aws

SUMMIT ONLINE

JAPAN

いかにしてスタートアップは
水産養殖の現実の課題を解決するか
-ウミトロンにおけるIoTと機械学習の取り組み-

岡本拓磨

CTO
ウミトロン株式会社

aws SUMMIT ONLINE

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

自己紹介

岡本拓磨

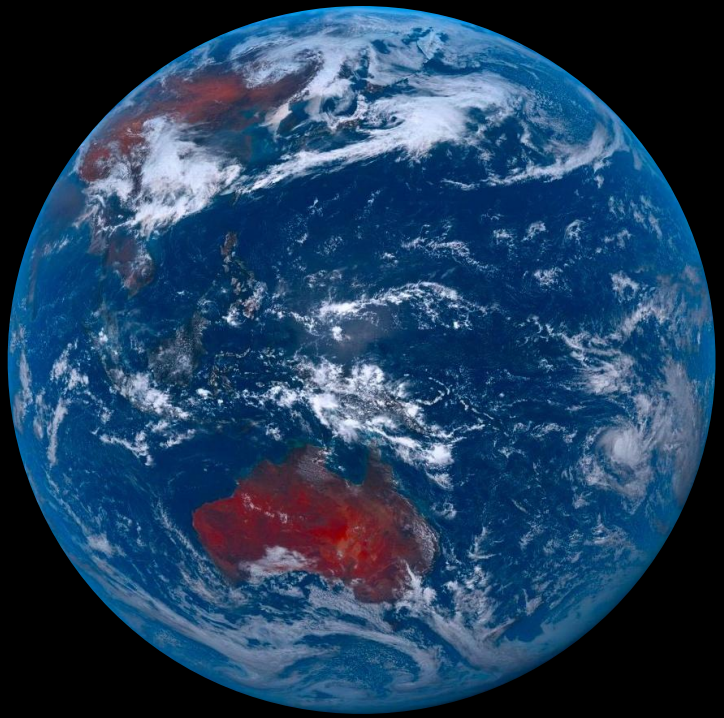
- ウミトロン株式会社
  - 共同創業者 / 取締役 / CTO
- 経歴
  - 東京理科大学・大学院
  - グリー株式会社(2012 -)
  - 株式会社メタップス(2014 -)
  - ウミトロン株式会社(2016 -)

UMITRON ALL RIGHTS RESERVED

UMITRON

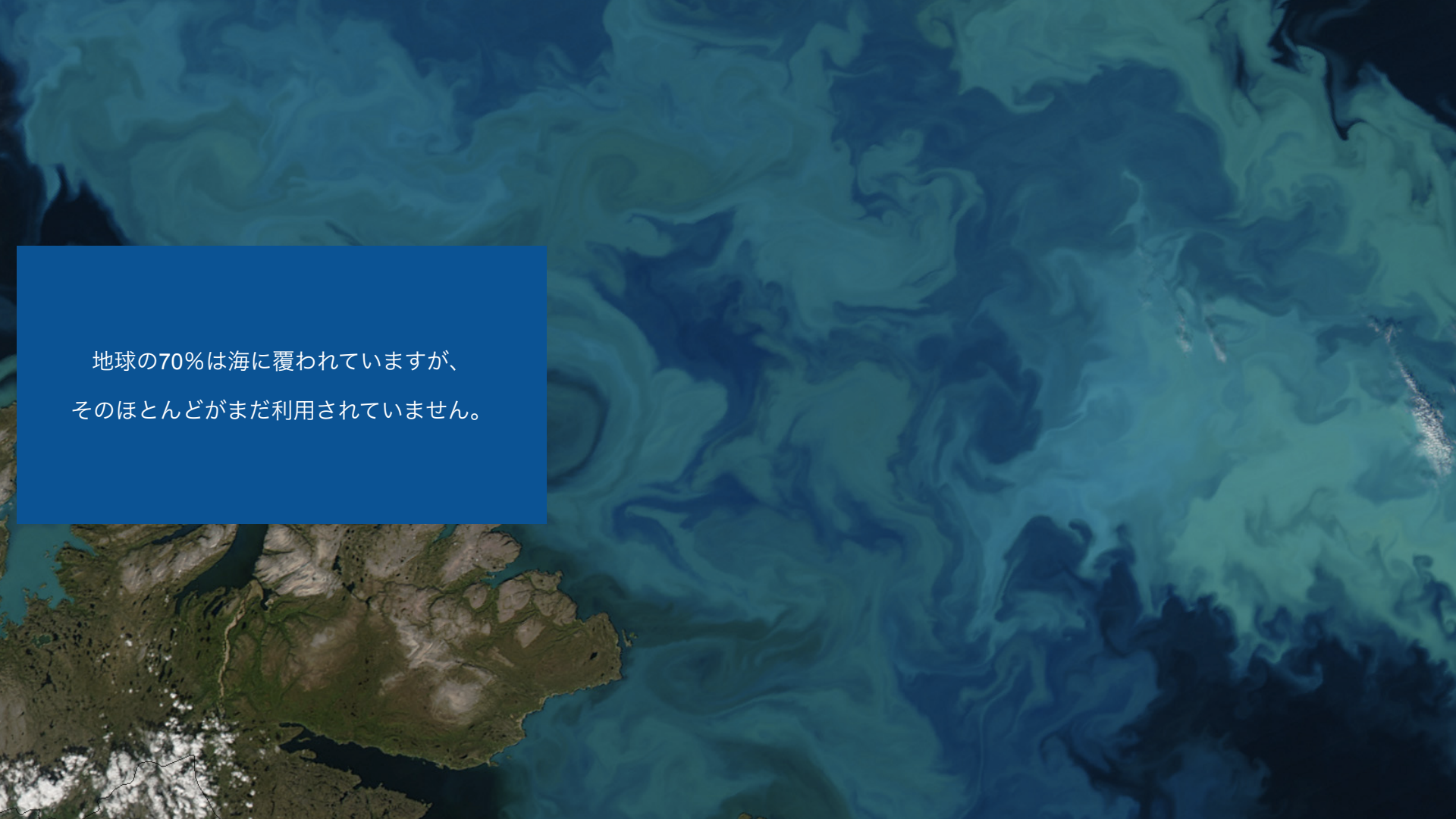
install Sustainable Aquaculture on Earth

