DICOM services for XNAT

Using the DICOM standard to streamline data storage and retrieval

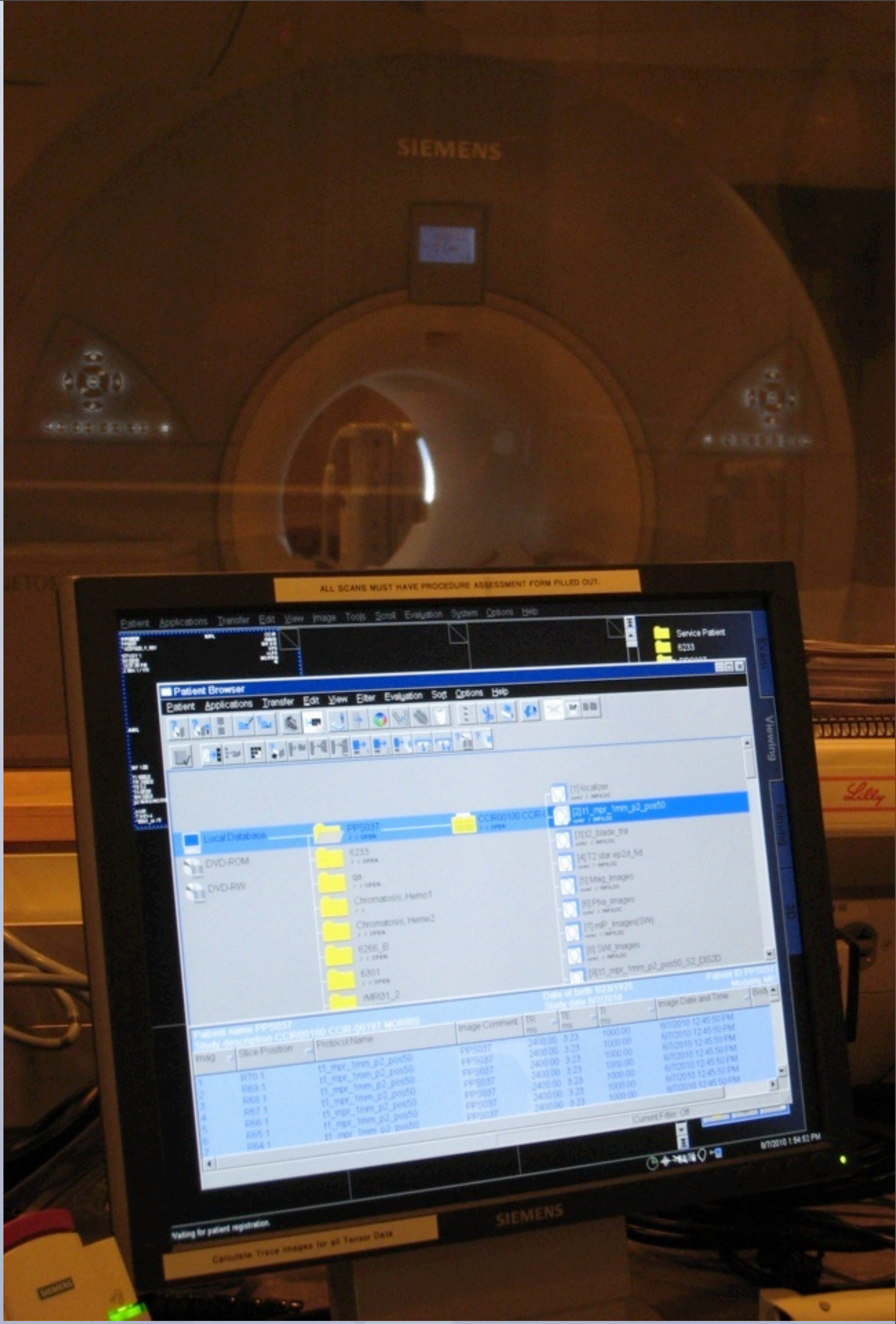
Plan

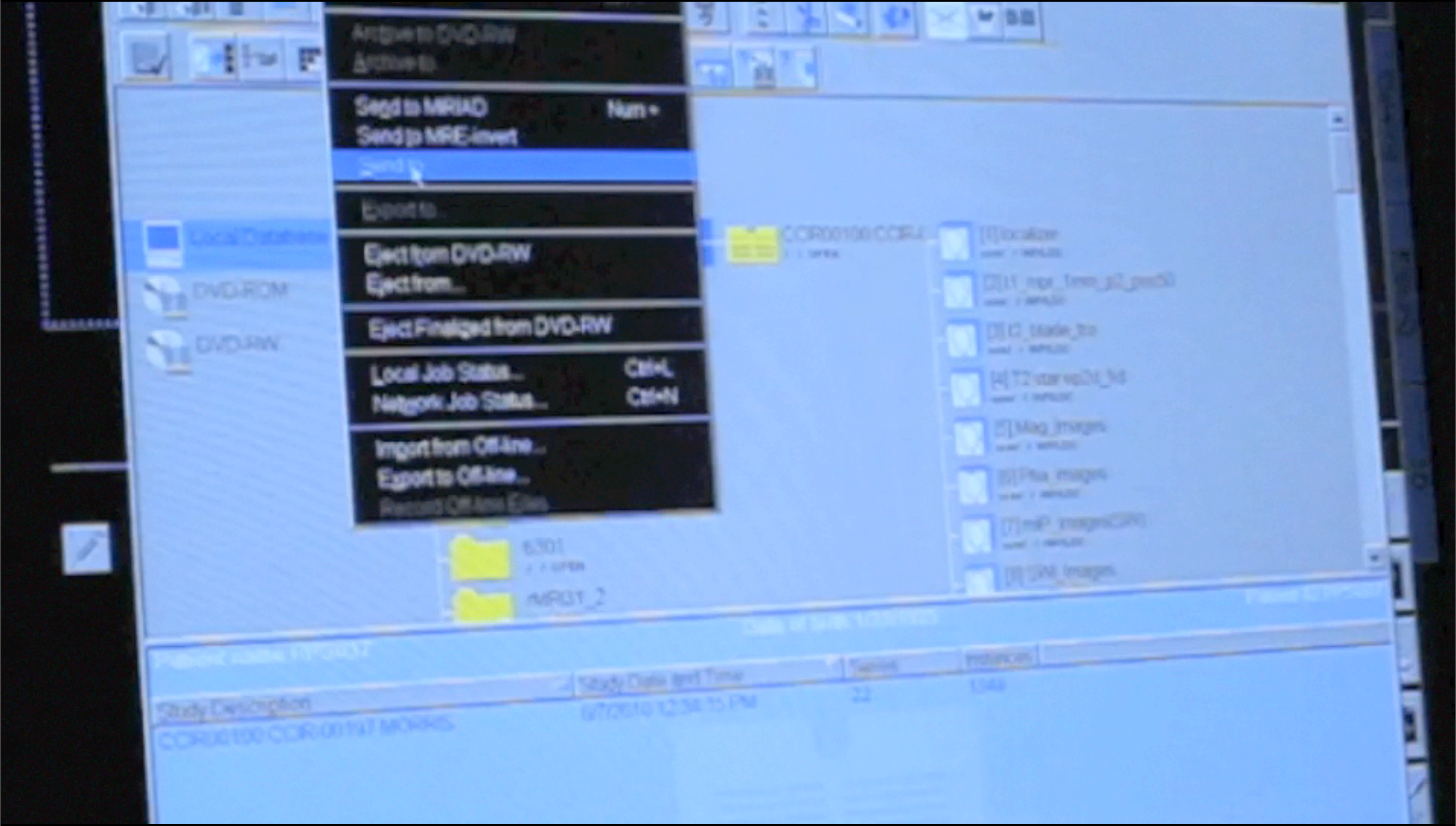
In this talk, we will:

- See how users can get DICOM data into and out of XNAT
- Identify the components XNAT uses for DICOM integration
- Learn how to configure and troubleshoot XNAT’s DICOM services

Introduction

- XNAT DicomServer simplifies data import
- DicomBrowser offers GUI and command-line tools for finding, deidentifying, and sending DICOM data
- XNAT Gateway provides DICOM query and retrieve services


SIEMENS
ALL SCANS MUST HAVE PROCEDURE ASSESSMENT FORM FILLED OUT.
Patient Browser
Local Database
DVD-ROM
DVD-RW
Waiting for patient registration
Calculate Trace Images for all Tensor Data
SIEMENS

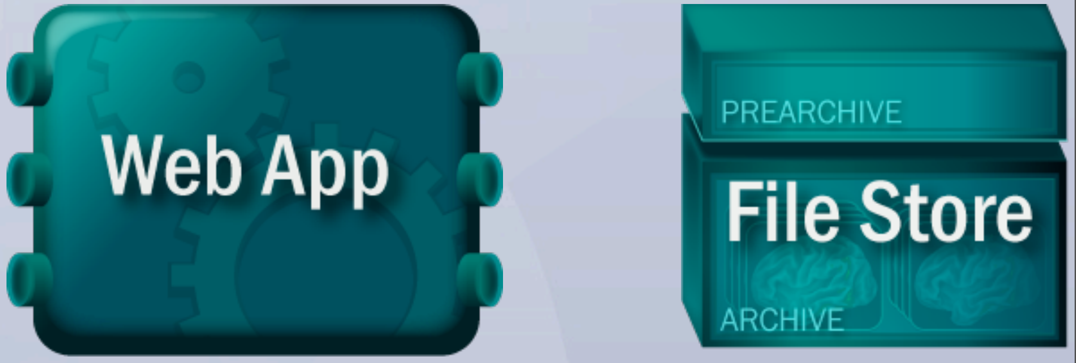


Study Description	Study Date and Time	Series	Instances
CCIR00100 CCIR00107 MORRIS	6/7/2010 12:38:15 PM	22	1044

TAZINTOSH.COM
Edouard Pagan

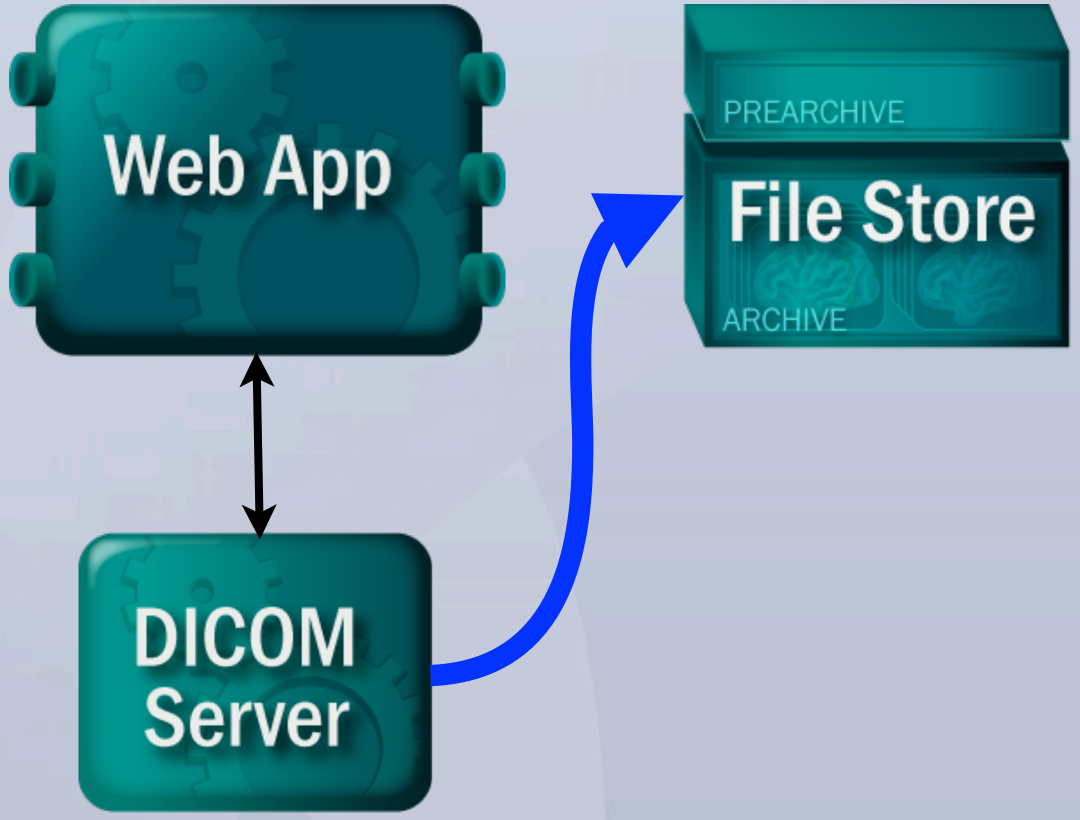
Components

- DicomServer uses the XNAT web application as a store for session metadata
- Imaging data are stored as files on disk, not in the XNAT database



