GO.DATA ANONYMIZATION

Krunal Badsiwal
EETAC - UPC

01

ANONYMIZATION

Private and sensitive data are not visible in Go.Data service.

02

PROTECTION

Private and sensitive data owned by an institution cannot be seen by other institutions, only by itself.

03

SEAMLESS MERGE

Although cases are anonymized, if the same patient are added later by the same or other institution, the system knows that these cases correspond to the same patient and can be seamlessly merged.

04

DECODE

The right to visualize sensitive data owned by an institution, can be transferred to other institutions keeping all the security mechanisms.

05

SECURITY

Although Go.Data server is compromised no sensitive data can be seen, as licenses are unique for each institution and stored in a different server.

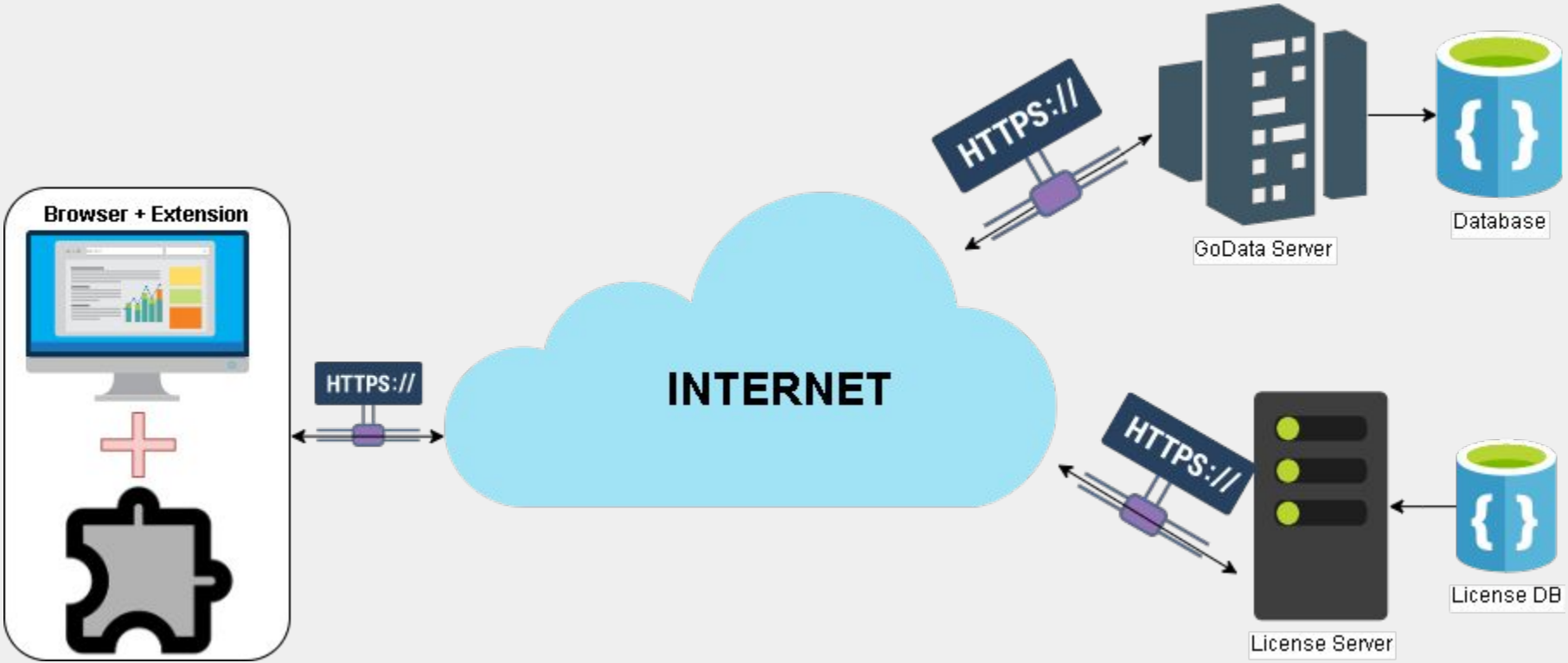
HASH ID

Each patient or case has a HashID that:
Identifies the patient inside the system
The HashId is generated from unambiguous data from the user, typically birth date and some unique ID for the user, i.e driver license, national security number or CIP in Catalonia.
From the hashID, it is NOT possible to recover original data, that it is both sensible and not anonymous.

