Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 54-1 (2021) 1138–1143

Towards the Interoperability of IoT Platforms: A Case Study for Data Collection and Data Storage

Antti Martikkala*. Andrei Lobov**
Minna Lanz*. Iñigo Flores Ituarte*

**Tampere University, Tampere, Finland (e-mail: author@tuni.fi).*
***NTNU, Trondheim, Norway (e-mail: author@ntnu.no)*

Abstract: Interoperability is one of the biggest challenges in IoT. There is a need to connect tools and services from different platforms to improve performance and flexibility as well as to get rid of vendor-locks. Moreover, fully established practices to classify IoT systems to support the development of multi-platform systems have not been grounded. This paper addresses these issues on a two-fold approach. Firstly, by introducing a classification that is based on the tasks performed in IoT system. Secondly by proposing a conceptual middleware to connect different IoT platforms as for a possible solution for interoperability issues in IoT.

Keywords: Internet of Things, IoT, Interoperability, Middleware, Classification, Agile manufacturing, Factory and Industrial Automation

1. INTRODUCTION

It is beneficial for all IoT end-users to be free from vendor locks, in other words from binding themselves to a single IoT solution provider. Firstly, to ensure openness it is required that different IoT platforms and tools can be used flexibly together. Secondly, the switch from a one platform to another must be effortless. (Noura, Atiquzzaman and Gaedke, 2019)(Khanna and Kaur, 2020)

Aliprandi (2011) states that interoperability is the key to true openness and innovation. By definition: *interoperability is a characteristic of a product or system, whose interfaces are completely understood, to work with other products or systems, present or future, in either implementation or access, without any restrictions* (AFUL Interoperability Working Group, no date). Here, seamless connection and communication between computerized systems is emphasized as interoperability.

Why IoT systems are not open and interoperable? The biggest reason is said to be the cyber-physical nature that requires lots of specialization and, the reason that platforms are often developed from a product-centric, bottom up approach. Further, companies make their decisions based on the effects for their business. To sum it up, interoperability is a trade-off between (1) the profitability of the business case, (2) their strategic position and (3) privacy and security considerations, while fulfilling legal requirements. (Mosterd, 2019) The current trends show that IoT platforms will not necessarily come towards harmonization and standardization and therefore heighten the need for new solutions.

Evidently, the complexity of the platforms is a struggle in IoT (Vogel *et al.*, 2020), and missing standardization makes it difficult to flexibly integrate different systems. As stated in Farnell Global IoT Survey (2019) 35% of respondents said that the biggest factor to accelerate of the benefits of IoT would be interoperability. According to GAO (2020) survey 30 out of 74 federal agencies stated interoperability as the most significant challenge to adopting IoT technologies. It is clear that Interoperability of heterogeneous IoT networking is still a challenge that is caused by a large number of different technologies, architectures, applications, communication protocols and security mechanisms that are employed in IoT systems (Mohd Aman *et al.*, 2020).

The great number of protocols that are used in IoT are racing to be the preeminent IoT protocols. Some of the protocols are proprietary and some are open. It is predicted that the different protocols used in IoT systems will co-exist, as they have their own strengths and weaknesses. (Mosterd, 2019) As stated, interoperability is one of the big challenges in IoT. Especially, small and medium sized enterprises (SMEs) face difficulties to integrate intelligent IoT solutions to discrete production steps, small lot sizes, and large variety of machinery with highly heterogenous communication protocols (Mahmood *et al.*, 2018)(Driate, Minoufekr and Plapper, 2019)(Toivonen, Järvenpää and Lanz, 2017). These companies rarely have enough financial capabilities to have commercial systems fitted to their needs (Mahmood *et al.*, 2018)(Driate, Minoufekr and Plapper, 2019). Therefore, good performance with lower cost could be achieved by utilizing different platforms and their tools together. This kind of system design approach helps to avoid vendor locks as well as improves flexibility to exploit a larger variety of IoT services.

