HAL
open science

IoT-based Smart Parking System for Sporting Event Management

Sylvain Kubler, Jérémy Robert, Ahmed Hefnawy, Chantal Cherifi, Abdelaziz Bouras, Kary Främling

To cite this version:

Sylvain Kubler, Jérémy Robert, Ahmed Hefnawy, Chantal Cherifi, Abdelaziz Bouras, et al.. IoT-based Smart Parking System for Sporting Event Management. 13th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MOBIQUITOUS 2016), Nov 2016, Hiroshima, Japan. pp.104-114, 10.1145/2994374.2994390 . hal-01531774

HAL Id: hal-01531774

https://hal.science/hal-01531774v1

Submitted on 1 Jun 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

IoT-based Smart Parking System for Sporting Event Management

Sylvain Kubler
University of Luxembourg
Interdisciplinary Centre for Security, Reliability & Trust
4 rue Alphonse Weicker
L-2721 Luxembourg
sylvain.kubler@uni.lu

Jérémy Robert
University of Luxembourg
Interdisciplinary Centre for Security, Reliability & Trust
4 rue Alphonse Weicker
L-2721 Luxembourg
jeremy.robert@uni.lu

Ahmed Hefnawy
Lyon 2 University,
DISP Lab,
21 avenue Jean Capelle,
F-69621 Villeurbanne
Ahmed.Hefnawy@univ-lyon2.fr

Chantal Cherifi
Lyon 2 University,
DISP Lab,
21 avenue Jean Capelle,
F-69621 Villeurbanne
Chantal.BonnerCherifi@univ-lyon2.fr

Abdelaziz Bouras
Qatar University,
DCSE, College of Engineering,
P.O. Box 2713 Doha-Qatar
abdelaziz.bouras@qu.edu.qa

Kary Främling
Aalto University,
School of Science and Technology,
Helsinki, Finland
kary.framling@aalto.fi

ABSTRACT

By connecting devices, people, vehicles and infrastructures 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 the problem of vertical silos; (ii) introduce recent IoT standards used as part of a recent Horizon 2020 IoT project to address this problem. The implementation of those standards for enhanced sporting event management in a smart city/government context (FIFA World Cup 2022) is developed, presented, and evaluated as a proof-of-concept.

Keywords

Internet of Things; Smart city; Messaging protocols; Standardization; Interoperability; Product Lifecycle Management

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2016 ACM. ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

1. INTRODUCTION

New Internet of Things (IoT) applications that leverage ubiquitous connectivity, system interoperability and analytics, are enabling Smart City initiatives all over the world [13]. 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) [12]. This is all the more true in a smart city environment, as it 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 [15]; 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) [1]; the *Open Platform 3.0*^{TM} at The Open Group that focus more on organization applications and practices [7]; the *OneM2M* global standards initiative that involves 8 standards bodies for Ma-

chine to Machine (M2M) communications [17]; or still the *IEEE Internet of Things (IoT)* initiative [8]. 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 cities. Sections 2 and 3 deal respectively with these two contributions. Section 4 presents a first proof-of-concept of the standards implementation in a smart parking context for sporting event management; conclusion and discussion follow.

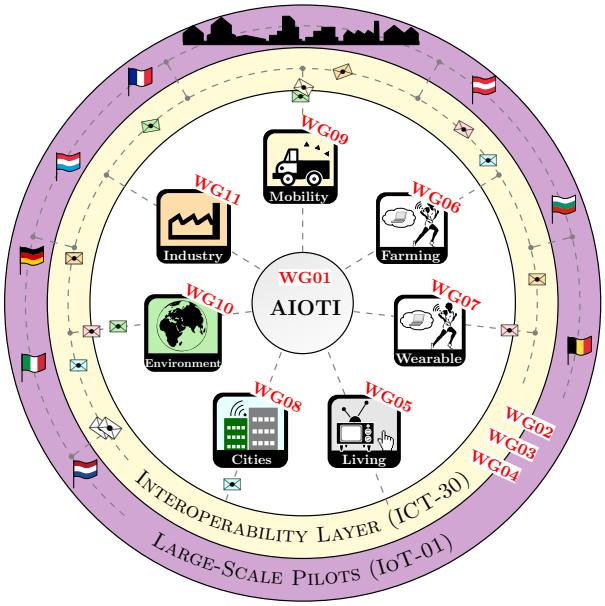
2. EUROPEAN IOT VISION

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 [18], 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 that offers a number of general- and specific-purpose functions in multiple sectors (farming, manufacturing, mobility, pervasive game...). Despite these efforts, it is a key challenge for the EU to turn those initial IoT platforms into economically viable entities and ecosystems. This is the current focus and goal of ongoing and future H2020 IoT-related Programmes. This new focus is discussed in greater detail in sections 2.1 and 2.2 respectively.

