Recent Developments in Free Medical Imaging Software

OrthancCon I, 2019

Andrew Crabb

The Johns Hopkins University

I Do Imaging

Why Free Medical Imaging Software?

Why Use It?

Medical imaging is well-served by free software

Benefits from collaborative imaging community

Source code often available

Can address specialist/niche/research needs

Why Write It?

Recognition and publicity

Free testing by demanding users

Contributions and improvements

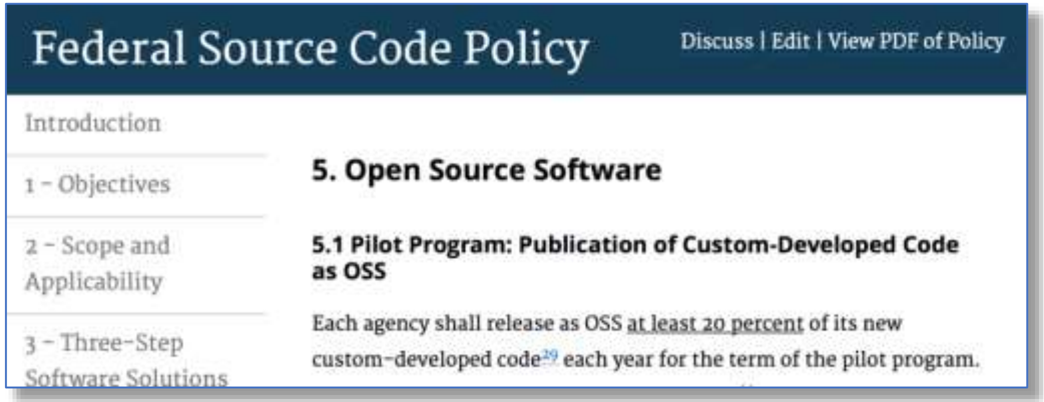
Sometimes required by sponsor

Imaging software is competing for the user’s most valuable asset: *time*

Today’s users are accustomed to high-quality free software

Many imaging areas are served by multiple free applications

Only the best software becomes self-sustaining

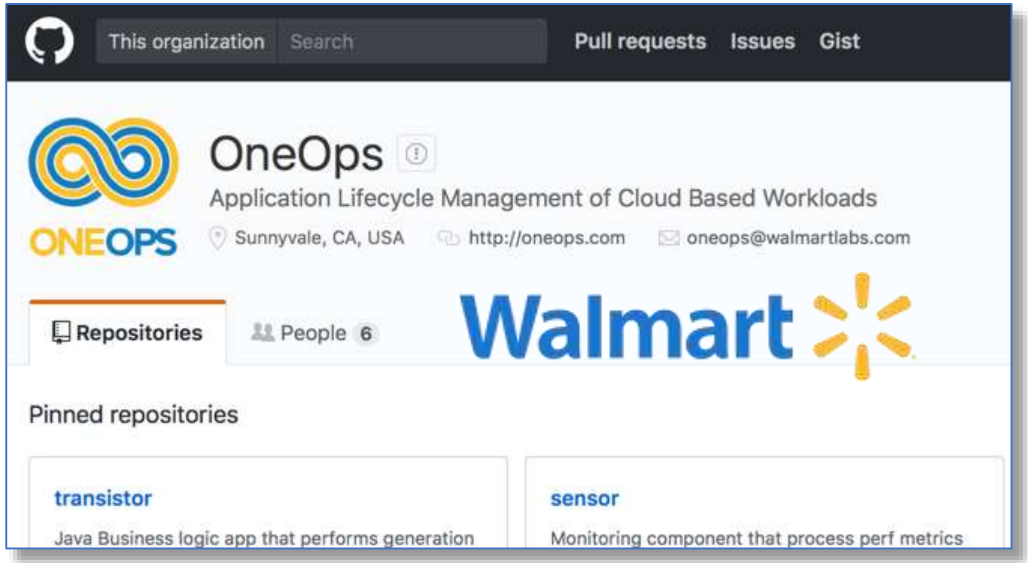

Discuss | Edit | View PDF of Policy
Introduction
1 - Objectives
2 - Scope and Applicability
3 - Three-Step Software Solutions

Federal Source Code Policy

5. Open Source Software

5.1 Pilot Program: Publication of Custom-Developed Code as OSS

Each agency shall release as OSS at least 20 percent of its new custom-developed code[29] each year for the term of the pilot program.



OneOps

Application Lifecycle Management of Cloud Based Workloads

Sunnyvale, CA, USA http://oneops.com oneops@walmartlabs.com

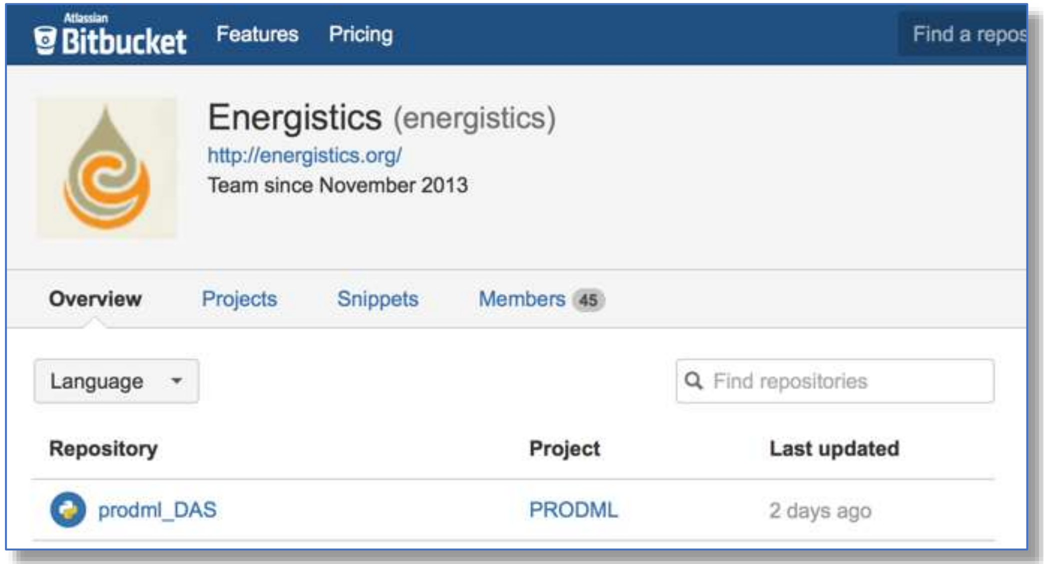
Pinned repositories

transistor

Java Business logic app that performs generation

sensor

Monitoring component that process perf metrics

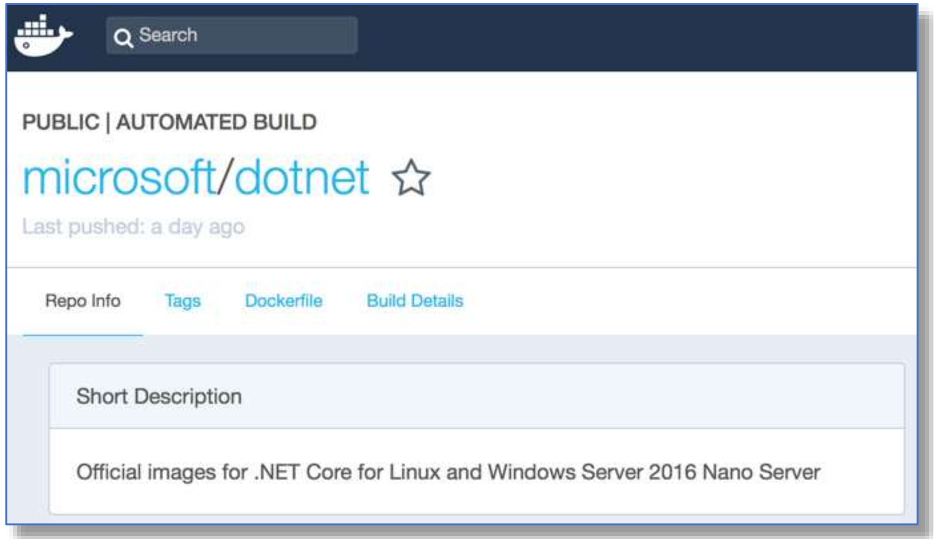


Energistics (energistics)

http://energistics.org/

Team since November 2013

Repository	Project	Last updated
prodml_DAS	PRODML	2 days ago



PUBLIC | AUTOMATED BUILD

microsoft/dotnet ☆

Last pushed: a day ago

Short Description

Official images for .NET Core for Linux and Windows Server 2016 Nano Server

Distributions

Source

GitHub/BitBucket repo

- `hg clone bitbucket.org/sjodogne/orthanc`

Platform Specific

HomeBrew (Mac)

- `brew install dcmtk`

apt/yum (Linux)

- `apt-get install python-dicom`

zypper (openSUSE)

- `zypper install orthanc`

Chocolatey (Windows)

Virtual Machines

Docker/DockerHub

- `docker run jodogne/orthanc`

Vagrant/VirtualBox

- `git clone xnat.git; ./run xnat setup`

Language Specific

Pip (Python)

- `pip search nifti     # (12 results)`

npm/yarn (Node JS)

- `npm search dicom     # (24 results)`

DICOM Libraries

DCMTK (OFFIS)

