

リアルタイムな双方向ファストデータ伝送を 実現する高速IoT基盤

- 開発・運用のこれまでとこれから -

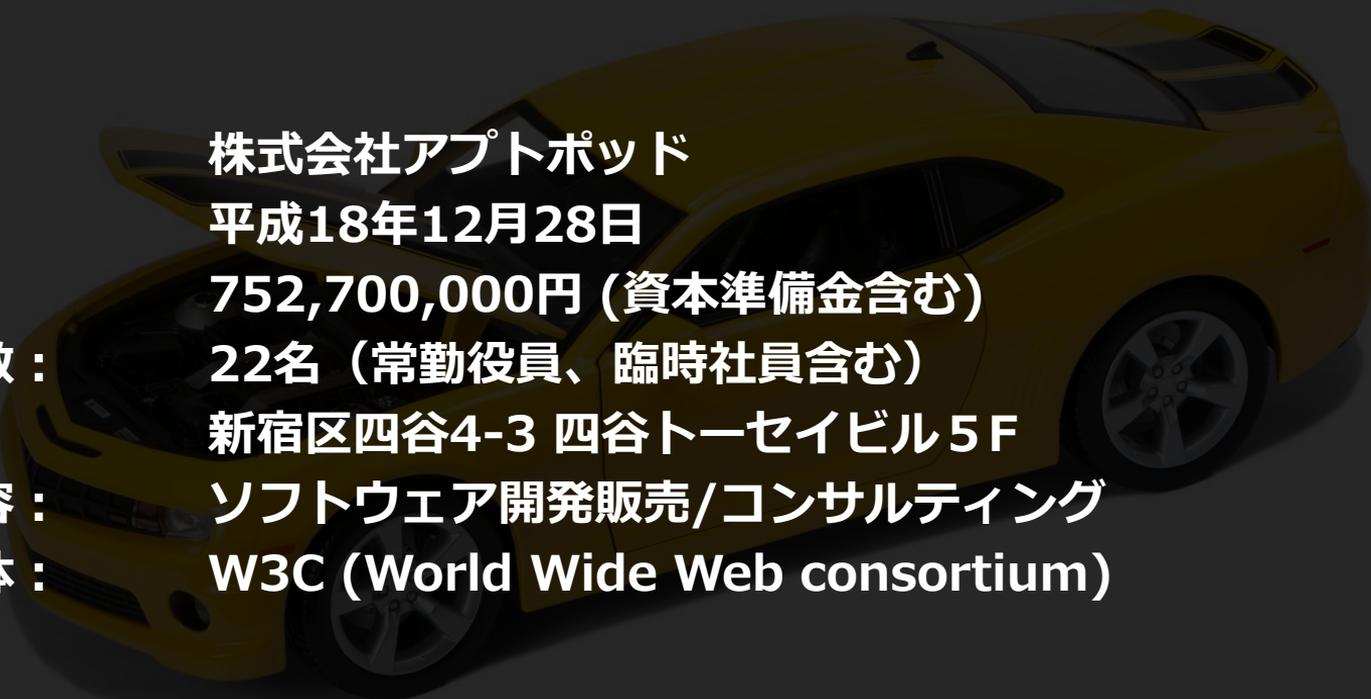
株式会社アプトポッド

代表取締役 坂元淳一

取締役CTO 梶田裕高

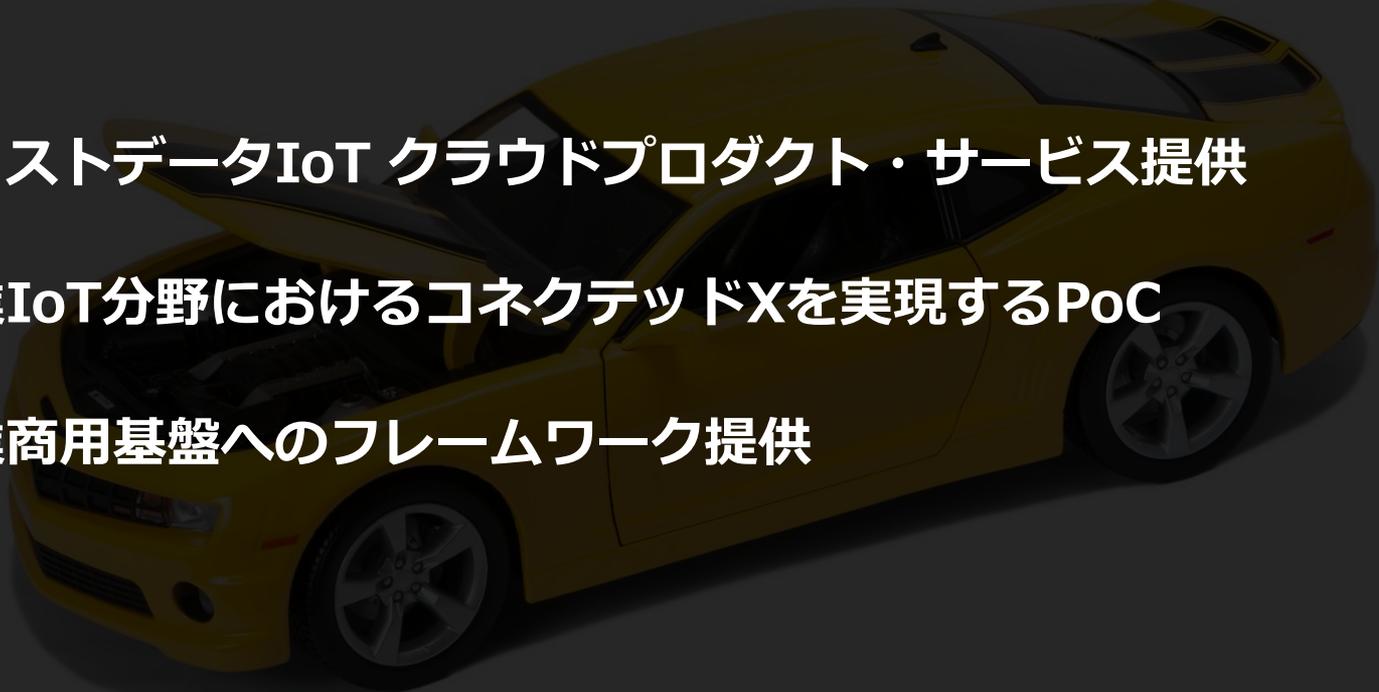
サーバー・インフラ リードアーキテクト 柏崎輝

簡単に会社概要ご紹介



社名：株式会社アプトポッド
設立：平成18年12月28日
資本金：752,700,000円 (資本準備金含む)
従業員数：22名 (常勤役員、臨時社員含む)
所在地：新宿区四谷4-3 四谷トーセイビル5F
事業内容：ソフトウェア開発販売/コンサルティング
加盟団体：W3C (World Wide Web consortium)

