aws SUMMIT

1-H1-3-16

# エッジデバイス上での機械学習の活用

Mitsuaki Nakata

Solutions Architect Amazon Web Services Japan

Shuhei Sonoda

Solutions Architect Amazon Web Services Japan

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws SUMMIT

# 自己紹介

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# 自己紹介

中田 光昭

技術統括本部

**IoT/AI Solution Builder**

機械学習を担当

園田 修平

技術統括本部

**IoT/AI Solution Builder**

エッジコンピューティングを含む
**IoT** 全般を担当

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Agenda

- エッジデバイス上での機械学習の活用について
- エッジデバイス上での機械学習の活用アーキテクチャ
- AWS サービスの活用
- デモ
- まとめ

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws SUMMIT

# エッジデバイス上での機械学習の活用について

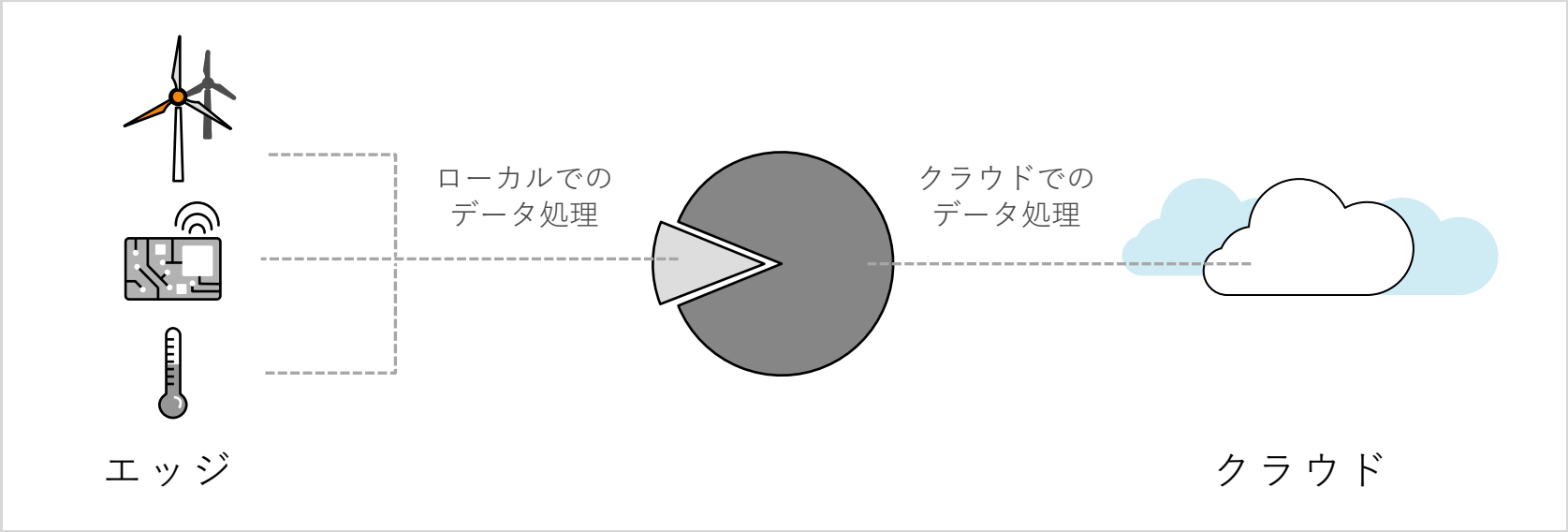
© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# エッジコンピューティングとは？

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

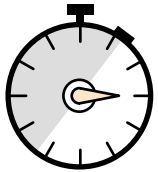
# エッジコンピューティングとは

クラウドと異なる場所にコンピューティングリソースと意思決定機能を備える力



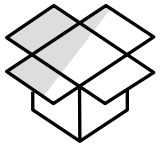

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# エッジコンピューティングが必要となるケース



レイテンシ

帯域幅



オフライン

セキュリティ

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# エッジデバイス上での機械学習の活用シーン

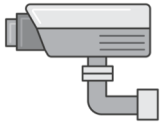
© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# エッジデバイス上での機械学習の活用シーン

自動運転

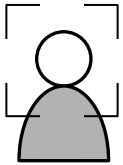
スマート農業

予兆保全



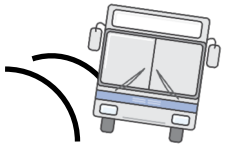
監視カメラ

産業ロボット

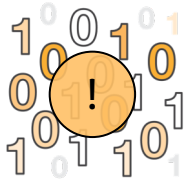


画像認識

音声認識



衝突回避



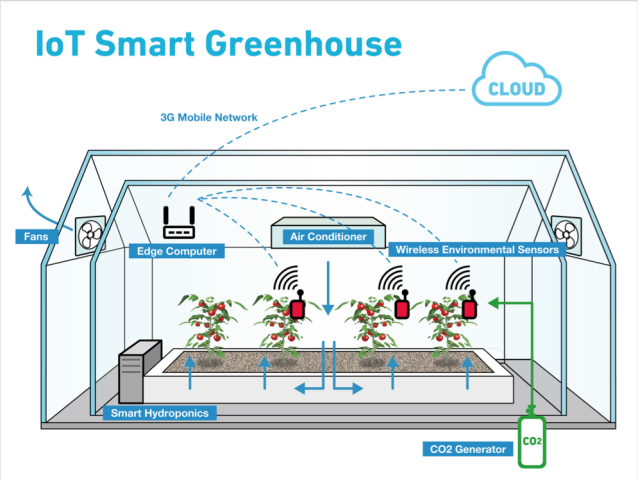

異常検知

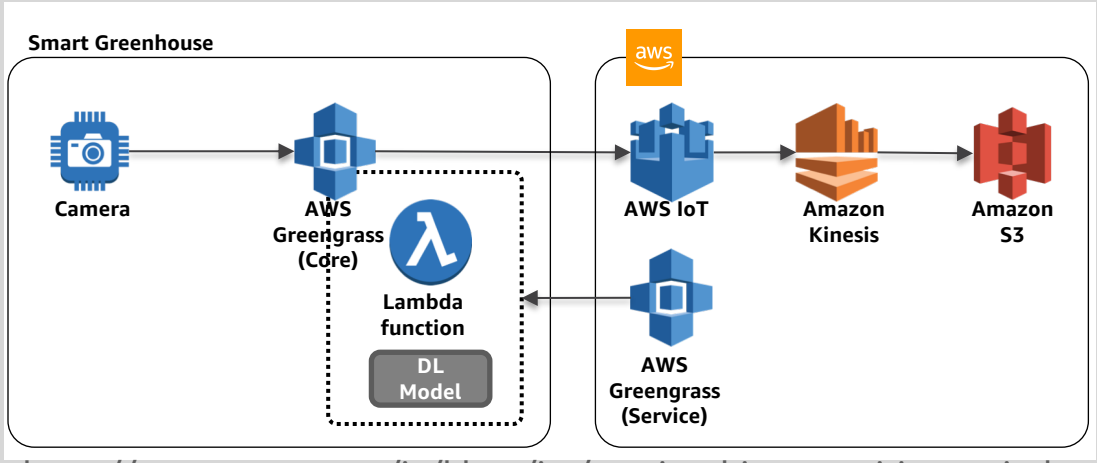
More

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# AWS 利用例: Yanmar IoT Smart Greenhouse

- センシングデータから作物への水やりをエッジで判断することにより自動化、一時的なオフラインにも対応
- 作物の検出や成熟度の判別をエッジで行うことによりネットワーク帯域の問題を解決




https://aws.amazon.com/jp/blogs/iot/aws-iot-driven-precision-agriculture/

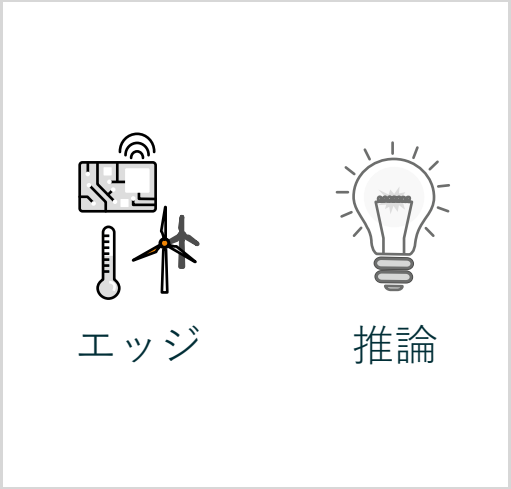
© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

