

## **PROJET**

### **CGLLS TMA Harmonia**

### **Spécifications techniques détaillées**

## **Document de spécifications**

<b>Client :</b>	CGLLS	<b>Fournisseur :</b>	Capgemini
<b>Référence :</b>	SPF_002/1.7		
<b>Date de référence :</b>	01/01/2010	<b>Date de mise à jour :</b>	12/02/2020
<b>Etat :</b>	<input type="checkbox"/> Draft	<input checked="" type="checkbox"/> à valider	<input type="checkbox"/> validé (diffusable)
<b>Diffusion :</b>	<input type="checkbox"/> interne	<input checked="" type="checkbox"/> contrôlée	<input type="checkbox"/> libre

## **Diffusion**

Destinataire	Société	Fonction
Catherine ROUARD Christian BERT	CGLLS	MOA

Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

#### CIRCUIT DE VALIDATION

Version	Rédaction		Vérification		Approbation	
	Nom	Date	Nom	Date	Nom	Date
1.0	Wilfried CHARPENTIER	27/01/2010	Thomas CLARISSE		Pascal MIARA	
1.1	Wilfried CHARPENTIER	27/01/2010	Thomas CLARISSE		Pascal MIARA	
1.2	Wilfried CHARPENTIER	03/05/2010	Thomas CLARISSE		Pascal MIARA	
1.3	Wilfried CHARPENTIER	27/01/2012	Guillaume LEY		Pascal MIARA	
1.4	Adel CHEBBINE	05/04/2013	Anne-Marie TASSEL		Pascal MIARA	
1.5	Antoine BASILE	23/10/2013	David DURIEUX	28/10/2013	David DURIEUX	28/10/2013
1.6	Mohamed DAHER ABDI	16/05/2017	David DURIEUX	16/05/2017	David DURIEUX	16/05/2017
1.7	Simon DOUILLET	12/02/2020				

#### HISTORIQUE DES ÉVOLUTIONS

Version	Objet de la version (citer les fiches de réception de document prises en compte)
1.0	Reprise des spécifications techniques détaillées du précédent marché au nouveau format du système qualité d'Avantias.
1.1	Spécifications des évolutions 2010
1.2	Spécifications des évolutions 2010 (lot 2)
1.3	Spécifications des évolutions 2011 et 2012
1.4	Spécifications des évolutions 2013 + actualisation des éléments de dimensionnement
1.5	Mise à jour de forme suite au passage TMA 2013 + ajout des éléments relatifs à la migration d'infrastructure de la Production
1.6	Spécifications des évolutions 2017
1.7	Spécification des évolutions 2020, migration serveurs (Windows server 2016) et BDD (MSSQL 2017)

#### DOCUMENTS RÉFÉRENCES

Origine	N°	Titre	Référence	Usage (*)
Capgemini	[1]	Plan de Gouvernance Projet	PGP_CGLLS_TMA_Harmonia v1.0.docx	Document de référence
Capgemini	[2]	Spécifications fonctionnelles détaillées	SPF00117 Spécifications fonctionnelles détaillées 1.10.doc	À lire au préalable
Capgemini	[3]	Spécifications techniques détaillées	SPF00117 Spécifications fonctionnelles détaillées 1.5.doc	À lire au préalable
Capgemini	[4]	Règles de nommage des	Règles de nommage des formulaires	À lire au préalable



Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

		formulaires Excel	Excel 1.9.doc	
--	--	-------------------	---------------	--

(\*) Indiquer le contexte de citation du document : à lire au préalable, document applicable, document de référence, document complémentaire, ...

Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

## SOMMAIRE

<b>1</b>	<b><u>OBJET DU DOCUMENT</u></b>	<b>7</b>
<b>2</b>	<b><u>ARCHITECTURE APPLICATIVE</u></b>	<b>7</b>
2.1	Déploiement des serveurs .....	7
2.2	Architecture réseau .....	7
2.3	Communication entre les serveurs .....	8
2.4	Poste Client.....	10
2.4.1	Système d'exploitation (OS = Operating System) et navigateurs supportés .....	10
2.4.2	Configuration écran.....	10
2.4.3	Ouverture sur la mobilité.....	10
2.5	Macro-Architecture des composants d'un serveur.....	10
2.6	Caractéristique des serveurs .....	11
2.7	Serveur Web .....	12
2.8	Serveur de base de données .....	12
2.9	Serveur applicatif.....	12
2.9.1	Composants techniques .....	14
2.9.2	Normes implémentées.....	14
2.9.3	Spring 2.5.....	14
2.9.4	Hibernate 3.3 .....	15
2.10	Sécurité .....	15
2.10.1	Sécurisation d'accès : certificat SSL serveur .....	15
2.10.2	Authentification : Login/mot de passe.....	15
2.11	Diagramme de composants .....	16
2.11.1	Application Web .....	16
2.12	Diagramme de Déploiement.....	16
2.13	Outils de Déploiement / architecture / Codage .....	17
<b>3</b>	<b><u>MODULES TECHNIQUES SPECIFIQUES</u></b>	<b>17</b>
3.1	Gestion des mails.....	17
3.1.1	Templates de mails.....	18
3.1.2	Syntaxe des informations contextuelles .....	19
3.1.3	Rendu des informations contextuelles.....	19
3.1.4	Stockage des templates .....	19
3.1.5	Gestion des erreurs .....	19
3.2	Gestion des droits applicatifs .....	20
3.3	Base de temps .....	20
3.4	Traces applicatives .....	20
3.4.1	Logs techniques.....	20
3.4.2	Logs fonctionnels .....	21
<b>4</b>	<b><u>FORMAT DE DONNEES HARMONISE</u></b>	<b>21</b>
4.1	Format des données XML .....	21
4.1.1	Format XML harmonisé .....	21
4.1.2	Schéma XML .....	22
4.1.3	Définition du Tronc Commun .....	25

Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

4.2	Format des données CSV.....	25
<b>5</b>	<b><u>GESTION DES FORMULAIRES</u></b>	<b>26</b>
5.1	Structuration d'un modèle de dossier .....	26
5.2	Structuration d'un dossier .....	29
5.2.1	Données relatives à l'espace d'échange .....	29
5.2.2	Données historiques .....	29
5.2.3	Export au format harmonisé .....	29
5.3	Génération des PDF .....	29
5.3.1	Gestion des « clichés » .....	29
5.3.2	Transmission aux partenaires.....	29
5.4	Traitement des formulaires .....	30
5.4.1	Instanciation de formulaires depuis le modèle .....	30
5.4.2	Affichage et saisie des formulaires .....	30
5.4.3	Enregistrement des formulaires.....	30
5.4.4	Formules .....	30
<b>6</b>	<b><u>MODELISATION</u></b>	<b>31</b>
6.1	Objets métier .....	31
6.2	fr.harmonia.model.....	31
6.2.1	fr.harmonia.model.acl .....	36
6.2.2	Fr.harmonia.model.dossier .....	38
6.2.3	Fr.harmonia.model.mail .....	41
6.2.4	Fr.hamonia.model.form.....	42
6.3	Services .....	52
6.3.1	Package fr.harmonia.service .....	52
6.3.2	Package fr.harmonia.service.impl.....	55
6.3.3	Couche d'accès aux données.....	56
6.3.4	Couche Vue .....	61
6.3.5	Couche Controller .....	61
6.4	Web Services .....	62
6.4.1	Web Services.....	62
6.5	Modèle physique de données .....	64
<b>7</b>	<b><u>NORMES DE CODAGE</u></b>	<b>65</b>
7.1	Conventions de nommage et de codage JAVA/J2EE .....	65
7.2	Conventions de nommage SQL.....	65
<b>8</b>	<b><u>DIMENSIONNEMENT</u></b>	<b>65</b>
8.1	Conditions de mesure de l'étude.....	65
8.2	Utilisation de la mémoire vive .....	65
8.2.1	Démarrage des serveurs .....	65
8.2.2	Estimation de l'utilisation de la mémoire durant la déclaration.....	66
8.3	Utilisation du CPU du serveur .....	67
8.3.1	Utilisation de la bande passante.....	68
8.3.2	Statistiques sur les temps de réponse des pages de l'application.....	68
8.4	Utilisation de l'espace disque .....	68
8.5	Statistiques de la base de données .....	69
8.6	Projection sur 5 ans de l'espace disque.....	71
8.7	Session – Informations generales .....	71

Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

8.8	Stockage – Informations generales .....	71
8.8.1	Données « vivantes » .....	71
8.9	Réseau.....	72
<u>9</u>	<u>CONFIGURATION DU SERVEUR</u> .....	<u>72</u>
9.1	Organisation des répertoires .....	72
9.2	Fichiers de configuration .....	72
<u>10</u>	<u>TERMINOLOGIE - ABREVIATIONS</u> .....	<u>72</u>



Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

## 1 OBJET DU DOCUMENT

Ce document décrit les spécifications techniques du système d'information Harmonia.

L'architecture physique de chaque serveur est décrite dans l'Annexe 4 au cahier des clauses techniques particulières (CCTP), intitulée « Spécifications hébergement ».

Pour l'essentiel, le présent document décrit la situation actuelle d'Harmonia en termes techniques, qui est issue :

- du CCTP n°2013-01;
- de la phase de spécifications menée entre octobre 2006 et janvier 2007 ;
- de l'évolution du projet Harmonia entre octobre 2006 et juin 2009, et notamment des évolutions réalisées jusqu'en juin 2009,
- de l'évolution du projet Harmonia entre juillet 2009 et mai 2017, et notamment des évolutions réalisées jusqu'en mai 2017.
- de l'évolution du projet Harmonia entre Janvier 2020 et Mars 2020, et notamment des évolutions réalisées au premier trimestre 2020. Ainsi que la migration des serveurs (Windows server 2008 R2 -> Windows server 2016) et des bases de données (MSSQL Server 2008 -> MSSQL Server 2017)

Dans la suite du document, le maître d'œuvre désigne la société ayant réalisé l'application Harmonia et ses évolutions entre septembre 2006 et Mars 2020.

## 2 ARCHITECTURE APPLICATIVE

### 2.1 DEPLOIEMENT DES SERVEURS

5 serveurs de production sont déployés dans le cadre du projet :

- S1 = serveur Harmonia de la Fédération des COOP (coopératives d'HLM),
- S2 = serveur Harmonia de la Fédération des ESH,
- S3 = serveur Harmonia de la Fédération des Offices (ou OPH),
- S4 = serveur Harmonia de la Fédération des EPL (ex – Fédération des SEM),
- S5 = serveur Harmonia du ministère en charge du logement (appelé dans le cadre d'Harmonia : « Ministère » ou « DHUP » ou encore « DGUHC », ancien nom de la DHUP) (\*).

(\*) Voir le glossaire au point 2.1 du CCTP.

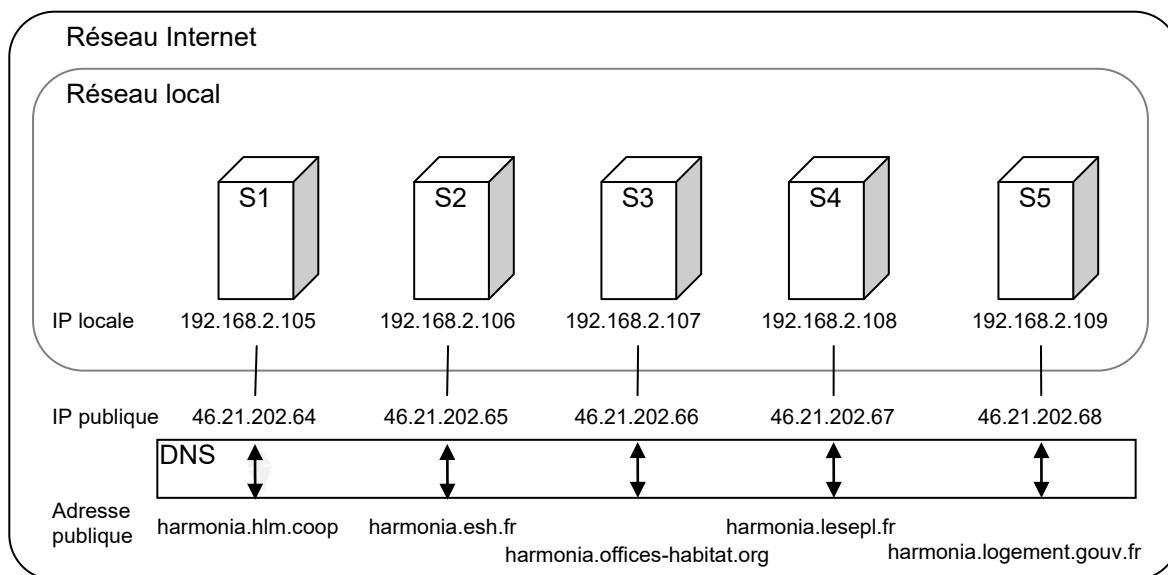
Depuis la migration du 9 octobre 2013, la solution retenue pour les serveurs est celle des machines virtuelles « Cloud Privé VMWare ». L'architecture des serveurs étant virtualisée, il est possible de modifier facilement les caractéristiques physiques allouées aux machines.

Cette solution comprend un ensemble de processeurs, mémoire vive et espace disque à répartir entre les serveurs. Les caractéristiques de chaque machines peuvent ainsi être adaptées en fonction des besoins des applications grâce à un simple paramétrage. Les 5 serveurs sont configurés au sein d'un même réseau local, facilitant ainsi leurs échanges.

### 2.2 ARCHITECTURE RESEAU

Les 5 serveurs sont installés au sein d'un même réseau local. Chacune des machines est accessible individuellement grâce à son IP publique.

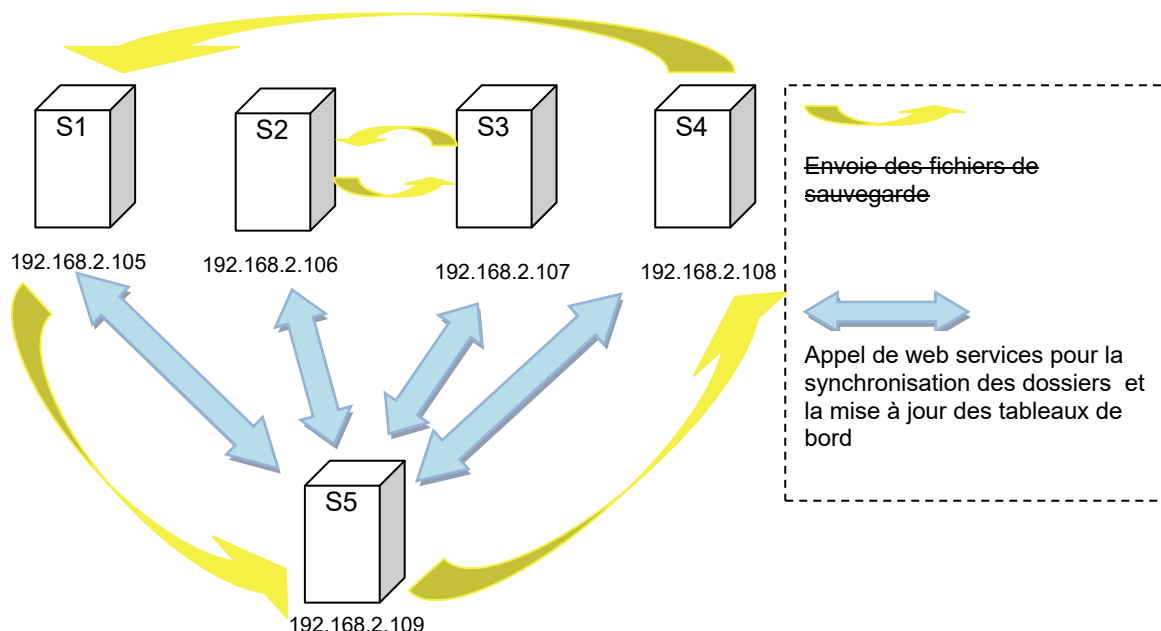




## 2.3 COMMUNICATION ENTRE LES SERVEURS

Les serveurs communiquent à l'intérieur du réseau local de deux façons :

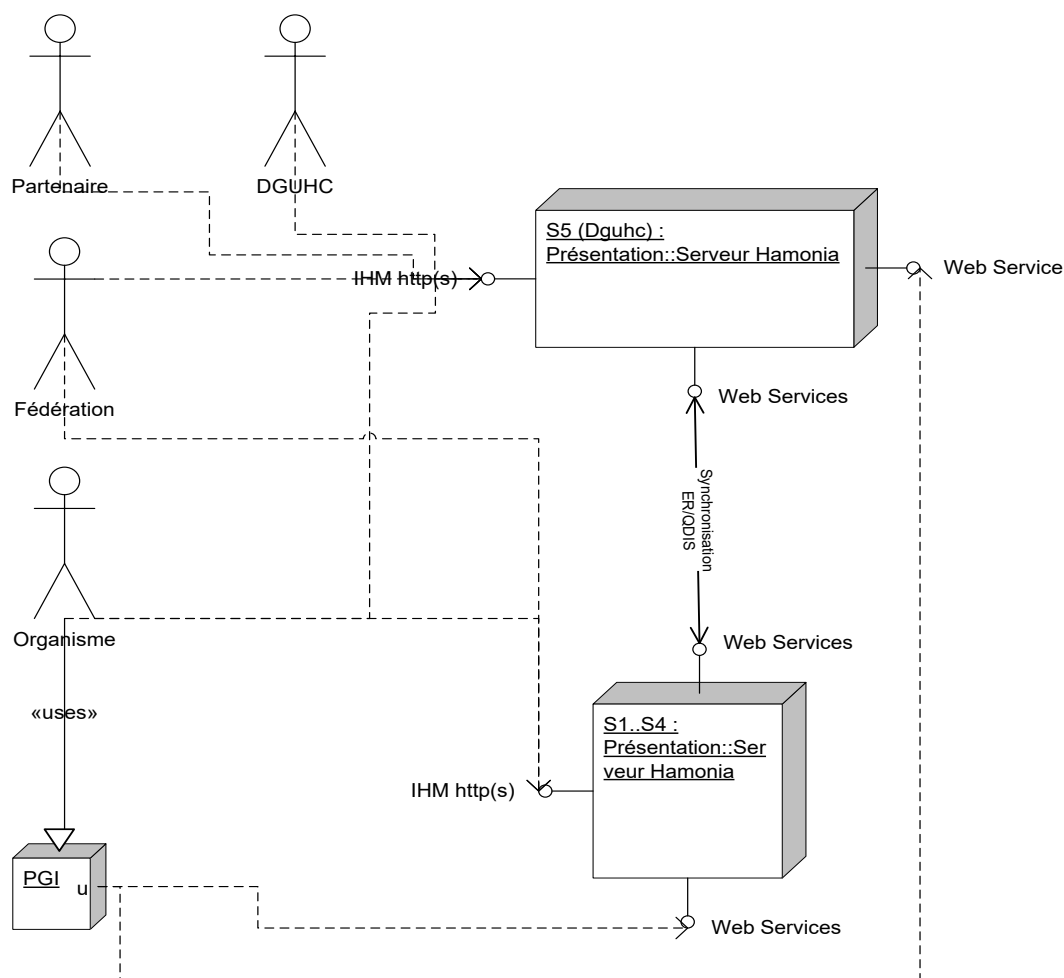
- Par l'intermédiaire de web services pour la synchronisation des dossiers de type ER et QDIS et la mise à jour des tableaux de bord entre la plateforme du ministère et celle des fédérations.
- ~~Par l'intermédiaire d'échanges de fichiers en réseaux pour la duplication des fichiers de sauvegarde de la base de données.~~



L'architecture générale des Web services d'Harmonia est résumée dans le schéma ci-après et les commentaires qui le suivent.



Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7



Les serveurs S1 à S4 communiquent avec le serveur S5 par le biais de Web Services en raison notamment de la fonction de synchronisation.

La description des ces Web Services est donnée plus loin dans ce document.

Les serveurs S1 à S5 offrent aux internautes une interface graphique d'utilisation.

