



Un environnement de développement et de production de web-services BioMOBY

Carrere S., Letort S. et Gouzy J.



Notre conception d'un web-service

Un web-service est l'encapsulation d'un programme déjà existant.

Ce programme manipule des fichiers en entrée et sortie
(~~STDIN & STDOUT~~)

- > on peut toujours utiliser ces programmes en ligne de commande
- > on peut les encapsuler via d'autres technologies (CGI, Mobylye)



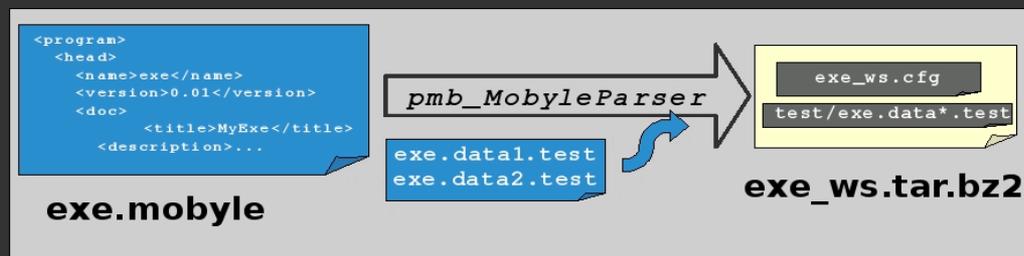
3 étapes

1. Génération d'un fichier de description [Moby XML]

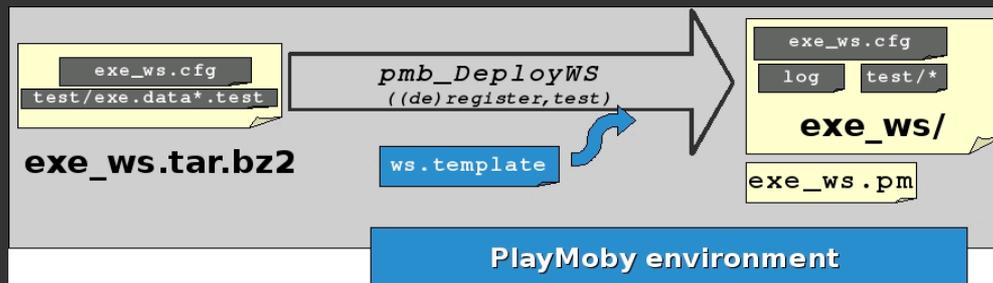
- Moby: C.Letondal *et al* , successeur de Pise;
grosse collection de descriptions d'applications
- Appli.pm: un module pour générer ces fichiers XML



2. Génération du web-service à partir de la description XML



3. Enregistrement et test



TP

1. Déployer l'environnement de travail

```
ssh stageXX@migXXX
tssh
wget http://lipm-bioinfo.toulouse.inra.fr/biomoby/playmoby.tar.bz2
tar xvfj playmoby.tar.bz2
mv playmoby playmobyXXX
cd playmobyXXX
cat README
./pmb_configure.pl
```



TP

2. Décrire un service

- choisir un programme a interfacier
(si pas d'idée voir dans .../playmoby/sample)
- décrire le programme via *Appli.pm*
- déployer le web-service dans l'arborescence



TP

3. Enregistrer / Tester / Debugger

Pieges:

- problemes d'écriture des fichiers tmp/log (droits apache ?)
- fichier test incorrect
- dispatcher mal configuré
- nom du package Perl incorrect

4. Surveillance



TP: Récapitulatif des Fichiers créés/modifiés

1. **pmb_MobyleParser.pl** produit l'archive **Webservice.bz2** contenant:
 - le fichier **.cfg** du webservice
 - les données de test
 - un fichier **log** (vide mais avec les bons droits)
2. **pmb_DeployWS.pl --pm**
 - crée le fichier **Webservice.pm** (LE webservice)
3. **pmb_DeployWS.pl --register**
 - crée le fichier **Webservice.pm**
 - crée le fichier **RDF** (**annuaire_webservice.rdf**)
 - ajoute une ligne dans le fichier **cfg/dispatcher/dispatcher.txt**
4. **pmb_DeployWS.pl --deregister**
 - copie le fichier **RDF** (**.bkp**) et vide l'original
 - supprime la ligne dans le fichier **cfg/dispatcher/dispatcher.txt**
5. **pmb_DeployWS.pl --test**
 - crée la version **Moby** des données de test (**XML**)
 - écrit le resultat du **Webservice** sur **STDOUT**



Appli.pm

But :

Avoir une description du programme.

