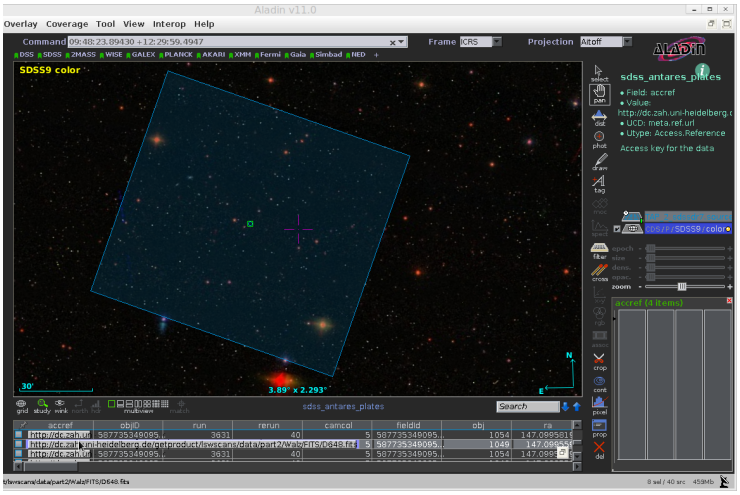
Dancing the VOltz



Hendrik Heinl, Dave Morris

November 1, 2021

Abstract

The shear amount of data available for astronomy today, offered by distributed services not only demands a specific set of skills from (data) scientists: it's obvious that automated processes and machine readable (machine "learnable") interfaces and standards are necessary to combine data from different sources, instruments and messengers. The IVOA defines such standards and protocols and introduces them to the community.
The process of the development of standards and their evolution is an ongoing task. At two Interoperability meetings (or "Interops") per year, the IVOA members assemble. These meetings are rather working meetings than typical conferences and understanding the many terms and abbreviations used throughout the sessions is quite challenging for a newcomer.
This tutorial shall give an introduction to the organisation of the IVOA, the working process, and also the terms used. It shall give an overview on existing standards, those currently in development and make it easier for newcomers to get into the IVOA meeting.
This tutorial is presented at the newcomers session for the IVOA Interoperability meeting in May 2021

Software: TOPCAT V4.8, Aladin V11.0PyVO, PyVO

1 Starting out

We will use the VO tools TOPCAT and Aladin. These client software can be found here:

Aladin: `http://aladin.u-strasbg.fr/`

TOPCAT: `http://www.star.bris.ac.uk/~mbt/topcat/`

We will also use PyVO and Astropy. For Linux users we recommend to use your distribution packages. In debian this would be `apt get install python3-astropy python3-pyvo`. For other OS you can follow the installation instructions:

Astropy: `https://docs.astropy.org/en/stable/install.html`

PyVO: `https://pyvo.readthedocs.io/en/latest/`

A comprehensive collection of the examples in this tutorial as well as installation scripts you will find on github.

2 TOPCAT, SCS and TAP

This tutorial will work along the use case of searching for neutrino data from a VO-service and combine these with data from catalogues using Topcat. In the

second part we will be using PyVO to find and access observational dataproducts on different VO services. We are well aware that scientifically speaking this does not make a lot sense (yet), but this tutorial is meant to give an overview over the existing VO standards and protocols, and how these are developed within the IVOA Working Groups and Interest Groups.

▷ **1** *Finding Neutrino data in the VO-registry* –

We start this use case with searching and accessing neutrino data that is available on VO services. Start Topcat and click on VO → Cone Search. In the Cone Search window enter `neutrino` as Keywords and click on Find Services. After a few moments you see the results coming in as a list of services providing data related to the keyword. You see that the list provides you with some information of what the data of the service contains. In the description you also get information about the different data providers.

VO Registry

The Registry is the central point for searches for services in the VO. VO services identify themselves to the Registry, providing metadata about the data they serve and the service protocols they support. Thus, the Registry is the entry point for data discovery in the VO.

Mark different rows and watch the URL change in the field below. This meta data is meant to give you an idea of what you can expect from the services and if they are of interest for you.

With this little example you already get an idea of what the VO standards and protocols are about: to standardize publishing, searching and accessing data and making it reusable.

Exercise: Of course you may not be interested in neutrino data. Try to find the messengers that are of importance for you instead: try radio, infrared or gamma ray, or whatever is suitable for you. Note: here the registry is searching for SCS services only. The searches for other service protocols (SIA, TAP, SSAP) work similar.

▷ **2** *Retrieve neutrino data through a Cone Search* –

For our example, we select the service provided by km3net by marking the service by the title ANTARES 2007-2017. We will get a bunch of neutrino events to use in the next steps. In Cone parameters for RA enter 170, for Dec enter 25 and and give a Radius of 30 degrees. Then click on OK. Within seconds the data will be loaded into Topcat and you see the table appear in the main window of Topcat.

The protocol we used is the Simple Cone Search (SCS).

SCS

SCS (Simple Cone Search) defines a standard web service interface for requesting data based on a cone projected on the sky described by a sky position and an angular distance. The protocol defines the query parameters the service understands and the format of the response. This enables users to query multiple services using the same criteria. Each service returns a machine readable VOTable document listing the astronomical sources or objects which are within the search criteria.

