Software Development for Virtual Observatories

BRAVO Workshop

February 2007

Rafael Santos

Warning!

- This presentation is *biased*.
- I'll talk about
  - VO software development, including some under-the-hood examples.
  - Some ideas we're playing with.
  - Some considerations for developers who would like to play with VO software development.
- I'll ***not*** talk about
  - VO for astronomers/astrophysics!
  - Some infrastructure issues (clusters, grid, databases, networks, operating systems, etc).

Introduction

- What are Virtual Observatories (VOs)?
- What can we do with Virtual Observatories?
- Relevant issues for software developers:
  - What should I know before starting software development for the VO?
  - Which tools can we use to access VO data?
  - How can we develop new VO tools?
  - **Development:** Which are the real VO development needs?
  - **Research:** Which are the challenges posed by VOs development?

What are VOs?

- From http://www.euro-vo.org/:

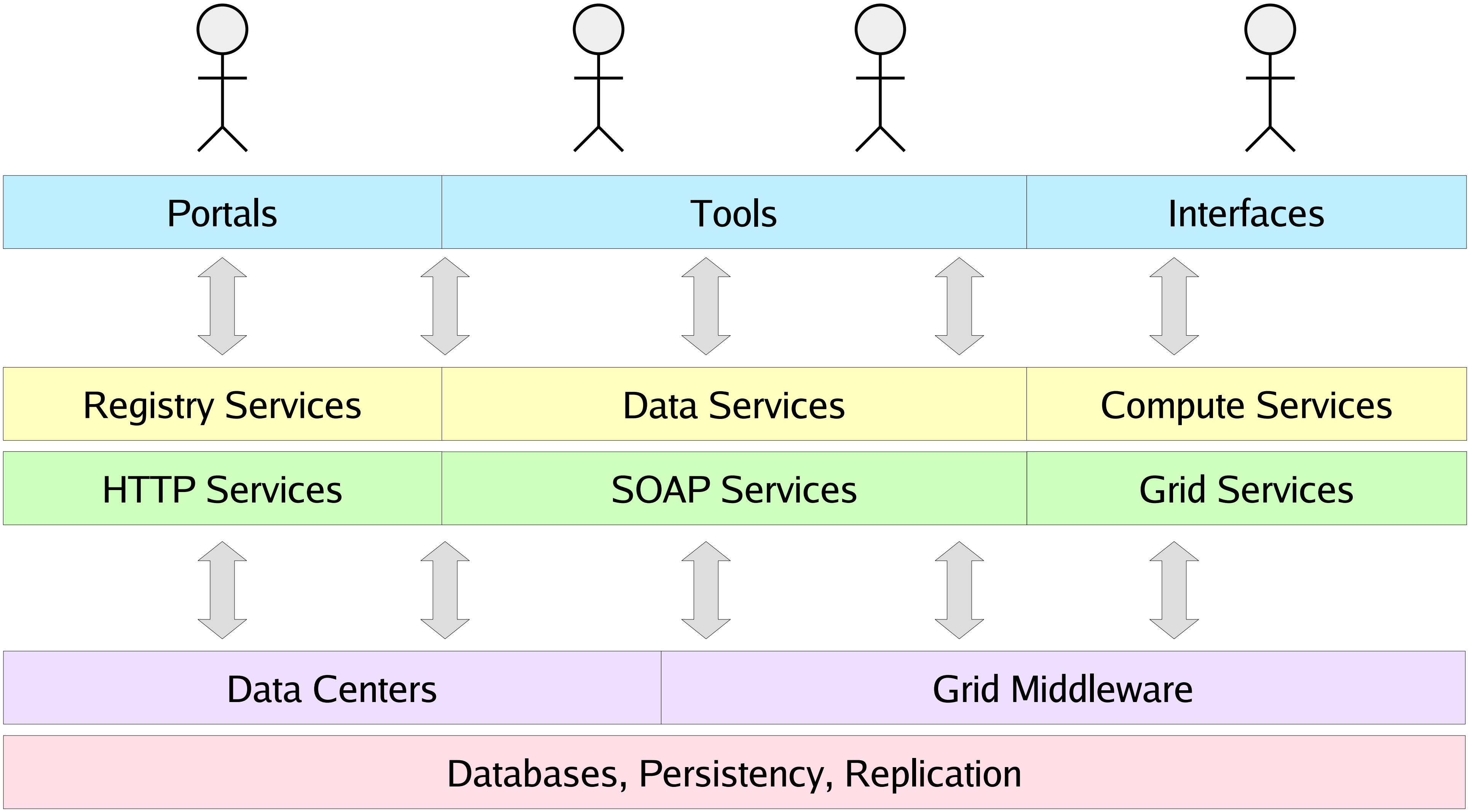
  A virtual observatory (VO) is a ***collection of interoperating data archives*** *and* ***software tools*** *that utilise the* ***Internet*** *to form a scientific research environment in which astronomical research programs can be conducted.*

  In much the same way as a real observatory consists of telescopes, each with a collection of unique astronomical instruments, the VO consists of a collection of ***data centres each with unique collections of astronomical data, software systems and processing capabilities****.*

Why VOs?

- Suppose we have a survey with associated data that could be useful for the astrophysics/astronomy community.
  - It is quite simple to put this data on the WWW.
  - Does it *really* allows others to use the data for further research?
  - In other words, is the data *really* ready to use?
- We could use data exchange formats and protocols that allow other software to use them.
- Other users could then acquire new data, do new experiments, etc., possibly making their results ready for use by others.

