

# Immersion dans l'architecture Lambda



**BEEBUZZINESS**

Creative Engineering & RichMedia Factory

Beebuzziness

Membres du Jury :

Gwenn Boussard

Michel Poitevin - Tuteur entreprise

Francis Brunet-Manquat - Tuteur universitaire

Auteur : Marie kersalé

2020



ubstream

## **Déclaration de respect des droits d'auteurs**

Par la présente, je déclare être le seul auteur de ce rapport et assure qu'aucune autre ressource que celles indiquées n'ont été utilisées pour la réalisation de ce travail. Tout emprunt (citation ou référence) littéral ou non à des documents publiés ou inédits est référencé comme tel.

Je suis informé(e) qu'en cas de flagrant délit de fraude, les sanctions prévues dans le règlement des études en cas de fraude aux examens par application du décret 92-657 du 13 juillet 1992 peuvent s'appliquer. Elles seront décidées par la commission disciplinaire de l'UGA.

A Grenoble,

Le 9 janvier 2020,

Signature

KERSALE MARIE

## Remerciements

J'adresse tout d'abord mes remerciements à Bastien Guillard, directeur de développement, pour m'avoir accordée sa confiance durant cette année d'alternance.

Je tiens à remercier mon tuteur, Michel Poitevin, développeur, pour avoir suivi et valorisé mon travail.

Je remercie tous les membres de l'équipe qui m'ont accompagnés et guidés tout au long de l'année : Antoine Begon, Julien Vienne, Clément Salin.

## Table des matières

I. Introduction.....	5
II. Beebuzziness et son environnement.....	6
II.1 Présentation de l'entreprise.....	6
II.2 Activité de l'entreprise.....	6
L'offre.....	6
Les concurrents.....	9
Les domaines activités.....	9
Les avantages concurrentiels.....	9
II.3 Organisation de l'entreprise.....	9
Les services.....	9
Les outils de communication.....	10
Les modes de régulations du travail.....	10
III. Cadrage mission.....	12
III.1 Description de la mission.....	12
Le contexte.....	12
Description de la mission.....	12
Les avantages de ce système.....	15
L'architecture Lambda plus en détail.....	15
Cadre et Objectifs.....	16
I.1 Cadre technique.....	18
Langages et outils utilisés.....	18
IV. Conclusion.....	19
V. Glossaire.....	20
Bibliographie.....	21

## Table des figures

Figure 1 : Site officiel Ubstream. [p6]

Figure 2 : Hub d'Auchan. [p7]

Figure 3 : Lecteur proposé par Beebuzziness pour consulter des edocuments. [p7]

Figure 4 : Le studio est un outil d'enrichissements de documents virtuels. [p7]

Figure 5 : Catalogue Auchan en ligne. [p8]

Figure 6 : Clients de Beebuzziness. [p8]

Figure 7 : Représentation générale des équipes de développement. [p10]

Figure 8 : Tableau des tâches à réaliser durant le sprint en cours. [p12]

Figure 9 : Accès des statistiques via le Hub. [p12]

Figure 10 : Analyse des consultations de médias dans le panneau de statistiques. [p13]

Figure 11 : Ici on peut voir sur quel réseau le document a été partagé. [P13]

Figure 12 : Ancien système de statistiques mis en place fin 2014. [p14]

Figure 13 : Architecture mise en place cette année. [p14]

Figure 14 : Architecture une fois ma mission achevée. [p17]

# I. Introduction

---

Etudiante en Licence professionnelle Métiers de l'informatique Application web à l'IUT 2 à Grenoble, je suis en alternance au poste de développeuse web au sein de Beebuzziness, entreprise éditrice de logiciels, dans le service développement.

J'ai rejoint cette entreprise Grenobloise en tant qu'intégratrice il y a 4 ans. Depuis deux ans j'ai intégré une équipe de développeurs et commencé mes premiers pas dans le développement. J'ai pu ainsi participer à la conception du Digital Signage\* (service de diffusion interactif de médias\*), du Reporting et des statistiques.

Ma mission est directement liée à ce dernier puisque mon équipe est en charge de la mise en place d'un nouveau système de statistiques.

L'architecture Lambda\* a été choisie et mise en production cette année et j'ai pour mission d'intervenir sur le Speed Layer, pièce manquante aujourd'hui de ce système.

Le déroulement de cette année d'alternance se découpe en 3 étapes, la première consiste à analyser et comprendre le système Lambda dans son ensemble, la seconde concerne la conception et écriture du code du Speed Layer, vu comme un système autonome et lors de cette dernière étape, on va se concentrer sur l'intégration du composant dans le système avec les deux autres composants de l'Architecture Lambda (détails chap. III).

Le choix d'effectuer cette alternance a été motivé par le fait de pouvoir à la fois valider mes acquis et enrichir mes compétences en développement pour une meilleure adaptation lors de nouveaux projets.

Je vais vous présenter dans un premier temps l'entreprise dans laquelle se déroule cette année d'alternance ainsi que son activité et son organisation puis nous évoquerons la mission, son déroulement et ses objectifs.

## II. Beebuzziness et son environnement

---

### II.1 Présentation de l'entreprise

Beebuzziness est une entreprise éditrice de logiciels, fondée en 2001 par Pierre-Nicodème Taslé.

Les équipes de Beebuzziness développent Ustream\*, une plateforme de gestion de contenu en mode SaaS\*.



*Figure 1 : Site officiel  
Ustream*

Le siège social et les activités commerciales sont implantés à Paris tandis que les parties production et R&D sont à Grenoble (45 employés).

La société Beebuzziness fût créée en 2001 dans le but de concevoir une technologie de dématérialisation et d'enrichissement de documents.