In this regard, this research has two objectives that assist IoT adopters towards IoT integration. *First*, we present a simple classification of IoT platforms to allow SMEs identify possible IoT applications to embed intelligence in their products or production systems. The answer to this objective is presented

10.1016/j.ifacol.2021.08.134

in section number 4. The results of this initial objective lead us to the following objective. *Second,* we present a solution for interoperability challenges between data collection services (i.e., sensors) and data storage services (i.e., storage platforms). In this direction, a middleware framework to connect both services is proposed and tested in a case study. The answer to this objective is presented in section 5.

2. METHODOLOGY

The methodology is in line with the objectives and include a two-step approach. *First*, existing classifications methods are investigated through literature review and used to compose an applicable IoT platform classification based on types of services and platforms. The literature review was performed by a grey literature review including online information. A grey literature review includes publicly available online information in the form of peer-reviewed academic literature, but also relevant information available on IoT online forums, business or industrial reports, which are not necessarily peer reviewed (Mahood, Van Eerd and Irvin, 2014). The methodological choices linked to the *second* objective are based on case study research to propose a solution to a specific challenge on interoperability between IoT data collection services (i.e., sensors) and data storage services (i.e., storage platforms). In this direction, a middleware framework to connect both services is proposed and tested in a case study. Conclusions and future work are outlined in section 6.

3. BACKGROUND TO RESEARCH

3.1. A Review of Terminology in IoT

The terminology related to IoT is somewhat diverse and therefore some terms used in the paper need to be explained. In this paper, *services* refer to the tasks that must be fulfilled to meet the specific requirements set for IoT system to be operational. (e.g., saving the sensor data) These services can be executed by tools provided by IoT platform.

In this regard, we use the term *platform* as a synonym for IoT platform. The platform is a host for services, which refers to all technology platforms in IoT that can be used to take care of one or several services in IoT system (e.g., device management, data collection, processing, and visualization). Another important term is *middleware*. Literature defines middleware as a software component that acts as an interface or service between applications (Craig and Serain, 2019)(Ibrahim, 2009). The purpose of middleware is to provide efficient and effective connection and integration function between applications. Thus, taking complexity of often dynamic communication channels with related addressing and search functions for its end nodes.

Research refer to IoT platforms as *"the middleware and the infrastructure that enables the end users to interact with smart objects"* (Mineraud *et al.*, 2016). In this context it is understood as an integration tool for sensors and data processing and analysis services. However, if we speak about integrating different platforms, a different level of middleware may need to be defined to provide communication and integration functions for platforms. Therefore, in this paper we use the term *middleware* to refer to a tool that can connect different platforms together. To conclude, the term *IoT solution* refers to an integrated bundle of IoT technologies that organizations or private persons can acquire to solve a specific problem with the objective to create new value.

3.2. Existing Classification for IoT Systems

There are many ways to classify IoT platforms in the literature. Firstly, platforms could be classified to three types such as (1) device-to-device, (2) cloud-based and (3) device-to-cloud platforms. Also, platforms can be classified according to their applicability (Vogel *et al.*, 2020).

Vogel et al. (2020) characterizes IoT platforms according to openness and concludes that there are different openness types such as, open standards, open APIs and open source. Different stakeholders look for different aspects in openness. For example, application providers see their platforms as open if it provides open APIs or open standards. For system integrators open standards are more relevant from these two. For operators, openness does not play a big role, they are happy as long as the platforms is operable. (Vogel *et al.*, 2020) However, the use of open platforms can promote interoperability and also reduce vendor locking (Ramirez Lopez, Aponte and Garcia, 2019)(Aghenta and Iqbal, 2019).

Mineraud et al. (2016) evaluated IoT platforms according to their capability to (i) support the integration of heterogeneous hardware, (ii) provide sufficient data management mechanisms, (iii) support application developers, (iv) support the formation of ecosystems, as well as (v) provide the dedicated marketplaces for the IoT.

3.3. Existing Tools to Connect IoT Platforms

Ideally IoT systems would communicate with same language, (i.e., protocols and practices). This is often the case when relying on a single vendor and platform provider. However, with multi-platform IoT systems the interoperability issues are almost always present.

We also conclude that the simpler the platforms are, the easier it is to build interoperable systems as visualized in Figure 1. New functionalities usually create complexity, for example automatic sensor provisioning can cause multiple interoperability issues throughout the system. The platforms that require more manual work to implement new sensors and functions are usually more open and flexible. This can lead to a situation where there is a trade-off between functionality and openness when implementing IoT systems.

