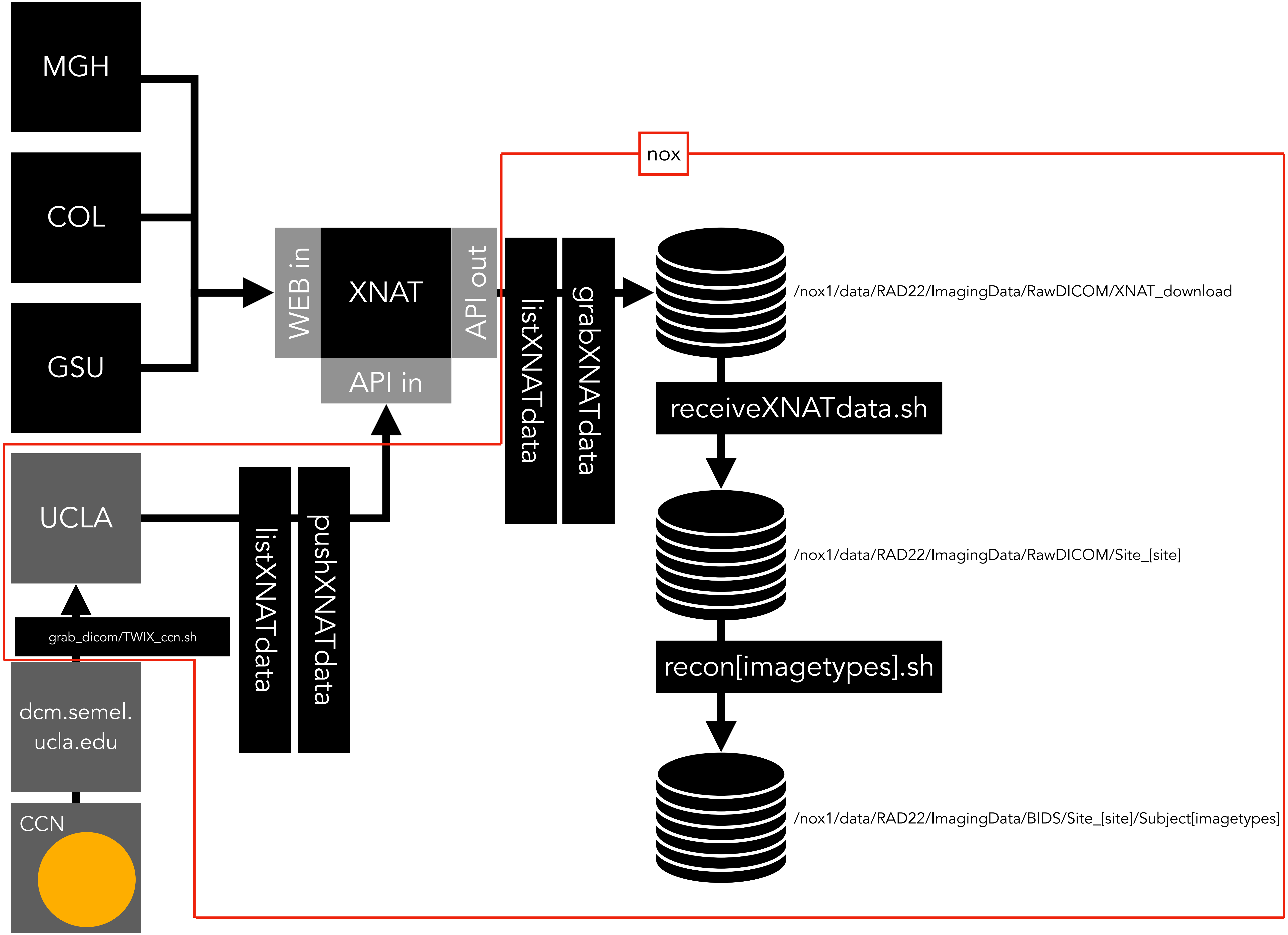
RADCO

Image Processing Pipelines - XNAT to reconstruction of NII

Robert Welsh, 2024-06-03



MGH

COL

GSU

WEB in

XNAT

API out

API in

UCLA

grab_dicom/TWIX_ccn.sh

dcm.semel.
ucla.edu

CCN

listXNATdata

pushXNATdata

nox

listXNATdata

grabXNATdata

/nox1/data/RAD22/ImagingData/RawDICOM/XNAT_download

receiveXNATdata.sh

/nox1/data/RAD22/ImagingData/RawDICOM/Site_[site]

recon[imagetypes].sh

/nox1/data/RAD22/ImagingData/BIDS/Site_[site]/Subject[imagetypes]

