

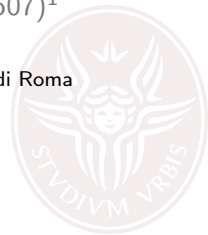
Group Project

Biometric Systems

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

¹Cybersecurity Master @ Sapienza Università di Roma

Fall 2019



SAPIENZA
UNIVERSITÀ DI ROMA

Table of Contents

General Concepts & Decisions

Front-end Implementation

Data-set Management

Biometric Scanning Integration

Performance Assessment

Conclusions



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**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.

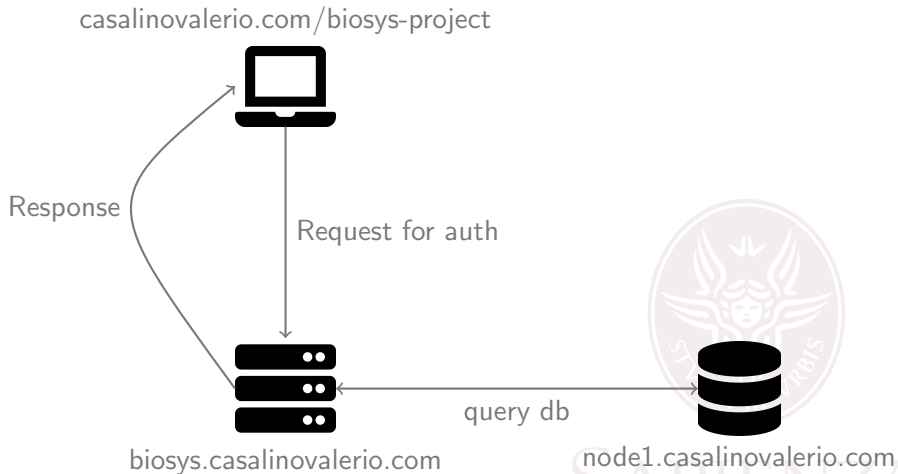
¹Hosted by Github Pages: <https://pages.github.com/>

²Hosted by Digital Ocean: <https://www.digitalocean.com/>

³Github project here: https://github.com/ageitgey/face_recognition

⁴Implemented by Bigchaindb: <https://www.bigchaindb.com/>

Overview scheme⁵



⁵Icons are licensed under CC-BY 4.0. <https://fontawesome.com/license>

Table of Contents

General Concepts & Decisions

Front-end Implementation

Data-set Management

Biometric Scanning Integration

Performance Assessment

Conclusions



SAPIENZA
UNIVERSITÀ DI ROMA

The web Application

The web application perform the authentication as follows:

- ▶ Captures frames in a canvas.
- ▶ Analyzes them through the opencv's javascript⁶.
- ▶ As the time one button is pressed, the canvas frame is sent to our server⁷, 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!

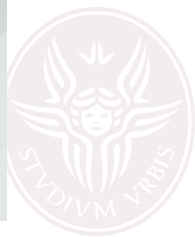
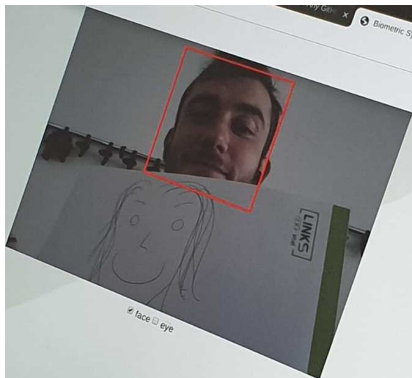


Table of Contents

General Concepts & Decisions

Front-end Implementation

Data-set Management

Biometric Scanning Integration

Performance Assessment

Conclusions



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 BigchainDB⁸.

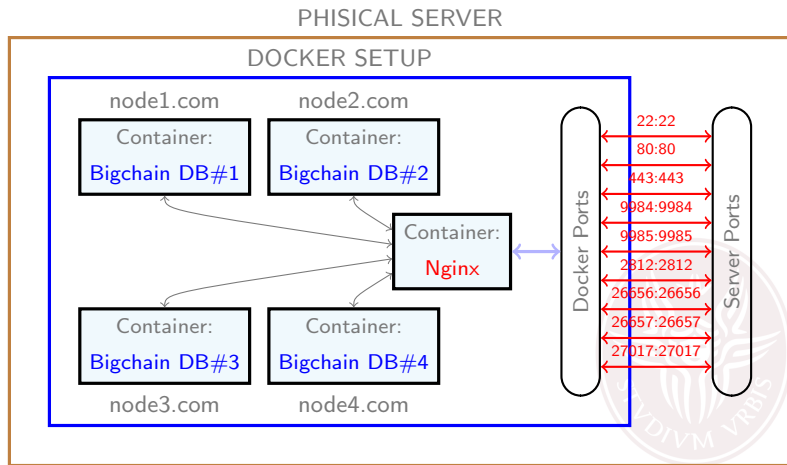
We also used Docker⁹ to deploy 4 containers running the application.



⁸Main page: <https://www.bigchaindb.com>. Documentation [here](#).

⁹Main page: <https://www.docker.com>.

Architecture Implementation¹⁰



¹⁰This 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

General Concepts & Decisions

Front-end Implementation

Data-set Management

Biometric Scanning Integration

Performance Assessment

Conclusions



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);  
...
```

Connecting the API to the DB

This is how we did it:



SAPIENZA
UNIVERSITÀ DI ROMA

Getting the response

This is how we did it:



SAPIENZA
UNIVERSITÀ DI ROMA

Table of Contents

General Concepts & Decisions

Front-end Implementation

Data-set Management

Biometric Scanning Integration

Performance Assessment

Conclusions



SAPIENZA
UNIVERSITÀ DI ROMA

How we tested

We tested our solution with a custom script.

Use the script (to edit)

```
./test.sh -d /path/to/test-faces
```

For more info on how the script works, just look at it¹¹, 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.

¹¹Script here: [this-is.temp](#)

Testing approach



SAPIENZA
UNIVERSITÀ DI ROMA

How it performed

Really well, it is a revolutionary project!



SAPIENZA
UNIVERSITÀ DI ROMA

Table of Contents

General Concepts & Decisions

Front-end Implementation

Data-set Management

Biometric Scanning Integration

Performance Assessment

Conclusions



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.

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!!