1 Introduction

2 Distributed Systems

Characterists from an distributed system: When we speak about an distributed system, theres an list of things that we need to think of, in order to understand the main characteristics that they share in computational terms. Without further abuse, the main characterist that more explain the nature of an distributed system are:

Concurrency, this stands for an system that can handle multiple proccesses that run simultaneously, this of course in different machines that in an certain time align with each other to reach an common goal.

Scalability, relatively to this precise characteristic, we know that if we have an distributed network, we must ensure that in an certain way we can scalate the system. This scalability can either be vertical(hardware) or either horizontal(number of instances), which has the goal of increasing the performance of an certain system.

Fault-tolerant, this is much of importance, because in some distributed systems it becomes incommensurable to not have counter-measures to faults. This faults could either be:

Transient faults, it occurs when there is an temporary fault that occur due to a sudden and brief disruption in the system.

Intermittent faults, which occurs sporadically and unpredictably, making them difficult to diagnose and detect.

Partial Faults, this one occurs when an partial part of the components fails, but the system still works without that failure of the component.

Byzantine Faults, finally it occurs when an component behaves malicously or incorrectly, this are the most severe and complex faults in an distributed system. There are more faults but those are what we as an group think are more relevant.

Coming back to the characteristics of an distributed system we also have transparency, which stands for an user not having the capability of knowing which processes are involved in an distributed system that he is interacting with.

Heterogeneity, an distributed system may have different resources, behing heteregeneous, means that it can have different pieces and still give the same importance for each of them and possibily dont even knowing his importance

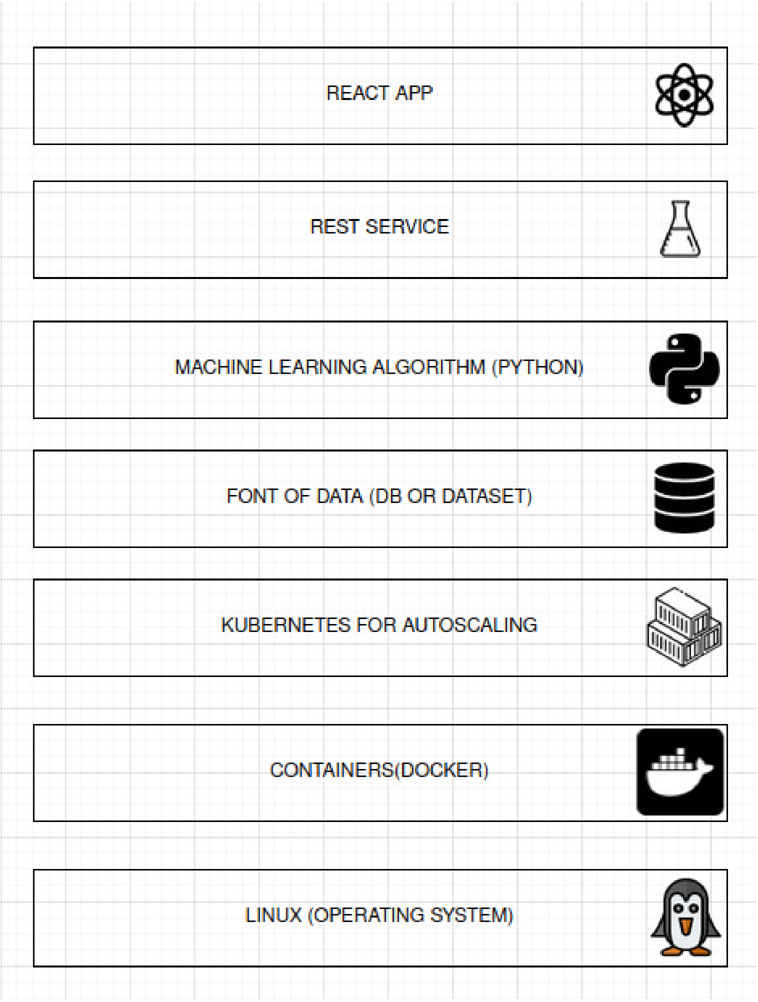
REACT APP
REST SERVICE
MACHINE LEARNING ALGORITHM (PYTHON)
FONT OF DATA (DB OR DATASET)
KUBERNETES FOR AUTOSCALING
CONTAINERS(DOCKER)
LINUX (OPERATING SYSTEM)

Figure 1: Arquitetura de camadas

and magnitude.

Resource sharing, an distributed system allows users to share resources, shuch as data, processing power, etc...

There are a lot of characteristics, and every characteristic may have an specific naming for an more specific situation, but those are the main characteristics and they do represent well how should an distributed system behave.

3 Information System

Integrating an AI facial recognition system into a Kubernetes-based infrastructure can have several benefits for organizations. Firstly, it enhances security by accurately identifying and verifying authorized personnel, while preventing unauthorized access. This helps prevent data breaches and reduces the risk of security incidents.

Secondly, the system can improve efficiency and streamline operations. With facial recognition, manual identity verification and access control procedures are eliminated, reducing wait times and improving overall productivity. Additionally, the system can help organizations comply with regulatory requirements by ensuring that only authorized personnel have access to sensitive data.

Overall, incorporating an AI facial recognition system into a Kubernetes infrastructure can provide significant benefits for organizations, improving security, efficiency, and compliance.

