

An Architectural Model for Fog Computing

Jabril Abdelaziz^a, Mehdi Adda^b*, Hamid Mcheick^a

^a *Mathematics and Computer Science Dep. University of Quebec at Chicoutimi, Chicoutimi (Qc), Canada*

^b *Mathematics, Computer Science and Engineering Dep. University of Quebec at Rimouski, Rimouski (Qc), Canada*

Abstract

The adoption of the Internet of Things raises many challenges. A variety of its applications require widespread distribution and high mobility support. In addition to low latency and real time services. To meet these challenges, the Fog Computing is arguably a suitable solution to leverage the Internet of Things with such requirements. Indeed, we believe that the nearness of Fog nodes to the edge of the network provides an environment for critical preemptive and proactive applications and services (e.g., predicting natural disasters). Thus, this paper proposes an architectural model for Fog Computing. First, it presents a middleware to abstract the underlying devices and to unify the sensed data. Second, it describes an Operational Layer intended for service presentation, management and transformation. An environment embracing such model will provide means for early data analysis, hence low latency and real time responses. In addition, to providing an ecosystem for direct collaboration between services leading to more sophisticated applications. A flood warning system exemplifies a use case scenario to illustrate the potential adaption and application of the presented model.

Keywords: *fog computing, service aggregation, internet of things.*

1. Introduction

The Internet of Things (IoT) is paving the road to a future where autonomous objects sense, actuate, interact and react with less to no-human interaction. That is, the human part in machine communication is blurring into a more sophisticated thing-to-thing communication model. In this model, things will search for other things and provide collaboration-based and aggregated services. Thus, leading on the one hand, to more sophisticated with added value applications, on the other hand, to even more complex issues and new implementation challenges.

In the Internet of things, sensors are used to collect data from their environment. The collected data are usually sent to the cloud for processing and analysis. In fact, recent trends in IoT are mostly toward a centralized cloud-based architecture. Such approach comes with many drawbacks related to mobility support, distribution, and context awareness [1]. In addition, the continuous torrent of heterogeneous and potentially irrelevant data comprises challenging tasks of filtration, homogenization, and data analysis speed. Such challenges, in consequence, increase the latency of data transmission processes and real-time responses.

In the Internet of Things, prior to the decision-making and the wisdom generation level, systems must come up with solutions

to allow things to discover the functionalities of other things, and to be able to use them efficiently, securely, and with minimal human intervention [2]. We foresee the ability of things to collaborate at the edge of the network as an opportunity to support highly heterogeneous systems, and to leverage the gain from the provided services and the sensing activities [3].

The edge computing is not a new idea, rather a convergence of experiences in cloud and mobile computing [4]. We have seen the application of such approach in mobile computing as, for example, Cloudlet [5]. Indeed, to avoid sending entirely all sensed data to the cloud, such approaches build applications on the edge of the network to handle mainly data gathering. For instance, the Edge-Centric Computing [6] merges the peer-to-peer and Cloud Computing paradigms, and this by running specific tasks and store data in a cloud-like server at the edge of the mobile network. However, this work focuses only on human-operated devices, therefore narrowing its application extent on the Internet of things. A broader approach has been early presented by Cisco Systems [7], [8] named Fog Computing.

The idea behind the Fog Computing is intellectually and technically seductive. In fact, an intelligent layer is formed between sensors and the Cloud, potentially dealing with communication, computing and access management. Fogs, the nodes in such layer, are to support mobility, real-time data analysis and decision-making processes.

* Corresponding author. Tel.: +14187231986 #1850

Fax: +14187241879; E-mail: Mehdi_adda@uqar.ca

© 2018 International Association for Sharing Knowledge and Sustainability.

DOI: 10.5383/JUSPN.10.01.003

Our interest lies in extending this new paradigm to embrace a thing-collaborative computing model. In such model, things could be enabled to exploit and collaborate with each other to achieve common or distinct goals. However, such an early collaboration increases even more the complexity of interactions in this model [9]. This increasing complexity is due to the huge number, the heterogeneous and the dynamic nature of the things involved. The different technologies of both offered and consumed services add to this complexity [10].

