TYPE Specialty Grand Challenge
PUBLISHED 22 August 2022
DOI 10.3389/friot.2022.959268

OPEN ACCESS

EDITED AND REVIEWED BY
Rajkumar Buyya,
The University of Melbourne, Australia

*CORRESPONDENCE
Marco Aiello,
marco.aiello@iaas.uni-stuttgart.de

SPECIALTY SECTION
This article was submitted to IoT Architectures,
a section of the journal
Frontiers in the Internet of Things

RECEIVED 01 June 2022
ACCEPTED 30 June 2022
PUBLISHED 22 August 2022

CITATION
Aiello M (2022), IoT architectures: from data to smart systems.
Front. Internet Things 1:959268.
doi: 10.3389/friot.2022.959268

IoT architectures: from data to smart systems

Marco Aiello*

Service Computing Department, Institute of Architecture of Application Systems, University of Stuttgart, Stuttgart, Germany

KEYWORDS

IoT architectures, ubiquitous and pervasive computing, artificial intelligence, smart systems engineering, software architecture

1 The vision of ubiquitous computing

"The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it." It was 1991 when Mark Weiser's pioneering article was published in the Scientific American, discussing a future where networked devices would be so ubiquitous that "no one will notice their presence" (Weiser (1991)). To a large extent, this is our reality today. We may have changed terminology and moved from ubiquitous computing to Internet of Things, but the substance has not changed. We are surrounded by data-capturing computing devices that are always on, are networked, and many of which can perform world-altering operations. Just think of hopping into a modern car. The driver is actually interacting with a distributed computing system on wheels. With up to 100 electronic control units and hundred of millions of lines of code, the modern automotive industry has become more about software than mechanics and aerodynamics. And yet, we think and interact with our car as we would have done in 1991. Or think at our symbiotic relationship with social media as experienced *via* our personal data capturing device: the smartphone. It is symbiotic because it satisfies the human need for emotional connectedness while it feeds content to the social media infrastructure, content that is essential for its existence and that is typically metabolized as targeted advertisement. The modern human being has feelings relatable to gymnophobia whenever leaving home without the smart phone, a fear of a sensation of nakedness, incompleteness that transcends the rational. And dually we accept any other human being we are close to or even interacting with to be concurrently doing something on their smart phone. The phone is such an integral and accepted part of who we are and how we behave that we can agree that Mark Weiser's prediction was correct: phones are our everyday life and we do not even notice anymore.

In many countries the level of penetration of mobile phones is above 80%, including population of any age (Statista (2022)). The number of active phone subscriptions is higher than 7 billion. If we consider Bluetooth, a technology often used for dynamic connectivity at the edge of the system, 4 billion Bluetooth Low Energy network interfaces are currently been shipped per year (Bluetooth SIG (2021)). Number projected to surpass 6 billions by 2025. And more generally, the predictions indicate that by 2030 the total amount of IoT devices worldwide, of any type, will reach the value of 24 billion (Transforma Insights (2020)). The unprecedented advancements in realizing IoT devices at affordable prices and the pervasive connectivity of wireless and wired Internet technologies, are essential building blocks to achieve the vision of a smart, connected world; but this by itself is not enough. In fact, IoT are basic, imprecise, physical measuring devices with the ability to perform one or two basic actions. They produce vast amounts of raw data which, most often, carries very little information as it is. Say that a passive infrared sensor (PIR) has detected movement three times in the last thirty minutes, do we know if an elephant was in the room, or if someone was entering and exiting that room to go to the printer, or if a butterfly was flying near the sensor?

Rarely IoT devices are useful in isolation, their utility comes from being part of a larger architecture that uses each one of them for a purpose. By 'architecture' here we mean the fundamental software and IoT elements and the intended relations and constraints governing their interactions (see Kruchten et al. (2006) for a classic overview from the software perspective). Specifically regarding IoT architectures, the first major purpose is that of context recognition. Understanding the state of the world through sensing. To achieve this, one needs to have models of the world, have many measurements over time, and have some form of process that can assign sensor data to a member of the class of possible states of that world. In other terms, it is the shift from, e.g., PIR, camera, and temperature data to a higher level concept of a room being occupied by a human working at a desktop computer. The second major purpose is that of actuation. Given a certain context of the world, and having a desired state for such a world, then how can such desired state be realized? Here is where the definition of smartness plays a role. The system should perform a set of actions that achieve the goal of the user in a way that satisfies the user and the user would judge as the best possible course of actions to take. Something he or she would have done. Or something done even better than what the human could have. Again, to achieve smartness, IoT devices need to be embedded in a larger architecture where there are components that can realize smartness.

