Access Control in Internet-of-Things: A Survey

Sowmya Ravidas[a,*], Alexios Lekidis[a], Federica Paci[b], Nicola Zannone[a]

[a]Eindhoven University of Technology, The Netherlands
[b]University of Southampton, UK

Abstract

The Internet of Things (IoT) is an emerging technology that is revolutionizing the global economy and society. IoT enables a collaborative environment where different entities – devices, people and applications – exchange information for service provision. Despite the benefits that IoT technology brings to individuals, society and industry, its wide adoption opens new security and privacy challenges. Among them, a vital challenge is the protection of devices and resources produced within IoT ecosystems. This need has attracted growing attention from the research community and industry, and several authorization frameworks have been designed specifically for IoT. In this survey, we investigate the main trends in access control in IoT and perform an extensive analysis of existing authorization frameworks tailored to IoT systems. Driven by the needs of representative IoT applications and key requirements for IoT, we elicit the main requirements that authorization frameworks for IoT should satisfy along with criteria for their assessment. These criteria and requirements form a baseline for our literature study. Based on this study, we identify the main open issues in the field of access control for IoT and draw directions for future research.

Key words: IoT, access control, literature study

1. Introduction

A recent technological evolution in the area of pervasive computing is the Internet of Things (IoT). IoT is a *"system of entities (including cyber-physical devices, information resources, and people) that exchange information and interact with the physical world by sensing, processing information, and actuating"* [4]. IoT provides advanced applications to industry and citizens that improve individuals' quality of life and contribute to the world's digital economy. The adoption of IoT is steeply increasing and several IoT applications are emerging, ranging from smart home [35], patient monitoring [53, 76] and industry automation [102] to intelligent transportation [50], disaster management [117] and infrastructure monitoring [62].

IoT combines the current Internet infrastructure and emerging technologies, to ensure the seamless intercon-

---
*Corresponding author

Email addresses: `s.ravidas@tue.nl` (Sowmya Ravidas),
`a.lekidis@tue.nl` (Alexios Lekidis),
`f.m.paci@soton.ac.uk` (Federica Paci),
`n.zannone@tue.nl` (Nicola Zannone)

nection of hundreds of billions of embedded systems and manage the services they provide while reducing the Internet infrastructure's cost and making it more scalable, flexible and reactive [111]. The adoption of IoT initially relied on the use of web-services to facilitate software reusability and reduce application development complexity. Although the integration with web-services was an important addition to existing Wireless Sensor Network (WSN) technologies, it also came with new challenges. While web-services are based on long-lived transactions, IoT applications are usually deployed in constrained-resource devices that only wake-up for a short period of time. To overcome the limited capabilities of IoT devices, a recent trend is to shift data storage, communication and computation from resource-constrained devices to the cloud [73] and edge devices [107].

Although IoT brings several benefits for individuals, industry and society [113], the use of resource-constrained devices along with the adoption of a plethora of technologies enlarges the attack surface and introduces new security vulnerabilities. According to the OWASP IoT project [3, 74], insecure access to web, backend APIs, cloud and mobile interfaces is one of the top vulnerabilities for IoT applications. Indeed, smart devices are typically configured and controlled via vendor apps, which can have a smartphone-based interface and a Web-based interface through a service running on a cloud infrastructure. Services expose a Web-API that allows to query and control user data and devices from the same vendor and other compliant devices from other vendors. Services from vendors can be composed with third party services e.g Facebook, Instagram using IFTTT Web service.

In this complex IoT ecosystem access control should be enforced at each of these interfaces.

However, commercial IoT frameworks fall short in implementing access control to this interfaces. Most of the IoT frameworks enforce coarse-grained access control policies [54, 96]: for instance, Nest Thermostat[1] grants access to all the capabilities of a smart device or to none, or the Apple Home Kit[2] distinguishes between full control of the device, view only control and local or remote control. Other IoT frameworks enforce slightly richer access control policies based on environmental conditions: for example, Samsung SmartThings[3] grants access whether the user is at home or away. But to track these factors SmartThings gains access to the GPS coordinates of the user smartphone, which allows real-time tracking of users and therefore violates their privacy.

These flaws in implementing access control policies leads to devices and apps to be easily exploited to gain unauthorized access to devices and to users and devices' data that they collect and store [24, 30, 40]. A real-world example of the consequences of having permissive or overprivileged interfaces is Internet-enabled baby monitors being remotely hacked and controlled [106]. The remote hackers could intercept live video feeds from baby monitor's camera and perform different actions including talking to babies and changing camera settings and even permissions to remotely control the baby monitor.

To address these limitations, a fine-grained authoriza-

---

[1]https://nest.com/thermostats/
nest-learning-thermostat
[2]https://www.apple.com/lae/ios/home
[3]https://www.smartthings.com

2

tion system that restricts access to IoT device's interfaces and data only to authorized users should be implemented. Such a system should enforce policies based on several factors like the capabilities of the smart devices, the relationships among the users using the devices, and environmental conditions such as time, and location [54].

Driven by this and similar considerations from other studies, recent years have seen an increasing interest in the field of access control for IoT in both academia and industry, which resulted in the emergence of several authorization frameworks for IoT. These frameworks are often based on different IoT technologies and rely on different underlying assumptions. This variety of solutions makes it difficult to evaluate their effectiveness, especially with respect to the target IoT applications.

Motivation. While there are a number of surveys that discuss security challenges in IoT [71, 91, 112, 114] such as privacy and network security, only a few address access control [82, 90, 103, 122]. Table 1 provides an overview of existing surveys on access control for IoT. As shown in the table, existing surveys have the following limitations:

- They only discuss some aspects related to access control. Most of the surveys focus on policy specification whereas policy management and evaluation are only partially considered or not investigated at all.
- They do not identify the requirements that access control systems for IoT should satisfy along with evaluation criteria to systematically analyze existing authorization solutions for IoT.
- They do not discuss the demands and requirements of the IoT application for which authorization

Aspects	Our Survey	Sicari et al. [103]	Roman et al. [90]	Zhang et al. [122]	Ouaddah et al. [82]
Policy Specification	✔	✳	✳	✔	✔
Policy Management	✔	✳	✗	✳	✳
Policy Evaluation & Enforcement	✔	✗	✳	✗	✳
IoT applications	✔	✗	✗	✗	✳

Legend

✔: full coverage   ✳: partial coverage   ✗: no coverage

Table 1: Comparison with existing surveys

frameworks are designed.

- They do not discuss the suitability of existing solutions to representative IoT applications.
- They do not analyze most recent papers proposing authorization solutions for IoT.

Contribution. In this survey we present a systematic analysis of existing authorization solutions for IoT that addresses the above issues in existing survey papers. Our goal is to identify open challenges in existing authorization solutions to drive the research and development of more effective access control solutions for IoT.

The main contributions of our survey are the following:

- A framework to enable a systematic and comparative analysis of authorization solutions for IoT. The framework consists of a set of requirements that authorization solutions for IoT should meet and a number of criteria for their assessment.
- A review of several recent authorization frameworks for IoT and their evaluation with respect to the requirements and criteria in the framework.
- Guidelines to design an authorization framework

tailored to specific needs and constraints of the most common IoT applications.

- Open challenges that need to be addressed when designing access control solutions for IoT.

Methodology. To perform our survey, we first provide an overview of relevant characteristics of IoT systems and enabling technologies based on a study of the literature and on current developments of IoT. Our analysis revealed that cloud computing and edge computing are often adopted as a baseline technology in IoT to facilitate the management of devices and resources in IoT ecosystems. To this end, we investigate how these computing paradigms have been adapted to IoT. Driven by real-world scenarios, we identify a set of non-functional requirements for IoT systems.

From these requirements, we elicit the requirements that authorization solutions for IoT should meet and a number of criteria for their assessment. Our requirements cover the main activities of the access control process, ranging from policy specification and management to policy evaluation and enforcement. To assess to what extent existing authorization frameworks meet the elicited requirements, we study the characteristics of the IoT environment in which authorization frameworks have been deployed (*IoT architecture style*, *communication protocols*, *data format*) to analyze the assumptions underlying the IoT environment and, in particular, the capabilities of nodes and their interconnections. Moreover, we study the properties of the proposed authorization frameworks (*access control model*, *policy evaluation strategy*, *deployment configuration*). We review several recent authorization frameworks for

IoT and evaluate them with respect to the identified requirements and criteria. We also discuss the suitability of existing solutions to representative IoT applications.

Our literature review reveals several important insights and help define further research directions in access control for IoT. In particular, we have observed an increasing interest in the development of authorization frameworks tailored to IoT systems. However, most of the proposed frameworks aim to provide a general solution to address the problem of authorization in IoT. Our analysis, on the other hand, shows that different IoT applications are characterized by different demands and, thus, there is not one solution that fit all IoT applications. Accordingly, the design of an authorization framework tailored to IoT should account for the specific needs and constraints of the target IoT application.

Organization. The remainder of the paper is structured as follows. We present an introduction to IoT in Section 2, followed by an overview of IoT enabling technology in Section 3. We discuss typical IoT applications and define the primary functional and non-function requirements for IoT systems in Section 4. We delineate the main requirements that authorization frameworks for IoT should satisfy in Section 5 and review existing frameworks in Section 6. We identify open issues and draw research directions for future research in Section 7 and conclude the paper in Section 8.

2. Internet of Things

In this section, we provide an overview of the main IoT elements and present how these elements are connected
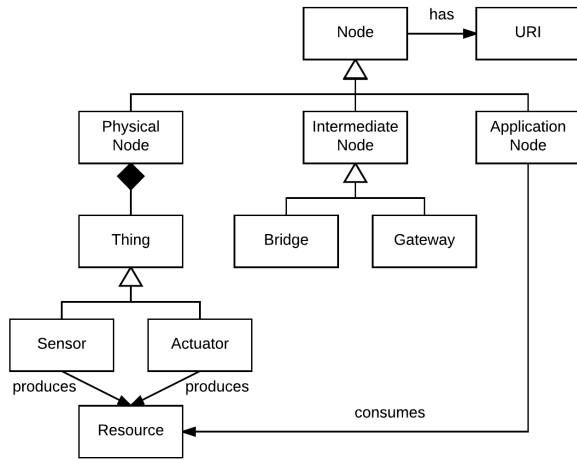
Figure 1: IoT Metamodel

to each other.

2.1. IoT Elements

Figure 1 presents the IoT metamodel (in form of UML class diagram) representing the main elements within an IoT system and their relationships. An IoT system consists of devices or smart objects that can interrelate and interconnect among themselves and with the environment to provide services to end-users. Hereafter, we refer to the components of an IoT system as *nodes*. To be able to connect with other nodes, a node should be equipped with a communication interface. In particular, every node is characterized by a Uniform Resource Identifier (URI) that uniquely identifies the node over the network. We distinguish three types of nodes: physical nodes, intermediate nodes and application nodes.

A *physical node* consists of things. A *thing* can be a sensor, an actuator or any other entity that can interact with the environment. A *sensor* is a device that detects events or changes in the environment. An *actuator* is responsible

for controlling a mechanism or a system. The output of sensors and actuators is usually referred to as *resource*. An *application node* consumes resources produced by physical nodes to provide services to end-users.

Physical nodes and application nodes can belong to different networks. *Intermediate nodes* are used to route traffic and connect two or more local area networks. An intermediate node can be a bridge or a gateway. A *bridge* connects local area networks that use the same protocol. In particular, a bridge forwards messages from one network to another based on the MAC address of the destination node. A *gateway* connects networks that use different protocols. Unlike bridges that are only able to forward messages, gateways are also able to perform message conversion to achieve connectivity across different networks.

2.2. IoT Architecture

An IoT architecture provides a high-level view of the functionalities and connectivity within an IoT ecosystem. In this section, we present an architecture pattern for IoT and discuss the architecture styles typically used in IoT.

2.2.1. Architecture Pattern

Several architecture patterns have been proposed for IoT [17, 18, 22, 33, 49, 63, 116]. The layers in a IoT architecture pattern provide a specific view on the IoT system, and their choice depends on the scope of the study. We observed that authorization solutions tailored to IoT environments are often deployed in the middleware due to the limited capabilities of IoT devices. To this end, differently from many existing architecture patterns for IoT, we separate network communication from the
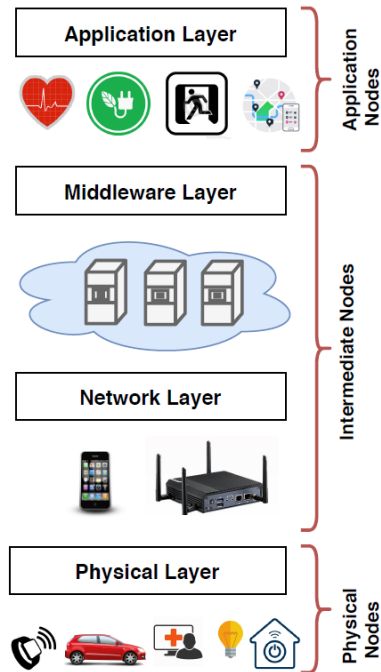
5

Figure 2: IoT Architecture

middleware. This distinction allows us to reason on the sharing of resources among nodes and end-users by abstracting from the actual protocols used for their transmission. In particular, we adopt a four-layered IoT architecture pattern (Figure 2), which consists of physical, network, middleware and application layers. Next, we describe the main functionalities of each layer.

Application Layer: The application layer aims to provide services to end-users. This layer comprises application nodes that handle the application logic as well as data semantics and presentation [22]. These nodes receive data from the middleware and process them depending on the end-user requirements and type of service provided. Moreover, the application layer encompasses APIs to facilitate the communication with the middleware and user inter-

faces through which end-users can access the services.

Middleware Layer: The purpose of the middleware is to ensure connectivity and interoperability within the IoT ecosystem. It consists of intermediate nodes[4] that process data received from lower layers and pass them on to the application layer. We will provide an overview of the main types of middleware used in IoT in Section 3.2.

Network Layer: The goal of the network layer is to support networking and data transfer between nodes [33]. The network layer implements the communication protocols required for data exchange within an IoT ecosystem. We will discuss the communication protocols typically used in IoT in Section 3.1.

Physical Layer: The purpose of the physical layer is to characterize the sensing and control capabilities of an IoT system. This layer comprises physical nodes such as sensors and actuators [17] that sense the environment and interact with it in response to changes or users' requests. These nodes produce resources (e.g., sensing data) that are passed to application nodes through the network and middleware layers.

2.2.2. Architecture Styles

Several architecture styles for IoT have been proposed in the last years. Although these architecture styles can vary based on the application domain, they can be classified into three main types based on the connectivity between physical nodes, middleware (intermediate nodes)

---

[4]Note that we use the term intermediate nodes to indicate any node that is in between the physical layer (comprised by physical nodes) and application layer (comprised by application nodes).

6

(a) Centralized IoT Architecture   (b) Connected IoT Architecture   (c) Distributed IoT Architecture
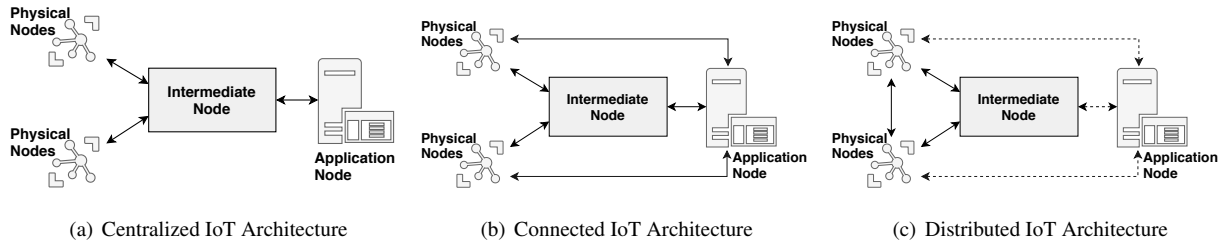
Figure 3: IoT Architecture Styles

and application nodes [29, 90]. An overview of the main IoT architecture styles is presented in Figure 3.

