Yacht Single Window: a case for a Vessel-to-Infrastructure Interaction Platform

Pierpaolo Baglietto, Giancarlo Camera, Massimo Maresca
CIPI University of Genoa and Padua
Genoa, Italy
{p.baglietto, g.camera, m.maresca}@cipi.unige.it

Andrea Parodi, Matteo Serratore, Leonardo Roncarolo
M3S SrL
Genoa, Italy
{a.parodi, m.serratore, l.roncarolo}@m3s.it

***Abstract*— One of the most important and promising application of the IoT technologies is in the interaction of "mobile" entities, like cars and trucks, with "static" infrastructures in which they are immersed and from which they depend, like motorways, parking sites, city facilities. This type of application is generally named "vehicle-to-infrastructure" communication. This paper describes the research outcomes of the Yacht Single Window (YSW) project which applied this paradigm to a new and different use case for the IoT: enabling the "vessel-to-infrastructure" communication in order to exploit the IoT technologies for the benefits of the leisure boats and yachts security, safety and improved connection.**

Keywords*— *IoT; Service Platform; e-navigation systems; vehicle-to-infrastructure; OSGI Framework; Virtual Object; Orchestration; Safety Smart Transportation; Smart Emergency Management; Safety and Security at Sea; Connected Vehicles;

I. INTRODUCTION

The area of "vehicle-to-infrastructure interaction/ communication" is widely described as one of the most interesting field of application of the Internet of Things[1]. Many research project[2] pioneered novel technologies and methodologies in order to approach the many aspects and challenges involved in this area as i) devices protocol in disconnected/heterogeneous networks ii) cloud-based platform for devices management iii) device virtualization and abstraction iv) data and devices security v) devices composition and programming.

The research project outcomes we are going to describe in this paper has explored, applied and demonstrated technologies belonging to the domain of the IoT in order to design a platform suitable for enabling the "**vessel-to-infrastructure**" interaction/communication paradigm, applied to the domain of the leisure boats and yachts navigation.

A. The Vessel Object

In the context of the Yachts Single Window project (YSW), the main target has been the enablement of a safe communication and interaction between i) the ship/boat (a vessel, in general) and ii) the infrastructure which normally the vessel interacts with. In particular, the domain of the "leisure boats" has been specifically considereommercial ships, either passenger or cargo, already benefits from many supporting technologies for monitoring, tracking and, in general, communicating. Foremost are Automatic Identification System (AIS) [3], Vessel Traffic Systems (VTS) [4], long range radios, satellite links. On the opposite, small leisure boats, amateur fishing boats and yachts in general lack this kind of support technology although VHS radio are used and, in many countries, mandatory beyond the 6 mile limit. Moreover, the sparse, heterogeneous and for many aspects "unpredictable" nature of this type of vessel makes them a perfect fit for considering them at the level of independent "mobile entities" requiring an interaction channel with their infrastructure.

B. The Infrastructure Object

The term "infrastructure" here is used in a slightly different way with respect to the "vehicle-to-infrastructure" paradigm. In the YSW case, infrastructure is any type of support entity that a vessel can interact with while in navigation or at anchor, like:

- emergency services, like Coastguard and other organization providing Search and Rescue (SAR) operations;
- port mooring services, for finding mooring while navigating towards destination;
- port facilities services, such as boat repair, fuel refilling and others types of 'support' services;
- safety related facilities such as endpoint for reporting anomalies/obstacles found while navigating.

C. The Communication Channel

The typical scenario which has been considered is the one where the leisure boat/yacht owner follows a "coastal" navigation route, which is quite common for this type of vessel. In this scenario, communication generally can take place via 3G/4G network coverage or, in the case of port facilities via WiFi. The approach that has been followed guarantees independence of the "vessel-to-infrastructure"

interaction modality with respect to the type of network available.