2.1 On-going & Future EU Initiatives

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. The FP7 framework constituted the ignition phase of the IoT program approach, the second phase (which started recently) consists building IoT ecosystems to enable citizen communities, SMEs and any public-private organization to join and contribute to the sustainability of the overall ecosystem.

To achieve this mission, the EU has recently 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 Figure 1* – and to expand those activities towards innovation within and across seven key industry sectors, as depicted in Figure 1 (see WG05 to WG11). 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, which will be funded by the H2020 IoT-01 Innovation



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

Figure 1: European/AIOTI Vision & Initiatives

Action Programme 2017-2020 (*cf.* Figure 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 emphasized in Figure 1. Several R&I projects and support actions have been funded under the ICT30 programme, which are further detailed in the next section.

2.2 R&I projects 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 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 Figure 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 reported in Table 1, in which we highlight the key topics that each project is primarily focusing on, namely:

Table 1: H2020 projects developed in the ICT-30 programme (2016-2019)

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	✗	✓	✓	✗

- *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* aim to use a common M2M service layer specifications (based on the ETSI's oneM2M standard); *AGILE* proposes a gateway access point that should integrate key IoT modules such as modularity, extensibility, 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 from across various IoT platforms. 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 challenge to be addressed in ICT30 is to make these APIs interoperable across 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 [14];

The next section provides a brief introduction about the IoT messaging (Open API) standards that lay the foundation of the bIoTope ecosystem and associated city use case pilots.

3. IOT STANDARDS USED IN 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.* [9, 3]. 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 [4]. 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). In resume, 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. Both standard specifications are briefly and respectively presented in sections 3.1 and 3.2.

3.1 O-DF standard specifications

As previously stated, O-MI can be used for transporting payloads in nearly any format (XML, JSON, CSV...). The accompanying O-DF standard 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-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 [14]* – that is generic enough for representing "any" object and information that is needed for information exchange in the IoT. It is

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

Table 2: 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.

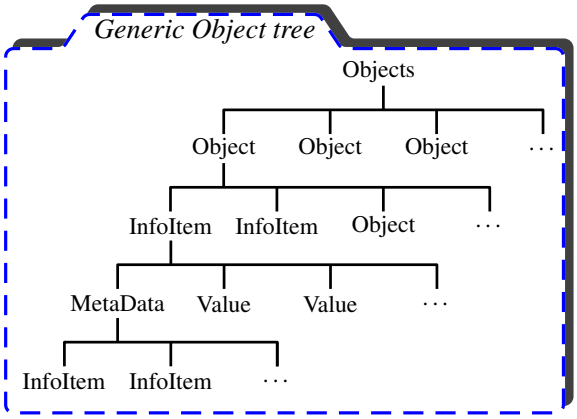


Figure 2: Open Data Format: generic "Object" tree

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 shown in Figure 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.* Figure 2). Every 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 4 will facilitate the understanding of O-DF and associated Object's tree/hierarchy.

3.2 O-MI standard specifications

A defining characteristic of O-MI is that nodes may act both as "servers" and as "clients" and therefore communicate 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 3. Another important feature of O-MI is that messages are "self-contained" in the sense that all the necessary information to enable the recipient to handle the message is contained in the message itself (e.g., actions to be performed, callback address, TTL...). Other relevant interfaces are presented in more details in [4, 10, 11] such as the "publication and discovery" mechanisms for data, services and meta-data using the "RESTful" URL-based queries.

There are several IoT messaging standards comparable with O-MI, and *vice-versa* (e.g., MQTT, AMQP...). Nonetheless, each standard is designed to address specific IoT communication requirements. To support this statement, let us introduce the four IoT communication models defined, in March 2015, by the Internet Architecture Board (IAB) in a guiding architectural document for networking of smart objects [16]. These four models are illustrated in Figure 3 and described hereinafter:

- *Device-To-Device (D2D):* two or more devices directly connect and communicate between one another rather than through an intermediary application server (*cf.* Silos 1, 2 and 3 in Figure 3);
- *Device-To-Gateway (D2G):* the IoT device connects to a local gateway device that may either (i) be connected to a Cloud service provider (*cf.* Silo 1 in Figure 3) or (ii) store and process device-related data at the edge (*cf.* Silo 2);
- *Device-To-Cloud (D2C):* the IoT device connects directly to an Internet Cloud provider to exchange data and services (*cf.* Silo 3 in Figure 3). Frequently, the device and Cloud service are from the same vendor (commonly referred to as "vendor lock-in");
- *Back-End Data-Sharing (S2S):* this model plays a key role in improving horizontal interoperability across sectors and platforms, thus breaking down traditional data silo barriers (*cf.* Figure 3). More concretely, this model shall facilitate *Server-To-Server* (S2S) information exchange based on open and standardized IoT interfaces (open APIs, standardized semantic vocabularies...), but shall also provide provisions for Analytics services, e.g. to filter, aggregate and analyze cross-domain and cross-platform information (*cf.* Figure 3).

Table 3 gives insight into well known IoT messaging standards [2], highlighting for which IoT communication model(s) they have been initially designed for. Our study reports CoAP (developed by IETF), MQTT (developed by IBM), AMQP (developed by OASIS), Data Distribution Service – DDS (developed by the Object Management Group), and XMPP (developed by Cisco). As emphasized in this table, O-MI is primarily aiming at improve horizontal interoperability across vertical silos (S2S). Although this paper is not intended to carry out a technical and thorough comparison between O-MI and the above-mentioned standards, a few striking differences and cornerstones of this standard can nonetheless be pointed out: O-MI uses text-based representations (XML, JSON...) instead of binary formats and can use any of the 'Communication' and 'Transport' level standards as its underlying protocol; O-MI provides a "REST-

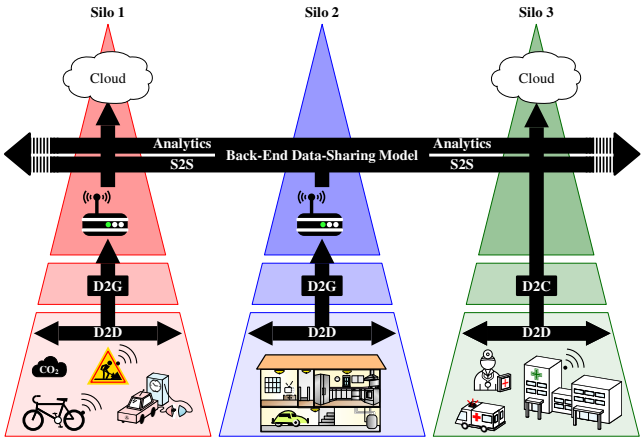


Figure 3: IoT communication model illustration

ful" URL-based query mechanism and, like DDS, is "Data-centric" meaning that middleware can understand the data (e.g., object identity, hierarchy...). This table highlights that three messaging protocols have the necessary provisions for S2S communications (DDS, XMPP, O-MI). Nonetheless, It is necessary to carry out a more thoroughly comparison analysis (i.e., looking at each standard specifications) in order to identify the pros and cons of each protocol, but this is out of scope of this study (see e.g. [**?**]).

Table 3: IoT standards *vs.* communication models

	DDS	MQTT	AMQP	CoAP	XMPP	O-MI
D2D	✓	✓		✓		
D2G	✓	✓	✓	✓		
D2C	✓	✓	✓			
S2S	✓				✓	✓

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

This use-case is a hypothetical scenario using an IoT-enabled smart parking system for enhanced parking services during the FIFA World Cup 2022 (Qatar). The Qatar government[3] expressed an interest in exploring and developing first proofs-of-concept using the O-MI/O-DF standards [6].

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 Figure 4). 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 notify the user about the disturbing situation (*cf.* red lights in Figure 4). 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 Figure 4. 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 $i = B$). Given the parking configuration, several O-MI edge nodes, denoted by O-MI node 1 to 4 in Figure 4, have been implemented to enable the peer-to-peer publication and discovery of parking-related information in a more or less aggregated form. Section 4.1 gives insight into how the overall infrastructure and scenario is 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., cross-domain services). Finally, section 4.2 focuses on the performance evaluation of the initial version of the O-MI/O-DF reference implementation.

[3]Qatar government is closely working with Qatar University to investigate and develop appropriate IoT solutions in view of this sporting event

4.1 Information Publication & Discovery

The O-MI node denoted by *O-MI node 1* in Figure 4 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 one (e.g., a gateway such as a server) or a distributed one (e.g., several gateways distributed over the four Areas). The arrows denoted by ① and ② in Figure 4 illustrate that any peer O-MI node (the stadium office in that case) can discover and access information items that are published by *O-MI node 1* (according to its access rights).