La dématérialisation permet de convertir un document pdf en document virtuel, utilisé par exemple dans les grandes entreprises pour la diffusion de rapport annuels ou plaquettes d'information.

La possibilité d'enrichir ces documents avec des ZZO, des liens internes, externes, qrcode et animations pour une consultation fluide.

En 2016, Beebuzziness déploie le Hub\* qui rassemble toutes les fonctions nécessaires à la gestion et à la diffusion de médias\* (détaillé ci-dessous).

### II.2 Activité de l'entreprise

#### **L'offre**

Le Hub\* (Figure 2) est un espace de stockage de documents virtuels autour duquel s'articule différentes fonctionnalités de consultation, de diffusion et d'analyse :

La consultation se fait à travers les fonctionnalités du player, lecteur de documents virtuels (Figure 3), et du studio pour l'enrichissement de documents (Figure 4).

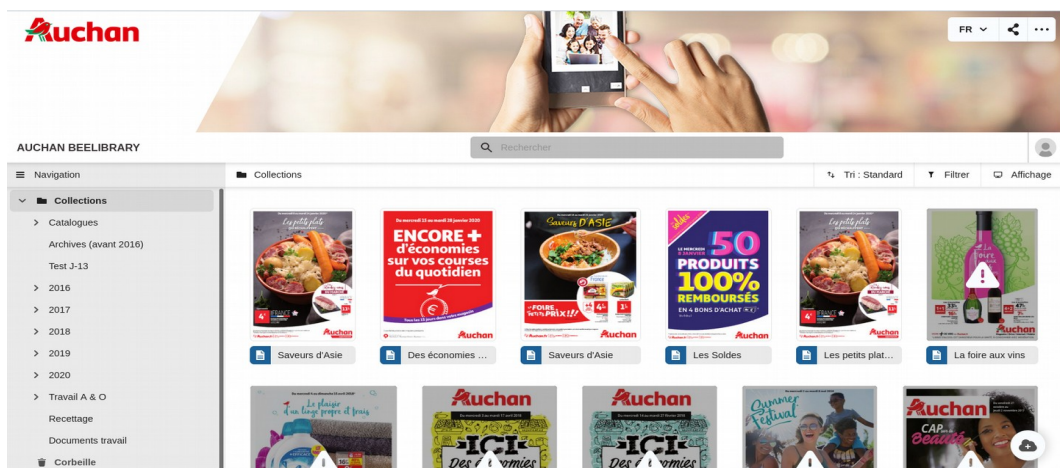


Figure 2 : Hub d'Auchan



Figure 3 : Lecteur proposé par Beebuzziness pour consulter des edocuments.

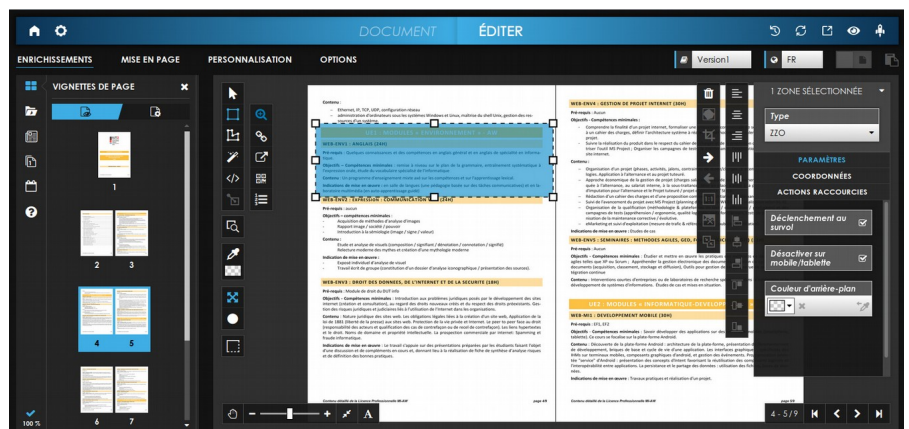


Figure 4 : Le studio est un outil d'enrichissements de documents virtuels.

La diffusion se fait via les options de partages et le Digital Signage\*, diffusion de contenus interactifs. Utilisé pour les points de vente, par exemple, il permet de créer du contenu personnalisé en fonction des stocks à vendre. Les programmes publicitaires personnalisés sont diffusables à la demande sur tous types d'écrans, y compris sur les terminaux mobiles.

Pour terminer, l'accès aux statistiques et reporting permettent d'avoir une vue sur l'activité de vos médias, nombres de visites sur un documents, nombres de retours, combien de temps l'utilisateur, nombre de partage sur les réseaux, quelles pages ont été consultées...

## Les types de clients

Beebuzinness travaille en étroite relation avec de nombreux clients dont certaines enseignes du CAC40 telles que Spie et Orange (Figure 6).

On retrouve deux catégories de clients :

- Ceux orientés produit tels que Electro dépôt et Auchan avec leur e.catalog (Figure 5).
- Ceux orientés information tels que Spie, BPI France, CMA, Orange, FA avec des e.document information.

