



Département Informatique
Route Forestière de l'Hurtault
77300 Fontainebleau

Faurecia
ZI de Brière les Scellés
91152 ETAMPES

Ajout de Fonctionnalités pour l'outil CMDB

Stage de DUT réalisé par Vincent Girardot du 31 Mai 2014 au 6
Juin 2014

Maître de Stage : Stéphane Bernard
Enseignant Tuteur : Denis Monnerat
Année Universitaire : 2013 – 2014

Remerciements :

Je tiens à remercier les personnes suivantes pour le soutien qu'elles m'ont apporté dans le bon déroulement de ce stage.

Dans un premier temps, l'équipe pédagogique de l'IUT de Fontainebleau pour l'ensemble des connaissances fournies au cours de ces deux dernières années.

Merci également à l'ensemble du service informatique de Faurecia du site de Brières-les-Scellés pour leur accueil au sein de l'entreprise et pour leur bonne humeur quotidienne.

Et enfin, merci à Mr Stéphane Bernard pour son implication en tant que maitre de stage et pour la confiance et la liberté qu'il m'a accordé durant ces deux mois.

Table des matières

I.	Présentation de l'entreprise	5
A.	Historique	5
B.	Localisation du stage	5
II.	Présentation du Projet CMDB	6
A.	Data Center, Frames et Virtual Machines	6
1.	Les Data Center	6
2.	Les Frames	6
3.	Les VM.....	7
B.	Le rôle de la CMDB	7
C.	Fonctionnement de la CMDB	8
1.	Structure de la CMDB	8
2.	Collecte des données	8
III.	Déroulement du stage	9
A.	Etat initial de la CMDB.....	9
B.	Méthode de travail.....	10
C.	Objectifs de ce stage.....	11
IV.	Réalisation	11
A.	Outils et langages utilisés	11
1.	Langage	11
2.	Bootstrap	11
3.	Open Flash Chart 2.....	12
4.	LAMP.....	12
5.	Doxygen	12
B.	Optimisation et maintenabilité du code.....	12
1.	Rationalisation du code	12
2.	Création et documentation de fonctions	13
C.	Modification de l'interface Web	14
1.	Définition de l'Interface Homme-Machine	14
2.	L'ergonomie de la CMDB.....	14
3.	Refonte graphique	15
D.	Ajout de graphique avec Open Flash Chart 2	16
1.	Procédure de création d'un graphique.....	16
2.	Graphique dynamique	17
E.	Lpar2rrd.....	17
1.	Les graphiques.....	18
2.	Les fichiers	18
F.	La création de rapport	19
1.	Les limites	19
2.	Réalisation du formulaire	19
G.	Modification du système d'administration.....	20
H.	Déplacement d'une machine virtuelle.....	21
1.	Procédure.....	21
2.	Mise en place.....	22
V.	Conclusion.....	23
VI.	Lexique.....	24

Introduction

Dans le cadre de mes études à l'IUT, j'ai dû effectuer un stage de deux mois en entreprise, étape nécessaire à la validation de ma formation.

J'ai d'abord recherché des stages dans le domaine du développement Web, ayant pour objectif de m'orienter vers cette voie dans le futur.

J'ai été surpris par l'absence de réponses au vu du nombre de candidatures envoyées. Compte tenu de cet environnement de recherche difficile, j'ai élargi ma prospection sans me limiter aux offres des entreprises.

Grâce à des relations personnelles, une opportunité s'est présentée chez Faurecia. Cette démarche s'est accompagnée d'un entretien auprès des services Ressources Humaines et Informatique.

L'entreprise avait démarré la construction d'un intranet avec une stagiaire de l'UTBM, dont le stage s'arrêtait en Février. Malgré son sérieux et ses efforts, le site n'était pas tout à fait terminé. On m'a alors proposé de reprendre son travail.

J'ai donc débuté le stage le 31 Mars 2014, avec pour objectif d'ajouter de nouvelles fonctions à l'outil appelé CMDB.

I. Présentation de l'entreprise

A. Historique

Le groupe Faurecia est né de la fusion des groupes ECIA (filiale de PSA Peugeot Citroën) et d'Epeda-Bertrand Faure en 1997. C'est une entreprise française, numéro 3 mondial des sièges complet et leader européen des extérieurs d'automobile.

Elle a pour client la quasi-totalité des constructeurs automobile, dont ceux des économies émergentes indiennes, chinoises et coréennes, et est présente dans le monde entier avec 4 grandes divisions.

L'entreprise possède 274 sites de production implantés dans 34 pays, ainsi que 94 000 collaborateurs. En 2013, son chiffre d'affaire était de 18,03 milliards d'euros pour un résultat négatif de 88 millions d'euros

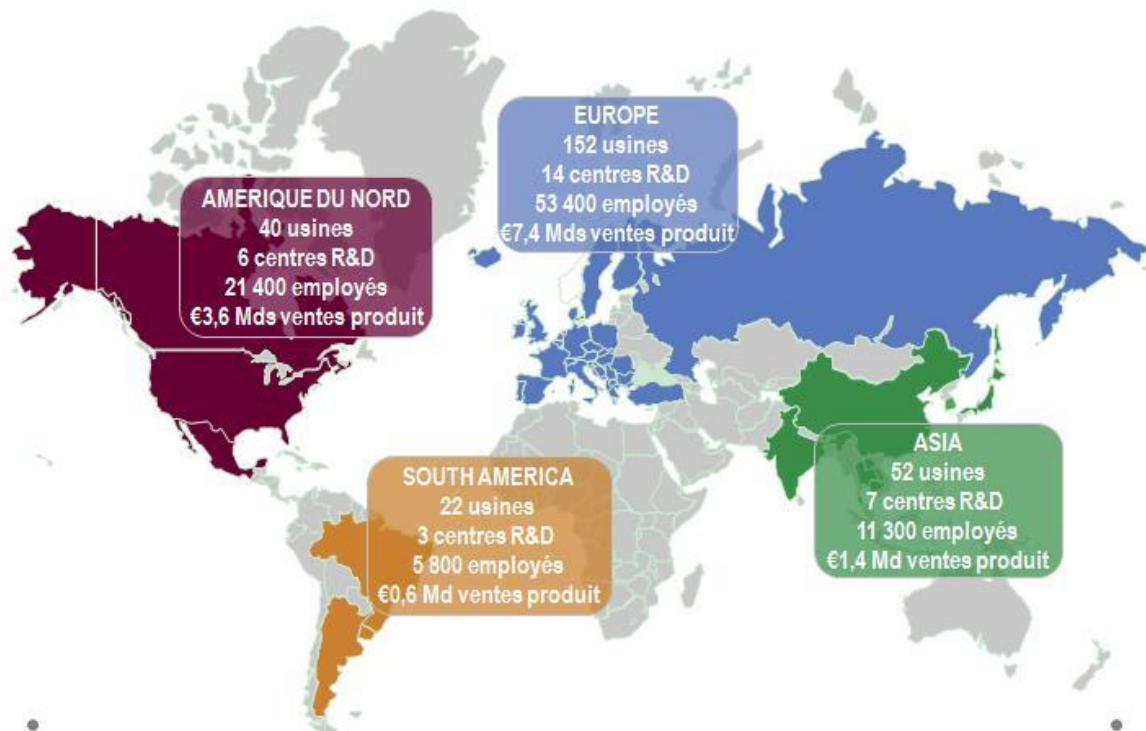


Figure 1 : Les 4 divisions de Faurecia

B. Localisation du stage

Le site de Brières-les-Scellés (Essonne) où s'est déroulé ce stage occupe une place stratégique dans le dispositif de R&D de Faurecia Siège d'automobiles. Il accueille les équipes d'innovation bien évidemment, mais également la division opérationnelle Europe du

Sud, ainsi que certaines fonctions supports du Groupe comme la comptabilité, la paie et l'informatique.

