



# Journée OV France GWS

1-2 février 2006

## Web Services : coté Serveur



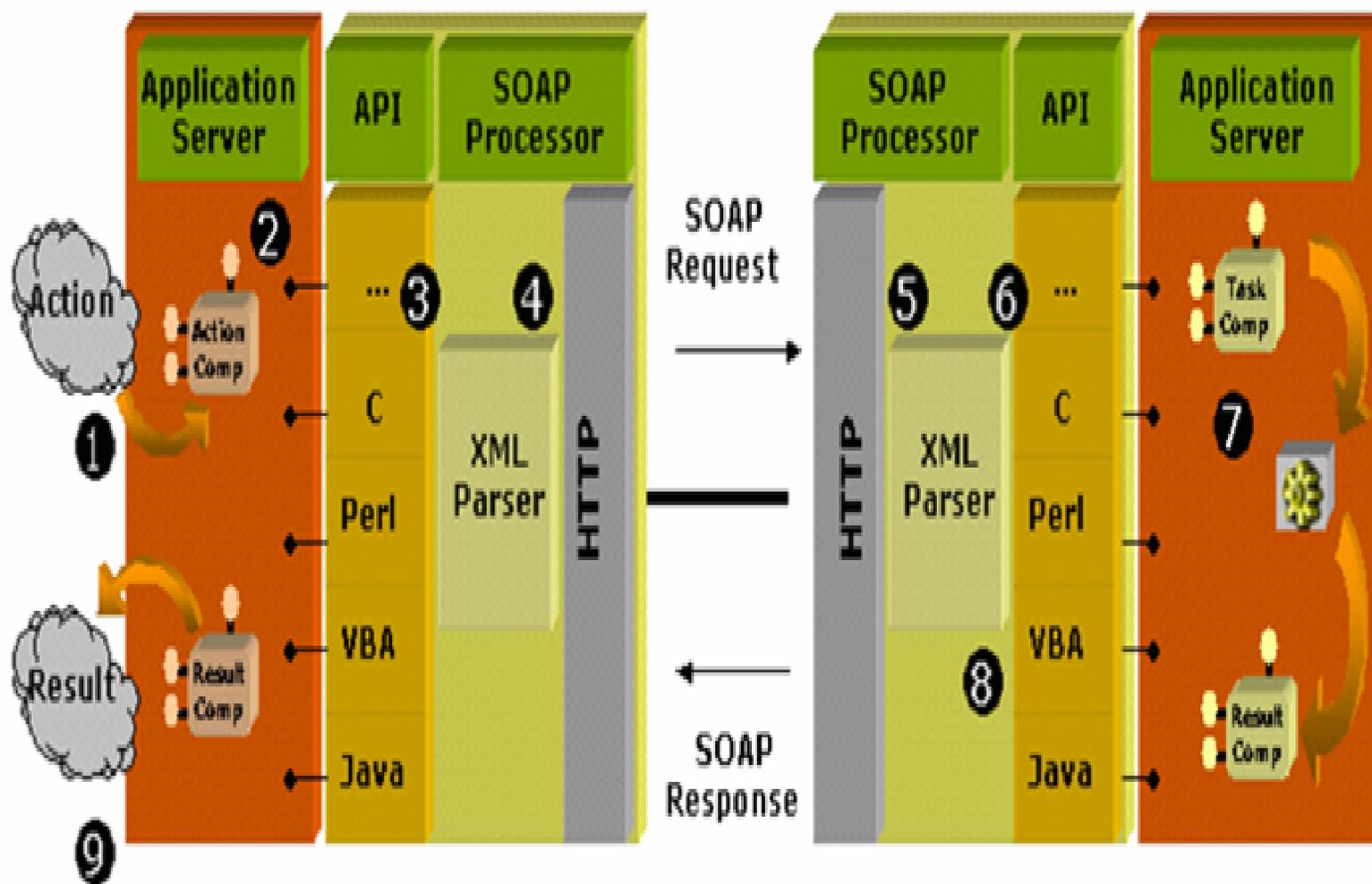
André Schaaff – Journée OV France GWS  
Strasbourg – 1er et 2 février 2006

# Avant propos

- **Non traité d'un point de vue pratique dans le cadre du tutorial OVF consacré aux Standards**
- **Certains acteurs seront uniquement consommateurs de services et se consacreront uniquement à l'aspect client**
- **D'autres acteurs ont déjà travaillé sur la partie serveur, d'autres souhaitent le faire**
  - **Comment procéder ? Quels outils ? Quels standards ?  
Quelle(s) limite(s) ?**



# Rappel



# Un Web Service pour quoi faire ?

- **Il faut se poser certaines questions :**
  - Qui va utiliser ce service ? Y a-t-il vraiment de la demande ?
  - Quelle sera la charge approximative ?
  - ...
  
