

BROSSARD Florian

Master ILC 1<sup>re</sup> année

24 avril 2018

---



MISSION EN ENTREPRISE  
MÉMOIRE PREMIÈRE ANNÉE

---



Tuteur universitaire : Stéphane CATELOIN  
Maître d'apprentissage : Sabrina MIGNON

# Liste des points à traiter

	Désactiver les todos une fois le document terminé . . . . .	1
	Ajouter un paragraphe sur le problème de l'ajout d'environnement et de la notion de type d'environnement ? . . . . .	17

## Remerciements

Je tiens à remercier l'ensemble des membres du personnel travaillant au sein des sociétés ES et Atos pour leur gentillesse, leur accueil et leur disponibilité.

Je remercie Mme. MIGNON Sabrina, chef de projet chez Atos et mon maître d'apprentissage, pour le temps qu'elle m'a accordé afin de me former et de m'aider.

Je tiens également à remercier M. CATELOIN Stéphane, mon tuteur à l'université, pour la rapidité et la clarté avec lesquelles il a su répondre à mes questions pendant mon apprentissage.

---

# Table des matières

<b>1</b>	<b>Contexte</b>	<b>1</b>
1.1	Le groupe Atos . . . . .	2
1.2	Le groupe ÉS . . . . .	3
1.3	Les activités d'Atos chez ÉS . . . . .	4
1.3.1	Composition de l'équipe TMA JAVA . . . . .	4
<b>2</b>	<b>Environnements de travail</b>	<b>5</b>
2.1	Intégration à l'équipe . . . . .	5
2.2	Technologies utilisés . . . . .	6
2.2.1	Développement JAVA . . . . .	6
2.2.2	Intégration continue et qualité de code . . . . .	7
2.2.3	GWT / GXT . . . . .	7
2.2.4	Base de données et persistance des données . . . . .	8
<b>3</b>	<b>Projets</b>	<b>9</b>
3.1	Histo-versions . . . . .	9
3.1.1	Les environnements . . . . .	10
3.1.2	Fonctionnalités attendues . . . . .	10
3.1.3	Premiers travaux effectués . . . . .	10
3.1.4	Modifications du batch JAVA . . . . .	11
3.2	pApps . . . . .	12
3.2.1	Analyse de la demande . . . . .	13
3.2.2	Organisation du projet . . . . .	16
3.2.3	Situation actuelle du projet . . . . .	16
3.2.4	Analyse de l'état actuel du projet . . . . .	17
3.2.5	Bilan du projet . . . . .	18
3.3	Futur projets . . . . .	18
<b>4</b>	<b>Bilan</b>	<b>19</b>
	<b>Glossaire</b>	<b>20</b>
	<b>Acronymes</b>	<b>21</b>



---

# Table des figures

1.1	Acquisitions et filiales d'Atos[1]	2
1.2	Organisation du groupe ÉS[2]	3
3.1	Panneau "Versions" d'histo-versions	9
3.2	Interface proposée pour pApps	14
3.3	Interface actuelle de pApps	15

# Chapitre 1

## Contexte

Désactiver les todos une fois le document terminé

Lors des deux ans de Master Ingénierie des Logicielles et des Connaissances (ILC), les étudiants doivent effectuer une partie de leur scolarité en apprentissage au sein d'une société. Le temps de travail est partagé entre le temps en entreprise et le temps à l'université, durant la première année, à hauteur de 2 jours à l'université et 3 jours en entreprise.

J'ai eu la chance de pouvoir intégrer la société Atos en tant qu'apprenti en travaillant au sein de l'équipe Atos localisée dans les bureaux d'un de leur client : Électricité de Strasbourg (ÉS).

Dans ce mémoire, vous trouverez le travail effectué en entreprise durant la première année d'alternance. Dans un premier temps nous présenterons les groupe Atos et ÉS, avec une description des activités d'Atos chez ÉS. Suite à cela, une partie du mémoire sera consacrée à l'environnement de travail, puis aux différents projets abordés et futurs. Nous concluons par un bilan général concernant les impressions dans le cadre professionnel et personnel.

### 1.1 Le groupe Atos

Atos est un leader international de la transformation digitale<sup>1</sup>, du conseil et des services informatiques. Le groupe est implantée dans 73 pays, compte plus de 100 000 collaborateurs et réalise plus de 13 Milliards d'euros de chiffres d'affaires annuel.

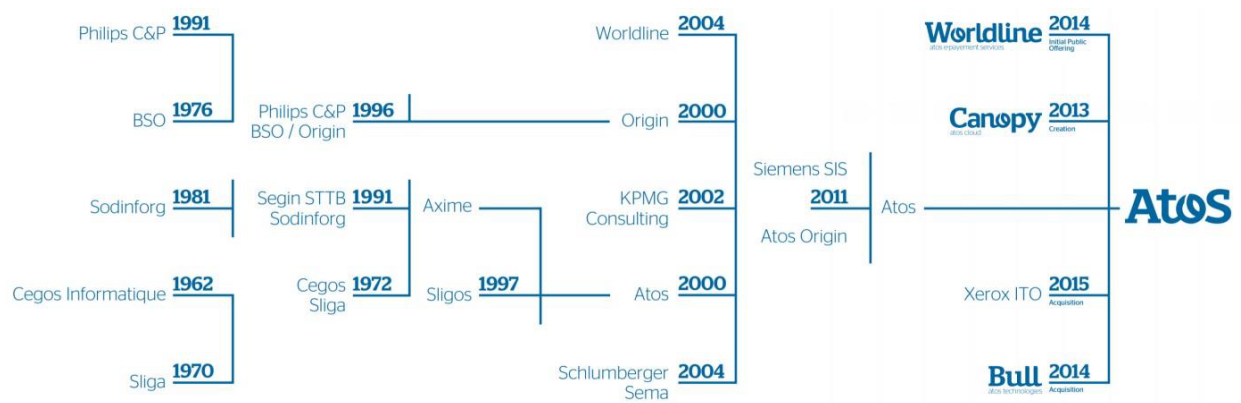


FIGURE 1.1 – Acquisitions et filiales d’Atos[1]

Atos étend ses activités par le biais d’acquisitions successives, parmi lesquelles on retrouve :

- L’infogérance et le conseil
- Le cloud
- Les services de e-paiement avec sa filiale Wordline
- La cybersécurité
- La défense et les systèmes critiques

Atos est également le partenaire de grands groupes tel que Microsoft, Cisco et Oracle afin de pouvoir fournir au client des solutions innovantes.