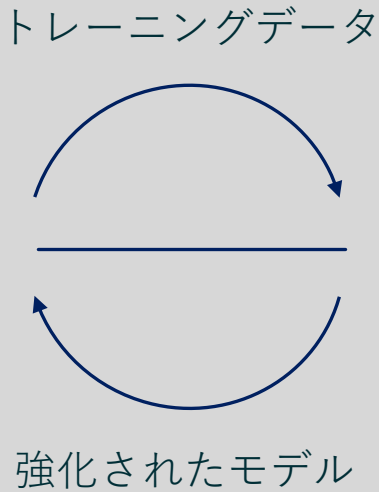
aws SUMMIT

# エッジデバイス上での機械学習の活用アーキテクチャ

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# 基本的なアーキテクチャ

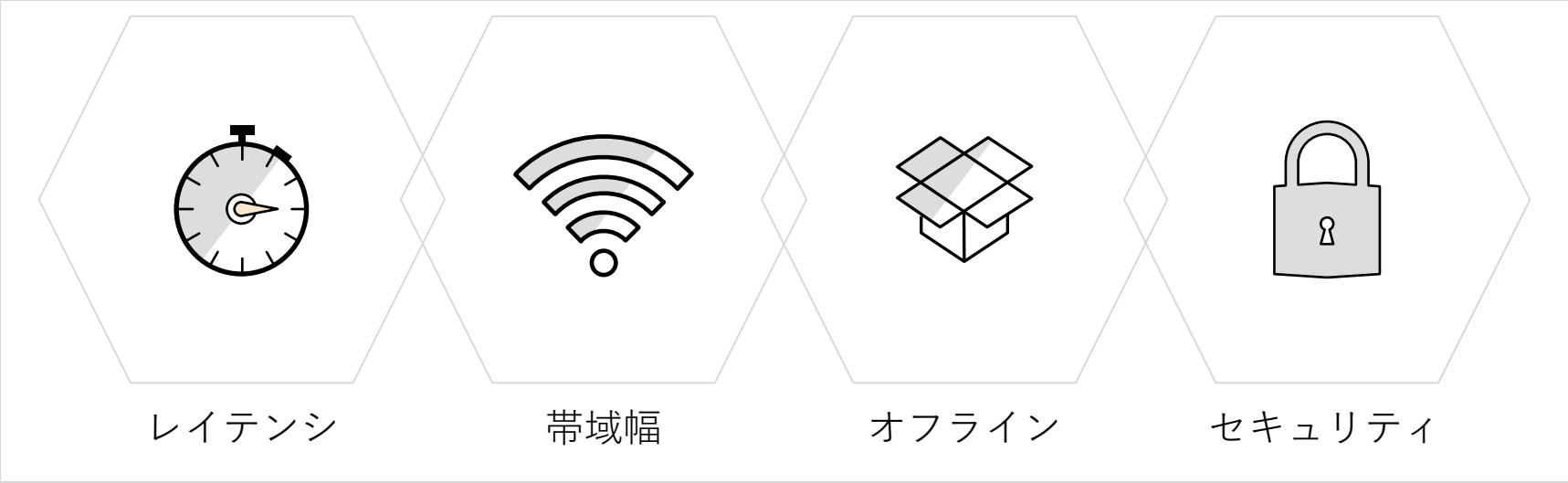


© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# アーキテクチャ設計上の考慮点

- 低レイテンシー、通信データ量の削減、オフライン動作、などエッジ側のメリットは多くあるが、いくつかアーキテクチャ設計上の考慮点がある




© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# アーキテクチャ設計上の考慮点

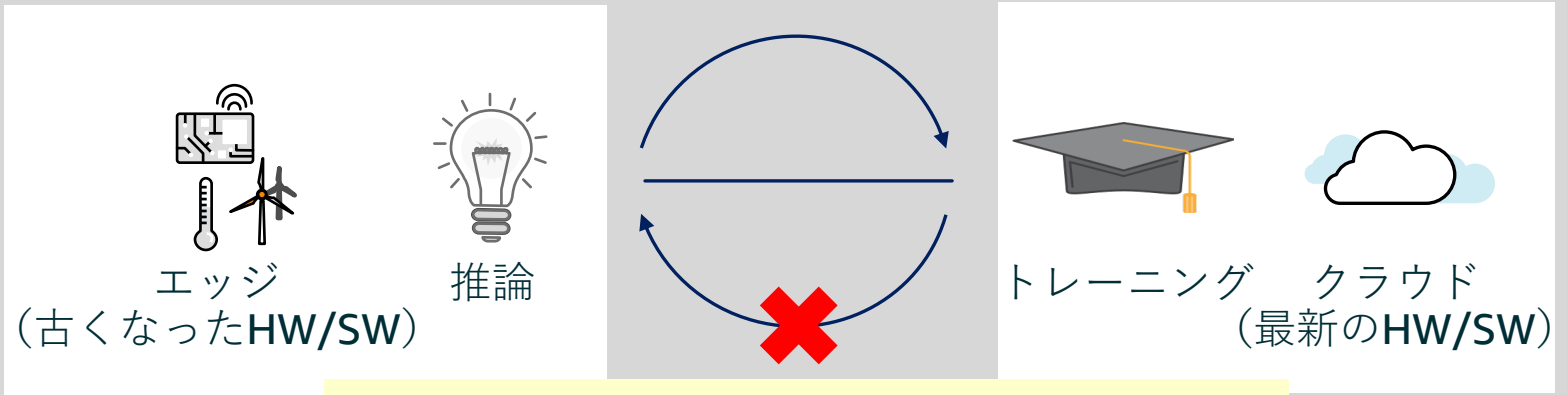
© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# エッジ側の考慮点

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# デバイスのライフサイクル

- デバイスはライフサイクルが長くなることが多い
  - デバイス展開後に HW/SW (OS) の更新を行うが難しい場合があり、常に最新の技術を利用できるわけではない
    ⇒クラウドと連携した機能拡張ができるようにしておくと良い

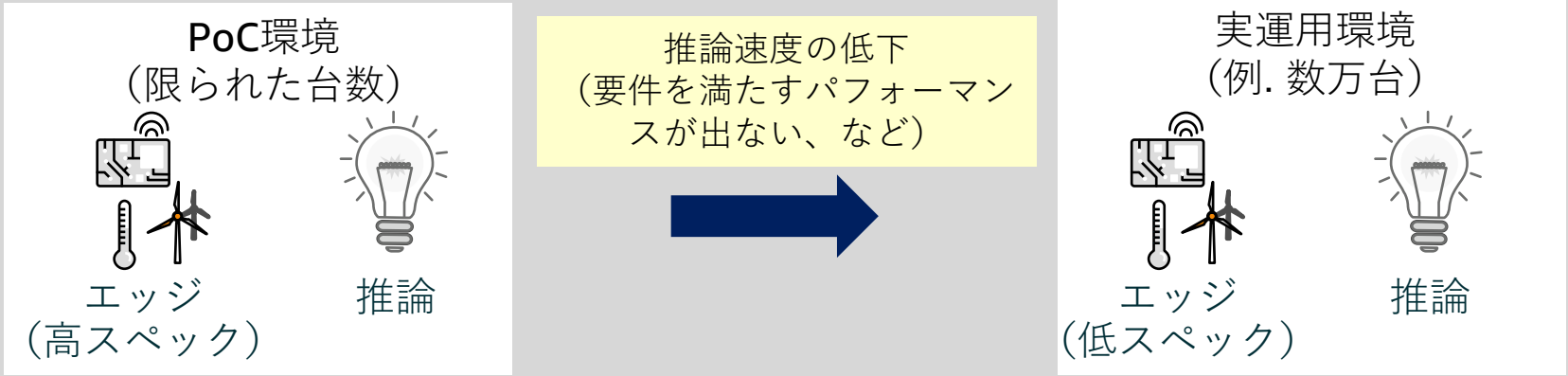



最新技術で作ったモデルが利用できない。など

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# デバイスの HW スペック

- 実運用環境においては必ずしも高スペックな HW が利用できるとは限らない
  - 高スペック ＝ 高価格、既設の HW を利用しないといけない場合もある
    ⇒クラウドと連携した機能拡張ができるようにしておくと良い



