Easing IoT Application Development Through DataTweet Framework

Soumya Kanti Datta and Christian Bonnet
Communication Systems Department, EURECOM
Sophia Antipolis, France
Emails: {dattas, bonnet}@eurecom.fr

***Abstract*—Current IoT application development frameworks have limited capabilities. There is a lack of horizontal development approach, interoperability with standards, adequate security mechanisms, integration of information centric networking among others. The paper counters the challenges using a novel DataTweet IoT application development framework. It decouples application logic (AL) from common IoT functionalities. This allows the IoT stakeholders to focus on the AL and use open source, standardized APIs for the latter. An automotive IoT application for Advanced Driver Assistance System (ADAS) is developed using the framework and its operational phases are highlighted. Initial evaluation shows that the utilization of the framework results in 72.1% reduction of hand-written codes.**

Keywords—DataTweet; Data processing; Internet of Things; Mobile Edge Computing; Security; Standards.

I. Introduction

The Internet of Things (IoT) have gained a lot of momentum from the industry, academia and consumers in recent years. The IoT market is expanding with applications and services. Still the IoT ecosystem faces an important challenge to enable experts and various stakeholders to rapidly develop applications. IoT being a vast umbrella that includes smart home, smart manufacturing, intelligent transportation system, smart grid, fitness and health management and more domains makes it even difficult for developers to create an application. The developers would need to master domain knowledge, interworking with physical things, communication technologies, protocols, gateway programming, cloud computing, semantic web, RESTful design and much more. All such factors together point to the need of IoT application development framework. It (i) simplifies development process, (ii) hides the complexities of programming & security mechanisms from developers, (iii) reduces time to market for industries, (iv) provides standard oriented APIs & interfaces and (v) offers state-of-the-art consumer experience.

This paper aims at easing the IoT application development process through proposed DataTweet framework. Its novel aspects and contributions are (i) open source APIs for common IoT functionalities (e.g. discovery, registration, management, security) to develop both vertical and horizontal applications, (ii) incorporating strong security mechanism by design, (iii) interoperability through oneM2M standard, (iv) separation of application logic from common IoT functionalities, (v) wide range of deployment capabilities and software tools for developers, (vi) in-built life cycle management of the developed application and (vii) integration of information centric networking (ICN). We have also examined the currently available frameworks, their merits and demerits. Apart from that, requirements of such a generic IoT application development framework is presented. Our proposed framework meets the requirements and address the limitations in existing literature.

Rest of the paper is organized as follows. Section II presents the status of current literature. Section III presents the generic requirements while Section IV describes the proposed framework, its two main components. Interoperability and deployment with respect to oneM2M architecture [15] is highlighted here. Section V describes an automotive IoT application developed with it, its operational phases and an initial evaluation. Finally, the paper concludes with a summary of contributions and future outlook.

II. State-of-the-Art

This section highlights the state-of-the-art in IoT application development frameworks as well as their merits and demerits.

A. *Domain specific IoT application development*

The paper [1] has provided an extensive study on IoT application development. It identifies the relevant stakeholders (e.g. software developers, domain experts), challenges and their approach for application development. It involves conceptual modelling with (i) domain specific concepts involving an Entity of Interest (EoI), a resource (e.g. sensor, actuator, storage or user interface) and a region (location of a device), (ii) functionality specific concepts involving application logic and interaction among computations elements, (iii) deployment specific concepts, and (iv) platform specific concepts which relates to operating system based drivers for a hardware device running an IoT application. The evaluation of the framework with respect to a fire detection application outlines the number of handwritten codes for the above mentioned four concepts and the generate lines of codes. In this case the percentage of generated code is almost 84% compared to the total lines of codes. This shows the ease of IoT application development with the framework.

