

# Rapport de Stage

Kevin Rivière

January 14, 2015

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Présentation de l'entreprise</b>	<b>3</b>
2.1	Secteur d'activité . . . . .	3
2.2	Fédération Nationale du Crédit Agricole . . . . .	3
2.2.1	Le Crédit Agricole . . . . .	3
2.2.2	Historique . . . . .	4
2.2.3	Organisation . . . . .	5
2.2.4	Fédération Nationale du Crédit Agricole . . . . .	6
2.3	Le service . . . . .	6
2.4	Positionnement du stage dans les travaux de l'entreprise . . . . .	6
<b>3</b>	<b>Travail effectué</b>	<b>7</b>
3.1	Sujet du stage . . . . .	7
3.1.1	Le site creditagricole.info . . . . .	7
3.1.2	JCMS . . . . .	8
3.1.3	Base de données . . . . .	10
3.1.4	Less . . . . .	11
3.1.5	Bootstrap . . . . .	11
3.1.6	Mantis . . . . .	11
3.2	Architecture . . . . .	12
3.2.1	Plateforme d'intégration continu . . . . .	13
3.3	Cahier des charges et planning . . . . .	14
3.4	Compte-rendu d'activité . . . . .	15
3.4.1	Apprentissage des technologies utilisées . . . . .	15
3.4.2	Partage vers les réseaux sociaux . . . . .	15
3.4.3	Brèves . . . . .	16
3.4.4	Adaptation des lanceurs sur mobile et tablette . . . . .	18
3.4.5	Navigation au sein d'un dossier sur un mobile et tablette . . . . .	21
3.4.6	Migration de divers pages . . . . .	22
3.4.7	Tests d'intégration continu . . . . .	22
3.4.8	Fil Twitter . . . . .	24
3.4.9	Redéfinition de la grille de lecture . . . . .	25
3.4.10	Mise en cohérence graphique de la page des brèves . . . . .	26
3.4.11	Cookies . . . . .	28
3.4.12	Mise en production et premiers retours . . . . .	28
3.4.13	Images des contenus sur le site . . . . .	29
3.4.14	Menu sur mobile . . . . .	29
3.4.15	Icône pour mobile . . . . .	31
3.4.16	Migration de Mantis . . . . .	32
3.4.17	Module évènementiel . . . . .	32
3.4.18	Flux RSS sur les écrans plasmas . . . . .	33
3.4.19	Mise en place de Bibliothèques sur l'espace interne . . . . .	34
3.4.20	Nettoyage de l'espace interne en préparation à la migration . . . . .	35
3.5	Interprétation et critique des résultats . . . . .	36
<b>4</b>	<b>Conclusion</b>	<b>37</b>

# 1 Introduction

Dans le cadre de mes études en école d'ingénieur à l'EPITA, un stage devait être effectué en entreprise afin de valider les acquis de la première année du cycle ingénieur. Le stage devait être un stage de développement. Mes recherches se sont basées sur deux grands thèmes différents :

- le développement d'applications en C/C++ ou en Java,
- le développement d'application web en PHP ou en Java.

Ayant fait du développement de petits projets durant la première année d'ingénieur, je voulais vraiment travailler sur un gros projet. De plus, les technologies du web ne me sont pas inconnues, j'avais déjà créé un site web, pour mon compte personnel, avant mon entrée à EPITA.

La principale attente que j'avais envers ce stage était de développer mes connaissances, peu importe le langage. Aussi, l'un des critères importants dans mes recherches était d'avoir un maître de stage disponible et m'aidant sur mes problèmes, c'est pour cela que mes recherches se sont concentrées sur les grandes entreprises, et non pas sur les start-up.

Durant mes recherches, je me suis rendu compte qu'il y avait peu d'offres de stage dans le développement d'application en C/C++ ou en Java, la plupart des offres portaient sur le développement de site web.

Deux offres de stage ont retenu mon attention, la première se déroulait chez Hachette Livre et avait comme sujet le développement d'applications ludo-éducatives mobiles. La seconde se déroulait à la Fédération Nationale du Crédit Agricole et avait comme sujet le développement d'un site internet.

La première offre semblait intéressante car elle m'aurait permis de développer mes connaissances dans le mobile. De plus, les connaissances requises portaient sur le C/C++, le développement orienté objet et l'utilisation de patrons de conception. Cependant, après un premier entretien, on m'a expliqué que le développement se faisait en Javascript ou en C# avec le moteur de rendu *Unity*.

La seconde offre était tout aussi intéressante, car en plus de pouvoir travailler avec de nouvelles technologies du web, il était question de travailler sur un site qui devait être adapté aux mobiles, tablettes et ordinateurs. De plus le stage se positionnait parfaitement dans la continuité du programme suivi à l'école, en effet, l'application web était développée en Java, comme dernier projet de l'année.

J'ai donc choisi la seconde offre, et j'ai effectué mon stage à la Fédération Nationale du Crédit Agricole, au sein du service Publication et Multimédia. Le sujet complet de ce stage est le suivant :

Au sein du département publications et multimédia de la Fédération Nationale du Crédit Agricole, vous intégrez une équipe de 6 personnes en charge des sites internet. Dans le cadre de la migration technique prévue des espaces internet et extranet qui reposent sur la solution JCMS (Jalios), des aménagements fonctionnels et une refonte graphique sont prévus.

Vous serez donc amené à intervenir en tant qu'intégrateur web principalement sur la couche présentation mais également initié aux subtilités de notre CMS (technologies Java J2E).

Vous devrez traduire les nouvelles spécifications d'interface et les implémenter au moyen des technologies Javascript, HTML et CSS, et tenir compte des contraintes liés aux différents supports : Tablettes, Mobile, Web.

Vous serez sous la responsabilité d'un chef de projet et d'un développeur senior.

Mes principales attentes concernant le stage à la Fédération Nationale du Crédit Agricole sont les suivantes :

- développer mes connaissances sur les dernières versions des langages web : HTML 5 et CSS 3, mais aussi en Javascript,
- travailler sur la problématique d'avoir un seul site adapté à une utilisation mobile, tablette et bureau,
- travailler sur l'intégration de Bootstrap au sein d'un site,
- appliquer et approfondir les notions de gestion de projet vu au cours de la première année du cycle ingénieur,
- avoir un maitre de stage à mon écoute et prêt à m'aider à progresser.

## 2 Présentation de l'entreprise

### 2.1 Secteur d'activité

Le Crédit Agricole est un acteur majeur du secteur financier français, et plus particulièrement sur les métiers bancaire et assurances (bancassurance). La Fédération Nationale du Crédit Agricole fait partie du groupe Crédit Agricole, elle dispose donc des mêmes secteurs d'activités.

### 2.2 Fédération Nationale du Crédit Agricole



Figure 1: Logo du Crédit Agricole

La Fédération Nationale du Crédit Agricole est une entité du groupe Crédit Agricole.

Afin de bien comprendre le rôle et le positionnement de la Fédération Nationale du Crédit Agricole, il est nécessaire d'expliquer ce qu'est le Crédit Agricole.

#### 2.2.1 Le Crédit Agricole

Le Crédit Agricole est une banque et assurance coopérative et mutualiste. Une banque coopérative est une société dont le capital est détenu sous forme de parts sociales, par ses sociétaires, qui sont également les utilisateurs des services bancaires. Ce modèle place le service comme une finalité, et est opposé au modèle de banque classique où le capital est détenue par des actionnaires et dont la rentabilité est la finalité.

Le mutualisme c'est une doctrine économique basée sur la mise en commun d'expériences et de moyens en vue d'offrir à ses bénéficiaires les meilleurs services aux meilleurs coûts. Les trois valeurs mutualistes qui font partie intégrante du Crédit Agricole sont :

- la responsabilité,
- la solidarité,
- et la proximité.

Un exemple de mise en pratique de ces valeurs est le soutien par les Caisses régionales, chaque année, de 10000 projets d'intérêt généraux pour un montant global de 25 million d'euros.

Un sociétaire est une personne ayant des parts sociales, non coté en bourse, représentant une partie du capital de la Caisse locale. Il est impliqué dans la gestion de sa Caisse locale en participant à l'Assemblée générale annuel et aux votes dont il dispose d'une voix quelque soit la quantité de ses parts. Depuis le mois de novembre 2014, le Crédit Agricole a franchi la barre des 8 millions de clients sociétaires.

Au niveau des chiffres, le Crédit Agricole dispose de 21 millions de clients particuliers, dont 6 millions en France (fin 2013). C'est aussi la banque de 9 agriculteurs sur 10 en France.

Le Crédit Agricole est composé de 2509 Caisses locales, qui sont regroupées en 39 Caisses régionales, il dispose d'un réseau de plus de 7000 agences et plus de 11000 distributeur et guichets automatiques de billets (DAB/GAB).

### 2.2.2 Historique

- 1885** Création de la première Caisse locale du Crédit Agricole dans le Jura. Le Crédit Agricole est en premier lieu une banque pour les agriculteurs, c'est pour cela que cette Caisse locale permettait aux agriculteurs d'emprunter des fonds afin de développer leurs activités.
- 1894** Jules Méline, ministre de l'Agriculture, crée le Crédit Agricole en faisant voter la loi autorisant la création de Caisses locales dont les sociétaires sont exclusivement des agriculteurs
- 1899** Création des Caisses régionales, grâce à la loi du 31 mars 1899, pour encourager la création de Caisses locales et de les fédérer.
- 1913** Grâce à un développement rapide, il y a des Caisses locales dans tous les départements de France.
- 1926** Création de la Caisse Nationale du Crédit Agricole, banque centrale du groupe. L'organisme devient public et dépend du ministère de l'agriculture.
- 1948** Création de la Fédération Nationale du Crédit Agricole pour permettre aux Caisse régionales de pouvoir entre elles sur l'organisation et la stratégie du groupe face à l'état.
- 1985** Le Crédit Agricole lance sa filiale d'assurance vie : Prédica.
- 1988** Grâce à la loi relative à la mutualisation de la CNCA, la Caisse nationale est affranchie de l'état. Son capital est détenu à 90% par les Caisses régionales et à 10% par le personnel.
- 1990** Création de Pacifica, compagnie d'assurance des biens. En 1993 le Crédit agricole devient le second groupe d'assurance en France.
- 1991** Le Crédit Agricole est autorisé à financer les grandes entreprises et devient une banque universelle. Elle peut donc financer toute clientèle en France et à l'étranger, et peut exercer tous les métiers de la banque et la finance.
- 1998** Le Crédit Agricole rachète Sofinco.
- 2001** La Caisse nationale est cotée en bourse sous le nom de Crédit Agricole SA.
- 2003** Rachat du Crédit lyonnais qui deviendra LCL en 2005.
- 2004** Naissance de Calyon banque d'investissement, qui deviendra plus tard Crédit Agricole Corporate and Investment Bank. Lancement de la filiale immobilière.
- 2006** Développement à l'étranger avec l'acquisition de Cariparma, une banque italienne.
- 2009** Lancement de BforBank, banque en ligne spécialiste de l'épargne.

### 2.2.3 Organisation

Aujourd'hui, le Crédit Agricole est composé de 2509 Caisses locales. Les Caisses locales sont des sociétés civiles coopérative dont les parts sont détenues par les sociétaires. En plus de participer à la gestion de la Caisse régionale dont elle est sociétaire, elle participe aussi au développement local en valorisant des initiatives sur son territoire.

Les Caisses locales sont regroupées en 39 Caisses régionales. Elle sont des banques régionales coopératives, elles ont un rôle central puisqu'elle assurent les fonctions commerciale, bancaire, financière et logistique du Crédit Agricole. Chaque Caisse locale est une entreprise indépendante et toutes partagent les valeurs mutualistes du groupe. Les Caisses régionales sont dirigées par un duo président (élu)/directeur général (cadre dirigeant).

Bien que les caisses régionales soient des entreprises indépendantes, elles font partie du groupe Crédit Agricole et doivent être coordonnées. Ce rôle est tenu par Crédit Agricole SA, qui garantit l'unité financière et veille au bon fonctionnement du réseau Crédit Agricole. De plus Crédit Agricole SA gère aussi les filiales internationales du Crédit Agricole.

Le Crédit Agricole SA est aussi chargé de concevoir les produits et services proposés dans les agences bancaire, afin permettre le développement des Caisses régionales.

Crédit Agricole SA acquiert 25% du capital des Caisses régionales sous forme de CCA (certificat coopératif d'associés), sans droit de vote.

La Fédération Nationale du Crédit Agricole est là pour partager les idées et les initiatives, c'est l'instance de réflexion des Caisses régionales.