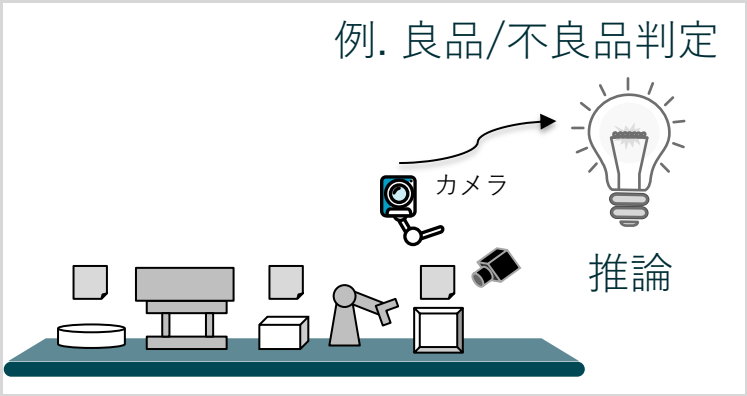

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

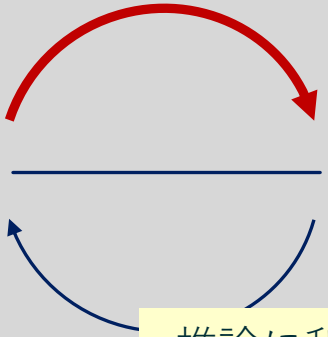
# モデルのライフサイクルの考慮点

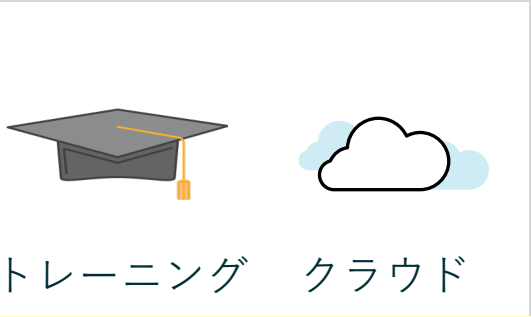
© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# モデルのパフォーマンスのモニタリング

- モデルの推論結果のパフォーマンスを継続的にモニタリングしていく必要がある
  - エッジ側のみで処理を完結させてしまうと、推論結果の正しさを後で評価することができない
    ⇒推論に利用したデータをクラウドに収集し、定期的に評価する



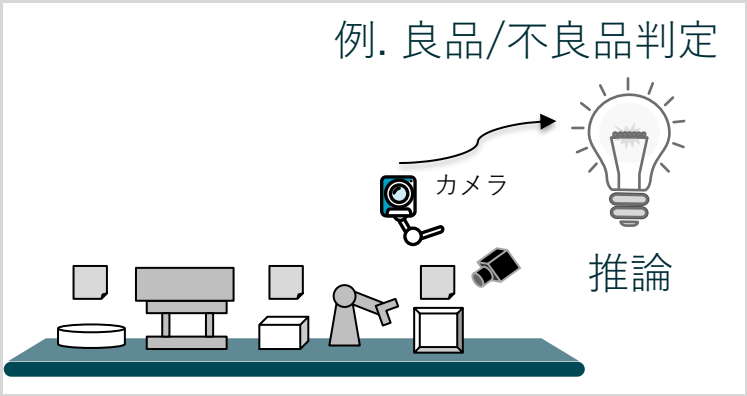


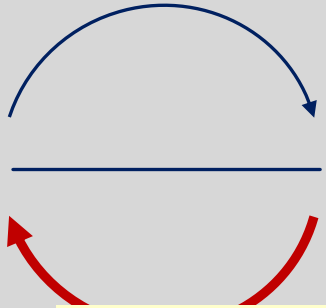

推論に利用したデータをクラウドで収集・評価

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# モデルの配信

- モデルの効率の良い配信の仕組みを用意する必要がある
  - 分類項目の追加、精度の向上など、モデルには定期的に更新していく可能性があるが、エッジへのモデルの配信ができなければ活用できない
    ⇒クラウドからエッジデバイスへのモデルの配信機能を用意



クラウドで更新したモデルをエッジデバイスに配信

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# アーキテクチャ設計上の考慮点
# まとめ

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# アーキテクチャ設計上の考慮点：まとめ

- エッジ側の考慮点
  - HW ライフサイクル、スペックなどを考慮し役割分担を設計。必要に応じてクラウド側で機能を補ってやると良い
- モデルのライフサイクルの考慮点
  - 推論自体はエッジで行う場合でも、推論結果の精度のモニタリングは必要。推論で利用したデータをクラウドで収集し、モデルの精度をモニタリングする必要がある
  - モデルは定期的に更新が入る可能性があるため、エッジデバイス側に効率よく配信する仕組みが必要

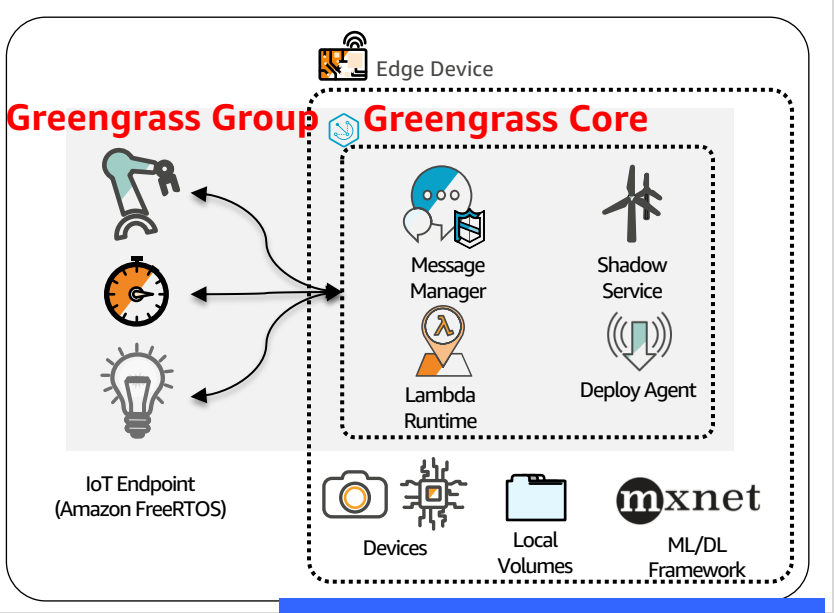
© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws SUMMIT

# AWS サービスの活用

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# AWS サービスを活用したアーキテクチャ

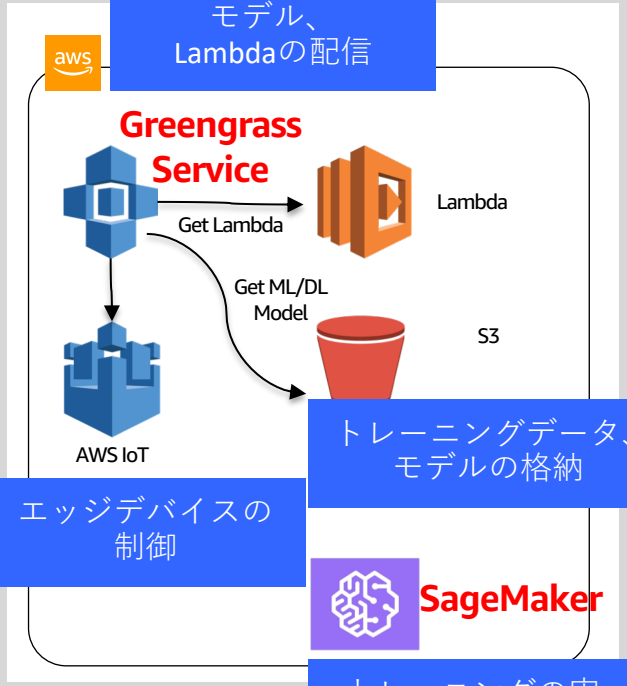



推論の実行
推論結果に基づくアクション
利用データのアップロード

トレーニングデータ

強化されたモデル

モデル、
Lambdaの配信




エッジデバイスの
制御

トレーニングデータ、
モデルの格納

SageMaker

トレーニングの実施

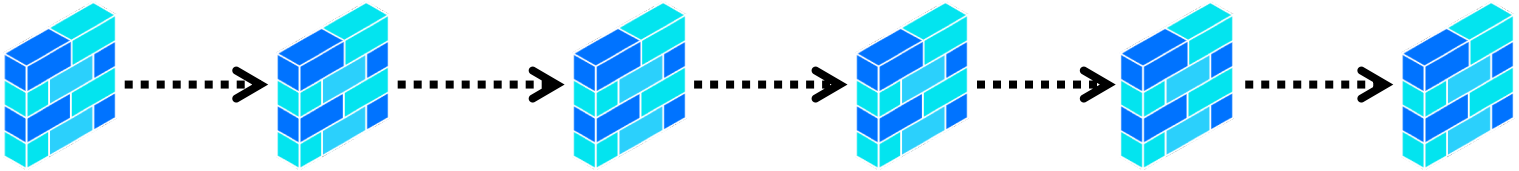
© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Amazon SageMaker

