TOSHIBA

デジタルサービス・プラットフォーム技術（DPF）研究会

CPSを支えるIoT基盤サービス HABANEROTS（ハバネロッツ）

株式会社 東芝　研究開発センター
コンピュータ＆ネットワークシステムラボラトリー
川添博史
2023年11月24日

プロフィール

- **～2019年　研究開発センター**
  - IoT共通基盤技術（「クラウドハブ」というコードネーム）の研究開発に携わる

- **2019～2022年　デジタルイノベーションテクノロジーセンター**
  - クラウドハブをベースとし、
    事業部門のCPS開発・運用のための共通基盤クラウドサービス
    「HABANEROTS（ハバネロッツ）」の開発に携わる

- **2022～　研究開発センター**
  - HABANEROTS 開発支援技術の研究開発 ほか

本日の流れ

HABANEROTSとは？

構成要素

① IoTデータ管理サービス
② IoTデータ管理コンソール
③ ホスティングサービス

アーキテクチャ

適用事例

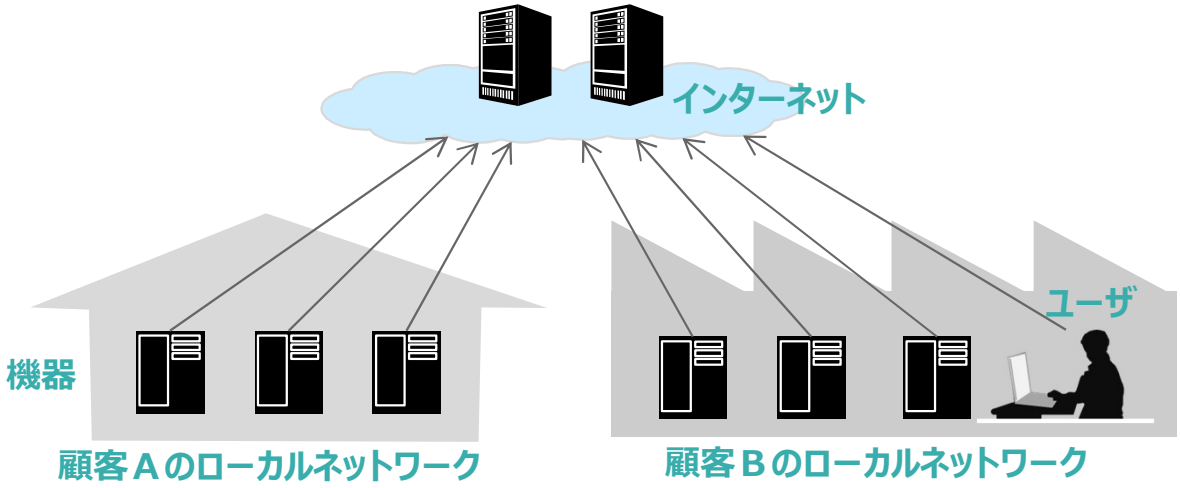
要素技術

・双方向通信API
・開発支援ツール ・・・ APIテスト自動生成

CPSサービスの本格運用における問題

【例： 産業向け機器のアフターサービス】
産業向け機器のサービスマン(ユーザ)が、顧客に納入した多数の産業機器の稼動ログや計測値をクラウド経由で監視し、必要に応じてメンテナンスのための制御を行う

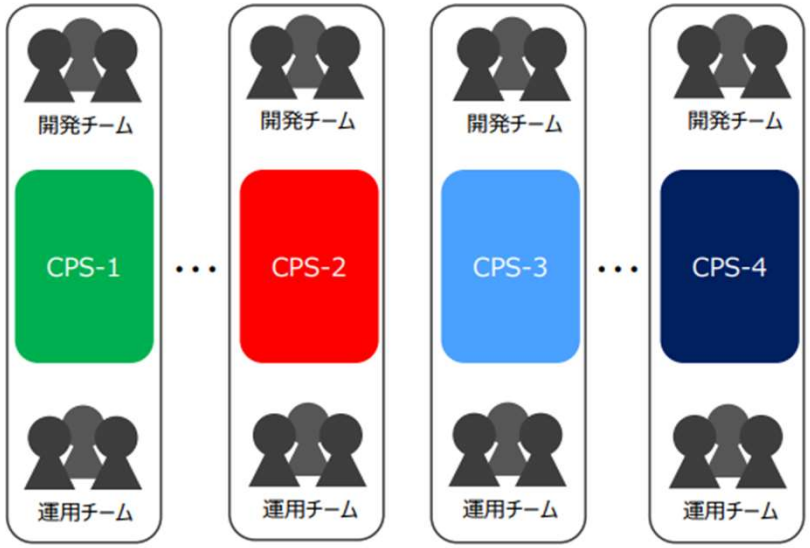
本格運用しようとすると問題が山積み



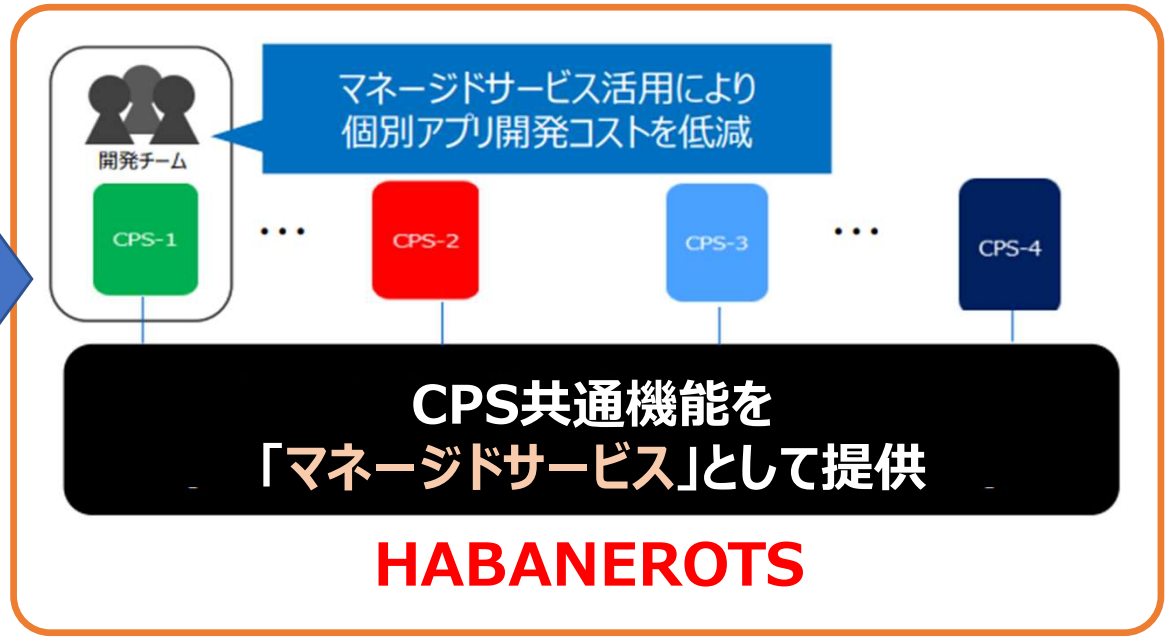
1. **遠隔監視・制御**
   - 顧客サイトのローカルネットワーク上の産業機器の可制御性を確保、リアルタイム性の高い監視・制御
2. **時系列データ収集・管理**
   - 計測データや稼動ログデータの蓄積、検索、統計処理、イベント通知等
3. **ユーザ・機器管理＆セキュリティ**
   - 認証とアクセス制御: どのユーザ(サービスマン、顧客)が、どの機器にどこまでアクセスできるのか
4. **ファイルデータ管理**
5. **可用性とスケーラビリティ**

