



Mobile web and cloud services enabling Internet of Things

Satish Narayana Srirama¹ 

Received: 6 November 2016 / Accepted: 26 November 2016 / Published online: 4 January 2017
© CSI Publications 2017

Abstract The Internet of Things (IoT) represents a comprehensive environment that consists of a large number of sensors and mediators interconnecting heterogeneous physical objects to the Internet. IoT applications are prominent in many areas such as smart city, smart workplace, smart plants, smart agriculture and various ubiquitous computing areas. The research roadmap of IoT spans across vast domains such as mobile computing, wireless and sensor networks, service oriented computing, middleware, cloud computing and big data analytics, taking advantage of several recent breakthroughs in the respective domains. Primarily, the challenges associated with realization of IoT scenarios can be summarized across three layers: sensing and smart devices layer, connectivity layer and cloud layer. The first layer deals with the physical objects, including energy-efficient communication of the devices and developing the associated standards so that the interaction among the devices is seamless. The connectivity layer deals with the sensor data acquisition and provisioning, through gateways and sinks. The top cloud layer deals with resource provisioning for storage and processing of the acquired data, in extracting domain specific information. The participation of smart phones both as sensors and the gateways, brings in the scope for mobile web services and mobile cloud services, into this cloud based IoT architecture. The paper takes the cross-layered approach and tries to address the primary challenges of IoT through mobile web and cloud services. The paper also discusses the state of the art of each of the

respective research domains along with scope for extensions and recent trends.

Keywords Internet of Things · Cloud computing · Mobile web services · Mobile cloud · Middleware · Big data

1 Introduction

Internet of Things (IoT) represents a comprehensive environment that consists of a large number of sensors and mediators interconnecting heterogeneous physical objects to the Internet. Internet has been around for a while and it has significantly changed how people have been communicating. Traditionally, the content available on the Internet was generated by the people for the people, which was also called *Web 1.0*. With the advent of social networking applications most of the content available today is dynamically generated by the participants and is being termed as *Web 2.0*. Semantic web is termed as *Web 3.0* [1]. IoT has evolved as the next advancement in this domain, where the things/devices are communicating among themselves in achieving something valuable for the people. IoT can be termed as *Web 4.0 and beyond*, with its support for intelligent connections. We can easily envision a scenario, where an individual's smart phone and smart home devices communicate together, in making his life better. For example, using the smart phone's location, destination set on the car's GPS (Global Positioning System) and the hour of the day, it can be predicted that the user is starting back from office and this can start the thermostat in his apartment, so that by the time he comes to the apartment, it is well heated. Similarly, the moment the individual is in the apartment, the lights and TV can turn on, and the microwave can already start heating his food.

✉ Satish Narayana Srirama
srirama@ut.ee

¹ Mobile & Cloud Lab, Institute of Computer Science, University of Tartu, Ulikooli 17, Tartu, Estonia

Today there are more connected things than the world population, and the surpassing of the connected things have already happened around 2008. Cisco predicts there will be over 50 billion connected things by the year 2020 and believes the market capture will be over 19 trillion dollars by 2025 [21]. IoT applications can be found in many areas such as environmental protection, smart city, smart workplace, smart plants, smart healthcare, smart agriculture and various ubiquitous computing areas [28, 35]. The connected things in these applications can be physical objects with sensing capabilities for temperature, humidity, motion detection etc., physical goods or even food items attached with tags such as active and passive RFID (Radio-frequency identification) whose movement can be tracked, mobile things such as smart phones and moving vehicles, and appliances such as fridge, TV, microwave, thermostat etc.

While the connected things are interesting, IoT offers several challenges in building ideal applications such as how the things can be communicating among each other?, and how the external agents can interact with them?, and how can we deal with these communication issues, still considering energy efficiency as most of the physical objects in IoT are battery powered/low energy devices? The research roadmap of IoT spans across vast domains such as mobile computing, wireless and sensor networks, service oriented computing, middleware, cloud computing and big data acquisition and analytics, taking advantage of several recent breakthroughs in the respective domains. Primarily, the challenges associated with realization of IoT scenarios can be summarized across three layers: sensing and smart devices layer, connectivity layer and cloud layer. The first layer deals with the physical objects, the connectivity layer deals with the sensor data acquisition and provisioning, and the top cloud layer deals with resource provisioning for storage and processing of the acquired data in extracting domain specific information. The participation of smart phones both as sensors and the gateways, brings in the scope for mobile web services and mobile cloud services, into this cloud based IoT architecture. This paper takes the cross-layered approach and tries to address the primary challenges of IoT through mobile web and cloud services. In the process, the paper will also discuss the state of the art of each of the respective research domains along with scope for extensions and recent trends.

Section 2 introduces the layers of cloud based IoT. Sections 3 and 4 discuss mobile web services and mobile cloud, respectively, along with their state of the art and related works. Section 5 deals with remote cloud based IoT data processing. Section 6 discusses a scenario and shows how the layers can work in tandem in achieving the goals. Section 7 concludes the paper along with research roadmap for IoT.

2 Cloud based Internet of Things

The layers of the cloud based IoT are shown in Fig. 1. Each layer targets specific challenges related to IoT and the mobile web and cloud services ease the interactions across the layers.

