w.zanen@eurotransplant.org

Abstract

Functional and technical specifications for the EDITH deceased patient registry

Specifications

EDITH deceased patient registry

Specifications

EDITH deceased donation transplant registry

This document will describe all components used and how they are used for the EDITH deceased patient registry (<https://edith-project.eu/>). It is not intended to be a manual for use of any of these components, those will be provided when the application is delivered.

Table of Contents

[Introduction 3](#_Toc21011947)

[Open platform 3](#_Toc21011948)

[openEHR 3](#_Toc21011949)

[Flexibility 3](#_Toc21011950)

[Resources 4](#_Toc21011951)

[Components 5](#_Toc21011952)

[Flow between components 6](#_Toc21011953)

[AWS cloud configuration 7](#_Toc21011954)

[openEHR archetypes and templates 9](#_Toc21011955)

[User management 11](#_Toc21011956)

[openEHR platform (Better) 12](#_Toc21011957)

[ETL 13](#_Toc21011958)

[Terminology adapter 14](#_Toc21011959)

[Form builder 15](#_Toc21011960)

[Pathfinder 16](#_Toc21011961)

[Statistics (Metabase) 17](#_Toc21011962)

# Introduction

## Open platform

The EDITH deceased donation transplant registry is based on an open platform: openEHR. Open platforms are open in the sense that data can be are available is a standard format, based on open clinical models and can be wholly and freely accessed. Data and models in an openEHR based platform can be used with any vendors product that adhere to these standards. These qualities make an open platform very flexible. More on open platform in this excellent white paper by the Apperta Foundation: <https://apperta.org/openplatforms/>.

## openEHR

'openEHR' is the name of a technology for e-health, consisting of open specifications, clinical models and software that can be used to create standards, and build information and interoperability solutions for healthcare. The various artefacts of openEHR are produced by the openEHR community and managed by the openEHR Foundation, an international non-profit organisation established in the year 2003 (<https://www.openehr.org/about/what_is_openehr>).

## Flexibility

The openEHR platform gives the EDITH registry flexibility in setup. Besides gathering the data over a long period of time in a registry from existing registries, it is important to offer countries who do not have a registry a platform that they themselves can use for starting a registry.

The current setup is a single instance openEHR platform with a central database and a single instance front-end. However other setups are possible for instance we can keep the central database, but created a federated landscape of smaller openEHR based platforms and frontends that can deviate both on user interface, language and data collection from the central database as long as the EDITH dataset acrhetypes are incorporated in the country specific dataset. In this setup any country can create their own database based on their own specific needs and still be able to seamlessly deliver data to the central EDITH registry.

You can choose different vendors for the openEHR platform both “free” open source solutions (for instance <http://ethercis.org/>) or paid solutions. You can also choose to develop your own front-end, find an open source version (and extend it) or buy a solution from a vendor. We compared many solutions and came to the conclusion that an open source solution would require considerable effort to adapt to the needs of a registry, therefor it has a high initial cost, but perhaps lower running cost than a paid solution. However as the EDITH deceased donation transplant registry has limited funding and an unclear timeline after the project is concluded, we have opted for a paid solution.

The product we use is the Better Platform with Better Pathfinder Lite front-end application ([www.better.care](http://www.better.care)) it offers a lot of tools we use in our registry out of the Box. We run it ourselves in the AWS cloud in Frankfurt. At the moment Better has started to offer a SAAS solution (Azure cloud in France), which might be better fit in the future as no technical support for servers, netwerk is needed any more.

## Resources

Resources mentioned in this document can be find on our public github:

<https://github.com/edith-project/deceased-registry>

# Components

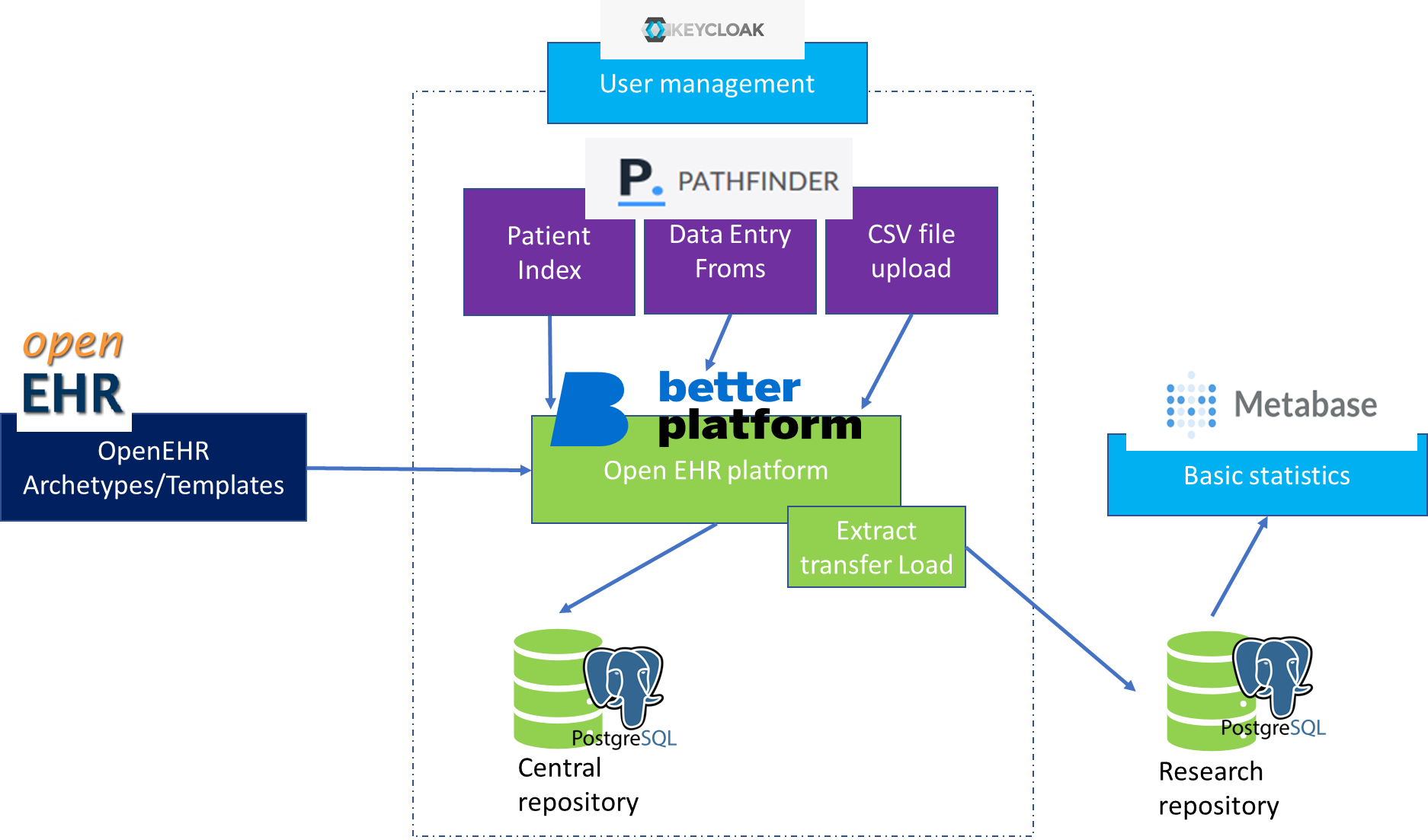
The registry uses several components to create , in this chapter we will describe them shortly and explain how the work together to provide the functionality as described in the requirements document.

