

IOT301, IOT308, AUT302

コネクティッド・ビークル最前線 大規模展開とEV要件

眞壽田 英輝 (Masuta, Hideki)

ソリューションアーキテクト Mobility 領域担当
アマゾンウェブサービスジャパン合同会社



自己紹介

眞壽田 英輝 (Masuta, Hideki)



アマゾンウェブサービスジャパン合同会社
ソリューションアーキテクト Mobility 領域担当

経歴

- 24年間ソフトウェア開発者

直近の経歴

- 自動車OEM様向け商用テレマティクスConnected Platform設計・開発・運用を3社担当
- インド都市向けITS・渋滞情報システム設計・開発・運用を担当
- オートバレーパーキングPoCにて経路生成MEC開発を担当

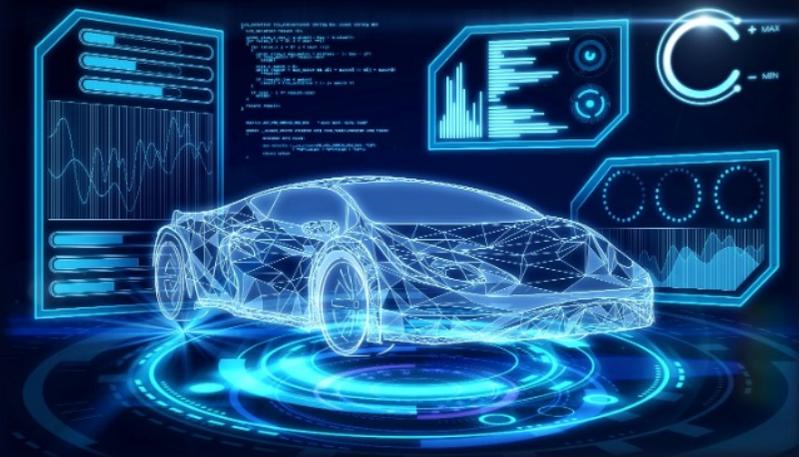
Agenda

- Connected Vehicleの課題
- Connected Vehicleに求められるソリューション
 - AWS Private CAを活用したTLS通信
 - 証明書の失効検知・有効期限の更新
 - AWS IoT Core MQTT 5サポート
 - MQTT5 活用ユースケース1 (Request/Response)
 - MQTT5 活用ユースケース2 (Message Expiry)
 - MQTT5 活用ユースケース3 (User Properties)
- 次世代Connected Vehicleに求められる要件
- AWS IoT Fleetwise概要
- Hands On紹介 AWS IoT FleetWise による EV バッテリーモニタリングソリューションの構築

Connected Vehicleの課題



接続数の増加・大容量データの収集と管理



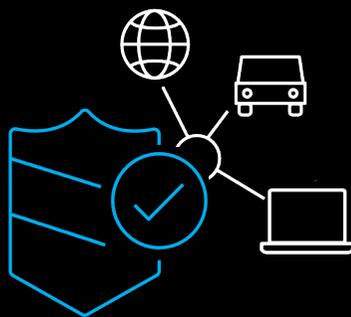
堅牢性・セキュリティ



車載ソフトウェアのテスト・解析

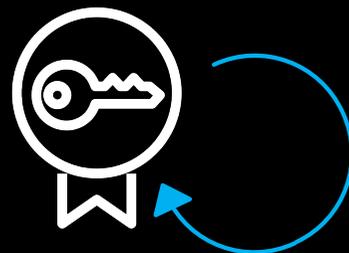
Connected Vehicleに求められるソリューション

CONNECT, MANAGE, AND SCALE YOUR DEVICE FLEET EASILY AND RELIABLY



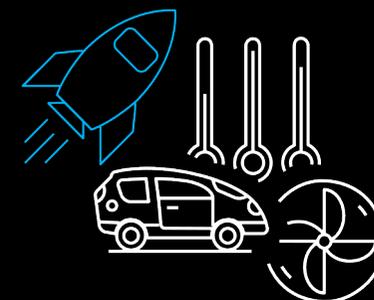
Connected vehicles device management

証明書管理



Connected vehicle device defender

証明書のリボケーション・
有効期限の更新

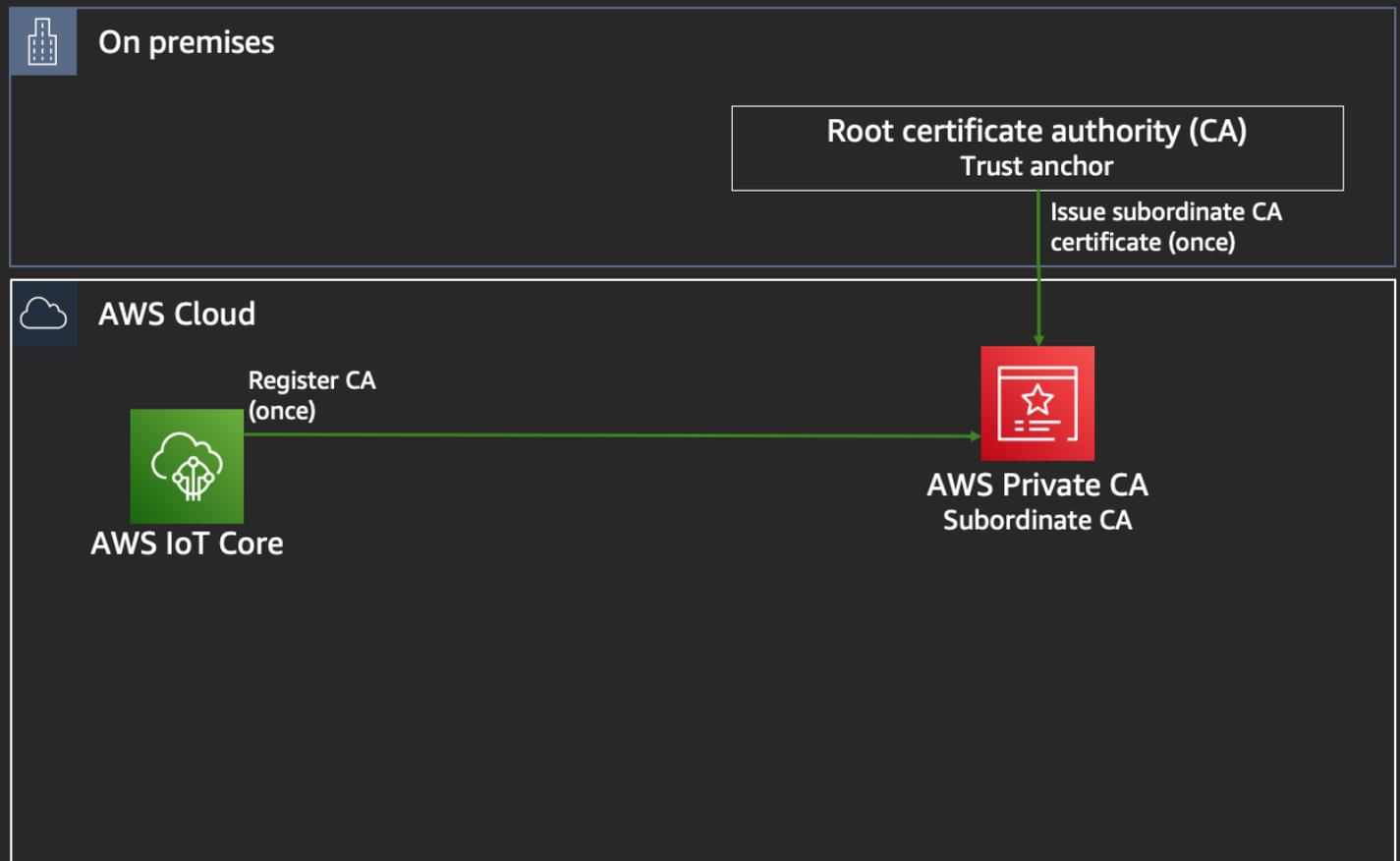
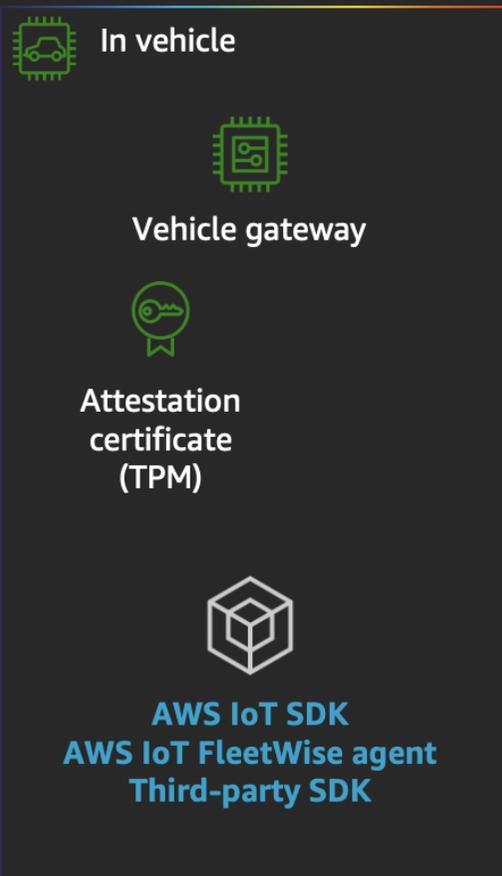


