



# SPECIFICATIONS

EDITH transplant recipient registry

[Abstract](#)

Functional and technical specifications for the EDITH transplant recipient registry

# Specifications

## EDITH transplant recipient 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.....	2
Open platform .....	2
openEHR .....	2
Flexibility .....	2
Resources.....	3
Components.....	4
Flow between components.....	5
AWS cloud configuration.....	6
openEHR archetypes and templates.....	8
User management .....	10
openEHR platform (Better) .....	11
ETL.....	13
Terminology adapter.....	14
Form builder .....	15
Pathfinder.....	16
Statistics (Metabase).....	20

# Introduction

---

## Open platform

The EDITH transplant recipient registry is based on an open platform: openEHR. Open platforms are open in the sense that data can be available in 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](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 archetypes 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, therefore it has a high initial cost, but perhaps lower running cost than a paid solution. However as the EDITH transplant recipient 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, network is needed any more.

## Functional Requirements

The solution is based on the requirement document EDITH design\_v1\_1.pdf (see Github repository).

## Resources

Resources mentioned in this document can be found 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.

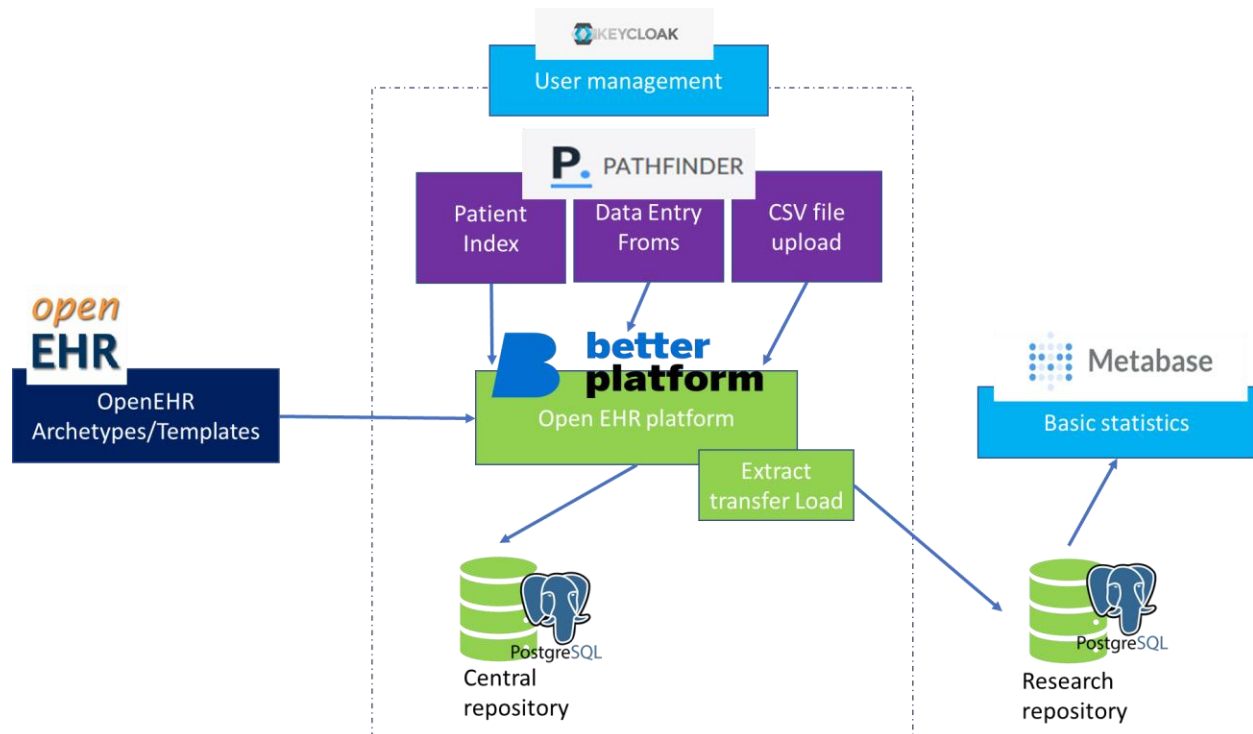


Figure 1 Schematic representation of components and how they interact

## openEHR: archetypes and templates

Archetypes 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 its 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:

Think!EHR Explorer & Terminology adapter	EC2 Instance M5.Large
Think!EHR Platform	EC2 Instance T3.medium
Instance storage (estimate 20Gb per Instance)	EBS General Purpose SSD
PostgreSQL	Database
PostgreSQL (estimate 5Gb p/y)*	DB Storage
PostgreSQL (estimate 5Gb p/y)*	DB Backup Storage
PostgreSQL (estimate 10Gb p/y)**	Data transfer

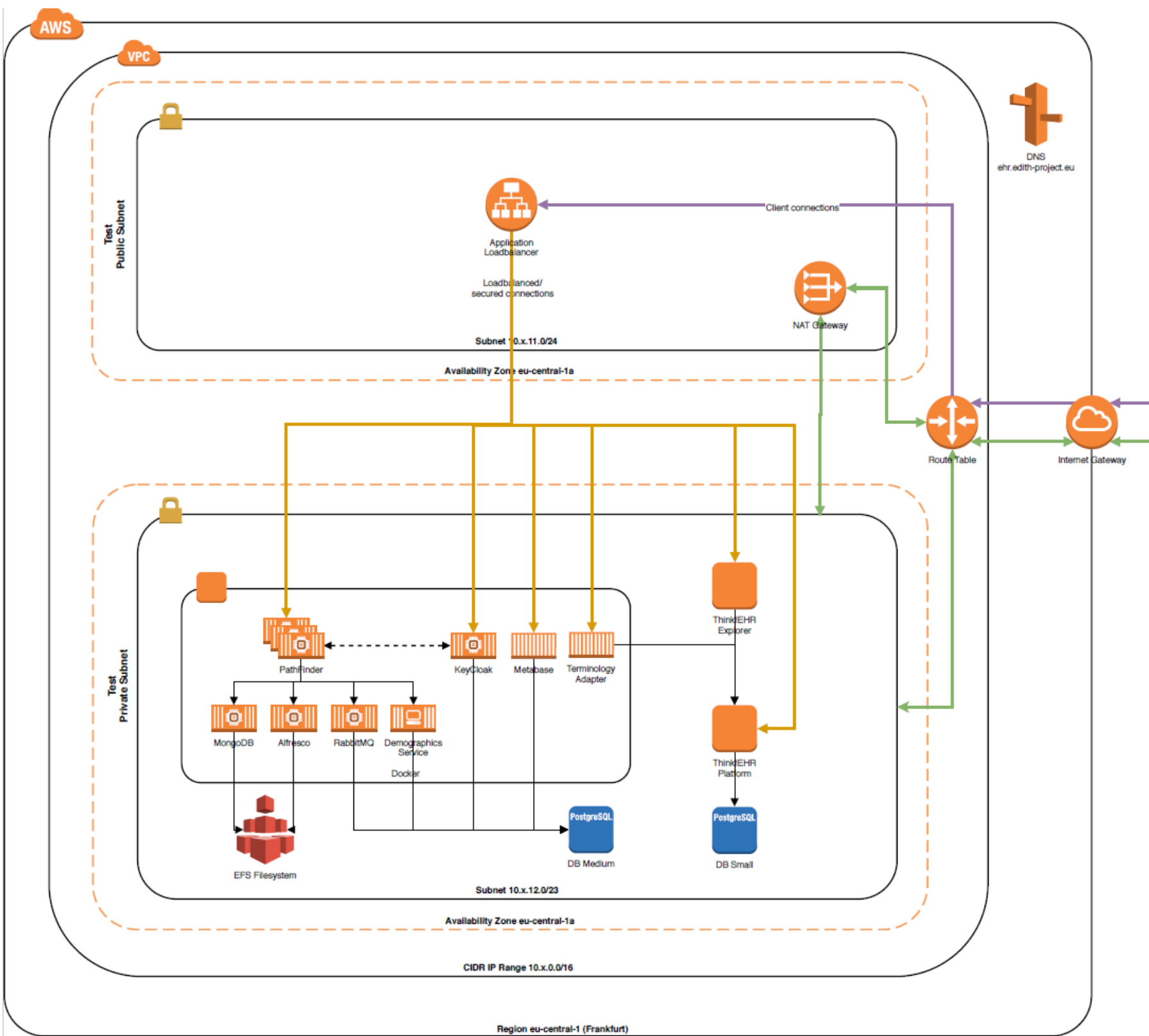
*Figure 2 AWS components used per environment (test, prod)*

