



AI /機械学習に活用できる AWSのエッジソリューションのご紹介

嶺 行伸

ソリューションアーキテクト

デジタルトランスフォーメーション本部

アマゾン ウェブ サービス ジャパン株式会社



自己紹介

嶺 行伸 (みね ゆきのぶ)

デジタルトランスフォーメーション本部

IoTソリューションアーキテクト

IoTの導入やPoC段階のご支援を担当

主に製造系のお客様のご支援をメインに活動中

好きなAWSサービス

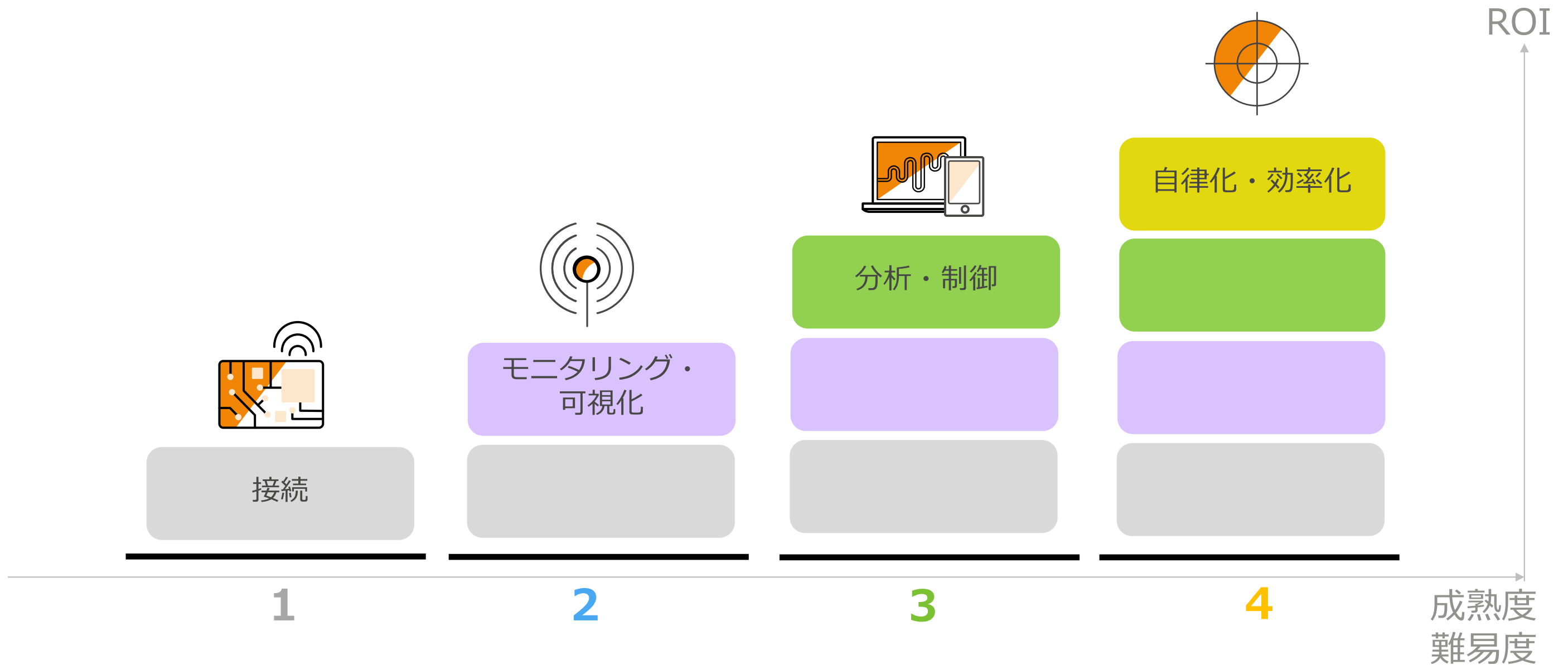
AWS Lambda / AWS IoT Greengrass



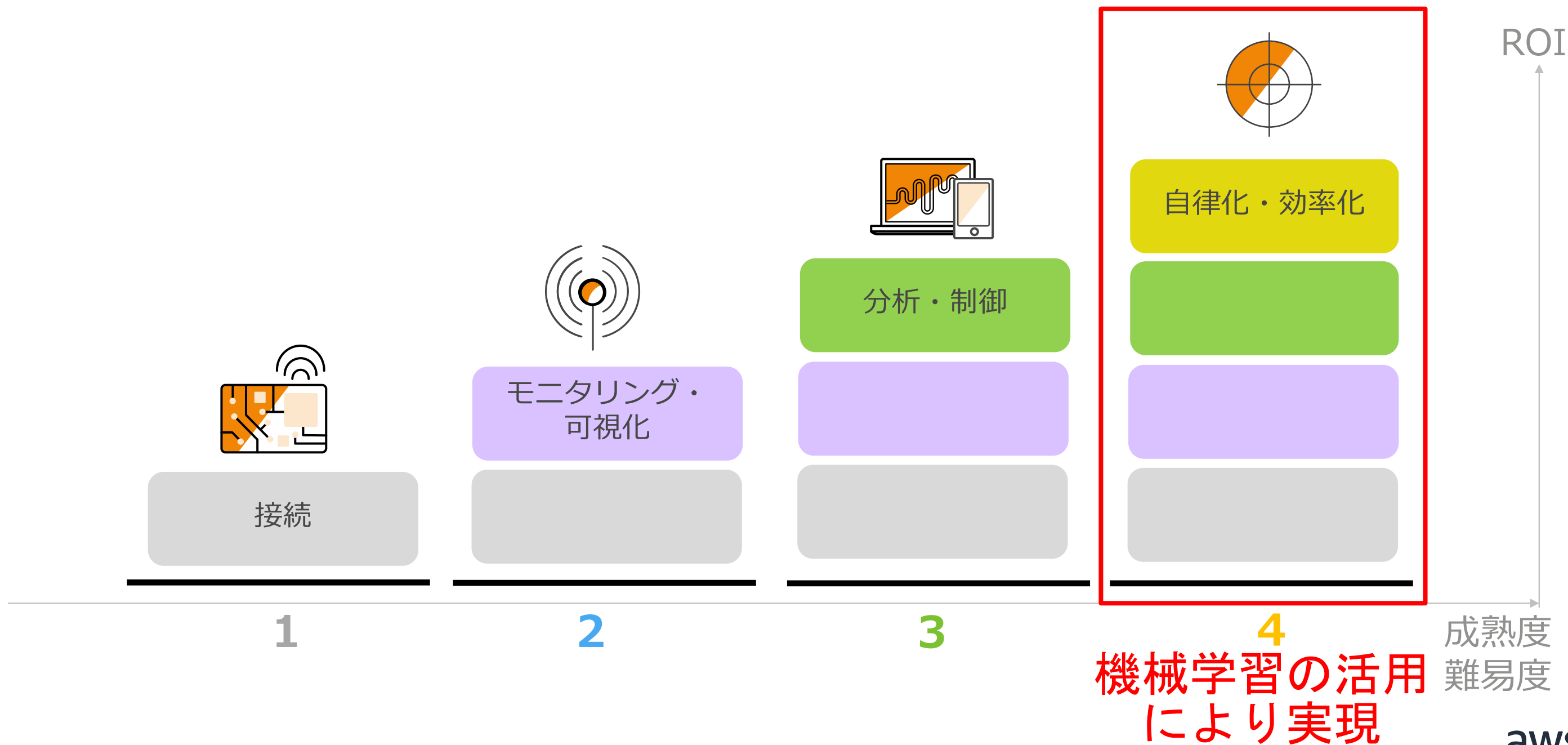
アジェンダ

- IoTにおける機械学習のユースケース
- エッジデバイスでの機械学習が必要なケースと要件
- AWSのエッジソリューションのご紹介
- アーキテクチャ例
- まとめ

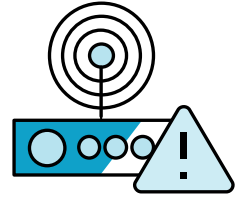
IoTシステムの進化のステップ



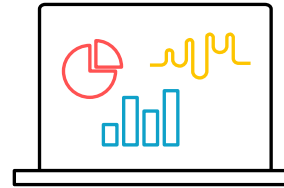
機械学習による自律化がトレンドに



IoTにおける機械学習のユースケース



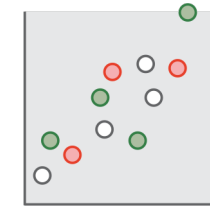
異常検知



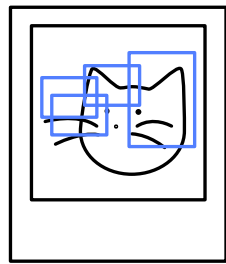
コンテキスト
解析



予知保全



需要予測



外観検査



来客分析



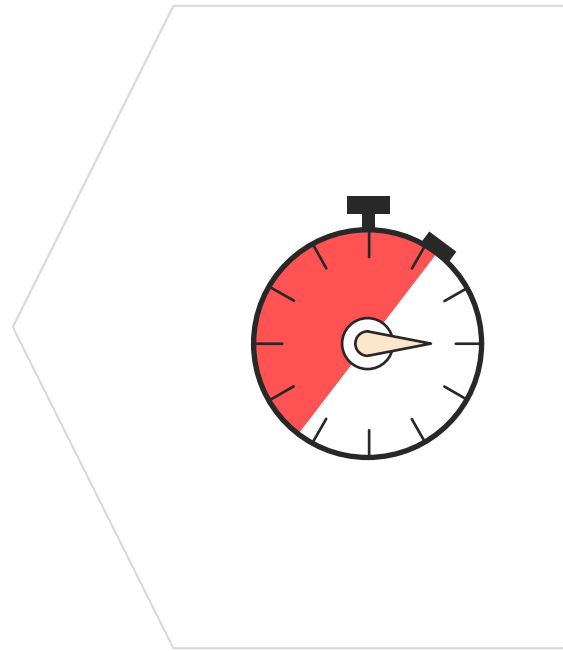
行動分析



顔認証
商品検索

エッジデバイスでの機械学習が必要になるケース

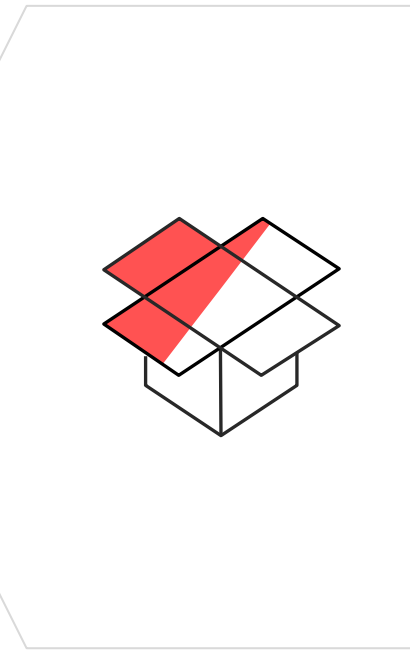
機械学習を導入する際、次の要件がある場合には、推論をクラウドではなく、エッジで行うことを検討する必要がある