Fig 1. Openness vs. complexity

One apparent situation where interoperability occurs is when data collection systems are connected to cloud spaces. Therefore, this paper concentrates on this issue. As stated, the direct communication from data collection systems to data storage is not usually possible. The endpoint for data storage is not compatible with data coming directly from the sensors. Often there are syntactical and semantic interoperability between these systems (Noura, Atiquzzaman and Gaedke, 2019). Therefore, there must be a system between to take care of tasks as data decoding and taking care of attaching required metadata, etc.

As the interoperability is a recognized problem there are already existing solutions to integrate platforms. Node-RED is a flow-based visual programming tool that helps to integrate multiple platforms, API's, services, etc. Node-Red is a flow-based visual programming tool on the web that can be used to create JavaScript functions. The platform has built-in nodes that can be used for different actions. If a specific node does not exist, the user can create it and share it with other users. The Node-RED tool can be run on different hardware and cloud platforms as Raspberry Pi, Nokia Innovation Platform, Siemens, MindSpehere, etc.(*Node-RED*, no date)

IFTTT (If This Then That) works on events similarly as Node-RED. An event triggers the application I.e. A piece of JavaScript code is run in response, which can be subsequently used to automate web-application tasks. In the context of IoT, it can be used in home automation, for example turning the fan on when associated temperature sensor measures a value higher than the set threshold value (IFTTT, no date).

EMQ X is an open source IoT MQTT message broker that can host large-scale MQTT client connections. It supports a large variety of IoT protocols, including MQTT, MQTT-SN, CoAP, LwM2M, and other TCP/UDP based proprietary protocols. These can be then run-on resource constrained IoT devices and/or Industrial servers. (*EMQ X Broker*, no date)

Another approach for interoperability is to use special architectures or frameworks that can be used as foundations for interoperable IoT systems. These systems can be extended by implementing existing modules and applications, or by writing use-case specific applications that exploit the defined core structures. LAURA is service-oriented and general open-source conceptual architecture, designed to support the deployment of IoT applications. The concept simplifies the development of applications reducing the need for specialized IoT low-level knowledge providing programmer friendly interfaces. The pre-defined architecture and communication channels between modules and layers of LAURA facilitates the development of IoT applications that integrate different IoT platforms. (Teixeira *et al.*, 2020) Similarly, Arrowhead Framework provides means for Service-oriented architecture via System-of-Systems approach, where optional application systems consume services provided by so-called core systems that provide means for service discovery, service registration and service authorization. (Hoikka, 2019) Mainflux also consists of installable core platforms components and optional ones. The core components are taking care of receiving and handling of the data and can be extended with optional components as database writers, databases, visualizations, etc. (Mainflux, no date)

4. IOT SYSTEM CLASSIFICATION FOR SMEs

4.1. Structuring IoT Systems

In our vision, the classification should be based on the idea of multi-platform IoT as the use of services from different platforms providers is essential part in flexibility. Similarly, Vogel *et al.* (2020) explain that from system integrators perspective flexibility requires free use of different services and a variety of tools to offer suitability for as many uses cases as possible.

According to Taivalsaari et al. (2018) IoT system consists of

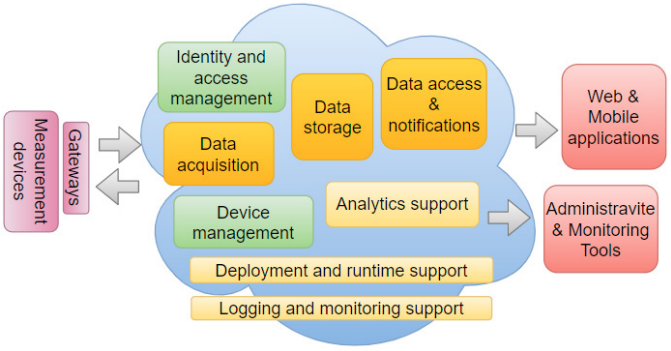

Fig. 2. Overview of an IoT architecture, adapted from *(Taivalsaari and Mikkonen, 2018)*.

four distinct architectural elements: devices, gateways, cloud and apps. Figure 2 deepens this by including instances inside the cloud. Nevertheless, this abstraction can be used to refine the framework for our classification further. Moreover, the architectures in IoT systems are not homogenous and usually systems cannot be defined as simple bottom to up structures with separate layers after another. Also, there are many other parallel functions as identity and access management, analytics support, deployment and runtime support etc. as shown in Figure 2.

