

A Standard-based Open Source IoT Platform: FIWARE

Flavio Cirillo*[†], Gürkan Solmaz*, Everton Luís Berz*, Martin Bauer*, Bin Cheng*, Ernoe Kovacs*

* NEC Laboratories Europe, Heidelberg, Germany

[†] University of Naples Federico II, Naples, Italy

E-mails: {name.surname}@neclab.eu

Abstract—The ever-increasing acceleration of technology evolution in all fields is rapidly changing the architectures of data-driven systems towards the Internet-of-Things concept. Many general and specific-purpose IoT platforms are already available.

This article introduces the capabilities of the *FIWARE* framework that is transitioning from a research to a commercial level. We base our exposition on the analysis of three real-world use cases (global IoT market, analytics in smart cities, and IoT augmented autonomous driving) and their requirements that are addressed with the usage of *FIWARE*. We highlight the lessons learnt during the design, implementation and deployment phases for each of the use cases and their critical issues. Finally we give two examples showing that *FIWARE* still maintains openness to innovation: semantics and privacy.

I. INTRODUCTION

The evolution in all fields of technology is accelerating the shift towards the Internet-of-Things (IoT) vision in any area where data are produced and analyzed.

The market of IoT platforms is comprised of a huge number of solutions. From a business perspective, we can cluster them in two classes of platforms: commercial and open source. The commercial platforms are more stable and more appealing for industrial and business-oriented scenarios due to the contractual support of the providers. The open-source platforms are usually implementing standards and maintained by communities of researchers and innovators, enhancing them with cutting-edge trends, driven by use case and practical requirements. Here we discover the transition of *FIWARE*¹ from research to commercial.

A. *FIWARE* and its evolution

The *FIWARE* framework is supported by the European Commission since almost a decade, through funding several projects either for developing and enhancing it (e.g., *FIWARE*, *FI-Core*, *FI-Next*, *SmartSDK*) or for using it in pilots (e.g., *City Platform* as a *Service - CPaaS.io*², *frontierCities*, *Fiware4Water*). Lately, the *FIWARE* framework is going through a transition phase towards business readiness. This is demonstrated by the European large scale pilots projects (each with 15-20 Million Euro of funds), such as *SynchroniCity*³, *Autopilot*⁴, *Internet of Food and Farm - IoF2020*, and *Acti*

focusing on smart cities, autonomous driving, agriculture & food and e-health, respectively.

II. PUBLIC GOVERNANCE AND GROWTH: GLOBAL IOT MARKET

Public governance is attentive to new technologies that have a strong potential for economic growth, public safety, and citizens' well-being. IoT is certainly promising all these benefits. This has triggered in the past decade a widespread adoption of a plethora of IoT solutions in many urban scenarios. However, closed commercial APIs and implementations hamper an open market development (vendor lock-in). For this reason, an open approach is preferred by public institutions, as commonly agreed, for instance, by the 140+ cities of the *Open & Agile Smart Cities (OASC)* network⁵. Horizontal harmonization is provided locally in cities such as *Milan*⁶ and *Helsinki*⁷ achieving good results but remaining regionally isolated, thus resulting in city lock-in. Both vendor lock-in and city lock-in discourage small and medium-sized enterprises (SMEs).

With this rationale, *SynchroniCity* project proposes to synchronize existing smart city solutions by overcoming their misalignments. More specifically, the synchronization targets enabling five *Minimal Interoperability Mechanisms (MIMs)*:

- 1) Context Management
- 2) Data Models
- 3) Ecosystem Transaction Management ("Marketplace")
- 4) Security
- 5) Storage

In 2019 the first three MIMs have been officially adopted by the *OASC* consortium [1].

The *SynchroniCity* project demonstrates the feasibility of such an approach by harmonizing eight cities (*Antwerp*, *Carouge*, *Eindhoven*, *Helsinki*, *Manchester*, *Milan*, *Porto*, and *Santander*). The final goal is the co-design of IoT services [2] and IoT applications, and the establishment of a shared market fostering economic growth.

A. Legacy IoT platforms are not neglected

In today's smart cities it is common to have platforms that are the outcome of past projects and pilots of city governments.

