

APOGEO: an automatic management system for astronomical portals

S. Zorba, A. Bignamini, F. Cepparo, C. Knapic, M. Molinaro, and R. Smareglia
INAF - Astronomical Observatory of Trieste

Introduction

Since several years IA2 (Italian Center for Astronomical Archive) develops and hosts web portals for accessing various astronomical archives. The experience gained over the years allowed to recognize recurrent architectural patterns and common requirements of all these interfaces, leading to develop APOGEO (Automatic PORTal GENERatOr). APOGEO consists in a set of Java EE tools in which we tried using VO standards as much as possible. These tools allow to generate portals in a standardized and reliable way, after a portal administrator has performed an aided configuration process.

Within a few weeks an interface generated by APOGEO will provide access to data of Asiago Astronomical Observatory (see **poster P4.8**).

Portal generation

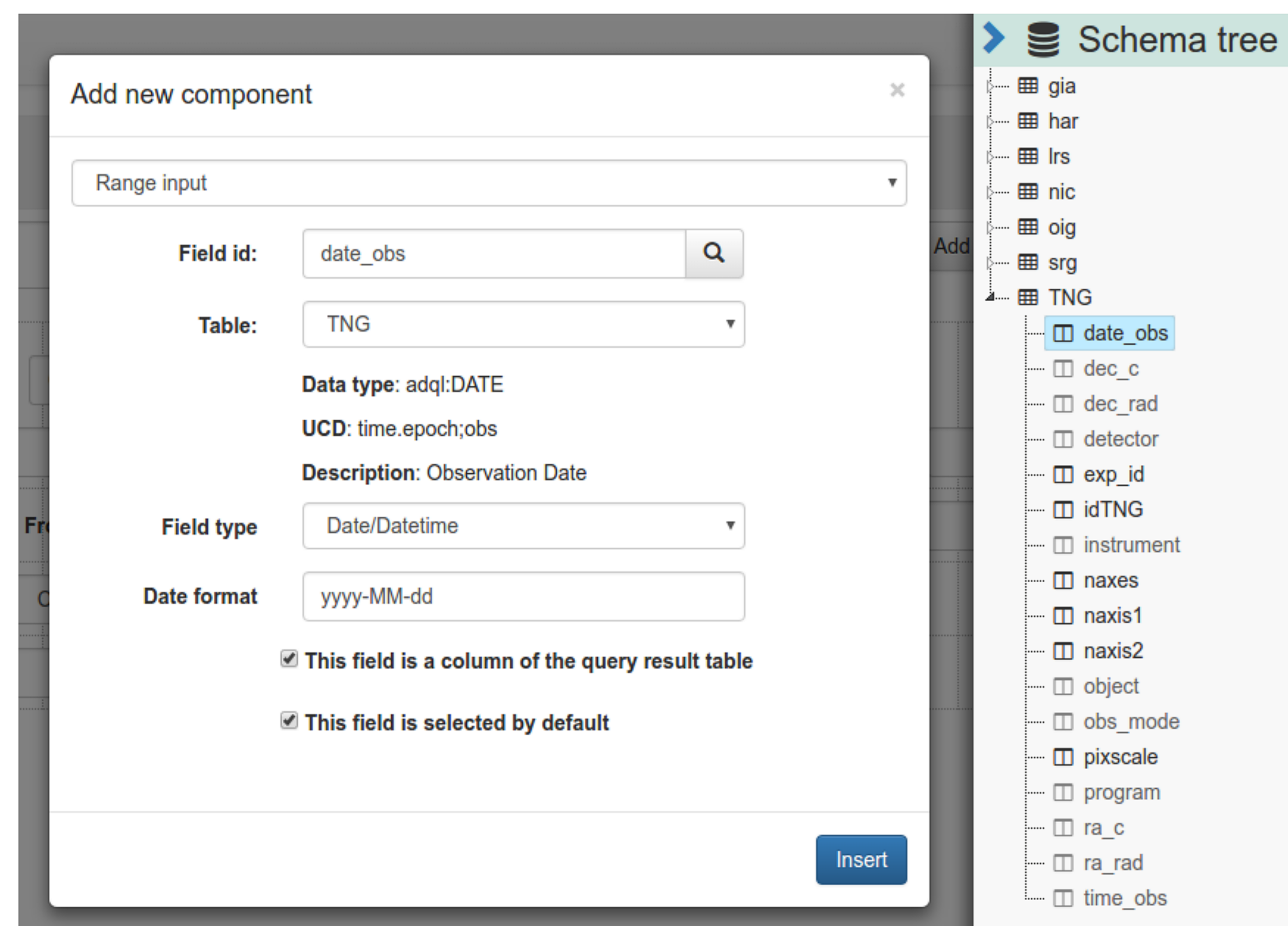
Portal set up process begins creating a TAP_SCHEMA, a particular schema defined into the IVOA standards to specify schemas metadata.

