

# Irods

## Retours d'expériences

Stockage et partage de données

28 juin 2019

EDYTEM – CNRS USMB MCC

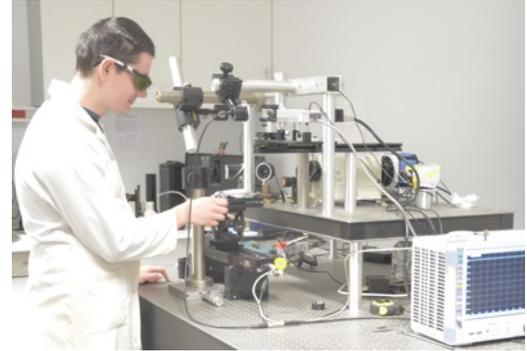
## AU menu ...



**EDYTEM**



**Projets vidéos**



**Images hyperspectrales**

# 4 équipes

Sociétés

Matières

Zone Critique

Morphodynamiques



Changement Environnemental et Sociétés



*Epistémologie, Politiques et Espaces de Montagne*



Ressources et Patrimoines



**42USMB/16CNRS**

InEE, InSHS, INSU, InC

1

# Pôles de compétences



**16** membres ...

6 BAP

## 2 - SUIVI VIDÉO

# Grotte de Prérouge

## Prérouge Acquisition des données

- Caméra IP sur le site pouvant enregistrer et transmettre des images ou des vidéos.
- Transmission réalisée par l'utilisation d'un modem 4G LTE
- Réception sur notre serveur en http par une servlet (serveur Apache Tomcat)
- Cette servlet utilise la bibliothèque java Jargon pour communiquer avec iRods.
- Référence : <https://github.com/DICE-UNC/jargon>

- Application Java EE/JSF et serveur Apache Tomcat
- Encore sous forme de prototype, mais a permis de tester les requêtes avec Jargon
- Utilisation de requêtes style SQL pour accéder aux données
- Simple d'utilisation si toutes les initialisations ont été effectuées (comme tout Jargon)
- Exemple : "SELECT COLL\_NAME, DATA\_NAME WHERE COLL\_NAME like " + repCour + "%" + " AND DATA\_NAME >= " + dataDeb + " AND DATA\_NAME <= " + dataFin + """, nbMax);
- Pas utilisé les métadonnées dans l'application. Seulement en test.
- => <http://193.48.120.232:8080/prerouge/>