- **Une fois le Web Service disponible**
  - Il faut le maintenir
  - Assurer la disponibilité : ne pas oublier qu'un Web Service se « consomme » comme un simple composant
  
- **Au CDS**
  - Name resolver : plus d'un million d'accès par an
  - Les autres sont utilisés par quelques personnes pour des applications très ciblées



# Outils

- **Java : Tomcat-Axis, il existe une implémentation C++ d'Axis**
- **Fortran : gSOAP, « The compiler enables the integration of (legacy) C/C++ and Fortran codes (through a Fortran to C interface) »**
- **.NET pour C#, etc.**
- **Perl : SOAP::Lite, ...**
- **Python : SOAPy (plus de maintenance), Zolera SOAP Infrastructure, etc.**
- **Au niveau de l'IVOA : .NET (principalement NVO) et Tomcat/Axis**



# Corrélation WSDL – code source

## ■ Exemple simple

- WSDL [Sesame](#) – Source Sesame
- N.B. : Sesame est un exemple très simple, ce service retourne uniquement une chaîne de caractères contenant le résultat

## ■ Exemple plus compliqué

- WSDL [VizieRBeta](#) – Source VizieRBeta
- N.B. : cet exemple est plus compliqué car des objets sont passés en paramètres

■ Nous verrons d'autres exemples demain...



# Contraintes liées aux spécifications IVOA

- Il y a essentiellement deux documents à l'heure actuelle qui devrait passer en recommandation dans les mois à venir :
  - VO Support Interfaces
  - VO Web Services Basic Profile
  
- D'autres travaux sont en cours mais devraient nécessiter un temps assez long avant d'aboutir au stade de la recommandation
  - Universal Worker Service
  - ...



# VO Support Interfaces

- Le rôle des Web Services est croissant au niveau du VO
- Il est important de décrire les interfaces de base que chaque Web Service devrait fournir
- Exemples
  - getAvailability
    - Dernier redémarrage du service, prochaine date de maintenance, ...
  - HarvestWebLog date de début, date de fin
    - IP des utilisateurs du service, type de requête, temps de réponse, ...
- Dans le détail : <http://www.ivoa.net/internal/IVOA/IvoaGridAndWebServices/VOSupportInterfaces-0.24.pdf>



# VO Web Services Basic Profile

- Un ensemble de règles à prendre en compte lors de l'implémentation de Web Services.
- Considère WS-I Basic Profile comme un minimum et l'étend en y ajoutant des règles spécifiques au VO
  - The getAvailability interface MUST be implemented
  - ...
- Un guide de « bonne implémentation » à l'intention des fournisseurs de services, automatisation des tests de conformité
- Dans le détail : <http://www.ivoa.net/internal/IVOA/IvoaGridAndWebServices/VO-WS-Basic-Profile-0.22.pdf>



# Tâches asynchrones

- Notion de contexte : associer une opération d'un service à une tâche
- Information à propos de l'état d'un contexte
- Manager le cycle de vie des contextes
- Notification des changements au niveau du contexte, exemple : "job completed"
- ...

<http://www.ivoa.net/internal/IVOA/IvoaGridAndWebServices/uws.html>



# Est-ce la bonne approche ?

- **Création de Web Services**
  - Relativement aisée mais pas toujours adaptée
  
- **Alternative : utilisation d'une architecture prenant en charge le wrapping des tâches**
  - Pratique pour des programmes existants dans des langages type Fortran
  - Ajoute la notion de workflow
  - Nécessite très peu de développement
    - En général un fichier de paramétrage
  
- **Cf. présentation AstroGrid et Groupe Workflow OVF**



# Travaux intéressants en cours

- **Mono** : [http://www.mono-project.com/Main\\_Page](http://www.mono-project.com/Main_Page)
- **Axis2** : <http://ws.apache.org/axis2/>



# How To : Installer Tomcat/Axis

- **Comment mettre en œuvre un serveur de Web Services ?**
  - **Pré requis : kit Java (pas seulement le JRE)**
  - **Télécharger Jakarta Tomcat et l'installer**
  - **Télécharger Axis (il existe deux versions : Java et C++)**
  - **Il faut ensuite copier le sous-répertoire axis (exemple pour la version 1.3 d'Axis : axis\_1.3/webapps/axis) vers le sous répertoire webapps de Tomcat.**
    - **Risque d'ambiguïté : on ne copie pas le répertoire axis\_1.3 ....**
  - **Il manque quelques librairies. Explications.**
  - **Comment tester ?**



# Références

## ■ Liens

- Astro-RG, <http://www.ivoa.net/twiki/bin/view/IVOA/AstroRG>
- GWS WG, <http://www.ivoa.net/twiki/bin/view/IVOA/IvoaGridAndWebServices>
- Tutorial ADASS 2003, <http://www.adass.org:8080/meetings/adass2003/events/tutorial>
- GGF, <http://www.gridforum.org/>
- SOAP, <http://www.w3.org/TR/soap/>
- WSDL, <http://www.w3.org/2002/ws/desc/>
- UDDI, <http://www.uddi.org/>
- WS-Security, [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=wss](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss)
- WSIF, <http://ws.apache.org/wsif/index.html>
- ...



# Rappel : Architecture du VO