D. Applicability of the IoT Approach

The aspects which have been defined above provides a paradigmatic scenario for the application of IoT technologies:

- heterogeneity of the objects which must be "connected", either "vessels" or types of "infrastructure", requires the concept of "Object Virtualization" and abstraction which has been defined for example in [2], in order to de-couple Object Interface and Real Object Implementation.
- highly unpredictable network availability requires the exploitation of IoT-specific protocols like MQTT[5] and COAP[6] which can guarantee data communication with low network overhead
- connection of many different entities belonging to different organizations/domains can be achieved thanks to a third-party service provider operating Cloud-based Service Delivery Platforms[7][8]
- extension with different type of services, such as the ones geared to tourist activities alongside the basic emergency services, requires a Platform providing a Service Creation Environment and a Service Execution Environment which can be open to external service developers, following the paradigm of the Virtual Object Composition for defining new types of services.

In the following sections of the paper we will describe the current state of the art in i) the domain of technologies and ICT support to navigation and ii) the domain of the available IoT solutions. We will then describe in detail the specific scenario that has been considered within the project, the IoT prototype platform which has been developed to support the scenario and the future directions which will be followed for these research activities.

II. State of the Art

A. IoT Platforms enabling Vessel-to-Infrastructure interaction

Available open source and commercial IoT platforms have to handle the automatic and smart communication among platform and devices using either a centralized architecture, where devices and infrastructure communicate only with the platform, or decentralized, where devices communicate both with the platform and with each other. Typically, an IoT platform has to be adaptable to various situations and scenarios. One of the most important characteristics is the ability to connect to heterogeneous devices via common IoT protocols. One of the most adopted IoT protocols is MQTT that allows fast and reliable connections while AMQP, HTTP and CoAP are other widely used protocols.

MQTT is implemented, among others, by big players like CISCO[25], Ericsson[26], IBM IoT[28] and Oracle[33], Evrythng [27] has an efficient message broker that supports translation from a protocol to another. CoAP is the only protocol used by IoT platform sensors oriented like OpenIoT[32] and Zatar[37]. Some platforms need a gateway for device connection (e.g. Ayla[22], LinkSmart[29], Microsoft Azure[30], Wotkit[36]), but they guarantee the possibility to use devices from various vendors.

Amazon's device shadow is an interesting feature that lets platform to communicate with devices [21] even when they are disconnected, sending all the updates once they return online. Platforms' scalability is fundamental to make the platform adaptable to various numbers of connections. It has to guarantee almost the same performances with different numbers of devices connected. Scalability can be achieved using a proper architecture. Almost all platforms analyzed have a cloud based architecture. Some of them have chosen very clever solutions for scalability. 2lemetry[19] uses a hybrid structure implementing a distributed data broker layer with nodes that communicates via P2P. CISCO Fog computing architecture is also a good choice for scalability and to not overload the network. Relying on the CISCO's network infrastructure, it distributes computations at the network's edges without sending all the data to the central platform. Modularity is also another important feature. Different modules can be installed depending on the users/developers' needs. Platforms reach this goal in different ways. Bosch Prosyst [34] uses OSGi to guarantee high modularity. ThingWorx[35] has a marketplace with installable plugins for the platform. IBM bluemix comes with different tools and services that can be chosen by developers. Other platforms implement very efficient services to integrate functionalities from different clouds (AirVantage[20] with Google cloud platform connector, Amazon, ThingWorx). Some of the modules that can be installed are dedicated to efficient data analysis and visualization. Very efficient tools come with most platforms (e.g. Bosch[23], BugLabs[24], Nimbits[31]).

B. e-Navigation systems, vessel-to-vessel communication and vessel tracking systems

Considering that 90% [9] of the freight is carried by ships employing more an estimated of 466,000 officers and 721,000 ratings [10], it is no wonder that the IMO (International Maritime Organization) and IALA (International Association of Lighthouse Authorities) have been promoting safety in many ways[11] [38] one of which is the use of communication and computer based instrumentation. Apart the adoption of Radar, three are the most prominent technologies that have changed or promise to change life at sea: AIS (Automatic Identification System), VTS (Vessel Traffic Services) and what is still in the early phases of conception/experimentation: e-Navigation.

AIS is technology originally developed for collision avoidance and is based on the exchange of messages between ships. The system uses two bands in the VHF range: 161.975 MHz (Channel 87B) and 162.025 MHz (Channel 88B). The channel capacity of approximately 10Kbps. The concept and experimental technology has been developed in the 90's and became mandatory (two receivers and one transmitter) following the 2000 Amendment to Chapter 5 of SOLAS requiring that AIS be fitted aboard all ships subject to the convention and of 300 gross tons (GT) and upwards engaged on international voyages, cargo ships of 500 GT and upwards not engaged on international voyages, and passenger ships irrespective of size built on or after July 1, 2002.