Simplified VO Architecture



Portals | Tools | Interfaces

Registry Services | Data Services | Compute Services

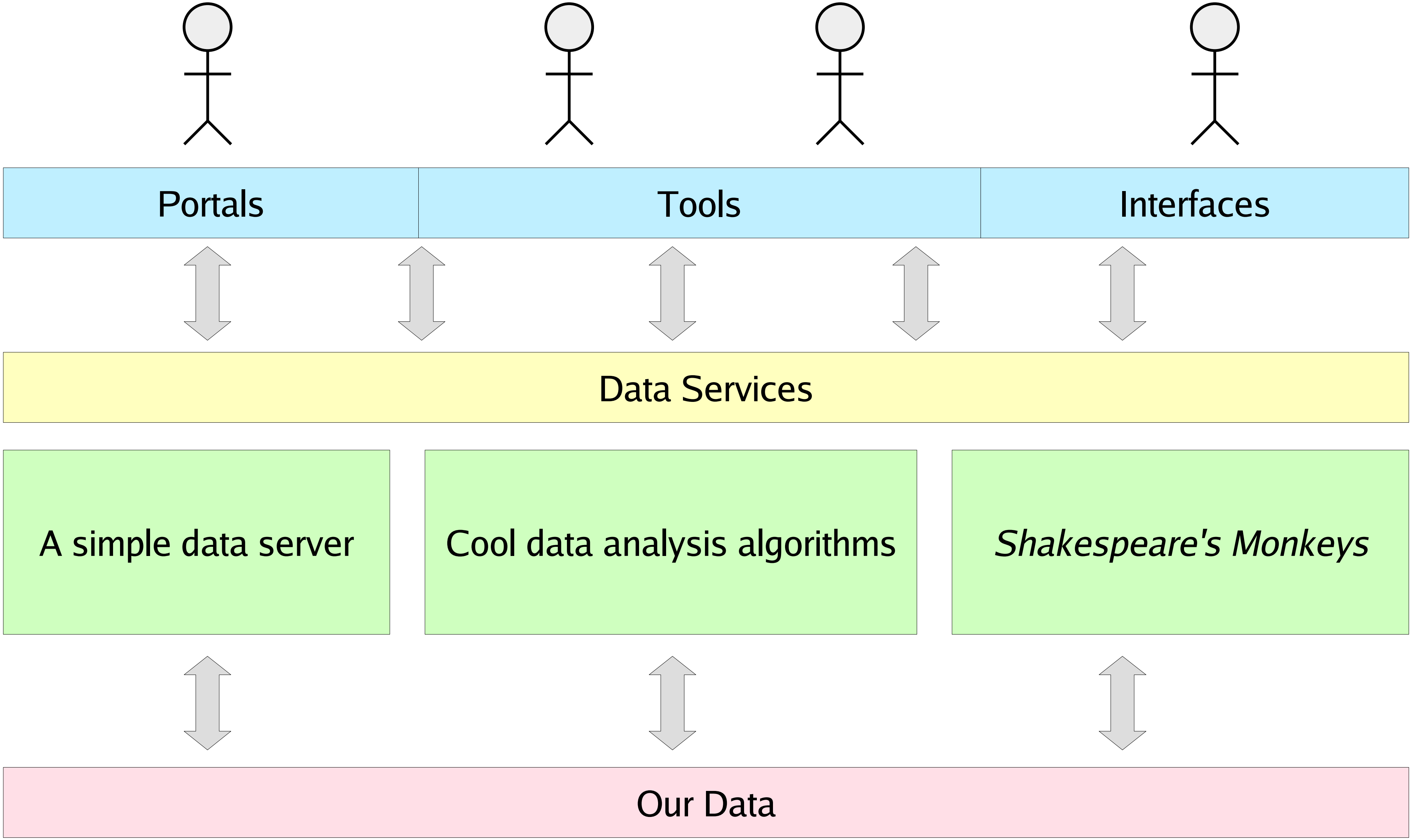
HTTP Services | SOAP Services | Grid Services

Data Centers | Grid Middleware

Databases, Persistency, Replication

Adapted from http://www.ivoa.net/Documents/Notes/IVOArch/IVOArch-20040615.html

Simplified VO Architecture



Portals

Tools

Interfaces

Data Services

A simple data server

Cool data analysis algorithms

Shakespeare's Monkeys

Our Data

In a VO...

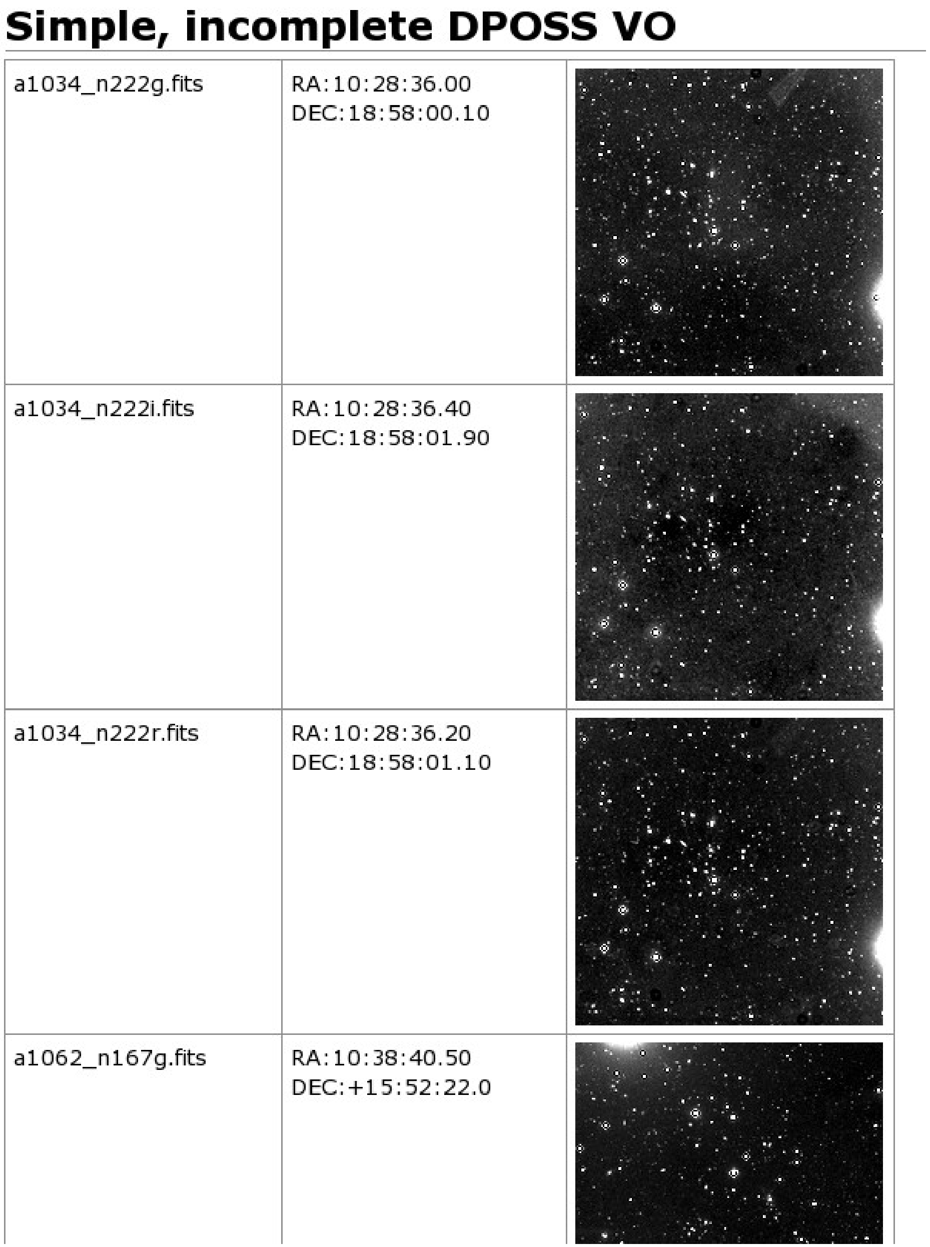
- ... as in the Web, the interface must be useful for humans, but also readable by computers (read XML), so we can automate some tasks.
- Internally the data format, storage methods, algorithms, languages, etc. may or not be standardized...
  - ... although CS people must consider building on the top of what is already done and issues like portability, replicability, readability, etc.

Why bother with further VO software development?

- Isn't it already done?
- *Biology easily has 500 years of exciting problems to work on.* – Donald E. Knuth.
  - More knowledge about biology leads to more questions and then to even more knowledge.
- And astronomy/astrophysics?
  - My guess: besides the same knowledge-questions-knowledge cycle, we will face an *enormous* amount of new raw data (new surveys.)
  - e.g. *Large Synoptic Survey Telescope*: 30 petabytes of data in five years: a pile of DVDs almost 70km high!

