



SUMMIT
ONLINE
JAPAN

【セルフペースハンズオン】

AWS IoTの基礎

AWS IoT Core

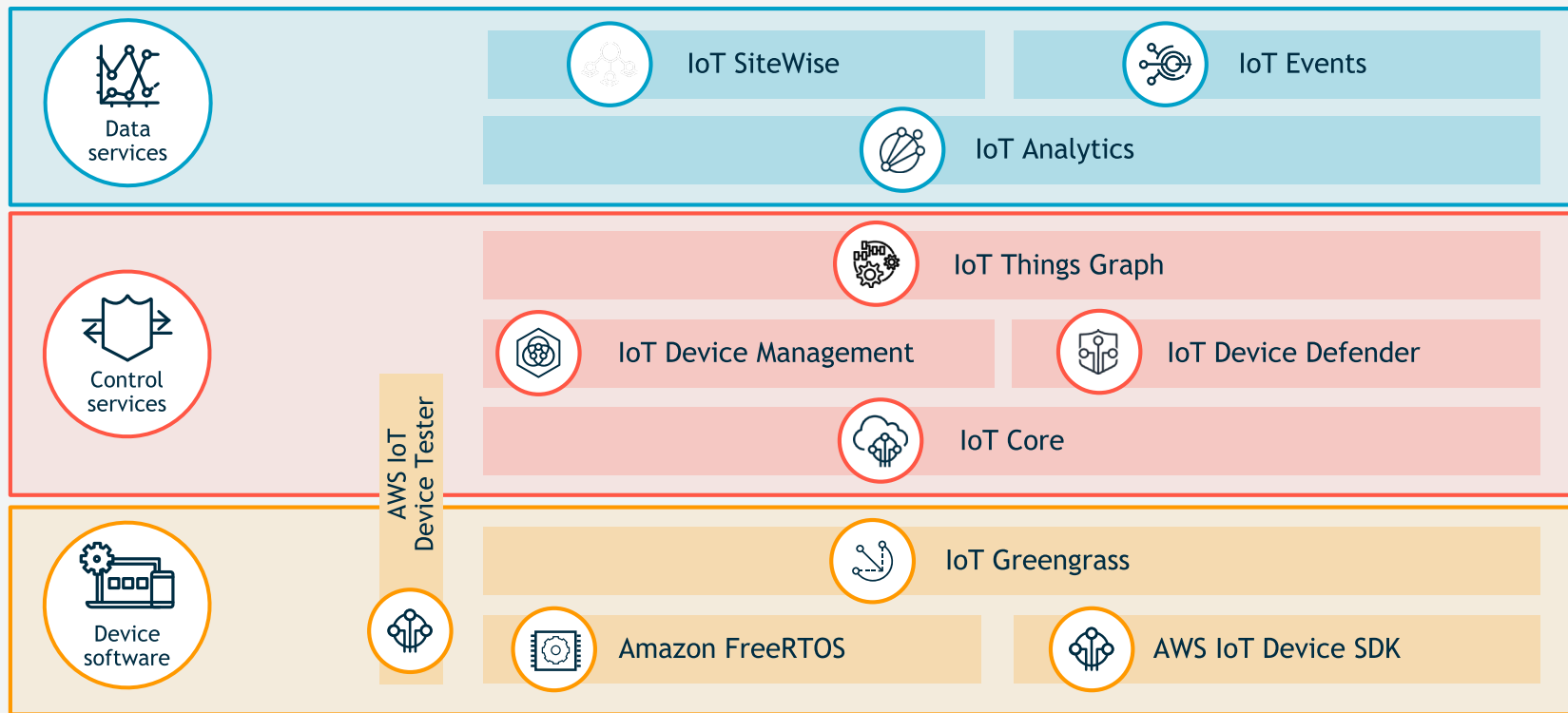
亀田 治伸

シニアアドボケイト

知っ得ハンズオンシリーズ

はじめてのIoT AWS IoT Core

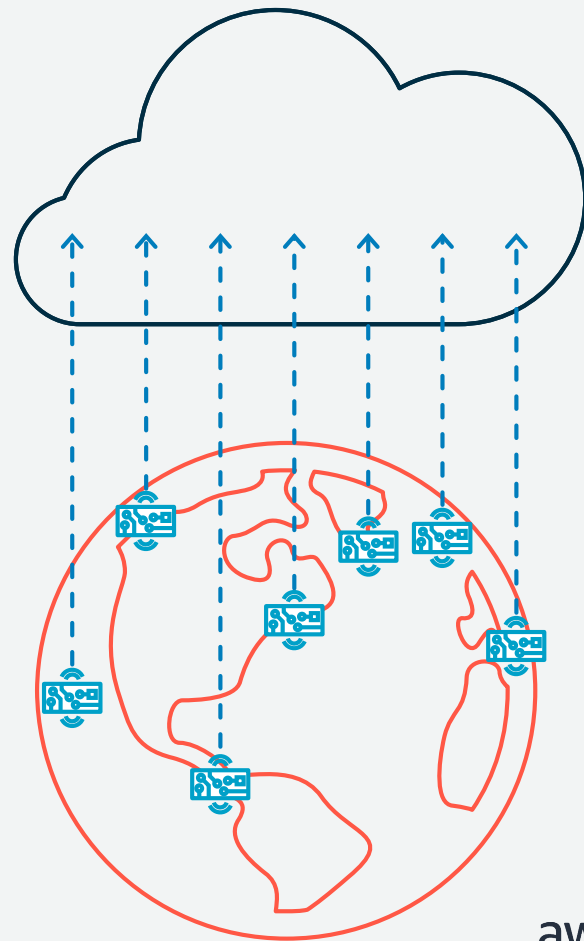
AWS IoT Architecture (Today)



