Data Replication and Access Testing using Onedata

Lead Partner:	CEA, PSNC
Main author:	Frederic Imbeaux
Version:	V1.0
Dissemination Level:	Public

Abstract

This is the report of our activity on Task 8.2.4, *Data Replication and Access Testing using Onedata.*

The Fusion Competence Centre foresees two main Use Cases for the utilization of Onedata tools: 1) replication of experimental data across multiple sites, to facilitate their access and analysis by different fusion laboratories 2) enabling data access for remote data processing distributed over the EOSC. We report here about the installation and performance tests of Onedata on various configurations, using IMAS files and APIs (the standard data format and infrastructure promoted by ITER Organization). This work provides feedback and recommendations for the improvement of the Onedata software, as well as on how it can be used in an efficient way for the Fusion Competence Centre Use Cases.

EOSC-HUB RECEIVES FUNDING FROM THE EUROPEAN UNION'S HORIZON 2020 RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO. 777536.

COPYRIGHT NOTICE

TERMINOLOGY

https://wiki.eosc-hub.eu/display/EOSC/EOSC-hub+Glossary

Contents

1 Introduction

Onedata has been developed in order to provide a solution for accessing laboratories data from anywhere, using for instance a laptop connected to Internet. This feature encompasses Unified Data Access solutions which are however more restrictive since they allow data sharing only by secured SSH connections between laboratories. Moreover, we expect that Onedata clients interested in data provided by a remote Onedata space, will experience much better data access performances. They can indeed geographically select a close available Onedata provider with replicated data of a remote space from another provider, since data replication between providers is a feature offered by Onedata (it is possible to restrict data replication). Consequently, our priority was to check if Onedata can satisfy this goal: getting access to remote data from a laptop connected to Internet close to IRFM (Cadarache, France). Another important characteristic of our Use Case is that the exchanged files come from the ITER Integrated Modelling and Analysis Suite (IMAS) infrastructure. These files have a highly hierarchized internal data structure and a specific API (the IMAS Access Layer) to access them. We have therefore assessed the performance of data access through this API when the data is served by Onedata components, using two different scenarios.

1.1 Accessing remote data files using Onedata

For tests purposes, the remote data files we want to access are located at PSNC (Poznan). Our (Onedata) client is a laptop connected to Internet, located at IRFM (CEA Cadarache, France).

We have identified two alternative scenarios for testing remote data access from PSNC using Onedata.

In the first scenario, only one PSNC provider will be available for the Onedata client laptop connected to Internet. However, in a second scenario, the laptop will be able in addition to access our Onedata IRFM provider located at IRFM. For this scenario we plan to replicate remote PSNC data to our Onedata space provided by our Onedata IRFM provider located on an IRFM server using OnedataProvider services.

With the latter scenario, we will be able in the same time to test Onedata replication functionality between PSNC and IRFM providers. We expect of course that this scenario will offer better performance in accessing data than the first one since our IRFM Onedata provider is closer to the client machine network.

1.2 Onedata installation and tests plan

The documentation of Onedata can be found here:
https://onedata.org/docs/doc/getting_started/what_is_onedata.html
For both scenarios described in the previous section we have set up:

- A laptop connected to internet at IRFM
- Onedata **space** - hosted at PSNC. This space contains shot pulse files. These files are the remote PSNC data that is accessed during test phase. In order to access data inside the space the request must be sent to the administrator of PSNC provider.

Moreover, we have installed for the first scenario:

- Oneclient software (**Oneclient** command line tool) provided by Onedata team. This tool allows to create local NFS mount point to the remote data stored inside Onedata Provider. In this following cases, we were accessing data stored inside space supported by PSNC.

For the second scenario, we have in addition:

- Installed Oneprovider and Onezone on two separate IRFM servers connected to Internet
- Created our own Onedata IRFM space for enabling remote PSNC data replication

Therefore, our test plan followed:

- Make the installation of Oneprovider, Onezone and Oneclient at IRFM
- Start IRFM based Onedata provider
- Create a Onedata space @PSNC for hosting pulse files
- Enable pulse files replication from PSNC to IRFM
- Execute tests for reading pulse files from PSNC and IRFM Onedata providers

2 Installing Onedata and preparing data for the tests

2.1 Installing Onedata

2.1.1 Installing Oneprovider and Onezone services at IRFM

We have first tested a standalone installation from the on-line documentation. We have encountered compatibilities issues (Onedata RPMs being non compatible with CentOS 7) and unsatisfied prerequisites with one required RPM missing in the repository. Therefore, we have followed Onedata's support team suggestion; we have decided to perform tests using Docker based installation. This procedure makes whole process a lot easier – both configuration settings and dependencies management since libraries are provided by the Docker image.

