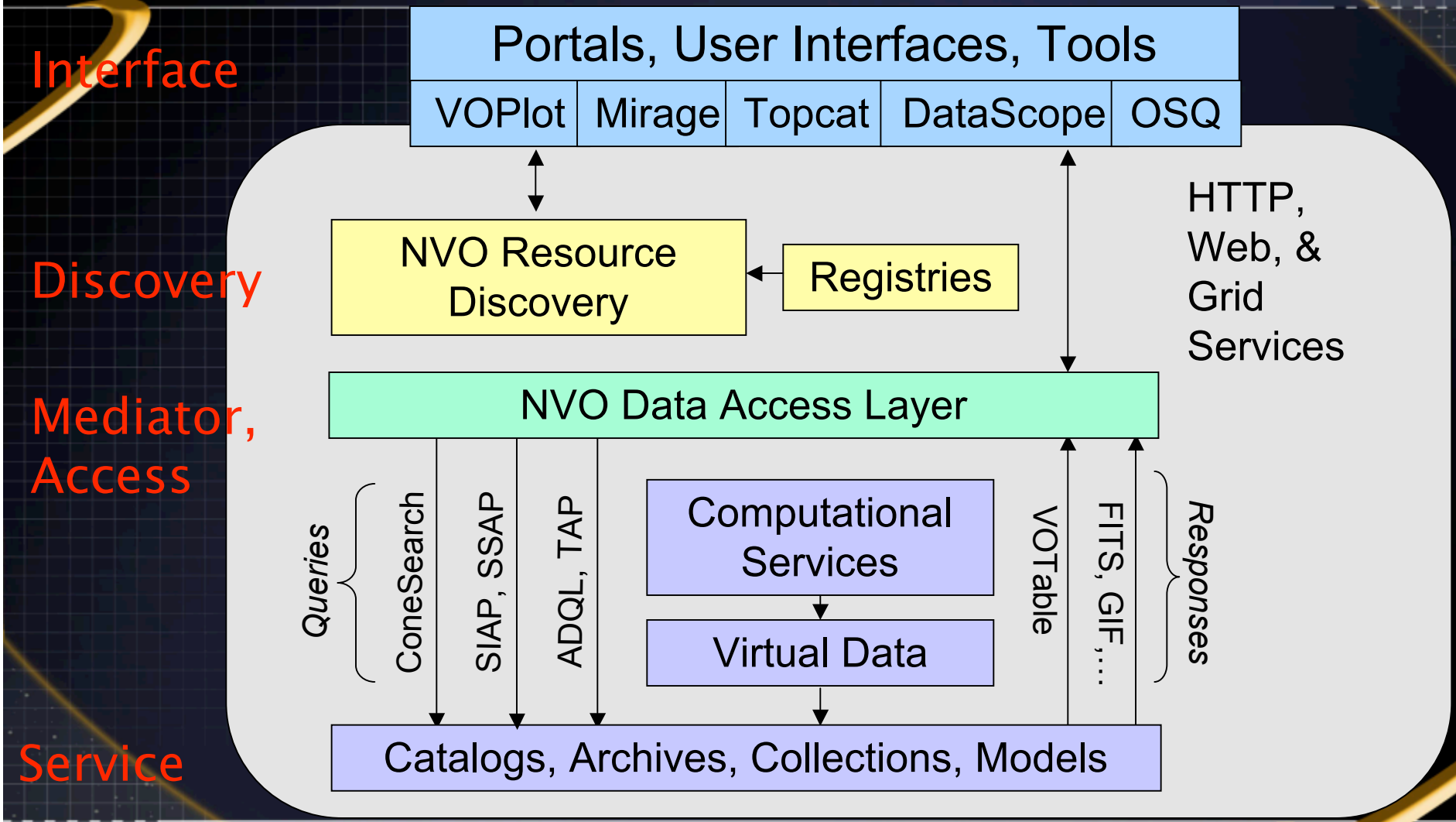
The Information Technology Challenges of the Virtual Observatory

Robert Hanisch
Space Telescope Science Institute
Baltimore, Maryland, USA
Project Manager, US National Virtual Observatory