A more detailed view about the network communications between *O-MI node 1* and *O-MI node 2* is provided in Figure 5 (see arrows denoted by "1" and "2a"), whose associated O-MI/O-DF subscription message ("1") is given in Figure 6. 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 (i.e., O-MI node 2). Rows 6 to 26 detail the message payload, or to be more exact, the part of the O-DF hierarchy that is subscribed (the summarized hierarchy view in Figure 6 helps to better understand how this information hierarchy has been thought/designed 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 ($\mathrm{P}_{A,1}$ to $\mathrm{P}_{A,3}$, Gate1...). Given the message interface setting, the stadium office receives a notification every time an InfoItem value changes. This is illustrated in Figure 5 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 decides to open (or not) the Gate. In the scenario depicted in Figure 5, 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 Figure 5).

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)

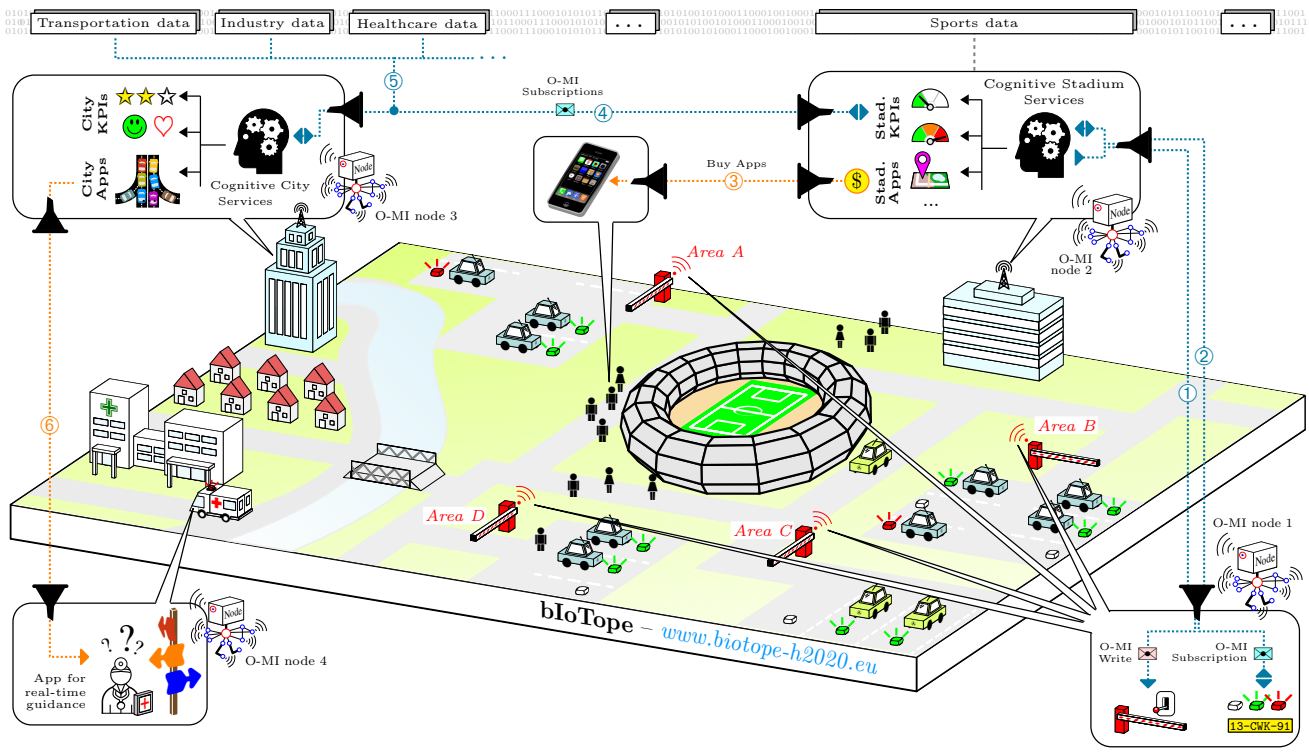


Figure 4: IoT-based Smart Parking System for Sporting Event Management using O-MI & O-DF standards

stadium free or fee-based Apps (see arrows denoted by ③ in Figure 4) that could potentially inform world cup spectators about how busy a drink and food sale booth is. Figure 4 highlights through the O-MI communication between *O-MI node 3* (city municipality/government) and *O-MI node 2* (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 Figure 4) so as to generate new knowledge and services (e.g., to provide indicators on 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, *etc.*

Such a cross-domain scenario considering an emergency situation in the stadium is illustrated in Figure 5, 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 Figure 5). From that moment on, the Hospital O-MI node can use the RESTful URL-based queries to discover information published by that node. This is shown in Figure 5 through 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`- 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 system 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 system is to tell the ambulance to access the emergency location *via* Area 4. Figure 5 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).

