



キーテクノロジー 「ThingVisor Factory」の紹介

早稲田大学 理工学術院総合研究所
金井 謙治

k.kanai@aoni.waseda.jp

スマートシティの実現における課題

- スマートシティの実現における課題
 - 相互運用性, データ流通, 拡張容易性
- VirIoTのユーザビリティ向上に向けて
- ビジュアルプログラミング
 - ビジュアル言語 vs テキスト言語
- ThingVisor Factory: IoTサービスを生み出す“工場”
 - ビジュアルプログラミングに基づくIoTサービス構築
- まとめと今後の展望

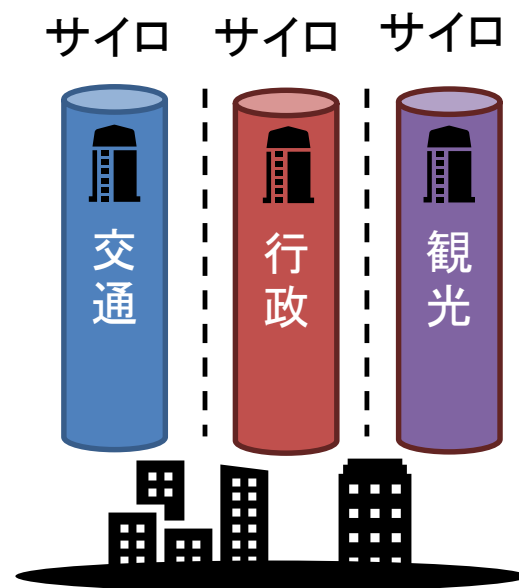


Google Cloud



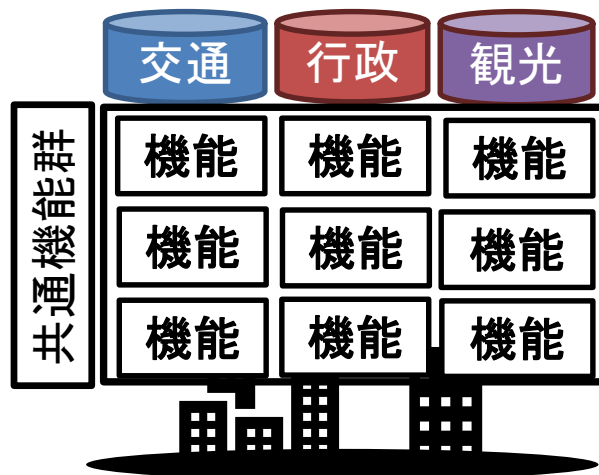
スマートシティの実現における課題

- サービスの再利用・横展開(**つながらない**)
 - 分野、都市間でIoTシステムが個別に実装(**システムのサイロ化**)
 - 構築したIoTシステムやサービスの再利用が困難
- 分野間データ利活用(**ながれない**)
 - 分野、都市毎に様々なデータが独立
 - 分野間、都市間を横断したデータ連携が困難
- 拡張性の低さ(**つづかない**)
 - 構築されたIoTシステムの拡張性が低い
 - 継続的かつ容易なサービスの更新が困難



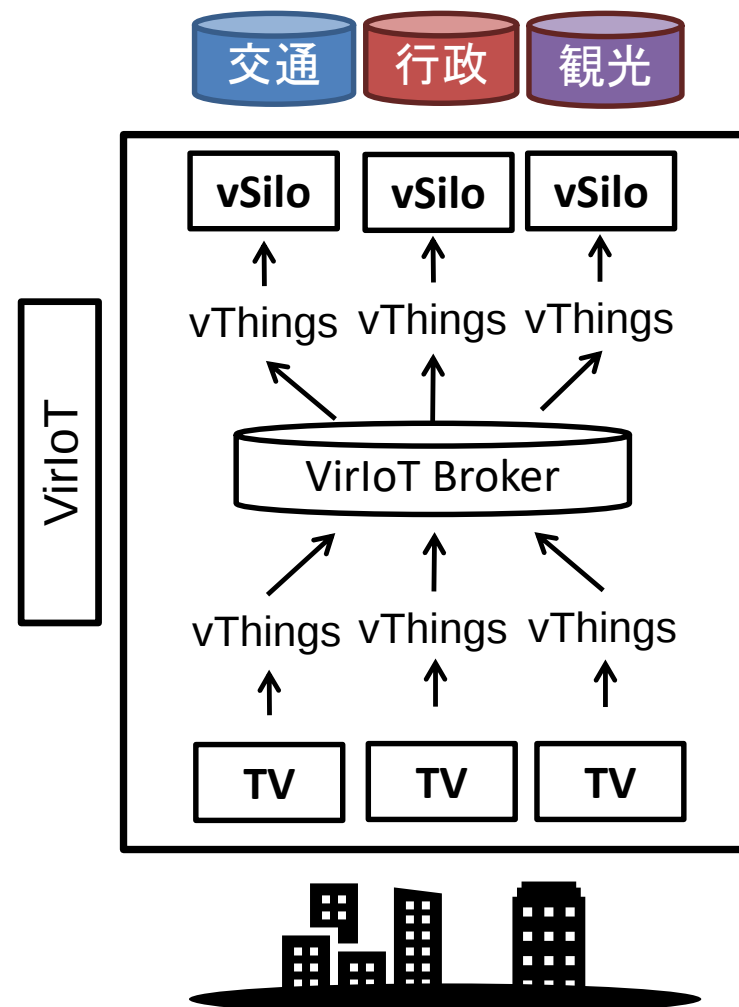
スマートシティの実現における課題

- 都市間・分野間をつなぐ(**相互運用性**)
 - 都市間・分野間でサービスの横展開を可能にする仕組み
 - 共通的な機能や標準インターフェースを定義
- 都市間・分野間にながす(**データ流通**)
 - 様々なデータを一つの論理的なデータに見せかける仕組み
 - 都市間・分野間でデータを仲介する仕組み
- 容易につづける(**持続性・拡張容易性**)
 - IoTシステムを機能ブロックに分けて構築
 - 機能単位に拡張・拡張できる仕組み