1. Désigne le processus au cours duquel une entreprise intègre les technologies disponibles pour ses activités.



## 1.2 Le groupe ÉS

Le groupe ÉS (Électricité de Strasbourg), énergéticien régional multi-énergies en Alsace depuis plus de 100 ans et filiale d'EDF, fait partie des grands acteurs du secteur énergétique français. Le groupe, organisé autour de trois grandes activités, est composé :

- d'un opérateur et gestionnaire de réseaux : Strasbourg Électricité Réseaux
- d'une filiale commerciale : ÉS Énergies Strasbourg
- de deux filiales spécialisées dans les services énergétiques et les énergies renouvelables : ÉS Services Énergétiques et ÉS Géothermie



FIGURE 1.2 – Organisation du groupe ÉS[2]

## 1.3 Les activités d'Atos chez ÉS

Dans le cadre du contrat de tierce maintenance applicative (TMA) entre Atos (anciennement Bull) et ÉS, plusieurs équipes d'Atos sont présentes au centre opérationnel de Mundolsheim du groupe ÉS. Elles sont organisées en deux lots :

- Le lot 1, dont je fais partie, est chargé de la maintenance applicative et évolutive JAVA, ainsi que des projets complémentaires.
- Le lot 2 est l'équipe décisionnelle.

### 1.3.1 Composition de l'équipe TMA JAVA

Le directeur de projet des équipes Atos est Jean-Pierre FONNE.

L'équipe TMA JAVA comprend 8 personnes :

- MIGNON Sabrina, référent projet
- RICHARD Alexandre, référent TMA
- QUERE Alexandre
- TERREAUX Mathias
- MONTESANTOS Alexis
- GRILL Stéphane
- MARLIER David
- TURIN Gary
- BROSSARD Florian (Apprenti)

On distingue deux activités au sein de cette équipe :

**La TMA :** Consiste en la réalisation de correctifs en cas d'anomalies déclarées, et/ou d'évolutions mineures suite à une demande ou à l'évolution d'un logiciel.

**Les projets :** Consiste en la création de nouvelles applications suivant le besoin du client.

Les utilisateurs déclarent les incidents via l'application PICTO. Ces demandes sont ensuite transférées au responsable de domaine fonctionnel qui valide l'incident et missionne la TMA pour le résoudre, suivant leur ordre de priorité.

## Chapitre 2

# Environnements de travail

### 2.1 Intégration à l'équipe

Durant la première semaine, j'ai suivi une formation concernant l'architecture ainsi que sur l'environnement de développement JAVA utilisé par l'équipe. Nous utilisons entre autre les framework GWT (Google Web ToolKit) et GXT.

Ma charge de travail est répartie majoritairement sur du temps de projet.

Durant le travail que j'ai effectué sur le projet pApps (Voir section 3.2 page 12), il m'a été demandé de créer une application ayant une interface différente des applications internes auxquelles les employés sont habitués. Pour cela, j'ai dû rechercher comment modifier l'aspect des composants graphiques du framework GXT. Ceci m'a donné l'occasion de faire une présentation à toutes l'équipe de la TMA JAVA et certains membres d'ÉS, permettant ainsi de partager l'information afin de gagner du temps à l'avenir.

## 2.2 Technologies utilisés

### 2.2.1 Développement JAVA

L'équipe TMA JAVA a à sa disposition plusieurs machines virtuelles. Chaque développeur dispose donc du même environnement de travail. Suivant la version de GXT utilisée, nous développons sur STS pour GXT 2 et sous Eclipse Mars pour GXT 3.

Le navigateur cible des applications développées par l'équipe est Internet Explorer.

Pour chaque projet nous utilisons un archetype Maven prédéfini (application web ou Webservice), chaque application est séparées en plusieurs couches :

- Couche DAO : Gère l'accès aux bases de données et les requêtes.
- Couche Service : Effectue les traitements métiers, fait appel aux Webservices
- Couche Server (application seulement) : Couche d'abstraction afin de faire appels au méthodes de la couche service.
- Couche RPC (application seulement) : Contient les interfaces synchrones et asynchrones des services de GWT, les DTO et les classes utilitaires. Cette couche permet de faire le lien entre le client et le serveur.
- Couche RS (Webservice seulement) : Point d'entrée des méthodes de la couche service. Fait appels aux méthodes de la couche service.
- Couche Webapp (application seulement) : Gère l'interface graphique, l'authentification et contient les ressources statiques de l'application (images, feuilles CSS, ...)
- Couche Interface (Webservice seulement) : Contient l'interface des services exposés par le Webservice.

Nous utilisons également Maven pour gérer les dépendences des projets.