**Remarque : certains points du schéma ci-dessus sont théoriques ; les Web Services reliant les PGI aux serveurs n'ont pas été mis en oeuvre.**

Explication : il était prévu en 2006 que les serveurs S1 à S5 offrent aux internautes des Web Services d'alimentation directe des données depuis leur PGI (progiciel de gestion immobilière). La mise en oeuvre de cette fonctionnalité n'a pas été réalisée. Elle a été jugée non prioritaire, au bénéfice d'autres fonctionnalités mises en oeuvre bien que non prévues initialement.

Pour des raisons de sécurité, il a été envisagé, lors des spécifications initiales en 2006, qu'un lien VPN puisse être établi entre certains partenaires (Ministère, fédérations) et leur serveur, pourvu que les matériels respectifs de l'hébergeur et du partenaire soient compatibles.

Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

Cette fonctionnalité n'a pas été en pratique mise en œuvre à la date de rédaction du présent document.

## 2.4 POSTE CLIENT

Pour mémoire, dans le CCTP (point 3.1.2 dans le cadre de la maintenance corrective et adaptative), il est indiqué : « Le titulaire s'assure en permanence que les différentes versions utilisées correspondent à l'une des deux ou trois dernières versions stables et utilisées en production de ces éléments.

Aucune version bêta ou équivalente n'est utilisée. ».

Ces principes s'appliquent à l'ensemble des sujets évoqués dans le présent point 2.2.

### 2.4.1 Système d'exploitation (OS = Operating System) et navigateurs supportés

Le titulaire s'assure qu'Harmonia continue d'être compatible avec les systèmes suivants, qui ont été testés avec succès jusqu'ici dans le cadre du projet :

- Windows 2000 SP4
- Windows XP SP1, SP2, SP3
- Windows Vista SP1, SP2
- Windows 7
- Windows 10
- Internet Explorer (IE) 11 ou plus
- Firefox 42.0 ou plus.
- Chrome 57 ou plus

~~Pas d'utilisation de cookies résidents.~~

### 2.4.2 Configuration écran

Les écrans favorisent des résolutions de 1024x768 (standard des écrans 17 pouces) mais fonctionnent également sur des résolutions 800x600 (écrans 14 pouces) ; avec utilisation d'ascenseurs. En mode 19 pouces ou plus, l'application est également utilisable.

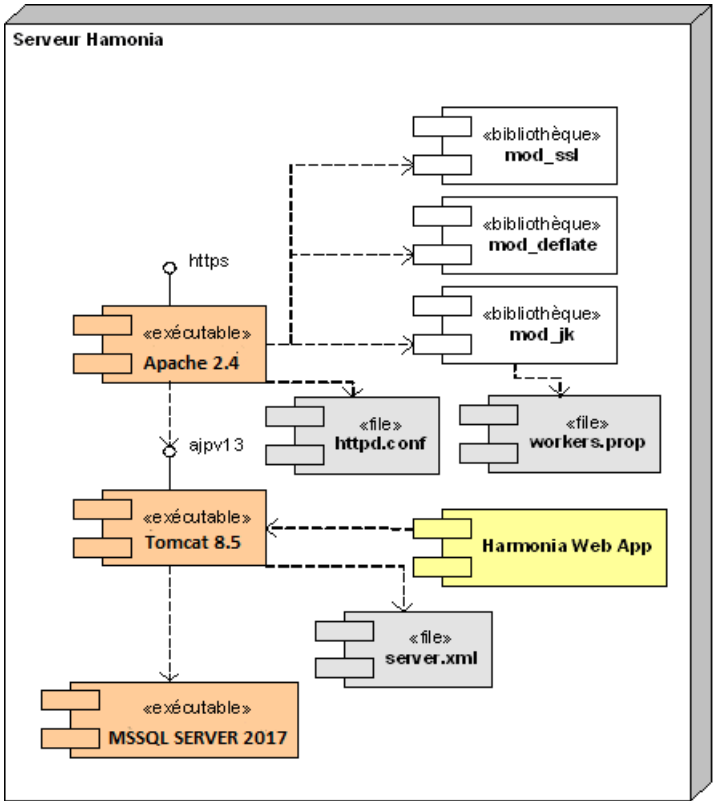
### 2.4.3 Ouverture sur la mobilité

L'application n'a pas été testée jusqu'à présent sur un système mobile et elle n'a pas vocation à l'être pendant la durée du marché.

## 2.5 MACRO-ARCHITECTURE DES COMPOSANTS D'UN SERVEUR

Chaque serveur Harmonia héberge les composants suivants :

- Un serveur web Apache Httpd 2.4.x
- Un serveur applicatif J2EE Jakarta Tomcat 8.5.x
- Un serveur de base de données Microsoft SQL Serveur 2017.



## 2.6 CARACTERISTIQUE DES SERVEURS

Les spécifications techniques actuelles des 5 serveurs sont les suivantes :

Nom serveur	OS	CPU	RAM	DD	IP Locale
SRV4951	Windows Server 2016 Standard	4 vCPU (virtual CPU)	4GO	102 GO	192.168.2.105
SRV4952	Windows Server 2016 Standard	4 vCPU (virtual CPU)	4GO	135 GO	192.168.2.106
SRV4953	Windows Server 2016 Standard	4 vCPU (virtual CPU)	4GO	125 GO	192.168.2.107
SRV4954	Windows Server 2016 Standard	4 vCPU (virtual CPU)	4GO	100 GO	192.168.2.108
SRV4955	Windows Server 2016 Standard	4 vCPU (virtual CPU)	16GO	425 GO	192.168.2.109

Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

## 2.7 SERVEUR WEB

Le serveur Web est le serveur Open Source HTTPD 2.4 d'Apache.

Il est configuré pour servir les contenus statiques de l'application :

- Images
- Feuilles de style (sauf celles gérées dynamiquement pour les formulaires)
- Javascripts
- Applets java

Les modules suivants sont activés ou ajoutés pour répondre à des fonctions particulières :

- Mod\_jk : le connecteur entre Apache et Tomcat.
- Mod\_deflate : module de compression des flux. Les formulaires étant potentiellement assez volumineux (une page pèse entre 50 et 100ko) , l'utilisation de ce module peut réduire leur taille (le facteur courant est 7) entre 7 et 15ko. (\*)
- Mod\_ssl : module d'implémentation du protocole https de sécurisation des échanges. (\*\*)

(\*) Certains utilisateurs n'ont pas des réseaux très rapides. Il est donc nécessaire que la taille soit la plus réduite possible.

(\*\*) L'ensemble de l'application est accessible intégralement et uniquement en https.

## 2.8 SERVEUR DE BASE DE DONNEES

Le serveur de base de données utilisé est Microsoft SQL Serveur 2017 Standard (voir aussi point 2.3 ci-dessus).

Il est utilisé pour stocker :

- La configuration de chaque application :
  - o Paramétrage propre au serveur (Web services,...)
  - o Paramétrages des droits utilisateurs généraux et des étapes
- Les tables de référence
  - o Organismes
  - o Partenaires
  - o Utilisateurs
- Les données applicatives
  - o Modèles de formulaires pendant 5 ans (4 types de QDIS,3 types d'ER et 6 autres types d'enquetes pour les fédérations)
  - o Formulaires de saisie par campagne et par organisme
- Les données saisies ou calculées comprises dans les formulaires

Afin de faciliter l'exploitation des serveurs (sauvegardes), d'augmenter la confidentialité des données et de gérer efficacement les transactions, les modèles xml des formulaires sont stockés sous forme de « blobs » dans la base de données.

Le cas échéant, pour accélérer d'éventuels traitements ensemblistes lourds en base de données, les procédures stockées en langage TransacSQL sont privilégiées. Par exemple : sauvegarde des dossiers Pour des raisons de performances, les contenus des formulaires (données saisies ou calculées) ne sont pas stockés sous forme XML mais en tant que champs relationnels dans une table associée.

Afin de permettre une récupération de l'application dans le cas d'une défaillance d'une base de données, une sauvegarde des bases de données est effectuée chaque nuit.

## 2.9 SERVEUR APPLICATIF

Le serveur applicatif utilisé est Jakarta Tomcat 8.5. Il nécessite l'installation du JDK 1.8.

Le serveur applicatif met à disposition de l'application web l'environnement nécessaire à son exécution.



Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

L'application web utilise elle un certain nombre de composants Open Source qui lui sont intégrés sous forme de bibliothèques.

Les principes de la maintenance corrective et adaptative (cf point 3.1.2 du CCTP) sont respectés.

Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

### 2.9.1 Composants techniques

Les composants actuellement utilisés sont les suivants :

- Java JDK 1.8
- Tomcat 8.5.x
- Spring 2.5.x (Framework MVC)
- Spring Security : authentification et gestion des autorisations
- Apache Log4j : log applicatifs (hors Spring qui possède sa propre infrastructure)
- Apache FOP 0.92 : génération des PDF
- Hibernate 3.3.x
- Microsoft SQLServer JDBC driver : librairie de communication entre J2EE et la base MSSQL Server
- Spring WS : implémentation des Web Services SOAP
- JavaCC 4.0 : pour le parser des formules

### 2.9.2 Normes implémentées

- JSP 2.0
- Servlets 2.4
- SOAP 1.2

### 2.9.3 Spring 2.5

Spring est le framework MVC de l'application Harmonia.

Ses principes d'inversion de contrôle (IoC) et de Programmation Orientée Aspect (AOP) en font un framework souple, facilitant l'extension et l'ajout de fonctionnalités.

Il est largement utilisé dans la communauté Open Source.

#### 2.9.3.1 JSP

Les JSP sont utilisées pour implémenter les vues de l'application. Les librairies de tags suivantes sont utilisées :

- JSTL : les taglibs standard de Sun
- Spring Taglib
- Spring Security Taglib : les tags relatifs à la sécurité

#### 2.9.3.2 Spring Security (acegi security)

Spring Security est le composant de sécurisation packagé avec Spring. Il gère les deux aspects de la sécurité :

- L'authentification : la gestion du login afin d'associer à une nouvelle session utilisateur une identité. Ce framework découple le mode d'authentification de telle manière qu'il est simple de le remplacer par un autre mode, en cas d'évolution.
  - o Le mode utilisé est la gestion d'un formulaire de type login/mot de passe stocké en base, mais Spring Security permet également les modes d'authentification suivants :
    - o HTTP X.509 client certificate exchange (basé sur le standard IEFT RFC)
    - o Annuaire LDAP
    - o La propagation transparente du contexte d'authentification pour les appels distants de type RMI et HttpInvoker (le protocole d'appel distant de Spring)
    - o Authentification anonyme par attribution automatique d'un contexte de connexion prédéfini



Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

- Intégration possible avec le Container Manager Authentication de JBoss, Jetty, Resin et Tomcat
- Voire même, intégration possible avec un système d'authentification propriétaire
- L'autorisation : la gestion des droits associés à l'utilisateur, ceci à quatre niveaux.
  - Dans la vue, par la mise à disposition d'une taglib simple permettant de masquer ou dévoiler des informations et des contrôles aux utilisateurs selon les droits qui leur ont été assignés
  - Sur la page, en contrôlant l'accès à une chaque page, selon les mêmes principes
  - Sur chaque méthode de chaque service, selon les mêmes principes
  - Voire même, gestion d'ACL (Access Control List) au niveau de chaque objet que l'on souhaite protéger.

#### 2.9.4 Hibernate 3.3

Hibernate est le moteur de mapping Objet-Relationnel utilisé dans l'application.

Il permet de faire abstraction de la base de données réellement utilisée, et découple les données telles qu'elles sont utilisées dans l'application de la manière dont elles sont stockées en base de données.

Il permet également de s'affranchir de l'écriture des requêtes SQL « simples » (par exemple, les requêtes unitaires d'insertion, modification, suppression) tout en laissant la possibilité d'écrire des requêtes SQL directement.

La connectivité à la base de données est assurée par le driver JDBC fourni par Microsoft avec SQLServer.

## 2.10 SECURITE

### 2.10.1 Sécurisation d'accès : certificat SSL serveur

L'ensemble de l'application est sécurisé par l'utilisation du protocole https.

Ce mode de sécurisation repose sur le cryptage des données échangées : un tiers ne peut pas comprendre les informations échangées s'il « écoute » les échanges.

Le protocole https a notamment nécessité l'installation sur chaque serveur Harmonia d'un « certificat SSL ».

Ces 5 certificats SSL ont été produits par le prestataire Thawte à la suite de l'envoi d'une demande du maître d'œuvre (fichier CSR).

Ces certificats ont été vérifiés par le maître d'œuvre, lors de leur remise, via une fonction édiée de Thawte.

Depuis l'année 2020 le certificat SSL utilisé sur le serveur du ministère est produit par le prestataire Certigna.

### 2.10.2 Authentification : Login/mot de passe

L'authentification est réalisée par l'utilisation d'un couple login/mot de passe.

Le composant Spring Security implémente cette fonctionnalité.

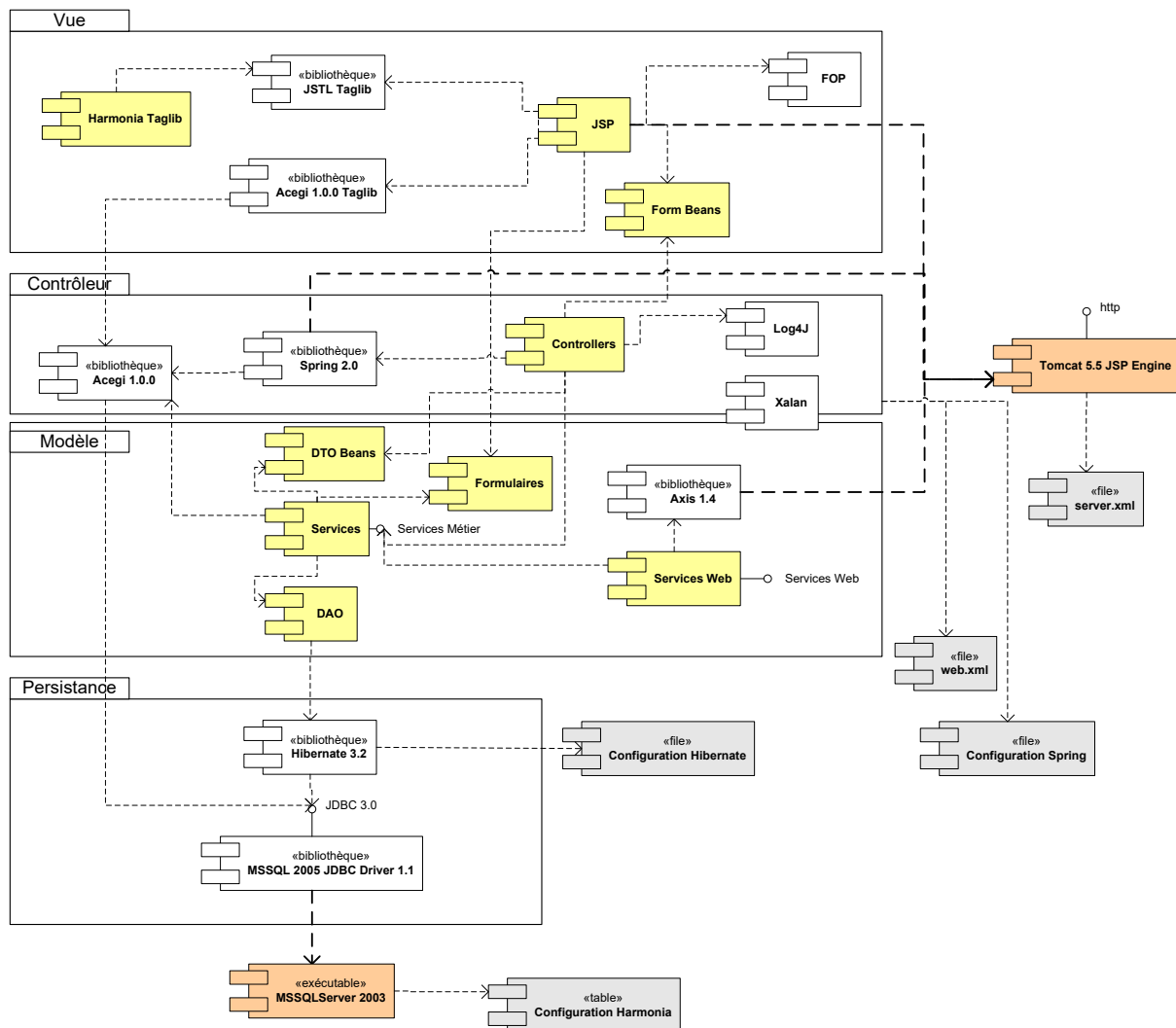
Le mot de passe est stocké dans la base, crypté par l'utilisation d'un algorithme MD5, qui rend impossible l'exploitation du mot de passe stocké, à moins de le connaître.

Seuls les administrateurs ont la possibilité de modifier le mot de passe d'un utilisateur, ils sont donc désignés avec prudence.

Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

## 2.11 DIAGRAMME DE COMPOSANTS

### 2.11.1 Application Web

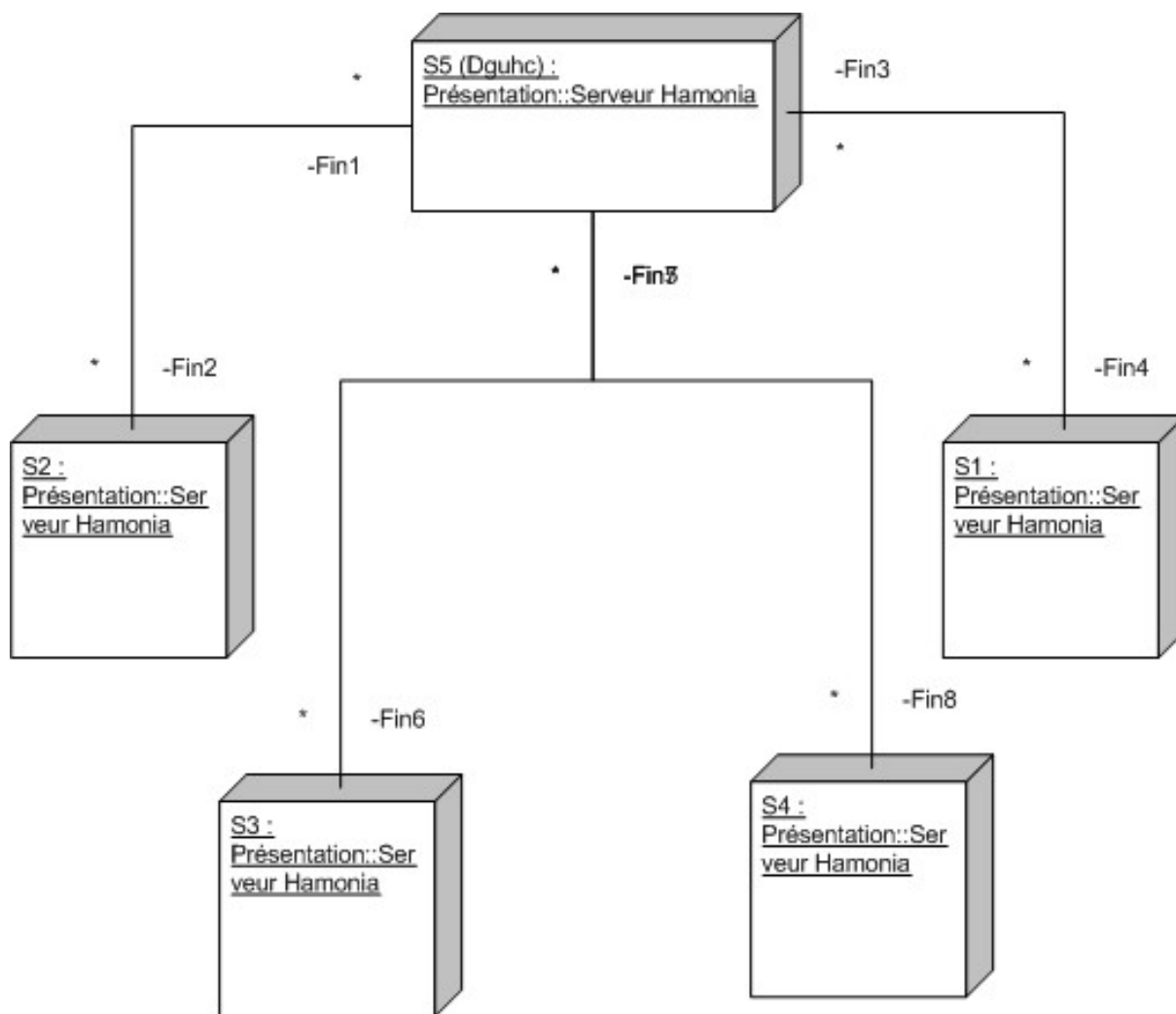


## 2.12 DIAGRAMME DE DEPLOIEMENT

L'application Harmonia est déployée sur les 5 serveurs, mais paramétrée de manière différente.



Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7



**Rappel** : conformément aux indications du CCTP (voir notamment le glossaire au point 2.1 du CCTP), la DGUHC est devenue en 2008 la DHUP. Dans le cadre d'Harmonia, les trois termes DGUHC, DHUP et Ministère sont à considérer comme étant équivalents.

## 2.13 OUTILS DE DEPLOIEMENT / ARCHITECTURE / CODAGE

Les outils suivants ont été utilisés au cours du développement :

- Ant : automatisation des scripts de compilation, test, déploiement
- Maven 2: automatisation des scripts de compilation, test, déploiement et automatisation des rapports d'avancement
- JUnit : automatisation des tests unitaires

## 3 MODULES TECHNIQUES SPECIFIQUES

Un certain nombre de modules ont été développés pour gérer spécifiquement certains aspects de l'application Harmonia.

### 3.1 GESTION DES MAILS

Ce module utilise l'API JavaMail pour procéder à l'envoi de mails.

Son fonctionnement est celui d'une file d'attente : dès que l'application envoie un message, il est ajouté dans la liste (qui est persistée en base) et est envoyé :

Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

- Immédiatement (cas des mails associés à l'espace d'échange dès que l'utilisateur décide de publier ses modifications)
- En différé. Il vient alors grossir une pile de mails destinés à être envoyés une fois par jour. Chaque utilisateur, en dehors du cas particulier du « contact direction », ne reçoit alors qu'un mail par jour.

### 3.1.1 Templates de mails

Un mail est composé à partir d'un « template » qui contient :

- Un titre de catégorie : une ligne qui indique le type du mail :  
« *Notification de passage en étape 3 d'un ER* »
- Un message d'introduction (header) : un message qui explique la catégorie :  
« *Voici la liste des ER passés en étape 3 aujourd'hui.* »
- Une ligne d'information contextuelle : cette ligne est répétée pour chaque dossier. Un certain nombre de mots-clés permettent d'y insérer des données relatives au dossier auquel elle se rapporte :  
« *OPHML Bobigny (code : 93067) – Année 2005 – ER passé en étape 3 le 12/05/2007 14 :30* »
- Un message de conclusion (footer) : un message qui ferme la catégorie :  
« *---°°°---* »

Tous les mails utilisent par ailleurs un template commun :

- Un titre :  
« *[Harmonia] Message de l'application Harmonia* »
- Un header  
« *Bonjour, ceci est un message de l'application Harmonia.* »
- Un footer  
« *L'application Harmonia.*  
  
*Ce message a été généré automatiquement, merci de ne pas répondre à cette adresse.  
Si vous pensez qu'il ne vous était pas destiné, merci d'en avvertir l'administrateur du site à l'adresse suivante : ..... »*

La traduction en schéma XML du fichier de paramétrage est la suivante :

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xs:element name="MAIL">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="COMMUN"/>
        <xs:element ref="MESSAGES"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="COMMUN">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="OBJET"/>
        <xs:element ref="HEADER"/>
        <xs:element ref="FOOTER"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="OBJET">
    <xs:complexType/>
  </xs:element>
  <xs:element name="MESSAGES">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="MESSAGE"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

```



Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

```

<xs:element name="MESSAGE">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="TITRE"/>
      <xs:element ref="HEADER"/>
      <xs:element ref="LIBELLE"/>
      <xs:element ref="FOOTER"/>
    </xs:sequence>
    <xs:attribute name="code" use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="TITRE">
  <xs:complexType/>
</xs:element>
<xs:element name="LIBELLE">
  <xs:complexType mixed="true">
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="DOSSIER"/>
      <xs:element ref="DOSSIER_URL"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="DOSSIER"/>
<xs:element name="DOSSIER_URL"/>
<xs:element name="HEADER">
  <xs:complexType/>
</xs:element>
<xs:element name="FOOTER">
  <xs:complexType/>
</xs:element>
</xs:schema>

```

### 3.1.2 Syntaxe des informations contextuelles

La ligne d'information contextuelle est en rapport avec un événement survenu sur un dossier :

- Notification de passage d'étape
- Transmission d'un ER
- Mise à disposition d'un récépissé...

Elle permet l'accès aux propriétés du dossier à travers une syntaxe java reconnue par le générateur de mail, qui donne accès à tous les attributs publics ou les getters publics d'un objet, en partant du dossier.

La phrase

« OPHML Bobigny (code : 93067) – Année 2005 - étape 3 le 12/05/2007 14 :30 »

aurait pu être paramétrée ainsi :

<DOSSIER>organisme.nom</DOSSIER>	(code :	<DOSSIER>codeFede</DOSSIER>)	–	Année
<DOSSIER>campagne.millesime</DOSSIER>	-	étape	<DOSSIER>etat.etapeCourante.numero</DOSSIER>	le
<DOSSIER>etat.etapeCourante.modifInfo.date</DOSSIER>				

Un tag particulier, <DOSSIER\_URL/>, permet de poser un lien dans le mail directement vers le dossier.

### 3.1.3 Rendu des informations contextuelles

Techniquement, le générateur utilise les fonctions d'introspection de Java pour parcourir l'objet et en retourner les attributs.

Les attributs de type numériques, dates et booléens sont affichés grâce à un formatage particulier (le même que celui utilisé dans les états).

Les autres données seront rendues par un toString().

En cas d'exception, une chaîne de caractères « xxx » remplace la valeur attendue et une erreur est loggée.

### 3.1.4 Stockage des templates

Les templates sont stockés sous forme de fichiers XML sur le serveur.

### 3.1.5 Gestion des erreurs

Le générateur de mail n'empêche jamais une transaction d'aboutir. En conséquence, la génération des mails est appelée après les transactions métiers.



Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

Deux types d'erreur peuvent se produire :

1. Pendant la génération du mail (c'est-à-dire, au moment où a lieu l'action). Cette erreur empêche la génération du mail, et l'utilisateur est averti que sa transaction est effectuée mais que la notification par mail n'a pas pu être réalisée. Une erreur est loggée.
2. Au moment de l'envoi du mail : l'envoi étant asynchrone, l'utilisateur ne reçoit pas de notification de l'erreur, mais l'erreur est loggée.

### 3.2 GESTION DES DROITS APPLICATIFS

La gestion de la vérification des autorisations associées à chaque utilisateur sur chaque objet est déléguée au composant Spring Security.

Il utilise les services définis par l'interface **ISecurityManager** pour attribuer ses droits à un utilisateur, en fonction du contexte :

- Organisme
- Etape du dossier en cours
- Présence ou non d'un verrou sur le dossier
- Type d'utilisateur
- ...

### 3.3 BASE DE TEMPS

Une base de temps commune doit être partagée entre les serveurs afin que les synchronisations entre les dossiers puissent être détectées sans risque d'erreur.

Les serveurs sont donc connectés à un serveur de temps commun.

Le service de temps Windows du serveur doit être configuré sur le service de temps du maître d'opérations du contrôleur de domaine principal situé à la racine de la forêt. Ce serveur est lui-même configuré soit sur son horloge interne, soit sur un service de temps externe (mais cela n'apporte pas de plus-value).

### 3.4 TRACES APPLICATIVES

Les seules traces applicatives sont des traces purement techniques qui ne contreviennent pas au principe de confidentialité des données.

#### 3.4.1 Logs techniques

Les logs techniques sont générés par l'application à plusieurs niveaux :

##### 3.4.1.1 Apache

- Log d'erreur (error log) : le log des erreurs d'exécution d'Apache. Toute nouvelle valeur dans ce log indique une erreur du serveur et doit donc faire l'objet d'une alerte et d'une analyse approfondie.
- Log d'accès (access log) : le log à la base des statistiques d'accès au site. On utilisera le format « Combined Log Format » qui permet de tracer des informations complémentaires sur les Referer et les navigateurs utilisés.
- Log mod\_jk (mod\_jk.log).
- Log d'accès SSL (request log).
- Log d'erreurs SSL (error log).

##### 1.1.1.1 Tomcat

- Log CATALINA: log principal du serveur.
- Log connecteur AJP: permet le log des requêtes reçues du serveur applicatif.



Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

- Log Harmonia : log spécifique de la webapp Harmonia géré par Spring. C'est ici que les erreurs de l'application elle-même sont consignées.

#### 3.4.2 Logs fonctionnels

Les logs fonctionnels peuvent être ajoutés dans un fichier de log supplémentaire par activation du paramètre « harmonia\_log\_level » de l'application Harmonia grâce à la console d'administration Tomcat.

Les niveaux de log possibles sont :

- ERROR : seules les erreurs sont consignées dans le fichier de log.
- WARNING : les alertes (définies comme telles lors du développement) sont également consignées.
- DEBUG : tous les événements sont consignés avec leurs paramètres.
- VERBOSE : tous les événements sont consignés avec leurs paramètres ainsi que les réponses aux appels.

Par défaut le système fonctionne en mode minimal de log.

## 4 FORMAT DE DONNEES HARMONISE

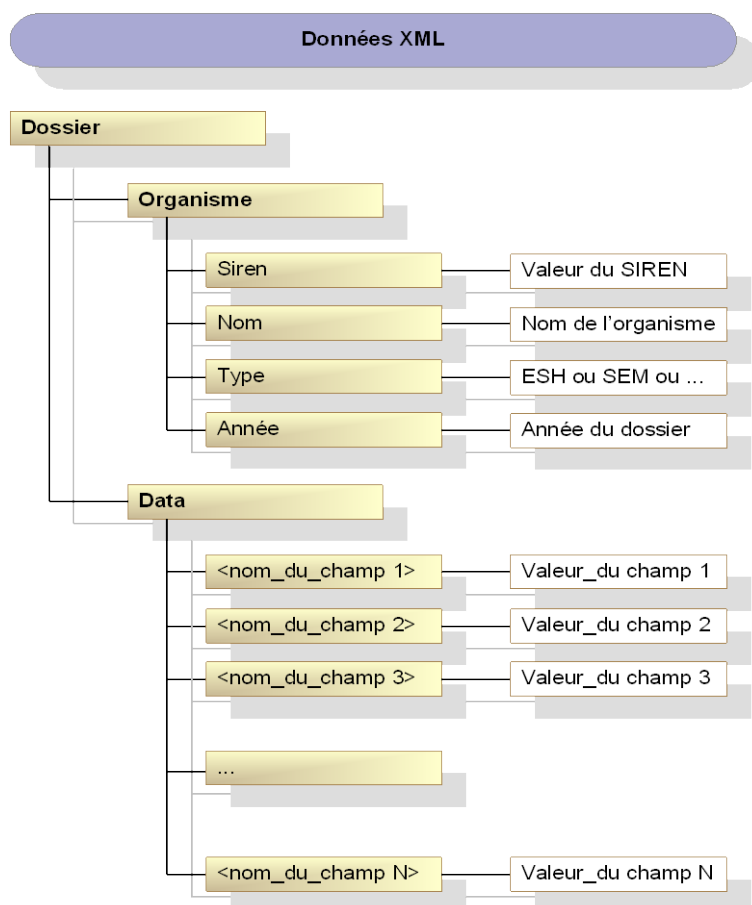
### 4.1 FORMAT DES DONNEES XML

#### 4.1.1 Format XML harmonisé

Les données importées ou saisies sont véhiculées à travers Harmonia par l'intermédiaire de fichiers XML.

Le graphique suivant présente la logique hiérarchique de l'organisation des données au sein d'un document XML.

Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7



Des attributs de gestion viennent enrichir les éléments XML. Voici une description de ces attributs :

- Sur l'élément <Dossier>, des informations faisant référence aux espaces de nommage (namespace, voir détail dans les schémas)
- Sur les éléments <nom\_du\_champ>,
  - o un attribut *dateHeure* permet de mémoriser l'instant exact de date de dernière modification. Cette information a son importance lors des synchronisations entre ER et QDIS pour permettre de détecter facilement les nouvelles valeurs de champs.
- Sur les éléments <nom\_du\_champ> issus de tableaux variables (nom de champ suffixé par « \_dyn »)
  - o un attribut *item* supplémentaire est défini pour les éléments <nom\_de\_champ> pour identifier le numéro de ligne auquel appartient le champ.

L'encodage des fichiers XML est iso-8859-15.

#### 4.1.2 Schéma XML

Les fichiers XML respectent une grammaire. Cette grammaire se traduit par des règles qui garantissent une cohérence entre tous les fichiers XML. L'ensemble des règles est traduit dans un fichier appelé schéma XML.

Un schéma XML peut être plus ou moins restrictif en fonction du niveau de contrôle, et donc de validité, que l'on souhaite atteindre. Dans la mesure où les schémas des différents QDIS, mais aussi des ER (bien que théoriquement dans une moindre mesure), évoluent d'une année à l'autre, le schéma défini dans le cadre du projet Harmonia possède les particularités suivantes :

- o Nommage des éléments XML conformes aux champs Excel (pas de banalisation du nom). Ceci permet de s'assurer que d'une année sur l'autre, un ancien schéma ne peut être utilisé avec de nouvelles données (limitation des risques d'erreur).

Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

- Typage des valeurs de champs en fonction des formats de cellule excel. Les types de champs disponibles à ce jour sont :
  - Chaîne de caractère (xs:string, ex : « total bilan »)
  - Valeur décimale (xs:decimal, ex : 999.50, -210.889, 14)
  - Valeur entière (xs:integer, ex : 999, -12, 0)
  - Valeur booléenne (xs:boolean, ex : true, 1, false, 0)
  - Date (xs:date, AAAA-MM-JJ, ex : 2002-09-24)
  - Date + heure (xs:dateTime, AAAA-MM-JJThh:mm:ss, ex : 2002-09-24T09:12:00)
- Typage des valeurs d'attributs conformément aux types précédents
- Utilisation de namespace. Afin de distinguer correctement des données issues du tronc commun, et les autres, les noms des éléments sont préfixés par un namespace (préfixé par « qdis : » ou « er : ») au sein des fichiers XML des QDIS. Dans un ER, seul le namespace du Ministère est utilisé. Les namespaces suivant sont utilisés dans le cadre d' Harmonia :

Fédération / Ministère	Namespace associé
FNESH	http://harmonia.esh.fr/qdis/<année de modification>
Fédération des COOP	http://harmonia.hlm.coop/qdis/<année de modification>
OPH	http://harmonia.offices-habitat.org/qdis/<année de modification>
Fédération des EPL	http://harmonia.lesepl.fr/qdis/<année de modification>
MINISTERE-SA	http://harmonia.logement.gouv.fr/dguhc/sa/er/<année de modification>
MINISTERE-OPH	http://harmonia.logement.gouv.fr/dguhc/office/er/<année de modification>
MINISTERE-SEM	http://harmonia.logement.gouv.fr/dguhc/sem/er/<année de modification>

#### 4.1.2.1 Schéma d'un ER

Voici le formalisme du schéma d'un ER associé au fichier XML décrit précédemment. En **surbrillance grise** sont décrites les parties variables du schéma.

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="unqualified"
  targetNamespace="namespace sa, office ou sem"
  xmlns:er="namespace sa, office ou sem">
  <xs:element name="Dossier">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="er:Organisme"/>
        <xs:element ref="er:Data"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="Organisme">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="er:Siren"/>
        <xs:element ref="er:Nom"/>
        <xs:element ref="er:Type"/>
        <xs:element ref="er:Annee"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="Siren" type="xs:string" />
  <xs:element name="Nom" type="xs:string"/>
  <xs:element name="Type" type="xs:string"/>
  <xs:element name="Annee" type="xs:string"/>

  <xs:element name="Data">
    <xs:complexType>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element ref="er:nom_champ_er1"/>
        <xs:element ref="er:nom_champ_er2_dyn"/>
        ...
      </xs:choice>
    </xs:complexType>
  </xs:element>
```

# Document de spécifications

Capgemini

Spécifications techniques détaillées

SPF\_002/1.7

```
<xs:element name="nom_champ_er1">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:decimal">
        <xs:attribute name="dateHeure" type="xs:dateTime" form="unqualified"/>
        <xs:attribute name="forcable" type="xs:boolean" default="false"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="nom_champ_er2_dyn">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:decimal">
        <xs:attribute name="dateHeure" type="xs:dateTime" form="unqualified"/>
        <xs:attribute name="item" type="xs:integer" use="required" form="unqualified"/>
        <xs:attribute name="forcable" type="xs:boolean" default="false"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
...
</xs:schema>
```

Voici un exemple de fichier XML associé :

```
<?xml version="1.0" encoding="iso-8859-15"?>
<er:Dossier xmlns:er="http://www2.logement.gouv.fr/dguhc/sa/er/2007"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www2.logement.gouv.fr/dguhc/sa/er/2007 nom-schema-er.xsd">
  <er:Organisme>
    <er:Siren>122 456 789</er:Siren>
    <er:Nom>ESH Lyon</er:Nom>
    <er>Type>ESH</er>Type>
    <er:Annee>2007</er:Annee>
  </er:Organisme>
  <er>Data>
    <er:_202 dateHeure="2007-02-06T18:06:00">2323.34</er:_202>
    <er:_204_dyn item="1">12522.23</er:_204_dyn>
    <er:_204_dyn item="2">125.50</er:_204_dyn>
    <er:_204_dyn item="3">169.78</er:_204_dyn>
  </er>Data>
</er:Dossier>
```

## 4.1.2.2 Schéma d'un QDIS

Voici le formalisme du schéma d'un QDIS associé au fichier XML décrit précédemment. En **surbrillance grise** sont décrites les parties variables du schéma.