レイテンシの制約



通信コスト

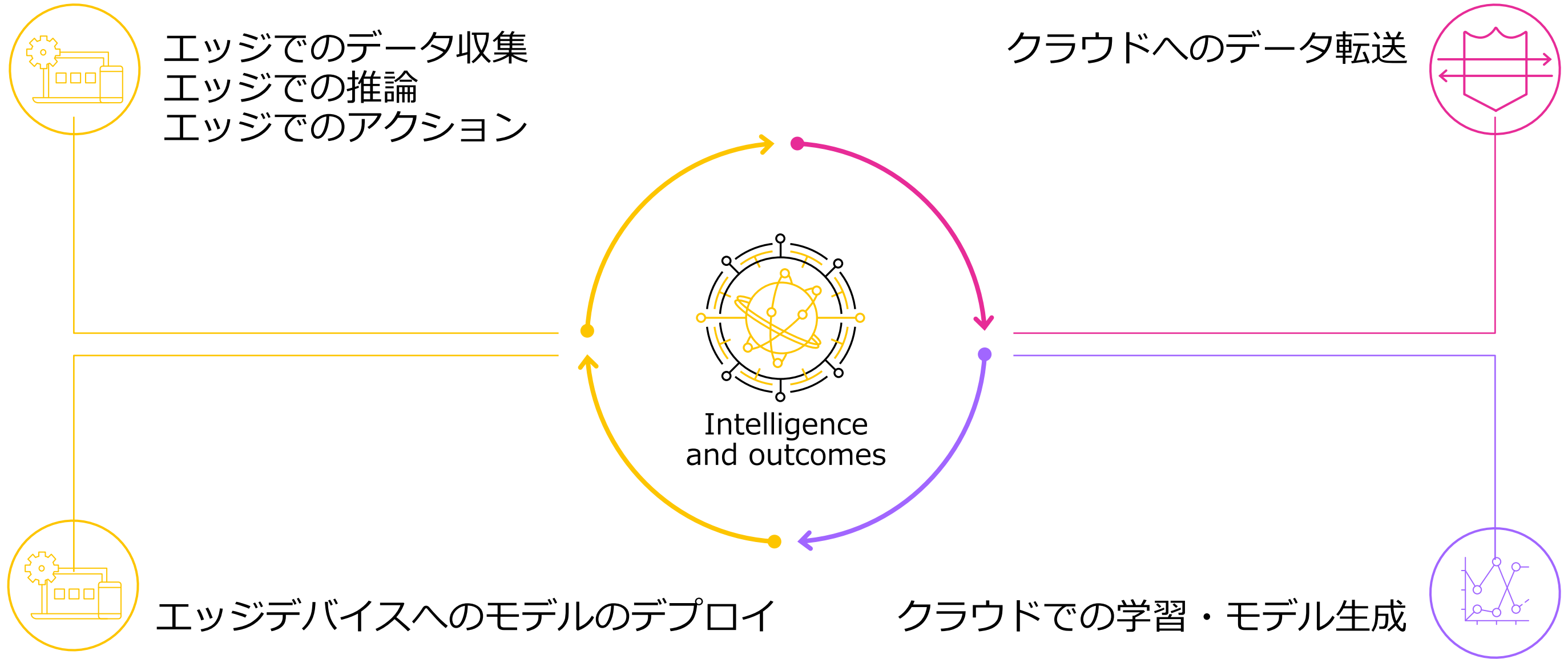


オフライン対応



セキュリティ
コンプライアンス

エッジデバイスでの機械学習に必要な要素





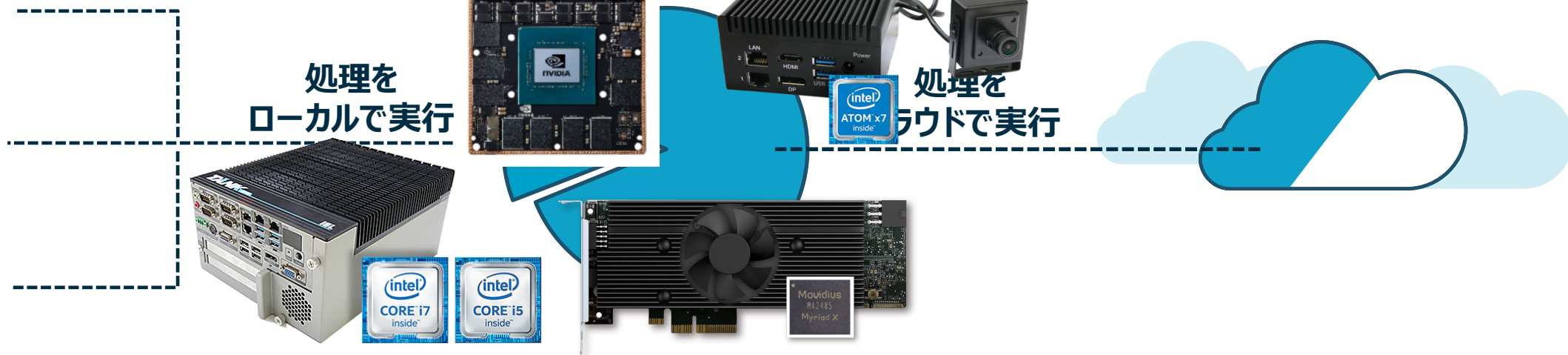
AWS IoT Greengrass

Moving to the edge

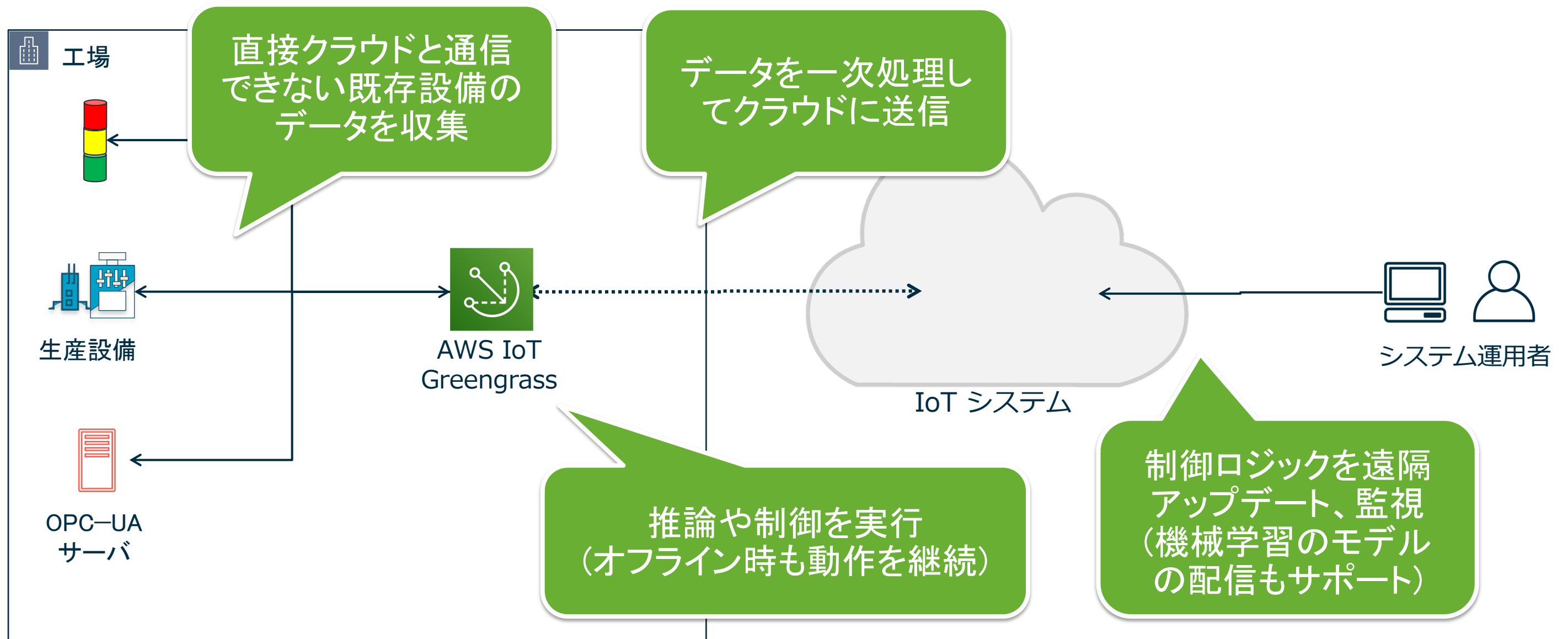
AWSの処理機能をデバイスの上に拡張

任意のデバイスで
処理を実行

開発、アップデート



工場におけるAWS IoT Greengrassの役割



エッジでの既存機器のデータ収集

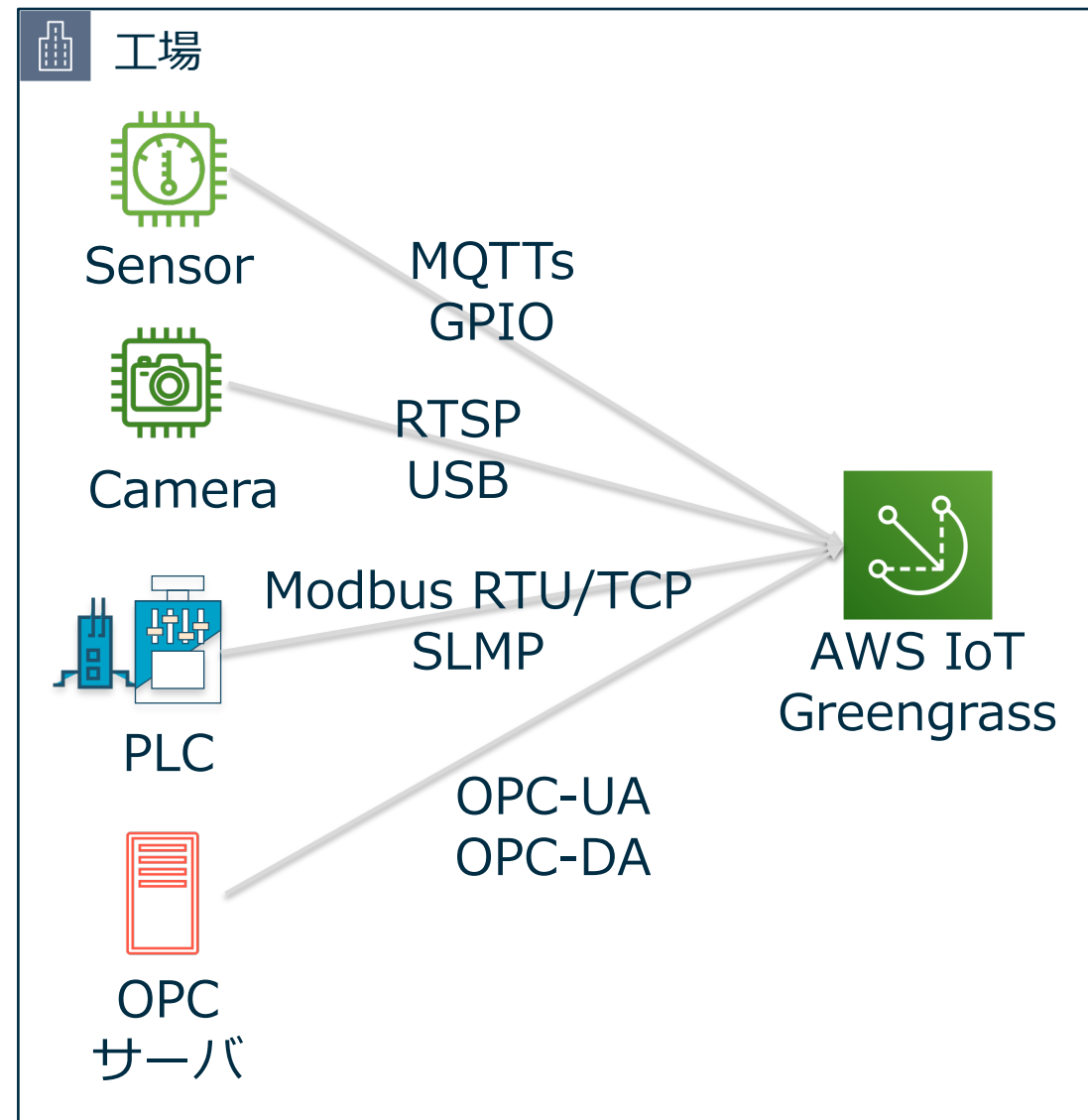
様々なプロトコルで既存機器と通信してデータを収集
広く使われているプロトコルはノンコーディングで利用可能

