IoT-based Smart Parking System for Sporting Event Management

Jeremy Robert*, Sylvain Kubler*, Ahmed Hefnawy†‡, Chantal Cherifi†, Abdelaziz Bouras‡§, Kary Främling¶

*Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg, Luxembourg
{*firstname.lastname*}@uni.lu
†DISP Lab, Lyon 2 University, Lyon, France
{*firstname.lastname*}@univ-lyon2.fr
‡Ministry of Information and Communications Technology (ictQATAR), Qatar
§DCSE, College of Engineering, Qatar University, Qatar
{*firstname.lastname*}@qu.edu.qa
¶Aalto School of Science and Technology, Finland
{*firstname.lastname*}@aalto.fi

***Abstract*—By connecting devices, people, vehicles and infrastructure everywhere in a city, governments and their partners can improve community wellbeing and other economic and financial aspects (e.g., cost and energy savings). Nonetheless, smart cities are complex ecosystems that comprise many different stakeholders (network operators, managed service providers, logistic centers...) who must work together to provide the best services and unlock the commercial potential of the IoT. This is one of the major challenges that faces today's smart city movement, and more generally the IoT as a whole. Indeed, while new smart connected objects hit the market every day, they mostly feed "vertical silos" (e.g., vertical apps, siloed apps...) that are closed to the rest of the IoT, thus hampering developers to produce new added value across multiple platforms. Within this context, the contribution of this paper is twofold: (i) present the EU vision and ongoing activities to overcome this vertical silos' issue; (ii) introduce recent IoT standards used as part of an ongoing Horizon 2020 IoT project. The implementation of those standards for enhanced sporting event management in a smart city/government context (FIFA World Cup 2022) is developed and presented as a proof-of-concept.**

I. Introduction

New Internet of Things (IoT) applications that leverage ubiquitous connectivity, system interoperability and analytics, are enabling Smart City initiatives all over the world [1]. These new applications introduce tremendous new capabilities such as the ability to connect, manage, and optimize complex sets of disparate information systems, sensors, devices, people and software solutions into a System-of-Systems (SoS) for use in smart cities.

Although the smart city paradigm paves the way for societal and economic opportunities (e.g., to reduce costs for societies, increase the service for the citizens in a number of areas, foster a sustainable economic growth...), they also pose architectural and structural issues that must be addressed for businesses to benefit. One of the most critical obstacles is the *vertical silos'* model that shapes today's IoT, which hampers developers – *due to the lack of interoperability and openness* – to produce new added value across multiple platforms (data is "siloed" in a unique system, cloud, domain, and stays there) [2]. This is all the more true in a smart city context, since a smart city is a complex ecosystem that comprises a wide range of interacting and cooperating actors such as users, software and network providers, financial institutions, logistic centers, *etc.*, that often result in a set of vertical silos (e.g., per domain) difficult to interconnect.

Several organisms and standardization fora understood this critical challenge and started to build up consortia and IoT initiatives to address it. Let us cite, for example, the *Web of Things* initiative at W3C that aims to create open ecosystems based upon open standards, including identification, discovery and interoperation of services across platforms [3]; the *Alliance for Internet of Things Innovation* (AIOTI) launched by the EU with the aim of strengthening links and building new relationships between the different IoT players (industries, SMEs, startups) [4]; the *Open Platform 3.0*^{TM} initiative at The Open Group that aims to rely on open standards that allow enterprises for more easily using technologies in business solutions [5]; the *OneM2M* global standards initiative that involves 8 standards bodies for Machine to Machine (M2M) communications [6]; or still the *IEEE Internet of Things (IoT)* initiative [7]. Although most of those initiatives promote various types of standards and specific technology enablers, they all share the same vision about relying as much as possible on open and interoperable standards to foster open ecosystems and unlock the commercial potential of the IoT.

Within this context, the contribution of this paper is twofold (i) present today's EU vision, and particularly the on-going and future activities carried out in AIOTI, (ii) present recent IoT standards published by The Open Group, which are today used in a recent H2020 Programme to fulfill horizontal interoperability in smart city settings. Sections II and III deal respectively with these two contributions. Section IV presents a first proof-of-concept of the standards implementation in a smart parking context for sporting event management; the conclusion and discussion follow.

