Cloud Integration of Industrial IoT Systems. Architecture, Security Aspects and Sample Implementations

Katalin Ferencz[1,2], József Domokos[2], Levente Kovács[3]

[1]Doctoral School of Applied Informatics and Applied Mathematics, Óbuda University, Bécsi út 96/b, 1034 Budapest, Hungary, ferenczkatalin@stud.uni-obuda.hu

[2]Faculty of Technical and Human Sciences, Târgu-Mureș, Sapientia Hungarian University of Transylvania, Sighișoarei road no. 2, 540485 Târgu-Mureş/Corunca, Romania, {ferenczkatalin, domi}@ms.sapientia.ro

[3]John von Neumann Faculty of Informatics, Óbuda University, Bécsi út 96/b, 1034 Budapest, Hungary, kovacs@uni-obuda.hu

Abstract: Today's industry is increasingly characterized by the integration of Internet of Things (IoT) devices and the rapidly spreading digitization trend, which are also known as the foundations of Industry 4.0. The industrial revolution characterizes our everyday life, as a result of which the integration of smart devices is gaining more and more space in industry as well. As a result of the introduction of Industry 4.0, the implementation of the automation of production processes also becomes an important issue, as well as the continuous collection of data and their storage in the cloud. As a result of these integration, the method of collecting data, the usability of cloud-based systems, the feasibility of artificial intelligence-based data analysis, the visualization of massive amounts of collected data, cybersecurity issues, etc. have become everyday issues. In this article, we provide a detailed overview of a general IIoT (Industrial IoT) system, its various cloud service-based implementations, and present our open-source alternative solution. At the same time, we present the study of the security aspects and the integration of SOC into an IIoT system. Taking these aspects into account, we present two sample applications that try to offer ready to use solutions to certain problems of today's industry.

Keywords: Industry 4.0; Internet of Things; Cloud Infrastructure; Cybersecurity; Sensor data collection, Data Processing

1 Introduction

In the last decade, the state of the industry's digital development and the impact of the fourth industrial revolution on real production processes have become central topics in the field of industry. But what does this actually mean for industry and factories? We can put it simply that we can be everyday observers of the "smartening" of industry, that is, the development of intelligent industry.

With the beginning of the fourth industrial revolution, it has become very important for industry to successfully compete with the rapid development of technology by adapting autonomous, self-organizing resources, cyber-physical manufacturing systems, and real-time communication. To achieve this, it became necessary to introduce industrial IoT (Industrial Internet of Things - IIoT) devices, artificial intelligence (AI) based software, cloud-based systems (for example, databases), large amounts of data (Big Data) and the use of cybersecurity and automation. With the beginning of the fourth industrial revolution, industry is undergoing a major transformation. The main feature of the industry we have known so far was the use of rigid standards and rules, which provided companies with a high degree of security. The new industrial revolution is trying to transform these rigid norms and reform the existing production methods, thereby providing new opportunities for the integration of intelligent devices and innovative solutions. The fundamental element of the fourth industrial revolution is the intertwining of information technology, data and automation, that is, digitization [22].

The digitized industry, which can also be called a smart factory, is characterized by the fact that it uses intelligent devices in the processes of production, warehousing, and transport, through which it is able to collect large amounts of data in a short time (using various sensors or entire sensor networks), store and process. Emphasis must be placed on the fact that the storage of large amounts of data has concrete value if it is utilized, if this data is also processed and analysed, which can be realized by machine learning and artificial intelligence. Using the analysed data, we are able to create user interfaces and dashboards where we can continuously display and monitor, even in real-time, all the processes in the factory. It is possible to model and optimize the correct operation of the system, and at the same time, by analysing the data, system anomalies can be detected and even unwanted failures can be predicted. All of these have an impact on correct decision making, optimization of production processes, the utilization of machines can be increased, or we can even predict the service life of parts, which results in predictive maintenance [16].

Due to the IIoT, an important feature of today's smart factories is that the equipment continuously sends data to databases, communicates with controllers, software systems, and data analysis applications. This continuous multi-directional communication makes units compatible with industrial IoT devices an

integral part of intelligent processes. By using various IoT hubs, any type of device can transmit data to the cloud, database or any other unit connected for use. We can store, manage and process the locally collected data using private or public cloud systems in the fastest and most cost-effective way [15]. There are many providers on the market who can provide these services for a fee, such as Amazon, Microsoft, Google, etc., or it is possible to use a self-hosted system. Their architecture is usually very similar, but it can be implemented by using various technologies and software [5].

It is important to consider that the increasing spread of the digital development of the industry does not only generate positive results, so it is necessary to examine its negative side as well. Increased digitization also results in problems that have not been emphasized in the industry so far, specifically in the area of cyber-security. System security topics need to be treated as a priority, because of smart devices built into industrial systems can appear as attack points, and the use of cloud services and remote access can also be treated as potential vulnerabilities. When designing traditional industrial systems (for example, communication network), cybersecurity was mostly not a priority, as they were closed systems, the system was only available on the site, in the factory area, and was not available from the outside. With the spread of the industrial integration of IoT devices, the consideration of the appropriate security aspects must also play a role during planning, but for this the industrial cyber security teams must have the appropriate knowledge, which was not needed in the industry until now.

In the remaining parts of the paper, a general architecture is presented, which takes the collected data through the stages of storage, processing, analysis and display through 4 different technological implementations. We will present the three best-known and most used services on the market (Amazon Web Services, Microsoft Azure and Google Cloud Platform) as well as our own, open-source system, which also results in a system that conforms to the general architecture. Then, the security issues caused by the integration of IIoT devices will come into focus, as it will receive special attention for the industry to properly know its own devices, their parameters and behaviour, as well as to have a cybersecurity protocol or strategy to protect its entire system. In order to present such a system, we created and present a descriptive architecture that takes into account the basic security regulations. Finally, in order to prove the viability of the system, two sample applications using open-source implementations are also presented.

We would like to give a comprehensive picture of how the use of IIoT devices can be implemented and utilized for industry, and how it will be a key player in Industry 4.0.

2 Application of Cloud Systems in Industry

The IoT systems integrated in the industry we are examining can generally be described with the following self-architecture (Figure 1) and contain four important parts: data collection, data storage, data processing and execution.