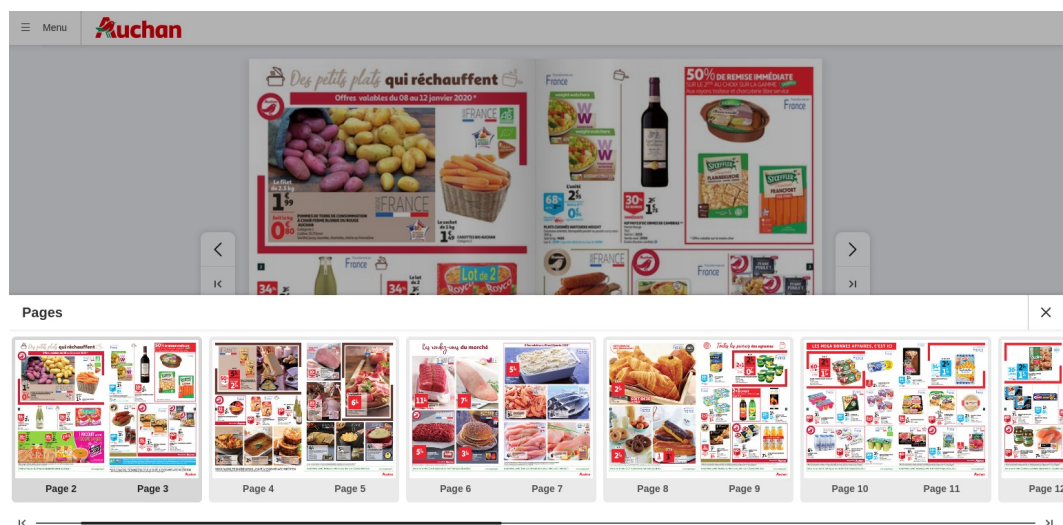


Figure 5 : Catalogue Auchan en ligne.



Figure 6 : Clients de Beebuzziness.



### **Les concurrents**

Les principaux concurrents sont tous les éditeurs de logiciel qui développent des solutions permettant de stocker du contenus :

- Vimeo , Youtube pour la partie vidéo.
- Pinterest pour les images.
- Entreprise du digital publishing, ex : plateforme issuu ( Youtube du pdf).
- Pearlthrees, concurrent direct de myHuB.

### **Les domaines activités**

Les Martechs représente le secteur d'activités. Qu'est ce que c'est ?

Désigne les sociétés et prestataires marketing dont les services sont essentiellement liés à une technologie ou des développements logiciels.

Le domaine d'activités de Beebuzziness se situe à la frontière entre les éditeurs de plateforme web de gestion, la diffusion et l'analyse de contenus digitaux en mode Saas\*.

Le marketing digital couvre l'ensemble des activités marketing déployées en ligne pour entrer en relation avec des clients ou prospects.

### **Les avantages concurrentiels**

Beebuzziness propose d'héberger vos contenus sur un cloud souverain localisé en France et répondant aux standards de sécurité. Vous administrez les rôles et les groupes, tout en gardant la maîtrise de l'unicité, de la validité et de la propagation de vos contenus sur tous types d'écrans, y compris sur les terminaux mobiles.

## ***II.3 Organisation de l'entreprise***

### **Les services**

Beebuzziness comprend 48 salariés repartis dans 5 services, l'administration, la production, la création, le marketing et le développement.

Le pôle de développement axé sur la R&D (Recherche et Développement), regroupe 30 employés dont 5 contrats d'apprentissage, 25 CDI et 10 externes.

Les équipes de développement sont réparties en 6 équipes :

- Hub : Espace de stockage vituel.
- Player : Lecteur, consultations médias stockés dans un Hub.
- ISS/POD/STAT : Digital Signage\*, impression à la demande, statistiques, reporting.
- Documents et virtualisation : Virtualisation, pdf et documents office.
- Médias / permissions : Virtualisation images, vidéos, liens et système de permissions.

- Studio : Enrichissements de médias de type document.

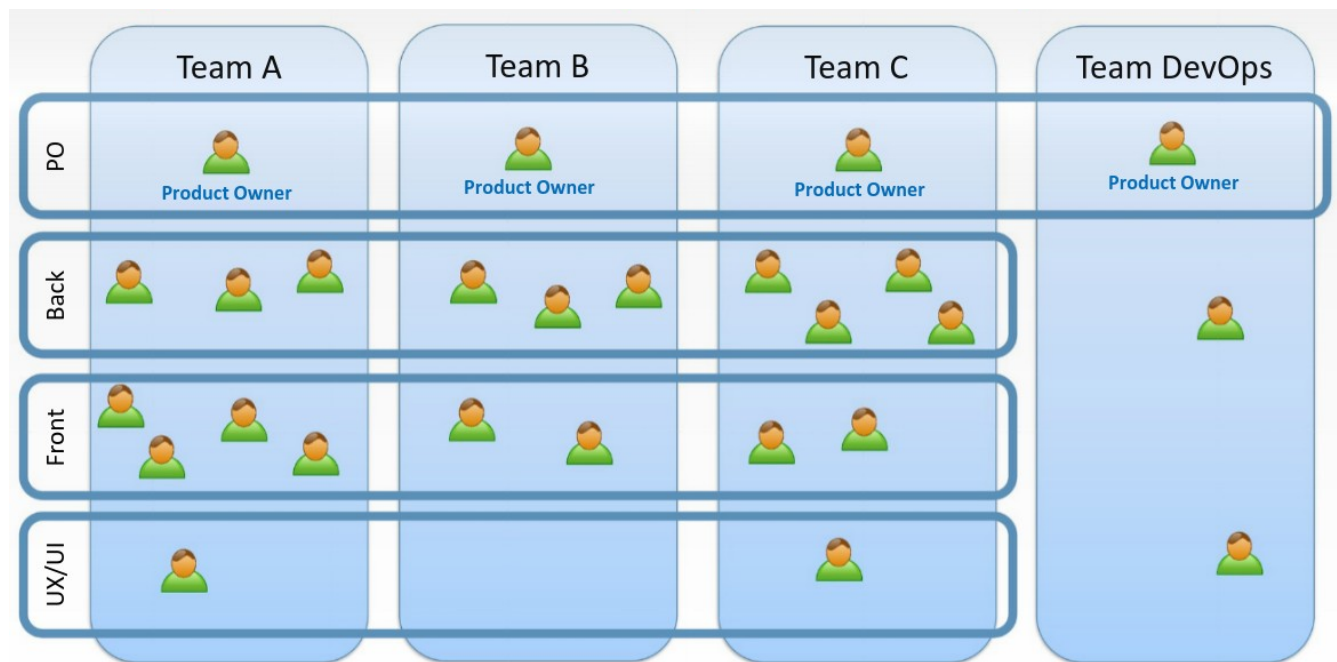


Figure 7 : Représentation générale des équipes de développement.

