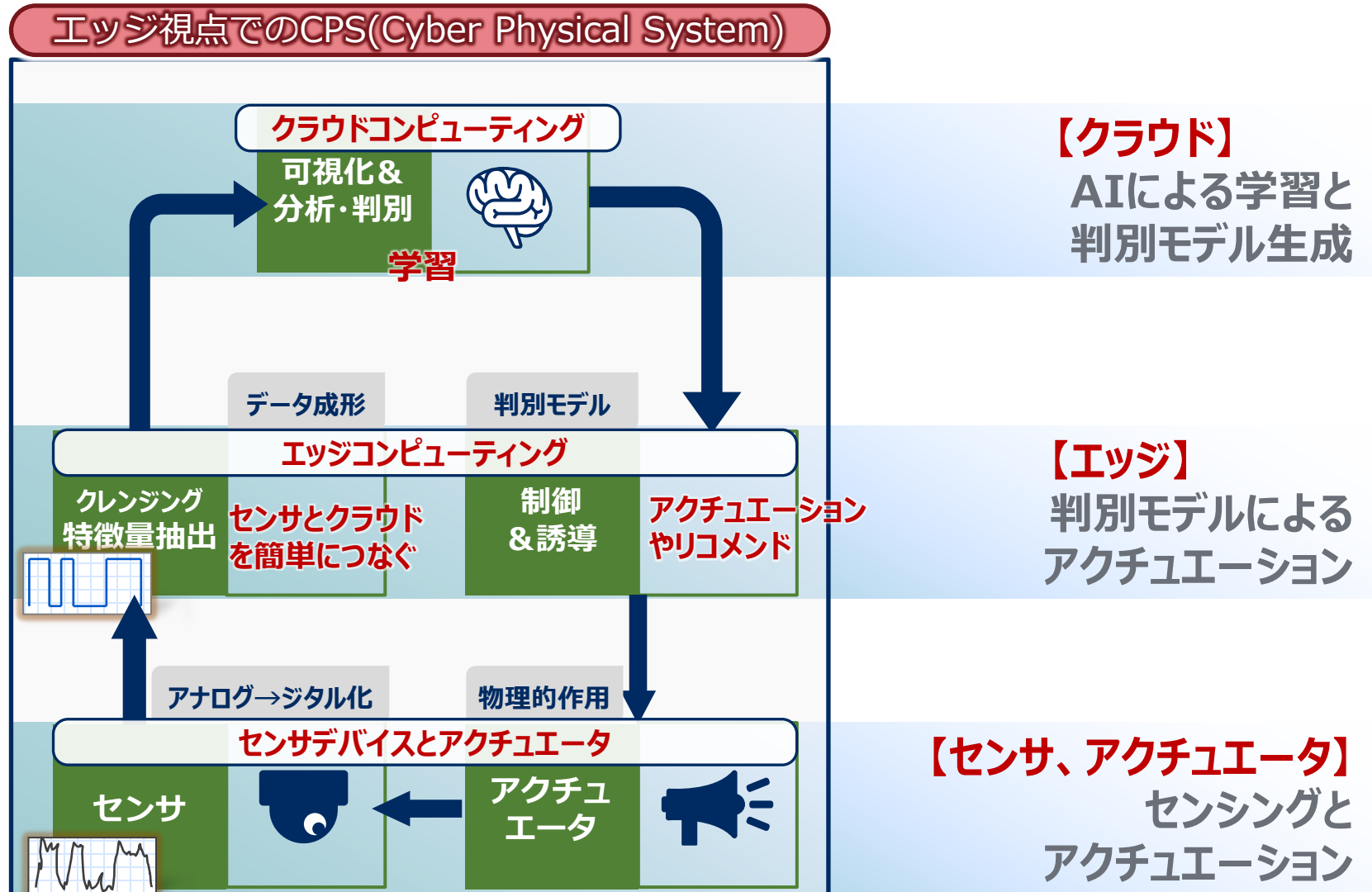




My-IoT開発プラットフォーム概要 (2021年7月版)

エッジコンピューティングの役割

デジタル化された実世界の事象をクラウドを活用して理解し、実世界へ作用させる



My-IoT開発プラットフォームのコンセプト

①市場課題認識

IoTは**一部の業種領域適用**に留まり、顧客の業務の中にまだ根付いていない



IoTプラットフォーム
(IoTシステム提供者)

画一性



相反する課題
「IoTギャップ」

メーカお仕着せで
価値検証の前に
高いコスト発生

多様性



エンドユーザ・利用者(IoTを業務活用したい)

②IoTギャップ

・提供側（プラットフォーム）

・顧客毎に要望が異なり、**有償のカスタマイズが必須**

・利用者側

・メーカお仕着せのため、自分に合ったシステム構築のためには、**高度なITリテラシーが必要**

・インフラ課題

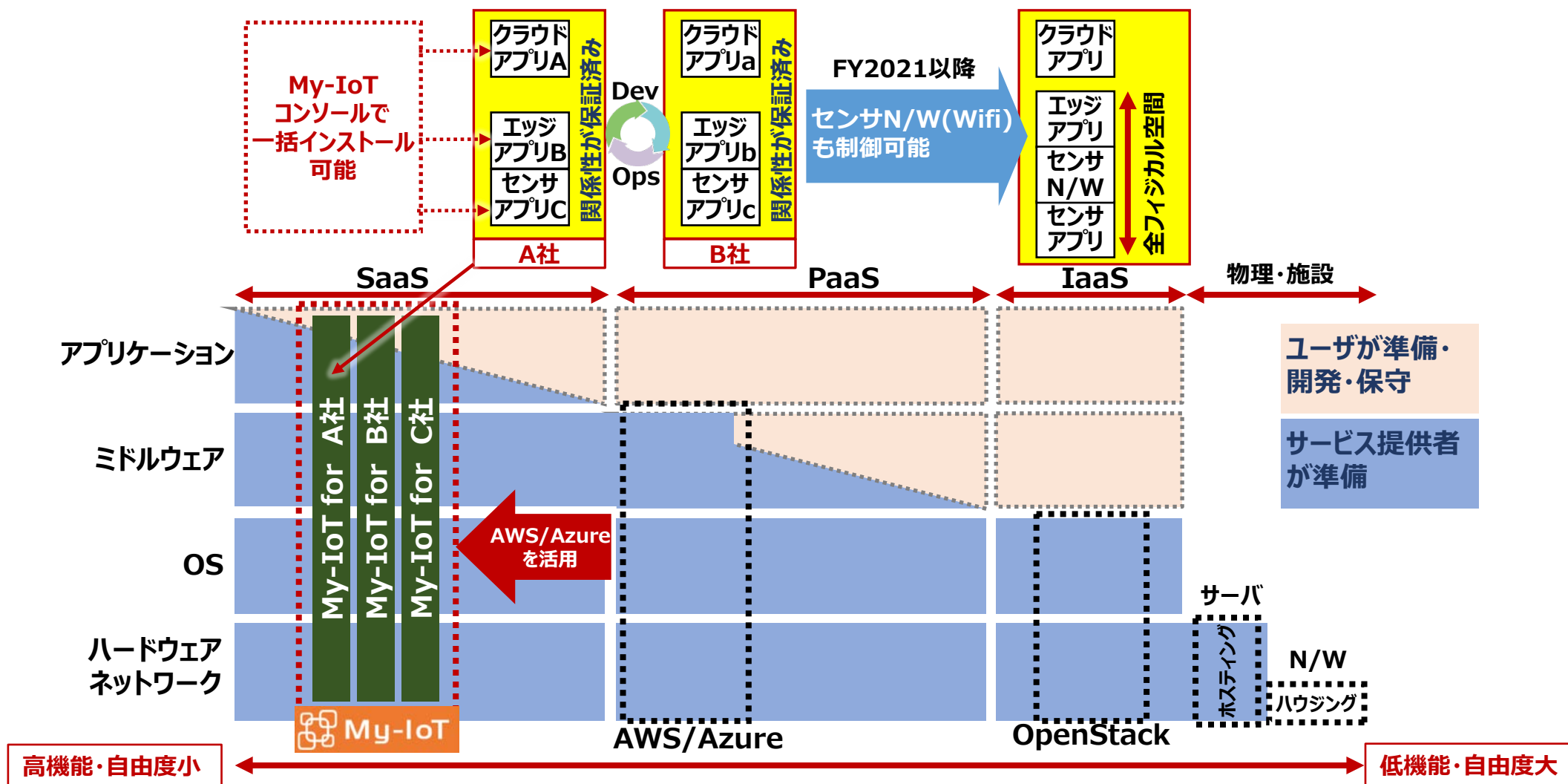
・データをAIで利活用しやすい形で収集するのは難しい。
クラウドスキルやセンサー仕様の理解も障壁