Centralized: Physical and application nodes are required to communicate with each other through intermediate nodes. This means that if an application node wants to retrieve resources from a physical node, a connection has to be established using the interfaces provided by the intermediate node. This architecture style is typically required for physical nodes with limited processing and storage capabilities.

Connected: Physical nodes have the ability to process information and forward it to intermediate nodes. In addition, physical nodes can provide resources directly to application nodes. This means that application nodes can directly connect to physical nodes through the interfaces they provide.

Distributed: Every node can communicate with each other. This means that every node has the potentiality to process information and provide services. Note that, differently from the other IoT architecture styles, the distributed IoT architecture does not require an intermediate node, although its use can facilitate the communication between nodes.

3. IoT Enabling Technology

Recent years have seen the emergence of new technology (and the adaptation of existing technology) to meet the demands of IoT applications and low-power and resource-constraint IoT devices. This section presents an overview of the main technologies enabling IoT, with a particular focus on communication protocols and middleware.

3.1. Network Layer

Communication protocols and their relations are usually represented using a network stack. A network stack is represented in layers for easier design and evaluation.[5] Each layer represents different functions and offers different methods for data handling. In this work, we use a four layer model that resembles the traditional five layer network model [104]. It comprises the application layer, transport layer, network layer and data link & physical layer.

Figure 4 presents the network stack describing the protocols commonly used in IoT. Moreover, we relate the standards upon which the physical & datalink layer

---

[5]Note that the layers in the network stack are different from the architecture layers presented in Section 2.2, although some layers may have the same name.
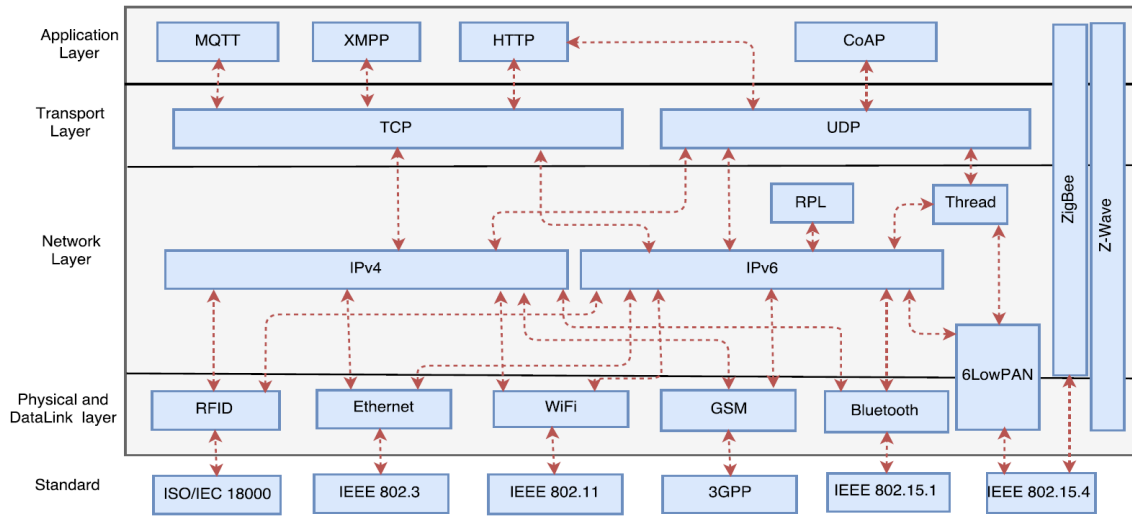
7

Figure 4: IoT Network Stack and Standards

protocols are defined. In the figure, layers are separated by solid lines. Arrows indicate that a given protocol is built on top of another protocol or built on a given standard. Next, we review the main protocols used in each layer.

Physical layer & datalink layer: Several protocols have been used in the physical layer & datalink layer within IoT ecosystems. We classify them based on the network type they support: Local Area Network (LAN), Personal Area Network (PAN) and Wide Area Network (WAN). The PAN protocols commonly used in IoT are Radio-Frequency Identification (RFID) [10], Bluetooth [26], ZigBee [13] and ZWave. RFID [10] is largely used within IoT environments to identify devices [99]. Bluetooth and its variant for low energy devices are a short-range wireless technology based on the IEEE 802.15.1 standard [26].

The IEEE 802.15.4 standard [15] is intended for low-rate wireless personal area networks (LRWPAN). It offers low-cost low-power communication to devices in close proximity. A protocol based on LRWPAN specification is Zigbee [13]. Although this protocol builds on LRWPAN, it has additional components for the network and application layers. Similar to Zigbee, the Z-Wave[6] protocol also works on low-frequency radio bandwidth. Z-Wave is a proprietary protocol that is not built on any specific standard. It provides the complete network stack from the physical layer to the application layer.

Among LAN protocols, traditional technologies such as Ethernet and Wi-Fi are often used in IoT. Ethernet [14] is a highly reliable protocol based on IEEE 802.3. WiFi is based on the IEEE 802.11 standard [11] and allows gateway devices to transmit information using radio-waves over high speed Internet connection.

For Wide Area Network (WAN), cellular technologies are often used in IoT. Global System for Mobile

---

[6]http://www.z-wave.com

8

Communications (GSM) [16] is the most commonly adopted protocol for long range communication, primarily used for data voice transmission and services based on the 3GPP specification.

Network layer: The most commonly used network layer protocols in IoT are IPv4 and IPv6. These protocols are variations of the Internet Protocol (IP) [6], both used to identify devices on the Internet based on unique addresses. They provide an addressing scheme that is used to identify a group of IoT devices geographically [49]. IPv6 requires a minimum MTU (Maximum Transmission Unit) size of 1280 bytes, whereas the IEEE 802.15.4 link layer allows a maximum frame size of 127 bytes. Hence, there is a need for additional protocols to perform packet compression in order to transmit IPv6 packets over IEEE 802.15.4.

The IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs) [77] is built on top of the LRWPAN specification. 6LoWPAN uses encapsulation and header compression mechanisms to transmit IPv6 packets over IEEE 802.15.4 networks, thereby creating a mapping between the link and network layer. This protocol aims to support IP for low power IoT devices. Thread[7] is another network layer protocol specifically designed for device-to-device communication in building automation, which is based on IPv6 and 6LoWPAN.

IoT systems also extend the architectures and protocols used in Wireless Sensor Networks (WSN) through the addition of web resources. A particular type of WSN IoT network architecture is low-power and lossy network (LLN). In such networks, devices and routers have memory and processing constraints. Moreover, routers typically support low data rates and are unstable. RPL [115] is an IPv6 routing protocol for LLN that efficiently routes multipoint-to-point (from devices to a central point), point-to-multipoint (from a central point to devices) and point-to-point (between the devices) traffic.

Transport layer: Transmission Control Protocol (TCP) [7] and User Datagram Protocol (UDP) [86] are widely used protocols in the transport layer for IoT. TCP is connection-oriented whereas UDP is connection less. This distinction makes TCP more reliable than UDP as TCP guarantees that all packets are delivered. However, it is not scalable for small data transmissions in IoT devices [120]. UDP is more suitable for real-time communication where delay is not tolerated.

Application layer: In the traditional Internet protocol stack, the most common protocol in the application layer is HTTP. It can be run over both TCP and UDP. However, HTTP is verbose and complex, and adds a significant parsing overhead. This may not be suitable for constrained devices. Moreover, HTTP inherits the limitations of the protocol at the transport layer on which it runs [120]. To overcome these limitations several application layer protocols have been developed. Their goal is to enhance communication performances by minimizing the overhead.

Constrained Application Protocol (CoAP) [100] is one of the most commonly used protocols for IoT devices and is based on the client-server model. It runs over UDP and performs asynchronous message exchanges. CoAP has low header overhead and hence simplifies message parsing [101]. HTTP and CoAP can also be

---

used in association with representational state transfer (REST) architecture [8] that makes it possible to access the resources of an IoT device through a URI. Another commonly used application layer protocol is Message Queue Telemetry Transport (MQTT) [12], a lightweight messaging protocol on top of the TCP/IP protocol. It is mainly used for communication with remote locations where network bandwidth can be limited. MQTT is based on the publish-subscribe paradigm, where the sender (i.e., the publisher) transfers the message to a broker that distributes the messages to the interested clients (i.e., subscribers). One of the commonly used MQTT brokers is Mosquitto [2]. Since MQTT runs on top of TCP, it may not be suitable for applications that require real-time processing. Another application layer protocol commonly used in IoT is Extensible Messaging and Presence Protocol (XMPP) [92]. This protocol is used for streaming XML elements and real-time exchange of structured data.

3.2. Middleware Layer

The middleware is often required to ensure connectivity, interoperability, storage and computation of data within an IoT ecosystem. Different types of middleware have been proposed for IoT [88, 99, 105]. In our study of the literature on access control for IoT, we observed that most of the existing solutions rely on cloud computing and edge computing as middleware.

According to the NIST [73], *cloud computing* is a computing paradigm that enables on-demand network, storage, applications and other services without any management effort. Cloud computing is become a core component of most of IoT platforms because it provides elastic and scalable data storage and processing. The adoption of cloud computing has opened new directions for technological enhancements in several IoT applications. The cloud has different role within the IoT architecture, depending on the application needs and requirements. We refer to Section 4.3 for a discussion.

Fog computing [107] and *edge computing* [45] have been recently proposed as new computing paradigms to reduce the communication latency and bandwidth required by the use of a remote cloud platform for data storage and processing. The underlying idea behind fog computing is that data generated by IoT devices are processed at the edge of the network, close to where they are generated. A network of micro data centers process and store critical data locally and then push all received data from IoT devices to a remote cloud platform for long-term storage. The computational processes being done by the micro data centers is usually referred as edge computing, while the network connections between the micro data centers and the cloud platform is referred to as fog computing.

4. IoT Applications and Requirements

Consumers, industries and governments are starting to realize the numerous benefits that IoT brings to the society, economy and environment. However, in order for these stakeholders to reap off the benefits of IoT, an IoT ecosystem has to address key requirements related to scalability, interoperability, performance, availability, reliability and dynamicity. As shown below, these requirements are not equally relevant for all IoT applications; instead, each application has different characteristics and,

10

thus, different sets of requirements. In this section, we introduce representative IoT applications and the key requirements that they have to satisfy.

4.1. Applications

IoT technology has been applied in a variety of applications domains. In this section, we examine the most representative applications of IoT [1, 47].

Smart Homes (⌂) Smart home applications involve the use of smart objects like thermostats, door bells, door locks, smart appliances (e.g., smart tv, smart fridge) and smart light bulbs that are remotely controlled by the home owners via their smart phones or home assistants. A smart home involves both human-to-machine and machine-to-machine interactions. An example of the former is the one in which a home owner has installed smart door locks and an alarm system at the door and windows and she would like to allow other users, e.g. visitors or family members, to unlock the door using their smart phone. An example of machine-to-machine interaction is a thermostat linked to a garage door that automatically increases the temperature in the house when the garage door opens or a smart fridge that automatically notifies the owner's smart phone that the milk is running low.

Health IoT (♥) Health IoT comprises medical devices such as insulin pumps, cochlear implants and pacemakers that include sensors to monitor patients' vital signs and actuators to respond to situations that can potentially threaten patients' life. One typical use case is diabetes monitoring where a patient is equipped with a wireless-enabled glucose monitor and an injection device that allows monitoring her blood glucose level, and receives alerts on her mobile phone for hypoglycemia and insulin dosage updates from her primary physician.

Smart Buildings (▤) Smart building applications focus on the use of smart components such as heating, ventilation, air conditioning (HVAC), smart lighting systems, and safety & security systems such as fire alarms to enhance the overall living quality of tenants and to save energy. These devices are usually managed centrally by a facility manager using a Building and Lighting Management System (BLMS). However, different areas of these buildings can be leased to different companies. These companies should retain control of the lighting and HVAC in their part of the buildings. Other parts of the buildings automation system such as entrance illumination and fire-alarm systems should be controlled either by all companies together or by a facility management company.

Connected Vehicles (🚘) Connected vehicle applications involve vehicles, roadside units and other infrastructure to communicate and share traffic and road information. In this setting, vehicles transmit their location, direction, speed, and other information such as vehicle ID and size to other vehicles in proximity. Vehicle-to-vehicle communication (V2V) enables safety use cases such as forward collision warning, blind-spot detection and traffic congestion. Similarly, vehicles' information can be shared with other components of the road infrastructure such as traffic lights, stop signs, tool booths and road crossing in (vehicle-to-infrastructure (V2I)) to support traffic jam notification, prediction of potential traffic jams and dynamic traffic light control.

Smart Manufacturing (🏭) Smart manufacturing appli-

	🏠	❤	🏢	🚗	🏭
Scalability	low	medium	medium	high	high
Heterogeneity	high	high	high	high	high
Lightweight	high	high	high	medium	high
Latency sensitive	low	high	medium	high	high
Reliability	low	high	medium	high	high
Dynamicity	medium	high	high	high	high
User involvement	high	high	low	low	low
Automation	medium	high	high	high	high
IoT architecture	connected	connected	centralized	distributed	connected
Administrative domain	single	single/ multi	multi	multi	multi

Table 2: Main characteristics of IoT applications

cations are typically based on an open and interconnected infrastructure that allows the management and monitoring of industrial and manufacturing processes. The infrastructure integrates different technologies such as microprocessors, cloud services, new generation control systems, software application, sensors and actuators to collect real-time data and process them to take prompt decisions based on reliable data. A popular use case is predictive maintenance where sensors, cameras and data analytics are used to determine when a piece of equipment is going to fail before it actually does. By leveraging streaming data from sensors and devices, the infrastructure can quickly assess the current condition of equipment, recognize warning signs, and deliver alerts to operators, who can trigger appropriate maintenance commands and processes.

4.2. Requirements

Realizing an IoT system is not an easy task due to the many requirements that need to be addressed. In this section, we discuss key functional and non-functional requirements that should be satisfied by any IoT system and for which IoT applications these requirements are more relevant. These requirements have been gathered from a study of the literature and an analysis of the main characteristics of the IoT applications and use cases presented in the previous section. A summary of these characteristics is presented in Table 2 and the requirements along with the IoT applications in which these requirements are particularly relevant are presented in Table 3.

Scalability: Scalability refers to the ability of being extensible in terms of number of users and physical nodes without negatively affecting the quality of the services provided by the IoT system [20]. Implementing this requirement involves an efficient way to manage physical nodes within the IoT system (**GR1**). Node management includes aspects such as node registration and identification, and the storage and processing of huge volume of data generated by physical nodes (**GR2**). This is a key requirement in application scenarios like connected vehicles where millions of cars can join the road infrastructure or in the smart manufacturing sector where new equipment to be monitored is often added to the interconnected infrastructure.

Interoperability: Interoperability is a significant requirement for all IoT applications. IoT systems usually consist of heterogeneous devices, services and applications from

12

Category	ID	Requirement	IoT application
Scalability	GR1	An IoT system must be scalable to manage a large number of devices.	🚗 📈
	GR2	An IoT system must be scalable to handle the resources produced by devices.	🚗 📈
Interoperability	GR3	All components of an IoT system must be able to communicate with each other.	🏠 ♥ ▦ 🚗 📈
Performance	GR4	The communication overhead should be low on the device side.	🏠 ♥ ▦ 🚗 📈
	GR5	The computation overhead should be low on the device side.	🏠 ♥ ▦ 🚗 📈
	GR6	The latency of information exchange between the nodes must be minimal.	♥ 🚗 📈
Reliability & Availability	GR7	An IoT system must guarantee an adequate level of reliability and availability of its nodes.	♥ 🚗 📈
Dynamicity	GR8	An IoT system must be able to handle the dynamicity of the nodes.	🏠 ♥ ▦ 🚗 📈
	GR9	An IoT system must be able to handle the dynamicity of the environment.	🏠 ♥ ▦ 🚗 📈
Usability	GR10	An IoT system must reduce user effort in system administration and configuration.	🏠 ♥

Table 3: IoT requirements along with the relevant IoT applications

different vendors and service providers that use different communications technologies and formats for data exchange [75]. Interoperability should be considered by both service providers and device manufactures to make sure that nodes can exchange information and resources with each other regardless the specific technology or protocols being used (**GR3**).