Data collection consists of environmental data measured by various sensors, or by using data provided by other industrial machines through SCADA (Supervisory Control and Data Acquisition) Systems, PLCs (Programmable Logic Controller), DCSs (Distributed Control System) etc.

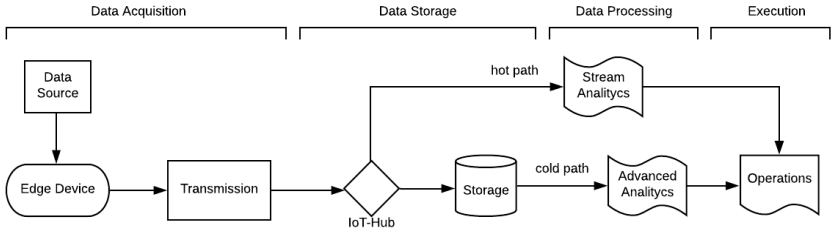

Figure 1
End-to-end data flow architecture in IIoT

The collected data is transmitted from the device to a higher-order unit using various network protocols, such as FieldBus, ControlBus, OPC (Open Platform Communications), Message Queue Telemetry Transport (MQTT), Advanced MQTT (AMQTT), HTTP, HTTPS, Constrained Application Protocol (CoAP), etc. Through network protocols, the data is transmitted to an IoT-Hub, which acts as a gateway between the device and the cloud.

Due to the versatile operation of the IoT-Hub, the incoming data is forwarded in several directions in accordance with the system requirements. The most common solution is to transfer these data to the database for immediate storage. It is possible to analyse these stored data at a later date using an algorithm based on artificial intelligence. This type of data processing is called the cold path. But as another solution, an IoT-Hub also provides the possibility of immediate data processing, which can provide real-time results, this is known as process analysis, that is, the hot path of data processing. As a result of, this type of analysis, the collected data can be used with the help of the obtained calculated data, for example, these data can provide valuable information. Based on these calculated data, it is possible to visualize data, create reports and alerts, and detect anomalies using various algorithms [2].

In the following, we will examine the general architecture described above in the case of Microsoft Azure, Amazon Web Services and Google Cloud Platform, as well as in the case of a system developed by us that uses open-source technologies. In each case, the system is built on the basis of the same

architecture, but with the help of technological equivalents provided by the given service provider.

2.1 Microsoft Azure

The figure below (Figure 2) shows the technological components provided by the Microsoft Azure service provider for the creation and deployment of an IIoT system.

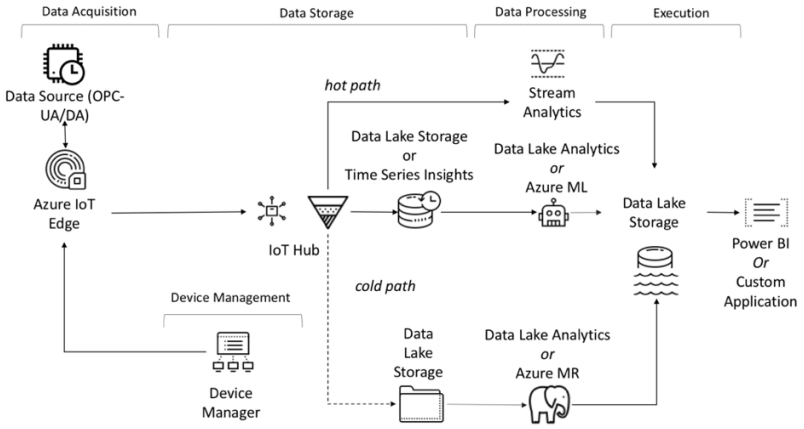

Figure 2
Azure IIoT architecture [20]

Data is collected from sensors and other industrial devices through Azure IoT Edge using various protocols such as, HTTP, AMQP, MQTT, OPC UA, etc. This forwards the data to Azure IoT Hub. The IoT-Hub is an intermediate layer used for device registration and management. From there, the data is processed by Stream Analytics, which forms the hot path of the analysis. Azure Stream Analytics is a real-time analytics and complex event processing engine designed to analyse and process large volumes of fast data streams simultaneously from multiple sources. The result of the analysis is stored in Data Lake Storage. It is possible to store this data in two types of databases, one is Time Series Insight and the other is Azure Data Lake Storage. Time Series Insight is suitable for storing, visualizing and querying large-scale time series data. Azure Data Lake Storage is a highly scalable data store that you can use to store data for later analysis and visualization. The stored data is cold path analysed by Data Lake Analytics or Azure ML (Machine Learning). Data Lake Analytics is a big data service for processing huge amounts of data. ML Analytics is an environment in which we can conduct advanced machine learning-based analytics for predictive modelling. Finally, Power BI or other custom applications can be used to display the data.

Power BI is a business analytics service that provides insights and tracks viewing real-world data.

2.2 Amazon Web Service (AWS)

Using the services provided by Amazon, the collected data is sent by Greengrass to the unit corresponding to the IoT-Hub, as shown in Figure 3. GreenGrass is software that extends the capabilities of the cloud to the local device and allows the device to collect and analyze data closer to the source of information.

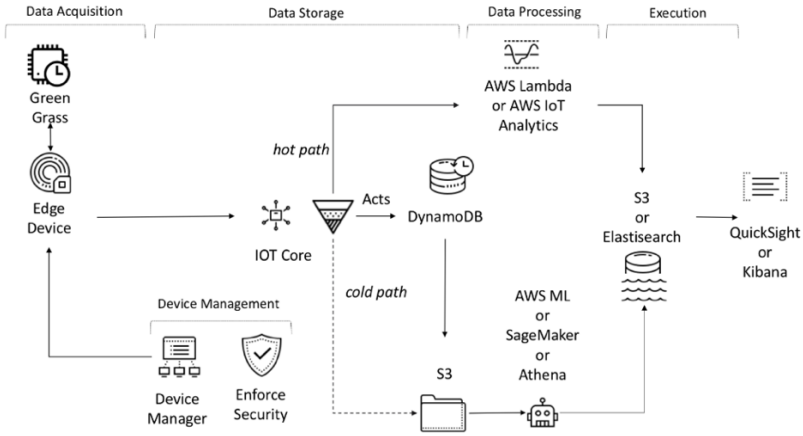

Figure 3
AWS IIoT architecture [30]

AWS IoT Core corresponds to the IoT-Hub, to which the devices send the collected data using various protocols, such as MQTT. There are several options for storing data here as well, depending on the type of data we want to store. DynamoDB can be used to store time series, or the S3 database can be used to store object data. In order to process the data quickly, we can use AWS Lambda serverless platform or AWS IoT Analytics. For more advanced analysis, AWS provides the user with several options: AWS ML, SageMaker, Athena. The results of the analyzes of fast and advanced algorithms are usually stored in S3 or Elastisearch type database storage. It is also possible to establish a direct connection to the QuickSight or Kibana visualization services that provide immediate visualization.

