



A161

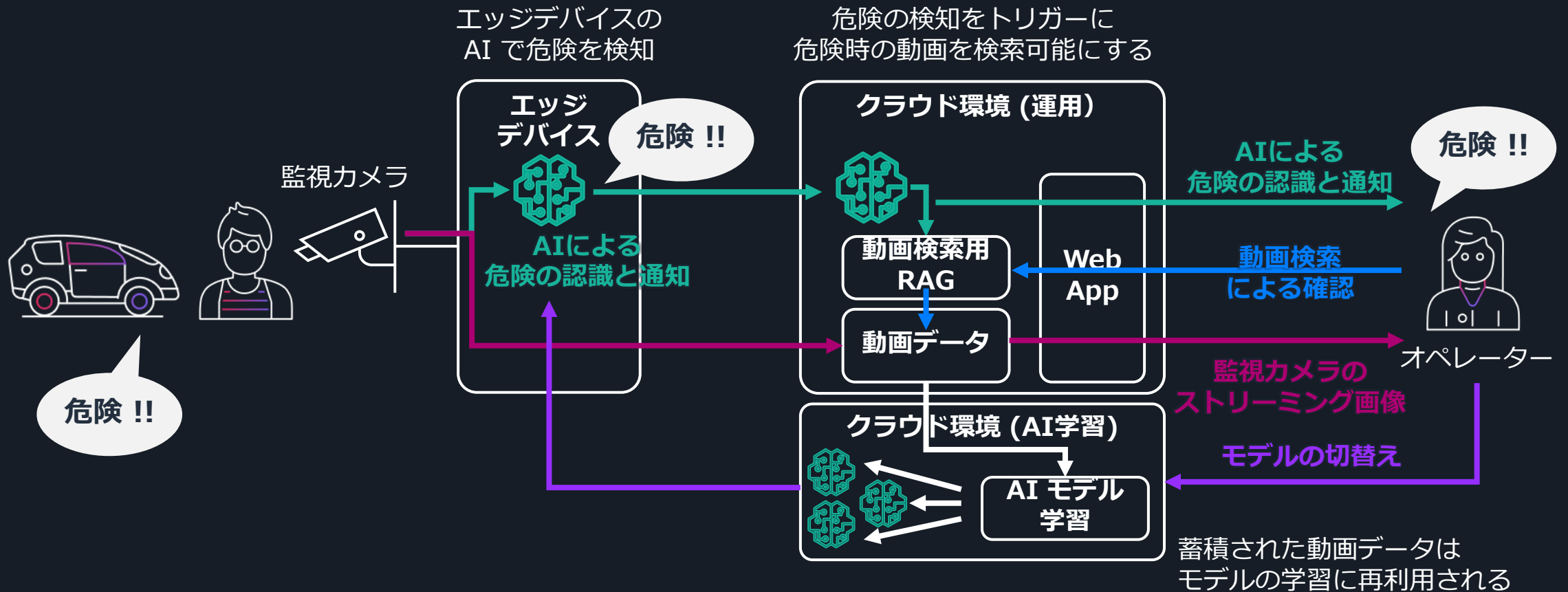
Physical AIへの第一歩

— AWS IoTで実現するPhysical AI基盤 —

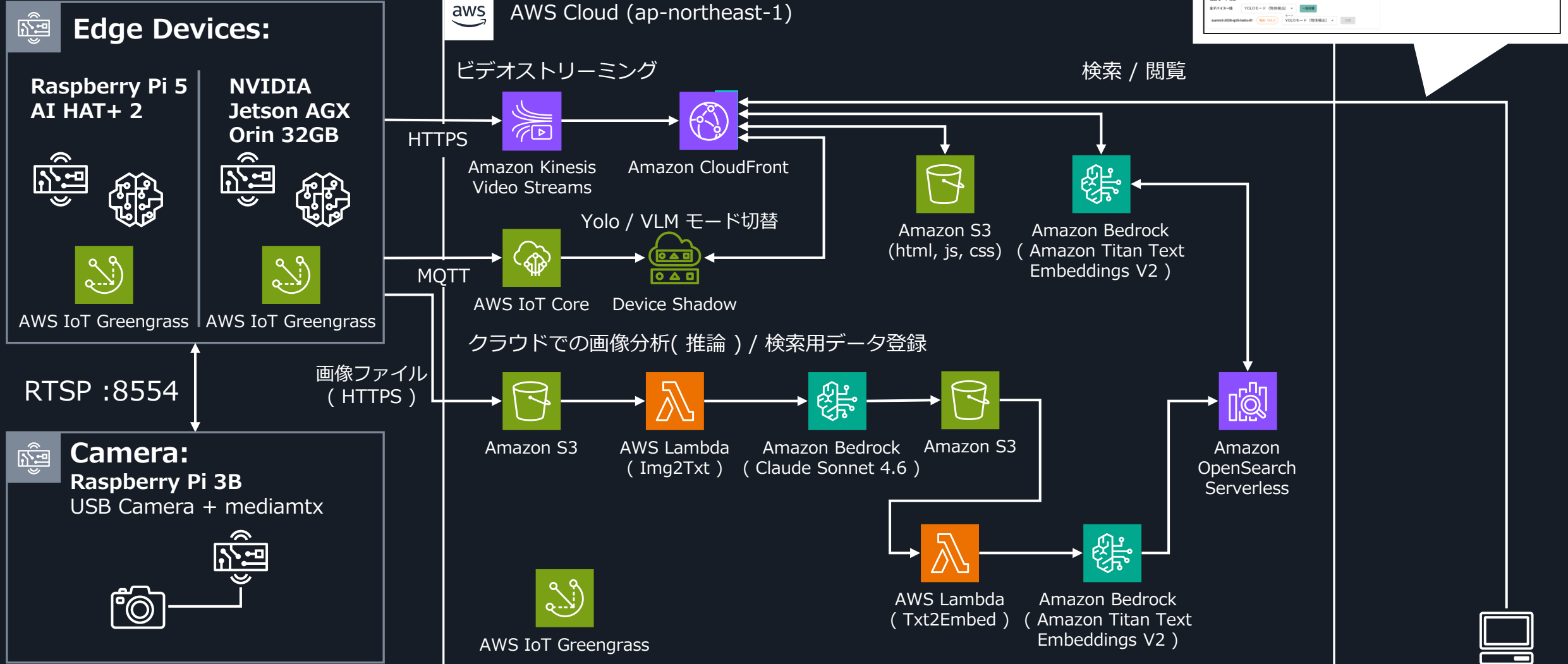
アマゾン ウェブ サービス ジャパン 合同会社

デモ概要

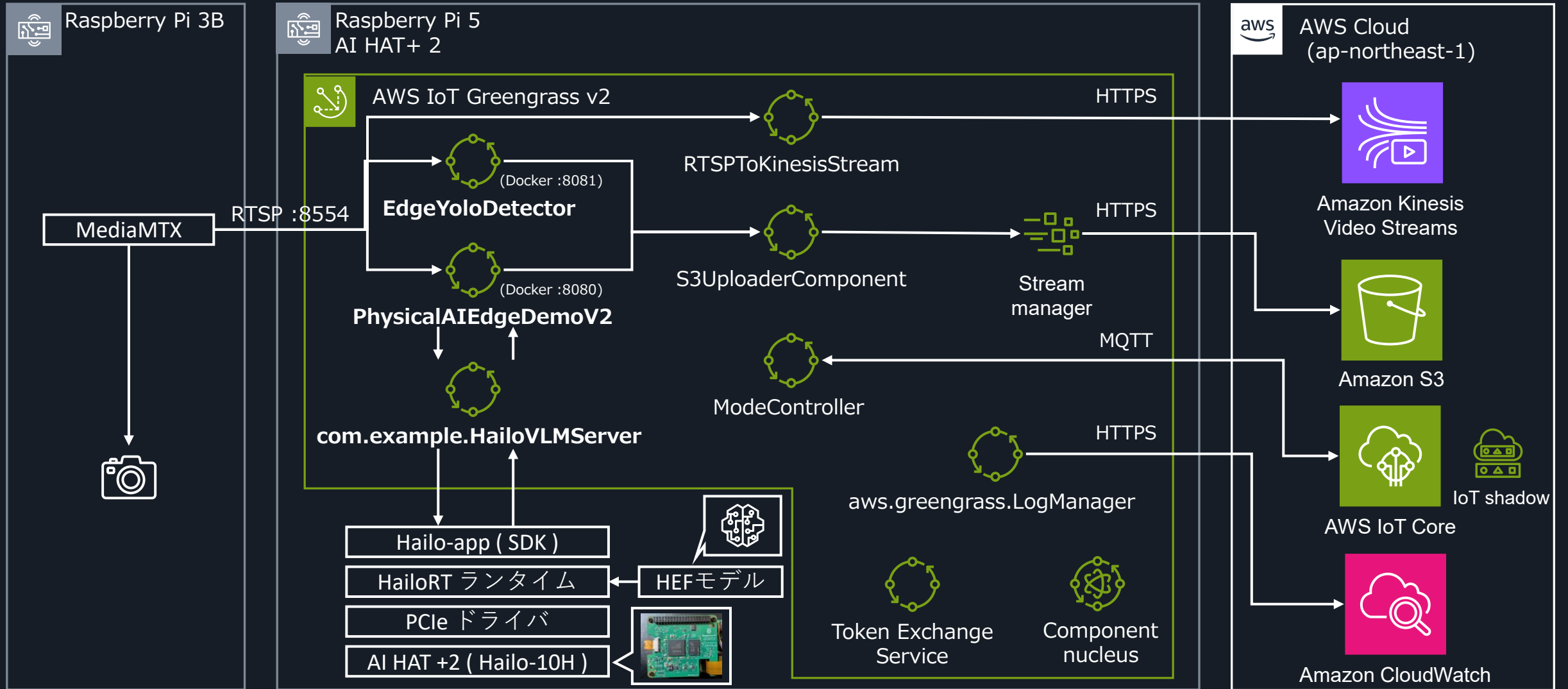
エッジデバイスの生成AIモデルでカメラ画像から危険を検知し、オペレーターに通知するとともに、動画検索可能なデータベースを構築