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

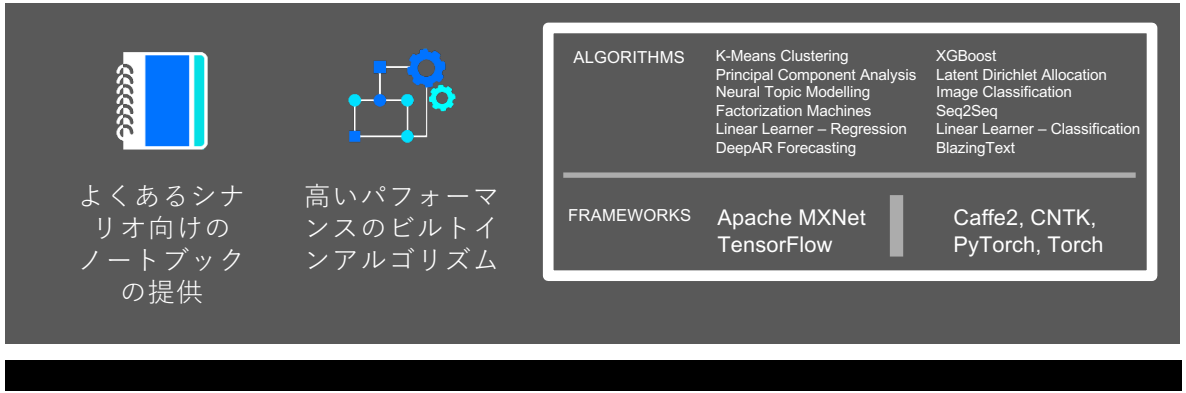
# Amazon SageMaker

Easily build, train, and deploy machine learning models

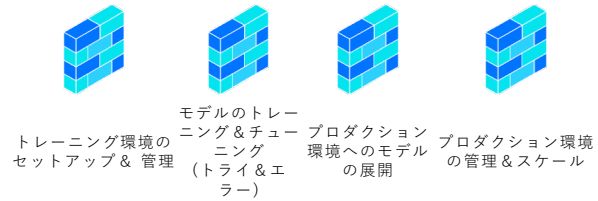



© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Amazon SageMaker




BUILD




© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Amazon SageMaker

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Amazon SageMaker

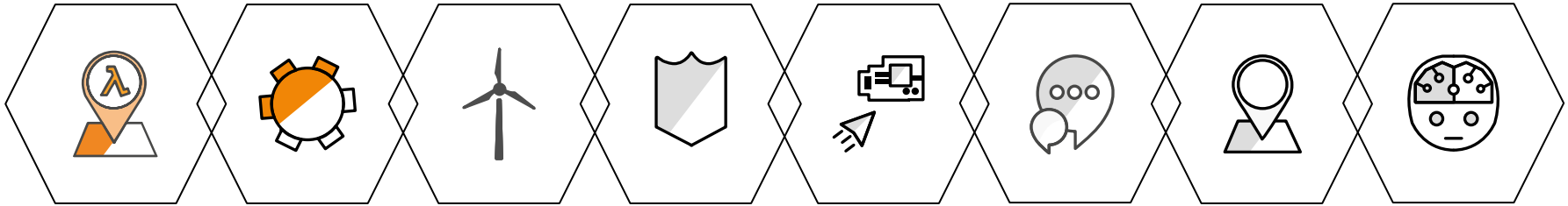
© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# AWS Greengrass