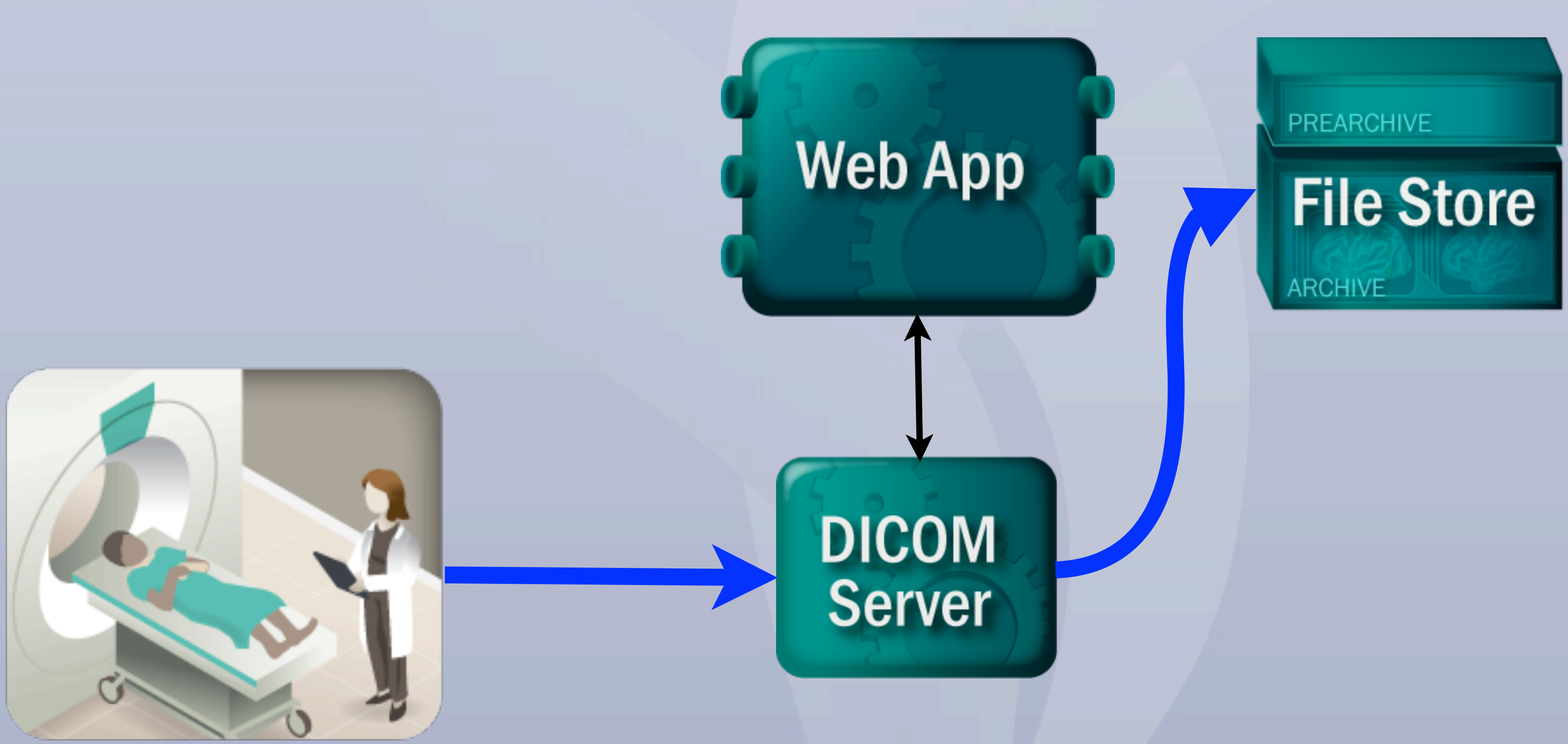
Components

- DicomServer:
  - is a DICOM C-STORE receiver (SCP)
  - receives DICOM data from a network and saves files to disk
  - translates DICOM metadata to XNAT metadata format
- XNAT can also save files and extract metadata



Components: DicomServer

Any DICOM C-STORE client can send data, including scanners...



Web App

PREARCHIVE

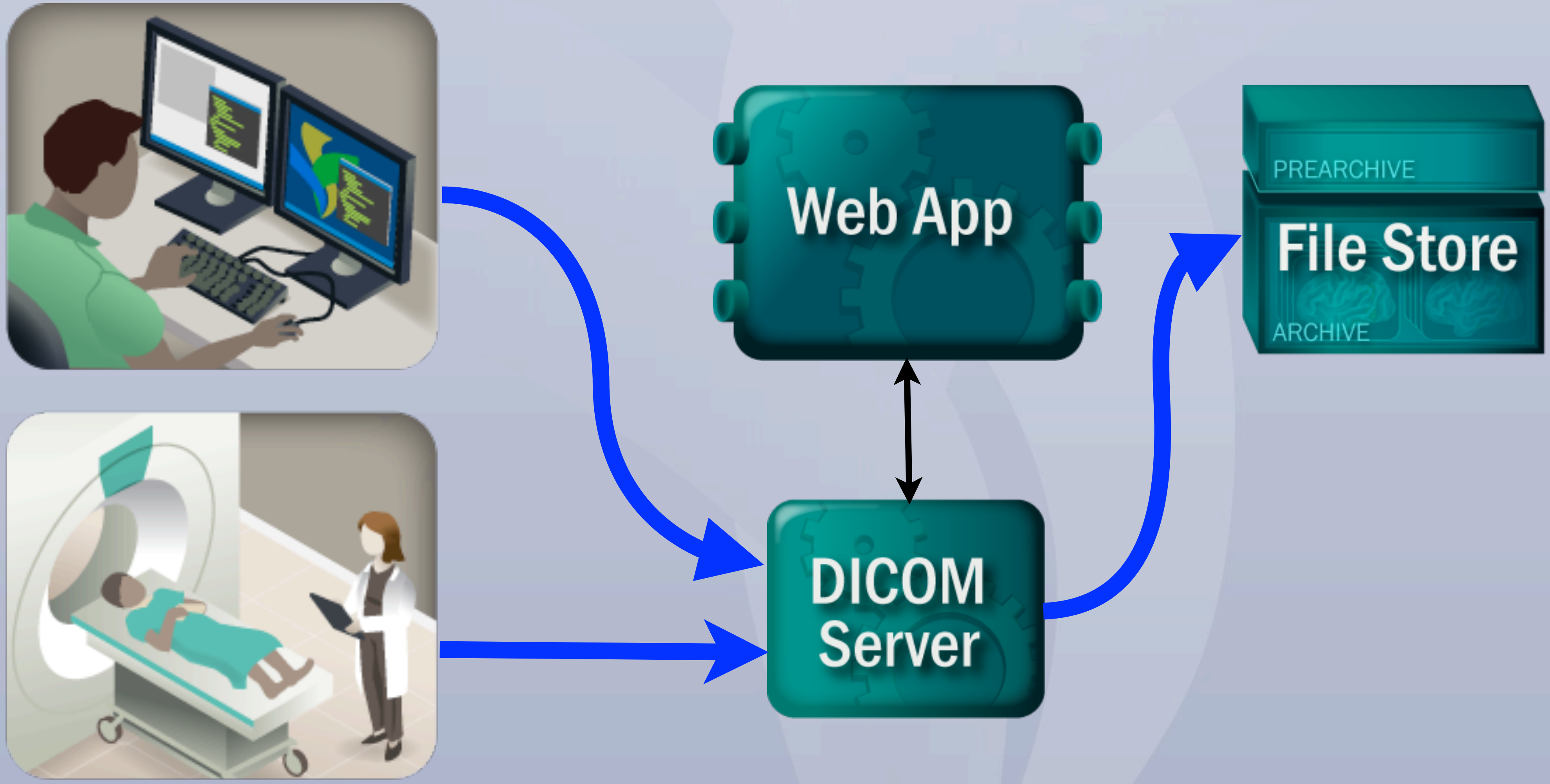
File Store

ARCHIVE

DICOM Server

Components: DicomServer

...also dedicated workstations, or viewers like OsiriX or DicomBrowser...