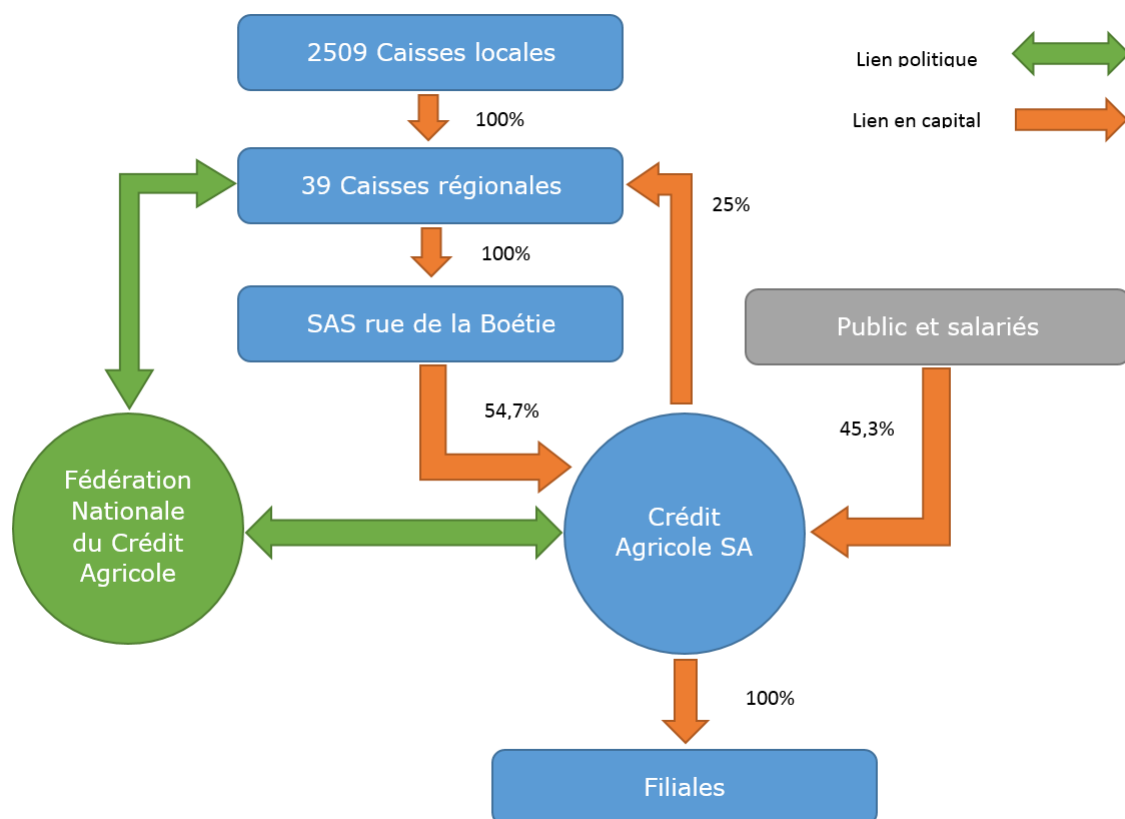


Figure 2: Organisation du Crédit Agricole

## **2.2.4 Fédération Nationale du Crédit Agricole**

La Fédération Nationale du Crédit Agricole est une association du groupe Crédit Agricole. Elle a 3 grandes fonctions au sein du groupe :

- orienter,
- représenter,
- gérer.

La Fédération Nationale du Crédit Agricole est le lieu où les grandes orientations du groupe sont prises. Elle est l'instance de réflexion des Caisses régionales, c'est pour cela qu'elle est qualifiée de "Parlement des Caisses régionales". Elle représente aussi les 39 Caisses régionales face aux pouvoirs publics et à Crédit Agricole SA.

Des réunions mensuelles ont lieu à la Fédération Nationale du Crédit Agricole, où se réunissent les présidents et directeurs généraux des Caisses régionales. Durant ces réunions, ils débattent des grandes orientations du groupe en matière commerciale, financière, technologique, sociale, etc. Ces grandes orientations sont validées par un bureau fédéral de 20 membres, composé de 10 présidents et 10 directeurs généraux. Il est présidé par un président de Caisse régionale et le secrétaire général est un directeur général de Caisse régionale.

La Fédération Nationale du Crédit Agricole ne sert pas uniquement de lieux où se déroulent les réunions mensuelles. Elle dispose d'équipes, 130 employés, qui préparent les dossiers liés aux grandes thématiques qui préoccupent le Crédit Agricole. Elle aide aussi à structurer les propositions des Caisses régionales, afin qu'elles puissent s'exprimer d'une seule voix, quand cela est nécessaire. En plus de cela, la Fédération exerce des missions qui lui sont propres, comme l'animation des clubs d'échanges et d'achats ou la gestion de la convention collective des Caisses régionales.

## **2.3 Le service**

La stage s'est déroulé au sein du service Publication et Multimédia. Équipe de 8 personnes, 2 pôles

## **2.4 Positionnement du stage dans les travaux de l'entreprise**

Le stage se positionne comme support dans les travaux de l'entreprise. Le projet, sujet du stage, avait débuté avant celui-ci et devait être livré à la moitié du stage. L'entreprise recherchait une aide pour assurer la livraison du projet dans les temps et pour assurer le support, au cas où des problèmes apparaîtraient lors de la mise en production.



## 3 Travail effectué

### 3.1 Sujet du stage

Le sujet du stage consistait à participer à une refonte du site internet de la Fédération Nationale du Crédit Agricole : [www.creditagricole.info](http://www.creditagricole.info), en vue de sa mise en production et corriger les éventuels bogues ensuite.

Le site est basé sur un système de gestion de contenu (SCG ou CMS en anglais), JaliOS Content Management System (JCMS) développé par JaliOS.

Le site utilisait la version 6 de JCMS, sortie en 2009, et le but du projet était de migrer le site sur la version 9 de JCMS, sortie en octobre 2014. En plus de ce changement de version, une refonte graphique était aussi prévue.

Les technologies utilisées pour mener à bien ce projet sont :

- Java Entreprise Edition 1.6,
- Javascript/JQuery,
- Bootstrap 3.0.1,
- Less/CSS,
- HTML 5.

#### 3.1.1 Le site [creditagricole.info](http://creditagricole.info)

Le site [creditagricole.info](http://creditagricole.info) fut lancé en 2009, il fait suite à [boetianet](http://boetianet), précédent site exclusivement interne. Le site est divisé en deux parties :

- une partie internet ouvert à tout le monde,
- et l'espace interne, réservé aux employés du Crédit Agricole.

Le site internet sert de vitrines aux 39 Caisses régionales en regroupant toutes leurs actions sur le territoire sous forme d'actualités et de vidéos. Aussi ce site permet aussi à la Fédération Nationale du Crédit Agricole d'avoir une présence sur internet. En effet, avant 2009 il était très difficile de trouver des informations sur cette instance du Crédit Agricole.

Le site internet est un donc un site éditorial, une grande partie des contenus du site sont issue des actualités des Caisses régionales, l'autre partie provient de la Fédération Nationale du Crédit Agricole qui propose des sujets de réflexion. Afin de relater toutes les informations, le site dispose de plusieurs types de contenus :

- article,
- dossier (un ensemble d'articles),
- brèves,
- vidéo,
- évènement,
- fiche initiative,
- communiqué de presse.

Généralement, les articles, évènements, fiches initiatives et certaines vidéos relatent l'actualité des Caisses régionales. Les dossiers et une partie des vidéos sont eux issus de la FNCA et apportent des sujets de réflexion.

L'espace interne sert d'espace d'échange au sein du groupe Crédit Agricole. Seuls les employés du Crédit Agricole peuvent s'inscrire, ceux des autres banques du groupe peuvent aussi s'inscrire mais ils auront un accès limité à l'espace interne.

L'espace interne est composé de clubs, ils servent d'espace de discussion et d'échange. Pour cela, il est possible de publier des brèves (court article) et des documents au sein d'un club. De plus, chaque club dispose d'un forum où les membres peuvent discuter librement. Il y a deux types de club :

- les club à accès libre, tout membre du site peut s'inscrire, ils ont généralement pour sujet un thème qui est commun à tous les employés du Crédit Agricole,
- les club privé, l'inscription se fait par une demande à l'administrateur du club, ils ont pour sujet des thèmes sensibles, par exemple le club des dirigeants.

La base de ce site est donc la rédaction de contenu et la collaboration c'est pour cela que l'entreprise a choisi un système de gestion de contenu. L'entreprise a décidé d'utiliser un SGC français développé par Jalios : JCMS.

### 3.1.2 JCMS

Comme tout SGC, JCMS permet de faire de la gestion de contenu avec gestion des droits. Publier des contenus sur un site web devient alors à la portée de tous car il suffit de déposer son contenu sur le site et JCMS s'occupe de générer une page HTML. Un SGC est généralement découpé en deux parties :

- la première partie, accessible à tout le monde, permet d'afficher les contenus publiés, elle est appelé front office,
- l'autre partie, privée, permet d'ajouter des contenus et de paramétrer / structurer le site, c'est le back office.

Au point de vue technique, JCMS est composé d'un ensemble de portail, qui correspond à une page. Un portail c'est un assemblage de portlet, caractérisé par un gabarit de page portail, associé à l'arborescence du site. Un portail est générique et n'est pas forcément associé à une seule page.

Un portlet est un élément du portail, elles servent à générer différents composants d'une page. Il existe différents types de portlets :

- portlets de construction de page,
- portlets de contenu,
- portlets de navigation,
- portlets fonctionnelles.

L'assemblage des différents types de portlets permet de construire des gabarits de pages portail. De plus, une portlet peut être, elle-même, composée de portlets.

Par défaut, JCMS contient un certain nombre de type. Un type c'est la structure d'un contenu ou d'une portlet. C'est une organisation technique d'un besoin fonctionnel, un type

est constitué d'un ensemble de champs (texte, lien, image, etc), par exemple un article est constitué d'un champ texte simple pour le titre, un champ texte riche pour le contenu et un champ image pour l'illustration.

Un type de contenu ou une portlet une fois instanciés peuvent être publiés en front office par un contributeur. On parle alors d'une publication. À cette instance, il est possible d'associer un gabarit de présentation, ce qui devient un contenu.

Il existe 2 types de gabarits : les gabarits d'affichage du contenu en pleine page dans le front office en fonction du type de contenu et les gabarits de requêtes pour afficher les résultats d'une recherche. Ces gabarits sont liés à un type et sont personnalisables, il est possible d'avoir plusieurs gabarits pour un type. Il y a différents types de gabarits :

- Gabarit de présentation de contenu : il définit les normes d'affichage d'un contenu, par exemple comment afficher un article,
- Gabarit de portlet : ils sont développés par Jalios ou par un développeur et permettent de définir l'affichage au sein de la portlet,
- Gabarit de la page portail : il définit l'organisation des différentes portlets à l'intérieur d'une portlet de type portail.

Les contenus et les portails peuvent être organisés de façon hiérarchique au sein de JCMS grâce aux catégories. Elles sont utilisées pour la navigation au sein du projet creditagricole.info.

JCMS est organisé en espaces de travail permettant notamment de donner à certains utilisateurs des droits de gestion (contribution, modification,...) sur les publications (contenu et portlet) et les catégories.

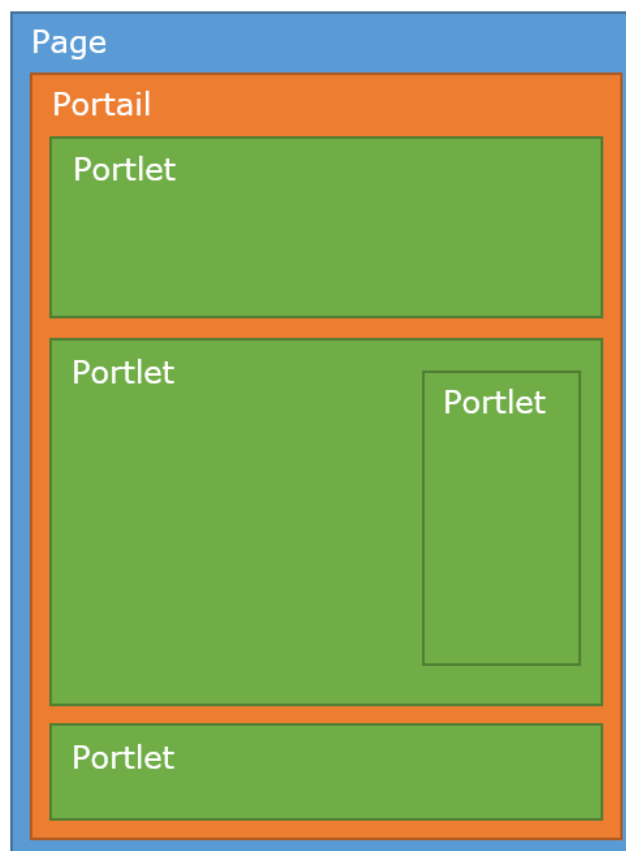


Figure 3: Structure d'une page

Comme vu précédemment, le site [creditagricole.info](http://creditagricole.info) propose un certain nombre de types de contenus qui ne sont pas forcément présents par défaut au sein de JCMS. Pour personnaliser ou ajouter des types de contenu ou des portlets, JCMS propose un développement par module, qui seront ajoutés à l'application au démarrage du serveur. Les modules contiennent les types de contenu, les gabarits d'affichage, les habillages personnalisés, mais aussi les images, les styles CSS/Less, des JavaServer Pages (jsp) qui permettent d'exécuter du java au sein de contenu HTML, etc. Le module doit avoir une structure précise afin de permettre l'intégration à l'application. Dans une but d'assurer une compréhension des actions d'un module, celui-ci doit disposer d'un fichier XML qui fera le liens entre l'application et les fichiers du module, c'est lui qui servira notamment à préciser à JCMS si un fichier est un gabarit d'affichage, un habillage, ... et à quel type il est relié.

Ce développement par module permet à tous les sites utilisant JCMS d'avoir une même base, mais aussi de partager leurs modules sur le site communautaire de Jalios. De là, les autres développeurs peuvent pendre les modules pour ajouter des fonctionnalités à leur site facilement.