AWS IoT Core launches native MQTT 5 support

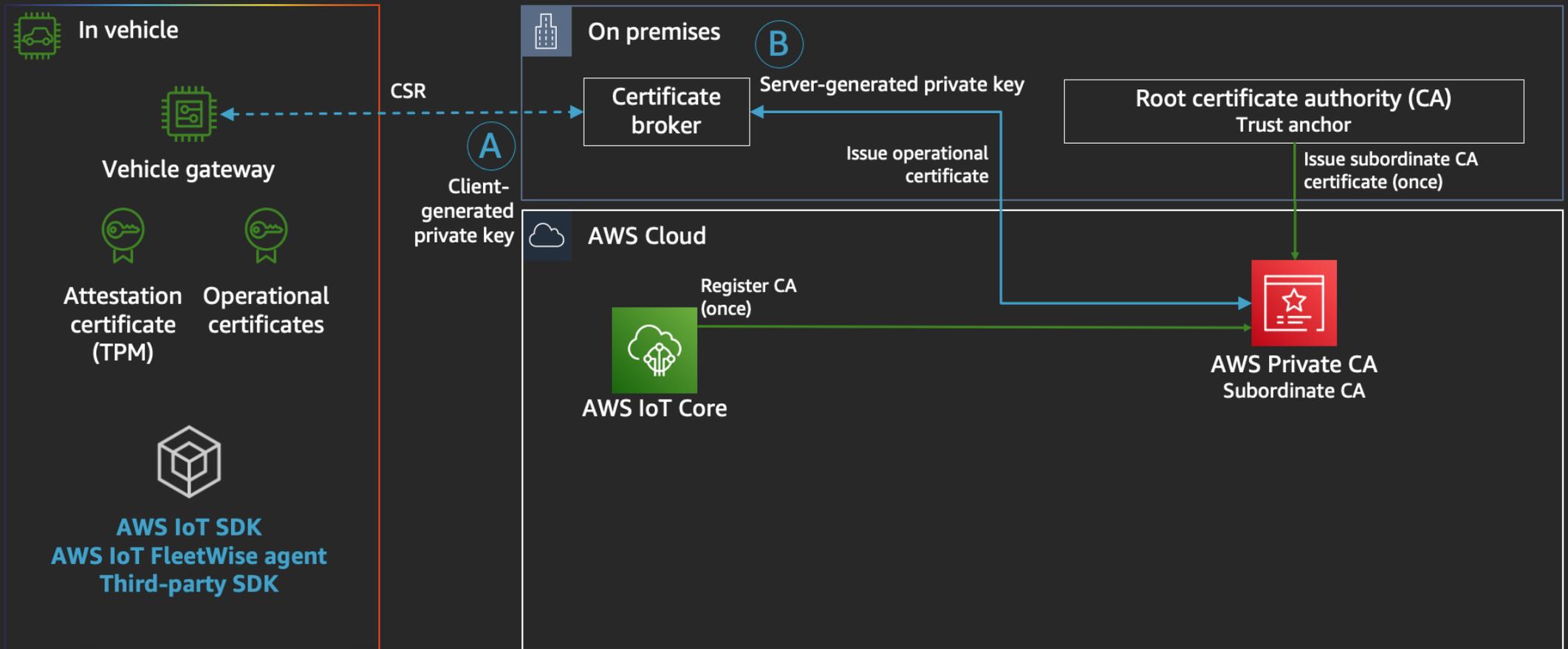
大規模な車両接続
MQTT 5のサポート

New

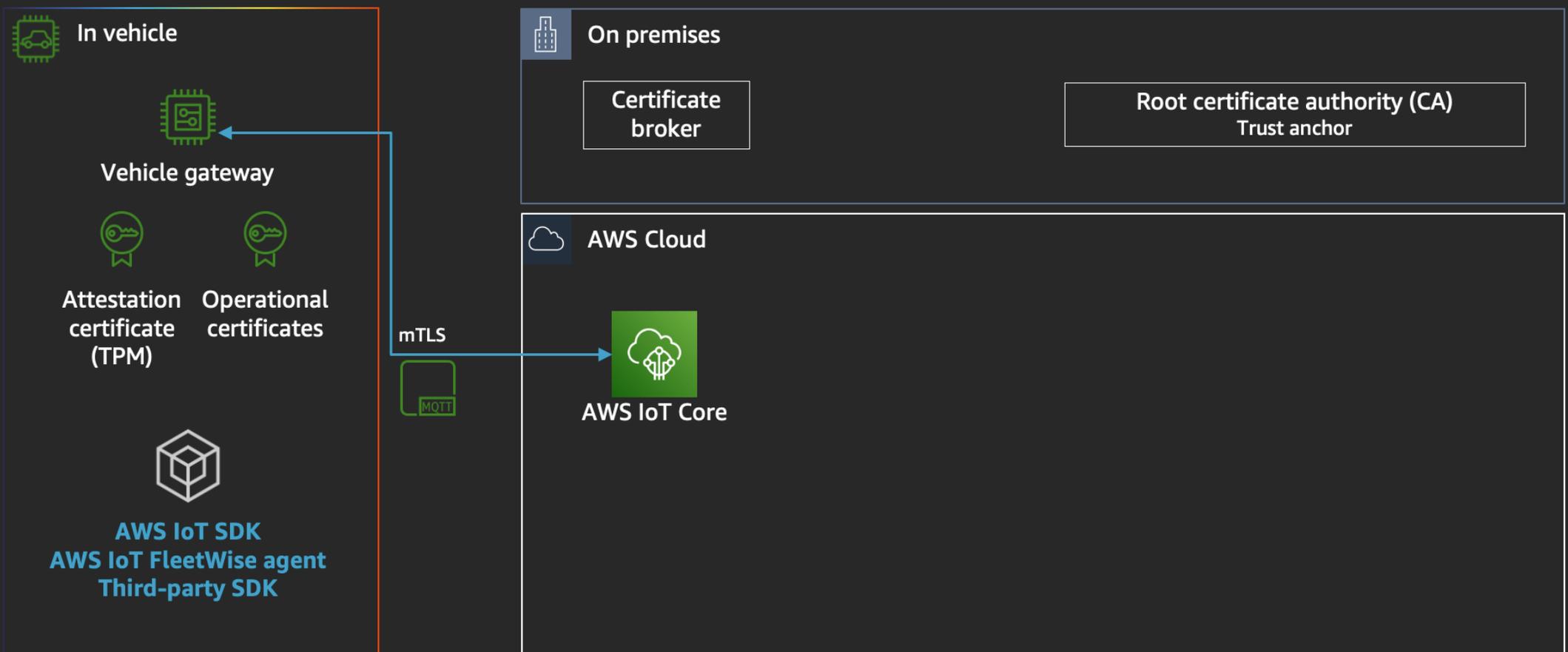
Best practice: AWS Private CAを活用したTLS通信



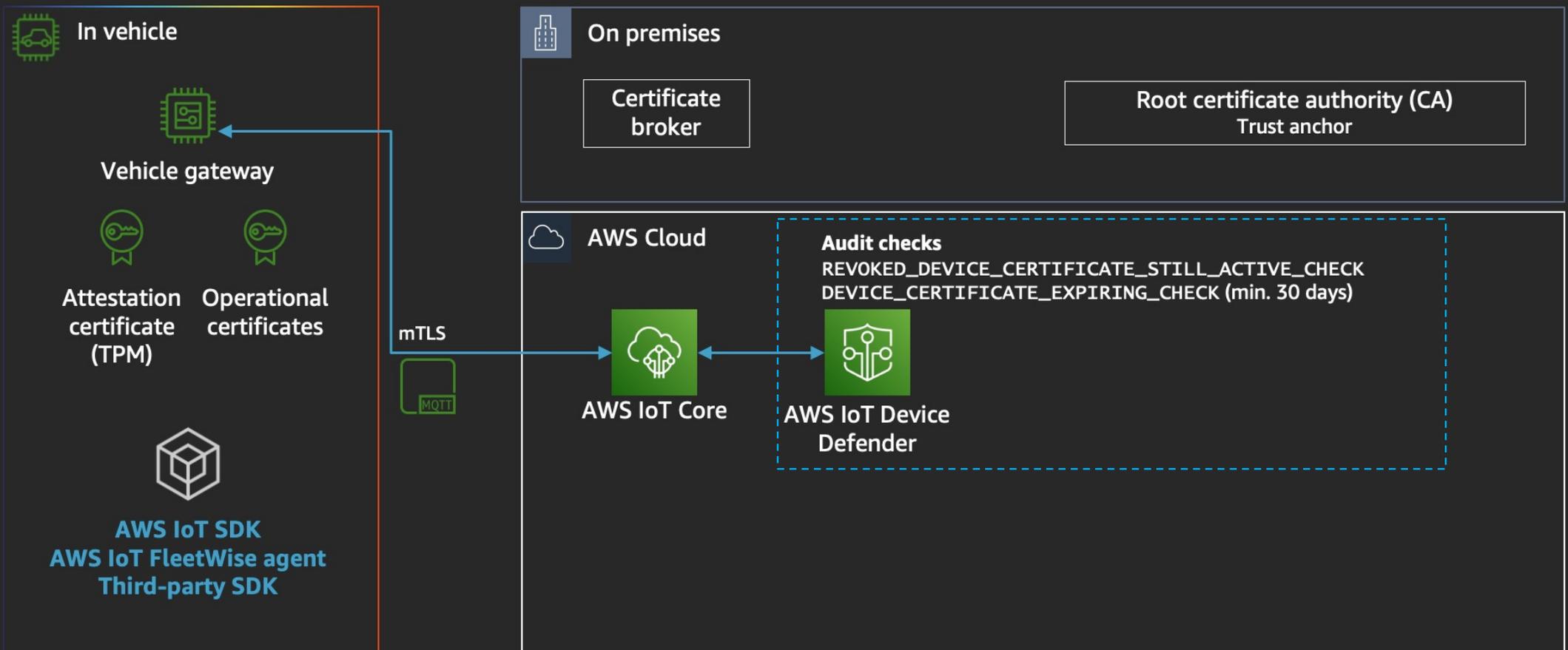
Best practice: AWS Private CAを活用したTLS通信



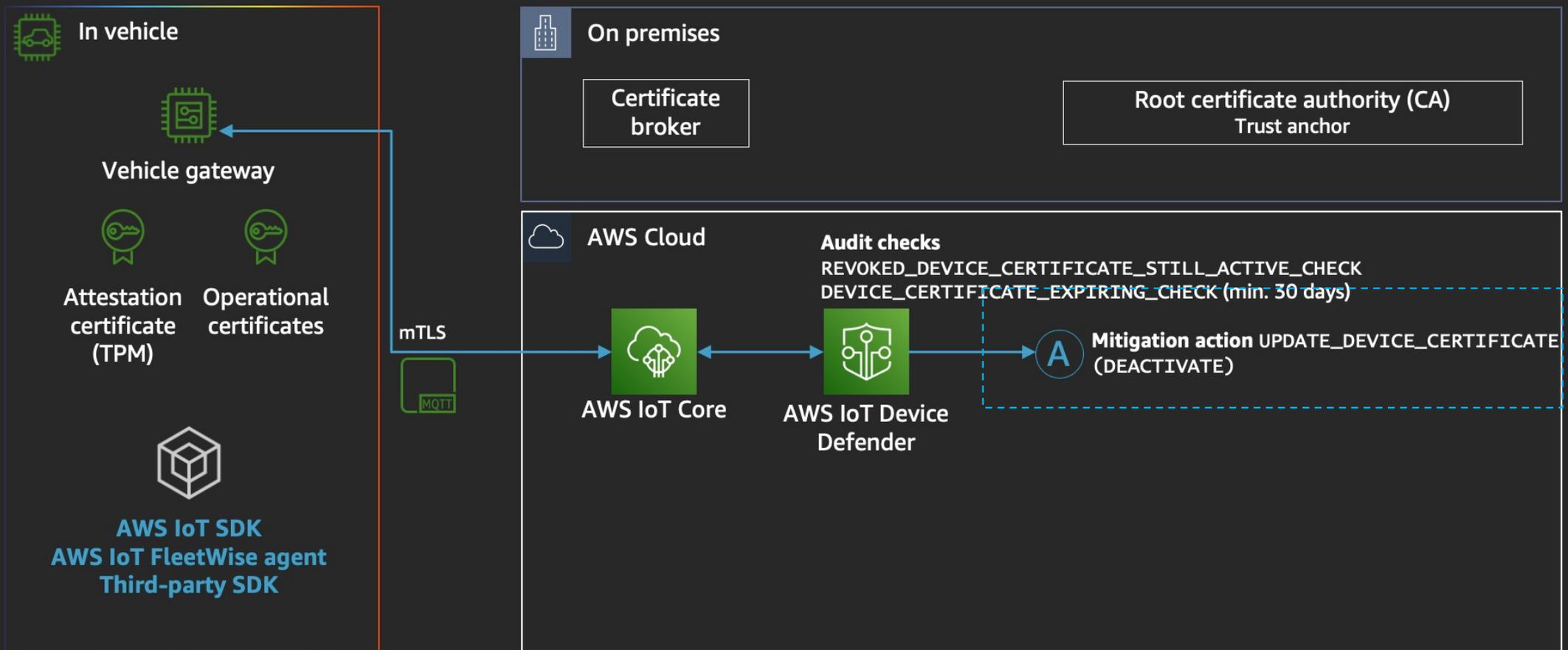
Best practice: 証明書の失効検知・有効期限の更新



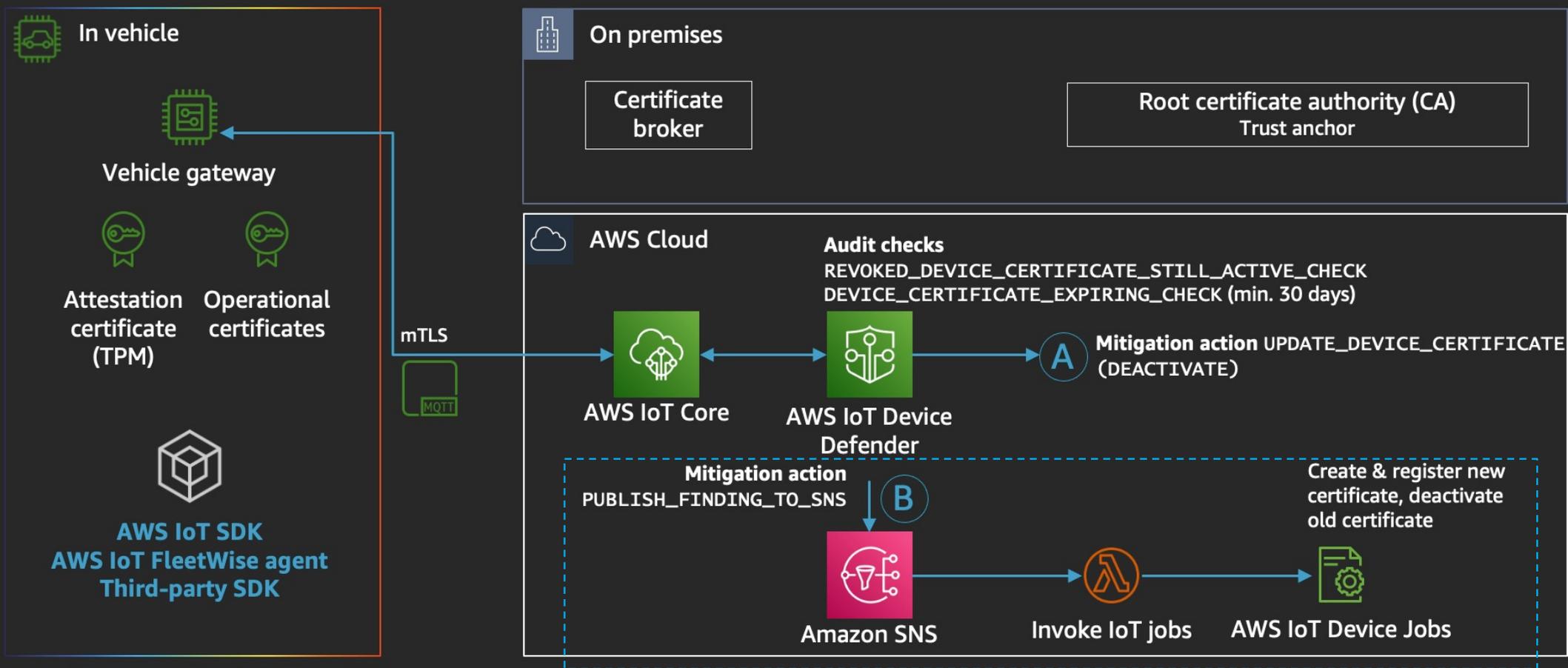
Best practice: 証明書の失効検知・有効期限の更新



Best practice: 証明書の失効検知・有効期限の更新



