



# BorderCloud

Web 3.0 Hosting & Factory

## Tutoriel Agrégation

### Structuration de contenus hétérogènes Accès aux données structurées avec MediaWiki



Session Tutoriel : 11<sup>ème</sup> Conférence Internationale Francophone  
sur l'Extraction et la Gestion des Connaissances

Karima Rafes, BorderCloud

07/01/2011



# Rappel du sujet du tutoriel

## **Agrégation, structuration de contenus hétérogènes et accès aux données structurées avec MediaWiki**

Les Wikis sont des outils très flexibles pour agréger du contenu hétérogène et regorgent rapidement d'informations que l'on souhaite pouvoir partager dans son Intranet ou sur le Web.

A l'occasion de la 11ème Conférence Internationale Francophone sur l'Extraction et la Gestion des Connaissances, l'entreprise BorderCloud décrira sa solution avec des logiciels libres pour structurer un Wiki et partager en temps réel les données de ce Wiki à travers un silo de données SPARQL.

Dans ce tutoriel, nous utiliserons MediaWiki, 4Store, l'extension Sémantique MediaWiki et LinkedWiki. Ces logiciels offrent une architecture compatible avec le Web des données (Linked Data) pour les développeurs et les chercheurs qui veulent collaborer sur une architecture commune pour avancer ensemble sur la roadmap du Web Sémantique.

# Emploi du temps

(24 janvier 2011 salle 8)



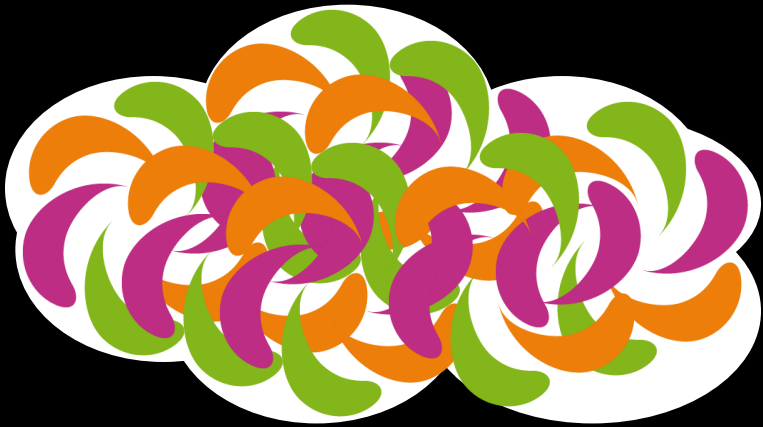
**Matin : 9h-12h15 - Pause café 10h-10h30**

- 🌀 Introduction, un exemple : un DataTricorder
- 🌀 Installation du serveur du service web et du silo SPARQL
- 🌀 Installation du Wiki

**Déjeuner : 12h15 à 13h45**

**Après-midi : 13h45 à 17h30 - Pause café 16h15-16h45**

- 🌀 Démarrage d'un wiki : agrégation de données non structurées ?
- 🌀 Maturation du wiki : structuration dans sa langue
- 🌀 Partage des données du wiki : réalignement



# Introduction

**Il nous faut un service 3.0 ?**

# Objectif : DataTricorder



- 🌀 Il fallait un service avec des données qui permet :
  - aux humains de consulter ces données en ligne
  - aux humains d'enregistrer ces données en ligne
  - à toutes les machines de consulter ces données en ligne
  - à toute les machines d'enregistrer ces données en ligne

Ces machines ont été imaginées dans une série TV des années 60. Ce sont les tricorders. VIDEO

# C'est quoi un Tricorder ?



Le tricordeur est un petit outil tenant dans la main et capable de remplir trois fonctions de base :

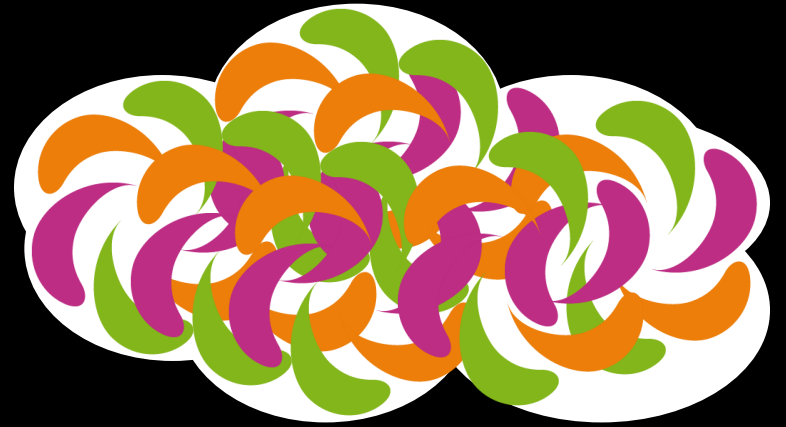
- 🌀 détecter,
- 🌀 enregistrer
- 🌀 et analyser.

