Astronomy ESFRI & Research Infrastructure Cluster

ASTERICS DADI Technology Forum 3

IVOA-GWS based implementation: interoperability feedback

Sara Bertocco

G.Taffoni, B.Major, P.Dowler, M.Molinaro, G.Jerse, C.Knapic, S.Gaudet, F.Pasian

OSSERVATORIO ASTRONOMICO DI TRIESTE

INAF ISTITUTO NAZIONALE DI ASTROFISICA

Strasbourg 22-23 March 2017

WP4: Data Access, Discovery and Interoperability (DADI)

Summary

- Project aims
- Integration work
- State of the art
- Interoperability demo

Summary

→ Project aims

→ Integration work

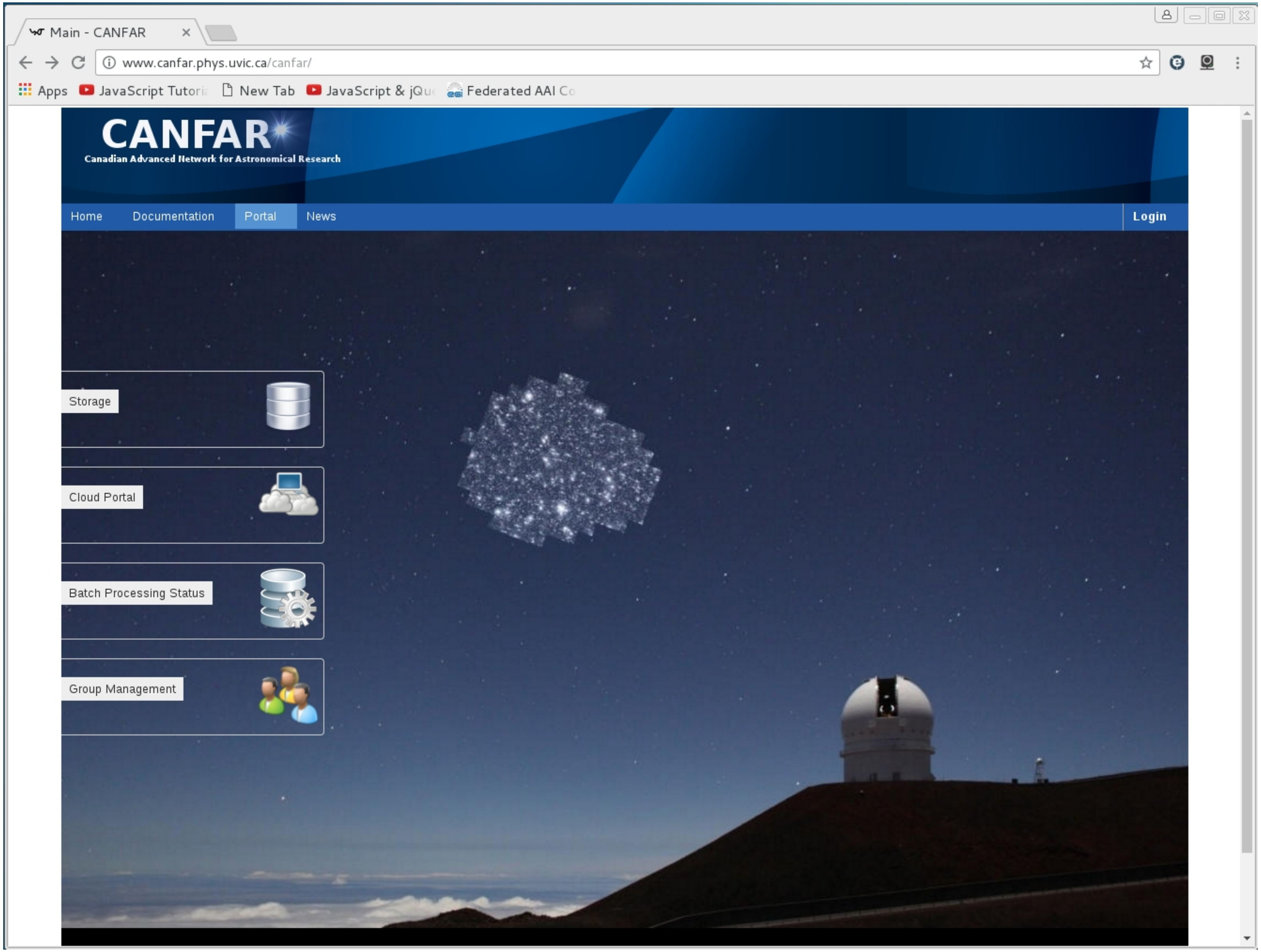
→ State of the art

→ Interoperability demo

FRAMEWORK

- In 2015, the EGI-Engage project in Europe partially funded an exploration of authentication and data access interoperability of some services in a joint project between the Canadian Advanced Network for Astronomical Research (CANFAR) and INAF-Osservatorio Astronomico di Trieste (INAF-OATs).
- SKA A&A community is interested in the full stack: users and access management.

CANFAR/CADC INFRASTRUCTURE



IVOA COMPLIANT SOFTWARE

https://github.com/opencadc **CADC**

Modules:

ac	Access Control (including GMS)
cdp	Credential Delegation Protocol implementation
vos	VOSpace standard implementation
reg	Registry Interface implementation (including VOSI)
uws	Universal Worker Service Pattern implementation
core	core utilities and logging

IVOA Standards and recommendations based

(http://ivoa.net/)

PROJECT TARGETS

- Deployment at INAF-OATs of an infrastructure twin of the already hosted at CANFAR, basically built on the same open source software libraries
- Interoperate the two infrastructures from the authentication and authorization point of view
- giving users of one infrastructure the ability to access their data stored indifferently on both infrastructures