The safety aim is fulfilled by notifying all nearby vessel of

ship position and intention. A position message is only 168 bit long (purged of the radio protocol overhead) and contains (in a packed format, information on position, speed, rate of turn, official ship id, and other parameters to correctly evaluate the above). Messages are sent at intervals dependent on the ship speed (max rate 2 seconds). Messages are not limited to positional information and can span several basic frames to contain other types of information (e.g. weather info, report messages, etc.). A vessel AIS system can also receive queries from land based systems. More recently AIS receivers have been installed on board of space Satellites (S-AIS).

AIS information has also been the main source of information for land monitoring services: Vessel Traffic Services (VTS). These systems merge information from various sources of information (Radars, image sensors, weather-channels, etc.) and are aimed at increasing sea safety by managing the traffic in constrained, dangerous or heavy trafficked areas. The approach is similar to that of air-traffic control where dedicated personnel keep watch of the field its representation on video monitors.

IALA is also supporting the e-Navigation initiative aimed at providing an integrated infrastructure where ships are nodes of an always connected rich information system. The e-Navigation concept recalls the Internet paradigm where information can easily be share. Several problems technical and not technical are present:

1. Confidentiality: not all information can be shared freely. This problem has already emerged in connection with AIS. On routes where pirates operate, captains simply turn off AIS transmitters to avoid being detected (Pirates can wait ships at leisure by accessing http://www.marinetraffic.com/).
2. Network: a worldwide fixed-mobile network with guaranteed delivery and security is a fierce technical challenge.
3. Reservation by ship captains. Shipmasters resent the constant increase in ship electronics especially when they cannot evaluate instruments limitations and errors.

These problems are tackled both methodologically and practically by experimental pilots in ever increasing areas of the World (Denmark, Singapore, some US inland waterways are some examples). These experiments are aimed at overcoming some limitations of existing systems by providing/requiring additional information and ship behavior.

One danger of all these initiative is the birth of a host of technically and incompatible systems. IMO is committed both in the test of significant pilots and standardization.

III. The Yacht Single Window Scenario: a case of Vessel-to-Infrastructure Interaction

The main purpose of YSW (Yachts Single Window) platform is to ensure the safety support of leisure boaters. The users can be tracked and monitored during their voyage. This information is supplied to the sea safety responsible authority.

Therefore boat (static and dynamic) information is not made public as is the case with the AIS. In fact, AIS devices can be intercepted by a very simple system composed of a of a VHF radio and freely available PC program. There are several sites that collect all information exchanged via AIS, the most famous of all is www.marinetraffic.com, where you can view the position, route of vessels and related information (including images).

Th purpose is not to replace AIS, but supply a simpler, cheaper and effective system suitable for small boats but essentially with the same capability of AIS and easy access to additional land based services.

As mentioned in previous sections, the VTS (Vessel Traffic Service) uses the AIS to identify and monitor the moving vessels. For vessels that are not required to have on board the AIS, it uses other systems to locate them (radar or other sensors). By connecting to the YSW platform, monitoring authorities will be in the position to monitor all the traffic in the area of competence not only that of commercial traffic. Thus the YSW platform can be considered as just another sensor data source and provides information akin to that contained in AIS messages plus other information from its registered users databases (vessel characteristics, owner information, the number of people on board, data from onboard sensors e.g. weather information if boat is equipped with specific instruments).

The following list provides an example of messages pertaining the boat's position (the most frequently sent message):

- User ID
- Navigational Status
- Speed over Ground (SOG)
- Course over Ground (COG)
- Latitude
- Longitude
- Timestamp

The following list provides an example of messages related to the information on the boat trip (usually sent only at the beginning):

- User ID
- Origin
- Destination
- Estimated Time of Arrival (ETA)
- Number of Persons
- Number of Children
- Number of Persons CRS (disabled, special needs, etc)
- Timestamp

Given that the problem of privacy and data security is always a very important issue in any field and more so for leisure boaters their crew and guests, YSW encrypts messages using the public key of the receiving server before sending data. Only the server that will receive them will be able to decrypt and make the data available to the competent authority.