The complexity and heterogeneity of IoT systems is evident. For example, data collection from digital models as digital twins is also possible. Also, edge computing and decentralized computing are good examples of diversity in the technology. When considering the previous reflections, we come to the conclusion that our classification into components of IoT system bases on the question: *what are the different tasks in the system from bottom to up?*

4.2. Proposed IoT Classification Based on Services

In our definition IoT systems can be said to consist of four main parts. In this paper we call these parts as services. Some simplification is done to make the concept easier to comprehend.

In our definition the four main services are: (1) *Data collection system*, (2) *Data storage*, (3) *Data analysis and* (4) *Dashboard (control and visualization).*

Moreover, there are also tasks in IoT systems that can be said to be more under the hood processes. These include device management, security, etc. For example, security is a system property that exists in several layers in the IoT stack from sensors to dashboards (Serpanos and Wolf, 2017).

As said a simple classification for IoT platforms is based on using the previously defined services in IoT systems as all platforms do not offer all the services that were presented. Therefore, IoT platforms can be divided into 9 types. Which include platforms that link to a single specific service (e.g., data collection system, data storage, data analysis, and dashboards), and platforms that interlink between services. In relation to platforms linked to a single specific service:

Type (A): Data collection and sensor management platforms take care of sensor systems and are usually attributed to specific communication technologies and IoT networks.

Type (B): Data storage platforms are platforms that offer tools for storing and querying the data as well as security.

Type (C): Visualization and control platforms (dashboards) provide tools for operators to visualize the data and control the system. There is some variation found in these platforms. Many of them are only made for visualization and cannot be used to control the devices. In that case separate tools for device management are needed.

Type (D): Data analysis platforms are in many cases just platforms capable of running an artificial intelligence (AI) application or mathematical models. Also, some easy-to-use data analysis tools exist.

Additionally, IoT platform providers interlink different services. For example, Type (BC) platforms are probably the most common types of multi service IoT platforms. These platforms include combined Data Storage and Dashboards.

Type (DC) platforms combine two services as Data Analysis and Dashboards. Type (BD) platforms include tools to handle the data and the results should be shown to the user by using a separate platform.

Type (BCD) platforms are almost complete IoT systems and have all the tools required to handle and present the data. These platforms can also offer a complete package for IoT from sensors to data handling and monitoring. Type (ABCD) platforms are complete packages. These systems have more reason to be offered as closed systems that does not provide much flexibility to be used with other tools.

To make a completely closed system a viable choice for a customer, the company must be big enough to be trusted to have continuity as well as to have all the required abilities for IoT. These systems could be product-specified systems, for example power plant monitoring or restroom towel dispenser monitoring solutions. The created classification is kind of a theoretical abstraction of the real world. Moreover, it can be used to clarify the connection points between different systems as well to substantiate the interoperability issues. As shown, different IoT platforms can provide different services. Therefore, a need to integrate these services becomes evident.

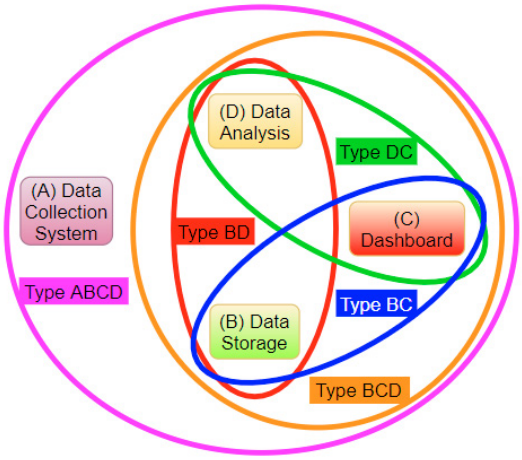

Fig 3. IoT platform classification based on services

5. INTEROPERABILITY IN IOT

5.1. Conceptual Middleware Framework

The concept of the middleware for alphanumeric sensor data is based on the idea of a modular adapter between data collection systems and IoT platforms. I.e. to improve flexibility with a platform that uses modules that can be installed to make it interoperable with different systems. The similar idea lies behind, software modules, USB-drivers, etc. Furthermore, the same concept could be used between different IoT tools/services that were presented earlier in the paper in Figure 3.