Summary

➡ Project aims

➡ Integration work

➡ State of the art

➡ Interoperability demo

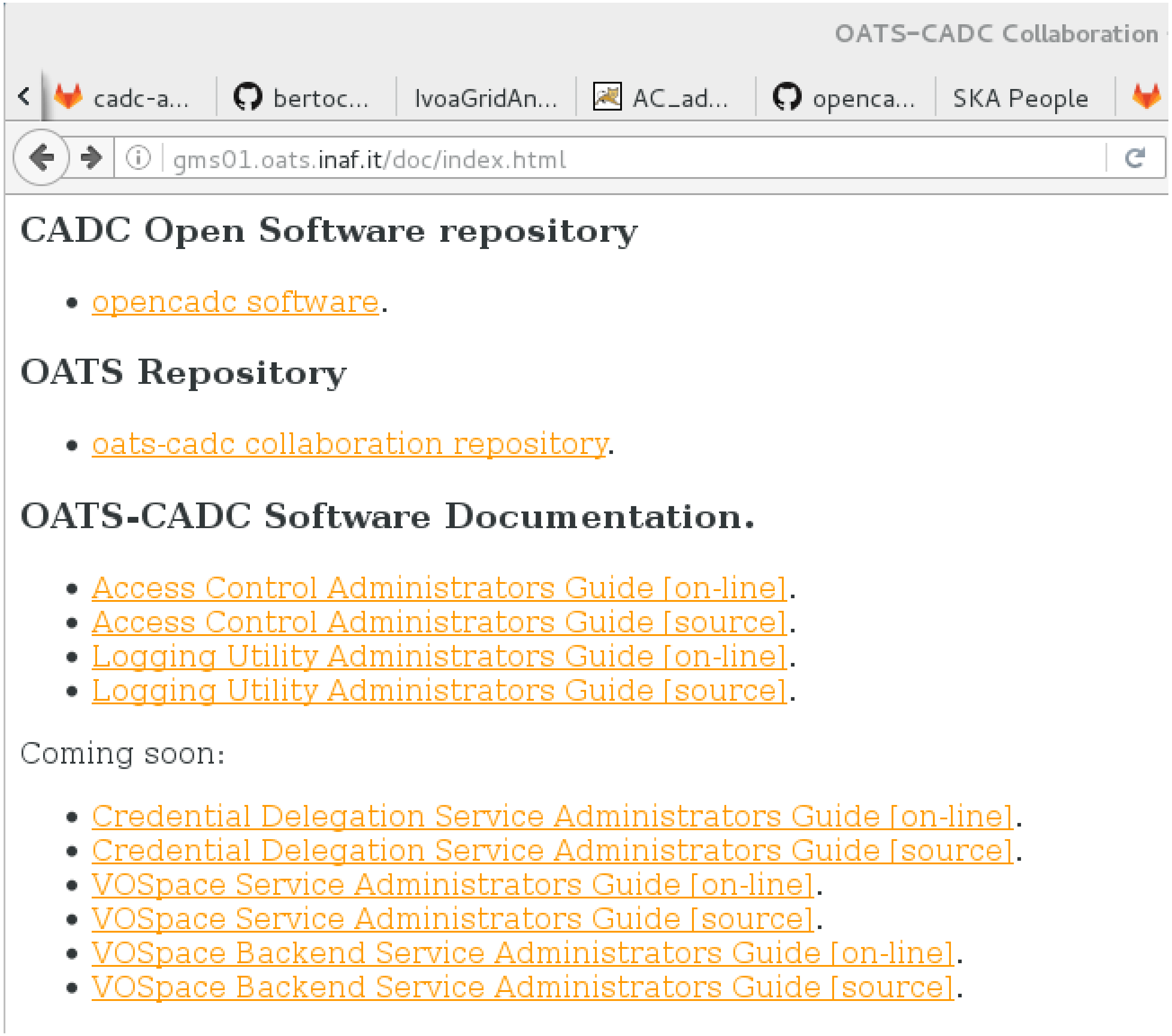
INTEGRATION WORK

- Documentation: Administrators and users guides
- Web services deployment description
- Vospace-backend development

INTEGRATION WORK: VOSPACE-BACKEND

- Takes care of the data storage
- Java application plugin based
- Posix based storage plugin available
- OpenStack Swift plugin work-in-progress

