

TOSHIBA

Leading Innovation >>>

OSAKA NDS Embedded Linux Cross Forum #5

IoTのコストを低減する Webベースのクラウド基盤技術

東芝情報システム株式会社
エンベデッドシステム事業部
2017年7月

自己紹介

東芝情報システム株式会社
エンベデッドシステム事業部
(参事) 平井 秀治

経歴：

平成2年に東芝情報システムに入社。

さまざまなWebシステム開発に20年間従事し、5年前から組み込み系のアプリ開発にも携わり、Web技術と組み込みアプリとの融合に取り組んでいる。

現在は、(株)東芝と連携し、本日ご紹介するクラウド基盤技術のパッケージソフト開発を推進している。

目次

1. IoTクラウド基盤

- 基本機能と応用機能(アプリケーション)

2. 基本機能

- 遠隔監視・制御
- アクセス認証・認可

3. 低コスト化

- 汎用APIとマイクロサービス
- CIやIaaSの活用
- 事例紹介：FlashAir™ IoT Hub

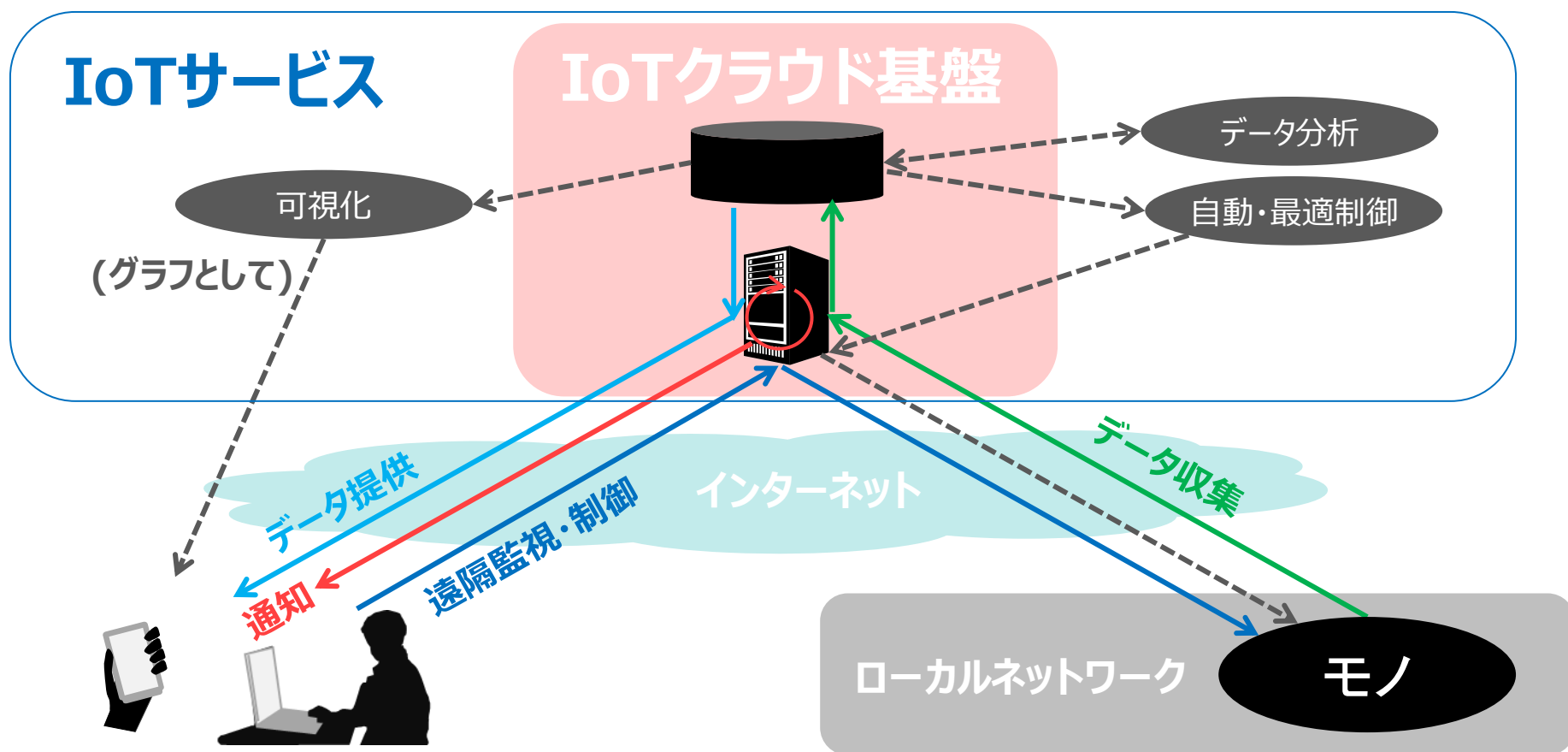
4. パッケージ提供について

1. IoTクラウド基盤

「IoTクラウド基盤」と「IoTサービス」

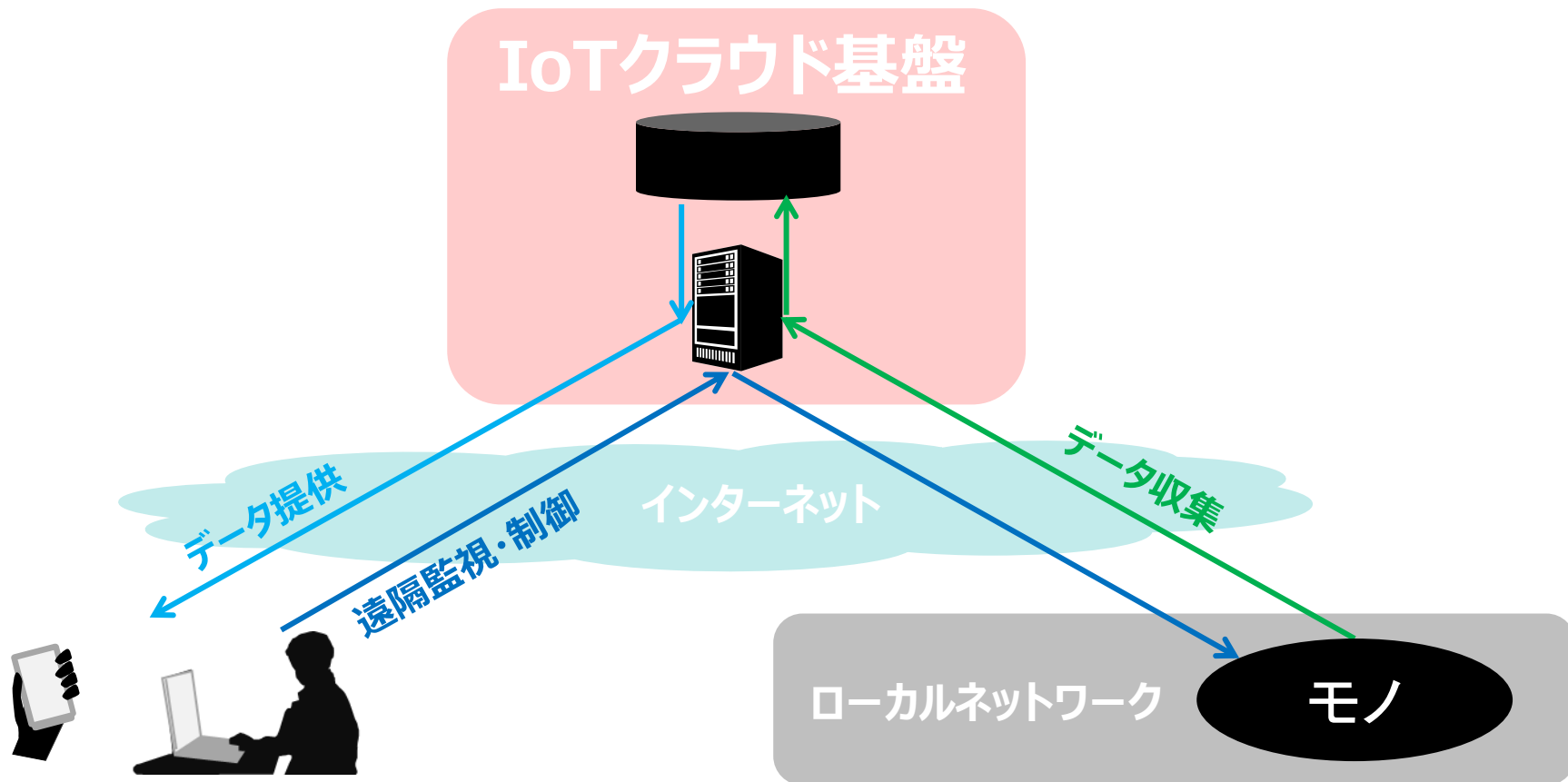
- IoTクラウド基盤: 汎用性の高い基本機能
- IoTサービス: 応用機能を含む全体

※本講演での便宜上の定義



IoTクラウド基盤

- 基本機能だけなら簡単に実現できる？
- デモレベルならYes

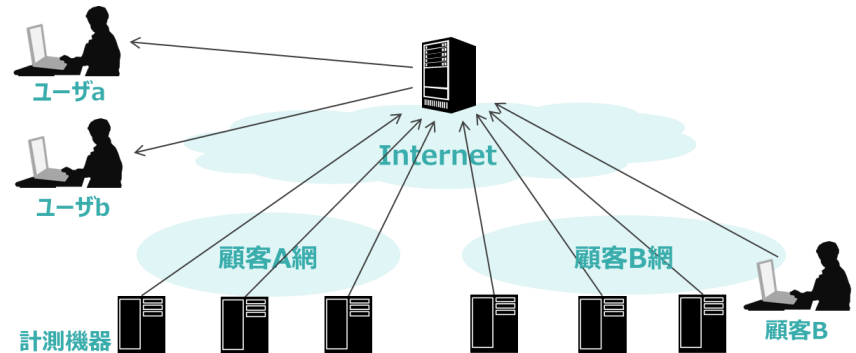


IoTサービスの本格運用における問題

【例：計測機器のアフターサービス】

計測機器のサービスマン(ユーザ)が、顧客に納入した多数の計測機器の稼動ログや計測値をネット経由で監視し、必要に応じてメンテナンスのための制御を行う

**本格運用しようすると
問題が山積み**



基本機能に限っても・・・

1. 遠隔監視・制御

- 顧客網の中の計測機器の可制御性, 通知や制御のリアルタイム性をいかに確保するか？

2. データ収集・管理・提供

- 貯まったデータの長期保持, バックアップは？ 検索・トラッキング, 統計処理も欲しい・・・

3. ユーザ・機器管理&セキュリティ

- 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. 可用性とスケーラビリティ

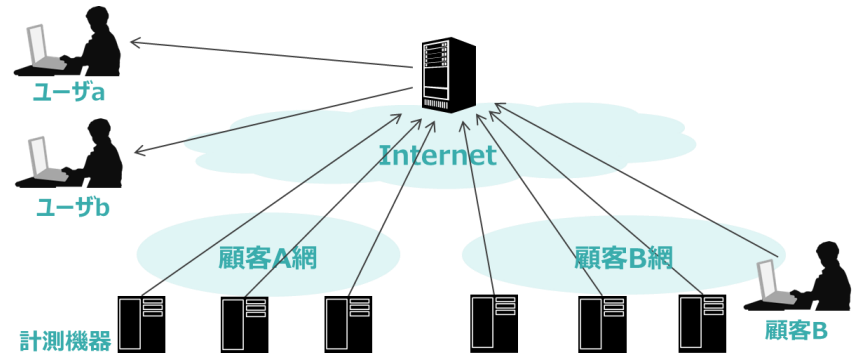
+ いかに安く済ませるか？・・・開発コスト, サービス運用・維持コスト, 設置等導入コスト