HABANEROTS

CPSサービスの開発・運用コストを減らす共通基盤
各事業部門がCPSを始めるために必要な基本機能・環境を提供する



各事業部門が個別にCPSを開発・運用するのは人材・技術・ノウハウが分散し非効率

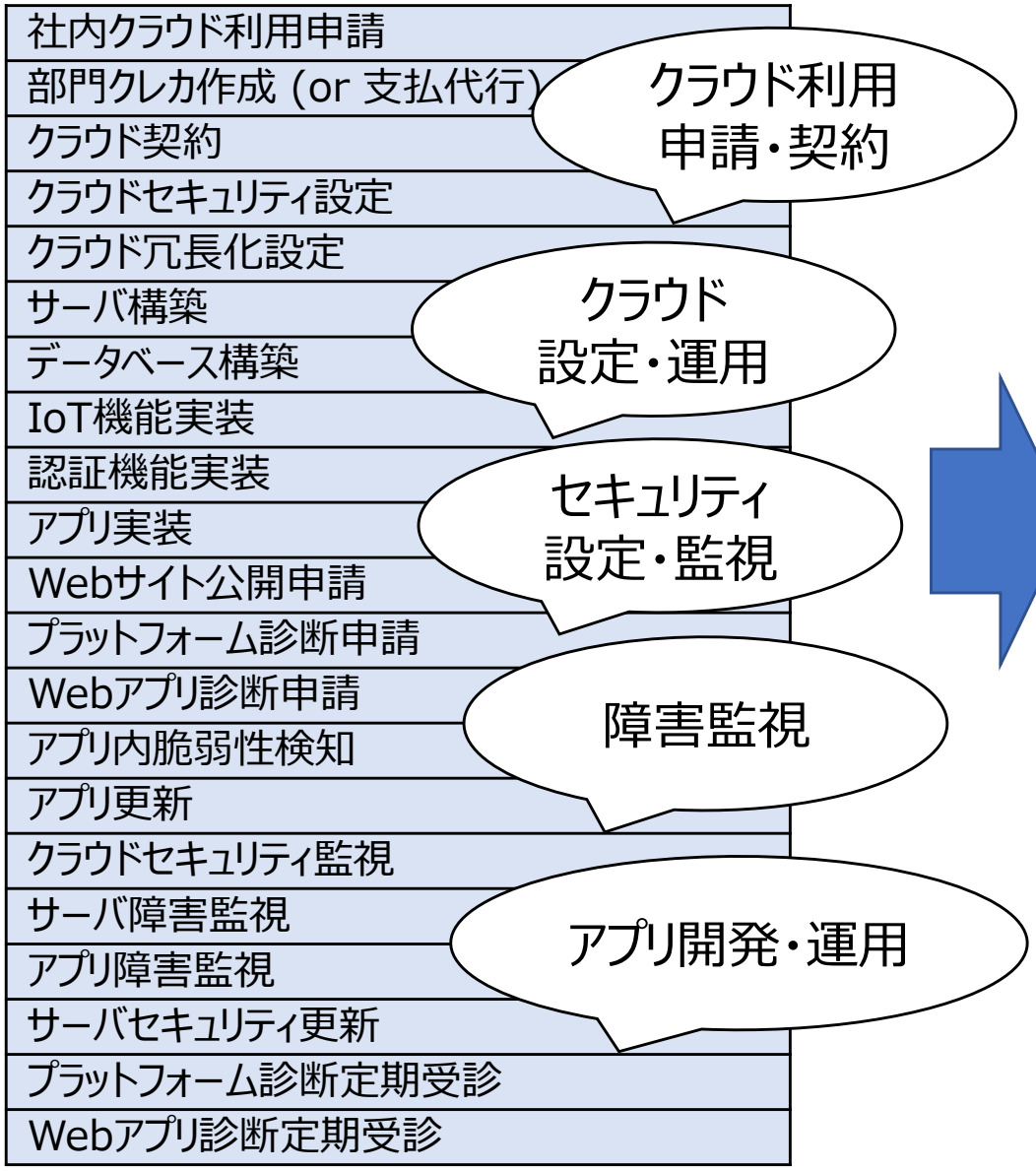


人材・技術・ノウハウの集約により
サービスの品質向上・競争力強化

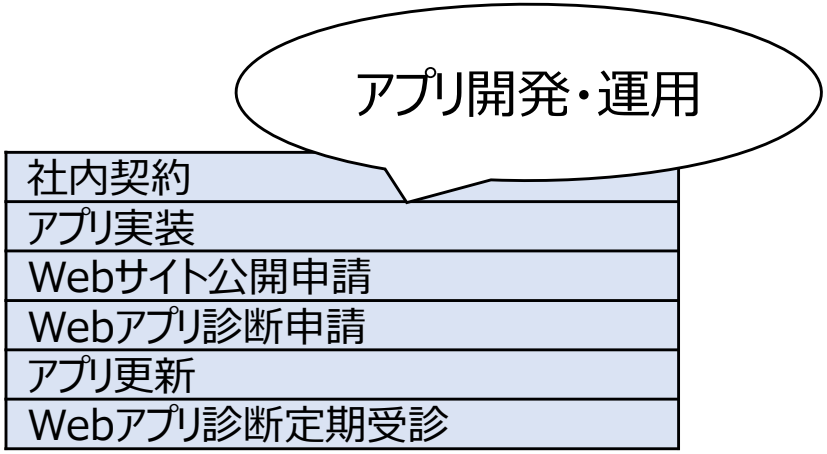
マネージドサービス ・・・ サービスの利用に必要な機器やソフトウェアの導入や管理、運用などの業務も一体的に請け負うサービスのこと。

HABANEROTSを使う利点

HABANEROTSを使わない場合



HABANEROTSを使う場合



HABANEROTSを使う利点

- クラウド利用申請やクラウド事業者との契約をせずともすぐサービス開始できる
- クラウド運用（構築、セキュリティ対応、障害対応）するクラウド専門家がいなくてもクラウドアプリを動かせる
- （できることが限定されているかわりに）必要なだけのコストでCPSサービス運用できる
- 社内のセキュリティ基準に沿ったサービス公開ができる
- 社内のWebデザイン基準に沿ったサインイン画面が使える
- IoT機器のデータ収集・蓄積が簡単にできる
- IoT機器の遠隔操作が簡単にできる
- データベースが使える
- アプリケーションの脆弱性チェックができる

