1 Introduction

This work was developed within the Distributed Information Systems Architecture discipline. Initially the group focused on looking for a problem for which we would develop a solution. After discussion, the team came to an agreement that we would work on developing a solution to streamline the process for members to enter stadiums to watch the football matches of the various teams in question. Then we try that the answer results in a distributed system, with an organized architecture.

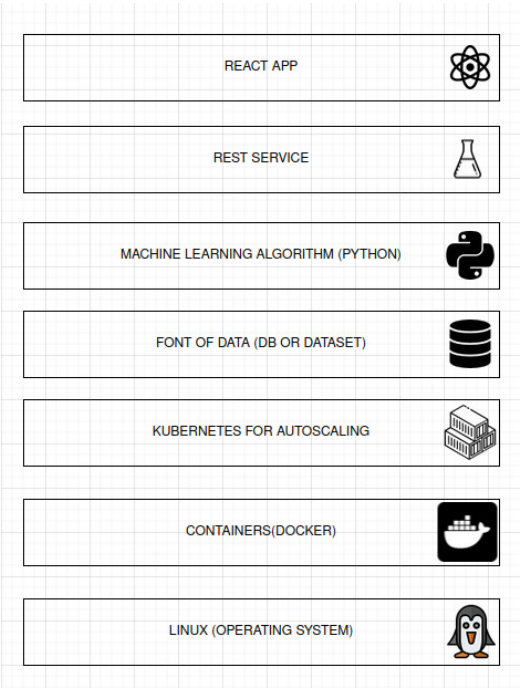

Figure 1: Example figure

2 Distributed Systems

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 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. Our solution is related to Distributed Systems due to the fact that we are aiming for it to be one. By definition, a Distributed System consists of a set of autonomous machines that are connected to the same network with different components that interact with each other, communicate and coordinate their actions using the exchange of messages as a resource. Therefore, our objective with this project is that if a microsystem fails, there is no compromise of the whole system. Furthermore, we intend that all components be able to communicate with each other, by exchanging messages as is done in a distributed system.

3 Information System

When there is a big game for example, when it comes to Portuguese football, when Porto and Benfica play with each other, there is always a huge line of people to enter the stadium, some of those supporters are, what we call partners of the club, that is they bought special passes that let them watch every game of the respective team they support. But event tought they are special supporters they have to wait in the same line as the normal people who bought tickets, so the system we want to develop will allow to create a new entrance that uses face recognition system that will help to identify which the supporter are partners and allow them to enter the stadium without waiting in any line. This will facilitate the work of the security/police officers that are at the entrance and allow them to focus only on the inspection work and finally will motivate the people to buy those speacil passes, just because it's faster and safer.

A stakeholder is a person, group, or organization that has an interest or concern in a business or project. This interest or concern can be financial, strategic, operational, ethical, or any other aspect related to the success or failure of the venture. For this system the main stakeholders we see having are the developpers, the project manager, the analyst, the designers, other members of the team and the football teams (a quick side note is that we are going to focus on football clubs, but the system can be integrated with other business).

4 Architecture

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.