UNIVERSITY OF BERGAMO

School of Doctoral Studies

Doctoral Degree in Engineering and Applied Science

XXXI Cycle

SSD: ING-IND/17

INNOVATIVE ARCHITECTURE FOR INDUSTRIAL MONITORING SYSTEM

Advisor

Chiar.mo Prof. Fabio Previdi

Doctoral Thesis

Andrea PICCININI

Student ID 1009986

Academic year 2017/2018

Abstract

The fast globalization process of the last decade required companies to optimize production decreasing downtimes and scraps. Additionally, companies require tools for decision support in maintenance, orders scheduling and marketing.
This thesis aims to develop a new innovative architecture for monitoring and SCADA systems able to centralize data and support data analytics for ideally every kind of productions. The core of the innovation is the definition and management inside the platform of a JSON configuration which describe the machine to be monitored.
First, a literature review about the *state of the art* of technologies for web platform and distributed web system is performed, afterwards the architecture is explained and some cases study provided in different areas of production.

Contents

List of Figures

Chapter 1

Introduction

For years, SCADA systems were responsible to monitor and control the production process and notify any anomalies detected. The concept of SCADA system is recently evolved going towards a global dimension. Today, Cloud technologies and a powerful Internet connection allow engineers to design global systems able to collect, elaborate and store big amount of data and to provide the results of such elaboration in every part of the world real-time. In the scenario just explained, it is easy to understand that a new concept of production can be imagined and companies are trying to implement new services in order to improve the competitiveness in their business model. In fact, in the last few years, big enterprises have led application studies where emerge the importance of data. In this new industrial revolution, indeed, the data is the most important valuable entity. Then, the quality of data has become fundamental to provide simple, optimum and fast response in order to solve problems or faults in production, management or customer's

service. The concept expressed in the previous sentence highlights the way which companies are moving and focusing their effort. Customer's services, in fact, make the difference in a market where information runs very fast and competitiveness between companies is very high. In the following chapters an innovative architecture for modern monitoring systems will be widely described. This new architecture allows to integrate data from different existing systems and provides a communication channel where messages (or commands) run in both directions, from the server to machines and from machines to the server.

1.1 A new architecture for monitoring systems

The work proposed is the result of 3 years of study in manufacturing industry. Joining companies' needs and exploiting the newest technologies in the software field, a distributed, global, web based monitoring system has been designed, developed and now tested in several companies. The purpose of the project is to overcome the limitation given by current market solution where the different parts which compose a monitoring system are usually sold separately and the integration of them requires new design and development phase with the risk to introduce some inefficiencies. Modern monitoring systems require also two ways of communication where machines send most of the data involved in the monitoring process, but they must be able to receive instruction such as setup parameters, production scheduling and process settings.

The last sentence mentioned outlines the biggest difference between a general system assembled with third parties software and devices, and a general system integrates such as the one described in this thesis. Along the explanation of the architecture, the monitoring system will be represented as simple as possible considering every service as single entity. The reader must keep in mind that, in order to avoid single point of failure, every entity in the architecture is replicated more than one time creating a distributed system where entities of the same service are managed by a load balancer that equally balance the computational load of requests to that service and redirect the data traffic on other entities in case of failure of one of them. the main objectives to achieve are the following:

- Create a distributed system which manages data collection, data storage and data visualization easy to configure and open to several protocols.

- Propose a server based architecture explaining why this solution, though still not used in practice, is more suitable for new era monitoring systems compared with the current available solutions.

- Create an architecture able to implement interfaces with other data system such as MES and ERP.

- Enable the chance to introduce machine learning algorithms toward predictive maintenance.

- Allow the access to data from everywhere.

Finally, the work proposed is to consider as a complete product ready to be installed and run, data are immediately ready to be consulted through a web browser and other applications can use the same real-time data to implement new services or access to the database for further analysis.

1.2 The State of the Art

Chapter 2 gives a complete landscape about how technologies have evolved since the born of web and which of them have been applied in distributed system such as the one proposed in this thesis. Since the limited memory available on ICT systems and the limited capacity to elaborate data going through the limited Internet bandwidth, SCADA systems and many of other enterprise software were developed as stand alone application properly customized ad hoc for the current needs of the company which requested the installation. From the born of DCS on, the technologies behind SCADA and monitoring systems progressively increased until now when all the data run on the Internet and are accessible worldwide. The amount of data available is also changed and a new generation of database management systems was necessary to excellently manage and elaborate data when the frequency of INSERT and SELECT operations become huge. One more major aspect concerns the ability of the system to give a feedback to machines and send commands since an efficient data acquisition process is no longer enough to guarantee high competitiveness to companies that relies on data analytics in the way to improve their production. Finally, field-bus

and Internet protocols have impressively evolved opening new opportunities to manage data and offer new services. All those changes contributed to require a new architecture to manage data and make them available introducing all these new technologies and implementing new mechanism to exchange data.

1.3 Architecture design in monitoring system

Chapter 3 presents the designed architecture for monitoring systems. This architecture wants to be an innovative solution for companies applicable for every kind of industry where machine or production lines must be controlled and maintained. The objective is to build a monitoring system that allow to access data worldwide and provide critical information about the status of production assets by elaborating data with machine learning algorithm developed ad hoc based on the tool which the company wants to monitor. In this way, the production is continuously monitored and analyzed reducing probability of unexpected faults and then downtime of machine. The company also monitors the complete process of production and with this information can certify its products ensuring high quality to its customers. Every asset or machine can be easily monitored by creating a configuration file described in JSON format that specify the protocol type and the variable wanted to be acquired. In order to do that, cloud technology is used. Cloud computing offers a series of services which help

engineers in design phase providing solution automatically redounded, 100% available, cross region and with potentially unlimited memory and computational capacity. All part of the architecture included configurations and protocol message format will be carefully explained along the chapter.

1.4 Problems description

In chapter 4 an analysis of common problems solvable using the studied architecture are proposed. This is not to intend as a unique solution of the problem itself, but widespread problems discovered interviewing managers and responsible around the companies on which I tried to solve proposing at first a way to collect data in an efficient and reliable way and working on the final target afterwards. Being the solution presented suitable for any kind of machine or plant, I could apply the aforementioned platform on all the use cases presented along the thesis. The analyzed problems are the following:

- Quality control: with the appearance of new powerful sensors and acquisition hardware, quality control systems offer the opportunity to apply high precision control on geometry, defects and process drifts.

- Fault detection (or predictive maintenance): cloud computing offers high performance CPU which enable fast computation of big amount of data. With this technology, introducing specific algorithms is possible to identify faults of tools of machines before

they appear.

- Scheduling: with a complete overview of the process and real-time data, a monitoring system may also offer functionalities of operations scheduling able to optimize the production and reduce delivery time.

The 3 topics cited are a selection of problems preparatory to introduce the applicative examples of the chapter 5.

1.5 Applicative examples

In chapter 5 some applicative examples of the problems explained in the previous chapters are described. All the work proposed were directly followed and in most of cases directed by me. The purpose of the chapter is to present some real experience which the use of the architecture proposed either solved or helped to solve the problems described in the previous chapter. Briefly, The cases study present the use of the monitoring system in the following areas:

- Using the edge computing to prevent damages and ensure maintenance to critical assets such as circuit breakers and continuous casting plant in steel production adding particular attention to keep high quality in the final product.

- Monitoring assets to increase the quality of production. In this specific context 2 projects will be presented, a complete monitoring system for textile looms and a monitoring system acts to

improve the traceability of molds for press machinery.

- Big data for data analytics act to build a simulation model for scheduling algorithm implementation. Here the elaboration of acquired data provides input, constraint and output on which train the algorithm built.

Moreover, the full architecture is becoming an industrial product to monitor machines, cells or plants. The number of customers which are approaching this solution are increasing due to the integrability, accessibility and expandability of the product.

1.6 Conclusions

The last chapter will give a review of the overall content of the thesis, providing a critical analysis on advantages and disadvantages of the solution provided. Furthermore, a brief comment of the solution implemented and the improvements which may be applied in the future.

Chapter 2

State of the Art

Along the chapter 2, the state of the art of monitoring systems is presented analyzing the evolution of technologies since the 2000s to today. From field-bus to web technologies for visualization, the data transmission is explained. For those people who dealt with BEinCPPS [7], they may find the following topics an extension of the BEinCPPS project which aims to integrate and experiment a Fl-based machine-factory (FI stands for Future Internet). In fact, BEinCPPS project and this thesis share the same goal.

2.1 Maintenance and monitoring in industry

A monitoring system observes and tracks the operations and activities of users, applications and services on a machine or enterprise systems [8]. In the last few years, for industries, this concept has evolved in what

is called Cyber-Physical System (CPS) which is defined as combination of physical assets and digital representation of a machine or production plant.

The term cyber-physical system was coined in 2006 by Helen Gill of the US National Science Foundation [9]. In detail CPSs are engineered systems that are designed to interact seamlessly with networks of physical and computational components. These systems will provide the foundation of our critical infrastructure and improve our quality of life in many areas [10]. According with 5C architecture [1], a CPS is characterized by a layer architecture represented in Figure 2.1:

- Smart Connection Level. This level involves in sensor networks, machine communication layer and low-level plug & play devices. Raw sensors data are gathered in this level and the same data will feed all the implementations of the higher levels of the architecture.

- Data to Information Conversion Level. This level provides the first algorithms for component machine health, predictive maintenance and data correlation (the role of predictive maintenance is well explained in [11]).

- Cyber Level. This level is the core of 5C architecture, a digital twin model is used to predict the future behavior of components or machines. This allows comparing physical and cyber results.

- Cognitive Level. In this level, the implementation of CPS generates a whole knowledge for the monitored system. Data become

information that supports decision making for expert users.

- Configuration Level. This level provides the feedback from cyber space to physical space applying control to make self-configure and self-adaptive features towards the Zero Defect Manufacturing (an example of life-cycle approach for ZDM is described in [12]).

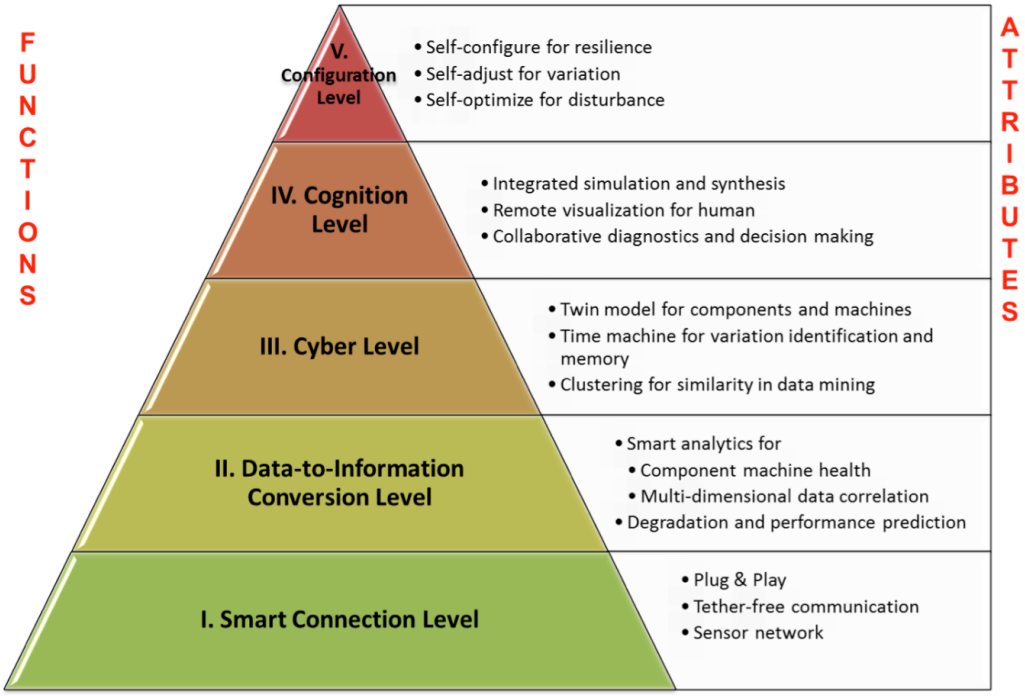


Figure 2.1: 5C architecure for CPS (source [1])

From his manuscript, Bartocci distinguishes CPS between two major contexts in which a monitoring system can take place: monitoring of real systems during their execution measuring Key Performance Index (KPI) and throwing alarms when the underlying KPIs drive out of bound, and monitoring model-based systems while running simulations the system generates data that must be evaluated a posteriori [13] [14].

This thesis focuses on the former context as described below. The implementation of a CPS aims to take advantage of use data collected from field to implement new services or applications and not just for the mere data visualization. In fact, before the coming of Industry 4.0, data were stored to monitor real-time process and compile statistics and reports. From this mere data acquisition, the first machine learning algorithms are applied for predictive maintenance and simulation models are integrated for real-time fault detection. These new features in monitoring systems guesses to move from systems with high memory and I/O database speed towards high computational performance The need of high computational power on server-side platform has favored the born of cloud systems and design concept oriented towards a centralized and modular architecture. Machine learning algorithms are then installed as modules of monitoring system platform and they extract data from database for the elaboration. According to the best practice of smart analytics in modern industry [15], the algorithm output is representative of the system status and can be read and understood from any kind of user in the company, expert or noob. Jay Lee recap this practice in an easy to understand model: translate the output of your algorithm in a scale of 5 values from 0 to 4, where 0 means that everything is working fine and 4 means that instant action must be taken to fix the system. In this way, the output of the algorithm is a direct and clear explanation of the state of the system that cannot be interpreted by anyone and is not subjected to the personal judgment of workers. Finally, an important aspect that very often is

kept in background is the security. CPSs are usually distributed across the world and collect huge amounts of information used to feed sophisticated algorithm that must provide reliable results for the company. Breaches in the data collection process could lead to wide-scale data leakage, theft or intentionally manumission [16].

In conclusion of this section, from the literature point of view, CPSs are classified as sophisticated systems placed above, albeit complexed and sophisticated, simulation model, static analysis and model checking. Many "experts" also declared that high-level programming languages (e.g., Java or C#) are not applicable to CPS due to their inability to address significant resource constraints at a high level of abstraction [17]. But is it really possible classify a programming language as applicable or not to CPSs? A programming language should be the means through create a tool with pre-designed features and characteristics. That is why the following sections will focus on the tools, devices and software which need particular attention in the discipline of CPS towards industrial monitoring systems and not on programming language used to code them or the method which they work with.

2.2 Field-bus protocols

Before being processed to provide output algorithm, data must be collected from machines. Since the 2000s, field-bus protocols move on Ethernet cable [18]. The application of Ethernet to connect field devices offers substantial advantages compared to field-buses as Ethernet enables the consistent integration on all levels of a company [19].

Most famous new field-bus protocols based on Ethernet standard are Profinet, OPC Unified Architecture (OPC UA), Sercos III, MTConnect and EtherCAT:

- Profinet is the first integrated Industrial Ethernet Standard for automation, and utilizes the advantages of Ethernet and TCP/IP for open communication from the corporate management level to the process itself [7]. The communication is cyclically and is divided into several phases. The first phase is the isochronous phase where Isochronous RT (IRT) frames are sent. Clock instant when a frame may be sent is scheduled during the initial configuration of the network. The difference between the standard IP protocol is that the addressing is based on sending time instead of MAC address. To do this, special hardware is required (category C switches).

- OPC UA is a platform-independent industrial middleware technology [20]. It defines methods for data manipulation: modeling and transport. All information is placed in a memory area called address space by using object-oriented methods. OPC UA is not limited to specific protocol communication and follows the client-server principle. A client or a server use APIs to service requests and responses. The communication stack involves in transport layer, secure channel and encoding level as depicted in Figure 2.2. OPC UA has created firstly for HMI and SCADA application in industry field.

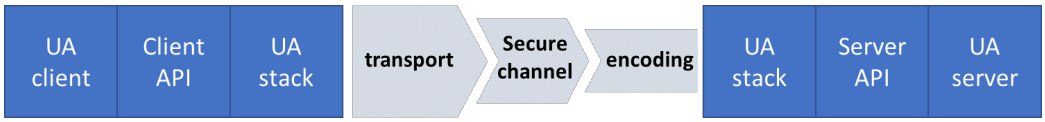


Figure 2.2: OPC UA stack

- Sercos III is oriented to the development of distributed motion control systems [21]. Devices are organized as double ring structure that ensure fault robustness. The communication is based on a time-slot method with cyclic telegram transmission on a master-slave principle. A master sends a Master Data Telegrams to slaves and slaves themselves can communicate each other by means Cross Communication and Controller-to-Controller communication profile. Ethernet frames transport only small amount of data, as well as Profinet, in order to achieve isochronous RT, special hardware is required.

- MTConnect is an open and extensible protocol designed for the exchange of data [22]. It allows manufacturers to collect data from machines, controllers and sensors in a client-server principle. An MTConnect agent serves client answering to HTTP requests with XML format. MTConnect organizes information into a hierarchical structure which represent the relationship (father-child) between any device. Only the MTConnect agent (the server of the communication) is allowed to write data while all the client agents can just read the information published. Using purely HTTP protocol, no special hardware is required. In fact, MTConnect has been created for monitoring and not for control. Thanks to its

features, MTConnect is probably the best protocol for mere monitoring in the industry of the future [23].

- EtherCAT is a real-time Ethernet protocol designed for industrial applications [24]. The communication is based on master-slave principle. The master creates a process image that involves the state of input-output slaves. To change the state of a variable, both master and slave must send a specific change command. In the network topology the master must be always available while slaves can turn off and on again. EtherCAT provides a high-performing hardware for real-time data transmission via Ethernet.

The protocols described above provide an interface towards the monitoring system or better towards a gateway which translate the field-bus protocol into a unified format useful to store data on the system server. Particular attention is due by monitoring system designers on the machines-gateway communication because of the wide variety of field-bus protocols.

2.3 IoT devices and protocols

Section 2.2 described the modality with data were collected from machines. This process may require high frequency sampling, but requests and responses, depending of the protocol used, are exchanged in a Local Area Network (LAN) and latency times can be under control. On the other side of the communication, data must be transmitted to a server or cloud by means the Internet network.

In that communication, data may require an aggregation to reduce the amounts of bytes to send and the connection to the Internet must be continuously monitored to avoid data loss. For this purpose, a gateway is not only a simple transmitter, but incorporates smart strategies to manage the communication [25]. Smart gateway performs a number of tasks, like, collecting the data and performing preprocessing, filtering the data and reconstructing it into more useful form, uploading only necessary data to the cloud.

More than managing the task mentioned above, as well as field-buses are managed by specific protocols, gateway-server communication is done by particular modality and protocols optimized for this purpose.

In this way IoT protocols are become the de facto standard to provide communication between smart component and cloud [26]. These protocols are Constrained Application Protocol (CoAP), Message Queue Telemetry Transport (MQTT) and eXtensible Messaging and Presence Protocol (XMPP):

- CoAP is a document transfer protocol following the client-server principle. Unlike HTTP, CoAP runs over UDP, this means that client and server communicate through connectionless datagrams in an asynchronous communication. For a M2M communication usually both client and server are implemented on the same device [27].

- MQTT is a broker based instant messaging protocol. It is developed by IBM and join the publisher-subscriber principle [28]. It runs over TCP and it is currently used on hundreds of Android

applications for notifications and instant chat.

- XMPP is an application profile of the XML that enables the near-real-time exchange of structured extensible data between any two or more network entities [29]. Very similar to MQTT, XMPP is based on publisher-subscriber principle. It is used by Facebook for chat and notification.

To better understand the difference between client-server and publisher-subscriber principles, the Figure 2.3 shows a sequence diagram of a simple message exchange.