HABANEROTSの構成要素

① IoTデータ管理サービス

- 個別のCPSに依存しない共通機能を提供する Web API サービス
- IoT機器の監視やデータ活用を行うためのプリミティブな機能を提供する

② IoTデータ管理コンソール

- デバイスやユーザー、アプリケーション、権限などの管理を行うための Web 画面

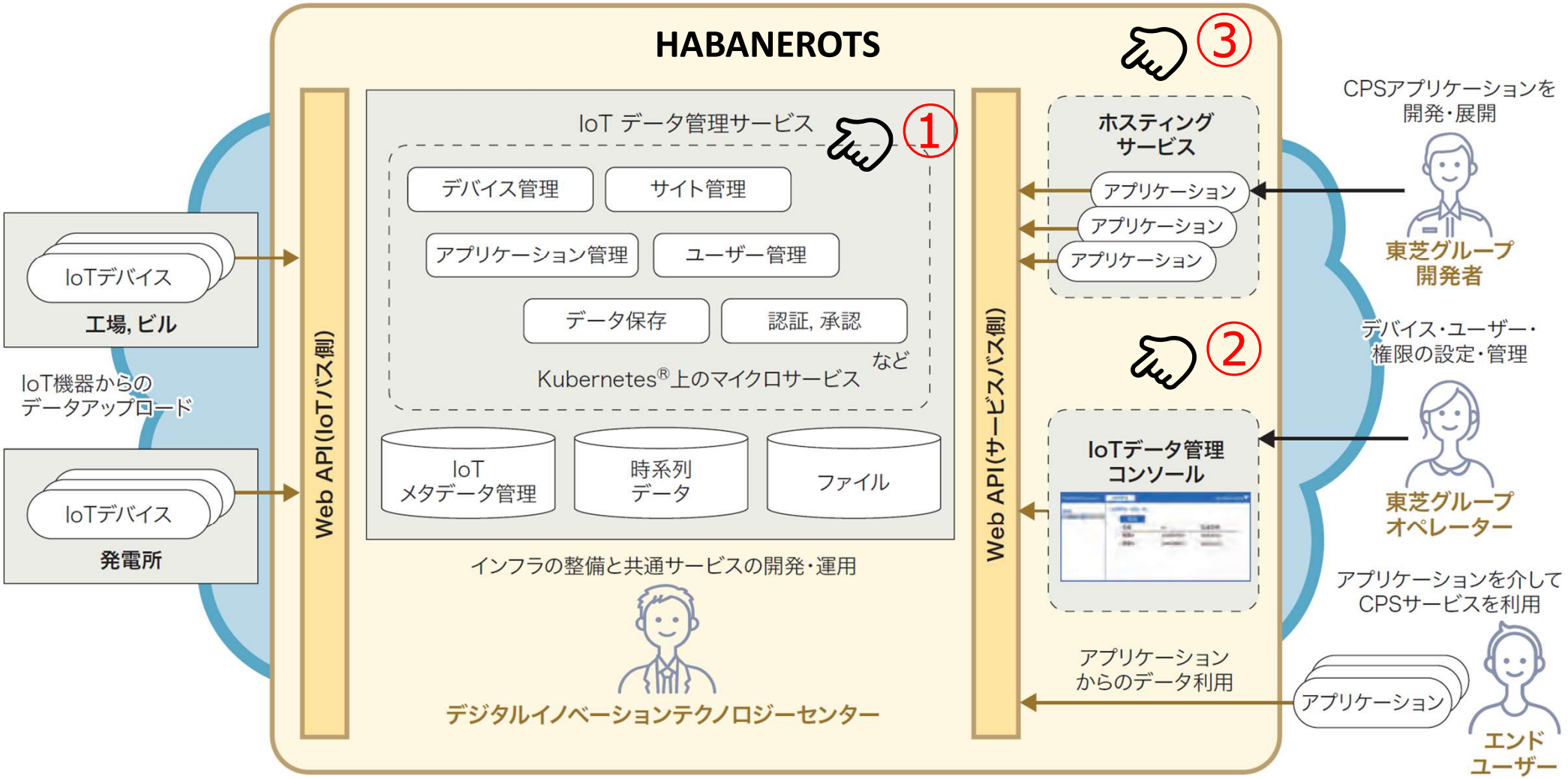
③ ホスティングサービス

- HABANEROTS のWeb APIを使って開発したCPSサービスを、HABANEROTS のクラウドサービス環境に展開・動作させるサービス

さらにソースコードは社内公開。事業部門が自前の環境を構築することも可能

HABANEROTSの構成要素

システム構成



①IoTデータ管理サービス

IoTデバイスの監視やデータ活用を行うために必要な基本機能をRESTful APIのセットとして提供

- **デバイス管理API**
  - デバイス認証・管理、グループ管理
- **ユーザ・権限管理API**
  - 認証主体（ユーザ・アプリ）の管理、ロール管理、認証・認可
- **時系列データ処理API**
  - デバイスから送信される時系列データの収集と利用、時系列データの構造管理
- **イベント購読・通知API**
  - 時系列データの閾値超え等の状態、長時間未送信（異常）状態のイベント購読・通知
- **ファイル管理API**
  - デバイス、アプリ、ユーザが登録する非構造データをファイルとして収集・管理
- **双方向通信API**
  - デバイスとアプリとの低遅延双方向通信