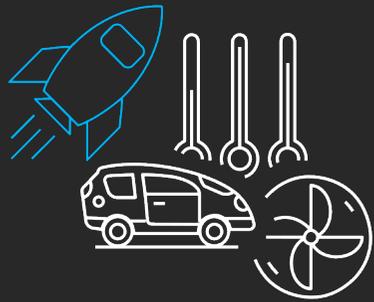
Best practice: 証明書の失効検知・有効期限の更新



New

AWS IoT Core MQTT 5サポート

AWS IoT CoreのMQTT 5.0で利用可能な機能

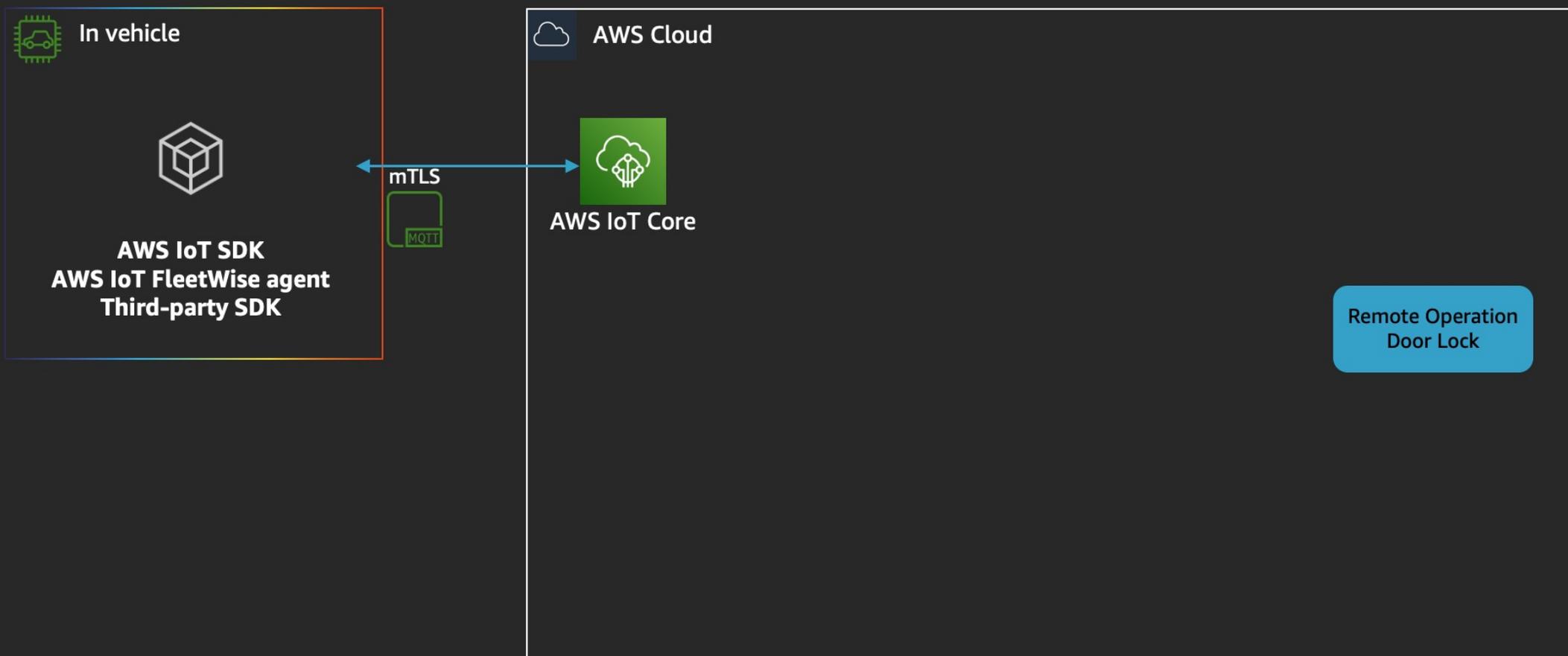


**AWS IoT Core launches
native MQTT 5 support**

- ◆ Shared Subscription (2023年1月13日の時点で未対応)
- ◆ Reason Codes
- ◆ Topic Alias
- ◆ Variable Header
- ◆ User Properties
- ◆ Request / Response
- ◆ Clean Start
- ◆ Session Expiry
- ◆ Message Expiry

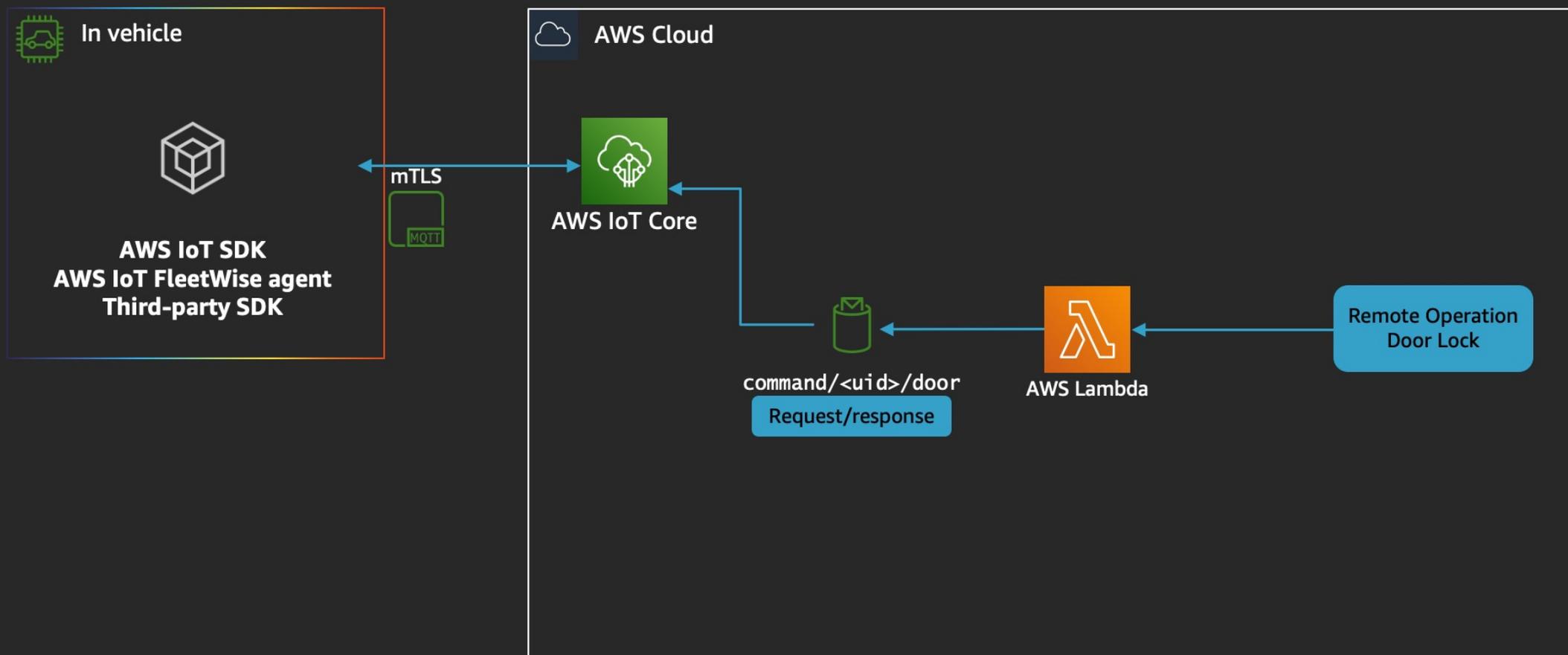
MQTT5 活用ユースケース1 (Request/Response)