AWS IoT Core



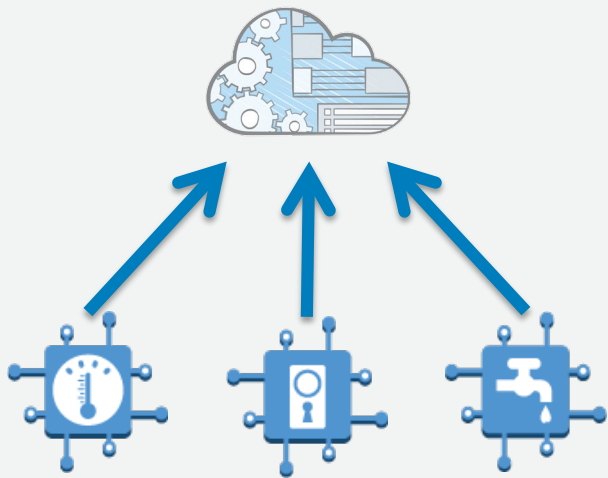
Control
services



AWS IoT : 2つの利用用途

データ収集

数十万規模のデバイスからの
データ収集

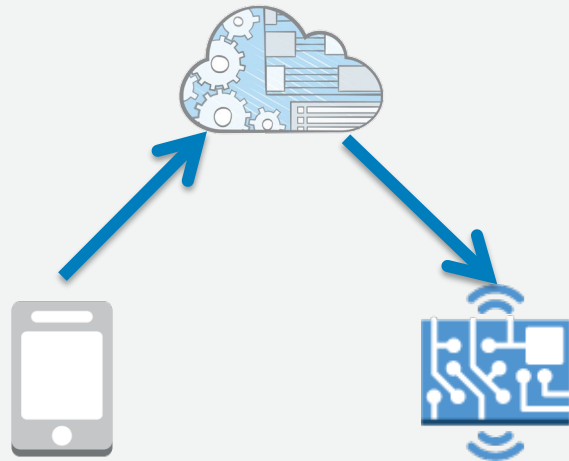


使用例

デバイスのデータの可視化
故障予測・異常検知

リモート制御

遠隔にあるデバイスをクラウドを
介してコントロール