A successful installation of Onedata 18.02.0-rc13 has been performed using the Docker image provided by Onedata - downloaded from Onedata web site. Two CentOS7 machines with Internet Access have been prepared for installing Oneprovider and Onezone services separately. In such a configuration, we have chosen to follow the **installation procedure – scenario 3**, described in the Onedata documentation.

In a second step, an IRFM Onedata account has been created at PSNC. This account was used for accessing Onezone services running at PSNC.

In a third step, public web certificates have been generated.

Setting our servers was a tedious task since network configuration on secured sites like CEA can become complex. Indeed, the Docker network resides inside the local server which, in turn, is located in the local domain behind a firewall. However, recent new features provided on the Onedata GUI to check if DNS names have been correctly set up make today this step much easier. Moreover, we have encountered several issues concerning communication port used between our provider and PSNC Onezone because these ports (IRC ports) are considered dangerous and are therefore blocked at CEA. Onedata support has rapidly provided us with new port numbers for solving this issue. Solving all these issues has been quite time consuming and we would recommend providing a diagnostic tool to detect and report the issues and thus facilitate the Onedata installation. This tool, executed after installation, would check the environment settings: matching client/server releases, potential blocked ports, writing/reading volumes permissions and so on.

2.1.2 Starting IRFM Oneprovider services

After installation, Oneprovider services are started at IRFM using the following command line:

```
./run_Onedata.sh –provider-fqdn \
one-data-irfm.partenaires.cea.fr –zone-fqdn \
Onezone.Onedata.edu.pl –set-lat-long
```

After execution of the previous script, the IRFM provider is automatically added and referenced by PSNC Onezone.

2.1.3 Installing Oneclient software on the client laptop

Oneclient provides mounts of data hosted by Onedata providers.
As described in the introduction, two tests were planned:

- In the first scenario, the IRFM client uses data provided by the PSNC provider. In this case, Oneclient enables to mount the data from the PSNC provider. We expect performance data access to be dependent on network load.
- In the second scenario, our IRFM client uses our own IRFM provider. In this case, as data replication occurs between providers, from PSNC to IRFM, we expect to experience much better performance data access as in the first test case, since the targeted data is local.

We are using the IMAS Access Layer from ITER Organization for reading data files. In a typical scenario, the whole Access Layer middleware should be compiled and installed on the host where data are supposed to be accessed. For the users convenience, it is also provided as a Docker image. The Docker environment has been installed on CentOS 7 laptop, together with the Docker image.
We have tested different ways to provide data to the Docker:

1) by Oneclient using NFS or resource mount,
2) by executing Oneclient inside the Docker.

We confirm that both solutions are running well and provide the same performances. During these tests, we have encountered an issue difficult to understand: a compatibility mismatch between Oneclient/Oneprovider releases. This incompatibility, not notified by the Onedata system, was causing a crash of the IMAS backend component when reading data.
As an example, Oneclient is executed using the following command:

```
oneclient -i -H one-data-irfm.partenaires.cea.fr -t token-xxxxxxxxx /Onedata
```

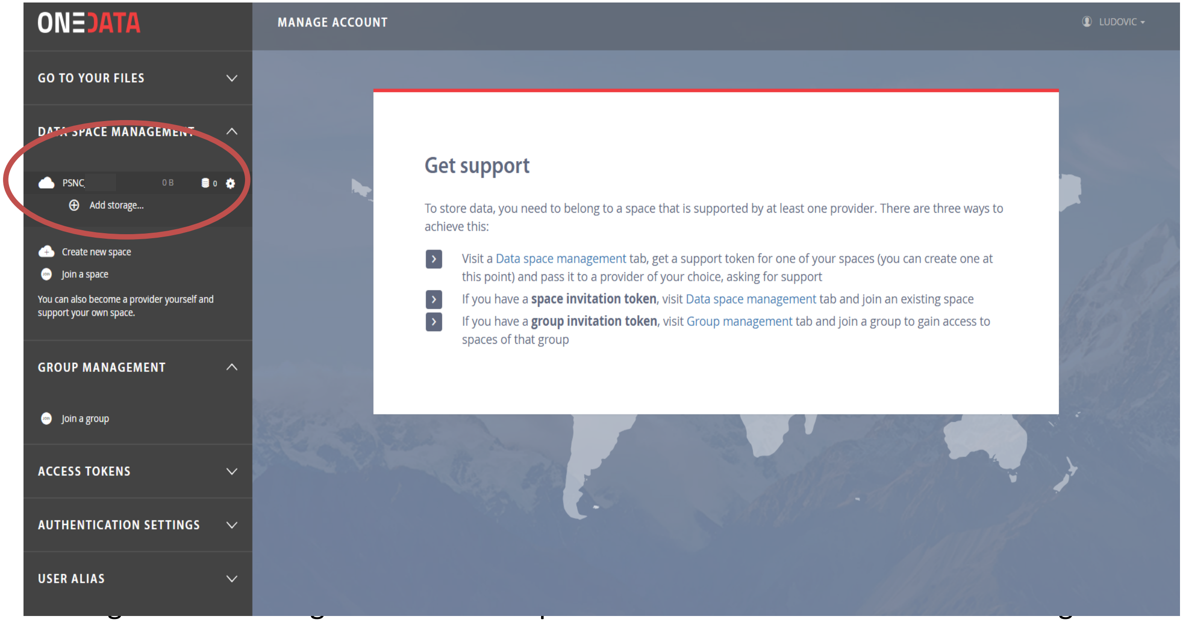
The effect of the previous command is to mount the /Onedata directory. The token token-xxxxxxxxx (generated using the PSNC Onezone GUI, this will be shown later) passed in the previous command (in –t argument) is used for authentication.