Application Loadbalancer	
Route53 DNS	Hosted zone
VPC	VPN
Backup Storage for Instance snapshots	S3

*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 archetypes 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)

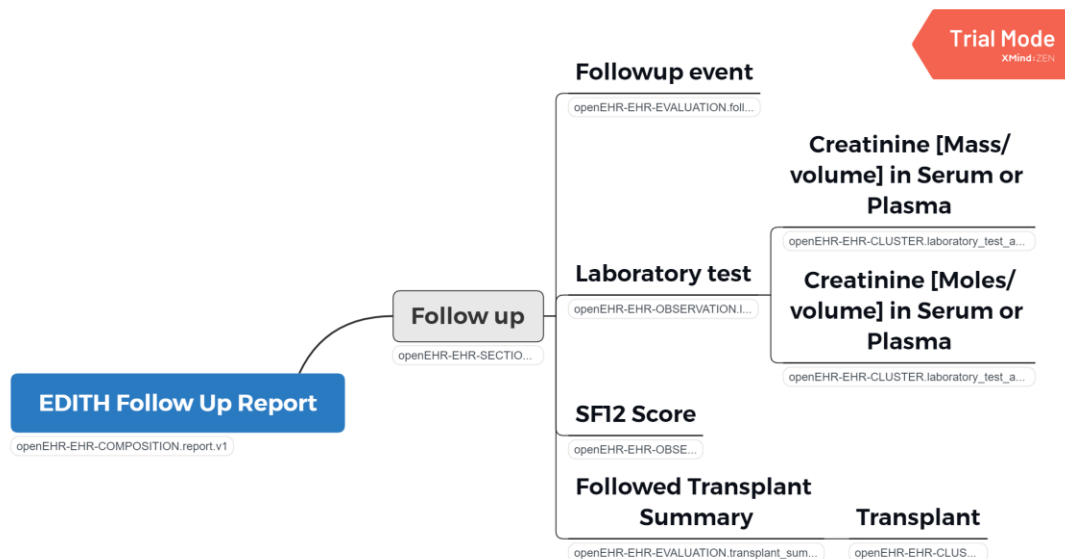


Figure 4 EDITH Follow Up archetypes mind map

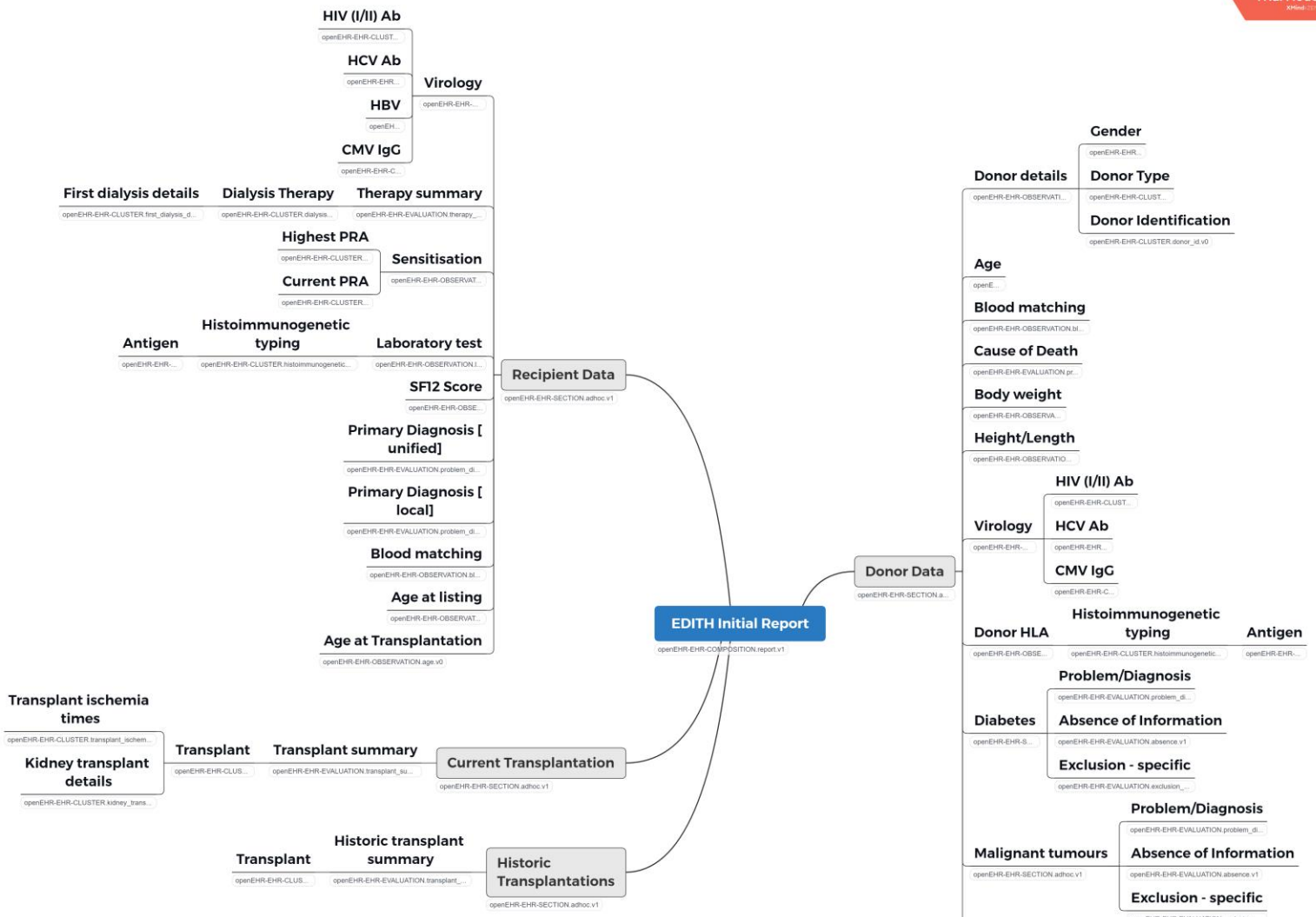


Figure 5 EDITH Initial archetypes mind map

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
2. 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

The platform can be configured by Admins

The following artefacts are uploaded to run the application:

- Templates (see openEHR archetypes and templates)

Forms (see

The screenshot displays the Think!Ehr admin console interface. The top header bar is blue and contains the text 'Think!Ehr' and 'thinkehr.test.edith-registry.eu:443 (2.4.3) /opt/thinkehr'. A dark grey sidebar on the left lists navigation options: Status, Calls, Templates, Forms, Users, Domains, Events, Views, Data Sources, Query, REST API, and OpenEhr REST API. The main content area is white and shows the 'Status' section expanded. It includes a 'Server Status' table with rows for