2.1 Sensing and smart devices

The sensing and smart devices layer primarily deals with IoT devices such as sensors and actuators. Sensors can detect events or changes in their environment such as motion, temperature or light in a room. Actuators control physical objects such as opening and closing of the doors and regulating power on/off/changing illumination of a smart lamp. There is significant research going on in this domain and it is evident that smaller and smarter devices are coming into the market regularly.¹ There are also several microcontrollers and kits being developed such as Raspberry PI² and Arduino,³ which can hasten building and prototyping digital devices and interactive objects that can sense and control physical objects. The smart devices can be connected by wired/wireless technologies using different communication protocols such as ZigBee [52], Z-Wave [27], WiFi/WiFi-direct [14], Bluetooth⁴ etc. Main research challenges in this space include energy-efficient communication of the devices and developing the associated standards so that the interaction among the devices is seamless.

2.2 Gateway/connectivity nodes

The connectivity layer primarily deals with the sensor data acquisition and provisioning. Here gateways come into picture which can collect the data from different sensors

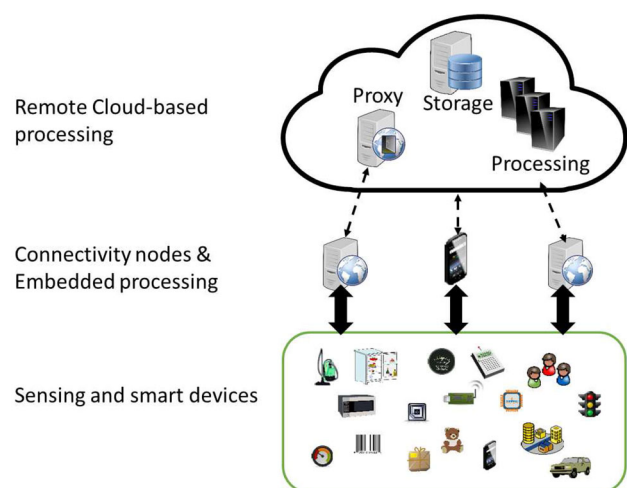


Fig. 1 Layers of the cloud based IoT

and can make it accessible to the external entities. The gateway may not always provide just the raw data from the sensors. Sometimes it is logical for it to process the data locally and provide the consolidated results to the requesting parties. This reduces the communication latencies and thus saves energy of the gateways, especially when mobiles are participating as gateways.

The participation of the mobiles as the gateways is interesting, as the devices generally have several in-built sensors such as GPS, accelerometer, gyroscope, magnetic field sensor etc., whose information can be provided directly. In addition, the mobile has significant knowledge of the user such as his preferences, context, location etc. which can be used in designing ideal IoT scenarios and applications. Mobile being gateways brings in the scope for mobile web services and mobile cloud services, which will be discussed further in the next sections.

Predictive analytics also helps the gateway layer in saving the energy of the physical objects. It is not always necessary to extract the sensor information dynamically and the intervals at which the sensor data is to be extracted can be configured. For example, in a scenario where sensors are deployed to detect forest fires, by applying proper data analytics on the collected sensor information over the years, it can be predicted that during the winter and rainy seasons we do not have to collect the sensor data as frequent as during the summers.

2.3 Remote cloud-based processing

The cloud layer primarily deals with the storage and processing of the sensor data, which is acquired from the gateways. Cloud computing [4] has emerged as one of the most prominent platforms instigating enterprise and social networking applications. With the advent of cloud computing, establishing startups have become very easy, especially because of its features such as on-demand, elasticity and utility computing. This is evident from the fact that several of the successful social networking applications such as Twitter⁵ and Instagram⁶ have initially been established on the cloud resources before becoming the successful ventures and opting for dedicated infrastructures. Cloud computing with its promise of virtually infinite amount of resources, has huge potential to also drive the IoT domain.

¹ <http://iotlist.co/>.

² <https://www.raspberrypi.org/>.

³ <https://www.arduino.cc/>.

⁴ <https://www.bluetooth.com/>.

⁵ <https://twitter.com/>.

⁶ <https://www.instagram.com/>.

Apart from providing the storage and processing capabilities, the cloud layer also acts as a proxy by offering the infrastructure for the middlewares required in realizing the IoT. The middlewares are discussed in the coming sections. The primary research issues from the cloud layer include resource provisioning, auto-scaling and dynamic deployment. They are elaborated further in Sect. 5.

3 Mobile web services

Mobile is the seventh mass media channel [2] which has significantly changed our lives since 2000. Smart phones have become a part of our daily lives and more than 80% of the world population today owns a mobile phone. There have been significant advances in the mobile device space during the last decade. They now have better CPU and memory capabilities, embedded hardware such as camera and Wi-Fi, and in-built sensors such as GPS, accelerometer, gyroscope and magnetic field sensor, making them usable in versatile scenarios. We also have achieved higher data transmission rates for mobiles with the advances in 3G, 4G, 5G and WiFi, paving way for mobile commerce and location based services. In addition, the advances in mobile marketing such as Google Play⁷ and App Store⁸ have made developing and distributing mobile applications relatively simple.

