



RAPPORT ANNUEL D'ALTERNANCE 2020 - 2021

**Maître d'apprentissage
M. Aymeric BLINTZ**

le cnam
Paris

**Tutrice
Mme. Nadira LAMMARI**



**Chargé de mission CFA AFIA
Mme. Angélique LANEAU**

Ivan CIPRANIC FIP 1

Remerciements

Je tiens à remercier mon maître d'apprentissage M. Aymeric BLINTZ ainsi que mes collègues de la Webteam pour leur encadrement, le savoir-faire ainsi que les compétences dont ils m'ont fait part durant cette année ce qui m'a permis de me développer professionnellement.

Je tiens également à remercier ma tutrice Mme. Nadira LAMMARI pour ses conseils ainsi que l'aide apportée pour la réalisation de ce rapport.

Enfin, je souhaite remercier l'équipe pédagogique pour les connaissances apportées tout au long de cette année ainsi que notre responsable du pôle alternance Mme. Angélique LANEAU pour les différentes offres dont elle nous a fait part ainsi que l'aide apportée durant notre recherche d'alternance.

Table des matières

I - Introduction

II - Présentation de l'entreprise

2.1 - Présentation du groupe BNP Paribas

2.2 - L'équipe WEBTEAM BDDF IT

2.3 - Socles techniques du service Webteam

2.3.1 Répartitions des applications sur les serveurs

2.3.2 MySQL Workbench et Serveur de base de données

2.3.3 Ancien socle WTF (serveur Linux, Redhat, Windows)

2.3.4 Nouveau socle Symfony Angular

III - Missions

3.1 - L'installation, le début,

3.2 - Refonte de WEP (WEBteam Portail)

3.2.1 Contexte de la mission

3.2.2 Qu'est-ce qu'une CMDB ?

3.2.3 Choix de la solution à mettre en place

3.2.4 Choix de la CMDB

3.2.5 Installation, premiers tests et problèmes rencontrés

IV - Première réécriture sous Symfony et Angular

4.1 Attribution du sujet

4.2 Évolution à venir

4.3 Structure de l'application

4.4 Difficultés rencontrées

V - Apports personnel

5.1 Apports techniques

5.2 Apports généraux

I - Introduction

Dans l'optique d'acquérir une première expérience professionnelle ainsi que des compétences directement issues du monde professionnel, j'ai décidé de suivre la formation en alternance d'ingénieur en informatique et systèmes de communication au sein du CNAM. Attiré par un poste dans le domaine du développement web, j'ai décidé de postuler en tant qu'analyste-développeur afin de pouvoir mettre à profit mes compétences acquises lors de mes années de DUT en matière de développement, mais aussi découvrir le monde professionnel.

Mon alternance se déroule en tant qu'apprenti analyste-développeur au sein du groupe BNP Paribas. J'ai intégré le service Webteam de l'entreprise, avec pour mission principale, la réécriture de différentes web applications internes sous un nouveau socle technique Symfony et Angular.

À travers ce rapport, je souhaite partager mon expérience de cette année d'apprentissage qui fut très riche. Vous trouverez au travers de ce document une présentation de l'entreprise dans laquelle j'ai réalisé la première année de mon contrat d'apprentissage ainsi que le service dans lequel j'ai œuvré. Par la suite, je détaillerai les missions réalisées au cours de l'année et plus particulièrement le projet qui m'a été attribué, avant de faire un bilan de cette année d'apprentissage.

II - Présentation de l'entreprise

2.1 - Présentation du groupe BNP Paribas

Présentation générale :

BNP Paribas est le premier groupe bancaire français. Ce groupe né de la fusion entre la BNP et Paribas est aujourd'hui la première banque de la zone euro et le premier groupe bancaire mondial en termes d'actifs.

Il est présent dans plus de 80 pays et compte environ 200.000 collaborateurs à travers le monde. Il dispose de l'un des plus importants réseaux bancaires internationaux.

Ses actions sont cotées sur le compartiment A d'Euronext Paris et entrent dans la composition de l'indice CAC 40.

BDDF :

La Banque de Détail en France (BDDF) fait partie du pôle Retail Banking du groupe BNP Paribas. Elle propose des solutions innovantes en matière de financement, moyens de paiement, gestion de patrimoine et assurances à ses 6,8 millions de clients particuliers, 600 000 clients professionnels et TPE, 31 000 entreprises (PME, ETI, grandes entreprises) et plus de 58 000 associations.

BDDF est composée de plus de 25 000 collaborateurs. Son rôle est de proposer différents services allant des offres de services bancaires à l'accompagnement personnalisé par des équipes expertes de divers domaines traités. Le rôle de BDDF est également de contribuer aux transformations nécessaires des modèles de production et de consommation tout en répondant aux besoins environnementaux et sociaux actuels.

2.2 - L'équipe WEBTEAM BDDF IT

Mon alternance se déroule au sein du service WEBTEAM de BNP Paribas. Elle est constituée de 2 teams managers, 6 chefs de projet et responsable d'applications répartis sur les 2 équipes, 7 ingénieurs développement internalisés (mutualisés) et 2 alternants.

Le service WEBTEAM est chargé de développer de nouvelles applications web internes de BNP Paribas ainsi que de maintenir son patrimoine applicatif web interne. Ce dernier est constitué de 125 applications intranet partagées entre les différents Responsables d'applications qui suivent les anomalies et les différentes demandes liées à ces applications. Ces applications sont réparties sur 78 serveurs (VM2016 ou Linux) et sont administrés et pilotées par l'équipe de production Webteam et ont les critères d'éligibilité suivant :

- Intranet uniquement
- Pas de données secrètes
- Disponibilité Low
- Moins de 1000 Users en simultané

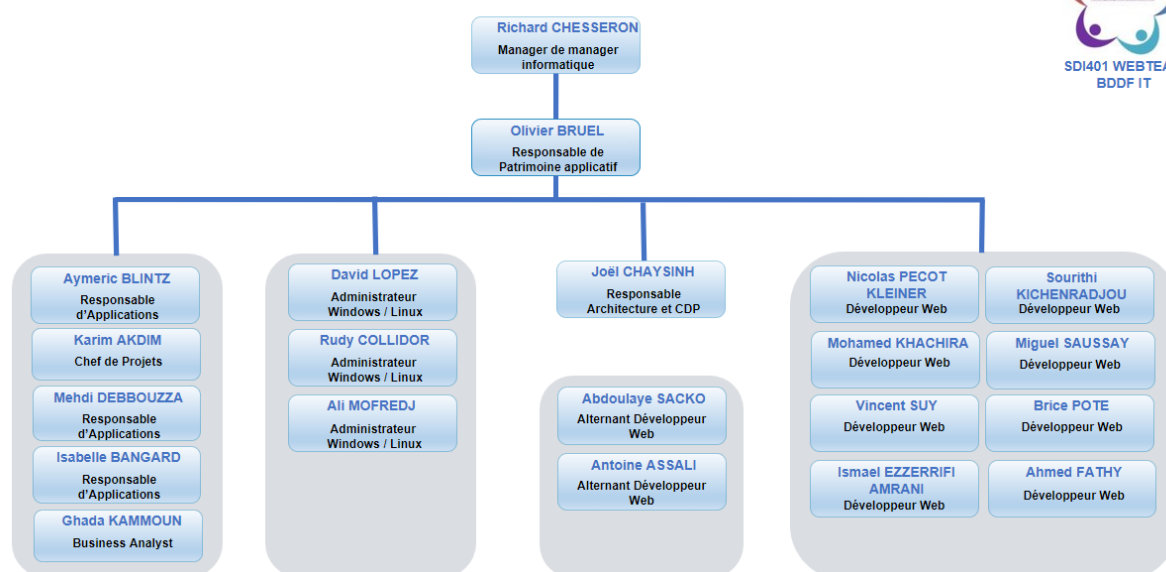
Les missions de la Webteam sont :

- La conception, le développement et la maintenance de sites web internes :

- En PHP, Apache, MySql ou POSTGRE
- Sur Windows ou Linux
- En environnement sécurisé (demilitarized zone DMZ) ou Intranet
- Spécificités :
 - En charge de l'exploitation en production des applications PHP développées, avec un hébergement assuré par IPS/BP2I