2.3 Google Cloud Platform (GCP)

The services provided by Google Cloud Platform allow the implementation of the architecture shown in the following figure (Figure 4). Google IoT Core supports

the MQTT and HTTPS protocols, so data can only be transmitted to the processing system via these protocols.

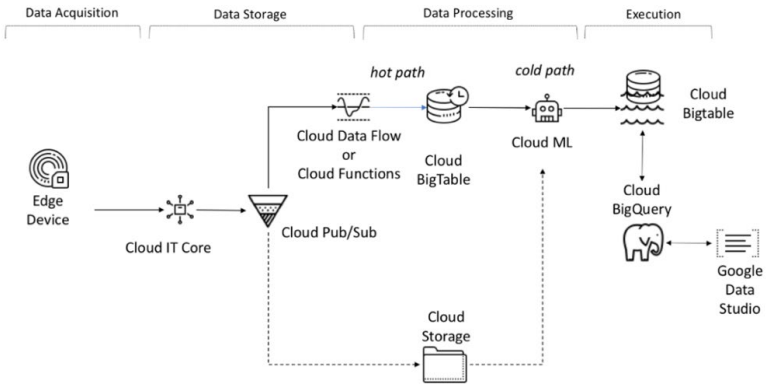

Figure 4
GCP IIoT architecture [20]

Google IoT Core corresponds to IoT-Hub, which acts as a central hub for connecting remote devices and uses the Google Pub/Sub service for data transfer. Cloud Data Flow or Cloud Functions service can be used to quickly process data and create Stream Analytics. Cloud Bigtable can be used to store time series or event type data, and Cloud Storage can be used to store object type data. Cloud BigQuery and Cloud ML services can be used for batch analysis of data. Google Data Studio is responsible for data visualization.

2.4 Open-Source Implementations

Considering that in many cases we are unable to subscribe to the above-mentioned service providers, or it is simply not allowed for the data to be managed by an external service provider, we also need systems that are built from our own components. In the following, we will examine the IIoT system designed and implemented by us using open-source components, which implements similar functionality to the systems presented above with a similar architecture. Figure 5 shows that the sensor data is provided, that is, in our case generated, by a Python program, which in reality can be replaced by any sensors. In this case, the generated sensor data (relative humidity, ambient pressure and temperature, pressure, and electrical energy) are generated based on the analysis of an existing data set, which is presented in detail in a previous article [12].

The data transfer was realized by an open-source implementation of the MQTT protocol called Mosquitto [9], which transmits the data to the Node-RED [21] framework. The Mosquitto implementation we use is only one of the many MQTT protocol implementations, as it is possible to choose from several options and

implementations as needed [32]. Node-RED acts as an open source IoT edge, which can fulfill several roles due to its versatile functionality. The data received by Node-RED can be analyzed and displayed immediately with the help of the Dashboard (User Interface) module using various built-in node diagrams (charts). This forms the so-called hot path of data processing.

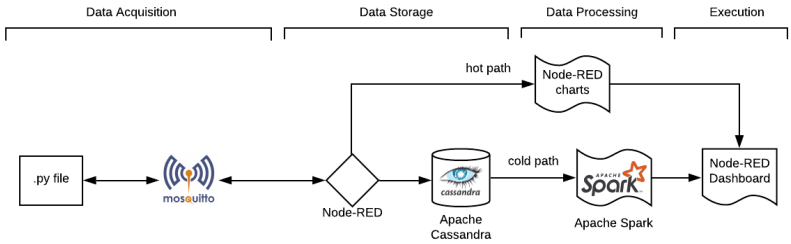

Figure 5
The architecture of the IIoT system developed by us using open-source components

The data is stored in a cluster built in the NoSQL type Apache Cassandra [3] open-source database system for post-processing and data analysis. In addition to the Apache Cassandra database, we can also choose MongoDB, Hbase, Redis, Couchbase or other database systems deemed suitable for the purpose [1]. The data processing can be realized by the open-source Apache Spark framework developed for analysis, which uses the Spark Machine Learning (ML) library [8], [10], [23]. A detailed description of the implemented system can be found in previously published conference papers [11], [12], [13]. Using the possibilities provided by the Spark Machine Learning libraries, we also performed data analysis using some ML Python-based libraries (pyspark.ml.features, pyspark.ml.regression, pyspark.ml.evaluation, etc.).

Node-RED and its accompanying Dashboard offer powerful tools for developing IoT solutions, but they do come with some limitations. One limitation is that Node-RED may require additional configuration and setup to work with certain hardware or protocols, which can be challenging for users with limited technical expertise. Additionally, while Node-RED provides a visual programming interface, complex applications may require extensive flows and can become difficult to manage and debug. As the Node-RED Dashboard offers basic data visualization capabilities it is possible that may not meet the advanced data visualization requirements of more complex industrial applications. Despite these limitations, Node-RED has found utility in industrial environments, where its ease of use, extensibility, and versatility make it a valuable tool for process automation, data acquisition and monitoring, predictive maintenance, and remote control of industrial systems [26], [4]. In this way Node-RED is being utilized in industrial settings to streamline operations, enable innovation in IoT solutions, and enhance productivity and automation [18].

2.5 Other Options for Implementation of Similar Systems

In addition to the globally widespread and well-known platforms (Azure, AWS, GCP) and open-source solutions presented above, many other targeted platforms and systems are available on the market.