Therefore, our work focuses on the early interaction and collaboration between things. Indeed, we foresee the aggregation of sensing activities, in the Fogs at the edge of the network, as an opportunity to support more advanced collaboration scenarios. The presented multilayer architecture abstracts the Internet of Things hardware, helps translate, describe and format the sensed data, in order to deliver high abstracted static and dynamic services; such services constitute the building blocks of the collaborative perspective of Fog Computing.

The next section discusses related work. We briefly review existing Internet of Things middleware solutions, to concentrate later on Fog Computing. Section 3 describes the main layers of our model. To exemplify the use of such model, section 4 illustrates and describes a use case of a flood warning system. The last section, concludes this paper and expresses our future research lines.

2. Related work

By nature, the Internet of Things is a dynamic environment with a hard to know topology. This nature accentuates more the challenges of the network and computing systems such as heterogeneity, interoperability, scalability, security and so forth. To tackle such challenges, various middleware-driven solutions have been proposed over the years. Solutions such as GSN [11], SENSEI [12], iCore [13], [14], SOCRADES [15], COBIS [16], Smart-M3 [17], WhereX [18] focus mainly on devices or data abstraction and are suitable for moderate service interactions [19]. However, IoT demands a more intense interaction and coordination between thing services. Besides, the virtualization of data and devices is commonly supported through the definition of virtual sensors. This mechanism of virtualization can be exploited to aggregate many physical devices or data streams under one virtual composed service as it is the case for Linksmart [20] (formerly named Hydra [21]) and SATware [22] respectively. Although this mechanism could support dynamic composition of services, neither the sensors to be aggregated nor the composed services are identified at runtime. They are indeed preconfigured.

Dynamic composition of things provided services is among the building blocks of the future IoT. This need for aggregating services has also led to the creation of aggregators like , for example, the “IF This Than That” (IFTTT) [23], Samsung SmartThings [24] and the Thing System [25]. Mainly designed for smart-home applications, such solutions provide software, hardware or both to mitigate the task of gathering thing services under one application. Although, the ease of use granted a wide adoption of these solutions, they still proprietary with a limited set of supported devices and services. This could result in interoperability and integration related issues for new IoT products.

Many of the aims at migrating data processing and analysis to the edge devices. It extends the cloud by pushing computing to the edge of the network, and allowing generic applications and services

to run directly on the edge resources. This new vision is motivated by the gain of low-latency and a better support of real-time processing and data analysis applications.

We believe that a Fog-based collaborative model will support highly heterogeneous systems, and enhances the quality of the Internet of Things applications. However, the status quo has yet to provide real time and collaboration enabling services. Some promising works, that we believe complementary to ours, have been presented in this direction. The Mobile Fog [10] presented a high-level programming model for the fog computing. This model is intended for latency-sensitive and on-demand scaling applications, but a more general approach is needed to deal with resource mobility. Filtering and unifying data are a principal issue in order to provide a middleware in the Fog environment. The use and presentation of data should adapt to the context of the demand [27]. For such issues, a goal-driven and context-aware data filtering approach have been the focus of the system in [28]. Though, in case of an aggregated or time-dependent sensing activity, issue like mobility support may arise. The composition of thing services has also gained interest. The model proposed in [29] uses artificial potential fields to deliver a decentralized service composition. Which is claimed to be suitable for local interaction. It is worth noting, however, that the selection of services assumes an in-advance standard and atomic services. Intelligence is moving to the edge of the network with many rising challenges and opportunities. We intend to bring collaboration to the edge of the network following the vision of fog computing. First, by providing a middleware to abstract the underlying heterogeneity based on the success of previous works, and second, by modeling the aggregation of thing services in order to support more elaborated collaborative applications.