¹<https://www.fiware.org/>

²<https://cpaas.bfh.ch/>

³<https://synchronicity-iot.eu/>

⁴<https://autopilot-project.eu/>

⁵<http://oascities.org/>

⁶<http://www.milanosmartcity.org/joomla/>

⁷<https://www.helsinki-smart.fi/>

TABLE I
DEVICE INTERFACES SUPPORTED BY FIWARE

IoT Devices	HTTP, MQTT, AMQP
Low-power wireless sensors	LoRaWAN, SigFox
Smart Industry	OPC-UA, ROS2

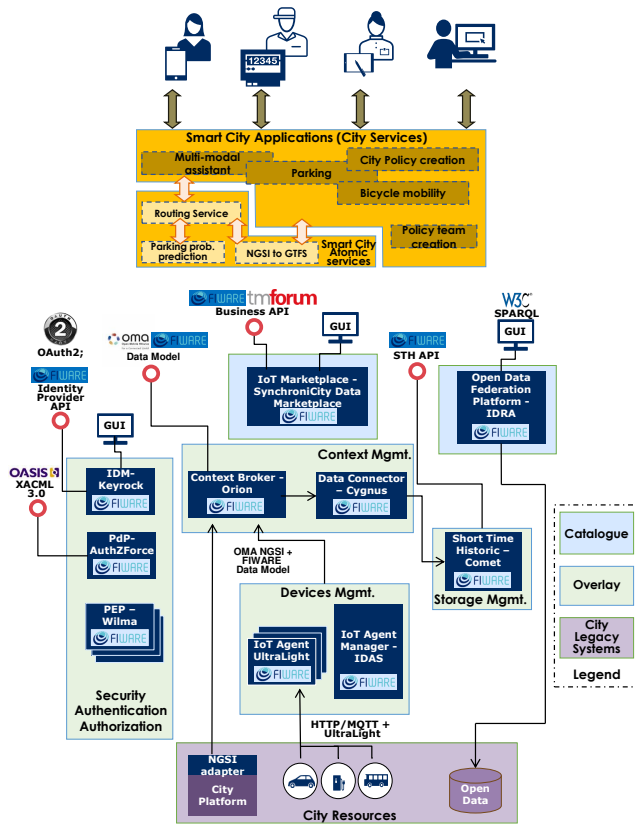


Fig. 1. Federation of pilot sites through the FIWARE IoT platform.

Often these solutions are ad-hoc and only used for the local smart city ecosystem and market. City governments are not keen on throwing away the obtained achievements, but they are interested in expanding the IoT business. Thus, instead of re-designing systems and replacing the already operational ones, in SynchroniCity a generic overlay platform over existing infrastructures is realized. This overlay finds in the FIWARE framework many building blocks to implement the five foreseen MIMs (see Fig. 1). Starting from the available city resources (the purple box in Fig. 1) data is integrated and exposed in several manners: sensor streams, open data, and historical time series.

B. Harmonizing smart cities data

The sensor streams are integrated with different approaches depending on the requirements and constraints given by the different city platforms. For most of the involved cities, ad-hoc integration modules are implemented and deployed. These obtain data from the running platforms and translate them to the Next Generation Service Interface (NGSI) [3] standard data format. Only in a few cases, such as Santander and Porto, sensor data are exposed through typical IoT interfaces, such as MQTT and HTTP. For native IoT device interfaces, the FIWARE framework offers IoT Agents (see Table I) that translate device interfaces to NGSI. Having all data formatted using the same data structure is not enough to ensure harmonization since data might be modeled very differently by

IoT developers. For instance, a location might be referred to as "position", "geolocation" or a term in another language. Therefore, the used adapters, both ad-hoc modules and IoT Agents, generate context data following the FIWARE/OASC data models⁸. These data models are defined by the FIWARE community and adopted by OASC, and they harmonize the description of data for several application areas such as parks and gardens, points of interest, parking, waste management, transportation, and weather.

The homogeneously modeled data is then handled by the context management layer that exposes context with a standard interface, which is NGSI, fulfilling the Context Management MIM. NGSI specifies both a context management interface with an HTTP binding and a context data format using JSON. SynchroniCity adopted the Orion Context Broker (CB) as the context manager implementation, which holds the latest attribute values for each entity (or "thing") and exposes NGSI query and subscription methods. Orion is conceived to work with high-frequency messages and to respond to queries with minimal latency. For stream-based applications, Orion offers an NGSI subscription interface, notifying with atomic notifications as soon as a matching attribute of a matching entity becomes available. A sensor generating a stream of data with small intervals between observations, such as an accelerometer, might create a flood of notifications in the network. Thus, a "throttling" parameter can be set in the subscription. This regulates data notification streams, instructing Orion to issue two notifications for the same subscription apart in time for at least the throttling period. The drawback is that it might result in data loss in case more than one value for a matching attribute is generated within the throttling period. The missing value is not an option for some applications. For this reason, SynchroniCity adopted the Comet Short Time Historic (STH)⁹ component. The Cygnus connector receives the NGSI data stream from Orion and forwards it to a persistent data sink, such as STH Comet (in the SynchroniCity case), but also to other commonly used data storage systems such as MongoDB, Hadoop Distributed File Systems (HDFS) for big data processing, or CKAN for Open Data publication. STH and Cygnus together address the Storage MIM.