Things we're working on

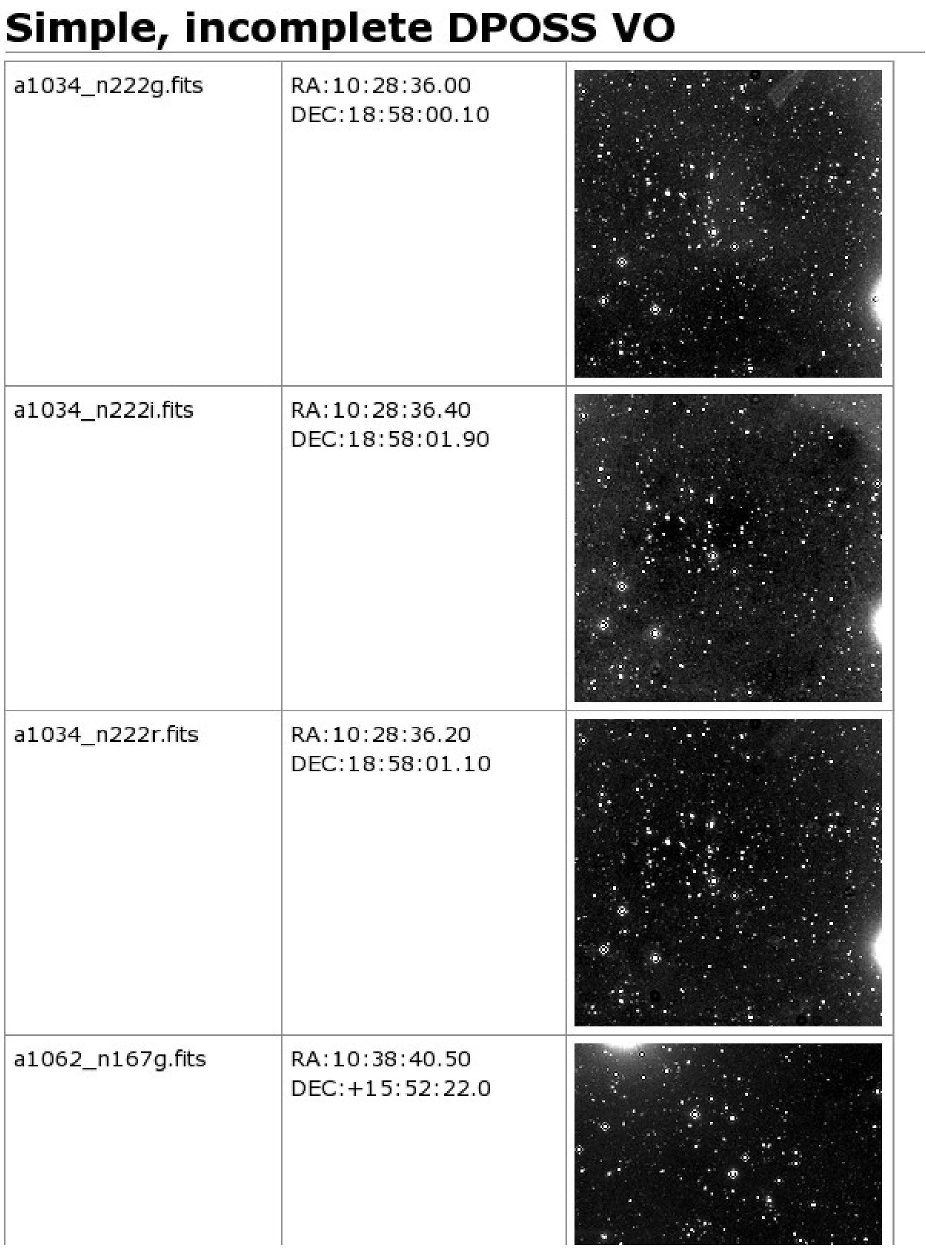
- DPOSS VO/CFVO (under construction).
  - Some 1760 FITS images.
  - Basic access to FITS headers, thumbnails.
  - Some integration with VO tools.
- Not really browsable (yet).



Simple, incomplete DPOSS VO

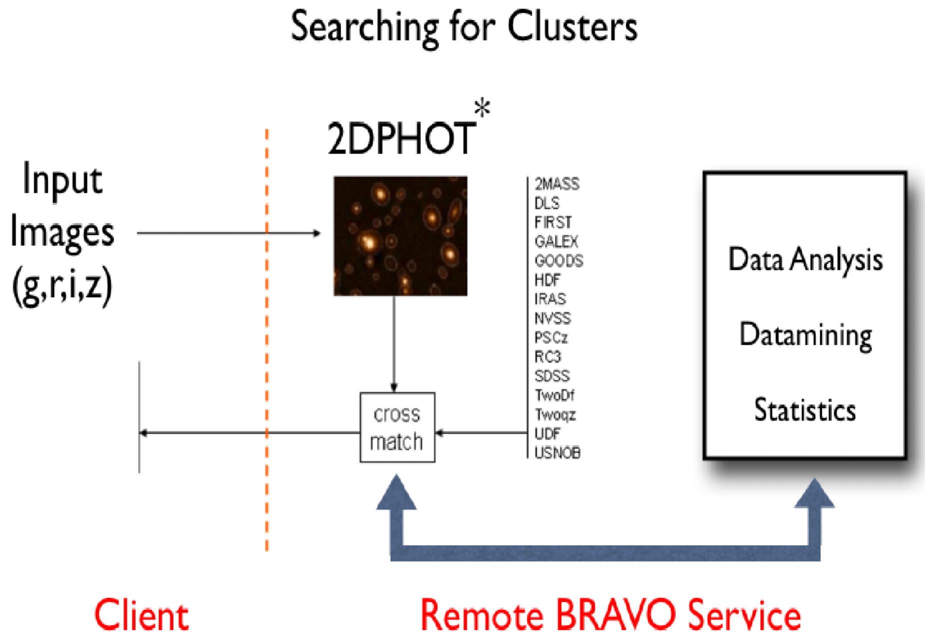
a1034_n222g.fits	RA:10:28:36.00 DEC:18:58:00.10	
a1034_n222i.fits	RA:10:28:36.40 DEC:18:58:01.90	
a1034_n222r.fits	RA:10:28:36.20 DEC:18:58:01.10	
a1062_n167g.fits	RA:10:38:40.50 DEC:+15:52:22.0	