The advances in the mobiles and the adaptation of component based service oriented architectures (SOA) [20] everywhere have made the space for mobile web services (MWS) [34]. In traditional SOA, the service providers offer certain functionality and are accessible using protocols such as SOAP (Simple Object Access Protocol), REST (Representational State Transfer) etc. The services are advertised using different solutions such as UDDI (Universal Description, Discovery, and Integration) or peer-to-peer based discovery mechanisms [16]. Mobiles can participate in this space both as web service clients as well as providers (Mobile Hosts) [3, 40]. Significant research has been conducted with MWS, dealing with quality of service (QoS) aspects such as scalability and security, Peer-to-Peer (P2P) based mobile web service discovery and integration with Enterprise Service Bus (ESB) based middleware, and the solutions are mature enough [41]. We can easily envision scenarios such as Mobile Social Networks in Proximity (MSNP) using MWS. The idea of MSNP is to detect the devices in proximity and using the services offered by them in realizing social networking applications and trust-based mechanisms [11].

⁷ <https://play.google.com/store>.

⁸ <https://itunes.apple.com/us/genre/ios/id36?mt=8>.

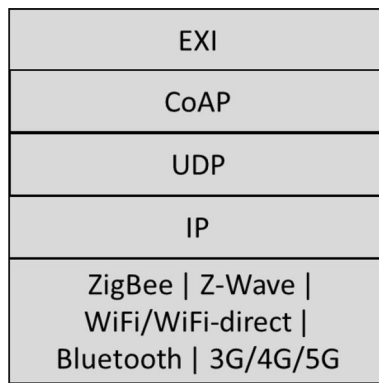


Fig. 2 Protocol stack for light-weight mobile web service provisioning for IoT mediation

With the feasibility of Mobile Hosts, the mobile devices can easily collect the sensor information, which can be acquired from the mobiles as and when necessary. However, when considering Mobile Hosts for sensor mediation, the efficiency of traditional approaches are still limited. The challenges mainly stem from the usage of fundamental protocol stack. HTTP (Hypertext Transfer Protocol) used for the information exchange in MWS is primarily based on Transmission Control Protocol (TCP) which has significant overhead. In addition, they use inefficient payload compression approaches such as Binary XML and JavaScript Object Notation (JSON). This makes the necessity for a completely new protocol stack (Fig. 2) when dealing with light-weight mobile web service provisioning for the sensor mediation [31].

At the low level of the protocol stack are the communication technologies which are ideal for IoT. User Datagram Protocol (UDP) can be used instead of TCP for the message transportation. Constrained Application Protocol (CoAP) [38] is a web transfer protocol based on REST and is specifically designed for use in IoT, targeting constrained nodes and networks. Efficient XML Interchange (EXI) [7], a binary XML format, can be used for efficiently compressing the message payload. The protocol stack has shown to achieve higher throughput and less resource consumption than the traditional MWS frameworks, especially for the IoT setups [31].

4 Mobile cloud

While the advances in the mobiles are significant and they are also being used as service providers, they still have certain limitations. Battery life is one space where the advances are not sufficient. Mobile battery still lasts only for about 1–2 h, if used for continuous computing. Wireless charging⁹

⁹ <http://www.androidcentral.com/wireless-charging>.

of mobiles is a good solution to deal with the problem. However, it is being supported only in small subset of smart phones, due to the issues with the wireless charging standardization [47]. Apart from battery life, we still have not achieved the same quality of experience as on the desktops due to weaker CPU, memory and storage capacity. So, it is still a good idea to take advantage of external resources for building resource intensive mobile applications. This brings in the scope for the cloud computing as a source of external resources, paving way for the mobile cloud domain. Mobile cloud applications thus achieve increased data storage capacity, availability of unlimited processing power and extended battery life.

Mobile cloud has generated significant research interest, and currently, there exist two binding models using which the integration of mobiles and cloud computing can be realized. They are the *task delegation* and the *mobile code offloading* models [26]. Before proceeding further with the binding models, it is interesting to define the mobile cloud ecosystem. One should not see mobile cloud to be just a scenario where mobile is taking the help of a much powerful machine in proximity. Similarly, we should not be seeing cloud as just a pool of virtual machines. An ideal mobile cloud system should efficiently take advantage of some of the key intrinsic characteristics of the cloud, such as elasticity, auto-scaling, utility computing model or parallelization using approaches such as MapReduce.

4.1 Task delegation model and mobile/multi cloud middleware

In the task delegation, the mobile follows the SOA model to invoke the services from multiple cloud providers. Since most of the cloud providers offer web service interfaces for their services, this invocation is similar to the MWS client. Typical scenarios here include process intensive services such as face recognition and sensor mining, and data synchronization using approaches such as SyncML.¹⁰

The critical challenges in this space include addressing cloud interoperability and dealing with unavailability of standards and mobile platform specific API. To address these issues several middleware based solutions have been proposed [12, 26, 49]. They typically follow the workflow mediation and adapter patterns so that the invocation of services from multiple cloud providers is handled through adapters at the middleware and the services are composed using workflow execution in realizing mobile mashup applications [12, 26]. A typical example for task delegation based mobile cloud application is the CroudSTag (Social Tagging Crowd on Cloud) [45], which helps in taking

¹⁰ <http://technical.openmobilealliance.org/Technical/technical-information/material-from-affiliates/syncml>.

pictures and videos at events such as conferences with the mobile, stores and processes (using MapReduce) them on public/private cloud infrastructure to recognize and tag the people using the face recognition technologies, and later uses social networking solutions such as Facebook to form groups of people with specific interests. Several other interesting scenarios [26] can easily be envisioned and task delegation is shown to be a reality even in the IoT domain [9].