外付けセンサ

画像、動画

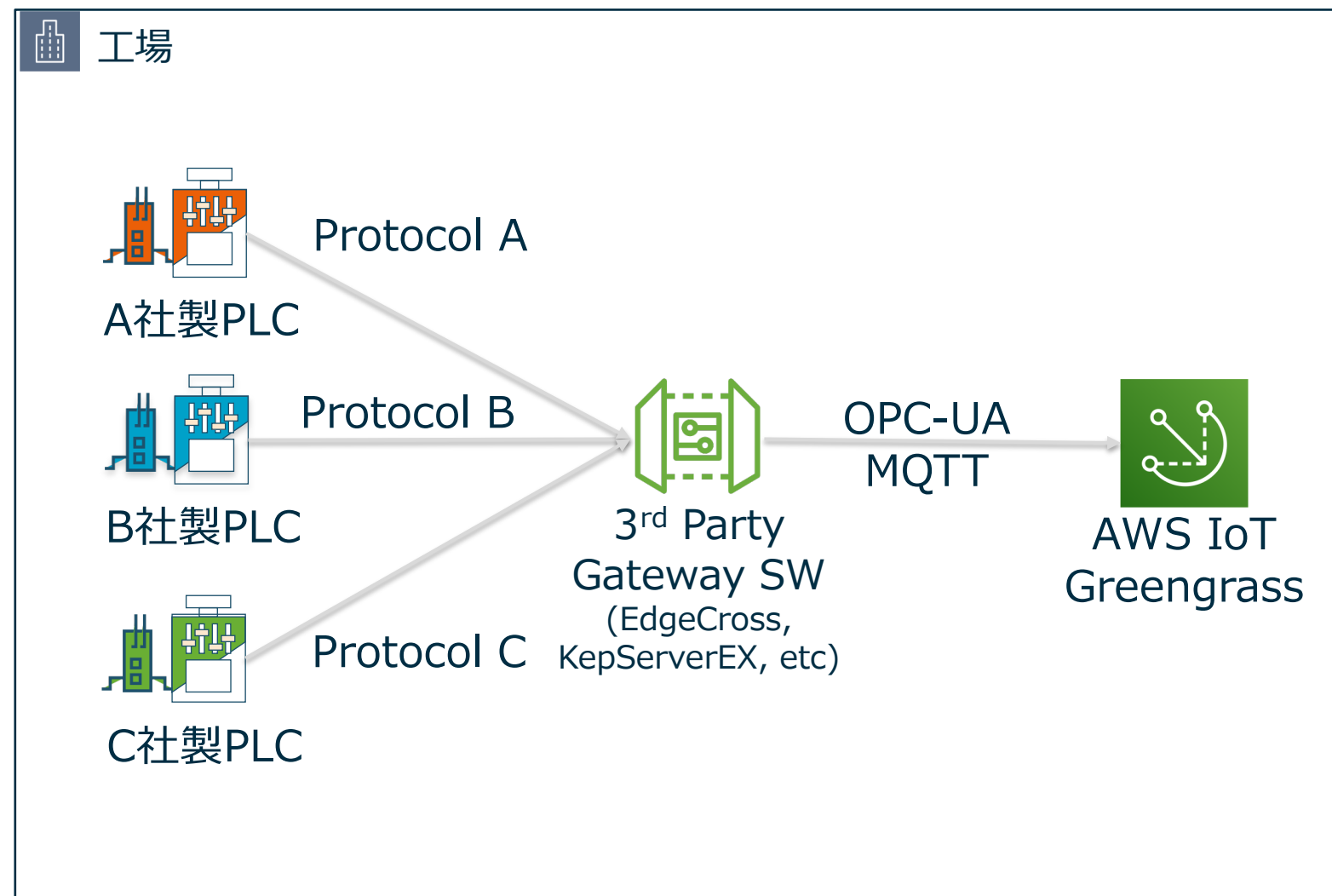
設備データ

集約された
設備データ



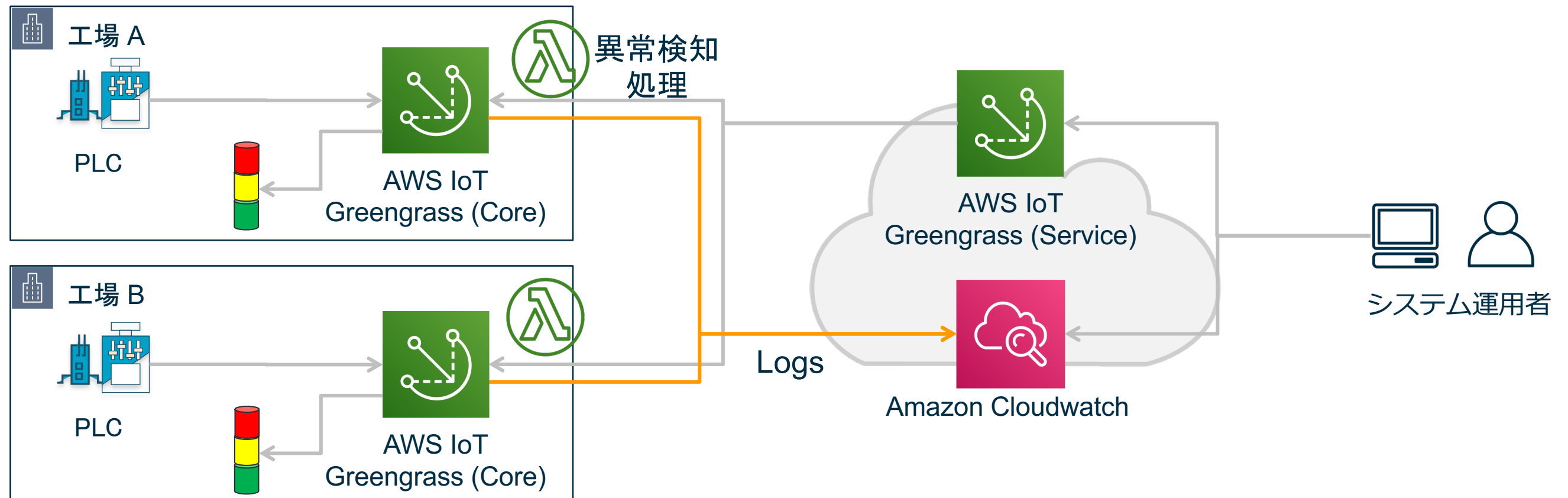
3rd party protocol conversion

各社のPLCのデータを汎用的なプロトコルに変換する
サードパーティーのGW製品との連携も可能



クラウドからプログラムを配信・運用

- AWS Lambda で開発したプログラムを配信してエッジで実行
- 現地に行く事なく機能のアップデート・ログ監視等の運用が可能



クラウドと Greengrass の Lambda 実行環境の違い

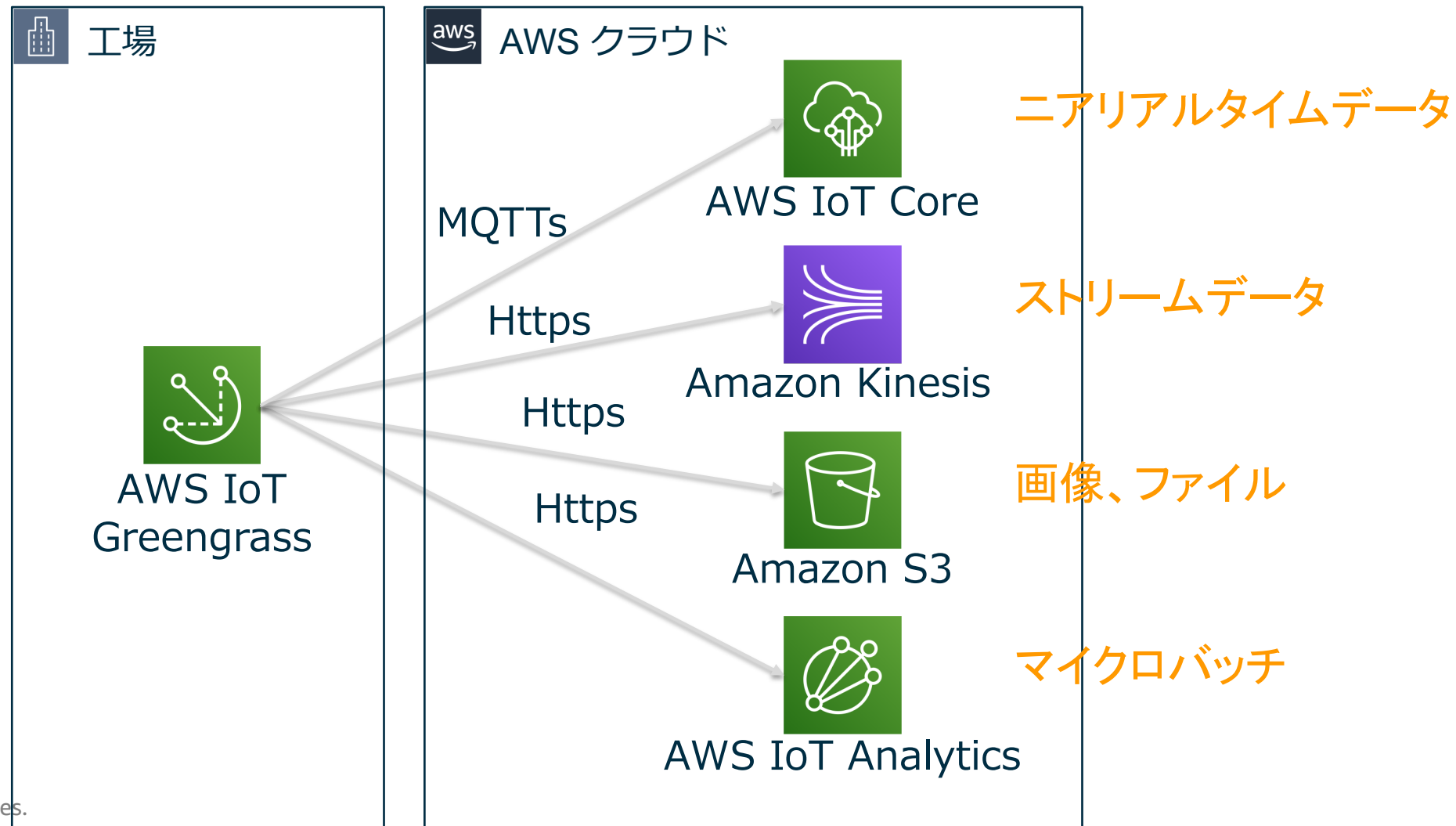
	On Cloud	On Greengrass
実行環境	Amazon Linux	GGC がインストールされているデバイスに依存
プログラミング言語	Node.js Java Python .NET Core and more...	Node.js: v6 / v8 / v12 Python: 3.7 / 2.7 Java: 8 C, C++