エッジを活用し、IoTの構築を容易化(工数1/10以下)できる技術開発

自分でIoTを構築し、価値検証し、自身に最適なIoTを構築 = **My-IoT**

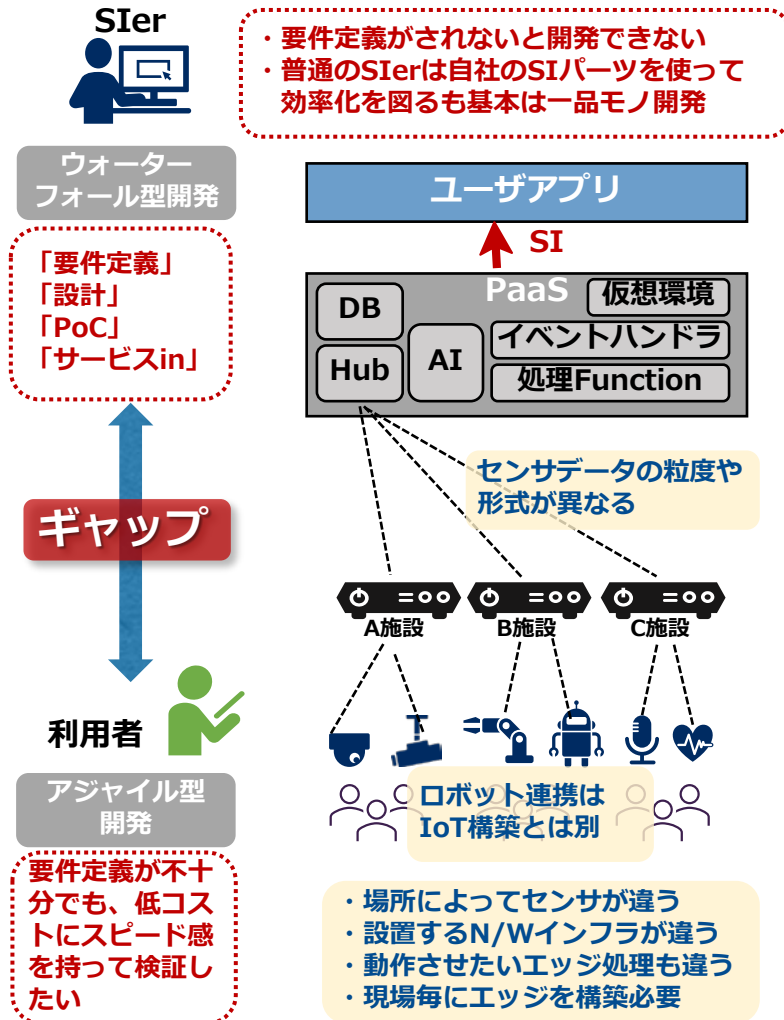
My-IoT開発プラットフォームの対象セグメント

- ① AWS上で動作するため、最新のPaaS機能を利用したうえで、ユーザ視点の機能をアプリケーションとして提供
- ② エッジとクラウドのアプリケーションをユーザが自由に組み合わせて「システムパッケージ」としてインストール可能



