|  |
| --- |
| EMREX  Solution Outline  For projekt UFM-IT |

|  |  |  |  |
| --- | --- | --- | --- |
| Dokumentplacering: | | | |
| C:\Users\Z8QNK\git\dk-emrex\docs\Solution Outline for the Danish EMREX project.docx | | | |
| Senest reviewet dato: | Senest reviewet af: | Senest godkendt dato: | Senest godkendt af: |
| <dd-mm-åååå> | <initialer> | <dd-mm-åååå> | <initialer> |

|  |  |
| --- | --- |
| Jira-sag | ST-1688 |

Dokumenthistorik

Nyeste revisioner indskrives i omvendt kronologisk rækkefølge. Godkendelse er markeret med fed skrift.

|  |  |  |  |
| --- | --- | --- | --- |
| Dato | Revision | Redaktør | Ændringer |
| <dd-mm-åååå> | <rn> | <initialer> | <Beskriv ændring. Evt. med henvisning til POB sag, ændringsanmodning, fejlrapport eller anden anledning> |
| 07.12.2016 |  | NK | Server domains added. Software levels adjusted. |
| 18.12.2015 | <rn> | FV | Technology changed from Oracle ADF to Java/Angular. |
| 23-10-2015 | <rn> | FV | Document is updated after review meeting 23/10 with Anders |
| 01-10-2015 | <rn> | FV | Document is updated after meeting 25/9. |
| 22-09-2015 | <rn> | FV | Section describing environments is added |
| 18-09-2015 | <rn> | FV | Sequence diagram added for the NCP |
| 01-09-2015 | <rn> | FV | Document initiated |

Referencer

|  |  |
| --- | --- |
| Reference | Titel |
| Ingen | Ingen |

Begreber og forkortelser

|  |  |
| --- | --- |
| Begreb/Forkortelse | Betydning |
| <begreb/forkortelse> | <forklaring eller henvisning til forklaring> |

Table of contents

[1 Introduction 5](#_Toc468883269)

[2 Requirements 6](#_Toc468883270)

[2.1 Requirements for the EMREX client: 6](#_Toc468883271)

[2.2 Requirements for the National Contact Point (NCP): 6](#_Toc468883272)

[2.3 Security 6](#_Toc468883273)

[2.4 Logging 7](#_Toc468883274)

[3 Implementation scope 8](#_Toc468883275)

[3.1 Implementation approach 8](#_Toc468883276)

[4 Solution outline 9](#_Toc468883277)

[4.1 The National Contact Point (NCP) 9](#_Toc468883278)

[4.1.1 NCP frontend 10](#_Toc468883279)

[4.1.2 NCP backend 10](#_Toc468883280)

[4.1.3 Sequence diagram 11](#_Toc468883281)

[4.2 The EMREX client 11](#_Toc468883282)

[4.2.1 EMREX client 12](#_Toc468883283)

[4.2.2 SMP functionality 13](#_Toc468883284)

[4.2.3 The EMREX backend - A STADS webservice 13](#_Toc468883285)

[4.2.4 Sequence diagram 13](#_Toc468883286)

[4.3 Security 15](#_Toc468883287)

[4.4 Logging 15](#_Toc468883288)

[5 Environments 16](#_Toc468883289)

[5.1 Overall environment setup 16](#_Toc468883290)

[5.2 New servers needed 18](#_Toc468883291)

Figures

[Figure 1 - Overview of the EMREX project scope 5](#_Toc431380030)

[Figure 2 - implementation approach 9](#_Toc431380031)

[Figure 3 – Solution overview for NCP 10](#_Toc431380032)

[Figure 4 - Sequence diagram for the NCP solution 11](#_Toc431380033)

[Figure 5 - Solution overview for the EMREX client 12](#_Toc431380034)

[Figure 6 - Sequence diagram for the EMREX client 14](#_Toc431380035)

[Figure 7 – Environments and interfaces for the EMREX project 17](#_Toc431380036)

**Tabels**

Har ikke fundet nogen tekst til listen over figurer.

# Introduction

This document relates to the EMREX project which UFM-IT is part of. The intension is to give a understanding of which high level technical “components” that must be developed to full fill the requirements from the EMREX project, and how they relates to the existing DANS and STADS system implementation in Denmark.

The objective is to establish a common view for UFM-IT and KMD teams. This common view will lead to decision on the scope and a work breakdown structure (WBS) of the tasks to be planned within DANS and STADS.

Before reading this document, the EMREX project document (“EMREX\_GD2015\_Malaga.pdf”) must be read. This document provides the basic understanding of the EMREX project, below figure is from this document.