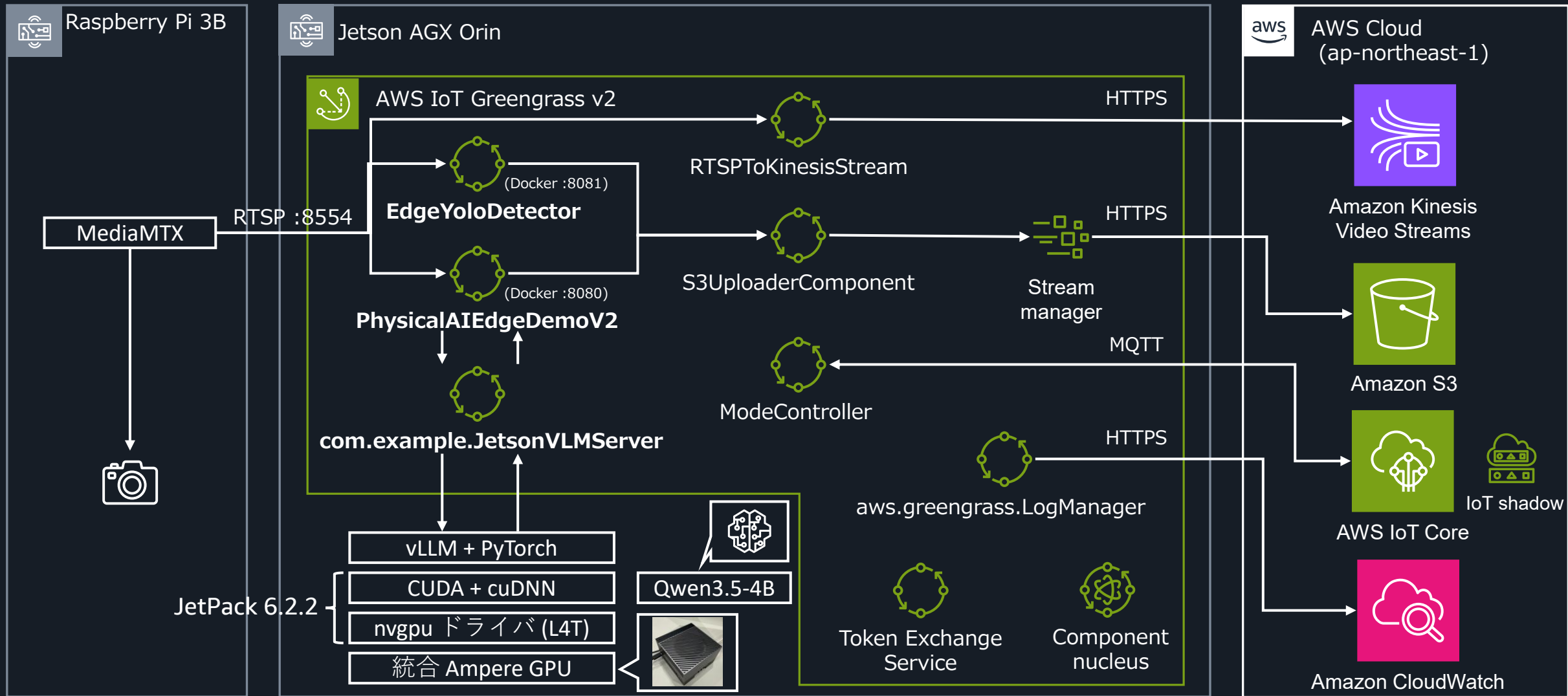
全体アーキテクチャ



アーキテクチャ in Raspberry Pi 5

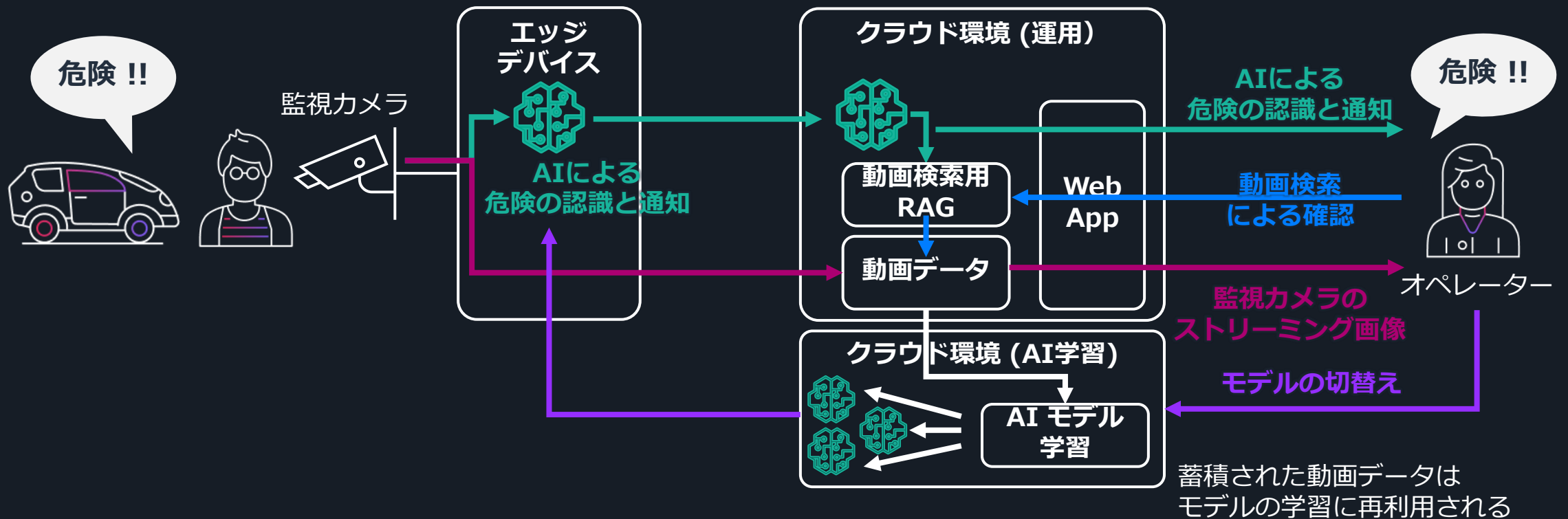


アーキテクチャ in Jetson AGX Orin



ユースケース

カメラが危険な事象を検出してから、オペレーターが動画を検索して危険な事象を確認するまでの流れを説明します。



ユースケース

① ストリーミング画像による監視



Generative AI Edge Camera

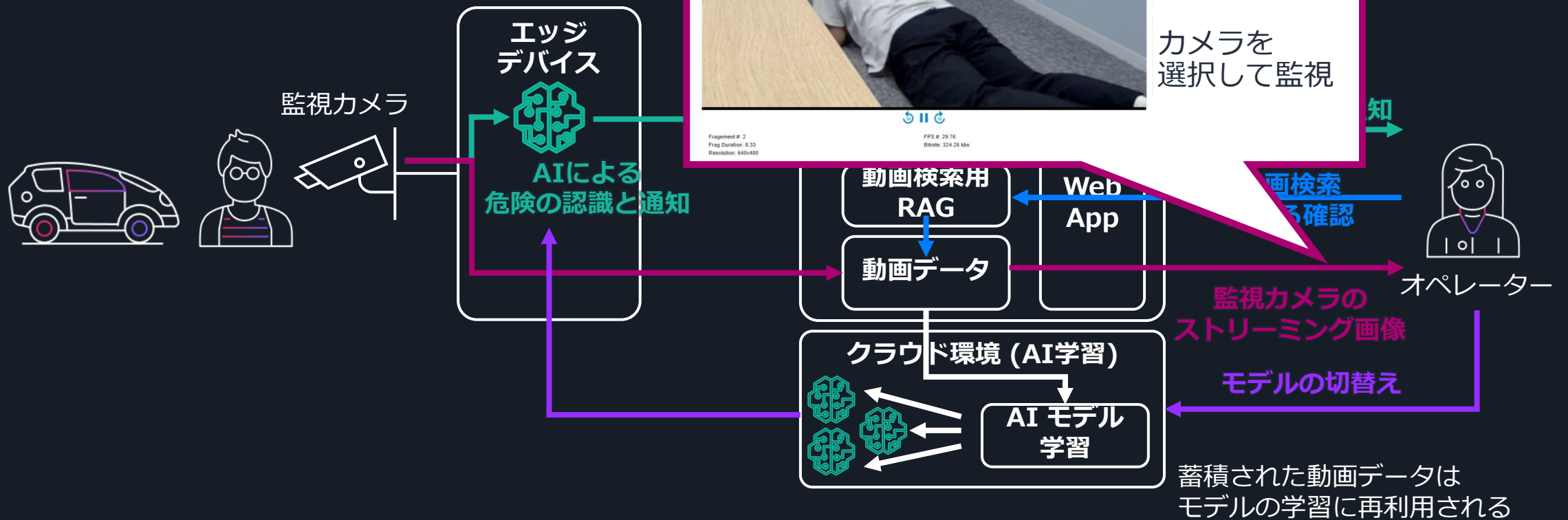
カメラ
summit-2026-pi5-hailo-01 再生開始

カメラを選択してください

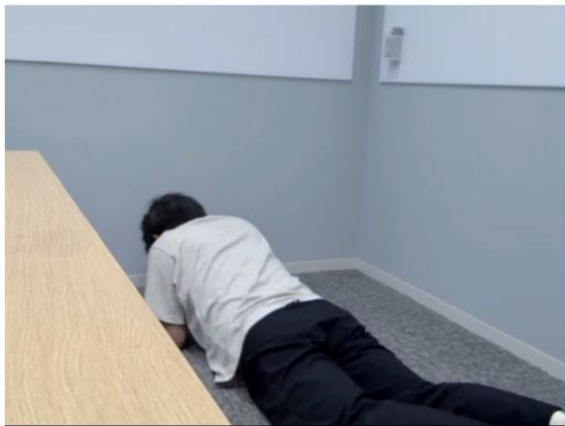
カメラを選択して監視

Fragment # 2
Frag Duration: 0.33
Resolution: 640x480

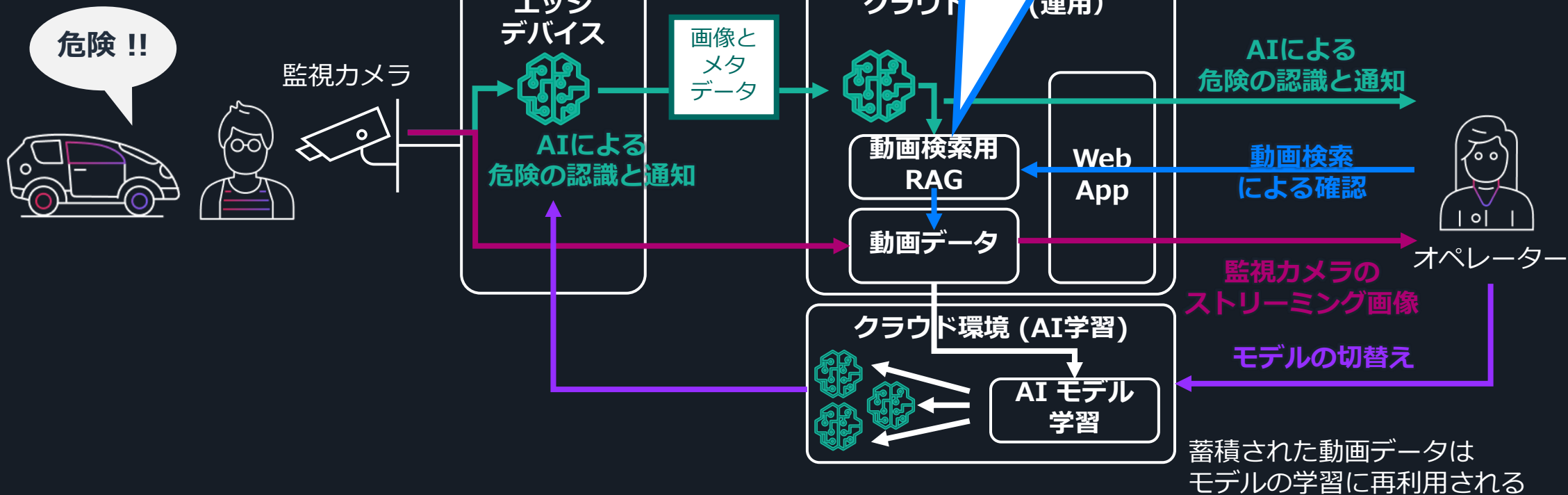
FPS: 29.76
Bitrate: 324.26 kbps



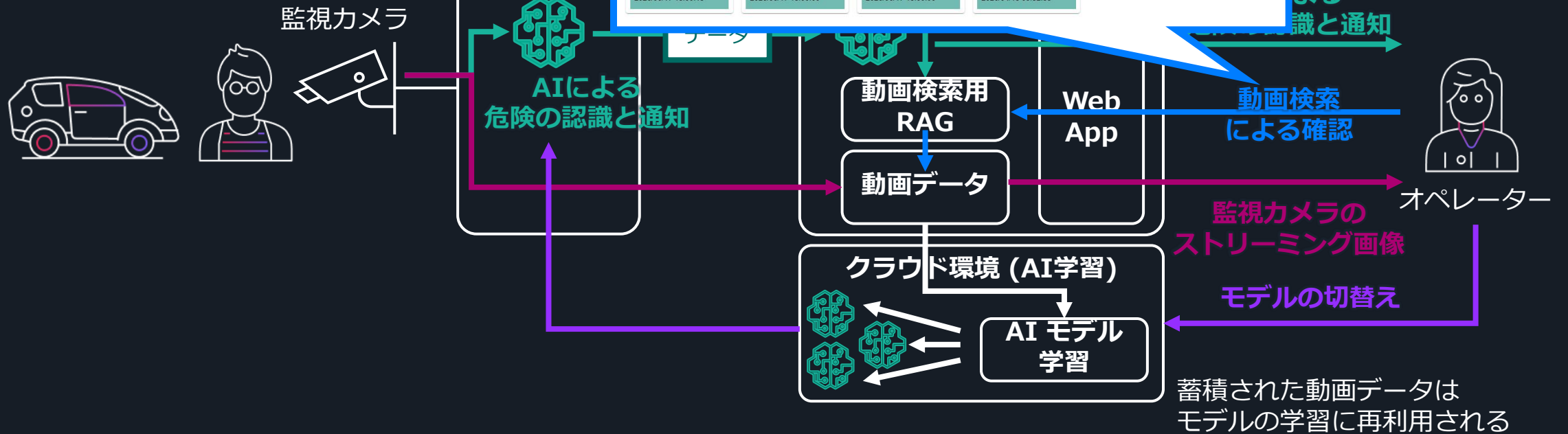
② エッジAIによる危険の検知



③ 動画検索用のRAGに記録



ユースケース



ユースケース

⑤ モデルの切替

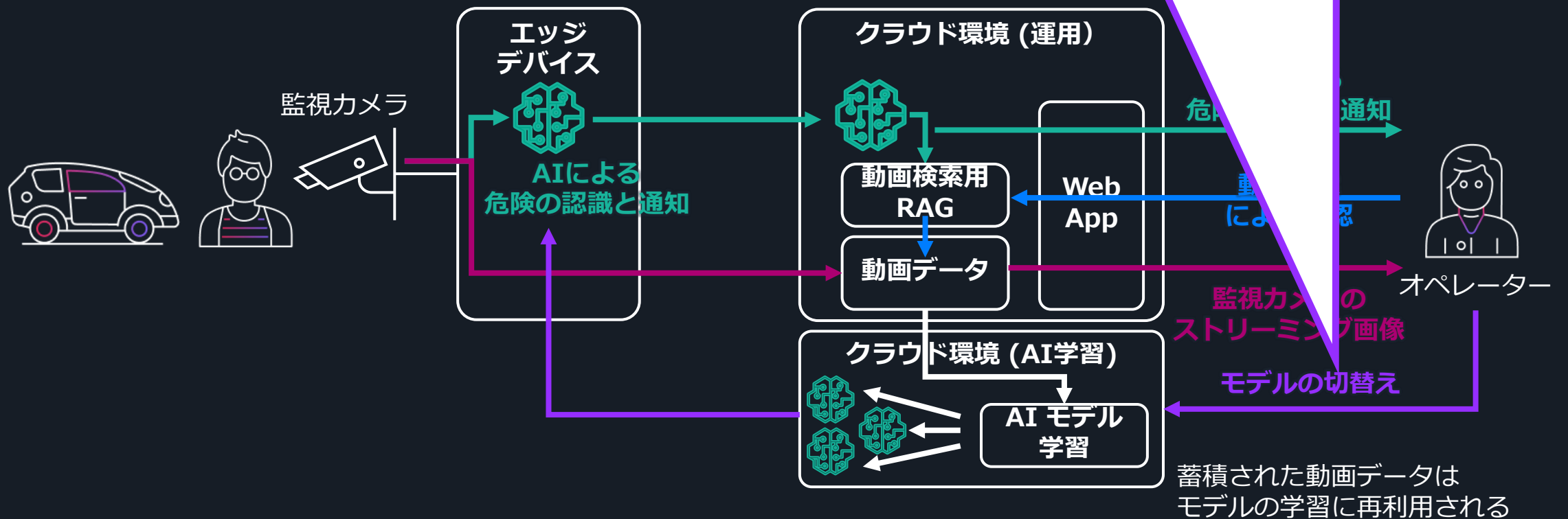
推論モード切替

全デバイス一括 YOLOモード (物体検出) 一括切替

summit-2026-jetson-agx-01 現在: VLM VLMモード (VLM画像分析) 切替

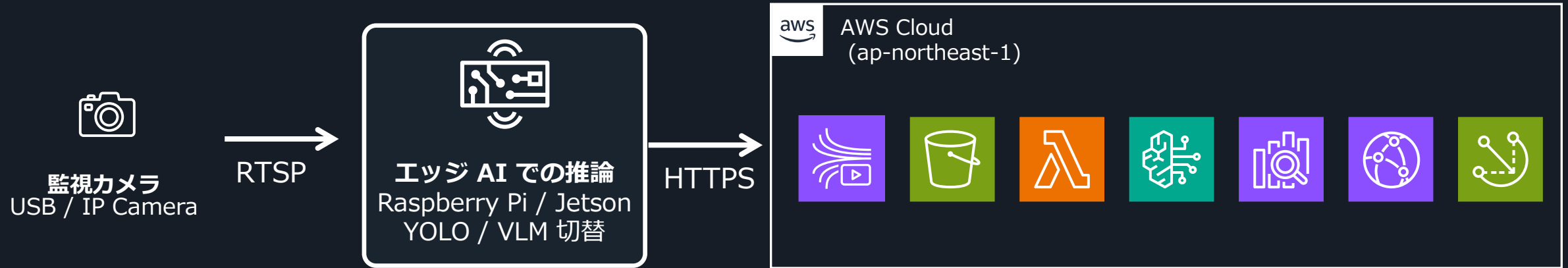
summit-2026-rpi5-hailo-01 現在: VLM VLMモード (VLM画像分析) 切替

YOLO と Qwen を切り替える



全体アーキテクチャ

カメラ映像をクラウドへ配信 / 分析 / 検索 する基盤



リアルタイム映像配信

RTSP → KVS → CloudFront で
世界中のブラウザから
ライブ映像を視聴可能



生成 AI で状況分析

Amazon Bedrock
