NVIDIA Jetson におけるエッジアプリケーションと マイクロサービスアーキテクチャ

Latona株式会社

伊藤陽生・村上 尚仁

2021/3/17

SORACOM Device Meetup #2 ～ラズパイやJetsonで使えるIoT通信～

スピーカー紹介

伊藤 陽生 / Yosei Ito

バックエンドエンジニア

1995年生まれ（25歳）
東京出身

趣味：
　漫画を読むこと
経歴：
　2020/12にjoin
　インフラからバックエンドの開発がメイン

村上 尚仁 / Takahito Murakami

バックエンドエンジニア

1996年生まれ（25歳）
マレーシア出身

趣味：
　ゲーム・麻雀
経歴：
　2020/12にjoin
　バックエンドの開発を担当

Latonaの会社概要

創立
2018年4月6日

所在地
東京都渋谷区神宮前 6-12-18 WeWork Iceberg 3F
愛知県名古屋市中村区名駅1-1-1 JPタワー名古屋 21F
京都府京都市下京区四条通室町東入函谷鉾町101
アーバンネット四条烏丸ビル

事業内容
AI・IoTプラットフォームの開発及び運営
AI・IoTプロダクト・技術・ソリューションの開発・提供
ファクトリーIoTソリューションの開発・提供
旅館・ホテル向けおもてなしアプリケーションの開発・提供
自動小売店舗ソリューションの開発・提供

従業員数
28名

資本金
430,600千円（資本準備金を含む）

インベスターズ
未来創生2号ファンド、マネックスベンチャーズ、
SMBCベンチャーキャピタル、三菱UFJキャピタル、
静岡キャピタル、ほか個人投資家など

エッジ開発における取り組み

特長:

nvidiaのJetsonを使用

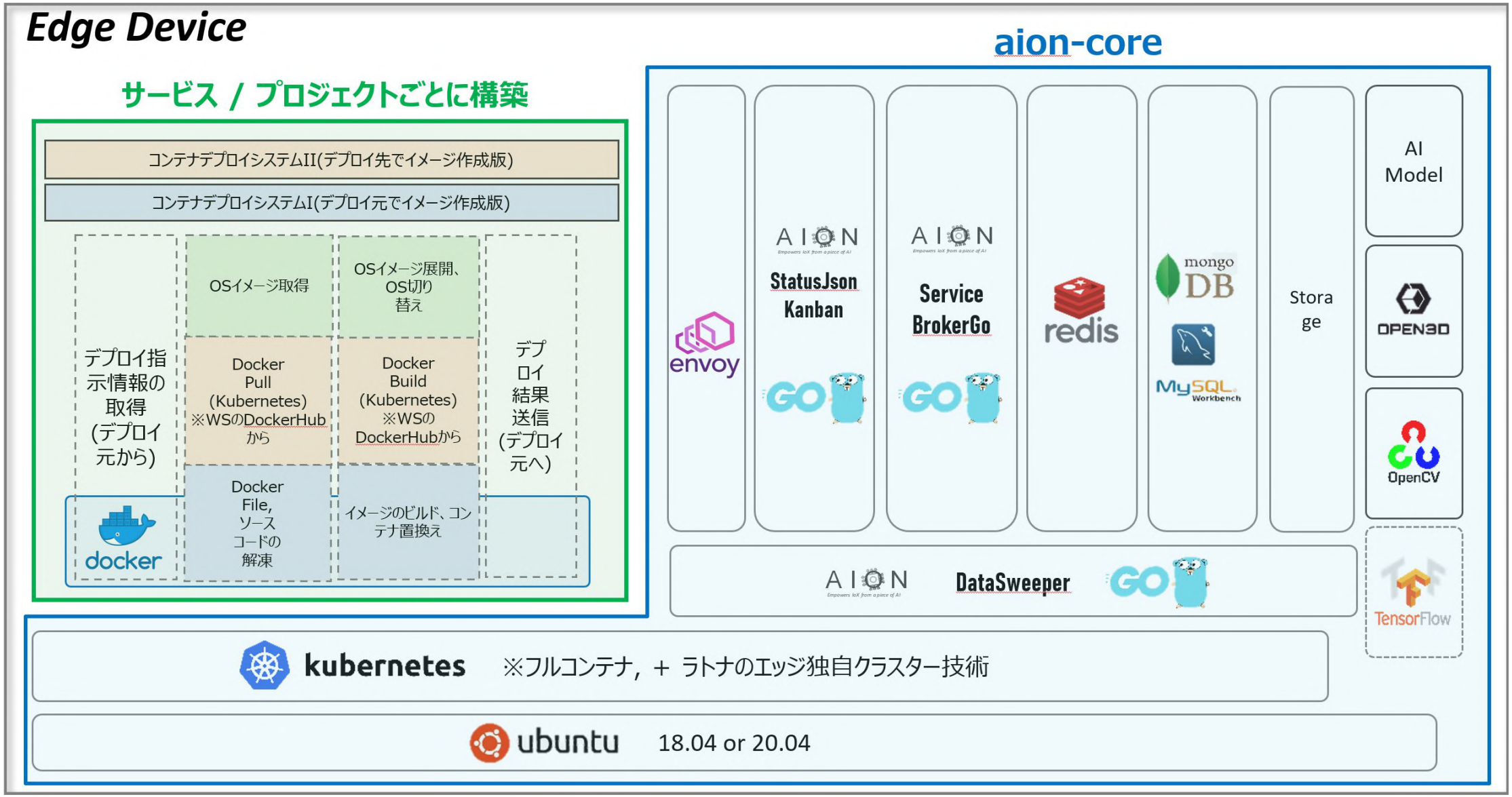
kubernetesをベースにマイクロサービスを構築

AION™の開発:

マイクロサービスのためのプラットフォーム

- 機能:
  - マイクロサービス間の通信プロトコルの提供
  - デプロイメントの管理
  - メッセージングログの永続化

AION™アーキテクチャ図



コンテナオーケストレーション技術(国内/国際特許取得)
Kubernetes等を利用したセンサシステム制御技術

特許名	コンテナオーケストレーション技術を利用したセンサ情報処理システム、センサ情報システムの制御方法、センサ情報処理システムの制御に用いるコンピュータプログラム、及び、その記録媒体。
出願日	2019年3月8日（2020年8月取得）
特許出願番号	特願2019－043061
