Fulfilling the IoT Vision: Are We There Yet?

Daniel Del Gaudio and Pascal Hirmer
Institute for Parallel and Distributed Systems, University of Stuttgart, Universitätsstraße 38, Stuttgart, Germany

Keywords: Internet of Things, Decentralized, Autonomous, Dynamic, Smart.

Abstract: The vision of the Internet of Things is enabling self-controlled and decentralized environments, in which hardware devices, equipped with sensors and actuators communicate with each other trough standardized internet protocols to reach common goals. The device-to-device communication should be decentralized and should not necessarily require human interaction. However, enabling such complex IoT applications, e.g., connected cars, is a big challenge, since many requirements need to be fulfilled. These requirements include, for example, security, privacy, timely data processing, uniform communication standards, or location-awareness. Based on an intensive literature review, in this overview paper, we define requirements for such environments and, in addition, we discuss whether they are fulfilled by state-of-the-art approaches or whether there still has to be work done in the future. We conclude this paper by illustrating research gaps that have to be filled in order to realize the IoT vision.

1 INTRODUCTION

In the Internet of Things (IoT), devices communicate with each other through standard internet protocols to reach common goals (Vermesan and Friess, 2013). These devices are usually attached with sensors and actuators to measure or alter the physical properties of the environment (Gubbi et al., 2013). For example, measuring a high humidity in a smart home could lead to opening a window.

Hence, the overall goal of the IoT is making people's lives easier and safer without requiring elaborate configuration or management tasks. Devices should seamlessly integrate themselves into everyday activities by autonomously collaborating (Stankovic, 2014; Brumitt et al., 2000).

However, in current IoT software solutions, the human user is still the core actor and has to make most of the decisions. For example, in smart home applications, turning on lights through a mobile application is already considered as an IoT application. In the IoT vision, however, devices should work together to make such decisions themselves. In security critical scenarios, a final decision of the human user could still be necessary, however, the decision making process should be supported by the devices. An example for an IoT application, which should work autonomously regarding decision making is autonomous driving, where cars (the devices) communicate with each other to reach a common goal, securely arriving at a destination. Figure 1 depicts a smart car that shares information about an upcoming obstacle directly with cars close by and via the cloud.

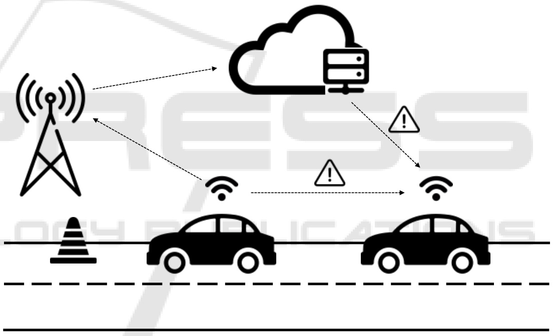

Figure 1: Connected cars sharing information about an upcoming obstacle.

To enable completely autonomous IoT environments, different requirements need to be fulfilled, including security, privacy, timely data processing, communication standards, or location-awareness. In this overview paper, we aim at defining these requirements based on an intensive literature review. The requirements are split into functional requirements FR_1 - FR_6 and non-functional requirements NFR_1 - NFR_8.

By doing so, we further investigate which requirements are already fulfilled and which requirements still need to be worked on in more depth in the future. Finally, we give a conclusion about research gaps that need to be filled in order to enable complex IoT scenarios, such as autonomous driving.

This paper is structured as follows: Section 2 contains related work. Section 3 describes functional and

Del Gaudio, D. and Hirmer, P.
Fulfilling the IoT Vision: Are We There Yet?.
DOI: 10.5220/0009439603670374
In *Proceedings of the 5th International Conference on Internet of Things, Big Data and Security (IoTBDS 2020)*, pages 367-374
ISBN: 978-989-758-426-8

non-functional requirements for IoT systems. Finally, Section 4 summarizes this paper and gives an outlook on future work.

2 RELATED WORK

This section describes related work to our paper.

Kyriazis and Varvarigou (Kyriazis and Varvarigou, 2013) describe challenges and enablers of smart, autonomous and reliable IoT environments. Furthermore, they propose an architecture for cross-platform IoT applications.