Performance: IoT systems often consist of resource constrained devices that have limited storing, networking and processing capabilities. Therefore, IoT solutions and protocols should be lightweight [79, 98] meaning that overhead due to communication (**GR4**) and computation (**GR5**) should be as low as possible on the device side. Performance can also be affected by the latency of transferring data between nodes due to the underlying network and middleware infrastructure. Delays in the transmission and processing of data should be minimal (**GR6**). This is a key requirement for safety-critical IoT applications like connected vehicles and patient monitoring, and for real-time applications like smart manufacturing. For example, in forward collision warning a delay in the transmission of the speed and location of a vehicle to the vehicle traveling on front could cause a collision between the two vehicles.

Reliability & Availability: Reliability refers to the proper functioning of an IoT ecosystem [20]. The data sensing, transmission and processing should be reliable in the sense that even if a failure or a malfunction occurs, the IoT ecosystem should still guarantee service delivery [88]. Reliability entails the availability of data sensing, communication and processing, and of the services and applications that consume the data. If critical data from sensors are not available, the IoT system may actuate the wrong decision. Therefore, an IoT system needs to guarantee reliability and availability of data, applications and services over time (**GR7**). The need of reliability and availability depends on the type of service delivered by an IoT application. Occasional unavailability and/or failure can be tolerated in smart homes, e.g. the smart fridge failing to notify the owner that she is running out of milk or a home assistant failing to remotely control the smart lighting system. However, reliability and availability are major requirements for safety related IoT applications, like connected vehicles and health IoT, and time critical applications, like smart manufacturing. For instance,

failures in a smart glucose monitoring device or in an injection device could be life threatening for the patient. Similarly, if a warning about a piece of equipment malfunctioning is not delivered to the operator, maintenance processes are not conducted with consequent disruption and delays of the manufacturing process.

Dynamicity: An IoT ecosystem is dynamic by design wherein the network topology and connectivity can constantly change [109]. For instance, physical nodes can leave the system or new physical nodes can join the system [34]. IoT systems should be able to adapt to the dynamicity of nodes (**GR8**). This is a critical requirement for IoT applications such as connected vehicles and smart manufacturing. Moreover, IoT systems are often employed in cyber-physical systems to monitor and manage IT infrastructures and the surrounding environment. In this setting, it is crucial that an IoT system is able to adapt to changes in the environment (**GR9**). The ability to adapt to changes is a key requirement for all IoT application scenarios.

Usability: Usability is a primary requirement in IoT applications that are characterized by a high user involvement and by the use of wearable devices like in smart homes and health IoT (**GR10**). Wearable devices often have very small displays, which makes user interaction and determining what information to display a tricky, but important factor. Device interfaces should also be easily customizable by users and facilitate the management and administration of the device itself.

The requirements above strictly influence the design and deployment of security mechanisms employed for the protection of the IoT system itself and resources produced and processed by the system. In Section 5, we discuss how the requirements in Table 3 affect the design of authorization systems tailored to IoT.

4.3. Discussion

IoT ecosystems are required to satisfy the requirements in Table 3, especially the ones of the target IoT application. The use of cloud provides a natural basis for the achievement of some of these requirements. However, the achievement of other requirements might require additional measures, depending on the type of IoT architecture adopted.

As the number of connected devices is increasingly growing, scalability (GR1 and GR2) is of utmost importance for IoT architectures. Cloud computing allows meeting these requirements since it offers a structural way to manage and remotely control the overall IoT ecosystem. The interaction of physical and application nodes with the cloud is enabled through the definition of interfaces. Physical and application nodes can use these interfaces to store or retrieve resources [43], thus providing interoperability (GR3). On the other hand, IoT applications like connected vehicles in which physical and application nodes can interact with each other without the presence of the cloud (distributed IoT architecture) are required to provide interfaces for the interaction with other nodes.

Cloud computing can also help in meeting performance requirements by relieving physical nodes from heavy computations (GR5). However, it might bring communication overhead to physical nodes (GR4) and introduce a delay in the communication between nodes [90] (GR6). This delay can be alleviated by the use of edge computing that brings cloud capabilities closer to physical nodes.

14

Cloud-based systems have usually a high uptime, thus ensuring the availability of the infrastructure (GR7) [84]. Hence, cloud-based IoT systems are usually reliable, although they are not robust against failures in the connectivity. One advantage of the connected and distributed IoT architectures is that, even if the connectivity to the cloud fails, application nodes can still access resources directly from physical nodes [90]. Cloud and edge computing also support the dynamicity of IoT environments [27] (GR8). In particular, the appearing and disappearing of physical nodes are typically handled by the cloud and edge nodes. On the other hand, in a distributed architecture, nodes are self-organizing and, thus, dynamicity is often handled using routing protocols such as RPL. Apart from the dynamicity of nodes, the IoT environment can also change rapidly (GR9). However, the cloud may not have any information about potential environment changes and has to rely on sensors and actuators to gather such information.

5. Access Control in IoT

While offering attractive opportunities and new business models, IoT opens several security and privacy issues. In this work, we focus on one of the main security issues in IoT, namely how to protect IoT devices and resources (data, applications, services) from being accessed by unauthorized users. A typical solution to address this issue is the adoption of an access control system that guarantees that only authorized entities (users and devices) gain access to IoT devices and resources. In this section, we investigate the requirements that access control systems for IoT should meet and identify design principles and

criteria to evaluate the current state of the art on this field.

The design of an access control system typically comprises three main components [94]: *policy*, which defines authorization requirements according to which access control is regulated; *model*, which provides a formal representation of access control policies and their evaluation; *mechanism*, which defines the low level implementation of the control imposed by the policy as formalized in the model. In this work, we identify requirements for the access control model and mechanism and discuss to which extent the access control models, policy evaluation strategies and enforcement architectures supported by existing access control systems for IoT satisfy the requirements. To this end, we first present an overview of the most popular access control models, reference architectures and policy evaluation strategies. Then, we introduce the requirements for an access control system and the criteria to evaluate whether the requirements are satisfied by authorization frameworks proposed for IoT. These requirements and criteria will serve as the baseline for the evaluation of existing authorization frameworks for IoT in Section 6.

5.1. Access Control Basics Concepts

In this section, we first review existing access control models and which concepts they support to specify an access control policy. Then, we introduce the main architectures and policy evaluation strategies that have been proposed to implement an access control system.

5.1.1. Access Control Models

Access control policies are formally represented according to an access control model. Several models have been

proposed in the literature. These models have different characteristics, which can influence the suitability of an authorization mechanism for IoT. Next, we present an overview of the most popular access control models.

Discretionary Access Control (DAC) [48]: DAC is based on the notions of ownership where a user has complete control over its own resources and devices, and can determine the permissions other users have on those resources and devices. Although many variations have been proposed, DAC is generally considered an identity-based access control model where access rights are assigned to users based on their identity. Various approaches to implement DAC have been proposed: *access matrix*, *authorization table*, *access control list* (ACL) and *capability list*. We refer to [94] for an overview of these approaches and, later in our analysis (Section 6), we differentiate between these implementations.

Mandatory Access Control (MAC) [67]: Differently from DAC, MAC relies on a set of system rules rather than being at the discretion of an object's owner. These rules are typically defined based on security labels associated to subjects and objects. Thus, similarly to DAC, MAC is considered an identity-based access control model.

Role-based Access Control (RBAC) [95]: RBAC relies on the notion of role to simplify the specification and management of access rights within an organization. A role comprises the set of permissions needed to carry out a certain job function. Users are assigned to roles and inherit the permissions assigned to the roles they have. Roles are often organized in a role hierarchy, which defines the inheritance of permissions between roles.

Organization-Based Access Control (OrBAC) [61]: OrBAC is based on three main concepts for the specification of access control policies, namely organization, concrete and abstract levels, and context. Organization is a structured group of active entities. Similarly to other access control models, concrete authorizations are specified in terms of subject, action, and object, defining which action a user can (or cannot) perform on an object. Concrete authorizations are derived from abstract permissions, which are defined in terms of roles, activities and views. As in RBAC, a role represents a job function within the organization. Activities group actions into an abstract set and views represent sets of abstract objects. Subjects in the concrete level are mapped to roles in the abstract level, actions are mapped to activities, and objects are mapped to views. The context represents a specific situation and is used in OrBAC to express dynamic rules.

Attribute-based Access Control (ABAC) [56, 119]: ABAC is a general-purpose access control model in which access rights are constrained with respect to the attributes of subjects, objects, actions and the environment. Policies and access requests are defined in terms of attribute names/values pairs. The applicability of a policy to a request is determined by matching the attributes in the request with the attributes in the policy. ABAC models often provides constructs to combine policies authored by different stakeholders and mechanisms to solve conflicts that can arise from these policies.

Usage Control (UCON) [83]: Similarly to ABAC, UCON allows the specification of policies in terms of subjects and objects' attributes. It also uses conditions to

16

express access constraints on the environment, thus providing the same expressiveness of ABAC. Moreover, UCON supports two additional decision properties, namely mutability of attributes and continuity of decision. Mutability of attributes accounts for changes of subjects and objects' attributes as a consequence of the usage. Continuity of decision denotes that permissions are checked not only at access time but also during the entire usage.

5.1.2. Reference Architectures

An authorization mechanism defines the low-level implementation of the access control model within the system. An authorization mechanism can be logically decomposed into key components that are responsible for the evaluation and enforcement of access control policies specified according to an access control model. Here, we discuss various reference architectures that have been proposed as a foundation for access control systems, namely *policy-based architecture*, *token-based architecture* and *hybrid architecture*, to study how the access control process is spread across the IoT ecosystem.

Policy-based Architecture: A widely adopted policy-based architecture is the one proposed by XACML [9], the de facto standard for the specification and enforcement of access control policies. The architecture comprises four main components:[8]

- *Policy Enforcement Point (PEP)* provides an interface with the system and is responsible for enforcing access decisions.

---

[8]The XACML reference architecture includes an additional component, called *Context Handler*. We omit this component here as its main function is to support the authorization process.
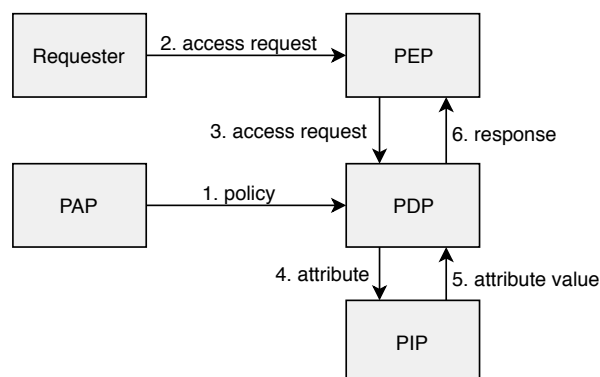
Figure 5: Policy-based Architecture

- *Policy Decision Point (PDP)* evaluates access requests against access control policies and determines whether access should be granted or denied.
- *Policy Administration Point (PAP)* acts as a policy repository and offers facilities for policy management.
- *Policy Information Point (PIP)* denotes the source of information (e.g., context information) needed for policy evaluation.

Figure 5 shows the interaction between these components. The PAP makes the policies available to the PDP (1). Upon receiving an access request (2), the PEP forwards the request to the PDP (3), which evaluates the request against the policies fetched from the PAP. If additional information is required for policy evaluation, the PDP queries the PIP (4,5). The PDP evaluates the request against the policies and returns a response specifying the access decision to the PEP (6), which enforces the decision.

Token-based Architecture: Solutions adopting a policy-based architecture typically provide a single, cen-

tralized point for the evaluation and enforcement of access control policies. This solution may not be suitable when resources are distributed across different nodes, which is a typical situation in many IoT applications. The last years have seen the emergence of token-based architectures as an alternative to policy-based architectures to deal with the needs of open and decentralized systems. Roughly speaking, in a token-based architecture, an authorization service encodes the permissions of users and devices in a token, which is then used to grant them access to resources and services. Various standards have defined reference token-based architectures and authorization protocols. These architectures and protocols vary in the way tokens are generated and in the flow of the authorization process.

A widely-used token-based authorization protocol is OAuth [36]. OAuth allows client applications (web, mobile or desktop) to access resources hosted on an HTTP server with the authorization of the resource owner. The authorization granted by the resource owner is encoded in an access token. The OAuth architecture encompasses two main components: an Authorization Server, which is responsible to generate access tokens, and a Resource Server, which hosts the resources and is responsible for their disclosure thus acting as the PEP. Figure 6 presents the OAuth architecture. The client application requests access to the resource owner (1), who provides the client application an authorization grant representing the resource owner's authorization (2). The client requests an access token to the Authorization Server by presenting the authorization grant received by the resource owner (3). The Authorization Server authenticates the client application and validates the authorization grant and, if
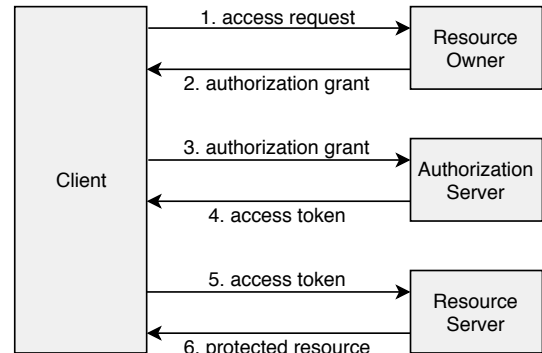
Figure 6: Token-based Architecture [36]

valid, issues an access token to the client application (4). The client application requests access to the resource to the Resource Server by presenting the access token (5). The Resource Server validates the access token and, if valid, it serves the request (6). Tokens can be reused for subsequent accesses until it is valid. When the validity of the token expires, the token is renewed through a refresh token without user intervention.

Hybrid Architecture: Token-based architectures require user intervention during policy evaluation. For instance, OAuth requires the resource owner to authorize a device or an application acting in her behalf the first time that the device/application requires access to a service or a resource, which may be inconvenient in IoT applications characterized by a large number of devices like smart manufacturing. To address this drawback, the Kantara Initiative has proposed User-Managed Access (UMA) [5]. UMA extends OAuth with the possibility of configuring policies in the Authorization Server to autonomously generate authorization tokens without user involvement. In this respect, UMA adopts an hybrid approach that combines features of policy-based and

18

token-based architectures.

5.1.3. Policy Evaluation Strategy

The architecture underlying the access control mechanism determines the strategy used for policy evaluation. In particular, we identified three main classes of policy evaluation strategies:

Run-time policy evaluation: policy evaluation is performed at request time. Upon receiving an access request, the PDP evaluates the request against the policies made available by the PAP and returns an access decision to the PEP for enforcement. This strategy is typically supported by frameworks adopting a policy-based architecture.

Off-line policy evaluation: policy evaluation is precomputed. An entity obtains assertions on its credentials, access permissions and other attributes from the resource owner or a trusted party. Then, when the entity requests access to a resource, the precomputed assertions are verified by the PEP at run time for enforcement. This strategy is typically supported by frameworks based on OAuth and UMA.[9]

Hybrid policy evaluation: Hybrid policy evaluation lies in between run-time and off-line policy evaluation. Specifically, part of policy evaluation is performed off-line, for instance using a token service that asserts the attributes and permissions of an entity within the IoT system. Then, additional verification activities are performed at request time. These activities encompass the verification of the token by the token service or by other external components (hereafter referred to as hybrid^t strategy) or the verification

of context or other constraints in order to make an access decision (hereafter referred to as hybrid^c strategy).

5.2. Requirements for Access Control in IoT

In this section, we discuss the requirements that access control systems for IoT should meet. These requirements aim to identify the main concepts and design principles that have to be considered in the design and development of access control systems tailored to IoT applications. In particular, the requirements have been distilled by applying the non-functional requirements for IoT introduced in Section 3 (see Table 3) to the different components of an access control system and from the analysis of key characteristics of IoT applications. Additionally, we have identified the ability to deal with the protection of resources and devices across multiple administrative domains, where different authorities are in control of (different parts of) the IoT ecosystem, as an important requirement in several IoT applications. Our list of requirements is reported in Table 4. The relevance of these requirements to IoT applications is presented in Table 5.

