DataTweet: An Architecture Enabling Data-Centric IoT Services

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

***Abstract*—This work proposes a data driven IoT architecture to mitigate the fragmentation and data silos in consumer IoT ecosystem. IoT data cycle is explored to understand data transformation from generation to processing, storage and dissemination. We have developed a framework called DataTweet focusing on data cycle based IoT services. The DataTweet architecture is presented along with its subsystems, services, requirements and common service functions necessary to implement the services. The main novelty of DataTweet is in creating a ubiquitous consumer data service for transmitting short messages to any computing platforms to generate actionable intelligence that is disseminated to consumers.**

***Keywords*— Data Cycle; DataTweet; Discovery; IoT; Named Data Networking; oneM2M standard; Smart City.**

I. Introduction

The IoT is set to bring numerous benefits to the consumers for the foreseeable future. But the IoT platforms and ecosystems suffer from silo-based and individualized implementations. This in turn results into fragmented consumer market in which IoT applications and services do not interoperate seamlessly [18]. Realizing the fact that the intelligence lies in the data itself while communication is increasingly becoming an enabler for advanced data analysis, this work takes a data driven approach to mitigate the problem. DataTweet proposes to study the IoT data cycle under a vision for exploring the idea of a ubiquitous consumer data service for transmitting short messages in a similar way to Twitter. A consumer of the DataTweet service or a data source can send a short message at a very low rate to other entities of the IoT ecosystem (e.g. Cloud servers, edge computing platforms or smart device of interested consumers). The communication can be handled through some thematic channels over various access networks like open 802.11 hotspots, LTE/LTE-A base stations, vehicle-to-infrastructure communication (using ITS-G5) [9]. The consumer or a device sends its message to the nearest available wireless network (for instance an 802.11 hotspot) that will forward it to the destination by the elements of the infrastructure network. From a broader perspective, such data generation to processing, storage and dissemination services have much larger scope. This is illustrated with the help of the water cycle analogy in Fig. 1. Consumer smart devices, vehicle and other sensors deployed in smart city domain generate data that are sent to the edge or cloud platform through nearest access points like water evaporating to the sky to form clouds. The collected data are processed to extract useful information in the computing platforms through semantic treatment of the data or by employing data mining and artificial intelligence algorithms. Base on the expressed interest, the cloud can shower (disseminate) necessary data to the consumer encapsulated in short messages at the target areas.

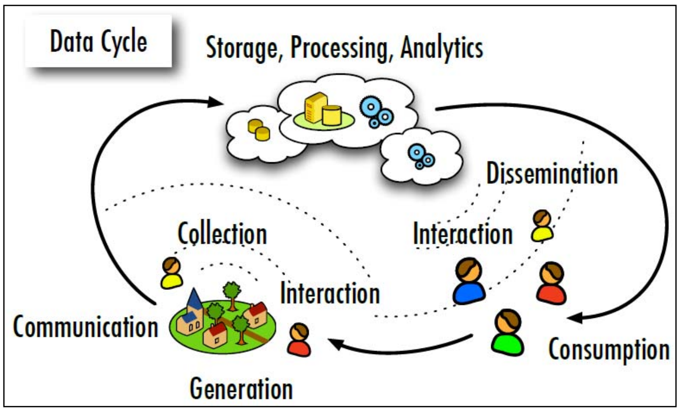

Fig. 1. Data Cycle and Services of Data Tweet for IoT ecosystem.

The data cycle model allows any device with sensor(s) to report data encoded in short messages. The raw data reaches a central or distributed computing platform where it undergoes transformation and evolves into a rich and structured information that can provide actionable intelligence to higher layer applications. Finally, the derived intelligence can reach consumer devices that paves the way for interesting context dependent IoT applications. The proposed data cycle model and DataTweet architecture fit well into smart city [11] and large scale crowd-sensing [12] based IoT projects. The work can mitigate a common problem of developing an efficient and scalable mechanism for efficient interconnection and information exchange among heterogeneous consumer devices, platforms and services. Another distinct advantage is that such data centric model can overcome the fragmented approaches of individualized solutions prevalent in the current IoT ecosystem. Fusion of data coming from several heterogeneous domains into a computing platform can potentially extend the smart city research avenues and commercial solutions. The main aim of the paper is twofold – (i) study the IoT data transformation at various stages and (ii) describe the DataTweet IoT architecture to exploit the data driven aspects for the mentioned benefits.