4.2 Implementation & performance

A first version of the O-MI and O-DF reference implementation has been released[4] 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 parking. Two screenshots of the parking emulator and monitoring tool are given in Figure 5. 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[5].

Along with this emulator/monitoring tool, we propose to

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

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

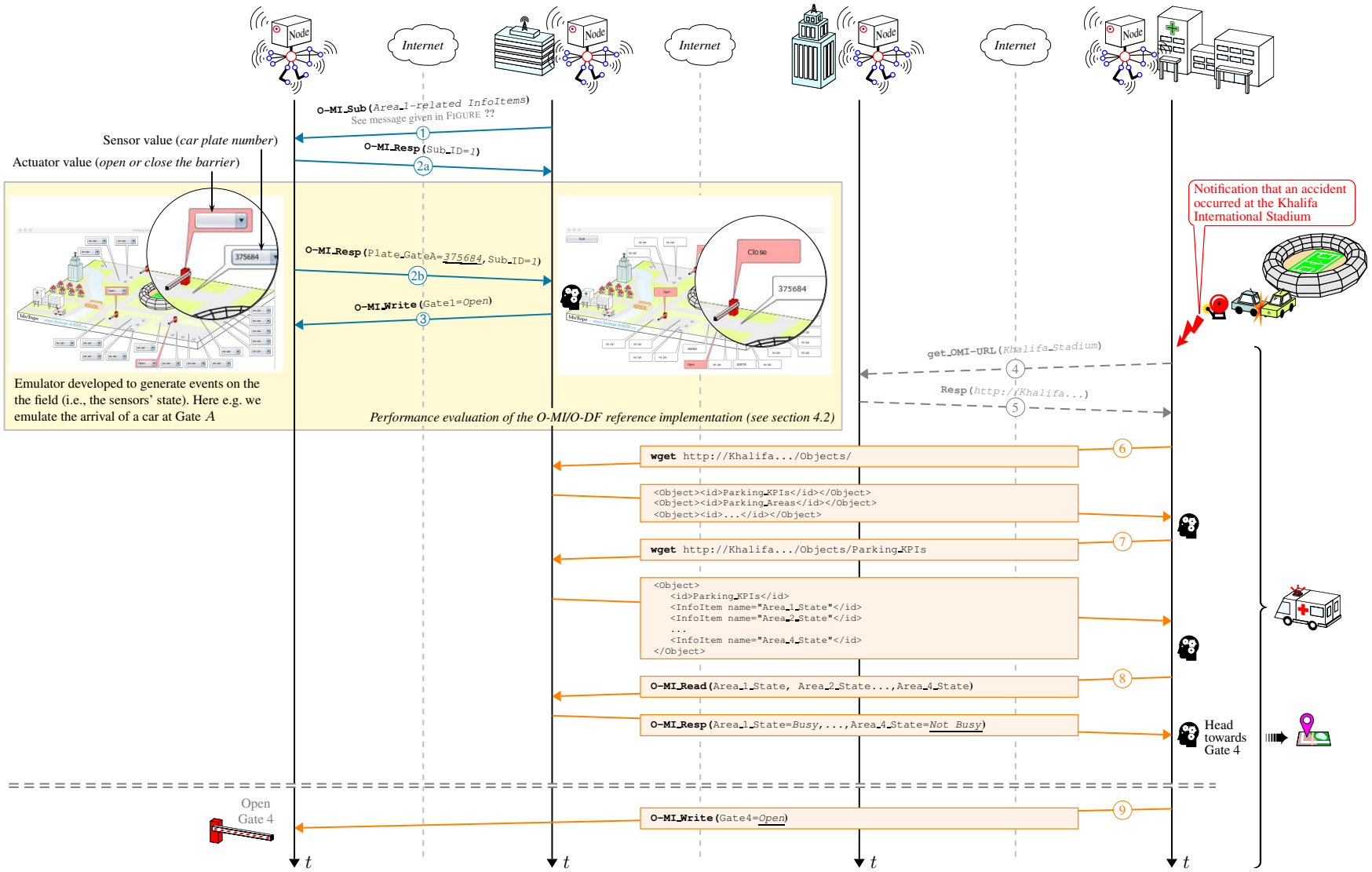


Figure 5: Smart parking scenario combining a parking emulator and monitoring tool based on the O-MI/O-DF reference implementation

```
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

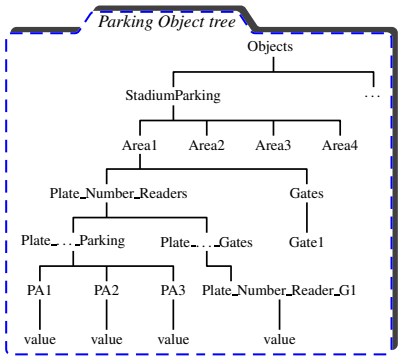


Figure 6: O-MI/O-DF message and associated information hierarchy when subscribing to Area 1-related data

study the performance of both the first version of the O-MI/O-DF reference implementation in terms of 'efficiency'. In this study, efficiency refers to the amount of data (referred to as 'data load' in the following) produced by the reference implementation to access one or more InfoItems and its efficiency ratio. To this end, a network analyser (Wireshark) has been used to analyze the network traffic considering the smart parking use case. At a more concrete level, 1 to 23 InfoItems are read on an incremental basis, i.e. one read request/response including 1 InfoItem (and associated Objects' tree) is performed, then one request/response including 2 InfotItems, and so on. This analysis is given in Figure 7, 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*) for each request/response and their respective acknowledgment (78 *bytes* in the reference implementation). The transient states of the TCP opening and closing operations have not been considered. (encapsulation being represented in white color in the histogram). If the O-MI/O-DF message payload needs to be fragmented into a set of frames (according to the Maximum Transmission Unit equals to 1500 *bytes* in our implementation), 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 *via* a web interface (see Figure 7(a)), the application protocol HTTP is constant (462 *bytes* for the request and 173 *bytes* for the response) and the message payload is growing depending on the O-DF information hierarchy; the higher the number of levels, the higher the size of the message as demonstrated in Figure 7(a).

In the performance evaluation process, O-DF is considered as payload in the reference implementation. However, one may argue that the payload is only the 'useful' data for the application, which can be either the value itself or the semantic (O-DF) structure depending on whether the application needs to understand the surrounding context (i.e., part of the O-DF tree). As a consequence, we decided to study these two considerations that implies using, on the one hand, the reference implementation that conveys the whole or part of the O-DF tree depending on the request (associated performance