Mobile Hosts [40] and Mobile Cloud Middleware (MCM) [12, 26] can work in tandem in realizing an ideal P2P communication between the gateway nodes and the cloud. We further extended the idea of Mobile Hosts to also support workflow execution (SCORPII Mobile Host (ScoMH) [9]). This allows moving workflow tasks dynamically between the mobile and the cloud based middleware paving way for the adaptive workflow mediation. This comes in handy when dealing with specific issues of IoT, which is discussed further in Sect. 6.

4.2 Mobile code offloading

Mobile code offloading also known as cyber-foraging [5] primarily deals with partitioning the mobile application so that certain resource intensive operations can be offloaded for remote execution to cloud-based surrogates. The approach is similar to traditional componentized models such as Remote Procedure Call (RPC) [39] and Remote Method Invocation (RMI) [48], and have regained significant interest in the mobile cloud ecosystem. The major research challenges associated with mobile code offloading include *what, when, where* and *how* to offload [24]?

Typically, code and system profilers are employed to identify the resource intensive parts of an application and the mobile context, which will be analyzed by the offloading decision engine in determining the benefit of offloading. Conceptually, offloading is preferable only when computation to communication ratio is significantly high. There are several well-known frameworks which dealt with this problem. MAUI [15] approached the problem through manual annotations. ThinkAir [30] extended the approach to also include the scalability aspects of the cloud. Zhou et al [53] developed a context-aware offloading framework that takes advantage of heterogeneous mobile cloud ecosystem. Alternatively, CloneCloud [13] tried to automate the complete procedure. There are also approaches such as EMCO [25] which dealt with analyzing the historical traces of offloading and injecting the knowledge back to the decision engine to improve the forthcoming decisions.

Code-offloading is shown to work well mostly in controlled environments such as nearby servers, also known as Cloudlets [37]. However, not many real-time applications

exist to prove its adaptability. There are several challenges and technical problems which are posing obstacles in deploying code offloading. They include: (1) Inaccurate code profiling: Code has non-deterministic behavior during runtime and the execution depends on several factors such as input, type of the device, execution environment, CPU and memory usage etc., making the profiling inaccurate. (2) Integration complexity involved due to the need of the surrogate to have similar execution environment as the mobile. (3) The need to dynamically configure the offloading system. (4) The need to also consider the resource availability of the cloud in order to deal with offloading scalability [24].

However, the major problem for code offloading adaptation seems to be evident from Fig. 3. The figure shows the durations for offloading a chess game which is based on min-max algorithm to different types of cloud instances. From the figure we can observe that when the mobile is executing the game on i9300 (Samsung Galaxy S3), it is beneficial to offload only when the cloud surrogate is based on m3.medium or better capable instance of Amazon EC2. The m3.medium costs \$0.067 per Hour and it is not cost efficient to guarantee the continuous availability of the instance to the mobile user. Alternatively, we can rely on multi-tenancy to make it cost efficient, however, the scalability of offloading is not shown to be significant, especially due to the need of surrogate virtual machine with similar execution environment (e.g. Android-x86) [24]. In summary, as the device capabilities are increasing, the applications that can benefit from the code offloading are becoming limited.

This glooms the adaptability of code-offloading, especially when task delegation is already a reality. However, there are certain subtle differences which actually make the need for both the approaches. In delegation model, the task is generally assumed to take significant time. For example

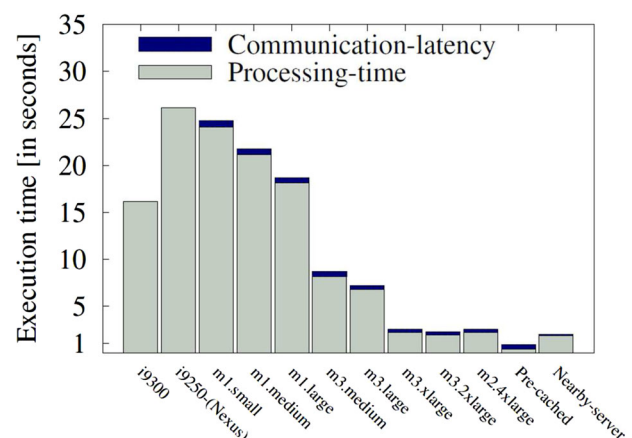


Fig. 3 Latencies of local execution and code offloading to different instance types

the CroudSTag application takes several minutes to finish the process at the middleware. When the result is ready it is sent asynchronously to the mobile using push notification mechanisms [50]. However, the task can't be executed at all, in the absence of Internet connectivity. Whereas, in code-offloading the execution times on the mobile and the surrogate do not differ drastically and the context plays a key role in offloading decision making. When there is no network connectivity or the engine decides not to offload, the method can perfectly run locally on the smart phone. However, significant research is to be conducted to address the defined challenges, in making code offloading reliable.