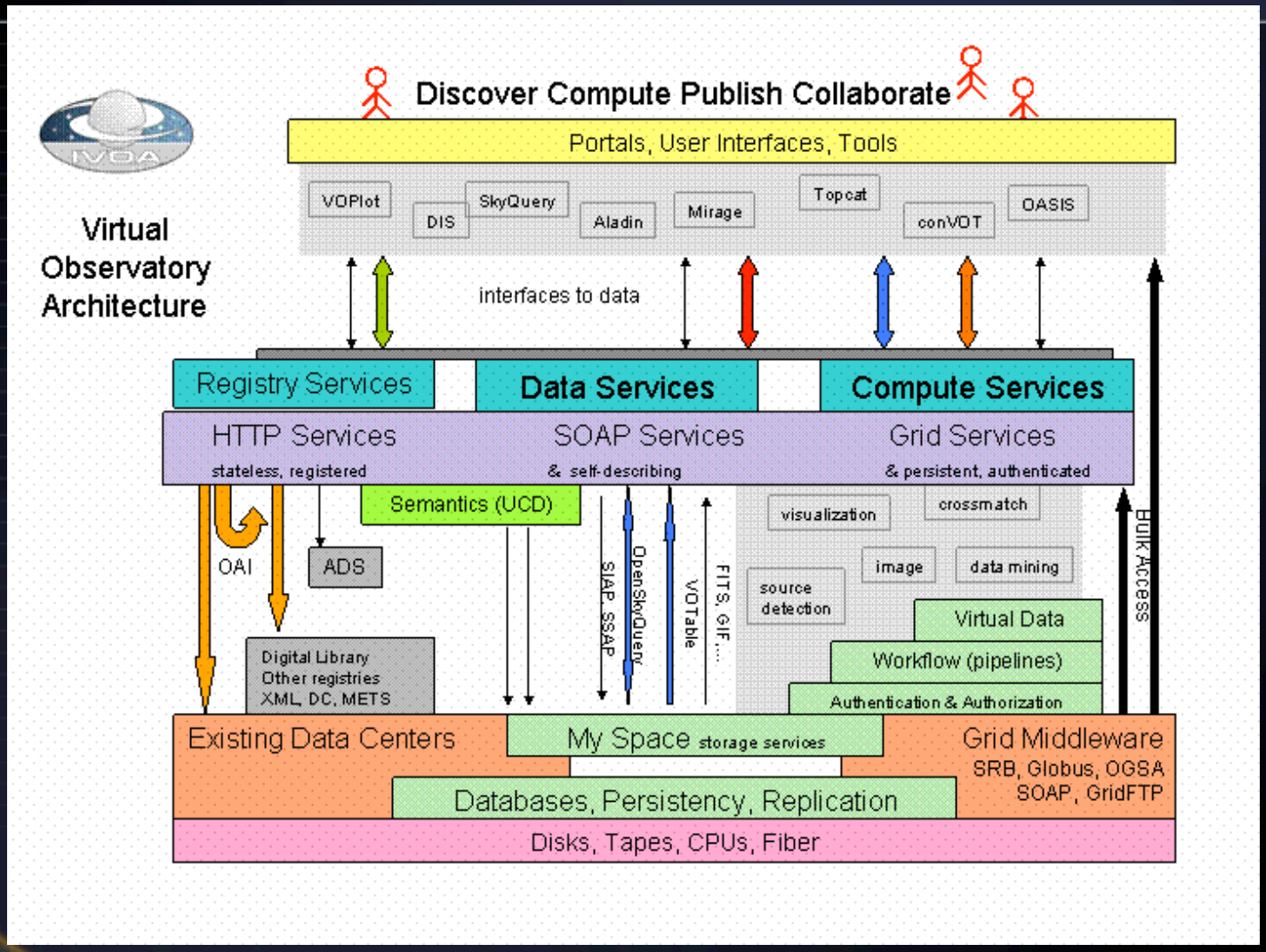
NVO origins in IT

- US NVO project originally funded by NSF Information Technology Research program
- NSF interest: show relevance of cyberinfrastructure to discipline-based research
- NVO focused on re-use, adoption of existing technologies, adaptation where necessary
- NVO approach has proven similar to that taken by other distributed data system projects
- Exploits and adapts from Grid infrastructure

VO architecture, simple view



VO architecture, detailed view



Interoperability challenges

- Metadata standards
- Data discovery
- Data requests
- Data delivery
- Units
- Database queries
- Distributed applications; web services
- Authentication and authorization

Solutions rely on *standards* and *interfaces*

Development areas

- Resource registries
- Data models
- Content description (UCDs)
- Data access layer (SIAP, SSAP, cone search)
- VO Query Language (ADQL, OpenSkyQuery)
- VOTable
- Grid and Web Services, workflows

Activities can be followed on IVOA web (http://ivoa.net)

Data Discovery

- Resource Metadata
  - Descriptions of data collections and the organizations responsible for them, data delivery services, computational services, software, etc.
  - Based on Dublin Core with astronomy-specific extensions
  - Represented as XML schema; extensible
  - Contents stored in Resource Registries that exchange metadata records through the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH)
- Identifiers
  - Rules for constructing URIs for IVOA resources
  - Adopted for data set linking by ADS, ADEC, AAS journals

The role of Resource Registries

- Used to discover and locate *resources*—data and services—that can be used in a VO application
- Resource: anything that is describable and identifiable.
  - Besides data and services: organizations, projects, software, …
  - Presently concerned with simple set of resource types
- Registry: a list of resource descriptions
  - Expressed as structured metadata
    to enable automated processing and searching