詳細	本技術は、コンテナオーケストレーションツールを利用したセンサ情報処理システムにおいて、信頼性の高いセンサ情報を使用しつつも稼働中のセンサの状態変更にも耐え得る堅牢性と、システム規模の変更などにも対応可能な柔軟性とを兼ね備えたセンサ情報処理技術を提供する。 コンテナオーケストレーションツールが用いられることにより、複数のコンテナのデプロイ、実行、管理及びスケジューリングなどを容易に行うことができる為、さらにシステム開発の負担を軽減することができる。 (オーケストレーションツールの例:Kubernetes, Apache Mesos 等)

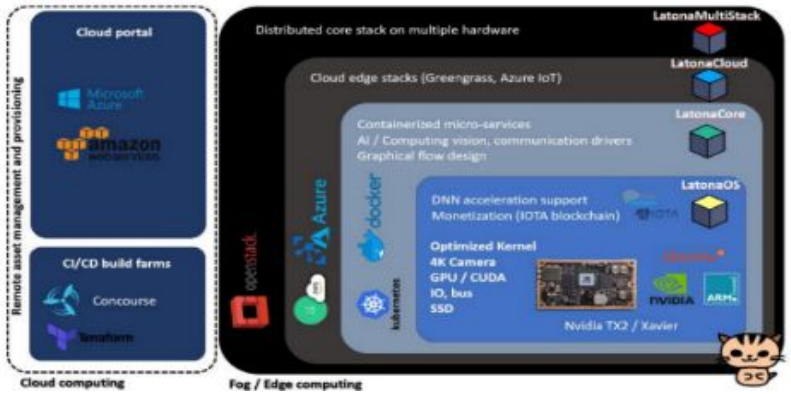
LatonaのKubernetes コンテナオーケストレーション on the edge(Latona IoT Core)の特徴

