

Chapter 2

The Resource Management Challenge in IoT

Abstract The purpose of this book is to discuss the issues regarding resource management in Internet of Things (IoT) from a holistic standpoint. In order to build the foundation on which our discussion will be conducted, in this chapter we first introduce some basic definitions that are relevant to our study, such as resource, resource management and IoT ecosystem. Then, we identify the main requirements and challenges related to resource management in the specific context of IoT. Finally, we describe the typical activities involved in the all-embracing resource management process according to our proposed holistic view. For all purposes, we will consider throughout this book a generic scenario of an IoT ecosystem consisting of several heterogeneous interconnected devices, whose data and services (virtual and physical resources) are used by several different applications that access such pool of resources via network. IoT devices include both resource-constrained and resource-rich devices. We consider resource-rich devices as those that have the hardware and software capability to support the TCP/IP protocol suite. Besides IoT devices, the IoT ecosystem also includes gateways, edge nodes and cloud data centres.

Keywords Internet of Things (IoT) • Resource management • Resource allocation • Resource discovery • Resource modelling

2.1 What Is a Resource in the Context of IoT Ecosystems?

Internet of Things (IoT) ecosystems are complex environments encompassing many heterogeneous components. The huge amount of data generated by sensor-instrumented objects of the real world in an IoT ecosystem will impose a great demand for processing and storage resources to be transformed into useful information or services. Some applications will be latency sensitive, while other applications will require complex processing including historical data and time series analyses. Therefore, considering the typical resource constraints of IoT nodes, it is difficult to envision a real-world, ultra-scale IoT ecosystem without

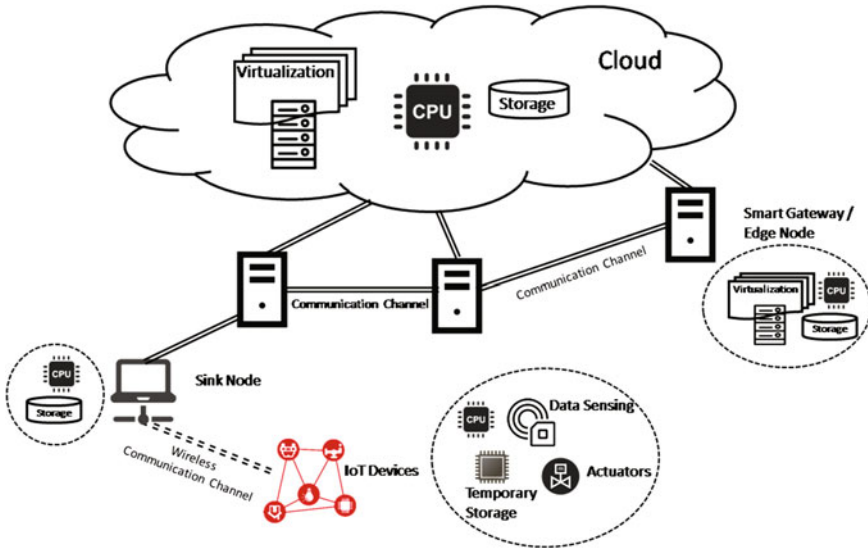


Fig. 2.1 Example of resources in an IoT ecosystem

including a cloud platform, or at least some powerful devices, for instance Smart Gateways [1] or edge/fog nodes [2]. In this complex scenario of IoT edge cloud, the definition of resource may range from physical resources, such as memory (storage), CPU, network bandwidth, energy, etc., to software resources. Procedures to perform information fusion, or to detect a complex event, or a virtualization function, are all examples of software resources. The model adopted to formally define the resource management process will depend on the definition for resource itself. So, in the same way as we are taking a holistic approach to address the resource management process, we will adopt a broad, generic and all-encompassing definition for resource:

A resource is any object which can be allocated within a system [3].

Figure 2.1 represents an example of the resources identified in a three-tier IoT ecosystem composed of cloud, edge and IoT devices.

