On the Modular Structure and Evolvability of Internet of Things Architectures

Tom Vermeire, Jeroen Faes, Peter De Bruyn and Jan Verelst

Department of Management Information Systems
Faculty of Business and Economics
University of Antwerp, Belgium
Email: {tom.vermeire,peter.debruyn,jan.verelst}@uantwerp.be,
jeroen.faes@student.uantwerp.be

***Abstract*—The development of the Internet of Things (IoT) is an important evolution for current businesses. The field of IoT is maturing and best practice solutions are emerging. However, as organizations are confronted with increasing and faster changes in their environment, applications using IoT should be able to adapt and evolve accordingly. This paper assesses the ability to implement changes in best practice IoT applications, using Normalized Systems Theory as a theoretical basis. Subsequently, a new architecture addressing some identified evolvability issues is proposed. In contrast to existing prescriptive work focusing on the interoperability and standardization of IoT applications, this paper evaluates design choices from the perspective of the ability to evolve.**

Keywords–Internet of Things; Evolvability; Normalized Systems Theory.

I. INTRODUCTION

Internet of Things (IoT) envisions a network of connected physical objects allowing the exchange of data. It is generally seen as a promising evolution in the current and future business landscape, with a strongly increasing impact (in terms of the number of connected devices and business spending) [1]. An increase in the number of IoT applications, their criticalness for organizations and the number of devices they are relying on, give rise to challenges regarding scalability and implementation of changes resulting from additional requirements. Consequently, it is of major importance that the design of IoT applications allows to cope with future requirements (and the modifications they imply).

This paper uses Normalized Systems Theory (NST) to assess the evolvability of design decisions concerning IoT applications. NST is a theory which provides prescriptive guidance on how to design evolvable software architectures and, more generally, modular structures. Considering IoT applications as modular structures (consisting of a set of applications, devices, etc.), we argue and demonstrate that NST can be applied in this context. It is asserted that the current best practice architecture, whereby organizations typically make use of a one-stop vendor solution for data collection, storage and processing, offers limited evolvability. Afterwards, an enhanced IoT architecture, which addresses the identified problems, is proposed. In this new architecture, the different data-related activities are separated and organizations will experience an increased flexibility to implement changes. In contrast to existing prescriptive work, mainly focusing on interoperability and standardization issues, this paper approaches the design problem of IoT applications from their potential to adapt.

The remainder of this paper is structured as follows. In Section II, an overview of related work is given. Section III explains how IoT applications can be seen as modular structures subject to change. Afterwards, Section IV presents and evaluates the current best practice IoT architecture. Section V proposes an enhanced architecture from an evolvability point of view and Section VI discusses the result. Finally, Section VII concludes and offers avenues for future research.

II. RELATED WORK

This paper focuses on IoT and its current best practice solutions, and uses NST to analyze them. Therefore, this section briefly summarizes related work regarding each of these concepts.

A. Internet of Things

The International Telecommunication Union [2] defines IoT as "*a global infrastructure for the Information Society, enabling advanced services by interconnecting (physical and virtual) things based on, existing and evolving, interoperable information and communication technologies*" (p. 1). The potential applications of IoT span many industries, including logistics, healthcare, manufacturing etc., with a varying focus on enterprises, governments and consumers [3][4] .

The first use of the term 'Internet of Things' is attributed to Kevin Ashton [5], who argued that computers were too dependent on information input from humans. Instead, he advocated a shift towards data gathering by 'things'. In order to achieve this goal, physical objects should be equipped with sensors and radio-frequency identification (RFID) technology. Although the term IoT was new then, earlier contributions demonstrated comparable ideas including a Coke machine connected to the Internet in 1982 [6] or visions on ubiquitous compution [7] in which computers would amalgamate in the environment and their actual power would originate from the connection between devices. More recently, Mattern and Floerkemeier [8] described IoT as a situation wherein the existing Internet is extended into reality through the embracement of everyday physical objects. Various IoT definitions exist having an alternating focus between the connected things, the Internet-related aspects and the semantics of the information [9]. Overall, the focus has shifted from merely identifying and monitoring physical things towards smart objects that autonomously perform computer tasks.