#### Les outils de communication

Moyens de communication entre les services :

- Slack
- Jira
- Outlook messagerie
- réunions

#### Les modes de régulations du travail

Le comité directionnel est en relation directe avec le directeur de développement, les chefs de produits et les accounts managers. Ils réalisent ensemble une « roadmap » (Figure 7) sur laquelle ils pourront s'appuyer pour mettre en place des objectifs de réalisations.

Les chefs de produits font suivre les informations à leur équipe de développement et les accounts managers aux intégrateurs.

Les équipes du pôle de développement se basent sur les méthodes agiles.

#### Préparation type d'un sprint :

La première étape fait suite aux comités PO\* (chefs de produits / Product Owner), ceux-ci ont comme base de travail la consultation de la roadmap (mise en place entre le comité directionnel et les chefs de

produits). Le PO travaille sur les spécifications fonctionnelles et propose des fonctionnalités à embarquer à son équipe.

Ensemble ils définissent le cadre technique et les fonctionnalités à venir lors du sprint planning qui permet d'organiser les 3 sprints à venir. L'équipe s'engage sur les fonctionnalités suivant des critères de temps, de connaissances de chacun et de priorités.

Pour les sprint N+1 et N+2, le raffinage (réunion d'analyse et de découpage) est mis en place afin que les développeurs soient opérationnels lors du sprint N.

Pour information, le PO [1] a pour mission principale de définir et conceptualiser un produit qui saura apporter le maximum de valeur aux utilisateurs, dans le temps et le budget imparti au projet.

Notre organisation est basée sur les principes de la méthode agile :

Daily meeting ou stand up est un moment dédié aux membres d'une même équipe où l'on échange sur les activités de la veille, les problèmes éventuels rencontrés et la prévision des activités du jour. Ces réunions de courte durée permettent de s'assurer de la progression du projet et d'un bon échange d'information au sein d'une équipe.

Sprint : Itération de trois semaines, une journée étant dédiée à la fin de chaque sprint (rétrospective, raffinage).

Démonstration : Présentation des nouvelles fonctionnalités développées depuis le sprint précédent à l'ensemble des salariés.

Rétrospective du sprint (bilan), est un échange sur le déroulement du sprint précédent, permet de définir des axes d'améliorations.

Sprint planning est destiné à définir l'objectif du sprint et son contenu. Les « user stories » à développer au cours de la prochaine itération seront décidées à ce moment.

Raffinage des stories à venir dans sprint n+1. Détailler, découper, estimer les fonctionnalités.

Jira ou tableau des tâches : Système de suivi de bogues, un système de gestion des incidents et des tâches à effectuer.

Kanban : Tableau blanc divisé en colonnes allant de la liste des tâches à effectuer à la liste des tâches terminées, en passant par toutes les étapes nécessaires. Permet de savoir sur quel environnement (développement, pré-production ou production) les fonctionnalités se situent. (Figure 8)

BurnDown Chart est un graphique généralement utilisé pour suivre l'avancement du *Sprint* (horizon court terme). Il permet de suivre l'évolution du travail restant en fonction du temps. Le but consiste donc à atteindre le niveau zéro le plus tôt possible.

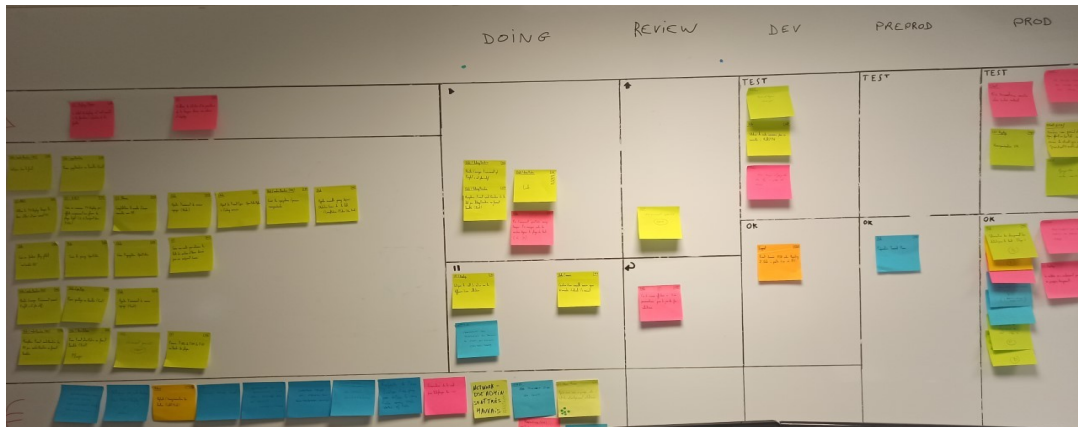


Figure 8 : Tableau des tâches à réaliser durant le sprint en cours.

### III. Cadrage mission

#### III.1 Description de la mission

##### Le contexte

Je me trouve dans l'équipe ISS/POD/STAT qui travaille en mode Agile sur des itérations de 3 semaines. Elle est en charge du Digital Signage\*, du reporting et des statistiques.