The middleware for data collection systems and data storages consists of four main blocks: (1) Endpoint, (2) Validator, (3) Parsers, and (4) Router as shown in figure 4. The endpoint is a connection point for data collection systems to forward their data. Usually, the endpoint is realized via an HTTP web API. To ensure the interoperability with different systems the endpoint is required to support as many protocols as possible.

After the data is received by the endpoint it is made available for the next modules in middleware. The second block in the middleware is the validator. The purpose of the validator is to recognize the type of the data and decide which parser to use.

The parser parses the sensor data from the message into a format supported by the receiving end. The sensor manager handles a database of the schemas, watch dog settings, sensors and their metadata, etc. This metadata includes sensor ID's that are linked with other information as location, type, accuracy, measuring frequency and so forth.

As said, to be able to parse the data from different data schemas and to forward it to different platforms we are presenting a concept of a modular architecture. These modules can be written by the operators or system integrators.

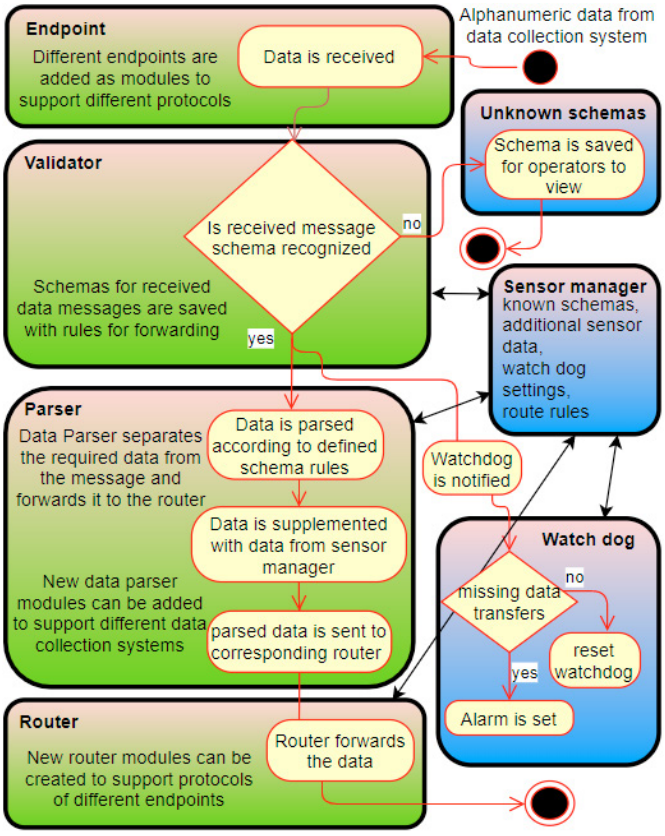

Fig. 4. Conceptual middleware framework

Also, a library or a "module store" could be used to share the modules with other users. Further, to improve the usability, different Artificial Intelligence (AI) methods could be used parse the required information from the data packet to understand its contents.

We suggest that, this kind of middleware is easy to use as the modules include all the required tools for connecting the supported platforms. On the other hand, it is not as flexible as systems that are based on data flow editing tools with visual or textual programming languages.

As implied, the middleware becomes useless or considerably less user-friendly if there are no existing modules that support the used platforms. To facilitate a more extensive support, the middleware should be constructed so that the advanced users could write new modules somewhat effortlessly and share those with other users.

5.2. Case Study for Middleware on Google Functions

Google Cloud Functions (GCF) is a scalable pay-as-you-go FaaS platform that can run code with no server management required. This serverless execution environment can be used for simple functions that are used with events emitted from cloud infrastructure and services. This means that the code is only executed when someone makes a request to the server. A request to server triggers the serverless container and creates the backend and responds to the data. The serverless container automatically scales up if the traffic increases and also scales down if traffic decreases.

