PRELIMINARY COMMUNICATION

Annals of Disaster Risk Sciences

Vol 3, No 1 (2020): Special issue on cyber-security of critical infrastructure

IoT Platform for Personal Data Protection

George Suciu, BEIA Consult International
Cristiana Istrate, BEIA Consult International
Mari-Anais Sachian, BEIA Consult International
Oana Chenaru, Societatea de Inginerie Sisteme
Gheorghe Florea, Societatea de Inginerie Sisteme

Address for correspondence: Cristiana Istrate, BEIA Consult International, Romania, e-mail: cristiana.istrate@beia.ro

Abstract

Since the establishment of IoT (Internet of Things), a variety of end devices become interconnected with one another, and thus, new types of security challenges appeared which have to be taken care of. Personal data, at the moment, have a higher risk of being hacked by various types of cyberattacks, as a result of the abundance of connectivity in the cloud realm. To face this type of challenges, the European Union decided to implement in 2018 the GDPR (General Data Protection Regulation) that implies that personal data of any kind can be shared with a third party only with their accord and can be, as well, deleted by them, whenever they desire. Henceforth, this paper introduces the PARFAIT project that will take into account this regulation and will integrate a platform with the purpose of protecting the personal data in IoT based applications, especially for smart home, smart office and smart hotel use cases.

Keywords

Cyber-security, IoT, GDPR, Smart home, Smart office, Personal data, Cloud

1. Introduction

The Internet of Things (IoT) (Ali, 2018) represents a model surfaced from the extensive advances in information and communication technology (ICT). The comprehensive IoT framework includes a system of objects or devices like sensors to be handled, embedded processors and radio frequency identification (RFID) labels, supplied to the IoT gateway and the server. One of the greatest challenges in data protection is by maintaining secure people's privacy. That is why the General Data Protection Regulation (GDPR) has been applied in all of the European Union states. It is based on transparency and accountability, but the core of GDPR is data protection. The main principles of the GDPR are fairness, lawfulness, transparency, purpose limitation, data minimization, data accuracy, storage limitation, and integrity (Goddard, 2017). The following principles, lawfulness, fairness, and transparency, describe data controllers' obligations to have legitimate grounds for the processing of personal data. The purpose limitation refers to the obligation of data controllers to only use the collected data for specific and well-defined purposes. Data minimization represents keeping the required personal data for the specified purposes and periodically reviewing and erasing it if it is not needed. Storage limitation represents keeping the personal data for no longer than the needed period, following policy with standard retention periods, according to the documentation obligations.

Taking into consideration this regulation, PARFAIT (Personal dAta pRotection FrAmework for IoT) project aims to integrate a platform with the purpose of protecting personal data in IoT based applications, especially for smart home, smart office and smart hotel use cases.

The paper is structured in sections. Section 2 contains examples of previous research concerning the privacy of personal data in IoT based applications. In section 3 are presented the design and the components of the PARFAIT platform. Section 4 provides the results of the undertaken tests for the further implementation of the platform. Finally, section 5 provides a summary of the project and elaborates the future work.

2. Related Work

The Internet of Things (IoT) depicts the interconnection of uniquely identifiable integrated devices to the existing Internet infrastructure. The IoT is also a concern regarding the connection of physical devices (cars, thermostats, smartphones, home lighting, tide sensors, smart meters) to the Internet. The IoT creates with its many security challenges, thus emerging new challenges when it comes to personal data protection, GDPR compliance, cyber-physical attacks, and the malicious intentions of third-party attackers to manipulate private data (Park, 2016).

IoT, in the last few years, had enormous growth; therefore, the integration of devices with IoT increased exponentially. Thus issues regarding the data protection of what is being sent from one end-point to another one without the consent of the user —henceforth emerging the necessity of new protection methods against security breaches in the IoT field and regulations so the processed data will not be used for malicious purposes (Koutli, 2019).

To secure at a European level the personal data, the General Data Protection Regulation was implemented in May 2018, which is a significant update to the data privacy regulations from 1995. This to the increase of practice to process personal data from cloud platforms, social media without the consent of the owner of the corresponding data (Truong, 2020) (Badii, 2020).

To ensure protection over personal data, new GDPR compliant platforms are emerging lately and one of them would be the Amazon AWS IoT (Amazon Web Services). The main advantages of this platform are the fact that it lets smart devices connect with the AWS Cloud and, at the same time, interact with one another. Also, it uses a wide variety of proprietary services like Amazon DynamoDB (database), Amazon S3, simple storage service, and Amazon Machine Learning. This platform resides in a cloud solution that uses protocols, such as MQTT and HTTP1.1 (Hyper Transfer Protocol) and a Websocket for its data shadowing system. The main security characteristics would be the authentication at the device level via X.509 certificate and secure communication of traffic over SSL/TLS.

Microsoft Azure IoT Suite is another GDPR based platform having in its management the identity and authentication of devices, which are registered and they can connect via AMQP (Advanced Message Queuing Protocol), MQTT, and HTTP protocols.

The examples mentioned above are one of many platforms having the capacity to be GDPR compliant. Mainly speaking the Snap4City platform ensures GDPR requirements and it can create IoT solutions for organizations (cities, infrastructures) , manipulate open and private data with IoT and IoE devices respecting GDPR, and create/use processes based on IoT dashboards (Badii, 2020).

3. The architecture of the solution

In this section we present the components, protocols and architecture of the personal data protection framework for IoT interoperability.

3.1 The MQTT Protocol

Message Queuing Telemetry Transport (MQTT) is a protocol (Rouse, 2020) that is used in IoT and for machine-to-machine (M2M) communication. When using this protocol, the publisher sends data to an MQTT message broker.

The broker forwards data to clients which have previously subscribed to certain topics. A topic is represented by a file path, and only the subscribers which show interest in a topic will receive data related to it. The MQTT Protocol is continuously being improved, and it is considered to play an essential role in various applications, as presented in Figure 1.