Restituer la description sous différentes formes Mobyly et (plus tard) Usage().



Appli.pm

Comment ?

4 tableaux associatifs

- A. general : les données générales du prog.
- B. inputs : les entrées du prog.
- C. outputs : les sorties du prog.
- D. params : les paramètres du prog.



Appli.pm

```
my $o_appli = New Appli( -general => \%h_general );
```

name	=> 'nemo,	# prog name
descr	=> 'no description.',	# string that describes the prog
authors	=> 'no author',	# authors names coma-separated
doc	=> 'no doc',	# string that documents the prog
cmd	=> undef,	# name of the command
version	=> '0.0',	# version of the prog
category	=> 'Service',	# category Biomoby of the prog
reference	=> [],	# prog. references
doclink	=> []	# urls of online doc.



Appli.pm

```
$appli->SetInputs( %h_inputs );
```

```
'input_name' => {  
  descr           => "une description de l'entrée.",  
  namespace       => 'Narcisse',  
  type_biomoby    => 'List_Text',  
  datatype        => 'string',  
  cmd             => '--locus=$value'  
}
```

```
$appli->SetOutputs( %h_outputs );
```

```
'output_name' => {  
  descr           => "une description de la sortie.",  
  namespace       => 'Narcisse',  
  type_biomoby    => 'FASTA',  
  datatype        => 'string',  
  cmd             => '--seq_out=$value'  
}
```



Appli.pm

```
$appli->SetParams( %h_params );
```

```
'option' => {  
  descr           => "un parametre pour optimiser.",  
  type_biomoby    => 'Integer',  
  datatype        => 'int',  
  cmd             => '--opt=$value',  
  default         =>      51,  
  min             => 0,  
  max             => 100  
}
```



Appli.pm

Et après ?

`$appli->GetMobyXml($output_dir);` #Génère la description Moby

`$appli->GetPlaymobyTestXml();` #Génère le fichier pmb.t par défaut

⊕ <http://lipm-bioinfo.toulouse.inra.fr/biomoby/playmoby/doc/Appli.html>



Les tests (i)

Par défaut

Comparaison littérale avec une donnée de référence:

1. soit le résultat de la dernière exécution de `pmb_Deploy.pl -test`
2. si pas de référence, lors du premier appel au script de surveillance
 - WARNING
 - sauvegarde du résultat comme référence

MonWS.pmb.t

Fichier XML (cf. Appli.pm) contenant:

- Pour chaque article de sortie
 - un test sur le format des données
 - un test sur le contenu



Les tests (ii)