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
  targetNamespace="namespace fédération"
  xmlns:qdis="namespace fédération"
  xmlns:er="namespace sa, office ou sem">
  <xs:import namespace="namespace sa, office ou sem" schemaLocation="schema-ER.xsd"/>
  <xs:element name="Dossier">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="qdis:Organisme"/>
        <xs:element ref="qdis>Data"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="Organisme">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="qdis:Siren"/>
        <xs:element ref="qdis:Nom"/>
        <xs:element ref="qdis>Type"/>
        <xs:element ref="qdis:Annee"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
```





# Document de spécifications

Capgemini

Spécifications techniques détaillées

SPF\_002/1.7

```
<xs:element name="Siren" type="xs:string" />
<xs:element name="Nom" type="xs:string"/>
<xs:element name="Type" type="xs:string"/>
<xs:element name="Annee" type="xs:string"/>

<xs:element name="Data">
  <xs:complexType>
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="qdis:nom_champ_qdis"/>
      <xs:element ref="er:nom_champ_er"/>
      ...
    </xs:choice>
  </xs:complexType>
</xs:element>

<xs:element name="nom_champ_qdis">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="dateHeure" type="xs:dateTime" form="unqualified"/>
        <xs:attribute name="item" type="xs:integer" use="required" form="unqualified"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
...
</xs:schema>
```

Voici un exemple de fichier XML associé :

```
<?xml version="1.0" encoding="iso-8859-15"?>
<qdis:Dossier xmlns:qdis="http://www.offices-hlm.org/qdis/2008"
  xmlns:er="http://www2.logement.gouv.fr/dguhc/sa/er/2007"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.offices-hlm.org/qdis/2008 nom-schema-qdis.xsd">
  <qdis:Organisme>
    <qdis:Siren>877 498 778</qdis:Siren>
    <qdis:Nom>ESH Paris</qdis:Nom>
    <qdis:Type>ESH</qdis:Type>
    <qdis:Annee>2007</qdis:Annee>
  </qdis:Organisme>
  <qdis:Data>
    <qdis:_201 dateHeure="2007-02-06T18:03:10">25.00</qdis:_201>
    <er:_202 dateHeure="2007-02-06T18:03:10">100.00</er:_202>
    <er:_78_NT dateHeure="2007-02-28T23:59:59">0.00</er:_78_NT>
  </qdis:Data>
</qdis:Dossier>
```

## 4.1.3 Définition du Tronc Commun

La notion de tronc commun est portée directement par le formalisme du nom de chaque champ spécifié dans le fichier excel. Un champ de même nom entre un ER et un QDIS ET ayant l'instruction qui précise la notion de tronc commun (avec l'information tc maîtrisée par le gestionnaire) garantit que le champ appartient au Tronc Commun (TC).

Ce travail repose sur l'analyse des fédérations pour définir champ par champ lesquels appartiennent au tronc commun.

## 4.2 FORMAT DES DONNEES CSV

Cette fonction n'est pas utilisée dans le cadre des imports Harmonia. Seul le format XML est communiqué aux organismes, partenaires et éditeurs.

Conformément aux spécifications initiées en 2006, le système Harmonia est en mesure de prendre en compte dans son import un format CSV.

Toutefois cette fonction n'a jamais été utilisée par un organisme.

L'encodage des fichiers CSV est l'iso-8859-15.

Le formalisme attendu est le suivant :

```
<Nature de la donnée>,<origine>,<nom>,<valeur>,<date-heure de génération>,<item>
```



Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

Elément d'une ligne CSV	Définition
Nature de la donnée	« fixe »   « dynamique ». Nature de la donnée en fonction de son appartenance à un tableau à nombre de lignes fixe ou variable
Origine	« er »   « qdis ». Origine de la donnée en fonction de son appartenance à un ER, ou bien à un QDIS. Si la donnée fait partie du tronc commun, il faut renseigner « er »
Nom	Nom du champ. Ex : _201a
Valeur	Valeur du champ. Ex : 122.00. Les champs texte doivent être signalés avec des guillemets
Date-heure de génération	Optionnel. Date et heure de génération dans le fichier CSV (format : AAAA-MM-JJThh:mm:ss, ex : 2002-09-24T09:12:00)
Item	Obligatoire si Nature = dynamique. Correspond au numéro de ligne de la donnée. Optionnel sinon.

Exemple de fichier CSV « valide »:

```
fixe,er,_201a,0.00,2007-02-08T14:14:21,
fixe,er,_266,123.00,,
fixe,er,_266,"Ceci est un \n texte",,
dynamique,er,_266,123.00,,4
fixe,qdis,_546_NF,14551.00,2007-02-08T14:14:21,
```

## 5 GESTION DES FORMULAIRES

### 5.1 STRUCTURATION D'UN MODELE DE DOSSIER

Le package **fr.harmonia.model.form** contient les classes et les sous-packages constituant un modèle de dossier.

Un modèle de dossier est stocké en base sous forme d'un fichier XML.

Voici les informations gérées par ce format:

- Les informations relatives à l'identité du modèle :
  - Identifiant
  - Année
  - Type d'organisme
- La liste des parties de modèle
  - Nom du fichier Excel
  - Nom de la partie
  - Ordre
- La liste des onglets de modèle
  - Nom de l'onglet
  - Nom de la partie à laquelle l'onglet est rattaché
  - Ordre
  - La structure HTML de l'onglet



Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

- La liste des champs de l'onglet
  - Nom du champ
  - Nom du champ N-1 à reprendre
  - Type de données
  - Valeur par défaut
  - Mode d'affichage
  - Format
  - Null / non null
  - Formule de calcul
  - Liste des contrôles de surface
    - Type de contrôle
    - Valeur de comparaison
- L'onglet de contrôle métier
  - Liste des contrôles métiers
    - Libellé
    - Formule de contrôle
    - Niveau de gravité
    - Message si contrôle non passé
- La liste des onglets d'autocontrôle
  - Nom de l'onglet
  - Nom de la partie à laquelle l'onglet est rattaché
  - Ordre
  - La structure HTML de l'onglet
  - La liste des champs de l'onglet
    - Nom du champ
    - Type de données
    - Mode d'affichage
    - Format
    - Formule de calcul

Voici le schéma (W3C XML Schema) associé et respectant les données manipulées ci-dessus :

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xs:element name="DOSSIER">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="PARTIES"/>
        <xs:element ref="CONTROLES"/>
      </xs:sequence>
      <xs:attribute name="annee" use="required"/>
      <xs:attribute name="id" use="required"/>
      <xs:attribute name="type_org" use="required"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="PARTIES">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="PARTIE"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="PARTIE">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="NOM"/>
        <xs:element ref="NOM_EXCEL"/>
        <xs:element ref="ONGLETS"/>
      </xs:sequence>
      <xs:attribute name="ordre" use="required"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```



# Document de spécifications

Capgemini

Spécifications techniques détaillées

SPF\_002/1.7

```

</xs:complexType>
</xs:element>
<xs:element name="NOM_EXCEL">
  <xs:complexType/>
</xs:element>
<xs:element name="ONGLETS">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="ONGLET"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="ONGLET">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="NOM"/>
      <xs:element ref="STRUCTURE_HTML"/>
      <xs:element ref="CHAMPS"/>
    </xs:sequence>
    <xs:attribute name="auto_controle" use="required"/>
    <xs:attribute name="ordre" use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="STRUCTURE_HTML">
  <xs:complexType/>
</xs:element>
<xs:element name="CHAMPS">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="CHAMP"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="CHAMP">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="NOM"/>
      <xs:element ref="REPRISE"/>
      <xs:element ref="VALEUR_DEFAULT"/>
      <xs:element ref="FORMULE"/>
      <xs:element ref="CONTROLES"/>
    </xs:sequence>
    <xs:attribute name="format" use="required"/>
    <xs:attribute name="mode_affichage" use="required"/>
    <xs:attribute name="null" use="required"/>
    <xs:attribute name="type" use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="VALEUR_DEFAULT">
  <xs:complexType/>
</xs:element>
<xs:element name="NOM">
  <xs:complexType/>
</xs:element>
<xs:element name="FORMULE">
  <xs:complexType/>
</xs:element>
<xs:element name="CONTROLES">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="CONTROLE"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="CONTROLE">
  <xs:complexType>
    <xs:choice>
      <xs:element ref="VALEUR_COMPARAIISON"/>
      <xs:sequence>
        <xs:element ref="LIBELLE"/>
        <xs:element ref="FORMULE"/>
        <xs:element ref="MESSAGE_ERREUR"/>
      </xs:sequence>
    </xs:choice>
    <xs:attribute name="niveau_gravite"/>
    <xs:attribute name="type"/>
  </xs:complexType>
</xs:element>

```



Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

