Alma Mater Studiorum - Università di Bologna

DOTTORATO DI RICERCA IN

INGEGNERIA ELETTRONICA, TELECOMUNICAZIONI E
TECNOLOGIE DELL'INFORMAZIONE

Ciclo 34

Settore Concorsuale: 09/E3 - ELETTRONICA

Settore Scientifico Disciplinare: ING-INF/01 - ELETTRONICA

CLOUD TECHNOLOGIES AND DATA-DRIVEN ALGORITHMS FOR INTERFEROMETRIC SENSORS.

Presentata da: Gianmarco Giorgi

Coordinatore Dottorato

Aldo Romani

Supervisore

Riccardo Rovatti

Esame finale anno 2022

Abstract

The PhD project developed in these years started by studying the motivations behind Industry 4.0 and the most popular data-driven algorithms. The study was then oriented on the main success factors for the realization of a connected product continuing then for a study of the cloud in more detail.

This study led me to analyze different components offered by the main providers, and to implement different solutions. The different solutions both at architectural and provider level allowed us to verify the differences between different implementations and their associated costs. The study of the cloud was then concluded with an exhaustive cost analysis that clearly highlights what are the gains or losses associated with the choice of a provider as a function of traffic exchanged.

The study is then articulated on the more purely signal processing part, it is first presented the principle of operation of the interferometric sensor and then analyzes the main issues related to the signals produced by the instrument. It then analyzes the main strategies for solving the problems exposed, and the proposed solution also trying to explain how we arrived at that solution through the analysis of the main criticality of the signal.

It concludes by analyzing the problems of implementation of the algorithmic part within the real sensor with limited processing capacity and the strategies undertaken to mitigate the impact of these issues on the final implementation. The analysis of the implementation is accompanied by some data about the timing with which it is possible to use the algorithm and its main limitations.

Table of contents

1 Introduction to Industry 4.0

1.1 Motivations behind the rapid evolution toward connected factories

Industry 4.0 bills itself as the fourth industrial revolution, setting out to make industries smart. As in the second industrial revolution, the assembly line allowed to compress costs for the production of otherwise inaccessible goods, the combination of IoT and AI seek to not only compress production costs, but also set the goal of developing a system capable of monitoring company production flows for continuous quality improvement. Companies that make use of these technologies are called "smart factories."

To make these change possible, companies have to face a strong internal renewal, first of all is to reconcile the needs of OT and IT in a single solution that allows a rapid migration to an IoT system. Another challenge companies face is leadership and corporate priorities. These two factors are considered the main factors for success in the transition to Industry 4.0 according to [2].

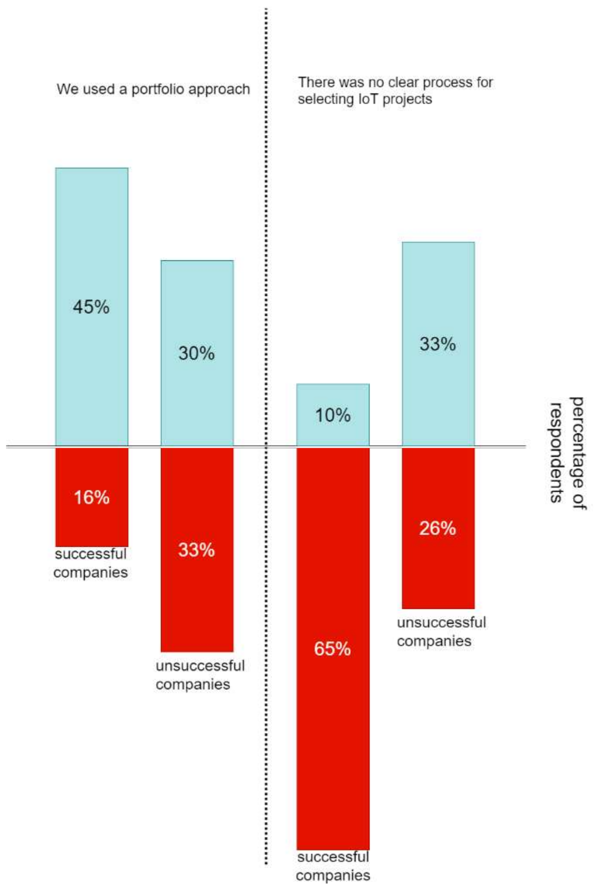

Figure 1: IoT project selection [2]

1.2 Business strategies for IoT

1.2.1 The key factors for a successful IoT ecosystem

At the heart of the fourth industrial revolution are smart and interconnected sensors, these sensors are able to communicate with each other and the outside world by improving their behavior over time. The interconnected objects used in this new industrial paradigm makes it possible to keep track of many parameters that currently were not. This makes it possible to observe the trend of machine parameters both vertically throughout the process and transversely within the production plant. Constant monitoring makes it easier to identify which point is more prone to errors in the production process, as well as the possible deterioration and/or breakage of production tools, identifying and preventing possible downtime and high economic losses.

Predictive maintenance was certainly one of the themes that drove the first peak [2] of IoT diffusion in an I4.0 perspective, but it is certainly still very topical for many companies, especially SMEs, that have not ridden the wave so suddenly.

Figure 2: Drivers for success [2]

This sensor paradigm in addition to bringing new perspectives brings into the field some issues such as those related to information security, data confidentiality and data ownership.

1.3The role of data in Industry 4.0

1.3.1 Confidentiality and sensitivity associated with data

In this new scenario the role of data plays a fundamental role, in fact in a connected system, where data are not confined, but can cross even several continents, the design of the infrastructure, to ensure confidentiality and security, must be done carefully. In recent years we have begun to change the paradigm with which the algorithms are developed, in fact, especially in industry, the algorithms were designed based on some assumptions about the signal and trying to verify if the assumptions

described the signal well enough. Lately, people are trying to develop algorithms where data itself describes the signal and characterizes his property; this is the case of AI.

Taking into consideration this second case, we can see how the lack of protection of a data cannot only provide a potential competitor with information on the parameterization of the production process. This may also give data that can give rise to an important commercial advantage; in fact, the exploitation of such data can give rise to more successful commercial solutions.

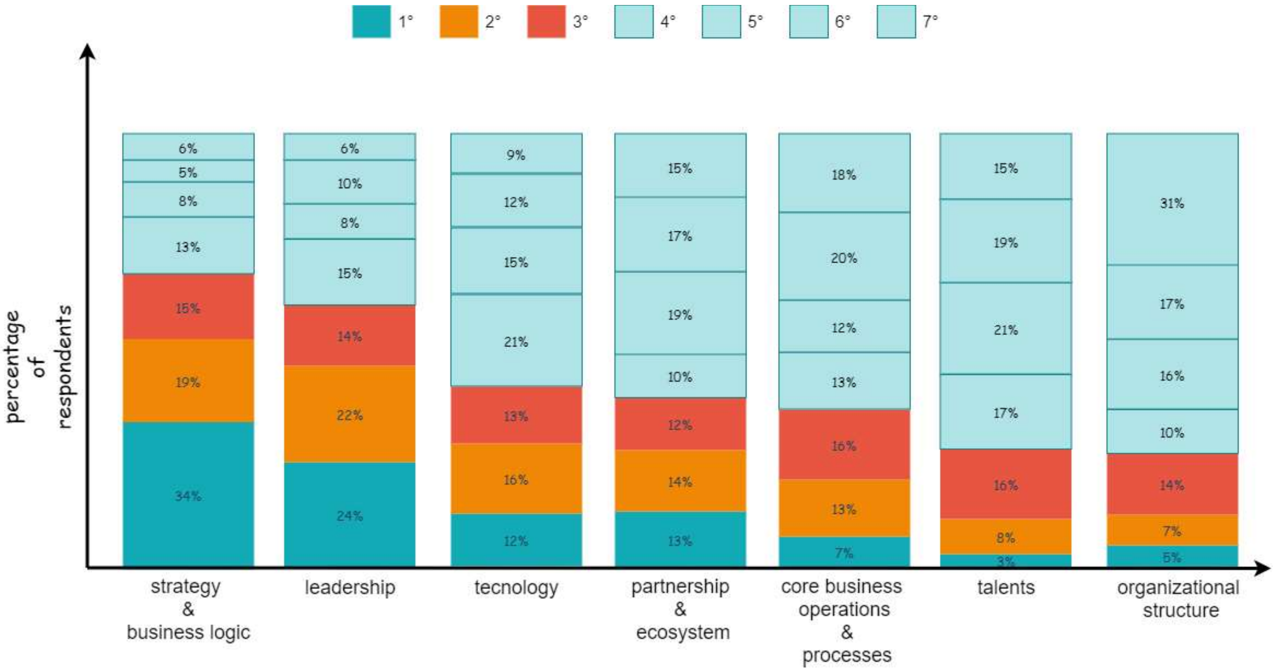

Figure 3: Impact of success factors on the success of an IoT project [2]

We speak of confidentiality of a data when the data cannot be disclosed outside a certain restricted circle of users; we speak instead of sensitive data when the data cannot be disclosed at all.

Data management in this scenario where data storage may be in one state, processing in another certainly offers great flexibility, but at the same time raises new challenges to be faced.

1.3.2 Presentation of data

In this section, we present the data that will then be used in subsequent sections of the thesis as the basis for evaluating the performance of the algorithms under analysis. All signals were obtained from a non-contact interferometric optical meter, whose working principle will be described later.

Fig.4 present a signal from a more complex geometry, this signal was obtained by scanning an air wedge. You can see three curves, the first is the curve of the plastic support that is constant throughout the signal, and the second is that of the wedge. At the beginning, there is a first part where the thickness is less than the plastic support, and then exceed it, while the last is the sum of the two previous curves. In

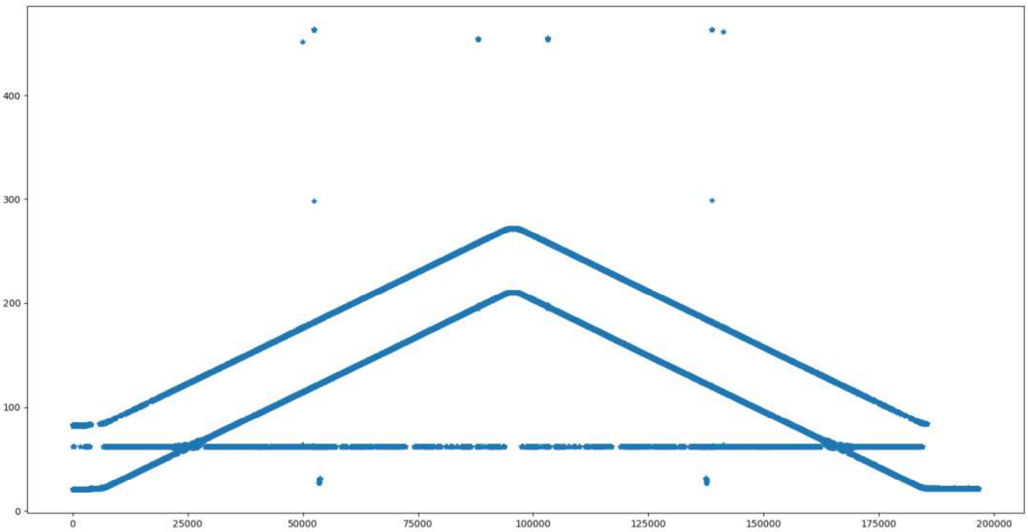

Figure 4: Air wedge signal

this case, the separation is more complex because the qualities are not available and there is the intersection of the curves. The intersection, which is the most interesting part of the signal, is certainly the most critical, even if an algorithm was able to separate the 3 surfaces before the intersection happens, it is not sure that it will be able to do it after, moreover the signal near the node is very noisy and discontinuous.

To correctly analyze the signal in this case means:

- Separate the surfaces that present the intersection
- Correct post-intersection tracking
- Tolerate interruptions
- Reduce noise

In **Fig.5** we present a more complex case that has been analyzed within the PhD, the signal has several complexities. The geometry is certainly complex; the signal is very noisy, so as not to present itself as a continuous signal but as the union of several signals close together. Other complexities that we can identify involve discontinuities and convergence of surfaces. In this case, the separation is very complex, in quality the two signals are difficult to separate, in thickness it is not possible to separate them given by their geometry, and the micro discontinuities present can induce error in the algorithm.

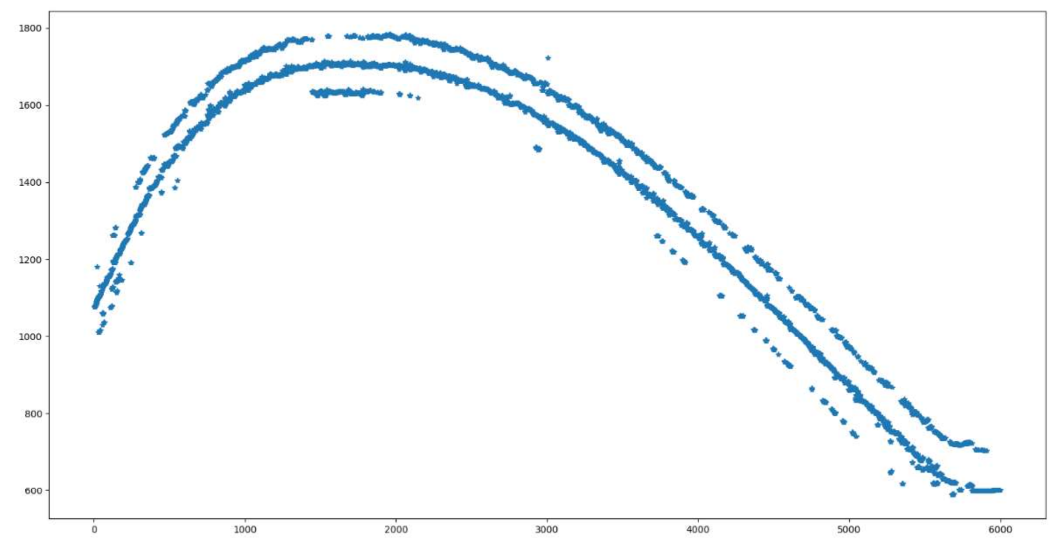

Figure 5: Double-layer material

Correctly analyze this signal implies:

- Correctly aggregate the micro surfaces
- Identify correctly the real surfaces and eliminate artifacts
- Correctly tracking the surfaces
- Tolerate interruptions
- Identify convergences of multiple surfaces and assign them to a single surface.
- Reduce noise

A brief illustration has been given of the data that will be explored in the next sections. The peculiarities that have been expressed can be attributed in part to the applications in which this sensor operates, such as geometries, but in other cases, such as noise and discontinuities, attributable to the type of technology used which is noisier than its contact counterparts are.