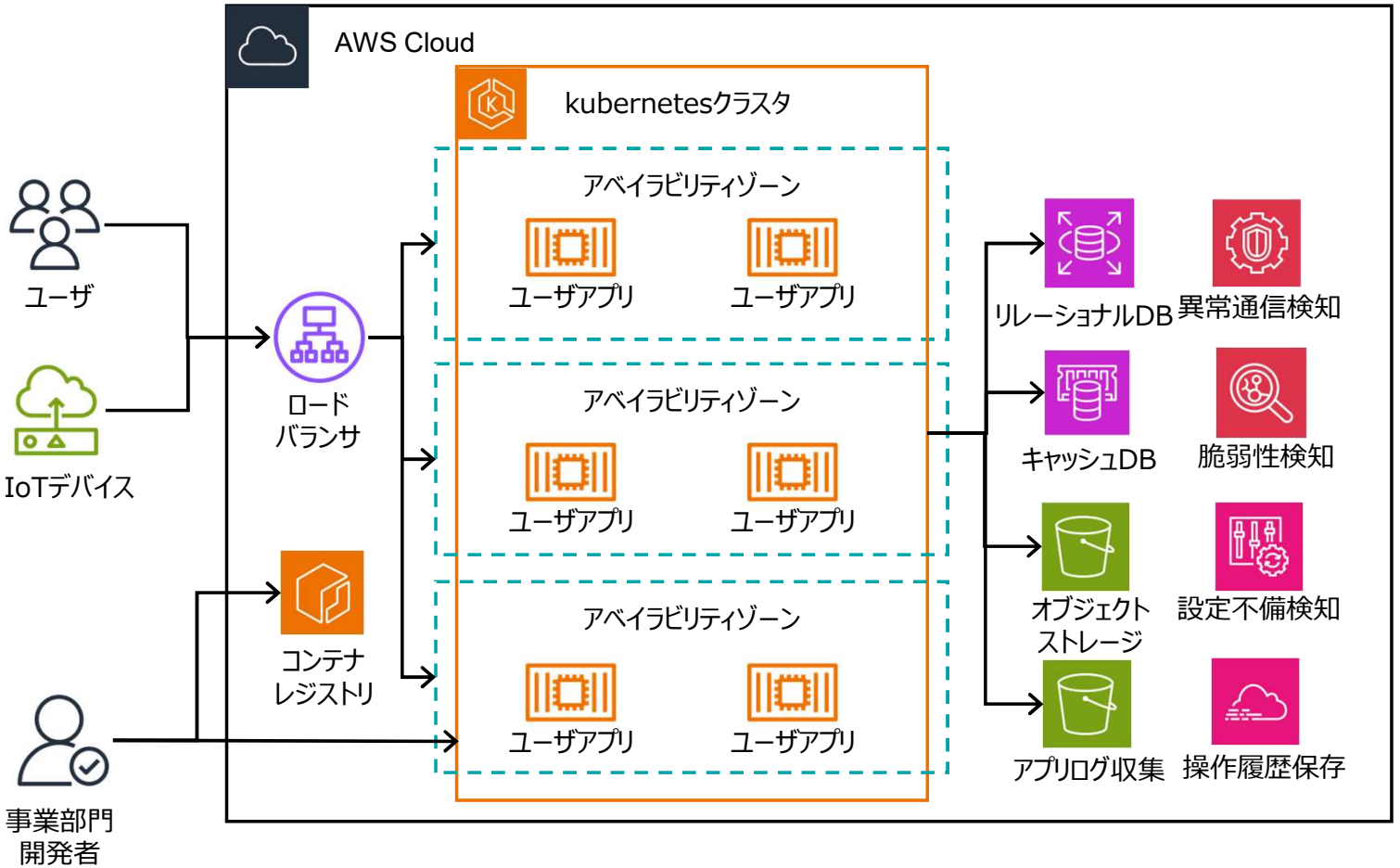
この後で紹介

- **遠隔制御API**
  - アプリ・ユーザからデバイス側へ発行するジョブ（遠隔制御命令）を管理
- **ファームウェア更新管理API**
  - デバイスのFWのセキュアなアップデート

