

Things as a Service を実現する Fed4IoT プラットフォームの研究開発

金井 謙治¹⁾ 吉田 英聖²⁾ 金光永煥¹⁾³⁾ 中里 秀則¹⁾

横谷 哲也²⁾ 向井宏明²⁾ 中村健一¹⁾⁴⁾ 上杉 充⁴⁾

1) 早稲田大学理工学術院総合研究所 〒169-8555 東京都新宿区大久保 3-4-1

2) 金沢工業大学 〒921-8501 石川県野々市市扇が丘 7-1

3) 東京工科大学 〒192-0982 東京都八王子市片倉町 1404-1

4) パナソニック株式会社 〒105-8301 東京都港区東新橋 1-5-1

E-mail: k.kanai@aoni.waseda.ac.jp, nakazato@waseda.jp, {yokotani, mukai.hiroaki}@neptune.kanazawa-it.ac.jp

あらまし 筆者らは、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) を行っている。本稿では、Fed4IoT において研究開発している Things as a Service を実現する仮想 IoT システムについて提案する。本 Fed4IoT が提供する仮想 IoT システムのアーキテクチャ初版を紹介するとともに、その要素技術である IoT デバイス仮想化について、静的に構築する手法および動的に構築する手法について説明する。

キーワード IoT プラットフォーム, 仮想 IoT システム, IoT デバイス仮想化

Research Development of Fed4IoT platform for Things as a Service

Kenji KANAI¹⁾ Eisei YOSHIDA²⁾ Hidehiro KANEMITSU¹⁾³⁾ Hidenori NAKAZATO¹⁾

Tetsuya YOKOTANI²⁾ Hiroaki MUKAI²⁾ Kenichi NAKAMURA¹⁾⁴⁾ Mitsuru UESUGI⁴⁾

1) Waseda University, WISE 3-4-1 Okubo, Shinjuku-ku, Tokyo, 169-0072, Japan

2) Kanazawa Institute of Technology, 7-1 Ohgigaoka, Nonoichi, Ishikawa, 921-8501, Japan

3) Tokyo University of Technology, 1404-1 Katakura-machi, Hachioji, Tokyo, 192-0982, Japan

4) Panasonic Corporation, 1-5-1 Higashishinbashi, Minato-ku, Tokyo, 105-8301, Japan

E-mail: k.kanai@aoni.waseda.ac.jp, nakazato@waseda.jp, {yokotani, mukai.hiroaki}@neptune.kanazawa-it.ac.jp

Abstract In this paper, we propose virtual IoT system that provides “Things as a Service”. This proposal is a part of our research activity called Federating IoT and cloud infrastructures to provide scalable and interoperable Smart Cities applications, by introducing novel IoT virtualization technologies. This paper introduces an 1st version of architecture of proposed virtual IoT system and explain static and dynamic instantiation methods of virtual IoT device.

Keywords IoT platform, Virtual IoT system, IoT device virtualization

1. はじめに

近年、Internet of Things (IoT)の普及により、IoT のビジネス化、サービス化について、様々に取り組みされている。その中でも、IoT プラットフォーム技術について広く注目を集めている。IoT プラットフォームは、IoT デバイスの生成するデータを収集、処理、蓄積、アプリケーション開発環境といった機能を提供する。特に、データ収集、蓄積については、IoT プラットフォームの主要技術である IoT ブローカーが提供することもある。このような IoT プラットフォームは、現在、様々なベンダーや標準化団体によって開発が進んでいる。これら IoT プラットフォーム技術は、現状、必ずしも水平統合型のアプローチで進んでいるとは言えず、各

プラットフォームがカバーする範囲が限られていることや、ターゲットとするアプリケーション領域において、物理 IoT デバイスからアプリケーションまで垂直統合されており (サイロ化)、多様なアプリケーション領域で横断的に利用することや相互運用することは想定されていない [1] (図 1)。この主な要因として、先に述べたように、IoT プラットフォーム技術においてデータ収集、蓄積を担う IoT ブローカーでは、各プラットフォームにおいて固有のデータフォーマットを定義しており、このフォーマットに従い、デバイス管理、セッション管理、データ管理を行っている。そのため、データフォーマットが固有化されていることから、異なる IoT プラットフォーム間で、IoT ブローカーを介