From a business perspective, IoT is often seen as a potential way of capturing, communicating, and processing data in more advanced ways and the ability to perform advanced analytics or provide enhanced cloud services with a vast impact on current business models [10][11]. Recently, a multitude of IoT platforms was developed allowing companies and governments to create IoT applications [12]. These platforms typically make use of the cloud to store the gathered data [4]. It is

 ISBN: 978-1-61208-708-5

generally believed that advances in the power, size and cost of computing chips will significantly increase the number of connected objects in the following years [13].

B. IoT architectures and their challenges

Existing research on IoT architectures mainly focuses on the interoperability issues and standardization of technology.

For instance, Sethi and Sarangi [14] provide an overview of different IoT architectures. First, a layered architecture classifies the different IoT aspects on the basis of protocols (Perception, Transport, Processing, Application and Business). Second, cloud and fog based architectures take systems architectures as a starting point. Third, Social IoT attempts to mimic human social relationships in the IoT architecture. They stress the importance of middleware for data storage, analysis and processing. This middleware should abstract hardware details for programmers. In this context, they refer to a middleware platform as an appropriate solution to connect things and applications. As already stated, the use of cloud platforms within IoT architectures has become common practice.

Schmid et al. [15] also recognize the trend towards IoT platforms and argue that these platforms themselves should be interoperable in order to create IoT ecosystems comprising different industries. The BIG IoT Architecture is proposed as a solution, where IoT resources such as data or functions are offered in a standardized way on a marketplace. This marketplace offers application programming interface (API) endpoints for customers and providers and, thus, makes it possible for IoT services to operate with combined forces. Figure 1 provides a simplified overview of the BIG IoT Architecture. From left to right, a distinction is made between the customers, the market place and the providers. Although the proposed architecture consists of several other components as well, we only focus on a high-level overview in the context of this paper.

Figure 1. Simplified BIG IoT Architecture.

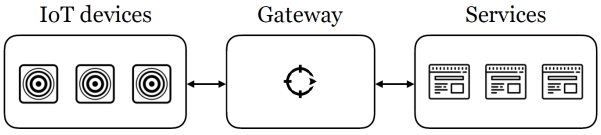

Figure 2. Simplified semantic IoT architecture.

Desai, Sheth, and Anantharam [16] see a similar challenge for IoT architectures. However, each domain can have a different standardized architecture and data model. In order to ensure the cooperation between multiple domains, standardization is required. Therefore, a semantic gateway is presented as a solution for the integration problem. The gateway aggregates annotated sensor data and connects with other physical things and (cloud) IoT services. The data annotation is realized by standard ontologies created by the semantic web community. A simplified overview of this semantic IoT architecture is shown in Figure 2. The gateway in the middle connects the IoT devices on the left side with the offered services on the right side.

Both examples provide a similar approach to encourage interoperability between different domains and their platforms: an intermediate agent ensures integration between things and IoT (cloud) services. Both approaches consider standardization as the primary challenge regarding interoperability, which the marketplace and semantic gateway each attempt to address.

Some authors within literature mention the evolvability of IoT applications as a relevant challenge. For instance, Weiser [7] pointed at the inability of existing operating systems to cope with changing hardware and software configuration. Wortmann and Flüchter [17] stated that being able to modify business models to IoT has become crucial and expressed the need for new design principles for applications to cope with updates of connected devices. Porter and Heppelmann [18] emphasized that organizations will be faced with continuously evolving IoT standards. Furthermore, the scalability of IoT applications is by many authors considered an important challenge [8][9][17]. A very specific and interesting situation was sketched by Priyantha et al. [19]. They investigated interoperable networks of sensors exposed to change and argued that current sensor-nets are not able to persist when new sensors with different protocols are added, possibly from different manufacturers. They provide two guidelines for IoT application design. First, sensors should be restricted to only generate structured data in order to be understandable for applications. Second, a programmatic description of the sensor's functionalities is prescribed. When sensors can be accessed in a structured way and programmatically by, for example, web services, the sensor-net is able to cope with newly added sensors and is, therefore, evolvable. It is stated that these findings are in line with the trend towards standardization in order to increase interoperability.