(Claude Sonnet 4.6) が
画像を文章化し
危険度を自動スコアリング
+ キーワード抽出



自然言語で映像検索

「赤い車」「危険な状態」と
入力するだけで
該当シーンの映像と画像を
即座に検索

デバイス管理

複数拠点のエッジデバイスをクラウドから一元管理

複数拠点のエッジ端末

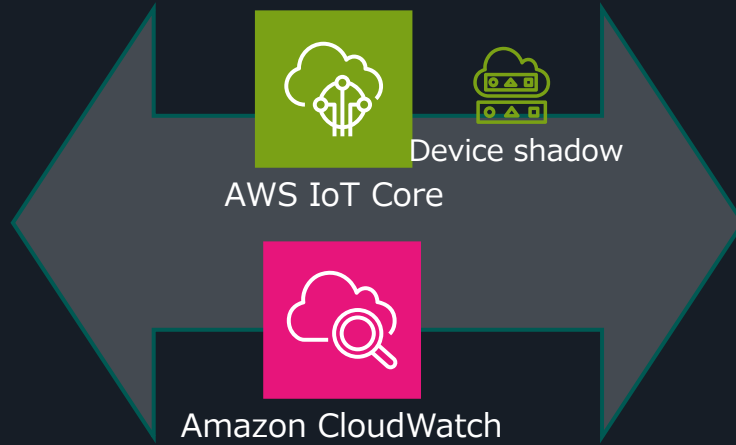
工場 A : RP5 × 3 台
YOLO モード稼働中



工場 B : Jetson × 5 台
VLM モード稼働中



倉庫 C : RP5 × 2 台
新規デプロイ待ち



AWS IoT サービス (①)

- AWS コンソールで一括デプロイ
- Fleet Provisioning で新規デバイス自動登録

Web アプリ (②)

- デバイスシャドウによるモードの切り替え

Amazon CloudWatch (③)

- ログ / メトリクス監視

① OTA デプロイ

コンポーネント単位で更新可能
AI モデル差替えもリモートで反映
ロールバック対応



② 遠隔モード切替

Device Shadow で desired 変更
ModeController が検知して切替
数秒で切替完了



③ 一元監視

LogManager → CloudWatch
全拠点のログを集約
異常検知アラート設定可能



クラウド連携 - 映像配信と検索基盤 -

映像配信と画像検索を2つのパイプラインで実現



すべて AWS マネージドサービス サーバー管理不要 / スケーラブル / 従量課金

✓ リージョン : ap-northeast-1 で完結

✓ Bedrock モデル : Claude Sonnet 4.6 + Titan Embed Text v2

エッジで動く生成AI - Raspberry Pi 5 -

Raspberry Pi で 2B パラメータ VLM がリアルタイム動作

パラメータ

2B

Qwen2-VL-2B / INT4

INT4 推論性能

40 TOPS

Hailo-10H AI HAT+ 2

ハードウェア構成

約 7 万円

RP5 + AI HAT+ 2 + ストレージ

用途で使い分け — YOLO vs VLM

	YOLO v8 nano (物体検出)	VLM (画像理解)