C. IoT Marketplace

Often, in smart cities, there are already many services that produce and use data for their purposes, such as urban facilities, public transportation, tourism operators. Unfortunately, these IoT providers have no interest in spending effort on

⁸<https://www.fiware.org/developers/data-models/>

⁹<https://github.com/telefonicaid/fiware-sth-comet>

sharing their data if there is no payback for it. On the other side, companies that might want to create smart city services need data and they are keen on paying a fee for datasets otherwise impossible to get. Here the necessity of having an IoT marketplace arises. Available data needs to be cataloged: a) as valuable assets in case of private companies, b) as open data in case of public institutions.

For the first scenario, a gap is identified, and, therefore, the project worked closely with the FIWARE community to close it. FIWARE officially adopted the TM Forum business API standard which is implemented within the SynchroniCity project. This software component, named SynchroniCity IoT Data Marketplace¹⁰, exposes a catalog of available data (either in Orion or in STH Comet) describing endpoints to access them, license, and price. Users can buy data items and the marketplace sets access rules in the security layer to allow the data exchange. To smooth the usage of the rather complex business API, scripts are available to automatically integrate the FIWARE data management by crawling Orion and publish the found data items in the marketplace.

In the case of open data, smart city governments usually have already published datasets, most of the time on their ad-hoc platforms or simply on their institutional website. That is not handy for a data consumer that is completely unaware of the city governance structure and, therefore, of the different institutional web pages. The situation is even worse when the open data website is only available in the local language. This situation is addressed by the Open Data Federation Platform IDRA¹¹ that was developed in the context of the FESTIVAL EU project that faced a similar scenario. Available datasets are federated in IDRA either via typical open dataset interfaces (such as CKAN) or simply via the web scraping functionality that crawls the generic website for datasets. All the federated datasets are then exposed by a graphical user interface.

D. Security

Identity management and access control requirements of SynchroniCity are not specific to smart cities and IoT. Therefore, common standards are used, such as OAuth2 protocol for authorization, and Oasis eXtensible Access Control Markup Language (XACML) 3.0 protocol for access control. The FIWARE framework already offers software components for both, namely KeyRock and AuthZForce respectively, that are already well integrated with the other FIWARE components (such as the IoT marketplace).

E. Lessons learned for the creation of a Smart Cities global market

Forgoing from a local to a global market, SynchroniCity tackles three aspects: the IoT service interface, the data models, and an IoT marketplace where data can be bought and sold.

The first barrier faced by SynchroniCity is that putting aside running legacy systems is not a solution for city governments,

even if this enables a business ecosystem and, perhaps, it boosts local IoT economic growth. Thus, an overlay on top of the running IoT infrastructure is designed and deployed in every pilot city. The overlay harmonizes live sensors data and open data. FIWARE offers off-the-shelf solutions in case of data exposed through common IoT interfaces (see Table I), and for open data (i.e., IDRA). In the case of data exposed through proprietary platforms, the burden of creating ad-hoc adapters cannot be avoided.

A technical issue occurred when data streams from sensors with very frequent observations are not optimally handled by the Orion CB component. With the latter, either too many notifications flood the network, or data may be lost. The solution for SynchroniCity is to use a time-series database as a data sink. A different approach to this problem is the usage of the Aeron IoT Broker component, as in the CPaaS.io scenario (see below).

The fact that FIWARE is supported by a community directly helps in making such a framework useful for IoT scenarios since it is driven by practical problems. Indeed, SynchroniCity could leverage the outcome of other projects to solve open data integration. SynchroniCity itself contributes to the IoT marketplace, releasing it as free to use¹², thus, closing a gap in FIWARE.

III. DATA ANALYTICS ON IoT FEDERATION: A SMART CITY SCENARIO

IoT services require IoT data. As seen above, in a smart city data may come from different public providers, such as public transportation or traffic management, and it may be centrally handled by the city governance. But what happens when data comes from private entities, such as a company, or even private citizens? Such private providers like to keep their own IoT infrastructure and their data, and not give data away to a centralized unknown platform. Providers are willing to share their datasets [4], if they are licensed and protected by access control, or even to earn money considering data as valuable assets. Such a scenario of fragmented IoT platforms is a nightmare for IoT service providers since they need to look for IoT providers and make a great effort on integrating heterogeneous data sources. Furthermore, an IoT service is typically composed of multiple data analytics routines, each exploiting a different set of IoT data and depending on each other. And what would happen if the IoT service provider wants to port its service in another environment? Simply additional effort on data providers discovery, data integration, and analytics orchestration.