OUTCOME: Software and documentation repositories



CADC Open Software repository

- opencadc software.

OATS Repository

- oats-cadc collaboration repository.

OATS-CADC Software Documentation.

- Access Control Administrators Guide [on-line].
- Access Control Administrators Guide [source].
- Logging Utility Administrators Guide [on-line].
- Logging Utility Administrators Guide [source].

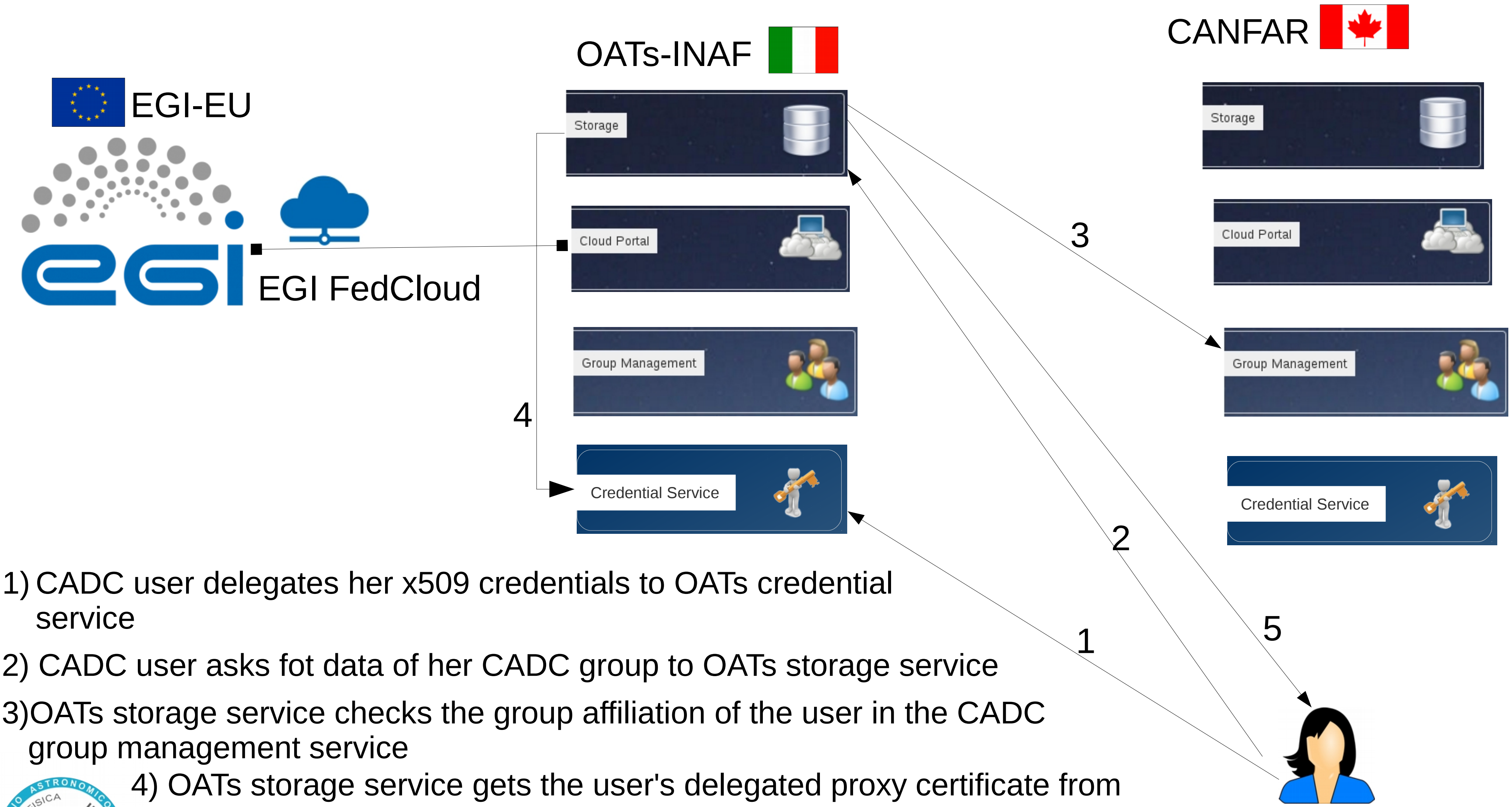
Coming soon:

- Credential Delegation Service Administrators Guide [on-line].
- Credential Delegation Service Administrators Guide [source].
- VOSpace Service Administrators Guide [on-line].
- VOSpace Service Administrators Guide [source].
- VOSpace Backend Service Administrators Guide [on-line].
- VOSpace Backend Service Administrators Guide [source].

Summary

- Project aims
- Integration work
- State of the art
- Interoperability demo

INTEGRATION: STATUS



1) CADC user delegates her x509 credentials to OATs credential service

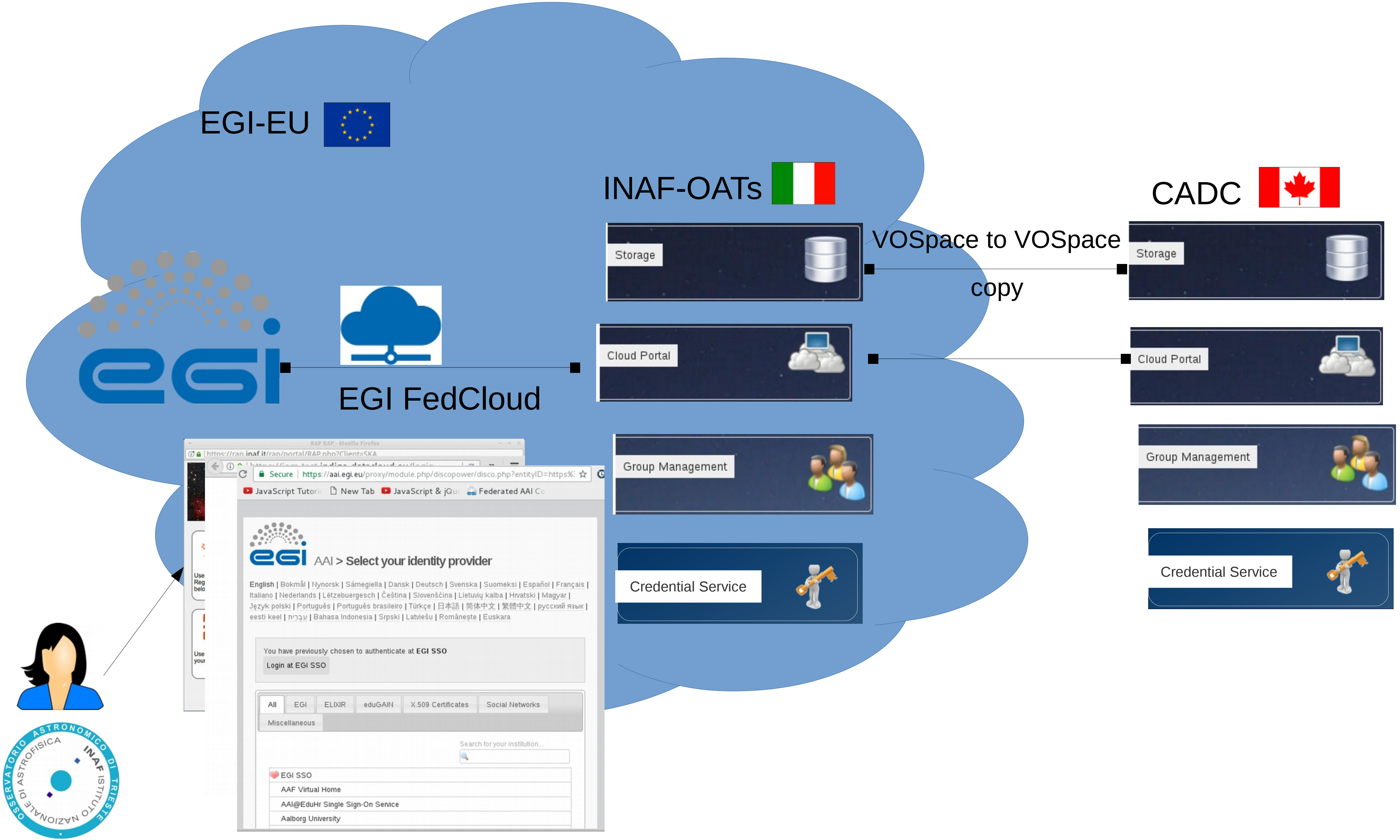
2) CADC user asks fot data of her CADC group to OATs storage service

