



Institut Universitaire
de Technologie
Aix*Marseille Université



RAPPORT DE STAGE
Diplôme Universitaire de Technologie
Spécialité Réseaux et Télécommunications

Développement web et administration
systèmes & réseaux

Hugo Blachere

Institut de Mathématiques de Marseille

Responsable entreprise : Olivier Chabrol

Responsable académique : Tin Nguyen

2019

Remerciements

Je tiens à remercier tout d'abord toutes les personnes qui ont contribué à la réalisation de mon stage de près ou de loin de façon générale ainsi que toute l'équipe de l'I2M Marseille pour son accueil et son amabilité au cours de ces 10 semaines de stage.

J'aimerais également remercier monsieur Pierre BARTHELEMY pour avoir retenu ma candidature et sans qui ce stage n'aurait pas été réalisable.

Plus particulièrement, un grand merci à Olivier CHABROL, qui à travers son rôle de tuteur a pu m'enseigner, les enjeux, les pratiques ainsi que les techniques professionnelles nécessaires à l'accomplissement de mes missions. Il a également pu partager ses connaissances ainsi que son expérience dans ce milieu, le tout dans la bonne humeur et avec une méthodologie exemplaire.

Je remercie également Guillaume CHAGNARD ainsi que Augustino DE SOUZA pour leur accueil et leurs conseils.

De même, je remercie Monsieur Tin NGUYEN pour son encadrement tout au long de ce stage et pour l'ensemble de ses conseils professionnels plus généralement.

Enfin Maximilien GUERRERO, mon binôme pour son travail fourni ainsi que sa bonne humeur, avec qui j'ai effectuée l'intégralité de mon stage.

Table des matières

I. Introduction.....	7
I.1. Préambule	7
I.2. Présentation de l'entreprise.....	7
I.3. Présentation du cadre de travail.....	8
I.4. Présentation du réseau informatique.....	8
I.5. Présentation des missions	10
II. Présentation des travaux réalisés	11
II.1 Configuration des switches.....	11
II.2 Préparation au développement.....	12
II.3 Développement avec Symfony.....	13
II.4 Développement d'une application DHCP	14
II.4.A Déploiement du serveur web.....	15
II.4.B Correction des erreurs	16
II.5 Développement de l'application Keyring.....	20
II.5.A Conception	20
II.5.B Développement de l'application	21
II.6 Le VPN avec OPEN-VPN.....	23
II.6.A Serveur VPN	23
II.6.B Clients VPN	24
II.7 Mise a jour serveurs I2M.....	24
II.8 Maintenance générale.....	25
II.8.A Installation d'un point d'accès sans fil.....	25
II.9 Projets personnels	26
III. Conclusion.....	27
IV. Glossaire.....	28
V Annexes	30
Configuration de l'hote virtuel :	30
Cahier des charges de l'application Keyring :	30
Diaporama comparatif Docker & Proxmox :	31
Plan réseau détaillé, Site de Chateau-Gombert :	31
Liste des VLANs sur les switches :	31
VI Bibliographie - Sitographie.....	32

I. Introduction

I.1. Préambule

Ce stage a été effectué dans le cadre de ma seconde année de **DUT*** en réseaux et télécommunications au pôle de recherche **I2M*** Marseille à Château-Gombert, il m'aura permis de valider mes acquis, d'apprendre énormément professionnellement au sein de l'entreprise ainsi que d'avoir une première réelle expérience professionnelle prolongée au sein d'une entreprise dans le domaine des réseaux et de l'informatique en général. Lors de ce stage, j'ai pu être accompagné de Maximilien GUERRERO avec qui nous avons postulé pour ce stage, en tant que binôme.

I.2. Présentation de l'entreprise

L'institut de Mathématiques de Marseille (I2M, UMR 7373) est une unité de recherche mixte, formé suite à la fusion du LATP (Laboratoire d'Analyse, Topologie et Probabilités) et de l'IML (Institut de Mathématiques de Luminy) au 1er janvier 2014. L'institut est en co-tutelle d'Aix-Marseille Université, Ecole centrale Marseille ainsi que du CNRS.

Ainsi, divisé en trois grands sites, il forme un grand ensemble de recherche sur Marseille : Luminy, Saint-Charles & Château-Gombert. Il comporte environ 300 Membres de personnel dont 130 enseignant-chercheurs, une soixantaine de doctorants, une vingtaine de chercheurs post-doctorants et des membres administratifs.

Vous retrouverez ci-dessous l'organigramme hiérarchique de l'I2M :

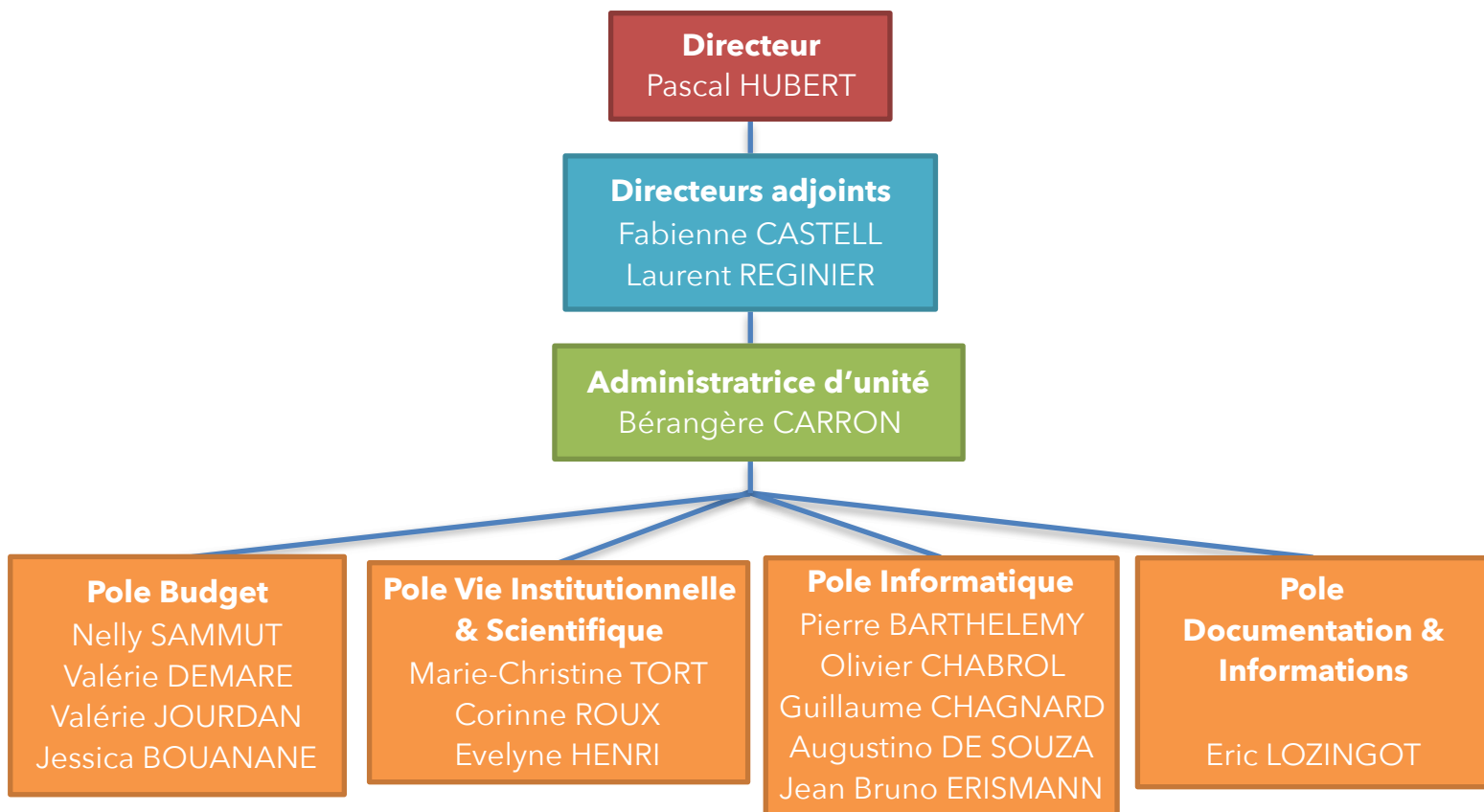


Figure 1 : Organigramme synthétique de l'I2M

I.3. Présentation du cadre de travail

Au nord de Marseille, sur le site de Château-Gombert que nous occupons principalement, les locaux se situent à quelques dizaines de mètres seulement de l'école centrale Marseille, le rez-de-chaussée est destiné à l'enseignement tandis que les étages supérieurs sont orientés recherche. Notre bureau quand a lui se situe au premier étage, pôle informatique, il accueille généralement 3 membres de personnel en continu. Cette pièce est divisée en deux, nous occupons une des de ces parties. Nous disposons de deux postes fixes sous **Ubuntu*** ainsi que des périphériques nécessaires, des casques antibruit sont également à notre disposition pour l'accès aux locaux technique et salles serveur.

Il existe aussi un service permettant de maintenir à jour les activités et les configurations mises en place au sein du réseau informatique de l'I2M : **DokuWiki***, un site web sur lequel les administrateurs réseau et informatique peuvent se connecter pour y retrouver l'historique des services et des configurations mises en place. Dans notre cas, on l'utilisera pour y maintenir un « Journal de bord » de notre stage.