To send data to YSW platform all that is necessary is an application on a smartphone/tablet. The device will send positional information at regular intervals. From this app it is also possible to notify details on the trip: such time and place of departure and destination port or number of people on board and also specify whether there are minors or people

with disabilities on board. Lastly the app provides support for emergency situations.

In addition to a virtually zero cost system (now everyone has a tablet or a smartphone), it is possible to upgrade seamlessly the system with a satellite connecttion (as GSM coverage at sea is available near the coast if at all). This upgrade, in addition to allowing the use of a satellite connection, also provides an automatic switch on the channel to be used. In fact, the system integrates a small router, capable of managing the most commonly available interfaces: GSM, Wi-Fi and satellite connection. A small, inexpensive computer unit can also be added (such as a Raspberry) to collect and send boat data.

The router can perform automatic switching between Wi-Fi, GSM and satellite connections in order to use the cheapest link available.

The YSW platform also provides a support fot the Hydrographic Institute of the Navy in Genoa by reporting on possible dangerous objects found by boaters during navigation. This contribution can be made by sending electronic documents via an app which will guide the boat owner in the report completion (e.g. by attaching a picture of the obstacle).

In addition to providing safety and security services for boaters, the YSW platform provides also the opportunity for third-party services, on commercial basis. It consists of contextual information channel to notify users of available commercial services: such as the presence of events/facilities near or at the port of destination.

The introduction of an IoT Platform in the context of the maritime safety scenario will move the solution towards the domain of the Internet of Things and, in particular, towards a system supporting Vessel-to-Infrastructure interaction. The main benefits provide by an IoT Platform application will be:

- *Infrastructure Virtualization*: in the maritime safety doamain, many infrastructure owners/operators can have a role: port authorities, coastguards, other providers of in-navigation safety services, tourist ports management organizations and operators, vessel maintenance providers, and so on. The Virtualization features of an IoT platform will provide a common model, e.g. Port "Virtual Object" that will interact with the Platform in the same way, independent from the specific real infrastructure owner IT systems.

- *Vessel Virtualization*: the introduction of a Vessel "Virtual Object" will allow to accommodate different implementation of onboard devices, preserving the homogeneity of the interaction of the single vessels with the IoT Platform, allowing the application of the same interaction model with every type of vessel, leaving the implementation details to the specific device provider.

- *Network-independent Vessel Connection*: while at sea, different network connection types can be available. Under certain conditions (coastal navigation) the ground-based 3G connection can be exploited. In open sea navigation usually the open-to-air VHF frequencies can be exploited or, in certain cases, satellite links. The project is also likely to capitalize from the new standards 802.11ah and 802.11af. The IoT Platform protocols (MQTT, CoAP, etc) will guarantee a low fingerprint, bandwith saving connection and, at the same time, robust delivery mechanism.

- *Service Creation and Orchestration*: The IoT Platform will provide a composition (Mashup) based programming environment, to enable the creation and execution of services aimed at leisure boaters, exploiting the availability of the many different Virtual Object, and also opening the maritime safety scenario to third-party service providers and developers.

In the following section we will provide the details of the IoT Platform, based on the OSGi Framework, which has been designed to support maritime safety in the Vessel-to-Infrastructure scenario.

IV. An OSGi based IoT platform for Vessel-to-infrastructure interaction

OSGi framework [12] is a good choice to create a modular platform and can be used as a base to build a complex IoT platform.

OSGi is a consortium formed by enterprises and research groups. The target is to create a modular Java environment. OSGi provides different open source releases and reference implementations. Open source groups and enterprises have developed their own OSGi framework implementation. Some remarkable open source examples are Apache Felix [16], Eclipse Equinox [17] and Knoplerfish [18]. One of the most used commercial OSGi implementation versions is PROSYST (Bosch group). OSGi adds modularity to Java with two main components: *bundles* and *services*. Bundles are the modules of an application. Services are shared objects that let bundles communicate to each other. Bundles can publish services to make some of their functions available to other bundles, and Services can then be found by the other bundles and used in a publish-find-bind model.

