

Architecte logiciel

2017-2019

Transition d'une architecture monolithique vers une architecture orienté service Api REST

Présenté par

MESSAOUDI BRAHIM



REMERCIEMENTS

Tout d'abord, je tiens à remercier tous les membres de l'entreprise Natexo de m'avoir accueilli au sein de l'entreprise afin de mener à bien mon alternance.

Un grand merci également à toutes les équipes de l'entreprise ici, en France ou en Espagne et surtout aux membres de l'équipe R&D, mon tuteur et mon manager Philippe Laurent ainsi tous les autres développeurs Elias, Sylvain, Peter et Benoit de m'avoir aidé à évoluer et de progresser et de mieux comprendre le domaine d'activité de cette entreprise.

Merci à NextAdvance, à tous les gens qui étaient là pour un déroulement parfait de notre formation et de nous avoir accompagné et conseillé pendant notre formation et d'avoir répondu à chacune de nos questions.

Je remercie également Madame Amandine Meurer de nous avoir partagé sa bonne humeur et sa gentillesse tout au long de notre formation.

Merci à ma formatrice Laure Bourgois de m'avoir accompagné lors de la rédaction de ce document.

Je souhaite également remercier mon formateur Mbengue Mohamadou de nous avoir accompagnés tout au long de la formation et de nous avoir aidés dans nos projets ainsi tous les autres formateur ou formatrices de NextAdvance

Je remercie mon groupe de projet : Rojo et Alan pour le bon déroulement du projet.

Merci à tous ceux qu'ont contribué de près ou de loin à la préparation de mon diplôme et à la rédaction de ce mémoire.

OBJETIF DU DOCUMENT

Face aux nombreux problèmes d'évolutivité et la réutilisabilité des services commun ainsi les problèmes de performances lié à l'architecture existante. La société Natexo est face à des vrais challenges afin de répondre au besoin de leurs cahiers d'échange constamment évolutif et instable. Avec l'évolution des langages de programmation et l'existence de nouvelles architectures logiciel, il existe aujourd'hui plusieurs solutions qui résolvent ce type de problèmes.

Ce document a pour objectif d'explique ces problèmes d'extensibilité d'un système d'informatique ainsi les solutions proposées par l'équipe R&D de la société

Le premier chapitre dans ce document est consacré pour la présentation de l'entreprise dans lequel nous allons voir l'historique et les différents pôles de l'entreprise ainsi le fonctionnement et le rôle de chaque acteur qui interagissent avec le système d'information de la société.

Dans le chapitre « PROBLIMATIQUE » nous allons commencer par des définitions de certains notions d'mailing afin de comprendre ensuite les processus de chaque pôle de cette société et nous permettrions de voir en détail les fonctionnalités qui composent ce système d'information. On finira ce chapitre par définir le type de l'architecture existante, ses avantages et surtout inconvénients.

Le troisième chapitre est consacré pour proposer deux d'architectures qui sont susceptibles de remplacer l'architecture existante, avec une analyse relative au cas de la société Natexo.

Dans le dernier chapitre nous allons voir la planification et la réalisation de cette migration qui accompagne la réalisation de nouveau produit nommer « Leadgen programmatic ».

SOMMAIRE

I.	LISTE DES TABLEAUX.....	6
II.	TABLES DES FIGURES.....	7
III.	INTRODUCTION.....	8
IV.	PRESENTATION DE L'ENTREPRISE	9
A.	Historique	9
B.	Secteur et entités de l'entreprise	10
C.	Pôle publicité (Advertising ou acquisition ou collecte).....	11
1-	Définition :	11
2-	Les canaux de diffusion :	11
D.	Pôle Monétisation :	12
1-	Définition :	12
2-	Natexo monétisation à travers le monde :	13
3-	Partenaire externe :	14
E.	Rôles et fonctionnement.....	14
F.	L'organigramme l'entreprise.....	17
G.	Mon rôle :	18
V.	LA PROBLEMATIQUE.....	19
A.	Définitions :	19
B.	Le processus d'emailing	22
1-	Le pôle de collecte (publicité ou acquisition).....	22
2-	Diagramme fonctionnel de la collecte	28
3-	Schéma d'architecture technique de la pôle collecte	30
4-	Le pôle Leadgen programmatic :	30
5-	Le pôle monétisation :	31
6-	Diagramme fonctionnel de la Monétisation	34
7-	Schéma d'architecture technique de la monétisation	35
C.	Autres produits de la société.....	35
1-	La plateforme des éditeur (Publisher) :	36
D.	Analyse de l'architecture existante.....	36
E.	L'architecture monolithique	36
1-	Définition.....	36
2-	Avantages de cette architecture	37
3-	Les problèmes de cette architecture.....	37
A.	Analyse des besoin métier de changement du d'architecture	39

B.	Conclusion	39
VI.	SOLUTIONS UTILISEES ACTUELLEMENT	40
A.	Architecture microservices	40
1-	Définition de l'architecture.....	40
2-	Définition d'un microservice.....	40
3-	Les avantages de MSA	41
4-	Les préoccupations de la migration vers MSA.....	42
B.	Architecture SOA	42
1-	Définition de cette architecture.....	42
2-	Définition d'un service	44
C.	Comparaison entre les deux architectures	44
1-	L'analyse de qualité de produit finale	45
2-	L'analyse des coûts de la réalisation	46
3-	L'analyse des délais de la réalisation	46
D.	Conclusion	46
VII.	ARCHITUCTURE CIBLE	48
A.	La transition.....	48
1-	Leadgen Programmatic.....	48
1-	Définition d'un web service	49
2-	Définition de REST	50
3-	La planification.....	51
B.	Réalisation	52
1-	Système de facturation.....	52
2-	Système de déduplication et livraison des leads.....	52
3-	Système de blacklist	53
4-	Service d'authentification avec OAuth.....	54
5-	Service des traces.....	54
6-	Service de statistiques (Leadgen)	54
C.	Choix des technologies.....	54
D.	Diagnostic de nouvelle architecte.....	60
1-	Avantages de cette architecture	61
E.	Les tests unitaires et test d'intégration :	62
F.	Gestion de projet	62
1-	Méthodologies	62
2-	Définition de Scrum.....	63
3-	Les outils utilisés	63

VIII.	CONCLUSION	65
IX.	DEFINITIONS ET LISTE DES ABREVIATIONS	67
X.	BIBLIOGRAPHIE.....	68
1-	Livres	68
2-	Site internet.....	68
3-	Conférences	68
XI.	ANNEXE	69