Fed4IoTが提供するVirloT

- Virtual IoT System (VirloT)
 - 機能の共通化
 - ThingVisor, VirloT Broker, vSilo
 - データモデルの共通化
 - NGSI-LD
- つながる・ながれる・つづく仕組み
 - ThingVisor
 - サービスを提供するためのデータ処理を担う
 - データ処理した結果をvThingとして流通
 - VirloT Broker
 - 都市間・分野間でのデータ仲介を担う
 - vSilo
 - サービスに対するデータ流通を担う
 - NGSI-LD
 - VirloT内の共通言語

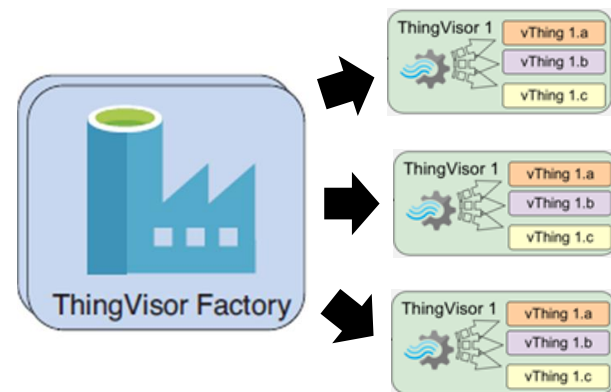


VirloTのユーザビリティ向上に向けて

- “VirloTの肝はThingVisor”

- ThingVisor

- サービスを提供するためのデータ処理を担う
 - データ処理した結果をvThingとして流通



- VirloTを通して、簡易にサービス展開するためには？

- どのように利用可能なThingVisorを知るのか？
 - どのようにThingVisorを作成すれば良いのか？
 - どのようにThingVisorを利用すれば良いのか？

- ThingVisor Factory: ThingVisorを生み出す“工場”

- VirloTに対して、機能拡張・利便性を提供
 - サービス提供者・市民に対してユーザインターフェースを提供

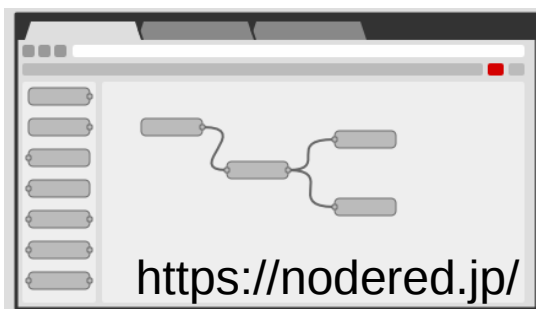
ThingVisor Factory

- 簡易にIoTサービスを構築するプラットフォーム
 - ビジュアルプログラミングに基づくIoTサービス構築
- 利便性
 - Webブラウザを介して、ドラッグアンドドロップでサービス開発
- 相互運用性
 - サービスの機能をユーザ間で共有
- データ流通
 - 機能間はネットワークを介して接続
- 拡張容易性
 - 各機能はコンテナ型仮想化ソフトウェア(Docker)で実装

ビジュアルプログラミング

- ビジュアル言語

- 図や絵といった視覚的オブジェクトで記述するプログラミング
 - ドラッグアンドドロップで操作
- ブロック型: 指示が記載されたブロックを積み重ねていくイメージ
 - Scratch, MakeCode, Blockly,...
- データフロー型: 処理ごとに線で結合させていくイメージ
 - **Node-RED**, MATLAB Simulink, MESH, SAM Labs, TensorFlow,...
- 独自ルール



```
1 #include <stdio.h>
2
3 int main(void){
4
5     printf("Hello world! \n");
6
7     return 0;
8
9 }
10
```

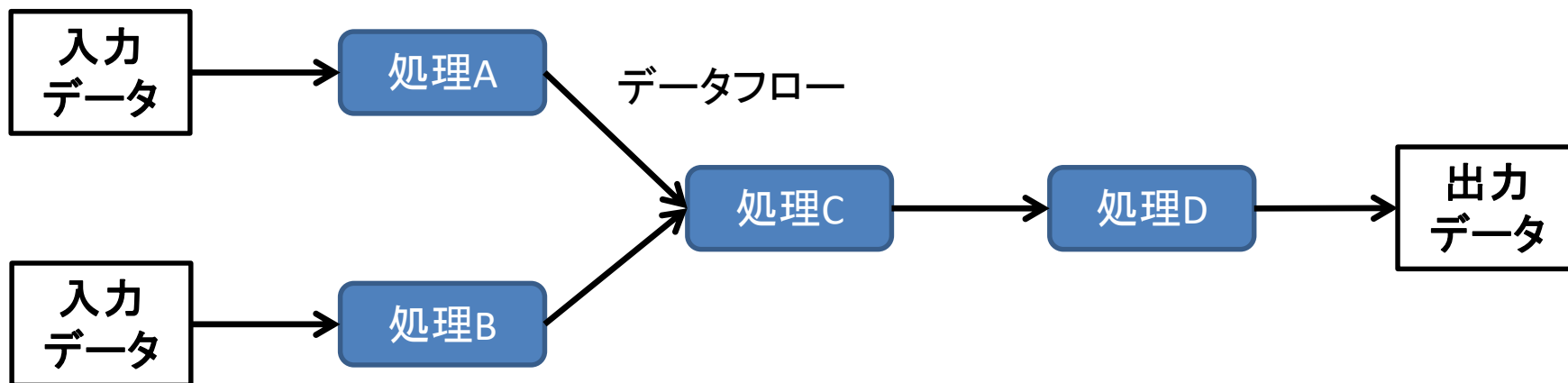
“Helloworld” in C

- テキスト言語

- 英数字などの文字で記述するプログラミング
 - C言語, C++, Java, Ruby, JavaScript, Python, Go,...

