REST and service registries in (the) Virtual Observatory

Matthew J. Graham (CACR/Caltech, NVO)

National Virtual Observatory (NVO)

- US national-level VO project since 2001
- 17 organization collaboration led by Alex Szalay (PI, JHU) and Bob Hanisch (Project Manager, STScI) includes:
  - STScI, CfA, GSFC, IPAC, Caltech, NCSA, NOAO, NRAO
- Transitioning to joint NASA/NSF-funded operational facility – Virtual Astronomical Observatory (VAO)

International Virtual Observatory Alliance (IVOA)

- 17 national members (Brazil joined 2009)
- Meet twice yearly to develop standards to support interoperability (ultimately endorsed by IAU Commission 5)
- Working Groups:
  - Applications
  - Data Access Layer
  - Data Modeling
  - Grid and Web Services
  - Resource Registry
  - Semantics
  - VOEvent
  - VO Query Language
  - VOTable
- Interest Groups:
  - Theory
  - Data Curation & Preservation
  - OGF Astro-RG
  - Data Mining

Web services in the IVOA

- Non-SOAP data access services (HTTP GET to CGI script/Java servlet):
  - SCS, SIAP, SSAP, TAP
- SkyNodes: allow astronomical databases to be queried with ADQL
  - OpenSkyQuery: distributes queries across SkyNodes
  - WESIX: aggregates OpenSkyQuery with Sextractor
- Footprint Services/STOMP: determine coverage and geometry of astronomical data sets
- Spectrum Services: searching and manipulation of spectral data
- CDS: access to CDS data holdings
- VOServices: miscellaneous utilities such as cosmological distances
- Infrastructure:
  - VOSpace
  - Universal Worker Service
  - Registry

VOSpace 1.x

- IVOA interface to distributed storage
- Non-prescriptive about implementation
- Data model:
  - A data object is associated with a node in a VOSpace service
- Functionality defined:
  - create/move/copy/delete/find/list nodes
  - associate arbitrary metadata (properties) with a node
  - links and containers
  - transfer data to/from a VOSpace service
  - expose third-party interfaces (capabilities) to the data

Caveat emptor

- VOSpace 1.x is not the most user friendly
  - SOAP-based
  - WS-Security infrastructure
  - Mix of synchronous/asynchronous operations
  - Hardcore developer-oriented NPNG experience

VOSpace revisioned

- VOSpace 2.0
  - functionally equivalent to VOSpace 1.1
  - RESTful interface:
    - HTTP GET/POST/PUT/DELETE
  - Security:
    - HTTPS with client authentication and valid X.509 certificate
  - Faults:
    - HTTP error codes

Resources and representations

- XML descriptions of VOSpace data model structures that are sent across the wire to interact with VOSpace service:
  - <node>
  - <transfer>
  - <properties>
  - <views>
  - <protocols>
  - <listing>

Example representation: Node

```
<node uri="ivo://nvo.caltech!vospace/myData/m42" xsi:type="vos:StructuredDataNode">
  <properties>
    <property uri="ivo://ivoa.net/vospace/core#mimetype" readonly="true">image/fits</property>
  </properties>
  <views>
    <accepts>
      <view uri="ivo://ivoa.net/vospace/core#view-any"/>
    </accepts>
    <provides>
      <view uri="ivo://ivoa.net/vospace/core#votable"/>
    </provides>
  </views>
  <capabilities>
    <capability uri="ivo://ivoa.net/std/siap">
      <endpoint>http://nvo.caltech.edu/siap</endpoint>
    </capability>
  <capabilities/>
</node>
```

Example: data transfer

- HTTP POST to http://nvo.caltech.edu/vospace/myData/table123/transfers:

```
<vost:transfer>
  <direction>pullFromVoSpace</direction>
  <vost:view uri="ivo://ivoa.net/vospace/core#votable"/>
  <vost:protocol uri="ivo://ivoa.net/vospace/core#http-get"/>
 </vost:transfer>
```

- Service responds with http://nvo.caltech.edu/vospace/myData/table123/transfers/345-fed-454
- HTTP GET to http://nvo.caltech.edu/vospace/myData/table123/transfer/345-fed-454 returns:

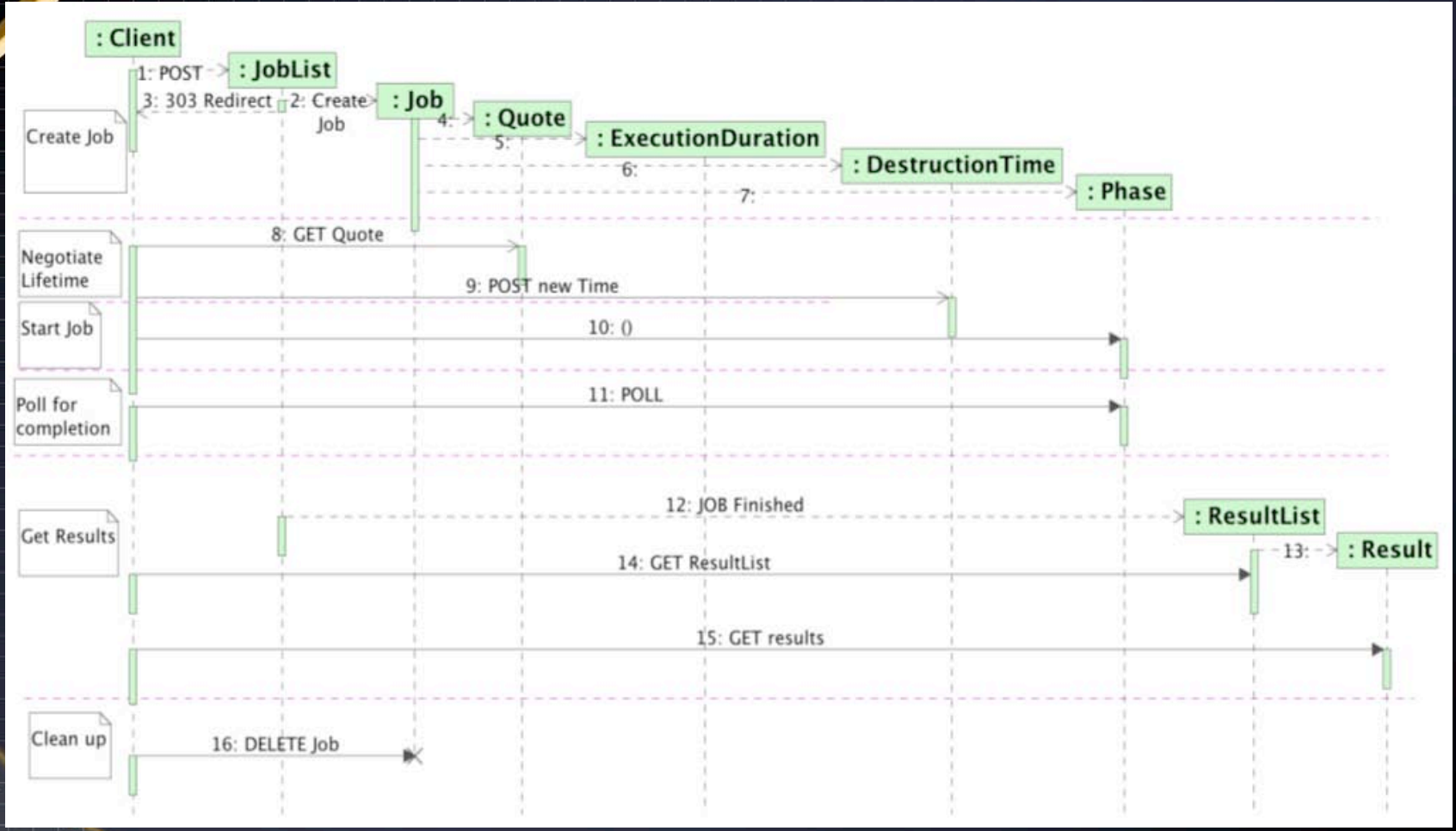
```
<vost:transfer>
  <direction>pullFromVoSpace</direction>
  <vost:view uri="ivo://ivoa.net/vospace/core#votable"/>
  <vost:protocol uri="ivo://ivoa.net/vospace/core#http-get">
    <vost:endpoint>http://nvo.caltech.edu:7777/79f45-3ab0-4de2-bd6c-
     ff016082fd</vost:endpoint>
  </vost:protocol>
</vost:transfer>
```