The contributions of the paper are – (i) developing the data centric architecture focusing on IoT data cycle, (ii) integration of the architecture components into oneM2M standard, (iii) outlining requirements of DataTweet services and subsystems, (iv) integration of vehicular sensors as IoT data source and ITS-G5 as communication medium towards connected vehicle domain [7] and (v) adding flavor of Named Data Networking (NDN) for the data dissemination. The rest of the paper is organized as follows. Section II outlines the DataTweet services and their requirements. Section III describes the DataTweet architecture, its subsystems and their mapping into real world components. Section IV focuses on IoT data cycle and how the data is represented and transformed through the architecture elements. Section V presents the integration of DataTweet architecture elements into oneM2M standard, its prototyping and deployment details. Finally, Section VI concludes the paper.

II. DataTweet Services and Requirements

In a rather simple way, the IoT data cycle forms a closed loop where information is generated, processed and consumed, as seen in Fig. 1. From there, four high level DataTweet services can be designed, namely – (i) Data consumption service, (ii) Data collection service, (iii) Data dissemination service and (iv) Configuration management service. The interworking of the services is challenged by heterogeneity of the M2M devices that are generating different kinds of data and the mobility aspect of the devices. These challenges are identified in depth and functional requirements are derived from them in subsections below.

A. Requirements for data collection service

- The first requirement for this service is the availability of a data generation subsystem from where the IoT data will be collected.
- The service should be able to interact with the generation subsystem in a session less manner following RESTful approach [1].
- The service should be able to send a query (GET) to obtain data as well as capable of receiving data over POST and/or PUT.
- The service should be able to access generation subsystem regardless of the communication technologies of the underlying network subsystem.
- The collection service should also provide appropriate facilities for IoT data storage and processing.
- The data collection service should also support resource discovery [2] initiated by a consumer subsystem to collect data from specific M2M devices in the data generation subsystem.

B. Requirements for data dissemination service

- This service should be able to communicate data to a consumer subsystem based on defined interests.
- Depending on consumer subsystem's context, this subsystem would either (i) disseminate raw M2M data collected by data collection service or (ii) process the M2M data to generate a high level abstraction and disseminate. For M2M data processing, the service must employ a processing and storage subsystem.
- The consumer subsystem should be able to avail this service over RESTful interactions.
- This service should allow authorized consumer subsystems (as determined by access control mechanisms [13]) to subscribe to occurrence of certain events. If one of the events occur, a push notification should be sent to the consumer.

C. Requirements for configuration management service

- This requires an administrative subsystem which is in charge of adding, updating and deleting configurations of generation subsystem [3], [10].
- The service should be able to push new configurations to generation subsystem through the processing and storage subsystem since the generation subsystem are not directly exposed.
- The service should allow GET method to read the current configurations of generation subsystem.
- The service requires the administrative subsystem to perform resource discovery for configuration update.
- The service should update access control policies after configuration update.
- It should make the services available through RESTful web services facilitating session less interaction.
- The service requires the utilization of processing and storage subsystem to communicate with generation subsystem over multiple technologies.
- The service requires that the processing and storage subsystem implement a device management framework to efficiently manage the configurations and generation subsystem.

D. Requirements for data consumption service

- Data consumption service requires availability of raw or processed M2M data from processing and storage subsystem.
- The functionalities of this service should be invoked from both the consumer and administrative subsystems.
- The service must provide facilities to register for a push notification as well as consuming data over GET request.
- The service requires it to be implemented in RESTful manner to maintain interoperability with other services.

E. Requirements for network subsystem

- The network subsystem should be flexible enough to allow multiple access technologies (LTE, BLE, NFC, ITS) and protocols (HTTP, CoAP, MQTT), these being used for data exchange between the processing and storage subsystem and data generation subsystem.
- It should support data transport through RESTful interactions between subsystems. If this support is not native, DataTweet should deploy network components implementing a way to support the DataTweet architecture and its elements, while procuring the best way to communicate the RESTful interactions.

III. Data Centric Architecture for DataTweet

This section describes the DataTweet architecture (Fig. 2), its subsystems and their mapping into real world components. The proposed architecture can be broadly classified into five subsystems as follows.