# Et un DataTricorder ?



Le service DataTricordeur est la source d'information de tout tricorder mais aussi le lieu du dépôt de tous les tricorders.

Avec le DataTricorder, les opérateurs des tricorders peuvent ainsi partager leurs découvertes avec le reste des êtres humains qui n'ont pas la chance de voyager sur d'autres planètes ;- ) C'est clair ?



# Installation serveur

# Création du serveur

## Le serveur est dédié car

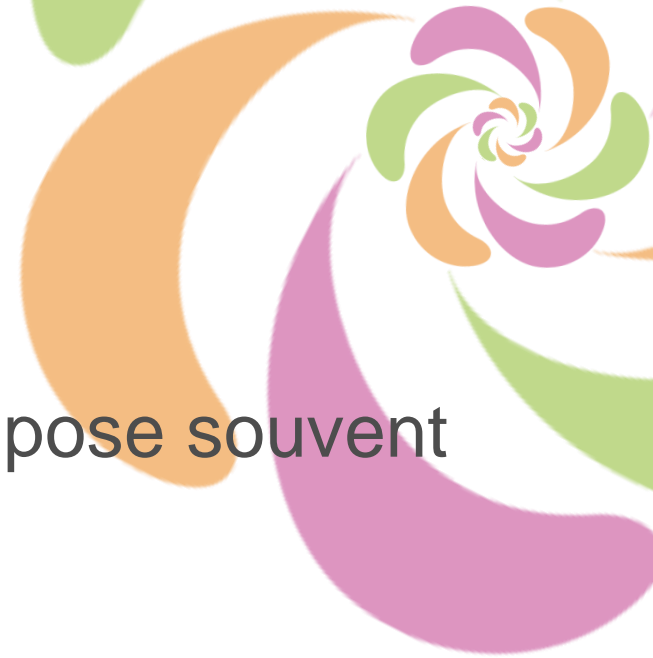
- La RAM sera utilisée intégralement par le service pour la performance des requêtes SPARQL.

## Le serveur doit être de type :

- Linux (préférence pour Redhat, CentOS, Fedora mais fonctionne aussi avec Ubuntu)
- 64bits (4Store ne fonctionne pas sur des systèmes 32bits)

Par exemple : on utilisera l'hébergeur Gandi.net. Pour installer un serveur dédié Ubuntu 64bits.

# On commence par



- ❁ La configuration HTTP car cela pose souvent des problèmes :
  - Mauvaise configuration DNS
  - Construction d'un triplestore avec des URI incorrecte
  - Le multilingue demande la mise en place de sous-domaine et d'un virtual hosting
  
- ❁ Installation du TripleStore : 4Store
  - Si le triplestore ne fonctionne pas sur le serveur, il est inutile de continuer une installation

# Installation : service HTTP

- 🌀 Nous installons ce qu'il manque sur le serveur :  
(d'habitude apache est déjà installé)
  - apt-get install apache2
- 🌀 On active le mode en lecture et écriture "mod\_rewrite" :
  - a2enmod rewrite
- 🌀 On vérifie la configuration de Apache
  - apache2ctl -S
- 🌀 On duplique le fichier « default » dans le répertoire :  
/etc/apache2/sites-available pour servir de modèle pour les nouveaux sites qu'on souhaite héberger. (slide suivant)
- 🌀 On active les sites web
  - a2ensite fr.datatricorder.org
  - ( a2dissite pour désactiver)
- 🌀 On relance Apache : /etc/init.d/apache2 reload  
et on TESTE

# Exemple de fichier de configuration Apache2 pour MediaWiki avec LinkedWiki

```
<VirtualHost *:80>
  ServerAdmin contact@datatricorder.org

  ServerName fr.datatricorder.org
  DocumentRoot /var/www/fr.datatricorder.org
  <IfModule alias_module>
    Alias /wiki "/var/www/fr.datatricorder.org/index.php"
  </IfModule>
  redirectMatch ^/sparql(.*)$ /wiki/Special:SparqlEndpoint$1

  <Directory />
    Options FollowSymLinks
    AllowOverride None
  </Directory>
  <Directory /var/www/fr.datatricorder.org/>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    allow from all
  </Directory>
```