Sur le site de Luminy, nous occupons le bâtiment TPR2, une pièce destinée aux invités est mise à notre disposition, nous l'occupons lorsque nous sommes en déplacement sur ce site.

Vous pourrez trouver en annexes de plus amples documents et de plus nombreuses photos concernant le stage et certains détails plus spécifiques.

I.4. Présentation du réseau informatique

Nous parlerons ici principalement de l'architecture réseau de Château-Gombert, notre site principal de travail, nous aborderons également le site de Luminy brièvement. Les autres sites ne seront pas étudiés en raison des informations manquantes dont nous n'avons pas pu prendre compte à ce sujet et du peu de temps passé sur ces sites.

A l'I2M, la majorité des stations de travail et des chercheurs utilisent le système d'exploitation **OSX***, développé par Apple, la part de personnel utilisant ce système d'exploitation est estimée à 66% alors que 32% utilisent Linux, seulement 2% des machines présente utilisent Windows.

Le réseau informatique quand à lui est divisé en 3 parties majeures, ces parties sont indépendantes les unes des autres, d'un côté nous avons le pôle Saint-Charles, qui possède sa propre topologie, le pôle de Luminy, que nous étudierons un peu plus en profondeur ainsi que le pôle de Château-Gombert sur lequel nous avons effectué la majorité de notre stage. Tout ces pôles sont régis par l'autorité de la **DOSI*** qui veille au bon fonctionnement et à la disponibilité ainsi que l'intégrité du réseau global et du fonctionnement des services applicatifs internes pour l'enseignement et la recherche.

Le responsable actuel du service informatique, Olivier Chabrol est présent dans l'institut depuis peu de temps seulement. Héritant d'un parc informatique vieillissant et résultant de la fusion de trois anciens laboratoires possédants chacun leurs propres services et politiques de gestion et de sécurité, la tâche est rendu d'autant plus complexe. De plus, les équipements comme les **routeurs*** ou **commutateurs*** sont à la propriété de la DOSI, mais ceux présent destinés à la partie recherche sont sous couvert de notre pôle informatique ce qui complexifie la gestion du réseau en général et qui nous donne un aperçu plutôt limité.

Sur le site de Château-Gombert le réseau **LAN*** est réparti sur la totalité du complexe, le bâtiment dispose de nombreux locaux techniques, un ou plusieurs à chaque étage pouvant contenir soit un salle serveur, une baie de brassage ou encore une zone de stockage.

Tous les sites sont indépendants, mais les même services sont généralement utilisés sur l'ensemble des complexes.

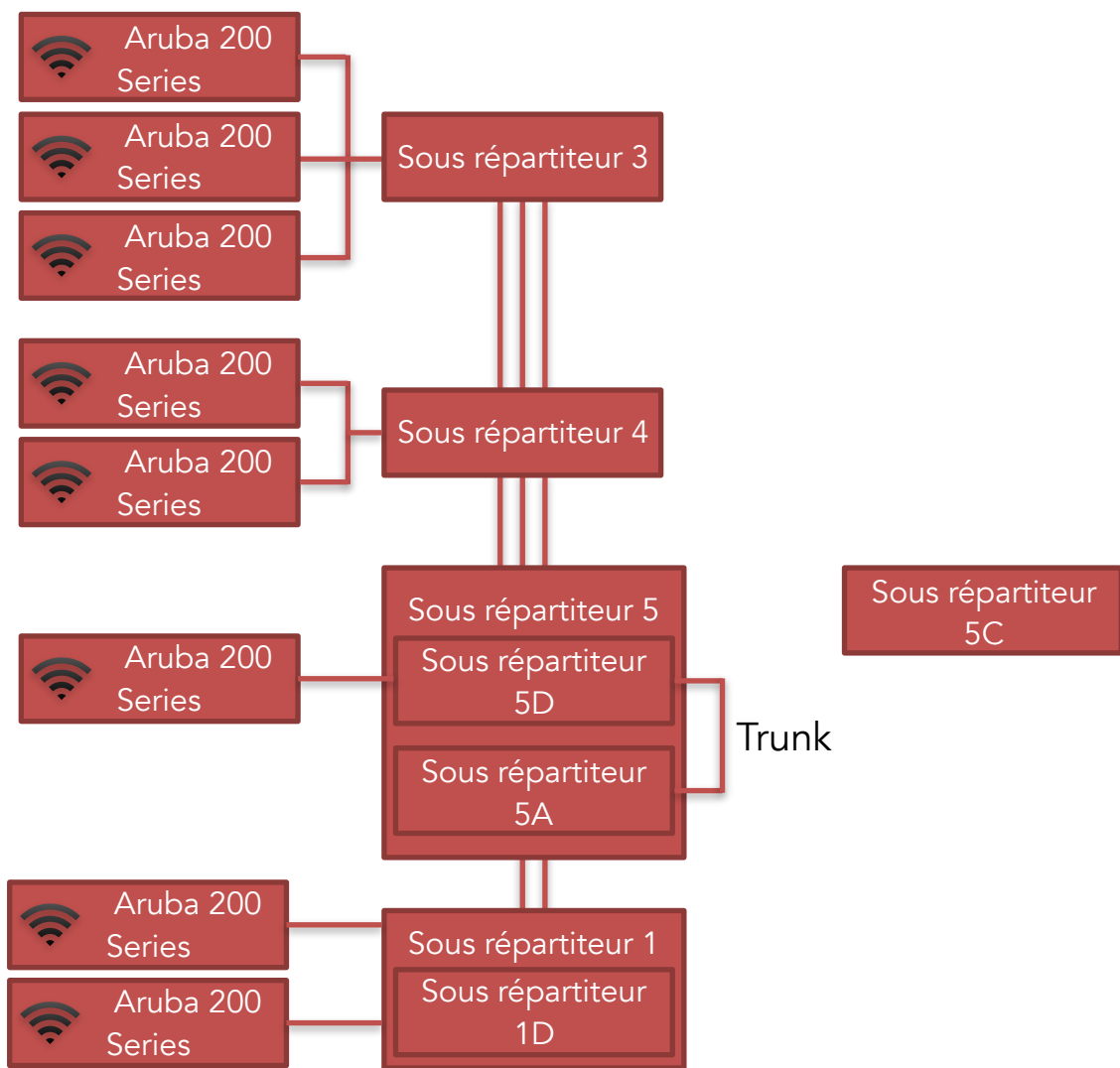


Figure 2 : Schéma topologique de l'infrastructure réseau sur le site de Château-Gombert

Quand au site de Luminy, il est conçu suivant une topologie dite « en étoile », les éléments réseau sont connectés à un élément central aussi appelé « noeud », ce noeud peut aussi bien être un concentrateur qu'un routeur. Cette architecture assez ancienne tout de même facilite l'ajout de nouveaux postes informatique par exemple mais si le noeud viens à faillir alors tout le réseau tombe en panne.

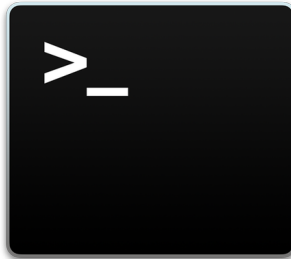
I.5. Présentation des missions

Mon stage à l'institut de mathématiques de Marseille a été composé de nombreuses missions aussi variées qu'intéressantes, permettant de découvrir les nombreuses facettes et débouchés possibles quand au métiers de l'informatique et des réseaux.

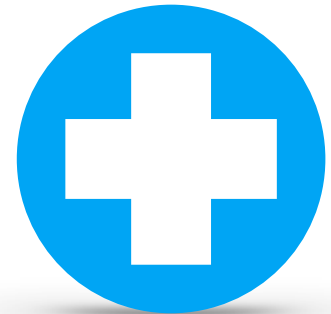
Mes principales missions au cours de ce stage peuvent être regroupées sous trois catégories principales :



Programmation



Administration réseaux et systèmes



Assistance

Figure 3 : Schéma résumant l'ensemble de mes missions de stage

- **Assistance aux utilisateurs** : Dans nos locaux du pôle informatique, nous sommes souvent sollicités par les utilisateurs, notre rôle est de répondre à leurs requêtes, dépanner et porter assistance aux usagers. Les demandes concernent généralement le changement ou remplacement de leur matériel informatique pour du matériel plus récent, des dépannages de leur machines ou de leur sites personnels ou encore de la simple assistance logicielle basique.
- **Maintenance du réseau informatique** : Le rôle du pôle informatique de l'I2M est également d'assurer la maintenance du réseau informatique sur l'ensemble des sites, certaines missions ont donc été orientés vers la maintenance et la bonne propagation de ce dernier.
- **Programmation Web** : Au cours de ce stage, certaines missions nous ont été confiées, concernant l'établissement d'applications afin de faciliter certaines tâches administratives. Ces applications sont généralement des réponses à des demandes du personnel.
- **Administration systèmes** : Mise en place de divers services tel que un **VPN*** pour permettre l'accès au réseau local depuis l'extérieur du réseau local, mises à jour et administration de serveur, ou d'autres tâches de maintenance système.