Gubbi et al. (Gubbi et al., 2013) define a vision of the IoT as a cloud-centric system to collect and analyze data from multiple sensors. In contrast, we consider the IoT as a highly distributed, decentralized, and autonomous system, where users and applications are the focus instead of data analysis.

The vision of the IoT stated by Miorandi et al. (Miorandi et al., 2012) is similar to ours. They see the IoT as a dynamic and distributed system of smart objects, which do not only collect data but also interact with the physical world. The vision described in this work is based on the one of Miorandi et al. Additionally to them, we evaluate our vision from a current state-of-the-art view. Furthermore, we define a set of functional and non-functional requirements that are, in our opinion, necessary to reach this vision.

3 REQUIREMENTS FOR AUTONOMOUS DECENTRALIZED INTERNET OF THINGS

In this section, we introduce requirements that need to be fulfilled in order to enable autonomous Internet of Things scenarios, such as autonomous driving. These requirements are based on an intensive literature review in the area of autonomous Internet of Things. We divided the requirements into functional and non-functional requirements.

3.1 Functional Requirements

We consider the following functional requirements: FR_1 Discovery, FR_2 Interoperability, FR_3 Portability, FR_4 Controllability, FR_5 Data Storage, and FR_6 Location-Awareness. Table 1 lists all functional requirements and their associated References. These requirements are described in detail in the following.

FR_1 – Discovery

An important requirement for decentralized, self-controlled Internet of Things is discovery. In order for devices to collaborate autonomously, they need to be aware of each other and the services they provide. When a new device enters the environment, other devices need to be notified about its existence to integrate it into the IoT environment. Furthermore, the device itself must be informed about the environment it is involved in. When a device leaves the environment or fails, other devices must also be informed about it to react accordingly, e.g., stop forwarding messages.

In centralized smart environments, devices register via a central component, similar to the discovery in service-oriented computing via UDDI (Weerawarana et al., 2005). However, this is not sufficient for the IoT vision, since this approach requires either setting up a discovery component for each environment, or each device being connected to an upper-lying network. This could lead to issues in terms of connectivity, and thus, flexibility, privacy, and scalability. Devices must be able to handle new or leaving devices themselves in a decentralized manner.

Datta et al. (Datta et al., 2015) categorize related work in the area of discovery into the following areas: distributed and peer-to-peer discovery services, centralized architectures, CoAP-based service discovery, semantic-based discovery, search engines for resource directory, and utilization of ONS and DNS.

In previous work, we introduce a life cycle method for device management in dynamic IoT environments that also includes informing new devices about already existing devices and vice versa (Del Gaudio et al., 2020).

Fredj et al. (Fredj et al., 2014) propose a semantic-based service discovery using ontologies. A semantic model that can be used to achieve discovery in such a manner is IoT Lite (Bermudez-Edo et al., 2016).

CoAP-based discovery mechanisms can make use of the resource discovery (*/.well-known/core*) interface of a CoAP server, through which provided services of the server can be retrieved (Shelby et al., 2014; Shelby et al., 2013). Cirani et al. (Cirani et al., 2014) propose an architecture for peer-to-peer-based autonomous resource and service discovery in the IoT. The architecture utilizes a central IoT gateway and the CoAP resource discovery interface.

In conclusion, the area of discovery, especially in the service domain is very well researched. For the IoT, many specialized approaches already exist. Hence, we can conclude that the discovery requirement can already be considered as fulfilled.

FR_2 – Interoperability

Table 1: Functional Requirements and associated references.

	Requirement Name	References
FR_1	Discovery	(Weerawarana et al., 2005) (Datta et al., 2015) (Fredj et al., 2014) (Bermudez-Edo et al., 2016) (Shelby et al., 2014) (Shelby et al., 2013) (Cirani et al., 2014) (Del Gaudio et al., 2020)
FR_2	Interoperability	(Akyildiz et al., 2002) (Shelby et al., 2014) (Hunkeler et al., 2008) (Meng et al., 2016)
FR_3	Portability and Software Provisioning	(Binz et al., 2012) (Franco da Silva et al., 2016) (Franco da Silva et al., 2017)
FR_4	Controllability of Flow	(Meyer et al., 2013) (Andrews et al., 2003) (White, 2004) (Seeger et al., 2018b) (Seeger et al., 2018a) (DelGaudio and Hirmer, 2019)
FR_5	Efficient Data Storage and Processing	(Madden et al., 2005)
FR_6	Location-Awareness/-Sensitivity	(Mahfouz et al., 2009) (Wang et al., 2013) (Witrisal et al., 2016)