Things we're working on



Simple, incomplete DPOSS VO

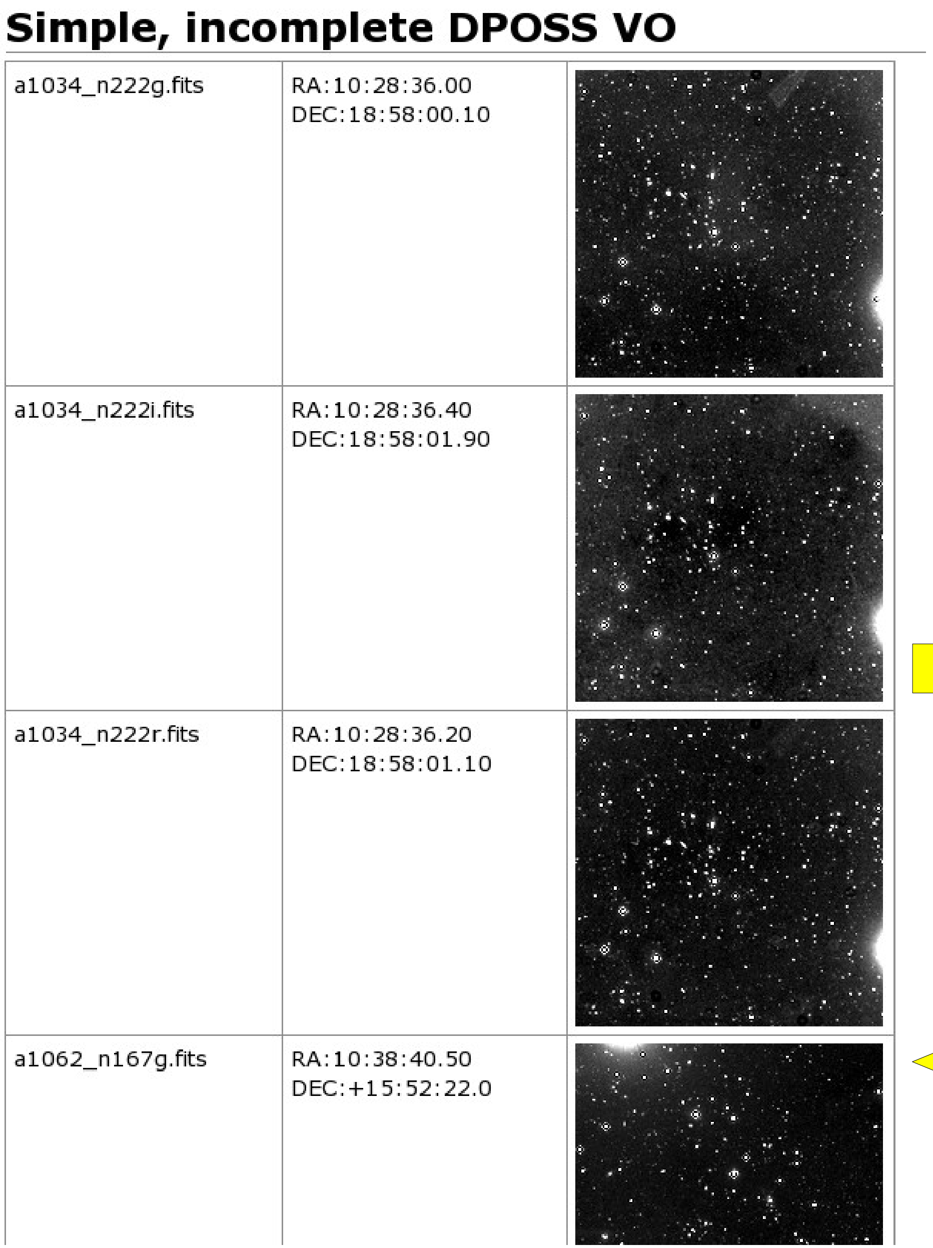
a1034_n222g.fits	RA:10:28:36.00 DEC:18:58:00.10	
a1034_n222i.fits	RA:10:28:36.40 DEC:18:58:01.90	
a1034_n222r.fits	RA:10:28:36.20 DEC:18:58:01.10	
a1062_n167g.fits	RA:10:38:40.50 DEC:+15:52:22.0	



* Two-dimensional Photometry Package developed by researchers at INPE, OAC, and ON

- Soon: VO-Compliance

Things we're working on

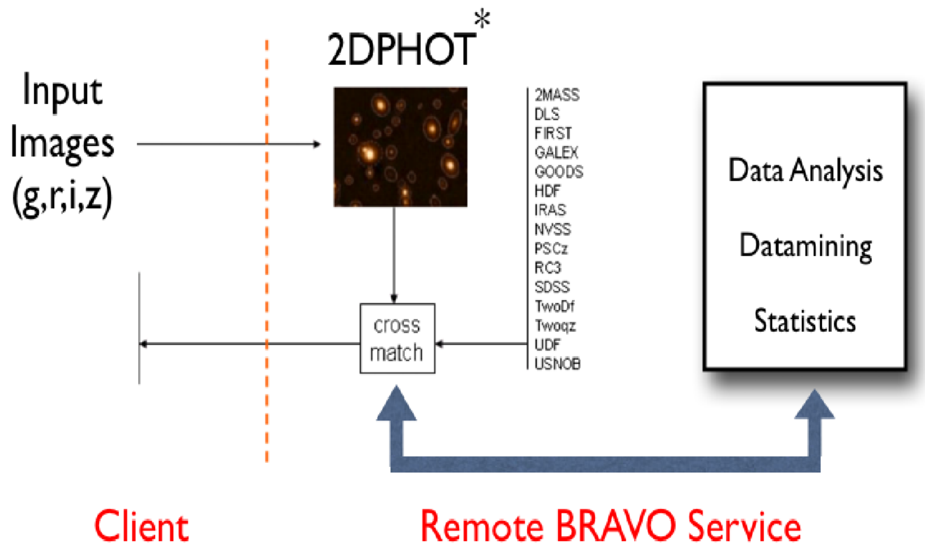


Simple, incomplete DPOSS VO

a1034_n222g.fits	RA:10:28:36.00 DEC:18:58:00.10	
a1034_n222i.fits	RA:10:28:36.40 DEC:18:58:01.90	
a1034_n222r.fits	RA:10:28:36.20 DEC:18:58:01.10	
a1062_n167g.fits	RA:10:38:40.50 DEC:+15:52:22.0	