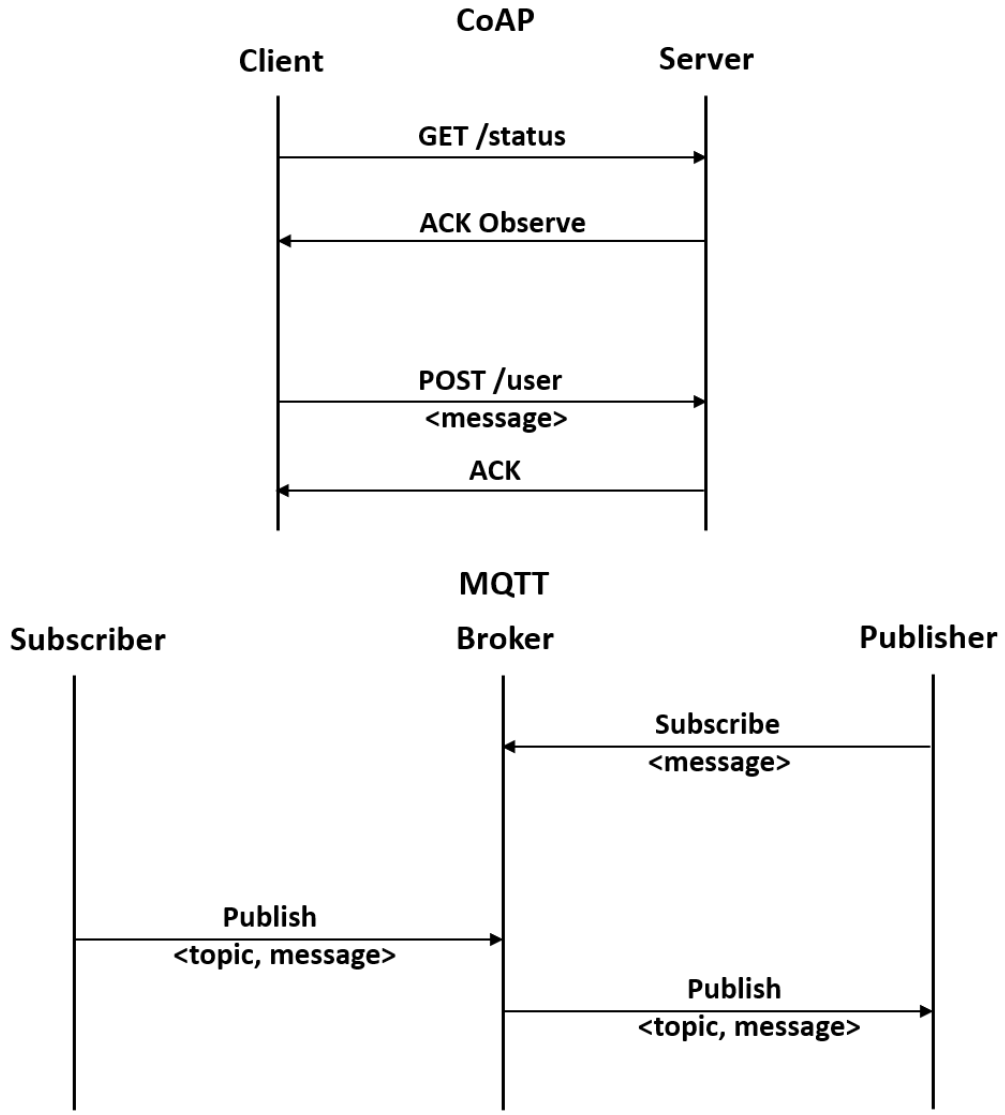


Figure 2.3: comparison between client-server and publisher-subscriber

2.4 From field-bus to cloud

To efficiently accomplish the tasks of data collection according to the modality described in section 2.2 and 2.3, the literature offers a general architecture for IoT gateway devices basically structured in 3 layers [2][30][31]:

- Perception layer. The Perception layer is also known as Device Layer. It consists of the physical objects and sensor devices (in industry also machines). This layer basically deals with the identification and collection of objects specific information by machines. The collected information is then passed to Transmission layer for its secure transmission to the information processing system. Here an IoT system deals with field-bus protocols and RFID or barcode tags.

- Transmission layer. The Transmission layer is also called Network layer. This layer securely transfers the information from machines to the information processing system (cloud). In this level of communication, IoT protocols play an important role to efficiently deliver data to the cloud.

- Application layer. The application layer provides additional services according to the case studied (manufacturing, food and beverage, ...). Those services are substantially algorithms which elaborate the information collected and generate new data to improve the production process. This activity is entirely done on the cloud. In literature, this layer is also split in Middleware, Application and Business layer.

To accomplish the functionality just explained, IoT Gateways has a modular software architecture which implement protocol conversion and device management in addition to information logging and caching. Figure 2.4 depicts an example of general software architecture for IoT

gateways.

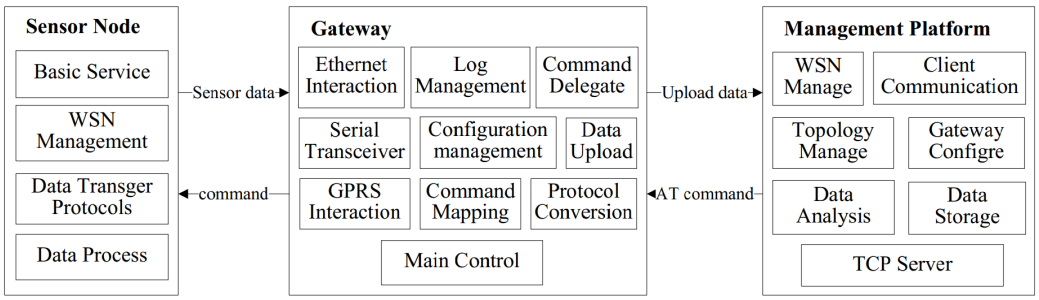


Figure 2.4: software architecture of IoT gateway (source [2])

2.4.1 Gateway communication and edge computing

As anticipated, to collect data from machines and store data in a centralized monitoring system become tricky when machines send data with several different protocols. The gateway communication machines-side is the most critical activity to face in the development of a general monitoring system architecture. In this case a gateway is a component (hardware and software) that implements drivers and interfaces for aggregating and collecting data from various field-bus. When the gateway has not an embedded configuration, the critical issue in data collection is summarized in instruct the gateway about how and which information retrieve. Substantially field-bus protocols can be very different between each others and general instruction for the gateway configuration are needed (for instance MODBUS is structured in registers and is untied from the concept of variable while MTConnect declares a variable for any data around the bus). It is right here where the

main difference between the architecture proposed in this thesis and a traditional monitoring system is made in place. In fact, while commercial gateway has an embedded configuration that instructs them about what collect and in which database store data without caring about cloud level, this design of architecture embeds configuration on the server (or cloud, from now on server and cloud will be treated as synonyms for the purpose of this manuscript) and instructs the gateway about what data collect managing the classification and the storage of data from a server point of view. Indeed, cloud-based monitoring systems are been supporting by even more (albeit not much as expected) recent researches [32] [33] [34] that provide an overview from a high level of abstraction of strengths of systems designed in this way. In behalf of cloud-based architecture and the author cited in the previous sentence, as author of this thesis I am forced to say that as regards my studies and experience in the manufacturing and IT fields, cloud-based systems are the only innovative solution for the next generation of monitoring systems and manufacturer or supplier who does not conform to this new design pattern will definitively increase the gap with the leaders of the market. This argumentation does not want to be a complaint against the current state of the art, but a clear-minded consideration that can find agreement by people who only work and deploy any kind of solution beside the real industrial world.

Unfortunately, literature on industrial manufacturing gateway strongly focuses on 5G connection [35], machine level communication [36] and energy efficiency [37], but only few isolated researches focus on the im-

portance to transfer these heterogeneous data directly to the cloud in a common format. In 2016 Wen defined the use of RESTful framework as a most innovative way to store data on the cloud from a gateway [38], similarly Jabbar presented an interesting work where data gathered from a warehouse are pushed to the cloud via REST-based framework. By the way, RESTful APIs are widely used in the web to perform data transfer based on human driven action, but cannot lead data at a high rate.

On the other side, the communication from the gateway to the cloud (or server) is required to be light, fast and standard. These are the reason why for that communication IoT protocols are used. The structure of the messages exchanged must translate field-bus protocol into a hierarchy of variables ready to be stored in a database on a server and ready to be simply used for visualization on HMI. For pure monitoring system, the gateway-server communication is one way: data flow solely from machines to database. Modern applications overcome the concept of monitoring and want to implement correction of process drifts by data elaboration on cloud. To do this, the gateway implements the instructions to decode and perform commands towards machines.

In some cases, certain kind of variables require high frequency acquisition in order to implement algorithm for quality check, process control or high frequency signals aggregation (e.g. peak current of brushless motor). Usually, when the control is not incorporated on the machine, the implementation of algorithms like the example reported are done with a separated hardware and acquisition system which takes

care only for this specific operation. Nevertheless, in such cases where the elaboration is performed before sending data to the cloud, we talk about *edge computing*.

Data is increasingly produced at the edge of the network, therefore, it would be more efficient to also process the data at the edge of the network[39]. The edge computing is a simplification of the bigger concept of *fog computing* where a kind of "small cloud" is installed at the edge of network or better is decentralized.

In detail, fog computing is a highly virtualized platform that provides compute, storage, and networking services between end devices and traditional cloud computing data centers, typically, but not exclusively located at the edge of network [40].

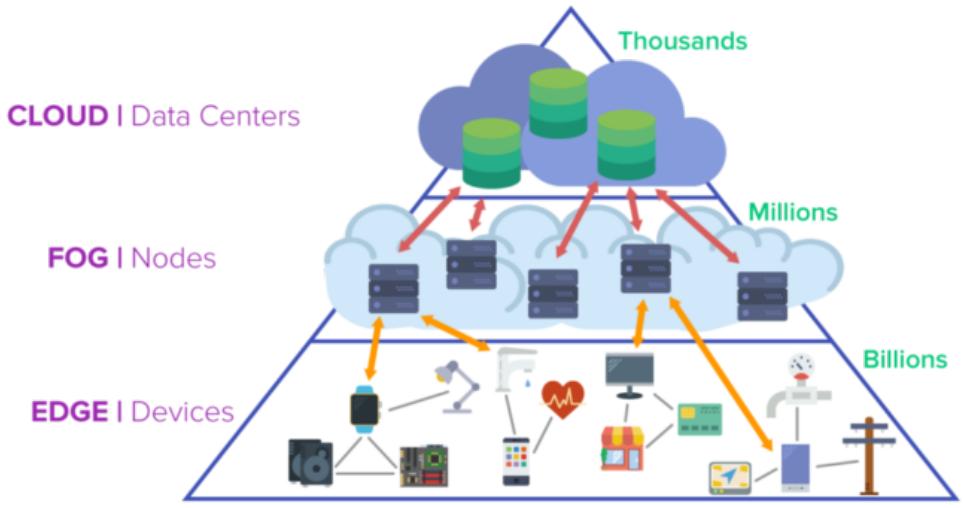


Figure 2.5: Computing layer (source [3])

The figure 2.5 represents the structure of computing in a modern and complex architecture. In a few words, fog computing is a decentralization of cloud computing when the amount of data start becoming

gigantic and edge computing is a separation of job for data processing towards the higher level which may be a fog computing layer or directly a cloud computing architecture.

2.5 NOSQL databases

Data from the gateway are then stored into databases. Information acquired by sensor may have high sampling frequency and generates large amount of data as well as holds database for write operations. Respect relational databases, NOSQL databases offer high I/O performance at the expense of update and delete operations. NOSQL, for Not Only SQL, refers to an eclectic and increasingly familiar group of non-relational data management systems; where databases are not built primarily on tables, and generally do not use SQL for data manipulation [41]. Truly, most of them allow SQL syntax to manipulate data although the core structure is completely different, and the SQL commands get translated in appropriate operations. Based on the features a database implements, according to CAP theorem [42], a database can cover maximum 2 of 3 basic characteristics [43]:

- Consistency. All clients see the same version of the data, even on updates to the dataset.
- Availability. All clients can always find at least one copy of the requested data, even if some of the machines in a cluster is down.
- Partition tolerance. The total system keeps its characteristic even when being deployed on different servers, transparent to the client.

Usually, as shown in Figure 2.6, relational databases are consistent while no-relational ones are partition tolerance. In a monitoring system, in order to avoid data losing and reduce latency time, databases for data collection are highly available.

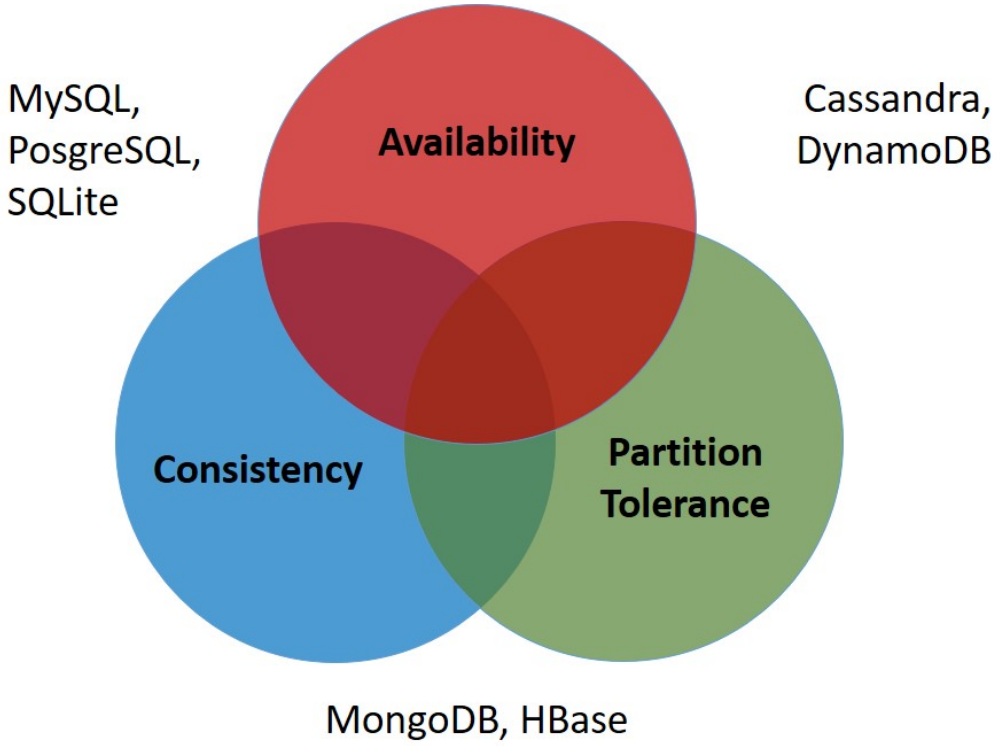


Figure 2.6: CAP theorem

The CAP Theorem is one example of a fundamental trade-off between safety and liveness in fault-prone systems. Examining this inherent trade-off yields some insights into how systems can be designed to meet an application needs, despite unreliable networks.

2.5.1 Time-series databases

A subgroup of NOSQL database is Time Series DataBase (TSDB). This kind of database is optimized for handling time series data or better any arrays of values indexed by timestamp. The need of databases optimized for time-series acquisition is the surprisingly increment of streaming data from many fields of business as financial (for stock market), industrial (for machine data), medical, etc... [44]. TSDBs are usually column oriented databases. As the name suggests, they store records organizing data in columns, or better the values of each column are stored consecutively. This mechanism brings several advantages in term of:

- High performance for aggregate query, in particular insertion of multiple records in the same command.
- Efficient partition and compression of data.
- Filter few columns in wide columns table.

On the other hand, single insertion, deletion or update makes decrease the performance of the DBMS.

A parent node of column oriented DB is column family DB, an example of it is Apache Cassandra [45]. Apache Cassandra offers the highest performance for I/O operation per second. On this topic, Netflix published the results fo a series of tests towards to verify the performance of its Cassandra cluster database on AWS [46] [47]. Table 2.5.1 shows the result of the tests, it considers performance and costs.

	48 nodes	96 nodes	144 nodes	288 nodes
Writes Capacity	174373 w/s	366828 w/s	537172 w/s	1099837 w/s
Storage Capacity	12.8 TB	25.6 TB	38.4 TB	76.8 TB
Nodes Cost/hr	174373 w/s	366828 w/s	537172 w/s	1099837 w/s
Test Driver instances	10	20	30	60
Test Driver Cost/hr	$20.00	$40.00	$60.00	$120.00
Cross AZ Traffic	5 TB/hr	10 TB/hr	15 TB/hr	30 TB/hr
Traffic Cost/10min	$8.33	$16.66	$25.00	$50.00
Setup Duration	15 min	22 min	31 min	66 min
AWS Billed Duration	1 hr	1 hr	1 hr	2 hr
Total Test Cost	$60.97	$121.94	$182.92	$561.68

2.5.2 Relational vs. NOSQL database

There are several differences between relational databases and NOSQL ones. First of all, they store data in different ways, relational databases data are structured and thus stored in tables following the schema defined in the design phase and when the schema is well designed (even though this property force engineers to struggle during design phase to avoid huge workaround due to thoughtlessness designing the DB schema), this way to save data avoid redundancy [48]. Au contraire, NOSQL databases are schemeless, data are not stored standardly, but the physical representation of data depends of the DBMS used. This different way to store data allows NOSQL databases to be more flexible, avoiding multiple *null* fields (e.g. when an attribute misses in a record) on behalf of nested data in JSON representation saving also storage

space [49].

Since relational and NOSQL databases store data in different way, also querying data is performed differently. Relational DBs use the structured and consolidated SQL syntax to retrieve data stored. NOSQL DBs have their own language to query data instead.

The biggest advantage of NOSQL DBs respect relational ones dealing big data is the scalability. NOSQL DBs are conceived to run on distributed systems composed by several cluster and machines where data are split and replicated on. Thanks to the computational power distributed on the nodes, parallelization of operations ensures high performance (although the variety of NOSQL DB on the market today makes hard figuring out the right choice for the purpose [50]). This means that, in order to increase the performance, NOSQL DBs scale horizontally increasing the number of nodes in a cluster while relational DBs scale vertically increasing the capacity of the hardware of the machine where it is installed.

The last important difference regards the properties guaranteed by the 2 type of DBs: ACID and BASE (see the manuscript of Eric Brewer who is considered the father of the BASE concept [51]). ACID refers to the property in a transaction of relational databases to be:

- **Atomic**, the transaction must be executed entirely, otherwise the operations done must be canceled and the database restored at the state before the transaction started.
- **Consistent**, during the execution of a transaction, the DB must be in a consistent state according to the integrity constraints de-

fined on it.

- **Isolated**, every transaction is independent from the others, when a transaction fails it does not interfere with the execution of the others.

- **Durable**, after a transaction commit, the update must be persistent and must never be lost.

On the other hand, BASE refers to NOSQL DBs. Starting from CAP theorem, Brewer said that a NOSQL DB provides high availability of information (Basically Available) and then they are inconsistent in some instants of time, information may not be the latest updated (Soft state). Consequently, data are consistent only when insertion and updating ends (Eventual consistency).

Both databases will continue to exist alongside each other with none being better than the other. The choice of the database to use will depend on the nature of the application being developed [52].

Last but not the least, in the last few years, the need of both: the high scalability feature of NOSQL databases and at same time the well defined structure of relational databases have led to born of a mixed architecture where relational DBMS (e.g. MySQL) are clustered in multiple nodes providing high availability with entity-relationship model [53] [54]. An example of these systems is Amazon Aurora, a native clustered relational database supplied as a service by Amazon and Galera, a clustering software that enables MySQL to scale horizontally maintaining consistency.

Finally, when should you choose a NOSQL database rather than a relational one? When you have to store with high input rate data in series that are conceptually related with any other content or when you do not need complex queries along relations between tables for example. Otherwise, when you have a logical well defined relational structure where your main tables grow averagely the same and you need complex queries joining several tables, then a relational database is what most suitable for you.