IoTサービスの本格運用における問題

【例：計測機器のアフターサービス】

計測機器のサービスマン(ユーザ)が、顧客に納入した多数の計測機器の稼動ログや計測値をネット経由で監視し、必要に応じてメンテナンスのための制御を行う

本格運用しようすると
問題が山積み



基本機能に限っても・・・

1. 遠隔監視・制御

本日のトピック

- 顧客網の中の計測機器の可制御性、通知や制御のリアルタイム性をいかに確保するか？

2. データ収集・管理・提供

- 貯まったデータの長期保持、バックアップ、検索・トラッキング、統計処理も欲しい・・・

3. ユーザ・機器管理&セキュリティ

本日のトピック

- 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？

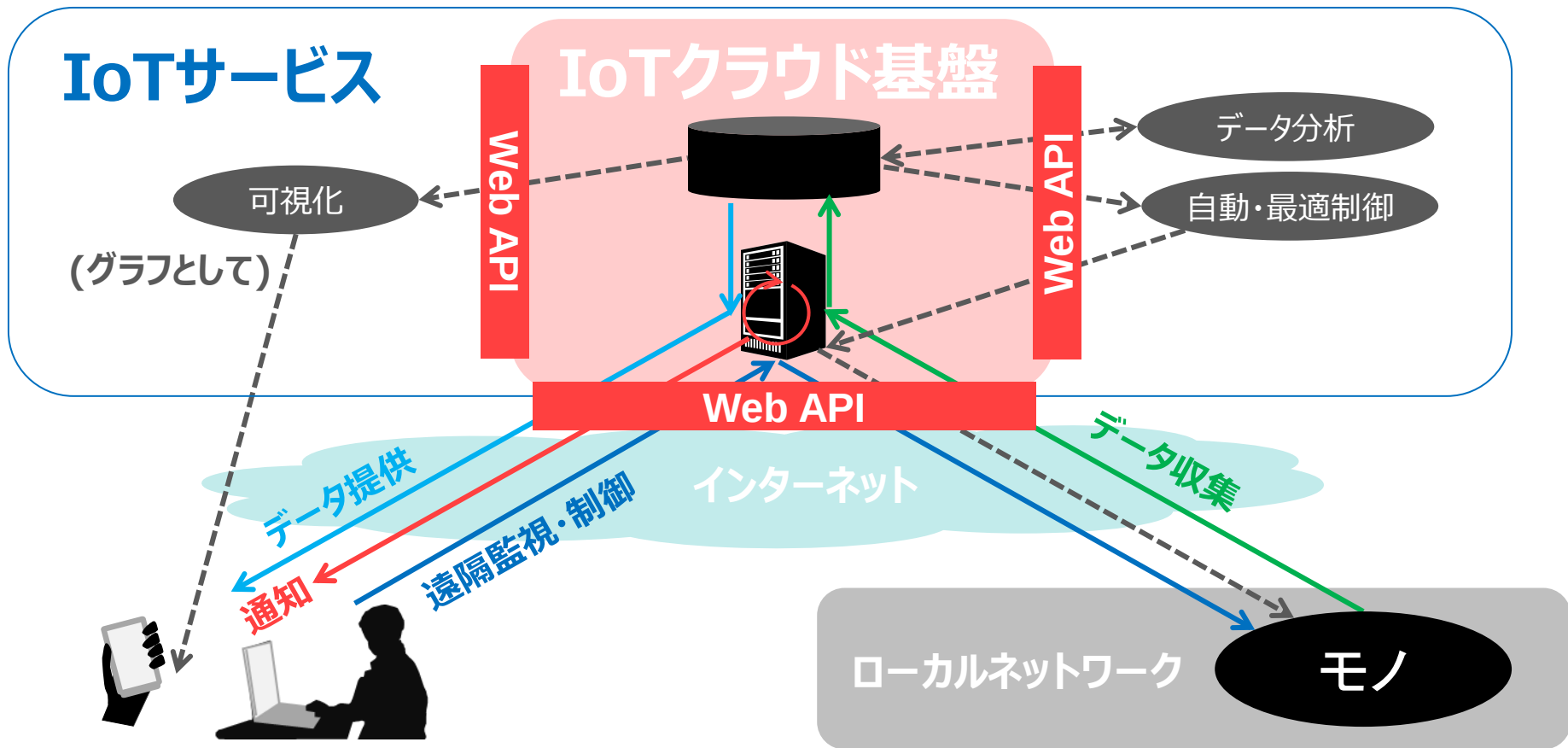
4. 可用性とスケーラビリティ

本日のトピック

+ いかに安く済ませるか？・・・開発コスト, サービス運用・維持コスト, 設置等導入コスト

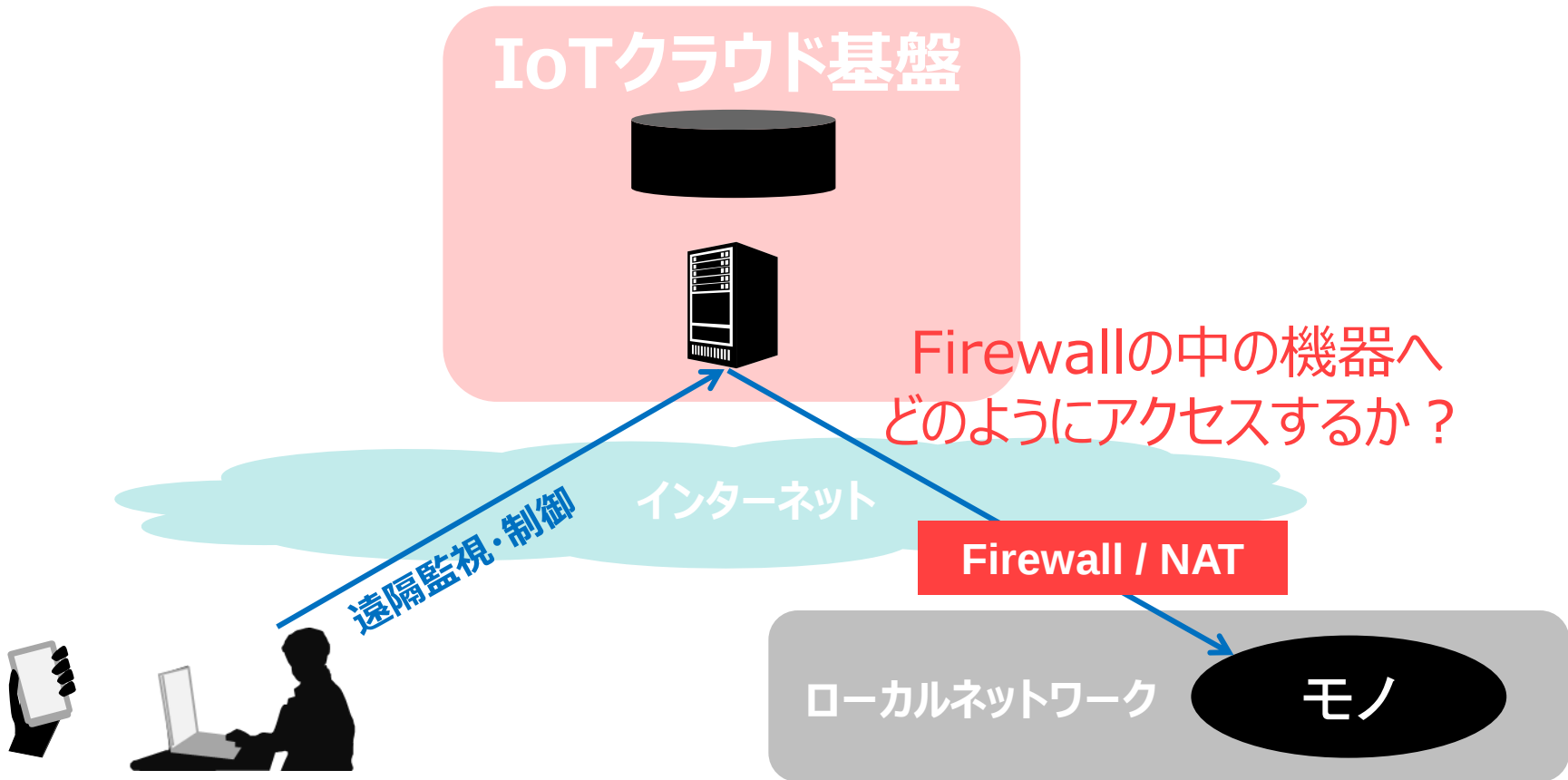
IoTクラウド基盤の基本アーキテクチャ

- 全ての機能をWeb APIとして提供
 - － 機能拡張, アプリ連携, セキュリティを考慮した場合の常識的な選択肢



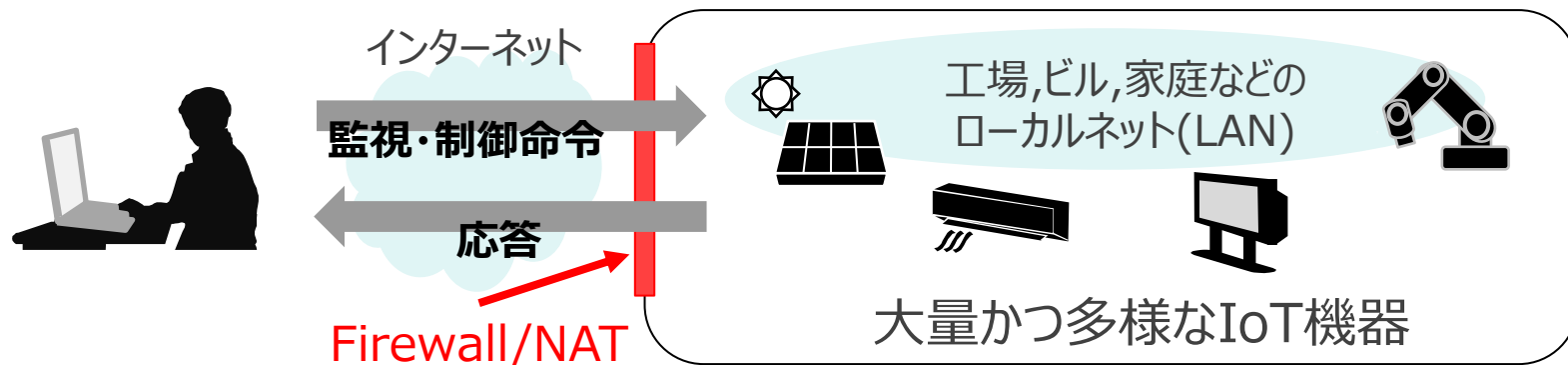
2. 基本機能

遠隔監視・制御



遠隔監視・制御

- インターネットを介した遠隔からの機器の監視・制御
 - 家電, 工場設備, ビル設備など



● 主要な要件

① 様々な環境におかれた機器に対する遠隔からのアクセスの実現

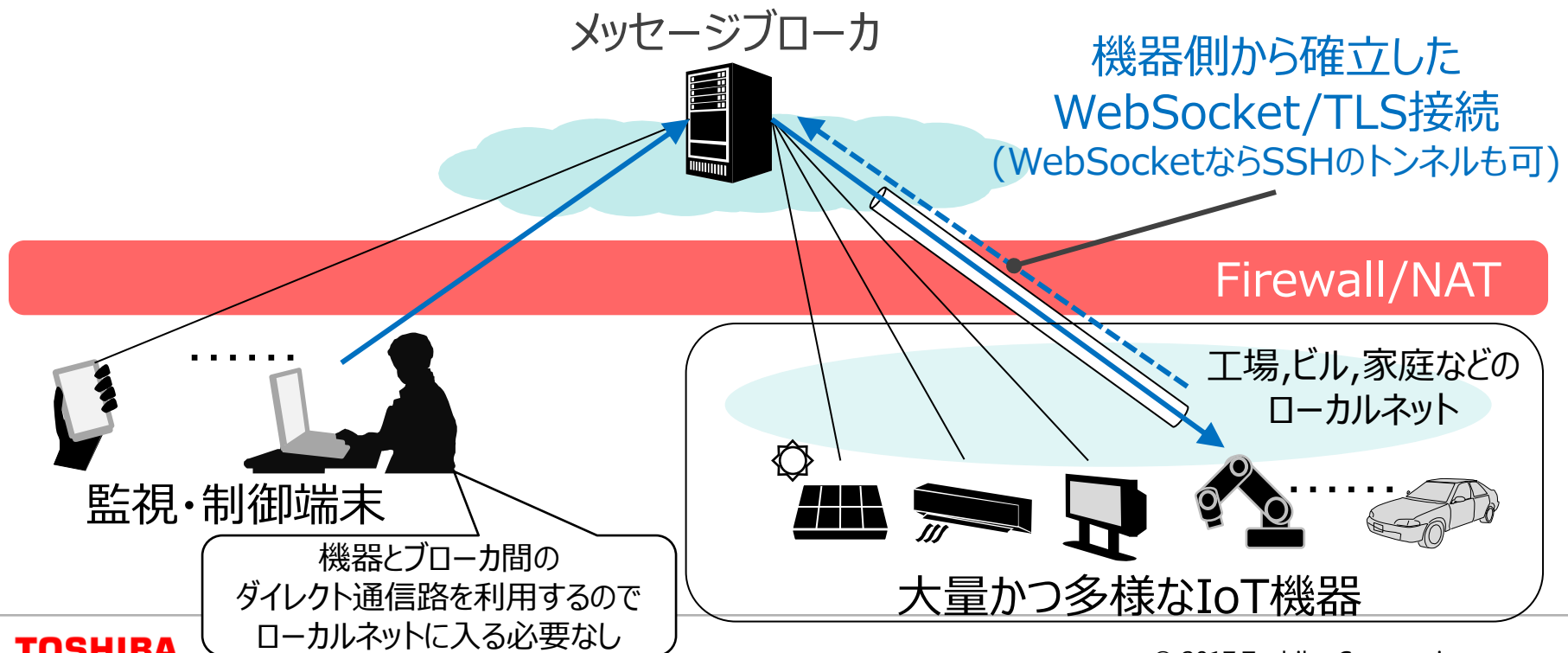
- 大部分の制御対象機器がローカルネット(LAN)の中
 - FirewallやNATの配下にあるため直接外部からアクセスできない
- LANの入り口に個別の専用ゲートウェイ設置はコスト高, 実現困難な場合も