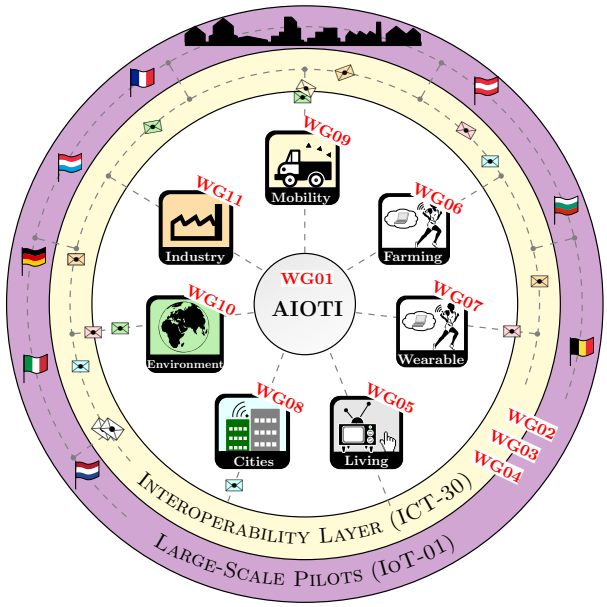
II. European IoT Vision & Initiatives

While in the US, IoT ecosystems are created around big, multinational players such as Apple or Google, the EU's strength is rather in smaller and agile companies. Several past EU initiatives gave rise to a multitude of IoT platforms, in various domains [8], let us cite the IERC cluster in which the IoT-A reference architecture, the OpenIoT cloud platform, BUTLER, *etc.*, were developed, or still the Future Internet-PPP Programme that contributed to the development of the FI-WARE cloud-based infrastructure, which offers a number of general- and specific-purpose function in multiple sectors (farming, manufacturing, mobility, pervasive game...). Despite these efforts, it is a great challenge to turn those initial IoT platforms into economically viable entities and ecosystems. This is the current focus/goal of the EU through new H2020 IoT-related Programmes. This new focus is further discussed in sections II-A and II-B respectively.

A. On-going Initiatives & Technology Transfer

Previous EU projects defined in the FP7 framework and focusing on IoT were developed between 2007-2015. Those projects have led to research in a number of IoT areas, along with the delivery of IoT platforms, architectures and demonstrators, as mentioned above. The FP7 framework constituted the ignition phase of the IoT program approach, the second phase now being the development of open and standardized IoT ecosystems based on these project outcomes, and much more (i.e., enabling citizen communities, SMEs and any public-private organization to join and contribute to the sustainability of the overall ecosystem).

In order to achieve this mission, the EU has launched the AIOTI alliance with the aim of assisting the European Commission in the preparation of future IoT research as well as innovation and standardisation policies in the creation of dynamic EU IoT ecosystems. On a more concrete level, the AIOTI alliance aims both to effectively support the transfer of previous project outcomes to upcoming H2020 projects – *this is achieved through the AIOTI Work Group WG01 (the former IERC cluster), as depicted in Fig. 1* – and to expand those activities towards innovation within and across seven key industry sectors as illustrated through WG05 to WG11 in Fig. 1. Ultimately, AIOTI is going to play an essential role in the designing of IoT Large Scale Pilots that will address all the industry sectors targeted by WG05-WG11 and will be funded by the H2020 IoT-01 Innovation Action Programme (2017-2020; Fig. 1). Nonetheless, in order to provide the necessary interoperability building blocks to enable information to flow easily, safely and efficiently between one or more domains/platforms, another H2020 Research and Innovation (R&I) Programme named ICT30 has been launched (2016-2019) as highlighted in Fig. 1. Several R&I projects and support actions have been funded under the ICT30 programme, which are further detailed in the next section.



WG01: IoT European Research Cluster (IERC)
WG02: Innovation Ecosystems
WG03: IoT Standardisation
WG04: Policy Issues
WG05: Smart Living Environments for Ageing Well
WG06: Smart Framing & Food Security
WG07: Wearables
WG08: Smart Cities
WG09: Smart Mobility
WG10: Smart Environment
WG11: Smart Manufacturing

Fig. 1. European/AIOTI Vision & Initiatives

B. R&I projects developed in the ICT30 Programme

The seven R&I projects developed in the ICT30 Programme aim to improve horizontal interoperability and provide first proofs-of-concept about how existing platforms for connected smart connected objects can easily, safely and reliably be integrated for a multiplicity of novel applications and services. Ultimately, this programme is expected to support convergence and interoperability of IoT standards (*cf.* WG02, WG03, WG04 in Fig. 1), thus opening up IoT ecosystems to developer communities and creative practices.

The ICT-30 portfolio is composed of two support action projects (Be-IoT and Unify-IoT) and seven R&I projects. The two support action projects are strongly complementary, UNIFY-IoT focusing more on scientific aspects, whereas Be-IoT on long-term impact-, community- and ecosystem-building success. In this section, we mainly discusse the seven R&I projects, as listed in TABLE I, in which it is highlighted the key topics that each project is primarily focusing on, namely:

- *Integration of devices:* this topic mainly refers to M2M communications capabilities, where turn-key M2M solutions and components are developed and easy to be deployed. For example, *TagItSmart* will develop innovative optical tags (using a new QR code ink technology) and associated Cloud services for enhanced product tracking throughout its life cycle; *INTER-IoT* and *symbiote* aims to use a common M2M service layer specifications (based on oneM2M and ETSI M2M specifications); and *AGILE* is proposing one gateway access point that should integrate key IoT modules such as modularity, extensibility,

TABLE I
H2020 PROJECTS DEVELOPED IN THE ICT-30 PROGRAMME (2016-2019) WITH THE AIM OF CREATING WIDE IOT PLATFORM ECOSYSTEMS

Project Name	Integration of devices	Creation of platforms	Interoperable APIs	Autonomous reasoning
AGILE – Adoptive gateways for diverse multiple environments	✓	✗	✓	✗
BIG IoT – Bridging the Interoperability Gap of the Internet of Things	✗	✓	✓	✓
bIoTope – Building an IoT oPen innovation ecosystem for connected smart objects	✗	✓	✓	✓
INTER-IoT – Interoperability of heterogenous IoT platforms	✓	✓	✓	✗
symbIoTe – Symbiosis of smart objects across IoT environments	✓	✓	✗	✗
TagItSmart – Smart Tags driven service platform for enabling ecosystems of connected objects	✓	✓	✗	✗
VICINITY – Open virtual neighbourhood network to connect intelligent buildings & smart objects	✗	✓	✓	✗

privacy and development toolkit management;