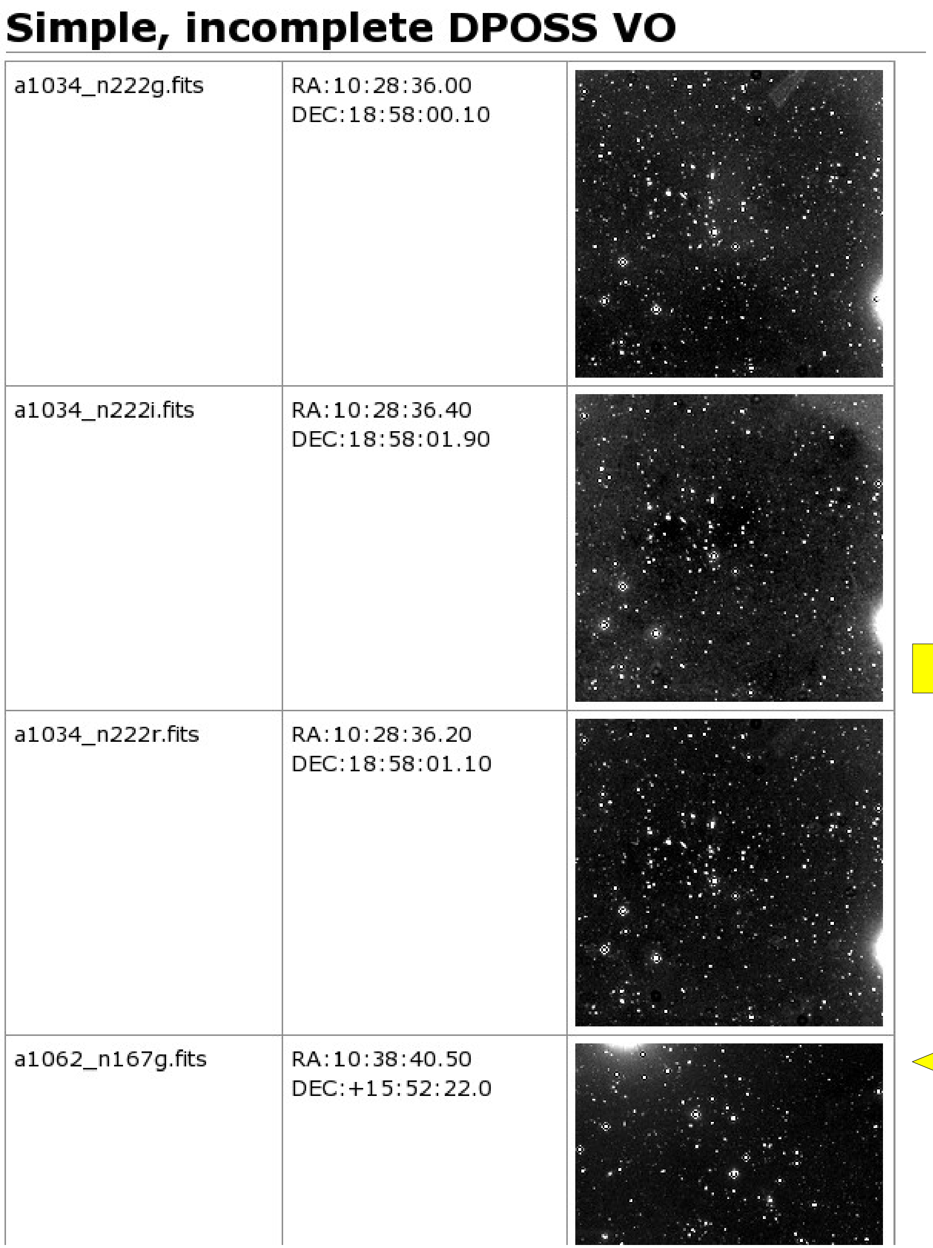
- GUI (Web) for 2DPHOT
- User can upload his/her images
- Parameter/Results Database

Searching for Clusters



* Two-dimensional Photometry Package developed by researchers at INPE, OAC, and ON

Things we're working on

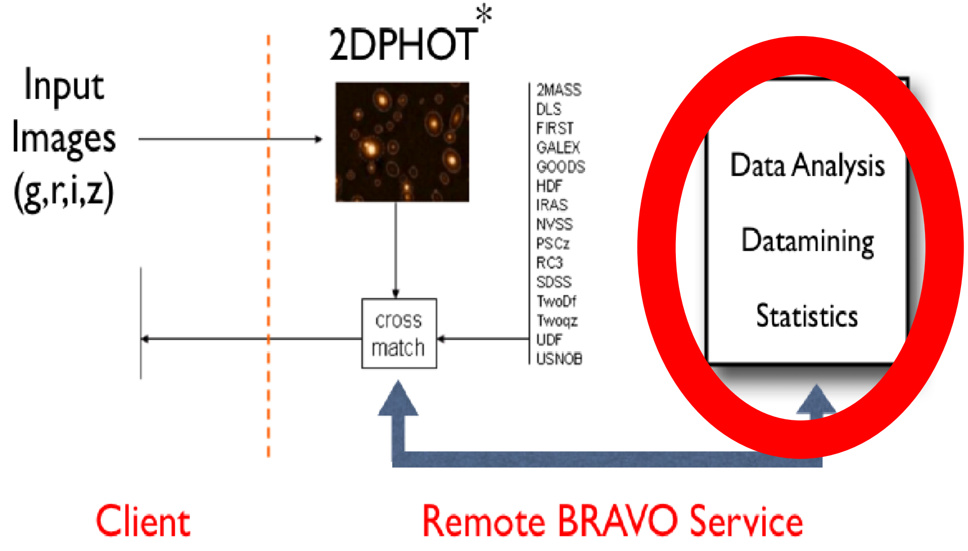


Simple, incomplete DPOSS VO

a1034_n222g.fits	RA:10:28:36.00 DEC:18:58:00.10	
a1034_n222i.fits	RA:10:28:36.40 DEC:18:58:01.90	
a1034_n222r.fits	RA:10:28:36.20 DEC:18:58:01.10	
a1062_n167g.fits	RA:10:38:40.50 DEC:+15:52:22.0	

- GUI (Web) for 2DPHOT
- Parameter/Results Database

Searching for Clusters

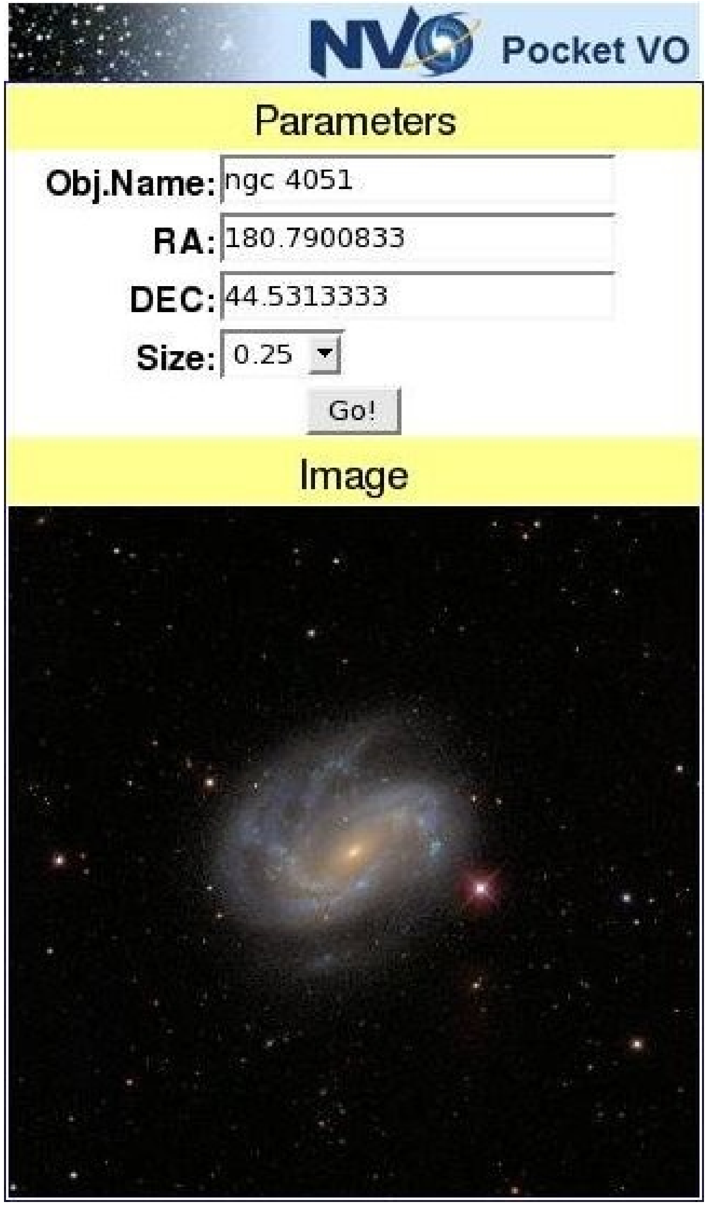


* Two-dimensional Photometry Package developed by researchers at INPE, OAC, and ON

Things we're working on

- The Pocket VO – a simple tool for educational purposes.
- Developed with Jordan Raddick (JHU), Iran Fernandes (LNA) at the NVO Summer School 2006.
- Requirements:
  - Should be portable and easily replicated.
  - Should be easily modifiable.
    - As long as the developers know something about the APIs.
    - We must do our part – simple code, no special frameworks, just a JEE container, any graphical WWW client, only lacks comments/documentation.
- Just for kicks, Pocket Pocket VO.

