気象情報

ホテル予約

災害情報

受発注

DX 時代のシステムアーキテクチャはどのようになるか？

現実世界には様々なデータやその変化(イベント)が存在し、
それらをリアルタイムに検知・分析・処理出来る必要がある

事件事故

バイタル

交通渋滞

健康状態

物流情報

ビジネスモデルの変化とイベントドリブン型アーキテクチャ

IoTの普及やビジネスモデルの変革に伴い、
イベントドリブン型アーキテクチャ
のユースケースが増えてくる

- **モノのサービス化**
  これまで購入していた様々なモノがサービス化されていく

  エアコンのサービス化
  エアコンの電源On/Offをイベントとして処理し、使った分だけ課金する
  (従量課金化、サブスクリプション化)

- **デジタルツイン**
  現実世界のセンサーデータの変化をデジタル化し、リアルタイムに様々な分析や予測に活用

  スマート街路灯(スマートシティ)
  事件事故の音声をイベントとして非常通報する

  出荷管理システム(スマート物流)
  受注をイベントとして受け取り、倉庫の出荷ロボットを制御する

重要視した IoT プラットフォームのポイント

データ利活用の促進のための

1. 標準的なデータモデル
2. 標準的なデータアクセスのためのオープンAPI
3. オープンソースソフトウェア(OSS)

FIWARE とは

- IoTアプリケーション開発のためのソフトウェアの集合体
- 次世代インターネット官民連携プログラム@EU で開発
- 次世代インターネット技術における EU の競争力強化と社会/公共分野のスマートアプリケーション開発を支援
- 欧州の自治体や、スタートアップで利用（110都市、25カ国で利用）
  - スペイン/サンタンデール市　スマートごみ収集
  - 日本/高松市　防災/河川水位測定
  - etc…

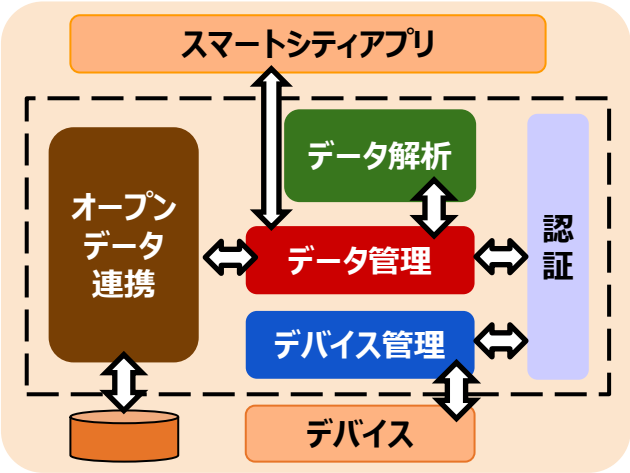
FIWARE
Open APIs for Open Minds

FIWARE の特徴

- オープンソースソフトウェア (OSS)
- 各コンポーネントは標準化された API (NGSI) に準拠
- 必要なコンポーネントを自由に組み合わせて利用可能

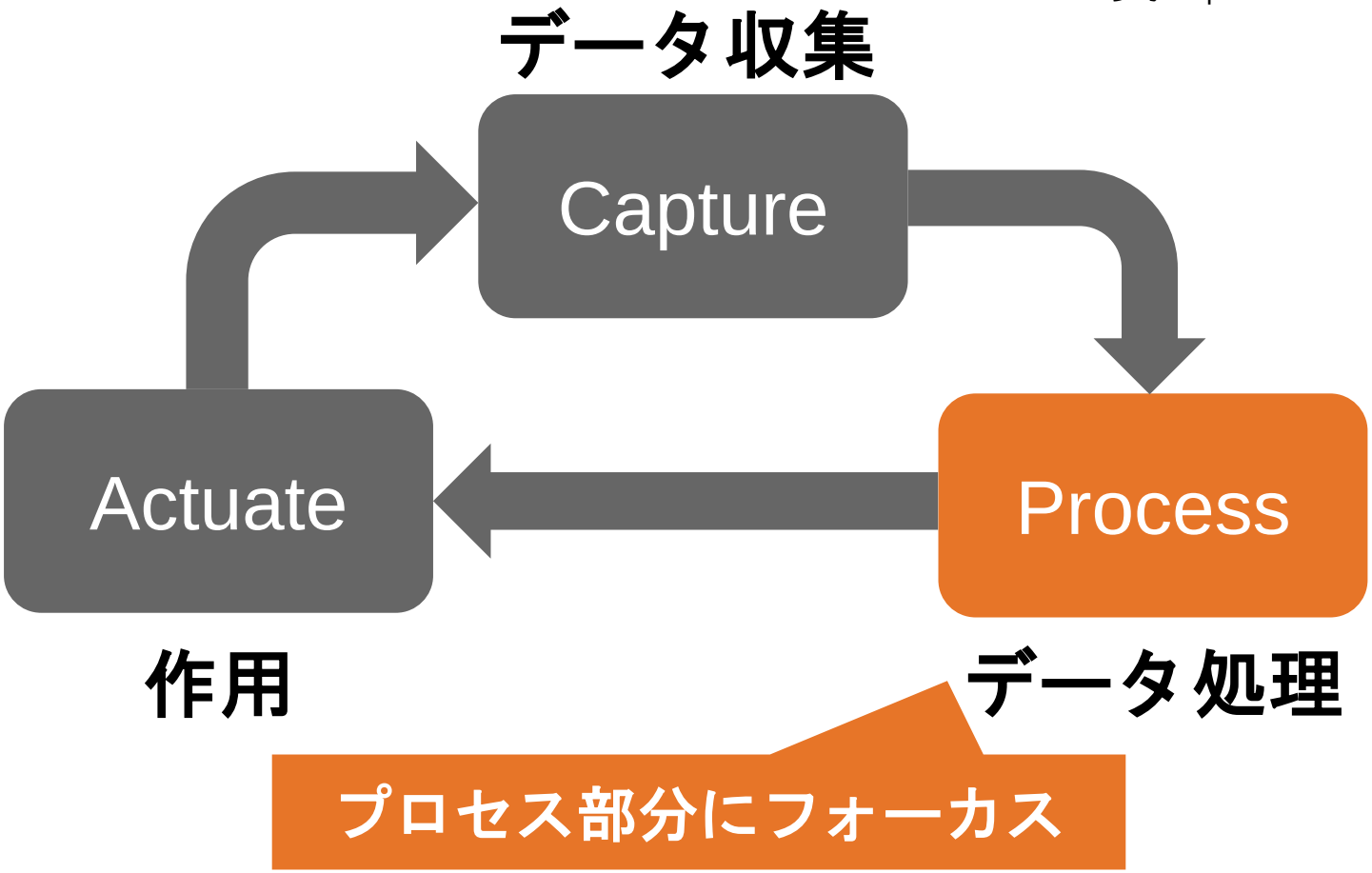
FIWAREコンポーネント群

用途にあわせて組み合わせる



FIWARE におけるデータ活用サイクル

参考: https://www.fiware.org/developers/



イベントドリブンを実現するためのFIWAREの課題と対策

課題

FIWAREには、イベントに応じて処理を実行する機能が不足

解決策

FIWARE に、FaaS を組み合わせることで不足の機能をカバーし、イベントドリブンの仕組みを確立する (コードネーム Meteoroid)