Surveillance

pmb_CheckWS.pl

Recherche tous les WS deployés dans nue arborescence donnée et les teste

--nomail : n'enverra que le mail de rapport à l'admin de l'instance PlayMOBY

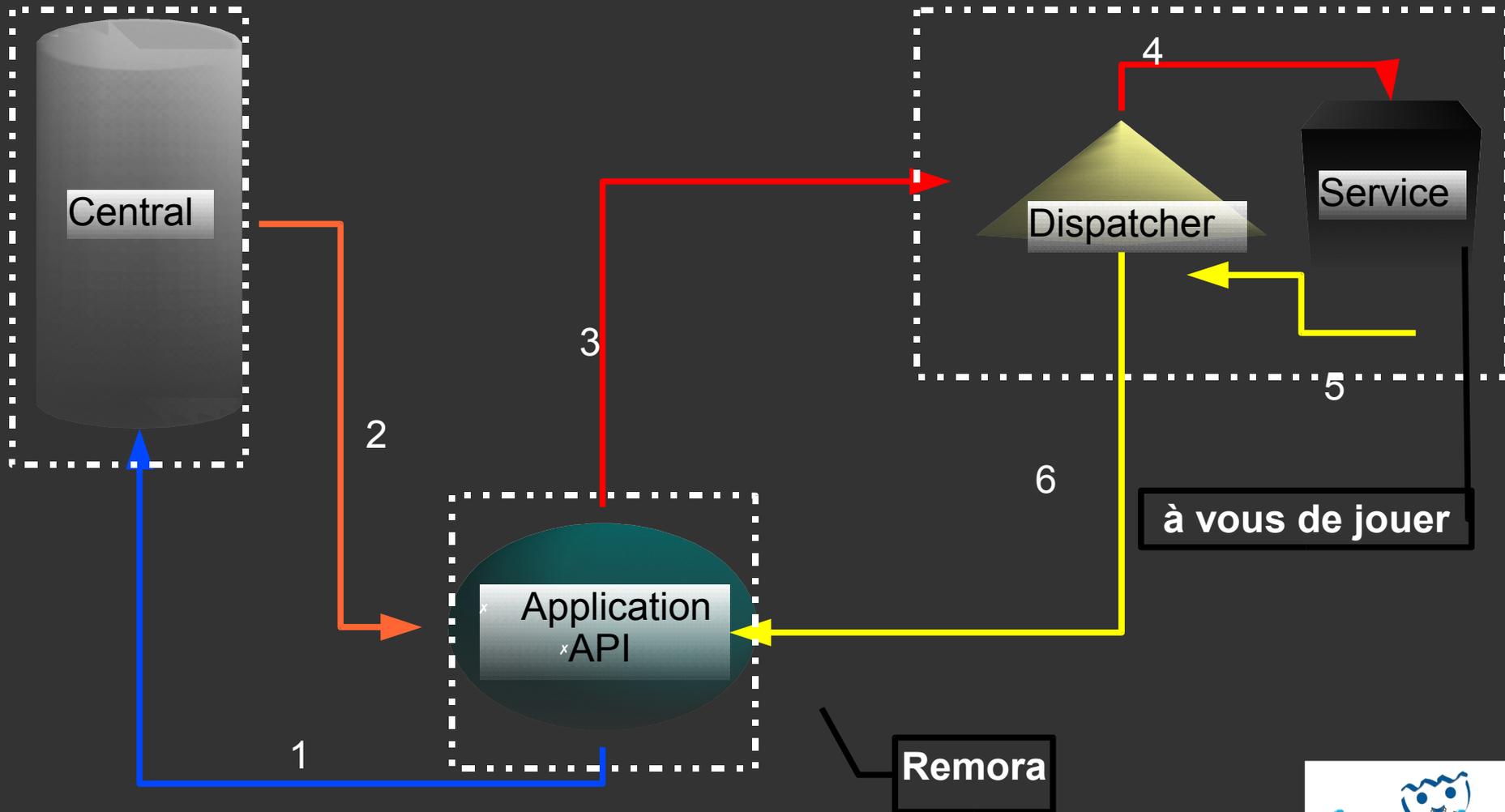
--bioworkflow : genere les rapports BioWorkFlow ainsi que le RSS

Rapport PlayMOBY

BlastToLocus_tags	2008.05.28 02:50:51	Execution time: 1 s.					
BioMOBY	<ul style="list-style-type: none">URI: iant.toulouse.inra.frRegistry: http://moby.ucalgary.ca/moby/MOBY-Central.plRDF: http://lipm-bioinfo.toulouse.inra.fr/biomoby/analysis/iant/rdf/parse/BlastToLocus_tags.rdfContact: sebastien.carrere@toulouse.inra.fr						
NarcisseLocus_tags_list	<table border="1"><tr><td>Format</td><td><input type="checkbox"/> Show perl code used to test <i>NarcisseLocus_tags_list</i> format ok</td></tr><tr><td>Content</td><td><input type="checkbox"/> Show perl code used to test <i>NarcisseLocus_tags_list</i> content ok</td></tr></table>	Format	<input type="checkbox"/> Show perl code used to test <i>NarcisseLocus_tags_list</i> format ok	Content	<input type="checkbox"/> Show perl code used to test <i>NarcisseLocus_tags_list</i> content ok		
Format	<input type="checkbox"/> Show perl code used to test <i>NarcisseLocus_tags_list</i> format ok						
Content	<input type="checkbox"/> Show perl code used to test <i>NarcisseLocus_tags_list</i> content ok						



le Central, le Dispatcher et le Web-service



Enregistrement d'un web-service

Parametres:

Description du service: nom, description, URI d'authentification

Interfaces: entrees / sorties / parametres

Acces: signature url (URL RDF), dispatcher url

Script:

```
pmb_RegisterService.pl -central_conf <central.cfg>  
-service_conf <default.cfg>  
-module <services.pm>  
[-special_conf]
```

Retour:

if success: generation du fichier RDF
 ajout au fichier dispatcher



Suppression d'un web-service

Parametres:

Acces: signature url (Url du RDF)

Script:

```
pmb_DeregisterService.pl      -central_conf <central.cfg>  
                              -service_conf <default.cfg>  
                              --remove
```

Retour:

**if success: “vidage” du fichier RDF
 suppression dans fichier dispatcher**