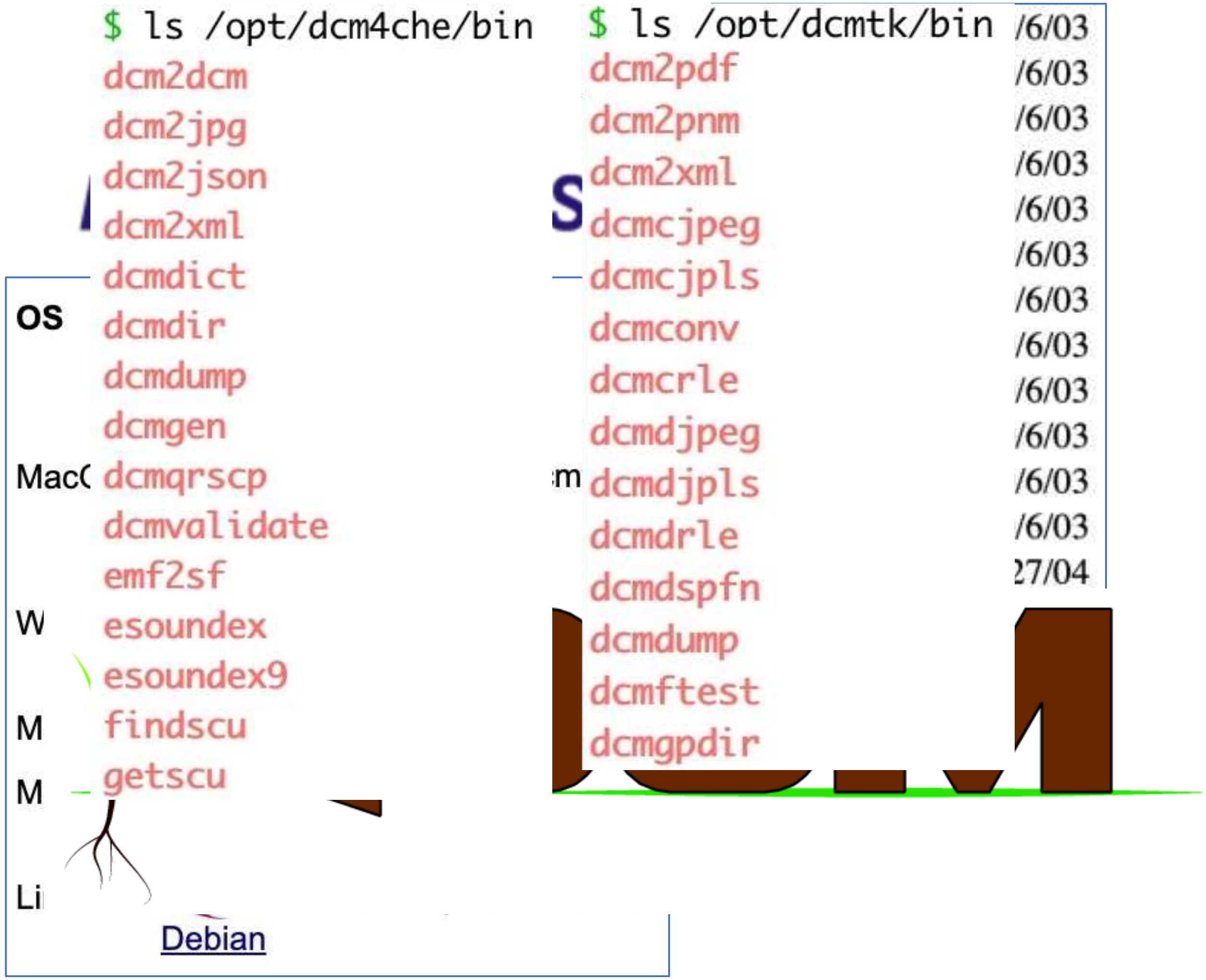
- C++ ‘reference’ DICOM library
- Steady enhancements since 2003
- Command line utilities

dcm4che (dcm4che.org)

- Java DICOM toolkit since ca. 2000
- Many command line applications
- Adding DICOMWeb capabilities

GDCM (Mathieu Malaterre)

- Grassroots DICOM
- C++, binds to Python, C#, Java, PHP
- SCU network operations



```
$ ls /opt/dcm4che/bin
dcm2dcm
dcm2jpg
dcm2json
dcm2xml
dcmdict
dcmdir
dcmdump
dcmgen
dcmqrscp
dcmvalidate
emf2sf
esoundex
esoundex9
findscu
getscu
```

```
$ ls /opt/dcmtk/bin
dcm2pdf
dcm2pnm
dcm2xml
dcmcjpeg
dcmcjpls
dcmconv
dcmcrle
dcmdjpeg
dcmdjpls
dcmdrle
dcmdspfn
dcmdump
dcmftest
dcmgpdir
```

DICOM Libraries

dicomParser (Cornerstone Project)

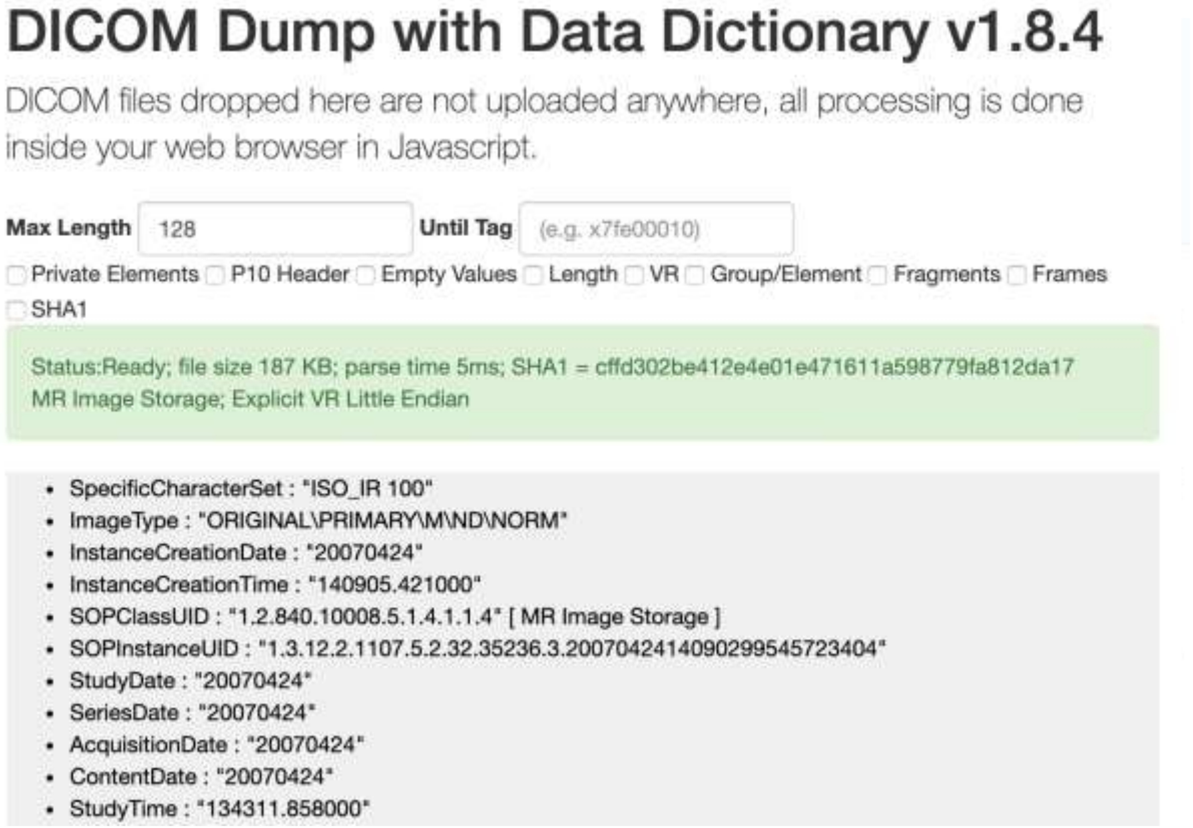
- Lightweight JavaScript library for parsing DICOM byte streams
- For HTML5 browsers, Node, Meteor

pydicom (Darcy Mason)

- Pure Python library, no dependencies
- Read pixel data with NumPy, Pillow

Ruby DICOM (Christoffer Lervåg)

- Full Ruby DICOM implementation



DICOM Dump with Data Dictionary v1.8.4

DICOM files dropped here are not uploaded anywhere, all processing is done inside your web browser in Javascript.

Max Length 128 **Until Tag** (e.g. x7fe00010)

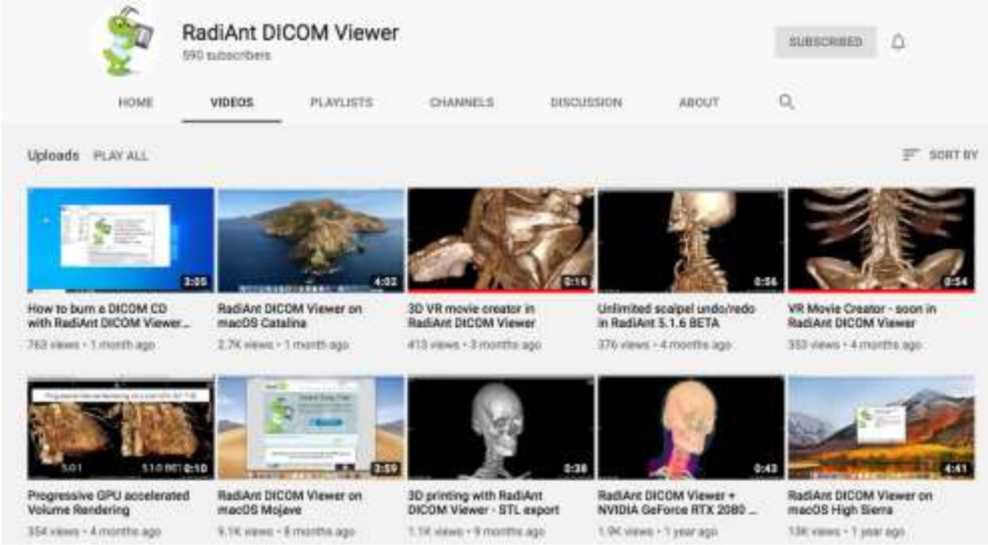
□ Private Elements □ P10 Header □ Empty Values □ Length □ VR □ Group/Element □ Fragments □ Frames □ SHA1

```
Status:Ready; file size 187 KB; parse time 5ms; SHA1 = cffd302be412e4e01e471611a598779fa812da17
MR Image Storage; Explicit VR Little Endian
```

```
• SpecificCharacterSet : "ISO_IR 100"
• ImageType : "ORIGINAL\PRIMARY\M\ND\NORM"
• InstanceCreationDate : "20070424"
• InstanceCreationTime : "140905.421000"
• SOPClassUID : "1.2.840.10008.5.1.4.1.1.4" [ MR Image Storage ]
• SOPInstanceUID : "1.3.12.2.1107.5.2.32.35236.3.2007042414090299545723404"
• StudyDate : "20070424"
• SeriesDate : "20070424"
• AcquisitionDate : "20070424"
• ContentDate : "20070424"
• StudyTime : "134311.858000"
```

RadiAnt

- Windows DICOM viewer from Poland
- Steady enhancements since its introduction
- Multiplanar reconstruction
- PET-CT image fusion
- High performance GPU-based 3D rendering
- PACS query/send/retrieve



