A PPARC funded project

Making Applications VO-Aware

Noel Winstanley
Jodrell Bank, AstroGrid

John Taylor
ROE, Votech

Talk Plan

- Summary of what services AstroGrid provides
- Ways to access AstroGrid services
- Workbench
- Astro Client Runtime (ACR)
- Connecting to ACR
- Code Examples
- Uses for the ACR

What can AstroGrid do for you?

AstroGrid

- A nearly complete Virtual Observatory (VO) system
- Release 1.2 - http://software.astrogrid.org
- Built upon Web Services – callable via SOAP
- Conforms to international (IVOA) standards where defined, elsewhere proposes new standards
- Useful for real work
- Security -final bit unfinished
  - Authentication & Authorisation

Registry

- A hierarchical database (XML)
- Contains records that describe
  - data collections (coverage, catalogue structure, access methods)
  - remote applications (purpose, parameters, invocation methods)
  - supporting web services – storage, security, etc
  - other useful resources – e.g. client-side applications (soon)
- Used to locate (resolve) all other VO Web Services
- Interrogate using XQuery, Keywords, or ADQL
- Records conform to IVOA standard schema
- Exchanges records with registries in other VO projects (harvesting)

IVO Resource Names (Ivorns)

- Each uniquely define a resource in a VO Registry
- Used in workflows and scripts to refer to
  - remote applications (CEA applications, SIAP Services)
  - remote files (in MySpace, VOStore, etc)
  - also used to identify users and other resources
- `ivo://<authority>/<name>#<more data>` - general form
- `ivo://org.astrogrid/galaxev` – CEA application
- `ivo://uk.ac.le.star/filestore-001` – myspace filestore
- `ivo://uk.ac.le.star/noelwinstanley` – a user (me)
- `ivo://uk.ac.le.star/noelwinstanley#votable/result.vot`
  - a file in my myspace (rooted from my home directory)

Myspace

- Distributed, location transparent, file storage
  - Each user has a single folder hierarchy – maintained by a *filemanager* service
  - Files in the hierarchy may be stored at different locations – each location is a *filestore*
  - clients typically interact only with the filemanager.
- AstroGrid services can read / write to Myspace
  - Place to stage results of long-running queries & computations
  - Used as a buffer for intermediate products of workflows
- Enables data to be kept near processing tools
- Being standardized in IVOA as VOSpace / VOStore

Remote Applications (CEA)

- Uniform method of registering and deploying remote applications. Encompasses:
  - dataset access (e.g. publishing a catalogue database or image collection)
  - data processing (e.g. X-matching, source extraction, simulations)
- Asynchronous invocation
  - service provides progress monitoring, notifications, and control
  - results can be retrieved from the service, or staged to myspace, ftp server, etc.
- CEA services can be called from scripts, UI, and workflows
- Being standardized as IVOA 'Universal Worker Service'