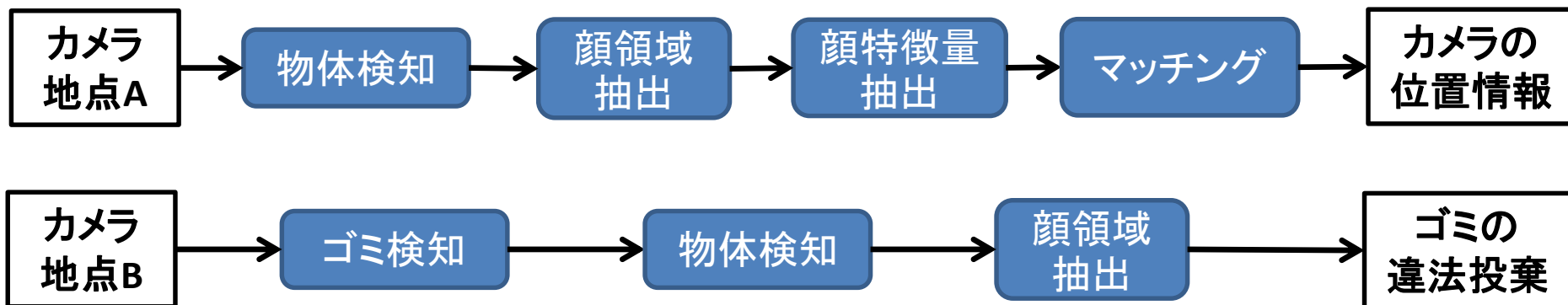
ThingVisor Factoryが扱うIoTサービス

- 前提条件
 - IoTサービスは機能単位に分割可能
 - 処理データは左から右にフローとして流れる(方向を持つ)
- データフロープログラミングに基づくIoTサービスの表現
 - IoTデータフローに従い有向グラフによって表現可能
 - (深層学習についてもこの概念に従い記述可能)



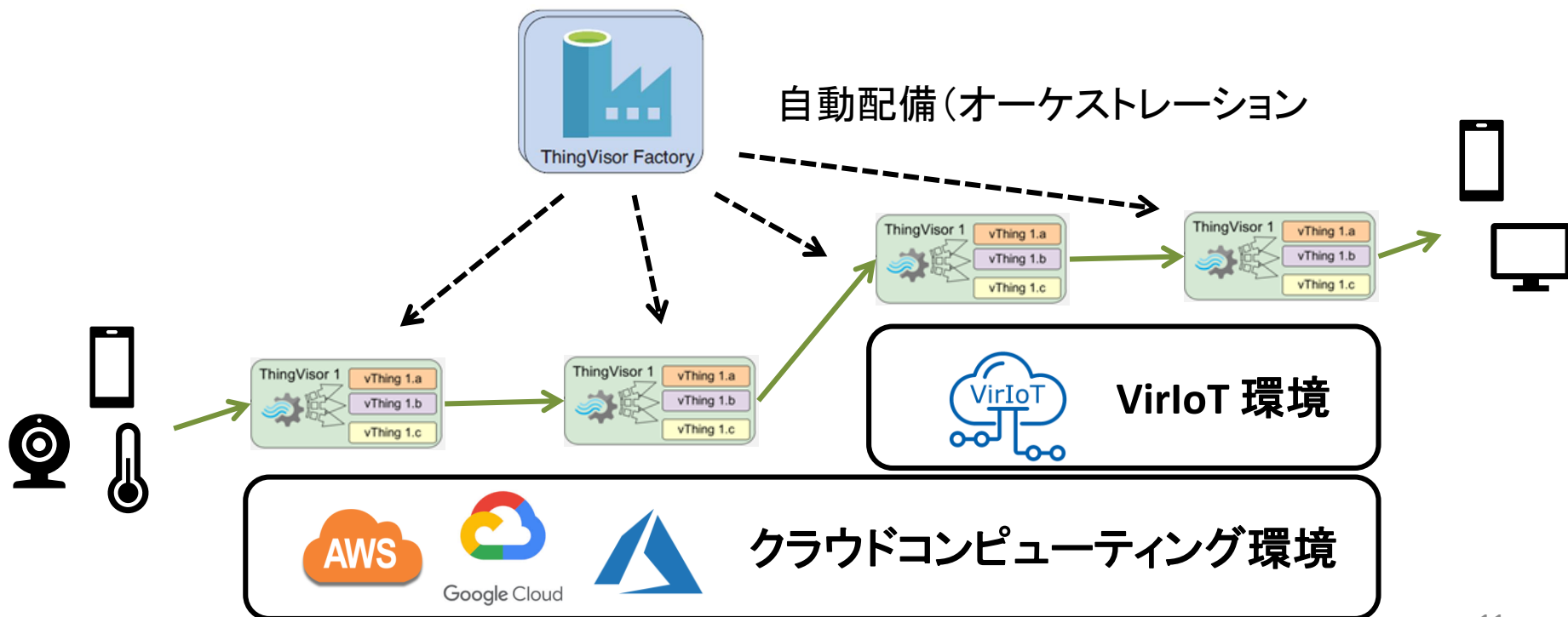
ThingVisor Factoryが扱うIoTサービス

- 前提条件
 - IoTサービスは機能単位に分割可能
 - 処理データは左から右にフローとして流れる(方向を持つ)
- データフロープログラミングに基づくIoTサービスの表現
 - IoTデータフローに従い有向グラフによって表現可能
 - (深層学習についてもこの概念に従い記述可能)
- 具体例