RadiAnt DICOM Viewer

590 subscribers

Uploads

How to burn a DICOM CD with RadiAnt DICOM Viewer...
763 views • 1 month ago

RadiAnt DICOM Viewer on macOS Catalina
2.7K views • 1 month ago

3D VR movie creator in RadiAnt DICOM Viewer
413 views • 3 months ago

Unlimited scalpel undo/redo in RadiAnt 5.1.6 BETA
376 views • 4 months ago

VR Movie Creator - soon in RadiAnt DICOM Viewer
353 views • 4 months ago

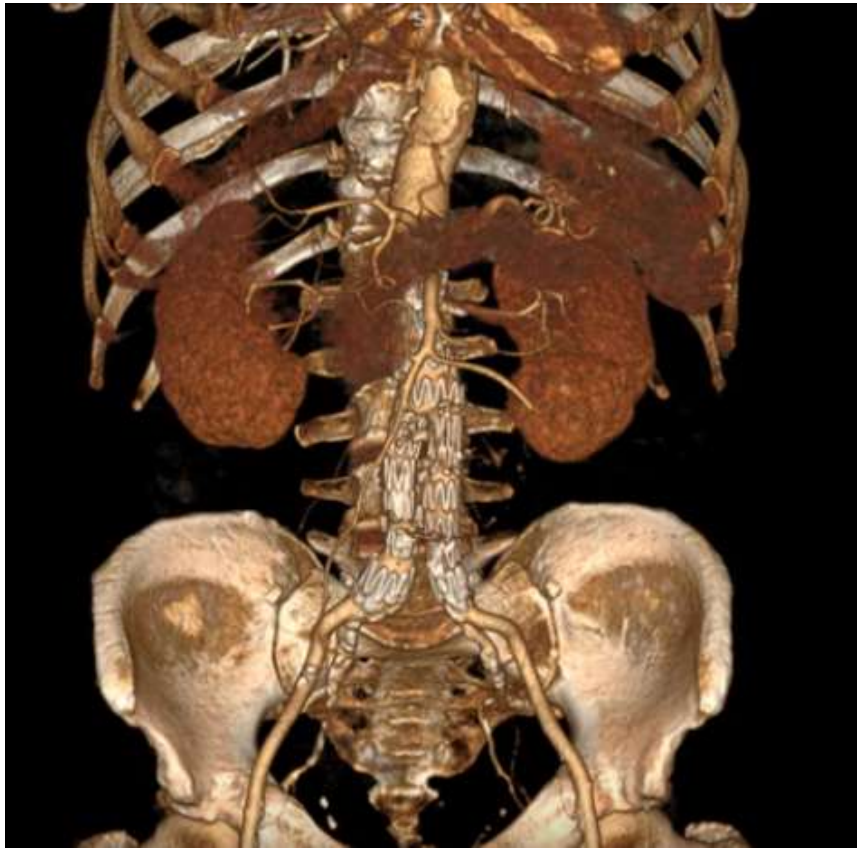
Progressive GPU accelerated Volume Rendering
354 views • 4 months ago

RadiAnt DICOM Viewer on macOS Mojave
9.1K views • 8 months ago

3D printing with RadiAnt DICOM Viewer - STL export
1.1K views • 9 months ago

RadiAnt DICOM Viewer + NVIDIA GeForce RTX 2080 ...
1.9K views • 1 year ago

RadiAnt DICOM Viewer on macOS High Sierra
13K views • 1 year ago



Horos

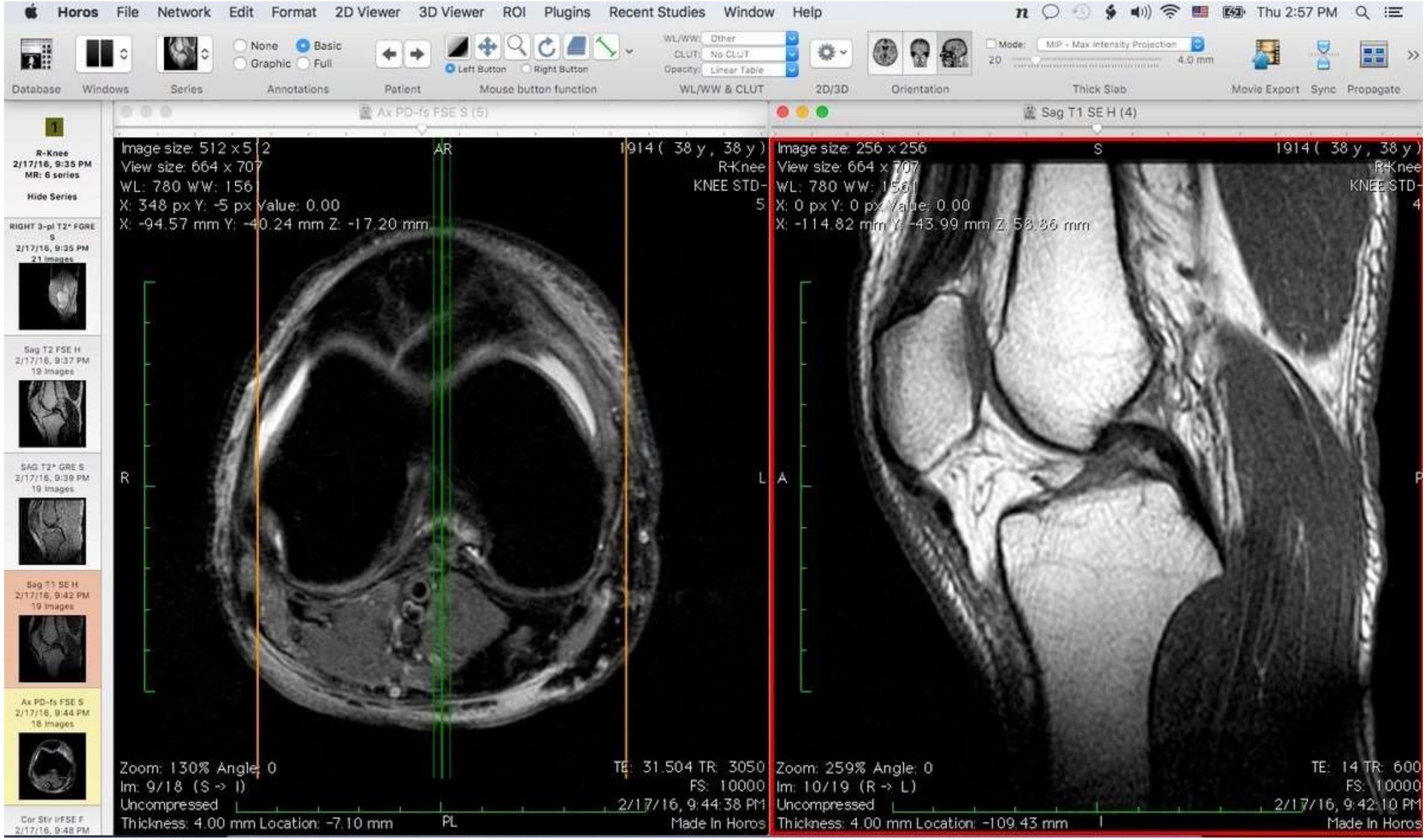


Image size: 512 x 512
View size: 664 x 707
WL: 780 WW: 1561
X: 348 px Y: -5 px Value: 0.00
X: -94.57 mm Y: -40.24 mm Z: -17.20 mm

1914 (38 y , 38 y)
R-Knee
KNEE STD-
5

Zoom: 130% Angle: 0
Im: 9/18 (S -> I)
Uncompressed
Thickness: 4.00 mm Location: -7.10 mm

TE: 31.504 TR: 3050
FS: 10000
2/17/16, 9:44:38 PM
Made In Horos

Image size: 256 x 256
View size: 664 x 707
WL: 780 WW: 1561
X: 0 px Y: 0 px Value: 0.00
X: -114.82 mm Y: -43.99 mm Z: 58.86 mm

1914 (38 y , 38 y)
R-Knee
KNEE STD-
4

Zoom: 259% Angle: 0
Im: 10/19 (R -> L)
Uncompressed
Thickness: 4.00 mm Location: -109.43 mm

TE: 14 TR: 600
FS: 10000
2/17/16, 9:42:10 PM
Made In Horos

OsiriX™

OpenJPEG

OpenGL

VTK

ITK

DCMTK

GDCM

Weasis

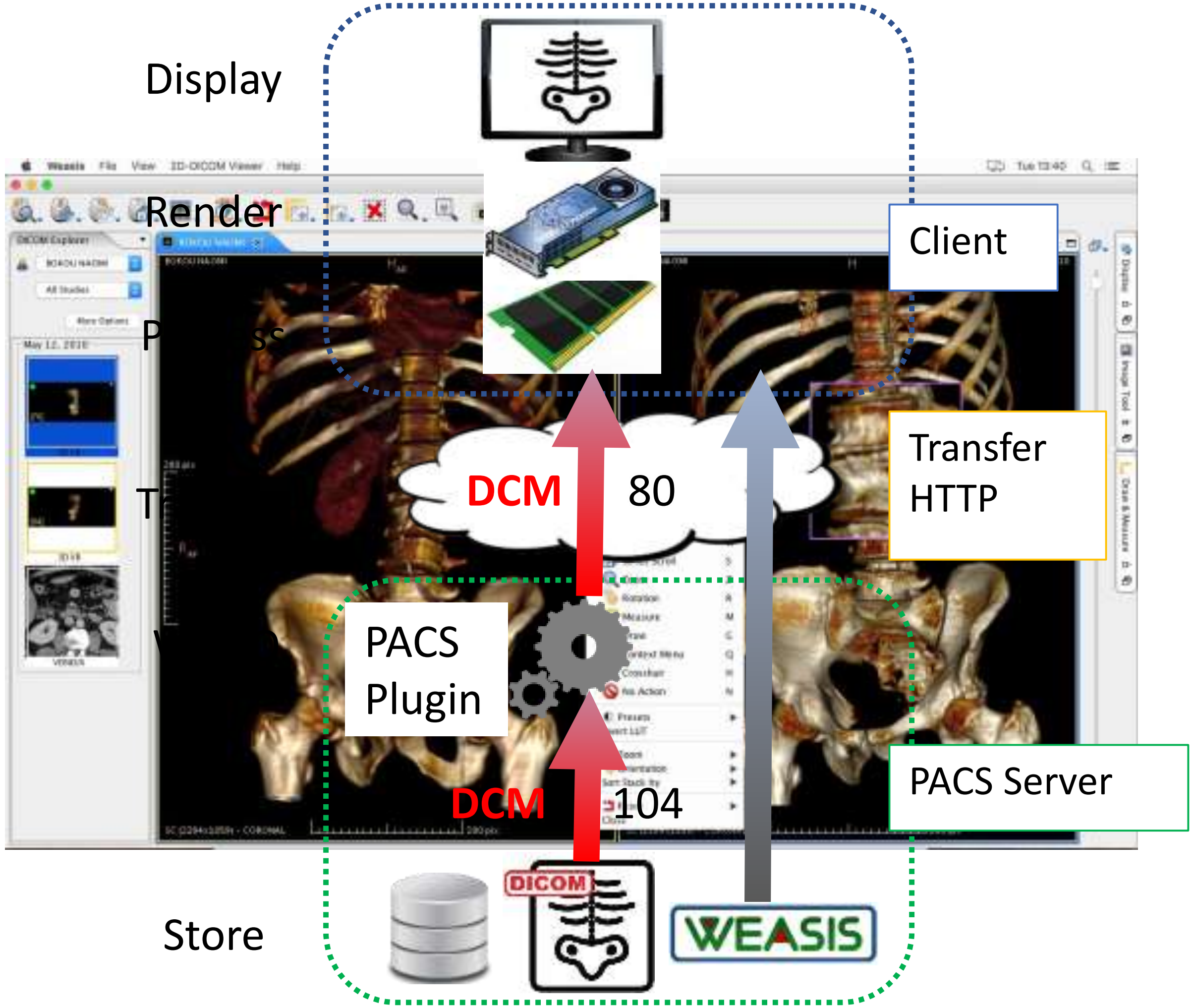
Long-term project (Nicolas Roduit)

