Web Services for X-ray and Optical Data Analysis

Andy Ptak (JHU)
Simon Krughoff (UW)
Andy Connolly (UW)

http://www.xassist.org

Overview

- WESIX is a web service for running the source detection and photometry program SExtractor on (optical) images
- XAssist is a package for automatically analyzing X-ray data
- Currently have AISR funding, main goals:
  - Add web services to XAssist
  - Combine XAssist and WESIX into similar web services
  - Create a common framework for web service analysis of multi-wavelength data

WESIX

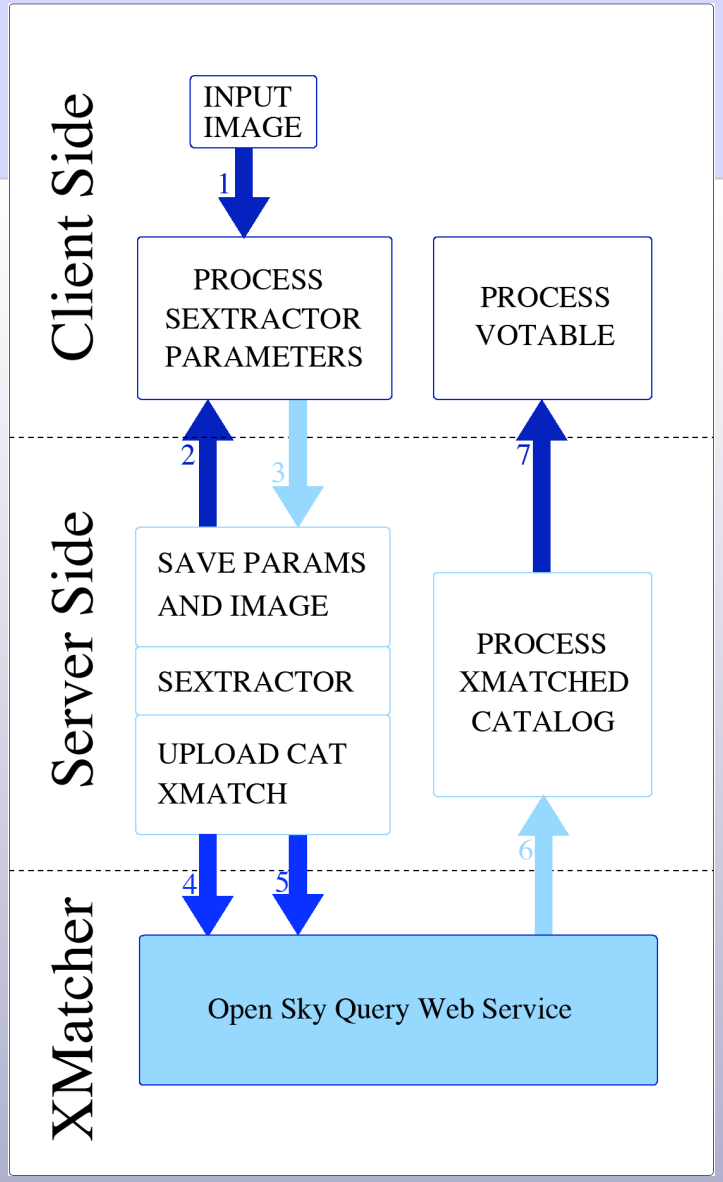
- WESIX is:
  - Web-Enabled Source Identification with X-matching
  - A web service with web page front end for extracting and cross matching sources in an astronomical image
- Inputs are a FITS file with extraction parameters and catalog fields for output
- Uses the SkyNode protocol from IVOA for cross matching with published catalogs.