```

<xs:element name="VALEUR_COMPARAISSON">
  <xs:complexType/>
</xs:element>
<xs:element name="LIBELLE">
  <xs:complexType/>
</xs:element>
<xs:element name="MESSAGE_ERREUR">
  <xs:complexType/>
</xs:element>
</xs:schema>

```

## 5.2 STRUCTURATION D'UN DOSSIER

Le package **fr.harmonia.model.dossier** contient les classes et les sous-packages constituant un modèle de dossier.

Un dossier est stocké en base sous forme relationnelle. L'historisation n'est pas conservée champ par champ à chaque modification. La conservation de l'historique se fait par la création d'un fichier XML lors d'un passage d'étape.

### 5.2.1 Données relatives à l'espace d'échange

Les données de l'espace d'échange sont stockées en base de données pour être accessibles rapidement dans l'interface.

### 5.2.2 Données historiques

Les versions précédentes sont stockées dans un fichier XML pour les étapes 3 et 6.

### 5.2.3 Export au format harmonisé

Les données stockées en base sous forme relationnelle sont exportées sous forme XML zippé.

## 5.3 GENERATION DES PDF

Les versions PDF des dossiers sont générées grâce au module FOP : une XSLT permet de fusionner la définition HTML d'un onglet de formulaire et les valeurs de ses champs en un fichier XML qui est ensuite transformé en PDF portrait ou paysage.

Si nécessaire, l'onglet est généré sur plusieurs pages.

Dans tous les cas, une attention particulière est portée par le titulaire sur la qualité de lecture du PDF pour l'utilisateur final (utilisation de l'espace dans la page...)

En cas d'erreur de génération, l'utilisateur est averti et le processus de génération s'arrête.

### 5.3.1 Gestion des « clichés »

Les clichés sont des PDF qui sont automatiquement générés par l'application et stockés sous forme de fichiers dans la base de données.

### 5.3.2 Transmission aux partenaires

Les partenaires peuvent recevoir des fichiers PDF + XML en fin d'étape 3 (au moment du passage à l'étape 4) ou lorsque le dossier est en étape 6.

Sur le serveur Harmonia ministère et uniquement sur celui-ci, les partenaires peuvent également recevoir les fichiers PDF + XML des dossiers qui passent en étape « 4.8 : à importer dans BOLERO ».

La transmission se fait par simple dépôt sur un répertoire protégé par Apache (login/mot de passe) sur le serveur concerné. Les partenaires doivent se connecter à l'url du serveur pour télécharger les fichiers qui les intéressent.

Si le paramétrage ne les y autorise pas, les partenaires ne peuvent pas voir les fichiers puisque ceux-ci ne sont pas déposés. Même remarque pour la consultation dans Harmonia, la recherche sur les organismes est limitée aux organismes qui acceptent la transmission au partenaire concerné.

Lorsque l'organisme change le paramétrage pour un partenaire, les règles suivantes sont appliquées :

- « Ajout du droit d'accès » : l'archive contenant les fichiers PDF et XML de l'organisme est déposée dans les répertoires Apache du partenaire concerné

Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

- dans le répertoire « transmis » si le document est en étape 4 ou supérieure
- dans le répertoire « fiabilise » si le document est en étape 6
- dans le répertoire « bolero » si le document est en étape 4.8
- « Retrait du droit d'accès » : l'archive contenant les fichiers PDF et XML de l'organisme n'est pas supprimé des répertoires du partenaire concerné

## 5.4 TRAITEMENT DES FORMULAIRES

### 5.4.1 Instanciation de formulaires depuis le modèle

L'instanciation (ou création) de formulaire passe par un traitement de mise en mémoire des informations issues du fichier XML principal généré par la macro excel.

A partir des informations décrivant :

- Les fichiers excel (si le dossier comprend plusieurs fichiers)
- les onglets
- les champs
- les formules
- les contrôles métier de validation
- les règles de calcul des écarts
- les règles de reprise des données de l'année précédente
- les informations de paramétrage des regroupements d'onglets pour l'impression

Un objet est instancié et mis en mémoire pour répondre aux demandes d'affichage. Cette instanciation est réalisée une fois au démarrage du serveur.

### 5.4.2 Affichage et saisie des formulaires

L'affichage d'un formulaire est en fait un affichage positionné sur un onglet précis. C'est un traitement basé sur la manipulation de deux types d'information :

- L'instanciation précédemment évoquée
- L'ensemble des champs « valués » des données issues de la base de données pour être chargé dans un objet dossier (hash map stockant les valeurs) de travail en session java.

En fonction du bon onglet de formulaire (déterminé pour une fédération ou pour le Ministère, pour une année et pour le type de formulaire ), les données sont intégrées à la place des variables sous forme d'input HTML et la page HTML est dynamiquement construite sur le serveur pour être retournée au client.

L'utilisateur peut alors saisir les champs dans les inputs prédéfinis.

### 5.4.3 Enregistrement des formulaires

L'enregistrement des formulaires en base est fait de manière unitaire en fonction de l'onglet modifié. Ceci permet de ne pas mémoriser l'ensemble des champs à chaque modification.

Lors du submit, l'ensemble des champs renseignés est récupéré par le serveur. Si les contrôles de saisies sont valides, le dossier est mis à jour en mémoire et évalué, puis les valeurs sont insérées en base.

### 5.4.4 Formules

La structure de classe des formules est contenue dans le package fr.harmonia.

Une « factory » est responsable de la construction des formules à partir d'une chaîne de caractères.

Un « adapter » est responsable de l'interfaçage de ces formules avec le dossier (il met à disposition de la formule les valeurs des champs en entrée, s'occupe de résoudre les dépendances entre champs, de gérer la synchronisation des appels à l'évaluation des formules).

#### 5.4.4.1 Parsing des formules

Le parsing des formules est effectué grâce à JavaCC (version 4.0).



Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

Une grammaire simplifiée des formules Excel est implémentée pour générer la classe Parser, qui construit un arbre d'évaluation à base de classes héritées de Node.

La classe Parser est utilisée par la factory pour générer les formules sous forme d'arbres d'évaluation.

#### 5.4.4.2 *Evaluation des formules*

Les formules sont évaluées par appel à la méthode evalNode() du nœud racine de l'arbre d'évaluation.

En cas d'erreur au cours de l'évaluation, l'utilisateur est averti de l'erreur.

#### 5.4.4.3 *Stockage des formules*

Les formules sont stockées dans le fichier XML dans leur forme textuelle. Elles sont parsées et transformées en arbre à chaque instanciation d'un modèle de formulaire. Cette instanciation est réalisée une fois au démarrage du serveur.

## 6 MODELISATION

### 6.1 OBJETS METIER

Les dossiers sont divisés en deux parties :

- La définition du dossier (ou formulaire), créée à partir du fichier XML généré par le générateur de formulaire
- L'instance du formulaire (le dossier), qui contient pour chaque champ, la valeur saisie.

Le modèle de formulaire contient toute la logique de présentation et de remplissage d'un formulaire.

#### 6.1.1 fr.harmonia.model

Ce package contient tous les objets métiers.

Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7



Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7



Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

Libelle	java.lang.String			Libellé de la campagne. Peut être l'année, ou encore "ER 2006". Permettra plus tard de gérer des formulaires multiples la même année
millesime	java.lang.Short			Année de la campagne
Ouverte	java.lang.Boolean			Campagne ouverte : les dossiers ont le droit d'être initialisés
Code	java.lang.String			Code
campagnesFormDef	Map <java.lang.String, CampagneFormDef>			

### Classe CampagneFormDef

Commentaire	Permet de filtrer l'affichage des modèles en fonction du type d'organisme (important surtout pour les ER : 3 types)
-------------	---

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
typeOrganisme	short			Type d'organisme auquel s'applique le modèle
formDefinition	FormDefinition			
campagne	Campagne			

### Classe Etape

Commentaire	La liste des étapes Persistence : fichier de configuration XML...
-------------	--

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
etpld	int		X	Numéro : 1 à 6
code	java.lang.String			Libellé court
libelle	java.lang.String			Libellé

### Classe EtapeFactory

Commentaire	Fiche EtapeFactory
-------------	--------------------

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
etapes	Map<java.lang.Short, java.lang.String>			

### Classe HarmoniaSession

Commentaire	Fiche HarmoniaSession
-------------	-----------------------

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
dossiersListeTravail	Dossier[]			La liste de travail de l'utilisateur associée à sa session
utilisateur	Utilisateur			Utilisateur associé à la session

### Classe Information

Commentaire	Contient les champs relatifs à un message d'information destiné aux utilisateurs
-------------	--

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
infold	java.lang.Long		X	Identifiant



Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

titre	java.lang.String			Titre du message
message	java.lang.String			Texte du message

### Classe ModificationInfo

Commentaire	Informations générales de dernières modifications d'un objet quelconque De nombreuses informations sont dé-normalisées car l'utilisateur peut disparaître, l'organisme changer de nom... Cette information ne possède pas de clé, elle est associée directement à chaque objet qu'elle qualifie.
-------------	--

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
modId	java.lang.Long		X	Identifiant unique de la modification
utilId	java.lang.Long			Id de l'utilisateur
utiNom	java.lang.String			Nom de l'utilisateur
utiPrenom	java.lang.String			Prénom
utiLogin	java.lang.String			Login
utiType	java.lang.Short			Type d'utilisateur (fédéré, organisme, partenaire)
structureId	java.lang.Long			Id de la structure
structureNom	java.lang.String			Nom de la structure (fédéré, organisme, partenaire)
dateModifi	java.util.Date			Date de modification

### Classe Organisme

Commentaire	Fiche organisme
-------------	-----------------

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
OrgId	java.lang.Long		X	Identifiant
moisCloture	java.lang.Short			Mois de fin de l'exercice comptable (SEM) (ex : 12 signifie le 31/12 de chaque année)
typeOrganisme	java.lang.Short			Le type d'organisme : SEM; ESH; COOP; OFFICE
Nom	java.lang.String			Nom de l'organisme
Siren	java.lang.String			Code Siren
codeFede	java.lang.String			Code attribué par la fédération
noDguhc	java.lang.String			N° DHC attribué par le Ministère
anneeDebut	java.lang.Short			Date de début de vie. Null = inconnue
anneeFin	java.lang.Short			Date de fin de vie. Null = inconnue
isFedere	java.lang.Boolean			Indique si l'organisme est fédéré ou non
responsable	Utilisateur			Responsable d'affaire de l'organisme
adresse	Adresse			

### Classe Partenaire

Commentaire	Un partenaire : les fédérations, la CGLLS, le Ministère, ...
-------------	--

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
parId	java.lang.Long		X	Identifiant
nom	java.lang.String			Nom : CGLLS, DGUHC, MIILOS
typePartenaire	java.lang.Short			Le type de partenaire : fédéré, CGLLS, CDC, autres...
compteHTTP	java.lang.String			répertoire dans lequel sont stockés les

Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

				fichiers exportés
adresse	Adresse			
accesPartenaire	Map <java.lang.String, AccesPartenaireAcl>			
accesCustoPartenaire	Map <java.lang.String, AccesPartenaireCustoAcl>			
accesPersoPartenaire	Map <java.lang.String, AccesPartenairePersoAcl>			

### Classe TypeOrganismeFactory

Commentaire	Fiche TypeOrganismeFactory
-------------	----------------------------

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
typeOrganismes	Map<java.lang.Short, java.lang.String>			

### Classe Utilisateur

Commentaire	Fiche utilisateur
-------------	-------------------

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
Utile	java.lang.Long		X	Identifiant
typeUtilisateur	java.lang.Short			Type d'utilisateur : O; D; F; P;
Nom	java.lang.String			Nom
Prenom	java.lang.String			Prénom
Civilite	java.lang.String			Civilité
Titre	java.lang.String			Titre
Email	java.lang.String			Mail
Tel	java.lang.String			N° tel
Fax(Mobile)	java.lang.String			Fax(Sur l'interface, il est remplacé par « Mobile », mais sur BDD, il reste encore en « Fax » -- Evols 2019 Besoin 15)
IsContact	java.lang.Boolean	False		True => l'utilisateur est un "contact privilégié" de sa structure
Login	java.lang.String			Login
Password	java.lang.String			Mot de passe crypté
IsServeur	java.lang.Boolean	False		Indique si l'utilisateur est en réalité un serveur qui utilise ce compte pour exécuter les web services
Profil	java.lang.String			Profil
Service	java.lang.String			Service
dateConn	java.util.Date			Date de connexion
dateLastConnexion	java.util.Date			Date de dernière connexion
Organisme	Organisme			
Partenaire	Partenaire			
modificationInfo	ModificationInfo			
Roles	HashMap <java.lang.String, Role>			
rolesEtape	HashMap <java.lang.String, RoleEtape>			

### Classe UtiRole



Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

<i>Commentaire</i>	La définition des rôles (simples et par étape) Persistance : fichier de configuration XML
--------------------	--

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
utild	Long		X	Identifiant
roleld	Long		X	Identifiant
code	java.lang.String			Code associé

### **Classe RoleEtape**

<i>Commentaire</i>	La définition des types d'utilisateurs qui peuvent se voir attribuer un rôle à une étape Persistance : fichier de configuration XML
--------------------	--

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
utild	Long		X	Identifiant
roleld	Long		X	Identifiant
etpid	Short		X	Identifiant
code	java.lang.String			Code associé

## 6.1.2 fr.harmonia.model.acl

### **Classe AccesAcl**

<i>Commentaire</i>	Définit les droits d'accès d'un partenaire à une campagne Abstraite
--------------------	--

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
isAcces	java.lang.Boolean			Accès autorisé aux utilisateurs de l'organisme ?