Using Java with OSGi gives the developer the possibility to create a scalable application. Bundles and services can be installed, uninstalled and updated dynamically without restart and reinstall the whole application (hot deployment).

The architecture of the IoT platform designed upon an OSGi framework is formed by bundles and services. OSGi gives the possibility to add objects dynamically, so that we can developeach Virtual Object as a bundle. In addition, if we want to add features to our platform we can do it without having to stop and reinstall the parts that are not affected by that, you can simply add another bundle and connect properly to the others. The main bundles designed and developed for this platform are

- The Orchestrator Bundle
- The Orchestrator GUI Bundle
- The VORegister Bundle (Virtual Object Register)
- The SCRegisterBundle (Service Composition Register)
- The Virtual Object Bundle

The Orchestrator Bundle is the main module that contains all the functions needed to handle automatic event based orchestration. It means that, once you define and start a

service composition, the orchestrator will react to events invoking actions on the chosen objects as the user has defined in the service composition.

The SCRegister Bundle contains all the service compositions created and saved in the platform with proper data structures readable from the orchestrator during the execution. All this structures are created during the deployment phase and contain the actions to call and the event to wait for during the automatic execution (Fig. 3). Fast access to that structures is guaranteed by the SCRegister service.

The Orchestrator GUI lets users to control the orchestrator and create, deploy, start service compositions.

VORegister is a register of virtual objects available on the platform. The VORegister service lets other bundle to access the data structure of VORegister in order to read and update the virtual objects' list.

Virtual object bundles are developed and designed to properly communicate with different types of real objects, following the paradigm defined in [2]. Each virtual object bundle has two communication interfaces: the first one is dedicated to communication between the bundle and the real objects and it differs from a virtual object type to another. It has to implement proper communication protocols and to handle the right type of data to and from the objects.

The second interface is standardized for all virtual objects bundles and it is between the bundle virtual object and the bundle Orchestrator. We decided to use the event manager designed for the OSGi framework that handles events dispatching using a publish/subscribe model. The event manager has two parts (event handler and event admin). *EventAdmin* is a service implemented inside the OSGi framework. Publisher bundles get the *EventAdmin* service in order to publish events synchronously (*sendEvent* function) or asynchronously (*postEvent* function). *EventAdmin* service [13] dispatches messages to java classes implementing the *EventHandler* interface [14]. The OSGi event manager is based on topics. Each *EventHandler* is registered associating a topic to the class. All OSGi events with that topic will be dispatched to that *EventHandler* implementation.

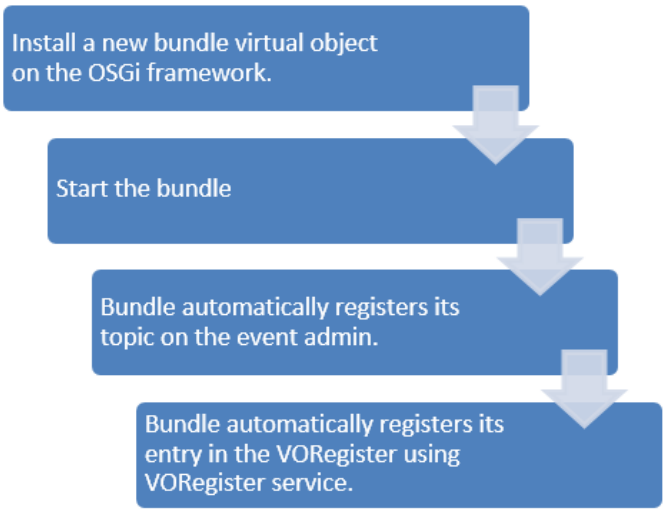

Fig.1. Register a virtual object

When a new virtual object is installed and become available on the platform a new entry in the VO register must be set. At the same time the virtual object registers an *EventHandler* service with the new topic (see Fig. 1).

Orchestrator uses two types of events: invoke action and notify event. Invoke action events are sent by the orchestrator to virtual objects. Notify events are sent by the virtual objects to the orchestrator. The orchestrator sends messages to virtual objects using the *EventAdmin* service and the *postEvent* function. Virtual objects listen to the events published on the topic they have registered with the *EventHandler* class. Inside *EventHandler* implementation there must be the "*handleEvent*" function. We decided to handle the responses to events asynchronously starting a new thread when a new event arrives on the topic.