2.2 Preparing data for the tests

2.2.1 Adding some data files at PSNC

As indicated previously, our PSNC collaborators have created an account for us used for accessing Onezone services running at PSNC.

Once logged in at https://onezone.onedata.edu.pl, we create a new space called PSNC (see fig.1)



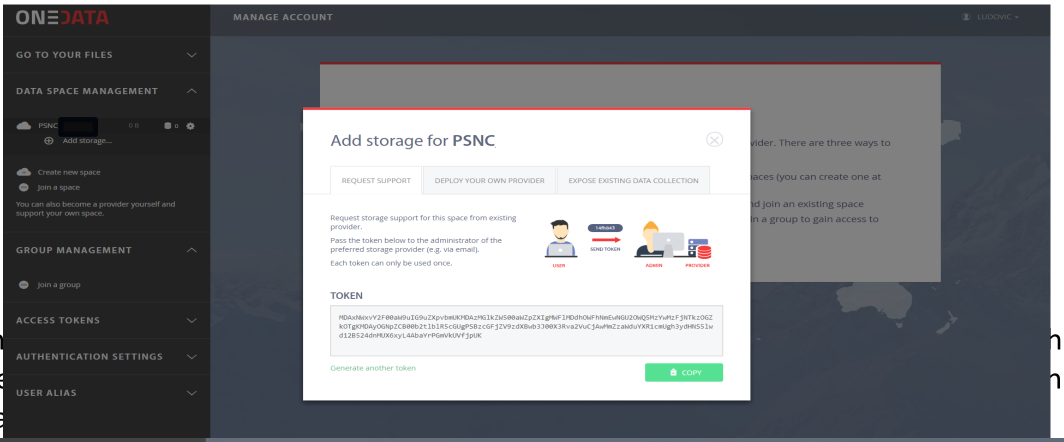
You can also become a provider yourself and support your own space.

Get support

To store data, you need to belong to a space that is supported by at least one provider. There are three ways to achieve this:

- Visit a Data space management tab, get a support token for one of your spaces (you can create one at this point) and pass it to a provider of your choice, asking for support
- If you have a **space invitation token**, visit Data space management tab and join an existing space
- If you have a **group invitation token**, visit Group management tab and join a group to gain access to spaces of that group

a token (see fig.2) that we send to PSNC admins as a request in order to provide storage support for the
space owned by the PSNC provider.



You can also become a provider yourself and support your own space.

Add storage for **PSNC**

Request storage support for this space from existing provider.

Pass the token below to the administrator of the preferred storage provider (e.g. via email).

Each token can only be used once.

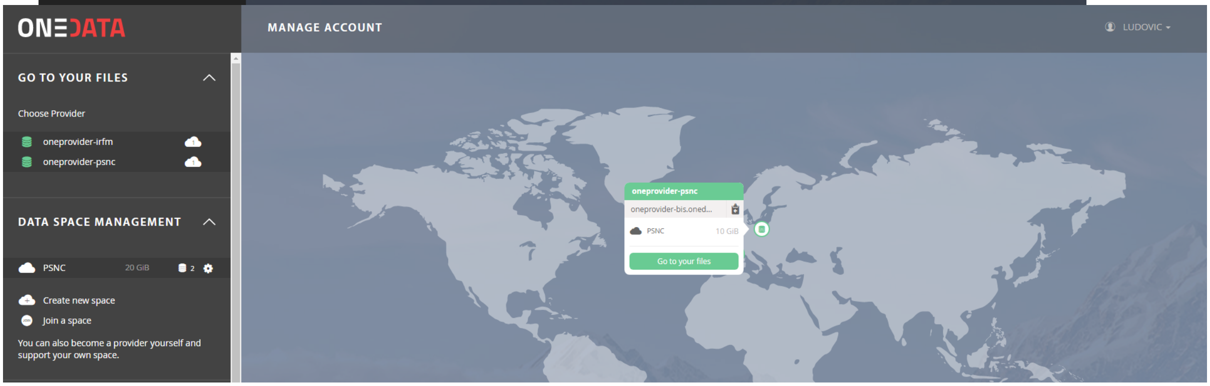
TOKEN

```
MDAxNWxvY2F00aW9uIG9uZXpvbmUKMDAzMGlkZW500aWZpZXIgMWFlMDdhOWFhNmEwNGU2OWQ5MzYwMzFjNTkzOGZ
kOTgKMDAyOGNpZCB00b2tlblR5cGUgPSBzcGFjZV9zdXBwb3J00X3Rva2VuCjAwMmZzaWduYXR1cmUgh3ydHNS5lw
d12B524dnMUX6xyL4AbaYrPGmVkUVfjpUK
```