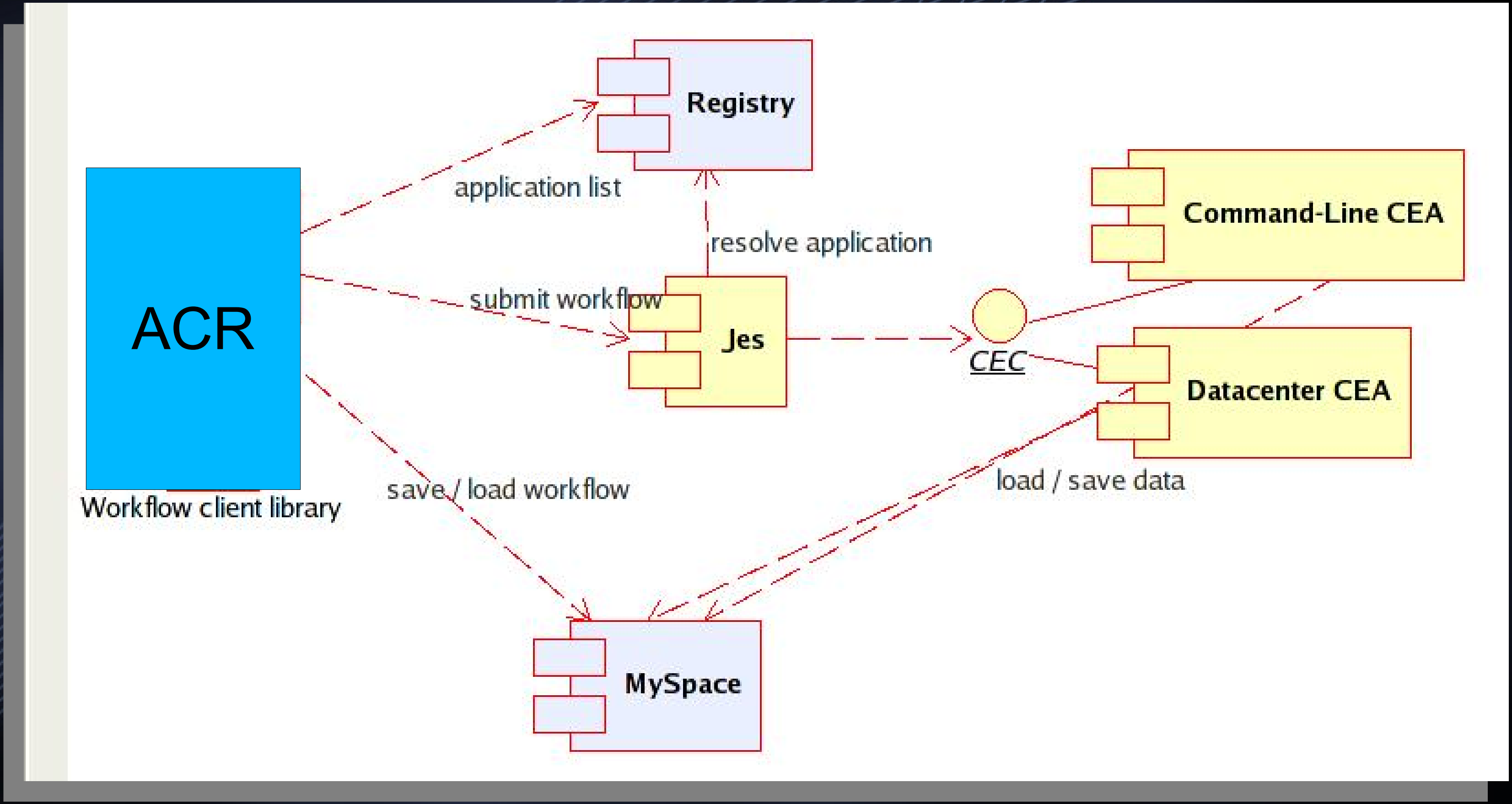
Workflow

- a workflow performs a complex piece of work
  - comprises one of more steps
  - steps can be composed in sequence and parallel
  - execution of steps controlled by loops and conditionals
- each step invokes a separate CEA application
  - input parameters – control and astronomical data
  - output parameters – results and logs, or intermediate for input to a subsequent step
- parameter values can be
  - inline (direct parameters)
  - references to external resources (indirect parameters)
    - http://..., ftp://..., ivo://... (myspace)
    - not file:/ (as file:/ resources are not accessible from remote services)

Executing Workflows

- Workflows are expressed as XML documents
- These documents are submitted to a Job Execution Server (JES) for execution
  - a new globally unique job ID is returned each time a workflow is submitted
- JES schedules & executes the workflow
  - decides on which CEA servers to execute steps
  - records log & results of step execution
  - evaluates workflow control structures to decide which steps to run next
  - executes workflow scripts – useful glue for control and light computation, (language is Groovy – interpreted Java)
- Workflow Documentation.
  - http://wiki.astrogrid.org/bin/view/Astrogrid/BuildingWorkflow

Workflow Execution – Service Collaboration



Accessing AstroGrid Services

How do Users work with AstroGrid?

- Two alternative user interfaces
- Portal – a web interface, accessed through browser
  - handy for occasional use
  - technical limitations of the web make it awkward for advanced tasks
- Workbench – GUI client
  - Java WebStart Application
  - richer user applications
  - also provides scripting access
- Info   http://software.astrogrid.org/userdocs/

How do Developers use AstroGrid?

- Three Alternatives
  - Call SOAP services directly, using WSDL
    - most basic – exposed to most complexity, necessary to understand services interact (e.g. resolution)
    - security – needs advanced SOAP handling.
  - Call AstroGrid delegate libraries
    - hides some complexity,
    - maybe not the cleanest or most reusable interface – developed for internal use.
    - Java-only – requires whole AstroGrid library stack
  - Call methods on the Astro Client Runtime (ACR)
    - Uniform facade interface to AstroGrid.
    - Simpler to learn & provides extra whistles and bells.
    - Language Neutral
- Info http://software.astrogrid.org/developerdocs/

Workbench

Definitions

- **Workbench** is a GUI pro[illegible] for working with AstroGrid
  - Available for [illegible], Solaris, Mac
  - Single-click lau[illegible] using Java WebStart
  - http://software.astrogrid.org/userdocs/workbench.html
  - Choose 'Stable Version'

Try it now!