2.2 Key Requirements of IoT

Before starting our discussion of resource management for IoT, it is important to have a minimum understanding of the peculiarities of such environment that make it so special, requiring novel and specific solutions for such a traditional problem that has been extensively studied in computer systems. One of the goals of this book is to discuss, in depth, the specific requirements of IoT in this context, and to

analyse how some existing solutions are addressing them. Therefore, we will revisit the IoT features later, at the light of the acquired knowledge on resource management. The idea of this chapter is to present a preliminary discussion to motivate for the need of new and tailored solutions.

The IoT can be seen at first glance as a sort of large-scale distributed system where the components have a high degree of **heterogeneity** with respect to hardware and software. In addition, the execution context of these systems is extremely dynamic. *Context* here has a broad meaning and it refers from the status of software components embedded in a device to the user's geographic location (which may vary in the presence of mobility) and his/her personal agenda. **Context awareness** is recognized as an important requirement of IoT systems [4] to properly accommodate the scale and heterogeneity factors and to provide useful information to meet application demands. Besides helping to build adaptive IoT systems that better fit the dynamic application needs and execution context, context awareness plays a key role in IoT to decide what data needs to be processed, based on its relevance to a given context [4].

The heterogeneity, scale and context awareness factors alone already make the resource management a very challenging issue. However, other classes of distributed systems such as clouds and ubiquitous systems share such features. We claim that the use of IoT ecosystems as infrastructure for running applications distinguishes from traditional practices in distributed systems, for the following main reasons. In dedicated distributed systems, the application software runs over infrastructures often dimensioned according to the worst case and peak scenarios. More recently, in cloud computing systems, although the service provision follows a *pay per use*, dynamic and elastic model, the application requirements are usually preestablished via formal or semiformal contracts between the client and the cloud provider. Therefore, resource allocation mechanisms in cloud platforms employ sophisticated strategies and algorithms to better allocate physical or virtual resources to applications, meeting contracts-based predefined application requirements. At runtime, such mechanisms monitor the status of the infrastructure to accommodate unforeseen demands in a scalable and elastic way, while respecting the contracts. In IoT, all the requirements of a resource allocation mechanism for cloud computing still hold, but some additional requirements emerge, as we will discuss next.

One major requirement is related to the ad hoc nature of IoT. In IoT scenarios, there can be opportunistic, ad hoc interactions among devices and users, leveraged by some specific contexts. For instance, a mobile device can make its resources available only for users that are in its neighbourhood for a given period of time. There is no sense in establishing any type of formal contract in this case. Instead, the device owner can have some type of incentive for making its resources available for other clients. It is also noticeable that in such scenarios, similarly to P2P networks, a same user can be a provider for a certain service while, at the same time, a client for services running in other devices. In this context, IoT systems do not fully adhere to the traditional contract-based client-server model assumed by several service-oriented distributed systems. Therefore, application requirements in

terms of quality of service (QoS) are not always guaranteed to be met by the infrastructure. It is more likely to have a mixed model of service provision, in which parts of such requirements, for instance the ones being served by a cloud-integrated IoT platform, are regulated by informal contracts to some extent, while part of them just inherits the Internet best-effort model. The provision of these resources can be partially controlled by one or more service providers, while partially provided in an ad hoc, opportunistic fashion that is highly dependent on the current state and availability of the interconnected devices. Such feature of exploiting opportunistic interactions for the service provision in IoT makes the process of resource allocating and management more complex than in traditional cloud computing systems. Moreover, the establishment of a pricing model also becomes much more intricate for the service providers.

Another key requirement is real-time processing. IoT systems potentially handle hundreds, thousands or millions of parallel requests and several types of applications demand fast response, within a strict time interval. In IoT, multiple applications with potentially different requirements will be sharing the same resources. A time critical application will demand some level of priority in the access to the shared resources, in detriment of noncritical applications, to guarantee its requirements will be met. Moreover, the nature of data produced by IoT devices also affects its processing. Sensors generate, possibly in a continuous way, a huge amount of data, typically consisting of time series values, which are sampled over a specific time period, thus characterizing a data stream. The input rate of a data stream ranges from a few bytes per second to a few gigabits per second. Such input rate can be irregular, unpredictable and bursty in nature. The inherent nature of data streams does not allow one to easily make multiple passes over a stream while processing it (still retaining the usefulness of the data). Data stream processing often requires online solutions, in which data is processed and valuable information is acquired on the fly, without having the complete view of the data and without using past information. Real-time and online processing poses different requirements in terms of resources, in comparison to applications running on traditional cloud platforms.

