

Implémentation Java de VOspace



Tuteur en entreprise : Grégory MANTELET

Tuteur pédagogique : Julien HARISTOY

Grégory ADAM



IUT	Robert Schuman	
Institut universitaire de technologie		
Université de Strasbourg		



□ Sommaire

- 1) Présentation de l'Observatoire
- 2) Le standard VOSpace
- 3) Les bibliothèques utilisées
- 4) L'implémentation du service
- 5) Les fonctionnalités de VOSpace disponible
- 6) Perspectives futures

□ Présentation de l'Observatoire

- Établissement de recherche et d'enseignement
- Fondé en 1881
- Unité mixte de recherche du CNRS et de l'Université de Strasbourg
- Composé de deux équipes de recherche :
 - GALHECOS
 - Le CDS
- Membre de IVOA



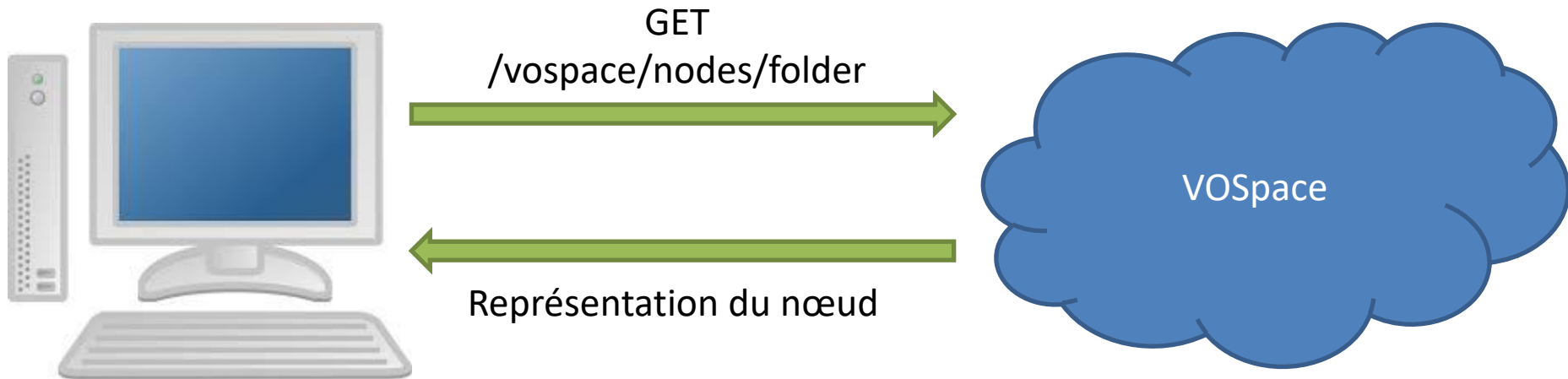
□ Le standard VOSpace

- Mise en place par l'IVOA
- Spécifie comment accéder des données
- Manière universelle d'échanger des données
- Les données sont représentées sous forme de nœuds