Clearly, the issue on how to provide an IoT architecture which ensures interoperability and allows for evolvability is considered relevant, challenging and open to further improvement.

C. Normalized Systems Theory

Originating from the field of software development, NST provides a number of design theorems that allow for the construction of evolvable software systems [20]. The theory is based on the domain of systems theoretic stability. Evolvability is seen by NST as the property of a software system that the impact of a change is not related to the size of the system. Assuming that the size of a software system is ever-increasing, this can be translated into Bounded Input Bounded Output (BIBO) stability.

Afterwards, NST has been formulated in a more general way, claiming that it can be applied to modular systems in general. According to the theory, every module should only contain one concern or change driver (Separation of Concerns or SoC), the use of a module by another module during its operation should be separated by a state (Separation of States or SoS), and a module used by or using other modules should be modifiable without impacting the others (Version Transparency or VT). It is shown that a violation of these theorems implies that a change of one module may impact other modules, coined as combinatorial effects. Since these effects depend on the size of a system, these are considered

 ISBN: 978-1-61208-708-5

harmful for future evolvability.

As the IoT environment is evolving and new applications in different domains are being developed, it is plausible to apply NST to this technology in order to evaluate the ability to cope with changes. Given the expected increase in the number of connected devices and applications, the idea of maintaining stability (i.e., an impact which is not dependent on the number of devices and applications) will only become of higher relevance.

III. An IoT applications as an evolving modular structure

An IoT application typically consists of several components or modules. As the purpose of IoT is to connect physical things, these things are building blocks of an IoT application. The connected physical things are referred to as the IoT devices. The data gathered by each IoT device have to be processed and stored. One way to achieve this is by building internal applications. Another often more cost-effective way is to outsource this resposibility to (cloud) platforms. The internal business applications and external platforms are seen as separate modules of the IoT application. These three high-level building blocks constitute the basis for a typical IoT application. As these building blocks themselves consist of different modules, a deeper modular structure can be derived. The connected things, for example, typically consist of sensors, actuators and a means of communication. Figure 3 offers a visual representation of the hierarchical modular IoT application structure.

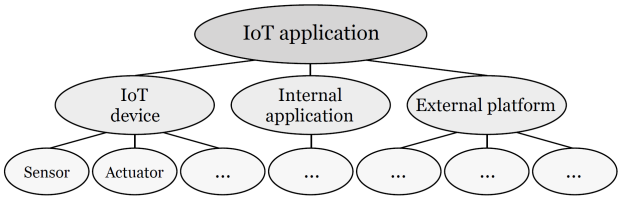

Figure 3. Modular IoT application structure.

It can be argued that the IoT environment is developing rapidly and that IoT applications will clearly be faced with changing requirements. Such changes can be due to legal requirements (e.g., General Data Protection Regulation) or technological requirements, such as the introduction of new types of sensors and actuators, new application domains and efforts towards standardization [21]. As a consequence, the modular structure might need to be changed: additional modules might need to be added or existing ones might need to be replaced by newer versions. Furtermore, different variants of the same module might be required to exist simultaneously. In the remainder of this work, it is assumed that the number of modules in a typical IoT configuration will grow over time and might become (theoretically) unlimited. This assumption is useful in order to detect NST combinatorial effects, as we will aim to do in the following sections.

IV. Best practice Internet of Things Architecture

We first discuss the current high-level architectural design of some well-known IoT platforms. Next, we evaluate their ability to adapt by using NST.