The requirements specified in Table 4 serve as a baseline for the analysis of existing authorization frameworks for IoT and the identification of gaps in the current state of the art. Note that the requirements in Table 4 are to be considered complementary to other conventional requirements typical of access control systems. We omit these conventional requirements here, as they are not specific to our discussion.

Policy Specification: IoT systems are open systems that are continuously growing with more and more entities (physical nodes and users) connected. As discussed in

---

[9]Recall that user intervention is only requested for the first access.

Category	IoT Req.	ID	Requirement
Policy Specification	GR1	**ACR1**	Access control models should allow the specification of fine-grained access control policies
	GR8, GR9	**ACR2**	Access control models should allow the specification of policies able to handle the dynamicity of nodes and IoT environments
Policy Management	GR1,GR2	**ACR3**	Access control models should be able to handle the complexity of IoT environments
	GR10	**ACR4**	Access control systems should facilitate users in policy management
	Multi Administrative domain	**ACR5**	Access control systems should enable policy management across multiple administrative domains
Policy Evaluation & Enforcement	GR2, GR10	**ACR6**	Access decision making should be automated
	GR4, GR5	**ACR7**	Access control systems should not significantly impact the computing and communication capabilities of resource-constraint devices
	GR4, GR5, GR6	**ACR8**	Access control systems should not affect the performance of the IoT system
	GR3	**ACR9**	The outcome of policy evaluation should be coherent across multiple administrative domains
	GR7	**ACR10**	Access control systems should be always operational

Table 4: Requirements for access control systems tailored to IoT environments

Section 4, IoT systems should be scalable to handle this complexity (GR1). Moreover, IoT systems should be able to handle the dynamicity of devices (GR8) and of the environment (GR9). These requirements require the access control system to support the specification of fine-grained access control policies (**ACR1**). In particular, the system should allow the specification of different access constraints for different users and physical nodes, which are tight to the devices' functionalities rather than to the devices themselves [68]. This requirement is key for all IoT applications. For example, in a smart home, the home owner may want to define a different access control policy on the smart door lock for their parents and for the cleaner [47]. Moreover, every IoT application requires the access control system to adapt to the dynamicity of the IoT ecosystem (**ACR2**), and support the specification of context-aware access control policies that impose conditions on the IoT ecosystem such as access time, location and status of the entities requesting access [98, 108]. For instance, a smart home's owner may want to grant her parents access only when they are in front of the door

or specify a policy that allows the cleaner to unlock the smart door lock only on a specific day of the week and time of the day. Failing to meet these requirements results in assigning users and applications more permissions than what needed (the so-called *overprivilege* [59, 108]), which can exploited, e.g., to compromise the system or to leak sensitive information.

Policy Management: The scalability and dynamicity of the IoT ecosystem can also lead to challenges in policy management. An access control system should be human-centric [108] and able to effectively manage the policies of multiple entities within the IoT system [82, 93]. In particular, an access control system should keep at the minimum the effort required from users to administer access control policies for multiple entities (**ACR3**) and facilitate users in policy management (**ACR4**). Usability is particularly important in scenarios such as smart homes and health IoT where end-users are in charge to define access control policies for their IoT devices and related resources but have little or no knowledge of security [65]. Moreover, in an IoT system, not all nodes might be under

	ACR1	ACR2	ACR3	ACR4	ACR5	ACR6	ACR7	ACR8	ACR9	ACR10
Smart Home (🏠)	●	●	○	●	○	●	◐	◐	○	●
Health IoT (♥)	●	●	◐	●	◐	●	●	●	◐	●
Smart Buildings (🏢)	●	●	●	○	●	●	◐	◐	●	●
Connected Vehicles (🚗)	●	●	●	○	●	●	◐	●	●	●
Smart Manufacturing (🏭)	●	●	●	○	●	●	●	●	●	●

Table 5: Relevance of requirements for IoT applications. Symbol ● is used to denote *high relevance*, ◐ *medium relevance*, and ○ *low relevance*.

the control of a single authority; instead, nodes can belong to or be managed by different administrative domains interacting together [75]. Therefore, an access control system for IoT should be able to support the management of access control policies for devices and resources across multiple domains (**ACR5**). This requirement is relevant in smart building applications where different parts of a building are rent out to different companies for office space [47]. Each company is given access to the building automation components such as the HVAC, lighting and fire alarm systems. For instance, the building owner may impose a policy stating that the lighting in the corridors is not adjustable, and it is automatically switch off when no occupancy is detected. The tenant companies instead would like to override the policy to be able to adjust the light brightness and color.

Policy Evaluation & Enforcement: Several IoT applications like health IoT, smart buildings, connected vehicles and smart manufacturing, are characterized by a strong presence of machine-to-machine interactions. These interactions require a high level of automation for the activities performed within the IoT ecosystem (e.g., data processing, communication) to guarantee the

scalability (GR2) and usability (GR10) of the IoT ecosystem. This need for automation is also reflected in the access decision making process (**ACR6**). In particular, access decision making should ideally require no user involvement. To determine whether an entity is allowed to access a certain device or a resource, its access requests have to be evaluated against the employed policies. However, IoT devices can have resource constraints [93] and, thus, computation should be minimal on the device side (GR6) and the employed network protocols lightweight (GR4,GR5). Therefore, it is desirable to limit the involvement of physical nodes in the authorization process (**ACR7**) and minimize the latency introduced by the authorization mechanism (**ACR8**), especially in IoT applications characterized by the use of resource constrained devices or latency critical applications like health IoT, connected vehicles and smart manufacturing. These requirements impose constraints also on the storage, retrieval and processing of context information for access decision making. When the IoT system is governed by multiple authorities like in connected vehicles, smart buildings and smart manufacturing applications, each administrative domain can employ an authorization mechanism based on a different access control model and/or use different data semantics. This can lead to interoperability issues during policy evaluation, which can result in an unauthorized disclosure of sensitive data and resources [21, 93]. Hence, the evaluation of an access control policy should be consistent across multiple administrative domains (**ACR9**). IoT systems should also guarantee the availability of nodes and, in particular, the ones involved in the authorization process. If one of these

nodes fails, the access control system must still be operational (**ACR10**) to meet the reliability requirements of IoT (GR7). Reliability requirements should be satisfied by any access control system regardless the IoT application where they are deployed.

5.3. Evaluation Criteria

The requirements in Table 4 define basic and desirable characteristics that an access control system for IoT should satisfy. To determine whether existing authorization solutions meet such desiderata, we have identified a number of evaluation criteria. These criteria aim to provide the basis for an assessment of the similarities and differences amongst existing authorization frameworks for IoT and their evaluation against the requirements in Table 4. The identified criteria can be grouped into two main categories.

The first category encompasses criteria concerning the properties of the authorization system. In particular, we identify the *access control model*, *policy evaluation strategy* and the *deployment configuration* of the access control mechanism within the IoT system as the main criteria to assess whether existing authorization frameworks meet the requirements in Table 4. The second category is used to assess the purpose of the proposed framework and the assumptions underlying the IoT ecosystem. It includes *IoT architecture style*, *communication protocol* and *application domain*. The IoT architecture style (Section 2.2.2) provides insights on the capabilities of nodes and their interconnections. The communication protocol used in the framework determines the communication and computing burden on physical nodes, providing additional insights

on the capabilities required from physical nodes. The application domain provides additional constraints and assumptions for the proposed framework. In our study, we have also observed that existing solutions differ significantly for *maturity level*. The extent to which access control mechanisms for IoT are actually applicable (and therefore tested) to real-world systems is important, and constitutes an additional key criteria for the evaluation of existing solutions.

In the remainder of the section, for each of the requirements in Table 4, we introduce the main criteria used to determine whether existing authorization frameworks for IoT meet the identified requirements.

Policy Specification: Authorization frameworks for IoT should support the specification of fine-grained (**ACR1**) and context-aware (**ACR2**) access control policies to meet the scalable and dynamic nature of IoT applications. The satisfaction of **ACR1** mainly depends on the underlying access control model and its support for specifying fine-grained access control policies. In particular, we evaluate the ability of a framework to selectively control access to devices and their resources based on the level of granularity in which policies can be expressed. On the other hand, we consider **ACR2** *fully satisfied* when the underlying access control model allows the specification of conditions on the context in a policy and *not satisfied* otherwise.

Policy Management: A key requirement of authorization frameworks for IoT is the ability to minimize the efforts required from users to administer access control policies of multiple entities (**ACR3**). We evaluate the ease of policy administration by considering whether an

22

authorization framework offers a single point for policy administration and facilitates the administration of policies for a large number of entities (e.g, it does not require defining a new policy every time a new entity is added to the IoT ecosystem) [19]. Accordingly, we consider the requirement *fully satisfied* when the access control system provides a single administrative point and adopts an access control model that minimizes the number of policies to be defined, *partially satisfied* when one of the two is supported, and *not satisfied* otherwise.

Usability (**ACR4**) is another important requirement especially in those IoT applications where the users in charge of defining the access control policies have no security knowledge. This requirement is considered *fully satisfied* by access control systems that support both approaches for semi-automatically or automatically generating access control policies and interfaces for policy configuration. The requirement is *partially satisfied* by approaches that support only one of these features and *not satisfied* if none of them is provided.

Management of policies across multiple domains (**ACR5**) is an important requirement in various IoT applications like smart manufacturing. Policy management is particularly challenging when devices are controlled by different authorities under different context conditions [23]. We consider this requirement *satisfied* when the authorization framework supports administrative policies or provides functions that allow entities to manage and delegate the control over devices and resources across multiple domains.

Policy Evaluation & Enforcement: Several IoT applications require a high level of automation to guarantee scalability and usability of the IoT ecosystem. This demand also reflects on the access decision making process (**ACR6**). To assess this requirement, we evaluate the degree of user involvement in the access decision process required by a given authorization framework. This involvement mainly depends on the architecture style adopted by the framework. For instance, in policy-based architecture, access decisions are made autonomously based on predefined policies without user intervention, thus *fully satisfying* the requirement. On the other hand, the satisfaction of the requirement by frameworks based on a token-based architecture depends on the standard adopted. For instance, OAuth requires users to give his/her consent the first time an application requests access to a resource or a service, thus *partially satisfying* **ACR6**. As discussed in Section 5.1.2, this issue has been addressed by hybrid architectures, e.g. based on UMA, in which tokens are generated by means of policies. Therefore, frameworks adopting an hybrid architecture *satisfy* the requirement.

The type of architecture also affects other requirements related to the performance of the access control system and to the overall performance of the IoT system in general. The performance of an access control system depends on a number of factors: *(i)* the capabilities of the components involved in making and enforcing an access decision; *(ii)* the time taken at request time to make the decision; and *(iii)* the communication among the components. Authorization frameworks for IoT should not introduce communication and computation overhead on resource-constrained devices (**ACR7**). This clearly depends on the architecture adopted to evaluate and enforce

access control policies along with the deployment of its components and the communication protocol employed by the IoT system. The architecture and its deployment determine the impact on the processing capabilities of IoT devices. Two opposite deployment solutions can be conceived for an access control system: one solution in which policy evaluation and enforcement are performed on constrained devices, and one solution in which the whole authorization process is externalized to other components. Clearly, the former does *not satisfy* the requirement while the latter *fully satisfies* it. Between these two extremes, we can find a large variety of solutions that satisfy **ACR7** to a certain degree, depending on the deployment of the access control components and the architectural style adopted. In addition, the communication protocol and data exchange format have a significant impact on the communication and computation overhead on devices.

The overhead in the computation and communication should not only be minimized for constrained devices but also for the whole access control architecture (**ACR8**). The overhead in this case is influenced not only by the location of the components involved in policy evaluation and by the adopted communication protocol, but also by the policy evaluation strategy. In particular, the policy evaluation strategy determines when access decisions are computed. On the other hand, we analyze the deployment of the components involved in policy evaluation together with the adopted communication protocol to assess the communication overhead in access decision making. For instance, the location of the PIP can help assess the overhead required to retrieval of context information. If context information is stored in a different node from

where policy evaluation is performed, there can be a delay due to its transfer.

Another critical requirement for policy evaluation and enforcement is interoperability across different administrative domains, which could potentially use different authorization frameworks (**ACR9**). To achieve interoperability among domains, policies should be interpreted and evaluated in the same way across different domains, e.g. using the same semantics across domains [52, 110]. We assume that this requirement is *satisfied* when the administrative domains use a standard format and data (semantic) model for communicating policies, tokens and authorization decisions, or when a solution for aligning policy semantics is provided.

Reliability and availability are other important requirements for an authorization framework for IoT (**ACR10**). Two main types of failures can affect the availability of the access control mechanism and the reliability of access decision decisions: failures occurring in the nodes involved in the authorization process and the lack of connectivity. In our analysis, we consider **ACR10** *satisfied* by authorization frameworks that adopt measures to deal with both types of failures. On the other hand, we consider the requirements *partially satisfied* if only one type of failure is addressed and *not satisfied* if none of these failures is addressed.

6. Analysis of Authorization Frameworks for IoT

Several frameworks and architectures have been proposed in the literature to enable authorization in IoT. In this section, we review existing proposals and analyze

them with respect to the requirements and evaluation criteria presented in the previous section. Then, we discuss their suitability to the IoT applications presented in Section 4.1.

6.1. Overview

Our analysis of the literature shows that a variety of approaches have been designed and developed to enable access control in IoT. These approaches can be broadly classified in two main categories based on the policy evaluation strategy and architecture. On one side, we have authorization frameworks [22, 23, 25, 28, 37, 38, 41, 46, 51, 60, 64, 66, 68, 70, 78, 81, 85, 87, 93, 97, 108, 118, 121] that adopt a policy-based architecture and a runtime policy evaluation strategy. Most of these frameworks are inspired to the XACML standard. On the other side, we have frameworks [32, 52, 55, 57, 58, 70, 89, 98] that adopt a hybrid-based architecture and policy evaluation strategy. A number of these frameworks build on top of OAuth by extending this standard to enable the generation of tokens based on the evaluation of access control policies like in [32, 41] whereas Rivera et al. [89] adopt UMA.

Regardless the type of access control architecture, different deployments and technologies are used to implement architecture. For example, PDP, PEP, PAP, and PIP could all be deployed in the cloud like in [22, 23, 78] or they could all be implemented on edge devices [64, 108] or a combination of both [93]. Some works [37, 38, 81] have also proposed authorization mechanisms based on blockchain technology.

Existing frameworks also vary significantly for maturity level. While a few [28, 32, 44, 46, 55, 57, 60, 64, 68, 69, 70, 78, 98] provide a prototype implementation, many [22, 23, 25, 37, 38, 41, 51, 52, 58, 66, 81, 85, 87, 89, 93, 97, 118] only remain at a conceptual level. In particular, Ray et al. [87] and Zhang and Tian [121] only propose an access control model tailored to IoT ecosystems and do not provide detail of the underlying IoT architecture and access control mechanism.

6.2. Requirements Assessment

A summary of our analysis with respect to the requirements is presented in Table 6. Tables 7 and 8 report a detailed analysis against the evaluation criteria concerning the access control system whereas Table 9 presents the analysis with respect to the criteria concerning the general characteristics of the underlying IoT ecosystem. This detailed analysis provides a rationale for the analysis in Table 6. In the tables, we use symbol "–" when a requirement/criterion is not applicable to a given framework, and symbol "?" when the information is not available.

Policy Specification: Two key requirements that have to be satisfied by an authorization framework for IoT are the specification of fine-grained (**ACR1**) and context-aware (**ACR2**) access control policies. The satisfaction of these two requirements mainly depends on the access control model adopted by the authorization framework to express access control policies.[10]

A number of existing frameworks adopt ABAC [22, 23, 57, 58, 64, 78, 87, 93, 98, 118] or UCON [51] as the

---

[10]Note that we consider the access control model in which users have to specify their policies. Therefore, even if a framework use capabilities tokens for authorization purposes, here we identify the access control model used to generate the tokens.

