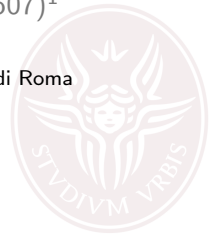# Group Project

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

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**[1] 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**[2] that queries the faces database and recognizes faces using the **@ageitgey's tool**[3].
- A **database based on Blockchain**[4] that is an open source wrapper for a blockchain database that can be queried with standard SQL syntax. Implemented on the API server too.
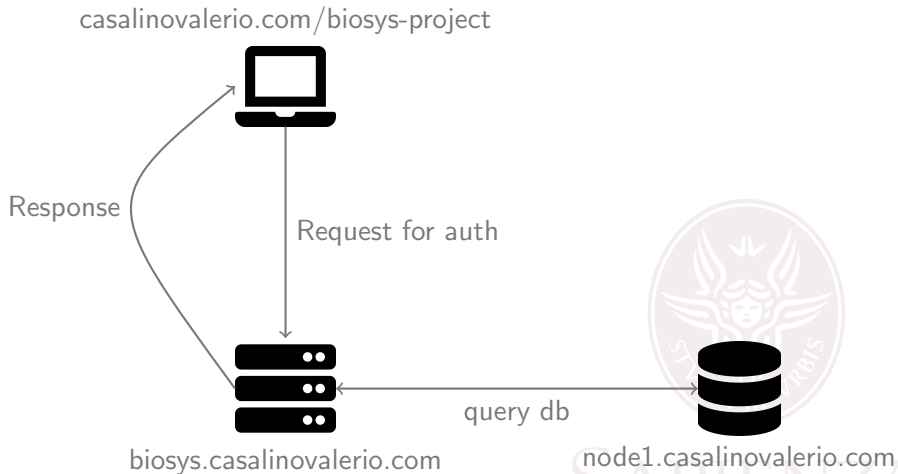
---

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

Overview scheme[5]

casalinovalerio.com/biosys-project

Response

Request for auth

biosys.casalinovalerio.com

query db

node1.casalinovalerio.com

Table of Contents

The web Application

The web application perform the authentication as follows:

- Captures frames in a canvas.
- Analyzes them through the opencv's javascript[6].
- As the time one button is pressed, the canvas frame is sent to our server[7], which can register the face, or match the face with an already registered user.
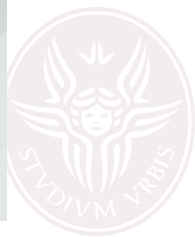- In the end, you can be registered, or you can get the authentication response status.

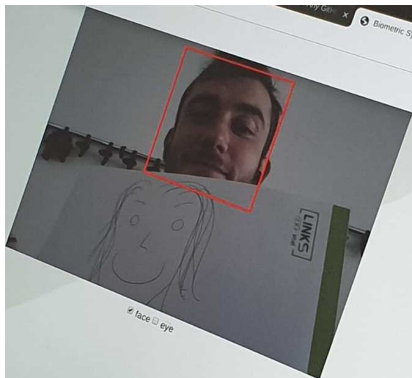---

[6] https://tinyurl.com/s2yprk7
[7] 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

The Block-chain database

As database for new faces, we implemented a **Block-chain**.
We used an open-source implementation of it, called BigchainDB[8].
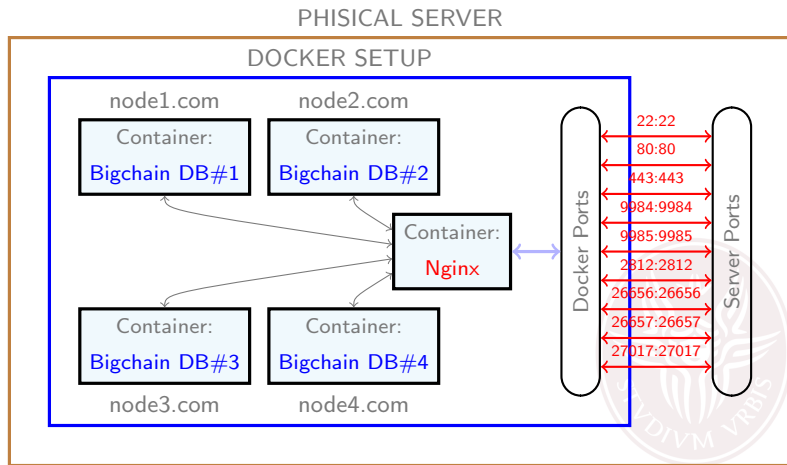We also used Docker[9] to deploy 4 containers running the application.

---

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

Architecture Implementation[10]

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...

SAPIENZA
UNIVERSITÀ DI ROMA

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);
...
```

Getting the response

This is how we did it:

Table of Contents

How we tested

We tested our solution with a custom script.

Use the script (to edit)

./test /path/to/test-faces

For more info on how the script works, just look at it[11], it is open source!

We let the script run on the same server for the longest time possible, until we could obtain a reasonable evaluation.

---

[11]Script here: https://tinyurl.com/wo3rk8s

Testing approach (Training)

We trained the system registering 8 people:

Real name	id	#samples in gallery	#probes
Bill Gates	bill	$\tilde{6}0$	$\tilde{1}0$
Barak Obama	doubleb	$\tilde{6}0$	$\tilde{1}0$
Margot Robbie	harley	$\tilde{6}0$	$\tilde{1}0$
Miriam Leone	mlion	$\tilde{6}0$	$\tilde{1}0$
Scarlett Johansson	redlucy	$\tilde{6}0$	$\tilde{1}0$
Robert Downey Jr	robman	$\tilde{6}0$	$\tilde{1}0$
Tom Hardy	tommy	$\tilde{6}0$	$\tilde{1}0$
Mark Zuckerberg	zuck	$\tilde{6}0$	$\tilde{1}0$

SAPIENZA
UNIVERSITÀ DI ROMA

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

Table of Contents

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.

Performance consideration here.

Greetings from the group

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

And this is a real cool catchy phrase!!