A. Architectural design

As mentioned earlier, several platforms have been developed to provide accessible IoT implementation capabilities to businesses. Several major technology players have developed an IoT cloud platform, each with its own vision on the architectural design of IoT applications. Typically, the cloud platform module is placed directly between the IoT devices and the internal applications of a company. Botta et al. [22] argued that the cloud is able to perform as a layer in-between things and business applications. They indicate that all complexity can be separated and that companies can focus on building the applications they need. Additionally, Gubbi et al. [3] refer to a general cloud framework as an intermediate agent between sensors on the one hand and private and public clouds on the other hand. This framework should allow developers to create applications without any complexity related to the cloud and sensor integration, as these are offered by the framework through services. The ability to create custom applications is deemed necessary, since a cloud platform usually does not offer a tailored solution for specific business problems [12].

As IoT implies, by definition, a very large number of devices and, therefore, a large amount of data, the cloud is presented by Botta et al. [22] as a solution for data storage, since it offers unlimited data storage capacity, on-demand and at low cost. Other advantages include the fact that the data stored in the cloud can be aggregated, protected by cloud security and distributed to the business applications to perform additional actions or visualizations. Furthermore, cloud platforms address a lack of sufficient computing power in IoT devices. The data is forwarded to a hub that performs data processing, in combination with aggregation. As infrastructure must be powerful enough to handle vast amounts of data, the unlimited processing power of the cloud offers a solution. In this way, the development and maintenance of IoT applications becomes more convenient and cost-effective for organizations when compared to in-house alternatives. At their turn, cloud platform providers are able to offer these services at lower prices due to economies of scale [12].

As an example of an IoT platform, AWS IoT Core can be considered. The product offers the possibility to connect all devices to the cloud platform, which in turn can integrate with other cloud and business applications via API calls [23]. Figure 4 represents this best practice architecture.

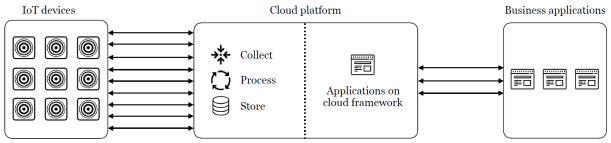

Figure 4. Best practice IoT architecture.

In some cases, such as with Google Cloud Platform, it is proposed to add a gateway between the IoT devices and the cloud data processing. The gateway is used to translate between different protocols used by connected devices. This is considered good practice because the gateway behaves in a similar way as an Enterprise Service Bus. The cloud platform only has to support the protocol of the gateway. Moreover, devices that are not directly connected to the Internet (e.g., Bluetooth devices), or cannot connect with the standards of the cloud platform, are still able to transfer data via the gateway. The IoT architecture in Figure 5 includes such a separated

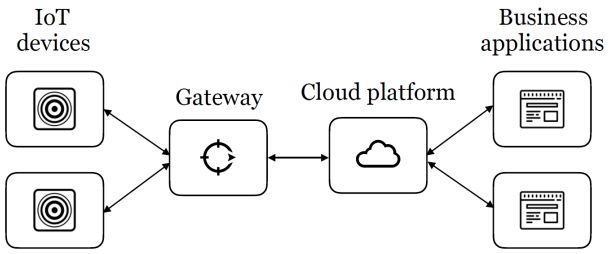

Figure 5. Simplified best practice with a gateway.

gateway [24].

In addition, a gateway should make it effortless to switch platforms, since no separate link to a specific platform is made for every IoT device. However, in this specific example, the gateway is from Google itself and works best with other cloud services of the company. At first sight, Google Cloud correlates with the use of a gateway proposed by Desai, Sheth and Anantharam [16]. However, the gateway of Google Cloud only aims at realizing integration at the technological level, whereas Desai, Sheth and Anantharam focus on semantic integration.

