

The AstroGrid MySpace System

Abstract

AstroGrid is a UK e-Science initiative to develop a complete Virtual Observatory infrastructure. It will provide seamless remote access to archives and databases. AstroGrid is the UK contribution to the global effort to develop a Virtual Observatory and it participates in the International Virtual Observatory Alliance (IVOA) initiative. In particular it is developing and demonstrating prototype infrastructure of the sort which will be required by the Virtual Observatory.

Much astronomy research involves accessing astronomical archives to perform database searches that yield files of results. The astronomer may wish to download such results to his own computer for further analysis, or to perform further queries on the results, either in isolation or in combination with searches of other archives. In any event, work space is required for both temporary and long-term storage of such data and software is needed for managing and accessing them. MySpace is AstroGrid's system for this purpose. Implemented as a distributed network of co-operating web services, the novel feature of MySpace is that it provides the astronomer with seamless access to a geographically dispersed work space. Typically storage space is made available close to the various archives, but MySpace hides all the details of accessing these remote systems. In addition to storing a user's results, MySpace is used by various components of the AstroGrid system as a convenient mechanism for storing temporary or intermediate results.

Background

MySpace is a discrete component within the AstroGrid architecture. MySpace is fully compliant with the emerging IVOA interface standards, making it an ideal plug-and-play component for use in other VO projects.

As can be seen from the layer diagram (*Fig 1* below), MySpace interacts with many of the other AstroGrid components and as such is a key element to our delivery strategy.