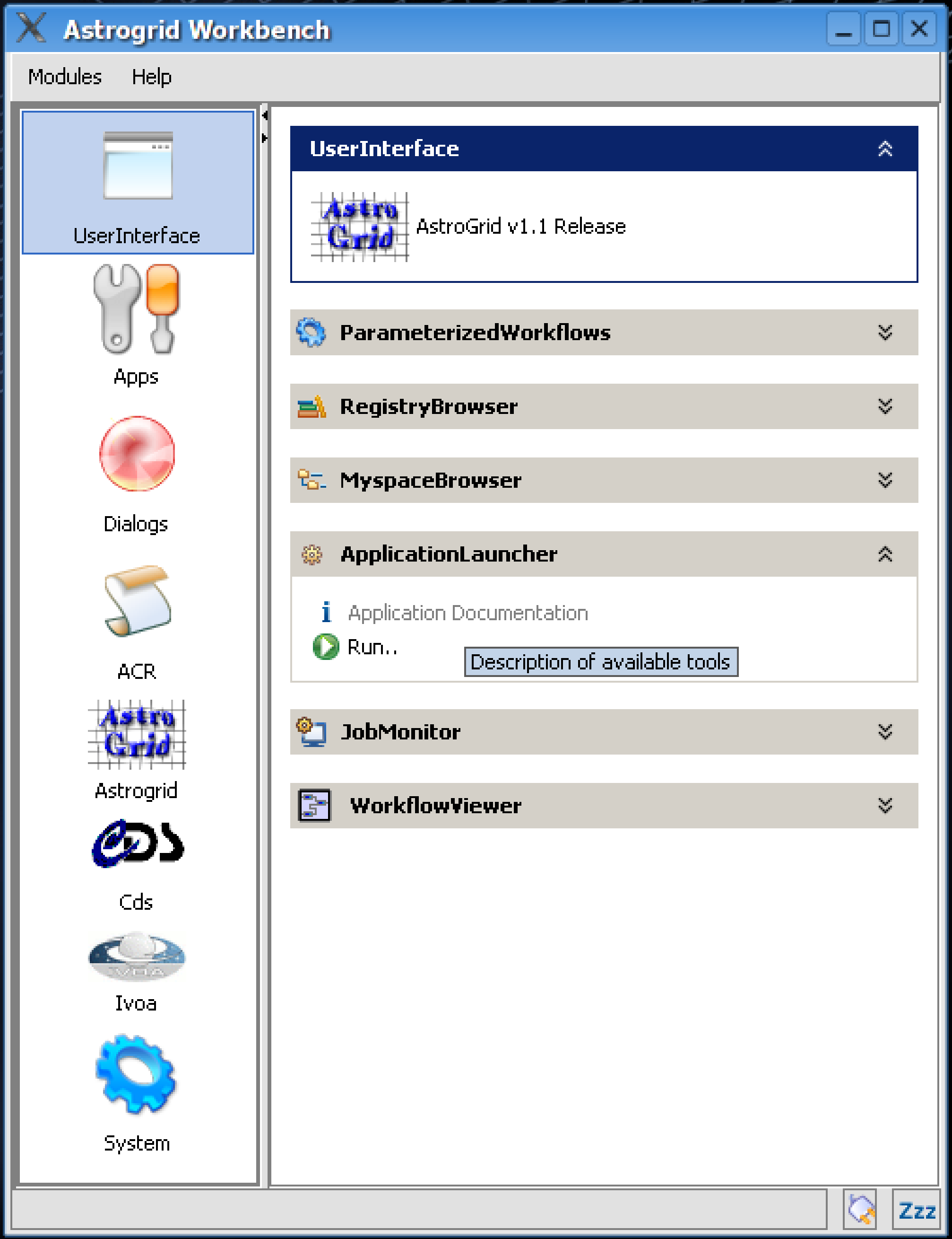
- **Astro Client Runtime (ACR)** is the library upon which the Workbench is built
  - provides simple access to all VO services
  - when the Workbench is running, the underlying ACR instance is accessible from other programs and scripts