IoT environments typically consist of numerous heterogeneous devices. Thus, we need communication standards that also constrained devices are capable of. In the IoT, standard internet protocols are used for communication. However, for constrained devices, adapted protocols and standards are required that are more lightweight, for example, using UDP for communication instead of TCP.

Interoperability can be divided into multiple layers (Akyildiz et al., 2002): Physical layer, data link layer, network layer, transport layer, and application layer. Protocols to enable communication between devices on the application layer have been developed in the past, e. g., the *Constrained Application Protocol* (CoAP) (Shelby et al., 2014) and the *Message Queuing Telemetry Transport* protocol (MQTT) (Hunkeler et al., 2008). MQTT adapts the publish-subscribe pattern (Eugster et al., 2003) and requires a central message broker.

In decentralized and self-controlled IoT environments, the message broker must be hosted by devices themselves, which contradicts to the asynchronous communication concept of the publish-subscribe pattern. CoAP is similar to HTTP but simpler and based on UDP. Hence, it is more appropriate for communication in IoT environments.

Another protocol that can, for example, be used for machine-to-machine communication in industrial IoT applications is *ZeroMQ* (ZMQ), as Meng et al. show in (Meng et al., 2016). ZMQ is specifically designed for machine-to-machine messaging communication in the IoT.

In conclusion, it can be stated that there are already many standards that exist for communication in IoT environments. Many of them are already established in research and industry (such as MQTT or CoAP). Hence, the requirement Interoperability can also be considered as fulfilled.

FR_3 – Portability and Software Provisioning

Since devices in heterogeneous IoT environments tend to have different physical and computational capabilities, portability is an important issue. More precisely, IoT devices communicating to reach a common goal need to support the same software solutions, i.e., the software needs to be portable amongst heterogeneous IoT devices. Therefore, software must be designed to be quickly deployable on different kinds of devices. Thus, an important requirement is portable software for IoT systems.

Since smart devices tend to be highly heterogeneous, portability of software is still a challenge. Low-level devices, such as the Arduino Boards or other micro controllers need to be flashed, so software usually cannot be deployed on them automatically. For higher-level devices, software deployment and management approaches, such as the TOSCA (Binz et al., 2012) standard can be utilized to model application stacks and dynamically deploy them on devices (Franco da Silva et al., 2016).

Even though portability is mostly provided by IoT devices and corresponding standards, automated provisioning of software in IoT environments is still being researched. There are first efforts to adapt the TOSCA standard to provision software into IoT environments (Franco da Silva et al., 2017), however, there still needs to be work done to enable reliable software provisioning in such complex environments.

FR_4 – Controllability of Flow

To control distributed heterogeneous devices in the IoT, the flow of data and execution must be controlled.

In centralized environments, a processing engine is typically used (Meyer et al., 2013). However, since requiring a central component is contradicting to the IoT vision, flow must be controlled by devices themselves. Thus, the data flow has to be controlled implicitly in the sense that every device knows its predecessor and successor.

Control flow and data flow are typically governed by a central instance using, e.g., BPEL (Andrews et al., 2003) or BPMN (White, 2004) engines. Since such central instances contradict with the vision of a decentralized and autonomous Internet of Things, we must seek for other solutions to control data and execution flow of IoT applications. For example, Seeger et al. propose a solution based on the choreography model (Seeger et al., 2018b; Seeger et al., 2018a). In previous work, we introduce a messaging engine, which is able to control the flow without any central component necessary (DelGaudio and Hirmer, 2019). Similarly, Bumgardner et al. introduce a decentralized IoT system, in which devices can exchange data. However, current state of the art approaches focusing on IoT are still mostly rudimentary and prototypical. Most established concepts originate from peer-to-peer systems, where file transfer is the usually the main goal. Hence, there are still missing approaches that are purely focused on the IoT.

