Rev. Roum. Sci. Techn.– Électrotechn. et Énerg.
Vol. **67**, *3*, pp. 337–341, Bucarest, 2022

Automatique et ordinateurs

ARCHITECTURE CONSIDERATIONS FOR COMMUNITIES OF SMART OBJECTS

IOAN COSMIN RADU[1]

Keywords: Internet of things; Architectures; Smart objects; Edge computing; Fog computing; Mist computing; Mesh network; Microservices; Raspberry Pi.

The article presents the origin of the Internet of Things (IoT) and how the concept has evolved over time. Four architectures for IoT implementations are identified and described: central point architecture, decentralized architecture, hybrid architecture, and semi-centralized architecture. Next, the concept of an intelligent object is defined, and its characteristic aspects are described. Finally, an example of implementation is given, and conclusions are drawn.

1. **INTRODUCTION**

"The Internet of Things (IoT) describes the network of physical objects, "things", that are embedded with sensors, software, and other technologies to connect and exchange data with other devices and systems over the internet" [1]. The Internet of Things was born from the idea of bringing the objects of our daily experience into the digital world. The coining of the phrase "Internet of Things" is attributed to Kevin Ashton [2]. He used this term in 1999 during a presentation while working for Procter & Gamble (P&G). At that time, the context in which the "Internet of Things", or IoT, for short, was used was that of using RFID (Radio-Frequency Identification) in the supply chain of P&G. The novelty came from linking this technology with the Internet, which was still a relatively new technology at that moment.

Kevin Ashton understood that because all the information in the knowledge base available on the Internet is currently entered by human users this may be a limit in the future related to the inability of human actors to enter huge amounts of data in a precise way.

In Kevin Ashton's view, people should not be the only vector for entering information on the Internet. If devices, such as sensors or video cameras, capable of collecting data from their environment also have an Internet connection, they could make the collected data available on the Internet, thus providing accurate information about entities in their environment, such as objects or processes.

Moreover, these entities may have associated actuators, *i.e.*, devices acting upon the environment, themselves connected to the Internet. As we put together the concepts of gathering data from the environment, making it available for access, and acting upon the surroundings based on it with limited or without human intervention, we can get automation and even a cybernetic loop, as we can obtain a self-adapting system.

Cybernetics can be put into the discussion when the system we refer to contains a feedback loop. More precisely, when a system performs a change in the environment, acknowledges its impact, and then adapts accordingly to this feedback.

A simple example of implementing such a system could be an agricultural irrigation system in which a sensor is used to measure the humidity in the soil. The processing and control unit observes when the humidity drops below a predefined threshold and activates the water pumps. The humidity sensor continues to feed data to the control unit, which decides to stop the pumps once the value sent by the sensor is above a specific target. [3]

Besides sharing characteristics with an automaton and a cybernetic system, the Internet of Things paradigm brings us closer to an environment in which computing is pervasive, accessible anytime, anywhere, and becoming embedded in our surroundings.

The most prominent traits of IoT could be interconnectivity, heterogeneity, dynamical nature, scalability (usually at a big scale) [4], objects' unique identification, meaningful communication, and the ability to sense and act upon the environment based on intrinsic intelligence. An IoT infrastructure should be able to connect a wide array of devices in a dynamical and scalable manner. This network of devices should offer the system the ability to sense the exterior environment and act upon it if necessary. Decisions should be made by the collective wisdom of the system's components.

2. **INTERNET OF THINGS (IOT) ARCHITECTURES**

We can distinguish at least four main architectures for the Internet of Things ecosystems: the central point architecture, the decentralized architecture, the hybrid architecture, and the semi-centralized architecture.

2.1. CENTRAL POINT ARCHITECTURE

The first one is that of a device or of several devices that gather data from the environment by using different kinds of sensors. The data is then sent to a central endpoint, on-premises or in the cloud, for storing and processing.

To complete the loop, we can also have instructions sent from a central point to actuators deployed on the terrain.

In this architecture, the devices from the field have very limited autonomy and power of decision and communication. Their purpose is to send data to a central point and then execute the instructions received from there.

A device might be able to connect itself directly to the central point, or a gateway might be used. Usually, as the devices do not have the capabilities of communicating over the Internet, the gateway is used.