Le site internet [creditagricole.info](http://creditagricole.info) est organisé en plusieurs modules :

- `FNCANewsletterPlugin` : ce module contient les types de contenu, les gabarits d'affichage, les images et les styles utilisés pour la visualisation et l'envoi de lettre d'information,
- `FNCAWebCartographiePlugin` : ce module contient toutes les JSP relatives à la configuration et à l'affichage des cartes de Google Maps sur le site,
- `FNCAWebGrIDSurePlugin` : ce module gère toute la partie authentification en utilisant un système de motif comme mot de passe, et s'appuie sur un webservice.
- `FNCAWebModelPlugin` : ce module est l'un des plus important car il contient tout les types de contenu, ainsi que leur gabarit d'affichage, nécessaire au bon fonctionnement du site,
- `FNCAWebPlugin` : ce module est le plus important, il contient tout ce qui est commun à toute les pages du site comme les styles, le menu, le logo, mais aussi des pages spécifique comme la page de connexion, la page d'accueil, etc.,
- `FNCAWebPortletPlugin` : ce module contient toutes les portlets du site,
- `FNCAXitiPlugin` : ce module contient le code à inclure pour avoir des statiques sur les visite du site [creditagricole.info](http://creditagricole.info).

### 3.1.3 Base de données

En informatique et plus particulièrement sur le web, la persistance des données est une fonction importante. Sur le web les données sont stockées dans des bases de données, qui sont gérées par des Système de Gestion de Base de Données (SGBD). Il existe deux types de SGBD : le SQL et le NoSQL.

Le SGBD le plus répandu est le SQL, les données sont stockées sous forme de tableau dans des "Tables". Le SQL est un SGBD relationnel, il est donc possible de lier des tables, afin d'éviter une redondance des données dans différentes tables, et permet une meilleure gestion des données.

Le NoSQL, qui veut dire *Not only SQL* en anglais, est un SGBD non relationnel. Il y a plusieurs familles de NoSQL qui ont différentes bases, mais les plus connues se basent sur un système de clé/valeur.

JCMS utilise les deux types de SGBD, le SQL est utilisé pour stocker certaines configuration ainsi que quelques données, mais les différentes instances de types et de portlets sont stockées en NoSQL sous forme d'XML. Ce dernier, nommé "store" au sein de JCMS, est composé d'un empilement d'opérations (créer, mettre à jour, supprimer), ce système permet de garder en mémoire tous les états de la base de données, et facilite le retour à un état précédent. Le store a facilité la migration, car un point de repère a été placé à la fin de celui-ci au début du développement, de façon à fusionner uniquement les opérations qui se trouvent après la marque.

#### **3.1.4 Less**

Le less est une extension du langage CSS. Le CSS permet de définir des styles pour les éléments d'une page HTML. Le CSS est un langage statique, les notions de variables, fonctions et boucles n'existent pas en CSS. Le less permet d'apporter ces notions au CSS, cependant, les navigateurs internet ne savent interpréter que du CSS. Le less est donc compilé afin de produire du CSS, cette compilation se produit au démarrage du serveur ou par un administrateur, mais elle n'a pas lieu à chaque chargement d'une page.

#### **3.1.5 Bootstrap**

Bootstrap est un ensemble d'outils, développé par Twitter, utile lors de développement de site web. Il est composé d'un ensemble de code HTML, CSS et Javascript qui permettent de mettre en page un site web facilement, et d'y ajouter quelques fonctionnalités dynamiques comme un carrousel ou un système d'onglet. Toutes les feuilles de styles sont faites en Less, et il est possible de les surcharger pour plus de personnalisation. Pour les fonctionnalités Javascript, Bootstrap utilise la bibliothèque JQuery, qui n'est pas compris dans ce dernier, il faut donc l'inclure à part.

#### **3.1.6 Mantis**

Pour le suivi des faits techniques, l'entreprise utilise l'outil OpenSource Mantis. Cet outil se présente sous la forme d'un site web hébergé au sein de l'entreprise. Seule une partie du service est inscrit. Il permet de reporter des bogues ou des améliorations, sous forme de ticket. Ce ticket peut être ensuite attribué à une personne et dispose de plusieurs états. L'outil permet aussi d'ajouter des commentaires et des images aux tickets afin de les rendre plus expressifs.

Quand un bogue ou une amélioration était résolu, en plus de changer l'état du ticket, il fallait préciser en commentaire les fichiers qui avaient été modifiés afin d'avoir une trace. De plus les commits devaient comporter le numéro et le titre du ticket qu'il résolvait.

## 3.2 Architecture

L'entreprise possède ses propres serveurs pour héberger le site creditagricole.info. Elle dispose de deux salles contenant en tout 9 serveurs :

- 3 serveurs de test, dont 1 serveur SQL,
- 6 serveurs de production, dont 2 serveurs SQL.

En plus des serveurs, des baies sont disponibles pour sauvegarder tous les documents, images, vidéos, etc. qui sont mis en ligne sur le site. L'entreprise utilise pour cela des NAS, cela leur permet de bien séparer le site des documents mis en ligne par les utilisateurs. De plus, pour plus de sécurité et pour éviter de mélanger les documents issus du site internet et les documents de la partie interne, ils disposent de deux baies différentes pour chaque partie du site.

Tous les serveurs (test et production) sont sous OS Windows Server 2008. Sur chacun d'eux sont installés les serveurs web Apache et applicatifs Tomcat, qui sont nécessaires au fonctionnement du site.

Les deux serveurs de test servent à héberger une version de test de chaque partie du site. La version de test du site internet est hébergée sur le serveur de test 2 (TST2), et l'espace interne est hébergé sur le serveur de test 3 (TST3).

Au niveau de la production, chaque partie du site (internet et espace interne) dispose de deux serveurs, un principal et un de secours. Le serveur de secours est appelé réplicat car il réplique toutes les modifications effectuées sur le serveur principal. En cas de problème avec le serveur principal, la bascule sur le réplicat se fait automatiquement.

La connexion par bureau virtuel est restreint aux serveurs de tests pour le service Publication et Multimédia. Pour déposer des fichiers ou effectuer des actions sur les serveurs de production, par soucis de sécurité, il faut passer obligatoirement par la DSI, aucune connexion par bureau virtuel n'est autorisée pour les personnes n'appartenant pas à la DSI.

En plus des serveurs de test, l'entreprise dispose d'un serveur sous OS Linux contenant une plateforme d'intégration continue JAlios Delivery Engine (JADE).

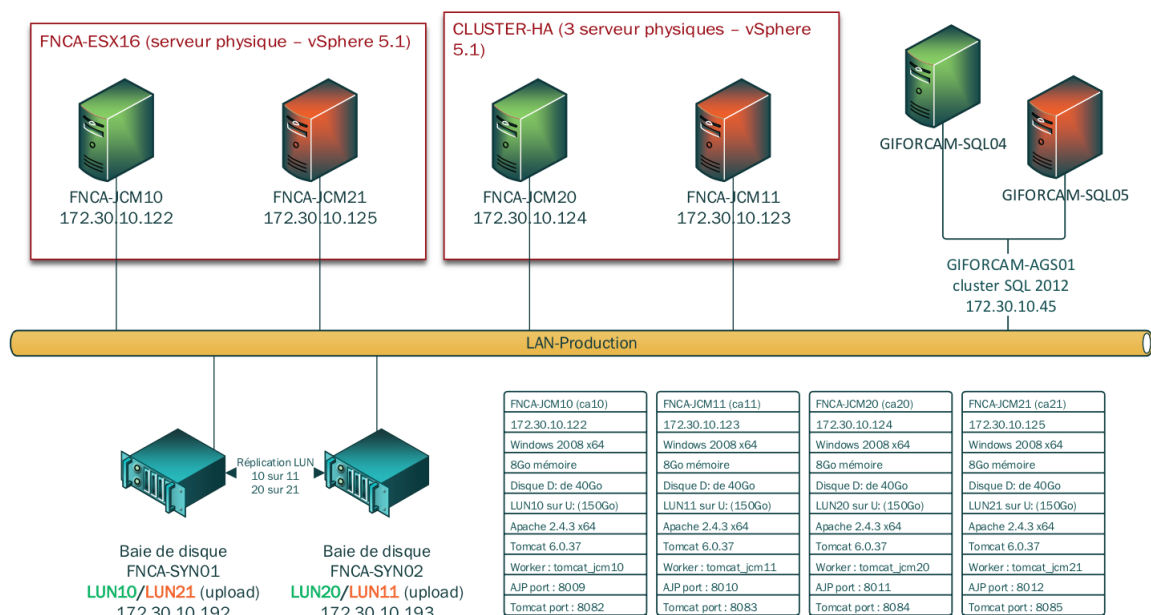


Figure 4: Architecture d'un environnement de production

### 3.2.1 Plateforme d'intégration continu

Pour les test d'intégration continue, l'entreprise a mis en place un serveur contenant une plateforme d'intégration continue, JADE, développée par Jalios. JADE permet de produire automatiquement des livrables, détecter les problèmes rapidement et centraliser les informations. JADE est livré sous forme de machine virtuelle par Jalios et les composants suivants y sont pré-installé :

- Jenkins, un outil Open Source d'intégration continue écrit en Java
- Subversion (SVN),
- des scripts permettant de produire des modules et des applications.

JADE exécute les scripts de tests et compilations périodiquement sur chaque module. Pour le projet `creditagricole.info`, un module est exécuté toute les demi-heures. Ces scripts permettent de générer des modules compilés, qui pourront être intégré au sein d'une application JCMS. Une fois tous les modules compilé, JADE génère une application JCMS avec les modules directement intégré, et ceci pour facilité le déploiement.

Les scripts fournit au sein de JADE sont composé des actions suivantes :

- téléchargement de la dernière version du module grâce à SVN,
- téléchargement des fichiers de configuration et du store de référence,
- copie des modules de Jalios et de l'application JCMS vierge,
- décompression de JCMS et déploiement des modules,
- préparation du démarrage d'une instance de Tomcat sur un port libre, déterminé automatiquement,
- démarrage de JCMS via Tomcat afin de vérifier que les classes du module sont bien compilées
- deuxième redémarrage de JCMS pour effectuer les tests unitaires,
- construction du livrable.

### 3.3 Cahier des charges et planning

Pour ce projet, l'entreprise ne disposait pas de cahier des charges, mais disposait d'un cahier fonctionnel réalisé par une agence web (BETC). En effet, le but du projet était une migration de la partie internet du site de JCMS 6 à JCMS 9, tout en simplifiant le site et en mettant son interface à jour. Les fonctionnalités disponibles sur l'ancienne version du site devaient toujours être disponibles sur la nouvelle version.

Bien qu'il n'y avait pas de cahier des charges, le but et les différents points furent définis lors de réunions au début du projet fin avril 2014. L'entreprise a fait appel à une agence web, BETC, pour produire un cahier fonctionnel. Le fait de passer par une agence permet d'avoir une vision externe sur le projet, et aussi d'avoir une charte graphique en lien avec les tendances du moment sur le web.

Après avoir eu lieu une période de réunions entre l'agence web et l'entreprise, durant laquelle il y eut un échange entre les maquettes fonctionnelles d'abord, puis les maquettes graphiques. Cette période prit fin avec la livraison du cahier fonctionnel le 16 juillet 2014. Ce cahier était composé de maquettes fonctionnelles du site et d'une charte graphique. Cependant, celui-ci n'était pas complet, seuls deux types de contenu avaient une maquette : le type article et le type dossier. Pour les autres pages, c'est l'entreprise qui s'est chargée de les faire.

La méthodologie agile fut utilisée lors de ce projet, ce qui permet de développer et valider des fonctionnalités chaque semaine. Cela s'est traduit par des réunions où les objectifs et différents points de migration d'une fonctionnalité étaient abordés, puis le développement (migration) de cette fonctionnalité pouvait commencer. Une fois le développement fini, les objectifs étaient vérifiés afin de valider la migration de la fonctionnalité.

Cette méthode avait un grand avantage, la réactivité, que ce soit au niveau de modification, correction ou création de fonctionnalités. Mais elle avait aussi des désavantages, certains points ne furent fixés qu'à quelques semaines de la fin du projet.

Aussi, la date de fin du projet a changé au cours de celui-ci, il était initialement prévu de se terminer début octobre. L'une des raisons de changement était le changement de version de JCMS, le site devait être développé avec JCMS 8, parce que la version 9 de JCMS n'était pas encore sortie au début du projet. De plus, la date de sortie de JCMS 9, initialement prévue à fin juin 2014, fut repoussée à début octobre 2014. Le développement du site a commencé avec la version 8 de JCMS, et ce n'est qu'une fois la date de sortie de JCMS 9 fixée que l'entreprise a effectué le changement. Le site a donc subi deux migrations, une de la version 6 à la version 8 de JCMS et une de la version 8 à la version 9 de JCMS. Peu après ce changement début septembre, la date de fin du projet fut fixée à début novembre 2014.



## 3.4 Compte-rendu d'activité

### 3.4.1 Apprentissage des technologies utilisées

Le stage a débuté par une période de formation aux technologies utilisées par l'entreprise. JCMS étant un SGC utilisé en milieu professionnel, il était difficile d'avoir des connaissances sur ce sujet en dehors de ce milieu. Cette période de formation fut composée de :

- lecture de documentation sur JCMS afin de comprendre son fonctionnement global et celui du développement en modules,
- tests sur une application JCMS vierge,
- apprentissage du less.

Pour le développement l'entreprise utilise l'IDE Eclipse, il est conseillé par Jalios qui fournit un outil pour le développement modulaire. En effet, chaque module est considéré comme un projet, et sont au même niveau que l'application web JCMS. Cependant, JCMS ne peut pas récupérer les modules en dehors de l'application, et la seule manière de les inclure est de passer par l'interface web une fois l'application lancée. Pour éviter cette manipulation à chaque modification d'un module, Jalios propose un outil qui permet de synchroniser les fichiers d'un module entre sa version au sein de JCMS et sa version en tant que projet.