イベントソース	S3, DynamoDB, Kinesis, ...(*1)	Local MQTT, GGC 起動時に自動起動
タイムアウト設定	最大 900s	上限なし (常駐を指定可能)
メモリ設定	最小 128MB 最大 3008MB	上限なし (デバイスに依存)
ローカルリソースアクセス (デバイス or ボリューム)	不可	可 (ホワイトリスト方式)
課金	実行回数と実行時間、割り当てたメモリ量 に応じて課金	無料(*2)

*1: https://docs.aws.amazon.com/ja_jp/lambda/latest/dg/invoking-lambda-function.html

*2: Greengrass 自体の課金は発生しません

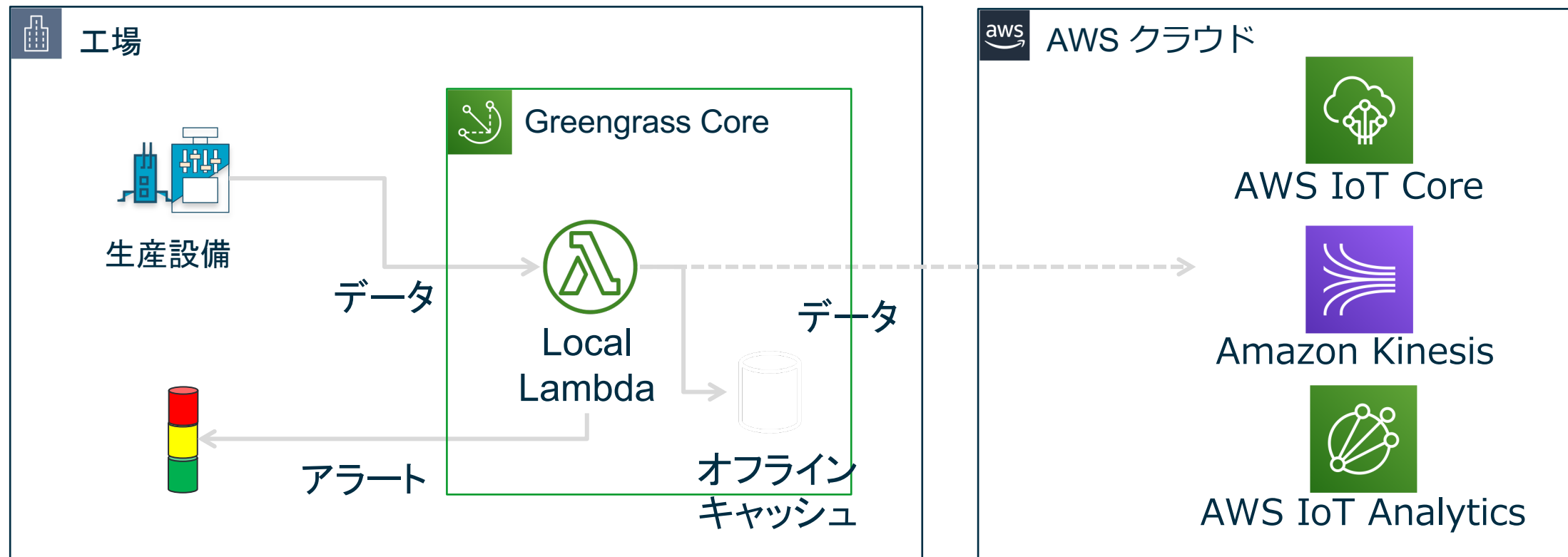
クラウドへのデータ送信

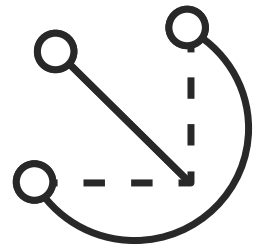
- データレイクでの分析やモデルの再学習のため、クラウドにデータを自動送信
- データの種別に応じて適切なサービスを選択可能