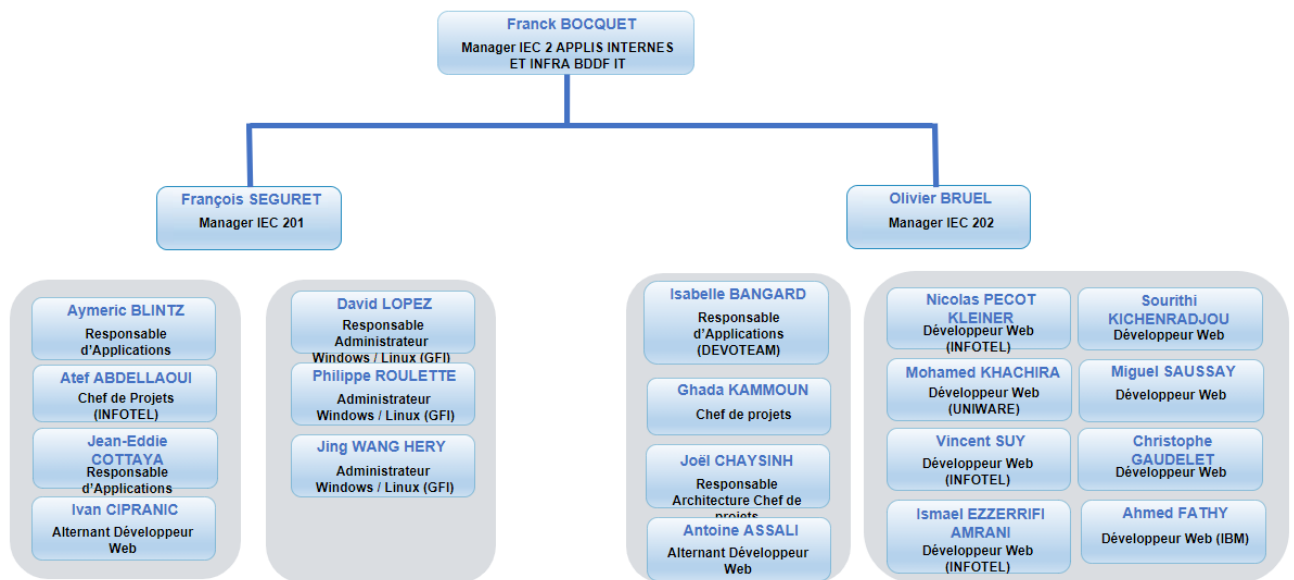
Durant cette année d'alternance, la Webteam a fait l'objet d'une réorganisation.

Lors de mon arrivée, l'équipe était sous la responsabilité de Richard CHESSERON et dirigée par un responsable du patrimoine applicatif.

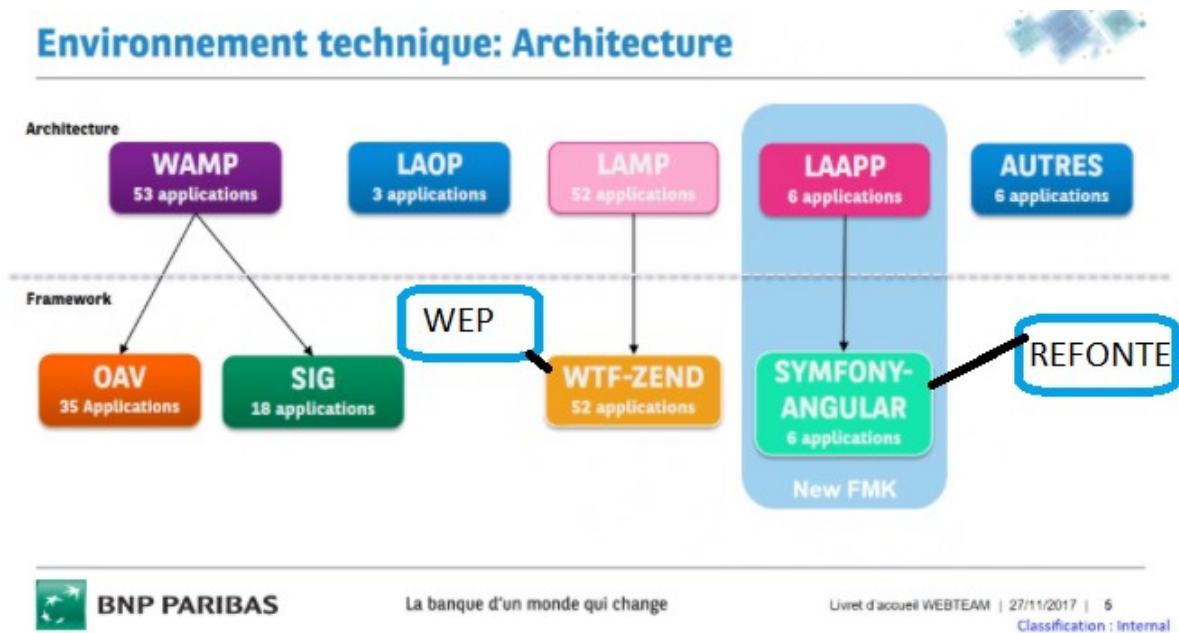


On peut voir qu'il y a un premier groupe composé des responsables d'applications, chefs de projets et Business Analyst, un deuxième groupe qui est l'équipe de production et un autre groupe constitué par les développeurs.

Désormais, et ce depuis janvier 2021, la Webteam est sous la responsabilité de Franck BOCQUET et dirigée par 2 manager d'équipes : François SEGURET et Olivier BRUEL.



2.3 - Socles techniques du service Webteam



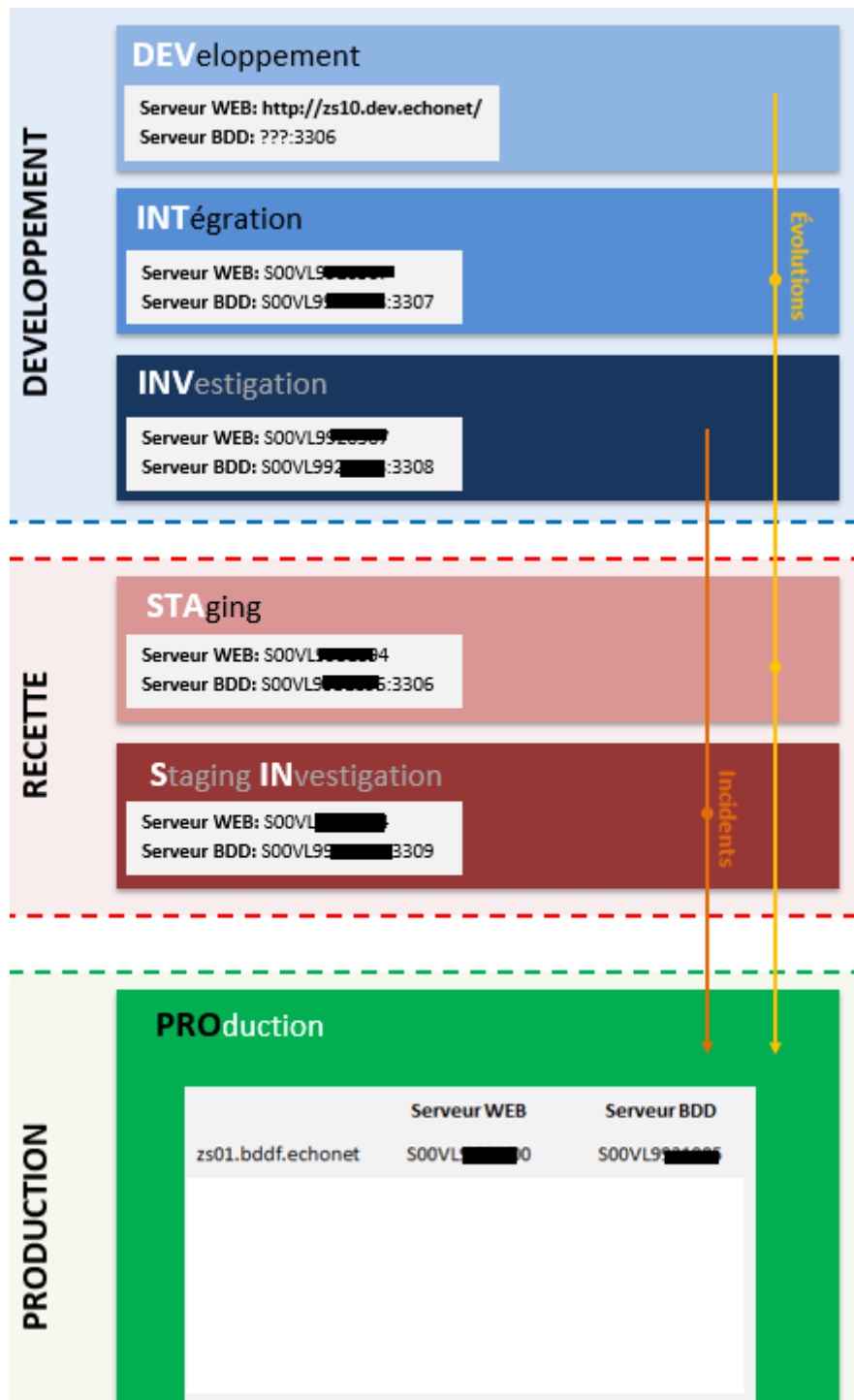
Les missions de mon alternance se situent majoritairement dans la partie LAAPP.

