# Towards a new web interface for LCGDM

Juan Valencia Calvellido

Universidad de Cádiz

DPM Workshop, Naples
9/10/2014

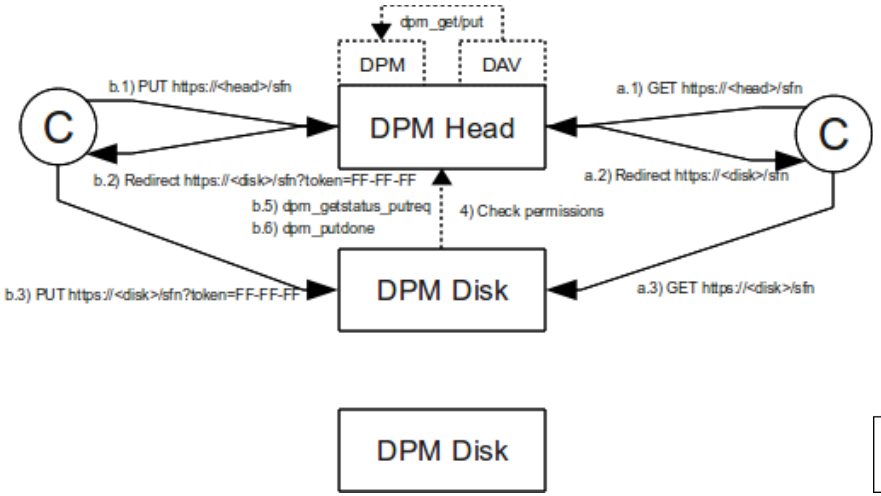
# Overview

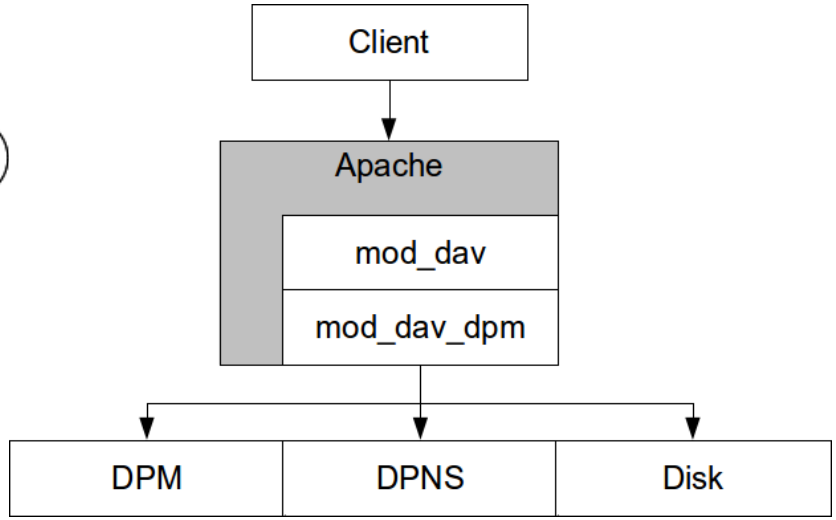
# Introduction

- I'm a Computer Science student and this is going to be my degree Thesis.
- This is an ongoing project and the goal is to provide a web interface that can be integrated with the rest of the DPM software stack, providing an easy and intuitive tool to both technical and nontechnical users to manage their data from and into the grid.

# Web access via WebDAV

- There is already web browser access for DPM via the developed HTTP/WebDAV frontend.

