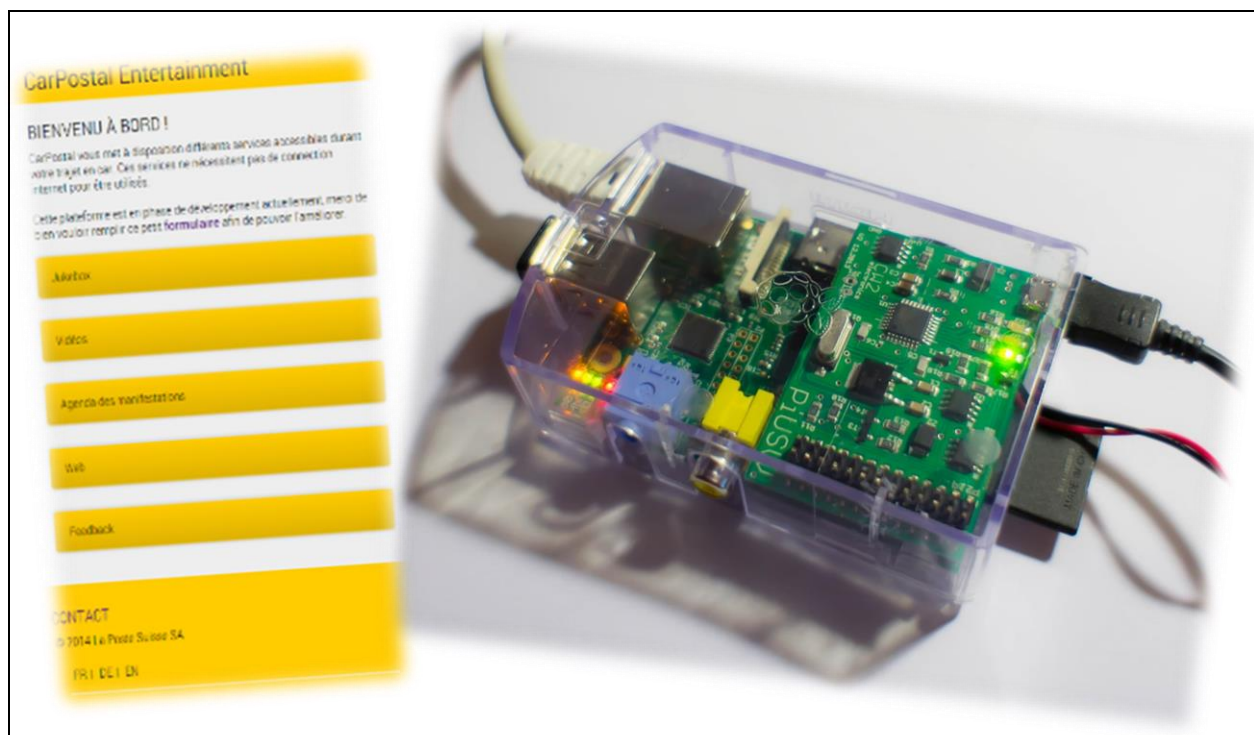


## Travail de Bachelor 2014

### CarPostal Entertainment



Etudiant-e : Célien Devanthéry

Professeur : Pierre-Yves Guex

Mandant : CarPostal SA

Déposé le : 28.07.2014

## RÉSUMÉ

Le but de ce travail est de favoriser la croissance du nombre de passagers dans les cars postaux en développant une plateforme pouvant accueillir des services à valeur ajoutée. Ces services doivent être accessibles par les occupants du car via leurs smartphones ou tablettes. La plateforme doit répondre à plusieurs contraintes, tel que le multilinguisme et le fait d'avoir une interface agréable et fonctionnelle pour l'utilisateur (UX Design).

Pour ce faire, plusieurs étapes ont été planifiées. Premièrement une analyse de solutions existantes, tant propriétaires qu'open-sources, a été réalisée. Suite à cette analyse, il en est ressorti que des solutions propriétaires existaient, mais qu'elles ne répondaient pas forcément aux besoins spécifiés. C'est pourquoi le choix s'est porté sur le développement d'un nouveau système.

Le développement a débuté par un choix des outils et des technologies à utiliser, avec notamment le choix d'utiliser un Raspberry Pi en tant que base de la plateforme. Puis le développement de la plateforme a été réalisé, avec notamment la configuration du hardware, la création de l'interface utilisateur et la réalisation de différents tests.

La troisième étape du projet comprend le développement de quelques services afin de pouvoir tester et valider le fonctionnement de la plateforme en environnement réel. Dans ce but, elle a été placée un certain temps dans un car de la ligne Sierre – Zinal. Les résultats seront ajoutés à ce rapport sur forme d'errata.

Pour conclure, la dernière partie du projet concerne la rédaction de ce rapport. Le but étant de documenter le travail réalisé pour ainsi permettre la continuité du développement. Un CD-ROM contenant tous les codes sources est disponible à la fin de ce rapport.

Mots-clés : CarPostal Entertainment, Travail de Bachelor, Raspberry Pi, FIG, 2014.

## AVANT-PROPOS

A l'heure actuelle, les smartphones et autres tablettes prennent une part de plus en plus importante dans la vie de tous les jours. Ce qui demande aux entreprises d'améliorer leurs prestations afin de répondre aux attentes de leurs clients.

Dans ce contexte, CarPostal a décidé d'installer une connexion internet avec un accès WI-FI à l'intérieur de ses cars. De plus, l'entreprise désire développer des services à valeur ajoutée à proposer à ses passagers. C'est dans ce but précis que ce travail de Bachelor intervient.

Une plateforme fonctionnelle pouvant accueillir différents services a été développée. Actuellement un jukebox, un lecteur de vidéos et un agenda des manifestations sont disponibles. Une amélioration de ces services, ainsi que l'ajout de nouveaux services sont nécessaires afin d'améliorer l'attractivité et l'utilité de la plateforme. Des pistes d'améliorations sont décrites dans ce rapport.

La principale difficulté rencontrée réside dans le fait que la connexion internet présente dans les cars est très variable, voir indisponible. Il a fallu développer un système utilisant du contenu stocké localement disponible en tout temps. Ce contenu est accompagné d'un mécanisme permettant sa mise à jour à distance.

Afin de développer cette plateforme, une planification a été réalisée pour obtenir une vue globale des étapes et de leur durée. Suite à cela le projet a été divisé en quatre phases distinctes.

## REMERCIEMENTS

Je tiens à remercier toutes les personnes m'ayant soutenu durant le travail de Bachelor, tout particulièrement :

M. Pierre-Yves Guex, professeur à la HES-SO, pour son encadrement et son suivi sur toute la durée du projet,

M. Alexandre Cotting, professeur à la HES-SO, pour sa précieuse expertise technique et ses conseils,

M. Laurent Flück, entrepreneur CarPostal et responsable de la région Anniviers, pour sa disponibilité,

Mes camarades présents en salle de projet pour leurs conseils et la bonne ambiance de travail qui y régnait,

Toutes les personnes qui ont participé à la traduction de la plateforme et à la relecture de ce document.

# TABLE DES MATIÈRES

<b>LISTE DES TABLEAUX .....</b>	<b>viii</b>
<b>LISTE DES FIGURES .....</b>	<b>ix</b>
<b>GLOSSAIRE ET LISTE DES ABRÉVIATIONS .....</b>	<b>x</b>
<b>1. INTRODUCTION.....</b>	<b>1</b>
<b>2. GESTION DE PROJET .....</b>	<b>2</b>
2.1. ORGANISATION DU PROJET .....	2
2.2. CAHIER DES CHARGES.....	2
2.3. NOMBRE D'HEURES RÉALISÉES .....	2
<b>3. ÉTAT DE L'ART .....</b>	<b>3</b>
3.1. PRÉSENTATION DES SOLUTIONS EXISTANTES.....	3
3.1.1. <i>Velimo</i> .....	3
3.1.2. <i>BLUE</i> .....	5
3.1.3. <i>Beam WI-Fi</i> .....	5
3.1.4. <i>Solutions Open-Source</i> .....	6
3.2. COMPARAISON DES SOLUTIONS .....	6
3.3. CONCLUSION.....	7
<b>4. DÉVELOPPEMENT DU SYSTÈME .....</b>	<b>8</b>
4.1. ENVIRONNEMENT ET OUTILS DE DÉVELOPPEMENT .....	8
4.1.1. <i>Adobe Dreamweaver</i> .....	8
4.1.2. <i>PuTTY</i> .....	8
4.1.3. <i>Microsoft OneDrive</i> .....	8
4.1.4. <i>Nginx</i> .....	9
4.1.5. <i>MySQL</i> .....	9
4.2. PRÉSENTATION DES COMPOSANTS DE LA PLATEFORME.....	10
4.3. ARCHITECTURE LOGICIELLE .....	11
4.3.1. <i>Modèle MVC</i> .....	11
4.3.2. <i>Organisation du code source</i> .....	13
4.4. HARDWARE .....	14
4.4.1. <i>Avantages</i> .....	14
4.4.2. <i>Inconvénients</i> .....	15
4.4.3. <i>Systèmes concurrents</i> .....	15
4.4.4. <i>Conclusion</i> .....	16
4.5. TECHNOLOGIES UTILISÉES.....	16
4.5.1. <i>Pour la configuration du serveur web local</i> .....	16
4.5.2. <i>Pour le développement de la plateforme</i> .....	17

4.6.	CONFIGURATION DU SERVEUR CENTRAL .....	17
4.7.	CONFIGURATION DU SERVEUR WEB LOCAL (RASPERRY PI) .....	18
4.8.	DESCRIPTIFS DES FONCTIONNALITÉS DE LA PLATEFORME .....	20
4.8.1.	<i>Page d'accueil</i> .....	20
4.8.2.	<i>Multilinguisme</i> .....	21
4.8.3.	<i>Responsive Web Design</i> .....	21
4.8.4.	<i>Mises à jour du contenu</i> .....	22
4.8.5.	<i>Formulaire de contact</i> .....	24
<b>5.</b>	<b>DÉVELOPPEMENT DES SERVICES .....</b>	<b>25</b>
5.1.	JUKEBOX .....	25
5.2.	LECTEUR DE VIDÉOS .....	26
5.3.	AGENDA DES MANIFESTATIONS .....	26
5.4.	CONTRÔLE DE LA CONNEXION INTERNET .....	27
5.5.	QUESTIONNAIRE .....	27
<b>6.</b>	<b>TESTS DE LA PLATEFORME .....</b>	<b>28</b>
6.1.	TESTS DE CHARGE .....	28
6.1.1.	<i>Création d'un scénario d'utilisation</i> .....	29
6.1.2.	<i>Test 1 : 50 utilisateurs</i> .....	30
6.1.3.	<i>Test 2 : 1'000 utilisateurs</i> .....	30
6.1.4.	<i>Conclusion</i> .....	31
6.2.	TEST DES SERVICES .....	32
6.2.1.	<i>Jukebox et lecteur vidéos</i> .....	32
6.2.2.	<i>Agenda des manifestations</i> .....	32
6.2.3.	<i>Formulaires de feedback et de contact</i> .....	32
6.3.	TEST DU SYSTÈME DE MISE À JOUR .....	33
6.4.	TEST SUR LA LIGNE SIERRE-ZINAL .....	34
<b>7.</b>	<b>PROBLÈMES RENCONTRÉS .....</b>	<b>35</b>
7.1.	ALIMENTATION DU RASPERRY PI .....	35
7.2.	SYNCHRONISATION DE L'HEURE .....	35
7.3.	ARCHITECTURE MVC .....	36
7.4.	GESTION DES LANGUES .....	36
<b>8.</b>	<b>ÉTAT DES LIEUX .....</b>	<b>37</b>
8.1.	ÉTAT ACTUEL DU PROJET .....	37
8.2.	AMÉLIORATIONS FUTURES .....	37
8.2.1.	<i>Améliorations des services existants</i> .....	37
8.2.2.	<i>Ajout de nouveaux services</i> .....	37

8.2.3.	<i>Intégration avec le système existant</i>	38
8.2.4.	<i>Amélioration de l'UPS</i>	38
8.2.5.	<i>Création d'un écran de contrôle</i>	38
8.2.6.	<i>Création d'une page d'administration</i>	38
<b>CONCLUSION</b>		<b>39</b>
<b>RÉFÉRENCES</b>		<b>40</b>
<b>DÉCLARATION DE L'AUTEUR</b>		<b>43</b>
<b>ANNEXE I : PLANIFICATION DU PROJET</b>		<b>44</b>
<b>ANNEXE II : CAHIER DES CHARGES</b>		<b>45</b>
<b>ANNEXE III : JOURNAL DE BORD</b>		<b>48</b>
<b>ANNEXE IV : RÉPONSE DE LUFTHANSA SYSTEMS</b>		<b>51</b>
<b>ANNEXE V : CAPTURES D'ÉCRAN DES SERVICES</b>		<b>54</b>
<b>ANNEXE VI : RÉSULTAT DU LOAD TEST AVEC 50 UTILISATEURS</b>		<b>56</b>
<b>ANNEXE VII : RÉSULTAT DU LOAD TEST AVEC 1'000 UTILISATEURS</b>		<b>58</b>
<b>ANNEXE VIII : INFORMATIONS POUR ACCÉDER AUX RESSOURCES</b>		<b>60</b>
<b>ANNEXE IX : TUTORIEL D'INSTALLATION DE LA PLATEFORME</b>		<b>61</b>

## LISTE DES TABLEAUX

<b>Tableau 1</b> : Comparaison des solutions existantes .....	6
<b>Tableau 2</b> : Récapitulatif du résultat du Load Test réalisé avec 50 utilisateurs.....	30
<b>Tableau 3</b> : Récapitulatif du résultat du Load Test réalisé avec 1'000 utilisateurs.....	31



## LISTE DES FIGURES

<b>Figure 1</b> : Velimo, présentation du système .....	4
<b>Figure 2</b> : Beam WI-FI, présentation du système .....	5
<b>Figure 3</b> : Dreamweaver .....	8
<b>Figure 4</b> : PuTTY .....	8
<b>Figure 5</b> : OneDrive.....	9
<b>Figure 6</b> : Nginx.....	9
<b>Figure 7</b> : MySQL .....	9
<b>Figure 8</b> : Vue globale des composants hardware de la plateforme.....	10
<b>Figure 9</b> : Le modèle d'architecture MVC.....	12
<b>Figure 10</b> : Organisation du code source de la plateforme .....	13
<b>Figure 11</b> : Le Raspberry Pi, Type B .....	14
<b>Figure 12</b> : Raspbian .....	16
<b>Figure 13</b> : Raspi-config .....	18
<b>Figure 14</b> : Page d'accueil en français de la plateforme vue depuis un smartphone.....	20
<b>Figure 15</b> : Illustration du principe de Responsive Web Design .....	22
<b>Figure 16</b> : Extrait du script permettant la mise à jour du contenu .....	23
<b>Figure 17</b> : Extrait du script permettant la mise à jour de la base de données .....	23
<b>Figure 18</b> : Aperçu de l'enregistrement dans la table du logiciel « cron » .....	24
<b>Figure 19</b> : Formulaire de contact .....	24
<b>Figure 20</b> : Aperçu de la table <i>songs</i> .....	25
<b>Figure 21</b> : Aperçu de la table <i>movies</i> .....	26
<b>Figure 22</b> : Aperçu de la table <i>events</i> .....	26
<b>Figure 23</b> : Aperçu de la table <i>feedbacks</i> .....	27
<b>Figure 24</b> : Logo de Gatling.....	28
<b>Figure 25</b> : Gatling – scénario de test.....	29
<b>Figure 26</b> : Graphique résumant le Load Test réalisé avec 1'000 utilisateurs .....	31

## GLOSSAIRE ET LISTE DES ABRÉVIATIONS

Ajax	: Asynchronous JavaScript and XML, ensembles de technologies servant à mettre à jour des pages web de manière invisible pour l'utilisateur.
ARM	: Type de processeur basé sur l'architecture RISC
Bande passante	: Débit de la connexion à internet.
CSS	: Cascading Style Sheet, langage utilisé pour mettre en page des sites web, également appelé feuilles de style CSS.
DNS	: Domain Name System, permet de retrouver l'adresse IP liée à un nom de domaine (exemple : l'adresse IP de <a href="http://www.carpostal.ch">www.carpostal.ch</a> est 194.41.184.143).
FTP	: File Transfer Protocol, protocole d'échange de fichiers.
GPRS	: General Packet Radio Service, norme pour la téléphonie mobile.
Hardware	: Ensemble du matériel informatique physique par opposition au Software.
HDMI	: High Definition Multimedia Interface, interface permettant le transfert de flux vidéos et audios.
HTML	: Hypertext Markup Language, langage utilisé pour représenter des pages web.
Interface réseau	: Élément d'un ordinateur qui assure la connexion avec un réseau.
IP	: Internet Protocol Address, adresse attribuée à chaque appareil connecté à internet.
JavaScript	: Langage de programmation de scripts utilisé pour rendre les pages web interactives.
jQuery	: Librairie JavaScript libre.
Media Queries	: Règles appliquées au niveau du CSS, afin de modifier le design d'une page internet sous certaines conditions.
Open-source	: Type de licence permettant l'accessibilité au code source, la redistribution et la création de travaux dérivés.
Overclocker	: Augmentation du signal de l'horloge d'un processeur afin d'augmenter ses performances.

PHP	: Hypertext Preprocessor, langage libre utilisé pour le développement de pages web dynamiques.
Pilote	: Programme informatique permettant à un ordinateur d'utiliser un périphérique externe.
RAM	: Random Access Memory, mémoire utilisée par l'ordinateur pour placer des données en cours de traitement.
Scripts	: Programme permettant d'exécuter des commandes.
SQL	: Structured Query Language, langage permettant la gestion de base de données.
SSH	: Secure Shell, protocole de communication sécurisé par un système de clés privées et publiques.
Temps de latence	: Temps écoulé entre une commande et sa réalisation.
UPS	: Uninterruptible Power Supply, dispositif permettant de fournir une alimentation constante à un système électrique ou électronique.
UX Design	: User Experience Design, concept pour améliorer la satisfaction des utilisateurs.
WYSIWYG	: What You See Is What You Get, programme permettant d'avoir le rendu en temps réel du code en cours de développement

## 1. INTRODUCTION

CarPostal Suisse SA est la plus importante entreprise de transport par cars de Suisse. Actuellement elle dispose d'un parc de plus de 2'200 véhicules et transporte environ 139 millions de voyageurs par année.

Afin d'améliorer son offre et de favoriser une croissance du nombre de passagers, CarPostal a installé une connexion internet à l'intérieur de ses cars avec un accès WI-FI. Par la même occasion, l'entreprise aimerait développer des services à valeur ajoutée à proposer à ses passagers. Il s'agirait d'installer une plateforme contenant divers services accessibles via WI-FI.

C'est dans ce but qu'intervient ce travail. Dans un premier temps, une analyse a eu lieu pour déterminer si une solution existe ou si un nouveau système doit être développé. Suite à cette analyse, la phase de développement a débuté. Cette phase s'est déroulée en trois parties, une partie développement de la plateforme de base pour débiter, suivie d'une partie développement des services et enfin une partie tests.

## 2. GESTION DE PROJET

### 2.1. ORGANISATION DU PROJET

Un diagramme de Gantt a été créé afin de planifier le déroulement du projet et d'avoir un aperçu global (cf. **Annexe I**). Cet aperçu a permis de déterminer quatre grandes phases.

Dans un premier temps, une phase d'analyse est nécessaire, afin de rechercher des solutions existantes, les analyser et déterminer si une solution répond aux critères.

Ensuite, une fois l'analyse terminée, une phase de développement doit débiter. Cette étape est définie par le résultat de l'analyse. Si une solution a été choisie elle sera mise en place, sinon un nouveau système sera créé.

La troisième phase est la phase de développement des services. Cette phase comprend également une période de test sur la ligne Sierre – Zinal.

Enfin, la phase finale est consacrée au rapport, à sa rédaction, sa mise en page, sa correction et son impression. Durant cette phase le code source est également nettoyé et commenté.

### 2.2. CAHIER DES CHARGES

Un cahier des charges a été réalisé au début du projet afin de définir les limites du travail. Il a été approuvé par M. Pierre-Yves Guex le 17 mai 2014 (cf. **Annexe II**).

### 2.3. NOMBRE D'HEURES RÉALISÉES

La durée prévue pour la réalisation du travail de Bachelor est de 360 heures. Ce nombre d'heure a été atteint, le projet a duré 367 heures. Un journal de bord détaillant les activités hebdomadaires effectuées se trouve dans les annexes (cf. **Annexe III**).

### 3. ÉTAT DE L'ART

Il existe actuellement sur le marché beaucoup de solutions de plateformes de divertissement pour avion. Toutes les compagnies faisant des vols long-courriers en possèdent une. Cependant, il est plus difficile de trouver des solutions adaptées aux cars. Après différentes recherches, plusieurs plateformes ont été trouvées.

Ces solutions seront présentées et analysées dans ce chapitre en fonction des critères suivants afin de définir si elles répondent aux besoins de CarPostal :

- Le type de licence (propriétaire ou open-source) de la plateforme
- Son prix
- Sa disponibilité ou non en Suisse
- La possibilité d'ajouter de nouvelles fonctionnalités et les coûts de développement y relatifs
- L'existence d'un mode de fonctionnement hors ligne
- Le multilinguisme (Français / Allemand / Anglais)
- L'accessibilité du support

#### 3.1. PRÉSENTATION DES SOLUTIONS EXISTANTES

##### 3.1.1. Velimo<sup>1</sup>

Velimo est une plateforme de divertissement pour cars et trains. Ce système a été développé par Lufthansa Systems<sup>2</sup> qui est une société de développement informatique spécialisée dans le développement de produits et services à valeur ajoutée pour leurs clients.