### 2.2.2 Intégration continue et qualité de code

Afin d'améliorer la productivité et l'efficacité des développeurs, plusieurs outils sont mis à notre disposition :

- SVN : notre logiciel de gestion de versions
- Jenkins : logiciel open source d'intégration continue. Il permet de déployer rapidement et simplement les différents projets, mais peut également lancer des tâches définies, notamment afin de lancer un scan SonarQube via la même interface que pour les déploiements.
- SonarQube : SonarQube est un logiciel libre permettant de mesurer la qualité du code source à l'aide de nombreuses règles définies. SonarQube lance régulièrement une analyse des différents projets afin d'afficher la qualité générale du code, proposant ensuite des solutions aux bugs potentiels (pointeur null par exemple) et mauvaises pratiques et des améliorations concernant la lisibilité du code, notamment via les règles de codage (indentation, conventions de nommage, etc...).

Ces logiciels nous permettent d'une part de travailler en groupe, d'autre part d'améliorer la qualité des projets en général, les rendant plus facilement maintenable, mais aussi et surtout ils nous permettent d'être réactifs, c'est-à-dire de pouvoir traiter et corriger plus de problèmes avant que les utilisateurs n'y soient confrontés.

### 2.2.3 GWT / GXT

Pour développer l'Interface Homme-Machine (IHM) des applications web, nous utilisons les frameworks GWT et GXT. Ceci nous permet d'utiliser un langage objet de haut niveau (en l'occurrence JAVA) ainsi que les outils disponibles pour ce langage (debuggers, bibliothèques). En résumé, ces deux framework nous permettent de développer nos applications de la même manière qu'une application JAVA Swing<sup>1</sup>, à ceci près que l'on utilise les widgets de GWT et de GXT pour former l'interface. Ensuite le compilateur de GWT transforme le code JAVA en code Javascript tout en gérant la compression des ressources (images, texte, etc...), la gestion du cache et des appels AJAX.

---

1. Bibliothèque graphique JAVA

### 2.2.4 Base de données et persistance des données

Nous utilisons le Système de Gestion de Base de Données (SGBD) Oracle pour stocker les données persistantes des applications. La gestion de la sauvegarde, suppression, mise à jour, et même le filtrage des différents objets sont effectués à l'aide du framework Hibernate. Pour ce faire chaque objet persistant se voit lié à une entité et un DAO dont tous les champs sont mappés<sup>1</sup> à une colonne d'une table à l'aide d'Hibernate. Cette table servira donc à stocker chaque objet d'une même classe JAVA.

Nous avons également à notre disposition deux logiciels afin de faciliter la gestion et la modification des base de données :

**PowerAMC** : Logiciel permettant de modéliser graphiquement une base de données. Nous pouvons ainsi facilement créer des relations entre plusieurs tables, des contraintes, des vues, etc..., à l'image des logiciels utilisant le glisser-déposer tel que PowerPoint. PowerAMC génère ensuite un script SQL à exécuter sur la base de donnée afin d'y ajouter, modifier ou supprimer les différentes table. Parmi ses autres fonctions, les plus intéressantes sont la rétro-ingénierie, permettant de générer un schéma à partir de la base de donnée, et la génération d'un script SQL modifiant la base de données, permettant de ne modifier que ce qui est nécessaire, et de garder les données présentes dans la mesure du possible.

**SQL Developer** : Outil graphique permettant de parcourir et d'interroger de multiple bases de données. Il permet entre autre d'exécuter des requêtes SQL directement sur les bases de données, de rechercher et filtrer les différentes tables et les champs de ces tables, ainsi que leurs données. Son éditeur de requête est doté d'un vérificateur de syntaxe affichant les erreurs ainsi que des proposition de corrections pour celles-ci. Ce logiciels possèdent d'autres fonctions utiles tel que le rollback, permettant d'annuler des modifications en base, ou encore l'importation et exportation de données, utile pour utiliser un jeu de données spécifique pour les tests unitaires d'une application.

---

1. Signifie ici faire correspondre.

# Chapitre 3

## Projets

### 3.1 Histo-versions

L'application histo-versions a été le premier projet qui m'a été confié. Le but de cette application est d'afficher l'historique des déploiements des différentes applications pour chaque environnement. Afin d'éviter tout malentendu, sauf mention contraire, nous parlerons de création et de suppression d'application au sein d'histo-versions en parlant de l'ajout ou la suppression de lignes dans la table listant les applications, utilisée par histo-versions.

HISTOVERSIONS 1.8.1RC07

Changelogs

Deploiements

Versions

Administration

BROSSARD Florian

Versions

64 bits

64 bits DMZ

Rafrâichir

Environnement	Type	Domaine fonctionnel	Application	recette	pré-production	production	Gomix production
Linux	application	Autre	histoversions	<a href="#">1.8.1RC07</a>			
Linux	application	RH	orp	<a href="#">1.1.6</a>	<a href="#">1.0.5</a>		
Linux	application	Transverse	papps	<a href="#">1.1.0RC04</a>			
Linux	application	GRD	tigre	<a href="#">1.4.5</a>	<a href="#">1.4.6</a>		

FIGURE 3.1 – Panneau "Versions" d'histo-versions

### 3.1.1 Les environnements

Voici le cheminement du déploiement d'une nouvelle version d'une application :

1. Lorsque le développement d'une évolution ou d'un correctif est terminé, une nouvelle version est déployée sur l'environnement dit de **recette**. Notre équipe a pleinement accès à cet environnement afin de déployer une nouvelle version.
2. Suite à la confirmation du chef de projet et du responsable de domaine fonctionnel l'application peut, suite à des tests, être déployée sur l'environnement dit de **production**. C'est l'environnement qu'utilisent les utilisateurs.
3. Éventuellement, une application peut être placée dans un environnement dit de **pré-production**, avant d'être placée sur l'environnement de **production**, si cela est nécessaire.

