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

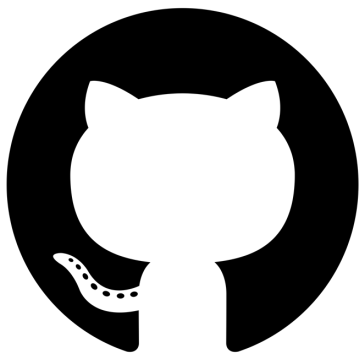
SAPIENZA
UNIVERSITÀ DI ROMA

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.

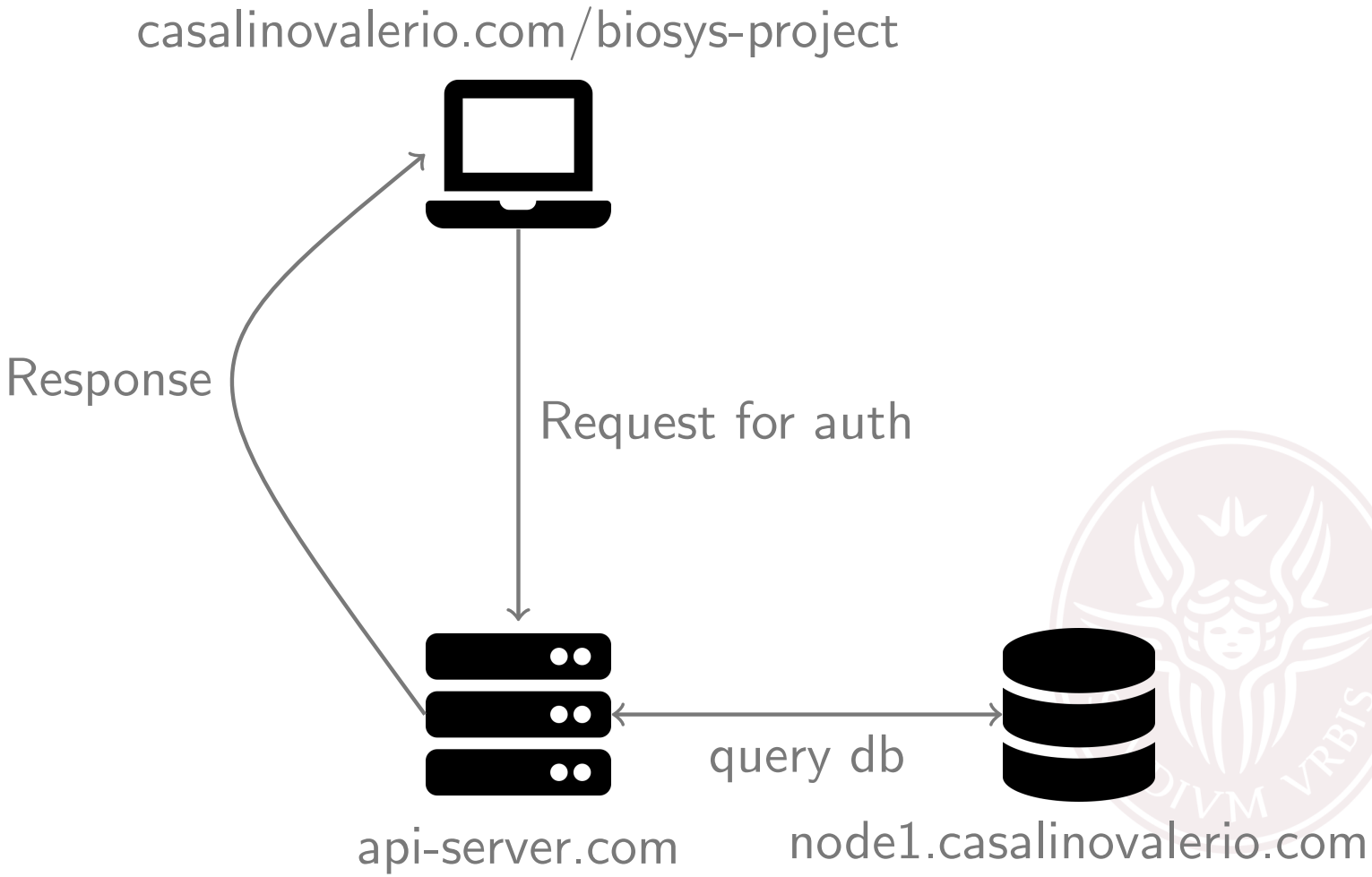
Overview

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

- A **web interface**[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]



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

Table of Contents

SAPIENZA
UNIVERSITÀ DI ROMA

The web Application

Some words on how it is done. opencv, javascript etc...

Does it work?