	Policy Specification		Policy Administration			Policy Evaluation & Enforcement				
	ACR1	ACR2	ACR3	ACR4	ACR5	ACR6	ACR7	ACR8	ACR9	ACR10
Neisse et al. [78]	✔	✔	✔	✗	✗	✔	✔	✗	✗	✔
Alshehri et al. [22]	✔	✔	✔	✗	✗	✔	✔	✗	✗	✻
Alshehri et al. [23]	✔	✔	✔	✗	✔	✔	✔	✗	✗	✻
Fremantle et al. [44]	—	✗	✗	✗	✗	✻	✔	✻	✗	✗
Fernandez et al. [41]	✻	✗	✔	✗	✗	✔	✔	✗	✗	✗
Cirani et al. [32]	?	✗	✻	✗	✗	✔	✔	✗	✗	✗
Rivera et al. [89]	?	✗	✻	✗	✗	✔	✔	✻	✗	✗
Seitz et al. [98]	✔	✔	✔	✗	✗	✔	✔	✔	✗	✗
Salonikias et al. [93]	✔	✔	✔	✗	✔	✔	✔	✗	✗	✻
Ye et al. [118]	✔	✔	✗	✗	✗	✔	✗	✔	✗	✗
Hussein et al. [57]	✔	✔	✔	✗	✗	✔	✔	✻	✗	✗
Hernandez-Ramos et al. [55]	?	✔	✻	✗	✗	✔	✔	✔	✗	✗
Gusmeroli et al. [52]	?	✗	✻	✗	✗	✔	✔	✗	✗	✗
Garcia et al. [46]	✻	✔	✗	✗	✗	✔	✗	✔	✗	✗
Dorri et al. [37, 38]	✗	✗	✗	✗	✗	✔	✔	✗	✗	✻
Ouaddah et al. [81]	✻	✗	✗	✗	✗	✔	✔	✗	✗	✻
Kim et al. [66]	✗	✻	✗	✔	✗	✔	✔	✻	✗	✗
Tian et al. [108]	✗	✻	✗	✔	✗	✔	✻	✻	✗	✗
Zhang & Tian [121]	✻	✔	?	✗	✗	✔	?	?	✗	?
Jindou et al. [60]	✻	✗	✔	✗	✗	✔	✻	✗	✗	✗
Guoping & Wentao [51]	✔	✔	✔	✗	✗	✔	✔	✗	✗	✗
Islam et al. [58]	✔	✔	✔	✗	✗	✔	✔	✗	✗	✻
Barka et al. [25]	✻	✗	✔	✗	✗	✔	✔	✗	✗	✗
Cirani & Picone [31]	—	✗	✻	✗	✗	✻	✔	✗	✗	✗
Bouij-Pasquier et al. [28]	✻	✔	✔	✗	✔	✔	✔	✻	✗	✗
Lee et al. [68]	✗	✗	✗	✗	✗	✔	✔	✗	✗	✗
Ray et al. [87]	✔	✔	?	✗	✔	✔	?	?	✗	?
Mahalle et al. [69]	✗	✗	✗	✗	✗	✔	✗	✗	✗	✗
Pinno et al. [85]	✔	✔	✻	✗	✗	✔	✗	✗	✗	✻
Kim et al. [64]	✔	✔	✔	✗	✗	✔	✔	✔	✗	✗
Sciancalepore et al. [97]	✻	✗	✗	✗	✗	✔	✗	✗	✗	✗
Mahalle et al. [70]	✗	✗	✻	✗	✗	✔	✗	✗	✗	✗
Schuster et al. [96]	?	✔	✔	✗	✔	✔	✔	✗	✔	✔

Legend

✔: satisfied ✻: partially satisfied ✗: not satisfied

—: not applicable ?: information missing

Table 6: Analysis of existing authorization frameworks for IoT with respect to requirements

underlying access control model and, therefore, satisfy both the requirements. In fact, both ABAC and UCON provide a flexible means to specify access control policies as conditions on attributes related to the entity requesting access and the resource being accessed. Moreover, ABAC intrinsically supports the specification of environment attributes that can be used to denote the context in which permissions hold, whereas UCON allows the specification of context-sensitive access constraints through conditions. Of particular interest is the work by Neisse et al. [78] that uses event-condition rules to expresses fine-grained authorizations: the action in the rule is performed when the

	Access Control Model	Context Awareness	Policy Generation	Policy Configuration	Multi Domain Administration
Neisse et al. [78]	ABAC	Yes	No	No	No
Alshehri et al. [22]	ABAC	Yes	No	No	No
Alshehri et al. [23]	ACL RBAC ABAC	Yes	No	No	Yes
Fremantle et al. [44]	–	No	No	No	No
Fernandez et al. [41]	RBAC	No	No	No	No
Cirani et al. [32]	?	No	No	No	No
Rivera et al. [89]	?	No	No	No	No
Seitz et al. [98]	ABAC	Yes	No	No	No
Salonikias et al. [93]	ABAC	Yes	No	No	Yes
Ye et al. [118]	ABAC	Yes	No	No	No
Hussein et al. [57]	ABAC	Yes	No	No	No
Hernandez-Ramos et al. [55]	?	Yes	No	No	No
Gusmeroli et al. [52]	?	No	No	No	No
Garcia et al. [46]	RBAC	Yes	No	No	No
Dorri et al. [37, 38]	ACL	No	No	No	No
Ouaddah et al. [81]	RBAC	No	No	No	No
Kim et al. [66]	ACL	Yes	Yes	Yes	No
Tian et al. [108]	ACL	Yes	Yes	Yes	No
Zhang & Tian [121]	RBAC	Yes	No	No	No
Jindou et al. [60]	RBAC	No	No	No	No
Guoping & Wentao [51]	UCON	Yes	No	No	No
Islam et al. [58]	ABAC	Yes	No	No	No
Barka et al. [25]	RBAC	No	No	No	No
Cirani & Picone [31]	–	No	No	No	No
Bouij-Pasquier et al. [28]	OrBAC	Yes	No	No	Yes
Lee et al. [68]	ACL	No	No	No	No
Ray et al. [87]	ABAC	Yes	No	No	Yes
Mahalle et al. [69]	DAC	No	No	No	No
Pinno et al. [85]	ACL + Capability + RBAC + OrBAC + ABAC+ UCON	Yes	No	No	No
Kim et al. [64]	ABAC	Yes	No	No	No
Sciancalepore et al. [97]	ABAC	No	No	No	No
Mahalle et al. [70]	Trust-based	No	No	No	No
Schuster et al. [96]	?	Yes	No	No	No

Table 7: Analysis of existing authorization frameworks for IoT with respect to the evaluation criteria concerning the access control system (1)

event is observed and the condition is satisfied. Besides being able to specify conditions on the context using data values acquired in a given moment in time (e.g., GPS location), these rules make it possible to specify context situations, which are composed data types modeling specific, complex conditions (e.g., the entity requesting access has to be not more than 100m from other entities nearby).

An exception is the framework proposed in [97]. Although based on ABAC, this framework only supports the verification of subject attributes, thus not satisfying **ACR2** and only partially satisfying **ACR1**. Similarly, frameworks based on RBAC or OrBAC [25, 28, 41, 46,

	Evaluation Strategy	Architecture Style	Deployment			
			PAP	PDP	PEP	PIP
Neisse et al. [78]	run-time	policy-based	cloud	cloud	cloud	cloud
Alshehri et al. [22]	run-time	policy-based	cloud	cloud	cloud	cloud
Alshehri et al. [23]	run-time	policy-based	cloud	cloud	cloud	cloud
Fremantle et al. [44]	off-line	token-based	–	external service	cloud	–
Fernandez et al. [41]	run-time	policy-based	external service	external service	local service	–
Cirani et al. [32]	hybrid^t	hybrid	external service	external service	local service	–
Rivera et al. [89]	off-line	hybrid	external service	external service	local service	–
Seitz et al. [98]	hybrid^c	hybrid	cloud	cloud + physical node	physical node	physical node
Salonikias et al. [93]	run-time	policy-based	cloud	edge	edge	cloud
Ye et al. [118]	run-time	policy-based	physical node	physical node	physical node	physical node
Hussein et al. [57]	hybrid^c	hybrid	external service	external service + edge	edge	–
Hernandez-Ramos et al. [55]	hybrid^c	hybrid	external service	external service + physical node	physical node	physical node
Gusmeroli et al. [52]	hybrid^t	hybrid	external service	external service	physical node	–
Garcia et al. [46]	run-time	policy-based	physical node	physical node	physical node	physical node
Dorri et al. [37, 38]	run-time	policy-based	edge	edge	edge	–
Ouaddah et al. [81]	run-time	policy-based	physical node	edge + physical node	physical node	physical node
Kim et al. [66]	run-time	policy-based	edge	edge	edge	external service
Tian et al. [108]	run-time	policy-based	edge	edge	edge	edge
Zhang & Tian [121]	run-time	policy-based	?	?	?	?
Jindou et al. [60]	run-time	policy-based	external service 1	external service 1	physical node	external service 2
Guoping & Wentao [51]	run-time	policy-based	external service 1	external service 1	external service 1	external service 2
Islam et al. [58]	hybrid^c	hybrid	cloud	cloud (+ physical node)	cloud (physical node)	cloud
Barka et al. [25]	run-time	policy-based	external service 1	external service 1	external service 2	–
Cirani & Picone [31]	hybrid^t	token-based	external service	external service	local service	–
Bouij-Pasquier et al. [28]	run-time	policy-based	external service	edge	edge	physical node
Lee et al. [68]	run-time	policy-based	external service	external service	external service	–
Ray et al. [87]	run-time	policy-based	?	?	?	?
Mahalle et al. [69]	run-time	hybrid	physical node	physical node	physical node	–
Pinno et al. [85]	run-time	policy-based	physical node	physical node	physical node	physical node + external service
Kim et al. [64]	run-time	policy-based	edge	edge	edge	edge
Sciancalepore et al. [97]	run-time	policy-based	physical node	application node + physical node	physical node	external service
Mahalle et al. [70]	run-time	policy-based	physical nodes	physical node	physical node	peers
Schuster et al. [96]	run-time	policy-based	?	?	?	physical node + external service

Table 8: Analysis of existing authorization frameworks for IoT with respect to the evaluation criteria concerning the access control system (2)

51, 60, 81, 121] only partially satisfy **ACR1** since these models only allow abstracting permissions at the level of role. Moreover, core RBAC does not support the notion of context and, thus, frameworks based on this model do not satisfy **ACR2**. On the other hand, OrBAC allows the specification of context-sensitive access constraints by explicitly representing the context in which permissions holds

and, thus, frameworks based on this model satisfy **ACR2**.

We also found a few frameworks [37, 66, 68, 69, 108] based on identity-based access control models and, in particular, ACLs. These models, in general, do not satisfy requirements **ACR1** and **ACR2** because they only allow a direct assignment of access rights to users and do not support the notion of context. It is worth noting that some

	IoT Architecture style	Communication Protocol	Data Format	Cross-domain Data Semantics	Node Failure Robustness	Application Domain	Maturity Level
Neisse et al. [78]	centralized	MQTT	?	No	Yes	–	prototype
Alshehri et al. [22]	centralized	–	–	No	Yes	Lighting†	design
Alshehri et al. [23]	centralized	–	–	No	Yes	Connected Vehicles†	design
Fremantle et al. [44]	centralized	MQTT	JSON	No	No	–	prototype
Fernandez et al. [41]	centralized	–	–	No	No	–	design
Cirani et al. [32]	distributed	6LoWPAN CoAP	?	No	No	–	prototype
Rivera et al. [89]	centralized	–	–	No	No	Traffic Lights†	design
Seitz et al. [98]	distributed	CoAP	XACML JSON	No	Yes	–	prototype
Salonikias et al. [93]	connected	–	–	No	Yes	Connected Vehicles	design
Ye et al. [118]	distributed	–	–	No	No	Wireless Sensor Network†	design
Hussein et al. [57]	connected	?	JSON	No	No	Smart Home†	prototype
Hernandez-Ramos et al. [55]	connected	6LoWPAN CoAP	JSON	No	Yes	–	prototype
Gusmeroli et al. [52]	connected	–	–	No	Yes	–	design
Garcia et al. [46]	distributed	6LoWPAN	?	No	No	Medical Sensor Networks	prototype
Dorri et al. [37, 38]	distributed	–	–	No	Yes	Smart Home†	design
Ouaddah et al. [81]	distributed	–	–	No	No	–	design
Kim et al. [66]	connected	–	–	No	No	Smart Home	design
Tian et al. [108]	connected	HTTP	?	No	Yes	Smart Home	product
Zhang & Tian [121]	?	–	–	No	No	–	design
Jindou et al. [60]	connected	HTTP	JSON	No	No	Smart Home†	prototype
Guoping & Wentao [51]	centralized	–	–	No	No	–	design
Islam et al. [58]	connected	–	JSON	No	Yes	Health Prescription Assistant	design
Barka et al. [25]	centralized	–	–	No	No	–	design
Cirani & Picone [31]	distributed	CoAP	?	No	No	–	prototype
Bouij-Pasquier et al. [28]	connected	CoAP	JSON	No	No	Health IoT†	prototype
Lee et al. [68]	distributed	Wi-Fi	?	No	No	Smart Lock†	prototype
Ray et al. [87]	?	–	–	No	No	Remote Healthcare Monitoring	design
Mahalle et al. [69]	distributed	Wi-Fi	?	No	No	–	prototype
Pinno et al. [85]	distributed	–	–	No	Yes	–	design
Kim et al. [64]	centralized	ZigBee	?	No	No	Smart Home	prototype
Sciancalepore et al. [97]	connected	–	–	No	No	–	design
Mahalle et al. [70]	distributed	?	?	No	No	–	prototype
Schuster et al. [96]	distributed	HTTP	?	No	Yes	Smart Home	prototype

Table 9: Analysis of existing authorization frameworks for IoT. Symbol † indicates that only a toy example is provided for demonstration purposes, but the framework is not designed specifically for that application domain.

of the frameworks based on RBAC (i.e., [46, 121]) and DAC (i.e., [66, 108]) have been extended, often in an ad-hoc fashion, to account for contextual information during policy evaluation. For instance, in [46] access is granted based on the user's health condition and other context information: if the user's health condition is

critical, access is given to any doctor or medical staff to face emergency. However, some of them are limited in the type of context information that can be specified in policies, thus only partially satisfying **ACR2**. In particular, the frameworks in [66, 108] only support the specification concerning the location of the requester.

Some authorization frameworks for IoT are not based on standard access control models. For instance, Mahalle et al. [70] propose a trust-based access control model in which access is granted based on the trustworthiness of the requester. This model, however, only allows the specification of permissions at device level, resulting in coarse grained access control policies. This leads to the problem of overprivilege as users and applications have more capabilities than needed [68, 108]. Moreover, it does not account for context information for access decision making, thus not meeting **ACR2**.

Finally, we observed that the framework in [44] does not evaluate access requests against access control policies represented according to a specific model. This framework is based on the OAuth 2.0 authorization protocol where access to a resource is granted explicitly by the resource owner. On the other hand, the frameworks in [32, 52, 55, 89] use policies to generate capability tokens but they do not discuss the access control model employed to generate those tokens. Accordingly, **ACR1** and the access control model for these frameworks are marked with symbol "?" in Tables 6 and 7, respectively.

The analysis of existing frameworks in light of requirements **ACR1** and **ACR2** shows that existing frameworks have adopted a variety of access control models (i.e., ABAC, UCON, OrBAC, RBAC, ACL or capabilities) to

express access control policies or rely on user intervention (i.e., frameworks based on OAuth 2.0). However, only frameworks based on ABAC and UCON allow the specification of fine-grained and context-aware access control policies and therefore fully satisfy both requirements. It is interesting to observe that we did not find any framework that uses MAC as the underlying access control model. We speculate that this is due to the fact that MAC is a very static and rigid access control model and, thus, not suitable to cope with the dynamicity characterizing most IoT applications.