Solutions for managing the resources involved in processing IoT data and delivering IoT services in a cost-effective and efficient way need to take these key requirements into account, besides further needs to be revealed during the discussion of the current literature used as basis for this book.

2.3 Key Activities for Resource Management in IoT

In order to describe the core activities of a generic framework for resource management in IoT, we consider a hypothetical model for an IoT ecosystem with either three or two tiers (Fig. 2.2), in which: (i) the *bottom tier* encompasses the things (IoT devices/nodes/smart objects), (ii) the *top tier* encompasses cloud nodes and (iii) an optional *middle tier* consists in Smart Gateways or edge nodes. Applications

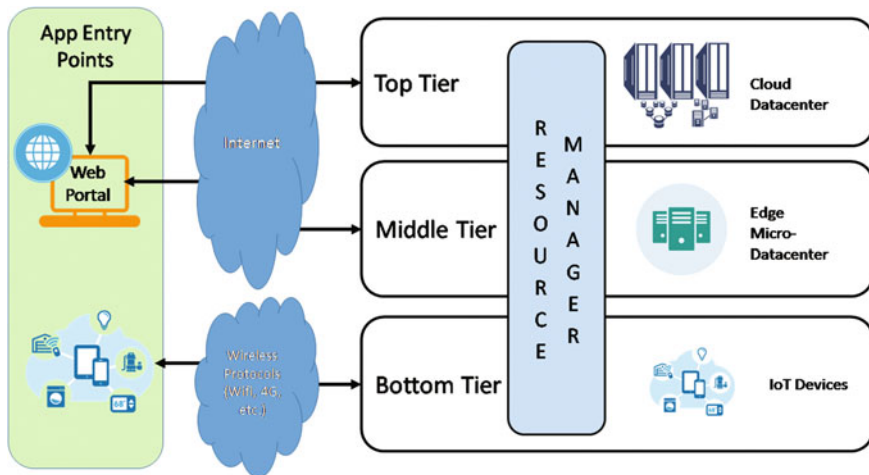


Fig. 2.2 RMS layers in IoT systems

access the system from different application entry points (AEP), sending their requests which are expected to be met by the IoT infrastructure. These entry points may themselves be physical devices which are part of the IoT bottom tier, for example, smart phones (a resource-rich IoT device), or they may be personal computers running Web portals (and thus accessing the system through the cloud) or even gateway or edge node.

Besides our hypothetical physical model organized in two or three tiers, we also assume a logical layered architecture for IoT systems. According to the authors in [5], a typical five-layer architecture for IoT encompasses: the objects layer (the lowermost in the system, representing the physical objects), the object abstraction, the service management, the application and the business layer. We claim that a resource management layer (RML) is orthogonal to the four uppermost layers proposed in [5]. The RML is responsible for all activities related to the resource management of the system. The RML will be implemented as a resource manager (RM) subsystem to be deployed in a distributed way among different hardware components in the two or three tiers of the IoT system (Fig. 2.2).

