Web of Things for Connected Vehicles

Soumya Kanti Datta, Rui Pedro Ferreira Da Costa, Christian Bonnet, Jérôme Härri
EURECOM, Biot, France
Emails: {dattas, ferreira, bonnet, haerri}@eurecom.fr

***Abstract*—The current Internet of Things (IoT) ecosystems consist of non-interoperable platform, data silos, heterogeneous devices, technologies and standards. Utilization of the Web standards and their best practices can bring harmony in the fragmented IoT market paving way for Web of Things (WoT). This work advocates for WoT based architecture to integrate the connected vehicles into the IoT ecosystems. We propose a data driven architecture for the same and its prototype experiences.**

Keywords—Connected Vehicle; M2M Gateway; Web of Things.

I. INTRODUCTION

Vehicles are the next frontier of computing and providing consumer centric services. In the recent times both industry and academia are examining the "Vehicle-as-a-Resource" model of computation [1]. In parallel, the Internet of Things (IoT) is fast becoming an enabler for accomplishing many novel technological fronts e.g., intelligent traffic and energy management, highly autonomous vehicle driving and more. The intersection of IoT with these technologies is opening up new areas of research. But at the same time, IoT ecosystem consists of fragmentation which creates interoperability issues. The web standards and best practices can provide an abstraction layer on top of the existing IoT platforms preserving interoperability. Thus, we have considered extending the Web of Things architecture for connected vehicles. Our proposed architecture utilized the best practices of both worlds (Web and IoT) and provides several functionalities needed for enabling the connected vehicles use cases – (i) vehicular data collection using a uniform mechanism, (ii) both IP and non-IP based vehicular communication, (iii) deriving actionable intelligence from raw vehicular sensor data and (iv) notifying vehicles about some decisions. A data centric model for the architecture can also potentially overcome fragmented approaches.

The contributions of the work are – (i) it examines the intersection of WoT and connected vehicles, (ii) a WoT architecture is designed to integrate connected vehicles, (iii) describing the prototyping experience and (iv) outlining the role and involvement of the World Wide Web Consortium (W3C) in WoT and vehicular APIs. The rest of the paper is organized as follows. Section II describes our proposed architecture. Section III outlines prototyping details. Section IV highlights the relevant activities from W3C and Section V concludes the paper.

II. WOT ARCHITECTURE FOR CONNECTED VEHICLES

The proposed architecture is portrayed in Figure 1. The architecture considers the vehicles as M2M devices and the onboard vehicular sensors as M2M endpoints. Together they are part of the data generation subsystem. The overall subsystems as well as their elements are shown in Figure 2.

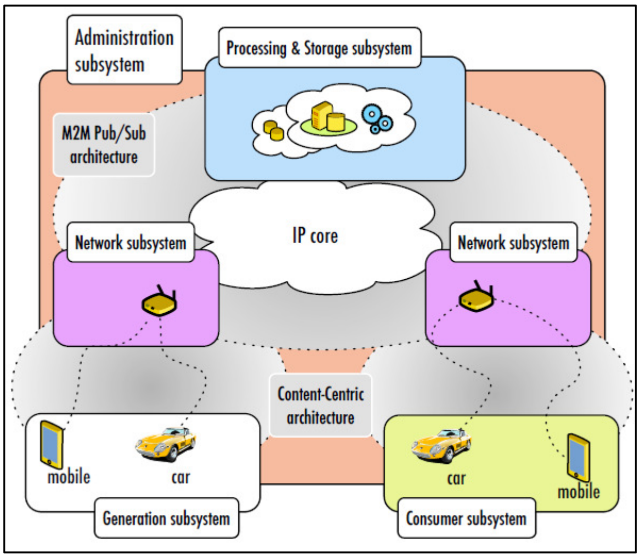

Fig. 1. A functional Web of Things architecture for connected vehicles.

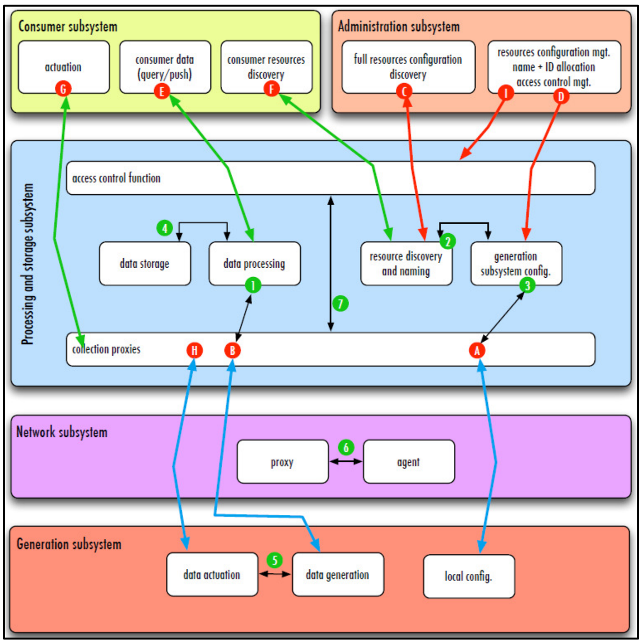

Fig. 2. WoT architecture subsystems and their software elements.