I. LISTE DES TABLEAUX

Tableau 1 - Comparaison entre MSA et SOA selon la qualité de produit attendu	46
Tableau 2 - Comparaison entre SOA et MSA selon les coûts de la réalisation	46
Tableau 3 - Comparaison entre SOA et MSA selon les délais.....	46
Tableau 4 - Liste des services a implémenté.....	51

II. TABLES DES FIGURES

Figure 1 – L’historique de Natexo	10
Figure 2 - Les domaines et les entités de Natexo	11
Figure 3 - Les canaux de diffusion.....	12
Figure 4 - Carte des pays ou Natexo est présente	13
Figure 5 - La liste des plateformes d’envoi utilisé par Natexo	14
Figure 6 - L’organigramme de la société Natexo.....	17
Figure 7 - Exemple d’une landing page pour les jeux.....	20
Figure 8 - Exemple d’une landing page pour les enquêtes (Survey)	20
Figure 9 - Exemple d'une ad.....	22
Figure 10 - Interface d'ajout d'un annonceur (Advertiser)	23
Figure 11 - Interface graphique de création d'un éditeur	23
Figure 12 - Interface de création d'une campagne.....	24
Figure 13 - Interface de création d'un Survey	25
Figure 14 - Interface d'accueil de projet de la collecte.....	26
Figure 15 - Interface de configuration et le teste d’un webservice	27
Figure 16 - Diagramme fonctionnel de la collecte	29
Figure 17 - Schéma d'architecture technique de la collecte	30
Figure 18 - Interface de création d'une Base	31
Figure 19 - Interface de création d'un kit email	32
Figure 20 - Interface pour uploader les images liées au kit crée.....	32
Figure 21 - Interface pour mettre à jour des bases de données local	33
Figure 22 - Diagramme fonctionnel de la monétisation.....	34
Figure 23 - Diagramme technique de la monétisation	35
Figure 24 - Schéma de présentation d'une application monolithique.....	36
Figure 25 - Schéma explicatif de l'architecture microservices.....	41
Figure 26 - Schéma expliquant le concept de SOA	43
Figure 27 - Décomposition dans SOA et MSA.....	45
Figure 28 - Diagramme de cas d'utilisation de Leadgen	48
Figure 29 - Diagramme fonctionnel de la plateforme collecte de Leadgen	49
Figure 30 - Schéma explicatif de concept des webservices REST	50
Figure 31 - Schéma technique de système de facturation	52
Figure 32 - Diagramme technique de Service de livraison	53
Figure 33 - Diagramme technique de nouveau système blacklist.....	53
Figure 34 - Schéma de fonctionnement de PHP	55
Figure 35 - Répartition des langages de programmation côté serveur web en 2016	56
Figure 36 - Diagramme d'évolution des framework PHP	58
Figure 37 - Schéma technique de la nouvelle architecture.....	61
Figure 38 - L’api doc de projet Leadgen (backend).....	62
Figure 39 - Le cycle de Scrum	64
Figure 40 - La page d'accueil public de projet Leadgen	69
Figure 41 - La page d'accueil pour les annonceurs	69
Figure 42 - La page d'accueil de projet Natexo App.....	70
Figure 43 - Page de statistique de Leadgen selon les éditeurs	70

III. INTRODUCTION

Les systèmes d'informations pour certaines entreprises deviennent de plus en plus complexes cela les rend difficilement extensibles pour répondre aux nouveaux cahiers des charges de ces entreprises, dans ce cas-ci il est nécessaire de définir une certaine démarche afin de mettre en place ces nouvelles fonctionnalités.

Dans certains cas une planification pour ajouter des nouvelles fonctionnalités et son intégration avec un code existant ne suffit pas pour assurer une meilleure extensibilité, les causes de ces problèmes peuvent être diverse dans le cas de Natexo les principaux problèmes sont liés au code existant qui se base sur une architecture monolithique.

Une transition dans ce cas comme celui de Natexo est nécessaire en plus de cette planification initiale, cette transition va nous permettre de passer d'une architecture monolithique qui est lorsque la charge d'une application augmente, l'extensibilité ce système devient un point critique vers une architecture orientée service.

La finalité de cette démarche nous a permis de reconstruire à partir de notre logique métier déjà implémenté un ensemble de briques bien découpées facilement maintenable et en construisant par ceci une base facilement extensible par l'ajout des nouvelles briques.

Dans ce document, nous allons aborder plus en profondeur les enjeux de cette transition, la planification détaillée pour la réaliser après avoir défini et citer les avantages et les inconvénients de chaque architecture, et on termine par les explications des détails techniques pour réaliser cette transition ainsi les technologies utilisées.

IV. PRESENTATION DE L'ENTREPRISE

Fondée en 2008 par Romain Piraux, Natexo Est spécialisée dans l'email marketing d'acquisition grâce à la collecte de prospects, les déploiements de campagnes marketing sur le canal email et la monétisation de base de données. La société est en pleine expansion à l'échelle mondiale à travers 8 filiales : France, Espagne, Italie, Royaume-Uni, USA, Latham, Pologne et Brésil en 2016, Natexo réalise un chiffre d'affaires de 22M € avec plus de 170 employés à travers le monde.

A. Historique

Après sa création en 2008, la société Natexo a passé par plusieurs et différentes évolutions jusqu'à aujourd'hui, la figure ci-dessous présente un schéma explicatif selon les dates des évolutions les plus importants de l'entreprise :



2014

Expansion du groupe : Acquisition de Taz media (Monétisation)

La présence de l'entreprise à l'échelle mondiale ne cesse pas de croître par le lancement des nouvelles agences en Pays-Bas et Russie

2016

Expansion du groupe : Acquisition de Ma Fringue et ouverture de la Pologne et du Brésil et Australie !

