

Interfaces riches et SURF



Constat :

L'interface utilisateur (de base) n'est pas adaptée à tous les besoins :

- Interface trop riche, trop floue, trop complexe
- Trop de liberté pour les utilisateurs
- Contraintes métiers à mettre en place

- Alfresco Explorer personnalisé
- (Embarquer l'entrepôt dans son application)
- APIs « remote » d'Alfresco
- Web Scripts

- Développement long et coûteux
- Les *points d'extension* sont peu nombreux
- Attention lors des mises à jours !
- Alfresco ne mise plus sur Alfresco Explorer

Exemple : *Parapheur Électronique ADULLACT*

- Standards (= interop)
 - Java Content Repository
 - CMIS (SOAP ou REST) (*ex: Joomla, Drupal*)
File-system évolué (check-int/out, recherche, ...)
- Spécifiques Alfresco
 - Web Services (SOAP)
 - REST (HTTP)
L'avenir d'Alfresco (stable = lifecycle)

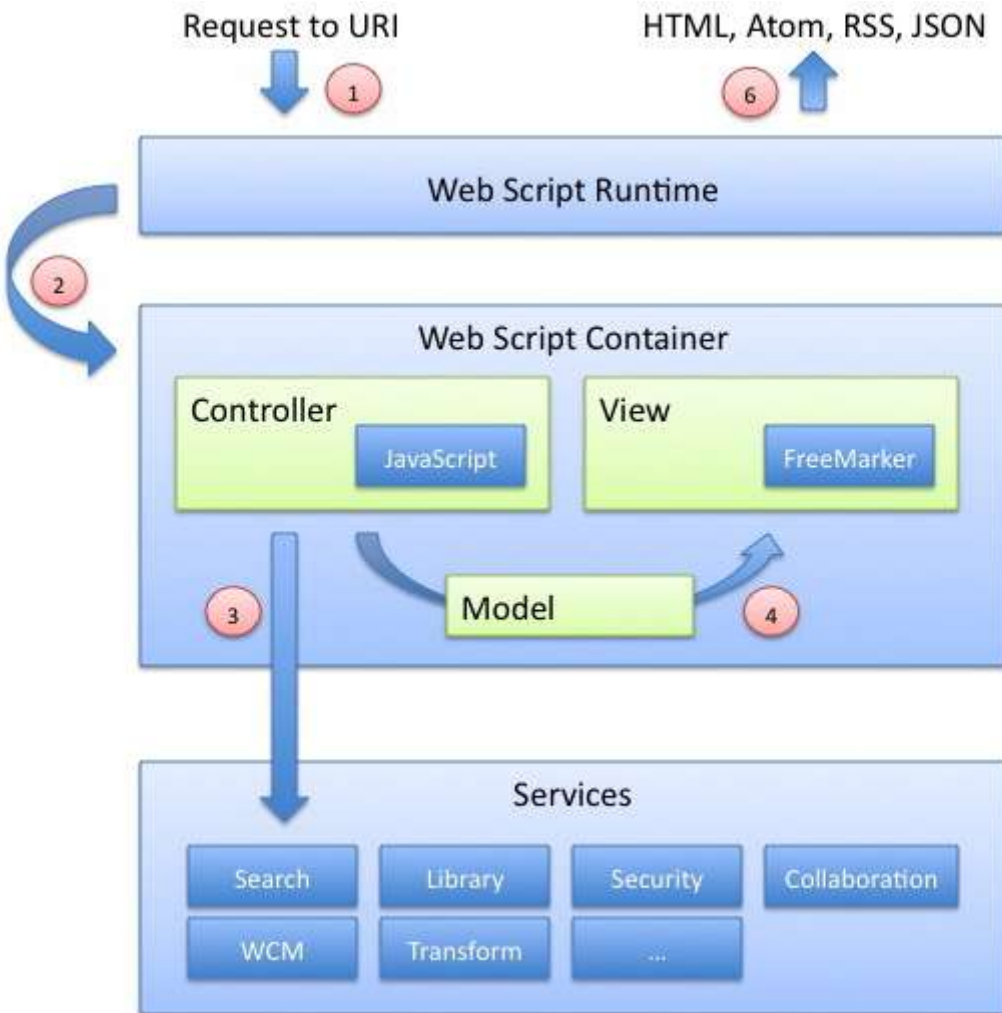
2 approches :

- Interface HTML
- API REST « custom »