速度	リアルタイム (30 fps 以上)	数秒~数十秒 / 1 フレーム
柔軟性	学習済みクラスのみ検出	自然言語プロンプトで任意に指定
向いているユースケース	侵入検知・車両カウント 人数計測など定型タスク	異常行動の検知 状況説明文の自動生成

エッジで動く生成AI - Jetson AGX Orin

Jetson で 4B パラメータ VLM がリアルタイム動作

パラメータ

4B

Qwen3.5-4B

推論性能

241TOPS

Jetson AGX Orin (32GB)

ハードウェア構成

約40万円

Jetson + ストレージ

用途で使い分け — YOLO vs VLM

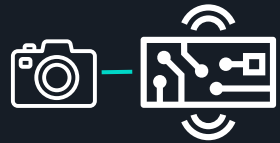
	YOLO v26 nano (物体検出)	VLM (画像理解)
速度	リアルタイム (30 fps 以上)	数秒~数十秒 / 1 フレーム
柔軟性	学習済みクラスのみ検出	自然言語プロンプトで任意に指定
向いているユースケース	侵入検知・車両カウント 人数計測など定型タスク	異常行動の検知 状況説明文の自動生成

<https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-orin/>

実装時の工夫 in Raspberry Pi 5

Kiro が自動で接続・把握できる環境を整える

SSH
ファイルを見たり、コマンド実行



RaspberryPi 3B
カメラ
RTSPサーバー



RaspberryPi 5
エッジAIデバイス
Greengrass



SSH 鍵認証への切替

Kiro からパスワードなしでSSH接続できるようにする

効果

Kiro から 各デバイスへ安全に接続可能。パスワード総当たり攻撃を防止し、セキュリティ向上。ログ調査等の作業を自動化。

対応内容

ssh-copy-id で公開鍵を RP3B / RP5 に配置し、パスワードでの認証を無効化

Steering でデバイス情報を集約

RP3B と RP5 の用途を明文化し Kiro に文脈を共有

効果

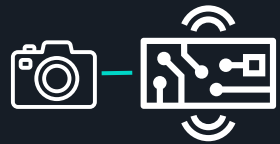
Kiro が Steering で読み込み両デバイスの役割を自動把握し対応が可能。「カメラ側」と「AI 推論側」の役割混同を防止