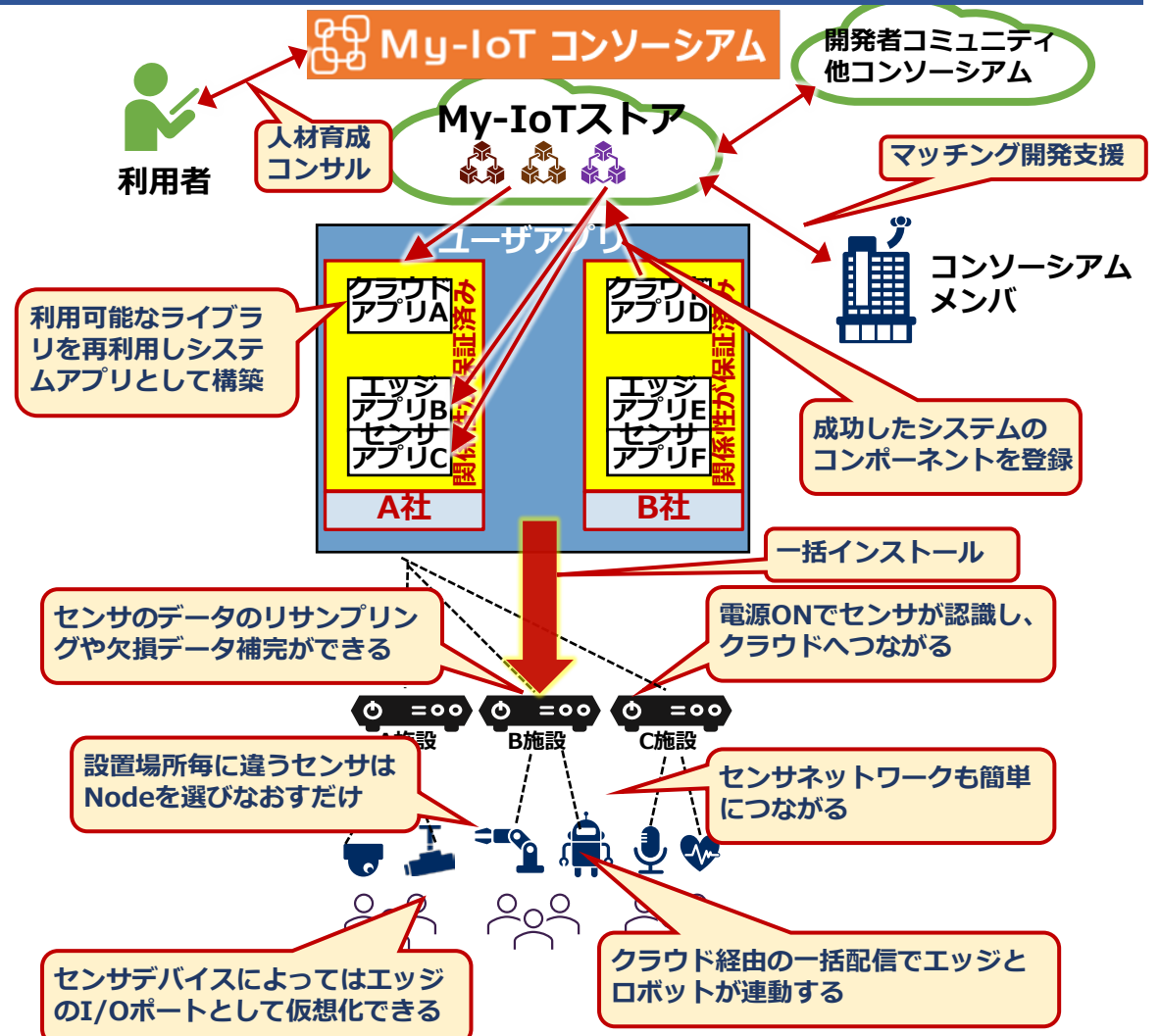
My-IoT開発プラットフォームのメリット

【従来】外部発注型のIoT構築

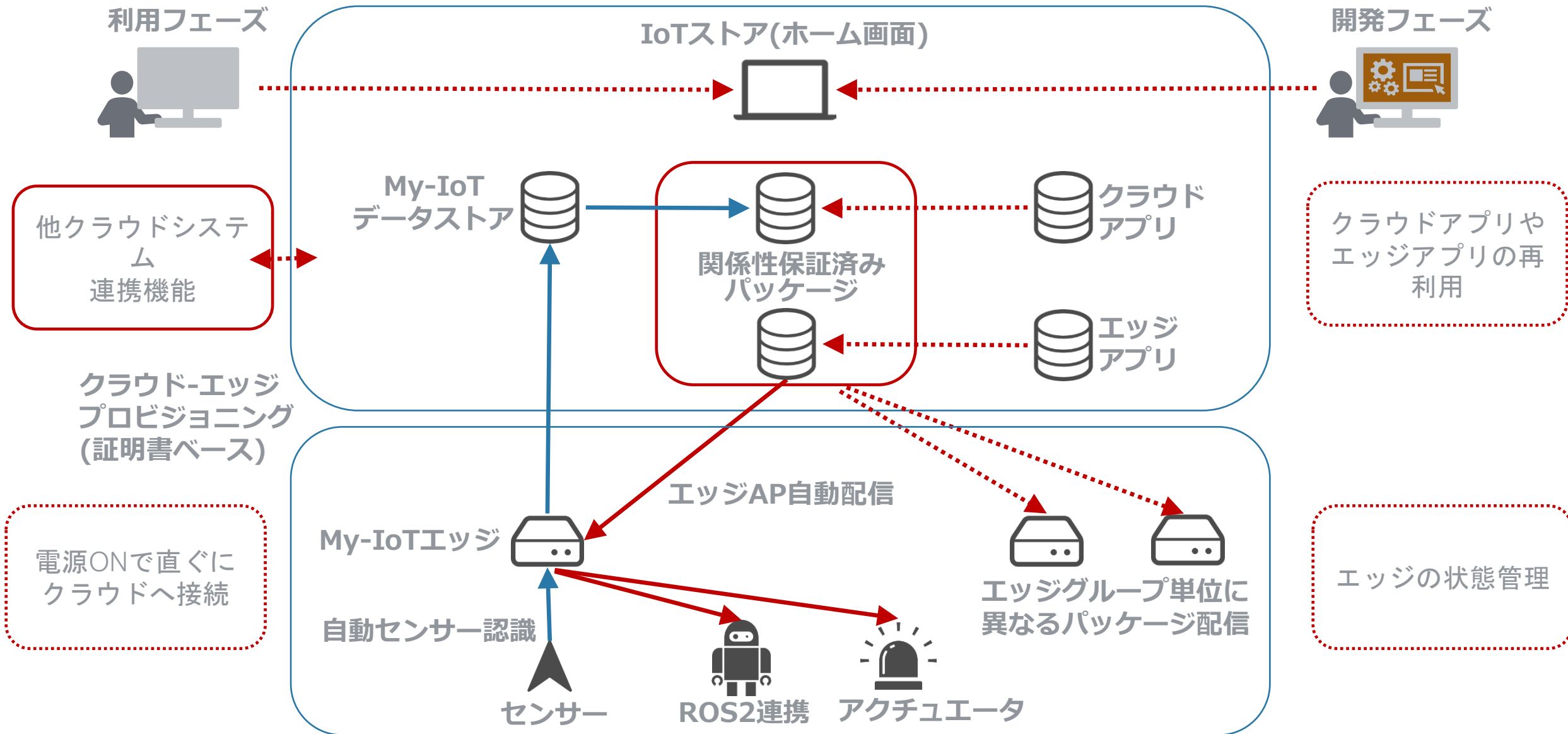


少しでも
解消したい

My-IoTが実現する(*一部は開発中)



My-IoT開発PF(プラットフォーム)の全体像



My-IoT開発PFができること

IoTストア画面にログインすることで、すべての機能を管理・利用できます。

管理

エッジの状態やグループの状態などの監視が可能

パッケージ

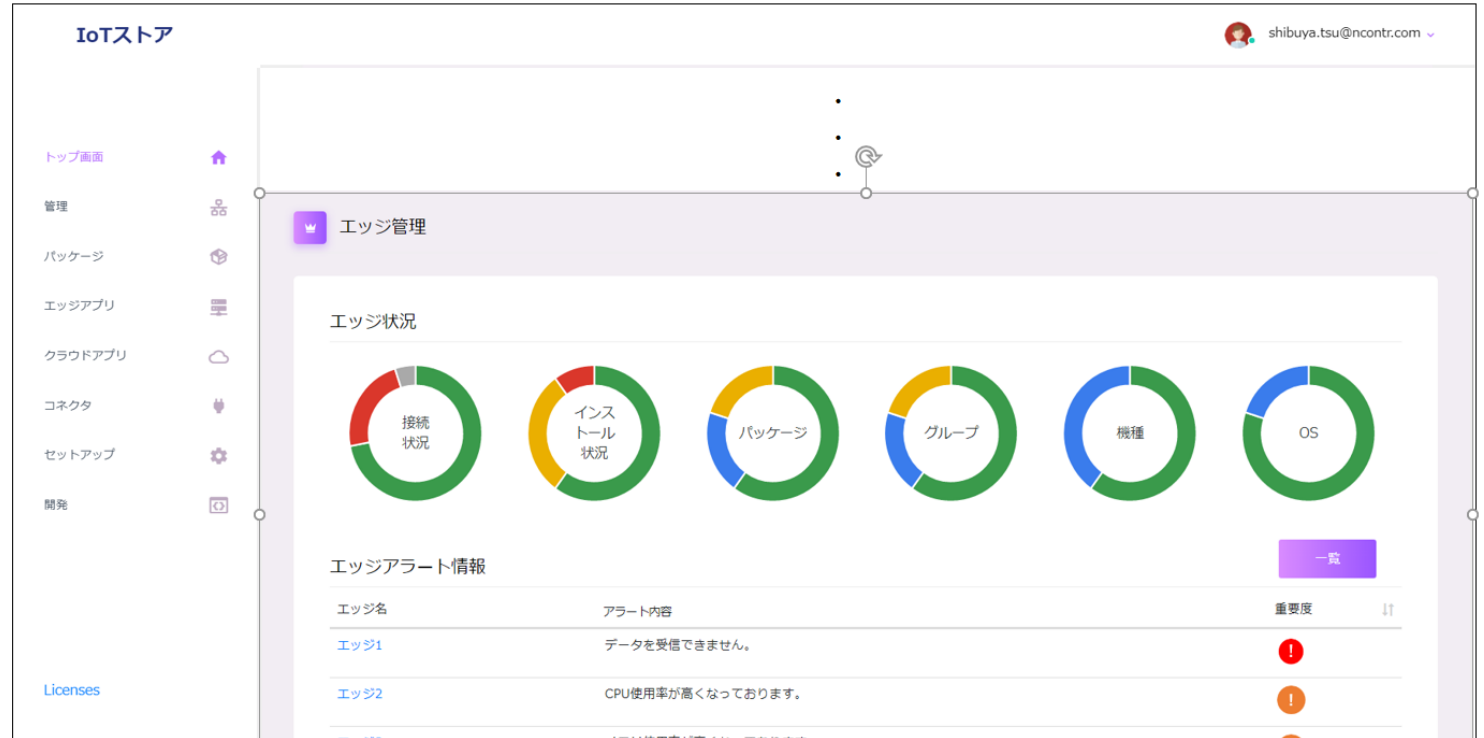
クラウドAPとエッジAPの関係性を保証したうえで、全階層へ一括インストールします