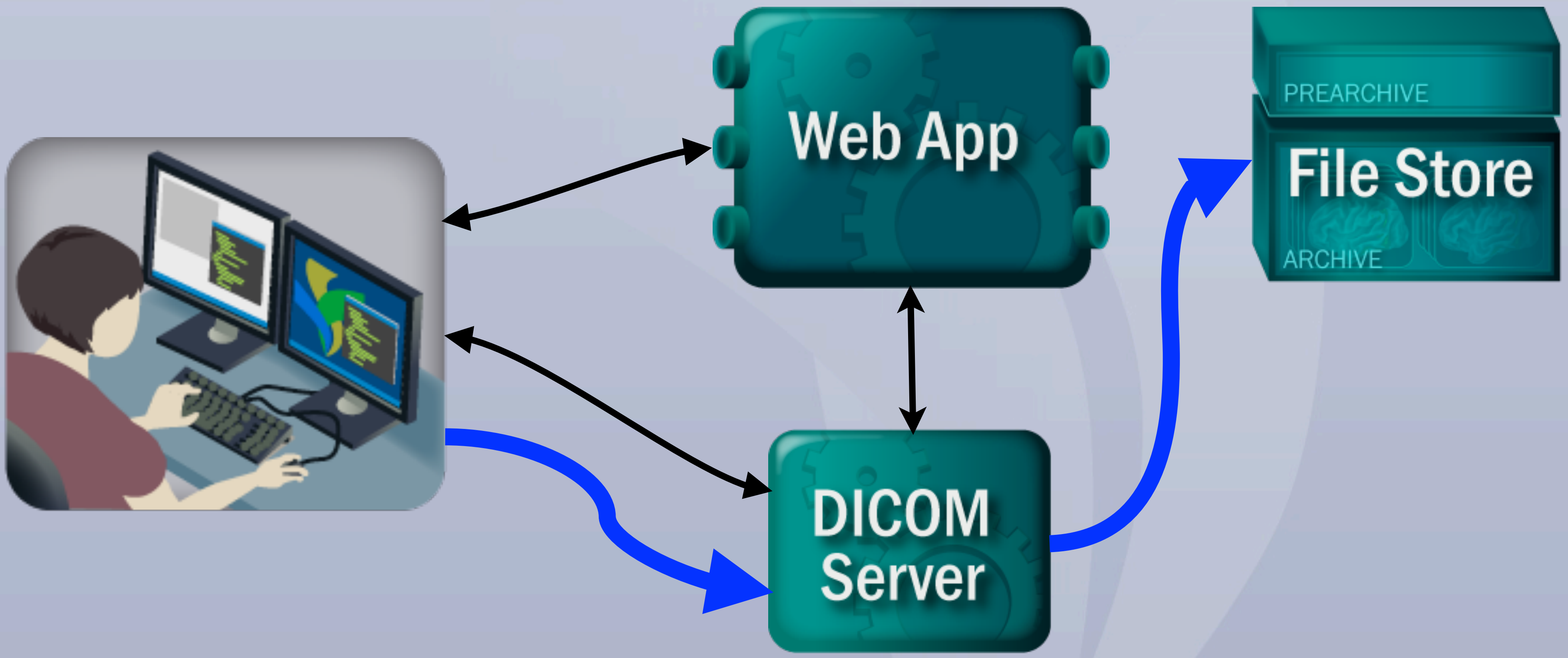


XNAT

Components: DicomServer

...or the XNAT upload applet, which:

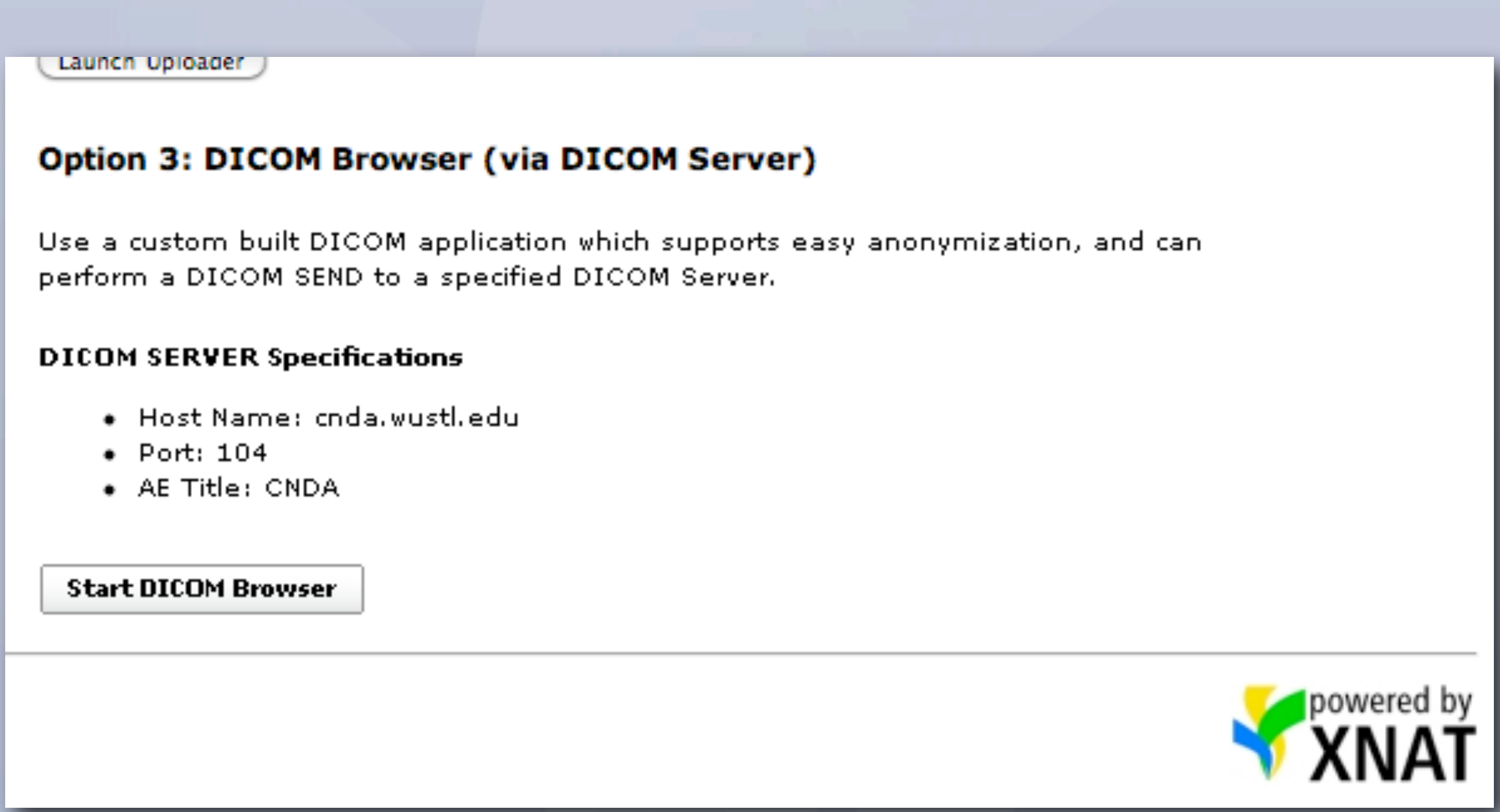
- guides user to label the XNAT session
- uploads DICOM or ECAT data



Uploading data

Upload client needs to know some connection parameters:

- server hostname
- port number
- DICOM AE title



Option 3: DICOM Browser (via DICOM Server)

Use a custom built DICOM application which supports easy anonymization, and can perform a DICOM SEND to a specified DICOM Server.

DICOM SERVER Specifications

- Host Name: cnda.wustl.edu
- Port: 104
- AE Title: CNDA

Uploading data

Which XNAT project receives the data can be specified in:

- Patient Comments
- Study Comments
- Study Description
- Accession Number
- or some other (configurable) DICOM attribute

Uploading data

Received data (usually) goes into XNAT prearchive

- Prearchive is a holding/review area
- Projects can be configured to skip prearchive step (“autoarchiving”)
- Autoarchiving can be specified per session in DICOM Patient Comments or Study Comments as `AA:true` or `AA:false`

XNAT upload applet

- Provides a GUI for finding, labeling, and uploading DICOM data (ECAT too)
- Uploader uses DICOM C-STORE to move DICOM data
- Also uses second channel (HTTP) to carry XNAT session metadata