L'achat de Ma Fringue : nouveau comparateur en mode en ligne, qui met l'accent sur la mise en œuvre d'une approche innovante

2017

La migration vers une architecture orientée service avec la réalisation de projet Leadgen

La migration des anciens systèmes et la réalisation de projet Leadgen

2019

Lancement de Leadgen programmatic

Lancement de Leadgen programmatic

Figure 1 – L'historique de Natexo

B. Secteur et entités de l'entreprise

L'entreprise est composée de deux pôles (domaines) importants le premier est la publicité (Advertising en anglais) il est appelé aussi la collecte, le second est la monétisation et chaque domaine possède diverses entités le schéma 2 montre les différentes entités de cette entreprise

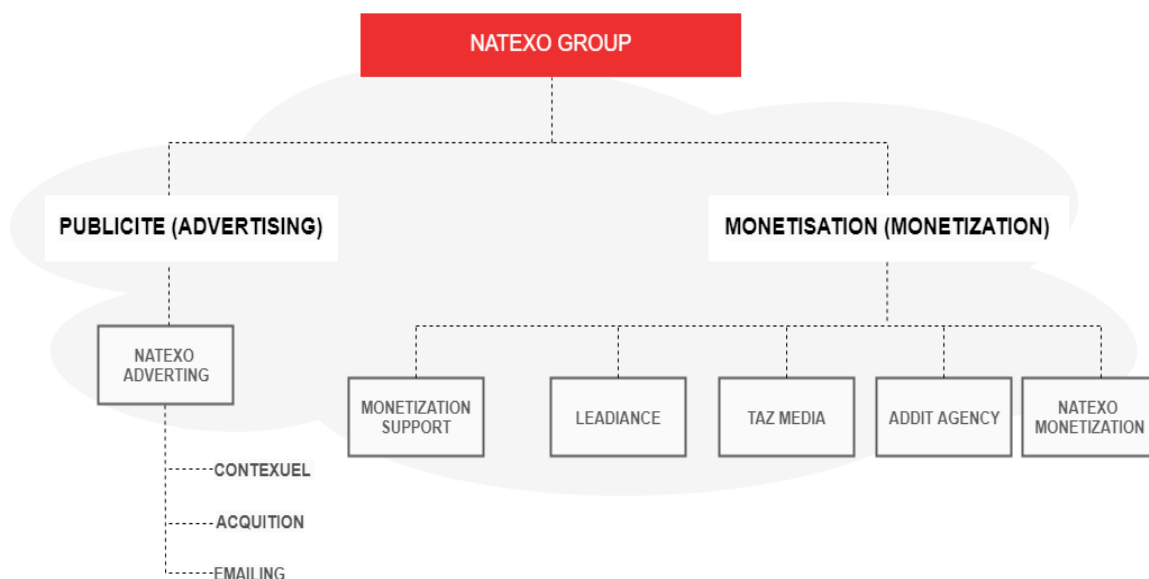


Figure 2 - Les domaines et les entités de Natexo

C. Pôle publicité (Advertising ou acquisition ou collecte)

1- Définition :

L'acquisition client, ou conquête client, est une démarche qui consiste à développer le portefeuille client d'une entreprise. Cette démarche comprend l'ensemble des actions marketing et commerciales qui visent à gagner de nouveaux clients. L'objectif est d'identifier les pistes commerciales pour ensuite les transformer en clients.

Ce pôle est composé de deux domaines jeux(sweepstake) et enquête (Survey)

2- Les canaux de diffusion :

La figure 3 correspond la liste des différents canaux pour faire la diffusion de nos enquêtes et de notre concours campagne

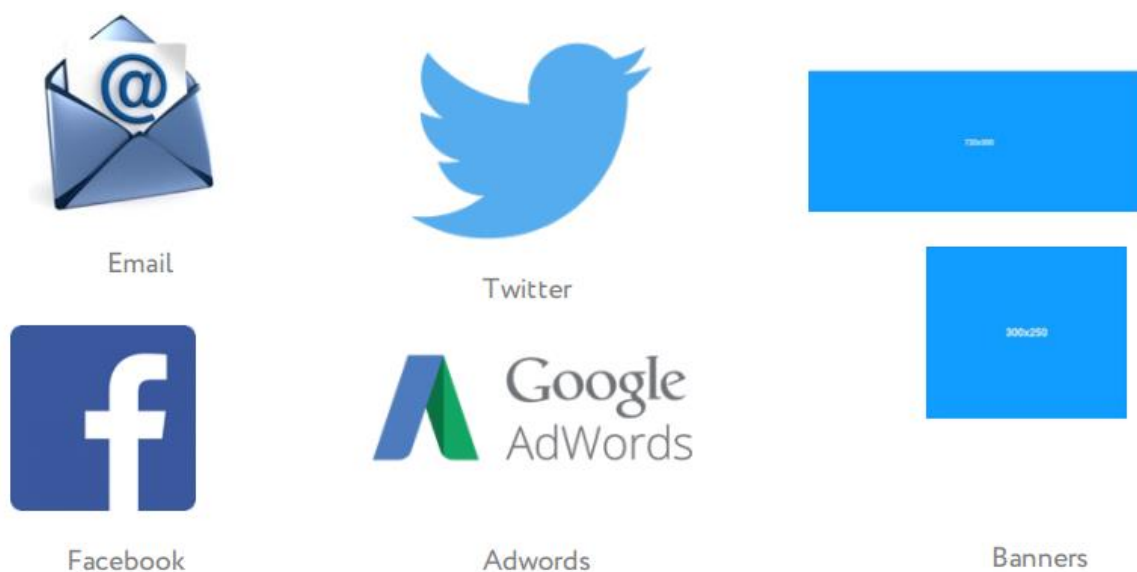


Figure 3 - Les canaux de diffusion

D. Pôle Monétisation :

1- Définition :

La monétisation des données est une démarche par laquelle une organisation revend à un tiers des données qu'il a pu collecter ou produire et dont il est propriétaire. La monétisation des données s'inscrit dans la démarche plus large de valorisation des données et, comme pour cette dernière, ses enjeux se sont considérablement développés avec l'accroissement exponentiel des volumes de données collectées dans un univers toujours plus connecté et digital.

Bien qu'elle soit très présente dans l'univers marketing, la démarche de monétisation des données couvrent également de nombreux autres champs d'activité. Un service de guidage GPS peut revendre ses données à des villes qui souhaitent optimiser leur plan de circulation et un service météo revendre les siennes à des exploitants agricoles. Dans le contexte marketing, la forme la plus connue et la plus courante de monétisation des données est probablement celle liée à la monétisation des données personnelles cependant il existe de nombreuses autres formes de monétisation des données réalisées à des fins marketing. Des données météo peuvent être revendues à des fabricants ou distributeurs de produits météo-sensibles, des données géographiques ou de circulation revendues par des solutions de géomarketing, etc.