デバイス	ホスト名	IP アドレス	用途
RP3B	bull	192.168.xxx.xxx	カメラサーバー
RP5	penguin	192.168.xxx.yyy	エッジ AI デバイス

実装時の工夫 in Jetson AGX Orin

※ Jetson においても ssh-copy-id で Kiro / Claude Code から直接デバッグ可能に

Loop Engineering で
効率的に開発



RaspberryPi 3B
カメラ
RTSPサーバー



Jetson AGX Orin
エッジAIデバイス
Greengrass

USB Device Mode
Type-C 接続するだけで固定IP
192.168.55.1 で ssh接続可能(※)



数時間単位の検証を Agent に完全委譲

ブランチ分離で擬似的な Sandbox を実現
ralph loop で完了条件を記載。達成までひたすらループ

効果：Qwen のモデル動作 /バージョンアップ /
パラメータ検証を Loop 内で自律的に完了。

コンテキストの永続化

調査結果/実行ログ/トラブルシューティング等の記録を
全てファイルに残す

効果：コンテキストウィンドウを超えても再説明
不要、同じ失敗を繰り返さない（失敗から学ぶ）

デモシステムのコスト

Amazon Bedrock Service (ナレッジベース、LLMなど)、Amazon OpenSearch Service のコストが支配的です。

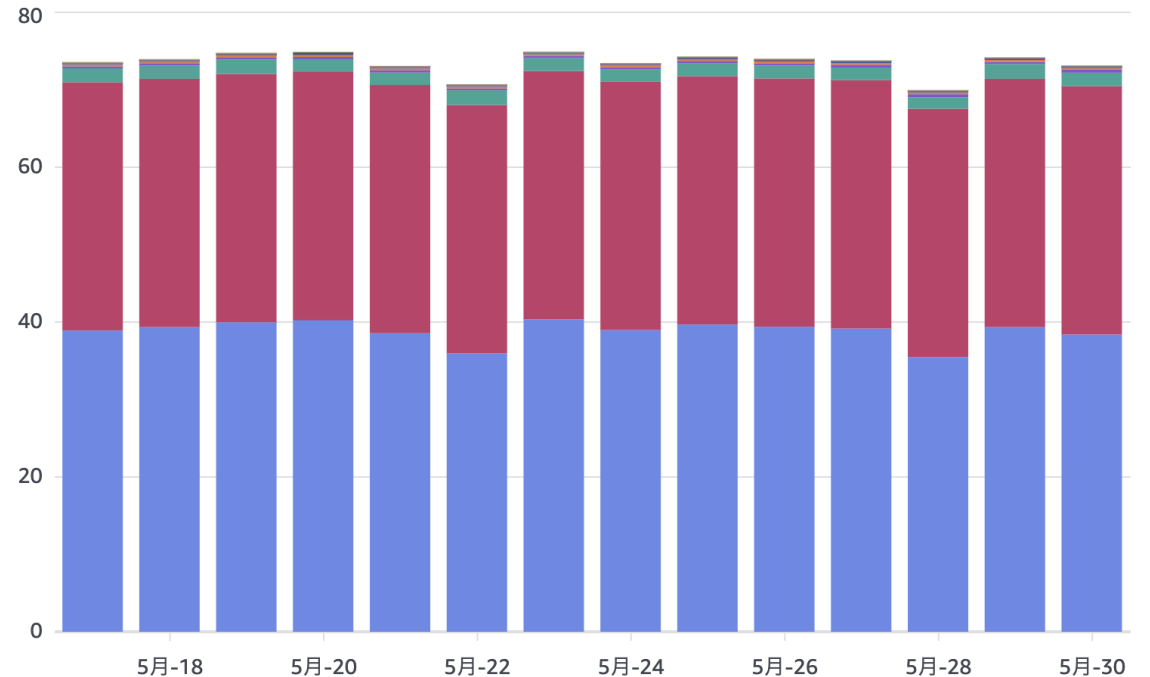
Amazon Kinesis Video Streams 等、その他のサービスのコストは、ごくわずかでした。

- Amazon Bedrock Service
… およそ \$35~40/日
- Amazon OpenSearch Service
… およそ \$32/日

本デモのテスト時の事例です。
条件が異なる場合、コストも変化するため、
参考情報としてご覧ください。

コストと使用量のグラフ

コスト (\$)



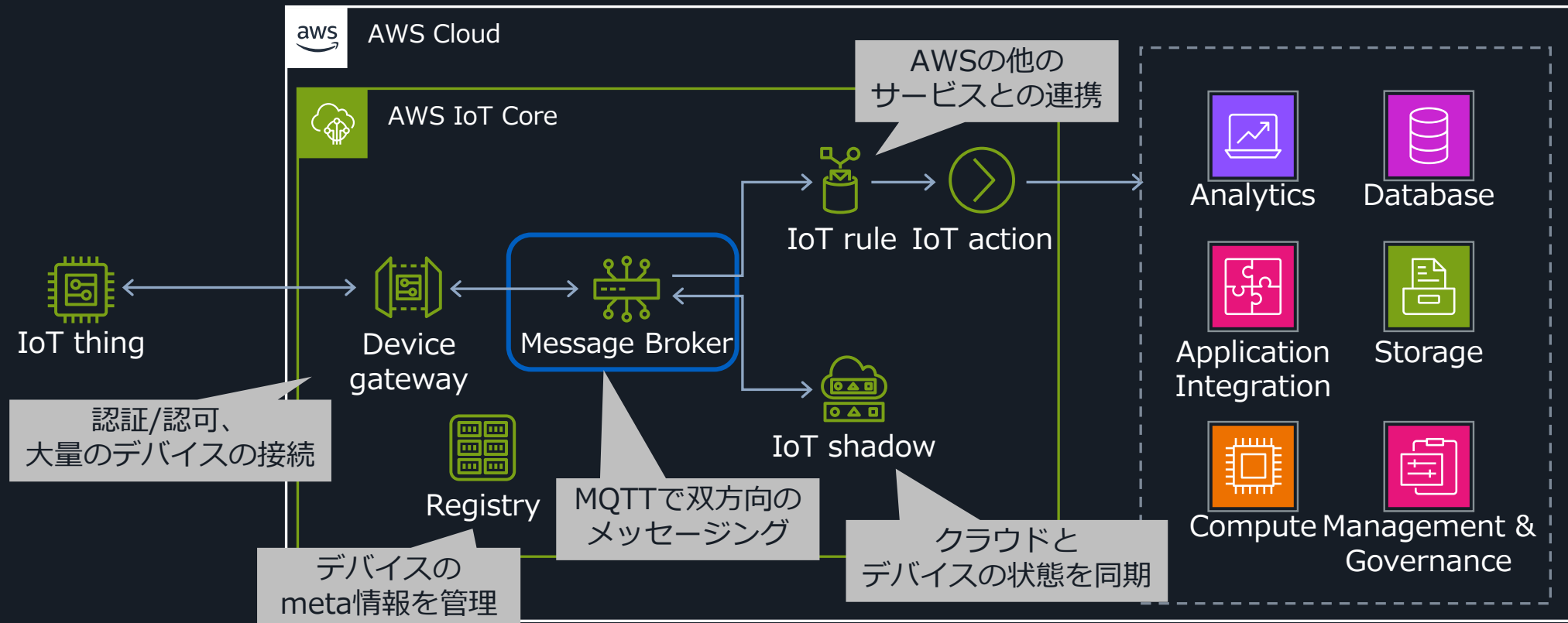
■ Bedrock Service ■ OpenSearch Service ■ CloudWatch ■ Kinesis Video Streams ■ Lambda ■ S3
■ EC2 Container Registry (ECR) ■ IoT ■ Bedrock ■ その他

