

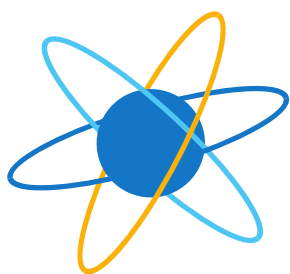
# SYSTÈME DE GESTION DES TRANSPORTS

---

Version 1.0

*6 mars 2014*

## Dossier d'Architecture Technique



Faculté  
des Sciences  
Aix\*Marseille Université





## Page d'informations

### CONTACTS

Prénom(s) & Nom	Adresse e-mail
Roland AGOPIAN	roland.agopian@univ-amu.fr
Ibrahima Sory BALDE	ibbaldes@yahoo.fr
Mehdi-Jonathan CADON	mehdi-jonathan@hotmail.fr
Lionel GAIROARD	lionel.gairoard@gmail.com
Anthony JULIEN	anthonyjulien2@gmail.com
Adrien LERICOLAIS	adrienmgs@gmail.com
Rémi MEZELLE	remi.mezelle@gmail.com
Ravi PACHY	pachy.ravi@gmail.com
Bien Aimé SUANGA WETO	maitreswing@gmail.com
Ahoua Khady TOURE	takamor91@yahoo.fr

### VERSIONS

Version	Date	Auteur(s)	Modification(s)
2.0	2014-03-03	Mehdi-Jonathan CADON	Création à partir de l'ancienne version
2.1	2014-03-04	Ravi PACHY	Mise à jour en fonction de la réunion interne du 2014-03-03
2.2	2014-03-06	Adrien LERICOLAIS	Mise à jour de la partie données

### DIFFUSION

Version	Date	Approbateur(s)
-	-	

### VALIDATION

Version	Date	Responsable(s)
2.1	-	-



## Table des matières

<b>1</b>	<b>Présentation du projet</b>	<b>4</b>
1.1	Introduction	4
1.2	le calendrier du projet	4
1.2.1	la phase de conception du projet	4
1.2.2	La phase de réalisation du projet	4
1.3	L'objectif du projet	4
1.4	Les enjeux du projet	4
1.5	Les exigences du projet	5
1.5.1	La localisation	5
1.5.2	L'interface utilisateur	5
1.5.3	Les cas d'utilisations	5
1.5.4	Les contraintes	5
1.6	Le planning	5
<b>2</b>	<b>Architecture fonctionnelle</b>	<b>6</b>
<b>3</b>	<b>Architecture technique</b>	<b>7</b>
3.1	Données	7
3.1.1	Problématique	7
3.1.2	Le stockage	7
3.1.3	Le transfert	7
3.1.4	L'import/export de données	8
3.2	Sauvegarde	8
3.2.1	Problématique	8
3.2.2	Cluster	8
3.2.3	Reprise sur panne	9
3.2.4	Sauvegarde en dure	9
3.3	Synchronisation	9
3.3.1	Introduction	9
3.3.2	Gestion des priorités & des profils de synchronisation	11
3.3.3	Mode connecté	11
3.3.4	Mode hors-ligne	11
3.3.5	Gestion des conflits	12
<b>4</b>	<b>Exploitation et préparation</b>	<b>13</b>



# 1 — Présentation du projet

## 1.1 INTRODUCTION

La présentation du projet s'articule autour de plusieurs points qui sont :

- le calendrier du projet ;
- son objectif ;
- ses enjeux ;
- ses exigences ;
- le planning ;

## 1.2 LE CALENDRIER DU PROJET

Le calendrier du projet permet de suivre l'évolution du projet. c'est dans celui ci qu'il faut indiquer les différentes phases du projet et les deadlines associées à ces phases.

### 1.2.1 LA PHASE DE CONCEPTION DU PROJET

Dans cette phase, il s'agissait de la rédaction du cahier des charges. Elle s'étendait sur la période du 07 février au 2 mars 2014. La version finale du cahier des charges a été livrée le dimanche 2 mars 2014.