isTxLegal	java.lang.Boolean			Envoi du dossier « légal » au passage en étape 4?
isTxFiabilie	java.lang.Boolean			Envoi du dossier fiabilisé au passage en étape 6?
campagneAutorise	Campagne			Un paramétrage est valable pour une campagne, il est automatiquement reporté sur la campagne suivante

### **Classe AccesPartenaireAcl**

<i>Commentaire</i>	Autorise un partenaire à accéder au serveur pour chaque organisme et pour chaque campagne, Définit les paramètres par défaut pour chaque type d'organisme, plus l'autorisation ou non donnée au type d'organisme de surcharger les paramètres Les paramètres sont reconduits d'une campagne à l'autre
--------------------	---

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
typeOrganisme	Int		X	Type d'organisme auquel s'applique le paramétrage (facultatif)
isCustoAcces	java.lang.Boolean			La modification de l'accès par l'organisme est-elle autorisée ?
isCustoTxLegal	java.lang.Boolean			La modification de l'envoi du dossier « légal » au passage en étape 4 par l'organisme est-elle autorisée ?

Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

isCustoTxFiabilie	java.lang.Boolean			La modification de l'envoi du dossier fiabilisé au passage en étape 6 par l'organisme est-elle autorisée ?
-------------------	-------------------	--	--	--

### Classe AccesPartenaireCustoAcl

Commentaire	Paramétrage des autorisations spéciales accordées par l'organisme aux partenaires Les paramètres sont reconduits d'une année sur l'autre Si aucun paramètre n'est défini par l'organisme, les paramétrages serveur s'appliquent
-------------	---

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
organisme	Organisme			Non obligatoire : si aucun trouvé, s'applique à tous les organismes

### Classe AccesPartenairePersoAcl

Commentaire	Autorise un partenaire à accéder au serveur pour chaque organisme et pour chaque campagne. Définit les paramètres par défaut pour chaque organisme, plus l'autorisation ou non donnée à l'organisme de surcharger les paramétrages. Les paramètres sont reconduits d'une campagne à l'autre
-------------	---

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
organisme	Organisme			Non obligatoire : si aucun trouvé, s'applique à tous les organismes

### Classe AclFactory

Commentaire	Fiche AclFactory
-------------	------------------

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
serveurAcl	ServeurAcl			

### Classe Profil

Commentaire	Un profil utilisateur prédéfini sur un serveur Persistance : fichier de configuration XML
-------------	--

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
profilId	int			Identifiant
nom	java.lang.String			Libellé du profil
code	java.lang.String			Code du profil
typeUtilisateur	java.lang.String[]			Types d'utilisateurs auxquels s'applique le profil
roles	java.util.List			
rolesEtape	java.util.List			

### Classe Role

Commentaire	La définition des rôles (simples et par étape) Persistance : fichier de configuration XML
-------------	--

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
------	-----------------	-----------------	-----------------------	-------------



Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7



Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

rattachType	java.lang.String			Type de rattachement : CHAMP, TABLEAU
rattachCellule	java.lang.String			Rattachement à la cellule
rattachPartie	java.lang.Short			Rattachement à l'onglet
rattachOnglet	Java.lang.Short			Rattachement à la partie
modifInfo	ModificationInfo			Informations sur le modificateur
ordre	long			Ordre

### Classe Cliche

Commentaire	Un cliché est un dossier PDF stocké en base avec les données XML du dossier
-------------	---

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
clild	java.lang.Long		X	Identifiant
XML	Fichier			Fichier XML
PDF	Fichier			Fichier PDF
ZIP	Fichier			Fichier ZIP
etpld	Int			Etape à laquelle a été fait le cliché

### Classe Commentaire

Commentaire	Un commentaire est placé sur un dossier, un onglet de dossier ou un champ
-------------	---

### Classe Dossier

Commentaire	Le dossier : un ER ou un QDIS
-------------	-------------------------------

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
forId	java.lang.Long		X	Identifiant
typeDossier	int			Type du dossier (ER ou QDIS)
verrouille	java.lang.Boolean	False		Dossier verrouillé ?
hasErreurType3		False		Indique si la dernière évaluation a au moins une erreur de type 3
verrouDate	java.util.Date			Date de pose du verrou
distantDosId	java.lang.Long			Id du formulaire sur le serveur distant
distantEtapeNo	int			Etat du formulaire distant
distantEtapeDate	java.util.Date			Date de passage d'étape du formulaire distant
distantInfoDate	Java.util.Date			Date de synchronisation du formulaire distant
modifInfo	ModificationInfo			Informations de dernière modification
utilisateur	Utilisateur			
contact	Utilisateur			
organisme	Organisme			
questions	java.util.List <Commentaire>			
commentaires	java.util.List <Question>			
etat	EtatFormulaire			Etat courant du formulaire
transmission	Transmission			
piecesJointes	Java.util.List <Fichier>			
cliches	Cliche[]			
modele	FormDefinition			
fields	Map <java.lang.String, Field>			

Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

evaluated	Map <java.lang.String, Field>			
campagne	Campagne			

### Classe EtatFormulaire

Commentaire	Associe un formulaire à une étape courante.
-------------	---

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
etatEtape4	java.lang.Boolean			Etat de l'étape 4 : En cours ou En attente
etatMainSaisie	java.lang.Boolean			Etat de la main de saisie : Fédération ou Organisme
retourFiab	java.lang.Boolean			Indique si le dossier a une demande de retour en fiabilisation
modifInfo	ModificationInfo			Informations de modification du passage d'étape
etapeCourante	Etape			

### Classe Fichier

Commentaire	Contient des PDF, des XML ...
-------------	-------------------------------

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
ficld	java.lang.Long		X	Identifiant
nom	java.lang.String			Nom
extension	java.lang.String			Extension
mimeType	java.lang.String			Type mime
file	Java.sql.Blob			Contenu

### Classe Field

Commentaire	Un champ de formulaire et sa valeur
-------------	-------------------------------------

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
value	Object			Valeur du champ
dernValue	Object			
modifInfo	ModificationInfo			Informations sur la modification
delete	java.lang.Boolean			
questions	Question[]			
commentaires	Commentaire[]			
versionAnterieures	Field[]			

### Classe PieceJointe

Commentaire	Pièce jointe à un dossier dans l'espace d'échange
-------------	---

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
modifInfo	ModificationInfo			Informations sur la dernière modification

### Classe Question

Commentaire	Question posée dans l'espace d'échange
-------------	--

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
------	-----------------	-----------------	-----------------------	-------------





Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

isFermee	java.lang.Boolean	False		Question ouverte ou fermée
reponses	java.util.List<Reponse>			

### Classe Reponse

Commentaire	Réponse à une question posée dans l'espace d'échanges
-------------	---

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
question	Question			

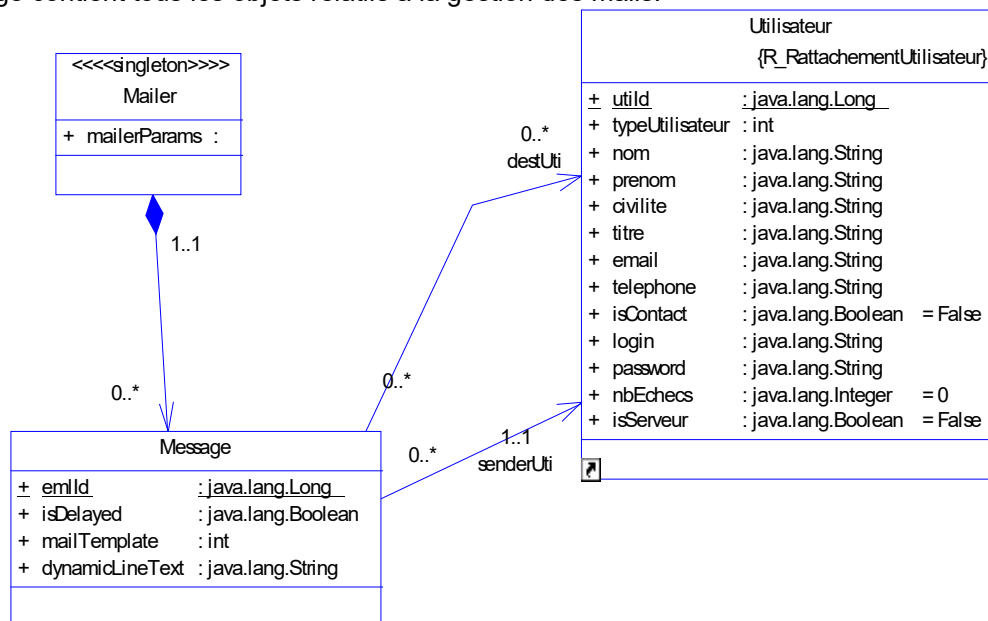
### Classe Transmission

Commentaire	Conserve toutes les informations relatives à une transmission
-------------	---

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
trxId	java.lang.Long		X	Identifiant
isSigne	java.lang.Boolean			La transmission a-t-elle fait l'objet d'une signature ?
certificat	byte[]			Certificat utilisé pour la signature
hashValuePdf	byte[]			Valeur de hash calculée du pdf
signaturePdf	byte[]			Valeur de hash signée du pdf
hashValueXml	byte[]			Valeur de hash calculée du xml
signatureXml	byte[]			Valeur de hash signée du xml
signataireModifInfo	ModificationInfo			Informations sur la modification
clicheSigne	Cliche			
recepisse	Fichier			
partenairesDestinat aires	Partenaire[]			

#### 6.1.4 Fr.harmonia.model.mail

Ce package contient tous les objets relatifs à la gestion des mails.



### Classe Message



Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

Commentaire	Représente un message à envoyer par mail
-------------	--

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
forId	long		X	Id du dossier
destId	long			Id du destinataire
Template	short			Template de mail à envoyer
typeUtilisateur	short			Type d'utilisateur (fédération, organisme, partenaire)
parId	java.lang.Long			
dynLineText	java.lang.String			Liste des paramètres à passer au template
email	java.lang.String			

### 6.1.5 Fr.hamonia.model.form

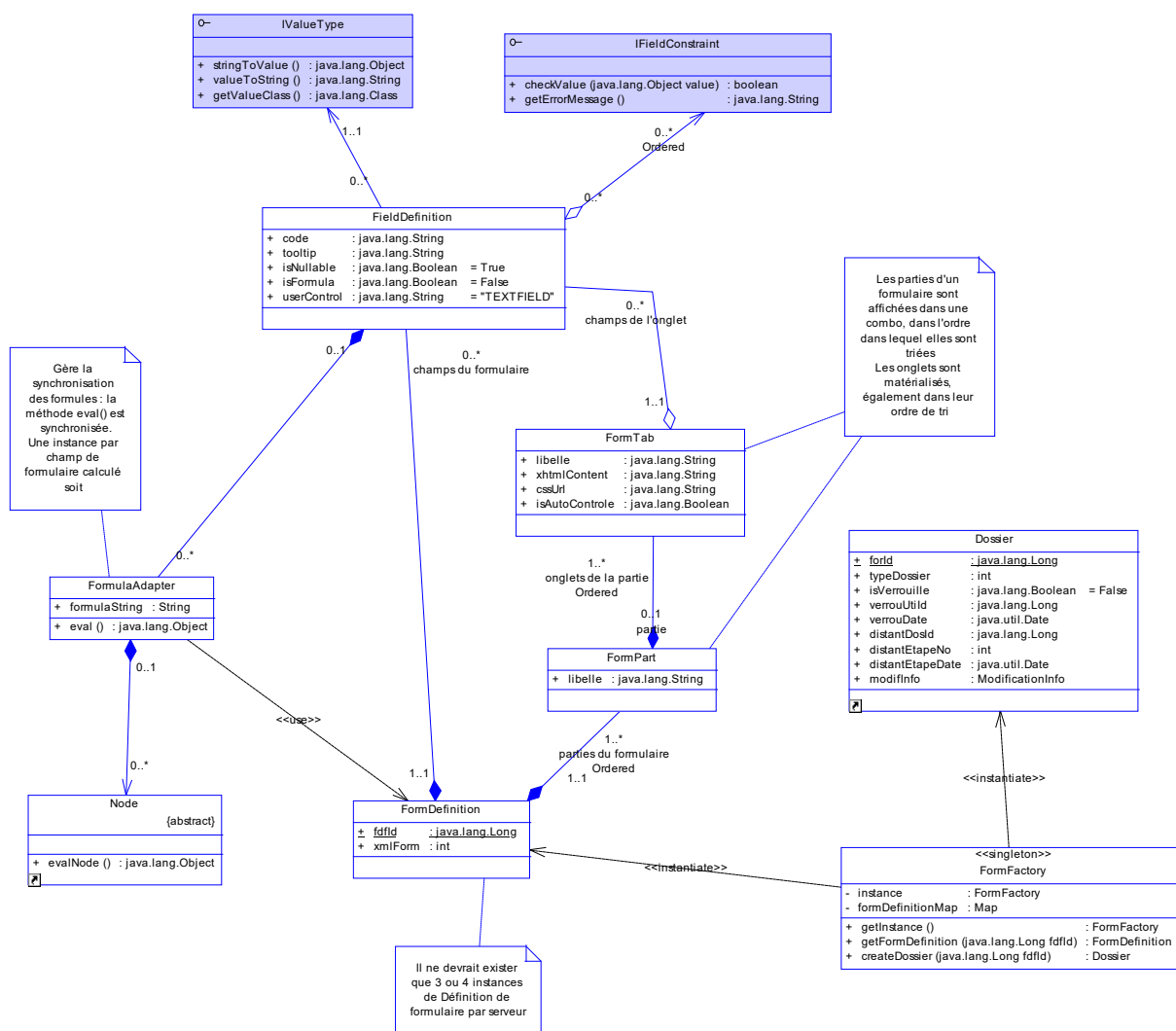
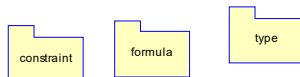
Ce package contient tous les objets relatifs à la gestion des formulaires.

# Document de spécifications

Capgemini

Spécifications techniques détaillées

SPF\_002/1.7



## Classe FieldDefinition

Commentaire	Définition d'un champ de données La persistance est faite en base dans un fichier XML
-------------	--

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
code	java.lang.String			code du champ
tooltip	java.lang.String			libellé d'aide
isNullable	java.lang.Boolean	True		champ nullable
isFormula	java.lang.Boolean	False		champ calculé



Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

userControl	java.lang.String	"TEXTFIELD"	Type de contrôle utilisé à l'affichage
tc	java.lang.Boolean		Indique si l'élément fait partie du TC

### Classe FormDefinition

Commentaire	Définition d'un formulaire de dossier
-------------	---------------------------------------

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
fdId	java.lang.Long		X	Identifiant
xmlForm	int			fichier xml contenant la définition

### Classe FormFactory

Commentaire	Factory qui instancie les définitions de formulaire
-------------	---

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
instance	FormFactory			Unique instance
formDefinitionMap	Map			liste des instances de définitions

Code	Type de résultat	Commentaire
getInstance	FormFactory	
getFormDefinition	FormDefinition	
createForm	Dossier	

### Méthode getFormDefinition

Code	Type de données	Commentaire
fdId	java.lang.Long	

### Méthode createForm

Code	Type de données	Commentaire
fdId	java.lang.Long	

### Classe FormPart

Commentaire	Partie de formulaire (un fichier excel)
-------------	---

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
Libelle	java.lang.String			

### Classe FormTab

Commentaire	Onglet de formulaire (un onglet Excel)
-------------	--

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
Libelle	java.lang.String			
xhtmlContent	java.lang.String			Contenu xhtml de l'onglet pour affichage



Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

cssUrl	java.lang.String			URL relative de la CSS associée à l'onglet
isAutoControle	java.lang.Boolean			Indique si l'onglet est un onglet d'autocontrôle

### Classe FormulaAdapter

Commentaire	Calcul des formules
-------------	---------------------

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
formulaString	String			Chaîne de la formule

Code	Type de résultat	Commentaire
Eval	java.lang.Object	Evaluation d'une formule. Méthode synchronisée

### Méthode eval

#### Interface IFieldConstraint

Commentaire	Représente une contrainte monoparamétrée sur un champ
-------------	---

Code	Type de résultat	Commentaire
checkValue	boolean	Valide la contrainte
getErrorMessage	java.lang.String	Récupère le message en cas d'erreur

### Méthode checkValue

Code	Type de données	Commentaire
Value	java.lang.Object	

### Méthode getErrorMessage

#### Interface IValueType

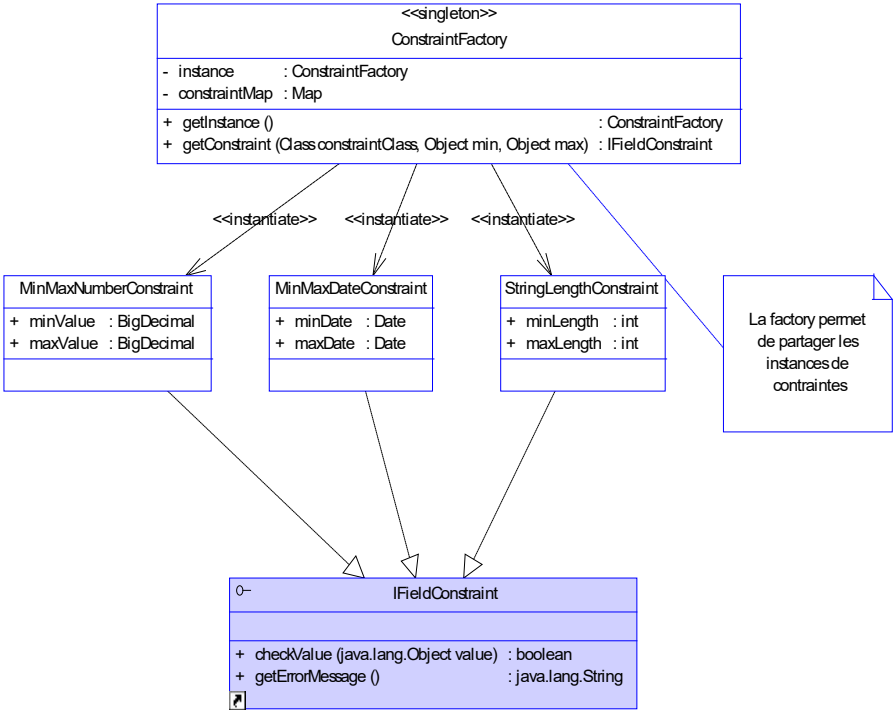
Commentaire	Type de données
-------------	-----------------

Code	Type de résultat	Commentaire
stringToValue	java.lang.Object	« Parse » une chaîne pour la transformer en valeur
valueToString	java.lang.String	Formate une valeur en une chaîne
getValueClass	java.lang.Class	Retourne le type d'objet sous-jacent

Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

### 6.1.5.1 *Fr.hamonia.model.form.constraint*

Ce package contient tous les objets relatifs à la gestion des contraintes de saisie et des formats des formulaires.



#### Classe ConstraintFactory

Commentaire	Factory qui crée un objet de contrainte par paramètres
-------------	--

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
instance	ConstraintFactory			instance unique
constraintMap	Map			Map des contraintes créées

Code	Type de résultat	Commentaire
getInstance	ConstraintFactory	Récupère l'instance unique
getConstraint	IFieldConstraint	Retrouve une contrainte

#### Méthode getConstraint

Code	Type de données	Commentaire
constraintClass	Class	Type de contrainte
min	Object	Paramètre 1
max	Object	Paramètre 2

#### Méthode getConstraint

Code	Type de données	Commentaire
constraintClass	Class	Type de contrainte
paramList	Object[]	Liste d'objet

#### Classe MinMaxDateConstraint



Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

Commentaire	Contrainte sur un champ date
-------------	------------------------------

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
minDate	Date			
maxDate	Date			

### Classe MinMaxNumberConstraint

Commentaire	Contrainte sur un champ nombre
-------------	--------------------------------

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
minValue	BigDecimal			
maxValue	BigDecimal			

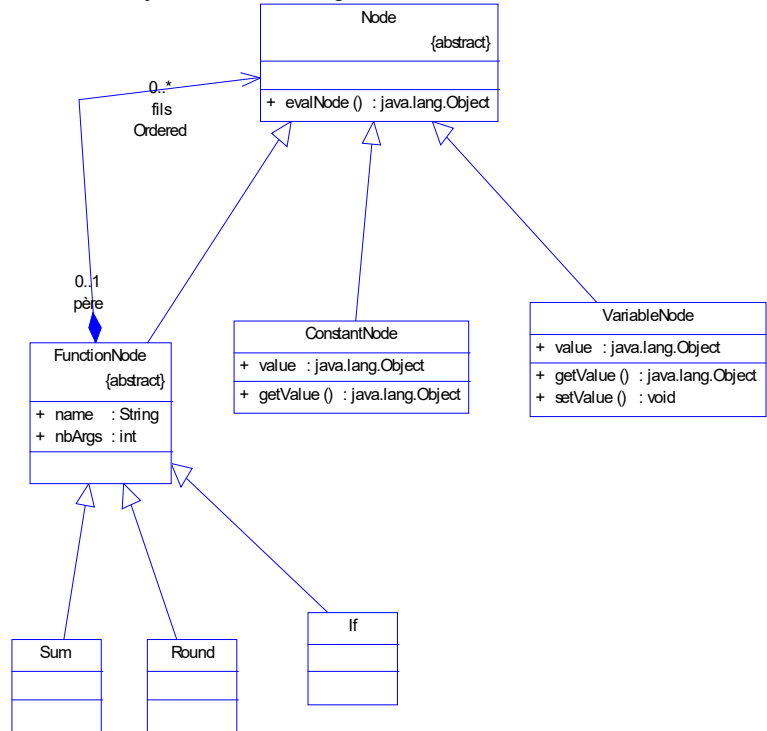
### Classe StringLengthConstraint

Commentaire	Contrainte sur la longueur d'une chaîne de caractères
-------------	---

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
minLength	int			
maxLength	int			

### 6.1.5.2 Fr.hamonia.model.form.formula

Ce package contient tous les objets relatifs à la gestion des formules dans les formulaires.



#### Classe ConstantNode

Commentaire	Une valeur fixe ( comme '566' par exemple)
-------------	--

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
value	java.lang.Object			

Code	Type de résultat	Commentaire
getValue	java.lang.Object	

#### Classe FunctionNode

Commentaire	Noeud d'un arbre de type fonction. Il est le seul a avoir des fils
-------------	---

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