AWS IoT Services

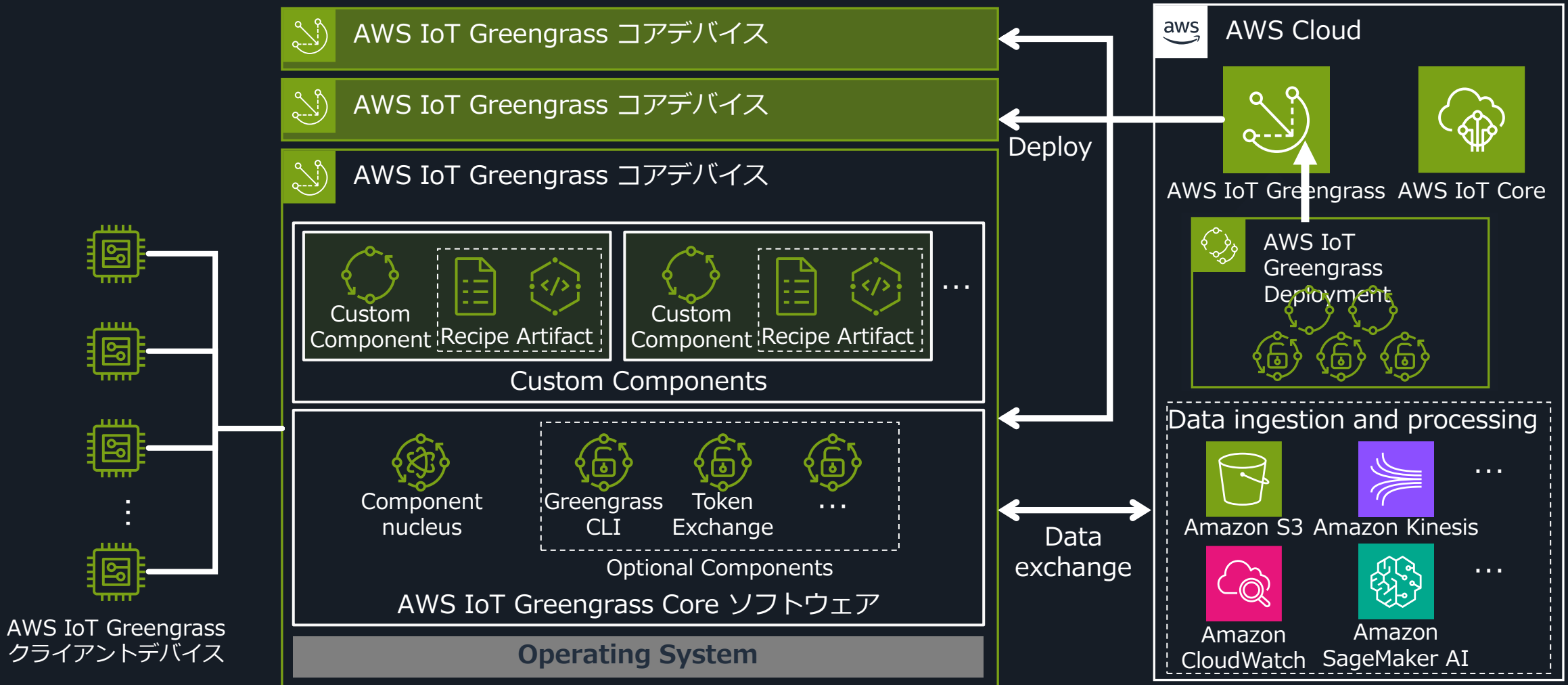


AWS IoT Core のアーキテクチャ概要

AWS IoT Core は、IoT デバイスを安全・簡単にクラウドアプリケーションや他のデバイスと通信できるようにするマネージドサービスです



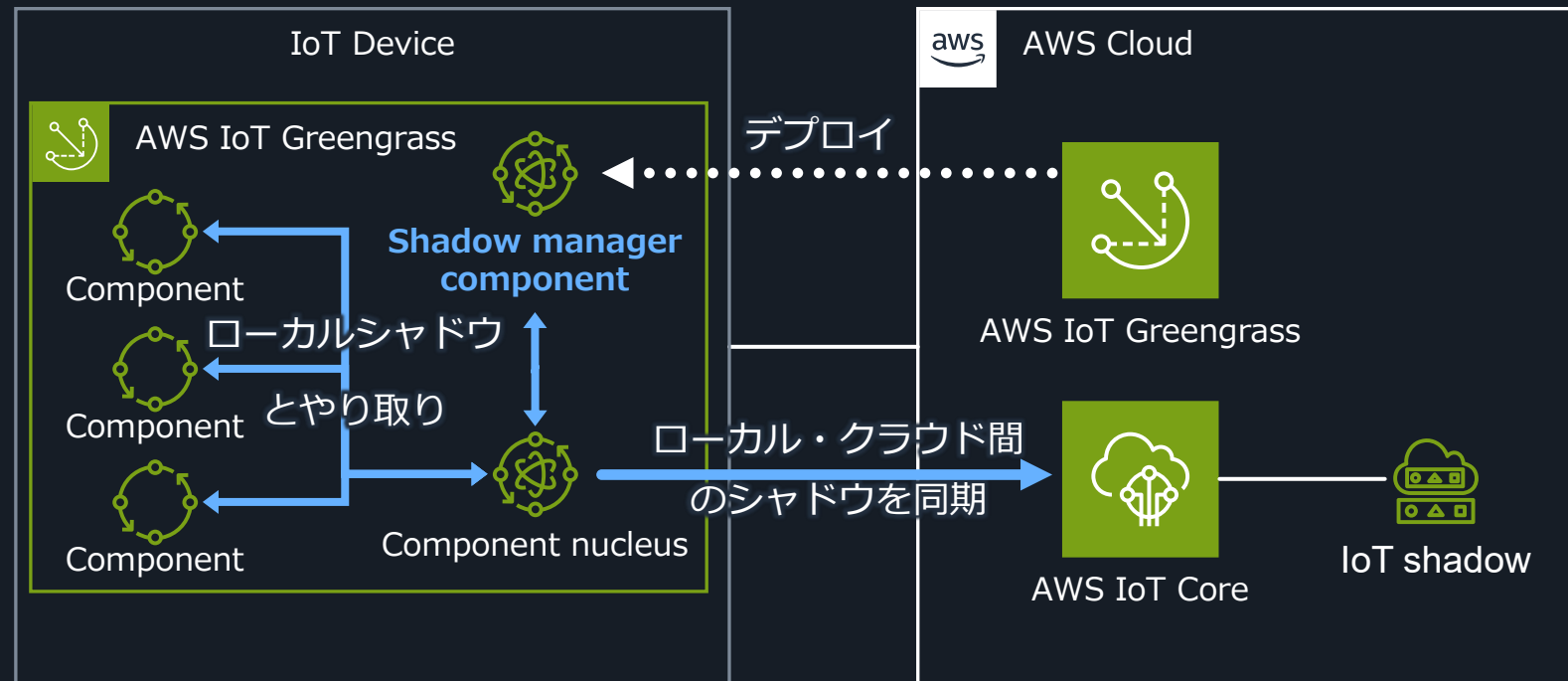
AWS IoT Greengrass のアーキテクチャ概要



シャドウマネージャークOMPONENT

`aws.greengrass.ShadowManager`

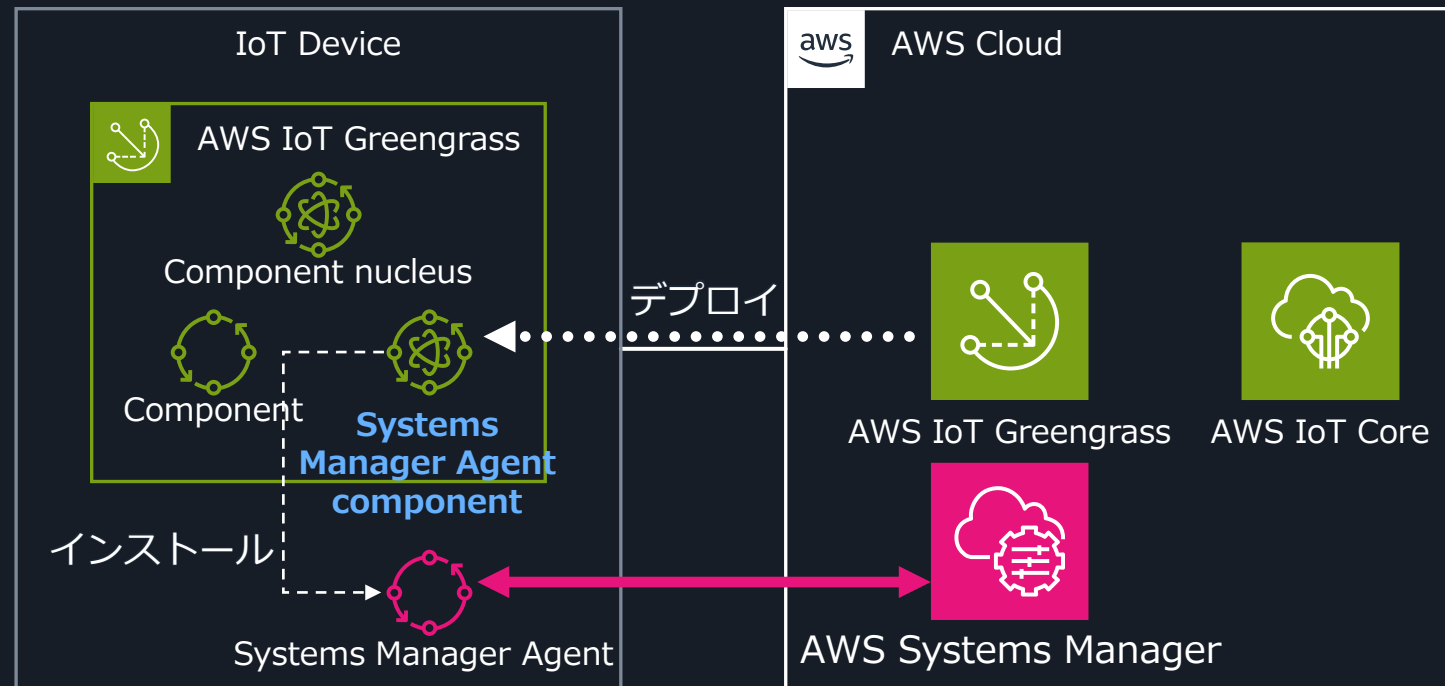
AWS IoT Greengrass コアデバイスでローカルシャドウサービスを有効にし、COMPONENTがプロセス間通信を使用してローカルシャドウとやり取りできるようにします。