□ Déroulement d'une requête simple

Exemple de GetNode



□ Déroulement d'une requête simple

Exemple de GetNode

Requête du client :

```
curl -v "localhost:8080/vospace/nodes/folder"
```

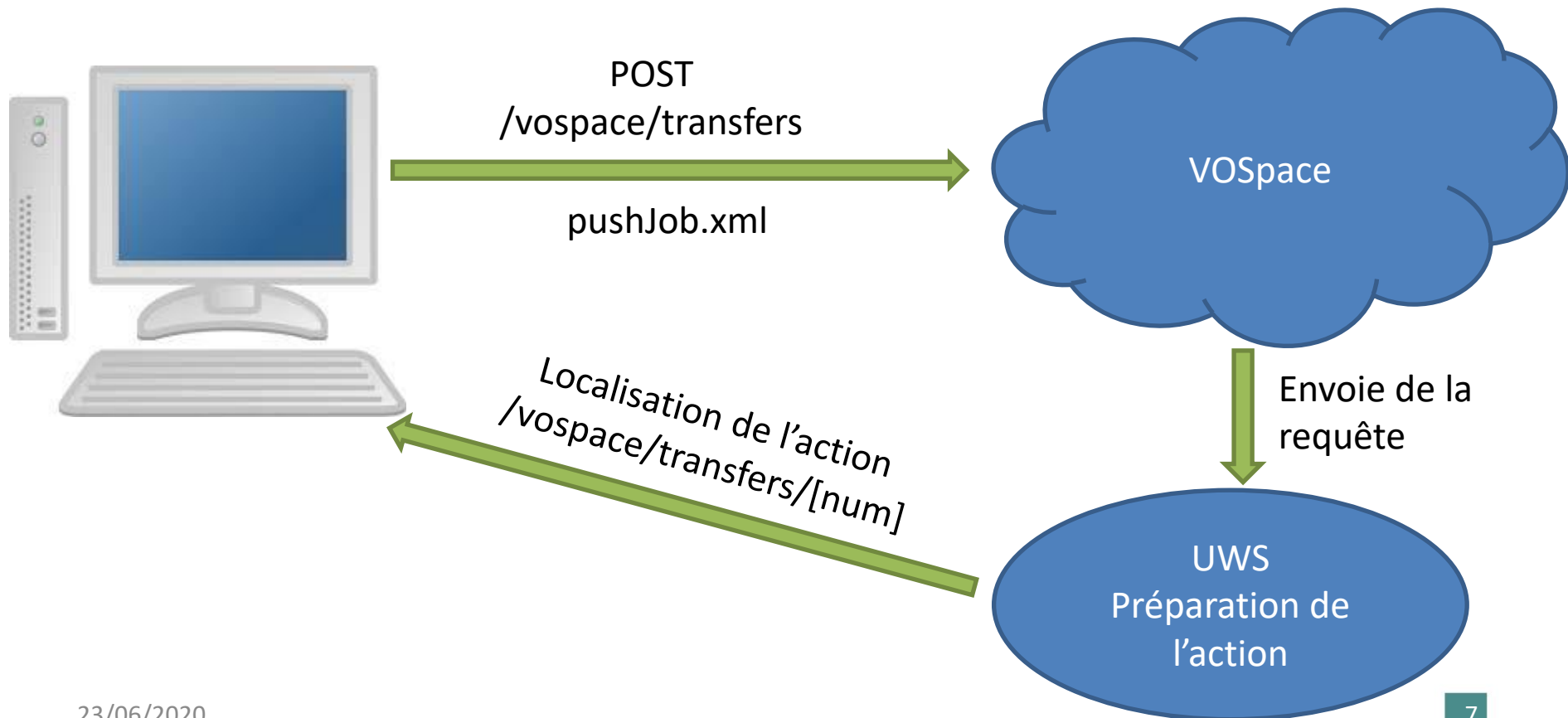
Réponse du serveur :

```
<vos:node xmlns:vos="http://www.ivoa.net/xml/VOSpace/v2.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-i
nstance" uri="vos://cds-vospace.com!vospace/folder" version="2.1" xsi:type="vos:ContainerNode">
  <vos:properties>
    <property uri="ivo://ivoa.net/vospace/core#mtime">06/02/2020 03:26:16</property>
    <property uri="ivo://ivoa.net/vospace/core#ctime">06/05/2020 10:56:07</property>
    <property uri="ivo://ivoa.net/vospace/core#mtime">06/02/2020 03:26:16</property>
    <property uri="ivo://ivoa.net/vospace/core#length">177414</property>
    <property uri="ivo://ivoa.net/vospace/core#subject">subject</property>
  </vos:properties>
  <vos:accepts>
    <view uri="ivo://ivoa.net/vospace/core#anyview"/>
  </vos:accepts>
  <vos:nodes>
    <vos:node uri="vos://cds-vospace.com!vospace/folder/anotherFolder" xsi:type="ContainerNode"/>
    <vos:node uri="vos://cds-vospace.com!vospace/folder/test_copy1" xsi:type="DataNode"/>
  </vos:nodes>
</vos:node>
```

□ Déroulement d'une requête complexe

Exemple d'un upload (PushToVOspace)

Début de négociation - 1/4



□ Déroulement d'une requête complexe

Exemple d'un upload (PushToVOspace)

Requête du client :

```
curl -v -X POST -d @pushJob.xml -H "Content-type: text/xml" "http://localhost:8080/vospace/transfers"
```

Avec pushJob.xml :

```
<vos:transfer xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
xmlns:vos="http://www.ivoa.net/xml/VOspace/v2.0" version="2.1">
<vos:target>vos://cds-vospace.com!vospace/mydata2</vos:target>
<vos:direction>pushToVoSpace</vos:direction>
<vos:view uri="ivo://ivoa.net/vospace/core#anyview"/>
<vos:protocol uri="ivo://ivoa.net/vospace/core#httpput" />
</vos:transfer>
```