The paper [2] evaluates the ideas put forward in the paper [1]. The authors used two criteria – expressiveness (ability to develop wide range of applications) and development efforts (denotes the no. of handwritten codes to develop an application, more such code means more efforts). During the evaluation based on expressiveness, the authors looked at the characteristics of IoT applications that can be modelled by their approach. The characteristics include the application logic, goal, topology, scale, entity type, interaction pattern and consumer logic. To calculate the development efforts, the authors experimented with building automation (regulating temperature, calculating average temperature and detecting fire). Their approach showed the limited lines of handwritten code that is necessary for the mentioned IoT application.

Authors of [3] focus on domain models with concepts and associates suitable for IoT applications. They broadly categorized the concepts into two sections – (i) the traditional Internet concepts consisting of a computational service, storage service or end-user application and (ii) thing oriented concepts involving EoI, resource, raw data, event, property, action and command. The associations are basically relations among the mentioned concepts. Using these modelling, the authors validated three applications – shared book reviews, HVAC maintenance and smart plants.

B. Cross domain IoT application development

Majority of the current literature focus on domain specific or vertical IoT application development frameworks. It is widely agreed that the main innovations and values from IoT can be derived through cross domain or horizontal applications. The paper [4] investigates a semantic based Machine-to-Machine Measurement (M3) Framework for cross domain development. The M3 framework allows (i) semantic annotation of sensor data originating at heterogeneous domains, (ii) semantic reasoning on sensor data based on Sensor based Linked Open Rules (S-LOR), (iii) infer high-level abstraction from raw sensor data, (iv) uniform nomenclature to describe sensor data, (v) generation of IoT application development templates. The framework is evaluated in terms of software performance and semantic best practices. Integration of the M3 into oneM2M standard has been described in [5]. An extension of the work is presented in [6] which considers extending the M3 framework for consumer mobile devices like smartphones and tablets. The original M3 is developed using Apache Jena Framework and is deployed in the Google Cloud Platform. For Android powered mobiles devices, AndroJena library has been used. The paper outlines how consumer centric IoT applications can benefit from device discovery, provisioning, semantic reasoning and actuation. The implementation is oneM2M standard oriented and the evaluation of the application is performed based on memory, CPU load and battery requirements. The results prove that the mobile application is lightweight.

C. IoT application development suite

The paper [7] presents IoTSuite which is a set of tools for rapid prototyping of IoT applications. The proposed approach includes several steps – (i) specifying domain vocabulary, (ii) compiling vocabulary specification, (iii) specifying application architecture, (iv) compiling architecture specification, (v) implementing application logic, (vi) specifying target deployment, (vii) mapping of computational services to a set of devices, (vii) implementing device drivers, (viii) linking and (ix) evolution. The IoTSuite is composed of – (i) an editor which helps relevant stakeholders to create high level specs, (ii) a compiler which translates the high level specifications into codes, (iii) a mapper which produces a mapping from a set of IoT services into a set of IoT devices, (iv) a linker that puts together the generated codes to be deployed on IoT devices and (v) runtime system. The paper also provides a comparison on existing IoT tool suites including IoTSuite, DiaSuite, ATaG, PervML, RuleCaster, VisualRDK, WoTKit, Context Toolkit and Pantagruel. another detailed such survey is available at [8].

D. Context and discovery based IoT applications

Involving context awareness and semantic based discovery into IoT application development is equally important. These aspects are examined in [9]. The proposed architecture uses a context manager that manages the user, service and environment contexts. Each of these context modules receives inputs from sensors in respective domains. But the user and service context modules depend on additional requirements like user requirement on service context and user requirement on environment context. The discovery mechanism has been intelligently merged on top of the entire context monitor. This in turn allows the user context module to run basic or sophisticated discovery for IoT applications. The same concept extends the service context module to run search functions on users and environment. This is helpful for service providers. Filtering capabilities are also integrated and allows three types of filtering – basic, semantic and context based filtering. The work is highly relevant to IoT application development to allow users or service providers to search for required devices or services. Involving semantic web technologies allows high richness in query as a part of discovery mechanism. The filtering methods provide a concise view of the available results. But the paper does not provide any implementation or prototyping details.

