



SITools

- Action de R et T "Système de gestion et d'accès aux données pour les laboratoires"
- Démontrer la faisabilité d'un système logiciel de gestion et d'accès aux données scientifiques spatiales qui soit
 - Léger
 - Portable
 - Générique
 - Modulaire
 - Interopérable
- Afin qu'il puisse être installé dans les laboratoires scientifiques et pris en charge par ces laboratoires.
- Composé d'un ensemble de briques permettant de composer un Système d'Information final (SITools)
 - avec un ensemble variable de fonctionnalités selon le besoin de l'entité l'utilisant.

- **Des projets pilotes**

- Le projet n'a de sens que s'il est soutenu et expérimenté par des laboratoires.

- » Pour définir et valider le besoin

- » Pour construire un prototype

- Qui rend un « service »

- Dans le but d'être la première pierre d'un projet complet

- **Interlocuteur**

- » *Medias*

- » *Base de données de Physique Submillimétrique*

- » *Base de données Planétaires*

- (CESR-Toulouse, IAS-Orsay, IPG-Paris, OMP-Toulouse et Observatoire de Bordeaux)

- **Le SIPAD-NG**
 - système opérationnel de préservation et d'accès aux données
 - met en œuvre des produits commerciaux éprouvés
 - met en œuvre un modèle de données sophistiqué
 - destiné aux centres de données ayant un mandat pour le long terme
- **SITTOOLS**
 - action de R&T destinée à fournir un ensemble d'outils indépendants et portables,
 - basés sur des produits OpenSource,
 - permet au travers de Web Services et d'une ou plusieurs applications cliente de constituer un système d'accès aux données.
 - ne constitue pas un tout :
 - » pas de fonction globale d'administration et de gestion, nécessaires à un système long terme.
 - destiné à mettre simplement et rapidement à disposition des données réparties sur différents laboratoires.

- Deux acteurs principaux

- L'utilisateur final :

- » le chercheur qui vient consulter un centre de données pour chercher des données utiles à ses travaux et les récupérer.

- Le gestionnaire :

- » l'entité qui va mettre en place et exploiter le système pour donner accès aux données qu'elle détient.