したデータ共有や連携は困難と言える。このような中、多様なアプリケーション領域の集合体であるスマートシティでは、アプリケーション技術、IoT プラットフォーム技術、IoT デバイスの相互運用、連携が重要視されており、IoT プラットフォームを横断した相互運用技術やその基盤が求められている。

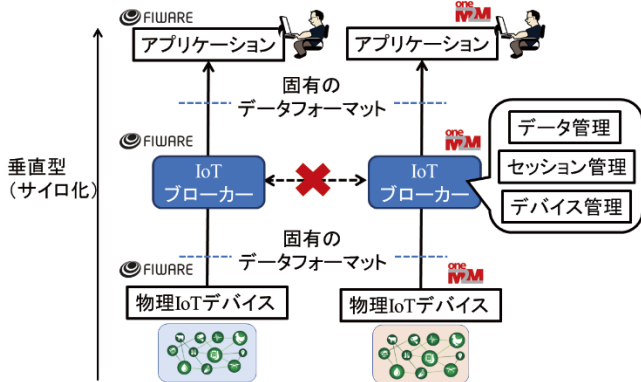


図 1. 想定するサイロ化された IoT プラットフォーム環境.

これに対して、本稿の筆者らは、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) を行っている。本 Fed4IoT では、各アプリケーションにおいてサイロ化された IoT プラットフォームに対して、拡張性と相互運用性を実現し、スマートシティにおけるアプリケーションの開発・展開の簡易化、持続可能性を実現することを目的としている。Fed4IoT では、図 2 に示す日欧にまたがる 4 つのスマートシティに対して 5 つのユースケースを設定し、各スマートシティ間で各ユースケースが生成するデータやコンテキストを効率的に共有する仮想 IoT システムを開発することを目指している。

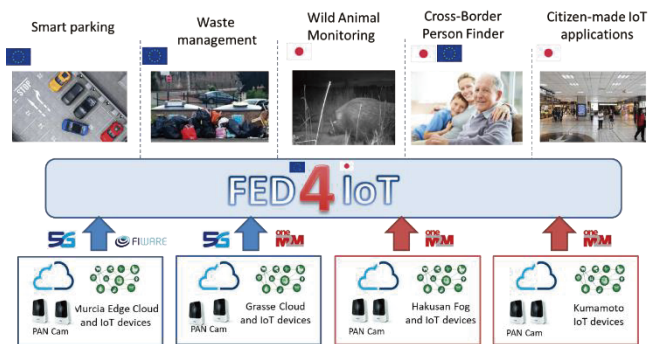


図 2. Fed4IoT が設定する 4 つのスマートシティと 5 つのユースケース.

その中でも本稿では、Fed4IoT が提供する仮想 IoT システムの概要とその主要技術である IoT デバイス仮想

化技術および IoT ブロカーの仮想サイロ化技術について紹介する。これら仮想化技術によって、アプリケーション、プラットフォーム、物理 IoT デバイスをそれぞれに分離し、特にアプリケーション開発者から見て、様々な物理 IoT デバイスが生成するデータ取得やその処理機能をサービス (Things as a Service) として提供することで、アプリケーション開発の促進を図る。

2. IoT プラットフォーム

[2, 3]で紹介されているように、現在、様々なベンダーや標準化団体により IoT プラットフォームの開発が進んでいる。本 Fed4IoT では、典型的な IoT プラットフォームとして、oneM2M[4, 5]と FIWARE[6]を挙げており、以下に、それぞれの概要を示す。