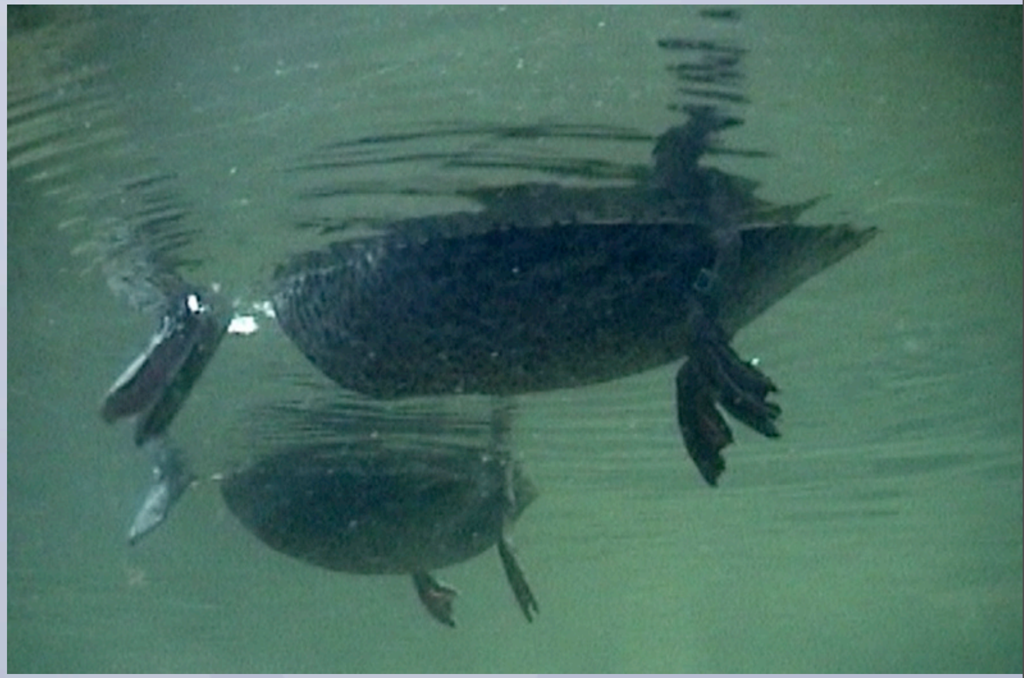
Data model conversion

DicomServer builds an XNAT session from each DICOM study it receives

- Data are reorganized into files in XNAT directory structure
- Session directory is placed in appropriate location (XNAT prearchive or archive)
- DICOM metadata are translated into XNAT session and scan information

Uploading, in detail

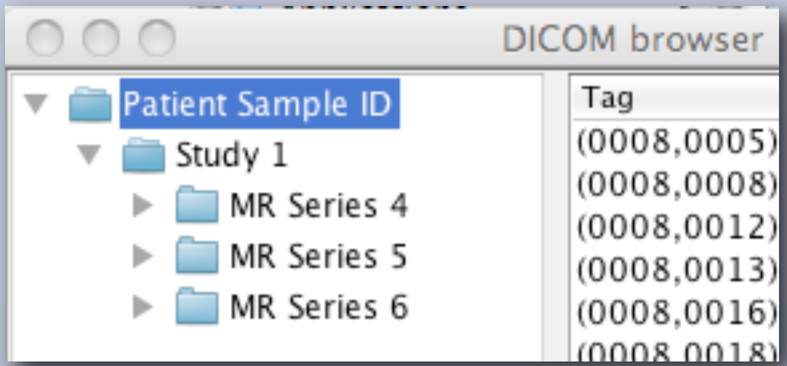
- DICOM C-STORE to transfer data
- No explicit end-of-study signal; DicomServer uses idle timeout to decide when a session is complete
- DicomServer translates DICOM metadata into XNAT format

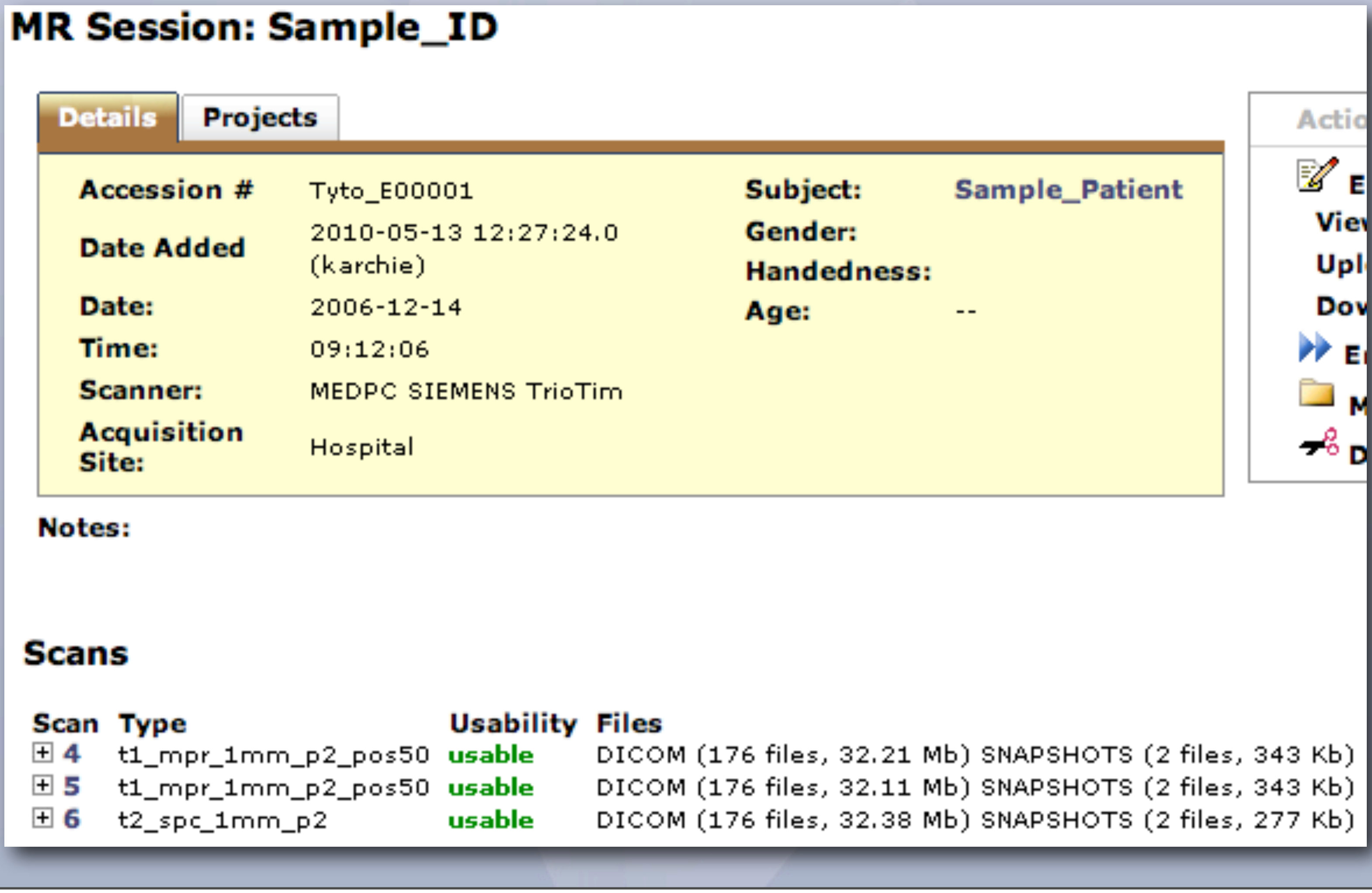


Metadata conversion

XNAT and DICOM have different metadata models

- DICOM study ~ XNAT session
- DICOM series ~ XNAT scan





MR Session: Sample_ID

Accession #	Tyto_E00001	**Subject:**	Sample_Patient
Date Added	2010-05-13 12:27:24.0 (karchie)	**Gender:**	
		Handedness:	
Date:	2006-12-14	**Age:**	--
Time:	09:12:06		
Scanner:	MEDPC SIEMENS TrioTim		
Acquisition Site:	Hospital		