Figure 1 Schematic representation of components and how the interact

openEHR: archetypes and templates  
Archteypes are the small clinical building blocks holding concepts like blood group or weight, templates describe a clinical use case. These templates can be uploaded to the openEHR platform (Better platform) which will automatically generate everything that is needed to store and access data based on these templates through open EHR API’s (documentation <https://specifications.openehr.org/releases/ITS-REST/latest/index.html>)

Keycloak: user management  
Keycloak is used to manage access to Pathfinder and Better Platform, user can use a single password to access these application. Metabase uses it own user access.

Pathfinder  
Pathfinder is the front-end application that end-users (national competent authorities) will use to enter data in an initial and a follow-up data entry form. A csv file upload is available to upload data in bulk.

Better platform  
The better platform provides a clinical data repository based on openEHR, specifications, in addition it provides a form builder, ETL (Extract transfer load), EHR explorer tool voor admins to quickly manage the platform.

Metabase  
Metabase is a basic statics application, it can be used to quickly generate overviews on the data. The registry will provide data export possibilities for the National Competent Authorities (NCA).

## Flow between components

Short summary of a typical workflow:

1. An engineer creates user accounts in keycloak
2. A Clinical modeler (or other employee at the EU registry) creates or updates openEHR template
3. Templates are uploaded to be better platform
4. Based on the template a Form is created or updated by the Clinical modeler
5. The form is tagged for pathfinder so it immediately becomes available in pathfinder
6. Data is collected by NCA’s and entered in the forms or uploaded via the bulk upload.
7. The clinical modeler designs a data query (AQL) in EHR explorer to extract data needed for research.
8. ETL is configured to use the query to extract data and the extraction is scheduled.
9. The user of an NCA can extract the data from Metabase.

# AWS cloud configuration

In AWS the following components are used:

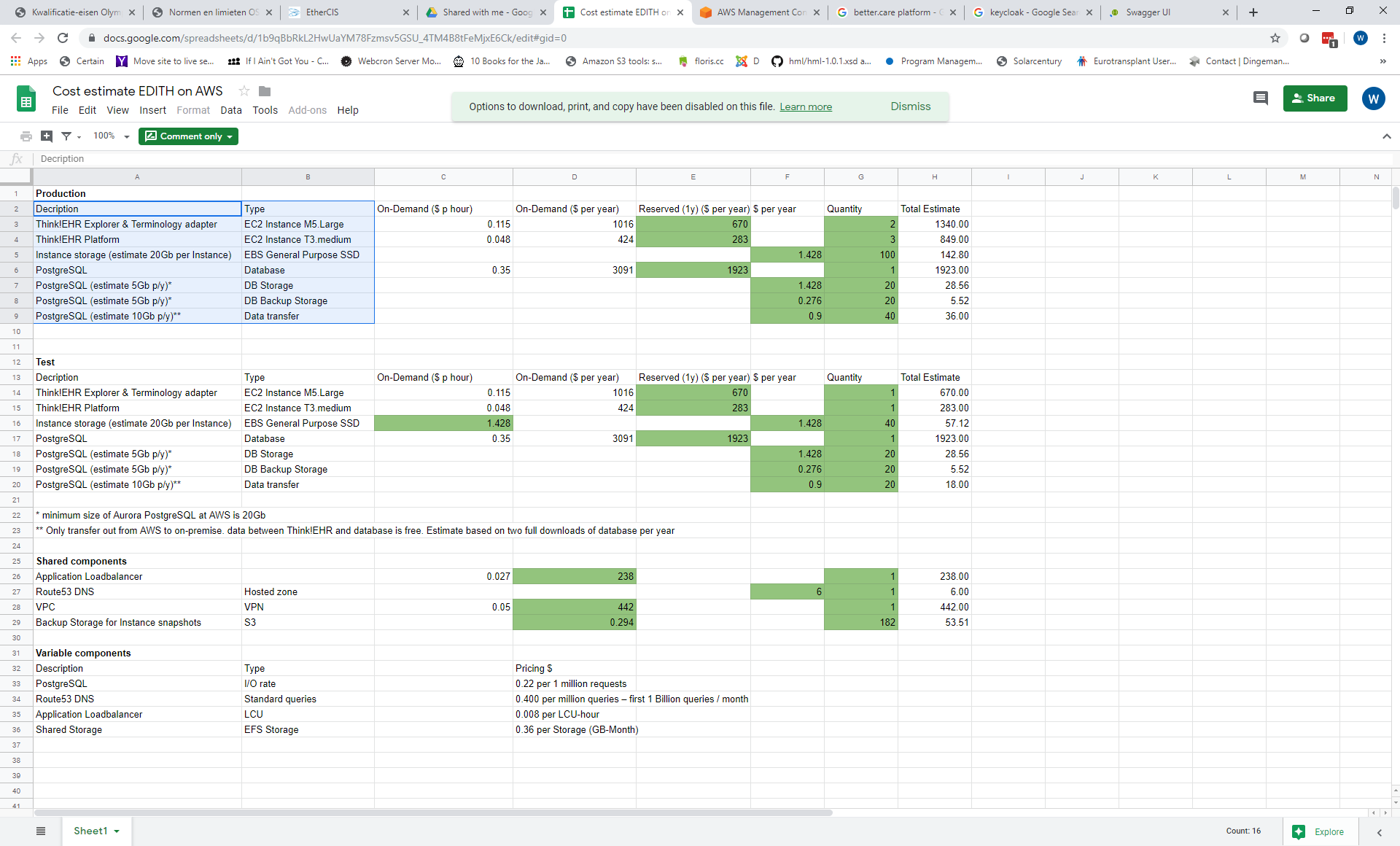


Figure 2 AWS components used per environment (test, prod)