J'ai été affecté au service « GIS informatique groupe » qui regroupe plus de 1 000 employés dans le monde et plus précisément à la cellule Advanced Systems & Replication qui est composée de 3 centres de compétences :

- Bases de données : Oracle et DB2, Dominique CAPOEN (France)
- SAN et Back Up : Yves Bomba (France)
- UNIX-PowerHA et Virtualisation : Stéphane BERNARD (France) et Ron NEAWEDDE (USA)

Le but de ce service est de fournir au groupe les infrastructures demandées et d'en assurer le suivi ainsi que le support de niveau 3 aux équipes d'exploitation.

II. Présentation du Projet CMDB

A. Data Center, Frames et Virtual Machines

Le projet CMDB (Configuration Management Data Base, ou base de données de gestion des configurations) est une base de données qui se rafraîchit quotidiennement en collectant des informations sur les machines virtuelles (appelées Virtual Machines ou **VM**) et physiques (appelées serveur ou **Frames**) contenues dans un **Data Center**.

1. Les Data Center

Les Data Centers sont des sites physiques sur lesquels sont regroupés les équipements constituant le système d'information de l'entreprise (ordinateurs centraux, baies de stockage, équipements réseaux et de télécommunications, etc.). C'est un service généralement utilisé pour remplir une mission critique relative à l'informatique et à la télématique. Faurecia possède aujourd'hui 4 Data Center :

- 1 situé en Allemagne dans la ville d'Hagenbach
- 2 situés en France dans les villes de Marcoussis et de Seloncourt.
- 1 situé aux Etats-Unis, qui est encore en construction.

Chaque Data Center contient plusieurs frames.

2. Les Frames

Une Frame est un serveur, un ordinateur extrêmement puissant. Il contient en effet tous les composants d'un ordinateur personnel classique mais dans des quantités nettement plus élevées. A titre de comparaison, un PC contient généralement 4 cœurs de CPU, alors que les frames les plus puissantes utilisées par Faurecia en comportent 48.

Cependant, une frame n'est pas destinée à être utilisée comme un ordinateur. Elle doit permettre de virtualiser plusieurs machines (les VM). Concrètement la frame va partager ses ressources afin de créer plusieurs ordinateurs.

Pour cela, Faurecia utilise la technologie d'IBM : PowerVM. Il s'agit d'une suite de logiciel et d'aptitude matérielles permettant de mettre en œuvre les fonctions de virtualisation.

Il est possible de communiquer avec les frames grâce à une technologie IBM appelée **HMC** (Hardware Management Console). Pour ce faire, la HMC échange avec un processeur dédié présent dans chaque frame appelé **hyperviseur**. Grâce à lui, il est possible de gérer toutes les fonctions de partitionnement.

En utilisant cette console, un administrateur peut donc configurer et gérer les partitions logiques (qui contiennent les VM), mais également la **CoD** (Capacity on Demand).

Cette dernière option permet d'installer des composants supplémentaires sur une frame. En effet il est possible de n'utiliser qu'une partie des composants de ces machines, et, lorsque l'utilisation des composants disponible est à son maximum, la CoD permet de « déverrouiller » les composants inutilisés. Cela permet d'en réduire les coûts d'utilisation.

3. Les VM

Comme dit précédemment les VM sont des ordinateurs virtuels hébergés dans une frame. Ces machines sont regroupées en **Cloud**. Un Cloud est un ensemble de processus qui consiste à utiliser la puissance de calcul et/ou de stockage de plusieurs machines.

Chez Faurecia, ils peuvent contenir plusieurs centaines de VM et sont dédiés à une seule application. Il y a donc, par exemple, un Cloud pour SAP et un pour Oracle. Le processus de création d'un Cloud est appelé un **déploiement**.

B. Le rôle de la CMDB

Etant donné la criticité des informations stockées dans les Data Center, il est vital d'assurer un suivi permanent des VM et des Frames qu'ils contiennent. En effet si une application comme SAP était coupé pendant plus de 4 heures cela provoquerait un arrêt de chaine chez les constructeurs et entraînerait des pénalités importantes pour l'entreprise. Hors, plus le temps passe, plus le nombre de VM augmente, ce qui rend de plus en plus complexe la gestion des clouds.

En collectant quotidiennement ces données, la CMDB permet un suivi quotidien de chaque élément du parc informatique, mais en donne également une vision globale.

Toutefois son rôle ne s'arrête pas là : si un incident matériel se produit (prenons par exemple la perte d'une frame), il est possible d'utiliser les données de la CMDB pour reconfigurer et réinstaller toutes les machines virtuelles qui étaient hébergées à l'intérieur. C'est donc un outil de surveillance et de **DRP** (Disaster Recovery Planning).

C. Fonctionnement de la CMDB

1. Structure de la CMDB

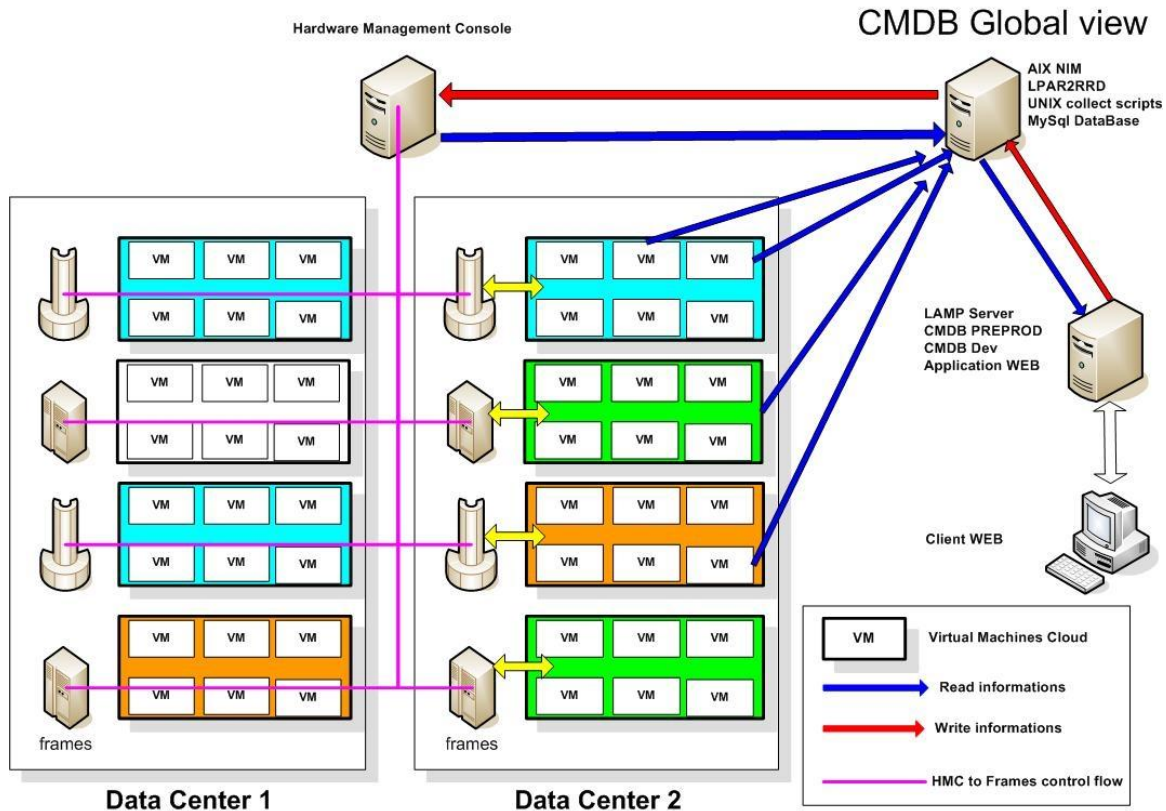


Figure 2 : Structure de la CMDB

2. Collecte des données

Chaque jour, à une heure prédéfinie, un script de lancement de collectes d'informations est exécuté. Ce script lance à son tour deux autres scripts.