Page 1

```
ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
<Directory "/usr/lib/cgi-bin">
  AllowOverride None
  Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
  Order allow,deny
  Allow from all
</Directory>

ErrorLog ${APACHE_LOG_DIR}/error.log

# Possible values include: debug, info, notice, warn, error, crit,
# alert, emerg.
LogLevel warn

CustomLog ${APACHE_LOG_DIR}/access.log combined

Alias /doc/ "/usr/share/doc/"
<Directory "/usr/share/doc/">
  Options Indexes MultiViews FollowSymLinks
  AllowOverride None
  Order deny,allow
  Deny from all
  Allow from 127.0.0.0/255.0.0.0 ::1/128
</Directory>

</VirtualHost>
```

Page 2

**Orange** : Lignes nécessaires à la création d'un "virtual hosting" fr.

**Vert** : Lignes pour obtenir des URI plus courtes comme sur Wikipédia.

**Violet** : proxy permettant de faire un point d'accès REST pour SPARQL

# Installation : service 4s-httpd

- 🌀 Le site [4store.org](http://4store.org) indique ces dépendances à installer :
  - `apt-get install build-essential libpcre3-dev librasqal2-dev libglib2.0-dev libncurses5-dev libreadline-dev`
- 🌀 Pour installer 4Store il faut le télécharger avec GIT et le compiler ...
  - `apt-get install git automake libnss-mdns`
  - `git clone git://github.com/garlik/4store.git 4store`
  - `./autogen.sh`
  - `./configure`
  - `make`
  - `make install`
- 🌀 On teste
  - `4s-backend-setup test`
  - `4s-backend test`
  - `4s-httpd -D test`
  - <http://fr.datatricorder.org:8080> pour voir si 4s-httpd fonctionne
  - `killall 4s-httpd`
  - `killall 4s-backend`
  - `4s-backend-destroy test`
- 🌀 On n'oublie pas de créer le répertoire pour les logs : `mkdir /var/log/4store`

# Installation : service datatricorderd

4store n'a pas de lanceur automatique. On va en fabriquer un rapidement pour lancer le silo de notre DataTricorder.

- 🌀 Installation du silo (Attention cette commande écrase aussi le silo précédent du même nom)
  - 4s-backend-setup datatricorder
- 🌀 On fabrique le script pour lancer le service automatiquement au démarrage du serveur
  - vim datatricorderd (slide suivant)
  - cp datatricorderd /etc/init.d/.
  - chmod 0755 /etc/init.d/datatricorderd
  - /etc/init.d/datatricorderd start
  - ps -AL (on teste que les processus 4s-httpd et 4s-backend fonctionne)
  - apt-get install links (pour tester 4s-httpd via ssh derrière un firewall)
  - links http://localhost:8081
  - /etc/init.d/datatricorderd stop
  - links <http://localhost:8081> (on teste que le service s'arrête)
- 🌀 On installe le service datatricorderd dans le système et on le lance pour la dernière fois manuellement
  - update-rc.d datatricorderd defaults
  - (update-rc.d -f datatricorderd pour le désinstaller)
  - /etc/init.d/datatricorderd start

```

#!/bin/sh -e
DAEMON_BACKEND="/usr/local/bin/4s-backend"
DAEMON_HTTPD="/usr/local/bin/4s-httpd"
DAEMON_KB="datarecorder" #argument à utiliser par le programme
DAEMON_PORT="8081"
DAEMONUSER="root" #utilisateur du programme
DAEMON_NAME_BACKEND="4s-backend"
DAEMON_NAME_HTTPD="4s-httpd"
DAEMON_NAME="datarecorderd"

PATH="/sbin:/bin:/usr/sbin:/usr/bin" #Ne pas toucher

test -x $DAEMON || exit 0

. /lib/lsb/init-functions

d_start () {
    log_daemon_msg "Starting system $DAEMON_NAME Daemon"
        $DAEMON_BACKEND $DAEMON_KB
        $DAEMON_HTTPD -p $DAEMON_PORT $DAEMON_KB
    log_end_msg 0
}

d_stop () {
    log_daemon_msg "Stopping system $DAEMON_NAME Daemon"
        killall $DAEMON_HTTPD
        killall $DAEMON_BACKEND
    log_end_msg 0
}

case "$1" in
    start)
        d_start
        ;;
    stop|force-stop)
        d_stop
        ;;
    restart|reload|force-reload)
        d_stop
        d_start
        ;;
    status)
        status_of_proc "$DAEMON_NAME_BACKEND" "$DAEMON_BACKEND" "system-wide $DAEMON_NAME_BACKEND" && exit 0 || exit
        $?
        status_of_proc "$DAEMON_NAME_HTTPD" "$DAEMON_HTTPD" "system-wide $DAEMON_NAME_HTTPD" && exit 0 || exit $?
        ;;
    *)
        echo "Usage: /etc/init.d/$DAEMON_NAME {start|stop|force-stop|restart|reload|force-reload|status}"
        exit 1
        ;;
esac
exit 0

```

# Installation : Sécurité...



Ici on va installer le minimum pour faire un TP ou un démonstrateur sur le réseau mais il faut être attentif à son serveur comme avec du lait sur le feu.