③ホスティングサービス

事業部門自身がクラウド契約・運用することなくアプリを稼働できる環境

- 開発者はDocker形式のアプリを実装・デプロイすれば
  冗長性とセキュリティが確保された環境で稼働可能

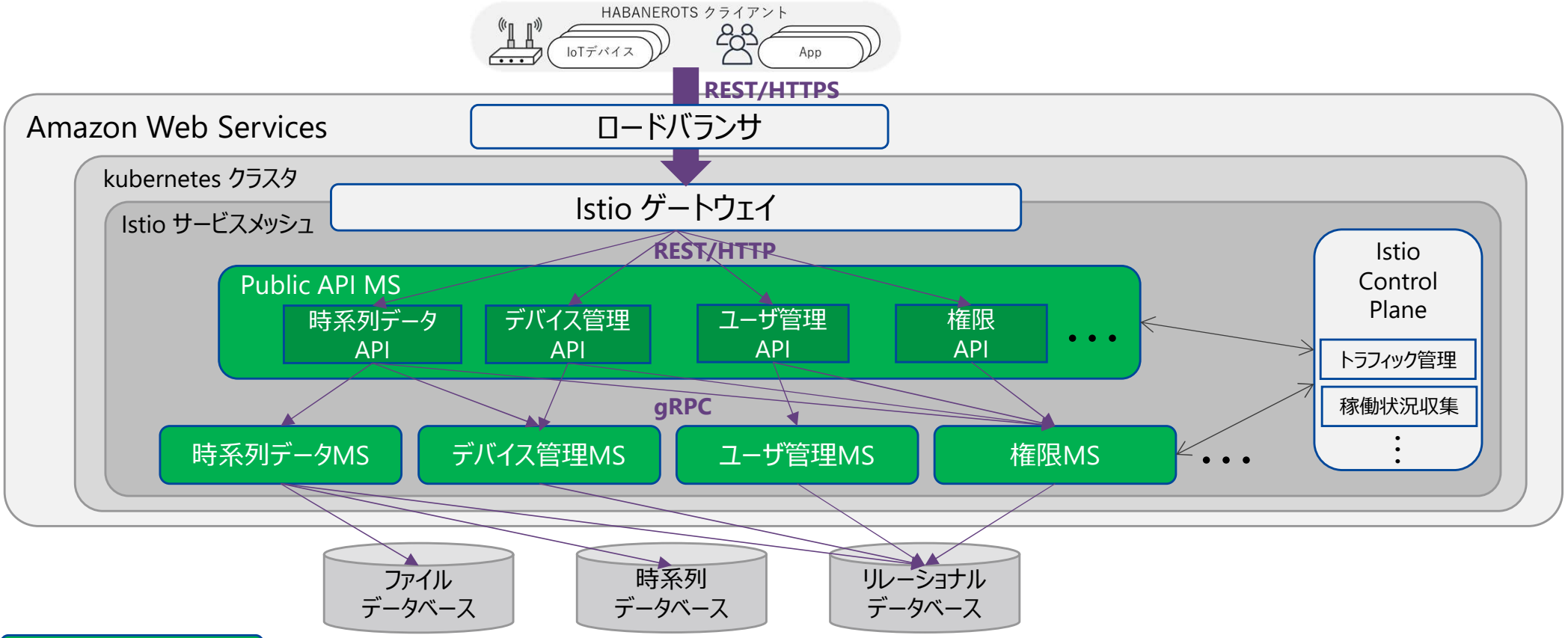


事業部門への提供機能

- アプリの登録・稼働手段
- アプリ脆弱性の確認ツール
- アプリから利用可能なデータベース
- ログ確認ツール
- アプリの死活監視

IoTデータ管理サービスのアーキテクチャ

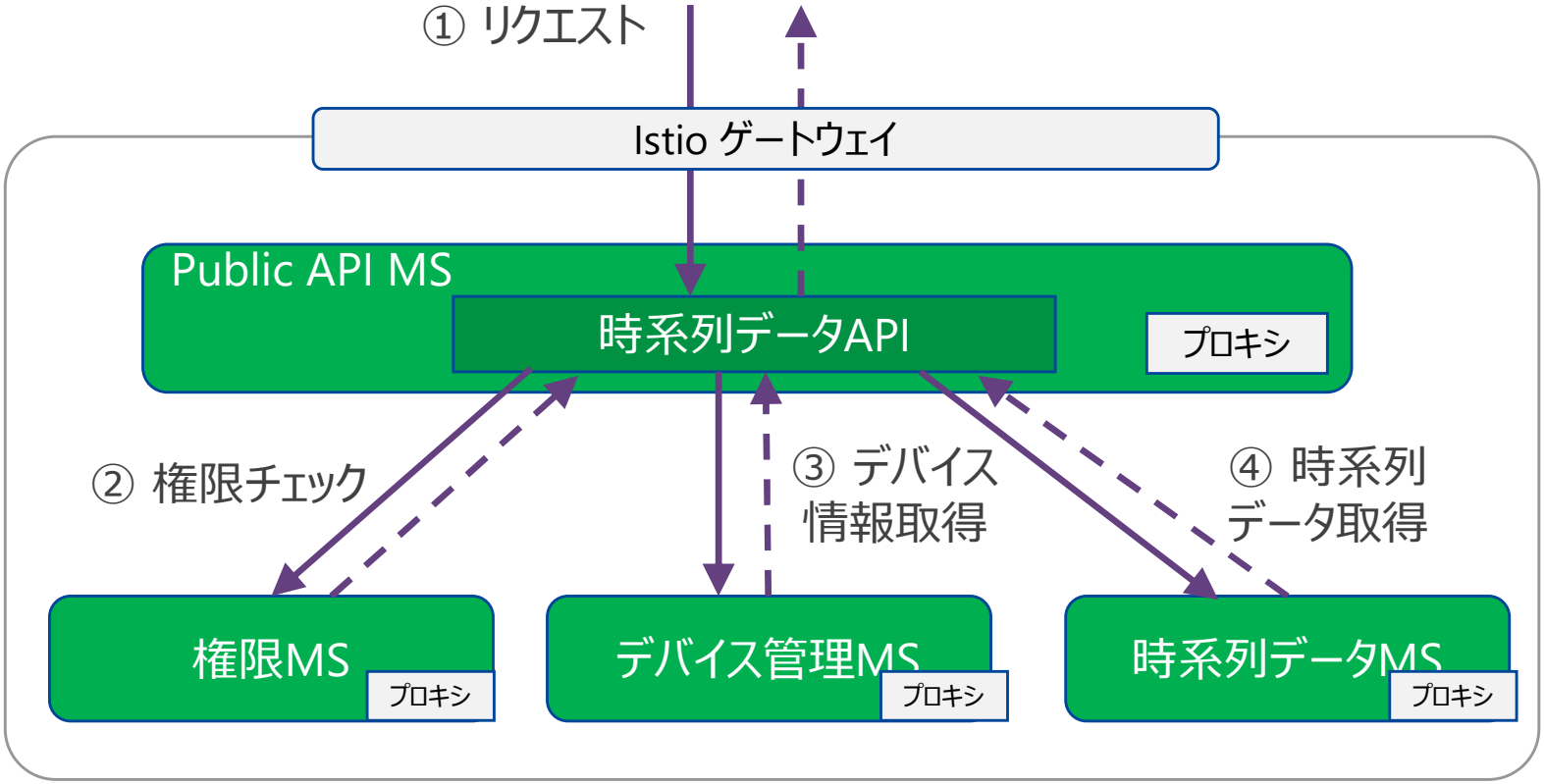
- **コンテナ**ベースの**マイクロサービスアーキテクチャ**＆**サービスメッシュ**を採用
  - 提供する機能群はそれぞれマイクロサービスとして実装、Dockerコンテナ上で動作
  - コンテナ・オーケストレーションツールとしてKubernetesを採用
  - 可観測性および通信ポリシー統制のためにサービスメッシュ（Istio）を採用