何をしている会社か？

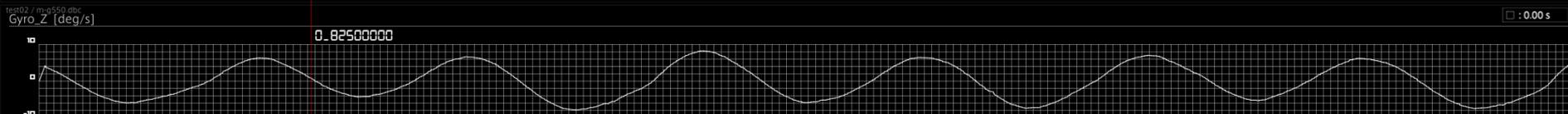
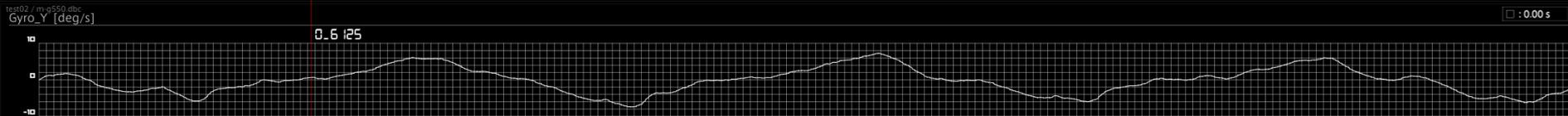
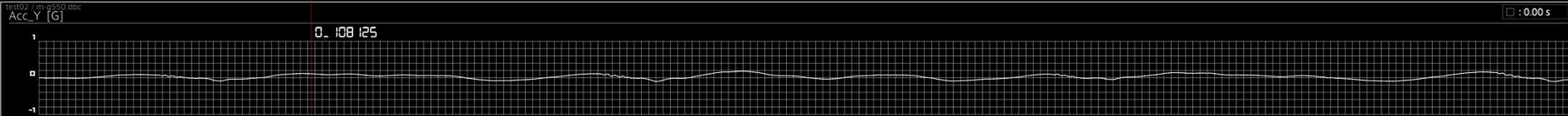
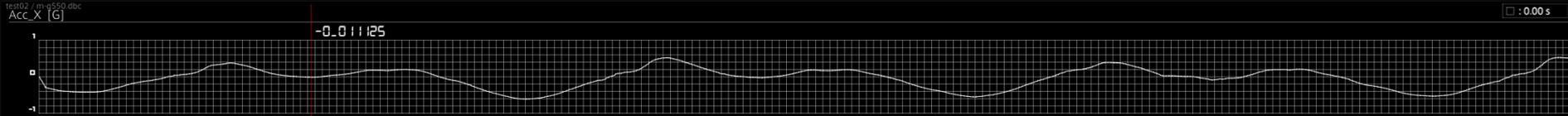
- **ファストデータIoT クラウドプロダクト・サービス提供**
 - **産業IoT分野におけるコネクテッドXを実現するPoC**
 - **産業商用基盤へのフレームワーク提供**
- 



なぜファストデータ?

なぜファストデータ？ 高精度6軸センサーデータ 1msecの場合(秒間)

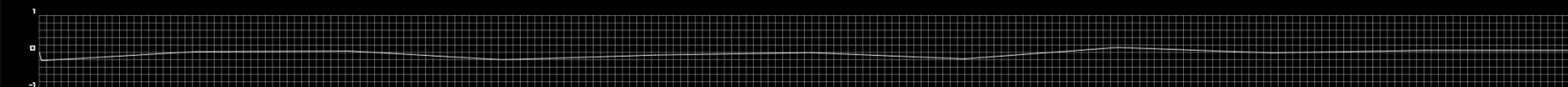
Project : - Screen : SE GRAPH



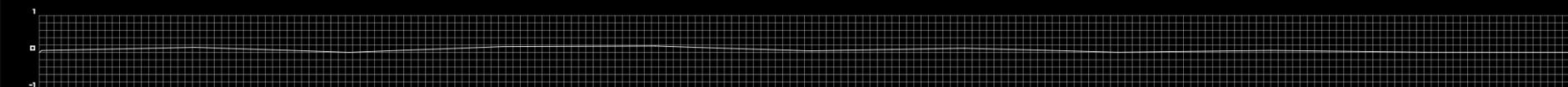
なぜファストデータ？ 高精度6軸センサーデータ 100msecの場合（秒間）

Project : - Screen : SE GRAPH

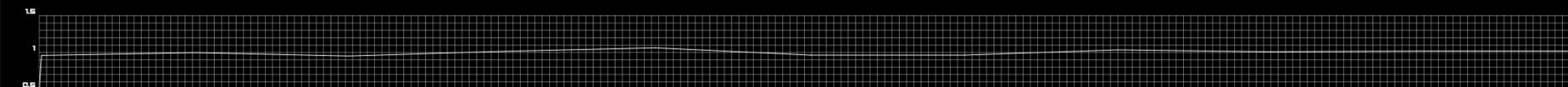
test02 / mu-a550 abc
Acc_X [G]



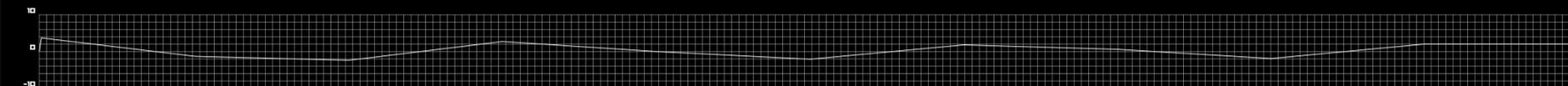
test02 / mu-a550 abc
Acc_Y [G]



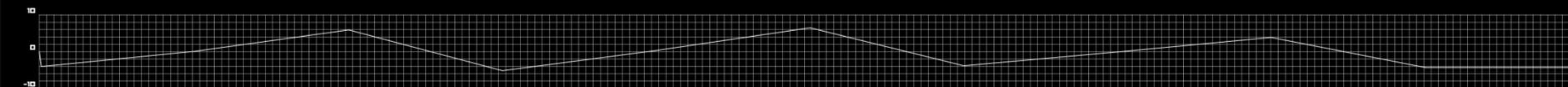
test02 / mu-a550 abc
Acc_Z [G]



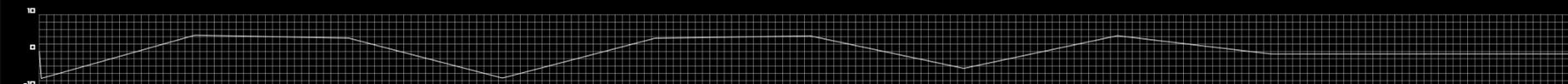
test02 / mu-a550 abc
Gyro_X [deg/s]



test02 / mu-a550 abc
Gyro_Y [deg/s]