A. City Platform as a Service

The CPaaS.io [5] EU-Japan project (City Platform as a Service-Integrated and Open) faced these problems in a smart city scenario. To overcome such a situation, CPaaS.io defined the following fundamental requirements to be addressed:

- i) allow easy integration of data sources into the platform (e.g., sensors operated by private citizens or established providers),

¹⁰<https://iot-data-marketplace.com>

¹¹<https://opendata.synchronicity-iot.eu/>

¹²<https://github.com/capossele/SynchroniCityDataMarketplace>

TABLE II
FIWARE-BASED COMPONENTS PRESENT IN THE CPaaS.IO PROJECT

CPaaS.io architecture layer	Component
Security and Privacy	KeyRock PEP-Proxy
Data Analytics Routines Management and Operation	FogFlow
Semantic Data & Integration	IoT Knowledge Server NGSI to RDF Mapper STH Comet
Virtual Entity	IoT Discovery Context Broker
IoT Data and Ingestion	IoT Broker
IoT Resource	IoT Agent: LoRaWAN to NGSI

- ii) offer services and federation capabilities among IoT platform instances,
- iii) support the deployment in many cities with distinct requirements (flexibility and elasticity),
- iv) self-orchestration of data analytics processing routines, each part of the same IoT service, among multiple IoT data providers,
- v) provide security mechanisms for the desired privacy and data protection,
- vi) demonstrate that the use cases developed in one city can be transferred to other cities.

To test the feasibility and effectiveness of the solution, CPaaS.io targets three smart city use cases spread among Europe (Amsterdam) and Japan (Sapporo, Yokosuka, Tokyo), such as water management, public event management, and public transportation.

The CPaaS.io platform exploits different technology to address the identified requirements, such as IoT platform federation, IoT data access control, security & privacy and data analytics tasks orchestration. Also, in this case, the FIWARE framework provides support on the European side, while the u2 platform [6] is used on the Japanese side. The solution of adopting two heterogeneous platforms validates the viability of a loosely coupled federation in the real world [5].

The European platform is FIWARE-based where FIWARE components are deployed and, in some cases, enhanced to provide the basis for a smart city data infrastructure. Table II presents a list of the FIWARE framework components in use by the CPaaS.io project and the mapping of each component onto the CPaaS.io functional architecture.

B. Urban water management scenario

As a relevant application scenario, Waterproof Amsterdam uses the CPaaS.io platform for water management. In the urban context, periods of drought and sudden heavy flashes of rainfall occur more and more often due to the urbanization trend and global warming. In both weather conditions, smart water management is required to ensure water availability, and to handle high pressure in the sewerage infrastructure to prevent street floods. The solution to this problem is water supply peak shaving. This is done by using smart and integrated rain buffers, such as rain barrels, retention rooftops, and retention buffers. These smart devices are centrally controlled

by the IoT water management service that makes use of weather information, sewerage capacity, and environment data to calculate the optimal water filling degree for each buffer.

The Waterproof application scenario necessitates several functional layers:

- *IoT Resources layer* for handling the communication with devices and sensors.
- *IoT Data and Ingestion layer* to persist and index, and to expose data to IoT data consumers.
- *Virtual Entity layer* that holds representations in the virtual world of the observed things.
- *Data Analytics Routines Management and Operation layer* to orchestrate the different data processing tasks for computing the operations for the smart devices.

The overall system architecture of the Waterproof solution is presented in Figure 2. Similar to the SynchroniCity case, the IoT resources are handled through IoT Agents, but in this case, using the Long Range Wide Area Network (LoRaWAN) protocol. Even though a LoRaWAN IoT Agent is already available in the FIWARE framework, a new one has been implemented in order to optimize it for the specific Waterproof use case, starting from the FIWARE IoT Agent library¹³ for accelerating the software development. For handling data, instead of the Orion Context Broker, as in SynchroniCity, the Aeron IoT Broker is used, which has additional features such as federation readiness both for query and subscription [7], and throttling-based aggregation of notifications. The latter permits to receive data notifications not more often than the throttling but encompassing all sensor observations pushed within a period through aggregation such as to avoid data loss. The aggregation can be appending all the data values in a set, or applying an actual function (e.g., averaging). The drawback is a bit higher latency compared to Orion, which is acceptable in the Waterproof use case as it is a smaller IoT scenario compared to a complete city in the case of SynchroniCity. Afterward, all the data are stored in the STH historical component, which is extended to handle also the historical storage of metadata.

The virtual world representation is kept within the FIWARE IoT Discovery which acts as a registry of the available data and resources with associated metadata. This layer abstracts real-world objects (things) from the observation points (sensors). The IoT Discovery allows us to query and subscribe for resources by requesting the relevant context, such as a street name.

