oneM2M Architecture based User Centric IoT Application Development

Soumya Kanti Datta, Amelie Gyrard, Christian Bonnet
Mobile Communications Department
EURECOM
Biot, France
Emails – {dattas, gyrard, bonnet}@eurecom.fr

Karima Boudaoud
Rainbow team
Laboratoire I3S-CNRS/UNSA
Sophia Antipolis, France
Email: karima@polytech.unice.fr

Abstract—This paper presents a user centric and cross-domain IoT application development framework for smart devices. The framework includes automatic discovery of M2M devices, provisioning of sensors and IoT domains, semantic reasoning on sensor data and actuation based on the suggestions. The framework integrates a lightweight version of the Machine-to-Machine Measurement framework and has been adapted for smart devices enabling them to reason on sensor data. This allows us to build cross-domain IoT applications for end users. The issues of interoperability, heterogeneity of domains and multimodality of M2M devices are handled by the semantic web technologies. The framework is integrated into oneM2M architecture and its prototype implementation is also presented. The performance of the mobile application is evaluated in terms of CPU load and power consumption. Finally overall contributions and future directions are summarized.

Keywords—Internet of Things; User-centric application; Semantic Web Technologies; Semantic Reasoning; oneM2M architecture.

I. INTRODUCTION

The Internet of Things (IoT) ecosystem is extending the Internet connection to physical objects like sensors, actuators, RFID tags and other mobile devices. This trend has been creating a range of novel applications and services in various domains like intelligent transportation, building automation, e-health. The ecosystem therefore opens new vistas for a multitude of novel applications and services. A primary goal of these services is to collect data from the physical things and process them to generate contextual information. This allows the applications to be aware of the surrounding situation of the end users and make dynamic and intelligent decisions. But due to multimodal physical things and heterogeneous domains, the tasks of uniform data representation, processing and interpretation are major challenge. The semantic web technologies are useful in such scenarios. Such technologies promote a common way to describe domain knowledge which easily tackles the issues related to heterogeneity and multimodality through interoperable data representation format and various semantic models [1]. The semantic web technologies further facilitates reasoning on raw sensor data to derive high level knowledge of the surroundings. The end users can definitely benefit from such high level abstraction. As a result, several related works have considered deploying semantic tools to create IoT based applications. Another aspect that is essential is rapid adaption of IoT based solutions by end users. To enable that we must create user-centric use case, develop IoT applications for smart devices and deploy such services in transportation systems, building automations, fitness monitoring and more scenarios.

To accomplish such goals, we propose to create user-centric IoT applications running in smartphone and tablets. Such devices are becoming more powerful in terms of hardware, multiple wireless technologies ensuring continuous network operations and increased storage capacity. To empower the users with IoT based solutions, we have developed a mobile application framework which enables the smart devices to dynamically discover physical things, provision sensor(s) and associated domain(s) to receive several cross-domain IoT application scenarios. Depending on the requirements, the user can select a scenario and the smart device will employ semantic web technologies to deduce new information from the sensor data and propose a list of recommendations to the user. The recommendations may include actuation which leads to intelligent control of surroundings based on the selected sensor data. The mobile application framework includes a lightweight version of already developed and deployed Machine-to-Machine Measurement (M3) Framework which performs the conversion of sensor data and reason on that using M3 ontologies, datasets and rules. The novel aspects of the mobile framework are – (i) dynamic discovery of M2M devices, (ii) cross-domain application scenarios, (iii) a lightweight engine to perform the sensor data conversion, reasoning and querying and (iv) actuation based on recommendations. These novel aspects promote the user-centric development approach. The entire M3 framework and its adaption for the smart devices have been developed by following the ETSI M2M and oneM2M architecture recommendations. This is another contribution of this paper which shows how to integrate the overall system with standardized referenced architectures. The contributions of the paper are – (i) porting M3 framework into smart devices enabling them to reason on sensor data, (ii) facilitating dynamic device discovery from smart devices, (iii) developing an approach for creating user centric applications and (iv) integrating the entire framework into oneM2M architecture.

Rest of the paper is structured as follows. Section II studies the related works and points out the limitations. Section III describes the M3 framework, its adaption to smart devices and four operational stages. Section IV presents how the entire system is developed using oneM2M architecture, a use case and performance evaluation of the prototype mobile application. Section V concludes the paper.

II. STATE-OF-THE-ART

We report about the related works on integrating semantics, reasoning engine in constrained mobile devices and ontology-based mobile applications in this section. It identifies the limitations of the current literature and lays the foundation for the proposed user-centric and cross domain IoT application development framework.