A. Generation subsystem

The generation subsystem comprises of physical things that generate data. It has two main functionalities – (i) data generation and (ii) maintaining a local configuration of entities. The former provides a generic and RESTful interface to the sensors to retrieve the measurements. It is to be noted that these sensors are heterogeneous in nature and generate multimodal data. To settle the inherent heterogeneity, Sensor Markup Language (SenML) is used. SenML provides a uniform way to represent and encode the measurements along with sensor specific attributes (unit, name, ID, timestamp etc.) creating a metadata [1]. The metadata settle the heterogeneity in data generation subsystem. The local configuration contains the device and endpoint descriptions for the purpose of automatic management of generation subsystem. The descriptions are generated by the smart M2M devices themselves following the specifications outlined in the IPSO Alliance Framework. It basically extends the CoRE Link Format to represent the M2M device and endpoint resources [3].

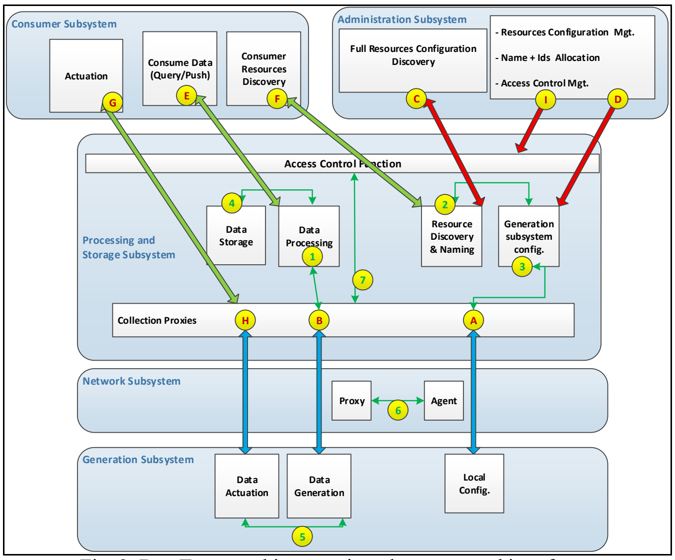

Fig. 2. DataTweet architecture, its subsystems and interfaces.

B. Network subsystem

The envisioned network subsystem presents a subtle challenge. Not all access technologies function in the same fashion, nor do they ensure us of their support for the project's requirements. Some services (IP communication, location) can be provided by the network operators and will be profited from. For fixed M2M devices and smartphones, IP communication can be used. As for vehicular environments the intelligent transport system (ITS) stack can be of service. All mechanisms hinge on the capability of the access to support RESTful communications, which require at least the presence of IP level interfaces. Natively supporting IP access technologies will not require any intervention from the presented DataTweet architecture. However, for those who may not support IP communications an alternative must be given, therefore requiring a DataTweet component be introduced in whatever underlying technology, which should still be as abstract as possible to ensure that it works upon the same principles across a range of access technologies. Along those lines, a simpler architectural approach usually adapts better to different technological environments. This is reflected in the introduction of a pair of components, the "Agent" and the "Proxy" (in Fig. 3) whose implementation specifics will be dependent upon the existing technologies. While the proxy will exist only once per access technology basis, the agent will be present among all the network devices aiming to support the DataTweet services. Either as a means of transporting local data or, if necessary, relaying any data from other agents to the upper structure of the architecture, the agent will keep a communication link with the proxy in whatever fashion the access technology allows (Fig 3, interface 6).

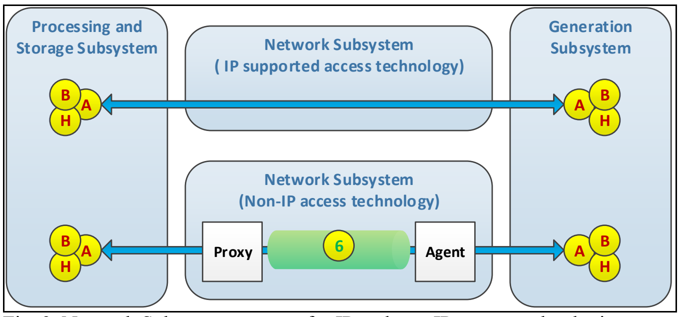

Fig. 3. Network Subsystem support for IP and non-IP access technologies.

To simplify the external interfacing of the subsystem, Network subsystem is isolated and self-contained, keeping it completely transparent (meaning no interfaces are created with other subsystems). While within the subsystem, the choice on the proxy-agent interface should be given to the management of the local technology, to promote the best means of delivery and higher efficiency. As such, the network components will work in a plug & play fashion, independently of the rest of the architecture and in a completely transparent way. They will only be deployed where they are required to support Data Tweet services over access technologies who cannot support RESTful communications.