Java WebStart

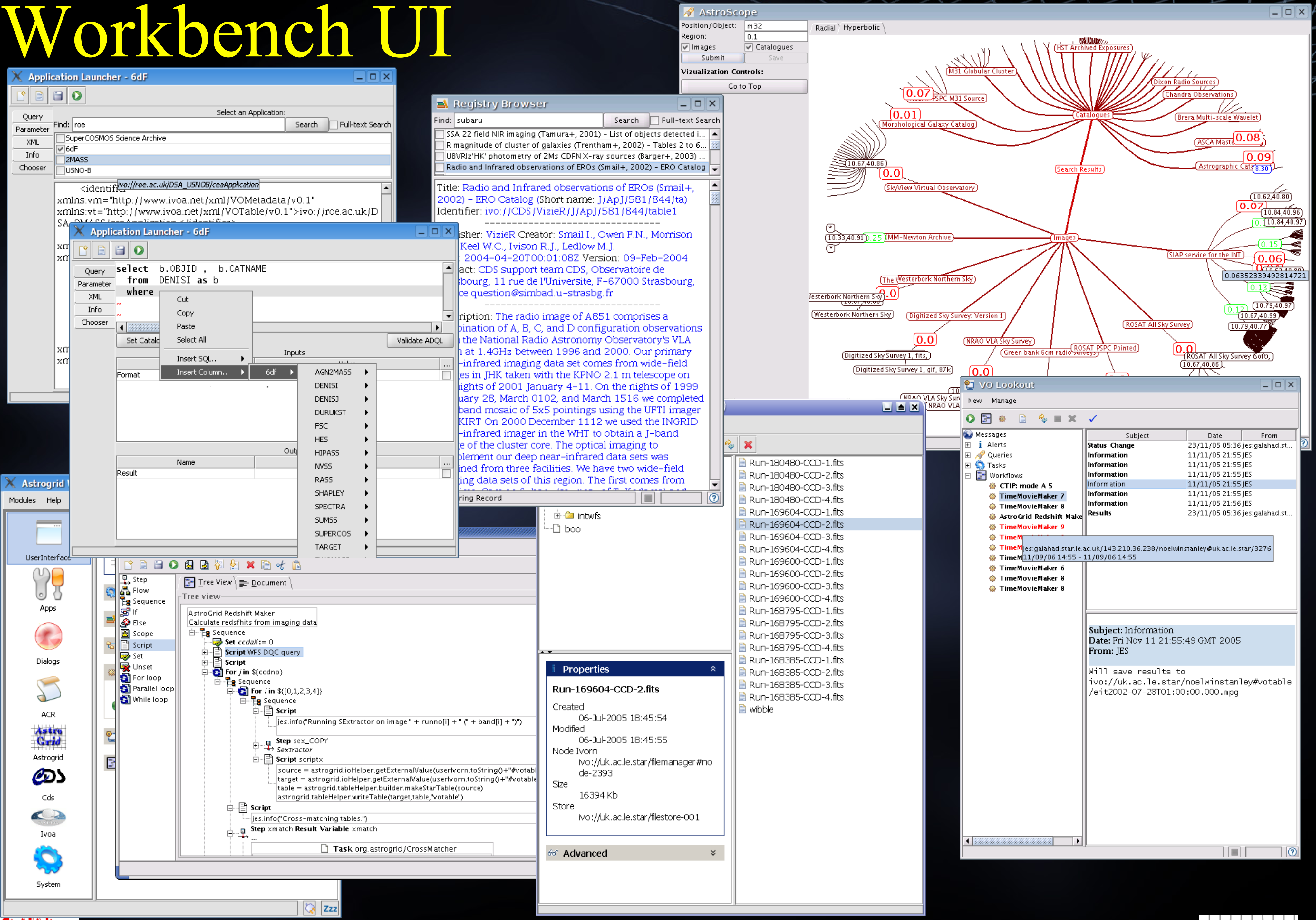
- Safely launches desktop applications by clicking browser hyperlinks
- The application is automatically downloaded and installed the first time it is launched.
  - Cached so that is starts instantly the next time.
  - When a new version released, the cached version of the application is updated
  - You always get the most uptodate version, no maintenance worries, no patches
- Applications and libraries are signed by the producing organization – you need to accept the trust dialogues.
- Handles native libraries, different operating systems, desktop integration.

Workbench

- Launcher
- Lists a suite of GUI Applications
- Also exposes some of the basic functions of the ACR
  - good for experimentation



Workbench UI



- SSA 22 field NIR imaging (Tamura+, 2001) - List of objects detected i...
- R magnitude of cluster of galaxies (Trentham+, 2002) - Tables 2 to 6...
- UBVRIz'HK' photometry of 2Ms CDFN X-ray sources (Barger+, 2003) ...
- Radio and Infrared observations of EROs (Smail+, 2002) - ERO Catalog

Title: Radio and Infrared observations of EROs (Smail+, 2002) - ERO Catalog (Short name: J/ApJ/581/844/ta)
Identifier: ivo://CDS/VizieR/J/ApJ/581/844/table1

isher: VizieR Creator: Smail I., Owen F.N., Morrison Keel W.C., Ivison R.J., Ledlow M.J. 2004-04-20T00:01:08Z Version: 09-Feb-2004
act: CDS support team CDS, Observatoire de sbourg, 11 rue de l'Universite, F-67000 Strasbourg, ce question@simbad.u-strasbg.fr

ription: The radio image of A851 comprises a bination of A, B, C, and D configuration observations the National Radio Astronomy Observatory's VLA at 1.4GHz between 1996 and 2000. Our primary -infrared imaging data set comes from wide-field es in JHK taken with the KPNO 2.1 m telescope on ights of 2001 January 4-11. On the nights of 1999 uary 28, March 0102, and March 1516 we completed band mosaic of 5x5 pointings using the UFTI imager KIRT On 2000 December 1112 we used the INGRID -infrared imager in the WHT to obtain a J-band e of the cluster core. The optical imaging to plement our deep near-infrared data sets was ined from three facilities. We have two wide-field ing data sets of this region. The first comes from

```
<identifi
xmlns:vm="http://www.ivoa.net/xml/VOMetadata/v0.1"
xmlns:vt="http://www.ivoa.net/xml/VOTable/v0.1">ivo://roe.ac.uk/D
```

```
select  b.OBJID ,  b.CATNAME
  from  DENISI as b
  where
```

```
AstroGrid Redshift Maker
Calculate redsfhits from imaging data
Sequence
  Set ccdall := 0
  Script WFS DQC query
  Script
  For j in ${ccdno}
    Sequence
      For i in ${[0,1,2,3,4]}
        Sequence
          Script
            jes.info("Running SExtractor on image " + runno[i] + " (" + band[i] + ")")
          Step sex_COPY
            Sextractor
          Script scriptx
            source = astrogrid.ioHelper.getExternalValue(userIvorn.toString()+"#votab
            target = astrogrid.ioHelper.getExternalValue(userIvorn.toString()+"#votable
            table = astrogrid.tableHelper.makeStarTable(source)
            astrogrid.tableHelper.writeTable(target,table,"votable")
    Script
      jes.info("Cross-matching tables.")
    Step xmatch Result Variable xmatch
      Task org.astrogrid/CrossMatcher
```