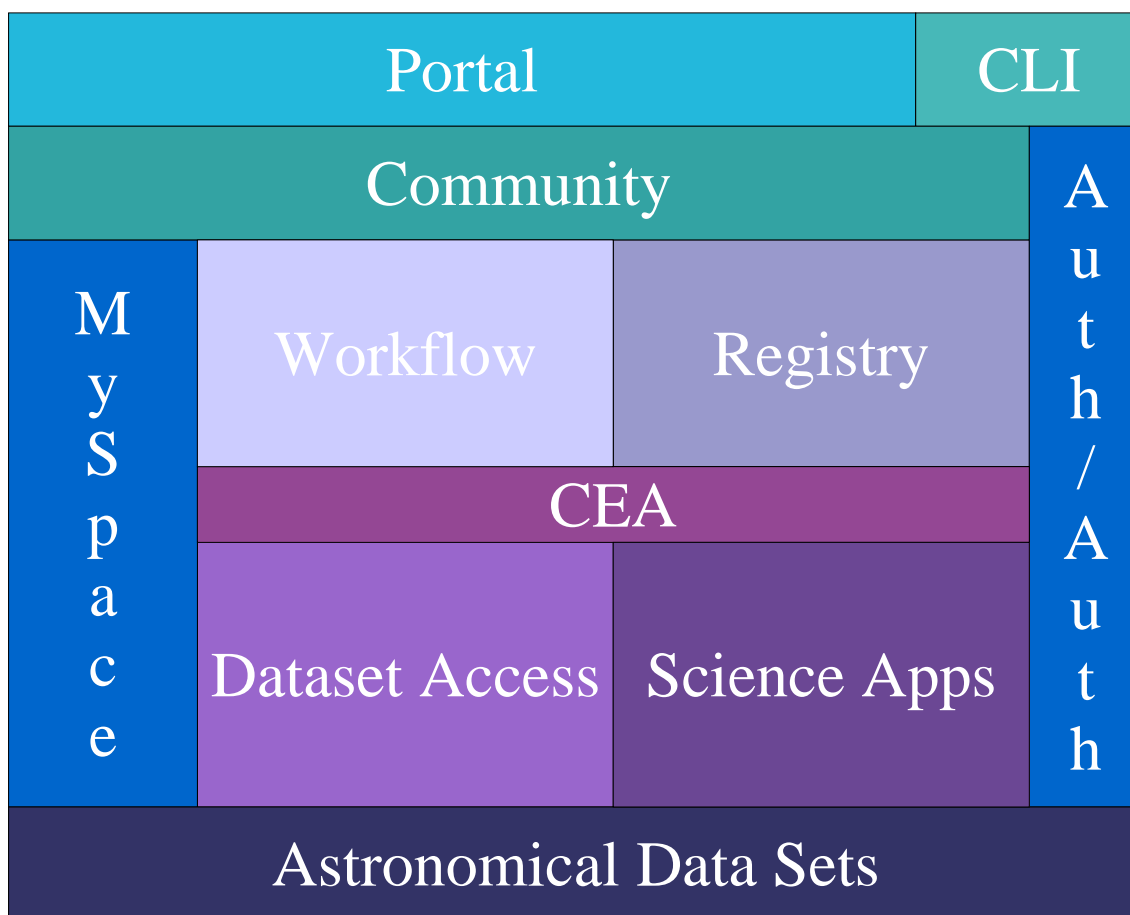


Fig 1

AstroGrid is implemented as a collaborating peer network of Web Services. As such there is no central management burden and the network can grow according to need without extensive administrative effort. To keep client and application writing simple, AstroGrid provide “delegate libraries” (currently only implemented in Java) which contain all the necessary service access and management code. Thus client and application developers can concentrate on the problem being solved rather than having to wrestle with a the technology that is AstroGrid.

Design

AstroGrid software is currently being deployed at data centres and astronomical research institutions providing access to data archives across the UK. This deployment includes several MySpace systems, providing work space for a variety of different purposes. There are two

contrasting roles which are likely to be particularly important: 'Cache systems' and 'Community systems'.

Cache systems:

large caches close to the data archives and for the storage of large, transient files generated from queries submitted to the archives,

Community systems:

longer-term, personal work space where astronomers can keep results and 'work-in-progress'.

It is important to state that role notwithstanding, the actual implementation of MySpace remains the same and it is perfectly possible, indeed probable, that any given deployment will function in both roles. The differentiation will be manifest in the Authorisations (see below) set against the various data storage areas within MySpace. Finally, all components can optionally be geographically dispersed from each other and from other components of the AstroGrid system.

Components

There are three major elements to the design of MySpace: the MySpace Manager, MySpace DataServer and MySpace Client. These are shown in the diagram below:

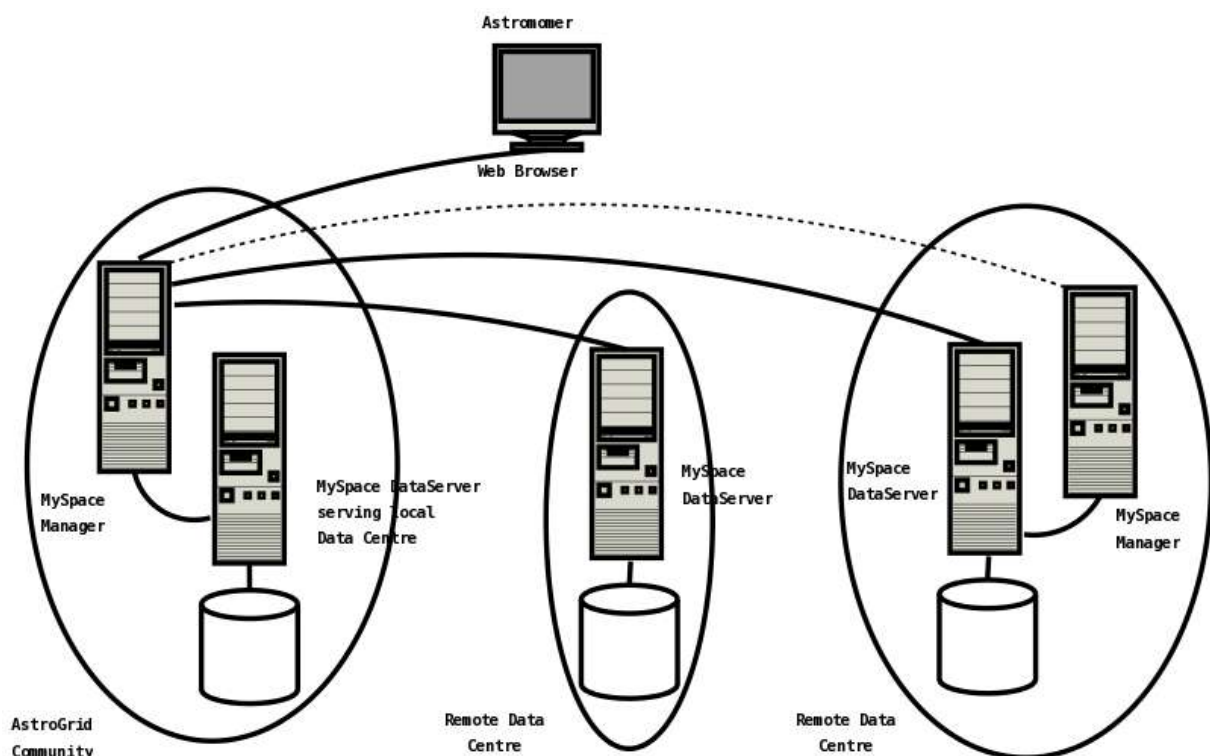


Fig 2

MySpace Manager

Embodies the intelligence of the MySpace system and is invoked by external components to access or manipulate files in MySpace. It manages and maintains the metadata describing the actual data assets, providing structure and access capabilities. It is here that storage transparency is implemented: MySpace is storage-agnostic and as

such the user need and should not be aware of where of how their data are being held.

MySpace DataServer

A repository where data are kept. It is invoked by the Manager to copy, delete *etc.* a specified file and has little local intelligence.

MySpaceClient

An interface between the user and the MySpace system, typically interacting with the MySpace Manager and transparently (to the user and client writer) with the MySpace DataServer. The current reference implementation of AstroGrid supplies a web browser based client (called the MySpace Explorer), but could equally well be a client-side application.

Although the three components of MySpace (Manager, DataServer and Client) are intended to work together, they can be installed separately. They communicate remotely and do not need to be co-located. This arrangement allows, for example, a Manager to control Servers at remote sites. MySpace interacts with other AstroGrid components to ensure that access is secure and controlled. Nonetheless, a MySpace system can, in principle, be deployed in isolation or with external Virtual Observatory components or indeed perhaps with non-Virtual Observatory web service based grid project components. MySpace is primarily intended to provide work space and thus an optional mechanism is provided to retire old data after a configurable 'expiry period' has elapsed.

Functionality

The principle purpose of the MySpace system is to provide data storage facilities to astronomers in as implementation-transparent manner as possible. It is a design goal that the astronomer neither knows nor cares where or how their data are being held. This implies a great deal of trust in the system and that trust will have to be earned through experience of using a reliable and robust system. What follows is a description of how we believe that trust can be justified.

Data storage and sharing

As has been mentioned, MySpace is data store agnostic: whatever storage mechanism is appropriate should be used. AstroGrid are providing a MySpace DataServer component which fulfils the role of data file storage. Later it is anticipated that this will be expanded to include data stored in relational databases and other storage mechanisms. However, there is no reason why an FTP or other data storage server might not be substituted. It is anticipated that regular and Grid FTP servers will be used as well as established data storage tools such as SRB or MyDB. This latter is a Johns Hopkins University project providing virtual database storage functionality. We are working with the MyDB project and others to propose an IVOA interface standard (VOSpace) for data storage components. Indeed, with support for this VOSpace interface standard, real plug-and-play substitution should be possible.

For this to work, the details of the data (i.e. The metadata describing these data) are held in the Manager component. It is here that the arbitrary containment hierarchy is held and the data access defined. The actual location of these data is determined by the MySpace and wider AstroGrid system, although the astronomer can specify a location if required. Generally speaking, data will be

held in the MySpace serving the astronomers AstroGrid Community unless there is good reason to move it elsewhere. One such reason is in support of the goal of minimising data traffic over the network. e.g. If an astronomer based in Leicester queries a data archive in the Royal Observatory in Edinburgh (ROE) before having the results further processed Jodrell Bank Observatory (JBO), there is no point transferring the results of the first step to Leicester only to have them moved to JBO by the very next step. AstroGrid and MySpace supports the notion of data routing to minimise this traffic. In the example give, the results from the ROE query would be moved directly to JBO and only when processing was complete would the data be finally transferred back to Leicester.

