Projet Industriel 2016-2017 – Souscription d’assurance par tablette tactile

Université d’Orléans – Atos

Dossier d’architecture technique

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Auteur | Libellé |
| 1.0 | 30/12/2016 | Vincent Montigny | Première version simple |
| 1.1 | 13/01/2017 | Vincent Montigny | Plus de précision |

# Introduction

Ce document a pour but de présenter l’architecture applicative de la solution proposée. Cette solution comporte, comme présenté dans les différentes SFD, deux applications distinctes, une pour client mobile et une pour client siège, les deux pouvant synchroniser leurs informations. Ce document présentera l’architecture applicative globale, puis détaillera ses composants.

Il présentera ensuite les outils utilisés pour la réalisation de la solution.

# Architecture applicative

## Architecture globale

Figure : schéma d’architecture globale

### Validité

Cette architecture est présentée comme celle valide pendant la durée du POC. A l’avenir et si elle était intégrée dans un SI existant, les bases de données, le type de serveur ou d’autres composantes pourraient être sujet à modification.

### Authentification/Utilisateurs

#### Gestion des profils

La gestion des utilisateurs des deux applications sera commune. A travers un annuaire LDAP qu’aucun des utilisateurs ne pourra remplir, stocké sur le serveur d’application. Le choix d’un annuaire et pas d’une base de données est habituelle dans les entreprises. Il est plus souvent consulté qu’il n’est modifié, plus facile d’accès depuis n’importe quel type de client, et les conventions LDAP sont parfaitement adaptés à la gestion des utilisateurs (qu’on peut diviser en groupes par exemple).

Les deux types de client auront à s’identifier, mais pas au même moment.

Le client mobile agira en mode déconnecté, et son identité ne sera vérifiée qu’à la synchronisation. De cette manière, il pourra prendre toutes les informations nécessaires même sans couverture réseau, avant de les envoyer, cette fois en mode connecté.

Du côté du client siège, l’identification est nécessaire dès l’entrée dans l’application, pour accéder à ses dossiers.

Nous stockerons pour l’instant 3 informations utilisateurs :

* UID (son identifiant)
* Nom
* Prénom

## Stockage des données

Pour le client mobile, chaque information que rentrera le client pourra se faire en mode déconnecté, comme nous l’avons spécifié. Il nous est donc imposé d’avoir un stockage des données interne au support mobile. Que ce soit sur tablette ou téléphone, les données devront donc être mises dans un dossier spécifique du support. Un dossier créé et géré par l’application semble être une bonne solution.

Pour le client siège, il accède directement à la base de données « réelle ». En effet, une fois les informations synchronisées, elles sont stockées en base. Pendant le POC, cette base est celle fournit avec le serveur, elle pourra être inséré dans un autre SI à l’avenir. Cette base de données regroupe seulement les informations textuelles, pas les photos, les métrages ou les éventuels documents associés.

Pour ces données spécifiques, on utilisera un autre support qui permettra la consultation de ces données, type GED.

Pour lister les utilisateurs, comme précisé plus haut, ils seront référencés dans un annuaire LDAP stocké sur le serveur d’applications. Egalement précisé plus haut, ce serveur est sujet à changement dans le cadre d’une intégration future.

Ci-après, le schéma d’architecture avec le cas d’un dossier et la procédure concernant les identifications/stockage de données

