

<b>AGORA</b>		Version 1.2
Y10		DATEDDDD

## ORGANISATION DU SERVEUR SVN G2

- Prise en compte dans la version : .....
- Numéro de fiche : aucun
- Date de diffusion : ..../..../....

### Historique des modifications / Gestion des versions :

VERSION	DATE DE VALIDATION	NOM DU VALIDEUR	OBJET DES MODIFICATIONS
0.91	17/02/2010	IAN FRENS	AJOUT DU § 3.4
1.0	20/04/2010		MISE À JOUR DES ROLES ET RESPONSABILITES (§1.3, §2.2)  AJOUT DU REPOSITORY SOCLEG2 (§2.3, §3.2)
1.1	23/04/2010		AJOUT DES RESTRICTIONS SUR :  TARGET, .SETTINGS, .CLASSPATH, .PROJECT, LOGBACK-TEST.XML ET LE POM.XML DU PROJET PARENT
1.2	17/06/2010		§1.5 : AJOUT DU CAS DES PANNEES DE DISQUES DURS.



Informatique  
Institutionnelle

A G O R A

Réf. :

## ORGANISATION DU SERVEUR SVN G2

### ANALYSE DETAILLEE

**Résumé :** Description détaillée des choix pris quand à l'organisation et l'utilisation du serveur Subversion pour les applications Web G2.

**Rédacteur :** Fabrice-Emmanuel Ory, Matthieu Bernis (§3.3), Yann CRUEGHE (v1.0 & 1.1)

**Vocabulaire :**

**SCM = Software Configuration Management, outil de Gestion de Configuration du Logiciel**

**VCS = Version Control System, sous-partie fonctionnelle d'un SCM.**

**SVN = Subversion, outil de VCS.**

**Repository = premier niveau de partition du serveur SVN.**

**Projet SVN = équivaut au module CVS. Est le 2<sup>ème</sup> niveau de partition des composants hébergés par SVN.**

<b>AGORA</b>	N° du Tome et libellé s'il y a lieu ?	1
Y10		DATEDDDD

## **SOMMAIRE**

<b>1.</b>	<b>OBJECTIFS.....</b>	<b>2</b>
1.1	AUTOMATISER.....	2
1.2	ADMINISTRATION MINIMALE.....	2
1.3	SÉCURITÉ.....	3
1.4	UTILISATION.....	5
1.5	SAUVEGARDES.....	6
<b>2.</b>	<b>ORGANISATION.....</b>	<b>7</b>
2.1	SERVEURS.....	7
2.2	RÔLES ET RESPONSABILITÉS.....	7
2.3	NATURE DES REPOSITORIES.....	8
2.4	ORGANISATION DES REPOSITORIES G2 METIER ET AGL.....	9
<b>3.</b>	<b>ADMINISTRATION.....</b>	<b>11</b>
3.1	CAS D'UTILISATION.....	11
3.2	ADMINISTRATION PAR REPOSITORY.....	11
3.3	OUTIL D'ADMINISTRATION.....	12
3.4	Montée de version du serveur SVN.....	12

<b>AGORA</b>	N° du Tome et libellé s'il y a lieu ?	2 - 00 - 00
Y10		DATEDDDD

## 1. OBJECTIFS

Ce document définit une organisation du serveur SVN qui répond aux besoins suivants :

- ◆ Automatiser le plus possible les échanges avec le serveur SVN. Réduire les interventions humaines.
- ◆ Limiter l'administration du serveur SVN aux seules actions qui ne sont pas automatisables.
- ◆ Réduire les risques en interdisant les actions sensibles.
- ◆ Préciser le bon usage d'un VCS dans le cadre d'un processus global de construction des packages à diffuser.
- ◆ Apporter des éléments de réponse aux besoins de sauvegarde des composants sources.

### 1.1 Automatiser

Les actions suivantes sont prises en charge par les outils, assistants ou robots déclenchés à la demande côté client (depuis Eclipse ou par Y20), sans passer par la ligne de commande :

- ◆ ajout et commit des composants textes et binaires,
- ◆ développements en parallèle (création/utilisation d'une branche),
- ◆ checkout multiples,
- ◆ fusion et comparaison de fichiers,
- ◆ pose d'un tag.

