Group Project

Biometric Systems

Valerio Casalino (1916394)[1] Mario Tobia Vendrame (1922290)[1]
Shaahin Sabeti Moghaddam (1917507)[1]

[1]Cybersecurity Master @ Sapienza Università di Roma

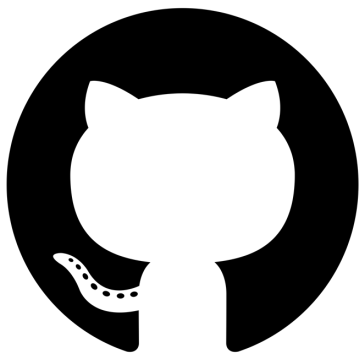
Fall 2019

Table of Contents

SAPIENZA
UNIVERSITÀ DI ROMA

Premise

Before we start, let us say that all of our work, included this own presentation, is open sourced and available on Github:



https://github.com/casalinovalerio/biosys-project

There is also a script to replicate our setup for future projects.

Overview

We wanted a face recognition based authentication application that is simple, yet particular. We deployed our test using:

- A **web interface** that works as a demonstrative placeholder. It gets the face with the camera, makes requests to our API server, which returns only a binary value for the success of the authentication.
- An **API server**[1] that queries the faces database and recognizes faces using the **@ageitgey's tool**[2].
- A **database based on Blockchain**[3] that is an open source wrapper for a blockchain database that can be queried with standard SQL syntax. Implemented on the API server too.

[1] Hosted by Digital Ocean: https://www.digitalocean.com/
[2] Github project here: https://github.com/ageitgey/face_recognition
[3] Implemented by Bigchaindb: https://www.bigchaindb.com/