- Le premier collecte les informations sur la partie UNIX des VM.
- Le second collecte des informations sur la partie « virtualisation » en interrogeant les HMC.

Les résultats sont stockés dans des fichiers textes formatés. Un deuxième programme est alors exécuté, dans le but d'alimenter les tables de la base de données avec les informations obtenues. Ce script fonctionne également en deux étapes :

- Dans un premier temps, il permet de transférer les données collectées la veille dans des tables dites d'**archivage**.
- Dans un second temps il injecte les données collectées dans la journée dans les tables dites **quotidiennes**.

Enfin la dernière fonction de la CMDB permet d'obtenir un « cliché » journalier de toutes les informations concernant la configuration des machines virtuelles. Cette fonction utilise des requêtes SQL sur les tables remplies précédemment, dont le résultat est stocké sous la forme de fichier qui sont utilisables par les programmes d'installation et de déploiement de Cloud. C'est ce dernier programme qui fait office de DRP.

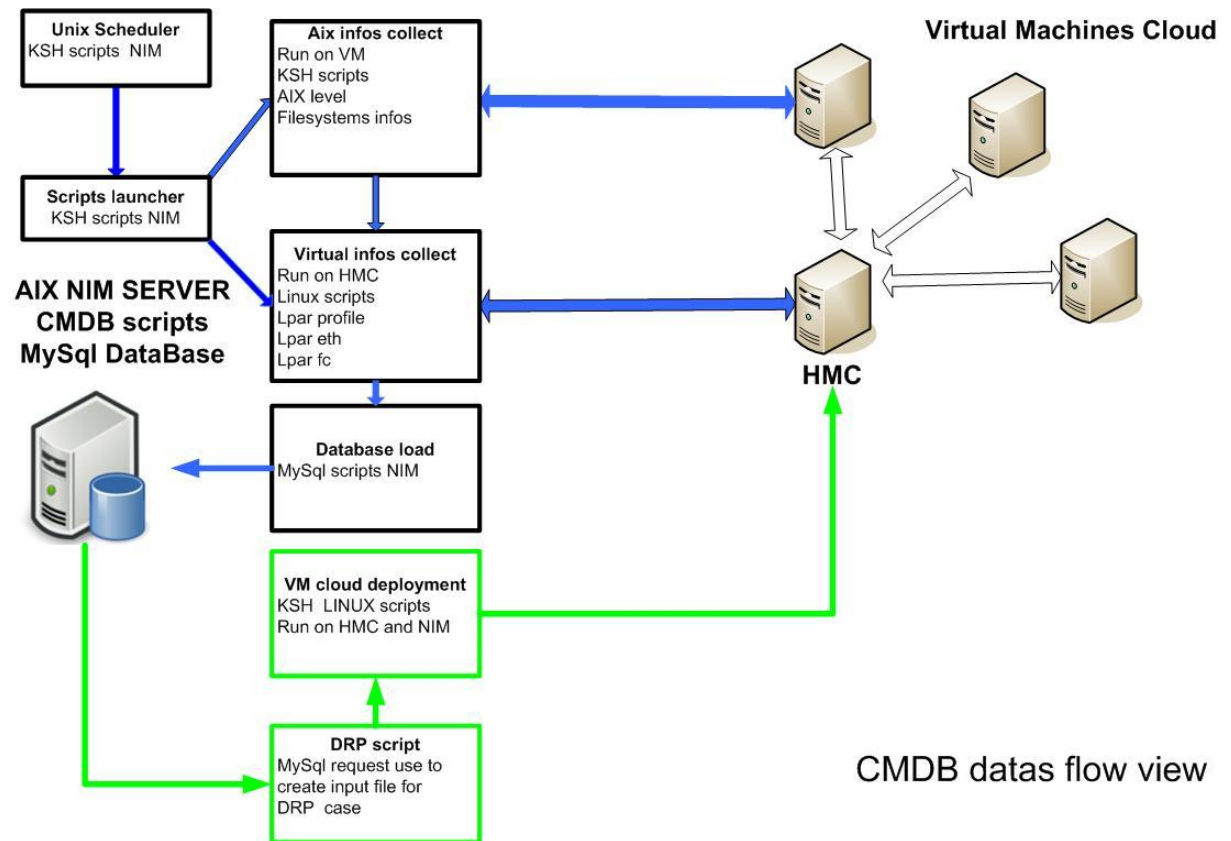


Figure 3 : Le processus de collecte de données

III. Déroulement du stage

A. Etat initial de la CMDB

L'outil CMDB est le fruit du travail de 6 personnes :

- **Nasser BAGTACHE** (Tunisie – Freelancer IBM sur ce projet) : Expert PowerVM/HA. C'est le père de ce projet car il a en a pensé et réalisé les première briques.
- **Stéphane BERNARD** (France) : Expert AIX et PowerVM/HA. Il a travaillé en tant que concepteur et coordinateur du projet.
- **Ron NEAWEDE** (USA) : Expert AIX et PowerVM/HA. Auteur des scripts de collecte et d'injection dans la base de données.
- **Dominique CAPOEN** (France) : Expert DBA Oracle/DB2. Il a apporté son expertise sur l'organisation et la structure des tables de la base de données.
- **Ruyi XU** (Chine) : Stagiaire de l'UTBM. Elle a réalisé l'application Web permettant d'afficher les données contenues dans la base de données.

- **Fabien LOURO** (France) : Alternant Faurecia. Il a apporté ses conseils sur la programmation Web.

A la date du 31 Mars 2014, la CMDB était en version **Alpha** : En effet le site était encore interne au service qui travaillait dessus, elle disposait de toutes les fonctions de bases, mais elle comportait encore quelques bugs, et son interface était très rudimentaire.