ファームアップ^o
リモート機器制御



AWS IoT と Amazon Kinesis



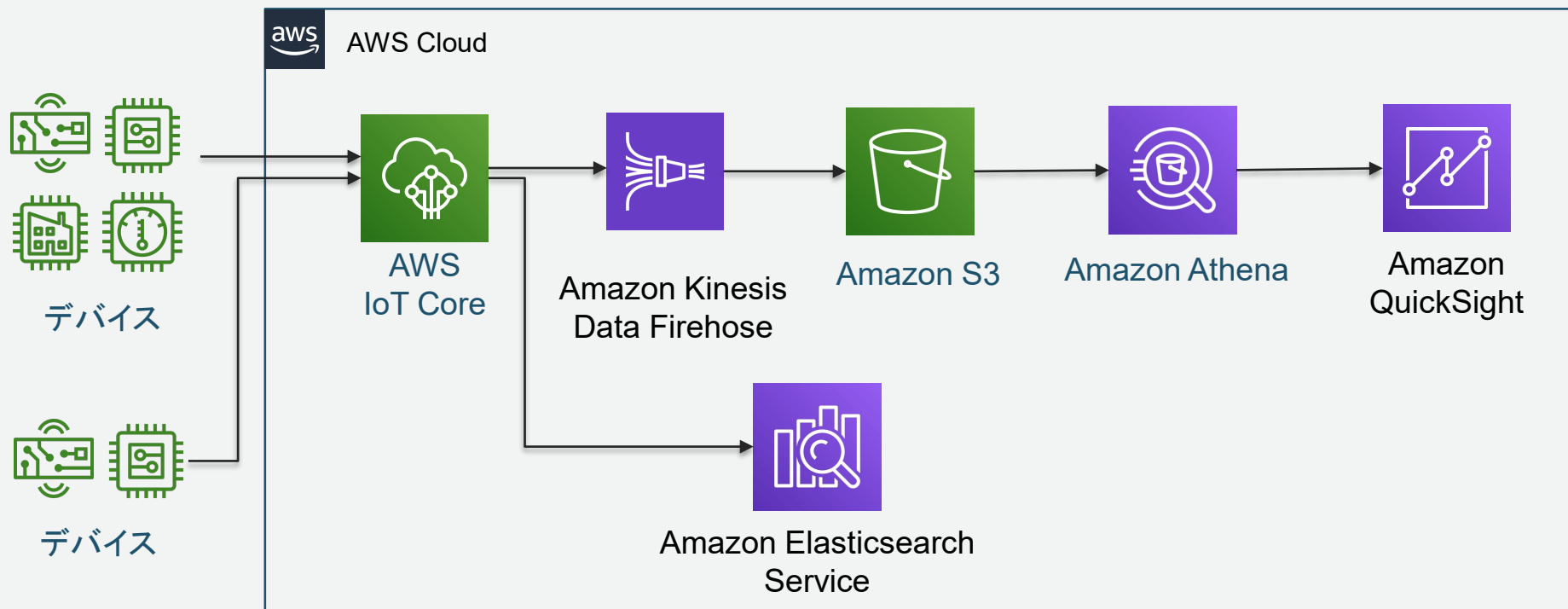
HTTP,WebSocket,MQTT
クライアント証明書認証
双方向制御



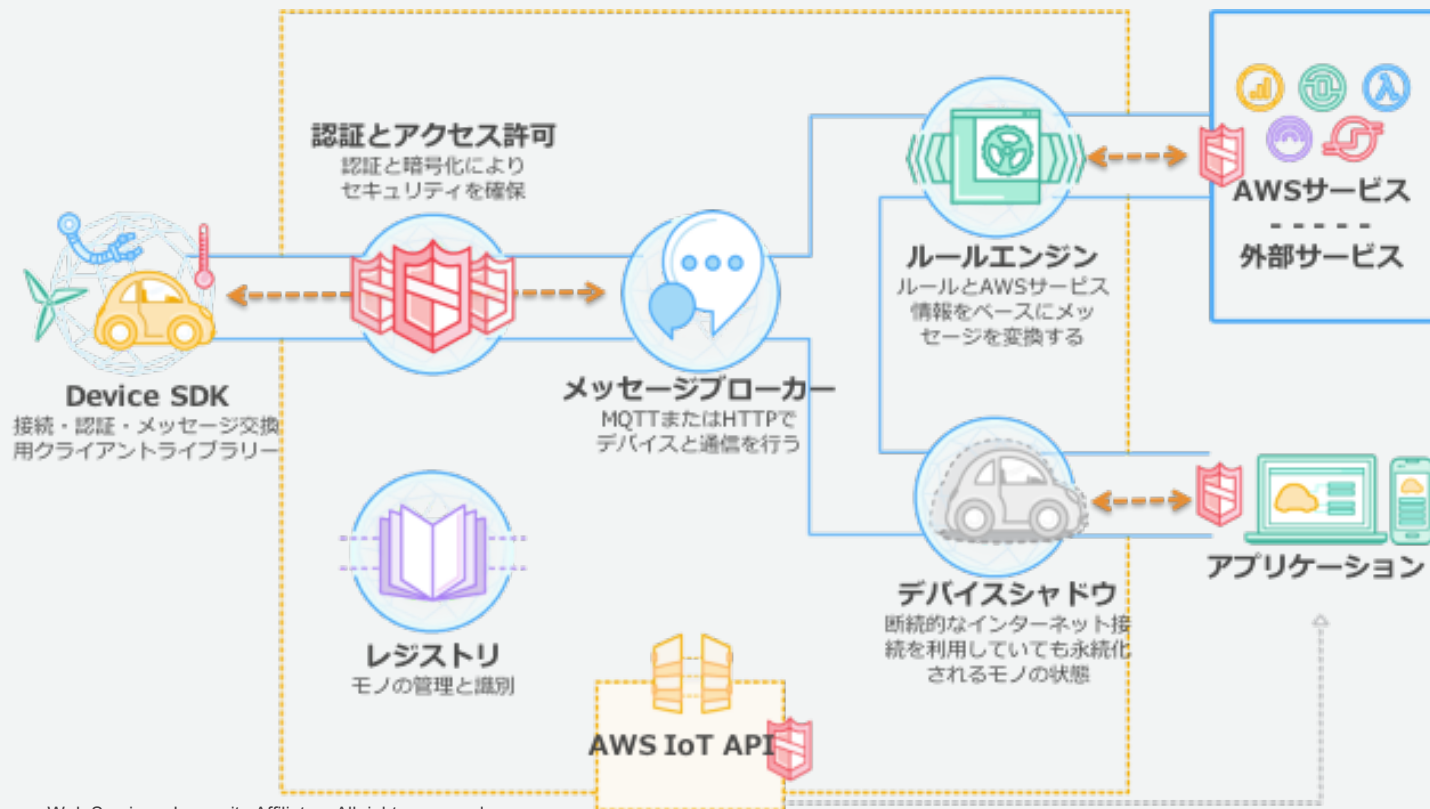
HTTP
要専用ライブラリ
データストリーム

約 10 倍の価格差

Question ?