Depuis début 2019, mon équipe est chargée de développer une nouvelle version de l'outil de statistiques délivré avec un Hub.

Les membres de cette équipe :

Michel : Développeur fullstack

Julien : Développeur fullstack

Antoine: Développeur fullstack

Clément : Product owner

Rony : Alternant développeur

##### Description de la mission

J'interviens directement sur le panneau de statistiques dont l'accès se fait via un Hub\* :

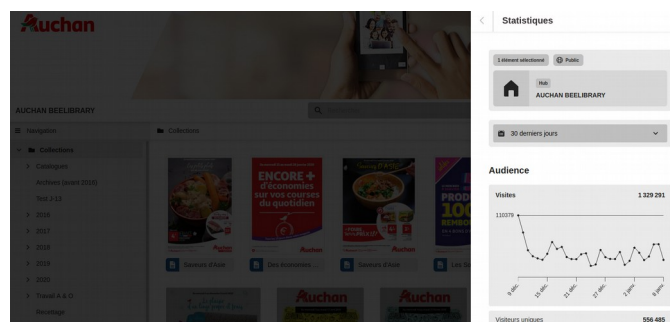


Figure 9 : Accès des statistiques via le Hub.

## Consultation



Figure 10 : Analyse des consultations de médias dans le panneau de statistiques.

## Engagement

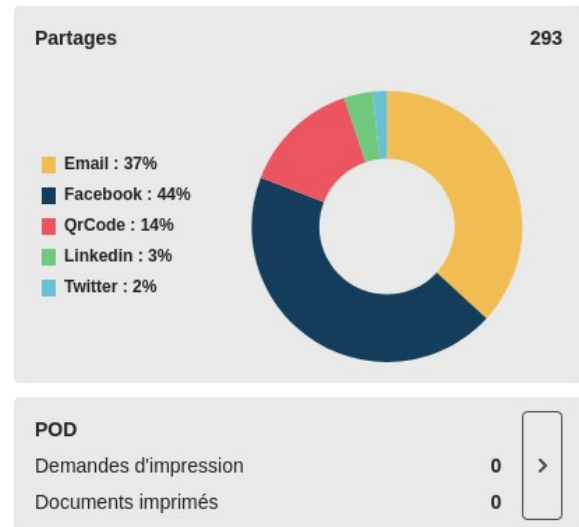


Figure 11 : Ici on peut voir sur quels réseaux sociaux le document a été partagé.

Je dois travailler sur l'architecture Lambda (Figure 5), conçue pour stocker et traiter de larges volumes de données tout en intégrant dans les résultats les données les plus récentes. Elle est composée de 3 niveaux, le Batch Layer (traitement de données par lots), le View Layer et le Speed Layer (traiter le flux des nouveaux événements en temps réel). Ma mission consiste à réaliser ce dernier niveau.

Les données représentent le nombre de visites d'un hub, d'un média, la durée d'une visite, combien de fois un visiteur est venu, les partages réalisés ...

Ce système global va apporter précision et richesse à la donnée collectée et permettre d'améliorer les performances de l'outil de statistiques.

Nous réalisons cela afin de remplacer l'ancien outil de statistiques (Figure 9) pour nos clients et pour une utilisation en interne.

L'utilité pour le client est de suivre et analyser la dissémination des médias, la rétribution des intermédiaires, les recommandations de média au sein de leur bibliothèque virtuelle.

Quant à l'usage en interne, cela va permettre de vérifier les fonctionnalités les plus utilisées et d'améliorer celles qui le sont moins avec un suivi, une analyse et une prédiction des parcours utilisateurs grâce à des statistiques provisoires.

Les clients bénéficiaient déjà d'un outil de statistiques depuis fin 2014, le premier projet conçu en Node.js (premier projet de l'entreprise dans ce langage) pour la partie back-end et réalisé en externe avec Angular pour le front-end.

Cet outil arrivait à ses limites puisque celui-ci ne permettait pas d'ajouter de nouvelles fonctionnalités facilement et ne supportait pas la montée de charge. Cela demandait une maintenance journalière qui monopolisait un développeur à temps plein pour vérifier le bon fonctionnement et récupérer la donnée non intégrée en cas de surcharge (lors de la mise en ligne des catalogues Auchan par exemple).