Après le branchement dans le car d'une boîte contenant un serveur, un routeur, un modem GPRS et un point d'accès WI-FI, les passagers utilisent leurs propres appareils (smartphone, tablette ou ordinateur portable) afin de se connecter au système. Une fois connectés, ils ont accès à différents services, tel que le visionnage de films, l'écoute de musique ou de livres

---

<sup>1</sup> Source : <https://www.lhsystems.com/solutions-services/industry-solutions-services/hospitality-solutions/velimo-road-n-rail.html>, visité en mai 2014

<sup>2</sup> Source : <https://www.lhsystems.com/solutions-services.html>, visiter en mai 2014

audio, le jeu, la lecture de journaux, de magazines ou de livres électroniques, l'accès à internet, ainsi que le visionnage de guides de voyage (cf. **Figure 1**).

**Figure 1** : Velimo, présentation du système



Source : [https://www.lhsystems.com/fileadmin/user\\_upload/files/en/information/br-velimo-24022014.pdf](https://www.lhsystems.com/fileadmin/user_upload/files/en/information/br-velimo-24022014.pdf), p.4, juillet 2014

Les passagers peuvent accéder au système Velimo avant, pendant et après leur trajet en car, afin de regarder la fin d'un film par exemple. De plus, selon le descriptif de la plateforme, une utilisation hors connexion à internet est possible étant donné que certaines fonctionnalités sont stockées sur le serveur dans le car. Enfin, cette solution a été testée et est en service depuis fin 2013 chez la compagnie de car allemande ADAC Postbus<sup>3</sup>.

Afin de compléter les informations trouvées, une prise de contact a été effectuée via mail le 16 mai 2014, leur réponse est parvenue le 18 juin 2014 (cf. Annexe IV). Dans cette réponse, Lufthansa Systems précise que la plateforme ne possède pas de restriction territoriale et donc peut être utilisée en Suisse. Le prix d'une telle plateforme dépend de plusieurs facteurs, tel que la taille de la flotte de véhicules et les fonctionnalités choisies. La plateforme a été conçue afin de s'adapter aux besoins spécifiques des clients, il est possible d'ajouter des services personnalisés. Le prix de développement de ces services peut être défini après une analyse

<sup>3</sup> Source : <https://www.lhsystems.com/media/news/latest-news/press-release/article/velimo-from-lufthansa-systems-provides-entertainment-and-information-for-adac-postbus-passengers.html>, visité en mai 2014

des besoins. La majorité du contenu est disponible localement sur le serveur dans le car, ce qui permet un fonctionnement hors ligne. Des services tels que regarder une vidéo, écouter de la musique ou lire un journal peuvent être utilisés sans connexion à internet. Actuellement la plateforme n'est pas multilingue, cependant cette fonctionnalité est en cours de développement.

### 3.1.2. BLUE<sup>4</sup>

BLUE est une plateforme de divertissement développée spécifiquement, par Lufthansa Systems, pour la compagnie de car américaine Greyhound. Etant donné que ce système est basé sur la plateforme Velimo décrite précédemment, les caractéristiques de cette dernière s'y appliquent.

### 3.1.3. Beam WI-Fi<sup>5</sup>

Beam WI-Fi est un système destiné aux services de transport de passagers qui a été développé par Mobile Onboard<sup>6</sup>. Il a pour but d'offrir aux passagers un accès à internet, qui peut être gratuit ou payant, depuis leurs smartphones, tablettes ou ordinateurs portables. Ce système permet également de brancher d'autres équipements, tel qu'une machine à tickets ou un système de surveillance embarqué.

Le boîtier (cf. **Figure 2**) contenant le système s'installe dans un car. À l'intérieur un serveur web, pouvant contenir notamment des pages d'informations, du contenu multimédia ou de la publicité, est accessible via WI-Fi par les passagers. La mise à jour du contenu se fait à distance.

**Figure 2** : Beam WI-Fi, présentation du système



Source : [http://mobileonboard.com/wp-content/uploads/042\\_CBW\\_1119\\_Industry-guide.pdf](http://mobileonboard.com/wp-content/uploads/042_CBW_1119_Industry-guide.pdf), p. 1, mai 2014

<sup>4</sup> Source : <https://www.lhsystems.com/media/news/latest-news/press-release/article/greyhound-launches-blueTM-an-exclusive-wi-fi-enabled-onboard-entertainment-system.html>, visité en mai 2014

<sup>5</sup> Source : <http://mobileonboard.com/beam-bus-wi-fi/>, visité en mai 2014

<sup>6</sup> Source : <http://mobileonboard.com/about-us/>, visité en mai 2014



Comme pour Velimo une prise de contact avec la société a eu lieu afin d'obtenir des informations complémentaires sur le système. Cependant aucune réponse de leur part ne nous est parvenue à ce jour.

#### 3.1.4. Solutions Open-Source

Afin de compléter l'état de l'art, des recherches de solutions open-source ont été réalisées, mais aucune solution n'a été trouvée.

### 3.2. COMPARAISON DES SOLUTIONS

Cette analyse a pour objectif de comparer les différentes solutions, ainsi que de définir si elles répondent aux besoins de CarPostal. Comme les solutions Velimo et BLUE sont similaires elles sont traitées ensembles. De plus, toutes les informations nécessaires n'ont pas été fournies par les sociétés concernées, ce qui limite la comparaison. Le tableau suivant présente un résumé des caractéristiques des solutions (cf. **Tableau 1**).

**Tableau 1** : Comparaison des solutions existantes

Critères	Velimo	Beam WI-FI
Type de licence	Propriétaire	Propriétaire
Prix	Dépend de plusieurs critères	n/a
Disponibilité en Suisse	Oui	n/a
Ajout de fonctionnalités	Possible	n/a
Fonctionnement hors ligne	Oui	Oui
Multilingue	Disponible durant le 3 <sup>e</sup> trimestre 2014	n/a
Support / Maintenance	En Allemagne	En Angleterre

Source : Données de l'auteur.

Pour commencer les deux solutions sont de type propriétaire. Les droits d'utilisation des solutions sont définis dans un contrat de licence lors de l'achat. Par conséquent l'utilisateur n'a pas le contrôle total, il n'a pas accès au code source et doit passer par le propriétaire pour ajouter de nouvelles fonctionnalités par exemple. Dans le cas de Velimo il est possible de personnaliser le système en fonction des besoins du client. Cependant cette personnalisation a un coût qui est défini après analyse et discussion entre le propriétaire et le client.

Une licence propriétaire implique un prix d'achat qui va dépendre de plusieurs critères tel que les fonctionnalités et le contenu choisis, ainsi que de la taille de la flotte de véhicules. Auquel il faut ajouter le coût de développement des fonctionnalités personnalisées et le paiement d'un abonnement annuel pour les mises à jour et la maintenance de la plateforme.

La solution Velimo est disponible en Suisse. Cependant elle a été développée par une société basée en Allemagne ce qui peut poser problèmes en cas de nécessité d'intervention rapide ou de remplacement de matériaux défectueux. De la même manière la solution Beam WI-FI a été développée en Angleterre.

Les deux solutions offrent du contenu hors ligne ce qui est un critère indispensable pour CarPostal. Du côté de Velimo tout le contenu des services principaux est enregistré dans le serveur local, ce qui permet de pouvoir y accéder en tout temps.

Enfin pour répondre au critère du multilinguisme la solution doit être disponible en français, allemand et anglais. Cette fonctionnalité est en cours de développement chez Velimo.

### 3.3. CONCLUSION

En l'état actuel de connaissance, Velimo convient d'avantage que Beam WI-FI à l'utilisation voulue par CarPostal. Cependant, une analyse plus approfondie doit être réalisée en possession de toutes les informations nécessaires. Il est notamment indispensable de connaître les prix avant de pouvoir prendre une décision.

C'est pourquoi, étant donné le manque d'informations et le nombre de jours limités à disposition pour la réalisation de ce travail, il a été décidé qu'un nouveau système sera développé afin de répondre au mieux aux besoins de CarPostal.

## 4. DÉVELOPPEMENT DU SYSTÈME

### 4.1. ENVIRONNEMENT ET OUTILS DE DÉVELOPPEMENT

Afin de développer le système, une sélection d'outils a été réalisée. Ce chapitre explique ces différents outils.

#### 4.1.1. Adobe Dreamweaver<sup>7</sup>

**Figure 3 :**  
Dreamweaver



Dreamweaver est un outil de développement de site web WYSIWYG développé par Adobe. Il permet de modifier les éléments HTML et PHP du site internet et de prévisualiser le résultat en direct. Il intègre également un module qui facilite la gestion du CSS, ainsi qu'un gestionnaire de site pour l'envoi des pages créées localement sur le serveur web. Dans le cadre de ce projet, il a été utilisé pour le développement du code source de la plateforme.

#### 4.1.2. PuTTY<sup>8</sup>

PuTTY est un logiciel libre qui permet notamment de se connecter en SSH sur un serveur distant. Il permet d'ouvrir une session en ligne de commande sur le serveur distant. Dans le cadre de ce travail, il a été utilisé pour configurer le serveur web local et le serveur central décrits ci-après.

**Figure 4 :** PuTTY



#### 4.1.3. WinSCP<sup>9</sup>

**Figure 5 :** WinSCP



WinSCP est un logiciel libre utilisant SSH afin de sécuriser le transfert de fichiers vers un serveur distant. Il a été utilisé afin de transférer le code source de la plateforme sur le serveur central.

<sup>7</sup> Source : [http://www.adobe.com/ch\\_fr/products/dreamweaver.html](http://www.adobe.com/ch_fr/products/dreamweaver.html), visité en juin 2014

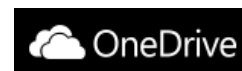
<sup>8</sup> Source : <http://www.putty.org>, visité en juin 2014

<sup>9</sup> Source : <http://winscp.net/eng/docs/lang:fr>, visité en juin 2014

#### 4.1.4. Microsoft OneDrive<sup>10</sup>

OneDrive de Microsoft est un service de stockage en ligne qui facilite la synchronisation de contenu sur plusieurs ordinateurs. Il a été utilisé pour la sauvegarde des fichiers et du code source. Des images du système d'exploitation y ont également été sauvegardées.

**Figure 6 : OneDrive**



#### 4.1.5. Nginx<sup>11</sup>

**Figure 7 : Nginx**



Nginx est un logiciel libre de serveur web qui va permettre d'afficher les pages du site web aux utilisateurs qui s'y connectent. Il a été choisi pour ces bonnes performances, ainsi que sa facilité d'installation et d'utilisation. De plus, Nginx a l'avantage d'être léger, ce qui est nécessaire étant donné que le système va être installé sur un mini-ordinateur, possédant des ressources limitées. Dans le cadre de ce projet, il a été utilisé afin d'héberger le code source de la plateforme et de l'afficher aux passagers.

#### 4.1.6. MySQL<sup>12</sup>

MySQL est le système de gestion de bases de données relationnelles le plus utilisé. Il a pour avantages d'être rapide, fiable, facile d'utilisation et open source. Il est très souvent utilisé avec PHP et possède une documentation complète. Une grande communauté de développeurs est active autour du système. Dans le cadre de ce travail, il a été utilisé pour stocker le contenu des différents services de la plateforme. L'accès aux données, ainsi que l'enregistrement se font grâce à des requêtes SQL.

**Figure 8 : MySQL**



<sup>10</sup> Source : <https://onedrive.live.com/about/fr-ch/>, visité en juin 2014

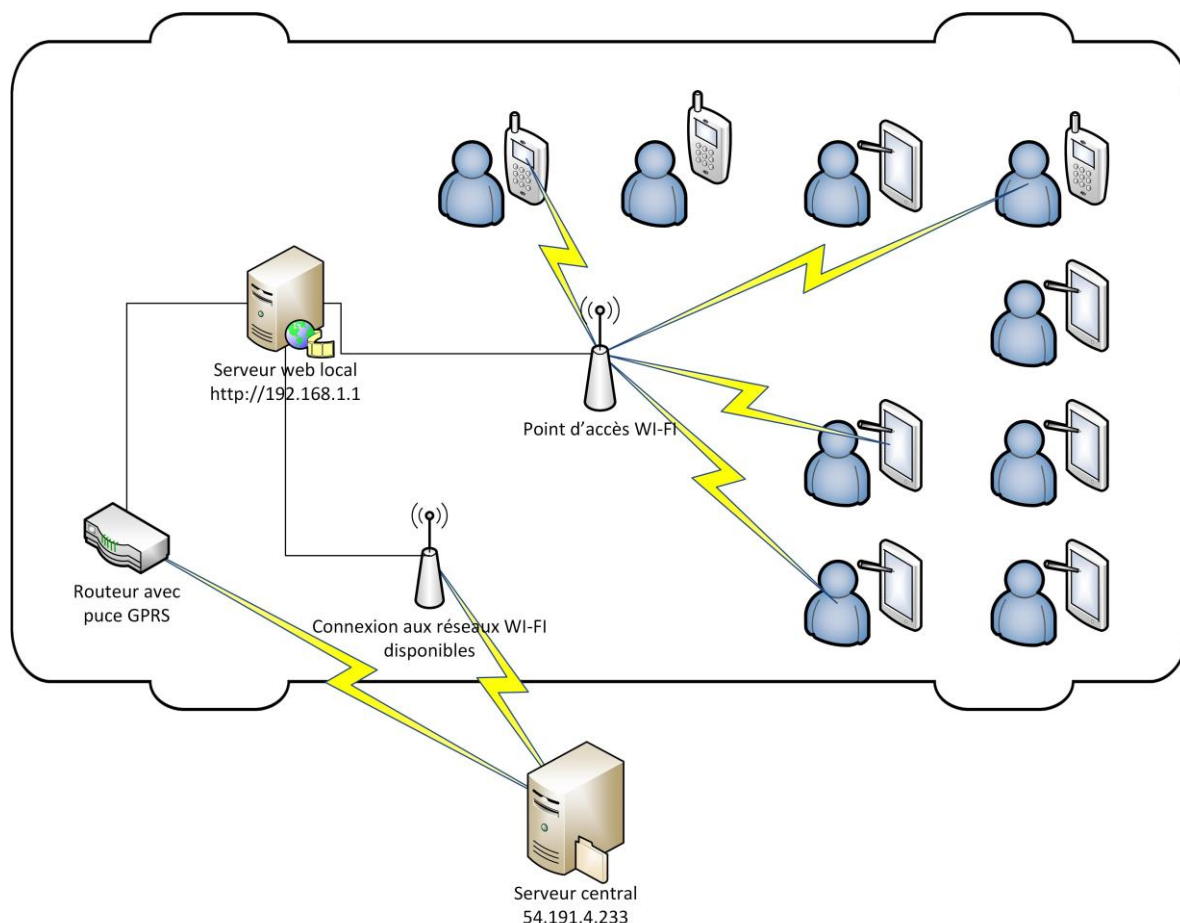
<sup>11</sup> Source : <http://wiki.nginx.org/NginxFr>, visité en juin 2014

<sup>12</sup> Source : <http://dev.mysql.com/>, visité en juin 2014

## 4.2. PRÉSENTATION DES COMPOSANTS DE LA PLATEFORME

La plateforme est composée de différents composants hardware. La **Figure 9** ci-après donne une vision globale de ces composants et de leur fonctionnement.

**Figure 9** : Vue globale des composants hardware de la plateforme



Le **serveur central** contient la dernière version du code source de la plateforme. Il permet au responsable de la maintenir et d'y ajouter du nouveau contenu. Actuellement le serveur est hébergé sur Amazon EC2<sup>13</sup> sur un compte personnel. Il possède une adresse IP fixe lui permettant d'être accessible en tout temps.

A l'intérieur du car se trouve un **serveur local** contenant la plateforme et son contenu. Sa fonction première est de créer un réseau WI-Fi accessible par les passagers. Le serveur est également relié au routeur du car lui permettant d'avoir une connexion à internet.

<sup>13</sup> Amazon EC2, service web permettant de louer des serveurs, <http://aws.amazon.com/fr/ec2/>, visité en juin 2014

Cette connexion est nécessaire pour que la plateforme puisse se mettre à jour depuis le **serveur central**, mais n'est pas indispensable à son bon fonctionnement puisque tout son contenu est stocké localement. Étant donné que la bande passante est parfois faible ou inexistante, le **serveur local** est également capable d'utiliser des réseaux WI-FI se trouvant sur le parcours du car afin de se mettre à jour.

Les passagers accèdent au contenu de la plateforme depuis leurs **appareils mobiles** en se connectant au WI-FI fourni par le **serveur local** et en se rendant depuis un navigateur web sur la page d'accueil de la plateforme (<http://192.168.1.1>) permettant d'accéder aux différents services.

## 4.3. ARCHITECTURE LOGICIELLE

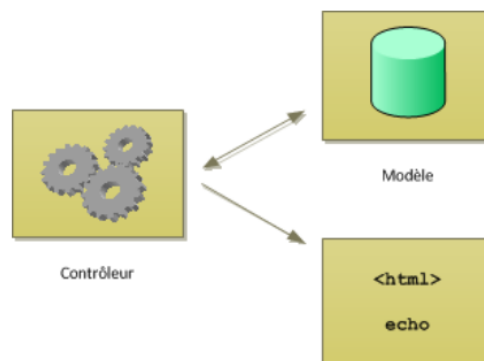
### 4.3.1. Modèle MVC

L'architecture logicielle de la plateforme est basée sur le modèle d'architecture MVC (cf. **Figure 10**), qui signifie Modèle – Vue – Contrôleur. Ce modèle a pour objectif de mieux organiser le code source du système en le séparant en trois parties distinctes :

- La partie « **Modèle** » gère l'accès aux données du système. Son rôle est de récupérer, traiter et transmettre les données au contrôleur.
- La partie « **Vue** » est responsable de l'affichage. Elle se contente d'afficher les données envoyées par le contrôleur. Elle est essentiellement composée de code HTML et de quelques fonctions PHP.
- La partie « **Contrôleur** » est le cerveau du système. Elle gère la logique du code, elle va récupérer les données du modèle, les analyser et transmettre le texte à afficher à la vue. Elle va également gérer les interactions avec l'utilisateur.

L'intérêt d'un tel système est de permettre une meilleure visibilité du code. La maintenance et l'ajout de fonctionnalités sont grandement facilités. De plus, la modularité offerte par cette architecture permet de modifier une partie de code sans affecter l'ensemble de la plateforme. Ce dernier avantage correspond bien aux spécificités de ce travail, car il sera facile d'ajouter de nouveaux services par la suite.

**Figure 10 :**  
Le modèle d'architecture MVC



Source : <http://fr.openclassrooms.com/informatique/cours/concevez-votre-site-web-avec-php-et-mysql/organiser-son-code-selon-l-architecture-mvc> , juin 2014

### 4.3.2. Organisation du code source

La figure suivante (cf. **Figure 11**) représente l'organisation des différents fichiers et dossiers du code source de la plateforme.

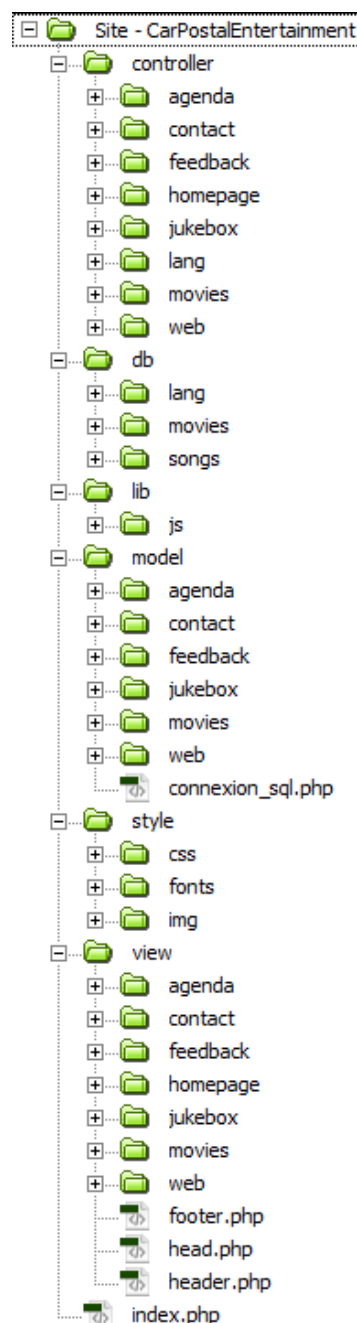
Chaque service, à l'exception de la page d'accueil, possède son propre contrôleur, son propre modèle et sa propre vue. Afin de gagner en visibilité tous les contrôleurs sont placés dans un même dossier (*controller*). Ce principe s'applique également aux modèles (*model*) et aux vues (*view*).

Le dossier *db* contient le contenu de la plateforme tel que les vidéos et les musiques, ainsi que les fichiers contenant les traductions.

Le dossier *lib* rassemble les librairies JavaScript utilisées. Alors que le dossier *style* contient la feuille de style CSS, les polices d'écritures ainsi que des images.

Le fichier *index.php* présent à la racine du site joue le rôle de contrôleur principal. Il contient les informations de connexion à la base de données et choisit le bon contrôleur pour afficher la page demandée par l'utilisateur.