FR_5 – Efficient Data Storage and Processing

To process data in smart environments in a decentralized manner, devices must be able to store data efficiently in time and place. Since smart devices tend to be much more vulnerable and mobile than larger machines or servers, data loss must be expected and handled by devices themselves. Traditional database systems are usually heavy-weight and come with many features, such as replication, scalability, or efficient data querying capabilities. However, the enhanced number of features and the consequently increasing size of these database systems makes them unsuitable for constrained IoT devices.

Hence, several database systems have been developed that are tailored for low-resource IoT devices. Those databases include, for example, in-memory databases, decentralized databases, key-value-stores, or lightweight document stores. Famous database systems for the IoT include TinyDB (Madden et al., 2005), SQLite, MongoDB Mobile, or HarperDB. Consequently, database developers have noticed the need for such lightweight systems that do only provide a subset of features in contrast to traditional database system, but are much more lightweight.

In conclusion, there are already many systems available that can even be deployed onto very low-resource wireless sensor networks, such as TinyDB. The requirement for efficient data storage on IoT devices can, therefore, be considered as fulfilled.

Regarding data processing, it is necessary to conduct data operations on the devices themselves instead of moving all the data to a central component. This is generally referred to as *edge or fog computing*. By doing so, latency can be reduced and, furthermore, the decentralized IoT system can be preserved. Consequently, there is a need for data processing techniques in edge cloud environments, especially focusing on stream processing. To realize this, lightweight stream and event processing systems have been developed, such as CEP Esper. However, there still needs to be work done regarding data processing on resource-limited devices. Oftentimes, only minor data operations can be conducted and data still needs to be transferred to a backend cloud environment for processing. This, however, contradicts with the idea of decentralization. Hence, there still is a need for more sophisticated concepts for data processing in IoT environments, enabling data processing as close to the data sources as possible.

FR_6 – Location-Awareness/-Sensitivity

In IoT environments, the location of each device plays an important role. For example, in connected car scenarios, localization needs to have a very high accuracy, especially if complete autonomy should be achieved. Furthermore, in Smart Factory or Smart Home scenarios, the location of sensors and actuators are crucial to develop functioning IoT applications. Especially moving devices, for example, smart phones or cars need to be locatable at all times.

There is a lot of research conducted regarding localization of IoT devices. The most common way for localization is GPS, which can nowadays provide a good accuracy. However, GPS has the limitation that it works only outside of buildings and, for example, not in a Smart Factory environments. Furthermore, for some scenarios (connected cars), the accuracy of GPS is not sufficient.

To cope with these issues, new localization approaches are currently being researched, for example, using Bluetooth Low Energy triangulation or UWB RTLS, achieving a high accuracy of ca. 8.5 centimeter derivation (Wang et al., 2013; Mahfouz et al., 2009). However, such accuracy can only be achieved in well-defined, closed environments.

The new communication standard 5G also provides localization features, which should be used for connected car solutions, since the accuracy is high and it can also be applied to scenarios outside and inside of buildings (Witrisal et al., 2016).

Currently, there is still a lot of research going on to increase the accuracy of localization approaches.

The experiences with 5G are still limited and need to be gathered in the future through the application in various IoT use cases.

In conclusion, in order to achieve completely self-controlled and decentralized IoT applications, there still needs to be work done regarding localization of IoT devices.

3.2 Non-functional Requirements

Additionally to the functional requirements, we evaluated the non-functional requirements: NFR_1 Privacy, NFR_2 Trust, NFR_3 Safety, NFR_4 Security, NFR_5 Reliability, NFR_6 Manageability, NFR_7 Adaptability, and NFR_8 Real-Time Capabilities. Table 2 lists all non-functional requirements and their associated references. These are specified in the following.

NFR_1 **– Privacy**

Regulations, such as the GDPR show that privacy becomes increasingly important to users (Tankard, 2016). However, the right of users to decide over their information is also a common one in the United States (Warren and Brandeis, 1890). Avoiding central components always leads to an increase in privacy, since no single instance has knowledge about the whole environment. Nevertheless, this does not ensure privacy for users in all cases sufficiently. Oftentimes privacy and quality of service need to be weighted against each other.