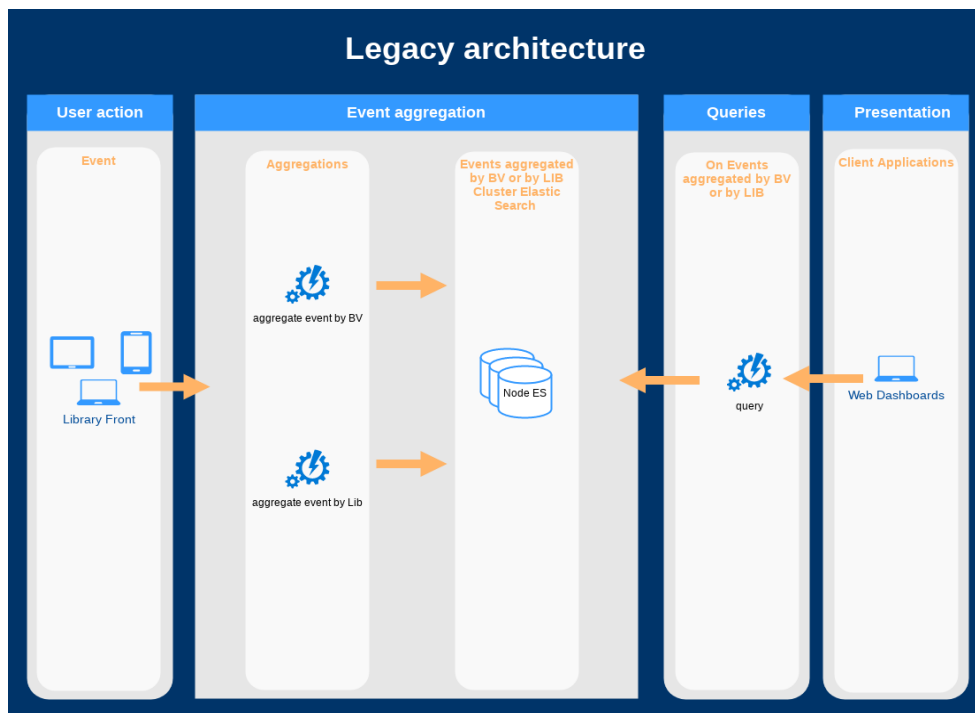


Figure 12 : Ancien système de statistiques.

Ces problèmes sont aujourd'hui résolus avec ce nouveau système de statistiques et ses avantages (Figure 10).

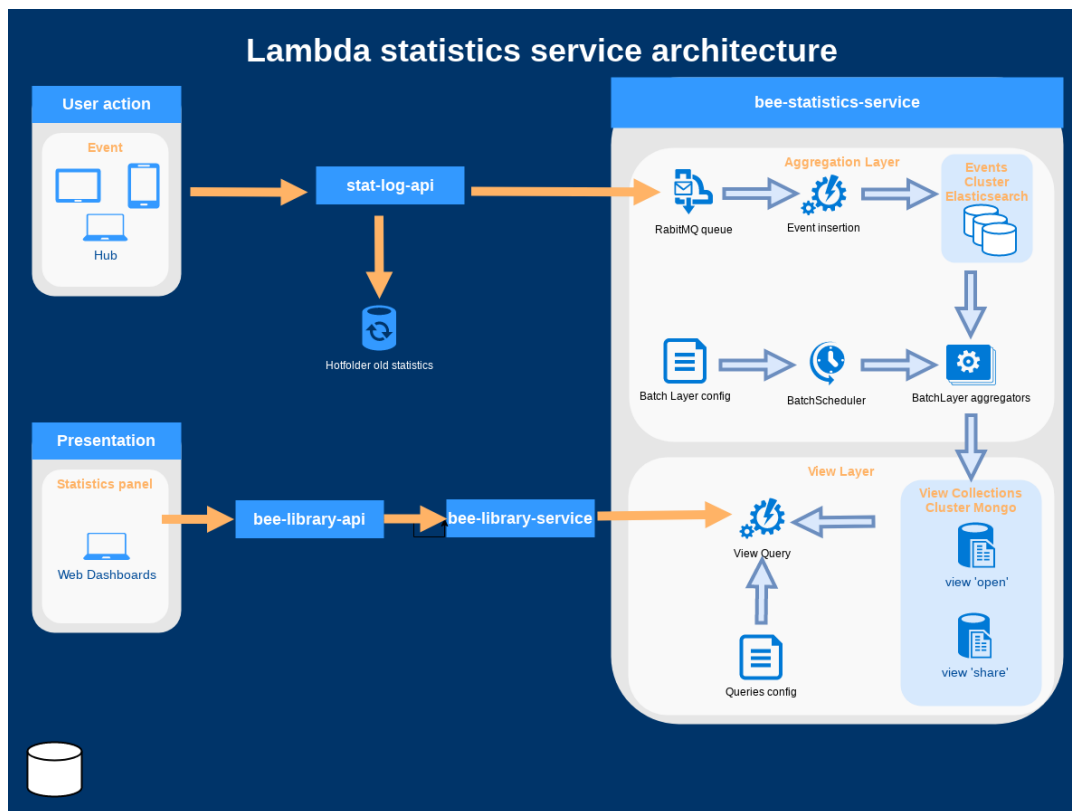


Figure 13 : Architecture mise en place cette année.

### Les avantages de ce système

Scalabilité – Chaque niveau étant géré de manière indépendante, cette architecture est conçue pour gérer de grands volumes de données.

Robutesse – Tolérance aux pannes et aux erreurs

Rapidité – Latence très faible pour l'écriture des évènements, l'exécution des requêtes et donc l'affichage de données même complexes en temps réel.

Evolutivité – Normalisation des évènements permettant de répondre à des besoins futurs avec les données accumulées.

## L'architecture Lambda plus en détail

L'architecture Lambda fournit un modèle de traitement temps réel sur des volumes importants de données en se basant sur :

- Le stockage des événements bruts normalisés dans un puit de données non modifiable
- Traitement des événements et génération d'agrégations
  - Batch layer / View Layer
  - Speed layer (temps réel)
- Requête des agrégations par les applications clientes (Analytics, IA, API).

Le système lambda est composé de 3 couches :

Le **Batch Layer** vise une précision parfaite en permettant de traiter *toutes* les données disponibles lors de la génération de vues, toutes les heures en se basant sur les heures pleines ou tous les jours en se basant sur ààà GMT. Il peut corriger les erreurs en recalculant l'ensemble des données, puis en mettant à jour les vues existantes. On va stocker, dans le View Layer des événements dans un format brut toutes les heures en se basant sur les heures pleines.

Le **Speed Layer** vient compléter, réguler, les chiffres du Batch Layer et permet de minimiser le temps de latence en fournissant des vues en temps réel des données les plus récentes pas encore disponible avec le Batch Layer. Le Speed Layer est donc complémentaire à la vue associée du Batch Layer.