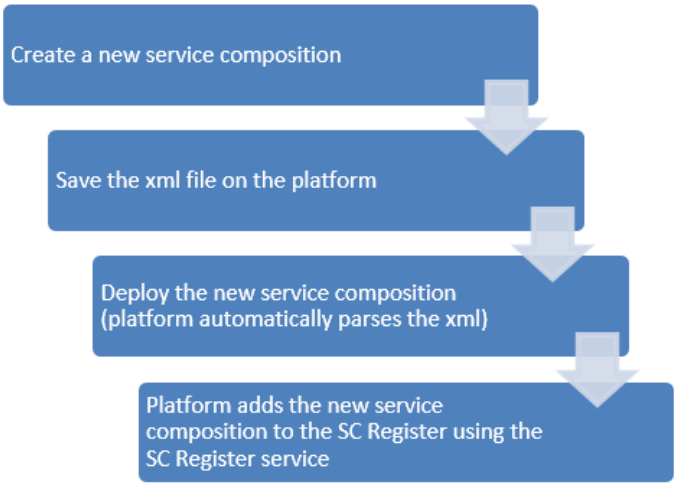

Fig.2. Deploy service composition

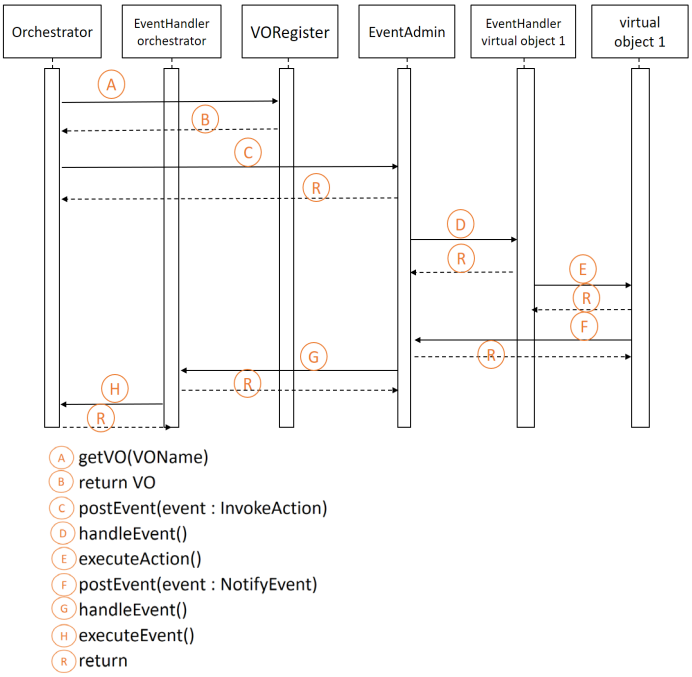

Fig.3. UML diagram execution of a service composition

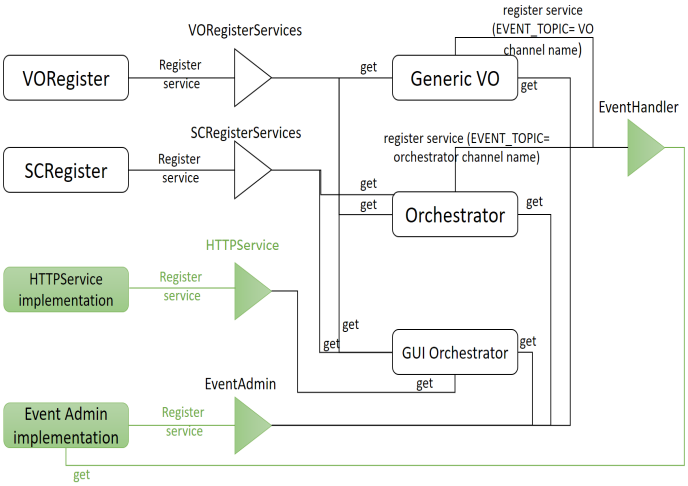

Fig.4. IoT platform architecture

The thread has to handle in the right way the functions to call after the request and, eventually, send a notification to the orchestrator. The events notified to the orchestrator are sent to the orchestrator's topic and are handled in the same way of invoke action events. (Fig. 3).