The gateway is commonly deployed near the field devices, and its role is to aggregate the data collected from the devices and send it to the central point. Of course, if the central point

[1] Faculty of Engineering in Foreign Languages, "Politehnica" University of Bucharest, Splaiul Independentei 313, Bucharest, 060042, Romania, E-mail: raduioancosmin@yahoo.com

wants to send a command to one of the devices, the communication passes through the gateway, which receives it and forwards it to the suitable device or group of devices.

One of the main disadvantages of this architecture is the single point of failure, to which the devices send the data, where the data is processed, and from where commands are sent to the devices. Also, the gateway introduces a point of failure, which can lead to losing connectivity to multiple devices if it fails.

By leveraging the edge computing trend, we can see that more processing power is shifted to the gateway device. This leads to the semi-centralized architecture, as we are going to see.

2.2. DECENTRALIZED ARCHITECTURE

Another type of architecture, the second one from our discussion, uses devices that are capable of dynamically joining or leaving a network of devices, knowing their neighbors, can communicate with them, and can even collaborate with them in achieving a task.

This architecture is decentralized. Each device can act on its own and as part of a community. The devices can act as a server and as a client at the same time, thus working in a peer-to-peer manner. The main advantage of this architecture is the lack of a single point of failure. Even communication failures can be handled as a message from a device could be sent via multiple pathways using a mesh network [5].

A device can join or leave dynamically a mesh network. The routing algorithms implemented by the mesh network can manage multiple routes and choose the best one to send the message. Also, a device doesn't need a direct connection to another device to communicate. For instance, if we consider three devices with wireless networking capabilities that are located such that the first device and the third one are in the range of the second device, communication can be established between the first device and the third one even if they are not directly visible to each other.

The main challenge of a completely decentralized architecture would be the setup and update of the devices, as they are not in connection with a central point. However, an updated device could be introduced in a community and have it spread the update to the other members. As more and more devices get updated, they can propagate the new setup further and further in a cascading manner, making the process very quick. Broadcast or multicast network strategies could be used for this.

2.3. HYBRID ARCHITECTURE

The third type of architecture is the hybrid one, obtained using a decentralized architecture and adding a central command point. By using this architecture, we can still retain reasonable and direct control of the smart objects, all while gracefully handling potential issues such as loss of communication.

2.4. SEMI-CENTRALIZED ARCHITECTURE

This is like fog computing, a term coined by Cisco. Another term used to describe it is edge computing.

This architecture results from taking a central point architecture and introducing several computing nodes between the central point and the devices on the field.

As previously stated, processing power is shifted to the gateway device; other functions are attributed to it, such as performing some analysis on the received data, making decisions based on the results, and sending commands to the devices without going through the central point. In this way, the gateway device becomes an edge computing device. In some cases, the edge device can even continue operating while being disconnected entirely from the central point.

Even though the edge device has some computing power, it still depends on the central point from which it receives instructions and updates.

An example of a scenario would be having the edge device run an application, let’s say a neural network model used for classifying the objects detected by a video camera, the field device in this case. What is important to note here is that the central point would be the one performing the training of the neural network, and then the resulting model would be deployed to the edge device.

During operation, the edge device would still be able to classify the objects even if the connection with the central point is lost. Also, there would not be any need to send a video stream to the central point as the edge device could only send an alert if a specific object is detected.

Using such a setup, the compute-expensive operation of training the neural network would still be performed rapidly on a capable central point. The edge device would only need enough power to run the trained model. There is no need to provide the computing power necessary for training a neural network to the edge device, as this would be an operation that is not frequently performed.

In this category, we could also fit the mist computing paradigm. With mist computing, the processing is done on the field device itself and not on an intermediary device, such as an edge device. After this stage, the processing of the raw data, the device can send the processed data to the cloud.

In conclusion, we recognize:

- central point architecture (like a classical client-server deployment)
- decentralized architecture (like peer-to-peer architecture)
- hybrid architecture (a decentralized architecture with devices that can self-organize and run independently, to this architecture, we also add a central point of command)
- semi-centralized architecture (like fog, edge, and mist computing)

Each of these architectures uses different shapes for a smart object:

- one entity representing a cluster of devices capable of computing, storage, communication, sensing the environment, or acting on the environment (this seems to be the case for the decentralized architecture and its variants, hybrid, and semi-centralized architectures)
- a distributed, abstract entity, with these capabilities distributed along the architecture (this seems to be the case for the central point architecture)

The smart object is composed of the hardware plus the associated software, and we will give more clarity to its definition further on.