Thus, we have two major aspects related to our IoT devices: data and actuation. From the users' point of view, both are characterized by high complexity. Data's complexity comes from its extremely large volume, its general inconsistency, its velocity, and its implicit meaning. Data is overwhelming for the average user and can rapidly lead to frustration. Actuation requires detailed knowledge of the physics of the world in which one is actuating and how this is translated into human experiences. Here the complexity lies in the understanding of cause-effect relations and in searching enormous spaces of possibilities to find optimal states for the user that he or she can consider smart. Interestingly, the problem of complexity reduction, if not shear data reduction, has characterized the history of our field since its birth. Just think of the original goal of Vannevar Bush with his 1945 Memex design (Aiello, 2018). He wanted to create a tool to support people in their intellectual tasks. The Memex was mechanical and based on microfilms; the first computer-based version of a similar tool was the oNLine System (NLS) of Douglas Engelbart demoed in 1968. Similarly to Bush, Engelbart was concerned with supporting humans with machines that helped navigate and take advantage of structured knowledge in a world of growing complexity. Today, most likely, we need a Memex for our IoT rich world. We need architectures that can alleviate us from the data and smart actuation complexities and deliver solutions that are safe, useful, and self-explaining.

2 Three challenges of IoT data

Raw data per se is useless. It is only of help when it supports us in attributing meaning to a situation and in turn taking actions that increase our utility. To better understand this, consider the following challenges of data.

1) **Data is not information**. Raw data coming from sensors, advanced metering infrastructures, mobile devices is in general meaningless. These are quantities measured at a given point of time which on their own do not mean much. If I say that your heart beat is at 78 beats per minute at 12:15 of a given day, is it good or bad? You can't really say as that number is hard to interpret on its own. On the other hand, if I have statistical data of people of your age group and historic data of you correlated with the activities you are executing, I can derive a conclusion of a healthy or potentially unhealthy value. This is a general issue. Individual data points usually mean very little; it is by comparing them with historical values and having a model of the domain (possibly implicitly encoded in a machine learning classifier) that those data points become useful. In other words, an IoT Architecture must have a component that can process raw data and use it to identify, up to a given level of certainty, the state of the world.
2) **Data is dirty and inconsistent**. IoT cheap, wireless, often battery based sensors, also means that the readings are not always reliable. A device will occasionally give one out of range, odd reading, or the quality of its readings will decrease when the battery begins to deplete. Messages will occasionally be omitted. Furthermore, idiosyncratic sensors placed in the same environment can give inconsistent readings. Think for example of a light returning a status of being on and a light sensor giving back a reading that indicates darkness. Clearly the two values are inconsistent. Therefore, we need techniques to deal with dirty and inconsistent data and extract the most plausible classification based on such readings. The component responsible for raw data processing also has diagnostic abilities to detect possible hardware or software failures which, on large scale systems, will be frequent.
3) **Data is sensitive**. Data coming from physical space, especially if populated by humans, carries information about those humans. In its most basic form it might help identify presence, but it can also provide more information on