The stakeholders of a football stadium include a variety of individuals and groups who have an interest in the operations and management of the facility. These stakeholders can include the stadium owners, management, staff, fans, and the local community. With the rise of facial recognition technology, stakeholders and their managers have had to consider the implications of implementing such a system within the stadium.

First and foremost, stadium owners and management may be interested in implementing facial recognition technology as a means of enhancing security within the facility. By using this system, they can monitor individuals who are entering and exiting the stadium and identify any potential threats or individuals who may have been banned from the stadium. This can help to increase safety for fans and staff members alike.

Facial recognition technology can also be used to improve the overall fan experience within the stadium. For example, fans can use this technology to gain access to the stadium without needing a physical ticket or pass. They can simply scan their face at the gate, and the system can quickly verify their identity and allow them to enter. Additionally, fans can use this technology to make purchases at concessions stands or merchandise booths, allowing for a more streamlined and efficient experience.

Of course, implementing facial recognition technology also has potential drawbacks and concerns that stakeholders and managers must consider. One major concern is the privacy of fans and staff members. There are fears that this technology could be used to track individuals beyond the stadium or that personal data could be shared with third parties without consent. Additionally, there are concerns about the accuracy of facial recognition technology and the potential for false positives or mistakes.

In light of these concerns, stakeholders and managers must take steps to ensure that any facial recognition technology used within the stadium is implemented responsibly and with transparency. This could include providing clear information to fans and staff members about how their data will be used and stored, establishing clear protocols for data protection and sharing, and allowing for opt-out options for individuals who do not wish to participate in the system.

Ultimately, the use of facial recognition technology within a football stadium can offer both benefits and challenges for stakeholders and managers. It is important that these individuals work together to carefully consider the implications of such a system and ensure that it is implemented in a responsible and ethical manner that prioritizes the safety and privacy of all individuals involved.

In distributed systems, the system boundary is the virtual ”fence” that separates the system from the outside world. It helps to define what the system is responsible for and what it is not responsible for. Understanding the system boundary is important for designing, developing, and managing the system effectively, as it allows us to identify resources and communication channels needed for the system to work and to secure and protect it from potential threats.

When naming several system boundaries, to touchable purpose of couse, we can have something like: Not behing able to save sensitive information about the envolved parts, the algorithmn needs to be right almost one hundred percent of the times, we can only inspect this once, we only can inspect one by one person, the system cannot go down, the system must be capable of scaling and the system must be regulated.

Requirements in project management are the specific objectives and criteria that must be met for the project to be successful. Effective management involves identifying, prioritizing, documenting, and tracking requirements throughout the project. Stakeholder engagement and feedback are also critical to ensure the project aligns with their needs and expectations. Properly managing requirements helps deliver value and minimize scope creep and delays. Now, moving more into what is our project, what we do see as an important requirement is: the system must scallate, the system must have an uptime of almost one hundred percent, the system must have robusteness, the system needs to be fault tolerant, the system must be fast, the system must respond in real time and the system AI algorithm must be right almost one hundred percent of the time.

4 Architecture

Why is architecture important?

An architecture in distributed system's can be very much of an increment of value, because:

It ensures that good practises are followed, what we mean by this is that by thinking more about how an system could be contructed, we will try to understand which details we need to think off, in order to have an more well designed distributed system.

It is tought, according to previous experiences of more influent people, because when we search uppon what could be the best to our idealization of system, we end up reading books and articles of more experienced people experiences and workovers.

We will not invent the wheels in some cases, this means that you may found some patterns for your architecture that you can follow and still be sure that the system will acomplish what you want to acomplish. What you may not found in one pattern, you can find in an composite of patters, by mixing them. Also, you can always take out ideas from them and use what you think is more advantegous for your system.

Finally, you will have a more robustness system, because by thinking and planning the system out will ensure that most of the cases will be tought and eventually take cared of.

5 Outline Requirements

The essential requirements for a software system include a database to store user facial images, face recognition models, an image processing engine to extract facial features from the images, high availability and fault tolerance to ensure the system remains operational even if some microservices fail, efficient communication protocols that minimize network latency and reduce overhead, and monitoring and logging mechanisms to track system performance and identify issues quickly.

6 Cost

The cost of implementing and maintaining a distributed system of face recognition can be high, especially if it is a large-scale system. Therefore, it is important to prioritize what is most important for the business.

Is it worth investing in the most expensive software to ensure maximum accuracy for the model, or is it more important to have less latency even if it results in some loss of accuracy?

Alternatively, should both factors be considered, with budget not being a constraint?

In this perspective, the cost of implementation is exponential, given that complex models are hard to train and time-consuming. Therefore, it is necessary to use the best software available.

The overall budget for this project can range from 165,000 to 265,000 euros. This budget is broken down into several categories, including human resources (20,000 euros), cameras (50,000 euros), servers (25,000 euros), network infrastructure (20,000 euros), software licenses (10,000 euros), and maintenance, training, and support (10,000 euros). Additionally, there is an optional expense for GPUs, which can range from 10,000 to 100,000 euros. These costs will enable us to complete the project efficiently and provide better service to our clients. The budget will be closely monitored to ensure that we stay within our financial constraints.