On … he

pre … a world

ma



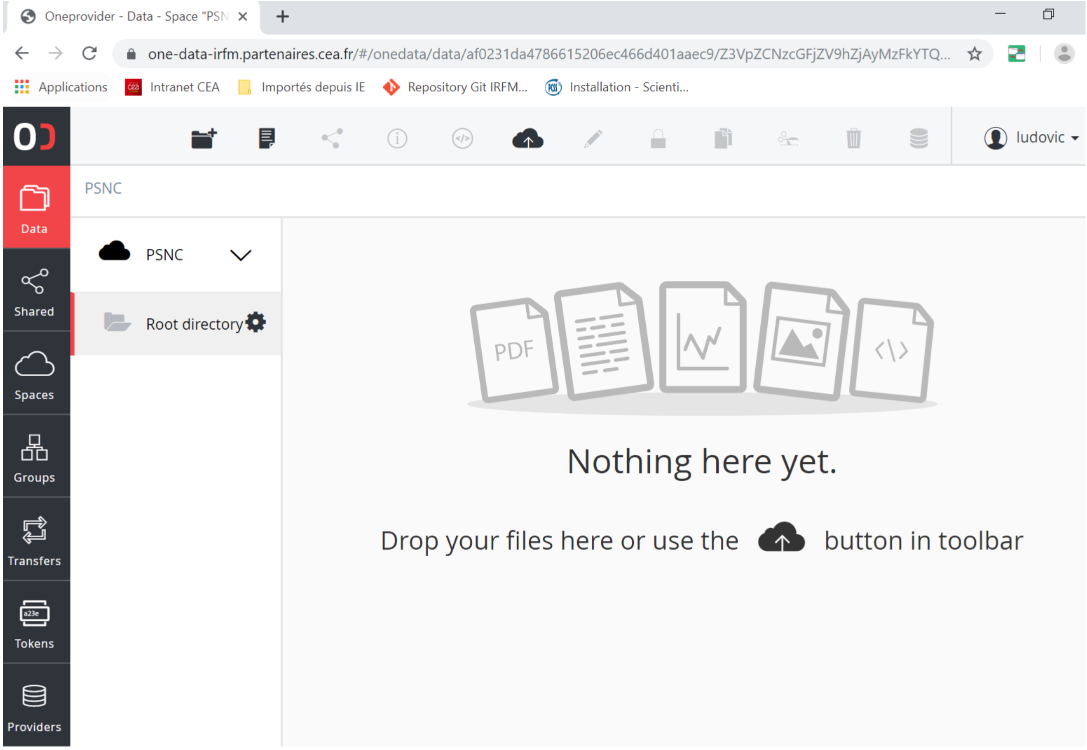
You can also become a provider yourself and support your own space.

Figure 3: Onezone GUI for adding storage support to PSNC provider

Clicking on 'Oneprovider-psnc' displays a new window (fig.3) with the PSNC space newly created. As indicated, PSNC space size is 10Gb – it has been allocated by the PSNC admin.