2 The Cloud

The cloud has the function of bringing together in one place, even if virtual, a series of functions that are not only important for IoT, but also for IT systems in general. Remaining within the IoT context, the cloud allows, in the most essential case, to collect and sort the messages produced by the devices. This first step allows to read, from anywhere in the world, the data of the connected IoT devices, or at least the data transmitted from the devices to the cloud. The major players in the cloud field, in addition to the simple message broker, offer with PaaS infrastructure a system of registration and authentication to the broker, an edge process manager, in order to remotely control which and how many services to start on the edge side. In addition to these services in the PaaS of the major cloud developers, offer a rule manager, a DB system, analytics systems, digital twins, RTOS operating systems, etc.

2.1 Technologies and declinations of the Cloud

To achieve a portfolio of services such as those mentioned, there may be different strategies and technologies within the same cloud. First, choose the type of technology to be used: IaaS, PaaS or SaaS. In addition to the technology to embrace, it is important to evaluate the main cloud architectures available: on premise, cloud, hybrid cloud or multi-cloud. The sectors in which the architecture is important are certainly those that make use of sensitive data such as the military, in the hospital sector, in sectors that must comply with specific laws, etc.

Within the IoT context, there is also the choice between a cloud-based IoT system, where the only concentrator is in the cloud, or whether to opt for a distributed architecture that makes use of distributed concentrators called edges.

2.1.1 Iaas, Paas e SaaS

To better understand how the three main cloud technologies influences the development of an application, let us take an example of IoT development: our goal is to develop a system for monitoring sea level along the Italian coasts. To create a monitoring infrastructure we need to recreate some of the services we mentioned at the beginning of the chapter starting from virtual machines, being in an IaaS environment. The first thing to note is that having only virtual machines available the management and updating of the infrastructure, such as applying security patches is our responsibility, this means that to ensure a certain SLA we need to carefully plan these operations. Secondly, we need to identify the applications that best match our needs, for example to create our message broker we will dedicate a virtual machine to an MQTT broker. This VM according to the data output rate will need different performances, to cope with the load variation it is possible to configure the VM for a dynamic vertical scalability, this scalability allows us to save money when the number of incoming messages is low and to cope with sudden load variations. Now if the variations are contained, the vertical scalability could satisfy our exigencies, but in

case of anomalous events, like a tidal wave, this flexibility could not be enough and a horizontal scalability could be necessary. In this case, it is necessary this exigency must be managed, it is seen very soon, making analogous reasoning to those just described, that the IaaS technology offers a great flexibility, it allows to customize very in detail the solution, but it requires an expert team for its management. This solution is therefore not suitable for companies that do not have a culture in this sense and that for the first time face the Cloud world.

Let us now analyze the constitution of the same service using PaaS technology, In this scenario the management costs are borne by the service provider, however we still

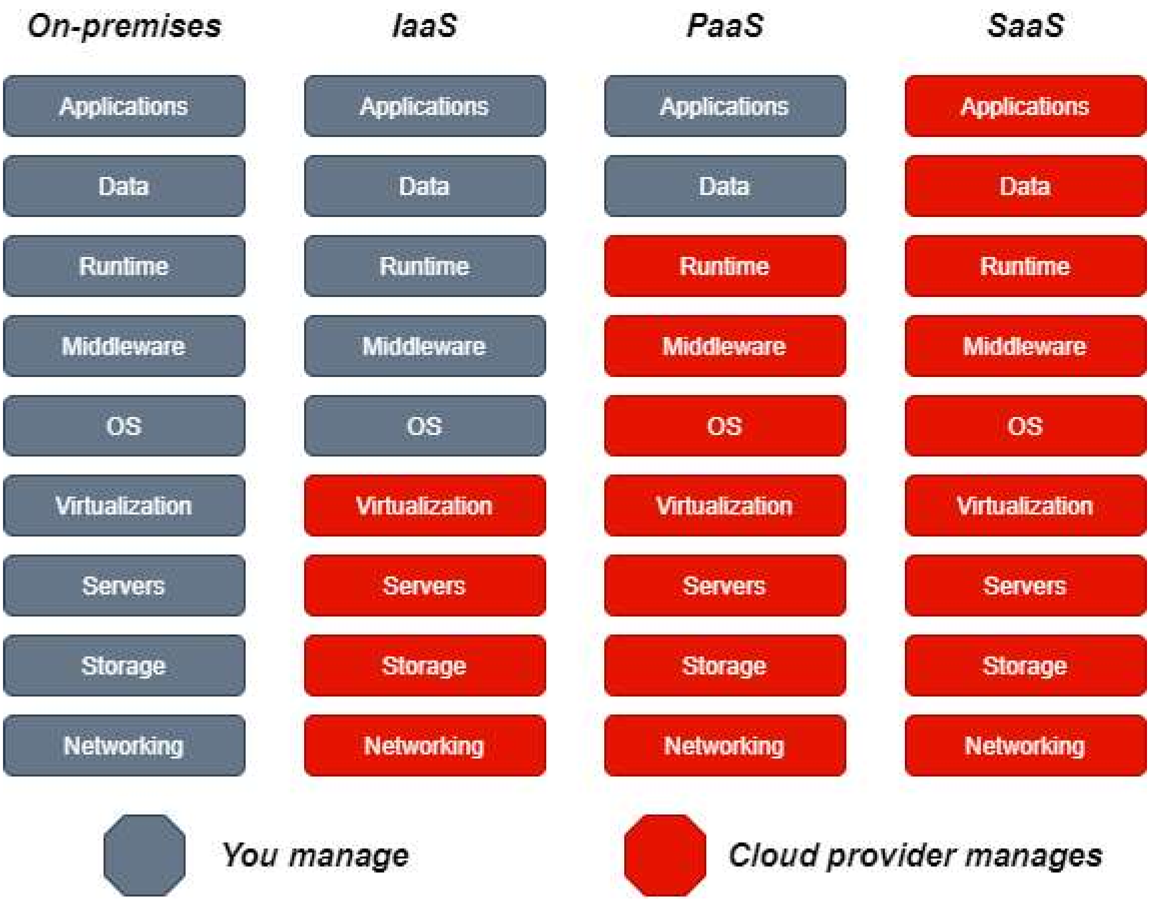

Figure 6: Key cloud-related tecnologies and management levels

have some things if not our responsibility, such as data management, choosing the best method of data storage, etc. However, these complexities are less burdensome than the complete management due to an IaaS system. Service composition is limited to the composition of individual services provided by the platform provider.

Ultimately, let us analyze SaaS technology, this technology already has all the bricks necessary to constitute the IoT system, what remains to be done by those who buy the system is the configuration of the service. As it is obvious to observe this type of approach is also suitable for those who have little experience in the field and exposes the developer to less inexperience. The SaaS platform is certainly the most expensive of those analyzed, but also the one with the most appeal, given the simplification it provides. For this reason, in my opinion, this technology is the one with the greatest economic return. Again, in my humble opinion, the motivation behind the explosion of SaaS compared to other technologies is due to the need to approach the cloud in a simpler way in order to solve contingent problems; this first experience has probably made mature a certain awareness about the advantages and costs offered by the cloud. This experience probably combined with the Covid-19 experience, which acted as a catalyst, has somehow pushed companies to adopt strategies to decrease costs.

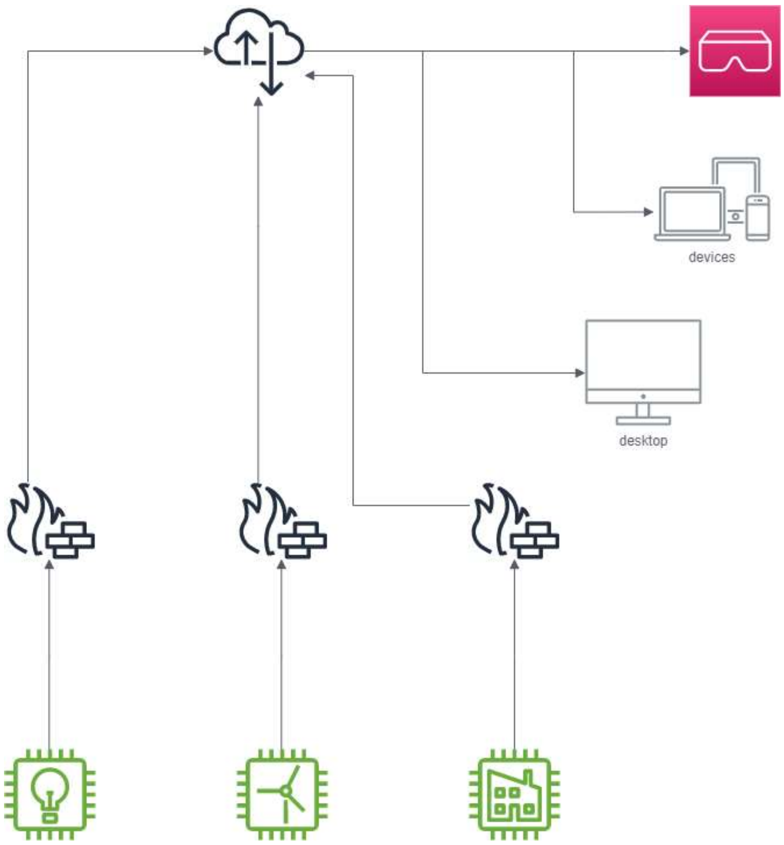

Figure 7: Cloud computing schema

2.1.2 Coud computing vs edge computing

Another choice you face when designing an IoT infrastructure is whether to choose between a centralized Cloud-based systems or on a distributed edge-based system, both solutions have their merits and drawbacks. A centralized system without intermediaries has less cost, on the other hand, some areas have to deal with the problem of latency, a centralized system is certainly less expensive, but the cloud may

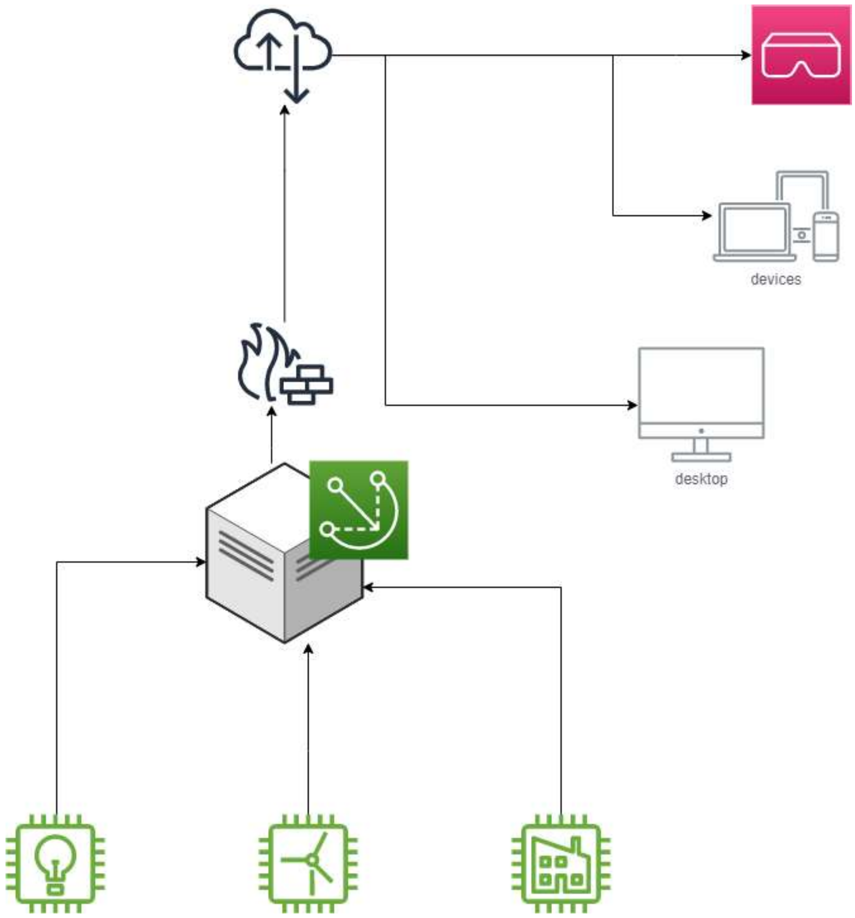

Figure 8: Edge computing schema

reside on a different continent from where the data is collected. In this case, latencies of exchanged messages can reach the order of seconds or tens of seconds. These timeframes are often unacceptable if there is a requirement for feedback from the cloud infrastructure. In addition, many times having an intermediate concentrator allows you to send already aggregated data to the cloud or to perform a first filter on which data is important and which is not. In addition, having one or more edges, as a low-cost perimeter-computing unit can be an advantage for adding intelligence to sensors or for closing feedback controls that require more stringent timelines than those outlined above.