La période de formation devait être la plus courte possible afin de pouvoir travailler le plus vite sur le site qui devait être en production deux mois plus tard.

### 3.4.2 Partage vers les réseaux sociaux

La première tâche confiée fut l'intégration de boutons de partage vers les réseaux sociaux. Ceux-ci existaient déjà dans la version précédente du site, et permettaient le partage sur Facebook ainsi que sur Twitter. Mais il manquait des balises d'identification de contenus.

Lorsque que l'on partage un contenu sur Facebook, celui-ci visite la page que l'on souhaite partager à la recherche de balises contenant des informations sur la page. À partir des informations recueillies il génère une fiche de présentation qui sera affichée sur le post de partage, ce qui est beaucoup plus parlant qu'un lien.

Les balises se présentent sous la forme de balise *meta* qui sont situées entre les balises *head* de la page. Elles permettent de donner plusieurs informations à Facebook :

- titre (balise *og:title*)
- description (balise *og:description*)
- url (*og:url*)
- image (*og:image*)

Si une de ces balises est manquante, Facebook ira chercher dans la page les informations manquantes. Ces informations ne sont pas prises au hasard, par exemple si la balise *og:title* qui précise un titre est manquante, Facebook prendra le titre de la page, qui se trouve dans la balise *title* au sein des balises *head*. Il est préférable de préciser ces informations afin de contrôler ce que Facebook va afficher sur le post de l'utilisateur et éviter de donner une mauvaise image à un potentiel visiteur avant qu'il n'arrive sur le site.

Les boutons de partage visibles sur les pages ont dû être mis à jour, pour se conformer aux dernières API's de Facebook et Twitter. Avant même de commencer à développer, il fallait

se documenter sur la manière d'intégrer ces boutons au sein du site. Pour cela Facebook et Twitter ont mis en place des pages de documentation à destination des développeurs.

Après les recherches et au début du développement, une autre solution fut proposée afin de faciliter l'intégration et d'avoir des statistiques. Cette solution était un service, nommé AddThis, qui permet une intégration des boutons de partage sur un site internet ou une application mobile. Cette solution est utilisée sur de nombreux site web car elle présente plusieurs avantages :

- intégration sur le site facilitée, car celui-ci se fait par l'ajout d'un lien vers un script ainsi qu'une balise avec un identifiant spécial (*addthis-sharing-toolbox*) là où l'on veut que les boutons apparaissent,
- une gestion des réseaux sociaux, sur lesquels seront partagés les pages, simple d'accès car tout se fait à l'aide d'une interface où il suffit de faire glisser le réseau social voulu de la liste de tous les réseaux sociaux disponibles vers celle choisie pour apparaître sur le site,
- des statistiques qui donnent le nombre de partage par jour ainsi que le nombre de clics sur les liens partagés.

Avant d'intégrer cette solution au sein du site, une vérification des droits d'utilisation a été faite. La solution propose deux offres, une payante et une gratuite. L'offre utilisée par l'entreprise serait l'offre gratuite, il fallait donc vérifier si son cadre d'utilisation incluait un usage professionnel. Cela n'était pas précisé, la solution fut donc intégrée.

### 3.4.3 Brèves

Le travail proposé était de refaire la partie concernant les brèves. Une brève est un article court ou un événement, sans image où l'information tient en un ou deux paragraphes. Du point de vue de JCMS, cela ne correspond pas à un type de contenus mais à un contenu article ou événement catégorisé en tant que brève.

Sur l'ancien site, les brèves étaient présentées en deux pages : une page lanceur et une page où la brève était entièrement affichée. Une page lanceur est une page sur laquelle on trouve une liste de contenus avec le plus souvent une pagination. Pour les brèves, le lanceur consistait à afficher le titre de la brève, le début de la brève ainsi que des informations complémentaires. Le problème de ce système fut qu'il nécessitait un clic et un changement de page avant de pouvoir lire la brève.

Ce qui fut proposé comme amélioration était d'avoir toutes les brèves sous forme de liste. Chaque brève est représentée par son titre et peut être ouverte en cliquant sur le celui-ci. Elle se déploie et laisse apparaître le corps de celle-ci. L'avantage de cette solution est de pouvoir lire entièrement la brève à partir de la page lanceur et du coup en pouvoir lire plusieurs à la suite sans changer de page. Pour ce faire l'utilisation de Bootstrap a été conseillée car celui-ci disposait déjà de classes permettant ce comportement.

Au cours du développement, un problème a été trouvé à cette présentation : elle ne permet pas de partager les brèves sur les réseaux sociaux. En effet, comme vu précédemment, pour être partagé un contenu doit pouvoir donner des informations au travers de balises, cependant ces balises sont uniques par page, par exemple on ne peut pas avoir plusieurs titres pour une même page. Or le but de cette présentation est d'avoir plusieurs brèves sur la même page.

Après un premier rendu, un problème de lisibilité est apparu sur les serveurs de test. Ce premier rendu utilisait les classes par défaut de Bootstrap sans surcharge. Elles présentaient chaque brève dans un rectangle délimité par une bordure grise, à l'intérieur le titre et le corps de la brève étaient séparés par une bordure. Le titre des brèves disposait d'une couleur de fond.

Sur un faible nombre de brève (l'application en locale disposait de 5 ou 6 brèves) ce problème de lisibilité n'était pas important, mais lorsqu'il y a une vingtaine de brèves affichées, il est difficile de savoir que le corps de la brève est relié au titre précédent. Pour palier à ce problème, il a été proposé de retirer la bordure entre la brève et le titre lorsque celle-ci était ouverte ainsi que la couleur en fond.

Le second rendu, bien que plus clair dans la lecture, n'était pas tout à fait satisfaisant. Les brèves étaient affichées en liste, et seul leur titre les différenciait, mais celui-ci n'était pas suffisant car il manquait une information importante, la date de publication. Il peut y avoir plusieurs brèves par jour, et ceci est facilité par leur forme courte. Sans date il était difficile de se repérer dans le flot de brèves. Elle a donc été rajoutée au même niveau que le titre mais alignée sur la droite en gris.

Une amélioration fut aussi apportée au niveau de la navigation à l'intérieur du lanceur. Jusqu'à maintenant, l'ouverture d'une brève se faisait par un clic sur le titre de celle-ci, or pour faciliter la navigation il a été proposé d'étendre la possibilité de clic au bandeau contenant le titre et la date. Cette solution fut validée et mise en place.

Cette présentation plus claire fut réutilisée pour la page des communiqués de presse. Contrairement aux brèves qui sont courtes, les communiqués de presse sont longs, mais cette présentation reste acceptable. Pour le lanceur des brèves il y en a une vingtaine de contenus par page, alors que pour le lanceur des communiqués de presse il y en a cinq. Cela permet d'avoir plusieurs communiqués de presse sur la même page avec une facilité de navigation.

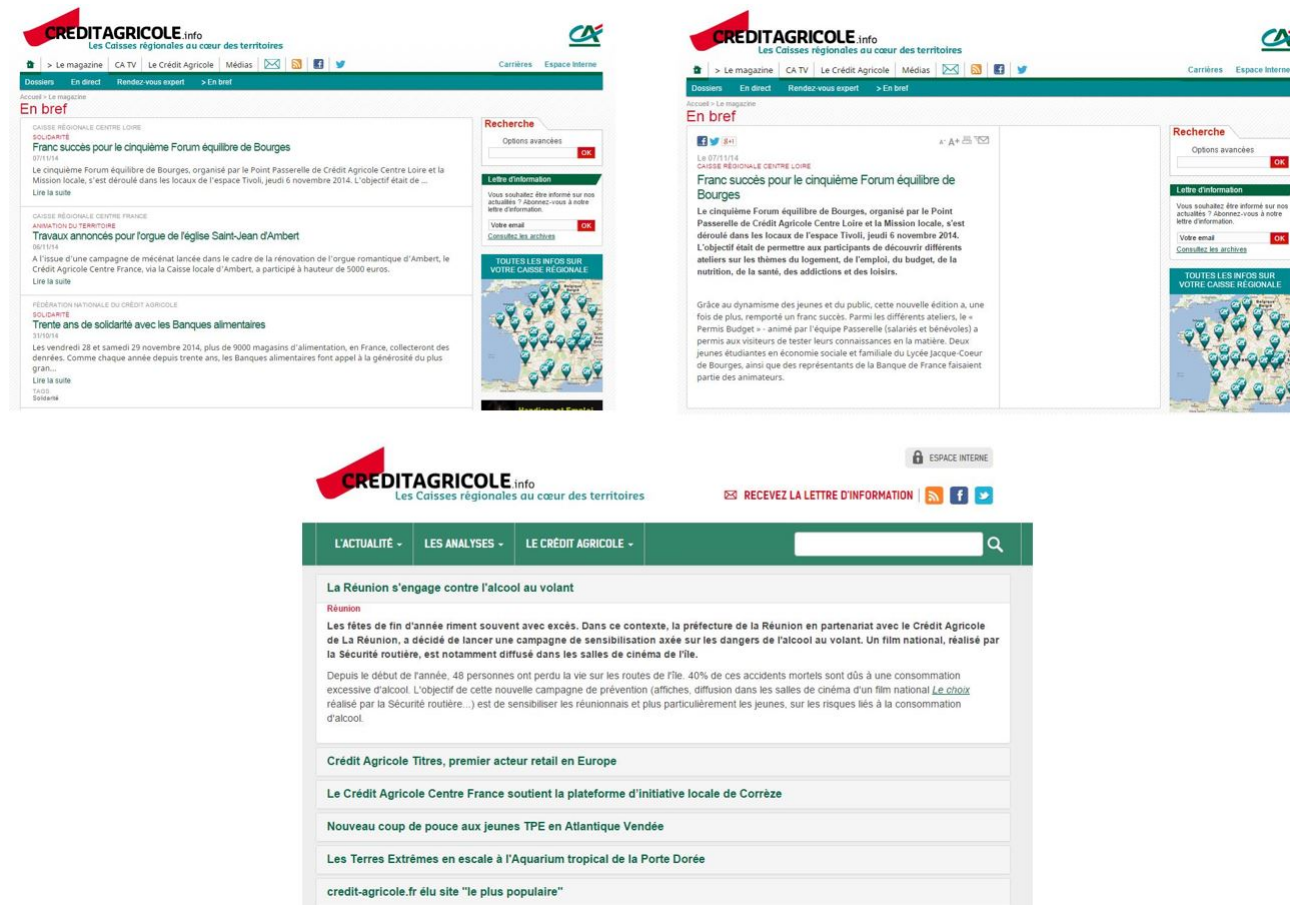


Figure 5: Les brèves avant avec le lanceur et page de contenu et après

La refonte de la partie brève comprenait aussi un bandeau sur la page d'accueil. Il permet d'afficher le titre de certaines brèves et de renvoyer vers celle-ci. Ces brèves sont choisies, cela

dépend d'une certaine catégorisation. Le bandeau affiche un titre après l'autre, de ce fait, il n'y a jamais deux brèves en même temps. Ce bandeau existait sur l'ancien site, et l'on pouvait mettre pause au défilement des titres.

Sur le nouveau site il a été décidé, dans un soucis d'épuration et de simplification, d'enlever les contrôles (jouer, pause) sur le défilement, mais de garder le défilement. Au niveau du développement, afin de ne pas garder de vieux code, il a été décidé de repartir de zéro. L'animation du titre qui disparaît pour laisser sa place au suivant nécessite de modifier le code HTML de la page après sa génération par le serveur. La modification d'un code HTML après sa génération est possible grâce à une interface, le DOM, qui est implémenté au sein du Javascript. De plus afin de faciliter l'utilisation du Javascript, la bibliothèque jQuery a été utilisée.

La méthode mise en place pour permettre cette animation consistait à :

- charger tous les titres des brèves à afficher dans le code HTML, ils sont tous cachés sauf la première, avec une propriété CSS, et ils disposent d'un identifiant unique,
- le nombre de brèves à afficher est sauvegardé,
- une fois la page chargée par le navigateur, il exécute la fonction en Javascript qui itère en boucle sur tous les titres, avec un intervalle de 5 secondes.

Aussi, il a été demandé une option permettant de ne pas afficher les événements passés. Cette option serait placée dans le back office, dans les options du module web. Pour se faire il suffisait une fois les résultats de la requête connus de tester leur date de validité. Ce test entraîne une réduction du nombre de brèves affichées car il se fait en aval de la requête, mais ne pouvait pas avoir lieu en même temps que celle-ci. En effet, les requêtes en JCMS sont définies par l'utilisateur au niveau d'une interface et ce qui est développé est l'affichage de ces résultats.



Figure 6: Bandeau sur la page d'accueil

#### 3.4.4 Adaptation des lanceurs sur mobile et tablette

L'un des objectifs principaux du projet était d'avoir un site web adaptatif, c'est à dire que l'interface doit s'adapter automatiquement à la taille de l'écran sur lequel il est visualisé. Le but d'avoir un site web adaptatif est d'éviter la multiplication des pages selon le format d'affichage, mais d'avoir une seule page. Par exemple, lorsqu'un article est affiché sur un ordinateur ou sur un mobile, c'est la même page qui doit être appelée. Cependant l'interface n'est pas la même, il se peut que des éléments soient modifiés, ajoutés ou supprimés.

Ces modifications sont le plus généralement décrites en CSS à l'aide des *media queries* introduites par le CSS3. Pour se faire il suffit d'avoir des propriétés de style différentes d'une même classe ou d'un même identifiant, et de préciser à l'aide des *media queries* celles qui doivent être utilisées selon la taille de l'écran. D'autres modifications plus spécifiques peuvent être apportées au niveau du serveur. Cela permet de sélectionner la partie de code qui va être exécutée selon l'environnement.