Registry Framework

Full Searchable Registry

VO Projects

Local Publishing Registry

Full Searchable Registry

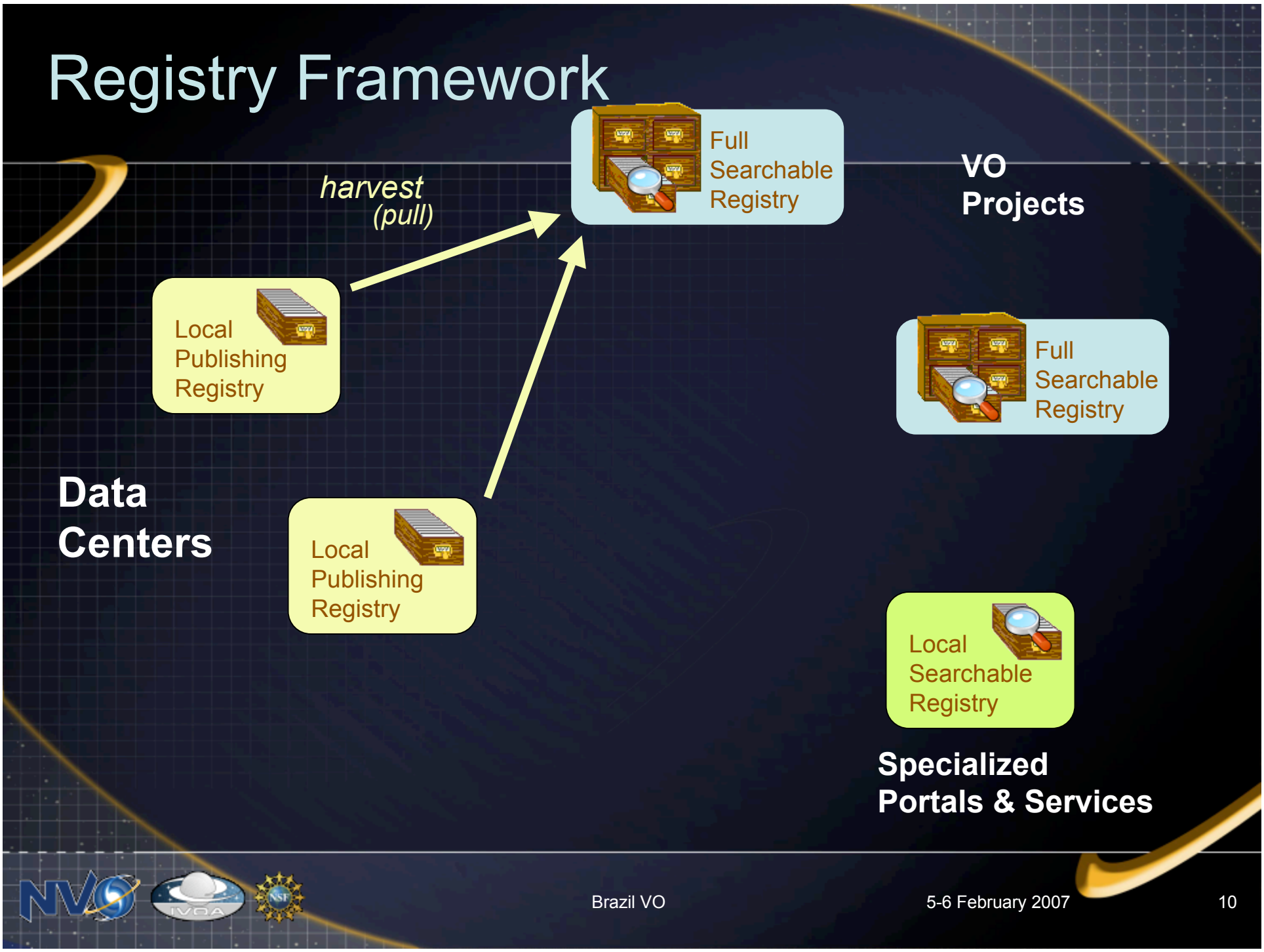
Data Centers

Local Publishing Registry

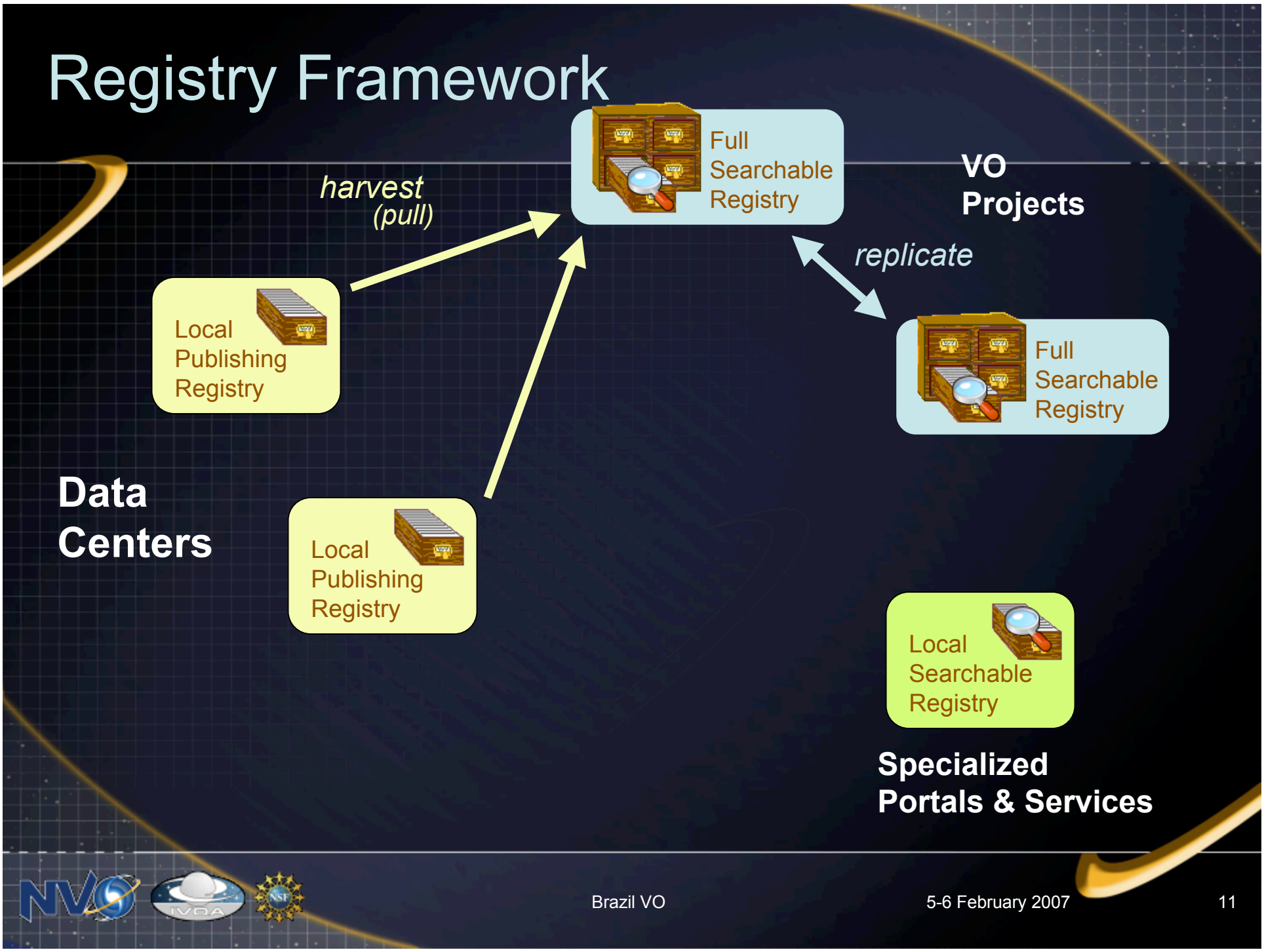
Local Searchable Registry

Specialized Portals & Services

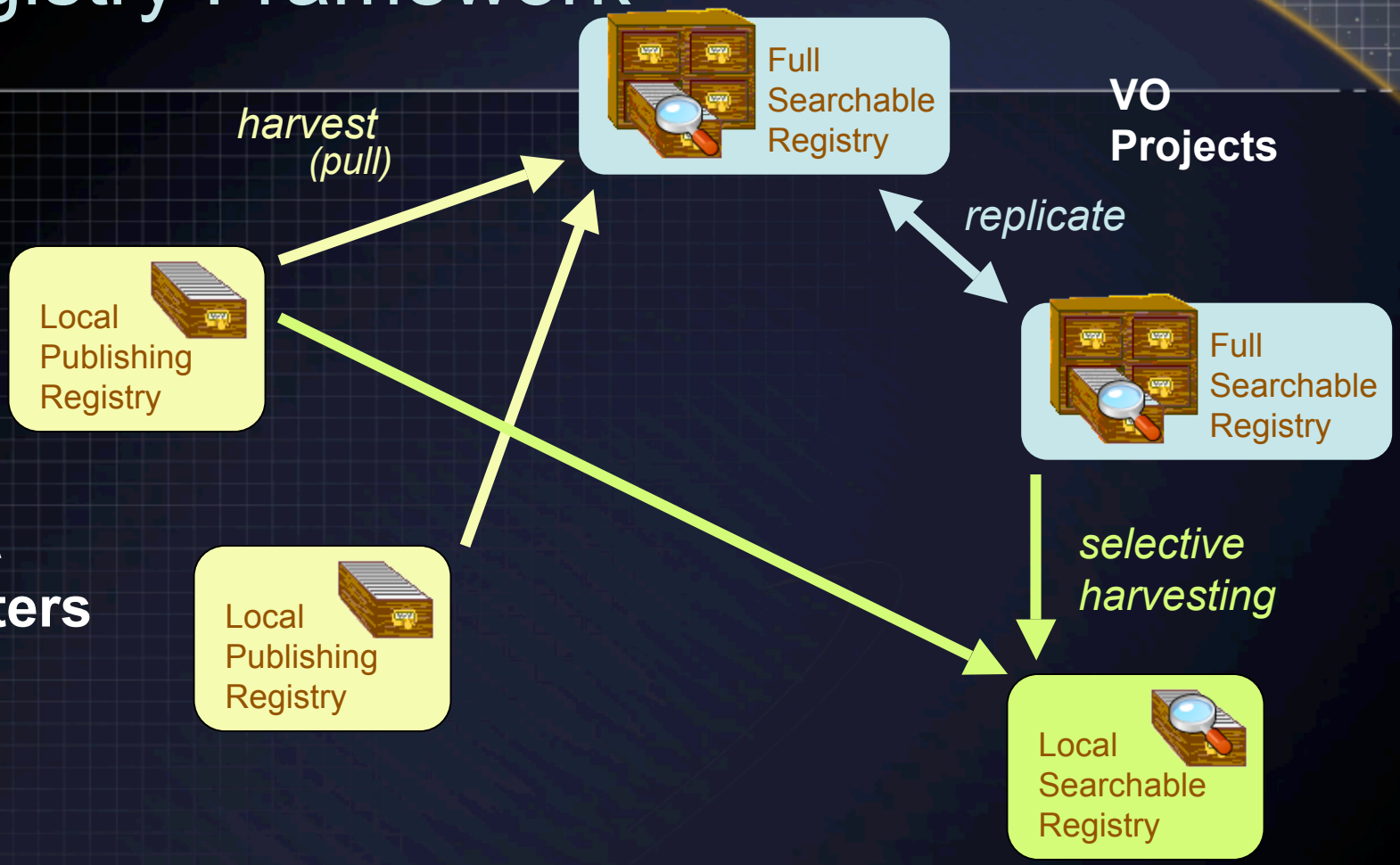
Registry Framework



Registry Framework



Registry Framework



Registry Framework

VO Projects

Data Centers

search queries

Client Applications

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

Resource Metadata: XML Schema

- Classes of Resources

  Organisation, DataCollection, Service, Registry

  – Specific classes inherit from generic <Resource>

- Organized into separate schemas:
  - Core resource metadata: VOResource
  - Various extensions schemas containing specific types
- Capable of describing…
  - Data centers, research organizations, missions, observatories
  - Data collections, archives
  - VO standard services: Cone Search, Simple Image Access
  - Existing Browser/CGI-based services
  - Web Services