2.3.1 Répartitions des applications sur les serveurs

Les web applications internes sont réparties sur 3 serveurs distincts :

- Serveurs de développement
- Serveurs de recette
- Serveurs de production

Ils y sont répartis en fonction de leur état d'avancements.



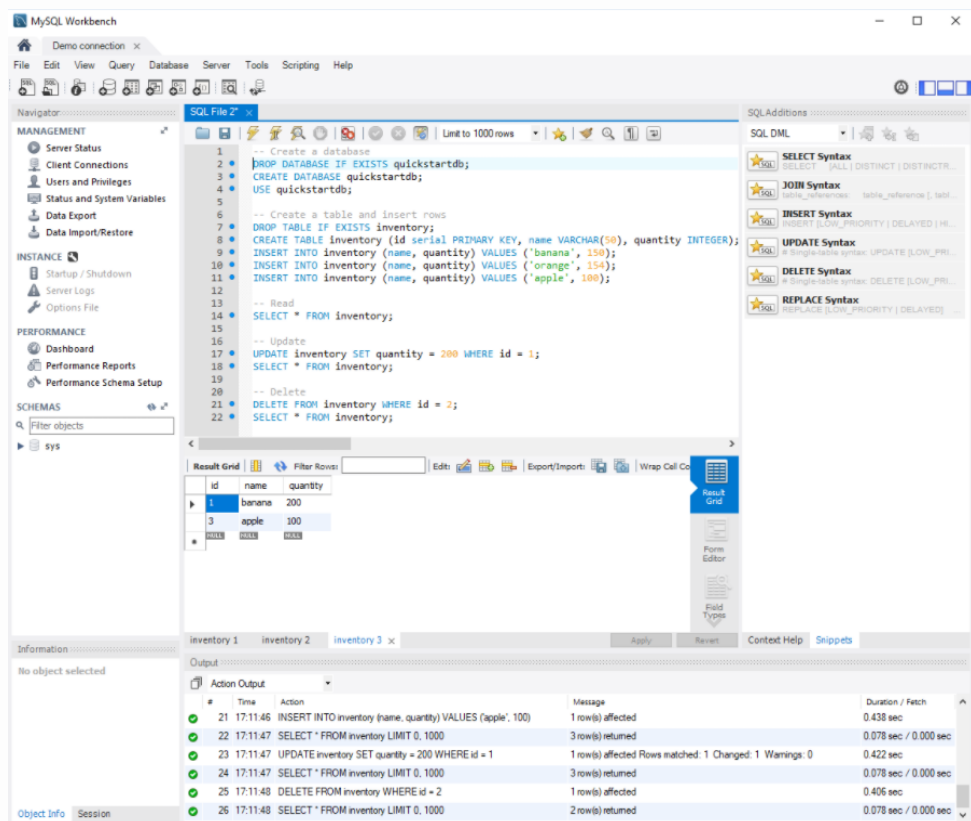
2.3.2 MySQL Workbench et Serveur de base de données

L'outil le plus utilisé afin de visualiser et modifier les bases de données est MySQL Workbench. Il est disponible sous Windows, Mac et Linux et permet de gérer des tables (ajout, modification, suppression) à l'aide d'une interface graphique simple d'usage.

L'interface graphique permet de réaliser un modèle de base de données qui inclue les interconnexions entre les tables mais aussi de générer le code SQL qui résultant du modèle. De plus, une connexion peut être réalisée avec une instance MySQL afin d'importer un modèle sur un environnement de développement ou de production. On peut également effectuer l'opération inverse et déduire la modélisation d'une table à partir d'une instance réelle d'une base de données MySQL.

MySQL Workbench permet notamment :

- La sauvegarde et restauration de tables
- Export des bases de Microsoft SQL Server vers MySQL
- Le management de connexion aux différents serveurs de base de données
- Effectuer des requêtes SQL variées



2.3.3 Ancien socle WTF (serveur Linux, Redhat, Windows)

Le socle technique sur lequel se trouve le plus d'applications est basé sur l'architecture LAMP (Linux, Apache, MySQL, PHP) et utilise les framework interne WTF qui est un framework PHP.

Plusieurs architectures sont utilisées :

- LAOP : Linux, Apache, Oracle, PHP
- WAMP : Windows, Apache, MySQL, PHP
- LAMP : Linux, Apache, MySQL, PHP
- LAAPP : Linux, Apache, Angular, Postgre, PHP

Les serveurs Windows sont des Windows Server 2016 et les serveurs Linux sont des distributions RedHat.

Les applications web internes sont disponibles sous différents navigateurs tels que edge, internet explorer et chromium. Certaines de ces applications sont actuellement sur PHP 4 et 5 et les nouvelles applications développées avec Symfony et Angular sont en PHP 7.

2.3.4 Nouveau socle Symfony Angular

La Webteam a entrepris de migrer ses applications vers un nouveau socle technique composé des frameworks Angular 7 et Symfony 4.



La partie Back End des nouvelles applications est désormais réalisée avec le framework Symfony 4. Symfony 4 est un framework PHP 7. Il offre des aides Ajax (Javascript asynchrone et XML), des plugins ainsi qu'une interface de générateur d'admin, ce qui rend la programmation d'applications complètes vraiment accessible. Symfony inclue également un système de routing. Grâce à l'annotation `@Route`, on peut facilement attribuer un chemin d'accès sous forme d'URL à des fonctions de l'application, qui seront donc appelés lorsque l'on effectue une requête à l'url formé par celle de l'api et de la fonction comme on peut le voir sur l'exemple ci-dessous :

Une application Symfony est décomposée en 3 parties distinctes :

- Une partie contenant les controllers, qui seront accessible facilement grâce au système de routage
- Les templates
- Les différentes classes d'entités
- Le dossier repository, qui contient les class apportant les fonctions permettant de récupérer des ensembles de données depuis la base de données
- Le dossier service, qui contient les class contenant la majeure partie de la logique de l'application

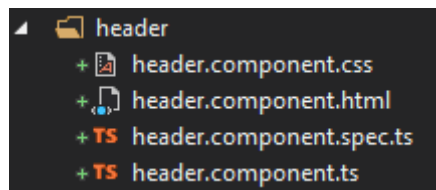
La partie Symfony n'intervient pas directement en Front End, elle fait office d'API qui fournit des données sous différents formats, notamment en JSON. Les templates Symfony ne seront donc, ici, pas utilisés. Elle gère les requêtes à la base de données et facilite ces dernières grâce à la librairie Doctrine qui permet de stocker des données sous formes d'objets, directement dans une table de base de données sans avoir à les décomposer en faisant appel à d'autres traitements.



La partie Front End est quant à elle réalisée sous Angular. Ce framework est orienté composants web. Un composant web est un fragment de l'application qui est autonome et indépendant du reste de l'application.