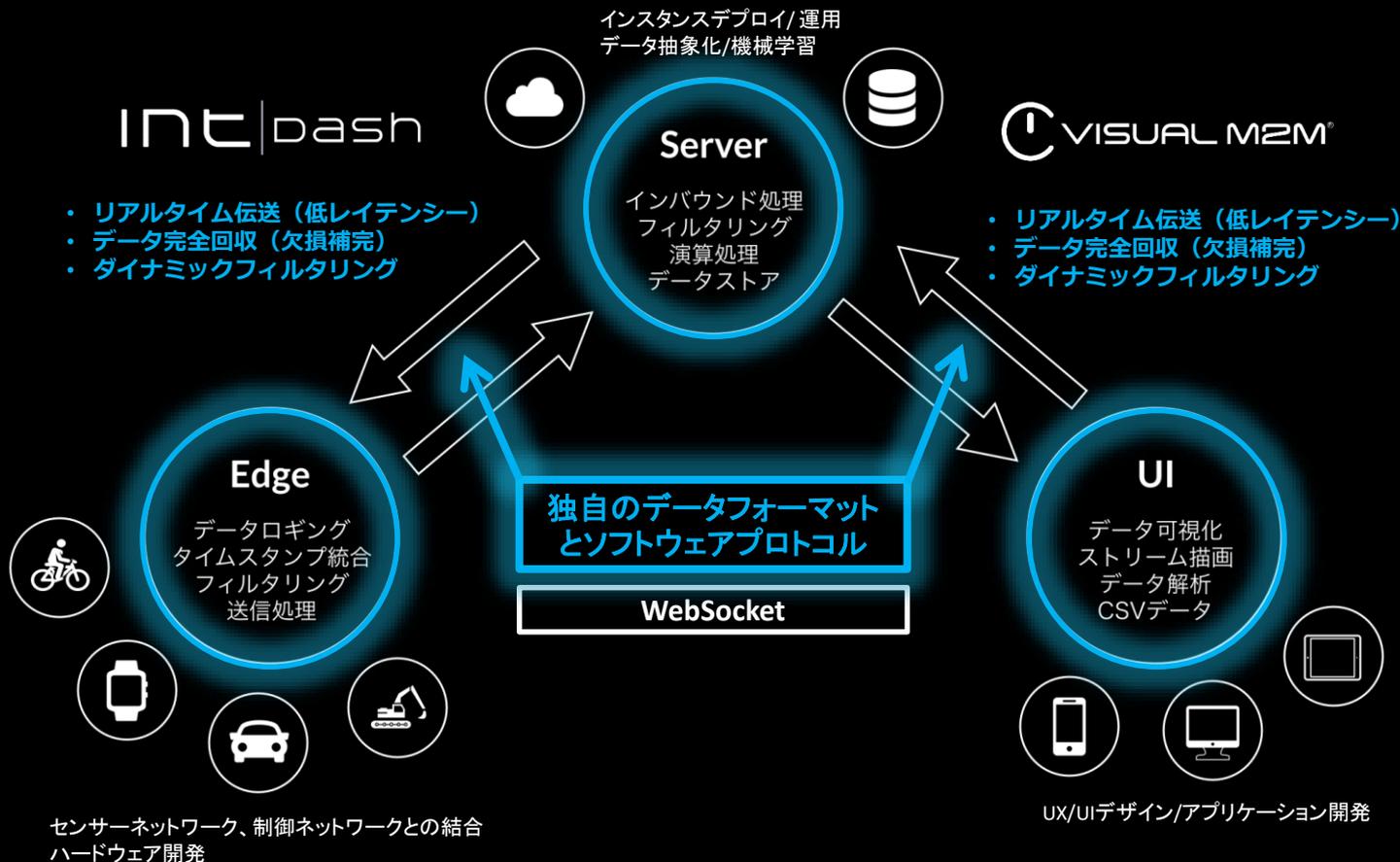
test02 / mu-a550 abc
Gyro_Z [deg/s]



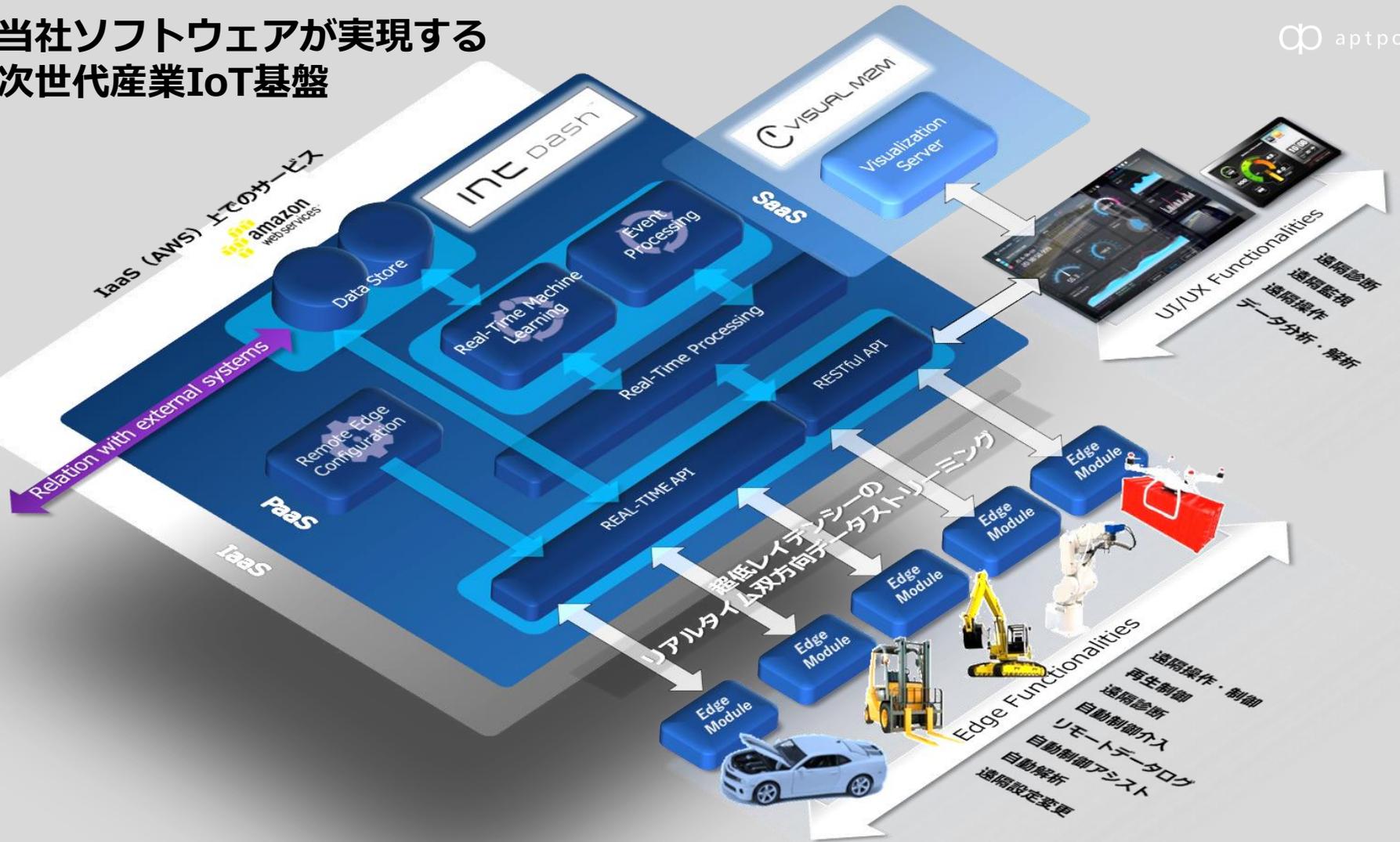
当社の提唱

“Internet of Behavior”