Universal Worker Service (UWS)

- Pattern for asynchronous services
- Resources:

URI	Description
/{jobs}	the Job List
/{jobs}/(job-id)	a Job
/{jobs}/(job-id)/phase	the Phase of (job-id)
/{jobs}/(job-id)/executionduration	the maximum execution duration of (job-id)
/{jobs}/(job-id)/destruction	the destruction instant for (job-id)
/{jobs}/(job-id)/error	any error message associated with (job-id)
/{jobs}/(job-id)/quote	the Quote for (job-id)
/{jobs}/(job-id)/results	the Results List for (job-id)

UWS example


: Client
: JobList
: Job
: Quote
: ExecutionDuration
: DestructionTime
: Phase
: ResultList
: Result
1: POST
2: Create Job
3: 303 Redirect
4:
5:
6:
7:
Create Job
8: GET Quote
Negotiate Lifetime
9: POST new Time
Start Job
10: ()
Poll for completion
11: POLL
12: JOB Finished
Get Results
13:
14: GET ResultList
15: GET results
Clean up
16: DELETE Job

Best practices

- Web Services Interoperability (WS-I) Organisation was formed to provide a set of guidelines (called profiles) on how SOAP-based services should be interoperable
- VO Web Service Basic Profile specifies section of WS-I Basic Profile that are relevant for the VOA
- Rules to take into account when implementing a web service and how to check conformance
- No official equivalent for RESTful services but “REST in the VO” (currently working draft) defines minimal guidelines for VO services

Resources and registries

- A *resource* is a VO element that can be described in terms of who curates it or maintains it and which can be given a name and a unique identifier
- The resource metadata that describes a resource is encoded in an XML schema known as *VOResource*
- A *registry* is a repository of resource descriptions and is employed by users and applications to discover resources