Kuzzle, thethings.iO, and Thinger.io are three notable platforms that offer Internet of Things (IoT) solutions for developers, businesses, and industries. Kuzzle provides a flexible, open-source backend solution for building IoT applications with real-time communication and data synchronization features. Kuzzle's advantages include high scalability, real-time communication, and advanced security features, while its disadvantages may include a steeper learning curve and limited community support [20]. Thethings.iO offers a cloud-based Enterprise IoT platform for managing and visualizing data from connected devices, while Thinger.io provides an end-to-end IoT platform with cloud-based data management, device connectivity, and application development tools. Thethings.iO's (free and paid plans) advantages include easy integration with IoT devices, a user-friendly dashboard, and comprehensive analytics, while its disadvantages may include limited customization options and potential scalability challenges for large-scale deployments [27]. Thinger.io (free and paid plans) offers benefits such as robust security measures, comprehensive support for diverse IoT protocols, and a thriving user community, but it may pose challenges in terms of user onboarding and scalability for extensive IoT deployments [28].

In terms of data visualization, we already have many options and solutions on the market. Grafana [6] is a popular and versatile data visualization and monitoring platform that empowers users to create dynamic and interactive dashboards for real-time data analysis. With its rich set of features and customizable options, Grafana offers a powerful solution for visualizing complex data from various sources. However, like any other software tool, Grafana also has limitations, including a potential learning curve for beginners and the possibility of lacking some enterprise-grade features or support that certain organizations may require. Nevertheless, Grafana's strengths lie in its flexibility, ease of use, and ability to deliver actionable insights through visually appealing dashboards.

These platforms are widely used in various industries for developing IoT solutions, ranging from smart home and healthcare to logistics and industrial automation, offering powerful tools and features for building modern IoT applications.

3 Security Issues of IIoT Systems

In general, the industrial systems presented above must always be operational and in a constantly active, working state, so minor or major shutdowns caused by unexpected attacks are not allowed and should definitely be avoided. Taking all these into account, industrial companies integrating intelligent manufacturing methods realize that the proper knowledge and protection of their systems, as well as the analysis of their data, must be an area to be treated as an important priority [24].

In order to manage or even prevent the various attacks arising from the integration of the technological innovations of Industry 4.0, a cyber-security protocol or strategy and the development of a new kind of security approach are needed. In the management of these threats, the Security Operations Center (SOC) plays a prominent role, which monitors the devices around the clock and centralizes the security supervision and control of the devices. The application of SOC enables real-time threat detection and response to security incidents. In addition, with the help of SOC, we can provide rapid security detection in order to identify, investigate, prioritize and solve security problems. As a result, it is possible to detect threats early and alert the relevant security teams, who can fix security problems in a short time without interrupting production processes, thus reducing the impact of a future attack.

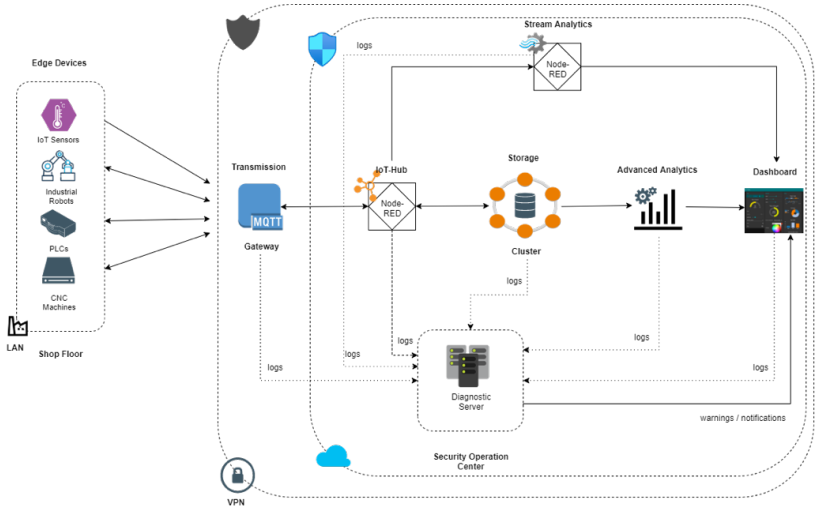

Figure 6
Proposed secure industrial system architecture (with SOC included)

An industrial environment can use a variety of devices such as IoT devices, microcontrollers, sensors, enterprise computers, internal networks, virtual environments, manufacturing equipment, etc. (Figure 6). An important operational

parameter of all these devices is that these can generate and transmit various type of data to a central unit. These data can be structured data and unstructured data, which can be analysed to obtain up-to-date system information about the state and operation of the entire system, which is essential to protect the system operation. The processing of the collected data is essential, since based on them, a model of the system's normal operation can be prepared, and based on this model, it is possible to filter out abnormalities and anomalies that can be detected in the system, as these can even result in system failure or be signs of unwanted interventions.

At the lowest (shop floor) of industrial systems are operational technologies, that is, OT (Operational Technology), whose role is to monitor and manage the tools of industrial processes. OT includes HMI (Human-Machine Interface), SCADA, PLC, pumps, sensors, robots, etc. Based on the operating principle of these devices integrated into the system, they can connect to the internal network of the factory environment. All kinds of communication are then carried out through these networks, that is, these devices can receive or transmit data and commands to higher decision making levels. From the shop floor, the data is transmitted to a more protected network (for example, using a Virtual Private Network) by several protocols (for example, MQTT, Fieldbus, Modebus, DNP3, Ethernet, Profinet,). Here, a unit serving as a gateway transmits this data to a hub, such as Node-RED can act as such an IoT-hub, or AWS IoT, or Azure IoT Hub, etc. This particular hub is able to transmit data in specified directions, so at this architectural level the data is transmitted to the database, which could be a SQL (for example, MSSQL - Microsoft SQL) or a NoSQL type of database built as a split cluster, such as the Apache Cassandra database system, which can be on-premises or can be rented as Software-as-a-Service. Depending on the quality and characteristics of the chosen storage method, it is possible to create advanced analytics by using algorithms based on artificial intelligence (the cold path of data processing). The IoT-Hub provides the opportunity to implement a real-time data processing, which is called Stream Analytics (the hot route of data processing). This analysis mode can provide results describing the current state of the system and useable information can be obtained based on the results of the analyses [17], [19].

3.1 Vulnerabilities of Industrial Systems

It is critical for any industrial system to know what features and vulnerabilities the integrated devices have. The importance of this lies in the fact that an attack in the absence of this knowledge can make the entire production or part of the production lines inoperable, which, depending on the industry, can even cause personal injury or property damage.