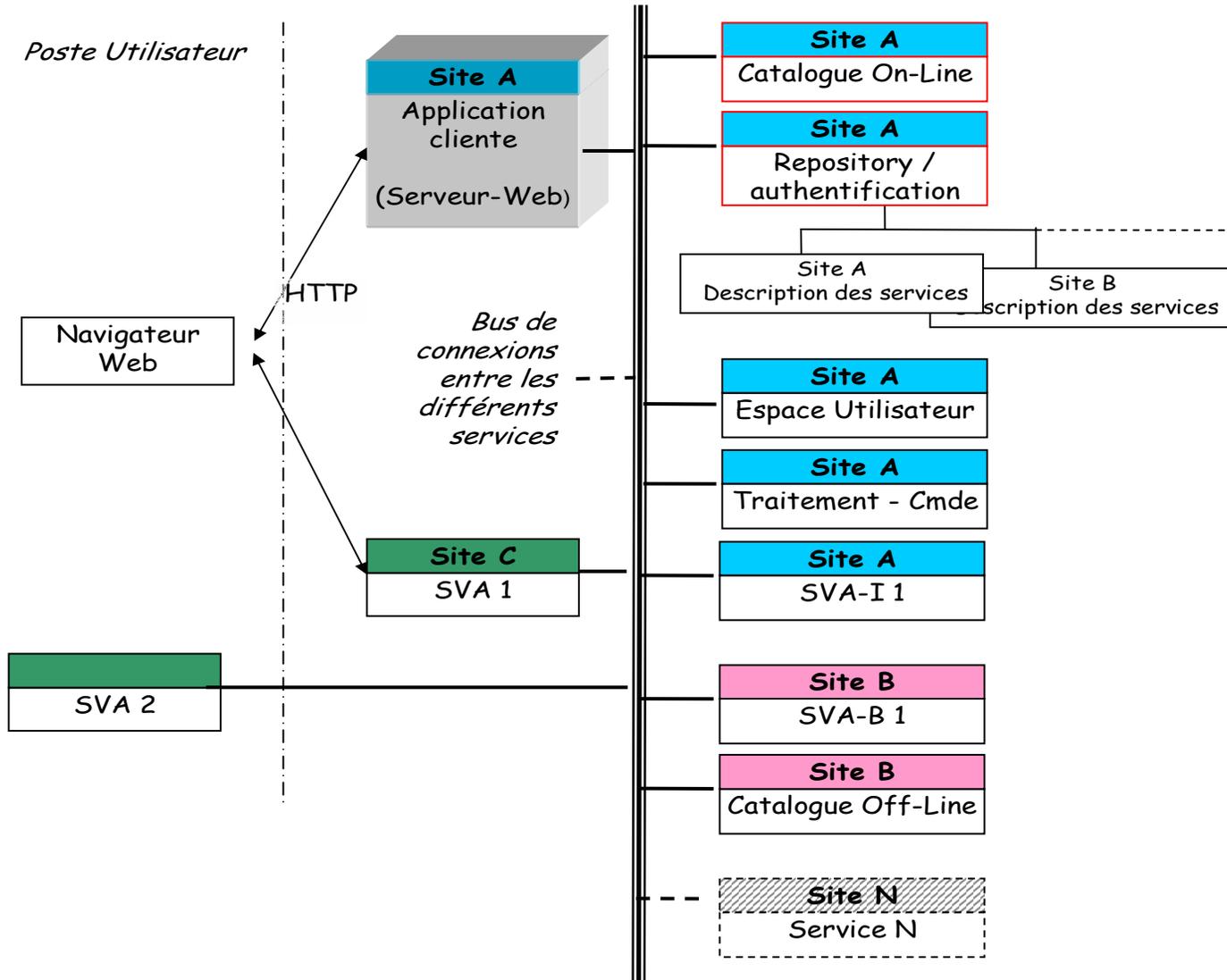
- » 3 fonctions séparées,

- le décideur : celui qui va décider la mise en place du système en donnant notamment les contraintes budgétaires,
- l'administrateur, celui qui va installer le système,
- l'exploitant, celui qui va le faire fonctionner au quotidien.

SITOOOLS visant plutôt les petites structures où une même personne pourra assurer les 3 fonctions: on préfère parler d'un « gestionnaire » qui regroupe l'ensemble des besoins.

- Basé sur un concept de services de base connectés entre eux pour offrir un service global incluant des services à valeur ajoutée.
- Pour mettre en place ce système, puis le faire fonctionner le gestionnaire devra enchaîner plusieurs phases :
 - installation, paramétrage, configuration, administration.
- Le système à mettre en place par le gestionnaire peut être un système complètement local, sur une même machine, ou un système distribué sur plusieurs machines voire sur plusieurs laboratoires.
- Le système global vu par l'utilisateur peut être un ensemble de système mis en place par différents gestionnaires.

SITTOOLS : schéma général



- 8 types de services :
 - » 1) catalogue « on-line »
 - » 2) catalogue « off-line »
 - » 3) repository / authentication utilisateurs
 - » 4) traitement commande
 - » 5) espace utilisateur
 - » 6) les Services à Valeur Ajoutée (SVA)
 - » 7) SVA-B
 - » 8) SVA-I

- La notion de type de service n'est caractérisée que par les interfaces de ces services.
 - SITOOOLS en fournira des « instances »
 - Les gestionnaires pourront créer d'autres « instance »

- Les services sont des Web-Services (SOAP/WSDL)

- **Catalogue On-line**

- donne accès aux méta-données et aux données accessibles en ligne.
 - » Les méta-données incluent les documents et les graphiques associés aux données et disponible en ligne.

- **Catalogue Off-line**

- reprend les fonctions du service « catalogue on-line », avec un accès batch aux données

- **Repository / Authentification Utilisateurs**

- fait le lien entre tous les services accessibles pour constituer un système global
- référence les services et les utilisateurs
- Permet de gérer l'authentification utilisateur et les droits d'accès à un service.

