Discovery-Driven Service Oriented IoT Architecture

Dimitrios Georgakopoulos and Prem Prakash Jayaraman
School of Computer Science and IT
RMIT University
Melbourne, Australia
dimitrios.georgakopoulos/
prem.jayaraman@rmit.edu.au

Miranda Zhang
Research School of Computer Science
Australian National University
Canberra, Australia
miranda.zhang@anu.edu.au

Rajiv Ranjan
School of Computer Science
Newcastle University,
United Kingdom
raj.ranjan@ncl.ac.uk

Abstract— **The Internet of Things (IoT) ecosystem is growing at a staggering pace. Each day, we are witnessing the emergence of new devices, smart phones, cameras and sensors that are connected to the internet. It is envisioned IoT will discover, integrate and exploit such devices and their data in the development of new services and products that can change and positively impact our lives. However, the core IoT functionality (such as discovery and integration) required to develop IoT service and products need to be developed to better support IoT application development. In this paper, we present a vision of a future IoT system architecture that is driven by service discovery across every layer of IoT. This includes on demand discovery and integration of devices, cloud storage and computing resources, as well as existing data analysis, visualisation and application integration services that can be dynamically selected and orchestrated as needed to create IoT applications. We provide descriptions of specific solutions that we are investigating at each of the IoT layer providing core functionalities for service-based discovery and integration.**

Keywords—Internet of Things, Service Oriented Architecture

I. INTRODUCTION

The next evolution of the Internet is the Internet of Things and Services where smart devices will communicate and consult with one another without any human intervention. Every reader reading this article will be aware of the explosive growth of Internet of Things (IoT) that involves (i) incorporating billions of internet-connected sensors, cameras, displays, smart phones, and other smart communicating devices, (which are collectively referred to as IoT 'things' [1, 2]), and (ii) harnessing their data and functionality to provide novel smart services and products that benefit enterprises, industries, and our society. The "things" have identities and virtual personalities operating in smart spaces using intelligent interfaces to connect and communicate within social, environment, and user contexts [3]. New things (sensors and devices such as smart phones) are evolving rapidly and producing data that is accessible via the Internet. Such IoT devices are responsible for exponential growth of internet data, so a grand IoT challenge is the development of technologies for analysing IoT data to distil high value information in real time.

Cloud computing has proven to be the de facto standard for delivering internet-based application services and in particular supporting IoT applications and services. At the application layer data analysis, visualisation, and orchestration services will continue to be developed and existing ones reused in the development of IoT applications. These trends will all come together in the future giving rise to a much more diversified IoT software infrastructure. In particular, we envision that the IoT universe will be comprised by a diverse set of IoT devices, cloud services, and software services that are owned, administered and operated by independent providers. These will be the ingredients of future IoT applications that power corresponding IoT services and products. Therefore, IoT will be a federated system where things and data, cloud resources, and software (e.g., for data analysis and visualisation) will be provided by independent providers with diverse interfaces, as well as business, cost, and QoS models. For example, an irrigation application that dynamically adjust irrigation patterns and water usage may fetch data from a weather station owned by a city council and use analytics service operated by a third-party provider. Such applications will require dynamic mash-ups of things and services integrated across the IoT layers using the cloud computing infrastructure. This leads to the most important and demanding challenge, i.e., how to discover and integrate IoT devices and data, cloud resources, and existing software services to suit the needs of an application autonomously? A related question that will govern the existence of IoT ecosystem is, how will IoT deliver what an individual person needs considering his/her preferences, situation, and task/activity at hand?

For IoT data to find its way to humans and machines that can consume and benefit from it, future IoT applications will have to manage the discovery, integration, and interoperability of things, cloud services, and third party applications seamlessly. Currently there is a major growth in IoT application development in competing enterprises silos, such as the Apple versus the Samsung universe. The challenge is how to seamlessly produce IoT application across such silos that are driven by consumers instead of silo owners/providers. For example, a reminder from a car that need oil change, should be sent to either the Apple iPhone or the Samsung Smart TV of its owner using sensors in a smart home that detects its owners' location and activities in real-time. An appointment is then made to a car service workshop and a navigation plan (using a third-part navigation service such as google navigation) is generated in real time based on the traffic and road conditions at the time of the car service appointment. All this should be done automatically but they can be adjusted by the owner by simply querying and changing the car service appointment that has been automatically inserted in his/her calendar.