The Waterproof data analytics is broken down into atomic analytics tasks that are linked in a chain or a generic topology. This brings the advantage of flexibility. With such an approach, the plugging of the right data streams to the tasks' inputs is not hardcoded, and, therefore, an external process can automatize it. For this purpose, CPaaS.io uses the FIWARE FogFlow [8] framework. The latter leverages the virtual world representation to discover the needed resources and plugs them to task inputs. The resources can be physical (e.g., a sensor stream), or generated, such as intermediate results of one of the analytics tasks (e.g., geographical data aggregation).

¹³<https://iotagent-node-lib.readthedocs.io>

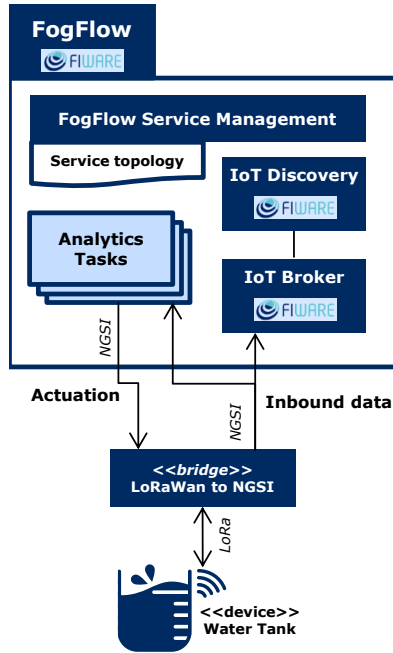


Fig. 2. Waterproof architecture in the CPaaS.io platform

With such an approach, porting the service topology into another environment is smooth, since FogFlow works at the higher abstraction level of the virtual world. When porting the Waterproof application, the only configuration needed is the description of the available computing nodes. In this application, analytics tasks are responsible for identifying heavy precipitations and for computing how to set up the water network to avoid damage. The analytics service subscribes to inbound data streams to constantly monitor the state of the water buffer and trigger actions when needed.

FogFlow is conceived to orchestrate tasks also among cloud and edge [8], which is a feature to be used in the future for exploiting the computing resources of the deployed devices.

C. Lessons learned from smart city services

Federating IoT systems is often not an option [7]. Therefore, CPaaS.io sets federation as a target aiming at integrating private and heterogeneous IoT systems towards a global Internet-of-Things. Luckily, the NGSI protocol has been designed to support federation, and the FIWARE framework already offers implemented solutions with the Aeron IoT Broker and NEConfMan IoT Discovery components [7].

The water management use case helped to identify other technical gaps not foreseen beforehand. For instance, as the LoRaWAN to NGSI bridge component requires high throughput and low latency, a new lightweight component has been developed by The Things Network¹⁴ to address those performance needs. Another gap was the support of metadata in time series storage that is addressed by extending the STH component. In NGSI, metadata means additional data about attribute values. For instance, in the Waterproof application, the time of measurement and measurement unit are metadata

¹⁴<https://www.thethingsnetwork.org>

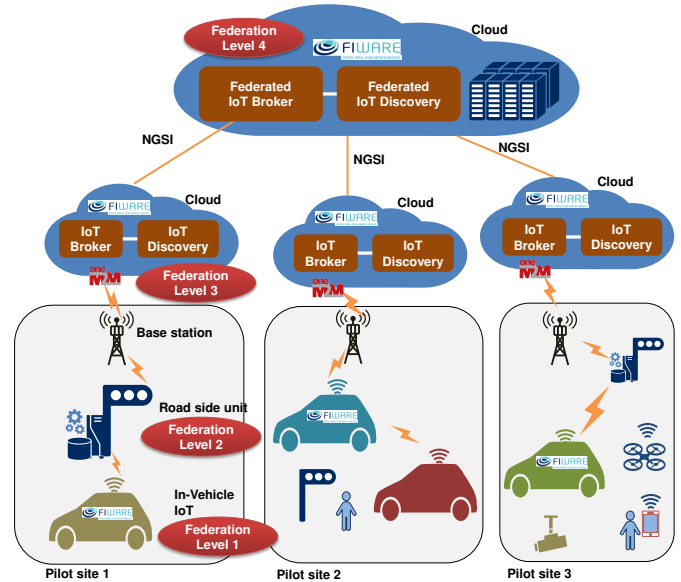


Fig. 3. Federation of large scale pilot sites through a FIWARE-based IoT platform.

used by the water management system and the metadata need to be historically persisted together with attribute values. The implementation of the new LoRaWAN IoT Agents and the extension of the STH component have been made as part of the CPaaS.io project thanks to the open-source nature of the FIWARE framework. In fact, in the first case, the available library has been used in the implementation and in the second case the source code of the component has been extended.