- 🌀 Sécurité réseau : Iptables, fail2ban,...
- 🌀 Restauration système : pas aborder ici...

# Installation :

## Sécurité réseau

On bloque tout sauf le service http et ssh.  
On ralentit les attaque sur l'accès ssh.

 En bref :

- apt-get install iptables fail2ban
- vim /etc/init.d/server\_iptables (slide suivant)
- chmod +x /etc/init.d/server\_iptables
- update-rc.d server\_iptables defaults

```
#!/bin/bash
# reset iptables
iptables -F
```

```
# Autorise les connections sortantes et sur l'interface "loopback"
iptables -P OUTPUT ACCEPT
iptables -A INPUT -i lo -j ACCEPT
iptables -A INPUT -d 127.0.0.0/8 -i ! lo -j DROP
```

```
# Autorise les connections deja etablies
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

```
# Autorise HTTP, SSH, ICMP-ping
iptables -A INPUT -p tcp -i eth0 --dport ssh -j ACCEPT
iptables -A INPUT -p tcp -i eth0 --dport 80 -j ACCEPT
iptables -A INPUT -p icmp -i eth0 -j ACCEPT
```

```
# Refuse a priori ce qui vient de l'exterieur
iptables -P INPUT DROP
iptables -P FORWARD DROP
```

# Références :

## Sécurité, Apache, 4store

### Apache sur Ubuntu :

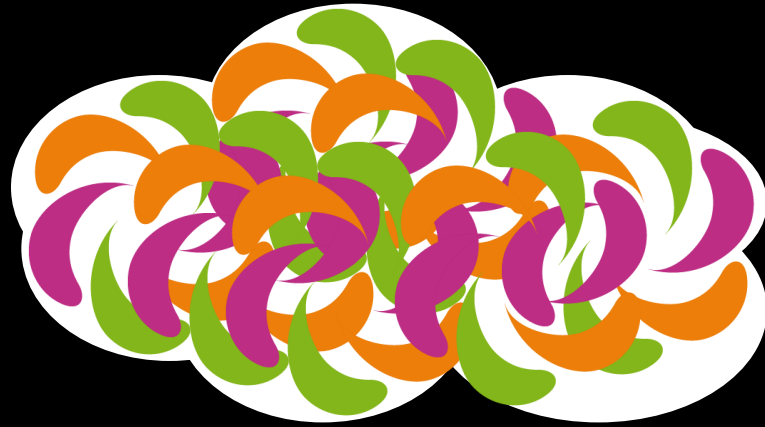
- <http://doc.ubuntu-fr.org/apache2>

### Sécurité de base sur Ubuntu :

- <http://brocas.org/blog/post/2008/01/17/Gandi-Hebergement-part-2%3A-securiser-son-serveur-Ubuntu>
- <http://www.chiroux.com/installation-dun-serveur-web-securise-sous-ubuntu-9-04server/>

### Documentation 4Store et FAQ

- <http://4store.org/trac/wiki/Documentation>
- <http://groups.google.com/group/4store-support>



# Installation du Wiki

# Installation : MediaWiki



- 🌀 Installation des dépendances de MediaWiki
  - apt-get install apache2 mysql-server php5 php5-mysql php-apc imagemagick
- 🌀 Télécharge la dernière version et on déploie le wiki :
  - mkdir /var/www/archives
  - cd /var/www/archives
  - wget http://download.wikimedia.org/mediawiki/1.16/mediawiki-1.16.1.tar.gz
  - tar vxzf mediawiki-1.16.1.tar.gz
  - cd mediawiki-1.16.1
  - cp -r ./\* /var/www/fr.datatricorder.org/
  - cd /var/www/fr.datatricorder.org/
  - chmod a+w config
- 🌀 On installe le Wiki et on supprime les fichiers d'installation
  - <http://fr.datatricorder.org>
  - mv config/LocalSettings.php .
  - rm -rf config

# Installation :

# Semantic MediaWiki

Semantic MediaWiki est une extension qui va permettre d'extraire les triplets du Wiki.


- 🌀 Téléchargement de la dernière version
  - `wget http://sourceforge.net/projects/semmediawiki/files/semmediawiki/semmediawiki-1.5.4/SemanticMediaWiki1.5.4.tgz/download`
  
- 🌀 Installation
  - `mv download SemanticMediaWiki1.5.4.tgz`
  - `tar vxzf SemanticMediaWiki1.5.4.tgz`
  - `mv SemanticMediaWiki /var/www/fr.datatricorder.org/extensions/.`
  - `vim /var/www/fr.datatricorder.org/LocalSettings.php`
    - **Insérer à la fin du fichier** LocalSettings.php :
    - `include_once("$IP/extensions/SemanticMediaWiki/SemanticMediaWiki.php");`
    - `enableSemantics('fr.datatricorder.org');` //corriger après la vidéo
  
- 🌀 Page d'administration de l'extension : on installe la BDD et on met à jour les données.
- 🌀 On teste en créant une première propriété

# Installation : LinkedWiki

Cette extension va permettre d'enregistrer les triplets du Wiki dans le silo de données pour offrir un point d'accès SPARQL .

## Installation :

- apt-get install php5-curl
- wget https://github.com/BorderCloud/LinkedWiki/tarball/v0.2.0.0 --no-check-certificate
- mv v0.2.0.0 BorderCloud-LinkedWiki-v0.2.0.0.tgz
- tar vxzf BorderCloud-LinkedWiki-v0.2.0.0.tgz
- mv BorderCloud-LinkedWiki /var/www/fr.datatricorder.org/extensions/LinkedWiki
- vim /var/www/fr.datatricorder.org/LocalSettings.php
  - **Insérer à la fin du fichier** LocalSettings.php :
  - `require_once( "{$IP}/extensions/LinkedWiki/LinkedWiki.php" );`
  - `$smwgDefaultStore = "SMW_LinkedWikiStore";`
  - `$wgLinkedWikiEndPoint = "http://localhost:8081/sparql/"; // 4store endpoint SPARQL 1.1`
  - `$wgLinkedWikiGraphWiki = "http://fr.datatricorder.org/wiki/"; // graph name where the data of this wiki will be recorded`
  - `$wgLinkedWikiLocalEndPoint = "http://fr.datatricorder.org/sparql/"; // address of your endpoint`

-  Page d'administration de l'extension Semantic Mediawiki : on vérifie que le silo est accessible par le wiki et on remet à jour les données pour en avoir une copie dans le silo.

# La fin de votre LocalSettings.php devra ressembler à ça