The goal of this paper is to present the challenges and propose a new solution for a discovery-driven IoT architecture. In particular, we propose a service-based IoT architecture where every IoT component (including IoT devices, cloud resources, and application components) is a service, allowing dynamic discovery, composition and integration of services based on application requirements. We start with a motivation in section II, followed by an example use-case in Section III that explains the need for a discovery-driven IoT architecture. In section IV, we present an abstract IoT model and discuss the main challenges. Section V presents our discovery-driven IoT architecture followed by discussion on state-of-the art in the literature in Section VI. Section VII concludes the paper.

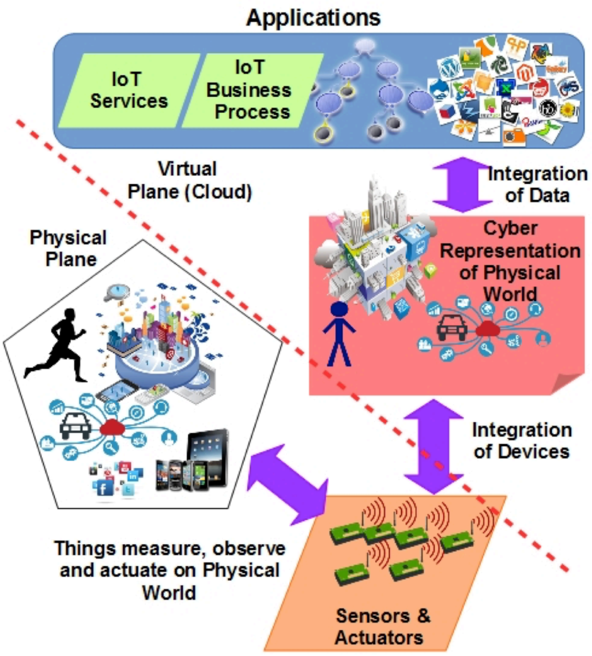

Figure 1: Conceptual Description of the IoT Ecosystem

Figure 1 presents a conceptual overview of IoT and its components

II. Motivation

The IoT ecosystem is spread across two planes namely the physical plane and the virtual plane. The **physical plane** represents the physical world comprising of objects such as smart devices, buildings, infrastructure, humans, cars etc. Changes in the physical world are captured by sensors and actuators (real and virtual). The **virtual plane** is the cyber representation of the physical world entities. In most IoT systems, the virtual plane is deployed on cloud computing infrastructure eliminating the need for owning, housing and maintaining computing resources.

The virtual plane comprises the cyber representation layer and the application layer. The cyber representation manages modeling of the physical entities as virtual entities. The sensors and actuators in the physical world change the states of the virtual entities based on changes in the physical world. The application layer in the virtual plane comprises IoT services (such as third party data analysis and visualization applications offered as a service), business processes (workflows and related workflow automation and application integration services) that control/monitor/detect and respond (decision making) to state changes in the physical world.

Consider an example “Give me the indoor temperature in Room 1.23” or “Set light level in Room 2.57 to 15”. To support the interactions between the virtual plane and the physical plane, the relation between IoT devices in the physical world and their virtual entities in the cyber representation layer needs to be modeled, which is done in form of associations. For example, the association will contain the information that the indoor temperature of Room 1.23 is provided by Sensor 456. This association is used by the application layer to control the room temperature of Room 1.23 by changing the state of the sensor 456. The virtual world and the corresponding ecosystem of components are created and maintained on cloud computing infrastructure (private or public). Cloud is a complementary technology that is required for widespread adoption of IoT.

III. Use-Case Example

To articulate the IoT challenges and our vision of a discovery driven architecture (which we refer as IoT blueprint), consider the city of Melbourne, Australia, where millions of smart meters have been rolled out. Consider now that home and business owners want to reduce energy use and that to do so they need a service that provides the ability to analyse and understand the energy-usage patterns of their premises. To provide such a service, we will need to find and and integrate 1) the customer’s smart meter and its data to be able to obtain the current and historical energy consumption from premises of the customer, 2) weather data available from third-party web service or a sensor in the area of the customer’s premises (e.g., a Netatmo[1] deployed by the customer or a neighbour) to be able to correlate energy consumption with environmental conditions, 3) cloud-based services to store the historical energy consumption and related environmental data for analysis purposes, and 4) a third-party data analytics service (e.g., an applet selected by the customer from an online store) that analyses the streaming data from the smart meter and the historical data in the cloud to provide insights into energy usage and make suggestions to the customer in real-time.