B. Evaluation

To evaluate the evolvability of an IoT application architecture, the following three types of modules are considered: connected devices, external platforms and internal business applications. This corresponds to the second modularity level in Figure 3. It is assumed that the number of instances of each of the three module types can become unlimited over time. In the current best practice architecture, three data activities (collection, storage and processing) are centralized in one module, the external platform. This can be considered a violation of the SoC theorem (as each of them is a concern that can change independently) and causes several issues.

Business applications are dependent on changes of the cloud platform for their internal working. To make use of services, these applications have to settle for the endpoints that are made available by the cloud platform provider. As a new version of the cloud platform can independently change internal data structures, data conversions, data aggregations and API endpoints, this possibly affects the internal working of business applications. For example, in case a certain service returns its result in a modified measurement unit, the invoking business applications should implement this change. The best practice IoT architecture proposes to connect business applications with the endpoints provided by the cloud platform. When these connections are not properly separated or encapsulated (thereby constituting a violation of VT), the impact of such a change is dependent on the size of the system (i.e., the number of service invocations) and, as a result, combinatorial effects occur. Moreover, a service used by the business for internal processes may not be available anymore in a new version of the platform. In that case, it might be necessary to look for another service provider to perform that specific task. However, since data collection, storage and processing are all centralized on the same platform, easily switching the provider for one of these services is not feasible. This is an implication of the SoC violation. Lamarre and May [12] have confirmed that businesses are usually not switching platforms. The difficulties described above may be a reason for this.

Furthermore, there are *dependencies between the cloud platform and the IoT devices of the organization*, as the platform is responsible for data collection. It is, for instance, possible that new versions of the cloud platform cause compatibility issues. In case the updated cloud platform does not support the original IoT devices, an update or a replacement of every device might be necessary. As the impact of this change clearly depends on the size of the system (i.e., the number of devices), combinatorial effects arise. Additionally, similar problems can occur as a consequence of IoT device updates. If the organization's IoT device vendor launches a new version of the device that is not supported by the used cloud platform, a normal extension of the IoT application is not possible. In that case, to be able to increase the size of the system (i.e., the number of devices), the organization might be forced to change the used cloud platform. As outlined above, a new (version of a) cloud platform will demand changes to the business applications and the IoT devices already in use. Alternatively, the organization could also look for another type of device that is compatible with its current cloud platform. However, using devices from different vendors with different technologies and protocols clearly increases the complexity of the IoT application. Again, these issues are a consequence of the fact that data collection, storage and processing are all performed on the same external platform and, therefore, not properly applying SoC.

As businesses might deploy applications on the cloud platform itself for business-specific processes, the dependency on the cloud platform increases further. These applications are typically built upon cloud frameworks for specific external platforms, making it more burdensome to switch between cloud platform providers. Therefore, we consider it safe to conclude that the centralization of data collection, storage and processing in one cloud platform potentially causes the occurrence of several combinatorial effects and, therefore, offers challenges regarding evolvability.

Two other important implications of the current best practice architecture need to be mentioned: firstly, as a consequence of the direct connection between the IoT devices and the cloud platform, the organization is not the owner of the data in its original form. The data is typically accessible for business applications by making use of API calls to the cloud platform. Companies have no control over possible conversions or aggregations that the cloud platform applies to the raw data before making it accessible. This implies (or can in the future imply) that not all raw data from the IoT devices may be accessible for the business, as the platform can implement these changes independently. Secondly, the centralization of data collection, storage and processing in one external platform possibly results in a vendor lock-in. Entrusting one external party with all these important responsibilities offers this party a considerable amount of power. As outlined above, switching platforms is a difficult undertaking, which reinforces the control of the platform provider. Additionally, outsourcing the three main data activities to a standardized platform raises the question as to what extent an organization is able to realize a competitive advantage. Opposed to what is usually desired, the strong dependency on one external provider will force businesses to adapt in function of the platform requirements.