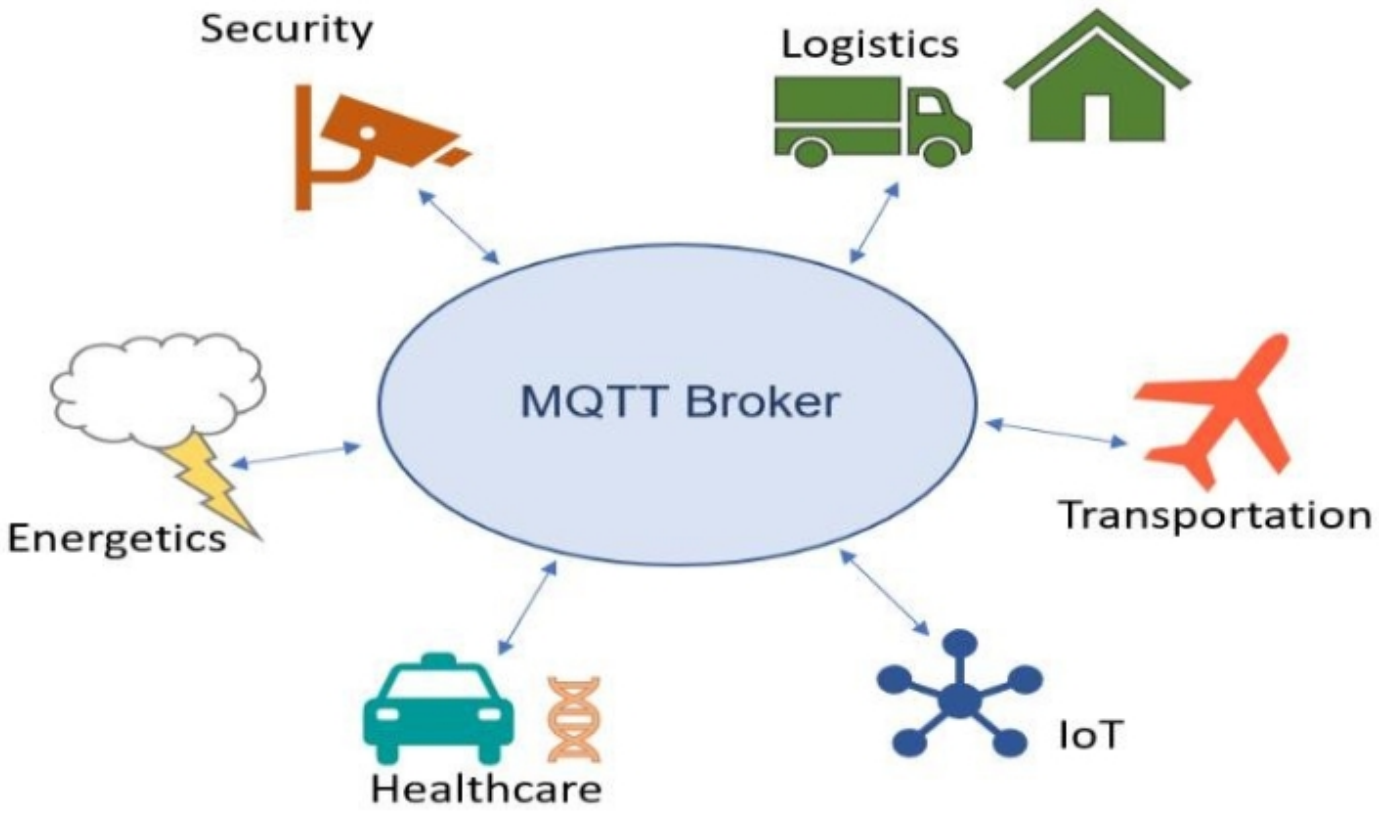

Figure 1. MQTT Applications

One of the most important aspects when using an MQTT broker is related to the situation in which the connection to the subscribers is interrupted. In this case, the broker will buffer all the messages which had not been sent to clients and will resend data when possible. The messages sent through MQTT consist of:

- A header (which has 2 bytes);
- An optional header;
- A message payload (of maximum of 256 MB);
- A QoS (Quality of Service) level - out of three possible;

When the first QoS level is implemented, the broker receives the message from the publisher and forwards it to the subscriber. None of the entities knows if the message has been correctly received, and after passing the message to the client, the broker does not save the information. The second QoS uses a forwarding checker mechanism, both between the publisher and the broker and between the broker and subscriber. If any failure is detected when sending data, the messages are being resent. After a new attempt of sending the message, if it is not received in a timely manner, it is sent again. As a result, the broker or the subscriber could receive the same packet multiple times.

3.2 The Publish-Subscribe Pattern

The publish-subscribe pattern (Using the Publish-Subscribe Model for Applications, 2019) refers to a messaging system that consists of message senders (publishers) and message receivers (subscribers). The publishers do not

send data to specific subscribers but organize the information into classes. The subscribers (clients) show interest in any data classes and only receive specific data. Neither the publishers or the subscribers know data related to the existence of the other entity. Data filtering can be topic-based (the publisher is responsible for organizing data in classes to which the receiver subscribes), or content-based (the subscriber is responsible for data classification, and it only receives data which matches the constraints defined by the subscriber itself). Other cases have a different implementation, in which the publishers send data to one or more topics, and the receivers subscribe to one or more topics they have interest in. Some systems use a broker, which is responsible for data filtering. It calls functions for storing and forwarding the information from publishers to subscribers and may prioritize messages in queues.

3.3 The OPC Protocol

The open platform communications (OPC) bolsters connectivity for automation and supervision infrastructures. It comprises the original OPC protocol and the unified architecture (UA) (Vrignat, 2018). OPC provides a technology to bolster heterogeneity and interoperability in control and automation applications, mainly addressing industrial manufacturing. This standard allows the scalability of the infrastructure and assures later expansions from diverse software/hardware entities. Figure 2 presents the general layout of communication utilizing the standard OPC protocol. The software applications play the role of data consumers and the hardware devices act as data sources, whereas the OPC interface acts as connectivity middleware, assuring the data flow. Beyond the main industrial application, OPC technology also extends its application to environments like educational systems, energy automation, building automation, virtualized environments.