- ユーザー側から送信したリモートコマンドが成功したかどうかを知りたい



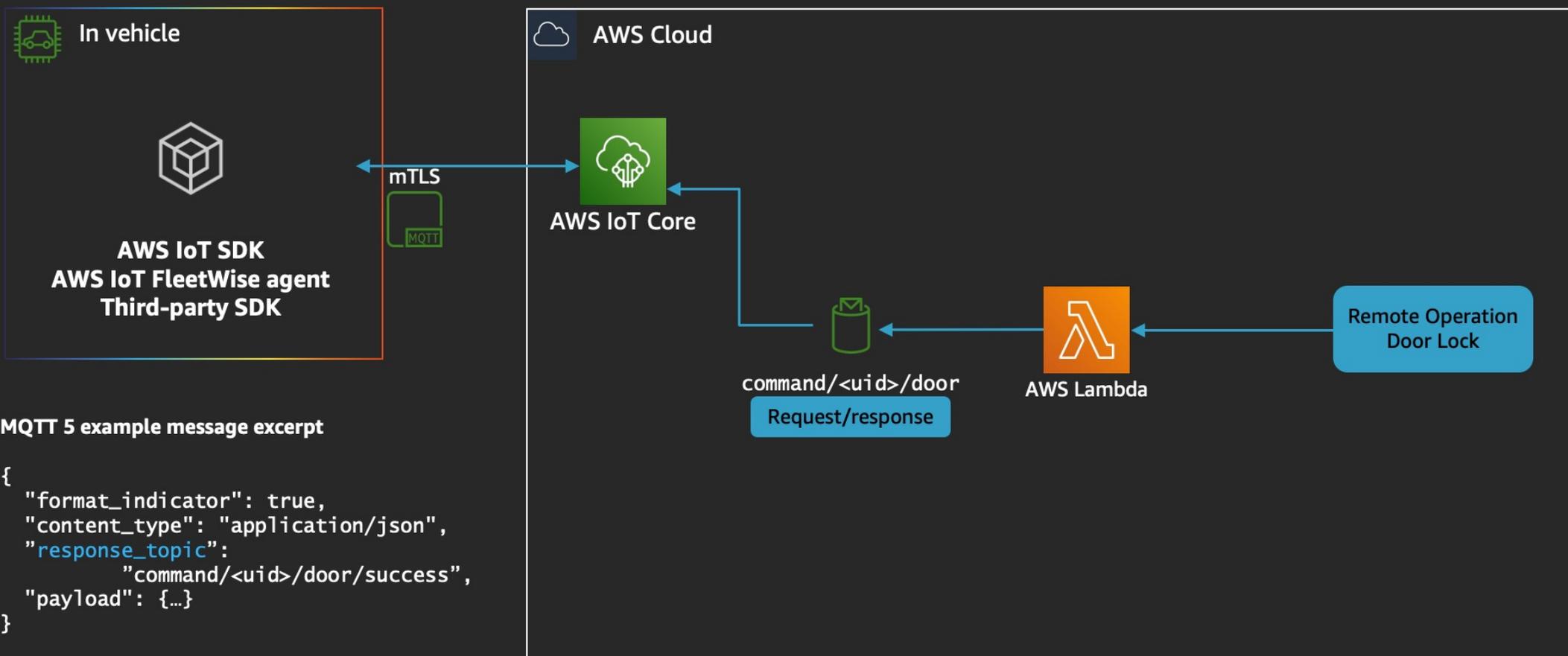
MQTT5 活用ユースケース1 (Request/Response)

- ユーザー側から送信したリモートコマンドが成功したかどうかを知りたい



MQTT5 活用ユースケース1 (Request/Response)

- ユーザー側から送信したリモートコマンドが成功したかどうかを知りたい

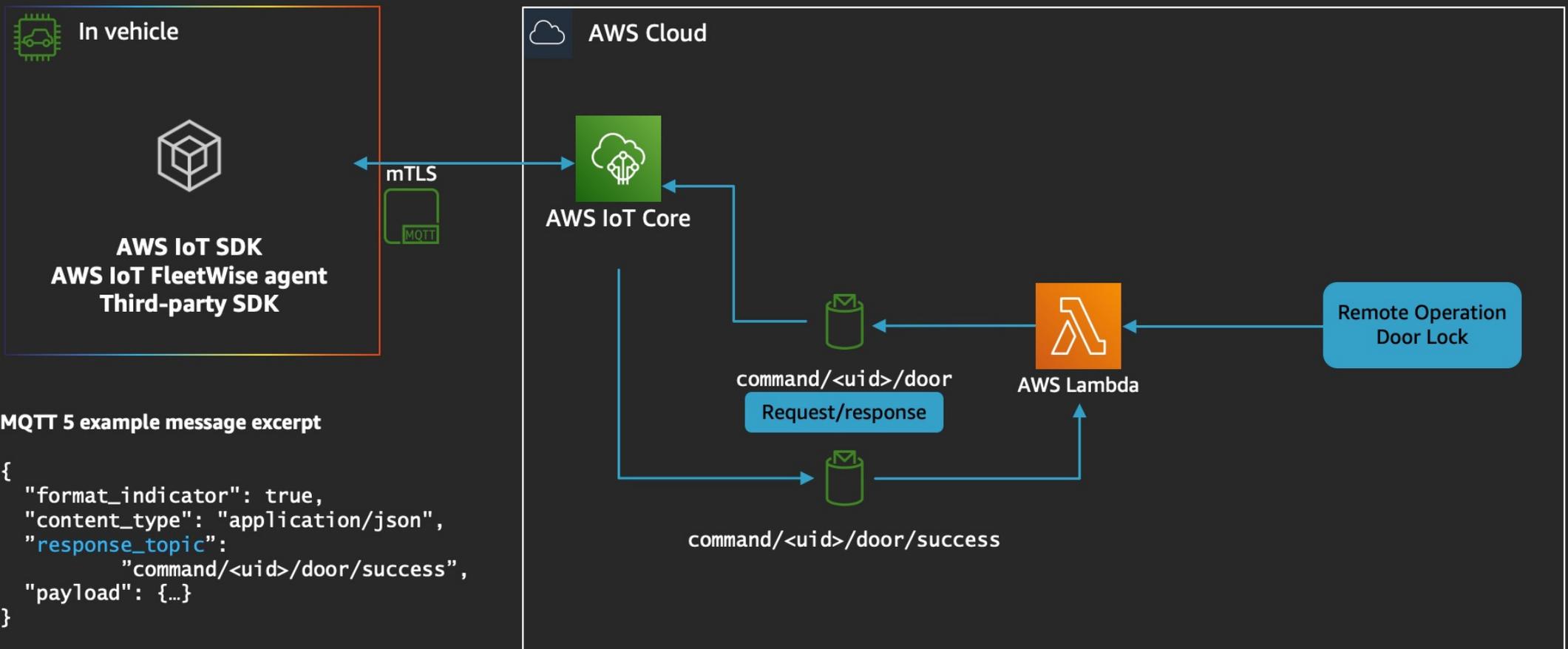


MQTT 5 example message excerpt

```
{  
  "format_indicator": true,  
  "content_type": "application/json",  
  "response_topic":  
    "command/<uid>/door/success",  
  "payload": {...}  
}
```

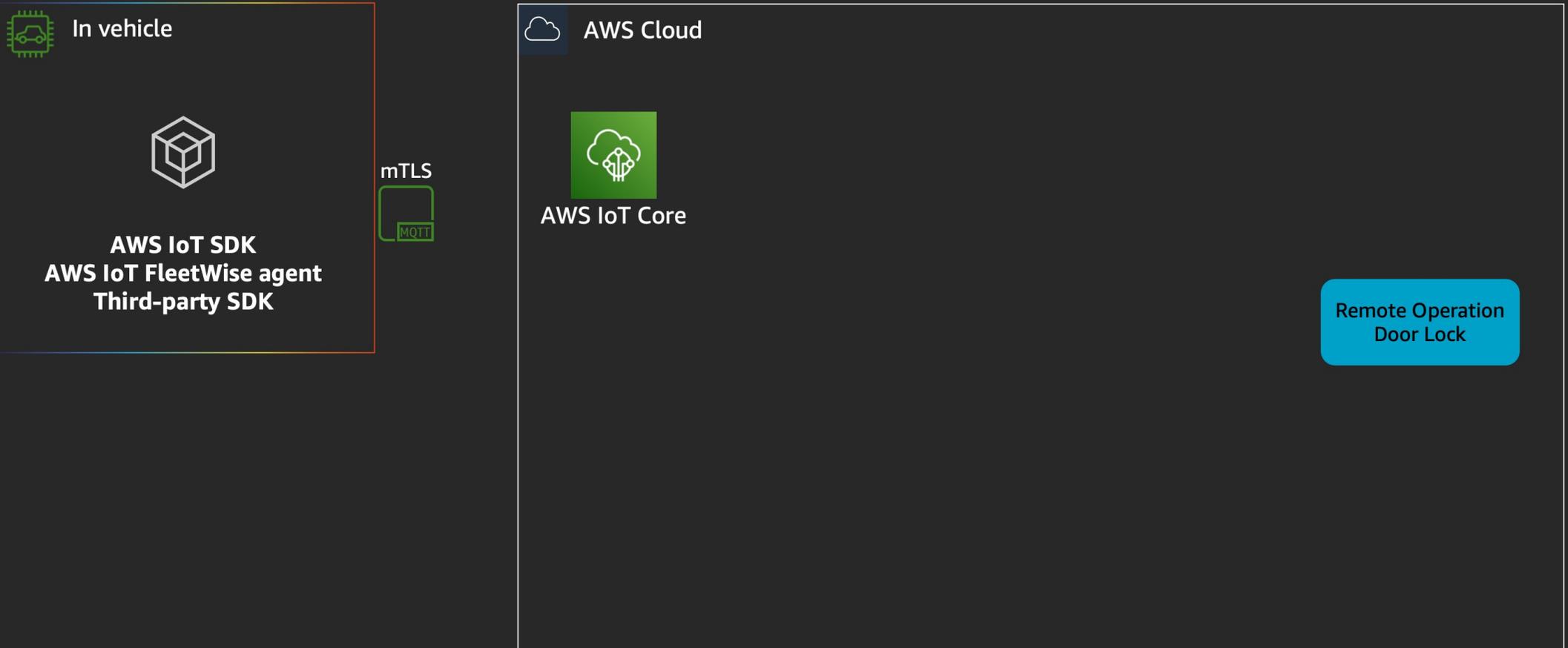
MQTT5 活用ユースケース1 (Request/Response)