The table we received from the SCS is a VO-Table, which not only contains the actual data, but also some metadata.

VOTABLE

VOTable is the standard data exchange format for tabular data in the VO. The VOTable standard defines an XML schema that combines data types, units, UCDs and data model references to provide machine readable metadata describing each column in the table, enabling data processing and display software to understand how to use it.

So let's have a look at it. We click on Views → Column Info. Here you can get some information about the columns. You see that the data provides us with a modified Julian Date of when the neutrino event was ovserved. If we scroll a bit to the right, we see the UCD (Unified Content Descriptors).

We can plot some of the data with Graphics, but so far we wouldn't have much more than positions to see.

The table we received from the SCS is a VO-Table, which provides not only the actual data, but also some metadata.

▷ **3** *Adding catalogue data to the neutrino candidates with TAP* –

In this step we want to combine our neutrino data with catalogue data from SDSS. Catalogue data typically is table data, so we will use the Table Access Protocol and the corresponding query language ADQL to do so. There is a lot more to say about TAP and ADQL than we will do here. A comprehensive ADQL introduction is linked in the references.

In Topcat go to VO → TAP. The TAP window will open and after a few moments you will see a list of services in the window. Analogously to the Cone Search window, you can use the TAP window to search the VO-registry for data. At Keywords we will enter `SDSS` to search for services providing us with those data. Often you will find the most recent catalogue data from dedicated services related by the publishers of the catalogues. You may also find them

on the TAP Service of Vizier. Many Services providers try to anticipate which data combination might be of interest for the users. So it's not uncommon to find surveys like 2MASS or SDSS on several services.

TAP

TAP (Table Access Protocol) defines a standard web service interface for querying tabular catalogs. The protocol defines how to represent the query parameters and the available response formats. This enables users to query multiple services using the same client software and query language.

As for SDSS you see a few options to chose from. We will select the GAVO data center (Note: the reasons here is due to the performance of the tutorial: we want query results to come in fast. Real science takes time and it is completely acceptable to run queries for several hours). Click on Use Service to get to the TAP window of the Service.

Give it a few moments to load the meta data of the service into the TAP window. On the left you see a tree of the catalogues and tables in this service, on the right is the metadata browser on these tables. Mark any table and see what the description of the table and the columns tell you about the data in the table. This meta data still is part of the TAP standard and is very helpful for data discovery and of course for the data access using ADQL.

In the upper left of the meta data at Find: enter `sdss` to find the table containing this survey. In this example we just want to collect the identifier of an object, the position in ra and dec and the colours in the u, g, r, i and z band. We will make use of the TAP feature TAP UPLOAD, to join our local table with results from the remote service. In the lower window at ADQL Text click on Examples → Upload → Upload Join. You see an example ADQL query appearing in the field:

```
SELECT
   TOP 1000
   *
   FROM sdssdr7.sources AS db
   JOIN TAP_UPLOAD.t1 AS tc
   ON 1=CONTAINS(POINT('ICRS', db.ra, db.dec),
                 CIRCLE('ICRS', tc.ra, tc.decl, 5./3600.))
```

Now at the first glance that may look confusing, so let's go through it step by step, customizing it to fit our science case as we go. Each ADQL query starts with `SELECT` followed by specifications of "what" to select, what to do with the selected records and how to return the results. `TOP 1000` means the first 1000 data records will be returned, which match the whole query.

With * we select all columns of matching data records. Here will will modify the query, because we are not interested in all of the columns of SDSS. Instead of * we will write: `antares.*, sdss.objid, sdss.ra, sdss.dec, sdss.u, sdss.g, sdss.r, sdss.i, sdss.z`. This means we want to keep all columns from our uploaded table (which we will call `antares`) and add the columns from the remote table (which we will call `sdss`). The phrase `FROM sdssdr7.sources AS sdss` specifies the table of the TAP service we want to query `sdssdr7.sources` and the short name we want to use for it `sdss`. The next lines are the query conditions. In our example we use the ADQL built in functions that defines the upload join. The first part `JOIN TAP_UPLOAD.t1 AS antares` specifies that we want to join the data from SDSS with data in the table of neutrinos from Antares that we uploaded along with the query, and the short name we want to use for it. Then we describe the criteria for joining the data by defining a circle around each object in our table of Antares neutrinos, and asking the database to check if any sources in the SDSS table lies within any of these circles. A little confusing may be the `1=CONTAINS`. This is due to the boolean result returned by the function `CONTAINS`. A result of `1` means `true` whereas `0` means `false`. `POINT` expresses a point, the right ascension and the declination in degrees. Our query takes the coordinates from the table using the columns `ra` and `dec`. Analogously `CIRCLE` expresses a cone in space, taking `ra` and `decl` from our local table. The last entry defines the radius of the circle in an angle in decimal degrees. The whole query looks like this:

```
SELECT
   TOP 1000
   antares.*, sdss.objid, sdss.ra, sdss.dec, sdss.u, sdss.g, sdss
       .r, sdss.i, sdss.z
   FROM sdssdr7.sources AS sdss
   JOIN TAP_UPLOAD.t1 AS antares
   ON 1=CONTAINS(POINT('ICRS', sdss.ra, sdss.dec),
                 CIRCLE('ICRS', antares.ra, antares.decl,
                     10./3600.))
```

Click on Run Query and the result should come in rather fast. Again have a look at the new table. You see that we added a few columns, but it seemed that we lost some rows. This is due to the query: the database did only return those data records that matched our conditions. Not matching records are discarded.