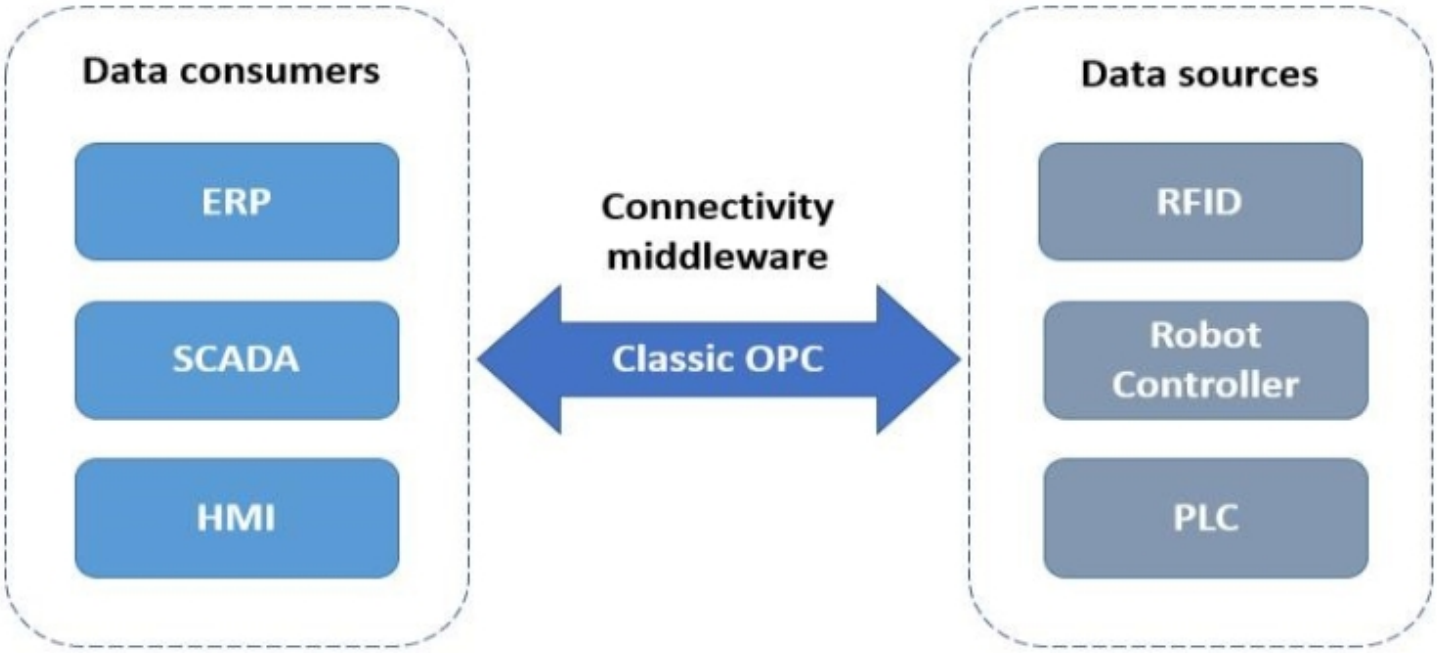

Figure 2. Classic OPC-based communication scheme in the automation system

3.4 PARFAIT architecture

The architecture consists in connecting the smart home information available through an MQTT or OPC connection to the PARFAIT infrastructure provided by Ericsson. Only these two interfaces were chosen at this point as MQTT is the most used IoT protocol and OPC is the most common process control protocol. These allows flexibility in integrating both new technologies and older or industry-specific elements. The main components are illustrated in Figure 3.

