TÉCNICO LISBOA

Health Care

NETWORK AND COMPUTER SECURITY

17 December 2021 | Campus Alameda - Group 26

THE TEAM:

Ana Albuquerque
ist1102209

André Proença
ist1102327

Joel Russo
Ist1102098

Privacy provision is a key issue for successful secure access to patients health information. Current approaches do not always provide patients with the ability to define suitable rules to access to their information in a secure way. This report presents an approach to give patients control over their information by means of external services. In this way, health information management and access control are kept independent and more secure.

1. Problem

A patient's medical records are extremely sensitive data. The historical records made by doctors facilitate the process of diagnosing a patient, ensuring their quality, which helps clinical staff to treat quickly and accordingly.
This data should be kept private, allowing only the discriminating staff to access it. We believe, that all healthcare facilities should have access to this type of information so that patients can receive healthcare anywhere and at any time.Therefore, the data should be protected from external agents (i.e., outside the medical institutions) and from unauthorised people within the institutions.

1.1. Solution Requirements

Client application requirements:

As a user (depending on my privilege) I should be able to...
- Read/write medical records (send requests to the system);
- Receive responses from the system (receive replies from the system).

As a Doctor and Nurse I should be able to...
- Read/update medical records.

As a Patient service assistants and Porters I should be able to..
- Read patients specific information in the medical records.

As a Ward clerk I should be able to..
- Register a patient in the system.

As a Patient I should be able to..
- Read my medical record.

As the system administrator I should be able to…
- Read/update medical records.

Security requirements:

As a system, I must…
- Ensure confidentiality and integrity of medical records;
- Ensure confidentiality and integrity of communications with the web application;

- Ensure that only authorised medical staff and patients have an account;
- Ensure different “roles” have access to different privileges;
- Ensure there is only one account per citizen;
- Prevent access to medical records if the user does not have privileges;
- Allow user A to change the privileges of user B, if user A is a system administrator;
- Ensure successful authentication of users;
- Mitigate brute force attacks on the authentication system (e.g. blocking IP’s);
- Minimize the impact of failures within the system (solution: do an "I'm alive with timeout" to the backup server);
- Minimize the impact of attacks inside the system.

As a user,

- I cannot repudiate my actions.

1.2. Trust Assumptions

Fully trusted:	Partially trusted:	Partially trusted:
- Hospital Server - Partner Lab server	- Certified user machine	- Not Certified user machine

2. Proposed solution

In order to simulate real systems and their interconnection, our solution is based on the development of two systems representing healthcare institutions. A hospital and a partner laboratory. The goal is to simulate two institutions sending confidential medical records of patients in a secure way from one side to the other. Authorised hospital and laboratory staff will be able to access their respective hospital and laboratory remotely or locally. Patients will also have remote or local access to the system in order to consult their medical records.