3. **SMART OBJECTS**

In the literature, we speak about the “Internet of Things”, this is the consecrated phrasing. However, we say that the Internet of Things comprises smart objects. According to the Cambridge Dictionary [6], we have the following definitions for a “thing” and an “object”:

- thing – used to refer in an approximate way to an

object or to avoid naming it.
– object – a thing that you can see or touch but that is not usually a living animal, plant, or person.

We accept a semantic similarity between a "thing" and an "object" as they are used in one another's definitions. Still, the situation gets even more confusing, as in our case, a "thing" is the equivalent of a "smart object".

To gain more clarity, we should discuss what makes an object smart and how it differentiates from an object that is not smart, a simple object. According to some of the literature [7], the leap between an object and a smart object is performed when the object is given the ability to handle the information it receives from the environment.

Unfortunately, this is quite a broad definition that leaves room for a lot of interpretation. For instance, we can consider a simple rock to have an interface, its surface. The way it handles an input, represented, for example, by a force applied to this surface, an input that could generate, for instance, movement, could be considered as its ability to manage the received information from the environment. Thus, the need for more explicit criteria to differentiate between a simple object and a smart object arises.

A suitable framework for defining a smart object could include three dimensions: awareness, representation, and interaction [8]. According to this framework, a smart object should be aware of its environment, have an internal representation of what it can sense, and decide to interact with the environment based on the inputs it receives.

This definition could be expanded by adding more dimensions, such as the ability of a smart object to interact with its peers and other smart objects and collaborate with them to achieve a goal. A smart object encompasses computing, storage (whether short-term or volatile), communication, sensing, and acting on the environment.

We can also have distributed smart objects with storage and computing capabilities separated from the sensing and acting capabilities.

Until now, as we tried to define and characterize a smart object, we have mainly discussed its ability to sense its surroundings, make decisions based on perceptions and on internal representation, and act upon them.

If we take a simplistic approach, we can conclude that a smart object is not very different from an automaton. Adding the ability of a smart object to dynamically change its reaction by introducing a form of a feedback loop brings it closer to a cybernetic system. Still, there does not seem to be enough differentiation between a smart object and previously mentioned concepts of automation and cybernetic system.

The true smartness of a smart object should lie in its capability to communicate with other smart objects and collaborate to achieve a goal. To achieve this, the smart object should be capable of dynamically joining or leaving a community, detecting the members of a community, their capabilities, and their eventual intentions, as well as advertising its capabilities and maybe intentions to the community.

4. IMPLEMENTATION EXAMPLE

We need to start and define the criteria for how to choose architecture. To define the criteria, we could look at the high cohesion – low coupling principle and apply it accordingly. We must define the use cases and how they can be conceptually defined. To derive the use cases, we can ask the 5W and 1H questions: Who? What?, When?, Where?, Why?, How?. This will help us determine the business actors and agents, the business processes, the business rules, and the business objects.

We must look at the business use cases and derive the IoT technical use cases. For each technical use case, we need to see how it can be implemented, the needed functionalities for the implementation, and then see what kind of smart object (single entity smart object or distributed smart object) is needed for each one of the functionalities. The smart objects will be assigned digital twins, avatars, in our conceptualized world.

After this, we can think about what kind of architecture would suit the implementation. A close-to-ideal smart object should have capabilities such as:
– compute power
– storage
– communication interfaces
– sensing the environment
– acting on the environment.

With this in mind, the following proposition of such a smart object was created.

As previously discussed, the smart object comprises both the hardware and the software that makes it "smart".

From the hardware perspective, our device has processing power, storage capabilities, sensors, actuators, and networking interfaces.

From a software perspective, we follow the microservice architecture [9], and we implemented the following components (Fig. 1):
– an advertising service that communicates what are the capabilities of the device
– a request-response service that can both send requests to other devices asking them to perform some operations and respond to an incoming request from other devices
– a bridging service that is used for implementing a mesh network
– a key generator used for creating keys for the communities of devices
– a key vault for storing the community keys
– a sensor log for storing the historical sensor readings