イベントの検出/通知

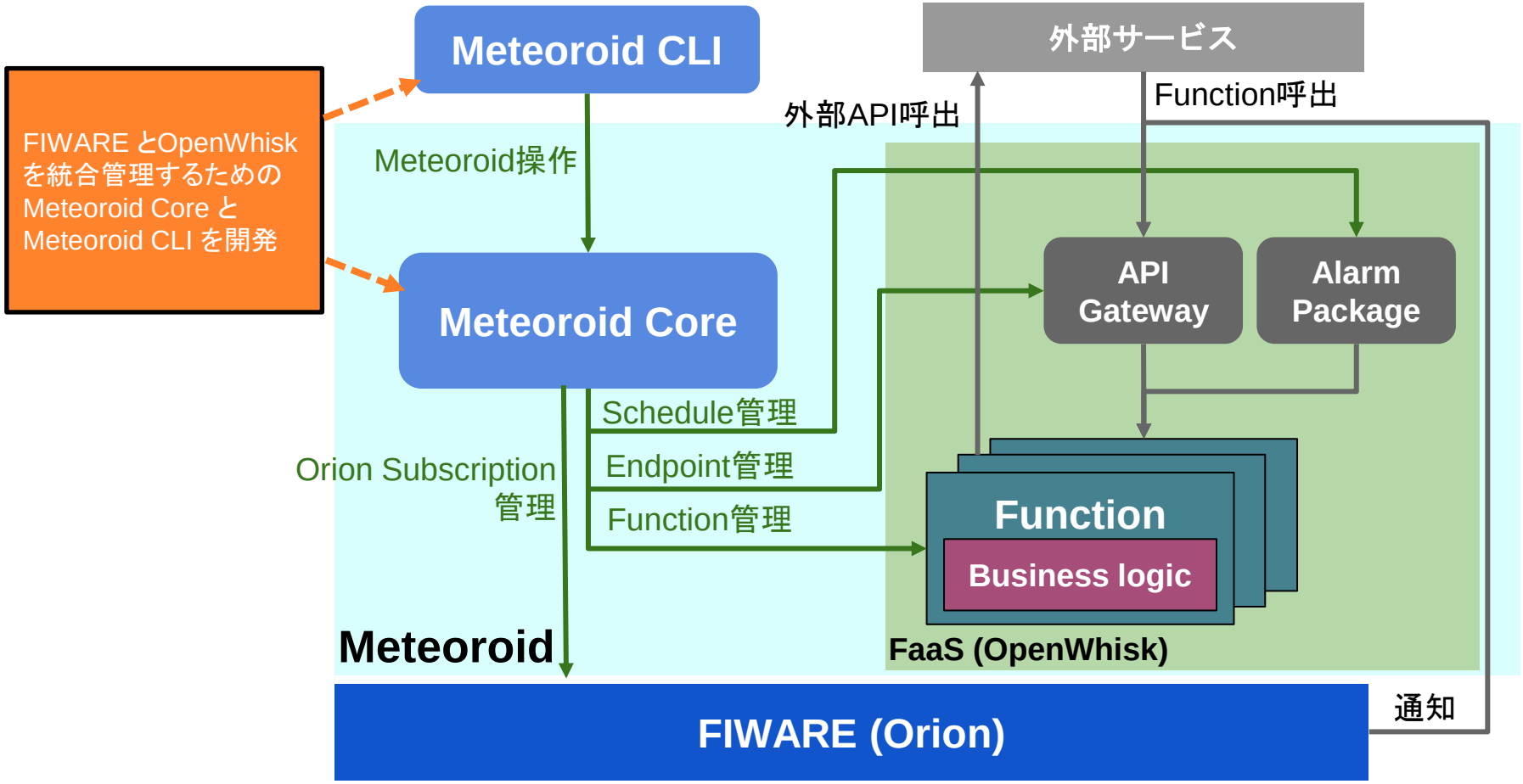
イベントドリブン処理の実行

FIWARE
Open APIs for Open Minds

×

APACHE
OpenWhisk

Meteoroid アーキテクチャ



本検証の成果

- **FIWARE Global Summit** @ベルリン(2019.10.24)
  - Integration Function as a Service capabilities in FIWARE

- **Okinawa Open Days 2019** @那覇(2019.12.12)
  - IoTプラットフォームにおけるFaaSの活用に向けた取り組み

- **OOL発のオープンソースとして「Meteoroid」を公開**
  https://github.com/OkinawaOpenLaboratory/fiware-meteoroid
  - Version 1.0 (2020.04.01)
    - OpenWhisk・Orion統合した基本機能
    - Schedule(定期実行処理)機能
    - Command Line Tool

GitHub

2020年度　活動内容

- **2020年度　活動内容**
  - **Meteoroid ユースケース検証**
  - タクシードライブレコーダー映像分析
  - kubernetes Ingress Controller 検証

Meteoroidとは何か

Meteoroidとは、FIWARE(IoTプラットフォーム)とApache OpenWhisk(FaaSプラットフォーム)を統合管理し、FIWAREエコシステム内におけるイベントドリブン処理を実現するOOL発のOSS