One way to improve privacy in decentralized IoT environments is to encrypt communication between devices. Many technologies to do this have been developed independently from the Internet of Things. Those are among others Virtual Private Networks, Transport Layer Security (TLS), DNS Security Extension, and Onion Routing (Weber, 2010). Since many IoT devices tend to have similar computing capabilities than typical computers, conventional privacy-ensuring standards can also be applied in the Internet of Things. Especially MQTT, which is based on the TCP transport protocol, can be encrypted with TLS. In regard to IoT-specific protocols like CoAP, efforts to encrypt communication is still in progress, i. e., Object Security of CoAP (OSCOAP) (Selander et al., 2017) for CoAP.

Since the goal of decentralized and autonomous IoT environments is that data must not be sent to central services hosted by third parties, privacy is ensured much more by design, because no single organization is able to collect all data from every user. Nonetheless, the problem gets partially shifted to a trust issue, because if two devices interact, there is also no central instance that controls who the respective other devices identity is.

NFR_2 **– Trust**

In decentralized environments, no controlling instance verifies transactions between parties. Thus, means are necessary to create trust between parties, i. e., the devices and their owners.

Yan et al. (Yan et al., 2014) identify the following objectives in regard to trust management in the IoT: Trust relationship and decision, data perception trust, privacy preservation, data fusion and mining trust, data transmission and communication trust, quality of IoT services, system security and robustness, generality, human-computer trust interaction, and identity trust. In their opinion, there are open issues especially in the areas trust evaluation, in terms of context awareness and user's subjective properties, comprehensive trust management frameworks, interoperation or integration of data transmission and communication trust technologies with other trust management mechanism, and human-computer trust interaction.

Chen et al. (Chen et al., 2011) describe a fuzzy trust model, where trust is evaluated by devices via observing its neighbor's packet forwarding behavior. The ratio of correctly forwarded packets to the total number of forwarded packets decides about how much a node can be trusted.

Sicari et al. (Sicari et al., 2015) state the following open issues in IoT trust management: a lack of a common trust negotiation language, object identity management, and trust negotiation mechanisms to handle data stream access control. This shows that there are still many open topics in terms of trust in the IoT that need further research.

NFR_3 **– Safety**

The more smart devices are integrated inside peoples lives, the more their lives rely on them. Especially in critical scenarios, such as autonomous driving, safety is an important requirement.

Currently, there are only a few approaches focusing on safety in IoT systems. For example, Celik et al. (Celik et al., 2018) introduce a system to conduct a static analysis on IoT application and check whether the application fulfills a set of predefined properties. Consequently, safety is one of the aspects that still needs further research in order to enable the IoT vision we aim for.

Table 2: Non-Functional Requirements and associated references.

	Requirement Name	References
NFR_1	Privacy	(Tankard, 2016) (Warren and Brandeis, 1890) (Weber, 2010) (Selander et al., 2017)
NFR_2	Trust	(Yan et al., 2014) (Chen et al., 2011) (Sicari et al., 2015)
NFR_3	Safety	(Celik et al., 2018)
NFR_4	Security	(Weber, 2010) (Stankovic, 2014) (Sicari et al., 2015)
NFR_5	Reliability	(Kyriazis and Varvarigou, 2013) (Qiu et al., 2016) (Newman and Watts, 1999)
NFR_6	Manageability	(Aceto et al., 2013) (Chiang and Zhang, 2016)
NFR_7	Adaptability	(Kyriazis and Varvarigou, 2013)
NFR_8	Real-Time Capabilities	(Yasumoto et al., 2016)

NFR_4 – Security

To ensure not only privacy, security, trust and safety but also integrity, IoT environments must be secure in terms of attacks (Weber, 2010; Stankovic, 2014).

Important requirements to achieve security are resilience to attacks, authentication, confidentiality, and access control (Weber, 2010; Sicari et al., 2015). Encryption standards like TLS and Datagram Transport Layer Security (DTLS) can be applied to secure authentication, increase confidentiality, and prevent specific kinds of attacks.