Desktop java imaging, PACS deployment

Web access using weasis:// protocol

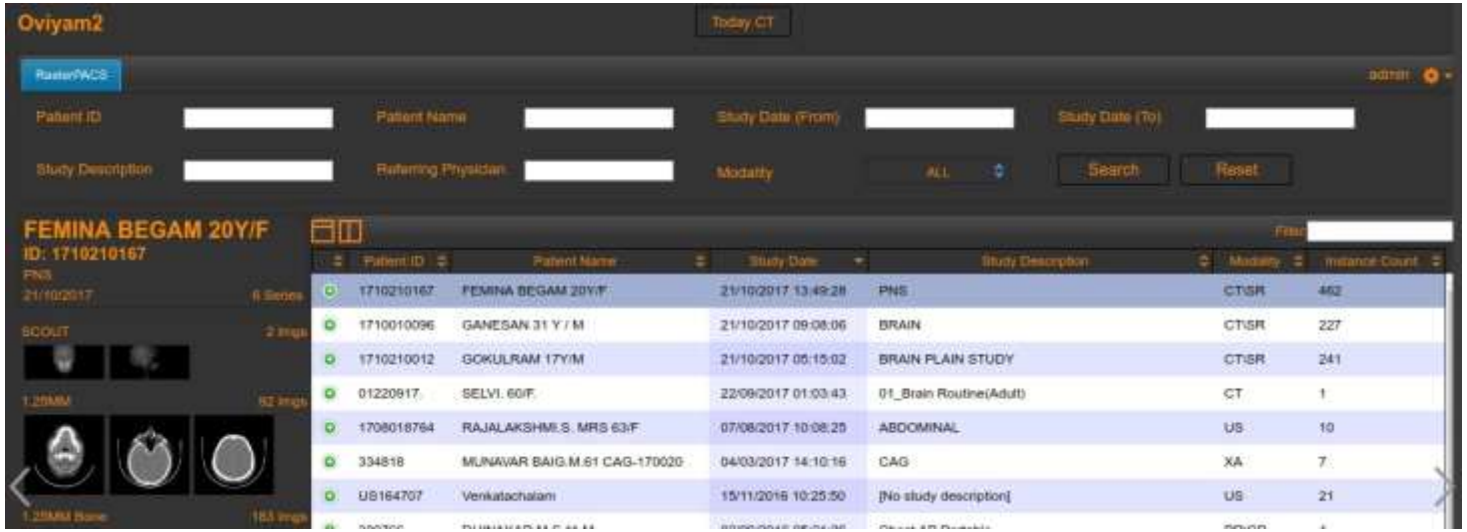
DICOM send, query, retrieve

DICOMWeb capabilities (Orthanc, DCM4CHEE)



Oviyam (Raster Images)

- Web-based DICOM viewer
- Fronts any DICOM server with WADO
- Displays images as JPG in browser



Oviyam2 Today CT

RasterPACS admin

Patient ID | Patient Name | Study Date (From) | Study Date (To)

Study Description | Referring Physician | Modality ALL | Search | Reset

FEMINA BEGAM 20Y/F
ID: 1710210167
PNS
21/10/2017 6 Series

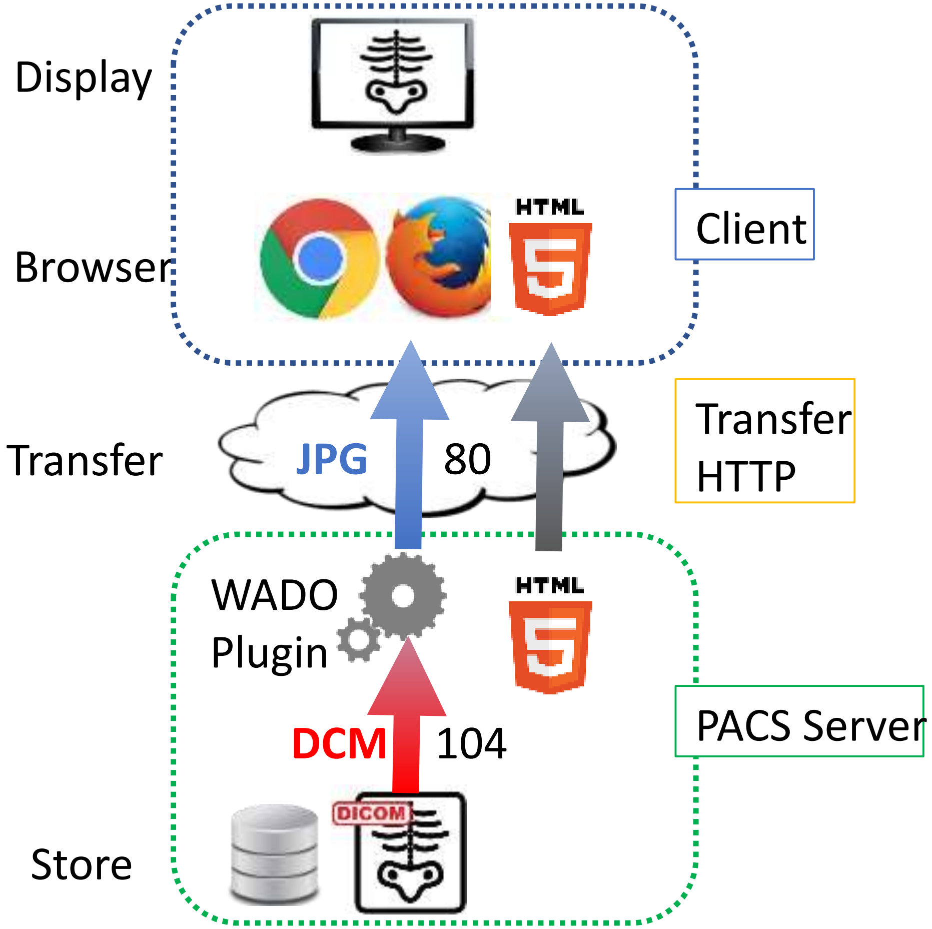
SCOUT 2 imgs

1.25MM 92 imgs

1.25MM Bone 183 imgs

Filter

Patient ID	Patient Name	Study Date	Study Description	Modality	Instance Count
1710210167	FEMINA BEGAM 20Y/F	21/10/2017 13:49:28	PNS	CT\SR	462
1710010096	GANESAN 31 Y / M	21/10/2017 09:08:06	BRAIN	CT\SR	227
1710210012	GOKULRAM 17Y/M	21/10/2017 06:15:02	BRAIN PLAIN STUDY	CT\SR	241
01220917	SELVI. 60/F	22/09/2017 01:03:43	01_Brain Routine(Adult)	CT	1
1708018764	RAJALAKSHMI.S. MRS 63/F	07/08/2017 10:08:25	ABDOMINAL	US	10
334818	MUNAVAR BAIG.M.61 CAG-170020	04/03/2017 14:10:16	CAG	XA	7
US164707	Venkatachalam	15/11/2016 10:25:50	[No study description]	US	21