### 3.1.2 Fonctionnalités attendues

Voici une liste des fonctionnalités de l'application :

- Afficher la liste des applications avec le numéro de la version déployée pour chaque environnement (**recette**, **pré-production**, **production** et leur équivalent **DMZ**)
- Afficher la liste des changelogs<sup>1</sup> pour chaque application
- Afficher la liste de tous les déploiements de chaque application dans chaque environnement
- Pouvoir effectuer plusieurs opérations dont la sélection des applications à afficher, l'ajout/suppression de ces applications et des changelogs dans histoversions et le filtrage des applications (par nom, version, environnements, etc...)

### 3.1.3 Premiers travaux effectués

Durant les premiers mois de mon alternance, j'ai débuté une amélioration de la qualité de code, évaluée par le logiciel SonarQube, me permettant ainsi de me familiariser avec plusieurs outils, tel que SVN et Jenkins, ainsi qu'avec le framework GXT 2. Dans le même temps, cela m'a permis de comprendre le fonctionnement de ce projet, facilitant grandement les futures modifications.

---

1. Un changelog, ou journal des modifications, est une liste des modifications[7].



Outre les modifications de l'interface et les corrections de bugs, la première tâche que j'ai effectuée a été de rendre périodique l'ajout automatique des déploiements dans la base de donnée. Cet ajout est réalisé à l'aide d'un batch<sup>1</sup> JAVA. Pour ce faire j'ai dû modifier ce batch afin de pouvoir l'utiliser en tant que tâche, au lieu d'être utilisé en tant que webservice, accessible depuis un lien web.

La première solution afin de pouvoir gérer la planification a été d'utiliser une fonctionnalité de Spring permettant d'instancier un objet, et de lancer une des méthodes de cet objet à un intervalle donnée. Afin de permettre la modification de cet intervalle, plusieurs variables ont été ajoutées dans le fichier de configuration d'Histoverions.

### 3.1.4 Modifications du batch JAVA

La premier changement majeure a été de déplacer l'écran d'administration dans une autre application sur laquelle j'ai travaillé : pApps (Voir section 3.2.1 page 13).

D'autre part, le batch JAVA qui était en place a nécessité plusieurs modifications suite à plusieurs demandes de fonctionnalités.

La première fonctionnalité à ajoutée était de créer une servlet, appelée depuis une URL, renvoyant la date du dernier déploiement enregistré pour une application et un environnement donné en paramètre. Avec cela, la servlet devait pouvoir mettre à jour les données de la tables concernant l'application et l'environnement spécifiés avant de renvoyer le résultat et, renvoyer un code HTTP et un message d'erreur adéquat dans le cas ou l'application ou l'environnement ne seraient pas trouvés. Cette fonctionnalité a nécessité de modifier le batch de tel sorte que l'on puisse spécifier sur quel environnement et quel applications effectuer le traitement, dans le but de ne mettre à jour que l'application dans l'environnement concerné, et donc de gagner en temps de réponse et de réduire le coup de l'appel à la servlet.

Suite au développement de cette serlvet, une nouvelle spécification a été formulée : supprimer les applications qui ne sont trouvées sur aucun des serveur connus en base. Cette spécification répond au cas où une application serait supprimée du serveur, auquel cas il est nécessaire de faire suivre l'information aux différentes tables d'Histoverions.

---

1. En informatique, un traitement par lots (batch processing en anglais) est un enchaînement automatique d'une suite de commandes (processus) sur un ordinateur sans intervention d'un opérateur[10]

## CHAPITRE 3. PROJETS

---

Après cette modification, la solution ayant été choisie concernant l'appel périodique au batch JAVA à fait apparaitre une erreur de conception :

- Les base de données sont sauvegardées chaque semaine le week-end, entrainant l'impossibilité de les interroger durant cette opération
- Le batch se lance toute les heures chaque jours
- Ce même batch supprime les applications qui ne sont pas trouvées, or, la base ne répond pas pendant sa sauvegarde
- De plus, dans le cas d'un appel à la servlet décrite plus haut, si le serveur n'existe pas, ou plus, mais que l'application existe, cette dernière seras tout de même supprimée de la base

Afin de pallier à cette erreur de conception, une autre fonctionnalité à été adopté à la place de l'utilisation d'un intervalle : l'utilisation d'expression CRON. Ce type d'expression permet de spécifier non seulement un intervalle à la seconde près, mais aussi les jours de la semaine, ou du mois auquel on souhaite lancer une tâche.

Pour finir, une légère modification à permis d'éviter de supprimer les applications dans le cas ou le batch est lancé à partir de la servlet.

### 3.2 pApps

Dans le cadre de mon apprentissage et de ma formation, le projet pApps m'a été confié.

Le projet pApps (pour portail applications) est né d'un besoin de proposer aux utilisateurs un portail d'accès centralisé aux applications. Ce projet remplacera donc l'actuel accès aux applications par leurs adresses.

### 3.2.1 Analyse de la demande

#### Fonctionnalités

Voici la liste des fonctionnalités spécifiées dans la documentation fonctionnelle :

- Afficher une liste des applications
- N'afficher que les applications auxquelles l'utilisateur a accès
- Afficher, si possible, une icône représentant l'application
- Afficher l'environnement des applications
- Déplacer la gestion des applications d'Histoversions vers pApps
- Modifier les styles de GXT pour créer une interface plus convivial et agréable, différentes des autres

Afin de répondre à ces attentes, nous pouvons utiliser un webservice déjà existant, nommé LDAP, permettant de lister les applications auxquelles l'utilisateur a accès. Les icônes des applications peuvent être recherchées sur le serveur, en supposant que l'on respecte un certain formatage du nom des fichiers. Dans le cas où une image n'est pas trouvée, on peut la remplacer par une image par défaut, le logo du groupe ÉS par exemple.