II. Présentation des travaux réalisés

II.1 Configuration des switchs

Les bâtiments de recherche sur le site de Chateau-Gombert étant des infrastructures anciennes, le réseau l'est également, de ce fait les configurations sur le matériel réseau sont encombrés de configurations successives ajoutés à travers le temps. Notre première mission consistait donc à vérifier les configurations des switchs sur l'ensemble du bâtiment et d'y supprimer les **vlan**s* inutilisés. La majorité des commutateurs sont de la marque **HP*** seulement les quelques autres seront des **Cisco***.

Pour cela on se connecte en **SSH*** au switchs depuis notre invité de commande du terminal en spécifiant l'option de commande « -p » qui précise le port sur lequel se connecter.

```
ssh -p 2262 manager@147.94.65.58  
show vlans
```

```
s2610pwr-CMI_6a# show vlans  
  
Status and Counters - VLAN Information  
  
Maximum VLANs to support : 32  
Primary VLAN : DEFAULT_VLAN  
Management VLAN :  
  
VLAN ID Name | Status Voice Jumbo  
-----+-----  
1 DEFAULT_VLAN | Port-based No No  
21 CMI-rech | Port-based No No  
22 CMI-ens | Port-based No No  
24 CMI-sunray | Port-based No No  
26 sunray-kiosk | Port-based No No  
27 jumpstart-re | Port-based No No  
29 vlansgd | Port-based No No  
114 PedagoPriv | Port-based No No  
129 TOIP-old | Port-based No No  
200 TOIP2 | Port-based Yes No
```

Figure 4 : Liste des vlans sur le switch, obtenu avec la commande « show vlans »

Néanmoins la procédure a été plus compliqué sur les switchs les plus anciens, en effet il a été nécessaire de spécifier l'**algorithme de chiffage*** utilisé par ssh sur le switch pour avoir ne serait-ce qu'une réponse de sa part. L'option ci-dessous a donc du être ajoutée a la commande de connexion :

```
-c 3des-cbc
```

L'option « -c » signifiant « cypher » pour « chiffage » en français permet de spécifier la méthode de chiffage utilisée, dans notre cas la méthode 3des-cbc est utilisée.

Mais la encore, la connexion est refusée, la clé de **chiffage*** cause maintenant problème, elle est reconnu trop petite par notre **client*** ssh utilisé (Open-ssh), qui refuse donc de se connecter par

soucis de sécurité. Nous avons donc du utiliser un autre client, en l'occurrence putty qui n'impose pas de restrictions de sécurité au niveau de la longueur de la clé.

Une fois ces deux actions menées, l'accès au switches était maintenant possible pour l'ensemble du site, il a donc été possible de lister les vlans présents et de les supprimer facilement par la suite avec la commande :

```
no vlan idvlan
```

Un **backup*** des switches a bien évidemment été effectué avant la suppression des vlans pour pouvoir récupérer leur configuration en cas d'erreur, pour ce faire il aura fallu copier/coller leur configuration dans un fichier texte. Pour visualiser la configuration des switches, on utilisera la commande :

```
show running-config
```

Toutes les modifications, édition et fichiers de configuration seront mis a jour sur le **wiki*** de l'I2M pour que les autres administrateurs du pôle informatique soient avertis des modifications et aient une traçabilité des événements.

II.2 Préparation au développement

Les projets qui nous auront été attribués par la suite concernent l'élaboration de deux applications web, nécessitant par conséquent la mise en place d'un « environnement de développement ». C'est a dire un ensemble d'outils et de pré-requis afin de pouvoir débiter la programmation de nos applicatifs. Nous auront donc besoin d'un serveur web afin de tester les modifications apportés à notre projet web ainsi que des outils de collaboration pour pouvoir travailler avec mon binôme. Pour ce faire, nous utiliserons **GitHub***, une plateforme collaborative utilisée majoritairement en programmation, un éditeur de texte ainsi qu'un serveur web, pour ma part j'utiliserais **Atom*** : Un éditeur de texte personnalisable, ainsi que un serveur web **Apache***.

Pour ce projet-ci, un des points les plus important sera la collaboration nous utiliserons essentiellement GitHub, qui est une plateforme collaborative sur laquelle il est possible de lier un répertoire, en l'occurrence notre répertoire de travail, de notre machine physique avec le **cloud*** par exemple pour y envoyer les créations ou modifications de code. Les autres collaborateurs peuvent ensuite récupérer ces modifications facilement et les versions de code du projet seront sauvegardés périodiquement permettant une récupération de version plus ancienne en cas de problème.

II.3 Développement avec Symfony

Les applications que nous allons développer seront écrites à l'aide de plusieurs langages de programmation et outils, tout d'abord nous allons utiliser le **framework*** Symfony comme support de programmation, un framework permet de créer la structure d'une application, c'est littéralement un « cadre de travail ». Dans notre cas, Symfony servira à l'élaboration de la base de l'application et nous fournira des outils facilitant son développement, il permet également de faciliter la compréhension et la vue globale de l'application en utilisant le modèle de programmation **MVC***.

Symfony utilise donc un concept développement très répandu dans le monde de la programmation : le Modele Vue Contrôleur (MVC) c'est une architecture qui permet de maintenir facilement une compréhension de l'application, du code et de la structure. Ce modèle s'appuie sur 3 points :

- *Le modèle* : Le modèle correspond à la partie « données » que l'on choisit de traiter et ce qui en découle (enregistrement validation des données, ...), Dans notre cas ce sera la base de données (BDD) **mysql*** que l'on administrera avec **phpmyadmin*** .
- *La vue* : C'est l'interface graphique, ce que va voir l'utilisateur et l'outil avec lequel il pourra interagir, les éléments graphiques. On se servira de pages web, écrites en **TWIG*** , **CSS*** et **JavaScript***, le code TWIG générera du code HTML pour l'affichage des pages.
- *Le contrôleur* : Il va servir de contrôle pour traiter les données des utilisateurs, il modifie les données du modèle et de la vue. Pour notre programme cela correspondra à toute la partie Symfony pure, au code PHP.

La mise en page quand elle est gérée par une bibliothèque CSS et JavaScript nommée **Bootstrap*** : elle permet de gérer la mise en page d'une page HTML en donnant des identifiants aux éléments de la page, de ce fait elle pourra adapter dynamiquement son contenu. On parle alors de **responsive web design (RWD)***.

On sait donc maintenant que Symfony agit comme le « Chef d'orchestre » de notre application, son code PHP régit les autres éléments de la base de données et de l'interface graphique pour donner place à l'application tel que l'utilisateur la perçoit.

Pour les deux applications que nous programmerons, la structure sera la même, les fichiers PHP du contrôleur seront stockés sous le chemin : *nomprojet/src/controller/* tandis que la vue sera interprétée par le TWIG sous *nomprojet/template/*, le modèle sera représenté par la base de données avec laquelle on interagira par le biais de **l'ORM*** Doctrine : Une « interface » à la base de données.

II.4 Développement d'une application DHCP

Nous passons maintenant à la reprise du développement d'une application qui régira le service **DHCP*** de l'I2M a Chateau-Gombert, cette application a principalement été développée par les stagiaires de l'année précédente mais elle présente toutefois des **bogues*** et des erreurs plus ou moins critiques. La « correction » de cette application permettra d'apprendre d'une part les bases de connaissances pour les logiciels et langages de programmation que nous allons utiliser dans un futur proche et d'autre part permettre le déploiement de l'application sur l'ensemble des sites.

Cette application permet donc, lorsqu'un nouvel utilisateur arrive sur un site (Nouveau chercheur, stagiaire, ...) de lui attribuer une adresse IP sur demande par l'intermédiaire d'une interface web. Ce site web va donc servir d'interface pour les administrateurs afin qu'ils n'aient pas à modifier directement le fichier de configuration du serveur DHCP. On peut donc avoir un suivi des utilisateurs du réseau ainsi que de leur machines, avec les informations entrés lors de l'ajout. L'application permet par conséquent de contrôler qui est sur le réseau, c'est ainsi un avantage majeur d'un point de vue de la sécurité.

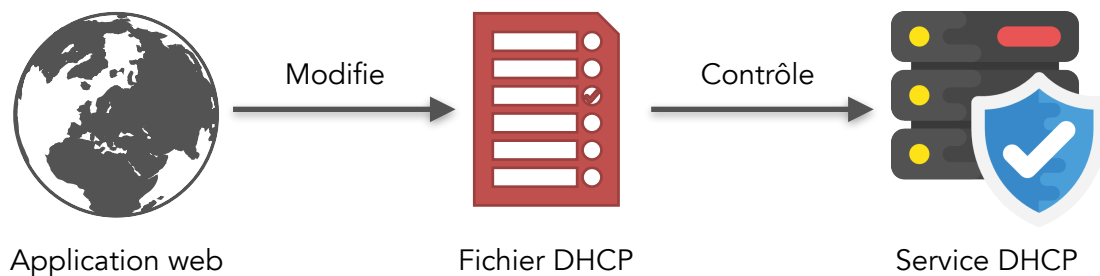


Figure 5 : Schéma résumant le fonctionnement de l'application DHCP

Nous allons ici nous intéresser à la partie application web et non administration du serveur et service DHCP, ces services étant déjà en place et fonctionnels. Vous retrouverez ci-dessous un aperçu de l'application en l'état actuel.