**Figure 11 :** Organisation du code source de la plateforme





## 4.4. HARDWARE

Le serveur web local présent à l'intérieur du car est un Raspberry Pi<sup>14</sup> (cf. **Figure 12**). Le Raspberry Pi est un nano-ordinateur, de la taille d'une carte de crédit, à processeur ARM qui permet l'exécution de plusieurs distributions du système d'exploitation Linux. Il a été choisi pour ce projet car il possède des qualités non négligeables.

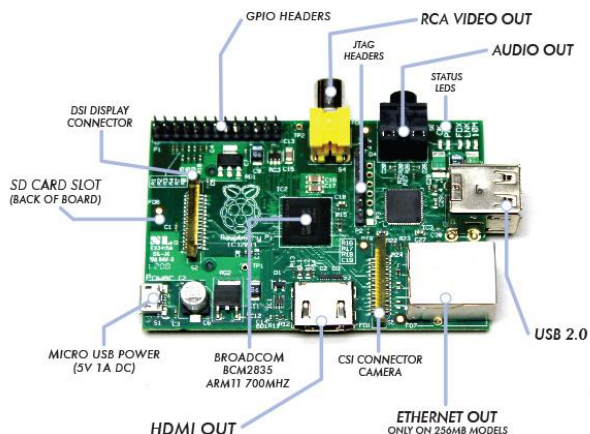
### 4.4.1. Avantages

Pour commencer il est de taille très petite et de ce fait peut être utilisé n'importe où. Sa petite taille ainsi que les composants qu'il embarque impliquent également que sa consommation électrique est faible. Il est recommandé de lui fournir du 500 mA sur 5 v, ce qui représente 2.5 Watts. Une utilisation de 10 heures par jour pendant une année représente une consommation d'environ 9 KWh. De plus, il n'embarque pas de ventilateur, ce qui le rend parfaitement silencieux.

Le disque dur du Raspberry Pi étant une carte mémoire, il est possible de facilement cloner tout le système sur d'autres cartes ou de remplacer une carte défectueuse. Le déploiement de la plateforme dans d'autres cars est donc facilité. Par ailleurs, il suffit de simplement changer la carte mémoire, afin de lui donner d'autres fonctionnalités.

Le Raspberry Pi possède de nombreux connecteurs. Ils permettent le branchement de clés WI-FI via les prises USB, d'un câble réseau via la prise Ethernet, ainsi que l'ajout d'un UPS via la prise GPIO nécessaires au fonctionnement de la plateforme.

**Figure 12 :**  
Le Raspberry Pi, Type B



Source : <http://www.pcmag.com/article2/0,2817,2407058,00.asp>, juin 2014

<sup>14</sup> Source : <http://www.raspberrypi.org/help/what-is-a-raspberry-pi/>, visité en juin 2014

Une énorme communauté de développeurs est présente autour de cet ordinateur. Ce qui fournit du support et de l'aide en cas de problème et qui prouve que le système est fonctionnel et fiable. Il existe de nombreux modules et accessoires pour le personnaliser selon les besoins.

Finalement un dernier avantage est son prix d'achat. Un Raspberry Pi Type B coûte environ CHF 40.-<sup>15</sup>, auquel il faut ajouter une boîte pour le protéger (CHF 15.-) et une carte mémoire de 8GB minimum (CHF 12.-). Dans le cadre de ce projet deux clés WI-FI (CHF 28.-), ainsi qu'un UPS (CHF 25.-) sont également nécessaires. Pour environ de CHF 120.- un serveur web local peut être réalisé.

#### 4.4.2. Inconvénients

Le Raspberry Pi possède néanmoins quelques petits inconvénients, à commencer par le fait qu'il fonctionne avec un processeur ARM. Ce type de processeur, malgré une consommation très faible, possède un déficit de puissance comparé à ces concurrents. Par ailleurs, tous les composants du Raspberry Pi sont directement soudés à la carte, il n'est donc pas possible de rajouter de la mémoire RAM.

Toutefois ces inconvénients ne devraient pas gêner son fonctionnement. Le Raspberry Pi va être installé à l'intérieur d'un car en tant que server web. Il a les capacités pour gérer 50 connexions simultanées. De plus, il est possible d'overclocker le processeur afin d'en améliorer ses performances en cas de nécessité.

#### 4.4.3. Systèmes concurrents

Il existe sur le marché d'autres mini-ordinateurs se rapprochant du Raspberry Pi, tel que le Cubieboard<sup>16</sup> et le BeagleBoard<sup>17</sup>. Cependant à ce jour, le Raspberry Pi reste le moins cher et le plus évolué grâce à l'énorme communauté qui l'entoure.

<sup>15</sup> Source : <http://www.pi-shop.ch/raspberry-pi>, visité en juin 2014

<sup>16</sup> Source : <http://cubieboard.org/>, visité en juillet 2014

<sup>17</sup> Source : <http://beagleboard.org/>, visité en juillet 2014

#### 4.4.4. Conclusion

Le Raspberry Pi apparaît comme le système parfait pour la création d'un premier prototype de plateforme. Il possède toutes les caractéristiques nécessaires à sa réalisation et ne nécessite pas un gros investissement financier.

### 4.5. TECHNOLOGIES UTILISÉES

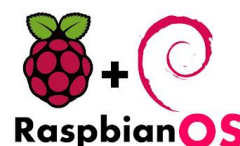
Ce chapitre énumère les différentes technologies utilisées pour la réalisation de cette plateforme. Dans un premier temps, les technologies utilisées pour la configuration du serveur web local sont décrites, puis par la suite, c'est au tour des technologies utilisées au niveau de la plateforme web et des services.

#### 4.5.1. Pour la configuration du serveur web local

Comme tous les ordinateurs le Raspberry Pi a besoin d'un système d'exploitation pour fonctionner. Étant donné les caractéristiques qui le définissent, un système d'exploitation spécialement adapté a été développé, il s'agit de Raspbian.

Raspbian est un système d'exploitation libre basé sur Linux/Debian<sup>18</sup> et optimisé pour fonctionner sur un Raspberry Pi. Ce système d'exploitation a été sélectionné dans le cadre de ce travail étant donné qu'il a été créé pour fonctionner sur un système possédant des ressources limitées. De plus, il est libre et très documenté sur le web, ce qui en fait une bonne solution.

**Figure 13 : Raspbian**



Toujours dans un souci d'optimisation des performances, la configuration du Raspberry Pi et de ces différentes fonctionnalités a été réalisée en ligne de commande sans environnement graphique afin de limiter les ressources utilisées.

Par ailleurs, des scripts batch ont été écrits afin de répondre à certains besoins de la plateforme.

<sup>18</sup> Linux/Debian : système d'exploitation libre, <https://www.debian.org/index.fr.html>, juillet 2014

#### 4.5.2. Pour le développement de la plateforme

Plusieurs technologies de développement ont été utilisées pour la réalisation de la plateforme, à commencer par le PHP. Ce langage de programmation open source est énormément documenté sur internet et relativement facile d'accès. Pour fonctionner il nécessite un module PHP sur le serveur web.

La plateforme doit être développée selon le principe de l'expérience utilisateur (*UX Design*). De ce fait il est important que les pages web s'affichent correctement sur tout type d'écran, que ce soit sur smartphone, tablette ou ordinateur portable. Grâce à l'HTML et aux feuilles de style CSS, il est possible d'utiliser des Media Queries qui permettent justement de le faire. Les feuilles de style CSS permettent également la mise en forme des pages web. Grâce à ces feuilles de style la mise en forme est séparée de la structure HTML d'une page, ce qui a pour avantage de réduire la complexité d'un document et de pouvoir modifier plus facilement le design d'une page.

Pour augmenter l'expérience utilisateur des langages tels que le JavaScript et l'Ajax ont également étaient utilisés. Ils permettent de modifier dynamiquement le contenu d'une page sans devoir recharger celle-ci. Afin de faciliter la manipulation de ces deux langages la librairie jQuery a été utilisée.

#### 4.6. CONFIGURATION DU SERVEUR CENTRAL

Comme expliqué précédemment, le serveur central possède la dernière version de la plateforme et de son contenu. Le serveur local s'y connecte afin de se mettre à jour.

La configuration de ce serveur central est rapide. Il doit posséder une adresse IP fixe afin d'être accessible par le serveur local à travers internet. Un serveur FTP doit être installé pour que son contenu soit accessible.

Pour plus de précision un tutoriel d'installation et de configuration de la plateforme est disponible en annexe (cf. **Annexe IX**). Ce tutoriel comprend l'ensemble des étapes décrites de manière détaillées et illustrées à l'aide de capture d'écran.

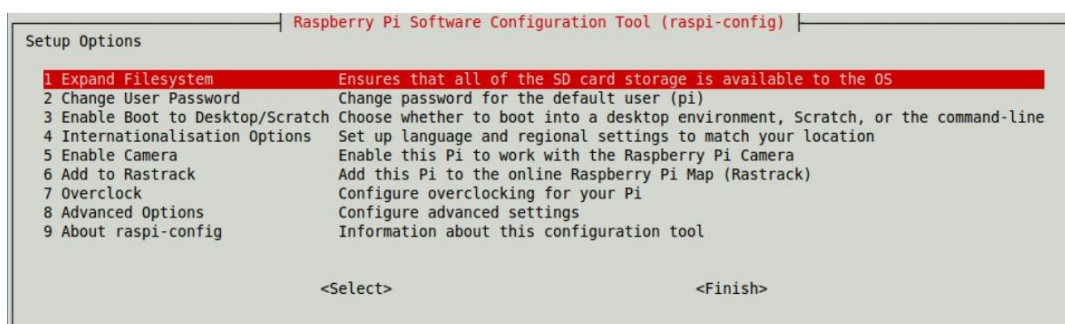
## 4.7. CONFIGURATION DU SERVEUR WEB LOCAL (RASPBERRY PI)

Dans ce chapitre les différentes étapes de configuration du serveur web local sont résumées, pour plus de précision se référer au tutoriel (cf. **Annexe IX**).

Pour commencer le système d'exploitation *Raspbian* doit être installé afin que le serveur puisse fonctionner. Une des particularités du Raspberry Pi est le fait que son disque dur est une carte mémoire. Le système d'exploitation doit être écrit sur la carte mémoire depuis un ordinateur.

Une fois le système d'exploitation installé et les différents câbles branchés, le Raspberry Pi est prêt à être configuré. Dans un premier temps des configurations de base, tel que la modification du type de clavier et l'activation du SSH, sont effectuées. Un utilitaire propre à Raspbian permet de le faire facilement (cf. **Figure 14**).

**Figure 14 : Raspi-config**



La plateforme nécessite trois interfaces réseaux afin de fonctionner : une interface câblée (**eth0**) et une interface WI-FI (**wlan1**) utilisées pour mettre à jour le serveur web, ainsi qu'une autre interface WI-FI (**wlan0**) servant de point d'accès à l'intérieur du car. L'interface **eth0** ne nécessite pas de configuration particulière, le branchement d'un câble réseau suffit à la faire fonctionner.

Le Raspberry Pi ne possède pas de carte WI-FI embarquée, par contre il possède deux ports USB où il est possible de brancher des clés WI-FI. Certaines clés, tel que l'**Edimax EW-7811Un**<sup>19</sup>, sont directement reconnues par le Raspberry Pi, alors que d'autres ne le sont pas

<sup>19</sup> Clé WI-FI compatible avec un Raspberry Pi et pouvant faire office de point d'accès WI-FI, [http://www.edimax.com/en/produce\\_detail.php?pd\\_id=347](http://www.edimax.com/en/produce_detail.php?pd_id=347), consulté en juillet 2014

et nécessitent l'installation d'un pilote spécifique. La prochaine étape est la configuration de ces deux interfaces.

L'interface **wlan0** fournit un point d'accès à la plateforme à l'intérieur du car. Il est donc nécessaire de lui attribuer une adresse IP fixe et d'installer un serveur DHCP qui va fournir des adresses IP aux passagers leur permettant d'accéder à la plateforme. Afin que l'interface **wlan1** puisse se connecter à des réseaux WI-FI pour avoir un accès à internet, il faut enregistrer les caractéristiques des réseaux dans un fichier. Si le réseau WI-FI est inconnu, **wlan1** ne pourra pas se connecter.

Une fois les interfaces configurées, le serveur web **Nginx** et le système de base de données **MySQL** sont installés afin de pouvoir accueillir la plateforme ainsi que son contenu.

Il reste encore le système de mise à jour à configurer. Ce système est constitué de plusieurs scripts batch s'exécutant à des moments spécifiques. Le **Chapitre 4.8.4 Mises à jour du contenu** fournit plus de détails à ce sujet.

Enfin, afin d'éviter de corrompre la carte mémoire et donc tout le système, l'installation d'un UPS a été nécessaire. Cet UPS permet de gérer l'alimentation du Raspberry Pi. En cas de coupure du courant il va continuer à alimenter le Raspberry Pi afin de lui permettre de s'arrêter proprement.

## 4.8. DESCRIPTIFS DES FONCTIONNALITÉS DE LA PLATEFORME

Ce chapitre explique les différentes fonctionnalités de base de la plateforme. Les services seront décrits dans un prochain chapitre.

### 4.8.1. Page d'accueil

La page d'accueil du site est accessible à l'adresse <http://192.168.1.1>. C'est le point d'entrée de la plateforme. Un texte d'introduction expliquant les caractéristiques principales de cette dernière y est affiché. Elle permet également d'accéder aux différents services offerts aux passagers et de changer de langue.

**Figure 15 :** Page d'accueil en français de la plateforme vue depuis un smartphone



#### 4.8.2. Multilinguisme

La plateforme intègre un système de gestion des langues étant donné qu'elle est destinée aux passagers du car qui sont souvent des touristes. Actuellement trois langues sont gérées (le Français, l'Allemand et l'Anglais), mais il est possible d'en ajouter d'autres.

Lorsqu'un utilisateur se connecte à la plateforme, le système vérifie si dans la session de l'utilisateur une langue a déjà été choisie lors d'une précédente visite. Si ce n'est pas le cas le français est choisi par défaut. Par la suite l'utilisateur a la possibilité de changer de langue en cliquant sur le bouton correspondant (cf. **Figure 15**).

A chaque changement de langue, celle-ci est enregistrée en session et un fichier contenant les traductions dans la langue sélectionnée est chargé. Ce mécanisme permet au système d'afficher les textes de la page dans la bonne langue.

#### 4.8.3. Responsive Web Design

Afin d'améliorer l'expérience utilisateur, le concept de Responsive Web Design a été utilisé. Ce concept regroupe des principes et des technologies permettant de concevoir des sites offrant une expérience de consultation optimale à l'utilisateur. En d'autres termes, le passager du car peut accéder à la plateforme depuis son smartphone, sa tablette ou son ordinateur portable de manière optimale.

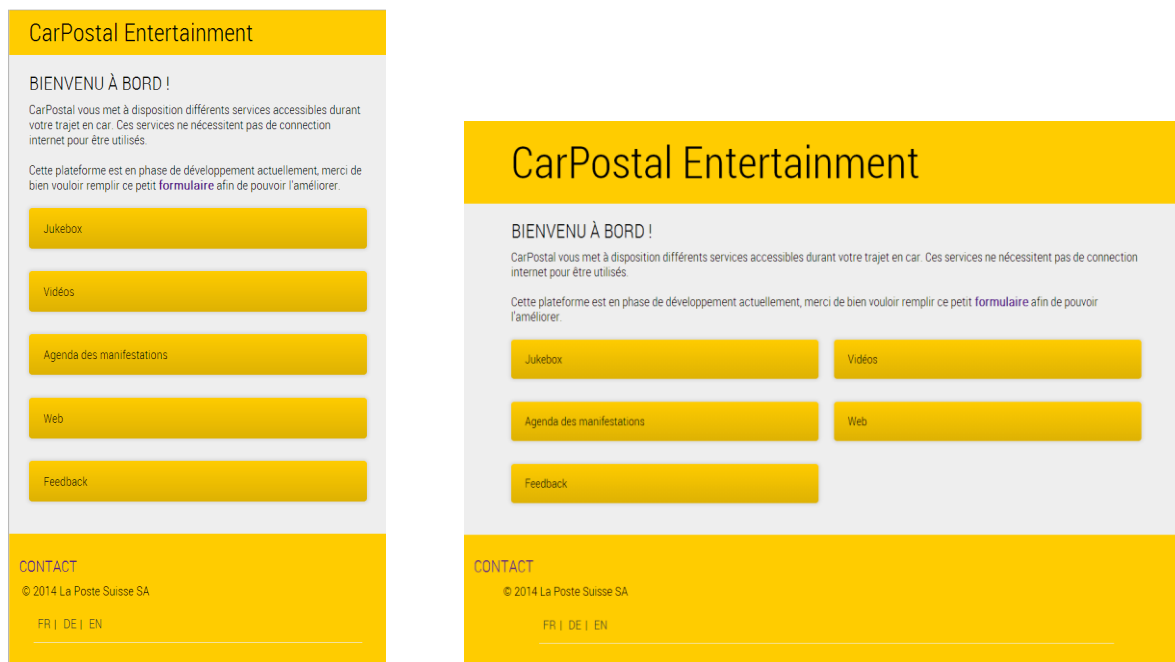
Le principal avantage réside dans le fait de créer une seule version de la plateforme qui va s'adapter en fonction de l'appareil qui s'y connecte. Le contenu sera affiché en tenant compte de la taille de l'écran afin de garder un design ergonomique et épuré. Pour illustrer ces propos la **Figure 16** représente l'affichage de la page d'accueil de la plateforme en version mobile à gauche et en version tablette à droite.

La plateforme est destinée à être utilisée par des appareils mobiles. Dans cette optique elle a été créée selon le principe du *Mobile First*<sup>20</sup> qui a pour but de privilégier la construction d'une interface ergonomique pour les appareils mobiles avant de réfléchir à l'interface destinée aux ordinateurs.

---

<sup>20</sup> Traduction : le mobile en premier



**Figure 16 : Illustration du principe de Responsive Web Design**

#### 4.8.4. Mises à jour du contenu

Actuellement la connexion internet via GPRS à l'intérieur des cars offre une bande passante très variable. La connexion varie énormément durant le trajet et n'est pas toujours disponible. Cependant un accès à internet est indispensable afin de pouvoir mettre à jour le contenu de la plateforme. Pour répondre à cette problématique un système a été développé.

Ce système permet au serveur du car de se connecter à internet soit en utilisant la connexion GPRS présente dans le car, soit en utilisant les réseaux WI-FI des bureaux de Poste lors des arrêts du car le long de son parcours.

Un script batch a été créé afin de déterminer quelle connexion utiliser pour réaliser la mise à jour. Il va dans un premier temps analyser si une connexion est disponible. S'il en trouve une il va lancer la procédure de mise à jour, autrement il va essayer de rafraîchir la connexion via GPRS. Si la connexion n'est pas disponible il va essayer de trouver un réseau WI-FI. Si aucun réseau n'a été trouvé le script s'arrête et la procédure ne sera pas lancée.

La procédure est composée d'un ensemble de scripts batch ayant chacun une fonction spécifique. Pour commencer un script mettant à jour le contenu du serveur local est lancé. Il va synchroniser le serveur local avec le serveur central afin de déterminer si du nouveau contenu est disponible.

**Figure 17** : Extrait du script permettant la mise à jour du contenu

```
rsync -avvvz --progress --partial-dir=".temp" -e "ssh -i  
/home/pi/FTPServer2.pem" admin@54.191.4.233:/home/admin/C  
arPostalEntertainment /var/www
```

La synchronisation s'effectue grâce au logiciel libre *rsync*<sup>21</sup>. Ce logiciel va contrôler dans le répertoire de réception si le fichier existe déjà et n'a pas été modifié. Seulement les nouveaux fichiers ou les fichiers modifiés sont synchronisés, ce qui permet d'accélérer significativement le processus. De plus, le protocole de communication SSH est utilisé afin de crypter et d'authentifier la source des données envoyées. Dans ce cas *rsync* utilise une clé publique afin de s'authentifier auprès du serveur FTP (cf. **Figure 17**).

Une fois la synchronisation du contenu terminée c'est au tour de la base de données de se mettre à jour. Pour se faire un script va exécuter le fichier mis à jour lors de l'étape précédente contenant la dernière version de la base de données (cf. **Figure 18**).

**Figure 18** : Extrait du script permettant la mise à jour de la base de données

```
mysql -h localhost -u root -pD4Ts24Ds5 CarPostalEntertain  
mentDB < /var/www/CarPostalEntertainment/db/content.sql
```

La fréquence d'exécution de la procédure de mise à jour est définie dans une table de logiciel *cron*<sup>22</sup>, un outil Linux permettant justement de définir à quel moment le script doit être exécuté (cf. **Figure 19**). Actuellement, étant donné les difficultés à obtenir une connexion à internet, la procédure est exécutée lors du démarrage du serveur et chaque cinq minutes par la suite.

<sup>21</sup> Rsync, programme de synchronisation de dossiers et de fichiers,  
<http://www.delafond.org/traducmanfr/man/man1/rsync.1.html>, juillet 2014

<sup>22</sup> Cron, programme d'exécution automatique de scripts,  
<http://fr.openclassrooms.com/informatique/cours/reprenez-le-controle-a-l-aide-de-linux/executer-un-programme-a-une-heure-differee>, juillet 2014

**Figure 19** : Aperçu de l'enregistrement dans la table du logiciel « cron »

```
# Execute main.sh at start up and every 5 min
@reboot bash /usr/local/bin/main.sh
*/5 * * * * bash /usr/local/bin/main.sh
```

Le moment d'exécution de la procédure est facilement personnalisable. Si la connexion à internet est améliorée dans les cars à l'avenir, il faudra diminuer la fréquence d'exécution. Il est également possible de lancer la procédure en se basant sur les horaires du car afin de déterminer à quel moment un réseau WI-FI est disponible.