- *Creation of platforms:* this topic refers to the definition, specification and extension of platforms, either Cloud-based or local (or both), depending on the pilot needs and requirements. For example, *TagItSmart* and *symbiote* will develop Cloud based services (*TagItSmart* will e.g. re-use available FIWARE components); *bIoTope* and *VICINITY* put particular emphasis on Edge nodes (e.g., based on Fog computing and distributed analytics), which is key to improve privacy and data ownership in IoT;
- *Interoperable APIs:* this topic refers to standardized and open APIs that must cope with the IoT peculiarities and requirements, e.g. to support efficient data publication, consumption and composition of heterogeneous information sources and services from across various platforms for Connected Smart Objects. Those APIs must provide the necessary messaging interfaces, along with generic content description models for IoT data representation (e.g., standardized vocabularies...). Each project will investigate, select (or may be develop) such open API solutions, although one of the major challenge will be to agree upon a common API for successful interoperability between the 7 R&I projects;
- *Autonomous reasoning:* this topic refers to context-aware and self-adaptation capabilities of the system/ecosystem. For example, both *BIG IoT* and *bIoTope* will develop, validate and deploy 'context brokers' that are able to discover, predict, validate and supply relevant 'Context(s)' to applications and/or entities requesting it (i.e., offering Context-as-a-Service). Based on relevant and accurate 'context' delivered by such brokers, systems will be able to factly and intuitively react to context changes [9];

The next section presents the Open API standards adopted and promoted in the bIoTope project.

III. OPEN IOT STANDARDS UNDERLYING BIOTOPE

bIoTope takes full advantage of Open API standards developed and officially published by The Open Group, notably the Open Messaging Interface[1] (O-MI) and Open Data Format[2] (O-DF) standards. Those standards emerged out of past EU FP6 and FP7 projects (e.g., PROMISE FP6, LinkedDesign FP7...), where real-life industrial applications required the collection and management of product instance-level information for many domains involving heavy and personal vehicles, household equipment, phone switches, *etc.* [10], [11]. Information such as sensor readings, alarms, assembly, disassembly, shipping event, and other information related to the entire product life cycle needed to be exchanged between products and systems of different organizations. Based on the needs of those real-life applications, and as no existing standards could be identified that would fulfil those requirements without extensive modification or extensions, the partner consortia started the specification of new messaging interfaces [12]. Those specifications have since then been further developed and published by the IoT WG of The Open Group. O-MI provides a generic Open API for any RESTful IoT information system, meaning that in the same way that HTTP can be used for transporting payloads in formats other than HTML, O-MI can be used for transporting payloads in nearly any format. The complementary – *but not compulsory* – O-DF standard is a generic content description model for Objects in the IoT, which can be extended with more specific vocabularies (e.g., using domain-specific ontology vocabularies). Both standards are further described in sections III-A and III-B respectively.

A. O-DF standard specifications

As mentioned above, O-MI can be used for transporting payloads in nearly any format (XML, JSON, CSV...) can also be used. The accompanying standard O-DF partly fulfills the same role in the IoT as HTML does for the Internet, meaning that O-DF is a generic content description model for things in the IoT. O-MI and O-DF are independent entities that reside in the OSI Application layer, where O-MI is specified at the 'communication' level and O-DF at the 'format' level.

O-DF is defined as a simple ontology, specified using XML Schema – *which might currently be the most common text-based payload format due to its flexibility, thus providing more opportunities for complex data structures [9]* – that is generic enough for representing "any" object and information that is needed for information exchange in the IoT. It is intentionally defined in a similar manner as data structures in object-oriented programming. O-DF is structured as a hierarchy with an "Objects" element as its top element, as depicted in Fig. 2, which can contain any number of "Object" sub-elements. "Object" elements can have any number of properties, referred to as InfoItems, as well as "Object" sub-elements. The resulting Object tree can contain any number of levels (*cf.* Fig. 2). Every

[1]https://www2.opengroup.org/ogsys/catalog/C14A
[2]https://www2.opengroup.org/ogsys/catalog/C14B

TABLE II
MAIN MESSAGING INTERFACES SPECIFIED IN THE O-MI STANDARD

Operations	Description
1 - Write	used to send information updates to O-MI nodes.
2 - Read	used for immediate retrieval of information from an O-MI node.
3 - Subscription	Two types of subscriptions can be performed: • *subscription **with** callback address:* the subscribed data is sent to the callback address at the requested interval. Two types of intervals can be defined: ***interval-based*** or ***event-based***; • *subscription **without** callback address:* the data is memorized on the subscribed O-MI node as long as the subscription is valid. The memorized data can be retrieved (i.e., polled) by issuing a new O-MI read request by using the subscription ID.
4 - Cancel	used to cancel a subscription before it expires.