Another trend for developing such IoT capabilities can be found in industry automation domain. In particular, Industry 4.0 (lead by EU) pursues connecting machines, forming work pieces and systems, and creating intelligent networks along the entire value chain that can control each other autonomously. Some examples for Industry 4.0 are machines that predict failures and trigger maintenance processes autonomously or self-organized logistics that react to unexpected changes in the production. To achieve the above objectives, there is a need

---

[1] https://www.netatmo.com/en-US/site

for systems that are able to discover and integrate services across IoTss physical and virtual layers on-demand.

However, the fundamental challenge in composing such applications is the ability to represent each component in the IoT ecosystem as a service and to be able to integrate it as needed by the application. For example, let's consider a simple case of a cloud integration of data from IoT devices. There are many cloud service providers from azure to amazon. The problem that many users of IoT will face in the near future is interoperability and ability to integrate services offered by different platform in order to develop IoT application. For example, the simplest question such as "where should I store my IoT sensor data in the cloud?" could be very complex to resolve within the context of IoT. These issues are addressed next, in Section IV.

IV. IoT: An Abstract Model

Figure 1 presented a conceptual view of the IoT echo system. In this section, we develop a model that abstracts this in to three IoT layers depicted in Figure 2, namely a device, cyber representation (data) and application layers.

This model will help articulate some of the underlying challenges in integrating and composing IoT solutions that will help end-users interact with and learn from the billions of things. The three layers as depicted in Figure 2 are:

Device Layer: The device layers maps to the physical plane. It consists of both real hardware devices such as wireless sensors, mobile devices and virtual devices such as social media sensors. The device layer focuses on discovery and integration of sensors, the metadata that describes the sensors and the data they produce.

Data Layer: This layer is responsible for integrating the data arriving from the sensor based on application needs. The function of this layer maps well with the cloud computing platform-as-a-service layer that provides services such as storage and processing. This layer will also make use of integration services offered by the application for e.g. in case of the example presented in Section III, the data about temperature from say Netatmo weather station maybe available only in Celsius while the application may require data in Fahrenheit. In such cases, the data layer may function as follows 1) provide interfaces to integrate conversion services provided by the application or 2) receive a signature of the application service and discover a suitable conversion service that can match the requirements of the application. In either cases, the process of checking compatibility between requested data and provided data is the responsibility of the data layer.

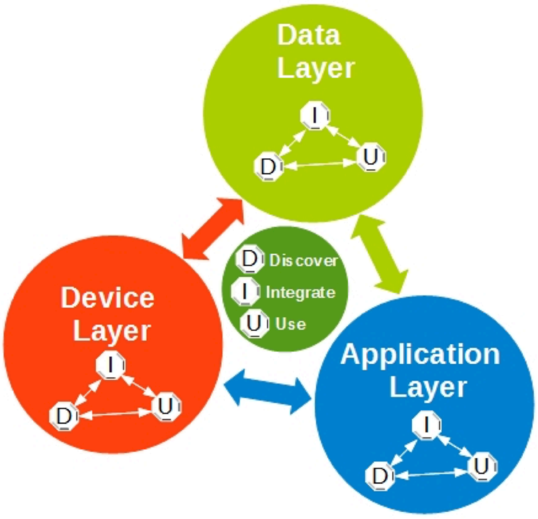

Figure 2: IoT Abstract Model

Application Layer: The application layer is responsible for analysing the incoming data sources to produce insights into the data to match the application requirements. This service could again be hosted on cloud infrastructure. However, our focus at this layer is not on the infrastructure but the services offered e.g. a big data services to analyse web logs. The application layer could take advantage of existing workflow description tools such as BPEL to describe the services and rules for orchestration.

Discover-Integrate-Use: The common function that is part of all the layers is the ability to discover, integrate and use the services offered across the layers. In the view of this paper, we define discovery as the ability to find resources required by a service provided to a customer, including third-party provided devices, cloud services, as well as analytics and integration components provided as web services. Each layer includes an integration service that is responsible for orchestrating the required resources within its layer, as well as providing the integration to the subsequent layer. For example, the data layer could provide an abstraction of cloud operations and internally manage the complexities of mapping those abstractions to actual cloud providers such as Amazon, Azure, GoGrid and so on. The abstract model will help end-users answer the previously described question i.e. where should I store my IoT data?