I. NVIDIA * ARM
世界の技術潮流のセンターに位置する、Nvidia * ARMのテクノロジーをコアに採用しています。

II. オープンソース
オープンソースを採用しているので、製品の開発コストが安価で済みます。

III. マイクロサービスアーキテクチャ
マイクロサービスアーキテクチャをあらゆる開発要素で採用、どの部品を構築しても別の製品で再利用可能な構成になっており、プラットフォーム開発に最適です。

IV. 主要クラウドとシームレスに連動
Kubernetesベースの技術は、AWSならLambda、AzureならIoT、GoogleならKubernetes自体と、クラウドからのリソース連携にそのまま連動できるアーキテクチャです。



接客業

ユースケース(小売向けの事例)

boxsta

DNP 大日本印刷 X LATONA

2019年11～12月期間限定実施

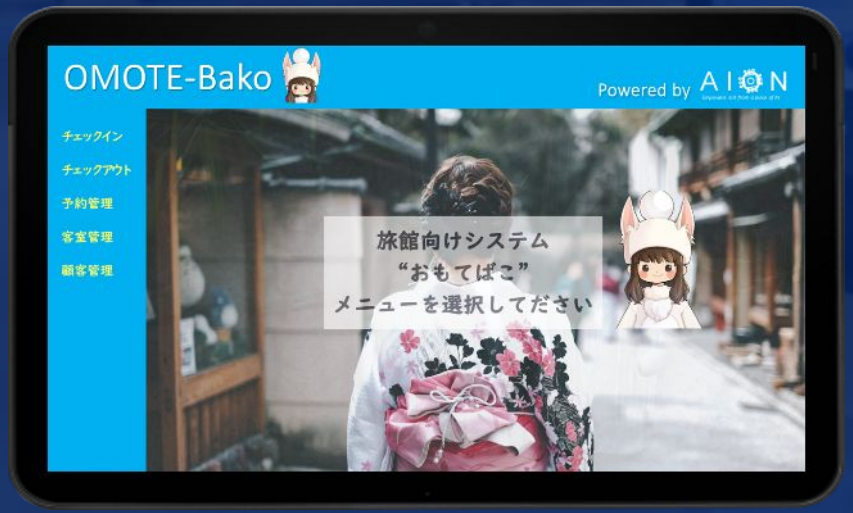
接客業

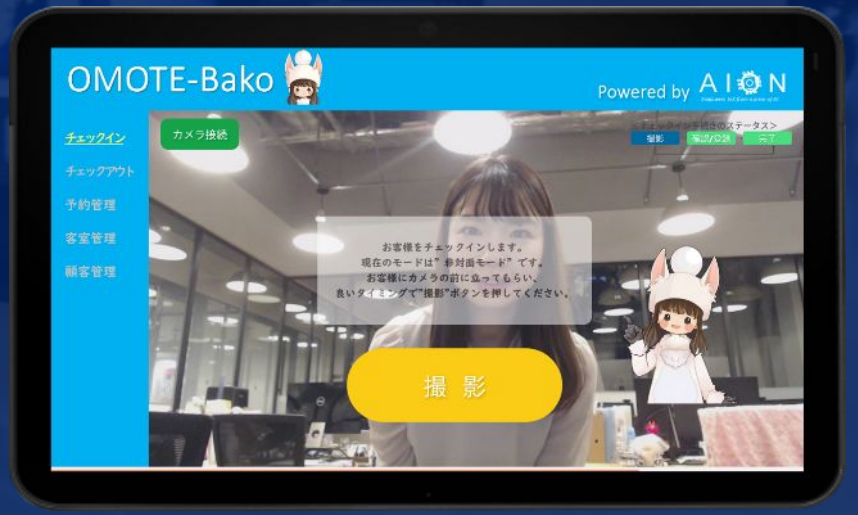
ユースケース(小売向けの事例)

