From Machine-to-Machine to the Internet of Things
Introduction to a New Age of Intelligence
By Jan Holler
Elsevier, 2014

4.3 An IoT architecture outline

We have now arrived at a better understanding of the design objectives and principles that capture the main desired characteristics of an M2M or IoT solution, and we have also identified some high-level capabilities that generally are needed. As described above, there is a rather widely accepted view of what a typical M2M solution looks like. However, there is no generally accepted M2M systems architecture or universal set of standards that is widely acknowledged. What is state of the art today is mainly coming from a few standardization bodies that have specified either protocols as systems components, or system and functional architectures for various parts of a complete end-to-end M2M architecture.

When it comes to IoT, there is today not a single widely accepted view of what a typical IoT solution looks like. However, as mentioned, there are a number of research activities on the European level that are converging on defining a reference architecture, and these activities are to be found as projects in the Internet of Things European Research Cluster (IERC 2013). The diversity in the possible and sought-after applications, as well as the diversity in deployment scenarios, together produces a large set of different requirements and constraints.

Attempting to produce a single architecture consequently results in a number of optional and conditional requirements, all depending on the particular problem at hand or application in focus. Nevertheless, the identified key features that are needed when building an M2M or IoT solution can now be put together into a larger context by proposing a single view of the main functional capabilities (see Figure 4.3). This is not a strict and formal functional architecture, but provides a conceptual overview. It also follows the approach of looking at the system capabilities from a layered point of view, including highlighting key functions that go across the layers.

Other approaches that are common in describing an architecture are the software approach and network approach that are more focused on how functions are distributed across a network topology. Here the different proposed functional layers and capabilities provided are discussed. For the sake of brevity, we use IoT as a collective term to include both M2M and IoT.

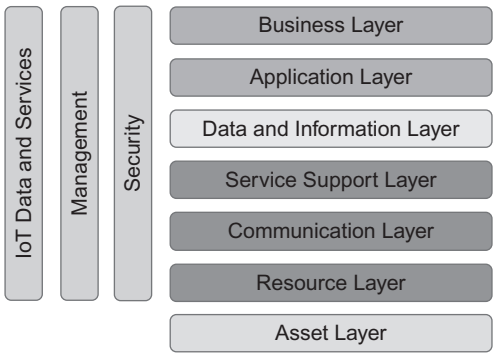

FIGURE 4.3

Functional layers and capabilities of an IoT solution.

At the lowest level is the **Asset Layer**. This layer is, strictly speaking, not providing any functionality within a target solution, but represents the *raison d'être* for any IoT application. The assets of interest are the real-world objects and entities that are subject to being monitored and controlled, as well as having digital representations and identities. The typical examples include vehicles and machinery, fixed infrastructures such as buildings and utility systems, homes, and people themselves – thus being inanimate as well as animate objects. Assets can also be of a more virtual character, being subjective representations of parts of the real world that are of interest to a person or an organization. A typical example of the latter is a set of particular routes used by trucks in a logistics use case. Information of interest may then be traffic intensity, roadwork, or road conditions based on the actual weather situation.

Assets are instrumented with embedded technologies that bridge the digital realm with the physical world, and that provide the capabilities to monitor and control the assets as well as providing identities to the assets. The Resource Layer provides the main functional capabilities of sensing, actuation, and embedded identities. Sensors and actuators in various devices that may be smartphones or Wireless Sensor Actuator Networks (WSANs), M2M devices like smart meters, or other sensor/actuator nodes, deliver these functions. This is also where gateways of different types are placed that can provide aggregation or other capabilities that are closely related to these basic resources. Identification of assets can be provided by different types of tags; for instance, Radio Frequency Identification

(RFID) as in (ISO/IEC RFID 2013), or optical codes like bar codes or Quick Response (QR) codes. The topic of devices and gateways is further dealt with in Section 5.1.