2.6 Web technologies

The last step to accomplish in order to finalize a monitoring system is the visualization module. Along the years Web has taken on greater prominence as a source of information for workers and managers [55]. The world of World Wide Web (WWW), since its born, continuously upgrades and improves in each its features and in the last few years it is probably the IT technology that currently changing faster than all the other. Since the web1.0 where information was only readable through the web2.0 where information can also be added, web technologies are projected towards fully exploit the web3.0 where contents change based on user needs. In a deeper analysis of the literature, the ultimate goal of the web3.0 would be to combine new technologies to open intelligent way for delivering dynamic, interactive and targeted contents specific to the user in smart interface [56]. An example of the web3.0 application are the digital assistants (Siri, Cortana and so on) [57] which actively interact with the user and the WWW to satisfy the user's requests.

Although web3.0 is mainly used in application like e-learning [58] and e-commerce, nowadays this technology is spreading also in application supporting companies in decision making activities and data consulting. To better understand what web3.0 is, some sentences taken from [59] are extracted below. According to Conrad Wolfram web3.0 is where computer will generate and think new information rather than humans. Google CEO, Eric Schmidt says web3.0 will be "applications which are pieced together - relatively small, the data are in the cloud and it can be run on any device (PC or mobile), very fast, very customizable and distributed virally(social network,email,etc)". Yahoo founder, Jerry Yang thinks that web3.0 is a collection of tools and techniques for creating programs and online application, which blur the distinction between professional, semi-professional and consumers. "...you don't have to be a computer scientist to create a program. We are seeing that manifest in Web2.0 and 3.0 will be a great extension of that, a true communal medium... the distinction between professional, semi-professional and consumers will get blurred, creating a network effect of business and application" - Jerry Yang [60]. Finally, Nova Spivak from Radar Network believes that web3.0 will be "The Semantic we" which will play a central role in the new generation.

In conclusion, biggest market players are strongly moving on web3.0. In particular services such as customer's assistant and search engines are the application leading the web towards this new generation of technology. About that, machine learning algorithms are hugely used elaborating server-side data to improve the ability of computers to

implement the web3.0. In this sight, also layer 1 and 2 technologies (from ISO-OSI) are working in this direction [61].

2.6.1 Languages and libraries

Since the born of web1.0, Hyper Text Markup Language (HTML) is the standard language (more precisely meta-language) for web development. Today, with its final release HTML5 [62], the HyperText Markup Language provides hundreds of features to guarantee the best user experience. In particular, HTML5 offers features for [63]:

- **Storage**, data can also be saved on the user's computer. Web app can work without Internet connection.
- **Motion**, cursor movements and clicks generate events and elements on the page can move dynamically.
- **Games**, Interactive games can run inside the browser without installing softwares or plug-in.
- **Audio**, songs and tracks are played in the browser without a plug-in.
- **3D**, WebGL creates interactive 3D effects and renderings using computer's graphic processor.
- **Video**, videos are embedded in a web page and played in the browser without a plug-in.

The variety of features implemented in HTML5 leads technologies like Java (client-side), Silverlights and flash to die.

Beside HTML5, JavaScript in all its shades (jQuery, Angular, ...) is the main scripting language for web browser [62] and essentially it is the responsible of dynamicity and animation of web pages. JavaScritp is an object-oriented language that uses prototype objects to model inheritance. Variables can be created by simple assignments without explicit declarations. Finally, values are freely converted from one type to another type with few exceptions; the access to a non-existing property of an object return a value represented as undefined, this value is uncastable. Modern websites and web application largely use JavaScript to interact with the users.

Last fundamental technology in the web applications development is the Cascading Style Sheet (CSS) used to format and style the layout of web pages. It is used to define text size, font and background colors, object positions, margins and so on. The purpose of CSS is to separate presentation and content, this enables for multiple HTML pages application the reuse of code biasing the content accessibility. The CSS specifications, as well as the HTML ones, are maintained by the World Wide Web Consortium (W3C) [64].

Chapter 3

Architecture design in monitoring system

In the big data and Industry 4.0 era, computing is being transformed from Distributed Control Systems (DCS) to Cloud systems. Data from transducers or sensors network are roughly moved towards a centralized architecture which, by means of Internet, provides a series of services [65]. Then, cloud computing is a centralized system where data gathered are stored, compressed or elaborated by specifics algorithms to provide new information in any kind of business.

Specifically, in the last decade in the field of industry, services like monitoring, scheduling and communications have transformed to web applications and provided by means the Internet connection on any kind of device.

This new paradigm of computing needs a new concept in the design of modern applications which optimize resources consumption and

latency time.

Notice that the architecture which will be presented afterwards involves in one single instance of any entities, however, modern architectures for critical production systems implement mechanism to avoid single point of failure and then all the entities are implicity or explicity redounded. For many services on public cloud this is implicity given, but in case it does not, in the design phase of the system, the entity redundancy must be foreseen.

3.1 Monitoring system base architecture

Nowadays, data collection is the main problem for companies which deal with even bigger amount of data day by day. In industry field, more than other sectors, data collection is a critical issue that insiders must face. From sensors to user's device passing through a storing structure, lot of aspects must be considered as following:

- Field-bus protocols. Generally, every manufacturer implements its own protocol. It is just the lack of few standards which makes hard to build a general acquisition gateway.

- Data format. The data format is another aspect not to underestimate, data from various field-bus protocols need to be translated in a unique data format which the monitoring platform can understand and manage.

- Cyber security. The theme of security and privacy of data is one

of the main obstacle which push companies away from the use of public cloud computing bias an on-premise solution.

- Database solution. Currently, also for small implementation, more than one DBMS are used.
- Web visualization. Data display is provided by means web technologies. The coming of web 2.0 favorite the web platform development.

The 5 points above suggests a new modular architecture as shown in Figure 3.1.

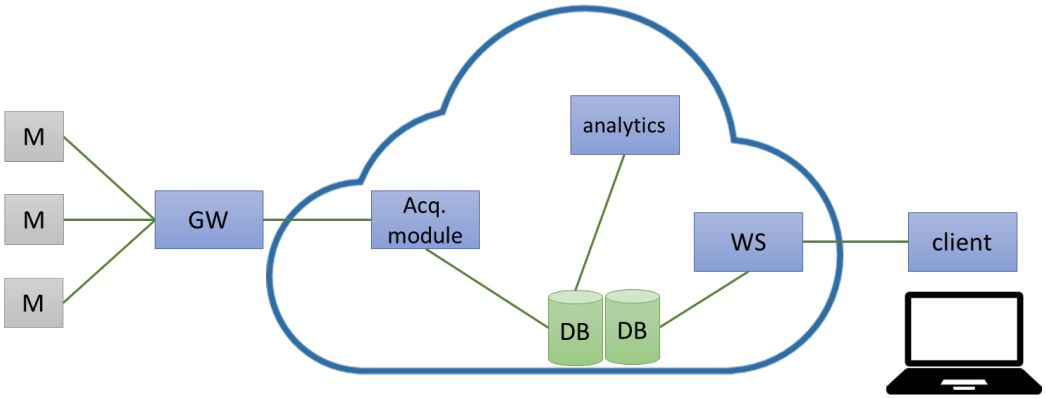


Figure 3.1: new architecture concept

3.1.1 Machine and gateway communication

The presence of many player on the automation market for years has provided several protocols on the field-bus layer. This is a problem when a company has different machines from different providers and it needs to interface with them to collect production data.

The first solution is provided implementing all the protocol drivers the company needs, in this case the effort for design and development may be very strong and depends on the number of different machines the company has. Moreover, the introduction of new machines may imply the development of new drivers.

The second solution proposed assumes the involvement of manufacturers. Inside the IoT concept, the architecture shown in figure 1 is led toward the one shown in Figure 3.2 where each machine communicates with the cloud through an IoT protocol. In this case a commercial gateway is suggested to use since the communication between gateways and acquisition module can be customized and adapted. Usually is just the manufacturer who install a single gateway for every machine. In the subsection 3.1.2 this theme will be treated and analyzed.

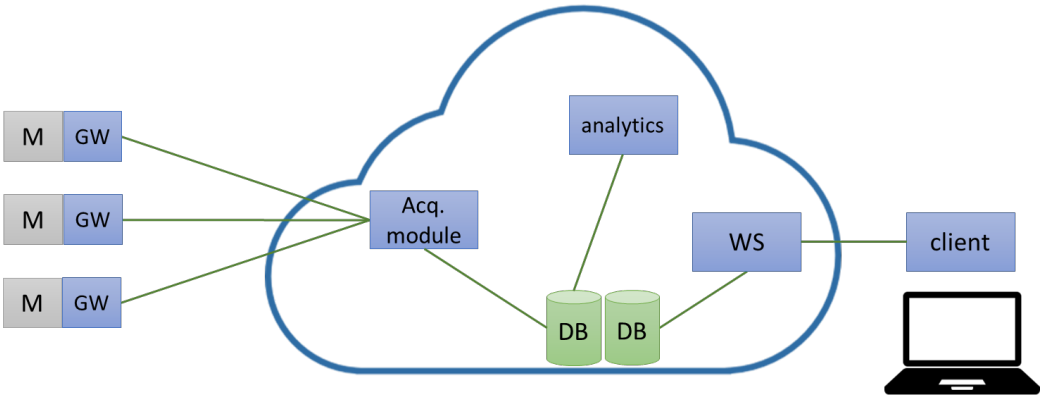


Figure 3.2: IoT architecture concept

Towards the Industry 4.0, the biggest players on market in machines manufacturing sponsor their own monitoring system. This is a great solution for those small enterprises which work with single brand ma-

chines. Same machines mean same standards and protocols, in this case the monitoring solution suggested by the provider is the easiest and optimal choice. On the other hand, more complex production systems need ad hoc implementation which often constitute high effort to be developed and low flexibility. During the discussion of this thesis, the matter of flexibility will be treated.

Once again, because of interruptions of connectivity, is a good rule to forecast a buffer to temporally maintain data until the connection with the cloud is restored.

3.1.2 Data format and IoT protocols

Refers to the architecture depicted in Figure 3.1, the acquisition module is the first service in cloud. It deals with data storing and provides a security layer against cyber-attack. However, the most important feature of the acquisition module is to translate machines data format in monitoring system data format. This is done by means a machine configuration that describes the way the gateway collects data from that machine, the way the acquisition module stores data and the way the web server displays information.

The machine configuration structure is as complicated as the monitoring system is flexible. A flexible monitoring system means of course high configuration time. Reduce this time is duty of developers which must implement fast wizard configuration with default settings. In the section 3.2 an example of machine configuration will be shown and described.

The second theme to face in the design of a communication gateway-cloud is to use a lite communication protocol which implements small overhead, structured payload and possibly with native encryption mechanism.

3.2 Analysis of machine configuration structure

In the architecture presented in Figure 3.1, the gateway needs to be taught about from which machines requests data, what data collect and how to establish a connection with the machine (what kind of protocol use). Substantially, the gateway must be configured for every machine communication it has to establish. There are 2 ways to accomplish this task:

- Centralized: for each machine the configuration is managed by the cloud. First, to start the data collection, the gateway needs the Internet connection to communicate with the cloud and asks for the machine specifications. A centralized management of configurations simply allows to add machine and instructs the gateway to start collecting data from it.

- Decentralized: the gateway stores locally the machine configurations and immediately collects data from machines. The advantage of this solution is that gateways can start collecting data without the Internet connection, on the other hand, the biggest

disadvantage is the upload or editing of configurations from remote devices.

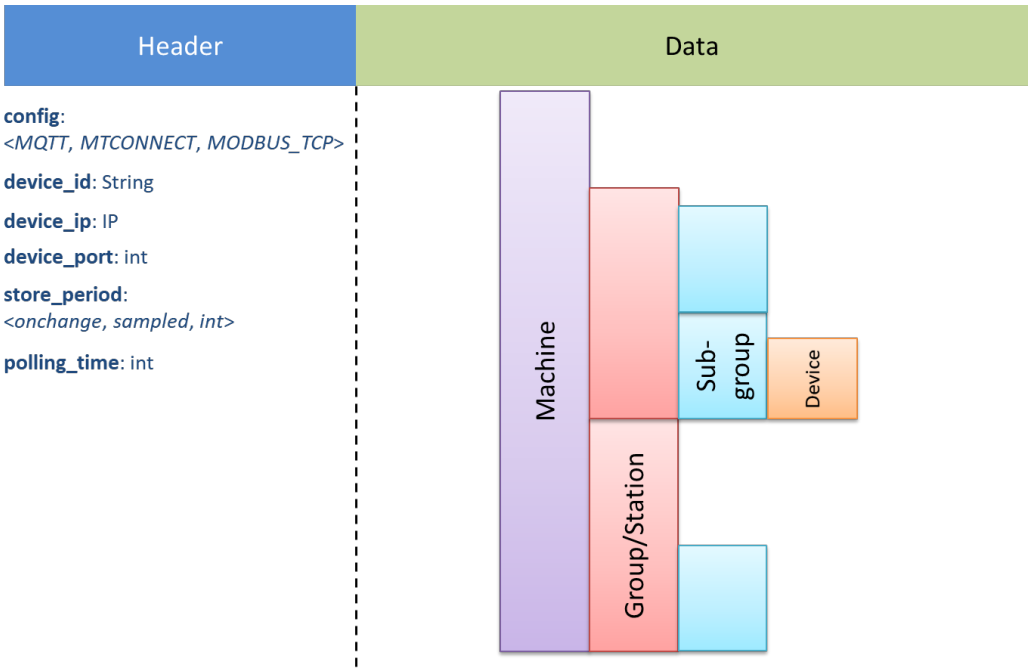


Figure 3.3: configuration structure

The Figure 3.3 represents an example of logical structure of a machine configuration. The configuration involves in 2 main parts: **header** and **data**. In the header the information about machine, protocol and sampling are stored, this information is fundamental to establish a connection with the machine and start to request data. In the data the physical structure of the machine is represented, variables are organized in global variable (of the machine), station variables, sub-group variables and device variables. This organization is also replicated on the web GUI to select which parameters the user wants to monitor.

Moreover, the configuration contains information about the store

frequency in the database or the indication of only real-time value. Below a snap of configuration suggested and applied in the work subject of the thesis.

```
{
"config":"MODBUS_TCP",
"device_id":"COS01",
"device_ip":"192.168.0.2",
"device_port":502,
"polling_time":1000,
"store_period":"5",
"machine":{
    "vars":[
        {
        "reg_address":7,
        "length":2,
        "datatype":"int",
        "var_id":"totale_pezzi_ok",
        "endian":"big",
        "store2db":false,
        "gw_type":"variable"
        },
        {
        "reg_address":21,
        "length":1,
        "datatype":"int",
        "var_id":"target",
        "endian":"big",
        "store2db":false,
        "gw_type":"variable"
        }
    ],
    "groups":[
        {
        "name":"StazioneA",
        "vars":[
            {
            "reg_address":152,
            "length":1,
            "datatype":"int",
            "var_id":"macchinaA(ignore)",
            "endian":"big",
            "store2db":true,
            "gw_type":"alarm",
            "extract":[{
                "mask":1,
                "value":1,
                "var_id":"allarme_inmanuale_A"
                },
                {
```

```
                    "mask":2,
                    "value":2,
                    "var_id":"allarme_portelle_A"
                    },
                    {
                    "mask":4,
                    "value":4,
                    "var_id":"allarme_blocco_A"
                    }
                    ]
                },
                {
                "reg_address":302,
                "length":1,
                "datatype":"int",
                "var_id":"velocita_stazioneA",
                "endian":"big",
                "store2db":true,
                "gw_type":"variable"
                }
            ]
            }
    ]
}
}
```

3.3 Web server architecture models

In the last decade, the most common web server architecture, based on 3 main actors: client, server and database, has evolved in many facets. Although the 3-tier architecture remain the most used, in relation of the purpose of the web application, its architecture can be different.

Charming web and mobile applications have to satisfy several requirements such as modern design, fast page loads, automatic scalability, high availability, fast time-to-market and attractive user experience. Each one of these features individually may appear easy to satisfy, but achieving all of them unfold to be quite difficult in practice. In fact, a survey by *New Bamboo* reveal that over 30% of web development teams deliver projects late or over-budget.

3.3.1 3-tier architecture

As said few lines above, the most common way of implement web applications is by means 3-tier architecture. It foresees to divide the application in 3 layers which typically map the physical infrastructure: graphical user interface, business layer and data access layer. As shown in Figure 3.4, the client requests a page to the server that renders the page content. Data are retrieved from the database by the server. The database is transparent for the client which via HTTP(S) implements a communication with the server. This design pattern sets some troubles regarding the requirements.

- A typical website performs more than 100 resources requested to the server, which make latency critical for performance

- With server-rendering, the client waits until the server has mounted the data through database queries and rendered it in a language such as Java, Python, PHP or .NET.

- The database must be highly available, which is extremely difficult if consistency is also needed, according to the CAP theorem already seen in chapter 2, section 2.5.

- The server needs to be scalable, which is difficult when shared state such as user sessions are involved.

- Business logic is duplicated between the client and server, this cause overhead communication between front-end and back-end

teams, making projects more complicated and delaying the time-to-market.

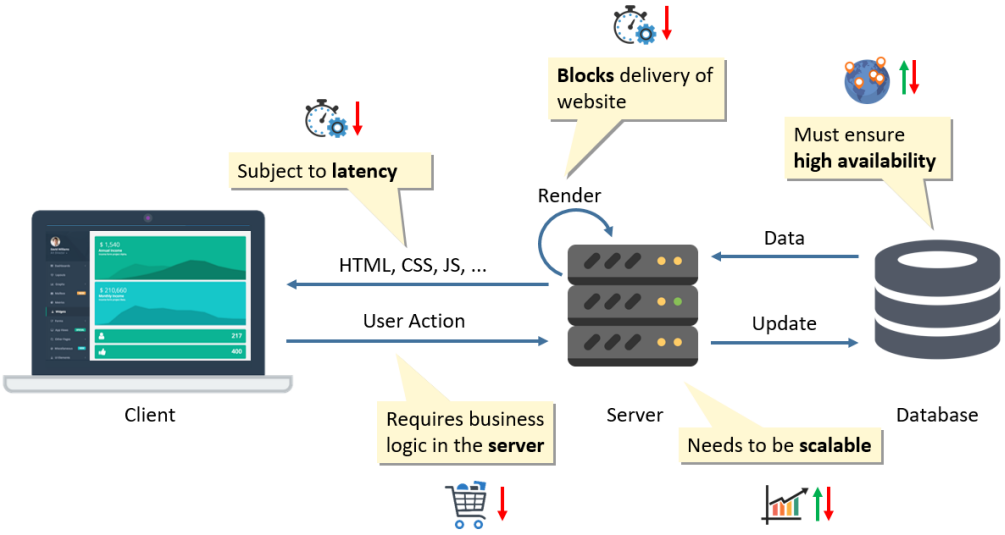


Figure 3.4: 3-tier architecture (source [4])

3.3.2 2-tier architecture

Respect 3-tier application, it does not implement the business layer (refer Figure 3.5). An example of 2-tier application is a Database as a Service (DBaaS). This has two main advantages:

- The website is rendered progressively in the client browser improving user experience. This kind of applications are usually developed in Single Page Application pattern (SPA). A SPA needs to download the template page once and update only the content it needs on demand.
- The business logic is moved to the client side where the Javascript

technology manages data. This reduces problems between back-end and front-end teams.

Although there are several great framework and technologies for developing 2-tier application front-ends (e.g. React by Facebook) and storing data (e.g. influxDB and MongoDB), some problems are not solved:

- Many latency-sensitive requests remain because of data comes in many separated JSON packets from HTTP/S requests.
- Scalability and high availability managed before from the business layer, are now shifted to the DBaaS.
- Consequence of the previous sentence, cloud databases are usually not suitable for handling server side business logic and access control.