Furthermore, data need not be static; it is perfectly acceptable for a DataServer to be something generating live data (e.g. a monitor of some sort). As long as the data it provides is accurately described in the Manager metadata, it is accessible via the MySpace system. This makes it possible to access data simply without requiring an extra “client” or “agent” presentation layer as would be necessary if the monitor were just another application.

Data may be shared between users in several ways: data can be copied, symbolically linked (c.f. Unix symbolic linking) or published. Copying involves multiple instances of the data being created. Linking data assets allows users to share access to a single instance of the data. This is done at the metadata layer (i.e. Managers contain metadata which “points to” metadata in other Managers) and allows individual manipulation of the respective containment hierarchies.

Security

Data held within MySpace must be secure. This means providing the capability to define the access rights users and administrators wish to support against data (both individual data and collections of data) and furthermore ensuring anyone trying to access these data are in fact who they claim to be. Finally, once accessed, it should be possible to determine who performed the access and what operations were undertaken. Together these three elements form the so-called security Triple-A: Authentication, Authorisation, Accounting.

Implementing security is always a balance between preventing unauthorised access to data and usability. Ultimately it comes down to a business decision: how valuable are your assets and how intrusive a security system are your users prepared to tolerate? Many astronomical data assets are public domain. As such, the main requirement is to prevent malicious damage rather than theft. However, private data also have to be protected. The former is achieved by only allowing restricted access (typically read only with strict storage and processing quotas) to unauthorised or unknown users. Authorised users on the other hand can be allowed much freer access.

Authentication

The first step in this process therefore is to determine whether a user is actually who they claim to be: Authentication. Several approaches have been considered and discarded for a variety of reasons – mostly down to ease of use considerations. The solution finally adopted is based upon single sign-on to a “trust network” of co-operating services. This means that once authenticated by one part of the network, a user is considered to be “trustworthy” by the other parts. This is similar to the Shibboleth approach, an effort we are monitoring closely and under active consideration by JISC for the academic community as a whole. Clearly this is not as secure as a certificate based system, but it is much easier to administer and until such time as personal certificates become wise-spread will remain the primary source of authentication information. The way this works in AstroGrid can be seen from the following diagram:

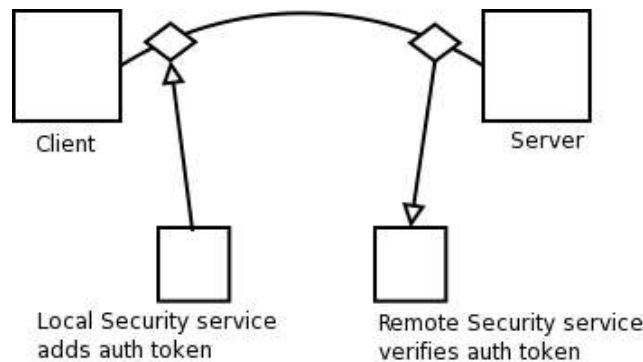
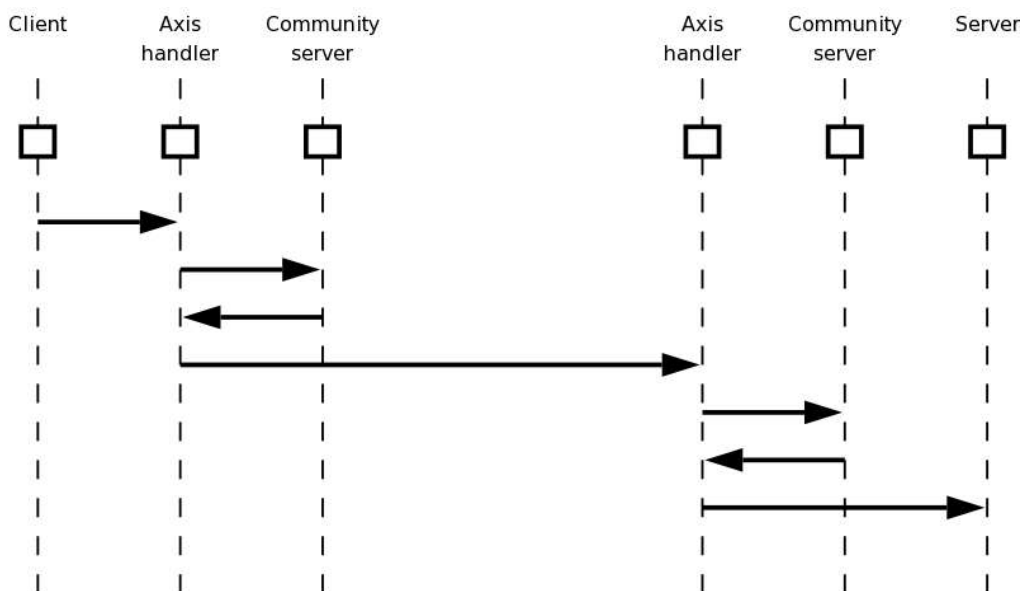