3. Model Architecture

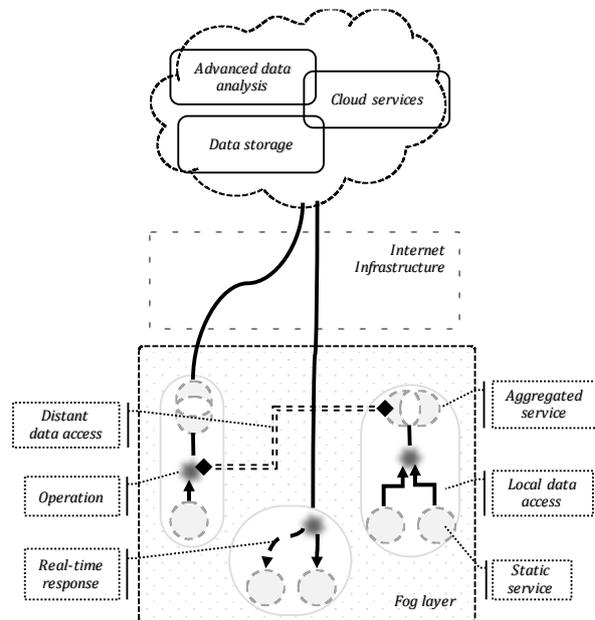


Fig. 1. The interaction the fog and the cloud as a nervous system.

We like to picture the relation between Fogs and the Cloud as the human nervous system. In the human nervous system, the sensory inputs are sent to the brain in the Central Nervous System (CNS). Received information is integrated and activity of all the body parts is coordinated accordingly. Although,

most of the sophisticated sensory processing takes place in the brain, feature detection, complex and quick responses also occur in peripheral sensory organs (e.g. eyes). Information from a sensory input are transmitted to the spinal cord and projected out to the motor neurons. Resulting in quick and instantaneous reactions, called reflex actions. These reflexes are the simplest type of neural circuit, yet they are of high degree of importance.

Similarly, devices in the Internet of things capture and send data to the Cloud (the brain) for analysis. Nevertheless, many of the data processing and analysis could happen near the peripheral devices, that is, the Fog. Thus, providing real-time responses (reflexes) in certain cases. Figure 1 depicts the Fog-Cloud interaction from this point of view.

The Internet of Things needs the ability of extracting relevant information from combinations of sensory inputs at the Fog level. Thus, to ensure the robustness, the maintainability, and to ease the task of such mechanisms, our scheme is to model such environment with high-level of abstraction, interfacing and interoperability.

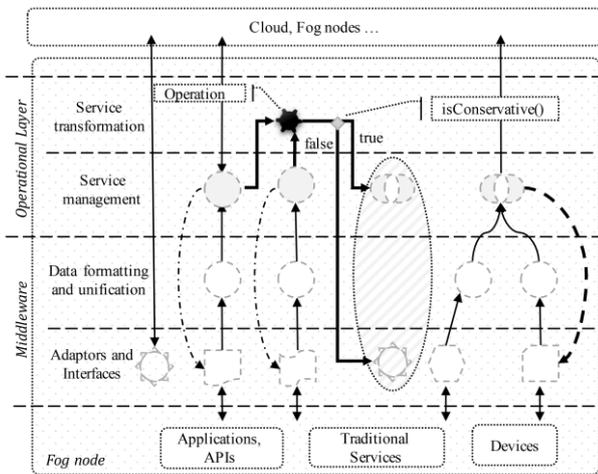


Fig. 2. Model for the Fog Service Aggregation.

As shown in Figure 2, we distinguish two main layers: The Operational Layer and the Middleware.

3.1. The Middleware

The Internet of Things is a nest for a huge number of heterogeneous devices and a source of huge amounts of data. The underlying swarm of sources of data comprises connected things spanning from physical sensors, actuators, controllers to virtual objects, APIs and classical web services.