- ユーザー側から送信したリモートコマンドが成功したかどうかを知りたい



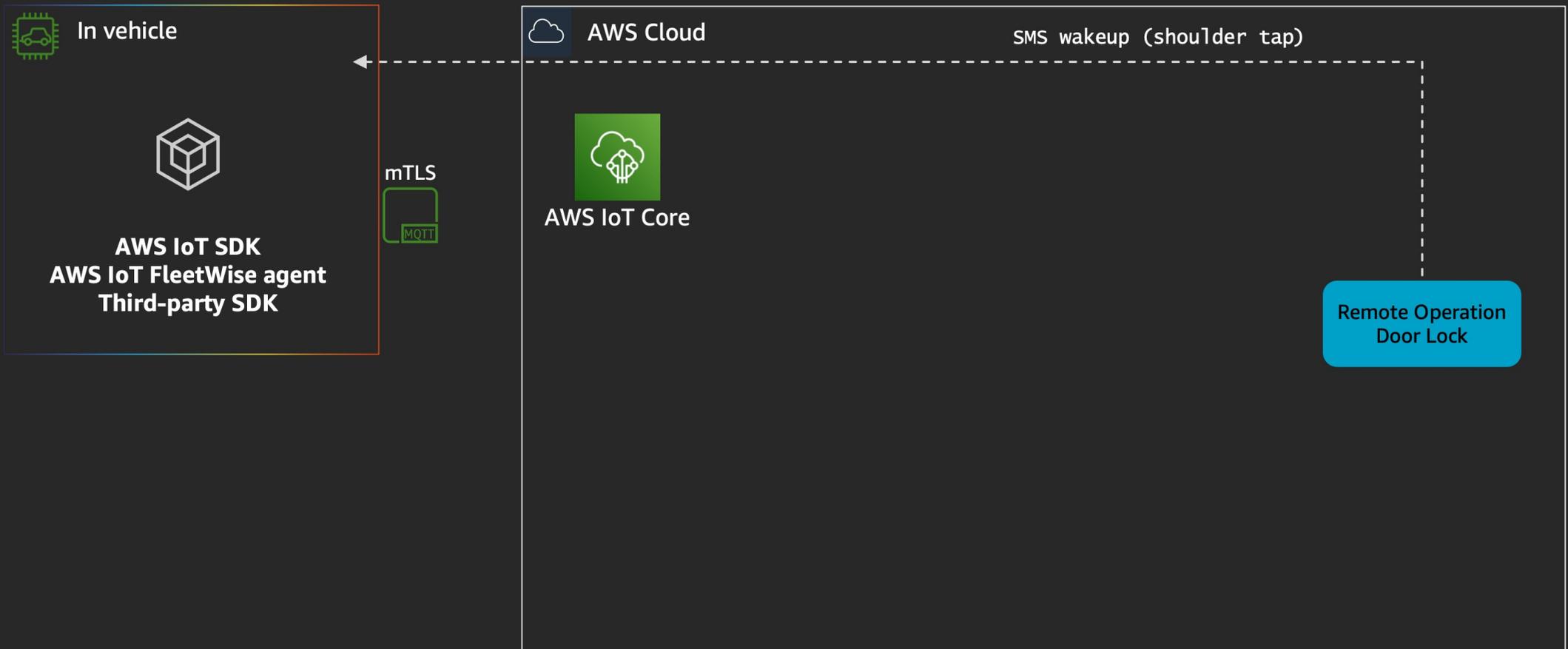
MQTT5 活用ユースケース2 (メッセージ有効期限、セッション有効期限、クリーンスタート)

- ユーザー側から送信したリモートコマンドに有効期限を設定し、古いコマンドを車両側に送信したくないケース



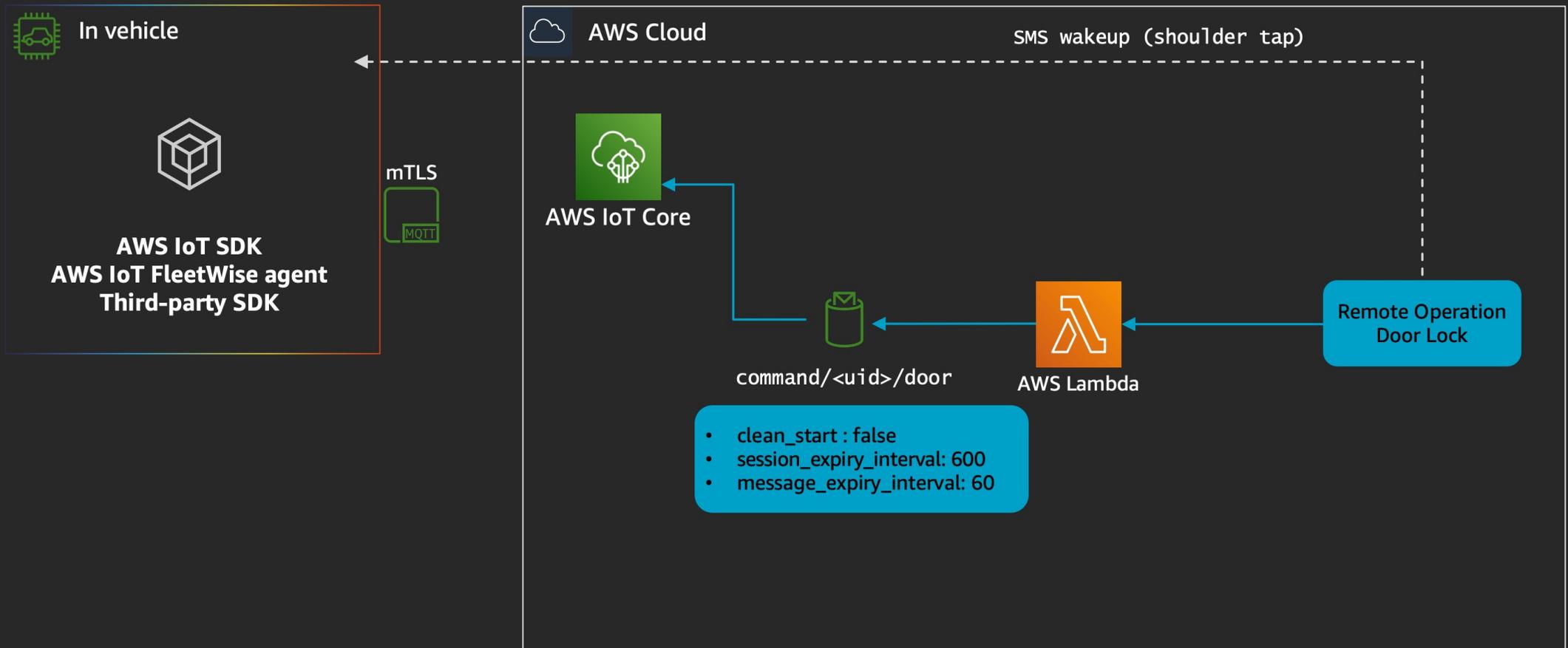
MQTT5 活用ユースケース2 (メッセージ有効期限、セッション有効期限、クリーンスタート)

- ユーザー側から送信したリモートコマンドに有効期限を設定し、古いコマンドを車両側に送信したくないケース



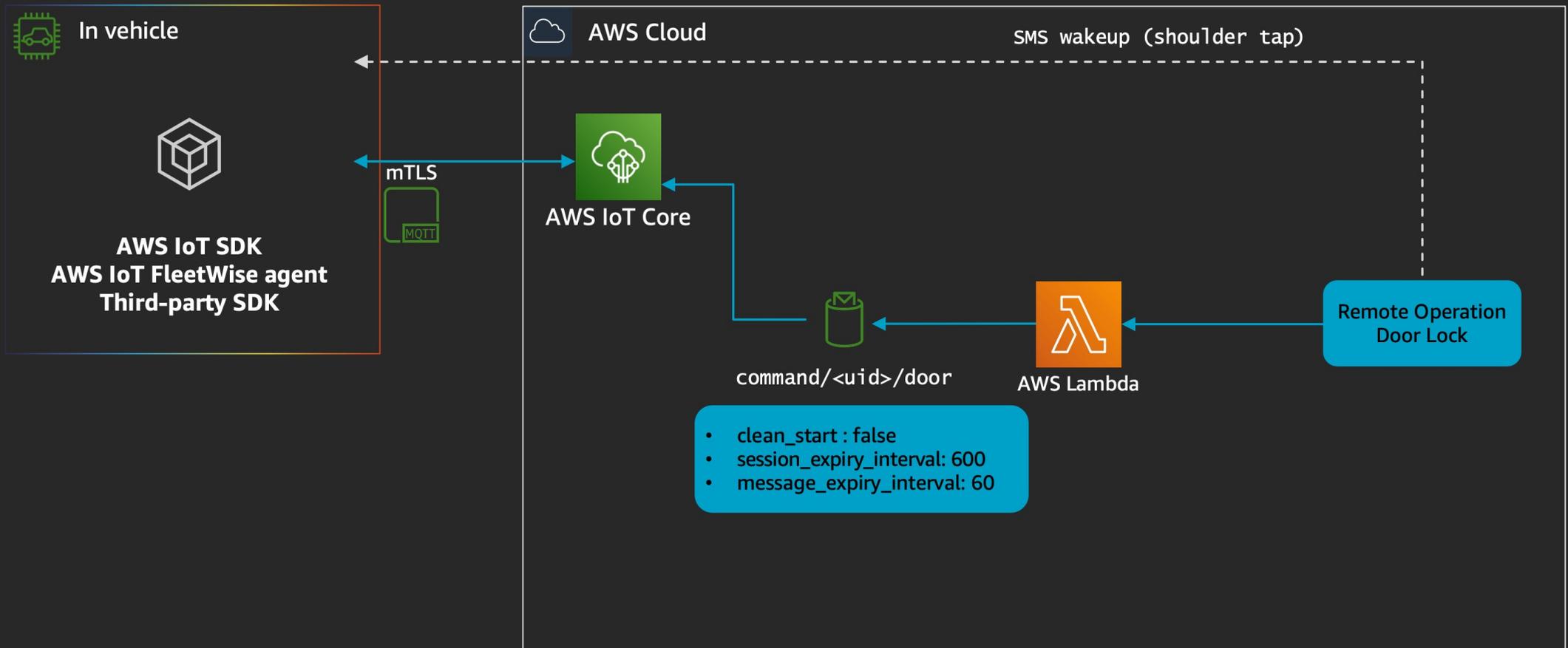
MQTT5 活用ユースケース2 (メッセージ有効期限、セッション有効期限、クリーンスタート)

- ユーザー側から送信したリモートコマンドに有効期限を設定し、古いコマンドを車両側に送信したくないケース



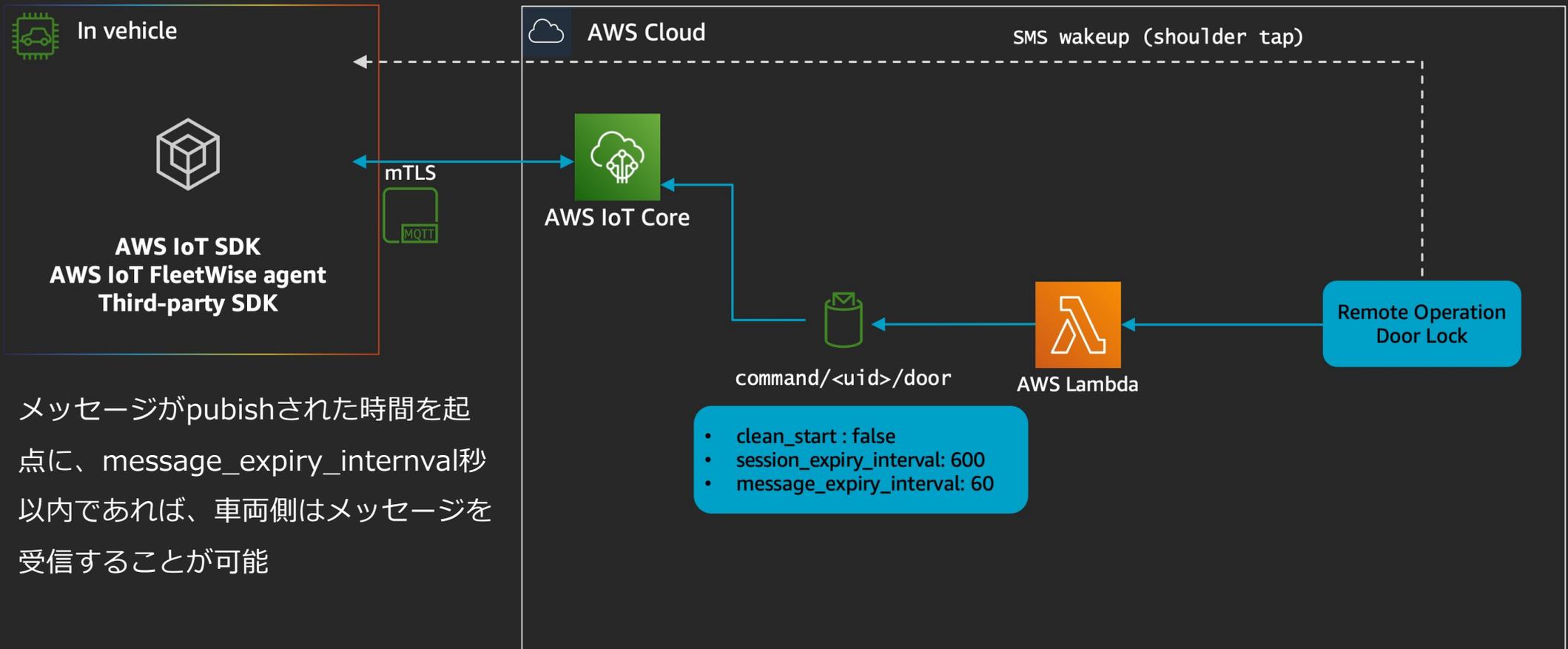
MQTT5 活用ユースケース2 (メッセージ有効期限、セッション有効期限、クリーンスタート)