当社ソフトウェアが実現する 次世代産業IoT基盤



データビジュアライゼーション



THE INNOVATION FINDER



エッジプロダクト

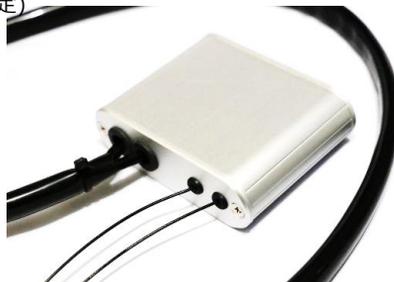
エッジハードウェア



車載ターミナルハードウェア
NEXCOM VTC100アプライアンスモデル

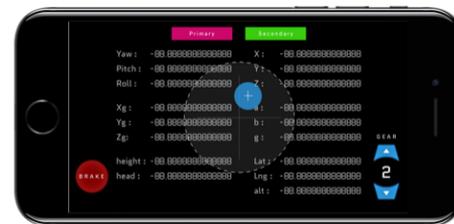


車載ターミナルハードウェア(自社開発：年度内出荷予定)



小型ターミナルハードウェア(自社開発)

スマートフォン エッジアプリ



DEMO VIDEO





自動車

- 自動運転をサポートする遠隔制御介入
- 機械学習によるリアルタイム故障診断



重機・建機・農機

- 遠隔オペレーションによる僻地作業
- 運転データモデルによる自動反復制御



ファクトリーオートメーション

- 複数拠点の稼働状況に連動した集中制御
- 工作機械のリモートメンテナンス



ロボティクス

- 遠隔制御介入
- リモートメンテナンス



ドローン

- 3D地図データと連動したロジスティクス



メディカル・ヘルスケア

- 精密データによる遠隔診断
- リアルタイムデータモニタリングによる自動診療



スポーツ

- 挙動データのリアルタイム解析による自動レビュー
- 遠隔ティーチング・コーチング



各種産業計測/研究開発

- 精密データの自動遠隔回収
- データ分析・解析ワークフロー

当社の技術チャレンジとアーキテクチャについて

株式会社アプトポッド
取締役/ CTO
梶田裕高

データロギング系

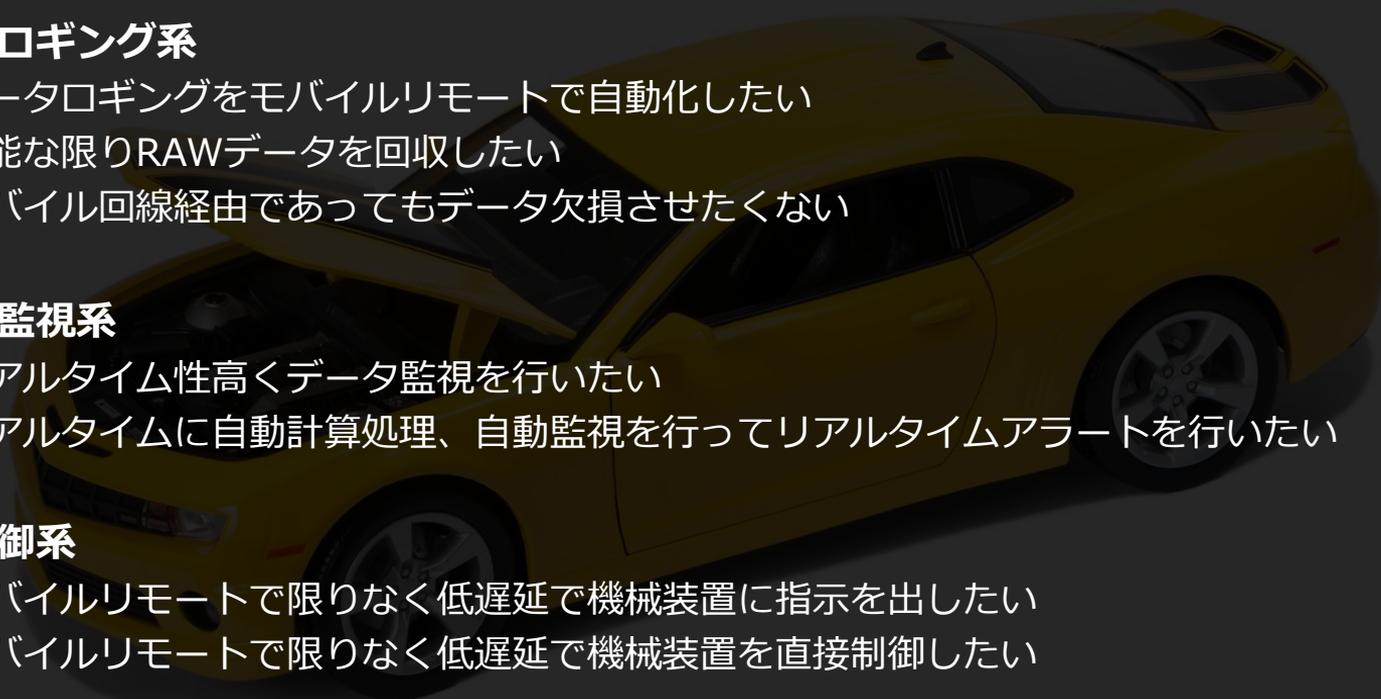
- データロギングをモバイルリモートで自動化したい
- 可能な限りRAWデータを回収したい
- モバイル回線経由であってもデータ欠損させたくない