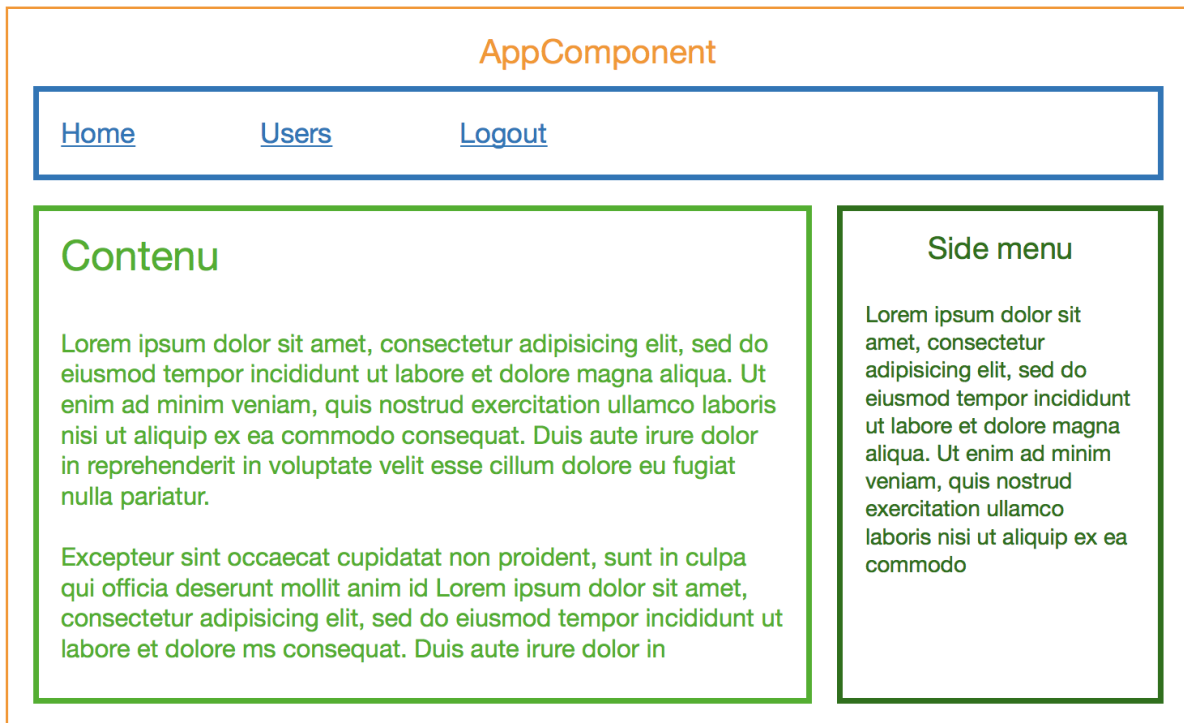
Un composant est constitué de 4 fichiers :

- Un fichier typescript qui contient les fonctions et la logique
- Un fichier html qui sert à générer la template du composant
- Un fichier spec.ts, contenant les tests unitaires
- Un fichier css ou scss afin de customiser la template du composant



Par exemple, une barre de recherche peut être considérée comme un composant web distinct. Une application Angular peut donc contenir plusieurs composants web capables de communiquer entre eux. Angular intègre également le concept de Single Page Application,

des applications qui tiennent sur une seule page. On aura donc en général un AppComponent, qui sera le composant principal qui contiendra et affichera les autres composants de l'application grâce à la balise <router-outlet> qui chargera dynamiquement les composants qui doivent être affichés à l'écran.



Ce chargement dynamique permet de réduire les chargements de pages puisqu'on ne recharge plus toute la page en entier mais uniquement certains composants affichés.

Les web components forment un standard qui n'est pas encore pris en charge sur tous les navigateurs, il faut donc utiliser des polyfill, qui sont des ensembles de fonctions simulant les fonctionnalités non prises en charge par ces navigateurs. La partie Angular s'occupe de générer l'interface que l'utilisateur verra lorsqu'il consultera une application et se charge de récupérer et afficher les données reçues par l'API sous Symfony.

Les applications Angular contiennent également des services. Les services sont des classes TypeScript dont l'instanciation est gérée par Angular. Il intègre le design pattern singleton puisqu'une seule instance de l'objet créé est utilisée dans l'ensemble de l'application. Ces services vont donc permettre de réutiliser du code dans tous les composants et services de l'application, ce qui facilite grandement l'échange de données.

C'est donc cette structure en service et composants qui permet à Angular de créer des applications modulaires facilement modifiables.

III - Missions

3.1 - L'installation, le début,

Un responsable d'application est chargé du suivi d'une ou plusieurs applications. Il mène des analyses poussées sur différents points :

- Les processus métiers
- Les anomalies existantes si l'application existait déjà lorsqu'il en prend la charge
- Les usages de l'applications

Ses fonctions sont donc :

- Suivre les évolutions, la gestion et l'analyse des anomalies
- Élaborer des tests
- Définir des solutions correctives
- Évaluer les coûts
- Contrôler les règles de fonctionnement et d'évaluation des applications

Les membres de la Webteam travaillent donc essentiellement sur :

- La maintenance des applications web internes déjà déployées
- Le développement de nouvelles applications web internes
- En ce moment, la migration des applications vers Edge, chromium

Au cours de mon premier mois j'ai pu suivre des formations internes diverses sur le site MyDevelopement. Ces formations servent à indiquer les bonnes pratiques lorsque l'on travaille dans le milieu bancaire et dans le monde de l'entreprise en général.

3.2 - Refonte de WEP (WEBteam Portail)

3.2.1 Contexte de la mission

Au sein d'une équipe s'occupant de plus de 120 applications réparties sur de nombreux serveurs, il est important de pouvoir consulter facilement les informations qui leurs sont liées pour réagir rapidement en cas d'incident ou de modification à apporter sur ces applications. La première mission qui m'a donc été attribuée est la refonte de WEP dans le cadre de la réécriture d'applications internes sous le nouveau socle technique Symfony et Angular. Il s'agit

d'une application web interne qui est utilisée principalement par le service de la Webteam. Elle consiste à répertorier l'ensemble du patrimoine applicatifs du service. On peut donc consulter les différentes informations liées aux applications internes telles que la liste de personne à contacter en cas de problème avec l'application, l'URL de production, de recette et de développement de l'application, savoir si elle utilise un fichier BATCH ou encore le serveur sur lequel elle se trouve ainsi que le nombre de connexion.

Voici ci-dessous une capture d'écran de l'onglet "Rechercher" de WEP. On peut voir les différents critères de recherches ainsi que différents éléments affichés directement dans la liste des applications comme le type de serveur et l'URL de production.

Accueil | **Rechercher** | Statistiques | Gérer | Tables de référence | Actualités | Affectations | ?

Accueil > Rechercher > Applications

Applications

Formulaire de recherche des applications

Nom de l'application ou code projet :

Serveur de l'application (Hostname) :

Serveur de la base de données :

Serveur de l'application (DNS) :

IDF Flux CFT :

URL Application :

Version OS :

Application critique :

Responsable :

Backup :

Version PHP :

Framework :

Statut Application :

Anonymisation des données clientèles :

Mes applications favorites : ☐

Liste des applications

175 élément(s)

<input type="checkbox"/>	Application	Statut Application	URL de production	Code projet	Responsable	Backup	Système d'Exp.	Version OS	Framework	Version PHP	
<input type="checkbox"/>	A&S - ALISE	Fermée	http://alise.bddf.echonet/				WINDOWS 2008	Win_2008	SIG	PHP 5	
<input type="checkbox"/>	A&S - ECAS (Estimation des Charges)	Fermée	http://ecas.bddf.echonet/					Win_2008	SIG	PHP 5	
<input type="checkbox"/>	A&S - OASIS	Fermée	http://oasis.bddf.echonet/		Aymeric BLINTZ	Patrice AGRON		Win_2008	SIG	PHP 5	
<input type="checkbox"/>	AD (Archivage débiteurs)	Ouverte	https://archivedebit.bddf.echonet/	prj1-208	Karim AKDIM	Aymeric BLINTZ	LINUX		WTF ZF6.3	PHP 5	
<input type="checkbox"/>	Administration du MDE	Ouverte	http://administration-mde.bddf.echonet/	prj1-130	Eddie COTAYA	Karim AKDIM	LINUX	RHEL 6.9		PHP 5	
<input type="checkbox"/>	ASTRO	Ouverte	http://astro.bddf.echonet/		Aymeric BLINTZ	Karim AKDIM	WINDOWS 2016		OAV	PHP 5	
<input type="checkbox"/>	AVANCE SUR BOURSE	Fermée	http://avancesurbourse.bddf.echonet/		Aymeric BLINTZ	Isabelle BANGARD	WINDOWS 2016		OAV	PHP 5	
<input type="checkbox"/>	Bases Mail	Fermée	http://basesmail.bddf.echonet/	n/a	Karim AKDIM	Joel			OAV	PHP 5	

Il s'agit donc d'une application importante dans la propagation d'informations au sein du service afin de centraliser les informations et avoir accès rapidement à l'écosystème d'une application.