OMOTE-Bako

- 顔認証によるチェックイン/チェックアウト
- エッジ上での予約管理/顧客管理
- スマートフォン連携による館内業務効率化

OMOTE Bako



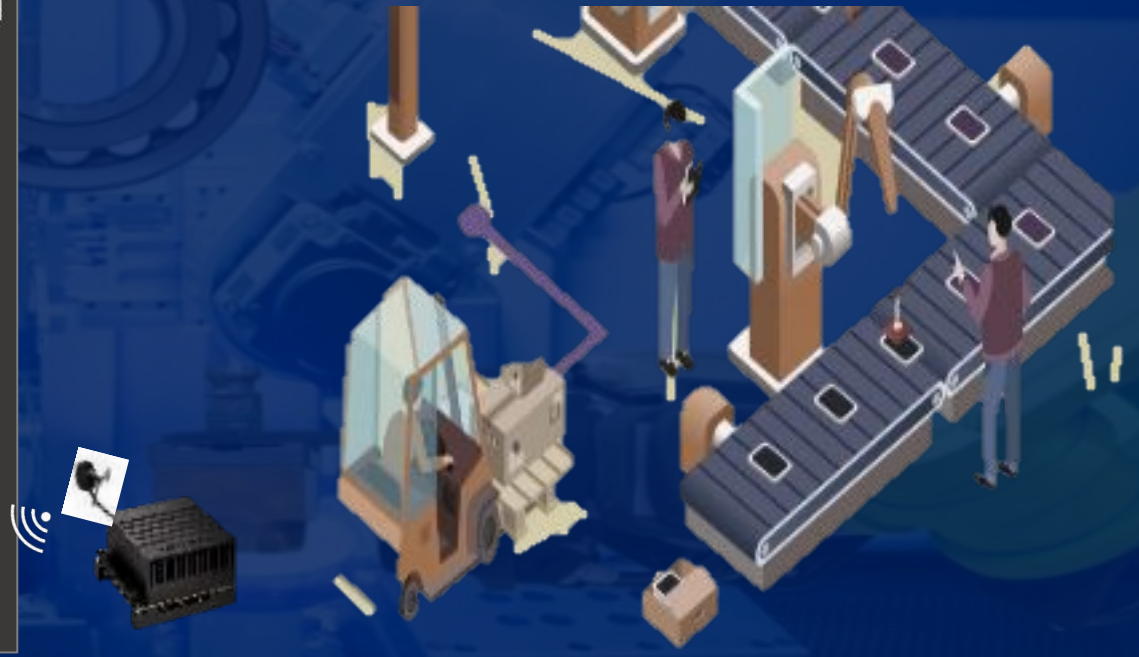


SORACOM使ってみた

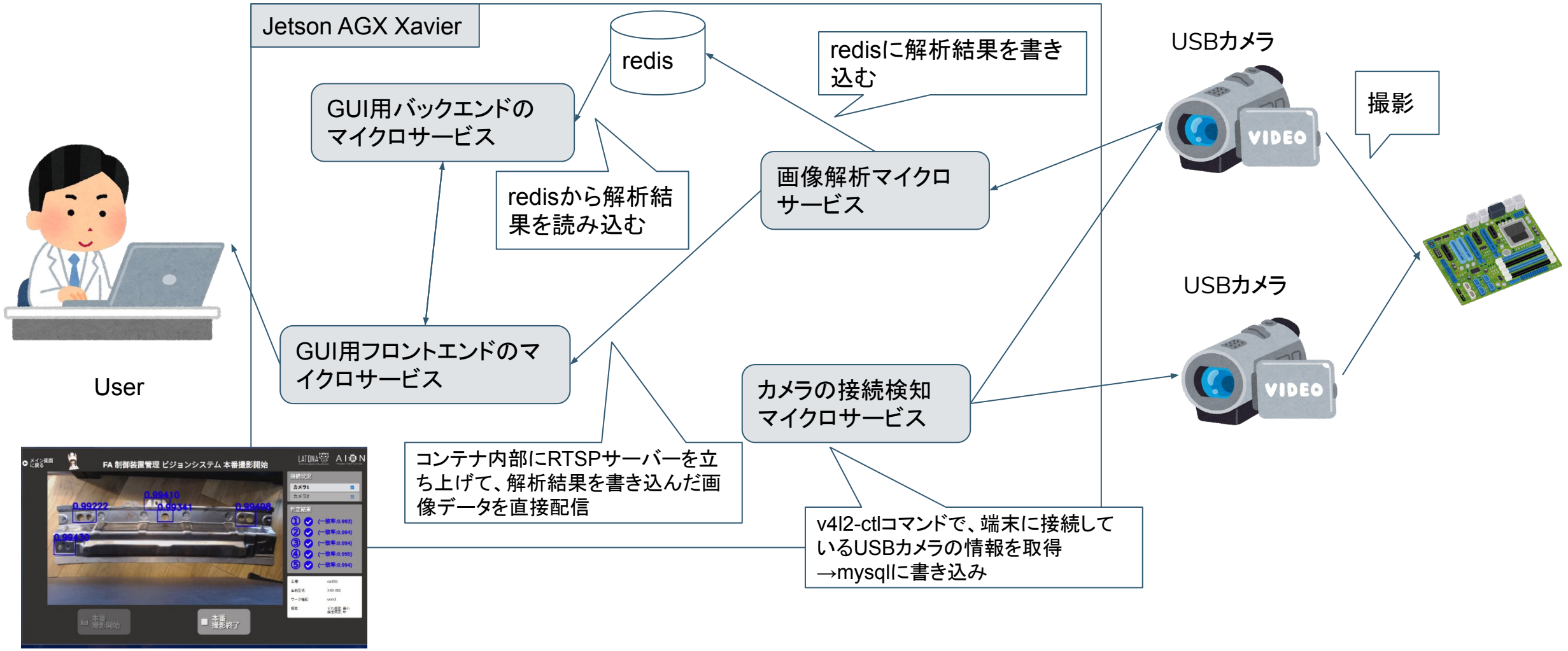
製造業

検品向けエッジAI（合否判定）

- エッジ×テンプレートマッチング(OpenCV)
- タッチパネルUI表示のポカヨケビジョンシステム



実際のシステム構成例(検品向けエッジAIシステム)