データ監視系

- リアルタイム性高くデータ監視を行いたい
- リアルタイムに自動計算処理、自動監視を行ってリアルタイムアラートを行いたい

遠隔制御系

- モバイルリモートで限りなく低遅延で機械装置に指示を出したい
- モバイルリモートで限りなく低遅延で機械装置を直接制御したい



世界最高速のIoTを目指して 当社の技術チャレンジ

1. 高粒度データストリーミング

- ミリ秒データのストリーム処理

2. 双方向リアルタイム伝送

- 超低レイテンシーの双方向伝送処理

3. リアルタイムインテリジェンス

- 変動する通信帯域に対応するダイナミックフィルタ処理
- 低サンプリングデータをRAWレベルに再現する予測補完モデルの実装
- 機械学習による予測モデル生成

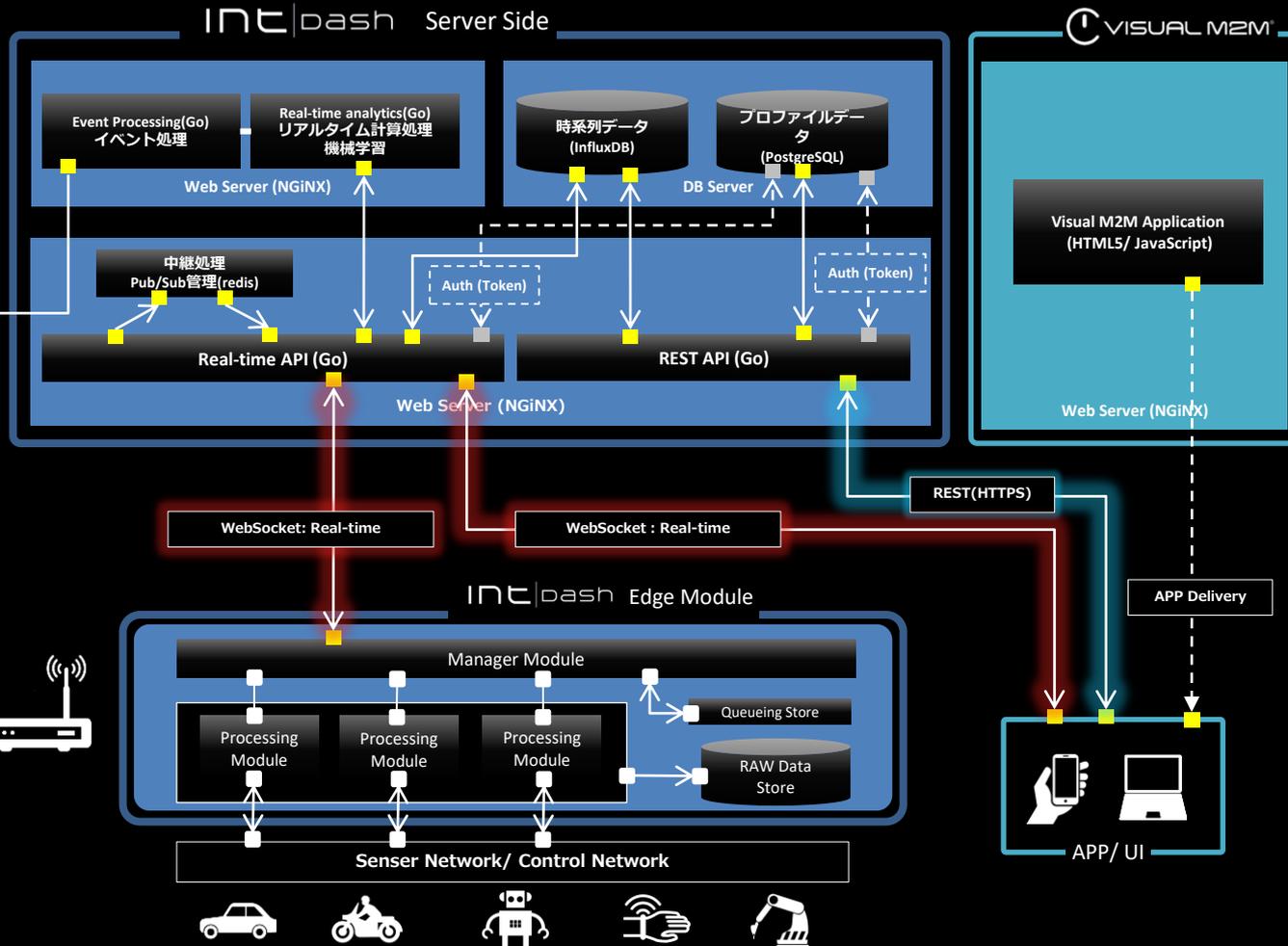
4. データ欠損補完

- 不安定な回線でもRAWデータを完全回収するデータ欠損補完

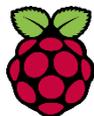
5. モバイル伝送最適化

- 2G,3GからLTEまで、モバイル帯域への最適化

Product Architecture



エッジ開発 センサーネットワーク



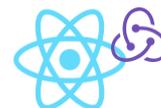
サーバー・インフラ開発



ANSIBLE



アプリケーション・UI/UX 開発



“End to End” coverage

Infrastructures



Frameworks / Tools



Communications



languages



Workflows



- **エッジからサーバー、UIまで一貫したポリシーでの設計と実装**
 - マルチレイヤでのアーキテクチャ統制
- **汎用的な産業データへのコミット**
 - CANなどの代表される産業データ
 - 未知の産業へのチャレンジ
- **広がる要件のコアプロダクトへの吸収**
 - コアプロダクト開発とPoC毎のカスタマイズ要件

産業特化のリクワイアメントに対して 全レイヤのエンジニアが共同で対峙

• 産業特化へのチャレンジ

- 自動車/建機におけるCAN, KWP, CCPなどのプロトコルへの対応
- ロボティクスにおけるCAN \leftrightarrow ROS変換など 産業分野別の標準技術・特化技術への対応

• 様々なセンサーインテグレーションに付随する調査と実験

- 高精度GPS (RTK), 各種アナログセンサー・デジタルセンサー など

• ネットワーク帯域、レイテンシーとの闘い

システム全体アーキテクチャポリシー

- 小さなサービスの組み合わせにより構成できるようにする：疎結合
- 最小1台構成でも問題なく柔軟に構成できる：外部依存を避ける
- 他のサービスと独立してデプロイできるようにする
- 他のサービスによる障害の影響を分離することができるようにする
- 特定の技術へのロックインを避けるようにする
- オンプレミスとの併用運用も考慮し、クラウドサービスのみ依存しないようにする