② 監視・制御のリアルタイム性の確保

- ユーザビリティ, アプリケーションとして必須

遠隔監視・制御

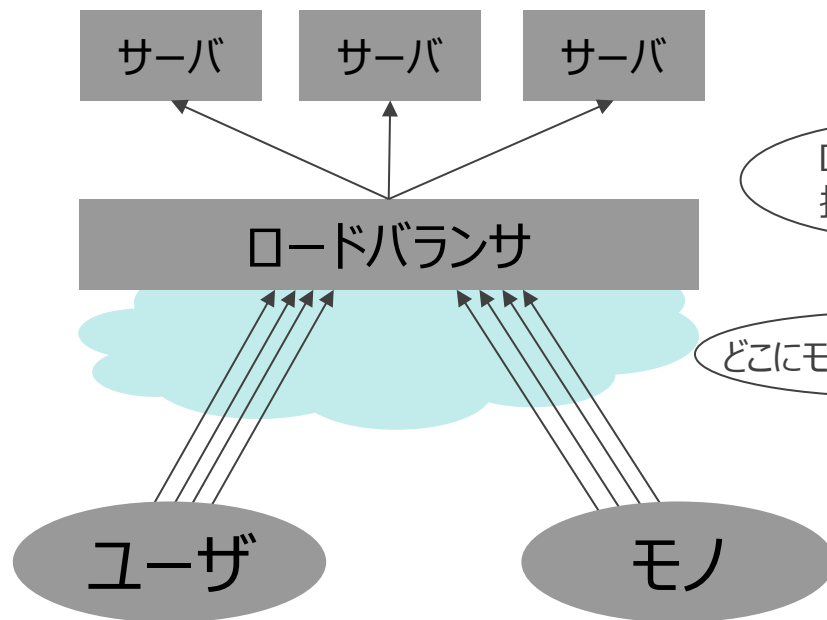
- 監視・制御メッセージを中継するメッセージブローカを活用
 - WebSocket, MQTT
- ポイントは機器側から確立して,それを維持すること
 - 専用GWや特別なFirewall設定なしでリアルタイムアクセス可能



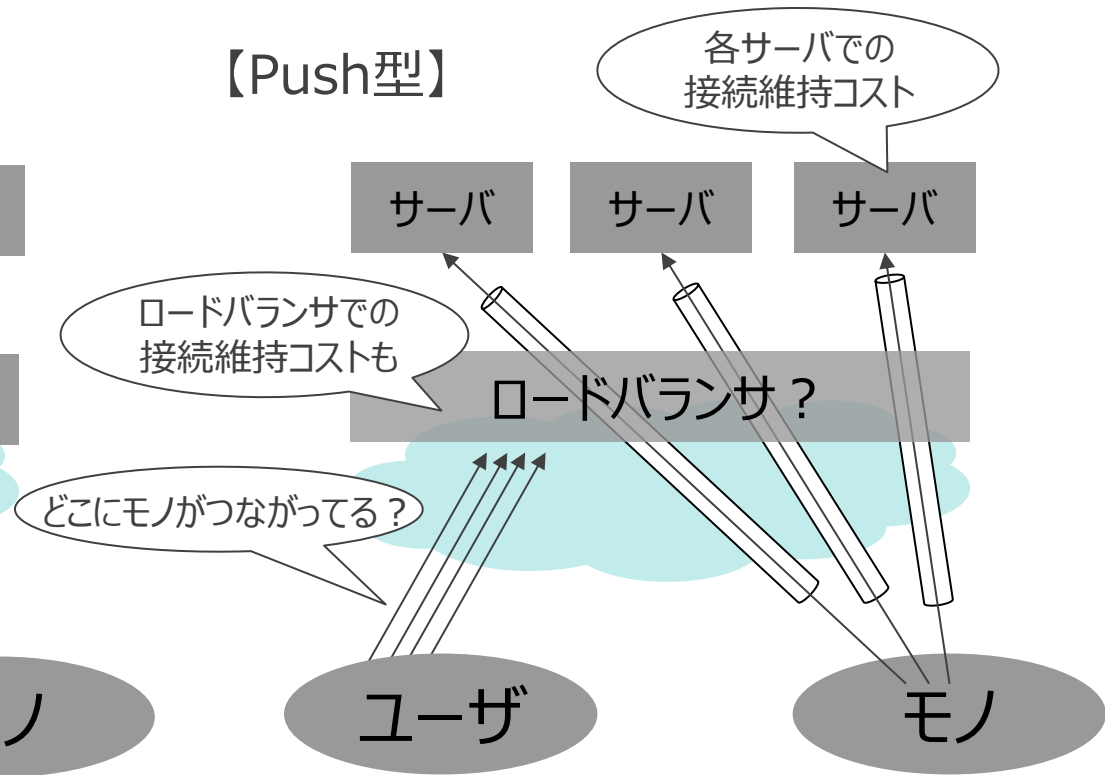
遠隔監視・制御の大規模化(スケールアウト)

- WebSocket等のプッシュ型通信はスケールアウトが困難
 - 一般的なPull型は既存のロードバランサで容易だが・・
 - セキュアなTLS接続は固定的なメモリ消費を伴い高コスト

【Pull型】



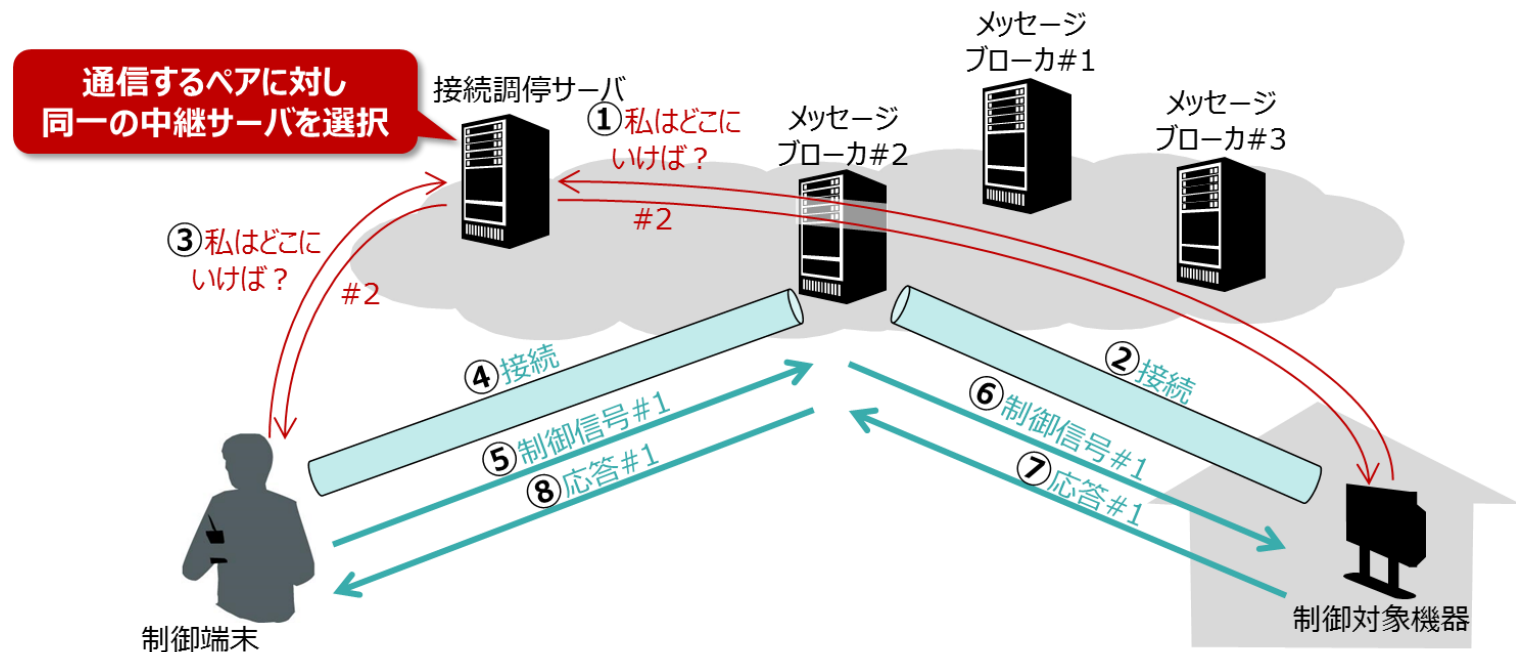
【Push型】



遠隔監視・制御の大規模化(スケールアウト)

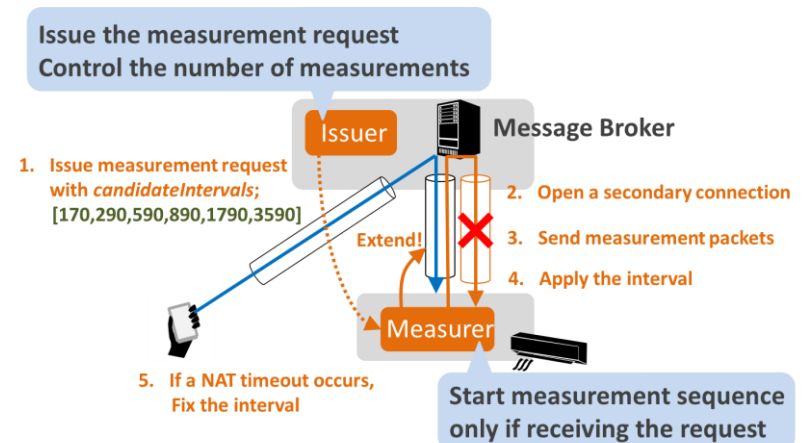
● 接続調停アーキテクチャ

- ロードバランサを使わず, クライアントに接続先ブローカを教える接続調停サーバを導入
- 相互通信する制御端末と対象機器を, 同一ブローカに接続し, 収容台数が増えても遅延を一定に保つ