2.1. oneM2M

oneM2M[4, 5]は、水平統合型の IoT プラットフォームの標準化の一つで、主に ETSI をはじめ、日本からは ARIB, TTC が関わっている。oneM2M では、共通アーキテクチャと通信方式の二つを標準化している。アーキテクチャは、非常に単純化されており、アプリケーションを表す Application Entity (AE) と共通のサービス機能群を表す Common Service Entity (CSE) がそれぞれ通信を行うモデルとなっている。oneM2M では、AE 同士が繋がることはなく、必ず通信には CSE を介するよう規定されており、CSE がアプリケーションの管理、デバイスの管理、セッションの管理等の機能を担う。データモデルを含めたこれらの機能は、リソースとして定義されており、このリソース操作のための REST API が提供されている。通信方式として、HTTP の他に、CoAP や MQTT が想定されている。また、リソースは木構造の階層構造で定義されており、XML や JSON で記述されている。

2.2. FIWARE

FIWARE[6]は、NEC をはじめ多くの欧州の大手企業によって開発されている IoT プラットフォームの一つである。FIWARE では、アプリケーション領域の横断型のデータ利活用を目的に、各アプリケーション領域が生成するデータは、IoT ブロカー (コンテキストブローカー) を介してデータの流通を行い、IoT ディスカバリーによって、デバイス、セッション、データの管理等を行う。アプリケーションと IoT ブロカー間は、NGSI-10 と呼ばれるプロトコルによって定義されており、IoT ブロカーと IoT ディスカバリー間は、NGSI-9 と呼ばれるプロトコルによって定義されている。また FIWARE では、FIWARE 内に流れるデータについて、そのデータモデルの標準化を行っており、こ

の標準化されたデータモデルと共通プロトコルによって高度なデータ検索を可能としている。各 NGSI プロトコルは、REST によって API が定義されており、その通信には主に HTTP が利用されている。

3. Fed4IoT が提供する仮想 IoT システム

3.1. Fed4IoT の狙い

先に述べたように、Fed4IoT の狙いは、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ことで、低コストに IoT プラットフォームを構築し、簡易にスマートシティアプリケーションを実装、展開することを実現する。また、Fed4IoT は、新規に標準の IoT プラットフォームを構築するのではなく、FIWARE や oneM2M といった既存の標準 IoT プラットフォームを効率的に活用することにある。そこで、Fed4IoT によって、IoT デバイスを仮想化させ、既存の IoT プラットフォームが提供する IoT ブローカーによる仮想サイロを構築することで、アプリケーション領域とプラットフォーム領域を分断する。Fed4IoT は、これら IoT デバイスの仮想化技術および IoT ブローカーによる仮想サイロ化技術を提供することで、IoT システム全体を仮想化することを図る。

3.2. Fed4IoT アーキテクチャ初版

本章では、[7]で提案されている IoT デバイス仮想化技術と IoT ブローカーによる仮想サイロ化技術についてその詳細な説明と仮想 IoT システム全体のアーキテクチャ初版をあらためて示す。

3.2.1. IoT デバイス仮想化

まず IoT デバイス仮想化技術の概要を図 3 に示す。IoT デバイス仮想化技術は、スマートシティに設置されている物理 IoT デバイスを仮想化し、デバイスから生成されるデータやコンテキスト情報、またその処理機能をサービスとして提供する Things as a Service を実現する技術である。ここで図 1 に示すようなサイロ化された IoT プラットフォーム環境を想定する。図 1 に示すサイロ化された環境では、物理 IoT デバイスは、そこで提供されるアプリケーションのためのみに利用され、特定の IoT プラットフォームが提供する固有のデータフォーマットにより定義された API、データ構造を持つ。この環境下において、Things as a Service を実現するためには、固有のデータフォーマットを吸収する必要がある。そこで、IoT デバイスを仮想化する際には、図 3 に示すように、データの取得機能とニュートラルなデータフォーマットへ変換する機能を内包する。また、物理 IoT デバイスが生成する生データの