5 Remote cloud-based IoT data processing

Cloud computing with its promise of virtually infinite amount of resources, can easily be envisioned in IoT scenarios where the discussed middlewares are established on the cloud infrastructure and are properly auto-scaled. The main research challenges here include the dynamic deployment, ideal resource provisioning and auto-scaling. When we are considering billions of interconnected things, the interactions across the layers would probably be in trillions and we need to achieve proper auto-scaling mechanisms which can handle such huge loads and be able to adapt the deployment configuration dynamically. Cloud heterogeneity is a very interesting feature and a particular type of cloud instance can be ideal for a particular task and sometimes it may be ideal to have multiple numbers of different types of instances to support a particular load. Such adaptive resource provisioning can be achieved by employing proper linear programming (LP) based decision mechanisms [33, 44]. Apart from LP approaches, there exist other solutions based on technologies such as Reinforcement learning [19], Queuing theory [36] etc. There is also significant effort going on, in achieving the dynamic deployment on multiple clouds such as Cloud Modelling Language (CloudML) [23, 42].

Other major challenges in this space include the sensor data storage and processing. The amount of data collected from IoT based appliances would be in the order of Zeta-bytes (10^{21}) by 2020, which need to be properly stored, analyzed and interpreted and presented.¹¹ This brings in the scope for big data acquisition and analytics and parallel processing into the IoT ecosystem [22, 51]. Obvious first choice for the task would be MapReduce [17], a parallel programming model and an associated implementation for processing large data sets, which is also ideal for commodity based cloud infrastructure. However, MapReduce is

shown to be efficient only for embarrassingly parallel algorithms and has limitations with iterative algorithms [43]. In addition, IoT mostly deals with streaming data. So it is ideal to employ message queues such as Apache Kafka¹² to buffer and feed the data into stream processing systems such as Apache Storm¹³ or Apache Spark Streaming.¹⁴

Another major challenge associated here is related with security. Since IoT specifically deals with sensitive data such as health-related, location-based, e-home etc., it is necessary to ensure the trust, privacy, data protection and security. Significant work [29] is going on to address the challenges, however, traditional security solutions such as encryption can't be applied directly to IoT, due to the resource constrained and battery powered nature of physical objects. However, the issues can be addressed with reasonable effort beyond the gateway layer. So, it is ideal to achieve the anonymization and authorization of IoT data already at the gateway layer. Moreover, in certain cases (e.g. in health care and location based tracking scenarios) there may be restrictions on movement of the data across/beyond the organization/national boundaries, thus the data cannot be pushed to the public cloud layers.

To address these issues, a new paradigm is coming into existence, the fog computing [6, 46]. The fundamental idea of fog computing is to push the processing, data analytics and networking tasks to as low as possible in terms of the cloud based IoT layers, thus moving away from the centralized management, in a configurable way [32]. Fog computing lets the data processing, to the extent possible, on all the layers, including sensors/sinks, gateways, cloudlets, mobile edge, middlewares and public clouds, and in that order of preference.

6 Scenario and the layers working in tandem

Now that the layers and their communication has been discussed, let us consider an IoT scenario where all the layers work in tandem. Figure 4 illustrates a scenario where a disabled person, Alice, intends to avoid crowds in urban areas. She uses her mobile device to collect data from surrounding smart objects such as sensors that can provide noise level, temperature, brightness of a particular location, while she is walking on the street. Meanwhile, other Mobile Hosts in proximity can also provide various spatial content such as points of interest, current location-based text or media content, traffic information,

¹¹ <http://www.telecomengine.com/article/intelligent-data-handling-success-mantra-iot>.

¹² <https://kafka.apache.org/>.

¹³ <http://storm.apache.org/>.

¹⁴ <http://spark.apache.org/streaming/>.

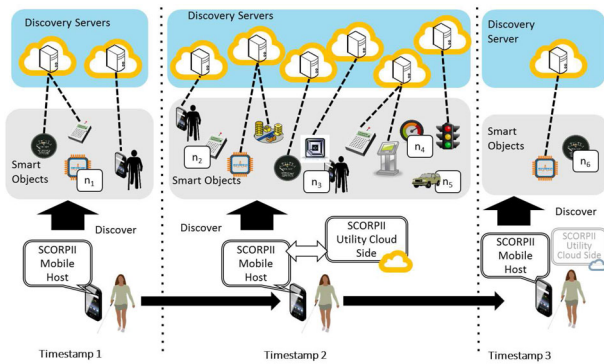


Fig. 4 IoT scenario of avoiding crowded urban areas by a disabled person

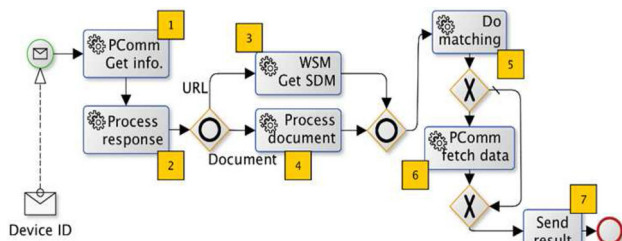


Fig. 5 Default service discovery workflow of the scenario (simplified)