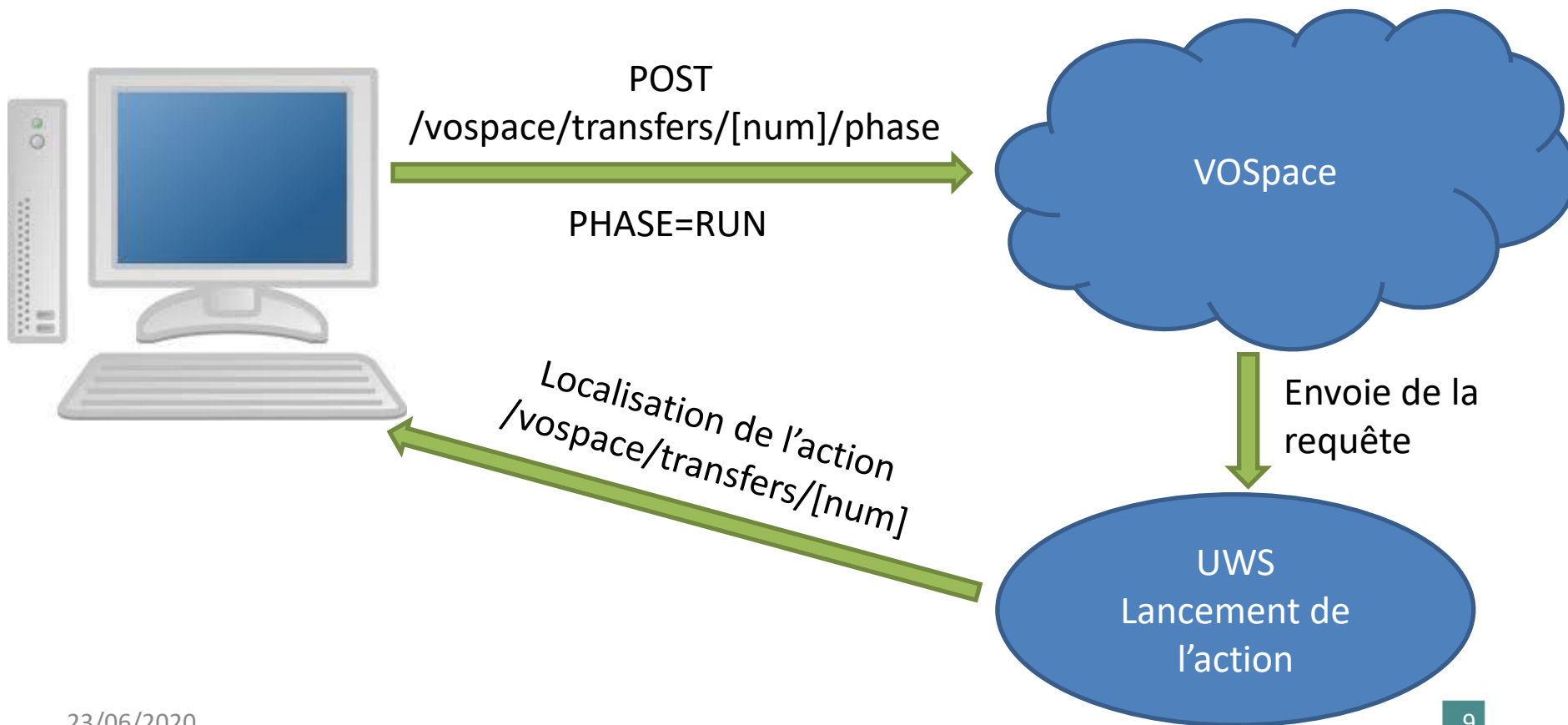
Réponse du serveur :

```
Location: http://localhost:8080//vospace/transfers/1592288902584
```


□ Déroulement d'une requête complexe

Exemple d'un upload (PushToVOspace)

Lancement de l'action - 2/4



□ Déroulement d'une requête complexe

Exemple d'un upload (PushToVOspace)

Requête du client :

```
curl -v -d "PHASE=RUN" "http://localhost:8080/vospace/transfers/1592288902584/phase"
```