オフライン対応

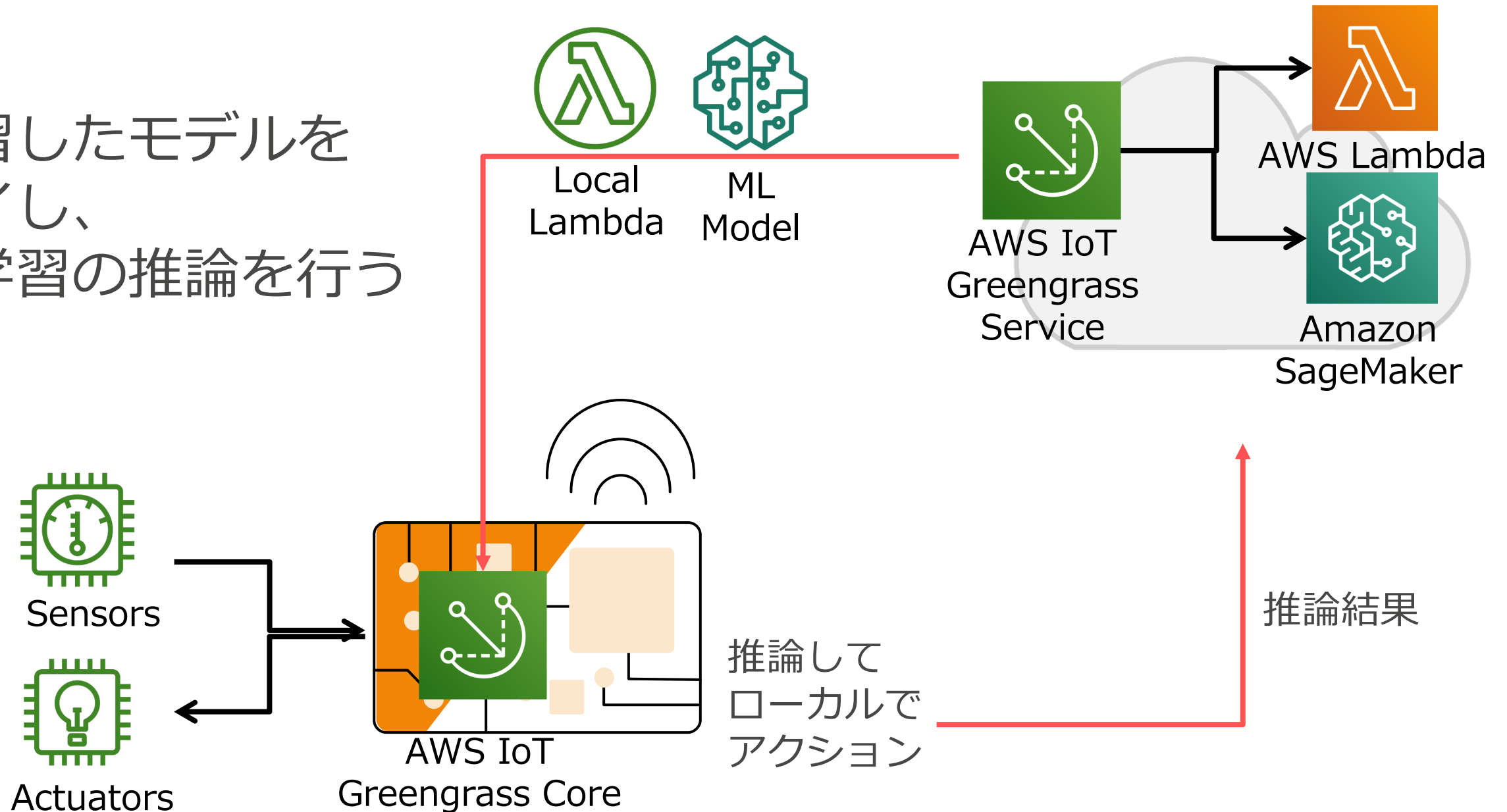
- オフラインになってもエッジでの処理は継続
- クラウドにアップロードすべきデータはキャッシュして、オンラインになったタイミングで順次アップロード



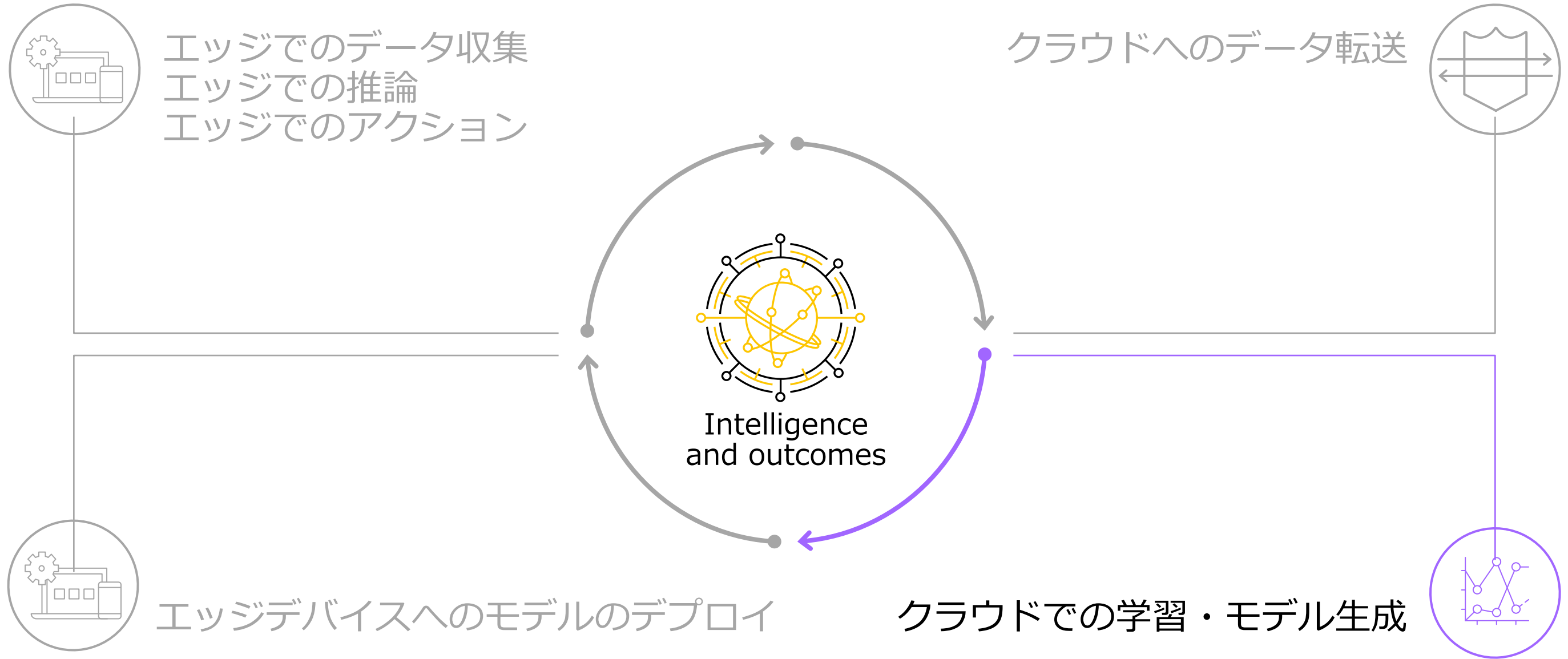


AWS IoT Greengrass ML Inference

クラウドで学習したモデルを簡単にデプロイし、エッジで機械学習の推論を行う



エッジデバイスでの機械学習に必要な要素

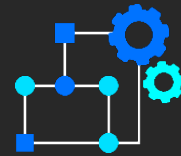


Amazon SageMaker

データサイエンティストや開発者が
容易に機械学習モデルを開発・活用するための
マネージドサービス



よくあるシナ
リオ向けの
ノートブック
の提供

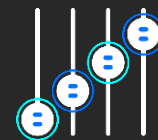


高いパフォーマ
ンスのビルトイ
ンアルゴリズム

開発



One-click
トレーニング

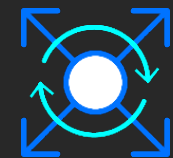


ハイパーパラメー
タの最適化

学習



One-click
デプロイメント

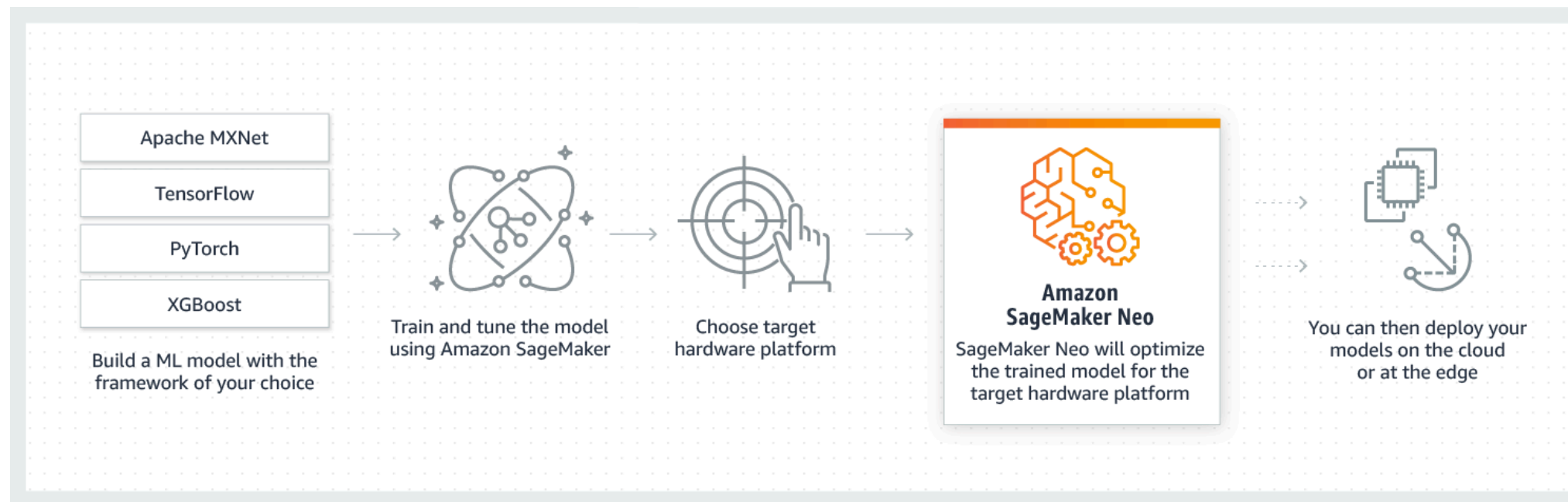


フルマネージドのホ
スティング
& 自動スケール

推論

Amazon SageMaker Neo

- Tensorflow や PyTorch などのモデルを、 EC2 インスタンスや Greengrass デバイス上で高速に動作するように変換するサービス
- 従来のDeep Learning フレームワークが 500MB-1GB 程度であるのに対し、 Amazon SageMaker Neo Runtime は 1MB 程度