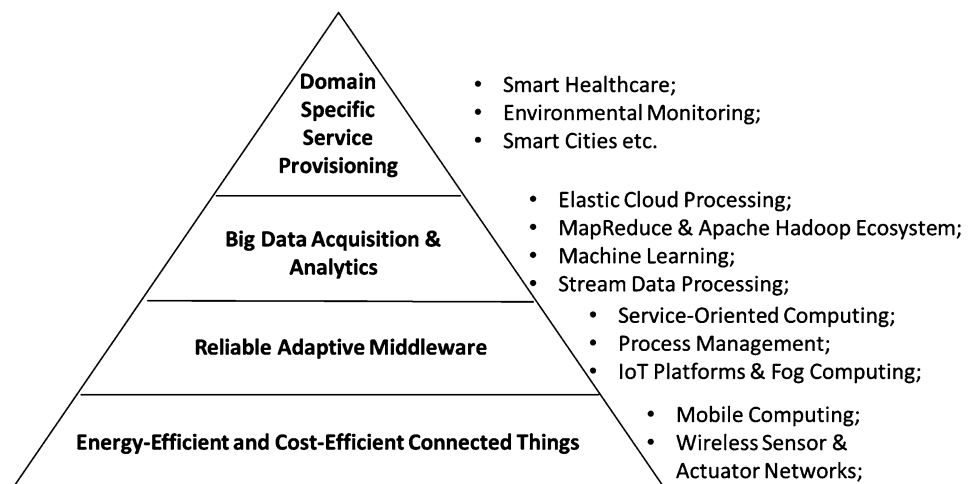
recommended travel route in the current location for disabled persons (e.g., n3 in Fig. 4) and so on.

The functionality of smart objects is described in service description metadata (SDM), which is retrievable either by direct P2P transmission using Bluetooth/Wi-Fi Direct, e.g. for nodes n1-n6 in Fig. 4 or via the discovery servers (assuming the smart objects can provide the addresses of discovery servers in some form such as RFID). Based on the SDM, the mobile device (SCORPII Mobile Host) can identify and retrieve the needed data from smart objects. By processing the collected data, Alice’s device can

provide a real time content mashup service to Alice to serve her purpose. Since the real time content mashup process requires interacting with the surrounding smart objects continuously and retrieving and processing the SDM from the discovery services, it can consume a lot of computational resources of the mobile device, if the number of smart objects is large (timestamp 2 in Fig. 4). In such scenarios the resource intensive process can be delegated to the cloud, based on some decision mechanism [9].

Figure 5 illustrates the default workflow of the use case. A localhost application (*reqApp*) in ScoMH browses smart objects in proximity and retrieves the names or IDs of the devices from their advertisements and sends them to the Workflow Manager of ScoMH. As Fig. 5 shows, each device ID is the input parameter of the workflow process. The Device ID will be forwarded to proximity communication service (PComm) and PComm will perform a simple GET request to the corresponding device (step 1). The response is analyzed for the type of the message (step 2). If it is a URL address, ScoMH will fetch the SDM from the respective discovery server (step 3). On the other hand, if the message contains a document, ScoMH will use a corresponding functional component to process the document to analyze if the document is a SDM (step 4). Afterwards, the SDM, which is retrieved from either step 3 or step 4, will be forwarded for service matchmaking (step 5). If the service described in SDM matches the requested type (in this case study, “current noise level”), the workflow manager will use PComm to retrieve the data from the corresponding device based on the URI described in SDM (step 6), otherwise, a simple “not match” response will be generated. Afterwards, the workflow service will send the result back to the *reqApp* (step 7). When there are lot of smart objects in proximity, steps 3 and 5 or the complete workflow of certain IDs can be delegated to the MCM. This dynamic migration of tasks between ScoMH and the

Fig. 6 Research roadmap of IoT



cloud based middleware is possible due to the support for adaptive workflow execution both at the mobile and the cloud [9].

In summary, this scenario shows how the IoT discovery can be made simple with the proposed layers [9]. Similarly, we can easily envision a scenario where gateway devices from several organizations mutually co-ordinate in saving energy of the respective physical objects and the gateways [8]. In this regard, it is also worth to notice that, IoT devices are also being used in business process management (BPM) [18], where the discussed adaptive mediation frameworks will have significant role to play [10].

7 Conclusions and future research directions

IoT represents a comprehensive environment that consists of a large number of sensors and mediators interconnecting heterogeneous physical objects to the Internet. Primarily, the challenges associated with realization of IoT scenarios can be summarized across the layers sensing and smart devices layer, connectivity layer and cloud layer. The paper tried to address the challenges in IoT with mobile web and cloud services and explained a scenario showing the layers work in tandem. In the process, we also have discussed the state of the art of each of the respective research domains along with further scope for extensions and recent trends. Figure 6 summarizes the research roadmap of IoT, along with the relevant technology ecosystem.

Acknowledgements The paper is based on the keynote talk delivered by the author at the International Conference of Advanced Computing (ICAC-2016). The keynote was earlier delivered at 8th IEEE/ACM International Conference on Utility and Cloud Computing (UCC 2015). The author would like to thank his colleagues and co-authors at Mobile & Cloud Lab, Chii Chang, Huber Flores, Carlos Paniagua, Pelle Jakovits, Mohan Liyanage, Jakob Mass, Alireza Ostovar and Jaagup Viil, who have contributed in achieving the respective results. This research is supported by the Estonian Research Council grant PUT360.