L'utilisation de Bootstrap permet de faciliter la création de site internet adaptatif, car celui-ci contient des classes utilisant les *media queries*. De plus, les classes sont basées sur un affichage en grille, la taille d'un élément ne dépend pas d'une taille donnée en pixel mais en pourcentage. La grille permet d'afficher 12 éléments par ligne. Il y a donc 12 choix de largeur

pour un élément. Par exemple, un élément ayant une largeur de taille 4 et un autre de taille 8 seront placés côte à côte, alors que deux éléments ayant une largeur de taille 12 seront l'un au dessous de l'autre. La force de Bootstrap réside dans le fait que l'on peut préciser une largeur différente selon la taille de l'écran. Par défaut Bootstrap propose 4 résolutions d'écran :

- mobile (largeur strictement inférieure à 768 pixels),
- tablette (largeur supérieur ou égale à 768 pixels),
- écran médium (largeur supérieur ou égale à 992 pixels),
- écran large (largeur supérieur ou égale à 1200 pixels).

Les deux dernières tailles correspondent aux ordinateurs (portable et fixe). Pour les mobiles, bien que leur résolution soit plus importante que 768 pixels, en réalité les éléments affichés à l'écran sont deux à trois fois plus définis et non pas deux à trois fois plus grands. De ce fait un mobile avec une résolution de 1920 par 1080 pixels affichera les éléments en 414 par 736 pixels avec une définition de l'ordre de 3. Mais un problème se pose en ce qui concerne les tablettes, en mode portrait elles sont reconnues comme des tablettes par Bootstrap, cependant en mode paysage elles sont reconnues comme des écrans médium. Un site peut donc être vu de deux manières différentes selon l'orientation dans laquelle est disposée la tablette.

L'ancien site [creditagricole.info](http://creditagricole.info) était en deux versions : la version mobile, accessible à l'adresse [m.creditagricole.info](http://m.creditagricole.info), et la version bureau. Sur le nouveau site, les deux versions étaient maintenues, mais accessibles à la même adresse. Le départage se faisant grâce au caractère adaptatif du site. Cependant le découpage était fait de manière à ce que le site mobile soit utilisé sur les mobiles et tablette. Donc selon l'orientation de la tablette on avait un site mobile ou un site bureau. Le travail proposé était d'adapter le site bureau pour une tablette, afin que la version mobile soit réservée aux mobiles.

Ce travail nécessitait de revoir une partie du site, la tablette ayant en effet une moindre largeur qu'un écran d'ordinateur. Toutes les pages présentant un seul contenu étaient adaptées pour un affichage sur tablette. Cependant les pages de lanceur étaient basées sur un modèle avec 3 colonnes, qui ne tenait pas sur un affichage tablette. La solution à cela était de les placer sur 2 colonnes sur les tablettes et 3 colonnes sur le site bureau. Mais le système de grille étant déjà en place rendait ce comportement impossible, il était basé sur :

- plusieurs lignes composées de 3 rectangles (de taille 4) chacun,
- ces rectangles contenaient un aperçu d'un contenu
- l'aperçu du contenu qui était généralement composé du titre ainsi que d'un visuel, avec parfois un chapeau,

Cette disposition était parfaite pour afficher 3 aperçus de contenu côte à côte, mais lorsque que l'on réduisait la fenêtre pour simuler le comportement sur tablette le rendu donnait : deux contenus côte à côte avec le troisième en dessous ainsi qu'un espace blanc, et ceci sur chaque "ligne". Le nouveau système mis en place enlevait ces lignes et était basé sur le comportement par défaut de Bootstrap, qui était de mettre le contenu qui ne rentre pas sur la ligne suivante. Au final il ne restait qu'une seule ligne avec des rectangles de taille 4 sur la version bureau et de taille 6 sur tablette. En se basant sur le comportement par défaut de Bootstrap, sur la version bureau, les aperçus étaient affichés 3 par 3 et sur les tablettes deux par deux. Le visiteur n'était donc plus perturbé par le changement entre la version bureau en mode paysage et la version tablette en mode portrait.

Lors du passage de plusieurs lignes à une seule, un problème graphique apparut. Les rectangles d’aperçu se plaçaient mal. En effet, comme le système précédent était basé sur plusieurs lignes, il n’était plus nécessaire de préciser une hauteur aux aperçus, puisque c’était la hauteur de la ligne qui était importante. Celle-ci était calculée par le navigateur, en prenant la hauteur de l’aperçu le plus haut qui la compose. Mais avec une seule ligne le navigateur plaçait les blocs d’aperçu où il y avait de la place ce qui entraînait des décalages avec des espaces blancs. Ce problème a été réglé en précisant une hauteur pour tous les blocs d’aperçu.

Après avoir adapté les lanceurs du site bureau à un usage sur tablette, le travail proposé était de les adapter aussi à une utilisation sur le site mobile. Comme pour les tablettes, Bootstrap adapte automatiquement l’affichage. Sur mobile, un lanceur est sous forme de liste où l’aperçu prend toute la largeur. Comme sur les versions bureau et tablette, on retrouve dans l’aperçu une image ainsi que le titre, cependant le chapeau a été retiré. Cette affichage permet se rapprocher d’une interface d’une application mobile native.

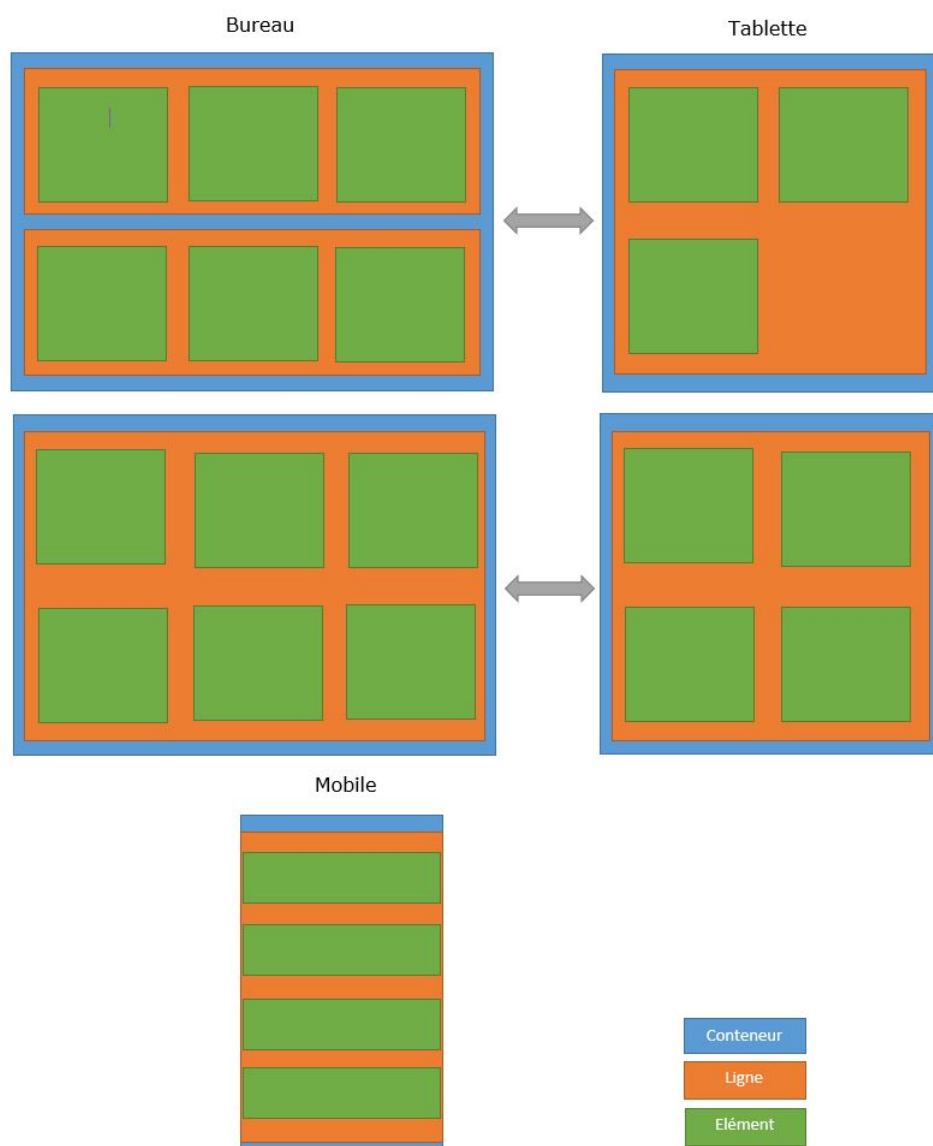


Figure 7: Adaptation sur bureau, tablette et mobile de l’affichage sur 3 colonnes

### 3.4.5 Navigation au sein d'un dossier sur un mobile et tablette

Ce travail consistait à réfléchir et proposer un moyen plus simple de naviguer au sein d'un dossier sur mobile et tablette. Pour rappel, un dossier est constitué d'un ensemble d'articles. Comme chaque contenu, un dossier peut être lié à d'autres types de contenu, généralement des vidéos, dossiers ou articles. La navigation au sein du dossier, sur la version bureau, se fait à l'aide d'une liste de liens qui se trouve à gauche de l'article ouvert. Dans la même colonne, plus bas, se trouvent les liens vers les contenus liés. Le problème est que sur mobile cette colonne apparaît au dessus de l'article ouvert. Le visiteur en arrivant sur un dossier voit en premier lieu le menu du dossier et non pas son contenu. Il faut donc trouver un moyen de naviguer au sein d'un dossier tout en prenant compte de la taille réduite des écrans de mobile.

La première solution proposée fut d'utiliser des flèches de navigation. Ces flèches seront placées en dessous de l'article ouvert, ce qui permettra au visiteur de les voir en arrivant sur la page l'article du dossier. De plus, elles n'apparaîtront que si nécessaire, par exemple sur le premier article du dossier il n'y aura pas la flèche vers le précédent article, de même pour le dernier article. À cela, s'ajoute un indicateur de position dans le titre du dossier pour pouvoir se repérer, mais aussi pour indiquer au lecteur qu'il y a une suite, et que le dossier ne se résume pas à un article. Il se place en fin de titre et indique la position de l'article ouvert par rapport à tous les articles du dossier, par exemple dans un dossier contenant 5 articles, si le lecteur est en train de lire le 3ème article du dossier, à côté du titre du dossier sera affiché "(3/5)". L'avantage de cette solution était qu'elle était optimale pour les mobiles, mais elle présentait plusieurs désavantages :

- pas d'accès direct à un article spécifique du dossier,
- pas de vision globale de l'ensemble des articles du dossier,
- l'interface est adaptée uniquement aux mobiles et pas adaptée aux tablettes.

Pour palier à ces problèmes, une seconde solution fut proposée : en plus des flèches de navigation, un menu déroulant viendrait se placer entre les flèches et permettrait un accès direct à un article du dossier. La difficulté dans cette solution fut de trouver les bonnes dimensions afin que des flèches ne soient pas trop petites ou que le menu déroulant ne soit pas trop petit non plus. En ce qui concerne les tablettes, en mode portrait elles bénéficient de la même navigation que celle pour le mobile, mais elles ont en plus des indications "Article suivant" et "Article précédent" au niveau des flèches de navigation. De plus, afin de retrouver l'esprit du mode bureau sur la tablette, l'image du dossier affichée en haut des autres articles se retrouve au dessus de la liste déroulante sur la version tablette. En mode paysage, les tablettes garderaient la version bureau du site, de ce fait sur tablette les dossiers disposent de deux présentations différentes selon l'orientation dans laquelle on tient la tablette.

Cependant, cette solution ne fut pas retenue pour la version finale du site. L'interface était trop différente de ce que proposait le site en version bureau. Seulement au lieu de retourner comme au début, l'idée que le menu soit placé au dessous de l'article fut gardée, ainsi que l'image pour les tablettes. Au final, la navigation au sein d'un dossier se fait au travers d'une liste de liens qui se situent en dessous de l'article ouvert. Les contenus liés ne sont pas affichés ni sur le mobile, ni sur les tablettes en mode portrait.



Figure 8: Dernière proposition de navigation sur mobile et tablette qui n'a pas été retenue

### 3.4.6 Migration de divers pages

Le travail proposé était de migrer plusieurs pages, c'est à dire les adapter à la nouvelle charte graphique et les adapter à une navigation sur mobile. De petites modifications ont été apportées à ces pages, c'est pour cela qu'elles sont regroupées en une seule partie.

La première page à avoir été retravaillée fut la page de fiche de présentation d'une Caisse régionale. Chaque Caisse régionale dispose d'une fiche de présentation sur le site internet. Sur cette fiche on retrouve les coordonnées, les principales informations, ainsi que les actualités relatives à la Caisse.

La tâche ne concernait pas que le site mobile mais aussi la version bureau. Lors de la migration, la page n'a pas été retravaillée. L'organisation de la page a été revue et les éléments on été adaptés à Bootstrap.

Quelques portlets concernant l'affichage de cartes ont du être adaptées à une navigation sur mobile. Sur la version bureau, ces portlets étaient composées d'une carte et d'une liste d'éléments affichés sur la carte à gauche de celle-ci. Le but était de ne pas afficher cette liste sur mobile.

La difficulté de cette tâche a été de comprendre tous les fichiers relatifs à l'implémentation de l'API V3 de Google Maps. Les fichiers ont très peu évolué lors de la migration et certaines configurations étaient obsolètes. Ce travail permit le nettoyage du code dans le but de supprimer des éléments de configuration inutiles.