2.1. Overview

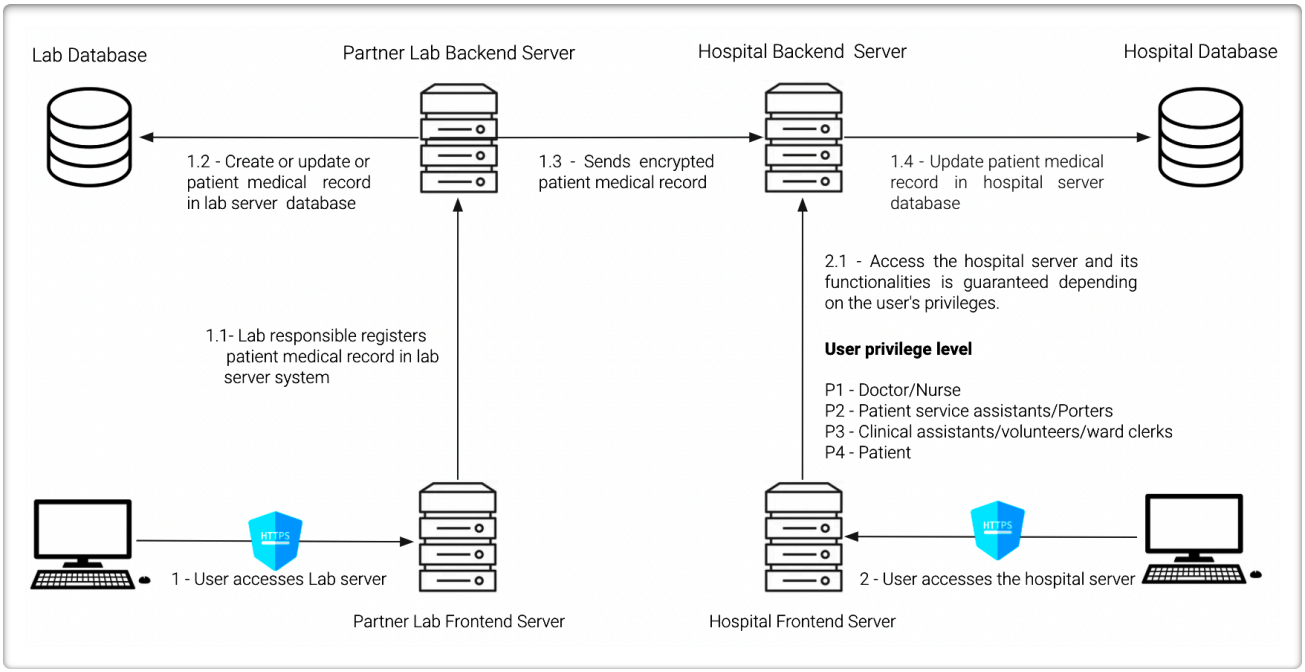

Figure 1. General Architecture.