results are given in Figure 7(a)) and, on the other hand, the REST interface whose payload consists only of the URL in the request[6] and the value in the response (associated performance results are given in Figure 7(b)). As can be observed in both figures, using the REST interface needs to send as many frames as InfoItems when reading more than one InfoItem; this is why the data load is continuously growing in Figure 7(b), and *vice-versa*, using an aggregation strategy as achieved with O-DF results in more efficient data load than sending one frame per infoItem. Looking deeper at both histograms, this aggregation strategy is paying off when embedding more than 7-8 InfoItems in a single message (cf. Figure 7(b)).

The number of frames and their size can impact the reliability and performance of the application depending, among other things, on the environment in which the application is being run. If the environment is noisy with a high bit/frame error rate (e.g., wireless network or in industrial environments), then it may be more sensible to send smaller frames (i.e., to adopt the REST interface strategy) at the expense of the global data load (which increases when reading more than 7-8 InfoItems in this specific case). Indeed, the higher the packet size, the higher the probability that an error occurs, which has a non negligible impact on the efficiency performance due to erroneous frames retransmission. However, the aggregated strategy (making use of O-DF) adds a generic semantic/vocabulary that is key to automate the reasoning in IoT applications (especially when addressing horizontal/S2S interoperability in the IoT), whereas the REST

[6]Embedded in the HTTP protocol as a plain-text, which means that the size of HTTP varies according to the URL (i.e., the number of digit, e.g. in the string `Objects/StadiumParking/Zone1/Gates/Gate1`).

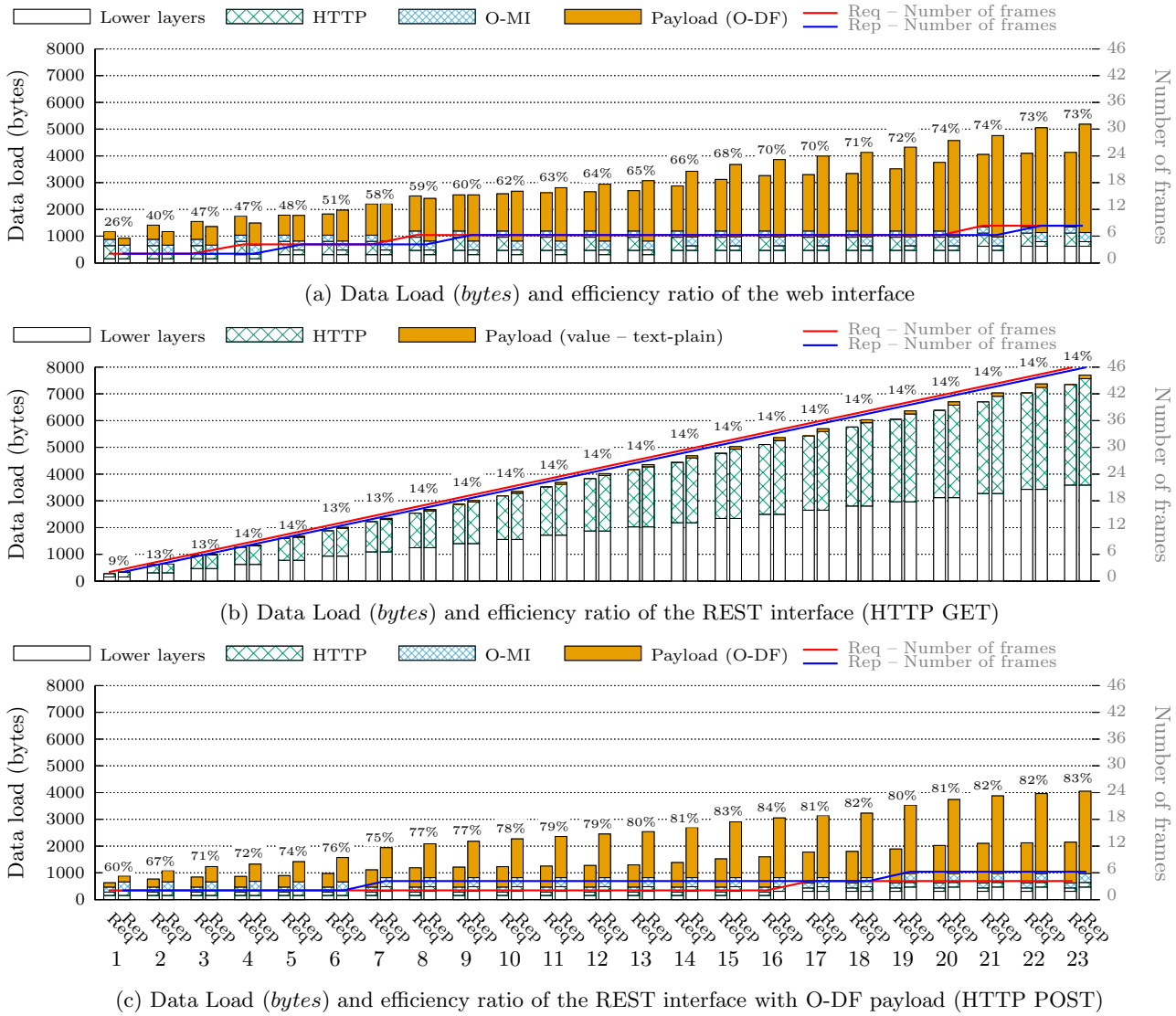


Figure 7: O-MI/O-DF reference implementation analysis with respect to the standard specifications

interface has the advantage of minimizing the load related to the HTTP layer, as evidenced when comparing the first request/response between Figures 7(a) and 7(b). These two advantages could potentially be combined in future reference implementation versions by sending to the REST interface (*via* HTTP POST) an O-DF payload. Furthermore, since REST-based messages are intended to be processed by machines/devices, we could even suggest to optimize the O-DF payload by removing human-readable constraints imposed by the web-interface of the reference implementation (i.e., spaces and carriage return/line feeds). Such an hybrid strategy has been set up, whose performance results are given in Figure 7(c). It can be noted that the size of the frames (and, as a result, the number of frames) and the global data load decrease compared with the web-version (*cf.* Figure 7(a)), and the efficiency ratio increases.

