



Tutorial OV France

30/1-1/2 2006

Grid, Web Services, SkyNode, ...

IVOA GWS - VOQL



André Schaaff – Tutorial OV France
Strasbourg – 1er février 2006

Web Services

■ Avant propos

- L'échange d'une information structurée et typée entre pairs dans un environnement décentralisé et distribué n'est pas un paradigme récent...
 - **Ces dix dernières années de nombreuses technologies (CORBA, RMI, DCOM, ...) ont été développées dans ce but**
 - **Par contre, par manque de consensus et par manque de simplicité au niveau du déploiement, elles n'ont pas suscité un engouement comparable à celui de technologies comme HTTP, XML, ...**

Terminologie : Web Service, service Web XML, services SOAP, ...



Les Web Services : une révolution ?

- Les Web Services ne sont pas une révolution mais un moyen simple de distribuer et de « consommer » des composants.
- Les acteurs majeurs de l'industrie informatique travaillent à la définition de standards (W3C, OASIS, WS-I et autres consortiums) et à la création d'outils dans ce domaine (Microsoft avec .NET, Sun avec Sun One,...).
- Distinctes au départ, des technologies Web Services et Grid se mêlent maintenant étroitement
- IVOA : les Web Services sont un moyen de ... pas une finalité

http://en.wikipedia.org/wiki/Web_services



Cas d'utilisation simple : Sesame

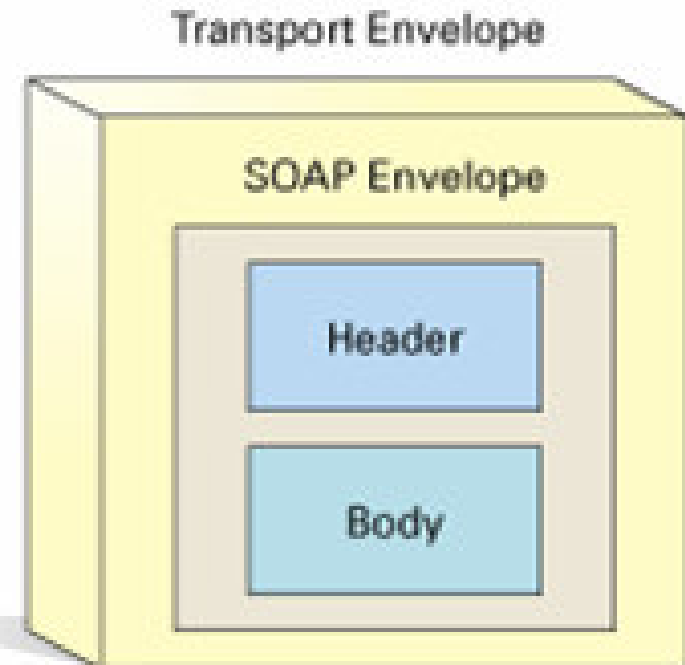
- Sesame est un résolveur de nom existant sous forme de cgi-bin
 - <http://vizier.u-strasbg.fr/viz-bin/nph-sesame/-oX?m31>
- Comment utiliser ce service dans une application ?
 - Utiliser l'URL (en espérant que celle-ci ne change pas) ?
 - Il serait sans doute intéressant de l'utiliser comme un composant quelconque de l'application
 - Un Web Service peut répondre à ce besoin, avec un « overhead » d'information sémantique et avec pour support des technologies standardisées comme HTTP et XML.
 - Un Web Service permet de transmettre des données structurées au client (exemple : un objet avec diverses propriétés plutôt qu'un fichier plat ou un document XML)