From a service-developer perspective, there is a need for data sources abstraction so no prior knowledge about embedded systems will be required, which is guaranteed via the Middleware. As depicted in the above figure (Fig. 2), this middleware comprises three layers; the Adaptation, the Filtering, and the Unification layer.

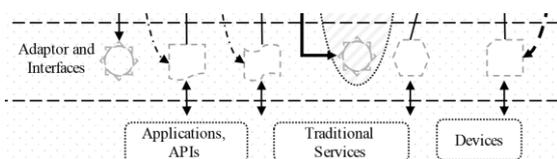


Fig. 3. The Adaptation Layer provides an abstraction of data sources.

The Adaptation layer (fig. 3) grants an abstract interfacing with the underlying data sources. This layer i) provides generic means to describe sensors and virtual objects and their configurations, ii) allows the definition of Connectors and Adapters each of which is able to handle communication with known and future coming things, and iii) handles the life cycle of data sources. By joining or leaving the network, a given object must not alter other independent services.

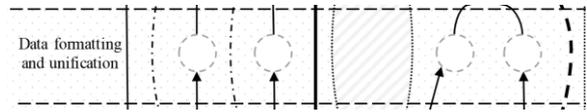


Fig. 4. Data formatting and unification layers

The Formatting and Unification layer (fig. 4) is responsible for delivering information description methods and data filtering mechanisms. In addition, it offers a unified and homogeneous view aiming the standardization of the filtered data. The resulting data are consumed through Unitary Services. Thus, inheritably loose coupled and discoverable.

3.2. Operations and Service Management

Overlaying the first three layers is the Operational layer (fig. 5). The idea is to deliver a set of different, yet homogenous services. A service is a faithful representation of one or many entities. We use the term Entity to refer either to a data source (sensor, API, web service) or data consumer (actuator, service). Thus, a service must comprise mainly a set of identifiers, capabilities and commands.

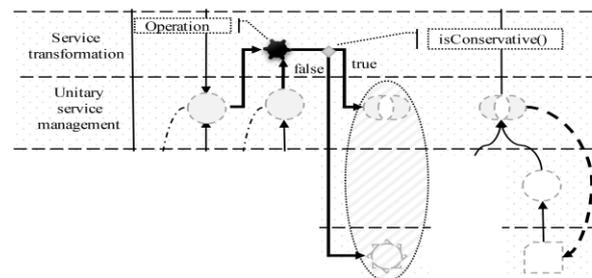


Fig. 5. The Operational Layer and service Operations.

Furthermore, a set of operations is defined in this layer to be applied on services. An *Operation* is any transformation, aggregation, composition, etc. of one or many services. The layer provides the necessary abstraction of the internal workflow of a given operation and the description of both input and output of each operation.

For instance, a threshold operation triggers a direct communication with the data source adaptor depending on predefined limit. The output of such operation might be a *reflexive response*, a notion we introduce to allow real-time responses in the Fog node.

We distinguish between two kinds of operations: conservative and non-conservative operations. A conservative operation is simply any operation for which the result belongs to the set of already defined data format. In contrast, a non-conservative operation results in a new data format. The new obtained data is presented seemingly as a new virtual data sources. Consequently, leading to the application of the data formatting processes.

4. Application Scenario

We are exploring the use of Fog Computing applications in the area of natural disaster, especially the case of flood warning systems. Indeed, using a variety of forecast models, it is commonly possible to predict, with a certain accuracy, the time-slot where a flood could happen. In addition to the probability of its occurrence. Upon a flood-warning based on such forecast, authorities proceed gradually to some actions in order to reduce risks and the likelihood of damages. Forecasts are inherently uncertain; above all, we are predicting the future. Hence, the mobilization of resources (human, logistic, etc.) would be a waste in case of “false” flood warning. A real-time perception and interaction with our environment will be of great value. Through Fog Computing, we can augment the accuracy of forecasts and warnings. Figure 6 illustrates a simplified view of a flood warning systems following the scheme of the proposed model.