サービス運用の課題

運用保守の難しさ

- 多くの場合、設置場所は
  - 物理的に遠隔地の工場
  - 外部ネットワークへ接続できない

  など、オンラインでのサポートが難しい環境
- 問題の発覚もリアルタイムでは難しい

トラブルが起きた際の対応例

1. 障害発生時にお客様の現場に行って対応
2. 電話で話しながらお客様自身に復旧用のコマンドを実行していただく
3. 端末を郵送で送り返してもらって対応

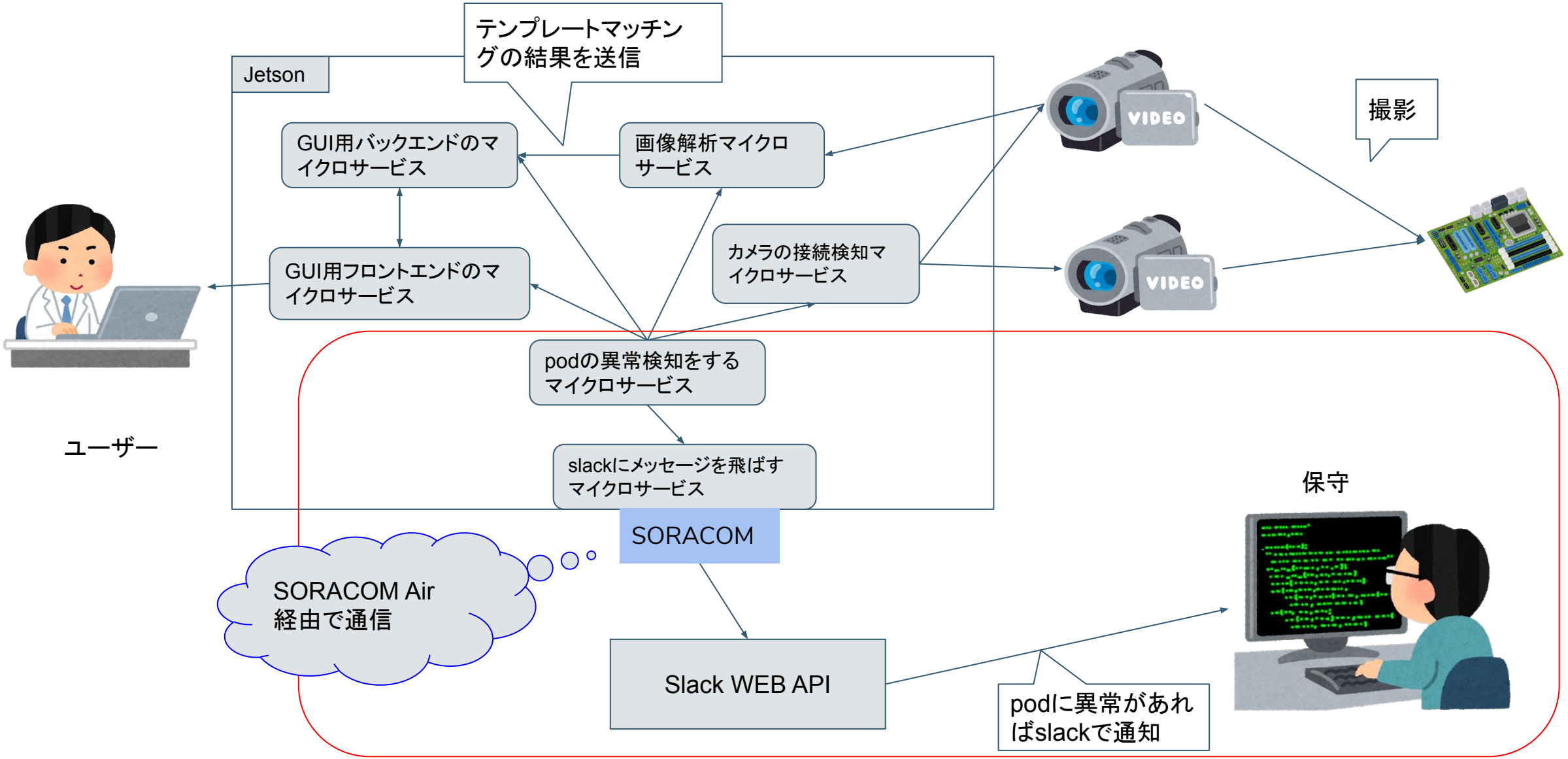
SORACOMを活用できそう

slackと連携したpodの異常検知システムを作りました

podの異常検知システムの概要

1. 稼働しているpodに異常発生！！
2. 異常検知用マイクロサービスが異常を検知
3. AION経由でメッセージを配信
4. slack投稿用マイクロサービスがメッセージを受信
5. SORACOM経由でslack WEB APIを実行
6. slackへアラートが飛ぶ
7. SORACOMで構築した閉域網やSSH接続環境でトラブルシューティング

SORACOMを活用したpodの異常検知システムの構成



実際に動かしてみました

実際に動かしてみた

稼働中のpod

```
sample-app-001-5694b5d8c4-4j7q6                    2/2     Running             0          17m
```

実際に動かしてみた

CrashLoopBackOffの発生！

```
sample-app-001-5694b5d8c4-4j7q6                  1/2    CrashLoopBackOff  0          17m
```