<http://193.48.120.232:8080/prerouge/>

Visualisation caméra Pré Rouge V 0 X

193.48.120.232:8080/prerouge/choixImages.xhtml

Débuter avec Firefox RENCi Cloud Browser

### Visualisation caméra Pré Rouge V 0.1 / Sélection d'une séquence

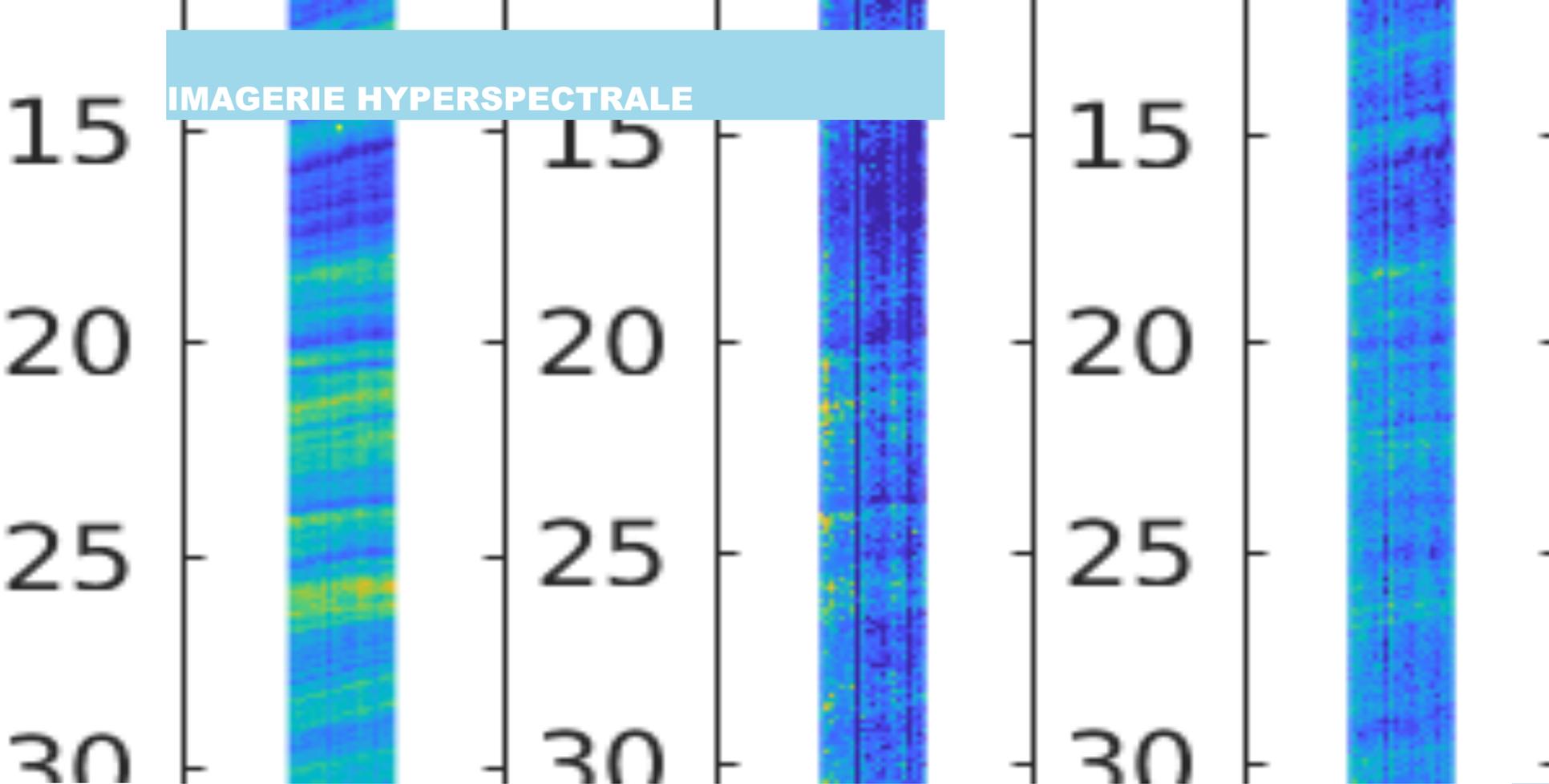
Date de début : 05/06/2019 15:42 Date de fin : 06/06/2019 15:42 Valider

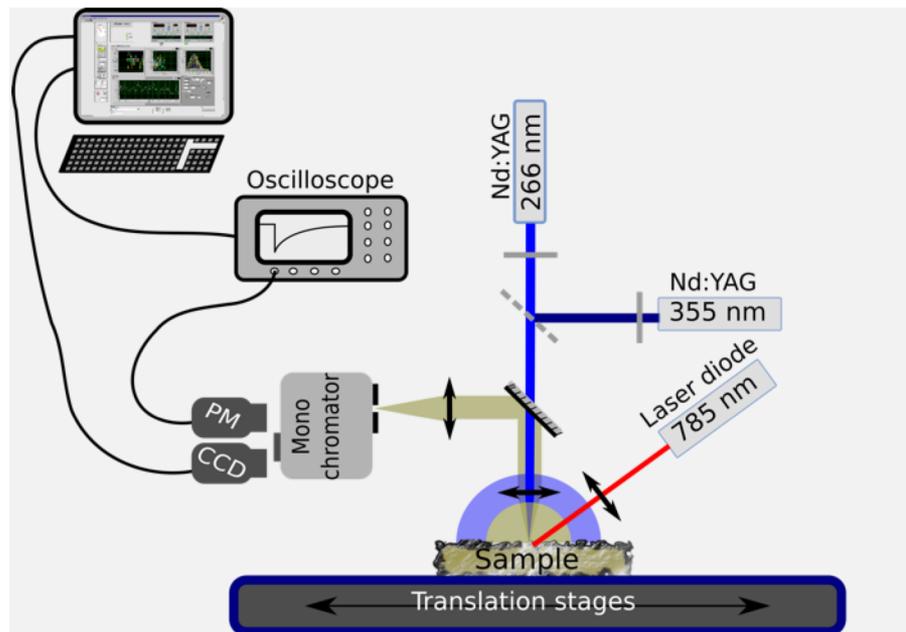
<< Début < Précédent Image du 05/06/2019 à 17h19mn00s Suivant > Fin >>

Etat : ON

[Retour au direct](#)