Il était initialement prévu de la réécrire sous le nouveau socle Symfony, Angular. Cependant la possibilité de remplacement WEP par une CMDB (configuration management database) a été proposée par un membre de l'équipe.

3.2.2 Qu'est-ce qu'une CMDB ?

Ce concept étant nouveau pour moi, j'ai dû procéder à des recherches afin de me renseigner sur ce qu'était une CMDB puis effectuer une étude de l'existant afin de voir les différentes solutions proposées en ligne. Lors de mes recherches, j'ai découvert les nombreuses fonctionnalités d'une CMDB qui sont :

- Disposer d'un ensemble complet de données d'un environnement informatique à un endroit centralisé et accessible
- Intégrer les données provenant de sources externes (telles que les fournisseurs)
- Connaître la composition des actifs critiques et des composants desquels ils dépendent
- Comprendre quels actifs sont utilisés pour quels processus opérationnels et quels utilisateurs en dépendent
- Fournir des informations afin de soutenir la prise de décision concernant l'environnement informatique, les coûts d'exploitation et les choix technologiques.
- Fournir un système de ticketing permettant d'effectuer des demandes sur les éléments hardware et software inscrits dans la base de données

Une CMDB est donc similaire à WEP dans ses fonctionnalités de base, il fallait donc faire un choix entre la réécriture de WEP et mettre en place une CMDB.

3.2.3 Comparaison entre WEP et CMDB

La webteam n'étant pas familière avec les CMDB, j'ai réalisé un comparatif entre la réécriture de WEP sous le nouveau socle Symfony Angular et la mise en place d'une CMDB, visant à départager les deux idées afin de décider laquelle allait être mise en place.

Afin de savoir quelle solution était la plus adaptée, j'ai choisi les critères suivant pour ma comparaison :

- Le temps utilisé par chacune des solutions
- Le coût
- La marge de modification que l'on peut apporter
- Les besoins matériels et logiciels

Suite à mes recherches, j'ai donc déterminé, à l'aide des critères définis précédemment, que la mise en place d'un CMDB avait les avantages suivants :

- Outil déjà prêt proposant une base pour la structure de données
- Nombreuses fonctionnalités déjà incluses
- Des rendus statistiques plus clairs et plus visuels que ceux présent sur WEP
- Rapidement modifiable sans toucher au code si l'on souhaite ajouter une nouvelle catégorie d'information

La réécriture avait elle aussi des avantages qui étaient :

- La personnalisation des fonctionnalités plus libre
- Pas de licence à payer
- Maîtrise du code et choix des technologies utilisées

La mise en place d'un CMDB présentait également quelques inconvénient comparée à la réécriture de l'application. Ses inconvénients sont :

- Le possible coût de la licence
- Les besoins software et hardware du CMDB (les technologies utilisées ne font pas forcément partie de celles utilisées au sein de l'équipe habituellement)

Tandis que les inconvénients de la réécriture étaient :

- Le temps de développement (test, debugging ...)
- Modification du code à apporter si on souhaite rajouter une nouvelle catégorie d'information
- Fonctionnalités à repenser et ajouter en partant de 0

J'en suis finalement arrivé à la conclusion que la mise en place d'une CMDB était plus avantageuse que la réécriture, j'ai ensuite réalisé une présentation à mon maître d'apprentissage ainsi qu'à mon manager et d'autres collègues. Le résultat a donc été d'opter pour la mise en place d'une CMDB.

Une fois la solution choisie, il fallait maintenant déterminer qu'elle CMDB allait être sélectionnée et installée au sein du service.

3.2.4 Choix de la CMDB

Mon manager ainsi que mes collègues m'ont donné des critères que l'a CMDB sélectionnée doit respecter pour pouvoir être mise en place dans le service, afin que je puisse trouver de potentielles candidates à présenter durant une seconde réunion. Ces critères supplémentaires étaient :

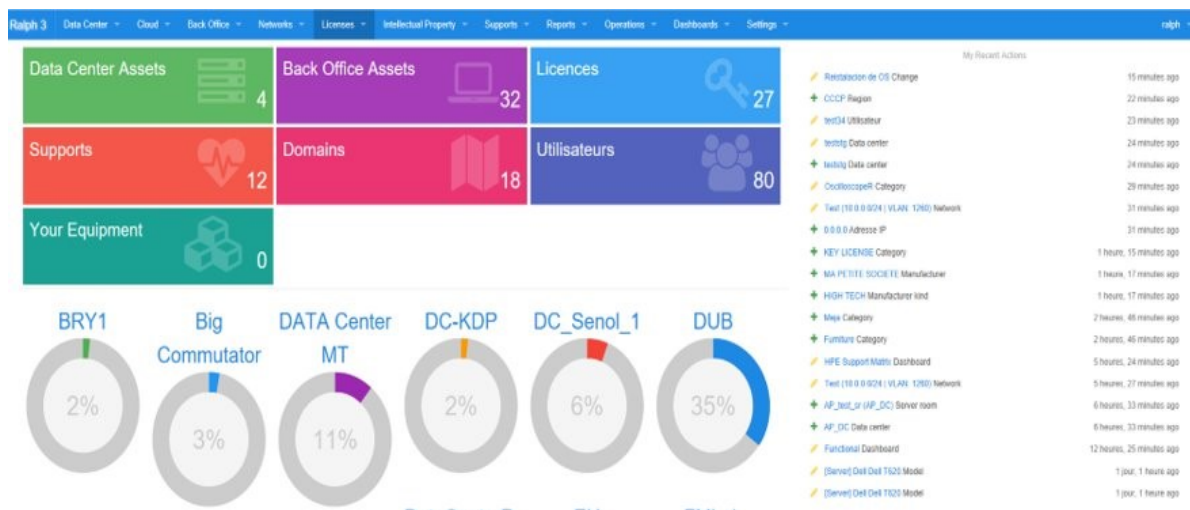
- La CMDB doit être open source, afin de pouvoir la modifier si besoin et continuer à la maintenir au cas où les éditeurs arrêtent les corrections
- La CMDB doit être SaaS, afin de ne pas stocker de données potentiellement sensibles à l'extérieur
- La CMDB doit être gratuite

Après plusieurs recherches et analyses de différentes CMDB proposées en ligne ainsi qu'à des tests effectués sur leurs démo, il y avait donc 3 candidats principaux qui répondent aux critères de sélection :



Ralph 3 :

Ralph 3 inclut les différentes fonctionnalités classiques d'une CMDB et intègre un historique des ajouts et modifications effectués, comme on peut le voir sur la capture d'écran ci-dessous.

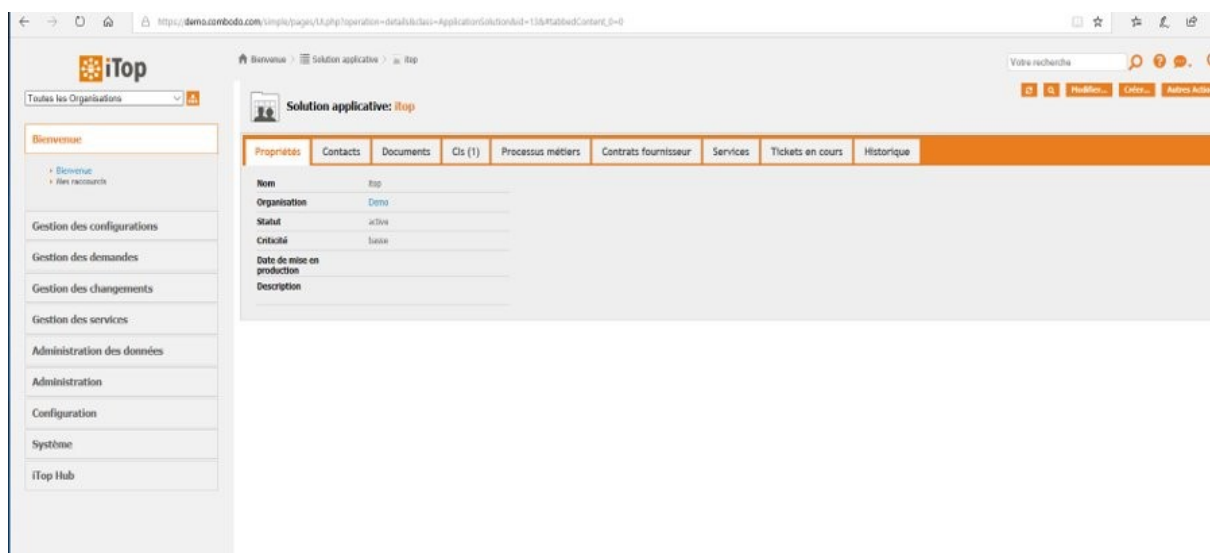


Il est écrit en Python 3 et utilise le framework Django, le système de gestion de base de données utilisé est MySQL.