Things we're working on



Parameters

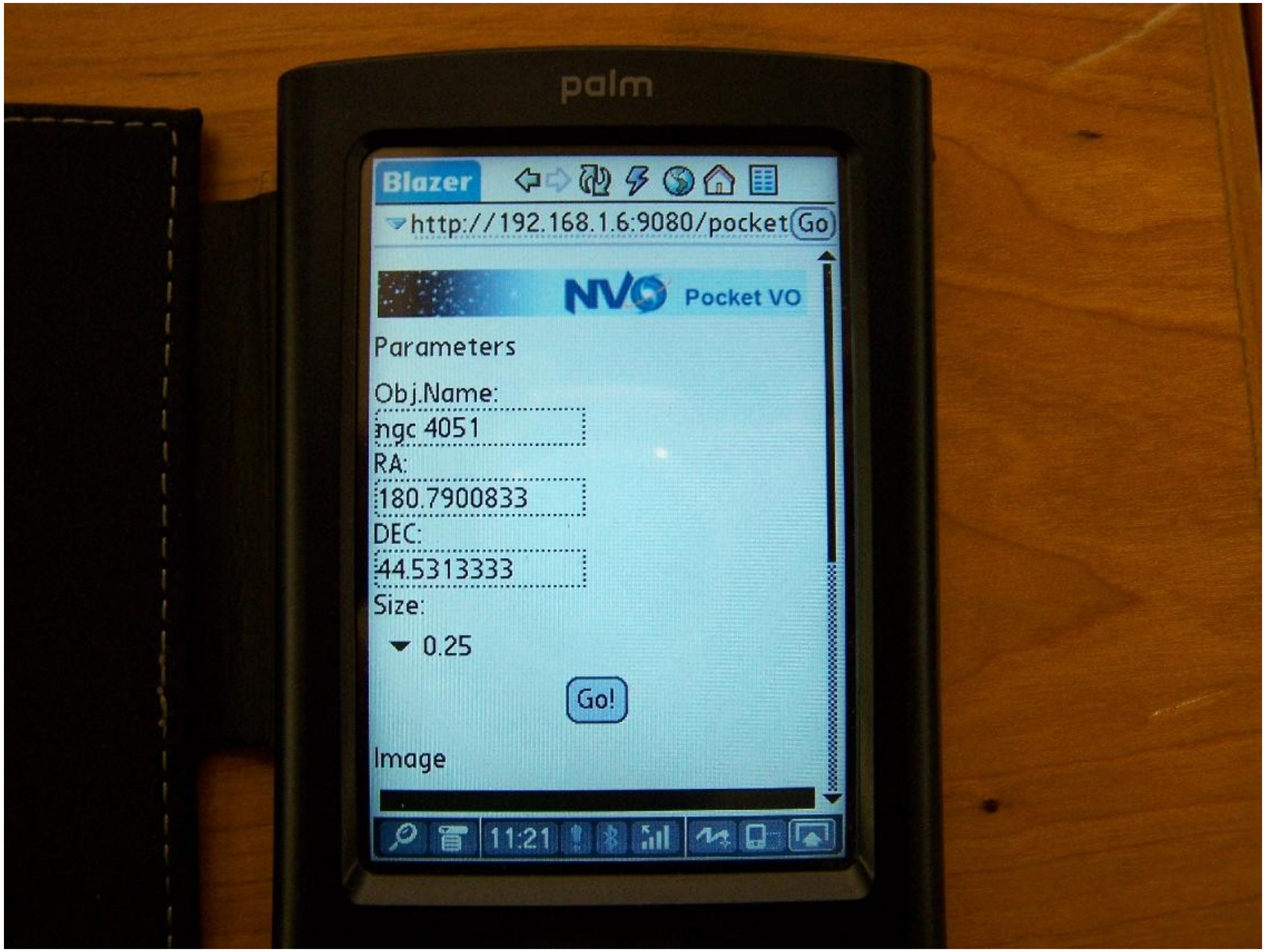
Obj.Name: ngc 4051

RA: 180.7900833

DEC: 44.5313333

Size: 0.25

Image



http://192.168.1.6:9080/pocket

Parameters

Obj.Name: ngc 4051

RA: 180.7900833

DEC: 44.5313333

Size: 0.25

Try it yourself!

http://www.lac.inpe.br/pocket/portable.jsp

(depends on other servers, e.g. Skyview).

Pocket VO: Under the Hood

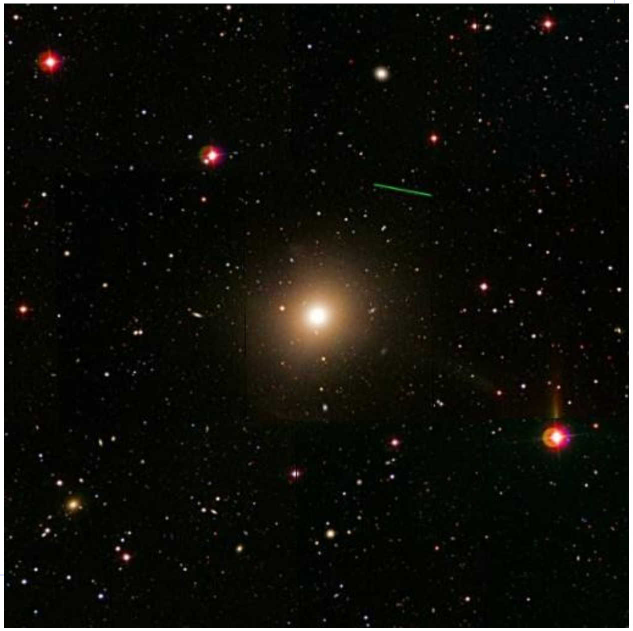
```java
package vo;

public class SesameDemo
  {
  public static void main(String[] args)
    {
    // locator creation
    SesameService locator = new SesameServiceLocator();
    // Sesame object
    Sesame myv = locator.getSesame();
    // Resolves the name for the object using a plain text format
    String result = myv.sesame("m89", "I");
    System.out.println(result);
    }
  }
```

```
# NGC 4321    #Q01013
#=Simbad: 1
%C AGN
%J 185.7289583 +15.8220833 (6) = 12 22 54.950  +15 49 19.50
%J.E [1799.00 1700.00  90] D 1999ApJS..125..409C
%V z +.005250 D [ .000017] 2002LEDA.........P
........
%I Z 99 - 30
#B 984
#---ServerTime(ms): 45
```