- **Traitement Commande**

- gère la commande de l'utilisateur de bout en bout.
 - » récupère la requête, extrait les données des catalogues concernés, effectue les traitements (utilisation de SVA-B), et livre les données à l'utilisateur.

- **Espace Utilisateur**

- gère l'espace sur lequel les résultats des commandes sont mis à disposition de l'utilisateur.
- est utilisé aussi par les Services à Valeur Ajoutée interactifs.

- **Les Services à Valeur Ajoutée (SVA)**

- Logiciels indépendants, s'interfaçant avec le système pour apporter de nouvelles fonctionnalités.

- Pas de spécification de périmètre d'un SVA.

- Cela peut être une application indépendante, ou cliente de services, web ou non web.

- Plus-value :

- » Définition du fonctionnement du SVA (méthode d'appel, type, paramètres d'appels, etc...),

- » Liens avec les éléments du modèle d'information (collections, jeux de données ou de browse)

- » Gestion des droits d'accès,

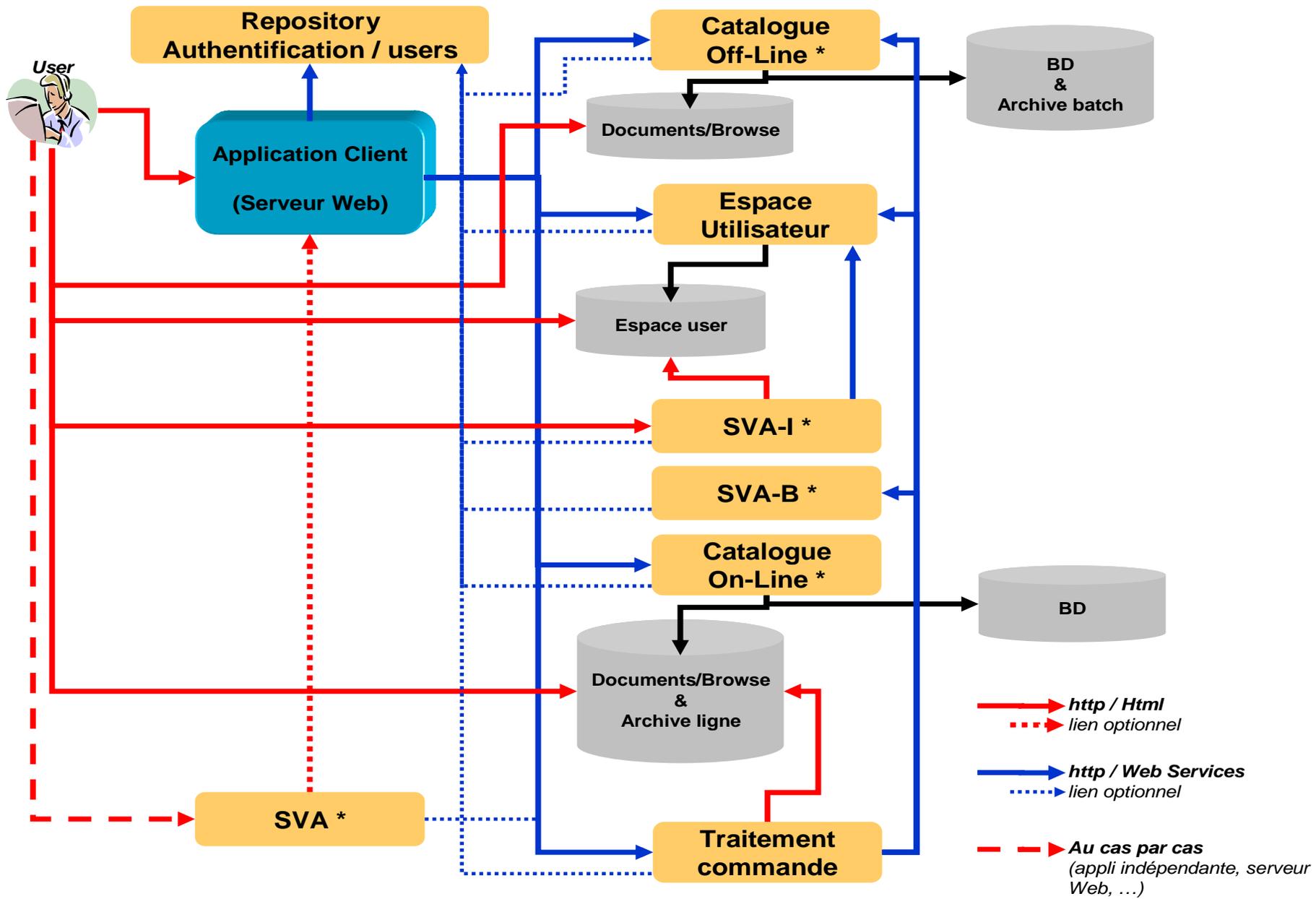
- » Limite la nécessité de rajouter des types de services