In some cases, an edge infrastructure has data privacy benefits, as it allows sensitive data to be processed directly at the edge or, if the computation is too costly, sensitive data can be obfuscated at the edge. Another advantage of the distributed solution is that it can work even if the network partitioning is tolerant; this means that the overall infrastructure is more resilient and can guarantee a higher SLA.

2.2 From OPC UA to Cloud

The OPC UA protocol, the first version of which was released in 2008, is a platform-independent service-oriented architecture that encapsulates all the specifications of classic OPC in a single extensible framework [3]. This protocol is now a de facto industry standard and is constantly evolving to embrace very different needs. This flexibility has allowed it to become one of the main methods of communication in the OT environment, and to evolve to the point of being able to connect cloud and OT in a single solution.

Our toy-case has been built trying to use industrial standards as much as possible as communication protocols, this approach, in our opinion, has allowed us to develop a highly enlargeable platform. This advantage allows us to integrate more sensors and to have a standard way to communicate with other sensors and/or actuators.

This solution implemented on the single sensor allows us to easily extend the platform to all sensors produced by Marposs S.p.A that support the OPC UA protocol.

2.2.1 OPC UA

The multilayer architecture allows obtaining a continuously evolving protocol, tailored on the specific application at the same time. Its structure is divided into 7 layers, starting from the lowest we have the layer that the configuration in the protocol, then we have the communication layer, this layer supports both the client-server and pub-sub connection paradigm. Going up, we find the two layers of information model, one more basic, which includes the structure of notifications, events and basic methods (browse, configure, execute, etc.) while the upper one defines the address space.

The last three layers define the information model of the specific use case. The first of these three identifies that basic layer common to all that defines the specifications

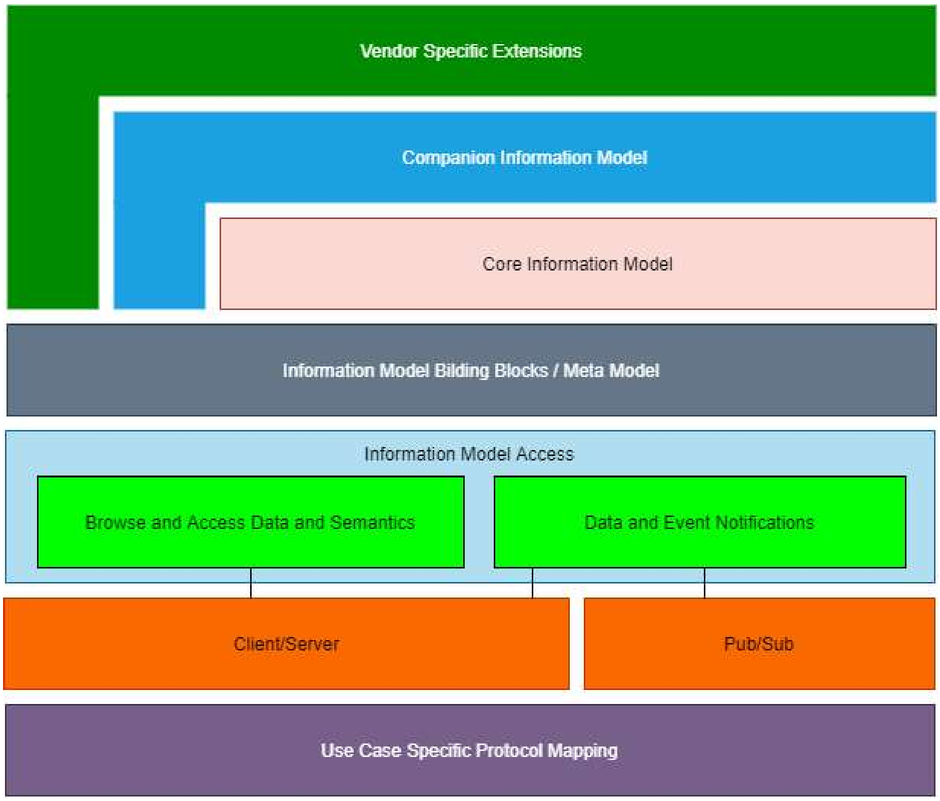

Figure 9: OPC UA protocol stack

for those basic functionalities that are more or less necessary for any application (state machine, file transfer, etc.).

The last two instead are the two parts of greater customization and represent, in my opinion, the real flexibility of the architecture. The second layer represents the standard layer divided by market sectors, for example to define the standard for CNC, a committee meets and establishes which data are to be exposed to protocol and which are not. This process can take many years to reach convergence, but it allows for a unified model that is not brand dependent. Since this standard cuts across multiple industries, it makes it a good candidate as a bridge protocol between the OT world and the cloud.

The last layer is the one that allows customizing the model by the sensor manufacturer, this kind of feature tries to meet the needs of individual products that may need to expose specific data or methods, such as to support brownfield applications.

2.2.2 Da OPC UA a MQTT

The transition between OPC UA and MQTT is a transition that can take place in two ways. The first is following the recent specifications introduced by the OPC foundation about the use of MQTT as a pub-sub protocol, which allows you to use the tools already established previously and be cloud compatible. The second one take in to account the developing a protocol gateway that translates the OPC UA protocol in a protocol based on MQTT, the latter for example is the way followed by Azure and AWS in their cloud have implemented gateway solutions as connectors between the OPC UA world and the cloud one.

2.2.3 Alternatives: Eclipse SparkPlug

Some possible alternatives to OPC UA was taken in to account. The one presented by Eclipse Foundation, StarkPlug, seems to be a good alternative. It has a smaller code footprint and uses less data compared to OPC UA, but assumes that the entire infrastructure is already MQTT ready, so it is a solution suitable for new development applications (Greenfield). This reason combined with the lack of market adoption made us discard the possibility of conducting more in-depth tests on this protocol.

2.3 IIoT Proof of Concept Development

The development of the PoC for the connection of Marposs S.p.A. sensors to the cloud has foreseen several steps aimed at verifying specific architectural solutions and related cost models. In fact, different implementations can give rise to very different costs, as it is equally true that the same architecture can give rise to very different costs if implemented on different providers. These reasons have pushed us to verify which solutions are the most suitable for our needs, comparing custom solutions with "off the shelf" solutions provided by providers. The cost analysis, we want to remember, is not secondary in a cloud solution, and is one of the design flows related to the platform. Finding cost-effective solutions that are easy to account for, in many cases, is of greater interest than the most technically performing solution.

At a strategic level, analyzing the portability of a solution to different clouds is also important, both at a cost level and at a purely technical level.

The reasons just listed led us to develop two separate solutions for each provider analyzed, for four solutions. We are aware that the providers analyzed do not allow us an exhaustive analysis, in fact for completeness it would have been necessary to analyze also Alibaba Cloud, whose roots in China and in the markets influenced by it makes it the ideal partner for those countries. However, the solutions have allowed us to make most of the considerations listed above.

2.3.1 AWS-based solution

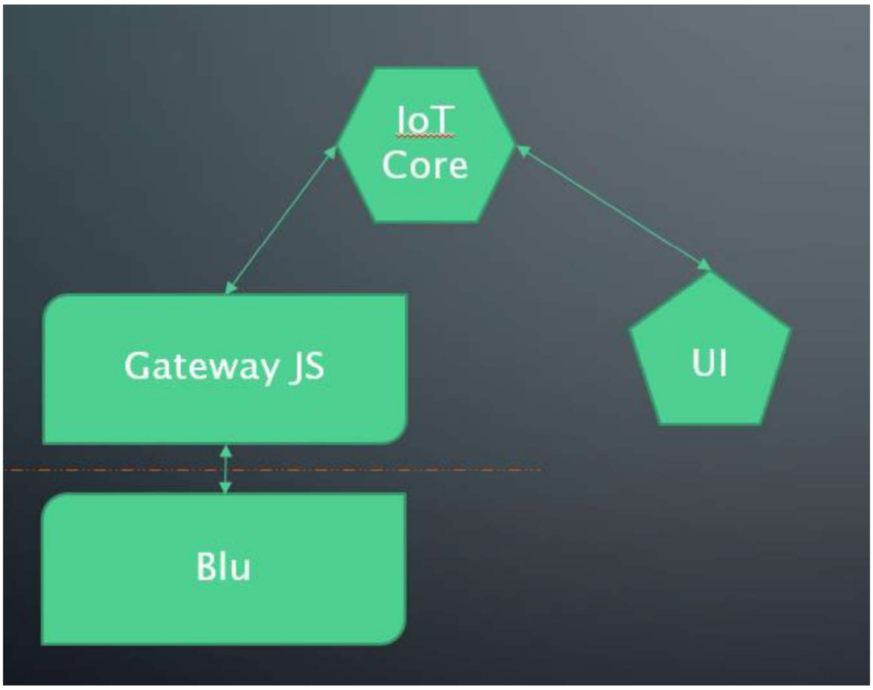

Figure 10: Our edge computing architecture for AWS

The first of the two AWS-based solutions consists of AWS IoT Core service, OPC UA gateway provided by AWS, OPC UA server hosted on the BLÚ sensor and the User Interface. The goal is to remotely configure the sensor, for this reason a DB was not sandwiched between the broker and the UI, for data storage. This very lightweight architecture allows to verify the feasibility of remotely configuring a sensor through the cloud and to verify the latency of the infrastructure. The OPC UA server residing on the sensor is part of the product and was already present, the gateway provided by AWS was in a fairly basic state of development and needed to be extended to have acceptable functionality and the GUI was developed ad hoc, with proprietary MVC, for this specific application.

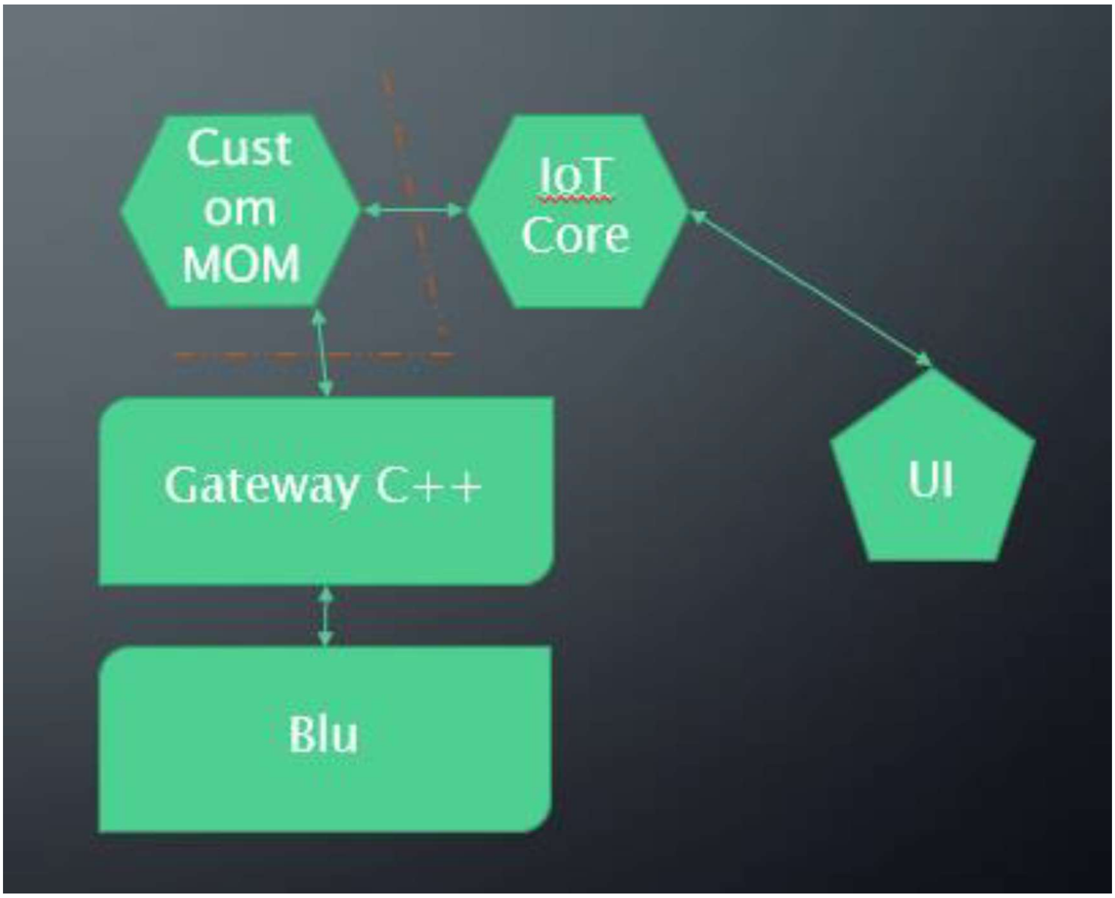

Figure 11: Our cloud computing architecture on AWS

The red dotted line indicates the logical division between cloud and device. This demarcation is intended only at the logical level and not physical, because the physical division occurs between the gateway and the service. This logical division, different from the physical one, is due to the AWS IoT Greengrass cloud module, this module allows to realize the so-called fog computing. The fog-computing paradigm allows you to see the device or one of its parts as residing within the cloud, it is then possible to provision processes directly on the device. This type of paradigm is very interesting for the enrichment of performance or for the remote management of components that make up our edge.

The second solution, changes the architecture by inserting an additional element between the gateway and the cloud service. The protocols supported by the AWS service are MQTT and HTTPS, but these two protocols are not able to encompass all industrial scenarios, for example the instruments connected with 5G technologies may not have a TCP/IP stack, making it impossible to connect them directly to the cloud. In these scenarios, it is necessary to place an intermediate module between the device and the cloud, this intermediate module acts as a gateway between the cloud protocols and the specific protocols with which the device communicates. In this case, there are also situations in which you are forced to use certain programming languages, but these languages do not have libraries mature enough to support all cloud providers.