The purpose of the **Communication Layer** is to provide the means for connectivity between the resources on one end and the different computing infrastructures that host and execute service support logic and application logic on the other end. Different types of networks realize the connectivity, and it is customary to differentiate between the notion of a Local Area Network (LAN) and a Wide Area Network (WAN). WANs can be realized by different wired or wireless technologies, for instance, fiber or Digital Subscriber Line (DSL) for the former, and cellular mobile networks, satellite, or microwave links for the latter. WANs can also be provided by different actors, where some networks can be regarded as public (i.e. offered as commercial services for the general public) or as private (i.e. dedicated networks that provide services in a more closed business or entirely company internal environment). Particularly in the mobile network industry, there are different models for how the communications services are provided that include wholesale of access, and dedicated virtual network operators that focus on managed M2M connectivity offerings without owning licensed mobile spectrum or actual network resources. When it comes to LANs, there are many examples of different types, and there is also no stringent definition of what might be considered a LAN or a WAN. Prime examples of LANs include Wireless Personal Area Networks (WPANs; also known as Body Area Networks, BANs) for fitness or healthcare applications, Home or Building Area Networks (HANs and BANs, respectively) used in automation and control applications, and Neighborhood Area Networks (NANs), which are used in the Distribution Grid of a Smart Electricity Grid. Communication can also be used in more *ad hoc* scenarios. Vehicle-to-Vehicle (V2V) is one example that can target safety applications like collision avoidance or car platooning. As opposed to the situation for WANs, the interface technologies used within LANs are characterized by being very industry segment-specific, supported by a plethora of different standards, or even being proprietary or at best de facto standards. LANs use both wired and wireless technologies. General examples of wired LANs include Ethernet and Power Line Communication (PLC), whereas twisted pair (KNX 2013) and (BACnet 2013) over RS-232 are two detailed examples from the building automation industry. Prominent examples of wireless LAN networking technologies include the (IEEE 802.11 2013) and (IEEE 802.15.4 2013) families, as well as (Bluetooth 2013), which has a recent protocol addition called (Bluetooth Low Energy

2013) that targets typical IoT applications. IEEE 802.15.4 is the basis for protocol stacks that target different M2M and IoT applications, for instance, the ZigBee specifications (Zigbee 2013a), the proprietary protocol stack (Z-Wave 2013) for home automation, and ISA100.11a (ISA100 2013) for industry automation. Many of the existing legacy industry-specific LAN protocol stacks do not use IP as the networking protocol, but there is a growing number of examples where the legacy protocol stacks are migrated towards IP, for instance, ZigBee IP, BACnet over IP, and IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) Bluetooth (IETF 6LoWPAN BTLE 2013). To provide an end-to-end communication service that bridges a LAN and a WAN, gateways are used. From a communication layer perspective, gateways are primarily used to do interworking or protocol translation at different levels of the protocol stack. This can involve the physical and link layers, but it can also involve interworking on the communications or messaging level, for example, to do interworking between a legacy protocol like ZigBee to exchanging service operations using HTTP as the means for communication. Section 5.2 deals with the various LAN and WAN aspects and technologies in detail.

As described earlier, IoT applications benefit from simplification by relying on support services that perform common and routine tasks. These support services are provided by the **Service Support Layer** and are typically executing in data centers or server farms inside organizations or in a cloud environment. These support services can provide uniform handling of the underlying devices and networks, thus hiding complexities in the communications and resource layers. Examples include remote device management that can do remote software upgrades, remote diagnostics or recovery, and dynamically reconfigure application processing such as setting event filters. Communication-related functions include selection of communication channels if different networks can be used in parallel, for example, for reliability purposes, and publish–subscribe and message queue mechanisms. Location Based Service (LBS) capabilities and various Geographic Information System (GIS) services are also important for many IoT applications. Of more specific relevance for IoT are services that relate to sensor originating data and actuation services, and services that relate to different tags like RFID. A directory that holds information of available resources and associated service capabilities that can function as a rendezvous mechanism is one example. In such a directory, nodes in WSANs can publish themselves with service descriptions and how to be reached. Applications then perform look-ups to find which device can provide the sensor reading of interest. Another directory service example is

the Object Naming Service of EPCGlobal (GS1 EPCGlobal 2013) that can resolve an RFID code to a URL where information about the tagged object can be found. Other repositories can hold information about the persistent real-world entities that are of interest to identify, monitor, and control, such as the Entity Directory from the SENSEI project or the Electronic Product Code Information Services (EPCIS) of EPCGlobal. In general, data storage for anything from raw data to knowledge representations, and processing capabilities such as data and event capture, filtering, and stream processing are different core common services for many IoT applications. Examples of different support services are provided in Chapters 6, 7, and 8.