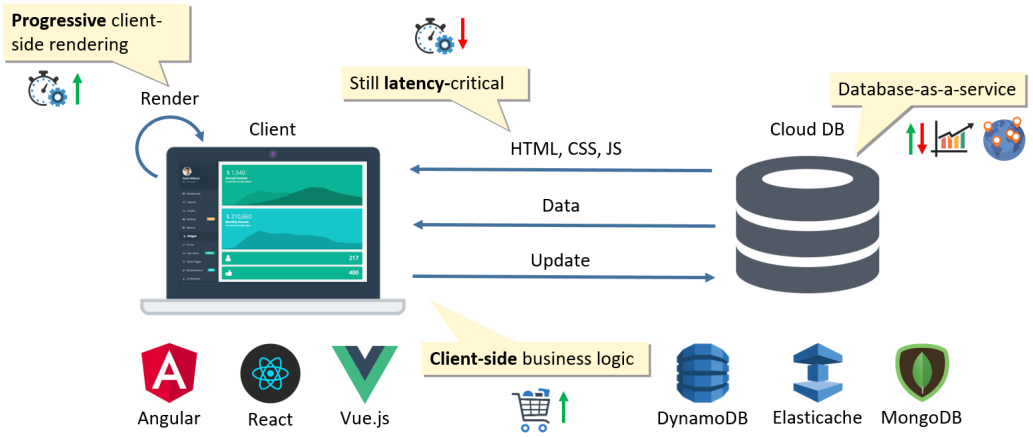


Figure 3.5: 2-tier architecture (source [4])

3.3.3 Serverless paradigm

The two common models of serverless architectures are Function-as-a-Service (FaaS) (e.g. AWS Lambda, Google Cloud Functions) and Backend-as-a-Service (BaaS) (e.g. Baqend, Firebase).

In the FaaS (Figure 3.6), the business logic is quickly scalable, easy to code and highly available. On the other hand, FaaS is completely stateless which means other functionalities such as data store, push notifications, etc... must be arranged in different cloud services. As a result, browser clients have to exchange data with many different APIs and the infrastructure may quickly become infeasible to manage.

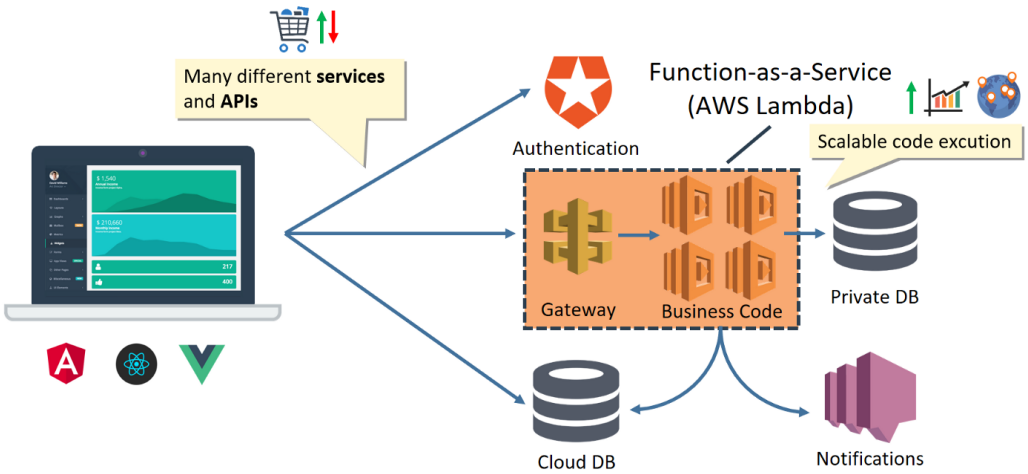


Figure 3.6: FaaS architecture (source [4])

This is where BaaS (Figure 3.7) joins the game, the main idea is to merge the readiness of FaaS with all the capabilities which a typical application, web or mobile, offers.

Basically, BaaS raises the abstraction level providing a single set of APIs to host and deliver all the service and the assets in the web ap-

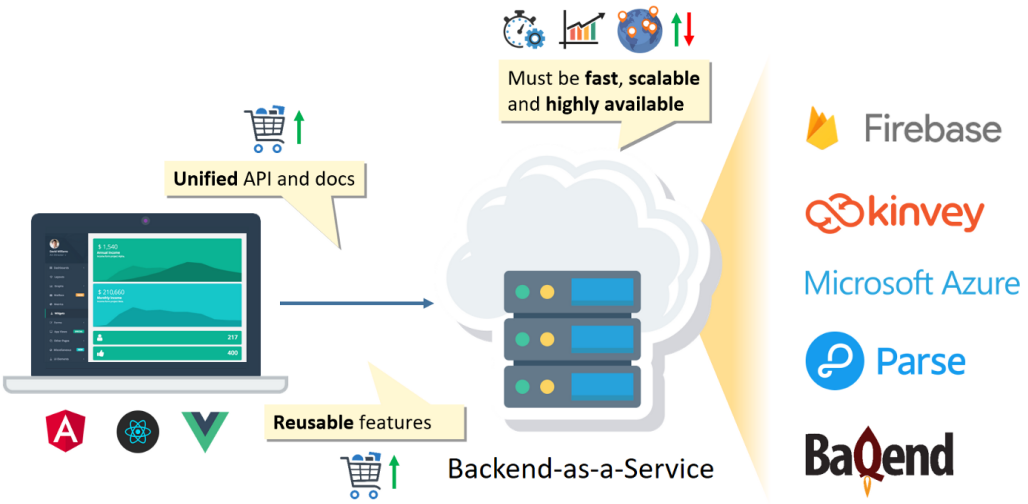


Figure 3.7: BaaS architecture (source [4])

plication. Thus, developers do not worry about manage the full stack of protocol services. If BaaS is shooting fish in a barrel, Platform as a Service (PaaS) gives full control over the server mapping server and database functionalities in a unique set of APIs using a framework such as Django. Nevertheless, small enterprises are reluctant to publish sensitive files or information on public cloud, the best way to implement such a complicated architecture is using a BaaS model. In those cases where all the system must reside on-premise, the 3-tier model is currently the most used. Usually, software houses develop both solutions.

3.3.4 Difference between on-premise and public cloud, pros & cons in architecture

Differences between a cloud and on-premise solutions are not limited to the model architecture. Cloud solution provides services ready to

use and managed from provider IT personnel. Approaching the public cloud is always advised managed services respect to implement the functionality on virtual machines.

On the other hand, on-premise installation is highly customizable with the hardware which better ensure high performance based on the services the application provides. For particular applications like fast analytics algorithms, choose the right hardware is the key for the success or the fall of the execution software.

Another aspect not to underestimate is the bandwidth, cloud datacenter usually has dedicated high speed Internet connection which ensures fast response, moreover the possibility to deploy the application on more than one datacenter and reduce the ping time for request from different part of the world. Spread the application in more than one datacenter also offer robustness of blackout or connection problem while on-premise a fault on electrical or Internet network may determine a denial of service.

3.4 Machine Learning algorithm

In the modern implementation of SCADA or monitoring systems, the simple visualization of real-time data is only one obligatory operation which leads to collect and store data for further analysis. Substantially, the architecture explained in this section is only the means to achieve the goal of smart factory where predictive maintenance algorithms, dynamic schedulers and adaptive machine controls improve the flexibility and the reconfigurability of factories.

Machine learning algorithms can be divided in 2 categories:

- Offline: the algorithm reads data from database and periodically performs the analysis on them to extract the result of the elaboration. The offline elaboration is used for periodical reports or decision-making support algorithms triggered by an event.
- Online: the algorithm reads real-time data and analyses them before being stored in database. On cloud there are services like Amazon Kinesis which accomplish to this purpose analyzing real-time stream data in various modality. The online elaboration is used for checking process drifts in real-time for the zero-defect manufacturing paradigm.

The integration of machine learning algorithms in a monitoring platform depends of its classification and the architecture model used for the platform implementation. Nowadays, cloud providers implement in their back-end modules act to implement machine learning algorithm for both offline and online categories. Using a cloud solution, in this case, is probably the best choice in order to have a scalable platform in term of computational performance which adapt itself to the amount of data to analyze.

Chapter 4

Problems description

In chapter 3 a general architecture of monitoring system has been presented. This technology is imported in the factories for a series of matters and problems which a company wants to solve. Related to the field of application of the company and the level of automation of it, the implementation and installation of a system act to monitor the productive process can improve and favorite the execution of determinate tasks for critical phases. In fact, despite the utility of a complete overview on the production system, the implementation of a web platform to monitor the process is not required to solve one of the problem mentioned following. As already said in the sections before, the data collection is the only fundamental phase which allows to acquire the data needed for the elaboration of themselves. The visualization phase can be done in multiple ways and the use of web technologies is only the most modern and widespread method in the industries of today. Moreover, the storage of the data is an optional phase when the ac-

cess of historical data is required. Anyway, the concept expressed and explained in chapter 3 have been used for the work proposed. In this thesis, 3 common problems for manufacturing industries will be explained (in this chapter) and some use cases presented (chapter 5). All the use cases are research works made or directly followed by me and described in manuscript already publish or submitted and under evaluation. This does not mean that the thesis aim to solve this problem, but only that the architecture presented is a tool which can be used to solve this kind of problem if applied accordingly.

4.1 Quality control in the production line

Since '90s, products quality was strictly correlated to the process information. In fact, all the information collected during the production process contribute to improves the process itself [66]. Following this perspective, producing high quality information is the first step to produce better products. Once again, monitoring and analyzing real-time data can give an important advantage when a company approaches the quality control on a single line or on the whole plant.

As explained by Ballou [66], Collecting data is not enough to ensure high quality data analytics and then improve the working process using the result elaborated. In fact, the quality of the process goes through the quality of data gathered. Nowadays, company databases are still containing wrong data and they continue to collect data relying on employees declaration.

The scenario briefly discussed above is the result of study of quality

research in the past and describes a situation that companies must have overcome in at least most of the process area of the production. Anyway, the quality of data is still the first step to improve the quality of products today and it is an important topic that companies must focus on before going ahead.

At any rate, improving the process means also increase the items and tools available for analysis and evaluation of the goods produced. In fact, modern technologies, by means high performance sensors, guarantee the detection of defects real-time during the assembling of components directly on the machine. In the last few years, with the progressively increasing of computing power, quality control systems such as vision systems and high frequency signal elaboration systems have contributed to improve the effort of human based control where high speed and tiny creeks make the human job impossible.

As mentioned in 2.4.1, edge computing aims the implementation of the cited vision systems and data processing systems. An example of virtual sensor exploiting edge computing will be given in 5.1. Moreover, vision systems are promoter of stand alone systems according to the paradigm of edge computing and help to institute a quality control system revolution appearing in thousands of application in the manufacturing industry. That is not only because the quality of cameras and optics have been increase hugely since 2000s, but the image processing which gives that added value needed to high reliability systems. Due to the importance of these systems, a brief summery of those is mandatory in order to understand how they work and in what way they can

be used.

4.1.1 Vision systems

Traditionally, surface product control and visual inspection are performed by humans. Although humans can do the job better than machines in some cases, they are slower than the machines and get tired quickly. In some environments (e.g., underwater inspection, nuclear industry, chemical industry etc.) inspection may be difficult or dangerous. Computer vision may effectively replace human job in those cases [67].

Figure 4.1 depicts how a vision system can be integrated in a monitoring system installed on a cloud. Although the information of the elaboration is used also locally to control the production, data are also sent to the cloud for reports and further analysis. In this case, perform the data processing on cloud requires wide band and fast connection in order to transmit high resolution images especially if the production line is very fast and vision system takes more pictures per second.

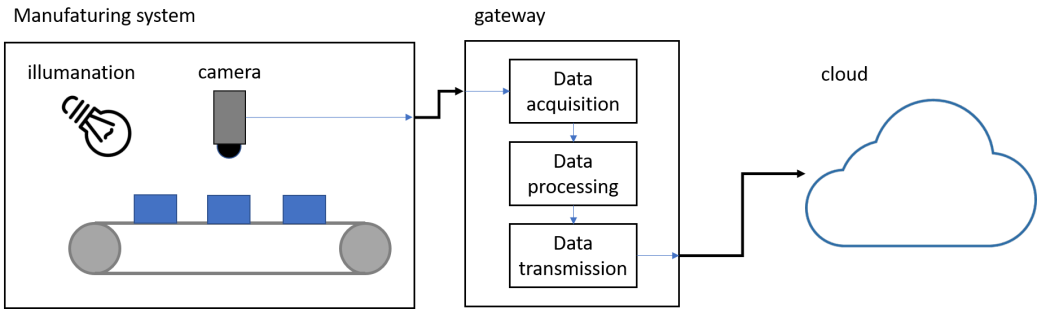


Figure 4.1: vision system with edge computing

Although vision systems are evolved impressively, most of applica-

tions for quality control on the production line in industry are still using very simple technology and algorithm. A common vision system in manufacturing involves 3 actors:*illumination* (or lighting), *camera* and *cpu/gpu*.

- Illumination. It is a critical part of a vision system and must be carefully designed. The main purpose of lighting is to provide a consistent scene which keeps the environment monitored by the camera apparently the same in every image captured [68]. Based to the subject object to the analysis, there are 3 aspects to consider in order to design a good illumination system. The first aspect is the lighting source (incandescent, fluorescent, halogen, Xenon, LED), they differ for intensity and color. The second aspect is the lighting arrangements (backlighting, front lighting, side lighting, structured lighting, ring lighting) while the third aspect is the lighting geometry (point lighting, diffuse lighting, collimated lighting)[68].

- Camera. Talking about vision sensors, 2 categories of camera can be identified. The first uses the CCD sensors, cameras with CCD sensors provide high quality images with no noise and distortion. On the other hand, the second category uses CMOS sensors, 100 times less power in terms of energy consumption than CCD and faster in pictures capturing. Having less quality then CCD, however CMOS sensors are widely used and in the next few years they are expected to cover almost the 100% of the market.

- Cpu or gpu. They are the entities on charge to elaborate the images with specific algorithms. Nowadays the market offers general purpose solutions with software which customize the application (in this case intended as algorithm) running the program with common computer hardware. Custom solutions involve dedicate hardware instead, this means that ASIC or FPGA elaborators are placed next to computers in the purpose to increase the speed of the elaboration.

Thanks to the research in the robotic field, vision systems are continuously evolving. Even though vision sensors are already very powerful and provide high quality images at very high frequency, machine learning algorithms need to improve for applying the intelligence and perspicacity of human in control where the high speed of the production line and the small defect of the piece inhibit the role of human.

4.2 Fault detection and predictive maintenance: safety and downtime reduction

Globalization and more stringent requirements push managers to optimize all systems involved in their organizations. In this scenario, maintenance plays a key role for reaching the KPI needed for equipment reliability and availability [6].

In the last few years, maintenance approach has been moved from

Repair Maintenance (RM) to Preventive Maintenance (PM) and Condition Based Maintenance (CBM) which, by continuously monitoring the equipment, estimate the degree of health of the equipment itself. In this way, maintenance operations are accomplished only when directly specified by the system analysis. CBM enables diagnostic detecting, insulating and identifying faults (EN 13306 2001) by monitoring signals of physical assets to mine information characterizing the health status of the system [69]. Machine condition monitoring can play a major role in plant maintenance. However, to take advantage of this relatively new maintenance strategy, it is necessary to have an understanding of the technology involved [70].

According to prognostic approach [71], the evolution of CBM is PM that aims to predict the future trend of equipment health condition. Here, the purpose of PM is to estimate the Remaining Useful Life (RUL) [72] of a physical system using intelligent algorithm and automated programs.

Although PM offers an important evolution in the field of maintenance, diagnostic always remains a fundamental activity in assets review. In fact, a physical system may ever suffer of faults never observed and foreseen of PM algorithms.

Diagnostic methods are mainly subdivided in Model Based Methods (MBM) and Process History Based Methods (PHBM) in relation to the type of knowledge used in the development of the model [69].

Basically, the first, MBM, is based on a deep knowledge of the physical process that enables the definition of a mathematical model

in a white-box approach. The second, PHBM, needs a big amount of historical data in order to define a relationship between input and output in a black-box approach.

The story of MBM began in the early 1970's applying the methodology to simple linear system [73]. It evolved afterwards when in 1978 Himmelblau applied the method to a chemical process [74]. Fault Detection Methods approach appeared in literature in 1975 instead, when Pau L.F. published the article "Failure diagnosis and performance monitoring" [75]. Since 1993, PHBM open an interesting alternative to MBM introducing black-box approach to solve fault detection problem [69]. The even increasing amount of data acquired and complexity of physical systems led engineers to experiment PHBM instead of MBM. Although PHBM can be implemented in relatively few time with good results, MBM is still preferable when the mathematical model of the physical system is known.

In any case, no matter the type of diagnostic method or the complexity of the diagnosis algorithm, in current industry process, it is necessary to provide a Decision Support System (DSS) by monitoring the operating conditions of plants and machines. Here, data collection is once again the first operation to accomplish since DSS wants to be a reliable and useful item supporting production. Based on these concepts, 5.2 and 5.3 will be presented in chapter 5. In these cases, sometimes the diagnosis requires high frequency data and the elaboration requires lot of resources in a little time, that is the reason why these activities are performed locally on a particular gateway prepared

for the purpose that contains the algorithm to run with the acquired data, transmitting only the aggregated data.

4.3 Production scheduling: optimization of resources

Think about monitoring and data acquisition can solve production scheduling problems may sound strange. What people do not think is that scheduling algorithms require a good deal of data and a deep knowledge of the process you need to schedule. Then, to schedule operations online and automatically, a stable and constant flow of data from production is necessary. Well scheduling the production is an hard work, but optimize the pallet processing through the machines of the production line reduce the takt-time and increase the throughput of the plant. For this reason, even more companies already automatically schedule the withdrawals of raw material from stores and the delivery of goods to the shipping area. Scheduling operations among the critical phases of production requires, however, extremely precision and knowledge of the process. In fact, complex algorithms are applied to solve *job shop* problem [76] or other similar issues. In a few words, the job shob problem can be described as a list of n jobs, each of them composed by several operations and each job must be executed entirely on one of the m machines available considering a certain number of constraints. The goal is to reduce the makespan [77]. In this way, data such as the right instant of availability of machines or the time to fulfill an order

are necessary to really optimize the process.

One of the cases proposed in chapter 5 deals with scheduling in the production. In that case the algorithm was implemented in a simulated environment in order to avoid downtime and to provide a tested algorithm ready for the production. Once again, the simulation model was created using the data acquired by the monitoring system installed on the plant. In this case, it is important to focus on the first part of the architecture: the one which collect data from machine and store them into the database. Even without a web platform, there are dozens of applications which allow to extract data from a database and organize them into structures suitable to be further elaborated with software of analysis such as excel or Matlab. By the way, simulation is another powerful tool which can be integrated with a monitoring system and to provide a real digital twin according to the definition of CPS released in chapter 2. Although the architecture proposed does not implement a digital twin natively, the integration with custom algorithm coded in a second time, ensure at the platform the capability to climb all the 5 levels of the 5C architecture. In fact, in spite digital twin is sometimes embedded in the most modern machines, data analytics on stored data can guarantee the same result in terms of timeliness of maintenance and quality of production.

Regarding the open topic of production scheduling, simulation aims to provide a forecast of the behavior of a plant or a machine offline avoiding risk of downtime and elaborating the best schedule solution for that production. Finally, even though monitoring system and sim-

ulation appear as tool operating in different field, they can cooperate and also coexist one inside the other in order to increase the quality of production and reduce the time to delivery of company orders.