Describing Resources with XML: VOResource

- Model: types of Resources
  - Generic Resource
  - Extensions: e.g. DataCollection, Service, ConeSearch, …
- VOResource: Family of XML schemas
  - Core schema: VOResource
    - Common set of metadata applicable to all resources
      including Dublin Core
    - Resource types: Resource, Service, Organisation
  - Extension schemas to describe specific kinds of resources
    - Extended type *inherits* generic metadata
    - adds metadata specific to the type of resource
  - Extensibility allows for evolution
    - Developers only need to support types of interest to them
    - Allows developers to experiment with non-standard extensions
  - Currently transitioning from v0.9 to v0.10
- Lastest status of metadata standards:
  http://www.ivoa.net/twiki/bin/view/IVOA/ResourceMetadata

Data Models

- Data models describe the nature of data sets in generic terms
- Data models are necessary for mapping or transforming data from diverse sources into a common representation
- FITS format is syntactic, not semantic; FITS + data models provides means to interpret data across organizations and software systems
- Formal UML representations

Data Models

- Component models
  - Characterization
  - Space-Time Coordinates
- Dataset models
  - Spectrum
  - Spectral lines (atomic, molecular)
  - Catalog
  - Image

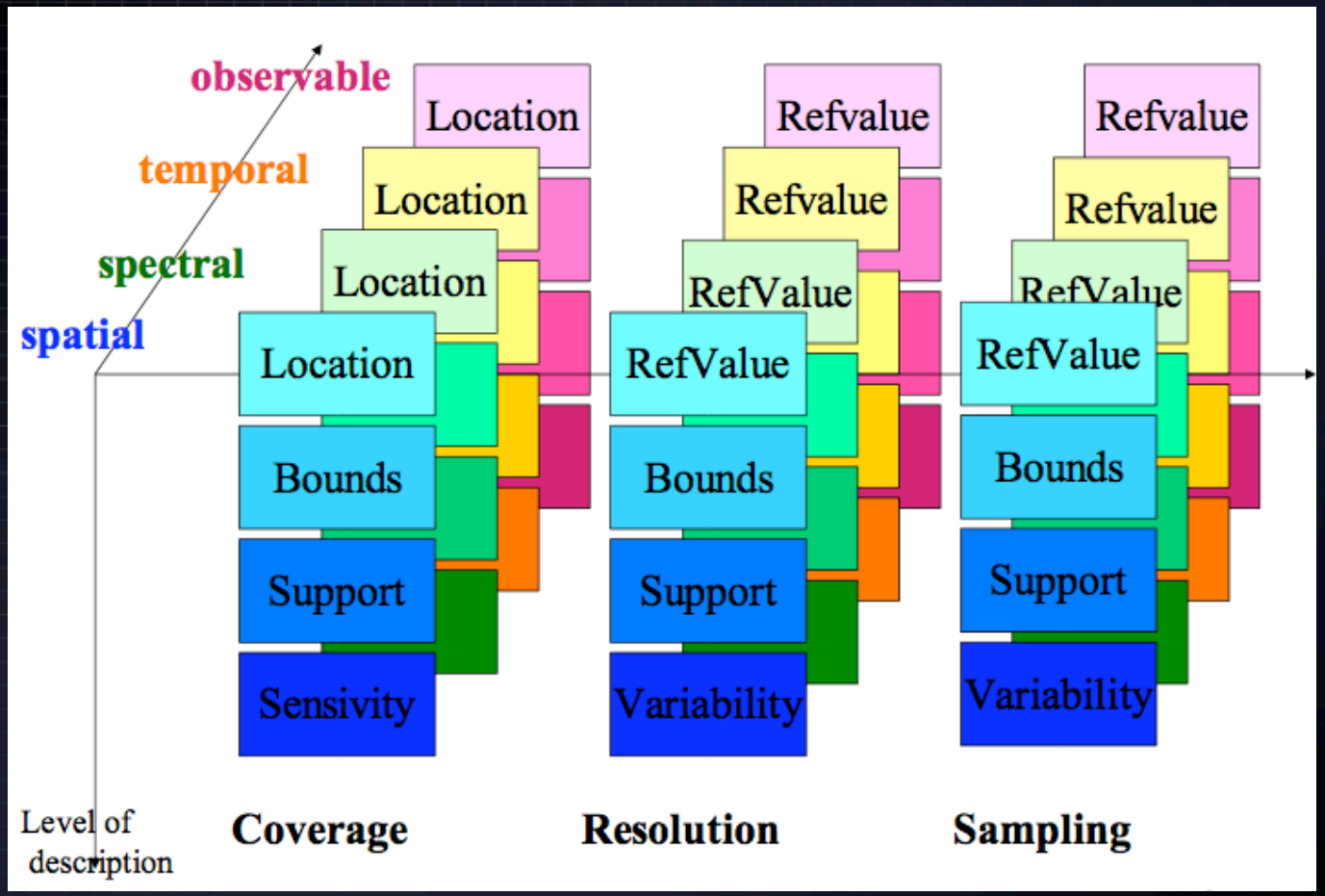
Spectral Data Model

- Spectral model describes spectrum with metadata for aperture, observation time, resolution, etc. - single description for all wavebands
- Needed for both discovery (give me all UV high resolution spectra of...) and analysis (is this feature instrumental or astrophysical broadening)
- Data description is at analysis level (resolution is abstracted, not separated into mirror, detector, atmospheric pieces)

