Image Management Using Orthanc in a Multi-Centre Research PET Study

OrthancCon I, 2019
Andrew Crabb
The Johns Hopkins University
I Do Imaging
Overview

• We are a Radiology lab running multiple studies, all different
• Five-year research neuro PET pharmaceutical imaging study
• Four US academic medical centres
• 600 subjects, 2 scans each
• Images uploaded by scanner techs or Radiology admins
• Research subjects: No Protected Health Information entered
• High variability: Subject codes, sites, scanners, staff
• Some scanners could not have a DICOM node configured
• Some sites insist on encrypted transport
Neuro PET
Requirements

- A small, flexible solution with quick deployment
- Stability – no memory leaks, restart on crash/reboot
- Scalable to ~ TB scale
- Client upload via DICOM transport and Web interface
  - Not every DICOM modality can be configured for our node
- Security – Encrypted transport
  - Potential to achieve HIPAA compliance
- Client multi language/environment support
- Hierarchical development environment
Orthanc

- Small
- Flexible
- Quick deployment
- Stable
- Scalable
- Upload via DICOM & Web
- Multi language

Runs easily in 2 GB, 1 core instance
Plain text config files
DICOM nodes in separate config
Compiles and installs easily on AWS Linux
Simple directory/symlink config files
orthanc and nginx daemons are stable
Auto restart with supervisor
orthanc easily handles our scale
Images stored in external DB

Language-independent with REST

Orthanc

We develop & test with curl & Postman
Clients in Python and Matlab
Why a Cloud Solution?

• Problems with DICOM transport:
  • Not (easily) encrypted over public internet
  • Ports (104 or custom) usually blocked
  • C-MOVE not possible behind NAT

• Cloud location is available from anywhere
  • Can allow only certain IP ranges

• Many features are easily deployed
  • Encryption of storage and transport
  • Backups of database instances
  • Logging of users and actions
Rapid Prototyping: Postman
HIPAA Compliance

Requires (amongst others):

- Encryption of data in transport
- Encryption of data at rest
- Access control
- Logging of all actions
- Configuration monitoring
- Verified backup/archive procedure
Possible HIPAA-Compliant Implementation

Encryption of data in transport
Encryption of data at rest
Access control
Logging of all actions
Configuration monitoring
Verified backup/archive procedure
## Clinical Medicine vs. Medical Research

<table>
<thead>
<tr>
<th>Clinical</th>
<th>Research</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read individually, now</td>
<td>Processed in batch, later</td>
</tr>
<tr>
<td>Read by humans</td>
<td>Processed automatically</td>
</tr>
<tr>
<td>Immediate reading</td>
<td>Delayed processing</td>
</tr>
</tbody>
</table>

- **Clinical**:
  - William / Will
  - William / Wiliam
  - Sarah / Sara
  - A010 / A010

- **Research**:
  - Nicknames
  - Misspelling
  - Alternate spellings
  - Spoken digits
Every human process includes random error

Medicine has a strong culture of reducing and correcting error
Automation achieves zero ambiguity by accommodating error
Why not just fix the errors?

- **Difficult**
  - Nuclear backups immediately propagated
  - PACS systems have limited access
  - Slow, heavyweight process
  - No history (provenance) of edits
  - Only conforms to one view of correctness

‘Fixing’ is an attempt to say that something did not happen, that did happen

The ‘fixing’ process itself will contain error

idoimaging.com/turku2017
What is the correct identity of this subject? It depends on who you ask.
A person enrolled in a study and who receives a scan has three identities:

<table>
<thead>
<tr>
<th>Identity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Personal Identity</td>
<td>Name, birthdate, hospital history number</td>
</tr>
<tr>
<td>Study Identity</td>
<td>Name or code wished to be used at scan time</td>
</tr>
<tr>
<td>Scan Identity</td>
<td>Name and history actually used at scan time</td>
</tr>
</tbody>
</table>
Eliminating Error in Software Development

Lightweight
Iterative
Version control

Code
Test

Input
Run
Output

Truth
Deterministic
Replicable
Correct

✔
✘
Accommodating Error in Identity

Lightweight
Iterative
Individual
Version control

Mapping

Test

Images

Process

Output

Truth

Deterministic

Replicable Correct

Correct Deterministic

✔

✘