Wouter Zanen

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 2](#_Toc20924816)

[Open platform 2](#_Toc20924817)

[openEHR 2](#_Toc20924818)

[Flexibility 2](#_Toc20924819)

[Components 4](#_Toc20924820)

[AWS cloud configuration 6](#_Toc20924821)

[openEHR archetypes and templates 8](#_Toc20924822)

[User management 8](#_Toc20924823)

[openEHR platform (Better) 8](#_Toc20924824)

[Statistics (Metabase) 9](#_Toc20924825)

# 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

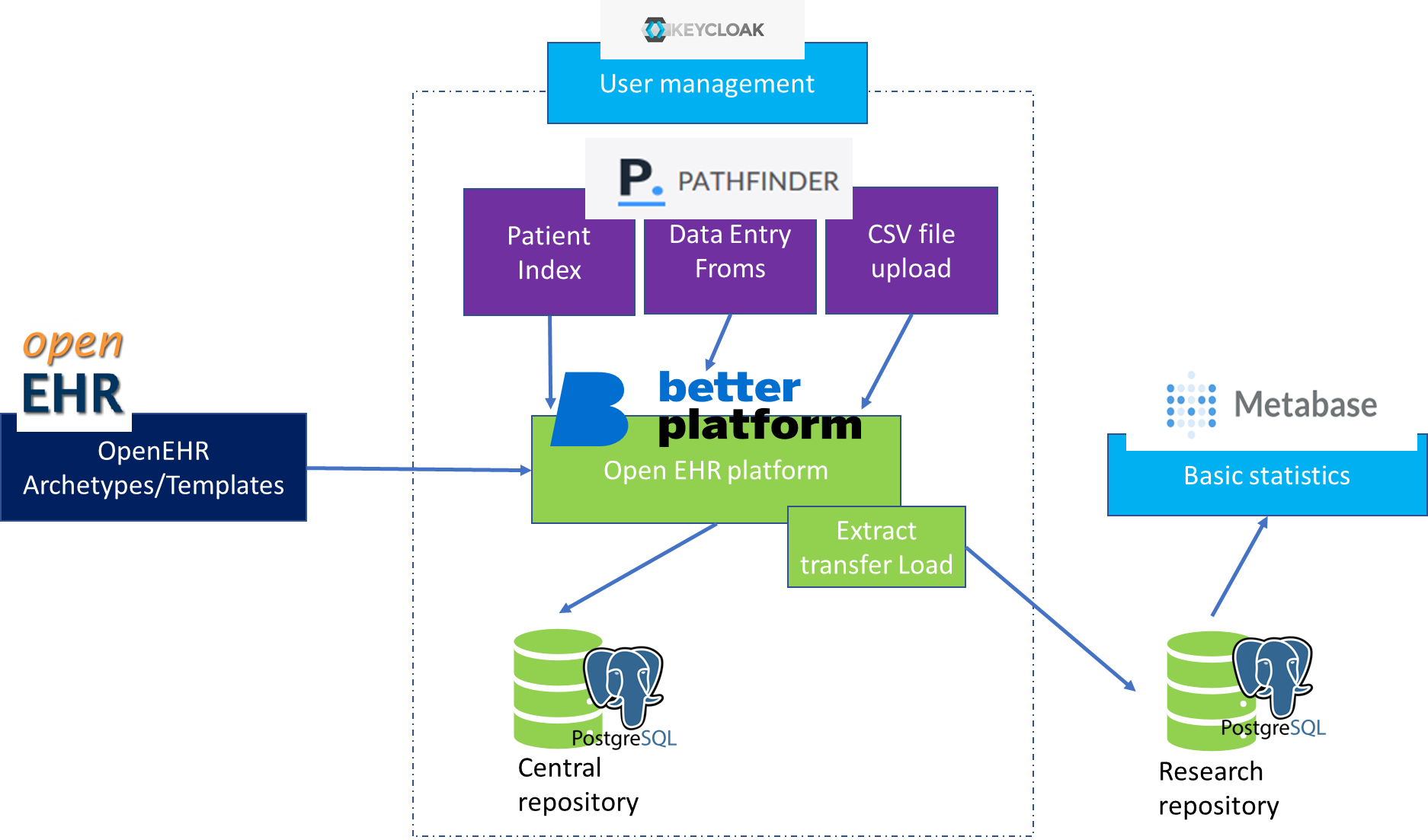
The registry uses several components to create , in this chapter we will describe them shortly and explain how the work together to provide the functionality as described in the requirements document.

