Implementing UWS for CEA

Paul Harrison

JBCA, University of Manchester

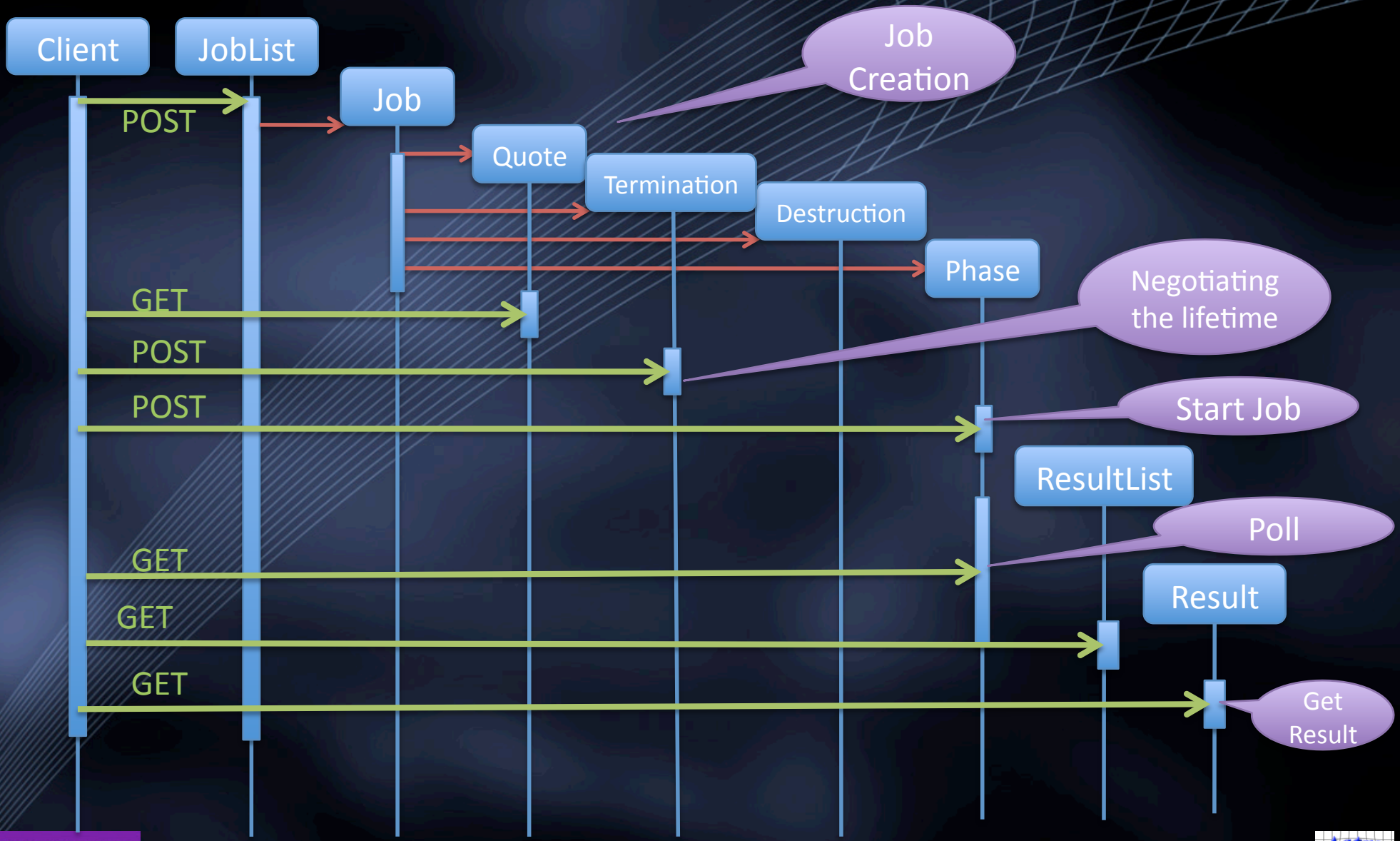
UWS and CEA?

- Universal Worker Service
  - A pattern for controlling a asynchronous stateful jobs
    - Has REST and SOAP interface definitions
- Common Execution Architecture
  - A pattern for controlling a asynchronous stateful jobs
    - An abstraction from underlying technologies
    - Has SOAP interface definitions
  - A model of an “application” and its parameters (UWS-PA)
  - Is implemented by Astrogrid as component known as a “CEC”
    - A container that makes applications VO compliant.