#### Gestion des applications

Cette fonctionnalité montre la proximité entre les applications pApps et Histoversions. En effet, l'application Histoversions utilise plusieurs table contenant les informations sur les applications, leurs versions et leurs déploiements. La table contenant les applications étant alimentée par un batch Informatica<sup>1</sup>, nous pouvons l'utiliser pour le projet pApps, permettant ainsi d'éviter la duplication de données. Il est néanmoins nécessaire de pouvoir saisir et modifier manuellement les applications via un écran car il n'est pas possible de lier directement les applications présentes dans cette table, aux applications récupéré par le webservice LDAP.

Cette fonctionnalité a nécessité de déplacer la gestion des applications d'Histoversions dans pApps. La difficulté de ce travail a été de porter une grille du framework GXT 2 vers GXT 3. Ce problème ayant déjà été soulevé dans d'autre application, deux solutions ont été envisagées : le portage manuelle et l'automatisation.

---

1. Technologie utilisée par le lot 1 (Voir section 1.3 page 4) permettant dans ce cas de récupérer des informations à partir des fichiers de projet Maven

## CHAPITRE 3. PROJETS

Dans le cadre de l'évolution de plusieurs autres applications, des recherches et essais avaient déjà été effectués par certains de mes collègues afin d'automatiser le processus de migration d'une version à l'autre de GXT. Ceux-ci n'ayant pas totalement abouti, notamment à cause des différences majeures entre ces deux versions, il a été nécessaire de faire la migration manuellement, posant quelques difficultés étant donné qu'un même objet ou méthode peut avoir un comportement différent, voire même ne pas avoir d'équivalent.

### Interface

Concernant l'affichage des environnements des applications, nous avons imaginé deux solutions :

- Plusieurs onglets regroupant chacun les applications d'un seul environnement, avec le nom de l'environnement en guise de label pour l'onglet
- Si l'on considère que chaque application est représentée par une tuile, on met une couleur de fond et un motif à cette tuile pour indiquer l'environnement.

On peut également combiner ces deux solutions en mettant par exemple une légende à un endroit visible de la fenêtre, indiquant pour chaque couleur le nom de l'environnement. Chaque élément de cette légende étant cliquable, on peut appliquer un filtrage sur les environnements.

C'est d'ailleurs cette solution que j'ai proposée en premier lieu :

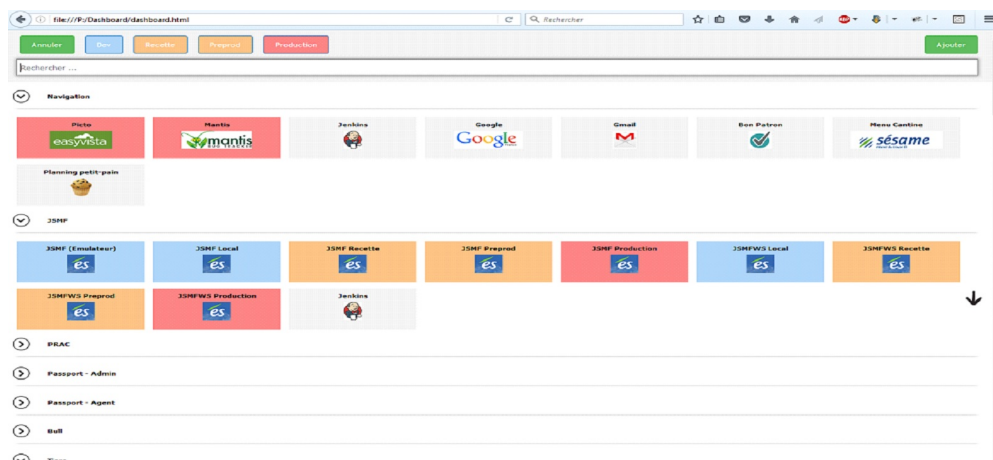


FIGURE 3.2 – Interface proposée pour pApps

## CHAPITRE 3. PROJETS

Cette première solution a été partiellement retenue :

- Un seul environnement est affiché à l'écran
- Sélection de l'environnement affiché à l'aide de boutons (fonctionnement similaire à des onglets)
- Applications affichées sous forme de tuiles avec une icône par défaut
- Choix de couleurs neutre (blanc largement majoritaire)
- Système d'accordéons permettant de classer les applications par domaine fonctionnel
- Barre de recherche

Voici un aperçu de l'interface actuelle :