Le **View Layer** est de la donnée grise, qui peut être reconstruite au besoin. Cela correspond à une couche serveur pour répondre aux requêtes. La sortie des couches Batch Layer et couches temps réel sont stockées dans la couche service, qui répond aux requêtes ad hoc en renvoyant des vues précalculées ou en construisant des vues à partir des données traitées.

A cela s'ajoute les **Queries** qui interrogent les vues pré-calculées (View Layer et Speed Layer) pour répondre avec une latence la plus faible possible aux requêtes des applications clientes. La sortie est généralement stockée sur des bases de données NoSQL rapides.

## Cadre et Objectifs

Je travaille seule sur cette mission tout en bénéficiant de l'encadrement de mon tuteur entreprise qui a conçu un accompagnement sur mesure ainsi que celui des membres de l'équipe. Un membre de l'équipe me rejoindra sur les phases d'intégration.

La durée prévisionnelle de cette mission est de 9 mois soit le temps de l'alternance.



Découpage de la mission :

- Octobre : Découverte et compréhension de l'architecture Lambda.
- Novembre-Décembre : Ecriture des composants de base de ce sous système : aggregation, agregation manager.
- Janvier : Integration du speed layer avec le service de consommation des évènements.
- Février : Gestion d'un double buffering sur le Speed Layer.
- Mars : Création des composants de communication entre le Batch Layer et le Speed Layer.
- Avril-Mai : Intégration avec le composant Batch Layer, validation et déploiement en production.
- Bonus : création d'une tâche Kubernetes pour gérer le speed layer.
- Fin Janvier : le Speed layer effectue des calculs qui ne sont pas exploités.
- Mi Avril : le Speed Layer est synchronisé avec le Batch Layer pour effectuer des calculs sur un double buffer. Les calculs ne sont pas exploités par l'IHM.
- Fin de l'alternance : L'IHM bénéficie des données calculées par le Speed Layer.

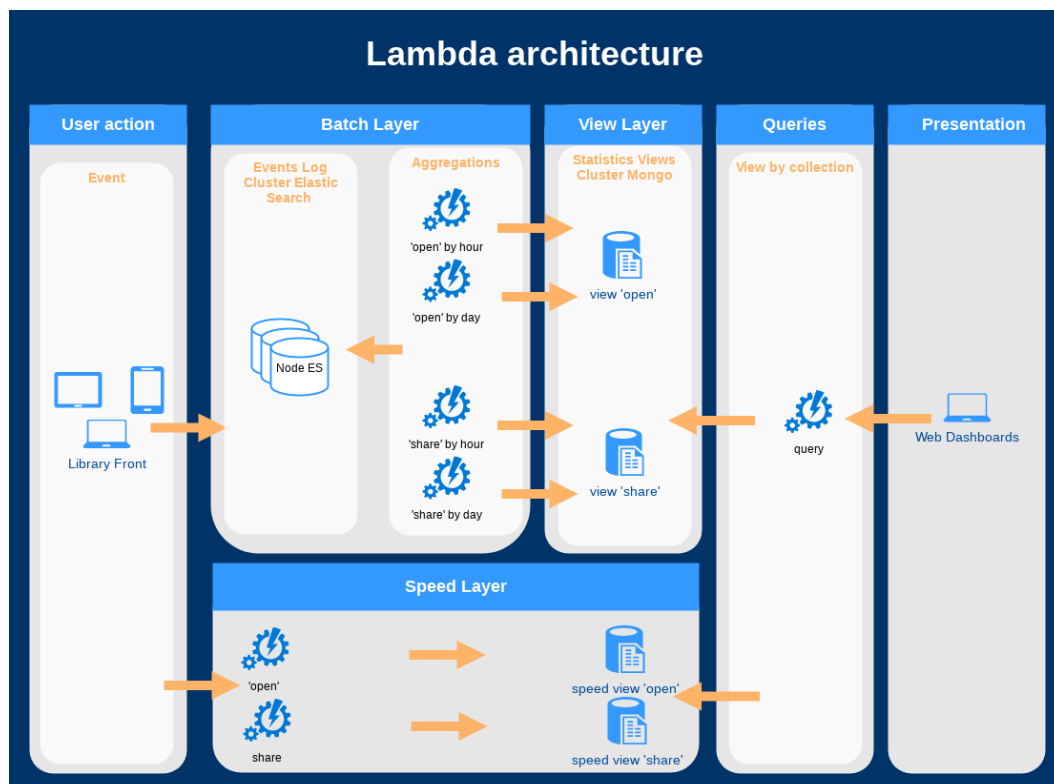


Figure 14: Architecture définitive, une fois ma mission achevée.

## *I.1 Cadre technique*

### **Langages et outils utilisés**

Nodes.js est une plateforme de développement Javascript. C'est le langage Javascript avec des bibliothèques. Node propose une solution rapide pour développer des applications backend. C'est un gain de temps pour le développeur et une économie d'argent pour l'entreprise. Coder les deux parties d'une application web dans le même langage.

#### TypeScript

Langage de programmation libre et open source développé par Microsoft. Améliore et de sécuriser la production de code Javascript en typant les objets manipulés.

Redis est une base clé-valeur en mémoire, qui peut éventuellement persister sur disque les données.

Redis est classé dans la catégorie des bases NoSQL, une solution open-source codée intégralement en C. Outil très rapide. Il est possible d'attribuer une durée de vie aux données insérées, ce qui permet de mettre en place un système de cache. Il est possible d'utiliser des streams pour travailler sur l'ensemble des données qui sont stockées. On peut le partitionner avec des prefixes pour regrouper/isoler de la donnée.