### 1.2 Administration minimale

Les équipes de développement métier G2 adressent une demande JIRA pour spécifier les personnes à affecter à chaque rôle (ex. : responsable métier). L'administrateur AGL affecte alors les utilisateurs spécifiés dans le rôle approprié.

Chaque équipe technique (AGL, BDD et STC) est responsable de l'administration de son propre repository :

- ◆ définition des groupes d'utilisateurs,
- ◆ restriction ou non des accès par path,
- ◆ création d'une branche,

<b>AGORA</b>	N° du Tome et libellé s'il y a lieu ?	3 - 00 - 00
Y10		DATEDDDD

- ◆ pose d'un tag,
- ◆ création de niveaux intermédiaires dans le système de fichiers,
- ◆ etc.

Le repository TEST est administré par les développeurs de l'équipe AGL. Seul l'administrateur SVN peut créer un nouveau repository.

### 1.3 Sécurité

Dans la solution cible, par défaut, tout utilisateur SVN identifié au LDAP institutionnel peut publier une nouvelle version de fichier (commit) ou un nouveau fichier (add).

Quelque soit le profil, les utilisateurs ne peuvent pas :

- ◆ commiter leurs modifications sans spécifier de commentaires.
- ◆ commiter le répertoire target du projet parent et des sous-projets maven
- ◆ commiter les ressources .settings, .classpath, .project et logback-test.xml

Les développeurs fonctionnels ne peuvent pas :

- ◆ commiter le fichier logback.xml,
- ◆ commiter le fichier pom.xml du projet parent,
- ◆ ajouter ou supprimer un tag, ni supprimer une branche,
- ◆ publier une nouvelle version du fichier *logback.xml* des applications métier,
- ◆ commiter leur travail ailleurs que sur le trunk et les branches des applications.

Les responsables métiers peuvent réaliser ces cas d'utilisation à l'exception de :

- ◆ supprimer un tag,
- ◆ supprimer une branche.

La sécurité des autres repositories est de la responsabilité des acteurs suivants :

- ◆ l'équipe du nom du repository ; par exemple l'équipe STC pour le repository STC,
- ◆ l'équipe AGL pour le repository TEST,
- ◆ les responsables applicatifs (R.A. AGORA) adressent des demandes à l'équipe Système pour administrer les repositories non G2.

<b>AGORA</b>	N° du Tome et libellé s'il y a lieu ?	4 - 00 - 00
Y10		DATEDDDD

Le tableau ci-dessous présente les fonctionnalités accessibles par rôle :

Fonctionnalité	Développeur métier	Responsable métier	Outils AGL	Admin SVN
Créer un repository	NON	NON	NON	OUI
Commit METIER (hors logback.xml)	OUI (trunk et branches)	OUI (partout)	OUI (partout)	OUI (partout)
Commit METIER du fichier logback.xml	NON	OUI	OUI	OUI
Commit METIER sans commentaires	NON	NON	NON	NON
Commit METIER du répertoire target de Maven	NON	NON	NON	NON
Commit METIER des .settings, .classpath, .project, logback-test.xml	NON	NON	NON	NON
Commit METIER du pom.xml du projet parent	NON	OUI	OUI	OUI
Création d'une branche	OUI	OUI	OUI	OUI
Suppression d'une branche	NON	NON	OUI	OUI
Ajout d'un tag	NON	OUI	OUI	OUI
Suppression d'un tag	NON	NON	OUI	OUI

La mise en œuvre de ces fonctionnalités est assurée par l'intermédiaire d'un hook de type pre-commit et de restrictions d'accès par profil au niveau des différents répertoires (trunk/tags/branches).