Run-180480-CCD-1.fits
Run-180480-CCD-2.fits
Run-180480-CCD-3.fits
Run-180480-CCD-4.fits
Run-169604-CCD-1.fits
Run-169604-CCD-2.fits
Run-169604-CCD-3.fits
Run-169604-CCD-4.fits
Run-169600-CCD-1.fits
Run-169600-CCD-2.fits
Run-169600-CCD-3.fits
Run-169600-CCD-4.fits
Run-168795-CCD-1.fits
Run-168795-CCD-2.fits
Run-168795-CCD-3.fits
Run-168795-CCD-4.fits
Run-168385-CCD-1.fits
Run-168385-CCD-2.fits
Run-168385-CCD-3.fits
Run-168385-CCD-4.fits
wibble

Run-169604-CCD-2.fits
Created
06-Jul-2005 18:45:54
Modified
06-Jul-2005 18:45:55
Node Ivorn
ivo://uk.ac.le.star/filemanager#node-2393
Size
16394 Kb
Store
ivo://uk.ac.le.star/filestore-001

Subject	Date	From
Status Change	23/11/05 05:36	jes:galahad.st...
Information	11/11/05 21:55	JES
Information	11/11/05 21:55	JES
Information	11/11/05 21:55	JES
Information	11/11/05 21:55	JES
Information	11/11/05 21:55	JES
Information	11/11/05 21:56	JES
Results	23/11/05 05:36	jes:galahad.st...

jes:galahad.star.le.ac.uk/143.210.36.238/noelwinstanley@uk.ac.le.star/3276
11/09/06 14:55 - 11/09/06 14:55

Subject: Information
Date: Fri Nov 11 21:55:49 GMT 2005
From: JES

```
Will save results to
ivo://uk.ac.le.star/noelwinstanley#votable
/eit2002-07-28T01:00:00.000.mpg
```

THE UNIVERSITY of MANCHESTER

Noel Winstanley - nw@jb.man.ac.uk

Astro Client Runtime

Astro Client Runtime (ACR)

- A library of virtual-observatory functions
- A common facade for the VO / AstroGrid
  - aim to integrate all VO standards, popular ad-hoc services, and suitable helper functions.
- uniform abstraction level and types
  - cleaner API, less special cases, lower learning curve
- single configuration
  - taken care of – client programmer doesn't need to care.
- simple deployment
  - part of workbench, trivial to install using Java WebStart
- Shared component – single signon, cached registry entries, myspace trees