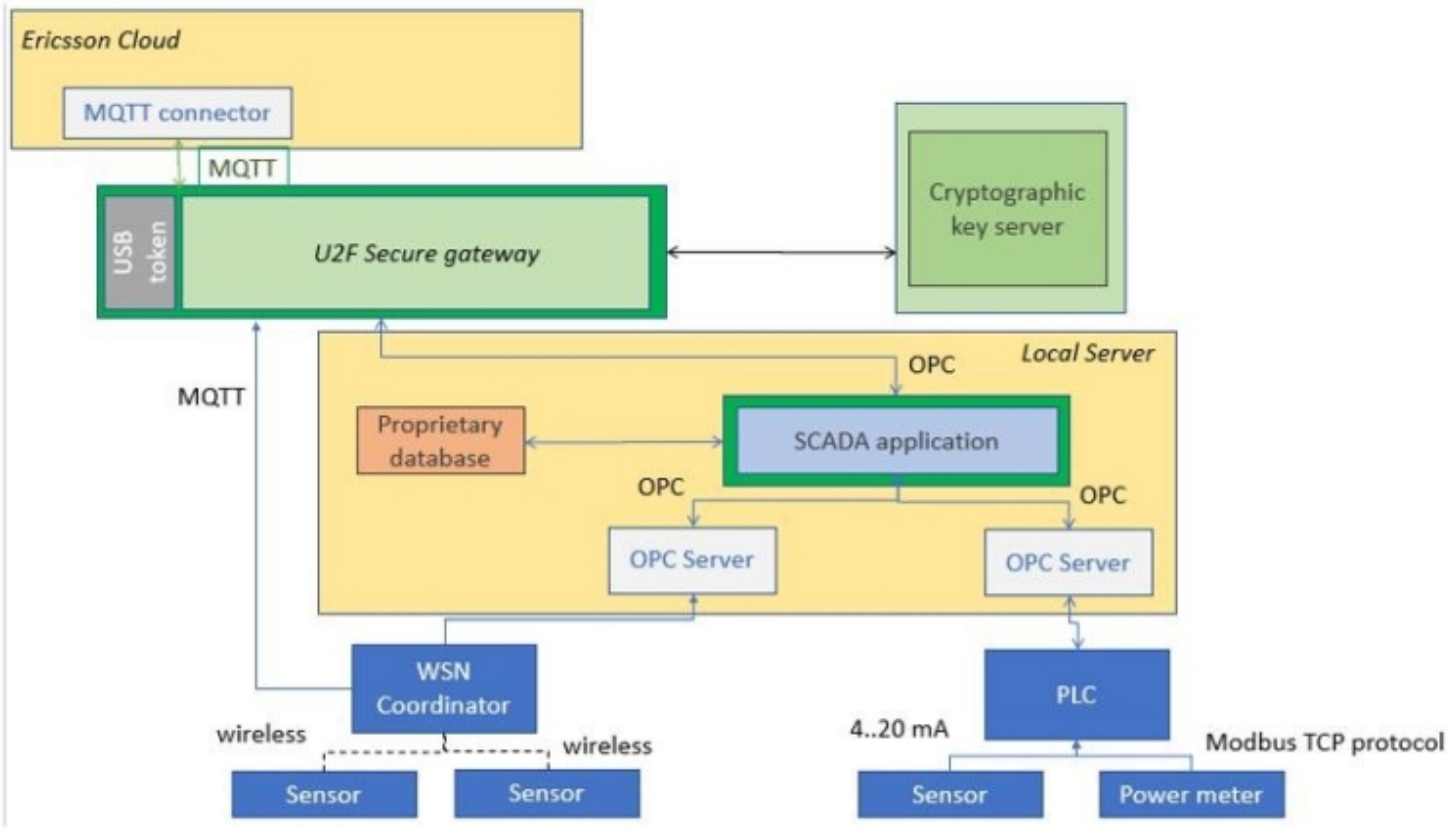

Figure 3. Smart Home use case architecture

Five processing levels were illustrated in this diagram:

1. *Local sensing and control elements*:
    a. IoT devices communicating over MQTT one or more measured parameters with a specific timestamp. MQTT is a lightweight publish-subscribe protocol running over TCP/IP (MQTT, 2019). A broker server routes messages from one or multiple clients using three types of messages: connect, for generating a connection between a client and a broker, disconnect, for detaching a client from a broker, and publish, for transmitting process data. A standard configuration in this case is a Wireless Sensor Network (WSN) gathering data from several wireless sensing devices through a WSN coordinator. This coordinator can store data locally or send it to a cloud platform through an MQTT interface;
    b. (optional) wired sensors communicating over standardized interfaces like 4..20 mA, 0..10V or industrial protocols. A Programmable Logic Controller (PLC) or embedded controller might be required to enable the physical interfaces required to collect this data or to implement control actions. This allows the possibility to control lighting, heating, ventilation and home devices according to a control strategy. Typical communication in this case involves the use of the OPC protocol (What is OPC?, 2020). OPC is a standard protocol interface which uses the client/server TCP model for accessing data from various hardware manufacturers. It is usually available as an OPC Server converting data from a specific hardware protocol, and a Windows application acting as OPC Client for collecting this data. This allows flawless integration with Supervisory Control and Data Acquisition (SCADA) applications, facilitating integration with standard process automation systems allowing control of lighting, heating, ventilation or home devices according to a control strategy smart home.

2. *The local SCADA server* (What is SCADA? , n.d.)**:** (optional, but commonly found in industrial applications) uses an application implementing communication drivers, data visualization, alarming, reporting and historical trending tools. It is used for facilitating the monitoring and control of automation systems, in any domain like smart home, oil and gas, energy, water management systems, manufacturing, production etc. A PC or a Human-Machine Interface (HMI) running a SCADA application and OPC interfaces allowing data

acquisition from a wide range of industrial protocols (Modbus, Profibus, BACnet etc.) which can be found in especially older smart home applications. This server usually allows local user interaction with the smart home for monitoring and control.

3. *The Secure Gateway*: device enabling data acquisition using MQTT or OPC and secure IoT data transmitting to the cloud. The secure gateway will implement the interconnection and trust functionality that needs to be achieved by limiting device registration through an U2F (Universal 2nd Factor) authentication procedure and transmitting encrypted data to the cloud.

4. *The Key server*: server managing a public and private key for device registration. This step is based on the Fast IDentity Online (FIDO) specification for IoT device authentication and provisioning, allowing automated and binding of applications and/or users to IoT devices. The aim of the FIDO Alliance is to provide means to improve user experience by simplifying the registration and login steps.

5. *The Ericsson cloud:* allows secure IoT data acquisition, visualization, and commands using MQTT. The platform registers a secure gateway by receiving its security key from the key server.

Implementing secure communication in such an architecture is limited by the security mechanisms available at the sensing level. This is because these usually represent off-the-shelf commercial devices that provide standard communication interfaces, usually with no implemented security mechanism like authentication or encryption. The communication between the WSN coordinator or SCADA server and the Secure Gateway is restricted by accepting only connections from registered devices, this step being available only after the user physical presence is certified. The secure gateway acts as a "trusted device" which is registered by the Ericsson cloud through the use of the security key. Public and private key pair is generated by an external cryptographic key server, and the private key can be imported in the gateway, market from this point as a secure element. This gateway enables the provisioning of data from IoT devices by implementing a scheme based on U2F while easing these operations at the user side. All further device setting modifications, migration to other applications or removing a registered device can be performed by the user through a web interface from the trusted device. The main components of the secure gateway are illustrated in Figure 4.