name	String			
nbArgs	int			

#### Classe If

Commentaire	Branchement conditionnel
-------------	--------------------------

#### Classe Node



Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

<i>Commentaire</i>	Noeud d'un arbre d'évaluation.
--------------------	--------------------------------

<i>Code</i>	<i>Type de résultat</i>	<i>Commentaire</i>
evalNode	java.lang.Object	

#### Classe Round

<i>Commentaire</i>	Arrondi
--------------------	---------

#### Classe Sum

<i>Commentaire</i>	Méthode de sommation
--------------------	----------------------

#### Classe VariableNode

<i>Commentaire</i>	Une variable : un autre champ en l'occurrence
--------------------	---

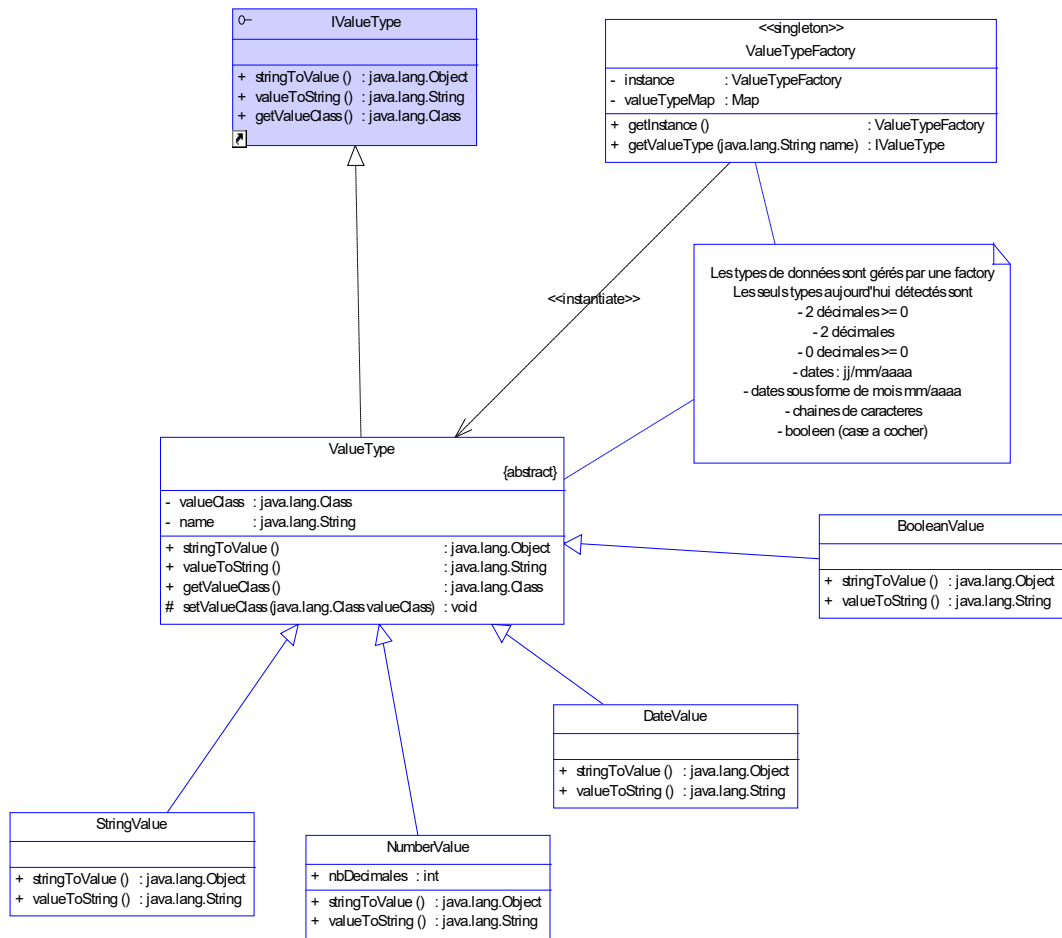
<i>Code</i>	<i>Type de données</i>	<i>Valeur initiale</i>	<i>Identifiant principal</i>	<i>Commentaire</i>
value	java.lang.Object			

<i>Code</i>	<i>Type de résultat</i>	<i>Commentaire</i>
getValue	java.lang.Object	
setValue	void	

#### **1.1.1.2 Fr.hamonia.model.form.type**

Ce package contient tous les objets relatifs à la gestion types de données dans les formulaires.

#### Package type



### Classe BooleanValue

Commentaire	Valeur Booléenne
-------------	------------------

Code	Type de résultat	Commentaire
stringToValue	java.lang.Object	
valueToString	java.lang.String	

### Classe DateValue

Commentaire	Valeur de type Date
-------------	---------------------

Code	Type de résultat	Commentaire
stringToValue	java.lang.Object	
valueToString	java.lang.String	

### Classe NumberValue

Commentaire	Valeur de type Nombre
-------------	-----------------------

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
nbDecimales	int			

Code	Type de résultat	Commentaire
stringToValue	java.lang.Object	
valueToString	java.lang.String	

Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

### Classe StringValue

Commentaire	Chaîne de caractère
-------------	---------------------

Code	Type de résultat	Commentaire
stringToValue	java.lang.Object	
valueToString	java.lang.String	

### Classe ValueType

Commentaire	Représente un type de valeur
-------------	------------------------------

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
valueClass	java.lang.Class			Classe java du type
name	java.lang.String			Nom du type

Code	Type de résultat	Commentaire
stringToValue	java.lang.Object	
valueToString	java.lang.String	
getValueClass	java.lang.Class	
setValueClass	void	Fixe la classe du type. Protégé car normalement uniquement fixé (« setté ») par la Factory

### Méthode setValueClass

Code	Type de données	Commentaire
valueClass	java.lang.Class	Classe java du type

### Classe ValueTypeFactory

Commentaire	
-------------	--

### Méthode getInstance

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
instance	ValueTypeFactory			
valueTypeMap	Map			

Code	Type de résultat	Commentaire
getInstance	ValueTypeFactory	
getValueType	IValueType	

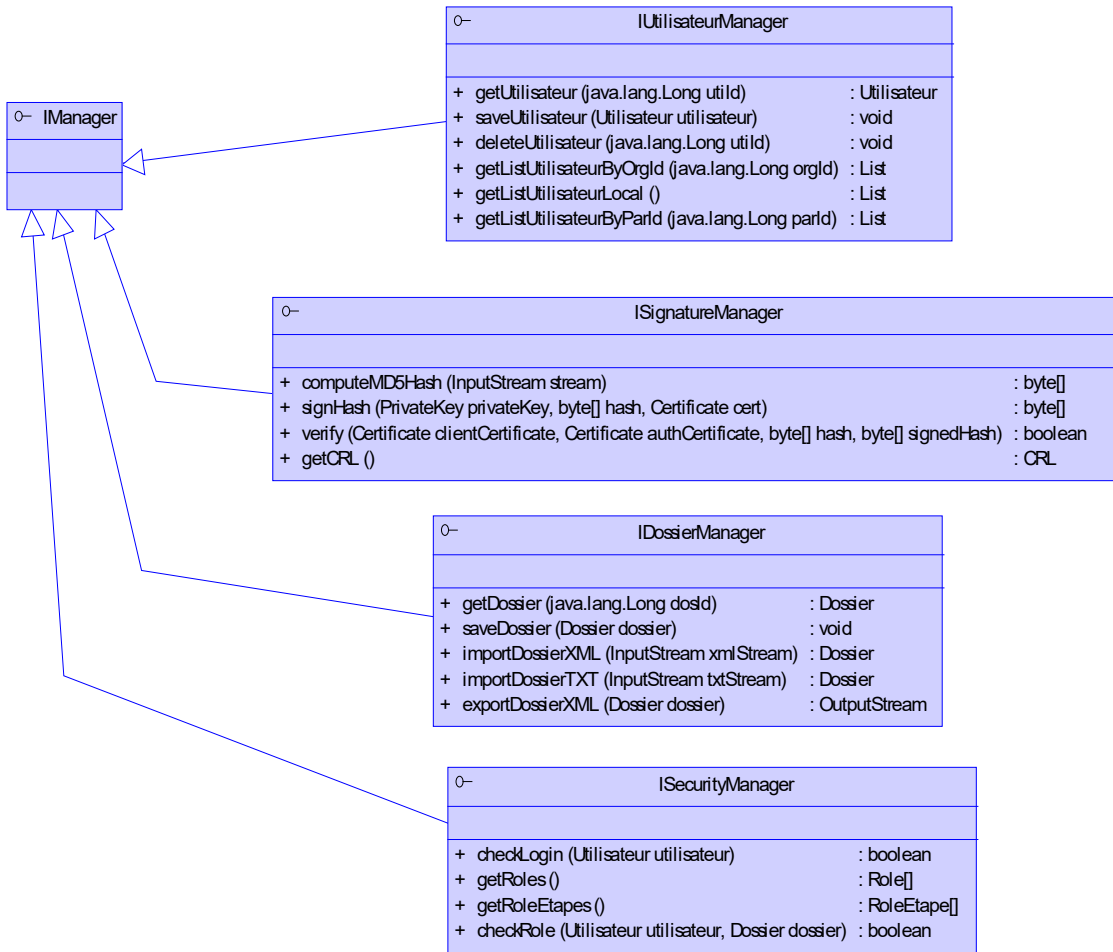
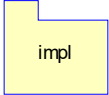
### Méthode getValueType

Code	Type de données	Commentaire
Name	java.lang.String	

## 6.2 SERVICES

### 6.2.1 Package fr.harmonia.service

Définit les interfaces des services disponibles sur les objets métier.



#### Interface IDossierManager

Commentaire	Gestion des dossiers
-------------	----------------------

Code	Type de résultat	Commentaire
getDossier	Dossier	lecture
saveDossier	void	écriture
importDossierXML	Dossier	import d'un dossier xml



Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

importDossierTXT	Dossier	import d'un dossier txt
exportDossierXML	OutputStream	export d'un dossier XML

#### Méthode getDossier

Code	Type de données	Commentaire
dosId	java.lang.Long	

#### Méthode saveDossier

Code	Type de données	Commentaire
Dossier	Dossier	

#### Méthode importDossierXML

Code	Type de données	Commentaire
xmlStream	InputStream	

#### Méthode importDossierTXT

Code	Type de données	Commentaire
txtStream	InputStream	

#### Méthode exportDossierXML

Code	Type de données	Commentaire
Dossier	Dossier	

#### Interface IManager

Commentaire	Interface dont héritent tous les services
-------------	---

#### Interface ISecurityManager

Commentaire	Gestion de la sécurité
-------------	------------------------

Code	Type de résultat	Commentaire
checkLogin	boolean	vérification login
getRoles	Role[]	récupération des rôles
getRoleEtapes	RoleEtape[]	récupération des rôles d'étape
checkRole	boolean	vérification de la possession d'un rôle

#### Méthode checkLogin

Code	Type de données	Commentaire
utilisateur	Utilisateur	

#### Méthode checkRole

Code	Type de données	Commentaire
utilisateur	Utilisateur	
dossier	Dossier	

#### Interface ISignatureManager

Commentaire	Service de signature. Il est également utilisé côté client.
-------------	---

Code	Type de résultat	Commentaire
------	------------------	-------------



Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

computeMD5Hash	byte[]	calcule un hash
signHash	byte[]	signe un hash
Verify	boolean	vérification d'une signature
getCRL	CRL	récupération des CRL

#### Méthode computeMD5Hash

Code	Type de données	Commentaire
stream	InputStream	

#### Méthode signHash

Code	Type de données	Commentaire
privateKey	PrivateKey	
hash	byte[]	
cert	Certificate	

#### Méthode verify

Code	Type de données	Commentaire
clientCertificate	Certificate	
authCertificate	Certificate	
Hash	byte[]	
signedHash	byte[]	

#### Méthode getCRL

#### Interface IUtilisateurManager

Commentaire	Service de gestion des utilisateurs
-------------	-------------------------------------

Code	Type de résultat	Commentaire
getUtilisateur	Utilisateur	lecture d'un utilisateur
saveUtilisateur	void	sauvegarde
deleteUtilisateur	void	effacement
getListUtilisateurByOrgId	List	liste des utilisateurs par organisme
getListUtilisateurLocal	List	liste des utilisateurs attachés au serveur
getListUtilisateurByParId	List	liste des utilisateurs par partenaire

#### Méthode getUtilisateur

Code	Type de données	Commentaire
utild	java.lang.Long	

#### Méthode saveUtilisateur

Code	Type de données	Commentaire
utilisateur	Utilisateur	

#### Méthode deleteUtilisateur

Code	Type de données	Commentaire
utild	java.lang.Long	

#### Méthode getListUtilisateurByOrgId

Code	Type de données	Commentaire
orgId	java.lang.Long	

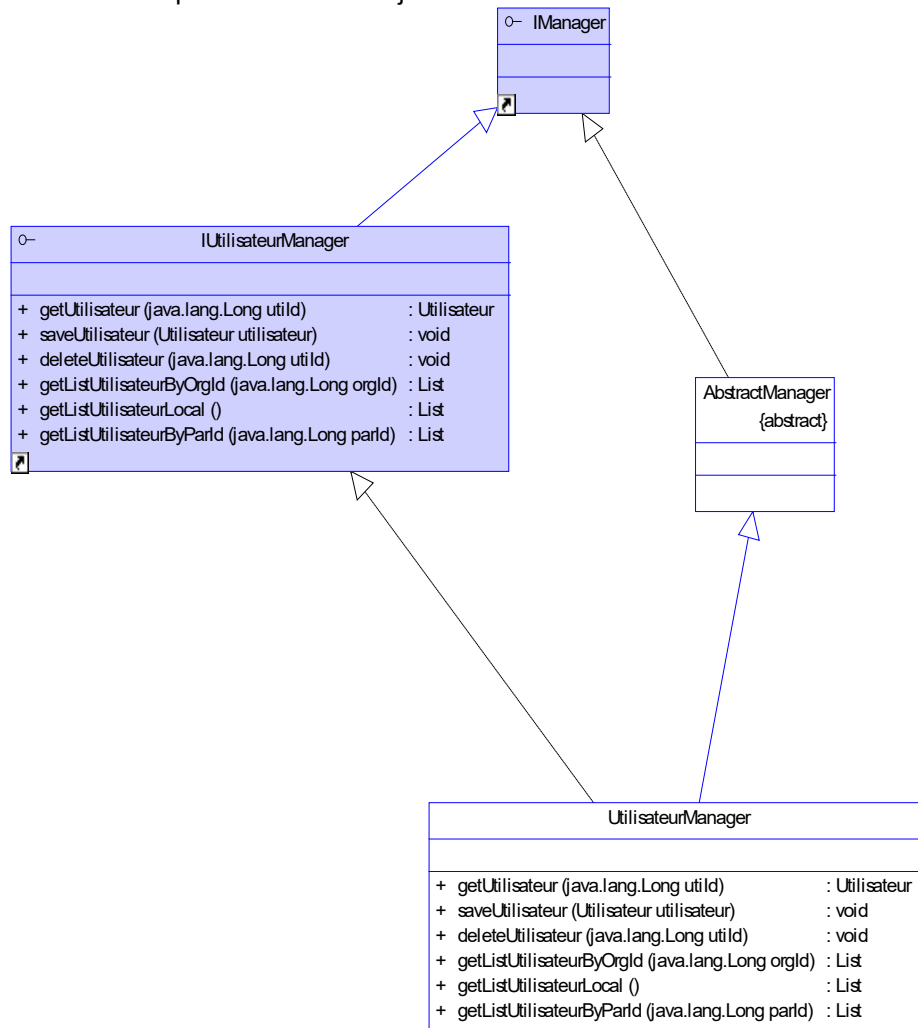
Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

### Méthode getListUtilisateurByParId

Code	Type de données	Commentaire
parId	java.lang.Long	

### 6.2.2 Package fr.harmonia.service.impl

Implémente les services disponibles sur les objets métier.



### Classe AbstractManager

Commentaire	
-------------	--

### Classe UtilisateurManager

Commentaire	
-------------	--

Code	Type de résultat	Commentaire
getUtilisateur	Utilisateur	
saveUtilisateur	void	
deleteUtilisateur	void	
getListUtilisateurByOrgId	List	
getListUtilisateurLocal	List	



Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

getListUtilisateurByParId	List	
---------------------------	------	--

#### Méthode getUser

Code	Type de données	Commentaire
utild	java.lang.Long	

#### Méthode saveUtilisateur

Code	Type de données	Commentaire
utilisateur	Utilisateur	

#### Méthode deleteUser

Code	Type de données	Commentaire
utild	java.lang.Long	

#### Méthode getListUtilisateurByOrgId

Code	Type de données	Commentaire
orgId	java.lang.Long	

#### Méthode getListUtilisateurByParId

Code	Type de données	Commentaire
parId	java.lang.Long	

### 6.2.3 Couche d'accès aux données

Cette couche permet l'accès aux données persistantes.

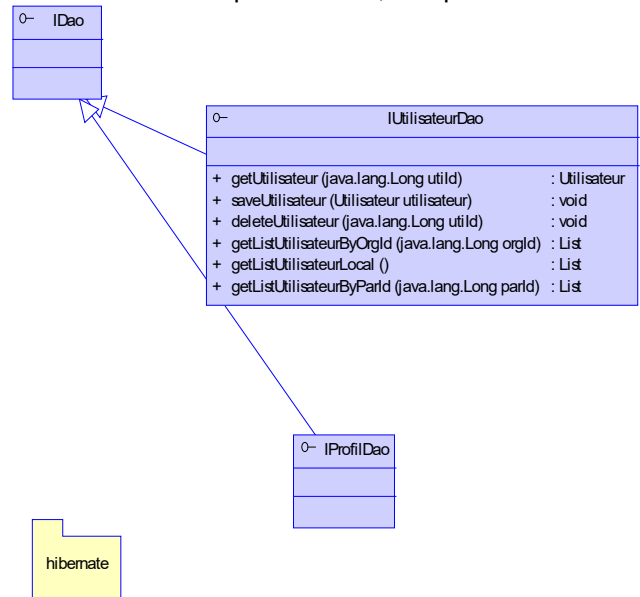
Une modélisation pour l'exemple a été faite avec un objet métier.



Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

### 6.2.3.1 Package fr.harmonia.dao

Contient les interfaces d'accès aux données persistantes, indépendamment du mode de persistance.



#### Interface IDao

Commentaire	Interface dont héritent toutes les interfaces de DAO
-------------	--

#### Interface IProfilDao

Commentaire	DAO des profils utilisateurs
-------------	------------------------------

#### Interface IUtilisateurDao

Commentaire	Interface d'un DAO Utilisateur : objet d'accès aux données utilisateurs
-------------	---

Code	Type de résultat	Commentaire
getUtilisateur	Utilisateur	lit un utilisateur
saveUtilisateur	void	sauvegarde un utilisateur
deleteUtilisateur	void	efface un utilisateur
getListUtilisateurByOrgId	List	Liste des utilisateurs par organisme
getListUtilisateurLocal	List	Liste des utilisateurs locaux
getListUtilisateurByParId	List	Liste des utilisateurs par partenaire

#### Méthode getUtilisateur

Code	Type de données	Commentaire
utild	java.lang.Long	

#### Méthode saveUtilisateur

Code	Type de données	Commentaire
utilisateur	Utilisateur	

#### Méthode deleteUtilisateur

Code	Type de données	Commentaire
utild	java.lang.Long	

Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

#### **Méthode getListUtilisateurByOrgld**

<i>Code</i>	<i>Type de données</i>	<i>Commentaire</i>
orgld	java.lang.Long	

#### **Méthode getListUtilisateurLocal**

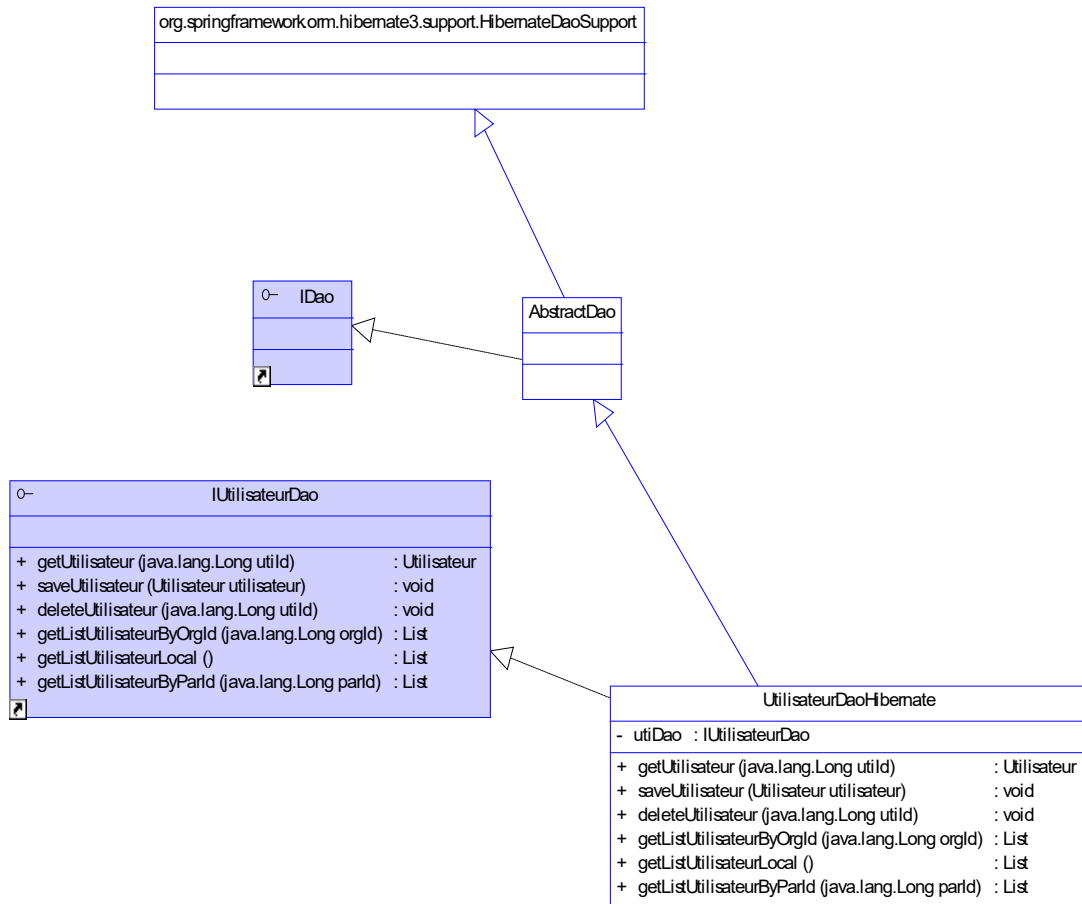
#### **Méthode getListUtilisateurByParld**

<i>Code</i>	<i>Type de données</i>	<i>Commentaire</i>
parld	java.lang.Long	

Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

### 6.2.3.2 Package fr.harmonia.dao.hibernate

Contient les implémentations Hibernate des interfaces d'accès aux données persistantes



#### Classe AbstractDao

Commentaire	
-------------	--

#### Classe UtilisateurDaoHibernate

Commentaire	
-------------	--

Code	Type de données	Valeur initiale	Identifiant principal	Commentaire
utiDao	IUtilisateurDao			

Code	Type de résultat	Commentaire
GetUtilisateur	Utilisateur	
SaveUtilisateur	void	
DeleteUtilisateur	void	
getListUtilisateurByOrgId	List	
getListUtilisateurLocal	List	
getListUtilisateurByParId	List	

Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

#### Méthode getUser

Code	Type de données	Commentaire
Utilid	java.lang.Long	

#### Méthode saveUtilisateur

Code	Type de données	Commentaire
utilisateur	Utilisateur	

#### Méthode deleteUser

Code	Type de données	Commentaire
utild	java.lang.Long	

#### Méthode getListUtilisateurByOrgId

Code	Type de données	Commentaire
orgId	java.lang.Long	

#### Méthode getListUtilisateurLocal

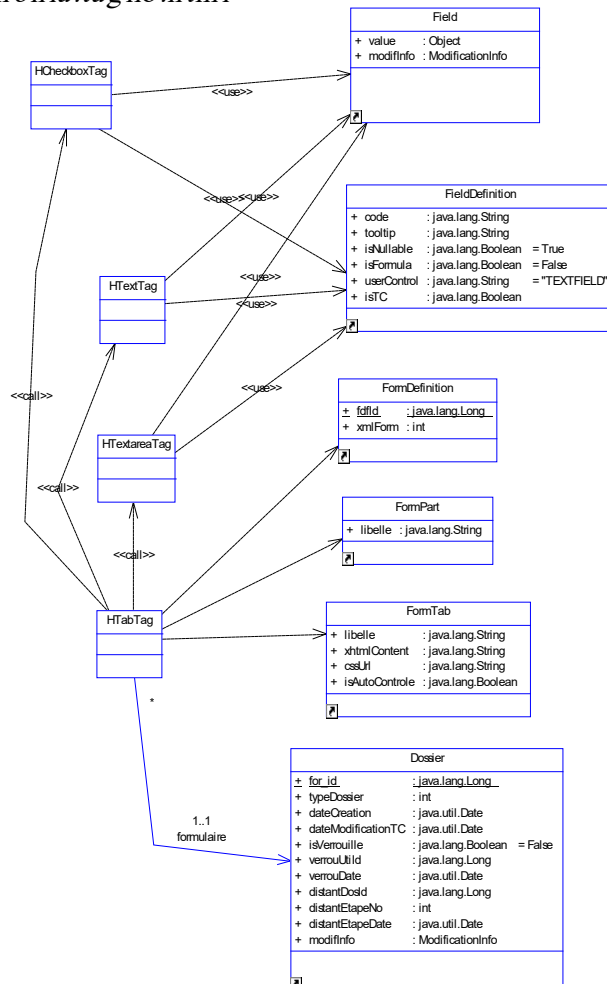
#### Méthode getListUtilisateurByParId

Code	Type de données	Commentaire
parId	java.lang.Long	

## 6.2.4 Couche Vue

Cette couche contient toutes les implémentations spécifiques au projet Harmonia des tags.

### 6.2.4.1 Package *fr.harmonia.taglib.html*

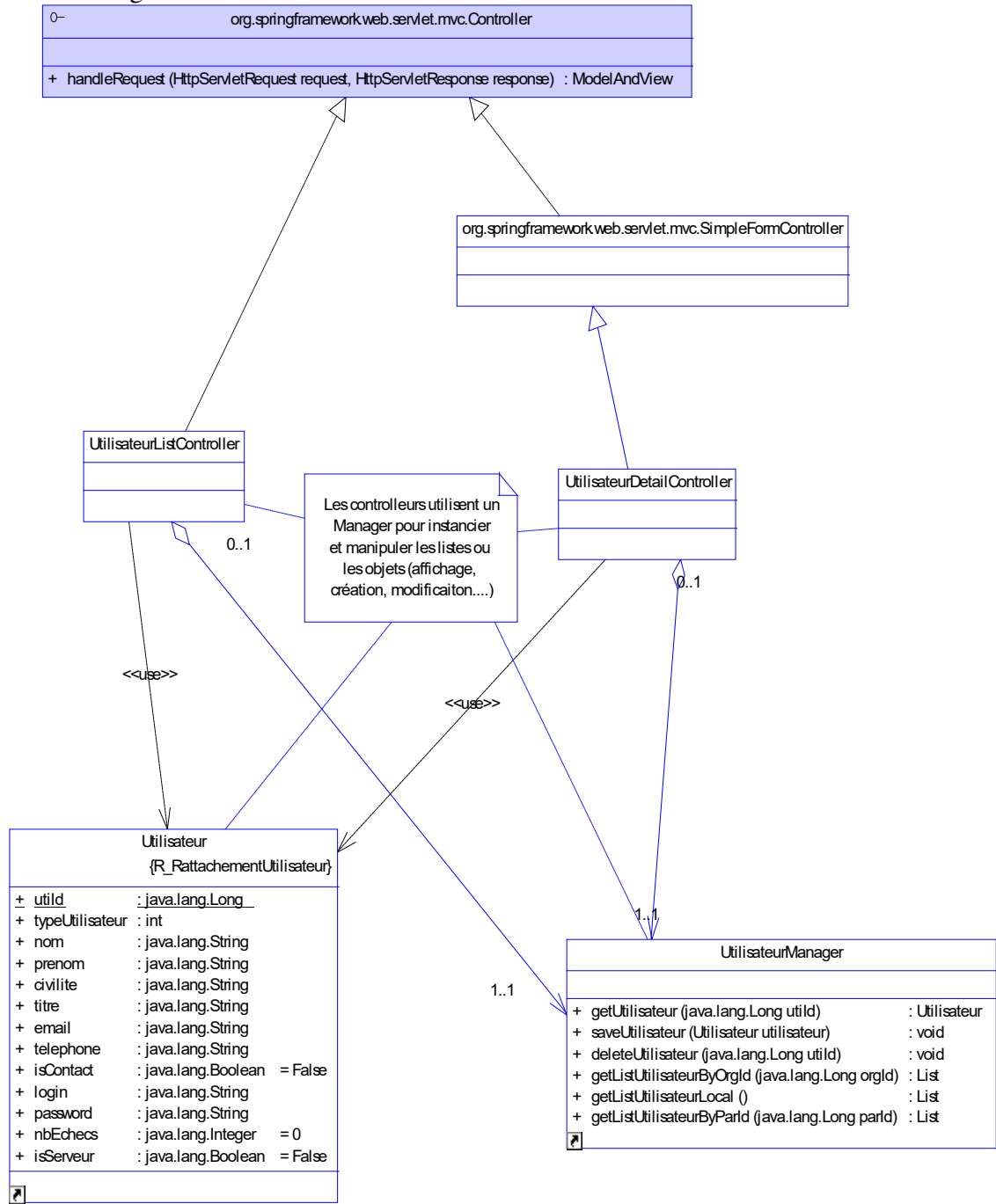


## 6.2.5 Couche Controller

Cette couche contient toutes les implémentations des contrôleurs qui gèrent les actions utilisateurs de chaque écran.

Une modélisation pour l'exemple a été faite avec un objet métier.

### 6.2.5.1 Package *fr.harmonia.controller*



## 6.3 WEB SERVICES

### 6.3.1 Web Services

#### 6.3.1.1 Principes communs à tous les web services

Plusieurs services web sont mis à disposition sur les serveurs Harmonia. Ils sont décrits dans les sous-sections de ce chapitre. Les **principes communs suivants** sont appliqués à chaque web service :

Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

- lorsqu'un appel à un web service exécute en ensemble d'opérations d'écritures en base de données, l'ensemble des opérations est effectuée avec une TRANSACTION SQL Server (BEGIN TRANSACTION, END TRANSACTION, et ROLLBACK en cas d'erreur)
- une méthode de gestion d'erreur est appelée et renvoie un code d'erreur en cas de problème :
- timeout (impossible de joindre l'hôte distant)
- mauvais couple login et mot de passe
- login et mot de passe inactivés
- rollback (retour) à la situation précédant l'appel pour problème dans la transaction
- blacklist : accès refusé pour ce demandeur

Dans les cas d'un timeout et d'un rollback, un deuxième envoi est automatiquement généré par l'appelant du web service.

- Les paramètres suivants sont communs à tous les paramètres d'entrée des Web Services. Ils permettent au serveur de vérifier l'identité du système appelant et de filtrer le cas échéant les droits selon l'utilisateur.

Les serveurs S1...S4 sont déclarés comme des utilisateurs de type « serveur » sur le serveur S5.

Le serveur S5 est déclaré comme utilisateur de type « serveur » sur les serveurs S1..S4.

- Login
- Mot de passe (crypté)

#### 6.3.1.2 Synchroniser les étapes de workflow (cycle de vie) des ER

Objet : Mettre à jour la table Dossier avec les informations relatives à l'étape courante des ER sur le serveur S5.

Plateforme : S1...S4.

Paramètres d'appel :

- Login
- Mot de passe (crypté)
- Dossier à synchroniser
  - Identifiant DGUHC du dossier  
OU
  - N° SIREN si identifiant distant non connu
  - Millésime  
OU
  - Millésime (synchronise tous les dossiers du millésime)

Retour :

- Liste d'informations sur l'organisme
  - Identifiant DGUHC (Ministère) du dossier
  - N° SIREN de l'organisme
  - Millésime de la campagne
  - N° de l'étape courante
  - Date de passage à l'étape courante

Règles de gestion particulière :

Seuls les dossiers que la fédération a le droit de voir (i.e. les dossiers fédérés de son type, soit les dossiers de ses adhérents) sont renvoyés.

En cas d'interrogation sur un dossier interdit, aucune information n'est retournée, aucune erreur n'est générée.



Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

Si le dossier n'est pas initialisé sur le serveur du Ministère, aucune information n'est retournée.

#### 6.3.1.3 Initialiser le formulaire

Objet : Permettre aux PGI d'initialiser le remplissage d'un dossier dans Harmonia. Le login/mot de passe est celui de l'utilisateur.

Plateforme : S1...S5.

Paramètres d'appel :

- Login
- Mot de passe (crypté)
- Dossier à synchroniser (non renseigné → tous)
  - o Millésime (l'organisme est déduit de l'utilisateur)
  - o Fichier XML au format Harmonisé

Retour :

- Code d'erreur ou de réussite
- Liste des champs en erreur

Règles de gestion particulière:

Seuls les champs renseignés dans le fichier XML sont modifiés dans le système.

En cas d'erreur, l'ancien champ n'est pas modifié.

#### 6.3.1.4 Importer les données du tronc commun

Objet : importer les données TC du dossier distant dans le dossier local.

Plateforme : S1...S5.

Paramètres d'appel :