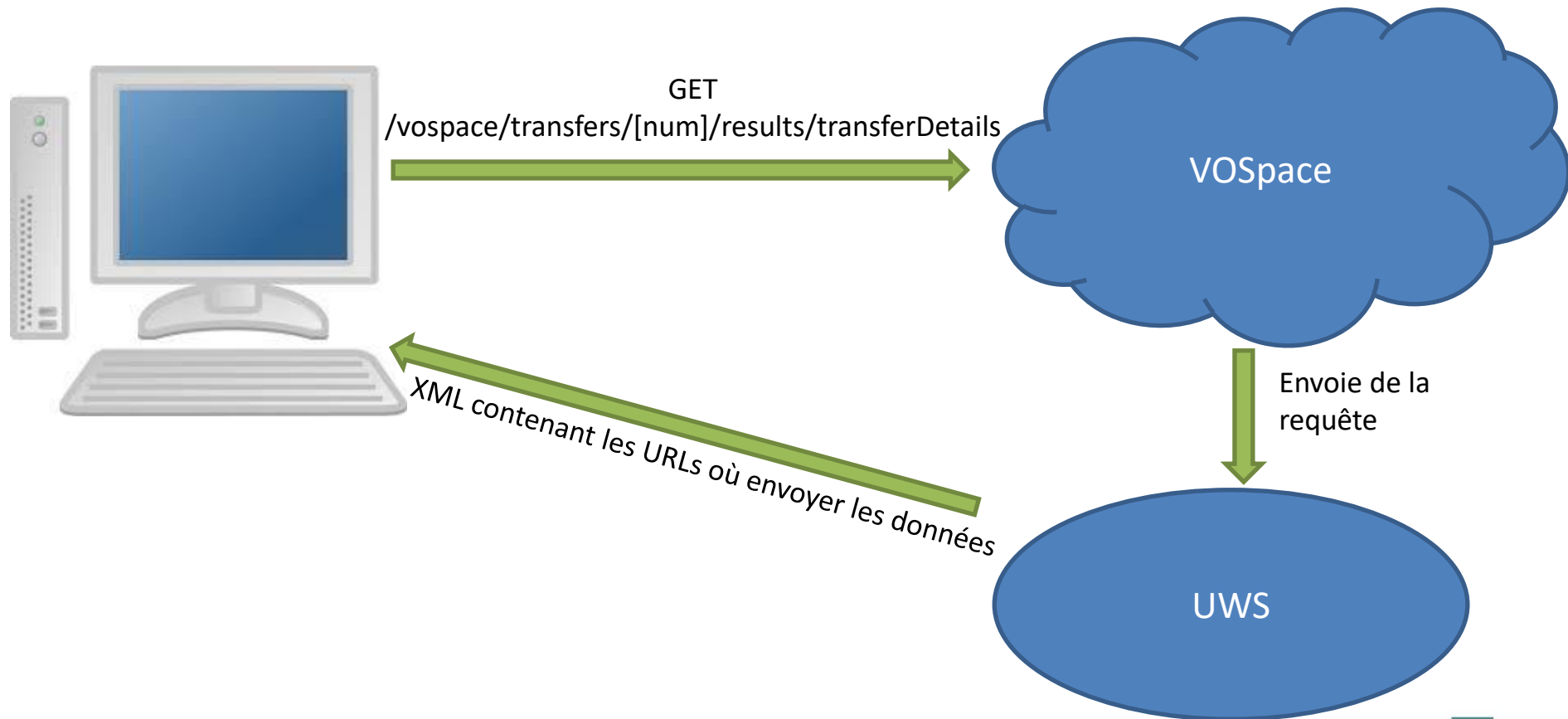
Réponse du serveur :

```
Location: http://localhost:8080//vospace/transfers/1592288902584
```

□ Déroulement d'une requête complexe

Exemple d'un upload (PushToVOspace)

Fin de négociation - 3/4



□ Déroulement d'une requête complexe

Exemple d'un upload (PushToVOSpace)

Requête du client :

```
curl -v "http://localhost:8080/vospace/transfers/1592288902584/results/transferDetails"
```

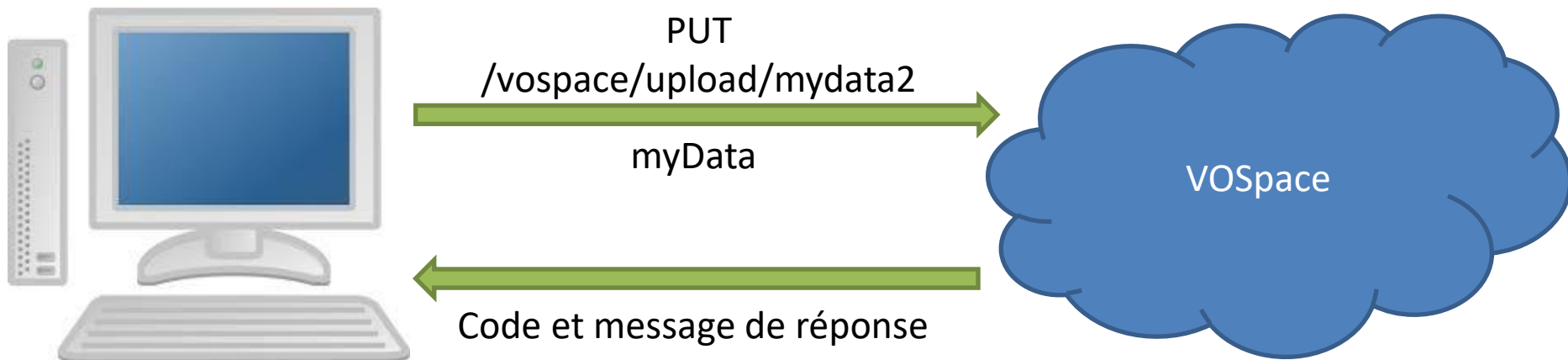
Réponse du serveur :

```
<vos:transfer xmlns:vos="http://www.ivoa.net/xml/VOSpace/v2.0" version="2.1">  
  <vos:target>vos://cds-vospace.com!vospace/mydata2</vos:target>  
  <vos:direction>pushToVoSpace</vos:direction>  
  <vos:protocol uri="ivo://ivoa.net/vospace/core#httpput">  
    <vos:endpoint>http://localhost:8080//vospace/upload/mydata2</vos:endpoint>  
  </vos:protocol>  
</vos:transfer>
```