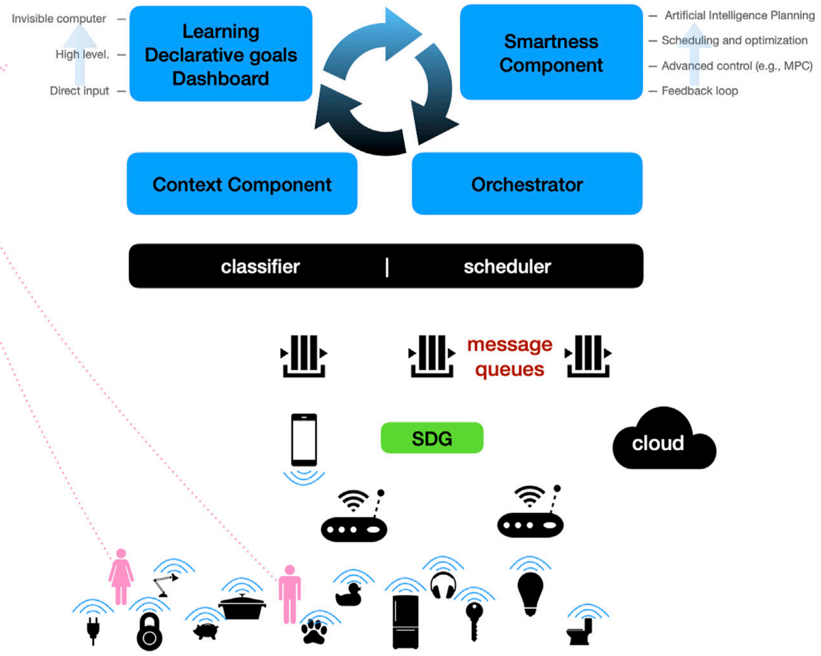

FIGURE 1
The generic smart IoT architecture.

which activities the people are carrying out. When collecting data one should always consider which privacy intruding information could be extrapolated from that data. A principle of *least information* should guide data collection and storage decisions, that is, one should collect the minimum amount of data that is necessary for the running of the application and that least intrudes into people's privacy. Minimizing sample frequency and sensor measuring capabilities are typical decisions that help minimize privacy intrusion. With very few exceptions, avoiding privacy issues is impossible for any IoT Architecture, but careful handling is possible and should be pursued.

3 Three challenges of IoT smart actuation

We established that data is useful only when it helps to determine the context, the state of the world in which the IoT application is executing. We could similarly say that context without smart actuation is useless. If we know what the situation is with a good level of accuracy, but we do not know what to do, then we will not deem any application useful or smart. The most basic form of actuation is a simple feedback loop. Every time that a certain state is detected, a specific action takes place. Classical example: every time that a passive infrared sensor detects movement in a bathroom, the light and fan are switched on for 10 minutes. Most likely we would consider this convenient, but not really smart. This brings us to the challenges that surround the problem of IoT Smart Actuation.

1) **Who is in control?** Who should trigger actuation? Is the change of context by itself, or a direct input of the user, or some combination of the two? Ideally, the user should do nothing explicit but just go about her/his normal business and actuation should occur to support her/his activities. Actuation should be transparent, even unnoticeable as if the computer was invisible to the user (Norman (1998)). In practice, often the users have to at least declare their preferences or goals to the system. In the worst case, the user must give the sequence of instructions for the system to follow.
2) **Smartness is a human attribution**. Something is considered smart only if the user recognizes it as such. Actuation should feel natural for the user and understandable. When a system makes a decision and changes the world, the user has to be able to interpret why such actuation has taken place and to understand its benefits. The user only can perceive something as being smart, or a terrible, stupid decision. Such perception is what in the end allows for an IoT Architecture to be accepted or not by the user. Smartness is associated with explainability. The user has to be able to understand why something happened and appreciate it as a smart solution. Optimal solutions from the point of view of the application, are not necessarily perceived as smart by the user.

3) **Is it worth it?** There is a clear hierarchy of actuation goals that should always be followed. The first goal must be the safety of the users. If safety is satisfied, then one can consider to satisfy goals of comfort and productivity. Finally, if these are satisfied, one can consider the optimization of concerns of economic savings and of sustainability issues. In other terms, while smart actuation is taking place, there is a number of strictly ordered concerns that cannot be violated, from safety of the users all the way down to sustainability. Saving a few euros of electricity is not worth risking the life of any human. This will be reflected in the architecture where the smartness component needs to account for such priorities and at the physical level where detectors for hazardous situations and actuations need to be installed.

