

# Atelier Web SAMP

*Thomas Boch [CDS]*

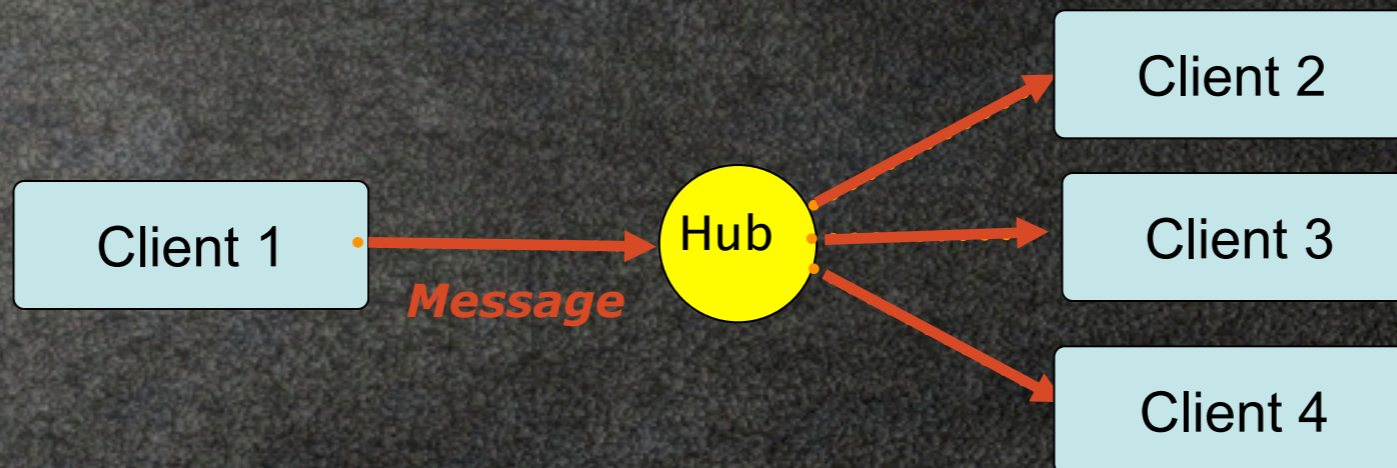
*Observatoire de Paris - 26 juin 2012*

# I. Rappels sur SAMP

# Caractéristiques

- SAMP = Simple Application Messaging Protocol
  - Protocole de communication entre applications de l'OV
    - *exemple : échange de tables entre TOPCAT et Aladin, envoi d'un spectre à VOSpec*
- Multi-plateformes
- Multi-langages
  - profil standard basé sur XML-RPC
  - implémentations clientes en Java, Python, Perl, C, IDL, ...
- Dernière version du standard :  
<http://ivoa.net/Documents/SAMP/20120411/REC-SAMP-1.3-20120411.pdf>