- データフォーマットの統一

- 規約の統一

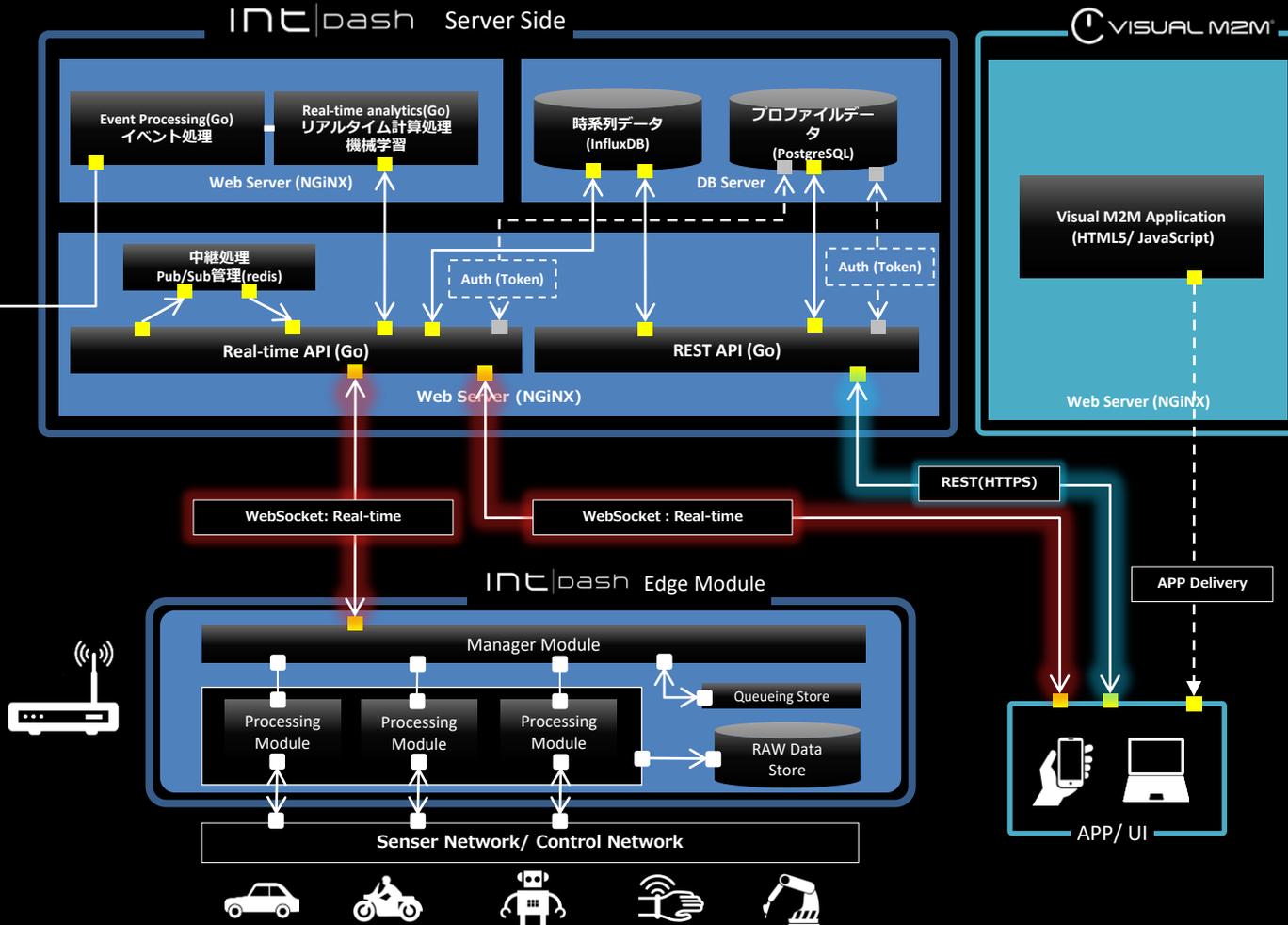
- Anywhere プロセッシング

- エッジ、サーバー、UIのどこでも、計算・イベント処理・フィルタリング・解析等の処理など、他のサービスを加えられるような設計

Product Architecture



- Mail/SMS
- メッセージ(MQTT他)
- APNS(Push通知)



なぜサーバーサイドにgoを採用したか

- クロスプラットフォームで動作する
- シンプルで学習コストが低い
- チームとしてやりやすい
- シンプルな言語でコンパイルおよび実行速度が早い
- 実行ファイル形式（バイナリ）なのでC言語に近い速さで動作する
- 標準ライブラリも充実
- 並列処理に優れている
- クロスコンパイルが便利 = クロスプラットフォーム対応が容易
- 比較的安全性が高い

なぜInfluxDBを採用したか

- 時系列に特化したDBなので、ファストデータのハンドリングにおけるパフォーマンスが高い
- 他サービスへの依存も少なく、データストア単体での切り離し、他環境への移行もしやすい
- 計算、集計等のメトリクス、イベント保存等解析にも適している

ファストデータにおける当社のプロトコル選択

これまでの実装

- HTTP → データ回収（非リアルタイム）
- MQTT → 一部データのリアルタイム伝送（連続データ系）

発生した課題

- データ種数の肥大化
- リアルタイム伝送対象のデータ数の増大
- 双方向データ

現在の実装

- WebSocket上にて独自プロトコルでの通信を行い、データ回収（非リアルタイム）とリアルタイム全二重伝送（連続データ系）を統合実現

ファストデータストリーミングにおける 通信プロトコルの課題と解決

株式会社アプトポッド
サーバー・インフラ リードアーキテクト
柏崎 輝

- **高頻度のデータすべてにヘッダーがつく**

- メッセージヘッダの最小サイズは軽量 (2バイト~) と言われているが、実際はトピック長分のサイズがメッセージ毎に付与される
- 例: トピック長が 16 Bytes, 1kHz データ
 - $(2+2+16) * 1000 = 20000$ Bytes/s
 - 1 秒で約 20KBytes がオーバーヘッドとなる

- **到達保証 (QoS) とリアルタイム性**

- 到達保証がクライアントとサーバ間で、別のクライアントまで保証されるわけではない
- 到達保証(QoS)を 1 or 2 に設定するとリアルタイム性が損なわれる

- 
- **セキュリティがMQTTブローカー依存**
 - topicへのpermissionや認証、セッション管理
 - システム全体として連携が必要となる：別管理
 - **データの送付先がブローカーを介すクライアントとした位置づけとなる**
 - ブローカーを介す分、内部処理コストが高くなり、レイテンシもその分かかってしまう

独自プロトコルの開発



- **通信プロトコルはWeb Socketを使用**

- APIサーバとの親和性を高く

- **リアルタイム性と回収性能の両立**

- 高頻度のデータを1本のコネクションで受信してデータの完全性を確保しながらリアルタイム処理への影響を最小化するためのフォーマット/規格

- **エッジ-サーバー間におけるネゴシエーション**

- 1接続で複数のストリームを開く
 - 宛先設定

- **ヘッダを極限まで減らす**

独自プロトコルにおけるデータフォーマット

基本構成

- 処理はSection単位で行われる
- SOS MarkerとEOS Markerにより再送かどうかどこが欠損しているかデータとして完全性が保たれているかが瞬時に判断可能となる



独自プロトコルにおけるデータフォーマット

リアルタイム性と回収性能の両立

- リアルタイム処理は、認識した Unit を即時に伝送する
- 回収性能面では、Section単位で処理を行う
- データの初めから終わりまでは、計測という単位で処理され、途中の切断や欠損も適切に処理する



切断や輻輳により送れなかった
(あとから再送)

リアルタイム処理を優先

データ条件

- CAN (データ部13Bytes) 1kHzのデータをEdge⇔Serverで往復

比較環境

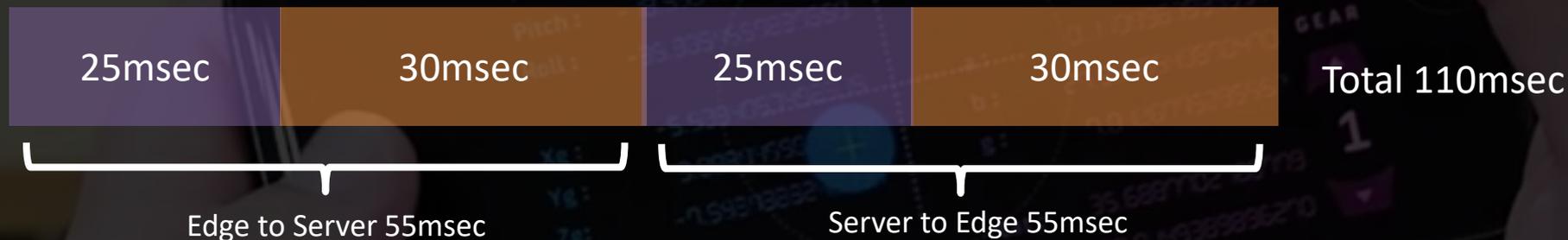
- MQTT (リアルタイムストリーム) + HTTP (回収処理)
- 独自プロトコル on WebSocket (リアルタイムストリーム +回収処理)

検証内容

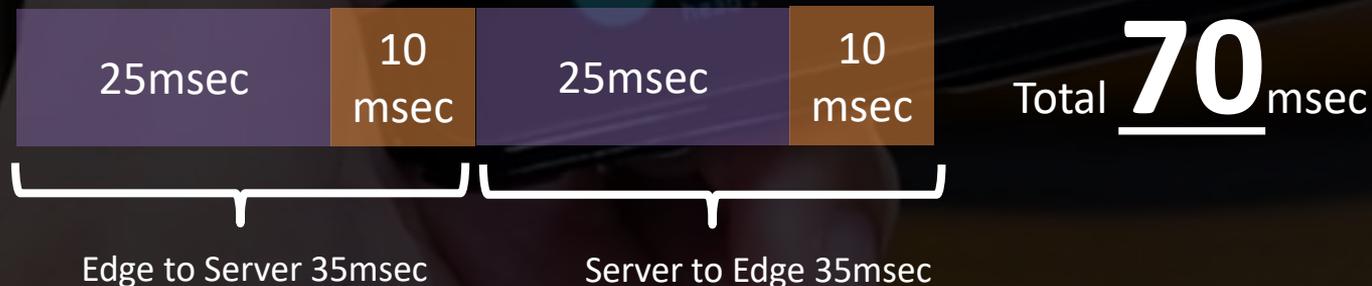
- モバイル回線環境再現 + 帯域制限
 - レイテンシー計測
 - 帯域制限による安定伝送の限界点計測