E. Limitations

While performing the literature survey, we noticed several limitations as mentioned below.

- Many papers limit their discussion to domain specific IoT application development limiting the benefits to a segment of IoT ecosystem.
- The state of security mechanisms in the mentioned frameworks is not clear.
- The current works except [4] and [6] do not outline their position with respect of IoT standards like oneM2M or W3C Web of Things. This in turn encourages data, product and implementation silos which stand in the way of global interoperability.
- The studied frameworks do not provide any guidelines for generic and wider deployment facilities including smartphones, cloud and edge devices.
- The IoT application logic must be separated from common IoT functionalities (e.g. resource discovery, management, registration, binding to protocols). This allows application logics to be decoupled from rest of the functionalities. This is lacking in many approaches presented in the literature.
- The data centric approach for IoT ecosystem [11] are not explored in the literature.
- Traditional HTTP based information fetching and dissemination does not work well in IoT scenarios involving high degree of mobility. For those scenarios, ICN is a better choice as identified in [12]. Current literatures do not investigate implementation of any ICN

mechanism as a part of IoT application development framework.

III. IoT Application Framework Requirements

This section highlights the generic requirements that such frameworks must provide. We have previously studied the IoT data cycle in [11] as a part of designing data centric IoT services. The requirements below are derived from them. The following list is non-exhaustive but comprise of the most important requirements.

- **Open source framework and APIs –** For faster adoption in the IoT and software developer community, the IoT application development framework must be open source. This will encourage collaborative development from several contributors and will generate diverse IoT functionalities and scenarios among other benefits. With many industries adoption the "API First" approach for IoT based development, the framework must incorporate open and easy to use APIs. These should address common IoT functions. Thus the software developers will be able to concentrate on building the application logic while utilizing the open APIs. This will potentially reduce the time-to-market for any IoT application.

- **Strong security mechanisms by design –** The framework must support state-of-the-art security mechanisms by design to protect the privacy and ensure consumer trust on the generated IoT applications.

- **Interoperability through standards –** To mitigate the current silo based approaches, the framework must integrate software components based on IoT standards. This will pave way for interoperability at consumer IoT devices, applications and services. Following the current trends and recommendation for Standard Development Organizations (SDO), the framework should provide services over RESTful interactions.

- **Separation of application logic from common IoT functionalities –** To allow the stakeholders in IoT ecosystem to focus more on the M2M/IoT application, the application logic should be decoupled from rest of the common IoT functions. The framework must provide appropriate APIs for the common functions to allow the developers reduce application development cycle and time-to-market.

- **Wide range of software tools –** In order to maximize the adoption of the framework, it must offer the developers a wide range of choices for software tools.

- **Deployment capabilities –** With the IoT umbrella growing bigger every day to house new scenarios, it is necessary to support wide range of deployment capabilities. This includes constrained and powerful IoT devices, wearables, smartphones, tablets, M2M gateways, edge devices (e.g. road side units, base stations) as well as cloud computing systems.

- **Life-cycle management –** The framework must also allow the stakeholders in managing the IoT application lifecycle. Extending, adding and removing application features must be supported.

- **Information centric networking (ICN) integration –** To cater to wide range of scenarios especially automotive IoT, the framework must support ICN for secure and robust data dissemination.

IV. DataTweet IoT Application Development Framework

This section dives deep into the proposed DataTweet IoT framework for rapid IoT application development and deployment. The software framework is depicted in Figure 1.