Now, we can start by copying pulse files in this PSNC space by clicking first to the ‘Go to your files’ button, then dragging our files in the location indicated (see fig.4). Files upload progress status is displayed (Fig.5). When the upload of the files is finished, we end up with the display of fig.6. Instead of using the GUI, we used also a more convenient way to upload the files (particularly adapted in a production context) by executing the following command lines:

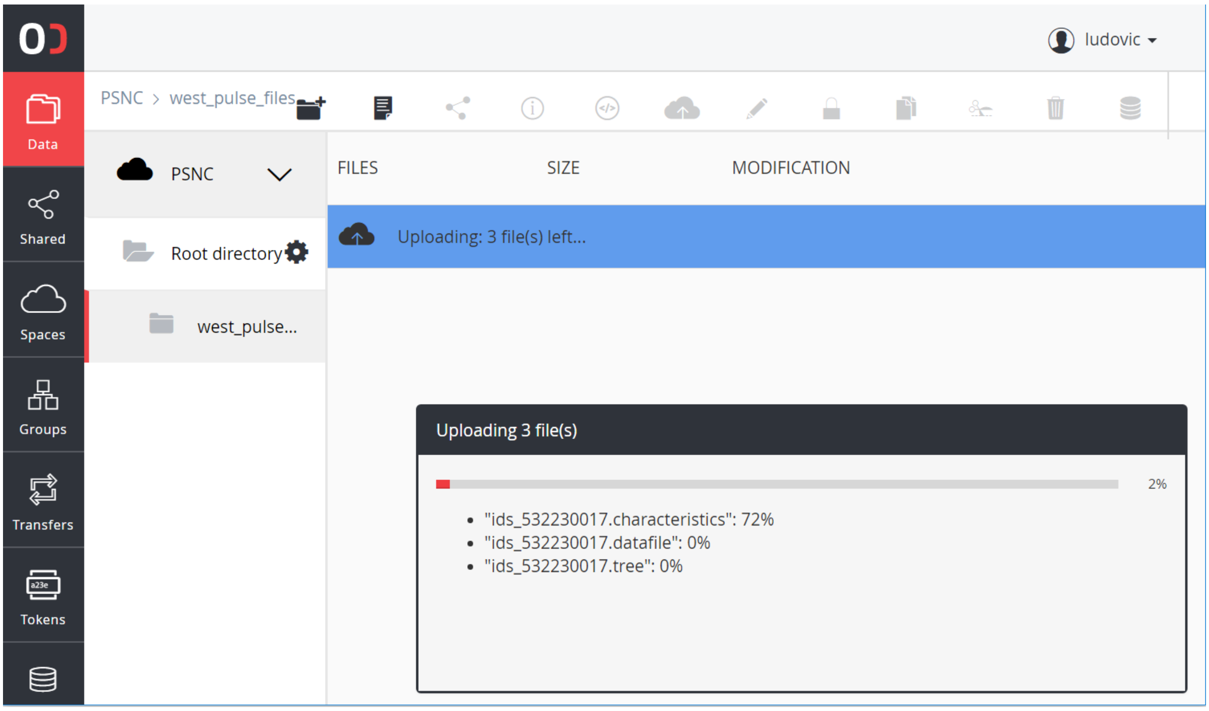
```
oneclient -i -H one-data-irfm.partenaires.cea.fr -t token-xxxxxxxxx /onedata
cp -rp location_of_some_pulse_files /onedata
```


Oneprovider - Data - Space "PSN
one-data-irfm.partenaires.cea.fr/#/onedata/data/af0231da4786615206ec466d401aaec9/Z3VpZCNzcGFjZV9hZjAyMzFkYTQ...
Applications
Intranet CEA
Importés depuis IE
Repository Git IRFM...
Installation - Scienti...
ludovic
PSNC
Data
Shared
Spaces
Groups
Transfers
Tokens
Providers
PSNC
Root directory
PDF

Nothing here yet.