IoT comprises billions of devices and provides the opportunity to develop many novel applications. However the primary challenge is to achieve impact at scale. Consider the example similar to section III, where a user would like to know the amount of rainfall in a given region. If there are no rain-gauges available in that region, an expert could look into the data from surrounding regions to estimate the rainfall. To achieve such an outcome autonomously based on a request from an application

- We need to identify the relevant data sets at the device layer
- Identify the corresponding data integration services e.g. what sort of storage will be required, what sort of compute will be required? If these are available via API's how to invoke them autonomously (e.g. amazon queuing service and azure queuing service could be described using a common ontology. However their performance and implementation dependencies will vary. How can this be captured and used during the discovery and integration process)

- Finally identify the data processing services required by an application which could be owned and operated by independent third-part providers.

The impeding challenge that lies underneath is the integration and interoperability issue. The interoperability can be addressed to some extent by the use of ontologies or knowledge bases. However, we still need good system developers who can understand the nuances of the open API's of various systems and integrate them for application needs. This again leads to creation of silos as a number of development efforts will be focused on specific technology binding that are not portable across providers. The analytics as-a-service platforms introduce another layer of complexity as they are further required to integrate the data from cloud data stores to produce meaningful outcomes. Currently there are limitations for machines to search and discover such services without human intervention. Further, it is hard for a common man to take advantage of these services without a deep understanding of it functions. This is where a smart IoT system can make a significant difference by making these processes autonomous and simply allowing users to compose application for example using workflows. The workflows could have in-built validation models and connects to various discovery services to find relevant devices, infrastructure components and processing components thereby hiding the internal complexities from the user.

There is a lot of work on discovering devices at the network layer [4] such as CoAP, UPnP etc. There are also works that describe devices to high level applications using ontologies such as the W3C Semantic Sensor Network [5] allowing semantic discovery of these devices. Similarly, for cloud layer service, there are ontologies and cloud recommender systems that can assist in finding the relevant resources based on user requirements [6]. At the analytics layer, there has been recent work focusing on SOA-based approach for integrating IoT [7, 8]. The analytics applications themselves could run on cloud infrastructure but owned and managed by independent entities willing to offer their platform as a service.

The challenge that lies ahead is the ability to unify these independent approaches into one framework that allows seamless discovery and integration of devices, infrastructure components and service. The key challenges from this perspective would be

Uniform Description: Describing the layers consistently so that the information can be discovered/queried

Discovery mechanisms: Semantic description is one mechanism. There are also probabilistic approaches and other hybrid approaches that fuse sematic and probabilistic reasoning.

Integration: Ability to integrate the discovered services with little human intervention. We are still a few steps from realising a true autonomous machine-to-machine communication for IoT application. Currently a number of these approaches require expert developers for the integration process

Representation: Workflow is an obvious way to represent the applications' requirement that can be transformed into different layers. Using SOA principles at this layer could