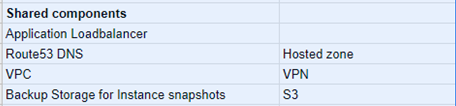
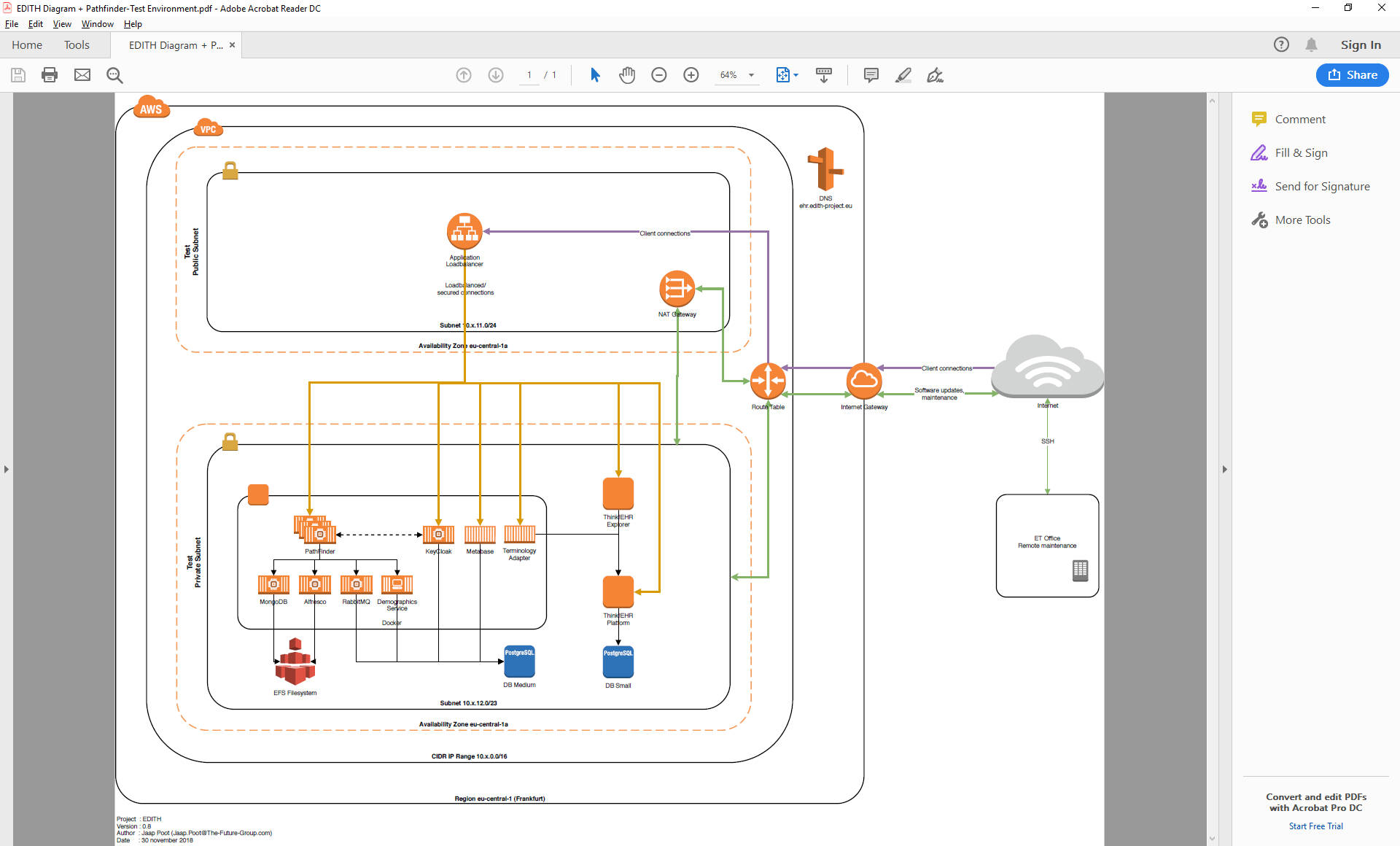


Figure 3 Components shared between environments

On the next page you can find a schema of the implementation used for a single environment (in this case Test). The environment has its own VPC (Virtual Private Cloud) and a public and private subnet to secure the data in the private subnet. An application load balancer is used to balance client traffic, this will make the setup more flexibility as we could scale easily to extra instances of pathfinder or Better platform (formerly known as Think!EHR platform). The Pathfinder and Metabase components run on the docker instances (easier to install) while the Better platform is not yet available on docker and needs a separate server.

Both the docker and platform servers use a Postgress database. Databases are back-upped daily and are retained for 3 days.



# openEHR archetypes and templates

**Description**

We use openEHR templates and arcehtypes to describe our dataset and create operational templates that can be used inside an openEHR platform.

**Use**

We have used Better’s ADL-designer to create the openEHR archetypes and templates. Most archetypes are downloaded from the international open Clinical Knowledge Manager at openEHR.org (<https://www.openehr.org/ckm/>). You can download Archetype and Templates files used in the project from our public Github repository.