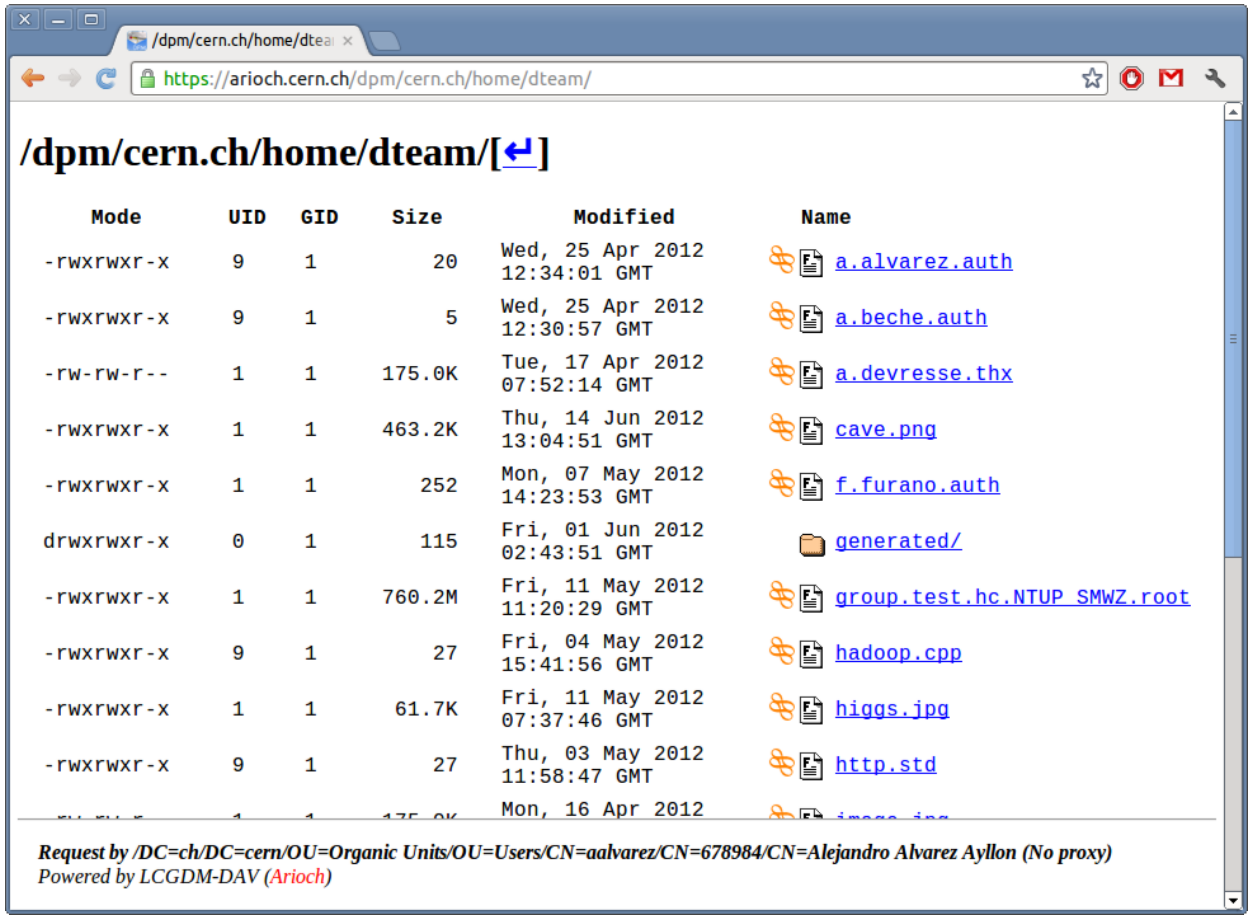



- It is built over Apache mod_dav and mod_gridsite.
- By the current visualization it is possible to download data and to navigate through a collection, but it has no interactivity.
- We are aiming to improve the actual web access by adding more functionality.

# Web access via WebDAV

Current web access:



# Outline

# Using jQuery

- Taking the present HTTP/WebDAV frontend implementation the current work is to build an API in the jQuery JavaScript library that can interact with it.
- This way it is possible to translate WebDAV standard methods like PROPFIND or MKCOL and use the information provided by them in an HTML document via jQuery.
- The API uses ajax calls to perform the HTTP/WebDAV requests and receive the data in an XHR object.

# How it works

## Example

```
$.fn.extend($,{
   Dav: function(res) {
      var api = function() {

         this.get = function(cob) {
            this.prepare(cob, 'GET');
            return this.send(cob);
         };

         this.prepare = function(cob, typ) {
            cob           = cob || {};
            cob.url       = resourceUrl;
            cob.headers   = cob.headers || {};
            cob.type      = typ || 'GET';
            cob.dataType  = cob.dataType || 'xml';
         };

         this.send = function(cob) {
            lastRequest = $.ajax(cob);
            return lastRequest;
         };
      };
      return new api;
   }
};
```

The other methods are similar to the GET method shown, though attending to particular characteristics. cob is the jQuery ajax call object.

In the prepare function the DAV call is built. Here it's needed to ensure integrity of the call object, verify the DAV method requested and set any authorization information (if necesssary).

Then the send function does the actual HTTP send through an ajax request.

# How it works

## Example

```
jQuery.Dav(url + 'testxml.xml').get({
   complete: function() {
      console.log('#get');
   },
   success:  function(dat, stat) {
      console.log(jQuery.Dav(dat).getNodesByTag('acl'));
   }
});
```

```
jQuery.Dav(url + 'test').mkcol({
   complete:  function(dat, stat) {
      console.log('#mkcol');
   },
   async: false
});
```

To use this API in a document we just call the function needed, which as we have seen is basically an ajax call.

Besides the standard methods some more functions are developed in order to operate properly. WebDAV answers with XML data so some parsers will be needed, also other methods to read a single property or navigate across the nodes received.

# DPM support

- Once we have standard WebDAV operations supported and tested then we extend the code to cover specific DPM operations.
- These include replicas management and metalink.