In summary, even if the data load generated by the initial version of the O-MI/O-DF reference implementation is non negligible, it remains acceptable for non real-time or critical time applications. Nonetheless, as explained in section 3.2 and evidenced through IoT-based smart parking use case, O-MI/O-DF standards have not been designed for such time-constrained applications, but rather to improve interoperability across distinct systems and organizations. Regarding the final smart parking infrastructure, the Qatar government has not yet decided on the technologies to be used/deployed on site, but we believe that our findings can help to decide how to use or properly adapt the O-MI/O-DF reference implementation to the final decisions and expectations. For example, if a significant proportion of automated services (without human in the loop) must be developed, we could propose more advanced frameworks that would take full advantage of the REST interface (i.e., both HTTP GET and POST - Figures 7(b)-7(c)), while taking into account the overall environment and selected technologies (e.g., if the network suffers from high packet loss rates, *etc.*). The self-adaptation capabilities of such frameworks could even take into consideration the final O-DF tree for deciding to

switch, when reading a certain number of InfoItems, between the HTTP GET and POST depending on whether or not the aggregation strategy is paying off, as previously discussed.

5. CONCLUSION

The IoT is playing an ever-important role in this new digital landscape, offering us ways to make our world smarter and more interconnected than it has ever been before. Having said that, there are still important challenges ahead that need to be addressed to enable businesses to make the most of the IoT. A crucial challenge is to overcome the "vertical silos" that shape today's IoT (e.g., vendor-lock in, siloed apps), which are closed to the rest of the IoT and hamper developers to produce new added value across multiple platforms. The EU has taken this challenge very seriously by launching several and complementary IoT Programmes. This paper offers an overview of such EU programmes, and particularly the vision underlying these programme initiatives. This paper further focuses one of granted project named "bIoTope" that makes use of recent Open API standards named Open Messaging Interface (O-MI) and Open Data Format (O-DF) to create innovative and federated IoT ecosystems and fulfill S2S communications in the IoT. Within this context, the paper briefly discusses the standard specifications, and presents a smart parking proof-of-concept (in the framework of the FIFA World Cup 2022) for enhanced sporting event management in a smart city context. This proof-of-concept also evaluates the implementation performance of the selected standards (O-MI/O-DF).