3)OATs storage service checks the group affiliation of the user in the CADC group management service

4) OATs storage service gets the user's delegated proxy certificate from the OATs CDP service to be able to make calls to each other on behalf of the initial user

5)OATs storage service returns data to the CADC user

INTEGRATION: FUTURE PLANS



RAP RAP - Mozilla Firefox

https://rap.inaf.it/rap/portal/RAP.php?Client=SKA

https://iam-test.indigo-datacloud.eu/login

Secure | https://aai.egi.eu/proxy/module.php/discopower/disco.php?entityID=https%3

JavaScript Tutori | New Tab | JavaScript & jQu | Federated AAI Co

AAI > **Select your identity provider**

English | Bokmål | Nynorsk | Sámegiella | Dansk | Deutsch | Svenska | Suomeksi | Español | Français | Italiano | Nederlands | Lëtzebuergesch | Čeština | Slovenščina | Lietuvių kalba | Hrvatski | Magyar | Język polski | Português | Português brasileiro | Türkçe | 日本語 | 简体中文 | 繁體中文 | русский язык | eesti keel | עִבְרִית | Bahasa Indonesia | Srpski | Latviešu | Românește | Euskara

You have previously chosen to authenticate at **EGI SSO**

Login at EGI SSO

All | EGI | ELIXIR | eduGAIN | X.509 Certificates | Social Networks | Miscellaneous

Search for your institution...

- EGI SSO
- AAF Virtual Home
- AAI@EduHr Single Sign-On Service
- Aalborg University

Summary

- Project aims
- Integration work
- State of the art
- Interoperability demo

DEMO: INTEROPERABILITY USE CASE

- Two users: Taffoni and Bertocco both working on project TB_project
- Taffoni (PI) has an account both at CADC and OATs and a space allocation at CADC. He wants share with Bertocco some data in his CADC storage area
- Bertocco has an account at OATs only
- Taffoni creates his project group at OATs: ivo://oats.inaf.it/gms/TB_project
- And add Bertocco as member
- Taffoni adds group-write permission to his CADC folder at group oats.inaf.it/TB_project
- Bertocco is now authorized to upload her data at CADC:
  - Delegates her credentials at CADC credential delegation service
  - Uploads data

Demo:interoperability step by step

- Jerse has an accont at ska-gms.ia2 and a space allocation “jerse”

- Taffoni creates a folder TB_collaboration in his storage area at CADC

```
java ca.nrc.cadc.vos.client.Main -d --cert=/root/certs/giu/giu_infn_key+cert.pem  --create=ContainerNode –target=vos://cadc.nrc.ca\!
vospace/taffoni/TB_collaboration
```