Characterization

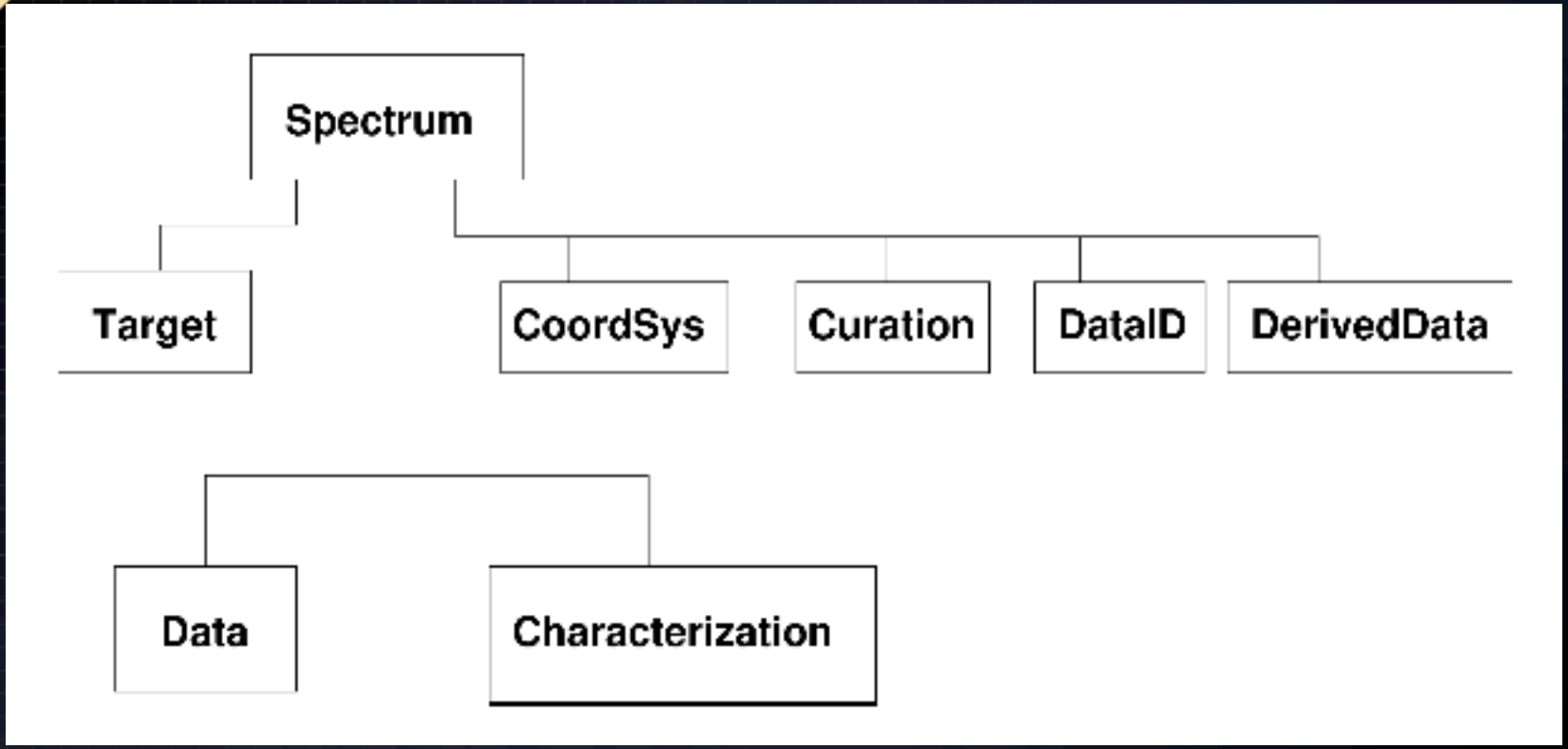
- Characterization model describes context of observation in space, time, spectral and (if needed) other domains.
- Standard way to describe any data. Where were we looking, when, etc. and at a variety of standard levels of description from coarse (e.g. approximate pointing direction) to fine (e.g. STC region on sky)

Data model overview



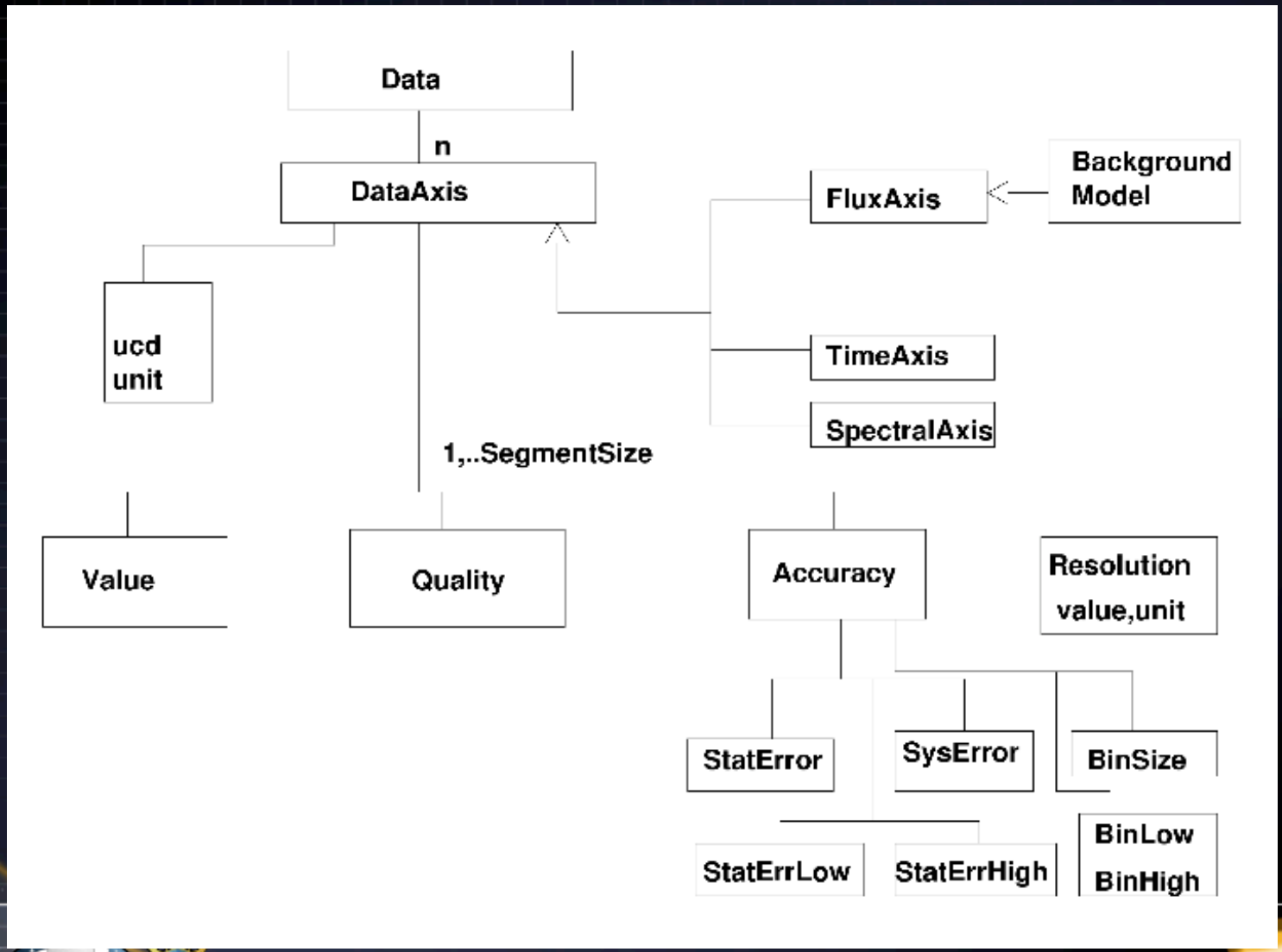
(M. Louys, CDS)