Chapter 5

Applicative examples

In this chapter will be proposed some example in order to solve or correct industrial thematics talked in chapter 4. The concepts explained in chapter 3 have been used even partially and applied on monitoring cases. Nevertheless, the common point of the applicative examples proposed is the acquisition of data from machines or devices in the manufacturing industry. The versatility of the architecture designed offers a wide range of applications where the use of the platform, and the softwares linked, aim to improve the production. Moreover, even though the system architecture has its own model and flow, the high modularity of it ensures to exchange modules or components to get always the best performance needed. For example, introduce another noSQL database rather than Cassandra is possible editing a single class of the software. In the cases proposed below, some of the components of the software have been replaced by other more suitable for the application. An aspect to emphasize is that the thesis is not proposing a

study of architecture or a theoretical platform, but a real concept that already runs in one or all its parts on industry plants and in the future it will be installed in several other productions. As already said, all the examples proposed are part of my work experience and treat work I have directly followed.

5.1 Slag detection in continuous casting process

Maintenance activities can be very expensive and most of cases the level of it may condition the quality of the final product. One of the most relevant sectors with this characteristic is the metallurgic industry, a capital intensive process with need of high efficiency plants. In this sector, in which a better quality of material is required [78], the casting phase is crucial for quality and, as critical part of casting, the control and the monitoring of the slag during the casting phase is fundamental to achieve the required performance. As a result, Slag Sensing Systems (SSS) cover a top level of attention in molten steel Continuous Casting Production (CCP) [79]. The purpose of this work was a vibrational analysis and a characterization of the process of CCP in order to develop a real time system of Vibrational Slag Detection (VSD) for a real application [5].

5.1.1 Plant and process description

As just said in the previous paragraph, one of the most critical operations in CCP is to detect the presence of slag. Nowadays, the most used and reliable sensor in this process exploit the difference of magnetic conductivity between slag and molten steel when they pass through the sensor. Thus, the sensor wraps the shroud (the pipe that allows the molten steel to flow under the ladle) where the steel passes at 1200Â°C, for this reason the sensor is subjected to several maintenance activities and replacements. Nevertheless, recent studies show how the vibrational behavior of the system changes when the slag passes through the shroud instead of steel. Those studies have come to the development of a new approach for slag detection named VSD (Vibrational Slag Detection). The plant is described as depicted in figure 5.1, molten steel flows from the ladle to the tundish through a nozzle valve: in order to keep the temperature homogeneous within the metal, generally the flow is isolated from external environment by a refractory shroud. Flow control is implemented by an operational arm, that sets the section of the nozzle valve affected by the flow.

Consequently, adding a tri-axial accelerometer on the arm, which acquires vibrational signal, and an industrial computer, that processes and monitors acquired signal, it is possible distinguish pure molten steel from mixed steel with slag, increasing final product quality and cleaning of the CCP [5].

In literature several articles deal with slag detection through a posteriori processing of vibrational signal, hard to apply in the industrial

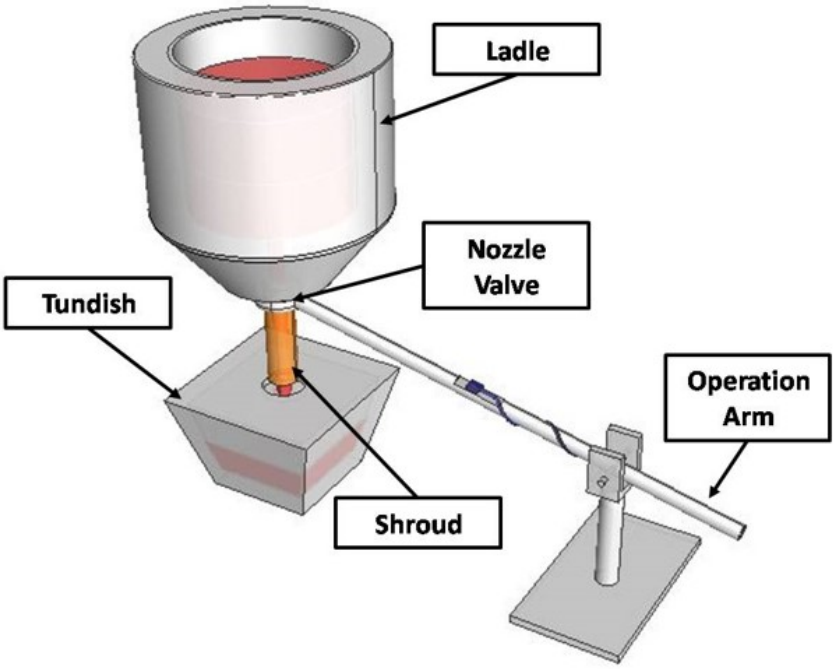


Figure 5.1: Plant structure (source [5])

context, while no significant studies have been found about real-time on-line detection [5]. In this work instead, an online application is provided and the new sensor is still used on the production plant by the company.

5.1.2 Acquisition set-up and data analysis

In order to perform preliminary analysis of the vibrational signal, an appropriate acquisition system is developed. It consists of a tri-axial accelerometer, located on the operational arm, an industrial computer and a software program. Taking into account the working temperature and the frequency bandwidth, a piezoelectric ceramic accelerometer TAA2230 (see table 5.1 for details), whose main technical specifications

Table 5.1: Accelerometer technical specifications

Tri-axial accelerometer TAA2230	
Parameter	**Value**
Frequency range	1 - 15000 Hz
Sensibility	100 mV/g
Dynamic range	+/- 50g peak
Temperature range	-50 / +151 Â°C

are reported in 5.1, is chosen as the most suitable sensor to measure vibrations [5].

The literature reports that the transition of the slag from the ladle to the tundish is due to the formation of a sink vortex, that sucks the slag into the nozzle [80]. Consequently, in the vibration signal three main principal pouring status can be identified [79]: pure molten steel status, the stable state before sink vortex formation, mixing slag status, the unstable state in which sink vortex appears and vibration amplitude becomes smaller, and all slag status, the stable state after vortex penetration [5].

As the acquisition of vibrations is done on the continuous casting plant, the environment may appear noisy and add lot of disturbance in the signals. Here, applying a good filter on vibrations is very important in order to analyze only those frequency components which contain information of steel and slag behavior. The study of filters is a topic that overcome the purpose of this thesis, then the figure 5.2 will present the signal filtered with a Chebichev filter used in the algorithm.

Finally, the filtered signals are elaborated on site with an industrial PC according to the edge computing paradigm and the trigger com-

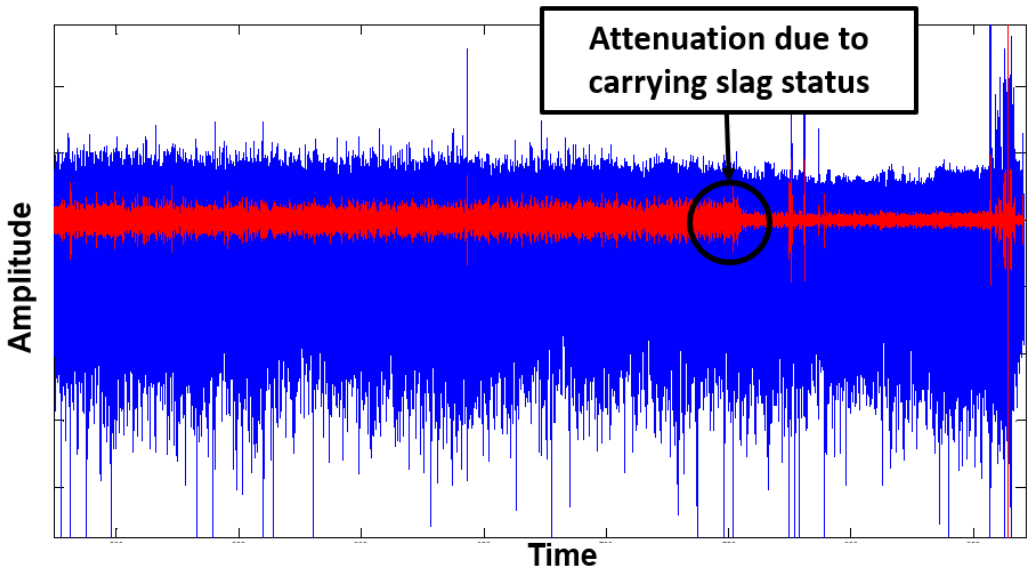


Figure 5.2: z-axis acceleration filtered (source [5])

mand is sent to a PC which manage the whole plant. The PLC shares also data with the production and information systems, thus the status of CCP can be propagated until it is shared remotely.

5.1.3 Results

After the experimental campaign, the company used as pilot declared a Slag Detection Accuracy (SDA) of 95% against our forecast that conjectured the index around 90%. Although the electromagnetic sensor is estimated with a SDA fo 98%, the VSD sensor costs more than 1000 times less compared to the previous. Now, after a period of test, the definitive architecture of the VSD system is shown in figure 5.3. Briefly, the signals acquired from the plant are quickly elaborated in an industrial PC near the plant, the result of the elaboration is secondly sent

to a PLC which manage all the actuators in the plant. The company

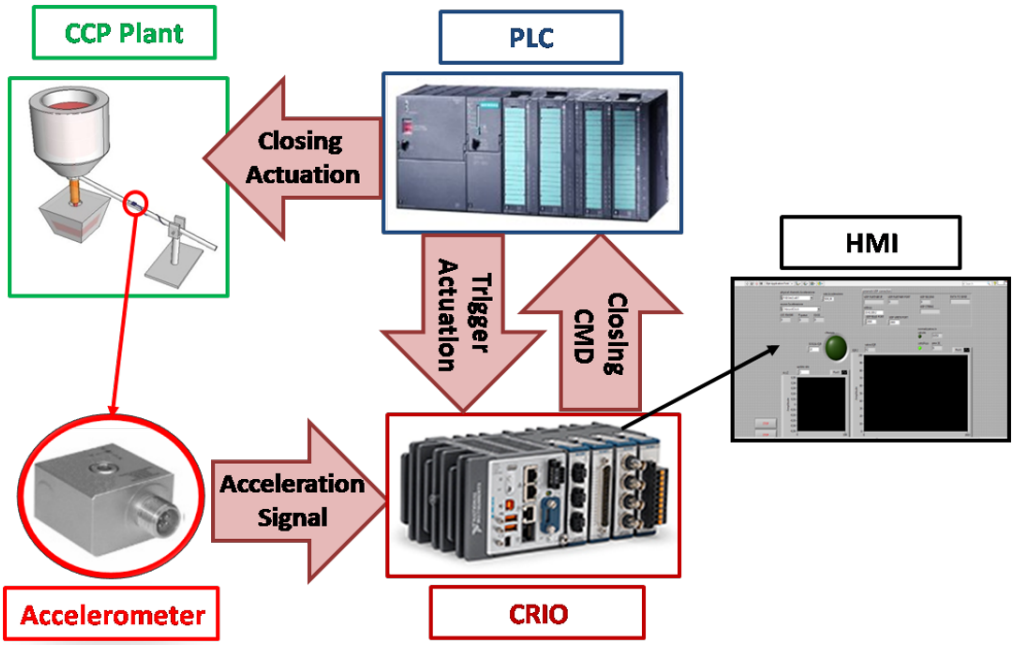


Figure 5.3: VSD system architecture (source [5])

decided to use their own GUI developing new panel pages in the current production monitoring system. The aggregated values of triggers and thresholds, with all other sensible data of the plant, are stored in a NOSQL database for metrics used for afterwards quality analysis and correction of errors.

5.2 Fault detection in circuit breaker

Maintenance strategies have evolved from standard cyclic maintenance to more advanced approaches like Condition Based Maintenance and Predictive Maintenance. Even if lot of work has been done regarding

these methodologies applied on machines or industrial plants, the same care is not applied to other critical devices. In the case of Low Voltage Circuit Breakers, in particular for critical applications, conservative approaches are usually applied. That implies a scheduled replacement of devices could occur with a significant Remaining Useful-Life. Not only for the cost of devices, but mainly for the scheduling of maintenance activities, this conservative approach force downtime and limited service of the production plant. The work proposed deals with Pattern Recognition techniques supporting the first step of prognostics of an electrical circuit breaker. The goal is to develop a methodology for Fault Detection.

5.2.1 methodology

Due to the complexity of the functional mechanism of this kind of equipment, hard to be physically modeled, at the base of the diagnostic model for LVCB, a Process History Based method has been used following the steps described below [6]:

1. Data collection. In absence of basic knowledge of the physical model, the system is over sensorized. Due to the high frequency signals to acquire, the data analysis must be performed on the gateway.

2. Feature extraction. Once data have been collected, a first elaboration acts to distinguish the normal operational status and the fault operational status is done.

3. Feature selection. The step 2 can generate several features in each fault operation status. In this step, the features identified are filtered and the most significant ones selected.

4. Classifier design: This step aims to discriminate predicted operation status of the system.

5. Evaluation. The result is one or more KPIs which enable the decision support.

For critical assets like the one used in the project, the rarely entry into operation and the large amount of data needed for the analysis required to arrange a test desk where perform a series of proof in row. In fact, as mentioned by the company customer of the project, the average life of the device without maintenance is 20.000/25.000 operations and several devices must be destroyed in order to have a good data set to start the feature extraction phase.

An aspect to take into consideration during the first 3 phases is that the device is over sensorized then the feature selection phase can only pick up those features which share signals from the same set of sensors. In fact, only the sensors needed are kept and the classifier streamlined to generate a productive release of the algorithm. Figure 5.4 shows the architecture for the system object of the study. The elaboration of the signals is performed at gateway level due to the wide bandwidth required for the transmission of data from the gateway to the cloud.

In particular, data collection phase in a first stage of the work involved in: tri-axial accelerometer placed on mechanical contacts, cur-

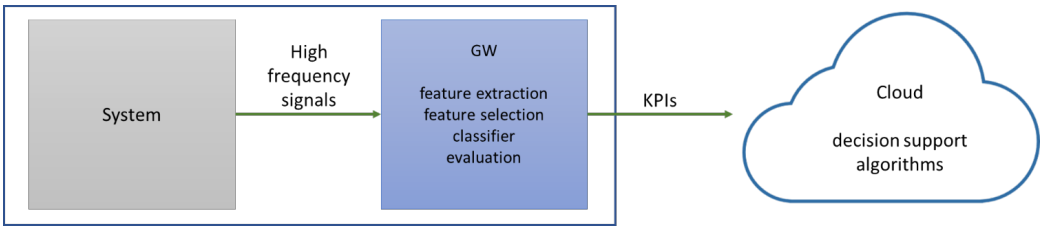


Figure 5.4: System architecture

rent sensor and angular position sensor of the main shaft. The feature extraction and selection phases highlighted one main feature based on analysis of vibration, then only the accelerometer sensors kept in the production system. A recap of the feature extraction and feature selection is briefly here. Regarding feature extraction were used the following algorithms: DTW, EMD and FFT Slicer for acceleration elaboration, RMS for current absorption elaboration and amplitude and duration for angular position of encoder elaboration. Feature selection, on the other hand, generates new features combining the ones already existing starting from the feature extraction phase. This is done using algorithms like Principal Component Analysis (PCA) and Fisher Discriminant Analysis (FDA). While in this section the focus is on the data collection, further information of the proposal can be find in [6].

However, the results obtained, shown in figure 5.5, indicate that from the start up to the first occurrence of the intermittent failure, represented by the dashed line, the classifier associates the conditions of functioning of the LVCB to the class of normal operation (blue line), while from the replication subsequent to the termination of service, the classifier associates the conditions of malfunctioning (red line). It

follows that the developed classifier is capable to discriminate between the normal operation and malfunction [6].

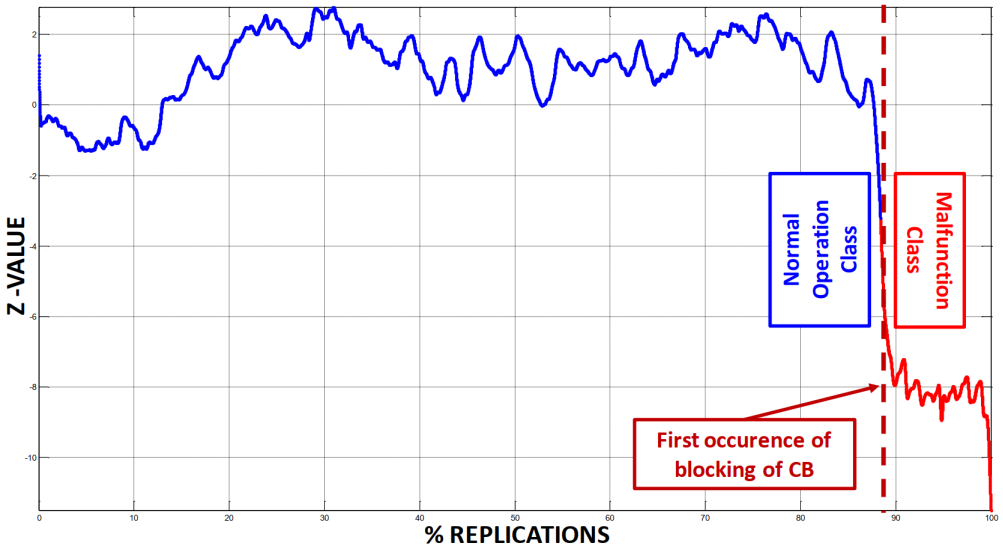


Figure 5.5: Result of classification for the component analyzed (source [6])

Since the circuit breaker has not got any interfaces to transfer data from sensors, the gateway involves the network sensors with its acquisition board for high frequency collection. The large amount of data to elaborate forces to over boost the computational power of the gateway. Even though, only the KPIs post elaboration are transmitted to the cloud, sending also the acquisition signal related to a fault is a best practice for further elaboration and recognition of the fault itself with more complex algorithm at the cloud level. Fortunately, the elaboration of the signal is performed only when the device opens its contacts and this may happen among 0 and 3 times a day reducing the effort of the CPU. This also means that I circuit breaker may also never reach 20.000 operations in the whole life of the plant, that is why the

company is considering whether industrialize the application or not.

In conclusion, the paper reports the application of a pattern recognition methodology applied to low voltage circuit breaker [6]. From an industrial perspective, this item is enabled by a reliable acquisition of signals since the device is often placed in tiny and narrow space. Moreover, this kind of devices are linked with voltage up to 1500V and reliability of hardware such as sensors and gateway is critical for a correct result of the elaboration.

5.3 Molds monitoring in press machinery

This section treats the maintenance of molds in manufacturing industry. In this field of application, maintenance is enabled by traceability of molds and monitoring of data process during the molding operations.

According to ISO 9000, -Traceability is the ability to identify and trace the history, distribution, location, and application of products, parts, materials, and services- . To enable traceability in the manufacturing industry, companies use different technologies that can be grouped into two classes depending on how the traceability system works. In particular, 2 main technics can be distinguished: optical technologies and non-optical ones. Traceability is enabled by optical technologies when assets are identified by code directly printed on their surface and readable by optical sensors. Most known methods are laser etching, electrochemical etching, dot peen, labeling and ink-jet. On the other hand, traceability is enabled by non-optical technologies when assets are identified by RFID devices. Respect data matrix and QR code

used in optical technics, RFID can be readable and writable. This means that few information regarding the current status of the asset monitored can be stored.

The meaning of traceability introduced above is necessary to understand the application explained in this section. The application is particularly suitable in the manufacturing industry, where companies share their assets with suppliers in outsourcing production. In fact, by using automated traceability technologies combined with an appropriate IT infrastructure [**?**], real time monitoring of the location and status of the assets are enabled, allowing the application of preventive maintenance (PM) policies and optimization of assets movement.