A. Integrating semantics in mobile devices

There are several works related to integrating semantics into mobile devices by exploiting user's location and other locally generated information. The authors Le-Phouc, Parreira, Reynolds and Hauswirth design "RDF on the Go", a RDF storage and SPARQL query processor for Android powered mobile devices [2]. They use Jena framework and ARQ Semantic Web Toolkit to execute SPARQL queries and lightweight Berkeley database for storing the RDF data. Their demonstration shows ten nearest cafes or fast food restaurants to the current location obtained from the GPS co-ordinates. This approach reduces network operations and limits the amount of user information exposed to remote server thereby preserving privacy. Similar to this, the paper [3] integrates semantics in mobile devices. The authors identified several issues related to Android content providers which prevent the data stored in the mobile devices to be used as a linked data interface. To overcome the issues, they extend the content provider into RDF content provider framework which includes a set of APIs and content resolver services. The implementation makes use of AndroJena and can find out if a contact of the user is attending the same conference from their calendars. Becker and Bizer query DBpedia, a semantic Wikipedia, through a mobile browser [4]. The application exploits user's location information and a linked data browser to propose nearby tourist attractions. The authors d'Aquin, Nikolov and Motta build the SmartProducts project[1], compliant with W3C SSN ontology [5] and design ontologies related to food and recipes. They enable mobile applications to expose data though a SPARQL endpoint on Android devices [6]. They integrate Sesame triple store in Android to store data in the SDCard, and the iJetty server to be compatible with Android.

AndroJena[2] provides a reasoning engine, SPARQL queries with ARQoid[3] which is a Jena's ARQ SPARQL query engine for Android platforms, ARQoid and TDBoid are still under development. μJena[4] is referenced by Apache Jena and provides a way to load and manipulate RDF data on a device, a reasoning engine optimized for resource-constrained devices. However, SPARQL queries and SWRL rules are not supported. MobileRDF[5] is a lightweight RDF API. Otsopack[6] [7] is a triple store for devices with limited computational resources. μOR [8] is an ontological reasoner optimized for resource constrained devices. iMoco integrates a triple store for the iPhone platform.

B. Reasoning engine for mobile devices

Delta-Reasoner [9] is a reasoning engine developed for constrained mobile devices. The engine is embedded into context-aware applications which can interpret the current situation of the mobile device user based on different sensor measurements. The sensors considered may include internal sensors (e.g. GPS), external sensors (e.g. indoor location) and pseudo sensors. There is no implementation of this tool available online for reusing purpose. Another reasoner called ELK is designed for lightweight OWL EL in [10]. The authors describe the algorithm which supports high performance reasoning in mobile devices and demonstrated it on a Google Nexus 4 running Android 4.2.

C. Ontology-based mobile applications

Ruta et al. [11] propose iDriveSafe 2.0, an ontology-based application on mobile phones in transportation domain. The primary goal of the application is to display vehicle health (emission, fuel consumption, gear level). It can also suggest use of safety devices according to the weather conditions (e.g., switch on the fog lamp if the weather is foggy) and detect the driver's state (careful, aggressive, tired). An ontology-based mobile application for healthcare applications is discussed in [12]. The work presents an in-depth discussion on semantic interoperability and uses the Shimmer sensor for acceleration data and the Philips DTI-2 sensor for skin conductance and acceleration measurements. But the resulting application developed just displays data on the mobile phones, no rules or suggestions are provided. Vincent et al. [13, 14, 15] design an ontology-based firewall to ensure privacy protection for smartphones. They propose two ontologies: the former to represent the Semantic Web Rule Language (SWRL)[7] privacy policies, inspired by the SOUPA framework, and the latter the digital identity on smartphones using well-known ontologies FOAF [8]and VCard[9]. Chien et al. [16] design tourism ontology to provide a museum/exhibit-guidance system. Skillen et al. [17] propose an ontology-based Android application to address the problem of the outdoor mobility of an elderly person such as shopping trip reminders or context-aware guidance.

D. Limitations

Our assessment reveals following limitations of the state-of-the-art:

- The reasoning engines and described semantic algorithms in a mobile application are largely based on the internal sensors of the mobile. Majority of the

---

[1] http://projects.kmi.open.ac.uk/smartproducts/ontology.html

[2] http://code.google.com/p/androjena/

[3] http://code.google.com/p/androjena/wiki/ARQoid

[4] http://poseidon.ws.dei.polimi.it/ca/?page_id=59

[5] http://www.hedenus.de/rdf/

[6] http://code.google.com/p/otsopack/

[7] http://www.w3.org/Submission/SWRL/

[8] http://xmlns.com/foaf/spec/

[9] http://www.w3.org/TR/vcard-rdf/

implementations do not consider external sensors belonging to different domains. There is no dynamic discovery of external sensors. This limits the scope of the application.

- Some works extends Android content framework for their mobile application. We proposed a method where such extension is not necessary at all.
- Current initiatives are largely focused on domain-specific scenarios and do not combine several heterogeneous IoT domains. Thus the use cases of the resulting applications are limited.
- The presented frameworks are not developed using a standard M2M IoT architecture.
- Current literature lacks a common description for sensors, measurements, units, domain names as well as domain knowledge. This raises interoperability issues. Also Standard Development Organizations (SDO) like ETSI do not explicitly describe the semantic computing components in their architecture [18].