レイテンシー検証 ソフトウェアレイテンシーの最小化に成功

MQTT(リアルタイム)+HTTP(回収処理)



WebSocket + 独自プロトコル (リアルタイム+回収処理)



通信レイテンシー

ソフトウェアレイテンシー

安定伝送検証

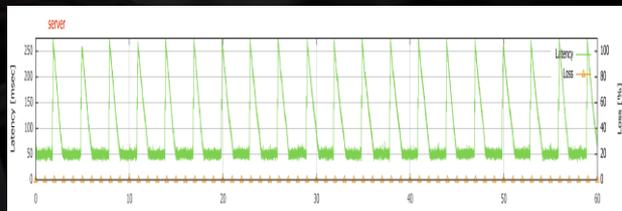
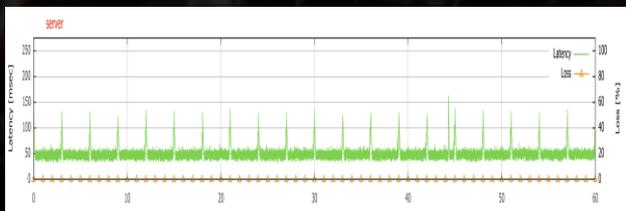
→ ナローバンド時のストリーム安定化に成功

MQTT(リアルタイム)+HTTP(回収処理)

1.5Mbps

800kbps

安定限界= 約1Mbps

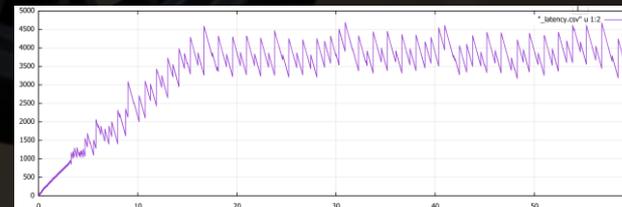
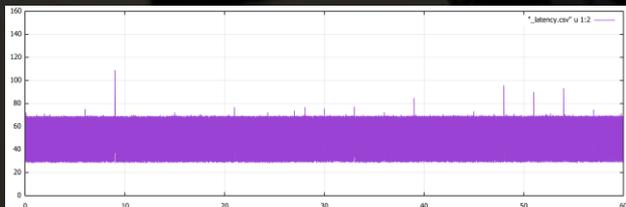


WebSocket + 独自プロトコル (リアルタイム+回収処理)

300kbps

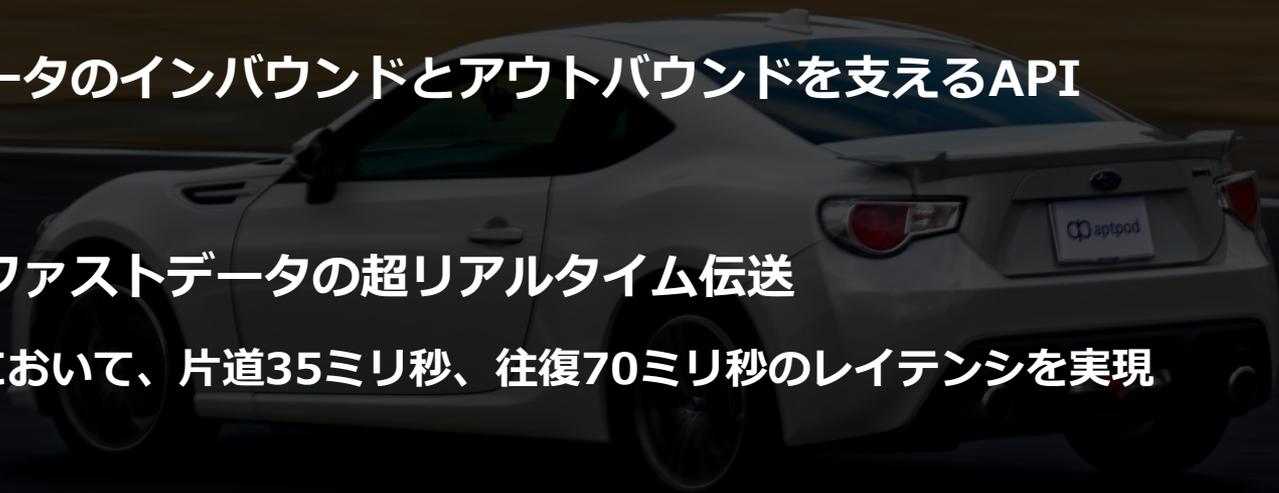
200kbps

安定限界=約 **300** kbps



結果的にここまでできました！

- リアルタイム性とデータ回収性の両立
- 超高速な時系列データのインバウンドとアウトバウンドを支えるAPI
- モバイル経由でのファストデータの超リアルタイム伝送
 - LTE通信環境時において、片道35ミリ秒、往復70ミリ秒のレイテンシを実現