C. Processing and storage subsystem

DataTweet services on collection and configuration management are implemented using the processing and storage subsystem. It includes functionalities of data storage, data processing & analytics, resource discovery & naming, configuration management of generation subsystem and collection proxies. This subsystem is generic and flexible enough to be deployed at a remote cloud system, an M2M gateway or even in a mobile application running in a smartphone. Further descriptions about the functionalities are given below. Although access control is a part of this subsystem, that discussion is out of scope of this work.

- **Collection proxies:** Consumer M2M devices which consist the generation subsystem may use many communication technologies (e.g. BLE, Wi-Fi). To create a generic functional architecture, it is necessary to consider IoT data exchange regardless of communication technologies. This is achieved using the collection proxies. Each technology is managed by a proxy which is further subcategorized into proxy-in (for sensors) and proxy-out (for actuators) [1]. Each proxy is developed using RESTful web service and can be accessed using an URI. The collection proxies can query the generation subsystem to obtain sensor metadata (in SenML format) or the metadata can be pushed to the proxies. Depending on the context, the metadata can be stored at the data storage for future processing or can be processed as soon as it comes.

- **Data storage:** This offers appropriate storage mechanisms. For delay tolerant network (DTN) based use cases, the metadata received by generation subsystem is stored at data storage as it is initially. For other scenarios, the data storage could be utilized to keep the high level abstractions and actionable intelligence generated by the data processing unit.

- **Data processing:** The raw measurement coming from generation subsystem does not convey much information. The first step towards intelligent processing of M2M data is to add additional attributes thereby creating a structured SenML metadata. The capabilities of SenML has been extended to use the same uniform metadata for actuators. The next step in data processing is to link the meaning of the M2M data from the point of view of the domain from which it is originating. This can be done by using semantic web technologies [4], [6]. Such processing results in high level abstraction of M2M data which on further treatment results in actionable intelligence. The intelligence could be treated as a contextual information and communicated to the end user or can be used by an M2M application to react to the situation. In either case, the derived intelligence becomes the stepping stone for actuation. Semantic web technologies could be utilized by the M2M data processing unit.

- **Resource discovery and naming:** To provide consumer centric service, the subsystems of the architecture must interwork without human intervention. They must be able to exchange metadata, manage configurations, process the information and react autonomously. To realize such vision of IoT, there must be mechanisms for discovering the resources, their capabilities & properties as well as the URIs to access them directly. Without discovery module, the consumer will have to manually configure the subsystems which will result in a static environment. Dynamic discovery can automatically learn about the M2M devices attached to the collection proxies, their configurations and capabilities etc. This module also assigns proper name to the generation subsystem. Named Data Networking is utilized for naming the processed data [14], [15].

- **Configuration management of generation subsystem:** This module is focused on automatic and efficient management of the M2M devices in generation subsystem. The deployment of IoT based solutions on a large scale is highly dependent on configuration management. However, there is a lack of unified approach towards management of huge volume of devices in generation subsystem. To solve that, this module utilizes the Open Mobile Alliance Lightweight M2M (OMA LwM2M) framework [3].

D. *Consumer subsystem*

The data consumption service is related to the consumer subsystem. It is deployed to consumer devices as a mobile application [16], [17]. From a generic perspective, its functionalities could be subdivided into three modules as mentioned below.

- **Discovery:** It provides search facilities to retrieve the URIs of required resources from processing and storage subsystem. This module triggers the discovery by sending a GET request to a well-known entry point at the "resource discovery & naming" module.

- **Consumer data:** It is accomplished by querying the URI(s) retrieved from the discovery phase. The consumer subsystem could also subscribe to a push notification [16] service from processing and storage subsystem. In that case, the later pushes the required data using a push notification framework.

- **Actuation:** Based on the intelligence received, the user may decide to react to the environment through actuators. This process could also be automated using an M2M application. For example, consider a smart vehicle driving through a foggy environment. When the vehicle deduces that there is fog in the surroundings, it could automatically switch on the fog lamps (actuation). Capabilities of SenML are extended to accommodation metadata exchange with actuators [1], [8].