**Configuration**

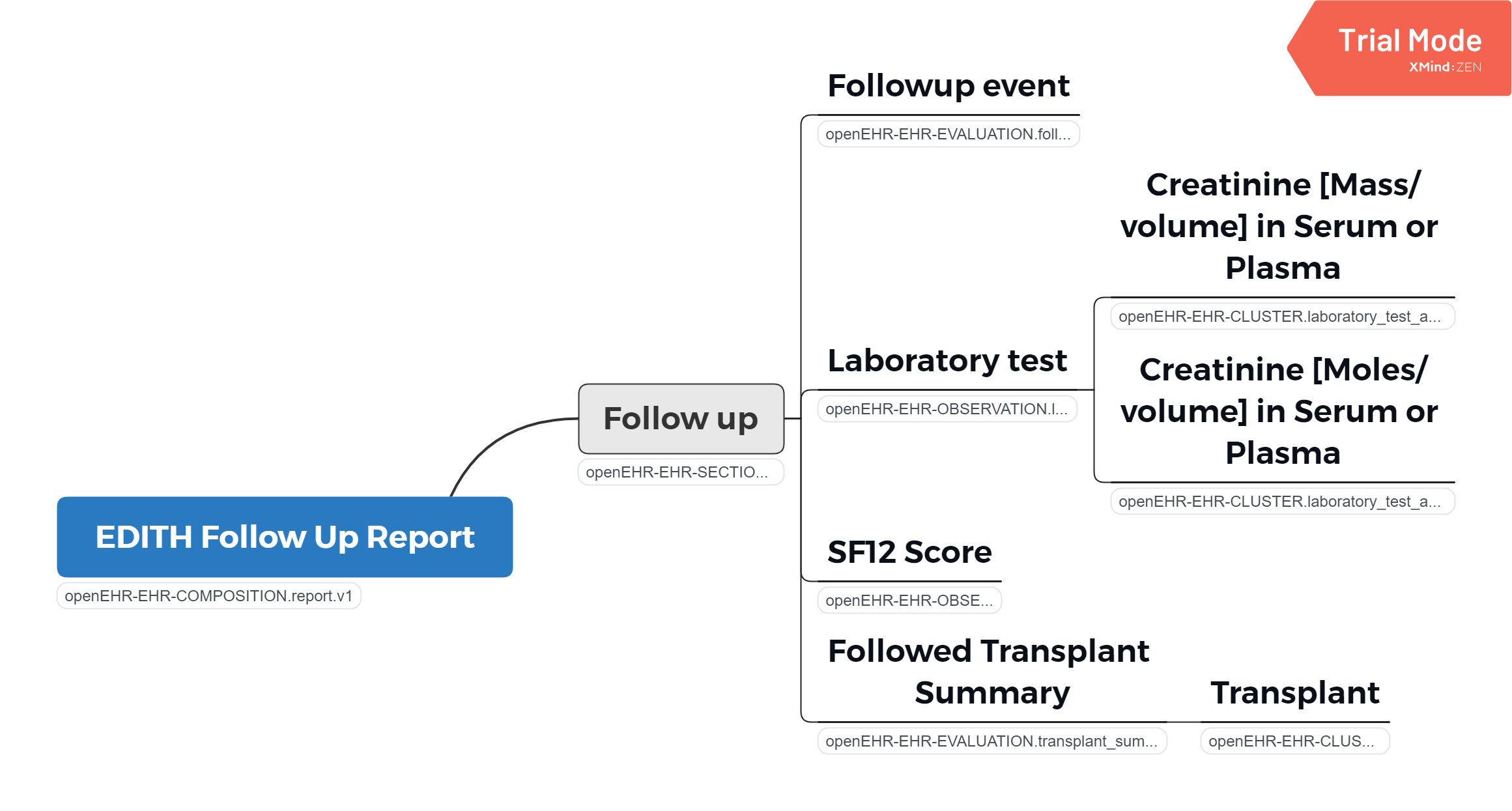
We have created 2 templates which intern use archetypes (both local and CKM)

Figure 4 EDITH Follow Up archetypes mind map

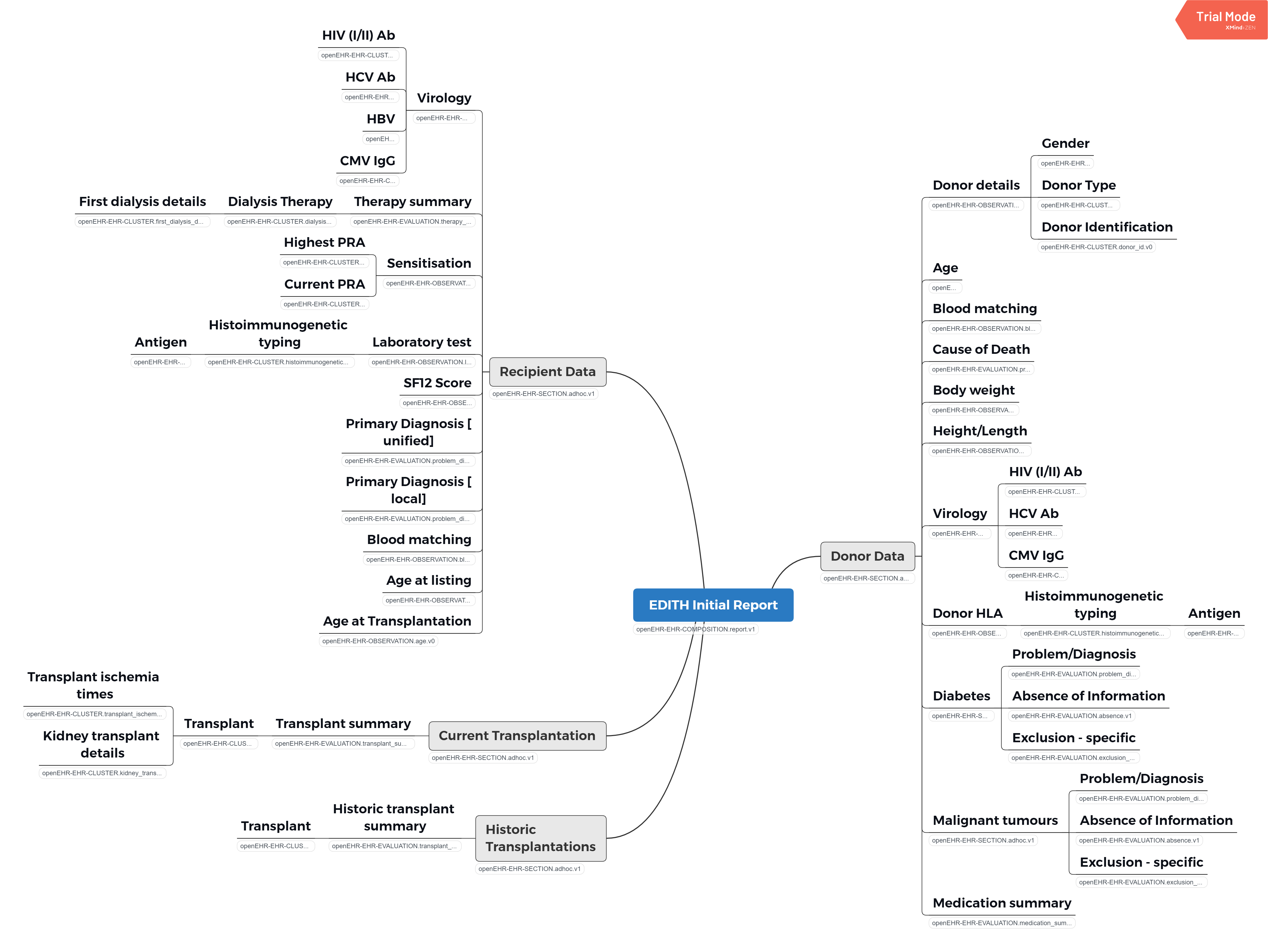
1. EDITH Initial, describes an initial follow-up record to be recorded as the first record after transplantations. Its fields contains information on donor, recipient and transplant for a kidney transplant