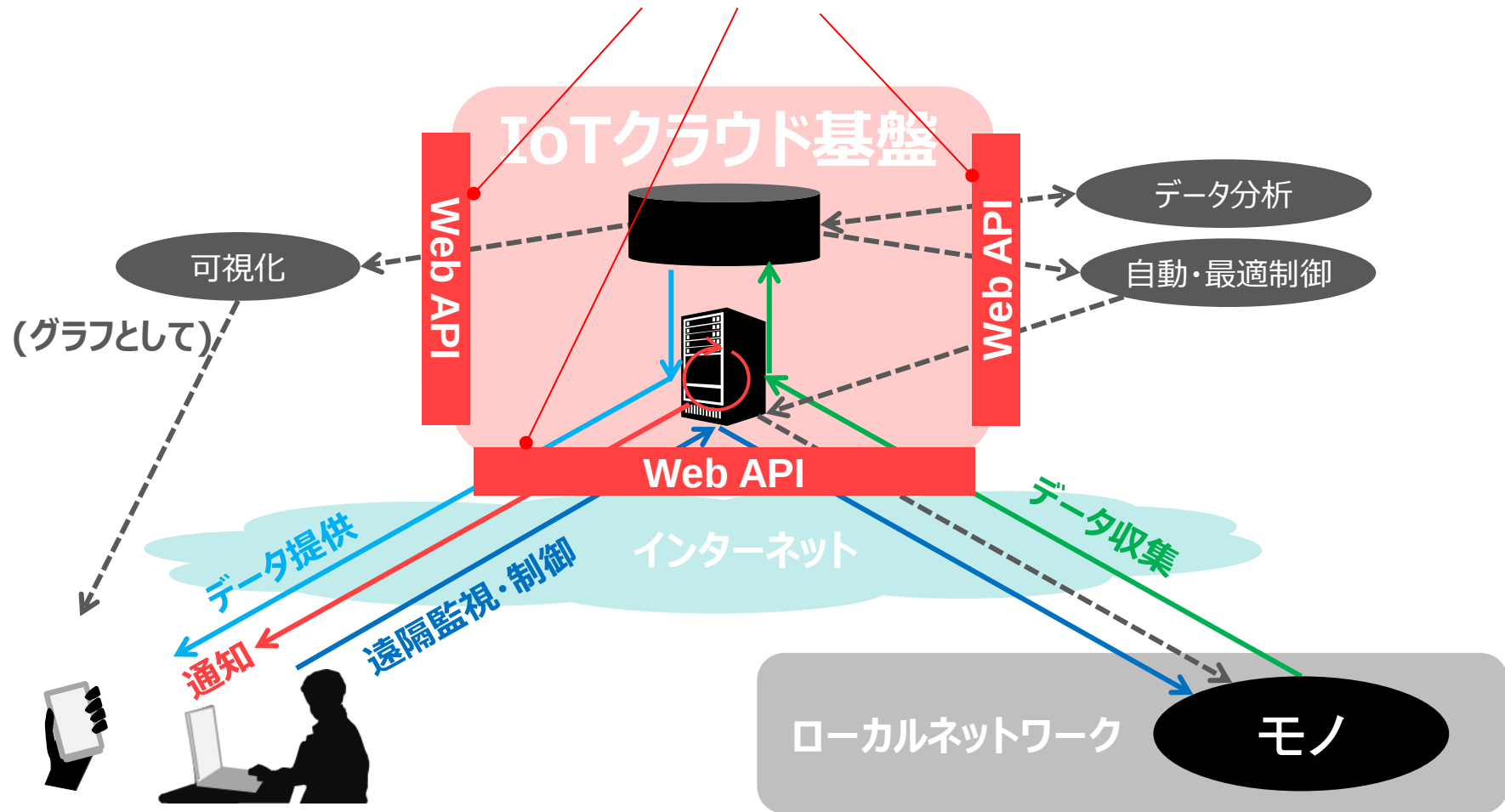
遠隔監視・制御の低コスト化

- IoT向けに特化した省リソース化
 - TLS/TCPソケットバッファサイズが問題。用途に応じた調整によりメモリ4GBのブローカ1台に12万台以上の機器収容も可能
- 接続維持にかかるコストの低減
 - TLS接続を維持するには、定期的な通信(keep-alive)が必要
 - 大規模時には無視できないコストに
 - 環境に応じて周期拡大することでサーバ負荷や通信量を低減



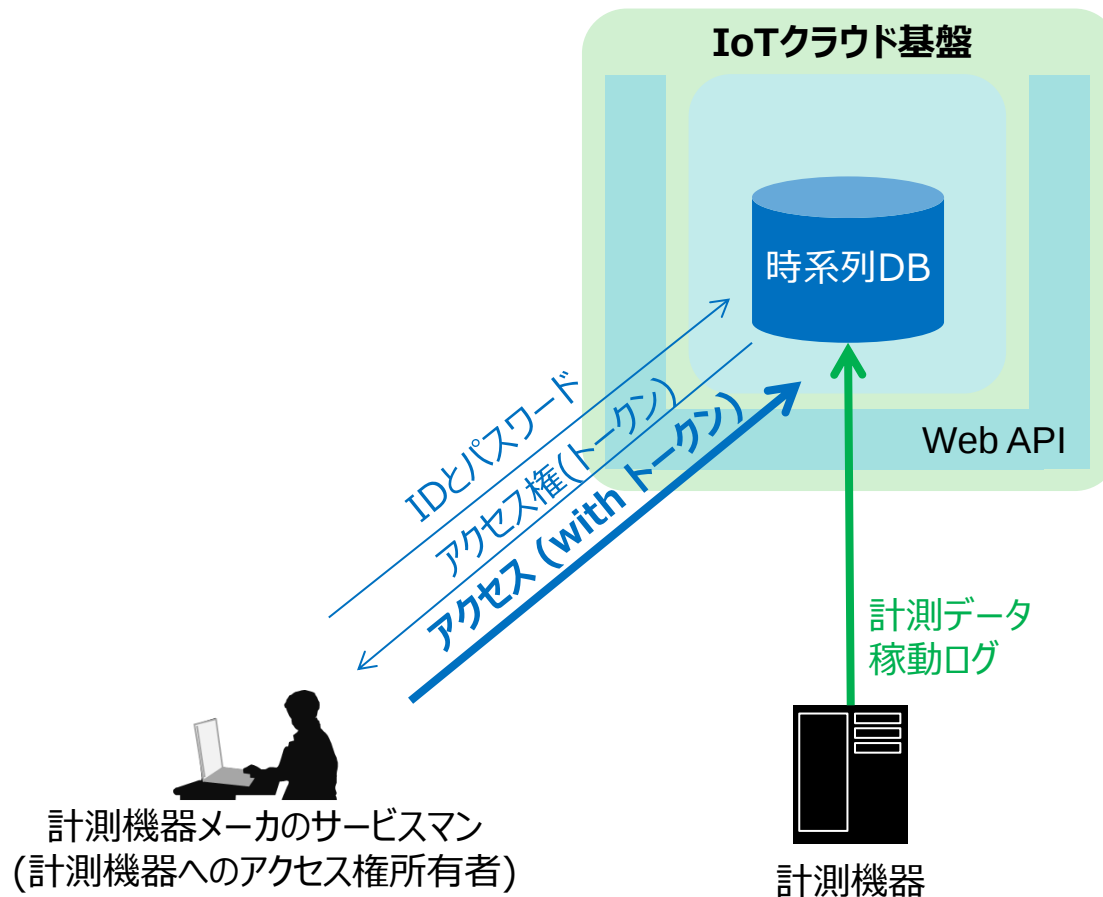
アクセス認証・認可技術

適切なアクセス権限をもったユーザにセキュアにアクセスさせるには？



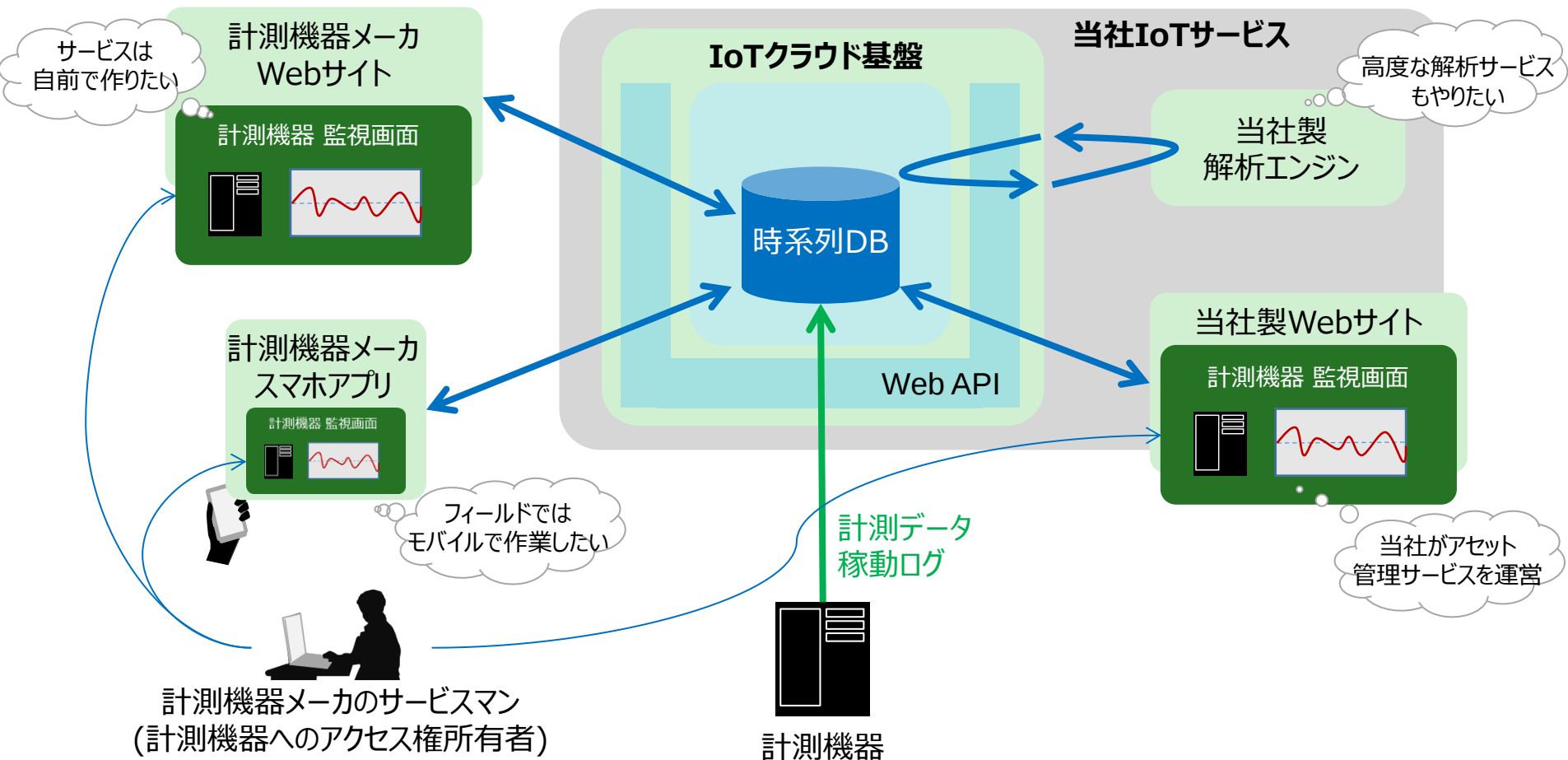
IoT向けアクセス認証・認可基盤

- 単にユーザのID,パスワードで認証すればよいだけでは？
 - － 話はそれほど単純ではない



IoT向けアクセス認証・認可基盤

- **問題:**「モノ」のアクセス権所有者と、実際のデータ利用者(クラウドハブを利用する連携システム・アプリ)は異なる
- **課題:** 多様な形態のデータ利用者にいかにアクセス権を与えるか？



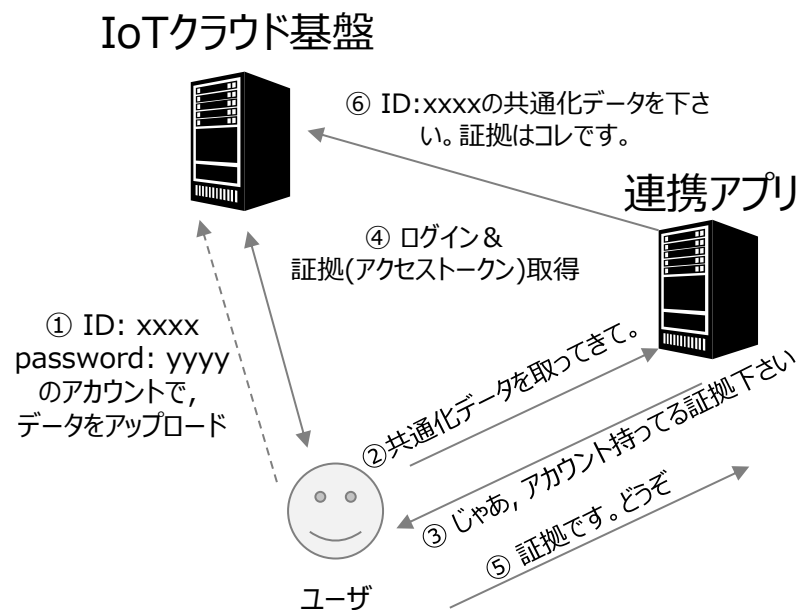
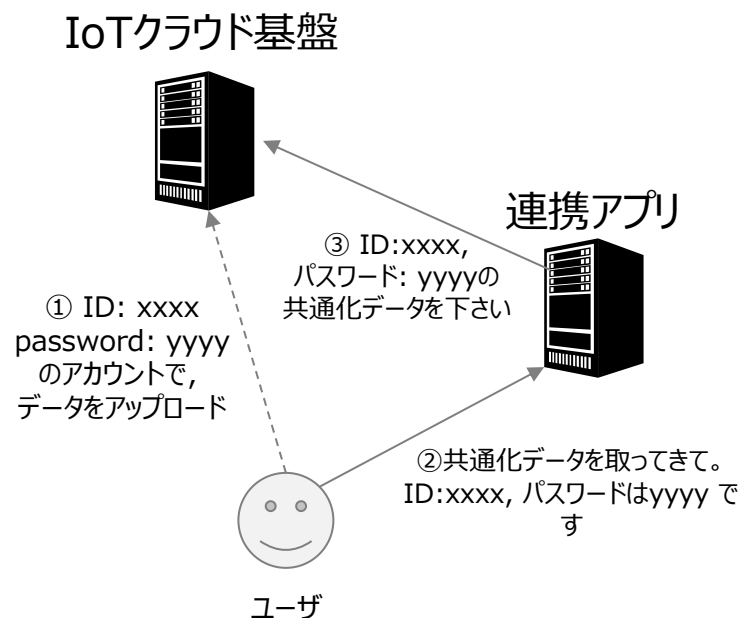
OAuth2.0（Web標準のアクセス認可プロトコル）