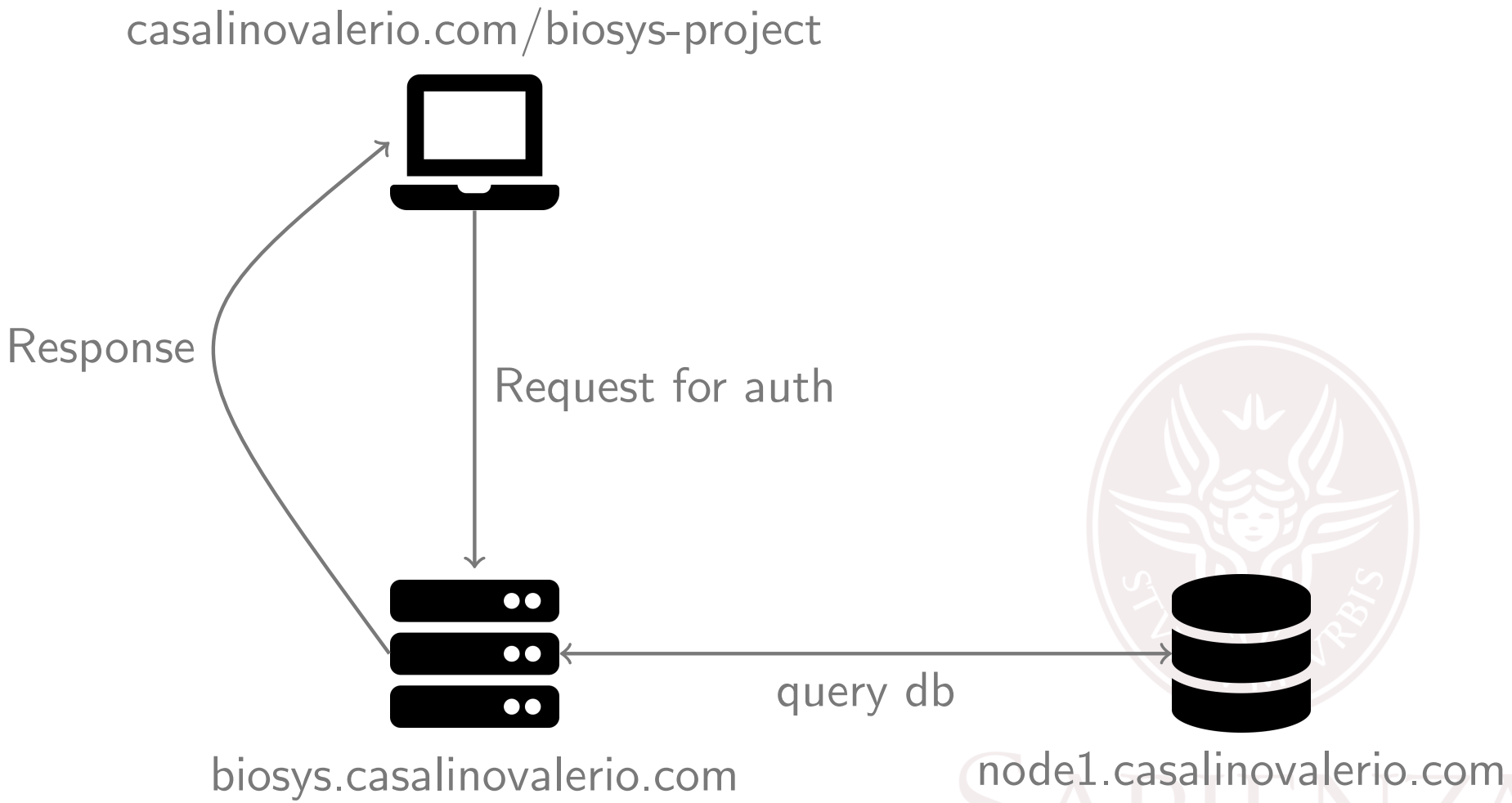
Description

We built a face recognition service based on **identification**, which matches the probe with an **open set** approach.

The gallery is stored in a **Blockchain database** and queried each time, but one of the nodes is stored in the same place in which the request is made, that makes the whole process **faster**.

Each match, if there is one, is returned with a **distance** parameter.

Overview scheme[4]



[4] Icons are licensed under CC-BY 4.0. https://fontawesome.com/license

Table of Contents

SAPIENZA
UNIVERSITÀ DI ROMA

The web Application

The web application performs the authentication as follows:

- Captures frames in a canvas.
- Analyzes them through the opencv's javascript[5].
- As the time one button is pressed, the canvas frame is sent to our server[6], which can register the face, or match the face with an already registered user.
- In the end, you can decide to be registered or you can query the system looking for a match of your face.

[5] https://tinyurl.com/s2yprk7

[6] https://biosys.casalinovalerio.com

Does it work?

We actually did some **serious** testing on it. As you can clearly see in the picture below, it works!