# Architecture

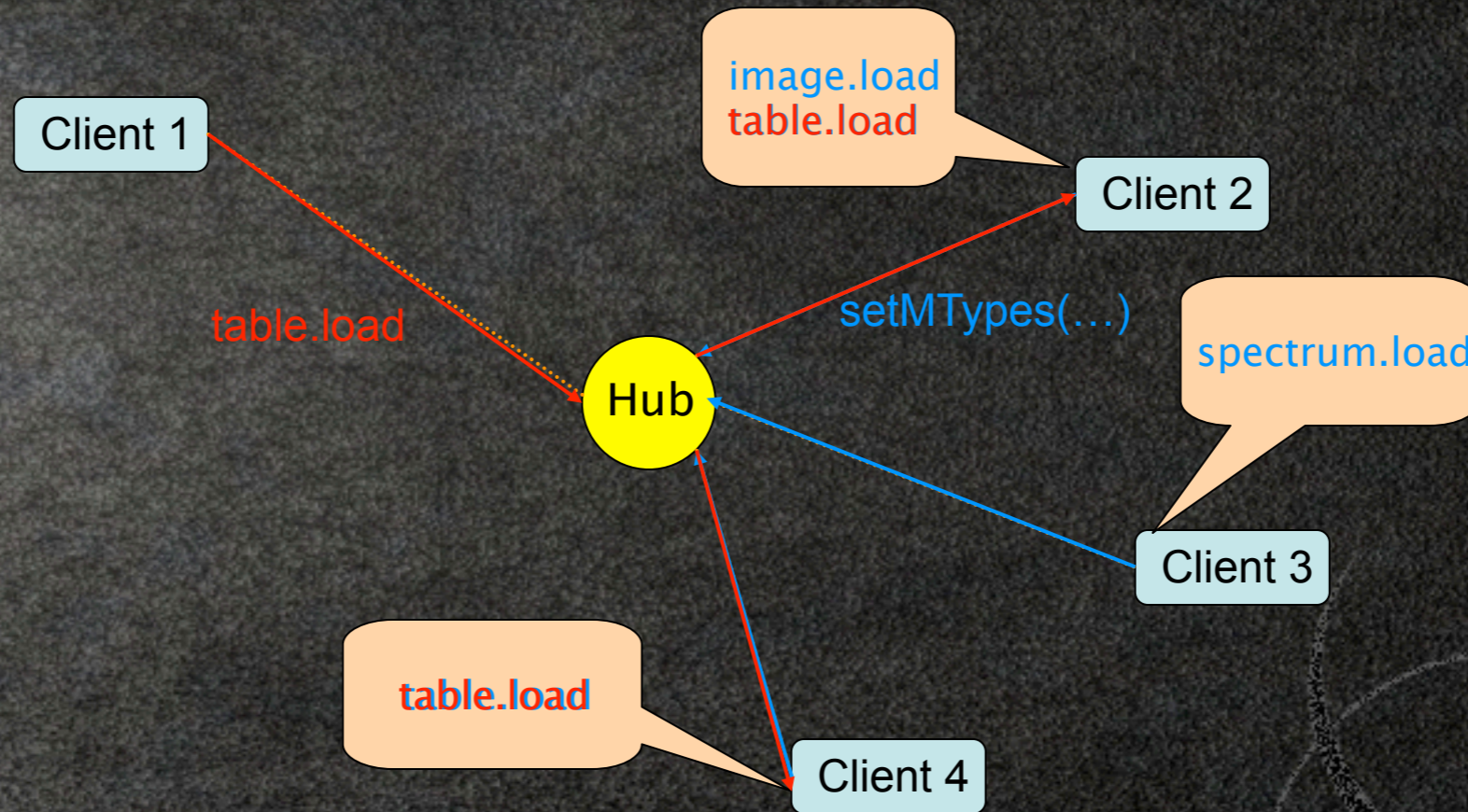


# MTypes (message-types)

- Message = MType + paramètres
- MType: définit la sémantique d'un message
  - exemple: chargement d'une image  $\longrightarrow$  **image.load.fits**
- Liste des mtypes :  
<http://www.ivoa.net/cgi-bin/twiki/bin/view/IVOA/SampMTypes>
- Flexible
  - possibilité d'ajouter de nouveaux messages
  - le hub n'est qu'un passe-plat, il ne connaît pas la sémantique des messages

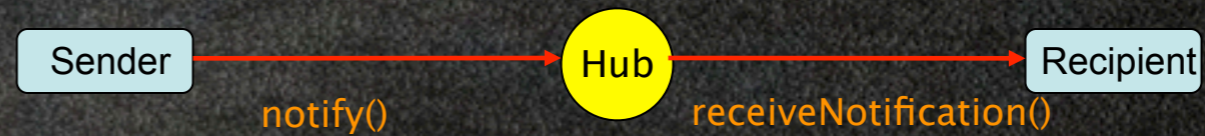
# Mécanisme de souscription

- Chaque client déclare les MTypes auxquels il souhaite souscrire
- Un message est transmis uniquement aux clients ayant souscrit au MType correspondant

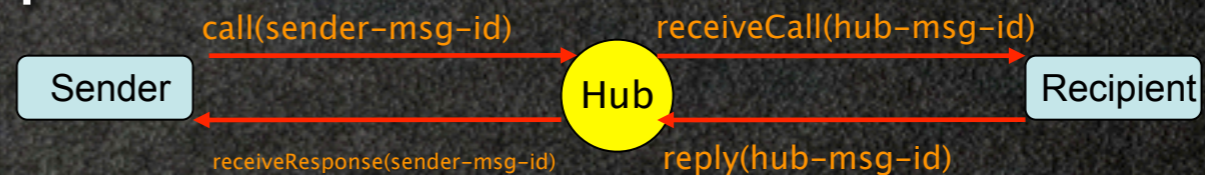


# 3 façons d'envoyer un message

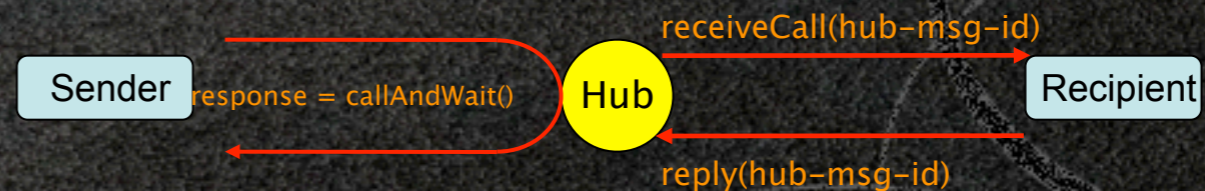
- Notification



- Asynchronous call/response



- Synchronous call/response





# APIs

- Le document SAMP définit des APIs de haut niveau, indépendamment de la couche de transport
  - mécanisme de découverte du hub
  - types de données SAMP (*string, list, map*) et conventions d'encodage des scalaires (*SAMP int, SAMP float, SAMP boolean*)
  - API du hub
  - API du client

# API du hub

- Liste des méthodes qu'un hub doit implémenter :
  - `register()`, `unregister()`
  - `setMetadata(map metadata)`, `getMetadata(...)`
  - `setMTypes(list mtypes)`, `getMTypes(client-id)`
  - `getRegisteredClients()`
  - `getSubscribedClients(list mtypes)`
  - `notify(...)`, `notifyAll(map message)`
  - `call(...)`, `callAll(map message)`
  - `response = callAndWait(map message)`
  - `reply(...)`