XAssist

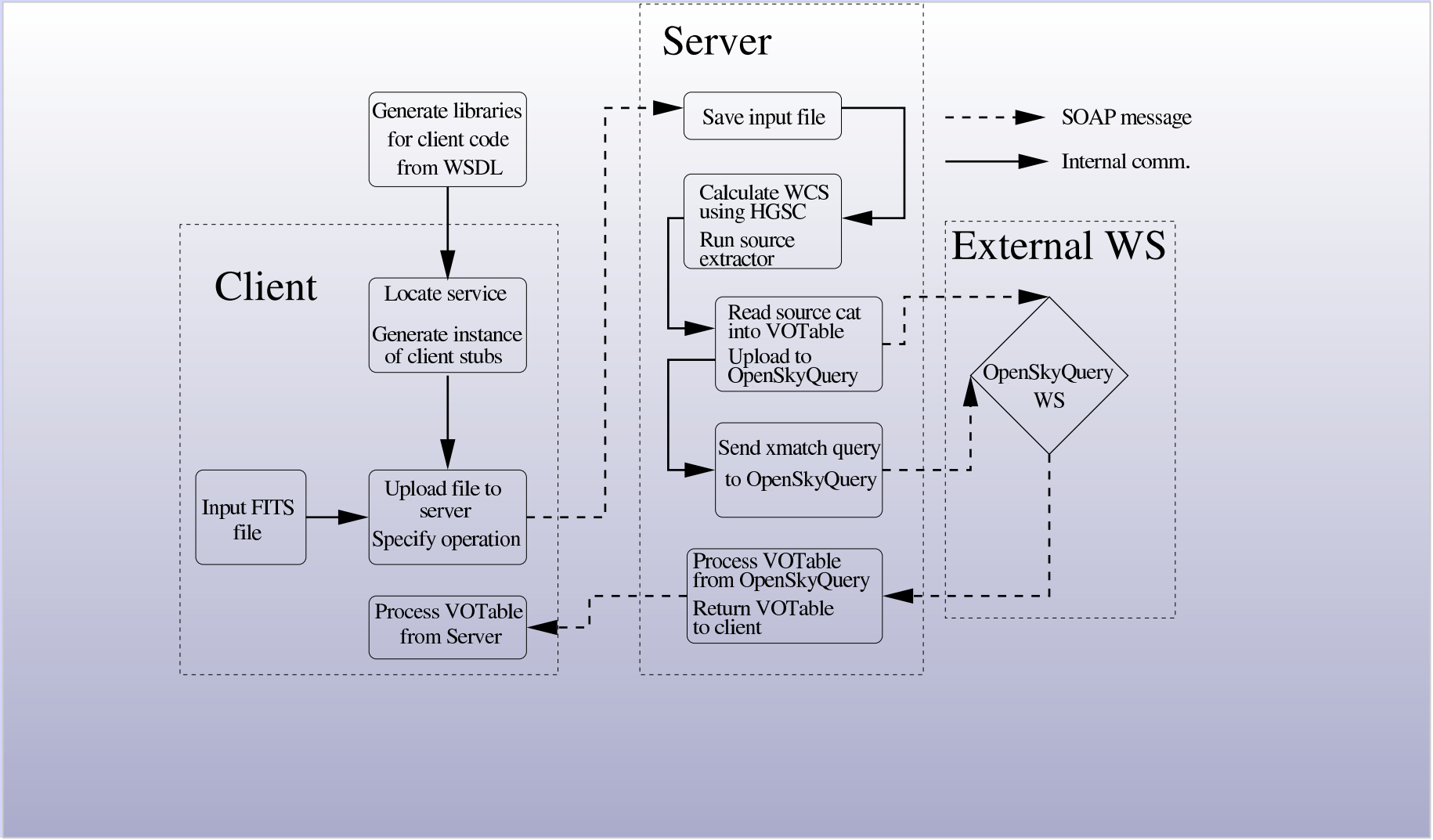
- Started as a previous AISR project (1998-2001)
- Performs data reduction, source detection, source characterization, extraction of spectra and images for each source
- Written mostly in Python and scripts existing mission-specific software as much as possible (CIAO for *Chandra*, XMM-SAS for *XMM-Newton*, HEADAS for *Suzaku*)
- XAssist running pipelines to process *Chandra* and *XMM-Newton* data
  - Pipeline source lists searchable via HEASARC, which links search results to field reports at XAssist web site
  - Japanese/US mission *Suzaku* support recently added, *Suzaku* pipeline to be started shortly

WESIX Structure

1. Read FITS image with WCS
2. Request default parameters
3. Send image and parameters
4. Upload source list
5. Send ADQL query
6. Receive XMatched catalog
7. Return catalog to client.



WESIX Workflow



Current WESIX Development

- Transitioning from SOAP, Java, Axis → Python, XML-RPC
  - Also considering JSON
- Improvements:
  - No SOAP implementation issues
  - Multiple input images for weighted source identification and detect+measure images
  - Vector quantities for measurement of multiple apertures
  - Generalized framework for interaction with other source identification applications (starting with XAssist)