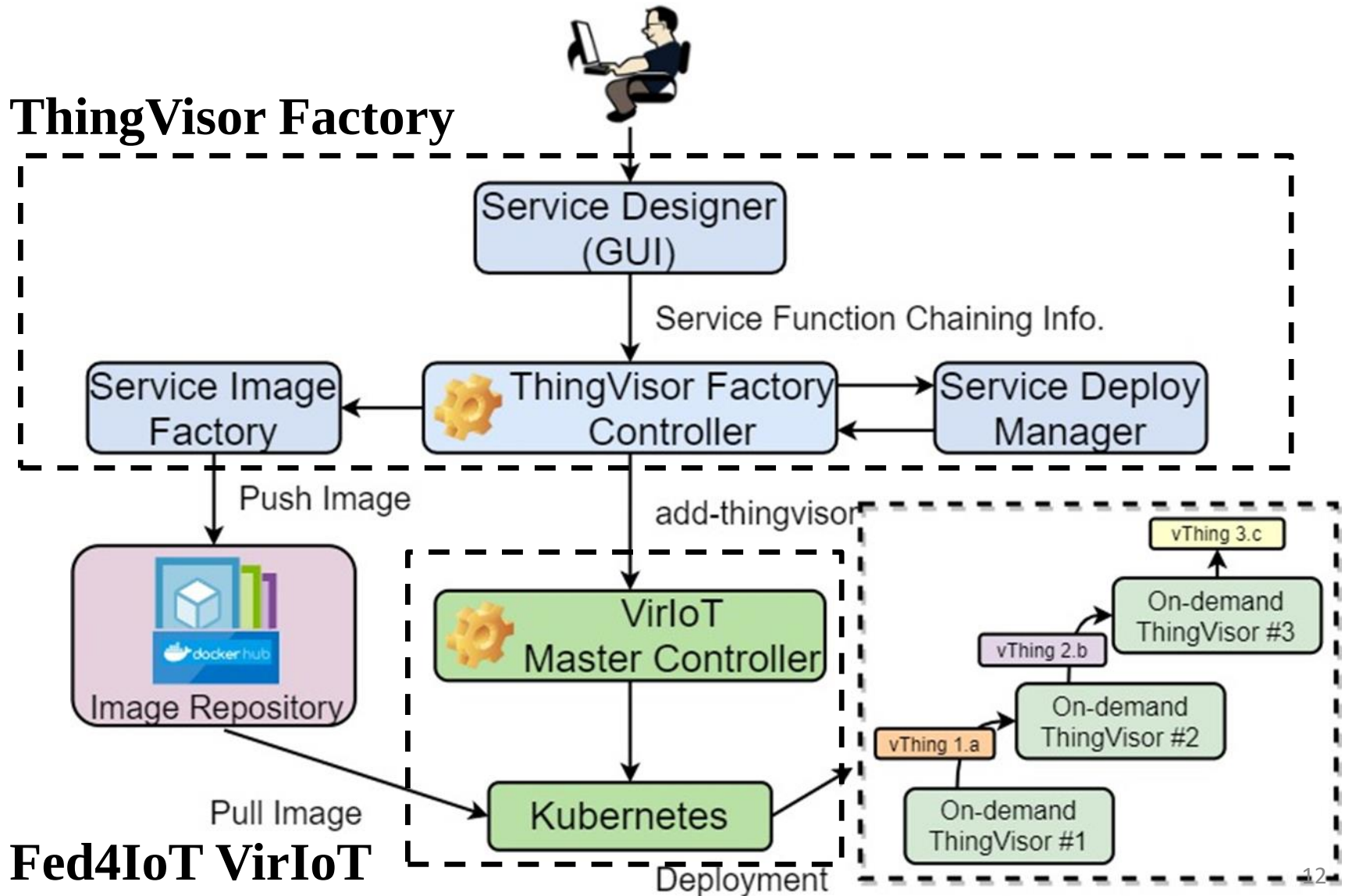
ThingVisor Factory

- ThingVisor Factoryの機能
 - IoTサービスの機能群をThingVisorとして定義
 - データフロープログラミングに基づき固有のサービスを構築
 - VirIoTを含むクラウド環境にThingVisorを自動配備
 - ThingVisor間のデータ流通も自動解決