E. *Administration subsystem*

This subsystem is also a part of the data consumption service but is centered on the administrative functionalities of the functional architecture. Administrative subsystem is primarily used to assign naming and addressing schemes to the other subsystems. It also determines the access control policies to be enforced at the processing and storage subsystem. To accomplish these functionalities, it needs the discovery and consume data modules which are same as that of consumer subsystem. This subsystem could be deployed at a terminal or in a mobile application.

F. *Mapping architecture elements*

Mapping of the DataTweet architecture elements with real world components is shown in Fig. 4. The consumer subsystem is aimed for the end user of the Data Tweet services. Nowadays services are consumed and accessed in several ways and through different platforms. Therefore, we chose to represent them in the current most popular ways such as a smartphone application and a web-interfaced application, which contain all the subsystem's functionalities. Processing and storage require resources and is deployed at cloud platforms. But to promote a distributed architecture with scalability, robustness and high QoS mobile edge computing [5], [19] can be considered. For the administrative subsystem, a better option can be to centralize its components to a few back end servers, thus having its management policies and access control information stored in a secured network. As for the Network subsystem, this is far more particular as its components perform tasks that are more susceptible to location. For instance, the network proxy is better deployed as a network application in a gateway style equipment (ITS RSU, LTE/LTE-A base station), to simplify its task of interfacing with the access technology and open communication links with the network agents. Agents which are distributed along the end user equipments, such as vehicles and smartphones, also in the form of a network application. Lastly, the Generation subsystem resides inside of the user's equipment, being composed only by the generating sensors, if they have the capability of transmitting the generated data, or otherwise as an application that interfaces directly with the generation sensors and manages them along with their data.

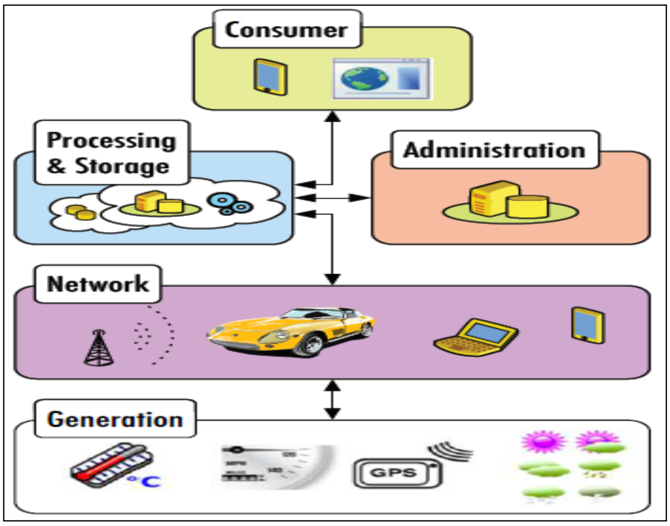

Fig. 4. Mapping of DataTweet elements into real world components.

IV. IoT Data Transformation

The IoT data goes through a series of transformation from generation to processing, storage and finally dissemination. This section describes the IoT data lifecycle. At the generation subsystem, the data are encoded using SenML which creates a metadata. For example, below is body temperature sensor.

{"e":[{"n":"BT-1", "v": 41, "u": "Cel", "t": "1380897199", "ver": "1.0", "type": "Temperature", "domain": "eHealth"}]]

The metadata is encoded using JSON. The body temperature sensor is named "BT-1" and generates a value of 41 degrees Celsius. The time stamp is calculated with respect to 16/01/1970 AD at 23:34:57 UTC. The type of measurement is temperature and the sensor belongs to eHealth domain. The version of software being used is 1.0. On the other hand, the configuration descriptions of generation subsystem are represented using CoRE Link Format [20] and has also been encoded using JSON. The M2M description could be consisting of a name, id, model, location and destination (URI of the proxy to which it is attached):

{"id": "G7612d", "model": "XYZ", "location": "Room313", "destination": "http://IP_addr:Port/313/AC-1", "name": "AC-1"}

Similarly, the M2M endpoint description consists of name & id of the endpoint, name of the M2M device to which it is attached and type of the endpoint.

{"id": "2wsxzaq1", "device": "AC-1", "name": "AirConditioner_313", "type": "Actuator"}

The collection proxies feed the SenML formatted M2M data to the data processing unit which generates a high level abstraction and it is stored in a data storage. From the above example, the processing unit analyses the temperature measurement of 41 degrees Celsius which is coming from a body temperature sensor in eHealth domain. This allows the deduction of fever which is a high level abstraction of the data generated by the sensor. The new intelligence is stored the data storage along with a timestamp as well as can be pushed to the consumer subsystem as a part of data consumption service.