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:

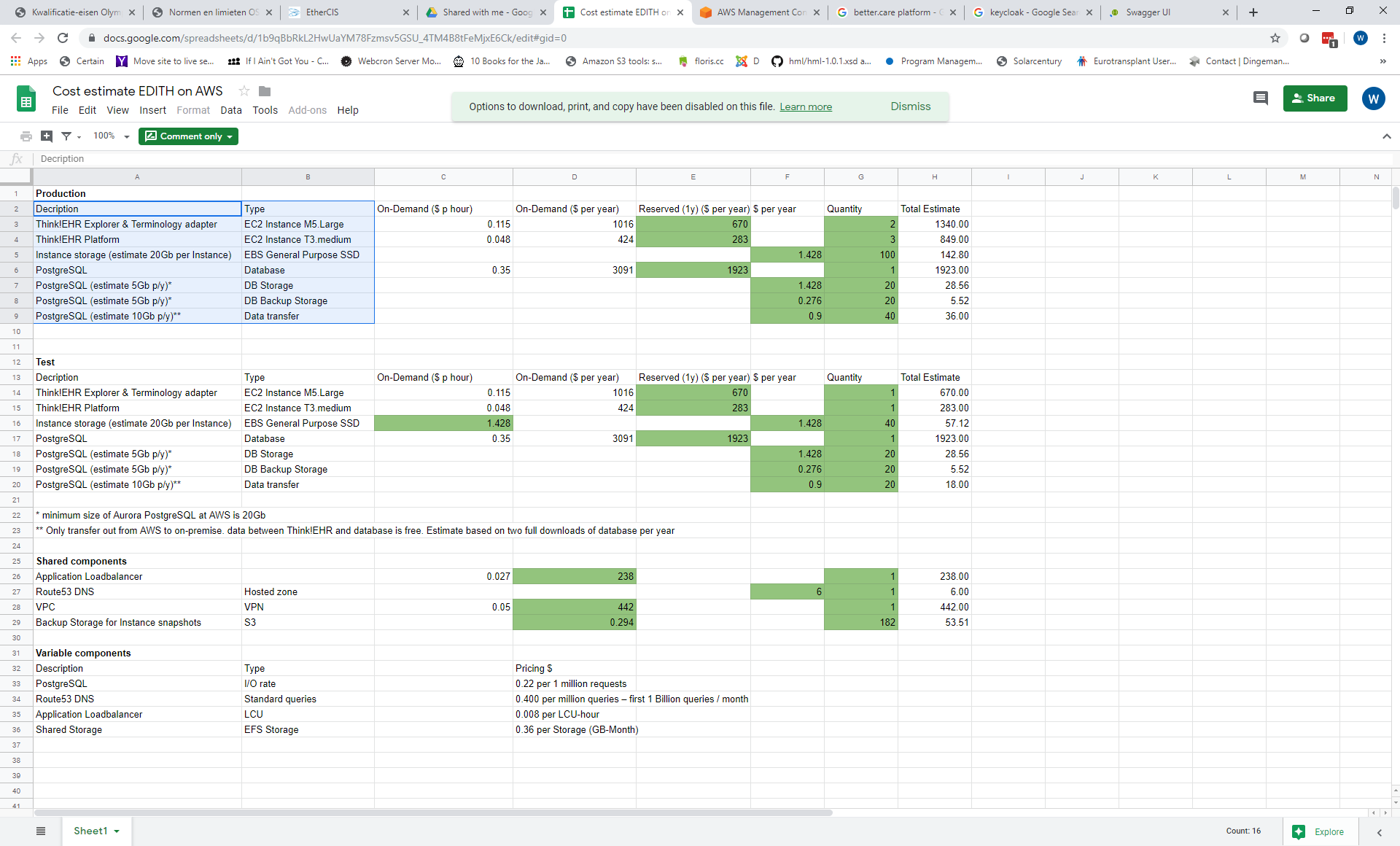


Figure AWS components used per environment (test, prod)