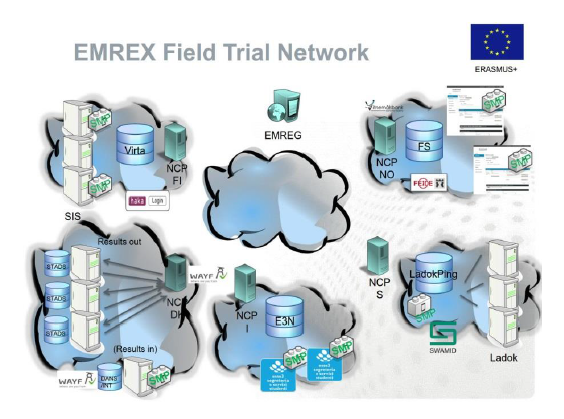


Figure 1 - Overview of the EMREX project scope

This document does not analyse the technical and functional dependencies to the INT delivery project for DANS.

# Requirements

Two main components have to be developed in the Danish part of the EMREX project.

1. The EMREX client
2. A National Contact Point (NCP)

Below are listed requirements and descriptions for the 2 main components, and the associated security requirements.

## Requirements for the EMREX client:

Each SIS or institution, depending on local implementation, that wishes to fetch results via the EMREX network, must implement a client. This client must provide the following functionality:

* Provide the student with a secure login.
* Enable the student to select which country they wish to fetch results from.
* Direct the student to the selected country.
* Display results that have been fetched.
* Provide a way to store the results in the local SIS

Note that the EMREX project delivers a SMP component; this is a component which can be used in the implementation of the EMREX client. The SMP is a library service that offers functionality to support the transfer of results, this includes also some security functionality.

## Requirements for the National Contact Point (NCP):

Each country that wishes to provide results to the EMREX network, must implement a NCP, or a number of NCPs, that will enable an EMREX client to fetch results from the given country or HEI. The NCP must provide the following functionality:

* Provide the student with a secure login / authentication.
* Enable the student to select and fetch their results from the desired HEI.
* Return the selected results to the EMREX client.

The output data format must be in the format similar to the ELMO format. The format that will be used will be a fork of the ELMO format, but customized for the EMREX purpose. This is referred to as the “EMREX application profile”

## Security

One of the important features of EMREX is the quality and reliability of the data. In order to ensure this, it is important that the transfer of data is done in a safe way. Several steps must be taken towards securing the EMREX data transfer:

* Double login: The student must log in, using secure login, in both the EMREX client and the NCP.
* Enforcing HTTPS: EMREX will be enforcing the use of HTTPS to transfer data.
* Digitally signing the data: EMREX will use signatures for the data to ensure that the NCP is a valid one. The public keys are stored in EMREG.
* Verification of student: Since a student has to log in on both sides of the transfer, we can validate whether it is indeed the same student. We validate on gender, date of birth and name. Since name can be written differently in different countries, we use a Levenshtein algorithm with threshold to check the name.

In addition, the data is never "touched" by the student, meaning that the data cannot be tampered with.

## Logging

There is currently not defined any requirements for logging.

# Implementation scope

In this section the high level implementation approach is described. This includes how the developed solution will integrate with the existing STADS and DANS application.

The high level components in the solution are the EMREX client and the NCP, both with a backend part.

Note that the EMREX client and the corresponding EMREX backend is related to a Danish mobility student (outgoing) who have studied outside Denmark The NCP (frontend + backend) is related to a mobility student (incoming) who have studied in Denmark.

## Implementation approach

The EMREX client and the NCP frontend are developed as standalone web applications. Both the EMREX client and the NCP will have a backend element in STADS. The EMREX backend will store the received results in STADS, the stored format will be a PDF file. The NCP backend will fetch the requested student results from STADS, the output format will be in the EMREX application profile.

Both the applications (not the backend part) will be deployed on a new EMREX infrastructure hosted by UFM-IT. The applications will be developed using Java / Angular stack, this is based on below assessment (In Danish).

|  |  |  |
| --- | --- | --- |
|  | **Oracle ADF** | **Java med Angular framework** |
| Flexibility in design of user interface. |  |  |
| Support for technology from at vendors and communities. |  |  |
| Good usage on mobil- and tablet devices (responsive design) |  |  |
| Degree of reuse of the Finish solution\* |  |  |
| The amount of reusable code when integrating the EMREX client frontend in DANS at a later point in time. |  |  |
| Raw developing cost for the NCP frontend and the EMREX client frontend. |  |  |
| License |  |  |
| Support for the technology in UFM-IT |  |  |

\* Please note that raw coding task is small in the larger project estimat, the time consuming part is deployment, configuration, certifikates, WAYF, test and communication with the NCP- and EMREX backend.

Note that there will be NO integration to DANS in the scope of this solution. In a future phase the EMREX Client might be merged into DANS.

The Danish SIS application is indicated with dotted line as a future part of the EMREX network



Figure 2 - implementation approach

# Solution outline

In this section the target solution is described in more details. The outset for the description is the 2 main components

1. National Contact Point (NCP)
2. The EMREX client

The description positions the 2 main components with the interfacing systems. Note that all elements colored blue in the drawings is new functionality which has to be implemented.

Please read the detailed Use Case description in the EMREX Confluence before continuing to 4.1 and 4.2:

<https://confluence.csc.fi/display/EMREX/Use+Cases>

## The National Contact Point (NCP)

In below figure the Danish NCP frontend and NCP backend is described into more details.