Les tableaux ci-dessous présentent les différents types de verrous positionnés par profil et par fonctionnalité :

**Tableau 1 - Hors fichier logback.xml et répertoire target**

Fonctionnalité	Arborescence de base				Branches				Tags				Trunk						
	evenement/ trunk/ branches/ tags/				C	R	U	D	C	R	U	D	C	R	U	D	C	R	U
Développeur métier	C	C	C	C	C	C	C	C	H	C	C	C	C	C	C	C	C	C	C

<b>AGORA</b>	N° du Tome et libellé s'il y a lieu ?	5 - 00 - 00
Y10		DATEDDDD

Responsable métier	C	C	C	C	C	C	C	H	C	C	C	H	C	C	C	C	C
Outil AGL	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
Administrateur	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
<aucun des 4>	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
<b>C(onfiguration)</b>	Fonctionnalités gérées par la configuration des droits R/W sur les répertoires																
<b>H(ook)</b>	Fonctionnalités gérées par un hook																

Tableau 2 - Cas particulier des fichiers logback.xml et pom.xml (parent)

Fonctionnalité	Arborescence de base				Branches				Tags				Trunk			
	evenement/ trunk/ branches/ tags/	C	R	U	D	C	R	U	D	C	R	U	D	C	R	U
Développeur métier	C	C	C	C	H	C	H	H	C	C	C	C	H	C	H	H
Responsable métier	C	C	C	C	C	C	C	H	C	C	C	H	C	C	C	C
Outil AGL	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
Administrateur	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
<aucun des 4>	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
<b>C(onfiguration)</b>	Fonctionnalités gérées par la configuration des droits R/W sur les répertoires															
<b>H(ook)</b>	Fonctionnalités gérées par un hook															

Tableau 3 - Cas particulier des ressources target, .classpath, .settings, .project et logback-test.xml

Fonctionnalité	Arborescence de base				Branches				Tags				Trunk			
	evenement/ trunk/ branches/ tags/	C	R	U	D	C	R	U	D	C	R	U	D	C	R	U
Développeur métier	H	C	H	H	H	C	H	H	H	C	H	H	H	C	H	H
Responsable métier	H	C	H	H	H	C	H	H	H	C	H	H	H	C	H	H

<b>AGORA</b>	N° du Tome et libellé s'il y a lieu ?	6 - 00 - 00
Y10		DATEDDDD

Outil AGL	H	C	H	H	H	C	H	H	H	C	H	H	H	C	H	H
Administrateur	H	C	H	H	H	C	H	H	H	C	H	H	H	C	H	H
<aucun des 4>	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
<b>C</b> (onfiguration)	Fonctionnalités gérées par la configuration des droits R/W sur les répertoires															
<b>H</b> (ook)	Fonctionnalités gérées par un hook															

## 1.4 Utilisation

Un VCS n'est pas un outil de sauvegarde (back-up) des composants en cours de développement. Les composants publiés dans Subversion doivent être profitables à l'ensemble des acteurs du projet en développement.

Pour ce faire, les fichiers publiés doivent suivre les règles syntaxiques de leurs formats respectifs (exemple : compiler) et avoir été testés unitairement par leurs auteurs.

Le serveur d'intégration continue garantit cette règle.

## 1.5 Sauvegardes

Le besoin de sauvegarde a 3 origines :

- ◆ les chutes des PC placés en position verticale,
- ◆ les pannes de disque dur,
- ◆ les erreurs des développeurs.

Le VCS ne peut pas être la solution à ce besoin de sauvegarde. En effet, les repositories ainsi utilisés sont pollués par des fichiers privés qui ne répondent pas à la qualité attendue :

- ◆ l'intégration continue travaille inutilement et signale de nombreuses erreurs,
- ◆ les développeurs récupèrent des composants qui ne sont pas utiles à leur travail,
- ◆ les statistiques extraites du VCS sont faussées.

Voici des éléments de réponse pour éviter les pertes de données dues à la chute du PC :

- ◆ Placer le PC au sol, en position verticale seulement si vous pouvez l'encadrer par des meubles stables des deux côtés. Vous évitez ainsi les coups de pied qui font basculer le PC.
- ◆ Placer le PC en position horizontale, sur votre bureau.
- ◆ Remplacer les disques durs à plateaux et têtes de lecture/écriture par des disques SSD (à mémoires). Quand le PC tombe sur le côté, les données du disque SSD ne sont pas perdues.