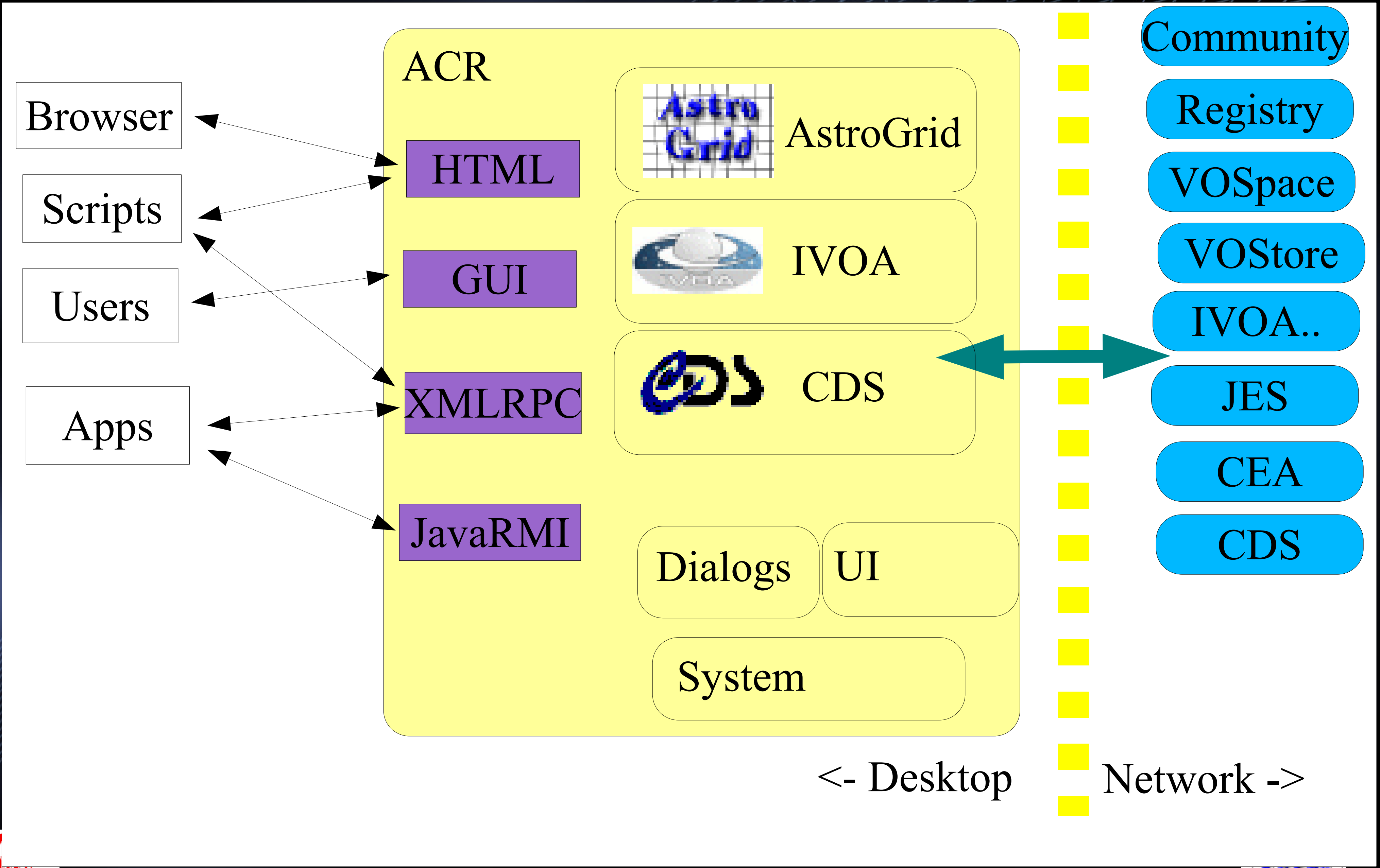
Design

- ACR designed to be accessible from all programming languages
- Procedural design, rather than OO (astronomer friendly)
- A service that runs on the user's desktop
  - accepts requests from other desktop applications
  - processes requests by calling webservices using the AstroGrid Java client libraries.
- Components
  - ACR provides a large set of components / services that can be called by any of the access methods
  - related components organized into modules.

Access Methods

- JavaRMI (Java, Groovy, Jython)
  - JVM-only inter-process communication
  - strongly typed
  - requires a minimal set of libraries
  - allows remote event listeners to be registered
- XMLRPC (Python, Perl, C++, C# , Java)
  - Forerunner of SOAP: http://www.xmlrpc.com/
  - simpler types than SOAP
  - implementations for a wide range of languages
- HTTP-Get (Shell, R, IDL, Matlab)
  - rough-n-ready procedure call
  - fallback for other languages

ACR Schematic



ACR Abilities

- AstroGrid
  - MySpace: read, write, list, create, delete
  - Registry: query, xquery, resolve
  - CEA & JES: query, build, execute, monitor
- IVOA – SIAP implemented, SSAP, SkyQuery to follow.
- VOTECH – plastic
- CDS – Simbad, Vizier, coordinates, UCD.
  - Vizier II to follow
- NVO – cone implemented, NED, other ad-hoc to follow
- UI
  - control workbench user interface
  - display registry / myspace dialogues to prompt for input

API Documentation

- Javadoc, with annotations for how to call from XML-RPC
- XMLRPC
  - everything is a string
- view docs

Connecting to ACR - Code

Java RMI

Import ACR classes

```java
import org.astrogrid.acr.Finder;
import org.astrogrid.acr.astrogrid.Registry;
import org.astrogrid.acr.builtin.ACR;
import org.astrogrid.acr.system.Configuration;

import java.net.URI;
import java.util.Iterator;
import java.util.Map;
```

Instantiate finder

Find running ACR, or execute new

Get reference to service

Alternative way to get service

Call service function

Tell program to exit

```java
public class Connect {

    public static void main(String[] args) {
      try {
        Finder f = new Finder();
        ACR acr = f.find();
        // retrieve a service - by specifying the interface class
        Configuration conf =
                (Configuration)acr.getService(Configuration.class);
        // call a method on this service.
        Map l = conf.list();
        for (Iterator i = l.entrySet().iterator(); i.hasNext(); ) {
            System.out.println(i.next());
        }
        // retrieve another service from the acr - this time by name
        Registry registry = (Registry)acr.getService("astrogrid.registry");

        // use this service..
        URI u = new URI("ivo://org.astrogrid/Pegase");
        System.out.println(registry.getResourceInformation(u));
        // returns a struct of data.
        // registry.getRecord(u) returns a org.w3c.dom.Document..

        u = new URI("ivo://uk.ac.le.star/filemanager");
        System.out.println(registry.resolveIdentifier(u));
        // returns a java.net.URL

      } catch (Exception e) {
            e.printStackTrace();
      }
      //shut the app down - necessary, as won't close by itself.
      System.exit(0);
    }
}
```

Python XML-RPC

Import xmlrpc library

Read ACR configuration file

Construct xmlrpc endpoint

Create client

Get reference to service

Call service function

```
#!/usr/bin/env python
# Noel Winstanley, Astrogrid, 2005
# minimal example of connecting to acr and calling a service.
import xmlrpclib
import sys
import os

#parse the configuration file.
prefix = file(os.path.expanduser("~/.astrogrid-desktop")).next().rstrip()
endpoint = prefix + "xmlrpc"
print "Endpoint to connect to is", endpoint

#connect to the acr
acr = xmlrpclib.Server(endpoint)

#get a reference to the registry service from the acr.
registry = acr.astrogrid.registry

#call a method
print registry.getResourceInformation('ivo://org.astrogrid/Pegase')
        # returns a struct of data

print registry.getRecord('ivo://org.astrogrid/Pegase')
        # return the xml of a registry entry (string)

print registry.resolveIdentifier('ivo://uk.ac.le.star/filemanager')
```