Figure 3 – Solution overview for NCP

### NCP frontend

This will be a set of webpages / functions that has to be developed; a UI design has to be developed for this.

The NCP frontend will be deployed on a WebLogic application server which handles the request from the EMREX client (from the other countries). WebLogic is chosen as it is the UFM-IT preferred application server; it can be used without any license cost for the project.

The webserver must be accessible from the internet. The NCP frontend holds info about which institutions that are EMREX enabled in Denmark, the student select an institution from this list. The NCP frontend will redirect the student to a secure login, using WAYF as a dispatcher. The student will login at the institution, i.e. KU, which hosted the student during the stay in Denmark. The assumption is that the student still has a valid login to the institution when returning to the home country.

Having logged into the institution, the NCP frontend will guide the student through the process of fetching the results from the courses which the user has completed. Finally the NCP frontend will submit a NCP response.

### NCP backend

The NCP backend is a set of webservices, currently 2 webservices has been identified. The webservices will either be reusing existing STADS webservices or will be developed from scratch. Further dataanalysis has to explorer that.

The webservices has to be deployed on all the decentral STADS instances (institutions) and be accessible from the NCP frontend. The webservice WS2, which returns the student results, has to deliver data in a XML file implementing the EMREX application profile format. The Webservice design and data mapping from STADS to XML will be detailed in a separate design from this.

### Sequence diagram

The overall flow for the NCP is described in more details in below figure.



Figure 4 - Sequence diagram for the NCP solution

## The EMREX client

In the figure below the EMREX client is described in details. There are 3 main elements (colored blue) which have to be implemented.

1. EMREX client
2. SMP functionality
3. The EMREX backend - A STADS webservice (STADS 1, STADS 2 etc.)



Figure 5 - Solution overview for the EMREX client

### EMREX client

The EMREX client will be deployed on the same WebLogic webserver as the NCP frontend, and must be accessible from the internet.

The EMREX client will be a web application which consists of a set of webpages/functions that has to be developed; a UI design has to be developed for this. The EMREX client will ensure the user login at a Danish university using WAYF, and implement interface to the SMP and the EMREX backend.

### SMP functionality

The SMP functionality is common open source code delivered by the EMREX project to each of the countries in the EMREX project. It will be implemented as a standalone webservice running on the Weblogic application server. Using the SMP ensures that the EMREX communication protocol is being followed. The SMP also provides a functionality to create a PDF document with the data received via the EMREX application profile.

Note that the SMP also performs a student match which returns the probability (a number) that the student sending the request is the same as the owner of the results. A threshold for the number has to be decided.

### The EMREX backend - A STADS webservice

The STADS webservice will be the webservice that imports the student results fetched by the EMREX client and store them in STADS. The format delivered to the STADS webservice is a PDF file and a XML-file (the EMREX application profile). The outset is that the webservice will store the data in an “import table” and a corresponding student ID).

The scope for this solution will not implement any new STADS UI functionality where the data can added into the existing data structure, this will be developed at a later stage if needed.

### Sequence diagram

The overall flow for the EMREX client is described in more details in below figure.



Figure 6 - Sequence diagram for the EMREX client

## Security

*20150901: In today’s EMREX call it was communicated that a general risk analysis is being conducted. Based on that, new requirements will be released, maybe with corresponding common EMREX security components.*

*Having received that, this document will be updated accordingly.*

## Logging

Expectation is that logging is needed in most of the involved components/systems, i.e. in the NCP backend (STADS).

# Environments

## Overall environment setup

To support the development and maintenance for the EMREX project below environments are needed.



Figure 7 – Environments and interfaces for the EMREX project

As illustrated in figure 7 it is assumed that the universities have a test login instance which can be used from the Development-, test- and QA environment. If this is not the case(Question1), the production login instance must be used also from the Development-, test- and QA environment. The result of this will be input to the planning of the test process.

## New servers needed

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Environment** | **Networkzone** | **Server - Function** | **Server OS version** | **Server  cores** | **Server memory** | **Domain** |
| Test | Internet facing | Applicationserver  (WebLogic 12.2.x) | RHEL7.0 | 2 | 8 GB | dans-emrexws.kmd.dk dans-emrex.kmd.dk |
| QA | Internet facing | Applicationserver  (WebLogic 12.2.x) | RHEL7.0 | 2 | 8 GB | emrex-test.stads.dk |
| Production | Internet facing | Applicationserver  (WebLogic 12.2.x) | RHEL7.0 | 2 | 8 GB | emrex.stads.dk |

UFM-IT is facing upgrade projects from WebLogic 11g to WebLogic 12c for DANS and STADS, the new servers will therefore be based on WebLogic 12c.

The server sizing is based on the fact that there will be very few transactions going through the above application infrastructure. The estimated number of students that will be using the applications once or twice a year will be around 500 students; this is equivalent to 500 – 1000 transactions on yearly basis.