Systems designed before the spread of the fourth industrial revolution were designed in such a way that the entire system would be isolated from the “outside

world" (not connected to internet or any public network), so external attacks were not possible. It should be mentioned here that a prominent part of the devices in the OT now have a certain level of their own security system or protocol (blocking of ports, username-password, etc.), but most of them are switched off during commissioning in order to ensure that the system is put into operation and its operation should be implemented as soon as possible, such as not having to constantly use passwords when configuring the machines. Thus, as a result, these devices can become easy targets for malicious attackers, due to the aforementioned "negligence". Taking this into account, the appearance of smart devices in the field of industry may question the security of the system, due to the fact that IoT devices are devices that can be easily connected to the Internet, thus creating a critical vulnerable interface on the internal network (in many cases inadequately protected). Another possible point of vulnerability is incomplete testing or code review during the design of various software, which can also generate new risk factors, such as forgotten authentication data (username and password) in the source code.

Due to some of the previously presented incomplete security protections typical of OT devices, the IT team is responsible for ensuring adequate protection of the communication network, thus guaranteeing the protection of the entire system. Communication protocols link endpoints back to the shop floor or controller, making them the most vulnerable parts of industrial systems. As an additional attack surface, we can also consider the insecure Internet connection, which also entails unsafe remote access. These enable easy hacking of industrial systems, in the event that unsafe and protected protocols are used. Besides that, the appropriate behaviour of employees is also a critical and at the same time a very common safety issue. It is also important to treat this factor with high priority, since they can be connected to the machines in the factory with USB, laptops and other portable devices, and in the event that these devices were previously infected in an outside, insecure network, and then connected to the industrial system can cause serious damage to your machines.

3.2 Types of Attacks

In the field of industry, it is increasingly seen that the transformed manufacturing technologies increase the attack surfaces, the purpose of which is usually industrial espionage or sabotage. Typically, these targeted attacks can have complex consequences, such as economic loss, production loss, downtime, and machine damage or in the worst case, human injury. With these in mind, we must prepare risk management, so we can build a proactive protection system.

Systems using smart devices are most often hit by denial of service (DoS) attacks, access control attacks, authentication attacks, protocol and application integrity attacks [25].

Several categories of threats can be distinguished [31]:

- Data Leakage: Smart devices capture large amounts of data for analysis, monitoring, and logging purposes. In "closed" industrial systems, these data are transmitted without encryption via the internal network to the database designated for storage. If packet capture devices are placed in the immediate vicinity of these routes for malicious purposes, valuable operational information, usernames or even passwords can fall into unauthorized hands.
- Unauthorized access: If the default usernames and passwords of the software and machines are not changed (typically not), then as soon as the IoT device is connected to the Internet, it becomes an easily attacked device. In order to avoid this, the best protection protocol can be the use of strong authentication, even two-step settings.
- Denial of Service (DoS): This is one of the most common types of attacks in the digital world. However, a particular form of it is Distributed Denial of Services (DDoS), which works by essentially observing IoT endpoints that have been compromised and are easily accessible for some reason, and then creates a "botnet". It will receive and forward commands to the specified endpoints. As a result, an attacker can easily overload the internal network, causing the device intended to be used by the real user to become unresponsive or a complete network crash.
- Faked data output: In this category, the attacker managed to get into the system and, with the aim of creating a vulnerability, places foreign code in the software of a specific device so that the device (for example, a sensor) transmits false, manipulated data to the center, that is, implements false data entry. Furthermore, the attacker can create a “backdoor” that allows him to communicate remotely, so he can transmit additional commands to the selected devices.

Since the spread of the fourth industrial revolution, there have been many attacks on large industrial companies and plants, which should serve as a warning to everyone that special attention must be paid to the cybersecurity of our systems.

4 IoT Device Integration Solutions

Through two sample practical examples, we will show how, thanks to today's advanced technology (smart, IoT devices), we have the options to store the collected data on demand, to use different IIoT communication protocols (in our case MQTT), as well as to incorporate the functionalities provided by the increasingly widespread Node-RED (IBM development). We will provide insight into how, through innovative IT developments, the data of certain industrial systems can be stored, displayed and analysed with simple implementations.

The architecture of both systems is structured similarly, just like the general architecture presented in Chapter 2, and the "real-time" sensor data is provided by a Python code, the logic of which tries to simulate the operating principles of some sensors of the Combined Cycle Power Plant (CCPP) [7]. The data describing the operation will communicate with the Node-RED development environment by using the Mosquitto MQTT broker, which will act as an IoT-Hub. From there, the data continues its journey in two directions: it is entered into the Apache Cassandra database cluster and displayed in real-time using the Node-RED Dashboard. Also, on the Node-RED Dashboard, we provide the possibility to read, display and modify the values of the sensors, thus resulting in the use of two-way communication of the MQTT broker.

4.1 Sensor Data (Generic Plant Equipment) Collection and Monitoring Application

We created an application for monitoring a general factory machine and its environment, the architecture of which is shown in Figure 7. Using various simulated sensors, we record the data measured during operation of a generic machine (speed, revolutions, status, working hours, etc.) and the changes affecting its environment parameters (temperature, light conditions and humidity).

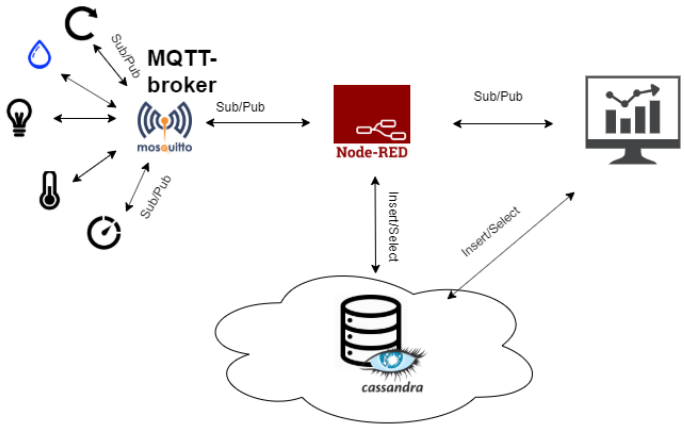

Figure 7
Block diagram of the sample system

These data are transmitted to Node-RED in real-time by a Mosquito MQTT broker and displayed on a graphical interface, as well as stored in a Cassandra NoSQL database. Considering the possibilities offered by the Node-RED Dashboard, we created an easy-to-understand user interface (Figure 8), as well as an associated logic, where we use the data, we consider it important in real-time, and we can also manipulate some selected parameters on the interface.