Amazon SageMaker Ground Truth

- データにラベル (Ground Truth) を付与するアノテーションを支援
- 以下の4タスクにはテンプレートが用意されており、自作も可能
- ラベルを付与するワーカーは、Amazon Mechanical Turk、外部ベンダ、自社のチームの3種類から選択可能

画像のラベル



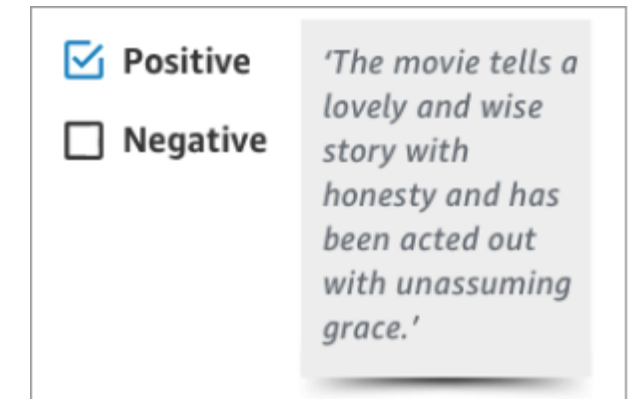
ラベルと位置



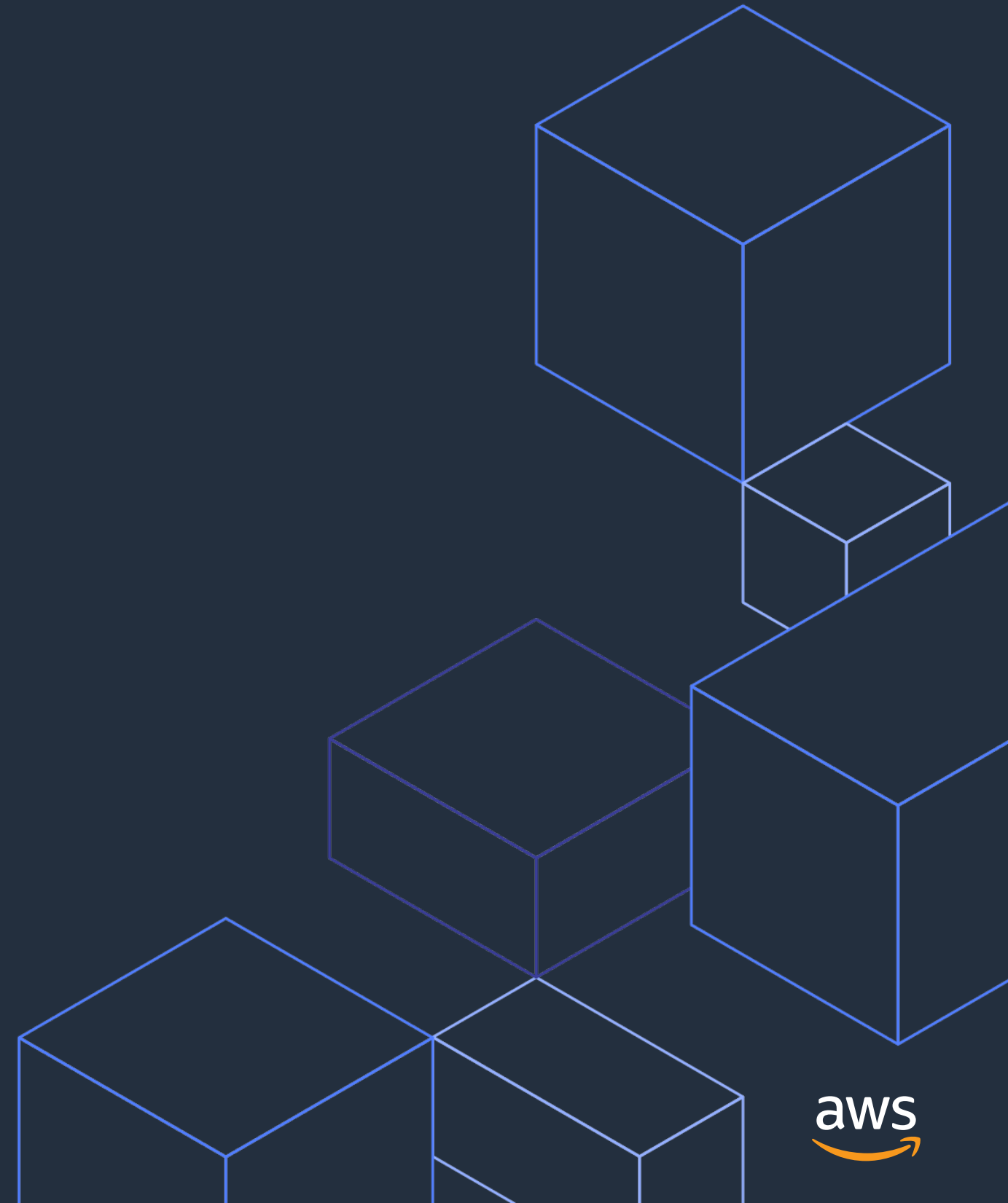
ピクセル単位のラベル



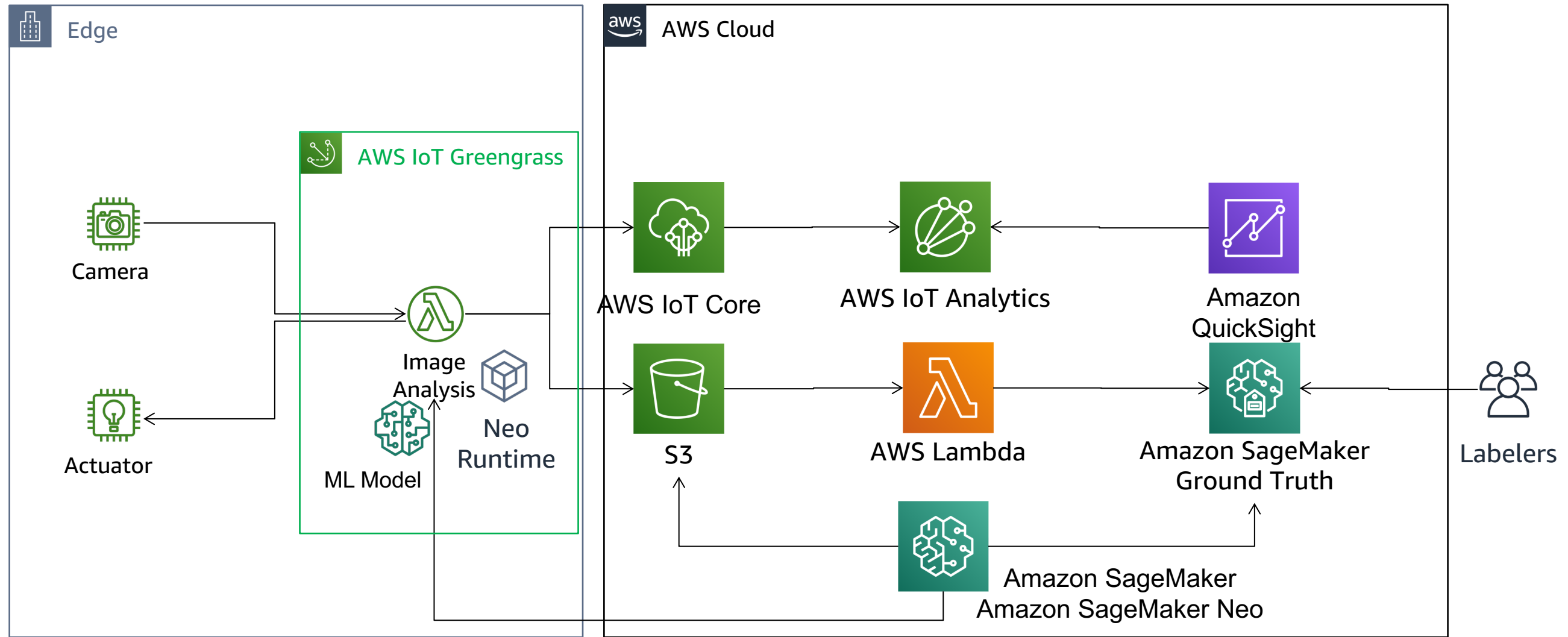
文章のラベル