### 1.2.2 LA PHASE DE RÉALISATION DU PROJET

Cette phase consiste en la rédaction du cadre de réponse au cahier des charges. A l'issu de cette phase, il faudrait fournir entre autres documents :

- le code source ;
- la documentation d'exploitation et d'architecture(DAT, PTI ...);
- les droits de propriété ;

L'étape de réalisation a commencé le lundi 3 mars 2014 et la deadline est fixée au 26 mars 2014.

## 1.3 L'OBJECTIF DU PROJET

L'objectif de la réalisation du projet est de fournir d'une part, à court terme une version finale du dossier d'architecture technique(DAT), les documents d'exploitation et d'autre part, à long terme la solution logicielle de type Transport Management System(TMS) attendue.

## 1.4 LES ENJEUX DU PROJET

Garmir Khatch espère améliorer la qualité de ses services Grâce au TMS, et le fonctionnement global de l'organisation grâce à l'assistance de l'outil et l'automatisation de certaines de ces tâches. Le TMS lui permettra non seulement de réduire les coûts relatifs à la logistique mais également de maintenir son professionnalisme, améliorant ainsi son image de marque par preuve de son efficacité et de son efficience sur le terrain.

## 1.5 LES EXIGENCES DU PROJET

cette section précise les services attendus et les exigences du projet.

### 1.5.1 LA LOCALISATION

La suite logicielle doit proposer des versions traduites en plusieurs langues(Anglais, Français, Espagnol, Arabe) avec des alphabets et des sens de lecture différents.

### 1.5.2 L'INTERFACE UTILISATEUR

La solution doit fournir au minimum et pour chacun de ces groupes utilisateurs une interface propre permettant de réaliser les actions liées à chacun d'eux.

### 1.5.3 LES CAS D'UTILISATIONS

Le TMS devra permettre :

- la gestion des niveaux de sécurité(cryptographie) ;
- la gestion des droits d'accès ;
- la gestion des sauvegardes et des préférences ;
- la gestion de la synchronisation ;
- la gestion des tableaux de bord ;
- la gestion des statistiques, des réquisitions & Waybills/ Delivery notes ;
- la gestion des chauffeurs et des prestataires.

### 1.5.4 LES CONTRAINTES

Il y a plusieurs contraintes définis par Garmir Khatch qui se doivent d'être respecter :

- **la contrainte de compatibilité** : La suite logicielle doit impérativement être compatible avec les matériels utilisés, les systèmes d'exploitation et, au minimum, les versions de logiciels.
- **la contrainte sur les coûts et les moyens de communications** : il faut limiter les coûts et se plier au préférences sur les moyens de communications définies dans le cahier des charges.
- **la contrainte de fonctionnalité** : La suite logicielle doit fournir un outil intégré permettant de numériser les documents qui doivent l'être, via une interface graphique clair permettant de fixer facilement les paramètres de numérisation.
- **la contrainte de délai** : Le prestataire s'engage à fournir des garanties contractuelles de délais, et à s'y tenir. Les éventuelles pénalités de retard seront fixées d'un accord commun avec Garmir Khatch.
- **la contrainte légale** : Les contraintes légales de chaque pays s'appliquant à Garmir Khatch lors de ses interventions, la solution devra être aux normes (ou pouvoir s'y adapter) de ces pays.
- **la contrainte réglementaire** : Garmir Khatch a une image et une éthique mondialement connue découlant de ses activités. Il conviendra de la prendre en compte lors de la réalisation de la solution.

## 1.6 LE PLANNING



## 2 — Architecture fonctionnelle



## 3 — Architecture technique

### 3.1 DONNÉES

#### 3.1.1 PROBLÉMATIQUE

Ce chapitre traite du stockage et du formatage des données sur les différents postes, et dans les différentes situations d'utilisation précédemment décrites. Cette problématique est divisée selon trois points clefs :

- Le stockage ;
- Le transfert ;
- L'importation et l'exportation.

