

# ORTHANC — Lightweight, RESTful DICOM Server for Healthcare and Medical Research

Sébastien Jodogne

FOSDEM, FOSS for Scientists, February 2nd, 2013



## 1 Introduction

- What is Radiotherapy?
- The Radiotherapy Process
- DICOM Format and Protocol

## 2 Orthanc

- Philosophy
- Orthanc in Action
- Current Applications

## 3 Conclusions

- How to Contribute?
- Summary

## What is Radiotherapy?

*“The medical use of **ionizing radiation**, generally as part of cancer treatment, to control or kill malignant cells.”*

## At CHU of Liège (Belgium)

- $\approx$  2400 new cases per year.
- First radiotherapy department in Wallonia.
- Second in Belgium (KUL).

## Modern Treatment Machines at Liège

Elekta



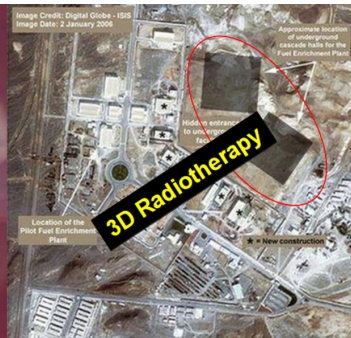
- Linear accelerator ( $\approx 20\text{MV}$ ).
- Embedded imaging for positioning (2D or 3D).

CyberKnife



- Linac on a robotic arm.
- Very high doses.
- Breath tracking.

# The Evolution of Radiotherapy



## Surgical Strike

Main Problem

How to identify the target? ⇒ **Medical Imaging**

# Multimodal Medical Imaging for Radiotherapy

**CT**



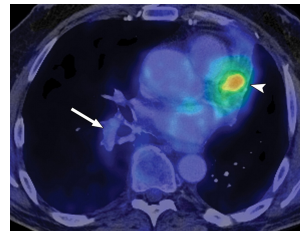
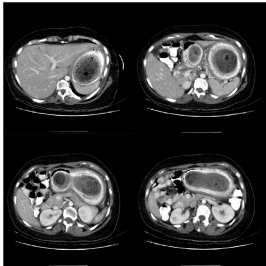
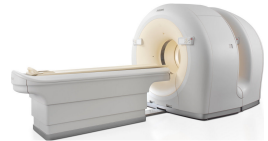
+

**MRI**

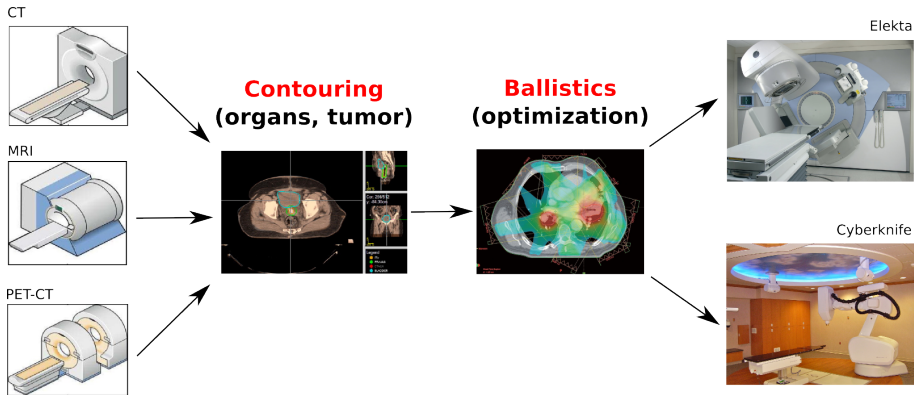


+

**PET-CT**



# Overall Process



## Software Challenges of Modern Radiotherapy

- Many images are produced and analyzed.
- Exchange of images between several medical departments.
- Each image is heavyweight ( $\approx 512\text{MB}$  per 3D image).
- Many manufacturers are involved  $\Rightarrow$ 
  - Proprietary, costly, closed software.
  - Limited interoperability.



# What Is DICOM?

# What Is DICOM?

## 1 File Format:

One **image** slice



+

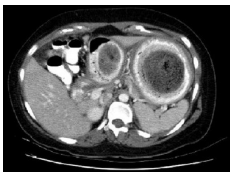
Patient **metadata**

```
<?xml version="1.0" en  
<person id="0847  
<name>Kris S  
<address>  
<street?<br/>  
<city>B
```

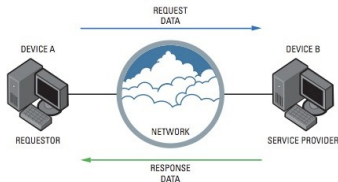
# What Is DICOM?

## 1 File Format:

One **image** slice + Patient **metadata**



## 2 Network Protocol — Early example of Web services:



- **Search** remote images,
- **Send** local images,
- **Retrieve** remote images.