みならず、コンテキスト情報もサービスとして提供するために、データ処理機能も内包することとする。ここで、データ処理は、サービスファンクションチェイニングによるネットワーク内分散処理によって実現することも想定している。また、仮想 IoT デバイスは、単一の物理 IoT デバイスのみならず、複数の拠点に設置されている物理 IoT デバイスを組み合わせることも想定している。これら仮想 IoT デバイスを構築するソフトウェアを「ThingVisor」として定義する。このように、ThingVisor により、多種多様な IoT デバイスを仮想的に生成することが可能となり、Things as a Service の実現に繋がる。

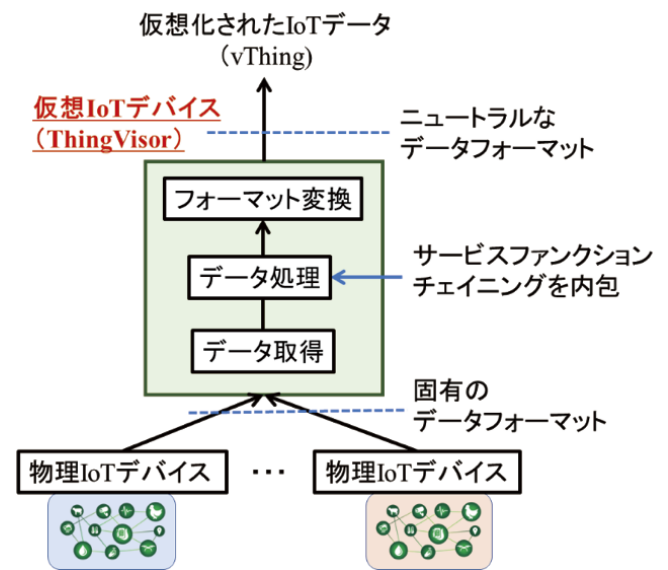


図 3. IoT デバイス仮想化技術の概要。

3.2.2. IoT ブローカーによる仮想サイロ化

次に IoT ブローカーによる仮想サイロ化の概要を図 4 に示す。先と同様に、図 1 に示すサイロ化された環境では、アプリケーションも利用している IoT プラットフォームに固有のものとなっている。そのため、このアプリケーションを再利用するためには、再度、想定する IoT プラットフォームに対応するデータフォーマットへ API も含めた変換する必要がある。そこで、これまで利用していたサイロ化された環境を仮想化するアプローチを取る。具体的には、図 4 に示すように、仮想 IoT デバイスから生成されるデータを取得し、特定のデータフォーマットへ変換後、仮想化された IoT ブローカーへデータを発行する。この枠組みを仮想サイロ (vSilo) として定義する。アプリケーション側は、これまでと同様に、これまで利用していた (仮想化された) IoT ブローカーへサブスクリプションすることで、アプリケーションに変更を加えることなく、仮想 IoT デバイスからデータを取得することが可能となる。

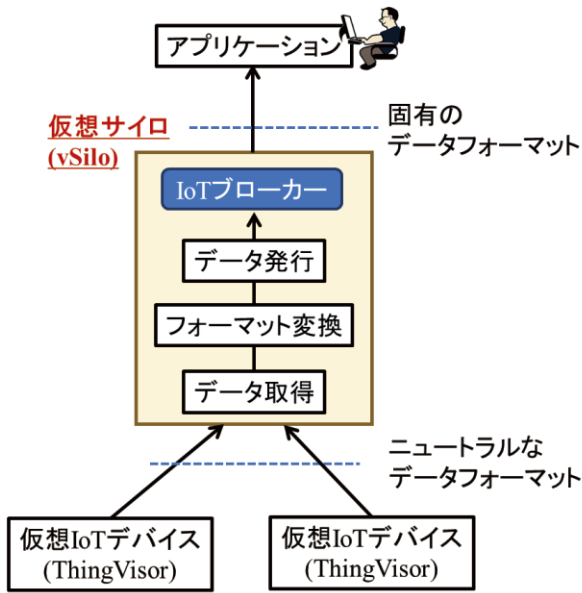


図 4. IoT ブローカーによる仮想サイロ化.