RabbitMQ permet de transporter et router les messages depuis les producteurs vers les consommateurs. utilise les exchanges et bindings pour savoir si il doit délivrer, ou non.

Elasticsearch est une base de données scalable horizontalement, un outil de recherche distribué en temps réel et un outil d'analyse. Une base de données scalable horizontalement permet de rajouter des espaces de stockage sans ré-indexer la data existante.

MongoDB est une base de données NoSQL orientée documents ce qui signifie qu'elle stocke les données au format de documents JSON.

## IV. Conclusion

---

En conclusion, la mission suit son cours, le sujet traité est complexe mais très intéressant avec un encadrement technique et humain poussé.

Lors de ce premier trimestre j'ai pu découvrir et comprendre l'architecture Lambda dans son ensemble et écrire les composants de base de ce sous système : agrégation, agrégation manager.

Les étapes à venir concerne l'intégration du Speed Layer avec le service de consommation des évènements, gestion d'un double « buffering » sur le Speed Layer et la création des composants de communication entre le Batch Layer et le Speed Layer.

Le rythme soutenu des cours ainsi que le flots d'informations à emmagasiner rapidement demande de la rigueur et beaucoup d'investissement avec lesquels je compte acquérir des automatismes et une rapidité d'exécution.

Je suis impatiente de découvrir les prochaines étapes de ma mission et me servir de l'exploration de tous ces langages et outils afin de devenir force de proposition sur les projets techniques.

## V. Glossaire

---

[1] Un **Hub** : Espace de stockage, consultation et diffusion de médias en ligne.

[2] **Interactive Smart Signage** : Solution Saas d’affichage dynamique interactif qui vous permet d’augmenter l’exposition de votre marque sans augmenter vos coûts d’exploitation, et d’engager votre audience de manière personnalisée dans une expérience interactive riche, virale et génératrice de valeur.

[3] **Médias** : Un type de documents, images, vidéos ou liens.

[4] **Méthode agile** : Processus itératif et adaptatif où les équipes travaillent dans une série de cycles courts, incorporant un retour d’information rapide, pour livrer des solutions adaptées aux besoins des clients.

[5] **Roadmap** :

Une roadmap agile s'articule autour des objectifs et des résultats désirés, plutôt que sur les fonctionnalités et les délais. Mettre en place une roadmap agile pour votre produit va vous aider à répondre à cette question avec des itérations sûres et contrôlées.

[6] **Saas** : La gestion en mode Saas (Software as a Service) ou logiciel en tant que service est un concept reposant sur le cloud. Il s’agit de s’abonner à des logiciels sous forme de services délivrés via internet plutôt que de les installer sur les serveurs de l’entreprise.

[7] **Ubstream** : Solution complète de gestion de contenu en mode Saas, qui permet aux organisations de toutes les tailles d’organiser, de valoriser et de diffuser efficacement leurs médias, grâce à des modules simples et rapides à prendre en main.

# Bibliographie

---

Site de Beebuzziness :

<https://ubstream.com/>

Les Martechs

<https://www.definitions-marketing.com/definition/martech/>

Lexique Agile Scrum

<https://agiliste.fr/lexique-agile-scrum/>

Méthode agile

<https://toucantoco.com/blog/methodes-agiles-kanban-revolutionner-management/>

Roadmap

<https://blog.trello.com/fr/roadmap-agile>

BDM :

<https://www.blogdumoderateur.com/definition-developpement-web/ss>

rabbitMQ

<https://www.rabbitmq.com/>

BDM : <https://www.blogdumoderateur.com/definition-developpement-web/ss>

## **Résumé : Immersion dans l'architecture lambda**

Marie kersalé

Beebuzziness, situé à Grenoble, propose une solution globale de communication numérique qui combine les avantages du numérique et de la rematérialisation à la demande en proposant Ustream, une plateforme regroupant toutes les fonctionnalités nécessaires à la gestion et à la diffusion des contenus virtuels.

Ma mission est liée aux statistiques puisque mon équipe est en charge, depuis début 2019, de réaliser une nouvelle version de l'outil de statistiques. Le nouveau système est basé sur l'architecture Lambda et j'interviens sur ce projet. Ma mission consiste à réaliser le Speed Layer, la dernière pièce manquante de l'architecture Lambda.

La mission se déroule en 3 phases :

- \* analyse et compréhension du système Lambda dans son ensemble,
- \* conception et écriture du code du Speed Layer, vu comme un système autonome,
- \* l'intégration du composant dans le système avec les deux autres composants de l'architecture Lambda.

La nouvelle version des statistiques permet d'ajouter facilement de nouvelles informations d'analyse en fonction des demandes clients sans coût supplémentaire pour l'entreprise.

Mots clés : Ustream, Hub, architecture Lambda, Speed Layer

## **Abstract : Immersion in lambda architecture**

Marie kersalé

Beebuzziness, located in grenoble, offers a global solution for digital communication who combines the advantages of digital and rematerialization on demand.

Ustream, platform brings together all the functionalities necessary for the management and distribution of content.

My mission is linked to statistics since my team is in charge of creating a new version of the statistics tool since the beginning of 2019. The new system is based on the Lambda architecture and I am involved in this project. My mission is to realize the Speed Layer, the last missing piece of the Lambda architecture.

The mission is in 3 steps:

- analysis and understanding of the Lambda system as a whole,
- design and writing of the speed Layer code, seen as a system autonomous,
- integration of the component into the system, including communication and synchronization of the component with the other two components of the architecture Lambda.

The new version of the statistics allows to easily add new analysis information according to customer requests without any additional cost for the company.

Key words : Ustream, Hub, Lambda architecture, Speed Layer