

Créer son Service BioMoby @ LIPM

Sébastien Carrere - carrere@toulouse.inra.fr

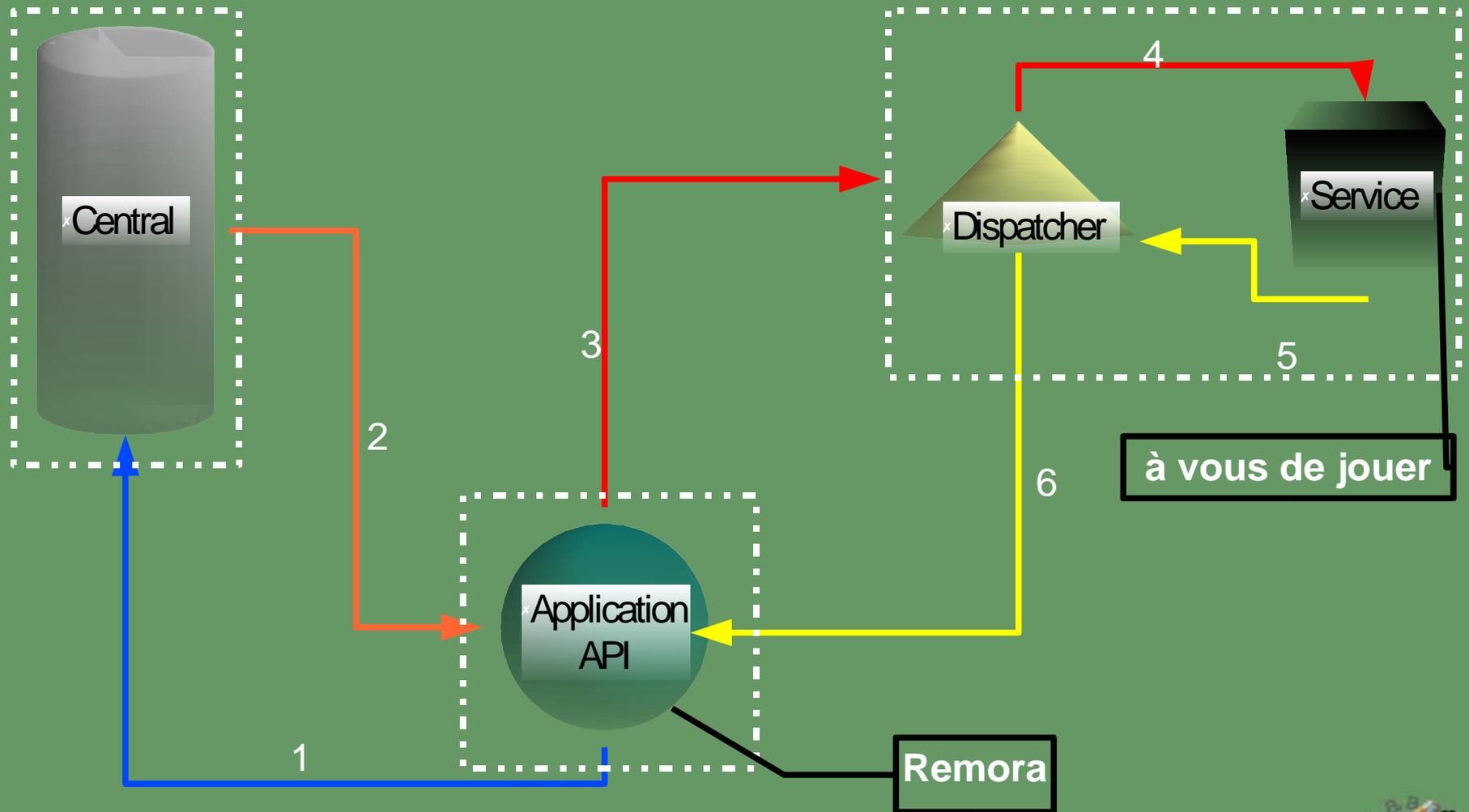
12 Mai 2005



Plan

- **1) Rappels sur le fonctionnement de BioMoby**
 - le Central, le Dispatcher et le Web-service
 - les Articles Primaires et les Paramètres
 - les NameSpaces, le typage des Objets et services
- **2) Principe de fonctionnement d'un Web-Service BioMoby**
 - à quoi ressemble un message XML:BioMoby
 - comment est structuré webservice BioMoby écrit en Perl
 - récupération des données
 - traitement & retour du message
- **3) TP**
 - [création] des objets/namespaces/serviceTypes nécessaires
 - écriture du fichier de configuration du Web-Service
 - écriture de la sub invoquée
 - enregistrement du service
 - test / debugage

le Central, le Dispatcher et le Web-service



les Articles Primaires et les Paramètres

- **Les articles primaires**

- Input / Output
- dans le cas général: OBLIGATOIRES
- 2 types: Simple objet / Collection d'objets (Homogène ou pas)

- **Les articles secondaires**

- Paramètres
- Optionnels
- Types prédéfinis String, Integer, Float, DateTime
- méthodes Enum, Min, Max , Default

les NameSpaces, le typage des Objets et services

- Un objet minimal peut être défini par un ID et un Namespace
 - exple: ID=PD000001 Namespace=ProDom_AC
- Mais on peut vouloir passer autre chose que des *Objets*
 - typage des données (ontologie) Registry
 - permet l'interopérabilité entre services (workflow)
 - Namespace Aware
- De même on peut typer les services (Parsing, Analyse, Retrieval ...)