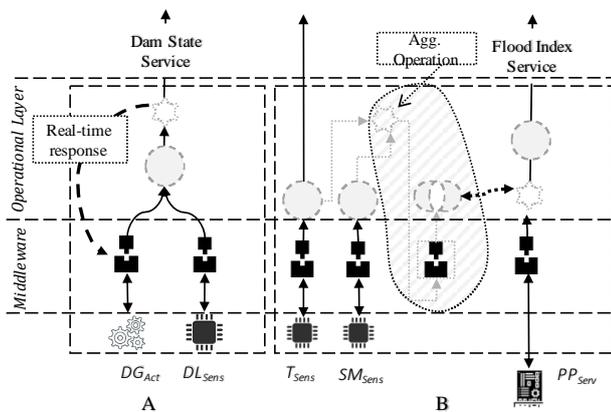


Fig. 6. Schematic representation of a flood warning system

In such scenario, a set of sensors and actuators could be deployed and distributed at the level of a Drainage Basin. The deployed devices can, for example, measure precipitation, river levels, local temperature, monitor the soil moisture and so forth. For the sake of simplicity, we consider four type of devices: Dam Level Sensor(DLSens), Dam Gate Actuator(DLAct), Temperature Sensor (TSens), and a Soil Moisture Sensor(SMSens). In addition to a traditional web service that provides the forecast of precipitation(PPSrv).

In order to prototype this application scenario, a simulation set is being put in place using:

- Arduino Uno devices as a data source for the sensing activities required for the dam level, temperature, and the soil moisture measurements.
- an LCD screen to simulate the state of the floodgate,
- the precipitation forecast is received as JSON data using the YQL query language from the Yahoo Weather services.

The runtime implementation is put in place using a Websocket-based RESTful server in Java language. Such an implementation provides an easy access to the Fog node sensing activity. Indeed, to monitor the activity of the upstream of data, a web client build with Javascript language is also provided.

4.1. A Threshold Operation

Figure 6(A) illustrates the threshold operation responsible of the floodgate. Both information from the DLSens sensor and the DLAct actuator are captured and formatted using serial

communication adapter with the devices. The captured data are delivered and streamed up from the Fog as a single service: The Dam State Service.

Regulating the flow from the floodgate does not need sophisticated analysis. Yet, a quick reaction is vital. Data from the Dam State service are subjected to the checkState() operation. Based on a threshold, the output from such operation may trigger the floodgate to change its state (open/close). As the floodgate actuator has been already virtualized, data representing its state are well known the system. By consequence, there will be no need to reformat or unify the result output. We call the checkState() operation a Conservative operation.

4.2. An Aggregation Operation

Using data from the Temperature T_{Sens} and the Soil Moisture SM_{Sens} Sensor, the Evapotranspiration index could be estimated.

Thus, an aggregation operation is defined to dynamically aggregate data from these sensors, we call it $evapOp()$. Since the output data of such operation has no prior description in the model, a processes of formatting and filtering is needed, flowed by the definition of a new service to deliver such data. Thus, the operation is considered as non-conservative (Fig. 6(B)).

For the sake of simplicity, we use the soil moisture, the evapotranspiration, and the rain precipitation to deduce a Flood Risk index value. Data from the traditional service of precipitation forecast PP_{Srv} need to be formatted. The Flood Index is defined to deliver data generated by the deduction operation illustrated with Figure 6(B).

5. Discussion and Future Work

Many solutions were proposed for the Internet of Things, yet few are designed for Fog Computing environments. In this paper, we discussed our vision of a model for Fog Computing. However, the case here is not about the middleware, rather the Operational Layer and the collaborative support it bestows for such environment.

This work is an ongoing research project, which opens doors for interesting challenges:

Service operation: the model needs a more formalized and complete definition of the set of operations to be performed on services: aggregation, threshold, data transformation, etc.