Figure 5 EDITH Initial archetypes mind map

1. EDITH Follow Up, describes information on the patients gathered by phycians on regular intervals. Also used to record death of a patient and failure of a transplanted graft.

# User management

**Description**

**For user management KeyCloak is used for authorizing users to the Better and Pathfinder applications. For Metabase the internal .user management is used.**

**Use**

KeyCloak is configured by the Technical admin of the platform

**Configuration**

We have three user levels:

1. NCA users (users that can enter and access data on country level), these users can access data including the local subject ID (patient ID that was defined by the NCA). NCA’s can download statistics from metabase with patient ID for there own country, but without ID for all transplants.
2. Europe functional admins (can access patients for all countries)
3. Admins can configure and setup components described in this document.

# openEHR platform (Better)

**Description**

Within this project we are using Better platform (formerly known as Think!EHR) version 2.4.3.

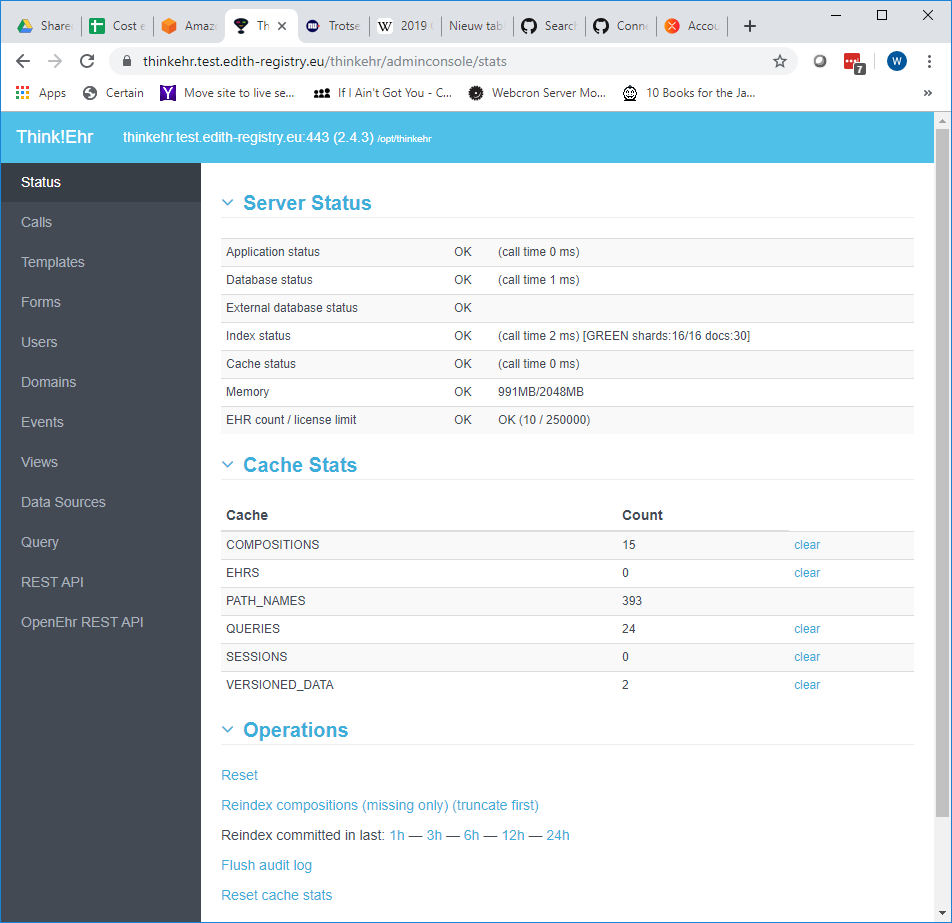
**Use**

Figure 6 Admin console Better platform

The platform can be configured by Admins

The following artefacts are uploaded to run the application:

* Templates (see openEHR archetypes and templates)
* Forms (see Form builder)
* Views; a view is created based on a AQL query (the openEHR query language) to create the patient list in the pathfinder application.

**Configuration**

The Better platform runs on a dedicated Linux server with a dedicated Postgress database.

It is a single node instance (but this platform can be clustered if needed)

It contains user account information for admins. Also a service account was created for the ETL user with only read access to the platform.

## ETL

**Description**

The Extract Transfer Load (ETL) tool from the Better platform is a tool to quickly access the data from a Better platform via an AQL query and transfer data to an relational database. As most statistics tools cannot access openEHR platforms directly it is an easy way to get data in to a database that the statistics application can use

**Use**

The platform can be configured by Admins. Connections are defined one to the openEHR instance and one connections to the Europe schema research database. And one for every country schema .