ThingVisor Factoryのアーキテクチャ

ThingVisor Factory

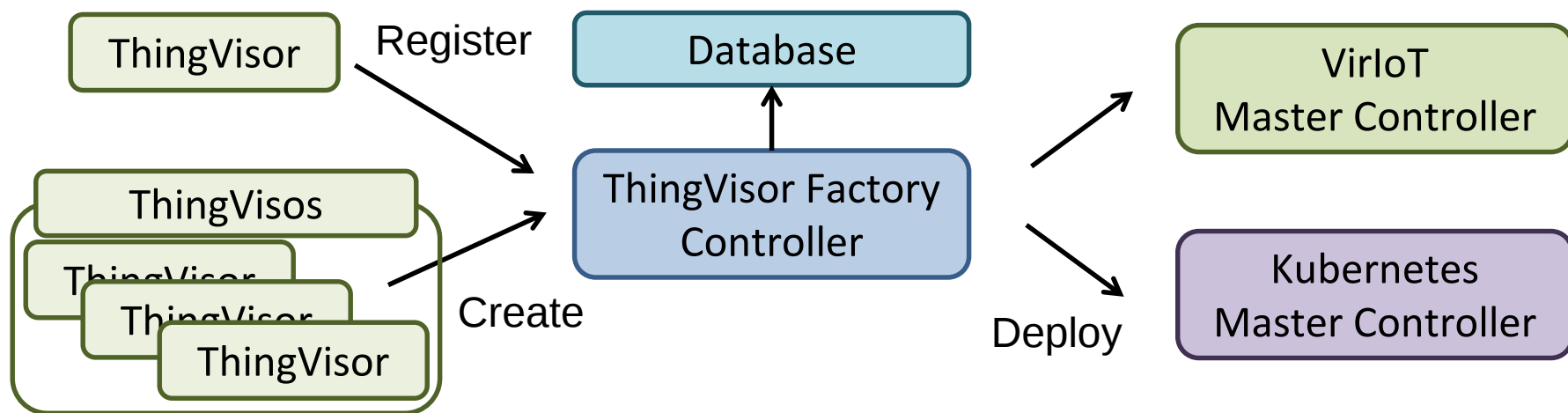


ThingVisor Factoryオーケストレーション

- ThingVisorの配備やスケーリングを管理
 - 動的な最適配備をサポート
 - 通信資源、計算資源のみならず地理的位置を考慮
 - エッジコンピューティングをサポート
 - 最適配備やタスクスケジューラはモジュールとして実装
 - 用途に併せて最適なアルゴリズムを選択
- ThingVisor間のルーティングを自動解決

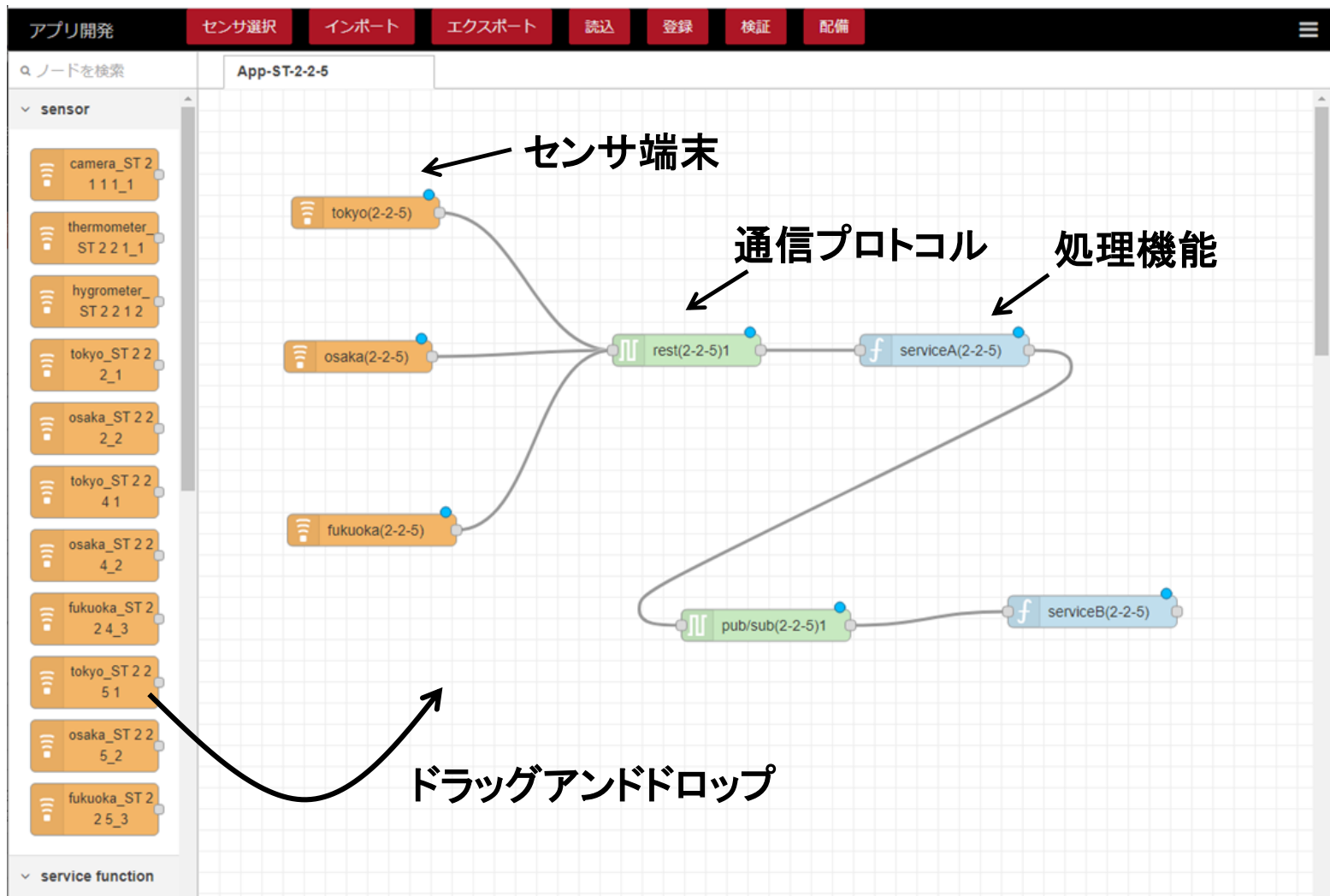
ThingVisor Factoryコントローラ

- 各機能呼び出すAPIを定義
 - ThingVisorのデプロイのためのCRUD
 - Create, Read, Update, Delete
 - ThingVisorのメタ情報の管理
 - ThingVisorのスケジューラー
 - KubernetesやVirloTとのインターフェース