UMITRON ALL RIGHTS RESERVED



2050年には100億人の人々が生き、

私たちの青い惑星「地球」で進化を続けます。

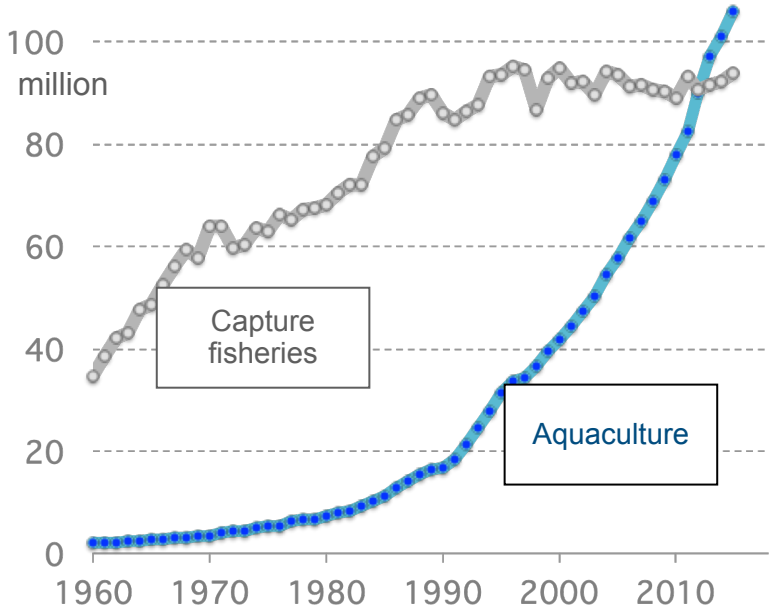


地球の70％は海に覆われていますが、

そのほとんどがまだ利用されていません。

UMITRONの使命は、地球に持続可能な水産養殖を実装し、養殖を次世代の食糧生産方法に移行させることです

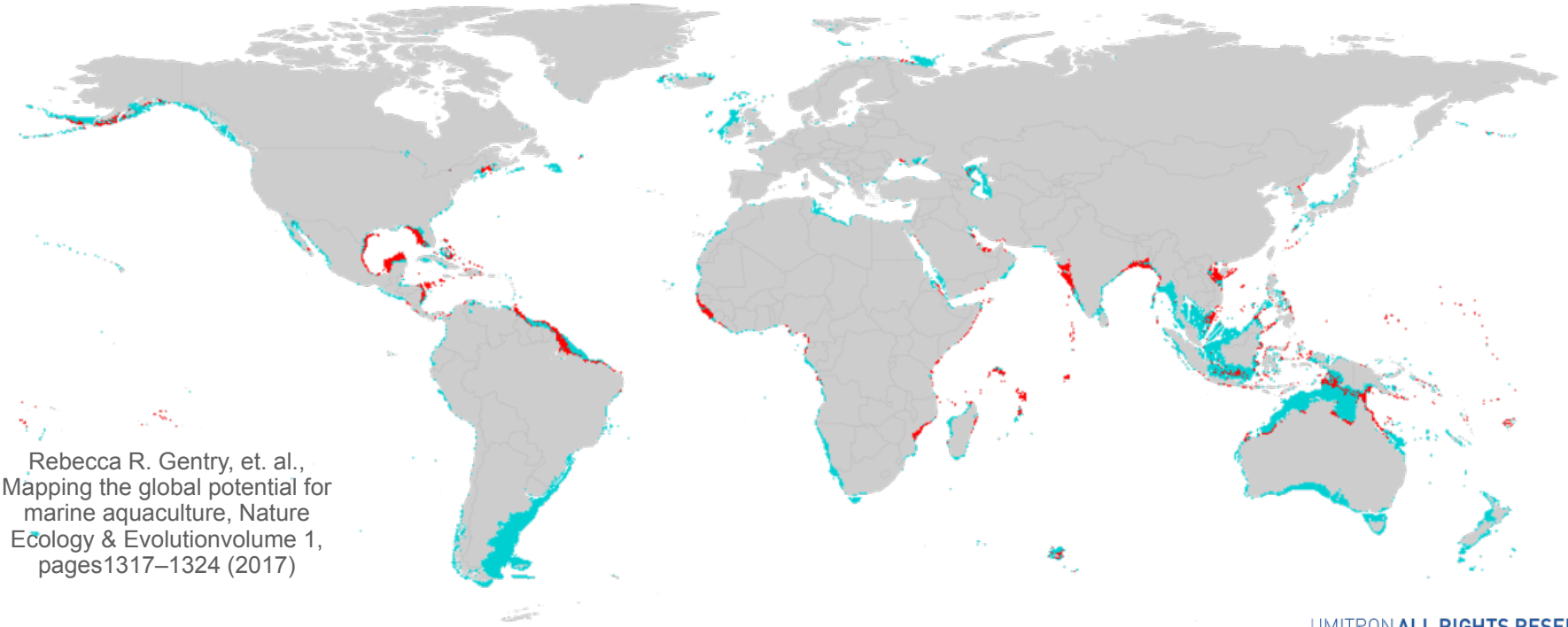
水産養殖はアジアを中心としたタンパク需要と中間所得層増加により急成長を続けている



- 人口増加と中間所得層増加により動物性タンパクの需要が急増
- 世界の漁業・養殖業事業者の85%がアジア地域に存在
- 近年は養殖業での雇用の伸びが顕著でアジア地域で1100万人が養殖業に従事

UMITRON ALL RIGHTS RESERVED

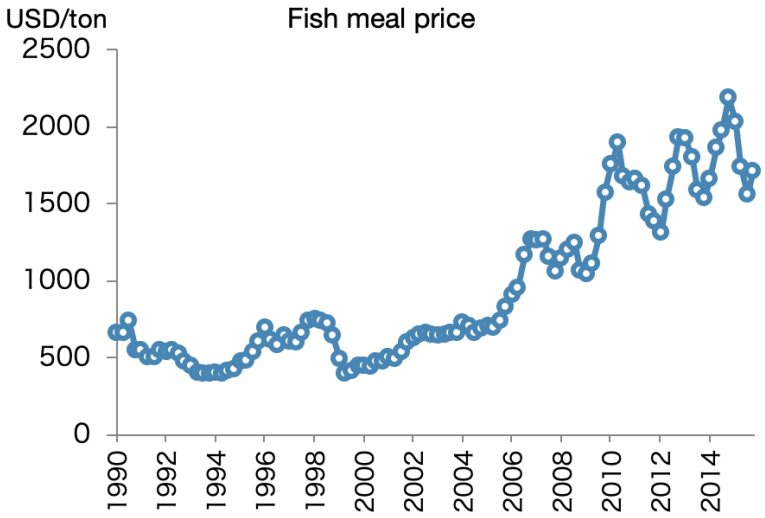
水産養殖は最も将来性の高い食糧生産方法の一つであり、沿岸海域を活用することで現在の世界の水産物消費量の100倍を生産可能

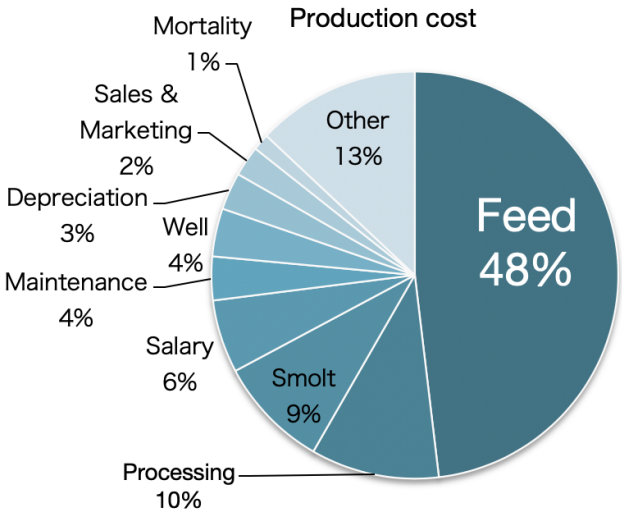


Rebecca R. Gentry, et. al., Mapping the global potential for marine aquaculture, Nature Ecology & Evolutionvolume 1, pages1317–1324 (2017)