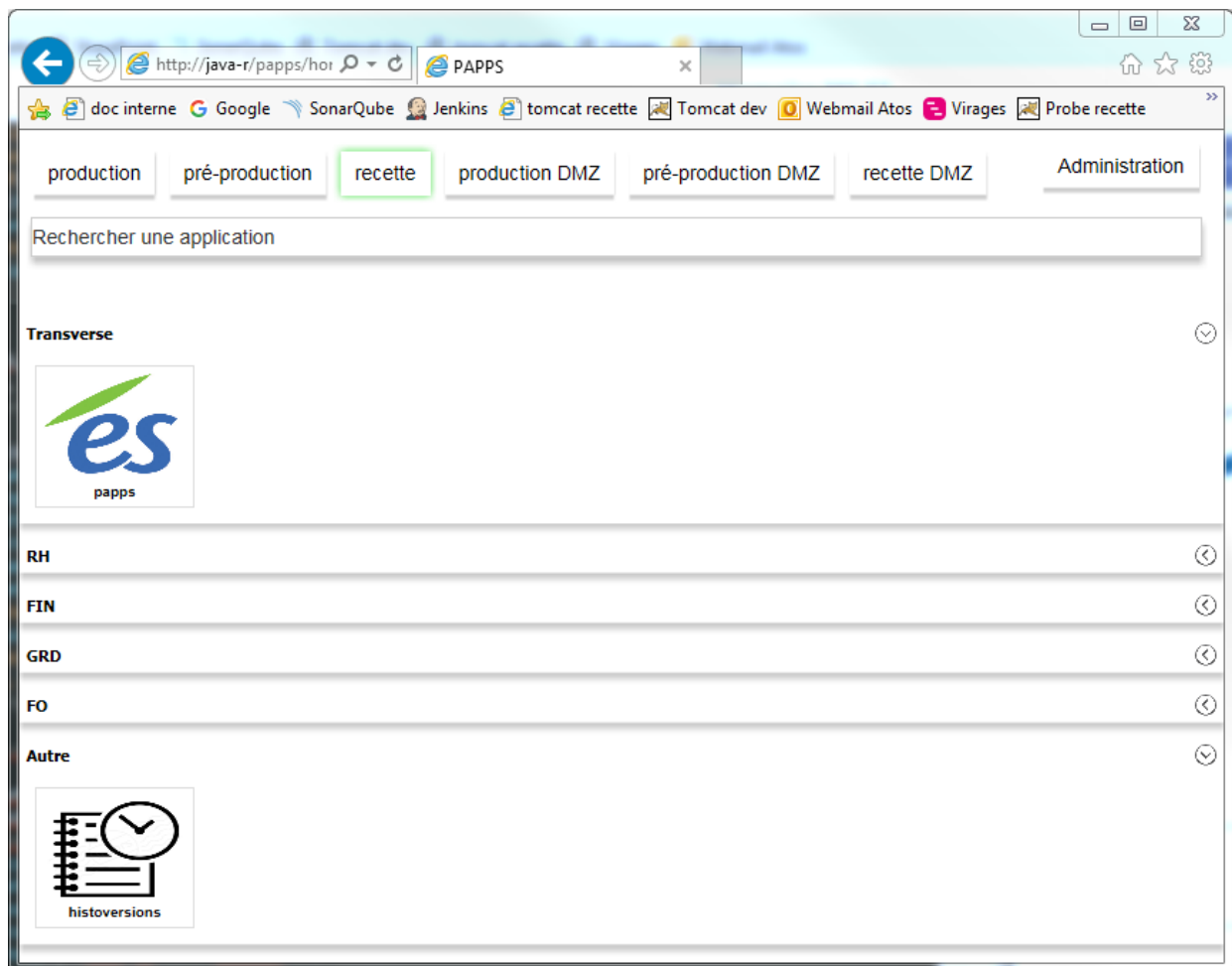


FIGURE 3.3 – Interface actuelle de pApps

### 3.2.2 Organisation du projet

Afin de prendre en compte au plus tôt les nouvelles demandes, les éventuels problèmes de conception, ainsi que d'assurer le suivi du développement de l'application, nous avons adopté un modèle orienté méthode agile<sup>1</sup> au lieu du modèle du cycle en V<sup>2</sup> habituel. Ceci implique l'organisation de réunion régulière avec le responsable de projet du groupe ÉS, afin de discuter et valider chaque partie de l'application une par une.

Cette méthode de travail permet notamment de gagner du temps, en évitant des erreurs de conception coûteuses en terme de temps et de modification, ou encore d'ajouter ou préciser certaines demandes et fonctionnalités. Tout cela permet de planifier plus facilement le développement d'une application.

### 3.2.3 Situation actuelle du projet

Une des tâche majeure a été de récupérer les droits applicatifs de l'utilisateur actuel afin de n'afficher que les applications auquel il est autorisé d'accéder. Pour ce faire, un webservice dédié existant à été utilisé.

Suite à la création du portail, et au portage de l'administration des applications, une fonctionnalité nécessaire au bon fonctionnement est apparue : le mode de compatibilité.

Le mode de compatibilité est en fait un raccourcis pour indiquer qu'une application doit être accéder depuis une URL différente suivant la version de GXT utilisée. Si l'application utilise GXT 2, aucun changement n'est nécessaire. En revanche, si elle utilise GXT 3, on doit modifier l'URL d'accès.

Prenons un exemple concret :

- L'application Histoverions utilise GXT 2
- L'application pApps utilise GXT 3
- Toute deux sont sur un serveur de **recette**, dont l'URL d'accès pour GXT 2 est `vm-tomcat64-r`
- Le serveur de recette utilisé habituellement pour les application GXT 2 force un mode de compatibilité concernant la version de JavaScript utilisée, contrairement à GXT 3.

---

1. Les méthodes agiles se veulent plus pragmatiques que les méthodes traditionnelles, impliquent au maximum le demandeur (client). Elles reposent sur un cycle de développement itératif, incrémental et adaptatif[12].

2. Le modèle du cycle en V (en comparaison avec les méthodes dites agiles) est un modèle conceptuel de gestion de projet imaginé[11].

## CHAPITRE 3. PROJETS

---

Connaissant ces différents points, une modification s'est avérée nécessaire. Cependant, étant donnée la nature du problème, et après analyse de pApps et d'Histoverions qui partagent les mêmes tables, seules de légères modifications ont été suffisantes pour le résoudre :

- Ajout d'une colonne indiquant si l'application nécessite le mode de compatibilité, dans la base de données et dans l'interface
- Ajout d'une colonne indiquant le lien utilisé par le mode de compatibilité dans la table contenant les environnements
- Modification des deux vues utilisées pour lister les environnements et les applications pour remplacer le lien du serveur, si le mode de compatibilité est utilisé

Que l'URL soit celui du mode de compatibilité ou non, l'accès aux applications reste le même, ce qui explique le peu de modifications nécessaires dans le code source.