FIWAREとは…

欧州の官民連携プロジェクトで開発されたOSSのIoTプラットフォーム。40近くのモジュールで構成され、標準化されたAPI(NGSI)を持つ。

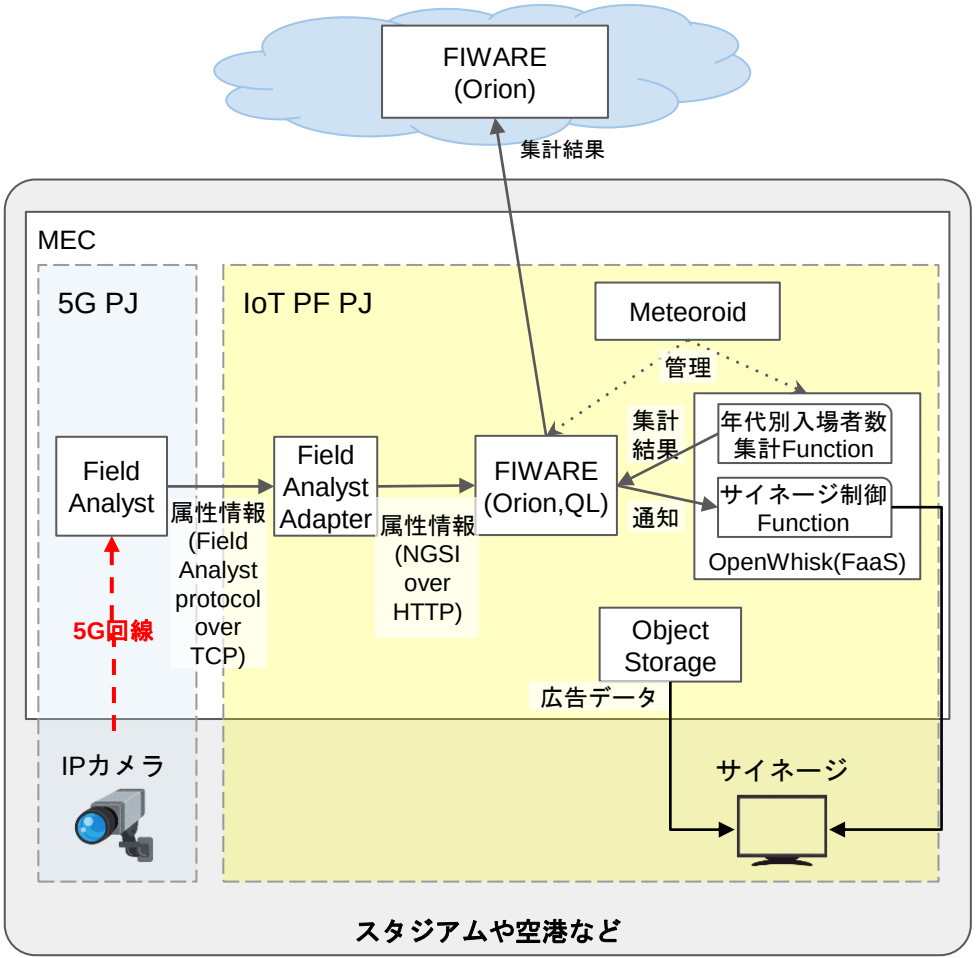
Apache OpenWhiskとは…

IBMが開発を主導しているOSSのFunction as a Service プラットフォーム。

Meteoroid ユースケース検証内容

ユースケース

- スタジアムや空港の入り口を想定
  (※実機はOOLの居室に設置)
- 入場者の属性(年齢)に応じた広告を
  サイネージ表示
- 個々人の属性情報を Field Analyst で収集し、
  MEC の FIWARE へ3秒間隔で通知・蓄積。
- 属性情報に従い Function でサイネージを
  制御。
  また、年代別入場者数を1日1回集計。
- 集計結果はクラウドの FIWARE に保存。
  広告データ(画像)は Object Storage から取得。



Meteoroid ユースケース検証改善活動

ユースケース検証からFIWARE・Meteoroidの課題を検出し、改善活動を実施。

課題

課題1 Function 開発の容易性

Function開発時にOrionやQunatumLeapのAPIリクエストを行うコードが複雑かつ冗長的。

課題2 Meteoroid CLI の使用性

Meteoroid CLI は Function の実行ログ取得など、開発中に必要となる機能が不足。

課題に対する改善活動

改善活動1 ライブラリ調査・開発

・Orionのクライアントライブラリの調査・動作確認
・Quantum Leapクライアントライブラリの開発
(OOLのGitHubで公開)

改善活動2 ドキュメント改善

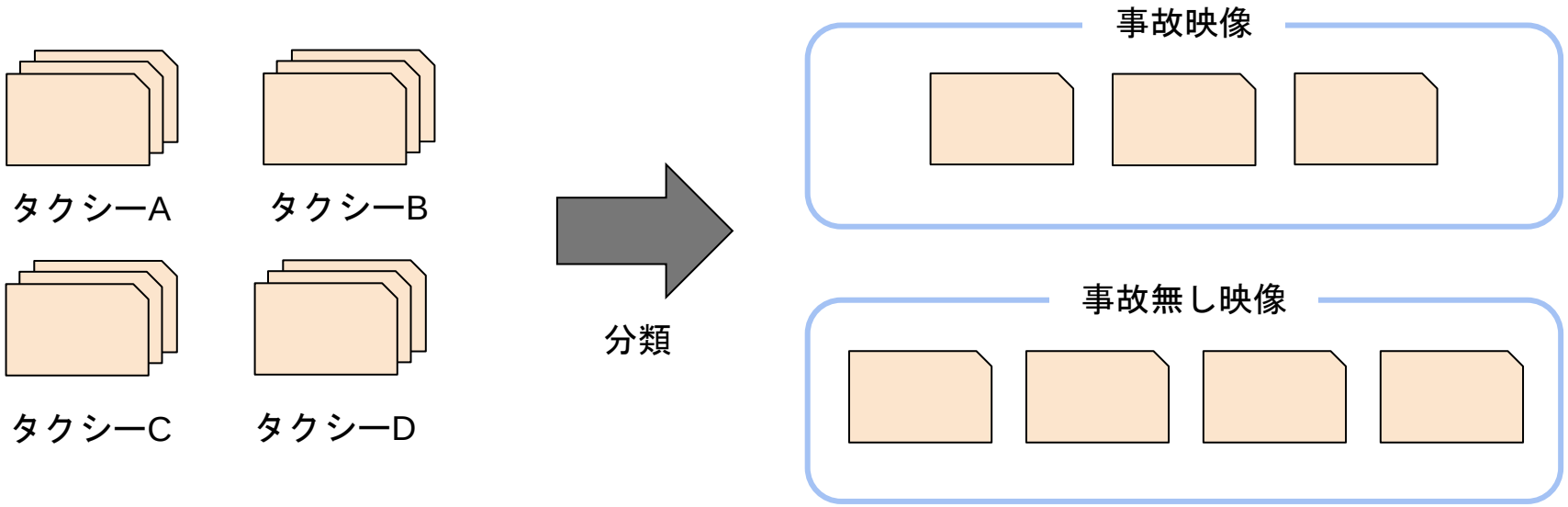
Functionの管理や開発に必要なコマンド(Meteoroid CLI や OpenWhisk CLI)を明示したドキュメントの作成、GitHubに公開。

2020年度　活動内容

- **2020年度　活動内容**
  - Meteoroid ユースケース検証
  - **タクシードライブレコーダー映像分析**
  - kubernetes Ingress Controller 検証

研究の背景と目的

- 交通事故が発生した際に警察へ事故映像を提供する必要があるが、交通事故が発生した部分の映像をタクシードライブレコーダーの中から抽出する作業を人手で行っており、手間がかかっている
- 機械学習などの仕組みを利用して、ドラレコデータから自動的に事故映像を抽出する