The experimental results and findings show that the aggregation strategy (achieved using O-DF) is suitable for maximizing the efficiency ratio when accessing a certain number of information items, even though the data load is non negligible for real-time applications. Furthermore, our study showed that several implementation designs of the standards could be achieved depending on the system environment, adopted technologies, and O-DF structure (which is specific to each use case). Even though this study does not pay much attention to the O-DF payload vocabulary – *which is a challenge on its own (e.g., how a node can understand what does an O-DF's Object or InfoItem named "*Area1*" or "*Parking_KPIs*" mean?)* – future work should be carried out on this matter (e.g., using 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 [5] 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).

6. REFERENCES

[1] AIOTI. Alliance for internet of things innovation (AIOTI), *European Commission*, 2015.

[2] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash. Internet of things: A survey on enabling technologies, protocols and applications. *IEEE Communications Surveys & Tutorials*, DOI: 10.1109/COMST.2015.2444095, 2015.

[3] A. Foster. *Messaging Technologies for the Industrial Internet and the Internet of Things*. PrismTech, 2015.

[4] K. Främling, J. Holmström, J. Loukkola, J. Nyman, and A. Kaustell. Sustainable PLM through intelligent products. *Engineering Applications of Artificial Intelligence*, 26(2):789–799, 2013.

[5] K. Främling, S. Kubler, and A. Buda. Universal messaging standards for the iot from a lifecycle management perspective. *IEEE Internet of Things Journal*, 1(4):319–327, 2014.

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

[7] A. Hefnawy, A. Bouras, and C. Cherifi. Iot for smart city services: Lifecycle approach. In *Proceedings of the International Conference on Internet of things and Cloud Computing*, page 55. ACM, 2016.

[8] https://www2.opengroup.org/ogsys/catalog/W145. The open group, "the nexus of forces in action – business use-cases of open platform 3.0^{TM}.

[9] I. I. Initiative. Towards a definition of the internet of things. *IEEE IoT Initiative white paper*, 2015.

[10] D. Kiritsis. Closed-loop PLM for intelligent products in the era of the Internet of Things. *Computer-Aided Design*, 43(5):479–501, 2011.

[11] S. Kubler, K. Främling, and A. Buda. A standardized approach to deal with firewall and mobility policies in the iot. *Pervasive and Mobile Computing*, 20:100–114, 2015.

[12] S. Kubler, K. Främling, and W. Derigent. P2P data synchronization for product lifecycle management. *Computers in Industry*, 66(0):82–98, 2015.

[13] 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*, pages 64–71, Roma, Italy, 2015.

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

[15] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos. Context aware computing for the internet of things: A survey. *IEEE Communications Surveys & Tutorials*, 16(1):414–454, 2014.

[16] D. Raggett. The web of things: Challenges and opportunities. *Computer*, 48(5):26–32, 2015.

[17] RFC7452. Architectural considerations in smart object networking, 2015.

[18] 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*, 21(3):20–26, 2014.

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