UMITRON ALL RIGHTS RESERVED

飼の原料である魚粉の価格は15年で3倍に高騰し
飼料コストは総生産コストの50%を占める





UMITRON ALL RIGHTS RESERVED

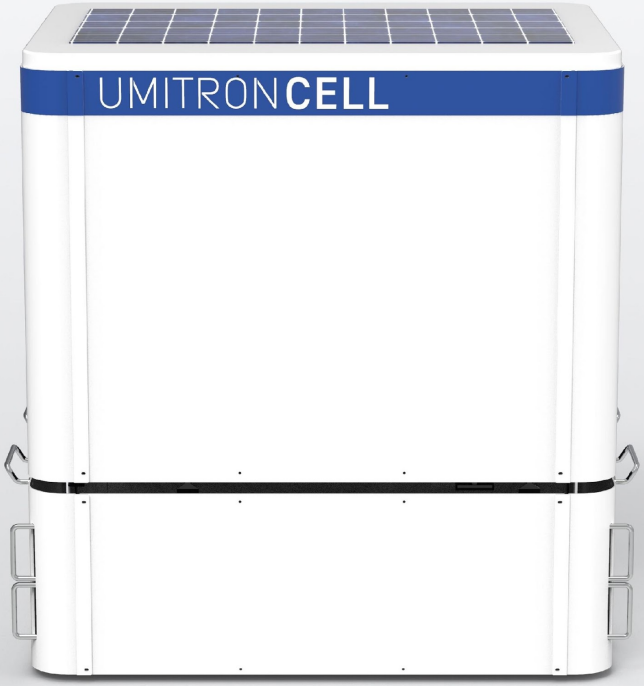
給餌効率をテクノロジーで最適化する

- 魚の摂餌行動から無駄餌の検知と削減
  - 機械学習で魚の摂餌行動を定量化
- 人手の制約を受けない魚の生育に最適な給餌
  - 日々変わる魚の食欲に対して最適な量・タイミングでの給餌を実現