The above concerns indicate that delicate design decisions need to be taken when realizing an IoT architecture to create a smart system. How the user controls and interacts with the system are crucial for its acceptability and its usefulness. Furthermore, there are important system constraints. In line with Asimov's laws on robotics (Asimov (1941)), also smart IoT solutions that deal directly or indirectly with humans should be self-preserving while never harming a human in their operations. Finally, there are concerns of privacy and of human interaction that go beyond the technology. The propositions of the recently published Vienna Manifesto on Digital Humanism apply, to a large extent, also to IoT architectures (Wertner et al. (2019)). To complete the above challenges, typical software architectural challenges of data processing, function distribution and optimization, heterogeneity masking, fault tolerance and resilience also affect IoT architectures.

4 IoT architectures

The three data and smart actuation challenges just identified are not the only challenges characterizing IoT architectures, others include the need to deal with large amounts of raw data, with the highly asynchronous nature of communication, and with the high volatility of the system. By considering all of these, one can mine a set of architectural best practices.

The common architecture is a data processing one which is cyclic in nature. Asynchrony is managed by appropriate middleware which is event-based and typically organized around message queues. The movement from data to context and information is typically handled by dedicated classifiers which, in some cases, can adapt dynamically to the changing data patterns. Smartness comes from actuation and reasoning techniques routed in control theory, optimization and scheduling, and Artificial Intelligence Planning. Smart actuation has to have some form of spatial and temporal representation, since actuation is location based and actions are not necessarily instantaneous, but can extend in time, even for hours (just think of heating a large meeting room to increase its temperature by 10°). This brings us to the layered architecture shown in Figure 1. The lower layer is the actual physical, IoT layer, where people and devices live. These are interfaced to the architectures by gateways or direct interaction with human devices such as phones and wearables. Software Defined Gateways (SDG) and cloud services are accessed by the higher layers of the architecture in an asynchronous way. Here is where message queues are employed. These are bidirectional. Data flows towards the upper layers and actuation commands flow from the upper layers to the devices, possibly via the gateways.

At the device level, a number of patterns for the device interoperability, interaction and communication have been identified by Reinfurt et al. (2016). To address the heterogeneity concerns and to ease development and portability, Dustdar et al. define a notion of Software-Defined-Gateway (Dustdar et al. (2017)). These are generic solutions for IoT architectures that support the integration of diverse IoT technologies within smart architectures, as described above. SDGs are a powerful software engineering tool to separate concerns and to increase the portability and deployability of IoT solutions. It is clear that IoT architectures do require some additional software engineering tools to complement the traditional ones available. Testing and debugging become more challenging. Bugs can hide themselves more ubiquitously than in traditional information system code. They can be in the software running on the cloud, but also in the firmware of an IoT device, or simply in a networking component. Isolating and reproducing a bug will be very difficult. Related to this is the fact that during development often the IoT components are simulated and the actual behavior of the hardware might deviate from the simulated one. Another potential source of run-time bugs after deployment.

The upper layer of the architecture is where the cyclic nature of *measure, decide and execute* emerges. The data from the message queues is processed to create contextual information. The data is cleaned, correlated, and fed to classifiers for the extraction of actionable information. The users then express their desires and requirements. This can happen explicitly by using dashboards, switches, apps, or it can happen implicitly. The system learns usage patterns and anticipates the needs of the user. Then, the desired state and the context information are fed to the smartness component. Based on these two inputs the "smart" decisions are made on what actuation is necessary, if any. The smartness component can be a simple feedback loop implementation or some more refined form of planing and scheduling. The plan of actions created by this component can then be fed to the orchestrator component which in turn decides how to deploy the plan and how to run the execution. The orchestrator manages the plan and translates the actions into messages to be placed in the message queues for IoT device consumption. This clearly changes the state of the world, in turn changing the context, and restarting the cycle of *measure, decide, and execute.*

The architecture in Figure 1 is a generalization of IoT architectures actually deployed. Most of these focus on the IoT and classification aspects, see for instance (Weyrich and Ebert (2016)) for a review of industrial IoT architectures, or (Fremantle (2015)) for a specific layered proposal with cloud backends, while Degeler and Lazovik (2014) extract patterns specifically for context extraction in smart environments. In my own research, together with colleagues, I designed, realized and deployed variations of the complete architecture in Figure 1 for smart offices (Georgievski et al. (2012)), for smart buildings (Georgievski et al. (2017)), and proposed it as a generic blueprint for smart energy systems (Aiello et al. (2021)).