実際に動かしてみた

異常検知用マイクロサービスがCrashLoopBackOffを検知し、AIONのgRPCサーバーにメッセージを配信

```
$ kubectl logs aion-watchdog-kube-001-6559774fd8-t4sc4 aion-watchdog-kube-001
:57:21 - DEBUG -     0 -  - success to connect status kanban server: localhost:11010
:57:21 - DEBUG -     0 -  - success to send request(message type: START_SERVICE_WITHOUT_KANBAN)
:57:23 起動に失敗したpodを検知しました。POD名：sample-app-001-5694b5d8c4-4j7q6, Reason: CrashLoopBackOff
:57:23 - DEBUG -     0 -  - success to send request(message type: OUTPUT_AFTER_KANBAN)
:57:23 kanbanデータの送信に成功しました
```

実際に動かしてみた

AION経由で、slack用マイクロサービスにslack通知イベントを通知

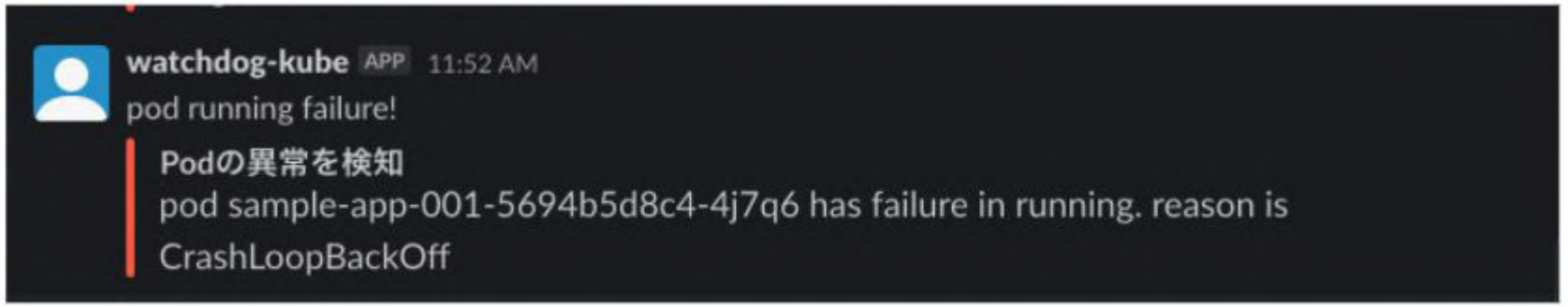
→整形しslackに送信

送信後のレスポンスのログが右です（一部省略）

```
$ k logs slack-message-client-kube-001-84f7796dbd-lm4kq slack-message-client-kube-001
2021/03/16 02:52:44
{
  "ok": true,
  "message": {
    "bot_id": "XXXXX",
    "type": "message",
    "text": "pod running failure!",
    "user": "XXXXXX",
    "team": "XXXXXX",
    "bot_profile": {
      "id": "XXXXXX",
      "deleted": false,
      "name": "watchdog-kube",
    },
    "attachments": [
      {
        "text": "pod sample-app-001-5694b5d8c4-4j7q6 has failure in running. reason is CrashLoopBackOff",
        "title": "Pod\u306e\u7570\u5e38\u3092\u691c\u77e5",
        "id": 1,
        "color": "ff6347",
        "fallback": "Pod\u306e\u7570\u5e38\u3092\u691c\u77e5"
      }
    ]
  }
}
```

実際に動かしてみた

slack WEB APIがslackにメッセージを投稿



まとめ

- オンラインでのサポートが難しい環境にSORACOMを導入してみました
  - ネットワークが無い、遠隔地にある、などの外部環境に左右されず、ソラコム経由で即時に障害検出→対応が実現できる
  - SORACOMさんのVPGを使えばセキュアなネットワークを確保できるので、セキュリティ的な懸念もクリアできそう
  - SORACOM Napterを使えばSSH接続とかも簡単にできそうなので、そのあとのトラブルシューティングもセキュアかつ楽に行える

今回作ったマイクロサービス

aion-watchdog-kube(https://github.com/latonaio/aion-watchdog-kube)

- 異常検知用マイクロサービス

slack-message-client-kube(https://github.com/latonaio/slack-message-client-kube)

- slackへメッセージを送信するマイクロサービス