The existing security standards that are applied to the Internet, can also be transferred in IoT environments. However, oftentimes, these approaches are not applied in order to avoid the overhead. Finding a good trade-off between overhead due to security mechanisms and efficiency is still one of the great challenges.

NFR_5 – Reliability

Physical things tend to be volatile and smart devices cannot ensure the same availability than pure cyber-physical systems. The real world tends to be much more dynamic (Kyriazis and Varvarigou, 2013). Thus, we need means to achieve robustness and failure safety in smart environments. Devices must be able to handle failures, exceptions, and data loss themselves autonomously, to ensure safety for users.

In their survey paper Smart, autonomous and reliable Internet of Things, Kyriazis and Varvarigou (Kyriazis and Varvarigou, 2013) introduce the necessity of links between smart objects and their circumstances like administration domains, conditions, and events to derive the reliability of those objects. Furthermore, they recommend to use models and knowledge generation methods to evaluate the reliability of devices dependent on specific circumstances. Such methods can be used to raise consciousness when interacting with volatile smart objects. Nonetheless, we need means to increase their reliability, despite of their physical weaknesses.

Qiu et al. (Qiu et al., 2016) propose a solution to increase robustness in IoT networks by adding shortcuts between nodes, which is based on the small-world network model (Newman and Watts, 1999).

In conclusion, there are already many works focusing on reliability and fault tolerance in IoT environments, however, coping with the failure of devices is still an important issue which requires further work in the future.

NFR_6 – Manageability

In centralized environments, management tasks, like monitoring, introducing new devices or new software, and network management, is done via central instances. This is not sufficient for our previously defined vision of the IoT. Thus, new concepts regarding manageability are required. Although IoT environments tend to be highly distributed, monitoring still is a task that is centralized by nature. A technique of cloud monitoring to increase scalability that can be adopted to IoT environments are agents to perform data collection, filtering and aggregation directly on devices (Aceto et al., 2013). An issue when managing IoT applications is to keep software and credentials up to date, especially when devices are mobile and the internet connection is error-prone(Chiang and Zhang, 2016). Since many techniques of Cloud monitoring can be adopted to the IoT, there is no need for fundamental research in terms of IoT monitoring. Nonetheless, we need means to decentralize and automatize monitoring tasks.

NFR_7 – Adaptability

Situations in smart environments are not consistent by nature. Devices can be mobile or destroyed by physical interactions. Thus, devices need to be able to mutually react to changing situations. Smart environments need to be designed in such way that they are flexible in terms of changing situations.

Kyriazis and Varvarigou (Kyriazis and Varvarigou, 2013) state that devices should be able to exchange information about experienced situations to overcome them in the future. To achieve this, devices must have means to store and communicate context information. They should be able to autonomously learn how to adapt to changing situations in a preemptive manner.

Adaptability to new situations and considering concept drifts is still an open research issue, which needs to be considered in future approaches.

NFR_8 – Real-Time Capabilities

Devices must be able to mutually react to changing situations in a timely manner. Thus, devices need to be able to process data in real-time, since they must react to real-time changing situations. Relocating data processing from the cloud to the edge of the network enables low latency for data stream processing. Yamaguchi and Shigeno (Yasumoto et al., 2016) survey the state-of-the-art of real-time data stream processing in the IoT. They conclude that real-time capabilities, i.e., the guarantee to process data in a given time, cannot yet be achieved in real-world scenarios. Especially for autonomous cars, this however is crucial and requires new technologies, such as 5G networks.

4 CONCLUSION AND FUTURE WORK

We see the most research gaps in the non-functional requirements, especially NFR_1, NFR_2, NFR_3, NFR_5, and NFR_7. Privacy is probably one of the biggest challenges in terms of our vision, since IoT applications must balance between governmental regulations, user's preferences, and service quality. Also, trust is an important issue. We must evaluate whom users trust more: centralized systems or each other. We see much potential in blockchains or related systems to solve this. In terms of reliability, systems like smart cars are already more reliable than human beings. However, when systems become more distributed, handling failures is not yet investigated enough.

In the future, we aim to do more work in FR_4, NFR_5, and NFR_7, building on our previous work.