<b>A G O R A</b>	N° du Tome et libellé s'il y a lieu ?	7 - 00 - 00
Y10		DATEDDDD

Les pannes de disque dur peuvent être contrecarrées par :

- ◆ Par l'utilisation de disques répliqués en RAID 1 sur le poste local.
- ◆ En sauvegardant l'espace D: (back-up du NAS) des utilisateurs CITRIX.
- ◆ Par une demande de récupération des données du disque en panne.

Le besoin lié aux erreurs des développeurs peut s'adresser ainsi :

- ◆ Décomposer les développements en micro tâches : organiser le travail de façon à pouvoir se synchroniser au repository une fois par jour.
- ◆ Pour les développements plus longs :
  - o utiliser un disque réseau sauvegardé par l'entreprise,
  - o ou utiliser une clé USB.

<b>AGORA</b>	N° du Tome et libellé s'il y a lieu ?	8 - 00 - 00
Y10		DATEDDDD

## 2. ORGANISATION

### 2.1 Serveurs

Actuellement, 2 serveurs SVN sont disponibles :

- ◆ Gliese est le serveur de Production :
  - SVN Collabnet version 1.6.5.
  - Uri racine de tous les repositories : <http://gliese.agora.msanet/svn/>.
  - Les données de ce serveur sont sauvegardées par un outil de back-ups planifiés.
  - Ce serveur mutualisé n'est pas dédié aux seuls composants des applications G2. En effet, il héberge des composants non G2.
- ◆ Tetris est le serveur d'essais, de tests et d'intégration :
  - SVN Collabnet version 1.6.5.
  - Uri racine de tous les repositories : <http://tetris.agora.msanet:16060/svn/>

La continuité de service n'est pas garantie sur ce serveur.

### 2.2 Rôles et responsabilités

- ◆ L'administration fonctionnelle du serveur SVN est de la responsabilité de l'équipe AGL pour les seuls repositories G2. L'équipe AGL assure les interventions suivantes :
  - création d'un nouveau repository,
  - ajout d'un utilisateur à un groupe,
  - maintenance des hooks.
- ◆ L'administration technique est prise en charge par l'équipe Système.
- ◆ Le développeur fonctionnel peut :
  - ajouter un project SVN dans le repository METIER (à l'aide du futur assistant de prise en maintenance),

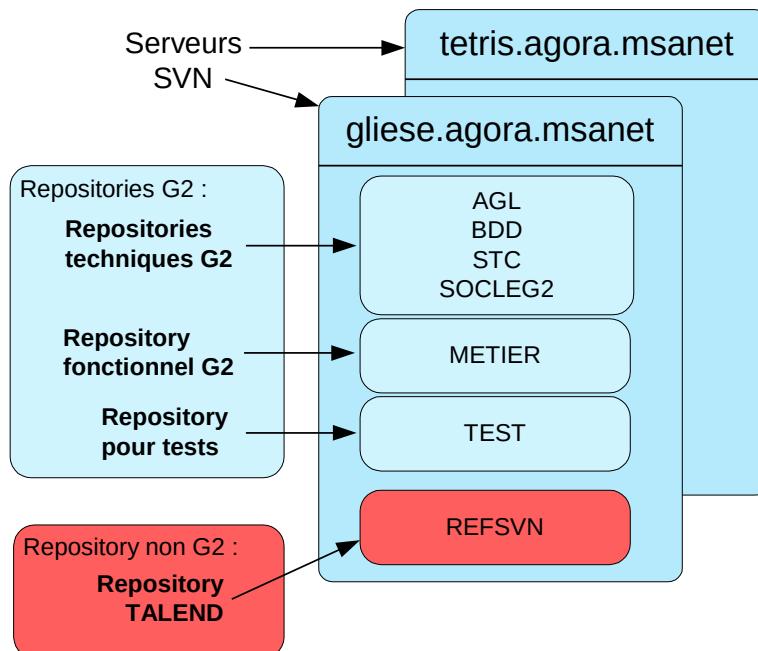
<b>AGORA</b>	N° du Tome et libellé s'il y a lieu ?	9 - 00 - 00
Y10		DATEDDDD

- o publier (commit) ou ajouter (add) des fichiers dans les projets du repository METIER,
- o créer une branche.
- ◆ Le responsable métier intervient pour :
  - o Créer l'arborescence de base d'un nouveau projet,
  - o publier une nouvelle version du fichier *logback.xml*,
  - o poser un tag de version fonctionnelle,
  - o fusionner 2 branches de développement.