# What Is DICOM?

## Part of the Solution

- De-facto standard in medical imaging.
- Mature (30 years old).
- Supported by any medical device.

# What Is DICOM?

## Part of the Solution

- De-facto standard in medical imaging.
- Mature (30 years old).
- Supported by any medical device.

## Part of the Problem

- Large and complex (over 4000 pages).
- Old format and protocol (redundant with XML, SOAP, ...).
- Huge number of files (3D images split slice-by-slice)
  - ⇒ Redundancies between DICOM files,
  - ⇒ Need for automated indexing.
- Not developer-friendly (complex parsing).

## Main Open-Source Software for DICOM

### DICOM Toolkits

- DCMTK (format, protocol),
- GDCM (format only).

### DICOM Databases (aka. PACS)

- Dcm4chee,
- Dicoogle.

# Main Open-Source Software for DICOM

## DICOM Toolkits

- DCMTK (format, protocol),
- GDCM (format only).

## DICOM Databases (aka. PACS)

- Dcm4chee,
- Dicoogle.

## Limitations

- Close-to-metal (network administrators?).
- No indexing.
- Windows support.

## Limitations

- Heavyweight.
- Database administration.
- Still need a DICOM toolkit to automate tasks.

## 1 Introduction

- What is Radiotherapy?
- The Radiotherapy Process
- DICOM Format and Protocol

## 2 Orthanc

- Philosophy
- Orthanc in Action
- Current Applications

## 3 Conclusions

- How to Contribute?
- Summary

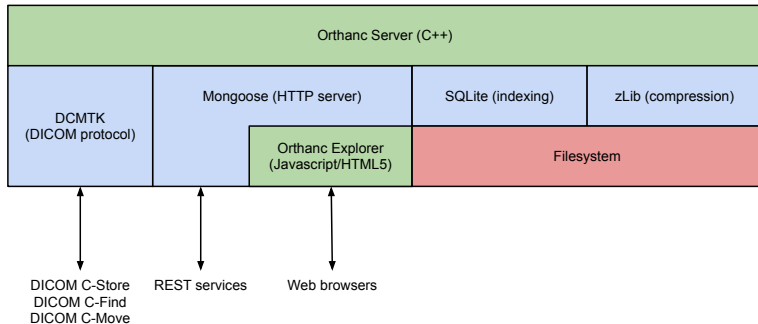


# ORTHANC

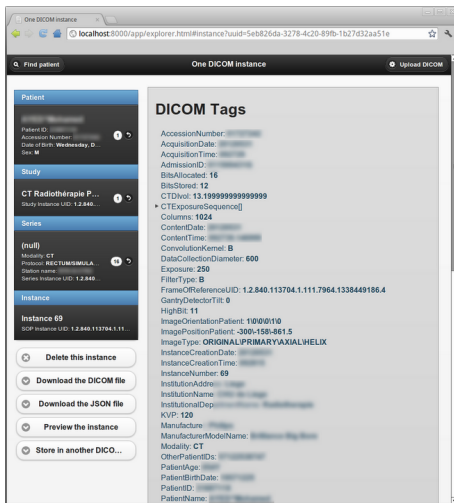
## Philosophy

- 1 **Compromise** between a DICOM toolkit and a PACS.
- 2 **Lightweight** DICOM store (standalone, cross-platform).
- 3 **Immediate to deploy** on any PC (no external database).
- 4 **RESTful API** to DICOM format and protocol.  
⇒ DICOM scripting (network administrators).
- 5 Convert DICOM files to **modern formats** (JSON and PNG).
- 6 Simple **Web interface** to explore the content of the store.
- 7 ...And, of course, **open-source** :)

# Software Architecture



# Web Interface in Action



The screenshot shows a web browser window displaying the Orthanc web interface. The browser address bar shows the URL: `localhost:8000/app/explorer.html#instance/uuid=Seb826da-3278-4c20-89fb-1b27d32aa51e`. The interface has a dark header with a search bar and the text "One DICOM instance" and "Upload DICOM".

The main content area is divided into two columns. The left column contains a sidebar with a tree view showing the hierarchy: Patient, Study, Series, and Instance. The right column displays the "DICOM Tags" for the selected instance, listing various DICOM metadata fields such as AccessionNumber, AcquisitionDate, AcquisitionTime, AdmissionID, BitsAllocated, BitsStored, CTDVol, CTE ExposureSequence, Columns, ContentDate, ContentTime, ConvolutionKernel, DataCollectionDiameter, Exposure, FilterType, FrameOfReferenceUID, GantryDetectorTilt, HighBit, ImageOrientationPatient, ImagePositionPatient, ImageType, InstanceCreationDate, InstanceCreationTime, InstanceNumber, InstitutionAddress, InstitutionName, InstitutionalDepartment, KVP, Manufacturer, ManufacturerModelName, Modality, OtherPatientIDs, PatientAge, PatientBirthDate, PatientID, and PatientName.