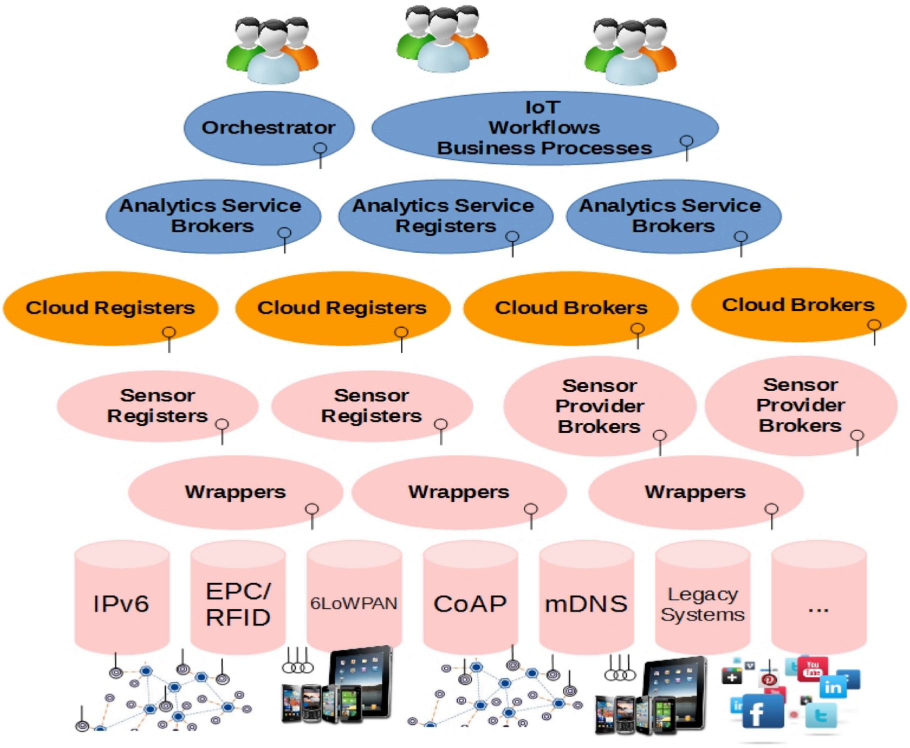

Figure 3: Architecture of Discovery Driven IoT Service

work. However the challenge that lies ahead is how can we make this simple and intuitive across the IoT ecosystem rather than developing custom dashboards that only work for specific types of data and technologies.

V. A DISCOVERY-DRIVEN SOLUTION FOR SERVICE ORIENTED IOT ARCHITECTURE

A discovery-based IoT solution development is a mechanism that will aid in enabling end-users to compose applications that access and process the IoT data without the need to know either the actual source of data, infrastructure capabilities and location nor the complexities of the data processing algorithms/services.

Figure 3 presents our proposed discovery-driven service oriented IoT architecture. In the proposed architecture, every component of the IoT stacks exposes itself as a service that can be discovered and integrated with other services to meet the application requirements. As it can be noted from the blueprint architecture, we align our model very closely with the SOA cloud paradigm by re-using the notion of service registrars and service brokers. The registrars maintain the service description at each layer of the IoT stack (device to processing). The service brokers are responsible to provide service integration at each layer of the IoT stack.

Device Layer: At the device layer, we present our previous work in the area developing a semantic IoT middleware namely OpenIoT [9]. The notion of OpenIoT is to develop a middleware for IoT that is driven by device discovery breaking away from the traditional vertical IoT solutions. The architecture of OpenIoT is presented in Figure 4.

In OpenIoT, the physical plane is the sensor middleware which collects, filters and combines data streams (e.g. signal processing algorithms, information fusion algorithms and social media data streams) stemming from virtual sensors or physical sensing devices (such as temperature sensors, humidity sensors and weather stations). This middleware acts as a hub between the OpenIoT virtual plane and the physical world, since it enables access to information stemming from the real world. The notion of virtual sensors in OpenIoT is to expose the device layer as a service to the virtual plane. It facilitates the interfacing to a variety of physical and virtual sensors such as IETF COAP compliant sensors (i.e. sensors providing RESTful interfaces), data streams from other IoT platforms (such as https://xively.com) and social networks (such as Twitter). The sensor middleware has the ability to stream W3C SSN compliant sensor data into the cloud. This allows a standardised representation of sensor data and the metadata in the cloud.

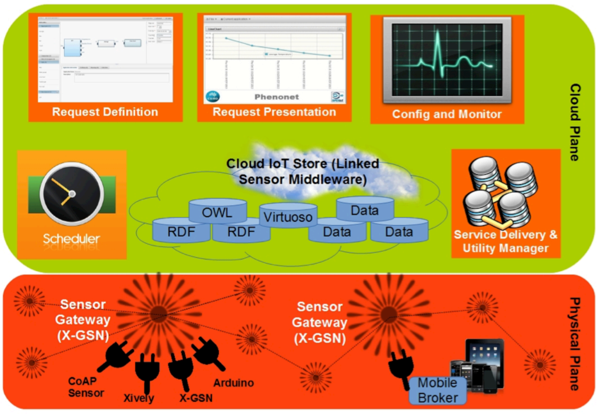

Figure 4: OpenIoT Architecture

Data Layer: In OpenIoT, the virtual plane (cloud) is fused with a combination of data and application components. This integration between the components of OpenIoT makes the system at the virtual layer less re-usable. The individual components can be replaced based on application needs but the notion of discovery is absent at the virtual plane of OpenIoT architecture. Our proposed blueprint architecture bridges this gap by introducing components in the data layer as services that can be discovered and integrated at run-time. To this end we propose a methodology to realise this layer built upon our previous work in cloud recommender systems.