radco-pipelines-xnat-to-reconstruction Robert Welsh, 2024-06-03

API Specifications

listXNATdata list current data sets available on the XNAT, this is also use to query if data set is on XNAT

grabXNATdata grab data from XNAT for specific session and site

pushXNATdata push a specific dataset to the XNAT

receiveXNATdata.sh check /nox1 for new data from XNAT and place in proper site directory

recon[iamgetypes].sh recon data for a given data type

grab_dicom_ccn.sh process to grab dicom data from the UCLA dicom server

grab_TWIX_ccn.sh process to grab TWIX data from the UCLA dicom server

python

listXNATdata

returns a list of sessions on the XNAT

Usage

```
listXNATdata  -x [database] [-b] [-d] [-h]
```

Input

-x [database]	— name of XNAT database to query
-b	— list only those sessions with binary data files
-d	— debug flag
-h	— help option, shows this help

Output

returns list (one line per session) of MR session that are available on the XNAT database specified, to STDOUT

format is [session]
[session]

Example

```
listXNATdata -d RADCO

RAD22_COL_AE100-090Aka_01
RAD22_COL_AE100-133A_01
...
```

Error exit codes

0 - no error
1 - XNAT database does not exist
9 - XNAT connection error

**[database] could be RADCO or sync_test
*should read configuration from ${XNAT_CONFIG} which will be a .json file

python

grabXNATdata

grab a specific dataset (single session) from XNAT

Usage

```
grabXNATdata  -x [database] -s [session] [-b] [-d] [-h]
```

Input

-x [database] — name of XNAT database to query
-s [session] — session to grab
-b — grab binary data only
-d — debug flag
-h — help option, shows this help

Output

returns all data related for that session into the present working directory

Example

```
grabXNATdata -d RADCO -s RAD22_GSU_BU101_140AKd_01
```

Error exit codes

0 - no error
1 - XNAT database does not exist
2 - XNAT session does not exist
9 - XNAT connection error

**[database] could be RADCO or sync_test
*should read configuration from ${XNAT_CONFIG} which will be a .json file

python

pushXNATdata

push a specific dataset (single session) from XNAT

Usage

```
pushXNATdata  -x [database] -s [session] [-b [file]] [-d] [-h]
```

Input

-x [database] — name of XNAT database to query
-s [session] — session to grab
-b [file] — push the named binary file (TWIX or PDF)
-d — debug flag
-h — help option, shows this help

Output

push a session from the present working directory into the XNAT

Example

```
pushXNATdata -d RADCO -s RAD22_UCLA_BU101_180AKd_01
```

Error exit codes

0 - no error
1 - XNAT database does not exist
2 - local session does not exist
3 - XNAT session already exists
9 - XNAT connection error

**[database] could be RADCO or sync_test
*should read configuration from ${XNAT_CONFIG} which will be a .json file

bash

grab_dicom/TWIX_ccn.sh

cron processes to grab data from CCN

Usage

```
grab_dicom_ccn.sh
grab_TWIX_ccn.sh
```

Input

none

Output

pulls data automatically, started by cron

Error exit codes

bash

receiveXNATdata.sh

process a list of sessions to be readied for recon

bash

recon[imagetypes].sh

cron jobs for reconstruction of images

directory tree under /nox2/data/RAD22

```
Scripts/
├── Analysis
├── Audit
├── DataOrg
│   ├── jobs_20240611
│   ├── LOGS
│   │   ├── cron
│   │   └── grab
│   ├── readyToProcess
│   ├── sessionLists
│   ├── tmpFileList
│   └── xnat_scripts
│       ├── env
│       └── __pycache__
├── PreProcess
│   ├── jobs_freesurfer_20240611
│   │   └── LOGS
│   └── jobs_rest_20240611
│       └── LOGS
├── QC
└── Recon
    ├── jobs
    └── LOGS
        ├── cron
        └── recon
```

${RAD22_CONFIG}

The environmental variable RAD22_CONFIG will point point a .json file that contains specific information about the XNAT server and the RADCO experiment This includes the following information:

do

```
export RAD22_CONFIG=/nox2/data/RAD22/rad22_config.json
```

rad22_config.json

```
{
    "descriptor" : [
              {
                  "name"     : "Robert Welsh",
                  "email"    : "rcwelsh@g.ucla.edu",
                  "version" : 1.0,
                  "this_file" : "rad22_config.json"
              }
    ],
    "xnat_server"  : "xnatccn.semel.ucla.edu",
    "exp_dir"      : "/nox2/data/RAD22",
    "series_list" : "expectedSeriesList.txt"
}
```