As discussed in [6], the main challenge regarding a resource management layer (RML) in the context of cloud computing is to perform the automated provisioning of resources. This is the same goal in IoT, but considering additional requirements, as aforementioned. The ultimate goal of the RML is deciding the best scheme of resource allocation according to the up-to-date system information so as to obtain the maximum utilization of the system resources. This function requires various strategies to engage the resources that meet the formally or informally established application QoS requirements. Although the resource allocation is the core of a RML, there are further activities that support such activity to enable the proper and continuous operation of the system. We identify the following activities as the main

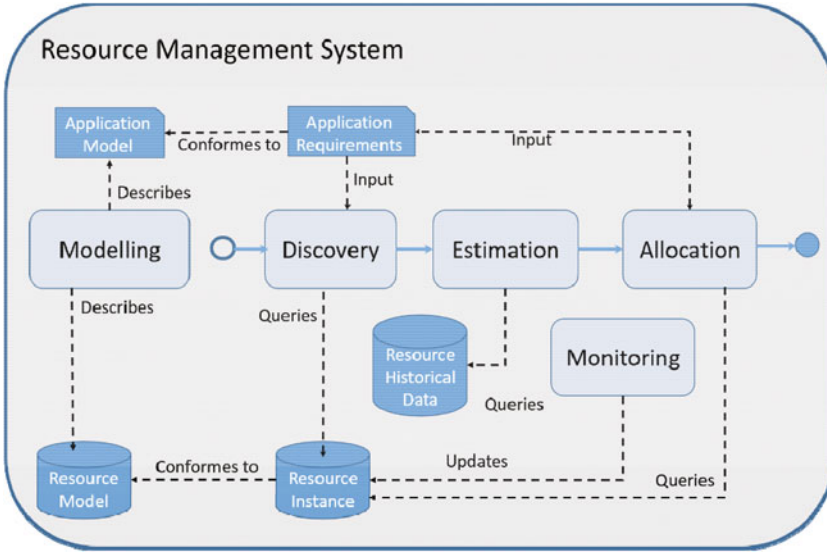


Fig. 2.3 Activities involved in the resource management

components of a typical workflow for a RMS in IoT: resource modelling, resource discovery, resource estimation, resource allocation and resource monitoring (Fig. 2.3).

The main goal of this book is to discuss all the activities encompassed in the RML, with an emphasis on the resource allocation. In the following, we briefly describe such activities, which will then be presented in the next chapters, in the light of proposals currently existing in the literature. It is important to mention that the literature search used as the starting point for the discussions presented in this book focused on works addressing resource management for IoT. The keywords used in the performed search did not explicitly include resource discovery, modelling or estimation. They are considered as support activities for the general process of resource management. Works focusing solely on such activities, without their integration in a more general framework for resource management, are not discussed in this book.

2.3.1 Resource Modelling

The first issue about the resource management process is how applications and resource management systems describe resources in an IoT ecosystem. The resource model is a vital part of any RML since it defines the entities, properties and relationships that build up the system, thus driving the whole operation of the resource manager. Moreover, using high-level models to represent resources

facilitates dealing with the inherent heterogeneity of IoT resource providers. Resource models can be regarded as the scheme or metadata for describing the resources within a system. They are typically created by designers of IoT systems and are used by the resource management systems and by IoT applications for different purposes. Therefore, an important feature of modelling solutions is the ease with which (i) designers can translate IoT resources to the available modelling constructs, (ii) the applications can describe their needs in terms of the resources and (iii) the RMS can, at runtime, use and manipulate resource information for its decision-making processes.

The resource model must properly represent the elements from the different tiers of an IoT ecosystem, ranging from low-level IoT devices and hardware elements (e.g. CPU and memory) to high-level service interfaces. Moreover, not only physical, but also virtual resources need to be represented. Houidi et al. [7] advocate that virtual resources require to be described according to their properties and functionalities similarly as services are described in existing service architectures.

Each tier in an IoT system can have its own modelling requirements leading to different representation languages/formats. For instance, in order to favour interoperability, IoT devices can be represented through Web technologies such as the SSN ontology [8] for annotating sensors and sensor networks, and Linked Data [9] for sensor data publishing and discovery. Network and computing resources may be represented using existing specifications, such as Network Description Language (NDL) [10, 11].