Google Cloud Functions appears to be an easy starting point for prototyping purposes for the suggested middleware/adapter to integrate different IoT platforms and tools. The platform creates an endpoint that can receive messages and trigger the middleware application.

ThingPark is a commercial solution that allows to deploy and manage IoT devices that are using LoRaWAN or Cellular communications. Finnish transmission network operator Digita is adapting ThingPark in their nationwide LoRaWAN network. The sensors are managed in Digita ThingPark web portal where device provisioning i.e. new device profiles and routing can be set. The messages are routed using HTTP REST where JSON and XML data formats are supported. Google Firestore is a automatically scaling NoSQL document database that supports REST API.

ThingPark and Firestore are not interoperable and therefore a middleware is needed. In this case study ThingPark platform forwards data through REST API POST messages in JSON format to the GCF endpoint HTTP address. The received message triggers the middleware backend application. The code run on the middleware parses the sensor data from the JSON and saves it to Firestore database. The architecture for the middleware is presented in figure 4. The parts with blue background were not carried out in the implementation and are still on a conceptual level.

The middleware created for this case study provides a solution for connecting only two platforms together. The middleware is a proof of concept that was created to demonstrate the design concept that will be developed further to support a larger variety of IoT platforms that are not interoperable. The idea is to extend it into a modular middleware that can be updated to support new platforms by installing modules.

6. CONCLUSIONS

Fully established practices have not been grounded and the IoT splitting into clear parts is still somewhat challenging. This paper addressed these issues by proposing a classification for IoT platforms and a conceptual framework for extendable IoT platforms. The classification made in the paper is abstract owing to the heterogeneity of platforms and underlying processes in the systems. Howbeit, it helps to clarify the connection points between different systems, and it could also support the implementation of multi-platform IoT systems.

The existing problem of interoperability is not going to be easily solved by standardization of IoT. The wide operating field, fragmentation and new emerging uses cases will keep IoT in everchanging state. Therefore, this paper suggested that a flexible middleware could be the solution. The main idea is to create a middleware that adopts modules that can be dynamically installed to make it interoperable with different platforms. The similar idea lies behind, software modules, and USB-drivers.

The conceptual middleware was tested in a case study to connect Digita ThingPark platforms with Google Firestore database. The middleware works as intended and data was successfully saved in the database. For future research, the conceptual framework for connecting different IoT tools could be developed further and be used in future multi-platform IoT implementations to test its practicality.

ACKNOWLEDGEMENTS

This research has received support from European commission structural fund project IoT Compass Hub.

REFERENCES

AFUL Interoperability Working Group (no date) *Definition: Interoperability*. Available at: http://interoperability-definition.info/en/ (Accessed: 11 March 2021).

Aghenta, L. O. and Iqbal, M. T. (2019) 'Low-Cost, Open Source IoT-Based SCADA System Design Using Thinger.IO and ESP32 Thing', *Electronics*, 8(8), p. 822. doi: 10.3390/electronics8080822.

Aliprandi, S. (2011) 'Interoperability and open standards: the key to a real openness', *International Free and Open Source Software Law Review*, 3(1), pp. 5–24. doi: 10.5033/ifosslr.v3i1.53.

Craig, I. and Serain, D. (2019) *Middleware*. Springer. Available at: https://andor.tuni.fi/permalink/358FIN_TAMPO/176jdvt/cdi_askewsholts_vlebooks_9781447133872.

Driate, A., Minoufekr, M. and Plapper, P. (2019) 'Knowledge management for manufacturing SMEs using industrial IoT', in *Proceedings - 15th Annual International Conference on Distributed Computing in Sensor Systems, DCOSS 2019*. Institute of Electrical and Electronics Engineers Inc., pp. 355–361. doi: 10.1109/DCOSS.2019.00077.

EMQ X Broker (no date). Available at: https://docs.emqx.io/ (Accessed: 16 December 2020).

Farnell Global IoT Survey 2019 (2019). Available at: https://fi.farnell.com/global-iot-survey-2019 (Accessed: 10 March 2021).