無駄餌削減による飼料コストの削減

魚の早期育成による給餌効率の向上

UMITRON ALL RIGHTS RESERVED

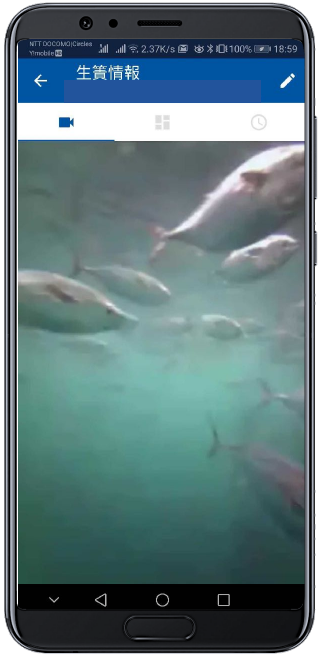


UMITRONCELL

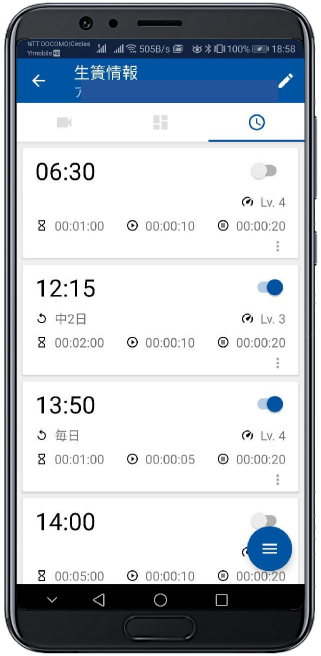
IoTと機械学習を用いた自動給餌器

UMITRON

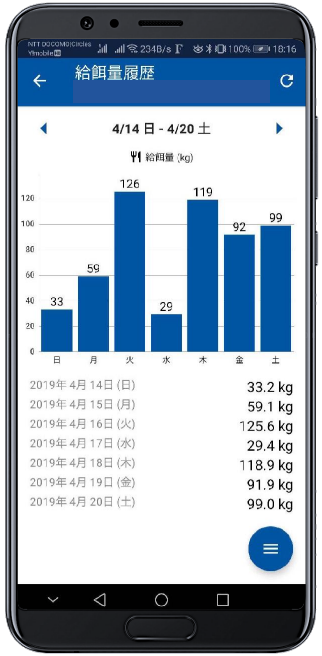
遠隔からの給餌制御とデータの蓄積



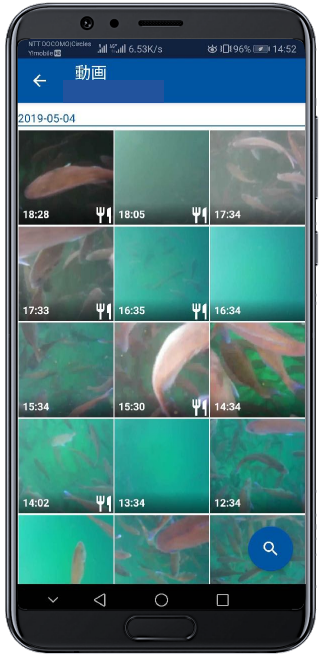
モニタリング



遠隔給餌制御



生育管理

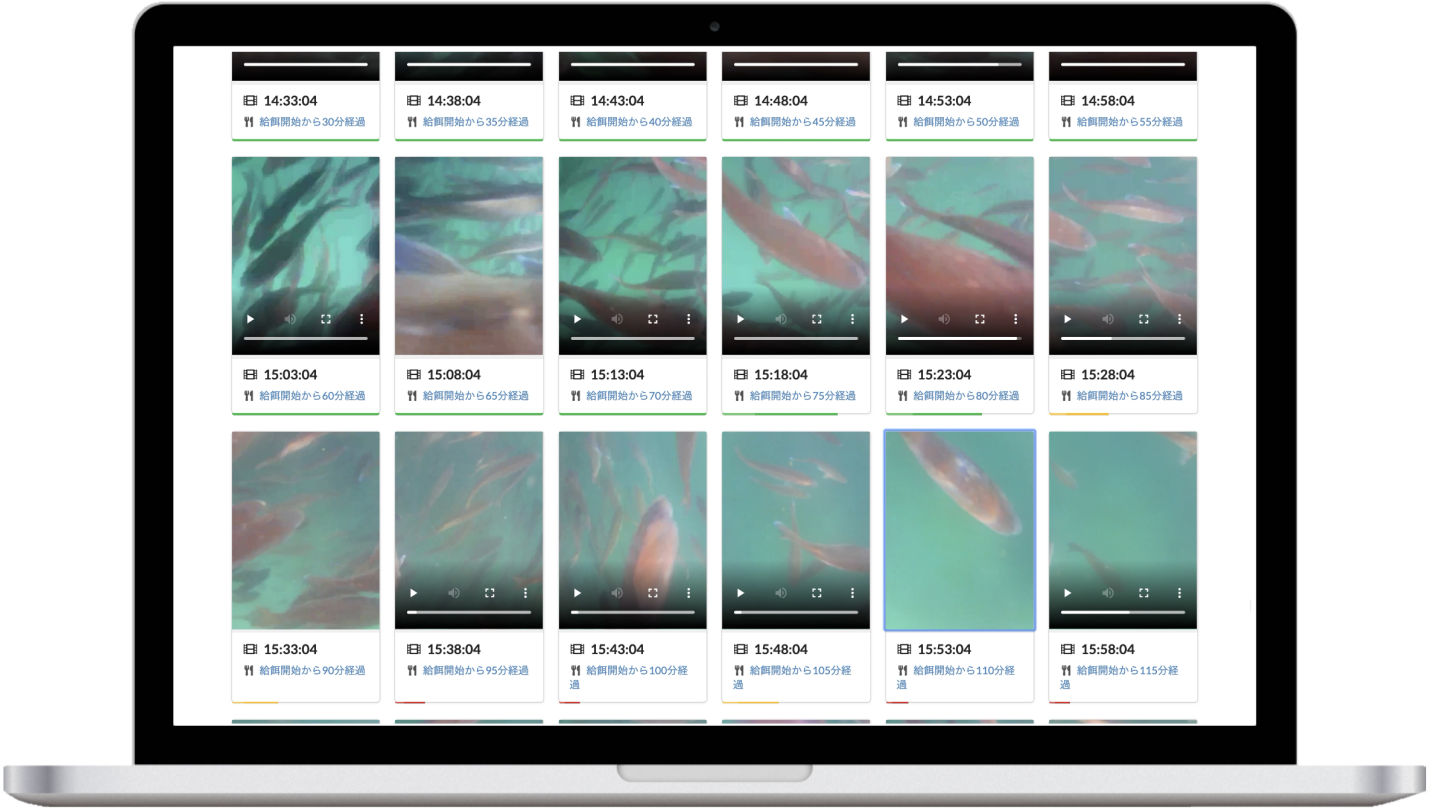


データ蓄積

UMITRON ALL RIGHTS RESERVED

UMITRON FAI

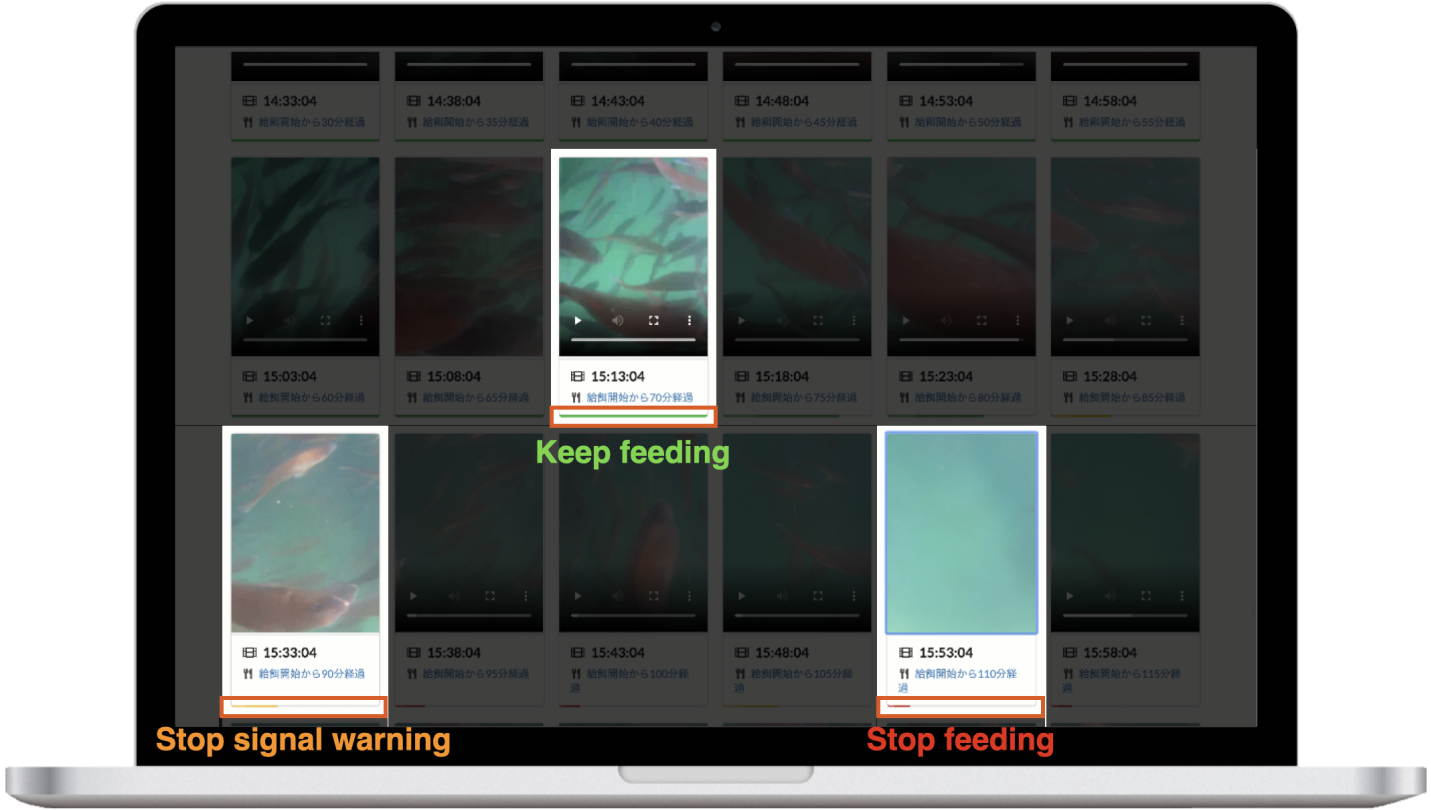
魚の食欲解析による給餌の最適化



UMITRON ALL RIGHTS RESERVED

UMITRON FAI

魚の食欲解析による給餌の最適化



UMITRON ALL RIGHTS RESERVED

給餌の最適化を実現させるために必要なこと

- IoTを用いた遠隔操作とデータ収集
  - 遠隔での給餌器制御
  - 安定したデータ取得
  - 制御ソフトウェアの継続的アップデート
- 機械学習を用いた魚の食欲解析
  - 取得したデータをリアルタイムで機械学習で処理する
  - 日々取得できるデータからの継続的な機械学習モデルの改善

UMITRON ALL RIGHTS RESERVED

IoTシステムを用いて実現したい機能

- リアルタイムな双方向通信
  - IoTデバイスからのデータ取得
  - 遠隔からデバイスの操作
- 洋上での独立・安定した稼働
  - 電源は太陽光のみ
  - ユーザもIoTデバイスの近くにいつもいるわけではない・すぐ行けるわけではない
- デバイスを設置してからも容易に改善を続けられる
  - 設置してから初めて分かる課題にも対応できる
    - さまざまな自然環境
    - 魚の生育データ

UMITRON ALL RIGHTS RESERVED

AWS IoTによるIoT基盤

- AWS IoTを基盤に採用
  - IoTデバイス、Webサーバ、ネイティブアプリの3者とのリアルタイム双方向通信
    - MQTT over WebSocket
    - 安定稼働
      - **インフラ運用がほぼいらない**
  - AWSの各種サービス連携
    - ログ保存といった、IoTデバイス -> Webサーバのリクエストを一部肩代わり
      - サービス全体の中で大部分のリクエストがこれ
      - ほとんどビジネスロジックを持たない
        - **開発は事業ドメイン固有のビジネスロジックに集中できる**