Systems Manager エージェントコンポーネント

`aws.greengrass.SystemsManagerAgent`

Systems Manager エージェントをインストールして、コアデバイスを管理できます。Systems Manager は、Amazon EC2 インスタンス、オンプレミスサーバー、エッジデバイスなどのインフラストラクチャを表示および制御するために使用できる AWS サービスです。

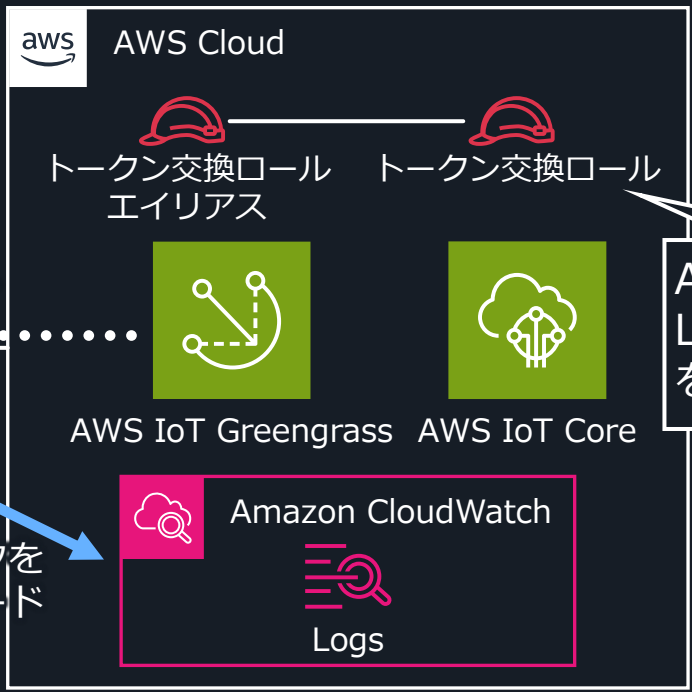
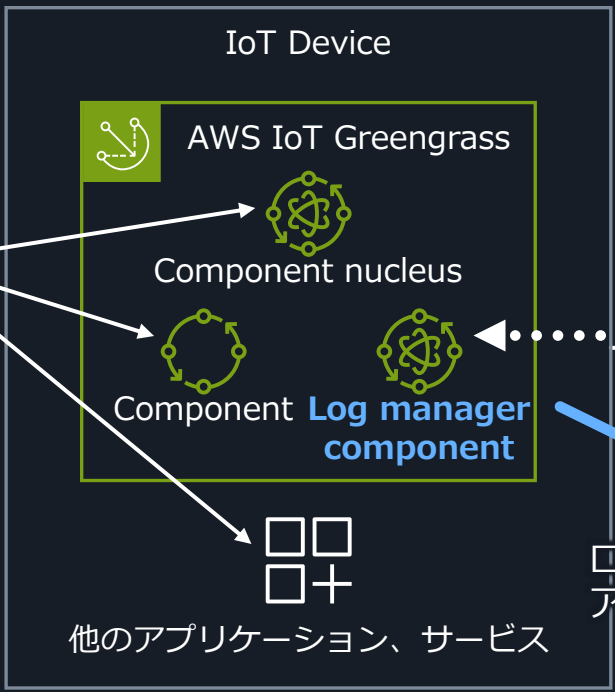


ログマネージャークOMPONENT

aws.greengrass.LogManager

AWS IoT Greengrass コアデバイスから Amazon CloudWatch Logs に、ファイルシステムのログをアップロードします。Greengrass コンポーネントに加え、Greengrass コンポーネントではないアプリケーションのログをアップロードできます。

Greengrass nucleus コンポーネント、その他の Greengrass コンポーネント、Greengrass ではないアプリケーションやサービスのログが対象