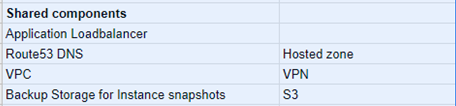
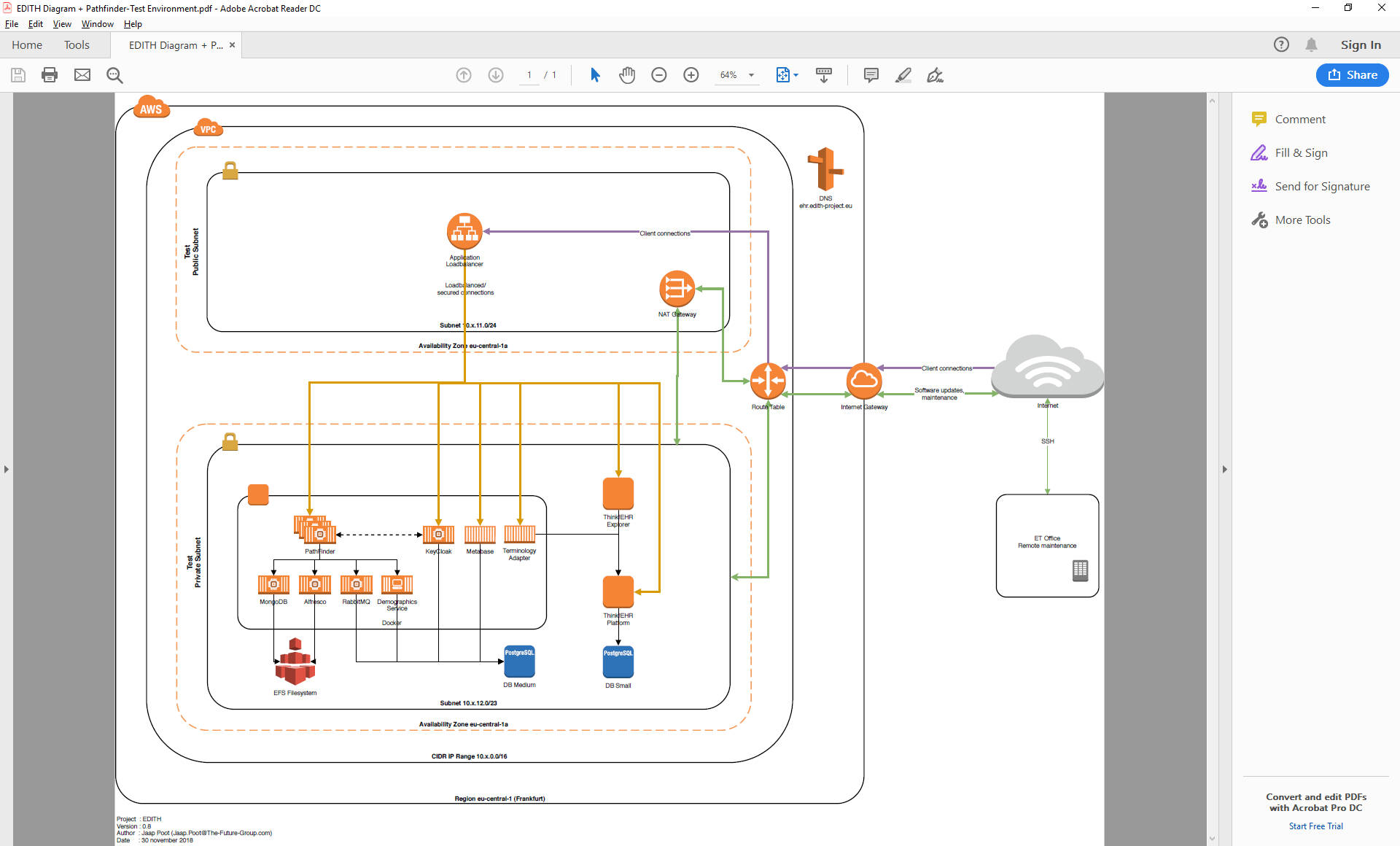


Figure Components shared between environments

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

# User management

# openEHR platform (Better)

## ETL

## Terminology adapter

## EHR Explorer

## Form builder

# Pathfinder

# Statistics (Metabase)

## Description

## Use

## Configuration