Trends

DICOMWeb

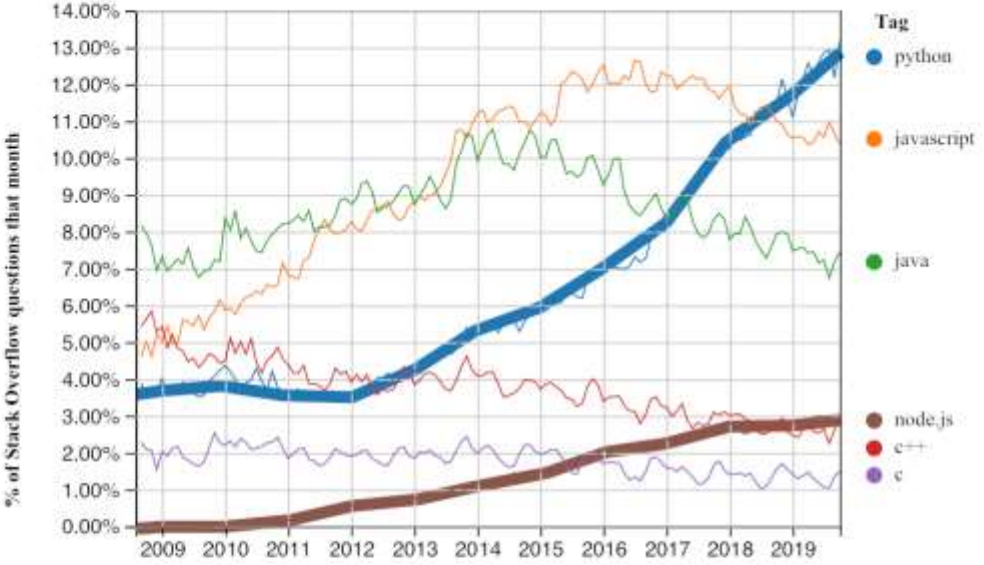
Cloud

JavaScript

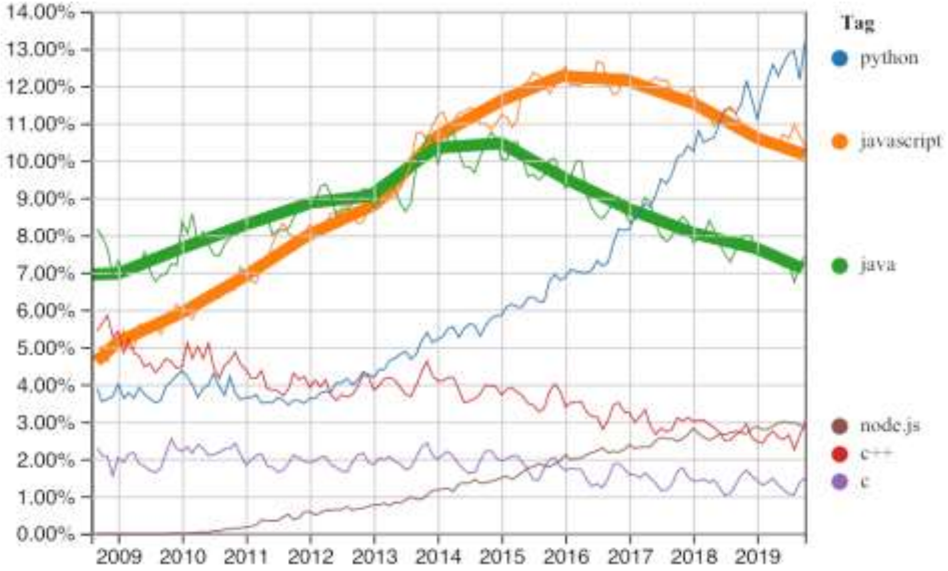
Python

Language Trends

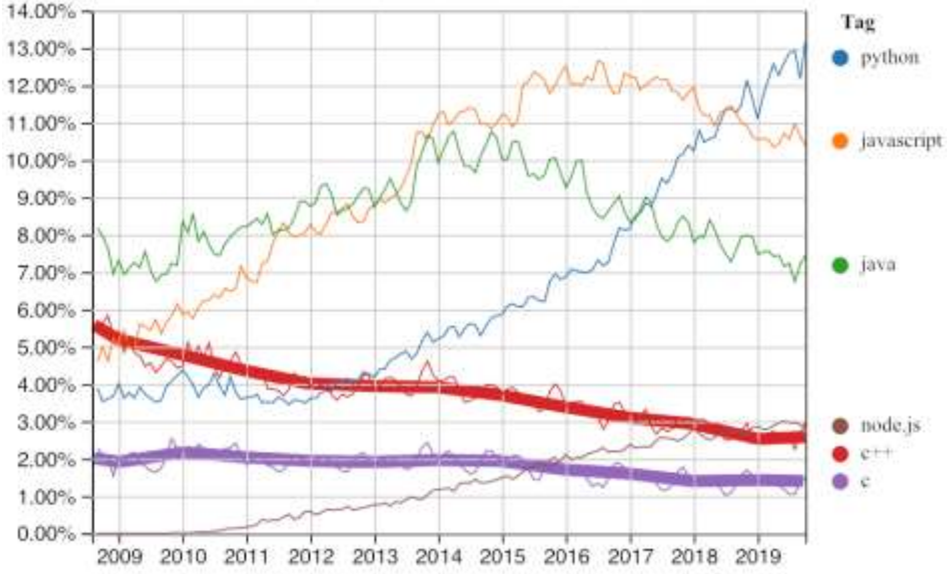
Trending:
Python, Node



Post-tending:
Javascript, Java

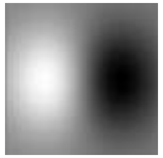


Steady:
C++, C



Lua

```
sitk = SimpleITK
gauss = sitk.GaussianSource (size, sigma,
center);
deriv = sitk.Derivative(gauss);
```



```
function IncomingHttpRequestFilter(method, uri, ip, u
   -- Only allow GET requests for non-admin users
   if method == 'GET' then
      return true
   elseif username == ‘heyitsme' then
```

```
-- @usage nmap -p4242 --script dicom-ping <target>
--
-- @output
-- PORT     STATE SERVICE REASON
-- 4242/tcp open  dicom   syn-ack
-- | dicom-ping:
-- |   dicom: DICOM Service Provider discovered!
```

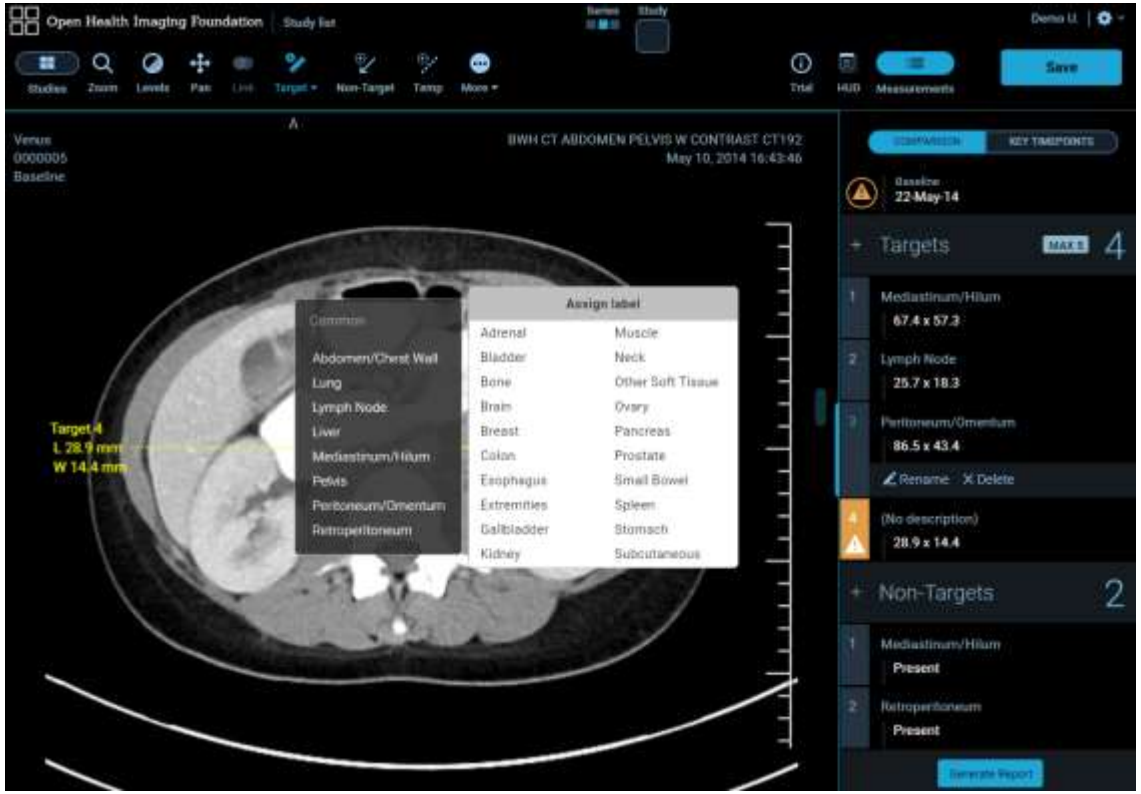
Cornerstone (Chris Hafey)

- Javascript library for building interactive image viewers
- Display in web browsers using HTML5 Canvas
- Independent of image container, transport
- Not constrained to an interaction paradigm
- Format-specific image loaders:
  - WADO Image Loader (WADO-RS)
  - Web Image Loader

OHIF Viewer

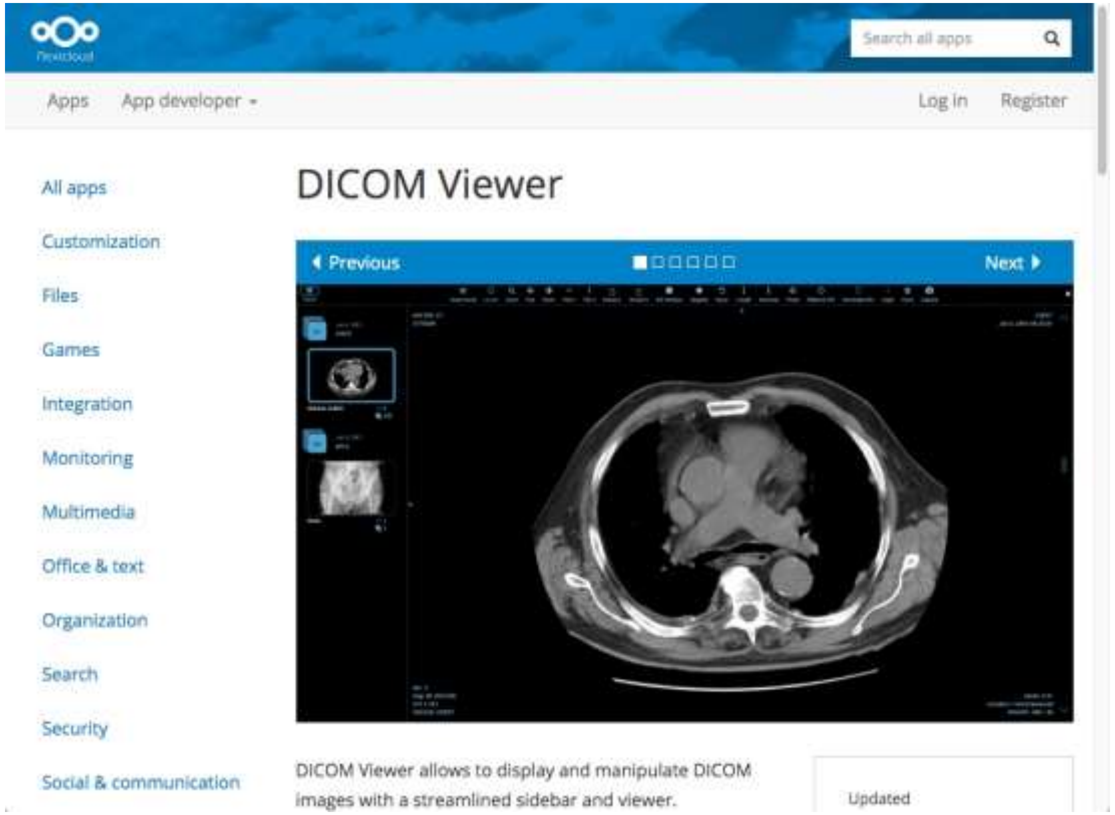
Open Health Imaging Foundation

- Zero-footprint DICOM viewer
- Supports DICOMWeb
- Extensions for: Cornerstone, microscopy, VTK