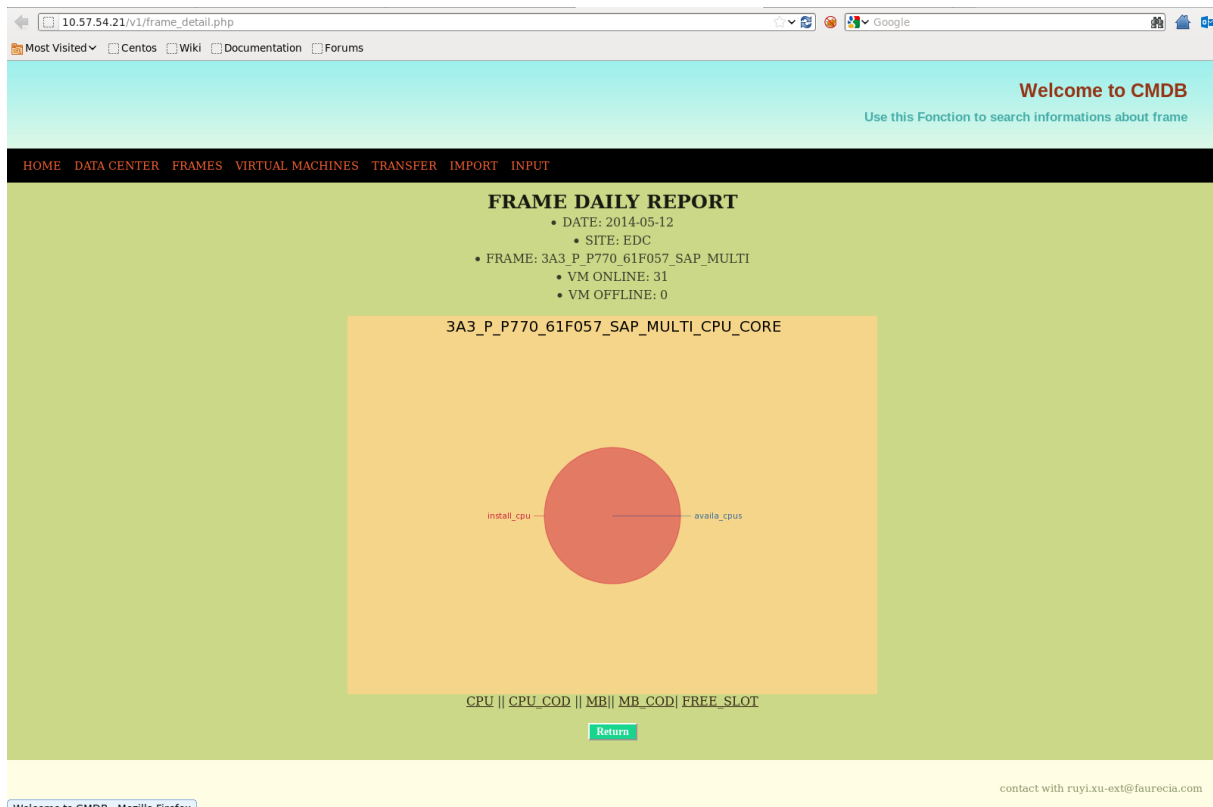


Figure 4 : La page de rapport de Frame au 31/03/2014

B. Méthode de travail

Il n'existe pas de liste précise des fonctions à ajouter. Deux fois par semaine, les membres de l'équipe se réunissent pour discuter des fonctions existantes, et des informations qu'il pourrait être utile d'afficher.

L'objectif est d'identifier les informations pertinentes afin de les intégrer dans la CMDB, mais également de supprimer les informations jugées inutiles. En effet les pages contiennent beaucoup d'informations et peuvent parfois mettre plus de 10 secondes avant d'être totalement chargées, c'est pourquoi il est nécessaire de se focaliser sur l'essentiel.

Cependant une information peut parfois être utile, mais peu lisible selon la manière dont elle est présentée, ce qui entraîne une modification de son affichage.

Enfin lorsqu'une proposition concernant une nouvelle fonction est faite, on discute dans un premier temps de sa pertinence. S'il s'avère qu'elle est utile, on regarde si il est possible de l'ajouter, et si l'ajout est possible techniquement, alors on débute le codage.

C. Objectifs de ce stage

Bien que ce ne soit pas à proprement parler une fonctionnalité, le premier objectif de ce stage est de fournir une documentation et un code propre, car il faut prendre en compte que cet intranet sera par la suite maintenu par des personnes n'ayant quasiment aucune connaissance en développement Web. Il est donc crucial de garder ces objectifs en tête durant toute la réalisation de ce projet.

Le deuxième objectif est de relier la CMDB à un outil de monitoring appelé lpar2rrd afin d'éviter aux équipes de naviguer entre deux sites pour obtenir toutes les informations sur les Data Center.

L'outil affiche des rapports mais il est impossible de les imprimer, il faut donc trouver un moyen de les exporter afin de pouvoir les manipuler plus facilement.

Enfin, il est nécessaire de procéder à une modification de la charte graphique, afin de rendre les informations plus visible, mais également pour donner un aspect plus professionnel au site.

IV. Réalisation

A. Outils et langages utilisés

1. Langage

L'intranet interagit avec une base de données, on utilise donc obligatoirement le HTML, le CSS, le PHP et le SQL. Cependant certaines pages du site sont relativement lourdes, et ces langages ne permettent pas de créer des pages Web dynamiques c'est pourquoi on utilisera également le Javascript.

Il est dans un premier temps utilisé pour contrôler le contenu des formulaires, mais également pour communiquer avec le serveur grâce à l'**AJAX**. Cela permet de modifier une partie du contenu de la page en fonction des actions de l'utilisateur sans avoir à en recharger l'intégralité.

2. Bootstrap

Bootstrap est un framework permettant d'obtenir rapidement un design épuré, professionnel.

Grâce à la documentation claire et riche en exemple, il est facile de construire rapidement une page web, et ce, sans avoir nécessairement de connaissances poussées en HTML et CSS ou Javascript.

De plus, il est disponible sous **licence MIT**, ce qui signifie qu'il est parfaitement légal de l'utiliser, le copier, le modifier, le fusionner, le publier, le distribuer, le vendre et de changer sa licence, et ce gratuitement, même pour un usage professionnel. C'est pour toutes ces raisons qu'il a été décidé de refondre le design à l'aide de cet outil.

3. Open Flash Chart 2

Open Flash Chart est une bibliothèque multi-langage qui permet de réaliser des graphiques simplement en manipulant des données au format JSON.

Le **JSON** (Java Script Object Notation) est un format de données textuelles dérivé de la notation des objets en Javascript. Il permet donc de représenter une information de manière structurée.

En combinant le JSON et l'AJAX il est possible de réaliser des graphiques dynamiques.

4. LAMP

LAMP (**L**inux **A**pache **M**ySQL **P**HP) est un ensemble de logiciels libres permettant de construire des serveurs de site web.

5. Doxygen

Ce logiciel est un générateur de documentation qui est capable de produire une documentation logiciel à partir du code source d'un programme et de ses commentaires s'ils sont écrits dans un format particulier.

B. Optimisation et maintenabilité du code

Le code laissé par la précédente stagiaire, bien que fonctionnel, est difficilement maintenable car toutes les fonctionnalités développées sont codées en « brut », autrement dit, il n'existe aucune fonction, et les sources ne comportent que peu de commentaires.

1. Rationalisation du code

Dans un premier temps, il faut analyser le code pour déterminer les parties pouvant être automatisées par la création de fonction ou l'utilisation de boucle. Durant cette étape, on ajoute également les commentaires. On sépare également le code CSS et une partie du Javascript des pages Web. Cela permet de rendre le code plus lisible.

Stocker la mise en page dans un fichier CSS présente un autre avantage :

Comme ce fichier est appelé dans chaque page du site, toute modification effectuée sur celui-ci est prise en compte sur l'intégralité du site. En écrivant le code CSS dans une balise style, il est nécessaire d'effectuer la même modification sur chaque page du site ce qui peut s'avérer fastidieux.

2. Création et documentation de fonctions

L'écriture de fonctions doit engendrer une simplification du code. Par conséquent, on n'en crée que si le code est déjà utilisé plusieurs fois, ou s'il est possible qu'il soit réutilisé plus tard.

Un exemple de fonction utile pour la CMDB est `printTable()`. Son rôle est de générer le contenu d'un tableau HTML à partir d'une requête SQL. Etant donné leur nombre important dans les pages de l'intranet, elle permet de simplifier considérablement le code.

Toutes les fonctions sont stockées dans un fichier `fonction.php`. En incluant ce fichier dans une page, on dispose de l'intégralité des fonctions codées depuis le début du projet.

Enfin, on utilise **Doxygen** pour documenter ce fichier. La syntaxe est similaire à celle du langage LaTeX qui permet de composer des documents.

```
279 /**  
280 * \fn create_pie_chart($title, $colorSet, $bg_color, $data)  
281 *  
282 * Fonction permettant de créer une pie_chart.\n  
283 * Exemple de pie-chart : Tous les rapports sur les CPU quotidiens.\n  
284 * Utilisation : create_pie_chart($title, $colorSet, $bg_color, $data)  
285 *  
286 * \param string $title : Titre de la chart  
287 * \param array $colorSet : Couleur hexadécimale. Voir color.php  
288 * \param string $bg_color : Couleur hexadécimale. Voir color.php  
289 * \param array $data : Le tableau est au format array("nom_de_valeur1" => valeur1, "nom_de_valeur2" => valeur2, ...);  
290 *  
291 * \return Retourne une chart (Element OFC). Il faudra convertir cette chart au format JSON pour l'afficher dans la page  
292 */  
293 function create_pie_chart($title, $colorSet, $bg_color, $data)
```

Figure 5 : Un commentaire respectant la syntaxe Doxygen

La documentation peut être générée dans plusieurs formats.