Application status, Database status, External database status, Index status, Cache status, Memory, and EHR count / license limit. Below this is a 'Cache Stats' table with columns for Cache and Count, listing COMPOSITIONS, EHRS, PATH\_NAMES, QUERIES, SESSIONS, and VERSIONED\_DATA. At the bottom, the 'Operations' section provides links for Reset, Reindex compositions, Reindex committed in last, Flush audit log, and Reset cache stats.

Server Status		
Application status	OK	(call time 0 ms)
Database status	OK	(call time 1 ms)
External database status	OK	
Index status	OK	(call time 2 ms) [GREEN shards]
Cache status	OK	(call time 0 ms)
Memory	OK	991MB/2048MB
EHR count / license limit	OK	OK (10 / 250000)

Cache Stats	
Cache	Count
COMPOSITIONS	15
EHRS	0
PATH_NAMES	393
QUERIES	24
SESSIONS	0
VERSIONED_DATA	2

**Operations**

[Reset](#)

[Reindex compositions \(missing only\) \(truncate first\)](#)

Reindex committed in last: [1h](#) — [3h](#) — [6h](#) — [12h](#) — [24h](#)

[Flush audit log](#)

[Reset cache stats](#)

Figure 6 Admin console Better platform

- 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. ETL tool from Better version 1.1.6 is used.