### 2.3 Nature des repositories

Le serveur gliese est mutualisé : il héberge actuellement des repositories pour les applications G2 et un repository pour l'application TALEND.

Voici le détail des repositories disponibles :



- ◆ les applications fonctionnelles G2 déposent leurs composants dans le repository METIER,
- ◆ les applications techniques G2 sont organisées par équipe : un repository est disponible par équipe technique : repository AGL pour les composants développés par l'équipe AGL et ainsi de suite pour BDD, STC et SOCLEG2,

<b>AGORA</b>	N° du Tome et libellé s'il y a lieu ?	10 - 00 - 00
Y10		DATEDDDD

- ◆ le repository TEST est utilisé pour les développements et l'administration SVN, à des fins de tests, quand le serveur SVN tetris n'est plus disponible : par défaut, le serveur tetris doit être utilisé pour réaliser ces tests,
- ◆ le repository REFSVN est utilisé en mode boite noire par l'ETL TALEND.

Remarque :

L'administration des repositories non G2 est à la charge des responsables applicatifs (R.A.) qui sont à l'origine de ces repositories. Pour ce faire, les équipes fonctionnelles adressent des demandes à l'équipe Système qui exécute les actions d'administration. Exemple : l'équipe Infocentre est responsable du repository REFSVN, administré par l'équipe Système.

## 2.4 Organisation des repositories G2 METIER et AGL

La description qui suit ne s'applique qu'aux repositories G2 METIER et AGL.

Les repositories BDD, STC et SOCLEG2 sont administrés et organisés par les équipes respectives BDD et STC. Ce document ne décrit pas leurs organisations.

Le repository REFSVN est utilisé en mode boite noire par l'application TALEND. L'organisation de ce repository n'est pas décrite dans ce document.

La structure préconisée pour les applications métier G2 et utilisée pour les développements AGL est la suivante :

```
<url racine>/<Repository>/<événement AGORA>/branches/<nom de la branche>/<structure metier G2>
<url racine>/<Repository>/<événement AGORA>/trunk/<structure metier G2>
<url racine>/<Repository>/<événement AGORA>/tags/<nom du tag>/<structure metier G2>
```

Avec :

<url racine> =	<a href="http://gliese.agora.msanet svn">http://gliese.agora.msanet svn</a>
<Repository> =	METIER ou AGL
<structure metier G2> =	pom.xml
	+ <événement AGORA>_ear/<composants ear>
	+ <événement AGORA>_metier/<composants métier>
	+ <événement AGORA>_web/<composants webapp>
<composants ear> =	pom.xml
	+ src/main/
<composants métier> =	pom.xml
	+ src/main/java
	+ src/main/resources
	+ src/test/java
	+ src/test/resources
<composants webapp> =	pom.xml
	+ src/main/java