Ici on utilise uniquement la génération de code HTML, ce qui permet de relier le résultat à l'interface Web. Cette documentation a pour rôle de répertorier l'ensemble des fonctions disponibles et d'en expliquer le fonctionnement sans avoir à en analyser le code. Cela doit permettre une prise en main plus rapide du code pour les personnes qui le maintiendront.



Figure 6 : La documentation générée grâce au commentaire de la figure 5.

C. Modification de l'interface Web

La modification de l'interface Web comporte deux étapes. Il faut tout d'abord repenser l'organisation des informations selon les critères de l'**Interface Homme-Machine** (IHM), mais c'est également l'occasion de procéder à une refonte graphique afin de donner un aspect plus professionnel au site.

1. Définition de l'Interface Homme-Machine

L'IHM est l'outil qui permet à l'utilisateur de contrôler et de communiquer avec une machine. Le principal objectif lors de la création d'une interface est la recherche de l'**ergonomie**.

Cependant, c'est une notion difficile à estimer, car chaque utilisateur perçoit d'une manière différente un nouvel outil en fonction de ses expériences passées. C'est pourquoi on estime le niveau d'ergonomie selon huit critères : la compatibilité, l'homogénéité, la concision, le pilotage, la rétroaction, la signifiante, la flexibilité et l'assistance.

2. L'ergonomie de la CMDB

Les points les plus travaillés lors de cette refonte ont été l'homogénéité, le pilotage, la signifiante et l'assistance, les autres critères étant déjà respectés dans la première version du site.

a) L'homogénéité

Le critère d'homogénéité, comme son nom l'indique, consiste à créer une interface homogène, ce qui permet de déclencher des automatismes chez l'utilisateur. Les pages principalement affectées par cette modification sont les pages de rapports. Elles contiennent toutes énormément d'informations, mais celles-ci sont très semblables (Caractéristique d'une VM/frame, tableau de bord...).

En adoptant une organisation similaire sur ces pages, l'utilisateur peut intuitivement deviner où se trouve l'information dont il a besoin.

b) Pilotage

Le pilotage est le critère qui permet de juger de la facilité de naviguer entre les actions. Pour améliorer ce point, de nombreux liens ont été ajoutés entre les pages du site. Une fois encore les pages les plus affectées sont les pages de rapports : l'intérêt est de pouvoir se déplacer rapidement sur la page d'un élément (VM ou frame) si une anomalie est détectée.

Ainsi, dès qu'un élément est nommé dans une page, il est possible d'accéder à sa page de rapport en cliquant sur son lien.

c) Signifiante

La signifiante est la capacité d'un élément de l'IHM à être significatif pour les utilisateurs. Ici, le respect de ce critère repose sur deux techniques : l'utilisation de couleurs qui sont-elles même significatives :

- Rouge pour les erreurs, le danger et l'affichage de machines éteintes.
- Jaune pour les anomalies.
- Bleu pour les indications.
- Vert pour le succès et l'affichage de machines allumées.

Et l'utilisation d'icônes pour représenter une action. Un bouton comportant une image de fichier déclenchera donc toujours le téléchargement d'un fichier, une croix indiquera toujours une suppression etc.

d) Assistance

L'assistance correspond à l'aide dont disposera l'utilisateur dans l'utilisation de l'interface, dans le traitement d'une erreur ou encore dans la protection des actions à risque.

Le site comporte très peu de champs texte et beaucoup de listes déroulantes, ce qui limite la saisie de valeur erronée.

Toutefois, si l'utilisateur doit remplir un champs texte, un code Javascript analyse la valeur entrée avant de soumettre le formulaire. Si une anomalie est détectée, un message apparaît pour indiquer à l'utilisateur son erreur, sinon le formulaire est envoyé.

De plus, chaque page du site contient des liens permettant d'envoyer des mails aux personnes travaillant sur la CMDB si l'utilisateur rencontre un problème.

3. Refonte graphique

La dernière étape concernant la modification de l'interface est la refonte graphique. Cette tâche consiste à modifier la mise en page du site pour lui donner un aspect plus professionnel. Bootstrap fournit un fichier CSS très fourni qui permet de créer une interface sobre et épuré très rapidement, elle n'a donc pas été modifiée.

Comme expliqué dans la partie sur l'IHM, l'information est découpée par blocs. Ce qui permet d'améliorer la lisibilité et de faciliter la recherche d'informations.

Le blanc et le gris sont désormais les couleurs de fond, par soucis de lisibilité (par rapport au vert de la première version).

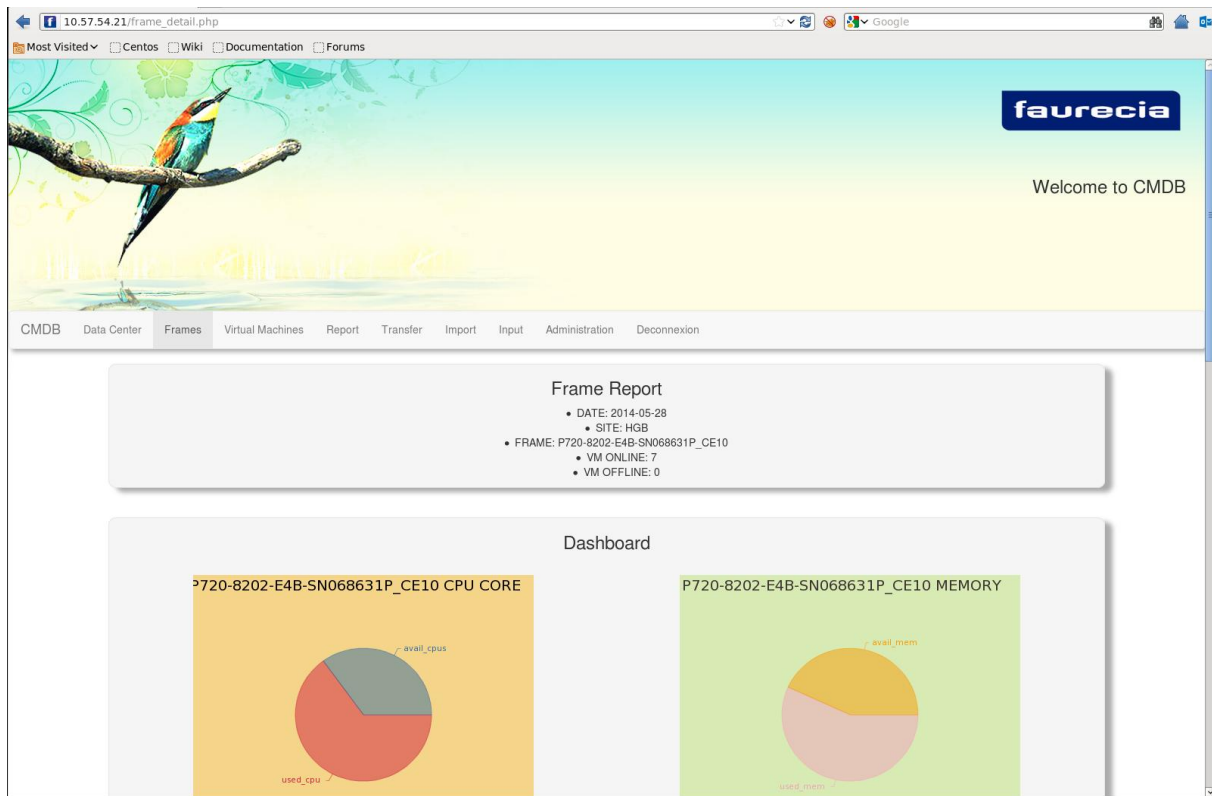


Figure 7 : La page de rapport de Frame au 06/06/2014 (avec Bootstrap)

D. Ajout de graphique avec Open Flash Chart 2

Les graphiques permettent à l'utilisateur d'obtenir une information plus rapidement que si elle était affichée dans un tableau HTML. Deux nouveaux graphiques ont besoin d'être ajoutés : le premier doit afficher le nombre chaque version installés d'un logiciel, le deuxième doit permettre de connaître l'espace disponible dans les répertoires des VM.