Drop your files here or use the button in toolbar

Figure 4: Onezone GUI for adding data files to the PSNC space



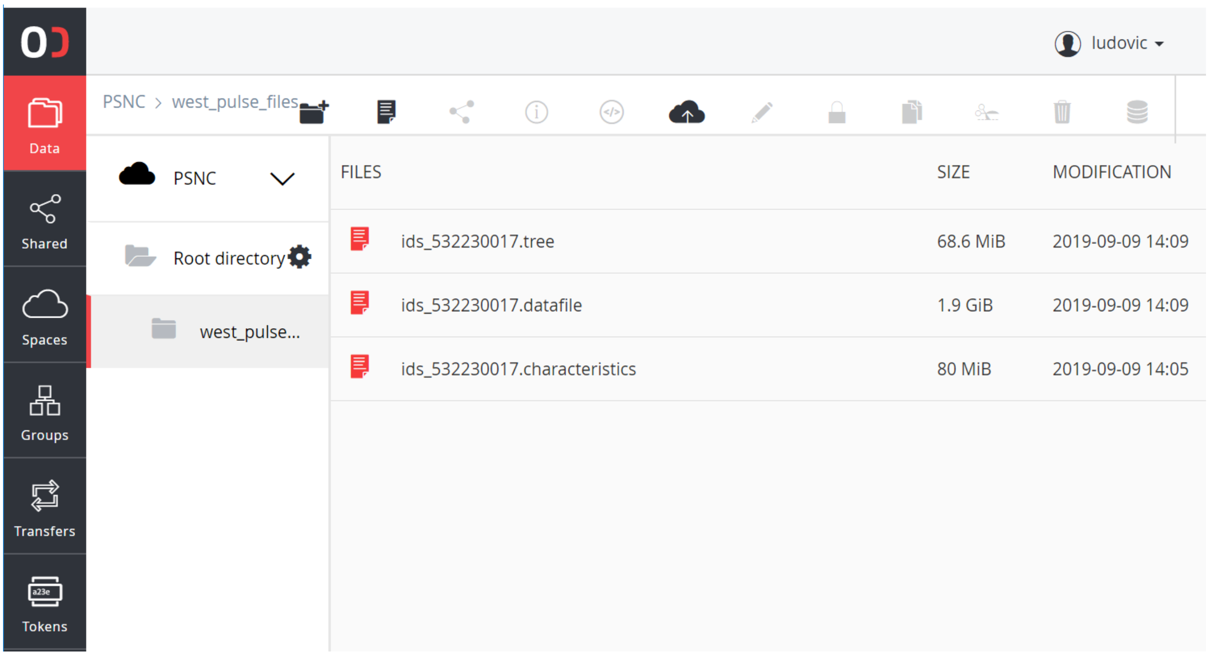
Uploading: 3 file(s) left...

Uploading 3 file(s)

2%

- "ids_532230017.characteristics": 72%
- "ids_532230017.datafile": 0%
- "ids_532230017.tree": 0%

Figure 5: files transfer upload progress displayed in the Onezone GUI



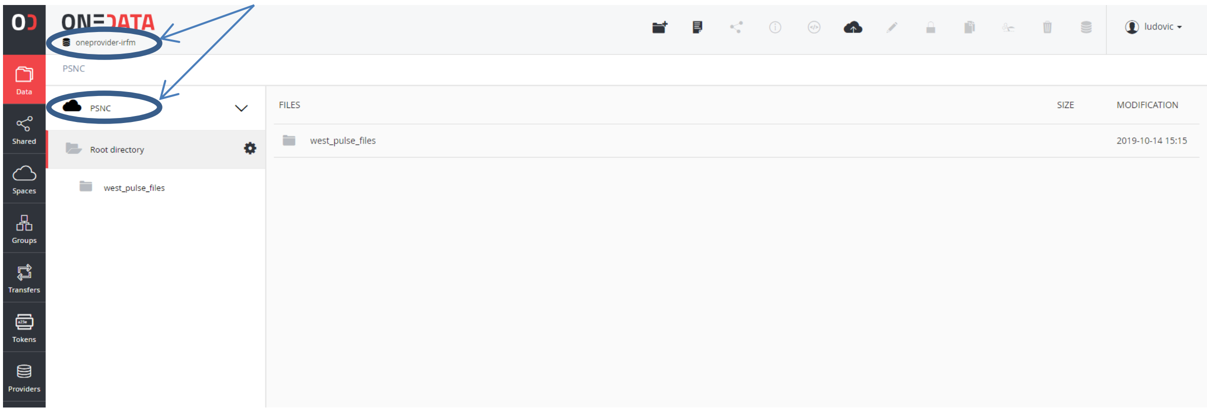
FILES	SIZE	MODIFICATION
ids_532230017.tree	68.6 MiB	2019-09-09 14:09
ids_532230017.datafile	1.9 GiB	2019-09-09 14:09
ids_532230017.characteristics	80 MiB	2019-09-09 14:05

Figure 6: transferred files displayed in the Onezone GUI after upload

For testing purposes, we have copied WEST experiment data files - with size about 2Go. WEST is the tokamak experiment operated by IRFM at Cadarache, France.

2.2.2 Replicating remote PSNC data to Onedata space hosted by the IRFM provider

In order to enable data replication from PSNC to IRFM, we have created our own Onedata space using Onezone services hosted at IRFM. Using the GUI, the difficulty was to check whether the replication was performed correctly. The GUI component (responsible for displaying data transfer) was not refreshing correctly. However, we could check the replication process using other tools. Fig.7 shows the new PSNC space hosted by our provider (Oneprovider-irfm).