- un nouveau type de service entraînant des modifications lourdes (repository, applications clientes, autres services utilisateur ...)

- 2 types de SVA où toutes les interfaces entre ceux-ci et le système sont complètement prédéfinies et documentées :
 - Services à Valeur Ajoutée de transformation (ou SVA batch ou SVA-B)
 - » effectuent des traitements sur les données archivées pour en changer la forme ou le contenu pendant l'exécution d'une commande,
 - Services à Valeur Ajoutée interactifs (ou SVA-I)
 - » Permettent à l'utilisateur de travailler de manière interactive sur des données présentes sur son espace utilisateur (tracé de graphiques par exemple).

- L'application cliente :
 - point de départ de toutes actions vis-à-vis de l'utilisateur.
 - s'interface avec la plupart des autres services,
 - » Propose les fonctions classiques d'un centre de données
 - Recherche,
 - Commande,
 - Execution de SVA,
 - ...
 - » Se base sur un repository

SITOOOLS : Interfaces



- Développement en 2 phases

- 1ère phase juin 2004

- » Prototype V1 : Valider les concepts

- Mise en place de l'architecture, définition des interfaces,
 - Développement des services à périmètres réduits

- Le système doit répondre à au moins un scénario d'utilisation démontrant le fonctionnement de bout en bout*

- Instanciation pour des données de surfaces planétaires
 - Avec intégration de 2 services existants (Visu 3D, BD Clementine)

- » Consultation industrielle en cours

- Pour un démarrage en janvier 2004

- 2ème phase

- » Prototype complet -> mi 2005

- Une 3ème phase en // consiste en un support à l'expérimentation dans les laboratoires

- **Technos !**
 - Quelques points techniques en rapport
 - avec le SIPAD-NG et SITOOLS

- **Serveur d'application**
 - **J2EE**
 - » **Prototype SIPAD-NG, avec JBoss**
 - » **Bilan**
 - **Prise en main lourde**
 - **Dépendant d'un produit**
 - **Utilisation des EJB Entities non pertinente pour nos problématiques**
 - **+ de contraintes que de bénéfices**

- **Passage des paramètres en XML**
 - D'un service à une application au travers d'une chaîne XML.
 - Cela permet de modifier les paramètres passés sans modification de la signature des méthodes utilisées.
 - !!! Par rapport à un objet Java, présente un inconvénient : il n'y a pas de vérification de type à la compilation.
 - Pas d'utilisation systématique car XML est verbeux
 - » Par exemple pour les longues listes de résultats

• MVC

- Le modèle MVC sépare l'affichage du traitement.
- Les actions sont traitées par un « Contrôleur » (le « C » du MVC), qui demande à un « Modèle » de se charger (le « M ») et qui demande à une « Vue » de s'afficher (le « V »).
- La vue utilise le modèle pour afficher les valeurs ; la vue fait juste des «get»/«set» sur le modèle, et n'a pratiquement pas de code java embarqué.
- Utilisation de Struts (framework d'apache) ou implémentation spécifique (+souple mais moins « propre »).

» FIN