#### 3.1.2 LE STOCKAGE

##### CÔTÉ SERVEUR

Le stockage des données côté serveur se fera grâce au SGDB Postgrès SQL. En effet, cette solution a été retenue pour plusieurs raisons :

- Il s'agit d'une solution libre ;
- Cette solution est compatible avec la plupart des systèmes d'exploitation (Window, Linux, Mac, ...);
- Elle est pratiquement conforme aux normes qui régissent le langage SQL, ce qui garantit une bonne portabilité en cas de migration sur une autre solution ;
- Il s'agit de la solution de prédilection de notre entreprise, ce qui garantit donc une bonne qualité de réalisation.

##### CÔTÉ CLIENT

Le stockage des données côté client se fera soit via une base de données locale (également sous Postgrès SQL), s'il est possible de la synchroniser avec la base de données centrale et que le périphérique le permet, soit via des fichiers au format XML. Le choix de l'utilisation de fichiers XML est dû aux raisons suivantes :

- Il s'agit d'un format universel, et compatible à l'import/export avec la plupart des SGDB (dont Postgrès SQL ;
- On obtient un taux de compression assez important sur ce format, ce qui peut favoriser le transfert des données dans des situations où la connexion est limitée ;
- Ce format est exploitable sur les périphériques Androïd.

#### 3.1.3 LE TRANSFERT

##### FORMAT

Eu égard des solutions retenues pour le transfert des données, le transfert des données se fera grâce à des fichiers XML. La possibilité sera donnée de compresser ces fichiers, dans un premier temps au format *zip*, mais ce choix pourra être remis en question selon les observations réalisées sur les données de test, afin de retenir une éventuelle meilleure solution.

## NORME

**3.1.4 L'IMPORT/EXPORT DE DONNÉES**

L'import et l'export de données se fera directement via une fonctionnalité de l'outil côté client, qui permettra de sélectionner les données à importer dans le contexte spécifique, ainsi que les données à exporter vers le serveur central, toujours selon ce contexte. Comme décrits dans la partie traitant de ce sujet, l'import/export de données pourra être réalisé selon différents supports (réseau, support physique) à la discrétion de l'exploitant.

**3.2 SAUVEGARDE****3.2.1 PROBLÉMATIQUE**

Il est indiqué dans le cahier des charges que la suite logicielle doit pouvoir synchroniser des données, à la fois entre postes clients et serveur local, mais également entre serveur local et serveur central. Cette demande pose les problèmes de la disponibilité du serveur local et de l'intégrité des données. Différentes architectures vont être proposées afin de garantir ces contraintes.

**3.2.2 CLUSTER**

Une première solution utilisant la technologie du *cluster* peut être envisagée. Cette technique permet de créer un groupe logique de serveurs qui s'exécutent simultanément tout en donnant l'impression aux utilisateurs de ne constituer qu'un seul serveur. En considérant le matériel existant, et le fait que l'achat de serveur dédié pour ce cluster serait couteux au client, une solution de *clustering* peut être effectuée grâce à de simples postes clients qu'il faut configurer comme des serveurs. Les étapes de cette configuration sera fournie dans le manuel de déploiement. Deux architectures différentes sont proposées :

- Deux postes clients ( configurés comme des serveurs maître/esclave) sont reliés directement par un cable ethernet, ce qui va permettre de dupliquer au fur et à mesure toutes les données. Le serveur maître transmet les données au serveur esclave, et en cas d'incident sur le serveur maître, c'est le serveur esclave qui reprend la main.
- Les deux postes ne sont pas directement reliés, mais c'est le routeur qui va transmettre les données aux deux serveurs. Cette solution permet d'augmenter la distance entre les deux serveurs, voir d'utiliser des locaux différents en évitant ainsi une coupure d'électricité au niveau des deux locaux, ou bien la propagation d'un incendie. Par contre cette solution, va augmenter de manière significative l'utilisation réseau, ce qui n'est pas négligeable vu les contraintes des communications réseaux.