The consumer subsystem understands the knowledge and based on preconfigured rules, initiate actuation of a room temperature controller to set a comfortable temperature. This command is also communicated using SenML extensions and could be encoded using XML as shown below.

PUT http://IP_addr:Port/313/AC-1
<senml bn= AirConditioner_313>
<e n="Temp-1" t="0" v="20" u="Cel" />
</senml>

V. Integration into oneM2M Architecture

DataTweet IoT architecture has been integrated into oneM2M architecture (Fig. 5) to provide interoperability and scalability. Details of oneM2M itself have been previously discussed at [8].

The generation subsystem corresponds to the sensors, actuators embedded into different devices like smartphone, vehicles. The architecture is independent of the underlying networking subsystem. This makes the DataTweet services run independently of communication technologies and protocols. The processing and storage subsystem can be deployed at the cloud (infrastructure node) as well as M2M gateways or LTE/LTE-A base stations (middle node). The consumer and administration subsystems are incorporated as application service nodes analogous to consumer smart devices (smartphones, tablets). The functional entities representing discovery, management, access control etc. are parts of the Common Service Entities (CSE) in oneM2M. A prototype of the system is developed. The generation subsystem refers to a vehicle equipped with several on-board sensors and an on-board unit (OBU) with ITS-G5 stack. The OBU has software agents (written in C language) for data generation and local configuration management. To use ITS-G5 technology, we need an intermediate node for protocol translation. The middle node collection proxy is attached to a road side unit (RSU) which relays the sensor metadata to the collection proxy. It basically extracts the metadata from ITS-G5 and then encapsulates that into IP packets. The middle node (i) also acts as a storage and processing unit, (ii) is developed as a collection of RESTful web services and (iii) is written using Python based Flask framework. Such M2M gateways also promote a distributed computing platform. The cloud system (infrastructure node) acts as the central processing and storage subsystem. It houses the Machine-to-Machine Measurement (M3) Framework [4], a data processing engine (written in Apache Jena Framework) that utilizes semantic web technologies to generate high level abstraction and actionable intelligence from raw vehicular sensor metadata. The consumer subsystem is equipped with a mobile Android application called Connect and Control Things

(CCT) [16], [17]. CCT allows expressing consumer requirements in terms of NDN interests which are routed to the cloud systems and data is showered according to the availability.

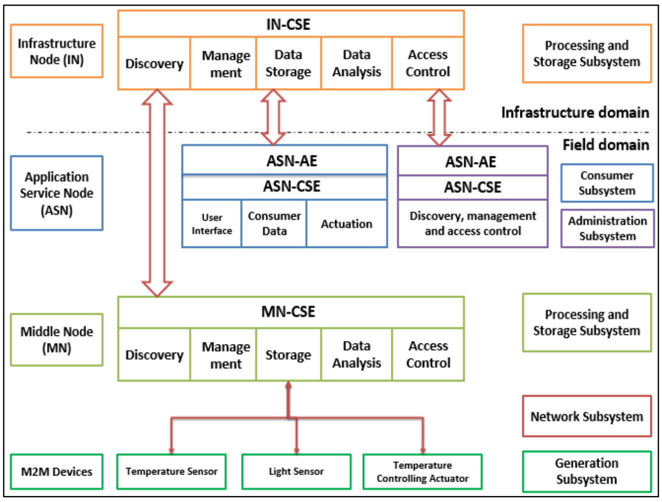

Fig. 4. Integration of DataTweet architecture into oneM2M standard.

VI. Conclusion

In a nutshell, this paper looks into the IoT data cycle and discusses an architecture to enable data centric IoT services. The DataTweet architecture draws analogy from the water cycle and is broadly classified into four main services. The requirements for each are analyzed. The four subsystems of the architecture are presented in details and they are mapped to the real world components. The common service functions in the subsystems are described. The transformation of IoT data at various architecture elements are examined. To provide consumer interoperability, DataTweet services and subsystems are integrated into the oneM2M architecture. As for future work, we are working on data dissemination aspects utilizing NDN based interest and naming of data.

Acknowledgment

This work is supported by French research project DataTweet (ANR-13-INFR-0008). The authors thank Amelie Gyrard for her work on design and development of M3 Framework.