検証結果

- モデルの作成方法
  - Youtubeのデータセットを利用し、Resnetのファインチューニングを行う
  - 音声のモデルと画像のモデルを作成する

- 解析対象のドラレコデータ
  - 動画本数：321本
  - 総時間：22.5時間
  - 含まれている事故の件数：7件

検証結果

- Youtubeで作成したモデルにて音声解析を実施
  - ドラレコデータの音声データをメルスペクトログラムを用いて5秒で1枚の画像に変換
    - 16,465枚の画像データに変換
  - 事故ありと分類された画像は343枚
    - TP：6、FP：337
    - Precision：$\frac{TP}{TP+FP}$=0.0175=1.75%
    - Accuracy：$\frac{TP+TN}{TP+FP+FN+TN}$=0.9794=97.94%

		予測結果	
		事故	無事故
真の結果	事故	TP:6	FN:1
	無事故	FP:337	TN:16,121

検証結果

- Youtubeで作成したモデルにて画像解析を実施
  - ドラレコデータの映像データを1秒で1枚の画像に変換
    - 82,769枚の画像データに変換
  - 事故ありと分類された画像5,244枚（自車事故：3,166、他車事故：2,078）
    - T：0、F：5,244

		予測結果		
		自車事故	他車事故	無事故
真の結果	自車事故	T:0	F:0	F:3
	他車事故	F:0	T:0	F:1
	無事故	F:3,166	F:2,078	T:77,521

課題

- 音声解析にて、事故と検出したものの実際は無事故と誤検出したものが多い
  - 原因
    - ドアの開閉音やクラクション等大きな音を誤検出しているパターンが多い
  - 検討している対処法
    - Youtubeにて作成したモデルをドライブレコーダーでファインチューニングをかける
    - 事故/無事故の2分類から音の種別による複数分類に変更する
- 画像解析にて、事故映像を検出することができていない
  - 原因
    - モデルの教師データとしていた映像は、車同士の事故映像を多く使っていたが、ドライブレコーダーの映像の中には車同士の事故は含まれておらず、人との接触や物との接触であった
  - 検討している対処法
    - 人や物と接触する事故映像のデータセットが存在するか確認する

アジェンダ

- **2020年度振り返り**
  - Meteoroid ユースケース検証
  - タクシードライブレコーダー映像分析
  - **Kubernetes Ingress Controller 検証**

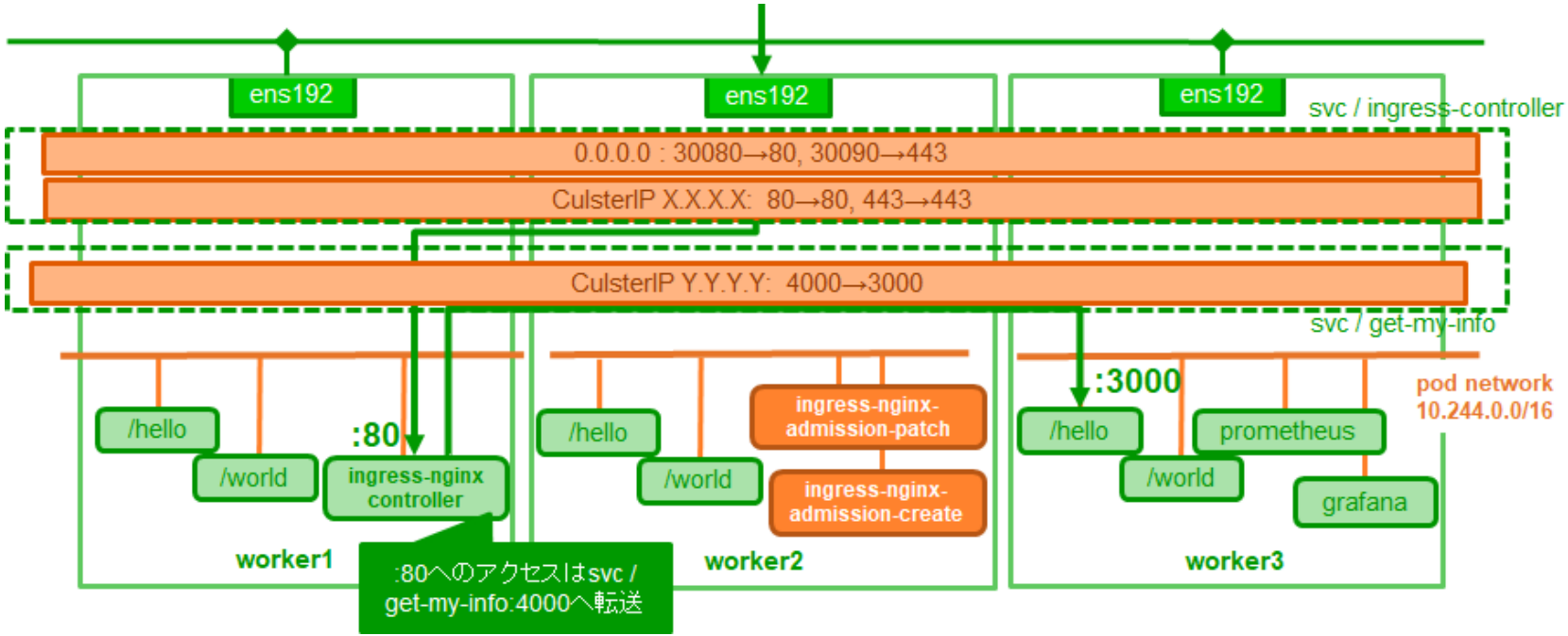
- 2021年度活動計画
  - X-Road 検証
  - FIWARE 可視化・分析ツール検証

2020年度　活動内容

- **2020年度　活動内容**
  - Meteoroid ユースケース検証
  - タクシードライブレコーダー映像分析
  - **kubernetes Ingress Controller 検証**

IngressController

- IngressControllerは、負荷分散やSSL終端等の機能を提供する



検証内容

- IngressControllerには様々な種類が存在するが、その中でも利用率の高い下記の５種を検証対象とする
  - Nginx
  - HAProxy
  - Istio
  - F5
  - GCP Cloud LB（机上調査のみ）
- ドキュメント調査による机上検証と実機検証を実施
  - 机上検証
    - 機能比較
      - 負荷分散方式等のLoadBalancerの主要機能の比較を行う
    - 非機能比較
      - 可用性、拡張性、運用・保守性の観点で比較を行う
  - 実機検証
    - ドキュメント通りに構築が可能か確認を行う
    - 机上検証を行った非機能面の動作確認を行う

検証結果

- 機能比較
  - LoadBalancerとしての基本的な機能に大きな違いは見られなかった
- 非機能比較
  - IngressControllerは、Kubernetes上にPodとして作成されるため、レプリカの作成等Kubernetesの機能を使うことができる
    - Kubernetesの機能を使う部分については各種において差分はない
- ドキュメント（構築・設定等）
  - Nginx、HAProxyはドキュメントの質が良く、個人ブログ等の記事も多いため構築や設定が容易だった
  - Istioはドキュメントの質が悪く、手順通りに構築や設定を行ってもエラーが出ることが多かった
  - F5はベンダー製品のため、公式以外のドキュメントは少なかった。公式ドキュメントは前のバージョンの記載が混在しており読み替える必要があった