- Orienté métier
- Rien n'empêche d'utiliser :
 - l'API REST « standard » d'Alfresco
 - les deux approches à la fois (AJAX / SURF)

- Clients « génériques » :
 - Plug-in MS Office 2003
 - Portlets Alfresco (*My Spaces, My Tasks, My Documents*)
 - DoCASU, Opsoro, JibeFramework
 - FlexSpaces

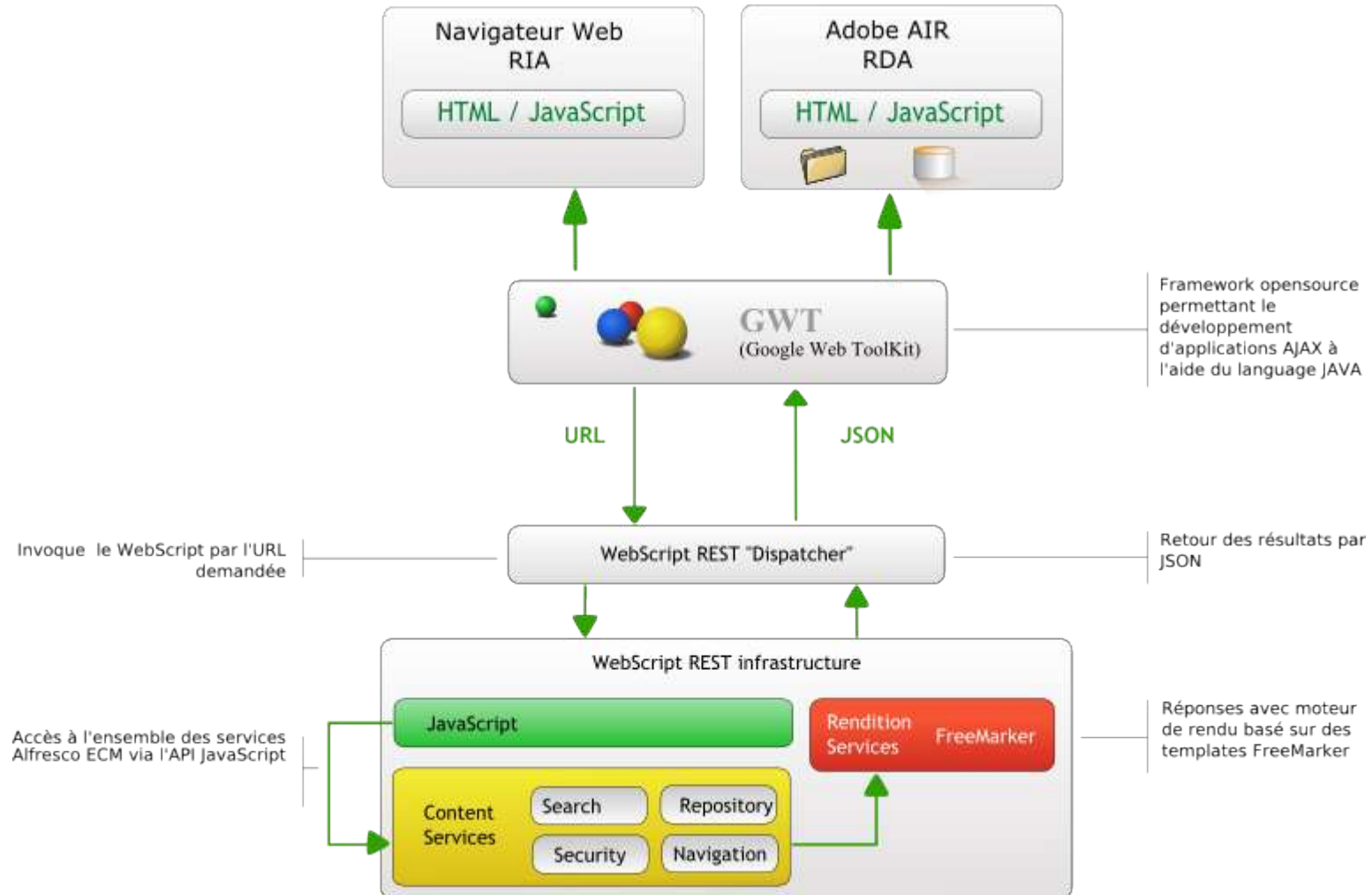
- Clients « spécialisés » :
 - Alfresco Share, Web Studio, Network
 - Adobe Acrobat.com
 - UBIC (EADS, dématérialisation RH)