The vehicle On Board Unit (OBU) is equipped with hardware tapped into the vehicle CAN bus and software elements that can extract CAN message signals to obtain the onboard sensor measurements. This corresponds to data generation block. The OBU also keeps a local configuration file needed for managing the registration, discovery procedure and user preferences. The generated data or the local configuration can be communicated to the rest of the ecosystem and it is done through a lightweight and RESTful web server. Several reasons motivate this choice – (i) ease of implementation, maintenance and popularity, (ii) standard interaction model and (iii) direct access through web interactions. The communication medium is provided by the network subsystem. It should support both IP (cellular, WLAN) and non-IP (ITS-G5) based traffic to be flexible and generic enough to adapt to different environments and ecosystems. In case of non-IP based technology or availability of multiple access technologies, there must be additional elements to provide a choice of access technology, and communication protocol selection. IP is the preferred method if available. Otherwise (for non-IP) encapsulation and protocol translation are implemented in the additional elements. The payload with sensor data can be encapsulated inside HTTP or CoAP [4] or MQTT packets This is accomplished at the collection proxy of processing and storage subsystem. This houses further elements to manage the configuration and registration of vehicles, allow discovery from authorized consumers, provide data analytics and storage. The raw data coming from the vehicle is transformed into high level abstraction and actionable intelligence by exploiting semantic web technologies. The processing and storage subsystem is presented to be a generic subsystem in this architecture and in real world it could be analogous to a cloud system and/or an M2M gateway. In either case, it has two web interfaces, namely north and south. Both are a collection of RESTful web services developed following the best practices guidelines of web. The generated actionable intelligence can be pushed to the consumer subsystem or it can discover the same using web services. The administration subsystem is in charge of enforcing the access control and other similar tasks. The software elements for all the subsystems are developed using a simple scripting model in which the local and remote devices (vehicles and smartphones) could be represented using software objects. Another key aspect of the WoT architecture is that it allows creation of a uniform framework for naming and describing the vehicles and the onboard sensors.

The architecture also utilizes core web properties like names (URI for addressing the vehicular sensors as well as M2M gateways, consumer applications), resources and low power protocols for IoT (e.g. CoAP). The main novel aspect of the overall work is to integrate ITS-G5 (a direct short range communication technology) for sensor data exchange between the vehicles and the processing and storage unit. The functionalities of each subsystem is decoupled from those of the other but are exposed using RESTful web services. This allows independent implementation of the software elements across the elements while preserving interoperability through the web interactions. The next section discusses the prototyping experiences of the architecture.

III. Prototyping of the WoT Architecture

We have developed a complete system identified in Figure 1 and 2. A mapping of the system components into real world infrastructure is shown in Figure 3. WoT allows the application logics to be decoupled from the underlying protocols, data formats and encodings, as well as the communication patterns and technologies. We describe in details about the prototyping of the mentioned system in this section and outline the protocols, data formats and encodings used in this work.

The vehicle is equipped with an ITS-G5 enabled OBU that runs a software agent written in C to get the measurements. They along with other sensor attributes are combined to form a metadata using Sensor Markup Language (SenML) [2]. The local configuration of the vehicle and the onboard sensors are represented using CoRE Link Format [3]. Both the metadata and configuration information are encoded using JSON before being communicated using the web services. The local configurations are transmitted first which allows the processing and storage unit to register the vehicle into its local storage. This in turn enables consumer devices to discover and manage them. Once the registration is successful, then the vehicle starts communicating the sensor metadata periodically. Since the ITS-G5 communication may not be able to support IP, software elements for collection proxies are written (in C) and deployed into a Road Side Unit (RSU). These elements are the counterparts of the agents deployed in the OBU. The collection proxies extract the JSON payload from the incoming traffic and encapsulate them into HTTP packets before communicating them to a Raspberry Pi. It offers the data analytics and storage capabilities. The Raspberry Pi and RSU together constitute the processing and storage unit.