2021年度活動方針

- **2021 年度 活動方針**
  - **スマートシティ基盤研究**
    スマートシティの基盤を支える都市 OS となり得る基盤の技術研究
  - **データ活用研究**
    都市 OS に蓄積されたオープンデータなどを活用するための技術研究、
    及び、ユースケース創出
- **技術キーワード**

  スマートシティ、FIWARE、X-ROAD、
  No Code / Low Code、BI (Business Intelligence) ツール

2021年度　活動内容

- **2021年度 活動内容**
  - **X-Road 調査**
  - FIWARE 間接続検証
  - データ連携検証
  - FIWARE 可視化分析ツール検証
  - BI ツール機能比較調査

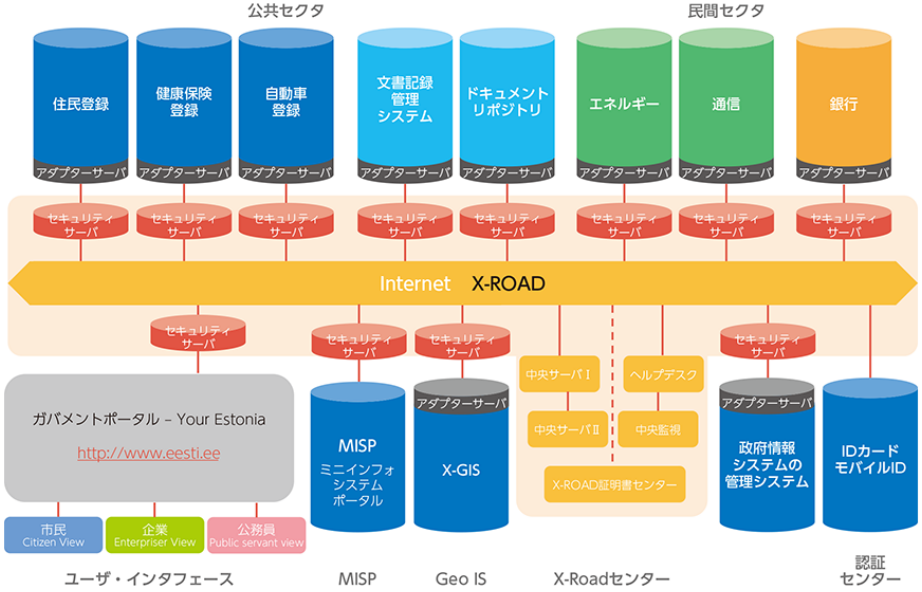
X-Road 調査

目的	スマートシティでは、多種多様なサービス・業種・都市のデータ流通を管理するためのプラットフォーム(都市OS)が必要とされている。都市OSとして注目されている X-Road の技術・実装について明確化する。また、X-Roadが複数の都市OS(FIWARE)をセキュアに接続する基盤となり得るか検証する。
ゴール	X-Road の構築、FIWARE との棲み分け ■ X-Road 概要調査 ■ X-Road アーキテクチャの明確化 ■ X-Road 機能評価 ■ X-Road と FIWARE の棲み分けや連携方式の検討

X-Road とは

■ **X-Road とは**

X-Road は、組織間で統一的で安全なデータ交換レイヤーを提供する電子政府基盤のオープンソースソフトウェア、及び、そのエコシステム。
分散されたデータベースやサービスをセキュアに連携させることができる。
電子政府先進国であるエストニアで 20 年近く開発・稼働しており、スマートシティの都市 OS として FIWARE とともに注目されている。

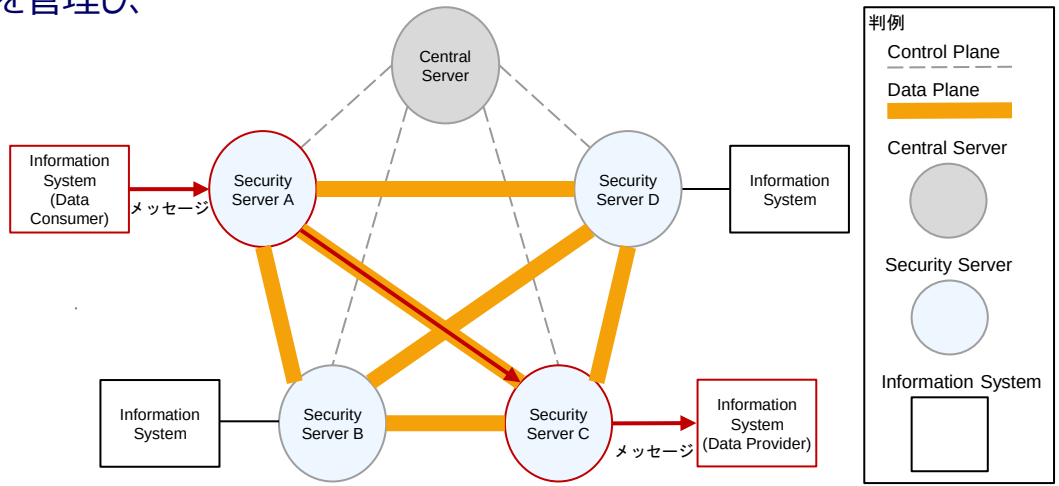


（出典）総務省 平成27年版　情報通信白書

X-Road 調査結果

- **X-Road のアーキテクチャを明確化**
  X-Road は サービスを提供・利用したい Client を管理するSecurity Server とそれら を集中管理する Central Server で構成される。

- **FIWARE との棲み分けを整理**
  - **X-Road**
    セキュアにデータを交換するシステムを管理し、
    データを保持しない
    - メッセージルーティング
    - TLS 暗号化
    - 時刻証明の付与
    - ログの監査
  - **FIWARE**
    データを標準的に管理し、
    データを保持する
    - データモデルの標準化
    - API の標準化



X-Road の Information System 間の通信

- **2021年度 活動内容**
  - X-Road 調査
  - **FIWARE 間接続検証**
  - データ連携検証
  - FIWARE 可視化分析ツール検証
  - BI ツール機能比較調査