The application must solve 2 main problem: improve the service of maintenance and monitor the use of the molds. In particular, improving the service of maintenance means schedule the next maintenance operation dynamically respect the effort of every mold. To do that, various techniques can be used, for example image analysis of molds or pieces produced, analysis of process parameter and simpler adding a static threshold on the number of shots performed.

On the other hand, monitoring the use of the molds means continuously check and verify the status of the production. Although the second feature implies the first, scheduling the maintenance from data production is not an effortless operation. In fact, the calculation of Remain Useful Life (RUL) is a hot topic in literature and required reliability of the result [81].

Starting from those features, the platform is integrated with the

following functionality:

1. Review historian data by plotting intuitive graphs and displaying raw data in a table.

2. Allow to upload photos using a smartphone with a dedicated application.

3. Collect mold data automatically when it is working on the press machine.

4. Allow to retrieve information from mold wherever it is, also offline.

5. Retrieve machine setup information from a server to help workers in setting up operation.

6. Notify critical condition of molds and call for maintenance.

Analyzing the requirements above, it seems clear that the system must be composed by a web application, a smartphone application, a mold identifier and a device for automatically collect production data.

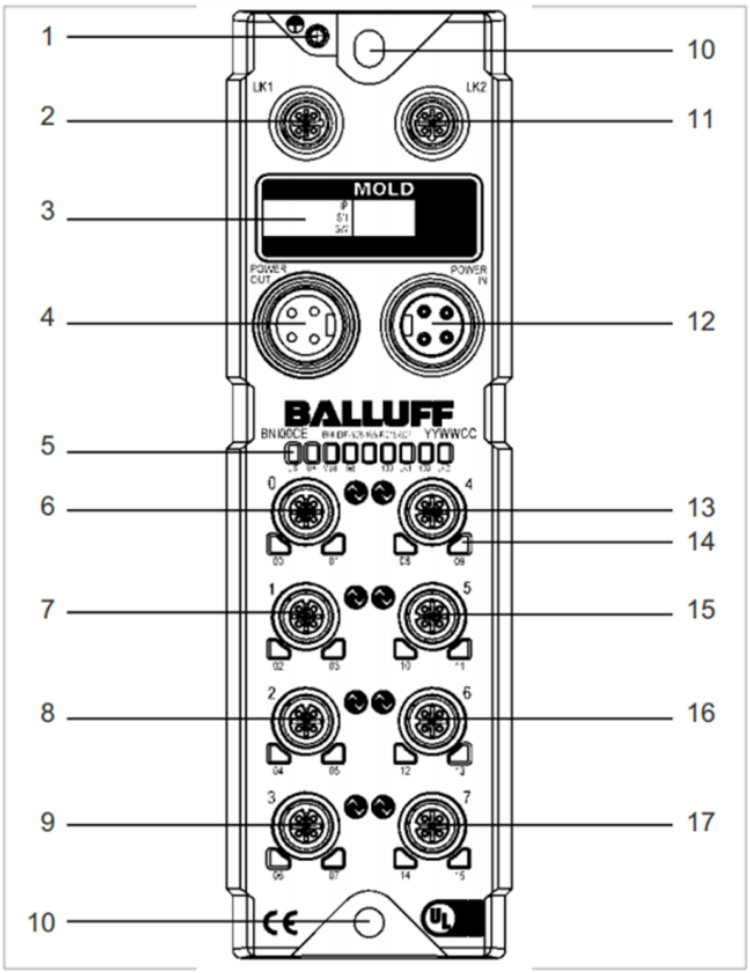
A web application allows users to access to information wherever in the world only with an Internet connection. Using a responsive layout, the information is displayable on any kind of device. In order to satisfy the requirement number 4, also a smartphone application is essential. The smartphone app contains locally on the phone all the functionality needed for the offline consultation. Based on the requirements above, a deep analysis on hardware and software technologies available has been done.

5.3.1 hardware requirements

Dealing with provider's machines, the hardware device is required to be plug & play and as less invasive as possible. Moreover, a standard application requires standard protocol (the topic of field-bus protocols was largely treated in chapter 3), this is not always possible since signal data come from machines. The solution is to install a stand-alone device able to capture the shot.

Discovering the market solutions, the attention fell down on a new device which is mounted on the press machine and acquires, through a proximity sensor, the shot instant. This device writes through NFC technology the information of the shot on a RFID tag screwed on the mold. Simultaneously, the same data are sent by HTTP protocol on a web server.

Searching on the market for a suitable device, a prototype built by Balluff caught the attention. The structure of the device is shown in Figure 5.6 and briefly involves in 2 ethernet ports, 1 shot counter trigger port, 1 port for a smart-light which report visually the state of mold and 1 or more ports suitable for RFID reader/writer. The communication between sensors and actuators of the device is based on IOlink protocol. Using more than one RFID writer, the device can manage multipart molds collecting data from different TAG at the same shot. In fact, it is not unusual in steel industry having molds composed by multiple parts, one which form the main unit (the skeleton) and several other which can be attached to the main part in order to constitute various products.



1	Ground connection	10	Mounting hole
2	Ethernet Port 1	11	Ethernet Port 2
3	Display	12	Power IN
4	Power OUT	13	Port 4 = N.C.
5	Status LEDs	14	Pin/Port LEDs
6	Port 0 = Shot counter	15	Port 5 = N.C.
7	Port 1 = Smartlight	16	Port 6 = N.C.
8	Port 2 = RFID unit	17	Port 7 = N.C.
9	Port 3 = N.C.		

Figure 5.6: Balluff device

The Balluff device can thus collect shot data and register both, new update on the RFIG tag and new record on the cloud history. Data on the tag are transferred using NFC technology [82], this technology also

enables the use of smartphones in production for maintenance purpose in an industry 4.0 scenario. Data are sent on cloud architecture via HTTP protocol using JSON format message. The message involves information such as tag ID, warning and alarm thresholds, site where the mold is placed, next maintenance activity date prediction, total number of shots and timing information.

Considering the data listed above and the requirements specified for the application, mold photos capturing feature, fundamental for RUL analysis, must be implemented separately. So, the use of the smartphone can enable this feature allowing maintainers to take pictures of molds.

5.3.2 software requirements

All the information collected by the Balluff device and the smartphone application is further organized in the software architecture and available either through web browser or the smartphone application. Although the device provides a prediction of next maintenance activity already, algorithm for RUL based on image elaboration can be implemented for a deeper analysis.

Exploiting the power and services on cloud infrastructure, the software application is deployed on Amazon Web Services. This provides, as shown in Figure 5.7, support for smartphone authentication, serverless implementation of function for data ingestion, noSQL database and virtual machine for web application.

According to the software architecture, shot data can follow 2 dif-

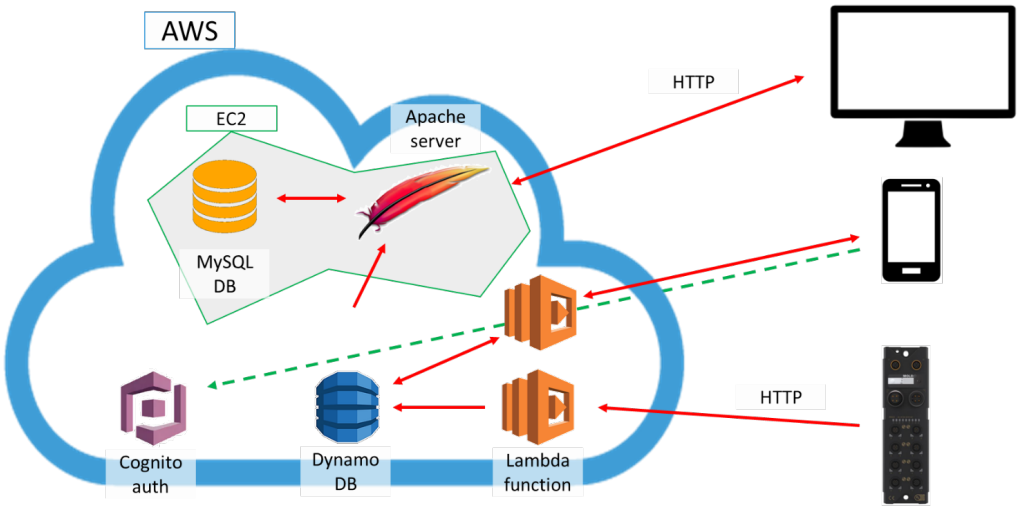


Figure 5.7: software architecture

ferent flows: from Balluff device or from smartphone application to the server. In any case, data are managed by Lambda function and a new record pushed into the Dynamo database and subsequently retrieve for visualization or analysis purpose. Since the smartphone application needs network authentication, it has 2 levels of access: the first occurs when the smartphone is offline, in this case the cloud synchronization is disabled. The second occurs when the smartphone is online and requires credential authentication through AWS Cognito. The authentication ensures the log of every activity on the cloud in addition to increase the level of security of the application.

The data management is performed using 2 different DBMS, the first leans on non-relational solution while the second needs a relational structure. In particular Dynamo DB stores shot records, it ensures fast insertion for unstructured data order by a timestamp and separable for

a key. MySQL, instead, provides the support for the web application, account profiles, privileges, setup sheets and documentation bound to molds are stored on it. The difference and strengths between the 2 solutions have been well explained in section 2.5.

Another important requirement is the interface with the monitoring system. In this way, process data can be associated with shot data. Parameters such as pressure, temperature and alerts can improve the estimation and reliability of algorithms as RUL.

5.3.3 results

The platform described above is implemented and tested in a plastic component manufacturer. In particular, the case study is focused on the injection molding process, a widely diffused process in the world of plastic production [83]: it expects that molten plastic material is injected at high pressure by presses inside the molds, which vary according to the type of product.

With system implementation, management expects that logistics costs will reduce, through the structuring of warehouse management, process control will increase, through setting automation of the process parameters, the product quality will improve, through the application of preventive maintenance policies, support for decision-making processes will increase, through automatic generation of periodic reports.

Installation of the system consists of the following activities:

- Subdivision of the warehouse into predefined compartments, in order to univocally define each possible location of the mold.

- Mounting of the NFC tags on the company molds, in order to identify them univocally, and relative initialization with the process parameters.

- Installation of the gateway, described above, one for each injection molding machine, in order to enable data transfer between molds and information systems.

- Assembly of an NFC Writer / Reader on each machine in order to identify the molds once they have been installed and to write the operating status on them.

- Installation of a proximity sensor to count the number of beats and to trigger the writing of data.

The interconnection is given by the interfacing of the gateways with the company Wi-Fi network through the use of special routers.

The benefits found following the implementation of the system are as follows:

- From the unstructured management of the warehouse to the structured management: in the past the mold research within the warehouse was chaotic and dispersive, after system implementation the operator can view the status of the mold on the smartphone and its precise location in the warehouse, once the mold ID has been entered. The same happens during relocation, when the operator places the mold in the position of competence. As a result, logistical times are reduced by 20%.

- From manual setting of the machine to self-configuration: in the past, the machine configuration was manual, after implementation, once the mold is placed in the machine, the ID is detected and through the high-level interfacing between the asset tracking platform and the monitoring and control system the machine is auto-configured. A reduction of non-conformities due to an incorrect insertion of the process parameters of 2% is noted.

- From visual analysis to preventive maintenance: in the past the operator visually checked the condition of the mold, after the implementation the number of measures for each mold the status is monitored in real time, in order to compare it with estimated thresholds, representing the permissible wear status. As a result, maintenance times are reduced of 7%.

- From paper information to data digitization: in the past, the transfer of data from production to administration took place on paper, the system implementation enables automatic data transfer and allows the generation of reporting for decision support.

5.4 SCADA system in textile machinery

Today, globalization and more stringent requirements push managers to optimize all systems involved in their organizations and in particular the operations processes which have great impact on productivity. In this scenario, concepts as availability and time to repair play a key role for achieving the desired Key Performance Indicators. To this end,

the advanced of communication technologies can support practitioners in managing more efficiently production processes using advanced approaches to plant / machine monitoring. Traditionally, Supervisory Control And Data Acquisition (SCADA) systems can support operations activities (mainly production and maintenance) and allow operators to monitor and supervise the entire production process. As the poor and simple products require high technology production process, textile industry follows the trend and firstly joins the most recent technologies in monitoring system field [84].

The work proposed below is a web-based SCADA for textile industry ables to collect data from weaving machines (looms). The work was commissioned by an important weaving machine manufacturer which operates worldwide. The system stores data in a database and provides a web view where users observe real-time data and historical data. This solution addresses also the task to analyze data to extract productivity and efficiency indexes. From Industry 4.0 point of view, this application accomplishes the task to provide an instrument for CPSs monitoring (each loom has its physical realization but also a virtual counterpart), according to the paradigm of the Internet of Things (every loom has its unique IP address). During normal operation these machines are frequently subject maintenance activities since the threads that compose warp and weft may break due to the tension applied. Thus, whenever a thread breaks, the loom stops and cannot weave until the thread is not adjusted. This kind of operation is not typically schedulable or predictable and requires the intervention of a human operator to re-

pair the fail and restart the machine. The proposed solution solves the problem to notify this information to human operators in the shortest possible time, considerably reducing loom downtime [84].

5.4.1 system architecture

The Figure 5.8 shows the global architecture of the monitoring system. In that architecture, 3 main components can be identified.

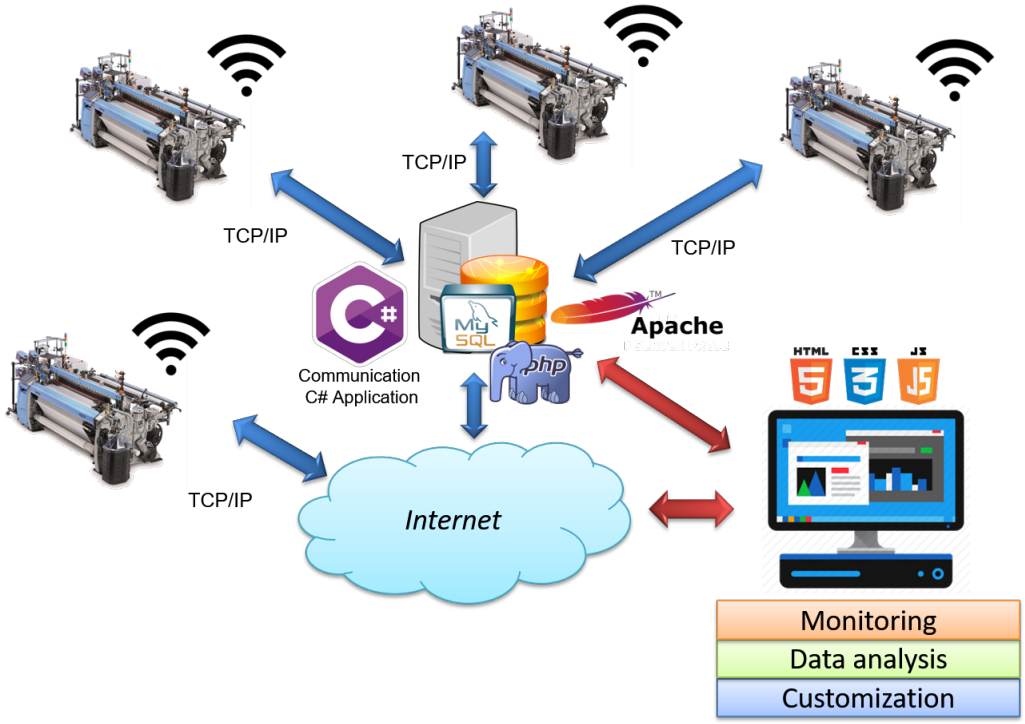


Figure 5.8: Monitoring system software architecture

- **Looms**, the machines monitored and supervised. They dispose of a Internet module, by means they can communicate over TCP/IP and through a proprietary protocol with other IP entities. Looms, through a polling policy, communicate their current status (e.g.

motor RPM, warp tension, weft density etc. etc.) and, eventually, the occurred faults. In particular, when a single parameter of the loom changes, a new event containing all the information about the machine status is generated and queued in an event buffer. A TCP client can then request for 1 or more elements of the mentioned buffer.

- **Server**, whose main tasks are collect data, store them into a database and provide information to client devices. Periodically a data collector module installed on the server interrogates looms, that are reachable through their IP address. The information received is saved into a relational database, in this case MySQL. The consultation of real-time data and alerts is provided by a web server. For this reason, Apache HTTP server has been adopted with PHP server-side scripting language.

- **Clients**, Typically clients are computers, tablet or smartphone which through a web browser show desired information according to user permissions.

Working on TCP/IP protocols, looms, server and clients can potentially be in three different LAN network accessing on the Internet connection. This means that a single server can monitor looms placed in different plants around the world. Although monitor a loom in the opposite part of the world introduces high latency on the TCP protocol, the large deadlines on the looms protocol do not compromise the functionality of the whole system.

5.4.2 system features

Periodically polling looms, it is possible to monitor machines status and get information about faults [84]. From the users' point of view, a global overview of machines would be very useful possibly with a graphical representation that reflect physical or logical topology of the plant or the plants. The Figure 5.9 shows an example of smart dashboard of the system proposed. In this case, the dashboard represents a physical disposition of looms in a weaving room of the plant. Each rectangular block is a machine and contains few important parameters of interest.

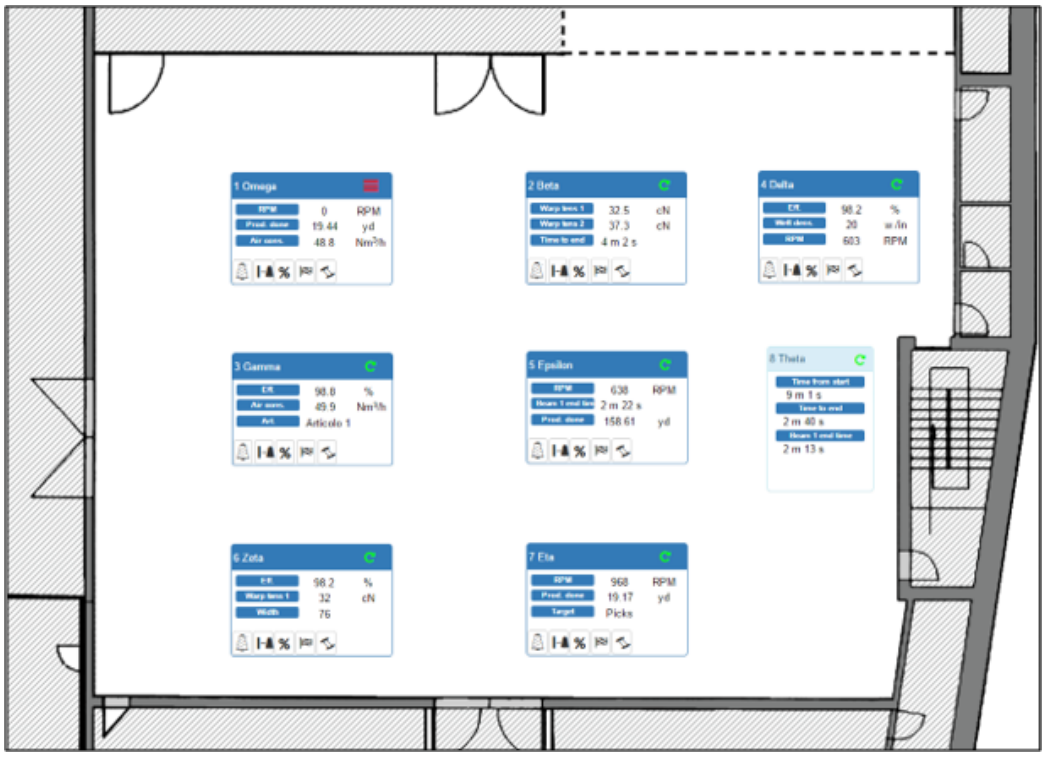


Figure 5.9: Example of dashboard

When a fault occurs, a message notifies the maintainer to fix the

problem, watching the dashboard the fault is visible on the right-top corner of the block representing the loom.