- 左図のやり方は危険

⇒ 連携アプリが、ユーザのパスワードを知ることが出来てしまうから

- 解決方法: OAuth2.0(+OpenID Connect)

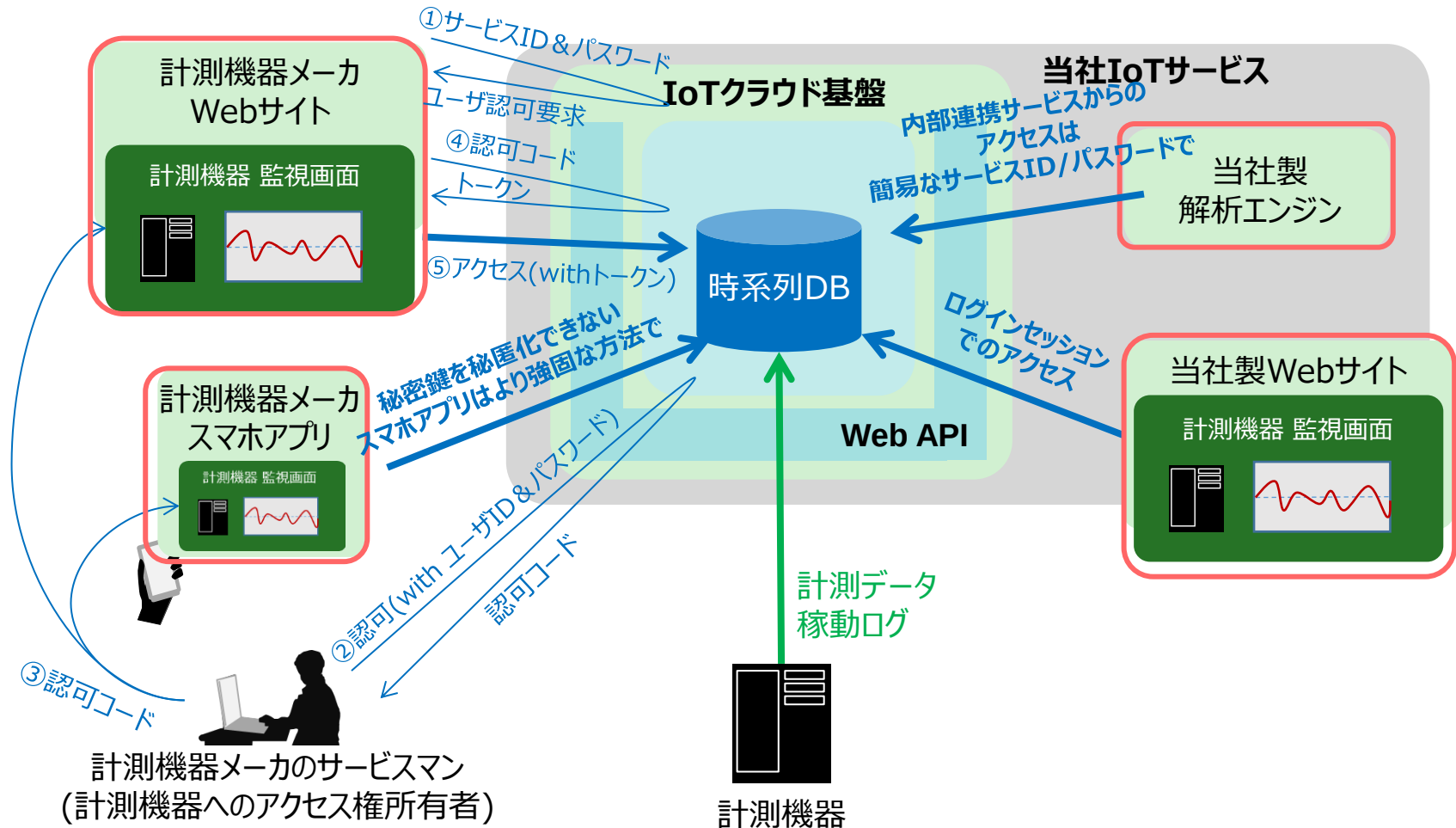
⇒ ログインはクラウドIoT基盤上で行い、アプリはユーザ認可の結果（アクセストークン）のみもらい、これでサーバ上のデータにアクセスする仕組み



※ 正確な手順とは異なり若干簡略化しています

IoT向けアクセス認証・認可基盤

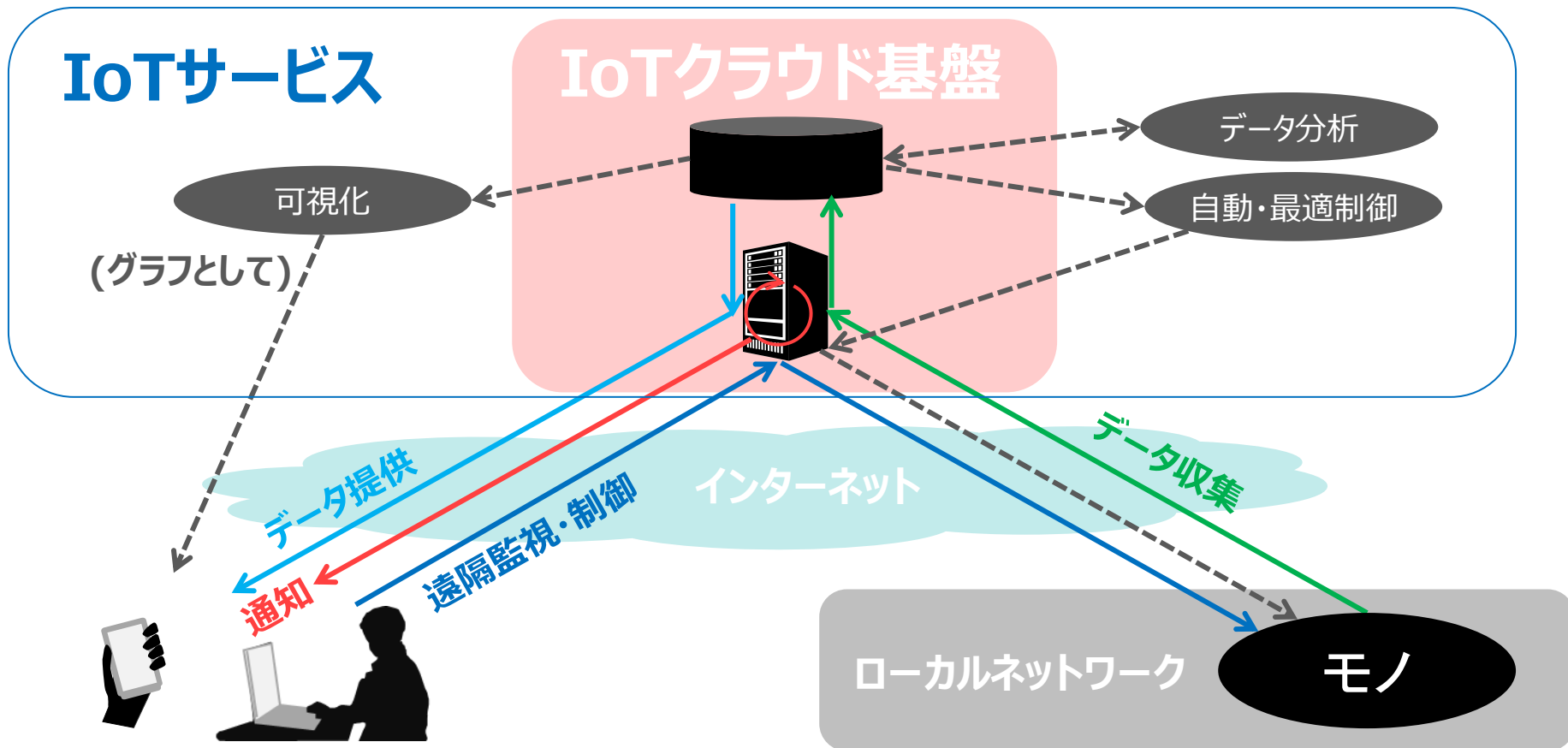
- OAuth 2.0に準じた複数の認可方式に対応することで、多様な連携アプリとセキュアに連携することが可能



3. 低コスト化

低コスト化

いかにサービス維持コストや開発コストを低減させるか？



低コストなIoTクラウド基盤への取り組み

- 基本機能そのものの改善によるサービス維持コスト低減

- いかにか少ない計算リソース・通信量で、大量のモノのデータ収集, モノとの相互通信を達成するか？

- クラウドは従量課金, 性能向上がコスト低減に直結

⇒ 前述の省リソースメッセージブローカ技術

- 多様なサービス固有要件への対応コスト低減

- 多数の案件に、いかにか少ない手数で対応できるようにするか？

- 基本機能と応用機能(サービス固有機能, アプリ)の役割の分離手法

⇒ 前述のOAuthベースの認証認可技術

- 基本機能を改変せずに多様なIoTサービスの機能要件を満たす手法

⇒ 汎用Web API&マイクロサービスアーキテクチャ

- IaaS・OSS活用によるサービス開発・運用コスト低減

- 結局いちばん食うのは開発やメンテナンスに係る人的コスト。どう下げるか？

⇒ CIの整備, IaaSの積極活用

汎用Web API&マイクロサービスアーキテクチャ

- 問題

- IoTサービスには各々固有の要件があり, 個別対応はコスト大
 - 機器情報の属性, 時系列データの種類・形式, 監視・制御命令フォーマット

- 課題

- いかにか基本機能のロジックやデータベースを改変せずに多様なIoTサービスの機能要件を満たすか？

- アプローチ

- 機能：あらゆるIoTサービス共通で使える汎用Web APIを整備
 - サービス固有要件は, 汎用APIを利用するアプリカラッパーサービスで対応
- データ構造：サービス固有データはオブジェクトデータとして管理
 - リレーショナルDBのJSON型や, ドキュメント指向DBが一般化
 - こうした技術を利用して, 個別スキーマ定義をなるべく避ける