Pocket VO: Under the Hood

```
public class SIAPDemo
  {
  public static void main(String[] args)
    {
    double ra = 188.9166667;  double dec = 12.5563611;  double size = 0.5;
    String imName = "/tmp/image.gif";
    // Create a connection with the SIAP service.
    SiapConnection siap = new SiapConnection("http://casjobs.sdss.org/vo/DR5SIAP/SIAP.asmx/getSiapInfo?”+
                                             ”&FORMAT=image/jpeg&BANDPASS=*&");
    SiapQuery query = siap.getSiapQuery(ra,dec,size); // Form the query.
    query.addParameter("opt", "G"); // Enable the graphics overlay (SDSS specific parameter).
    QueryResponse qr = query.execute(); // Execute the query and fetch results.
    boolean path = false;
    if (qr.getRecordCount() > 0) // Did we get results?
      {
      QueryRecord r = qr.getRecord(0);
      path = r.getDataset(imName); // Download the image (may be jpeg!).
      }
    if (!path) // Maybe Sloan does not have it, should we ask DSS?
      {
      siap =
        new SiapConnection("http://skyview.gsfc.nasa.gov/cgi-bin/vo/sia.pl?”+
                           "digitized&");
      query = siap.getSiapQuery(ra,dec,size,"image/gif"); // Form the query.
      qr = query.execute();
      if (qr.getRecordCount() > 0)
        {
        QueryRecord r = qr.getRecord(0);
        path = r.getDataset(imName); // Download the image.
        }
      }
    }
  }
```



Pocket VO: Under the Hood

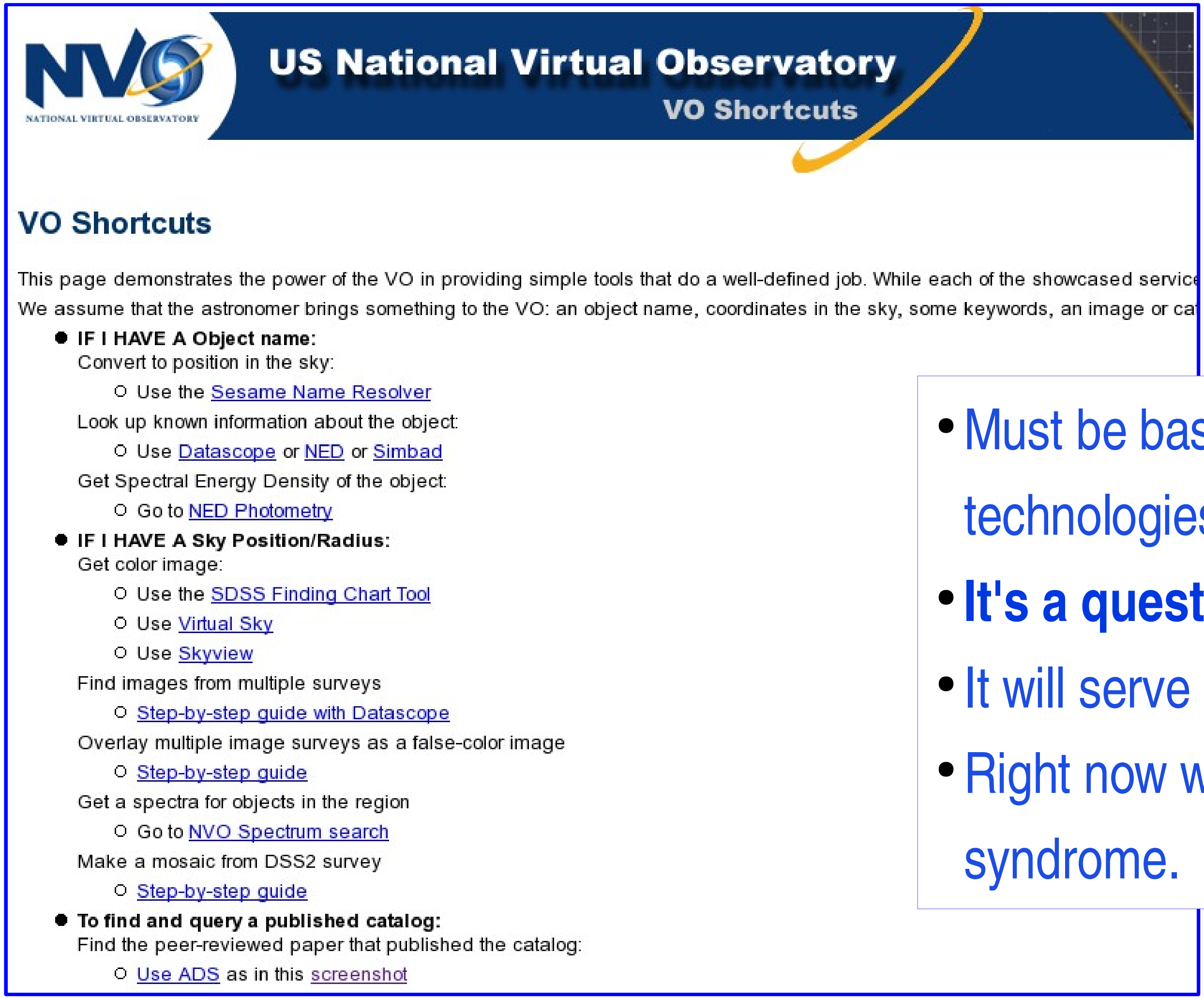
- OK, I lied.
- The final code could be really simple...
  - ... The Sesame demo has just a try/catch block, some imports.
  - There are several layers of software that allows the development of simple applications.
- Some of those layers are quite complex:
  - One **must** understand the architecture behind the layers.
  - One **must** understand the data and services' formats.

Another example: wrapping cone searches

```java
public class ConeSearchExample // Shamelessly adapted from a NVOSS example
  {
  public static void main(String[] args) throws Exception
    {
    String coneUrl="http://chart.stsci.edu/GSCVO/GSC22VO.jsp?";
    double ra  = 180.7900833;     double dec = 44.5313333;     double sr  = 0.25;
    URL coneSearch = new URL(coneUrl+"RA="+ra+"&DEC="+dec+"&SR="+sr);
    VOTWrap.VOTable vot = VOTWrap.createVOTable(coneSearch.openStream());
    // Assume one resource and one table.
    VOTWrap.Resource res = vot.getResource(0);
    VOTWrap.Table tab =  res.getTable(0);
    // Which fields are ID, RA and Dec?
    int fID = -1; int fRA = -1; int fDec = -1;
    for (int f=0;f<tab.getFieldCount();f++)
      {
      VOTWrap.Field field = tab.getField(f);
      if (field.getID().equalsIgnoreCase("ID")) fID = f;
      if (field.getID().equalsIgnoreCase("RA")) fRA = f;
      if (field.getID().equalsIgnoreCase("DEC")) fDec = f;
      }
    // Get the RA and DEC for the objects.
    for(int r=0;r<tab.getTableData().getTRCount();r++)
      {
      VOTWrap.TR row = tab.getTableData().getTR(r);
      VOTWrap.TD td_id = row.getTD(fID);
      VOTWrap.TD td_RA = row.getTD(fRA); VOTWrap.TD td_Dec = row.getTD(fDec);
      System.out.println((r+1)+": ID:"+td_id.getPCDATA()+" RA:"+td_RA.getPCDATA()+
                               " Dec:"+td_Dec.getPCDATA());
      }
    }
  }
```

Things we're working on

- A *Technical Portal* for education purposes.
- Aim: be sort of like the http://www.us-vo.org/shortcuts/ page for developers.