Another functionality concerns about elaborates fault information for tracking defects in piece of cloth: when a loom stop weaving due to a fault or a manual stop, in the piece a defect may appear and the defect is much more remarked much more the stop lasts.

In addition to the functionalities illustrated, the system provides historical analysis over data collected by means a series of filters selectable by the user.

The system is now an important product of the company which has gained competitiveness on the market offering an low cost SCADA system to small and medium enterprises.

5.5 Simulation for plant scheduling

The last case presented is focused on simulation. As explained in chapters 2 and 4, the simulation can be either a tool integrated in a monitoring system or a stand alone tool used for first analysis on a production system or a part of it. A production system is a complex structure characterized by several rules to follow and different goals to achieve taking into account various constraints set by the environment. Many actors are involved in the design phase and in the production, planning and control activities [85]. These systems, during their life, need to continuously adapt to new market requirements, increasing the customization level of the products, reducing cycle time, keeping high quality level at lower costs. In order to achieve these objectives engineers must con-

sider reconfigurability and flexibility aspects in the design phase of the plant.

The work proposed deals with the reconfiguration and scheduling optimization of a complex Flexible Manufacturing System and aims at proposing a complete methodology from problem statement to analysis and solution that leverage on Discrete Event Simulation (DES). Furthermore, a case study in an Italian company working in the automotive sector has been carried out in order to test and validate the developed methodology. In more details, the application proposed aims to optimize the production scheduling minimizing the Work In Process (WIP) after introducing a new working phase in the plant. To this purpose, DES is used since it allows to implement several *what if* scenarios and define the best configuration, taking into account the several constraints of a real production system. About that, a deep analysis of the plant is needed first, a strong data gathering performed and an intensive simulation activity run.

5.5.1 simulation process

DES is a form of computer-based modeling that provides an intuitive and flexible approach to represent complex systems. The simulation process can be recount in 7 steps:

1. *Determine the goals*: goals suggest which areas are the focus.

2. *Collect data*: in this step is important to analyze the actual system and check out what information are relevant to build the model.

3. *Build the model*: An algorithm (the model) is built using dedicated softwares.

4. *Validate the model*: before simulating and collecting the results, a validation phase is fundamental. When the model is declared not validate, the process must restart from step 2.

5. *Simulate*: define a simulation campaign and run it on acceptable time horizon.

6. *Analyze the results*: results must be critically analyzed to decide whether represent a valid information for the final goal. A negative answer could force the simulation process to restart from step 2

7. *Final documentation*: as a final step, the documentation report should be filled with the information extracted from the simulation.

To summarize the list above, the figure 5.10 represents graphically the flow of operations. Even though the thesis is focused monitoring systems and data acquisition, data is the first requirement for simulation: step 2 of the list above highlights how data are fundamental. Picking up data from a production database favorite the work and lighten the effort.

Anyway, for this specific work, the goal is to perform *what if* analysis pretending the behavior of production after placing a new phase in the process. To do that, the next steps were followed:

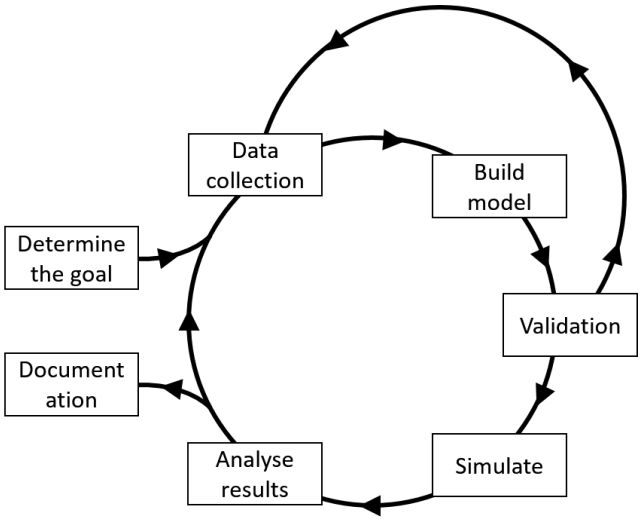


Figure 5.10: Iteration of simulation process

1. AS-IS model definition. Here the current plant is modeled with a specific software. Processing times, failures, resources, order scheduling and all the other information are used to define constraints and activities in the model.

2. Validation. Here the AS-IS model is compared to the real plant. Simulation results and real data collected are compared and various indexes of error calculated on chosen KPIs in order to obtain a degree of likelyhood.

3. TO-BE model definition. Once validated the model, it is modified with the integration of the new production phase and a series of scenarios defined to test several scheduling strategies.

4. What if cases. The scenarios defined in the previous phase are then simulated and the results analyzed.

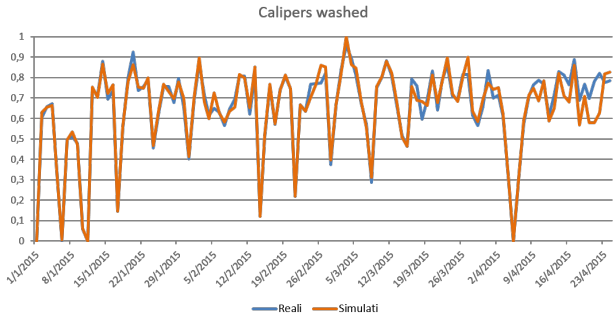


Figure 5.11: Comparison between real and simulated production

5.5.2 results

First of all, figure 5.11 shows the results provided by the validation. For semplicity, only one KPI analysis is shown. However, the likelyhood between KPIs was performed using the Relative Squared Error (RSE) (5.1), where y_i is the real amount of pieces produced in day i and $\hat{y}_i$ is the same value obtained in the simulation environment.

$$RSE = \frac{\Sigma(y_i - \hat{y}_i)^2}{\Sigma y_i^2} \tag{5.1}$$

For the validation phase, the worst RSE obtained is 4.87% since the limite established to accept the model was 5%. Nevertheless, the what if analysis showed a dangerous backlog of pieces in front of the new production phase due to a different setup constraints compared to the previous production phases. The solution proposed was to optimize automatically the schedule in the early phases of production in order to carry out containers with the right constraints to be processed in the new plant. Next steps involve the test of the scheduling proposed

and the integration of it into the SCADA system of the company.

Chapter 6

Conclusion and Future Work

In this thesis, a new architecture for monitoring and SCADA systems was presented. Chapter 2 has given a review of the state of the art of web technologies, distributed systems architecture and DBMS available focusing on how the improvements in such technologies have favored the spreading of web platform to deliver services inside the company or towards customers. The even increasing of computational performance and bandwidth offer day by day new solutions in this field and then even more applications are migrating towards web services. These considerations are the starting point of the design of the architecture subject of the thesis compliant with SOA and accessible from everywhere worldwide. Moreover, a critic against the current literature has been moved, due to the lack of topics related to communication gateway-server not in terms of protocol or technology used, but at the information level.

Chapter 3 treats the general features of the architecture and the mechanisms which the monitoring system is based. In particular, how data flows, how the system got configured and what database solution was chosen in order to increase the performance. In fact, the database was identified as the most critical part of the system in terms of performance. The objectives of the architecture are the followings:

- Providing a monitoring system which ensure to avoid single point of failure using a distributed approach.
- Collecting data from different machines using several protocols without customize the tool but configuring it with a JSON file remotely.
- Storing any type of data creating the structures needed on the database, no matter the protocol used by the machine.
- Displaying the data collected through dashboard highly customizable by an editor in the platform.
- Planning a way to introduce artificial intelligence by means machine learning algorithms for predictive maintenance or general data analytics.
- Keeping the platform available for every kind of production.

These objectives have been achieved by developing a web platform using the latest technologies available. In particular, choosing python as programming language for the server side platform, it throws open for implementations of new intelligent modules and supported by a

big community of developers who make available even more libraries. Installing the platform on cloud and designing the infrastructure following the cloud principles, single points of failure are averted. For the rest, the innovation dwells in the JSON configuration which set all the structure where data are stored and dynamically builds web contents to customize the HMI. This configuration, managed at cloud level, allows moreover to easily implement a feedback of the information to the machine or plant monitored.

Furthermore, chapter 4 provided a description of current problems companies are facing and want to overcome in order to provide new services and improve the satisfaction of customers. Here, an overview of topics which predictive maintenance, quality control and production analysis has been presented also introducing a brief content of the literature of those. The problems explained were an introduction for a series of cases study where the use of the monitoring system proposed can overcome the issue or even helps to reduce the disadvantage of implementing none solution.

Finally, chapter 5 explained a series of cases study where the system, or part of it, has been used. Although the topic of predictive maintenance is hotly followed, it is not the only field of application where a monitoring system like this can be installed. In fact, first of all, the quality of a production is measured based on the quality of data which are acquired and the n elaborated to extract useful indexes to analyse the production.

Next steps in the development of the system are the extension of

bus protocol drivers in order to increase the number of machines integrable in the system and improve the mechanism to integrate custom algorithms for data analytics. Moreover, the implementation of a VPN connection from the system dashboard to the machine HMI for remote diagnosis and maintenance.

Nevertheless, final considerations of the work proposed are:

- Industry which wants to adopt a monitoring system solution able to easily scale, add, remove and edit rules in data collection, provide a centralized web-based point of access to all the data and apply sophisticated algorithms for analysis should move towards a cloud based system like the one explained in this thesis
- Systems for small stand alone applications which do not need to be integrated with any other software and temporary test purpose systems can continue to adopt old style architecture since the effort to convert these applications does not really provide performance improvements in the production line
- The privacy and ownership of the data gathered is not a good reason not to prefer a cloud deployment instead of an on-premise one

Appendix A

Configuration: customize the system with a file

Since section 3.2 introduced the concept of configuration to customize the monitoring system and made it flexible to acquire data from different machines, this thesis has not treated the topic in detail. In this appendix, the whole example of the snap proposed is shown and explained.

```
{
"config":"MODBUS_TCP",
"device_id":"COS01",
"device_ip":"192.168.0.2",
"device_port":502,
"polling_time":1000,
"store_period":"5",
```

```
8  "machine":{
9      "vars":[
10         {
11         "reg_address":1,
12         "length":2,
13         "datatype":"int",
14         "var_id":"totale_cicli_macchina",
15         "endian":"big",
16         "store2db":false,
17         "gw_type":"variable"
18         },
19         {
20         "reg_address":3,
21         "length":2,
22         "datatype":"int",
23         "var_id":"totale_lotto_prodotto",
24         "endian":"big",
25         "store2db":false,
26         "gw_type":"variable"
27         },
28         {
29         "reg_address":5,
30         "length":2,
31         "datatype":"int",
32         "var_id":"totale_pezzi_scatola",
33         "endian":"big",
34         "store2db":false,
35         "gw_type":"variable"
36         },
37         {
38         "reg_address":7,
39         "length":2,
40         "datatype":"int",
41         "var_id":"totale_pezzi_ok",
42         "endian":"big",
43         "store2db":false,
44         "gw_type":"variable"
45         },
46         {
47         "reg_address":9,
48         "length":2,
49         "datatype":"int",
50         "var_id":"totale_pezzi_ko",
51         "endian":"big",
52         "store2db":false,
53         "gw_type":"variable"
54         },
55         {
56         "reg_address":21,
57         "length":1,
58         "datatype":"int",
59         "var_id":"target_macchina",
60         "endian":"big",
61         "store2db":false,
62         "gw_type":"variable"
63         },
```

```
64          {
65          "reg_address":151,
66          "length":1,
67          "datatype":"int",
68          "var_id":"macchina(ignore)",
69          "endian":"big",
70          "store2db":true,
71          "gw_type":"alarm",
72          "extract":[{
73              "mask":1,
74              "value":1,
75              "var_id":"pc_offline"
76              },
77              {
78              "mask":2,
79              "value":2,
80              "var_id":"attesa_automatico_remoto"
81              },
82              {
83              "mask":4,
84              "value":4,
85              "var_id":"mancato_transito_pallet"
86              },
87              {
88              "mask":8,
89              "value":8,
90              "var_id":"allarme_lotto_terminato"
91              }
92              ]
93          },
94          {
95          "reg_address":301,
96          "length":1,
97          "datatype":"int",
98          "var_id":"velocita_macchina",
99          "endian":"big",
100         "store2db":true,
101         "gw_type":"variable"
102         },
103         {
104         "reg_address":401,
105         "length":2,
106         "datatype":"int",
107         "var_id":"tempo_automatico",
108         "endian":"big",
109         "store2db":true,
110         "gw_type":"variable"
111         },
112         {
113         "reg_address":402,
114         "length":2,
115         "datatype":"int",
116         "var_id":"tempo_allarme",
117         "endian":"big",
118         "store2db":true,
119         "gw_type":"variable"
```

```
        }
    ],
    "groups":[
        {
        "name":"Ascensore",
        "vars":[{
            "reg_address":102,
            "length":1,
            "datatype":"int",
            "var_id":"ascensore(ignore)",
            "endian":"big",
            "store2db":true,
            "gw_type":"warning",
            "extract":[
            {
            "reg_address":157,
            "length":1,
            "datatype":"int",
            "var_id":"ascensore(ignore)",
            "endian":"big",
            "store2db":true,
            "gw_type":"alarm",
            "extract":[{
                "mask":1,
                "value":1,
                "var_id":"allarme_inmanuale_ascensore"
                },
                {
                "mask":2,
                "value":2,
                "var_id":"allarme_portelle_ascensore"
                },
                {
                "mask":4,
                "value":4,
                "var_id":"allarme_blocco_ascensore"
                }
                ]
            },
            {
            "reg_address":307,
            "length":1,
            "datatype":"int",
            "var_id":"velocita_ascensore",
            "endian":"big",
            "store2db":true,
            "gw_type":"variable"
            }
            ]
        },
        {
        "name":"StazioneA",
        "vars":[
            {
            "reg_address":152,
            "length":1,
```

```
176                 "datatype":"int",
177                 "var_id":"macchinaA(ignore)",
178                 "endian":"big",
179                 "store2db":true,
180                 "gw_type":"alarm",
181                 "extract":[{
182                     "mask":1,
183                     "value":1,
184                     "var_id":"allarme_inmanuale_A"
185                     },
186                     {
187                     "mask":2,
188                     "value":2,
189                     "var_id":"allarme_portelle_A"
190                     },
191                     {
192                     "mask":4,
193                     "value":4,
194                     "var_id":"allarme_blocco_A"
195                     },
196                     {
197                     "mask":8,
198                     "value":8,
199                     "var_id":"allarme_manca_pezzo_A"
200                     },
201                     {
202                     "mask":16,
203                     "value":16,
204                     "var_id":"allarme_alim_pezzo_A"
205                     }
206                     ]
207                 },
208                 {
209                 "reg_address":302,
210                 "length":1,
211                 "datatype":"int",
212                 "var_id":"velocita_stazioneA",
213                 "endian":"big",
214                 "store2db":true,
215                 "gw_type":"variable"
216                 }
217             ]
218             },
219             {
220             "name":"StazioneB",
221             "vars":[
222                 {
223                 "reg_address":153,
224                 "length":1,
225                 "datatype":"int",
226                 "var_id":"macchinaB(ignore)",
227                 "endian":"big",
228                 "store2db":true,
229                 "gw_type":"alarm",
230                 "extract":[{
231                     "mask":1,
```

```
232                 "value":1,
233                 "var_id":"allarme_inmanuale_B"
234                 },
235                 {
236                 "mask":2,
237                 "value":2,
238                 "var_id":"allarme_portelle_B"
239                 },
240                 {
241                 "mask":4,
242                 "value":4,
243                 "var_id":"allarme_blocco_B"
244                 },
245                 {
246                 "mask":8,
247                 "value":8,
248                 "var_id":"allarme_manca_pezzo_B"
249                 },
250                 {
251                 "mask":16,
252                 "value":16,
253                 "var_id":"allarme_alim_pezzo_B"
254                 }
255                 ]
256             },
257             {
258             "reg_address":303,
259             "length":1,
260             "datatype":"int",
261             "var_id":"velocita_stazioneB",
262             "endian":"big",
263             "store2db":true,
264             "gw_type":"variable"
265             }
266         ]
267         },
268         {
269         "name":"StazioneC",
270         "vars":[
271             {
272             "reg_address":154,
273             "length":1,
274             "datatype":"int",
275             "var_id":"macchinaC(ignore)",
276             "endian":"big",
277             "store2db":true,
278             "gw_type":"alarm",
279             "extract":[{
280                 "mask":1,
281                 "value":1,
282                 "var_id":"allarme_inmanuale_C"
283                 },
284                 {
285                 "mask":2,
286                 "value":2,
287                 "var_id":"allarme_portelle_C"
```

```
                },
                {
                "mask":4,
                "value":4,
                "var_id":"allarme_blocco_C"
                },
                {
                "mask":8,
                "value":8,
                "var_id":"allarme_manca_pezzo_C"
                },
                {
                "mask":16,
                "value":16,
                "var_id":"allarme_alim_pezzo_C"
                }
                ]
            },
            {
            "reg_address":304,
            "length":1,
            "datatype":"int",
            "var_id":"velocita_stazioneC",
            "endian":"big",
            "store2db":true,
            "gw_type":"variable"
            }
        ]
        },
        {
        "name":"StazioneD",
        "vars":[
            {
            "reg_address":155,
            "length":1,
            "datatype":"int",
            "var_id":"macchinaD(ignore)",
            "endian":"big",
            "store2db":true,
            "gw_type":"alarm",
            "extract":[{
                "mask":1,
                "value":1,
                "var_id":"allarme_inmanuale_D"
                },
                {
                "mask":2,
                "value":2,
                "var_id":"allarme_portelle_D"
                },
                {
                "mask":4,
                "value":4,
                "var_id":"allarme_blocco_D"
                },
                {
```

```
                "mask":8,
                "value":8,
                "var_id":"allarme_manca_pezzo_D"
                },
                {
                "mask":16,
                "value":16,
                "var_id":"allarme_alim_pezzo_D"
                }
                ]
            },
            {
            "reg_address":305,
            "length":1,
            "datatype":"int",
            "var_id":"velocita_stazioneD",
            "endian":"big",
            "store2db":true,
            "gw_type":"variable"
            }
        ]
        },
        {
        "name":"StazioneE",
        "vars":[
            {
            "reg_address":156,
            "length":1,
            "datatype":"int",
            "var_id":"macchinaE(ignore)",
            "endian":"big",
            "store2db":true,
            "gw_type":"alarm",
            "extract":[{
                "mask":1,
                "value":1,
                "var_id":"allarme_inmanuale_E"
                },
                {
                "mask":2,
                "value":2,
                "var_id":"allarme_portelle_E"
                },
                {
                "mask":4,
                "value":4,
                "var_id":"allarme_blocco_E"
                }
                ]
            },
            {
            "reg_address":306,
            "length":1,
            "datatype":"int",
            "var_id":"velocita_stazioneE",
            "endian":"big",
```

```
400             "store2db":true,
401             "gw_type":"variable"
402             }
403         ]
404         },
405         {
406         "name":"Discesore",
407         "vars":[
408             {
409             "reg_address":158,
410             "length":1,
411             "datatype":"int",
412             "var_id":"discensore(ignore)",
413             "endian":"big",
414             "store2db":true,
415             "gw_type":"alarm",
416             "extract":[{
417                 "mask":1,
418                 "value":1,
419                 "var_id":"allarme_inmanuale_discensore"
420                 },
421                 {
422                 "mask":2,
423                 "value":2,
424                 "var_id":"allarme_portelle_discensore"
425                 },
426                 {
427                 "mask":4,
428                 "value":4,
429                 "var_id":"allarme_blocco_discensore"
430                 }
431                 ]
432             },
433             {
434             "reg_address":308,
435             "length":1,
436             "datatype":"int",
437             "var_id":"velocita_discesore",
438             "endian":"big",
439             "store2db":true,
440             "gw_type":"variable"
441             }
442         ]
443         }
444     ]
445 }
446 }
```

As printed, the configuration is divided in 2 parts, the **header** and the **body**. The header contains the information needed to connect the gateway to the machine and how to manage data. In fact, the type of protocol (MODBUS_TCP) , the IP address (192.168.0.2) and the

IP port (502) are specified here. Not in this case, but for other types of protocol, the header may contain credentials to securely connect to the machine and/or path to certificates. The header also contains instructions about polling time (expressed in milliseconds) and the time between saving into the database consecutive samples of the same variable (expressed in seconds).