To address these scenarios, it was necessary to add an intermediary to our cloud infrastructure. The choice of the intermediary can be different depending on the specific need, for example in the case of sensors connected to the cloud via an NBIoT (5G) network the intermediary can be realized through an MQTT gateway, translating UDP packets into TCP. In the case, constrained by the language, the solution may be to rewrite part of the library, but this way is not always possible because of SW licenses that could constrain library itself or introducing an intermediary more flexible than the cloud service and then connect the cloud with the intermediary. This last solution is the one that has been adopted in our case. The intermediary we have chosen has allowed us to solve both the described problems, since it is a more flexible MQTT broker than the one do AWS and supports bridging with the AWS broker, at the same time however it has a module to support CoAP requests, one of the most used protocols in NBIoT environment.

Once solved the communication problem, it remains to decide where to place this intermediate entity. It is possible to instantiate it either on the device together with the gateway, in this way it would remain on the sensor (obviously this is possible only in case the device has the adequate computing power) or on the cloud, for example by subscribing an EC2 instance. The two solutions represent the two logical separations just mentioned, which in this case coincide with the physical separations. In the end, both solutions have their merits and demerits, and it is necessary to evaluate which solution is more suitable for one's needs.

2.3.2 Microsoft Azure-based solution

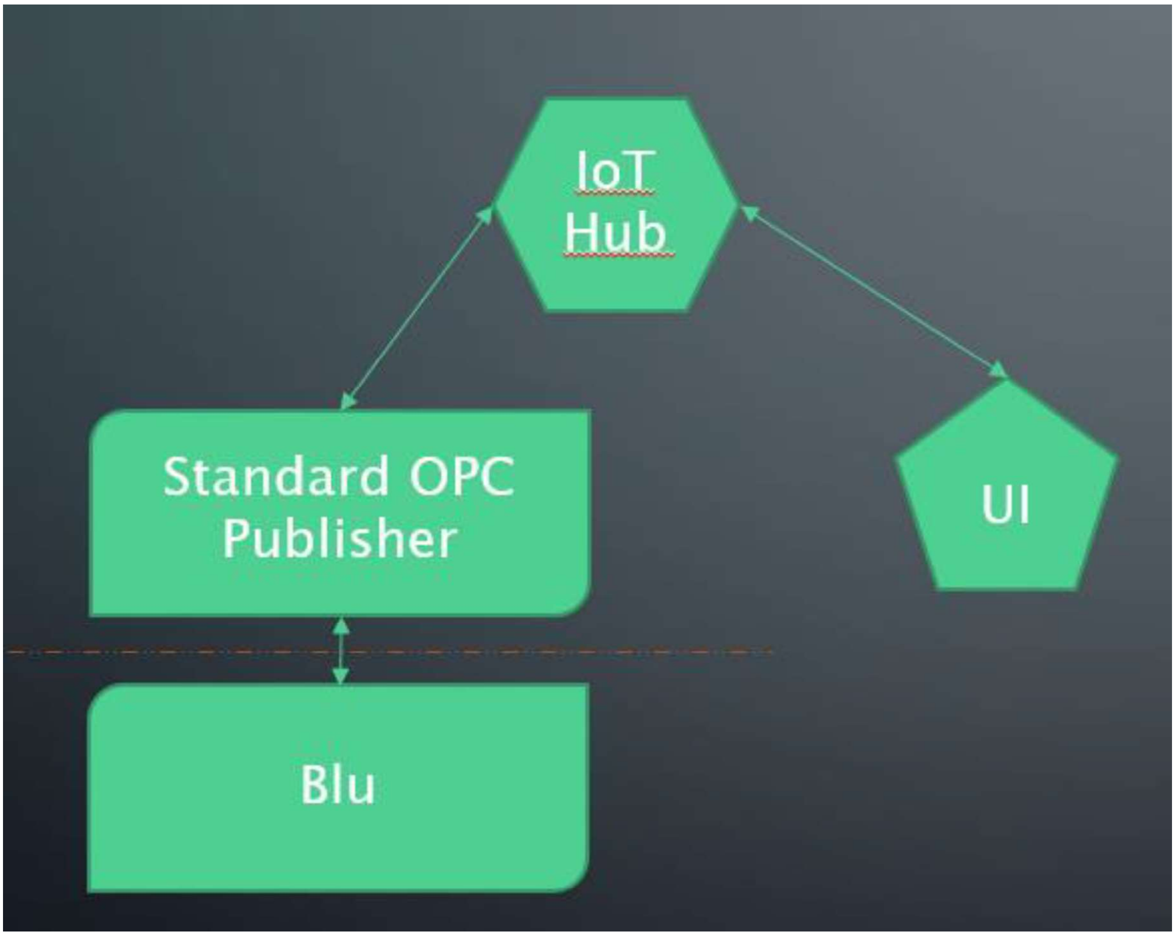

Figure 12: Our edge computing architecture for AWS

The Azure-based solution, as in the AWS case, is divided into two solutions: the cloud-based solution that makes fog computing possible, and the custom solution. Both solutions are based on the management of appropriate containers deputies to perform the gateway activity, the custom solution provide an appropriate configuration of the container to perform the tunneling of communication through port 443, generally accessible in the enterprise environment, while the standard solution provides the unmodified container and the edge management software. In the latter architecture, the container that abstracts the gateway dialogues directly with the management module that in turn dialogues with the cloud, but this module is more difficult to customize and does not provide for communication through the cloud thanks to the HTTPS protocol port. For this reason, this solution is certainly more flexible in some ways, but it requires an agreement with the customer's IT in order to establish the network architecture necessary to minimize risks, but at the same time to allow a proper functioning of the IIoT system.

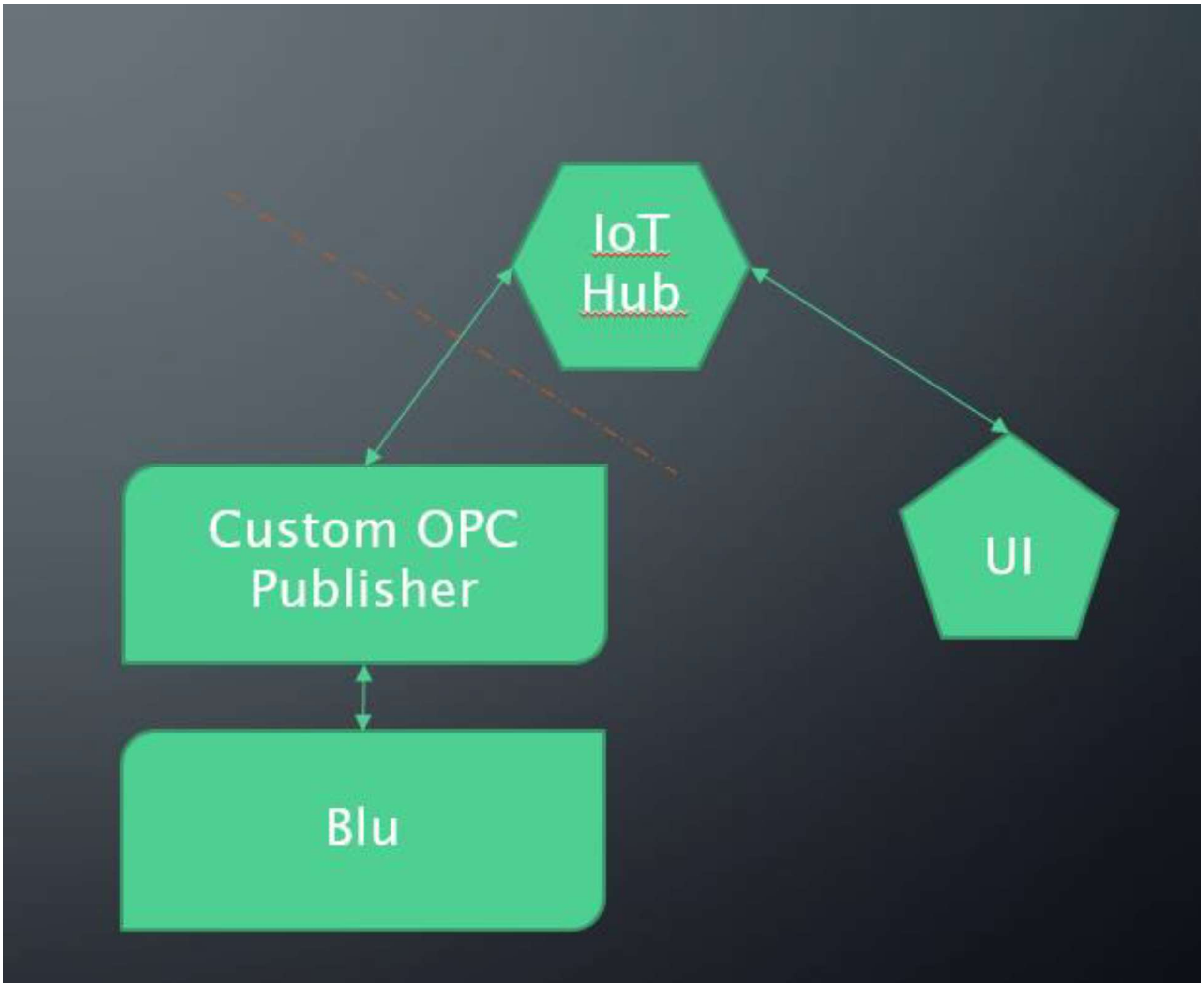

Figure 13: Our cloud coputing architecture on Azure

As with AWS-based solutions, the two proposed architectures aim to solve two different problems: the first creates an integrated cloud-side analytics platform, while the second enables data exchange in the most congenial conditions in an enterprise environment.

2.3.3 User Interface

The GUI gives another important component that goes to compose our IoT infrastructure, this component allows to visualize the data coming from the sensors that publish data in the cloud and to manage a minimal configuration. The goal of the GUI in addition to visualizing the data collected through the cloud tries to verify the actual latency between the device that sends the data A and a device that receives it B. In order to minimize synchronization errors between the clocks of two different devices the same device, in our case, the edge, sent the data to the cloud applied a timestamp and re-read the data through the GUI calculating the difference through the same clock.

The GUI structure is low on a custom MVC framework and developed internally in a non-PhD related activity. The latency test allowed us to understand that a real-time configuration of the meter is possible, because the overall roundtrip latency is around 70 ms (up to 150 ms), in both providers. In order to understand if these values are acceptable and to understand what the limit values are, we took as reference the limit latencies associated with interactive data application. In this way, we have found that the acceplable range extends up to 150 ms (in interactive data application scenario values below 100 ms may still affect the quality), while values higher than 400 ms are unacceptable [4]. The values reported are intended as "mouth-to-ear" values, therefore one-way. These values allow us to say that the values recorded by our connection are within the acceptable range.

2.4 Cost assessment

In cloud solutions, the cost is certainly one of the qualifying figures of merit, for this reason we now want to evaluate the differences in cost between the two solutions that have, as constraint ,the fact that they have been developed through the components provided by the provider.

As a first element of distinction between the Azure solution and the AWS solution, we have the method of cost calculation: Azure has a monthly subscription type approach with a certain number of available resources, while AWS has a pay-as-you-go (PAYG) approach. This difference leads to a first complexity in determining the most cost effective solution; the second complexity comes from the fact that different solutions have different costs on the two providers. To be clearer if I take solution **A** that has a lower cost than a solution **B** on AWS it is not necessarily the case that on Azure solution **A** is cheaper than solution **B**.

In this chapter, given the complexity just exposed, we do not want to provide a solution, or rather, we will not provide the best provider for IoT solutions, since this strongly depends on the use case, but we will try to provide a general method of reasoning valid to choose the right partner.

In order to make a correct cost estimate it is necessary to be clear:

- The number of sensors
- Average connection time
- The number of messages and their size
- Message direction (device to cloud and/or vice versa)
- Edge/fog computing structure

The estimation based on Azure is very simple; its plan provides an unlimited number of devices that you can connect to the cloud in continuous connection. This plan in turn is divided into two tiers S and B, the first provides two-way messaging and fog computing, while tier B does not. In turn, the tiers are divided into 3 categories based on the number of daily messages available; the first receives up to 400k messages, the second up to 6 million messages and the last up to 300 million messages. To each class and each tier correspond different prices, for the three B classes we have respectively 12.5$, 62.5$ and 625$ per month, while for the S we have 31.25$, 321.5$ and 3125$.

Let us now turn to the cost estimate based on the AWS infrastructure, in this case the pricing is fully PAYG and thus highly dependent on consumption. To get a meaningful estimate, on the understanding that the number of daily messages cannot be greater than 400k messages/day we assumed to have an infrastructure consisting of 100 connected sensors. The daily messages available are equivalent to sending 4.6 messages per second; distributing these messages among the hundred sensors, we have that each sensor can send 2.76 messages per minute. Trying to frame this scenario, we speak of a very slow monitoring, and example related to the possibility of monitoring the status of the sensors. For the estimation of the cost elements are, the connection time, the number of messages and the number of edge with cloud functionality.

In our hypothesis:

- we have for the connection fee only USD 0.42 obtained as 0.096 USD/Mmin * 100 * 30 days * 24 h/day * 60 min/h
- For the messaging part, instead, we have USD 14.4 obtained as 1.2 USD/Mmsg * 400kmsg/day * 30 days ,
- If we imagine to have an edge every 10 sensors, the cost for fog computing is USD 1.8 calculated as 10 * 0.18 USD/device

As a result, we have USD 16.62.