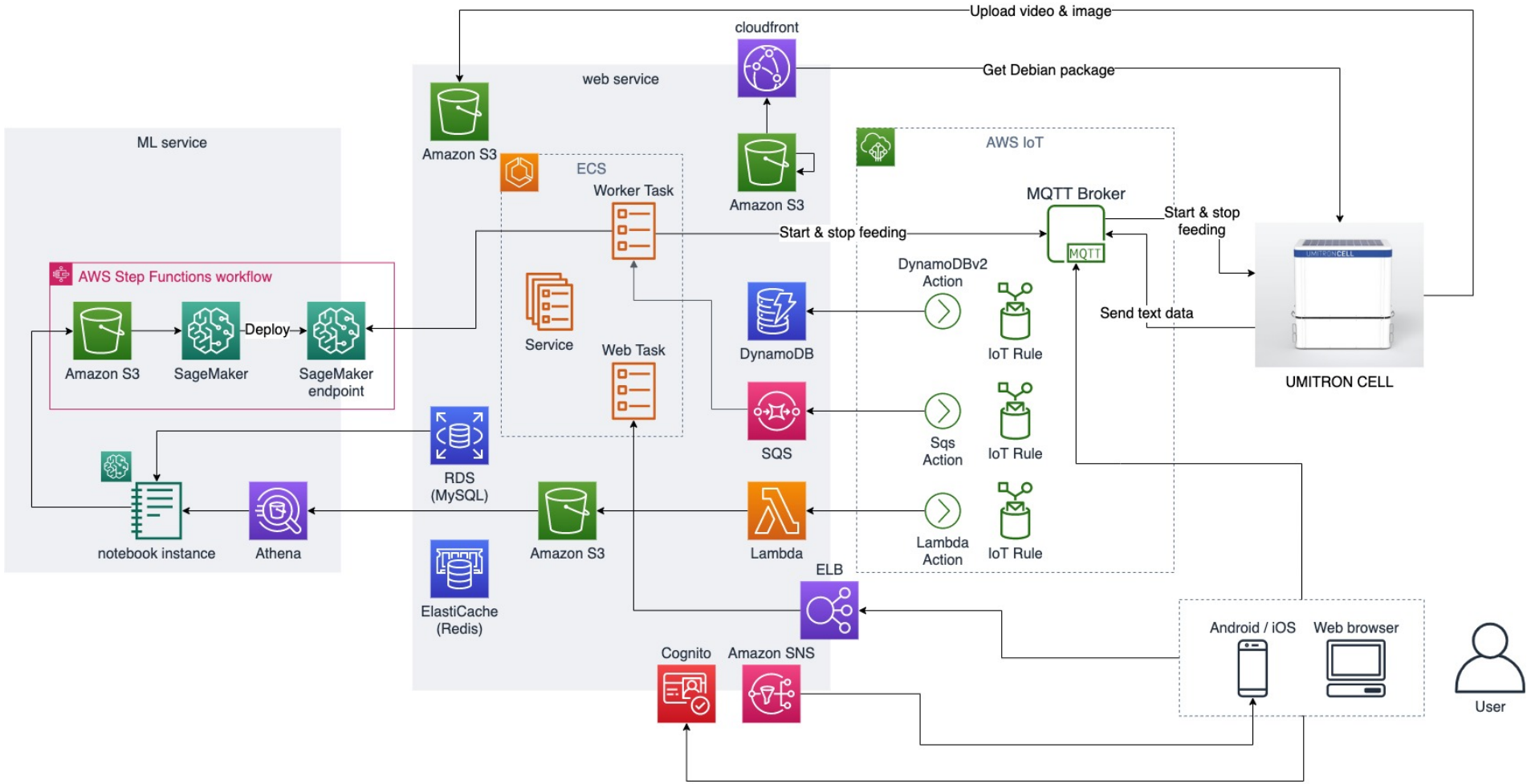
UMITRON ALL RIGHTS RESERVED

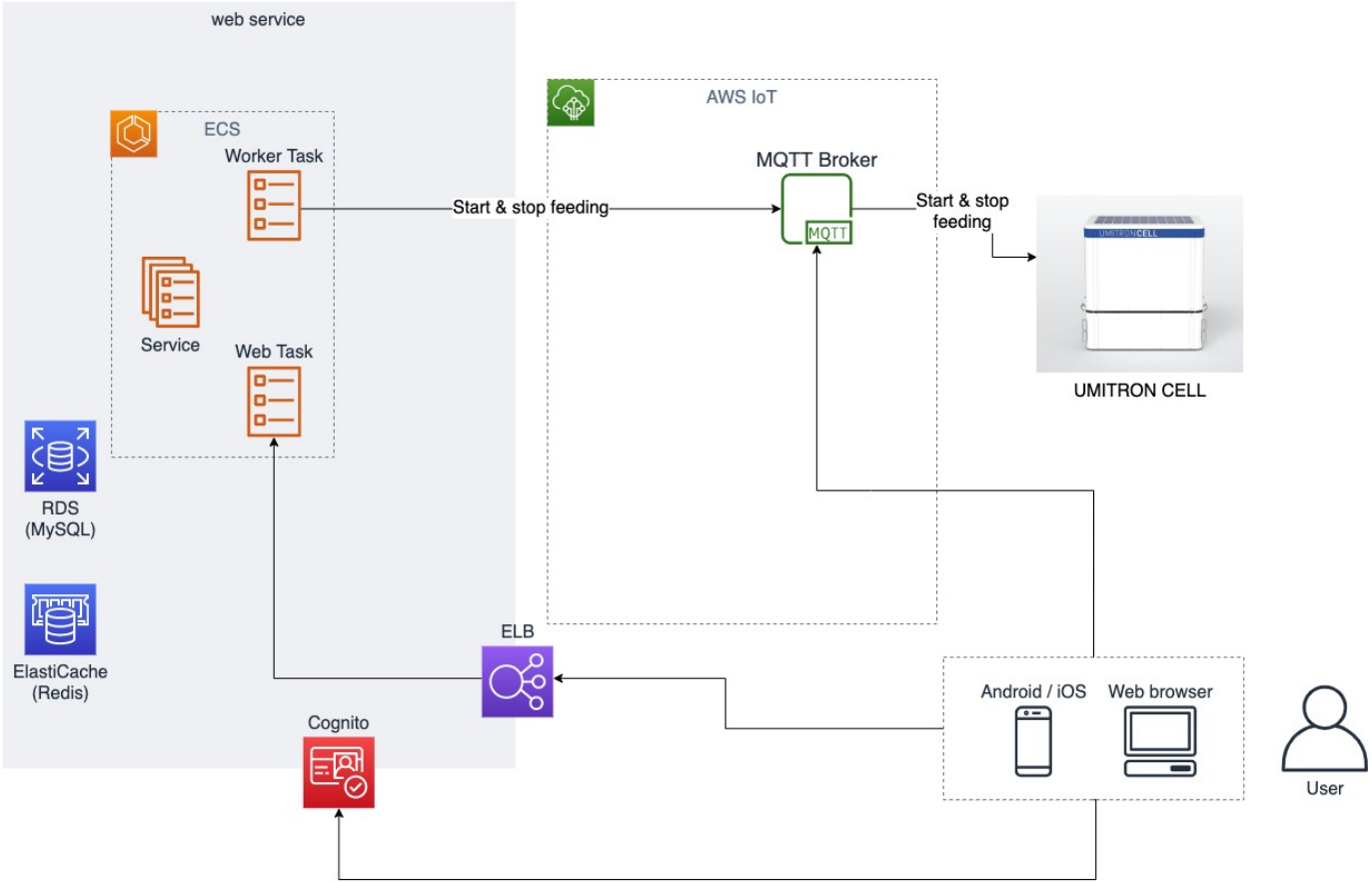
IoTデバイスの構成

- Raspberry Pi + Raspbian + Go
  - Raspberry pi + Raspbian
    - あとからソフトウェア変更が簡単なアーキテクチャ
      - IoTデバイス設置後も大幅な変更
      - **電力的にはモーターが大きい**
        - 省電力マイコンにこだわらない
    - 汎用品を使う
      - Linuxの既存資産を利用
      - 広く使われてるので、情報が多い
    - sshでトラブルシューティング
  - Go
    - 高い生産性 + パフォーマンス
    - バイナリ配布だけで動く、アップデートできる