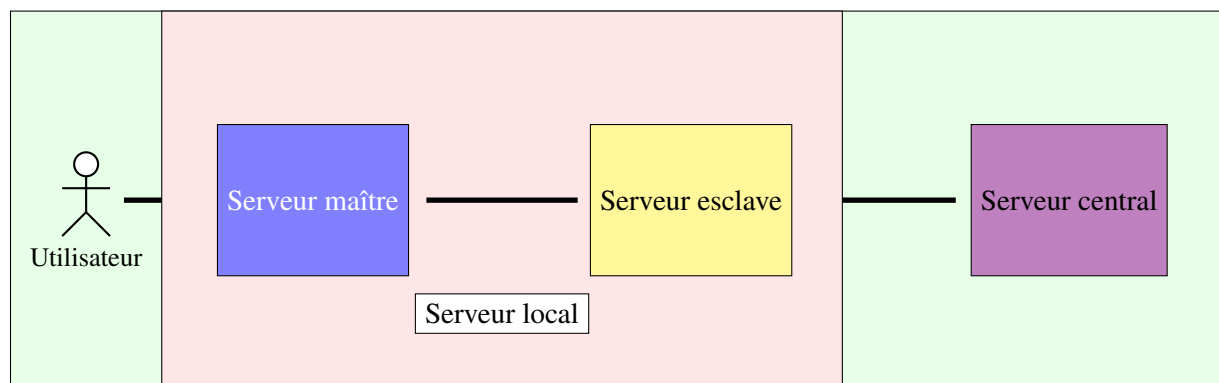


FIGURE 3.1 – Cluster, serveur maître et esclave directement connecté



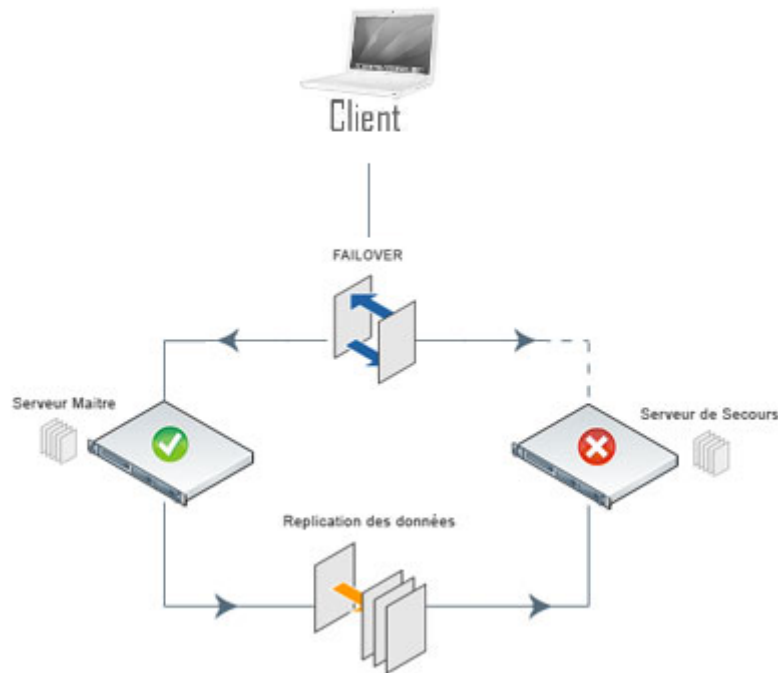


FIGURE 3.2 – Cluster, serveur maître et esclave directement connecté

### 3.2.3 REPRISE SUR PANNE

### 3.2.4 SAUVEGARDE EN DURE

Une deuxième solution peut être mise en œuvre de manière complémentaire au *cluster* afin de mieux respecter la contrainte concernant l'intégrité des données. Un support mobile de stockage est utilisé afin de directement sauvegarder le contenu des serveurs. Le support peut être un disque dur, une clé usb, une carte mémoire...