2- Natexo monétisation à travers le monde :

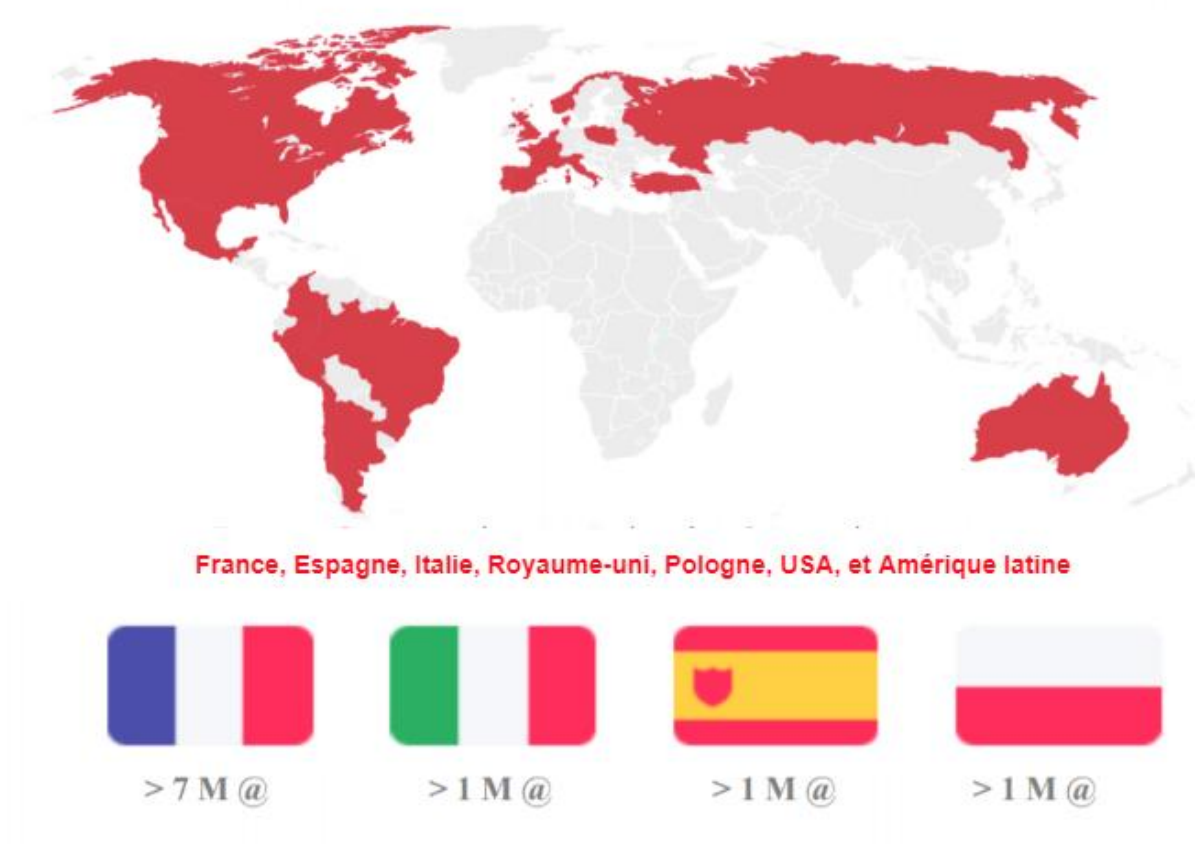


Figure 4 - Carte des pays ou Natexo est présente

Le schéma ci-dessus présente les bases d'emails que Natexo possède par pays, ses emails proviennent de :

- La collecte par des enquête (Survey)
- La collecte pas des jeux (sweepstake)
- Campagne de courrier électronique direct (monétisation)
- Enregistrement (Coregistration en anglais), parrainage (sponsoring en anglais)

3- Partenaire externe :

Les fournisseurs de ce pôle sont les principales plateformes d'envoi



Figure 5 - La liste des plateformes d'envoi utilisé par Natexo

E. Rôles et fonctionnement

Cette partie présente les différents rôles de chaque acteur dans la société qui interagissent avec le système d'information :

- Le responsable d'Affaires

Son rôle principal est de maintenir une relation à long terme avec nos partenaires, l'interprète, l'animateur et les responsables des grands comptes sont responsables du contact et de l'organisation. Deux compétences fortes les caractérisent :

- Le premier est une approche méthodique et structurée de la gestion de campagne et une capacité à interagir avec les clients.
- En charge du suivi des opérations de leurs clients, ils sont proactifs et font leurs propres recommandations à nos partenaires. Ils ont une vision globale des performances de chaque campagne grâce à des outils d'analyse et des tableaux de bord permettant de suivre l'efficacité du service proposé

- Le directeur commercial

Les responsables commerciaux ont pour mission de dégager des marchés et de générer de nouvelles opportunités commerciales "à partir de zéro" ou via des ventes "croissantes" ou "croisées" développant le portefeuille de clients de Natexo. Ils maîtrisent parfaitement nos produits, afin d'être force de proposition auprès de leurs prospects. Ils se distinguent par leurs compétences en matière de négociation, leur appétit pour les chiffres, l'acquisition de nouveaux clients et leur ténacité. Ils ont un

contact facile, une communication claire et une forte force de conviction qui valorise et défend les forces de Natexo et de ses produits.

- Le gestionnaire de circulation

Les responsables de la circulation sont en charge de la mise en place et du suivi des campagnes de nos clients. Ils travaillent en étroite collaboration avec les membres des équipes de vente et de création pour assurer l'efficacité du travail, du brief initial au déroulement de la campagne. Ils surveillent ce processus, à l'aide d'un système de suivi et d'une interface de campagne, et sont en mesure de procéder aux ajustements nécessaires en fonction des statistiques établies. Les responsables de la circulation sont des experts en matière de reporting. Ils établissent des rapports de campagne quotidiens et mettent en œuvre d'excellentes capacités d'organisation et d'analyse, ainsi qu'un excellent sens de la communication.