Exercise: You can control the ADQL JOIN behaviour by using the alternatives `LEFT OUTER JOIN`, `RIGHT OUTER JOIN` or `FULL OUTER JOIN`. The default command is `INNER JOIN`. Play around with these options, they might become usefull in future.

ADQL

ADQL (Astronomical Data Query Language) is a query language based on a subset of SQL used to query tabular data in TAP services. The ADQL standard defines a common subset of SQL that can be used to access data in all the common relational database platforms used in astronomy. This enables users to query multiple services using the same query without having to worry about the specific dialect of the relational database platform hosting the data.

With the colors of SDSS we could now analyze the neutrino sources and do more reasonable science. But here we want to talk about the metadata for a moment. You remember how we clicked on Examples in the TAP-Service window of Topcat, and Topcat completed an ADQL query for us. Did you notice how Topcat "knew" which columns contained the data to complete positions ? And Topcat did this for both tables, our local one, and the one in the remote service. The secret here lies in the metadata that is provided by VO-standards. In Topcat's main window go to Views → Column Info. In the new opening window you will find the metadata on the columns, like a human readable description, the units of the data, and the Unified Content Descriptors (UCD). These are part of the "magic" working in the background of the VO.

UCDs – VO semantics

UCDs (Unified Content Descriptors) describe the content of table columnms in a machine readable way. Thus, the machines "know" something about the data and you can make use of this in your programs. The data type of a column may describe it as a floating point number, the units may say it is measured in degrees, but the UCD allows us to identify it is a *right ascension position coordinate*, enabling data processing and display software to understand how to use it.

▷ **4** *Getting a Table with DataLink* –

Once you have a certain level of interoperability, you can go on and add more functionality and interactivity between clients and servers or in between clients. One of the standards developed for this is Datalink. Some services and clients can already consume datalinks. Luckily, the GAVO service, Topcat and Aladin are capable of this. So at this point make sure that Topcat and Aladin are running. Then go to the Topcat TAP window, select the GAVO service and run the following query to obtain a datset containing DataLinks:

```
SELECT
   TOP 10000
   *
   FROM sdssdr7.sources AS sdss
   JOIN TAP_UPLOAD.t1 AS antares
   ON 1=CONTAINS(POINT('ICRS', sdss.ra, sdss.dec),
                 CIRCLE('ICRS', antares.RA, antares.Decl,
                     5./3600.))
   JOIN lsw.plates as plates
   ON 1=CONTAINS(POINT('ICRS', plates.centeralpha, plates.
       centerdelta),
                 CIRCLE('ICRS', antares.RA, antares.Decl, 1.))
   AND plates.accsize<200000000
```

Note that we added another `JOIN ON` with a geometry defined by the antares data, but also on the remote table lsw.plates. Here we want to get some information about scanned plates from the "Landessternwarte Heidelberg´´. The snippet `AND plates.accsize<200000000` is added because some of the FITs files exceed 1 GB of data volume, which would take quite some time to download. So for the sake of the tutorial, we limit the filesize. Again, click on Run Query. It will take a few seconds until the data is available. Then go Topcat's main window, click on Views → Activation Actions. A new window will open. On the left side click on Invoke Service and check the box next to it. See if in the main window as Action `View DataLink Table` is selected. Now Topcat is prepared to interprete the DataLink block of the VOTable. Click on the flash-button below to "Invoke Action on first row".

Datalink

The Datalink standard connects meta data of datasets to the data, related data products, and web service capabilities that can act upon the data, without having to download the whole dataset.

▷ **5** *Using DataLink* –

A new window will open and show you the DataLink Table. Here you find four rows: `#auxiliary`, `#proc`, `#this`, and `#preview`. If you scroll through these four entries, you notice, that `#auxiliary`, `#this`, and `#preview` contain links to fits file, and a possible Action is `View image internally`. With this, you can open a small window in Topcat that will either show you a 1/4 scaled image, the whole dataset, or a small cutout of the image. Let's use a proper image analysis software to have a look at the images: Aladin. Mark `#auxiliary` and in the drop down menu at the buttom of the window selec at Type: `FITS_Image` and

next to it at Action `Send FITS Image`. Then switch to Aladin and see how the image is loaded. You can open the Topcat Data Display and select a different row of our table, and then repeat the step of sending the FITS-image to Aladin. Note that whenever you hover over the "SAMP" planes in Aladin, how it will compute the covered area of the image and display it in the main window. If your screen is big enough, you can easily display Topcat and Aladin on the same desktop and switch between them.