IoTデータ管理サービスのアーキテクチャ

- サービスの構成要素を役割ごとにマイクロサービスに分割して疎結合化
- マイクロサービス更新時の影響の最小化や柔軟なスケーリングを可能に

- さらにマイクロサービス間のリクエストをプロキシ経由で追跡し
  エラー発生個所や性能ボトルネックを特定できるように（可観測性）



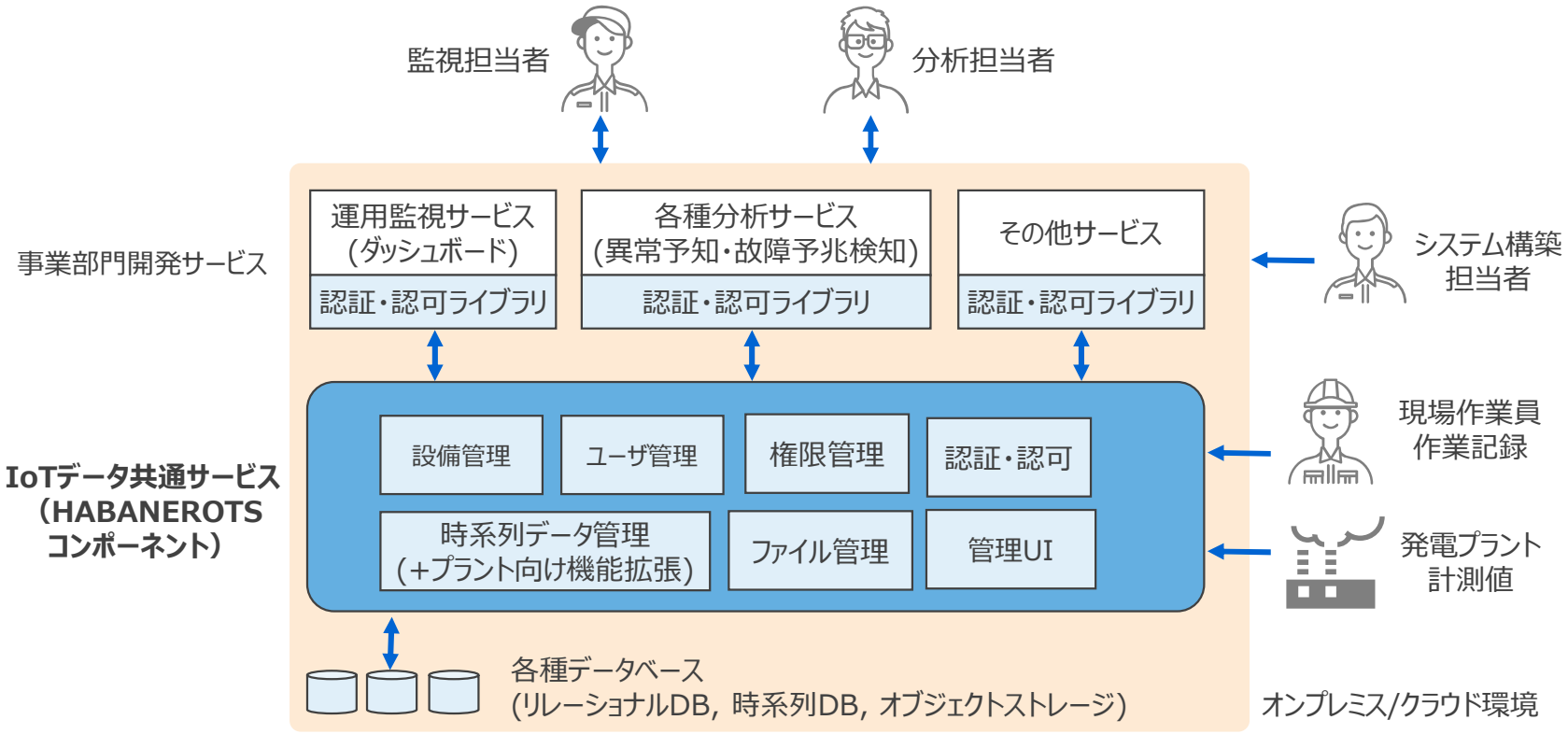
移植性とスケーラビリティ・可用性・運用性を両立させる方針で設計

展開事例：電力IoT基盤

- **エネルギーシステム向けIoT基盤のコアコンポーネントとしてHABANEROTSを適用**

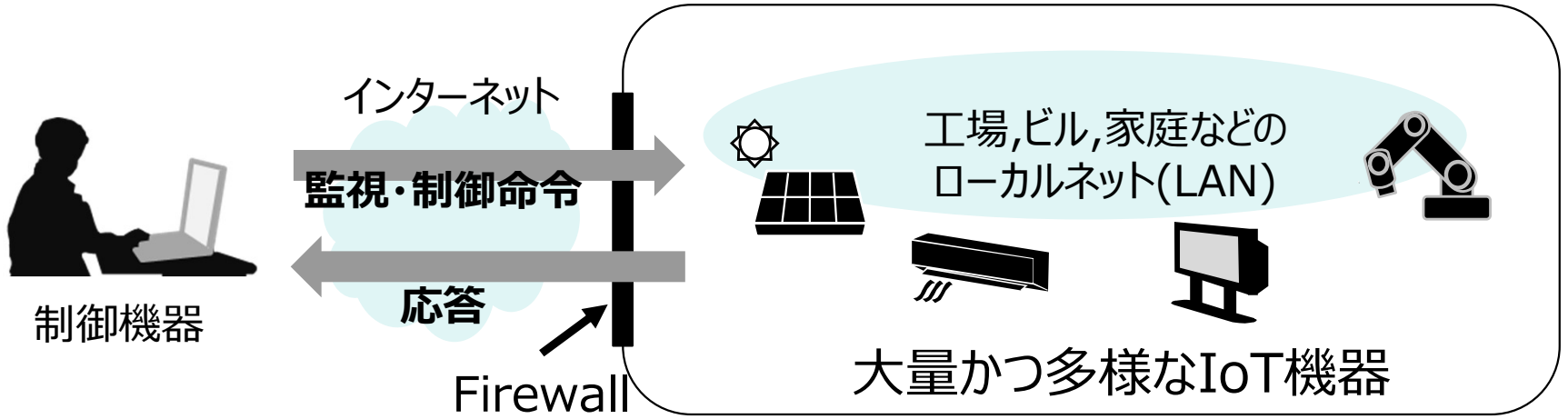
関連拡張：

- 発電プラントの用途を想定したAPI拡張（正常値の絞り込み、統計処理など）
- 環境構築簡易化手法（設備・信号などの発電所データを一括登録できる初期設定ツール）
- 認証・サービス連携の仕組みの構築（シングルサインオン）



双方向通信API

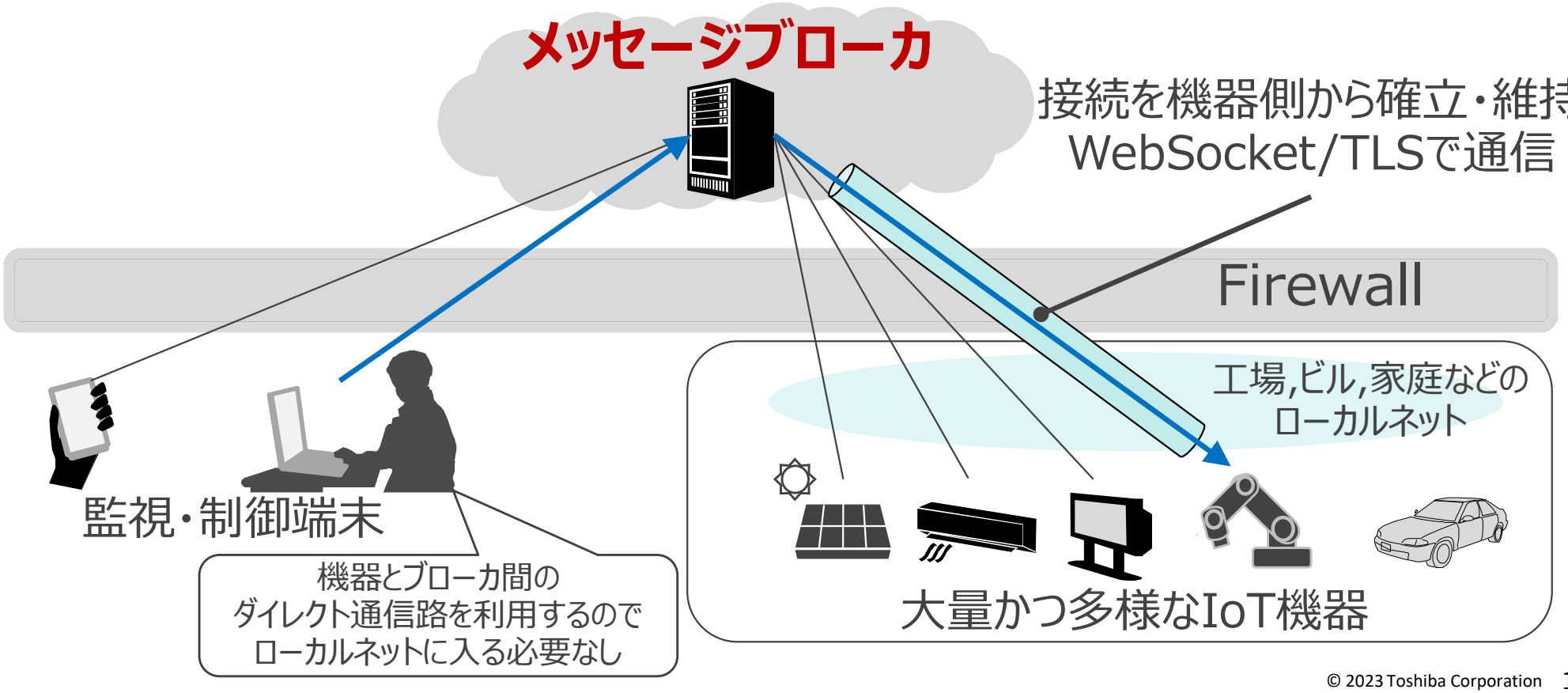
- **インターネットを介した機器の遠隔制御における課題**
  - **① 機器に対する遠隔からのリアルタイム双方向通信の実現**
    - 大半の機器はFirewallの内側にあるため外部から直接アクセスができない
    - 産業機器ではとくにリアルタイム性が重要
  - **② スケーラビリティの確保**



- **方針**
  - ① 制御端末とIoT機器との間で制御信号を中継する「**メッセージブローカ**」
  - ② 収容する機器数が増えてもクラウド側の負荷を一定に保つ「**接続調停方式**」