#### 4.8.5. Formulaire de contact

Un formulaire permet aux utilisateurs de transmettre leurs commentaires. Afin de l'afficher ou de le masquer il suffit de cliquer sur le bouton *Contact* affiché sur chaque page de la plateforme. L'analyse de ces commentaires permet de constamment améliorer la plateforme et par la même occasion d'améliorer l'expérience utilisateur.

Une fois envoyés les commentaires sont stockés dans la table *comments* de la base de données du serveur local. La table enregistre le nom, l'email, le message ainsi que la date et l'heure de l'envoi.

**Figure 20** : Formulaire de contact

Le formulaire de contact est présenté sur un fond jaune. Il contient les éléments suivants :

- Un titre "CONTACT" en haut à gauche.
- Un champ de texte pour "VOTRE NOM".
- Un champ de texte pour "VOTRE EMAIL".
- Un champ de texte plus grand pour "VOTRE MESSAGE\*", avec une icône de crayon à droite.
- Un bouton bleu arrondi en bas à droite avec l'inscription "Envoyer".

Par la suite un script exporte les commentaires de la base de données et les transmet au serveur central en suivant la même procédure qu'au chapitre précédent. Le script est également exécuté au démarrage du serveur et chaque cinq minutes par la suite.

## 5. DÉVELOPPEMENT DES SERVICES

Une fois la plateforme terminée, quelques services ont été développés afin d'analyser son potentiel. Actuellement trois services principaux sont disponibles ; un jukebox, un lecteur de vidéos et un agenda des manifestations. Un autre petit service permet de voir si une connexion internet est disponible et affiche plusieurs liens vers des sites internet. De plus, durant la phase de développement et de test un questionnaire a été ajouté afin de collecter les feedbacks des utilisateurs. Des captures d'écran de ces différents services sont disponibles en annexe (cf. **Annexe V**).

### 5.1. JUKEBOX

Le jukebox permet d'écouter les différentes musiques stockées sur la plateforme. Les informations relatives à ces musiques sont enregistrées dans la table *songs* de la base de données (cf. **Figure 21**).

La table enregistre dans l'ordre un identifiant unique, le titre, l'artiste, le genre de musique, la date de création, la date d'ajout à la base de données et le chemin d'accès. Le fichier audio n'est pas stocké physiquement dans la base de données, seulement sont chemin d'accès l'est. Ce procédé permet de limiter grandement la taille de la base de données. Le chemin d'accès va permettre de retrouver le fichier audio afin de l'afficher à l'utilisateur.

**Figure 21** : Aperçu de la table *songs*

Field	Type
id	int(11)
title	varchar(50)
artist	varchar(50)
genre	varchar(50)
dateCreation	date
dateAdded	date
path	varchar(50)

Pour ajouter de nouvelles musiques, aucune modification du code source n'est nécessaire. Il suffit de les transférer vers le serveur web local et de les insérer dans la base de données. Par la suite, les musiques enregistrées dans la base de données sont récupérées afin d'être affichées à l'utilisateur.

Au niveau technique, le fonctionnement est le suivant : le contrôleur responsable du jukebox appelle son modèle qui va récupérer les musiques enregistrées dans la base de données et les triées par date d'ajout, puis la vue est appelée afin de les afficher à l'utilisateur.

## 5.2. LECTEUR DE VIDÉOS

Le lecteur de vidéos permet de visionner les différentes vidéos stockées sur la plateforme. Les informations relatives aux vidéos sont enregistrées dans la table *movies* (cf. **Figure 22**).

La table enregistre dans l'ordre un identifiant unique, le titre, la description, la source, la date de création, la date d'ajout à la base de données, le langage de la vidéo, le chemin d'accès, une image représentant la vidéo et le créateur.

Pour l'affichage des vidéos, ainsi que l'ajout de nouvelles, le lecteur de vidéos fonctionne exactement de la même manière que le jukebox.

Au niveau technique, c'est le contrôleur responsable du lecteur de vidéos qui effectuera les différentes actions.

## 5.3. AGENDA DES MANIFESTATIONS

Ce service permet d'afficher les manifestations d'une région. Ces manifestations sont stockées dans la table *events* (cf. **Figure 23**).

La table enregistre dans l'ordre un identifiant unique, le titre, la description, la date, le lieu, le nom de contact, son téléphone, son email, et son site internet.

Les manifestations enregistrées sont récupérées et affichées chronologiquement par mois. Dans un premier temps le jour, le titre et le lieu sont affichés. Par la suite l'utilisateur a la possibilité d'appuyer sur la manifestation qui l'intéresse afin d'afficher plus de détails. A noter que les manifestations déjà passées ne sont pas affichées.

**Figure 22** : Aperçu de la table *movies*

Field	Type
id	int(11)
title	varchar(50)
description	text
source	varchar(50)
dateCreation	date
dateAdded	date
language	varchar(5)
path	varchar(50)
poster_path	varchar(50)
creator	varchar(50)

**Figure 23** : Aperçu de la table *events*

Field	Type
id	int(11)
title	varchar(255)
description	text
date	date
location	varchar(50)
contact_name	varchar(50)
contact_tel	varchar(13)
contact_email	varchar(50)
contact_website	varchar(50)

Au niveau technique, le contrôleur responsable de l'agenda appelle le modèle qui va récupérer les différentes manifestations et les trier chronologiquement, puis la vue est appelée afin de les afficher.

#### 5.4. CONTRÔLE DE LA CONNEXION INTERNET

Ce service permet à l'utilisateur de savoir si une connexion à internet est disponible ou non. Si une connexion est disponible des liens vers des sites internet vont s'afficher, autrement un message explicatif est affiché.

Au niveau technique, le contrôleur responsable va appeler son modèle qui va déterminer si une connexion est disponible, puis la vue est appelée afin d'afficher le résultat.

#### 5.5. QUESTIONNAIRE

Afin de collecter les feedbacks des passagers du car, un formulaire a été réalisé. Il est accessible sous la rubrique *Feedback* et est disponible en français, allemand et anglais. Les questions suivantes y sont posées :

- Quelle est l'utilité d'une telle plateforme ? (0 à 10)
- Quels sont les éléments à améliorer ?
- Quels services aimeriez-vous avoir ?
- Est-ce que vous seriez intéressé par une carte du trajet interactive indiquant les différents arrêts et les points d'intérêts à proximité ?
- Autres commentaires

Les réponses sont stockées dans la table *feedbacks* de la base de données (cf. **Figure 24**). La table enregistre les réponses dans l'ordre des questions et ajoute un identifiant unique ainsi que la date. Par ailleurs, le champ *q2* n'est pas utilisé car la question relative a été supprimée du questionnaire.

Au niveau technique, le contrôleur responsable va récupérer les données saisies par l'utilisateur depuis la vue et les transmettre au modèle afin qu'il les enregistre dans la base de données.

**Figure 24** : Aperçu de la table *feedbacks*

Field	Type
id	int(11)
date	datetime
q1	int(2)
q2	int(1)
q3	text
q4	text
q5	int(1)
q6	text

## 6. TESTS DE LA PLATEFORME

Une fois le développement terminé, différents types de tests ont été réalisés sur la plateforme afin d'en valider son fonctionnement.

Pour commencer le Raspberry Pi a subi des tests de charge (Load Test) afin de simuler l'utilisation simultanée de la plateforme par plusieurs utilisateurs et de déterminer ses capacités maximales.

Par la suite différents autres tests liés à des coupures d'alimentation ou de connexion internet ont été effectués. Ces tests ont permis d'analyser le fonctionnement de la plateforme dans plusieurs états d'utilisation et de révéler de potentiels problèmes.

Pour finir la plateforme a été installée dans un car afin de tester son fonctionnement en environnement réel. Durant cette phase les passagers ont eu la possibilité de répondre à un questionnaire afin de transmettre leur feedback.

### 6.1. TESTS DE CHARGE

Les ressources du Raspberry Pi étant limitées, des tests de performance ont été réalisés afin d'analyser et de valider l'utilisation de ce dernier en tant que serveur web local. Ces tests ont été réalisés avec *Gatling*<sup>23</sup> un outil open source permettant d'analyser et de mesurer les performances d'un site internet. Cet outil crée des tests de charge et affiche les résultats sous forme de tableaux et graphiques.

**Figure 25 :** Logo de Gatling



Source : <http://gatling-tool.org/>, juillet 2014

En augmentant le nombre d'utilisateurs et donc de sessions ouvertes sur la plateforme, il est possible de mesurer le délai que met une page à s'afficher. Ce délai doit être le plus court possible étant donné qu'une personne attend une à deux secondes maximum le chargement d'une page avant d'abandonner et d'en ouvrir une autre.

<sup>23</sup> Gatling, outil d'analyse et de mesure de performances, source : <http://gatling-tool.org/>, visité en juillet 2014

### 6.1.1. Création d'un scénario d'utilisation

Avant de créer le scénario il a fallu sélectionner les fonctionnalités les plus souvent utilisées par les utilisateurs et demandant beaucoup de ressources. Le choix s'est porté sur le visionnage de vidéos et l'écoute de musiques.

De plus, afin que le scénario puisse être validé, des buts à atteindre ont été définis. Pour que la plateforme passe le test, au minimum 50 utilisateurs doivent pouvoir l'utiliser simultanément et le temps de latence (*response time*) pour l'affichage d'une page ne doit pas dépasser 1.5 seconde.

Cette analyse a permis de créer le scénario suivant (cf. **Figure 26**) : l'utilisateur ouvre la page d'accueil du site (1) et change la langue d'affichage (2). Puis il ouvre la page des vidéos et lance trois vidéos (3). Après 15 secondes il retourne sur la page d'accueil (4). Il ouvre la page du jukebox et lance trois musiques (5). Enfin, il retourne sur la page d'accueil après 15 secondes (6).

**Figure 26** : Gatling – scénario de test

```
val scn = scenario("Open movies and songs")
1 .exec(http("Go to the home page")
    .get(""/""))
    .pause(5)
2 .exec(http("Change the language")
    .get(""/""))
    .pause(5)
3 .exec(http("Watch some movies")
    .get("/movies.php"))
    .pause(15)
4 .exec(http("Go back to the home page")
    .get("/index.php"))
    .pause(5)
5 .exec(http("Listen to some songs")
    .get("/audio.php"))
    .pause(15)
6 .exec(http("Go back to the home page")
    .get("/index.php"))
```

Ce scénario va être exécuté par Gatling en fonction du nombre d'utilisateurs choisis. De plus, Gatling permet de définir une certaine plage durant laquelle les utilisateurs vont exécuter une étape du scénario. Par exemple si le test est réalisé avec 100 utilisateurs et la plage est définie à 20 secondes, cinq utilisateurs par seconde (100 / 20) effectueront l'étape du scénario. Cette fonctionnalité permet de mieux simuler le fonctionnement en environnement réel, car tous les utilisateurs ne se connectent pas en même temps. Dans le cadre de ce projet, tous les tests ont tous été exécutés avec une plage de 20 secondes.



### 6.1.2. Test 1 : 50 utilisateurs

Pour commencer, un test avec 50 utilisateurs a été lancé cinq fois afin d'assurer la cohérence des résultats. Les données du test (cf. **Annexe VI**) avec les moins bons résultats ont été prises en compte.

Le tableau suivant (cf. **Tableau 2**) représente une partie de ces résultats. Chaque ligne représente une étape du scénario avec sa désignation, les requêtes effectuées et le temps de latence en milliseconde. La colonne représentant le temps de latence maximum (encadrée en vert) démontre que ce temps est inférieur à la limite de 1.5 seconde (1500 millisecondes) fixée précédemment quelle que soit l'étape. Ce qui démontre que la plateforme peut être utilisée simultanément par au moins 50 personnes.

**Tableau 2** : Récapitulatif du résultat du Load Test réalisé avec 50 utilisateurs

STATISTICS <span>Expand all groups   Collapse all groups</span>												
Requests ^	Executions				Response Time (ms)							Req/s ^
	Total ^	OK ^	KO ^	% KO ^	Min ^	Max ^	Mean ^	Std Dev ^	95th pct ^	99th pct ^		
Global Information	300	300	0	0 %	10	1150	40	107	50	650		5
Go to the home page	50	50	0	0 %	20	430	42	64	170	430		1
Change the language	50	50	0	0 %	10	60	22	9	30	60		1
Watch some movies	50	50	0	0 %	10	50	21	4	30	50		1
Go back to the home page	100	100	0	0 %	10	860	43	112	40	650		2
Listen to some songs	50	50	0	0 %	10	1150	66	194	380	1150		1

### 6.1.3. Test 2 : 1'000 utilisateurs

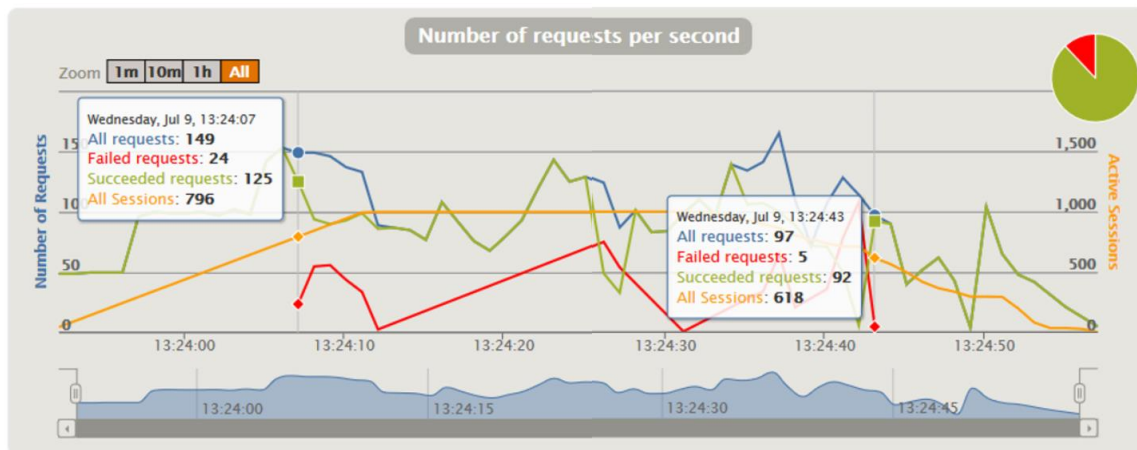
Ce deuxième test a été réalisé avec 1'000 utilisateurs afin de déterminer le nombre d'utilisateurs maximal que la plateforme peut accueillir avant de rencontrer des erreurs. Comme pour le premier test, il a été lancé cinq fois et les moins bons résultats ont été pris en compte (cf. **Annexe VII**).

Le graphique suivant (cf. **Figure 27**) montre l'évolution du nombre d'utilisateurs connectés (en orange) durant l'avancement du scénario, ainsi que l'état des différentes requêtes effectuées.

Jusqu'aux alentours de 800 sessions actives, toutes les requêtes réussissent. Cependant, passé ce niveau, des requêtes échouées commencent à apparaître. Ce qui montre les limites de la plateforme. Durant le déroulement du test l'utilisation de la plateforme était difficile, des

pages d'erreurs (502 Passerelle incorrecte) sont apparues. Dès que le nombre de sessions actives diminuent aux alentours de 600, le nombre de requêtes échouées diminuent également.

**Figure 27 :** Graphique résumant le Load Test réalisé avec 1'000 utilisateurs



Le tableau suivant (cf. **Tableau 3**) démontre que les temps de réponse moyens et maximaux ont grandement augmenté par rapport au premier test avec 50 utilisateurs. Ce qui est normal étant donné l'augmentation massive du nombre de connexion.

**Tableau 3 :** Récapitulatif du résultat du Load Test réalisé avec 1'000 utilisateurs

STATISTICS											
Requests ^	Executions				Response Time (ms)						
	Total	OK	KO	% KO	Min	Max	Mean	Std Dev	95th pct	99th pct	Req/s
Global Information	6000	5279	721	12 %	10	3290	653	685	2020	2880	92
GoToHomePage	1000	922	78	8 %	10	1600	329	513	1450	1500	15
ChangeLang	1000	946	54	5 %	10	1510	415	507	1440	1480	15
WatchMovies	1000	916	84	8 %	10	2580	539	619	1500	2440	15
ReturnToHomePage	1000	858	142	14 %	10	2850	821	693	2340	2610	15
ListenJukebox	1000	778	222	22 %	10	3290	1013	760	2670	3050	15
HomePageToFinish	1000	859	141	14 %	10	3290	800	710	2690	2970	15

#### 6.1.4. Conclusion

En l'état actuel Le Raspberry Pi arrive techniquement à gérer l'utilisation simultanée de la plateforme par 50 utilisateurs, représentant la capacité maximale d'un car, avec des bons temps de réponse. Cependant l'ajout de nouveaux services plus gourmands en termes de ressources pourraient modifier ce résultat à l'avenir.

## 6.2. TEST DES SERVICES

Le contenu des services est stocké dans le Raspberry Pi à l'intérieur du car. Une coupure de la connexion internet n'aura aucune conséquence pour un passager en cours d'utilisation d'un des services de la plateforme.

En revanche, en cas de coupure de l'alimentation, le Raspberry Pi s'éteint et donc la plateforme n'est plus accessible. Mais que se passe-t-il si un passager était en train de regarder une vidéo ou de consulter les événements ? Pour répondre à cette question chaque service a été testé afin d'observer son comportement dans un tel cas.

### 6.2.1. Jukebox et lecteur vidéos

Les musiques et les vidéos sont lues en streaming. Ce qui signifie que lorsque le passager lance une musique ou une vidéo celle-ci est téléchargée progressivement sur son appareil mobile. En cas de coupure du serveur la musique continuera à s'exécuter tant qu'elle a été téléchargée en mémoire. Si elle n'a pas complètement été téléchargée elle s'arrêtera avant la fin.

Le passager aura encore accès aux musiques et vidéos qu'il a déjà écoutées ou visionnées durant le trajet pour autant qu'il ne change pas de page, puisqu'elles ont déjà été téléchargées sur son appareil mobile.

Par ailleurs, lors du redémarrage du serveur la musique ou la vidéo en cours de lecture reprend à l'endroit où elle s'est arrêtée.

### 6.2.2. Agenda des manifestations

Lors de l'ouverture de la page contenant l'agenda des manifestations, ces dernières sont chargées entièrement. Une coupure de courant n'aura donc aucune influence. Le contenu sera toujours accessible par le passager ayant ouvert la page tant qu'il ne la quitte pas.

### 6.2.3. Formulaire de feedback et de contact

En cas de coupure du serveur pendant la saisie des données par le passager, celles-ci ne pourront pas être sauvegardées dans la base de données. Lorsque que l'utilisateur clique sur le bouton d'envoi, son navigateur internet va lui retourner une page d'erreur. Sur un appareil

*IOS*<sup>24</sup> et en utilisant *Safari*<sup>25</sup> comme navigateur, le passager a la possibilité de revenir sur la page du formulaire et les données saisies précédemment seront toujours là. Il aura simplement à cliquer sur le bouton d'envoi une fois le serveur redémarré. Cependant sur un appareil *Android*<sup>26</sup> utilisant *Chrome*<sup>27</sup> ou *Firefox*<sup>28</sup> comme navigateur, les données ne sont pas sauvegardées et le passager devra les saisir à nouveau.

### 6.3. TEST DU SYSTÈME DE MISE À JOUR

Il a fallu également analyser le comportement du système de mise à jour de la plateforme en cas de coupure du courant ou de la connexion internet en pleine mise à jour.

En cas de coupure de la connexion internet, *rsync* interrompt la synchronisation et les fichiers partiellement transférés sont stockés dans un dossier temporaire pour être utilisés lors de la prochaine synchronisation. Le code source de la plateforme est uniquement modifié lorsque la synchronisation est terminée. Par ailleurs, pour plus de sécurité la mise à jour de la base de données s'effectue seulement une fois que le transfert du nouveau contenu s'est correctement exécuté. Tant que la base de données n'a pas été mise à jour le nouveau contenu ne sera pas affiché sur la plateforme, et donc il n'y aura pas de problèmes d'affichage pour l'utilisateur.

Dans le cas d'une coupure de courant le même principe s'applique. Étant donné que *rsync* stocke les fichiers téléchargés dans un dossier temporaire avant de modifier le code source de la plateforme, le fonctionnement de celle-ci ne sera pas altéré.

<sup>24</sup> **IOS** : système d'exploitation pour les appareils mobiles d'Apple, <https://www.apple.com/chfr/ios/>

<sup>25</sup> **Safari** : navigateur internet développée par Apple, <https://www.apple.com/chfr/safari/>

<sup>26</sup> **Android** : système d'exploitation développé par Google, <http://www.android.com/>

<sup>27</sup> **Chrome** : navigateur internet développé par Google, [http://www.google.com/intl/fr\\_ch/chrome/browser/](http://www.google.com/intl/fr_ch/chrome/browser/)

<sup>28</sup> **Firefox** : navigateur internet libre développé par la fondation Mozilla, <https://www.mozilla.org/fr/>

#### 6.4. TEST SUR LA LIGNE SIERRE-ZINAL

La plateforme a été installée dans un car de la ligne Sierre – Zinal en vue d’analyser son comportement dans un environnement réel. La phase de test a débuté le lundi 28 juillet et a duré une quinzaine de jours.