- Le responsable d'acquisition

Les responsables des acquisitions assurent la génération de leads en déployant diverses campagnes sur des canaux marketing appropriés. Les responsables des acquisitions se caractérisent par leur polyvalence, ils ont une bonne vision des différents canaux de marketing numérique et de la publicité électronique. La maîtrise de cet aspect prend tout son sens lors de la mise en place de campagnes : connaître leurs canaux (affichage, emailing...), maîtriser pleinement leurs outils analytiques (web Analytics, outils de reporting, plateformes de suivi...), être créatif dans leurs propositions de hooks clients... parce que les nouveaux clients seront attirés par de bonnes campagnes créatives et efficaces

- Le responsable de monétisation

Les chargés de comptes monétisation sont responsables de plusieurs bases de données sur lesquelles ils diffusent des campagnes d'e-mailing. Ils sélectionnent leurs offres et négocient les tarifs, planifient et optimisent leurs campagnes pour optimiser leurs résultats. Les moniteurs de gestion de compte sont des experts qui établissent une segmentation pertinente et développent une approche personnalisée pour chacune de leurs bases de données. Ils ont un niveau élevé d'autonomie et de responsabilité dans la prise de décision. Leur compréhension de l'environnement numérique et ses grandes capacités d'analyse leur permettent une gestion rigoureuse et efficace de leurs bases de données pour un ECPM encore plus élevé.

- Le directeur général

En tant que chefs d'équipe, les directeurs généraux ont la responsabilité de gérer et de guider l'équipe opérationnelle et commerciale dont ils sont responsables, de susciter l'enthousiasme et de stimuler leur motivation pour atteindre les plus hautes performances. Animés par les résultats, ils maîtrisent tous les aspects de l'entreprise et établissent des plans d'action pour s'assurer que les membres de l'équipe atteignent leurs buts et objectifs de manière plus efficace. Ils possèdent d'excellentes compétences en

matière de gestion et de négociation et sont des acteurs clés dans le domaine numérique. Leur capacité d'analyse de statistiques lui permet de prendre la bonne décision pour améliorer constamment les performances de Natexo sur leur marché. Très stratégique, mais tout aussi charismatique, la vision de nos managers conduit nos équipes internationales à travers le monde à conduire ses membres au succès.

- Responsable de production

Au sein du service de production, les équipes sont chargées de livrer les créatifs des campagnes publicitaires de nos clients. Gérant le processus de création, les concepteurs Web du service de production travaillent en étroite collaboration avec les équipes opérationnelles, qui fourniront les spécifications et les délais pour la création. Le département de production gère le processus interne pour s'assurer que le client et les utilisateurs bénéficient de créations agréables et performantes. Forts de leur grand sens créatif et de leur grande expertise du marketing par courrier électronique, les concepteurs de sites Web assemblent les modèles de conception typographique, les photos gravées, les illustrations, les copient et préparent la publicité finale.

- Ressources humaines

Les coordonnateurs des ressources humaines travaillent en partenariat avec les gestionnaires et leurs équipes, ainsi qu'avec chaque employé, pour proposer des programmes et des services qui créent un environnement de travail propice à la responsabilisation des employés et à leur implication dans l'entreprise. Nos valeurs d'entreprise de bien-être et de performance sont intégrées à tous les aspects de la gestion des ressources humaines. Axés sur les personnes, les responsables des ressources humaines ont également une vision globale de notre entreprise afin de gérer les effectifs autour de deux objectifs : développer les compétences des personnes tout en organisant le groupe afin de maximiser le succès de l'entreprise. Ils maîtrisent parfaitement les questions de recrutement et les ressources humaines les plus récentes et les plus récentes, afin d'anticiper au maximum les besoins ou les problèmes futurs du groupe.

- La finance

Les membres de l'équipe des finances sont fascinés par les chiffres et par les économies réalisées. Qu'il s'agisse de concevoir des processus pour soutenir nos opérations internationales, de fournir une analyse prospective ou de gérer nos comptes quotidiens, nos comptables et professionnels de la finance créent les solutions les plus innovantes pour maintenir notre entreprise sur la bonne voie passionnée par les chiffres, la connaissance approfondie de notre environnement financier par notre équipe Finance assure le maintien de la santé de l'entreprise. En tant qu'expert financier, ce sont leur réactivité, leur travail acharné et leur détermination qui en font une excellente équipe financière.

- R&D (Recherche et développement)

Dans ce pôle R&D, les développeurs, les statisticiens et les propriétaires de produits jouent un rôle clé dans la conception, l'installation, les tests et la maintenance des systèmes de Natexo. Selon les besoins particuliers de Natexo, ils pourraient être chargés de l'écriture et du codage de programmes individuels ou de la fourniture d'une ressource logicielle entièrement nouvelle, travaillant comme un pont entre les besoins opérationnels et les besoins des clients, ils utilisent les informations clients et les informations commerciales pour concevoir ou redéfinir, développer, tester et mettre en œuvre des programmes et des fonctionnalités complexes. Ils fournissent également un support technique et une expertise aux équipes de Natexo et résolvent tous les problèmes techniques. En effet, ils réagissent quotidiennement aux problèmes et corrigent les programmes et les bogues si nécessaire. Leur compétence proactive leur permet de concevoir des solutions possibles aux problèmes anticipés.