La clé de voûte : SOAP

■ Simple Object Access Protocol

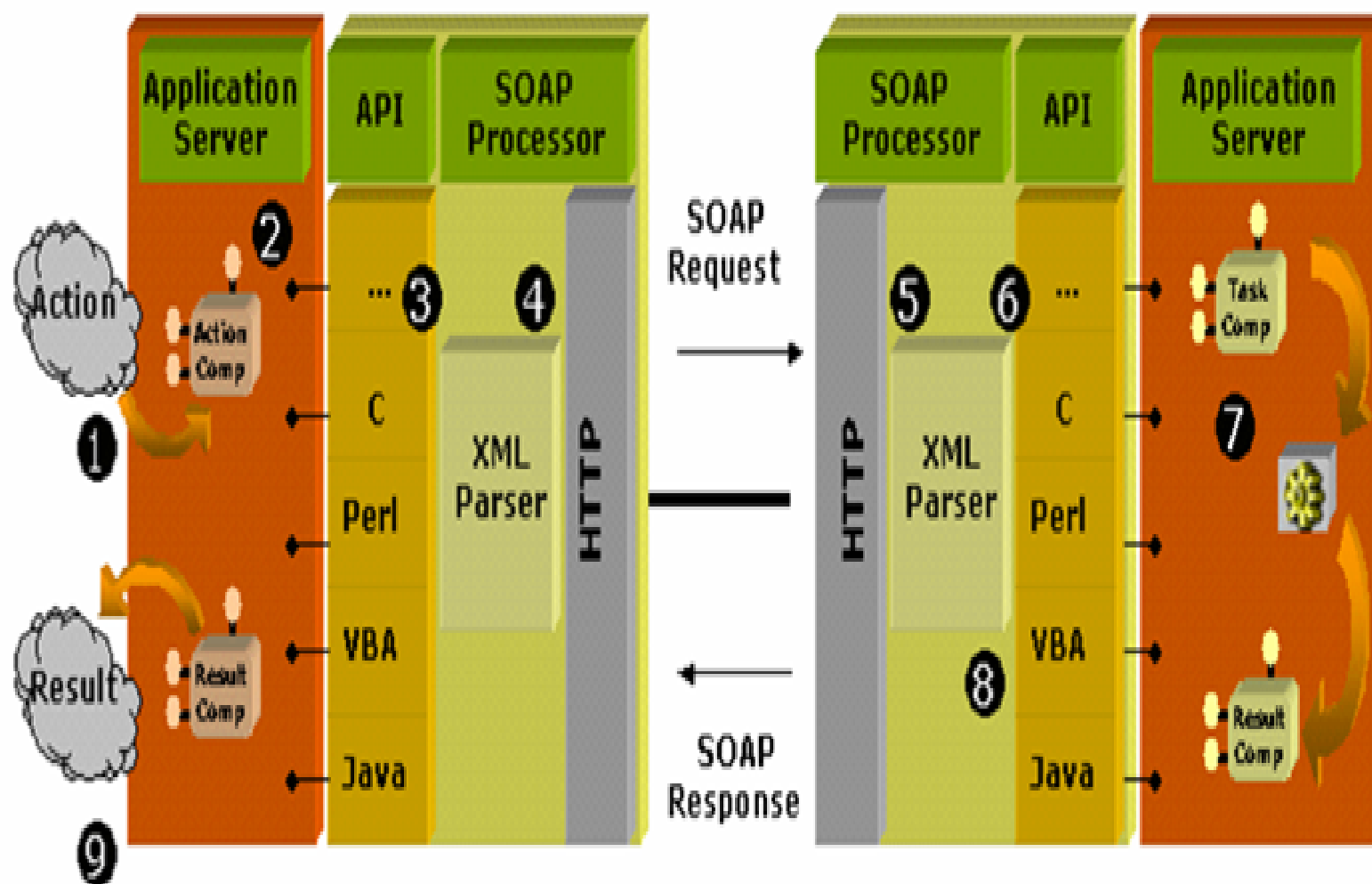
- **But** : fournir un protocole d'échange entre clients et serveurs
- SOAP est basé sur HTTP et XML
- SOAP est une « enveloppe » XML transportée via HTTP



http://en.wikipedia.org/wiki/Simple_Object_Access_Protocol



De la requête à la réponse...



Message SOAP (1)

■ Exemple de requête pour le Web Service Sesame

■ Au niveau d'un client Java :

```
mySesame.Sesame("M51"); // cf. T.P.
```

...