汎用Web API&マイクロサービスアーキテクチャ

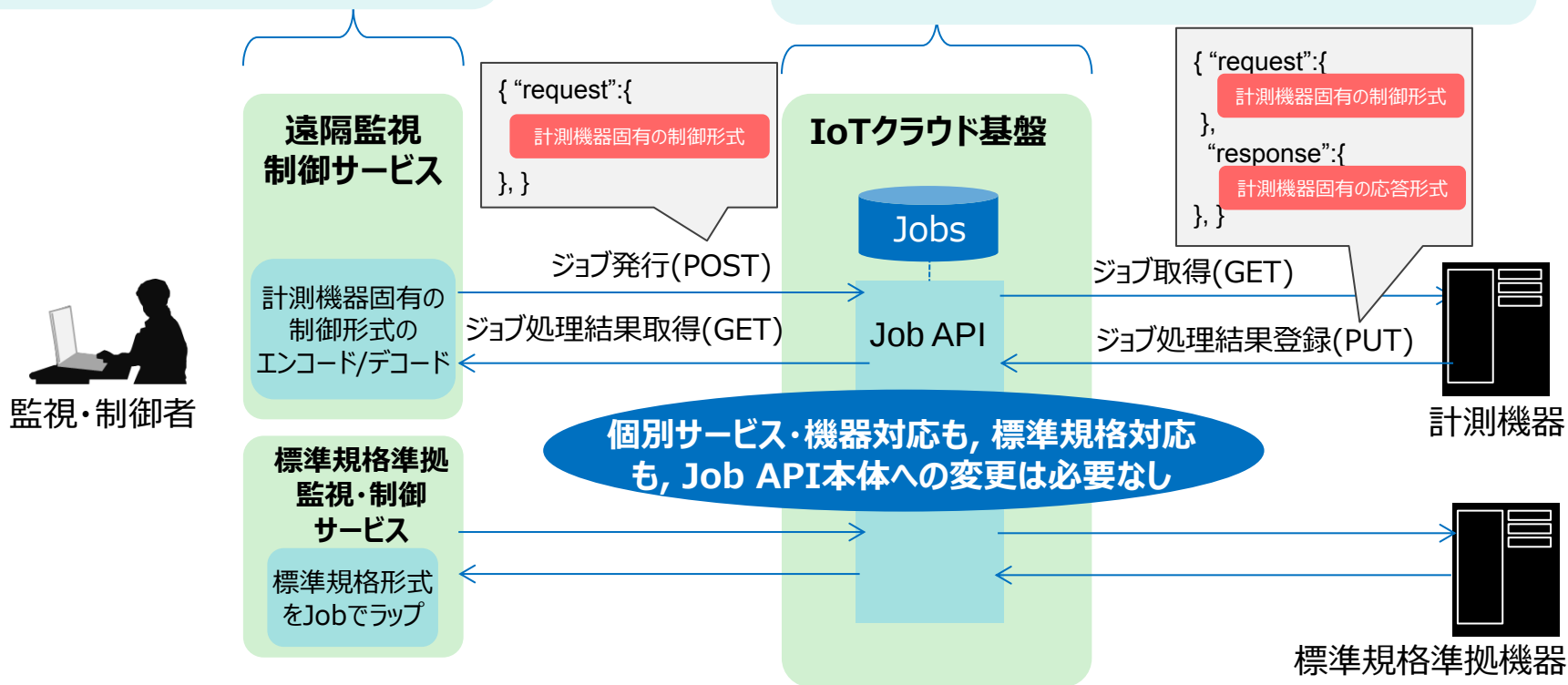
- FlashAir™ IoT Hubにおける汎用API
 - **Job API**
 - 遠隔監視・制御命令(ジョブ)を扱う
 - **Measurement API**
 - 機器からの時系列データアップロード, 時系列データ・統計データの取得
- 以下で公開されています
 - <https://iot-hub.flashair-developers.com/ja/api.html>

汎用Web APIによるマイクロサービスアーキテクチャ

- Job API: 遠隔監視・制御用高信頼・低遅延メッセージングAPI
 - “Exactly Once”なメッセージングをRESTfulなセマンティクスで実現

機器やサービス固有の処理への対応や標準規格等への対応を、コアから切り離れた**小さなサービス**として実現できる

クラウドハブは、ジョブの中身には一切関与せず
ライフサイクル管理, 排他制御, 順序保証, タイムアウト制御, 高信頼・低遅延配送を担当



CIの整備・IaaSの積極活用

- Jenkins, Gitlab CI などCIツールの活用
 - ソースコードをコミット
 - テストが自動実行
 - Docker化 → Dockerイメージをクラウドのリポジトリに登録
 - コマンド 1 つで準本番環境や本番環境に展開
 - 徐々にイメージを新しいものに切り替えていくローリングアップデート
- IaaSの活用
 - 上記のDockerクラスタ管理機能だけでなく、人的コストを低減するさまざまなサービスを提供
 - データベースのマルチデータセンター運用, データバックアップ
 - 計算機やデータベースのメトリック監視やアラート通知
 - オートスケーリングや, サーバ証明書自動更新などなど

TOSHIBA

Leading Innovation >>>



事例紹介 **IoTの開発・運用コストを低減する** **FlashAir™ IoT Hub**



G+1 チェック

ツイート

さらに速く、もっと快適に、
撮った写真をその場でシェア。



新発売

基本性能と無線転送速度が進化した、
第4世代FlashAir。

無線LAN搭載SDメモ리카ード

 **FlashAir**

引用元: <http://www.flashair.info>

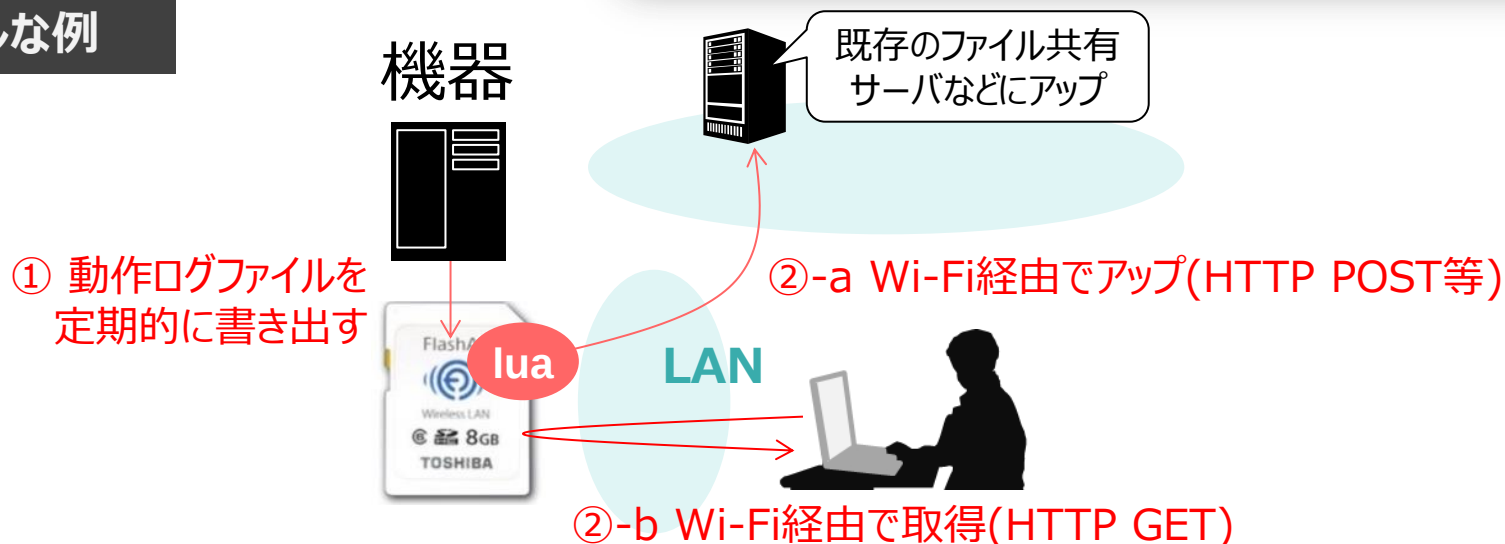
FlashAir™単体のIoT

- 既存の機器をカンタンにIoT化できるアダプタとして有力
 - SDスロットがあればOK, 技適をとる必要なし, 大容量キャッシュで信頼性向上
- スクリプト言語(lua)で簡単な組み込みアプリも開発可能
 - 開発者向けのサイトも整備 →

引用元: <https://flashair-developers.com/ja/>



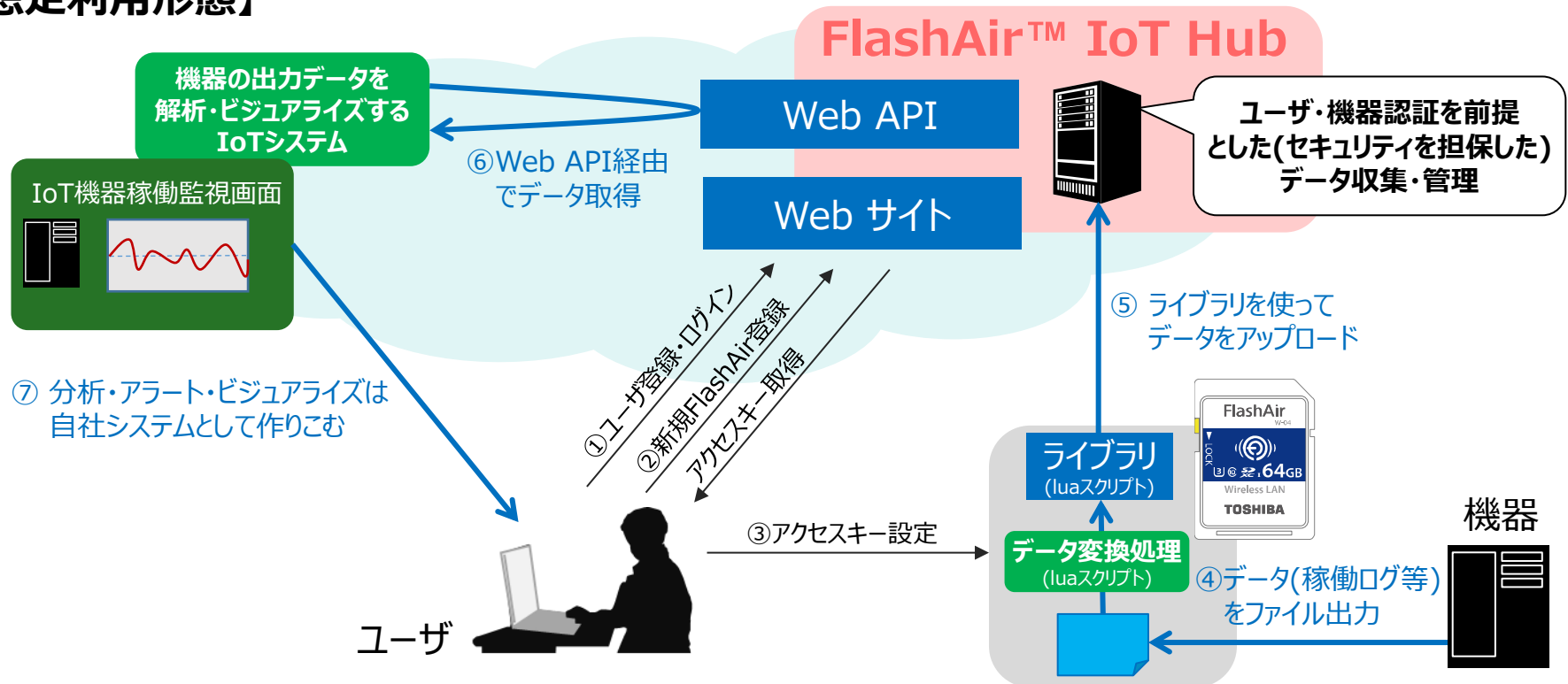
シンプルな例



FlashAir™ IoT Hub

- FlashAirを使ったIoTサービス開発をサポートする東芝のオンラインサービス
 - <https://iot-hub.flashair-developers.com>
 - FlashAir … 無線LAN搭載SDカード
 - SDカードスロットがあれば, レガシーな機器をIoT化できる