UMITRON ALL RIGHTS RESERVED

UMITRON


Upload video & image
cloudfront
Get Debian package
web service
Amazon S3
ECS
Worker Task
Amazon S3
AWS IoT
MQTT Broker
Start & stop feeding
Start & stop feeding
MQTT
ML service
AWS Step Functions workflow
Amazon S3
SageMaker
Deploy
SageMaker endpoint
Service
Web Task
DynamoDB
DynamoDBv2 Action
IoT Rule
Send text data
UMITRON CELL
SQS
Sqs Action
IoT Rule
RDS (MySQL)
notebook instance
Athena
Amazon S3
Lambda
Lambda Action
IoT Rule
ELB
ElastiCache (Redis)
Cognito
Amazon SNS
Android / iOS
Web browser
User

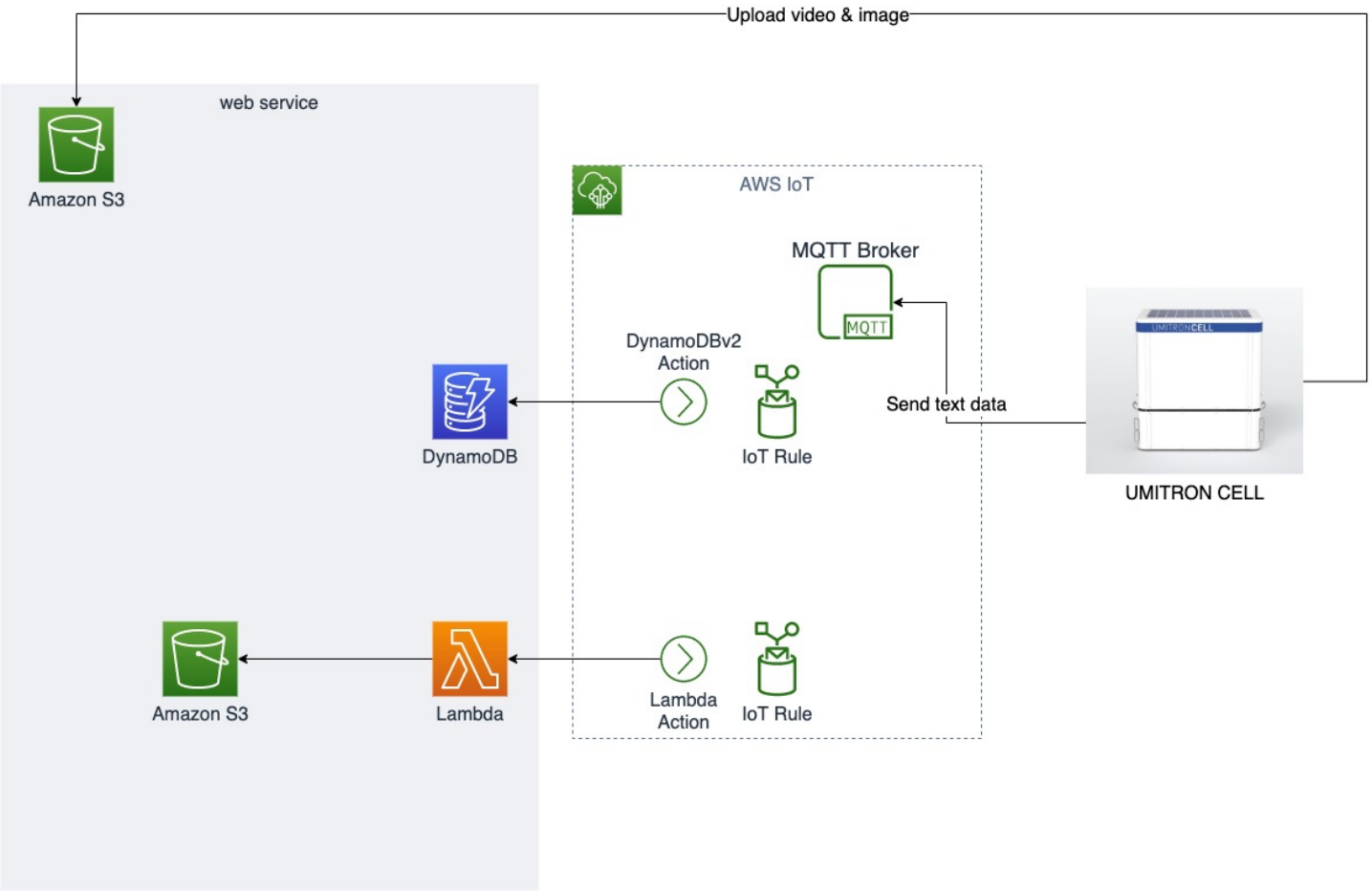


UMITRON ALL RIGHTS RESERVED

AWS IoTによるリアルタイム双方向通信

- Device Shadow
  - desiredにはWebサーバ、ネイティブアプリ -> IoTデバイスへのリクエスト
    - 給餌開始/停止命令
    - リアルタイムモニタリング開始/停止命令
  - reportedには IoTデバイスから今の状態を反映
    - 給餌の進捗状況
  - 現在のIoTデバイスの状態を確認するのにShadowだけ見ればいい
  - 非同期で命令をIoTデバイスに送れる
    - いつも万全なネットワーク状態ではない
    - ユーザが一時的にデバイスの電源を落としている場合もある