F. L'organigramme l'entreprise

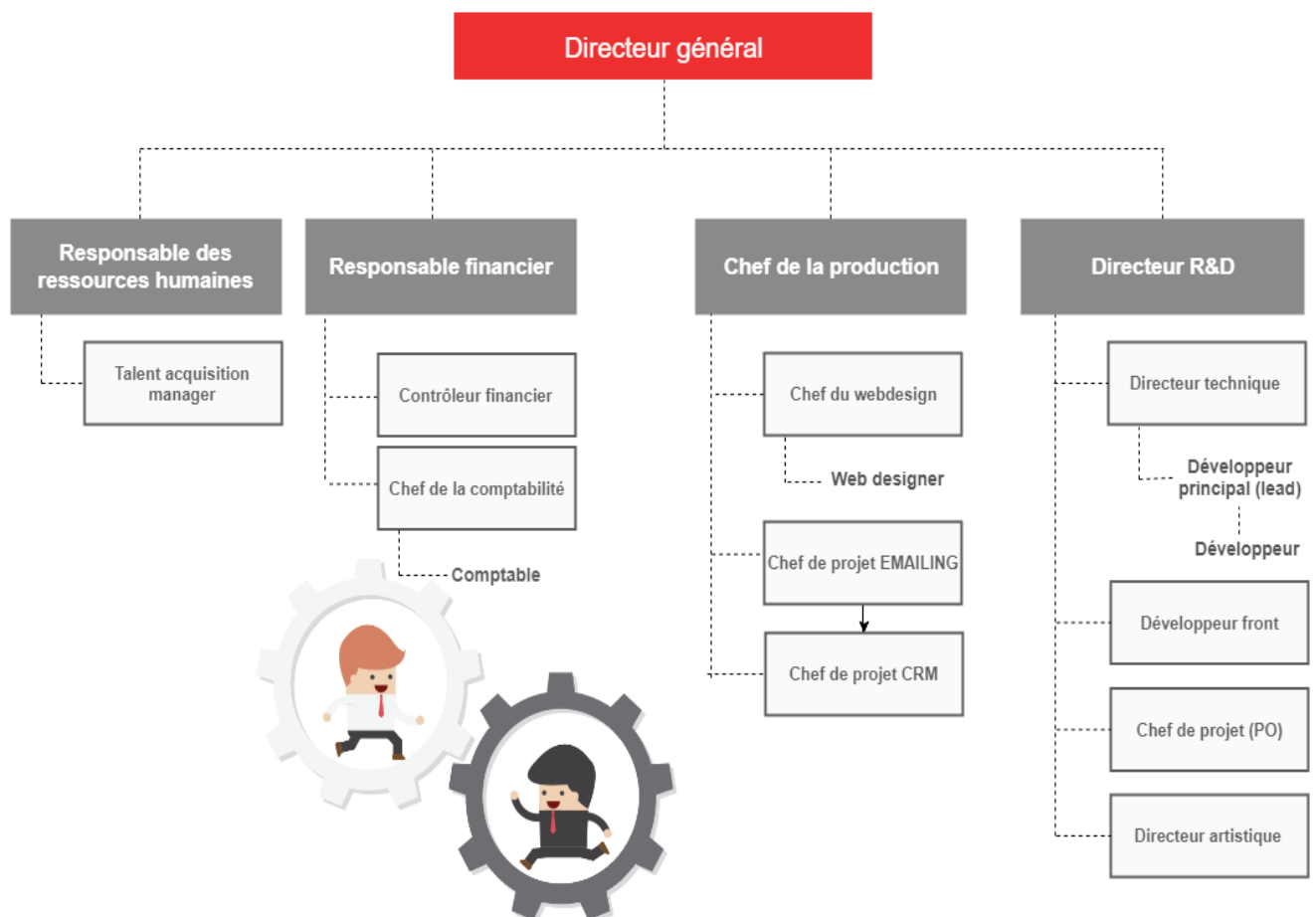


Figure 6 - L'organigramme de la société Natexo

G. Mon rôle :

J'ai commencé à Natexo en octobre 2017, pendant ma première année j'étais avec deux autres développeurs sur la partie support, cette partie la maintenance et la correction des différents dysfonctionnements de système d'information.

Mon travail a été de répondre aux différents tickets concernant tout bug informatique présent sur les différentes plateformes de l'entreprise, cela m'a permis de connaître et de maîtriser le Framework Symfony ainsi la maîtrise de ce domaine d'activité qui reste complexe vu le nombre d'acteurs qui contribuent à réaliser cela et le nombre de processus et services qui sont mis en place, afin d'assurer un meilleur déroulement depuis la création d'une campagne emailing jusqu'à la livraison des leads aux clients.

Depuis septembre 2018 j'ai rejoint les autres quatre développeurs sur la partie release avec la décision de l'entreprise de développer un nouveau produit Leadgen programmatic qui a été lancé en mars 2019, ce nouveau produit nous a permis d'améliorer les services de l'entreprise ainsi l'adaptation de nos techniques d'emailing aux nouvelles méthodes présentes actuellement.

Leadgen programmatic et Natexo app nous a permis de revoir notre architecture existante sur les anciens projets et m'a permis de coopérer à la planification et la réalisation d'une transition architecturale, dans la suite de ce document nous allons expliquer et aborder en détail le processus d'emailing ainsi la transition apportée sur l'ensemble des projets et la mise à la place de ce nouveau produit.

V. LA PROBLEMATIQUE

Pour élaborer la problématique, et en proposer une lecture claire et argumentée, nous devons d'abord définir certaines notions d'emailing nécessaires.

La section suivante comporte :

- Les notions d'emailing,
- Le processus d'emailing depuis la collecte des emails (leads) jusqu'à la livraison au client,
- Le processus de la monétisation.

Nous détaillerons enfin la problématique dans la page 37

A. Définitions :

a) Collecte :

C'est le processus qui consiste à collecter des informations relatives à des internautes pour nos clients (annonceur). Ce processus permet de compléter leurs bases de données. Il est réalisé de deux manières propres à Natexo, jeux (quiz ou sweepstake en anglais) et les enquêtes (Survey en anglais). Un **lead** est le terme utilisé pour désigner les informations récupérées pour chaque internaute.


b) La landing page :

La landing page, appelée également « page d'atterrissage », désigne la page sur laquelle arrive un internaute après avoir cliqué sur un lien (lien commercial, lien email, lien bandeau publicitaire, etc..). L'optimisation de la landing page est une problématique transverse du marketing digital car elle concerne la plupart des leviers marketing digitaux. Il s'agit d'un élément majeur des campagnes. En effet, elle conditionne la transformation d'un simple **clic** en un prospect (plus ou moins qualifié) ou en un client.

c) Jeux (quiz ou sweepstake en anglais) :

Consiste la méthode la plus utilisée par Natexo pour collecter les leads et son but est d'attirer les gens avec des cadeaux pour les faire s'abonner

Win the newest phone on the market!



STEP 1 OUT OF 3

Participate now for your chance to win!

Mr

Mrs / Miss

First name

Surname

Email

DATE OF BIRTH:

Day

Month

Year

Click the button to agree to our [Terms](#) and [Privacy Policy](#).

CONTINUE - IT'S 100% FREE

Yes, I have read and agree to receive advertisements via Email, Phone, SMS, and/or Post about the products or services of the all sponsors listed [here](#) (click here for your communication preferences). I can [unsubscribe](#) my consent without consequences anytime. To continue without giving advertisement consent, [click here](#).

Figure 7 - Exemple d'une landing page pour les jeux

d) Enquête (Survey en anglais) :

C'est la deuxième méthode de la collecte qui se base sur un questionnaire, la figure 8 présente un exemple d'une landing page pour les enquêtes :



APROVECHA AHORA 3 MESES GRATIS



Solicita información sin compromiso

Nombre *

Apellido *

Email *

Teléfono *

Provincia *

☐ Optin *

* campos obligatorios

ENVIAR

Al hacer click aquí acepto ser contactado por AcquaJet y sus partners.