III. ADAPTING M3 FRAMEWORK FOR SMART DEVICES

The Machine-to-Machine Measurement (M3) framework germinated to solve the above limitations and extend the capabilities of IoT applications by developing cross-domain and user-centric use cases. The architecture of M3 is given below [18].

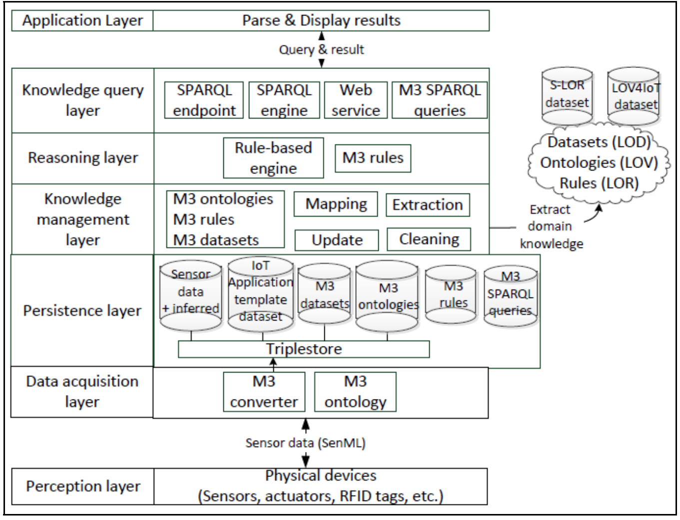

Fig. 1. Architecture of M3 framework

The data acquisition layer queries the physical devices to obtain the sensor metadata structure in Sensor Markup Language (SenML) format [22]. The metadata is then converted to RDF in a uniform way using M3 ontology. It is an extension to W3C SSN ontology and the "Observation Value" concept is extensively used to provide a basis for reasoning on sensor metadata. The persistence layer acts as a storage unit of M3 ontologies, datasets, rules, sparql queries. There is a triple store to house the semantic sensor data. This layer can also generate IoT application templates (collection of M3 ontologies, datasets, rules and generic sparql query) based on type of sensor and the associated domain name. These two can be easily obtained from the sensor metadata. The knowledge management layer extracts domain specific knowledge from existing rules, datasets and ontologies and updates them into M3 rules, M3 datasets and M3 ontologies maintaining a uniform format to maintain interoperability. This layer also combines domain-specific knowledge to create cross-domain scenarios. The reasoning layer deduces high-level knowledge from the sensor data and domain information by using reasoning engines and M3 rules extracted from Sensor-based Linked Open Rules (S-LOR). The knowledge query layer executes a generic sparql query on the deduced knowledge and the application layer parses the result of the query and updates that to the user interface. Currently a prototype of M3 framework is running on a cloud infrastructure and is available [18]. This stores all the templates needed to build various kinds of applications for IoT. Apache Jena Framework [28] is used to develop the M3 and as a result the implementation is quite complex in nature.

We adapt a lightweight version of the M3 into smart devices due to various reasons. Firstly, the Jena Framework can not be directly integrated into smart devices. Secondly, the requirements for the smart devices are different where only one application template is required and can be easily downloaded from the cloud. The smart devices need not have the entire set of IoT application templates. The lightweight M3 is shown in Fig.2 and empowers such devices to semantically annotate sensor data, deduce high-level abstraction of the data and propose a list of suggestions to the end users. The overall application process utilizing the adaption is divided into four phases as described in subsequent sections.

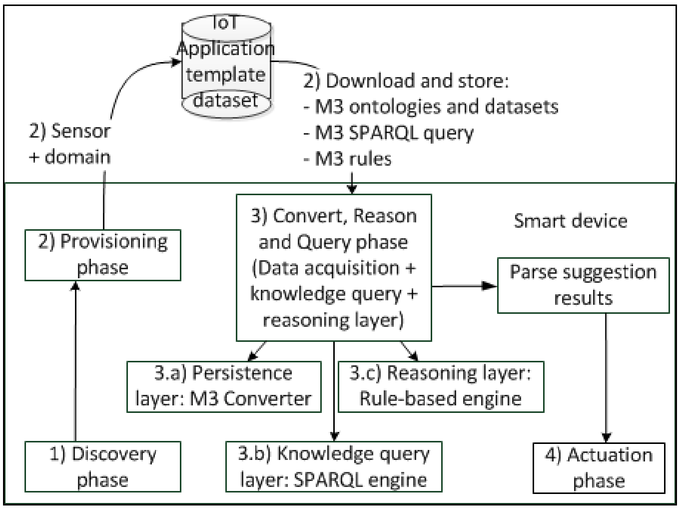

Fig. 2. The M3 framework adapted to the smart devices

A. Discovery Phase

This is the first phase of the application where it discovers the M2M devices and endpoints belonging to the user. The operational flow of discovery phase is portrayed in Figure 3.