From this analysis we see that for low amounts of data the AWS platform generally offers a lower cost than Azure tier S, on the contrary if we compare it with tier B we see that for low amounts of data the AWS platform is cheaper until it reaches the cross point at 286Kmsgs and thus becomes more expensive.

Another aspect to take into account is the movement of large amounts of information such as reading a log, in these cases, the number of messages exchanged is dependent on the size of the single message and the individual message sizes therefore affect the overall cost of the transfer. In this scenario, AWS has a message size (for pricing) that is 25% larger, resulting in 20% fewer messages being transferred from the device to the cloud. If we take as an example the transfer of a log file of 1.6 GB (equivalent to 400k messages * 4KB of message). We can see how in the Azure case the threshold corresponding to 12.5 USD is reached, while AWS uses a lower number of messages, equal to 320k, to transfer the same message reducing the cost from 16.75 to 13.75 USD.

2.5 Solution comparison

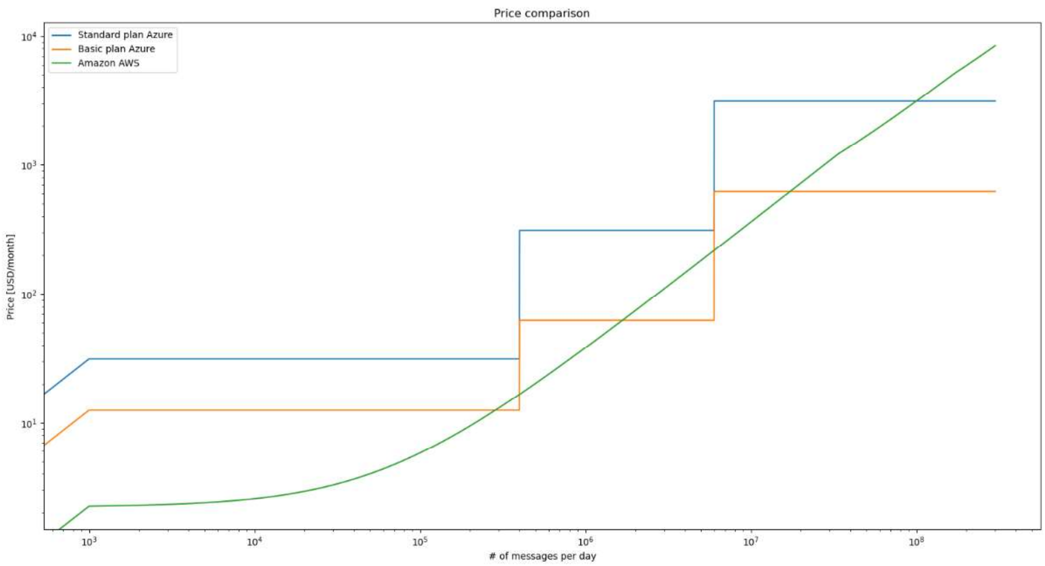

Figure 114: AWS vs Azure cost comparison

The two solutions both have their field of application. The intrinsic differences that differentiate the two solutions, both economically and technically, make each solution suitable for a specific use case. For example the difference in cost of the two platforms, as can be seen in **Fig.14**, which is in logarithmic scale, as you can see there are 3 cross points for the Basic, one for each class, while only one for the Standard.

These points trivially indicate the economic equivalence of the two solutions, but the difference between the two solutions can be appreciated outside of these points, in these regions the differences between one solution and another may involve differences of a few thousands of dollars per month. For example if we look at **Fig.15**, it is clear that Azure is more economical when I need to exchange many messages, while AWS is more advantageous in the regions of subscription switching and especially in the last one.

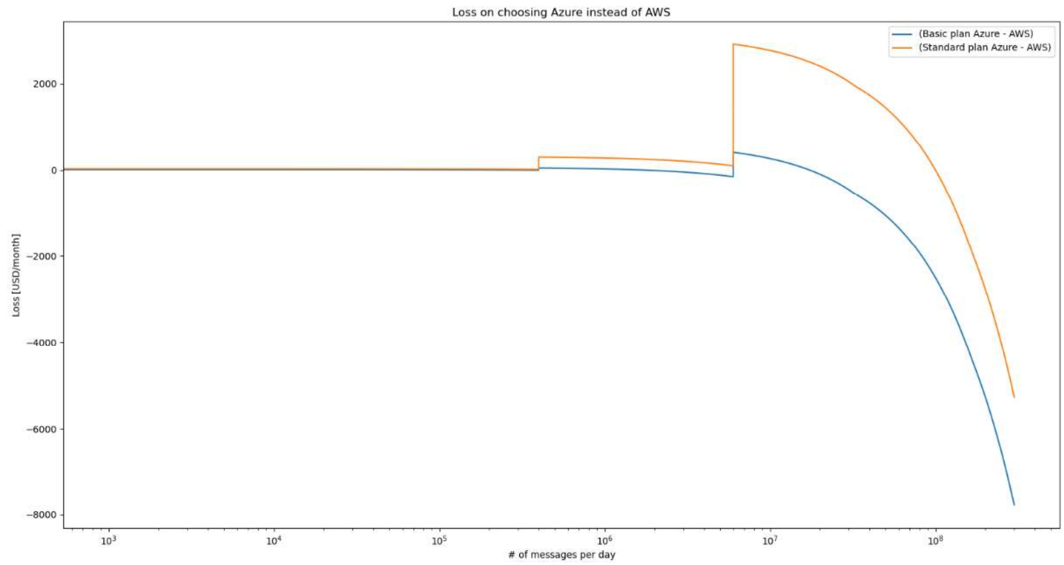

Figure 15: Azure choice over AWS: loss function

From a technical point of view, the differences can be summarized in the following table:

Technologies	Azure	AWS
Fog computing	FaaS	FaaS + CaaS
Broker Cloud	Partial MQTT support + AMQP + HTTPS	HTTPS + almost full MQTT support
Network enterprise support	Leaved to container implementation. No support for fog computing	With MQTT tunneling on 443 port
Message size	4KB	5KB
Cloud to device messages	Method invocation	MQTT subscription

Table 1: Technical differences between AWS and Azure

Let us now try to analyze the individual items in order to better understand these differences in order to understand how they influences the choices that the architect, designing a new interconnected system, must make.

Fog computing is that paradigm of cloud computing that allows extending cloud capabilities to edge devices, specifically it allows to manage edge resources as if they were logically resident on the cloud. This type of paradigm allows resources to be dynamically managed and device fleets to be managed directly through the cloud. The FaaS technology solution enables this management through function execution, while the CaaS one through container execution.

The broker represents the heart of the IoT infrastructure, this service manages the sorting of messages between producers and consumers, verifies the identity of devices and manages the protocols through which you can interact with it. As you can see from the table, the differences between the two providers are small and are limited to Azure's support of the AMQP protocol and its limited support of MQTT topics. These differences define how sensors or edges can interact with the cloud and what requirements they must have in order to connect.

Among the differences that the two providers highlight there is the message billing size. It is necessary to consider this when the message size grows because in these cases at the same data exchanged between the sensor and the cloud there could be a difference in the number of messages exchanged thus affecting the final price of the solution.

The bidirectional communication is not always a required aspect in the implementation of an IoT infrastructure, but it is certainly an aspect to be taken into account in case you are designing this type of infrastructure. In the two cases, this issue is addressed in a different way, in the case of AWS the two-way is achieved through MQTT subscriptions, but it is up to the user to define and manage the topics. As far as Azure is concerned, the communication to the device is done through the invocation of methods; this invocation consists in sending a message with specific characteristics to the Hub that forwards it to the device using specific topics to which the device subscribes. In this way, the problem of topic structure is solved upstream, but leaves less flexibility.

As a last argument, we want to address the possibility of implementing the solution in a business context, where firewall and network structure in general is not under our control. Either in these scenarios, it is possible to make an agreement between the parties in order to define the rules that allow the implementation of the solution or it is possible to develop solutions that allow not having to change the network access rules. This is possible by defining outgoing connections only and defining a port known to be available for outgoing connections: port 443. By tunneling traffic through this interface, you can solve many of the issues related to this topic. AWS natively supports message tunneling through this port, while Azure only partially, in fact in unplanned solutions an agreement must be made.

Data-driven algorithms are those algorithms that try to "get data-driven"; by this statement, we mean that these algorithms are developed from data. Typical examples of such algorithms are clustering algorithms, neural networks, SVMs, and unsupervised learning. Unlike the normal processes that lead to the definition of an algorithm through assumed or a priori knowledge, these approaches try to extract from a large amount of data the relevant information to perform best in that task. This approach has been very successful in recent years, managing to outperform in many domains, from cybersecurity to medical diagnostics over traditional techniques. The main problem that this method encounters is in having a large amount of data available; in fact, these methods are typically used a lot in the head of BigData, cloud processing with IoT support for collecting data.

However, in many other cases, this type of approach can also be used on the smart sensor node; depending on the type of algorithm you want to implement the sensor node will need to have more or less processing power at its disposal. You can find many examples of application of ML algorithms or the application of neural networks to microcontrollers among the demonstrators of many leading companies, this kind of trend shows a growing interest in posting this kind of algorithms closer and closer to the data source. The motivation, in my opinion, is due to the need to enrich, clean

Figure16: NCG, Marposs S.p.A. Product range

or contextualize the signal already in the early stages of processing. This type of approach in fact allows not propagating, to the subsequent stages of processing, unnecessary data that could make more complex or more expensive downstream processes.

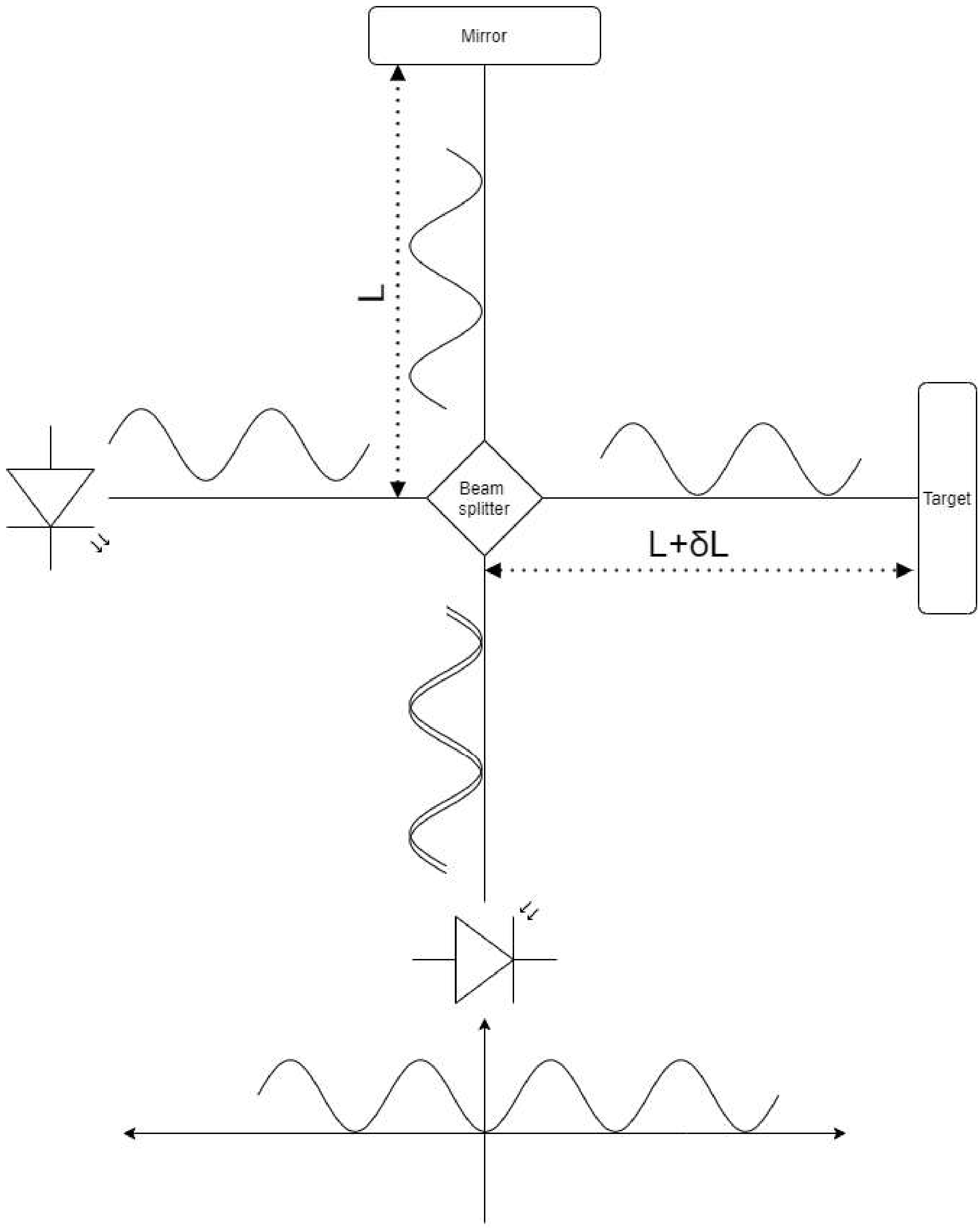

Figure 17: Michelson interferometer, working principle

3.1 NCG (Non-contact gauge) Sensor

NCG is a sensor to conduct non-contact measurement of semiconductors or transparent multi-layer materials. The main reasons for adopting non-contact systems compared to the more classic and economical contact transducers is due mainly to two factors. The first one is due to material damage, if the material cannot be ruined and the contact would ruin. The second one is due to inaccessibility, if the material or the quantity to be measured is not externally accessible. In the case of a multi-layer transparent infrared, the only tests possible with a contact transducer are destructive tests.