- ユーザー側から送信したリモートコマンドに有効期限を設定し、古いコマンドを車両側に送信したくないケース



MQTT5 活用ユースケース2 (メッセージ有効期限、セッション有効期限、クリーンスタート)

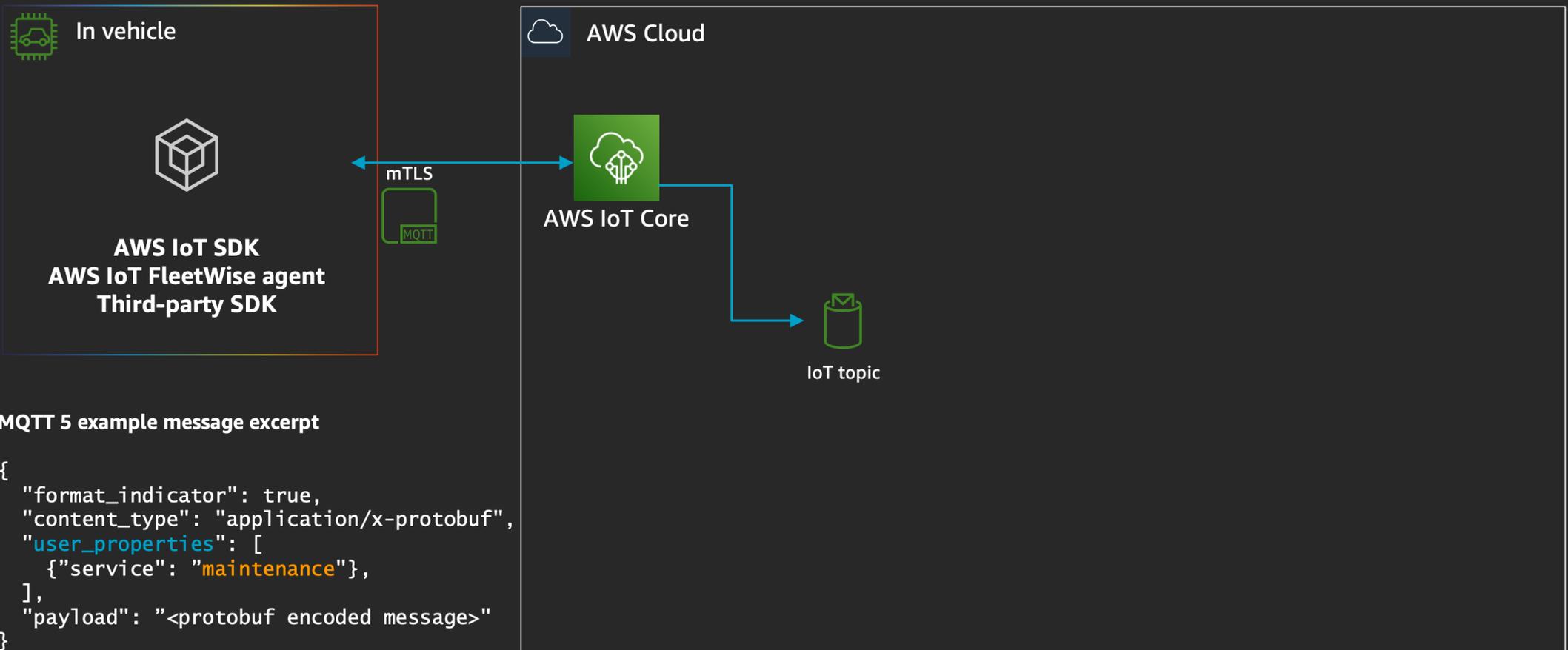
- ユーザー側から送信したリモートコマンドに有効期限を設定し、古いコマンドを車両側に送信したくないケース



メッセージがpublishされた時間を起
点に、`message_expiry_interval`秒
以内であれば、車両側はメッセージを
受信することが可能

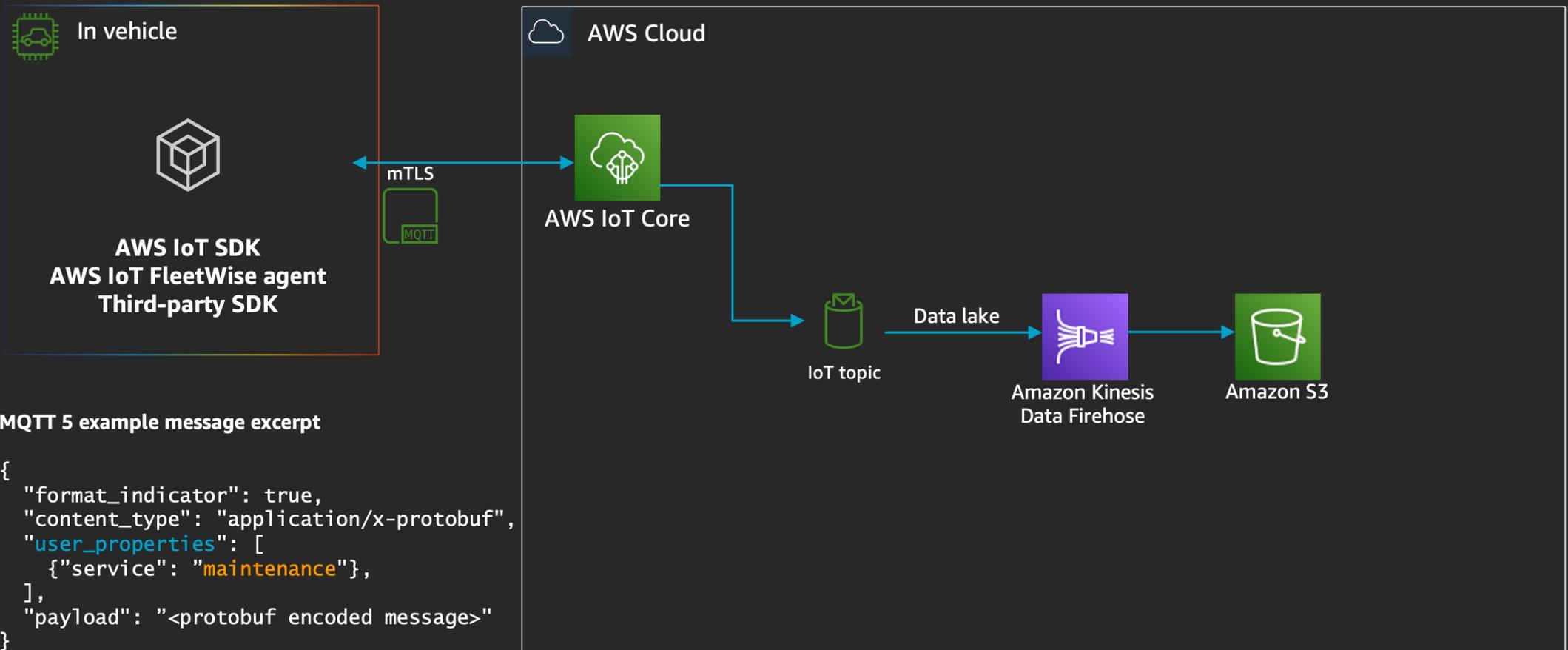
MQTT5 活用ユースケース3 (User Properties)

- 同一トピックで受信したある特定のメッセージをルーティングし、別の処理を実行したいケース
- 以下の例では、protobuf でエンコードされているメッセージの中で、**maintenance**のメッセージのみ、JSONデコードしてTimestreamにInsertするケース



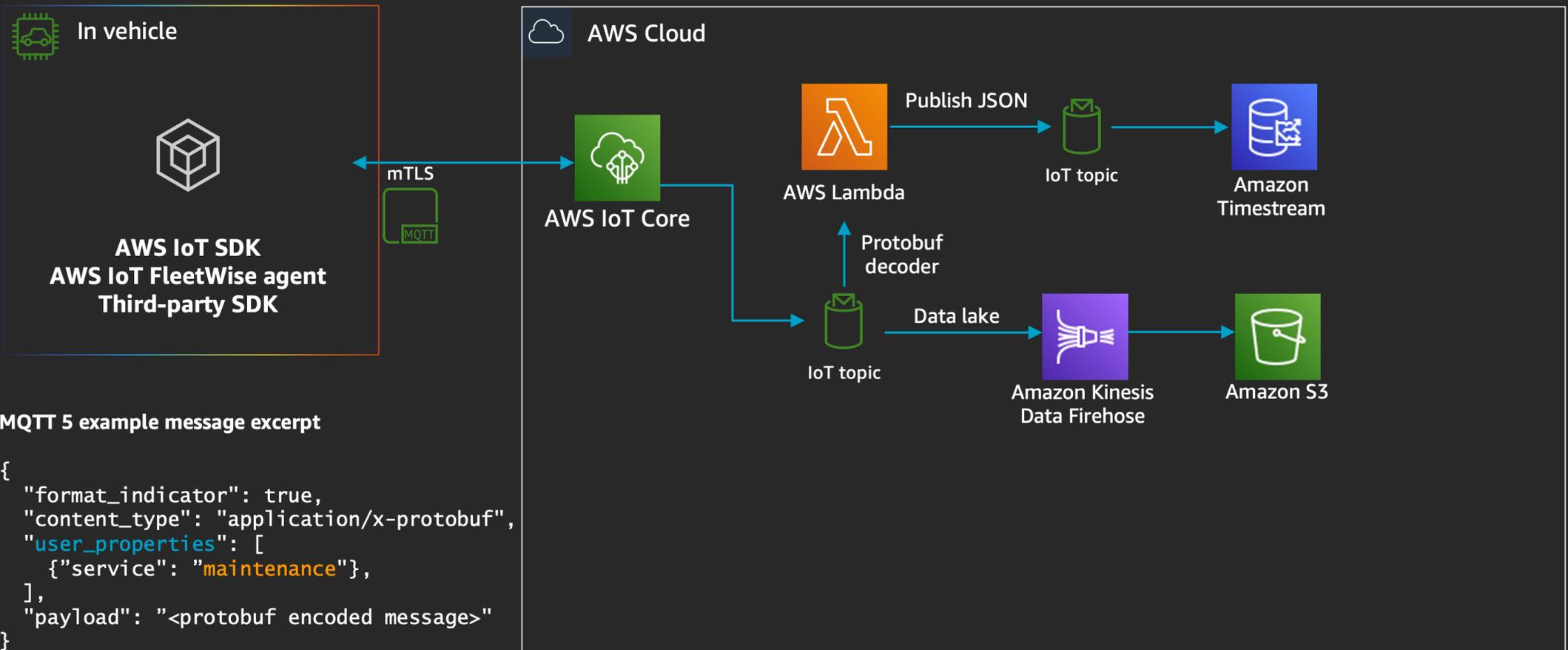
MQTT5 活用ユースケース3 (User Properties)