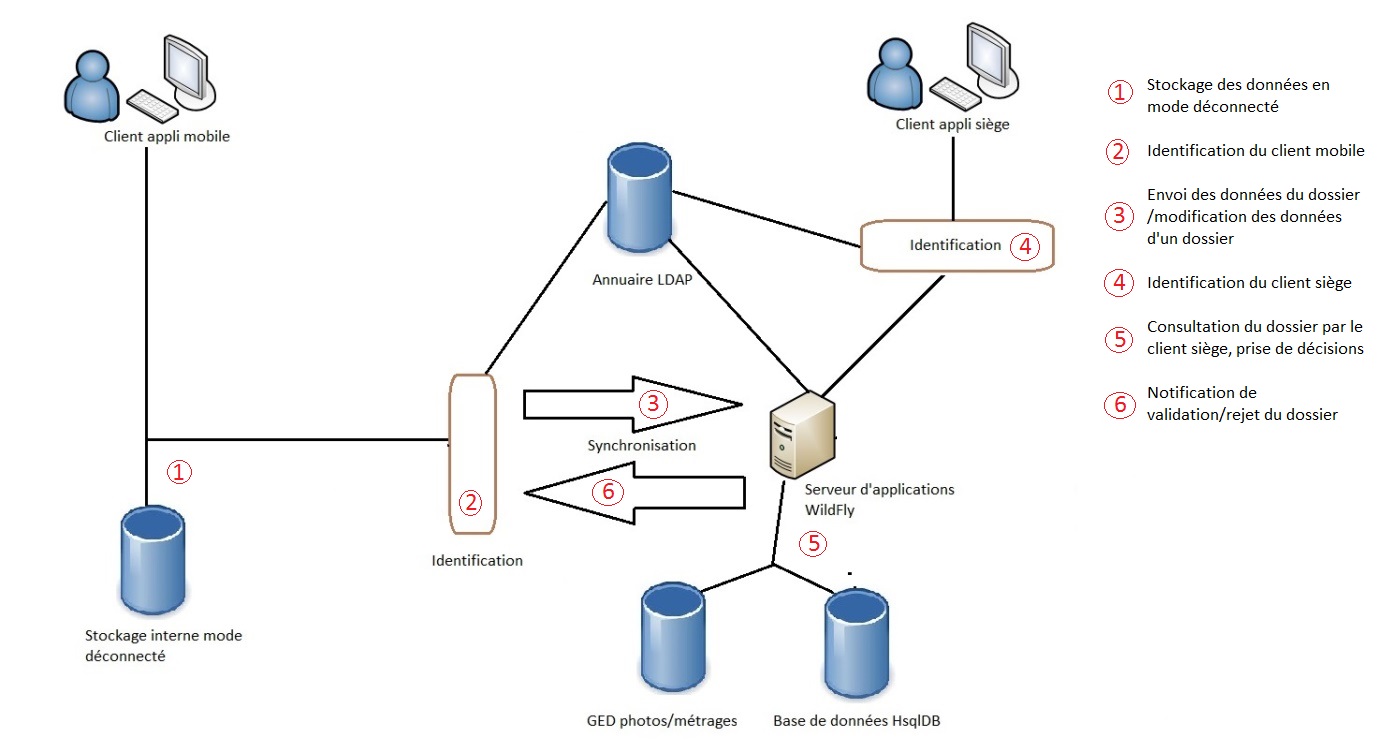


Figure : Procédure identifications/données

## Modèle de données

Le modèle de données n’est pas encore totalement spécifié, il fera l’objet d’un document à part de conception,

## Couche IHM

Les différents écrans des deux applications sont spécifiés dans leurs SFD respectives.

# Liste des composants logiciels

Nous n’utiliserons que des outils OpenSource pendant la durée du POC.

## Environnement de développement

Les applications seront réalisées en Java. L’application mobile sera donc une application Android.

Pour l’application Siège, nous utiliserons l’IDE Eclipse Neon, la version la plus récente à l’heure actuelle, gratuite et performante.

L’organisation du projet se fera à l’aide de Maven, pour la gestion claire et complète du projet.

Nous utiliserons des EJB comme modèle de données, avec Hibernate comme ORM. Cet outil a été utilisé dans un projet d’étude récemment, et il remplit pleinement les demandes du projet. Ci-après, sa position entre le modèle java et la couche De données en base.

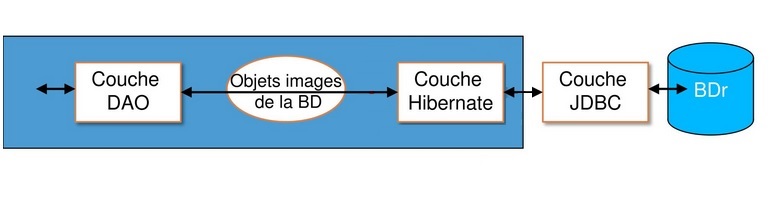


Figure : place de hibernate dans l'ORM

Nous utiliserons également AngularJS 1.6.1 pour la partie Web. Cet outil est très récent et très performant. Bien que nous n’ayons jamais travaillé dessus, de nombreuses aides sont disponibles et permettront de réaliser une solution optimale.

Pour l’application Mobile, nous utiliserons Android Studio comme environnement de développement, version 2.2.2. C’est une solution OpenSource efficace et très utilisée.

## Serveur d’application

Comme on peut le voir sur la figure 1, nous utiliserons WildFly 10.1.0 comme serveur d’application. Il fournit également une base de données que nous utiliserons aussi durant le POC. Nous avons déjà utilisé cet outil dans un projet récent, et il apparaît répondre parfaitement à nos besoins.

## Système de Versionnage

Nous utilisons Git comme système de versionnage, avec un dépôt situé sur le site de GitHub.

## Récapitulatif

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Application Client** | | | **Application Siège** | | |
| Champ d’application | | Nom | Version | | Nom | Version |
| Environnement de Développement | | Android Studio | 2.2.2 | | Eclipse | Neon |
| Gestion du modèle | |  |  | | EJB |  |
| Persistance | |  |  | | Hibernate |  |
| Serveur d’application | |  |  | | WildFly | 10.1.0 |
| Base de données | | Dossier interne |  | | HsqlDB (fournit avec Wildfly) |  |
| Framework Web | |  |  | | AngularJS | 1.6.1 |
| Versionnage | |  |  | | Git | 2.11.0 |