References

1. Aghaei S, Nematbakhsh MA, Farsani HK (2012) Evolution of the world wide web: from web 1.0 to web 4.0. *Int J Web Semant Technol* 3(1):1
2. Ahonen T (2008) Mobile as 7th of the mass media. Cellphone, Cameraphone, iPhone, Smartphone. *Futuretext*
3. AlShahwan F, Moessner K (2010) Providing soap web services and restful web services from mobile hosts. In: 2010 Fifth international conference on internet and web applications and services (ICIW). IEEE, pp 174–179
4. Armbrust M, Fox A, Griffith R, Joseph AD, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A, Stoica I et al (2010) A view of cloud computing. *Commun ACM* 53(4):50–58
5. Balan R, Flinn J, Satyanarayanan M, Sinnamohideen S, Yang HI (2002) The case for cyber foraging. In: Proceedings of the 10th workshop on ACM SIGOPS European workshop. ACM, pp 87–92
6. Bonomi F, Milito R, Zhu J, Addepalli S (2012) Fog computing and its role in the internet of things. In: Proceedings of the first edition of the MCC workshop on mobile cloud computing. ACM, pp 13–16
7. Castellani AP, Gheda M, Bui N, Rossi M, Zorzi M (2011) Web services for the Internet of Things through CoAP and EXI. In: 2011 IEEE international conference on communications workshops (ICC). IEEE, pp 1–6
8. Chang C, Loke SW, Dong H, Salim F, Srirama SN, Liyanage M, Ling S (2015) An energy-efficient inter-organizational wireless sensor data collection framework. In: 2015 IEEE international conference on web services (ICWS). IEEE, pp 639–646
9. Chang C, Srirama S, Mass J (2015) A middleware for discovering proximity-based service-oriented industrial Internet of Things. In: Proceedings of the 2015 IEEE international conference on services computing (SCC). IEEE, New York, pp 130–137
10. Chang C, Srirama SN, Buyya R (2016) Mobile cloud business process management system for the internet of things: a survey. *ACM Comput Surv (CSUR)*. doi:10.1145/3012000
11. Chang C, Srirama SN, Ling S (2014) Towards an adaptive mediation framework for mobile social network in proximity. *Pervasive Mobile Comput* 12:179–196
12. Chang C, Srirama SN, Liyanage M (2015) A service-oriented mobile cloud middleware framework for provisioning mobile sensing as a service. In: 2015 IEEE 21st international conference on parallel and distributed systems (ICPADS). IEEE, Melbourne, pp 124–131
13. Chun BG, Ihm S, Maniatis P, Naik M, Patti A (2011) Clonecloud: elastic execution between mobile device and cloud. In: Proceedings of the sixth conference on computer systems. ACM, pp 301–314
14. Conti M, Delmastro F, Minutiello G, Paris R (2013) Experimenting opportunistic networks with WiFi Direct. In: *Wireless Days (WD)*, 2013 IFIP. IEEE, pp 1–6
15. Cuervo E, Balasubramanian A, Cho Dk, Wolman A, Saroiu S, Chandra R, Bahl P (2010) Maui: making smartphones last longer with code offload. In: Proceedings of the 8th international conference on mobile systems, applications, and services. ACM, pp 49–62
16. Curbera F, Duftler M, Khalaf R, Nagy W, Mukhi N, Weerawarana S (2002) Unraveling the web services web: an introduction to SOAP, WSDL, and UDDI. *IEEE Internet Comput* 6(2):86
17. Dean J, Ghemawat S (2008) Mapreduce: simplified data processing on large clusters. *Commun ACM* 51(1):107–113
18. Dumas M, La Rosa M, Mendling J, Reijers HA et al (2013) *Fundamentals of business process management*, vol 1. Springer, Berlin
19. Dutreilh X, Kirgizov S, Melekhova O, Malenfant J, Rivierre N, Truck I (2011) Using reinforcement learning for autonomic resource allocation in clouds: towards a fully automated workflow. In: 7th international conference on autonomic and autonomous systems (ICAS 2011), pp 67–74
20. Erl T (2008) *SOA: principles of service design*, vol 1. Prentice Hall, Upper Saddle River
21. Evans D (2011) The internet of things: how the next evolution of the internet is changing everything. *CISCO White Pap* 1:14
22. Fan W, Bifet A (2013) Mining big data: current status, and forecast to the future. *ACM SIGKDD Explor Newsl* 14(2):1–5
23. Ferry N, Chauvel F, Rossini A, Morin B, Solberg A (2013) Managing multi-cloud systems with CloudMF. In: Proceedings of the second Nordic symposium on cloud computing & internet technologies. ACM, pp 38–45