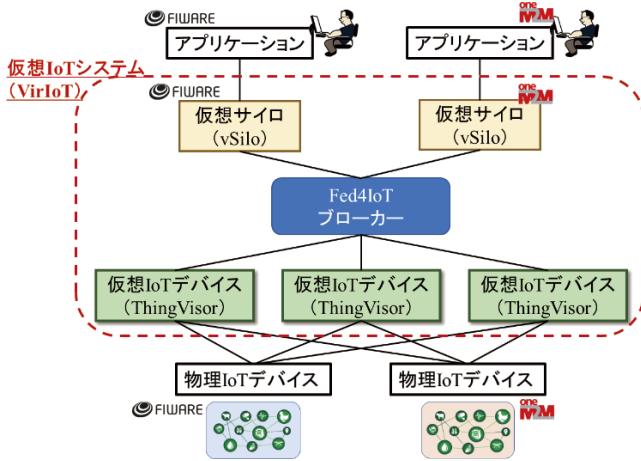


図 5. Fed4IoT が提供する仮想 IoT システムの全体図.

3.2.3. IoT システムの仮想化

最後に、上記仮想 IoT デバイスと仮想サイロを含む Fed4IoT が提供する仮想 IoT システム (VirIoT) の全体図を図 5 に示す。図に示すように、仮想 IoT デバイスを仮想サイロ間で共有するために、Fed4IoT では IoT システムのためのブローカーを間に設置し、このブローカーを介してデータ転送を行う。このように、アプリケーション間と物理 IoT デバイス間を分離することで、アプリケーション間において、IoT プラットフォームの違いを吸収しつつ物理 IoT デバイスを仮想的に共有することを可能となる。また、この仮想 IoT デバイスの構築について、アプリケーション開発者以外も提供することで、Things as a Service を実現している。

4. 仮想 IoT デバイスの構築手法

本章では、仮想 IoT デバイスの構築手法について述べ

る。先に示したように、仮想 IoT デバイスは ThingVisor によって構築されるが、この仮想 IoT デバイスの実体 (インスタンス) を生成するのに静的な手法と動的な手法の 2 種類が挙げられる (図 6)。静的に生成する手法は、予め仮想 IoT デバイスを実現するインスタンスをレポジトリに登録しておき、必要に応じてレポジトリからインスタンスをプルし展開する。ユースケースとしては、様々な地点の気温や天気といった気象情報のような一般的なセンサの組み合わせや多地点の監視カメラのようなアプリケーションにとって汎用的なセンサの組み合わせを想定しており、主にパブリック用途を想定している。一方で、アプリケーション開発者が独自にセンサを組み合わせるようなユースケースも存在する。この場合は、要求に合わせて ThingVisor をインスタンス化する必要があることから動的に生成する手法となる。ユースケースとしては、スマートホームといったプライベート用途、また、プライバシー情報を含んだよりセキュリティを担保する用途を想定している。次の節にて、動的に生成する手法の詳細を述べる。

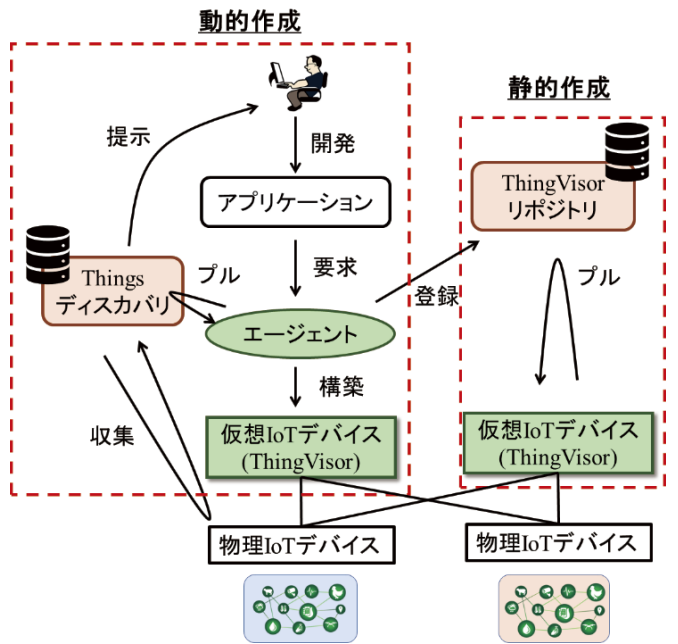


図 6. 仮想 IoT デバイスの構築手法.