For each country a query is defined to retrieve country specific data from the openEHR, platform as well as European query (which excludes the Subject ID).

For every query a mapping is generated to a relational database schema.

The ETL processes are scheduled to run every hour, so the statistics database is maximum 1 hour behind on the openEHR database. It does a full refresh.

**Configuration**

The AQL queries can be found within the Github repository.

## Terminology adapter

**Description**

The terminology adapter contains all terminologies used within the archetypes. Examples are the HLA nomenclature, ICD-10 etc.

**Use**

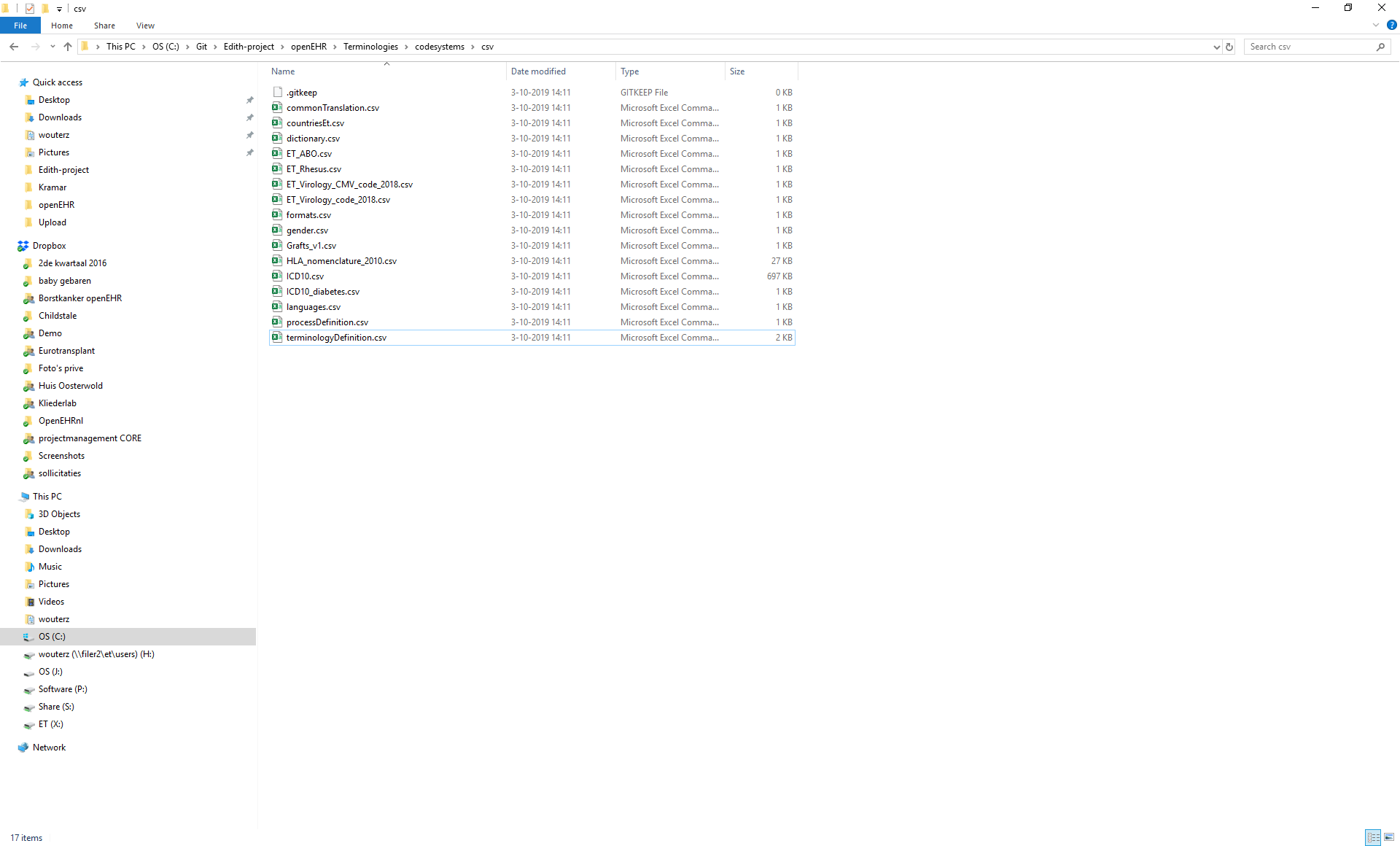
The platform can be configured by Admins. Connections are defined one to the openEHR instance and one connections to the Europe schema research database. And one for every country schema .

A nomenclature is defined within a CSV file with at least the following columns:

* code, the code for a row within the Nomenclature
* descripition, a description that can be displayed for a specific code
* parent, optional parent of the code if the code has a hierarchy

**Configuration**

The following terminology are defined (full csv files can be found in Github):



## 

## Form builder

**Description**

Form builder is a drag and drop tool to create forms based on openEHR templates and the better platform

**Use**

Admin creates or updates a form based on a template. Tags it for use within pathfinder. Uploads the form into your OpenEHR platform. The form will automatically be used in Pathfinder.