DHCP CMI Home Add Service actif

voici le nombre d'utilisateur : 1650 provenant du fichier : /etc/dhcp/dhcpd.conf

1	dhcp1	6A:BF:E5:F1:60:0F	01-01-0001	147.94.64.6
2	80Y13	14:B3:1F:0B:E9:92	01-01-0001	
3	81T61	0c:4d:e9:b2:a3:38	01-01-0001	
4	morganMoranceyPortable	9C:EB:E8:47:14:58	01-01-0001	147.94.64.62
5	ezproxy	48:4D:7E:DF:F6:AA	01-01-0001	147.94.64.3
6	khuyenLe	00:80:8E:8A:90:A8	01-01-0001	
7	copieurbibli	00:80:91:B5:59:51	01-01-0001	copieurbibli
8	copieurdea	00:80:91:B5:59:6B	01-01-0001	copieurdea
9	copieuradmin	00:80:91:B5:58:6A	01-01-0001	copieuradmin
10	duncanSussfeld	80:00:0B:C7:EE:0D	01-01-0001	
11	i2m_DEKUV	DC:4A:3E:F7:68:D7	01-01-0001	
12	i2m_Macbookair-Lydia	98:5a:eb:c9:11:69	01-01-0001	
13	i2m_samir	08:62:66:6C:6b:5D	01-01-0001	
14	i2m_damien	a4:4c:c8:4d:f4:79	01-01-0001	
15	i2m_25E21	38:c9:86:12:7a:a1	01-01-0001	147.94.65.92
16	i2m_kschneider	5c:f9:38:ac:c2:84	01-01-0001	
17	90K23	70:5A:0F:D5:12:5C	01-01-0001	
18	nellyPerso	BB:70:F4:9C:C4:D2	01-01-0001	

200 @ app_dhcp_main 7765 ms 2.0 MB 2 5 in 47.18 ms anon. 74 ms 3.4.11

Figure 6 : Aperçu de l'application DHCP a sa dernière version

II.4.A Déploiement du serveur web

Pour pouvoir visualiser et tester notre application web, nous devons mettre en place un serveur web de test afin de pouvoir observer nos modifications. Pour cela nous allons utiliser le **paquet*** apache sous Linux pour héberger nos applicatifs respectifs sur nos ordinateurs personnels.

Nous commençons donc à mettre chacun en place un serveur web sur nos postes de travail, pour cela on récupère le paquet apache depuis la bibliothèque **APT*** puis nous mettons à jour sa configuration pour l'adapter a la configuration déjà existante de l'application en **production***, étant donné que notre application comportera du PHP, des paquets annexes seront nécessaire pour l'interprétation du PHP.

```
apt update
apt upgrade
apt install apache2 php php-xml
```

Apache et Php sont dès lors installés, nous mettons alors à jour sa configuration en créant un nouveau fichier dans le répertoire suivant : `/etc/apache2/sites-available/` On y importera la configuration déjà existante du service déjà en état de fonctionnement. (Voir annexe pour configuration hôte virtuel) Le site est maintenant présent dans la liste des sites disponible et il est prêt a être activé, on l'activera donc avec la commande :

```
a2ensite
```

Un module est également nécessaire pour le bon fonctionnement de l'application, on active donc le module nécessaire permettant la réécriture d'URL :

```
a2enmod rewrite
```

Afin de sécuriser l'accès a l'application, on protège la consultation en mettant un `.htaccess` lié a une authentification par mot de passe. Pour cela on crée donc un fichier « `.htpasswd` » et on y insèrera les informations de connexion :

```
nomutilisateur:motdepasse
```

Notre service web est maintenant fonctionnel, il ne reste plus qu'a importer le site en copiant les fichiers de l'application dans le répertoire partagé public : celui qui sera visible et accessible par internet.

```
cp -r cheminapplication /var/www/dhcp/public/
```

L'option « `-r` » permettant de copier récursivement les fichiers et répertoires.

Le site web fonctionne a présent, on peut y accéder en entrant l'adresse ip de notre machine qui héberge le serveur ou en entrant <http://localhost> sur notre machine.

II.4.B Correction des erreurs

Le service est maintenant pleinement fonctionnel et en ligne, hébergé sur nos machines. L'objectif est donc maintenant de réparer la majorité des bugs tout en comprenant les rouages et les mécanismes, à commencer par l'espace "Description" lorsque l'on veut ajouter un nouvel hôte sur notre application.

Hostname: 18X51
MAC Address: AA:AA:AA:AA:AA
Date:
 Check to set fixed IP
IP Address: 192.168.0.1
 Expire ?
Expiration date:
Description: description...

Figure 7 : Aperçu de la fenêtre « Ajouter hôte » de l'application

Lors de la saisie d'une **adresse MAC*** quand on veut ajouter un hôte : lorsque le caractère retour à ligne est entré, il apparaît vide dans les **logs*** de notre application une fois validé, nous devons donc y remédier, pour cela nous avons donc décidé de remplacer les sauts à la ligne par des caractères pipe « | » automatiquement pour faciliter la lecture.

```
if (substr_count($this->_commentary, "\n") >=1 || substr_count($this->_commentary, "\r") >=1){  
    $this->_commentary = str_replace ("\r\n","|", $this->_commentary);};
```

Nous ajoutons également un champ « date d'expiration » pour l'attribution d'un **bail DHCP*** temporaire, jusqu'à maintenant l'ajout au DHCP était sans limite dans le temps. Afin d'ajouter un élément de calendrier lorsque l'on clique sur un des deux champs de date on utilise l'élément **date picker** avec le code suivant :

```
$("#expirationDate").datepicker({  
    language:"fr", });
```

L'élément de calendrier qui apparaît est en fait une fonction : **datepicker**, c'est une fonction issue d'une **bibliothèque*** javascript, la bibliothèque *jQuery*, elle recense un grand nombre de fonctions javascript que l'on peut utiliser en effectuant un appel de fonction très simple.

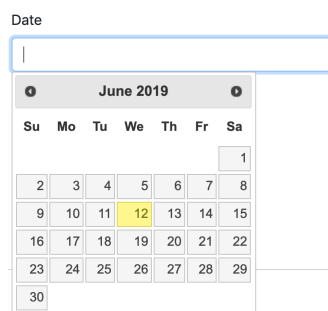


Figure 8 : Aperçu de l'utilisation de la fonction « datepicker »

Nous avons également ajouté un contrôle de champ pour vérifier que l'adresse MAC soit écrite dans le bon format, c'est à dire Hexadécimal :

```
<input type="text" class="form-control" id="macAddress" name="macAddress"
pattern="[0-9A-Fa-f]{2}[:-]){5}([0-9A-Fa-f]{2}"
placeholder="AA:AA:AA:AA:AA" value="{ host.macAddress }">
```

Enfin, nous avons mis en place l'option « expiration » pour le service DHCP, l'état du champ varie en fonction de l'état de la case à cocher :

Si la case n'est pas cochée, alors le champ d'expiration est désactivé

```
if(!$("#expirationDate").is(":checked")) {
    $('#expirationDate').prop('disabled', true);}
```

<input type="checkbox"/> Expire ?	Expiration date
	<input type="text"/>

Si la case est cochée, alors le champ d'expiration est activé

```
if ($("#expirationDateCheck").is(':checked')) {
    $('#expirationDate').prop('disabled', false);}
```

<input checked="" type="checkbox"/> Expire ?	Expiration date
	<input type="text"/>

Les baux DHCP peuvent donc maintenant avoir une durée finie. Pour les invités par exemple, qui ne restent qu'une durée déterminée.

Il est également nécessaire de mettre en place une barre de recherche, qui permettra de chercher par utilisateur, nom, machine ou adresse MAC. Pour cela on va donc premièrement interagir avec la partie contrôleur de notre application (*DhcpController*) :

```
public function listHost(Request $request){
    if ($request->isXmlHttpRequest()){
        $dhcp = Dhcp::readDhcpFile($this->container-
>getParameter('app.dhcpFile'));
        $id = $request->get("host");
        $a = $dhcp->getHosts();
        $hint = "";
```

A l'aide de ce code, nous pouvons récupérer la liste des hôtes contenant le caractère « a » par exemple et les afficher un par un à l'aide de la boucle dans la variable “*hint*”, avec un peu de html, on obtient un premier résultat convainquant.

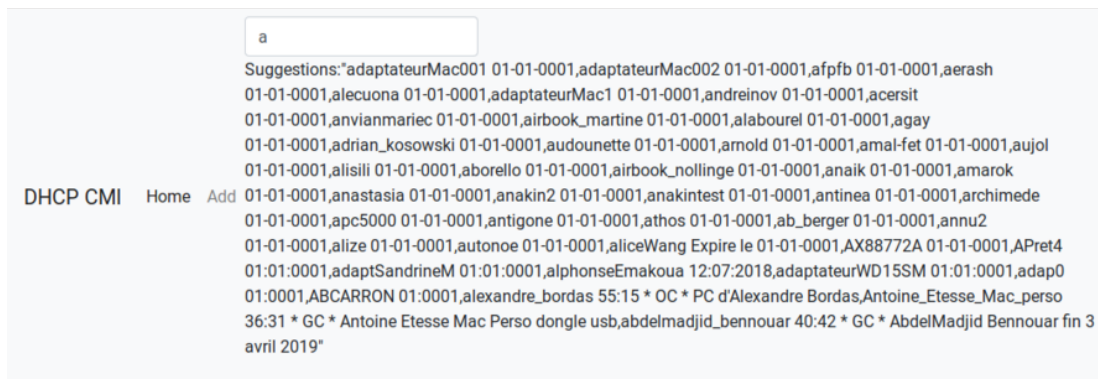


Figure 9 : Résultat d'une recherche par « a »

Il nous reste maintenant à pouvoir traiter les hôtes afin de les afficher sous la forme d'un tableau bien structuré, comme celui présent sur la page d'accueil principale. Pour cela, on va passer du côté de la vue de l'application : *HTML/Twig/Javascript/Ajax/Json*.