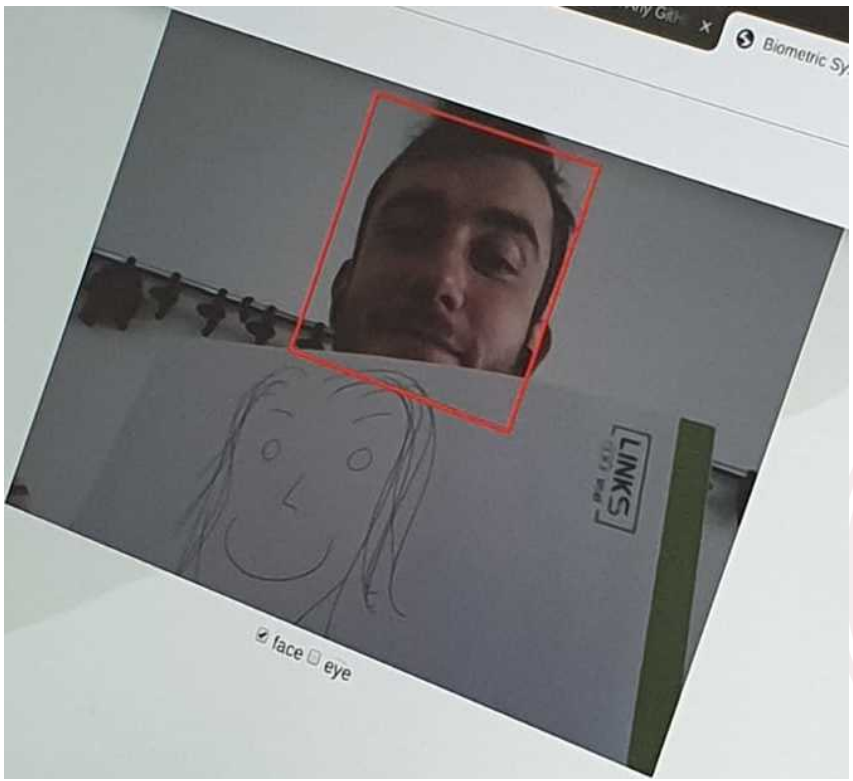


Table of Contents

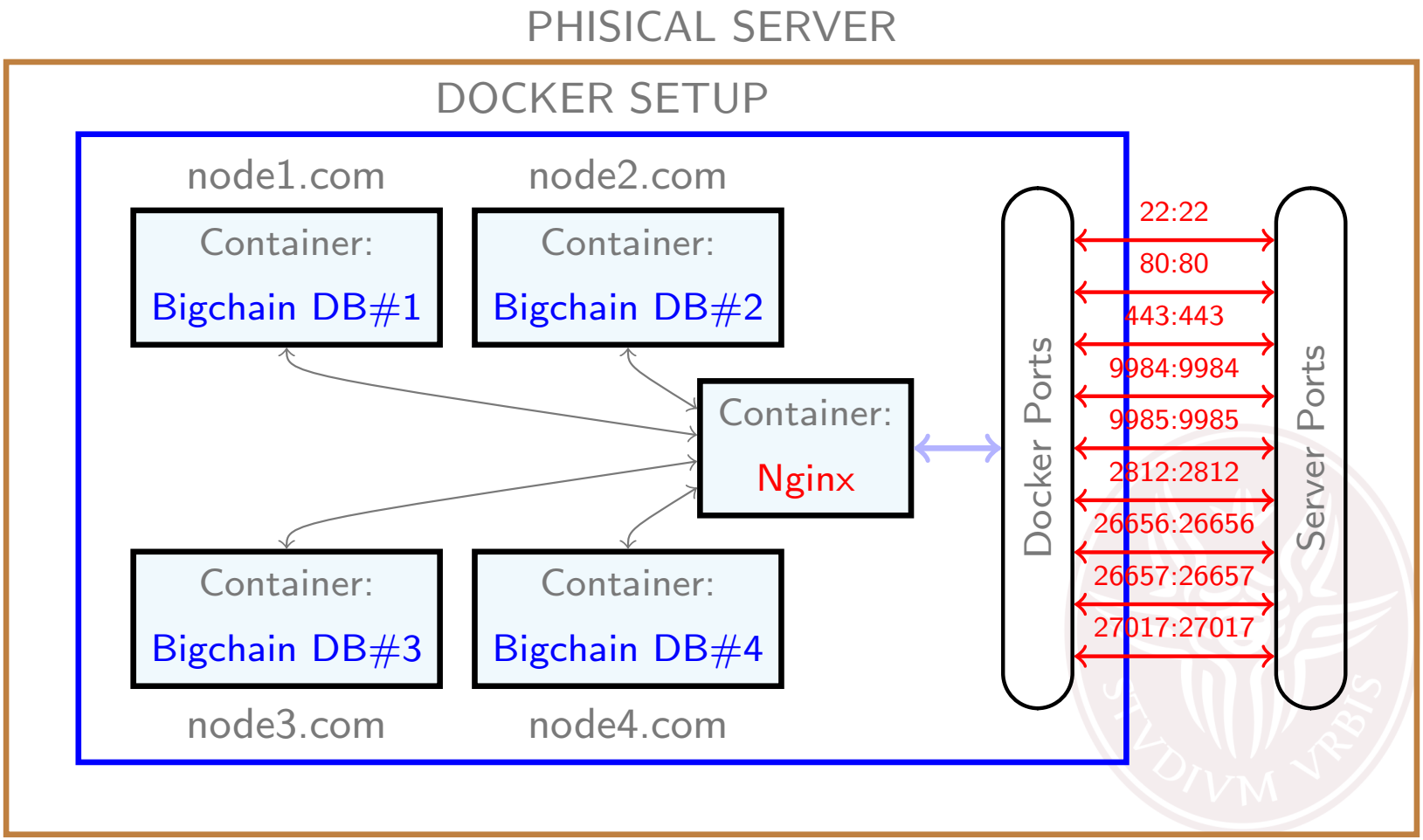
SAPIENZA
UNIVERSITÀ DI ROMA

The Block-chain database

As database for new faces, we implemented a **Block-chain**.
We used an open-source implementation of it, called BigchainDB[7].
We also used Docker[8] to deploy 4 containers running the application.

[7] Main page: https://www.bigchaindb.com. Documentation here.
[8] Main page: https://www.docker.com.

Architecture Implementation[9]



[9] This is absolutely not meant for a real deployment!!

How to interact with the DB

We are assuming that we have an enstablished connection set up.

Query data

```
connection.searchAssets('AwesomeAsset')
.then(assets => console.log('Found assets:', assets))
// Read the console to look at the assets
```

Load data (make a transaction)

```
// Create transaction first (txTransferBob)
driver.Transaction.signTransaction(txTransferBob,
alice.privateKey);
conn.postTransactionCommit(txTransferBobSigned);
```

Simple as that...

Table of Contents

SAPIENZA
UNIVERSITÀ DI ROMA

Connecting the Web app to API server

Send faces function in web app

```
  ...