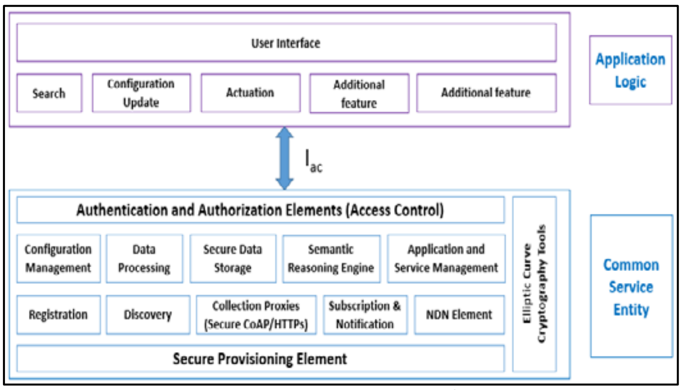

Fig. 1. The two entities and their components of DataTweet Application Development Framework.

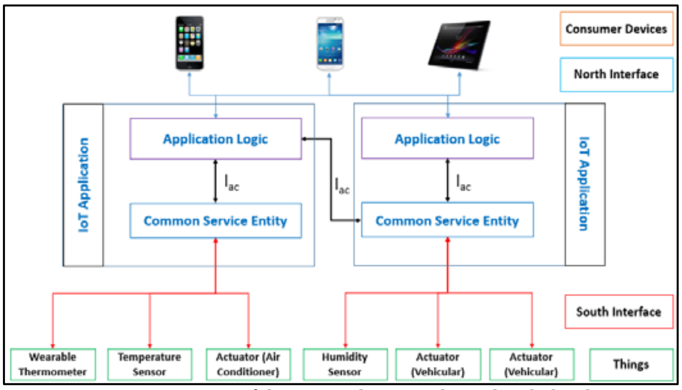

Fig. 2. Interaction of the IoT applications derived with the above framework with consumer devices and physical things.

The framework itself and any IoT application derived using that are composed of two main entities – (i) application logic (AL) and (ii) common service entity (CSE) that houses the mentioned common IoT related functionalities. The interface between AL and CSE is denoted as I_{ac}. The CSE interacts with the IoT devices like sensors, actuators, RFID or NFC tags through south interface. The AL allows the consumers to connect with the underlying IoT application through north interface. To promote cross domain IoT application scenario, CSE can be connected sensors and actuators belonging to different domain of operations (e.g. building automation, fitness monitoring, transport). Figure 2 portrays how the application derived using the framework interacts with consumers and physical things.

A. Application Logic

The application logic is where all stakeholders of the IoT application are involved. AL includes design of the user interface (UI) and its operational phases. Since the consumers directly interact with the AL, it also enables search (basic and advanced semantic searches), configuration updating capabilities, sending command to actuators and any additional features that is necessary. The AL belonging to one application may need to interact with CSE of another to support horizontal IoT use cases. The functionalities of the AL are mainly developed using web services. The interactions taking place at the north interface is mainly accomplished by RESTful web services. Another important feature is that AL is separate from the common IoT functions residing in CSE. The decoupling allows developers focus more on the consumer centric aspects of the overall IoT application and utilize the open APIs provided for the CSE.

B. Common Service Entity

Identifying and combining several common functionalities necessary to any IoT application scenario is another stepping stone for simplifying the framework. The CSE achieves that purpose along with providing open source and standard APIs to the developers. All included functionalities include security algorithms by design. This conforms to the "API First" and security by design criteria from the list of requirements. We describe the common functionalities below.

1) Collection Proxies

In order to cater to a wide range of communication technologies and protocols used by heterogeneous physical things, we introduce a software module called "collection proxies". It includes the software drivers and libraries for popular IoT protocols like HTTP, CoAP and MQTT as well as communication technologies including Bluetooth Low Energy (BLE), Wi-Fi and 3G/LTE. Another important contribution of this module is that it provides binding to the mentioned protocols. This is a significant extension to the proxy-in and proxy-out concepts introduced in [13].

2) Registration

The registration module achieves two purposes within the framework – (i) it allows AL of an authorized IoT application to register with a registrar CSE in order to allow the AL use the IoT devices connection to registrar CSE thereby facilitating cross domain and inter application interaction (done through I_{ac} interface) and (ii) it also allows underlying IoT devices to register themselves to provide data to applications. The IoT device and endpoint descriptions (CoRE Link based or semantic based [14]) are stored locally in a secure storage of the CSE. In case of CoRE Link based description, the payload supports both XML and JSON based encodings. For semantic based payload, JSON-LD is supported which is a serialization format of RDF. This gives developers more choices on the software tools.

