UMM
UNIVERSITÄTSMEDIZIN
MANNHEIM

Medizinische Fakultät Mannheim
der Universität Heidelberg

Universitätsklinikum Mannheim

Integrating Machine Learning Pipelines into Clinical Workflows with DICOM® Message Queues

Victor Saase[1], Andreas Junge[1], Holger Wenz MD PhD[1], Máté Maros MD[1]

Institution:

[1]Medical Faculty Mannheim, Heidelberg University, Mannheim, Germany

Problem:

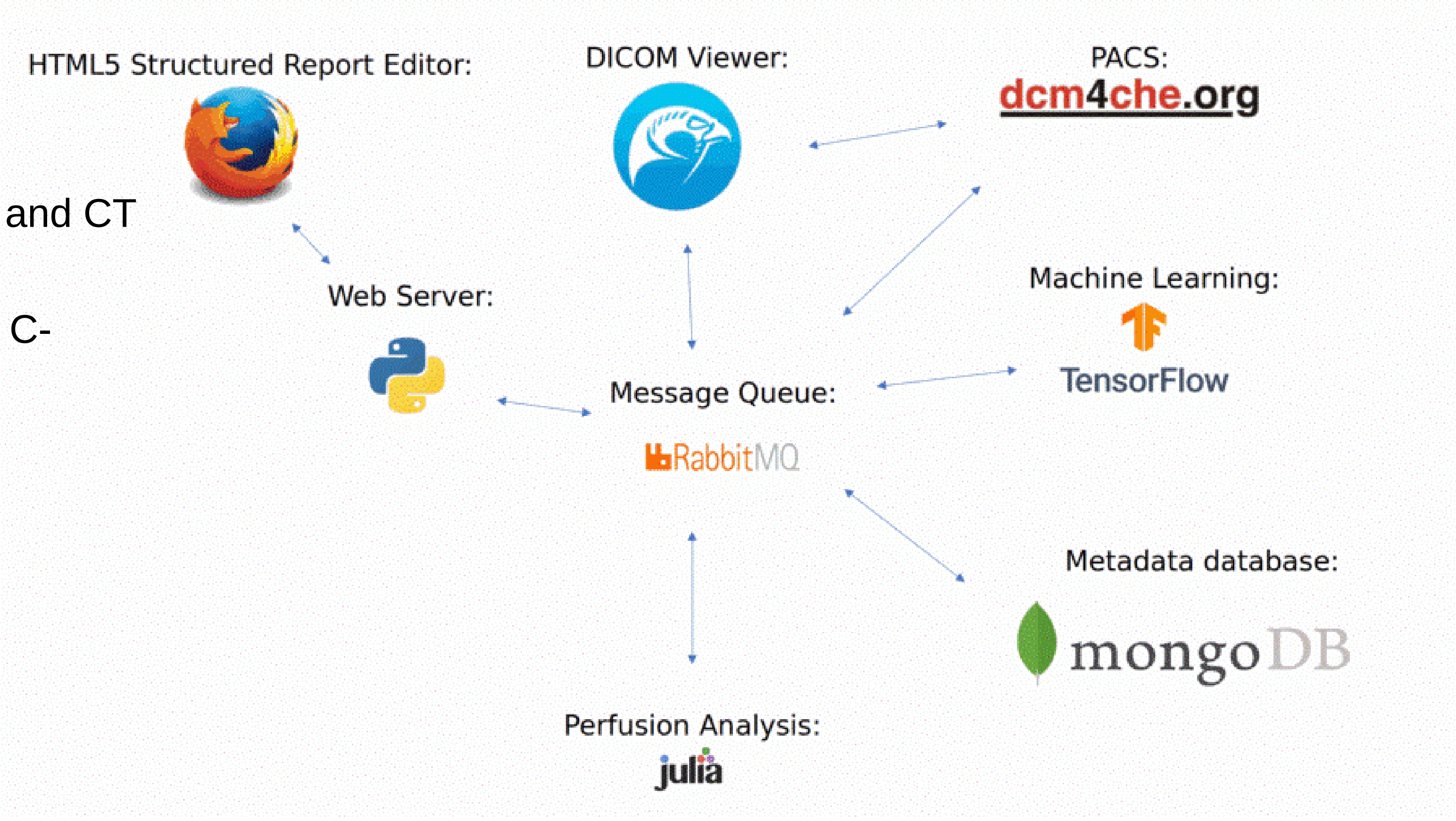
- Many Machine Learning prototypes are developed by independent researchers, but clinical adoption is lacking
- Companies have little interest in using a standard communication architecture, which leads to vendor lock-in and costly suppport contracts
- Radiologists lose control by using black box closed-source solutions which often fail without helpful error messages and any chance to recover

We present an open-source prototype that implements a modular, distributed and parallel analysis pipeline in MRI and CT reporting workflows of neuroradiology. The modular software architecture allows us to quickly exchange parts of the pipeline for experiments and testing while ensuring stability and performance of the remaining system.

We use DICOM® metadata as messages for the standardized communication between modules. Modules can be implemented in various programming languages (at the moment libraries for JavaScript, Python and Julia are available), thus enabling us to choose the best tools for the specific requirements and lowering the entry barrier for developers. Modules can be distributed across the network, making use of dedicated hardware, thereby reducing performance and data transmission bottlenecks.

Modules publicly available at the moment:

- Python
  - automatic 3D registration/reconstruction for MRI and CT
  - convert DICOM® to NIfTI and back
  - PACS clients with C-FIND, C-STORE, C-MOVE, C-GET, WADO
  - MongoDB connection
  - open study in Horos viewer
- Julia
  - experimental DSC-MRI perfusion analysis
- JavaScript
  - study browser
  - report editor



Example workflow illustrating the concept:

- AI-assisted structured reporting
  - User enters patient query string in web-based study browser
  - Query message “find.studies” is published to message queue in DICOM® binary format
  - PACS connection module listens on queue, receives query, communicates with PACS and publishes “found.study” messages
  - Study browser lists found studies
  - User selects study, triggering “get.study” message
  - Study is downloaded by PACS client module, which publishes “stored.study” message with the URI where the DICOM® files can be found locally
  - Viewer module sends study to viewer and automatically opens study
  - in parallel AI-modules process the data which conform to their specific DICOM® based input filters, triggering further “stored.series” messages which can be processed by other modules
  - Viewer module, which still listens for “stored.study” and “stored.series” messages, automatically sends AI-created images to viewer
  - Web-based report editor can process messages to guide the creation of automatic structured report templates, incorporating information about MRI sequence series published on the message queue and structured reports in DICOM® format created by AI-modules
  - User edits report in the editor, potentially triggering more messages for AI-modules
  - Final report is published by the user as “stored.instance” message in DICOM® SR format and uploaded by PACS module

The software is available in open source at github.com/dcmq
Contributions are welcome.

DICOM® is the registered trademark of the National Electrical Manufacturers Association for its standards publications relating to digital communications of medical information.