ARCHITECTURE I

01	License server based on Digital Rights Management (DRM) techniques.	• Protected Licenses • Secure sharing
02	Using Public/Private Key cryptography for license creation.	• Digital signatures • No key exchange required • Strong security provided
03	Using symmetric cryptography for case encryption.	• Secret key Encryption/Decryption • Fastest cryptography

ARCHITECTURE



ARCHITECTURE III

1. The **user** edits and views cases and contacts through the **browser**.
2. This data is stored inside the Go.Data Server.
3. The **License Server** is connected to the Go.Data API and encrypts all sensitive data as cases are added. Licenses are generated for the institution that owns this sensible data.
    a. This license is unique to each case and institution pair.
    b. Licenses are stored out of GoData in the license server minimizing risks.
    c. Specific license for each case is needed to be able to decrypt sensible data.
    d. Licenses for an institution cannot be used by other institutions because they require private key of the owner institution.
    e. Original institution can generate additional licenses to other institutions for their owned cases.
4. Go.Data extension installed in the browser, decrypts the case and shows the sensitive data.

LICENSE SERVER

The license server provides 2 functions:

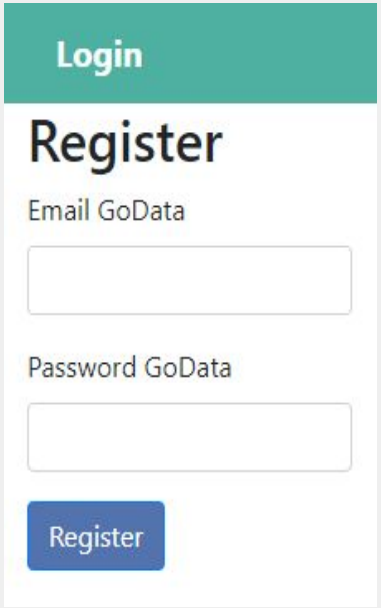
- Generates, stores and manages licenses.
- Auto Encrypts new added cases when they are added.

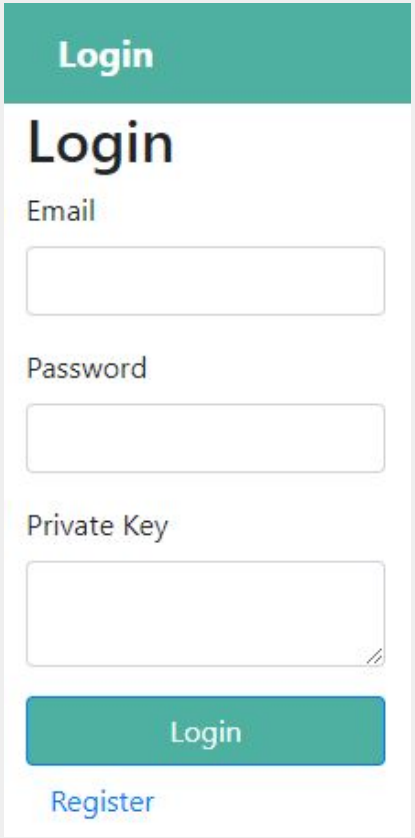
This second function can be separated for increased security.

USE CASES

1. **User level:**
    a. What the user see in the browser...
2. **Technical level:**
    a. What is happening between browser, godata, license server and extension.
    b. i.e communications
    c. what happens with the keys inside license, how are encrypted, what public/private keys are used.