1. Procédure de création d'un graphique

La méthode de création de graphique est identique quelque soit le type : il faut dans un premier temps créer les éléments qui le composent (parts de camembert, lignes, axes, barres, titres...).

Ces éléments sont des **objets**, ils sont donc composés d'un certain nombre d'attributs et de méthode (fonctions spécifiques à un objet). Ces attributs sont remplis à l'aide de **setters** (méthode permettant de fixer les valeurs des attributs). On peut noter que certains éléments peuvent eux même être composés d'autres éléments (les lignes par exemple, qui sont composées d'objets point).

Dans un premier temps, on va définir l'apparence de l'élément (couleur, taille, forme...). Une fois l'apparence choisie, on va lui attribuer des valeurs, grâce à la méthode `set_data()`. Cette méthode prend en paramètre un tableau de valeur. Une fois ces éléments créés, on les lie avec un objet chart grâce à la méthode `add_element()` de la classe et on convertit l'objet créé au format JSON.

Il ne reste plus qu'à insérer le graphique dans la page à l'aide d'une fonction de la bibliothèque Open Flash Chart.

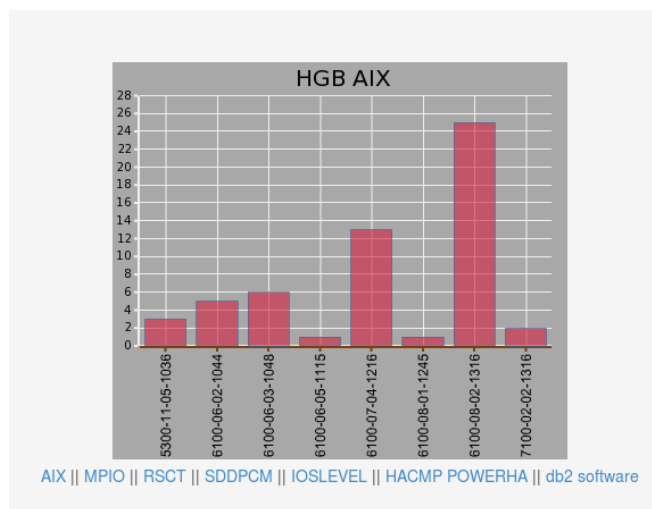


Figure 8 : Graphique des versions du logiciel AIX

2. Graphique dynamique

Le graphique permettant de visualiser l'espace disponible dans les répertoires pose toutefois un problème : la différence de taille peut être extrêmement importante entre deux répertoires, de ce fait, certains ne sont pas visibles.

Pour corriger ce problème ; il faut créer une option de zoom. Comme dit précédemment, les graphiques sont stockés dans le format JSON. Les éléments du graphique peuvent être manipulés comme des tableaux dans cet objet. Pour zoomer, on va donc diviser par deux la valeur maximum de l'axe des ordonnées, et réutiliser la fonction d'affichage de la bibliothèque OFC. Cela aura pour conséquences d'augmenter la taille des barres.

Si l'utilisateur veut dézoomer, on procédera de la même manière, mais au lieu de diviser la taille de l'axe des ordonnées, on la multipliera.

E. Lpar2rrd

Lpar2rrd est un outil qui permet de surveiller les frames et les VM sur lesquelles il est installé. Sa fonction principale est de générer des graphiques à partir des données collectées en temps réel, et de les afficher dans une interface Web. Il produit également des fichiers d'alertes lorsqu'il détecte une activité inhabituelle sur une machine (virtuelle ou physique), mais propose également des optimisations quant à la répartition des composants d'une frame entre ses VM.

Récupérer les données depuis lpar2rrd pour les intégrer dans les pages de la CMDB est une des fonctions à mettre en place en priorité car cela simplifiera le travail de surveillance en centralisant les informations.

1. Les graphiques

Les graphiques lpar2rrd sont générés grâce à un script PERL. Ce script s'exécute depuis une URL et affiche le résultat sous la forme d'une image au format .png.

Insérer cette information dans une page web est donc extrêmement simple, car il suffit de connaître le lien vers script, et les arguments à lui passer. On regroupe ces deux informations dans une URL (en passant donc les arguments par la méthode GET), et on insère le résultat dans l'attribut src d'une balise image.

Ainsi lorsqu'une page contenant des graphiques est chargée, une fonction PHP se charge de créer tous les liens nécessaires en fonction des informations saisies précédemment par l'utilisateur (Data Center, Frame et éventuellement VM), et une boucle permet ensuite de créer les balises images et d'insérer les liens dans ces balises.

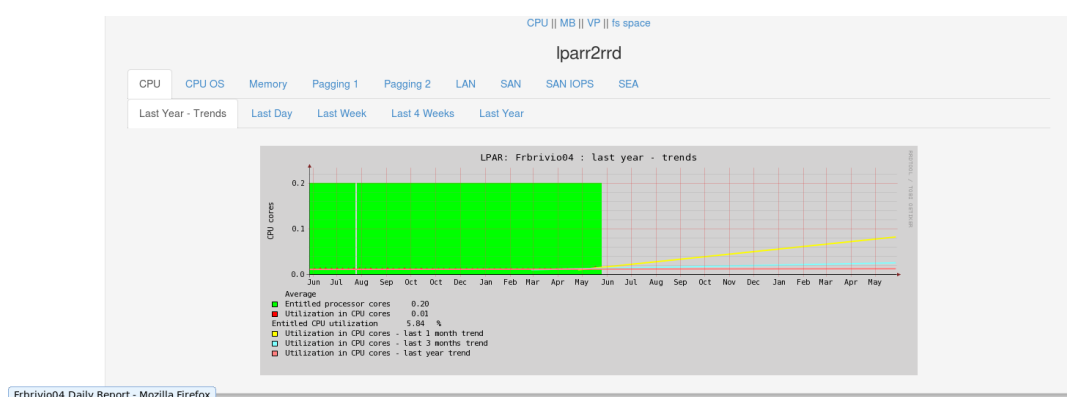


Figure 9 : Un graphique lpar2rrd

2. Les fichiers

L'organisation des fichiers varie selon leurs contenus.

Les fichiers logs (alerte, erreur), contiennent entre 0 et 500 lignes, et chaque ligne correspond à un message d'erreur. Les fichiers d'optimisation sont eux, au format CSV (Comma Separated Value), c'est-à-dire que l'information est également organisée en ligne, mais les valeurs sont séparées entre elles par un caractère séparateur (généralement une virgule ou un point-virgule).

Les fichiers CSV sont assez simples à analyser. En effet il existe des fonctions spécifiques au traitement de ce format en PHP, notamment la fonction fgetcsv(). Elle permet d'analyser un fichier de ce type ligne par ligne en renvoyant leur contenu sous la forme d'un tableau.

Les fichiers logs sont un peu plus complexes à gérer. Il faut disposer de certains droits pour y accéder, or la plupart des utilisateurs de la CMDB n'y ont pas accès, cependant son contenu est accessible depuis une page Web.

Il est donc possible de contourner le problème des droits en récupérant directement le contenu du fichier depuis la cette page.

Pour cela, on utilise la fonction `file_get_contents()` qui va récupérer le code HTML de la page. On va stocker le résultat dans une variable et en supprimer toutes les balises, afin de ne garder que les logs.