These devices are interferometers, the band of wavelengths framed depends on the material to be inspected, and they work by returning an interference signal.

3.1.1 Working principle

The principle of operation of the Marposs S.p.A. interferometer is very similar to that of Michelson, in which a beam of coherent light is passed through a beam splitter (a semi-transparent mirror) that divides the beam in two; the first half is propagated on a fixed mirror, while the second is reflected on a moving mirror. The light reflected from the two mirrors is then fed into a photodetector that detects the optical path difference between the two branches.

Our interferometer extends this operation on the light source, in the Michelson interferometer, the source was mono-frequency, in our case the source emits in a known band. This variation manifests an interference on the wavelengths related to the thickness of the material analyzed; specifically the law that describes this trend is as follows:

$$I = I_1 + I_2 + 2\sqrt{I_1 I_2}\cos\left(4\pi t\frac{n(\lambda)}{\lambda}\right)$$

3.1.2 Applications and Usage

NCG gauges are very precise, but certainly more expensive than their contact counterparts, moreover these gauges can be used only if the material under examination is transparent in the band used. This tool, however, offers the possibility to solve some problems of considerable importance such as:

- Measuring multilayer objects in which the intermediate layers are not accessible
- Measuring materials where contact is not allowed because it could ruin the surface texture.

In these two categories certainly interferometric technology, but not only, is a better alternative to contact measurement.

3.2 Data produced by NCG Sensor: Measurement and related metrics

Measurement data from the interferometric type sensor are generally of three classes: measurement, quality, and “peakiness”. The first is obvious, the second indicates the reliability of the measurement data, and the third indicates the accuracy of the data. A low quality number is generally associated to a noisier and more imprecise signal, therefore less reliable. As far as peakness is concerned a high peakiness number means that the data is very reliable from the metrological point of view, while a low value obviously indicates the opposite, it should be noted that this does not mean that the measured value is wrong, but that the data is probably associated with a greater relative error.

These three data characterize each measurement value, i.e. if we have a multilayer material composed of 5 layers, we would have 5 measurement values each of which would have: the thickness value, the quality associated to it and its peakiness. These 18 values characterize the material from the sensor's point of view for each measurement value. The measurement of a material is generally carried out not on a single point, but on a series of points by scanning the piece.

3.3 Interferometer signal issues

The data produced by interferometric sensors can suffer from a number of problems, due firstly to the characteristic of being non-contact, secondly to the application issues arising from the scans.

The first problem with non-contact sensors in a multilayer material occurs when the probe is referenced and the gauge becomes a distance meter, in this condition to adequately reconstruct the profile of the material following a scan it is essential that all the measurement values are present. This necessity is due to the fact that it is necessary to attribute the correct refractive index to each material they pass through, otherwise the output measurement values do not correctly represent the physical reality.

However, this condition is not always true, to overcome this problem it is possible to apply a hold policy, that is, in case of lack of a measure, the last valid measure is proposed again at the output. It is clear that if I have a multilayer material with a high number of layers, but I am interested only in the first n, in case of lack of the first, this is replaced by the next and the hold policy does not work anyway.

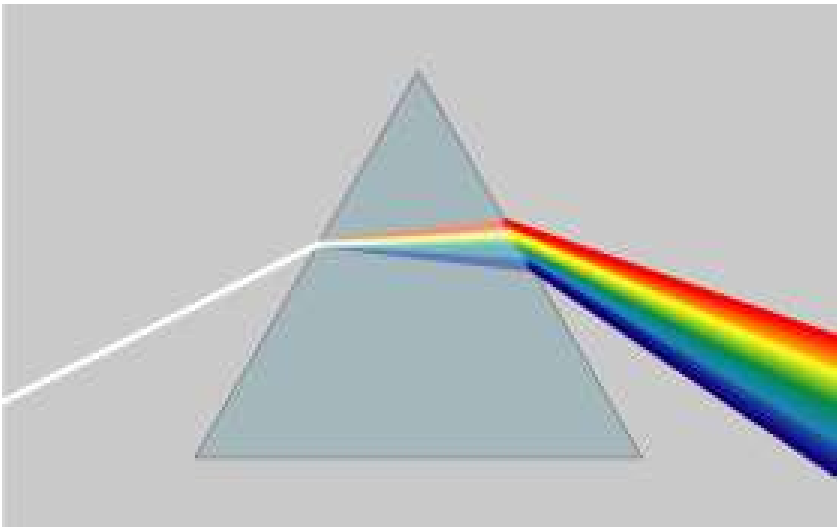

Figure 12: An exmple of chromatic dispersion, taken from Wikipedia [6]

The problems just expressed, require special attention in the analysis because they can invalidate the measures returned by the sensor. In this field there are not many solutions, since the subject is very complex, the main techniques that seek to remedy this problem address the problem a posteriori, analyzing the entire scan. These techniques, although effective, do not allow a real-time analysis of the piece, making them unsuitable for many applications.

Another problem that arises from the analysis of the material of interest is the dispersion of some materials, such as glass and plastics. The dispersion of refractive index is the characteristic of some materials to present different refractive indexes at different wavelengths, this characteristic that is generally exploited to create spectrometers and that creates chromatic aberration here is manifested by the difficulty of being able to accurately determine the measure, specifically this property acts on the peakiness making it very low.

In order to solve this problem it is necessary to compensate this effect directly on the interference signal, provided we know the characteristics of the material under examination, in many cases this is not true, in these cases it is necessary to approximate or estimate the dispersion of the material under examination in order to measure it.

The general approach is to approximate through appropriate tables the trend of dispersion in order to make a correct compensation.

3.4 Solving approaches: State of the Art

After presenting, the main problems related to interferometric measurement we now want to present some solutions that currently represent the state of the art. We will divide this sub-chapter into two parts, in the first part will be presented the solution to the problem of optical dispersion, in the second part, however, will be analyzed the solution to signal artifacts created during scanning.

The solution to the problem of dispersion can be solved by making an appropriate compensation; this compensation can be derived from the knowledge of the dispersion law of the material. In many however, this law is not known or in others, you only have some tabulated values. In the second case the solution is quite simple and consists in finding an appropriate law that interpolates the tabulated values, in the first case is available a database containing the laws for the materials of interest or in estimating empirically the law. The empirical estimation of the law, however, requires a high number of tests and generally is based on the knowledge of the refractive index and the Abbe number defined as:

$$V_d = \frac{n_d - 1}{n_F - n_C}$$

The Abbe number is used to classify the dispersivity of a glass according to the figure, in fact this number allows to have an estimate of the dispersivity of the material.

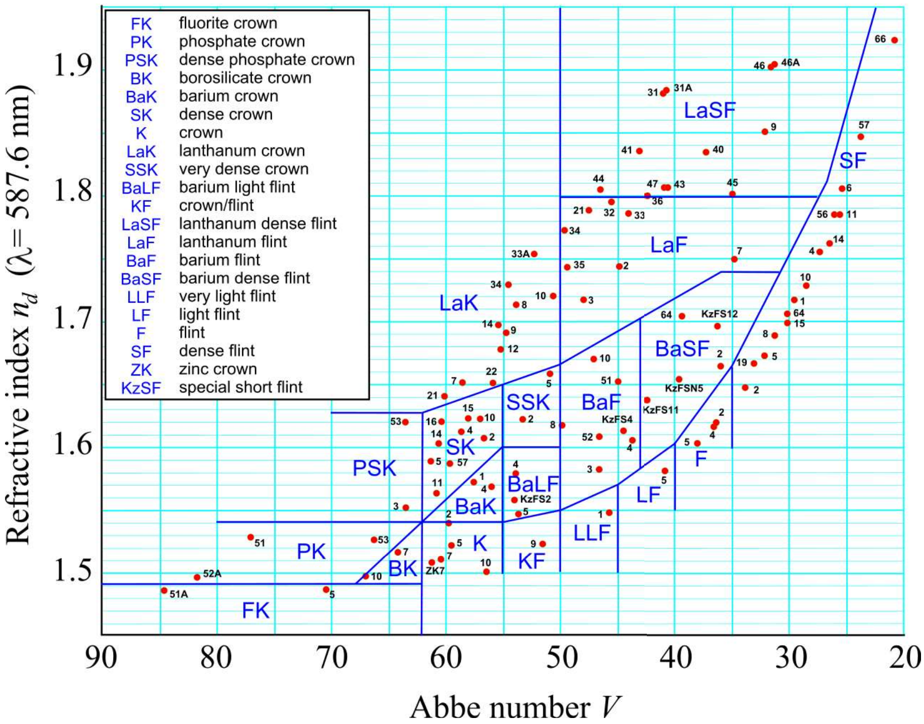

Figure 19: Glass classification based on Abbe number, taken from Wikipedia [6]

The main problem of this method comes from its lack of generality, in fact this method, assuming ad hoc tests, needs to redo the empirical tests of curve estimation for each material of interest.

The solution to the problem of artifacts instead is done with the knowledge of the entire scan a posteriori, that is, the entire signal of interest. This type of approach allows you to have an overview of the trend of n surfaces of interest. In this way is drawn a first reference taking into account the predominant cluster, is made a parabolic regression on this cluster and identifies the point of maximum or minimum. Around this point is applied DBSCAN, you take the cluster with more points and consider it as a reference, on this cluster is made a parabolic regression and seeks the point of minimum or maximum. The output value of the maximum or minimum point is returned.

All the algorithms described are limited to the extraction of a single measure, and were designed to solve a specific problem and consequently inflexible, in many cases with few variations they can be extended to the case of multiple measures, but the output stability parameters remain tailored to the use case on which they were developed.

The main shortcomings that the algorithm has are due to the tracking of a curve only, only at the point of maximum or minimum (absolute? Local points how much do they affect?).

3.5 Suggested Solution

We will now analyze the proposed solutions in order to solve or at least mitigate the problems seen on the solutions considered now state of the art.

3.5.1 Color dispersion

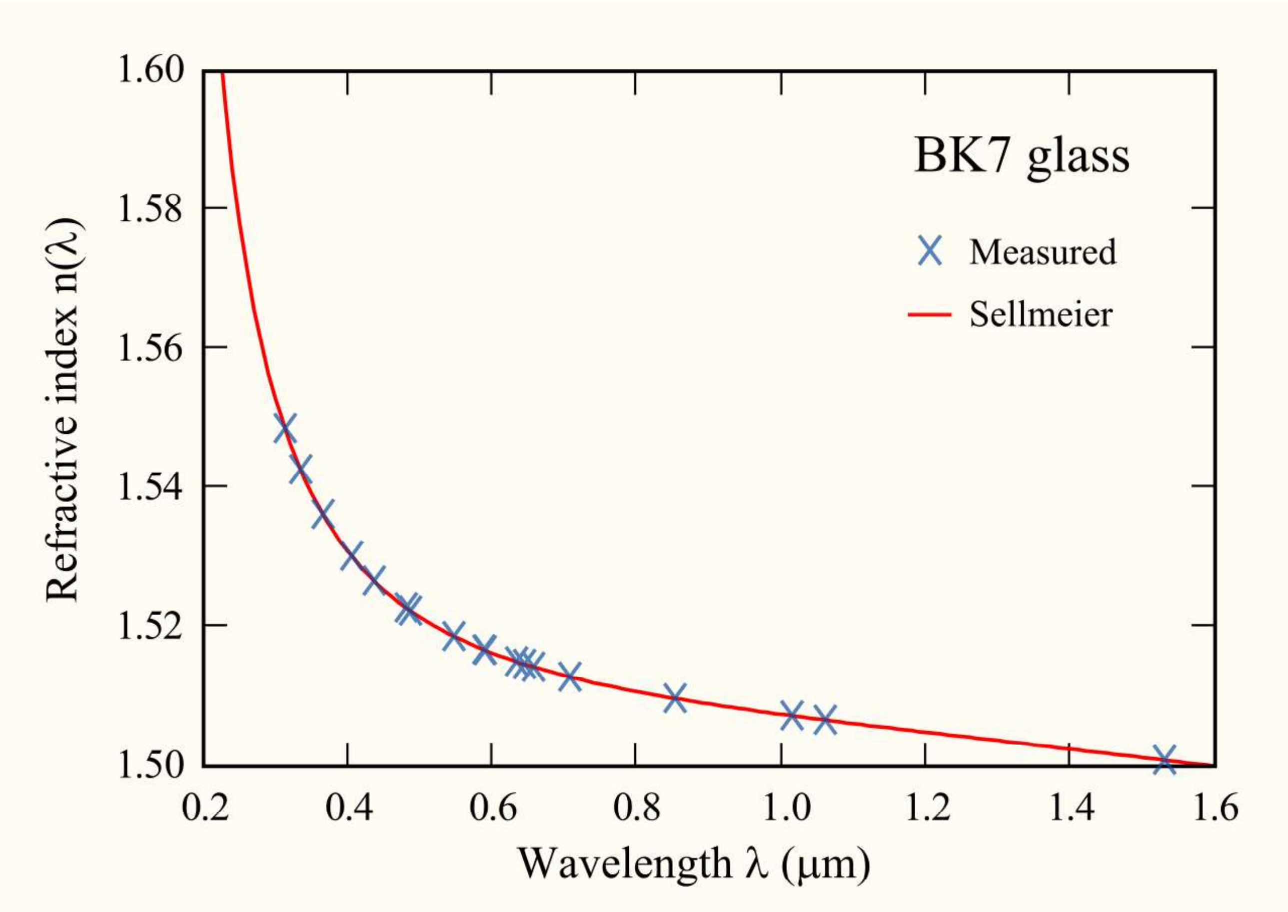

Figure 13: Color dispersion on λ of BK7 glass, taken from Wikipedia [8]

The problem of chromatic dispersion significantly deteriorates the spectrum of the FTT signal from which it extracts the measurement information, many times the deterioration makes it impossible to obtain a measurement value from the signal, and this problem is mainly attributable to the loss of periodicity of the signal. The solution we propose tries to combine simplicity of configuration and effectiveness in the compensation of the phenomenon, so we try to adopt a strategy with few parameters, the number of Abbe and the refractive index, but that offers a good approximation of the dispersion curve.