- Taffoni creates the group TB_project at OATs

```
java  ca.nrc.cadc.ac.client.Main --create --group=ivo://oats.inaf.it/gms?TB_project --cert=/root/certs/giu/giu_infn_key+cert.pem -d
```

- Taffoni adds Bertocco as member of group TB_project at OATs

```
java  ca.nrc.cadc.ac.client.Main --add-member --group=ivo://oats.inaf.it/gms?TB_project --userid=sabe
--cert=/root/certs/giu/giu_infn_key+cert.pem -d
```

- Taffoni adds to group ivo://oats.inaf.it/gms#TB_project group-write permissions to his CADC folder “TB_collaboration”

```
./cadc-vos  --set --target=vos://cadc.nrc.ca~vospace/taffoni --group-write=ivo://oats.inaf.it/gms#TB_project -d –
cert=/root/.ssl/giu_proxy.pem
```

- Bertocco delegates her credentials to CADC delegation service

```
java ca.nrc.cadc.cred.client.Main -d --resourceID=ivo://oats.inaf.it/cred --delegate --daysValid=31 --cert=/root/infn_cert/sara_infn_key+cert.pem
```

- Bertocco upload her data to CADC folder

```
java ca.nrc.cadc.vos.client.Main -d --cert=/root/infn_cert/sara_infn_key+cert.pem --copy --src=/root/testfile.txt
--dest=vos://cadc.nrc.ca\!vospace/taffoni/TB_collaboration/testfile.txt
```

DEMO

QUESTIONS

?

BACKUP SLIDES

OPERATIONS: USER MANAGEMENT

- Register a new user
- List pending users
- User approval
- List users
- User registration reject

OPERATIONS:
GROUPS MANAGEMENT

- Create a new group
- List existing groups
- Search: Is this user a member of this group?
- Add a group member
- Remove a group member
- Add a group admin
- Remove a group admin
- Remove a group

OPERATIONS:
DELEGATION MANAGEMENT

- Delegate x509 credentials
- View the currently delegated x509 certificate
- Get create a new proxy certificate based on the one that has been delegated

OPERATIONS:
VOSPACE FUNCTIONALITIES

- Create a folder
- Upload a file
- Download a file
- View a file description
- Cheange read and write permission of a file

AVAILABLE CLIENTS:
JAVA COMMAND LINE

- There are java clients for: VOSpace, GMS, and CDP
- Java clients distributed in github.com/opencadc

  Usage (cdp example):

```
git clone https://github.com/opencadc/cdp        Download
cd cdp/cadc-cdp
gradle build                                     Build
cd build/distributions/
tar -xvf cadc-cdp-1.1.1.tar
cadc-cdp-1.1.1/bin/cadc-cdp -h                   Get help
```

- More details in backup slides

AVAILABLE CLIENTS: PYTHON COMMAND LINE

- https://github.com/opencadc/vostools (recently moved)
- The PyPi location is here: https://pypi.python.org/pypi/vos/

AVAILABLE CLIENTS: RESTFUL API

- Restful API documentation is available starting here:

  http://www.canfar.net/docs/api/

Operations: users management (2)

```
git clone https://github.com/opencadc/ac
cd ac/cadc-access-control-admin
gradle build
cd build/distributions
tar -xvf cadc-access-control-admin-1.0.2.tar
```

Operations: users management (3)

cadc-access-control-admin-1.0.2/bin/cadc-access-control-admin -h

```
Usage: userAdmin <command> [-v|--verbose|-d|--debug] [-h|--help]
   [--cert=<Cert File or Proxy Cert&Key PEM file> [--key=<Unencrypted Key File>]]
Where command is
--list                          : List users in the Users tree
--list-pending                  : List users in the UserRequests tree
--view=<userid>                 : Print the entire details of the user
--approve=<userid> --dn=<dn> : Approve user with userid=<userid> and set the
                             : distinguished name to <dn>
--reject=<userid>               : Delete this user request
--v|--verbose                   : Verbose mode print progress and error messages
-d|--debug                      : Debug mode print all the logging messages
-h|--help                       : Print this message and exit

Authentication and authorization:
  - An LdapConfig.properties file must exist in directory ~/config/
  - The corresponding host entry (devLdap or prodLdap) must exist
    in your ~/.dbrc file.
```