【想定利用形態】



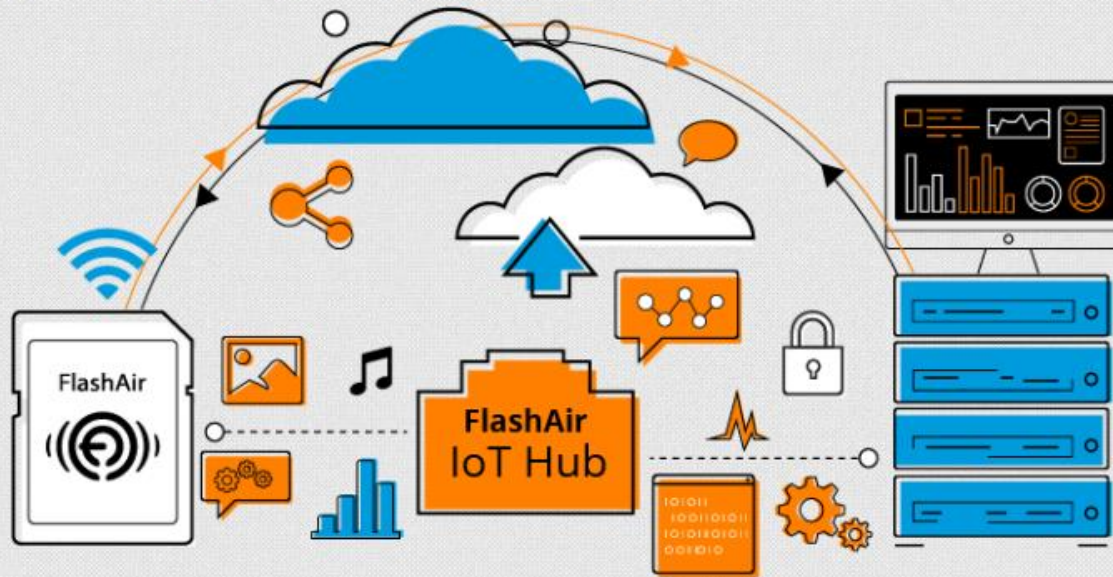
FlashAir™ IoT Hub

- FlashAirを使ったIoTサービス開発をサポートする東芝のオンラインサービス

【想定利

FlashAirとクラウドをつなぐ

FlashAir IoT Hubは、FlashAirを利用した自作のデバイス開発をサポートするウェブサービスです。
簡単な設定で、FlashAirをクラウドサービスや独自デバイスにつなぐことができます。



Japanese / English お問い合わせ

FlashAir™ IoT HubBeta

ホーム チュートリアル APIドキュメント

ログイン

メールアドレスで登録する方

メールアドレス

パスワード

パスワード(確認)

☐ 利用規約に同意する

登録する (無料)

ソーシャルアカウントで登録する方

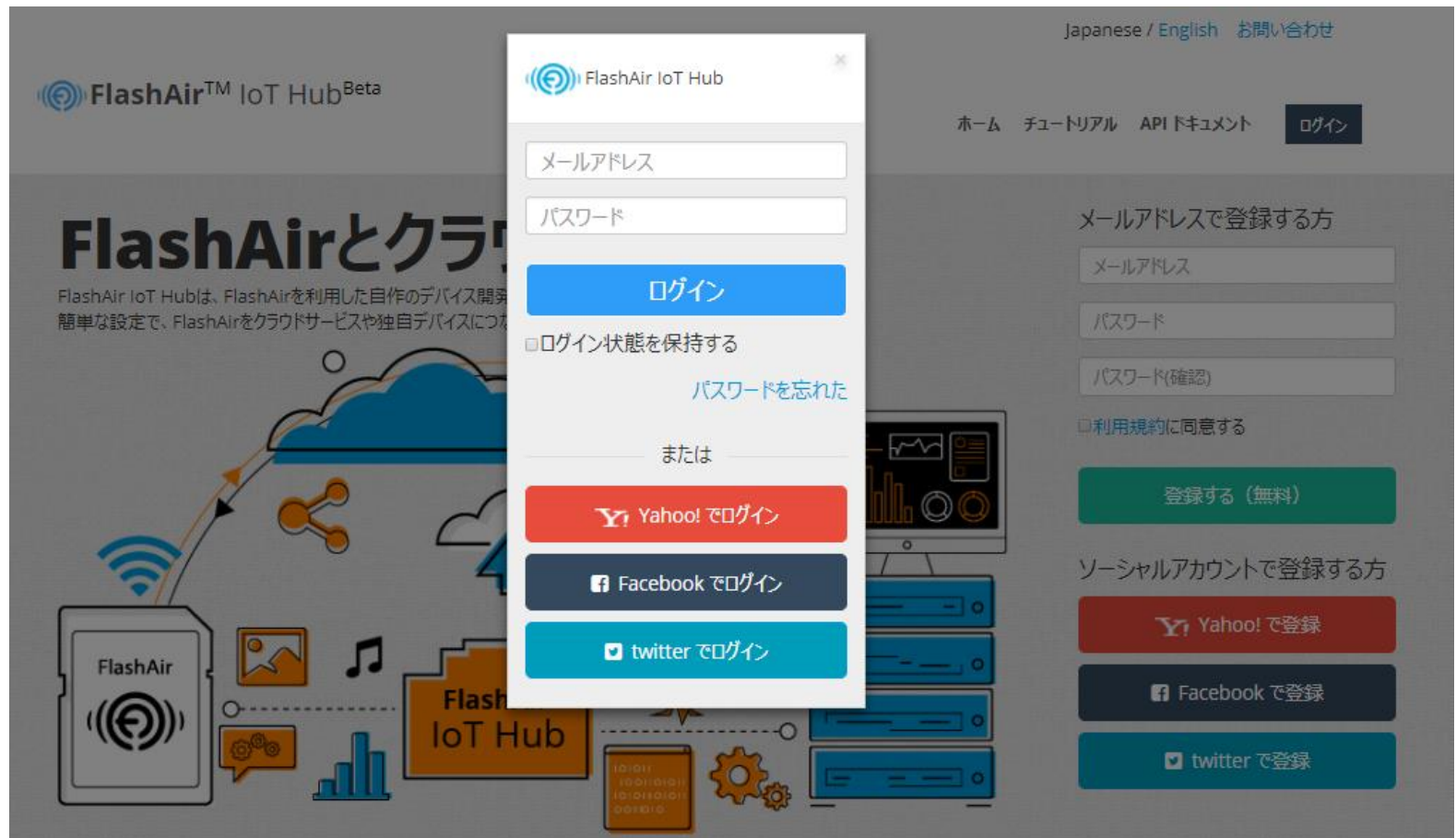
Y! Yahoo! で登録

f Facebook で登録

🐦 twitter で登録

FlashAir IoT Hub : Webサイト

- ユーザ登録・ログインすると、FlashAirの新規登録・管理が行える



FlashAir IoT Hub : Webサイト

- ユーザ登録・ログインすると、FlashAirの新規登録・管理が行える

The screenshot shows the FlashAir IoT Hub Beta website interface. At the top, there's a navigation bar with links for 'Japanese / English', 'お問い合わせ', '開発者画面', 'チュートリアル', 'API ドキュメント', 'FlashAirの追加登録', 'FlashAirの数: 5', and 'アカウント設定'. The main content area displays a FlashAir device card for 'FlashAir (2017/2/24 7:55:06)' with a dropdown menu for '名前の編集'. Below the card, it shows the ID '284594009127524260', a download button for 'アクセストークン取得', a download button for 'スクリプト', and a button for '登録削除'. The '最終アクセス: 無し' is also shown. Below this, there's a section titled '以下の手順をご確認の上、FlashAirの登録を完了してください。' (Please confirm the following steps and complete the FlashAir registration). It contains three numbered steps: 1. 'アクセストークンとLuaスクリプトを取得し、FlashAirに保存する' (Get the access token and Lua script and save them to FlashAir), 2. 'FlashAirを無線LANに接続する' (Connect FlashAir to wireless LAN), and 3. 'FlashAirの登録完了' (FlashAir registration complete). Each step has an icon and a detailed description. Step 1 includes instructions on downloading the access token and script to the PC and saving them to the FlashAir device. Step 2 includes instructions on connecting the FlashAir device to the wireless LAN and setting the script execution. Step 3 includes instructions on inserting the FlashAir device and confirming successful connection. Red arrows and text annotations highlight key actions: '①登録をクリックすると' (Clicking registration leads to...) points to the 'FlashAirの追加登録' link; '②アクセスキーが払い出される' (Access key is issued) points to the 'アクセストークン取得' button; and '③アクセスキーをFlashAirに書き込めば準備OK' (Writing the access key to FlashAir is preparation OK) points to the 'アクセストークン取得' button.

Japanese / English お問い合わせ
開発者画面

FlashAir™ IoT Hub Beta チュートリアル API ドキュメント FlashAirの追加登録 FlashAirの数: 5 アカウント設定

FlashAir (2017/2/24 7:55:06) 名前の編集

ID: 284594009127524260 アクセストークン取得 スクリプト 登録削除

最終アクセス: 無し

①登録をクリックすると

②アクセスキーが払い出される

③アクセスキーをFlashAirに書き込めば準備OK

以下の手順をご確認の上、FlashAirの登録を完了してください。

- 1 アクセストークンとLuaスクリプトを取得し、FlashAirに保存する
「アクセストークン取得」を選択してPC上にファイルをダウンロードします。
「スクリプト」を選択してPC上にファイルをダウンロード・展開します。
ダウンロード・展開したファイルを、FlashAirのルートディレクトリに書き込みます。
アクセストークン取得 スクリプト
- 2 FlashAirを無線LANに接続する
FlashAirをステーションモードに設定して無線LANに接続します。詳細はFlashAir Developersをご確認ください。
同時に、スクリプトの実行設定（LUA_RUN_SCRIPT=/bootscript.lua）と、GPIOを使用する場合はその設定（GPIO_MODE）を行います。
無線LANの設定方法
- 3 FlashAirの登録完了
FlashAirを抜き差しし、無線LAN接続に成功すれば登録が完了します。

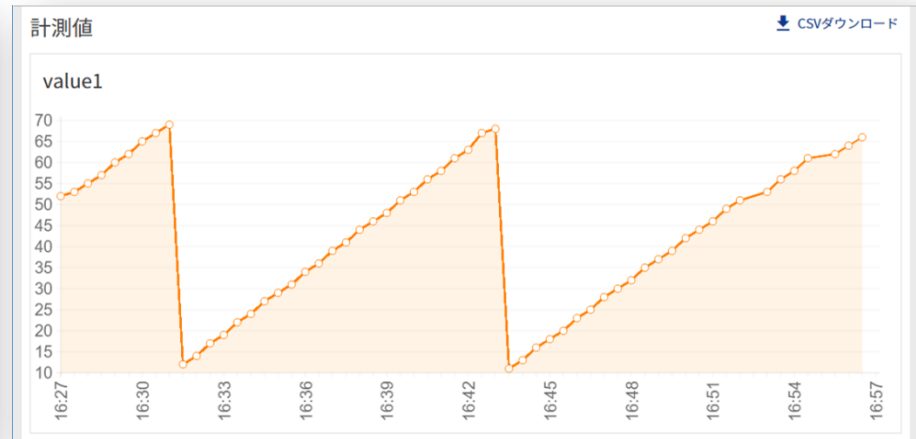
FlashAir IoT Hub : Webサイト

- Webサイト経由で現状できること
 - ユーザ登録・認証・管理
 - ユーザによるFlashAir登録・管理・更新(表示名の変更)・削除
 - **動作確認用簡易ユーティリティ (PIO制御、グラフ表示)**