Il est ensuite possible d'analyser le fichier car chaque ligne possède le même nombre de chaînes de caractères. Chaque chaîne est conservée dans une variable en utilisant la fonction `scanf()`. Celles qui sont utiles sont concaténées et stockées dans un tableau. Chaque case de ce tableau contient donc une erreur.

Dans les deux cas, on traite les tableaux renvoyés directement dans la page HTML.

F. La création de rapport

L'objectif principal de la CMDB est de stocker des données concernant la configuration des machines présentes dans les Data Center pour permettre aux utilisateurs d'en assurer le suivi. Cependant ces derniers n'ont pas accès à la base de données en elle-même, uniquement à l'interface Web.

Cette interface leur fournit les informations sous la forme de rapports.

Ces rapports sont des requêtes SQL dont le résultat est affiché sous la forme d'un tableau HTML, exportable sous Excel.

Toutefois, ils présentent un inconvénient. Pour répondre aux besoins de tous, ils affichent le résultat d'une requête sur l'ensemble des attributs d'une table. Or, chaque utilisateur peut avoir besoin d'informations très spécifique pour travailler.

Pour répondre à ce besoin, il faut créer un formulaire qui simule une interface de gestion de SGBD comme le fait PhpMyAdmin avec MySQL (toute proportion gardée).

1. Les limites

Ce formulaire permet uniquement d'effectuer de requêtes de sélection, et seulement sur des tables qui peuvent lui être utiles. Par exemple, un membre du service d'exploitation n'aura pas accès aux tables de **DRP**.

Il ne permet pas non plus d'utiliser de clause `WHERE`, qui filtre les résultats.

2. Réalisation du formulaire

Ce formulaire est séparé en deux parties : une liste déroulante permettant de choisir la table à parcourir et des cases à cocher (check box) qui correspondent aux attributs de la table sélectionnée.

Lorsque l'utilisateur choisit une table, une requête AJAX récupère la liste des attributs de cette table et les affiche sous la forme de check box. Cette méthode permet d'éviter toute erreur de saisie de l'utilisateur, et toute tentative d'**injection SQL**. Lorsque l'utilisateur clique sur une case, un script génère une requête SQL à partir des valeurs sélectionnées dans le formulaire.

Cette requête est elle aussi envoyée en AJAX vers un script PHP qui génère un tableau HTML avec son résultat. Cela permet ainsi à l'utilisateur d'avoir en permanence un visuel du rapport qu'il est en train de générer.

The screenshot shows the 'Generate Report' interface in the CMDB application. The 'Table' dropdown is set to 'frames'. Under 'Attribute', several checkboxes are selected: collect_date, site, frame_name, serialnb, and lpar_number. Below the form, a table displays the generated report data.

collect_date	site	frame_name	serialnb	lpar_number
2014-05-27	HGB	P720-8202-E4B-SN068631P_CE10	068631P	7
2014-05-27	HGB	P750-8233-E8B-SN063C1BP_CE10	063C1BP	13
2014-05-27	HGB	P750-8233-E8B-SN063C1CP_BE18	063C1CP	15
2014-05-27	HGB	P770-9117-MMB-SN063C1DP_BE18	063C1DP	2
2014-05-27	HGB	P770-9117-MMB-SN063C1EP_CE10	063C1EP	2
2014-05-27	HGB	US_MEL-P710-8231-E2B-SN06750FP	06750FP	1
2014-05-27	HGB	US_PLY-P710-8231-E2B-SN067510P	067510P	1
2014-05-27	HGB	deaugunia1001	06B8692	1
2014-05-27	HGB	ora002-8204-E8A-06B8602-BE18	06B8602	1
2014-05-27	HGB	p720_DEMO-8202-E4B-SN0602B9R	0602B9R	1
2014-05-27	HGB	p740-8204-ERD-SN06A8R1T_CE10	06A8R1T	25

Figure 10 : La page de création de rapport

G. Modification du système d'administration

La CMDB va être utilisé par deux services : l'infrastructure et l'exploitation (qui supporte et maintient l'infrastructure déployée). Les besoins de ces services sont différents, c'est pourquoi un système d'authentification a été mis en place.

Il permet de délivrer des informations différentes selon le service de l'utilisateur. Ainsi toutes les fonctions de déploiement sont invisibles pour un compte appartenant à un membre de l'exploitation.

Le système d'authentification actuel utilise la base de données utilisateur de PhpMyAdmin, ce qui ne permet pas de gérer les droits correctement. On va donc créer une table utilisateur dans la base de données de la CMDB pour pouvoir gérer plus simplement l'ajout, la suppression et la modification des membres

La table membre est peu complexe. En effet on a besoin de très peu d'informations : un identifiant utilisateur, un login, un mot de passe et un rang.

Il faut bien évidemment **hacher** les mots de passes. Pour cela on utilise la fonction sha1() qui est plus sécurisée que md5(). En utilisant cette fonction, on sait par avance que

les mots de passes stockés dans la base de données feront 40 caractères. Il faut donc prendre cet élément en compte lors création de la table.

Le rang est stocké sous la forme d'un petit entier qui permettra, d'effectuer des tests pour afficher ou non dans les pages Web.

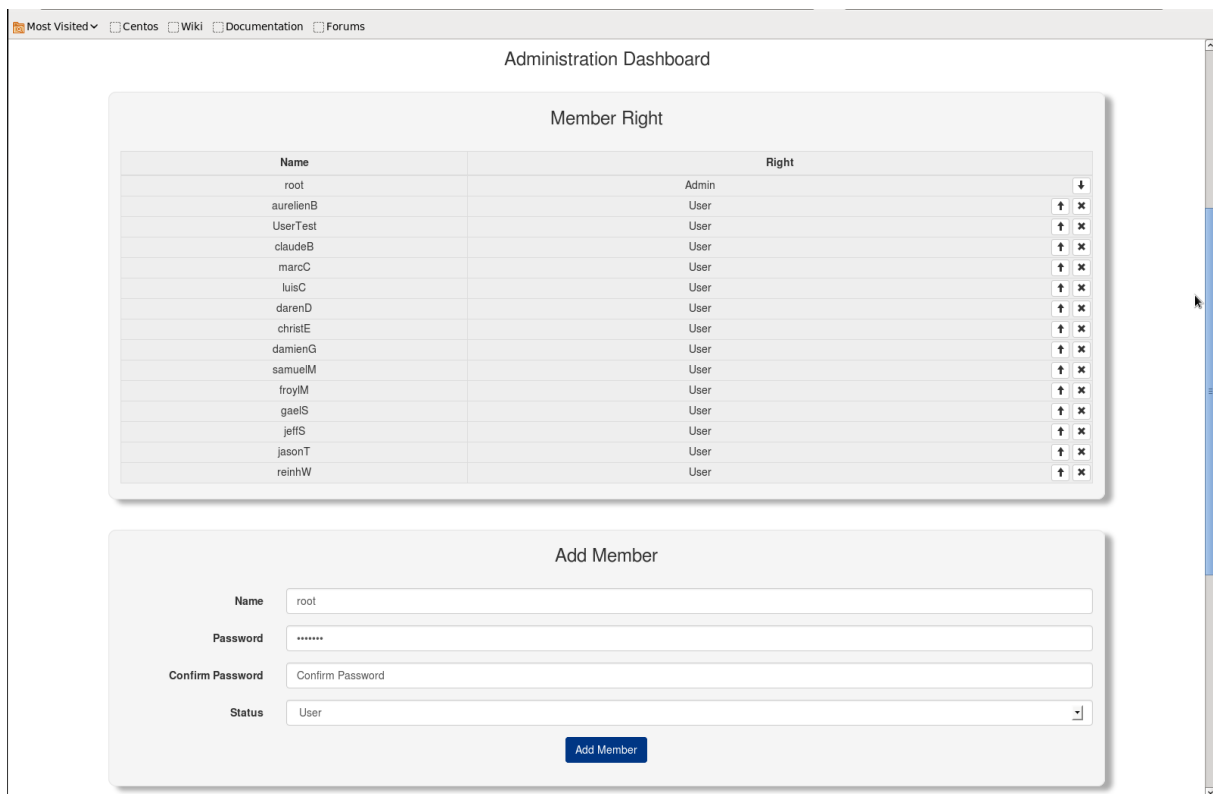


Figure 11 : Page d'administration

H. Déplacement d'une machine virtuelle

Il est possible, par le biais de la HMC, de déplacer une VM d'une frame vers une autre, grâce à un outil appelé Live Partition Mobility, et ce, sans arrêt de la machine.