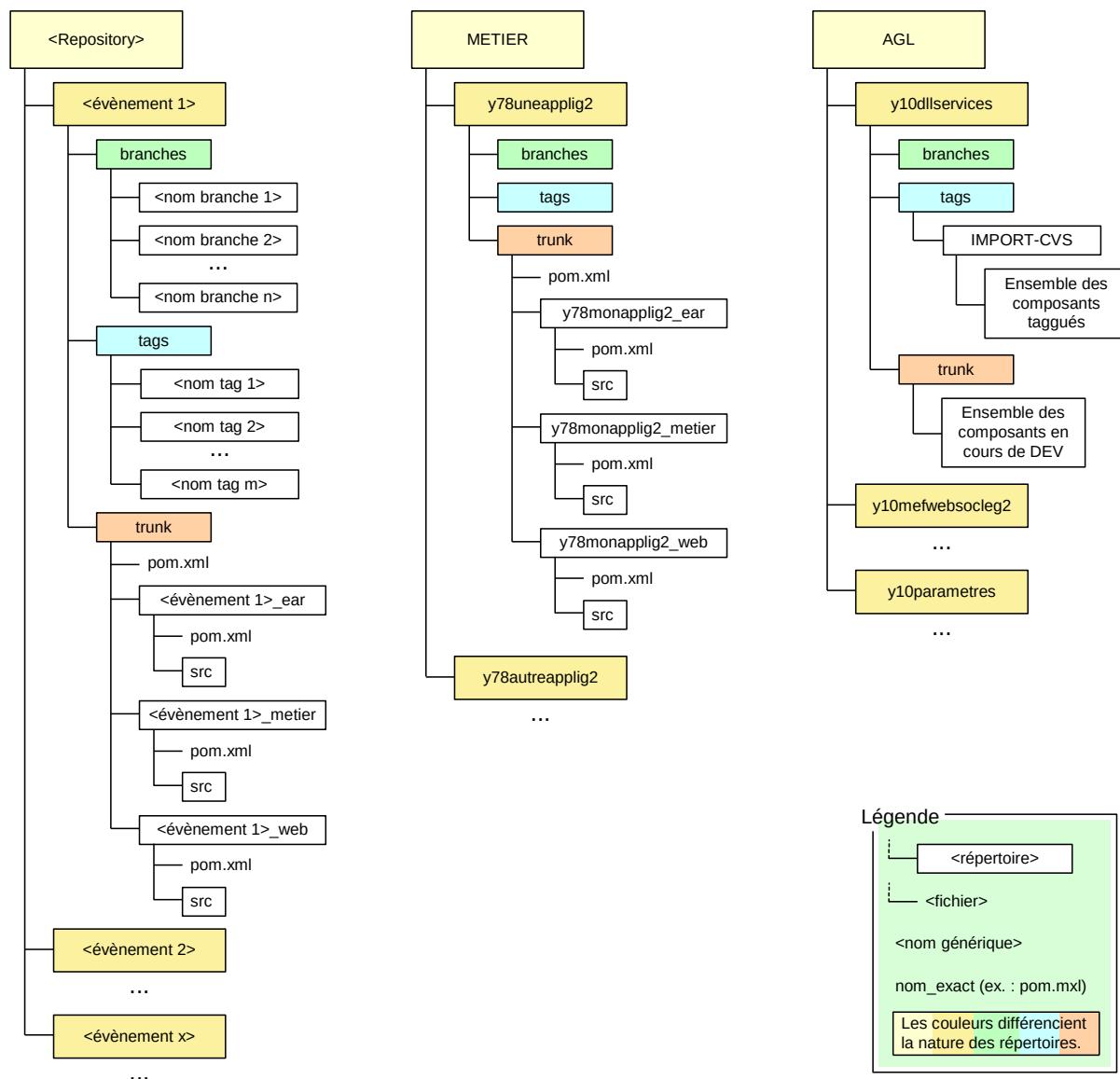
<b>AGORA</b>	N° du Tome et libellé s'il y a lieu ?	11 - 00 - 00
Y10		DATEDDDD

+ src/main/resources  
+ src/main/webapp  
+ src/test/java  
+ src/test/resources

<b>AGORA</b>	N° du Tome et libellé s'il y a lieu ?	12 - 00 - 00
Y10		DATEDDDD

Le schéma ci-dessous présente :

- ◆ l'organisation type à gauche,
- ◆ un développement métier en cours dans trunk, au centre,
- ◆ un développement AGL en cours et le même taggué IMPORT-CVS, dans la colonne de droite :



#### Remarque :

Pour simplifier la présentation, seuls les principaux répertoires et fichiers ont été représentés. Ainsi, les composants des répertoires src n'apparaissent pas dans les schémas présentés ci-dessus.