References

[1] Datta, S.K.; Bonnet, C.; Nikaein, N., "An IoT gateway centric architecture to provide novel M2M services," in Internet of Things (WF-IoT), 2014 IEEE World Forum on, pp.514-519, 6-8 March 2014.

[2] Datta, S.K.; Da Costa, R.P.F.; Bonnet, C., "Resource discovery in Internet of Things: Current trends and future standardization aspects," in Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on, pp.542-547, 14-16 Dec. 2015.

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

[4] Gyrard, A.; Datta, S.K.; Bonnet, C.; Boudaoud, K., "Cross-Domain Internet of Things Application Development: M3 Framework and Evaluation," in Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on, pp.9-16, 24-26 Aug. 2015.

[5] Datta, S.K.; Bonnet, C.; Haerri, J., "Fog Computing architecture to enable consumer centric Internet of Things services," in Consumer Electronics (ISCE), 2015 IEEE International Symposium on, pp.1-2, 24-26 June 2015.

[6] Gyrard, A.; Datta, S.K.; Bonnet, C.; Boudaoud, K., "A Semantic Engine for Internet of Things: Cloud, Mobile Devices and Gateways," in Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2015 9th International Conference on, pp.336-341, 8-10 July 2015.

[7] Ning Lu; Nan Cheng; Ning Zhang; Xuemin Shen; Mark, J.W., "Connected Vehicles: Solutions and Challenges," in Internet of Things Journal, IEEE , vol.1, no.4, pp.289-299, Aug. 2014.

[8] Datta, S.K.; Gyrard, A.; Bonnet, C.; Boudaoud, K., "oneM2M Architecture Based User Centric IoT Application Development," in Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on, pp.100-107, 24-26 Aug. 2015.

[9] Eckhoff, D.; Sofra, N.; German, R., "A performance study of cooperative awareness in ETSI ITS G5 and IEEE WAVE," in Wireless On-demand Network Systems and Services (WONS), 2013 10th Annual Conference on, pp.196-200, 18-20 March 2013.

[10] Perumal, Thinagaran; Datta, Soumya Kanti; Bonnet, Christian, "IoT device management framework for smart home scenarios," in Consumer Electronics (GCCE), 2015 IEEE 4th Global Conference on, pp.54-55, 27-30 Oct. 2015.

[11] Zanella, A.; Bui, N.; Castellani, A.; Vangelista, L.; Zorzi, M., "Internet of Things for Smart Cities," in Internet of Things Journal, IEEE , vol.1, no.1, pp.22-32, Feb. 2014.

[12] Bin Guo; Zhiwen Yu; Xingshe Zhou; Daqing Zhang, "From participatory sensing to Mobile Crowd Sensing," in Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference on, pp.593-598, 24-28 March 2014.

[13] Gusmeroli, S.; Piccione, S.; Rotondi, D., "IoT Access Control Issues: A Capability Based Approach," in Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on, pp.787-792, 4-6 July 2012.

[14] Amadeo, M.; Campolo, C.; Iera, A.; Molinaro, A., "Named data networking for IoT: An architectural perspective," in Networks and Communications (EuCNC), 2014 European Conference on, pp.1-5, 23-26 June 2014.

[15] Hail, M.A.; Amadeo, M.; Molinaro, A.; Fischer, S., "Caching in Named Data Networking for the wireless Internet of Things," in Recent Advances in Internet of Things (RIoT), 2015 International Conference on, pp.1-6, 7-9 April 2015.

[16] S. K. Datta, C. Bonnet and N. Nikaein, "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, Singapore, 2014, pp. 1-6.

[17] S. K. Datta and C. Bonnet, "Connect and Control Things: Integrating Lightweight IoT Framework into a Mobile Application," Next Generation Mobile Applications, Services and Technologies, 2015 9th International Conference on, Cambridge, 2015, pp. 66-71.

[18] R. Fantacci, T. Pecorella, R. Viti and C. Carlini, "Short paper: Overcoming IoT fragmentation through standard gateway architecture," Internet of Things (WF-IoT), 2014 IEEE World Forum on, Seoul, 2014, pp. 181-182.

[19] I. Stojmenovic and S. Wen, "The Fog computing paradigm: Scenarios and security issues," Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on, Warsaw, 2014, pp. 1-8.

[20] S. K. Datta and C. Bonnet, "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, Taipei, 2014, pp. 61-68.