# API du client

- Ensemble des méthodes que le hub peut appeler, pour un client donné :
  - `receiveNotification(...)`
  - `receiveCall(...)`
  - `receiveResponse(...)`

# Profils

- Profil = ensemble de règles définissant comment les API de haut niveau sont mappées sur des implémentations
- Profils existants :
  - *Standard profile* : profil pour applications standalone
  - *Web profile* : profil pour applications Web

# Exemple d'une session SAMP (*standard profile*)

- Lecture du *lockfile*
- Inscription auprès du hub
- Déclaration des métadonnées
- Envoi d'une notification
- Déclaration des MTypes supportés
- Déclaration du endpoint XML-RPC
- Traitement d'un message reçu
-

# Lockfile

```
# SAMP lockfile written at  
2012-05-16T22:26:23+0000
```

```
# Required keys:
```

```
samp.secret=HyP0f1AQVZZxqmHLa26W
```

```
samp.hub.xmlrpc.url=http://10.37.129.2:8001/
```

```
samp.profile.version=1.3
```

# Requête XML-RPC pour s'enregistrer

```
<?xml version='1.0'?>
<methodCall>
  <methodName>samp.hub.register</methodName>
  <params>
    <param><value>
      <string>HyP0f1AQVZZxqmH1a26W</string>
    </value></param>
  </params>
</methodCall>
```

# Réponse du hub

```
<?xml version="1.0">  
<methodResponse>  
  <params>  
    <param><value>  
      <string>app-id:p435Zregi5WhQjPB2j</string>  
    </value></param>  
  </params>  
</methodResponse>
```



# Implémentations existantes

- JSAMP : <http://software.astrogrid.org/doc/jsamp/>
  - Hub (inclus dans Aladin, TOPCAT, VOSpec, applauncher, ...)
  - Toolkit client
- SAMPy : <http://pypi.python.org/pypi/sampy/>
  - Hub
  - Toolkit client
- En C ? (Mike Fitzpatrick, VAO)

# II. Le profil Web SAMP

# Motivations

- Pouvoir faire bénéficier les pages Web des fonctionnalités SAMP
- besoin de plus en plus fort à mesure que les applications Web deviennent des applications à part entière
- Solutions pré-existantes :
  - Web SAMP connector (*J. Berthier, R. Savalle et J. Normand.*)
  - plugin Firefox (*S. Derrière*)

# Difficultés

- environnement d'exécution des pages Web : sand box
- impose des limitations sur les ressources auxquelles on peut accéder
  - pas d'accès aux fichiers locaux
  - accès limités aux ressources réseau

# Différences par rapport au profil standard

- Mécanisme de découverte du hub
- Communication du client vers le hub
- Communication du hub vers le client (callback inversé)
- Accès à des URLs tierces (chargement d'une image, d'un catalogue)

# Mécanisme de découverte du hub

- ~~Lecture du fichier \$HOME/.samp~~
- Le hub est accessible depuis l'URL <http://localhost:21012/>

# Communication du client vers le hub

- Par défaut, les navigateurs appliquent la *same origin policy*
  - accès uniquement aux ressources provenant de la même origine
- Différentes stratégies pour contourner cette limitation :
  - CORS (Cross Origin Resource Sharing) : standard W3C implémenté dans navigateurs récents
  - Flash cross-domain policy
  - Silverlight cross-domain policy

- CORS : ajout de **Access-Control-Allow-Origin: \*** dans l'entête HTTP de la réponse du hub

- Flash : le hub fournit un fichier /crossdomain.xml :

```
<?xml version="1.0"?>
```

```
<!DOCTYPE cross-domain-policy
  SYSTEM "http://www.adobe.com/xml/dtds/cross-domain-policy.dtd">
<cross-domain-policy>
  <site-control permitted-cross-domain-policies="all"/>
  <allow-access-from domain="*" />
  <allow-http-request-headers-from domain="*" headers="*" />
</cross-domain-policy>
```



# Communication du hub vers le client

- Dans le profil standard, le client définit son callback endpoint (URL serveur XML-RPC où on peut le joindre)
- Dans le profil Web, le mécanisme de callback est inversé :
  - à intervalles réguliers, le client va demander au hub s'il a des messages pour lui

# Accès à des URLs tierces

- Le hub traduit les URLs et fait office de proxy pour permettre au client le chargement des données

- `image.load.fits(http://eso.org/siap/image42)`



`image.load.fits(http://localhost:21012/translator/client-private-key?http://eso.org/siap/image42)`

# Implémentations de hub supportant le profil Web SAMP

- JSAMP 1.3
- SAMPy 1.2.1

# Liens

- <http://ivoa.net/samp>
- Standard : <http://www.ivoa.net/Documents/SAMP/>
- Liste des MTypes : <http://www.ivoa.net/cgi-bin/twiki/bin/view/IVOA/SampMTypes>
- Liste des logiciels compatibles SAMP : <http://www.ivoa.net/cgi-bin/twiki/bin/view/IVOA/SampSoftware>