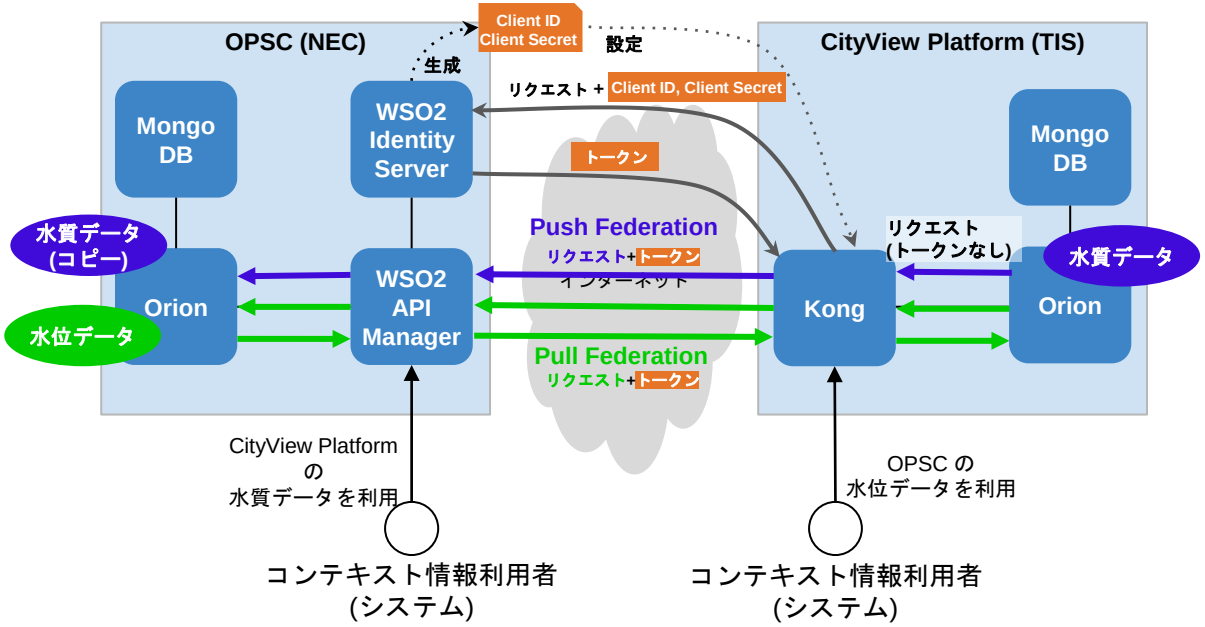
FIWARE 間接続検証 概要

課題 ・ 背景	異なる FIWARE ベンダーにおけるデータ連携には、認証方式の違いや API のバージョンの違いによって容易に連携できない課題がある。 これによって、NGSI でインターフェースやデータモデルを標準化している事による相互運用性の効果が得られなくなる。
ゴール	FIWARE 間におけるデータ連携手法を確立する。
内容	下記の観点で調査、検証する。 ● **Step 1: FIWARE 間を接続するアーキテクチャの調査** ○ FIWARE 間を接続する方法の調査 ○ API のバージョン差異の相互運用性の調査 ● **Step 2: 認証基盤を含む FIWARE 間接続のアーキテクチャの調査・検証** ○ 認証基盤や認証方式の調査・検証（テストベッドで検証） ○ NEC FIWARE と TIS FIWARE の基盤を利用した接続検証 ○ ベストプラクティスの整理

検証結果

【NEC】Open Platform for Smart City と 【TIS】CityView Platform の相互接続

Open Platform for Smart City と CityView Platform の FIWARE を Kongを使用しアクセストークンを付与したフェデレーションができることを確認



- **2021年度 活動内容**
  - X-Road 調査
  - FIWARE 間接続検証
  - **データ連携検証**
  - FIWARE 可視化分析ツール検証
  - BI ツール機能比較調査

データ連携検証　概要

課題・背景	Orion で管理するコンテキスト情報をオープンデータ化する場合、個人情報となる属性を秘匿化・匿名化する必要がある。 FIWARE では、テナントレベルのアクセス制御機能が存在するが、属性情報レベルでのアクセス制御ができない課題がある。
ゴール	FIWARE における属性情報の秘匿化・匿名化やアクセス制御のベストプラクティスを整理する。
内容	以下の手法の実現可能かを確認する。 **1. 属性情報を秘匿化・匿名化したコンテキスト情報を公開用のOrionへ転送** **2. 属性情報レベルのアクセス制御が可能な認証基盤・API Gatewayの活用**

検証結果

FIWARE の機能や Kong の標準プラグインでは属性情報の秘匿化・匿名化ができない。属性情報の秘匿化・匿名化を実現するためには Meteoroid の利用や Kong プラグインの開発・検証が必要

■ 各ステップの検証結果

- **Step1. 属性情報を秘匿化・匿名化したコンテキスト情報を公開用の Orion に転送**
  現状、FIWARE における個人情報の秘匿化・匿名化は、Meteoroid を利用したファンクションによる実装でしか実現できない。(ファンクションは下記 GitHub リポジトリで公開済み)
  https://github.com/OkinawaOpenLaboratory/fiware-meteoroid/tree/master/sample_functions/anonymous_context

- **Step2. 属性情報レベルのアクセス制御**
  Kong ACL Plugin では、仕様の制約により属性情報レベルのアクセス制御は実現できない。属性情報レベルのアクセス制御を実現するには Kong プラグインの開発・検証が必要。

■ FIWARE における秘匿化・匿名化を実現するための課題

- **秘匿化・匿名化を実現する手段が Meteoroid に限定される**
  Meteoroid を利用することでプラットフォームの構成や使い勝手が複雑になる。Meteoroid 以外の実現手段として、属性情報の秘匿化・匿名化の機能を持った Kong プラグインを開発する方法が考えられる。

- **2021年度 活動内容**
  - X-Road 調査
  - FIWARE 間接続検証
  - データ連携検証
  - **FIWARE 可視化分析ツール検証**
  - BI ツール機能比較調査

FIWARE 可視化分析ツール検証 活動内容

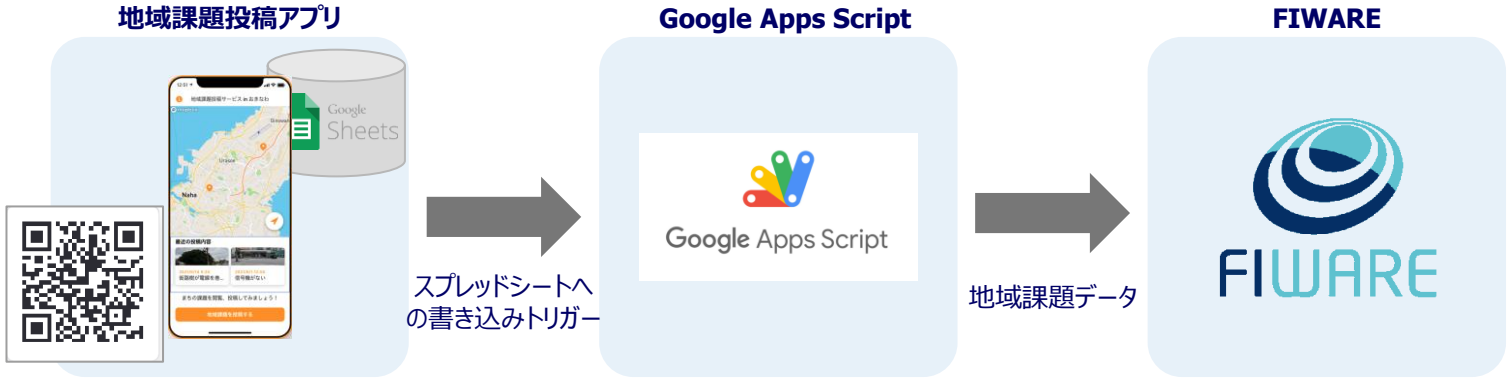
目的	FIWARE と Glide や Google Data Studio を連携させ、容易にデータを可視化・分析することを可能にするツールを調査検証する。
ゴール	FIWARE のデータを活用した Glide アプリ/ Google Data Studio ユースケースを創出する。
内容	下記の観点で調査検証する。 ■ Glide / Google Data Studio 連携方式の検討 ■ アプリ/ダッシュボード(ユースケース)開発