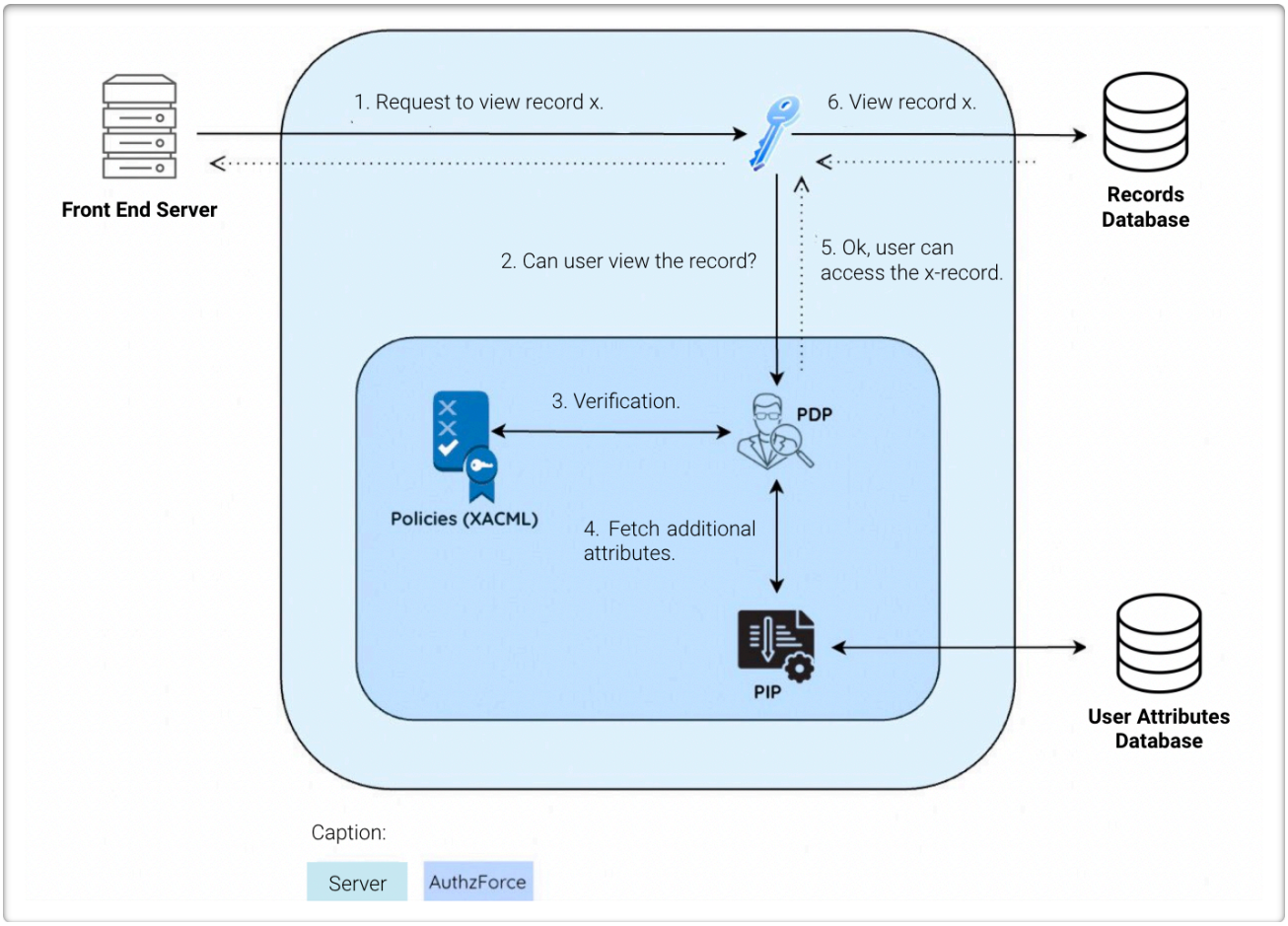

Figure 2: Happy Path - Record Access Control Architecture.

In the AuthzForce area, the service analyses the request and then generates an XACML authorisation request, which is "fed" into the AuthzForce PDP Engine. Then the PDP evaluates the request according to the policies it is configured with, and, if necessary, also retrieves other attribute values from the database in order to execute its decision (PIP). This decision is then sent back from the engine, and depending on the result, the server generates a response and sends it to the client application