Nextcloud DICOM viewer

Nextcloud

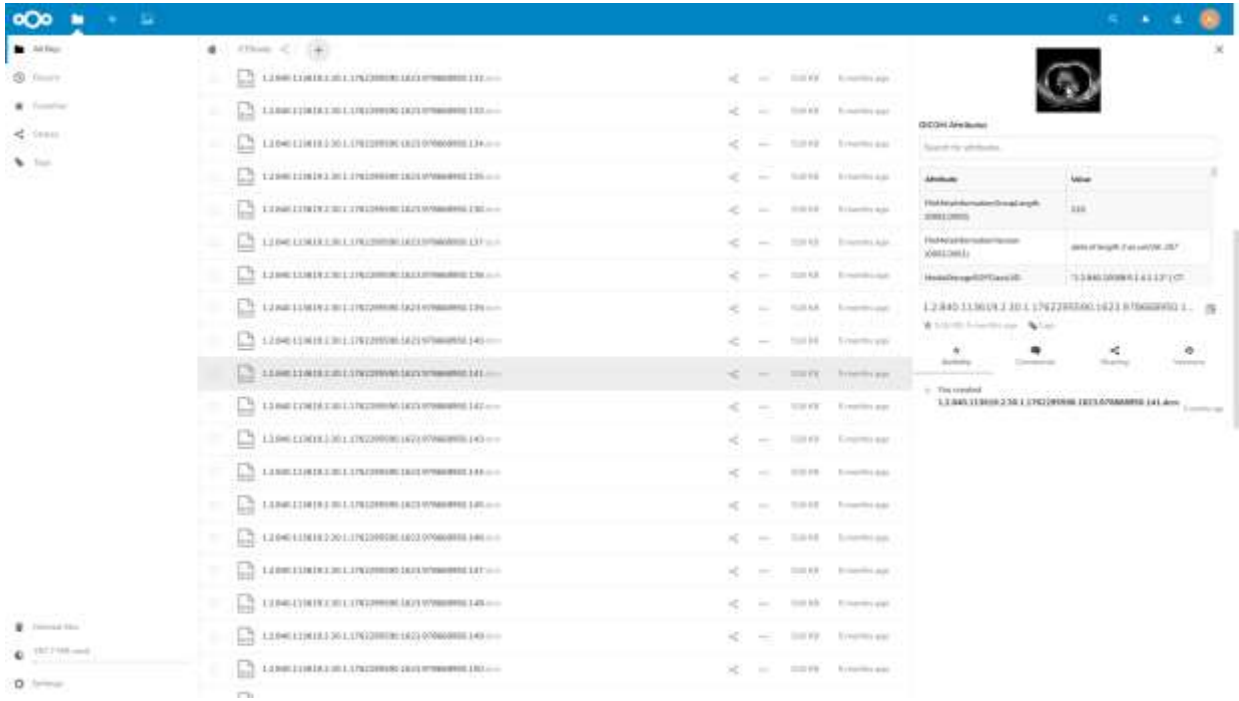


All apps
Customization
Files
Games
Integration
Monitoring
Multimedia
Office & text
Organization
Search
Security
Social & communication

DICOM Viewer

DICOM Viewer allows to display and manipulate DICOM images with a streamlined sidebar and viewer.

Updated



VTK-JS and ITK-JS (Kitware)

JavaScript ports of foundation packages used in dozens of imaging applications

VTK-JS

VTK: Visualization Toolkit

3D scientific data manipulation and visualization

JavaScript port of core VTK features

3D data rendering in a browser

Coding similar to VTK C++/Python

ITK-JS

ITK Insight Toolkit

Image analysis library in C++

Compiled to asm.js and WebAssembly

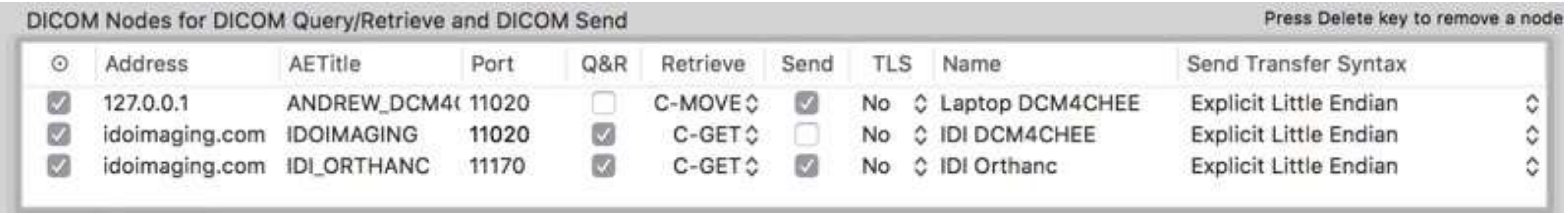
Spatial analysis in a browser or Node.js

Supports all file formats of ITK

DICOMWeb

- DICOM: Proprietary transport
- Nodes identified by three fields:
  - Address (DNS or IP)
  - Application Entity (AE) Title – 16 character string
  - Port number (Port 104 reserved for DICOM)

C-FIND	“Locate study matching these criteria”
C-GET <uid>	”Send this study to my IP”
C-MOVE <uid> <source AET> <dest AET>	“Send this study from source to dest”
C-STORE <data>	“Store this study/series”



DICOM Nodes for DICOM Query/Retrieve and DICOM Send

Press Delete key to remove a node

⊙	Address	AETitle	Port	Q&R	Retrieve	Send	TLS	Name	Send Transfer Syntax
☑	127.0.0.1	ANDREW_DCM4(	11020	□	C-MOVE ◇	☑	No ◇	Laptop DCM4CHEE	Explicit Little Endian ◇
☑	idoimaging.com	IDOIMAGING	11020	☑	C-GET ◇	□	No ◇	IDI DCM4CHEE	Explicit Little Endian ◇
☑	idoimaging.com	IDI_ORTHANC	11170	☑	C-GET ◇	☑	No ◇	IDI Orthanc	Explicit Little Endian ◇

DICOMWeb

Provides standard REST interface to DICOM image store

Removes much of the complexity of DICOM transport

	DICOM	**DICOMweb**	

Query	C-FIND	QIDO-RS
		Query on ID for DICOM objects
			`GET /studies?...`
			`GET /studies/id/series?`

Retrieve	C-GET	WADO-RS
		Web access of DICOM objects
			`GET /studies/id`
			`GET /studies/id/series/id`

Store	C-STORE	STOW-RS Store over the web	`POST /studies/id`

DICOMWeb Servers

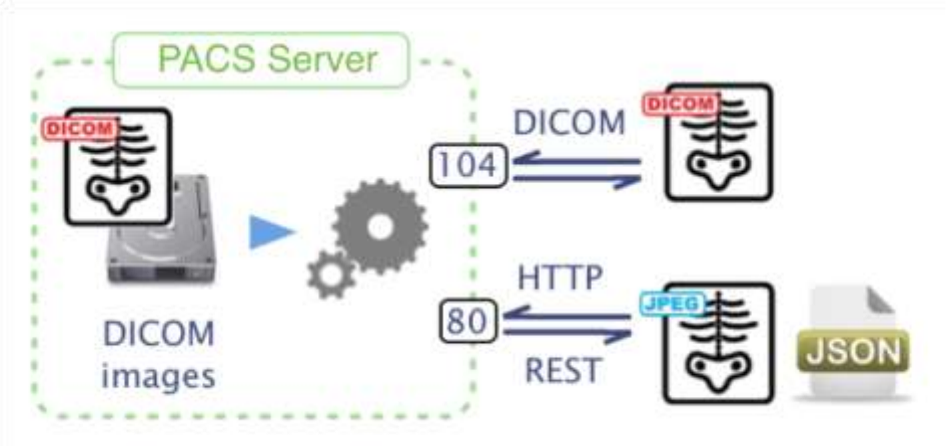
DCM4CHEE (dcm4che.org)

Comprehensive DICOM archive in Java

Full implementation of DICOM standard + HL7

Requires Wildfly (JBoss), database, LDAP

Available on Docker containers



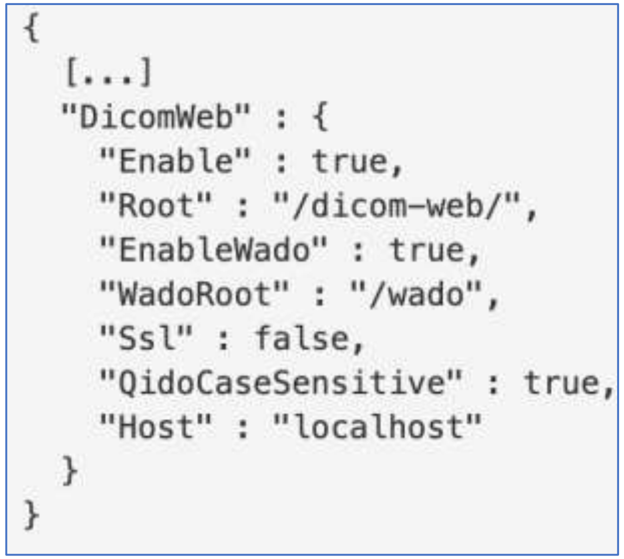
ORTHANC

Orthanc Book   Content

DICOMweb plugin

Contents

- DICOMweb plugin
  - Compilation
  - Installation
  - Options



```
{
  [...]
  "DicomWeb" : {
    "Enable" : true,
    "Root" : "/dicom-web/",
    "EnableWado" : true,
    "WadoRoot" : "/wado",
    "Ssl" : false,
    "QidoCaseSensitive" : true,
    "Host" : "localhost"
  }
}
```

Dicoogle (University of Aveiro)

Dicoogle

Platform-independent PACS (Java)

Implemented on dcm4che

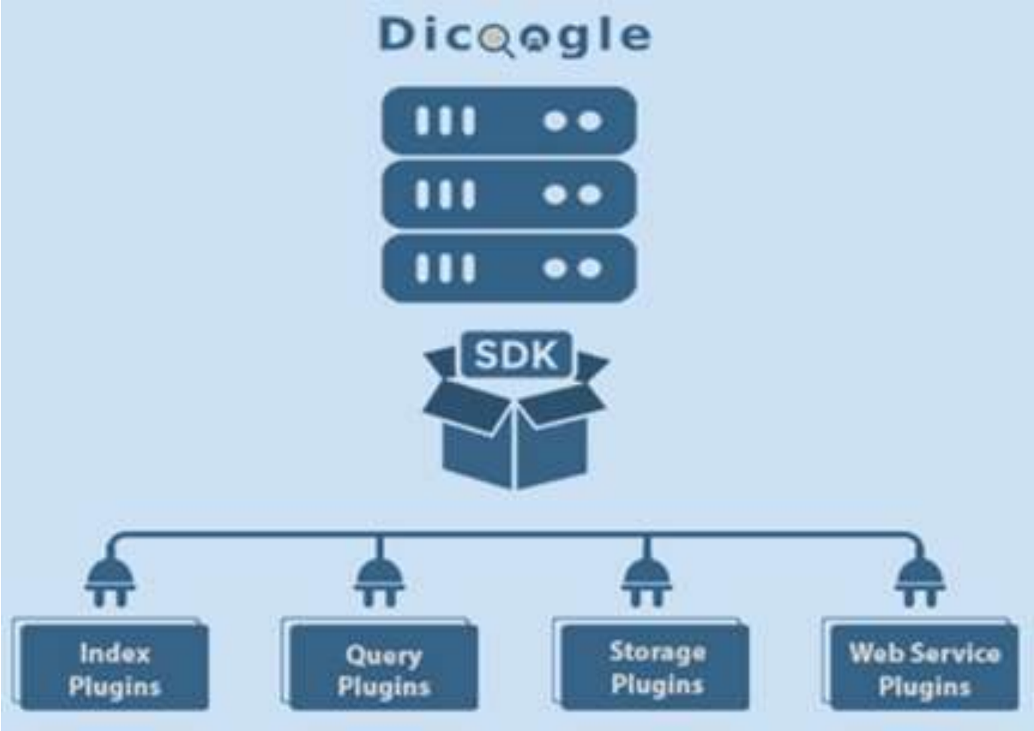
Highly modular: Major functions (store/index/search) by plugins