**Configuration**

Form builder runs on tools.marand.si. Find the form description files in our Github. Two forms are created:

* Edith Initial to capture the initial form
* Edith Follow Up to capture the Follow Up form

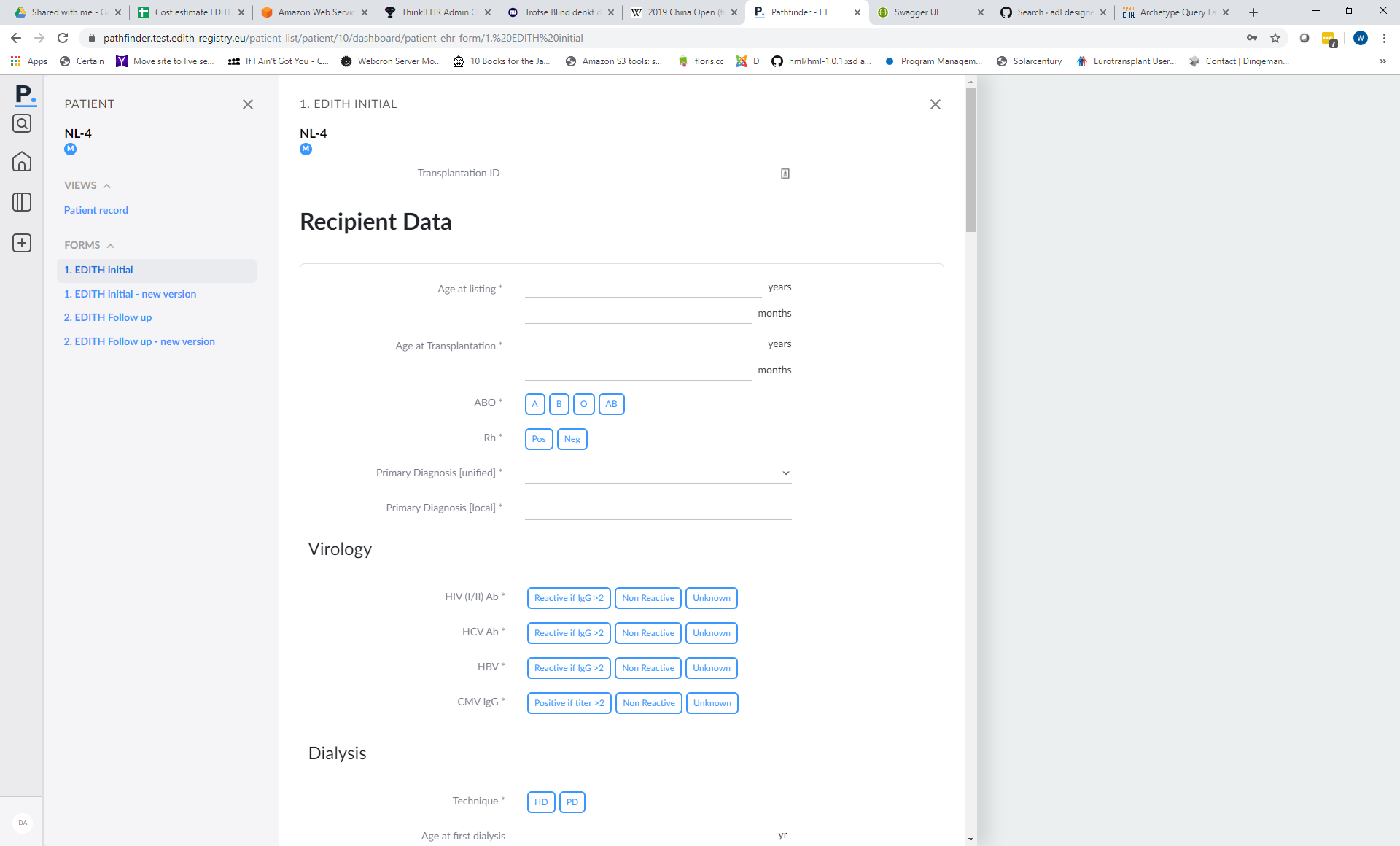


Figure 7 Part of the Edith Initial form

# Pathfinder

**Description**

Better Pathfinder is on openEHR front-end for quickly publishing forms based on openEHR so that endusers can use them. It is focused on data capturing and needs the Better openEHR platform to run on top off.

**Use**

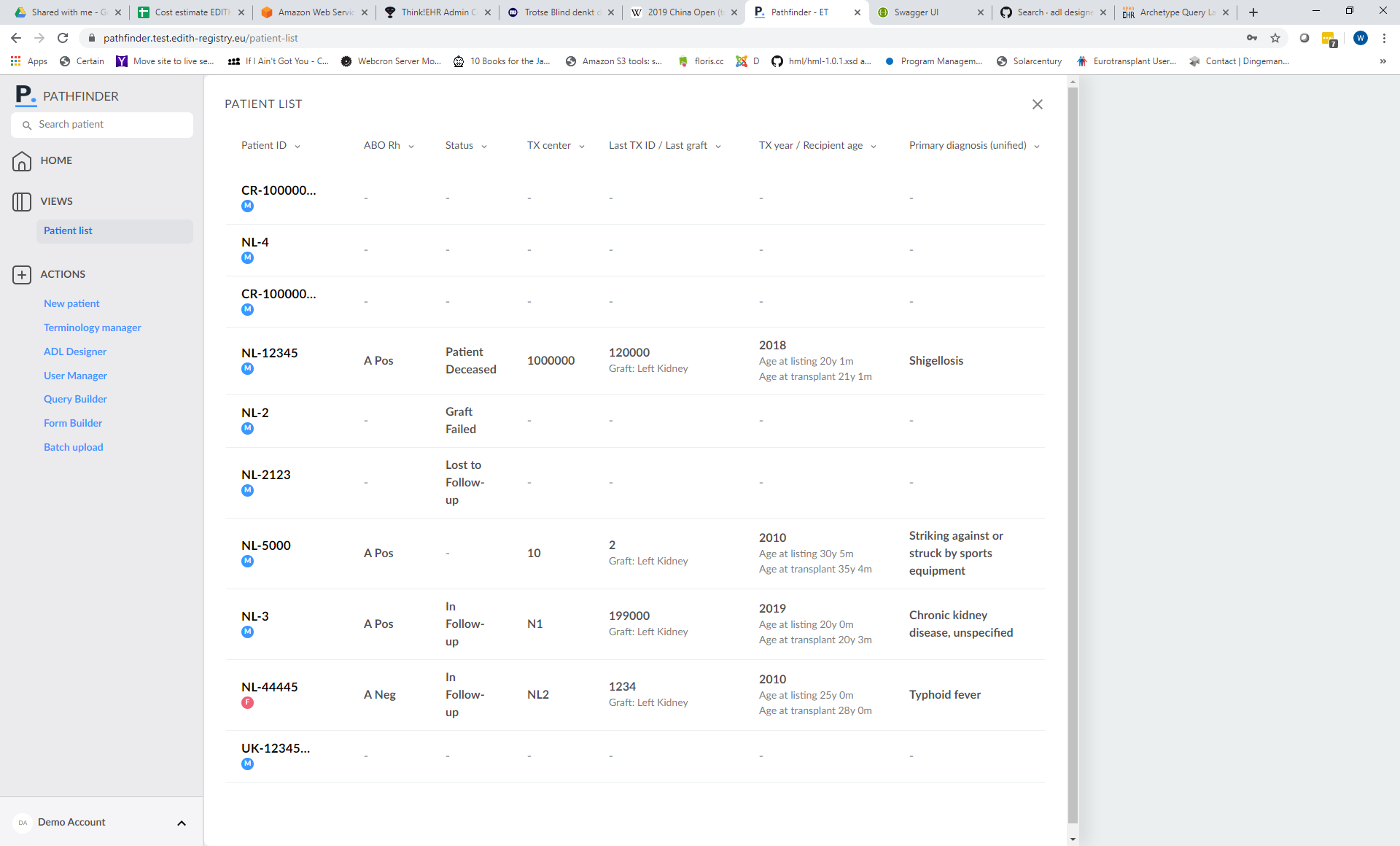
Better Pathfinder is on openEHR front-end for quickly publishing forms based on openEHR so that endusers can use them. It is focused on data capturing and needs the Better openEHR platform to run on