```
//modifier en fonction de votre wiki
$wgScriptPath      = "";
$wgScriptExtension = ".php";
$wgArticlePath     = "/wiki/$1"; // car dans le http.conf : Alias /wiki "/var/www/fr.datatricorder.org/index.php"
$wgUsePathInfo     = true; // pour activer $wgArticlePath

// Pour Semantic Mediawiki
include_once("$IP/extensions/SemanticMediaWiki/SemanticMediaWiki.php");
enableSemantics('fr.datatricorder.org');

//Pour LinkedWiki
require_once( "{$IP}/extensions/LinkedWiki/LinkedWiki.php" );
$smwgDefaultStore = "SMW_LinkedWikiStore";
$wgLinkedWikiEndPoint = "http://localhost:8081/sparql/";
$wgLinkedWikiGraphWiki = "http://fr.datatricorder.org/wiki/";
// car dans le http.conf :redirectMatch ^/sparql(.*)$ /wiki/Special:SparqlEndpoint$1
$wgLinkedWikiLocalEndPoint = "http://fr.datatricorder.org/sparql/";

//activate subpage (pour la suite)
$wgNamespacesWithSubpages[NS_MAIN] = true;

//Pour ParserFunctions (#ifeq etc...) (pour la suite)
require_once( "$IP/extensions/ParserFunctions/ParserFunctions.php" );
$wgPFEnableStringFunctions = true;
```

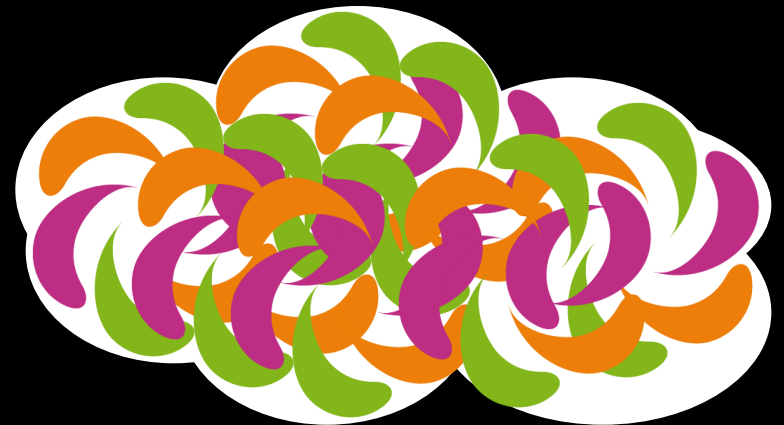
# Ce qu'il faut savoir :

- 🌀 Votre Wiki est enregistré dans une BDD MySQL (qu'il faut sauvegarder pour pouvoir restaurer le service)
- 🌀 Votre Silo Sparql est alimenté en temps réel par votre Wiki.
  
- 🌀 En cas de désynchronisation entre votre Wiki et votre silo, vous devez :
  - Arrêter votre silo : `/etc/init.d/datatricorderd stop`
  - Démarrer le backend de votre silo : `4s-backend datatricorder`
  - Supprimer le graph du wiki qui n'est plus synchronisé : `4s-delete-model datatricorder http://fr.datatricorder.org/wiki`
  - Redémarrer votre silo : `/etc/init.d/datatricorderd restart`
  - Aller sur la page Administration de SémanticMediawiki est mettre à jour les données.

# Références

- 🌀 Installer MediaWiki sur Ubuntu 10.04
  - [http://www.mediawiki.org/wiki/Manual:Installing\\_Mediawiki\\_on\\_Ubuntu\\_10.04](http://www.mediawiki.org/wiki/Manual:Installing_Mediawiki_on_Ubuntu_10.04)
- 🌀 Installer Semantic MediaWiki
  - <http://semantic-mediawiki.org/wiki/Help:Installation>
- 🌀 Installer LinkedWiki