V. Towards a Normalized IoT Architecture

Based on the evaluation above, we propose a modified architecture for IoT applications. In this architecture, *every IoT device is connected with a company gateway*, which is in

turn connected with the IT landscape of the enterprise. Raw data from the IoT devices is stored and business applications can use the original data. Although the data is not directly sent to the cloud, the possibility to connect with external (cloud) platforms still exists in the proposed architecture. The gateway between the business landscape and external platforms ensures connections with the API endpoints offered by the platform providers. A graphical overview is given in Figure 6.

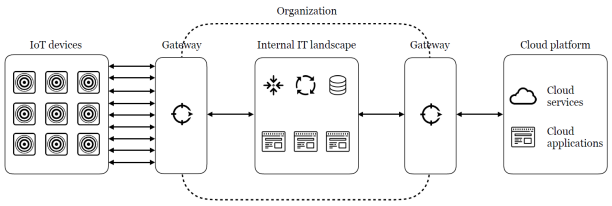

Figure 6. Proposed IoT architecture.

First, regarding *data gathering and storage*, it is the organization itself which assumes the responsibility for data gathering, since a link between the IoT devices and the information systems architecture of the organization is established (instead of working directly via an external cloud platform). This implies that the organization itself has full control over the raw data. In this way, the organization can decide freely on data types, conversions and aggregations and keeps ownership of data in its original form. The organization can also decide on how the data is stored. In case the company has data storage capacity, internal databases can be used. Furthermore, it remains possible to rely on an external cloud provider. In any case, only a lower degree of dependency on external platforms remains: there are no manipulations on original data by external parties and, since sensors are not directly connected with the external platforms, changes within these platforms have no impact on the IoT devices the company uses. In case an update of the external platform results for instance in new data requirements, the organization itself can perform the necessary data conversions if needed.

Second, regarding the *IoT devices*, it can be argued that updates can be implemented with a limited impact. The organization is free to choose which type(s) of devices it uses and by properly separating the connection between the IoT devices and in-house information systems, the impact of modifications to existing devices can be controlled and centralized into one location. Opposed to the best practice IoT architecture, the proposed architecture encapsulates the connection with external IoT devices and changes regarding these connections will not result into combinatorial effects.

Third, regarding *data processing*, the organization can still make use of an external analysis platform if preferred. It can load the relevant data from its custom or cloud databases into the platform and perform the necessary analyses. To use generated insights in custom business applications, connections with these platforms can be established via the typical API calls to the platform. Also here, if these calls are properly separated and encapsulated in the organization's business applications, the impact of future changes is contained.

In essence, our proposed architecture attempts to separate the different services offered by external platforms. Although a platform provider can still be used for multiple activities, these activities must be separated from each other and managed by a stateful controller in the organization's application. The proposed architecture improves switching opportunities by placing the internal IT landscape between the IoT devices and the external platforms. Therefore, it is expected that the likelihood of a vendor lock-in is reduced.

Furthermore, it was stated in Section IV that the positioning of an external platform between IoT devices and the internal IT landscape hampers an organization's ability to realize a competitive advantage. Indeed, making use of standardized packages for general problems offers no unique business value to the operations. The proposed architecture, however, provides decision freedom to the organization itself regarding which platforms or services to use for which functionality and which time, and to revise those decisions later on. This ensures that the business can autonomously decide where added value is created and what differentiates them from competitors.

VI. Discussion

In the proposed IoT architecture, it is possible to use various external technology providers. It can be argued that this increases complexity when compared to the current best practice architecture, where one external platform offers a one-stop solution. However, the newly proposed architecture offers an improved ability to evolve. In general, organizations that want to use IoT in their operations face a tradeoff between initial complexity and evolvability. In the short term, limiting apparent complexity with a one-stop solution might be a natural choice. Nevertheless, only an evolvable modular IoT architecture will enable an organization to create a sustainable competitive advantage.