WebUI plugins: Front-end pluggable components in JS

Implements DICOMWeb

Complex querying: Free text

Strong developer support



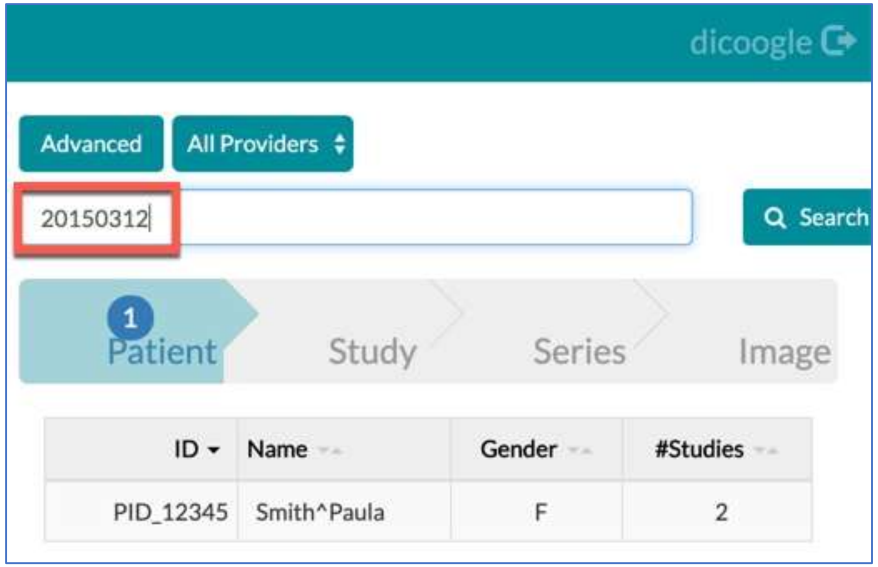
Lucene

Amazon S3

YEOMAN

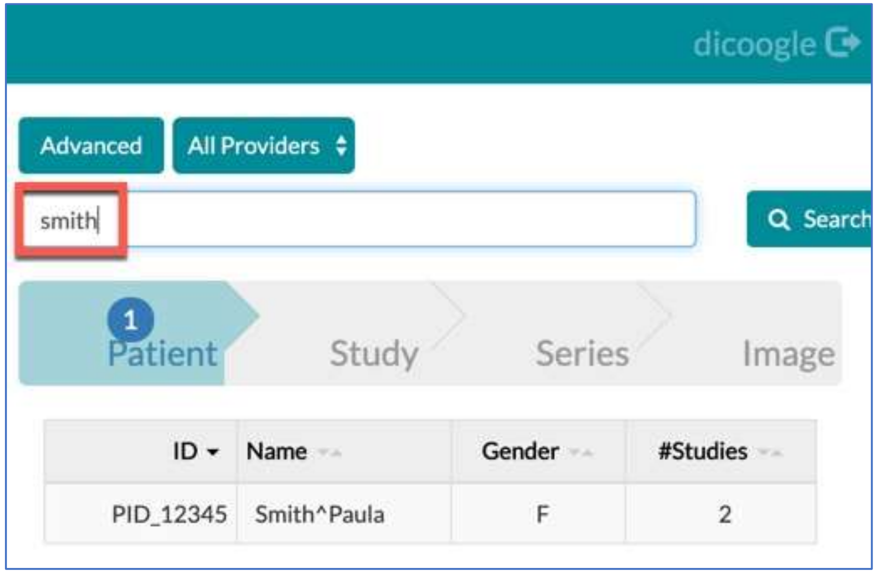
JSON

Dicoogle



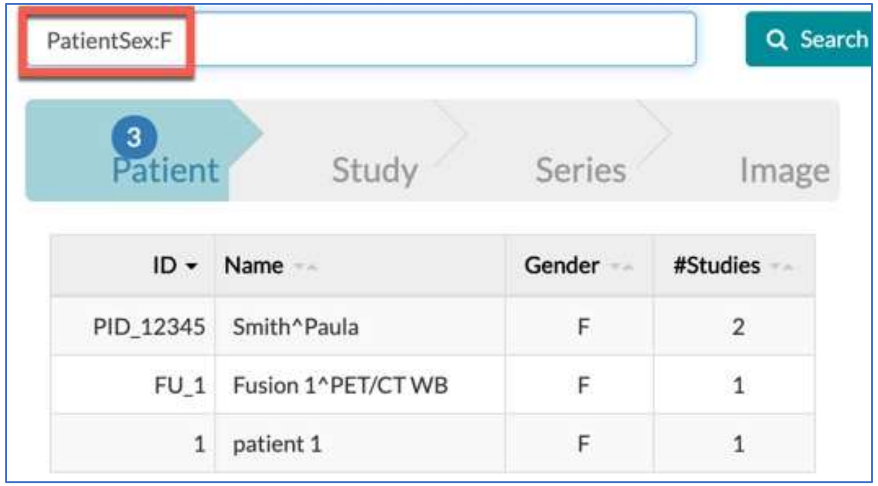
20150312

ID	Name	Gender	#Studies
PID_12345	Smith^Paula	F	2



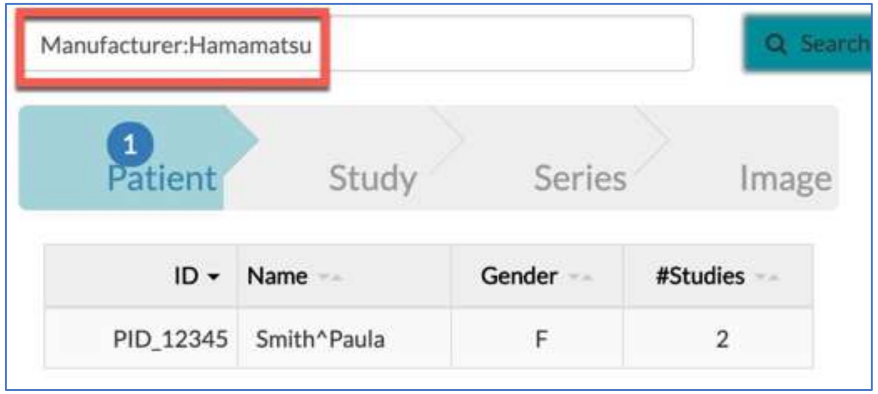
smith

ID	Name	Gender	#Studies
PID_12345	Smith^Paula	F	2



PatientSex:F

ID	Name	Gender	#Studies
PID_12345	Smith^Paula	F	2
FU_1	Fusion 1^PET/CT WB	F	1
1	patient 1	F	1

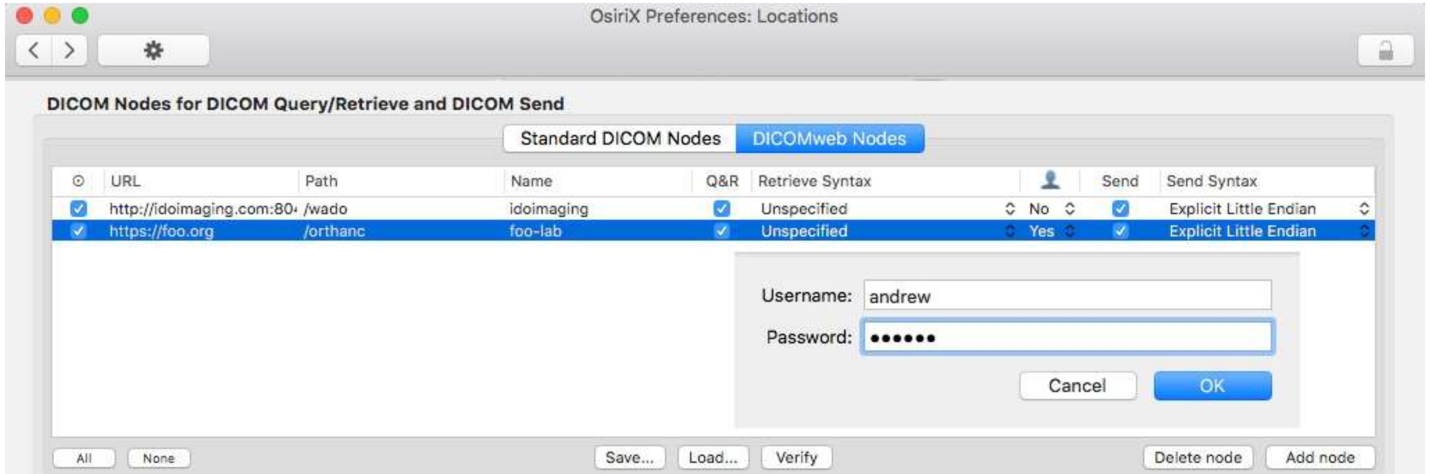


Manufacturer:Hamamatsu

ID	Name	Gender	#Studies
PID_12345	Smith^Paula	F	2

DICOMWeb clients

Osirix



OsiriX Preferences: Locations

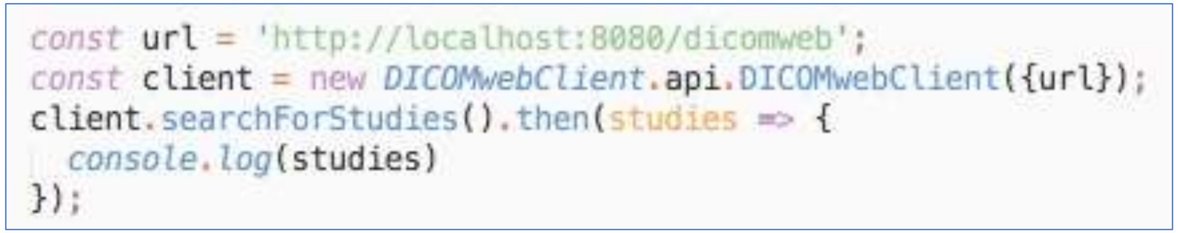
DICOM Nodes for DICOM Query/Retrieve and DICOM Send

⊙	URL	Path	Name	Q&R	Retrieve Syntax	👤	Send	Send Syntax
☑	http://idoimaging.com:80	/wado	idoimaging	☑	Unspecified	No	☑	Explicit Little Endian
☑	https://foo.org	/orthanc	foo-lab	☑	Unspecified	Yes	☑	Explicit Little Endian