When the TAP_SCHEMA is defined, the portal administrator can configure the portal using a web application called PortalGenerator.

Currently PortalGenerator has a wizard-like structure, with a sequence of configuration steps that have to be followed in order. However, over the time we are introducing various extra steps, to meet the requirements of different kind of portals, and most of these steps are optional. So, a dashboard-like structure, in which the user can select the component he or she wants to configure, could be more appropriate.

A mandatory step is the selection of the tables that will be used for building the queries. This implies some constraints on the allowed database structures: there must be a main table and a set of secondary tables (usually one for each telescope instrument) that have to be joined on the main table in a simple way. If the astronomical archive doesn't follow this structure it is often possible to create some views to bring the database to appear as desired.

The other fundamental step consists in the search form design. The PortalGenerator provides a set of interactive grids in which the administrator can insert various components such text inputs and labels, positioning them using the mouse. Each HTML input is directly mapped to a database column. In this way the generated portal will be able to build SQL queries based on which HTML fields are visible and selected.



Inserting a datetime range input component during the TNG archive generation

Each portal needs to have a main grid, but other secondary grids can be added. The secondary grids will be placed into a tabbed panel that can be optionally activated, allowing an advanced search mode. This has been used for implementing searching on single telescope instruments, but a grid isn't bind to a specific table, so other uses of this feature are possible.

Internally the PortalGenerator stores the portal configuration in a XML format. Pieces of this XML representing single grids can be edited from the PortalGenerator grid editor too, using a JavaScript library that provides code syntax highlighting and autocomplete. This allows creating a set of similar form fields simply doing copy and paste and it is useful in all situations like tables renaming or reusing the same layout of an older version of the portal. The JAXB framework is used to convert XML to Java objects.

In the final step it is possible to generate the portal war package file, that can be deployed in an application server like GlassFish or Tomcat.

APOGEO evolution

Archive files management

Each archive stores its scientific data on files (usually in FITS format) and the archive database needs a way to reference these files.

First APOGEO implementation supposes files are stored into the disk of the same machine that hosts the portal web application, because absolute file paths are written into a dedicated database column. If files are stored into another machine, they need to be mounted on the portal machine using NFS.