Operations: groups management (2)

cadc-access-control-1.1.6/bin/cadc-access-control -h

```
--create --group=<uri>
--get --group=<uri>
--delete --group=<uri>

--add-member --group=<uri> --userid=<u>
--remove-member --group=<uri> --userid=<u>

--add-admin --group=<uri> --userid=<u>
--remove-admin --group=<uri> --userid=<u>
```

Operations: Delegation management (2)

```
git clone https://github.com/opencadc/cdp
cd cdp/cadc-cdp
gradle build
cd build/distributions/
tar -xvf cadc-cdp-1.1.1.tar
```

Operations: Delegation management (3)

```
cadc-cdp-1.1.1/bin/cadc-cdp -h

Usage: cadc-cdp [-v|--verbose|-d|--debug] --resourceID=<CDP service to use> <op> ...

   [--cert=<Cert File or Proxy Cert&Key PEM file> [--key=<Unencrypted Key File>]]

Help: cadc-cdp <-h|--help>

  --resourceID specifies the CDP service to use (e.g. ivo://cadc.nrc.ca/cred)

  <op> is one of:

  --delegate [--daysValid=<days>]

          create new proxy certificate on the server

  --get --userid=<username> [--out=<file>] [--daysValid=<days>]

  --get --userdn=<user distinguished name> [--out=<file>] [--daysValid=<days>]

          get a new (shorter) proxy certificate from the server;

  --view

          view the currently delegated proxy certificate
```

Operations: vospace functionalities (2)

```
git clone https://github.com/opencadc/vos
cd vos/cadc-vos
gradle build
cd build/distributions
tar -xvf cadc-vos-1.0.7.tar
```

Operations: vospace functionalities (3)

```
cadc-vos-1.0.7/bin/cadc-vos -h


Usage: java -jar cadcVOSClient.jar [-v|--verbose|-d|--debug] [--xsv=off]
   [--cert=<Cert File or Proxy Cert&Key PEM file> [--key=<Unencrypted Key File>]]
  Note: --xsv=off disables XML schema validation; use at your own risk


Help:
    <-h | --help>
```

Operations: vospace functionalities (4)

Create node:

```
    --create[=<ContainerNode|LinkNode|StructuredDataNode|
UnstructuredDataNode>]
    --target=<node URI>
    [--link=<link URI>]
    [--prop=<properties file>]
```

Note: --create defaults to creating a ContainerNode (directory).

Note: --link is only required when creating a LinkNode. It is the URI to which

the LinkNode is pointing.

View node:

```
    --view --target=<target URI>
```

Delete node:

```
    --delete --target=<target URI>
```

Operations: vospace functionalities (5)

Set node:

--set --target=<target URI>

[--content-type=<mimetype of source>]

[--content-encoding=<encoding of source>]

[--group-read=<group URIs (in double quotes, space separated, 4 maximum)>]

[--group-write=<group URIs (in double quotes, space separated, 4 maximum)>]

[--lock]

[--public]

[--prop=<properties file>]

[--recursive]

Operations: vospace functionalities (6)

Copy file:

```
--copy --src=<source URI> --dest=<destination URI>
[--content-type=<mimetype of source>]
[--content-encoding=<encoding of source>]
[--prop=<properties file>]
[--noretry]
[--quick]
```

Note: --noretry disables the retry of failed transfers (when the server indicates it was temporary)

Note: One of --src and --target may be a "vos" URI and the other may be an absolute or relative path to a file. If the target node does not exist, a DataNode is created and data copied. If it does exist, the data and properties are overwritten.

Note: Source and destination URIs may include HTTP-like query parameters, some of which will result in additional operations being performed.

Note: If the --quick options is supplied, and a download is being performed, transfer negotiation will be replaced with an optimized download process.

Operations: vospace functionalities (7)

Move file/node:

```
--move --src=<source URI> --dest=<destination URI>
```

Note: If the source URI refers to a VOSpace node, then move is a recursive operation: the

source nodes, and all subnodes, are moved.

Note: Only files can be moved from the local file system to VOSpace. Similarly, only files

can be moved from VOSpace to the local file system.