Ralph 3 inclut les fonctionnalités suivantes :

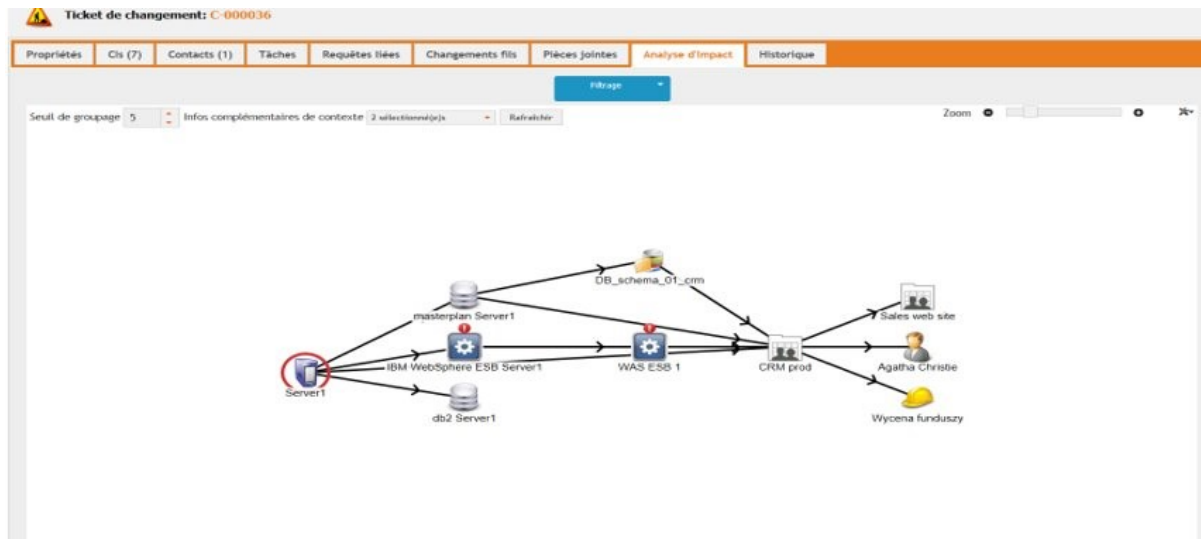
- Permet de répertorier différents types de données (applications, employé, etc.)
- Visualisation des historiques de modifications apportées
- Personnalisation graphique possible et simple
- Proche de ce que fait WEP tout en ajoutant des fonctionnalités plus visuelles telles que des graphes statistiques disponible sous plusieurs formes
- Permet l'ajout de nombreuses catégories de données à stocker rapidement (matériels, membres)
- Signaler un changement
- Génération de rapport (par exemple l'ensemble des logiciels qui ont un bug signalé au cours de la dernière semaine)
- Assignment de tâches à des employés
- Signaler un incident (logiciel, hardware ...)
- Possibilité de modifier les propriétés déjà présentes (ajout de label, catégories ...)

iTop :



iTop est la seule des candidates a intégré l'analyse d'impact.

Il s'agit d'une fonction permettant de visualiser sous formes de schéma, quels éléments seront affectés par un incident sur un logiciel, un serveur ou une base de données. De cette manière on peut avoir un meilleur aperçu d'un incident beaucoup plus rapidement qu'avec WEP.



Cependant, le schéma d'analyse d'impact n'est consultable qu'en cas de panne signalée, on ne peut pas y avoir accès en temps normal, on perd donc une part de vision sur l'environnement d'une application lorsqu'il n'y a pas de problème.

iTop inclut les fonctionnalités suivantes :

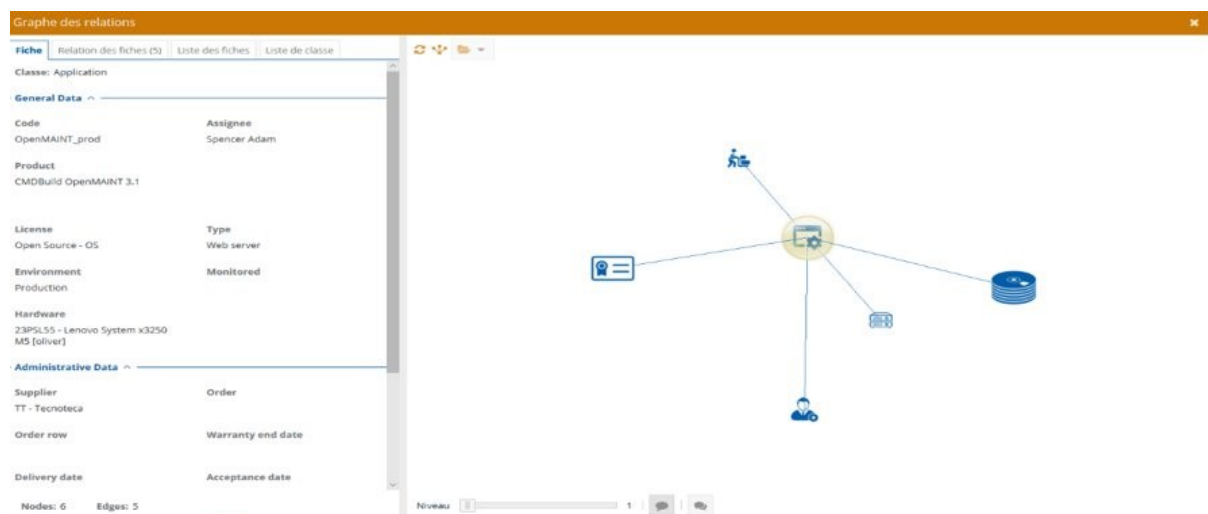
- Permet de répertorier différents types de données (applications, employé, etc.).
- Permet de visualiser l'impact d'un changement (par exemple quelles applications seront -touchées par un changement sur un serveur ou d'un OS)
- Assignment de tâches à des employés
- Signaler un incident (logiciel, hardware ...)
- Possibilité de modifier les propriétés déjà présentes (ajout de label, catégories ...).
- Rendu graphique des événements propres à un élément
- Possibilité de centraliser toutes les demandes et problèmes hardware et software
- Pas besoin de toucher au code (sauf fichier XML dans certains cas) pour réaliser des modifications

cmdBuild :

cmdBuild inclut les fonctionnalités suivantes :

- Permet de répertorier différents types de données (applications, employé, etc.).
- Signaler un incident (logiciel, hardware ...)
- Possibilité de modifier les propriétés déjà présentes (ajout de label, catégories ...).
- Rendu graphique des évènements propres à un élément
- Système de Relation Map

Ce qui démarque cmdBuild des autres candidats est son intégration de la notion de relation map. Une relation map est la représentation visuelle, sous forme de graphe, des liens entre un logiciel et son écosystème.



On peut voir sur l'image ci-dessus, prise lors de tests effectués sur la démo accessible en ligne sur le site de cmdBuild, dans la partie gauche, les différentes informations d'une application et dans la partie droite, la version graphique des liens entre une application et son/ses serveurs, ses fichiers batch ou encore les bases de données auxquelles elle accède.