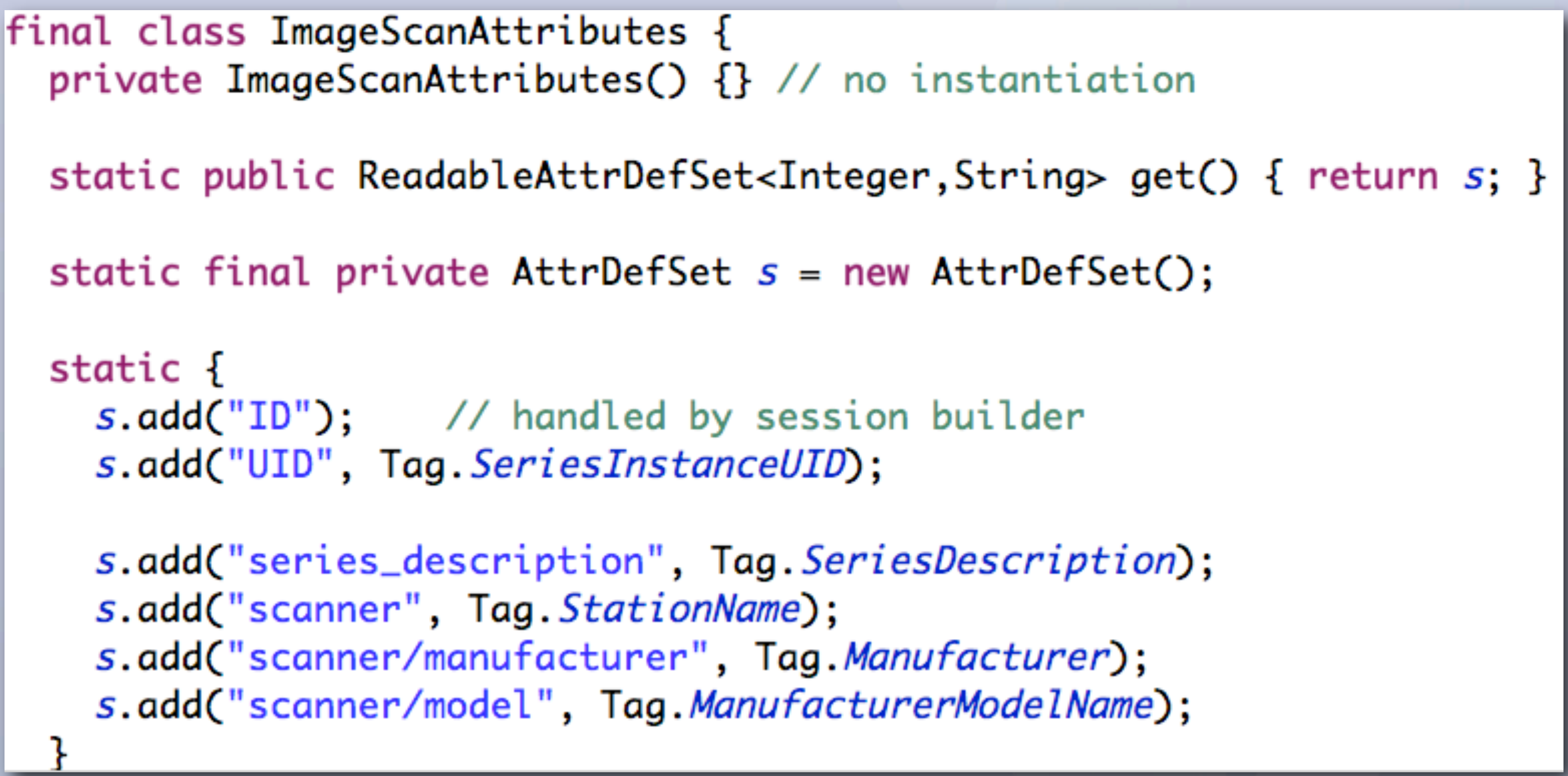
Notes:

Scans

Scan	Type	Usability	Files
⊞ 4	t1_mpr_1mm_p2_pos50	usable	DICOM (176 files, 32.21 Mb) SNAPSHOTS (2 files, 343 Kb)
⊞ 5	t1_mpr_1mm_p2_pos50	usable	DICOM (176 files, 32.11 Mb) SNAPSHOTS (2 files, 343 Kb)
⊞ 6	t2_spc_1mm_p2	usable	DICOM (176 files, 32.38 Mb) SNAPSHOTS (2 files, 277 Kb)

Metadata conversion

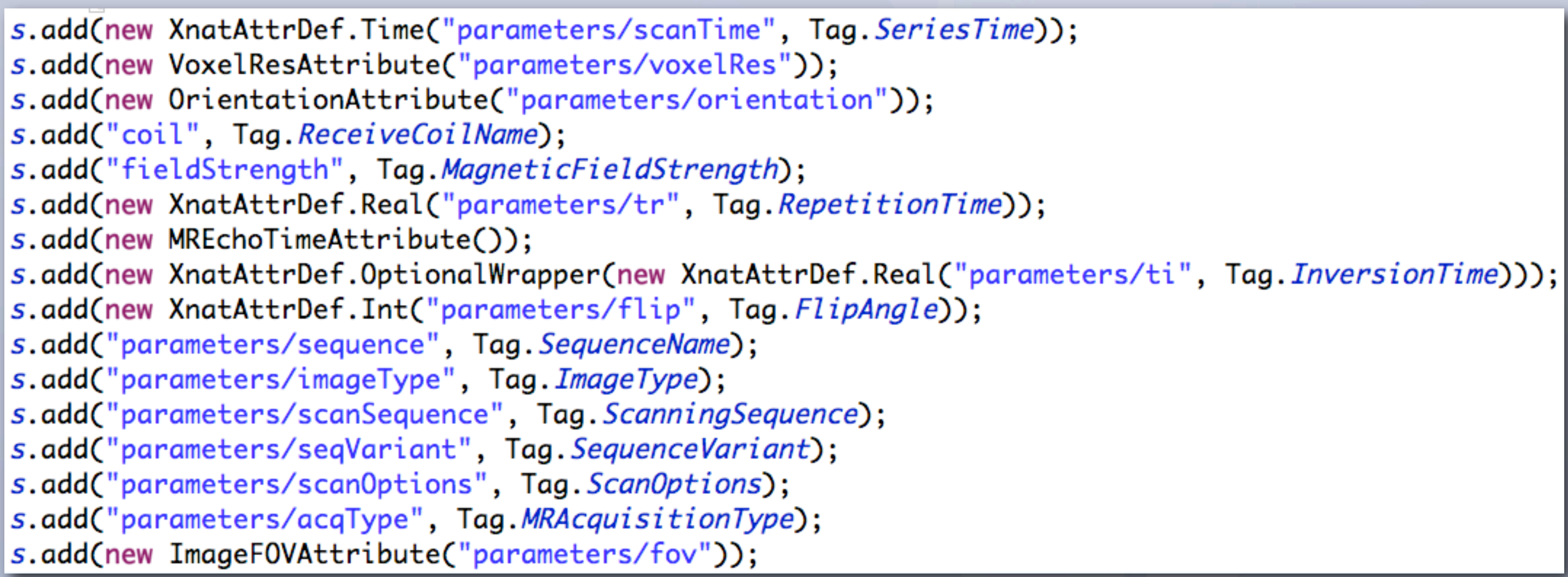
Each XNAT metadata field is derived from one or more DICOM fields



```
final class ImageScanAttributes {
  private ImageScanAttributes() {} // no instantiation

  static public ReadableAttrDefSet<Integer,String> get() { return s; }

  static final private AttrDefSet s = new AttrDefSet();

  static {
    s.add("ID");     // handled by session builder
    s.add("UID", Tag.SeriesInstanceUID);

    s.add("series_description", Tag.SeriesDescription);
    s.add("scanner", Tag.StationName);
    s.add("scanner/manufacturer", Tag.Manufacturer);
    s.add("scanner/model", Tag.ManufacturerModelName);
  }
```