## 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 .

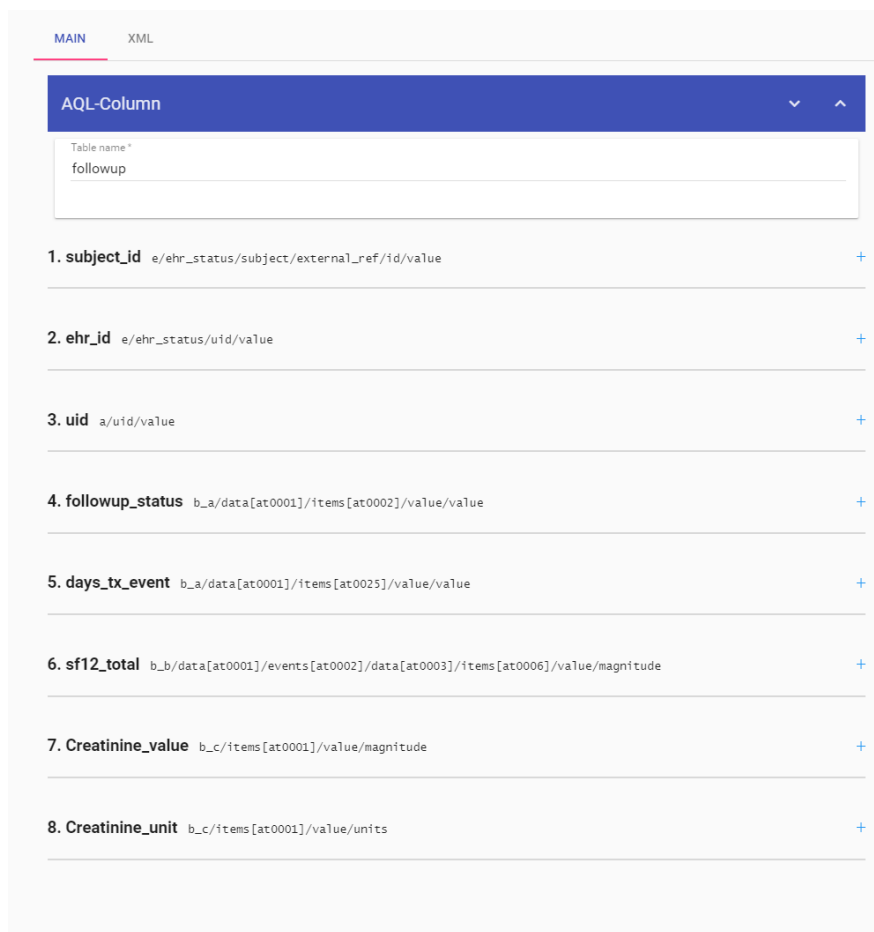
For each country a query is defined to retrieve country specific data from the openEHR, platform as well as European query (which excludes the Subject ID).

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.



The screenshot shows a web interface with two tabs: 'MAIN' and 'XML'. The 'MAIN' tab is active. Below the tabs is a header 'AQL-Column' with a dropdown arrow and an up arrow. Under this header is a text input field labeled 'Table name \*' containing the text 'followup'. Below the input field is a table with 8 rows, each representing a mapping from an AQL field to a SQL field. Each row has a blue '+' icon on the right side.

AQL-Column	
Table name * followup	
1. subject_id	e/ehr_status/subject/external_ref/id/value
2. ehr_id	e/ehr_status/uid/value
3. uid	a/uid/value
4. followup_status	b_a/data[at0001]/items[at0002]/value/value
5. days_tx_event	b_a/data[at0001]/items[at0025]/value/value
6. sf12_total	b_b/data[at0001]/events[at0002]/data[at0003]/items[at0006]/value/magnitude
7. Creatinine_value	b_c/items[at0001]/value/magnitude
8. Creatinine_unit	b_c/items[at0001]/value/units

Figure 7AQL to SQL mapping example

# Terminology adapter

## Description

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

## Use

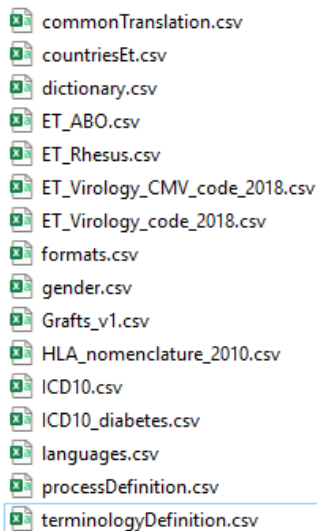
The platform can be configured by Admins. Connections are defined one to the openEHR instance and one connections to the Europe schema research database. And one for every country schema .