AWS IoT Core



MQTT QoS

QoS=0 At most once



ベストエフォート型。
メッセージの到達を保証していない。
オーバーヘッドは少ない。

QoS=1 at least once



保証型。
メッセージの到達は保証。
オーバーヘッドはQoS=0に比べて大きい。

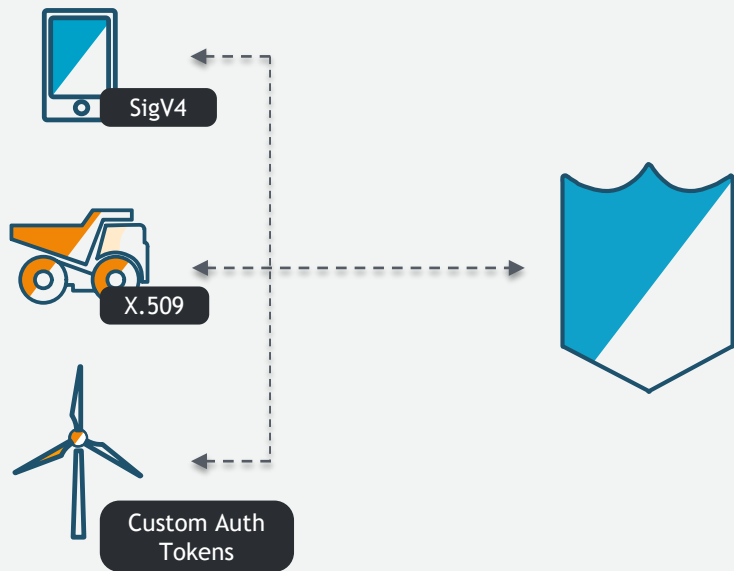
Identity Service

証明書によるデバイスごとの認証

独自もしくは AWS IoT が発行する
X.509証明書による認証が可能。他にも
SigV4 やカスタム認証トークン、
Cognitoをサポート

ポリシーによるデバイスごとの認可

デバイスごとに MQTT の Topic レベル
でアクセスコントロールが可能



ルールエンジン

シンプル&慣れた構文でフィルタ

SQL文を使って、トピックのフィルタや、オプションの WHERE 句で JSON ペイロードに対する条件を記述することが可能

データ変換、拡張

組み込み関数による文字列操作、算術計算、UUID や Timestamp、シャドウのコンテキストの付与が可能

ルーティング

10を超える AWS サービスや 3rd party のサービスにルーティング可能



Analytics
Kinesis



Artificial
Intelligence
EMR



Messaging
SQS
SNS



Database
Redshift
DynamoDB

















Manage
CloudWatch



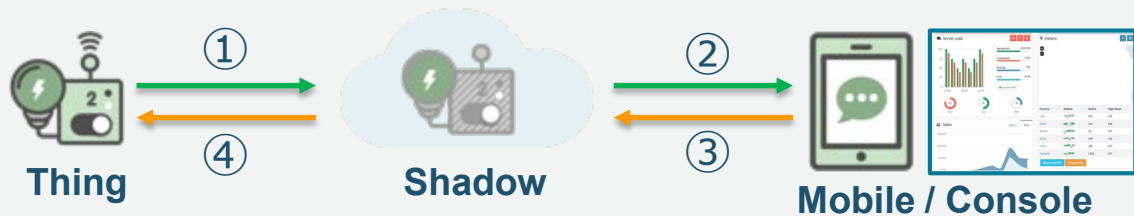
```
SELECT * FROM 'things/thing-2/color'  
WHERE color = 'red'
```



AWS IoT Core ルールエンジン

- | | |
|--|---|
| <input type="radio"/>  DynamoDB テーブルにメッセージを挿入する
DYNAMODB | <input type="radio"/>  Amazon Kinesis Firehose ストリームにメッセージを送信する
AMAZON KINESIS FIREHOSE |
| <input type="radio"/>  データベーステーブル (DynamoDBv2) の複数列にメッセージを分割する
DYNAMODB V2 | <input type="radio"/>  CloudWatch にメッセージデータを送信する
CLOUDWATCH メトリクス |
| <input type="radio"/>  メッセージデータを渡す Lambda 関数を呼び出す
LAMBDA | <input type="radio"/>  CloudWatch アラームの状態を変更する