Fig 3

By using Apache Axis “handlers”, delegates are able to insert and check secure tokens verifying the source of any given SOAP packet, without intervention on the part of either client or server developer. This does rely upon a co-operating Community Service to work with any level of granularity, but is not totally reliant upon such a service. In the absence of a Community Server at either end of the call, the delegate will insert the “anonymous” user profile and the receiving service can then provide the default access as configured for unknown users. The sequence diagram below shows the operations:



Authorisation

Once a service has established the user is who they claim to be, the next step is to verify that they are entitled to perform the requested operation. Again, this is undertaken by the Community service. A server registers the operations it wishes to support together with Access Control Lists which specify who is allowed to do what (user, group, community, world etc). Upon receipt of an authenticated SOAP message, the server simply queries the Community service with token and operation and

receives an authorisation message in return. The service then processes accordingly. As with Authentication, the Authorisation will default back to configuration defaults if no Community service is available.

Accounting

As part of the Authorisation process, accounting information will be recorded. This is currently under consideration and not yet implemented.

Robustness

For a data store to be of real use, it must be reliable. This means that the data it holds must be accurate, readily available (subject to security considerations) and backed up. As with security, the actual level of service is ultimately a business decision: how valuable is the data asset? What is the impact of non-availability? What downtime in the server can be tolerated? What price is the business prepared to pay for these services? Typically astronomical data is written infrequently and read often. Downtime in the order of 10s of minutes is probably acceptable, especially when a job can resume in the event of interruption. Of greater importance is data provenance. In order to provide insights into some of these questions, AstroGrid is tracking efforts by the Digital Curation Centre (<http://www.dcc.ac.uk>).

Thus AstroGrid approaches these considerations as follows:

Fault tolerance

All AstroGrid components are designed to manage faults gracefully and recover from error conditions back to a known state. In addition, it is planned to include integrity checking capabilities for data transfers and the like. At present, data transfers are as reliable as the underlying mechanisms (e.g. FTP, SOAP, HTTP GridFTP etc)

Integrity

Because the underlying AstroGrid DataServers use standard data storage mechanisms (e.g. file systems, databases etc), standard backup and recovery procedures provide the data integrity guarantees required. Similarly where a high availability requirement exists, standard data centre processes (e.g. Shared RAID disk arrays etc) can be used to keep uptime at a maximum.

Fail-over

To support high availability in the event of hardware failures, similar contingencies can be adopted (shared disk arrays, mirrored data stores etc) and the AstroGrid registry can be used to define alternative DataServers in the event of unavailability. It is planned that the service access delegates will transparently manage these events.

Conclusion

MySpace is a vital, integral component within AstroGrid. It provides light-weight yet secure distributed data asset management and storage services and is compliant with the IVOA standards. In addition, being storage agnostic, it can inter-operate with other data stores and can benefit from their added functionality and maturity.

Due to the modular nature of AstroGrid and its conformance to the IVOA interface standards, the functionality delivered by MySpace could easily be replaced by an alternative, compliant system. Of course the converse is true and its very modularity recommends it for use as a plug-and-play component in other VO projects.