沖縄版 Civic Issues Tracking ユースケース

ノーコードツール Glide※ と FIWARE との連携

- **ユースケースイメージ**
  沖縄版 Civic Issues Tracking である「地域課題投稿アプリ」から投稿した地域課題データをオープンデータとして公開

- **実装イメージ**
  - アプリはノーコードツール Glide で開発
  - 投稿された地域課題データは Google Apps Script (GAS) のサーバーレス関数を使用して FIWARE へ登録



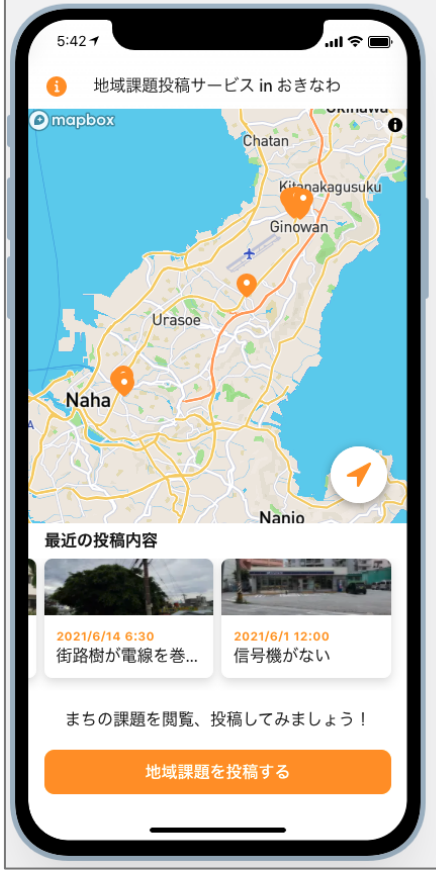
- 市民が地域課題を投稿
- Google スプレッドシートへの蓄積

- NGSI への変換
- Orion への登録

- オープンデータとして公開

※ Glide：Google のスプレッドシートを利用し、ノーコードでPWAアプリを作成することが出来るサービス

地域課題収集アプリ



沖縄 Issues Post

作者 Okinawa Open Laboratory

Made with Glide from a Google Sheet, without writing any code.

全画面表示

アプリのシェア

スマホやタブレットでこのコードをスキャンしてプレビューします。

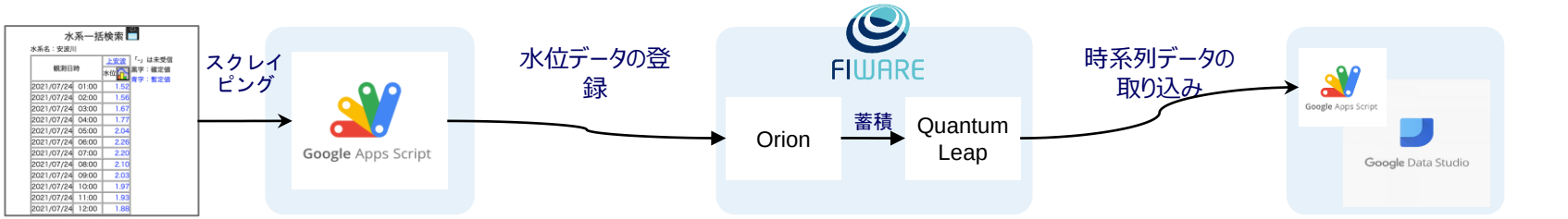
投稿画面

水位データ可視化ユースケース

Google Data Studio※ と FIWARE の連携

- **ユースケース**
  国土交通省が一時間毎に更新する河川の水位データを Google Data Studio で可視化する。
  水質データベース：http://www1.river.go.jp/

- **ユースケースの実装**
  1. 水位データを FIWARE へ登録する GAS の開発
  2. 水位データを時系列 DB (QuantumLeap)に蓄積させる Subscription の設定
  3. Google Data Studio に QuantumLeap のデータを取り込むコネクタを開発し、水位の変化を可視化



国交省の水位オープンデータ

1
水位データを Web スクレイピングで取得し Orion に登録

2
登録された水位データを Quantum Leap で蓄積

3
Quantum Leap Connector で時系列データを取得し Google Data Studio でグラフ化

※ Google Data Studio：データを分析、見える化するBIツールで、データの接続や加工、ビジュアライズ、共有することができる。

Google Data Studio で可視化した水位レポート

現在の水位一覧

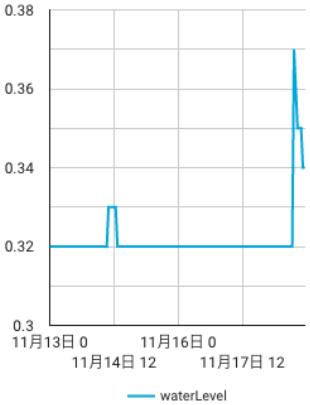
id	riverName	waterLevel
takaebashi	新川川	0.37
taihoohashi	大保川	1.05
shimohunn	普久川	0.86
kawada	福地川	0.2
kannabashi	漢那福地川	0.88
kamiaha	福地川	0.73
inatukibashi	羽地大川	0.29