  - <http://www.mediawiki.org/wiki/Extension:LinkedWiki>

**Démarrage d'un wiki :**  
**agrégation de données non structurées ?**

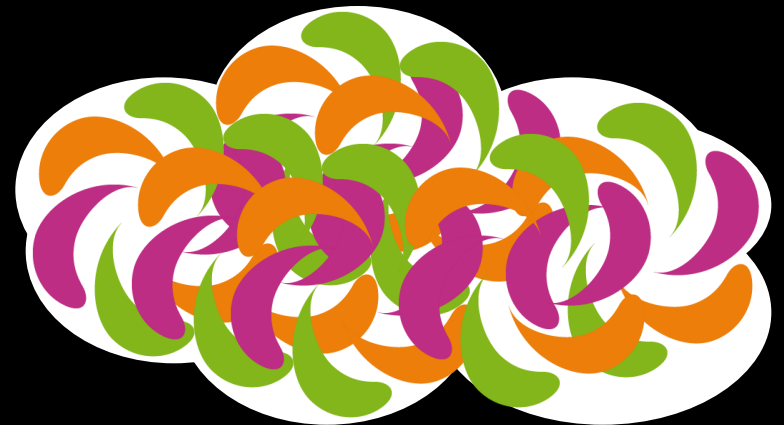


# Agrégation de données semi structurées



- 🌀 Création des fiches
  - Impact sur le silo ?
- 🌀 Création d'une catégorie comme dans Wikipedia
  - Impact sur le silo ?
- 🌀 Et première requête Sparql

# **Maturation du wiki : structuration dans sa langue**



# Structuration oui mais dans sa langue

---



- 🌀 Création d'une infobox
- 🌀 Création d'un template
- 🌀 Insertion des propriétés
- 🌀 Partage du template dans le reste du Wiki
- 🌀 On teste avec des requêtes Sparql

# Et mon tricorder ?

- 🌀 Les données structurées ou non peuvent être enregistrer par tous dans leurs langues.
- 🌀 Tout le monde peut rajouter des propriétés dans leurs langues
- 🌀 Les développeurs des tricorders peuvent :
  - créer de nouvelles requêtes
  - modifier les infobox pour demander de nouvelles données aux contributeurs
  - développer des bots pour alimenter les wikis automatiquement

Oui mais mon tricorder veut croiser les données francophones avec d'autres langues  
**\$lang.datatricorder.org. Comment faire ?**

# Références

## Pour commencer avec Mediawiki

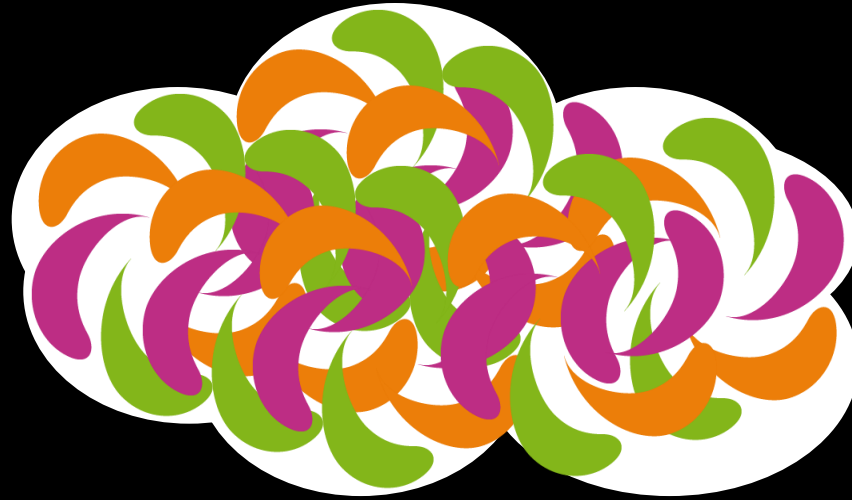
- [http://www.dailymotion.com/video/x6j8vn\\_la-syntaxe-sur-mediawiki-tutoriel-w\\_tech](http://www.dailymotion.com/video/x6j8vn_la-syntaxe-sur-mediawiki-tutoriel-w_tech)

## Utiliser Semantic MediaWiki

- <http://semantic-mediawiki.org>

## Utiliser LinkedWiki

- <http://www.mediawiki.org/wiki/Extension:LinkedWiki>



# **Partage des données du wiki : réalignement**

# Le problème



- Fr.datatricorder.org référence les silos SPARQL disponibles dans le monde.
- Il existe déjà une ontologie pour harmoniser la découverte de silos SPARQL :  
The Vocabulary of Interlinked Datasets  
(voID <http://vocab.deri.ie/void/guide>)

Comment aligner fr.datatricorder.org sur l'ontologie voID de manière invisible pour les contributeurs du Wiki ?

# En résumé ?

## Point de vue requêtes

 Mon tricorder veut pouvoir faire cette requête :