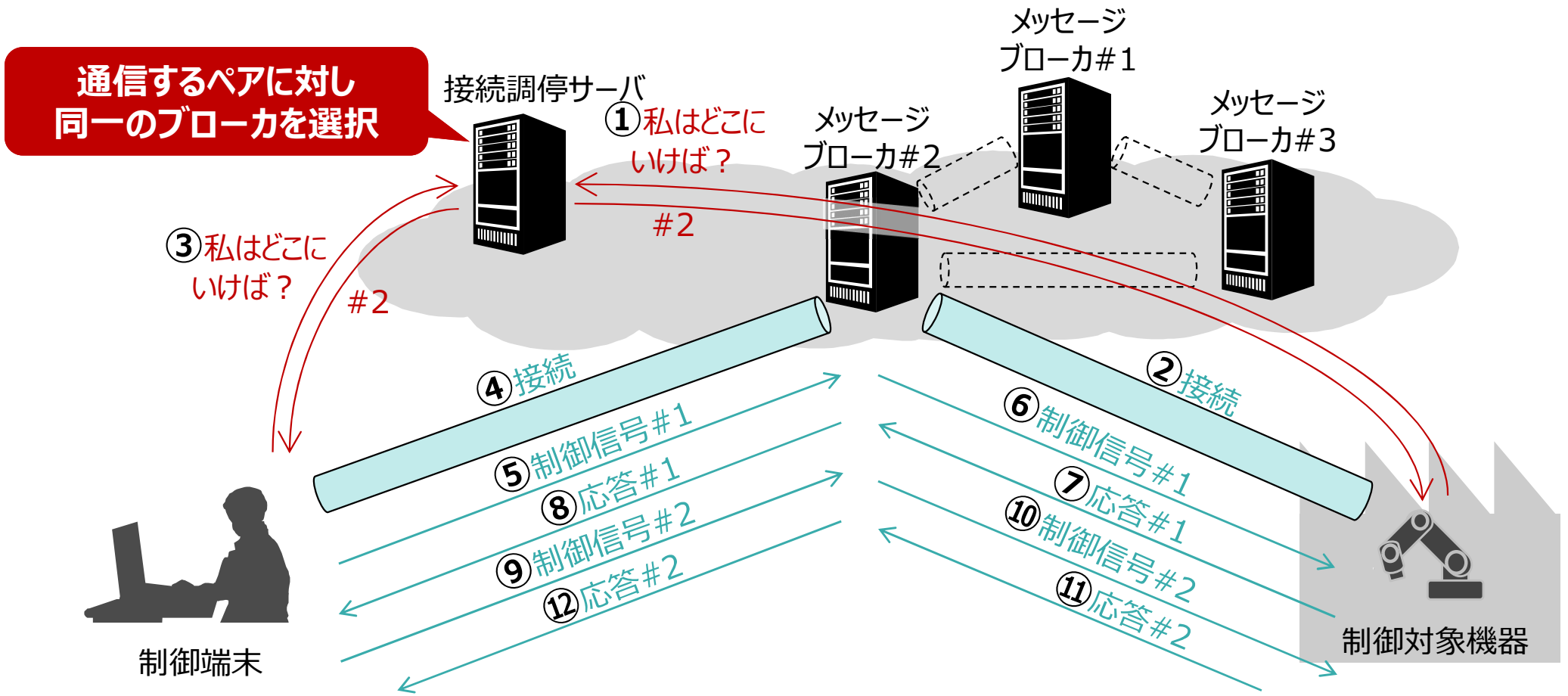
双方向通信API

- **監視・制御信号を中継する省リソースなクラウドサーバ**
  - 機器側から確立した通信接続を維持し、外部からローカルネット内の機器へのアクセスを実現
  - セキュアでリアルタイムな通信のため、Web通信技術であるWebSocket/TLSを活用
  - 制御信号の特性を利用しバッファ確保量を最適化、収容可能数を従来実装から約20倍に
    ⇒ ノートPC(メモリ4GB)クラスのブローカ1台に12万台以上の機器を収容可能



双方向通信API

- 相互通信する制御端末と対象機器を、同一メッセージブローカに接続
- 大量接続収容時にも、メッセージブローカ間通信が発生せず、クラウドの処理量が抑えられる



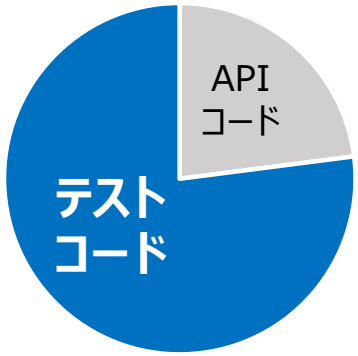
- 100万台規模でも正常動作することを開発環境で確認

開発支援の取り組み ーAPIテスト自動生成ー

- **API開発では、様々なパターンのリクエストで期待通りのレスポンスが得られるかどうかチェックする必要あり。**
- **工数短縮に向け、API仕様書からテストシナリオを自動生成するツールを開発**

HABANEROTS

- テスト関連コードが **約77%**
- すべて開発者が手作業でコーディング
- 境界値テスト、フォーマット違反テストなど、単調な作業が多い（が、品質担保には必要）