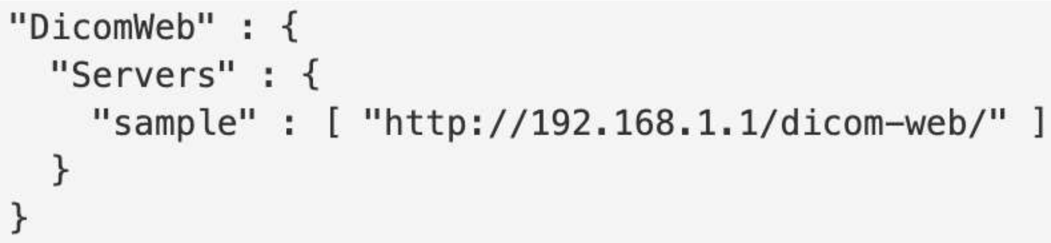
Username: andrew

Password: ••••••

Dicomweb Client (dcmjs.org)



```
const url = 'http://localhost:8080/dicomweb';
const client = new DICOMwebClient.api.DICOMwebClient({url});
client.searchForStudies().then(studies => {
  console.log(studies)
});
```



```
"DicomWeb" : {
  "Servers" : {
    "sample" : [ "http://192.168.1.1/dicom-web/" ]
  }
}
```

DICOMcloud (Zaid Safadi)

DICOMweb Server

Open source DICOMweb server that implements RESTful services

Implements QIDO-RS, STOW-RS, WADO-RS, WADO-URI

‘Azure friendly’ – written in C#

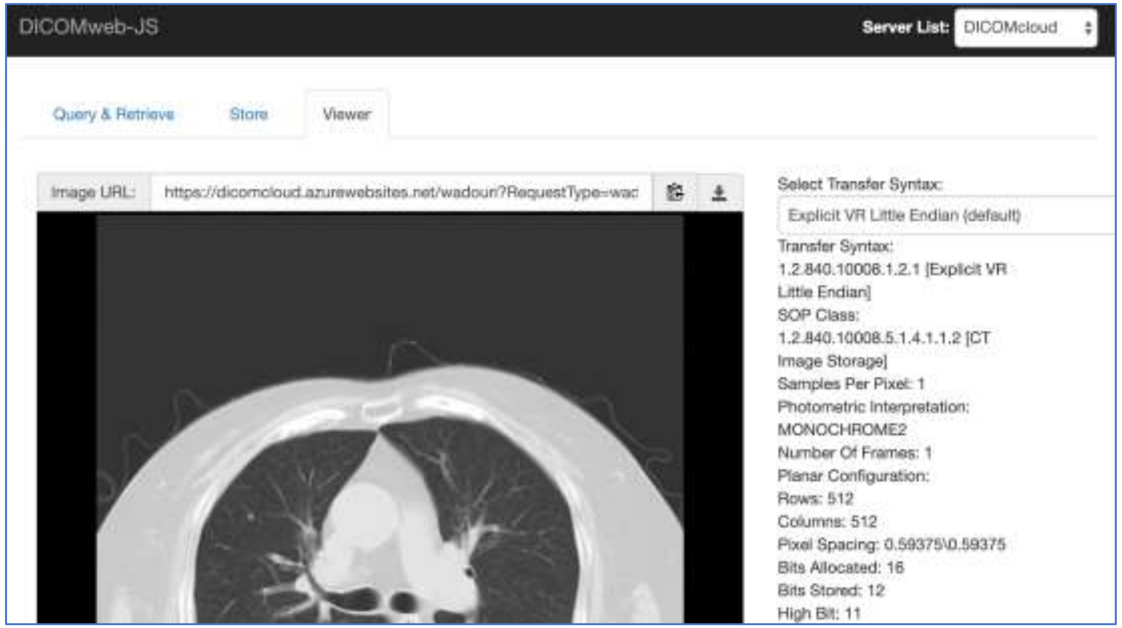
Uses Fellow Oak .NET DICOM library

Live demo on Azure

DICOMweb-js Client

JavaScript image viewer

Works with any DICOMweb server



Server List: DICOMcloud

Image URL: https://dicomcloud.azurewebsites.net/wadouri?RequestType=wad

Select Transfer Syntax: Explicit VR Little Endian (default)

Transfer Syntax: 1.2.840.10008.1.2.1 [Explicit VR Little Endian]
SOP Class: 1.2.840.10008.5.1.4.1.1.2 [CT Image Storage]
Samples Per Pixel: 1
Photometric Interpretation: MONOCHROME2
Number Of Frames: 1
Planar Configuration:
Rows: 512
Columns: 512
Pixel Spacing: 0.59375\0.59375
Bits Allocated: 16
Bits Stored: 12
High Bit: 11

Imaging without DICOM?

More usual in: research, smaller imaging modalities

- Frequent conversion to and from DICOM

Working at the file level is common

- Often working away from clinical PACS systems

Specialized file formats for simplicity (NIFTI, MINC)

- Popular with investigators and developers for ease of adoption

Lack DICOM’s specialized transport protocol

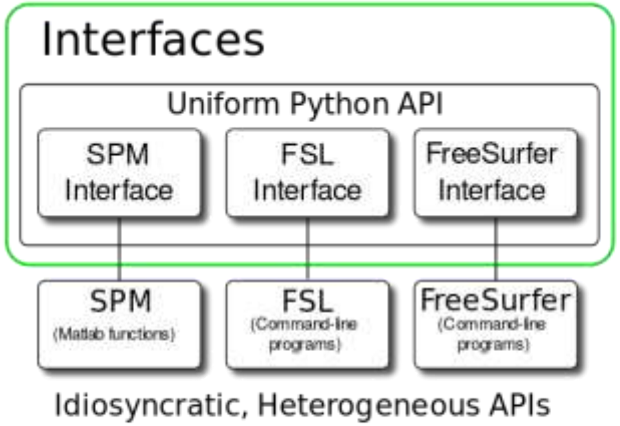
DICOM

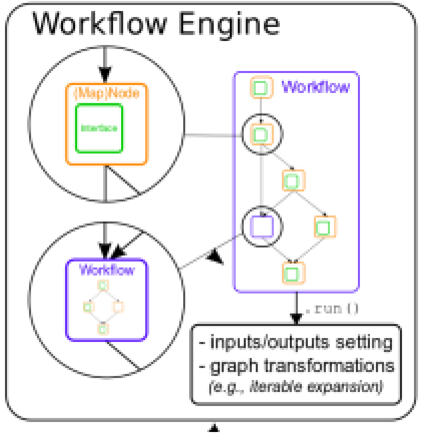
Nifti

NIPY: Neuroimaging in Python

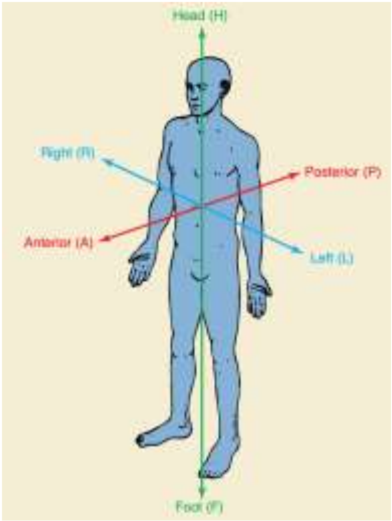
NIPY.org: Python processing of neuroimaging data
Projects in pipeline processing, computational anatomy, file I/O, functional MRI, machine learning, electrophysiology, data visualization

Nipype: A uniform interface to existing neuroimaging software





Nibabel: Read/write access to (neuro)imaging file formats
NIFTI, GIFTI, Analyze, MINC, MGH
Full coverage of coordinate systems and affines



BIDS: Brain Imaging Data Structure

A data exchange format using simple file formats and a defined directory structure

Primarily MRI/fMRI, with extensions for PET, EEG, MEG

Images in NIfTI, tabular data in TSV, key-value pairs in JSON

```
my-experiment
    participants.tsv
    sub-01
        anat
            sub-01_t1.nii.gz
            sub-01_t1.json
        func
            sub-
            01_task_bold.nii.gz
            sub-
            01_task_bold.json
        dwi
            sub-01_dwi.nii.gz
```

```
{
  "TaskName": "N
Back",
  "RepetitionTime":
0.8,
  "EchoTime": 0.03,
```

BIDS Apps

Portable neuroimaging pipelines that understand BIDS datasets

Apps are stored in Docker Hub

Run in Docker or Singularity

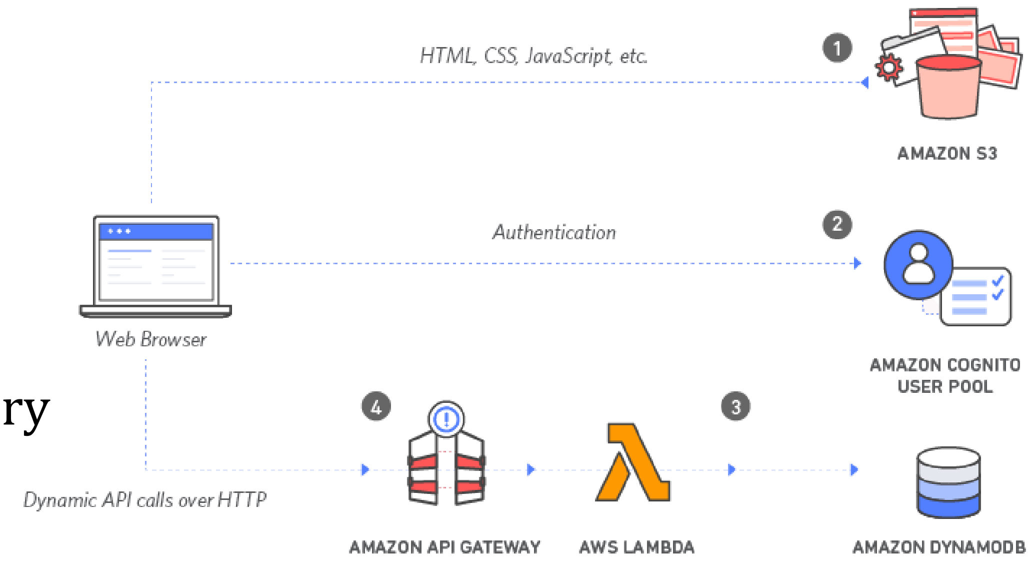
Each has the same core command line arguments

Integrate into automated platforms

What's next?

Web-native serverless PACS

Highly granular, highly scalable, highly available



NoSQL databases

May suit formats with a sparsely populated dictionary

Non-DICOM storage, transport

Would eliminate DICOM dependency and translation

Meta-PACS

Tagging, multiple identities, de-identification, error handling, grouping