4.1. 動的な仮想 IoT デバイス構築手法

動的に仮想 IoT デバイスを構築する (インスタンス化する) ためには、各スマートシティで利用可能な IoT デバイスを探索するフェーズと、探索フェーズによって発見された IoT デバイスからアプリケーション開発者の要求に合わせて仮想 IoT デバイスをインスタンス化するフェーズの二つに分けられる。

▶ IoT デバイス探索フェーズ

まず、IoT デバイス探索フェーズでは、各スマートシティで提供されている IoT デバイスの情報を収集する（図 6 における収集フェーズ）。この際、IoT デバイス情報として、デバイスのタイプや生成されるデータのみならず、設置されている場所情報やデータ取得のためのアクセス先の情報（例えばブローカー情報）、さらにデータ取得のための API といったコンテキスト情報を収集しデータベースとして保持する。これら収集したコンテキスト情報の一部（設置場所、センサタイプ、センサデータ等）を、アプリケーション開発者へ提示することで、アプリケーション開発者が自由にセンサを組み合わせることで独自の仮想 IoT デバイスを開発する。

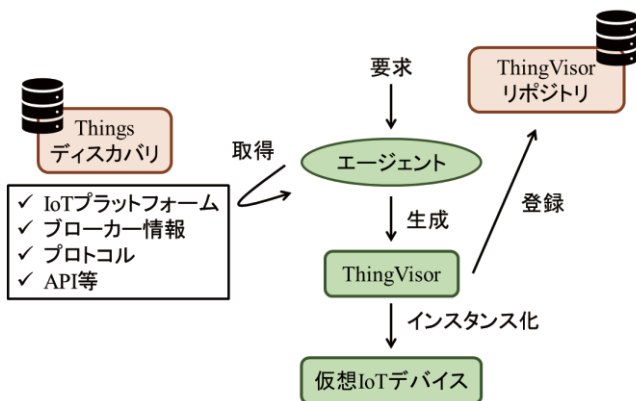


図 7. 仮想 IoT デバイス構築フェーズの流れ

▶ 仮想 IoT デバイス構築フェーズ

次に、アプリケーション開発者による仮想 IoT デバイス構築のための要求を受け取った後、エージェントが仮想 IoT デバイスをインスタンス化する ThingVisor を動的に生成する。図 7 に示すように、要求メッセージが生成された段階で、エージェントは、どの IoT デバイスを要求されているのかを把握し、その要求に合わせて「Things ディスカバリ」から、そのデバイスが利用している IoT プラットフォーム、そのブローカー情報、プロトコル、API を取得する。この情報をもとに、ThingVisor を動的に生成し、仮想 IoT デバイスをインスタンス化する。また、必要に応じて、仮想 IoT デバイスを共有するために、ThingVisor をリポジトリに登録する。

以上のステップにより、アプリケーション開発者の要求に従って、動的に仮想 IoT デバイスを構築し、アプリケーションが物理 IoT デバイスの情報を知ることなく、IoT デバイスの仮想化を実現する。