On formule donc un code *HTML/Javascript* pour faire fonctionner notre requête dynamiquement, pour cela on va utiliser un type de requête particulière : *AJAX*

Ce type de requête permet de régénérer dynamiquement une page *HTML* en fonction d'une action. Dans notre application ce sera en fonction de notre recherche, lorsque l'on enregistre une frappe au clavier, cela déclenchera notre fonction *AJAX* qui prendra la forme suivante :

```
function searchHost()
{
    $.ajax({
        url: "/listHost",
        method: "post",
        data: {host: $("#searchHost").val()},
        dataType: 'json',
        success: function (data) {
            $("#tablehost").find("tr").remove();
            $js = JSON.parse(data);
            $("#nbHost").text($js["dataLenght"]);
            var numLine = 0;
            $.each($js["data"], function(index, element) {
                var classCss = "";
                numLine = numLine + 1;
                if (element.fixedIpAddress != "1")
                {
                    classCss = 'class="table-warning"';
                }
            });
            $('#tablehost').append('<tr '+classCss+'> <td id="ligne">' + numLine + '</td>');
        }
    });
}
```

Figure 10 : Fonction AJAX générant dynamiquement les résultats d'une recherche

Notre tableau sera donc évolutif, il modifiera automatiquement son contenu en fonction de la recherche effectuée par l'utilisateur, les informations relatives aux hôtes sont **parsés*** dans un document au format JSON puis affichés en HTML.

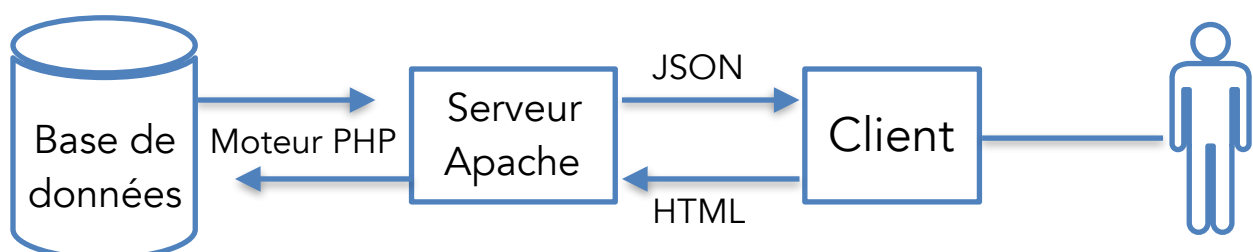


Figure 11 : Echanges effectués entre les différents éléments lors d'une recherche

Maintenant la recherche fonctionnelle, nous allons améliorer le listage des hôtes, la page d'accueil.

Le listage des hôtes fonctionne avec une boucle qui affiche les hôtes que le programme trouve dans le fichier des hôtes du serveur DHCP parcouru, il applique ensuite un traitement de style *HTML* et *CSS* pour y prendre la forme de tableau.

Cependant, plusieurs points sont à améliorer, les entrées du tableau doivent être numérotées par ligne et ces mêmes lignes doivent rediriger vers la page de l'hôte concerné lorsque l'on clique dessus pour une éventuelle modification des informations. Vous trouverez ci-dessous le code de la boucle générant la liste des hôtes.

```
{% for host in hosts %}
    {% set class = '' %}
    {% if (host.fixedIpAddress != "") %}
        {% set class='class="table-warning"' %}
    {% endif %}
    <tr {{ class|raw }}><th scope="row">{{ loop.index0 }}</th>
    <td><a href="/modifyHost?id={{ host.hostname }}">{{ host.hostname }}</a></td>
    <td><a href="/modifyHost?id={{ host.hostname }}">{{ host.macAddress }}</a></td>
    <td><a href="/modifyHost?id={{ host.hostname }}">{{ host.dateModif|date }}</a></td>
    <td><a href="/modifyHost?id={{ host.hostname }}">{{ host.expirationDate }}</a></td>
    <td>{{ host.fixedIpAddress }}</td>
</tr>
{% endfor %}
```

Figure 13 : Code TWIG de la boucle permettant de générer la liste des hôtes

Pour indexer les lignes de notre tableau, nous avons donc ajouté la ligne suivante :

```
<tr {{ class|raw }}><th scope="row">{{ loop.index0 }}</th>
```

La **balise*** `<tr>` permettant donc de créer une nouvelle colonne qui contiendra le code *TWIG* « `{{ loop.index0 }}` » permettant d'afficher l'itération de la boucle à la quelle on se situe, donnant donc lieu à l'affichage de notre tableau avec comptage des lignes.

L'application est maintenant corrigée en grande partie et améliorée en nombreux points, nous épargnons ici toute modifications mineures ou corrections graphiques à petite échelle. Même si l'application n'est pas encore parfaite, un autre demande viens interrompre notre progression.

II.5 Développement de l'application Keyring

Suite a une demande du pole administratif, il nous est demandé d'établir une application web pour pouvoir avoir un suivi des badges, clés et cartes d'accès des bâtiments et laboratoires de recherche. Pour cela nous avons donc décidé d'utiliser comme base pour notre application le projet DHCP déjà existant. La partie visuelle, base de donnée ainsi que sa façon de fonctionner sera remise a neuf pour l'adapter aux besoins du pole administratif.

Cette application sera donc elle aussi écrite a l'aide du framework symfony, en utilisant les même outils et méthodes de collaboration.

II.5.A Conception

Les premiers jours de conception de l'application on été consacrés a l'établissement du cahier des charges (cf annexes), de schémas de conception de l'application ainsi que de la mise en place de la base de donnée pour notre application.

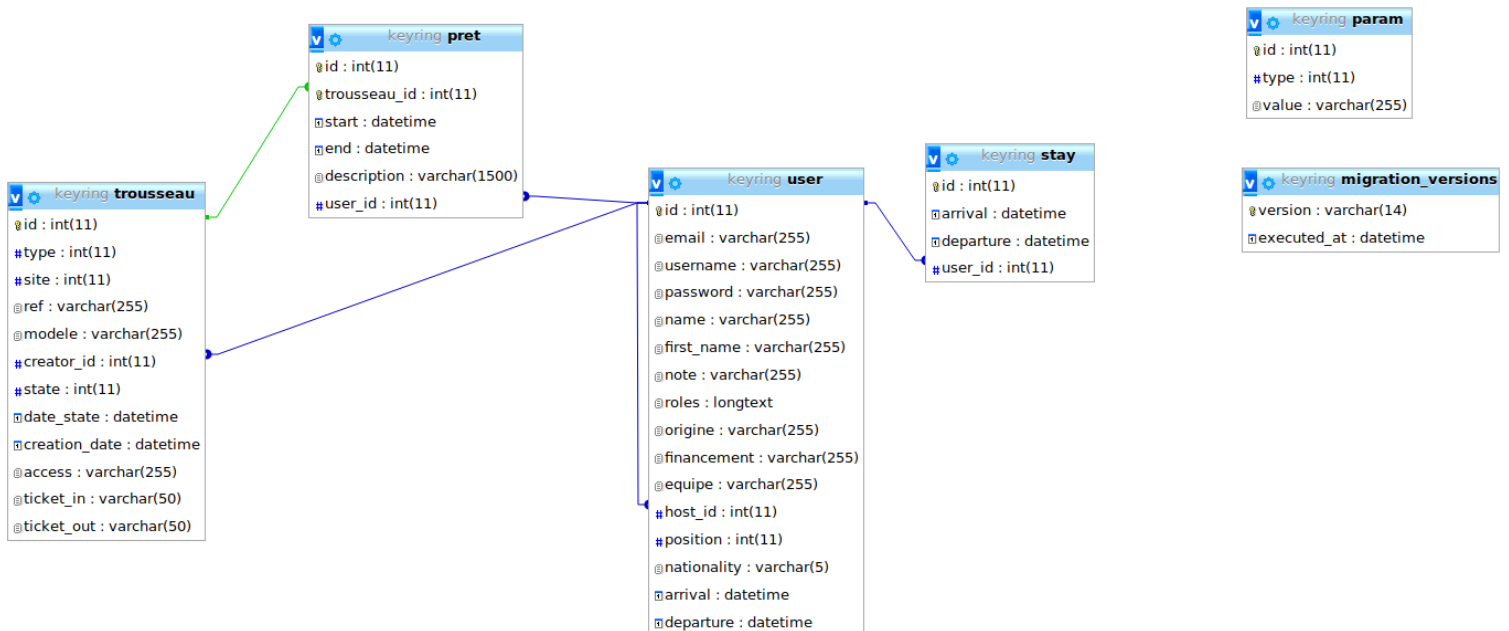


Figure 14 : Schéma de conception de la base de donnée de l'application Keyring

Les jours suivants auront permis le choix d'un style graphique et la conception d'un logo pour la même application. On aura choisi un thème sombre, un logo minimaliste représentant l'anneau d'un porte-clés ainsi que le nom Keyring.