Avant d'effectuer cette opération, il est nécessaire de vérifier que le transfert est possible. Pour cela il existe une commande sur la HMC, mais il serait plus pratique de connaître cette information directement depuis la CMDB.

1. Procédure

Pour qu'une VM soit transférée, il faut que 3 conditions soient respectées :

- La frame d'arrivée possède suffisamment de composants pour faire tourner la VM.
- La frame d'arrivée doit se trouver dans le même Data Center que la frame de départ.
- Les deux frames doivent avoir l'option Live Partition Mobility activée.

Si ces 3 points sont respectés, on peut alors lancer la commande de transfert depuis la HMC.

2. Mise en place

Cette fonctionnalité n'a pas été achevée par manque de temps, cependant la première partie des tests a été réalisée. Actuellement l'utilisateur sélectionne une VM et, à partir de ce choix un script PHP renvoie la liste des frames candidates à l'hébergement de la VM.

La deuxième partie de ce test consiste à supprimer de cette liste toutes les frames dont l'option Live Partition Mobility est désactivée, cependant, cette information n'est pas stockée dans la CMDB. Pour achever ce script il existe deux solutions :

a) Modification des scripts d'insertion

La première solution consiste à modifier les scripts d'insertion dans la table frames pour stocker cette option.

Dans ce cas il suffit de préciser dans la requête SQL qui récupère la liste des frames du Data Center de ne sélectionner que celles dont le nouvel attribut `live_partition_mobility` sont à la valeur `true` (car un booléen serait probablement la meilleure option pour stocker cette donnée).

b) Créer un script qui récupère cette information

On peut également créer un script qui récupère cette information, et l'exécuter grâce à la fonction `exec()`. Cette fonction renvoie le résultat de la commande passée en argument dans un tableau. Il suffit donc que le script affiche la liste des frames candidates lors de son traitement pour pouvoir exploiter la liste dans l'interface Web.

Ces deux théories n'ont pas pu être testées, cependant elles sont relativement simples à mettre en place, surtout la première qui ne nécessite qu'une modification de ligne dans le script PHP.

Site	Frame	Note
HGB	P720-8202-E4B-SN068631P_CE10	
HGB	P770-9117-MMB-SN063C1DP_BE18	You have to install more Memory 16384/3584 (COD : 262144)
HGB	P770-9117-MMB-SN063C1EP_CE10	You have to install more Memory 16384/3072 (COD : 262144)

Figure 12 : Ecran de transfert de VM

V. Conclusion

Ce stage fut une expérience très enrichissante, tant sur le plan technique (apprentissage de nouveaux outils comme AJAX et Doxygen) que sur le plan professionnel, car ce je n'avais encore jamais travaillé en entreprise.

Grâce à mon maitre de stage, j'ai également eu la chance de découvrir un aspect du monde de l'informatique dont je n'avais jusqu'à présent qu'une vague idée à savoir le déploiement de cloud. Je tenais donc à le remercier une fois de plus pour tout le travail qu'il a fourni afin de me faire comprendre l'utilité de mon travail, notamment grâce à la visite d'un des Data Center de l'entreprise.

J'ai disposé d'une grande liberté pendant ce stage ce qui était à la fois un avantage et un inconvénient car j'ai pu travailler de façon très autonome, mais j'ai dû apprendre à répartir mon temps de travail entre deux activités, à savoir l'écriture du code et sa documentation, et je pense que cela a été mon point faible pendant ce stage.

En effet toutes les fonctionnalités demandées ont été ajoutées et commentées afin d'être facilement maintenable, mais je pense que le code aurait encore pu gagner en lisibilité en poussant la rationalisation et la documentation des sources originales encore plus loin.

Dans l'ensemble, je tire toutefois un bilan positif de ces deux mois car ils m'ont permis de confirmer mon intérêt pour le développement Web, ce qui m'incite à poursuivre mes études dans cette voie.

VI. Lexique

AJAX : Concept de programmation Web dont l'objectif est de faire communiquer une page et un serveur Web sans déclencher de rechargement de la page.

AIX : Système d'exploitation de type UNIX commercialisé par IBM. C'est l'OS installé sur toutes les VM de Faurecia.

Alpha : L'alpha est une version interne d'un logiciel. Il est accessible à un nombre réduit de personnes, ne dispose pas encore de toutes ses fonctionnalités et comporte encore un nombre important de bugs.

Capacity on Demand (CoD) : Option permettant d'installer des composants supplémentaires sur une frame.

Cloud : Ensemble de processus consistant à utiliser la puissance de calcul et/ou de stockage de serveurs informatiques distants à travers un réseau. Chez Faurecia, les clouds sont composés de VM et sont chacun dédié à une seule application.

Data Center : Les Data Centers sont des sites physiques sur lesquels sont regroupés les équipements constituant le système d'information de l'entreprise (ordinateurs centraux, baies de stockage, équipements réseaux et de télécommunications, etc.).

Déploiement : Processus de mise en place d'un cloud.

Doxygen : Logiciel libre capable de produire une documentation logicielle à partir du code source d'un programme et de ses commentaires.

DRP : Disaster Recovery Planning. Désigne un outil permettant de reconstruire rapidement un *cloud* en cas de défaillance matérielle.

Ergonomie : L'ergonomie est l'adaptation du travail à l'homme.

Frame : Serveur utilisé pour créer des machines virtuelles. Elles sont contenues dans des *Data Center*.

Hachage : Les fonctions de hachage sont des fonctions qui servent à calculer des empreintes à partir d'une donnée passée en paramètre.

HMC : Hardware Management Console. Petit serveur permettant de communiquer avec les frames d'un Data Center via un processeur dédié appelé *hyperviseur*.

Hyperviseur : Processeur d'une frame utilisé pour communiquer avec la HMC.

Injection SQL : Une injection SQL est l'exploitation d'une faille de sécurité d'une application communiquant avec une base de données. Elle repose sur l'envoi d'une requête SQL non prévue par le système généralement par le biais d'un formulaire.

Interface Homme Machine (IHM) : Interface permettant à l'Homme de communiquer avec un système. La qualité d'une IHM est jugée sur son *ergonomie*.

JSON : Format de données textuelles générique dérivé de la notation des objets Javascript. Il permet de représenter une information de manière structurée.

LAMP : **L**inux **A**pache **M**ySQL **P**HP. Ensemble de logiciel libre qui permettent de créer des serveurs Web.

Licence MIT : Licence logiciel permettant à toute personne recevant le logiciel de l'utiliser, le copier, le modifier, le fusionner, le publier, le distribuer, le vendre et de changer sa licence. Ces droits sont également valables pour des entreprises.

Lpar2rrd : Logiciel de monitoring des Frames et des VM.

Objet : La Programmation Orientée Objet est un style de programmation qui repose sur l'utilisation de briques logicielles appelées objets. Ces objets comportent des variables qui leurs sont propres (attributs) et des fonctions (méthodes). On interagit avec les attributs de ces objets à l'aide de méthode appelées setters.

VM : Machines virtuelles hébergées dans des frames. Elles sont regroupées en *cloud*.