Runtime implementation: implementing such model on a Fog-enabled device is a tricky task. Key challenges reside in reliability, security and development of lightweight environment.

Thing collaboration: to support a dynamic and autonomous thing collaboration, we concentrate on two core research contributions: discovery and context awareness.

Acknowledgments

This work is supported by the Department of Mathematics and Computer Science at the University of Quebec at Chicoutimi (UQAC) and the University of Quebec at Rimouski (UQAR), Quebec, Canada.

References

- [1] S. Yi, C. Li, and Q. Li, “A Survey of Fog Computing,” in *Proceedings of the 2015 Workshop on Mobile Big Data - Mobidata '15*, 2015, pp. 37–42. <https://doi.org/10.1145/2757384.2757397>

- [2] G. Kortuem, F. Kawsar, D. Fitton, and V. Sundramoorthy, "Smart objects as building blocks for the Internet of things," *IEEE Internet Comput.*, vol. 14, no. 1, pp. 44–51, Jan. 2010. <https://doi.org/10.1109/MIC.2009.143>
- [3] B. Ottenwalder, B. Koldehofe, K. Rothermel, and U. Ramachandran, "MigCEP: Operator Migration for Mobility Driven Distributed Complex Event Processing," in *Proceedings of the 7th ACM international conference on Distributed event-based systems - DEBS '13*, 2013, pp. 183–183. <https://doi.org/10.1145/2488222.2488265>
- [4] D. Linthicum, "Responsive Data Architecture for the Internet of Things," *Computer*, vol. 49, no. 10, pp. 72–75, 2016. <https://doi.org/10.1109/MC.2016.302>
- [5] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The Case for VM-Based Cloudlets in Mobile Computing," *IEEE Pervasive Comput.*, vol. 8, no. 4, pp. 14–23, Oct. 2009. <https://doi.org/10.1109/MPRV.2009.82>
- [6] P. Garcia Lopez et al., "Edge-centric Computing: Vision and Challenges," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 5, pp. 37–42, Sep. 2015. <https://doi.org/10.1145/2831347.2831354>
- [7] F. Bonomi, "Connected vehicles, the internet of things and fog computing," presented at the VANET, 2011.
- [8] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing - MCC '12*, 2012, pp. 13–13. <https://doi.org/10.1145/2342509.2342513>
- [9] I. Stojmenovic and S. Wen, "The Fog Computing Paradigm: Scenarios and Security Issues," 2014, pp. 1–8.
- [10] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwalder, and B. Koldehofe, "Mobile Fog: A Programming Model for Large-Scale Applications on the Internet of Things," in *Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing - MCC '13*, 2013, pp. 15–15. <https://doi.org/10.1145/2491266.2491270>
- [11] K. Aberer, M. Hauswirth, and A. Salehi, "Infrastructure for Data Processing in Large-Scale Interconnected Sensor Networks," in *2007 International Conference on Mobile Data Management*, 2007, pp. 198–205. <https://doi.org/10.1109/MDM.2007.36>
- [12] M. Presser, P. M. Barnaghi, M. Eurich, and C. Villalonga, "The SENSEI project: integrating the physical world with the digital world of the network of the future," *IEEE Commun. Mag.*, vol. 47, no. 4, pp. 1–4, Apr. 2009. <https://doi.org/10.1109/MCOM.2009.4907403>
- [13] TCS, ALU, UPRC, SIEMENS, VTT, CNET, M3S, SAG, AMBIENT, UNIS, ATOS, "iCore, Final architecture reference Model," 2014.
- [14] "Internet Connected Objects for Reconfigurable Ecosystem Project." [Online]. Available: <http://www.iot-icore.eu>. [Accessed: 13-Oct-2016].