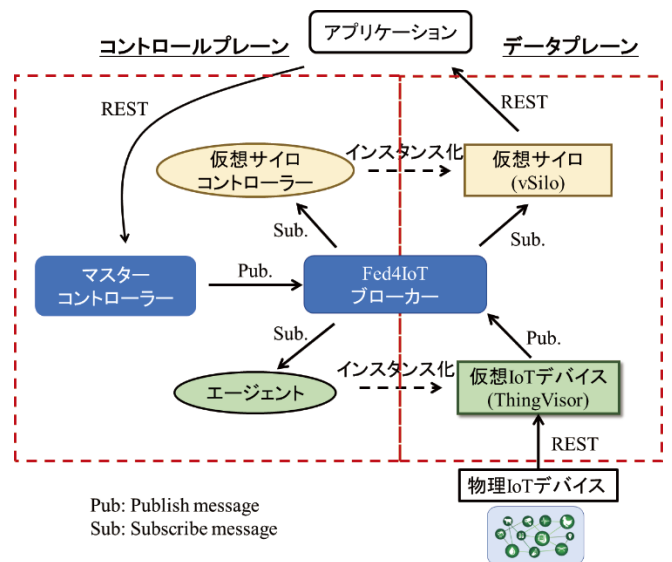


図 8. 仮想 IoT システムにおけるデータプレーンとコントロールプレーンのメッセージの流れ。

表 1: Pub/Sub トピックの例

	トピック名
エージェント側	/fed4iot/[エージェント ID]/control
仮想サイロ側	/fed4iot/[仮想サイロ ID]/control
データプレーン	/fed4iot/[仮想 IoT デバイス ID]/data

▶ 通信プロトコル

図 8 に、Fed4IoT が提供する仮想 IoT システムにおけるデータプレーンとコントロールプレーンのメッセージの流れを示す。図に示すように、データプレーンもコントロールプレーンも Pub/Sub 型プロトコルを採用する。データプレーン側は仮想 IoT デバイスによって収集されたデータは、Fed4IoT ブローカーへデータをパブリッシュする。仮想サイロは、Fed4IoT ブローカーにあらかじめサブスクライブすることによって、パブリッシュされたデータが転送され、最終的にアプリケーションへ届けられる。コントロールプレーン側も同様に、アプリケーションからの要求メッセージをマスターコントローラーが Fed4IoT ブローカーへパブリッシュし、あらかじめサブスクライブしていた ThingVisor のエージェント、仮想サイロコントローラーへ制御メッセージが転送され、仮想 IoT デバイス、仮想サイロのインスタンス化を行う。コントロールプレーン側も Pub/Sub 型を採用する理由としては、表 1 のように仮想サイロ、エージェント等の ID をトピックに付加することで、管理を簡易にすることを期待している。

5. フィールド展開について

本稿で紹介した仮想 IoT システムについて、現在フィールド展開を行っており、提案している動的に仮想 IoT デバイスを構築する手法について検証を行っている。展開しているフィールドは、図9に示すように、早稲田大学と金沢工業大学の各キャンパスを対象としており、早稲田大学側は、oneM2M のオープンソース実装の一つである OpenMTC[8]を利用しており、金沢工業大学側は、FIWARE[9]を利用している。仮想 IoT システムは、さくらクラウド[10]上に構築しており、IoT デバイスの仮想化およびサイロの仮想化については、Docker によりイメージを作成し、コンテナによりインスタンス化している。また、通信プロトコルについては、MQTT[11]および Apache kafka[12]を採用している。図9に、金沢工業大学に設置されているセンサーデータと早稲田大学に設置されているセンサーデータを組み合わせた仮想 IoT デバイスの出力結果を示す。