Les pages de flux RSS ont également été migrées. Bien que les flux RSS sont de moins en moins utilisés sur internet, ils sont encore très utilisés au sein de l'entreprise. Les navigateurs n'affichent pas les flux RSS de la même manière, il y a ceux qui affiche le flux dans une interface adaptée, et d'autre qui l'affiche avec une feuille de style, quand elle est donnée, sinon ils affichent une page XML.

Les flux RSS générés par le site ne contenaient pas de feuille de style. Le tâche était donc d'associer une feuille de style aux flux.

### 3.4.7 Tests d'intégration continu

L'entreprise disposait de JADE comme plateforme d'intégration. Elle permet d'effectuer les tests de chaque module, cependant les modules ne contenaient aucun fichier de test. La plupart d'entre eux n'en n'avaient pas besoin car il ne contenait pas de classes. Le travail proposé était



donc de produire des tests des fonctions des différentes classes des modules développés par l'entreprise.

En Java et avec JCMS, la production de test passe par la création de classes qui contiennent des méthodes qui testent les méthodes d'autres classes. Pour faire une classe de test, Jalios met à disposition une classe, *JcmsTestCase*, dont il faut hériter. Cette classe permet de mettre en place un environnement de test, avec un store et des configurations spéciales. De ce fait, il est possible de lancer les tests directement depuis Eclipse, qui se chargera de démarrer l'application JCMS.

Cependant, il n'était pas possible de tester toutes les méthodes d'une classe. En effet, certaines fonctions étaient spécifiques à un élément, qui ne pouvait pas être recréé simplement dans un environnement de test.

### 3.4.8 Fil Twitter

Sur les maquettes de la page d'accueil, il apparaissait deux vignettes côte à côte qui serviraient à afficher de la publicité ou des informations à propos des offres du Crédit Agricole. Le responsable proposa de remplacer une vignette par un fil d'actualité Twitter. Ce fil Twitter regrouperait les tweets de toutes les Caisses régionales et certaines autre entités du Crédit Agricole, mais ne remonterai pas les tweets du compte de la Fédération Nationale du Crédit Agricole. En effet, la plupart des tweets de la FNCA renvoie vers le site [creditagricole.info](http://creditagricole.info) et il serait redondant de retrouver les tweets concernant les nouveaux contenus du site sur la page d'accueil. La tâche confiée était d'essayer d'intégrer ce fil de tweet.

Avant tout développement, il fallait se renseigner sur la manière d'intégrer un fil de tweet au sein d'une page. Pour aider l'intégration Twitter propose un site internet dédié aux développeurs où on trouve toute la documentation nécessaire sur les APIs et les modules de Twitter. Afin de créer un fil de tweet, il faut au préalable créer une liste de comptes. Cette liste va constituer l'ensemble des comptes dont les tweets vont apparaître dans le fil. Après avoir choisi la liste il suffit d'ajouter le code fourni dans la documentation à l'endroit où le fil Twitter va apparaître sur la page.

Au cours de l'intégration, un problème de taille s'est présenté, une vignette faisait 200 pixels de hauteur et le fil Twitter est obligatoirement composé d'un bandeau informant la nom de la liste de tweet, ce bandeau occupait quasiment toute la hauteur disponible. Plusieurs maquettes de divers assemblages furent proposées, et une seule fut retenue. Sur la maquette retenue les deux vignettes restent, elles sont l'une au dessus de l'autre au lieu d'être côte à côte, le fil Twitter se place à droite des vignettes et ainsi garde la largeur initiale. De cette manière le fil Twitter dispose de plus de 400 pixels de hauteur et peut donc être affiché en présentant environ 3 tweets.

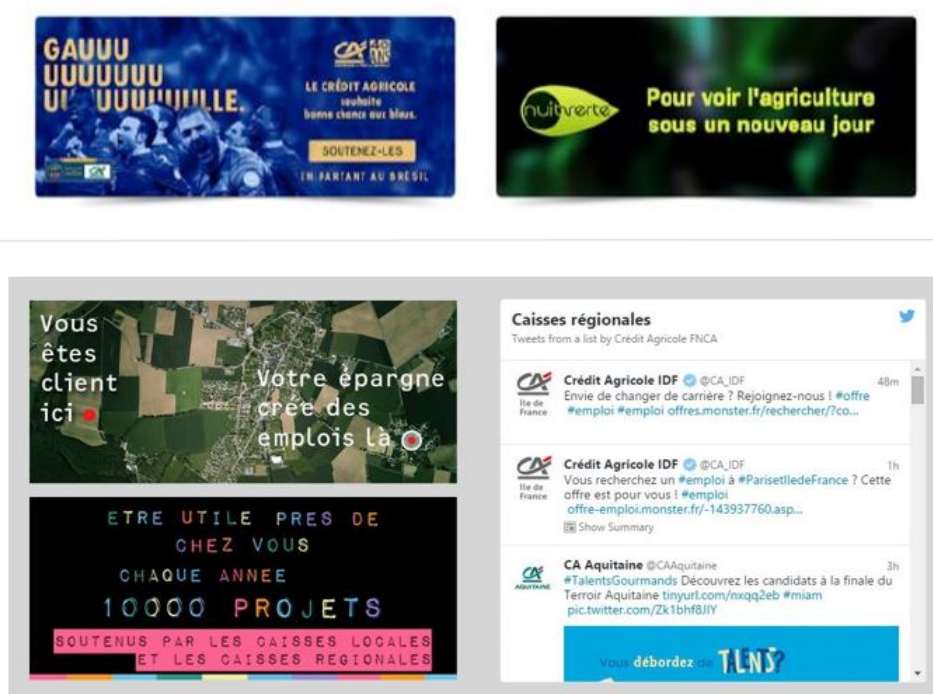


Figure 9: Le fil twitter avant et après intégration

### 3.4.9 Redéfinition de la grille de lecture

A deux semaines de la livraison finale, une réunion a eu lieu dans le but de vérifier la cohérence graphique du site. Comme cela a été expliqué, le cahier fonctionnel ainsi que la charte graphique étaient incomplets. Au cours de cette réunion la grille de lecture du site a été figée :

- la rubrique L'actualité, qui comprend des articles, vidéos et brèves, est celle qui accueille le plus gros volume de contenus : il y a plusieurs brèves et articles par semaine, et en général plusieurs vidéos par mois.
- la rubrique Les analyses produit moins de contenus, mais ceux-ci sont plus complets, plus longs à fabriquer et plus pérennes

La page d'accueil met donc en avant d'abord L'actualité, au travers de 6 contenus qui sont rapidement renouvelés, et ensuite les analyses au travers de seulement 3 blocs, chaque contenu restant longtemps sur la home. Les brèves, qui ne sont pas illustrées, remontent sur la page d'accueil au travers d'une simple liste et non dans un "bloc". Par ailleurs le haut de la page d'accueil est constitué d'un grand bandeau qui laisse une large part à l'image et met en valeur un contenu, par exemple le dernier dossier, une rubrique institutionnelle, une campagne de communication... Cette grande zone visuelle permet de donner un air de famille entre les sites [creditagricole.info](http://creditagricole.info) et [credit-agricole.fr](http://credit-agricole.fr) ; par ailleurs le .info reprend de polices du .fr Sur les pages lanceur, les contenus de la rubrique L'actualité peuvent être filtrés par Caisse régionale pour valoriser la proximité géographique alors que ceux de la rubrique Les analyses peuvent être filtrés par thème (agriculture, développement durable, santé...). Les pages lanceur sont agrémentées d'un grand bandeau purement décoratif (contrairement à la page d'accueil ce bandeau ne fait pas lien vers un contenu).

Les pages de contenu sont découpées en 3 parties :

- le contenu (article, vidéo,...)
- une colonne à côté du contenu listant d'autres contenus liés à l'article (sur le même thème ou dans la même région) ; dans le cas des dossiers il contient la liste des articles et vidéos du dossier,
- un lanceur vers d'autres contenus du même type, situé en dessous des deux parties précédentes, afin d'inciter à rester sur le site.

La colonne de contenus liés est à droite quand on est dans L'actualité, et à gauche quand on est dans Les analyses.

Cette colonne ne doit jamais être vide ; les contenus de la rubrique Les analyses ne posent pas de problème ; en revanche pour la rubrique Les actualités un développement a été réalisé pour remonter automatiquement des contenus se rapportant à la même Caisse régionale et d'autres contenus du même thème.

Cette logique de filtrage des contenus par Caisse régionale et/ou thème se retrouve aussi dans l'interface de recherche.

### 3.4.10 Mise en cohérence graphique de la page des brèves

La page des brèves n'était pas en cohérence avec les grandes lignes définies lors de la réunion sur la grille de lecture. La page était composée d'une seule des 3 parties nécessaires : la partie lanceur. De plus, même si la première brève était ouverte et aurait pu correspondre à la partie contenu, il manquait la partie qui envoie vers d'autres contenus.

Même si un modèle persistait, il y avait deux manières d'accéder à un contenu à partir du menu :

- on accède à une page de lanceur qui liste un ou plusieurs type, celui-ci renvoie vers la page de contenu avec l'aspect vu précédemment. Cet accès est utilisé pour les types articles et dossier,
- la seconde manière permet d'accéder directement à l'aspect vu précédemment. Dans ce cas l'accès au contenu est plus rapide, il est utilisé pour les vidéos.

En ce qui concerne la page des brèves, c'est la seconde solution qui fut adoptée. Ce qui avait motivé en premier lieu la liste en accordéon était l'accès rapide aux brèves. Mais un aspect graphique fut emprunté au modèle des pages lanceurs : un grand bandeau avec image en fond.

En résumé, la page de brèves serait organisée de manière à avoir une brève ouverte, avec une colonne à droite, afin de voir les contenus liés. En dessous de cette brève se trouve le lanceur en accordéon, celui-ci permet de voir les brèves mais pas leur contenu lié. Pour voir le contenu lié, il faut ouvrir la brève, ce qui va conduire remplacer la 1ère brève, ouverte, par celle-ci.

De plus cette nouvelle organisation permet de partager à nouveau une brève sur les réseaux sociaux. La brève ouverte peut être partagée sur les réseaux sociaux. En effet la condition pour partager un contenu était d'avoir une page spécifique pour celui-ci. Ainsi, les balises de renseignement sur le contenu peuvent être intégrées à la page.

Bien que le partage était de nouveau disponible pour les brèves, il était difficile d'accès. Pour partager une brève qui aura été lue dans le lanceur, il faudra d'abord l'ouvrir afin qu'elle devienne la brève principale et, enfin, elle pourra être partagée. Dans cette organisation il y avait un clic et un chargement de page en trop.

La solution serait de pouvoir partager une brève directement depuis le lanceur. Cependant, la solution d'AddThis montre ses limites, elle ne permet pas de partager plusieurs contenus différents au sein d'une même page. Il a donc été décidé de ne pas utiliser AddThis pour les brèves qui sont dans le lanceur mais de faire un développement spécifique. Les boutons de partage ne partageraient pas l'URL de la page actuelle mais plutôt celle de la brève "ouverte". Cependant en utilisant une autre solution, les avantages d'AddThis tel que la personnalisation des liens ou l'outil de statistique ne seront pas disponibles.

Après la mise en production du site, un bogue est apparu au niveau de cette page. Le comportement par défaut de la page était de ne pas ré-afficher la brève principale dans le lanceur en dessous. Cependant, sur le serveur de production, ce comportement n'était pas reproduit et la brève principale était affichée aussi dans le lanceur.

La partie où la brève principale est affichée et la partie lanceur sont dans deux JSP différentes. Pour communiquer des informations entre les deux JSP il n'est pas possible de passer par la page qui les contient, car elle est générique. La communication entre les deux JSP se faisait alors à l'aide de variables de JCMS relative à la page, et donc communes aux deux JSP.

Le problème était qu'en production, il y avait des optimisations qui étaient faites, comme la compression des pages HTML et l'optimisation du javascript. Après avoir activé une à une les optimisations sur le site en local, le problème a pu être isolé, il y avait de l'interférence entre l'optimisation du javascript et les variables relatives à la page.



Figure 10: La page des brèves telle qu'elle est actuellement sur le site

Le problème a été signalé à Jalios pour qu'il puisse le corriger dans la prochaine version de JCMS 9.

### 3.4.11 Cookies

Afin de pouvoir suivre l'évolution des visites sur le site et avoir diverses informations concernant les visiteurs, le site utilise des outils de suivi tel que Xiti et Google Analytics. Pour collecter les données, ces outils utilisent des cookies qui permettent de stocker des informations sur le terminal de l'utilisateur. Une directive de l'Union Européenne datant de 2009, transposée par une ordonnance dans la législation française en août 2011, oblige les sites internet utilisant les cookies, dans le but de recueillir des informations sur les utilisateurs, doivent les en informer. De plus depuis décembre 2013, afin de faciliter son adoption, la CNIL a émis une recommandation relative à la directive. L'entreprise a choisi de se conformer à cette obligation avec la refonte du site.

Le travail proposé était donc d'inclure un message informant l'utilisateur sur l'utilisation des cookies. La CNIL propose deux solutions dans leur recommandation, qui se font au travers de bandeau sur la page :

- la première solution consiste à simplement informer l'utilisateur de l'utilisation de cookies, et qu'il accepte l'utilisation de ceux-ci en continuant sa navigation sur le site,
- la seconde solution bloque l'utilisation des cookies et demande à l'utilisateur son accord avant d'en déposer. Si l'utilisateur refuse, les cookies, servant à recueillir des informations sur l'utilisateur, doivent être bloqués.