IV. RESEARCH AND INNOVATION: AN AUTOMATED DRIVING SCENARIO

In CPaaS.io and SynchroniCity, we have seen requirements for smart cities. But how can we handle extreme scenarios where applications need very low latency, high computation power, ubiquitous computing, and different administrative IoT domains? This is the boundary situation of the smart mobility domain which is the main focus of the AUTOPILOT project.

AUTOPILOT aims to leverage IoT technologies for enhancing automated driving. The smart mobility use cases include applications of car sharing, car re-balancing, autonomous platooning, and automated valet parking. One key aspect of the future smart mobility scene is providing interoperability between various components which are in-vehicle (e.g., autonomous cars), on-site (e.g., roadside units - RSUs), or in the cloud (e.g., traffic operation center). Hence, the project aims to support the flow of IoT information from the vehicle to the cloud and back to the vehicle passing through on-site units and network infrastructures. Besides, such applications need to be able to work when a car passes from one city to another, and, therefore, it passes road administration boundaries typically handled by different institutions.

A. Federation of large scale pilots

Together with common data models, the decentralized IoT architecture aims to enable long-distance automated travels, as

described in [9]. For instance, when an autonomous vehicle which is registered in Spain travels to the Netherlands, it should be able to operate using information coming from the city through which it is currently driving, while still pushing its key measurements to its home city. Information such as braking distance in different environments, which may indicate the status of its tires or the brake pads, may still be stored in the Spanish pilot site whereas the current location of the vehicle is stored in real-time on servers in the Netherlands. As another example, traffic situation information (e.g., congestion, traffic lights) from different cities are normally served by different vendors that do not federate data with each other. Enabling the sharing of data allows an autonomous car traveling from one country to another to learn the traffic situation in its current country.

For realizing the federation of pilots, the combination of open-source FIWARE Aeron IoT Broker and NEConfMan IoT Discovery¹⁵ components is used to transparently route IoT datasets and data streams from providers to consumers [7]. The transparency is given by the hidden brokering of the requests among IoT actors, where the IoT providers declare the available data and the IoT consumers specify the data of interest. An actor can play both roles, for instance, a car might be a data provider with geographic localization or a consumer of traffic information. The exchange might go from one pilot site to another as illustrated in Fig. 3. Federation starts from the vehicle (Level 1) with the in-vehicle IoT platform, where our lightweight version of the IoT Broker, called *ThinBroker*¹⁶, can run on a server in the car. This component can run even in small devices such as a Raspberry Pi.

The autonomous vehicles in the same region are connected (vehicle-to-vehicle communications) or as shown in pilot site 1, cars can connect to a roadside unit. In this case, RSUs are considered as the “edges” with processing and storage capabilities where IoT platform components are deployed as the Level 2 federation. In automotive scenarios, the IoT devices involved are not simply sensors but rather compound devices exposing many different interfaces. For that reason, instead of implementing several brand new IoT Agents, the existing interworking between oneM2M and other IoT technologies is leveraged [10]. As illustrated on the cloud side of Fig. 3, each pilot site has a oneM2M implementation for device-to-device communication as well as interworking between several IoT platforms. The cloud-side IoT platform is considered as Level 3 of the federation. This is formed again by IoT Brokers that (besides routing requests to/from oneM2M) also handle data locally which are managed by the pilot site administration. Finally, there exists a federated IoT platform that operates across all pilot sites. Level 3 and 4 of the federation operate with common information models (data models). AUTOPILOT defines a new data model supporting the IoT information for autonomous driving vehicles, considering the AUTOPILOT use cases and the existing legacy models such as SENSORIS¹⁷ and DATEX II¹⁸. This data model

covers IoT information such as vehicles, road infrastructure, road conditions, traffic situations (e.g., congestion), accidents, pedestrians, cyclists, vulnerable road user detection, events, road obstacles, potholes. The data is then mapped into the standard NGSI data format.

B. Semantic Interoperability

Interworking between oneM2M and NGSI is achieved through semantic annotations [10]. In AUTOPILOT the interoperability features are provided by the Semantic Mediation Gateway (SMG) technology. SMG offers bidirectional context translations between oneM2M’s *Mca* reference point and the NGSI interface. While the syntactic representations in oneM2M and NGSI are different, interoperability is enabled through the agreement on semantics, i.e. the same meaning, mapping homologous underlying concepts, which enables an SMG to do the translation. Figure 4 shows the basic setup for AUTOPILOT where the bottom layer consists of autonomous vehicles, RSUs, and pedestrians (e.g., people with smartphones) interacting with the top layer consisting of autonomous driving applications or traffic operation centers (top of Fig. 4). The interaction is provided through the FIWARE-based IoT platform. Some devices such as mobile devices of pedestrians connect and push data to the oneM2M platform. In some cases data is directly pushed to the FIWARE platform, or, otherwise, acquired by the SMG from oneM2M. Nevertheless, other third-party platforms or components can receive information directly through oneM2M’s *Mca* interface.