<b>AGORA</b>	N° du Tome et libellé s'il y a lieu ?	13 - 00 - 00
Y10		DATEDDDD

### 3. ADMINISTRATION

#### 3.1 Cas d'utilisation

L'outil d'adminstration choisi, VoilaSVN, répond aux cas d'utilisation suivants :

- ◆ possibilité de créer des repositories à des niveaux différents du système de fichier du serveur,
- ◆ définition des utilisateurs et des groupes,
- ◆ définition des droits par groupe, par utilisateur et par chemin d'accès,
- ◆ possibilité de s'authentifier au LDAP institutionnel.

La création de la structure standard (trunk, branches et tags) se fait manuellement ; l'outil identifie et affiche alors la nature spécifique de chacun de ces répertoires.

#### 3.2 Administration par repository

Le compte administrateur est le même sur les 2 serveurs, gliese et tetris :

- ◆ Login : svnadmin
- ◆ Mot de passe : demander à l'équipe Système.

L'administration des repositories G2 est prévue ainsi sur gliese :

Repository	Utilisation	Administré par l'équipe
METIER	Stocker les composants des applications fonctionnelles G2.	AGL
AGL	Stocker les composants techniques AGL G2.	AGL
BDD	Stocker les composants techniques BDD G2.	BDD
STC	Stocker les composants techniques STC G2.	STC
SOCLEG2	Stocker les composants techniques SOCLE G2.	STC
TEST	Espace de test de secours si le serveur tetris est indisponible.	AGL
<non G2>	Cf. responsable fonctionnel à l'origine de la création du repository.	Système

Remarques :

<b>AGORA</b>	N° du Tome et libellé s'il y a lieu ?	14 - 00 - 00
Y10		DATEDDDD

- ◆ Les repositories non G2, placé sur le serveur Gliese dans un but de mutualisation sont administrés par l'équipe Système.
- ◆ L'équipe AGL répond au besoin d'un nouveau repository G2.

### 3.3 Outil d'administration

Plusieurs outils d'administration ont été essayés sur le serveur d'intégration tetris :

- ◆ SvnControl,
- ◆ USvn,
- ◆ VoilaSVN.

Il en existe d'autres (Submin, AccessSvn, SvnManager, ...). La plupart des outils fonctionne avec la notion de branches, tags et trunk au niveau du repository, et non pas sur des sous-niveaux. Ils permettent ainsi de gérer les projets comme autant de repositories sur une même racine du système de fichiers.

Ceci ne correspond pas au fonctionnement souhaité par Agora, à savoir 5 repositories AGL, STC, SOCLEG2, BDD et METIER ; et chaque application est découpée en trunk, branches et tags.

VoilaSVN permet de gérer les groupes, les utilisateurs, les droits, la création de dossier sur plusieurs niveaux et tout cela sur différents repositories.

C'est une application Web qui nécessite l'installation d'un serveur Tomcat 6 ou équivalent.

Le manuel utilisateur de VoilaSVN se trouve ici : [http://voilasvn.com/help/2\\_0/](http://voilasvn.com/help/2_0/).

### 3.4 Montée de version du serveur SVN

La mutualisation de l'ensemble des repositories AGORA sur la même machine, gliese, et la même instance SVN a pour impacts :

- ◆ une adhérence totale entre tous les repositories de l'entreprise,
- ◆ la configuration globale du serveur SVN est impossible ; il est obligatoire de configurer chaque repository,
- ◆ une étude d'impacts détaillée et des tests de non régression poussés conditionnent toute montée de version du serveur SVN,
- ◆ cette étude d'impacts est rendue difficile par l'utilisation d'un partie des repositories en mode boîte noire,

<b>AGORA</b>	N° du Tome et libellé s'il y a lieu ?	15 - 00 - 00
Y10		DATEDDDD

- ◆ la version du serveur SVN risque fort d'être figée de même que la version des clients connectés à ces repositories (plug-in Subclipse par exemple).

Par conséquent, le responsable fonctionnel, les acteurs Socle et AGL du projet G2 demandent l'installation d'une instance SVN dédiée sur le serveur de Production.