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

- code, the code for a row within the Nomenclature
- description, 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):



- commonTranslation.csv
- countriesEt.csv
- dictionary.csv
- ET\_ABO.csv
- ET\_Rhesus.csv
- ET\_Virology\_CMV\_code\_2018.csv
- ET\_Virology\_code\_2018.csv
- formats.csv
- gender.csv
- Grafts\_v1.csv
- HLA\_nomenclature\_2010.csv
- ICD10.csv
- ICD10\_diabetes.csv
- languages.csv
- processDefinition.csv
- terminologyDefinition.csv

# 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 8 Part of the Edith Initial form



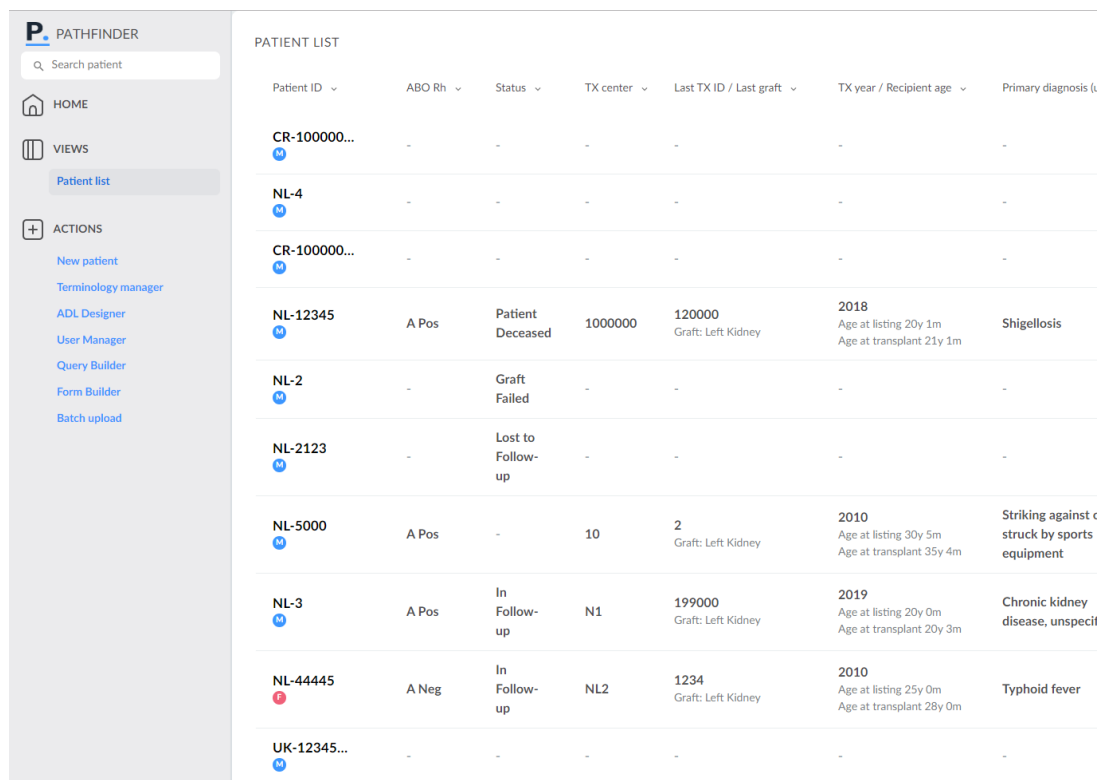
# Pathfinder

## 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 of.

## 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



P. PATHFINDER						
Search patient						
HOME						
VIEWS						
Patient list						
ACTIONS						
New patient						
Terminology manager						
ADL Designer						
User Manager						
Query Builder						
Form Builder						
Batch upload						
PATIENT LIST						
Patient ID	ABO Rh	Status	TX center	Last TX ID / Last graft	TX year / Recipient age	Primary diagnosis (L)
CR-100000...	-	-	-	-	-	-
NL-4	-	-	-	-	-	-
CR-100000...	-	-	-	-	-	-
NL-12345	A Pos	Patient Deceased	1000000	120000 Graft: Left Kidney	2018 Age at listing 20y 1m Age at transplant 21y 1m	Shigellosis
NL-2	-	Graft Failed	-	-	-	-
NL-2123	-	Lost to Follow-up	-	-	-	-
NL-5000	A Pos	-	10	2 Graft: Left Kidney	2010 Age at listing 30y 5m Age at transplant 35y 4m	Striking against c struck by sports equipment
NL-3	A Pos	In Follow-up	N1	199000 Graft: Left Kidney	2019 Age at listing 20y 0m Age at transplant 20y 3m	Chronic kidney disease, unspecil
NL-44445	A Neg	In Follow-up	NL2	1234 Graft: Left Kidney	2010 Age at listing 25y 0m Age at transplant 28y 0m	Typhoid fever
UK-12345...	-	-	-	-	-	-