UWS Objects


Joblist
Job
Phase
Quote
Termination Time
Destruction Time
ResultList
Result
Result

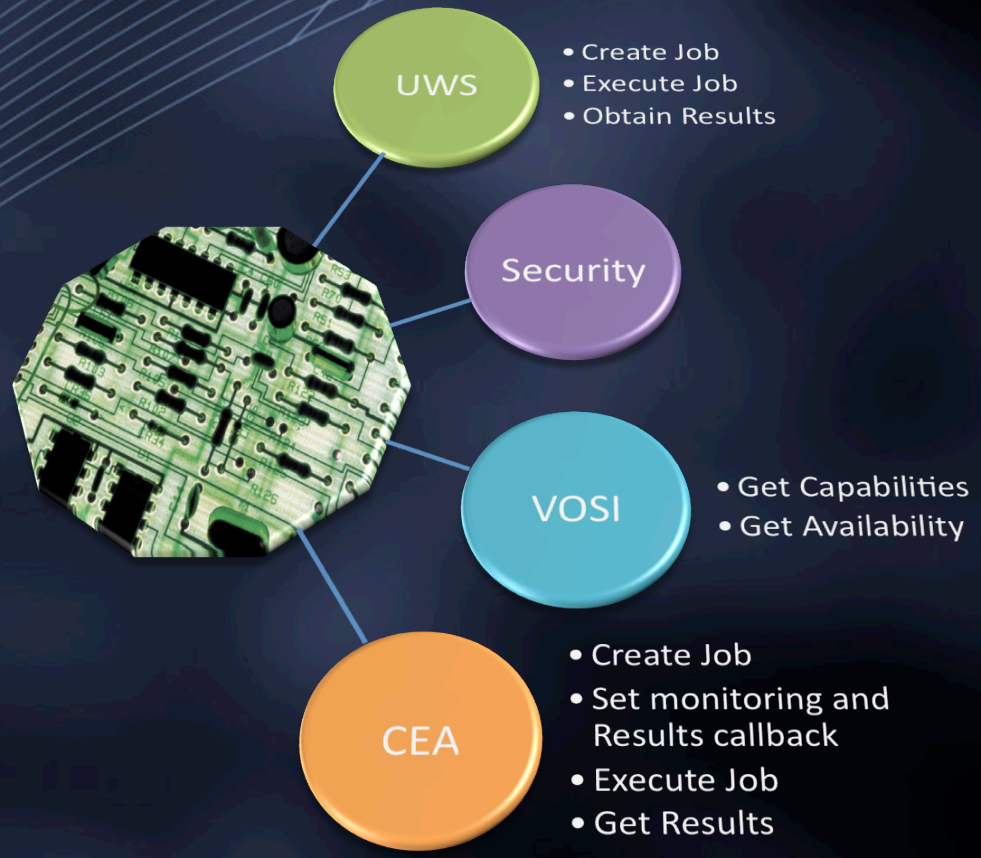
UWS calling pattern



CEA CEC Component

Common Execution Controller

- J2EE Component using
  - Spring
  - Axis 1.3 for SOAP
  - JAXB2 for XML binding
  - Chiba for XForms



Demo –To Show

- Driving same core from CEA (SOAP) and UWS (REST) interfaces
  - Driving from a Browser
  - Driving from the commandline
  - Driving from VODesktop
- Features
  - Asynchrony
  - Job control
    - Automatic killing
    - Automatic destruction
  - Putting results in VOSpace
  - Retrieving results directly – VOSpace not needed.
- Applications
  - Demo Application – Test application to show job control
  - Real Application – MERLIN imager.

Proposed Changes UWS 0.3

1. Destruction Time object – the time when the job description and any results will be deleted. Termination time to be a CPU duration.
   - /(jobs)/(jobid)/destruction
2. Control of job execution is done via POST to the phase – run, abort
3. New phase – ABORTED
4. Error object – any errors associated with the job can be obtained with a HTTP GET
   - /(jobs)/(jobid)/error

   More detail in http://www.ivoa.net/internal/IVOA/AsynchronousHome/ImplementingUWS-0.1.pdf

Issues to be resolved

- Schema
  - Job and result list
  - Extensions of the schema.
  - +WSDL for the SOAP version…
- Specify the name of the http parameter for all POST style operations
- Specify what should be returned for each operation
  - A 303 redirect to GET the URI just POSTed to?
- What is the Quote "don't know" value.
- Results before completion – potentially useful, but was disallowed by the 0.3 version.
- UWS interface type in the Registry Schema Model.
- UWS HTML representation
  - How complete the UI?
  - What technologies can be assumed for building the UI?
- Security – who is allowed what access to a job.
- (Pseudo-)Synchronous operation?

Conclusions

- UWS relatively easy to implement
  - REST style – easy to drive.
- UWS Pattern allows
  - Shorter Standards to be written
    - Only the specifics of the particular protocol need to be specified – all detail of how to invoke the service is contained within the pattern.
  - Generic software components to be written both on client and server that can deal aspects of several protocols.
- Astrogrid CEA is a potential "DAL v2 Toolkit"
  - Written in a modular component style.
    - Astrogrid DSA already built around CEA core.