It is assumed that the M2M devices and endpoints are already attached to an M2M gateway. The configurations of such devices are stored in the local storage of the gateway. When the application is launched, it queries the discovery web service of the gateway which is located at a well-known entry point at the gateway. It internally retrieves the list of M2M devices, endpoints and associated domains based on the access rights the user has. The list is communicated to the mobile application running in the Android devices. The application displays the information in the user interface. The mobile application enables learning about the sensors and associated domains. This makes the application generic enough to be used in various scenarios. This is one of the novel features of the proposed framework.

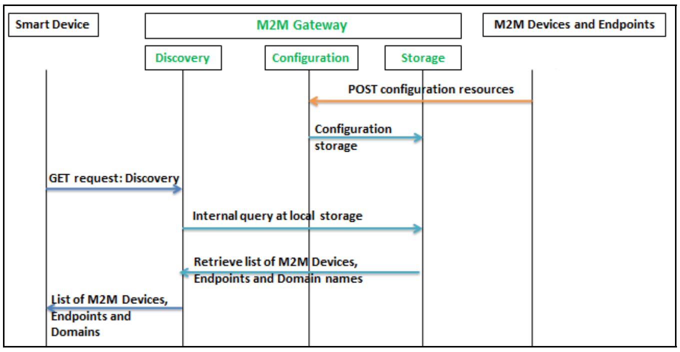

Fig. 3. Operational flow of discovery phase

B. Provisioning phase

After the discovery phase, the provisioning phase begins. It is illustrated in Figure 4. The user selects a sensor (e.g. light sensor) and an associated domain (e.g. weather). The application then queries a web service (searchTemplate) of the persistence layer of the M3 framework with the sensor and domain information. The layer internally retrieves (from IoT application template dataset) a list of previously defined cross-domain scenarios involving the selected sensor and domain. The application receives and presents the list to the user. For example, based on light sensor and weather domain, the M3 framework will proposed four cross-domain scenarios – (i) Weather, Luminosity and Emotion, (ii) Weather, Tourism and Clothes, (iii) Weather, Tourism and Activities and (iv) Weather, Transportation and Safety Device. Each scenario accomplishes a different goal for the user. Weather, Tourism and Activities scenario is useful when a user is in vacation as the application in that case will propose activities based on outside weather. Depending on the requirement, the user selects one scenario and the mobile application queries another web service (generateTemplate) in the M3 framework to download the related application template into local storage. Each application template contains M3 ontologies for the domains, M3 datasets and M3 rules. The final web service query (to getSparqlQuery) is required to obtain a generic SPAQRL query. For a given combination of the sensor type and domain name, the templates are locally cached which reduces the network operations.

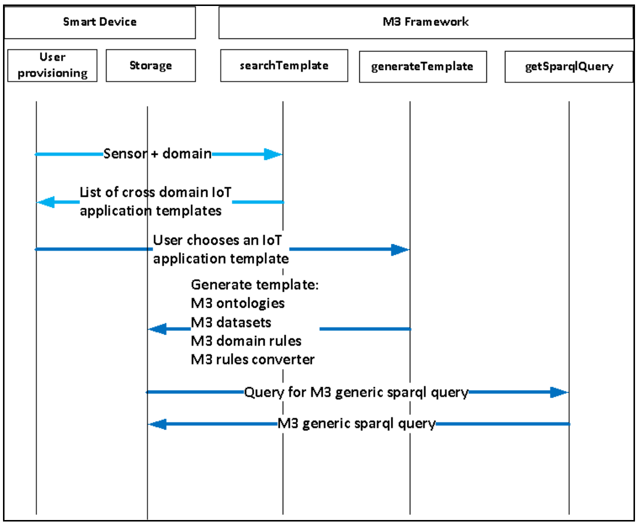

Fig. 4. Operational flow of provisioning phase

C. Convert, reason and query phase

The above template contains all the domain knowledge necessary to interpret sensor data and combine the inferred data with cross-domain knowledge. After, the provisioning phase, the M3 converter queries the M2M gateway to get the SenML sensor metadata. The M3 converter then semantically annotates them with Resource Description Framework (RDF) using "M3 converter rules" so that converted RDF sensor data is expressed in uniform way. This step is necessary as the real sensor data can come from any M2M gateway which may or may not follow uniform nomenclature. To infer additional knowledge, the application uses the M3 reasoner again with "M3 domain rules". This generates a high level abstraction (called deduction) from the sensor measurement type, measurement value and associated domain name. The inferred data are updated in the storage. Finally, the application loads M3 cross-domain knowledge and inferred data to execute the SPARQL query which gives one or more "suggestions". These deduction and suggestion(s) are reported to the user at the user interface or using notification at the notification bar. These above steps are presented in Fig. 5 and relate to the functionalities of the reasoning layer, knowledge query layer and application layer of the M3 framework. A reasoning engine is integrated into the proposed framework to achieve the operations of the phase [29].