The possibility of changing the parameters is implemented on the Dashboard so that we can intervene in real-time, thereby simulating the real operation at some level. Reading and writing to the database is also done using the Node-RED platform.

With the help of a Python code, we create a simulation describing the operation of a general industrial machine and its environment, thus creating a sufficient amount of sensor data. Using this code, we can generate sensors and put them in a JSON (JavaScript Object Notation) object, as well as give each of them an initial value. Due to the data provided by the MQTT broker and the possibility that the code generating the sensor data not only transmits the data to the broker, but can also receive it, we can override the value as desired. We want to store this generated data and display it on the interface in real-time.

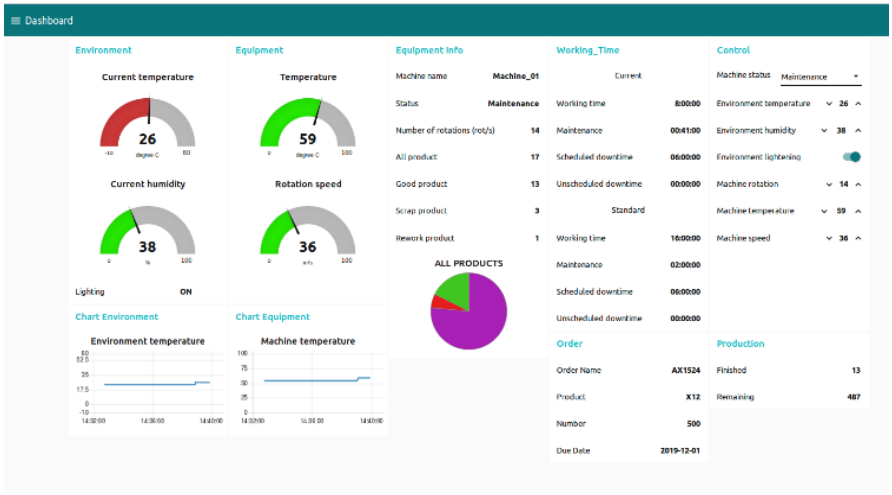

Figure 8
User Interface created using Node Red Dashboard

In the basic case, the sensors created by the simulation act as data senders that register on the MQTT broker. The interested parties will be the Apache Cassandra NoSQL database node and the Node-RED dashboard elements, so these will subscribe to the broker. These interested parties tell the MQTT broker exactly what data these are interested in. When we change the value of a parameter on the interface, it will already behave as a data sender. The above-mentioned Node-RED Dashboard can be created with the help of the dashboard elements in the node panel, which can be displayed on the interface by placing them in the flow panel. We have to configure these elements based on the information that defines the operation of the system, and if the operation of these elements is related to each other, we can connect them, so that they receive the previous output value as an input parameter. Writing program code is necessary if we want to process the

data of the Dashboard elements in the background (function node), or if we want to write them to a database, or if we want to perform any other operation with the data managed by the given element.

We use the NoSQL Apache Cassandra database to store all kinds of data. We chose Apache Cassandra because it is important for data from IoT devices that the storage system is scalable, always available and fault-tolerant. Also, with Cassandra, it is easy to implement storage and access in the cloud. Cassandra can store data on multiple nodes to provide enhanced security and high availability. Taking this into account, we built a 3-node system, ensuring that our data is not lost even in the event of a more critical shutdown. We were then able to display statistical information using the stored data.

In order to simulate real production, a simple Node-RED process can be used to create products with different ratings at regular intervals (every 10 minutes): defective, recyclable and good; thereby modelling a real production process.

4.2 Partial Simulation of the CCPP Operation

By analyzing the historical data of the previously mentioned CCPP database and using them, we created a partial simulation of the operation of the power plant (Figure 9).

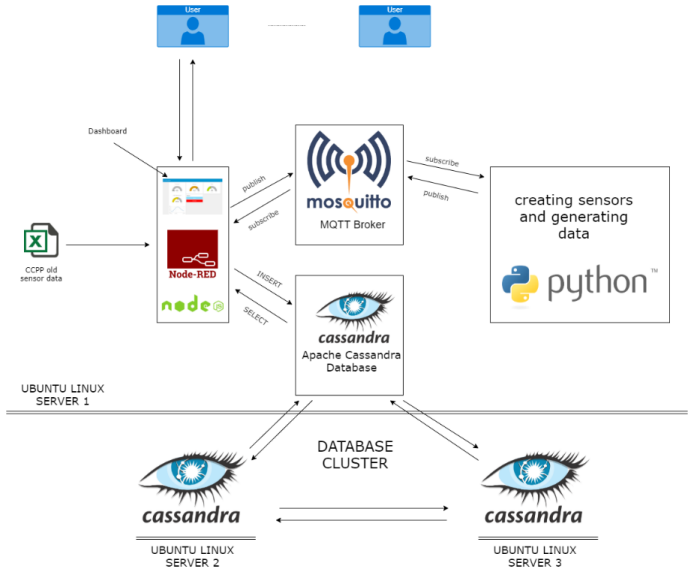

Figure 9
Block diagram of the system built to simulate CCPP

Starting from the data describing the most important system presented in Tüfekci's article [19], we wrote the historical data set into the Apache Cassandra and then using an application created in Python, we created sensor values similar to the historical data by simulation. Due to the efficiency of the simulation, there is no need for physical devices, it reduces the time interval for its reproduction, and enables simulation. We generated the simulated sensor data using the Python code, and using the Mosquito MQTT broker, we transmitted the data to the database via Node-RED and, if necessary, we can modify the values of the sensors on the user interface. In addition to storing the created data, using Node-RED's user interface (dashboard), we can continuously display and even modify the current values of sensors. Figure 10 shows the user interface displaying the sensor data and the possibility to modify the data.