Spectrum data model

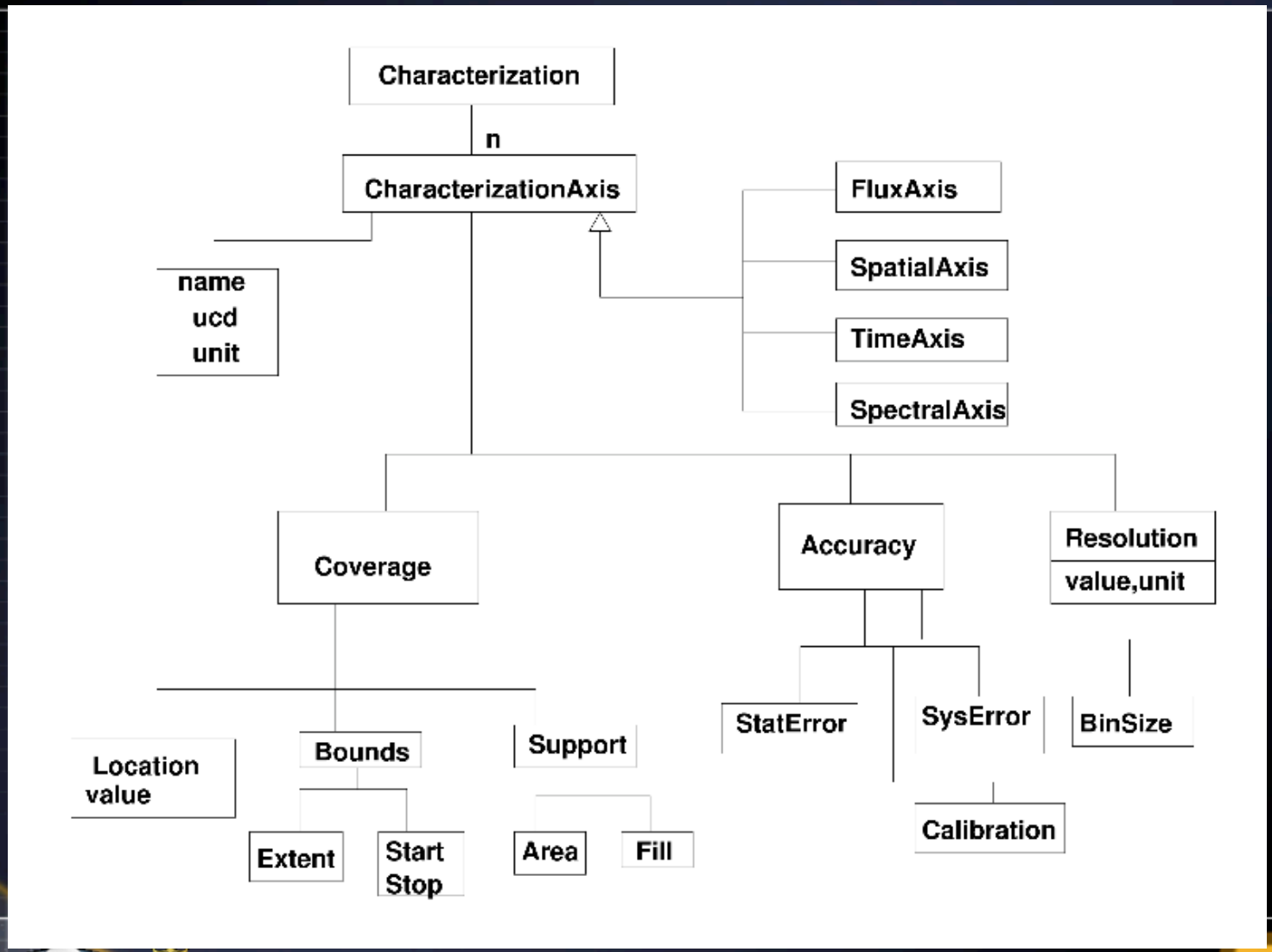


(J. McDowell, CfA)

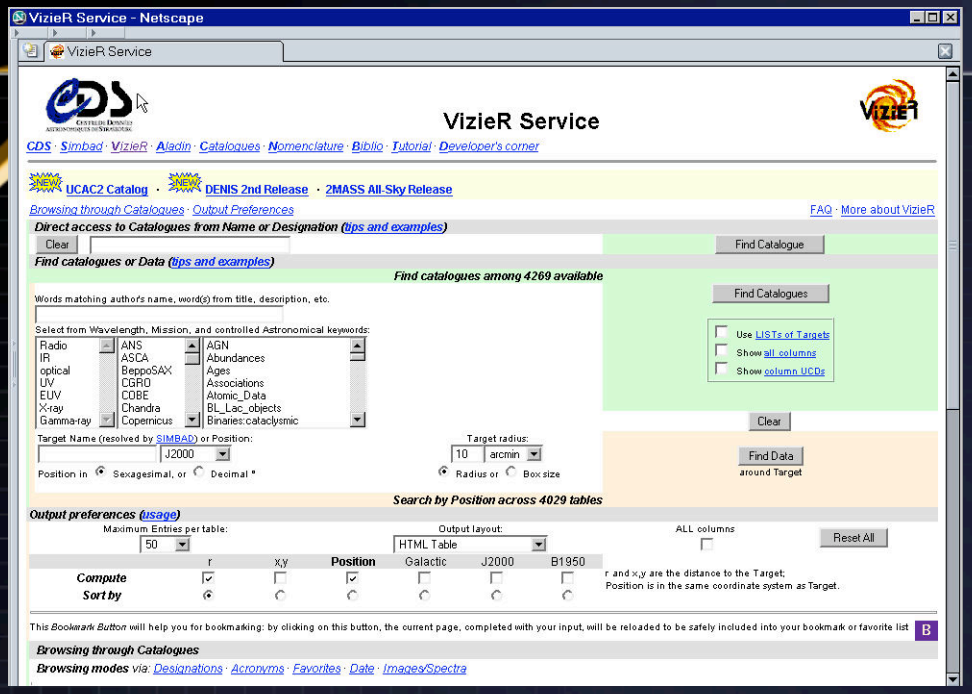
Spectrum data model



Spectrum data model



What's in a name?



VizieR: Contains more than 4000 astronomical catalogues consisting of one or several tables.

Frequency: column name	Frequency: unit
956 Vmag	1263 mag
62 V	4 ---
21 V0	1 10-17W/m2/nm
11 Vmax	
8 Vmag2	
7 <Vmag>	
6 mag	
4 <V>	
4 Vmagph	
4 A0	
4 Vcs	
4 Vmagav	
3 V91	
3 V(HB)	
3 VMAGp	
3 Vpred	
3 Vmin	
3 Vmag1	
3 V2mag	
2 V42	

Result was truncated to 20 lines (out of 144).
Reload with 10 lines.

Problem: as the catalogues come from many different sources, the original descriptions are very heterogeneous: "Give me all tables containing the V magnitude in the Johnson system."
144 different names for Johnson V.

Unified Content Descriptors

- A generic syntax and agreed-upon vocabulary for astronomical quantities; an ontology for astronomical measurements
- Derived from maintenance of thousands of astronomical catalogs, where many names used to represent the same quantities
- Hierarchical structure with top level categories such as *arith, em, instr, obs, phot, phys, pos, spect, stat, time*
- For example:

  pos.eq.ra

  pos.eq.dec

  phot.mag;em.opt.V
- Used to augment metadata descriptions, column headings, etc.