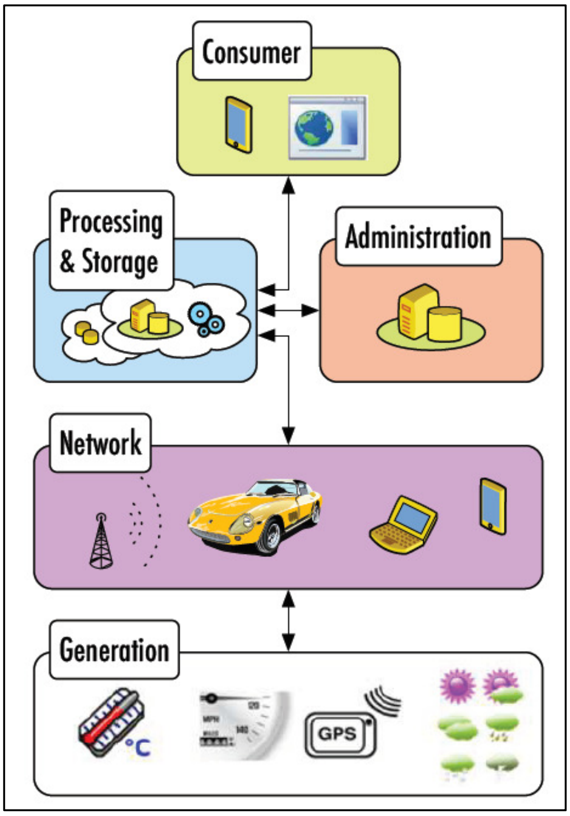

Fig. 3. Mapping of WoT architecture elements into real world components.

As mentioned before, the functionalities of the Raspberry Pi are developed a Python script and lightweight Flask framework is utilized for providing HTTP based web services. The APIs for discovery, management, registration, data processing and storage are exposed using the mentioned web services. The naming and access control are done from the administration subsystem running on a laptop terminal. The consumer smartphones are equipped with an Android application (developed in Java). It supports the consumer related functions. The application logics of the different subsystems are developed using different programming languages but data interoperability is maintained through used of standard protocols like SenML and CoRE Link Format.

IV. Role of W3C in WoT and Vehicular API

W3C has been active in the relevant spaces and we provide some information on Web of Things Interest Group (WoT IG) and Vehicle Information Access API [5] [6].

A. *W3C Web of Things Interest Group*

Following a workshop in 2014, W3C chartered the WoT Interest Group at the beginning of 2015. The group has been since working on IoT/WoT use cases and requirements and a technology landscape survey. The IG is composed of four task forces – (i) Things descriptions, (ii) API and protocols, (iii) Discovery and provisioning and (iv) Security, privacy and resilience. The things description task force is looking into on the data models that things expose to applications and ways to serialize this with JSON. The proposed architecture exploits CoRE Link Format to achieve the same. The API and protocols task force work on identifying web service APIs exposed to the consumer applications and binding of the payloads to web protocols like HTTP and CoAP. The discovery and provisioning task force focuses on understanding the various discovery mechanisms that searches for things - (i) around a client, (ii) on the network, (iii) in a directory and (iv) across peers as well as how things are provisioned in a WoT architecture. Finally, the security, privacy and resilience task force is examining the approaches for securing the IoT. The WoT architecture has all these components embedded into it.

B. *W3C Vehicle Information Access API*

This API is published from W3C Automotive Working Group [7] and exists as a draft as of today [5]. The API aims to enable connectivity through In-Vehicle Infotainment (IVI) system and vehicle data access protocols. The main benefit is that it will allow exposing the data through RESTful web services and vehicular data can be accessed from mobile browsers or applications. The recommended specifications are independent of the data source (e.g. OBD-II, CAN). The properties of vehicles exposed through the API depend on the automotive manufacturers. The specifications also include guidelines for security and access control. This is a future work and will be integrated into the presented WoT architecture.

V. Conclusion

In a nutshell, this work advocates for utilizing the web standards for IoT and the connected vehicle domain would fit well into the resulting WoT. We propose a WoT architecture and examined its intersection with connected vehicles. Our architecture interoperates with ITS-G5 for data exchange between vehicles and processing and storage units. We also outlined prototyping and programming experience. Finally, W3C's role and WoT IG and Automotive WG activities are presented. As for future work, we will integrate the Vehicular Information Access API into the architecture.

Acknowledgment

This work is supported by French research project DataTweet (ANR-13-INFR-0008).

References

[1] Abdelhamid, S.; Hassanein, H.; Takahara, G., "Vehicle as a resource (VaaR)," in Network, IEEE , vol.29, no.1, pp.12-17, Jan.-Feb. 2015.

[2] IETF Draft on Media Types for Sensor Markup Language (SENML), available at https://tools.ietf.org/pdf/draft-jennings-core-senml-04.pdf.

[3] RFC 6690, Constrained RESTful Environments (CoRE) Link Format.

[4] RFC 7252, The Constrained Application Protocol (CoAP).

[5] W3C Working Draft Vehicle Information Access API, https://www.w3.org/TR/vehicle-information-api/, accessed on 24 February 2016.

[6] W3C Web of Things Interest Group Wiki, https://www.w3.org/WoT/IG/wiki/Main_Page.

[7] W3C Automotive Working Group, https://www.w3.org/auto/wg/.