Policy Management: Several IoT applications require the access control system to deal with the management and protection of several entities (**ACR3**). As discussed in Section 5.3, this requirement is satisfied by authorization frameworks that offer users a single point for policy administration and make use of a flexible access control model. These criteria are usually met by authorization frameworks based on ABAC and UCON (and on RBAC and OrBAC to a certain extent) that are used in combination with a message broker to regulate the subscription to and the publishing of resources (e.g., [22, 23, 78]). In particular, ABAC and UCON provide a flexible approach for the specification of access control policies. RBAC and OrBAC have been introduced to simplify the specification and management of access control policies compared to identity-based models (e.g., DAC and MAC), but they may require defining new roles with specific permissions leading to role explosion [39]. A single administrative point is also offered by authorization frameworks deployed as external services [25, 32, 41, 51, 57, 60, 89] or in the

cloud [58, 98]. This is also the case of the frameworks proposed by Salonikias et al. [93] and by Bouij-Pasquier et al. [28] in which the PAP is deployed in the cloud or in an external service respectively, whereas the other components of the access control system are deployed in edge nodes. On the other hand, the requirement is not satisfied by frameworks in which policies have to be deployed in physical nodes and/or adopt an identity-based access control model (e.g., [37, 38, 46, 66, 68, 69, 81, 108, 118]). In fact, both approaches require to set the permissions for each new device. In particular, some frameworks use ACLs, motivated by the simplicity of policy specification, but they have not been proven at a large scale. An exception is the framework by Mahalle et al. [70] that, although policies are deployed in physical nodes, they are predefined and do not need to be deployed when new devices or services are added to the system. However, the evaluation of these policies requires users to define and retrieve information about experience, knowledge and recommendations about other devices. Thus, we mark **ACR3** partially satisfied by this framework. On the other hand, **ACR3** is not satisfied by frameworks of Fremantle et al. [44]. This framework is based on OAuth and, thus, a user has to manually grant permission to each application and device that request access to his/her resources (see Section 5.1.2). The frameworks proposed in [32, 52, 55, 89] could potentially satisfy the requirement depending on the access control model adopted to issue capability tokens. Since those frameworks offer a single point for policy administration but the access control model is not discussed, we mark the requirement partially satisfied for these frameworks.

An aspect that is neglected by most of the existing authorization frameworks for IoT is usability (**ACR4**). This is a key requirement in scenarios like smart homes and health IoT where the users that are in charge of protecting devices and resources, often lack the security expertise necessary to specify access control policies [65, 72]. Therefore, access control systems for these IoT applications should provide users with an interface that suggests the access control policies to be enforced, displays the current policies, and allows to modify the policies as needed. Only two frameworks [66, 108] provide a full solution to usability. Kim et al. [66] proposed an access control system called CARA, which automatically suggests the access control policies to be assigned to the visitors of a smart home. These policies are defined based on three main access control constraints: *presence*, which requires the visitor to be in the house in order to access a device/resource, *logs*, which requires the device to maintain logs, and *ask for permission*, which requires the visitor to explicitly ask the permission to access a device/resource to the home owner. These three constraints are used to define four basic policy configurations – full, restricted, partial and minimal control – on resources/devices that reflect the level of trust the home owner places into the visitor. For instance, if the visitor is highly trusted by the home owner, e.g. a family member, he will be assigned to the full policy configuration that gives him full access and control to all devices and resources in the house when he is physically in the house. Policy configurations are automatically assigned to users based on the social relationship between the home owner and the user, which is inferred based on the social network

graph information or phone usage information, e.g. users called more frequently. The automatic assignment of a policy configuration to a visitor, on one side, simplifies the owner's task but, on the other side, can lead to assign a wrong policy configuration to visitors. The authors suggest that policy configurations are preloaded into the devices by the manufacturer and changed manually by the home owner if needed.

Existing authorization frameworks typically assume that policies are predefined by users (possibly with automated aid as in the case of [66]). However, this permission model is not suitable when users have to confirm the permissions asked by IoT applications. In this setting, applications can require more permissions than what actually needed, thus resulting in *overprivilege*. To address this issue while minimizing user burden, Tian et al. [108] propose an approach to automatically generate access control policies to grant access to devices/resources to the smart phone app that the home owner uses to control the resources/devices. The approach derives the policies by identifying any possible discrepancy between the functionality exhibited by the mobile app's code, e.g. unlocking the door, and the app description, e.g. switching on the coffee machine. If the functionality of the app code matches the one in the app description, the app is automatically authorized to perform it. Otherwise, if there is a mismatch, the app is automatically blocked. If the analysis of the app code reveals that the app not only switches on the coffee machine but also unlock the door, the unlock door functionality is flagged as a mismatched functionality. The generated policies are displayed to users through the mobile app interface: the verified functionalities are labeled in green while mismatches are labeled in red so that the home owner can understand that they represent a potential risky behavior of the app.

Another aspect that has been marginally investigated is the management of access control policies across multiple administrative domains (**ACR5**) that is particularly relevant in IoT applications like connected vehicles, smart buildings and smart manufacturing. Our analysis of the literature shows that most of the existing authorization frameworks fail to meet this requirement as they implicitly assume that resources and devices are under the control of a single authority. Notable exceptions are the works in [23, 87], which propose an approach for policy administration tailored to IoT ecosystems, and the work by Salonikias et al. [93]. In particular, the latter introduces the notion of propagation policy and proposes a policy propagation method to update all PDPs (deployed in edge nodes and under the control of possible different authorities) when policies are modified in a centralized PAP. On the other hand, Bouij-Pasquier et al. [28] introduce a collaboration layer to handle multi-party collaborative interactions. In particular, the authors propose a negotiation of access rules for cross-domain sharing of resources and information.

Policy Evaluation & Enforcement: The ability to automate the evaluation of an access request (**ACR6**) is an important requirement in all IoT applications where a multitude of devices and users share information. Assuming that a user is always available to evaluate if access to certain resource should be granted is not realistic. To automate the evaluation of an access re-

quest, existing frameworks adopt either a policy-based [22, 23, 25, 28, 37, 38, 41, 46, 64, 66, 68, 70, 78, 81, 85, 87, 93, 97, 108, 118, 121] or a hybrid architecture [32, 52, 55, 57, 58, 89, 98]. In a policy-based architecture access requests are evaluated against a predefined set of access control policies; while in hybrid architectures authorization tokens are issued based on the evaluation of access control policies. The only frameworks that do not fully satisfy the requirement is the one by Fremantle et al. [44] and Cirani and Picone [31]. The framework proposed in [44] is based on the OAuth protocol, which requires the resource owner to grant access to the application the first time an authorization token is issued. Similarly, Cirani and Picone [31] require the resource owner's involvement in the issuing of tokens. In particular, they account for three operational modes to obtain the tokens: *owner-to-owner*, in which a user registers his/her own device and obtains a token with all permissions on the device; *reactive owner-to-any*, in which the owner grants permission upon a user's request; and *proactive owner-to-any*, in which the owner proactively grants permission to a user.

Another key requirement for authorization frameworks designed for IoT applications is that they should not introduce communication and computation overhead on resource-constrained devices (**ACR7**). This requirement is typically addressed by outsourcing the most computationally expensive operation, namely policy evaluation, to an external service while performing only the enforcement of the access decision on constrained devices. Our analysis shows that most of the frameworks that adopt a policy-based architecture [22, 23, 25, 28, 37, 38, 41, 66, 68, 78, 81, 93, 108] ex-

ternalize the PDP and the PAP (i.e., these components are not deployed in the physical node), thus fully satisfying the requirement. The only exceptions are the frameworks proposed in [46, 118] in which the PDP and the PAP run on the constrained device. Similarly, frameworks that rely upon a token-based or a hybrid architecture [32, 52, 55, 57, 58, 89, 98] fully satisfy the requirement. These frameworks employ dedicated services for the generation and issue of authorization tokens, and only the validation of the token is performed on the device. However, an aspect that should be considered is the size and format of the authorization token that could introduce a computation overhead on a constrained device. Lightweight standards to represent tokens like JSON should be preferred over XML-based formats like the one supported by SAML.

On the other hand, the performance of an access control system (**ACR8**) not only depends on the location of the components involved in the policy evaluation and communication protocol, but also on the policy evaluation strategy. Frameworks that use an off-line evaluation strategy (i.e., [44, 89]) or an hybrid strategy in which only context constraints are verified at run-time (hybrid^c) and their verification does not require retrieving information from other components (e.g., [46, 55, 98, 118]), do not introduce latency in the access decision making process. Similarly, latency is limited if policy evaluation is performed on the edge like in [28, 57, 64, 108]. On the other hand, policy-based frameworks in which the access control mechanism is deployed in the cloud or provided as an external service (e.g., [22, 23, 25, 41, 60, 78, 121]) might introduce delay due to additional communication. This is

also the case of frameworks that require validating tokens at run-time (hybrid^t) like in [31, 32, 52], or that require retrieving contextual information from external sources or from the cloud like in [60, 65, 93, 96, 121]. On top of this, the communication protocol has a significant impact on the overall performance of the IoT ecosystem where frameworks based on lightweight protocols like MQTT and CoAP provide better performance compared to the ones based on HTTP. Frameworks based on blockchain technology (e.g., [37, 38, 81, 85]) also do not satisfy the requirement due to time required to confirm a transaction. Every time an access control policy has to be added to or retrieved from the blockchain, a new transaction has to be created and added to the blockchain. Before a transaction can be added to the blockchain, special nodes called miners run a consensus protocol that requires them to verify each transaction. The time to complete the validation process is typically in the order of minutes [80], which is clearly unsuitable for most IoT applications, especially for the ones that are latency sensitive.

Other key requirements for policy evaluation are interoperability (**ACR9**) and reliability/availability of components (**ACR10**) involved in the evaluation of the policies. However, despite their importance these two requirements are only marginally considered by existing authorization frameworks for IoT. Some of the frameworks only scratch the surface of the interoperability problem because they use a standard like XACML to specify the access control policies [98] or they encode the capability token in JSON [44, 55, 57, 60, 98]. Interestingly, Seitz et al. [98] provide an encoding of SAML assertions in JSON, while the others propose an ad-hoc format to encode tokens.

However, using a standard only facilitates the exchange of policies or tokens across multiple domains but not their interpretation. If different authorities define their policies based on different semantic models, the collaborative evaluation of these policies can result in granting access to users for which access should be denied.

Reliability and availability (**ACR10**) is *fully satisfied* by those frameworks that can tolerate the failure of an architectural component and of the communication among them. Most of the frameworks partially satisfy the requirement because they only address the reliability/availability of the components but not of the communication among them. To address the failure of an architectural component, three main solutions have been adopted by existing authorization frameworks. Some frameworks have deployed the components in the cloud [22, 23, 58, 78, 93, 98], which guarantees that the components are evenly distributed across different servers, which are connected to work as one. Therefore, if one server fails, downtime is avoided. Salonikias et al. [93] instead ensure reliability and availability by replicating the PDP and the PEP and by defining propagation policies that specify how access control policies should be exchanged between PDPs. Frameworks based on blockchain [37, 38, 81, 85] propose to deploy and maintain a copy of the components of the authorization framework in all nodes forming the blockchain, thus ensuring resilience against failures of architecture components. The only framework proposed that fully satisfies **ACR10** is the one proposed by Neisse et al. [78], which adopts a reliable communication protocol like MQTT besides addressing the reliability of the architectural components.

34

Implementation and Evaluation: An important aspect is the *applicability* of an access control framework to real IoT applications. In this respect, most of the proposed frameworks [22, 23, 25, 37, 38, 41, 51, 52, 58, 66, 81, 89, 93, 97, 118] only present the architecture of the access control mechanism and demonstrate the authorization flow among the components based on a realistic IoT use cases. For example, Dorri et al. [89] have illustrated their access control framework based on a smart home scenario. However, use cases do not provide insights on the effectiveness of the framework in realistic IoT settings. Only implementing the framework on a real IoT system and evaluating its performance and usability can provide such insights. Nonetheless, only few of the proposed frameworks have been implemented and evaluated [32, 46, 68, 70, 78, 98], while other works only report a prototype implementation [28, 44, 55, 57, 60, 64]. For instance, Neisse et al. [78] have proposed an authorization framework for MQTT brokers. The enforcement of access control policies is performed by a PEP that is integrated into the browser, while policy evaluation is done by an external PDP and Context Manager. The MQTT broker has been implemented using the Mosquitto library and its performance evaluated in terms of overhead introduced in the communication by implementing the PEP in the MQTT broker. Cirani et al. [32] have instead focused on evaluating the performance of their access control framework on constrained devices. In particular, they evaluated the energy and memory consumption of policy evaluation on a Contiki-based devices. To run the evaluation, they used the Cooja simulator and considered Zolertia Z1 nodes with 92KB ROM and 8kb RAM. Similarly, Garcia et al. [46] have evaluated the performance of their framework on constrained devices but using a real testbed rather than a simulation environment like Cooja. The testbed consisted of Arduino Mega 2560 board3 with 16 MHz processor, 256 kB of Flash Memory, 8 kB of SRAM, and 4 kB of EEPROM.

6.3. Discussion

Our analysis of the literature shows that there is no *one-size-fits-all* authorization framework for all IoT applications. Each IoT application has its own set of requirements that should be satisfied when designing an authorization framework specific to that application. The main difference lies in the requirements imposed by each application on policy management and evaluation, while the requirements on policy specification are the same for all IoT applications. Regardless the specific application, an authorization framework for IoT should support the specification of fine-grained (**ACR1**) and context-aware (**ACR2**) policies. Both requirements are satisfied when the authorization framework adopts either ABAC or UCON as the underlying access control model (see Section 6.2 for a detailed discussion).

With respect to the requirements on policy management and evaluation, we can divide IoT applications in three main groups: the first group is formed by smart homes, the second one by health IoT and the third group is composed by smart buildings, connected vehicles and smart manufacturing. Requirements imposed by each group of applications on policy management and evaluation are quite different as shown in Table 5. The only exception is requirement **ACR10** on the reliability and availability of the architectural components involved in the authorization

process, which is required by each group of applications.

In smart homes, home owners are in charge of specifying the access control policies to protect a relatively small number of IoT devices but they typically do not have the necessary security knowledge. Therefore, usability (**ACR4**) is a key requirement in smart homes and should be addressed by minimizing the efforts of home owners in specifying access control policies. The ideal authorization framework for smart homes should be based on a centralized and policy-based architecture where access decision are made based on access control policies that are not defined by the home owners but automatically generated taking into account the context of access. Moreover, the PAP should support home owners in the configuration and modification of their policies. Since latency can be tolerated in smart homes applications, a run-time policy evaluation strategy can be adopted. The PDP could be deployed on edge devices like IoT gateways or on a local cloud. The frameworks presented in [64, 66, 108] are specifically designed for smart homes but they do not address all relevant requirements for this IoT application. Kim et al. [66] and Tian et al. [108] provide a mechanism to suggest access control policies to home owners, but the policies generated are coarse-grained and only impose simple conditions on the environment. In contrast, Kim et al. [64] does not consider the usability issues related to the specification of access control policies by lay users, but allow the specification of fine-grained and context-aware access control policies. None of the frameworks ensure reliability and availability of the components involved in the policies evaluation and enforcement.

Similarly to smart homes, health IoT applications re-quire lay users to be responsible for the specification of access control policies and, thus, have to satisfy the same requirements with respect to usability. In contrast, health IoT applications involve a large number of IoT devices, e.g. insulin pumps and pacemakers, that feed sensitive medical data directly in patients' electronic healthcare records. Therefore, it is important that the authorization framework minimizes the effort of lay users (e.g., patients and medical staff) in administering policies for multiple devices (**ACR3**), takes into account the constrained capabilities of medical devices (**ACR7**) and reduces the latency in decision making (**ACR8**), which can be potentially life threatening. In order to meet all these requirements, an authorization framework for health IoT applications should adopt a hybrid policy evaluation strategy and architecture. Similarly to smart homes, the PDP should be configured with access control policies that are automatically generated rather than being defined by patients. Moreover, to take into account the constrained capabilities of medical devices and minimize the latency to evaluate and enforce the policies, the PDP should be deployed on an edge device while the PEP could be located on the devices. Garcia et al. [46] and Ray et al. [87] have proposed on an authorization framework that enables remote patient monitoring. While Garcia's framework supports the specification of context-aware access control policies and a lightweight mechanism that efficiently runs on constrained sensor nodes, Ray and colleagues only propose a fine-grained access control model inspired to the XACML and NIST NGAC [42] standards. Both frameworks do not provide a solution to address key requirements like usability of policy specification and adopt a runtime policy

evaluation strategy and a policy-based architecture that increase the latency of access decision making.