Registry Framework

Full Searchable Registry

VO Projects

Local Publishing Registry

Full Searchable Registry

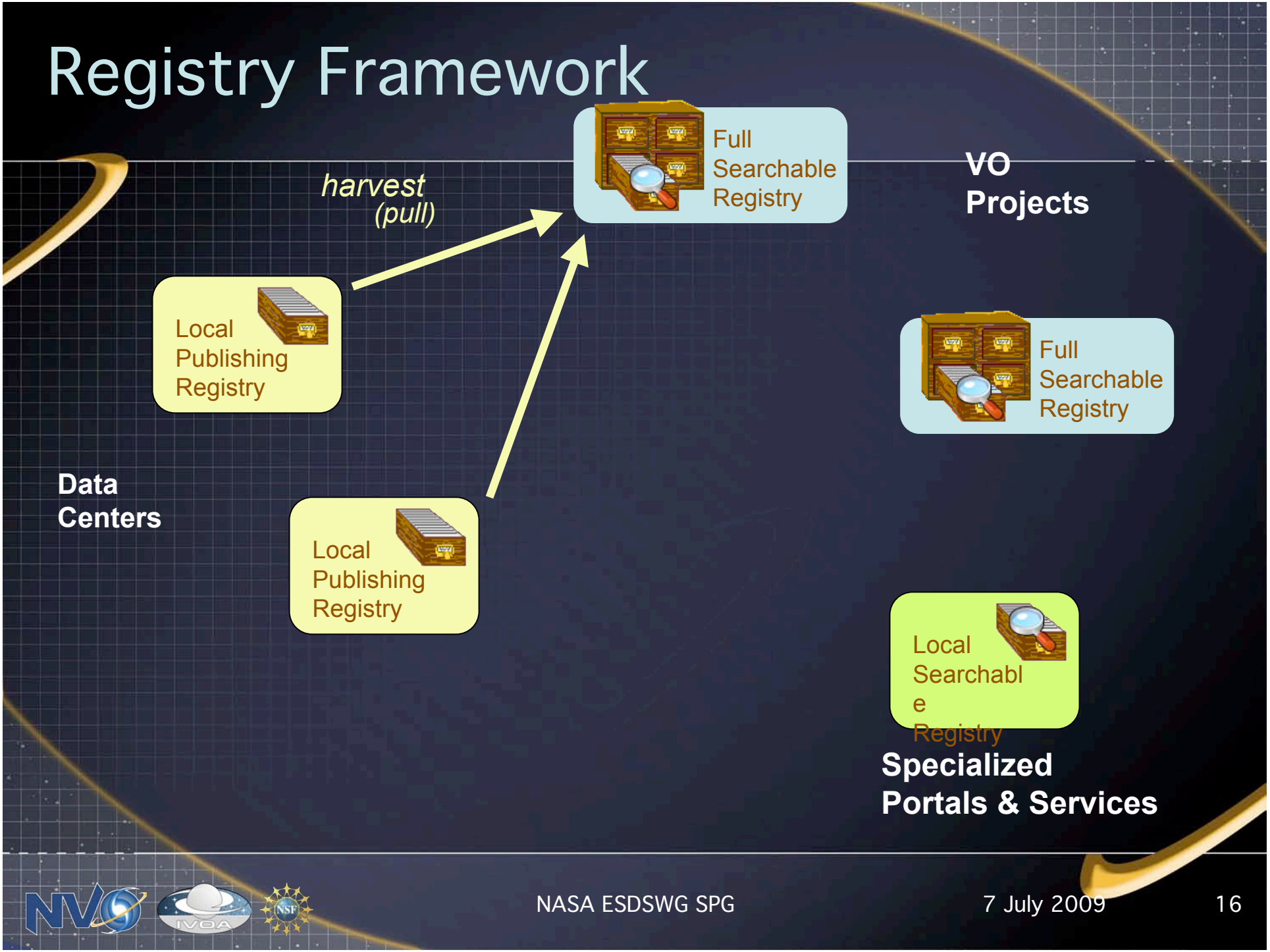
Data Centers

Local Publishing Registry

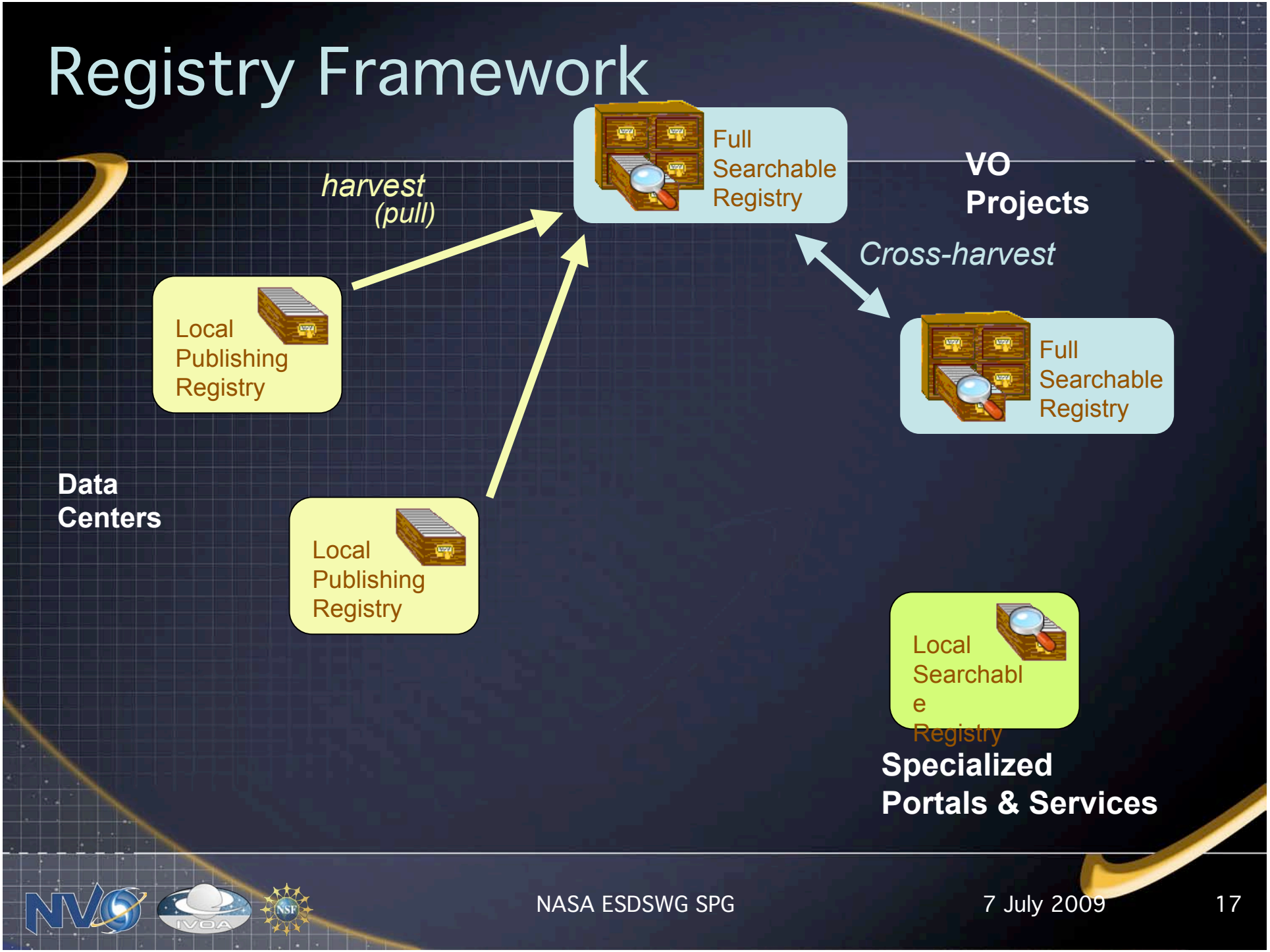
Local Searchable Registry

Specialized Portals & Services

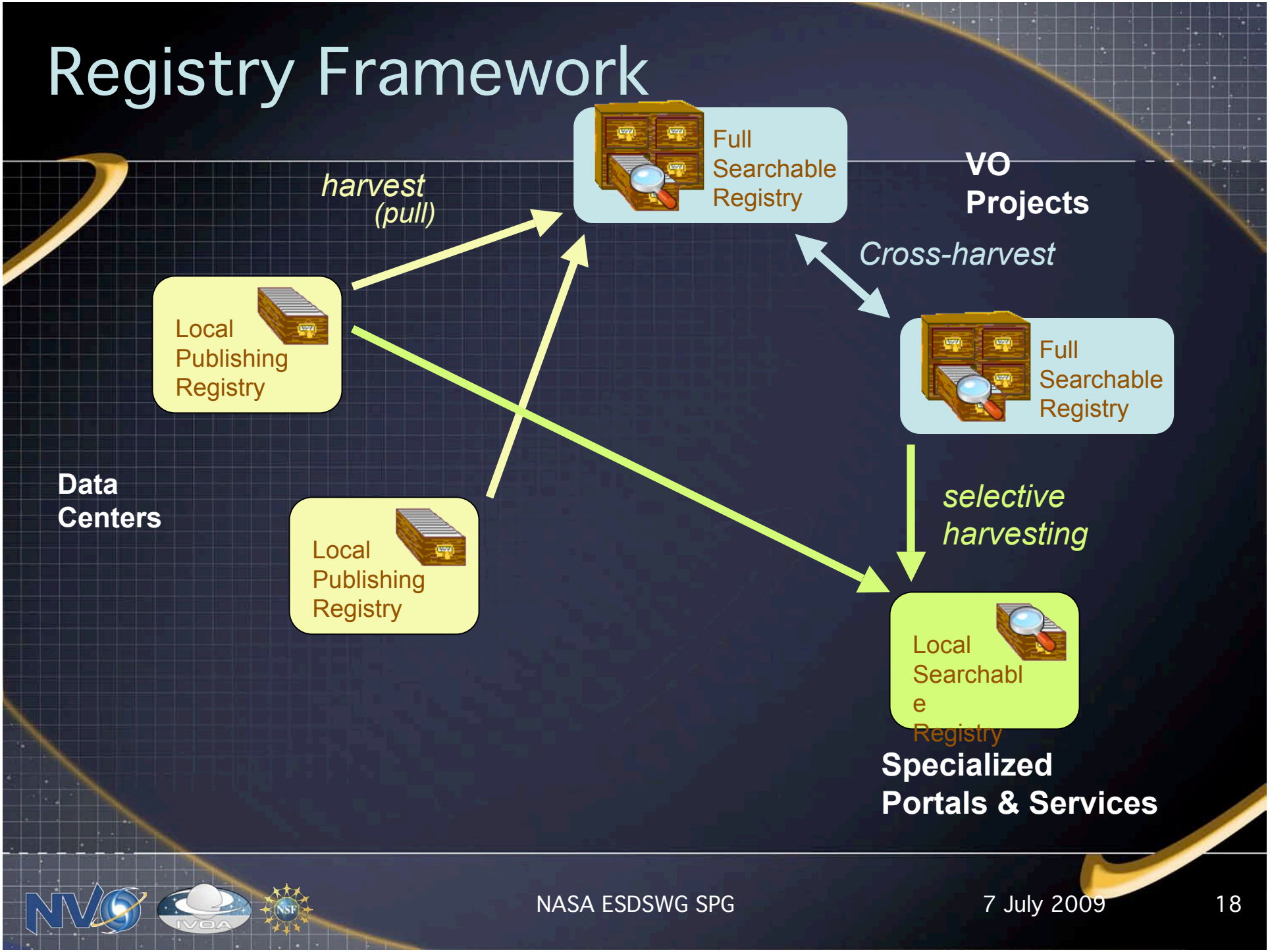
Registry Framework



Registry Framework



Registry Framework



Registry Framework

Full Searchable Registry

VO Projects

Local Publishing Registry

Full Searchable Registry

Data Centers

Local Publishing Registry

search queries

Client Applications

Local Searchable Registry

Specialized Portals & Services

Registry Framework

Full Searchable Registry

VO Projects

Local Publishing Registry

Full Searchable Registry

Data Centers

Local Publishing Registry

search queries