□ Déroulement d'une requête complexe

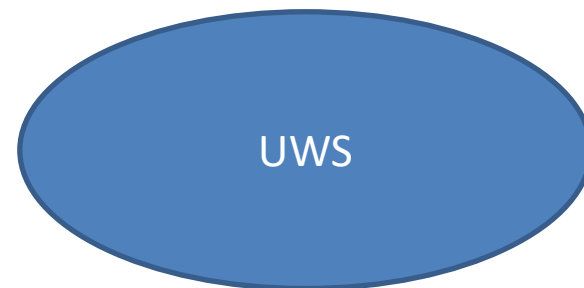
Exemple d'un upload (PushToVOSpace)

Upload des données - 4/4



Requête du client :

```
curl -v -T myData "http://localhost:8080//vospace/upload/mydata2"
```



□ Le standard VOSpace

RESTful et Micro-service

RESTful :

- Réponses formatées
- Responsabilité partagée entre le serveur et le client
- Sans état
- Interface uniforme

Micro-service :

- Service petit et léger
- Faible couplage entre les services de l'application.

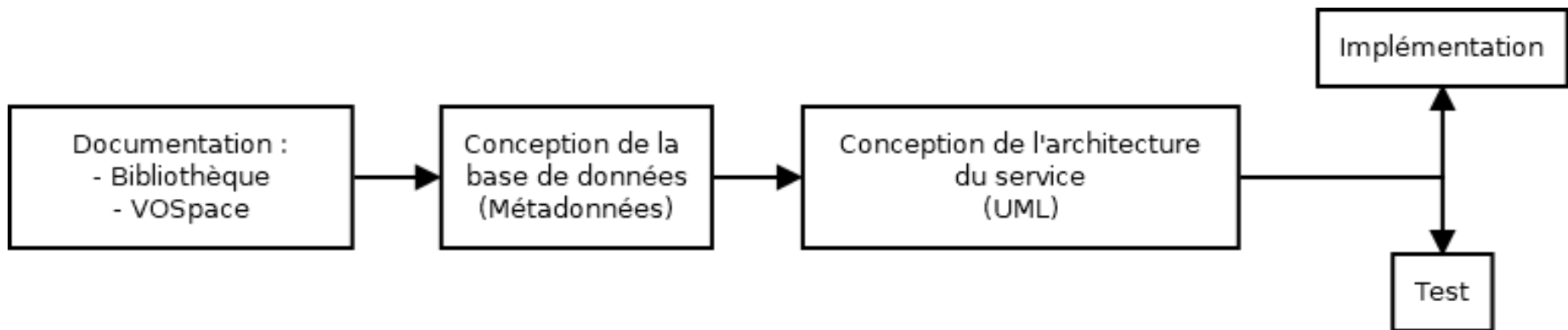
□ Bibliothèques et outils utilisés

- Vert.x : Gestion du côté API HTTP
<https://vertx.io/>
- H2 : Gestion de la base de données
<https://www.h2database.com/html/main.html>
- Gradle : Gestion des bibliothèques, plugins et ressources/dépendances, de la compilation et de l'exécution des tests unitaires
<https://gradle.org/>
- UWS-Lib : Manipulation asynchrone des données
<http://cdsportal.u-strasbg.fr/uwstuto/>

The logo for Vert.x, featuring the word "VERT.X" in a bold, sans-serif font. The "X" is stylized with a purple dot and a purple stroke.