In order to achieve this goal we have relied on the laws of approximation of dispersion in the literature, among these the ones we have chosen are the laws of Sellmeier and Cauchy, these laws work in different situations and for this reason we have considered both. Let us introduce first Sellmeier's law and then extend the conclusion to the simplest solution. As you can see from the figure Sellmeier's law approximates very well the measured values, at the same time you can see in the figure the two laws differ only outside the visible band.

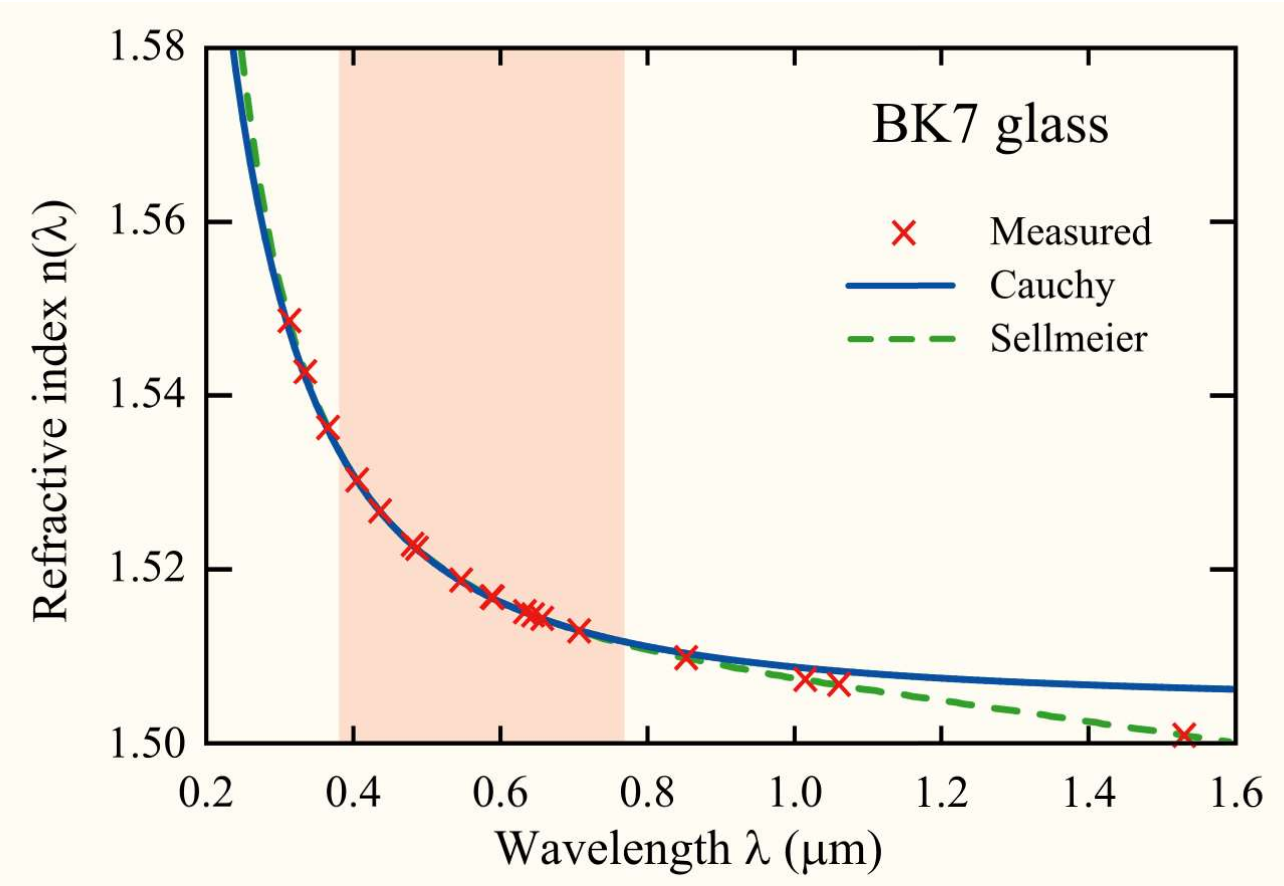

Figure 21: Approximation comparison between Sellmeier's law and Cauchy ones, taken from Wikipedia [8]

Sellmeier's law approximates the refractive index trend as follows:

$$n^2(\lambda) = 1 + \sum_i \frac{B_i \lambda^2}{\lambda^2 - C_i}$$

In general, the value and the number of coefficients is determined experimentally, but in our case, having available only two coefficients the dispersion law is reduced to only two coefficients returning the form:

$$n^2(\lambda) = 1 + \frac{B_0 \lambda^2}{\lambda^2 - C_0}$$

This simplification returns a sufficient approximation to compensate for our use cases as well as allowing us to calculate the coefficients B and C using the following system of transcendent equations:

$$\begin{cases} \sqrt{1+\dfrac{B_0{\lambda_d}^2}{{\lambda_d}^2-C_0}} = n_d \\ \sqrt{1+\dfrac{B_0{\lambda_f}^2}{{\lambda_f}^2-C_0}} - \sqrt{1+\dfrac{B_0{\lambda_c}^2}{{\lambda_c}^2-C_0}} = \dfrac{n_d-1}{V_d} \end{cases}$$

Where V_d is the Abbe number, n_d is the refracting index of the material at $\lambda_d = 583.5\ nm$, while $\lambda_f = 486.13\ nm$ and $\lambda_c = 656.27\ nm$.

The analytical solution of this system exists, but its complexity inevitably leads to numerical errors, for this reason it is convenient an iterative resolution of the system, the iterative resolution, however, can give rise to instability and be sensitive to the initial point, to overcome this problem we tried an iterative method that minimizes these two aspects. The method that allowed us to obtain a stable solution is the Banach's fixed-point theorem [12]; in order to solve the system through this method is necessary to put it in the following way:

$$\begin{cases} B_{k+1} = (n_d-1)\dfrac{{\lambda_d}^2-C_k}{{\lambda_d}^2} \\ C_{k+1} = {\lambda_f}^2 - \dfrac{B_k{\lambda_f}^2}{\left(\dfrac{n_d}{V_d}+\sqrt{1+\dfrac{B_k{\lambda_c}^2}{{\lambda_c}^2-C_k}}\right)^2-1} \end{cases}$$

The simplest approximation law, Cauchy's law, can be written in the form:

$$n = B + \sum_i \frac{C_i}{\lambda^{2(i+1)}}$$

The constraint of the two parameters leads us to have to simplify the law as follows:

$$n = B + \frac{C_0}{\lambda^2}$$

As it is trivial to observe the resolution of this law, that is the extraction of its coefficients does not give rise to transcendental systems, it is therefore possible to set the following system:

$$\begin{cases} B + \dfrac{C_0}{{\lambda_d}^2} = n_d \\ B + \dfrac{C_0}{{\lambda_f}^2} - B - \dfrac{C_0}{{\lambda_c}^2} = \dfrac{n_d - 1}{V_d} \end{cases}$$

With obvious symbolism, through some algebraic steps it is possible to solve the system obtaining:

$$\begin{cases} B = n_d \dfrac{C_0}{{\lambda_d}^2} \\ C_0 = \dfrac{\dfrac{n_d - 1}{V_d}}{\dfrac{1}{{\lambda_f}^2} - \dfrac{1}{{\lambda_c}^2}} \end{cases}$$

This method has allowed us to obtain two methods of compensation having the differences seen in the figure. The use of Cauchy compensation is preferable in cases where it correctly approximates the dispersion, in fact, its simplicity does not give rise to indeterminate regions as it happens for that of Sellmeier, in fact in the latter as it is trivial to observe, as a function of the parameter C_0 you can have zeros in the domain.

As a last note we want to present a peculiar feature of the approximations developed, in many cases it is not possible to have available the Abbe number and the refractive index for the materials under consideration and it is therefore important to have available a method of estimation. Once a reasonable refractive index value has been set for the material being analyzed, it is possible to estimate the Abbe number by compensating for the artifact created by our compensation method. This artifact is presented as an asymmetry of the FTT peak associated with the material of interest, in particular if the steepest part of the peak is on the left then the Abbe number set is too low, vice versa if the steepest part of the peak is on the right then the value set is too low. With this methodology, it is possible to infer the correct parameters to properly compensate for chromatic dispersion.

3.5.2 Artifacts

The approach to the solution has been to identify an elaboration chain composed of some simpler modules, which would allow obtaining a certain degree of flexibility and at the same time a satisfactory effectiveness and stability. This solution was the result of many attempts that have highlighted specific criticalities of the signals under analysis allowing us to design the current solution.

The scheme of the GRAB algorithm is a general one identified by three modules: Clustering, Classification, and Filtering. Between filtering and classification, a feedback ring allows a continuous improvement of the reliability of the output signal.

The clustering module is responsible for the identification of surfaces in the input signal; this is possible because in this signal the surfaces are generally associated to the main clusters. The same module must also recognize and discard false surfaces and noise exclusion is a plus.

The second module must correctly attribute the new surfaces identified by the clustering module with those previously identified, giving continuity to the signal. This task can be seen as a classification task if you think of it in the term of classifying the new input surfaces into the classes of the previous surfaces.

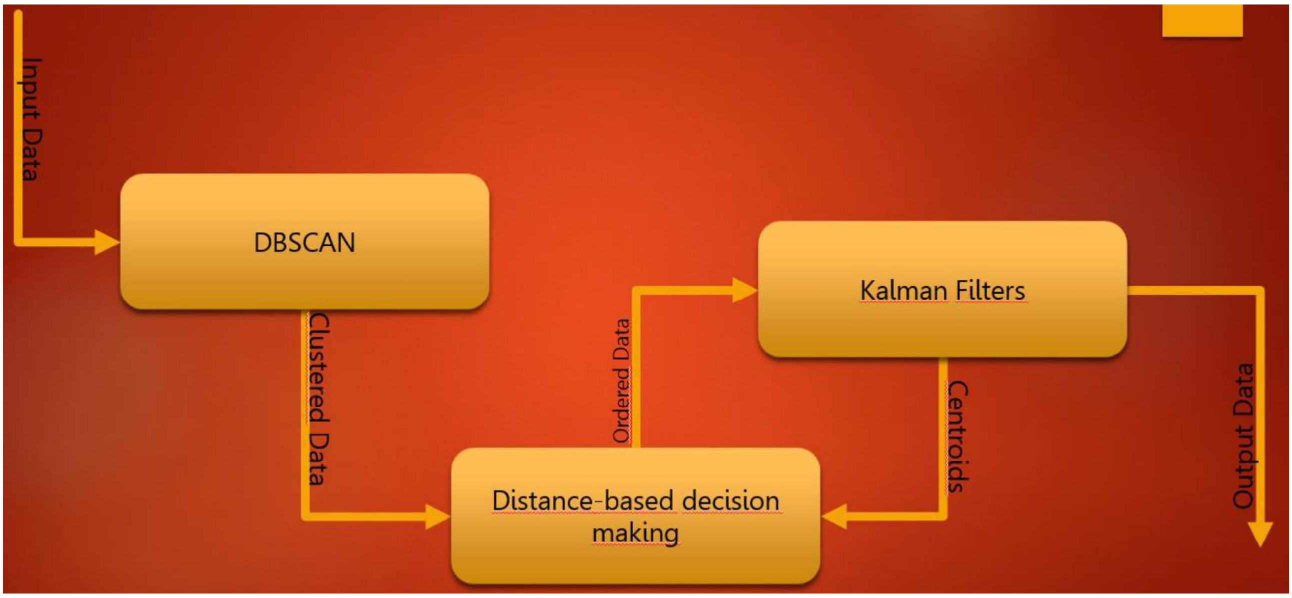

Figure 142: GRAB algorithm schema

The filtering part has the task of reducing the noise of the output signal from the classification module and provide the predictions that will go into feedback to the classifier.

This structure allows obtaining a great flexibility thanks to two key factors, the first is the modularity, and in fact, the algorithm can exclude one or more modules according to the constraints due to the processing resources of the device and the criticality of the signal under examination. The other element of flexibility is given by the use of macro classes of algorithms within the modules, for example within the filtering module can have Kalman filters [13], Gaussian filters, Kolmogorv-Zurbenko filters [14], etc.