It should be mentioned that the proposed IoT architecture correlates to some extent with previous research. Schmid et al. [15] recommend the BIG IoT architecture in which a marketplace integrates consumer applications with providers of sensors, storage, and other IoT services. The marketplace acts as an intermediate agent and translates messages between each coupled consumer and provider. The gateways in the newly proposed architecture have a similar purpose: creating a standardized means of communication between internal business applications and external parties. A difference, however, is that the proposed gateways are managed internally by the organization. The marketplace from the BIG architecture is managed by an external party. This implies that, when the BIG IoT architecture is implemented, API calls to the marketplace have to be separated properly and, in theory, another internal gateway is required between business applications and the marketplace. Similarly, Desai, Sheth, and Anantharam [16] introduced a semantic gateway between IoT devices and IoT services. However, in our approach, gateways are not placed directly between IoT devices and IoT services in order to give ownership of raw data to the business itself. Moreover, the semantic gateway immediately performs data annotations and aggregations before sending information to service providers. It can be argued that, when raw data is important, these operations should be avoided in the gateway itself.

In conclusion, some existing approaches from other perspectives have already proposed architectures similar to our solution from a modularity and evolvability perspective. However, an important difference is that our solution includes an indirect connection between IoT devices and external platforms implying first-handed control over IoT data. Moreover, our architecture stresses that all externalities should be properly separated from the internal IT applications to facilitate future

 ISBN: 978-1-61208-708-5

changes. As every external technology should be considered as a change driver, every dependency needs to be encapsulated in a separate module (SoC) with a version transparent interface (VT).

VII. Conclusion

IoT is a promising technological trend with (potential) applications in various industries. As IoT is maturing and business agility becomes key, it is of major importance that IoT applications are able to enable such agility as well. Based on the NST, this paper assessed the extent the current state-of-the-art IoT architectures are evolvable and presented a new IoT architecture addressing the issues found.

The current best practice IoT architecture is often a one-stop vendor solution, in which the responsibilities of data collection, storage and processing are combined in one external cloud platform. To employ the gathered insights, organizations can make use of web services offered by the platform. Although the current best practice architecture has several advantages (cost reduction, use of external computing power, limited complexity), this also presents some weaknesses from an evolvability perspective: combinatorial effects (i.e., a specific type of ripple effects) may occur as a result of changes to external platforms and connected devices. This hampers the ability to extend and adapt IoT applications in order to address changing requirements. Furthermore, assigning different data-related activities to one external party results in the absence of raw data ownership and the risk of a vendor lock-in. The newly proposed IoT architecture aims to enhance evolvability and increase an organization's control by separating the different data-related activities and using indirect connections between the internal IT landscape and external modules (platforms and connected devices).

The main contribution of this paper is its analysis of current best practice IoT architectures in terms of evolvability on a theoretical grounding which is, as far as we know, new. On the one hand, this might increase our conceptual understanding of present IoT architectures. On the other hand, this might provide practitioners with additional guidance on how to design their IoT solutions in a more evolvable way. In particular, we believe that our proposed normalized IoT architecture might prove valuable in that respect. Another contribution of this paper is situated in the demonstration of the feasibility of applying NST in a new domain (i.e., IoT) for which it has not been applied earlier. However, our work is also subject to some limitations and opportunities for future work. For instance, while we have indicated the need to isolate and encapsulate all external dependencies within the internal IT landscape, this does not guarantee that the application internally is fully evolvable and free of combinatorial effects (as also here, the NST principles should be applied for that purpose). Also, our discussion of a normalized IoT architecture was purely conceptual and not tested in practice. Therefore, future research could examine the technological feasibility of our proposal in practice.

References

[1] Gartner, "Gartner says 8.4 billion connected "things" will be in use in 2017, up 31 percent from 2016," https://www.gartner.com/newsroom/id/3598917, 2017, electronically retrieved on April 8th, 2019.

[2] International Telecommunication Union, "Series y: Global information infrastructure, internet protocol aspects and next-generation networks," http://www.itu.int/ITU-T/recommendations/rec.aspx?rec=11559, 2012, electronically retrieved on April 8th, 2019.

[3] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," Future generation computer systems, vol. 29, no. 7, 2013, pp. 1645–1660.