REFERENCES

Aceto, G., Botta, A., De Donato, W., and Pescapè, A. (2013). Cloud monitoring: A survey. *Computer Networks*, 57(9):2093–2115.

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). A survey on sensor networks. *IEEE communications magazine*, 40(8):102–114.

Andrews, T., Curbera, F., Dholakia, H., Goland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D., Thatte, S., et al. (2003). Business process execution language for web services.

Bermudez-Edo, M., Elsaleh, T., Barnaghi, P., and Taylor, K. (2016). Iot-lite: a lightweight semantic model for the internet of things. In *2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*, pages 90–97. IEEE.

Binz, T., Breiter, G., Leyman, F., and Spatzier, T. (2012). Portable cloud services using tosca. *IEEE Internet Computing*, 16(3):80–85.

Brumitt, B., Meyers, B., Krumm, J., Kern, A., and Shafer, S. (2000). Easyliving: Technologies for intelligent environments. In *International Symposium on Handheld and Ubiquitous Computing*, pages 12–29. Springer.

Celik, Z. B., McDaniel, P., and Tan, G. (2018). Soteria: Automated iot safety and security analysis. In *2018 {USENIX} Annual Technical Conference ({USENIX}{ATC} 18)*, pages 147–158.

Chen, D., Chang, G., Sun, D., Li, J., Jia, J., and Wang, X. (2011). Trm-iot: A trust management model based on fuzzy reputation for internet of things. *Comput. Sci. Inf. Syst.*, 8(4):1207–1228.

Chiang, M. and Zhang, T. (2016). Fog and iot: An overview of research opportunities. *IEEE Internet of Things Journal*, 3(6):854–864.

Cirani, S., Davoli, L., Ferrari, G., Léone, R., Medagliani, P., Picone, M., and Veltri, L. (2014). A scalable and self-configuring architecture for service discovery in the internet of things. *IEEE Internet of Things Journal*, 1(5):508–521.

Datta, S. K., Da Costa, R. P. F., and Bonnet, C. (2015). Resource discovery in internet of things: Current trends and future standardization aspects. In *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, pages 542–547. IEEE.

Del Gaudio, D., Reichel, M., and Hirmer, P. (2020). A life cycle method for device management in dynamic iot environments. In *Proceedings of the 5th International Conference on Internet of Things, Big Data and Security*.

DelGaudio, D. and Hirmer, P. (2019). A lightweight messaging engine for decentralized data processing in the internet of things. *SICS Software-Intensive Cyber-Physical Systems*.

Eugster, P. T., Felber, P. A., Guerraoui, R., and Kermarrec, A.-M. (2003). The many faces of publish/subscribe. *ACM computing surveys (CSUR)*, 35(2):114–131.

Franco da Silva, A. C., Breitenbücher, U., Hirmer, P., Képes, K., Kopp, O., Leymann, F., Mitschang, B., and Steinke, R. (2017). Internet of things out of the box:

Using tosca for automating the deployment of iot environments. In *CLOSER*, pages 330–339.

Franco da Silva, A. C., Breitenbücher, U., Képes, K., Kopp, O., and Leymann, F. (2016). Opentosca for iot: automating the deployment of iot applications based on the mosquitto message broker. In *Proceedings of the 6th International Conference on the Internet of Things*, pages 181–182. ACM.

Fredj, S. B., Boussard, M., Kofman, D., and Noirie, L. (2014). Efficient semantic-based iot service discovery mechanism for dynamic environments. In *2014 IEEE 25th Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC)*, pages 2088–2092. IEEE.

Gubbi, J., Buyya, R., Marusic, S., and Palaniswami, M. (2013). Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7):1645–1660.

Hunkeler, U., Truong, H. L., and Stanford-Clark, A. (2008). Mqtt-sa publish/subscribe protocol for wireless sensor networks. In *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE'08)*, pages 791–798. IEEE.

Kyriazis, D. and Varvarigou, T. (2013). Smart, autonomous and reliable internet of things. *Procedia Computer Science*, 21:442–448.

Madden, S. R., Franklin, M. J., Hellerstein, J. M., and Hong, W. (2005). Tinydb: an acquisitional query processing system for sensor networks. *ACM Transactions on database systems (TODS)*, 30(1):122–173.