Traditionally, cloud service recommenders are used as intermediary software applications between the cloud providers and the end-users such as government decision makers, IT consultants, CIOs, small and medium enterprises (SMEs), scientists, radiologists, and the like. It equips end-users with advanced techniques and mechanisms for optimizing the management of their applications (e.g. web applications, scientific experiments, medical imaging, IPTV, etc.) on cloud services (e.g. compute server, storage disks, databases, etc.). In particular it supports: (i) transparent decision support and cost estimation of cloud services; and (ii) detection of opportunities for optimising cloud service consumption.

The cloud recommender system proposed in our previous work [6] aids in network-QoS aware selection of cloud services. It takes into account real-time and variable network QoS constraints. We also developed a unified domain model [10] based on [11, 12] that captures the complexities and functionalities of the various cloud provided services. The cloud recommender system allows end-users and machines to identify cloud resources that satisfy the QoS constraints of the application. It is capable of supporting a utility function that combines multiple selection criteria pertaining to storage, compute, and network services. In cloud recommender, we provide a clear formulation of the research problem by identifying the most important cloud service selection criteria relevant to specific real-time QoS-driven applications, selection objectives, and cloud service alternatives. The cloud recommender approach is different to current approaches as depending on the complexity and requirement of the storage and compute, the SLA for the composed application will vary significantly. The cloud recommender also encompasses an orchestrator that integrates the ability to facilitate and provision cloud services on-demand using abstract interface definitions. By using the cloud recommender service as a means to identify suitable cloud infrastructure service on-demand we achieve the following outcomes 1) ability to place sensor data in the cloud based on location of incoming data, 2) place/replicate sensor data in the cloud based on application needs, 3) negotiate the storage of data on a pay-as-you-go model, 4) expose the data as a service and 5) provide orchestration services for integration of resources on-demand.

Application Layer: The application layer as stated earlier in OpenIoT is fused with the data layer. Hence, the application and the data reside together. In order to repurpose the data to be used by other applications, the system will need to tap into the device layer discovering appropriate service. OpenIoT take the view of sensors-as-a-service by employing discovery to find sensors. We push the envelope further by defining every component of an IoT stack as a service allowing dynamic composition and integration of services e.g. find an infrastructure service that can provide CPU processing at *x* cycles/second to store data from *s* sensors.

To present an example of how the proposed discovery-driven service oriented IoT architecture is used, consider again our smart grid example we discussed in Section III. Figure 5 presents the steps involved in realising such an application based on the proposed discovery-driven service oriented IoT architecture.

Step 1: The customer's analyser service for energy consumption is implemented via workflow manager/orchestrator that first queries the device register to obtain the information of the smart meter and the weather station at/near the customer's premises. The smart meter is an energy consumption measurement device that provides its data via a web service that is managed by the customer's energy distributor. The weather station is a sensing device that is offering its data as a service over protocols such as CoAP/Bluetooth. The sensing-as-a-service implementation conceals the underlying weather station by exposing a web service endpoint to access the data.

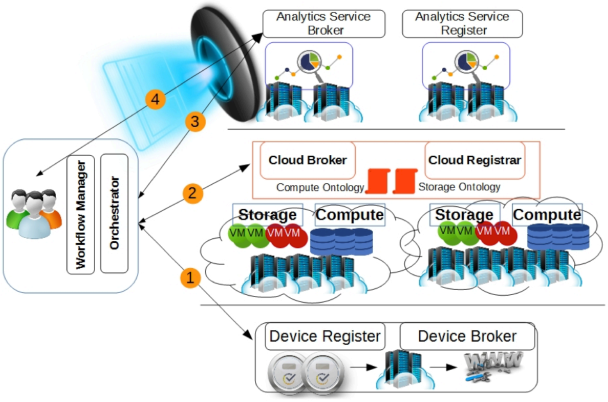

Figure 5: A smart energy grid application using the proposed discovery-driven IoT Architecture

Step 2: The cloud brokers and registrars are queried to find the suitable cloud storage to store the smart meter and weather station data, as well as compute resources for performing the data analysis.

Step 3: The data analytics applet compares energy usage of the household with that of others of similar size and occupancy. In Section II we assumed that data analytic components are selected by the customer. Alternatively, a data analytics service broker (as shown in Figure 5) can be queried to automatically select this or an alternative data analytics service. Finally the orchestrator creates the service needed by the customer, by integrating the independent services at each layer (i.e. at the device, infrastructure and analytics layers).

Step 4: The integrated service presents the energy usage analysis results to the customer.