3) Configuration Management

The configuration of IoT devices being used by applications can change due to mobility, ownership, location, life cycle etc. Therefore, the IoT application must have capabilities to understand the IoT device configuration as well as equip the consumers with a way to add, modify or delete configuration. This software module pertains to both these tasks and implements the Open Mobile Alliance Lightweight M2M (OMA LwM2M) technical specifications. Thus the CSE not only benefits from an open and popular standard but allows self-management of IoT devices and fine grain control over management operations. The module depends on the previous module for IoT device registration. Apart from that it provides service enablement and un-registration functions.

4) Discovery

Any IoT application essentially needs to search for physical things to get their descriptions, data and send actuation commands [16]. The search function of the AL allows consumers to enter queries based on keywords (e.g., "search all temperature sensors in my house"). Also an IoT application can itself initiate the search. In either case, the query is forwarded to the discovery module in CSE which then searches in the local database for required devices or data. The module implements three kinds of discovery mechanisms.

- Search in the vicinity for BLE beacon enabled devices e.g. things utilizing physical web approach.
- Search in a network using protocol like SSDP.
- Search in a local directory where the devices have already registered.

To allow richness in consumer query, semantic based discovery is implemented in the AL. This allows search functions to identify which mechanism to adopt for device discovery. Following that, the available results can be filtered and ranked based on query parameters.

5) Data processing and semantic reasoning engine

The data processing module is composed of a data collection service that fetches raw data from available sensors. To promote interoperability at data processing stage and utilization of a standard set of vocabulary, we advocate for semantic web technologies for data processing. Thus a semantic reasoning engine is integrated into the framework. The data collection service receives data in Sensor Markup Language (SenML) which supports XML, JSON, EXI and CBOR encodings. Then the data is converted into RDF using a converter before being fed to the reasoning engine. Finally, SPARQL queries are run to generate high level abstractions as well suggestions for consumers given the application scenario. Semantic reasoning is particularly useful for cross domain IoT application development. These modules implement the M3 Framework described in [4] and also promote a data centric approach.

6) Subscription and notification

This module allows sending notifications for a subscription API that tracks – (i) configuration changes of a thing, (ii) occurrence of a predefined event. The notification function can also be used to send alerts in case of emergency.

7) Application and service management

This module takes care of life cycle management for the developed IoT application. It includes capabilities to configure, update, troubleshoot and support the overall functions of the application.

8) Security

For the overall framework, this module comprises of mechanisms for – (i) sensitive data handling (e.g. location of things), (ii) security administration, (iii) establishing security association, (iv) enforcing access control policies for identification, authorization and authentication, (v) identity management, (vi) securing the local storage and (vii) secure provisioning of registered IoT devices. As a part of sensitive data handling, the module incorporates Elliptic Curve Cryptography (ECC) based methods to protect AL and consumer credentials. During security associate establishment between the IoT application and consumer devices/physical things, confidentiality, integrity and signature verification are performed. The access control (AC) is utmost important to limit the discovery to both authorized users as well as authorized things. AC also necessary for consumer accessing IoT device configurations for adding, updating or deleting them. ECC is also used for security key generation, exchange as well as encryption and decryption of discovery and configuration metadata exchange. AC based on RBAC and DCapBac [10] (capability based access control) are provided. Utilization of CoAP further assures DTLS based security for the data transfer with sensors and actuators.

9) Named Data Networking (NDN)

Integrating NDN module is a major contribution of this work. The inherent issue with HTTP based IoT systems is that they are not designed to handle mobility and suffer from centralized DNS mechanism. With NDN, information can be disseminated to any consumer device based on an interest. NDN utilizes inbuilt security checks for data originator and incorporates a message digest [12]. This allows the framework to move beyond the client-server paradigm of the Internet.

