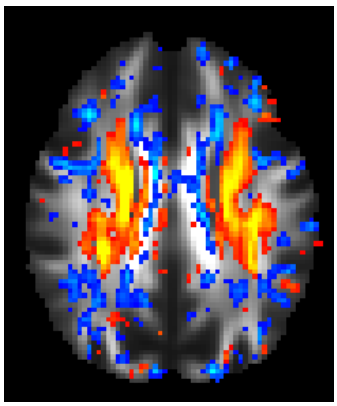
SCYFER
the science of prediction

Draft Architecture "*DxClassification*" API

Medical Imaging Classification



FICHe
Future Internet CHallenge eHealth

FI-WARE
Open APIs for Open Minds

Version: 0.1
Status: Draft
Date: 16-02-2015

Draft Architecture "*DxClassification*" API

Scyfer developed a Deep Learning toolset to train and classify medical images. Using FI-WARE and FI-STAR components this toolset will be turned into a closed API called *DxClassification*. This API can be used by designated medical institutes to analyse and classify medical images. Currently the Deep Learning toolset is capable of analysing 3D MRI scans and classifying neurodegenerative diseases. In the future the toolset will expand to support different types of images and disease classifications (e.g. lung and breast cancer or strokes). These expansions will become part of the *DxClassification* API.

Scyfer Deep Learning toolset

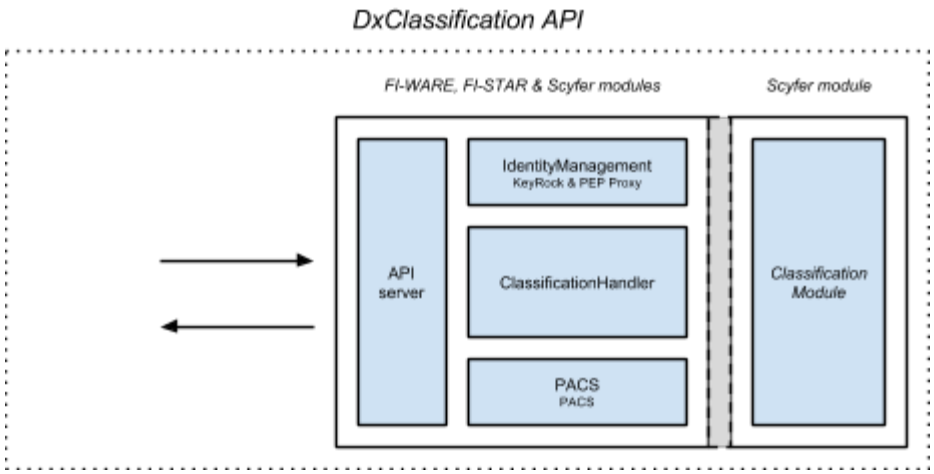
The toolset consists of a TrainingAlgorithm and a ClassificationModule. The algorithm is used during develoment and testing for creating the module. Only the module will be used for the API. [Explain in 2 sentences how the module works and processes millions of variables]. To deliver swift results, the module has to run on a heavy processing machine or separately on HPC instances. [Provide arguments why no data processing modules from FI-WARE are used]

The *DxClassification* API

The following modules make up the first version of the API:

- APIserver – Scyfer: A server providing RESTful API end-points through which hospital information systems, diagnostic systems or medical devices can request disease classifications by submitting (a collection of) images and meta data. This module will be written in either Java, Python or Node.js using open-source modules.
- IdentityManagement – FI-WARE KeyRock & PEP Proxy: User Interface to manage customer access to the API. Internally used by the APIserver and ClassificationHandler to determine if a request to the API is allowed and can be processed.
- ClassificationHandler – Scyfer: Internal logic to handle classification requests and results. Receives, normalises and stores images in the PACS, sends images and meta data to ClassificationModule, returns results to APIserver, registers requests, reports about customer API usage for billing. This module will be written in either Java, Python or Node.js and uses the database structure provided by KeyRock.
- PACS – FI-STAR PACS: Image archive for saving and retrieving medical images used for classification requests. This archive is also a requirement for compliance.

The following diagram shows the interdependency between the FI-WARE, FI-STAR and Scyfer modules:



Cloud & Private Infrastructure

The API architecture will be developed as a standalone software package which can be deployed at any cloud provider or local machine on a customer's premises. The API itself is relatively light weight – using a low-end machine or VM from any cloud provider the API should be able to handle hundreds of requests per second. The ClassificationModule processes images and classifies diseases based on millions of data-points. This requires parallel processing (GPU or CPU) and therefore a higher-end machine or VM. Due to their modularity, the API and ClassificationModule can run on s single machine or separate at two different machines either in the cloud or locally.

Usage of FI-WARE

- Data/Media Context Management: The API is relatively simple – it processes generic disease classification requests based on (a collection of) images and meta data. These requests are not based on context or specific events. The ClassificationModule takes care of the image processing and analyses millions of data points in these images to classify a possible disease. No database or external data source is needed to perform these classifications, the data is built into the ClassificationModule itself.
- Connection to the Internet of Things: Not relevant for this application.
- Application/Data Delivery: The first version of the API does not support custom reporting or mashups. It is expected when more customers and classifications are added to the system some form of custom reporting will become relevant and SpagoBI or Wirecloud would be a good solution to offer these functionalities.
- Advanced Web-based User Interface: To start lean no interfaces will be build into the API. It is expected when more customers and classifications are added to the system the requirements arise for 3D interfaces to display classifications inside scans. XML3D is an excellent tool to visualise diseases as it supports rendering of 3D models and features to interact with these models from a browser.
- Advanced middleware: Not relevant for this application.
- Robotics: Not relevant for this application.

- Security: As described above, the KeyRock and PEP Proxy will be used for Identity Management. Data handled by the system is anonymous and stored on local machines or cloud providers. The security solutions on these infrastructures are considered to be sufficient for the first version of the API.
- Cloud infrastructure: The core activities of Scyfer are focussed on developing and deploying Deep Learning solutions. Therefore all software will be run at existing cloud providers or at the customer's premises. It is expected when more customers and classifications are added to the system elasticity of the infrastructure becomes more important. As existing cloud providers like AWS, Rackspace or Microsoft have excellent orchestration tools these will be used to manage a scalable cloud infrastructure.
  The only module categorized as "Cloud infrastructure" that will be used for the development of the API is the FI-STAR PACS Specific Enabler. This module is a key element in running a medical image solution. The time for development, commercialisation and certification of the *DxClassification* API can be shortened tremendously using this module.