FILES	SIZE	MODIFICATION
west_pulse_files		2019-10-14 15:15

Figure 7: the newly created PSNC space hosted by Oneprovider-irfm

3 Executing the tests

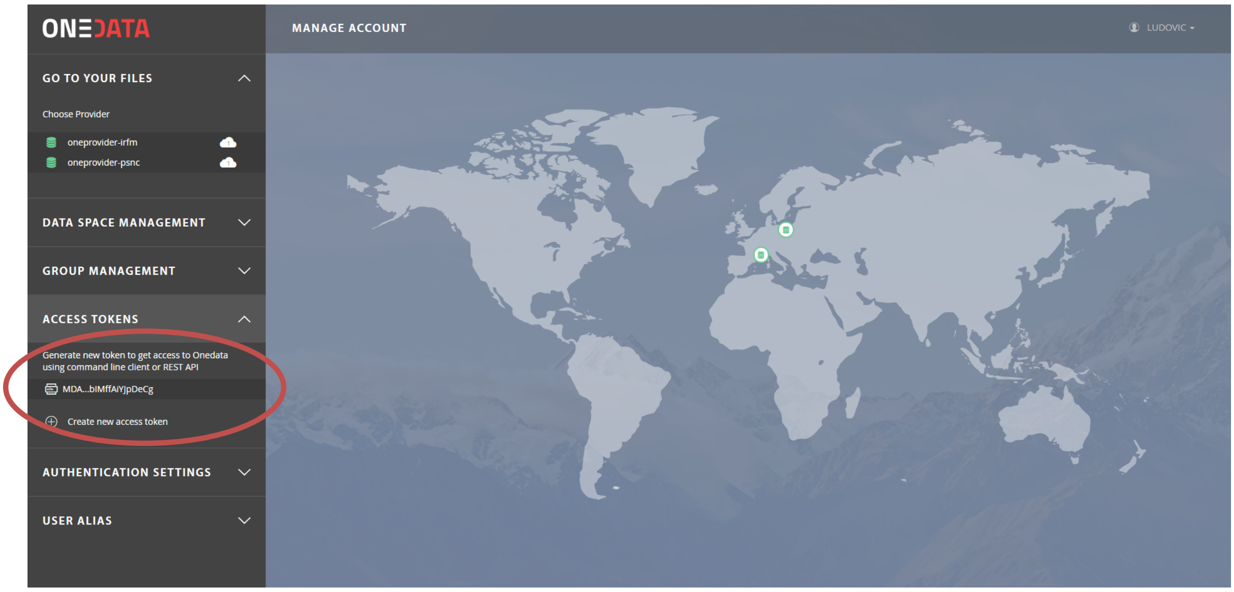
Data files, used during the test, have been generated by the current WEST Plasma Reconstruction Chain at IRFM. These files are fully representative to our Use Cases. As indicated previously, data files are hosted at PSNC - provided by the Onedata PSNC provider. Using Onedata replication feature, a copy of these data files were provided by the IRFM provider as well.
Our tests consist of reading these data files using the IMAS data access layer. Times to achieve data loading between the two providers will be compared.

3.1 Mounting the Onedata spaces using Oneclient

First, we have used data files from the PSNC space provided by the PSNC provider 'Oneprovider-psnc'. We execute Oneclient from the command line from a laptop linux session connected to Internet:

```
oneclient -H Oneprovider.Onedata.edu.pl -t token-xxxxxxxxx /Onedata
```

The previous command mounts the data in the directory `/Onedata` which is a link to a directory recognized by IMAS. In Scenario #1, data files are accessed from the mounted directory by the IMAS Access Layer.
The token used in the previous command has been generated from Onezone services hosted at PSNC using the GUI as depicted on fig.8.



Generate new token to get access to Onedata using command line client or REST API

MDA...bIMffAiYJpDeCg

Figure 8: Generating a token for Oneclient access to PSNC space

In the Scenario #2, we use the space located at IRFM provider – it contains replicated data from PSNC. We execute Oneclient from the command line inside a laptop Linux session:

```
oneclient -i -H one-data-irfm.partenaires.cea.fr -t token-xxxxxxxxx /Onedata
```

As previously, the command mounts the data in the folder `/Onedata` which is a link to a directory recognized by IMAS, the difference with Scenario #1 being that the provider resides on the local network.
It is known that the IMAS MDS+ backend has poor read/write performances when data are located on a NFS disk. This performance issue is caused by the large number of lock/unlock

operations. The MDS+ I/O performs these operations - internally. In order to evaluate the impact of NFS on our tests, we have also tested the use of `SSHFS` mounts with no-lock option enabled for both scenario 1 and 2. Therefore, we have performed 4 tests (plus one local access test given as reference).

3.2 Performing the tests

In order to test performance, we execute the following python commands under ipython:

```
import imas
imas_entry = imas.ids(53223,17)
imas_entry.open_env(‘test_user’,’test’,’3’)
%timeit imas_entry.magnetics.get()
```

The last line of the program above measures the time to load the “magnetics” data structure.

Table 2 gives the time to execute the Python program shown above.