Le message suivant transite du client vers le serveur :

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:apachesoap="http://xml.apache.org/xml-soap" xmlns:impl="http://cdsws.u-strasbg.fr/axis/services/Sesame"
  xmlns:intf="http://cdsws.u-strasbg.fr/axis/services/Sesame" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:wSDL="http://schemas.xmlsoap.org/wSDL/" xmlns:wSDLsoap="http://schemas.xmlsoap.org/wSDL/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <mns:Sesame xmlns:mns="http://cdsws.u-strasbg.fr/axis/services/Sesame"
      SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <name xsi:type="xsd:string">M51</name>
    </mns:Sesame>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```



Message SOAP (2)

■ Exemple de réponse pour le Web Service Sesame

Le serveur reçoit le message SOAP du client, le parse, détermine le service concerné et lui transmet les paramètres. Le service renvoie un résultat au serveur qui l'enveloppe (SOAP) et transmet le message SOAP au client. Celui-ci parse le message et transmet les données au client concerné.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:SesameResponse soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:ns1="http://cdsws.u-strasbg.fr/axis/services/Sesame">
      <SesameReturn xsi:type="xsd:string"># M51 #Q01227 #=Simbad: 1 %J 202.4682083 +47.1946667 (6) = 13 29
52.370 +47 11 40.80 %J.E [10800.00 10800.00 90] D 1999ApJS..125..409C %I.0 APG 85 %C IG %@ 8056,0 #---
ServerTime(ms): 1
      </SesameReturn>
    </ns1:SesameResponse>
  </soapenv:Body>
</soapenv:Envelope>
```



Message SOAP (3)

■ Récapitulatif

■ Au niveau du client :

- Il faut être en mesure de formaliser une requête suivant le format SOAP afin de la soumettre à un serveur « comprenant » SOAP
- Il faut être capable de « parser » le format SOAP afin d'interpréter la réponse du serveur
- ...

■ Au niveau du serveur :

- Il faut être capable de « parser » le format SOAP afin d'interpréter la demande d'un client
- Il faut être capable de formaliser un résultat suivant le format SOAP afin de renvoyer un résultat au client
- ...

■ Pas très compliqué mais fastidieux à mettre en œuvre, il vaut mieux utiliser les outils existants



La description : WSDL

■ Web Service Description Language

- **But** : décrire les services comme un ensemble d'opérations et de messages abstraits relié à des protocoles et des serveurs réseaux, auto description du service
- XML et schéma XML
- Le même rôle que l'OMG-IDL pour les utilisateurs de CORBA
- Un fichier WSDL contient tout ce qu'un client potentiel doit connaître pour pouvoir l'utiliser
 - Il décrit les interfaces, les ports, ...
- Cette description du Web Service permet de générer un « proxy »
 - Un groupe de fichiers dans le langage du client
 - Permet de masquer l'utilisation distante du service
- Exemple : la description du Web Service Sesame (doc)
- <http://en.wikipedia.org/wiki/WSDL>



Le référencement, la découverte, ... : UDDI

■ Universal Description, Discovery and Integration

- **But** : référencement des services dans un annuaire mondial d'entreprises permettant d'automatiser les communications entre prestataires, fourniture d'API d'exploration, ...
- Spécification
 - actuellement version 3.*
 - initialement 300 sociétés
 - très orientée B2B
- Centralisé mais avec réplication sur plusieurs sites

■ Au niveau IVOA

- L'utilisation de UDDI n'a pas été retenue (?)
- Se référer à la présentation consacrée au « Registry »



Rapide comparatif

■ CORBA

- Rapide et fiable
- Développement fastidieux
- Déploiement non trivial
- Sécurisé
- Pas adapté au Web
- En perte de vitesse
- ...

■ RMI

- Java uniquement
- Développement simple
- Déploiement non trivial
- Pas adapté au Web
- Peu utilisé
- ...

■ SOAP

- Basé sur XML
- Développement simple
- Déploiement simple (serveur Web)
- Sécurisation en cours
- Bénéficie d'un large soutien de l'industrie (IBM, Microsoft, Apple, ...)



IVOA Grid and Web Services WG

- **Global Grid Forum, Astro-RG**
- **VO Support Interfaces**
- **VO Web Services Basic Profile**
- **Sécurité**
- **VOSpace/VOStore**
- **Tâches asynchrones**



GGF

■ Global Grid Forum

- 2 réunions par an (~600 participants)