□ L'implémentation du service

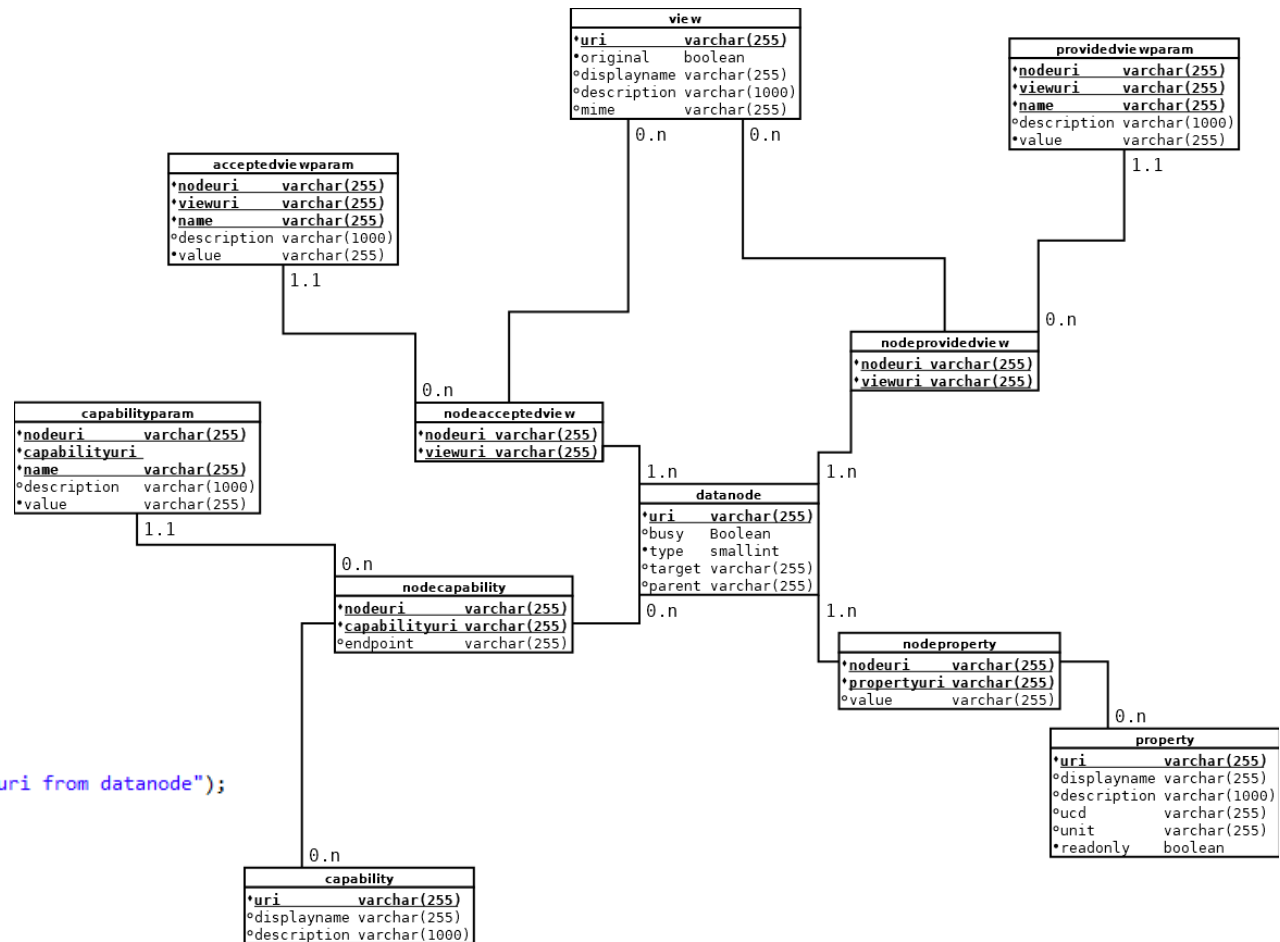
Étapes du stage



L'implémentation du service

La base de données (Métadonnées)