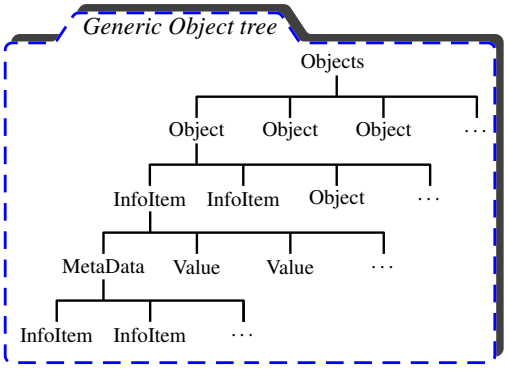


Fig. 2. Open Data Format: generic "Object" tree

Object has a compulsory sub-element called "id" that identifies the Object. The "id" should preferably be globally unique or at least unique for the specific application, domain, or network of the involved organizations. The proof-of-concept developed in section IV provides greater insight into O-DF and facilitates understanding of the information hierarchy.

B. O-MI standard specifications

A defining characteristic of O-MI is that nodes may act both as "servers" and as "clients" and therefore communicate directly with each other or with back-end servers in a peer-to-peer manner. Typical examples of exchanged data are sensor readings, lifecycle events, requests for historical data, notifications, *etc.* One of the fundamental properties of O-MI is that O-MI/O-DF messages are "protocol agnostic" so they can be exchanged using HTTP, SOAP, SMTP, or similar protocols. Four operations are supported, as summarized in TABLE II. Another important feature of O-MI is that O-MI/O-DF messages are "self-contained" in the sense that all the necessary information (e.g., the actions to be performed, the callback address...) to enable the recipient to handle the message is contained in the message itself. Other relevant interfaces are presented in more details in [12] such as the "publication and discovery" mechanisms for data, services and meta-data using the "RESTful" URL-based queries.

IV. PROOF-OF-CONCEPT: IOT-BASED SMART PARKING

This use-case is a hypothetical scenario using IoT-enabled smart parking system for supporting enhanced parking services during the FIFA World Cup 2022 (Qatar). Qatar University – *which is closely working with Qatar city/government to investigate and develop appropriate IoT solutions to effectively manage this sporting event* – expressed an interest in exploring and implementing the O-MI/O-DF standards [13].

In our scenario, each spectator has a unique profile that holds personal information, payment tools, and booked stadium seat numbers. Parking spots are booked in-advance through an online booking system that optimizes the spot allocation (e.g., to enable a car owner to be as close as possible to his/her stadium seat). Upon parking spot allocation, users may enter their car plate number to get fast track access to the stadium, which has several outer gates (see Fig. 3). Fast track gates have sensors to read the car plate numbers and check their eligibility to get in. Another sensor located at each parking spot reads the car plate number to check whether the car is or not at the right spot. If not, a signal as a warning (e.g., light or acoustic) will be issued to the notify the user about this situation (*cf.* Fig. 3). In this proof-of-concept, we consider a simplified parking that is composed of four parking spot areas, respectively denoted by Areas A to D in Fig. 3. Those areas are respectively composed of 3, 6, 3 and 3 parking spots denoted by $\mathrm{P}_{i,j}$ where i is the area index ($i \in \{A..D\}$) and j the corresponding Spot index (e.g., $j \in \{1..6\}$ for $j = B$). Given the parking configuration, several O-MI edge nodes, denoted by O-MI node 1 to 4 in Fig. 3, have been implemented to enable the peer-to-peer publication and discovery of parking-related information in a more or less aggregated form. Section IV-A gives insight into how the overall infrastructure and scenario is supported/achieved based upon O-MI/O-DF., as well as how such information can be used for developing enhanced management services at the stadium level but also at the city level (i.e., opportunities for innovative cross-domain services). Finally, section IV-B focuses on the performance evaluation of the initial release of the O-MI/O-DF reference implementation.

A. Information Publication & Discovery: Smart Parking

The O-MI node denoted by *O-MI node 1* in Fig. 3 is responsible for collecting and publishing parking-related information (e.g., on-site car plate numbers, cars parked at the right spot...). From a physical infrastructure perspective, this O-MI node can be either a centralized node (e.g., a gateway such as a server) or a distributed node (e.g., several gateways distributed over the four Areas). As highlighted by the arrows denoted by ① and ② in Fig. 3, any peer O-MI node (the stadium office