Exercise: For beginners: If you are interested in more interoperability, send the table we obtained in the earlier tutorial step to Aladin via SAMP and see where the sources of our neutrinos are in the images. We don't claim that you actually see the neutrino sources in the images. But if you do, let us know, because it would be multi messenger astronomy! Select data points in Aladin and watch how they are highlighted in Topcat.

Exercise: For advanced: If you feel ready to get into the depths of the VO: save the Table of this last tutorial step to a VO-Table and look at it in your favourite text editor. Can you figure out which bit is defining the DataLink functionality ?

Exercise: For the VO-pro: You may have noticed that we did not talk about `#proc` above. That is because it's linked to special service side implemented features, in this case the SODA standard, which can be used to perform a cutout of the FITS images. Play around with it.

3 PyVO and ObsTAP

PyVO is the Python API to the VO standards. You can use it for data discovery and data access in the same way as you would use VO clients like Topcat or Aladin. But you can also use it embedded in your own code, so that you may access data remotely from an automated script. We will show with three simple (and one not so simple) scripts how you can use PyVO in the multimessenger context.

▷ **6** *Example 1: PyVO and TAP* –

Have a brief look at the few lines in example1.py. Note, that two lines are necessary to supress warnings – services and clients often are a bit off, especially in regard of recent standards. They still work, but warnings will be raised. We

don't want to pretend that everything in the VO is perfect, but within a tutorial, warnings might be confusing. If you are curious what's going on, simply comment the according lines and run the script. The script performs a very simple query on the GAVO obscore service. Have a look how the service object is built by `pyvo.dal.TAPService()`. This is the convention within PyVO: `pyvo.dal.SERVICETYPE(parameters)`. The actual search is performed with the service object method `search`. This will send the query to the server, and the last line saves the resulting VOTABLE into the same folder as the script.

▷ **7** *Example 2: PyVO and SCS* –

This example shows analogously the Simple Cone Search from PyVO. The service object is built following the convention introduced in the last tutorial step, and as you see, we again use the service provided by km3net to obtain a list of neutrino events. Here we chose a much smaller radius though. We want to keep the following steps fast. Note the last lines of the script: we save the result to a local file, but we also use SAMP to broadcast the result to Topcat. To make this work, you need to make sure that topcat is running before you run the script. You will find the result as a table in Topcat now.

SAMP

SAMP (Simple Application Messaging Protocol) enables astronomy applications on the same desktop or laptop machine to share data with each other. It also enables applications to notify each other about what data points a user has selected, enabling the two applications to coordinate their display and selection views.

▷ **8** *Example 3: PyVO and Obscore* –

This example now is a bit closer to the real world. The script will make use of the neutrino data we get from example 2, which we saved locally. So we take different steps from here: we load the data from the folder example2. Note that we have a fallback exception in the code, in case something went wrong during the last tutorial step. We also introduce a longer TAP query. From the first part of the tutorial you are familiar with the TAP-upload already. But here we are comparing our local data with a special table on the TAP service: ivoa.obscore. A service providing this table follows the special standard Obscore, which is dedicated to observational data. Each obscore-service provides this table with exactly defined columns which enables the users to use the same query repeatedly on each obscore-service. Of course there is no garantuee that one will receive any results, but it's making searches all across the VO possible, and pretty easy. Here we query the gavo dc. only, and send the results to Topcat.

▷ **9** *Topcat, SAMP and Aladin* –

ObsCore

ObsCore Is a common data model for describing astronomical observations. It defines the core components of metadata needed to discover what observational data is available from a service. Every service that advertises itself as an ObsTAP service must provide this standard view of metadata lsiting the observational data available from that service.. This means that a user can build an ADQL query based on the ObsCore data table and apply the same query to all the ObsTAP services.

Look at the results in Topcat. We will have a list of 50 images. Note: it's a table that does not contain the actual images, but urls to the images. Topcat can not deal with these links. Therefore, start Aladin, then go back to Topcat and use SAMP to sent the data to Aladin. In Aladin you now find this table and can download the fits files to your local machine. Hover over the rows to see the coverage of the images in relation to the position on the sky you were searching. Scroll a bit left and click on the buttons in the column "Preview" to download smaller Versions of the images in Aladin.

4 Key Standards

FITS

Though no VO-standard, FITS of course is a well establish data format in astronomy, and therefore is widely adapted by the VO and used in many client software. Due to the limits in regard of the space of metadata e.g. the lack of standardized vocabulary, the demand for a metarich standard arose: the VO-Table standard.

VO-Table

The centre of most of the VO standards is the VO-Table standard. This is the format of choice for data exchange within in the VO. Services will serve clients VO-Tables, and clients sent VO-Table via SAMP. Based on XML, VO-Tables provide the flexibility, extendability and precision needed to annotate data. Throughout many discussions about standards you will see XML snippets being presented as a solution for a problem. If you read a bit into it, these soon will make a lot of sense for you too.

5 Acknowledgements

The Authors thank the ESCAPE project from the HORIZON 2020 research and innovation programm under granz n°824064 for making this contribution to the 2021 May Interop possible.