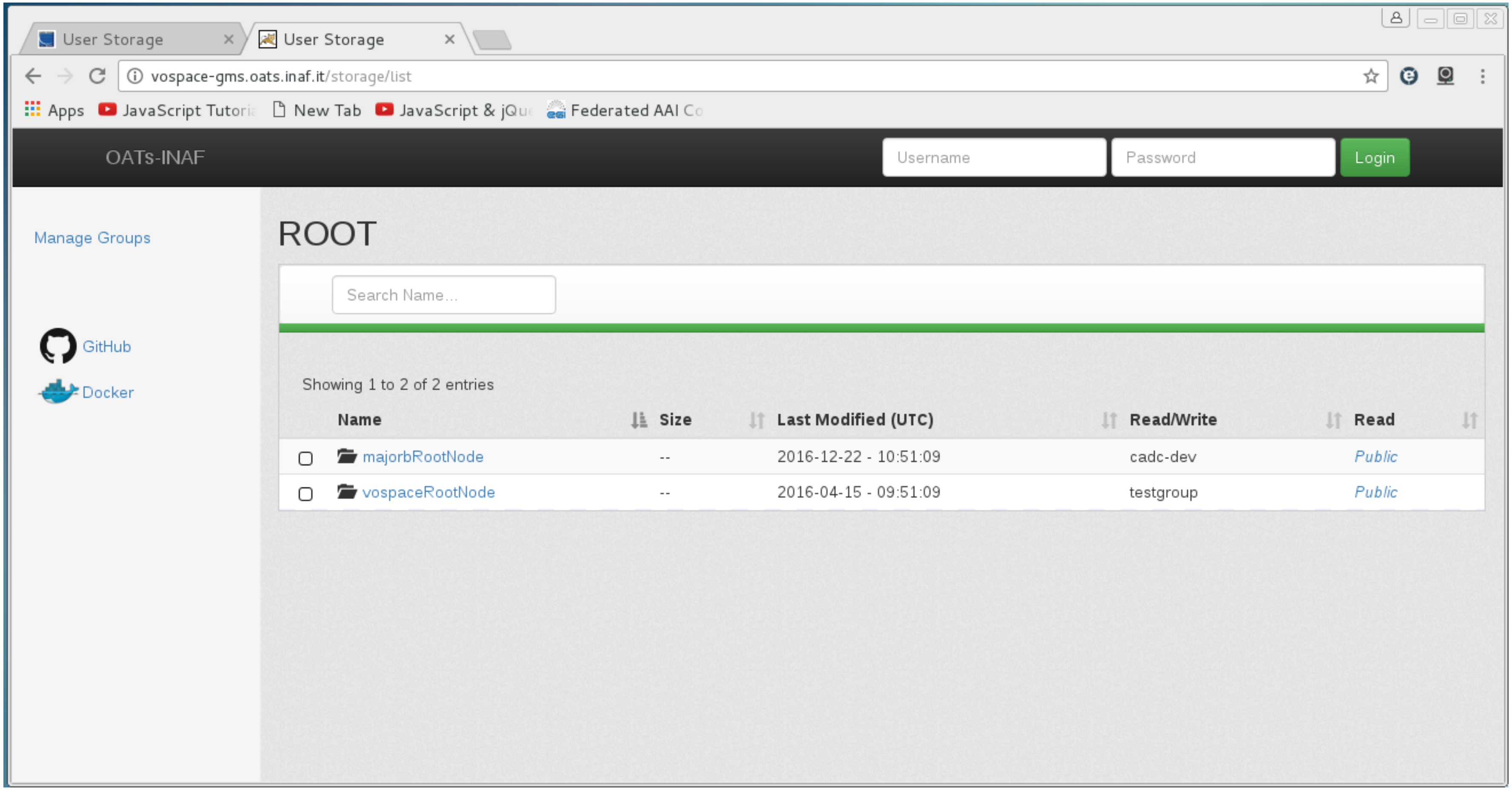
Note: If the destination URI referes to a VOSpace node, that node must be a directory. If the

directory exists, the source URI will be moved into that directory. If the directory doesn't

exist, the source URI will be moved into the parent directory and will be renamed to the name

specified in destination URI.

AVAILABLE CLIENTS: WEB GUI



ROOT

Showing 1 to 2 of 2 entries

	Name	Size	Last Modified (UTC)	Read/Write	Read
□	majorbRootNode	--	2016-12-22 - 10:51:09	cadc-dev	*Public*
□	vospaceRootNode	--	2016-04-15 - 09:51:09	testgroup	*Public*

Work in progress