Mahfouz, M. R., Fathy, A. E., Kuhn, M. J., and Wang, Y. (2009). Recent trends and advances in uwb positioning. In *2009 IEEE MTT-S International Microwave Workshop on Wireless Sensing, Local Positioning, and RFID*, pages 1–4. IEEE.

Meng, Z., Wu, Z., Muvianto, C., and Gray, J. (2016). A data-oriented m2m messaging mechanism for industrial iot applications. *IEEE Internet of Things Journal*, 4(1):236–246.

Meyer, S., Ruppen, A., and Magerkurth, C. (2013). Internet of things-aware process modeling: integrating iot devices as business process resources. In *International conference on advanced information systems engineering*, pages 84–98. Springer.

Miorandi, D., Sicari, S., De Pellegrini, F., and Chlamtac, I. (2012). Internet of things: Vision, applications and research challenges. *Ad hoc networks*, 10(7):1497–1516.

Newman, M. E. and Watts, D. J. (1999). Renormalization group analysis of the small-world network model. *Physics Letters A*, 263(4-6):341–346.

Qiu, T., Luo, D., Xia, F., Deonauth, N., Si, W., and Tolba, A. (2016). A greedy model with small world for improving the robustness of heterogeneous internet of things. *Computer Networks*, 101:127–143.

Seeger, J., Deshmukh, R. A., and Bröring, A. (2018a). Dynamic iot choreographies–managing discovery, distribution, failure and reconfiguration. *arXiv preprint arXiv:1803.03190*.

Seeger, J., Deshmukh, R. A., and Broring, A. (2018b). Running distributed and dynamic iot choreographies. In *2018 Global Internet of Things Summit (GIoTS)*, pages 1–6. IEEE.

Selander, G., Mattsson, J., Palombini, F., and Seitz, L. (2017). Object security of coap (oscoap). *Internet Engineering Task Force (IETF) Internet-Draft work in progress*.

Shelby, Z., Bormann, C., and Krco, S. (2013). Core resource directory.

Shelby, Z., Hartke, K., and Bormann, C. (2014). The constrained application protocol (coap). Technical report.

Sicari, S., Rizzardi, A., Grieco, L. A., and Coen-Porisini, A. (2015). Security, privacy and trust in internet of things: The road ahead. *Computer networks*, 76:146–164.

Stankovic, J. A. (2014). Research directions for the internet of things. *IEEE Internet of Things Journal*, 1(1):3–9.

Tankard, C. (2016). What the gdpr means for businesses. *Network Security*, 2016(6):5–8.

Vermesan, O. and Friess, P. (2013). *Internet of things: converging technologies for smart environments and integrated ecosystems*. River Publishers.

Wang, Y., Yang, X., Zhao, Y., Liu, Y., and Cuthbert, L. (2013). Bluetooth positioning using rssi and triangulation methods. In *2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)*, pages 837–842. IEEE.

Warren, S. D. and Brandeis, L. D. (1890). Right to privacy. *Harv. L. Rev.*, 4:193.

Weber, R. H. (2010). Internet of things–new security and privacy challenges. *Computer law & security review*, 26(1):23–30.

Weerawarana, S., Curbera, F., Leymann, F., Storey, T., and Ferguson, D. F. (2005). *Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging and More*. Prentice Hall PTR, Upper Saddle River, NJ, USA.

White, S. A. (2004). Introduction to bpmn. *Ibm Cooperation*, 2(0):0.

Witrisal, K., Meissner, P., Leitinger, E., Shen, Y., Gustafson, C., Tufvesson, F., Haneda, K., Dardari, D., Molisch, A. F., Conti, A., et al. (2016). High-accuracy localization for assisted living: 5g systems will turn multipath channels from foe to friend. *IEEE Signal Processing Magazine*, 33(2):59–70.

Yan, Z., Zhang, P., and Vasilakos, A. V. (2014). A survey on trust management for internet of things. *Journal of network and computer applications*, 42:120–134.

Yasumoto, K., Yamaguchi, H., and Shigeno, H. (2016). Survey of real-time processing technologies of iot data streams. *Journal of Information Processing*, 24(2):195–202.