5 Concluding remarks

IoT is a true technical and societal revolution, giving to our everyday objects a digital life that intertwines with our physical one. The possibilities to use this technology to improve the safety, comfort and sustainability of our daily activities are limitless. Our task is to harness IoT's full potential, and to do so we need to build efficient, tailored software architectures. The section on IoT Architectures of Frontiers on IoT is dedicated to describing innovative designs, implementations, evaluations, and experiences with IoT architectures in any possible physical domain.

Author contributions

The author confirms being the sole contributor of this work and has approved it for publication.

Conflict of interest

The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Aiello, M., Fiorini, L., and Georgievski, I. (2021). *Handbook of smart energy systems*. Heidelberg: Springer. chap. Software Engineering Smart Energy Systems.

Aiello, M. (2018). *The web was done by amateurs*. Heidelberg: Springer.

Asimov, I. (1941). *Three laws of robotics*. New York: Asimov, I. Runaround.

Bluetooth SIG (2021). Bluetooth 2021 market update. Available at: https://www.bluetooth.com/wp-content/uploads/2021/01/2021-Bluetooth_Mar ket_Update.pdf.

Degeler, V., and Lazovik, A. (2014). "Architecture pattern for context-aware smart environments," in *Creating personal, social, and urban awareness through pervasive computing* (Hershey: IGI Global), 108–130.

Dustdar, S., Nastić, S., and Šćekić, O. (2017). *The Internet of Things, people and systems*. Heidelberg: Springer.

Fremantle, P. (2015). *A reference architecture for the Internet of Things*. Colombo, Sri Lanka: WSO2 White paper, 02–04.

Georgievski, I., Degeler, V., Pagani, G. A., Nguyen, T. A., Lazovik, A., Aiello, M., et al. (2012). Optimizing energy costs for offices connected to the Smart Grid. *IEEE Trans. Smart Grid* 3, 2273–2285. doi:10.1109/TSG.2012.2218666

Georgievski, I., Nguyen, T. A., Nizamic, F., Setz, B., Lazovik, A., Aiello, M., et al. (2017). Planning meets activity recognition: Service coordination for intelligent buildings. *Pervasive Mob. Comput.* 38, 110–139. doi:10.1016/j.pmcj.2017.02.008

Kruchten, P., Obbink, H., and Stafford, J. (2006). The past, present, and future for software architecture. *IEEE Softw.* 23, 22–30. doi:10.1109/MS.2006.59

Norman, D. A. (1998). *The invisible computer: Why good products can fail, the personal computer is so complex, and information appliances are the solution*. Cambridge, MA: MIT press.

Reinfurt, L., Breitenbücher, U., Falkenthal, M., Leymann, F., and Riegg, A. (2016). "Internet of Things patterns," in *Proceedings of the 21st European conference on pattern languages of programs* (New York: ACM), 1–21.

Statista (2022). Number of smartphone users worldwide. Available at: https://www.statista.com/statistics/330695/number-of-smartphone-users-w orldwide.

Transforma Insights (2020). Global IoT market to grow to 24.1 billion devices in 2030, generating $1.5 trillion annual revenue. Available at: https://transformainsights.com/news/iot-market-24-billion-usd15-trillio n-revenue-2030.

Weiser, M. (1991). The computer for the 21st century. *Sci. Am.* 265, 94–104. doi:10.1038/scientificamerican0991-94

Wertner, H., Lee, E. A., Akkermans, H., Vardi, M., Ghezzi, C., Magnenat-Thalmann, N., et al. (2019). *Vienna manifesto on digital humanism*. Vienna: DIGHUM.

Weyrich, M., and Ebert, C. (2016). Reference architectures for the Internet of Things. *IEEE Softw.* 33, 112–116. doi:10.1109/MS.2016.20