# IMAGERIE HYPERSPECTRALE



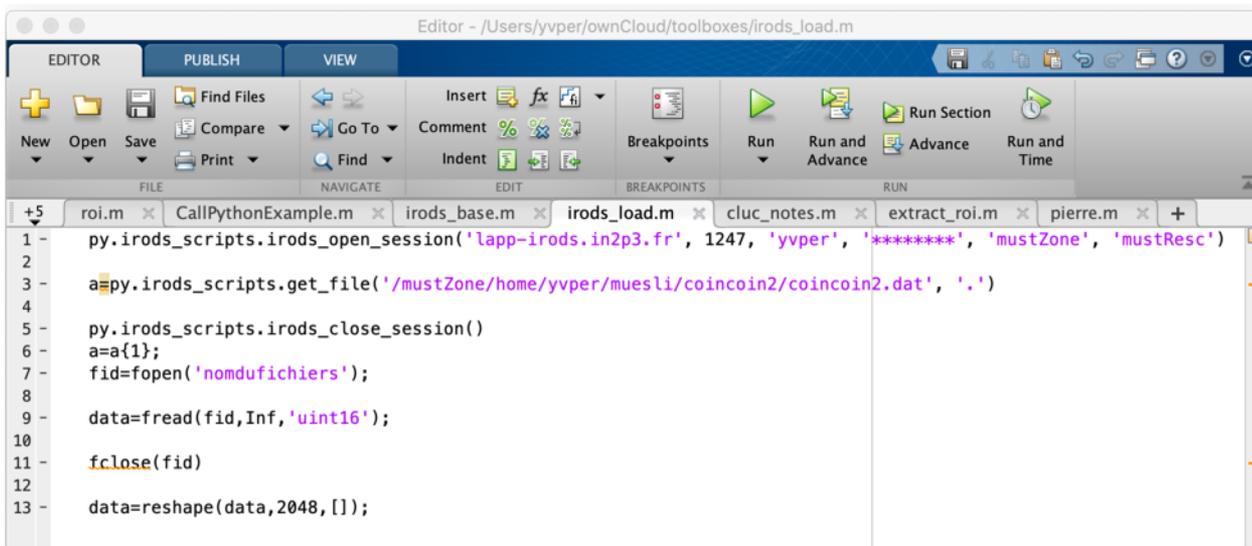


MUESLI (Muesli Uses Emission Spectroscopy for Long core Imaging)

- Pas (trouvé) d'accès à iRods via un système de fichier utilisable directement sous Labview-Windows
- Depuis 2018, Labview permet d'appeler des fonctions Python dans les VI
- Publiée récemment, la bibliothèque python-irodsclient (et très récemment l'authentification PAM)
- Développement d'une bibliothèque utilisant python-irodsclient, pour nos besoins
- Utilisation de cette bibliothèque (wrapper) pour créer une bibliothèque de VI
- Référence : <https://github.com/irods/python-irodsclient>

- En fait, quels sont les besoins ?
- Comme d'habitude : lister un répertoire, lire/écrire un fichier, rechercher, etc.
- Bibliothèque de VI : `irods_open_connexion(server, port, user, ...)`,  
`irods_put_file(filename)`, `irods_get_file(filename)`,
- `irods_append_file(filename, buffer)`, ..., `irods_close_file(filename)`,  
`irods_close_connexion()`.
- Utilisation des métadonnées : `irods_add_meta(key1, val21, ...)` pour inclure les métadonnées de l'expérience,
- pour un accès futur aux données via une application Java EE/JSF.
- Il n'a a plus qu'a tirer des fils...

- Même problème qu'avec Labview : Accès à iRods sous Windows dans une application spécialisée.
- La bibliothèque utilisée pour les VI Labview fonctionne parfaitement avec Matlab.



The screenshot shows the MATLAB Editor window with the following code in the script editor:

```
1 py.irods_scripts.irods_open_session('lapp-irods.in2p3.fr', 1247, 'yvper', '*****', 'mustZone', 'mustResc')
2
3 a=py.irods_scripts.get_file('/mustZone/home/yvper/muesli/coincoin2/coincoin2.dat', '.')
4
5 py.irods_scripts.irods_close_session()
6 a=a{1};
7 fid=fopen('nomdufichiers');
8
9 data=fread(fid,Inf,'uint16');
10
11 fclose(fid)
12
13 data=reshape(data,2048,[]);
```

- Jargon et python-irodsclient répondent à nos besoins.
- Bibliothèques difficiles à mettre en oeuvre, besoin de la documentation et d'exemples.
- Le développement des ses propres bibliothèques (wrappers) en facilite l'utilisation
- ... *utilisateurs ext*
- ... *métadonnées*