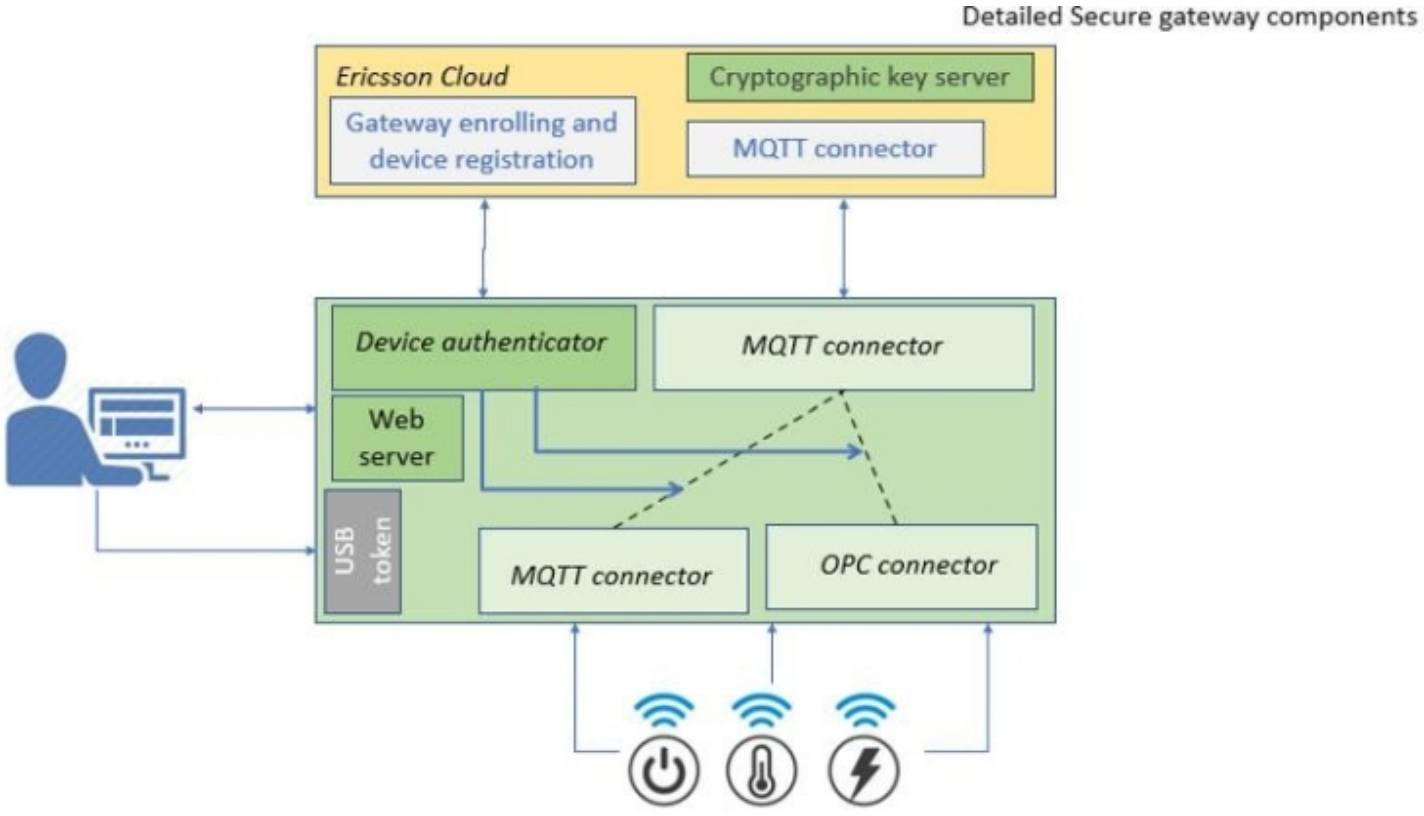

Figure 4. Main components of the secure gateway

Three types of actions are possible:

- Smart gateway enrolling (linking a smart gateway to a user, enabling access from a cloud server to multiple locations);
- Configuration actions (IoT device registration, settings update, etc.);
- Operation actions:
    - continuous transmitting data the cloud
    - data visualization and commands

The operation sequence starts with the *secure gateway enrollment* (with the U2F token plugged in) in the Ericsson cloud, using a local web configuration interface residing on the gateway.

This step handles linking each device to a certain cloud user, thus enabling access to data from multiple owned devices. During secure gateway enrollment, a binding process with the cloud server will be started. This will involve entering a gateway local web interface security information such as account name, password and IP address needed to connect to the Ericsson cloud. These credentials are owned by the smart gateway owner or smart home application provider, being conditioned by the access to the Ericsson cloud. After this information is validated, the cloud starts the enrollment process and the U2F USB device (U2F - FIDO Universal 2nd Factor authentication, 2020) will start to blink to validate user presence. After the button is pressed, a pair of the public and a private key are generated. And the public key and key handle of the U2F token will be assigned to the user's account.

During the *configuration stage***,** the user is able to register IoT/OPC devices in the gateway, delete registered devices or change his registration information on the Ericsson cloud. All registered devices will be available in the user area on the cloud. The user will initiate an IoT device registration and binding request or configuration update with the help of the secure gateway. This process will be initiated from the user interface available in the secure gateway and will involve mapping an input MQTT or OPC input interface data to an output MQTT interface. For this, he is able to search available OPC servers and select from available tags or add an MQTT device by listening on a specific port and select the devices connected to the gateway. The user will not have access to the cloud interface, all configuration taking place using the U2F web application of the secure device. To update the configuration the gateway will validate the configuration process with the cloud using its own key and the configuration of user presence through the USB token. For this, a message for request new configuration is sent to the cloud with the public key, the cloud checks gateway access rights and send back an acknowledgment; afterward a message including the device and data configuration is sent to the cloud. The cloud should check the connection with all devices in the configuration and should send "communication ok." The communication link remains active as long as it is not disabled or modified from the gateway interface. Deleting a device from the configuration can be done similarly. Any changes to the device settings will require the user presence validated by the USB token to be performed.

Once an IoT device is registered in the cloud, it will be able to *transmit data continuously*without further user or secure gateway actions (the link between the input and output interfaces will remain active). The user's presence is required for deleting a device registration.

Data visualization and manual commands can be performed from the user dashboard application residing on the Ericsson cloud. Received data is displayed in numerical and graphical format, for real-time or, respectively, historical trends visualization. The historical request will allow receiving the value of an IoT device at a particular past moment or for receiving buffered data over a predefined interval.

4. Implementation experimentations

In this section we present the results of the practical implementation using open source software and hardware.