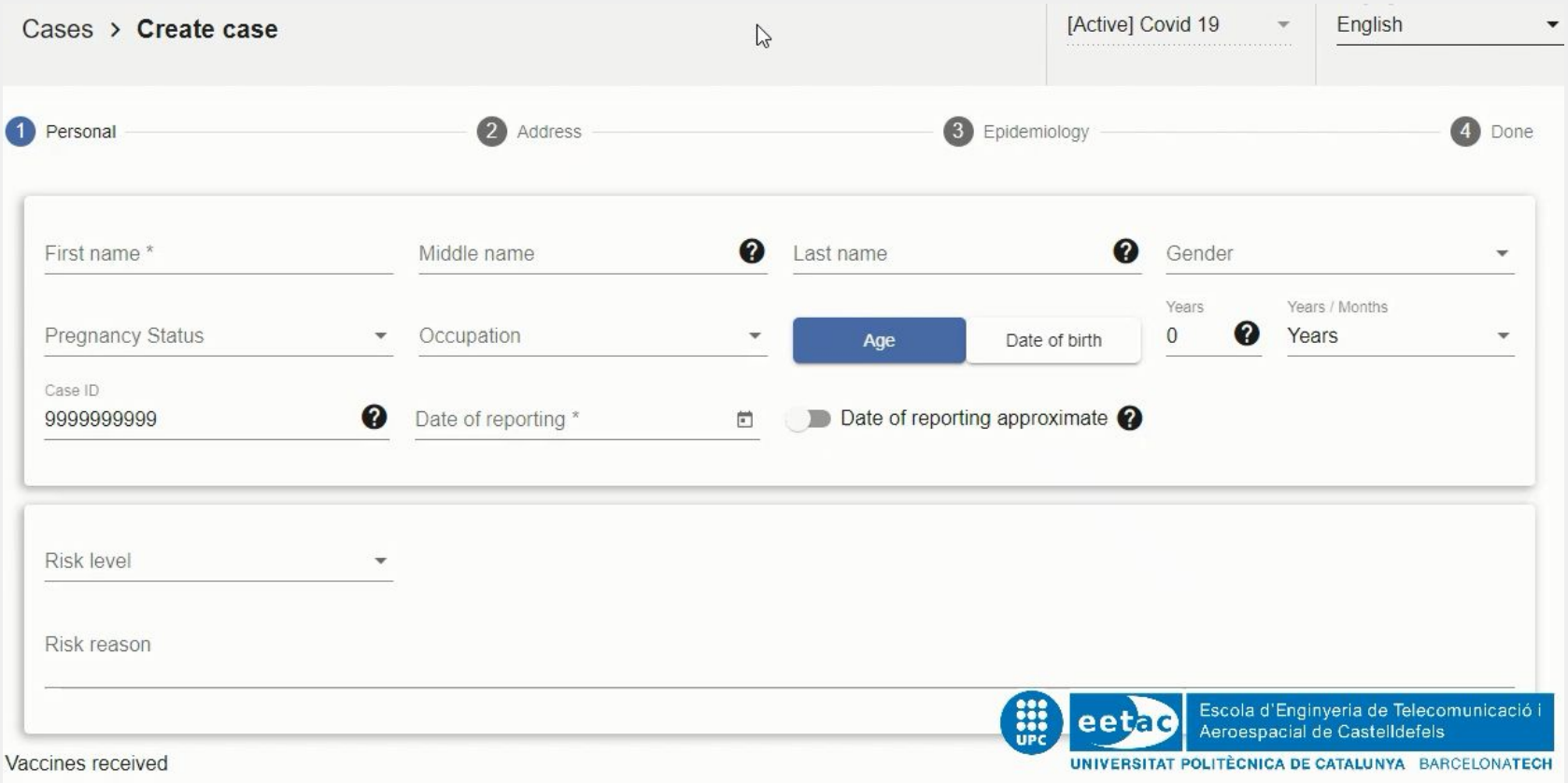
USE CASE 0. AUTHENTICATION IN THE LICENSE SERVER