Dave Morris works as a Research Software Engineer at the Royal Observatory, Edinburgh (ROE)

Hendrik Heinl works as Ingénieur d'études at the Centre de Données astronomiques de Strasbourg (CDS)

6 IVOA Components

The IVOA is organised in "Groups of interest and competence", with some groups being in charge of the development and evolution of standards (mainly the Working Groups), whereas other groups are focus on the practices of data interoperability and the demands of specific groups within the astronomical community (e.g. the Radio astronomy IG or the Solar System IG). IGs help to identify demands, WGs work out if and how VO standards can reply to these demands.

Working Groups

The following WGs are established in the IVOA and are assigned a specific focus on the VO standards. Often a standard will touch the focus of several WGs. Then a joined session of the WGs will be held at the Interop meetings.

Application WG

As the title Applications WG 1 implies, the focus of this WG is on the perspective on the VO from the view of an application developer. Thus, the Sessions include discussions and exchange of VO standards implementation, but also new demands to these standards deriving from the use of applications. The application WG could be linked to any standard or protocol an application might speak or consume, so the involvement is everywhere in the VO. You will find further information here:
https://wiki.ivoa.net/twiki/bin/view/IVOA/IvoaApplications

Data Access Layer WG

The Data Access Layer WG is tasked with the definition and formalizatio of standards for remote data access. As such, the WG is in charge of standards like SCS, SIAP, TAP/ADQL and others. A more detailed description can be

found here:
https://wiki.ivoa.net/twiki/bin/view/IVOA/IvoaDAL

Data Model WG

The role of the Data Modeling group is to provide a framework for the description of metadata attached to observed or simulated data.
https://wiki.ivoa.net/twiki/bin/view/IVOA/IvoaDataModel

Grid and Web Services WG

The aim of the GWS WG is to define the use of Grid technologies and web services within the VO context and to investigate, specify, and implement required standards in this area.
https://wiki.ivoa.net/twiki/bin/view/IVOA/IvoaGridAndWebServices

Resource Registry WG

The Resource Registry WG is responsible for maintaining the live registry and extending the metadata to describe new service.
https://wiki.ivoa.net/twiki/bin/view/IVOA/IvoaResReg

Semantics WG

The Semantics Working Group (Charter) superseded the UCD Working Group in 2006 as it was realised semantic resources beyond UCDs were required. To this day, the main focus of the Semantics WG is the maintenance and development of Vocabularies.
https://wiki.ivoa.net/twiki/bin/view/IVOA/IvoaSemantics

Interest Groups

IGs address a specific topic and often are linked to external communities. The Data Curation and Preservation IG has bounds to the Research Data Alliance (RDA), whereas IGs like Radio astronomy IG or Solar System IG link to specific fields in astronomy. IGs connect the IVOA with these different communities and enable outreach, but also input from these communities.

Data Curation and Preservation IG

Data Curation and Preservation IG
https://wiki.ivoa.net/twiki/bin/view/IVOA/IvoaDCP

Education IG

Education
https://wiki.ivoa.net/twiki/bin/view/IVOA/IvoaEducation

Knowledge Discovery IG

Knowledge Discovery IG
https://wiki.ivoa.net/twiki/bin/view/IVOA/IvoaKDD

Operations IG

Operations IG
https://wiki.ivoa.net/twiki/bin/view/IVOA/IvoaOps

Radio astronomy IG

Radio astronomy IG
https://wiki.ivoa.net/twiki/bin/view/IVOA/IvoaRadio

Solar System IG

Solar System IG
https://wiki.ivoa.net/twiki/bin/view/IVOA/IvoaSS

Theory IG

Theory IG
https://wiki.ivoa.net/twiki/bin/view/IVOA/IvoaTheory

Time Domain IG

Time Domain IG
https://wiki.ivoa.net/twiki/bin/view/IVOA/IvoaVOEvent

Other Groups and Committees

Eventually there are formal and organisational groups where decisions are prepared and formalized.

Exec

IVOA Executive
https://wiki.ivoa.net/twiki/bin/view/IVOA/IvoaRepMin

TCG

IVOA Technical Coordination Group
https://wiki.ivoa.net/twiki/bin/view/IVOA/IvoaTCG

CSP

IVOA Standing Committee on Science Priorities
https://wiki.ivoa.net/twiki/bin/view/IVOA/IvoaSciencePriorities

SCSP

Standing Committee on Standards and Processes
https://wiki.ivoa.net/twiki/bin/view/IVOA/IvoaSciencePriorities

7 Standards

Datalink

ADQL

Obscore

SCS

SAMP

TAP

UCD

VOTable

8 References

Demleitner, M. ADQL-Course

Demleitner, M., Becker, S. PyVO Documentation

Fernique, P. Aladin Documentation

Taylor, M. TOPCAT Documentation

9 Glossary

ADQL : (Astronomical Data Query Language) is the language used by the International Virtual Observatory Alliance (IVOA) to represent astronomy queries posted to VO services. ADQL is based on the Structured Query Language (SQL), especially on SQL 92. The VO has a number of tabular data sets and many of them are stored in relational databases, making SQL a convenient access means. A subset of the SQL grammar has been extended to support queries that are specific to astronomy.

agent : Software that acts or works on behalf of a user.