Ajouter un paragraphe sur le problème de l'ajout d'environnement et de la notion de type d'environnement ?

### 3.2.4 Analyse de l'état actuel du projet

Concernant pApps, plusieurs choix sont encore en cours de discussion, notamment concernant les applications qui n'ont pas d'icônes.

Précisons que peu d'applications au sein de l'ÉS ont une icône les représentant. Imaginons que pApps intègre toutes les applications, ce qui à terme pourrait être le cas, nous nous retrouverions devant une page avec, par exemple, une cinquantaine d'applications ayant l'icône par défaut. Nous pouvons donc nous poser la question suivante : Est-ce un choix pertinent d'utiliser systématiquement la même icône ?

Le problème est qu'il est plus difficile de trouver visuellement l'application souhaitée, si la seule différence entre toutes ces applications est le nom de l'application situé en dessous du logo par défaut. Ceci remet en cause la pertinence du choix de l'utilisation d'une icône par défaut.

À l'heure actuelle, une proposition est de remplacer le logo par défaut par le nom de l'application, écrit dans une police plus grande.

### 3.2.5 Bilan du projet

Ce projet à été une très bonne introduction aux différentes technologies utilisées dans cette entreprise, mais également à la création d'un projet.

En effet, cette expérience m'a permis de voir qu'il est indispensable de bien comprendre les différentes problématique et de prévoir quelle seront les répercussion d'un éventuelle changement, ceci afin d'éviter d'être plus tard confronter à un changement radical de l'implémentation d'une application.

Il m'a également d'écrire une documentation technique, destinée aux développeurs, afin de détaillé l'implémentation et l'organisation des différentes partie de l'application, en complément des commentaires présent dans le code source.

Je me suis rendu compte qu'il était plus facile de prendre du recul quand au choix de conception d'une application en écrivant sa documentation. En effet, s'il est difficile d'expliquer une partie qui ne devrait pourtant pas être complexe, nous pouvons nous demander si ce n'est pas à cause d'un mauvais choix concernant l'utilisation d'un objet plutôt qu'un autre ou encore l'écriture d'un code trop compliqué et, ou, mal pensé.

## 3.3 Futur projets

Pour le moment, en plus du projet pApps, plusieurs transferts de compétences sont prévus. Cela consistera entre autres à comprendre la partie métier visée par ces applications ainsi que leur conception. Je serai par la suite amené à gérer les différents retours d'anomalies et évolutions demandées liés à ces applications. Certain transfert ont déjà été effectuées, par exemple pour l'application Rachel, un portail permettant d'envoyer des demandes de raccordement électrique à ÉS en passant par plusieurs formulaire, et donnant accès au suivis du dossier.

Mis à part cela, et étant donné la nature des activité d'Atos pour le groupe ÉS, il est difficile de prévoir la création de projet sur le long terme.



# Chapitre 4

## Bilan

Cette première année d'expérience m'a permis de découvrir de nouvelles manières de concevoir des applications. J'ai également appris à utiliser l'intégration continue, ce qui est un véritable avantage par rapport à mes précédents projets. Je m'aperçois que tout cela facilite énormément le développement d'un projet en évitant les problèmes de dernières minute et en donnant une manière de respecter certaines règles quant à l'écriture du code.

Étant au sein d'une équipe compétente et agréable, je trouve toujours réponse à mes questions et j'ai toujours plaisir à travailler. De plus, le fait de travailler avec deux anciens apprentis du master ILC m'aide beaucoup.

Tout cela à été très formateur et m'a permis d'en apprendre beaucoup concernant l'informatique et le développement logiciel.

Je me suis rendu compte que les métiers de l'informatiques ne se limite pas qu'à la production de code source ou à la conception de solution logicielle répondant à une demande. L'un des aspect essentiel de ces métiers reste la compréhension d'un métier qui n'est pas le notre, afin de comprendre et prévoir les problématiques, ainsi que l'apprentissage d'un vocabulaire pour comprendre et être compris.

Le dernier aspect qui plaît est le partage de l'information. C'est à dire expliquer quelque chose sur lequel nous avons passés certaines fois plusieurs heures pour le comprendre, transmettre ses connaissances, les améliorer, et en acquérir de nouvelles.

# Glossaire

**Eclipse** Eclipse est un IDE libre et extensible pour le langage JAVA. 6, 21

**GWT** Framework permettant de créer et maintenir des applications web dynamiques mettant en œuvre JavaScript, en utilisant le langage et les outils Java. 5–7

**GXT** Framework permettant de créer des applications web riches en utilisant GWT. 5–7, 10

**Hibernate** Hibernate est un framework open source gérant la persistance des objets en base de données relationnelle[9]. 8

**Machine virtuelle** Une machine virtuelle est une simulation d'un appareil informatique créée par un logiciel. 6

**Maven** Outil de gestion et d'automatisation de production des projets logiciels Java. 6

**REST** REST (representational state transfer) est un style d'architecture logicielle, notamment utilisé dans les applications web[5]. 5

**Spring** Le framework Spring permet de construire et de définir l'infrastructure d'une application JAVA. 21

**SVN** Subversion (en abrégé svn) est un logiciel de gestion de versions[13]. 7, 10

**Webservice** Logiciel sans interface, accessible depuis une URL, permettant la communication et l'échange de données entre applications et systèmes hétérogènes. 6

# Acronymes

**AJAX** L'architecture AJAX, acronyme d'Asynchronous JavaScript and XML, permet de créer des applications web dynamique et interactive[6]. Cela permet entre autre de modifier une partie de la page sans devoir la recharger entièrement. 7

**DAO** Un DAO (Data Access Object) est un patron de conception permettant d'ajouter une couche d'abstraction concernant le stockage de données. 8

