

Données distribuées: L'expérience GLU

Pierre Fernique
CDS

Le but

- **Répertoire distribué décrivant des ressources Web** (qq milliers)
- Pour :
 - éviter de hard coder des URLs
 - disposer d'un dictionnaire des services (=> disposer d'informations sur les ressources Web)

=> *Penser UDDI*

Les contraintes d'un tel répertoire

- Faible volume de données (qq Mo)
- Nombreux sites clients (qq dizaines)
- Nombreux sites serveurs (qq dizaines)
- Requêtes rapides (qq millisecondes)
- Synchronisation lente (qq heures)
- Le plus fiable possible

=> Penser DNS

Les "particularités" du GLU

- **Une distribution par domaines**

Un domaine peut être **thématique** (Astronomie, Planéto...) ou bien **institutionnel** (Observatoire de Strasbourg, CNES, CADC...)

=> Evite les duplications inutiles
ou non souhaitées

- Les mises à jour sont faites "**a priori**" et non "à la demande".

=> pas d'attente sur les requêtes



```

<RESOURCE ID="CDS/aladin/Skyview">
  <NAME>Skyview</NAME>
  <DESCRIPTION>HEASARC SkyView image server</DESCRIPTION>
  <QUERY>
    <URL>
      http://128.183.240.32:80/cgi-bin/pskcall?SCoord=equat
    </URL>
    <VAR NAME="1">
      <DESCRIPTION>Object or Coord.</DESCRIPTION>
      <TYPE REF="Target (COO|SIMBAD)" />
    </VAR>
    <VAR NAME="2">
      <DESCRIPTION>Width (pixel)</DESCRIPTION>
      <VALUE DEFAULT="true">300</VALUE>
    </VAR>
  </QUERY>
</RESOURCE>

```

Server selector

Choose an image server or a data server and fill in the associated form drawn below

Image servers:

- Aladin
- SkyView**
- Others...

HEASARC SkyView image server

Fill in all these fields and press the SUBMIT button

Target:
(ex.: "M99" or "05 47 17.0 -51 04 03")

Width (pixel):

Height (pixel):

Surveys:

File access:

- Local

Aladin Java

Load... Save... Help on Get FAQ Print... Quit

J2000 00:39:50.68 +42:01:45.1 012 Target: 00:42:40.34 +41:15:23.4 Size: 7.49 deg

M31

Scale

Skw IRAS 60 micron

Zoom 2x

1.0 deg

(c) ULP/CNRS 1999-2001 - Centre de Données astronomiques de Strasbourg

ION>

Trois clés

- Chaque site s'appuie sur trois éléments :
 - **sa Config** : la description des desiderata du site (sur quels "domaines" il veut publier, de quels domaines il veut recevoir... *(maj par le manager du site)*)
 - **son Dico local**: les infos à distribuer *(maj manuelle)*
 - **son Dico global**: Infos reçus des autres + les siennes *(maj automatique)*

*=> C'est le dico global qui servira
aux requêtes des applications*

Base 1 du protocole :

Pas d'accusé de réception :

- Chaque site compare ce qu'il "a" (dico global) par rapport à ce qu'il "veut" (config, dico local). S'il n'est pas content, il redemande constamment, avec un délais doublé à chaque fois.
- Exemple: *Je veux recevoir les infos concernant l'astronomie. Je vois que je ne suis pas encore dans la liste des "Clients" en astronomie. Je demande, redemande... mon inscription jusqu'à que j'y sois.*
- Remarque : Même idée que Ethernet

Avantages/inconvénients

- **Inconvénients:**
 - C'est souvent redondant (réponse qui croise la question)
 - C'est lent : peut prendre plusieurs minutes même si tout est ok.
- **Avantages :**
 - Ça ne peut que marcher tôt ou tard qq soit l'état du réseau ou des machines ou si les autorisations sont en cascade, ...
 - Ça auto régule le flux.

=> Rentre dans les contraintes

Base 2 du protocole :

Authentification par un tiers

- **Niveau 1** : savoir qui est "serveur" et qui est "client" pour un domaine afin de vérifier qu'un site à bien le droit d'émettre ou de recevoir des infos
=> *Solution : un manager par domaine qui est seul habilité à distribuer ces deux listes*
- **Niveau 0** : savoir qui est "manager" de domaines
=> *Solution : un site principal gère la liste des managers de domaines*

Avantages/Inconvénients

- **Avantages :**

- Mécanisme relativement simple par rapport à une authentification par clé publique/privée
- Délégation des authentifications par domaine
- Gère également les autorisations d'inscriptions

- **Inconvénients:**

- Niveau de sécurité moyen (car l'identification n'est basée que sur le nom de la machine)
- Un site "root" pour la création/suppression des domaines

Base 3 du protocole :