- 同一トピックで受信したある特定のメッセージをルーティングし、別の処理を実行したいケース
- 以下の例では、protobuf でエンコードされているメッセージの中で、**maintenance**のメッセージのみ、JSONデコードしてTimestreamにInsertするケース



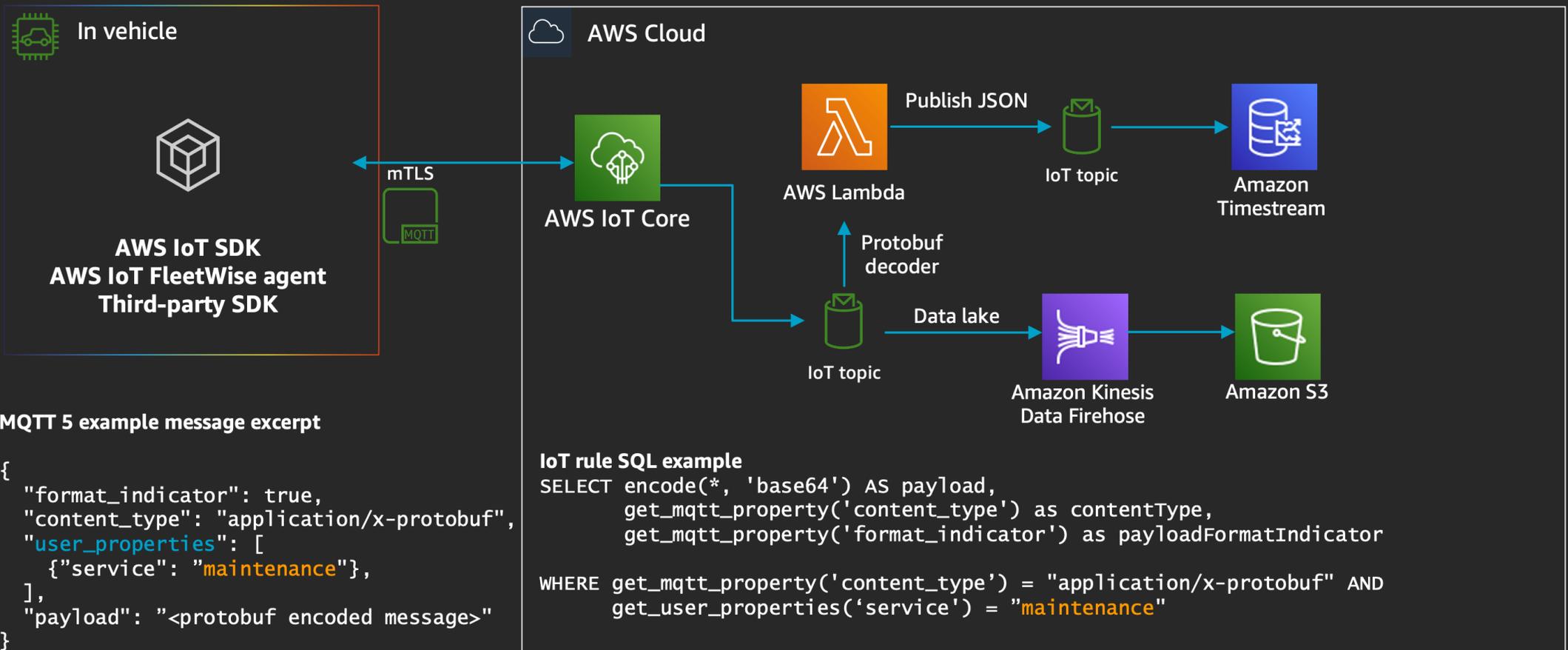
MQTT5 活用ユースケース3 (User Properties)

- 同一トピックで受信したある特定のメッセージをルーティングし、別の処理を実行したいケース
- 以下の例では、protobuf でエンコードされているメッセージの中で、**maintenance**のメッセージのみ、JSONデコードしてTimestreamにInsertするケース



MQTT5 活用ユースケース3 (User Properties)

- 同一トピックで受信したある特定のメッセージをルーティングし、別の処理を実行したいケース
- 以下の例では、protobuf でエンコードされているメッセージの中で、**maintenance**のメッセージのみ、JSONデコードしてTimestreamにInsertするケース



Agenda

- Connected Vehicleの課題
- Connected Vehicleに求められるソリューション
 - AWS Private CAを活用したTLS通信
 - 証明書の失効検知・有効期限の更新
 - AWS IoT Core MQTT 5サポート
 - MQTT5 活用ユースケース1 (Request/Response)
 - MQTT5 活用ユースケース2 (Message Expiry)
 - MQTT5 活用ユースケース3 (User Properties)
- 次世代Connected Vehicleに求められる要件
- AWS IoT Fleetwise概要
- Hands On紹介 AWS IoT FleetWise による EV バッテリーモニタリングソリューションの構築