C. Deployment capabilities

From the industrial perspective, wide range of deployment capabilities are necessary for IoT applications. The framework allows creation of applications that can be deployed in a smartphone, an IoT gateway (edge device) or a cloud server. This is possible due to the decoupling of AL from common IoT functions. The CSE is developed in such a way that it can generate executables for various runtime environments.

D. Interoperability with oneM2M standard

As a first step towards interoperability, we have mapped the components of the proposed DataTweet framework to that of the oneM2M standards.

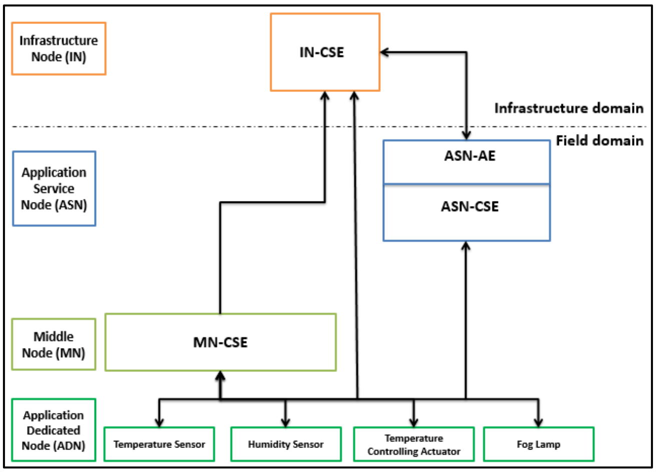

Fig. 3. Deploying proposed IoT framework components as a part of oneM2M standard architecture.

The AL of any IoT application corresponds to the Application Entity (AE). The interface I_{ac} (in Figure 1) is identical to M_{ca} interface of the oneM2M architecture. In both frameworks, CSE and its functionalities remain identical. In terms of the deployment and architecture, our framework fits well with the following oneM2M architecture as shown in Figure 3. As seen, the application dedicated nodes (ADN) consist of physical things like sensors and actuators. The middle node (MN) corresponds to IoT gateways or edge devices and is composed of just the CSE. The consumer devices like connected cars, smartphones or laptop computers form application service nodes (ASN) which comprises of both the AL and CSE. The IoT applications developed using the framework run here as well as can run the in the infrastructure node (cloud system) as an extension to the oneM2M standard. The framework also follows oneM2M recommendations including utilization of RESTful web services for exchanging data and control information, binding to common protocols and utilization of semantic web technologies for uniform vocabulary, metadata and reasoning engine.

V. Prototyping and Evaluation

In this section, we describe an automotive IoT application developed using this framework and its phases of operation. We have envisioned an advanced driver assistance system (ADAS) application for night time driving. This is an extension to [17]. The application utilizes the raw data collected from vehicular and environmental sensors. If the humidity (environmental) sensor data processing reveals there is fog, the ADAS application will automatically switch on the fog lamp. We explain the utilization of AL and CSE for this particular use case through the different phases of operations. Security key exchange for establishing association and access control is not shown but are included in the application. Note that the derived application is running on a smartphone attached to the vehicle. The first phase of operation (Figure 4) includes vehicular and environmental sensors registering their configurations to the CSE which stores the thing descriptions in a secure local storage. When the driver is initiating the search for available sensors, the request is forwarded to the discovery module which employs the directory based search in this scenario. The discovered list of devices is sent to the AL which is then updated to the application user interface.

In the second phase of operation (Figure 5), the driver provisions the IoT application for ADAS with a humidity sensor (assuming it exists in the discovery list). Then the AL instructs the collection proxies to collect data from the use sensor. The collected data is in SenML form and undergoes transformation into RDF at data processing module and later a high level intelligence is measured from that using the semantic reasoning engine. This can be accomplished in a smartphone as the M3 framework can be ported to an Android powered smartphone. If the high level intelligence suggests there is fog in the driving environment, then a notification alert can be sent to the driver. The embedded intelligence in the IoT application can also use the “actuation” feature in AL to turn on the fog lamp automatically. This will send actuation command to the collection proxies module which will relay the command (turn on) to the fog lamp. This completes the third and final phase of operation of the scenario and is shown in Figure 6.