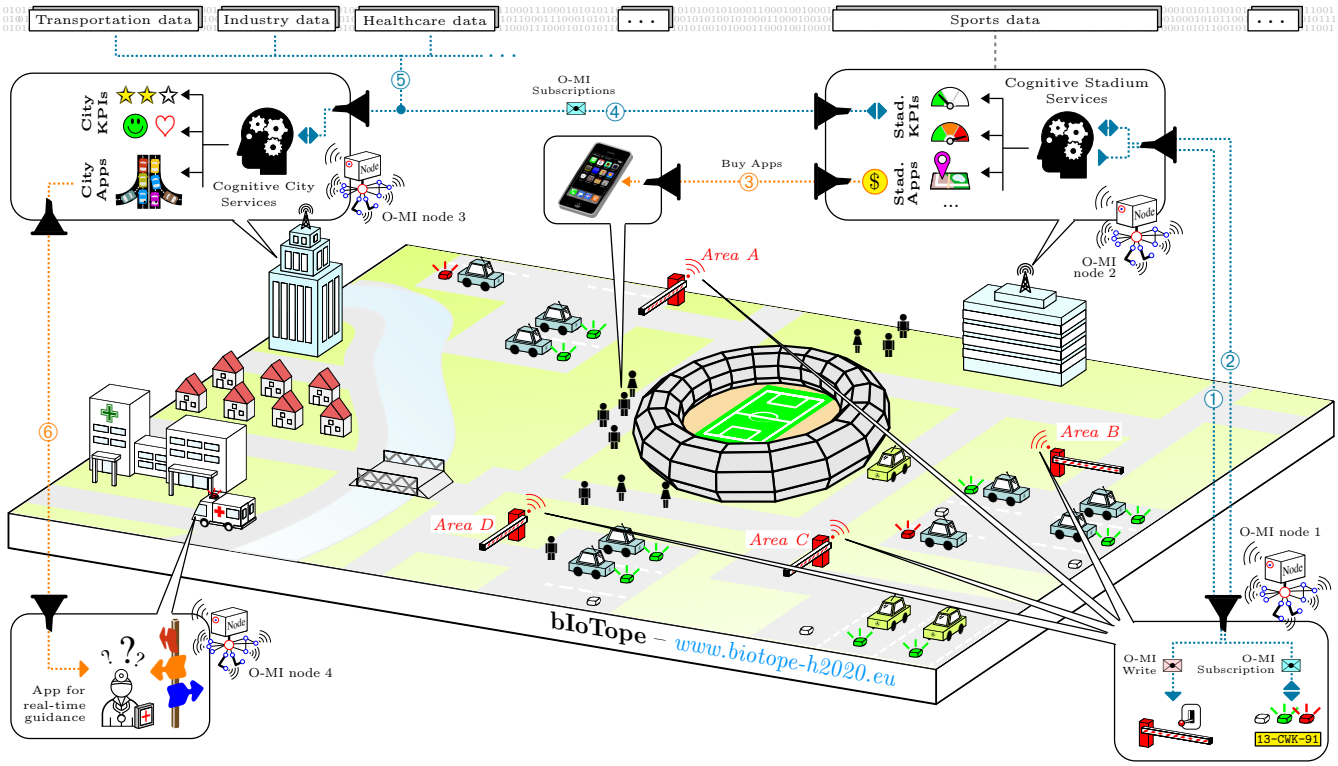


Fig. 3. IoT-based Smart Parking System for Sporting Event Management using the O-MI & O-DF standards

in that case) can discover and access information items that are published by *O-MI node 1* (according to the access rights given at the peer node).

A more detailed view about the network communications between *O-MI node 1* and *O-MI node 2* is provided in Fig. 4 (see arrows denoted by "1" and "2a"), whose associated O-MI/O-DF subscription message ("1") is given in Fig. 5. Rows 1 to 5 detail the message interface where the operation is set to "read" with an interval set to "-1" and a specific callback address (see row 4), meaning (according to the standard specifications) that the subscribed data values must be returned, in an event-based manner, to the stadium office in that case (i.e., O-MI node 2). Rows 6 to 26 detail the message payload, or to be more exact on part of the O-DF hierarchy that is subscribed. The simplified/summarized hierarchy view in Fig. 5) helps to better understand how this information hierarchy has been thought/design for this specific use case. In this example, the stadium office (O-MI node 2) subscribes to *Plate_Number_Readers* information related to Area 1 ($P_{A,1}$ to $P_{A,3}$, Gate1...). Given the message interface setting, the stadium office receives a notification every time an InfoItem value changes. This is illustrated in Fig. 4 through arrow denoted by "2b", where a car whose plate number is `375684` arrived in front of Gate 1; a notification is then automatically pushed to the stadium office (O-MI node 2) that can thus decide to open (or not) the Gate. In the scenario depicted in Fig. 4, the car is authorized to get in the parking and, to this end, an O-MI write request is sent to O-MI node 1 (see arrow denoted by "3" in Fig. 4).

The information collected by the stadium office can be further processed (e.g., using language processing and reasoning algorithms) and turned into (i) new key performance indicators (KPIs) such as the number of free parking spots, car queue length in front of each gate...; or still into (ii) stadium free or fee-based Apps (see arrows denoted by ③) that could potentially inform world cup spectators about how busy a drink and food sale booth is, *etc.* Fig. 3 highlights through the O-MI communication between *O-MI node 2* (city municipality/government) and *O-MI node 3* (stadium avatar) how the city can discover, access and use the stadium KPIs for various purposes, e.g. to combine them with KPIs from other domains (see arrows denoted by ⑤ in Fig. 3) so as to generate new knowledge and services (e.g., to provide indicators of the city health, citizens' well-being, number of free parking spots and real-time traffic state in the whole city...), which may potentially benefit other sectors such as public transportation, industry, energy, and so forth.