Durant cette phase les passagers ont eu accès à la plateforme et ont pu transmettre leur feedback sur leur expérience. L’ajout de nouveaux contenus a également été testé. L’analyse des résultats de cette phase seront intégrés à ce présent rapport sous forme d’errata, étant donné que le test se termine après la date du rendu du rapport.

## 7. PROBLÈMES RENCONTRÉS

Ce chapitre a pour objectif d'expliquer certains problèmes apparus durant le développement de la plateforme.

### 7.1. ALIMENTATION DU RASPBERRY PI

Le premier problème à traiter se situait au niveau de l'alimentation du Raspberry Pi. Il a fallu déterminer comment l'alimenter dans un car. Une première solution était de le brancher sur une prise allume-cigare. Après un premier essai infructueux relatif à une prise-cigare défaillante, le deuxième fut concluant. La tension électrique mesurée à l'intérieur du Raspberry Pi est la même que s'il était branché sur une prise normale. Une autre solution consistait à le brancher directement au routeur présent dans le car via prise USB. Cette solution fonctionne également et présente l'avantage de faciliter l'accès internet en reliant le Raspberry Pi au routeur via un câble réseau. L'une ou l'autre de ces solutions peut être utilisée en fonction de la configuration du car.

De plus, lorsque le car s'arrête, l'alimentation est coupée nette et le Raspberry Pi s'éteint brusquement. Cet arrêt pose souci au niveau logiciel et peut corrompre la carte mémoire contenant tout le système d'exploitation et rendre la plateforme inutilisable. Ce cas de figure est arrivé pendant la phase de test après plusieurs retraits de l'alimentation. Afin de le résoudre, un UPS spécialement créé pour le Raspberry Pi a été installé. Cet appareil permet en cas de coupure de l'alimentation de continuer à alimenter le Raspberry Pi afin qu'il puisse s'éteindre correctement.

### 7.2. SYNCHRONISATION DE L'HEURE

Le Raspberry Pi ne possède pas d'horloge temps réel<sup>29</sup> (*Real Time Clock*), en cas d'arrêt l'horloge du système s'arrête également. Lors du redémarrage suivant l'horloge reprend à l'heure où elle s'est arrêtée. Plusieurs fonctionnalités de la plateforme utilise l'heure du système, il est donc important qu'elle soit correcte. Afin de corriger ce problème il existe un programme qui synchronise l'heure du Raspberry Pi avec le réseau NTP<sup>30</sup>. Cependant il

<sup>29</sup> **Horloge temps réel** : Real Time Clock (RTC), horloge de l'ordinateur gardant l'heure actuel

<sup>30</sup> **NTP** : Network Time Protocole, protocole permettant la synchronisation de l'heure de l'ordinateur avec une heure de référence, <http://www.ntp.org/>

nécessite une connexion internet pour fonctionner. A chaque démarrage du système et à chaque fois qu'il se connecte à internet l'heure est synchronisée. Par ailleurs, pour éviter que l'heure ne soit plus synchronisée pendant un certain temps il est possible d'installer un module RTC sur le Raspberry Pi.

### 7.3. ARCHITECTURE MVC

La mise en place de l'architecture MVC a apporté quelques désagréments. Dans un premier temps le développement de la plateforme n'a pas été réalisé dans un style MVC pur. Par la suite, avec le développement des différents services et l'ajout de nouveaux morceaux de codes, des problèmes de navigation et de gestion des langues sont survenus. Afin de corriger ces problèmes et de rendre le code plus lisible la structure des dossiers et des fichiers a été revue et un contrôleur central a été ajouté.

### 7.4. GESTION DES LANGUES

Le système qui permet de changer de langue fonctionne très bien sur les navigateurs internet mobiles testés, à savoir *Chrome*, *Firefox* et *Safari*. Il fonctionne également sur la version bureau de *Chrome* et sur *Internet Explorer*<sup>31</sup>. Par contre sur la version bureau de *Firefox* le système ne semble pas fonctionner, il est impossible de changer de langue. La solution à ce problème n'a pas été trouvée actuellement. Cependant étant donné que la plateforme est destinée en premier lieu aux appareils mobiles ce problème n'est pas critique au bon fonctionnement de celle-ci.

---

<sup>31</sup> **Internet Explorer** : navigateur internet développé par Microsoft, <http://windows.microsoft.com/fr-ch/internet-explorer/ie-9-worldwide-languages>

## 8. ÉTAT DES LIEUX

### 8.1. ÉTAT ACTUEL DU PROJET

Actuellement la plateforme est fonctionnelle et accueille quatre services proposant du contenu spécifique au Val d'Anniviers. Elle a été testée en environnement réel pendant un certain temps sur la ligne Sierre – Zinal afin de recueillir les impressions des passagers et d'analyser les éventuels problèmes.

### 8.2. AMÉLIORATIONS FUTURES

Ce chapitre énumère de possibles améliorations à développer. D'autres idées d'améliorations seront apportées par l'analyse des feedbacks des passagers ayant utilisés la plateforme.

#### 8.2.1. Améliorations des services existants

Des améliorations peuvent être apportées aux services existants pour les rendre plus attractifs.

Au niveau du jukebox et du lecteur de vidéos, un système de filtres pourrait être ajouté. Il permettrait à l'utilisateur de filtrer seulement un type de musique ou de trouver une musique spécifique. Un système de tri serait également utile, afin de permettre à l'utilisateur de trier selon la date d'ajout ou selon les auteurs par exemple. Quand beaucoup de contenu sera présent sur la plateforme, il sera nécessaire de créer un affichage par pages afin de limiter le nombre de musiques et de vidéos affichées sur une page.

Pour l'agenda des manifestations, il faudrait améliorer l'affichage pour rendre le contenu plus lisible en ajoutant une liste permettant de sélectionner le mois à afficher par exemple. Cela permettra de limiter le nombre d'événements affichés en même temps.

#### 8.2.2. Ajout de nouveaux services

L'ajout d'une carte du trajet interactive affichant la position actuelle du car, les arrêts et les différents points d'intérêts à proximité serait un service intéressant. Il donnerait une vraie valeur ajoutée à la plateforme. Une collaboration avec les offices du tourisme de la région serait nécessaire afin de déterminer les points d'intérêts.



Un autre service possible serait d'afficher les correspondances disponibles aux différents arrêts du trajet. Par exemple, pour la ligne Sierre-Zinal il serait intéressant d'avoir les correspondances lors de l'arrivée à la gare de Sierre et à la gare de Vissoie où plusieurs lignes de bus se croisent.

### 8.2.3. Intégration avec le système existant

Actuellement deux WI-FI sont disponibles à l'intérieur du car, celui du car et celui de la plateforme. Afin d'améliorer cette situation, il faudrait en supprimer un. La plateforme pourrait remplacer la page d'accueil actuel du WI-FI du car et un nouveau service permettrait aux passagers de s'enregistrer directement depuis la plateforme afin de pouvoir utiliser internet. Le WI-FI du car ne serait plus nécessaire et les passagers utiliseraient le WI-FI de la plateforme pour se connecter.

### 8.2.4. Amélioration de l'UPS

L'UPS utilisé fonctionne grâce à six piles. Cependant ces piles ne sont pas rechargées et devront être remplacées d'ici quelque temps. Il faudrait utiliser un UPS qui se recharge lorsque l'alimentation est branchée afin d'améliorer le système et de le rendre autonome. Un tel UPS existe sur le marché et devra être testé afin de l'installer sur la plateforme.

### 8.2.5. Création d'un écran de contrôle

Un écran de contrôle pour le chauffeur du car pourrait être installé. Cet écran lui permettrait de transmettre des alertes ou des annonces de retard à la centrale. Un tel système nécessiterait une connexion permanente à internet. L'écran pourrait aussi afficher l'état de la plateforme en temps réel.

### 8.2.6. Création d'une page d'administration

Une page pourrait être créée afin d'administrer la plateforme à distance. Cette dernière permettrait de monitorer son comportement et d'y effectuer des maintenances. Elle permettrait également la gestion du contenu, avec la possibilité d'en ajouter du nouveau, et de modifier ou supprimer le contenu existant.

## CONCLUSION

Une plateforme fonctionnelle pouvant accueillir différents services a été implémentée. La plateforme intègre le multilinguisme et son design respecte la notion d'expérience utilisateur, tel que demandé dans la description du travail de bachelor. Cependant, l'accès à internet également mentionné dans cette description ne peut être garanti, étant donné qu'il dépend de la connexion déjà présente dans les cars et non de la plateforme.

Comme mentionné dans le chapitre précédent, la plateforme nécessite des améliorations afin de répondre complètement aux attentes des clients. En ce sens l'analyse des feedbacks reçus durant la phase de test sera importante. Des adaptations seront également à envisager en cas d'utilisation sur plusieurs lignes, notamment au niveau du contenu des services qui est actuellement adapté seulement au Val d'Anniviers.

Une version de démonstration de la plateforme est disponible en ligne. Les informations pour y accéder sont disponibles en annexe (cf. **Annexe VIII**).

## RÉFÉRENCES

### ÉTAT DE L'ART

Lufthansa Systems. (08.07.2013). *Greyhound launches "BLUE™", an exclusive WI-FI enabled onboard entertainment system*. Récupéré sur

<https://www.lhsystems.com/media/news/latest-news/press-release/article/greyhound-launches-blueTM-an-exclusive-wi-fi-enabled-onboard-entertainment-system.html>

Lufthansa Systems. (24.02.2014). *ADAC/Postbus uses Lufthansa Systems Velimo* [Vidéo].

Récupéré sur <https://www.youtube.com/watch?v=VRXPWDu43xk>

Lufthansa Systems. (30.10.2013). *Velimo from Lufthansa Systems provides entertainment and information for ADAC Postbus passengers*. Récupéré sur

<https://www.lhsystems.com/media/news/latest-news/press-release/article/velimo-from-lufthansa-systems-provides-entertainment-and-information-for-adac-postbus-passengers.html>

Lufthansa Systems. *Velimo – Premier Infotainment for Road 'n' Rail* [PDF]. Récupéré sur

[https://www.lhsystems.com/fileadmin/user\\_upload/files/en/information/br-velimo-24022014.pdf](https://www.lhsystems.com/fileadmin/user_upload/files/en/information/br-velimo-24022014.pdf)

Mobile Onboard Limited. *Beam Wi-Fi. Innovative, flexible on-bus Wi-Fi made easy*. Récupéré

sur <http://mobileonboard.com/beam-bus-wi-fi/>

Mobile Onboard Limited. (08.01.2014). *Industry guide passenger comfort & entertainment*

[PDF]. Récupéré sur [http://mobileonboard.com/wp-content/uploads/042\\_CBW\\_1119\\_Industry-guide.pdf](http://mobileonboard.com/wp-content/uploads/042_CBW_1119_Industry-guide.pdf)

Phil Tottman. (10.07.2013). *Lufthansa develops mobile entertainment system for Greyhound buses*. Récupéré sur

<http://www.mobile-ent.biz/brands/entertainment/lufthansa-develops-mobile-entertainment-system-for-greyhound-buses/041360>

### DÉVELOPPEMENT DU SYSTÈME

Andrew Tridgell, Paul Mackerras & Wayne Davison. (22.06.2014). *rsync*. Récupéré sur

<http://rsync.samba.org/ftp/rsync/rsync.html>

Car Postal Suisse SA. *Car Postal Suisse SA*. Récupéré sur <http://www.carpostal.ch/pag-startseite/pag-ueberuns.htm>

Chris Clay. (16.01.2014). *Raspberry Pi: 11 reasons why it's the perfect small server*. Récupéré

sur <http://www.zdnet.com/raspberry-pi-11-reasons-why-its-the-perfect-small-server-7000025206/>

Dan Mundy. (04.11.2013). *How to connect your Raspberry Pi to two Wi-Fi networks at the same time* [Publication sur blog]. Récupéré sur

<http://www.processthings.com/post/66023171876/how-to-connect-your-raspberry-pi-to-two-wi-fi-networks>

Dave Conroy. (10.07.2013). *Turn Your Raspberry Pi into a WiFi Hotspot with Edimax Nano USB EW-7811Un (RTL8188CUS chipset)* [Publication de blog]. Récupéré sur <http://www.daveconroy.com/turn-your-raspberry-pi-into-a-wifi-hotspot-with-edimax-nano-usb-ew-7811un-rtl8188cus-chipset/>

Debian Addict. (13.07.2005). *Installer un serveur FTP : vsftpd*. Récupéré sur <http://www.debianaddict.org/article47.html>

Ducky Pond. (05.09.2013). *How to setup a web server with Nginx/PHP on Raspberry Pi* [Publication de blog]. Récupéré sur <http://www.ducky-pond.com/posts/2013/Sep/setup-a-web-server-on-rpi/>

Jeremy Morgan. (08.12.2012). *The Raspberry Pi Web Server Speed Test* [Publication sur blog]. Récupéré sur <http://www.jeremymorgan.com/blog/programming/raspberry-pi-web-server-comparison/>

Jeremy Osborn. *Responsive Web Design, Promote Yourself Responsively: Build a Portfolio for Every Device* [Cours en ligne / Vidéos]. Récupéré sur <http://gymnasium.aquent.com/catalog/course/101>

Lady Ada. (17.02.2014). *Setting up a Raspberry Pi as a WiFi access point*. Récupéré sur <https://learn.adafruit.com/setting-up-a-raspberry-pi-as-a-wifi-access-point/overview>

Laurent Hinoul. (18.02.2014). *How to install TP-Link TL-WN725N on Raspberry Pi* [Publication sur blog]. Récupéré sur <http://laurenthinoul.com/how-to-install-tp-link-tl-wn725n-on-raspberry-pi/>

Ludovic Lemaire. (03.01.2014). *Le Raspberry Pi et sa consommation électrique*. Récupéré sur <http://www.jeremymorgan.com/blog/programming/raspberry-pi-web-server-comparison/>

Mathieu Nebra. (04.01.2013). *Reprenez le contrôle à l'aide de Linux : Exécuter un programme à une heure différée*. Récupéré sur <http://fr.openclassrooms.com/informatique/cours/reprenez-le-controle-a-l-aide-de-linux/executer-un-programme-a-une-heure-differee>

Mathieu Nebra. (26.07.2013). *Concevez votre site web avec PHP et MySQL : Organiser son code selon l'architecture MVC*. Récupéré sur <http://fr.openclassrooms.com/informatique/cours/concevez-votre-site-web-avec-php-et-mysql/organiser-son-code-selon-l-architecture-mvc>

MySQL. *SQL Statement Syntax*. Récupéré sur <http://dev.mysql.com/doc/refman/5.7/en/sql-syntax.html>

Stanislas Quastana. (21.06.2011). *HTML5 – ce qu'il faut savoir sur la balise VIDEO* [Publication sur blog]. Récupéré sur <http://blogs.technet.com/b/stanislas/archive/2011/06/21/html5-ce-qu-il-faut-savoir-sur-la-balise-video.aspx>

Stéphane Landelle. (16.09.2013). *First Steps with Gatling*. Récupéré sur <https://github.com/excilys/gatling/wiki/First-Steps-with-Gatling>

Victor. (20.02.2014). *Les alternatives au Raspberry Pi* [Publication sur blog]. Récupéré sur <http://the-raspberry.com/alternatives>

W3schools. *HTML5 Video*. Récupéré sur [http://www.w3schools.com/html/html5\\_video.asp](http://www.w3schools.com/html/html5_video.asp)


W3schools. *HTML5 Audio*. Récupéré sur [http://www.w3schools.com/html/html5\\_audio.asp](http://www.w3schools.com/html/html5_audio.asp)

## DÉCLARATION DE L'AUTEUR

Je déclare, par ce document, que j'ai effectué le travail de bachelor ci-annexé seul, sans autre aide que celles dûment signalées dans les références, et que je n'ai utilisé que les sources expressément mentionnées. Je ne donnerai aucune copie de ce rapport à un tiers sans l'autorisation conjointe du RF et du professeur chargé du suivi du travail de bachelor, y compris au partenaire de recherche appliquée avec lequel j'ai collaboré, à l'exception des personnes qui m'ont fourni les principales informations nécessaires à la rédaction de ce travail et que je cite ci-après :

- M. Pierre-Yves Gux
- M. Alexandre Cotting

Célien Devanthéry

A handwritten signature in black ink, appearing to read 'Célien', with a long horizontal stroke extending to the right.

## ANNEXE I : PLANIFICATION DU PROJET

	①	Nom	Durée	% de complét	Début	Fin	Prédécesseurs	Ressources
1		☐Analyse comparative de solutions existantes	18j	86%	28/04/2014	21/05/2014		
2	👤👤👤	Effectuer les recherches de solutions existantes	13j	100%	28/04/2014	14/05/2014		Célien
3	👤👤👤	Définir les critères de comparaison	1j	100%	15/05/2014	15/05/2014	2	Célien
4	👤👤	Analyser / tester les solutions trouvées en fonction des critères	1j	100%	16/05/2014	16/05/2014	3	Célien
5	👤	Contact solutions trouvées pour questions	0.5j	100%	19/05/2014	19/05/2014	4	
6	👤👤	Classer les solutions trouvées	0.5j	100%	19/05/2014	19/05/2014	5	Célien
7	👤	Sélectionner la meilleure solution, définir comment l'installer,	0.5j	0%	20/05/2014	20/05/2014	6	Célien
8	👤	Création d'un nouveau système si aucune solution existe	2j	0%	20/05/2014	21/05/2014	6	Célien
9		☐Implémentation du système	22.5j	0%	22/05/2014	23/06/2014	1,7,8	
10	👤	Définir les besoins matériels nécessaires (alimentation...) pour	0.5j	0%	22/05/2014	22/05/2014	7,8	Célien
11		☐ Si solution existante	13j	0%	22/05/2014	10/06/2014		
12	👤	Installation	2j	0%	22/05/2014	26/05/2014	10	Célien
13	👤	Customizing	5j	0%	26/05/2014	02/06/2014	12	Célien
14	👤	Test du système dans un car	1j	0%	02/06/2014	03/06/2014	13	Célien
15	👤	Analyse du test et corrections des problèmes / Amélioration	5j	0%	03/06/2014	10/06/2014	14	Célien
16		☐ Si nouveau système	22j	0%	22/05/2014	23/06/2014		
17	👤	Création de l'architecture MVC client-server	1j	0%	22/05/2014	23/05/2014	10	Célien
18	👤	Configuration du hardware	3j	0%	23/05/2014	28/05/2014	17	Célien
19	👤	Tests de performance	2j	0%	28/05/2014	30/05/2014	18	Célien
20	👤	Implémentation du site qui va accueillir les services	5j	0%	30/05/2014	06/06/2014	19	Célien
21		Traduction (FR - DE - EN)	2j	0%	06/06/2014	10/06/2014	20	Traducteur
22	👤	Implémentation du système de mises à jour (via les wifi de	3j	0%	10/06/2014	13/06/2014	21	Célien
23	👤	Test du système dans un car	1j	0%	13/06/2014	16/06/2014	22	Célien
24	👤	Analyse du test et corrections des problèmes / amélioration	5j	0%	16/06/2014	23/06/2014	23	Célien
25		☐Implémentation de services	19j	0%	23/06/2014	18/07/2014	9	
26	👤	Définition des services à implémenter avec CarPostal	1j	0%	23/06/2014	24/06/2014		Célien
27	👤	Implémentation des services	10j	0%	24/06/2014	08/07/2014	26	Célien
28		Traduction (FR - DE - EN)	1j	0%	07/07/2014	08/07/2014	27FD-1j	Traducteur
29	👤	Mise en place dans car pour tests feedback des utilisateurs	5j	0%	08/07/2014	15/07/2014	27,28	Célien
30	👤	Analyse des feedback et corrections des problèmes / amélioration	3j	0%	15/07/2014	18/07/2014	29	Célien
31		☐Finalisation du projet	11.5j	0%	10/07/2014	25/07/2014		
32	👤👤	Finalisation du rapport	9j	0%	10/07/2014	22/07/2014	29DD 1j	Célien
33	👤👤	Relecture et corrections	2j	0%	23/07/2014	24/07/2014	32	Correctrice
34	👤👤	Impression reliure	0.5j	0%	25/07/2014	25/07/2014	33	Célien

Planification réalisée avec Gantter, <http://www.gantter.com/>

## ANNEXE II : CAHIER DES CHARGES

### Cahier des charges - CarPostal Entertainment

#### Objectif du travail

CarPostal aimerait développer des services à valeur ajoutée à proposer aux passagers de leurs bus afin de favoriser une croissance du nombre de passagers. Pour ce faire, ce travail a pour but dans un premier temps d'implémenter un système qui va accueillir les différents services et, dans un deuxième temps, de développer ces services.

#### Etape 1 : Analyse de solutions existantes (échéance = 21.05.2014 / durée = 18 jours)

**Etat de l'art** – Pour commencer des recherches de solutions existantes tant propriétaires qu'open-source doivent être réalisées.

**Comparaison** – Une fois ces recherches menées, les différentes solutions trouvées doivent être analysées et comparées entre elles à l'aide de différents critères à définir.

**Choix** – Suite à l'analyse, la solution remplissant tous les critères nécessaires sera retenue. Si une telle solution n'existe pas, un nouveau système sera créé.

#### Délivrables

Document contenant une analyse comparative des différentes solutions et le choix final de la solution (solution existante ou nouveau système)

#### Etape 2 : Implémentation du système (échéance = 20.06.2014 / durée = 22 jours)

Un Product Backlog sera créé en fonction du système choisi.