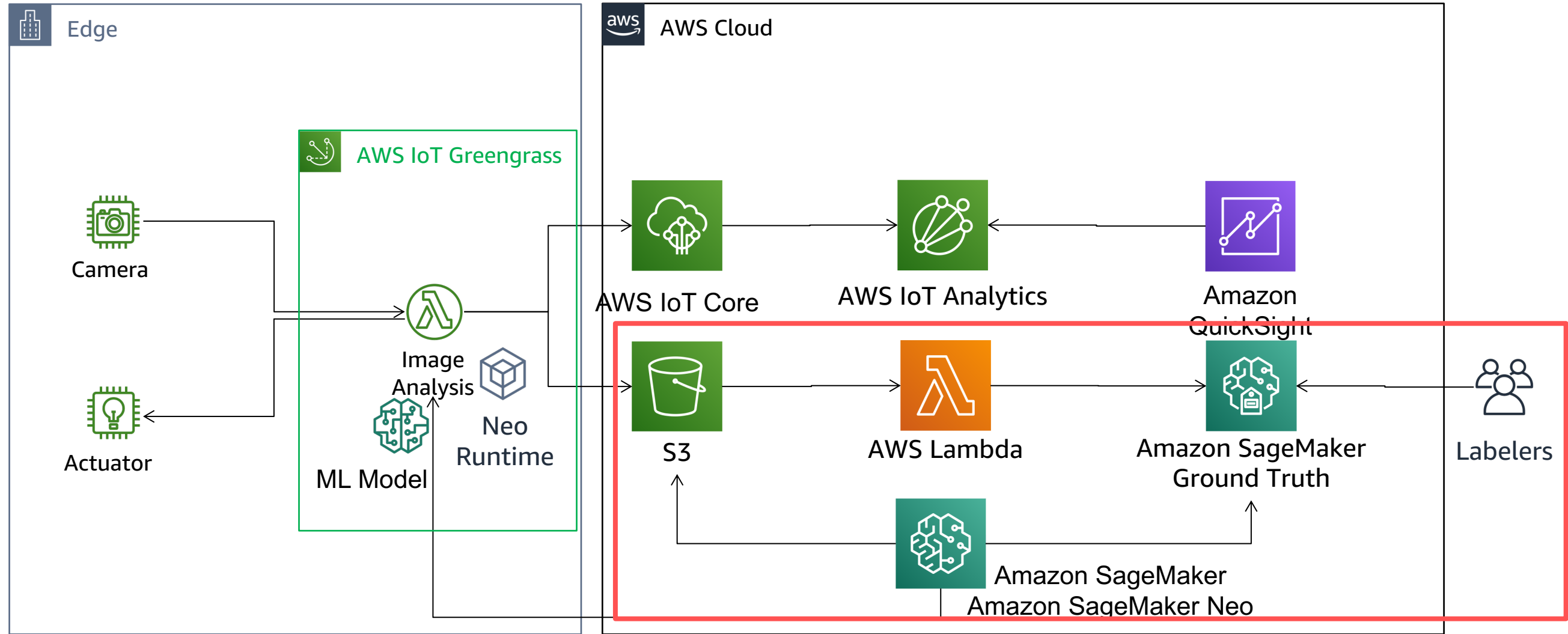
アーキテクチャ例



エッジ推論のアーキテクチャ例

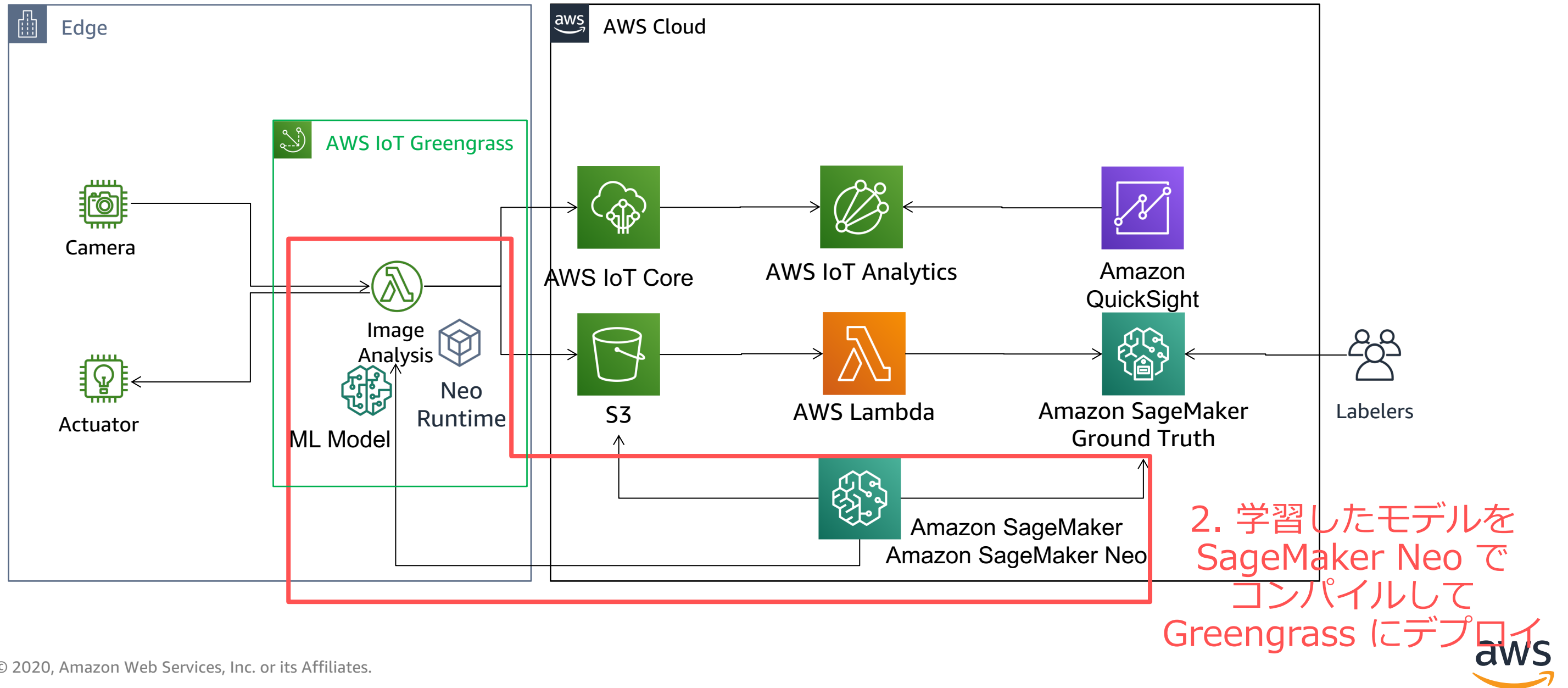


エッジ推論のアーキテクチャ例

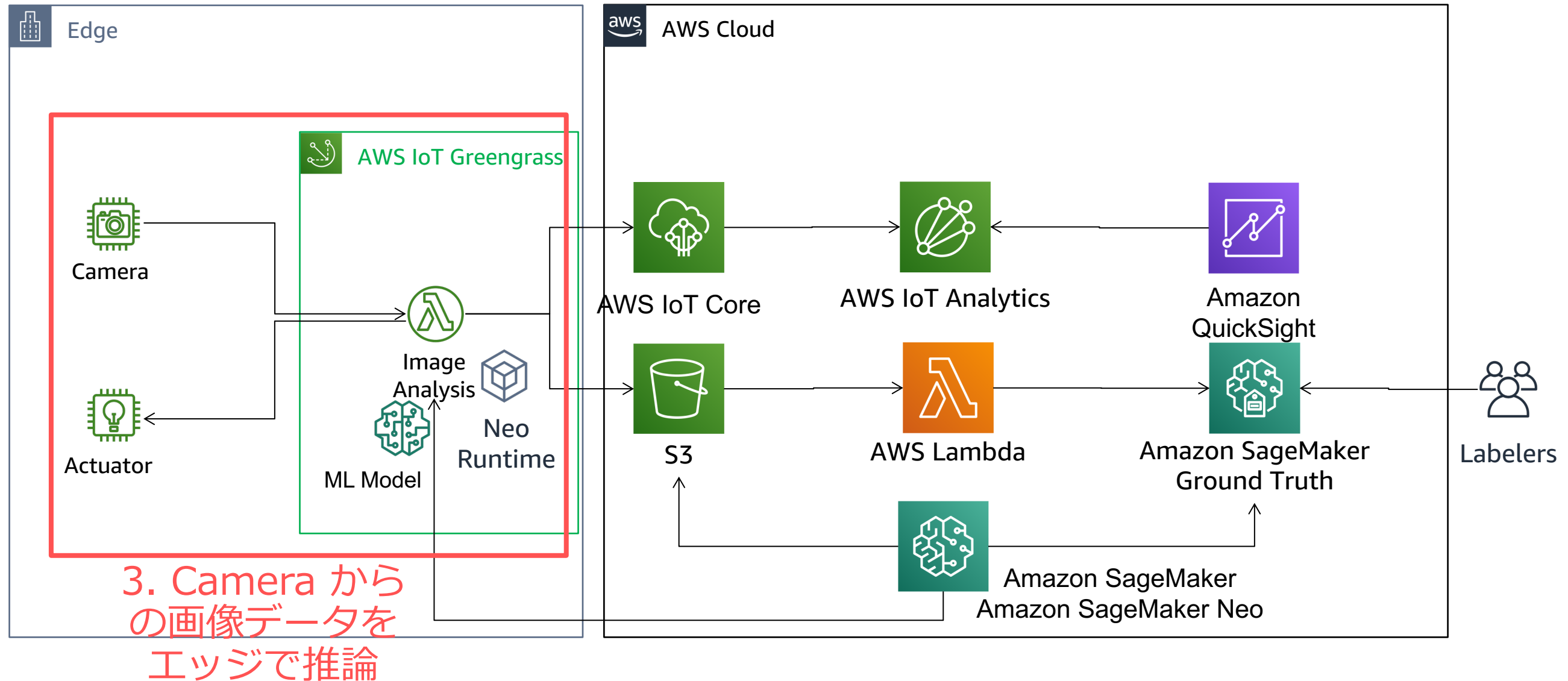


1. 蓄積したデータと SageMaker Ground Truth で付与したラベルから学習してモデルを生成

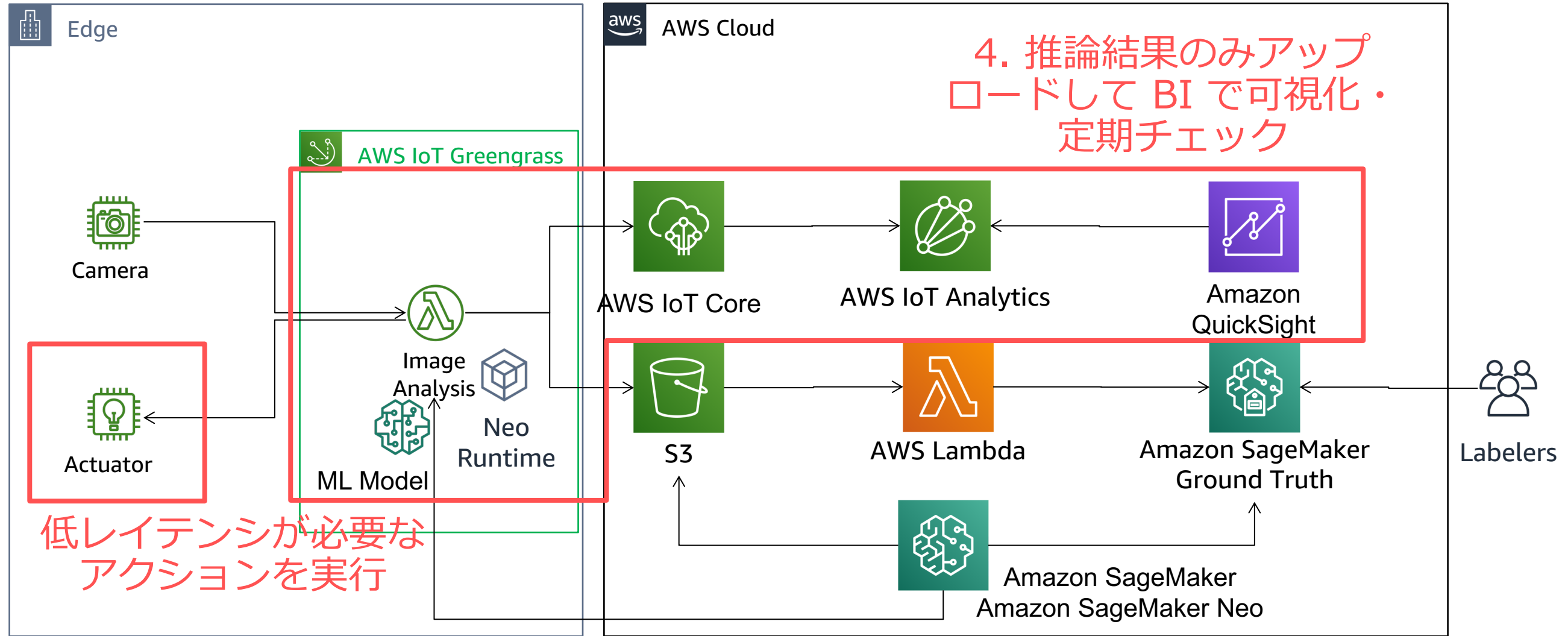
エッジ推論のアーキテクチャ例



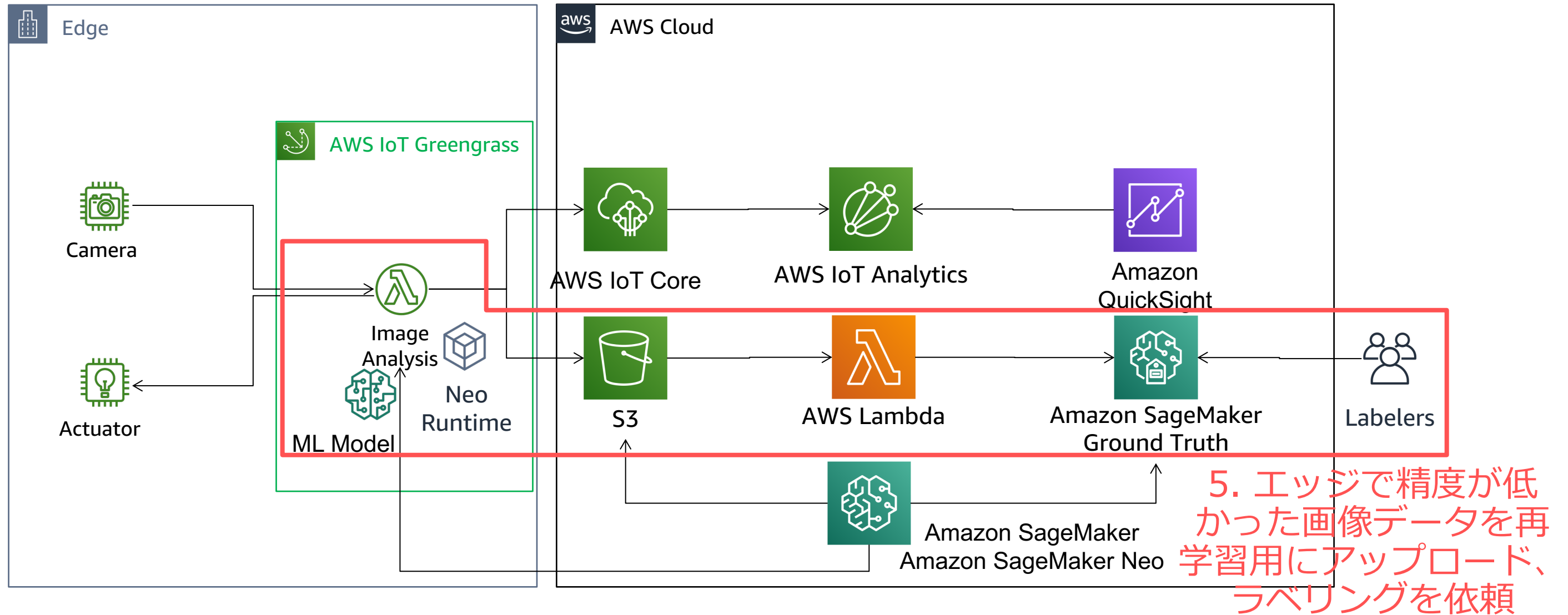