AJAX : (Asynchronous Javascript + XML) A framework for adding dynamic interactions within web pages.

Aladin : An interactive tool that allows the user to visualize digitized images and catalogs from many sources.

ant : "A Java -based software build tool, similar to Unix ""make"". "

API : (Application Programming Interface) The documentation of the interface to a software library or tool.

applet : A small program that runs in a larger client context, often Java programs embedded in Web pages.

architecture : The overarching design of a computer, network, or software system.

array : A data structure for software elements where each element has a unique identifying index number.

ASCII : (American Standard Code for Information Interchange) Formally, an encoding of common alphanumeric symbols. Often used to mean a human readable representation with no 'special' characters or formatting.

ASP : (Active Server Pages) A technology that enables dynamic web pages using server -side scripting.

asynchronous services : Web services where there may be a delay between the request for a resource (or service) and the response where the client is not expected to wait for the server.

attributes : 1. In XML, characteristics of an element that are specified within the '¡', '¿', brackets after the name of the element, e.g. . 2. In database management systems, the term attribute is sometimes used as a synonym for field (i.e. column).

bindings : Associations between defined web interfaces and the services that provide them.

authentication : In computer security, verification of the identity of a user and/or the user's eligibility to access a service.

C# : An object-oriented programming language from Microsoft that is based on C++ with elements from Visual Basic and Java.

C++ : An object-oriented version of the C programming language.

callbacks : Routines that are automatically invoked when some event occurs or situation is encountered.

Carnivore : Open source registry developed at Caltech, supporting publishing, searching, and harvesting. Primarily intended for use by data providers who want to set up their own registry.

certificate : An electronic document that verifies the owner of a public key, issued by a certificate authority.

CGI : (Common Gateway Interface) A protocol that defines how data is passed to server applications using HTTP.

client : A computer program or terminal that requests information or services from another computer (a server) on the network.

code stubs : Code, usually generated by software tools, which defines the interfaces to some component but typically does not include any implementation of its functionality.

cone search : A VO protocol that requests information near a specified location in the sky.

container : An element that acts as a parent and contains child elements.

CORBA : (Common Object Request Broker Architecture) An open, vendor-independent architecture and infrastructure that computer applications use to work together over networks.

crossmatch : Find objects from two or more datasets that are near each other in the sky.

cyberinfrastructure : A research environment in which advanced computational services are available to researchers through high-performance networks.

daemon : A program or process that runs in the background unattended and may be invoked by another process to perform its function.

DAL : (Data Access Layer) The VO protocols that define how VO applications access data resources.

data model : A formal description of how data may be structured and used.

database management system : A collection of programs that enables storage, modification, and information extraction from a database. Also see RDBMS.

DataScope : A web-based VO tool that finds information from many VO sources near a specified point in the sky.

distributed database : A database where the underlying data is stored on multiple servers.

distributed computing : Spreading the workload for processing tasks over multiple machines.

DCOM : (Distributed Common Object Model) A protocol that allows communication and manipulation of objects over a network connection.

DHCP : (Dynamic Host Configuration Protocol) A protocol that automatically manages IP addresses for a set of nodes in a network.

DOM : (Document Object Model) A W3C standard in which a structured document such as an XML file is viewed as a tree of elements.

.NET : Microsoft's language-independent application development platform for creating web applications and web services.

element : 1. A single item in an array. 2. In XML, a node in document. Each element starts with a and ends with .

federation : The dynamic combination of information from separate sources of information.

FITS : (Flexible Image Transport System) The IAU-approved standard format for astronomical data.

FTP : (File Transfer Protocol) A protocol used to exchange files over the internet.

footprint : The region of the sky that has been observed by one or more telescopes.

GPL : (GNU Public License) A software license which allows for redistribution but requires both original and modified source code to be made available.

Grid : Massive distributed computing capabilities currently available on the Internet.

grid computing : Applying the resources of many computers in a network to a single problem at the same time.

GUI : (Graphical User Interface) A graphics-based user interface that incorporates movable windows, icons and a mouse.

GWS : (Grid and Web Services)

HPC : (High Performance Computing) Typically refers to supercomputers used in scientific research.

HTML : (Hypertext Markup Language) A standard document format used on most web pages which makes it easy for one document to refer to another.

HTTP : Communications protocol used to access most web pages.

HTTPS : A secure version of the HTTP protocol.

HTTP GET : An HTTP request where any parameters of the request are included in the URL itself.

HTTP POST : An HTTP request where data is sent to the server.

IDL : (Interactive Data Language) A popular data analysis programming language used by scientists.

instance document : An XML document that conforms to a schema.

interface : A structured interaction between two entities, often a client and a server.

interoperability : The ability of software and/or hardware on different machines from different vendors or sources to share data or collaborate without special effort on the part of the user.

interpreted language : A programming language that is parsed and executed without needing any explicit compilation.

IRAF : (Image Reduction and Analysis Facility) A software system for astronomical data analysis including both tools, libraries and languages developed primarily at the National Optica Astronomy Observatories (NOAO).