- Requêtes préparées
- Pool de connections
- Try-with-resources

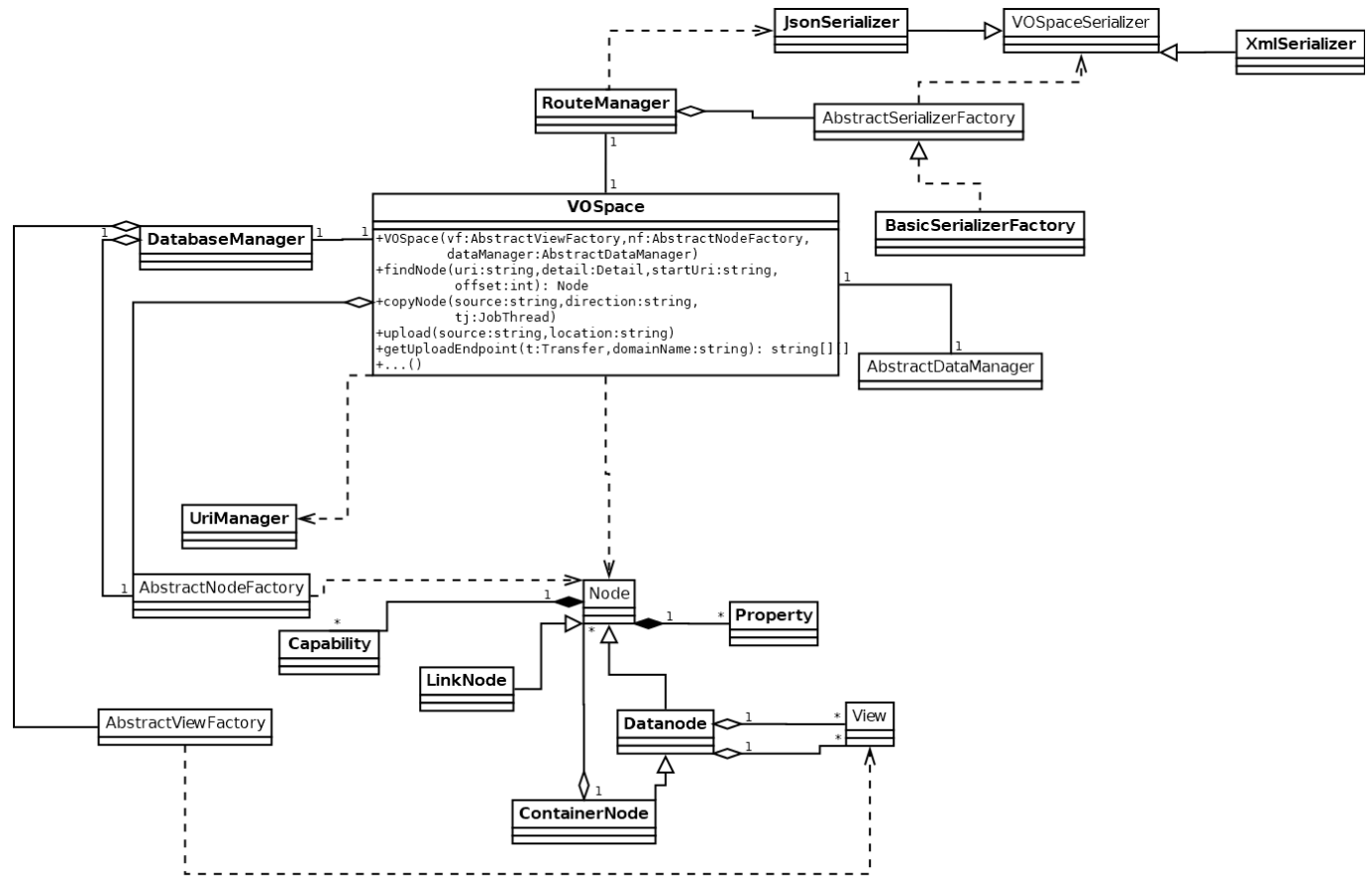


```
try(
    Connection conn = cp.getConnection();
    Statement stmt = conn.createStatement();
    ResultSet rs = stmt.executeQuery("select uri from datanode");
){
    //traitement des resultats
}
catch(Exception e) {
    //gestion des erreurs
}
```