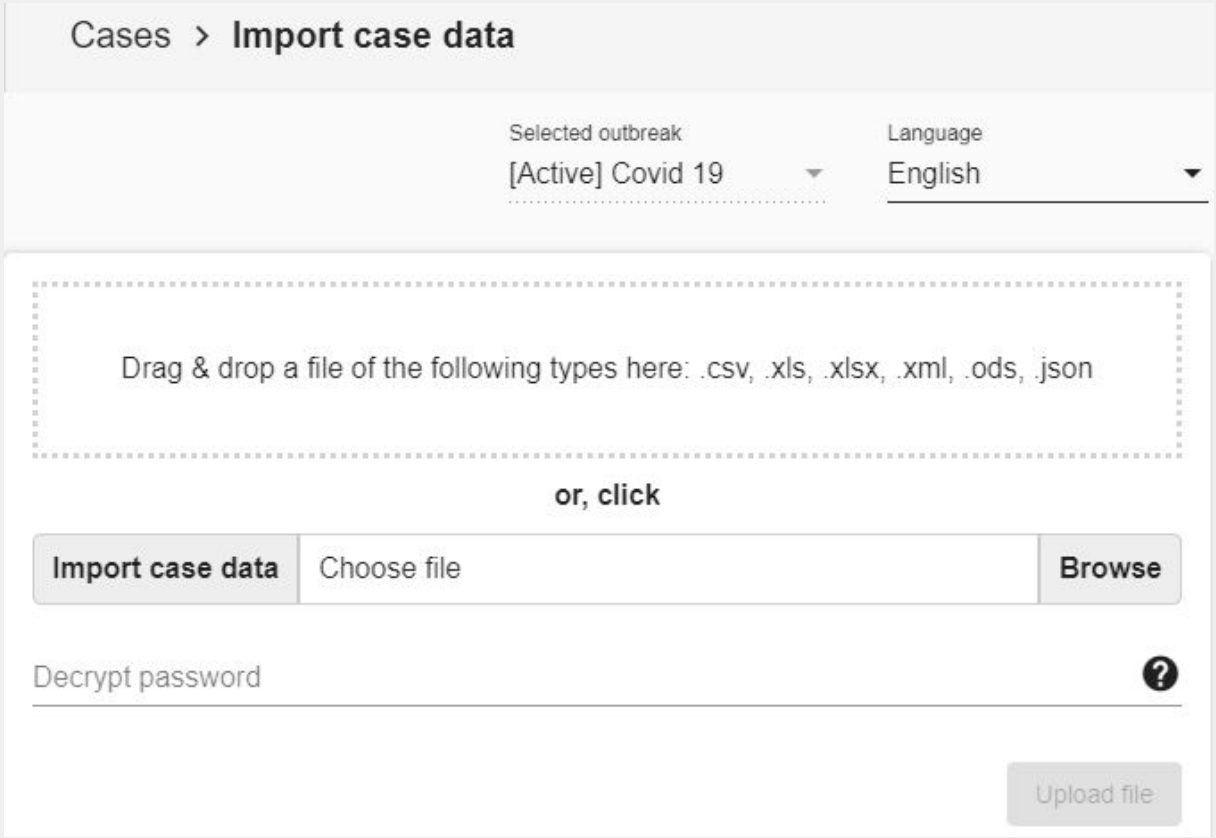


- User/Hospital email address
- Password
- Private Key
- Login Button

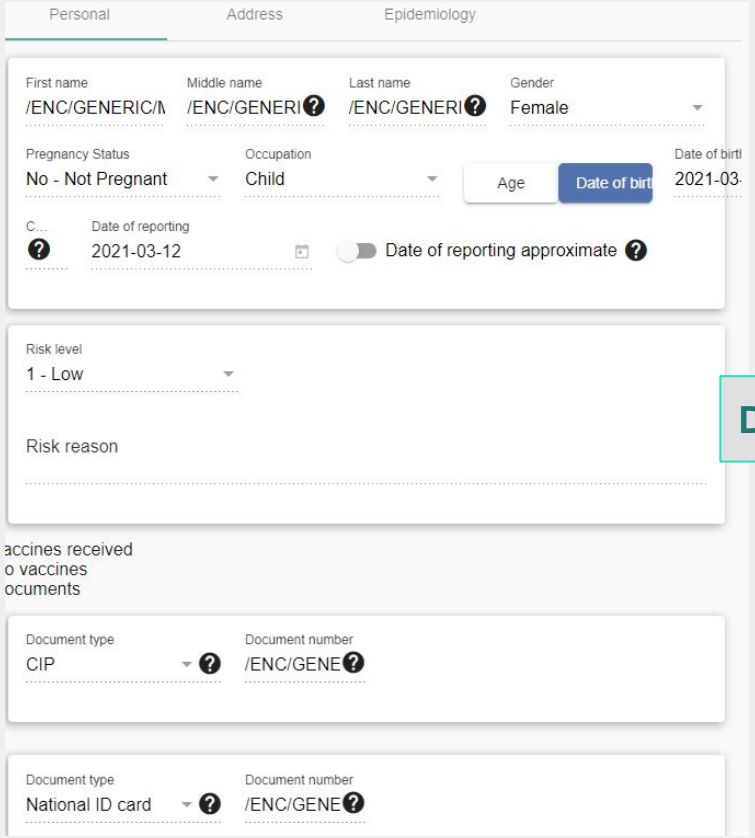
USE CASE I. MANUAL DATA ENTRY IN GODATA



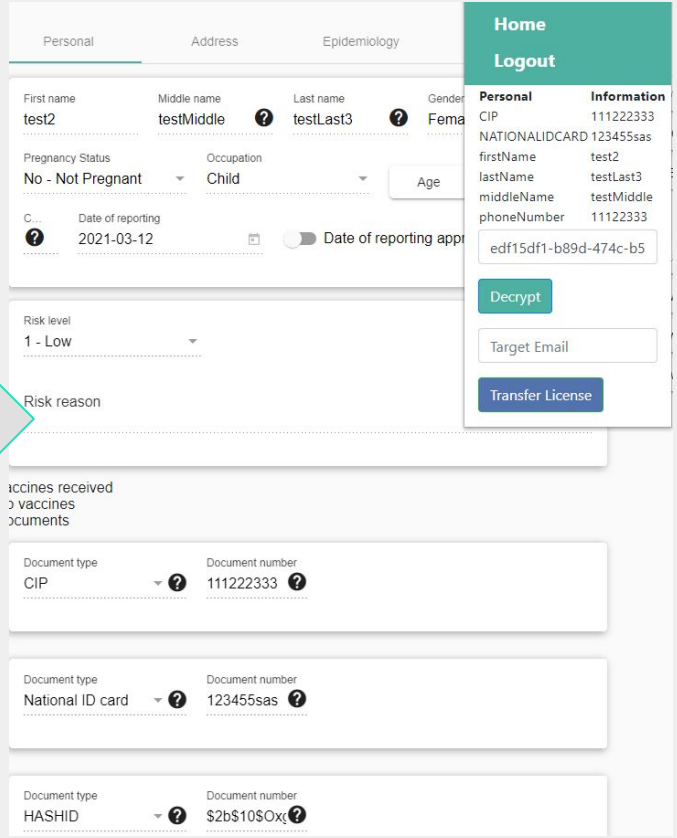