VI. RELATED WORK

In the section, we present a discussion on current work in across the three layers of the IoT stack namely the device, the data and the application layer.

Device Layer: There have been works to develop common representation of sensor and sensor data such as SensorML[2], OGC/SWE[3], W3C SSN [5], HyperCat and several other ontologies/semantic models. But the challenge is to develop models that can cater to the ever expanding universe of Internet of Things. These mechanisms should allow applications/users to access the IoT data without knowing the actual source of information. To achieve this, the models should have means to exchange applications context to discover appropriate and related data.

There have been numerous standardization efforts to develop protocols for resource constrained IoT devices that support discovery inherently such as 6LoWPAN[4], CoAP[5], XMPP, CoRE link format specification. Other industry standards that employ IP-based service discovery including SLP, UPnP, JINI and Salutation are not directly application to resource constrained devices due the complexity of formats and high communication demands. Technologies such as mDNS and mBonjour tough optimized for 6LoWPAN still relies on IP multicast and entails more communication overheads. Unfortunately the discovery capabilities of these technologies are restricted at the resource level and are able to only guarantee awareness about presence of other devices. There is a significant challenge in bridging the gap between the awareness at device level and awareness across applications that use the data from the things. This is where novel discovery algorithms are required that can from the application provide user's context while take advantage of the mutual device awareness presents at the device level.

Jara et al., [4] present a discovery driven approach to interact with IoT. The focus of the work is to provide global resource discovery in particular devices in the IoT ecosystem. The major thrust of this work is devoted towards understanding the current IoT network level device discovery mechanism and providing an architecture that integrates existing mechanisms and protocols. They present a web-service based platform to integrate the data from devices with a mobile phone application. However, much of the focus is at

---

[2] http://www.opengeospatial.org/standards/sensorml

[3] http://www.opengeospatial.org/ogc/markets-technologies/swe

[4] http://www.ti.com/lsds/ti/wireless_connectivity/6lowpan/overview.page

[5] http://coap.technology/

addressing integrating challenges at the devices layer that are consumed by custom-built smartphones applications.

Data Layer: Proliferation of cloud computing has revolutionized hosting and delivery of IoT-based services. There have been an exploding of new data processing technologies, like scalable cloud computing hardware infrastructure (from vendors like Amazon, Azure, Google); software paradigms in distributed message queue (e.g. Apache Kafka, Kinesis), data storage (e.g. MongoDB, Cassandra), parallel processing (e.g. Hadoop, Spark, Storm) and distributed data mining (e.g. Mahout). However, with the constant launch of new cloud services and capabilities almost every month by both big (e.g., Amazon Web Service, Microsoft Azure) and small companies (e.g. Rackspace, Ninefold), decision makers (e.g. application developers, CIOs) are likely to be overwhelmed by choices available. The decision making problem is further complicated due to heterogeneous service configurations and application provisioning Quality of Service (QoS) constraints. These issues further complicate the on-demand service composition for IoT applications. The cloud description and discovery area has been well studied [6, 11, 12] via cloud recommender systems, cloud ontologies and service-oriented cloud computing. [12] presents a unified ontology to define cloud services using abstract definitions while [11] presents more concrete definitions of a cloud ontology taking into consideration current genre of cloud computing providers.

Application Layer: This layer encompasses the cloud computing software-as-a-service model. Service-Oriented Cloud Computing models provides means for service description, discovery, innovation, composition and interoperation [7, 11, 13, 14] including Semantic Web (OWL-S), Web Components, BPEL, Petri nets, Model Checking/Finite State Machines and π calculus. Colombo et al. [13] present a service oriented architecture for collaborative automation. The system enables orchestration of manufacturing service in a production line. They have used the OWL-S service ontology to describe and discover services. However, the work is limited to a single domain and does not deal with the complexities of integrating services offered by third party providers at internet scale. The proposed blueprint architecture can support service composition and descriptions at a conceptual level by the means of workflows. Languages such as BPEL could be used to represent workflows that are responsible for automated orchestration and validation of composed services. However, solving this is outside the scope of this paper.