Quelle que soit la solution choisie, le message ou le choix de l'utilisateur peut être sauvegardé, sous forme de cookie, durant un période de maximum 13 mois.

L'entreprise a décidé de mettre en place la première solution. Pour ce faire, un message sera placé en bas à droite de la fenêtre, sous forme de pop-up, lors d'une première visite sur le site et restera affiché tant que l'utilisateur n'aura pas fermé la pop-up. Cette pop-up est discrète et permet au visiteur de naviguer à travers le site sans devoir la fermer. Il a été décidé de sauvegarder le cookie 13 mois afin d'éviter que le message n'apparaissent trop souvent.

Sur son site, la CNIL propose des exemples de code pour la gestion de l'affichage du bandeau. Ce code était en Javascript, et a pu être facilement intégré au site internet. Afin que la bandeau puisse apparaître sur toutes les pages du site, le code devait être placé à un endroit où il serait constamment appelé. Au sein de JCMS cela se traduit par la création d'une JSP qui sera appelée juste avant l'appel du *Footer* qui contient une partie du Javascript de la page.

### 3.4.12 Mise en production et premiers retours

Afin de préparer la mise en production du site, un serveur de pré-production fut mis en place quelques semaines avant la mise en production. Ce serveur permettait de faire les derniers tests avec une base de données semblable à celle en production. Il a aussi notamment servi à faire des tests sur la version mobile.

En effet il était impossible d'accéder aux serveurs de tests depuis un mobile, cependant cela a été rendu possible avec le serveur de pré-production. Le serveur deviendra le serveur de production à la date de lancement du site. Contrairement aux serveurs de tests, l'équipe de développement n'avait pas directement accès aux serveurs et devait passer par la DSI pour déposer de nouveaux fichiers.

La date de mise en ligne du site, initialement prévue le 4 novembre 2014, a été repoussé au 12 novembre car la période de pont n'était pas propice et la communication n'avait pas été anticipée sur la mise en ligne du site. Le décalage d'une semaine a permis de rectifier le tir et la communication autour de la nouvelle version fut en place à temps.

Le site a été mis en ligne, comme prévu, le 12 novembre. Il n'y a pas eu de problème majeur lors de la mise en ligne, seulement l'URL du serveur de collecte des statistiques Xiti qui n'avait pas été mis à jour provoquant une absence de données Xiti durant les jours qui ont suivi.

Il y a eu dès la mise en ligne des retours sur les aspects graphiques et ergonomiques de la version mobile :

- les vignettes dans les lanceurs étaient dégradées et trop petites,
- la police utilisée prenait trop de temps à charger, notamment dans le menu,
- le menu recouvre le contenu une fois ouvert et dans le cas d'écran trop petit il ne s'affichait pas entièrement avec aucune possibilité de défiler verticalement.

Le travail proposé fut de corriger ces problèmes, afin d'améliorer la navigation sur mobile.

### 3.4.13 Images des contenus sur le site

Le problème avec les images était leur mauvaise qualité sur mobile et en général sur les miniatures. La source de ce problème est une fonctionnalité de JCMS qui permet de déposer une grande image sur le site, puis il s'occupe de créer des images au bon format lorsqu'elle est appelée. Par exemple, une image de 1920 par 1080 pixels est déposée sur le site, dans une page elle est affichée en 640 par 360 pixels, lors de l'appel, JCMS va créer une image réduite en plus de celle originale. Cependant, lors de cette création l'image va perdre en qualité.

La perte de qualité est accentuée par le fait que sur mobile, les éléments sont mieux définis et lorsque qu'une image n'est pas adaptée à cette définition, elle est légèrement pixelisée. Pour palier à ce problème une solution fut proposée, demander à JCMS des images deux fois plus grandes que celles qui seront affichées. Ce procédé permet d'avoir une image mieux définie, et sur les mobiles elle apparaît nette. Cependant, cette solution présente un défaut, comme les images sont deux fois plus grandes, elles sont plus lourdes mais dans une moindre proportion, cela reste donc acceptable.

De plus sur mobile les images pour les aperçus dans les lanceurs étaient trop petites. Elles ont donc été agrandies en passant d'une taille de 100 par 55 pixels à 130 par 74 pixels, mais comme les images sont deux fois plus grandes que l'espace pour les afficher, ils faisaient 260 par 148 pixels.

De même, ce procédé a été réutilisé pour le site en version bureau/tablette, mais seulement sur les lanceurs. Sur ces versions ce procédé permet d'économiser un peu de bande passante, parce que les images qui sont affichées dans les lanceurs ont la même définition que celles qui sont affichées dans un contenu, il n'y a donc plus besoin de charger une nouvelle image entre le lanceur et le contenu. Pour les autres images, par exemple celle du contenu, leur taille est proche de l'original, de ce fait il n'y a pas de perte de qualité, et donc pas de nécessité à charger des images deux fois plus grandes que celles qui seront affichées. De plus cela conduirait à travailler avec de trop grandes images, de l'ordre de 1200 par 650 pixels.

### 3.4.14 Menu sur mobile

Bien qu'il avait un bon rendu en local, le menu sur mobile était inutilisable en production. Ce problème avait un niveau d'importance haute, du fait qu'il nuisait à la navigation sur le site mobile. Même si ce sont les éléments de Bootstrap qui ont été utilisés, certains choix ont fait que le menu était inutilisable :

- comme dit précédemment, la police prenait trop de temps à charger et à s'afficher. Entre le moment où le menu était ouvert et le moment où la police s'affichait il y avait un délai d'au moins 5 secondes,

- la gestion des sous-menu n'était pas optimale. Le menu du site est composé de catégories, et ces catégories disposaient d'un sous menu qui s'ouvrait lorsque l'on cliquait sur la catégorie,
- le défilement verticale était impossible dans le menu, et il pouvait arriver d'avoir un sous-menu qui dépassait de l'écran,
- impossible de savoir si on avait cliqué sur le bon menu, car la taille du texte était petite et à part le chargement de page rien n'indiquait que la clic avait bien été fait.

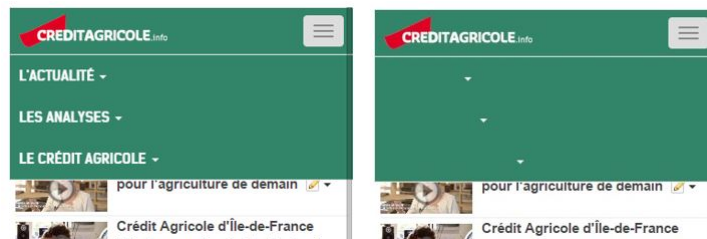


Figure 11: Le menu sur mobile tel qui doit apparaitre et avec le bogue

Plusieurs propositions ont été faites afin de trouver une solution aux problèmes. La première chose qui fut modifié dans le menu fut la police. Le site [creditagricole.info](http://creditagricole.info) utilise une police spéciale pour les titres et les menus, cette police n'est pas une police standard du web. De ce fait pour s'afficher sur le site elle est d'abord téléchargée par le navigateur et chargée à la volée. Cependant, sur mobile ce téléchargement et ce chargement à la volée prennent plus de temps, la connexion et le processeur étant plus lent sur mobile que sur un ordinateur. Plusieurs polices standards du web, elles sont reconnues par tous les navigateurs, ont été testés, et finalement la police Verdana a été retenue.

Une première version fut proposée. Contrairement à celle qui était en ligne, la barre de navigation, qui donne accès au menu, était en haut de la page et non plus fixée en haut de la fenêtre. Quand le menu était ouvert, il ne chevauchait plus la page mais comme il est en haut pousse celle-ci. De cette manière il est possible de défiler verticalement dans le menu. Cependant, comme il n'était plus fixe le seul moyen d'avoir accès au menu était de remonter tout en haut de la page. En plus d'avoir été changée, la police a été grossie pour permettre de ne plus se tromper lors du clic. Même si la navigation au sein du menu était améliorée, la navigation dans les sous-menu n'avait pas changé et le fait de devoir remonter en haut de la page pour accéder au menu était pénible.

Le seconde version mettait l'accent sur la navigation dans les sous-menus. Ce qui a été proposé était d'afficher le contenu de tout les sous-menus, ainsi pour arriver à un élément du sous-menu il y avait moins de clics. Bien sûr, la taille des sous-menus n'était pas importante et quasiment tout le menu tenait sur l'écran sans devoir défiler verticalement. De plus afin d'améliorer la visibilité, les catégories sont en gras et les éléments sont légèrement décalés vers la droite. Aussi, chaque élément est délimité par une bordure fine, comme ce qui se fait dans les applications natives sur mobile, et afin de bien voir la différence entre catégorie et élément du sous-menu, ils auront des couleurs différentes. Une fois que l'on clique sur un élément sa couleur de fond change pour montrer à l'utilisateur qu'il a bien cliqué et où il a cliqué.

La deuxième version réglait les problèmes de navigation dans les sous-menu, cependant, la barre de menu reste en haut de la page. Le problème est que si la barre de menu est de nouveau fixée en haut de la fenêtre, il n'est plus possible de scroller dans le menu. Le menu faisant partie de la barre de navigation, il se retrouverait lui aussi fixé.



La troisième version a essayé d'apporter une solution à ce problème. Pour cela il fallait dissocier la barre de menu du menu lui même, de ce fait il sera possible de scroller dans le menu. Cependant, avec le système en place, le menu qui s'ouvre en accordéon, un rendu avec un menu qui scroll ne sera pas ergonomique. Il fut proposé de mettre en place un menu, sous forme de panneaux, qui ne prendrait pas toute la largeur. Ce type de menu est utilisé dans la plupart des applications mobiles natives.

La mise en place du nouveau menu fut délicate. En effet, le menu en place utilisait les classes de Bootstrap. Il fallait donc garder le comportement par défaut de Bootstrap pour la version bureau, et forcer un autre comportement sur mobile. Le premier rendu était proche du comportement d'un menu d'une application native, celui-ci s'ouvrait en poussant la page vers la droite. Cependant, il présentait un défaut majeur, une fois ouvert la partie de la page, décalée à droite, qui devait être cachée parce qu'elle était hors de l'écran était toujours accessible par un scroll horizontal. Pour palier ce problème un second rendu fut proposé avec le menu qui, en s'ouvrant, ne pousse plus la page mais la chevauche. La largeur de la page ne change pas une fois le menu ouvert.

Cette troisième version fut retenue pour une mise en ligne.



Figure 12: Le menu mobile final

### 3.4.15 Icône pour mobile

Les systèmes d'exploitation mobile permettent de mettre un lien vers un site internet directement sur l'écran l'accueil, sous forme d'icône. Il a donc été proposé de rajouter le code nécessaire pour avoir une icône personnalisée. Après avoir parcouru la documentation mise à disposition par Apple et Google, l'ajout d'une icône personnalisée se fait à l'aide d'une balise *meta* qui se situe dans le *header*. Pour une intégration parfaite sur iOS, il faudra 4 tailles d'icône :

- une de 60x60 pixels pour les iPhones,
- une de 120x120 pixels pour les iPhones rétina,
- une de 76x76 pixels pour les iPads,

- une de 154x154 pixels pour les iPads rétina.

Android est compatible avec la solution d'Apple et ne requiert pas de développement supplémentaire.

### 3.4.16 Migration de Mantis

L'entreprise dispose de plusieurs serveurs de test pour le site [creditagricole.info](http://creditagricole.info), mais elle dispose aussi de serveurs pour d'autres usages. L'applicatif de suivi de bogue Mantis est hébergé sur l'un de ces serveurs. Dans un soucis de réaffectation des serveurs, l'applicatif léger, qui avait un serveur dédié, sera déplacé sur le serveur de test 2 du site (TST2). Le travail proposé était d'assurer cette migration et de documenter chaque étape, afin de pouvoir effectuer une migration dans le futur, si nécessaire.

Avant de faire la migration, il était important de vérifier que le serveur destinataire disposait des programmes nécessaires au bon fonctionnement de l'applicatif. Les deux serveurs disposaient du même système d'exploitation Windows Server 2008, et tous deux disposaient du programme Internet Information Service (IIS), qui fait office de serveur web. Cependant sur le serveur de destination, il manquait PHP et MySQL qui sont nécessaires au fonctionnement de l'application. Cependant, comme le serveur de destination était un serveur de test à usage interne, il n'était pas nécessaire d'installer les versions les plus récentes de ces programmes, mais celles qui étaient compatibles.

Après avoir installé les programmes nécessaires, et avant de migrer l'applicatif, il a été testé qu'un applicatif vierge pouvait fonctionner sur le serveur. Une fois le test effectué, la base de données pouvait être migrée sur le serveur de destination. Ensuite, au lieu de recopier tout l'applicatif du serveur source au serveur destinataire, il fut décidé de garder l'applicatif installé pour les tests, et d'importer uniquement les fichiers de configuration.

Durant la migration, le rapport de nouveaux bogues a été stoppé, pour éviter qu'ils ne soient pas migrés. A la fin de la migration la nouvelle adresse fut communiquée à tous les utilisateurs, et l'applicatif qui était sur le serveur source fut arrêté, pour prévenir de rapporter un bogue sur le mauvais serveur.

### 3.4.17 Module évènementiel

Avant la migration, le site était composé d'un module évènementiel. Ce module sert à mettre en avant un message par le biais d'une portlet, qui s'affiche sous forme de pop-up sur la page d'accueil. Il est utilisé pour les grandes occasions, par exemple pour les fêtes de fin d'année ou pour prévenir la mise en ligne du nouveau site, car il empêche toute navigation sur la page d'accueil tant qu'il n'est pas fermé. Une fois fermé, la portlet ne réapparaît qu'après une période définie (généralement 12 heures). Ce comportement est possible grâce à l'utilisation de cookies.