Local Searchable Registry

Client Applications

Specialized Portals & Services


NVO
IVOA
NSF

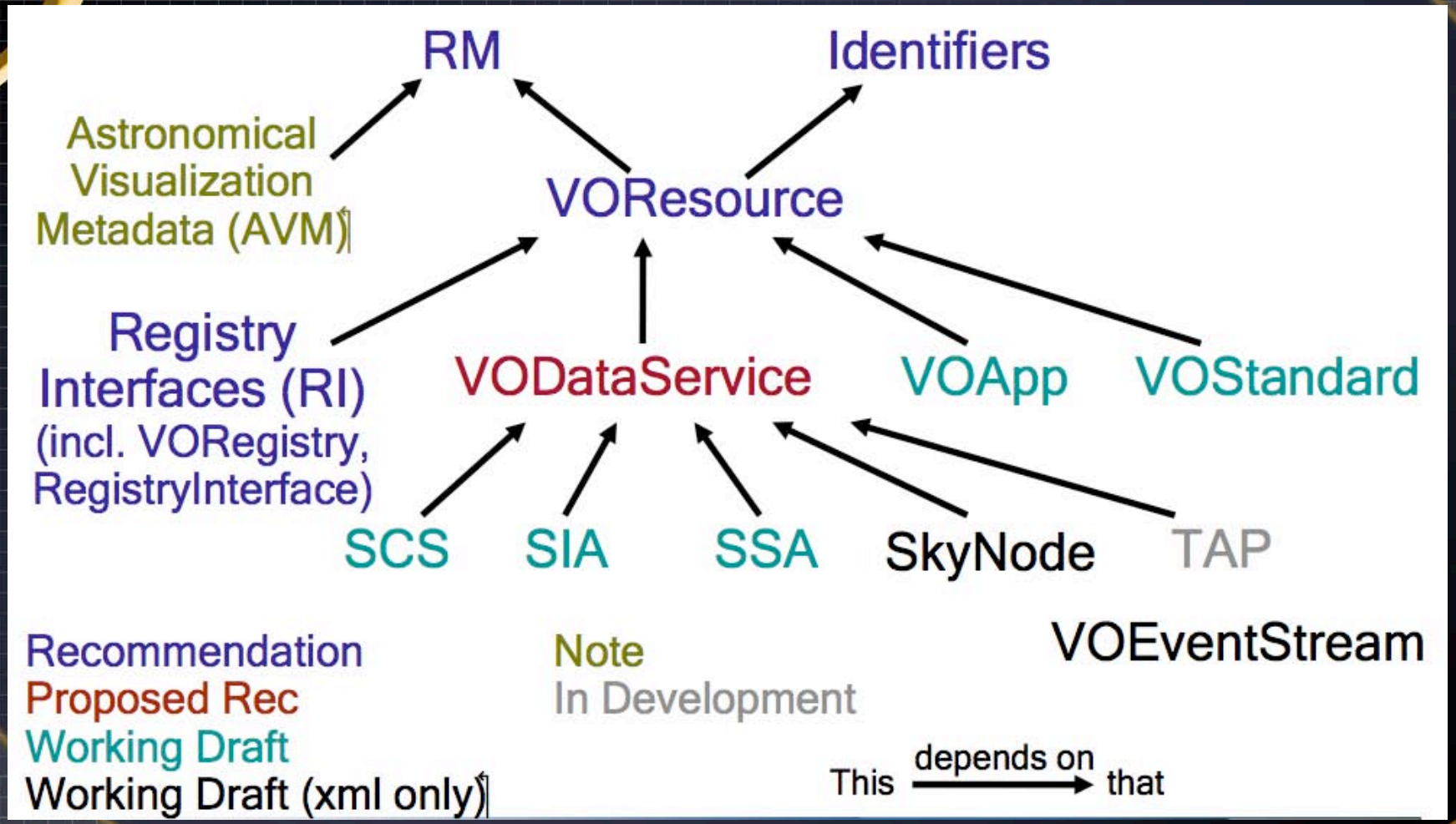
VOResource example: Organisation

```
<resource xsi:type="Organisation" xmlns:vr="http://www.ivoa.net/xml/VOResource/v1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <title>NCSA Radio Astronomy Imaging</title>
    <shortName>NCSA-RAI</shortName>
    <identifier>ivo://rai.ncsa/RAI</identifier>
    <curation>
      <publisher ivo-id="ivo://ncsa.uiuc/NCSA"> National Center for Supercomputing Applications </publisher>
      <creator>
        <name> Dr. Richard Crutcher </name>
        <logo> http://rai.ncsa.uiuc.edu/rai.jpg </logo>
      </creator>
      <date>1993-01-01</date>
      <contact>
        <name>Dr. Raymond Plante</name>
        <email>rplante@ncsa.uiuc.edu</email>
      </contact>