show output

Perl XML-RPC – same pattern

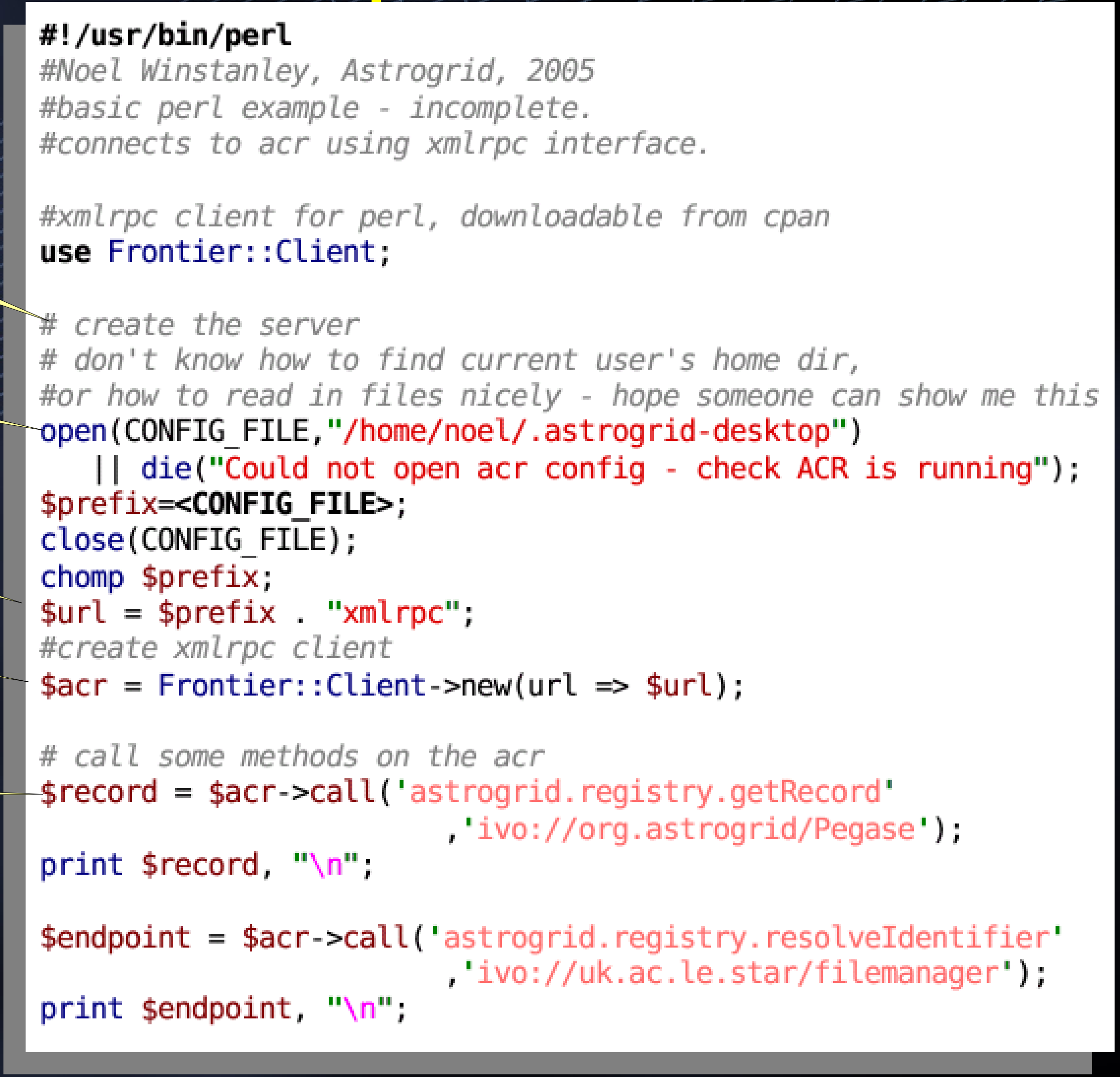
Import xmlrpc library
- alternatives?

Read ACR configuration file

Construct xmlrpc endpoint

Create client

Call service function



```perl
#!/usr/bin/perl
#Noel Winstanley, Astrogrid, 2005
#basic perl example - incomplete.
#connects to acr using xmlrpc interface.

#xmlrpc client for perl, downloadable from cpan
use Frontier::Client;

# create the server
# don't know how to find current user's home dir,
#or how to read in files nicely - hope someone can show me this
open(CONFIG_FILE,"/home/noel/.astrogrid-desktop")
   || die("Could not open acr config - check ACR is running");
$prefix=<CONFIG_FILE>;
close(CONFIG_FILE);
chomp $prefix;
$url = $prefix . "xmlrpc";
#create xmlrpc client
$acr = Frontier::Client->new(url => $url);

# call some methods on the acr
$record = $acr->call('astrogrid.registry.getRecord'
                    ,'ivo://org.astrogrid/Pegase');
print $record, "\n";

$endpoint = $acr->call('astrogrid.registry.resolveIdentifier'
                    ,'ivo://uk.ac.le.star/filemanager');
print $endpoint, "\n";
```