#### Solution existante (durée = 13 jours)

**Besoins** – En cas de solution existante il faudra définir les besoins que nécessite l'installation de cette solution.

**Customizing** – Le système devra être adapté/modifié en fonction des besoins spécifiques de CarPostal.

**Installation et tests** – Une fois le système implémenté il sera installé dans un car pour une période de tests.

**Analyse des résultats** – Une analyse des résultats du test et des feedbacks des clients sera réalisée afin d'améliorer le système.





### Nouveau système (durée = 22 jours)

**Architecture** – En cas de création d'un nouveau système, il faudra commencer par créer une architecture client-serveur MVC.

**Composants** – Un choix des composants hardware et software adaptés aux besoins sera effectué.

**Implémentation** – Une fois les composants choisis, les parties hardware et software seront développées en fonction de l'architecture définie et des différentes spécifications que doit posséder le système (multilinguisme FR – DE – EN, UX design, système de mise à jour du contenu).

**Installation et tests** – Une fois le système implémenté il sera installé dans un car pour une période de tests.

**Analyse des résultats** – Une analyse des résultats du test et des feedbacks des clients sera réalisée afin d'améliorer le système.

### Délivrables

Product Backlog

Système (hardware et software) fonctionnel, testé et prêt à recevoir les services

### Etape 3 : Implémentation des services (échéance = 17.07.2014 / durée = 19 jours)

**Choix des services** – Une discussion avec le responsable de CarPostal aura lieu afin de définir quels services devront être implémentés dans le système. Le choix des services sera également défini par le temps à disposition et leur complexité.

Création du Product Backlog en fonction des services choisis.

**Implémentation** – Les services choisis seront implémentés.

**Tests** – Une fois les services développés et installés dans le système, une phase de test aura lieu.

**Analyse des résultats** – Une analyse des résultats du test, ainsi que les feedbacks des clients sera réalisée afin d'améliorer le système.

### Délivrables

Product Backlog

Système complet, testé et fonctionnel



#### Etape 4 : Finalisation du rapport (échéance = 25.07.2014 / durée = 12 jours)

Le rapport sera rédigé, mis en page et corrigé.

Le code source du système sera également nettoyé afin de le rendre réutilisable pour une éventuelle suite du projet.

#### Délivrables

Deux dossiers reliés + deux CDs (avec code source) + version PDF

Résumé de deux pages du TB + version PDF

Décompte des heures effectuées

**Réalisé le 15 mai 2014**

Par Célien Devanthéry

**Approuvé le 17 mai 2014**

Par Pierre-Yvex Guex



## ANNEXE III : JOURNAL DE BORD

### SEMAINE 1 (28.04.2014 AU 04.05.2014)

Séance d'introduction avec M. Pierre-Yves Guex	
Entretien avec M. Alexandre Cotting	
Recherches de solutions existantes	3h30

### SEMAINE 2 (05.05.2014 AU 11.05.2014)

Préparation de la mise en page du rapport	
Lecture des règlements	
Préparation du cahier des charges	
Création d'un diagramme de Gantt avec la liste des tâches et leur durée	
Recherches de solutions existantes	
Définition des critères de comparaison des solutions trouvées	15h30

### SEMAINE 3 (12.05.2014 AU 18.05.2014)

Préparation du cahier des charges	
Analyse comparative des solutions trouvées	
Rédaction du rapport	
Séance avec M. Pierre-Yves Guex	
Préparation des mails de demande de complément d'informations	29h30

### SEMAINE 4 (19.05.2014 AU 25.05.2014)

Analyse comparative des solutions trouvées	
Choix des technologies à utiliser	12h00

### SEMAINE 5 (26.05.2014 AU 01.06.2014)

Choix des technologies à utiliser	
Création de l'architecture du système	17h00

### SEMAINE 7 (09.06.2014 AU 15.06.2014)

Entretien avec M. Alexandre Cotting	
Configuration du Raspberry Pi	
Création de la plateforme	22h00

**SEMAINE 8 (16.06.2014 AU 22.06.2014)**

Mise en place du Responsive Web Design	10h00
--	-------

**SEMAINE 9 (23.06.2014 AU 29.06.2014)**

Mise en place du Responsive Web Design	
Formulaire de contact	
Implémentation du multilinguisme	
Création du serveur central	
Création des scripts pour le système de mise à jour	
Traduction de la plateforme	
Rédaction du rapport	45h00

**SEMAINE 10 (30.06.2014 AU 06.07.2014)**

Amélioration de la configuration du Raspberry Pi	
Amélioration des scripts	
Traduction	
Tests du Raspberry Pi dans un car	
Entretien téléphonique avec M. Laurent Flück de CarPostal	
Rédaction du rapport	42h30

**SEMAINE 11 (07.07.2014 AU 13.07.2014)**

Création du système de mise à jour par WI-FI	
Tests de performance	
Modification de l'architecture MVC	
Création de la base de données	
Création du script pour la mise à jour de la base de données	
Gestion du lancement des scripts avec cron	
Création des services	59h30

## SEMAINE 12 (14.07.2014 AU 20.07.2014)

Création des services

Rédaction du rapport

Création du formulaire de feedback

Séance avec M. Pierre-Yves Guex

Rédaction du tutoriel

54h00

## SEMAINE 13 (21.07.2014 AU 27.07.2014)

Installation de l'UPS

Création des affiches pour le car

Rédaction du rapport

Finalisation et impression du rapport

56h30

DURÉE TOTAL DU PROJET

367 HEURES

## ANNEXE IV : RÉPONSE DE LUFTHANSA SYSTEMS

dirk.zorembsky@lhsystems.com

À celien.devanthery@students.hevs.ch

mer. 18 juin 17:15

### AW: Request from Célien Devanthery

Dear Mrs. Devanthery,

thank you for your interest in Velimo, our innovative platform which allows passengers to access the Internet, check emails, go online shopping, watch their favorite movies or television shows, read newspapers and magazines, play online games while they are on a bus or train ride. Please find the answers to your questions below:

- Is your platform working in Switzerland ? Are you selling it in Switzerland ?

LH Systems: The Velimo platform has no territorial restrictions and will be sold in Switzerland

- What would be the price for using this platform in a bus per year ?

LH Systems: The pricing depends on certain factors like the selected functionality, selected content or the size of the fleet. Lufthansa Systems would like to discuss the pricing after a detailed requirement and scope discussion with a potential customer.

- Is it possible to customize the system by adding new specific services (for example a quiz) ? What would be the estimated price for this development ?

LH Systems: Velimo is designed in a very modular way which allows customer specific enhancements. A pricing for any enhancement requires a detailed requirement analysis together with the customer.

- As the buses don't have a permanent connection to the Internet, which services can be used offline ?

LH Systems: Due to the fact that all content is locally on the bus server the main services like watching movies, listen music or read newspapers can be used without a permanent internet connection.

- Is your platform multilingual (French - German - English) ?

LH Systems: Currently we are developing the multilingual feature which will be available in Q3 2014. The languages will be predefined by the customer.

I hope those information's are helpful for you and I would be very delighted if we could exchange more information may be together with the bus company you represent.

Best Regards

Dirk Zorembsky  
Hospitality Solutions  
Lufthansa Systems AS GmbH  
HAM AB/E  
Am Prime Parc 1  
65479 Raunheim

Germany  
Phone: +49 (0)69-696-26915  
Mobile: +49 (0)175-294-9846  
E-Mail: dirk.zorembsky@LHsystems.com  
www.LHsystems.com

Sitz der Gesellschaft / Corporate Headquarters: Lufthansa Systems AS GmbH, Norderstedt,  
Registereintragung / Registration: Amtsgericht Norderstedt 3688NO  
Geschaeftsfuehrung / Management Board: Bernd Appel

-----Ursprüngliche Nachricht-----

Von: Lufthansa Systems [mailto:fraacmweb@lhsystems.com]  
Gesendet: Freitag, 16. Mai 2014 09:51  
An: LSY, HAM SALES EMEA INDUSTRIES  
Betreff: Request from Celien Devanthery

This is a request via www.LHsystems.com.

Name: Celien Devanthery  
Company:  
E-mail: celien.devanthery@students.hevs.ch  
Phone:  
Region: EMEA

Message: Dear Sir or Madam,

I'm a Swiss IT student from the HES-SO Valais. For my Bachelor's thesis I must implement an entertainment/infotainment system for a bus company. After some researches I've found your solution Velimo Road 'n' Rail and I have a few questions about it:

- Is your platform working in Switzerland ? Are you selling it in Switzerland ?
- What would be the price for using this platform in a bus per year ?
- Is it possible to customize the system by adding new specific services (for example a quiz) ? What would be the estimated price for this development ?

- As the buses don't have a permanent connection to the Internet, which services can be used offline ?
- Is your platform multilingual (French - German - English) ?

Your answers will help me to define the platform that will fit the most the needs of the company that I represent.

Yours faithfully,

Célien Devanthery



## ANNEXE V : CAPTURES D'ÉCRAN DES SERVICES

### CarPostal Entertainment

#### JUKEBOX

Une sélection de musiques pour vous divertir durant le trajet.

WHO CARES - SAM BROWN  
10-06-2014

3:08

Pop / Rock

BREEZE - GYAKO  
31-05-2014

3:17

Electro

SONRIE - GUSTAVO CROCHENCI  
18-05-2014

3:13

jazz

### CarPostal Entertainment

#### WEB

Une connexion internet est disponible !

#### LIENS

- CarPostal
- Val d'Anniviers
- Valais
- CFF

### CarPostal Entertainment

#### LES VIDÉOS

Profitez de votre trajet avec cette sélection de vidéos de la région.

TWISTER - VALS D'ANNIVIERS ET D'HÉRENS [FR]  
03-07-2014

Anniviers et Hérens  
Les vals de la Vercofly  
3/7/2014

3:01

Source: EcoleTwistair

VALAIS. GRAVÉ DANS MON COEUR [FR]  
23-05-2014

0:30

Source: Valais Wallis

## CarPostal Entertainment

### AGENDA DES MANIFESTATIONS 2014

Cet agenda contient les manifestations à venir de la région.

#### JUILLET

26 - Inauguration des Parours historiques d'Anniviers - **Vissoie**

26 - Fête Patronale de Fang(Saint-Germain d'Aux.) - **Fang**

26 - FangEpicuria - l'abricot dans tous ses états - **Fang**

26 - Vercojazz - **Vercorin**

26 - Vercojazz festival off - **Briey**

26 - Vercojazz festival off - **Vercorin**

26 - Randonnée en montagne avec découverte des plantes et Qi Gong - **Vercorin**

27 - Fête des prémices à l'alpage de Rouaz - **Saint-Luc**

27 - Fête Patronale d'Ayer(Sainte-Anne) - **Ayer**

27 - Fête à l'alpage de Moiry - orchestres - cantines - messe - **Grimenz**

27 - Concert de musique de Chambre - violons, violoncelle, alto et clarinette - **Saint-Jean**

27 - Vercojazz - **Vercorin**

28 - Gala des Vins : dégustation-vente de plus de 40 vins de la région - **Chandolin**

## CarPostal Entertainment

### FEEDBACK

QUELLE EST L'UTILITÉ D'UNE TELLE PLATEFORME SELON VOUS?

(AUCUNE UTILITÉ ---> TRÈS UTILE)\*

QUELS SONT LES ÉLÉMENTS À AMÉLIORER?

QUELS SERVICES AIMERIEZ-VOUS AVOIR EN PLUS?

EST-CE QUE VOUS SERIEZ INTÉRESSÉ PAR UNE CARTE DU TRAJET INTERACTIVE INDIQUANT LES DIFFÉRENTS ARRÊTS ET LES POINTS D'INTÉRÊTS À PROXIMITÉ?\*

OUI

☐

NON

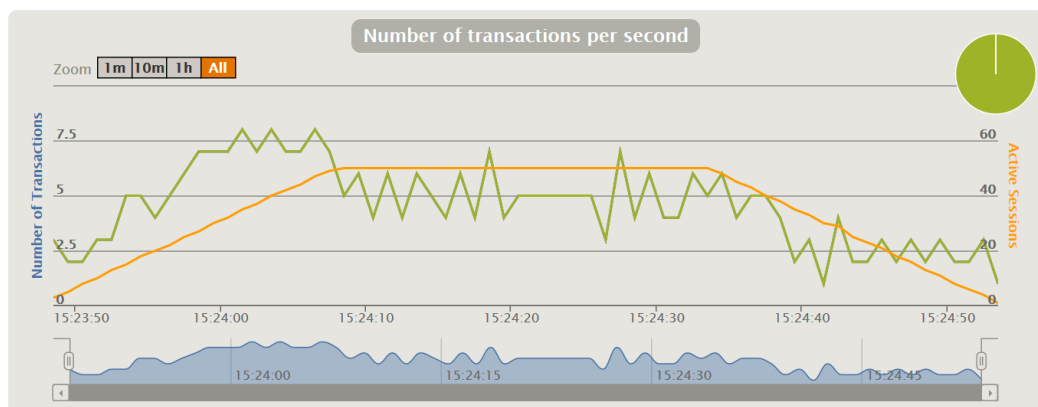
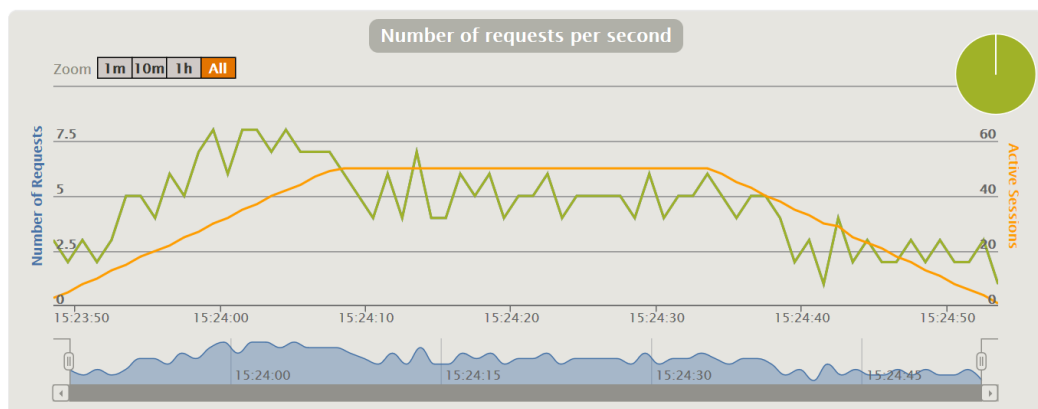
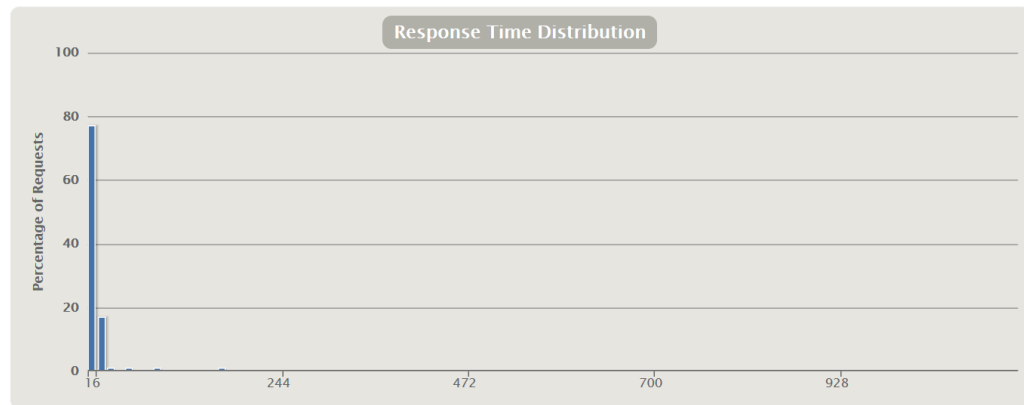
☐

AUTRES COMMENTAIRES

Envoyer

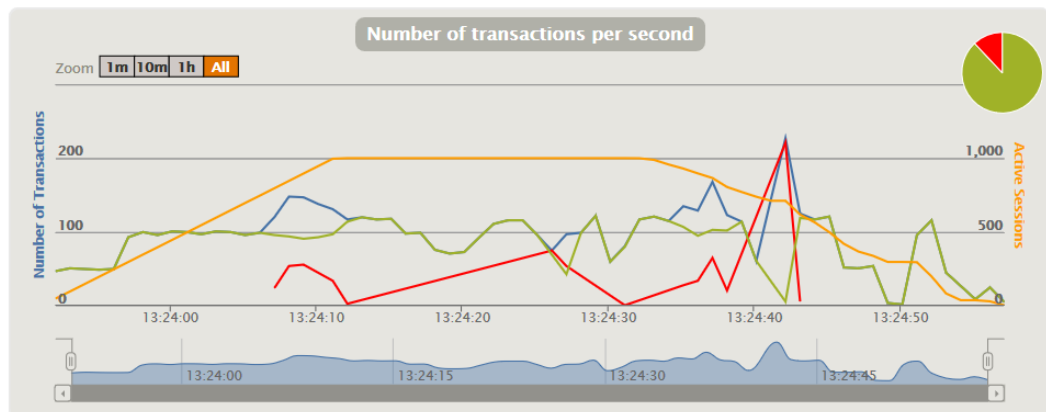
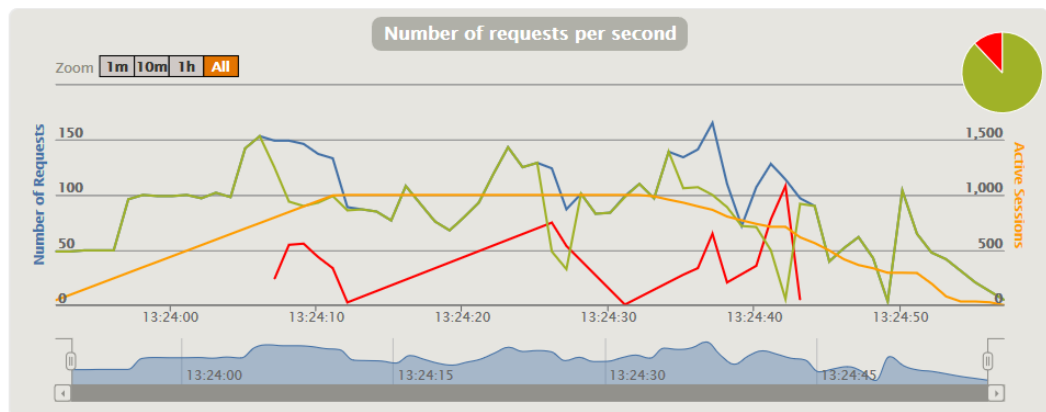
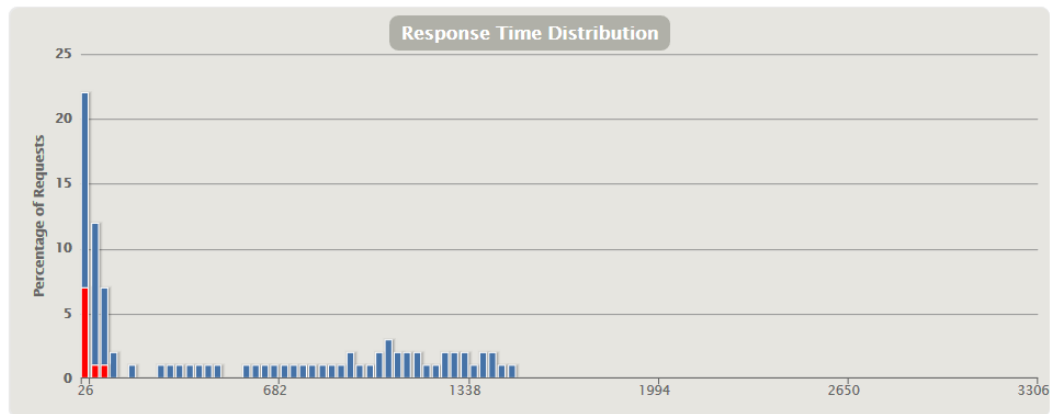
## ANNEXE VI : RÉSULTAT DU LOAD TEST AVEC 50 UTILISATEURS





## ANNEXE VII : RÉSULTAT DU LOAD TEST AVEC 1'000 UTILISATEURS





## ANNEXE VIII : INFORMATIONS POUR ACCÉDER AUX RESSOURCES

### VERSION DE DÉMONSTRATION

Adresse : <http://ec2-54-191-4-233.us-west-2.compute.amazonaws.com/demo/>

Nom d'utilisateur : admin

Mot de passe : d45gfThy654

### RASPBERRY PI

#### INFORMATIONS DE BASE

Adresse IP : 192.168.1.1

Nom d'utilisateur : pi

Mot de passe : carpostal14

#### MYSQL

Nom d'utilisateur : root

Mot de passe : D4Ts24Ds5

### SERVEUR CENTRAL

#### INFORMATIONS DE BASE

Adresse IP : 54.191.4.233

Clé SSH : disponible sur le CD-ROM dans le dossier *SSH*

#### MYSQL

Nom d'utilisateur : root

Mot de passe : D4Ts24Ds5

#### VSFTPD

Nom d'utilisateur : pi

Mot de passe : 47Gb662f3CnL

## ANNEXE IX : TUTORIEL D'INSTALLATION DE LA PLATEFORME



# TUTORIEL D'INSTALLATION DE LA PLATEFORME DE SERVICES

TRAVAIL DE BACHELOR CARPOSTAL ENTERTAINMENT

CÉLIEN DEVANTHÉRY

25.07.2014

## CONTEXT

Ce tutoriel explique les différentes étapes nécessaires à la configuration de la plateforme de services dans le cadre du Travail de Bachelor « CarPostal Entertainment ». Ce travail a pour objectif la création d'une plateforme web contenant des services accessibles via WI-FI par les passagers des cars postaux.