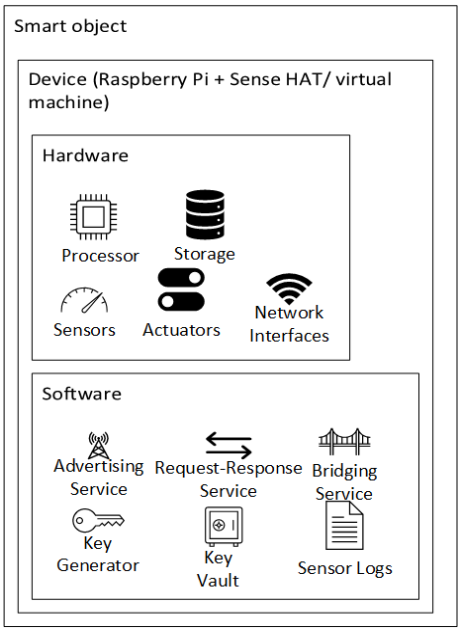

Fig. 1 – Schematics of a smart object implementation. The operating system of the device is Raspbian.

The sensing capabilities are offered by a Sense HAT (hardware attached on top) which plugs in as an add-on board in the Raspberry Pi. This board can also be simulated by software within the Raspbian operating system. The Sense HAT offers sensors (gyroscope, accelerometer, magnetometer, temperature, barometric pressure, humidity), input capabilities (by a five-button joystick), and an 8×8 RGB LED matrix, which in our case will be the way it interacts with the surrounding medium, by displaying a message.

The Sense HAT capabilities can be accessed by using Python. The rest of the features were implemented by using Java.

Each device is a server and a client, enabling a peer-to-peer architecture. The device is advertising its capabilities, such as the sensor readings it might be able to provide, alert messages it can display on the screen, its location, its name, and the IP address at which it can be contacted.

This advertisement can be done using broadcast or multicast in case we want to have a more targeted approach and save networking bandwidth, for example.

The protocol used for sending these messages is UDP (User Datagram Protocol), as no confirmation is expected if a particular advertisement was received or not. In contrast, messages containing requests from other devices and data such as sensor readings are sent using TCP (Transmission Control Protocol) to ensure the delivery of the message.

These advertisement messages can be sent in plain text to be accessible to all interested parties. For enhanced security, the messages can be encrypted using a shared secret in the form of a symmetric key. This symmetric key is what identifies a device as being part of a community, and it is pre-shared when the devices are configured. The secret is shared by an authority. It could be the user who configures it, for example. Like this, only the members of a community can decrypt the messages.

In this way, each community of devices speaks its language, ensuring security and encryption.

Each device includes a symmetric key generator to be the initial starting point of a community of devices. A device and be part of one or more communities of devices, as depicted in Fig. 2.

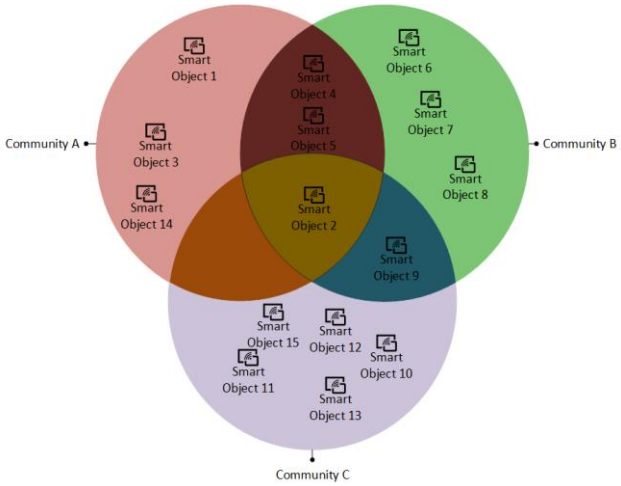

Fig. 2 – Communities of smart objects. Some smart objects can be part of several communities.

A device can then pick up what capabilities are available in its environment and request the devices it can access. Besides this, each device can store historic data from its sensors and offer this data to requestors or perform different local analyses.

From a connectivity point of view, each device bridges all its communications ports and is configured with a single IP address, which becomes how another device can contact it. This bridging configuration also allows the messages to be forwarded between the interfaces and for messages from other devices to be forwarded to their destinations if the device itself is not the intended destination.

The device acts as a switch. In this way, we can ensure that a device that does not have a direct line of sight, a direct connection, to another device can still communicate with it if there is a path to that device by leveraging the connections of the community. To avoid problems raised by the possibility of multiple active paths, STP (spanning tree protocol) is used. This also ensures that if a path is disconnected, another one will become active in its place if it exists. Such a configuration enables the possibility of using complete or partial mesh network topologies for redundancy in unreliable environments. It offers resiliency, and together with the peer-to-peer behavior of the devices, it can eliminate the existence of single points of failure in the architecture.