US National Virtual Observatory

VO Shortcuts

VO Shortcuts

This page demonstrates the power of the VO in providing simple tools that do a well-defined job. While each of the showcased servic

We assume that the astronomer brings something to the VO: an object name, coordinates in the sky, some keywords, an image or ca

- **IF I HAVE A Object name:**
  Convert to position in the sky:
  - Use the Sesame Name Resolver

  Look up known information about the object:
  - Use Datascope or NED or Simbad

  Get Spectral Energy Density of the object:
  - Go to NED Photometry
- **IF I HAVE A Sky Position/Radius:**
  Get color image:
  - Use the SDSS Finding Chart Tool
  - Use Virtual Sky
  - Use Skyview

  Find images from multiple surveys
  - Step-by-step guide with Datascope

  Overlay multiple image surveys as a false-color image
  - Step-by-step guide

  Get a spectra for objects in the region
  - Go to NVO Spectrum search

  Make a mosaic from DSS2 survey
  - Step-by-step guide
- **To find and query a published catalog:**
  Find the peer-reviewed paper that published the catalog:
  - Use ADS as in this screenshot

- Must be based on free, open CMS/e-learning technologies so we can add some nifty tricks.
- **It's a quest!**
- It will serve other purposes at INPE.
- Right now we're suffering from the NIH syndrome.

So you want to be a VO software developer...

- ***Bias warning!***
- Which languages and tools?
  - **Java.**
  - Open, simple, free, runs on all major OSs.
  - Several APIs for different VO related tasks (databases, image processing, concurrent processing, remote computing, AI, data mining, number crunching, etc).
  - Support for both WWW-based interfaces and rich client interfaces.
  - Rich client applications run on any OS via WWW (applets).
  - Same language (and some APIs) for desktop and web applications.
- Other languages offer *some* of those capabilities.

So you want to be a VO software developer...

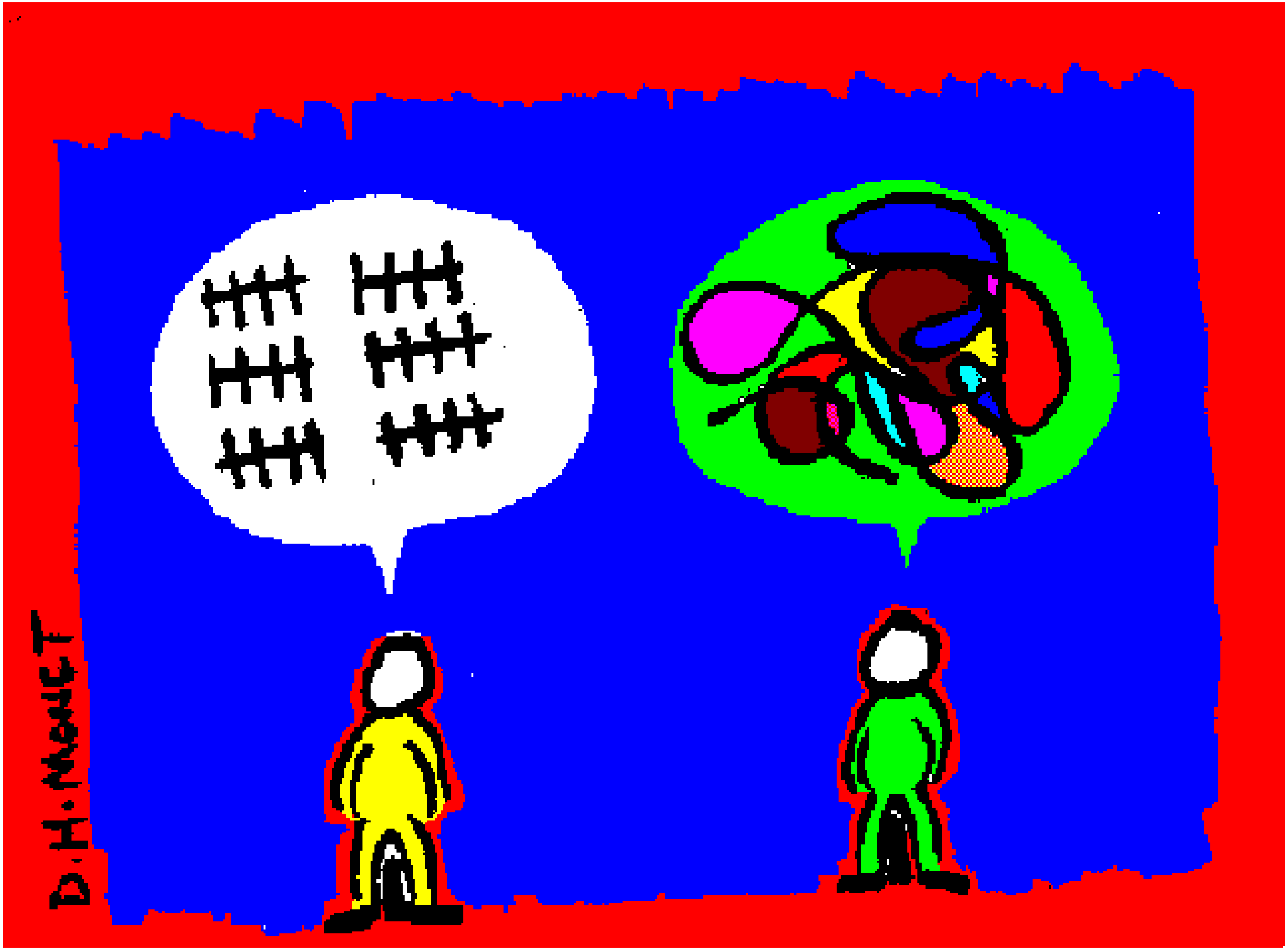
- VO APIs are also available in Python, C#.
- More specific environments/languages: IDL, IRAF, PYRAF.
- I'm a big fan of UNIX-based systems:
  - Several free Linux flavors.
  - Sysadmin scripts in several languages!
  - Free (*libre*) databases, languages, servers, utilities, etc.
- SQL, XML knowledge useful.
  - VOTable essential!

So you want to be a VO software developer...

- Careful consideration of platforms and languages, *even with separation between interfaces and implementations*.
  - In other words, avoid “closed” software, special requirements, human-centric, sysadmin-depending systems.
  - Consider system replicability, human readability.
- Of course, it all depends on the tool being created.

- **The major challenge for CS people is...**

Final Considerations



home.ca.inter.net/~dmonet/cartoon/archive/b10.gif

Thanks!

Questions?