Data Formats

- Astronomy's Flexible Image Transport System – FITS – retained
  - 25-year heritage
  - Worldwide adoption for both archival and run-time applications
  - IAU endorsement
  - Syntax, limited semantics
- VOTable
  - XML-based standard for tabular data
  - Standard schema
  - Java, C++, C#, and Perl software libraries
  - Complements FITS
  - Incorporates semantics

Sample VOTable

```
<?xml version="1.0"?>
<VOTABLE version="1.1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="http://www.ivoa.net/xml/VOTable/VOTable/v1.1">
  <COOSYS ID="J2000" equinox="J2000." epoch="J2000." system="eq_FK5"/>
  <RESOURCE name="myFavouriteGalaxies">
    <TABLE name="results">
      <DESCRIPTION>Velocities and Distance estimations</DESCRIPTION>
      <PARAM name="Telescope" datatype="float" ucd="phys.size;instr.tel"
        unit="m" value="3.6"/>
      <FIELD name="RA" ucd="pos.eq.ra;meta.main" ref="J2000"
        datatype="float" width="6" precision="2" unit="deg"/>
      <FIELD name="Dec" ucd="pos.eq.dec;meta.main" ref="J2000"
        datatype="float" width="6" precision="2" unit="deg"/>
      <DATA>
        <TABLEDATA>
          <TR><TD>010.68</TD><TD>+41.27</TD></TR>
          <TR><TD>287.43</TD><TD>-63.85</TD></TR>
        </TABLEDATA>
      </DATA>
    </TABLE>
  </RESOURCE>
</VOTABLE>
```

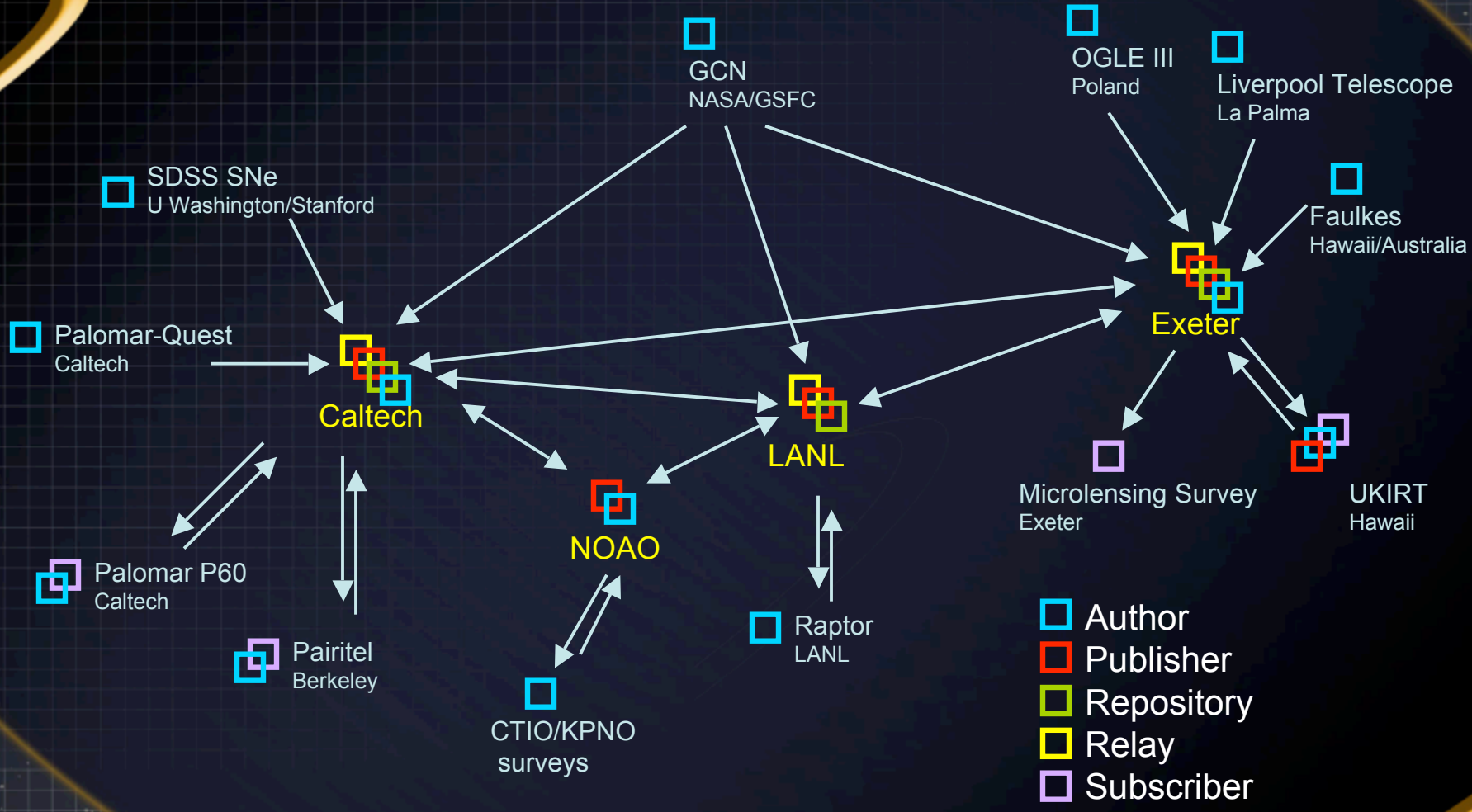
Data Formats

- Space-Time Coordinates
  - Standard representations of locations of astronomical objects in space, wavelength (energy), and time
  - Represented as XML schema
- VOEvent
  - Standard representation of transient event (gamma ray burst, supernova, flaring star, discovery of solar system object, etc.)
  - Represented as XML schema

VOEvent structure

- *Who*: Publisher, Contact, etc.
- *WhereWhen* (== STC): RA, Dec, UTC
- *What*: Hierarchy of named parameters, Units, Semantic type (UCD), References, Descriptions
- *How*: How was the evidence gathered: camera, telescope, etc
- *Why*: Probability list of interpretation: supernova, comet, asteroid, ...
- *Citation*: Link to other VOEvent: Follow-up, Supercede, Retraction
- *Reference*: Supporting material, e.g. lightcurve, image

VOEventNet


GCN
NASA/GSFC
OGLE III
Poland
Liverpool Telescope
La Palma
SDSS SNe
U Washington/Stanford
Faulkes
Hawaii/Australia
Palomar-Quest
Caltech
Caltech
Exeter
LANL
NOAO
Microlensing Survey
Exeter
UKIRT
Hawaii
Palomar P60
Caltech
Pairitel
Berkeley
Raptor
LANL
CTIO/KPNO
surveys
Author
Publisher
Repository
Relay
Subscriber