Figure 9 Pathfinder - Patient list

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

### Patient list

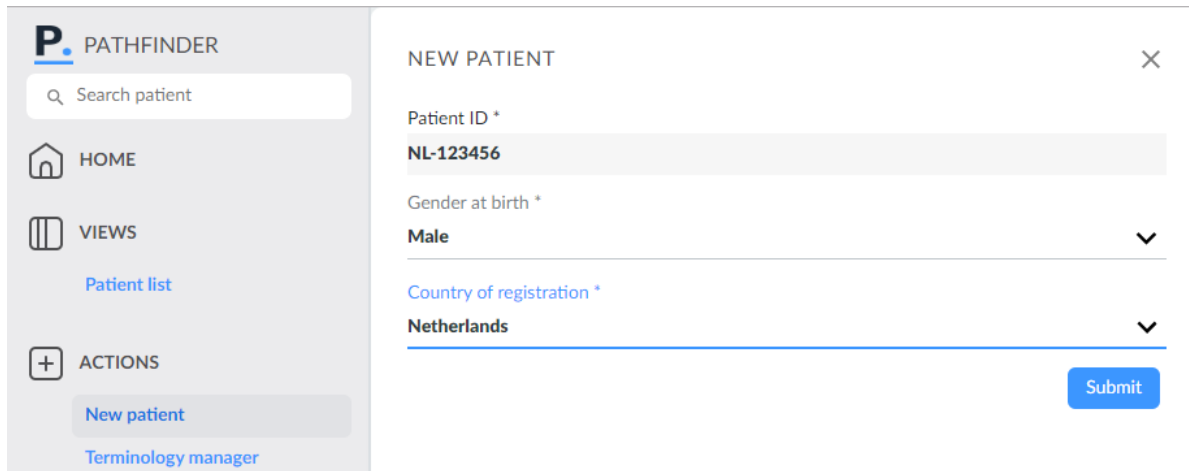
A list of all patients (visible for you user), you can select patients from here.

### Search patient

Find a patient by patient ID.

## New Patient

Register a new patient ID. Patient ID should start with countrycode and followed by a pseudonymised patient ID (Like NL-12345) generated by the NCA .



**P. PATHFINDER**

Q Search patient

**HOME**

**VIEWS**

Patient list

**ACTIONS**

New patient

Terminology manager

**NEW PATIENT** ✕

Patient ID \*

**NL-123456**

Gender at birth \*

**Male** ▼

Country of registration \*

**Netherlands** ▼

**Submit**

Figure 10 Register new patient

## Batch Upload

Upload a csv file (format can be found on Github) containing multiple Initial or Follow Up records . Before uploading the csv file select the correct template from which the upload file was generated: "Edith Follow Up" or "Edit Initial".

## Patient view

Contains patient record, forms.

## Patient record

The screenshot shows the 'PATIENT RECORD' interface for patient NL-5000. On the left is a sidebar with navigation icons and links for 'Summary' and 'Documents'. The main area is titled 'SUMMARY' and contains several data cards:

- Transplant Center:** 10
- Transplant:** Last transplantation ID 2, Last established graft Left Kidney
- Primary Diagnosis:** Unified Striking against or struck by sports equipment, Local Dum
- Transplant year/Recipient age:** Transplant date 2010, Recipient age at transplant 30y 5m / 35y 4m
- Donor Identification:** Donor ID 3

Figure 11 Patient record with summary

The patient record contains a summary of the latest values for a patient. Also you can access all Initial and Follow-up form under the documents section. Within the documents session you can also edit the documents if they contain false information.

## Forms

By opening one of the two follow-up forms: "Edith Initial" or "Edith Follow-up" you can enter the respective form for this patient.