Pour réaliser cette plateforme un Raspberry Pi est utilisé. Il doit fournir un point d'accès WI-FI à l'intérieur du car pour que les passagers puissent s'y connecter et accéder aux services.

Un accès à internet via WI-FI est déjà disponible à l'intérieur des cars postaux. Cependant la qualité de la connexion est très variable et parfois nulle. C'est pourquoi un système de mise à jour du contenu a été créé. Ce système permet au Raspberry Pi de se connecter à internet soit en utilisant la connexion du car si elle est existante, soit en utilisant les WI-FI disponibles aux alentours des bâtiments postaux le long du parcours du car. Une fois connecté à internet il peut accéder au serveur central afin de synchroniser son contenu.

Le Raspberry Pi est alimenté directement par le car. En cas d'arrêt du car l'alimentation est coupée et le système s'éteint immédiatement ce qui peut poser problème au niveau logiciel. Afin de résoudre ce problème un UPS<sup>1</sup> a été ajouté au Raspberry Pi lui permettant de s'éteindre de manière sûre après l'arrêt du car.

Ce tutoriel nécessite des connaissances préalables dans la configuration d'un système linux en ligne de commande.

---

<sup>1</sup> UPS : dispositif qui fournit à un système électrique ou électronique une alimentation constante, même en cas de coupure de l'alimentation

# TABLE DES MATIÈRES

<b>1. MATÉRIEL ET OUTILS NÉCESSAIRES .....</b>	<b>2</b>
1.1. HARDWARE .....	2
1.2. SOFTWARE.....	2
<b>2. CONFIGURATION DE SERVEUR WEB LOCAL .....</b>	<b>3</b>
2.1. INSTALLATION DU SYSTÈME D'EXPLOITATION .....	3
2.1.1. Téléchargement de l'image .....	3
2.1.2. Écriture de l'image sur la carte SD.....	3
2.1.3. Modification du fichier « config.txt » .....	4
2.2. BRANCHEMENT DES DIFFÉRENTS COMPOSANTS .....	4
2.3. CONFIGURATION DE BASE .....	5
2.3.1. Raspi-config .....	5
2.3.2. Suppression de wolfram-engine .....	6
2.3.3. Mise à jour du système .....	6
2.4. INSTALLATION DES CLÉS WI-FI .....	7
2.4.1. Branchement des clés WI-FI .....	7
2.4.2. Configuration des interfaces réseau.....	8
2.5. CONFIGURATION DE WLAN1 .....	9
2.6. CONFIGURATION DE WLAN0 (POINT D'ACCES WI-FI) .....	11
2.6.1. Installation du logiciel.....	11
2.6.2. Configuration du serveur DHCP .....	12
2.6.3. Configurer le réseau WI-FI.....	13
2.6.4. Configurer le NAT.....	13
2.6.5. Test du point d'accès.....	15
2.7. ACCÈS AU SYSTÈME PAR SSH .....	15
2.8. CONFIGURATION DU SERVEUR WEB .....	17
2.8.1. Installation de Nginx et PHP .....	17
2.8.2. Création d'un nouveau site.....	17
2.9. CONFIGURATION DE LA BASE DE DONNÉES .....	18
2.9.1. Installation de MySQL.....	18
2.9.2. Création d'une nouvelle base de données .....	18
2.10. CRÉATION DES SCRIPTS.....	19
2.10.1. Mise à jour du contenu (udpate.sh).....	19
2.10.2. Mise à jour de la base de données (updateDB.sh).....	20
2.10.3. Envoi des commentaires et des feedbacks (sendComments.sh).....	20
2.10.4. Envoi des fichiers log (sendLogs.sh).....	21

2.10.5.	<i>Script central (main.sh)</i> .....	21
2.10.6.	<i>Lancement automatique du script</i> .....	24
2.11.	INSTALLATION DE L'UPS.....	24
<b>3.</b>	<b>CONFIGURATION DU SERVEUR CENTRAL</b> .....	<b>26</b>
3.1.	SSH .....	26
3.2.	ADRESSE IP FIXE .....	26
3.3.	CONFIGURATION DU SERVEUR FTP .....	26
3.3.1.	<i>Installation du serveur</i> .....	26
3.3.2.	<i>Création d'un nouvel utilisateur</i> .....	27
3.4.	CONFIGURATION DU SERVEUR WEB [FACULTATIF] .....	27
<b>RÉFÉRENCES</b>	.....	<b>29</b>

# 1. MATÉRIEL ET OUTILS NÉCESSAIRES

Les différents composants et outils utilisés pour la réalisation de la plateforme sont décrits dans ce chapitre.

## 1.1. HARDWARE

- 1x **serveur FTP**
- 1x **Raspberry Pi Type B**
- 1x **adaptateur** avec **câble micro-USB** pour l'alimentation, ou seulement un câble micro-USB (possibilité de le connecter à l'ordinateur)
- 1x **SD Card** 8GB minimum, Type 10 recommandé
- 2x clés **USB WI-FI** (Edimax EW-7811Un recommandé)
- 1x **CW2 PI UPS**<sup>2</sup>
- 1x **clavier** connectable par USB
- 1x **lecteur de carte SD**
- 1x **écran avec prise HDMI** ou 1x **écran avec prise VGA** avec un **adaptateur HDMI vers VGA**<sup>3</sup>
- 1x **câble réseau** avec **connexion à internet**

## 1.2. SOFTWARE

- **Win32 Disk Imager**<sup>4</sup>, pour écrire l'image du système d'exploitation sur la carte SD
- **PuTTY**<sup>5</sup>, pour se connecter au Raspberry Pi via SSH

---

<sup>2</sup> <http://www.piups.net/index.php#home>

<sup>3</sup> <https://www.modmypi.com/pi-view-raspberry-pi-hdmi-to-vga-convert>

<sup>4</sup> <http://sourceforge.net/projects/win32diskimager/>

<sup>5</sup> <http://www.putty.org/>

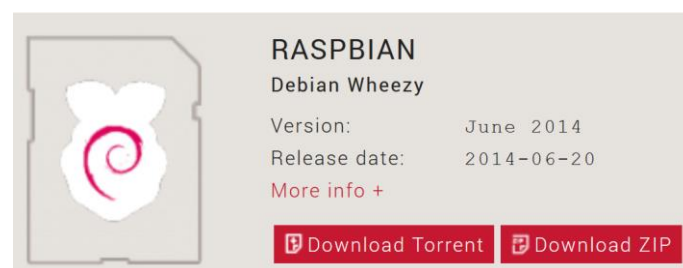
## 2. CONFIGURATION DE SERVEUR WEB LOCAL

### 2.1. INSTALLATION DU SYSTÈME D'EXPLOITATION

#### 2.1.1. Téléchargement de l'image

Pour cette plateforme le système d'exploitation Raspbian, spécialement développé pour le Raspberry Pi, a été utilisé.

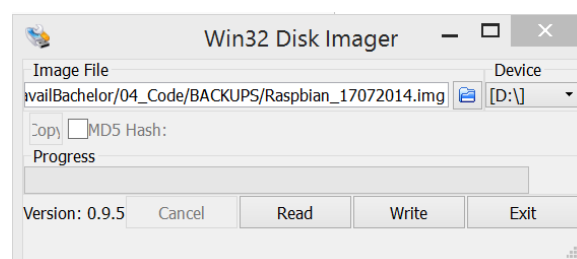
Se rendre sur <http://www.raspberrypi.org/downloads/> et télécharger la dernière version de Raspbian Debian Wheezy.















#### 2.1.2. Écriture de l'image sur la carte SD

Une fois l'image téléchargée insérer la carte SD dans le lecteur de l'ordinateur et ouvrir Win32 Disk Imager afin d'écrire l'image sur la carte.

Sélectionner l'image téléchargée sous **Image File** et la carte SD insérée sous « Device » et cliquer sur **Write**.



À la fin du processus une fenêtre indiquant la fin de l'opération s'affiche. Le contenu de votre carte SD devrait être plus ou moins le suivant :

 bootcode.bin	10.06.2014 09:19	VLC media file (.bin)
 cmdline.txt	07.01.2014 20:13	Document texte
 config.txt	10.06.2014 08:56	Document texte
 fixup.dat	10.06.2014 09:19	Fichier DAT
 fixup_cd.dat	10.06.2014 09:19	Fichier DAT
 fixup_x.dat	10.06.2014 09:19	Fichier DAT
 issue.txt	07.01.2014 22:34	Document texte
 kernel.img	10.06.2014 09:19	Fichier d'image disq...
 LICENSE.oracle	25.09.2013 21:57	Fichier ORACLE
 start.elf	10.06.2014 09:19	Fichier ELF
 start_cd.elf	10.06.2014 09:19	Fichier ELF
 start_x.elf	10.06.2014 09:19	Fichier ELF

### 2.1.3. Modification du fichier « config.txt »

Si vous possédez un écran avec une prise HDMI vous pouvez passer directement au **Chapitre 3**. Autrement afin de pouvoir brancher le Raspberry Pi à un écran VGA il est nécessaire de modifier le fichier **config.txt** présent sur la carte SD.

Ouvrir le fichier **config.txt** et le modifier tel qu'indiqué dans l'image suivante :

```

19
20 # uncomment if hdmi display is not detected and composite is being output
21 hdmi_force_hotplug=1
22
23 # uncomment to force a specific HDMI mode (this will force VGA)
24 hdmi_group=2
25 hdmi_mode=8
26
27 # uncomment to force a HDMI mode rather than DVI. This can make audio work in
28 # DMT (computer monitor) modes
29 hdmi_drive=2
30
31 # uncomment to increase signal to HDMI, if you have interference, blanking, or
32 # no display
33 config_hdmi_boost=4
34

```

Enregistrer et fermer le fichier.

## 2.2. BRANCHEMENT DES DIFFÉRENTS COMPOSANTS

Lors de la première utilisation du Raspberry Pi il est nécessaire de le brancher à un écran et un clavier afin de le paramétrer. Par la suite, l'accès se fera via **SSH** depuis le logiciel **PuTTY** directement depuis l'ordinateur. Le clavier et de l'écran ne seront donc plus nécessaires.

Insérer la carte SD dans le Raspberry Pi et brancher le câble réseau, le clavier et l'écran.



Brancher l'alimentation. Les LEDs clignotent et le Raspberry Pi démarre.



## 2.3. CONFIGURATION DE BASE

### 2.3.1. Raspi-config

Lors du premier démarrage un outil permettant de configurer le Raspberry Pi s'affiche. Grâce à celui-ci, il est possible d'adapter le Raspberry Pi à ses besoins.

```

Raspberry Pi Software Configuration Tool (raspi-config)

Setup Options

1 Expand Filesystem      Ensures that all of the SD card storage is available to the OS
2 Change User Password   Change password for the default user (pi)
3 Enable Boot to Desktop/Scratch Choose whether to boot into a desktop environment, Scratch, or the command-line
4 Internationalisation Options Set up language and regional settings to match your location
5 Enable Camera          Enable this Pi to work with the Raspberry Pi Camera
6 Add to Rastrack        Add this Pi to the online Raspberry Pi Map (Rastrack)
7 Overclock              Configure overclocking for your Pi
8 Advanced Options       Configure advanced settings
9 About raspi-config     Information about this configuration tool

<Select>                  <Finish>

```

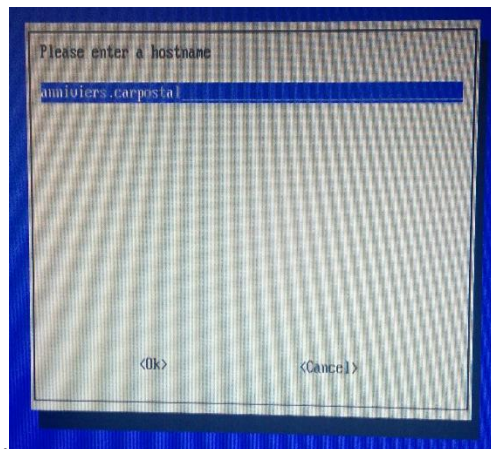
Sélectionner la ligne **1 Expand Filesystem** et appuyer sur la touche « Enter » pour l'ouvrir.



Ouvrir la ligne **2 Change User Password** afin de modifier le mot de passe de l'utilisateur par défaut **pi**.

Ouvrir la ligne **4 Internationalisation Options** et ouvrir **I2** afin de changer de zone et **I3 Change Keyboard Layout** afin de modifier la configuration du clavier. Choisir **Other / German (Switzerland) / German (Switzerland) – French (Switzerland)**.

Ouvrir la ligne **8 Advanced Options** et sélectionner **A2 Hostname** afin de modifier le nom au Raspberry Pi.



Sélectionner la ligne **8 Advanced Options** et ouvrir **A4 SSH** afin d'autoriser la connexion SSH.

Quitter l'outil pour redémarrer le Raspberry Pi.

### 2.3.2. Suppression de wolfram-engine

Suppression de **wolfram-engine** afin d'éviter des messages d'erreurs au démarrage :

```
| sudo apt-get remove wolfram-engine
```

### 2.3.3. Mise à jour du système

Mettre à jour le système.

```
| sudo apt-get update && sudo apt-get upgrade
```

Éteindre le système afin de passer à la suite

```
| sudo halt
```

## 2.4. INSTALLATION DES CLÉS WI-FI

Afin de pouvoir mettre à jour le contenu du serveur, celui-ci doit avoir un accès à internet. La configuration du WI-FI permet au Raspberry Pi de se connecter aux réseaux des offices postaux.

Si des clés WI-FI **Edimax EW-7811Un** sont utilisées, aucun driver ne doit être installé, elles sont directement reconnues par Raspbian. Autrement il est possible qu'un driver spécifique à la clé WI-FI utilisée doive être installé<sup>6</sup>. De plus, certaines clés ne peuvent pas fonctionner en tant que point d'accès WI-FI.

### 2.4.1. Branchement des clés WI-FI

Brancher les deux clés WI-FI et démarrer le Raspberry Pi et contrôler que les clés WI-FI soient reconnues par le système ; **wlan0** et **wlan1** doivent s'afficher.

#### ifconfig

```
root@anniviers:/home/pi# ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:55:33:ce
          inet addr:153.109.141.31  Bcast:153.109.141.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1718 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1441 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1263500 (1.2 MiB)  TX bytes:277292 (270.7 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:1 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:76 (76.0 B)  TX bytes:76 (76.0 B)

wlan0     Link encap:Ethernet  HWaddr 64:66:b3:07:1e:62
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2537 errors:0 dropped:21 overruns:0 frame:0
          TX packets:2352 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:502476 (490.6 KiB)  TX bytes:1610272 (1.5 MiB)

wlan1     Link encap:Ethernet  HWaddr 80:1f:02:af:4b:56
          inet addr:10.1.0.149  Bcast:10.1.1.255  Mask:255.255.254.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2678 errors:0 dropped:82 overruns:0 frame:0
          TX packets:795 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:602781 (588.6 KiB)  TX bytes:121989 (119.1 KiB)
```

<sup>6</sup> Tutoriel pour installer une clé WI-FI « TP-Link tl wn725n », <http://laurenthinoul.com/how-to-install-tp-link-tl-wn725n-on-raspberry-pi/>

### 2.4.2. Configuration des interfaces réseau

Éditer le fichier de configuration des interfaces selon l'image suivante :

```
| sudo nano /etc/network/interfaces
```

```
GNU nano 2.2.6      File: /etc/network/interfaces

auto lo

#
# eth0 config
#
iface lo inet loopback
iface eth0 inet dhcp

#
# wlan0 => DHCP Access Point inside the car
#
allow-hotplug wlan0
iface wlan0 inet static
    address 192.168.1.1
    netmask 255.255.255.0

#
# wlan1 => WI-FI connection for updating content
#
auto wlan1
allow-hotplug wlan1
iface wlan1 inet dhcp
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp
```

L'interface **wlan0** sert de point d'accès à l'intérieur du bus, il utilise un serveur DHCP et possède une adresse IP static. L'interface **wlan1** sert à se connecter à des WI-FI pour mettre à jour le contenu. L'interface **eth0** représente la connexion par câble réseau et à la même fonction que **wlan1**.

Sauvegarder et fermer le fichier en appuyant sur **Ctrl + x, y** et **Enter**.

Attribuer l'adresse IP statique définie dans le fichier pour **wlan0**

```
| sudo ifconfig wlan0 192.168.1.1
```

## 2.5. CONFIGURATION DE WLAN1

Afin que **wlan1** puisse se connecter à des réseaux WI-FI, les caractéristiques de ces derniers doivent être sauvegardées dans le fichier **wpa\_supplicant.conf**.

Éditer le fichier **wpa\_supplicant.conf** selon les caractéristiques des WI-FI. L'image suivante montre des exemples de configuration de réseaux.

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
    ssid="TECHNOPOLE"
    scan_ssid=1
    key_mgmt=NONE
    priority=1
}

network={
    ssid="AndroidAP"
    psk="CarPostal14"

    # Protocol type can be: RSN (for WPA2) and WPA (for WPA1)
    proto=WPA

    # Key management type can be: WPA-PSK or WPA-EAP (Pre-Shared or Enterpr$
    key_mgmt=WPA-PSK

    # Pairwise can be CCMP or TKIP (for WPA2 or WPA1)
    pairwise=TKIP

    #Authorization option should be OPEN for both WPA1/WPA2 (in less common$
    auth_alg=OPEN

    priority=3
}
```

Le réseau **TECHNOPOLE** est un exemple de configuration pour un réseau non sécurisé. Le réseau **AndroidAP** est un exemple de réseau sécurisé par **WPA1**. Il est possible de définir la priorité des réseaux. Dans cet exemple le système essaiera d'abord de se connecter à **AndroidAP** puis à **TECHNOPOLE**, le réseau avec la priorité la plus haute est choisi en premier.

Redémarrer le service réseau

```
nano service networking restart
```

Redémarrer l'interface **wlan1**, s'il est correctement configuré il se connecte au réseau WI-FI précédemment configuré :

```
nano ifdown wlan1 && ifup wlan1
```

```
root@anniviers:/home/pi# ifdown wlan1 && ifup wlan1
Internet Systems Consortium DHCP Client 4.2.2
Copyright 2004-2011 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/wlan1/80:1f:02:af:4b:56
Sending on LPF/wlan1/80:1f:02:af:4b:56
Sending on Socket/fallback
DHCPRELEASE on wlan1 to 10.1.1.1 port 67
iptables has been restored !
ioctl[SIOCSIWAP]: Operation not permitted
ioctl[SIOCSIWENCODEREEXT]: Invalid argument
ioctl[SIOCSIWENCODEREEXT]: Invalid argument
Internet Systems Consortium DHCP Client 4.2.2
Copyright 2004-2011 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/wlan1/80:1f:02:af:4b:56
Sending on LPF/wlan1/80:1f:02:af:4b:56
Sending on Socket/fallback
DHCPDISCOVER on wlan1 to 255.255.255.255 port 67 interval 5
DHCPREQUEST on wlan1 to 255.255.255.255 port 67
DHCPOFFER from 10.1.1.1
DHCPACK from 10.1.1.1
bound to 10.1.0.149 -- renewal in 1786 seconds.
```

En cas d'erreurs, redémarrer le système :

```
sudo reboot
```

Contrôler l'état des interfaces :

```
ifconfig
```

Contrôler que le fichier wpa\_supplicant ne contienne pas d'erreurs :

```
sudo wpa_supplicant -c/etc/wpa_supplicant
/wpa_supplicant.conf -w -iwlan1
```

## 2.6. CONFIGURATION DE WLAN0 (POINT D'ACCÈS WI-FI)

Un serveur **DHCP** doit être installé afin que les passagers du bus obtiennent une adresse IP lorsqu'ils se connectent au WI-FI.

### 2.6.1. Installation du logiciel

Installer le serveur **DHCP** pour configurer le point d'accès WI-FI :

```
sudo apt-get install hostapd isc-dhcp-server
```

```
pi@anniviers ~ $ sudo apt-get install hostapd isc-dhcp-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  isc-dhcp-server-ldap
The following NEW packages will be installed:
  hostapd isc-dhcp-server
0 upgraded, 2 newly installed, 0 to remove and 1 not upgraded.
Need to get 1,305 kB of archives.
After this operation, 2,783 kB of additional disk space will be used.
Get:1 http://mirrordirector.raspbian.org/raspbian/ wheezy/main hostapd armhf 1:1.0-3 [419 kB]
Get:2 http://mirrordirector.raspbian.org/raspbian/ wheezy/main isc-dhcp-server armhf 4.2.2.dfsg.1-5+deb70u6 [887 kB]
Fetched 1,305 kB in 1s (1,176 kB/s)
Preconfiguring packages ...
Selecting previously unselected package hostapd.
(Reading database ... 68817 files and directories currently installed.)
Unpacking hostapd (from .../hostapd_1%3a1.0-3_armhf.deb) ...
Selecting previously unselected package isc-dhcp-server.
Unpacking isc-dhcp-server (from .../isc-dhcp-server_4.2.2.dfsg.1-5+deb70u6_armhf.deb) ...
Processing triggers for man-db ...
Setting up hostapd (1:1.0-3) ...
insserv: warning: script 'mathkernel' missing LSB tags and overrides
Setting up isc-dhcp-server (4.2.2.dfsg.1-5+deb70u6) ...
Generating /etc/default/isc-dhcp-server...
insserv: warning: script 'mathkernel' missing LSB tags and overrides
[FAIL] Starting ISC DHCP server: dhcpcd[....] check syslog for diagnostics. ... failed!
failed!
invoke-rc.d: initscript isc-dhcp-server, action "start" failed.
```

Un message d'erreur apparaît à la fin de l'installation lors du démarrage. C'est normal étant donné qu'il n'est pas encore configuré.

### 2.6.2. Configuration du serveur DHCP

Éditer le fichier de configuration du serveur selon l'image suivante :

```
| sudo nano /etc/dhcp/dhcpd.conf
```

```
# option definitions common to all supported networks...
#option domain-name "example.org";
#option domain-name-servers ns1.example.org, ns2.example.org;

default-lease-time 600;
max-lease-time 7200;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
authoritative;
```

Ajouter à la fin du fichier les lignes suivantes qui permettent de définir les caractéristiques du réseau :

```
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.10 192.168.1.200;
    option broadcast-address 192.168.1.255;
    option routers 192.168.1.1;
    default-lease-time 600;
    max-lease-time 7200;
    option domain-name "local";
    option domain-name-servers 8.8.8.8, 8.8.4.4;
}
```

### 2.6.3. Configurer le réseau WI-FI

Créer un fichier de configuration en insérant les informations suivantes :

```
sudo nano /etc/hostapd/hostapd.conf
```

```
interface=wlan0
driver=rtl871xdrv
ssid=CarPostalEntertainment
hw_mode=g
channel=6
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
```

Le driver mentionné fonctionne avec les clés Edimax. En cas d'utilisation d'autres clés, le driver devra être changé.

Modifier le fichier suivant :

```
sudo nano /etc/default/hostapd
```

Éditer la ligne commençant par **DAEMON\_CONF** :

```
DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

### 2.6.4. Configurer le NAT

Afin d'autoriser plusieurs passagers à se connecter au réseau WI-FI et d'avoir accès à internet il faut configurer le **NAT**. Le but est de faire correspondre les adresses IP internes du réseau aux adresses externes.