IoT resource and services can be described with different levels of abstraction for end users/developers, and different parameters can be exposed to be tuned for optimization purposes during allocation. The granularity used to represent and expose resources is also an important issue in resource modelling [12]. For instance, a resource management system can use a particular representation model that exposes a very detailed (low-level) description of resources to the applications, giving more flexibility and allowing for a better customization of resource usage (to the applications). In this case, this flexibility comes at the cost of a harder optimization problem to be solved by the RMS. For example, if an application request is defined based on the physical specifications of each machine, the optimal matching of the request with available resources becomes harder to achieve since the possible optimization solution space is narrowed by the request itself. Finally, a resource model can be described using a formal, semiformal or informal notation. The adoption of more formal representations of resources provides more rigour to the model and allows using automatized tools for consistency checking, for verifying some properties of the model and also to exploit techniques such as reasoning, during the activity of resource discovery. Considering all these features, existing proposals for resource modelling can be analysed according to their degrees of abstraction, formalism and granularity. Other relevant characteristic are their expressivity and flexibility.

Once the resource model is built according to the selected notation, the IoT infrastructure can expose the (physical or virtual) resources for the consumer applications. IoT applications are built upon such resources, and are ultimately the

triggers for the resource consumption. Applications can be specified by using a programmatic or declarative API, or using some high-level domain-specific language (DSL). Therefore, besides the **resource model**, the RML also needs to encompass a proper **application model**, which is closely related to the former. A mapping process should usually take place to translate application requirements to definitions and commands understandable by the entities that will execute the requested services (thus consuming resources). Depending on the adopted notation, the process of translating the application requests to a machine-readable specification will be more or less complex. Independently on the adopted approach, after a mapping process, an application will typically contain (i) a *set of descriptive functional requirements*, such as, for instance, the geographical location of interest, type of sensing data (along with optional information on data rate and/or time interval of interest), events to be detected or monitored, actions to be executed upon occurrence of events and (ii) an *optional set of nonfunctional (or QoS) attributes*, such as data accuracy, maximum end-to-end delay, data freshness and maximum monetary cost to be paid for the usage of the IoT infrastructure. Regarding the QoS attributes, applications usually require that the data produced by IoT devices are transmitted or processed according to some constraints, which can be strict in case of critical scenarios (for instance, time critical applications). Therefore, for some classes of applications, QoS requirements are very important and there may be a need of using some SLA enforcement to assure their meeting. On the other hand, some IoT applications are built upon opportunistic, ad hoc interactions, which preclude any formal agreements and the service is provided on a basis of a best-effort model.

Existing proposals addressing the resource modelling activity are presented and discussed in Chap. 3.

2.3.2 Resource Allocation

The processing of an application will incur in a workload for the IoT system and will produce one or more outputs. Such workload can be interpreted as the amount of resources needed to accomplish the specific tasks required by the application (e.g. sense, process and transmit 50 temperature samples, or acquire process and transmit one JPEG image, etc.). The workload may encompass the memory (e.g. bytes) and processing load (e.g. MIPS) consumed by the application, the use of sensing devices and of network bandwidth. The outcome of the application workload processing can be as simple as a scalar value (for instance, the current temperature of a given point in space), a detected event (presence of fire or occurrence of a temperature above a threshold) or it can be a preprocessed data stream continuously generated, to be used to feed some complex stream processing system.

The final goal of the resource allocation activity, which is the responsibility of a **resource allocation system** (RAS) within the RML, is to properly accommodate

the workload of all the applications currently using the IoT system, by allocating the required (virtual or physical) resources so that expected outcomes of all the applications are provided and the QoS requirements are met. This implies identifying the several fine-grained execution units that compose an application that will produce the respective workload. Then, such execution units need to be distributed among the elements of the system, preferably in a fair, balanced way, so that the overall utilization of the system resources is optimized while satisfying the needs of simultaneous applications. The main inputs of such a resource allocation system are (i) the resource modelling (the schema used to describe the types of resources existing in the whole IoT system), (ii) current status of IoT resources (the resource instances along with their instantaneous utilization) and (iii) the application modelling, encompassing the set of functional and nonfunctional requirements.