VII. DISCUSSION AND CONCLUSION

As evident from the discussion, most of the current state-of-the art are disconnected efforts focusing on specific layers of the IoT stack. There is a clear gap in IoT architectures that integrate service oriented concepts across the layers allowing autonomous composition of IoT applications. The vision presented in this paper addresses these gaps by proposing a blueprint architecture for a discovery driven service oriented IoT architecture. By doing so, we also presented the challenges in discovery, description and representation of service and integration. The proposed blueprint architecture aims to address these challenges by embedding discovery and integration of services at each of the devices, data and applications layers. We provided discussions into how the proposed model can be realised by presenting some of our related previous work in the areas of sensor discovery in the OpenIoT and cloud discovery in Cloud Recommender systems.

However, the challenge that still needs to be addressed is the ability to orchestrate and integrate these services autonomously by consuming the corresponding web services based on their service descriptions. This is not a trivial task as currently the integration is managed by system developers and programmers. However to realise the true potential of services that IoT offers, the orchestration of these services based on application requirements needs to be more smart. The proposed blueprint architecture in this paper establishes the foundation for further investigation, realisation and development of IoT-based smart services and applications that provide autonomous discovery and integration.

References

1. K. Ashton. That 'Internet of Things' Thing. In the real world, things matter more than ideas. Jun 22, 2009 [cited 2015 2nd July]. Available from: http://www.rfidjournal.com/articles/view?4986.
2. O. Vermesan, P. Friess, P. Guillemin, S. Gusmeroli, H. Sundmaeker, A. Bassi, et al. Internet of Things Strategic Research Roadmap. 2009.
3. R.D. Sriram, A. Sheth. Internet of Things Perspectives. IT Professional. 2015;17(3):60-3.
4. A.J. Jara, P. Lopez, D. Fernandez, J.F. Castillo, M.A. Zamora, A.F. Skarmeta. Mobile discovery: discovering and interacting with the world through the Internet of things. Personal Ubiquitous Comput. 2014;18(2):323-38.
5. M. Compton, P. Barnaghi, L. Bermudez, R. García-Castro, O. Corcho, S. Cox, et al. The SSN ontology of the W3C semantic sensor network incubator group. Web Semantics: Science, Services and Agents on the World Wide Web. 2012;17(0):25-32.
6. M. Zhang, R. Ranjan, S. Nepal, M. Menzel, A. Haller. A declarative recommender system for cloud infrastructure services selection. Proceedings of the 9th international conference on Economics of Grids, Clouds, Systems, and Services; Berlin, Germany. 2437801: Springer-Verlag; 2012. p. 102-13.
7. D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, D. Savio. Interacting with the SOA-Based Internet of Things: Discovery, Query, Selection, and On-Demand Provisioning of Web Services. Services Computing, IEEE Transactions on. 2010;3(3):223-35.
8. T. Teixeira, S. Hachem, V. Issarny, N. Georgantas. Service Oriented Middleware for the Internet of Things: A Perspective. In: Abramowicz W, Llorente I, Surridge M, Zisman A, Vayssière J, editors. Towards a Service-Based Internet. Lecture Notes in Computer Science. 6994: Springer Berlin Heidelberg; 2011. p. 220-9.
9. M. Serrano, Q. Hoan Nguyen Mau, P. Danh Le, M. Hauswirth, J. Soldatos, N. Kefalakis, et al. Defining the Stack for Service Delivery Models and Interoperability in the Internet of Things: A Practical Case With OpenIoT-VDK. Selected Areas in Communications, IEEE Journal on. 2015;33(4):676-89.
10. S. Nepal, M. Zhang, R. Ranjan, A. Haller, D. Georgakopoulos, editors. An Ontology-based System for Cloud Infrastructure Services' Discovery. COLLABORATECOM; 2012.
11. T. Wei-Tek, S. Xin, J. Balasooriya, editors. Service-Oriented Cloud Computing Architecture. Information Technology:

New Generations (ITNG), 2010 Seventh International Conference on; 2010 12-14 April 2010.

12. L. Youseff, M. Butrico, D. Da Silva, editors. Toward a Unified Ontology of Cloud Computing. Grid Computing Environments Workshop, 2008 GCE '08; 2008 12-16 Nov. 2008.

13. A.W. Colombo, F. Jammes, H. Smit, R. Harrison, J.L.M. Lastra, I.M. Delamer, editors. Service-oriented architectures for collaborative automation. Industrial Electronics Society, 2005 IECON 2005 31st Annual Conference of IEEE; 2005 6-10 Nov. 2005.

14. N. Milanovic, M. Malek. Current solutions for Web service composition. Internet Computing, IEEE. 2004;8(6):51-9.