## 3.3 SYNCHRONISATION

### 3.3.1 INTRODUCTION

L'objectif de cette section est de décrire les méthodes proposées afin de répondre au problème de synchronisation. Les échanges de données pouvant être limités (pas de réseau, liaison satellitaire uniquement) il convient de pouvoir choisir précisément les éléments que l'on souhaite synchroniser<sup>1</sup>.

Pour permettre la gestion de ces éléments, on introduit la notion de *priorité* ; à chaque catégorie d'éléments (e.g. planning des transports, liste des fournisseurs, informations sur les véhicules, ...) peut être associée une priorité de laquelle dépendra la synchronisation ou non. À partir de cette « hiérarchie d'importance », on associe des *profils de synchronisation* paramétrables qui - une fois activés - gère la synchronisation de façon transparente en fonction des choix de l'utilisateur. Pour résoudre les problèmes de connexion et assurer le fonctionnement de la suite logicielle indépendamment de la liaison réseau, deux modes sont prévus :

- Connecté
- Hors-ligne

1. Il est à noter que la synchronisation est bi-directionnelle : on reçoit les informations du serveur autant qu'on en envoie.

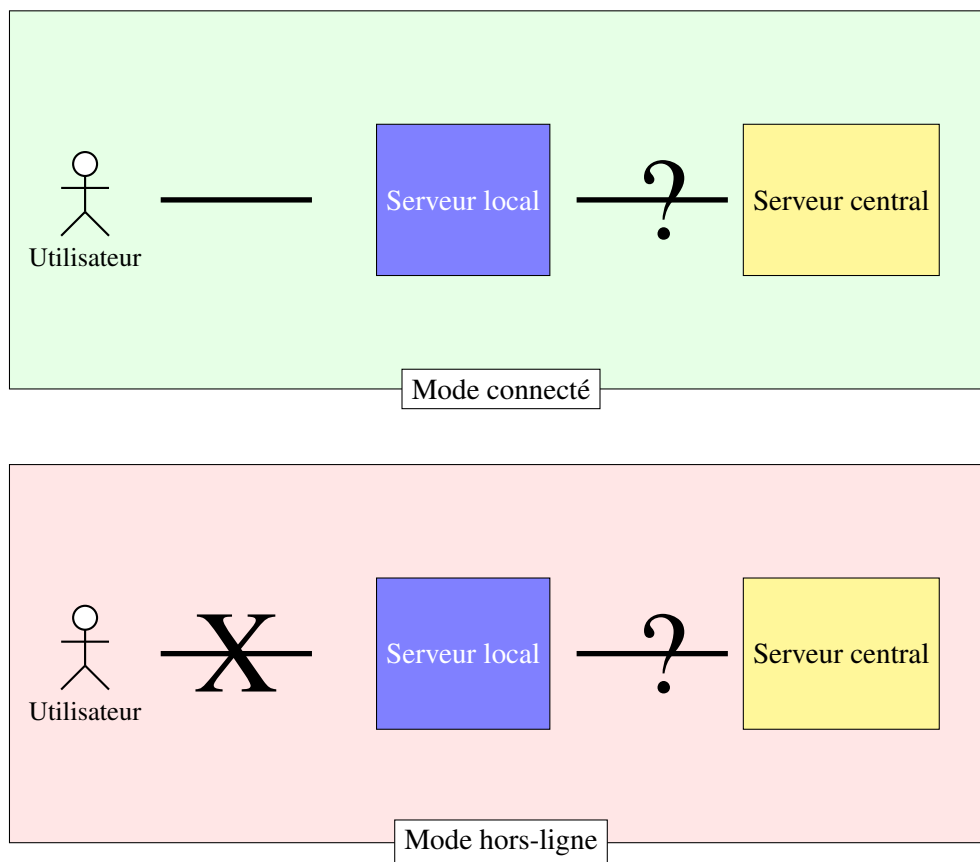


FIGURE 3.3 – Utilisation des modes connecté et hors-ligne

### 3.3.2 GESTION DES PRIORITÉS & DES PROFILS DE SYNCHRONISATION

Une priorité permettra à l'application de « choisir » ce qui transitera sur le réseau. Cinq priorités seront définies (par ordre d'importance croissant) pour qualifier des éléments synchronisables :

1. Négligeable
2. Secondaire
3. Normal
4. Important
5. Crucial

L'utilisateur pourra gérer les priorités à synchroniser en les associant à un profil : lorsque l'utilisateur choisit un profil de synchronisation, seuls les éléments ayant une priorité incluse dans le profil sont synchronisés.

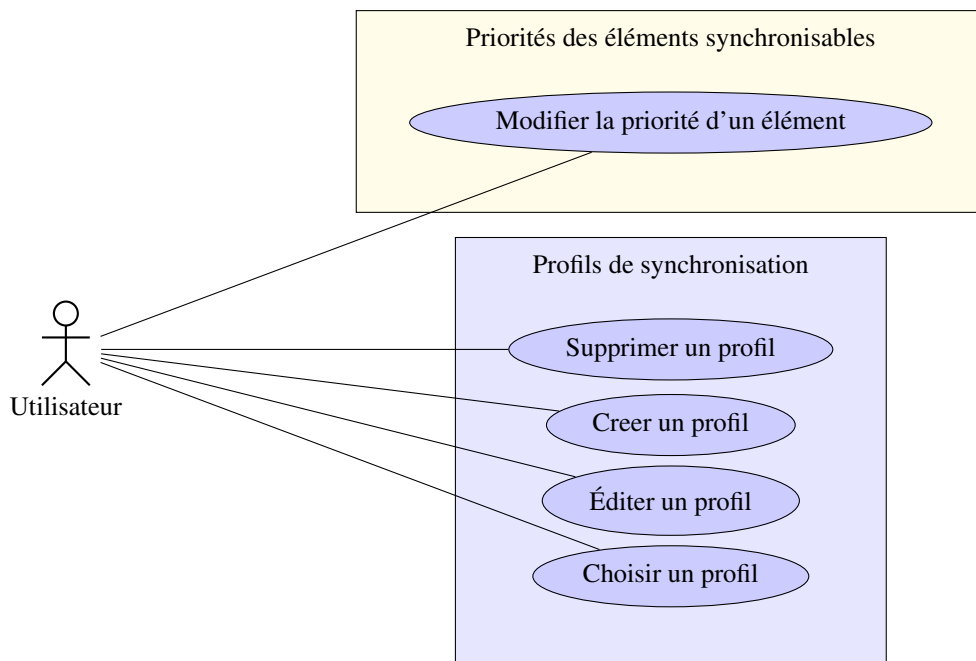


FIGURE 3.4 – Diagramme des cas d'utilisation pour la synchronisation

### 3.3.3 MODE CONNECTÉ

Dans ce mode, on suppose qu'il y a une liaison réseau entre l'utilisateur et le serveur local ; toutes les modifications sont effectuées en « temps réel<sup>2</sup> ». Pour ce mode, l'utilisateur doit :

1. Se connecter
2. Utiliser la suite logicielle
3. Se déconnecter après utilisation

L'utilisateur ne pourra effectuer que les modifications dont il a les *permissions*.