- [15] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, and D. Savio, "Interacting with the SOA-Based Internet of Things: Discovery, Query, Selection, and On-Demand Provisioning of Web Services," *IEEE Trans. Serv. Comput.*, vol. 3, no. 3, pp. 223–235, Jul. 2010. <https://doi.org/10.1109/TSC.2010.3>
- [16] C. Decker et al., "Collaborative business items," in *2007 3rd IET International Conference on Intelligent Environments*, 2007, pp. 40–47. <https://doi.org/10.1049/cp:20070345>
- [17] J. Honkola, H. Laine, R. Brown, and O. Tyrkko, "Smart-M3 Information Sharing Platform," in *Proceedings of the The IEEE Symposium on Computers and Communications*, Washington, DC, USA, 2010, pp. 1041–1046. <https://doi.org/10.1109/ISCC.2010.5546642>
- [18] A. Puliafito, A. Cucinotta, A. L. Minnola, and A. Zaia, "Making the Internet of Things a Reality: The WhereX Solution," in *The Internet of Things: 20th Tyrrhenian Workshop on Digital Communications*, D. Giusto, A. Iera, G. Morabito, and L. Atzori, Eds. New York, NY: Springer New York, 2010, pp. 99–108. https://doi.org/10.1007/978-1-4419-1674-7_10
- [19] T. Teixeira, S. Hachem, V. Issarny, and N. Georgantas, "Service Oriented Middleware for the Internet of Things: A Perspective," in *Proceedings of the 4th European Conference on Towards a Service-based Internet*, Berlin, Heidelberg, 2011, pp. 220–229. https://doi.org/10.1007/978-3-642-24755-2_21
- [20] "LinkSmart® Open Source Middleware Platform," LinkSmart®. [Online]. Available: <http://www.linksmart.eu>. [Accessed: 20-Dec-2016].
- [21] M. Eisenhauer, P. Rosengren, and P. Antolin, "A Development Platform for Integrating Wireless Devices and Sensors into Ambient Intelligence Systems," in *2009 6th IEEE Annual Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks Workshops*, 2009, pp. 1–3. <https://doi.org/10.1109/SAHCNW.2009.5172913>
- [22] D. Massaguer, B. Hore, M. H. Diallo, S. Mehrotra, and N. Venkatasubramanian, "Middleware for Pervasive Spaces: Balancing Privacy and Utility," in *Middleware 2009: ACM/IFIP/USENIX, 10th International Middleware Conference, Urbana, IL, USA, November 30 – December 4, 2009. Proceedings*, J. M. Bacon and B. F. Cooper, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 247–267. https://doi.org/10.1007/978-3-642-10445-9_13
- [23] "IF This Than That," *IFTTT*. [Online]. Available: <https://ifttt.com/>. [Accessed: 19-Dec-2016].
- [24] "Samsung SmartThings." [Online]. Available: <https://www.smartthings.com/>. [Accessed: 03-Jan-2017].
- [25] "The Thing System." [Online]. Available: <http://thethingsystem.com/>. [Accessed: 20-Aug-2016].
- [26] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wirel. Commun. Mob. Comput.*, vol. 13, no. 18, pp. 1587–1611, 2013. <https://doi.org/10.1002/wcm.1203>
- [27] C. Perera, P. P. Jayaraman, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "MOSDEN: An Internet of Things Middleware for Resource Constrained Mobile Devices," in *Proceedings of the 2014 47th Hawaii International Conference on System Sciences*, 2014, pp. 1053–1062. <https://doi.org/10.1109/HICSS.2014.137>
- [28] N. Narendra, K. Ponnalagu, A. Ghose, and S. Tamilselvam, "Goal-Driven Context-Aware Data Filtering in IoT-Based Systems," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, 2015, pp. 2172–2179. <https://doi.org/10.1109/ITSC.2015.351>
- [29] E. Rapti, A. Karageorgos, and V. C. Gerogiannis, "Decentralised Service Composition using Potential Fields in Internet of Things Applications," *Procedia Comput. Sci.*, vol. 52, pp. 700–706, 2015. <https://doi.org/10.1016/j.procs.2015.05.079>