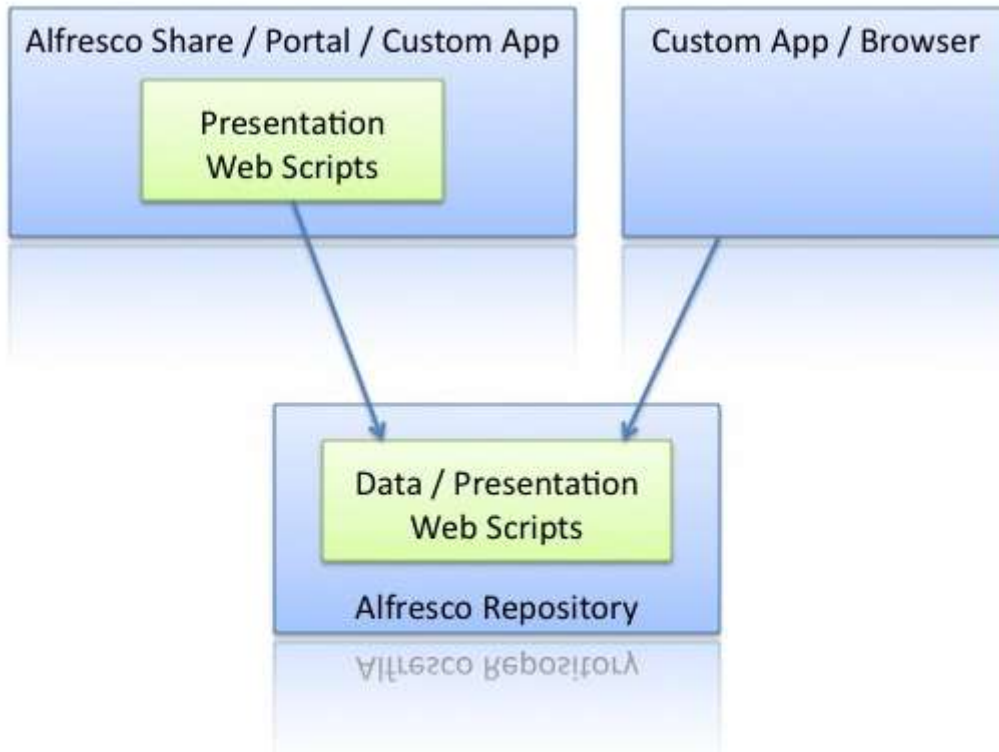


N'importe quel client HTTP !

- Navigateur Web
- Application desktop
- Silverlight, Flash, ...
- Portail xNet (PHP, SURF)
- ERP, CRM, LoB, ...
- Mashups

Source : http://wiki.alfresco.com/wiki/3.0_Web_Scripts_Framework





- Dans Alfresco
 - Accès à l'entrepôt
- Hors d'Alfresco
 - Client HTTP

- Met à profit les Web Scripts (hors d'Alfresco)
- Page + Composants
 - Template : définit des régions
 - Page : met des composants dans les régions
 - Composant : Web Script
 - Associations entre pages : plan de site
 - ...
- Le tout à base de XML + JS + FTL uniquement !
(nota : on peut utiliser JSP et Java également)

Limitation :

Découlant de la composition des pages par composants, SURF ne supporte que HTTP GET.