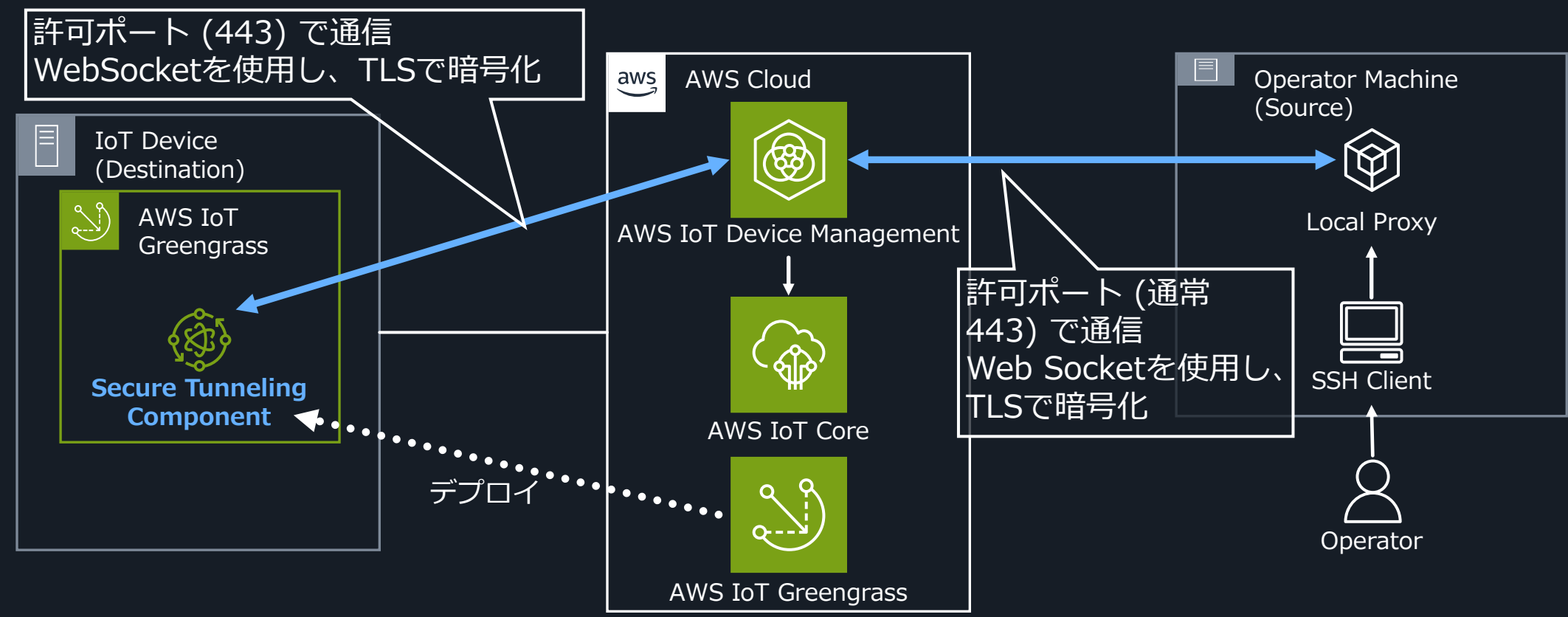


Amazon CloudWatch Logs に書き込むことを許可する必要がある

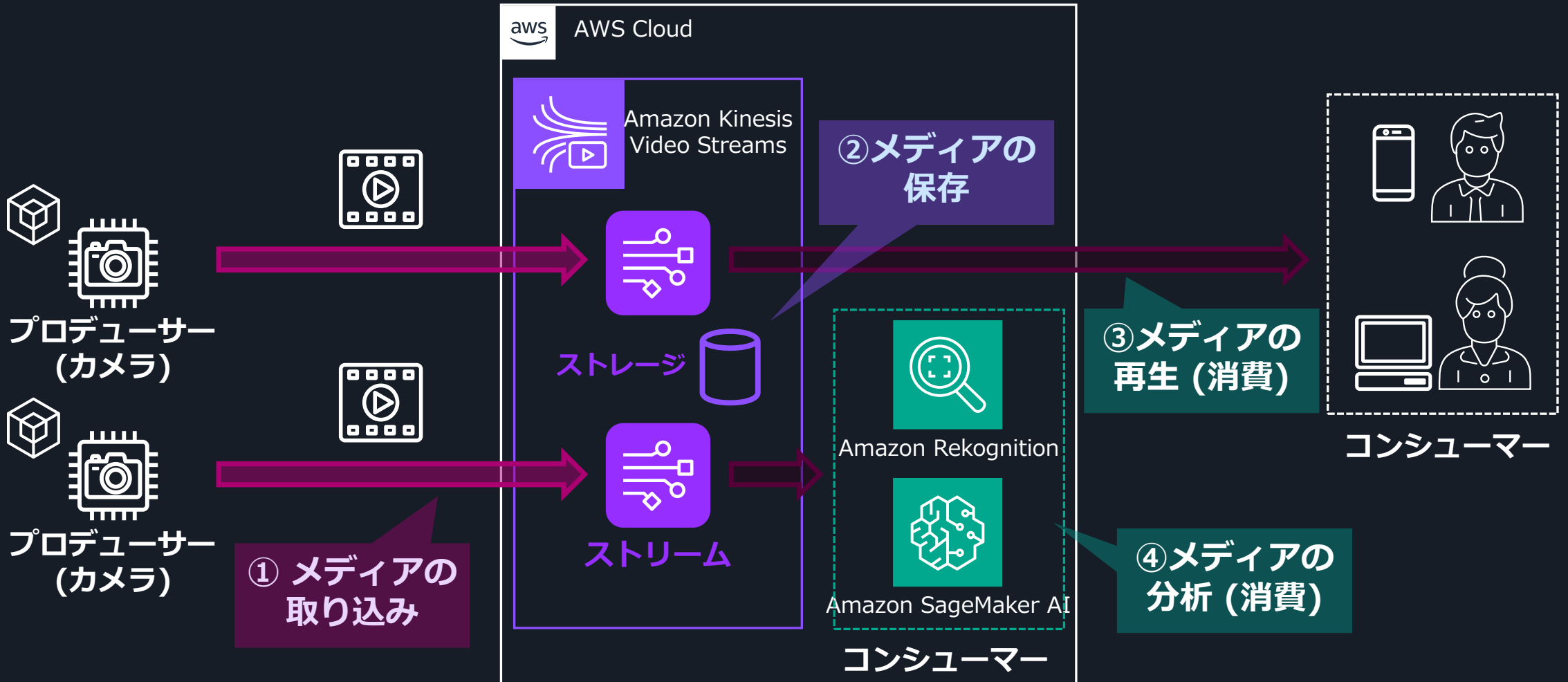
セキュアトンネリングコンポーネント

[aws.greengrass.SecureTunneling](https://aws.github.io/greengrass/SecureTunneling)

セキュアトンネリングを使用して、リモートサイトのポート制限されたファイアウォールの背後にデプロイされているデバイスにアクセスできます。

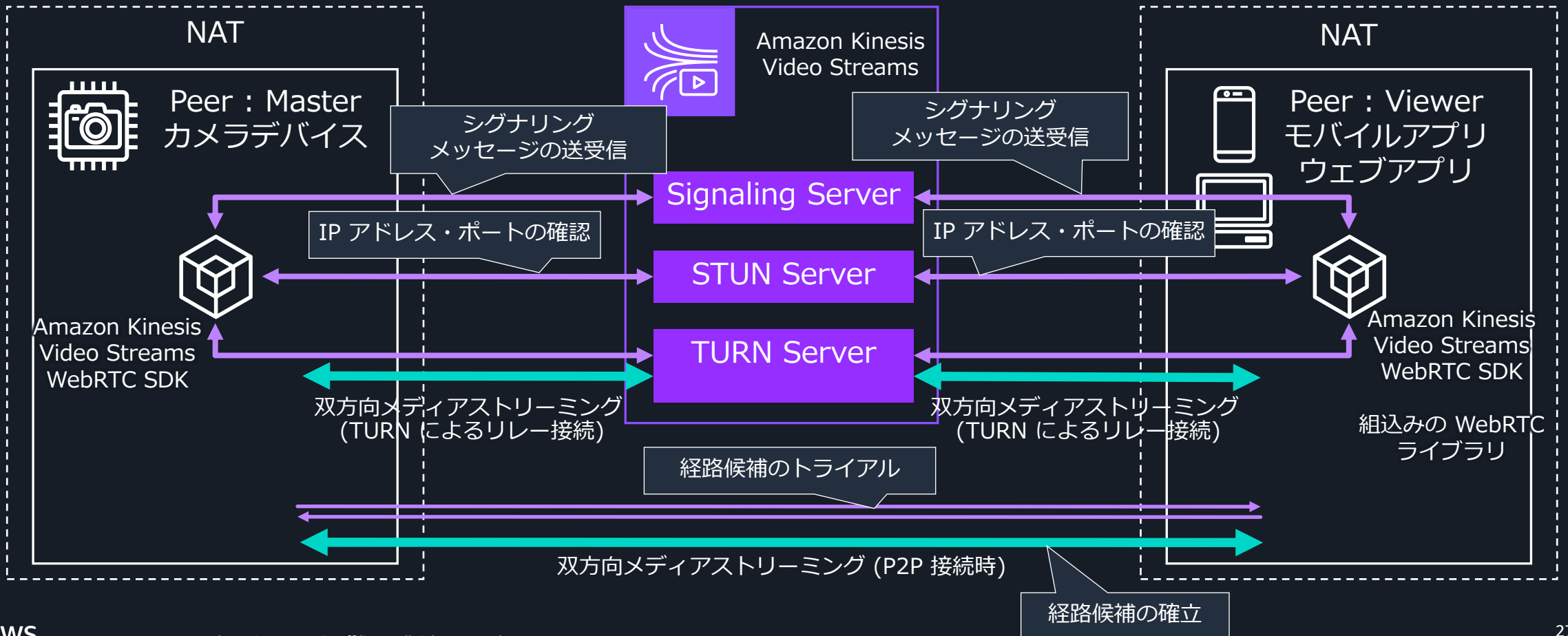


Amazon Kinesis Video Streams – Streams のアーキテクチャの概要



Amazon Kinesis Video Streams – WebRTC のアーキテクチャ概要

Amazon Kinesis Video Streams - WebRTC は標準規格に準拠した
双方向のメディアストリーミングを実現するためのサービス



本資料の入手方法



本資料の入手方法

本資料は、[Amazon Web Services ブログ “AWS Summit Japan 2026 ~AWS IoT サービスを活用した展示の一部をご紹介~”](#) よりダウンロードできます。

Physical AIへの第一歩 AWS IoTで実現するPhysical AI基盤 (A161)

Physical AIの実現には、クラウドで学習したAIモデルをエッジデバイスにデプロイし、エッジデバイスのセンサーデータをクラウドに送信するための安定した基盤が不可欠です。本ブースでは、AWS IoT Core、AWS IoT Greengrass、Amazon Kinesis Video Streams などを用いてデバイス接続からエッジコンピューティング、映像配信を Raspberry PiやJetsonを利用したライブデモで体験いただけます。A/Bデプロイやロールバック等の本番運用機能、生成AI活用に関するご紹介させていただきます。IoT をこれからはじめる方からエッジAI検討中の方まで、Physical AIへの第一歩をここから踏み出してください。

赤枠がブースのおおよその展示場所を示しています。

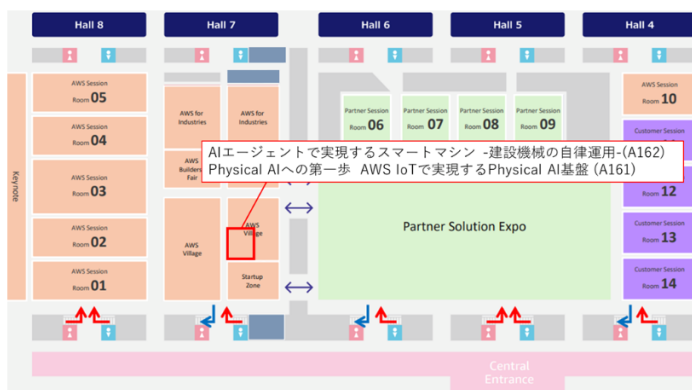


図2：A162 と A026 展示場所

AWS IoT サービスを利用する
他の展示も紹介していますので
参考にしてください



ブログのURL



Thank you