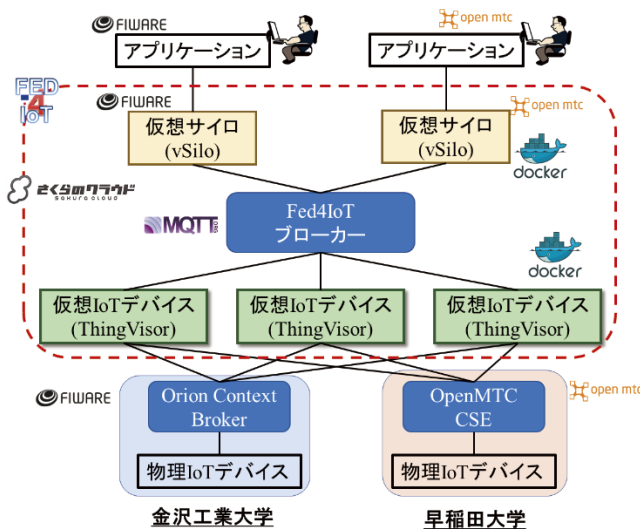


図9. フィールド展開のイメージ図。

```
[{"id": "urn:ngsi-id:bb440c270ee41a39c0373d1d0968ee", "type": "vThing", "WAS": [{"type": "Temperature", "value": 26, "unit": "degreeC"}, {"type": "KIT": [{"type": "Temperature", "value": 26, "unit": "degreeC"}]}], [{"id": "urn:ngsi-id:bb440c270ee41a39c0373d1d0968ee", "type": "vThing", "WAS": [{"type": "Temperature", "value": 26, "unit": "degreeC"}, {"type": "KIT": [{"type": "Temperature", "value": 26, "unit": "degreeC"}]}], [{"id": "urn:ngsi-id:bb440c270ee41a39c0373d1d0968ee", "type": "vThing", "WAS": [{"type": "Temperature", "value": 26, "unit": "degreeC"}, {"type": "KIT": [{"type": "Temperature", "value": 26, "unit": "degreeC"}]}], [{"id": "urn:ngsi-id:bb440c270ee41a39c0373d1d0968ee", "type": "vThing", "WAS": [{"type": "Temperature", "value": 26, "unit": "degreeC"}, {"type": "KIT": [{"type": "Temperature", "value": 26, "unit": "degreeC"}]}]
```

図10. 仮想 IoT デバイスが生成するセンサーデータ例。

6. まとめと今後の展望

本稿では、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 Fed4IoT について紹介するとともに、その要素技術の一つである IoT デバイス仮想化技術について、静的に構築する手法と動的に構築

する手法を紹介した。今後は、現在構築している仮想 IoT システム全体のオーバーヘッド、ネットワーク負荷を含め、その性能を評価していく予定である。

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)」によるものである。

文 献

- [1] 長尾, 亀津, “ビジネス価値につながる IoT の実現-最適な IoT プラットフォームをどう選択するか-”, 野村総合研究所, IT Solutions Frontier, 2018 年 2 月. [online]: https://www.nri.com/-/media/Corporate/jp/Files/PDF/knowledge/publication/it_solution/2018/02/ITSF180205.pdf?la=ja-JP&hash=F0ADC26F649E66BC83E7816B4BE9A06B7326124F
- [2] 木下, “IoT プラットフォーム標準化動向”, インターネット白書 Archives 2017, [online]: <https://iwparchives.jp/files/pdf/iwp2017/iwp2017-ch02-01-p114.pdf>
- [3] 総務省, “国際的な IoT の進展状況”, [online]: <http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h28/pdf/n2300000.pdf>
- [4] 山崎, “M2M 技術のアーキテクチャ及び oneM2M 標準化動向”, 通信ソサエティマガジン No.29, 2014.
- [5] 原田, 前大道, 山崎, “水平統合型 IoT プラットフォーム標準規格 oneM2M の最新動向”, NTT 技術ジャーナル 2018 年 2 月.
- [6] 竹内, 寺澤, “データ利活用型都市経営を実現する情報プラットフォーム: FIWARE”, NEC 技報, Vol.71, 2018 年 11 月, [online]: <https://jpn.nec.com/techrep/journal/g18/n01/pdf/180110.pdf>
- [7] A. Detti, G. Tropea, G. Rossi, J. A. Martinez, A. F. Skarmeta, H. Nakazato, “Virtual IoT Systems: Boosting IoT Innovation by Decoupling Things Providers and Application Developers”, Global IoT Summit, 2019.
- [8] OpenMTC [online]: <https://www.openmtc.org/>
- [9] FIWAER [online]: <https://www.fiware.org/>
- [10] さくらクラウド [online]: <https://cloud.sakura.ad.jp/>
- [11] MQTT [online]: <http://mqtt.org/>
- [12] Apache kafka [online]: <https://kafka.apache.org/>