- Login
- Mot de passe (crypté)
- ID du formulaire distant s'il est connu
- Campagne et N° SIRET sinon

Retour :

- ID du formulaire distant
- Date de dernière modification du formulaire
- Date de dernière modification du TC
- Etape courante
- Date de passage de l'étape
- Liste des données du TC
  - o Nom du champ
  - o Valeur
  - o Date de dernière modification

## 6.4 MODELE PHYSIQUE DE DONNEES

Un diagramme est fourni ci-après.



MPD 4.1.x



Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

## 7 NORMES DE CODAGE

### 7.1 CONVENTIONS DE NOMMAGE ET DE CODAGE JAVA/J2EE

Les normes de codage édictées par Sun sont respectées. Elles sont disponibles aux adresses suivantes :

Conventions de codage :

<http://java.sun.com/docs/codeconv/index.html>

Conventions de nommage :

<http://java.sun.com/blueprints/code/namingconventions.html>

Convention d'écriture des JSP :

[http://java.sun.com/developer/technicalArticles/jaserverpages/code\\_convention/](http://java.sun.com/developer/technicalArticles/jaserverpages/code_convention/)

En particulier, les règles suivantes seront scrupuleusement respectées :

- Externalisation systématique des chaînes de caractère utilisées comme libellés
- Limitation de l'utilisation de scripts dans les codes JSP, au profit de l'utilisation des Taglibs.

### 7.2 CONVENTIONS DE NOMMAGE SQL

Les noms de tables, de colonnes, d'index, de contraintes... sont en majuscules.

Les colonnes de clés primaires sont fabriquées d'un trigramme et du suffixe \_ID : TRG\_ID

Les colonnes de clés étrangères sont fabriquées ainsi : TRG\_<FONCTION>\_ID, où TRG\_ID est le nom de la clé primaire référencée, et <FONCTION> un code fonctionnel représentant la relation.

Les noms de contraintes sont suffixés par \_PK, \_FK...

Les noms des index sont suffixés par \_IDX

Seules les clés primaires, étrangères ou alternatives font l'objet de contraintes de non nullité en base.

## 8 DIMENSIONNEMENT

### 8.1 CONDITIONS DE MESURE DE L'ETUDE

Les éléments présentés sont basés sur les statistiques techniques d'utilisation collectées entre le 04 et le 05 avril 2013 sur les serveurs Harmonia :

- L'espace disque de chaque serveur
- Les bases de données de chaque serveur
- Les logs de performance Windows (« Perflogs ») activés sur le serveur <https://harmonia.logement.gouv.fr> pour les programmes Apache et Tomcat
- Les logs d'accès Apache (« access\_logs ») sur le serveur <https://harmonia.logement.gouv.fr>, notamment l'identifiant de session Tomcat (« jsessionid ») qui est systématiquement loggé et permet de distinguer une session (équivalent à un « utilisateur simultané ») d'une autre.

### 8.2 UTILISATION DE LA MEMOIRE VIVE

#### 8.2.1 Démarrage des serveurs

Les machines ont été observées au démarrage des serveurs Tomcat pour déterminer la quantité de RAM utilisée a minima par Tomcat pour démarrer, charger l'application web Harmonia et monter en mémoire les modèles de formulaires.

Plusieurs mesures (avec 0 puis n modèles) ont permis de déterminer que :

- La RAM utilisée pour charger Tomcat et Harmonia est de 246 Mo
- La quantité de RAM consommée pour charger un modèle dépend du serveur, puisque les modèles sont différents



Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

Serveur	Unité	MINISTERE	EPL	ESH	COOP	OPH
2013						
Mémoire Tomcat au démarrage	Mo	1080	268	332	432	339
Nombre de modèles chargés		20	5	9	14	9
Estimation de la taille d'un modèle en mémoire	Mo	41,7	4,4	9,5	13	10

Par ailleurs, l'impact de la connexion d'un utilisateur à Harmonia a été constaté empiriquement par l'augmentation constatée de mémoire lors de la connexion d'un utilisateur (il est très difficile d'estimer la taille réelle d'une session).

L'augmentation moyenne constatée est de 15 Mo.

#### 8.2.2 Estimation de l'utilisation de la mémoire durant la déclaration

L'étude a été menée sur le serveur du Ministère sur la base des logs du 05 avril 2013. L'objet était de mesurer l'utilisation réelle du serveur, de déterminer un « profil type » d'utilisateur pendant la phase de déclaration, et d'estimer la quantité de ressources à réserver pour un utilisateur afin de définir la capacité réelle de montée en charge du serveur.

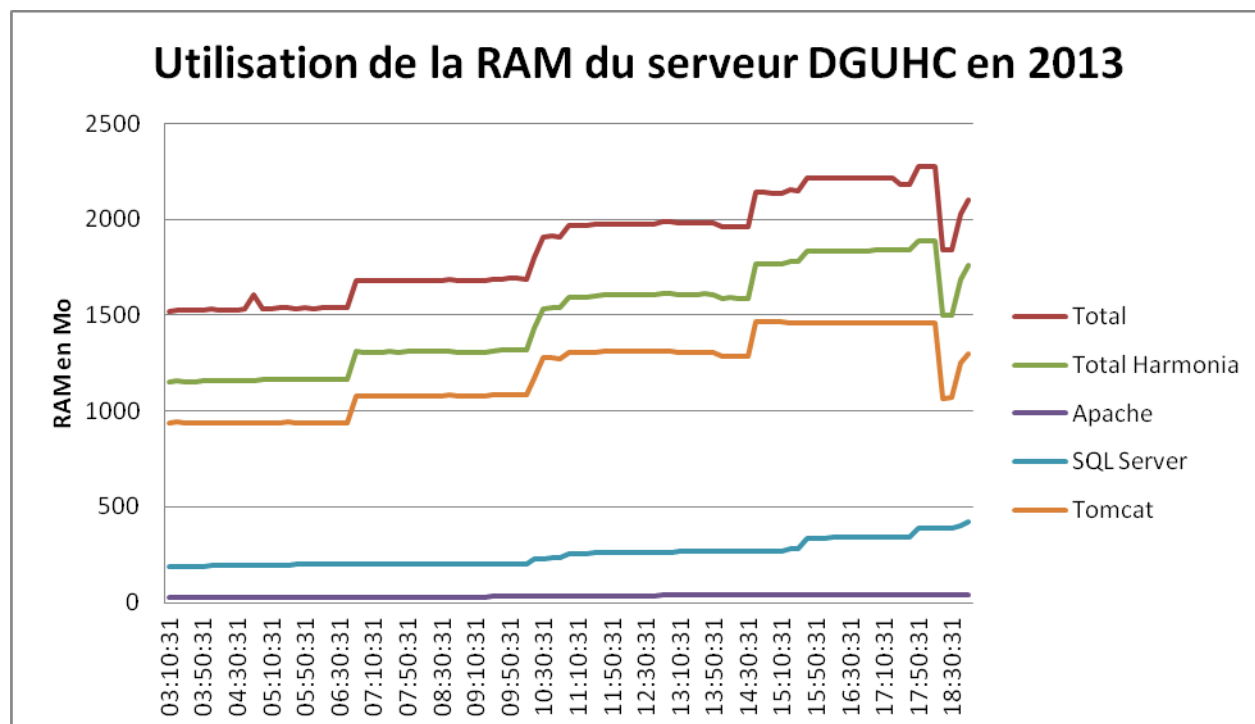
Le graphique suivant montre l'utilisation de la mémoire vive (RAM) faite par les différents composants d'Harmonia :

- Le serveur web Apache
- Le serveur applicatif Tomcat
- Le serveur de base de données SQL Server
- Le total de ces 3 composants : Total Harmonia
- Le total des ressources consommées : Total

On constate que :

- SQL Serveur utilise de 185 Mo en début de journée à 425Mo en fin de journée.
- Tomcat utilise entre 994 et 1500 Mo
- Apache utilise 40 Mo de mémoire

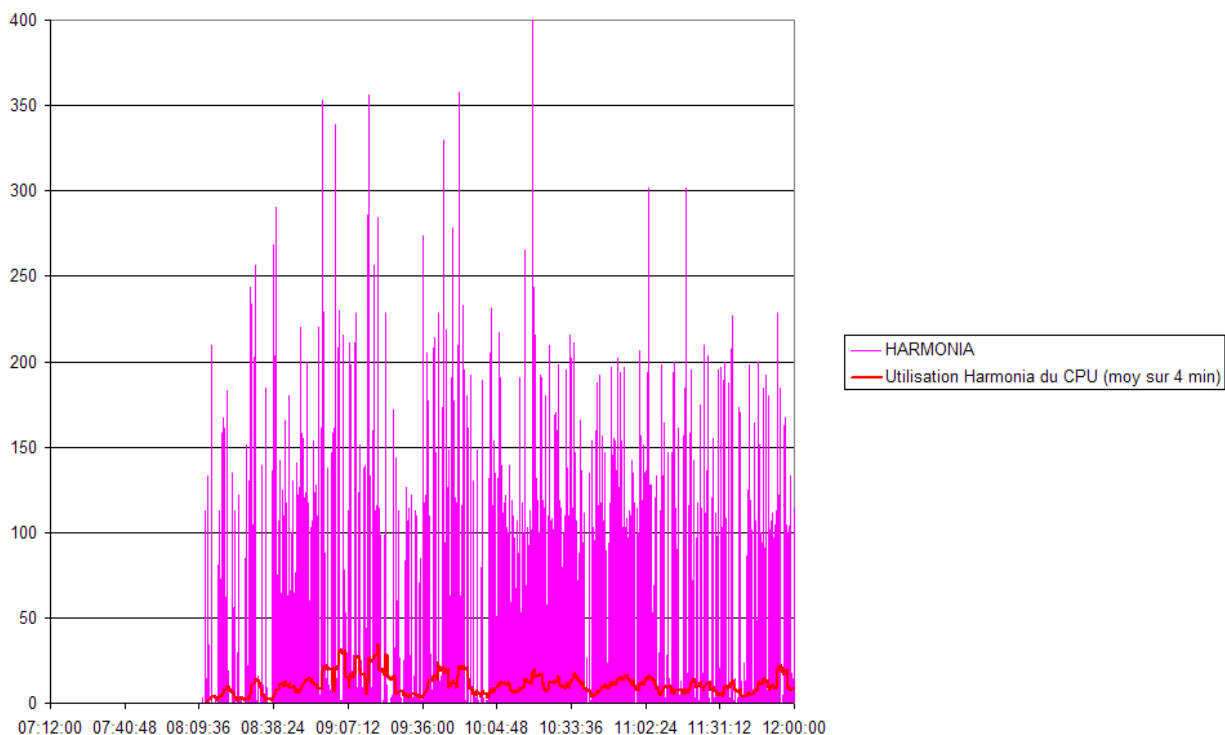
Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7



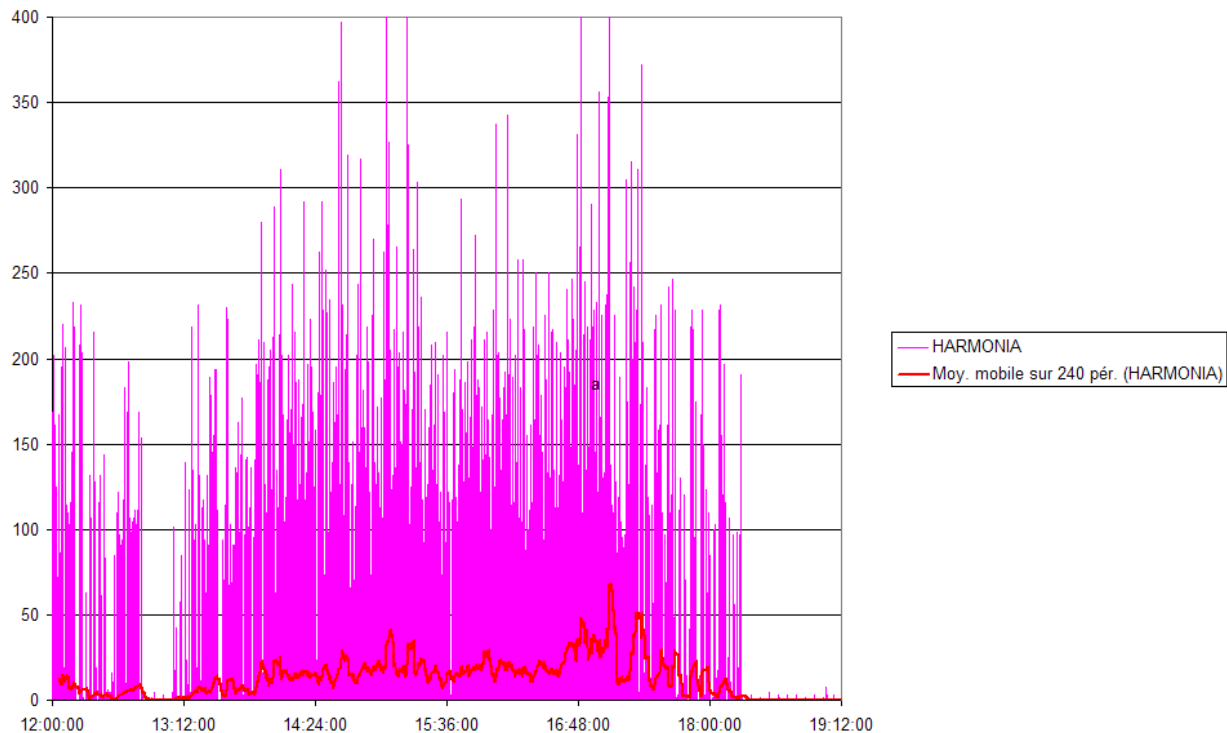
### 8.3 UTILISATION DU CPU DU SERVEUR

Les statistiques d'utilisation du CPU montrent que le CPU est rarement saturé (400 est le maximum, soit 100% d'utilisation du CPU).

La moyenne glissante sur 4 minutes montre une utilisation limitée des ressources CPU (maximum 25%).



Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7



### 8.3.1 Utilisation de la bande passante

Les statistiques d'Apache permettent d'obtenir les statistiques suivantes :

- Moyenne sur 1 seconde
  - bande passante maximum entrante : 800 ko/s
  - bande passante maximum sortante : 2,3 Mo/s
- Moyenne sur 5 minutes
  - bande passante maximum entrante : 43 ko/s
  - bande passante maximum sortante : 101 ko/s
- Total journalier
  - total entrant : 91 Mo
  - total sortant : 1,8 Go

### 8.3.2 Statistiques sur les temps de réponse des pages de l'application

La taille moyenne des pages (fichier .do) est de 150ko.

Le temps de réponse moyen pour ces pages est de 0,8s.

Le temps de réponse maximum pour 50% des pages est de 0,12s.

Le temps de réponse maximum pour 75% des pages est de 0,64s.

Le temps de réponse maximum pour 90% des pages est de 0,87s.

## 8.4 UTILISATION DE L'ESPACE DISQUE

Serveur	Unité	Ministère	EPL	ESH	COOP	OPH
---------	-------	-----------	-----	-----	------	-----



Document de spécifications						
Capgemini	Spécifications techniques détaillées				SPF_002/1.7	

Disque C utilisé	Go	180,00	51,40	49,30	39,10	39,90
dont Programmes Harmonia	Go	78,20	2,96	7,87	6 ,02	5,00
logs, répertoires temporaires,backup	Go	26,40	22,50	7,19	2,77	7,19
données de la base SQL	Go	31,80	0,58	8,23	4,29	2,01
Disque C libre	Go	169,00	8,40	10,5	5,77	4,97

## 8.5 STATISTIQUES DE LA BASE DE DONNEES

L'étude a ici pour objet

- Une photo de la volumétrie des principales tables
- Une estimation de la volumétrie d'un dossier en étape 6

Serveur	Unité	Ministère	EPL	ESH	COOP	OPH
Taille globale du schéma Harmonia	Mo	1238	139	842	805	368
Nombre de dossiers maximal par campagne (source campagne de collecte des comptes 2007)		933	162	280	165	280
Nombre de dossiers en base		4638	786	2450	2314	2483
dont Etape 1		287	1	28	19	45
dont Etape 2		37	0	0	1	3
dont Etape 3		93		0	1	2
dont Etape 4 et +		3072	782	2383	2276	2409
Taille moyenne des fichiers générés						
AR transmission	ko	70	64	66	67	66
Nombre		3092	782	2383	2289	2410
ER / QDIS PDF	ko	758	362	268	315	270
Nombre		3281	3	2291	1212	3
ER / QDIS xml	ko	536	46	64	117	58
Nombre		3281	3	2291	1212	3
ER / QDIS ZIP (PDF + XML)	ko	651	302	220	305	196
Nombre		6345	785	4673	3489	2412
Nb de pièces jointes		13425	18	8343	1098	878
Etape 1,2,3		87	0	0	0	0
Etape 4 et +		13338	18	8343	1098	878

Document de spécifications						
Capgemini	Spécifications techniques détaillées				SPF_002/1.7	

Taille moyenne	ko	1331	1352	667	857	748
Nb de pièces jointes moyen par dossier						
Etape 1,2,3		0.21	0	0.0	0	0
Etape 4 et +		4.34	0,02	3.5	0.48	0.36
Taille moyenne des fichiers associés au dossier	Ko					
Etape 1, 2, 3 <sup>(1)</sup>	Ko	10461	1194	27165	24933	20
Etape 4,5 <sup>(2)</sup>	Ko	6498	393	2617	782	530
Etape 6 <sup>(3)</sup>	Ko	7079	631	2771	1021	661
Table VALEUR						
Nb de lignes	k	22839	610	2359	6224	2653
DATA	Mo	1983	46	180	550	207
INDEX	Mo	3449	95	392	933	430
Table FICHIER	Mo	26610	327	7645	2750	1338
Autres Tables	Mo	135	8	24	53	32
Taille moyenne d'un dossier hors fichiers <sup>(4)</sup>	ko	467	70	85	266	98
Taille moyenne d'un dossier en étape 6 <sup>(5)</sup>	ko	7545	701	2856	1287	759

(1) : Taille moyenne d'une PJ x Nombre de pièces jointes moyen par dossier en étapes 1,2,3  
+ Nombre x « Taille moyenne d'un ER / QDIS PDF » / « Nombre de dossiers en étapes 1, 2 et 3 »  
+ Nombre x « Taille moyenne d'un ER / QDIS XML » / « Nombre de dossiers en étapes 1, 2 et 3 »

(2) : Taille moyenne d'une PJ x Nombre de pièces jointes moyen par dossier en étapes 4 et +  
+ « Taille moyenne d'un ER / QDIS ZIP » (*cliché de l'étape 4*)  
+ « Taille moyenne d'un AR de transmission » (*AR de l'étape 4*)

(3) : Taille moyenne d'une PJ x Nombre de pièces jointes moyen par dossier en étapes 4 et +  
+ « Taille moyenne d'un ER / QDIS ZIP » x 2 (*cliché de l'étape 4 et de l'étape 6*)  
+ « Taille moyenne d'un AR de transmission » (*AR de l'étape 4*)

(4) Table Valeur + autres tables / Nombre total de dossiers

(5) Taille moyenne d'un dossier hors fichiers + Taille moyenne des fichiers associés au dossier en étape 6



Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

## 8.6 PROJECTION SUR 5 ANS DE L'ESPACE DISQUE

Suite à la mise en place d'une purge de la base de données, sa taille devrait se stabiliser autour des valeurs actuelles

Serveur	Unité	Ministère	EPL	ESH	COOP	OPH
Taille globale de la base en 2013	Mo	31 800	551	8 240	4 290	2 110

Ces valeurs ne tiennent pas en compte l'ajout de nouvelles enquêtes sur les différentes plateformes.

## 8.7 SESSION – INFORMATIONS GENERALES

Nombre de sessions simultanées :

En théorie l'ensemble des organismes inscrits sur chaque plate-forme peut être connecté simultanément de même que l'ensemble des autres utilisateurs (Ministère, fédérations, partenaires).

Les pics de saisonnalité sont centrés sur la période juin – juillet de chaque année.

Sur cette période, on peut estimer une fréquentation pouvant aller en pointe jusqu'à 500 sessions simultanées sur la plate-forme du Ministère.

Sur les plates-formes des fédérations, on peut estimer en pointe, pour la même période juin – juillet, un pourcentage de sessions simultanées correspondant à 70% des organismes inscrits. La période de décembre / janvier sur les plateformes des fédérations proposant des enquêtes fédérales peut également occasionner de l'activité.

En dehors de ces périodes d'activité, les nombres de sessions simultanées sont très sensiblement plus faibles.

Pour la plate-forme des ER, on peut les estimer comme fluctuant entre 20 et 500.

Pour les plate-formes des fédérations, on peut les estimer comme fluctuant entre 5 et 50% des organismes inscrits sur chaque plate-forme.

Taille d'une session : 8 Mo maxi

Durée d'une session inactive : 60 minutes.

Aucun message ne précise que la session va se terminer. En cas de connexion après le temps de session, un message indique qu'il est nécessaire de s'identifier de nouveau.

## 8.8 STOCKAGE – INFORMATIONS GENERALES

L'évaluation de la taille nécessaire d'espace disque passe par une évaluation des informations qui la composent.

Les données « vivantes » que sont les données XML et les PDF photo sont stockées en tant que BLOB dans le SGBD. Ce qui équivaut sensiblement à la taille des fichiers eux même.

Les données « de gestion » sont les composants logiciels installés sur chaque serveur.

### 8.8.1 Données « vivantes »

#### 8.8.1.1 Evaluation de la taille des données XML

En partant du principe qu'un fichier XML de données comporte 8 000 champs renseignés de données pour les ER des ESH, des COOP et des OPH, la taille du XML associée est de l'ordre de 500 Ko.



Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

En prenant comme base de calcul le serveur du Ministère où tous les organismes déclarent (fédérés + non fédérés) : 900 organismes environ, on obtient  $900 * 500 \text{ Ko} = 450 \text{ Mo}$ .

Données qu'on multiplie par 2 pour tenir compte de la sauvegarde du document transmis (étapes 3 et 6), soit 900 Mo.

#### 8.8.1.2 *Evaluation de la taille des données PDF*

Le nombre d'écrans de formulaires avait été estimé en 2006 à environ 60 pour l'ER et chaque QDIS.

Un PDF compte en moyenne 2 pages par écran.

La taille d'une page PDF est de 15 Ko environ en moyenne.

En partant du principe que tous les onglets d'un dossier (même multi parties) sont sauvegardés en PDF, la taille globale d'un PDF d'un ER (supposition qu'un ER est plus volumineux qu'un QDIS) est de :

$$15 * 2 * 60 = 1,8 \text{ Mo}$$

En prenant comme base de calcul le serveur du Ministère où tous les organismes déclarent (fédérés + non fédérés) : 900 organismes environ, on obtient  $900 * 1,8 \text{ Mo} = 1,6 \text{ Go}$ .

Données qu'on multiplie par 2 pour tenir compte de la sauvegarde du document transmis (étapes 3 et 6), soit **3,2 Go**.

#### 8.8.1.3 *Evaluation de la taille fichiers attachés*

Avec des aléas très forts, on peut estimer à 3 Go minimum ces données à terme rapproché.

## 8.9 RESEAU

Les études faites en 2006-2007 par le maître d'œuvre annonçaient qu'il fallait un débit de 2 Mbits/seconde sur le serveur du Ministère.

Les statistiques fournies par l'hébergeur sur la période mai- septembre 2008 montrent qu'en pointe la consommation de bande passante est restée nettement inférieure à 2 mbits/seconde sur chacun des 5 serveurs.

## 9 CONFIGURATION DU SERVEUR

La configuration est renseignée dans le manuel de paramétrage.

Voici de façon non exhaustive les items principaux qui y sont décrits.

### 9.1 ORGANISATION DES REPERTOIRES

Disque C:\

OS, applicatifs, Webapp

Données de la base de données

### 9.2 FICHIERS DE CONFIGURATION

OS

- Locales : fr\_fr

Apache :

- httpd.conf
- mod\_ssl.conf
- mod\_jk.conf
- workers.properties

Tomcat

- server.xml

SQLServer

## 10 TERMINOLOGIE - ABREVIATIONS

ACL : Access Control List





Document de spécifications		
Capgemini	Spécifications techniques détaillées	SPF_002/1.7

AOP : Aspect Oriented Programming (Programmation Orientée Aspect)

DAO: Data Access Object

IoC : Inversion of Control

JSP : Java Server Pages

MVC: Modèle - Vue - Contrôleur (Model View Controller)

SOAP: Simple Object Access Protocol