USE CASE 2. BULK DATA ENTRY IN GODATA



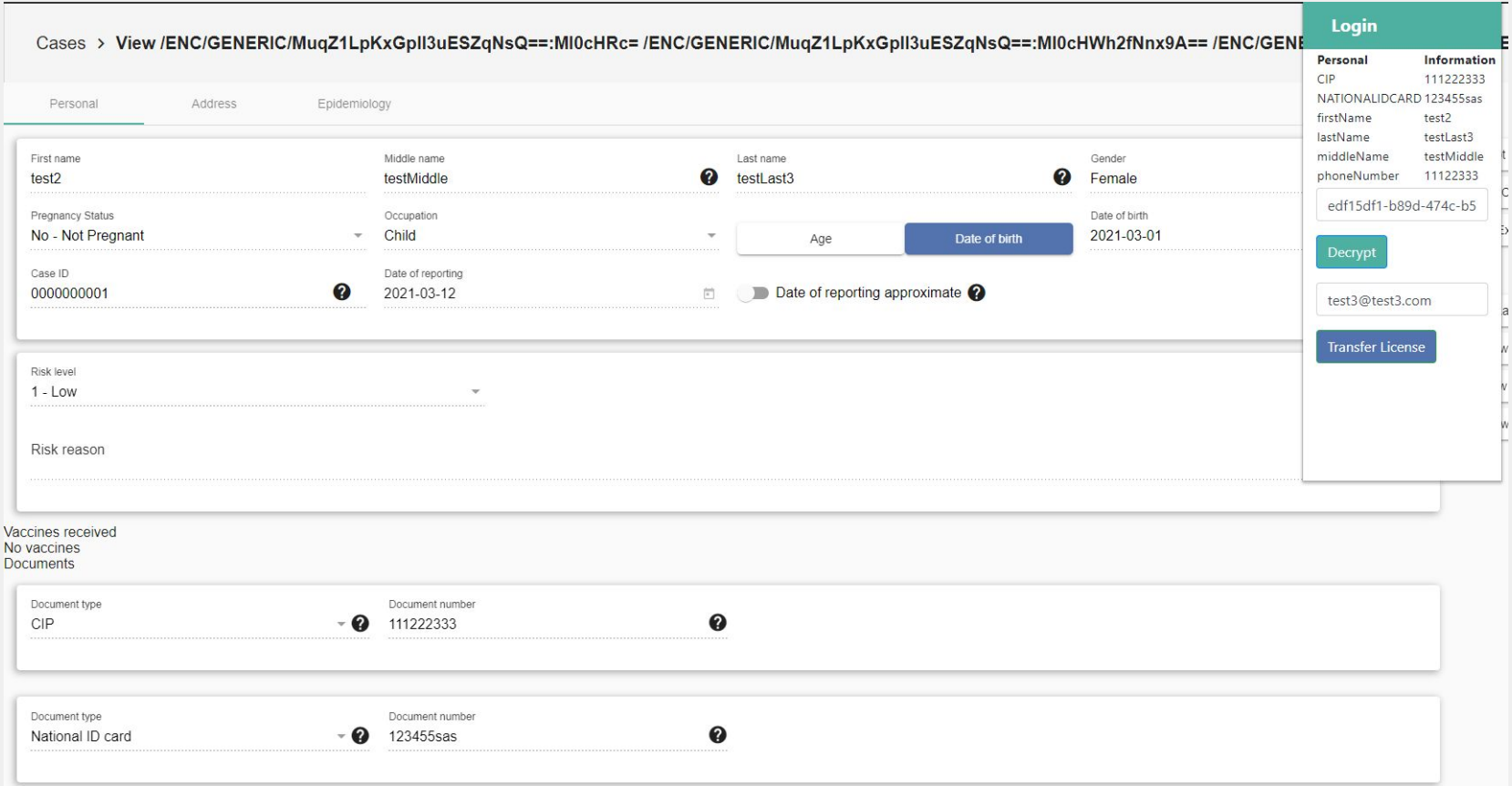
USE CASE 3. DECRYPTION OF PRIVATE DATA



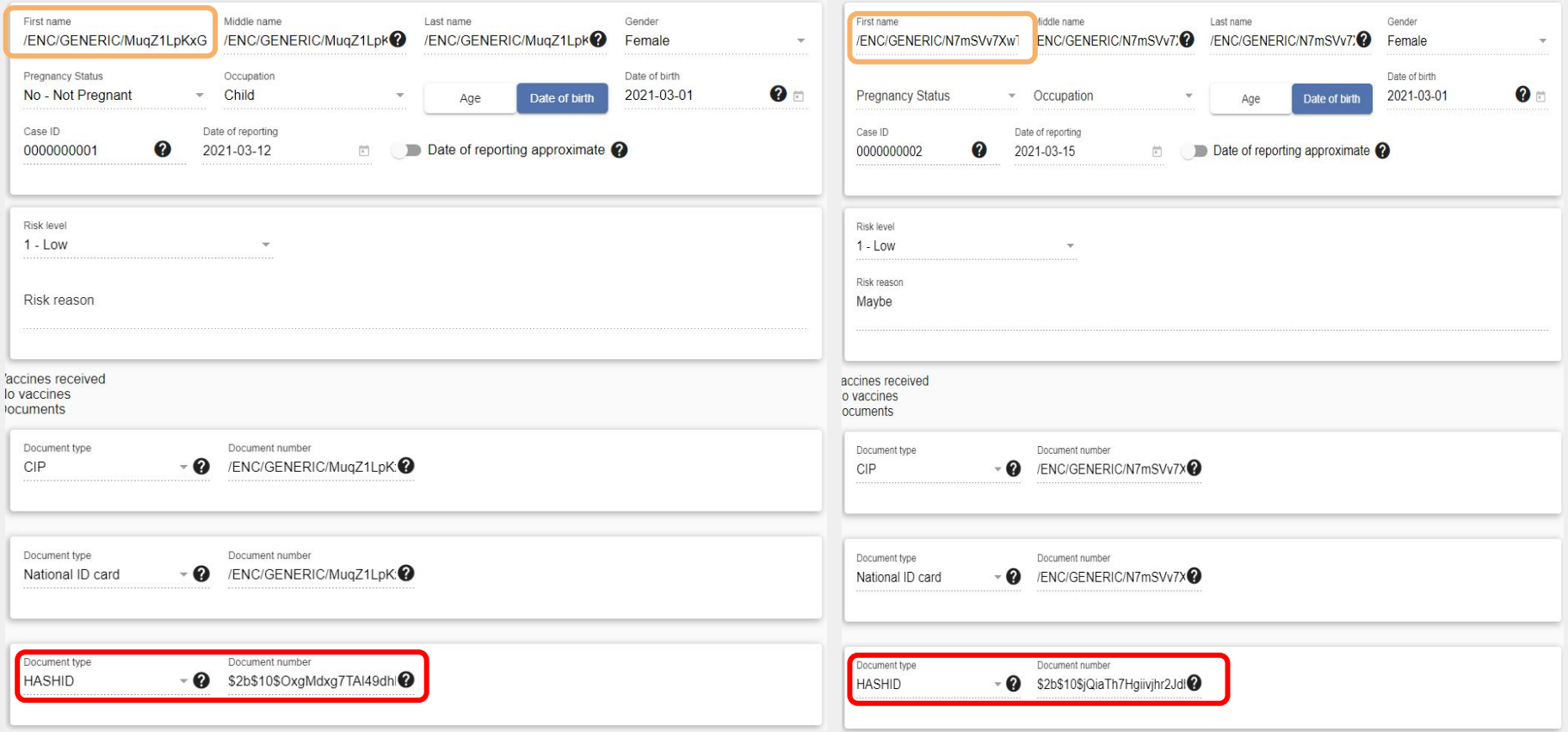
Decryption



USE CASE 4. TRANSFER OF PRIVATE DATA TO OTHER INSTITUTION



USE CASE 5. MERGE OF TWO CASES OF THE SAME PATIENT



NEXT STEPS

1. Integration of the extension directly into godata except the DRM License Server?

2. Do it have sense to maintain the licenses out of Go.Data in other physical server?

THANKS

Does anyone have any questions?
EETAC

- **Krunal Badsiwal**

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik** and illustrations by **Stories**

Please keep this slide for attribution.