GAO (2020) *GAO-20-577 Internet of Things*.

Hoikka, H. (2019) *Evaluation of Arrowhead Framework in Condition Monitoring Application*.

Ibrahim, N. (2009) 'Orthogonal classification of middleware technologies', in *3rd International Conference on Mobile Ubiquitous Computing, Systems, Services, and Technologies, UBICOMM 2009*, pp. 46–51. doi: 10.1109/UBICOMM.2009.24.

IFTTT (no date) *IFTTT helps every thing work better together*. Available at: https://ifttt.com/ (Accessed: 16 December 2020).

Khanna, A. and Kaur, S. (2020) 'Internet of Things (IoT), Applications and Challenges: A Comprehensive Review', *Wireless Personal Communications*. Springer, pp. 1687–1762. doi: 10.1007/s11277-020-07446-4.

Mahmood, K. *et al.* (2018) 'A Performance Evaluation Concept for Production Systems in an SME Network', in *Procedia CIRP*. Elsevier B.V., pp. 603–608. doi: 10.1016/j.procir.2018.03.182.

Mahood, Q., Van Eerd, D. and Irvin, E. (2014) 'Searching for grey literature for systematic reviews: challenges and benefits', *Research Synthesis Methods*. John Wiley and Sons Ltd, 5(3), pp. 221–234. doi: 10.1002/jrsm.1106.

Mainflux (no date) *About - Mainflux*. Available at: https://mainflux.readthedocs.io/ (Accessed: 16 December 2020).

Mineraud, J. *et al.* (2016) 'A gap analysis of Internet-of-Things platforms', *Computer Communications*. Elsevier B.V., 89–90, pp. 5–16. doi: 10.1016/j.comcom.2016.03.015.

Mohd Aman, A. H. *et al.* (2020) 'A Survey on Trend and Classification of Internet of Things Reviews', *IEEE Access*. Institute of Electrical and Electronics Engineers Inc., pp. 111763–111782. doi: 10.1109/ACCESS.2020.3002932.

Mosterd, L. (2019) 'The Openness between Platforms. What Changes in an IoT Context', *Master Thesis, Delft University of Technology*.

Node-RED (no date). Available at: https://nodered.org/ (Accessed: 16 December 2020).

Noura, M., Atiquzzaman, M. and Gaedke, M. (2019) 'Interoperability in Internet of Things: Taxonomies and Open Challenges', *Mobile Networks and Applications*. Springer New York LLC, 24(3), pp. 796–809. doi: 10.1007/s11036-018-1089-9.

Ramirez Lopez, L. J., Aponte, G. P. and Garcia, A. R. (2019) 'Internet of things applied in healthcare based on open hardware with low-energy consumption', *Healthcare Informatics Research*. Korean Society of Medical Informatics, 25(3), pp. 230–235. doi: 10.4258/hir.2019.25.3.230.

Serpanos, D. and Wolf, M. (2017) *Internet-of-things (IoT) systems: Architectures, algorithms, methodologies*, *Internet-of-Things (IoT) Systems: Architectures, Algorithms, Methodologies*. Springer International Publishing. doi: 10.1007/978-3-319-69715-4.

Taivalsaari, A. and Mikkonen, T. (2018) 'On the development of IoT systems', in *2018 3rd International Conference on Fog and Mobile Edge Computing, FMEC 2018*. Institute of Electrical and Electronics Engineers Inc., pp. 13–19. doi: 10.1109/FMEC.2018.8364039.

Teixeira, S. *et al.* (2020) 'LAURA architecture: Towards a simpler way of building situation-aware and business-aware IoT applications', *Journal of Systems and Software*. Elsevier Inc., 161, p. 110494. doi: 10.1016/j.jss.2019.110494.

Toivonen, V., Järvenpää, E. and Lanz, M. (2017) 'Managing production complexity with intelligent work orders', in *IC3K 2017 - Proceedings of the 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*. SciTePress, pp. 189–196. doi: 10.5220/0006507801890196.

Vogel, B. *et al.* (2020) 'What is an open IoT platform? Insights from a systematic mapping study', *Future Internet*. MDPI AG, 12(4), p. 73. doi: 10.3390/FI12040073.