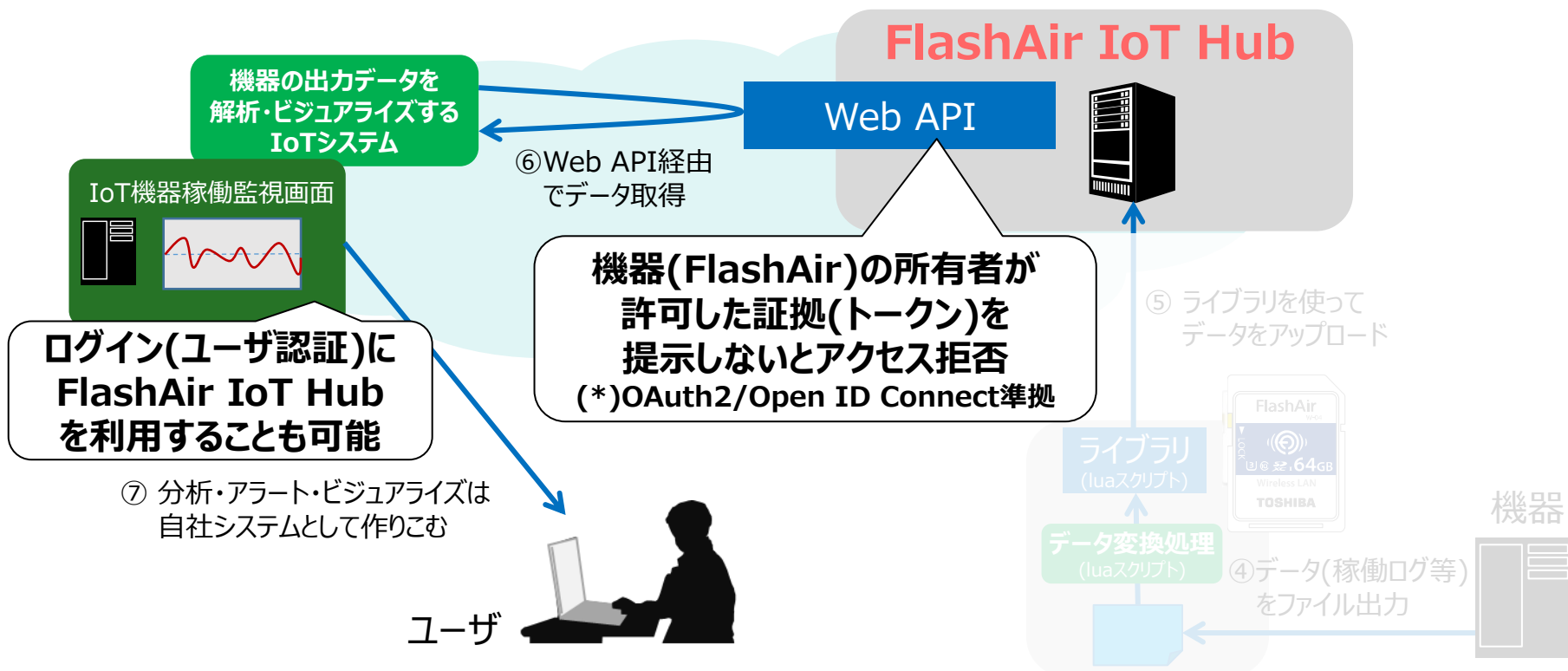
The screenshot shows the FlashAir IoT Hub web interface. At the top, there's a FlashAir logo and a dropdown menu showing 'FlashAir (2016-04-01T04:48:07.035Z)' with a '名前編集' (Edit Name) link. Below this, the ID '165456585236155844' is displayed, along with links for 'アクセストークン取得' (Get Access Token) and '登録削除' (Delete Registration). The main section is titled 'PIO' and contains a table with columns '名前' (Name) and 'High / Low設定' (High / Low Setting). The table lists five input pins (入力ピン1 to 入力ピン5) with their respective settings.

名前	High / Low設定
入力ピン1	High
入力ピン2	Low
入力ピン3	Low
入力ピン4	Low
入力ピン5	Low



FlashAir IoT Hub : Web API

- Web API
 - 外部プログラムが機能を利用できるように整備した通信プロトコル・データ形式
- FlashAir IoT Hubは、基本機能をWeb APIとして開発者に公開
 - Web APIは、Web標準の認証・認可方式(*)で保護



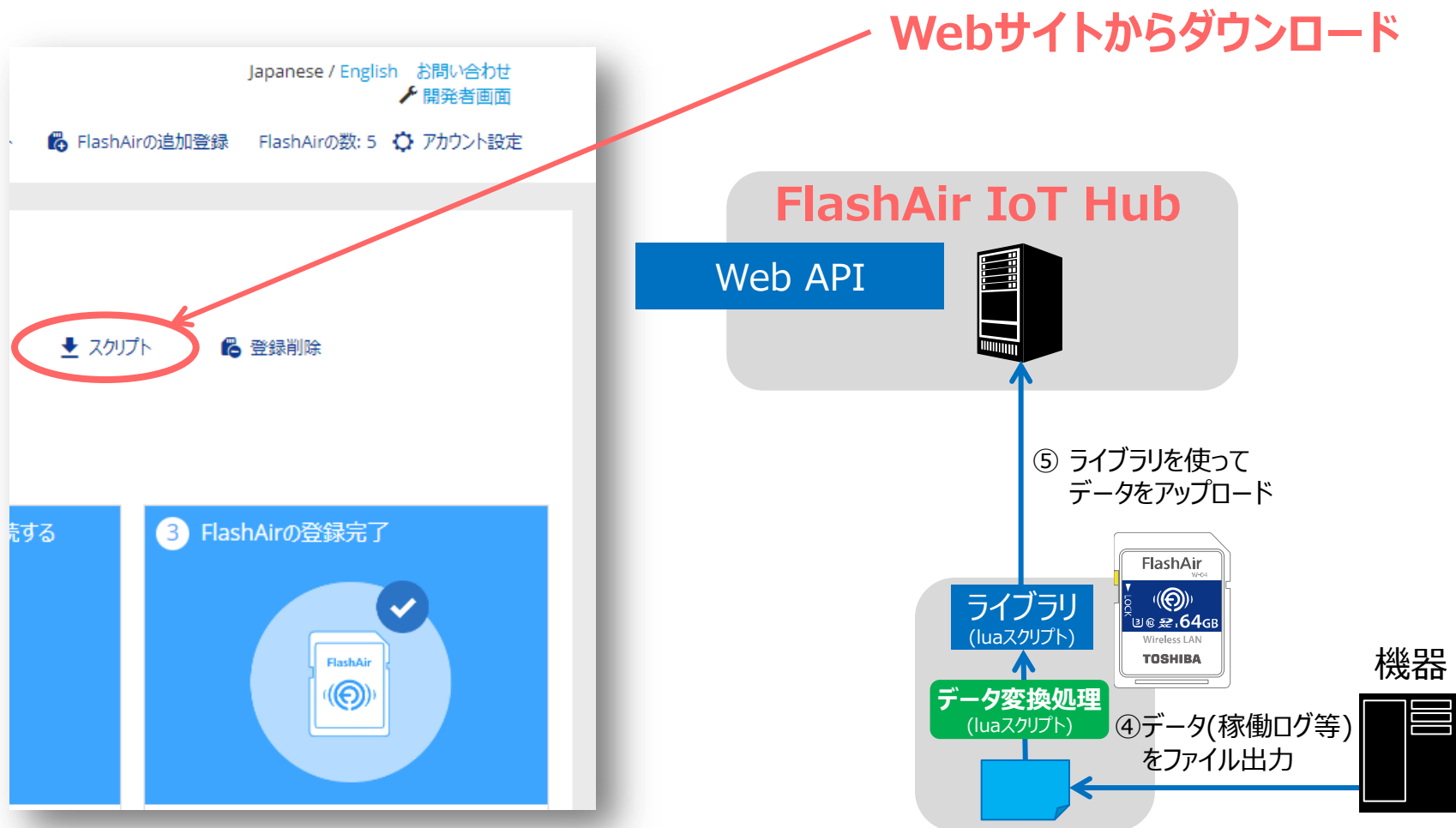
FlashAir IoT Hub : Web API

- 現在のAPIラインナップ

API	説明
Authorization API	OAuth2.0(RFC6749他)の認可サーバ機能を提供するAPI。
User API	ユーザ情報を扱うAPI。
FlashAir Device API	FlashAirのデバイス情報を扱うAPI。
Job API	FlashAirに対する遠隔監視・制御命令(ジョブ)を扱うAPI。
Measurement API	FlashAirが接続された計測データを扱うAPI。
PIO API	FlashAirのPIOのHigh/Low情報を取得するAPI。
Image API	FlashAirからアップロードされた画像データを扱うAPI。
Logging API	FlashAirからアップロードされたログを扱うAPI。

FlashAir IoT Hub との接続・通信用ライブラリ

- FlashAir側は、ライブラリを使うことで、FlashAir IoT HubのWeb APIを意識せずに開発可能



4. パッケージ提供について

クラウド基盤のご提供について

ここまでご紹介してきた技術を用いて、(株)東芝では、機器のIoT化に必要な共通機能を高効率なクラウドサービスと通信アダプタとして提供することで、IoT化に必要な様々なコストを低減する取り組みを進めています。

これらの技術をパッケージ化し、東芝情報システム(株)よりご提供させていただきます。

デモ展示もさせて頂きましたので、ご興味のある方は、是非、お問い合わせ願います。

パッケージ名称：「クラウドハブ（仮）」

上市予定：2017年8月

クラウドハブのご提供形態

◎ 組み込み機器ソフトウェアは、ソースコードを提供させていただきます。

- ① サーバー側中継ソフトの提供（実行モジュール）
- ② 組み込み機器側エージェントソフトの提供（C言語ソース）

⇒ **既存メンテナンス基盤へ適用可能**

*** カスタマイズも容易**

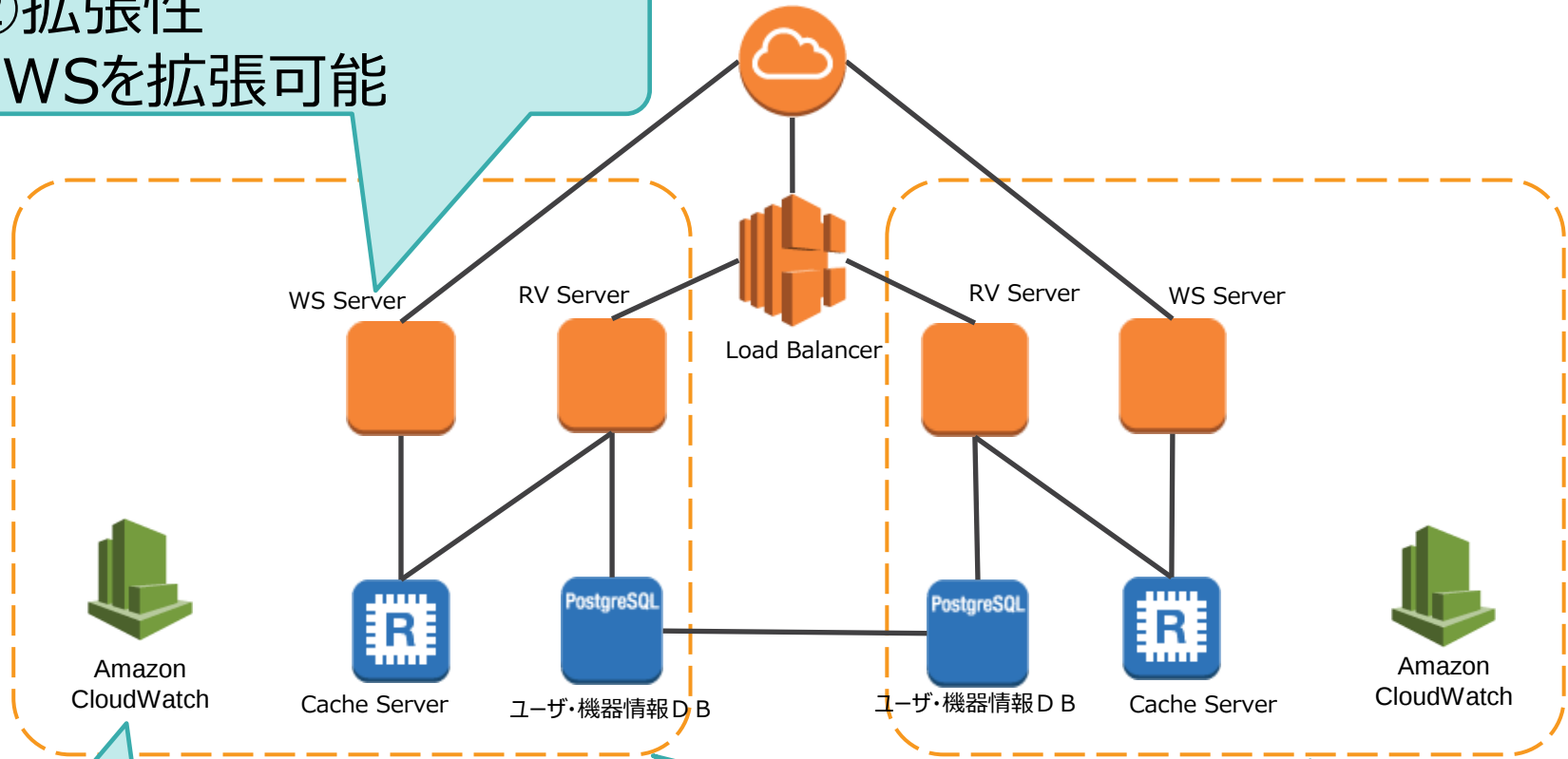
- ・AWS／Azure／オンプレ等 環境を選ばず適用可能
- ・既存認証機能との連携も可能

※本パッケージには以下サポートが用意されております。

- ① セキュリティパッチ版の提供
- ② Cloud Hubの使い方に関するメールによる問い合わせ対応

クラウドハブ適用例（AWS構成例）

②拡張性
WSを拡張可能



③監視
AWSのサービスを利用



①可用性
多重化により実現

TOSHIBA

Leading Innovation >>>