Such a cross-domain scenario considering an emergency situation in the stadium is illustrated in Fig. 4, where a notification about the emergency is sent to the city hospital. The hospital system then accesses the city registry that contains the list of O-MI nodes available in the city to get the URL of the corresponding stadium O-MI node (the Khalifa International Stadium in this scenario; see arrows denoted by ④ and ⑤ in Fig. 4). From that moment on, the Hospital O-MI node can use the RESTful URL-based queries to discover

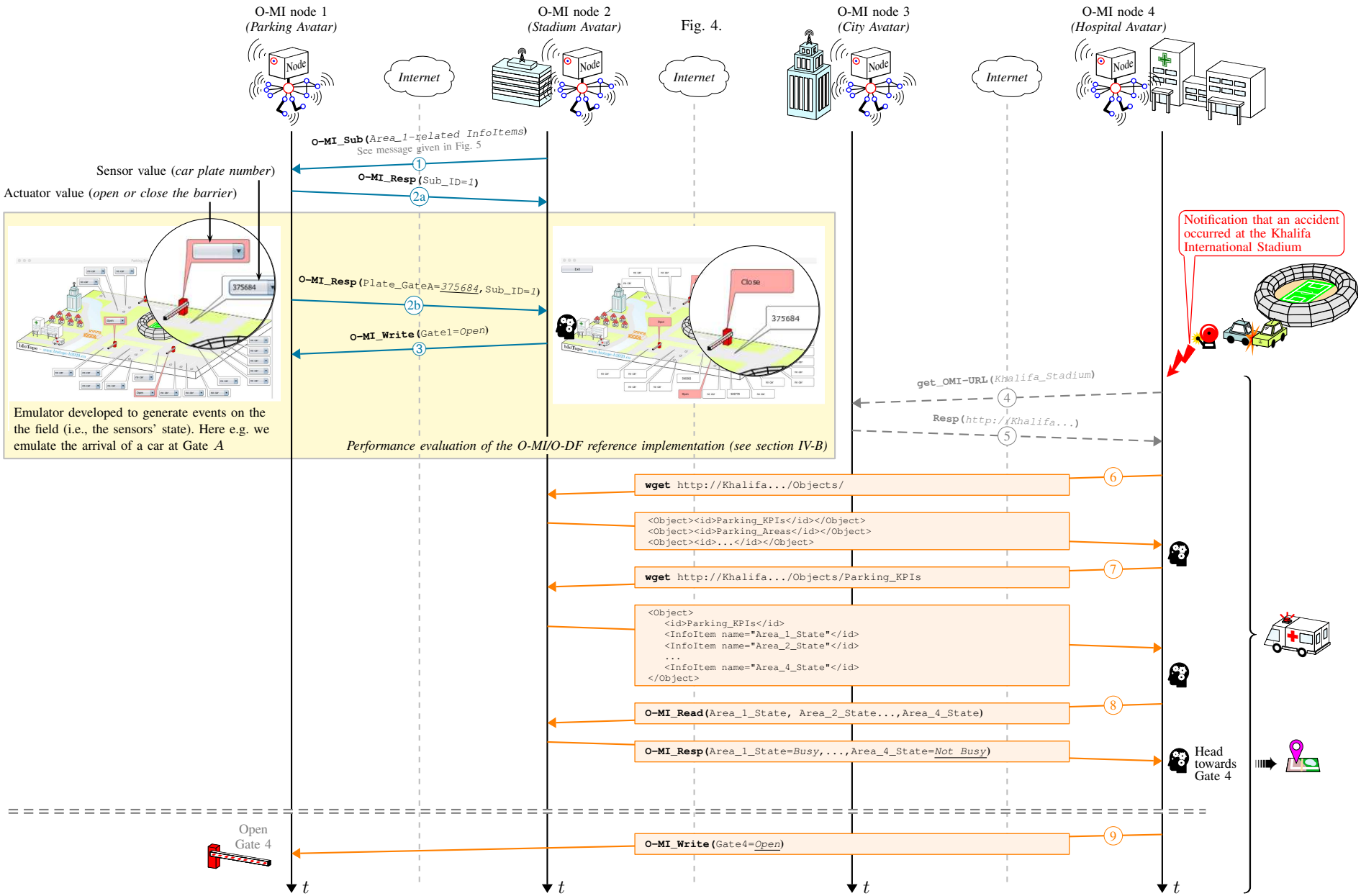


Fig.4. Sequence diagram related to the smart parking system, which combines a smart parking emulator and monitoring tool and the O-MI/O-DF reference implementation

```
1  <?xml version="1.0"?>
2  <omi:omiEnvelope xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
3   xmlns:omi="omi.xsd" version="1.0" ttl="-1">
4    <omi:read msgformat="odf" interval="-1" callback="http://khalifa/">
5      <omi:msg>
6        <Objects xmlns="odf.xsd">
7          <Object>
8            <id>StadiumParking</id>
9            <Object>
10             <id>Area1</id>
11             <Object>
12               <id>Plate_Number_Readers</id>
13               <Object>
14                 <id>Plate_Number_Readers_Parking</id>
15                 <InfoItem name="PA1"/>
16                 <InfoItem name="PA2"/>
17                 <InfoItem name="PA3"/>
18               </Object>
19               <Object>
20                 <id>Plate_Number_Readers_Gates</id>
21                 <InfoItem name="Plate_Number_Reader_G1"/>
22               </Object>
23             </Object>
24           </Object>
25         </Object>
26       </Objects>
27     </omi:msg>
28   </omi:read>
29 </omi:omiEnvelope>
```