In the case of the analyzed signals and taking into account the processing capabilities

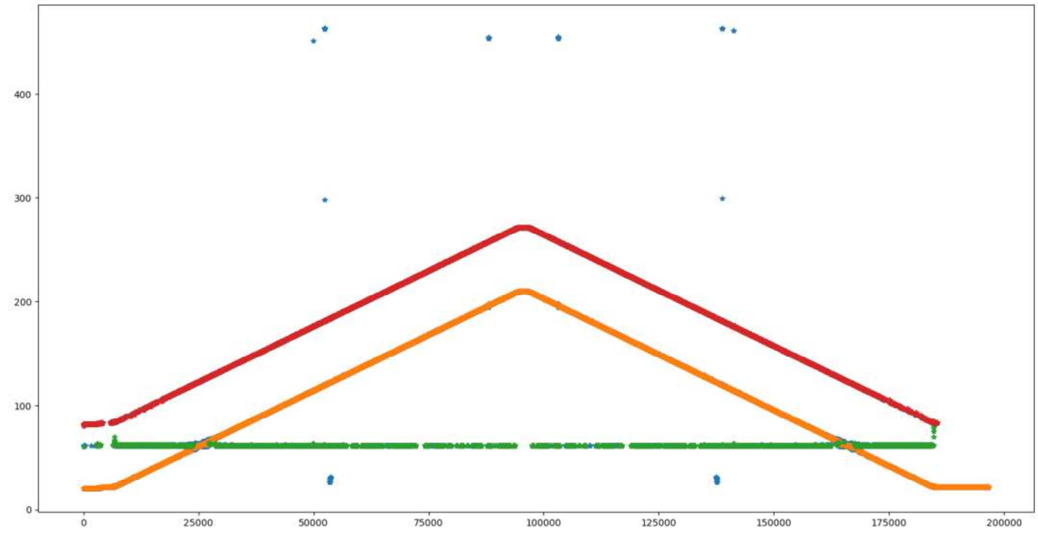

Figure 153: Wedge analyzed by GRAB

(although the algorithm was initially designed offline) of the target device, we found

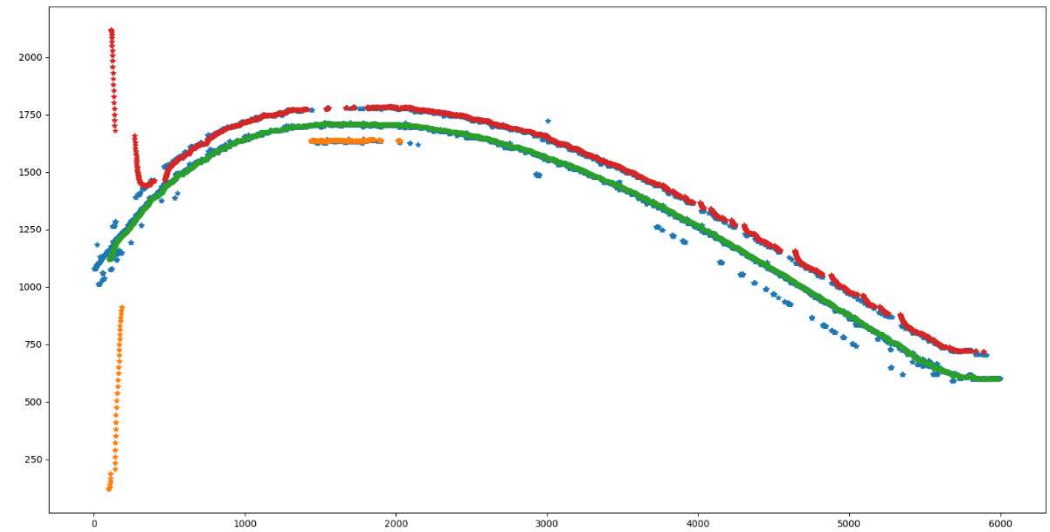

Figure 164: Double-layer material analyzed by GRAB

that the best combination of these three modules was inserting DBSCAN as clustering algorithm, a custom classifier tailored to our use case and Kalman as filtering module. This combination produced excellent results both in terms of performance (patently from the data in the figure we obtained the results **Fig.23** and **24**), reconstructing the reference profile correctly, and in terms of execution time on the target.

3.5.3 Developing GRAB algorithm

The evolution that the algorithm has had during its development will give us the opportunity to highlight the main criticalities of the signals under examination. The main problem of following surfaces is due to artifacts present within the signal, there are sporadic discontinuities, noise, and absence of signal and points of union between multiple surfaces. These issues require different strategies to be solved, first we tried to solve the discontinuities and then solve the other issues as well.

The first implementation that allowed us to partially solve the problem of discontinuities was one that made use of polarized DBSCAN; this polarization was introduced through the inclusion of a variable number of points within the dataset on

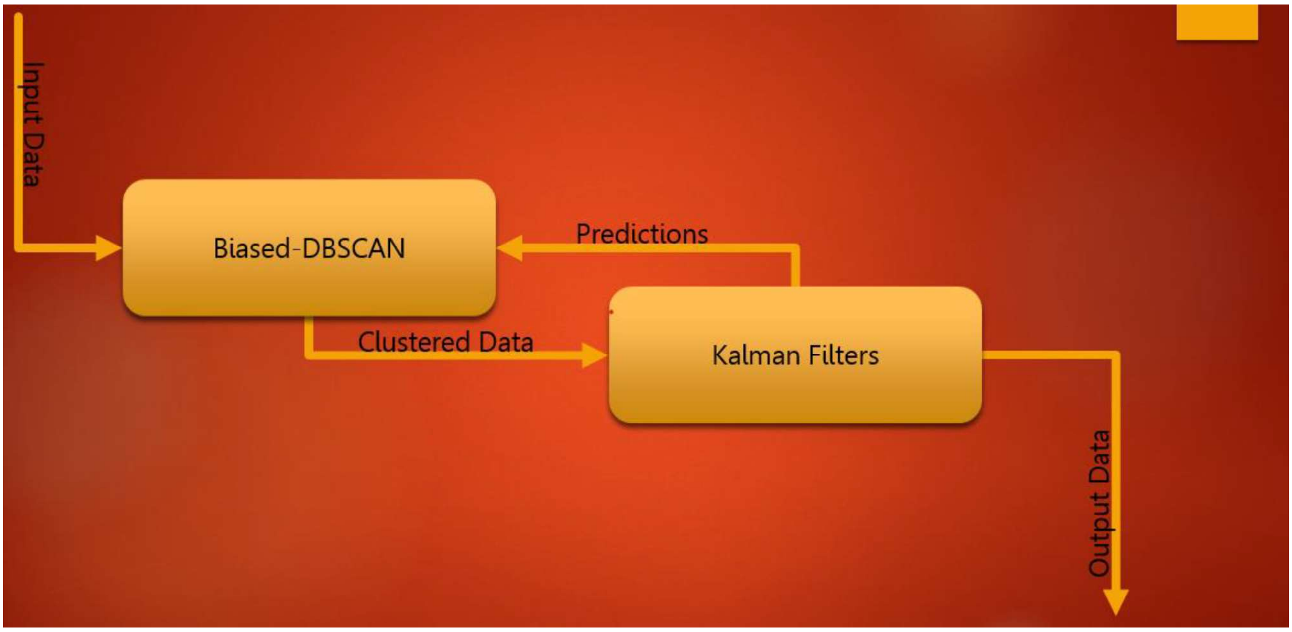

Figure 175: Biased DBSCAN algorithm schema

which to do the clustering, these points were derived from predictions related to the trend of clusters based on past values. The polarization was then done by controlling the parameters of the clustering algorithm; given the nature of DBSCAN, it is trivial to observe how this allows adding memory to the clustering process. The necessary predictions were entrusted to the Kalman filters, the reason for this choice is to mitigate noise on the one hand and offer predictions on the other, with only one processing module.

As can be seen in **Fig.25** this architecture composed of only two feedback modules allows to fill some discontinuities and to recognize as artifacts some clusters that otherwise would have been recognized as surfaces. The main problem in this solution is related to the predictions of the Kalman filter, when the low-frequency noise is intense, the predictions are less accurate and risk going to bias wrongly DBSCAN. To solve this problem we tried to make the signal more stable between the clustering algorithm and the filtering by adding an additional regression module.

Regarding the regression module, we have tried both robust polynomial regressions and linear regressions (e.g. Theil-Sen) without improving the performance compared to the previous solution. Given the errors that the feedback of the Kalman filter could introduce into the algorithm and given the poor results obtained through the inclusion of a regression module we preferred to opt for a lighter implementation and more suitable for an implementation on the sensor, but that would require a parameterization, especially of DBSCAN, not trivial.

The association between the filters and the clusters was however required and in this light implementation they were attributed according to the measurement value, i.e. the cluster with the lowest average measurement value was attributed to the first filter, the second was fed with the next one etc. This type of approach, while simple, still allowed this algorithm to solve both the noise and discontinuity problems.

Until now, we have tried to solve only two of the problems that generally occur in the signals of our interest, when we tried to apply the solution found to solve also the problems of crossover and signal loss this algorithm becomes no longer adequate. The proposed solution however also solves the problems described and is the solution that has been designed as a modification to that developed until then to solve these two additional cases.

3.6 On-device implementation

The GRAB algorithm designed according to the logic described above was intended to be implemented in an embedded context, so the algorithm was kept simple. Modularity, as mentioned earlier, is an element of reducing the computational cost of the algorithm.

The target is a device equipped with a Cortex A-9 with four cores, the algorithm has an entire core that shares with the operating system and must respect the real-time constraints required by the use of the device.

The complexities faced to implement and execute this algorithm are many; the first is the lack of highly efficient computing libraries (for DBSCAN) by the operating system, the choice of libraries for the implementation for Kalman filtering, the adaptation between data.

The need to use very efficient libraries not only allows us to perform calculations in an optimized way (e.g. decreasing the number of instructions needed), but also to optimize the organization of data in memory, this aspect allows to reduce the latencies of data fetch. The problem inherent in this management however is that, generally, different libraries have different data structures creating the need to interpose between the two libraries an adaptation layer. Depending on the complexity of this layer, the benefit of using high performance libraries may be lost.

The aim of this specific PhD activity is to verify if 1 processor core shared with the operating system is sufficient, if it is possible to use high performance libraries for the implementation (e.g. recompiling them for the current operating system and/or backporting them) and if the use of these libraries is advantageous over a more naive implementation.

The implementation activity, carried out in C++, was able to establish that it is possible to use the GRAB algorithm on board the device, imposing some limitations on the programming parameters of the device. In particular, we are forced to decimate the output rate downstream of the algorithm, in particular, it is not possible to provide an output value for each new measurement value because in this scenario it would be necessary to complete the execution of the algorithm every 250 us.

A more likely scenario is the one offered by the possibility of providing a new value every 5 ms, in this way it is possible to produce a measurement value every 20 acquired measures. Tests have shown that it is possible to extend up to 100 acquired measures and process the entire window within 5 ms, allowing us to create overlap between consecutive windows.

To better understand the impact that the window has on processing time, it is necessary to better analyze the complexities of each module. Analyzing the complexity of DBSCAN we have an $O(n^2)$,with n number of points to be clustered, and therefore the size of the window. The classification module has a complexity that depends only on the number of surfaces under trace and clusters exiting the clustering module, while the Kalman filters have a complexity that depends on their number, the order of the filter.

From this picture we can see that as the window increases, the complexity $O(n^2)$ becomes the general complexity of the actual algorithm. However, this fact is not supported by the tests conducted, in which the growth would not seem to be parabolic, but it was not possible to investigate further. (May be affected by huge constant term or linear one?)

In order to make available the libraries for the computation of the clustering and filtering modules, we tried to compile the source code of the libraries on the target, but the compiler version too obsolete did not allow to arrive at a correct compilation for the versions of interest. Changing compiler would have required updating also the dependencies connected to it and in the end, it was considered too expensive.

The second strategy was to try to backport the libraries from a newer operating system version to the current one, but was immediately discarded, because the complexity of this operation was not less than the previous method.

The last tried solution that turned out to be the one used to date is to use a chroot in which a newer version of the operating system was installed and that allowed the installation of the required libraries. This method, which poses many maintenance issues, has allowed us to obtain a build and run system suitable for our purposes without changing the rest of the ecosystem.

4 Conclusions

The objectives behind my studies were mainly two. The first one was to investigate the differences between the main cloud providers, both in terms of price, also analyzing the different cost models, and in terms of technology. On the other hand, the second objective was related to the implementation of data-driven algorithms for the improvement of sensor edge performance. This last argument finds a strong correlation with the first one since data-driven algorithms, as the term says, are based on data that could be provided by a cloud infrastructure.

The contribution that this research has brought on the choice of a cloud provider in the IoT field is to have identified a method of selection of providers characterized by successive steps that allow having an overview on costs. Regarding the development of algorithms on embedded systems, the contribution that this research has brought concerns the resolution of two problems related to interferometric sensors. This contribution provides two additional methodologies to those currently available for solving the problems already widely discussed. Regarding the chromatic dispersion, the research work proposes a simple method for compensation both during calibration and at the implementation level; in opposition, the method proposed for the correct reconstruction of the material under examination is more complex and sophisticated, trying, however, to maintain a high flexibility and extensibility.

In order to obtain an even closer relationship between the topics covered it would be appropriate in future developments to complete the feedback loop between the cloud implementation and the algorithmic part creating a virtuous circle that allows an optimal parameterization of the algorithm or their refinement.

5 Bibliography

[1]Ortiz, Jesús & Marroquin, William & Cifuentes, Leonardo. (2020). Industry 4.0: Current Status and Future Trends. 10.5772/intechopen.90396.
[2]https://azure.microsoft.com/en-us/resources/beyond-predictive-maintenance/
[3]https://opcfoundation.org/about/opc-technologies/opc-ua
[4]https://www.itu.int/rec/T-REC-G.114-200305-I
[5]https://en.wikipedia.org/wiki/Dispersion_(optics)
[6]https://en.wikipedia.org/wiki/Abbe_number
[7]https://en.wikipedia.org/wiki/Sellmeier_equation
[8]https://www.mlpack.org/
[9]https://opencv.org/
[10]https://scipy-cookbook.readthedocs.io/items/robust_regression.html
[11]https://en.wikipedia.org/wiki/Theil%E2%80%93Sen_estimator
[12]https://en.wikipedia.org/wiki/Banach_fixed-point_theorem
[13]https://en.wikipedia.org/wiki/Kalman_filter
[14]https://en.wikipedia.org/wiki/Kolmogorov%E2%80%93Zurbenko_filter