We can also have distributed devices. For example, having the storage and the compute capabilities separated from the sensing and acting capabilities. If we decide to use only a subset of a device's capabilities.

5. CONCLUSIONS

We introduced the concept of the Internet of Things and its evolution over the years and identified four relevant architectures for IoT. The concept of a smart object was characterized, and an implementation example was given.

By leveraging this multipurpose implementation, all the architectures discussed previously are possible: central point architecture, decentralized architecture, hybrid architecture, and semi-centralized architecture. Of course, for some of these deployments, only a subset of the device features might be used.

To conclude, the device, with all the connectivity capabilities and the operations enabled by software running on it, fit our definition of a smart object.

The Internet of Things paradigm has the potential to be applied to all activity domains in a way like what happened with automation and, later, with information technology.

IoT implementations can already be seen in diverse fields [10], ranging from industry, logistics, and agriculture to services, healthcare, and education.

For example, in healthcare, IoT implementations can be used for monitoring a patient's vital signs or her/his environment [11,12], while in education, a similar approach could be used to monitor a learner's reaction to different learning techniques to assess their effectiveness [13,14].

Received on 25 November 2021

REFERENCES

1. Oracle, https://www.oracle.com/ro/internet-of-things/what-is-iot/, Online, Accessed 23 November 2021.
2. K. Ashton, *That 'Internet of Things' thing in the real world, things matter more than ideas.*, https://www.rfidjournal.com/articles/view?4986, Online, Accessed 23 November 2021.
3. J. Birsan, D. Stavarache, M.I. Dascalu, I.B. Pavaloiu, A.M. Neagu Trocmaer, *Internet of Things in education: A case study for learning agriculture*, Journal eLearning & Software for Education, **2** (2017).
4. K.K. Patel, S.M. Patel, *Internet of Things-IoT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges*, International Journal of Engineering Science and Computing, **6**, 5, p. 6123 (2016).

5. Y. Zuo, Z. Ling, Y. Yuan, F.M.G. Tomescu, *A hybrid multi-path routing algorithm for industrial wireless mesh networks*, EURASIP Journal on Wireless Communications and Networking a SpringerOpen Journal, 2013.
6. Cambridge Dictionary, https://dictionary.cambridge.org/; Online, Accessed 19 August 2019.
7. C.G. García, D. Meana-Llorián, B. C. P. G-Bustelo, J. M. C. Lovelle, *A review about Smart Objects, Sensors, and Actuators*, J. Special Issue on Advances and Applications in the Internet of Things and Cloud Computing, p. 8 (2017).
8. G. Kortuem, F. Kawsar, V. Sundramoorthy, D. Fitton, *Smart objects as building blocks for the internet of things*, J. USIR, 2009, p. 31.
9. S. Stoja, S. Vukmirovic, N. Dalcekovic, D. Capko, B. Jelacic, *Accelerating performance in critical topology analysis of distribution management system process by switching from monolithic to microservices*, Rev. Roum. Sci. Techn. – Électrotechn. et Énerg., **63**, *3*, pp. 338–343 (2018).
10. P. Ray, *A survey on Internet of Things architectures*, Journal of King Saud University – Computer and Information Sciences, 2016.
11. A. Vasilateanu, I. C. Radu, A. Buga *Environment crowd-sensing for asthma management*, The 5th IEEE International Conference on E-Health and Bioengineering (EHB 2015), Iasi, 2015.
12. A. Voina, A. Topor, G. Alecu, C. Voina, F. Babarada, D. Manuc, *Integrated sensors networks into an acquisition platform for the air quality monitoring*, Rev. Roum. Sci. Techn.– Électrotechn. et Énerg., **62**, *3*, pp. 305–310 (2017).
13. I.C. Radu, *Emergent technologies for learning analytics*, The 13th International Scientific Conference eLearning and Software for Education, Bucharest, 2017.
14. M.I. Dascalu, B. Tesila, C.-N. Bodea, C. Mustata, A. Gheorghiu, I. C. Radu, *A big data analytics tool for social learning management systems*, The 14th International Scientific Conference eLearning and Software for Education, Bucharest, 2018.