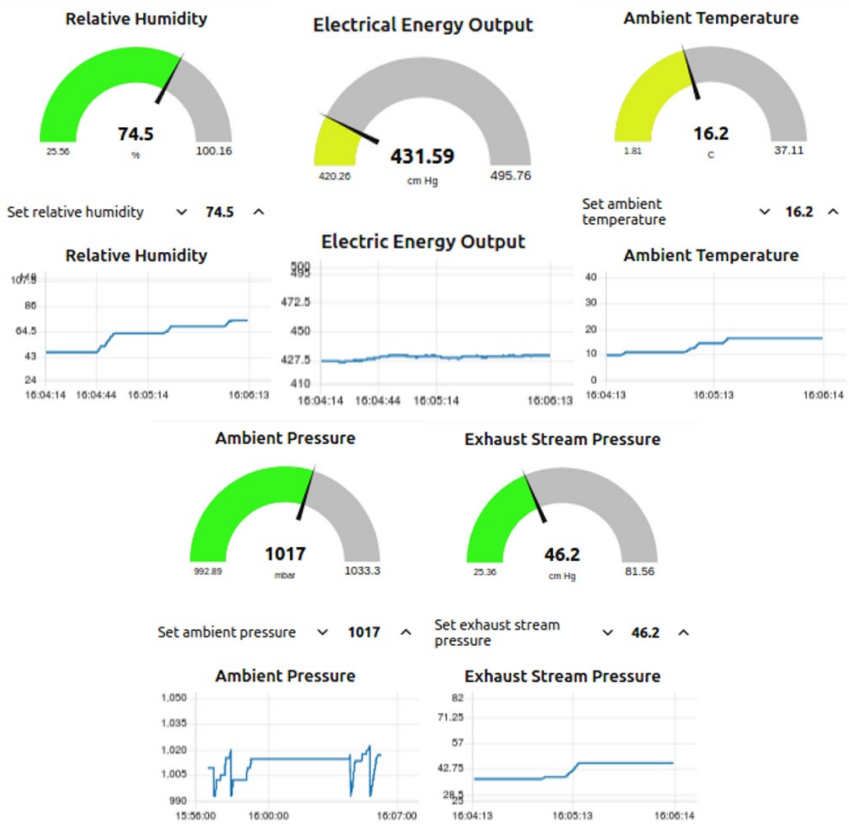

Figure 10
Generated sensor data on the dashboard

To import the historical data set (9568 data points from 5 sensors) into the database, we created a data flow in Node-RED, with the help of which the system can simply write the contents of the excel file into the database. Then, we were

able to display these data using the Node-RED Dashboard module. As a result, we can interpret 6 years of sensor data changes.

An important part of our work is that we were able to create a model in Node-RED by analysing the historical data and with the help of the correlations provided by Tüfekci, thus creating real data describing the operation of our plant system.

4.3 Application of Apache Spark for Statistical Processing of Collected Data

With the built-in functionality of Apache Spark, we can examine basic descriptive statistical analyses by running a few commands. The following descriptive statistics were examined during the research:

- Mean, Standard deviation, minimum and maximum values:
  - count – number of items in that column;
  - mean – arithmetic mean (sample mean), undistorted;
  - estimate of expected value;
  - stddev – standard deviation, the mean squared deviation from the arithmetic mean - how much our values deviate from the average;
  - min. and max. – determination of minimum and maximum value;
  - 25%, 50%, 75% – approximate percentage of quartiles;
- Median – the mean value of the data set, gives a robust estimate and is not sensitive to extreme values.
- Variance: it is used to describe the fluctuation of the data around the mean, that is, the value of the variance is small when our data move around the mean. The variance corresponds to the square of the standard deviation.
- Skewness: determines the offset of the peak of the distribution from the center position.
- Kurtosis: an indicator that describes the shape of the data set vertically.
- Covariance: gives the co-movement of two different variables.
- Correlation: it gives the magnitude and direction of the linear relationship between two values, that is it defines their relationship to each other.

Furthermore, we were able to perform outlier detection with the help of some functions and commands with the functionality of Apache Spark. With this architecture system, we performed the various descriptive data processing statistics indicated above, which were presented in the [14] article.

Conclusions

Based on the various solutions of the presented IIoT systems, it can be seen that we can "easily" implement the same system using cloud-based components developed by various service providers (Azure, AWS, GCP) for varying fees. The preparation of these implementations in the given field requires appropriate professional (industrial) and technical knowledge.

In the case of Microsoft, Amazon and Google, the services used are included in the product family they distributions are included, in most cases without compatibility problems. Using the products of multinational companies, all data and analysis logic end up in the cloud, on the service provider's servers,

The system was created as an in-house implementation: however it uses open-source, free services, and implements the same system as the previous three. In this case, we also need the right knowledge, but it is not so clear how to use one or another service, since these are not from the same service provider. Creating your own system open-source requires the creation and administration of your own hardware infrastructure.

Even new algorithms can be developed and tested in the own data analysis section. In order to achieve this, however, considerable human resources are also required.

The industrial integration of IoT or industrial IoT devices is unavoidable for any industrial representative who wants to stay competitive in today's digitized industry. On the other hand, it is a serious challenge to continuously maintain security without risking availability. Industry representatives create ever more connected small worlds around them, so it is necessary to prepare for protection against the threats and vulnerabilities associated with digitization. Companies integrating the principles of Industry 4.0 must be up to date in the field of cyber-security and must make it possible to quickly detect and respond to threats. The proven SOC systems and software (for example, SIEM) provide a solution for this, but it is also important to educate human resources on security awareness.

It should also be taken into account that no tool can fully protect against all cyber-attacks, but it is possible to prepare for them by taking precautions, and in the event of an attack, reduce possible negative effects and minimize system downtime.

The presented implementations properly reflect that with the help of the open-source Node-RED development environment, it is easy to implement the communication of different IoT technologies with each other. In this way, it can provide cost- and time-efficient solution options for those representatives of the industry who wants to test the effectiveness of a new smart system. At the same time, Node-RED has a distinct advantage in the field of data visualization, as well as the MQTT protocol over traditionally used industrial solutions, since spectacular results can be achieved by applying simple logic, so it proves to be

much more efficient in terms of cost and time. Thanks to this, the interface can be easily expanded as needed, in order to be able to monitor all sensor or machine data that we deem important.

In industry, the use of IoT devices mostly receives special attention due to the optimization of processes, the detection of errors and the implementation of predictive maintenance, since cost reduction can be achieved with the help of their integration, and by analysing and using the collected data, we can get to know our systems better.

Based on the above implementations, we can evaluate the development and integrability of today's technology in such a way that it is properly applied for use in the industrial field, for the use of IIoT. But it is not necessary to limit yourself to the use of just one of these services, even if they are paid large open source, but they can be integrated and used together, and in many cases their combined use is recommended, depending on what task we want to use them for.