エッジアプリ

Node-REDフローとスクリプトとともにPythonやROS2と連動して複雑なエッジ処理が可能です

クラウドアプリ

My-IoTのバックエンドのAWSで動作しLambdaとAPI Gatewayが利用し、センサデータが格納されているMy-IoTデータをアクセスできます



エッジAP開発

My-IoTの接続されたエッジをエッジ環境とすることで、センサなどを繋げながらエッジアプリケーションの開発が可能です

My-IoT開発PFが対応しているエッジ

現在、標準で対応しているエッジは以下のラズパイシリーズになります。

今後はエッジを開発する企業様と連携し、My-IoT開発PFに対応あるいは推奨するエッジ・ハードウェアを増やして行く方針です。

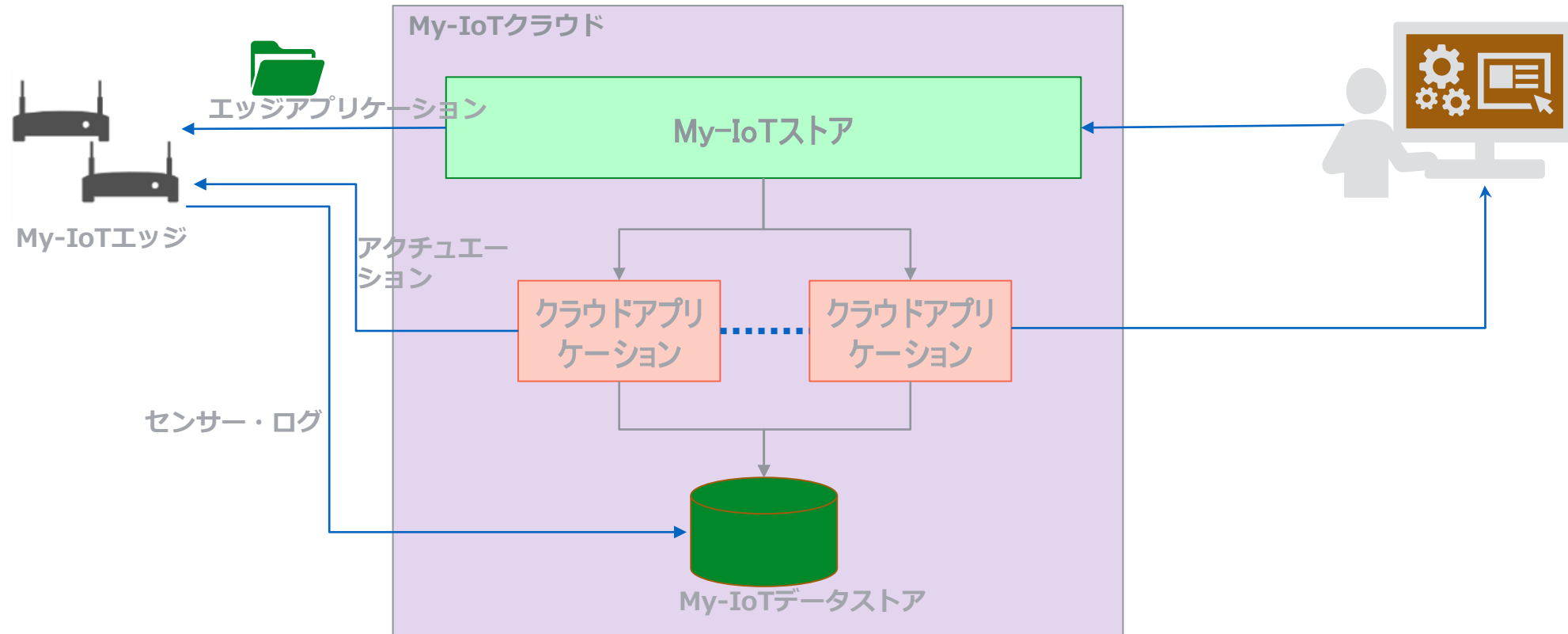
デバイス	OS
Raspberry Pi 3 Model B	Ubuntu 20.04(64bit) Raspberry Pi OS xx(32bit)
Raspberry Pi 3 Model B+	Ubuntu 20.04(64bit) Raspberry Pi OS xx(32bit)
Raspberry Pi 4 Model B	Ubuntu 20.04(64bit) Raspberry Pi OS xx(32bit)

ソフトウェア構成

My-IoTの基本的なソフトウェア構成は、

「My-IoTストア」「My-IoTデータストア」「エッジアプリケーション」「クラウドアプリケーション」です。

- My-IoTストアから配信したエッジアプリケーションでデータをMy-IoTデータストアに送信して蓄積し、同じくクラウドアプリケーションでMy-IoTデータストアに蓄積されたデータを利用して解析や各種処理、エッジやセンサに対するアクチュエーションを行います。



IoTストア、My-IoTデータストア

IoTストアは、My-IoTを利用するためのWebアプリケーションです。

ブラウザからIoTストアにアクセスすることで、利用者はMy-IoTエッジの管理やIoTシステムの構築を行うことができます。開発者は、IoTストアに公開されるアプリの登録や管理を行うことができます。

My-IoTデータストアとは、My-IoTクラウド上に存在する契約者ごとのデータベースです。

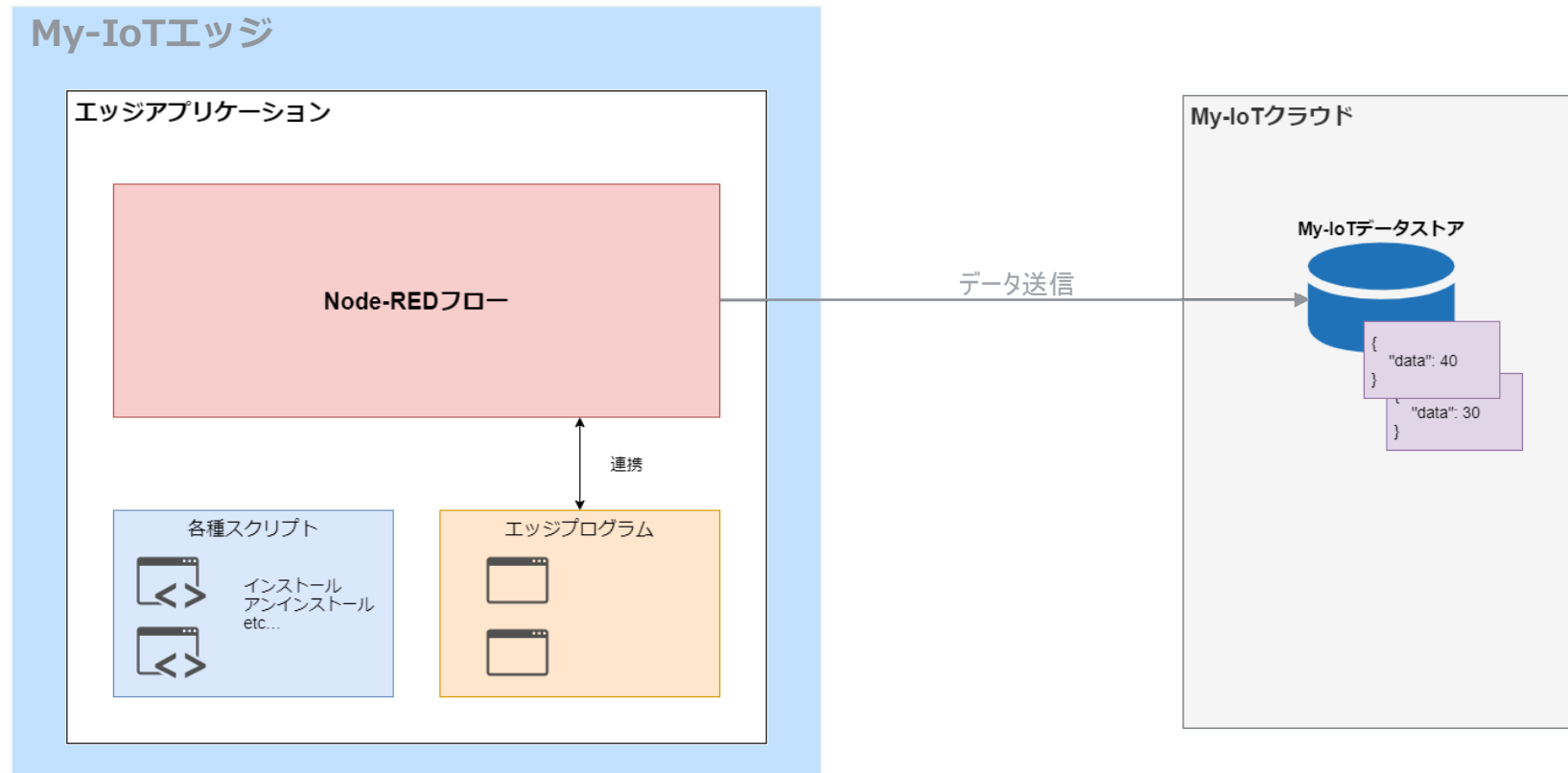
エッジアプリケーションやクラウドアプリケーション、APIを通じてデータを蓄積、利用することができます。