#Test	Mounting of the One Data space	Access to	Loading time(s)	Remarks
1	**Local access (reference)**	IRFM local network	real 0m1.541s user 0m1.027s sys 0m0.340s	Data files are located on a disk at IRFM. The python script uses a local IMAS installation. Neither Onedata or Docker are used here.
2	**Oneclient access via NFS mount**	PSNC Onedata provider, Scenario #1	real 2m42.815s user 0m0.884s sys 0m0.778s	Oneclient is executed to access the Onedata space. The python script is executed in the Docker.
3		IRFM Onedata provider, Scenario #2	real 0m45.474s user 0m0.882s sys 0m0.759s	
4	**Oneclient access via SSHFS (no-lock option enabled)**	PSNC Onedata provider, Scenario #1	real 0m22.719s user 0m0.826s sys 0m0.724s	Oneclient is executed to access the Onedata space which is mounted in the Docker using SSHFS with no-lock option enabled. The python script is executed in the Docker.
5		IRFM Onedata provider, Scenario#2	real 0m4.877s user 0m0.835s sys 0m0.719s	

Table 1: time comparisons to load data from PSNC and IRFM providers, local access test result is given as reference

In order to be sure these results are reproducible, these tests have been repeated after having removed the two Onedata providers (PSNC and IRFM) and restarted our Onedata services at IRFM. Then all steps described above have been repeated to prepare the data and build the different contexts (with and without `SSHFS` mount) for executing the

Python script file. When repeating the test, only one issue has occurred because we had provided this time a pulse file (~5.2Gb) larger that the available space size (5Gb). Replication was exceeding our space size limit causing our Onedata server to be disabled (turning into grey on the GUI world map). Onedata support has confirmed the issue and indicated that the fix was available from version 19.02.0RC1. In order to continue the tests without upgrading, the PSNC space has been increased to 10Gb by the PSNC admin (as shown by fig.3 which is a capture performed during the second execution of these tests).

3.3 Analysis of the results

The results have been reported in table 1. The 'local access' result has been performed from data provided on a 100Mbits/s LAN attached storage. As expected, this represents the best performance - obtained in our tests. It provides a reference for the other measurements (~1.5 s to read the magnetics data). Using Onedata, as explained above, better performances are obtained by the MDS+ backend when using SSHFS with no-lock option enabled in tests #4 and #5 (compared to tests #2 and #3 with NFS mounts) for both scenarios 1 and 2. Without data replication, remote magnetics data access using Oneclient + SSHFS takes ~22.7s using the PSNC provider in test #4. Using data replication to access a local Onedata folder (test #5) speeds up data access by a factor of ~5 with respect to remote access (test #4). The difference in performance between 'local access' (test #1) and local replication (test #5) is essentially due to the use of a slower WAN network (~30 Mbits/s) in test #5. Note we have checked that executing Oneclient outside or inside the Docker makes no difference.

4 Conclusions

Onedata has been tested using real environments and use cases using servers at PSNC and IRFM, hosting real data files produced by the WEST tokamak. Our tests have covered the following features:

- GUI authentication to remote PSNC Onezone services and browsing the client web application
- Token generation from remote PSNC Onezone services for enabling IRFM provider registration to PSNC Onezone services
- IRFM Onedata provider registration to PSNC Onezone services using Onedata generated token
- Remote/local Onedata spaces creation using Onezone and Oneprovider services running at PSNC and IRFM
- Token generation to use for remote data support storage request of PSNC space
- Data replication between PSNC and IRFM providers
- Token generation from Onezone services for accessing data using Oneclient
- Oneclient access to each provider using generated data access tokens

At the end, we were able to read data from PSNC and IRFM providers.
For all these features, no issue has been observed except an issue difficult to identify caused by releases mismatch between client and server and the difficulty to see if data replication has been performed or not (refresh issue of the GUI component displaying files transfer progress).

Onedata installation at IRFM was the most difficult task and required regular interaction with the Onedata team to identify and solve the issues, which were mainly:

- Bad setting of machine names in DNS
- Blocked communication ports considered as dangerous by CEA
- Mismatch between Onedata client and server releases

As indicated previously, we have encountered the latest issue at compilation and also at runtime (having tested different versions of Onedata).

In order to minimize potential issues caused by bad installation settings, we recommend Onedata team to develop a diagnostic tool which could check the installation, supporting efficiently installation by new Onedata users.

Our last recommendation concerns the current requirement of matching client and server releases. We are surprised by such requirement - it will become unpractical as the number of Onedata clients and servers increases with time. Therefore, it is required that clients with newest releases continue to support older servers (backward compatibility).

Finally, we consider that Onedata is a very good candidate for data sharing and replication between fusion institutes and laboratories. We hope that our tests and recommendations will contribute making Onedata easier to install and more practical to use.