Unlike smart homes and health IoT, which are characterized by a high user involvement, smart buildings, connected vehicles and smart manufacturing applications mainly involve a large number of IoT devices that directly communicate with each other. Often these devices are not managed by a single authority but they belong to different administration domains. Therefore, unlike smart homes and health IoT applications, usability is not a key requirement. On the other hand, the ability of supporting the management of policies across different domains (**ACR5**), ensuring interoperability among domains (**ACR9**), and automating an access control decision (**ACR6**) are fundamental requirements. Similarly to health IoT applications, smart buildings, connected vehicles and smart manufacturing applications are time-critical applications and, therefore, the authorization framework should adopt a policy evaluation strategy and an architecture that reduce the latency of the access decision making process (**ACR8**). Therefore, the authorization framework should be similar to the one discussed for health IoT applications but the PDP should also provide functionalities to take access decisions based on policies from different administrative domains and guarantee the correct interpretation of these policies. While there are no authorization frameworks specific to smart buildings and smart manufacturing applications, Salonikias et al. [93] proposed an authorization framework for connected vehicles that satisfies most of the above requirements except the one related to latency. The authors proposed an XACML-like architecture that consists of multiple PDPs and PEPs located at the edge,

while a single PAP deployed in the cloud is responsible to maintain and propagate access control policies to the PDPs. However, the communication among the PAP and the PDPs increases the time needed to take an access decision, thus not satisfying **ACR8**.

7. Lessons Learned and Open Challenges

This section summarizes the lessons learned that should be taken into account when designing an authorization mechanism for IoT and discusses open challenges.

There is no need of new access control models. Several of the analyzed works indicate that access constraints for IoT systems should account for the context (e.g., location, time) and often propose ad-hoc (extensions of) policy languages to represent such access constraints. As discussed in the previous sections, both ABAC and UCON have proven capable to express a large range of access control policies and allow the specification of fine-grained and context-aware access control policies. In particular, these models allow the specification of permissions at the level of devices' functionalities, which is necessary to avoid application overprivilege. Therefore, regardless the IoT application, any authorization framework should adopt one of these models as the underlying access control model.

There is no one-size-fit-all authorization framework for all IoT applications. Although many authorization frameworks for IoT have been proposed, only few of them have been designed for a specific IoT application and even fewer address all the unique requirements of the application that they are meant to protect. The main research challenge is thus to design an authorization framework

that satisfies the requirements related to policy management and evaluation specific to a target IoT application.

OAuth is not suitable for most IoT applications. Many initiatives from standardization bodies and industry aim to adapt the OAuth authorization protocol to IoT. OAuth could be potentially applied to smart home applications because it partially solves the usability issues related to lay users being in charge of specifying access control policies. In particular, OAuth does not require home owners to specify access control policies but they have to grant access to a smart device or appliance when it is requested. However, the permissions granted by home owners are coarse-grained because they give access to the whole IoT device rather than only on specific resources and services provided by the device itself. Moreover, OAuth requires human intervention to take an authorization decision, which makes it unsuitable for IoT applications involving a high number of devices that directly communicate with each other like connected vehicles and smart manufacturing and for IoT applications involving machine-to-machine interaction. UMA improves over OAuth with respect to human involvement in access decision making because it does not require the resource owner to be online at the time of access request but it handles requests based on access control policies predefined by the owner. However, this introduces usability issues because home owners might not be security experts. UMA has also been proposed to achieve privacy and compliance with data protection principles like informed consent because users explicitly grant access to their personal data and determine who access their data, for how long,

and under what circumstances. While UMA certainly empowers users with control over their personal data, it does not address the requirements on collecting users' consent imposed by the GDPR[11], the new EU regulation on data protection. The GDPR requires individuals to explicitly give their consent to collect their personal data and that when the consent is given they are informed of data collection purposes and the nature of the data processing in a clear, ease to understand and concise language. The mechanism adopted by UMA to obtain users' consent is explicit but not informed because consent is collected without any clear and concise explanation of the purpose for which the data are accessed and how they are going to be processed.

Besides distilling the above lessons learned, we have identified a number of aspects that have been neglected by most of existing frameworks and for which solutions are yet to be provided.

Usability: Usability is a largely unexplored aspect for IoT applications like smart home and health IoT, which are characterized by a high user involvement. The few efforts that have aimed to address usability issues do not take into account that these applications involve both machine-to-machine and user-to-machine interactions. They either generate access control policies for different users that could interact with the IoT devices or to restrict a device's access to another device.

Multi-domain policy administration: Most of the proposed frameworks assume that access control policies for IoT devices and resources are managed by a single authority.

---

[11] https://gdpr-info.eu/

However, this is only a realistic assumption for smart home applications where only home owners are in charge of protecting the smart devices and appliances in their home. Other applications like smart buildings, connected vehicles and smart manufacturing involve users, IoT devices, services managed by authorities that belong to different domains. An authorization framework for these applications should support a policy governance model able to reconcile policies from different domains. The framework should also provide a solution to resolve inter-operability issues due to the use of a different semantic to define the policies across different domains.

Reliability and Availability: Reliability and availability of the components involved in the evaluation and enforcement of the access control policies is as important as the reliability and availability of the IoT devices deployed in a particular IoT application. If the PDP fails to take an authorization decision, this affects the performance of the whole IoT ecosystem. Despite the importance of this requirement, only one of the analyzed authorization frameworks for IoT has addressed it.

Lack of validation: Our analysis reveals that most frameworks (e.g., [22, 23, 41, 93, 118]) are still at a conceptual level and lack a proof-of-concept implementation. Although a few frameworks have been implemented (e.g., [32, 78, 98]), they often lack a validation within large scale IoT systems. This makes it difficult to evaluate whether they meet requirements such as scalability and performance, and thus to assess their suitability to cope with realistic IoT scenarios. We believe that this is a step necessary for the transfer of research efforts into real IoT applications.

8. Conclusion

This paper has provided an analysis of existing authorization frameworks for IoT. Our goal was to identify the main research trends and developments in this area.

We have identified several important requirements to support access control in IoT driven by the non-functional requirements to be met by IoT systems and the demands of IoT applications. By analyzing the current state-of-the-art against these requirements, we observed that there is no one-size-fits-all access control system for all IoT applications. The main research challenge in the design of an authorization framework for IoT lies in devising an architecture that meets the requirements specific to the target IoT application.

References

[1] Interagency Report on Status of International Cyber-security Standardization for the Internet of Things (IoT). `https://csrc.nist.gov/CSRC/media/Publications/nistir/8200/draft/documents/nistir8200-draft.pdf`. Accessed: 2018-02-21.

[2] Mosquitto: An Open Source MQTT v3.1/v3.1.1 Broker. `https://mosquitto.org`.

[3] OWASP Top 10 IoT Vulnerabilities. `https://www.owasp.org/index.php/Top_IoT_Vulnerabilities`. Accessed:2018-11-15.

[4] Standard for an Architectural Framework for the Internet of Things (IoT). `https://standards.ieee.org/develop/project/2413.html/`. Accessed: 2018-02-20.

[5] User-Managed Access (UMA) 2.0 Grant for OAuth 2.0 Authorization. `https://kantarainitiative.org/file-downloads/rec-oauth-uma-grant-2-0-pdf/`. Accessed: 2018-02-26.

[6] Internet Protocol. RFC 791, 1981.

[7] Transmission Control Protocol. RFC 793, 1981.

[8] REST. Semantic Web Standard, W3C Semantic Web, 2011.

[9] eXtensible Access Control Markup Language (XACML) Version 3.0. OASIS Standard, OASIS, 2013.

[10] Information technology – Radio frequency identification for item management – Part 6: Parameters for air interface communications at 860 MHz to 960 MHz General. ISO/IEC 18000-6:2013, International Organization for Standardization, 2013.

[11] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Std 802.11ac-2013, IEEE Standards Association, 2013.

[12] MQTT Version 3.1.1. OASIS Standard, OASIS, 2014.

[13] ZigBee Specification. Standard, ZigBee Alliance, 2014.

[14] IEEE Standard for Ethernet. IEEE Std 802.3-2015, IEEE Standard Association, 2015.

[15] IEEE Standard for Low-Rate Wireless Networks. IEEE Std 802.15.4-2015, IEEE Standard Association, 2015.

[16] GSM/EDGE Radio Transmission and Reception. 3GPP TS 45.005, 3GPP, 2017.

[17] M. Aazam, I. Khan, A. A. Alsaffar, and E.-N. Huh. Cloud of Things: Integrating Internet of Things and cloud computing and the issues involved. In *Proceedings of International Bhurban Conference on Applied Sciences & Technology*, pages 414–419. IEEE, 2014.

[18] M. R. Abdmeziem, D. Tandjaoui, and I. Romdhani. Architecting the Internet of Things: State of the Art. In *Robots and Sensor Clouds*, pages 55–75. Springer, 2016.

[19] T. Ahmad, U. Morelli, S. Ranise, and N. Zannone. A Lazy Approach to Access Control as a Service (ACaaS) for IoT: An AWS Case Study. In *Proceedings of the 23nd ACM Symposium on Access Control Models and Technologies*, pages 235–246. ACM, 2018.

[20] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys Tutorials*, 17(4):2347–2376, 2015.

[21] S. Alam, M. M. Chowdhury, and J. Noll. Interoperability of security-enabled Internet of Things. *Wireless Personal Communications*, 61(3):567–586, 2011.

[22] A. Alshehri and R. Sandhu. Access Control Models for Cloud-Enabled Internet of Things: A Proposed Architecture and Research Agenda. In *Proceedings of International Conference on Collaboration and Internet Computing*, pages 530–538. IEEE, 2016.

[23] A. Alshehri and R. Sandhu. Access Control Models for Virtual Object Communication in Cloud-Enabled IoT. In *Proceedings of International Conference on Information Reuse and Integration*, pages 16–25. IEEE, 2017.

[24] L. Babun, A. K. Sikder, A. Acar, and A. S. Uluagac. Iotdots: A digital forensics framework for smart environments. *CoRR*, abs/1809.00745, 2018.

[25] E. Barka, S. S. Mathew, and Y. Atif. Securing the Web of Things with Role-Based Access Control. In *Codes, Cryptology, and Information Security*, pages 14–26. Springer, 2015.

[26] Bluetooth SIG Working Group. The building blocks of your Bluetooth device. Bluetooth Core Specification, 2017.

[27] A. Botta, W. De Donato, V. Persico, and A. Pescapé. Integration of Cloud Computing and Internet of Things: a Survey. *Future Generation Computer Systems*, 56:684–700, 2016.

[28] I. Bouij-Pasquier, A. A. El Kalam, A. A. Ouahman, and M. De Montfort. A Security Framework for Internet of Things. In *Cryptology and Network Security*, pages 19–31. Springer, 2015.

[29] I. Bouij-Pasquier, A. A. Ouahman, A. A. El Kalam, and M. O. de Montfort. SmartOrBAC security and privacy in the Internet of Things. In *Proceedings of International Conference on Computer Systems and Applications*, pages 1–8. IEEE, 2015.

[30] Z. B. Celik, L. Babun, A. K. Sikder, H. Aksu, G. Tan, P. McDaniel, and A. S. Uluagac. Sensitive information tracking in commodity iot. In *Proceedings of USENIX Security Symposium*. USENIX Association, 2018.

[31] S. Cirani and M. Picone. Effective authorization for the Web of Things. In *Proceedings of World Forum on Internet of Things*, pages 316–320. IEEE, 2015.

[32] S. Cirani, M. Picone, P. Gonizzi, L. Veltri, and G. Ferrari. IoT-OAS: An OAuth-Based Authorization Service Architecture for Secure Services in IoT Scenarios. *IEEE Sensors Journal*, 15(2):1224–1234, 2015.

[33] L. Da Xu, W. He, and S. Li. Internet of Things in industries: A survey. *IEEE Transactions on Industrial Informatics*, 10(4):2233–2243, 2014.

[34] K. Dar, A. Taherkordi, R. Rouvoy, and F. Eliassen. Adaptable Service Composition for Very-large-scale Internet of Things Systems. In *Proceedings of Middleware Doctoral Symposium*, pages 11:1–11:2. ACM, 2011.

[35] M. Darianian and M. P. Michael. Smart home mobile RFID-based Internet-of-Things systems and services. In *Proceedings of International Conference on Advanced Computer Theory and Engineering*, pages 116–120. IEEE, 2008.

[36] W. Denniss and J. Bradley. OAuth 2.0 for Native Apps. RFC 8252, Internet Engineering Task Force (IETF), 2017.

[37] A. Dorri, S. S. Kanhere, and R. Jurdak. Blockchain in Internet of Things: Challenges and Solutions. arXiv: 1608.05187, arXiv.org, 2016.

[38] A. Dorri, M. Steger, S. S. Kanhere, and R. Jurdak. Blockchain: A distributed solution to automotive security and privacy. arXiv: 1704.00073, arXiv.org, 2017.

[39] A. Elliott and S. Knight. Role explosion: Acknowledging the problem. In *Software Engineering Research and Practice*, pages 349–355, 2010.

[40] E. Fernandes, J. Jung, and A. Prakash. Security analysis of emerging smart home applications. In *Proceedings of Symposium on Security and Privacy*, pages 636–654. IEEE, 2016.

[41] F. Fernández, Á. Alonso, L. Marco, and J. Salvachúa. A model to enable application-scoped access control as a service for IoT using OAuth 2.0. In *Proceedings of Conference on Innovations in Clouds, Internet and Networks*, pages 322–324. IEEE, 2017.

[42] D. Ferraiolo, R. Chandramouli, R. Kuhn, and V. Hu. Extensible Access Control Markup Language (XACML) and Next Generation Access Control (NGAC). In *Proceedings of International Workshop on Attribute Based Access Control*, pages 13–24. ACM, 2016.

[43] G. C. Fox, S. Kamburugamuve, and R. D. Hartman. Architecture and Measured Characteristics of a Cloud Based Internet of Things. In *Proceedings of International Conference on Collaboration Technologies and Systems*, pages 6–12. IEEE, 2012.

[44] P. Fremantle, B. Aziz, J. Kopecký, and P. Scott. Federated Identity and Access Management for the Internet of Things. In *Proceedings of International Workshop on Secure Internet of Things*, pages 10–17. IEEE, 2014.

[45] P. Garcia Lopez, A. Montresor, D. Epema, A. Datta, T. Higashino, A. Iamnitchi, M. Barcellos, P. Felber, and E. Riviere. Edge-centric Computing: Vision and Challenges. *SIGCOMM Comput. Commun. Rev.*, 45(5):37–42, 2015.

[46] O. Garcia-Morchon and K. Wehrle. Modular context-aware access control for medical sensor networks. In *Proceedings of Symposium on Access Control Models and Technologies*, pages 129–138. ACM, 2010.

[47] L. S. S. Gerdes, G. Selander, M. Mani, and S. Kumar. Use Cases for Authentication and Authorization in Constrained Environments. RFC 7744, Internet Engineering Task Force (IETF), 2014.

[48] G. S. Graham and P. J. Denning. Protection: principles and practice. In *Proceedings of Spring Joint Computer Conference*, pages 417–429. ACM, 1972.

[49] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29:1645–1660, 2013.

[50] J. A. Guerrero-ibanez, S. Zeadally, and J. Contreras-Castillo. Integration challenges of intelligent transportation systems with connected vehicle, cloud computing, and Internet of Things technologies. *IEEE Wireless Communications*, 22(6):122–128, 2015.