IVOA : (International Virtual Observatory Alliance) An international collaboration formed in June 2002 to coordinate Virtual Observatory activities worldwide.

IVORN, IVOA identifier : A standardized, unique ID used in a number of contexts within the VO.

Java : An object-oriented, platform-independent programming language, developed by Sun, and modeled after C++.

JDBC : (Java Database Connectivity) The standard Java interface for access to SQL -based DBMS 's.

KDIG : (Knowledge Discovery Interst Group)

markup language : A language for annotating a document to enable each component to be appropriately formatted, displayed, or used.

metadata : Information or labels that describe data.

middleware : An intermediate level of computer software typically used to provide a common interface to heterogeneous lower level components.

MIME : (Multipurpose Internet Mail Extensions) The most common protocol for encoding, transmitting, and decoding non-text files via e-mail.

Mirage : VO-enabled tool for exploratory analysis and visualization of images and multi-dimensional numerical data.

mosaic, mosaicking : A virtual observation made by combining multiple observations at different positions resulting in a combination of images that abut and/or overlap into a single larger image.

MySQL : An open source SQL relational database management system (RDBMS).

name resolver : A service that translates object names into astronomical coordinates.

namespace : 1. The set of names in a naming system. 2. In XML, a collection of names, identified by a URI reference, that are used in XML documents as element types and attribute names.

NED : (NASA Extragalactic Database) An astronomical resource containing extensive information on extragalactic objects. NED also provides a name resolver.

NESSSI : (NVO Extensible Scalable Secure Service Infrastructure) A VO web service that runs secure, asynchronous services on the Grid.

OAI : (Open Archive Initiative) A protocol that defines an interface for the sharing of metadata.

OASIS : (On-line Archive Science Information Services) A VO Tool to access and display astronomical image and catalog data.

object-oriented : Programming based on the concept of an "object", a data structure associated with specific routines that define the behavior of the object.

ontology : A formal description of the vocabulary used in a field, especially describing the relationships between various concepts within that subject.

open source : Software, usually free, created by a development community where the source code is distributed as well as compiled code.

OpenSkyQuery : A VO Service that enables crossmatching of astronomical catalogs and selection of catalog subsets.

operating system : The program framework in which all other programs run on a computer, e.g. Windows XP, MacOS? X, Linux, etc.

parameters : Inputs to or elements in a system which can be varied.

parse : To separate into more easily understood parts.

peer-to-peer : Communication between two or more computers where the protocol is symmetric between the participants so that each participant can make the same requests or give the same responses. This contrasts with client-server protocols

Perl : A programming language frequently used for web scripts and to process data passed via HTML forms.

PHP : (PHP [personal home page] Hypertext Preprocessor) A scripting language used to create dynamic Web pages.

PLASTIC : (PLatform for AStronomical Tool InterConnection?) A protocol that allows collaboration between multiple processes running on the user's desktop.

platform : The combination of a computer's operating system software and hardware.

portal : A web site that serves as a starting point to other destinations or services on the web.

protocol : A set of rules that define the interactions between two or more components.

proxy : A piece of software that acts on behalf of a user or another piece of software. An agent is a client proxy.

proxy certificate : A certificate that is used in the place of another, typically with a limited lifetime.

query : To interrogate a collection of data such as records in a database.

RDBMS : (Relational DataBase? Management System) A DBMS that uses represents data using a relational database.

RDF : (Resource Description Framework) A recommendation from the W3C for creating metadata structures that define data on the web.

registry : The "yellow pages" for the VO. Collects and stores basic information about archives, data collections, databases, and other resources.

relational database : A database that stores data in a structure consisting of one or more tables (aka relations) of rows and columns, which may be interconnected.

resource metadata : The metadata that describes services and data resources available in the VO.

REST : (Representational State Transfer) An approach to web services that uses the standard HTTP GET and POST protocols.

RMI : (Remote Method Invocation) A Java protocol for distributed computing.

ROME : (Request-Object Management Environment) A VO tool to manage the execution of a task that requires many subtasks.

RPC : (Remote Procedure Call) Protocols for distributed computing where the interaction is represented as the client computer invoking discrete services/calls on the server.

RSS : (Rich Site Summary). An XML format for sharing content among different Web sites such as news items.

Ruby : An object-oriented programming language.

SAX : (Simple API for XML) A standardized interface for parsing XML documents using callbacks.

schema : 1. A description of the structure and rules an XML document must satisfy. 2. In SQL, a description of the tables and columns in the database.

script : A simple program usually written in an interpreted language.

semantics : The expression of the meaning of symbols or names. In the VO, the actual scientific meaning of data and services.

serialization : The process of converting an object into a format that can be stored or transmitted across a network.

server : A computer system in a network whose services may be invoked by one or more clients.

servlet : A program, typically Java, that runs on a web server in response to a web request.

SESAME : A web service interface to the SIMBAD name resolver.

sexagesimal : Numeral system with number 60 as the base.

SExtractor : (Source Extractor) A tool that detects sources in astronomical images.

SIA, SIAP : (Simple Image Access Protocol) A VO protocol thatsupports queries for images available in a given data collection near a given position on the sky.