Fig. 4 shows the IoT service delivery platform's architecture. The bundles are represented by ovals, while services are triangles. Rows starting from the base of the triangle are linked to the bundles registering a service (the bundle that makes some of its functionalities available for other bundles). Rows starting from the peak of the triangle are linked to the bundles that want to use that service. The green objects are bundles and services yet implemented and deployed on the framework chosen (Apache Felix [16] has been used for deploying this platform in the Apache Karaf container [15]).

V. CONCLUSIONS AND FUTURE WORKS

We described how an IoT Platform can enable and support Vessel-to-Infrastructure communication and interaction, providing a seamless way to connect leisure boats and yachts to the many dedicated service providers, for safety and security while at sea but also for delivering touristic and other important services to the boat owners.

While the IoT Platform described in this paper already proved to be beneficial to the purposes of the Yacht Single Window (YSW) project, many aspects still have to be investigated in order to provide a truly operational platform, like the adoption of a Mobile Ad hoc Network infrastructure and the availability of the Platform services in a narrow band connection scenario like for example the VHF channels.

References

[1] IoT European Research Cluster, http://www.internet-of-things-research.eu/

[2] The iCore Project, http://www.iot-icore.eu/

[3] AIS Automatic Identification System, Recommendation ITU-R M.1371-4

[4] VTS, Vessel Traffic Services, http://www.worldvtsguide.org/

[5] MQTT, http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html

[6] RFC7252, The Constrained Application Protocol (CoAP)

[7] R. Minerva, Internet of Things and 5G: IoT communication challenges, M2M Forum 2015, Milan (Italy).

[8] M.Stecca, C.Moiso, M.Fornasa, P.Baglietto, M.Maresca, A Platform for Smart Object Virtualization and Composition , IEEE Internet of Things Journal

[9] International Chamber of Shipping, http://www.ics-shipping.org

[10] Overview of the International Shipping Industry, Shipping and World Trade

[11] International Convention for the Safety of Life at Sea (SOLAS), 1974, http://www.imo.org

[12] OSGi Architecture. (2016). https://www.osgi.org/developer/architecture/

[13] OSGi EventAdmin https://osgi.org/javadoc/r4v42/org/osgi/service/event/EventAdmin.html

[14] OSGi EventHandler https://osgi.org/javadoc/r4v42/org/osgi/service/event/EventHandler.html

[15] Apache Karaf http://karaf.apache.org/

[16] Apache Felixhttp://felix.apache.org/

[17] Eclipse Equinoxhttp://www.eclipse.org/equinox/

[18] Knoplerfishhttp://www.knopflerfish.org/

[19] 2LEMETRY http://2lemetry.com/iot-platform

[20] AirVantage (Sierra Wireless) https://doc.airvantage.net/av/

[21] Amazon AWS IoT http://aws.amazon.com/it/iot/

[22] Ayla https://www.aylanetworks.com/platform/aylas-iot-cloud-fabric

[23] Bosch IoT suite https://www.bosch-si.com/products/bosch-iot-suite/structure-architecture/suite-components.html

[24] BugLabs http://buglabs.net/

[25] Cisco fog computing http://www.cisco.com/web/solutions/trends/iot/overview.html

[26] Ericsson IoT framework http://www.ericsson.com/research-blog/internet-of-things/computational-engine-internet-things/

[27] Evrythng https://evrythng.com/

[28] IBM IoT https://internetofthings.ibmcloud.com/

[29] Linksmart https://linksmart.eu/redmine

[30] Microsoft Azure IoT http://www.microsoft.com/en-us/server-cloud/internet-of-things/azure-iot-suite.aspx

[31] Nimbits http://www.nimbits.com/howto_rest_api.jsp

[32] OpenIoT http://www.openiot.eu/

[33] Oracle IoT https://cloud.oracle.com/iot

[34] Prosyst (BOSCH group) http://www.prosyst.com/startseite/

[35] Thingworx http://www.thingworx.com/iot-platform

[36] Wotkit http://sensetecnic.com/iot-platform/

[37] Zatar http://www.zatar.com/

[38] IMO e-Navigation framework, http://www.imo.org/en/OurWork/Safety/Navigation/Pages/eNavigation.aspx