Cette fonctionnalité peut s'avérer être un outil puissant pour la Webteam puisqu'il permet en un instant d'avoir une vision globale et claire des éléments principaux rattachés à une application.

cmdBuild est écrit en Java et JavaScript et le système de gestion de base de données utilisé est PostgreSQL.

Ci-dessous, une capture d'écran que j'ai prise lors de tests réalisés sur la démo en ligne de cmdBuild, illustrant la représentation des informations d'un logiciel via cmdBuild.

The screenshot shows the cmdBuild application interface. At the top, there's a header 'Fiches Application' with a search bar and navigation icons. Below this is a table listing applications. The table has columns: Product, License, Type, Environment, State, and Hardware. Two applications are listed: 'CMDBuild OpenMANT 3.1' and 'CMDBuild READY2USE 2.0'. The second application is selected, and its details are shown in a sidebar on the left. The details are organized into sections: 'General Data' and 'Administrative Data'. The 'General Data' section includes fields for Code, Product, License, Environment, Hardware, Assignee, Type, and Monitored. The 'Administrative Data' section includes fields for Supplier, Order, and Warranty end date.

Product	License	Type	Environment	State	Hardware
CMDBuild OpenMANT 3.1	Open Source - OS	Web server	Production	Available	23P5L55 - Lenovo System x3250 M5 [silver]
CMDBuild READY2USE 2.0	Open Source - OS	Web server	Production	Available	23P5L55 - Lenovo System x3250 M5 [silver]

General Data	
Code	READY2USE_prod
Product	CMDBuild READY2USE 2.0
License	Open Source - OS
Environment	Production
Hardware	23P5L55 - Lenovo System x3250 M5 [silver]
Assignee	Gray Jade
Type	Web server
Monitored	Monitored

Administrative Data	
Supplier	TT - Tecnoteca
Order	
Warranty end date	

Après avoir présenté ces 3 CMDB et discuter une nouvelle fois des différents atouts de chacun, des nouveaux critères de recherches ont été instaurés :

- La CMDB doit être open source
- La CMDB doit être SaaS
- La CMDB peut finalement être payant
- La CMDB doit intégrer la fonction de représentation de l'écosystème d'une appli sous forme de schéma afin d'apporter une meilleure vision de cette dernière et réagir plus rapidement en cas d'incident ou de demande d'ajout.

Une nouvelle sélection a donc dû être faite, cependant, suites à mes nouvelles recherches, j'ai découvert que les CMDB en ligne payantes proposant une représentation de l'écosystème d'une appli de manière graphique, similaire aux relations maps, qui était la fonction la plus intéressante pour le service Webteam et la plus avantageuse dans l'installation d'une CMDB, proposaient leurs produits en SaaS. De plus, les CMDB, de par leur nature, intègre un système de ticketing, cependant la Webteam possède déjà une application web interne appelée ASAP permettant d'effectuer des demandes de modifications ou de corrections sur une application et de notifier des incidents, il fallait donc que le système de ticketing puisse être désactivable sans que cela nuise au bon fonctionnement de la CMDB.

Suite à une nouvelle réunion dans laquelle nous avons discuté des CMDB payantes, le choix de la CMDB s'est finalement porté sur cmdBuild puisqu'elle intègre de façon claire la représentation graphique de l'écosystème des applications grâce aux relations map.

3.2.5 Installation, premiers tests et problèmes rencontrés

Une fois le choix de la CMDDB effectué, il fallait ensuite l'installer et la déployer en local afin que je puisse faire une démonstration après y avoir intégré les données déjà présentes dans WEP.

Cependant au fur et à mesure de l'avancée de la mission, plusieurs problèmes se sont présentés.

Tout d'abord, le premier problème, qui est assez récurrent mais inévitable dans une entreprise aussi grande que BNP Paribas, est un problème de droit d'accès.

En effet, pour télécharger des fichiers, le site doit être autorisé par le pare-feu de l'entreprise. Cependant, certains liens de téléchargements sont bloqués par ces derniers. Les clés USB sont également bloquées pour le transfert de fichier vers l'ordinateur, on ne peut donc pas télécharger des fichiers à l'extérieur et les récupérer sur les machines internes. Il a donc fallu effectuer une demande auprès du service sécurité, appuyée par mon manager.

Le second problème est que les demandes entre différents services et également parfois au sein du même service, peuvent prendre un temps plus ou moins important, ce qui peut donc Parfois retarder les projets mais qui est inévitable dans une entreprise aussi grande que BNP Paribas puisque les services et leurs membres sont sollicités par d'autres demandes et sont donc occupés sur ces dernières ce qui peut ralentir l'exécution de nouvelles tâches.

Après quelques jours d'attente la demande a été refusée une première fois, le service sécurité souhaitait avoir davantage de détails avant d'autoriser le téléchargement. Après avoir envoyé un rapport plus détaillé de l'application et de son besoin, le service sécurité a finalement une nouvelle fois refusé les droits d'accès au téléchargement du CMDDB et m'a indiqué que je pouvais essayer de contacter le 13, qui est le support interne de BNP Paribas.pour faire autoriser la demande. Cependant le 13 n'a pas pu non plus débloquent les accès au téléchargement de cmdBuild.

Mon poste de travail ne remplissant pas les prérequis hardware pour pouvoir déployer cmdBuild en local, il fallait également m'attribuer un serveur ou une VM afin que je puisse le déployer dessus.

En attendant de trouver une solution au problème, un collègue m'a indiqué que je pouvais faire une demande d'accès à une VM (POC) afin d'installer cmdBuild directement depuis cette dernière pour le tester et faire une première démo interne.

Cette demande a également pris du temps puisque c'est un service différent qui s'en occupe, les réponses prennent parfois plusieurs jours à arriver, ajouter aux rythmes de l'alternance, cela peut donc retarder plus ou moins de manière importante la mission.

Cependant cette VM personnelle sur laquelle je peux réaliser plusieurs tests librement avec des outils internes, n'a pas résolu le problème puisque cette dernière bloquait également certains téléchargement et connexion depuis l'extérieur.

On m'a par la suite attribué un espace sur un serveur afin que je puisse y installer cmdBuild pour le tester et faire une démonstration en important la base de données de WEP. J'ai donc pu commencer l'installation des différents éléments nécessaires au bon fonctionnement de cmdBuild. J'ai dans un premier temps dû régler des problèmes de compatibilités entre les différents éléments nécessaires à cmdBuild en cherchant sur des forums puis j'ai installé tomcat. Cependant les serveurs linux (Redhat) n'avaient pas une version compatible avec les dernières versions du JDK de java, nécessaire au bon fonctionnement de cmdBuild, il ne pouvait pas s'exécuter.

Le projet a finalement été mis en pause le temps de trouver une solution au problème. Après ça j'ai continué à me former sur Symfony grâce à openclassroom et d'autres tutoriels en ligne.

IV - Première réécriture sous Symfony et Angular

4.1 Attribution du sujet

Le second projet qui m'a été attribué est une réécriture sous Symfony et Angular d'une web application interne actuellement écrite sous l'ancien socle technique.

Afin de déterminer quelle application va être réécrite en première, il a fallu obtenir des informations avec le métier de chaque application et effectuer une sélection selon plusieurs critères, les principaux étant :

- L'application ne doit pas être intégré dans une autre application existante dans les 3 ans à venir
- Elle ne doit pas comprendre d'ajout à faire en urgence sur la version actuelle
- Elle ne doit pas avoir un grand nombre de connexions par jour

La première candidate potentielle était LCNET (Lettres chèques NET), cette application permet de commander des chèques chez des imprimeurs.

Finalement, après avoir obtenu un entretien skype avec le métier, ils nous ont appris que l'application ne répondait plus aux normes RGPD (Règlement général sur la protection des données) instaurées en Europe, puisque des informations importantes liées aux données bancaires des clients étaient transmises par mail via l'application et qu'elle allait donc être finalement arrêtée et intégrée ultérieurement dans une autre application.

La seconde candidate la plus susceptible de faire l'objet d'une réécriture était l'application Franchise. Franchise est une application interne de consultation d'information sur les différentes agences en lien avec BNP Paribas utilisées par les conseillers et autres corps de métiers afin de suivre et collaborer avec ces agences.