Figure 8 Pathfinder - Patient list

The following function can be used be End User within PathFinder.

Patient list  
A list of all patients (visible for you user), you can select patients from here.

Search patient  
Find a patient by patient ID

New Patient  
Register a new patient ID

Bulk Upload  
Upload a csv file (format can be found on Github) containing multiple Initial or Follow Up records.

Patient view

Contains patient record, Follow-up forms.

Patient record

Summary, documents.

**Configuration**

In the Github repository the forms and import formats can be found. A manual will be uploaded to the Github repository

# Statistics (Metabase)

**Description**

**Metabase is an opensource statistics program running on a docker instance in our configuration. It uses the postgress research database as its source database.**

**Use**

All NCA’s will get access to their own dataset as well as the full dataset anonymized. They can run some statistics on within metabase, but the main purpose is to be able to download the data in a flat and readable format.

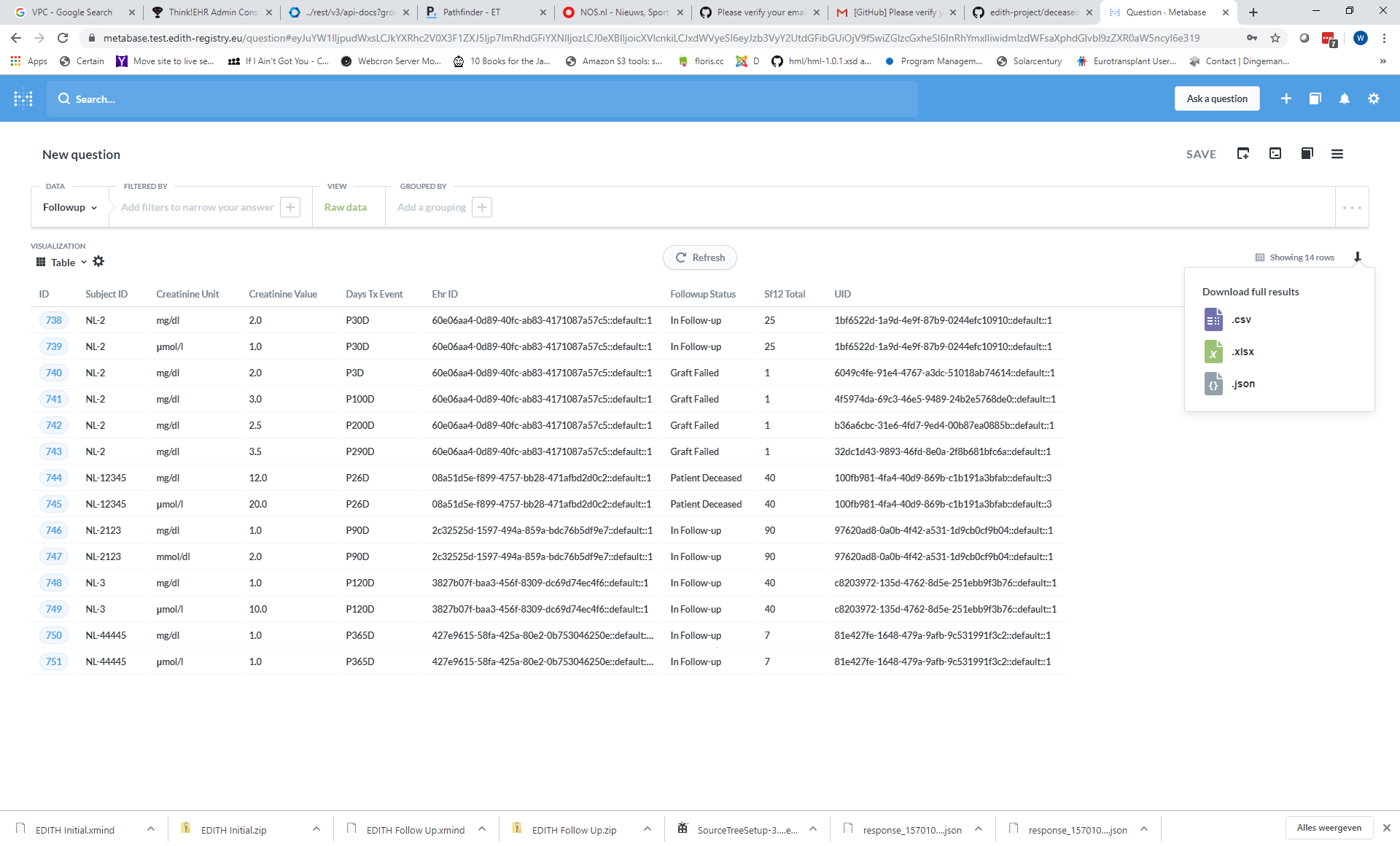
Downloads are available in csv, xlsx and json format. Columns are configurable.

Figure 9 Metabase download

**Configuration**

<< to be determined >>