Lessons from SOAP

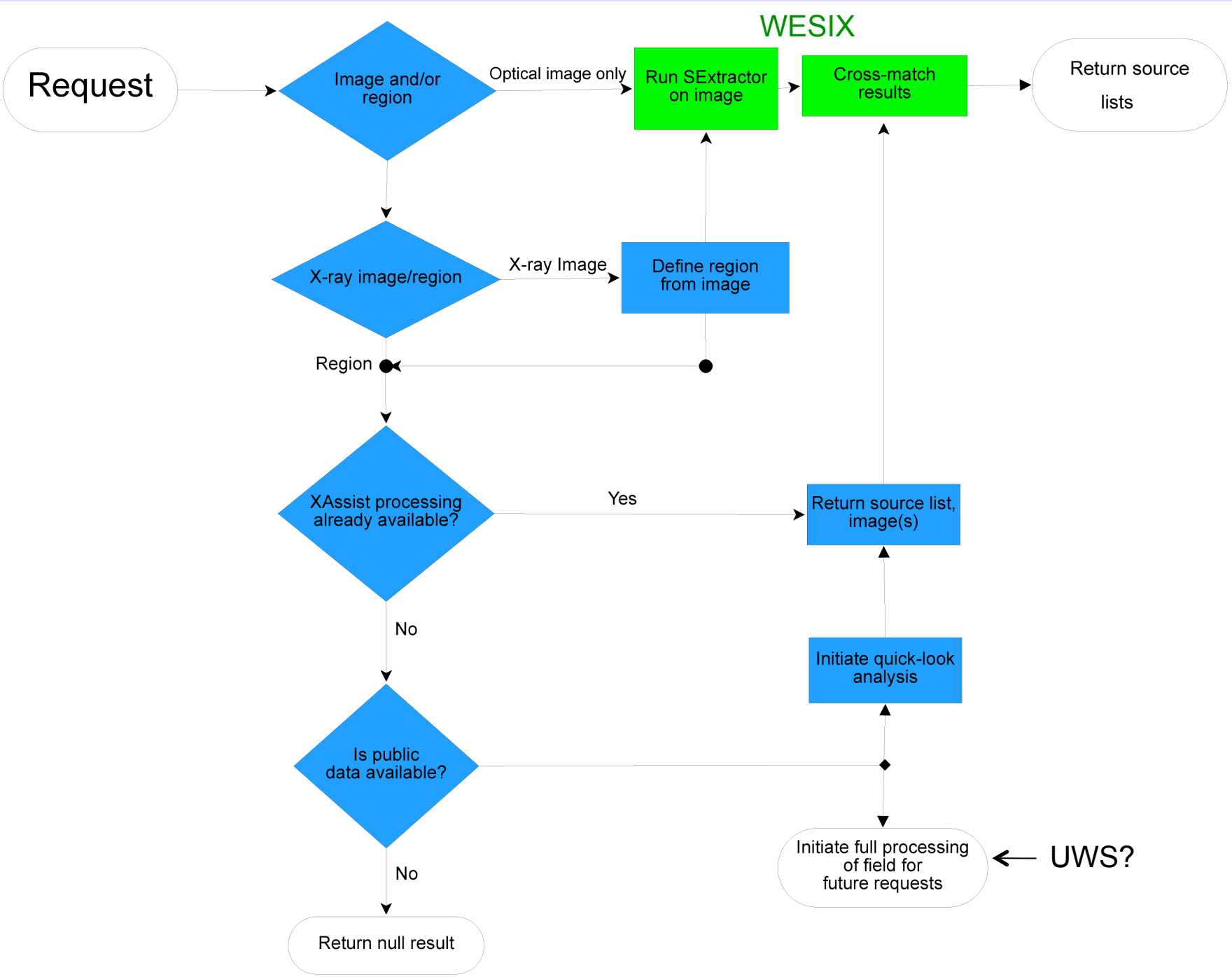
- Autogeneration – Most SOAP problems stem from limitations in implementation of code generation tools.
- Mismatch between auto-generated WSDLs for complex types (e.g. VOTable) in .NET and Java/Axis
- Inability for code generators to deal with recursions (which are allowed in SOAP standard)
- Minimal to no support for advanced technologies (SOAP with attachments)
- Bottom line: SOAP should be easy, but isn't.

Current XAssist Development

- Adding web service for querying pipeline status, searching pipeline database and requesting processing (initial versions done and undergoing testing)
- Exposing individual XAssist processing steps as web services
  - Exposure at a given position ("footprint")
  - Extracting spectra and image for a given source or position
    - Computing Bayesian confidence intervals for source significance
- Adding "quick-look" processing option
  - SExtractor for source detection
  - Streamlined (and often approximate) versions of other processing steps

Python XML-RPC

- Python standard library xmlrpclib contains both client and server classes that are very easy (Python SOAP support has been sluggish)
- All functionality will be mirrored in web applications and RESTful services, via cherrypy (probably) or django
- Planning several levels of RPC
  - Admin: process control for available cpus
  - User: request processing of a given field, query pipeline data, compute upper-limit, etc.
  - Internal: communication between processes running locally within cluster/grid (migrate to SAMP?)


WESIX
Request
Image and/or region
Optical image only
Run SExtractor on image
Cross-match results
Return source lists
X-ray image/region
X-ray Image
Define region from image
Region
XAssist processing already available?
Yes
Return source list, image(s)
No
Initiate quick-look analysis
Is public data available?
Initiate full processing of field for future requests
UWS?
No
Return null result

Joint WESIX/XAssist Development

- Testing/calibration of SExtractor on X-ray images
  - If X-ray image supplied to WESIX, run WESIX with X-ray specific defaults
- Add options to WESIX parameter input to allow X-ray dataset to be supplied
  - Spawn request to XAssist web service to check for existing processing of field
    - Yes: return data
    - No: start quick-look processing and add field to queue for full processing
  - Allow PGP key to be supplied for proprietary data

Future Plans

- AJAX GUI for XAssist and WESIX
- Creating portal to allow users to specify source lists and/or regions to monitor for available data, data processed automatically to get source lists / upper-limits
- Web service access to XAssist and WESIX lends itself to distributed processing of X-ray and optical data
  - Will start joint analysis of *Chandra*, *XMM* and optical data
    - Optical images archived at major observatories
    - HST overlap with *Chandra* and *XMM*

Summary

- WESIX and XAssist are separately being developed to be more flexible and capable
- Joint web service access (xml-rpc probably) to both will open up multi-wavelength virtual observatory analysis capability using “intelligent” systems
  - Distributed analysis of large datasets that are public
  - Correlations even when there are “upper-limits” and extended sources (often precludes simply using catalogs)