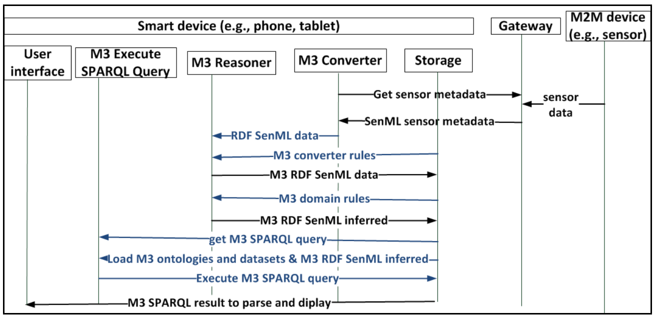

Fig. 5. Operational flow of convert, reason and query phase

D. Actuation phase

The suggestions as proposed in the previous phase often need actuation based on the domain(s) of interest. For instance, in Weather, Transport and Safety Device scenario, if the application deduces that there is heavy rain, it will propose to activate wiper, rear wiper and fog lamp. The user can select an actuator and request a change in its state. Such actuation could also be automatic in the sense where the application itself sends a request to the concerned actuator. In either case, the steps shown in Figure 6 take place.

After an actuator is selected, the mobile application connects to the M2M gateway to retrieve the proxy-out URI and destination URI of the actuator. The use of such URIs is explained in [21]. Once the URIs are received, the application uses HTTP POST request to send a new value to the actuator and a HTTP response 204 with no content is sent back if the request is successfully executed. Once the actuator updates its state, the new value is then pushed to the gateway which in turn pushes it to the mobile application. This is accomplished using push notification mechanism and acts as a feedback to the end user. The actuation phase is another novel aspect and extension to current the current M3 capabilities.

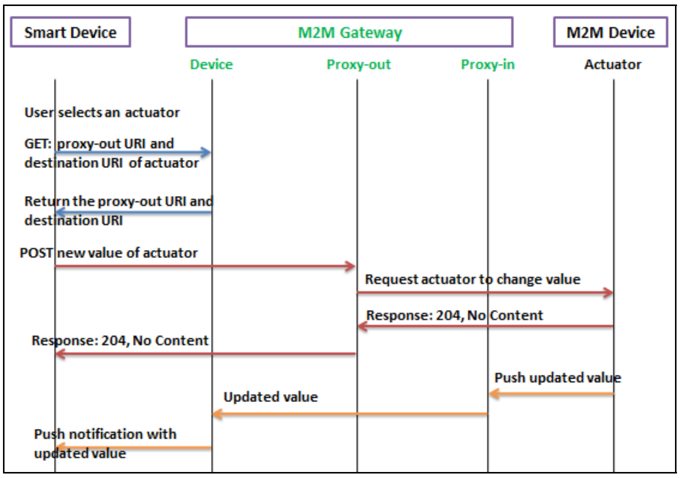

Fig. 6. Operational flow of actuation phase

Depending on usage scenarios, the lightweight M3 framework can also be integrated into the physical things. The semantic management challenge for the things is not considered in this paper.

IV. PROPOSED ARCHITECTURE AND PROTOTYPE IMPLEMENTATION

The entire system and the operational flows mentioned in the above section are fully integrated into an architecture based on recently released oneM2M recommendations [24]. The functional architecture is presented in Figure 7 and is primarily divided into field and infrastructure domain. The elements of each domain are composed of two entities. An Application Entity (AE) contains the application logic for the end-to-end M2M solutions e.g. application for automated driving, fitness monitoring. Each AE must have a unique identity. On the other hand, a Common Service Entity (CSE) represents a set of common service functions (discovery, device management etc.) of the M2M ecosystem. The architecture utilizes SenML to exchange metadata among the M2M devices, M2M gateways and end-user smart devices. Furthermore, CoRE Link Format [23] is used to describe the M2M devices and associated endpoint(s). These descriptions are ultra-lightweight and are stored in the M2M gateway [19]. Utilization of SenML and CoRE Link Format also preserves interoperability across the nodes and domains. The following subsections further illustrate each components of the architecture and the functionalities of the in-built CSE. Descriptions on different reference points are out of scope of this work. oneM2M also provides a basis of utilizing the Open Mobile Alliance Lightweight M2M (OMA LwM2M) Technical Specifications for efficient management of the physical things. Utilization of OMA LwM2M is out of scope for this paper and has been presented in [27].

A. Application Service Node

An application service node (ASN) contains at least one AE and a CSE. In this case, the ASN-AE and ASN-CSE are embedded into the mobile application running in the smart devices. The ASN-AE contains the user-centric application logic and a user interface for interaction. The ASN-CSE implements the modules for dynamic discovery of physical things, provisioning, lightweight M3 framework and actuation. In order to simplify the implementation of actuation, we have extended the capabilities of SenML to exchange metadata for actuators also [20]. This allows the same generic software implementation of SenML to be used for both sensor and actuator metadata exchanges.