HTML Interface

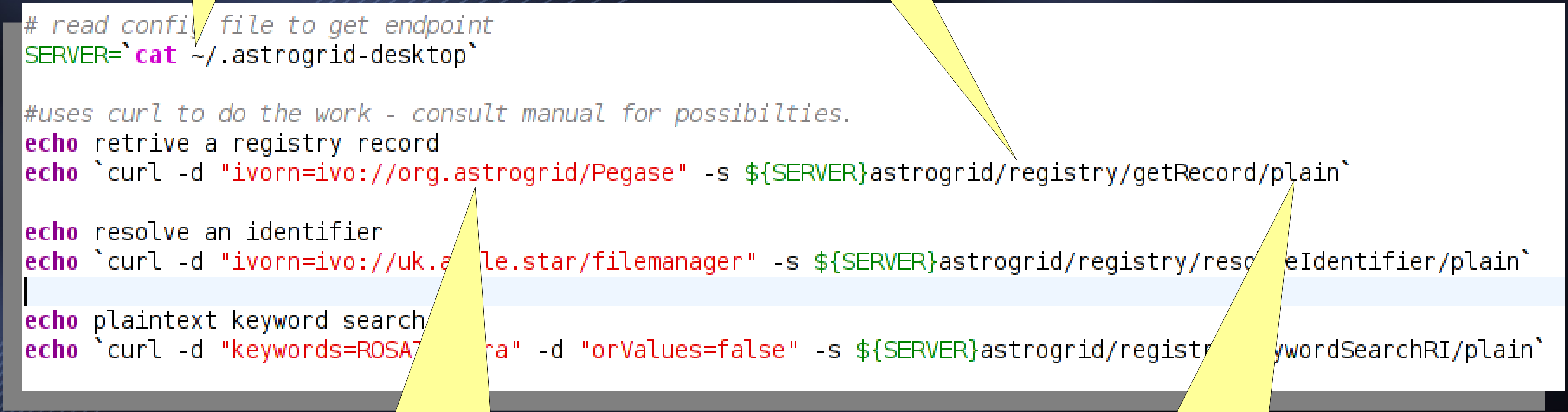
- Rudimentary UI exposing all ACR functionality
- generated HTML forms
- easy to browse available functions & try them out
  - especially useful for HTTP access method
- documentation not quite in sync or as complete as the javadoc
  - trust the javadoc
  - plan to have the html UI link to the javadoc in future.
- Also a GUI interface to some acr functions
  - possible this will go away soon.

show n tell

Shell – raw HTTP

Determine server endpoint

function name



```
# read confi  file to get endpoint
SERVER=`cat ~/.astrogrid-desktop`

#uses curl to do the work - consult manual for possibilties.
echo retrive a registry record
echo `curl -d "ivorn=ivo://org.astrogrid/Pegase" -s ${SERVER}astrogrid/registry/getRecord/plain`

echo resolve an identifier
echo `curl -d "ivorn=ivo://uk.a  le.star/filemanager" -s ${SERVER}astrogrid/registry/resc  eIdentifier/plain`

echo plaintext keyword search
echo `curl -d "keywords=ROSAT   ra" -d "orValues=false" -s ${SERVER}astrogrid/registr   ywordSearchRI/plain`
```

parameters

result format

show results

- s using HTML interface

Uses for the ACR

Possibilities

- Framework for Workbench UI.
- New VO Applications
  - Sampo (fi)
- Adding more VO functionality to existing applications
  - Topcat (uk), Aladin (fr), hope to see more soon
- PLASTIC – vizualization tool interop. See John.
- Scripting – (python, perl)
  - utilities and scripted workflows
  - system testing, administration
- Access VO into existing wrapper systems
  - PyRaf, Parceltongue

ACR Scripting

- VO Commandline
  - unix-ey small composable commandline programs.
  - `vols`, `voget`, `voput`, `reg-query`, `ls-jobs` ...
  - implemented as Python scripts calling the ACR
- Python (Perl/..) workflows
  - script contains control flow
  - performs work by querying data services & running CEA applications via ACR
  - more interactive / iterative development than batch JES workflows
  - can integrate desktop apps, ACR dialogues, etc

References

- Tutorial Webpage http://wiki.astrogrid.org/bin/view/Astrogrid/AgTechWorkshopJan06
  - **Scripting exercise**
  - API documentation
  - Getting started instructions for the different access methods
  - Code examples in Java, C++, Python, Perl
  - Dev docs also available from workbench in-program help.
- Homepage & WebStart link: http://software.astrogrid.org/userdocs/workbench.html

Future Plans

- Will maintain backwards compatibility
- To add
  - missing service types - SSAP, SkyNode
  - other ad-hoc astronomy webservices
  - STIL, XPath/XML helpers
- Track developing standards (VOStore, SSO)
- Multi-session ACR for server-side apps
- Refine Workbench applications
  - improve scriptability
- Improve performance
  - myspace especially – underlying service needs to be improved.
- Improve error reporting.
- Get ACR interfaces approved by IVOA in some way