Éditer le fichier **/etc/sysctl.conf** en enlevant le **#** devant la ligne suivante :

```
sudo nano /etc/sysctl.conf
```



```
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
```

Lancer la commande suivante pour activer le **NAT** immédiatement :

```
sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"
```

Lancer les commandes suivantes afin de créer les correspondances **wlan0** <-> **wlan1** et **wlan0** <-> **eth0** dans la table NAT :

```
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
sudo iptables -t nat -A POSTROUTING -o wlan1 -j MASQUERADE
sudo iptables -A FORWARD -i eth0 -o wlan0 -m state --state
RELATED,ESTABLISHED -j ACCEPT
sudo iptables -A FORWARD -i wlan1 -o wlan0 -m state --state
RELATED,ESTABLISHED -j ACCEPT
sudo iptables -A FORWARD -I wlan0 -o eth0 -j accept
sudo iptables -A FORWARD -I wlan0 -o wlan1 -j accept
```

Exécuter la commande suivante afin de sauvegarder la table dans un fichier :

```
sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"
```

Afin de restaurer la table après chaque modification de l'état de **wlan1** ou **eth0**, créer un script dans le dossier **/etc/network/if-pre-up.d** :

```
sudo nano /etc/network/if-pre-up.d/restoreIptables
```

Insérer le code suivant dans le fichier :

```
#!/bin/sh
#Restore iptables
iptables-restore < /etc/iptables.ipv4.nat
echo "iptables has been restored !"
```

Modifier les droits et rendre le fichier exécutable :

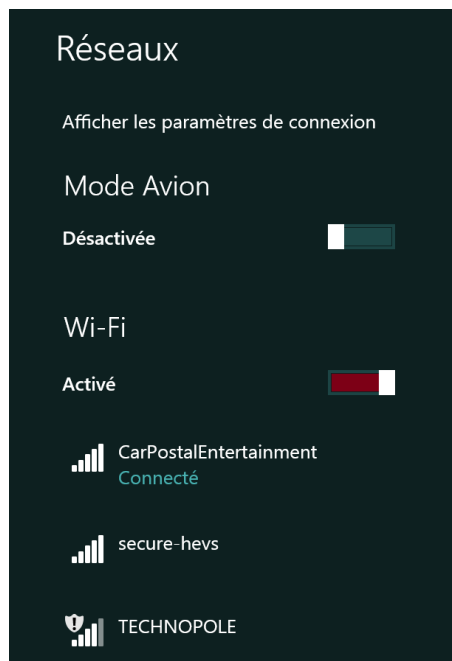
```
chmod 777 /etc/network/if-pre-up.d
chmod +x /etc/network/if-pre-up.d
```

Redémarrer le système

```
| sudo reboot
```

### 2.6.5. Test du point d'accès

Se connecter au WI-FI **CarPostalEntertainment** qui apparaît dans la liste des réseaux disponibles.



Tester que la connexion internet fonctionne depuis **wlan1** en désactivant **eth0** :

```
| sudo ifdown eth0
```

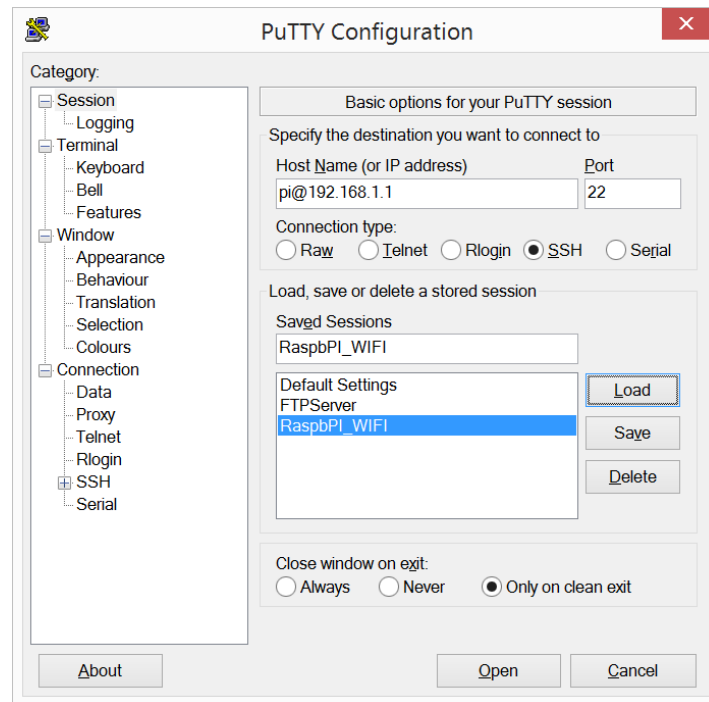
Tester que la connexion internet fonctionne depuis **eth0** en désactivant **wlan1** :

```
| sudo ifdown wlan1 && ifup eth0
```

## 2.7. ACCÈS AU SYSTÈME PAR SSH

À partir de maintenant il est possible de se connecter au Raspberry Pi via **SSH** directement depuis l'ordinateur grâce au logiciel **PuTTY**. L'écran et le clavier ne sont plus nécessaires.

Ouvrir **PuTTY** et entrer l'adresse IP du serveur:



Cliquer sur **Save** pour sauvegarder les infos, puis sur **Open** pour ouvrir une session.



La session s'ouvre et le mot de passe est demandé.

## 2.8. CONFIGURATION DU SERVEUR WEB

### 2.8.1. Installation de Nginx et PHP

**Nginx** est le serveur web qui va héberger le contenu de la plateforme à l'intérieur du car. La plateforme est développée en **PHP**.

Installer **Nginx** et **PHP**

```
sudo apt-get install nginx
sudo service nginx start
sudo apt-get install php5-fpm php-apc
sudo service php5-fpm restart
```

Créer le dossier qui va contenir la plateforme :

```
sudo mkdir /var/www/CarPostalEntertainment
```

Stopper le serveur et désactiver le site par défaut :

```
sudo service nginx stop
sudo unlink /etc/nginx/sites-enabled/default
```

### 2.8.2. Création d'un nouveau site

Créer un nouveau site en créant un nouveau fichier :

```
sudo nano /etc/nginx/sites-available/carpostal
```

Ajouter ce contenu à l'intérieur :

```
server {
    listen 80;

    root /var/www/CarPostalEntertainment;

    index index.html index.php;

    server_name carpostal.local;

    location ~ /\.php$ {
        fastcgi_pass unix:/var/run/php5-fpm.sock;
        fastcgi_index index.php;
        include fastcgi_params;
    }
}
```

Activer le nouveau site :

```
cd /etc/nginx/sites-enabled
sudo ln -s ../sites-available/carpostal
sudo service nginx start
```

## 2.9. CONFIGURATION DE LA BASE DE DONNÉES

Le système de gestion de base de données **MySQL** est utilisé pour la plateforme.

### 2.9.1. Installation de MySQL

Installer **MySQL** avec **PHP** :

```
sudo apt-get install mysql-server
sudo apt-get install php5-mysql
```

Durant installation **MySQL** demande la saisie d'un mot de passe pour l'utilisateur **root**.

### 2.9.2. Création d'une nouvelle base de données

Se connecter à **MySQL** :

```
sudo su
mysql -p
```

Et entrer le mot de passe choisi durant l'installation.

Créer la base de données **CarPostalEntertainmentDB** :

```
CREATE DATABASE CarPostalEntertainmentDB
DEFAULT CHARACTER SET utf8
DEFAULT COLLATE utf8_general_ci ;
```

Par la suite cette base de données sera remplie et mise à jour depuis le serveur web central.

## 2.10. CRÉATION DES SCRIPTS

Des scripts ont été créés afin de mettre à jour le contenu de la plateforme et de contrôler son fonctionnement.

Ouvrir le dossier allant contenir les scripts :

```
sudo su
cd /usr/local/bin
```

### 2.10.1. Mise à jour du contenu (update.sh)

Créer le script **update.sh** qui va permettre la mise à jour du contenu :

```
nano update.sh
```

Insérer le contenu suivant :

```
# !/bin/bash

rsync -avvvz --progress --partial-dir=.temp -e "ssh -i
/home/pi/FTPServer2.pem"
admin@54.191.4.233:/home/admin/CarPostalEntertainment /var/www
```

La commande **rsync** est utilisée afin de synchroniser le contenu du serveur web local avec le serveur central. La connexion est sécurisée par SSH, il est nécessaire d'indiquer le chemin vers la clé publique permettant l'authentification sur le serveur central. La procédure pour obtenir cette clé est mentionnée dans le **Chapitre 3.1 SSH**.

### 2.10.2. Mise à jour de la base de données (updateDB.sh)

Créer le script **updateDB.sh** qui va permettre la mise à jour de la base de données:

```
nano updateDB.sh
```

Insérer le contenu suivant :

```
# !/bin/bash

mysql -h localhost -u root -pD4Ts24Ds5 CarPostalEntertainmentDB <
/var/www/CarPostalEntertainment/db/CarPostalEntertainmentDB.sql
```

MySQL met à jour la base de données locale en utilisant **CarPostalEntertainmentDB.sql** qui a été synchronisé grâce à **update.sh**.

### 2.10.3. Envoi des commentaires et des feedbacks (sendComments.sh)

Créer le script **sendComments.sh** qui va permettre l'envoi des commentaires et des feedbacks sur le serveur central :

```
nano sendComments.sh
```

Insérer le contenu suivant :

```
# !/bin/bash

#Export the tables messages and feedbacks into
comments.sqlmysqldump -u root -pD4Ts24Ds5 CarPostalEntertainmentDB
messages feedbacks >
/var/www/CarPostalEntertainment/db/comments.sql

#Sync comments.sql with the main serverrsync -avvz --progress -e
"ssh -i /home/pi/FTPServer2.pem"
/var/www/CarPostalEntertainment/db/comments.sql
admin@54.191.4.233:/home/admin/CarPostalEntertainment/db/comments.
sql
```

Les commentaires et les feedbacks des passagers sont enregistrés dans la base de données. Lors du lancement du script les tables **messages** et **feedbacks** sont exportées dans le fichier **comments.sql** qui est synchronisé avec le serveur central grâce à **rsync**.

#### 2.10.4. Envoi des fichiers log (sendLogs.sh)

Créer le script **sendLogs.sh** qui va permettre l'envoi des logs sur le serveur central :

```
nano sendLogs.sh
```

Insérer le contenu suivant :

```
# !/bin/bash

echo "Sending nginx log..."

rsync -avvz --progress -e "ssh -i /home/pi/FTPServer2.pem"
/var/log/nginx/error.log
admin@54.191.4.233:/home/admin/LogsFromServers/nginx_"$HOSTNAME".log

echo "Sending system log..."

rsync -avvz --progress -e "ssh -i /home/pi/FTPServer2.pem"
/var/log/syslog
admin@54.191.4.233:/home/admin/LogsFromServers/syslog_"$HOSTNAME".log
```

Les logs de **Nginx** et du système sont synchronisés avec le serveur central grâce à **rsync**, afin de détecter les éventuelles erreurs.

#### 2.10.5. Script central (main.sh)

Le script **main.sh** permet de vérifier qu'une connexion internet est disponible avant de lancer les différents autres scripts précédemment créés.

Créer le script :

```
nano main.sh
```



Insérer le contenu suivant :

```
# !/bin/bash

# Get the current date

now=$(date)

# Checking the internet connection, if up the script continue,
else attempting to restart eth0 and wlan1, if still down exit the
script

echo "Script started at "$now

echo "Checking the internet connexion..."

x=`ping -c1 173.194.45.37 2>&1 | grep unreachable`

if [ ! "$x" = "" ];
then

    echo "It's down!! Attempting to restart eth0..."

    /sbin/ifdown eth0 && /sbin/ifup eth0

    sleep 2

    y=`ping -c1 173.194.45.37 2>&1 | grep unreachable`

    if [ ! "$y" = "" ];
    then

        echo "It's still down, attempting to restart
wlan1..."

        /sbin/ifdown wlan1 && /sbin/ifup wlan1

        y=`ping -c1 173.194.45.37 2>&1 | grep unreachable`

        if [ ! "$y" = "" ];
        then

            echo "It's still down, no connexion
available! The script has been stopped at "$now

            exit
```

```
        fi

    fi

else

    echo "Network's up!! The script continue..."
fi

# Executing the script for updating the content of the web server
echo "Updating the content..."
/usr/local/bin/update.sh

# Executing the script for updating the database
sleep 3
echo "Updating the database..."
/usr/local/bin/updatedB.sh

# Executing the script for sending news comments
echo "Sending news comments..."
/usr/local/bin/sendComments.sh

# Executing the script for sending logs
echo "Sending logs..."
/usr/local/bin/sendLogs.sh

# Finishing
wait
echo "The script finished at "$now
```

### 2.10.6. Lancement automatique du script

Afin que le script soit exécuté automatiquement le programme **cron** est utilisé. Il permet d'enregistrer l'exécution d'un script à des moments définis.

Ouvrir **crontab** :

```
| crontab -e
```

Insérer les lignes suivantes à la fin du fichier :

```
| # Execute main.sh at start up and every 5 min
| @reboot bash /usr/local/bin/main.sh
| */5 * * * * bash /usr/local/bin/main.sh
```

Le script **main.sh** est exécuté au démarrage de la plateforme et toutes les cinq minutes par la suite, si une connexion à internet est disponible.

Relancer le service afin de prendre en compte les modifications :

```
| /etc/init.d/cron reload
```

## 2.11. INSTALLATION DE L'UPS

L'UPS permet de gérer l'alimentation du Raspberry Pi. Lors de l'arrêt du car l'alimentation est coupée et l'UPS prend le relai afin d'alimenter le Raspberry Pi pour qu'il ait le temps de s'arrêter de manière sûre. L'UPS utilisé a été spécialement développé pour le Raspberry Pi. La batterie est composée de six piles.

Éteindre le Raspberry Pi

```
| halt
```

Brancher l'UPS sur le Raspberry PI.



Connecter l'alimentation dans le port micro-USB de l'UPS afin de démarrer le Raspberry Pi.

Des LEDs indiquent l'état du système :

- **Verte** => le système est alimenté par la prise micro-USB.
- **Orange** => la batterie n'a plus la puissance nécessaire pour alimenter le Raspberry Pi en cas de coupure de courant, il faut remplacer les piles.
- **Rouge** => le système fonctionne sur batterie.

Installer le logiciel

```
sudo su
wget http://www.piups.net/support/piusvmonitor.deb
dpkg -i piusvmonitor.deb
```

Le logiciel contrôle l'alimentation, si elle est coupée il va arrêter le Raspberry Pi après une dizaine de secondes.

### 3. CONFIGURATION DU SERVEUR CENTRAL

Le serveur central contient la dernière version de la plateforme. Le serveur à l'intérieur du car s'y connecte afin de mettre à jour son contenu. Ce tutoriel a été réalisé sur un serveur hébergé par **Amazon EC2**<sup>7</sup>.

#### 3.1. SSH

Afin de sécuriser les connexions le serveur doit générer une paire de clés ; une clé privée qu'il conservera et une clé publique qui sera transmise au serveur distant devant se connecter. Ces clés sont générées automatiquement par **Amazon EC2** lors de la création d'une nouvelle instance. Pour les autres cas, la procédure est décrite à cette adresse : <http://www.linux-france.org/prj/edu/archinet/systeme/ch13s03.html>.

#### 3.2. ADRESSE IP FIXE

Le serveur doit posséder une adresse IP fixe afin d'être accessible.

#### 3.3. CONFIGURATION DU SERVEUR FTP

Le serveur FTP va permettre de gérer le code source, ainsi que le contenu de plateforme.

##### 3.3.1. Installation du serveur

Installer le serveur FTP **vsftpd** :

```
sudo su
apt-get install vsftpd
```

Éditer le fichier de configuration du serveur :

```
nano /etc/vsftpd.conf
```

Vérifier que les lignes suivantes correspondent :

```
# Run standalone? vsftpd can run either from an inetd or as a standalone
# daemon started from an initscript.
listen=YES
```

<sup>7</sup> Service permettant de louer des serveurs,  
<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html>

```
# Allow anonymous FTP? (Beware - allowed by default if you comment this out).
anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
local_umask=022
```

Redémarrer le serveur :

```
service vsftpd restart
```

### 3.3.2. Création d'un nouvel utilisateur

Cet utilisateur sera utilisé par le serveur local afin d'accéder au contenu du serveur FTP. Le répertoire de l'utilisateur contiendra la dernière version du code source et du contenu de la plateforme.

Créer l'utilisateur et lui donner un mot de passe :

```
adduser admin
```

Créer un nouveau répertoire allant contenir la dernière version de la plateforme :

```
mkdir /home/admin/CarPostalEntertainment
```

Le serveur local synchronise son contenu avec le contenu de ce répertoire. Il est indispensable que la version de la plateforme présente dans ce répertoire soit fonctionnelle à 100%.

Modifier les droits du répertoire :

```
chmod -R 755 /home/admin
```

### 3.4. CONFIGURATION DU SERVEUR WEB [FACULTATIF]

Cette partie est facultative, elle n'est pas nécessaire au bon fonctionnement de la plateforme.

Dans le cadre de ce travail, le serveur central sert également de serveur de développement et de démo. Pour ce faire Nginx et MySQL sont utilisés. Pour les installer il suffit de reprendre les étapes des **Chapitres 0 et 2.9**.

## RÉFÉRENCES

### RASPBERRY PI

Raspbian, <http://www.raspberrypi.org/downloads/>, consulté le 10.06.2014

Configuration du point d'accès, <https://learn.adafruit.com/setting-up-a-raspberry-pi-as-a-wifi-access-point/overview>, consulté le 10.06.2014

Installer une clé WI-FI TP-Link, <http://laurenthinoul.com/how-to-install-tp-link-tl-wn725n-on-raspberry-pi/>, consulté le 04.07.2014

Configuration des clés WI-FI, <http://www.processthings.com/post/66023171876/how-to-connect-your-raspberry-pi-to-two-wi-fi-networks>, consulté le 04.07.2014

### SERVEUR WEB LOCAL

Installation de Nginx, <http://www.ducky-pond.com/posts/2013/Sep/setup-a-web-server-on-rpi/>, consulté le 10.06.2014

### BASE DE DONNÉES MYSQL

Installation de MySQL, <http://www.ducky-pond.com/posts/2014/Feb/how-to-install-and-optimize-mysql-on-raspberry-pi/>, consulté le 10.06.2014

Création d'une nouvelle base de données, <http://dev.mysql.com/doc/refman/5.0/en/charset-applications.html>, consulté le 10.06.2014

### SCRIPTS

Script batch, <http://fr.openclassrooms.com/informatique/cours/reprenez-le-controle-a-l-aide-de-linux/introduction-aux-scripts-shell>, consulté le 26.06.2014

Rsync, <https://www.digitalocean.com/community/tutorials/how-to-copy-files-with-rsync-over-ssh>, consulté le 26.06.2014

Cron, <http://fr.openclassrooms.com/informatique/cours/reprenez-le-controle-a-l-aide-de-linux/executer-un-programme-a-une-heure-differee>, consulté le 26.06.2014

## UPS

Configuration, <http://www.piups.net/support/>, consulté le 24.07.2014

## SERVEUR CENTRAL

Serveur FTP, <http://www.debianaddict.org/article47.html>, consulté le 26.06.2014