なぜAWS？

- プロジェクト毎に乱立するサーバと管理

- ↳ 顧客によっては別のクラウドやオンプレミスパターンも

- 日本だけでなく海外リージョンを利用するプロジェクトも多い

- 成長型のプロジェクトにおけるスケーラビリティシナリオの確保

- **公式 SDK・サードパーティを含めた周辺ツールの充実**

- Infrastructure as Code 環境が整っている

- **Terraformの利用**

- コードとして書く事で、レビューを通せる
- コンソール手動操作のオペミス排除
- セキュリティグループや IAM Role の乱立を防止

- **マルチクラウド横断**

- terraform の記述方法さえ覚えれば、どこでも使用可能
- 要望にてGCEで構築したが、ドメインはRoute53を使いたい、等

- 16 のリージョンによって、グローバル展開が容易

- サイジングの事前見積やハードウェア保守からの開放

 - EBS の変更機能は本当に嬉しい

- 自由なネットワーク設計

 - パブリック・プライベートサブネット・CIDRも自由

AWS Service連携

将来的なチャレンジ・構想



・ Greengrass 連携

– メッセージングは弊社技術を使いたい

– リモートコンフィギュレーション・ファームウェアアップデート・証明書配布等、いいところ取りをしたい

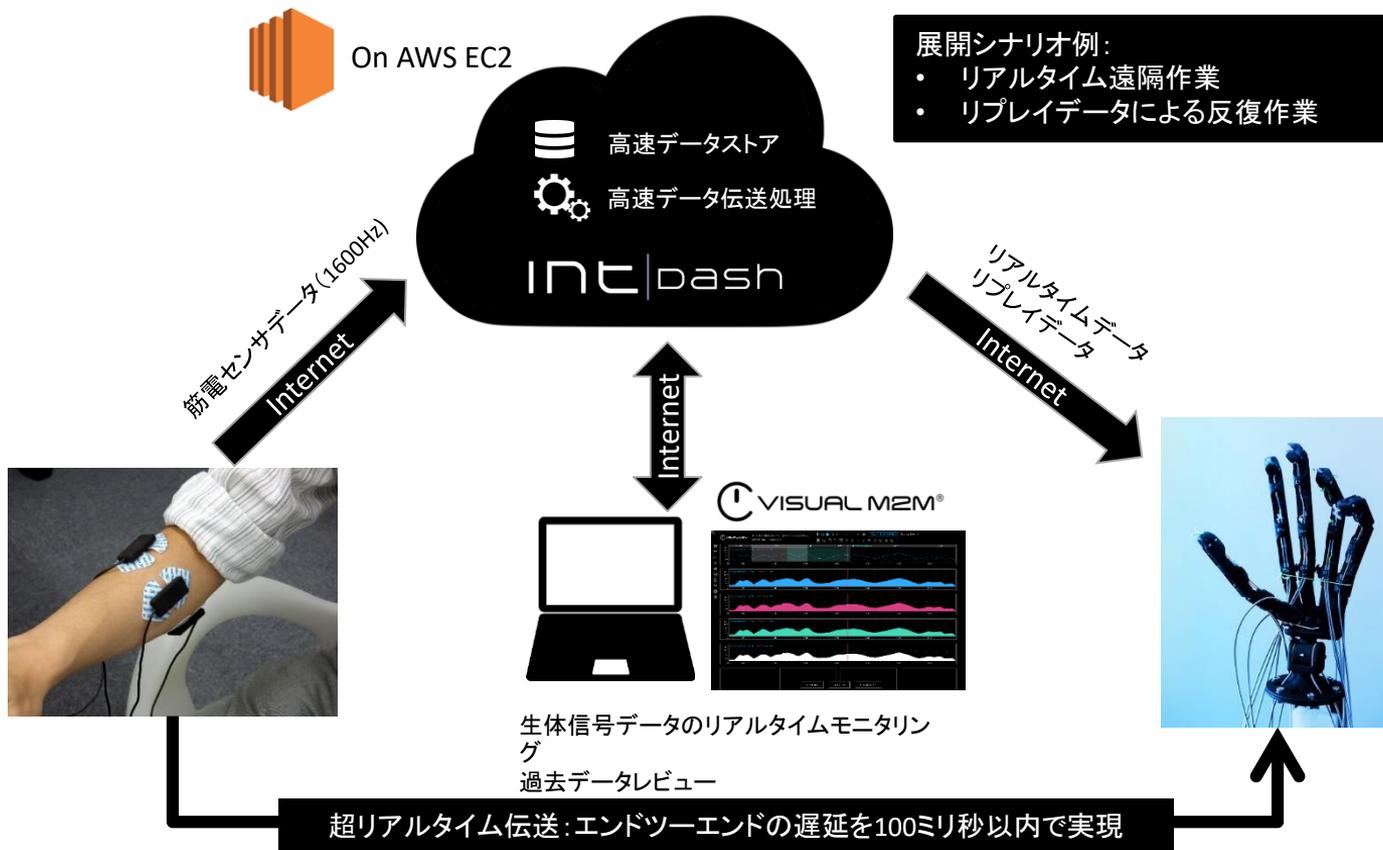
最後に

お知らせ

Pavilionブース展示してます！！

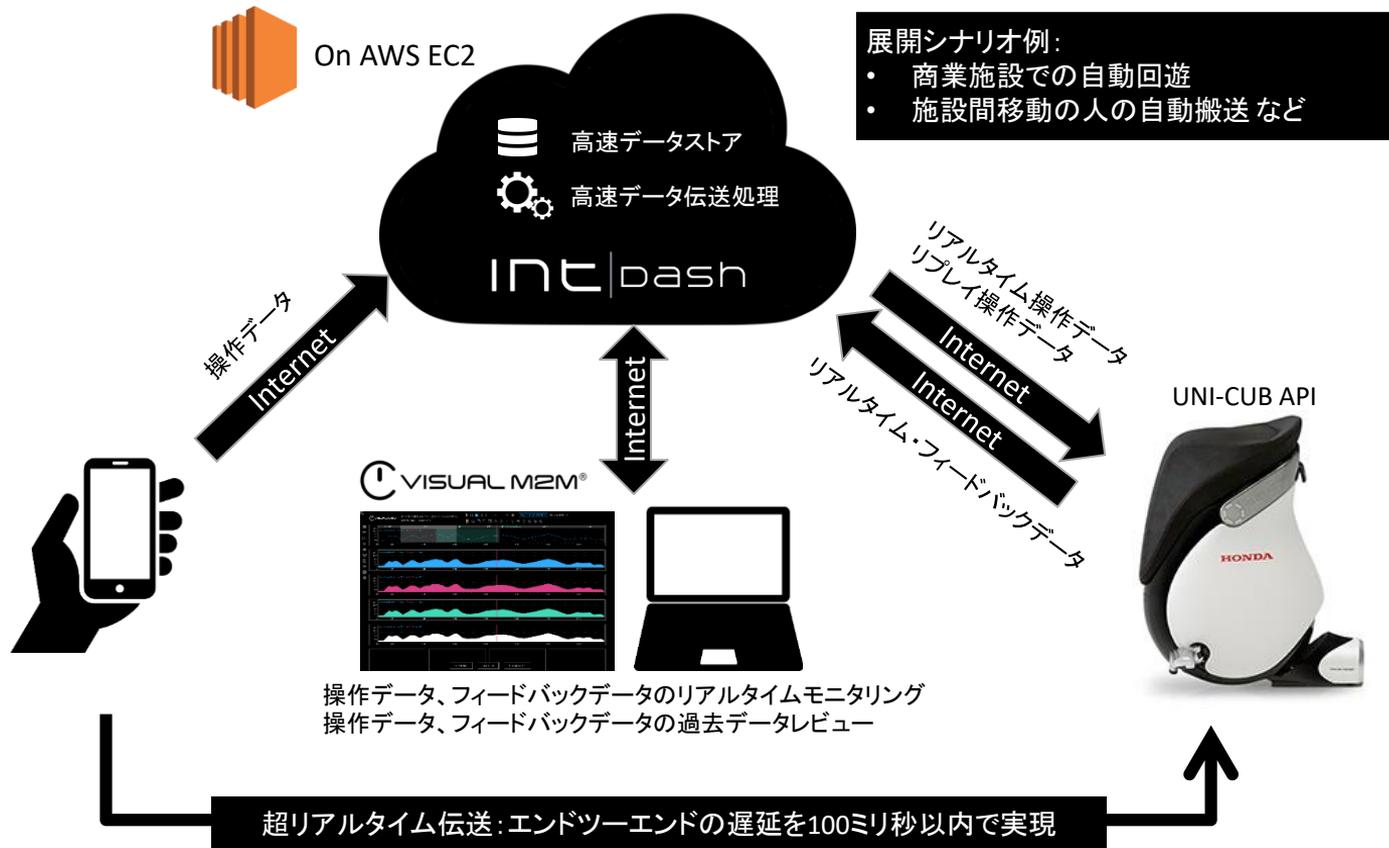
来週Interop 2017にて実機デモをご紹介します！！

是非ご来場ください！！



HONDA UNI-CUB β の遠隔操作デモ

-UNI-CUB APIとの連携-





全レイヤーでエンジニア絶賛募集中です！！

recruit@aptpod.co.jp

ご清聴ありがとうございました

 aptpod