Transporte gratis a domicilio

A precio de supermercado

Agua fría o caliente con solo pulsar un botón

Ideal para preparar alimentos infantiles

Para más información sobre la política de privacidad consulte aquí

Figure 8 - Exemple d'une landing page pour les enquêtes (Survey)

e) Lead :

Dans un contexte commercial, un lead correspond à un contact d'un client potentiel également appelé prospect.

Selon les organisations, le terme lead peut varier. En effet, certains parlent d'un client qui est déjà établi en tant que prospect pendant que d'autres parlent d'un contact commercial. Ce qui est commun à tous, c'est qu'un lead deviendra potentiellement un futur client. Les équipes commerciales sont donc en charge de convertir un maximum de leads pour entretenir un bon taux de conversion.

Un lead basique est composé principalement d'un email et on pourra notamment trouver des leads composés d'un email et un nom et prénom et une adresse, cette composition dépend de type de formulaire affiché lors de la procédure de la collecte.

f) Annonceur (Advertiser en anglais) :

Entreprise ou organisation investissant dans la publicité et, par extension, dans toute forme de communication (promotion, marketing direct, relations publiques). Les annonceurs sont les clients des agences et des supports publicitaires. Par exemple Adidas ou Nike.

g) Editeur (Publisher en anglais) :

Dans le cas de la monétisation l'éditeur est un partenaire externe qui possède des bases de données emailing et qui recherche des campagnes emailing à diffuser dans l'objectif de faire de la marge. En revanche dans le pôle de la collecte l'éditeur peut être des entreprises ou des organisations qui possèdent des site web avec un trafic considérable qui permettra à Natexo de diffuser les campagnes de leurs clients (les campagnes des annonceur), par exemple leboncoin.fr ou easyvoyage.com

h) Hook :

La phrase d'accroche dans une page de publicité

i) Une base de données monétisation :

Une base de données monétisations correspond à une liste bien définie d'email pour pouvoir la monétiser a Natexo, on distingue deux types de base données monétisation :

- Une base de données interne : c'est une base de données qui contient que des emails collectés par Natexo.
- Une base de données externe : c'est une liste d'emails fourni par un client avec de monétiser cette dernière

j) Un kit :

Un kit emailing est un email déjà formaté proposé aux affiliés souhaitant utiliser leurs bases d'adresses opt-in (une adresse active) pour promouvoir l'offre du marchand affilié.

k) Appel à facturation (AAF) :

Document comptable envoyé qui est une fois traité, fera l'objet d'une facture. Un AAF est une demande de paiement, il correspond à une prestation effectuée qui demande un paiement, il y a 2 types d'AAF :

- Les AAF reçu par les clients
- Les AAF que nous générons et envoyons aux éditeurs.

1) Ad :

Ads est le pluriel du mot « ad », qui, en anglais, est le terme informel désignant une publicité (advertisement). De plus en plus utilisé dans le jargon webmarketing, le terme ad, ou ads, est associé aux campagnes marketing et à l'exposition d'une publicité auprès d'une cible particulière

La figure suivante illustre un exemple d'une ad



Figure 9 - Exemple d'une ad

B. Le processus d'emailing

Sachant que la procédure d'emailing peut changer d'une société d'emailing à l'autre, alors voici le processus d'emailing suivi par Natexo selon les différents pôles de l'entreprise et la plateforme utilisée afin de réaliser ses étapes :

1- Le pôle de collecte (publicité ou acquisition)

a) L'ajout de l'annonceur et l'éditeur :

Initialement un responsable de la collecte doit ajouter l'annonceur d'éditeur concerné par cette campagne si ces derniers n'existent pas dans les bases de données de société

Afin de réaliser cette étape des interfaces disponibles sur la plateforme appelée Admin Natexo accessible sur admin.natexo.com

Les deux figures suivantes présentent les interfaces d'ajout d'un annonceur et un éditeur respectivement :

Figure 10 - Interface d'ajout d'un annonceur (Advertiser)

Figure 11 - Interface graphique de création d'un éditeur

b) La création de la campagne :

Cette deuxième étape à la demande de l'annonceur (client) est assurée par l'un des responsables d'acquisition qui procède à la création et la configuration de la campagne emailing.

A la fin de cette procédure on aura une campagne complète qui devrait :

- Avoir un nom.
- Être accordé à un annonceur.
- Avoir une date de début et date de fin
- Choisir la liste des champs à collecter
- Être accordé à une filiale de Natexo qui est divisé par pays
- Préciser le choix de passer ou pas par le système de déduplication
- Avoir des restrictions par rapport au leads collectés, ses restrictions peuvent être sur tous les champs collecter (email, téléphone, adresse IP, l'Age, adresse ou code postal)
- Avoir un ou plusieurs phrases ad
- Choisir le type de livraison des leads

Appelé admin collecte accessible sur <http://collecte.admin.natexo.com> est la plateforme dans laquelle ses employés interne peuvent réaliser cette étape primordiale

La figure ci-dessous présente l'interface de création d'une campagne

Natexo Hello Admin | Home | Advertisers | Campaigns | Surveys | Publishers | Billing | Statistics | Front | Search | Surveys US

Create a campaign for

(A) SELECT THE ADVERTISER

Select your subsidiary then select the advertiser for which you create a campaign.

Subsidiary: Natexo subsidiary | Advertiser: Advertiser

(B) GENERAL INFORMATION

Name of the campaign: Name of the campaign

Customer reference: Customer reference

Display the advertiser logo: ☐

Select a pdf document | Select a file

Required format: .pdf

Activity category: Activity category

Searching for an advertiser:
You must have created your advertiser before creating a campaign. If you can not find your advertiser, make sure it exists in the list of advertisers for the corresponding subsidiary.

Naming a campaign:
Your campaign depends on an IO. Including the signing date of that IO. Give it a name such as XXXXXXXX_SURVEY_IDate

IO reference:
The reference is the number written in the top right corner of the IO. It may contain figures, letters and name of customer.

Viewing the advertiser:
If your campaign is sponsoring, you can choose to display or not the advertiser logo on the first page, under the entry form.