The prototype application is developed in Android. The reasoning engine for the lightweight embedded framework is implemented using AndroJena which is an Android library of Jena framework. This can also be considered as an extension to the capabilities of CSF. It is also possible to share semantically annotated data between two such applications running in different smartphones. Since M3 terminology is used to design and develop the application, the data sharing can be done following a uniform manner.

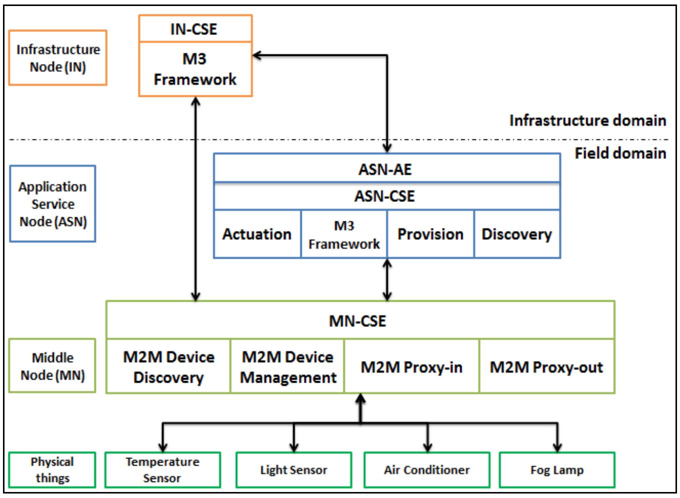

Fig. 7. oneM2M architecture facilitating user-centric cross-domain IoT applications for smart devices

B. Middle Node

According to oneM2M documentation, a middle node (MN) contains only CSE and not AE. It communicates with an infrastructure node (IN) and an ASN in this architecture. The M2M gateway corresponds to the MN. The MN-CSE contains several common service functions (CSF) such as: (i) dynamic discovery, (ii) M2M device management, (iii) security and access rights, (iv) subscription & notification and (v) M2M data management & repository. These CSFs are implemented as RESTful web services and are clustered into two interfaces – north and south. The north interface communicates with the ASN and implements the web services for discovery, security & access rights, subscription & notification. The south interface is directly linked with the physical things and manages the web services like proxy-in and proxy-out [21], M2M device management and M2M data management & repository.

In the prototype implementation, the M2M devices containing the physical things are described using CoRE Link Format. When such a device is linked to the MN, the description is pushed to the M2M device management web service. This CSF analyses the device configuration and stores that into a local repository. The discovery request parameters coming from ASN-CE are also in the same format. The discovery web service extracts the parameters from the GET request and internally queries the repository to retrieve a list of the matching M2M devices. A detailed mechanism of the discovery is out of scope of the paper.

C. Infrastructure Node

An infrastructure node (IN) provides an M2M service in the infrastructure domain. IN contains a CSE and zero or more AE. This node interacts with one or more MNs and/or one or more ASNs. The IN-CSE contains the M3 framework portrayed in Figure 1. At the provisioning phase, the interaction between the smart device (ASN) and the M3 framework (IN) happens over Mca reference points. This interaction is also done over RESTful web services as explained before.

It can also be envisioned that, for an Android powered MN, the discovery, provisioning, M3 framework can be implemented at the MN-CSE. This will further enrich the MN functionalities. The ASN can therefore be replaced with an Application Dedicated Node (ADN). It comprises of a single AE and no CSE and communicates to the MN to receive the alerts coming out of the semantic reasoning. For example, the deductions and suggestions computed by the lightweight M3 framework could be communicated to the ADN as push notification from the MN.

For smart endpoints supporting a global addressing scheme (e.g. sensors and actuators are IPv6 enabled), then the M2M gateway and its functionalities can be pushed on a cloud computing platform. Then, geographically distant endpoints can also be accessed from the same application. Such scenario can also be supported by the same architecture.

D. Use case

We demonstrate the usefulness of the cross-platform IoT application development and the above architecture using a use case. We have developed a scenario where vehicle equipment(s) can be controlled according to the weather conditions. It is assumed that the car belonging to the end user contains a light sensor and a precipitation sensor. These sensors are connected to a smart M2M gateway. The mobile application first queries the gateway to discover a list of sensors and domain names. In this case, the discovery reveals that the car has a light sensor and a precipitation sensor and both of these sensors are linked to the weather domain. If the user selects precipitation sensor and weather domain, then the mobile application queries the M3 repository (Figure 1) to obtain a list of the available cross-domain application templates. The list consists of four such scenarios: (i) Weather, Luminosity and Emotion, (ii) Weather, Tourism and Clothes, (iii) Weather, Tourism and Activities and (iv) Weather, Transportation and Safety Device. If the user chooses the last option, based on the precipitation sensor measurement, the application can deduce whether the rainfall is light, medium or heavy. According to the deduction, the user is notified to turn on the rear wiper, wiper, fog lamp etc. The user can thereafter choose to activate the equipment or the application can be configured to perform the actuation automatically. Similarly if the combination of light sensor and weather is chosen, then based on light sensor measurement the application can infer if the outside is bright sunny. In that case, it will suggest activating the sun visor.