var canvas = document.getElementById("canvasOutput");
picture.src = canvas.toDataURL();
...
xhr.open('POST', 'url/send-faces.php', true);
...
```

Recognize face function in web app

```
  ...
var canvas = document.getElementById("canvasOutput");
picture.src = canvas.toDataURL();
...
xhr.open('POST', 'url/reco-faces.php', true);
...
```

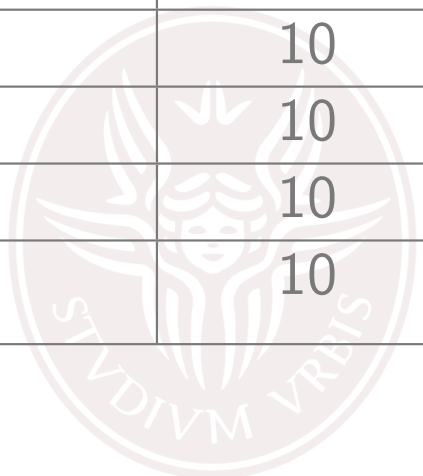
Table of Contents

SAPIENZA
UNIVERSITÀ DI ROMA

Testing approach (Training)

We trained the system registering 8 people:

Real name	id	#samples in gallery	#probes
Bill Gates	bill	60	10
Barak Obama	doubleb	60	10
Margot Robbie	harley	60	10
Miriam Leone	mlion	60	10
Scarlett Johansson	redlucy	60	10
Robert Downey Jr	robman	60	10
Tom Hardy	tommy	60	10
Mark Zuckerberg	zuck	60	10



Testing approach (Querying)

Then, we collected 10 other random pictures from the same people, plus 10 pictures for other 5 people:

- Jimmy Fellon
- Donald Trump
- Tim Cook
- Jeff Bezos
- Alfred Hitchcock

To be quick, we used yet another script[10] to upload pictures to be queried.

Use the script

./test /path/to/test-faces http://server.com /path/to/results.cvs

[10] This one: https://git.io/JvsQv

Home-made gallery results (raw)