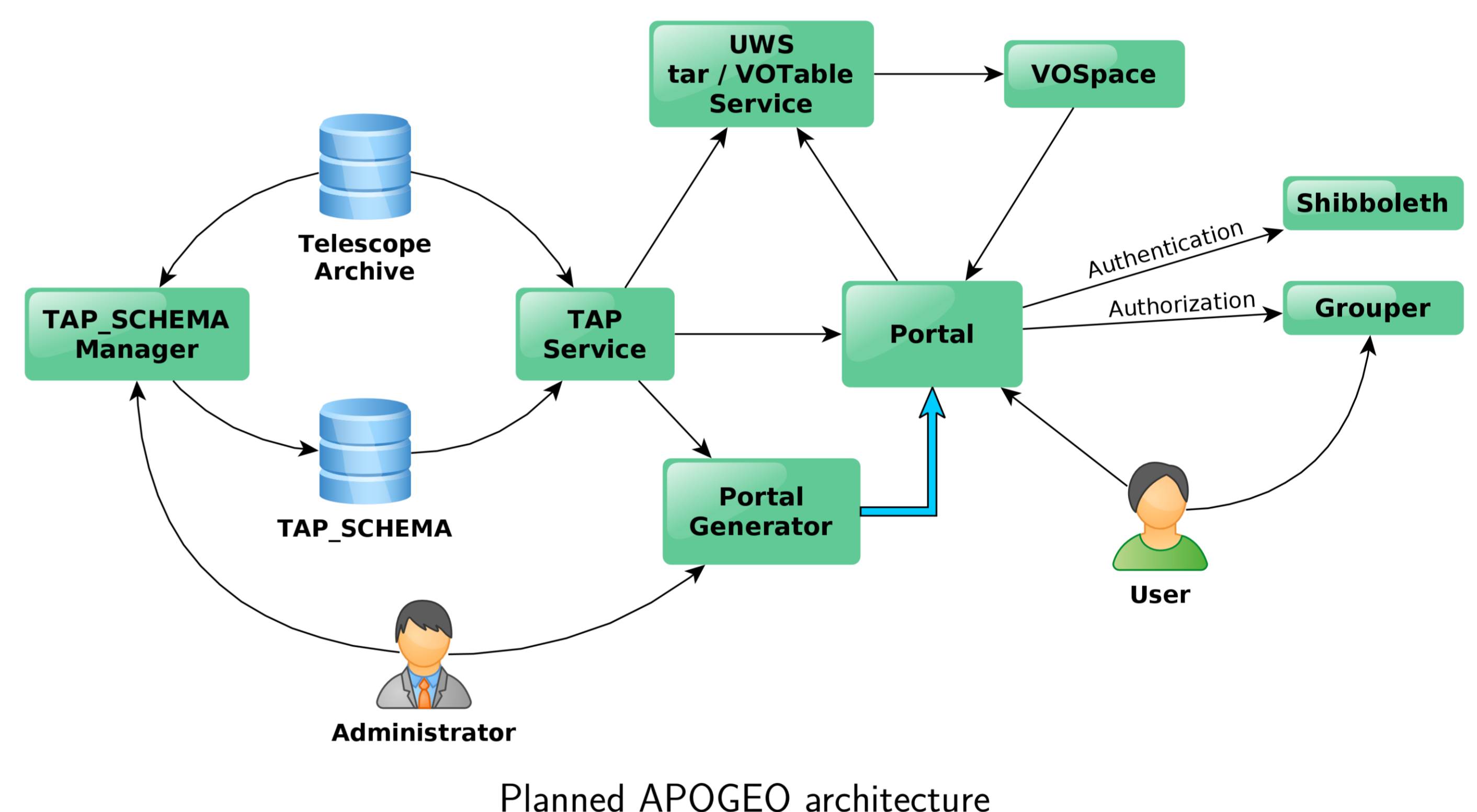
Trying to generate a portal for a didactic project that references its files using URLs of an external web site we added the possibility to choose between a "file mode" and "URL mode".

In the near future we plan to retrieve files from a VOSpace service.

Users files management

Generated portals include a feature that performs creation of tar file archives from a selection of portal files made by the user and a feature that creates VOTable files from a query executed on the portal. Both these tasks are executed asynchronously using the UWS IVOA standard and the portal implements a polling mechanism checking for their completion. Each UWS job can generate multiple files if the selection made by the user is too wide.

Users have to be able to delete these generated files but the UWS standard doesn't allow to modify a completed job, so files created by UWS service should be managed by another service after their creation. Also in this case the VOSpace could be a natural choice for our future implementations.



Planned APOGEO architecture

TAP in the core

Currently APOGEO components queries directly the astronomical archive MySQL database but, considering the system involves the TAP_SCHEMA creation, it could be possible to setup a TAP service and make queries on it.

The main aspect that has to be considered in this case is SQL queries made by portals includes a specific MySQL construct (SQL_CALC_FOUND_ROWS) to obtain paginated results and known the total results count in the same time. This can be achieved in TAP defining a custom UDF in the TAP service capabilities.

Authorization and authentication

Portals could have private data, so they need a way to manage authentication and authorization.

IA2 portals before APOGEO defined an username and a password for each observational programs. Now users are referred to people account and their permissions are defined using Grouper, an enterprise access management system provided by the Internet2 Community.

Principal Investigators (PIs) can be added to groups which names are mapped to observational programs names and they can access only programs related to the groups they belong to. Moreover PIs can login into Grouper UI and add Co-Investigators (Cols) into their groups. In this way there is no need to share credentials between PIs and Cols anymore.

Currently authentication is made calling directly an LDAP server, selected by the user from a list of possible servers. In the future we will support a federated SSO using Shibboleth.

Astronomical software/services used by APOGEO

- ▶ CADC UWS and TAP libraries
- ▶ CDS NameResolver and UCD validator services
- ▶ ARI UcdValidator
- ▶ sampjs and STIL libraries