- Des sessions orientées entreprises + des réunions de WG et RG
- Astro-RG initié lors du GGF 10
 - Chairs : Masatoshi Ohishi et Nic Walton
 - **But** : explorer les potentialités des technologies Grid dans le domaine de l'Astronomie
- GGF12 : quelques remarques...
- GGF15 à Boston :
 - VOSpace, VOStore, UWS
- A venir
 - GGF16, Athènes, 13-16 février 2006
 - http://www.ggf.org/gf/event_schedule/index.php?id=131



VO Support Interfaces

- **But** : définir les interfaces standards.
- 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, ...

<http://www.ivoa.net/internal/IVOA/IvoaGridAndWebServices/VOSupportInterfaces-0.24.pdf>



VO Web Services Basic Profile

- **But** : définir un ensemble de règles à prendre en compte lors de l'implémentation de Web Services.
- Considère WS-I Basic Profile (Microsoft, Sun, ...) 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

http://en.wikipedia.org/wiki/WS-I_Basic_Profile



Sécurité

- **But** : définir des mécanismes d'authentification pour le VO
- Exemple : Single Sign On
- Utilisation de signatures numériques mais différentes approches sont possibles
- Authentification par signature numérique des messages SOAP (messages plus inclusion de justificatifs d'identité signés par une autorité de certification)
- Travail basé sur WS-Security (OASIS)

<http://www.ivoa.net/internal/IVOA/IvoaGridAndWebServices/security-architecture-v0.1.html>



VOSpace / VOStore

- **But** : intégrer les différentes technologies qui permettent à des utilisateurs d'accéder à des espaces de « stockage »
 - Exemples : MYDB (NVO), MYSpace (AstroGrid)
cf. schéma « Architecture du VO » en Annexe
- Définition d'interfaces standards afin de permettre la communication entre les différents espaces de travail
- ...

<http://www.ivoa.net/internal/IVOA/IvoaGridAndWebServices/vospace.pdf>



Tâches asynchrones, ...

- **But** : permettre l'exécution de tâches potentiellement très longues
- 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/VOSupportInterfaces-0.24.pdf>



SkyNode

- **But** : définir des nœuds (conteneur de données + interfaces), permettre l'exécution de requêtes, de croisements, ... au travers par exemple d'OpenSkyQuery
- Basé à l'origine sur l'utilisation de SQLServer (+ .Net) mais il est possible d'utiliser d'autres SGBD (exemple MySQL (+ Tomcat/Axis) utilisé lors de l'école d'été NVO à Aspen en septembre 2004)
- Interfaçage via SOAP (interfaces standardisées pour tous les noeuds)
- Mapping SQL-ADQL (Astronomical Data Query Language, XML)
- Une petite démo
 - <http://openskyquery.net/Sky/skysite/>
 - Généralités, exécution de requêtes, croisements, importation de données personnelles, etc.



SkyNode (2)

■ Il existe différents « Kits » permettant de déployer un SkyNode

- NVO - <http://www.openskyquery.net/Sky/SkySite/help/nodehelp.aspx>
- JVO - <http://jvo.nao.ac.jp/download/skynode-toolkit/>
- ESA

■ CDS : en cours pour VizieR

- Exemple : <http://cdswww.u-strasbg.fr/Tycho2BasicSkyNode/services/BasicSkyNode>
- Cela soulève quelques problèmes :
 - VizieR ~ 5000 catalogues
 - SkyNode : prévu au départ pour un seul catalogue
 - Propositions pour la prochaine réunion Interop de Victoria



Conclusion

- Très critiqués au début (basés sur XML, overhead important, ...) les Web Services se banalisent en raison de leur simplicité de mise en œuvre, des investissements lourds des principaux acteurs du marché informatique mais également en raison de l'absence de réelle alternative...
- Des travaux intéressants (et importants) à mener au sein du WG IVOA Grid and Web Services (standardisation, sécurité, GGF (liens STIC et industrie), etc...)
- Le T.P. permet d'aborder l'aspect client, la partie serveur est abordée lors de la journée Web Services OVF



Références

■ Liens

- Astro-RG, <http://www.ivoa.net/twiki/bin/view/IVOA/AstroRG>
- GWS WG, <http://www.ivoa.net/twiki/bin/view/IVOA/IvoaGridAndWebServices>
- VOQL, <http://www.ivoa.net/twiki/bin/view/IVOA/IvoaVOQL>
- 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/>
- Sun Java WS, <http://java.sun.com/webservices/index.jsp>
- UDDI, <http://www.uddi.org/>
- WS-Security, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss
- OMG, <http://www.omg.org>
- ...



Annexe : Architecture du VO