IoTストア サイドメニュー



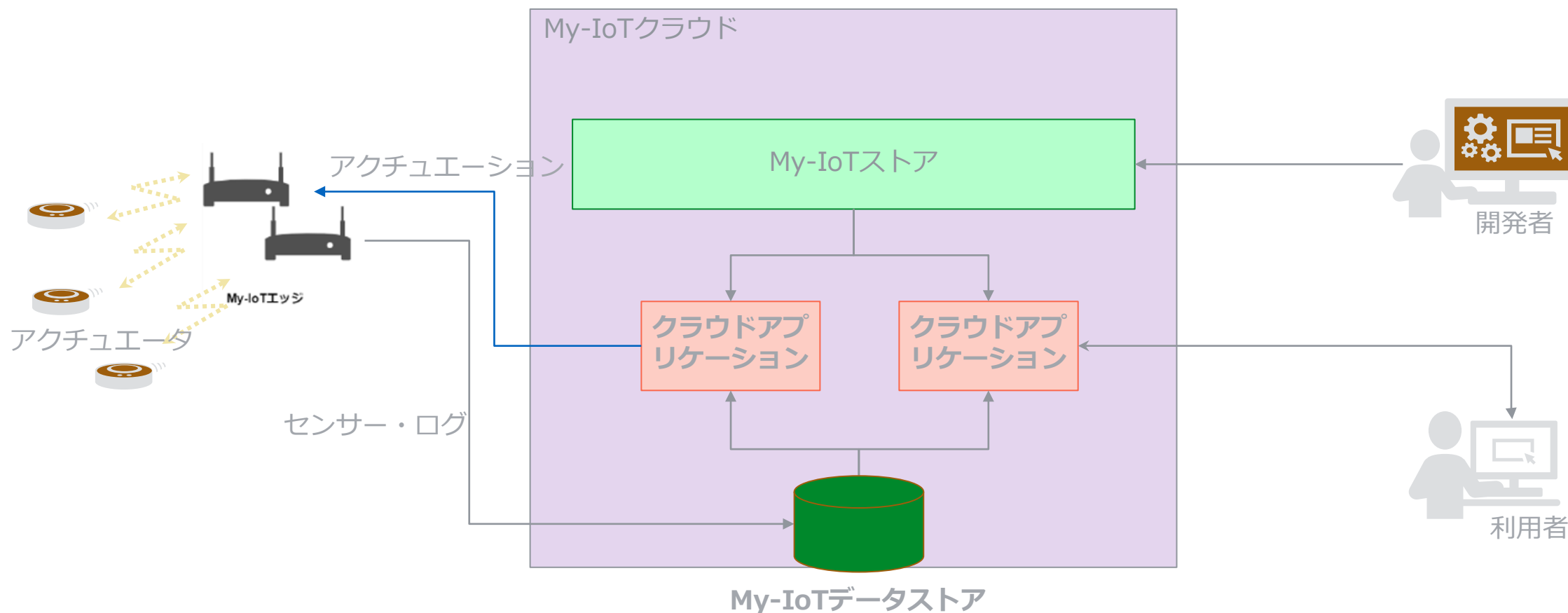
エッジアプリケーション

エッジアプリケーションとは、My-IoTエッジ上にインストールされ動作するアプリケーションのことです。
エッジアプリケーションは、Node-RED(OSS)で構築されたNode-REDフローと呼ばれるアプリケーションと、
その他スクリプトなどのコンポーネントから構成されています。



クラウドアプリケーション

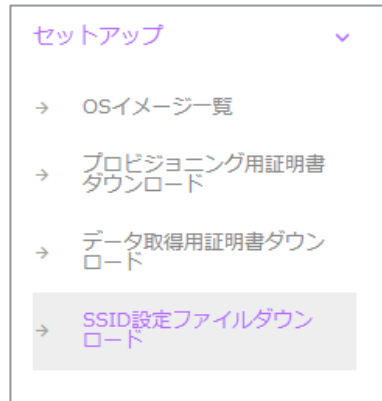
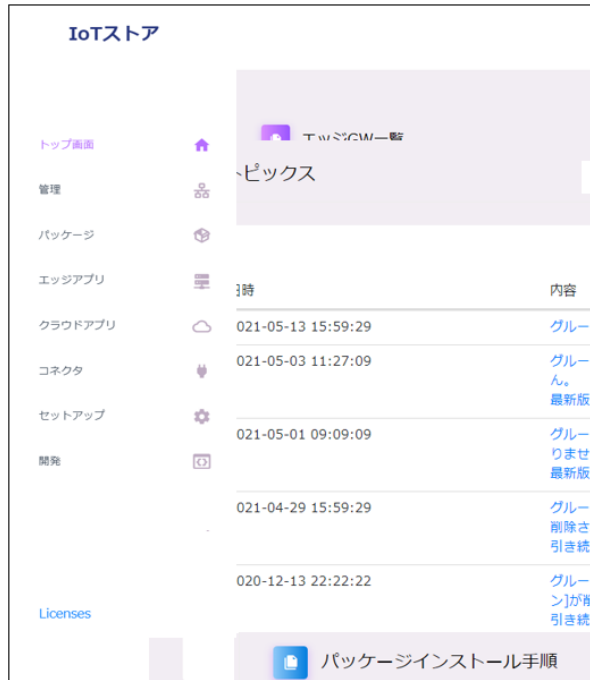
クラウドアプリケーションはエッジアプリケーションからMy-IoTデータストアに送信、蓄積されたデータを解析や各種処理し結果の表示を行います。またMy-IoTエッジやアクチュエーターに対する指示も可能です。クラウドアプリケーションは開発者が開発、登録を行い、利用者が目的に応じてMy-IoTストアから選択して使用します。



初期作業(ラズパイの場合)

利用者は初期作業として次の作業を行います。

- (1)IoTストアのWeb画面より「OSイメージ」「証明書」「SSID」をダウンロードしSDカードに書き込みます
- (2)SDカードをエッジデバイスに挿入し起動します。これにより自動でMy-IoTへの接続が開始されます
- (3)Web画面からエッジグループを作成し、接続したエッジを登録することでIoTシステムの構築準備が完了します。(複数のエッジをまとめた「グループ」という単位でエッジを管理します)