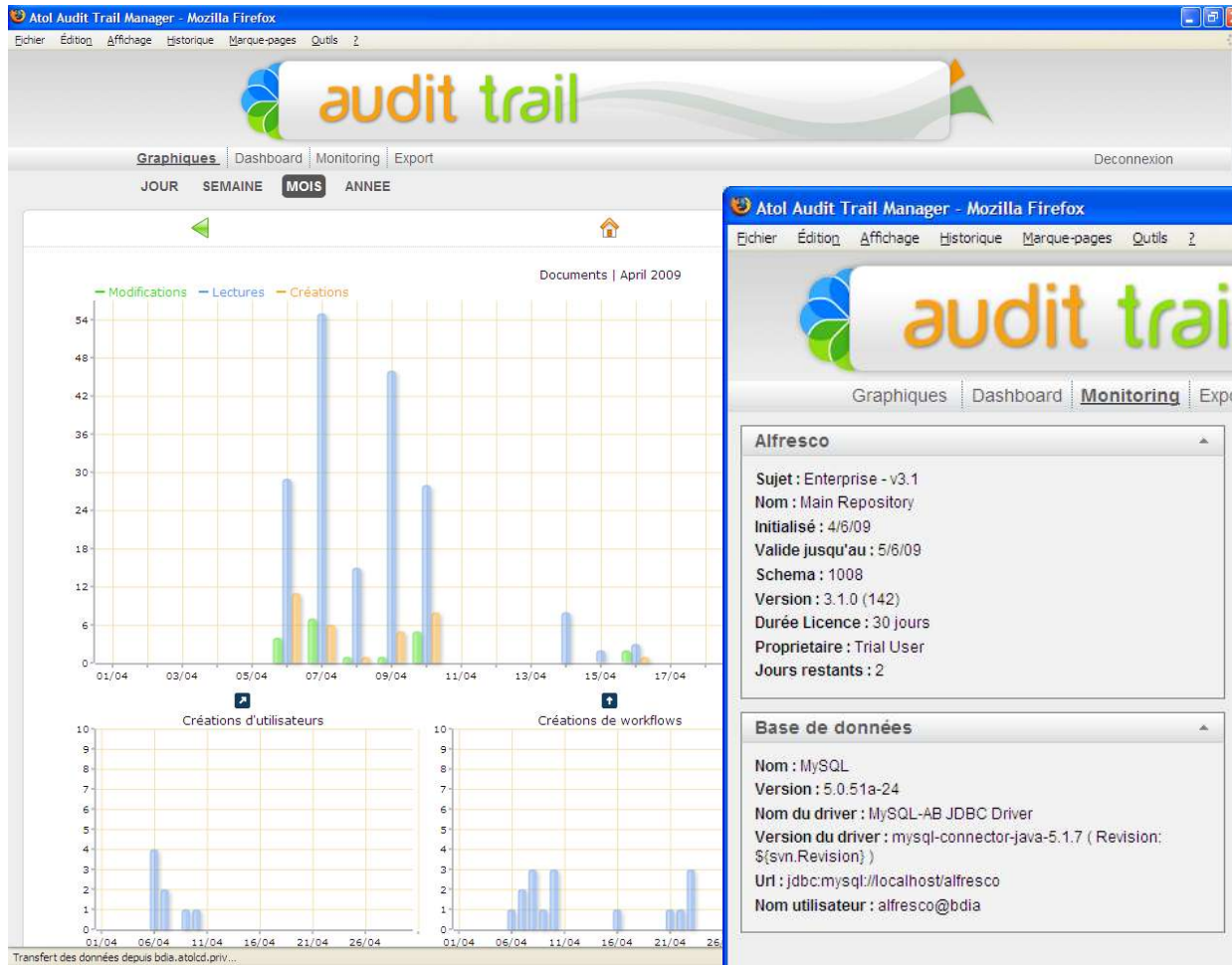
Pour HTTP POST/PUT/DELETE, il faut utiliser des Web Scripts en accès direct, i.e. AJAX ou un rendu HTML simple.

(en train de changer ?)

(page non *cacheable*, point d'amélioration : ALFCOM-2453)

Avantages de cette architecture :

- Architecture / déploiement
 - Appli SURF sur DMZ, Alfresco sur LAN privé
- Montée en charge
 - N applis SURF, M serveurs Alfresco
 - REST (pas de données de session)



Alfresco

- Sujet : Enterprise - v3.1
- Nom : Main Repository
- Initialisé : 4/6/09
- Valide jusqu'au : 5/6/09
- Schema : 1008
- Version : 3.1.0 (142)
- Durée Licence : 30 jours
- Propriétaire : Trial User
- Jours restants : 2

Runtime

- Memoire libre : 38 Mo
- Memoire max : 493 Mo
- Memoire totale : 127 Mo

Configuration Serveur

- CIFS : activé
- Nom serveur : BDIAA
- FTP : activé
- NFS : désactivé

Base de données

- Nom : MySQL
- Version : 5.0.51a-24
- Nom du driver : MySQL-AB JDBC Driver
- Version du driver : mysql-connector-java-5.1.7 (Revision: \${svn.Revision})
- Url : jdbc:mysql://localhost/alfresco
- Nom utilisateur : alfresco@bdia

Stores

Index Lucene

En attente de bdia.atolcd.priv...

Thomas Broyer

Atol Conseils & Développements

3 bd Eiffel, 21600 Longvic

03 80 68 81 68 – contact@atolcd.com

www.atolcd.com