4.1. Implementation of the U2F security mechanism

The secure gateway was implemented on a Raspberry Pi 3 controller, running a Ubuntu Mate operating system. It was used Feitian ePass Fido USB (ePass FIDO®-NFC Security Key - FIDO Alliance Certified Showcase, 2020) as a second-factor authentication device. This device has a press button for acknowledging user presence in case of registration, configuration or update actions. The setup also includes a laptop used as a key server generator linked over Ethernet with the Raspberry gateway. To implement this functionality, the server uses the python u2f server library.

For the gateway to be able to send data to the Ericsson cloud, the token must be plugged in the gateway at all times. The first step is to get a register challenge JSON blob. A challenge command, as illustrated in Figure 5, will get data from the U2F Server and store it in a .json file. An u2f-host command to register the device using the challenge.json file. The output data will be written in a register.json file, as shown in Figure 6. The device will start to blink and the user will need to press the button to acknowledge the registration action. After this, the server finalizes the registration by generating the public and a private key for that device, as presented in Figure 7.


```
{"challenge": "6l8aRM6f35hwrramrt7sKt7gDkvTamt2rYrMgMYE9ro", "version":
"U2F_V2", "appId": "http://localhost:8081"}
```


Figure 5. Challenge message example


```
{ "registrationData": "BQQOtd__bgnv8V6_T-E4914xE-
Pb6ji1YMUoP0LDLDCGtzCHPwbkMLlxlo6C6fawnQ7671o85nSbek9v0m3_fK7fQBLviOeAd
zHiknazlys7eXtC9DBraClKAhYO-
2SuxHnyFS9Jfk2nNrib1dtJJNcfRJrOBGILWIIlXzSt5xV4VBgwggIbMIIBBaADAgECAgRA
xBIlMAsGCSqGSIb3DQEBCzAuMSwwKgYDVQQDEyNZdWJpY28gVTJGIFJvb3QgQ0EgU2VyaWF
sIDQ1NzIwMDYzMTAgFw0xNDA4MDEwMDAwMDBaGA8yMDUwMDkwNDAwMDAwMFowKjEoMCYGA1
UEAwwfWXViaWNvIFUyRiBFRSBTZXJpYWwgMTA4NjU5MTUyNTBZMBMGByqGSM49AgEGCCqGS
M49AwEHA0IABK2iSVV7KGNEdPE-
oHGvobNnHVw6ZZ6vB3jNIYB1C4t32OucHzMweHqM5CAMSMDHtfp1vuJYaiQSk7jb6M48Wte
jEjAQMA4GCisGAQQBgsQKAQEEADALBgkqhkiG9w0BAQsDggEBAVg0BoEHEEp4LJLYPYFACR
GS8WZiXkCA8crYLgGnzvfKXwPwyKJlUzYxxv5xoRrl5zjkIUXhZ4mnHZVsnj9EY_VGDuRRz
KX7YtxTZpFZn7ej3abjLhckTkkQ_AhUkmP7VuK2AWLgYsS8ejGUqughBsKvh_84uxTAEr5B
S-OGg2yi7UIjd8W0nOCc6EN8d_8wCiPOjt2Y_-
TKpLLTXKszk4UnWNzRdxBThmBBprJBZbF1VyVRvJm5yRLBpth3G8KMvrt4Nu3Ecoj_Q154I
JpWe1Dp1upDFLOG9nWCRQk25Y264k9BDISfqs-
wHvUjIo2iDnKl5UVoauTWaT7M6KuEwl4wRAIgU5qU72pCVD-
```


Figure 6. Registration message example

```
$ curl http://localhost:8081/bind -d'data= { "registrationData":
"BQQOtd__bgnv8V6_T-E4914xE-
Pb6ji1YMUoP0LDLDCGtzCHPwbkML1xlo6C6fawnQ7671o85nSbek9v0m3_fK7fQBLviOeAd
zHiknazlys7eXtC9DBraC1KAhYO-
2SuxHnyFS9Jfk2nNrib1dtJJNcfRJrOBGILWIIlXzSt5xV4VBgwggIbMIIBBaADAgECAgRA
xBIlMAsGCSqGSIb3DQEBCzAuMSwwKgYDVQQDEyNZdWJpY28gVTJGIFJvb3QgQ0EgU2VyaWF
sIDQ1NzIwMDYzMTAgFw0xNDA4MDEwMDAwMDBaGA8yMDUwMDkwNDAwMDAwMFowKjEoMCYGA1
UEAwwfWXViaWNvIFUyRiBFRSBTZXJpYWwgMTA4NjU5MTUyNTBZMBMGByqGSM49AgEGCCqGS
M49AwEHA0IABK2iSVV7KGNEdPE-
oHGvobNnHVw6ZZ6vB3jNIYB1C4t32OucHzMweHqM5CAMSMDHtfp1vuJYaiQSk7jb6M48Wte
jEjAQMA4GCisGAQQBgsQKAQEEADALBgkqhkiG9w0BAQsDggEBAVg0BoEHEEp4LJLYPYFACR
GS8WZiXkCA8crYLgGnzvfKXwPwyKJlUzYxxv5xoRrl5zjkIUXhZ4mnHZVsnj9EY_VGDuRRz
KX7YtxTZpFZn7ej3abjLhckTkkQ_AhUkmP7VuK2AWLgYsS8ejGUqughBsKvh_84uxTAEr5B
S-OGg2yi7UIjd8W0nOCc6EN8d_8wCiPOjt2Y_-
TKpLLTXKszk4UnWNzRdxBThmBBprJBZbF1VyVRvJm5yRLBpth3G8KMvrt4Nu3Ecoj_Q154I
JpWe1Dp1upDFLOG9nWCRQk25Y264k9BDISfqs-
wHvUjIo2iDnKl5UVoauTWaT7M6KuEwl4wRAIgU5qU72pCVD-
bq68tETIKZ8aw7FRKviPVyFZc5Q8BlC0CICTc7_QuTWZFHwxGIotQO639WIllrPf1QqtvHC
yzzKg_", "clientData":
"eyAiY2hhbGxlbmdlIjogIjZsOGFSTTZmMzVod3JyYW1ydDdzS3Q3Z0RrdlRhbXQyclly TW
dNWUU5cm8iLCAib3JpZ2luIjogImh0dHA6XC9cL2RlbW8ueXViaWNvLmNvbSIsICJ0eXAiO
iAibmF2aWdhdG9yLmlkLmZpbmlzaEVucm9sbG1lbnQiIH0=" , "version" : "U2F_V2"
}'
```