エッジ推論のアーキテクチャ例



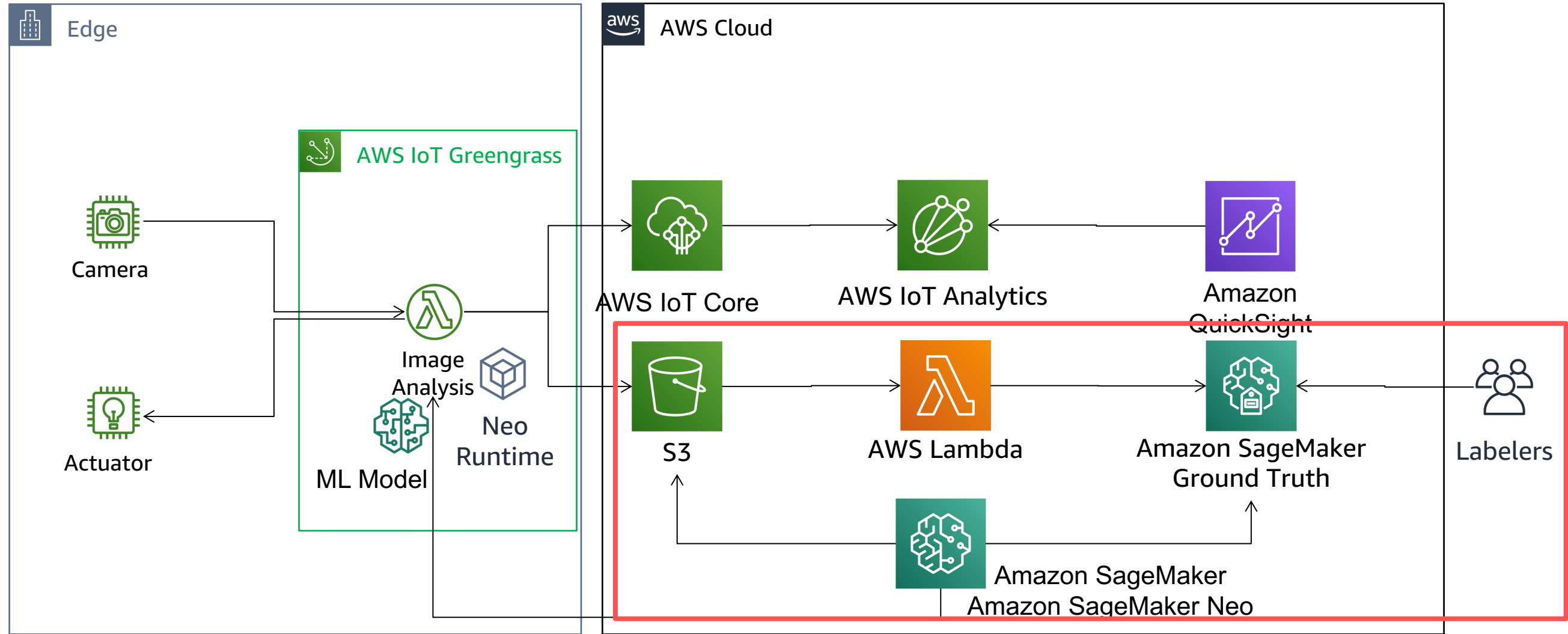
エッジ推論のアーキテクチャ例



エッジ推論のアーキテクチャ例



エッジ推論のアーキテクチャ例



6. 精度が低かった画像と正しいラベルを含めたデータセットで再学習することで精度改善

まとめ

- 外観検査など、現場課題への機械学習ソリューションの導入にはエッジ推論が適している
- エッジ推論では、エッジデバイスでのデータ収集と推論、データや推論結果のクラウドへの送信、クラウドでの推論モデル開発と精度改善、エッジへのデプロイ、といったサイクルを構築する事が重要
- AWSでは、サイクルの構築に必要なサービスとエッジソリューションを全て提供している