O-MI

O-DF

O-MI

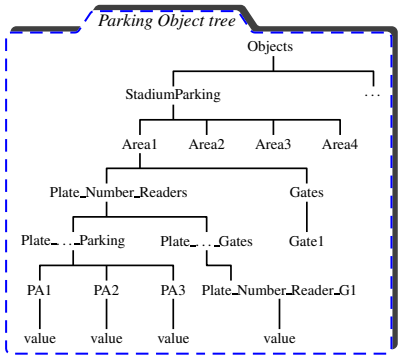


Fig. 5. O-MI / ODF message and associated information hierarchy when subscribing to (some of) Area 1-related information

information published by that node. This is shown in Fig. 4 through the arrows denoted by ⑥ to ⑧ (discovery mechanism invoked using the Unix `wget`). The first `wget` (see arrow ⑥) is composed of the stadium O-MI node's URL to which "Objects" is added so as to access the first level of the information hierarchy, which includes `Parking_KPIs`-related and `Parking_Areas`-related information. The Hospital O-MI node refines the discovery by accessing `Parking_KPIs` information (see arrow ⑦), which gives access to the current state of each parking Area. Given this, the Hospital O-MI node sends an O-MI read request to receive all Areas' state (see arrow ⑧). Based on the response (`Area_4_State`=*Not busy*, while `Area_1_State`=*Busy*...), the decision made by the Hospital avatar (e.g., the ambulance) is to access the emergency location via Area 4. Fig. 4 highlights (see arrow ⑨) that the O-MI RESTful discovery mechanism can further be used to go through the O-DF tree, e.g. up to the opening of the Gate 4's barrier (if a write access is given to the ambulance).

B. Implementation & performance

A first version of the O-MI and O-DF reference implementation has been released[3] and used as foundation of our smart parking system's proof-of-concept. As a complement of this reference implementation, a smart parking emulator and monitoring tool has been developed for both emulating the sensor/actuator events occurring on the field (e.g., the arrival of a car at a specific parking area...) and – *from the client side* – for visualizing the current state of the subscribed/read data. Two screenshots of the parking emulator and monitoring tool are given in Fig. 4. From an implementation perspective, and according to the O-MI and O-DF reference implementation guidelines, it is necessary to develop an internal software agent that periodically pushes the emulated data to an internal database (internal to the reference implementation). This data is then published and made available (depending on the access rights) for any peer O-MI node[4]. In our use-case, a servlet has been developed to (i) listen the O-MI messages received on the callback address, and (ii) parse/extract the contained sensor data values to display them on the monitoring tool (*cf.* Fig. 4).

An interesting performance indicator is the data load produced by the O-MI/O-DF reference implementation to read one or more InfoItems. To this end, a network analyser (Wireshark) has been used to analyze such an aspect when reading information made available in the smart parking scenario. At a more corporate level, 1 to 23 InfoItems are read on an incremental basis (i.e., one request/response including 1 InfotItem and the related Object hierarchy, then one request/response including 2 InfotItems, and so on). This analysis is given in Fig. 6, where data load is composed of:

- a constant part related to the sum of the Ethernet protocol (26 *bytes*), the IP and TCP headers (respectively 20 and 32 *bytes*). The encapsulation is represented by the purple color in Fig. 6. If the O-MI/O-DF message payload needs to be fragmented into a set of frames, the sum of the encapsulation is multiplied by the number of frames so as to obtain the global data load;
- a variable part related to the type of request/response. Indeed, when using the reference implementation (see both graphs on the left of Fig. 6), the application protocol HTTP is constant (462 *bytes* for the request and 173 *bytes* for the response) and the message payload – *encoded with XML in this version of the reference implementation* – is growing depending on the O-DF information hierarchy (the higher the number of levels, the higher the size of the message). When using the

[3]Github: https://github.com/AaltoAsia/O-MI

[4]A web-interface is supported facilitating the use of the O-MI operations (read, write...): http://biotope.sntiotlab.lu:8080/html/webclient/index.html

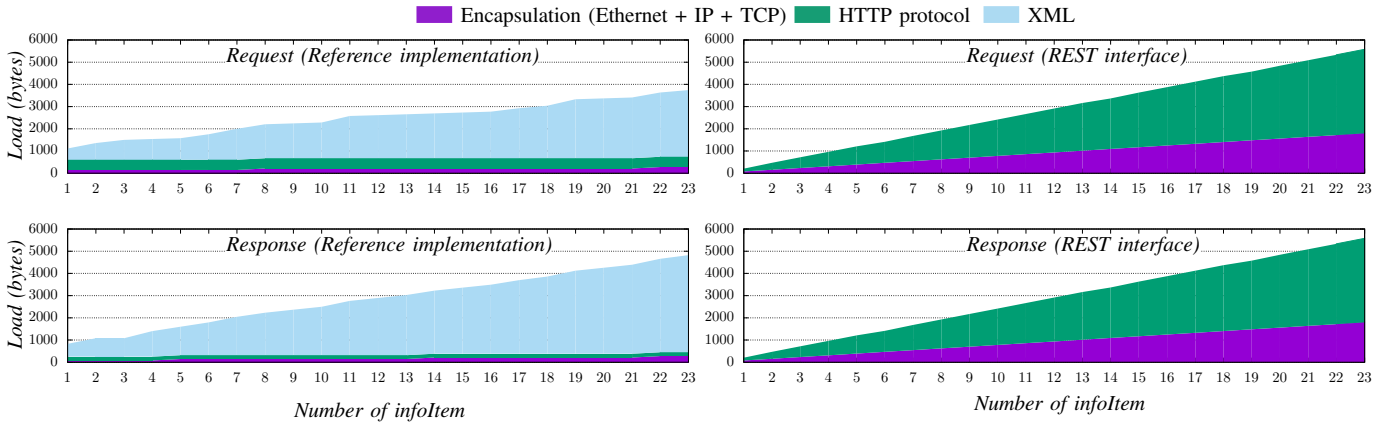


Fig. 6. Load (in bytes) to read InfoItems thanks to the reference implementation or the REST interface