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# AWS Greengrass

## Extend AWS IoT to the edge



クラウドから配布した Lambda をローカルで実行

ローカルでのメッセージングと Lambda のトリガ

データと状態の同期（ローカルシャドウ）

クラウド同様のセキュリティポリシー

Greengrass Core 自体のアップデート（OTA）

OPC-UA のプロトコルアダプタ

ローカルリソースアクセス

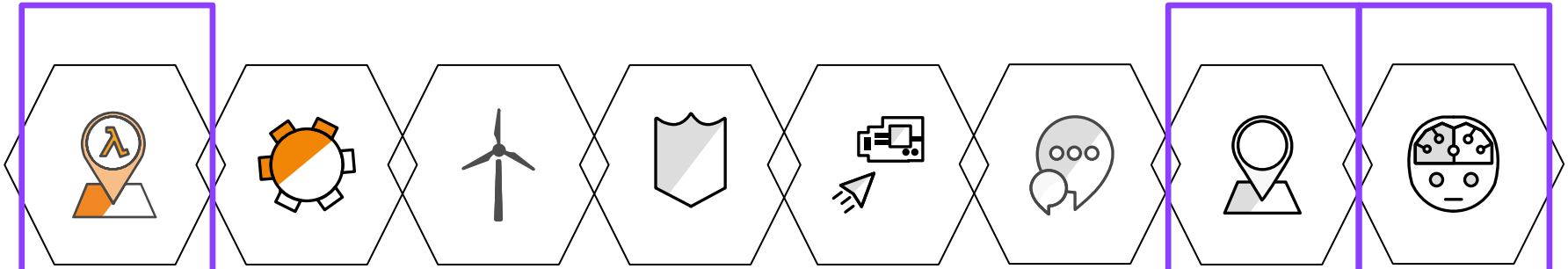
ローカルで ML/DL の推論を実行

**2018.4.4 GA Announce!!**

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# AWS Greengrass

## Extend AWS IoT to the edge



クラウドから配布した Lambda をローカルで実行

ローカルでのメッセージングと Lambda のトリガ

データと状態の同期（ローカルシャドウ）

クラウド同様のセキュリティポリシー

Greengrass Core 自体のアップデート（OTA）

OPC-UA のプロトコルアダプタ

ローカルリソースアクセス

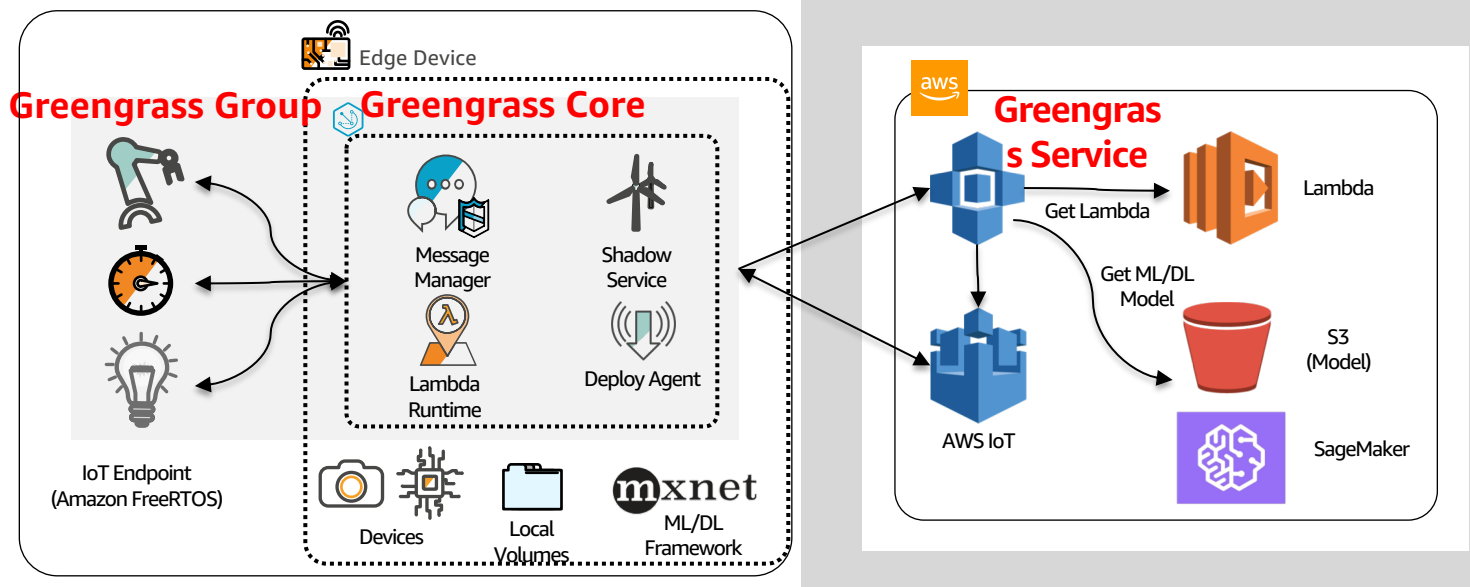
ローカルで ML/DL の推論を実行

**2018.4.4 GA Announce!!**

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# AWS Greengrass の構成要素

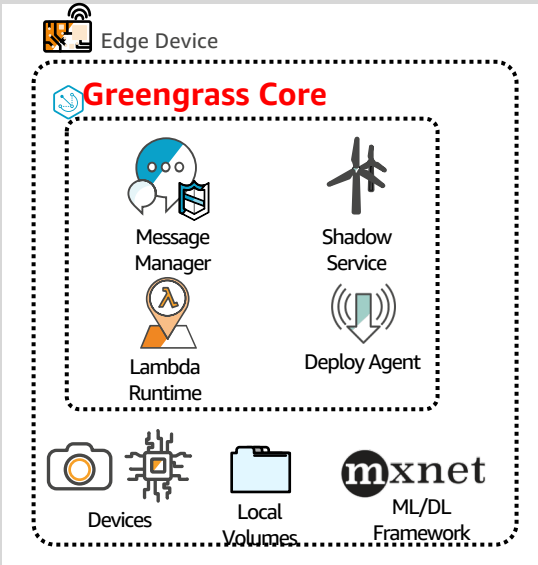
Greengrass は エッジデバイスにインストールするソフトウェア(Greengrass Core)と、そのふるまいの定義(Greengrass Group)やアプリケーションコードをデプロイするクラウドサービス(Greengrass Service)で構成される




© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Greengrass Core

- エッジデバイスにインストールするソフトウェア
  - 起動すると常駐するプロセスとして動作する
  - 動作させるハードウェアはユーザーが要件にあわせて選択



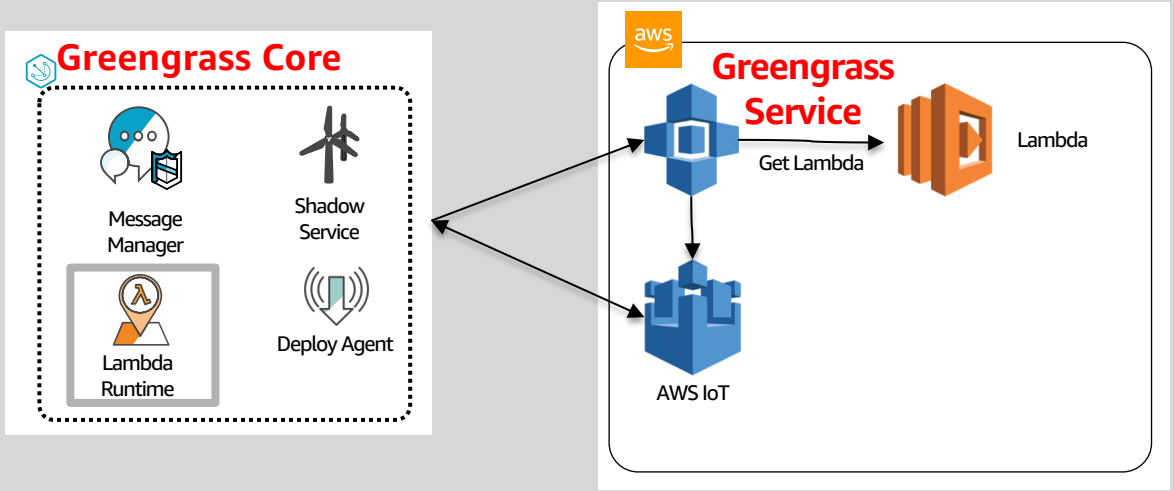

[動作条件]
- OS: Linux kernel > 4.4
- CPU: >1GHz, Arch = ARMv7l, ARMv8, x86_64
- Memory: >128 MB RAM
- その他、Greengrass 動作に必要な依存関係などの詳細
  - https://docs.aws.amazon.com/greengrass/latest/developerguide/what-is-gg.html#gg-platforms

- Greengrass Core 上で実行したいアプリケーションが必要とするスペックを確認する
  - CPU, メモリサイズ, ストレージ容量
  - ネットワークインターフェース
  - 周辺機器（カメラ、GPUなど）

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Lambda をローカルで実行

- Greengrass Core は Lambda runtime を保持し、Greengrass Service により Deploy された Lambda をエッジデバイス上で実行できる




© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# AWS Greengrass ML Inference

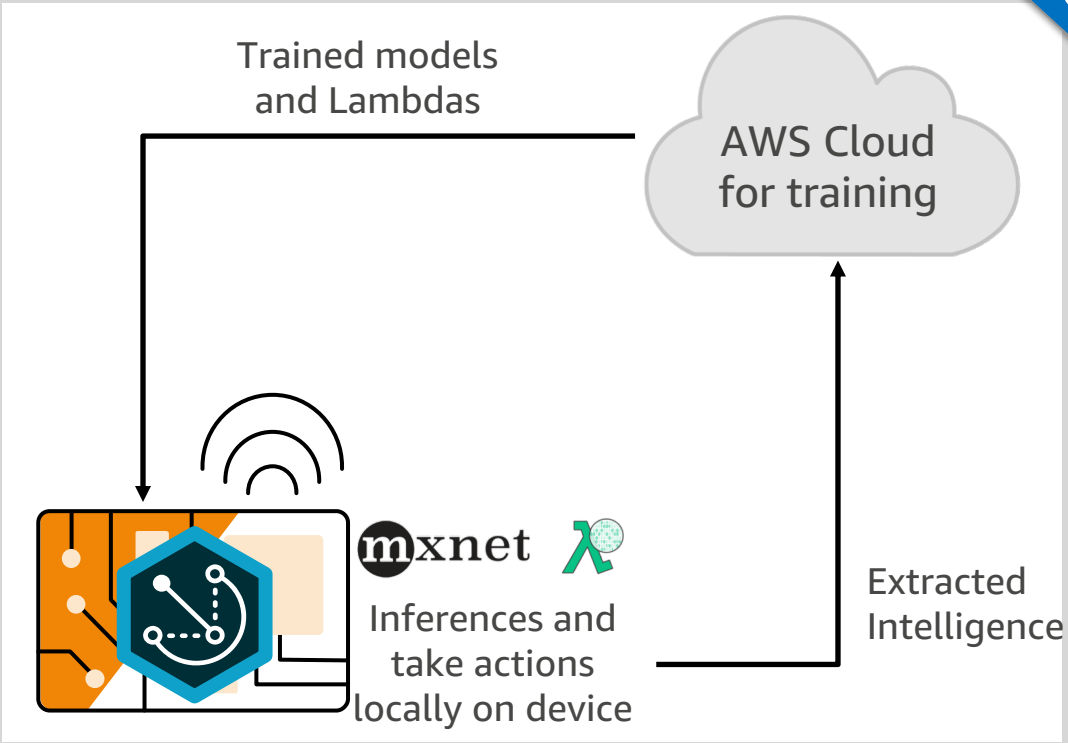
© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# AWS Greengrass ML Inference

2018.4.4 GA

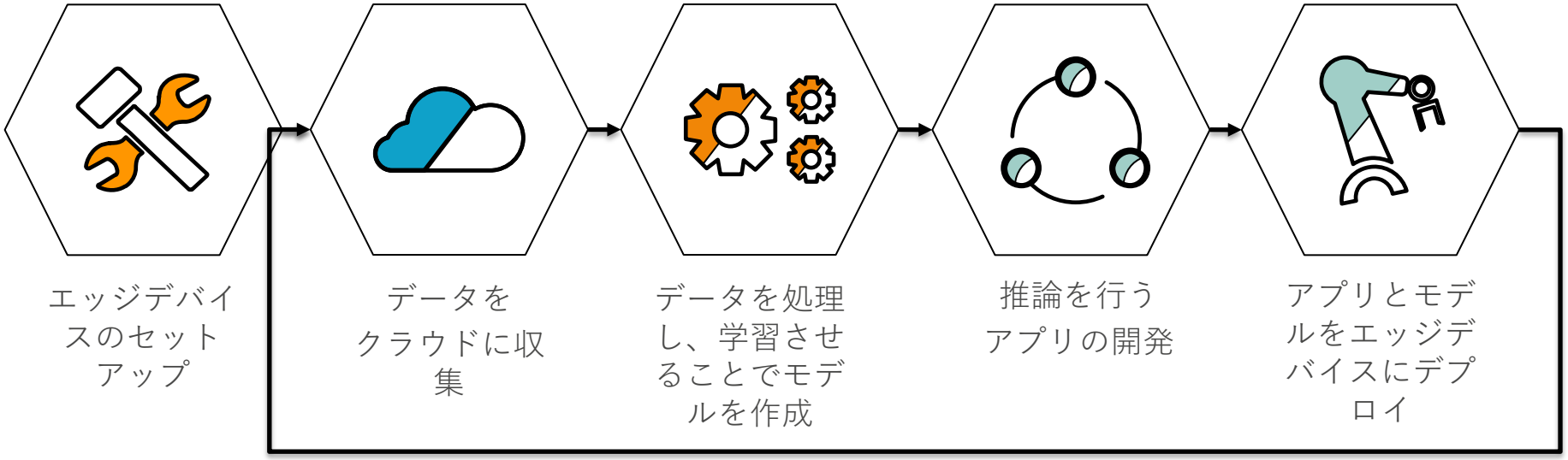
**エッジ上での機械学習の実行を簡単にする各種機能を提供**

- クラウドで学習したモデルを簡単にデプロイ可能
- MXNet/TensorFlow をデバイスに簡単に組み込める
- ローカルで機械学習の推論を行うための実装例を提供
- GPU/FPGA 活用が可能




© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

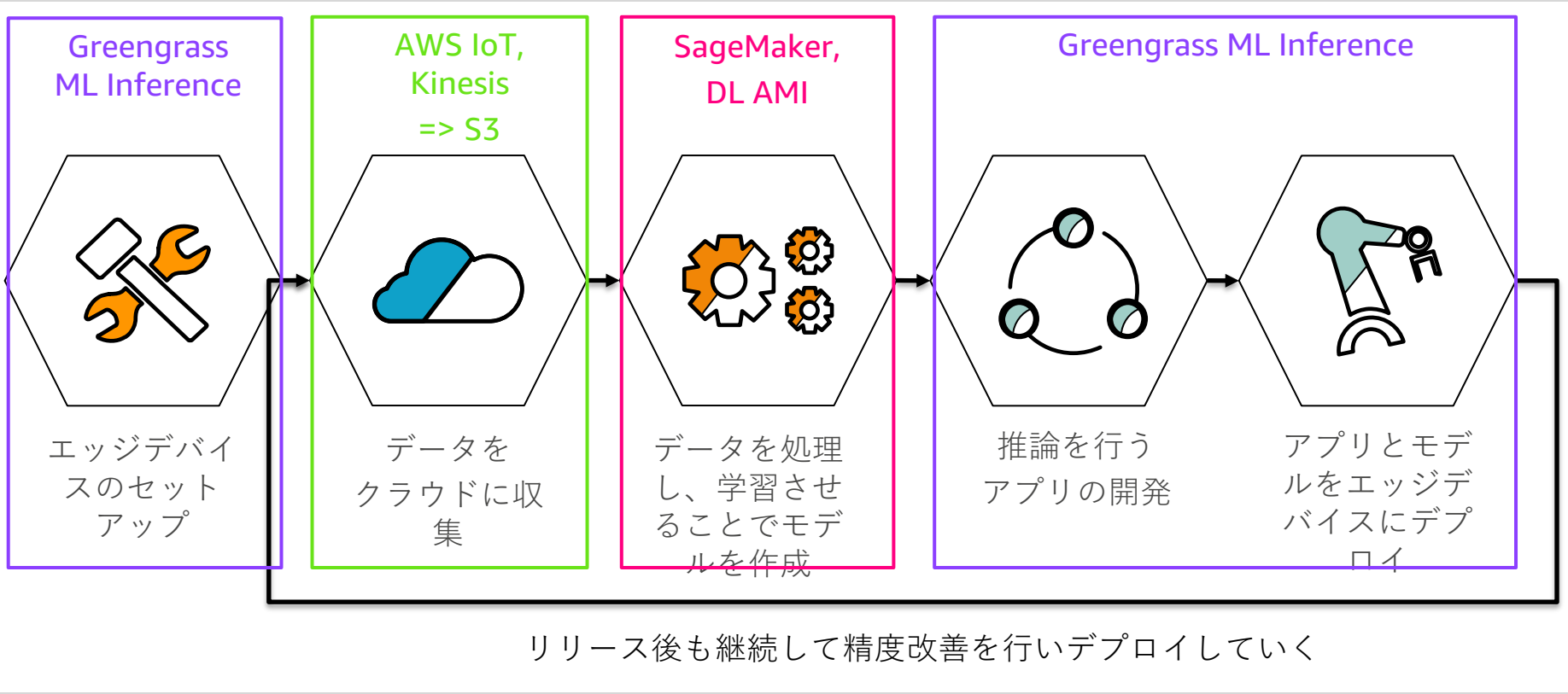
# エッジで機械学習を実行するための開発フロー




リリース後も継続して精度改善を行いデプロイしていく

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# 開発フローに対応する AWS サービス

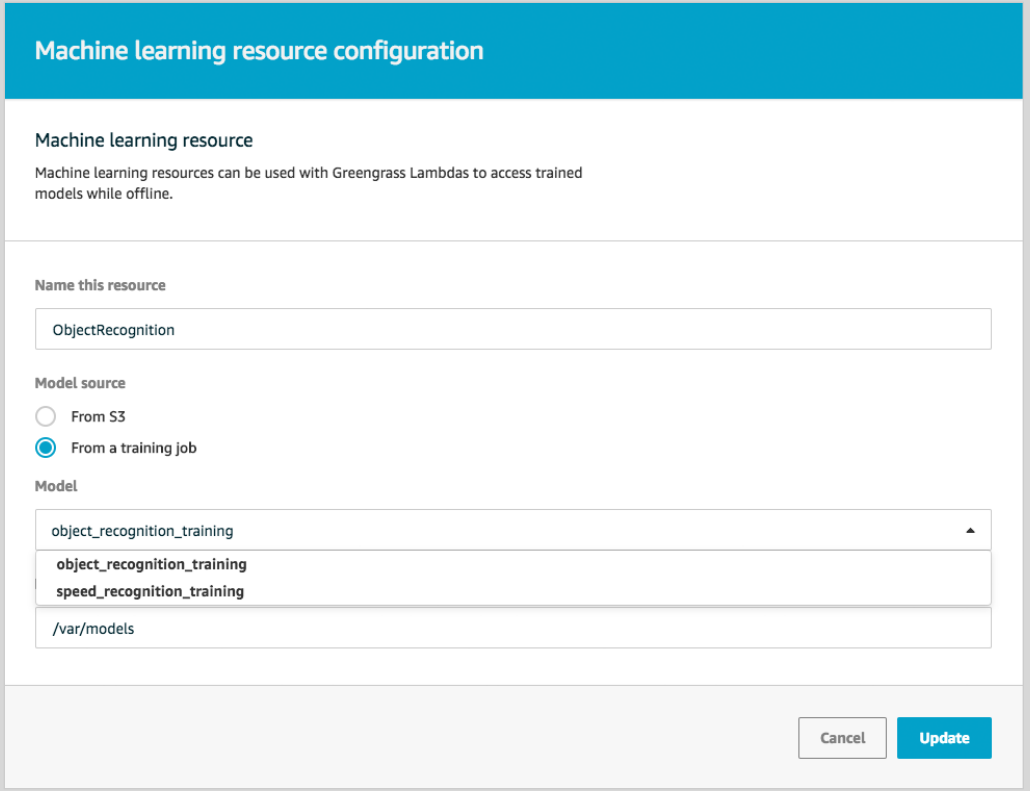



© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# 学習済みモデルのエッジデバイスへのデプロイ

**クラウドで学習したモデルをデバイスにデプロイ**

- 「機械学習リソース」として学習済みモデルを Greengrass グループに追加できる
- 設定したモデルが Greengrass デバイスにデプロイされる
- Greengrass コンソールから Amazon SageMaker の学習済みモデルを指定できる
- 独自のモデルを追加できる (MXNet などの ML フレームワークに依存しない)