Differently, the body of the configuration contains the variable to read from the machine. These variables are organized hierarchically in 4 levels, from the top: *machine* level, to the bottom: *device* level going through *group* and *subgroup* levels. The user can decide to use all or a limited selection of levels, but always from the top to the bottom. For instance, the machine level can contain a list of variables and a list of groups containing in turn other variables and a list of subgroups. Anyway, the level machine can not contain directly a list of devices without containing groups and subgroups first. However, a configuration may contain hundreds or even thousands of variables split hierarchically in different levels.

Depending of the protocol used, every element in the body contains common and custom fields. In the example presented above is shown a MODBUS configuration and presents the following custom fields:

- **reg_address**, address of the 16 bits register where the variable begins.

- **length**, number of consecutive 16 bits registers which contains the variable.

- **endian**, indicates with the string *big* or *little* if the sequence of registers is to sort big-endian or little-endian, in case the field length is equal to 1, this field is of course irrelevant but mandatory.
- **extract**, optional array of variable needed when a single register contains multiple boolean values. This is very common using MODBUS protocol, an extract element contains a **mask**, used to perform an AND operation, and a **value**, used to compare the result of the previous operation and then produce a boolean value.

In any case, every variable is characterized by the following fields:

- **var_id**, the identifier of the variable, it must be unique in the configuration.
- **datatype**, indicates the type of the variable (int, text, boolean, ...).
- **data2store**, gives the information whether the variable is real-time only or the server must save it in a persistent database.
- **gw_type**, the monitoring system uses this field to subdivide every variable in 3 different types: *variable* as data information or *warning* and *alarm* as data to notify with 2 level of critical issue.

Thus, the configuration contains both, information regarding the structure of the machine and the information regarding the ”method” for data retrieving. The first is used to generate content from a GUI point of view while the second is used from a gateway point of view to retrieve data from the machines. Since several parameters in the

header configuration are shared between all the variables, the structure analyzed is a trade-off between level of customization of the system in the acquisition phase and simplicity to tune the system in phase of configuration.

Creating a new configuration, adding or removing variables of an existing configuration or completely deleting one from the web interface directly gives, in a few seconds, the instruction at the system to include or exclude variables. This process involves, but not only, the creation of new structures in the database to store the new variables added and the automatic update of the web interface with the new information. The creation of the structures in the database is a critical activity for the system because there is not a direct feedback for the user that can see only an abstract view of the structures, but they are essential to store data properly. That is why, the software parts which manage configurations, database and data exchange between gateway and acquisition module must be fully and widely tested before release the architecture.

Bibliography

[1] Jay Lee, Behrad Bagheri, and Hung-An Kao. A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing Letters*, 3:18–23, 2015.

[2] Qian Zhu, Ruicong Wang, Qi Chen, Yan Liu, and Weijun Qin. Iot gateway: Bridgingwireless sensor networks into internet of things. In *Embedded and Ubiquitous Computing (EUC), 2010 IEEE/IFIP 8th International Conference on*, pages 347–352. IEEE, 2010.

[3] Why companies must talk about edge computing. `https://www.01net.it/aziende-edge-computing/`.

[4] Felix Gessert. Lessons learned building a backend as a service: A technical deep dive. `https://medium.baqend.com/`, 2017.

[5] A Piccinini, V Pesenti Campagnoni, S Ierace, and F Floreani. A vibrational approach to slag sensing system: development and industrial application. *IFAC-PapersOnLine*, 49(12):1412–1417, 2016.

[6] V Pesenti Campagnoni, Stefano Ierace, Fabio Floreani, and Sergio Cavalieri. A pattern recognition methodology for fault detection:

A circuit breaker case study. In *Proceedings of the 10th World Congress on Engineering Asset Management (WCEAM 2015)*, pages 279–287. Springer, 2016.

[7] Beincpps. `http://www.beincpps.eu/`.

[8] Phil Bartle. The nature of monitoring and evaluation; definition and purpose. *Retrieved January*, 26:2015, 2007.

[9] Michael J de C Henshaw. Systems of systems, cyber-physical systems, the internet-of-things... whatever next? *Insight*, 19(3):51–54, 2016.

[10] Matthew NO Sadiku, Yonghui Wang, Suxia Cui, and Sarhan M Musa. Cyber-physical systems: A literature review. *European Scientific Journal, ESJ*, 13(36), 2017.

[11] R Keith Mobley. *An introduction to predictive maintenance*. Butterworth-Heinemann, 2002.

[12] O Myklebust. Zero defect manufacturing: a product and plant oriented lifecycle approach. *Procedia CIRP*, 12:246–251, 2013.

[13] Ezio Bartocci, Jyotirmoy Deshmukh, Alexandre Donzé, Georgios Fainekos, Oded Maler, Dejan Ničković, and Sriram Sankaranarayanan. Specification-based monitoring of cyber-physical systems: a survey on theory, tools and applications. In *Lectures on Runtime Verification*, pages 135–175. Springer, 2018.

[14] Georgios Fainekos, Oded Maler, Dejan Nickovic, and Sriram Sankaranarayanan. Specification-based monitoring of cyber-physical systems: A survey on theory, tools and applications. *Lectures on Runtime Verification: Introductory and Advanced Topics*, 10457:135, 2018.

[15] Jay Lee, Hung-An Kao, and Shanhu Yang. Service innovation and smart analytics for industry 4.0 and big data environment. *Procedia Cirp*, 16:3–8, 2014.

[16] Jacob Wurm, Yier Jin, Yang Liu, Shiyan Hu, Kenneth Heffner, Fahim Rahman, and Mark Tehranipoor. Introduction to cyber-physical system security: A cross-layer perspective. *IEEE Trans. Multi-Scale Comput. Syst*, 3(3):215–227, 2017.

[17] Xi Zheng, Christine Julien, Miryung Kim, and Sarfraz Khurshid. Perceptions on the state of the art in verification and validation in cyber-physical systems. *IEEE Systems Journal*, 11(4):2614–2627, 2017.

[18] Paulo Bartolomeu, Muhammad Alam, Joaquim Ferreira, and Jose Fonseca. Survey on low power real-time wireless mac protocols. *Journal of Network and Computer Applications*, 75:293–316, 2016.

[19] Frithjof Klasen, Volker Oestreich, and Michael Volz. *Industrial communication with fieldbus and ethernet*. VDE-Verlag, 2011.

[20] Lars Dürkop, Jahanzaib Imtiaz, Henning Trsek, Lukasz Wisniewski, and Jürgen Jasperneite. Using opc-ua for the autocon-

figuration of real-time ethernet systems. In *Industrial Informatics (INDIN), 2013 11th IEEE International Conference on*, pages 248–253. IEEE, 2013.

[21] GM Martinov, AB Lyubimov, AI Bondarenko, AE Sorokoumov, and IA Kovalev. An approach to building a multiprotocol cnc system. *Automation and remote control*, 76(1):172–178, 2015.

[22] Kang B Lee, Eugene Y Song, and Peter S Gu. Integration of mtconnect and standard-based sensor networks for manufacturing equipment monitoring. In *ASME 2012 International Manufacturing Science and Engineering Conference, MSEC*, pages 4–8, 2012.

[23] Linxia Liao, Raj Minhas, Arvind Rangarajan, Tolga Kurtoglu, and Johan De Kleer. A self-aware machine platform in manufacturing shop floor utilizing mtconnect data. In *Proc. Annual Conference Of The Prognostics And Health Management Society*, pages 1–9, 2014.

[24] Gianluca Cena, Ivan Cibrario Bertolotti, Stefano Scanzio, Adriano Valenzano, and Claudio Zunino. Evaluation of ethercat distributed clock performance. *IEEE Transactions on Industrial Informatics*, 8(1):20–29, 2012.

[25] Mohammad Aazam, Pham Phuoc Hung, and Eui-Nam Huh. Smart gateway based communication for cloud of things. In *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference on*, pages 1–6. IEEE, 2014.

[26] Pratikkumar Desai, Amit Sheth, and Pramod Anantharam. Semantic gateway as a service architecture for iot interoperability. In *Mobile Services (MS), 2015 IEEE International Conference on*, pages 313–319. IEEE, 2015.

[27] Matthias Kovatsch. Demo abstract: Human-coap interaction with copper. In *Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011 International Conference on*, pages 1–2. IEEE, 2011.

[28] Konglong Tang, Yong Wang, Hao Liu, Yanxiu Sheng, Xi Wang, and Zhiqiang Wei. Design and implementation of push notification system based on the mqtt protocol. In *International Conference on Information Science and Computer Applications (ISCA 2013)*, pages 116–119, 2013.

[29] Peter Saint-Andre. Extensible messaging and presence protocol (xmpp): Core. 2011.

[30] Ning Ye, Yan Zhu, Ru-chuan Wang, and Qiao-min Lin. An efficient authentication and access control scheme for perception layer of internet of things. 2014.

[31] Rafiullah Khan, Sarmad Ullah Khan, Rifaqat Zaheer, and Shahid Khan. Future internet: the internet of things architecture, possible applications and key challenges. In *Frontiers of Information Technology (FIT), 2012 10th International Conference on*, pages 257–260. IEEE, 2012.

[32] Armando W Colombo, Thomas Bangemann, Statmatis Karnouskos, Jerker Delsing, Petr Stluka, Robert Harrison, Francois Jammes, Jose L Lastra, et al. Industrial cloud-based cyber-physical systems. *The IMC-AESOP Approach*, 2014.

[33] Dimitris Mourtzis, Ekaterini Vlachou, Nikolaos Milas, and Nikitas Xanthopoulos. A cloud-based approach for maintenance of machine tools and equipment based on shop-floor monitoring. *Procedia CIRP*, 41:655–660, 2016.

[34] Bo Cheng, Jingyi Zhang, Gerhard P Hancke, Stamatis Karnouskos, and Armando Walter Colombo. Industrial cyberphysical systems: Realizing cloud-based big data infrastructures. *IEEE Industrial Electronics Magazine*, 12(1):25–35, 2018.

[35] Martin Wollschlaeger, Thilo Sauter, and Juergen Jasperneite. The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0. *IEEE Industrial Electronics Magazine*, 11(1):17–27, 2017.

[36] Mohammad Ghazivakili, Claudio Demartini, and Claudio Zunino. Industrial data-collector by enabling opc-ua standard for industry 4.0. In *2018 14th IEEE International Workshop on Factory Communication Systems (WFCS)*, pages 1–8. IEEE, 2018.

[37] Hassan Harb and Abdallah Makhoul. Energy-efficient sensor data collection approach for industrial process monitoring. *IEEE Transactions on Industrial Informatics*, 14(2):661–672, 2018.

[38] Zhigang Wen, Xiaoqing Liu, Yicheng Xu, and Junwei Zou. A restful framework for internet of things based on software defined network in modern manufacturing. *The International Journal of Advanced Manufacturing Technology*, 84(1-4):361–369, 2016.

[39] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5):637–646, 2016.

[40] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pages 13–16. ACM, 2012.

[41] ABM Moniruzzaman and Syed Akhter Hossain. Nosql database: New era of databases for big data analytics classification, characteristics and comparison. *arXiv preprint arXiv:1307.0191*, 2013.

[42] Seth Gilbert and Nancy Lynch. Perspectives on the cap theorem. *Computer*, 45(2):30–36, 2012.

[43] Jing Han, E Haihong, Guan Le, and Jian Du. Survey on nosql database. In *Pervasive computing and applications (ICPCA), 2011 6th international conference on*, pages 363–366. IEEE, 2011.

[44] Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Bill Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD*

workshop on Research issues in data mining and knowledge discovery, pages 2–11. ACM, 2003.

[45] Artem Chebotko, Andrey Kashlev, and Shiyong Lu. A big data modeling methodology for apache cassandra. In *Big Data (BigData Congress), 2015 IEEE International Congress on*, pages 238–245. IEEE, 2015.

[46] Datastax academy. Apache cassandra nosql performance benchmarks. `https://academy.datastax.com/planet-cassandra/nosql-performance-benchmarks`, 2017.

[47] Adrian Cockcroft and Denis Sheahan. Benchmarking cassandra scalability on aws over a milion writes per second. `goo.gl/pSLGbQ`, 2017.

[48] Jan L Harrington. *Relational database design and implementation*. Morgan Kaufmann, 2016.

[49] Jie Lian, Sheng Miao, Michael McGuire, and Ziying Tang. Sql or nosql? which is the best choice for storing big spatio-temporal climate data? In *International Conference on Conceptual Modeling*, pages 275–284. Springer, 2018.

[50] Vincent Reniers, Dimitri Van Landuyt, Ansar Rafique, and Wouter Joosen. On the state of nosql benchmarks. In *Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering Companion*, pages 107–112. ACM, 2017.

[51] Eric Brewer. Cap twelve years later: How the" rules" have changed. *Computer*, 45(2):23–29, 2012.

[52] Douglas Kunda and Hazael Phiri. A comparative study of nosql and relational database. *Zambia ICT Journal*, 1(1):1–4, 2017.

[53] Alexandre Verbitski, Anurag Gupta, Debanjan Saha, Murali Brahmadesam, Kamal Gupta, Raman Mittal, Sailesh Krishnamurthy, Sandor Maurice, Tengiz Kharatishvili, and Xiaofeng Bao. Amazon aurora: Design considerations for high throughput cloud-native relational databases. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pages 1041–1052. ACM, 2017.

[54] Esan Wit and Jan-Willem Selij. Galera performance and scaling analysis. 2014.

[55] Fernando Almeida, José D Santos, and José A Monteiro. E-commerce business models in the context of web3. 0 paradigm. *arXiv preprint arXiv:1401.6102*, 2014.

[56] Juan M Silva, Abu Saleh Md Mahfujur Rahman, and Abdulmotaleb El Saddik. Web 3.0: a vision for bridging the gap between real and virtual. In *Proceedings of the 1st ACM international workshop on Communicability design and evaluation in cultural and ecological multimedia system*, pages 9–14. ACM, 2008.

[57] Ruhi Sarikaya. The technology powering personal digital assistants. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[58] Maria Dominic, Sagayaraj Francis, and Anthony Pilomenraj. E-learning in web 3.0. *International Journal of Modern Education and Computer Science*, 6(2):8, 2014.

[59] Keshab Nath, Sourish Dhar, and Subhash Basishtha. Web 1.0 to web 3.0 evolution of the web and its various challenges. In *Optimization, Reliabilty, and Information Technology (ICROIT), 2014 International Conference on*, pages 86–89. IEEE, 2014.

[60] *Proc. Web 3.0: A Vision for Bridging the Gap between Real and Virtual Communicability (ACM 2008)*, 2008.

[61] George Hatzivasilis, Ioannis Askoxylakis, George Alexandris, Darko Anicic, Arne Bröring, Vivek Kulkarni, Konstantinos Fysarakis, and George Spanoudakis. The interoperability of things: Interoperable solutions as an enabler for iot and web 3.0. In *2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pages 1–7. IEEE, 2018.

[62] Gary Anthes. Html5 leads a web revolution. *Communications of the ACM*, 55(7):16–17, 2012.

[63] Scott Murray. *Interactive Data Visualization for the Web: An Introduction to Designing with.* "O'Reilly Media, Inc.", 2017.

[64] W3c web site. `https://www.w3.org/`.

[65] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation computer systems*, 25(6):599–616, 2009.

[66] Donald Ballou, Richard Wang, Harold Pazer, and Giri Kumar Tayi. Modeling information manufacturing systems to determine information product quality. *Management Science*, 44(4):462–484, 1998.

[67] Elias N Malamas, Euripides GM Petrakis, Michalis Zervakis, Laurent Petit, and Jean-Didier Legat. A survey on industrial vision systems, applications and tools. *Image and vision computing*, 21(2):171–188, 2003.

[68] Krishna Kumar Patel, A Kar, SN Jha, and MA Khan. Machine vision system: a tool for quality inspection of food and agricultural products. *Journal of food science and technology*, 49(2):123–141, 2012.

[69] Srinivas Katipamula and Michael R Brambley. Methods for fault detection, diagnostics, and prognostics for building systems-a review, part i. *Hvac&R Research*, 11(1):3–25, 2005.

[70] Jason Tranter. The fundamentals of, and the application of computers to, condition monitoring and predictive maintenance. In *COMADEM 89 International*, pages 372–377. Springer, 1989.

[71] Andrew KS Jardine, Daming Lin, and Dragan Banjevic. A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical systems and signal processing*, 20(7):1483–1510, 2006.

[72] Ranganath Kothamasu, Samuel H Huang, and William H VerDuin. System health monitoring and prognostics-a review of current paradigms and practices. *The International Journal of Advanced Manufacturing Technology*, 28(9-10):1012–1024, 2006.

[73] Rolf Isermann and Peter Balle. Trends in the application of model-based fault detection and diagnosis of technical processes. *Control engineering practice*, 5(5):709–719, 1997.

[74] David Mautner Himmelblau. *Fault detection and diagnosis in chemical and petrochemical processes*, volume 8. Elsevier Science Ltd, 1978.

[75] Louis-François Pau. Failure diagnosis and performance monitoring. 1975.

[76] David Applegate and William Cook. A computational study of the job-shop scheduling problem. *ORSA Journal on computing*, 3(2):149–156, 1991.

[77] José Fernando Gonçalves, Jorge José de Magalhães Mendes, and Mauricio GC Resende. A hybrid genetic algorithm for the job shop scheduling problem. *European journal of operational research*, 167(1):77–95, 2005.

[78] Lifeng Zhang and Brian G Thomas. State of the art in evaluation and control of steel cleanliness. *ISIJ international*, 43(3):271–291, 2003.

[79] Da-peng Tan and Li-bin Zhang. A wp-based nonlinear vibration sensing method for invisible liquid steel slag detection. *Sensors and Actuators B: Chemical*, 202:1257–1269, 2014.

[80] DaPeng Tan, ShiMing Ji, PeiYu Li, and XiaoHong Pan. Development of vibration style ladle slag detection methods and the key technologies. *Science China Technological Sciences*, 53(9):2378–2387, 2010.

[81] M Keskilammi, L Sydänheimo, and M Kivikoski. Radio frequency technology for automated manufacturing and logistics control. part 1: Passive rfid systems and the effects of antenna parameters on operational distance. *The International Journal of Advanced Manufacturing Technology*, 21(10-11):769–774, 2003.

[82] Vedat Coskun, Busra Ozdenizci, and Kerem Ok. A survey on near field communication (nfc) technology. *Wireless personal communications*, 71(3):2259–2294, 2013.

[83] Li Zhekun, Rajit Gadh, and BS Prabhu. Applications of rfid technology and smart parts in manufacturing. In *ASME 2004 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 123–129. American Society of Mechanical Engineers, 2004.

[84] A Scarpellini, Luca Fasanotti, A Piccinini, Stefano Ierace, and Fabio Floreani. A web-based monitoring application for textile machinery industry. In *Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI), 2016 IEEE 2nd International Forum on*, pages 1–6. IEEE, 2016.

[85] Xiang Li. Operations management of logistics and supply chain: Issues and directions. *Discrete Dynamics in Nature and Society*, 2014, 2014.