|  |
| --- |
| DMIS  Laan van Vollenhove 2931  3706 AK Zeist  The Netherlands |
|  |  |  |
| Subject: XDS Security Test |  |  |

|  |
| --- |
|  |

# 

[System Details & Architecture 3](#_Toc63689821)

[Scope 5](#_Toc63689822)

[Input / output validation 5](#_Toc63689823)

[Login / logout (session) management and JWT tokens 5](#_Toc63689824)

[Image Upload functionality 7](#_Toc63689825)

[Not in Scope 7](#_Toc63689826)

[Black Duck Results 7](#_Toc63689827)

[Attachments 8](#_Toc63689828)

# System Details & Architecture

A demo setup is provided in a container called *docker-compose-forcare-all-in-one* – a schematic is given below.

**Note: the domain localhost should be repaced with url of test environment, which will be shared once the environment is accessible for testing purposes.**

A screenshot of a cell phone

Description automatically generated

**Figure 1: High-level overview.**

The container consists of the following XDS components:

* **ILM Service:** The ILM service is a component that provides a simple service API to perform information lifecycle management on documents in the XDS network.
* **forAdmin:** ForAdmin provides a web-based interface for component configuration and monitoring. It handles multiple components, even on different machines.
* **forAudit:** ForAudit is an IHE ATNA Audit Repository. It captures audit logs sent via UDP or TCP/SSL (syslog, per IHE ITI ATNA) and also supports HTTP or HTTPS based audit transactions.
* **forBridge:** ForBridge is the implementation of an XCA-gateway, and can act as either an Initiating Gateway or a Responding Gateway.
* **forConnect:** ForConnect is a broker component that converts legacy information streams in standardized formats such as DICOM and HL7 into XDS-compliant requests and vice versa.
* **forIndex:** ForIndex is a Document Registry and Master Patient Index implementation. It supports XDS.b, and handles HL7 ADT messages to build/serve a patient index.
* **forNotify:** ForNotify provides an isolated module to enhance workflow, for example send out e-mail notifications and/or HL7 messages based on DSUB notifications and provide a Policy Information Point service to support workflow attributes.
* **forRoute:** ForRoute is a broker component that uses Enterprise Integration Patterns to allow queuing and modification of messages.
* **forStore:** ForStore is an XDS Document Repository. The function of the **forStore:** component is to convert the ebXML XDS transactions to a suitable format for storage and to retrieve these transactions from the file system.
* **forView:** ForView is a generic XDS consumer as well as an XDS source. It is used to visualize the content of the registry and repository (or repositories) in the XDS affinity domain.
* **forWorkflow:** ForWorkflow is a component that manages patient workflows throughout XDS/XCA networks.

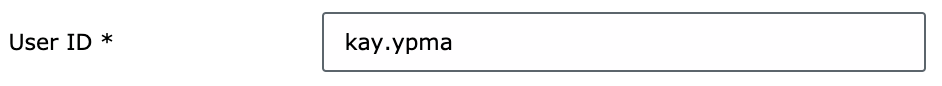
A PACS system is simulated with a dcm4che docker image. The system provides 4 publicly accessible end-points:

* <https://localhost/admin>
* <https://localhost/audit>
* <https://localhost/imageupload>
* <https://localhost/viewer>

The admin credentials for all urls are:

|  |  |
| --- | --- |
| **username** | **password** |
| root | root |

User administration is possible via <https://localhost/admin>. Note that in the user management it is possible to see the roles associated with an user. If you want to logon as a user found in user management use the ID after ‘User ID’:

****

**Figure 2: User ID in user management panel.**

And use as password the part before the *dot (.)*, i.e.: kay. In the README file of the docker container more instructions are available. Additional technical document is available in [1].

# Scope

Of primary interest for this test is the public interfacing component ForView and the ImageUpload functionality.

## Certificate Management API

A new certificate API has been developed and is now part of forAdmin. Check if the API is sufficiently secured. Access forAdmin via <https://localhost/admin>.

Graphical user interface, application, table

Description automatically generated

## Login / logout (session) management and JWT tokens

A variety of network calls are made to the ForView backend – checking the network calls via developers tools after logging into forView gives a good overview. Check-up what happens with these calls after a user logged out. Validation of parameters might be worth checking. Figure 3 and 4 provide visuals on how a) to find the JWT and b) the refresh call made to refresh the token.

A screenshot of a cell phone

Description automatically generated

**Figure 3: JWT in Session Storage.**

**A screenshot of a computer screen

Description automatically generated**

**Figure 4: Call to refresh end-point.**

Note that it is possible to call the refresh end-point with the following parameters:

* lastUpdateTime
* request.preventCache
* patientID

## Image Upload functionality

Upload DICOMDIR or zip files with DICOM files. Testing this functionality is possible via the following url (login with for example, user: root and password: root): <https://localhost/imageupload/?assigningAuthority=1.3.6.1.4.1.21367.2005.3.7#/start> (assigningAuthority is domain configuration).

Example data files are available in [2] to replay the following. After uploading it is possible to enter a patient ID – use an existing ID found for a given patient in ForView.

A screenshot of a cell phone

Description automatically generated

**Figure 5: Study details.**

After a successful upload the DICOM data is available for the user via ForView.

# Not in Scope

The docker environment launched for testing purposes is working with self-signed certificates. Production sites are *not* configured with self-signed certificates.

# Black Duck Results

See project ‘Forcare’ in [Black Duck](https://blackduck.philips.com/).

# Attachments

[1] Technical Reference Manual

[2] Example DICOM files