Le module avait été développé pour l'ancienne version du site et fonctionnait avec JCMS 6. Le travail proposé fut d'adapter ce module pour le nouveau site, donc pour JCMS 9. Afin de ne pas faire trainer de vieux morceaux de code, et au vu de la simplicité du module, il fut décidé de re-développer entièrement le module.

Le développement fut rapide du fait de sa similarité avec le bandeau informant l'utilisation de cookies. Cependant, deux questions se posèrent :

- Comment centrer la portlet, même après un redimensionnement de la page ?
- Comment détecter la page d'accueil ?

Pour centrer un élément, on peut utiliser le CSS, pour cela on le place au centre de la page moins la moitié de sa longueur et sa largeur. Le problème ici est que la taille de la portlet n'est pas connue, cela peut être une image, une vidéo ou même un texte. La solution de ce

problème était dans l'utilisation du Javascript, grâce à lui, il est possible de récupérer la taille d'un élément sur la page et de définir une nouvelle position pour cet élément.

Au lieu de détecter la page d'accueil, pourquoi ne pas simplement mettre le code nécessaire sur la page d'accueil ? Cela n'est pas possible car le développement doit se faire dans un module, il doit être possible de désactiver sans devoir changer le code de la page d'accueil. Avec JCMS, il est possible d'appeler une JSP à la fin du header, afin d'ajouter des balises dans celui-ci, ou avant le footer. Cependant, il n'est pas possible d'appeler une JSP pour une certaine page. La solution proposée à ce problème était d'appeler la JSP qui génère la pop-up avant le footer et détecter dans celle-ci si la page qui était chargée était bien la page d'accueil. Pour détecter la page d'accueil il suffit de vérifier son identifiant. Cette solution n'est pas la plus optimale mais comme il est impossible de détecter une page avant qu'elle soit appelée, c'était la meilleure des solutions.

Après avoir vu une animation d'étoiles se déplaçant dans l'univers, il a été proposé de rapatrier le code de cette animation et de le modifier afin d'en faire un effet de neige qui tombe. L'animation était en Javascript et utilisait la balise HTML canvas. Cette balise HTML est utilisée pour avoir des rendus dynamiques d'images, elle permet d'avoir une zone de dessin.

L'effet est un ensemble de ronds blancs, des particules, de différentes tailles. Chaque particule a une vitesse de chute qui lui est assignée à sa création, de manière aléatoire. De plus, les particules effectuent des déplacements horizontaux, qui sont aussi aléatoires. Lorsqu'une particule touche un des côtés (gauche ou droit) de la fenêtre sa trajectoire est inversée horizontalement. Une fois arrivée en bas de la fenêtre, sa position verticale est remise à zéro.

Après quelques tests, le rendu était satisfaisant et il fut proposé d'intégrer l'animation au sein du module événementiel. L'animation sera une option à activer dans les propriétés du module.

Un problème survint lors de tests plus poussés : l'animation n'avait pas la même fluidité sur tous les navigateurs. La neige de l'animation, les particules, était constituée de petits ronds blancs de différentes tailles. Les navigateurs ne gèrent pas le même nombre de particules à un même niveau de fluidité :

- 500 particules pour Google Chrome, Safari et Opéra
- 100 particules pour Internet Explorer,
- et 10 particules pour Firefox.

Plusieurs optimisations ont été effectuées afin d'avoir plus de particules gérées par Firefox, mais le problème était que Firefox gère assez mal le dessin de multiples ronds. Il a donc été décidé de ne pas activer l'option sous Firefox et de baisser le nombre de particules sous Internet Explorer.

Il a aussi été demandé de supprimer toute dépendance à la bibliothèque JQuery pour l'effet neige. En effet, elle n'est pas utilisée pour gérer la génération et les déplacements des particules, elle servait uniquement à repérer la balise canvas dans le document HTML. De cette manière, l'effet neige pourra être utilisé sur d'autre projet qui ne dispose pas de JQuery, car le chargement d'une bibliothèque demande des ressources, mais surtout de la bande passante.

#### **3.4.18 Flux RSS sur les écrans plasmas**

La Fédération Nationale du Crédit Agricole possède plusieurs écrans plasma, au sein de ses locaux, pour diffuser des informations au public et aux employés. Parmi les informations diffusées, on retrouve des actualités du site [creditagricole.info](http://creditagricole.info).



Figure 13: Plugin événementiel sur le site avec l'effet neige

Pour permettre la diffusion d'information sur les écrans plasmas, l'entreprise utilise une solution développée issue d'une Caisse régionale. Elle permet de contrôler toutes les diffusions sur tous les écrans à partir d'une interface web simple. Pour cela, il faut créer des chaînes, qui contiennent l'information, par exemple une chaîne avec la météo. Ces chaînes sont ensuite rassemblées dans des boucles, qui sont ensuite diffusées sur un ou plusieurs écrans. Un écran ne peut diffuser qu'une boucle.

Une chaîne sert à diffuser les actualités du site. Elle crée une animation à partir du flux RSS récupéré sur le site et une feuille de style CSS, qui est stockée au sein du logiciel. Il est possible de choisir une feuille de style personnalisée, du moment qu'elle se trouve dans un répertoire spécial. Le travail proposé consistait à mettre à jour la feuille de style en y apportant une cohérence graphique avec la nouvelle version du site.

Un problème se posa au cours du développement, la version du CSS supportée par le logiciel. Le nouvel aspect du site intègre des ombres au niveau des contours de bloc, ce qui est une nouveauté apportée par le CSS3, qui n'est pas forcément disponible au sein du logiciel. La seule manière de savoir la version supportée était de tester, mais le logiciel ne permettait pas d'importer soi-même les fichiers CSS.

La DSI a été contactée afin de mettre en place le fichier CSS au sein de la solution, cependant l'absence de documentation sur la solution posait problème. Un rendez-vous fut pris avec les développeurs de la solution, qui entre temps en avait fait un produit. Le rendez-vous aura lieu après la fin du stage.

### 3.4.19 Mise en place de Bibliothèques sur l'espace interne

Après la migration du site internet, l'entreprise avait prévu de migrer aussi l'espace interne. Comme le site internet était migré et que les problèmes de démarrages ont été corrigés, la migration de l'espace interne pouvait être lancée. Pour cela, des réunions furent organisées afin de définir les grands axes de cette migration. Un des points d'évolution était la création de Bibliothèques. L'espace interne est composé d'un ensemble de clubs, et certains de ces clubs ne servent uniquement qu'à diffuser des documents au sein du gestionnaire de documents. L'idée était de faire évoluer ces clubs en bibliothèques, en enlevant toutes leurs fonctionnalités sauf celle de partage de documents. Cela permet de bien faire la différence entre les clubs normaux et ceux

qui servent uniquement au partage de documents.

Le travail proposé consistait à appliquer ces modifications à la version actuelle de l'espace interne. Il a été décidé d'accompagner le plus possible les utilisateurs dans cette migration en apportant, le plus possible, les fonctionnalités au fur et à mesure. De ce fait, l'utilisateur ne sera pas trop perdu lors de mise en ligne du nouvel espace interne.

La tâche était de créer un nouveau type d'espace de travail, bibliothèque, et de créer un onglet "Bibliothèques" dans le menu. La difficulté de ce travail était de travailler avec la version 6 de JCMS, lors de la migration du site internet il n'y avait pas eu besoin de travailler avec une version 6. Il y avait peu de différences techniques entre la version 6 et la version 9 de JCMS, en revanche aux niveaux graphique et ergonomique, où il y avait de nombreuses différences. De plus le site internet n'utilise pas du tout les espaces de travail, alors que c'est l'essence même de l'espace interne.

Dans un premier temps, afin de ne pas trop perturber l'utilisateur, il a été demandé pour les bibliothèques que ça soit un club qui s'ouvre directement sur la page de partage de documents, au lieu de la page d'actualité. De plus, la page de présentation des clubs regroupera aussi les bibliothèques sans distinction entre les deux types. Un fois qu'un membre s'inscrit à une bibliothèque, au lieu de l'ajouter au menu club, il ira dans le menu bibliothèque.

#### **3.4.20 Nettoyage de l'espace interne en préparation à la migration**

Avant la migration du site internet, le site internet et l'espace interne formait un seul et même projet. Ils partageaient la même application JCMS et certaine partie du code était communes aux deux sites. Avant de migrer les deux site sous JCMS 9, il était nécessaire de les séparer. La scission a permit plus de sécurité en passant l'espace interne en *HTTPS*.

Lors de la scission, l'espace interne a garder la plupart des pages mobile du site internet. Le travail proposé était de supprimer ces références à la version mobile du site internet. Cependant, comme certaine parties du code était partager entre les deux sites, ce travail ne consistait pas à simplement supprimer les fichier des version mobile, mais aussi à adapter certaine pages et méthodes pour enlever leur dépendance aux élément du site mobile.

Comme pour les bibliothèque la difficulté ici était de travailler avec une version 6 de JCMS. De plus, une fois les modifications faites, il était difficile de tester si tout fonctionnais au vu de la multitude de page et de fonctions disponible.

### 3.5 Interprétation et critique des résultats

Dans son ensemble le site internet mis en ligne respecte le cahier fonctionnel émit par l'agence web BETC. Cependant, quelques ajustements ont dû être fait car le cahier fonctionnel n'était pas complet. Ces ajustements pour être de l'ordre de l'adaptation, par exemple pour les contenus vidéo, mais aussi de l'ordre de la création, par exemple pour les brèves. Aussi certaines maquettes n'ont pas été respecté, du fait de l'intégration de nouvelles fonctionnalités, comme par exemple le module Twitter. Lors de la mise en ligne aucun problème majeur n'a été détecté, les autres problèmes, notamment concernant le menu sur mobile, ont été corrigés rapidement.

Au niveau fonctionnel, le site correspondait parfaitement aux attentes, cependant au niveau du code, il n'a pas pu être optimisé par faute de délai très court. Par exemple le code CSS produit aurait pu être optimisé, il y a de nombreuses règles redondante au sein d'un même fichier, sans compter les règles inutilisées. Aussi une fois le site livré, nous nous sommes rendu compte que l'organisation n'était optimal, en effet, il aurait été plus intéressant de faire les règles générales pour le mobile et les *media queries* pour la version bureau. Les règles CSS aurait pu aussi être regroupé par fichier, non pas par rapport au module auquel il appartient mais selon la version du site auquel il appartient (mobile, bureau, tablette).

Le manque d'un cahier des charges posa quelques problèmes, tous les points concernant le projet ne furent pas fixés à l'avance. Par exemple la page des brèves a été modifié de nombreuses fois avant d'avoir un rendu satisfaisant.

De plus, la recherche d'information concernant une fonctionnalité était compliqué, car en l'absence de cahier des charges et avec un cahier fonctionnel non complet, la plupart des points qui on été fixés étaient regroupé dans un classeur. Ce classeurs contenait des échanges de mails, des comptes rendus de réunions et des maquettes, ils étaient organisé par date. Il était donc difficile d'avoir une vue d'ensemble sur le projet. Ce point aurait pu être corrigé en tenant à jour un document rangé par fonctionnalité et non pas par date.

## 4 Conclusion

Le stage s'est déroulé à la Fédération Nationale du Crédit Agricole au sein du service Publication et Multimédia. Il avait comme sujet le travail sur une migration d'un site internet, [www.creditagricole.info](http://www.creditagricole.info), dans le but de le mettre en ligne, et assurer une partie du support. Le sujet tournait autour d'une problématique, avoir une interface cohérente et adaptée, que ce soit sur bureau, tablette ou mobile.

Le site est basé sur un système de gestion de contenu (SGC), JCMS. Le but du projet était de migrer le site de la version 6 de JCMS à la version 9. Pour cela les technologies utilisées étaient le Java, Le Javascript et JQuery, Bootstrap, le Less et le CSS, et enfin l'HTML 5.

Pour ce projet, l'entreprise ne disposait pas de cahier des charges, mais d'un cahier fonctionnel et une charte graphique réalisés par une agence web BETC. L'absence de cahier des charges s'explique en partie par le fait que le but du projet était de migrer le site et non pas d'ajouter de nouvelles fonctionnalités.

Le projet a débuté fin avril 2014 et le site a été mis en production le 12 novembre 2014. Le stage ayant début le 8 septembre durant la seconde moitié du projet, il a donc fallu rapidement se familiariser avec le projet.

# Glossary

**JADE** JAlios Delivery Engine. 12, 13, 22

**JCMS** JAlios Content Management System. 7–11, 13–16, 18, 23, 26–29, 32, 33, 35, 37

**SGBD** Système de Gestion de Base de Données. 10, 11

**SVN** Subversion. 13

## Liste des figures

1	Logo du Crédit Agricole . . . . .	3
2	Organisation du Crédit Agricole . . . . .	5
3	Structure d'une page . . . . .	9
4	Architecture d'un environnement de production . . . . .	12
5	Les brèves avant avec le lanceur et page de contenu et après . . . . .	17
6	Bandeau sur la page d'accueil . . . . .	18
7	Adaptation sur bureau, tablette et mobile de l'affichage sur 3 colonnes . . . . .	20
8	Dernière proposition de navigation sur mobile et tablette qui n'a pas été retenue	22
9	Le fil twitter avant et après intégration . . . . .	24
10	La page des brèves telle qu'elle est actuellement sur le site . . . . .	27
11	Le menu sur mobile tel qui doit apparaitre et avec le bogue . . . . .	30
12	Le menu mobile final . . . . .	31
13	Plugin événementiel sur le site avec l'effet neige . . . . .	34