Where the Resource, Communication, and Service Support layers have concrete realizations in terms of devices and tags, networks and network nodes, and computer servers, the **Data and Information Layer** provides a more abstract set of functions as its main purposes are to capture knowledge and provide advanced control logic support. Key concepts here include data and information models and knowledge representation in general, and the focus is on the organization of information. We refer to a Knowledge Management Framework (KMF) as a collective term to include data, information, domain-specific knowledge, actionable services descriptions as, for example, represented by single actuators or more complex composite sensing and actuation services, service descriptors, rules, process or workflow descriptions, etc. The concept of KMF is further described in Section 5.7. The KMF needs to integrate anything from single pieces of data from individual sensors to highly domain-specific expert knowledge into a common knowledge fabric. Key concepts to construct the KMF include semantic annotation, Linked Data (Bizer et al., 2009), and building different ontologies. Knowledge is highly dynamic, and different techniques are used to capture knowledge as insights, as well as consume knowledge to learn, draw conclusions, propose or even make decisions based on past experiences, current knowledge, and predicted outcomes of certain actions.

The **Application Layer** in turn provides the specific IoT applications. There is an open-ended array of different applications, and typical examples include smart metering in the Smart Grid, vehicle tracking, building automation, or participatory sensing (PS). Part III of this book is devoted to providing examples of different IoT applications.

The final layer in our architecture outline is the **Business Layer**, which focuses on supporting the core business or operations of any enterprise, organization, or individual that is interested in IoT applications. This is where any integration of the IoT applications into business processes and

enterprise systems takes place. The enterprise systems can, for example, be Customer Relationship Management (CRM), Enterprise Resource Planning (ERP), or other Business Support Systems (BSS). The business layer also provides exposure to APIs for third parties to get access to data and information, and can also contain support for direct access to applications by human users; for instance, city portal services for citizens in a smart city context, or providing necessary data visualizations to the human workforce in a particular enterprise. The business layer relies on IoT applications as one set of enablers out of many (e.g. field force automation), and takes care of necessary orchestration and composition to support a business process workflow. A detailed discussion on business integration is provided in Section 5.4.

In addition to the functional layers, three functional groups cross the different layers, namely Management, Security, and IoT Data and Services. The former two are well known functions of a system solution, whereas the latter one is more specific to IoT.

Management, as the name implies, deals with management of various parts of the system solution related to its operation, maintenance, administration, and provisioning. This includes management of devices, communications networks, and the general Information Technology (IT) infrastructure as well as configuration and provisioning data, performance of services delivered, etc. M2M management aspects that are covered in Chapters 7 and 8.

Security is about protection of the system, its information and services, from external threats or any other harm. Security measures are usually required across all layers, for instance, providing communication security and information security. Trust and identity management, and authentication and authorization, are key capabilities. From an IoT perspective, management of privacy via, for example, anonymization, is in many instances a specific requirement.

The final functional group of our outlined architecture is denoted **Data and Services**. Data and Service processing can, from a topological perspective, be done in a very distributed fashion and at different levels of complexity. Basic event filtering and simpler aggregation, such as data averaging, can take place in individual sensor nodes in WSANs, contextual metadata such as location and temporal information can be added to sensor readings, and further aggregation can take place higher up in the network topology. More advanced processing is, for instance, data mining and data analytics that can be done in near real-time. This functional group thus represents the vertical flow of data into knowledge, the

abstraction of data and services in different levels, and the process steps of extracting knowledge.

As the knowledge layer is focused on the organization and representation of knowledge, this functional group is focused on the different processing steps in the data and services value chain, thus at different levels of granularity and abstraction. Different technologies are used to support the different levels of knowledge extraction, processing, reasoning, and decision-making. Well-known technologies here include stream processing, analytics, machine learning, reasoning, and inferencing. Section 5.6 provides a description of the technologies and tools for data and service processing.

What is not reflected in the architecture outline is the lifecycle aspect of an IoT system solution. Lifecycle aspects of interest include the planning phase for any deployment, and the design phase that involves both systems integration and application development, where APIs and SDKs are important. The actual deployment involving the steps of configuration and provisioning takes place before any solution is put into actual operation. These different steps are outside the scope of this book, but some aspects related to the deployment phase are covered in Chapter 8.

The approach to reach an applied architecture is provided in Chapter 7, whereas Part III of the book provides examples of applied architectures and information on system solutions for selected and typical M2M and IoT applications.

4.4 Standards considerations

The purpose here is not to provide an overview of relevant standards, but to provide an overview of the landscape in which various relevant standards are developed. It is not exhaustive, but will serve as an illustration that the standardization around M2M and IoT is rather complex and multi-dimensional (see Figure 4.4). The primary objective of any technology-oriented standardization activity is to provide a set of agreed-upon specifications that typically address issues like achieving interoperability in a market with many actors and suppliers.

The first consideration is that standards are developed across a number of different industries. There are a number of standardization organizations and bodies, both proper Standards Development Organizations (SDO) as well as special interest groups and alliances that develop standards specifications. Different national and international bodies ratify standards