ThingVisor Factoryの操作画面

- Service Designer: Graphical User Interface (GUI)



ThingVisor Factoryの機能紹介

- 利用可能な機能ブロック
 - **Sensor**
 - 利用可能なセンサ端末の選択、登録、削除が可能
 - **Service Function**
 - IoTサービスの処理機能
 - パッケージ化されている処理機能を選択可能
 - Dockerによるコンテナ型仮想化ソフトウェアによるパッケージング
 - **Connector**
 - 各機能ブロックを“ネットワークを介して”連結させる機能
 - REST (HTTP), Pub/Sub型通信が利用可能
 - **Program**
 - ブラウザ上でPythonプログラミングが可能

ThingVisor Factoryの機能紹介

- センサの登録と削除

tokyo

osaka

fukuoka

センサ選択

選択	センサ名	センサ種類	アドレス	アドレスタイプ	緯度	経度	設置場所
☐	センサ_ST-1-1-1_1	センサ	{"URL":"/xx/vv/zzz"}	その他	11.11	22.22	ビル1 4 F
☐	センサ_ST-1-1-1_2	センサ	{"URL":"/xx/vv/zzz"}	その他	11.11	22.22	ビル1 4 F
☐	センサ_ST-1-1-1_3	センサ	{"URL":"/xx/vv/zzz"}	その他	11.11	22.22	ビル1 4 F
☐	センサ_ST-1-1-1_4	センサ	{"URL":"/xx/vv/zzz"}	その他	11.11	22.22	ビル1 4 F
☐	カメラ_ST-1-1-2_1	カメラ	{"IP":"0.0.0.0"}	IP	33.33	44.44	駅北口
☐	センサ_ST-1-4-1	センサ	{"URL":"/xx/vv/zzz"}	その他	11.11	22.22	ビル1 4 F
☐	Svセンサ_ST-1-6-1_1	センサ	{"URL":"/xx/vv/zzz"}	その他	11.11	22.22	ビル1 4 F
☐	CIセンサ_ST-1-6-1_2	センサ	{"URL":"/xx/vv/zzz"}	その他	11.11	22.22	ビル1 4 F
☐	CIセンサ_ST-1-6-1_3	カメラ	{"URL":"/xx/vv/zzz"}	その他	11.11	22.22	ビル1 4 F
☒	camera_ST-2-1-1-1_1	camera	{"address":"127.0.0.1"}	IP	35.309221389	139.047827778	waseda university
☒	thermometer_ST-2-1-1_1	Temperature sensor	{"name":"temp"}	NDN	35.309221389	139.047827778	waseda university
☒	hygrometer_ST-2-2-1-2	Humidity sensor	{"name":"humi"}	NDN	36.30922138	140.047827778	waseda university(研究室A)
☒	tokyo_ST-2-2-2_1	Vibration sensor	{"name":"vib1"}	NDN	35.309221389	139.047827778	tokyo
☒	osaka_ST-2-2-2_2	Vibration sensor	{"name":"vib2"}	その他	36.30922138	140.047827778	osaka
☒	tokyo_ST-2-2-4-1	Vibration sensor	{"name":"vib1"}	NDN	35.309221389	139.047827778	tokyo
☒	osaka_ST-2-2-4_2	Vibration sensor	{"name":"vib2"}	NDN	36.30922138	140.047827778	osaka
☒	fukuoka_ST-2-2-4_3	Vibration sensor	{"name":"vib3"}	NDN	37.30922138	141.047827778	fukuoka
☒	tokyo_ST-2-2-5-1	Vibration sensor	{"name":"vib1"}	NDN	35.309221389	139.047827778	tokyo
☒	osaka_ST-2-2-5_2	Vibration sensor	{"name":"vib2"}	NDN	36.30922138	140.047827778	osaka
☒	fukuoka_ST-2-2-5_3	Vibration sensor	{"name":"vib3"}	NDN	37.30922138	141.047827778	fukuoka

アプリ開発ページ

登録

削除

ThingVisor Factoryの機能紹介

- センサ登録画面

センサが提供する
プロトコルを指定

センサ登録

センサ名

ネットワークカメラ001

センサ種類

カメラ

アドレスタイプ

IP

アドレス

{ "address": "127.0.0.1" }

緯度

35.309221389

経度

139.047827778

設置場所

早稲田大学

アクセス権限レベル

1

出力プロトコル

REST(Server)