**DMZ** En informatique, une zone démilitarisée (ou DMZ, de l'anglais demilitarized zone) est un sous-réseau séparé et isolé du réseau local par un pare-feu. Ce sous-réseau contient les machines étant susceptibles d'être accédées depuis Internet.[3]. 10

**DTO** Un objet de transfert de données (Data Transfer Object en anglais) est un patron de conception utilisé dans les architectures logicielles objet. Il permet de simplifier la transmission de données entre les différentes couches d'une application. 6

**IDE** . Un IDE (pour Integrated Development Environment) rassemble un ensemble d'outils permettant d'augmenter la productivité des développeurs en un seul logiciel. 20

**IHM** Interface Homme-Machine. 7

**ILC** Ingénierie des Logicielles et des Connaissances. 1, 24

**Rétro-ingénierie** La rétro-ingénierie, ou ingénierie inverse, est l'activité qui consiste à étudier un objet pour en déterminer le fonctionnement interne ou la méthode de fabrication.[4]. 8

**SGBD** En informatique, un système de gestion de base de données (abr. SGBD) est un logiciel système destiné à stocker et à partager des informations dans une base de données[8]. 7

**STS** Sprinsource Tool Suite est une distribution du logiciel Eclipse facilitant le développement d'application Web JAVA utilisant le framework Spring. 6

## Acronymes

---

**TMA** La Tierce Maintenance Applicative consiste à externaliser la maintenance de toutes ou d'une partie des applications d'une entreprise auprès d'un prestataire. 4, 6

**ÉS** Électricité de Strasbourg. 1, 3–5, 12, 14, 18, 24

# Bibliographie

- [1] *Atos one of world's fastest growing systems integrators/service providers*. 20 jan. 2016. URL : <https://twitter.com/alanlepo/status/689806517520965633>.
- [2] *Organisation du groupe*. URL : <http://www.es-groupe.fr/content/download/1224/8476/version/1/file/organigramme.pdf> (visité le 29/12/2017).
- [3] WIKIPÉDIA. *Zone démilitarisée (informatique)*. 15 mar. 2017. URL : [https://fr.wikipedia.org/wiki/Zone\\_d%C3%A9militaris%C3%A9\\_\(informatique\)](https://fr.wikipedia.org/wiki/Zone_d%C3%A9militaris%C3%A9_(informatique)) (visité le 29/12/2017).
- [4] WIKIPÉDIA. *Rétro-ingénierie*. 25 mar. 2018. URL : <https://fr.wikipedia.org/wiki/R%C3%A9tro-ing%C3%A9nierie> (visité le 23/04/2018).
- [5] WIKIPÉDIA. *Representational state transfer*. 28 déc. 2017. URL : [https://fr.wikipedia.org/wiki/Representational\\_state\\_transfer](https://fr.wikipedia.org/wiki/Representational_state_transfer) (visité le 29/12/2017).
- [6] WIKIPÉDIA. *Ajax (informatique)*. 14 déc. 2017. URL : [https://fr.wikipedia.org/wiki/Ajax\\_\(informatique\)](https://fr.wikipedia.org/wiki/Ajax_(informatique)) (visité le 29/12/2017).
- [7] WIKIPÉDIA. *Changelog*. 26 déc. 2016. URL : <https://fr.wikipedia.org/wiki/Changelog> (visité le 29/12/2017).
- [8] WIKIPÉDIA. *Système de gestion de base de données*. 11 nov. 2017. URL : [https://fr.wikipedia.org/wiki/Syst%C3%A8me\\_de\\_gestion\\_de\\_base\\_de\\_donn%C3%A9es](https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_gestion_de_base_de_donn%C3%A9es) (visité le 29/12/2017).
- [9] WIKIPÉDIA. *Hibernate*. 19 déc. 2017. URL : <https://fr.wikipedia.org/wiki/Hibernate> (visité le 29/12/2017).
- [10] WIKIPÉDIA. *Traitement par lots*. 1<sup>er</sup> déc. 2017. URL : [https://fr.wikipedia.org/wiki/Traitement\\_par\\_lots](https://fr.wikipedia.org/wiki/Traitement_par_lots) (visité le 29/12/2017).
- [11] WIKIPÉDIA. *Cycle en V*. 17 sept. 2017. URL : [https://fr.wikipedia.org/wiki/Cycle\\_en\\_V](https://fr.wikipedia.org/wiki/Cycle_en_V) (visité le 30/12/2017).
- [12] WIKIPÉDIA. *Méthode agile*. 5 déc. 2017. URL : [https://fr.wikipedia.org/wiki/M%C3%A9thode\\_agile](https://fr.wikipedia.org/wiki/M%C3%A9thode_agile) (visité le 30/12/2017).
- [13] WIKIPÉDIA. *Apache Subversion*. 31 déc. 2016. URL : [https://fr.wikipedia.org/wiki/Apache\\_Subversion](https://fr.wikipedia.org/wiki/Apache_Subversion) (visité le 03/01/2018).

# Résumé

Dans le cadre de mon Master informatique Ingénierie des Logicielles et des Connaissances, j'ai effectué mon apprentissage au sein du groupe Atos.

Ce mémoire présente le groupe Atos et ÉS, ainsi que la première année de mon travail pour ÉS dans l'équipe Atos.

**Mots-clés :** Atos, ÉS, Application Web, GXT, GWT, REST, JAVA

---

# Abstract

Through my Master degree in computer science, I worked within the Atos group under an apprenticeship contract.

This report presents the Atos and ÉS groups, as well as the first year of my work for the ÉS group within the Atos team.

**Key-words :** Atos, ÉS, Webapp, GXT, GWT, REST, JAVA