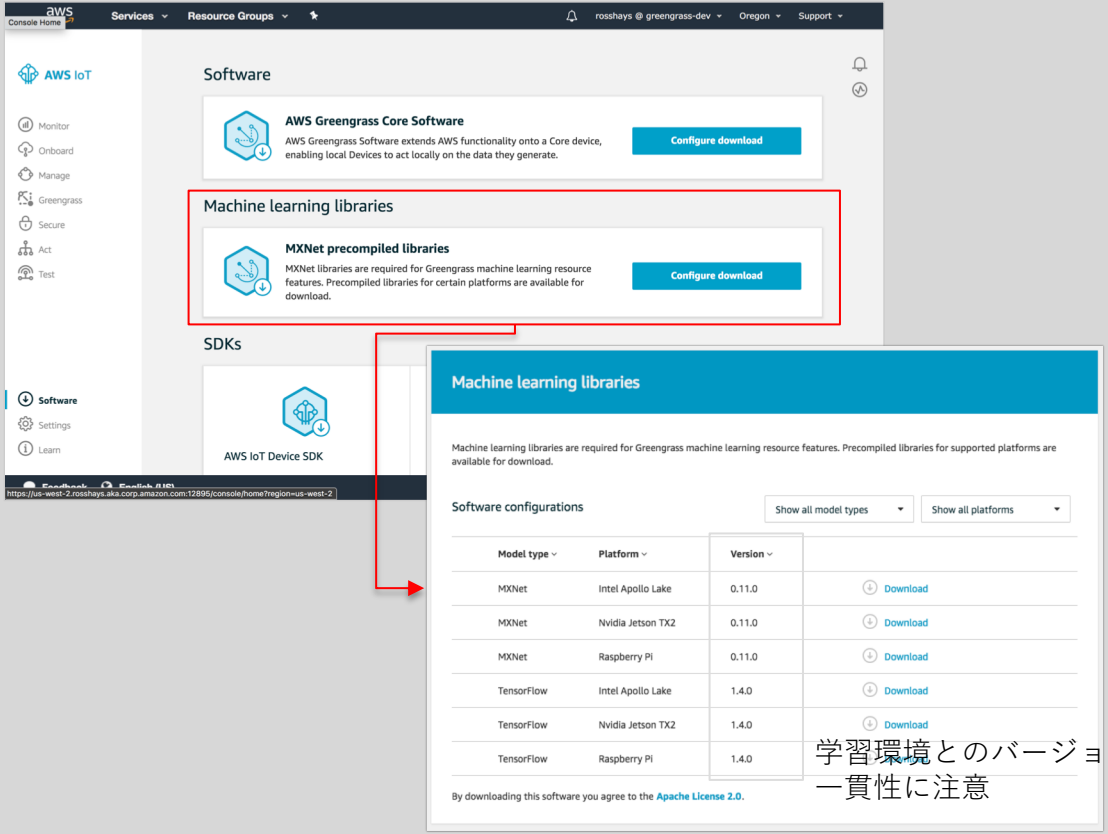

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# 事前ビルドした MXNet/TensorFlow の提供

**事前にビルドされたMXNet/TensorFlowパッケージを提供。対象デバイスでゼロからのビルドが不要に。**

- Intel Atom E3900 (Apollo lake)
- NVIDIA Jetson TX2
- Raspberry Pi

その他デバイスや MXNet/TensorFlow 以外の ML フレームワークの利用を制限するものではありません。(通常のビルドを行うことで利用可能)




学習環境とのバージョンの一貫性に注意

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# 推論を行う Lambda のサンプルコード

**推論を行う Lambda の実装例を提供**

- 学習済みモデルの読み込み
- ローカルで生成されたデータをモデルに適応して推論
- 推論結果に応じたアクション

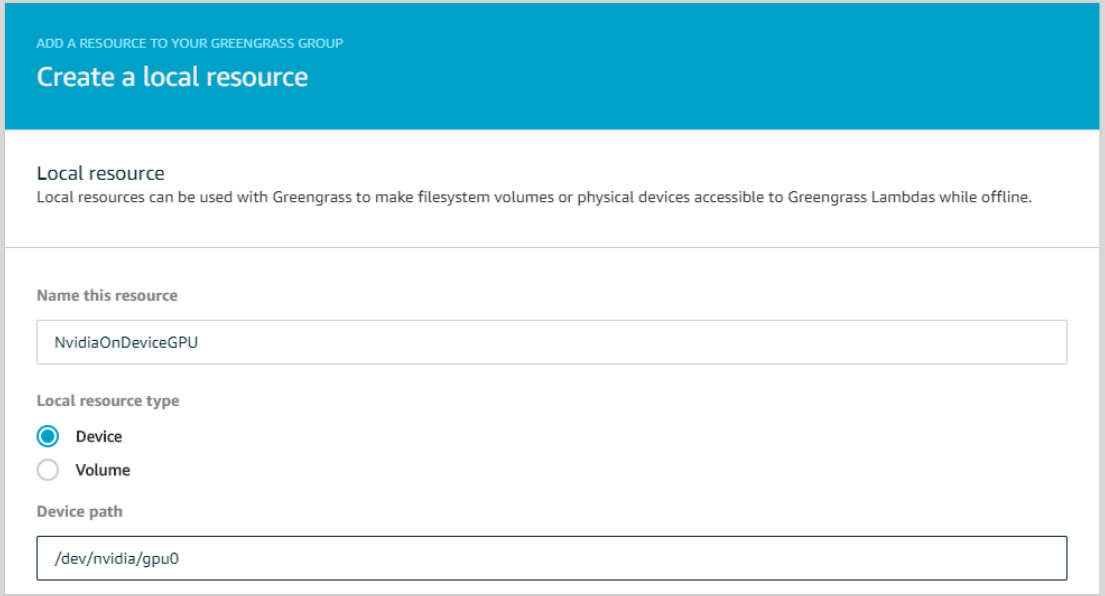
```python
# load_model_and_predict.py
import mxnet as mx
import picamera

# ...

# model files pulled down by Greengrass
_prefix = "/trained_models/" # "local Path defined in Greengrass ML resource"
params_file = _prefix+"-0000.params"
symbol_file = _prefix+"-symbol.json"

# Load the network parameters from default epoch 0
sym, arg_params, aux_params = mx.model.load_checkpoint(_prefix, 0)

# Load the network into an MXNet module and bind the corresponding parameters
mod = mx.mod.Module(symbol=sym, label_names=label_names, context=context)
mod.bind(for_training=False, data_shapes= input_shapes)
mod.set_params(arg_params, aux_params)

# ...

# Run forward on the image
mod.forward(Batch([mx.nd.array(img)]))
```

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# HW アクセラレータへのアクセス

**Lambda からデバイス上の GPU や FPGAといった ML アクセラレータにアクセスすることで推論を高速化**

- コードの実装は不要
- 「ローカルリソース」としてアクセラレータを宣言することで Lambda からの呼び出しが可能
- カメラやストレージといった HW も同様にアクセス可能