SIMBAD : (Set of Identifications, Measurements and Bibliography for Astronomical Data) An astronomical database provides extensive information on both galactic and extragalactic objects. SIMBAD also provides a name resolver.

Simple Spectral Access : See SSAP.

SkyNode : A VO protocol (and the services that implement it) that provides an ADQL interface to astronomical databases.

SkyPortal : A web site that supports translation of a user ADQL query into queries of one or more SkyNodes.

SkyServer : Web service that presents data from the Sloan Digital Sky Survey.

SkyView : Web site and VO-enabled distributable tool that generates images from survey data.

SMTP : (Simple Mail Transfer Protocol) A protocol used to send and receive email.

SOA : (Service Oriented Architecture) An approach to distributed computing that focuses on services that communicate with each other.

SOAP : (Simple Object Access Protocol) A protocol for invoking remote services by exchanging XML -based messages.

socket : The low-level software element that makes a connection to the network. Normally a client connects to a socket on a server.

source code : The version of a program normally written or edited by a programmer and either compiled into an executable program, or run directly using an interpreter (see interpreted languag).

SQL : (Structured Query Language) The standard language used to communicate with RDBMS s

SQL Server : Microsoft's RDBMS software.

SRB : (Storage Resource Broker) Middleware developed at the San Diego Supercomputing Center that provides standardized access to a number of very large data resources.

SSAP : (Simple Spectral Access Protocol) A protocol that returns a set of spectra in a specified region of the sky. Similar to SIA but has many more options.

SSDL : (SOAP Service Description Language) SSDL is a SOAP -centric description language for web services that enables protocol -based integration.

SSL : (Secure Sockets Layer) A protocol for managing the security of a message transmission over the Internet.

standalone application : A computer program capable of operating without external resources.

STC : (Space-Time Coordinates) An IVOA standard for describing a region or position in both space and time.

STILTS : (STIL Tool Set) A set of VO tools for processing of tabular data based on the UK Starlink Tables Infrastructure.

TLA : (Three Letter Acronym) A tribute to the use of acronyms in the computer field.

token : An item in a string of text that can be separated out by a parser, such as a single word in a sentence or a number in a comma-delimited list.

TOMCAT : An HTTP server that can run Java servlets.

TOPCAT : (Tool for OPerations on Catalogs And Tables) An interactive graphical viewer and editor for tabular data, designed for but not limited to astronomical tables.

Treeview : A VO-enabled viewer for hierarchical file structures.

UCD : (Unified Content Descriptors) A formal vocabulary for astronomical data that is controlled by the IVOA.

URI : (Uniform Resource Identifier) An address standard for a resource available on the Internet.

URL : (Uniform Resource Locator) The global address of documents and other resources on the World Wide Web. The first part of the address specifies the protocol to be used when accessing the resource, the remainder describes its network location.

validator : A tool that checks some element of a system for conformance to a standard.

virtual data : Data product that is dynamically generated when needed.

VOClient : A software suite callable from many languages which implements data access in the VO.

VOEvent : A VO standard for representing, transmitting, publishing and archiving the discovery of a transient celestial event.

VOEventNet : A peer-to-peer cyberinfrastructure to enable rapid and federated observations of the dynamic night sky.

VOPlot : A VO Tool for visualizing astronomical data from VOTable sources.

VOSpace : A distributed storage concept for the VO.

VOStat : A VO and web-enabled statistics package.

VOTable : An XML -based encoding scheme for astronomical tables and catalogs, established by the IVOA in order to provide an unambiguous way to transmit tables between computer programs

W3C : (World Wide Web Consortium) An international consortium where member organizations, a full-time staff, and the public work together to develop web standards.

WCS : (World Coordinate System) A detailed specification of the conversion between coordinates within a file and physical coordinates, especially between pixel and celestial coordinates in an image.

WCS Fixer : A VO web service that corrects the WCS information in a given FITS image.

web service : Software available over the web using a standardized XML messaging system.

WESIX : A VO web service to the standard astronomical image analysis package SExtractor together with a crossmatching service.

wget : Free software package for retrieving files using HTTP, HTTPS and FTP.

workflow : A sequence or network of tasks and associated information needed to pass from one task to another to accomplish some goal.

WS : see web service

WSDL : (Web Services Description Language) An XML document that describes and locates a web service.

XMatch : see crossmatch

XML : (eXtensible Markup Language) A markup language that provides a file format for representing data.

XML-RPC : (XML-Remote Procedure Call) A web service protocol that utilizes XML technology to implement an RPC protocol.

XOP : (XML Binary Optimized Packaging) A standard specifying how binary data should be represented in XML.

XPath : (XML Path Language) a language that describes how to locate specific content within an XML document.

XQuery : A standard language for querying XML data.

XSL : (eXtensible Stylesheet Language) A standard for describing how to transform XML documents into other documents in either XML or other formats.

XSLT : (eXtensible Stylesheet Language Transformations) A conversion tool that implements XSL.

YAML : (YAML Ain't Markup Language) A data serialization language based on XML and other languages.