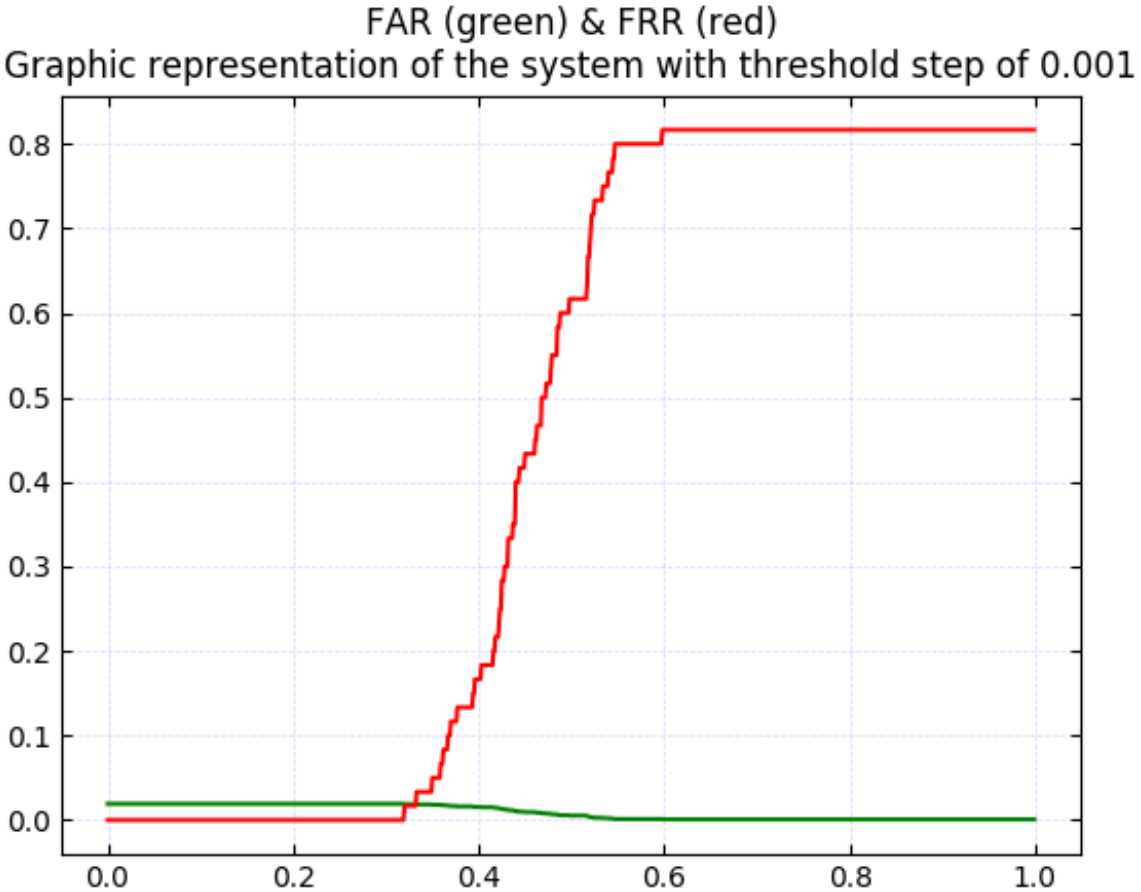
We let the script run and the results can be seen in this spreadsheet:

Google Sheets

https://tinyurl.com/rbk4dnc

Home-made gallery Results (Description)

We analyzed the results with a python script[11], and obtained these FAR and FRR:



[11] https://git.io/Jvsp6

LFW Results (raw)

We surely had more to test... That's why, instead of using our home-made data set, we used the **Labeled Faces in the Wild**[12]. This time, we used 75 registered users and 150 impostors, getting these results:

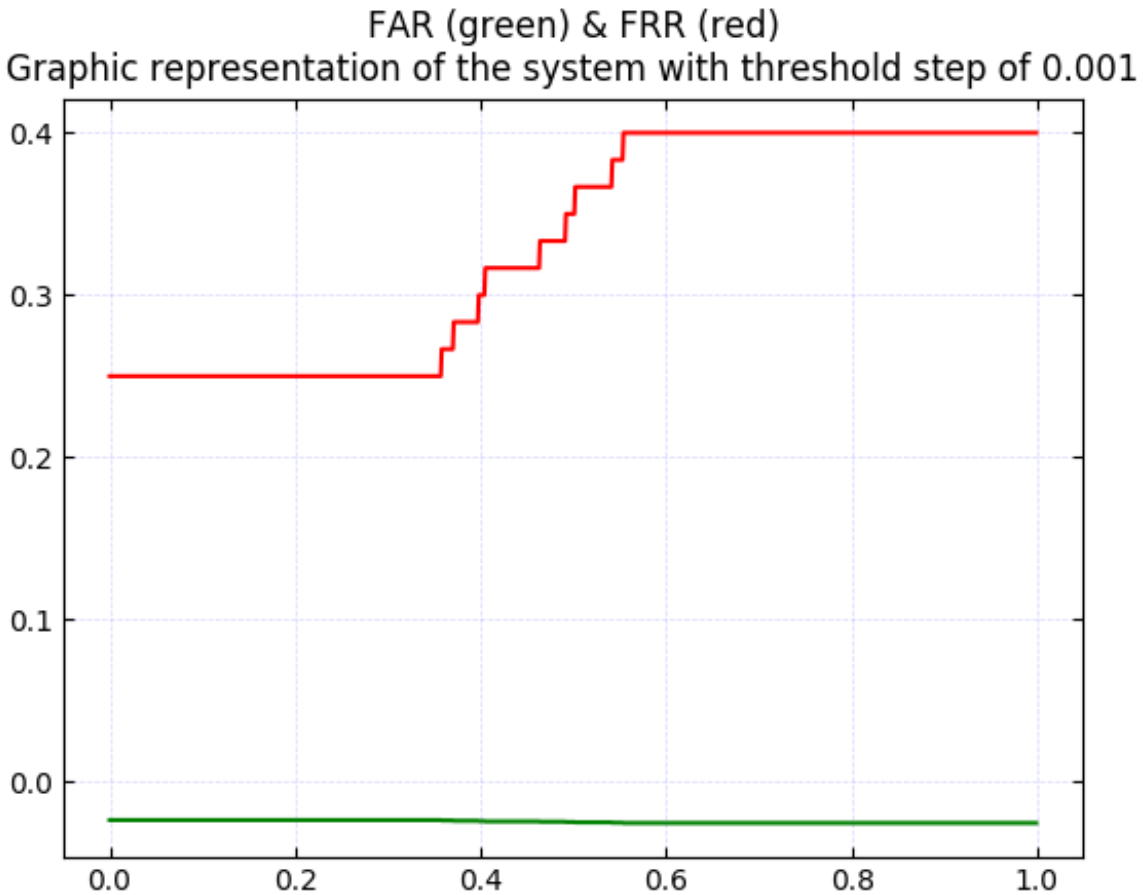
Google Sheets

https://tinyurl.com/skdethl

[12] http://vis-www.cs.umass.edu/lfw/

LFW gallery Results (Description)

We analyzed the results with a similar version of the precedent python script[13], and obtained these FAR and FRR:



[13] https://git.io/JvGB8

Table of Contents

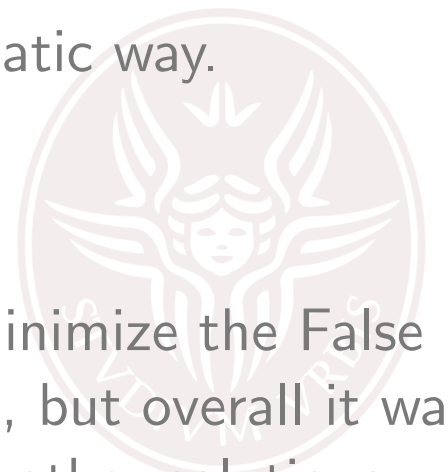
SAPIENZA
UNIVERSITÀ DI ROMA

Conclusions

It wasn't an easy project, but we did our best. At the end of it all, we learned:

- How to gather information on standards and OS solutions available.
- Deploy an actual service relying on ourselves.
- How to evaluate a BS performance in an automatic way.

The software that we used was clearly designed to minimize the False Acceptance Rate penalizing the False Rejection Rate, but overall it was simple to understand and simple to use compared to other solutions.



Greetings from the group

This is a really great ending message from the "creative" chilled-capibaras!

And this is a real cool catchy phrase!!