出力プロトコル - アクセス先

メソッド

POST

プロトコル

HTTP

パス

/waseda/camera/001

Dockerリポジトリ名

tvf/cam001

メタ情報

{ "info1": "sensor", "info2": 123 }

説明

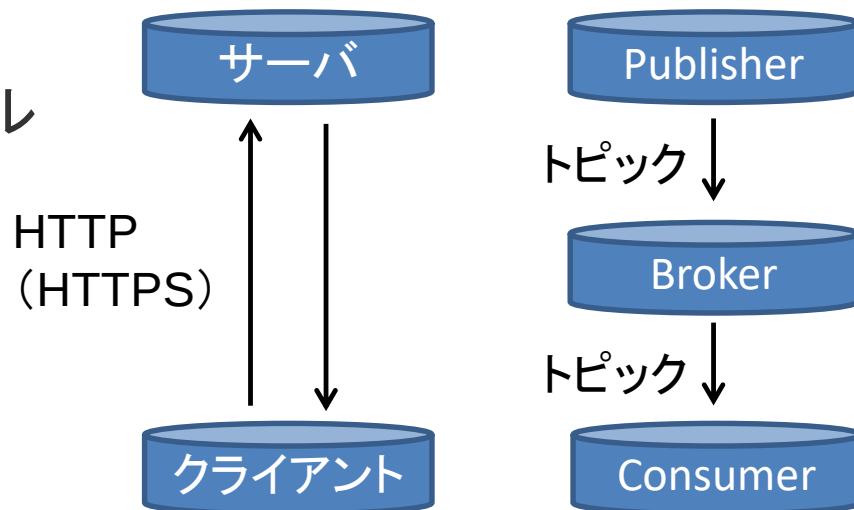
テスト用ネットワークカメラ

登録

ThingVisor Factoryの機能紹介

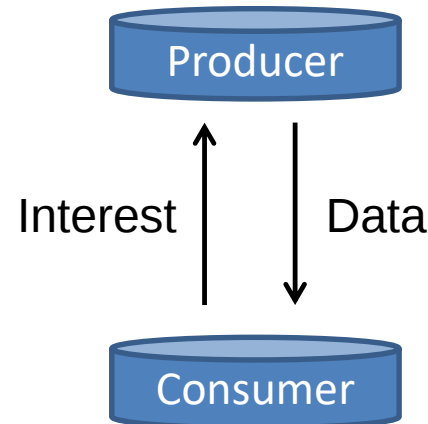
- 現状サポートしているプロトコル

- REST (HTTP)
 - POST, GET
- Pub/Sub (トピック型)
 - MQTT, Apache Kafka



- サポート予定のプロトコル

- 情報指向ネットワーク (ICN) : 次世代ネットワークプロトコル
 - InterestとDataのペアでデータ取得
 - IPアドレスではなく、“コンテンツ名”で経路制御
- 実装中のオープンソースソフトウェア
 - NDN
 - Cefore/Cefpyco (NCIT開発)



ThingVisor Factoryの機能紹介

- ProgramノードによるPythonプログラミング

The screenshot displays the ThingVisor Factory web interface. At the top, there is a navigation bar with buttons: 一覧 (List), 読み込み (Load), 登録 (Register), 検証 (Verify), and 配備 (Deploy). The main workspace shows a workflow graph with nodes: a green node labeled 'ST-1-4-4_1', a blue node labeled 'service_ST-2-2-1_1', and a green node labeled 'PubSub_ST-1-4-5_1'. Below this, there is an orange node labeled 'python_ST-2-1-2-1'. A blue callout bubble points to this node with the text 'コードのチェック機能も有り' (Code check function also available).

On the right side, the 'python_ST-2-1-2-1 ノードを編集' (Edit python_ST-2-1-2-1 node) panel is open. It contains buttons for 削除 (Delete), 中止 (Cancel), and 完了 (Complete). Below these is the 'プロパティ' (Properties) section with fields for 名前 (Name) set to 'python_ST-2-1-2-1', 説明 (Description) set to 'python', and メタ情報 (Meta-information) set to '{"info1": "python", "info2": 123}'. There is also a 'ファイルを選択' (Select file) button with the text '選択されていません' (Not selected).

At the bottom of the panel is the 'コード' (Code) section, which is highlighted with a red border. It contains a text editor with the following Python code:

```
1 msg = "hello world!"
2 print(msg)
```