UMITRON ALL RIGHTS RESERVED

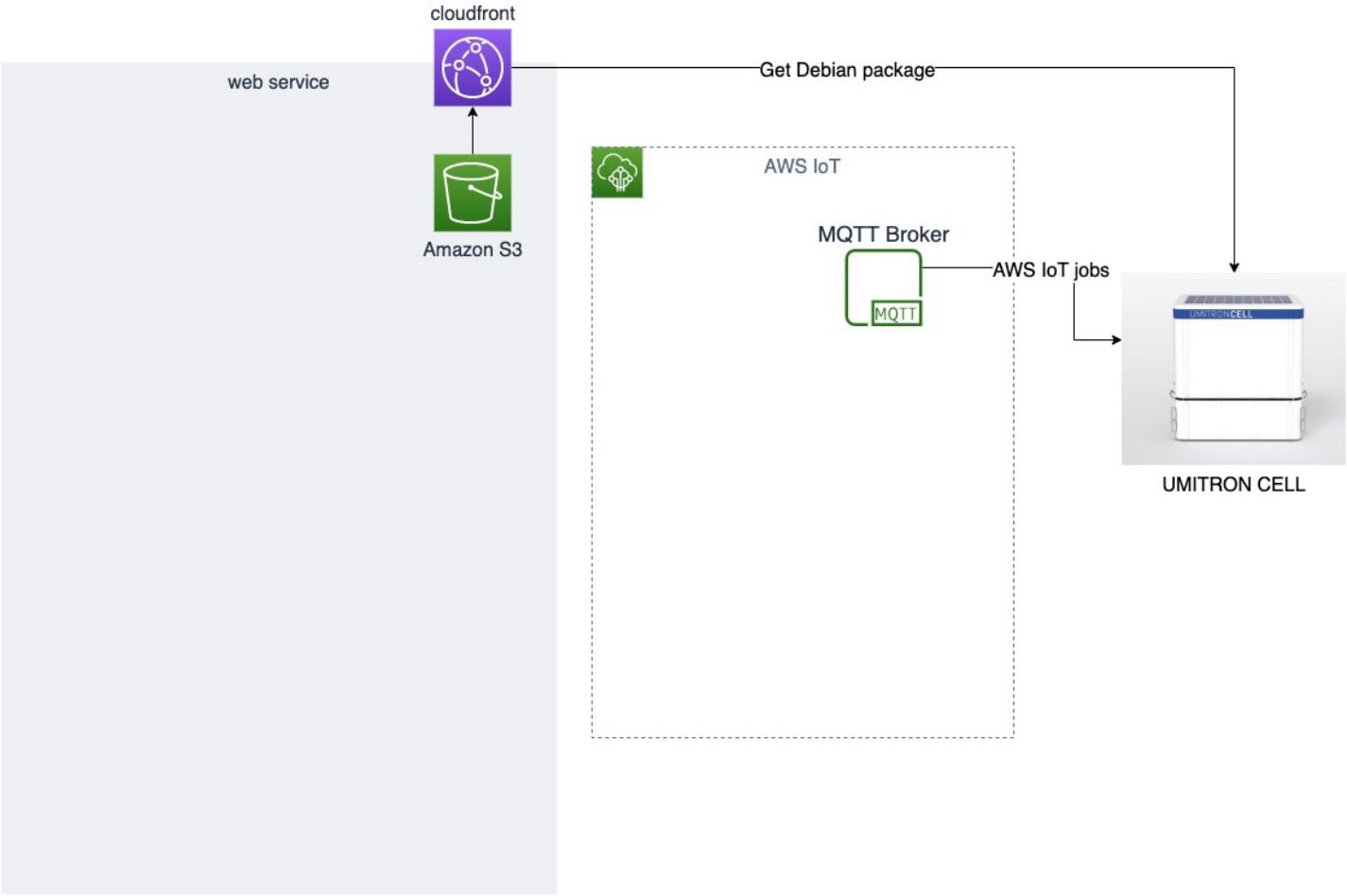


UMITRON ALL RIGHTS RESERVED

AWS IoTによるIoTデバイスのデータ収集

- AWS IoT Rule -> Amazon DynamoDB
  - AWS IoT -> AWS IoT Rule -> DynamoDB
    - テキストデータの保存
    - 一番多いリクエストがWebサーバを経由しない
  - DynamoDB
    - 大量のデータ保存
    - アプリケーションから参照できるくらい低レイテンシ
- Amazon S3
  - デバイスから直接S3にアップロード
    - 動画と画像の保存
    - PreSigned URLを利用
  - 大容量ファイルのアップロードがWebサーバを経由しない
    - IoTデバイスから動画のアップロードは時間がかかる

UMITRON ALL RIGHTS RESERVED



UMITRON ALL RIGHTS RESERVED

AWS IoTによるIoTデバイスのソフトウェアのデプロイ

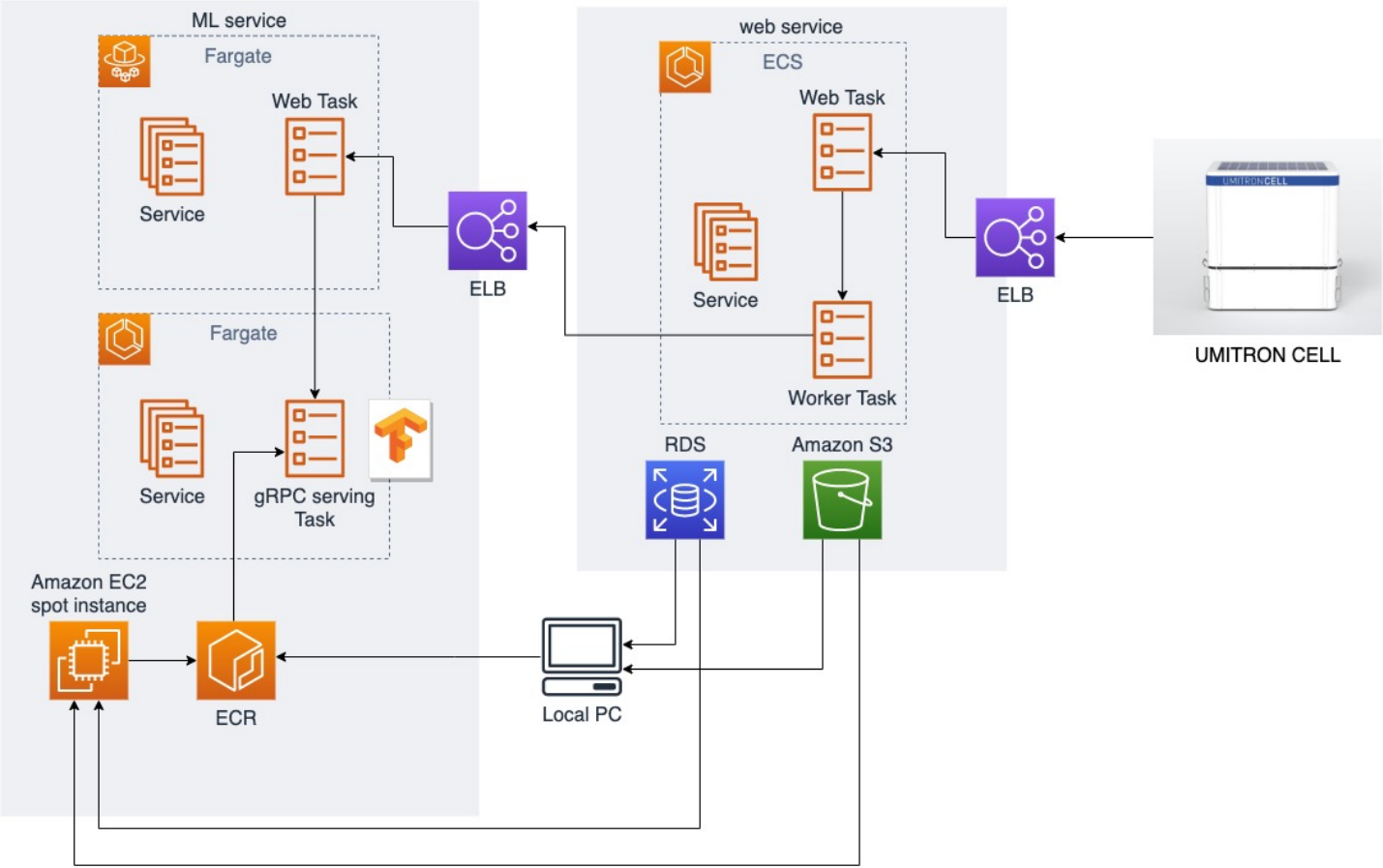
- Debian package + AWS IoT jobs
  - Debian package
    - Goのバイナリ + 使用ライブラリの依存 + 設定ファイル
    - packageはS3 + CloudFrontでホスティング
  - AWS IoT jobs
    - jobからDebian packageのバージョンを取得してapt-get
    - jobを作ったときにデバイスがオフラインでも、
      オンラインになったときにアップデートできる