Data Access Layer

- Cone Search
  - Simplest possible astronomical query: return a list of objects or observations within a certain radius of a given position on the sky
  - Response is encoded as VOTable
  - `http://casjobs.sdss.org/vo/dr4cone/sdssConeSearch.asmx/ConeSearch?&RA=180&DEC=-1&SR=0.1`
- Simple Image Access Protocol (SIAP)
  - Extends Cone Search to allow specification of image size
  - Response includes metadata about images, encoded as VOTable
  - Images are referenced by URL
  - `http://skyview.gsfc.nasa.gov/cgi-bin/vo/sia.pl?&POS=180,-1&SIZE=0.1`
- Simple Spectrum Access Protocol (SSAP)
  - Astronomical spectra have more subtleties and variations in representation than images → access protocol is more complicated
  - Query supports more qualifiers and response adds more metadata, again encoded as VOTable
  - Spectra referenced by URL or encoded in-line in the VOTable
  - Latter versions will include support for spectral energy distributions and time series

Data Access Layer

- SIAP enhancements
  - Solar data: AstroGrid has private extension in which time range is the primary search key
  - Planetary data: NVO working with NASA Planetary Data System to incorporate search by name, instrument, mission, etc.
  - SIAP V2.0 to build on flexibility of SSAP, and will support solar and planetary data

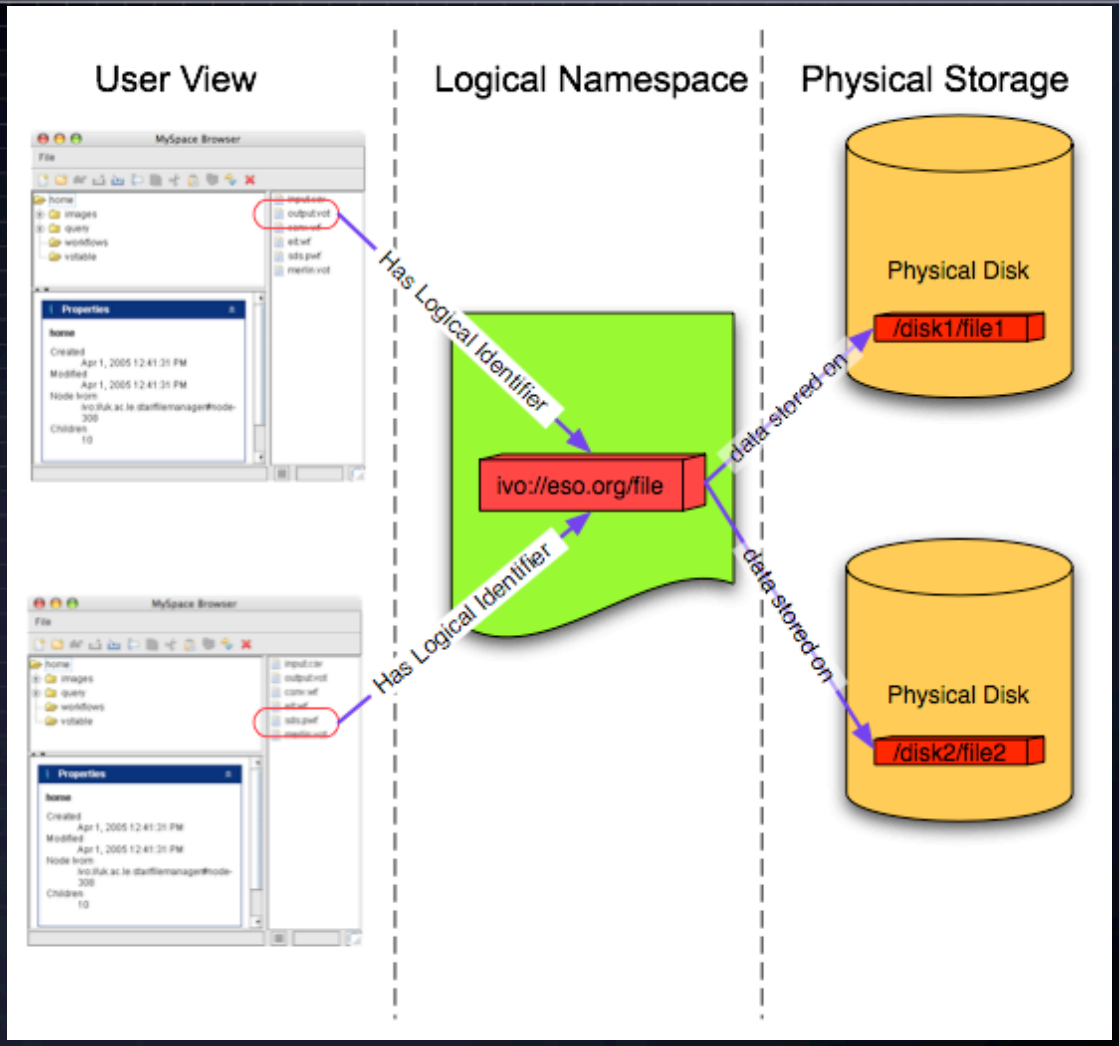
Database Queries

- Astronomical Data Query Language (ADQL)
  - Standard grammar for database queries
  - Core SQL functions plus astronomy-specific extensions
  - String ~~and XML~~ representations
- SkyNode
  - Standard interface wrapper for relational databases
  - Accepts ADQL query
  - “Full” SkyNodes support positional cross-match function
  - OpenSkyQuery portal provides users with interface for understanding database structure and contents and for constructing queries
- Table Access Protocol (TAP)
  - Catalog and database access via ADQL, but without cross-match

Web Services

- Goal is for all IVOA services to to support basic REST (HTTP GET) requests and web services interfaces (WSDL, SOAP)
- Standards
  - Basic service profile (registration, aliveness test, runID, usage logs)
  - Security (single sign-on, authentication and authorization)
  - Asynchronous services
  - Distributed data storage (VOSpace)
  - Workflow
- Developments are complementary to and coordinated with Global Grid Forum through the GGF Astronomy Research Group

VOSpace architecture



VOSpace data nodes

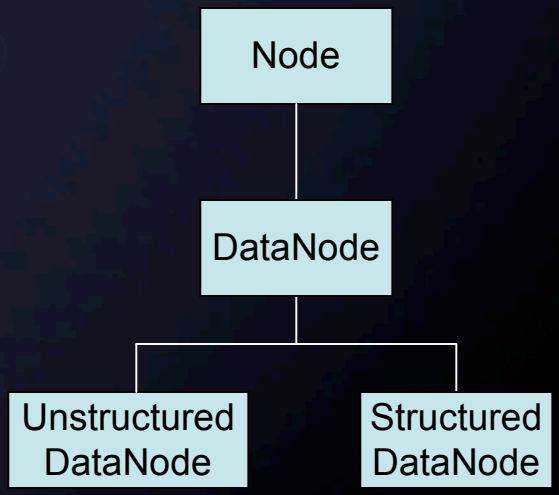
- Each node contains a map of key:value properties:

```
<node uri="vos://nvo.caltech!vospace/mydata1">
    <properties>
        <property uri="ivo://net.ivoa.vospace/properties/create.date">
        2006-09-11T13:35:51Z
        </property>
    </properties>
</node>
```

readonly="true"

- There are currently four types of node:

```
<node/>
<node xsi:type="vos:DataNode"/>
<node xsi:type="vos:UnstructuredDataNode"/>
<node xsi:type="vos:StructuredDataNode"/>
```



Recap

- Everything here should now be familiar!