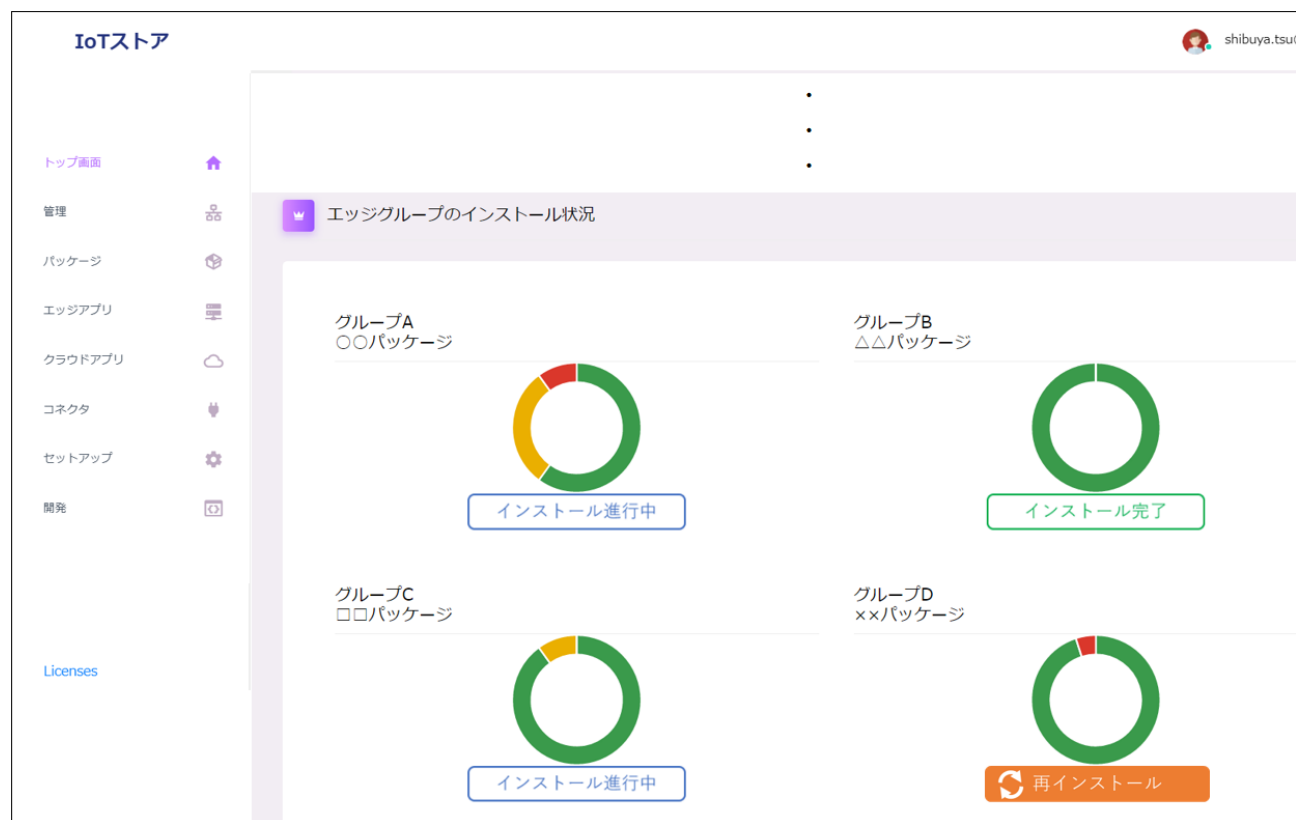
エッジグループ名	オンラインエッジ数	オフラインエッジ数	インストール済み数	設定したパッケージ	更新日時	管理
グループ1	12	2	0	-	2020-09-03 16:27:54	エッジ登録 パッケージ編集 削除
グループ2	4	23	4	フローA	2020-09-03 16:27:54	エッジ登録 パッケージ編集 削除
グループ3	12	3	12	フローB	2020-09-03 16:27:54	エッジ登録 パッケージ編集 削除
グループ4	40	4	40	フローC	2020-09-03 16:27:54	エッジ登録 パッケージ編集 削除

未所属エッジ数	パッケージ未インストールエッジ数	パッケージ削除中エッジ数	管理
20	10	10	詳細

My-IoTシステム構築

My-IoTには、さまざまなエッジとクラウドのアプリケーションが登録されています。利用者は目的に合わせて自由にアプリケーションを組み合わせ、システムパッケージを作成することができます。システムパッケージを作成して配信すると自動的にエッジとクラウド両方にアプリケーションのインストールが行われ、自分だけのIoTシステムが構築されます。構築に複雑な操作や手順は必要ありません。

IoTストア

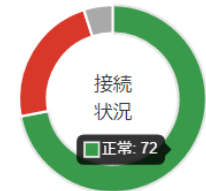


エッジの状態管理

Web画面のダッシュボードでは、さまざまな情報を一目で確認することができます。グループごとのエッジ接続状況やシステムパッケージのインストール状況、エッジで発生しているエラーやリソース枯渇などのアラート情報といった、多数のエッジを管理する上で欠かせない情報を素早く把握、確認できます。



グラフにマウスオーバさせると詳細が表示されます

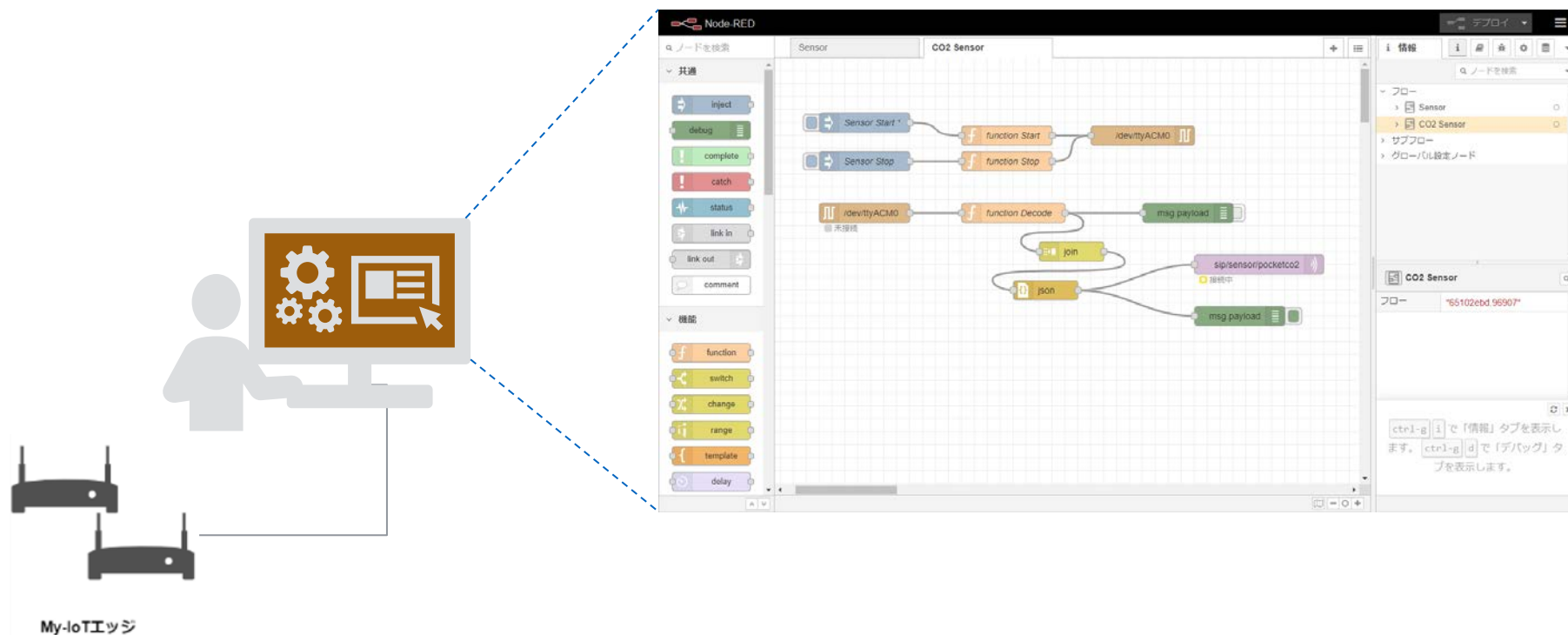


インストール進行中

エッジアプリケーションの開発

My-IoTに接続されているエッジをWeb画面からメンテナンス機に設定することで、エッジアプリケーションを開発可能です。開発者は、エッジにディスプレイやキーボードなどを接続するだけで、ブラウザを使ってエッジアプリケーションを開発します。