UMITRON ALL RIGHTS RESERVED

機械学習システムの構築における課題

- **課題だったこと**
  - 開発速度
    - 継続的に機械学習サービスを改善するための開発環境
    - 属人化によるキャッチアップコストの高さ
  - 機械学習サービスのインフラパフォーマンス
    - 限られた人数での運用
    - Webエンジニアと機械学習エンジニアのスキルセットの差異
- **課題ではなかったこと**
  - 継続的なデータの収集
  - 実用に耐えうる精度のモデルの生成
  - 推論結果のアプリケーションでの利用

UMITRON ALL RIGHTS RESERVED



UMITRON ALL RIGHTS RESERVED

初期の機械学習システムの課題

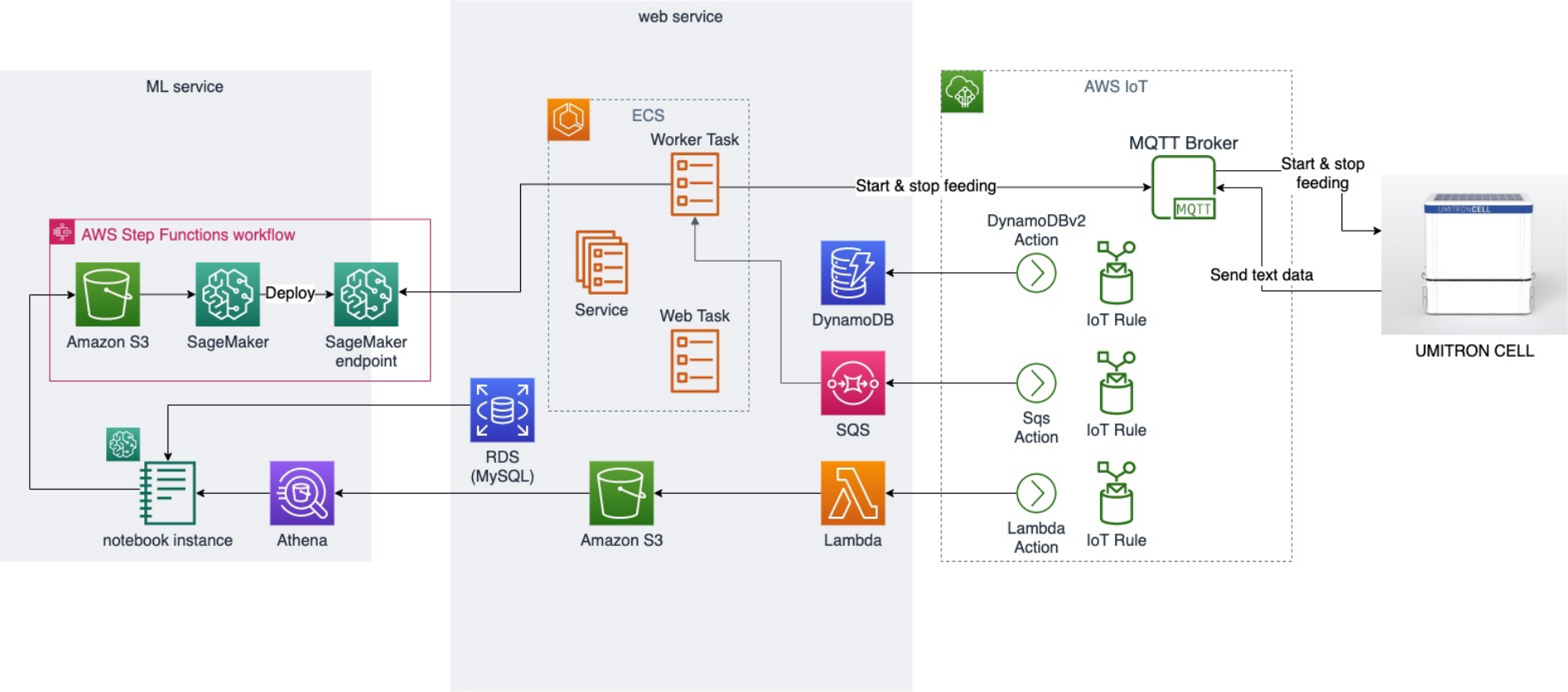
- 複雑な環境による開発の難しさ
  - 学習環境が統一されていない
    - ローカルPCとAmazon EC2 spot instance
  - 訓練データの管理方法が統一されていない
  - デファクトな構成でないため学習コストが高い
    - Webエンジニア、新入社員のキャッチアップコスト
- 機械学習サービスのインフラ運用の難しさ
  - Webサービスの負荷を機械学習サービスが捌けない
    - 機械学習サービスのパフォーマンスがユーザに影響を与える
    - 機械学習エンジニアだけインフラ運用できない

UMITRON ALL RIGHTS RESERVED

Amazon SageMakerを機械学習システムの基盤に採用

- フレームワークとしてのAmazon SageMaker
  - 統一された構成
  - 学習コストをAmazon SageMakerのみに集約
  - Webエンジニア、新入社員がキャッチアップしやすい
- フルマネージドなインフラ
  - インフラ運用コストの削減
  - 機械学習エンジニアでも運用できる
- Webサービスと機械学習サービスを疎結合にする
  - Amazon SageMakerのendpointで2つのサービスを結合
  - Amazon SageMakerの知識だけで互いを運用できる

UMITRON ALL RIGHTS RESERVED



UMITRON ALL RIGHTS RESERVED

Amazon SageMakerを採用した効果

- 開発環境の簡素化
  - 学習環境が統一
  - 必要なインフラ知識をAmazon SageMakerに集約
  - Amazon SageMakerさえわかれば誰でもわかる
  - 開発環境で社内特有の知識が少ない
- スケールが用意
  - インフラ運用コストが少ない
  - Webサービスが受ける影響がAmazon SageMakerのパフォーマンスのみに限定される
  - 機械学習エンジニアだけでインフラをスケールできる

UMITRON ALL RIGHTS RESERVED

今後の機械学習システムの発展

- 訓練データ作成の自動化
  - Athenaやnotebookでの訓練データ作成
- アノテーションの自動化
  - Amazon SageMaker Ground Truth
- レポーティングの自動化
  - 学習したモデルの評価

継続的に収集できる訓練データでモデルの改善イテレーションを高速化

UMITRON ALL RIGHTS RESERVED

Thank you!

岡本拓磨

CTO
ウミトロン株式会社

aws SUMMIT ONLINE

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.