

Consigna¹: federated filesharing

federation as an afterthought

Victoriano Giralt¹ Luis Melendez²

¹Central Computing Facility University of Malaga

²Central Computing Facility University of Cordoba

TERENA EuroCAMP
Dubrovnik
November 14th 2007

1

¹Consigna is Spanish for left luggage

Outline

1 Motivation

Outline

- 1 Motivation
- 2 Development

Outline

- 1 Motivation
- 2 Development
- 3 Demo

Outline

- 1 Motivation
- 2 Development
- 3 Demo
- 4 Implementation

Why a file sharing application?

e-mail is not the best method for exchanging files

Our users need a simple method for file exchange

Why a file sharing application?

e-mail is not the best method for exchanging files

Our users need a simple method for file exchange

- It should be useable by members and non members

Why a file sharing application?

e-mail is not the best method for exchanging files

Our users need a simple method for file exchange

- It should be useable by members and non members
- It should not be a “*black hole*”

Why a file sharing application?

e-mail is not the best method for exchanging files

Our users need a simple method for file exchange

- It should be useable by members and non members
- It should not be a “*black hole*”
- It should be easy and unobstrusive

Why a file sharing application?

e-mail is not the best method for exchanging files

Our users need a simple method for file exchange

- It should be useable by members and non members
- It should not be a “*black hole*”
- It should be easy and unobstrusive
- The browser is *THE* interface

The way to a federated application

from local users to federation

We have evolved the application through several access control methods

The way to a federated application

from local users to federation

We have evolved the application through several access control methods

- **User/password**

AuthN/AuthR at the application level

- Heavy administration
- Impedes usage by non members
- + May be used from any location

The way to a federated application

from local users to federation

We have evolved the application through several access control methods

- User/password
- **IP based**

Location based AuthR

It has the concepts of

- *Inside* (organization's IP space)
may upload **and** download
- *Outside* (rest of the Internet)
may upload **or** download

Files are password protected

DAD

The way to a federated application

from local users to federation

We have evolved the application through several access control methods

- User/password
- **IP based**

Location based AuthR

- + Low administrative burden
- + Easy for non members
- Unusable for roaming members

The way to a federated application

from local users to federation

We have evolved the application through several access control methods

- User/password
- IP based
- **WebSSO**

Centralized AuthN/AuthR

- + Maybe added to IP based control
- + roaming members location become *inside*

The way to a federated application

from local users to federation

We have evolved the application through several access control methods

- User/password
- IP based
- WebSSO
- **Federated**

Shibboleth based AuthN/AuthR

- + Lazy sessions allow old behaviour
- + Collaborating members from other institutions can be treated like local members.
- + Abuse can be traced
- + Richer AuthR possibilities

Best seen than told

or one image is worth a thousand words

Let's see it working

The gory details

adding code to connect to the federation

We have based our development on QuiXplorer
altering several parts, though not many:



The gory details

adding code to connect to the federation

We have based our development on QuiXplorer
altering several parts, though not many:

- IP control added by the University of Basque Country

The gory details

adding code to connect to the federation

We have based our development on QuiXplorer
altering several parts, though not many:

- IP control added by the University of Basque Country
- Interface hints: the green and red dots

The gory details

adding code to connect to the federation

We have based our development on QuiXplorer altering several parts, though not many:

- IP control added by the University of Basque Country
- Interface hints: the green and red dots
- Login links



The gory details

adding code to connect to the federation

We have based our development on QuiXplorer altering several parts, though not many:

- IP control added by the University of Basque Country
- Interface hints: the green and red dots
- Login links
- Session info for authenticated access

The gory details

Apache configuration

We have used Shibboleth lazy sessions
in order to allow for unauthenticated access

The gory details

Apache configuration

We have used Shibboleth lazy sessions
in order to allow for unauthenticated access

```
<Location /consigna>  
  AuthType shibboleth  
  ShibRequireSession Off  
  require shibboleth  
</Location>
```

The gory details

Attribute release

We require minimal identity information

The gory details

Attribute release

We require minimal identity information

- Whatever is in REMOTE_USER, as user ID

The gory details

Attribute release

We require minimal identity information

- Whatever is in REMOTE_USER, as user ID
- If common name is available, we show it

The gory details

Code changes and additions

We have modified several files

The gory details

Code changes and additions

We have modified several files

index.php

- Inclusion of our function definitions
- Call of our access control funcion
- New action definitions for launching login

The gory details

Code changes and additions

We have modified several files

`.include/init.php`

- Added code for session initialization
- Added code for loading session data into global array

The gory details

Code changes and additions

We have modified several files

`.include/header.php`

- Added code for checking credentials
- Added code for showing user information
- Added code for displaying login links

The gory details

Code changes and additions

We have modified several files

`.include/fun_*.php`

- Cosmetic changes
- Added code for checking credential
- Added code for checking AuthR
- Added code for saving credentials in the database

The gory details

Code changes and additions

and we have added one file for all our functions

`.include/luis.php`

- IP address checking
- Download authorization
- Checking authentication
- Displaying AuthR privileges
- Logout (testing)



Summary

At the application level, functionality is improved

Summary

At the application level, functionality is improved

- Normal location related control preserved

Summary

At the application level, functionality is improved

- Normal location related control preserved
- Identity is optional \Rightarrow public service

Summary

At the application level, functionality is improved

- Normal location related control preserved
- Identity is optional \Rightarrow public service
- Shibboleth session \Rightarrow additional privileges

Summary

Relating to the integration work

Summary

Relating to the integration work

- Access to the code is definitely a plus

Summary

Relating to the integration work

- Access to the code is definitely a plus
- Lazy sessions have been great

Summary

Relating to the integration work

- Access to the code is definitely a plus
- Lazy sessions have been great
- The task has been both easy

Summary

Relating to the integration work

- Access to the code is definitely a plus
- Lazy sessions have been great
- The task has been both easy and fun

Summary

Relating to the integration work

- Access to the code is definitely a plus
- Lazy sessions have been great
- The task has been both easy and fun
- Help accepted at <https://forja.rediris.es/projects/tfconsigna>