Figure 2.4 depicts the activities involved in the resource allocation for IoT systems. The first step in a resource allocation is *planning*, in which a global view of the available resources is analysed to verify if the IoT system as a whole can accommodate the application requirements. Moreover, considering that the IoT system comprises multiple tiers (things, cloud and edge tiers), the planning includes the decision about which tiers are to be engaged in the execution. Next, the individual execution units (or tasks) need to be provided with the specific resources required for their execution. This step is basically a *task mapping* process, where a selected element (node) of the IoT system will be assigned a given task. There may be collaboration among entities of the IoT system to complete the required tasks of an application. Moreover, there are dependencies among different tasks of a same application that need to be considered. Therefore, as another step of the resource allocation process, the *temporal scheduling of computational tasks* should be determined respecting the application time constraints and the dependencies between tasks. Finally, it is necessary to *schedule the communication* among the entities participating in the execution of tasks on available communication

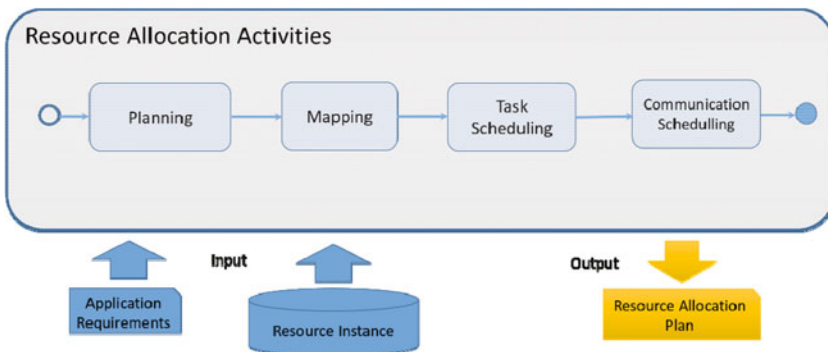


Fig. 2.4 Resource allocation activities

channels. Task mapping and scheduling must be done in an efficient and cost-effective way for both the providers and the clients (end users and applications).

Considering the dynamicity of the IoT scenario and the different requirements of several concurrent applications accessing the system, it is clear that resource allocation is not a trivial issue in IoT. It is typically considered as an optimization problem, and multiple techniques have been employed to solve it, pursuing different performance goals and meeting different constraints. It is important to mention that resources can be either provided by physical elements or by virtualized entities. When virtual entities are involved in the resource allocation, the decision about the creation/instantiation of such virtual entities is part of the typical process of resource provision in cloud systems.

This book has a special emphasis on discussing existing proposals for the resource allocation activity. Such discussion is presented in Chap. 6.

2.3.3 Resource Discovery, Monitoring and Estimation

Before the necessary resources can be allocated, they need to be discovered in the IoT ecosystem. Resource discovery in such a heterogeneous system is in itself a challenge, worsened by the current lack of standardization of protocols and formats in the field. Moreover, the high scalability of the IoT requires the RMS to provide some mechanism able to register and discover resources and services in a self-configured, efficient and dynamic way. In the traditional Internet, the DNS service discovery protocol (DNS-SD) provides a way of using standard DNS programming interfaces, servers and packet formats to browse the network for services. Although such protocol has been originally designed originally for resource-rich devices, there are some proposals for lightweight versions tailored for IoT environments, such as [13]. In Chap. 4 we discuss some works developed in the context of resource management that include service discover mechanisms.

Another activity that is necessary so that a RMS tackles the highly dynamic nature of IoT systems is resource monitoring. The execution environment is extremely dynamic, including variations related to the user, the network, the physical environment and the devices. The monitoring of these environmental variations is essential to provide a high-quality service. Moreover, the monitoring about current resource usage is also necessary for the purpose of keeping an optimum or near to optimum resource allocation. This implies that resource management for IoT should be context-aware, and the allocation process cannot be static, but rather dynamic and adaptive to accommodate such variations in the execution context. Solutions for detecting environmental changes and adapt to them will enable the delivery of enhanced context-based services, helping to provide the more cost-effective service provision and resource usage depending on the situation.