# Outline

# jQuery UI

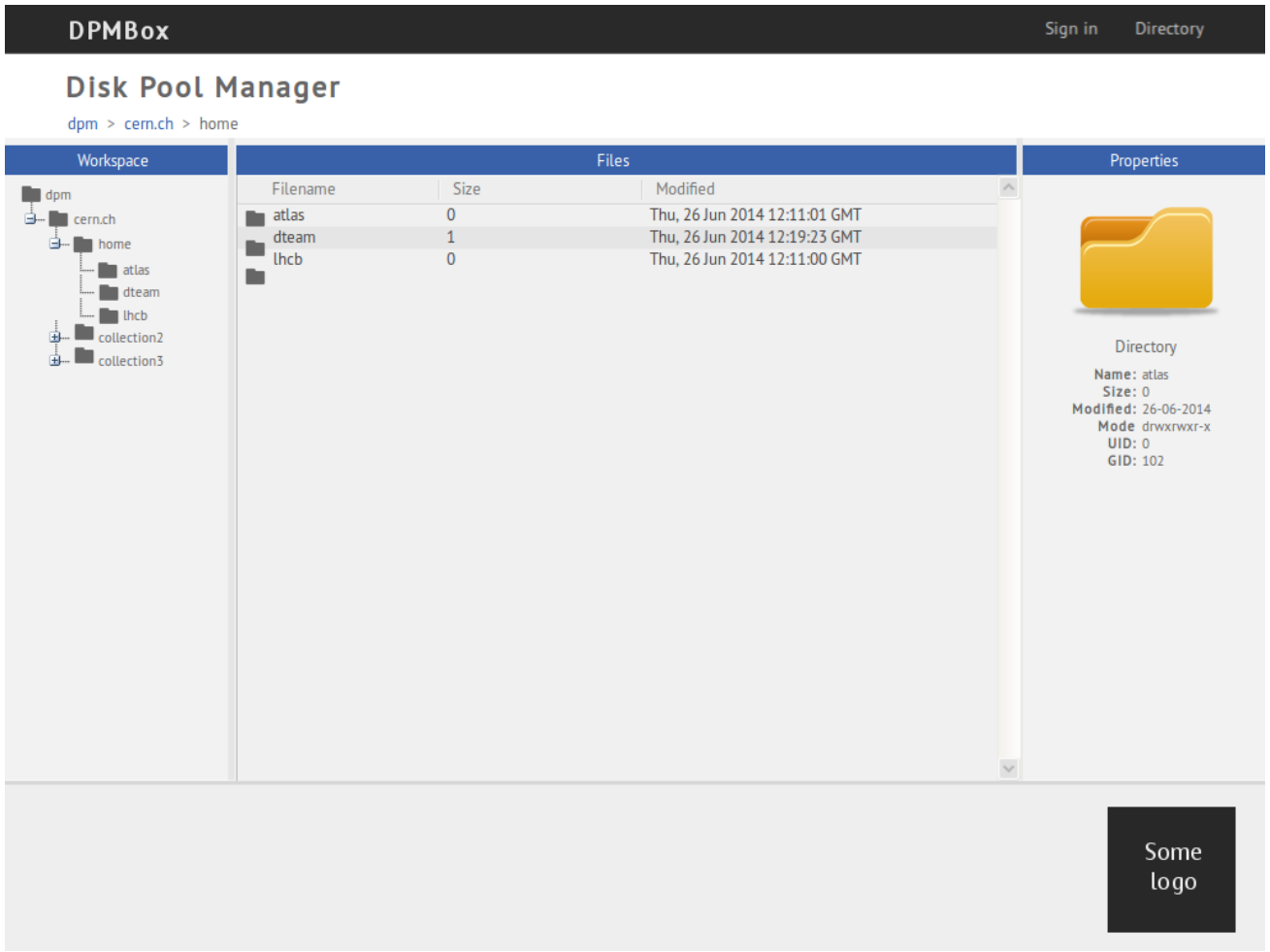
- Using jQuery UI a GUI will be designed and this interface will use the data provided by the WebDAV API previously created.
- Since we are using jQuery and jQuery UI and these are widely known libraries the interface could be very customizable and could be easily adapted for other uses or modified for extended support in the future.

**UI Setbacks:**

- Deep collections: The directory/collection tree will be loaded with lazy loading and in asynchronous way.
- Big number of data nodes inside a collection: Use pagination in case of hundreds of files when opening a directory.

# Interface layout

## Tentative design:



# jQuery UI mockup

- Basic demo of the proposed layout built using jQuery UI.
- It is not yet functional but it can show the actual appearance on a browser.

**calvellido.es/dpmdemo**

# Further Reading



Project webpage
*calvellido.es/DPMBox*


A. A. Ayllon, A. Beche, F. Furano, M. Hellmich, O. Keeble and R. B. Da Rocha
*Web enabled data management with DPM & LFC*
CERN, Geneva 1211, CH


Network Working Group
*WebDAV specification RFC 2518*
webdav.org/specs/rfc4918.html