1 - 10 / 10

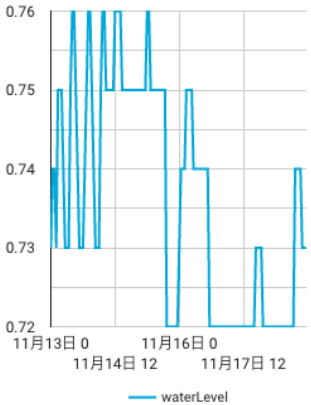
waterLevel
0.2 1.5

riverName
大保川 漢那福地川 普久川 福地川 億首川
新川川 辺野喜川 羽地大川

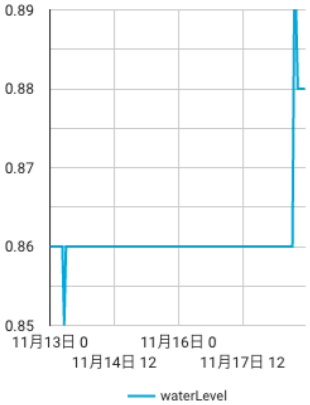
辺野喜川水位



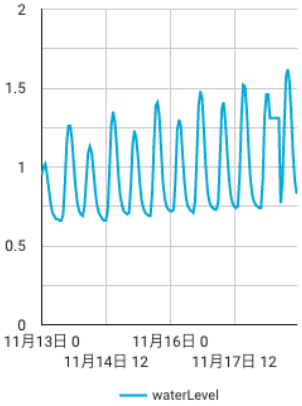
福地川水位



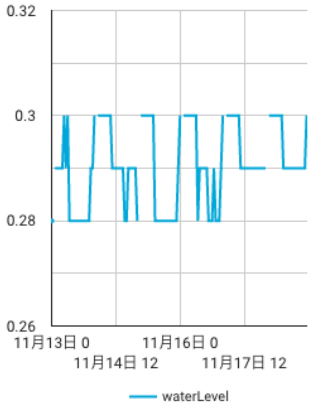
普久川水位



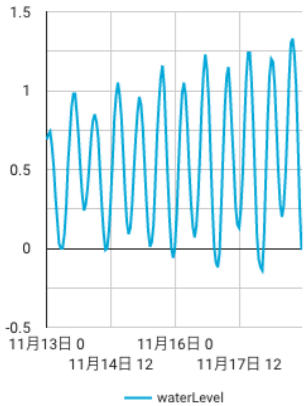
漢那福地川水位



羽地大川水位



億首川水位



- **2021年度 活動内容**
  - X-Road 調査
  - FIWARE 間接続検証
  - データ連携検証
  - FIWARE 可視化分析ツール検証
  - **BI ツール機能比較調査**

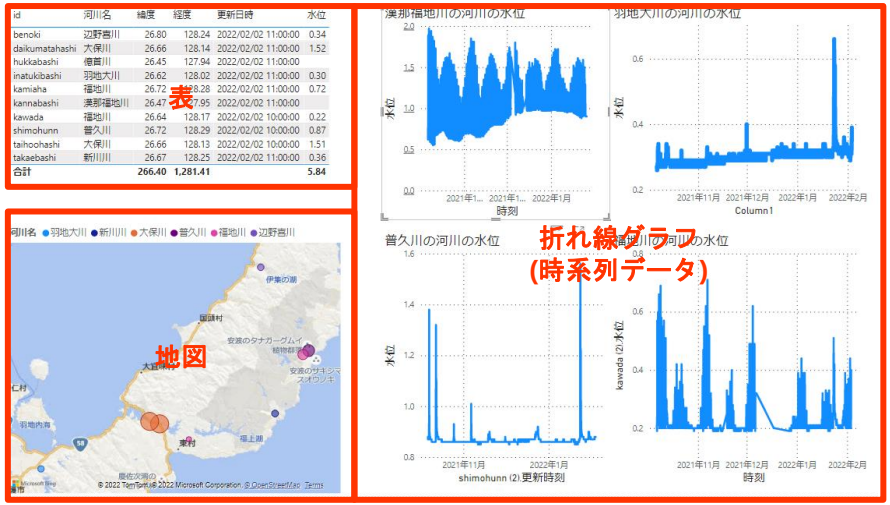
BI ツール機能比較調査　概要

目的	Google Data Studio で可視化した内容を Power BIで可視化し、機能比較を実施する。
ゴール	Google Data Studio と Power BI を比較し、各 BI ツールの使いやすさや使いどころをまとめる
内容	Google Data Studio と Power BI の機能比較を下記の内容で実施する。 ■ Power BI による河川水位データの可視化 ■ 比較観点の洗い出し ■ 比較観点に基づいた Google Data Studio と Power BI の調査・動作確認

BI ツールごとのダッシュボード

Power BI を使用し Google Data Studio で可視化した内容と同様のダッシュボード(河川の折れ線グラフ、表、水位マップ)の作成することが可能。

Microsoft Power BI Desktop



Google Data Studio

比較結果

Power BI、及び、Google Data Studio は大きな機能の差はない。データを分析し、可視化したい場合は Power BI を使用する。ダッシュボードを無償で共有したい場合はGoogle Data Studio を使用する。(Power BI Service は有償)

比較結果

分類	BIツールの主な機能	観点No	比較観点	Power BI		Google Data Studio(無償)
				Desktop(無償)	Service(有償・机上調査のみ)	
1.収集	データ入力・加工	1-1	データソースの種類	一部の Google 系サービスを除き、主要なデータソースに接続可能	Desktop と組み合わせることで主要なデータソースに接続可能	Google 系サービスに接続できるコネクタは多いが、Excel や Azureといった主要なデータソースは接続不可
		1-2	データ量の上限	無制限に収集可能	Pro: 10GB Premium: 100TB	不明(キャッシュサイズに依存)
		1-3	データの更新頻度	一部データソースは自動更新可能 (ExcelやWebは自動更新不可)	自動更新が可能	
		1-3	データの加工	データの変換や結合が可能		データの変換は出来るが結合する機能が不足
		1-4	対応するデータ型	主要なデータ型(日付や緯度経度など)に対応済		主要なデータ型(日付や緯度経度など)に対応済
2.分析	OLAP (オンライン分析処理)	2-1	複数データソースから収集したデータの分析	ダッシュボード上で分析に必要な主な操作が可能		可視化するデータが階層構造になっていないため、分析に必要な操作が一部不可
	データマイニング	2-2	クラスタリング、AIなど	標準である分析ツール、python、Rスクリプトを用いて分析が可能		機能無し
	シミュレーション	2-3	What-if分析	分析ツールの感度分析を用いることでシミュレーションが可能		データ操作を行えるパラメーターをでシミュレーション可能
3.可視化・共有	レポーティング (ダッシュボード)	3-1	グラフの種類	主要なグラフの作成が可能。また、独自のグラフを作成することが可能	主要なグラフの作成が可能。Desktop で作成したグラフを表示可能	主要なグラフの作成が可能。独自のグラフの作成はできない。
		3-2	テンプレート機能	PBIT 拡張子でテンプレートの公開、使用が可能	Microsoft Apps Source でテンプレートの公開、使用が可能	Report Gallery でテンプレートの公開、使用が可能
		3-3	分析結果の共有	PDF で可能	PDF、URL、htmlコード、SharePoint で可能	PDF、URL、htmlコード、メールで可能
		3-4	共有範囲の制限	個人利用用のため権限付与する機能は有していない	個人単位で閲覧・編集権限付与が可能	

機能有

機能が一部不足

機能無し