The real-time system, consisting of in-vehicle components, oneM2M and FIWARE IoT platform, is tested at a major pilot site in the Netherlands. For instance, data coming from WiFi scanners of the autonomous vehicle and RSUs are used by analytics modules to estimate crowdedness information in the regions of interest. The contextualized crowdedness information is provided using the standardized data models in NGSI format. This information is then shared with the automated driving application to decide the travel route of the autonomous vehicles. As a result of having an agreement on common semantic concepts, the sharing of information is possible across all levels of the federation.

C. The experience of IoT-augmented automated driving

The AUTOPILOT project showcases that through a standard-based interworking approach different IoT platforms can run in harmony without causing significant extra costs or overhead. The main benefits of the proposed interworking approach can be summarized as follows.

- Although there are many different and site-specific solutions, automated driving can be successful at the European scale, considering long travels.
- Different pilot sites do not have to adapt their specific devices or software to connect to different IoT platforms.
- Without much effort to adapt, autonomous vehicles can leverage a combination of various IoT services supported by different IoT platforms (e.g., crowd estimation service using FIWARE IoT platform, geofencing service by the Huawei OceanConnect platform).

¹⁵<https://github.com/Aeronbroker>

¹⁶<https://fogflow.readthedocs.io/en/stable/broker.html>

¹⁷<https://sensor-is.org/>

¹⁸<https://www.datex2.eu/>

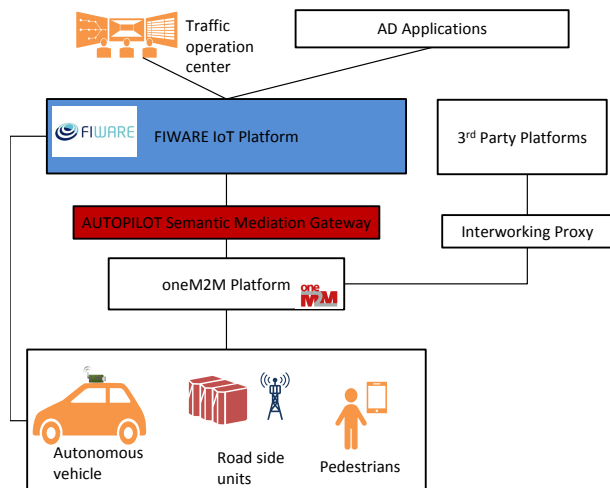


Fig. 4. Vehicle, RSUs, or pedestrians connecting to the FIWARE platform directly or through the oneM2M platform.

- The federation at different layers can be achieved considering multiple pilot sites across Europe.

One drawback of the interworking approach is to guarantee the adaptation of heterogeneous data from a vast number of possible services. To prevent service-specific efforts for conversion between different formats, common and standardized information models are a must. Hence, the AUTOPILOT has a specialized activity group having members from different stakeholders such as IoT architecture developers, pilot site leaders and use case leaders to agree on the common data models.

V. THE ROAD AHEAD

FIWARE has a very lively and active community that is continuously expanding. In this section, we include two key areas where FIWARE has the potential to improve.

A. Semantics: NGSI-LD

NGSI-LD is the evolution of the NGSI Context Interfaces [3] that has originally been standardized by the Open Mobile Alliance (OMA) and further developed in FIWARE. NGSI-LD is standardized as a group specification by the ETSI Industry Specification Group for cross-cutting Context Information Management (ISG CIM). The specification has been published [11] in early 2019 and NGSI-LD is expected to become the new core interface for FIWARE GEs in the course of 2019^{19,20,21}.

The main new feature is that NGSI-LD is now based on JSON-LD, which enables a semantic grounding. The *LD* stands for *linked data* which practically means that all elements are represented as URIs. Thus, the relevant concepts such as entity types can be explicitly defined in an ontology. Thus, the agreement between different applications and sources is on the level of explicitly specified semantics as

provided in an ontology. This enables supporting a level of semantic interoperability on top of an otherwise heterogeneous IoT landscape, which is especially relevant in cross-cutting and large-scale application areas as can be found in smart cities. With the semantic modeling, existing semantic tools for ontology and rule-based reasoning can be used on information retrieved through many NGSI-LD requests.