### 3.3.4 MODE HORS-LIGNE

Contrairement au mode connecté, le mode hors ligne permet à l'utilisateur de modifier toutes les informations dont il dispose localement. Lorsqu'il souhaitera synchroniser ses informations avec le serveur, il devra se connecter et c'est à ce moment là que le serveur vérifiera que

2. Un écart de temps pourra être constaté dans le cas où le réseau n'offre qu'un faible débit.

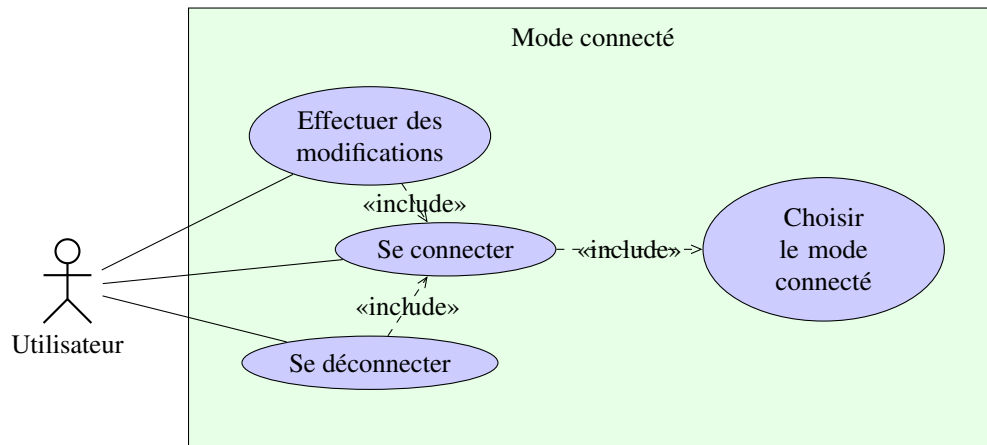


FIGURE 3.5 – Diagramme des cas d'utilisation pour le mode connecté

l'utilisateur n'outrepasse pas les droits qui lui sont accordés. Si c'est le cas, le serveur rejettera les modifications locales.

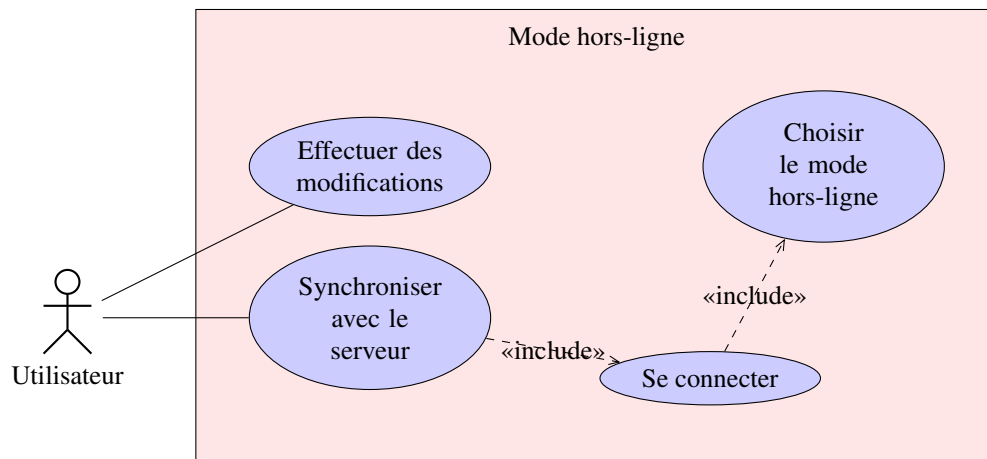


FIGURE 3.6 – Diagramme des cas d'utilisation pour le mode hors-ligne

### 3.3.5 GESTION DES CONFLITS

Durant la synchronisation, que ce soit en mode connecté ou hors-ligne, des conflits peuvent survenir au niveau du contenu des informations. Il reviendra à l'utilisateur le soin de gérer ces conflits en choisissant l'information correcte qui doit être enregistrée sur le serveur.



## 4 — Exploitation et préparation