24. Flores H, Hui P, Tarkoma S, Li Y, Srirama S, Buyya R (2015) Mobile code offloading: from concept to practice and beyond. *IEEE Commun Mag* 53(3):80–88
25. Flores H, Srirama S (2013) Adaptive code offloading for mobile cloud applications: exploiting fuzzy sets and evidence-based learning. In: *Proceeding of the fourth ACM workshop on mobile cloud computing and services*. ACM, pp 9–16
26. Flores H, Srirama SN (2014) Mobile cloud middleware. *J Syst Softw* 92:82–94
27. Gomez C, Paradells J (2010) Wireless home automation networks: a survey of architectures and technologies. *IEEE Commun Mag* 48(6):92–101
28. Gubbi J, Buyya R, Marusic S, Palaniswami M (2013) Internet of Things (IoT): a vision, architectural elements, and future directions. *Future Gener Comput Syst* 29(7):1645–1660
29. Jing Q, Vasilakos AV, Wan J, Lu J, Qiu D (2014) Security of the internet of things: perspectives and challenges. *Wirel Netw* 20(8):2481–2501
30. Kosta S, Aucinas A, Hui P, Mortier R, Zhang X (2012) Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading. In: *INFOCOM, 2012 proceedings IEEE*. IEEE, pp 945–953
31. Liyanage M, Chang C, Srirama SN (2015) Lightweight mobile web service provisioning for sensor mediation. In: *2015 IEEE international conference on mobile services (MS)*, pp 57–64. doi:10.1109/MobServ.2015.18
32. Liyanage M, Chang C, Srirama SN (2016) mePaaS: mobile-embedded platform as a service for distributing fog computing to edge nodes. In: *17th international conference on parallel and distributed computing, applications and technologies (PDCAT-16)*
33. Mao M, Li J, Humphrey M (2010) Cloud auto-scaling with deadline and budget constraints. In: *2010 11th IEEE/ACM international conference on grid computing (GRID)*. IEEE, pp 41–48
34. Pashtan A (2005) *Mobile web services*. Cambridge University Press, Cambridge
35. Perera C, Zaslavsky A, Christen P, Georgakopoulos D (2014) Sensing as a service model for smart cities supported by Internet of Things. *Trans Emerg Telecommun Technol* 25(1):81–93
36. Salah K, Elbadawi K, Boutaba R (2016) An analytical model for estimating cloud resources of elastic services. *J Netw Syst Manag* 24(2):285–308. doi:10.1007/s10922-015-9352-x
37. Satyanarayanan M, Bahl P, Caceres R, Davies N (2009) The case for VM-based cloudlets in mobile computing. *IEEE Pervasive Comput* 8(4):14–23
38. Shelby Z, Hartke K, Bormann C (2014) The constrained application protocol (CoAP). Technical report. <https://www.rfc-editor.org/rfc/rfc7252.txt>
39. Srinivasan R (1995) RPC: remote procedure call protocol specification version 2. RFC 1831, IETF. <https://datatracker.ietf.org/doc/rfc1831/>. Accessed 17 Dec 2016
40. Srirama S, Jarke M, Prinz W (2006) Mobile web service provisioning. In: *International conference on internet and web applications and services (ICIW 2006)*, pp 120–125
41. Srirama SN (2008) *Mobile hosts in enterprise service integration*. Ph.D. thesis, RWTH Aachen University, Germany
42. Srirama SN, Iurii T, Viil J (2016) Dynamic deployment and auto-scaling enterprise applications on the heterogeneous cloud. In: *9th IEEE international conference on cloud computing*. IEEE, pp 927–932
43. Srirama SN, Jakovits P, Vainikko E (2012) Adapting scientific computing problems to clouds using mapreduce. *Future Gener Comput Syst* 28(1):184–192
44. Srirama SN, Ostovar A (2014) Optimal resource provisioning for scaling enterprise applications on the cloud. In: *The 6th IEEE international conference on cloud computing technology and science (CloudCom-2014)*
45. Srirama SN, Paniagua C, Flores H (2011) Croudstag: social group formation with facial recognition and mobile cloud services. *Procedia Comput Sci* 5:633–640
46. Vaquero LM, Rodero-Merino L (2014) Finding your way in the fog: towards a comprehensive definition of fog computing. *ACM SIGCOMM Comput Commun Rev* 44(5):27–32
47. Wagoner A (2015) These android phones can handle wireless charging. <http://www.androidcentral.com/these-android-phones-can-handle-wireless-charging>
48. Waldo J (1998) Remote procedure calls and java remote method invocation. *IEEE Concurr* 6(3):5–7
49. Wang Q, Deters R (2009) Soa’s last mile-connecting smartphones to the service cloud. In: *2009 IEEE international conference on cloud computing*. IEEE, pp 80–87
50. Warren I, Meads A, Srirama S, Weerasinghe T, Paniagua C (2014) Push notification mechanisms for pervasive smartphone applications. *IEEE Pervasive Comput* 13(2):61–71
51. Zaslavsky A, Perera C, Georgakopoulos D (2012) Sensing as a service and big data. In: *International conference on advances in cloud computing (ACC-2012)*, Bangalore, pp 21–29
52. ZigBee Alliance (2004) ZigBee specification, V1.0. <http://home.deib.polimi.it/cesana/teaching/IoT/papers/ZigBee/ZigBeeSpec.pdf>. Accessed 17 Dec 2016
53. Zhou B, Dastjerdi AV, Calheiros RN, Srirama SN, Buyya R (2015) A context sensitive offloading scheme for mobile cloud computing service. In: *2015 IEEE 8th international conference on cloud computing*. IEEE, pp 869–876