Figure 15



Figure 16

II.5.B Développement de l'application

La création de la Base de données aura donc été effectuée en suivant le plan créé précédemment à l'aide d'un des outils intégré au framework Symfony : l'ORM Doctrine, une ORM est un outil qui permet de faciliter la création d'une base de donnée, c'est une couche d'abstraction pour le langage SQL (cf exemple), dans notre cas l'ORM Doctrine est prévue pour fonctionner avec notre framework Symfony. Il suffit de lui soumettre les tables et objets que l'on souhaite créer puis de lui spécifier les caractéristiques qui la définissent :

En se plaçant à la racine du répertoire de notre application, on peut créer facilement une base de données et des tables de la façon suivante :

```
php bin/console doctrine:database:create
```

Puis

```
[yugohug0:dhcp imac$ php bin/console make:entity
Class name of the entity to create or update (e.g. GentleKangaroo):
[ > Voiture
created: src/Entity/Voiture.php
created: src/Repository/VoitureRepository.php

Entity generated! Now let's add some fields!
You can always add more fields later manually or by re-running this command.

New property name (press <return> to stop adding fields):
> Poids
Field type (enter ? to see all types) [string]:
[ > integer
Can this field be null in the database (nullable) (yes/no) [no]:
> no
updated: src/Entity/Voiture.php

Add another property? Enter the property name (or press <return> to stop adding fields):
>

Success!
```

Figure 17 : Création d'une entité dans la base de donnée

Une fois l'entité créée, la modification est enregistré dans doctrine, elle n'est pas encore appliquée sur notre base de donnée réelle pour cela on utilise deux autres commandes :

```
php bin/console make:migration
php bin/console doctrine:migrations:migrate
```

Doctrine va agir comme un intermédiaire entre nos modifications et l'écriture en Base de donnée, cette commande va permettre de faire « migrer » nos modifications dans la BDD.

L'aspect visuel lui est contrôlé par le moteur de templating TWIG, il permettra de générer du code HTML sans avoir à réécrire tout le même code pour avoir le même rendu récursivement sur toutes les

pages, pour un pied de page par exemple. Il permettra également d'insérer des variables dans le HTML pour faire varier le contenu dynamiquement

En reprenant le projet DHCP et en l'adaptant a nos besoins, on obtient une première version :

VIVS Cle et Badge				Accueil		Ajouter un trousseau		Ajouter un Pret		Recherche	
0	carte	Chateau-Gombert	carte	Carte magnétique	Supprimer						
1	dfg	Chateau-Gombert	dfg	Carte Hertzienne	Supprimer						
2	modele2	Luminy	tata	Carte magnétique	Supprimer						
3	test	Chateau-Gombert	test3	clef	Supprimer						
4	toto	Luminy	tatat	Carte Hertzienne	Supprimer						
5	toto	Chateau-Gombert	tatat	Carte Hertzienne	Supprimer						

Figure 18 : Première version de notre application

Le thème a ensuite pu être mis en place (Thème sombre) et les fonctionnalités ajoutées, notre application fonctionne de façon simplifiée.

L'administrateur se connecte a son compte sur l'application, pour accéder a l'interface d'administration ensuite il est possible d'enregistrer les prêts de clés ou badges. Pour cela 2 éléments sont nécessaires, les clés : l'objet prêté regroupant cartes et badges et les utilisateurs, on crée donc notre clé en rentrant ses informations, de même pour les utilisateurs et notre prêt est enfin possible. Le principal atout de cet méthode est la persistance des données, au vu des invités pouvant être souvent récurrent, l'utilisateur existe déjà et n'a pas besoin d'être recrée, il suffit alors de le rechercher et ses informations suivront.

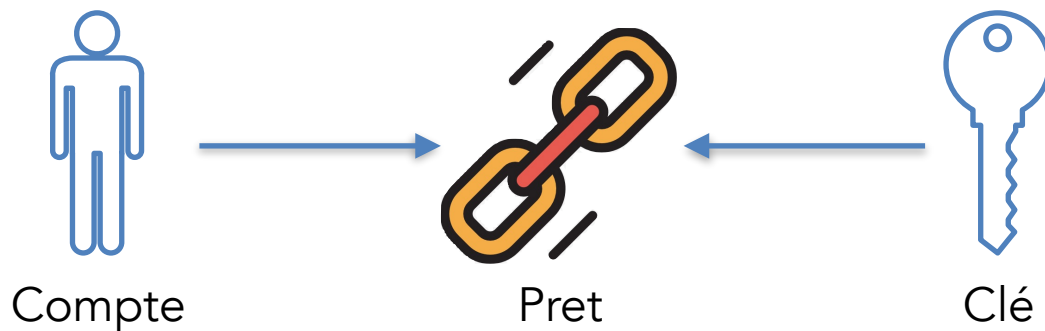


Figure 19 : Schéma explicatif de la création d'un prêt

Un des points critiques de notre application a également été la sécurité, il a fallu créer une page d'authentification, chiffrer les données de la BDD et assurer les droits des utilisateurs non connectés conduisant a une application finale de cette forme :

KeyRing							Clefs		Pret		Utilisateur		Paramètres		Compte: hugo.blachere@etu.univ-amu.fr		Deconnexion		Recherche	
#	Type	Modele	Site	Référence	Accès	Etat	Modifier		Supprimer											
0	Carte Hertzienne		Luminy	272	Porte TPR2	Actif	Modifier		Supprimer											
1	Carte Hertzienne		Luminy	669	TPR2	Actif	Modifier		Supprimer											
2	Carte Hertzienne		Luminy	674	314	Actif	Modifier		Supprimer											
3	Clef	Bricard	Luminy	13214.B49364	Bureau 121	Actif	Modifier		Supprimer											

Figure 20 : Capture de l'interface principale de l'application Keyring terminée

II.6 Le VPN avec OPEN-VPN

II.6.A Serveur VPN

Il nous est maintenant demandé de mettre en place un service VPN pour permettre au personnel d'accéder aux services internes de l'I2M depuis l'extérieur du réseau local. Pour cela nous allons utiliser le service OpenVPN, un service libre permettant de créer une liaison dite « tunnel » chiffrée.

Suite a une incompréhension de notre part, nous avons installé une mauvaise version de OpenVPN qui devient payante a partir d'un certain nombre d'utilisateurs. Nous avons donc du réinstaller la bonne version, pour cela Maximilien et moi-même avons trouvé un script permettant de faciliter la mise en place du service disponible sur GitHub au lien suivant :

<https://raw.githubusercontent.com/Angristan/openvpn-install/master/openvpn-install.sh>

Nous avons donc mis en place le service de la façon suivante, premièrement nous avons donc téléchargé le script depuis le dépôt GIT :

```
curl -O https://raw.githubusercontent.com/Angristan/openvpn-  
install/master/openvpn-install.sh
```

Nous lui avons ensuite donné les droits d'exécution :

```
chmod +x openvpn-install.sh
```

Puis nous l'avons exécuté en mode super utilisateur :

```
sudo ./openvpn-install.sh
```

Finalement, on applique les paramètres désirés au fil de l'installation, puis on personnalise nos comptes pour la connexion, tout le processus dans le script est très explicite. On veut également vérifier le nombre de tentatives de connexion par soucis de sécurité, pour éviter certain types d'attaques, pour cela nous allons utiliser un service nommé fail2ban

Fail2ban est un service permettant de limiter le nombre de tentatives de connexion en se basant sur les fichiers de logs d'un ou plusieurs services. Il fonctionne de la façon suivante : l'application va être exécutée en arrière plan et vérifier les fichiers de logs, lorsque une ligne dans les logs correspond a une expression que l'on lui aura spécifié sera présente, il bloquera les requête correspondant a l'utilisateur responsable.

Malheureusement il est impossible pour nous d'utiliser ce service, l'authentification des utilisateurs se faisant du coté du client, c'est a dire sur l'ordinateur de l'utilisateur, il n'y a donc pas de logs permettant de voir qu'un utilisateur se trompe de mot de passe. Suite a l'utilisation impossible de fail2ban, nous cherchons un autre moyen de contrôler l'authentification des utilisateur et de prévenir les attaques par tentatives. Nous cherchons donc un script permettant de faire un deuxième contrôle de mot de passe coté serveur. Pour cela nous utiliserons le projet de *alonbl* sur GitHub :

<https://github.com/OpenVPN/openvpn/blob/master/sample/sample-scripts/auth-pam.pl>

Il faut maintenant assurer la connexion des clients aux service de VPN pour cela on utilisera le même script que pour la mise en place du serveur, en l'exécutant une seconde fois, il permettra de créer les comptes clients avec leur nom d'utilisateur ainsi qu'un mot de passe pouvant être optionnel au besoin.

II.6.B Clients VPN

Il est nécessaire pour les utilisateurs d'installer le client OpenVPN ou un autre client VPN pour pouvoir établir la liaison tunnel entre l'utilisateur et le serveur.

Pour les utilisateurs sous macOS et Windows, on pourra utiliser le client OPNVPN ou Tunnelblick : Un client VPN libre et open-source.