エッジアプリケーションはローコード(Node-RED)開発できます。また、PythonやCMake可能なプログラムでも開発可能です。

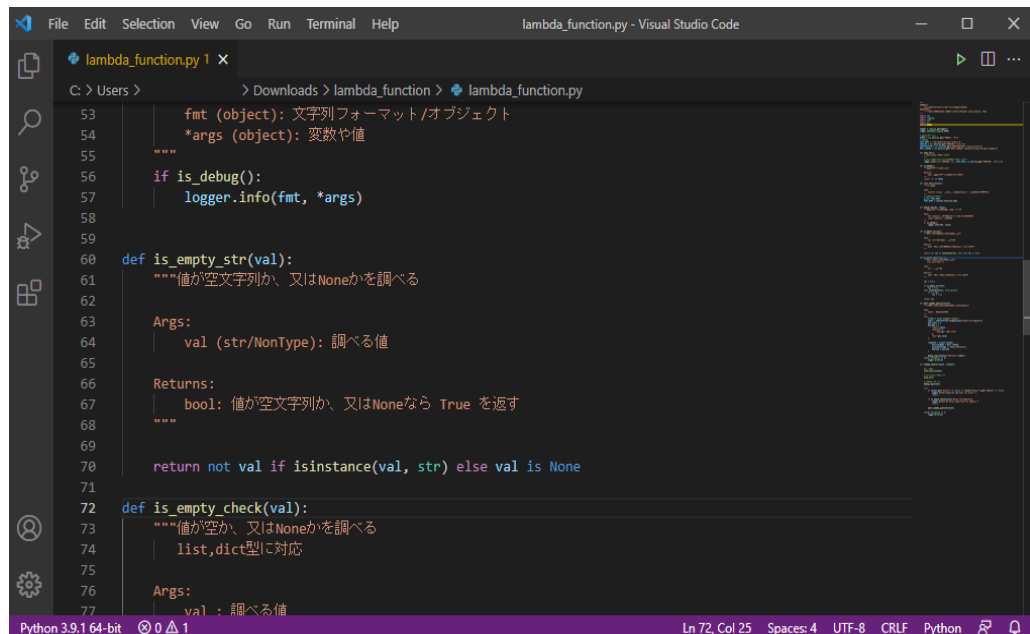


クラウドアプリケーションの開発

My-IoTを用いることでエッジデバイスからクラウドへ送信されたデータを目的に応じて処理するクラウドアプリケーションを簡単に作成・導入することができます。

例えばエッジデバイスから送信されたデータをもとに、特定の条件でクラウド側からアラートを発生させるクラウドアプリケーションを作るとします。開発者はMy-IoTに蓄積されている実際のMy-IoTデータストアを参照しながら、クラウドアプリケーションで行う処理(受信データ処理の条件分岐・通知作成など)を手元で作成したPythonプログラムを実際に動かし検証しながら実装します。

クラウドアプリケーションの作成

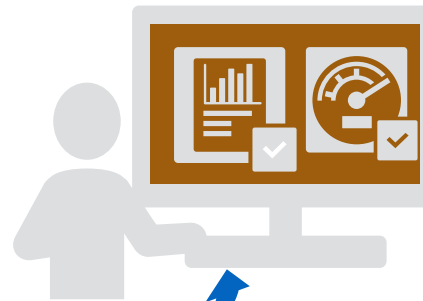


```
File Edit Selection View Go Run Terminal Help
lambda_function.py - Visual Studio Code

lambda_function.py 1 x
C:\Users\> Downloads\lambda_function> lambda_function.py

53     fmt (object): 文字列フォーマット/オブジェクト
54     *args (object): 変数や値
55     ....
56     if is_debug():
57         logger.info(fmt, *args)
58
59
60 def is_empty_str(val):
61     """値が空文字列か、又はNoneかを調べる
62
63     Args:
64         val (str/NonType): 調べる値
65
66     Returns:
67         bool: 値が空文字列か、又はNoneなら True を返す
68     """
69
70     return not val if isinstance(val, str) else val is None
71
72 def is_empty_check(val):
73     """値が空か、又はNoneかを調べる
74         list,dict型に対応
75
76     Args:
77         val: 調べる値
```

クラウドアプリケーションの実行・検証



クラウドアプリケーションの登録



セキュリティ

(1) サインアップ/ログイン

My-IoTコンソーシアムに入会するとIoTストアへのアクセスとサインアップが可能になります。IoTストアへアクセスするとログイン画面が表示されサインアップ時に登録したメールアドレスと、パスワードを入力してログインします

(2) エッジデバイスのセキュリティ

My-IoTではAWSで用意されている証明書を利用したプロビジョニングの仕組みを利用し、登録済みのMy-IoTエッジであることの認証を行い、My-IoTエッジ-クラウド間のセキュアな通信が可能となります

ログイン画面

Sign in with your username and password

Username

Password

[Forgot your password?](#)

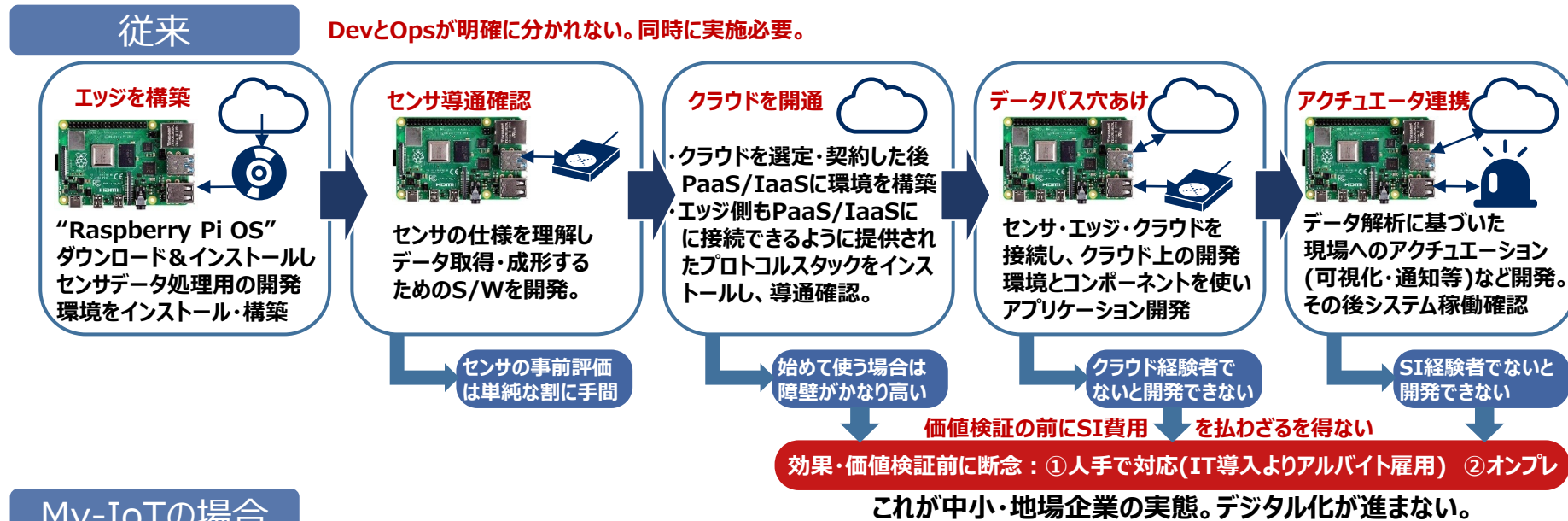
Sign in

Need an account? [Sign up](#)