Après avoir obtenu un rendez-vous skype avec le métier, on nous a donc confirmé que l'application étaient toujours utilisé en interne, qu'elle n'allait pas être intégré à une autre application et qu'elle n'avait pas d'ajout particulier prévu mais qu'éventuellement il faudrait supprimer une fonctionnalité qui n'était plus utilisée.

Finalement le choix ne s'est pas porté sur Franchise puisque les fonctionnalités sont peu nombreuses et qu'il serait plus intéressant de réécrire une autre application.

Suite à différentes réunions avec le métier d'autres applications, l'application sélectionnée a été MNS PFAI.

Il s'agit d'une application de messagerie permettant à ses utilisateurs d'envoyer un message à plus de 33 000 bureaux métiers afin de les notifier des dernières informations les concernant. Lors de la création d'un message, MNS-PFAI l'enregistre dans la base de donnée et envoie également le nouveau message à une API d'un autre service nommé Mon Bureau métier qui va ensuite l'afficher sous forme de notification ainsi que dans une pop-up chez ses utilisateurs, la première fois qu'ils se connectent après la création du message.

4.2 Évolution à venir

A l'heure actuelle, MNS PFAI ne permet pas à ses utilisateurs de choisir les bureaux métiers auxquels les messages sont envoyés. Cette évolution sera donc apportée, dans un premier temps, par un autre développeur, directement sur la version actuelle de l'application puis sera intégrée dans la réécriture sous Symfony/Angular.

4.3 Structure de l'application

La structure de l'application, à la suite de la réécriture, sera la suivante :

Back End :

La partie Symfony, qui fera office d'API, sera constituée de plusieurs controllers.

MessageController.php, qui contient les fonctions de création, consultation et modification de message.

Elle sera constituée des différentes fonctions renvoyant les informations souhaitées par la partie front end.

AdministrationController, qui sera chargé de la création des utilisateurs effectué et l'attribution de leur rôle qui seront réalisés par un administrateur.

LoginController, qui sera chargé de la connexion et déconnexion des utilisateurs de l'application. Lorsqu'il va recevoir les identifiants de l'utilisateur envoyé par la partie front-end, LoginController va utiliser les fonctions du service login.service afin de fournir un token d'identification à la partie front-end pour connecter l'utilisateur.

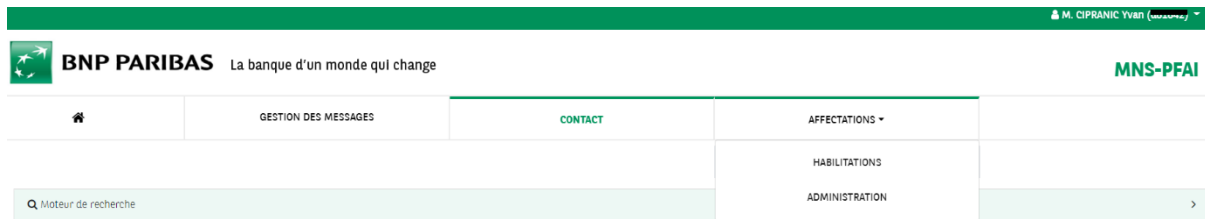
Front end :

La partie Angular contiendra tout d'abord un fichier AppComponent.ts qui sera le composant principale du Front End de l'application. Il est constitué des divers composants nécessaires au bon fonctionnement de l'application.

AppComponent affichera donc, à tour de rôle, grâce à la balise <router-outlet> dans son fichier html correspondant, en fonction de ce que l'utilisateur souhaite faire, les X composants différents en fonction de ce que l'utilisateur souhaite faire.

Les autres composants seront donc :

HeaderComponent.ts, qui servira de header à l'application et contiendra des redirections vers les différents onglets de l'application comme on peut le voir sur la capture d'écran ci-dessous.



FooterComponent.ts, qui servira de footer à l'application et qui, avec le header, seront affichés en permanence par AppComponent puisqu'ils leurs balises sont intégrées en brut dans le code comme on peut le voir sur le morceau de code ci-dessous.

```

app.component.html
src > app > app.component.html > div.bg-color
1 <div class="bg-color">
2   <app-header></app-header>
3   <app-sidebar (collapsedEvent)="receiveCollapsed($event)"></app-sidebar>
4   <section class="container-bnp mb-5 mt-150" [ngClass]="{collapsed: collapsedSideBar}">
5     <router-outlet></router-outlet>
6   </section>
7   <app-footer></app-footer>
8 </div>

```

Liste-Messages-Component.ts, qui sera le composant en charge de l'affichage des messages ainsi que des différentes options qui leur sont liées telles que la suppression et la modification. Lorsqu'un utilisateur va cliquer sur l'onglet "Messages", Angular enverra une requête GET à l'API Symfony afin de récupérer la liste des messages qui seront finalement mis en forme dans le fichier HTML correspondant.

MessageComponent.ts, qui sera le composant en charge de la page de consultation et/ou de modification d'un message en particulier. Ce composant fait appel à plusieurs fonctions du service message.service qui est chargé de contacter l'API Symfony en construisant différentes URL afin de récupérer les bonnes informations comme on peut le voir dans l'exemple ci-dessous.

```

3 export class MessageService {
4     httpOptions = {
5         headers: new HttpHeaders({
6             'Content-Type': 'application/json'
7         })
8     };
9     constructor(private httpClient: HttpClient) { }
10
11     // API: GET /messages/:id
12     public getApplicationById(id: number): Observable<Message> {
13         return this.httpClient.get<Message>(API_URL + '/messages/' + id);
14     }

```

Une fois que l'API aura renvoyé les informations demandés, MessageComponent les mettra en formes.

AdministrationComponent, qui sera le composant en charge de la gestion de l'attribution et modification des rôles des utilisateurs. Il effectuera ses requêtes à l'API en passant par l'URL backend /administration

4.4 Difficultés rencontrées

La première difficulté était l'apprentissage de deux frameworks en même temps. En effet, même si Angular et Symfony apportent leurs lots de facilités dans leurs utilisations, il faut tout de même comprendre les différents concepts qui sont utilisées ainsi que les appliquer correctement.

La seconde difficulté était de répartir chaque partie de l'application sous 2 framework différents. J'étais jusque-là habitué à utiliser un seul framework lorsque je développais, la transmission de données, l'implémentation de la logique ainsi que la navigation au sein de l'application étaient donc différentes puisque tout se passait sur le même framework. Ce fut donc une bonne expérience puisque j'ai découvert de nouvelle manière d'implémenter et de relier le front end et le back end d'une application.

V - Apports personnel :

5.1 Apports techniques

Au cours de cette première année d'alternance j'ai pu découvrir différentes technologies telles que les frameworks Angular et Symfony mais aussi avoir un aperçu des environnements de différentes applications internes, leurs fonctionnements, leurs communications avec d'autres outils ainsi qu'avec différents serveurs et base de données.

La recherche d'information a aussi été très importante, lors de cette phase de recherche il faut savoir comment bien se documenter afin d'avoir un maximum d'informations sur la mise en place de différentes solutions, leurs avantages et inconvénients, surtout lorsque l'on travaille sur un concept des technologies que l'on ne connaissait pas du tout auparavant.

J'ai pu d'une part, mettre en pratique ce que j'ai appris durant mes années de formation en DUT et FIP1, notamment lors de la compréhension de la documentation des applications grâce au module MSI du CNAM ainsi qu'en base de données, d'autre part j'ai pu approfondir mes connaissances en développement web en découvrant et en me formant sur les frameworks Symfony et Angular.

5.2 Apports généraux

Durant cette première année alternance, j'ai pu découvrir la vie en entreprise ainsi qu'au sein d'une équipe Informatique.

J'ai également pu assister à diverses réunions, des réunions consacrés aux développeurs afin de voir l'avancement de chacun sur les projets qui lui ont été assignés, des points secteurs, qui permettent d'avoir une vision globale de l'état du service et connaître les chiffres de l'entreprise ainsi que les nouvelles décisions prises.

J'ai également pris consciences qu'il ne fallait pas rester dans son coin et ne pas hésiter à poser des questions afin de mieux comprendre le fonctionnement de divers éléments au sein de l'entreprise, surtout au niveau technique, qui ne sont parfois pas évident à comprendre (multitude de serveurs, fichiers batch, flux cft ...) lorsque l'on n'a pas l'habitude de les manipuler.