Tunnelblick se télécharge de manière tout a fait standard, on l'installe puis on clique droit sur l'icône dans la barre de tache, on lui associe un fichier utilisateur crée précédemment a l'aide du script d'installation. Ce même script peut être relancé même une fois l'installation d'openVPN terminée pour créer d'autre fichiers client protégés par mot de passe ou non. Une fois le fichier utilisateur associé, il suffit de cliquer sur « Se connecter » et la connexion apparait valide.

Sous Linux, pour se connecter au service VPN a l'aide d'un mot de passe (facultatif) et d'un fichier avec l'extension .ovpn (obligatoire) il faut taper la commande suivante :

```
sudo openvpn --config nomdetonfichier.ovpn --auth-user-pass
```

II.7 Mise a jour serveurs I2M

Notre dernière tache consistait a mettre a jour le système d'exploitation sur lequel était hébergé tout les services sur le site de Chateau-Gombert : Proxmox

Proxmox est un OS hyperviseur pour machines virtuelles (VM) et conteneurs LXC sur lequel sont hébergés les services de l'I2M. La version étant extrêmement ancienne, une mise a jour est nécessaire car certains outils ne sont plus supportés et par soucis de sécurité une fois de plus.

Pour ce faire nous avons donc du utiliser un serveur assez puissant pour y transférer les services en déjà en route le temps d'effectuer la mise a jour sur les serveurs principaux.

Nous avons donc créé une clé USB dite « Bootable » sous Proxmox sur laquelle on peut faire démarrer notre serveur de sauvegarde afin d'y transférer les services

Pour cela on télécharge le système d'exploitation Proxmox ainsi qu'un utilitaire de « flash » qui va permettre de rendre notre clé USB **bootable***, on utilisera *Balena Etcher* : Un utilitaire de flash open-source.

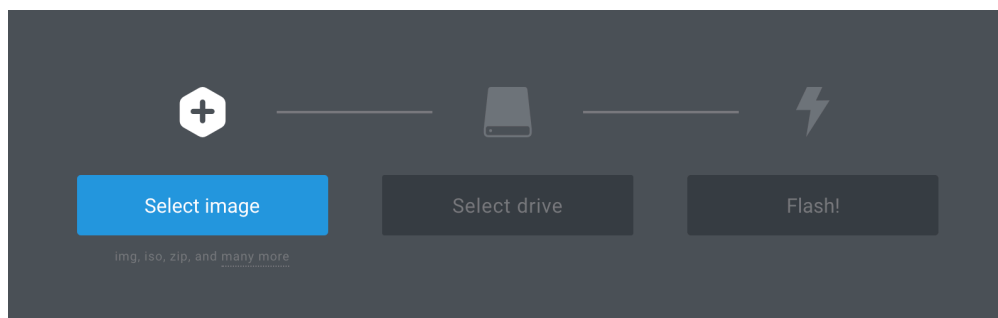


Figure 18 : interface de Balena Etcher

On choisit donc l'image de notre OS téléchargé au format ISO, notre support de démarrage : une clé USB, puis on lance le flash.

Une fois le processus terminé, on connecte la clé au serveur temporaire puis on appuie sur la touche F12 de son clavier lors de son démarrage pour accéder à son BIOS*, depuis cette interface on choisit la clé comme support de démarrage et notre serveur démarre alors sous Proxmox, la machine est prête à accueillir les conteneurs et VM.

On fait donc une sauvegarde de nos services en cliquant sur le bouton backup à côté de nos machines virtuelle ou conteneurs. On se connecte par la suite en SSH à notre instance proxmox :

```
ssh utilisateur@ip
```

Puis on copie ces sauvegardes de notre instance en cours sur notre serveur temporaire :

```
scp /var/lib/vz/dump/nomsauvegarde utilisateur@ipserveurtemporaire
```

Sur le serveur temporaire on effectue une restauration de la sauvegarde avec la commande :

```
qm restore nomimage idvm
```

On peut maintenant démarrer les conteneurs et VM depuis l'interface web

Nous sommes maintenant soumis à un autre problème, la nouvelle version de proxmox ne permet pas de lancer un conteneur avec un point de montage de la machine physique, c'est à dire de « Partager » une partie de l'arborescence de la machine physique avec le conteneur. Nous avons donc dû, à l'aide d'un montage NFS, d'associer un répertoire de notre machine physique au conteneur.

Nous avons également proposé une solution alternative à leur système actuel de gestion de service, à l'heure actuelle les services de l'I2M sont hébergés sur des VM ou conteneurs LXC différents nous avons donc proposé de faire migrer l'ensemble de l'architecture sous Docker : Une solution de conteneurisation légère et adaptative. Pour cela, il nous a été demandé d'effectuer une étude comparative des deux systèmes, vous trouverez donc en annexe un lien vers notre présentation.

II.8 Maintenance générale

Il nous a également été confiés plusieurs missions visant à améliorer les conditions de travail des employés et à leur porter assistance lors de difficultés techniques.

II.8.A Installation d'un point d'accès sans fil

Notre première intervention consistait à l'installation d'un point d'accès sans fil afin d'étendre le réseau WI-FI Eduroam, un réseau qui vise un accès sans fil aux étudiants et personnels de l'enseignement lors de leur déplacements.

Pour cela nous avons donc dû récupérer une borne d'accès WIFI dans les locaux de la DOSI, puis nous avons pris des mesures pour estimer la taille de câble nécessaire à utiliser lors de l'installation et du passage du câble RJ45 dans le faux-plafond. Nous avons par la suite configuré le pare-feu pour autoriser la communication de l'équipement. Puis nous avons procédé à l'installation du point d'accès.

Nous avons également déplacé certaines imprimantes pour placer les imprimantes couleur au centre des étages du bâtiment ainsi que les imprimantes noir & blanc aux extrémités des bureau pour faciliter l'impression pour le personnel du bâtiment.



Figure 19 : Pose d'un point d'accès sans fil

II.9 Projets personnels

Parmi notre temps libre, Olivier a su nous assister lors de projets personnels, comme pour la mise en place d'un serveur de stockage à usage personnel avec redondance de disque à l'aide de la méthode R.A.I.D 1 ainsi que l'utilitaire MDADM, il a aussi su m'assister avec l'utilisation de docker pour la mise en place de services supplémentaires et de leçons théorique à titre personnel.

```
CONTAINER ID    IMAGE          NAMES          COMMAND          CREATED          STATUS
f0b614b8e5dc   plexinc/pms-docker:latest    NAMES          "/init"          20 hours ago    Up 20 hours (healthy)
                plex
d15493e823ac   bitnami/dokuwiki:latest     dokuwiki       "/app-entrypoint.sh ..."  5 days ago     Up 4 days
0:8005->80/tcp
146d19581a24   itzg/minecraft-server      minecraft     "/start"          3 weeks ago    Up 16 hours (healthy)
0.0:8001->25565/tcp
7230b7b3cbb7   odarriba/timemachine       timemachine   "/entrypoint.sh"  3 weeks ago    Up 4 days
8/tcp, 0.0.0.0:636->636/tcp
ce98d90f3b3d   portainer/portainer        portainer     "/portainer"      3 weeks ago    Up 4 days
000/tcp
006786af6bc5   linuxserver/heimdall       heimdall     "/init"           3 weeks ago    Up 4 days
0/tcp, 0.0.0.0:8004->443/tcp
d0wn@d0wn:~$
```

Figure 20 : Liste des services mis en place un serveur personnel

III. Conclusion

À travers mon expérience de stage à l'I2M Marseille, j'ai pu découvrir le fonctionnement interne d'une entreprise, notamment les enjeux de sécurité informatique.

Connu pour son travail fourni dans le domaine de la recherche, j'ai eu la chance d'observer et de participer à la naissance de nouveaux projets informatiques au sein de l'I2M. Ainsi, mes missions ont été très variées et m'ont permis de développer mes connaissances dans le domaine de la programmation, de l'administration système et du réseau. J'ai eu l'occasion d'effectuer des missions valorisantes et très diverses qui sont une richesse importante, elles m'ont apporté des connaissances que je pourrais utiliser dans ma future carrière professionnelle et futures expériences. De même, j'ai rencontré des gens extrêmement intéressants qui m'ont donné envie de poursuivre dans mes études en licence professionnelle et dont je garderai un très bon souvenir.

Même si certaines étapes ne se sont pas toujours déroulées comme prévu, elles ont su nous servir de leçons et ont améliorés notre façon de procéder et notre méthodologie. C'est une expérience forte qui a su renforcer ma conviction de poursuivre dans l'administration système.

Ce stage a su m'apporter des connaissances ainsi que des pratiques du monde d'entreprise mais il surtout su m'apporter une méthodologie qui est, je pense l'un des plus importants atouts dans les métiers de l'informatique.

IV. Glossaire

DUT, Diplôme Universitaire de Technologie

I2M, Institut de Mathématiques de Marseille

UBUNTU, Système d'exploitation, une des nombreuses et populaire distributions de Linux.

Dokuwiki, Moteur de wiki libre. Permet dans notre cas d'avoir un historique des travaux effectués.

OSX, Système d'exploitation conçu par Apple.

DOSI, Direction Opérationnelle des Systèmes d'Information

LAN, Local Area Network (Réseau de Zone Locale)

Commutateur, Switch, Equipement reliant plusieurs segments dans un réseau.

Routeur, Equipement réseau permettant de relier plusieurs réseaux entre eux.

VPN, Virtual Private Network, Réseau privé virtuel, Outil permettant d'accéder a un réseau local distant.