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

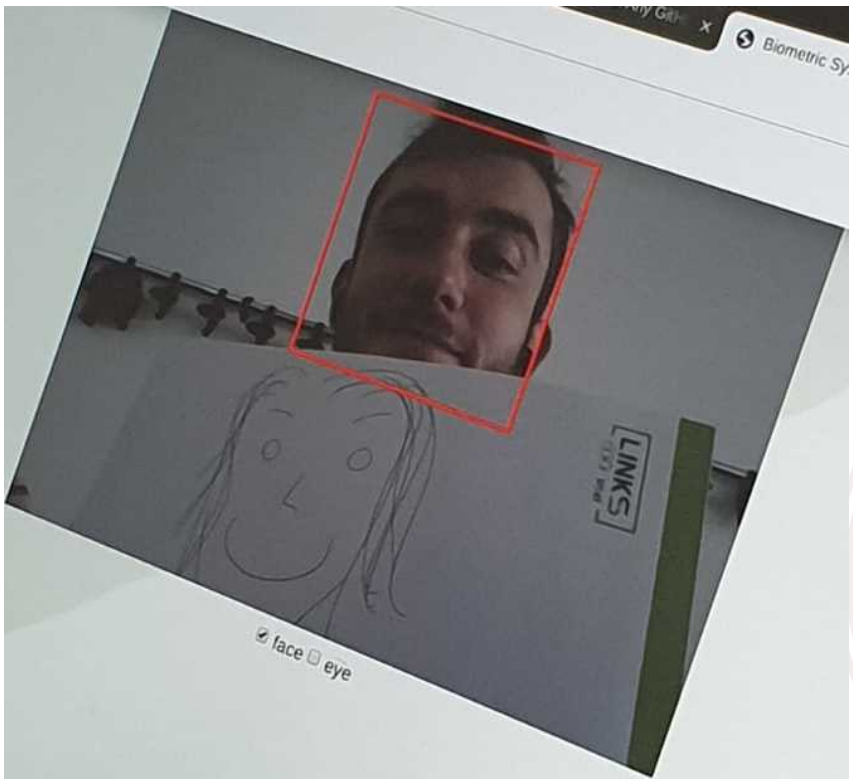


Table of Contents

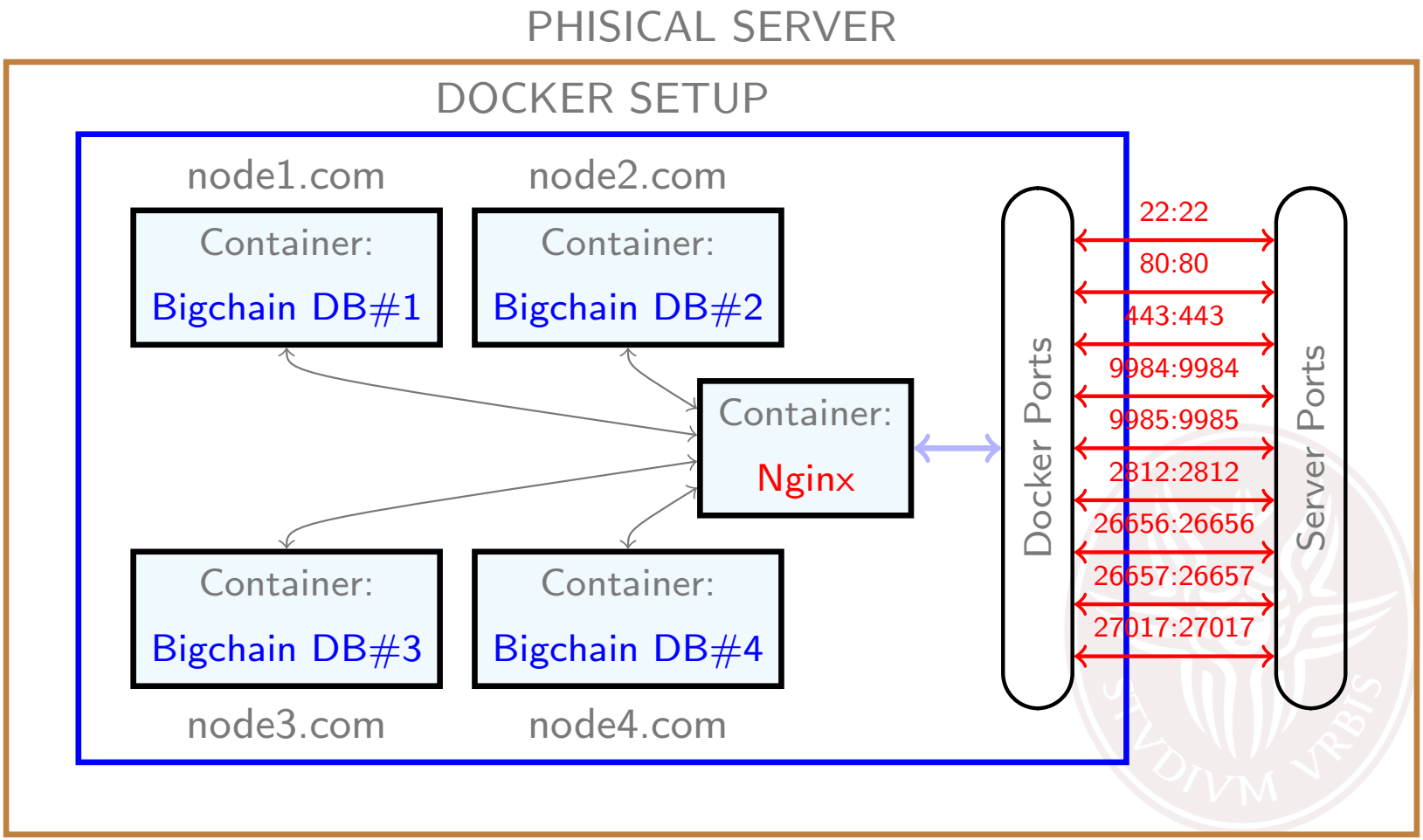
SAPIENZA
UNIVERSITÀ DI ROMA

The Block-chain database

As database for new faces, we implemented a **Block-chain**.
We used an open-source implementation of it, called BigchainDB[6].
We also used Docker[7] to deploy 4 containers running the application.

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

Architecture Implementation[8]



[8] 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

This is how we did it:

Connecting the API to the DB

This is how we did it:

Getting the response

This is how we did it:

Table of Contents

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 it[9], 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.

[9] Script here: `this-is.temp`

Testing approach

How it performed

Really well, it is a revolutionary project!

SAPIENZA
UNIVERSITÀ DI ROMA

Table of Contents

SAPIENZA
UNIVERSITÀ DI ROMA

Conclusions

Greetings...

Actual deployment considerations...

Performance considerations...

The Group

This is a great ending message from chilled-capibaras!

This is a real cool catchy phrase!!