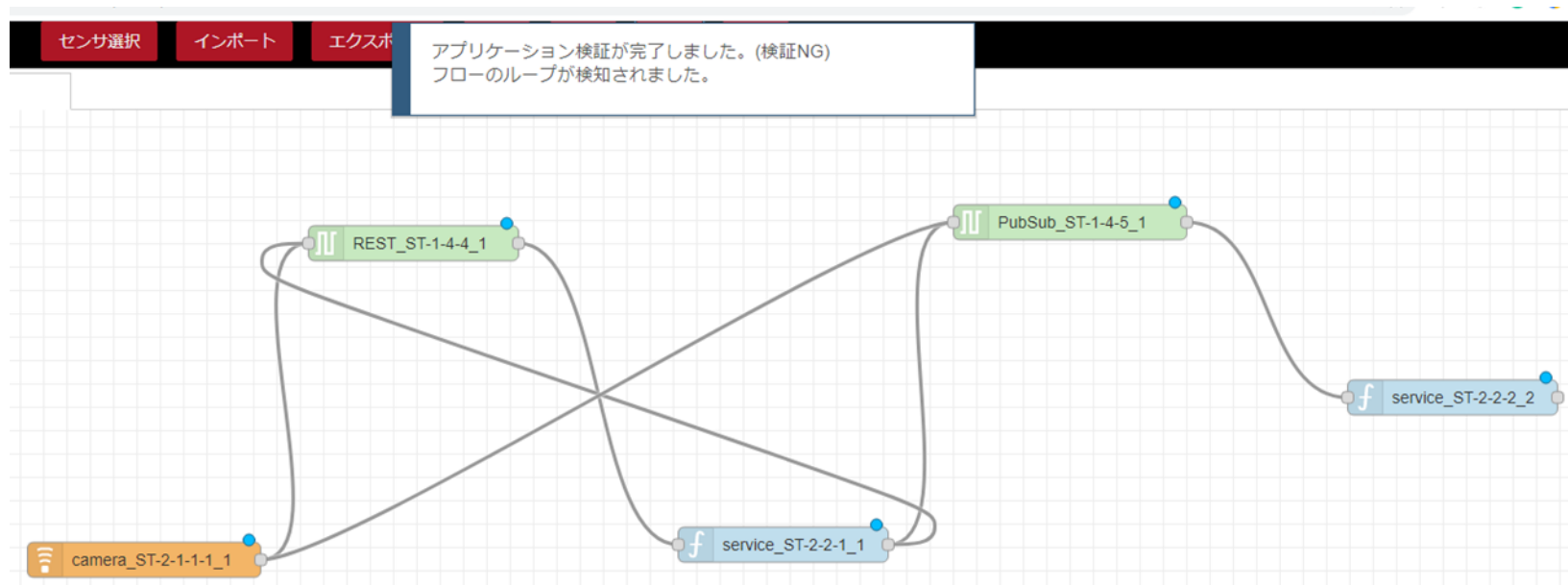
ThingVisor Factoryの機能紹介

- インポート/エクスポート
 - 作成中のIoTサービスをローカル環境にファイルとして保存/呼び出し
 - 作成しているIoTサービスは、JSONファイルで記述



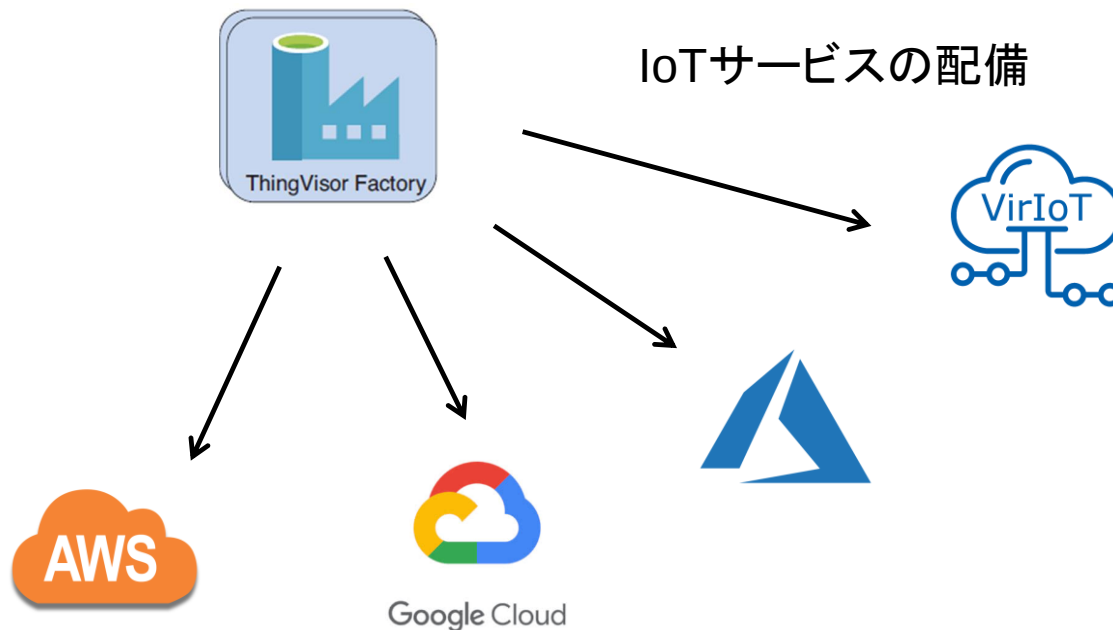
ThingVisor Factoryの機能紹介

- IoTサービスの検証
 - 作成したIoTサービスに矛盾が無いかどうかを検証
 - IoTサービスの始点がセンサ端末になっているか
 - 浮きフローがあるかどうか
 - 各機能が正しい通信プロトコルで接続されているか
 - IoTサービスにループが無いか
 - など



ThingVisor Factoryの機能紹介

- IoTサービスの配備
 - 作成したIoTサービスをクラウド環境やVirIoT環境へ自動配備
 - ThingVisor Factoryコントローラへ作成したIoTサービスを送信
 - コントローラがクラウド環境、VirIoT環境へ自動配備



まとめと今後の展望

- つながる・ながれる・つづく仕組みのためのVirIoT
- ThingVisor Factory: “IoTサービス”を生み出す工場
 - ビジュアルプログラミングに基づくIoTサービスの構築
 - クラウド環境への自動配備(オーケストレーション機能)
- ThingVisor Factoryはまだまだ発展途上
 - カタログサービスとしての側面
 - 利用可能なIoTセンサ、サービス、プラットフォームを提示
 - ユーザ間でのIoTデータやサービスの効率的な共有
 - ユーザビリティ向上に向けたGUIの洗練化
- 少しでもご興味がございましたらご連絡ください
 - Email: k.kanai@aoni.waseda.jp

発表文献リスト

- K. Kanai, H. Hidenori, H. Kanemitsu, A. Detti, “ThingVisor factory: thing virtualization platform for things as a service”, CCIoT’20, pp.7-12, Nov.2020.
- 金井, 中里, 金光, “Things as a Service を実現するThingVisor FactoryとIP/ICN間のThings共有アーキテクチャの提案”, 電子情報通信学会 信学技報CS2019-101, 2020年2月.

謝辞

本研究成果は、戦略的情報通信研究開発推進事業（国際標準獲得型）「スマートシティアプリケーションに拡張性と相互運用性をもたらす仮想 IoT-クラウド連携基盤の研究開発 (Fed4IoT) 【JPJ000595】」によるものである

Thank you Questions?