E. Performance evaluation

We have evaluated the performance of the mobile application in terms of CPU load and power consumption. These two metrics are very important since the application is running on mobile devices with limited battery life. The application has been tested on three Android powered devices. Table 1 lists the CPU load results as measured from DDMS tool of Eclipse ADT. The loads are measured during the main four operations phases of the application.

TABLE I. CPU USAGE OF THE MOBILE APPLICATION

Android device	Discovery	Provisioning	M3 Framework	Actuation
Samsung Galaxy S2	1%	1%	14%	1%
Nexus 5	2%	1%	17%	1%
Nexus 7	1%	1%	21%	1%

TABLE II. POWER CONSUMPTION OF THE MOBILE APPLICATION

Android devices	Power consumption (mW)	
	Mobile Data	Wi-Fi
Samsung Galaxy S2	277	214
Nexus 5	301	250
Nexus 7 (Wi-Fi only)	--	379

It appears from the Table I that only semantic computing requires higher CPU load while rest of the computations and network operations require negligible amount of CPU loads. This is due to the fact that SenML and CoRE Link Format are implemented using JSON. Creating and parsing JSON objects and arrays are implemented in an ultra-lightweight fashion which reflects in the CPU load. Also the CPU load for the semantic computing does not discourage the users to use the application. The reasoning engine of M3 framework utilizes the Android library version of AndroJena and the CPU load is dependent on AndroJena. The software implementation could be further optimized to reduce the CPU load.

Table II provides the power consumption results of the application for the same devices. These values are obtained using Power Tutor. The above measurements are average power consumption taken over the life cycle of the application mentioned in the use case. A detailed look into Power Tutor revealed that around 85% of the power is spent at the display hardware. Therefore the network operations and the CPU computations take minimal power. The dissipation at the display can be reduced by lowering the brightness value of the smart device [25, 26]. Thus the overall performance of the mobile application is quite satisfying. Therefore such user centric application will empower the end users and enable rapid adaption of IoT enabled services.

V. CONCLUSION

In this paper, we describe a deeper analysis of the state of the arts related to integrating semantics in mobile devices. The M3 framework and its lightweight version embedded into a mobile application are discussed. The utilization of the lightweight M3 framework in developing user-centric cross domain application is described. The four different phases of operations are detailed. An oneM2M architecture incorporating the entire components is presented along with its prototype implementation. The performance evaluation of the prototype mobile application demonstrates the lightweight implementation. The novel features of the framework and the overall contributions of the paper are outlined. In a nutshell, the work demonstrates that adoption of semantic based computing is feasible on Android powered devices. As for future work, we are developing more use cases of cross domain IoT applications and plan to work on the reference points of the oneM2M architecture.

ACKNOWLEDGMENT

The authors would like to thank the semantic web experts (Ghislain Atemezing, Payam Barnaghi and Bernard Vatant), colleagues, friends and students for their valuable inputs and fruitful discussions. This work is supported by the Com4Innov Platform of Pole SCS[10] and French research project DataTweet[11] (ANR-13-INFR-0008).

REFERENCES

[1] Luigi Atzori, Antonio Iera, Giacomo Morabito, The Internet of Things: A survey, Computer Networks, Volume 54, Issue 15, 28 October 2010, Pages 2787-2805.

[2] Danh Le Phuoc, Josiane Xavier Parreira, Vinny Reynolds, and Manfred Hauswirth. Rdf on the go: Rdf storage and query processor for mobile devices. In ISWC Posters&Demos, page 12, 2010.

[3] Jérôme David, Jérôme Euzenat, Maria Rosoiu, et al. Linked data from your pocket. In Proc. 1st ESWC workshop on downscaling the semantic web, pages 6–13, 2012.

[4] Christian Becker and Christian Bizer. Dbpedia mobile: A location-enabled linked data browser. In Proceedings of the World Wide Web (WWW) Workshop on Linked Data on the Web, 2008.

[5] M. Compton, P. Barnaghi, L. Bermudez, R. Garcia-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, et al. The ssn ontology of the w3c semantic sensor network incubator group. Web Semantics: Science, Services and Agents on the World Wide Web, 2012.

[6] d'Aquin, Mathieu; Nikolov, Andriy and Motta, Enrico (2011). Building SPARQL-Enabled Applications with Android devices. In: 10th International Semantic Web Conference (ISWC2011), 23-27 Oct 2011, Bonn, Germany.

[7] Aitor Gómez-Goiri, Pablo Orduña, Javier Diego, and Diego López-de Ipiña. Otsopack: Lightweight semantic framework for interoperable ambient intelligence applications. Computers in Human Behavior, 30:460–467, 2014.

[8] Safdar Ali and Stephan Kiefer. µor–a micro owl dl reasoner for ambient intelligent devices. In Advances in Grid and Pervasive Computing, pages 305–316. Springer, 2009.