```
SELECT ?dataset ?endpoint ?numberOfTriples
WHERE {
  ?dataset rdf:type void:Dataset ;
           void:sparqlEndpoint ?endpoint ;
           void:statItem [
                                 scovo:dimension void:numberOfTriples ;
                                 rdf:value ?numberOfTriples;
                             ]
}
```

# La même requête dans le wiki avant réalignement


🌀 Pour le moment, c'est plutôt ça :

```
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
PREFIX cat:<http://fr.datatricorder.org/wiki/Spécial:URIResolver/Catégorie:>
```

```
PREFIX prop:<http://fr.datatricorder.org/wiki/Spécial:URIResolver/Attribut:>
```

```
select ?silo ?nbtriplet ?lien {  
    ?silo rdf:type cat:Silo_de_données ;  
        <http://fr.datatricorder.org/wiki/Spécial:URIResolver/Attribut:Point_d'accès> ?lien ;  
        prop:Nombre_de_triplets ?nbtriplet  
}
```



On commence par la structure :  
un nœud [ ] ? (blank node)

---

- 🌀 Chaque implémentation a son interprétation de ce que doit être un « blank node »
- 🌀 Pour 4Store, c'est une URI
- 🌀 Pour un wiki, une URI = une page
- 🌀 Pour un humain, une page d'une autre page est une sous-page (subpage)
  - Activation de l'option  
`$wgNamespacesWithSubpages[NS_MAIN] = true;`

# Modification invisible : Vive la Transclusion !



- 🌀 La sous-page sera automatiquement visible dans l'infobox `{{:DBpedia/Nombre de triplets}}`
- 🌀 On ajoute les nouvelles propriétés et on le masque avec un template pour automatiser la création d'une sous-page :  
Valeur, Dimension & Détail\_du\_silo\_de\_données
- 🌀 On ne modifie que le template de l'infobox pour impacter les changements dans tout le Wiki
  - Installation de l'extension ParserFunction pour traiter cette transclusion dans le template.  

```
require_once( "$IP/extensions/ParserFunctions/ParserFunctions.php" );  
$wgPFEnableStringFunctions = true;
```



# Maintenant ?

## Avec la bonne structure

PREFIX rdf:<<http://www.w3.org/1999/02/22-rdf-syntax-ns#>>

PREFIX cat:<<http://fr.datatricorder.org/wiki/Spécial:URIResolver/Catégorie:>>

PREFIX prop:<<http://fr.datatricorder.org/wiki/Spécial:URIResolver/Attribut:>>