CLOUDWATCH アラーム |
| <input type="radio"/>  SNS プッシュ通知としてメッセージを送信する
SNS | <input type="radio"/>  Amazon Elasticsearch Service にメッセージを送信する
AMAZON ELASTICSEARCH |
| <input type="radio"/>  SQS キューにメッセージを送信する
SQS | <input type="radio"/>  Salesforce の IoT 入力ストリームへのメッセージの送信
SALESFORCE の IOT |
| <input type="radio"/>  Amazon Kinesis ストリームにメッセージを送信する
AMAZON KINESIS | <input type="radio"/>  IoT Analytics チャネルにメッセージを送信
IOT ANALYTICS |
| <input type="radio"/>  AWS IoT のトピックにメッセージを再パブリッシュする
AWS IOT の再パブリッシュ | <input type="radio"/>  Step Functions ステートマシンの実行を開始する
ステップ関数 |
| <input type="radio"/>  Amazon S3 バケットにメッセージを格納する
S3 | |

デバイスシャドウ



```
{
  "state": {
    "desired": {
      "color": "RED",
      "engine": "ON"
    },
    "reported": {
      "color": "BLUE",
      "engine": "ON"
    },
    "delta": {
      "color": "RED"
    }
  },
  "version": 10
}
```

デバイスの状態の取得と保存に使用する JSON ドキュメント

1. Thing は自身の状態を reported としてシャドウに登録
2. Mobile は Thing の状態をシャドウを参照することで取得
3. Mobile は Thing の状態の更新を desired としてシャドウに登録
4. Thing は自身の状態 (reported) と要求されている状態 (desired) の差分 (delta) の通知を受け取り、要求されている状態に自身を変更する。1 に戻って、変更した状態を通知する

Cloud9

AWSが提供する統合開発環境（IDE）
クラウドで稼働し Pay – per – Use
Lambda関数の直接デプロイが可能

EC2 / 外部サーバ（SSH） 両方で実行可能

Thank you!