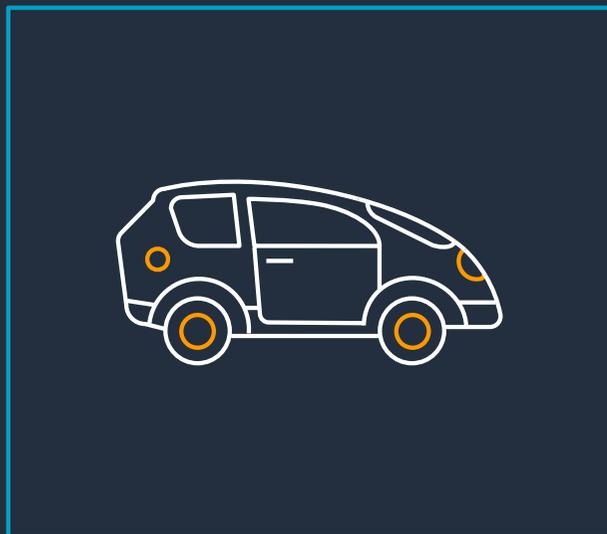
次世代Connected Vehicleに求められる要件

背景：車載ソフトウェアの増加に伴い、車のブラックボックス化が進行する

要件：何か問題が発生した場合に、ある条件下において特定の車両に限定してデータを取得する

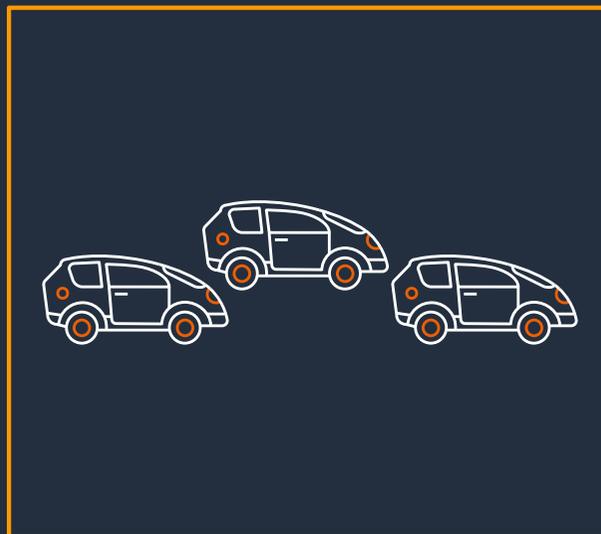
想定ユースケース

単一車両：課題追跡



車両識別番号

特定車種：バッテリーヘルス



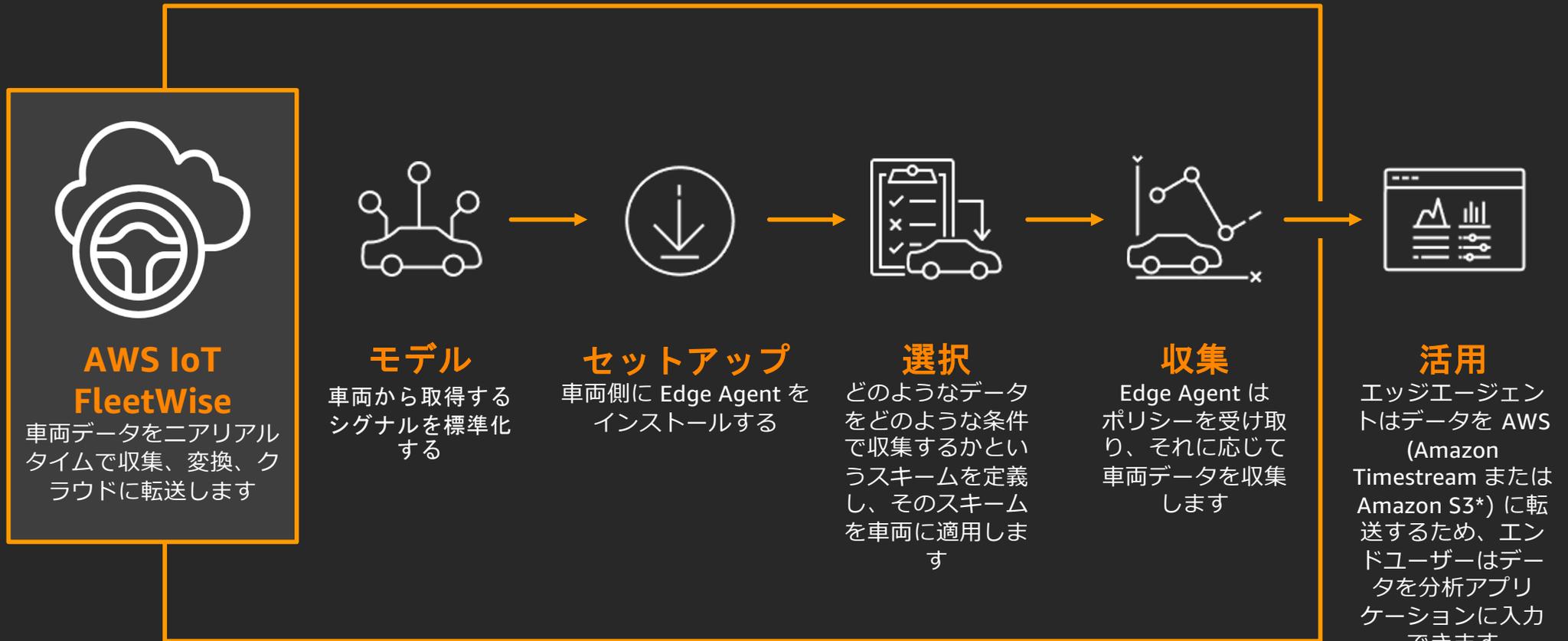
北米における2020年型EVの全車種

過電流が発生したEV車両のBMS状態追跡

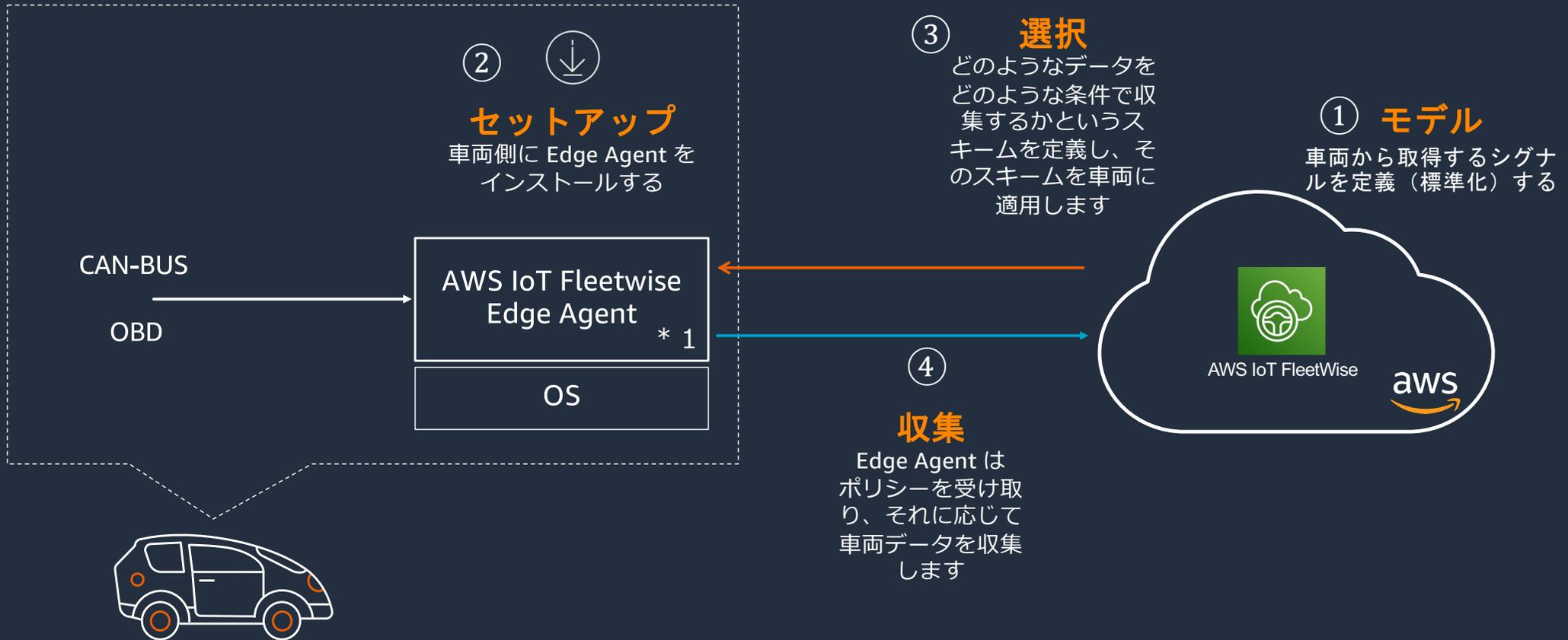


例えば450アンペア以上の電流を検出した際に、BMSの状態を50msec間隔で取得したい

AWS IoT Fleetwise 概要



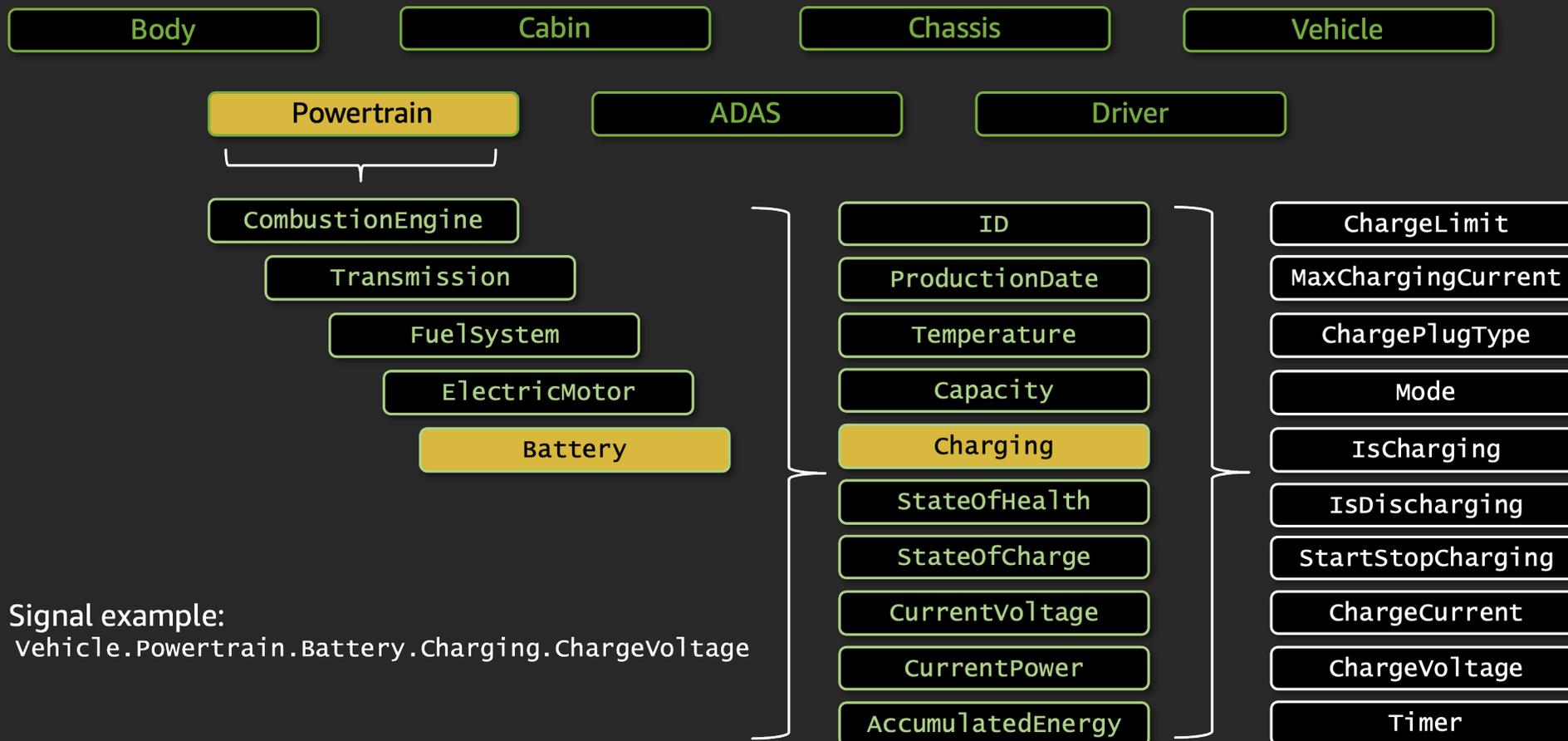
AWS IoT Fleetwise 概要



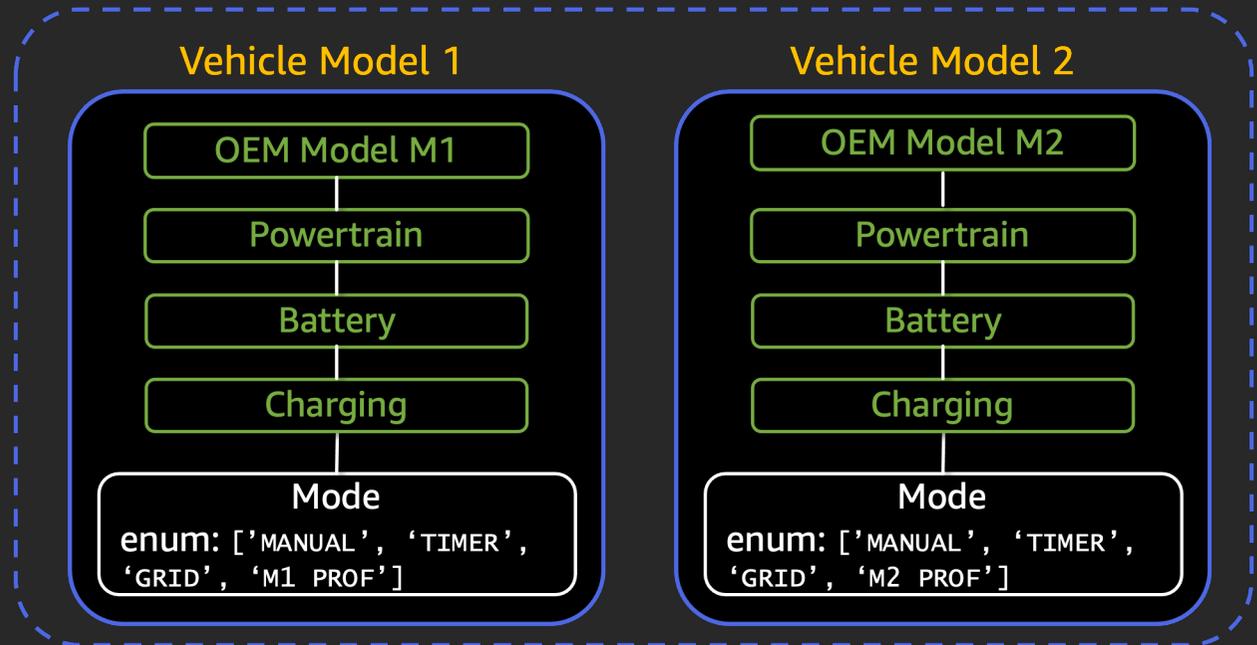
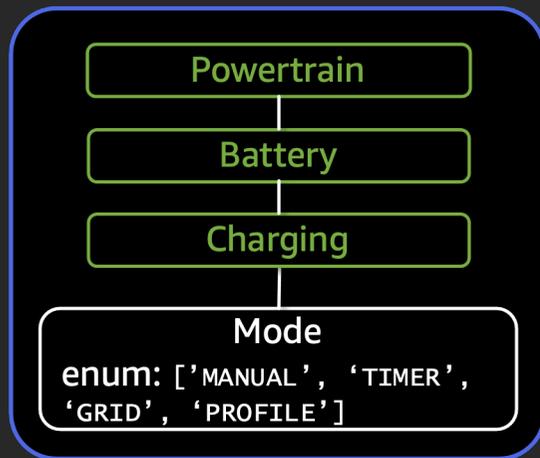
* 1

- 車載側にはEdge Agentと呼ばれるデーモンが常駐する。Edge Agent (リファレンス実装) は、ほとんどの組み込みLinuxベースのプラットフォームで動作する。QNXはaicas (AWSパートナー) がサポート
- ソースコードはGithubにてオープンソースで公開

Modeling vehicle data with VSS



Securely mapping private signals to a global signal catalog



VSS global signal catalog



OEM proprietary signals
(VSPEC, CAN DBC)

Hands Onご紹介

AWS IoT FleetWise による EV バッテリーモニタリングソリューションの構築



<https://aws.amazon.com/jp/blogs/news/building-an-ev-battery-monitoring-solution-with-aws-iot-fleetwise-part-1-2/>

Thank you!

眞壽田 英輝 (Masuta, Hideki)

ソリューションアーキテクト Mobility 領域担当
アマゾンウェブサービスジャパン合同会社