```
select ?silo ?nbtriple ?lien {  
  ?silo rdf:type cat:Silo_de_données ;  
  <http://fr.datatricorder.org/wiki/Spécial:URIResolver/Attribut:Point\_d'accès> ?lien ;  
  prop:Détail_du_silo_de_données ?dimensionnbtriple .  
  ?dimensionnbtriple prop:Dimension <http://vocab.deri.ie/void#numberOfTriples> ;  
  prop:Valeur ?nbtriple .  
}
```

Il reste maintenant à réaligner les attributs...facile !

# Réalignement Objectifs



<b>Avant</b>	<b>Après</b>
cat:Silo_de_données	void:Dataset
prop:Valeur	rdf:value
prop:Dimension	scovo:dimension
prop:Détail_du_silo_de_données	void:statItem
<a href="http://fr.datatricorder.org/wiki/Special:URIRresolver/Attribut:Point_d'accès">http://fr.datatricorder.org/wiki/Special:URIRresolver/Attribut:Point_d'accès</a>	void:sparqlEndpoint

# Comment importer une ontologie dans un wiki



 Vous devez :

- Être connecté sous le login WikiSysop
- Créer la page « MediaWiki:Smw import **rdf** » dont le titre contient l'acronyme de votre ontologie
- Mettre dans cette page les propriétés à faire exister dans votre wiki (les espaces en début de lignes sont obligatoires sauf pour la première ligne)

Prefix\_de\_l'ontologie|[lien\_vers\_de\_la\_doc nom de l'ontologie]  
nom\_de\_la\_propriété1\_dans\_l'ontologie|type\_dans\_le\_wiki  
nom\_de\_la\_propriété2\_dans\_l'ontologie|type\_dans\_le\_wiki

# Réalignement

## Phase 1 : importation des ontologies nécessaires



### MediaWiki:Smw import scovo

[http://purl.org/NET/scovo#\[http://vocab.deri.ie/scovo The Statistical Core Vocabulary \(SCOVO\)\]](http://purl.org/NET/scovo#[http://vocab.deri.ie/scovo The Statistical Core Vocabulary (SCOVO)])  
dimension|Type:URI

### MediaWiki:Smw import rdf

[http://www.w3.org/1999/02/22-rdf-syntax-ns#\[http://www.w3.org/TR/2004/REC-rdf-primer-20040210 Resource Description Framework \(RDF\) \]](http://www.w3.org/1999/02/22-rdf-syntax-ns#[http://www.w3.org/TR/2004/REC-rdf-primer-20040210 Resource Description Framework (RDF) ])  
value|Type:Number

### MediaWiki:Smw import void

[http://rdfs.org/ns/void#\[http://vocab.deri.ie/void/guide Vocabulary of Interlinked Datasets \(voiD\)\]](http://rdfs.org/ns/void#[http://vocab.deri.ie/void/guide Vocabulary of Interlinked Datasets (voiD)])  
Dataset|Category  
statItem|Type:Page  
sparqlEndpoint|Type:URL

# Pourquoi ne pas importer les ontologies entières

- ❁ Par expérience, un silo a une performance inversement proportionnelle
  - aux nombres de triplets (de moins en moins vrai)
  - Aux structures complexes des ontologies (car les requêtes seront alors très très complexes)
  - Aux nombres d'attributs (en fonction de l'implémentation des silos)

## Conclusion :

**Inutile de plomber le silo avec des ontologies qui n'ont encore aucune donnée à partager avec d'autres machines.**

# Réalignement

## Phase 2 : modification des pages des attributs ou des catégories à aligner

- 🌀 Dans la page catégorie « Silo\_de\_données » insérer `[[Importé de::void:Dataset]]`
- 🌀 Dans la page attribut « Valeur » insérer `[[Importé de::rdf:value]]`
- 🌀 Dans la page attribut « Dimension » insérer `[[Importé de::scovo:dimension]]`
- 🌀 Dans la page attribut « Détail\_du\_silo\_de\_données » insérer `[[Importé de::void:statItem]]`
- 🌀 Dans la page attribut « Point\_d'accès » insérer `[[Importé de::void:sparqlEndpoint]]`

# Et voilà ! Le tricorder peut fonctionner !



```
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX void:<http://rdfs.org/ns/void#>
PREFIX scovo:<http://purl.org/NET/scovo#>
SELECT ?dataset ?endpoint ?numberOfTriples
WHERE {
    ?dataset rdf:type void:Dataset ;
              void:sparqlEndpoint ?endpoint;
              void:statItem ?item.
    ?item    scovo:dimension void:numberOfTriples ;
            rdf:value ?numberOfTriples.
}
```

Il reste à mettre à jour la page `Liste_des_silos` qui décrit cette requête pour indiquer aux développeurs ou machines abonnées à cette page que la requête vient de changer.

Il reste à faire...



... les tricorders !!



# Références :

---



 Un autre exemple de réalignement :

– Des tricorders dans des supermarchés ?

[http://www.youtube.com/view\\_play\\_list?p=084E8D512A1FED2B](http://www.youtube.com/view_play_list?p=084E8D512A1FED2B)

Merci !



Des questions ?

Karima.rafes@bordercloud.com

<http://www.bordercloud.com>

Web 3.0 Hosting & Factory