Metadata conversion

Some conversions are nontrivial and require their own Java classes



```
s.add(new XnatAttrDef.Time("parameters/scanTime", Tag.SeriesTime));
s.add(new VoxelResAttribute("parameters/voxelRes"));
s.add(new OrientationAttribute("parameters/orientation"));
s.add("coil", Tag.ReceiveCoilName);
s.add("fieldStrength", Tag.MagneticFieldStrength);
s.add(new XnatAttrDef.Real("parameters/tr", Tag.RepetitionTime));
s.add(new MREchoTimeAttribute());
s.add(new XnatAttrDef.OptionalWrapper(new XnatAttrDef.Real("parameters/ti", Tag.InversionTime)));
s.add(new XnatAttrDef.Int("parameters/flip", Tag.FlipAngle));
s.add("parameters/sequence", Tag.SequenceName);
s.add("parameters/imageType", Tag.ImageType);
s.add("parameters/scanSequence", Tag.ScanningSequence);
s.add("parameters/seqVariant", Tag.SequenceVariant);
s.add("parameters/scanOptions", Tag.ScanOptions);
s.add("parameters/acqType", Tag.MRAcquisitionType);
s.add(new ImageFOVAttribute("parameters/fov"));
```

Metadata modification

Upload applet and DicomBrowser have a scripting language for specifying changes to DICOM metadata (a.k.a. “anon scripts”)

```
(0010,0010) := "Subject_346"
-(0010,0030)              // Patient's Birth Date
(0008,0080) = "CCIR 3T" : (0008,0080) := "WUSM CCIR"
(0020,000d) := new UID
(0010,0020) := replace[(0010,0020)," ","_"]
```

Metadata modification

Each XNAT project may be configured with a script that is applied to DICOM data by the upload applet:

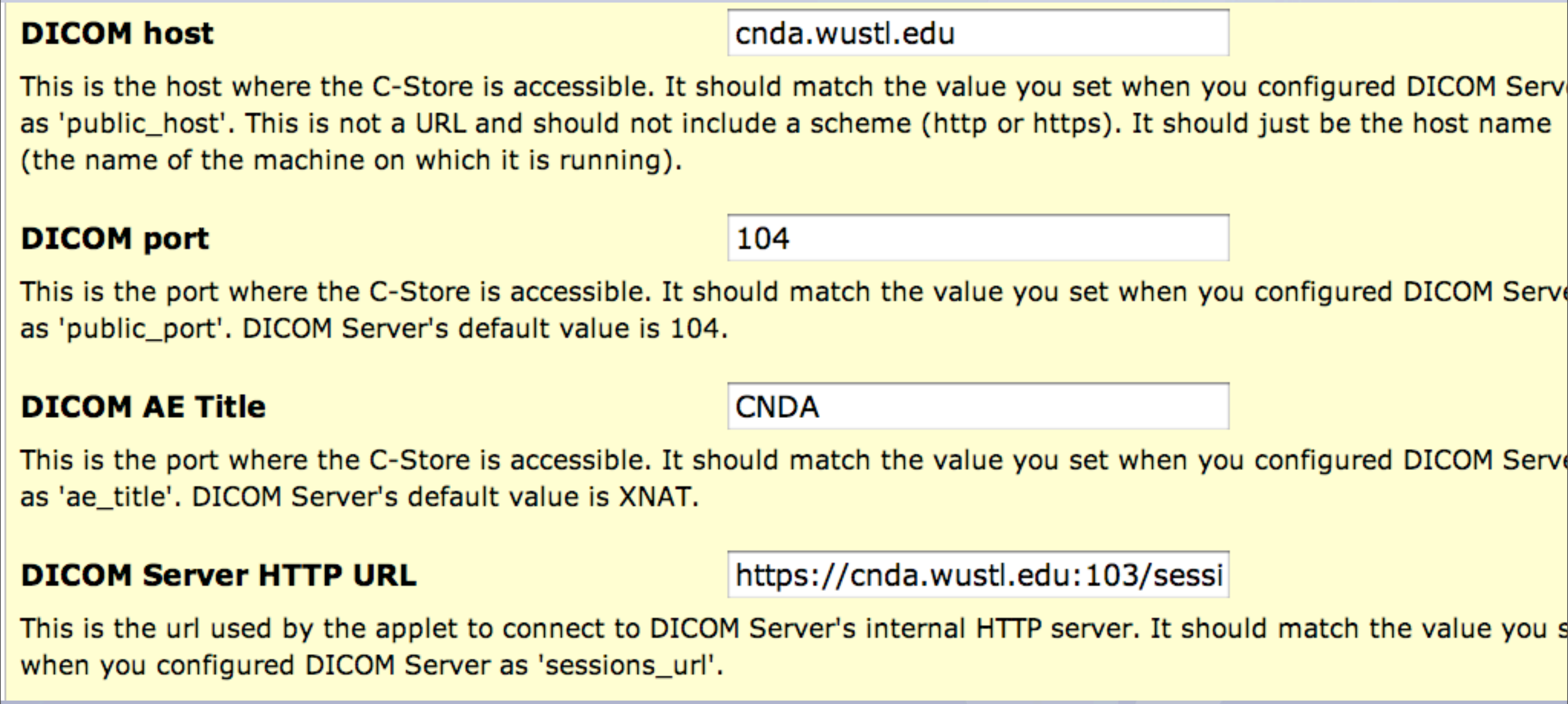
http://*my.xnat.org*/REST/projects/*MyProject*/
resources/UPLOAD_CONFIG/files/dicom.das

Configuration: basic

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
<comment>Simple DicomServer configuration</comment>

<entry key="arcspec">/opt/xnat/cache/archive_specification.xml</entry>
<entry key="xnat_url">http://my.xnat.org:8080/xnat</entry>
<entry key="user">admin</entry>
<entry key="password">adminpw</entry>
<entry key="http_port">8180</entry>

<entry key="log4j.rootLogger">WARN,R</entry>
        ...
</properties>
```

Configuration:port mapping

Can map port 104 to 8104 (via iptables) without changing DicomServer config



DICOM host cnda.wustl.edu

This is the host where the C-Store is accessible. It should match the value you set when you configured DICOM Serv as 'public_host'. This is not a URL and should not include a scheme (http or https). It should just be the host name (the name of the machine on which it is running).

DICOM port 104

This is the port where the C-Store is accessible. It should match the value you set when you configured DICOM Serv as 'public_port'. DICOM Server's default value is 104.

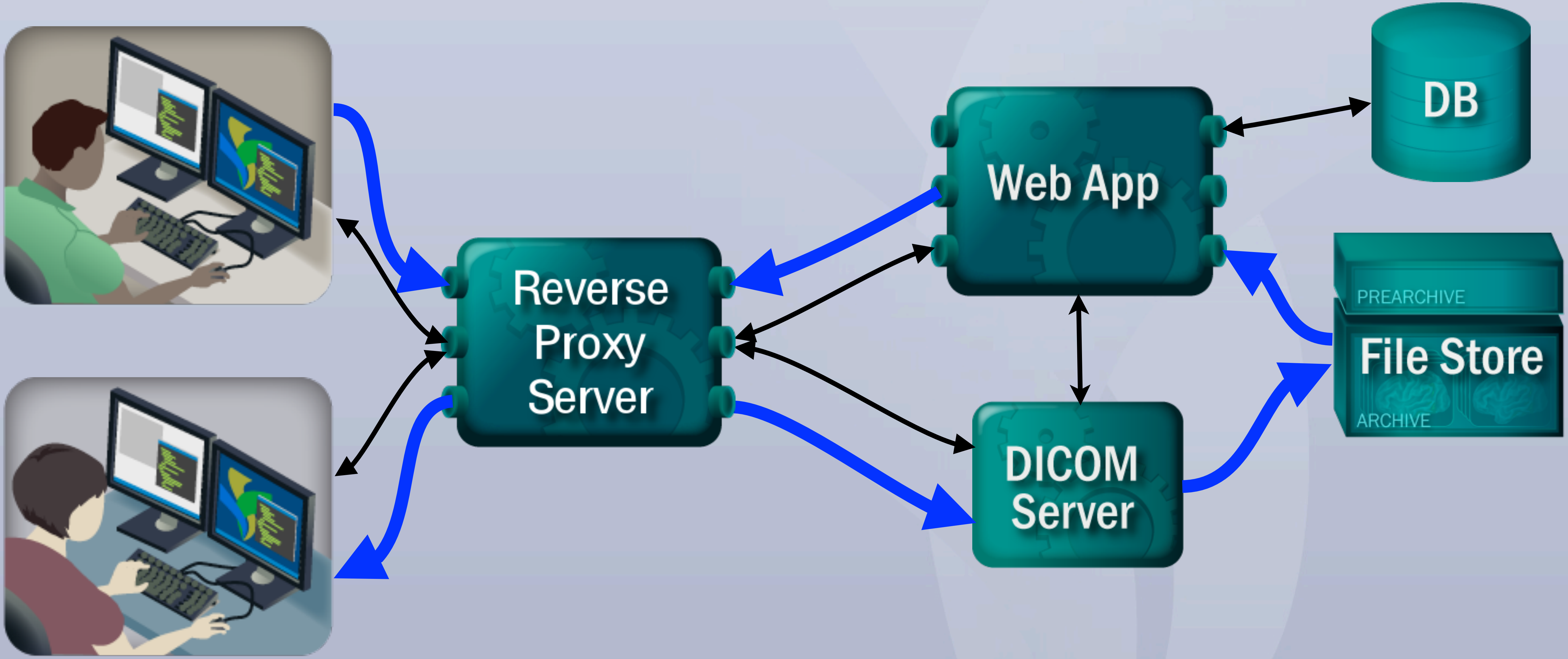
DICOM AE Title CNDA

This is the port where the C-Store is accessible. It should match the value you set when you configured DICOM Serv as 'ae_title'. DICOM Server's default value is XNAT.

DICOM Server HTTP URL https://cnda.wustl.edu:103/sessi

This is the url used by the applet to connect to DICOM Server's internal HTTP server. It should match the value you s when you configured DICOM Server as 'sessions_url'.

Configuration:reverse proxy



Configuration:reverse proxy