VLAN, Réseau local virtuel permettant d'améliorer la gestion du réseau.

HP, Hewlett-Packard, Compagnie d'électronique et d'informatique

Cisco, Entreprise d'équipements réseau

SSH, Secure Shell Access, protocole de communication sécurisé

GitHub, Logiciel de travail collaboratif et de versioning.

Atom, Editeur de texte collaboratif et personnalisable

Apache, Serveur web HTTP le plus utilisé sur le web

Cloud, Exploitation a distance d'un serveur informatique a travers internet.

Framework, Ensemble d'outils et de composants utilisé pour créer la structure d'une application et faciliter son développement.

MVC, Modèle Vue Contrôleur, motif d'architecture logicielle définissant son architecture.

MySQL, Système de gestion de base de donnée le plus utilisé au monde

PhpMyAdmin, Application web de gestion de base de donnée

TWIG, Moteur de template principalement utilisé en PHP

PHP, HyperText PreProcessor, langage de programmation pour établir des sites web dynamiquement.

HTML, HyperText Markup Language, langage de balisage pour la conception de pages web

CSS, Cascading Style Sheets, Langage de mise en page généralement utilisé pour les sites web

Javascript, Langage de stripping également utilisé habituellement pour les sites web

Bootstrap, Collection d'outils web

RWD, Responsive Web Design, Gestion de site web adaptatif aux dimensions du support

ORM, Object-relationnal mapping

DHCP, Dynamic Host Configuration Protocol, Protocole réseau qui assure la configuration automatique réseau d'un hôte.

Bogues, Ensemble d'erreurs pouvant survenir lors de l'exécutions d'une application.

Paquet, Fichier a télécharger pouvant contenir une application, notamment sous Linux.

APT, Advanced Packaging Tool, Gestionnaire de packet pour les distributions Linux.

Mettre en production, Déployer a grande échelle.

Adresse MAC, Adresse matérielle unique d'un périphérique possédant un connectivité réseau.

Logs, Historique des événements.

Bail DHCP, Durée de possession d'une adresse IP alloué par un service DHCP a une machine

Bibliothèque, Ensemble de fonctions ou de codes a disposition.

Parser, Découper le code en éléments fonctionnels

Balise, Element servant a définir le début et fin d'un block en HTML.

Bootable, Signifie démarrable, une clé USB bootable est une clé sur laquelle l'on peut démarrer.

V Annexes

Configuration de l'hôte virtuel :

```
<virtualHost *:80>
  ServerName dhcpcmi.i2m.univ-amu.fr
  DocumentRoot /var/www/dhcp/public
  <Directory "/var/www/dhcp/public/">
    AuthType Basic
    AuthName "Restricted Content"
    AuthUserFile /etc/apache2/.htpasswd
    Require valid-user
    AllowOverride None
    Order Allow,Deny
    Allow from All
    <IfModule mod_rewrite.c>
      Options -MultiViews
      RewriteEngine On
      RewriteCond %{REQUEST_FILENAME} !-f
      RewriteRule ^(.*)$ index.php [QSA,L]
    </IfModule>
  </Directory>
  ErrorLog /var/log/apache2/dhcp_error.log
  CustomLog /var/log/apache2/dhcp_access.log combined
  <Directory /var/www/dhcp/public/bundles>
    <IfModule mod_rewrite.c>
      RewriteEngine Off
    </IfModule>
  </Directory>
  # optionally set the value of the environment variables used in
  the application
  #SetEnv APP_ENV prod
  #SetEnv APP_SECRET <app-secret-id>
  #SetEnv DATABASE_URL *****
</VirtualHost>
```

Cahier des charges de l'application Keyring :

Le pôle VIVS aurait besoin d'un logiciel ou d'une application (autre qu'Excel) permettant de gérer de façon conviviale clés et badges du labo.

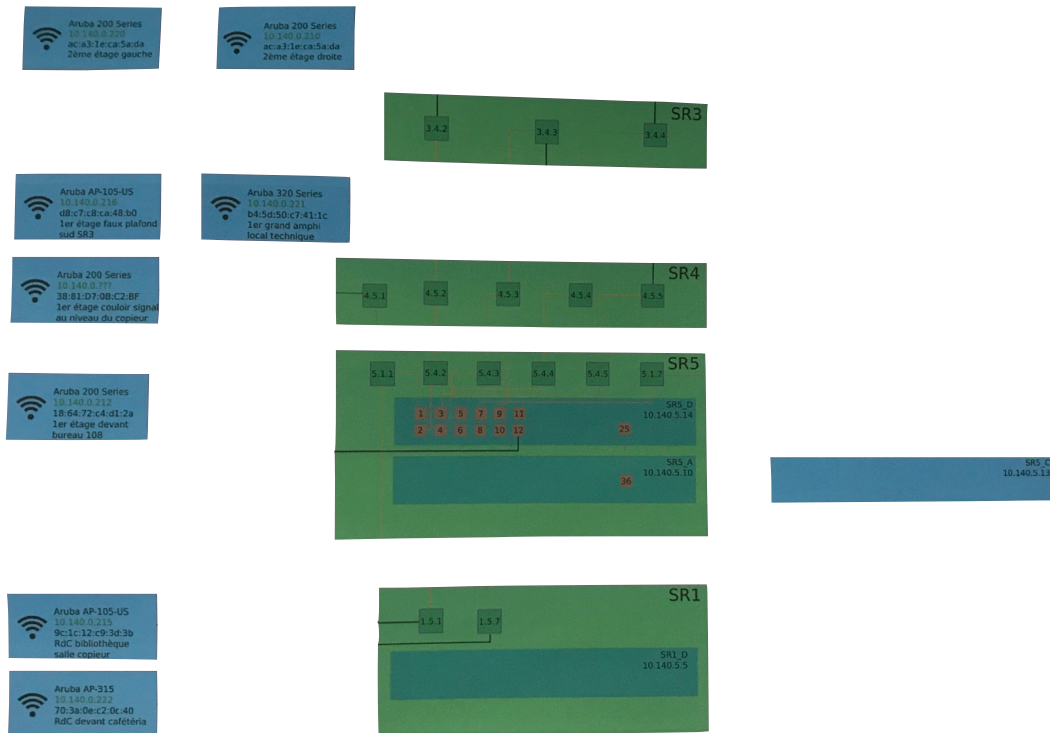
Il serait intéressant que ce logiciel se charge :

- automatiquement des relances de non restitution,
- des pertes, vol, badges défectueux,
- d'avoir les dates d'activation, de prolongation, de désactivation... avec référence des Tickets ENT (pour Château Gombert uniquement),
- d'avoir un formulaire en ligne des demandes qui pourrait figurer sur le site du labo et permettant par la même occasion d'informer l'ensemble des pôles (biblio, vivs, budget, informatique...) de l'arrivée de quelqu'un au labo.
- de pouvoir visualiser rapidement la disponibilité des clés par bureau (pour Château Gombert).

Diaporama comparatif Docker & Proxmox :

Disponible en suivant ce lien : <https://mega.nz/#!vIc3VC7L!Ux0ZsgT76M7ytdA0TTEcZdZgEM7coJOuPr4R4hfyGLc>

Plan réseau détaillé, Site de Chateau-Gombert :



Liste des VLANs sur les switches :

```
s2610pwr-CMI_6a# show vlans

Status and Counters - VLAN Information

Maximum VLANs to support : 32
Primary VLAN : DEFAULT_VLAN
Management VLAN :

VLAN ID Name | Status | Voice | Jumbo
-----+-----+-----+-----
1 | DEFAULT_VLAN | Port-based | No | No
21 | CMI-rech | Port-based | No | No
22 | CMI-ens | Port-based | No | No
24 | CMI-sunray | Port-based | No | No
26 | sunray-kiosk | Port-based | No | No
27 | jumpstart-re | Port-based | No | No
29 | vlansgd | Port-based | No | No
114 | PedagoPriv | Port-based | No | No
129 | TOIP-old | Port-based | No | No
200 | TOIP2 | Port-based | Yes | No
```

Code source du projet Keyring : <https://github.com/yugohug0/Keyring>

VI Bibliographie - Sitographie

Cahiers de l'admin GNU/Linux Debian (Octobre 2004) - Raphael Hertzog

Wiki-Ubuntu : <https://doc.ubuntu-fr.org/Accueil>

Git-Hub : <https://github.com/Angristan/OpenVPN-install>

Site web de Tania Rascia : <https://www.taniarascia.com/setting-up-a-basic-cron-job-in-linux/>

Docker Hub : <https://hub.docker.com/>

MaThèseEn180Secondes : <https://www.youtube.com/watch?v=UlnSv0q8Aao>

Linux.developez : https://linux.developez.com/formation_debian/nfs.html

OpenVPN : <https://openvpn.net/>

Fail2ban : https://www.fail2ban.org/wiki/index.php/Main_Page

Docker, Plex and encryption : <https://www.youtube.com/watch?v=geyXNXJ1S6A>

Tutoriels Symfony : <https://symfonycasts.com/>

Site web Symfony : <https://symfony.com/>

Site web Twig : <https://twig.symfony.com/>

Proxmox : <https://www.proxmox.com/en/>

Docker : <https://www.docker.com/>

Projet Keyring : https://github.com/olivierChabrol/keyring_git