The new NGSI-LD specification also foresees native support to historical data. It permits applications to directly request complete time series from the data context management together with complex IoT data query capabilities.

B. Privacy: Data Usage Control

Soon, many advancements are expected in the field of distributed privacy and data usage control for securely sharing local domain data spaces. As presented in the previous sections, these features are a serious blocking point for moving from experimental IoT to a real global IoT. In the latter, sensitive data (e.g. healthcare data, industrial IoT data, crowdsourced data) share the same infrastructure with the less sensitive data. Topics such as continuous control on the usage of the data, also after data have been shared, shall be addressed. The collaboration between FIWARE and International Data Spaces Association (IDSA) is a strategic alliance since the requirements and reference architecture of the latter are the inputs for the evolution of the framework in the hands of the former. The initial results are already available [12], showing the feasibility and the compatibility of the visions of FIWARE and IDSA.

VI. ACKNOWLEDGEMENTS

This work has been partially funded by the European Union's Horizon 2020 Programme under Grant Agreement No. 731993 AUTOPILOT (AUTOMated driving Progressed by Internet Of Things), No. 723076 CPaaS.io (City Platform as a Service - integrated and open), and No. 732240 SynchroniCity (Delivering an IoT enabled Digital Single Market for Europe and Beyond). The content of this paper does not reflect the official opinion of the European Union. Responsibility for the information and views expressed therein lies entirely with the authors.

REFERENCES

- [1] OASC, "Annex 1: Minimal interoperability mechanisms (mims)," Jan 2019. [Online]. Available: <https://oascities.org/wp-content/uploads/2019/06/OASC-MIMs.pdf>
- [2] F. Cirillo, D. Straeten, D. Gmez, J. Gato, L. Diez, I. E. Maestro, and R. Akhavan, "Atomic services: sustainable ecosystem of smart city services through pan-european collaboration," in *2019 Global IoT Summit (GIoTS)*, June 2019, pp. 1–7.
- [3] OMA, "NGSI Context Management," Open Mobile Alliance, Tech. Rep. Approved Version 1.0, May 2012. [Online]. Available: http://www.openmobilealliance.org/release/NGSI/V1_0-20120529-A/OA-TS-NGSI_Context_Management-V1_0-20120529-A.pdf
- [4] A. Pauer, L. Nagel, T. Fedkenhauser, Y. Fritzsche-Sterr, and A. Resetko, "Data exchange as a first step towards data economy," PwC, Tech. Rep., March 2018. [Online]. Available: <https://www.pwc.de/en/digitale-transformation/data-exchange-as-a-first-step-towards-data-economy.pdf>
- [5] N. Koshizuka, S. Haller, and K. Sakamura, "Cpaas.io: An eu-japan collaboration on open smart-city platforms," *Computer*, vol. 51, no. 12, pp. 50–58, Dec 2018.

¹⁹<https://github.com/ScorpioBroker/ScorpioBroker>

²⁰<https://djane.io>

²¹<https://github.com/FIWARE/context.Orion-LD>

- [6] K. Sakamura, "Open IoT platform and IoT engine," Jun 2016. [Online]. Available: <https://www.tron.org/wp-content/uploads/2016/06/Open-IoT-Platform-and-IoT-Engine.pdf>
- [7] F. Cirillo, N. Capuano, S. P. Romano, and E. Kovacs, "Liots: League of iot sovereignties. a scalable approach for a transparent privacy-safe federation of secured iot platforms," in *2019 Local Computer Networks (LCN)*, 2019.
- [8] B. Cheng, G. Solmaz, F. Cirillo, E. Kovacs, K. Terasawa, and A. Kitazawa, "FogFlow: Easy programming of IoT services over cloud and edges for smart cities," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 696–707, April 2018.
- [9] F. Cirillo, F.-J. Wu, G. Solmaz, and E. Kovacs, "Embracing the future internet of things," *Sensors*, vol. 19, no. 2, 2019. [Online]. Available: <https://www.mdpi.com/1424-8220/19/2/351>
- [10] E. Kovacs, M. Bauer, J. Kim, J. Yun, F. L. Gall, and M. Zhao, "Standards-based worldwide semantic interoperability for IoT," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 40–46, December 2016.
- [11] ETSI, "Context Information Management (CIM); Application Programming Interface (API)," ETSI ISG CIM, Tech. Rep. ETSI GS CIM 009 V1.1.1, January 2019. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/CIM/001_099/009/01.01.01_60/gs_cim009v010101p.pdf
- [12] Á. Alonso, A. Pozo, J. M. Cantera, F. de la Vega, and J. J. Hierro, "Industrial data space architecture implementation using FIWARE," *Sensors*, vol. 18, no. 7, 2018.