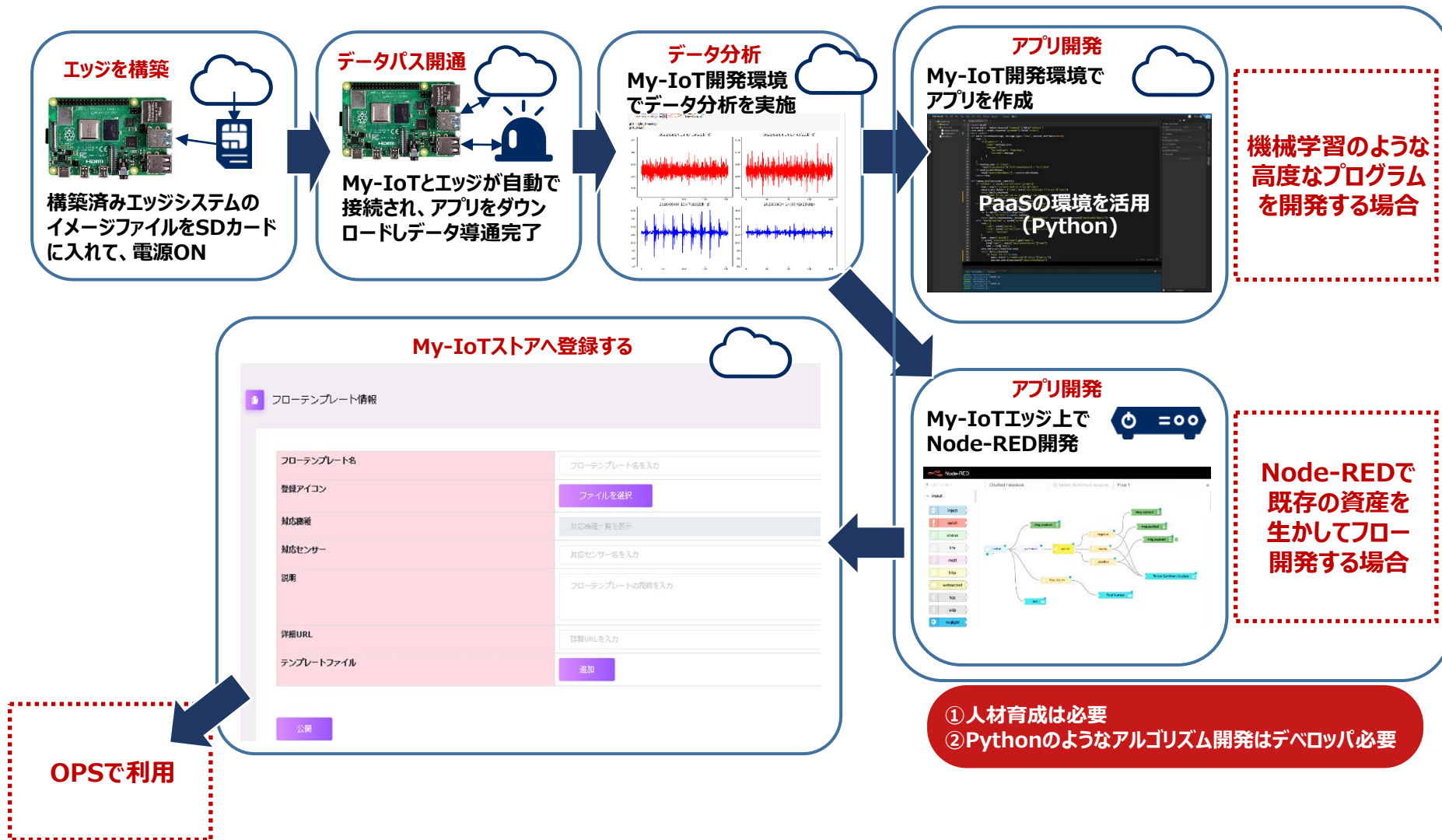
AWSの証明書を利用したエッジ認証



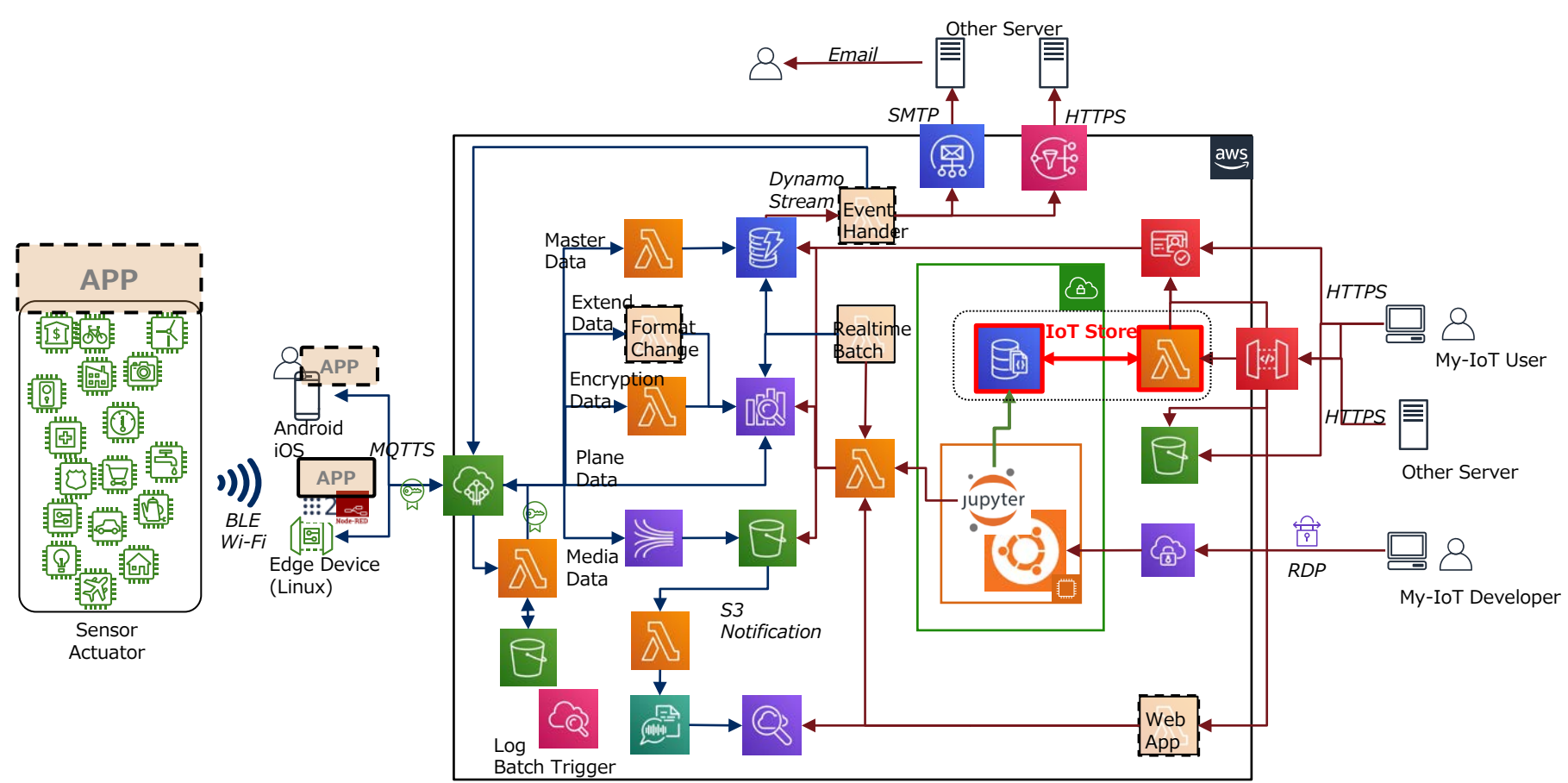
補足)My-IoT開発プラットフォーム活用例(利用シーン)



補足)My-IoT開発プラットフォーム活用例(開発シーン)



補足)My-IoT開発プラットフォーム(on AWS)アーキテクチャ



※赤枠箇所より、薄オレンジ部へのモジュール配信が可能、点枠2箇所については今後実装



My-IoT

あなたのためのIoT