Acknowledgement

This work was supported by the Collegium Talentum Programme of Hungary.

References

[1] Amghar, Souad, Safae Cherdal, and Salma Mouline. "Which NoSQL database for IoT applications?." 2018 international conference on selected topics in mobile and wireless networking (mownet). IEEE, 2018

[2] Andreeski, Cvetko, and Daniela Mechkaroska. "Modelling, Forecasting and Testing Decisions for Seasonal Time Series in Tourism." Acta Polytechnica Hungarica 17.10 (2020): 149-171

[3] Apache Cassandra, Open Source NoSQL Database, Accessed on: Aug. 17, 2022 [Online] Available: https://cassandra.apache.org/

[4] Badii, Claudio, et al. "Industry 4.0 synoptics controlled by IoT applications in Node-RED." 2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics). IEEE, 2020

[5] Chaczko, Zenon, et al. "Biomimetic middleware design principles for IoT infrastructures." Acta Polytechnica Hungarica (2020)

[6] Chakraborty, Mainak, and Ajit Pratap Kundan. "Grafana." Monitoring Cloud-Native Applications: Lead Agile Operations Confidently Using Open Source Software. Berkeley, CA: Apress, 2021, 187-240

[7] Combined Cycle Power Plant Data Set, Accessed on: Aug. 17, 2022 [Online] Available: https://archive.ics.uci.edu/ml/datasets/Combined+Cycle+

[8] Díaz, Manuel, Cristian Martín, and Bartolomé Rubio. "State-of-the-art, challenges, and open issues in the integration of Internet of things and cloud computing." Journal of Network and Computer applications 67 (2016): 99-117

[9] Eclipse Mosquitto, An open source MQTT broker, Accessed on: Aug. 17, 2022 [Online] Available: https://mosquitto.org/

[10] Ed-daoudy, Abderrahmane, and Khalil Maalmi. "A new Internet of Things architecture for real-time prediction of various diseases using machine learning on big data environment." Journal of Big Data 6.1 (2019): 1-25

[11] Ferencz, Katalin, and József Domokos. "IoT Sensor Data Acquisition and Storage System Using Raspberry Pi and Apache Cassandra." 2018 International IEEE Conference and Workshop in Óbuda on Electrical and Power Engineering (CANDO-EPE). IEEE, 2018

[12] Ferencz, Katalin, and József Domokos. "Rapid prototyping of IoT applications for the industry." 2020 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR). IEEE, 2020

[13] Ferencz, Katalin, and József Domokos. "Using Node-RED platform in an industrial environment." XXXV. Jubileumi Kandó Konferencia, Budapest (2019): 52-63

[14] Ferencz, Katalin, József Domokos, and Levente Kovács. "A statistical approach to time series sensor data evaluation using Apache Spark modules." 2022 IEEE 16th International Symposium on Applied Computational Intelligence and Informatics (SACI) IEEE, 2022

[15] Galambos, Péter. "Cloud, fog, and mist computing: Advanced robot applications." IEEE Systems, Man, and Cybernetics Magazine 6.1 (2020): 41-45

[16] Galambos, Péter, et al. "Design, programming and orchestration of heterogeneous manufacturing systems through VR-powered remote collaboration." Robotics and Computer-Integrated Manufacturing 33 (2015): 68-77

[17] Jelacic, Bojan, et al. "Security risk assessment-based cloud migration methodology for smart grid OT services." Acta Polytechnica Hungarica 17.5 (2020): 113-134

[18] Kanagachidambaresan, G. R. "Node-Red Programming and Page GUI Builder for Industry 4.0 Dashboard Design." Role of Single Board Computers (SBCs) in Rapid IoT Prototyping. Cham: Springer International Publishing, 2021, 121-140

[19] Katalin, Ferencz, and Domokos József. "Ipari IoT szolgáltatások és nyílt forráskódú rendszerek áttekintése: Overview of Industrial IoT services and

open source systems.", Energetika-Elektrotechnika– Számítástechnika és Oktatás Multi-konferencia (2020): 69-74

[20] Kuzzle Documentation. Accessed on: April 23, 2023 [Online] Available: https://docs.kuzzle.io/core/2/guides/introduction/what-is-kuzzle/

[21] Node-RED, Low-code programming for event/driven applications, Accessed on: Aug. 17, 2022 [Online] Available: https://nodered.org/

[22] Okanović, Andrea, et al. "Innovating a model for measuring competitiveness in accordance with the challenges of industry 4.0." Acta Polytechnica Hungarica 17.7 (2020): 67-88

[23] Pääkkönen, Pekka. "Feasibility analysis of AsterixDB and Spark streaming with Cassandra for stream-based processing." Journal of Big Data 3.1 (2016): 1-25

[24] R. Peters, "Securing the Industrial Internet of Things in OT Networks", Fortinet, Dec. 18, 2018. Accessed on: Apr. 24, 2021. [Online] Available: https://www.fortinet.com/blog/industry-trends/securing- the-industrial-internet-of-things-in-ot-networks

[25] Russell, Brian, and Drew Van Duren. Practical internet of things security. Packt Publishing Ltd, 2016

[26] Tabaa, Mohamed, et al. "Industrial communication based on modbus and node-RED." Procedia computer science 130 (2018): 583-588

[27] thethings.iO Documentation. Accessed on: April 23, 2023 [Online] Available: https://developers.thethings.io/docs/getting-started

[28] Thinger.io Documentation. Accessed on: April 23, 2023 [Online] Available: https://docs.thinger.io/

[29] Tüfekci, Pınar. "Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods." International Journal of Electrical Power & Energy Systems 60 (2014): 126-140

[30] Veneri, Giacomo, and Antonio Capasso. Hands-on industrial Internet of Things: create a powerful industrial IoT infrastructure using industry 4.0. Packt Publishing Ltd, 2018

[31] Weissman, David, and Anura Jayasumana. "Integrating IoT monitoring for security operation center." 2020 Global Internet of Things Summit (GIoTS) IEEE, 2020

[32] Wukkadada, Bharati, et al. "Comparison with HTTP and MQTT in Internet of Things (IoT)." 2018 International Conference on Inventive Research in Computing Applications (ICIRCA). IEEE, 2018