The screenshot shows the '2. EDITH FOLLOW UP - NEW VERSION' form for patient NL-5000. The form is titled 'Transplantation Follow Up' and includes a brief instruction: 'After 6 months of the transplantation procedure, an informative follow-up should be gathered from the patient in order to know what is the current status and his well being.' The form is divided into three sections:

- Process Identification:** Transplantation ID (text input field)
- Follow Up Status:** Follow-up Status (dropdown menu with 'Select option in the dropdown.'), Days between transplant and event (text input field with 'days' unit)
- Laboratory Test Creatinine:** Referent to values of the analyt result. At least one of both dimensional units needs to be filled (mass/volume or moles/volume). Creatinine dimensional unit (radio buttons for '[mass/volume]' and '[moles/volume]')

Figure 12 Follow-up form

## **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 postgres 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.

New question SAVE    

DATA: Followup VIEW: Raw data GROUPED BY: Add a grouping Filtered by: Add filters to narrow your answer

Visualization: Table Refresh Showing 14 rows

ID	Subject ID	Creatinine Unit	Creatinine Value	Days Tx Event	Ehr ID	Followup Status	Sf12 Total	UID
738	NL-2	mg/dl	2.0	P30D	60e06aa4-0d89-40fc-ab83-4171087a57c5:default::1	In Follow-up	25	1bf6522d-1a9d-4e9f-87b9-0244efc10910:default::1
739	NL-2	μmol/l	1.0	P30D	60e06aa4-0d89-40fc-ab83-4171087a57c5:default::1	In Follow-up	25	1bf6522d-1a9d-4e9f-87b9-0244efc10910:default::1
740	NL-2	mg/dl	2.0	P3D	60e06aa4-0d89-40fc-ab83-4171087a57c5:default::1	Graft Failed	1	6049c4fe-91e4-4767-a3dc-51018ab74614:default::1
741	NL-2	mg/dl	3.0	P100D	60e06aa4-0d89-40fc-ab83-4171087a57c5:default::1	Graft Failed	1	4f5974da-69c3-46e5-9489-24b2e5768de0:default::1
742	NL-2	mg/dl	2.5	P200D	60e06aa4-0d89-40fc-ab83-4171087a57c5:default::1	Graft Failed	1	b36a6cbc-31e6-4fd7-9ed4-00b87ea0885b:default::1
743	NL-2	mg/dl	3.5	P290D	60e06aa4-0d89-40fc-ab83-4171087a57c5:default::1	Graft Failed	1	32dc1d43-9893-46fd-8e0a-2f8b681bf6ca:default::1
744	NL-12345	mg/dl	12.0	P26D	08a51d5e-f899-4757-bb28-471afb2d0c2:default::1	Patient Deceased	40	100fb981-4fa4-40d9-869b-c1b191a3bfab:default::3
745	NL-12345	μmol/l	20.0	P26D	08a51d5e-f899-4757-bb28-471afb2d0c2:default::1	Patient Deceased	40	100fb981-4fa4-40d9-869b-c1b191a3bfab:default::3
746	NL-2123	mg/dl	1.0	P90D	2c32525d-1597-494a-859a-bdc76b5df9e7:default::1	In Follow-up	90	97620ad8-0a0b-4f42-a531-1d9cb0cf9b04:default::1
747	NL-2123	mmol/dl	2.0	P90D	2c32525d-1597-494a-859a-bdc76b5df9e7:default::1	In Follow-up	90	97620ad8-0a0b-4f42-a531-1d9cb0cf9b04:default::1
748	NL-3	mg/dl	1.0	P120D	3827b07f-baa3-456f-8309-dc69d74ec4f6:default::1	In Follow-up	40	c8203972-135d-4762-8d5e-251ebb9f3b76:default::1
749	NL-3	μmol/l	10.0	P120D	3827b07f-baa3-456f-8309-dc69d74ec4f6:default::1	In Follow-up	40	c8203972-135d-4762-8d5e-251ebb9f3b76:default::1
750	NL-44445	mg/dl	1.0	P365D	427e9615-58fa-425a-80e2-0b753046250e:default::1	In Follow-up	7	81e427fe-1648-479a-9afb-9c531991f3c2:default::1
751	NL-44445	μmol/l	1.0	P365D	427e9615-58fa-425a-80e2-0b753046250e:default::1	In Follow-up	7	81e427fe-1648-479a-9afb-9c531991f3c2:default::1

Figure 13 Metabase download

## Configuration

<< to be determined >>