2.2. Deployment

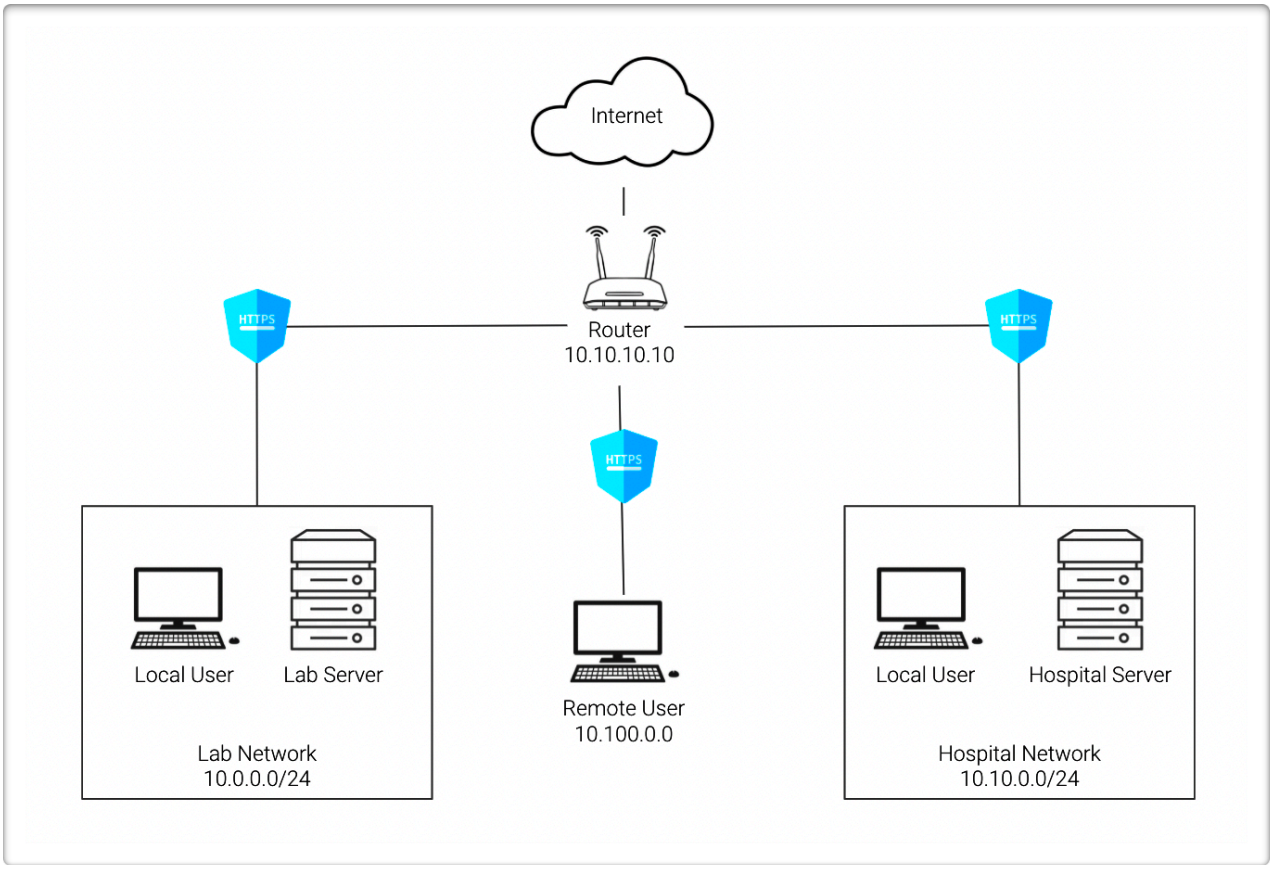

Figure 3: Network Architecture.

As you can see, 4 (four) virtual machines will be deployed.

- 1 for the router that will simulate the internet;
- 1 for the hospital, where the frontend, backend and database servers will run;
- 1 for the partner laboratory, where the frontend, backend and database servers will run;
- 1 for local or remote users which will access the frontend servers. To change the location of users on the network, just change the properties of the VM in the hypervisor (VirtualBox, for example). This way we avoid creating multiple VM's for each local and remote user.

2.3. Secure channel(s) to configure

Who will be communicating?

- The health institutions (Hospital and Laboratory) via secure TLS communication, to send medical records securely.
- The remote and local users with the health institutions, (Hospital and Laboratory) via TLS secure communication, to perform a certain action. (For example, consulting medical records, etc...)

Which keys will exist and how will they be distributed?

- Each Health Institution will be a certifying entity that issues its own certificate. Each of these institutions will also have an asymmetric key pair, which will be used to communicate and establish secure connections. The keys and certificates are stored in their respective health institutions virtual machines.
- The virtual machine, where users will access the Web Servers, will also have an asymmetric key pair and a certificate.

2.4. Secure protocol(s) to develop

Who will be communicating?

- The health institutions (Hospital and Laboratory) via this secure communication, to send medical records securely.

Security properties to ensure:

- The Protocol must ensure integrity, confidentiality and non-repudiation.

3. Considered technologies

- Frontend: Angular, Java, CSS, JS
- Backend: NodeJS, Java
- Database: SQL, MongoDB
- Security policy language: XACML

4. Plan

TO DO

4.1. Milestones

Basic Version:

- Design system, set up the login method in Health Institutions frontend servers, set up and configure AuthzForce PDP engine.

Intermediate Version:

- Ensure secure communication via TLS, Use asymmetric cryptography for communication establishment, custom protocol development, encrypt server databases

Advanced Version:

- Backup servers, database replication, fault tolerance.

4.2. Effort commitments

TO DO

5. References

AuthzForce (Community Edition)