Figure 7. Server message response example in case of a successful registration

When trying to change the device or data configuration on the gateway or start sending data to the cloud server, the user will need to authenticate. For this, he will initiate a challenge that includes a key handle. Afterward, the u2f-host command will be initiated to allow authentication, as displayed in Figure 8. The device should start to blink and will wait for the USB token to be pressed. The final command will be a JSON file including the device signature key and client key. The server will check this message and if the client key corresponds to the registered devices it will enable data connection for the device.


```
{ "keyHandle": "J7xZtI6AG7bQusZy8Q41-
WMqd8qaldZpcrqSASxYJnczvU30KTKWhvfTk4NuT0U4jr9WkD84u_OXo41oz69t6A"
,"signatureData":
"AQAAAAEwRQIgCkF19nsODvtXFhDinR97mVxFBn4rrhbQkC0R_HFQs8oCIQDvjPHBOr-
jAo7JTKEcFnmjbfHAZCAprjneOYLWm5n2Eg", "clientData":
"eyAiY2hhbGxlbmdlIjogIi15M3U4ZXRCNDN6M3VNbk9rODhFZzFYOUxhMkFYYTVNUUl1MC
1RQ2tlMmciLCAib3JpZ2luIjogImh0dHA6XC9cLzE3Mi4xMC4xMC4xNDM6ODA4MSIsICJ0e
XAiOiAibmF2aWdhdG9yLmlkLmdldEFzc2VydGlvbiIgfQ"}
```


Figure 8. Authentication request example

4.2. Integration with the Ericsson platform through MQTT

Ericsson provides a dashboard to monitor the published data. An example of a client code that publishes temperature and humidity data in a 10 seconds interval is run in order to demonstrate the connectivity with the Ericsson IoT Platform. The script is being run and the connection is established as presented in Figure 9.

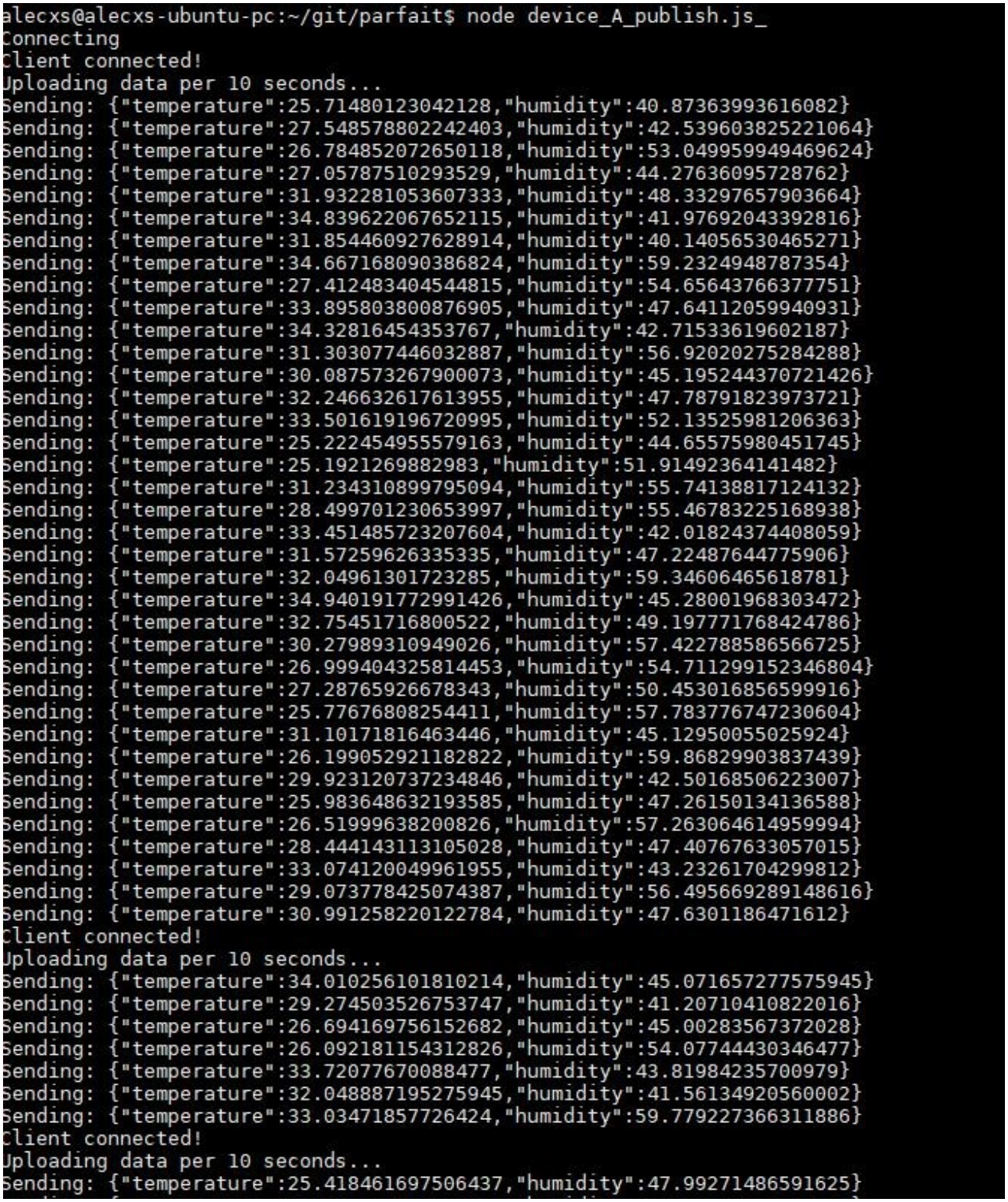

Figure 9. Connectivity establishment

It is verified in the web interface that the data exists. as shown in Figure 10.