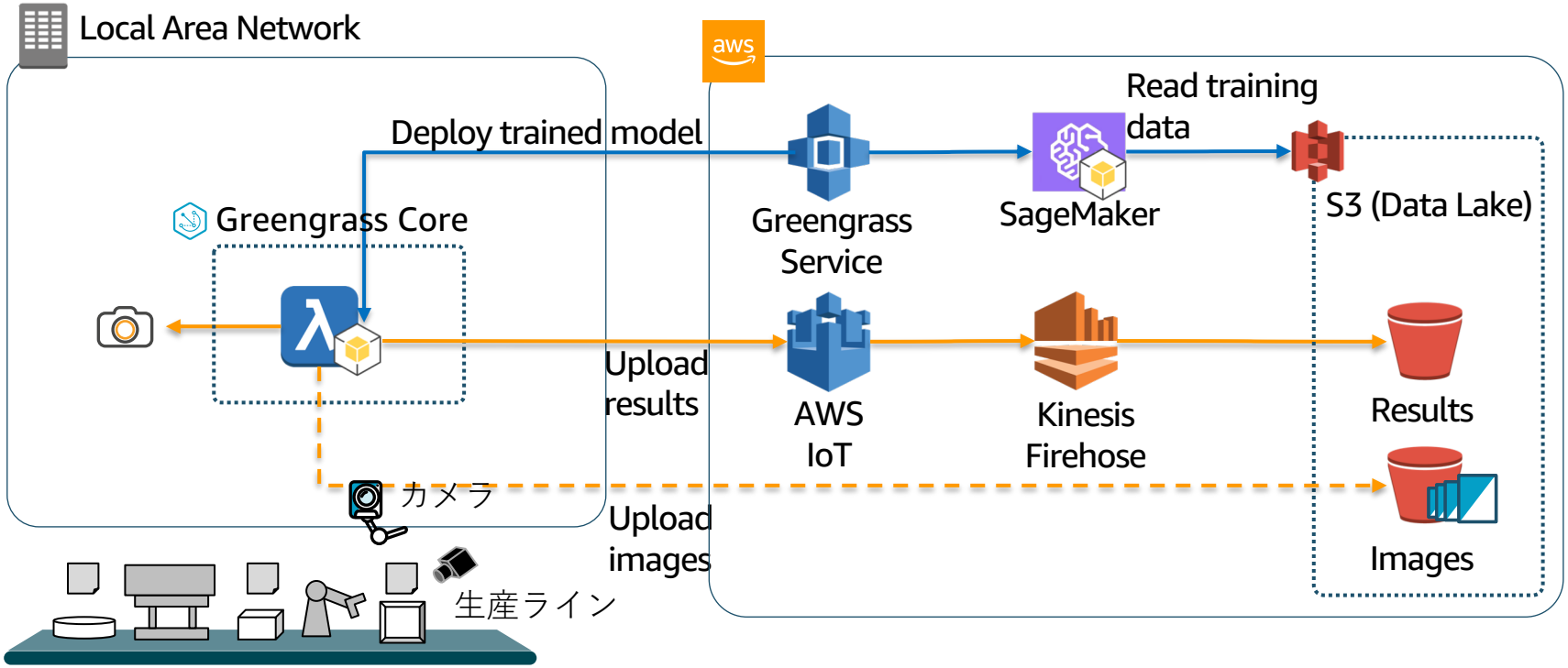



© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# アーキテクチャ例

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# アーキテクチャ例　製品の外観検査



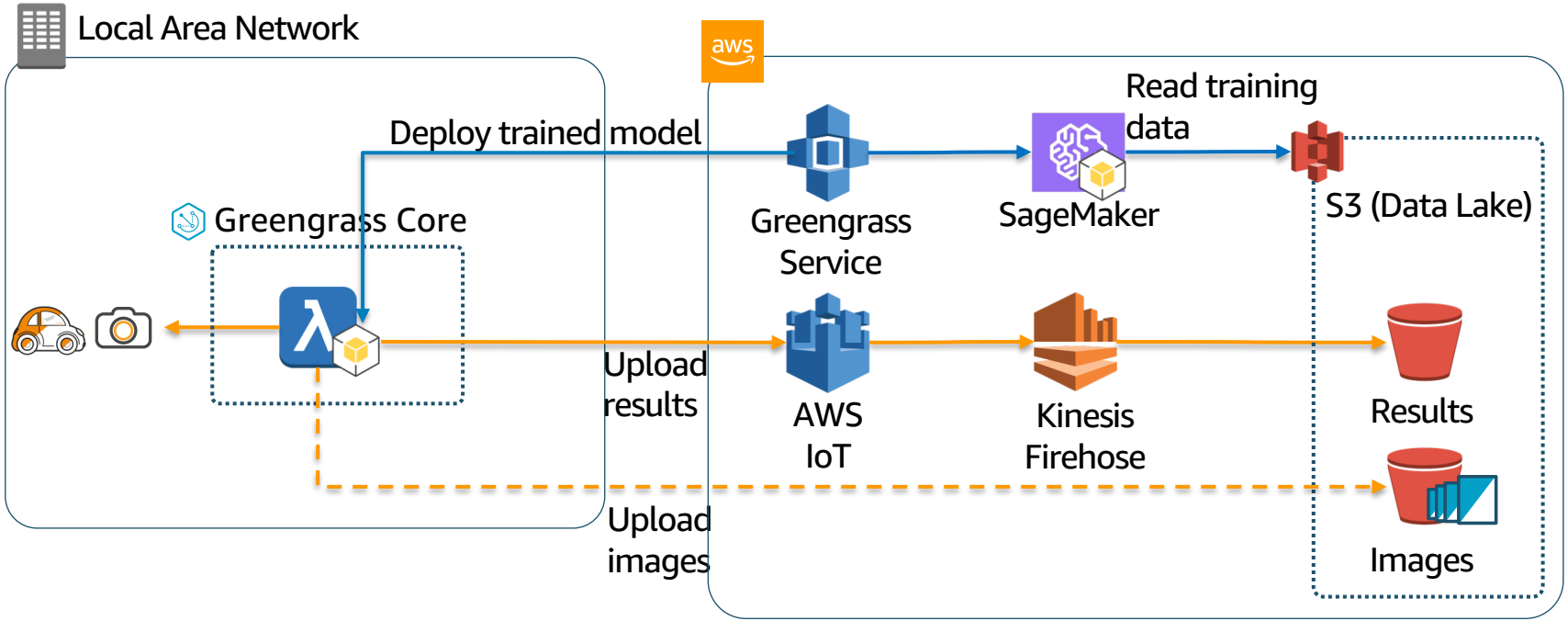

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws SUMMIT

# Demo

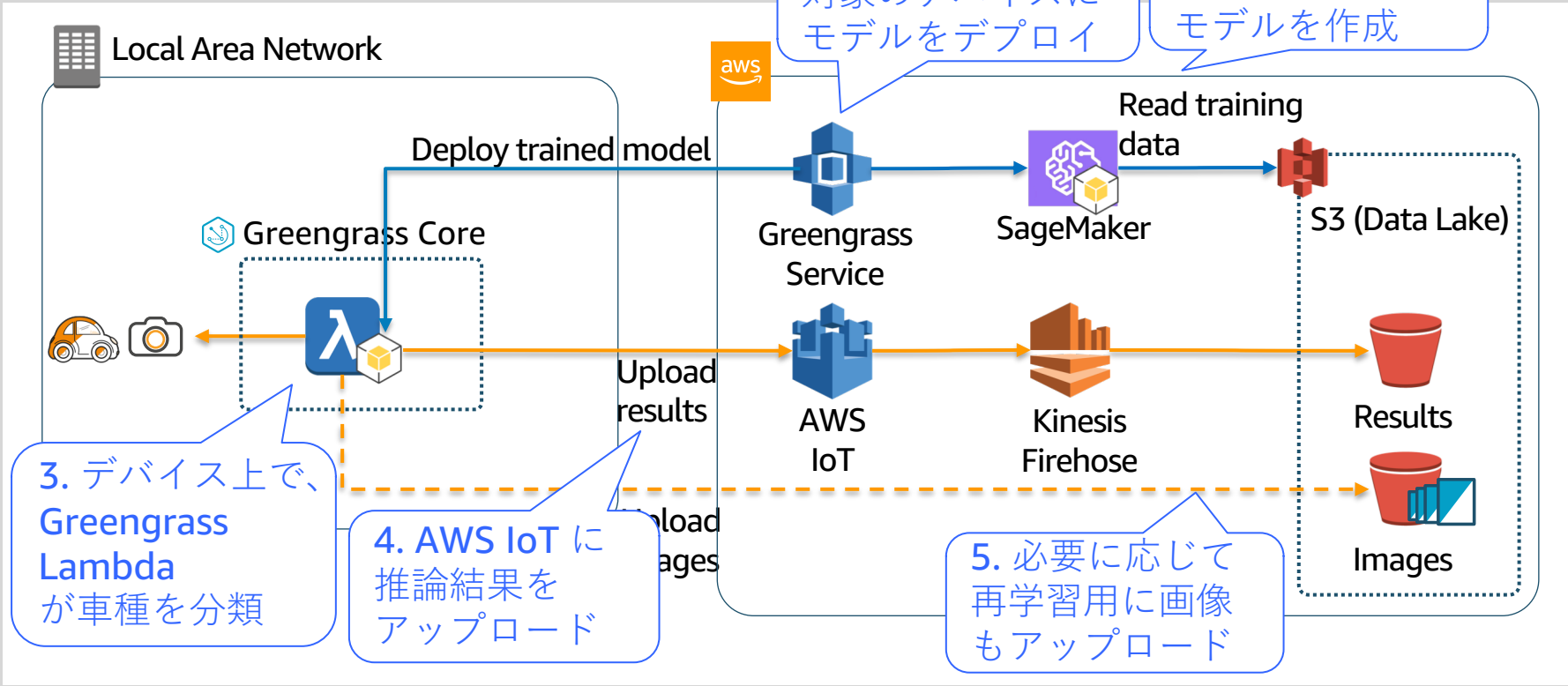
© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# デモのアーキテクチャ図




© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# デモのアーキテクチャ図




© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws SUMMIT

# まとめ

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# まとめ

- エッジデバイス上での機械学習の活用は進んでいる
- エッジ、クラウドの役割分担の設計が重要
- AWS サービスを活用し、エッジデバイス上での機械学習活用を加速

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

受付でお配りしたアンケートに
本セッションの満足度やご感想などを
ご記入ください

アンケートは受付、パミール**3F**の**EXPO**展示会場内にて回収させて頂きます

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws SUMMIT

# ご清聴ありがとうございました！

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.