```
<entry key="arcspec">/opt/xnat/cache/archive_specification.xml</entry>
<entry key="xnat_url">http://my-hidden-host.local:8080/xnat</entry>
<entry key="user">admin</entry>
<entry key="password">adminpw</entry>

<entry key="http_port">8180</entry>
<entry key="xnat_public_url">https://my.xnat.org</entry>
<entry key="scp_public_host">my.xnat.org</entry>
<entry key="scp_public_port">104</entry>
<entry key="sessions_url">https://my.xnat.org:8443/session</entry>

<entry key="log4j.rootLogger">WARN,R</entry>
  ...
```

Troubleshooting

- Three places to find diagnostic info:
- DicomServer log (and received file log)
- upload applet Java console
- session management web service

Troubleshooting FAQs

Where did my upload go?

- If DicomServer can’t determine to which project a session should go, it goes to the Unassigned prearchive
- Only admin accounts can view the Unassigned prearchive
- If the session isn’t in Unassigned, check main DicomServer log and received log

Troubleshooting

Why didn’t my session autoarchive?

Common reasons:

- The session is already archived in that project (Study Instance UID)
- The session label is already used in that project
- The project isn’t set to autoarchive

Troubleshooting

Why did my study get split?

- DICOM C-STORE protocol doesn't carry any information about study size
- DicomServer has to guess when the sender has finished with a study
- Once DicomServer starts autoarchiving a study, newly received files go into the prearchive

Summary

- XNAT uses DicomServer to receive DICOM data
- Users shouldn’t (and usually don’t) need to know about DicomServer

Links

DicomServer:

http://www.xnat.org/DicomServer

DicomBrowser:

http://nrg.wustl.edu/projects/DICOM/DicomBrowser.jsp

http://nrg.wustl.edu/projects/DICOM/DicomBrowser-cli.html

XNAT Gateway:

http://www.xnat.org/XNAT-DICOM+Gateway

Questions?

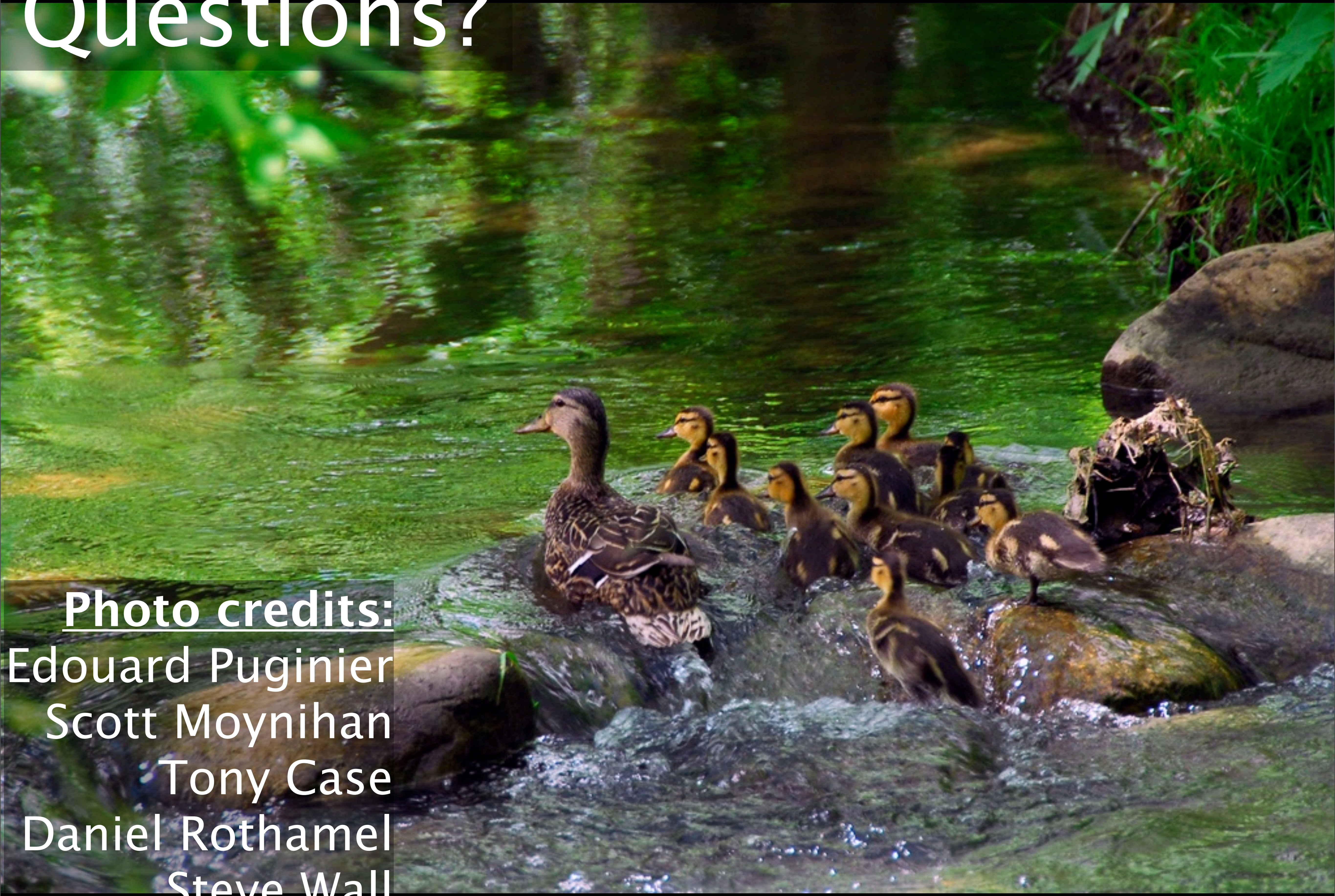


Photo credits:

Edouard Puginier
Scott Moynihan
Tony Case
Daniel Rothamel
Steve Wall