Figure 10. Ericsson platform with the received data

Referring to Figure 11, the data can be seen with different granularities.

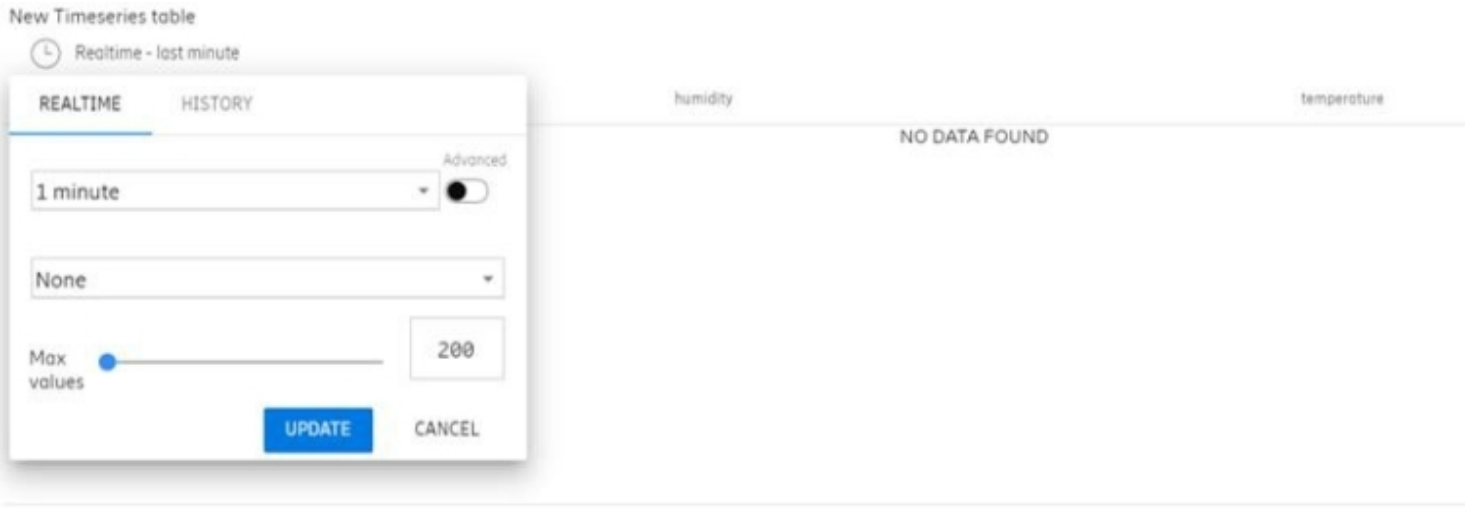

Figure 11. Ericsson platform with a different granularity selection

5. Conclusions and Future Work

The paper presents an architecture of the PARFAIT project, that is connecting the smart home data available through an OPC or MQTT connection to the infrastructure provided by Ericsson, along with the implementation and experimentation results, by focusing on the U2F security mechanism and the integration with the Ericsson platform through MQTT. The purpose of PARFAIT is to develop a platform for personal data protection in IoT based applications such as smart homes, smart offices and smart hotels. As future work, PARFAIT will integrate the U2F secure gateway with the Ericsson platform through MQTT, while demonstrating the users' data privacy.

Acknowledgment

This work has been supported in part by UEFISCDI Romania through ITEA3 program project PARFAIT, partially funded by the Operational Programme Human Capital of the Ministry of European Funds through the Financial Agreement 51675/09.07.2019, SMIS code 125125.

References

(n.d.). From http://internetofthingsagenda.techtarget.com/definition/MQTT-MQ-Telemetry-Transport

(n.d.). From https://docs.oracle.com/cd/B10501_01/appdev.920/a96590/adg15pub.htm

(n.d.). From What is SCADA? : https://inductiveautomation.com/resources/article/what-is-scada

(2019, December 27). From Using the Publish-Subscribe Model for Applications: https://docs.oracle.com/cd/B10501_01/appdev.920/a96590/adg15pub.htm

(2020). From What is OPC?: https://opcfoundation.org/about/what-is-opc/

(2020). From U2F - FIDO Universal 2nd Factor authentication: https://www.yubico.com/authentication-standards/fido-u2f/

(2020). From ePass FIDO®-NFC Security Key - FIDO Alliance Certified Showcase: https://fidoalliance.org/showcase/epass-fido-nfc-security-key/

Ali, B. &. Awad A.I. (2018). Cyber and physical security vulnerability assessment for IoT-based smart homes. *Sensors*, 817.

Badii, C. B. (2020). Smart City IoT Platform Respecting GDPR Privacy and Security Aspects. *IEEE Access*, 23601-23623.

Goddard, M. (2017). The EU General Data Protection Regulation (GDPR): European regulation that has a global impact. International Journal of Market Research. 703-705.

Koutli, M. T. (2019). Secure IoT e-Health applications using VICINITY framework and GDPR guidelines. . *15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 263-270.

MQTT. (2019). From http://mqtt.org/

Park, N. H. (2016). Security and Privacy Mechanisms for Sensor Middleware and Application in Internet of Things (IoT). *International Journal of Distributed SensorNetworks*.

Rouse, M. (2020, January 3). From internetofthingsagenda: http://internetofthingsagenda.techtarget.com/definition/MQTT-MQ-Telemetry-Transport

Truong, N. B. (2020). GDPR-Compliant Personal Data Management: A Blockchain-Based Solution. *IEEE Transactions on Information Forensics and Security* (pp. 1746–1761.). IEEE.

Vrignat, P. R. (2018). OPC UA: Examples of Digital Reporting Applications for Current Industrial Processes. . *DEStech Transactions on Engineering and Technology Research.*