à quoi ressemble un message XML::BioMoby

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" SOAP- ...">
<SOAP-ENV:Body>
  <namespace3:Multalin xmlns:namespace3="http://biomoby.org/">
    <body>
      <![CDATA[
        <?xml version='1.0' encoding='UTF-8'?>
        <moby:MOBY xmlns:moby='http://www.biomoby.org/moby-s'>
          <mobv:mobvContent>
            <moby:mobyData queryID='1'>
              <moby:Simple moby:articleName="">
                <FASTA_AA_multi>&gt;SMc02591_AA-Y02591
                MSVPASSRERKSYWISLVSLLAAVPLAVLVGSRGEFAAWLQRRMEPPLTV
                VVELFLVPRQADGFTLSLALTGSPILLK
                &gt;SMc04141_AA-gst9
                LSLAIFPVLVLYVIFSRQLIRGITAGAVK
                </FASTA_AA_multi>
              </moby:Simple>
              <moby:Parameter moby:articleName='gapcost'>
                <Value>5</Value>
              </moby:Parameter>
            </moby:mobyData>
          </moby:mobvContent>
        </moby:MOBY>
      ]]>
    </body>
  </namespace3:Multalin>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```



comment est structuré un web service BioMoby écrit en Perl

```
sub MonWebService
```

```
{
```

```
    my ($caller, $message) = @_;  
    #Recuperation de la liste des requetes dans le message
```

```
    foreach my $query (@a_queries)
```

```
    {
```

```
        #recuperation du numero de la requete  
        #recuperation des articles
```

```
        foreach my $input_article (@a_input_articles)
```

```
        {
```

```
            my ($article_name, $article) = @{$input_article};
```

```
            # Recuperation des input  
            # Recuperation des parametres
```

```
        }
```

```
        #Ecriture des fichiers temporaires de données  
        #Execution du traitement
```

```
        #Ajout du resultat au message de reponse du Webservice
```

```
    }
```

```
    #Retour du message
```

```
}
```



récupération des données (i)

#Recuperation de la liste des requetes dans le message

```
my @a_queries = getInputs($message);
```

```
foreach my $query (@a_queries)
```

```
{
```

```
    #recuperation du numero de la requete
```

```
    my $query_id = getInputID($query);
```

```
    #recuperation de la liste des articles
```

```
    my @a_input_articles = getArticles($query)
```

```
    foreach my $input_article (@a_input_articles)
```

```
    {
```

```
        my ($article_name, $article) = @{$input_article};
```

```
        # Recuperation des input
```

```
        if (isSimpleArticle($article))
```

```
        {
```

```
            #Recuperation d'un article simple
```

```
        }
```

```
        elsif (isCollectionArticle($article))
```

```
        {
```

```
            #Recuperation d'un article collection
```

```
        }
```

```
        # Recuperation des paramètres
```

```
        elsif (isSecondaryArticle($article))
```

```
        {
```

```
            getNodeContentWithArticle($query, 'Parameter', $article_name);
```

```
        }
```

```
    }
```

récupération des données (ii)

#Recuperation d'un article simple

```
if (isSimpleArticle($article))
{
    foreach my $subnode ($article->childNodes)
    {
        #si plusieurs input Simple
        #recuperer la bonne donnée dans la bonne variable
        my $isa_test = &_IsTheCorrectObject($subnode->nodeName, 'FASTA');
        if ($isa_test)
        {
            $fasta_data = &_getNodeContent($article);
        }
    }
}
```

#Recuperation d'une Collection

```
elseif (isCollectionArticle($article))
{
    #Meme chose mais une etape prealable pour eclater la collection
}
```

traitement & retour du message (i)

#Ecriture des fichiers temporaires de données

!! parfois il faut nettoyer les données passées (lignes vides par exple)

#Execution du traitement

à partir des données écrites en dur en temporaire

#Ajout du résultat au message de reponse du Webservice

1. `cat` du résultat dans un variable
2. XMLisation si nécessaire (encodage des images en base 64)
3. `$reponse .= simpleResponse("<output_type1>$result_data</output_type1>", $output_name1, $query_id);`

ET ENFIN ...

#Retour du message

```
return SOAP::Data->type('base64' => (responseHeader() . $moby_response .responseFooter()));
```

NB: Un service renvoie toujours un message meme vide ! (gestion d'erreur)

proposition d'organisation

```
atg:/home/dupont/  
    /Biomoby/  
    /Biomoby/dupont.pm  
        /MonWebservice#1/  
            /default.cfg  
        /MonWebservice#2/  
            /default.cfg  
    /bin/  
    /data/  
    /lib/  
        /ParamParser.pm  
        /biomoby_utils.pl  
    /cfg/  
        /atg_cmd.cfg  
        /Central/  
            /mobycentral.cfg  
            /genopole.cfg  
        /Dispatcher/  
            /dispatcher.cfg
```

<http://lipm-svn.toulouse.inra.fr/svn>



étapes

1. Quelles sont les entrées/sorties de mon service

2. Existent-t-elles dans l'annuaire local ?

-> oui: tant mieux !

-> non: Existent-t-elles dans l'annuaire canadien ?

-> oui: on les crée dans l'annuaire local

-> non: on les crée dans les 2 annuaires

(d'abord en local, et dans le canadien si on veut publier le service)

Registry

RegisterObject

3. idem pour le type de service

4. idem pour les NameSpace

5. Ecriture du fichier de configuration du web service

Service.cfg

6. Ecriture de la sub du web service

webservice.pm

7. Enregistrement du Web Service

RegisterService

8. Debug

DebugWS



URL

<http://genopole.toulouse.inra.fr/bioinfo/services/biomoby/formations/LIPM/html/index.html>