[51] Z. Guoping and G. Wentao. The Research of Access Control Based on UCON in the Internet of Things. *Journal of Software*, 6(4):724–731, 2011.

[52] S. Gusmeroli, S. Piccione, and D. Rotondi. A capability-based security approach to manage access control in the Internet of Things. *Mathematical and Computer Modelling*, 58(5):1189–1205, 2013.

[53] M. Hassanalieragh, A. Page, T. Soyata, G. Sharma, M. Aktas, G. Mateos, B. Kantarci, and S. Andreescu. Health monitoring and management using Internet-of-Things (IoT) sensing with cloud-

based processing: Opportunities and challenges. In *Proceedings of International Conference on Services Computing*, pages 285–292. IEEE, 2015.

[54] W. He, M. Golla, R. Padhi, J. Ofek, M. Dürmuth, E. Fernandes, and B. Ur. Rethinking Access Control and Authentication for the Home Internet of Things (IoT). In *Proceedings of USENIX Security Symposium*, pages 255–272. USENIX Association, 2018.

[55] J. L. Hernandez-Ramos, A. J. Jara, L. Marın, and A. F. Skarmeta. Distributed capability-based access control for the Internet of Things. *Journal of Internet Services and Information Security*, 3(3/4):1–16, 2013.

[56] V. C. Hu, D. Ferraiolo, R. Kuhn, A. Schnitzer, K. Sandlin, R. Miller, and K. Scarfone. Guide to Attribute Based Access Control (ABAC) Definition and Considerations. NIST Special Publication 800, NIST, 2014.

[57] D. Hussein, E. Bertin, and V. Frey. A Community-Driven Access Control Approach in Distributed IoT Environments. *IEEE Communications Magazine*, 55(3):146–153, 2017.

[58] S. M. R. Islam, M. Hossain, R. Hasan, and T. Q. Duong. A conceptual framework for an IoT-based health assistant and its authorization model. In *Proceedings of Annual Computing and Communication Workshop and Conference*, pages 616–621. IEEE, 2018.

[59] Y. J. Jia, Q. A. Chen, S. Wang, A. Rahmati, E. Fernandes, Z. M. Mao, and A. Prakash. ContexIoT: Towards Providing Contextual Integrity to Appified IoT Platforms. In *Proceedings of Network and Distributed System Security Symposium*, 2017.

[60] J. Jindou, Q. Xiaofeng, and C. Cheng. Access Control Method for Web of Things Based on Role and SNS. In *Proceedings of International Conference on Computer and Information Technology*, pages 316–321. IEEE, 2012.

[61] A. A. E. Kalam, R. E. Baida, P. Balbiani, S. Benferhat, F. Cuppens, Y. Deswarte, A. Miege, C. Saurel, and G. Trouessin. Organization based access control. In *Proceedings of International Workshop on Policies for Distributed Systems and Networks*, pages 120–131. IEEE, 2003.

[62] S. D. T. Kelly, N. K. Suryadevara, and S. C. Mukhopadhyay. Towards the implementation of IoT for environmental condition monitoring in homes. *IEEE Sensors Journal*, 13(10):3846–3853, 2013.

[63] R. Khan, S. U. Khan, R. Zaheer, and S. Khan. Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges. In *Proceedings of International Conference on Frontiers of Information Technology*, pages 257–260. IEEE, 2012.

[64] J. E. Kim, G. Boulos, J. Yackovich, T. Barth, C. Beckel, and D. Mosse. Seamless integration of heterogeneous devices and access control in smart homes. In *Proceedings of International Conference on Intelligent Environments*, pages 206–213. IEEE, 2012.

[65] T. H.-J. Kim, L. Bauer, J. Newsome, A. Perrig, and J. Walker. Challenges in Access Right Assignment for Secure Home Networks. In *Proceedings of USENIX Conference on Hot Topics in Security*, pages 1–6. USENIX Association, 2010.

[66] T. H. J. Kim, L. Bauer, J. Newsome, A. Perrig, and J. Walker. Access right assignment mechanisms for secure home networks. *Journal of Communications and Networks*, 13(2):175–186, 2011.

[67] L. LaPadula, D. E. Bell, and L. J. LaPadula. Secure computer systems: Mathematical foundations. *Draft MTR, The MITRE Corporation*, 2, 1973.

[68] S. Lee, J. Choi, J. Kim, B. Cho, S. Lee, H. Kim, and J. Kim. FACT: Functionality-centric Access Control System for IoT Programming Frameworks. In *Proceedings of Symposium on Access Control Models and Technologies*, pages 43–54. ACM, 2017.

[69] P. N. Mahalle, B. Anggorojati, N. R. Prasad, and R. Prasad. Identity Authentication and Capability Based Access Control (IACAC) for the Internet of Things. *Journal of Cyber Security and Mobility*, 1(4):309–348, 2013.

[70] P. N. Mahalle, P. A. Thakre, N. R. Prasad, and R. Prasad. A fuzzy approach to trust based access control in internet of things. In *Proceedings of International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems*, pages 1–5. IEEE, 2013.

[71] R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkernan. Internet of Things (IoT) Security: Current Status, Challenges and Prospective Measures . In *Proceedings of International Conference for Internet Technology and Secured Transactions*, pages 336–341. IEEE, 2015.

[72] M. L. Mazurek, J. P. Arsenault, J. Bresee, N. Gupta, I. Ion, C. Johns, D. Lee, Y. Liang, J. Olsen, B. Salmon, R. Shay, K. Vaniea, L. Bauer, L. F. Cranor, G. R. Ganger, and M. K. Reiter. Access control for home data sharing: Attitudes, needs and practices. In *Proceedings of SIGCHI Conference on Human Factors in Computing Systems*, pages 645–654. ACM, 2010.

[73] P. Mell and T. Grance. The NIST Definition of Cloud Computing. SP 800-145, NIST, 2011.

[74] D. Miessler. Preparing to Release the OWASP IoT Top 10 2018. https://goo.gl/kyeXkf. Accessed:2018-11-15.

[75] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac. Internet of Things: Vision, applications and research challenges. *Ad Hoc Networks*, 10(7):1497–1516, 2012.

[76] J. Mohammed, C.-H. Lung, A. Ocneanu, A. Thakral, C. Jones, and A. Adler. Internet of Things: Remote patient monitoring using web services and cloud computing. In *Proceedings of IEEE International Conference on Internet of Things, IEEE Green Computing and Communications, and IEEE Cyber, Physical and Social Computing*, pages 256–263. IEEE, 2014.

[77] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler. Transmission of IPv6 Packets over IEEE 802.15.4 Networks. RFC 4944, Network Working Group, 2007.

[78] R. Neisse, G. Steri, and G. Baldini. Enforcement of security policy rules for the internet of things. In *Proceedings of International Conference on Wireless and Mobile Computing, Networking and Communications*, pages 165–172. IEEE, 2014.

[79] K. T. Nguyen, M. Laurent, and N. Oualha. Survey on secure communication protocols for the Internet of Things. *Ad Hoc Networks*, 32:17–31, 2015.

[80] A. Ouaddah, A. A. Elkalam, and A. A. Ouahman. FairAccess: a new Blockchain-based access control framework for the Internet of Things. *Security and Communication Networks*, 9(18):5943–5964, 2016.

[81] A. Ouaddah, A. A. Elkalam, and A. A. Ouahman. Towards a novel privacy-preserving access control model based on blockchain technology in IoT. In *Europe and MENA Cooperation Advances in Information and Communication Technologies*, Advances in Intelligent Systems and Computing 520, pages 523–533. Springer, 2017.

[82] A. Ouaddah, H. Mousannif, A. A. Elkalam, and A. A. Ouahman.

Access control in the Internet of Things: Big challenges and new opportunities. *Computer Networks*, 112:237–262, 2017.

[83] J. Park and R. Sandhu. The UCON$_{ABC}$ Usage Control Model. *ACM Trans. Inf. Syst. Secur.*, 7(1):128–174, 2004.

[84] P. Patel, D. Bansal, L. Yuan, A. Murthy, A. Greenberg, D. A. Maltz, R. Kern, H. Kumar, M. Zikos, H. Wu, et al. Ananta: Cloud scale load balancing. *SIGCOMM Computer Communication Review*, 43(4):207–218, 2013.

[85] O. J. A. Pinno, A. R. A. Gregio, and L. C. E. D. Bona. ControlChain: Blockchain as a Central Enabler for Access Control Authorizations in the IoT. In *Proceedings of Global Communications Conference*, pages 1–6. IEEE, 2017.

[86] J. Postel. User datagram protocol. RFC 768, 1980.

[87] I. Ray, B. Alangot, S. Nair, and K. Achuthan. Using Attribute-Based Access Control for Remote Healthcare Monitoring. In *Proceedings of International Conference on Software Defined Systems*, pages 137–142. IEEE, 2017.

[88] M. A. Razzaque, M. Milojevic-Jevric, A. Palade, and S. Clarke. Middleware for Internet of Things: a survey. *IEEE Internet of Things Journal*, 3(1):70–95, 2016.

[89] D. Rivera, L. Cruz-Piris, G. Lopez-Civera, E. de la Hoz, and I. Marsa-Maestre. Applying an Unified Access Control for IoT-based Intelligent Agent Systems. In *Service-Oriented Computing and Applications (SOCA), 2015 IEEE 8th International Conference on*, pages 247–251. IEEE, 2015.

[90] R. Roman, J. Zhou, and J. Lopez. On the Features and Challenges of Security and Privacy in Distributed Internet of Things. *Computer Networks*, 57(10):2266–2279, 2013.

[91] A.-R. Sadeghi, C. Wachsmann, and M. Waidner. Security and privacy challenges in industrial Internet of Things. In *Proceedings of Design Automation Conference*, pages 1–6. IEEE, 2015.

[92] P. Saint-Andre. Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence. RFC 3921, Network Working Group, 2014.

[93] S. Salonikias, I. Mavridis, and D. Gritzalis. Access control issues in utilizing fog computing for transport infrastructure. In *Critical Information Infrastructures Security*, LNCS 9578, pages 15–26. Springer, 2015.

[94] P. Samarati and S. Capitani de Vimercati. Access control: Policies, models, and mechanisms. In *Foundations of Security*

Analysis and Design, LNCS 2171, pages 137–196. Springer, 2000.

[95] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based access control models. *Computer*, 29(2):38–47, 1996.

[96] R. Schuster, V. Shmatikov, and E. Tromer. Situational access control in the internet of things. In *Proceedings of Conference on Computer and Communications Security*, pages 1056–1073. ACM, 2018.

[97] S. Sciancalepore, G. Piro, P. Tedeschi, G. Boggia, and G. Bianchi. Multi-domain access rights composition in federated IoT platforms. In *Proceedings of Workshop on Recent Advances in Secure Management of Data and Resources in the IoT*, 2018.

[98] L. Seitz, G. Selander, and C. Gehrmann. Authorization Framework for the Internet-of-Things. In *Proceedings of International Symposium on A World of Wireless, Mobile and Multimedia Networks*, pages 1–6. IEEE, 2013.

[99] P. Sethi and S. R. Sarangi. Internet of Things: Architectures, Protocols, and Applications. *Journal of Electrical and Computer Engineering*, 2017, 2017.

[100] Z. Shelby, K. Hartke, and C. Bormann. The Constrained Application Protocol (CoAP). RFC 7252, Internet Engineering Task Force (IETF), 2014.

[101] Z. Sheng, S. Yang, Y. Yu, A. Vasilakos, J. Mccann, and K. Leung. A survey on the IETF protocol suite for the Internet of Things: standards, challenges, and opportunities. *IEEE Wireless Communications*, 20(6):91–98, 2013.

[102] F. Shrouf, J. Ordieres, and G. Miragliotta. Smart factories in Industry 4.0: A review of the concept and of energy management approached in production based on the Internet of Things paradigm. In *Proceedings of International Conference on Industrial Engineering and Engineering Management*, pages 697–701. IEEE, 2014.

[103] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini. Security, privacy and trust in Internet of Things: The road ahead. *Computer Networks*, 76:146–164, 2015.

[104] T. Socolofsky and C. Kale. A TCP/IP Tutorial. RFC 1180, Network Working Group, 1991.

[105] Z. Song, A. A. Cárdenas, and R. Masuoka. Semantic middleware for the Internet of Things. In *Proceedings of International Conference on the Internet of Things*, pages 1–8. IEEE, 2010.

[106] M. Stanislav and T. Beardsley. HACKING IoT: A Case Study on Baby Monitor Exposures and Vulnerabilities. Technical report, Rapid7, 2015.

[107] I. Stojmenovic and S. Wen. The fog computing paradigm: Scenarios and security issues. In *Proceedings of Federated Conference on Computer Science and Information Systems*, pages 1–8. IEEE, 2014.

[108] Y. Tian, N. Zhang, Y.-H. Lin, X. Wang, B. Ur, X. Guo, and P. Tague. SmartAuth: User-Centered Authorization for the Internet of Things. In *Proceedings of USENIX Security Symposium*, pages 361–378, 2017.

[109] R. Tönjes, P. Barnaghi, M. Ali, A. Mileo, M. Hauswirth, F. Ganz, S. Ganea, B. Kjærgaard, D. Kuemper, S. Nechifor, D. Puiu, A. Sheth, V. Tsiatsis, and L. Vestergaard. Real Time IoT Stream Processing and Large-scale Data Analytics for Smart City Applications. In *Poster presented at European Conference on Networks and Communications*, 2014.

[110] D. Trivellato, N. Zannone, M. Glaundrup, J. Skowronek, and S. Etalle. A semantic security framework for systems of systems. *International Journal of Cooperative Information Systems*, 22(1), 2013.

[111] R. van der Meulen. Gartner says 6.4 billion connected "things" will be in use in 2016, up 30 percent from 2015. Press release, Gartner, 2015.

[112] E. Vasilomanolakis, J. Daubert, M. Luthra, V. Gazis, A. Wiesmaier, and P. Kikiras. On the Security and Privacy of Internet of Things Architectures and Systems. In *Proceedings of International Workshop on Secure Internet of Things*, pages 49–57. IEEE, 2015.

[113] O. Vermesan, P. Friess, P. Guillemin, S. Gusmeroli, H. Sundmaeker, A. Bassi, I. S. Jubert, M. Mazura, M. Harrison, M. Eisenhauer, and P. Doody. Internet of Things Strategic Research Roadmap. In *Internet of Things – Global Technological and Societal Trends*, pages 9–51. River Publishers, 2011.

[114] R. H. Weber. Internet of things–new security and privacy challenges. *Computer Law & Security Review*, 26(1):23–30, 2010.

[115] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. RFC

6550, Internet Engineering Task Force (IETF), 2012.

[116] M. Wu, T.-J. Lu, F.-Y. Ling, J. Sun, and H.-Y. Du. Research on the architecture of Internet of Things. In *Proceedings of International Conference on Advanced Computer Theory and Engineering*, volume 5, pages 484–487. IEEE, 2010.

[117] L. Yang, S.-H. Yang, and L. Plotnick. How the internet of things technology enhances emergency response operations. *Technological Forecasting and Social Change*, 80(9):1854–1867, 2013.

[118] N. Ye, Y. Zhu, R.-c. Wang, R. Malekian, and L. Qiao-min. An Efficient Authentication and Access Control Scheme for Perception Layer of Internet of Things. *Appl. Math*, 8(4):1617–1624, 2014.

[119] E. Yuan and J. Tong. Attributed Based Access Control (ABAC) for Web Services. In *Proceedings of International Conference on Web Services*, pages 561–569. IEEE, 2005.

[120] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi. Internet of things for smart cities. *IEEE Internet of Things journal*, 1(1):22–32, 2014.

[121] G. Zhang and J. Tian. An extended role based access control model for the Internet of Things. In *Proceedings of International Conference on Information, Networking and Automation*, volume 1, pages 319–323. IEEE, 2010.

[122] Y. Zhang and X. Wu. Access Control in Internet of Things: A Survey. *IEICE Transactions on Communications*, 2016.