    </curation>
    <content>
      <subject>radio astronomy</subject>
      <subject>data repositories</subject>
      <description> The Radio Astronomy Imaging Group at the National Center ... </description>
      <referenceURL>http://rai.ncsa.uiuc.edu/</referenceURL>
      <type>Organisation</type>
      <contentLevel>Research</contentLevel>
    </content>
    <facility>Berkeley-Illinois-Maryland Array (BIMA)</facility>
</resource>
```

What registries are out there?

- IVOA Registry of Registries
- STScI Searchable Registry
- NCSA Publishing Registry
- Caltech NVO Registry
- HEASARC Publishing Registry
- WFAU Publishing Registry
- Leicester Publishing Registry
- CASU Publishing Registry
- MSSL Publishing Registry
- VizieR Publishing Registry
- VOParis Registry
- GAVO Data Center Registry
- ESAVO Registry Resource
- Sternberg Astronomical Institute Catalog Access Services Publishing Registry
- JVO Publishing Registry

Extension schemas



Registry interfaces/harvesting

- Registries have two SOAP-based interfaces
  - Search interface:
    - Search (uses ADQL)
    - KeywordSearch
    - GetResource
    - GetIdentity
  - Harvesting interface (OAI-PMH standard):
    - Identify
    - ListIdentifiers
    - ListRecords
    - GetRecord
    - ListMetadataFormats
    - ListSets

Summary

- SOAP suffers from a number of issues:
  - RPC mindset
  - interoperability issues
  - code binding in toolkits
  - WS-*
- REST is simpler but:
  - Architectural style not an infrastructure
  - Manipulating information not invoking behaviour
  - Uniform interface semantics
  - Verbs in URLs
  - No standard formal description?
  - Fits well with Semantic Web ideas