A. Evaluation

At first, we evaluated the proposed framework with respect to the limitations found in state-of-the-art and proposed requirements. The framework successfully meets all requirements. An initial evolution the prototype application reveals that hand-written codes needed for the application is reduced by 72.1% compared to the one without the framework. This shows the importance of the open source CSE.

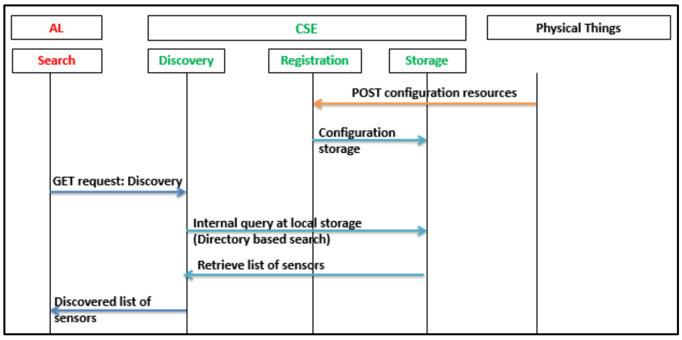

Fig. 4. Phase I with registration and discovery.

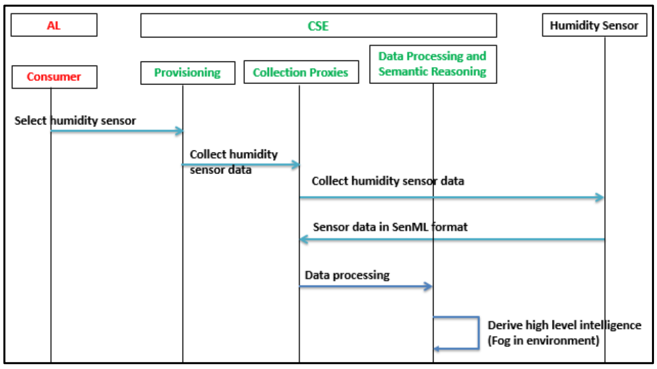

Fig. 5. Phase II with provisioning data processing with semantic reasoning.

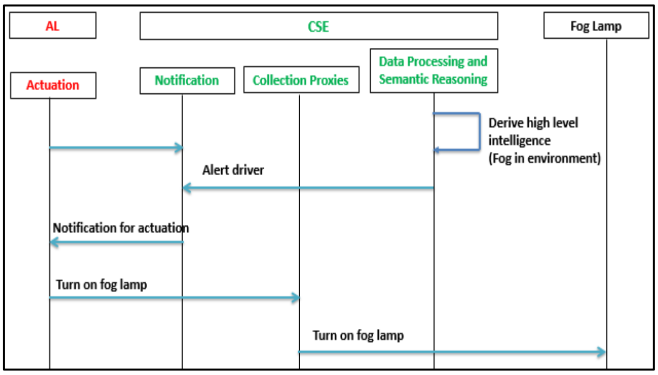

Fig. 6. Phase II with provisioning data processing with semantic reasoning.

VI. Conclusion

The paper aims to ease the IoT development process through the proposed DataTweet framework. It introduces significant values and innovations that are not covered by currently available frameworks. The paper also lists several requirements for a generic IoT application development framework. The application logic is decoupled from the common service entity leading to stakeholders focusing more on the AL. Through the open source APIs of the CSE, the handwritten codes can be reduced to a great extent as found in case of the described use case. The contributions of the framework are highlighted throughout the paper. As future work, we are integrating the framework into a test bed to further examine its performance for industrial applications.