At the bottom of the sidebar, there are several buttons: "Delete this instance", "Download the DICOM file", "Download the JSON file", "Preview the instance", and "Store in another DICOM..."

## Features

- **“Patient → Study → Series → Instance”** transparent indexing.
- **Preview** of images inside the browser.
- **Send** to other DICOM modalities.
- **Download ZIP** files.
- **Anonymization.**

# Scripting with Python — Automatic Bridge to the PACS

```
sys.path.append('Orthanc/Resources/Samples/Python')
from RestToolbox import *

# Event polling loop
current = 0
while True:
    # Wait for the occurrence of new events in Orthanc
    r = DoGet('http://localhost:8042/changes', { 'since' : current })

    # React when a new instance is received
    for change in r['Changes']:
        if change['ChangeType'] == 'NewInstance':

            # Send the new instance to the PACS
            instancelid = str(change['ID'])
            DoPost('http://localhost:8042/modalities/pacs/store', instancelid)

            # Remove the instance from Orthanc
            DoDelete('http://localhost:8042/instances/%s' % instancelid)

current = r['Last']
time.sleep(1)
```

# Current Applications of Orthanc

- **DICOM buffer** for transmission errors:
  - Nuclear Medicine → Radiotherapy,
  - Treatment Planning System → Treatment machines.
- Collect in-room images for **adaptive radiotherapy** research.
- **Anonymization platform** for clinical research with firms.
- Automated **quality control of PET-CT scanners**.
- Automated **quantification of focal fibrosis** on cardiac MRI.
- **Mini-PACS** on desktop computers ⇒ medical and image processing research.

## 1 Introduction

- What is Radiotherapy?
- The Radiotherapy Process
- DICOM Format and Protocol

## 2 Orthanc

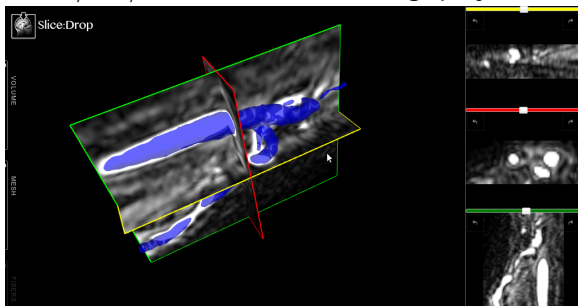
- Philosophy
- Orthanc in Action
- Current Applications

## 3 Conclusions

- How to Contribute?
- Summary

# How to Contribute?

- Develop a professional, Web-based **DICOM viewer (WebGL)**  
⇒ MIP, PET/CT/MRI fusion, windowing, projection,...



- Port to **OS X, RHEL and CentOS**.
- **FUSE filesystem** on the top of the REST API.
- Support **WADO**, standardize **WADO-RS** (RESTful DICOM).
- Implement **DICOM-RT** primitives (for radiotherapy).
- Develop a **Matlab/Octave toolbox**.

# Summary



<https://orthanc.googlecode.com/>

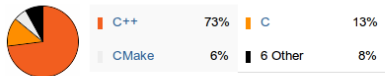
- Open-source DICOM store.
- Lightweight, scriptable, versatile.
- Availability:
  - Part of Debian Med,
  - Fedora package waiting for approval,
  - Windows binaries.
- **Publications:** ISBI 2013, ESTRO 2013, 2 @ BHPA 2013.



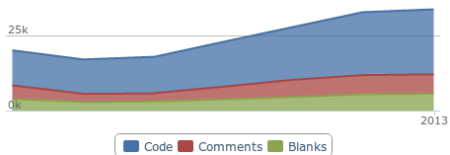
# Appendix

# Ohloh Statistics

## Languages



## Lines of Code



## Focus on Automated Testing

- Continuous Integration Server,
- Unit testing,
- Integration tests through REST API,
- Valgrind,
- Trello to track developments.

## Desirable Features for a Novel DICOM Viewer

- Web-based  $\Rightarrow$  accessible from any PC in the hospital without installation.
- Cross-platform, lightweight (WebGL, XTK, Slice:Drop).
- Side-by-side display, link the two series of images (with zoom).
- Windowing (Hounsfield units, 16bpp).
- 3D view + 2D projections (axial, coronal, sagittal).
- PET/CT/MRI superposition and alignment (offsets).
- MIP rendering (Maximum Intensity Projection  $\Leftrightarrow$  EXT\_blend\_minmax).
- Display DICOM-RT (doses and structures).
- Possibility to use non-destructive image compression.
- Speed, smoothness, stability, simplicity (of course).
- Other features: Consider OsiriX (*but OS X only!*).