[9] Boris Motik, Ian Horrocks, and Su Myeon Kim. Delta-reasoner: a semantic web reasoner for an intelligent mobile platform. In Proceedings of the 21st international conference companion on World Wide Web, pages 63–72. ACM, 2012.

[10] Yevgeny Kazakov, Markus Krötzsch, and Frantisek Simancik. Elk reasoner: Architecture and evaluation. In ORE, 2012.

[11] Michele Ruta, Floriano Scioscia, Filippo Gramegna, and Eugenio Di Sciascio. A mobile knowledge-based system for on-board diagnostics and car driving assistance. In UBICOMM 2010, The Fourth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, pages 91–96, 2010.

[12] Hang Liu. Improving semantic interoperability in remote patient monitoring. Master's thesis, 2013.

[13] Johann Vincent, Tom Dubin, Christine Porquet, et al. Protection de la vie privée basée sur des ontologies dans un système android. APVP 2012 (Atelier Protection de la Vie Privée, 3ème édition), 2012.

[14] Johann Vincent, Christine Porquet, Maroua Borsali, and Harold Leboulanger. Privacy protection for smartphones: an ontology-based firewall. In Information Security Theory and Practice. Security and Privacy of Mobile Devices in Wireless Communication, pages 371–380. Springer, 2011.

[15] Johann Vincent, Christine Porquet, Idriss Oulmakhzoune, et al. Ontology-based privacy protection for smartphone: A firewall implementation. In International Conference on Secure Networking and Applications (ICSNA), 2011.

[16] Hung-Yu Chien, Shyr-Kuen Chen, Ching-Yang Lin, et al., "Design and Implementation of ZigBee-Ontology-Based Exhibit Guidance and Recommendation System," International Journal of Distributed Sensor Networks, vol. 2013.

[17] Kerry Skillen, Liming Chen, Chris D Nugent, Mark P Donnelly, and Ivar Solheim. A user profile ontology based approach for assisting people with dementia in mobile environments. In Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE, pages 6390–6393. IEEE, 2012.

[18] Gyrard, A.; Datta, S.K.; Bonnet, C.; Boudaoud, K., "Standardizing generic cross-domain applications in Internet of Things," Globecom Workshops (GC Wkshps), 2014, pp.589,594, 8-12 Dec. 2014.

[19] Datta, S.K.; Bonnet, C., "Smart M2M Gateway Based Architecture for M2M Device and Endpoint Management," Internet of Things (iThings), 2014 IEEE International Conference on, and Green Computing and Communications (GreenCom), IEEE and Cyber, Physical and Social Computing(CPSCom), IEEE, pp.61,68, 1-3 Sept. 2014.

---

[10] http://www.pole-scs.org/

[11] http://www.agence-nationale-recherche.fr/?Projet=ANR-13-INFR-0008

[20] Datta, S.K.; Bonnet, C.; Nikaein, N., "CCT: Connect and Control Things: A novel mobile application to manage M2M devices and endpoints," Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference on , vol., no., pp.1,6, 21-24 April 2014.

[21] Datta, S.K.; Bonnet, C.; Nikaein, N., "An IoT gateway centric architecture to provide novel M2M services," Internet of Things (WF-IoT), 2014 IEEE World Forum on , vol., no., pp.514,519, 6-8 March 2014.

[22] Cullen Jennings, Zach Shelby and Jari Arkko. Media types for sensor markup language (SenML). IETF Network Working Group, Internet-Draft, https://tools.ietf.org/html/draft-jennings-core-senml-00.

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

[24] TS-0001-oneM2M-Funtional-Architecture-V-2014-08. http://www.onem2m.org/images/files/deliverables/TS-0001-oneM2M-Functional-Architecture-V-2014-08.pdf

[25] Datta, S.K.; Bonnet, C.; Nikaein, N., "Android power management: Current and future trends," Enabling Technologies for Smartphone and Internet of Things (ETSIoT), 2012 First IEEE Workshop on , vol., no., pp.48,53, 18-18 June 2012.

[26] Datta, S.K.; Bonnet, C.; Nikaein, N., "Minimizing energy expenditure in smart devices," Information & Communication Technologies (ICT), 2013 IEEE Conference on , vol., no., pp.712,717, 11-12 April 2013.

[27] Datta, Soumya Kanti; Bonnet, Christian, "A lightweight framework for efficient M2M device management in oneM2M architecture," Recent Advances in Internet of Things (RIoT), 2015 International Conference on , vol., no., pp.1,6, 7-9 April 2015.

[28] https://jena.apache.org/

[29] Gyrard, A.; Datta, S.K.; Bonnet, C.; Boudaoud, K., "A semantic engine for Internet of Things: cloud, mobile devices and gateways," in 4th International Workshop on Extending Seamlessly to the Internet of Things (esIoT-2015), co-located with IMIS-2015, 8-10 July 2015.