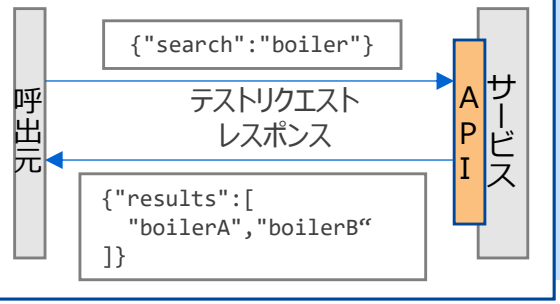
public-apiマイクロサービス
コード行数

現状の開発フロー

様々なパターンのテストコードを作成する手間がスピーディなAPI公開の足かせに

設計
実装
テスト
リリース

テストリクエストに対してAPIからのレスポンスが適切か検証



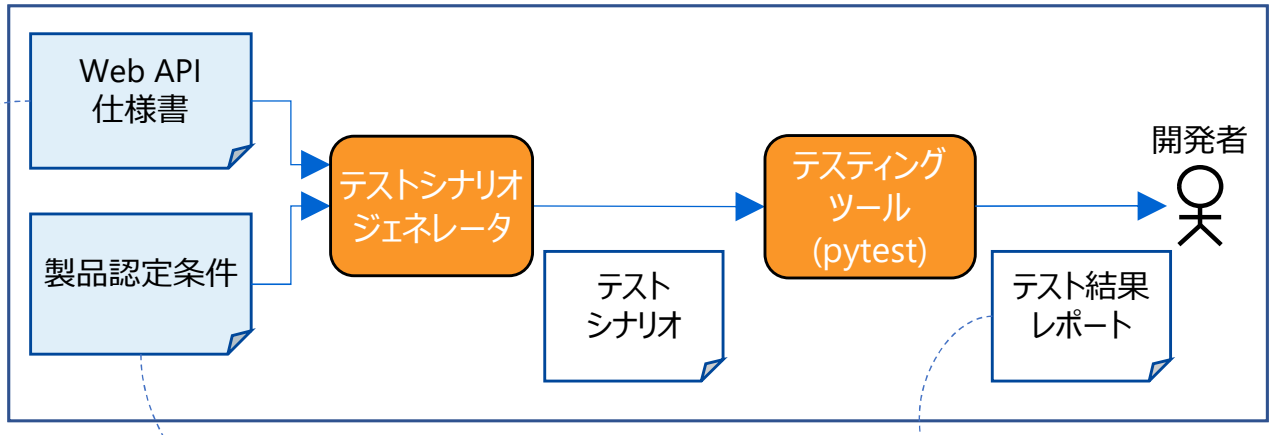
目指す姿

**テスト工程をツールで自動化
API開発の作業工数を短縮**

設計
実装
テスト
自動化
リリース

開発支援の取り組み －APIテスト自動生成－

構成図



API仕様書（OpenAPI形式）

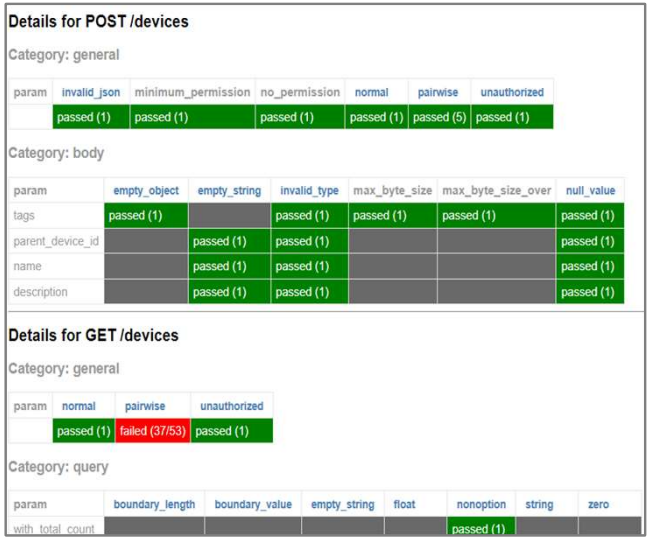
```yaml
post:
  summary: デバイス登録
  operationId: postDevices
  tags:
    - Device API
  security:
    - ClientAuthOAuth2: []
  description: |-
    <details> <summary> Version history </summary> <ul>
    <li> <code>auth_method</code> was introduced in <code>v2.5.0-pre-it5</code>.</li>
    <li> <code>resource_group_id</code> was documented in <code>v2.7.0-pre-it1</code>. H
    </ul> </details>

    デバイスを登録します。
  requestBody:
    content:
      application/json:
        schema:
          type: object
          properties:
            name:
              type: string
              description: デバイス名
            description:
              description: デバイスの説明
              type: string
            parent_device_id:
              type: string
              description: 親デバイスID。指定すると親デバイス認証情報を用いて時系列データの登
```

製品認定条件

- 正常系テストがあること
- 不正な型を指定した時のテストがあること
- 不正なフォーマットを指定した時のテストがあること
- 壊れたJSONを指定した時のテストがあること
- 認証ヘッダ漏れのテストがあること
- 必須パラメータがない時のテストがあること
- 数値パラメータの境界値テストがあること
- null を指定した時のテストがあること
- ・・・

結果レポート



Details for POST /devices

Category: general

param	invalid_json	minimum_permission	no_permission	normal	pairwise	unauthorized
	passed (1)	passed (1)	passed (1)	passed (1)	passed (5)	passed (1)

Category: body

param	empty_object	empty_string	invalid_type	max_byte_size	max_byte_size_over	null_value
tags	passed (1)		passed (1)	passed (1)	passed (1)	passed (1)
parent_device_id		passed (1)	passed (1)			passed (1)
name		passed (1)	passed (1)			passed (1)
description		passed (1)	passed (1)			passed (1)

Details for GET /devices

Category: general

param	normal	pairwise	unauthorized
	passed (1)	failed (37/53)	passed (1)

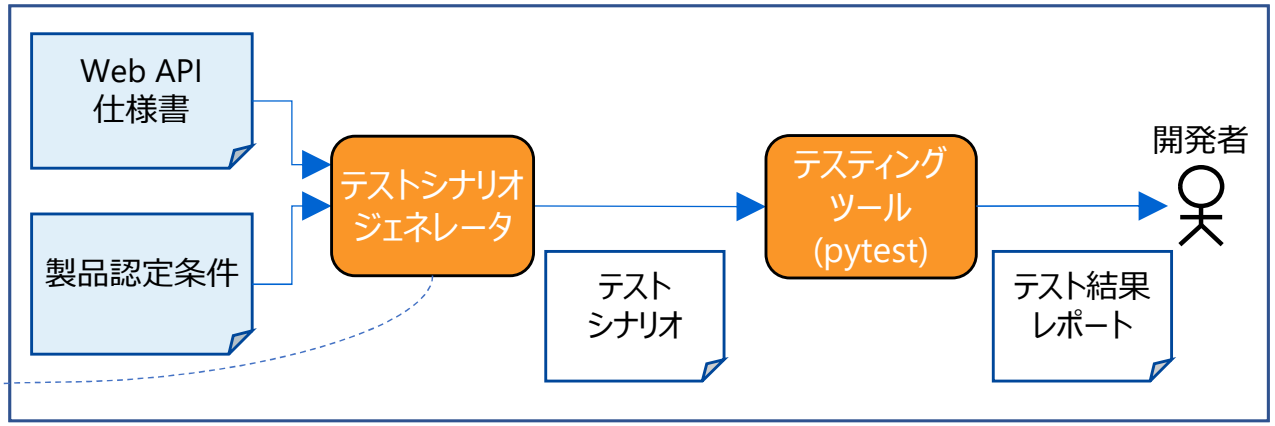
Category: query

param	boundary_length	boundary_value	empty_string	float	nonoption	string	zero
with_total_count					passed (1)		

（そのまま製品認定時のエビデンスに利用可能）

開発支援の取り組み －APIテスト自動生成－

構成図



仕様書に直接書かれていないAPI仕様を推定することで幅広いテスト生成を可能に

- 確認すべきステータスコードや、事前/事後条件・入力すべきパラメータの値を推定
- RESTful（Web APIの一般的な設計セオリー）を前提として推定

確認すべきステータスコードの推定

- 入力パラメータの誤り → 400 Bad Request
- 認証ヘッダ漏れ → 401 Unauthorized
- 不正ID → 403 Forbidden または 404 Not Found (方針次第)

事前/事後条件・入力すべきパラメータの推定

- 例：デバイス情報取得API （GET /devices/{device_id}）
  - 事前条件：デバイス作成API （POST /devices）
  - パラメータ {device_id} に入力すべき値：デバイス作成APIのレスポンスの値
  - 事後条件：デバイス削除API（DELETE /devices/{device_id}）

開発支援の取り組み ーAPIテスト自動生成ー

現在、HABANEROTSの109本のAPIを対象に1873件のテストを生成

仕様と実装の乖離の発見などに活用

- 壊れたJSONを指定しても成功してしまう
- 特定の条件で 500 Internal Server エラーが出る
- 特定のエラー時のステータスコードがAPIで揃っていない
- null指定時の成否がAPIで揃っていない
- ・・・

まとめ

IoT共通基盤 **HABANEROTS** を紹介しました。
事業部門が**CPSを始めるために必要な基本機能・環境を提供するマネージドサービス**です。事業部門の開発・運用コストを抑え、人材・技術・ノウハウの集約によりサービスの品質向上・競争力強化を図ります。

構成要素
① **IoTデータ管理サービス**：共通機能の Web API サービス
② **IoTデータ管理コンソール**：Web 管理画面
③ **ホスティングサービス**：事業部門自身がクラウド契約・運用することなくアプリを稼働できる環境

アーキテクチャ：コンテナベースのマイクロサービスアーキテクチャ＆サービスメッシュを採用
展開事例：エネルギーシステム向けIoT基盤

要素技術：
・双方向通信API：スケーラビリティのあるリアルタイム双方向通信
・開発支援の取り組み：API仕様書からテストを自動生成

TOSHIBA

※ 本文に掲載の商品の名称は
各社が商標として使用している場合があります