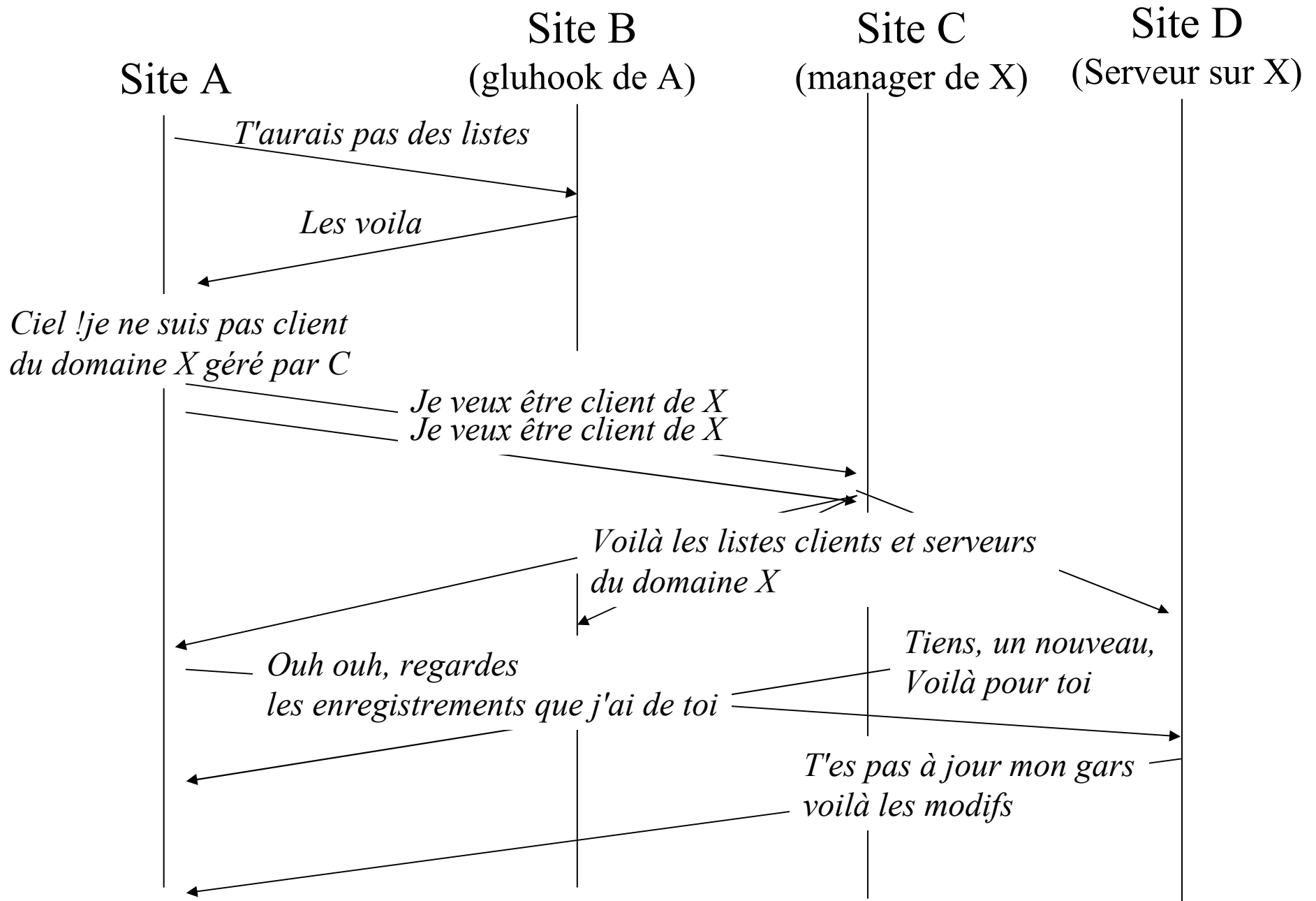
Une numérotation absolue

- Chaque **enregistrement est numéroté**
=> Permet de vérifier si un enregistrement est périmé ou non
- Cette numérotation n'est pas incrémentielle mais **basée sur l'horloge** du site émetteur
=> Evite le blocage des mises à jour en cas de crash/redémarrage de l'émetteur
(reprendrait la numérotation à zéro)

Base 4 du protocole :

Un point d'entrée

- Chaque site doit coder en dur "**1 parrain**" dans sa config (ou gluhook) pour entrer dans le système
- => A charge pour le "parrain" de lui retransmettre les listes clients et listes serveurs pour tous les domaines (le parrain doit nécessairement être manager d'au moins 1 domaine pour disposer de ces infos)



Actuellement...

- Principalement utilisé par le CDS
- Une trentaine de sites décrivant 500 ressources
- Daemon en C (unix),
 - + clients "locaux" en C et Perl
 - + clients "déportés" en java ou via URL.

Les sites GLU



Questions ?

Un enregistrement GLU

```
<RESOURCE ID="CDS/aladin/Skyview">
  <NAME>Skyview</NAME>
  <DESCRIPTION>HEASARC SkyView image server</DESCRIPTION>
  <QUERY>
    <URL>
      http://128.183.240.32:80/cgi-bin/pskcall?SCoord=equat
    </URL>
    <VAR NAME="1">
      <DESCRIPTION>Object or Coord.</DESCRIPTION>
      <TYPE REF="Target (COO|SIMBAD)"/>
    </VAR>
    <VAR NAME="2">
      <DESCRIPTION>Width (pixel)</DESCRIPTION>
      <VALUE DEFAULT="true">300</VALUE>
    </VAR>
    <VAR NAME="3">
      <DESCRIPTION>Height (pixel)</DESCRIPTION>
      <VALUE DEFAULT="true">300</VALUE>
    </VAR>
    <VAR NAME="4">
      <DESCRIPTION>Surveys</DESCRIPTION>
      <VALUE DEFAULT="true">
        EGRET (3D)
      </VALUE>
      <VALUE>EGRET >100 MeV</VALUE>
      <VALUE>EGRET <100 MeV</VALUE>
      <VALUE>COMPTEL</VALUE>
      <VALUE>HEAO 1 A-2</VALUE>
      <VALUE>BRCC-Cat Broad</VALUE>
    </VAR>
  </QUERY>
</RESOURCE>
```