□ L'implémentation du service

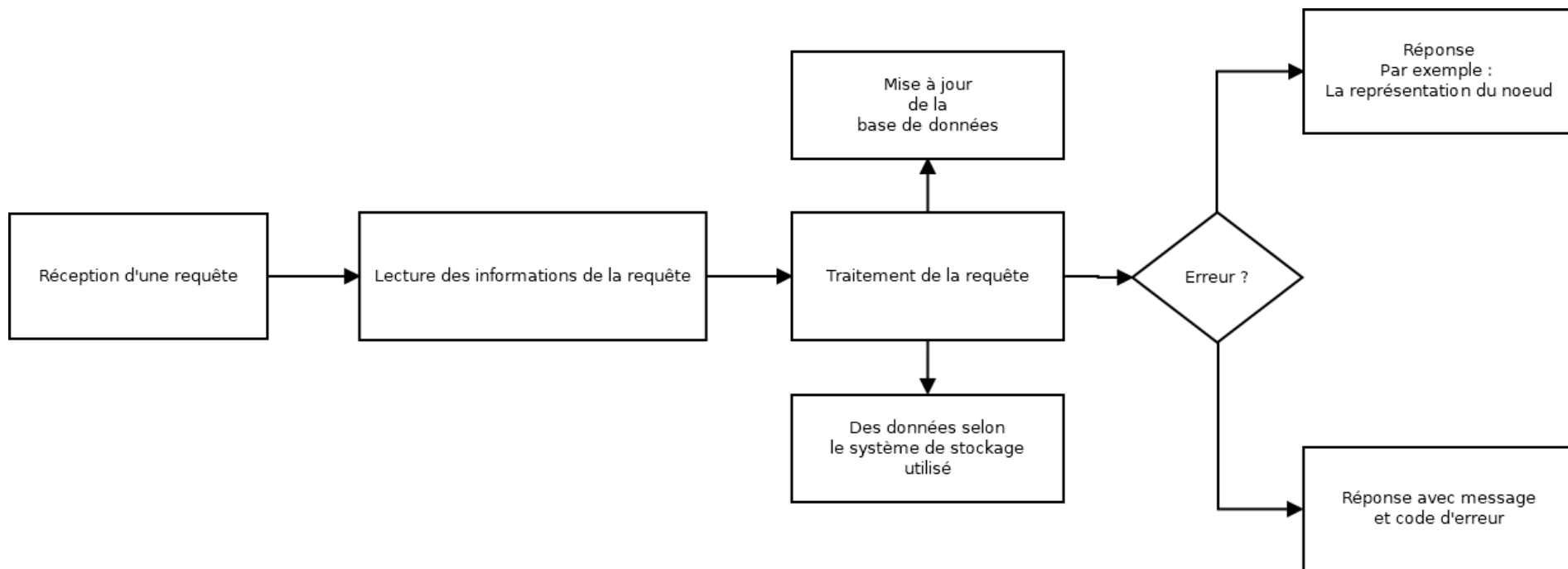
L'architecture du service

- Générique
- Extensible



□ L'implémentation du service

Traitement d'une requête



□ Fonctionnalités

Métadonnées du service :

- GetProtocols ■
- GetViews ■
- GetProperties ■

Métadonnées des nœuds :

- GetNode ■
- SetNode ■

Création et manipulation des nœuds :

- CreateNode ■
- MoveNode ■
- CopyNode ■
- DeleteNode ■

Transferts de données:

- PushToVoSpace ■
- PullToVoSpace ■
- PullFromVoSpace ■
- PushFromVoSpace ■

□ Perspectives futures

- Authentification et gestion de permissions



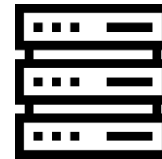
- Interaction avec des services et des outils déjà existants (ex : recherche/visualisation d'images ou de spectres, console SQL)



- Gestion des vues (formats) de download/upload



- Support d'architectures de stockage différentes



- Ajout d'un système de corbeille lors de la suppression d'un nœud



- ...

Perspectives futures

VOSpace Explorer

HOME / tables



hip2.b64
output.vot
sample.vot
sample2.vot
vizier_votable.vot

.../tables/vizier_votable.vot

URI: vos://cds-vospace.com/vospace/tables/vizier_votable.vot

Type: file

Properties:

- Created: Jun 19, 2020 5:56 PM
- Metadata: Jun 19, 2020 5:56 PM
- Modified: Jun 19, 2020 5:56 PM
- Size: 27754 byte

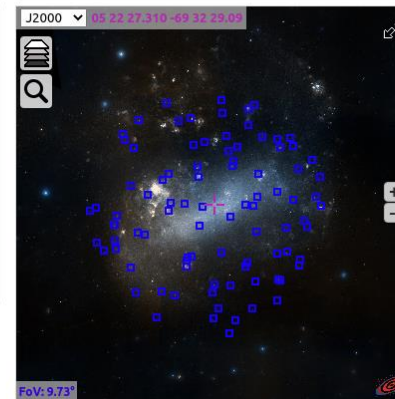
Accepted views:

- anyview

```
adql=# SELECT TOP 10 * FROM "vizier_votable.vot"
```

_RAJ2000	_DEJ2000	HIP	n_HIP	Sn	So	RArad	DErad	Plx	e_Plx	pmRA
73.1006289471	-69.4039212633	22656		5	0	73.10066045	-69.40394251	5.29	0.56	-4.56
73.4303501742	-68.7148172927	22758		5	1	73.43033745	-68.71480076	-3.77	1.77	1.9
73.5594054893	-69.2101228178	22794		5	0	73.55938707	-69.21012223	1.37	1.18	2.69
73.8852403695	-69.9625190658	22900		5	0	73.88523129	-69.96252329	0.22	1.34	1.28
74.0544286937	-68.323260031	22945		5	0	74.05444975	-68.32320026	2.11	0.71	-3.2
74.1547610294	-70.7821964137	22974		5	0	74.15466563	-70.78229398	3.83	0.83	12.92

e.g. help, |l, SELECT TOP 10 * FROM "myTable.vot" ; t or i for historic nav.



□ Conclusion personnelle

- Utilisation de nouvelles bibliothèques et outils
- Expérience de télétravail
- Application de concepts vus en cours (ex : design pattern, test unitaire)
- Communication avec une base de données en Java
- Git/Gitlab