Finally, sometimes it is useful trying to estimate the amount of resources to be used to better assure the successful completion of the application. Strategies for resource estimation are usually based on keeping historical data of the consumption. Such data is obtained through the monitoring activity. In cases where there is an estimation and possibly reservation of resources, there may be eventually non-utilized resources (for instance, because there was over provisioning or because the execution conditions changed). Therefore, it may be necessary for the system to be able to reclaim such non-utilized resources, returning them to the pool of available resources. Chapter 5 discusses some proposals for resource estimation in IoT. Proposals that exploit the activity of resource monitoring with the goal of estimating resources to be consumed based on historical data (obtained through the monitoring) are also described in such Chapter. Proposals that leverage the monitoring activity to augment the efficacy of the resource allocation strategy, endowing it with context-aware adaptation features are discussed as part of the resource allocation activity in Chap. 6.

In the following chapters, we present a thorough discussion on each issue related to our holistic view of resource management in IoT, presenting the challenges involved in each activity and existing proposals to tackle them. Our discussions are grounded on the results of a painstaking review of the state of the art in this area, carried out by our research group.

References

1. Aazam M, Huh E-N (2014) Fog computing and smart gateway based communication for cloud of things. In Future Internet of Things and Cloud (FiCloud), 2014 International conference on, 2014. IEEE, pp 464–470
2. Garcia Lopez P, Montresor A, Epema D, Datta A, Higashino T, Iammitchi A, Barcellos M, Felber P, Riviere E (2015) Edge-centric computing: vision and challenges. ACM SIGCOMM Comput Commun Rev 45(5):37–42
3. Tanenbaum AS, Woodhull AS (1992) Operating system concepts. Learning 2:3
4. Perera C, Zaslavsky A, Christen P, Georgakopoulos D (2014) Context aware computing for the internet of things: a survey. IEEE Commun Surv Tutorials 16(1):414–454. doi:[10.1109/SURV.2013.042313.00197](https://doi.org/10.1109/SURV.2013.042313.00197)
5. Al-Fuqaha A, Guizani M, Mohammadi M, Aledhari M, Ayyash M (2015) Internet of things: a survey on enabling technologies, protocols, and applications. IEEE Commun Surv Tutorials 17(4):2347–2376
6. Zhang Q, Cheng L, Boutaba R (2010) Cloud computing: state-of-the-art and research challenges. J Internet Serv Appl 1(1):7–18
7. Houidi I, Louati W, Zeghlache D (2008) A distributed virtual network mapping algorithm. In 2008 IEEE international conference on communications. IEEE, pp 5634–5640
8. Compton M, Barnaghi P, Bermudez L, García-Castro R, Corcho O, Cox S, Graybeal J, Hauswirth M, Henson C, Herzog A (2012) The SSN ontology of the W3C semantic sensor network incubator group. Web Semant: Sci Serv Agents on the World Wide Web 17:25–32
9. Bizer C, Heath T, Berners-Lee T (2009) Linked Data-the story so far. Int J Semant Web Inf Syst 5(3):1–22
10. Beckett D, McBride B (2004) RDF/XML syntax specification (revised). W3C Recommendation 10

11. Van der Ham JJ, Dijkstra F, Travostino F, Andree HM, de Laat CT (2006) Using RDF to describe networks. *Future Gener Comput Syst* 22(8):862–867
12. Endo PT, de Almeida Palhares AV, Pereira NN, Goncalves GE, Sadok D, Kelner J, Melander B, Mangs J-E (2011) Resource allocation for distributed cloud: concepts and research challenges. *IEEE Netw* 25(4):42–46
13. Jara AJ, Martinez-Julia P, Skarmeta (2012) A Light-weight multicast DNS and DNS-SD (ImDNS-SD): IPv6-based resource and service discovery for the Web of Things. In: *innovative mobile and internet services in ubiquitous computing (IMIS)*, 2012 sixth international conference on, 2012. IEEE, pp 731–738

Resource Management for Internet of Things

Delicato, F.C.; F.Pires, P.; Batista, T.

2017, VIII, 116 p. 6 illus., 4 illus. in color., Softcover

ISBN: 978-3-319-54246-1