Acknowledgment

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

References

[1] Pankesh Patel, Damien Cassou, Enabling high-level application development for the Internet of Things, Journal of Systems and Software, Volume 103, May 2015, Pages 62-84, ISSN 0164-1212.

[2] Pankesh Patel, Ajay Kattepur, Damien Cassou, Georgios Bouloukakis. Evaluating the Ease of Application Development for the Internet of Things. [Technical Report] 2013.

[3] Pankesh Patel, Animesh Pathak, Thiago Teixeira, and Valérie Issarny. 2011. Towards application development for the internet of things. In Proceedings of the 8th Middleware Doctoral Symposium (MDS '11). ACM, New York, NY, USA, Article 5, 6 pages.

[4] A. Gyrard, S. K. Datta, C. Bonnet and K. Boudaoud, "Cross-Domain Internet of Things Application Development: M3 Framework and Evaluation," Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on, Rome, 2015, pp. 9-16.

[5] A. Gyrard, S. K. Datta, C. Bonnet and K. Boudaoud, "Integrating machine-to-machine measurement framework into oneM2M architecture," Network Operations and Management Symposium (APNOMS), 2015 17th Asia-Pacific, Busan, 2015, pp. 364-367.

[6] S. K. Datta, A. Gyrard, C. Bonnet and K. Boudaoud, "oneM2M Architecture Based User Centric IoT Application Development," Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on, Rome, 2015, pp. 100-107.

[7] Soukaras, D., Patel, P., Song, H., Chaudhary, S.: Iotsuite: a toolsuite for prototyping internet of things applications. In: The 4th International Workshop on Computing and Networking for Internet of Things (ComNet-IoT), co-located with 16th International Conference on Distributed Computing and Networking (ICDCN 2015).

[8] Patel, P., Jardosh, S., Application Development Approaches for the Internet of Things: A Survey, in IEEE Region 10 Symposium 2015.

[9] V. Suraci, S. Mignanti and A. Aiuto, "Context-aware Semantic Service Discovery," 2007 16th IST Mobile and Wireless Communications Summit, Budapest, 2007, pp. 1-5.

[10] Jose L. Hernandez, Antonio J. Jara, Leandro Marinc and Antonio F. Skarmeta G6meza. DCapBAC: Embedding Authorization logic into Smart Things through ECC optimizations. International Journal of Computer Mathematics, 1-22,2014..

[11] S. K. Datta, C. Bonnet, R. P. Ferreira Da Costa and J. Härri, "DataTweet: An architecture enabling data-centric IoT services," 2016 IEEE Region 10 Symposium (TENSYMP), Bali, 2016, pp. 343-348.

[12] S. K. Datta and C. Bonnet, "Integrating Named Data Networking in Internet of Things architecture," 2016 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), Nantou, 2016, pp. 1-2.

[13] S. K. Datta, C. Bonnet and N. Nikaein, "An IoT gateway centric architecture to provide novel M2M services," Internet of Things (WF-IoT), 2014 IEEE World Forum on, Seoul, 2014, pp. 514-519.

[14] S. K. Datta and C. Bonnet, "Describing things in the Internet of Things: From CoRE link format to semantic based descriptions," 2016 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), Nantou, 2016, pp. 1-2.

[15] J. Swetina, G. Lu, P. Jacobs, F. Ennesser and J. Song, "Toward a standardized common M2M service layer platform: Introduction to oneM2M," in IEEE Wireless Communications, vol. 21, no. 3, pp. 20-26, June 2014.

[16] S. K. Datta, R. P. F. Da Costa and C. Bonnet, "Resource discovery in Internet of Things: Current trends and future standardization aspects," Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on, Milan, 2015, pp. 542-547.

[17] S. K. Datta, C. Bonnet and J. Haerri, "Fog Computing architecture to enable consumer centric Internet of Things services," 2015 International Symposium on Consumer Electronics (ISCE), Madrid, 2015, pp. 1-2.