[4] I. Lee and K. Lee, "The internet of things (iot): Applications, investments, and challenges for enterprises," Business Horizons, vol. 58, no. 4, 2015, pp. 431–440.

[5] K. Ashton, "That 'internet of things' thing," RFID Journal, vol. 22, no. 7, 2009, pp. 97–114.

[6] Carnegie Mellon University, "The "only" coke machine on the internet," https://www.cs.cmu.edu/~coke/history_long.txt, 2018, electronically retrieved on April 8th, 2019.

[7] M. Weiser, "The computer for the 21st century," Scientific American, vol. 265, no. 3, 1991, pp. 94–105.

[8] F. Mattern and C. Floerkemeier, "From the internet of computers to the internet of things," in From active data management to event-based systems and more. Springer, 2010, pp. 242–259.

[9] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," Computer Networks, vol. 54, no. 14, 2010, pp. 2787–2805.

[10] J. Bughin, M. Chui, and J. Manyika, "An executive's guide to the internet of things," McKinsey Quarterly, vol. 4, 2015, pp. 92–101.

[11] A. Bosche, D. Crawford, D. Jackson, M. Schallehn, and P. Smith, "Defining the battlegrounds of the internet of things," http://www.bain.com/publications/articles/defining-the-battlegrounds-of-the-internet-of-things.aspx, 2016, electronically retrieved on April 8th, 2019.

[12] E. Lamarre and B. May, "Making sense of internet of things platforms," https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/making-sense-of-internet-of-things-platforms, 2018, electronically retrieved on April 8th, 2019.

[13] S. Chen, H. Xu, D. Liu, B. Hu, and H. Wang, "A vision of iot: Applications, challenges, and opportunities with china perspective," IEEE Internet of Things journal, vol. 1, no. 4, 2014, pp. 349–359.

[14] P. Sethi and S. R. Sarangi, "Internet of things: architectures, protocols, and applications," Journal of Electrical and Computer Engineering, vol. 2017, 2017.

[15] S. Schmid et al., "An architecture for interoperable iot ecosystems," in Proceedings of the International Workshop on Interoperability and Open-Source Solutions, 2016, pp. 39–55.

[16] P. Desai, A. Sheth, and P. Anantharam, "Semantic gateway as a service architecture for iot interoperability," in Proceedings of the 2015 IEEE International Conference on Mobile Services (MS), 2015, pp. 313–319.

[17] F. Wortmann and K. Flüchter, "Internet of things," Business & Information Systems Engineering, vol. 57, no. 3, 2015, pp. 221–224.

[18] M. Porter and J. Heppelmann, "How smart, connected products are transforming competition," Harvard Business Review, vol. 92, no. 11, 2014, pp. 64–88.

[19] N. Priyantha, A. Kansal, M. Goraczko, and F. Zhao, "Tiny web sservice: design and implementation of interoperable and evolvable sensor networks," in Proceedings of the 6th ACM conference on embedded network sensor systems, 2008, pp. 253–266.

[20] H. Mannaert, J. Verelst, and P. De Bruyn, Normalized Systems Theory: From Fondations for Evolvable Software Toward a General Theory for Evolvable Design. Koppa, 2016.

[21] L. Columbus, "2017 roundup of internet of things forecasts," https://www.forbes.com/sites/louiscolumbus/2017/12/10/2017-roundup-of-internet-of-things-forecasts/#4d55f6451480, 2017, electronically retrieved on April 8th, 2019.

[22] A. Botta, W. De Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and internet of things: a survey," Future generation computer systems, vol. 56, 2016, pp. 684–700.

[23] AWS, "Aws iot core," https://aws.amazon.com/iot-core/, 2019, electronically retrieved on April 8th, 2019.

[24] G. Cloud, "Overview of internet of things," https://cloud.google.com/solutions/iot-overview, 2018, electronically retrieved on April 8th, 2019.

 ISBN: 978-1-61208-708-5