REST interface, the final value is directly embedded in the HTTP protocol as a plain-text, which means that the size of HTTP varies according to the URL (the number of digit). However, it needs to send as many frames as InfoItems when reading more than one InfoItem; this is why the data load is continuously growing in Fig. 6.

In conclusion, even if the data load is non negligible based on the analysis results, it remains acceptable for non real-time or critical time applications. It should also be noted that the initial release of the O-MI/O-DF reference implementation has not been designed to address such critical time applications, but rather to provide first interoperability build blocks for the creation of SoS across sectors and/or in complex environments.

V. Conclusion

This paper provides an overview of today's EU vision on the IoT and related initiatives. Among those initiative, an on-going H2020 project named "bIoTope" uses recent Open API standards named Open Messaging Interface (O-MI) and Open Data Format (O-DF) as foundation of IoT ecosystems. Within this context, this paper briefly discusses the standard specifications, and presents a potential smart parking use case in which those standards can be deployed for enhanced sporting event services and management. The O-MI/O-DF reference implementation is also evaluated in this paper.

It is worth noting that this study does not focus on the payload vocabulary that, obviously, is still a key challenge in IoT applications (e.g., how a node can understand what does an O-DF's Object or InfoItem named "*Area1*" or "*Parking_KPIs*" mean?). To address this issue and improve the interoperability in a smart city context, it is of the utmost importance to comply with generic and/or domain-specific vocabularies (e.g., semantic web technologies and ontologies). Such a topic is gaining traction both in the research and industrial sectors, let us cite e.g. (i) schema.org (W3C) that defines extensive collections of concepts like Thing, Person, Event or Organization; or still (ii) Mobivoc intiative [14] that participates in the development of the Open Mobility Vocabulary. Such vocabulary compliance will be addressed during the bIoTope project.

Acknowledgment

The research leading to this publication is supported by the EU's H2020 Programme for research, technological development and demonstration (grant 688203), as well as the National Research Fund Luxembourg (grant 9095399).

References

[1] P. Neirotti, A. De Marco, A. C. Cagliano, G. Mangano, and F. Scorrano, "Current trends in smart city initiatives: Some stylised facts," *Cities*, vol. 38, pp. 25–36, 2014.

[2] S. Kubler, M.-J. Yoo, C. Cassagnes, K. Främling, D. Kiritsis, and M. Skilton, "Opportunity to leverage information-as-an-asset in the IoT – the road ahead," in *3rd International Conference on Future Internet of Things and Cloud*, Roma, Italy, 2015, pp. 64–71.

[3] D. Raggett, "The Web of Things: Challenges and opportunities," *Computer*, no. 5, pp. 26–32, 2015.

[4] AIOTI, "Alliance for Internet of Things Innovation", *European Commission*, 2015.

[5] The Open Group, "The Nexus of Forces in Action – Business Use-Cases of Open Platform 3.0^{TM}," *Open Platform 3.0 initiative white paper*, 2014.

[6] J. Swetina, G. Lu, P. Jacobs, F. Ennesser, and J. Song, "Toward a standardized common M2M service layer platform: Introduction to OneM2M," *IEEE Wireless Communications*, vol. 21, no. 3, pp. 20–26, 2014.

[7] I. I. Initiative, "Towards a definition of the Internet of Things," *IEEE IoT Initiative white paper*, 2015.

[8] O. Vermesan and P. Friess, *Internet of Things – From Research and Innovation to Market Deployment*. River Publishers, 2014.

[9] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the Internet of Things: A survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 414–454, 2014.

[10] D. Kiritsis, "Closed-loop PLM for intelligent products in the era of the Internet of Things," *Computer-Aided Design*, vol. 43, no. 5, pp. 479–501, 2011.

[11] K. Främling, J. Holmström, J. Loukkola, J. Nyman, and A. Kaustell, "Sustainable PLM through intelligent products," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 2, pp. 789–799, 2013.

[12] K. Främling, S. Kubler, and A. Buda, "Universal messaging standards for the IoT from a lifecycle management perspective," *IEEE Internet of Things Journal*, vol. 1, no. 4, pp. 319–327, 2014.

[13] A. Hefnawy, A. Bouras, and C. Cherifi, "Integration of smart city and lifecycle concepts for enhanced large-scale event management," in *IFIP International Conference on Product Lifecycle Management, Doha (Qatar)*, 2015.

[14] L. Halilaj, I. Grangel-González, G. Coskun, and S. Auer, "Git4voc: Git-based versioning for collaborative vocabulary development," *CoRR Journal*, 2016.