Figure 12 - Interface de création d'une campagne

c) La création de Survey

Cette étape a pour objectif d'associer la campagne initialement créée à une landing page tel que pendant cette on choisit le nom d'affichage du landing sur le site des éditeurs, et le thème de cette landing page, la figure 13 correspond à la page de la création d'un Survey :

Natexo

Home Advertisers Campaigns **Surveys** Publishers Billing Statistics Front

Hello Admin | Home Settings Power

Surveys US

Create a Topic

General information | Thematics questions | Tracking | Campaigns | Sponsors

A Identification

Country of collect: Country

Please specify a theme: Theme

Survey name for the URL: ?

Internal survey name: ?

Survey name displayed in the website: ?

Survey name displayed in the publisher website: ?

☐ Publish the topic on the Publisher interface ?

B Registration page model

Template

Technical name:
This name will be used in the URL of the operation. No brand will therefore appear in the name.

Internal name:
This name will be used only in the interface to identify your operation. So you are allowed to use the brand name.

Marketing phrase:
This name will be used in the public page title. This sentence should therefore be catchy.

Publishing phrase:
This name will be used in the publisher page.

Publisher Interface :
The Publisher Interface allows publishers to view directly the topics to diffuse.
For more information, see the training.

Sponsor media:
A media sponsor is different from a traditional sponsor. It is a

Figure 13 - Interface de création d'un Survey

d) Livraison des leads

Cette phase est la dernière étape de processus d'emailing, pendant cette étape les commercial (employés interne de la société) procède à la livraison des leads, cette livraison peut être effectué par :

- Email :

Cette méthode d'envoi des leads vers un client a assuré par un ensemble de batch (programme lancé automatiquement et quotidiennement) et a pour rôle de générer un fichier en format csv contenant les leads désirés et collectés par la campagne d'emailing afin de les envoyer au client.

Ces batchs (worker asynchrone) se trouve dans le projet qui assure la génération des landings pages

Voici à quoi ressemble l'interface d'accueil de cette application :

SelonVous.com
A smarter survey for a smarter user

S'INSCRIRE SE CONNECTER

Accueil Comment ça marche ?

INSCRIPTION GRATUITE
Rejoignez nos panels de consommateurs et découvrez en exclusivité nos enquêtes rémunérées

100 pts OFFERTS

100% GRATUIT

1 Donnez votre opinion
2 Gagnez des points
3 Recevez des cadeaux

CIVILITE*
NOM*
PRENOM*
BRAHIM.MESS@GMAIL.COM

Date de naissance* :
JOUR MOIS ANNÉE

* Champs obligatoires

☐ J'accepte les Conditions Générales d'Utilisation
☐ J'accepte que NATEXO France collecte mes données personnelles dans le cadre de cette enquête pour les transmettre à ses partenaires commerciaux et aux autres sociétés du groupe NATEXO

Figure 14 - Interface d'accueil de projet de la collecte

- Ftp en format csv :

Cette manière d'envoi semblable à la première méthode, en revanche dans cette méthode les fichiers csv générés sont placés dans des dossiers ftp de nos clients.

- Webservice :

Dans cette solution apportée à la livraison de leads, un ensemble de webservice de type REST ou SOAP est mis en place afin d'envoyer les leads collectés, une équipe de production composée de deux développeurs assure l'ajout et la maintenance de ses webservices.

Ces webservices sont implémentés dans la plateforme Admin collecte, des interfaces sont disponibles pour leur configuration comme la spécification des champs à envoyer au client et la fréquence d'envoi des leads ainsi qu'une interface de test.

La figure 15 illustre l'interface qui permet de configurer et tester un webservice.

Figure 15 - Interface de configuration et le teste d'un webservice

- Manuellement :

Cette dernière méthode d'envoi de lead est effectuée après une demande d'un client, elle est assurée par l'équipe de production, ses dernières procèdent à la génération des exports en format csv depuis la base de données directement avec des requêtes SQL selon les critères de recherche donnés par le client, par la suite ses exports sont transmis par le gestionnaire de trafic au client par email ou par l'envoi en ftp

e) Filtrage des leads par un système de déduplication

Ce système permet de vérifier l'unicité des leads envoyés au client, autrement dit pour ne pas avoir des doublant de leads envoyés au même client, pour cela deux systèmes sont mis en place :

- Un premier système qui utilise un service web externe coté client qui a le rôle de vérifier l'existence de ce lead prêt à être envoyé dans ses bases de données, cette méthode est intégrée dans chaque système d'envoi de lead, on trouve ce système dans la plateforme d'admin collecte et plateforme de landing page ainsi la plateforme Natexo admin pour pouvoir faire les tests,
- La deuxième méthode est un système développé par Natexo, ce système est implémenté dans chaque plateforme disposée d'un système d'envoi.

a) Filtrage des leads par un système de blacklist

Ce système a pour objectif de bien cibler les internautes lors de l'envoi de cette campagne emailing et filtrer les leads collecter afin de répondre aux critères spécifiés par les annonceurs

b) Facturation

Le système le plus important dans ce processus tel qu'il permet à Natexo de généré les factures pour ses clients(annonceur) et de générer les AAF (appel au facturation) et les facteurs des éditeurs ou les annonceurs. Cette rémunération pour un annonceur est basée sur un clic ou un lead tandis qu'un éditeur reçoit des appels au facturation que Natexo doit lui payer, le calcul de ces coûts pour un éditeur est basé sur milles (1000) envoi d'email ou milles diffusion.

Afin de répondre à ce besoin un ensemble de batch sont implémentés et qui tournent quotidiennement, leur fonctionnement se base sur les statiques enregistrés à temps réel des campagnes créés,

On trouve ce système intégré dans les plateformes suivantes :

- Admin collecte.
- Natexo admin accessible sur <http://admin.natexo.com>
- Admin monétisation accessible sur <http://monetization.admin.natexo.com>

c) Statistiques réels

Ce système permet au commercial de suivre les campagnes emailing quotidiennement ou mensuel. A Natexo, il existe deux types de statistique, le premier est partiel relative précisément à ce pôle, un deuxième est global pour cela une base de données dédié à ces statistique, cette base est alimentée par des batchs quotidienne et mensuel (la fréquence d'exécution de batch change selon le type de statistique) ou par des mises à jour à temps réel lors de la collecte ou la diffusion des emails sur une campagne emailing.

La lecture et l'affichage de ses statistiques est dupliqué les deux projet Natexo admin et admin collecte.

2- Diagramme fonctionnel de la collecte

Le figure 16 présente le diagramme fonctionnel de la collecte :

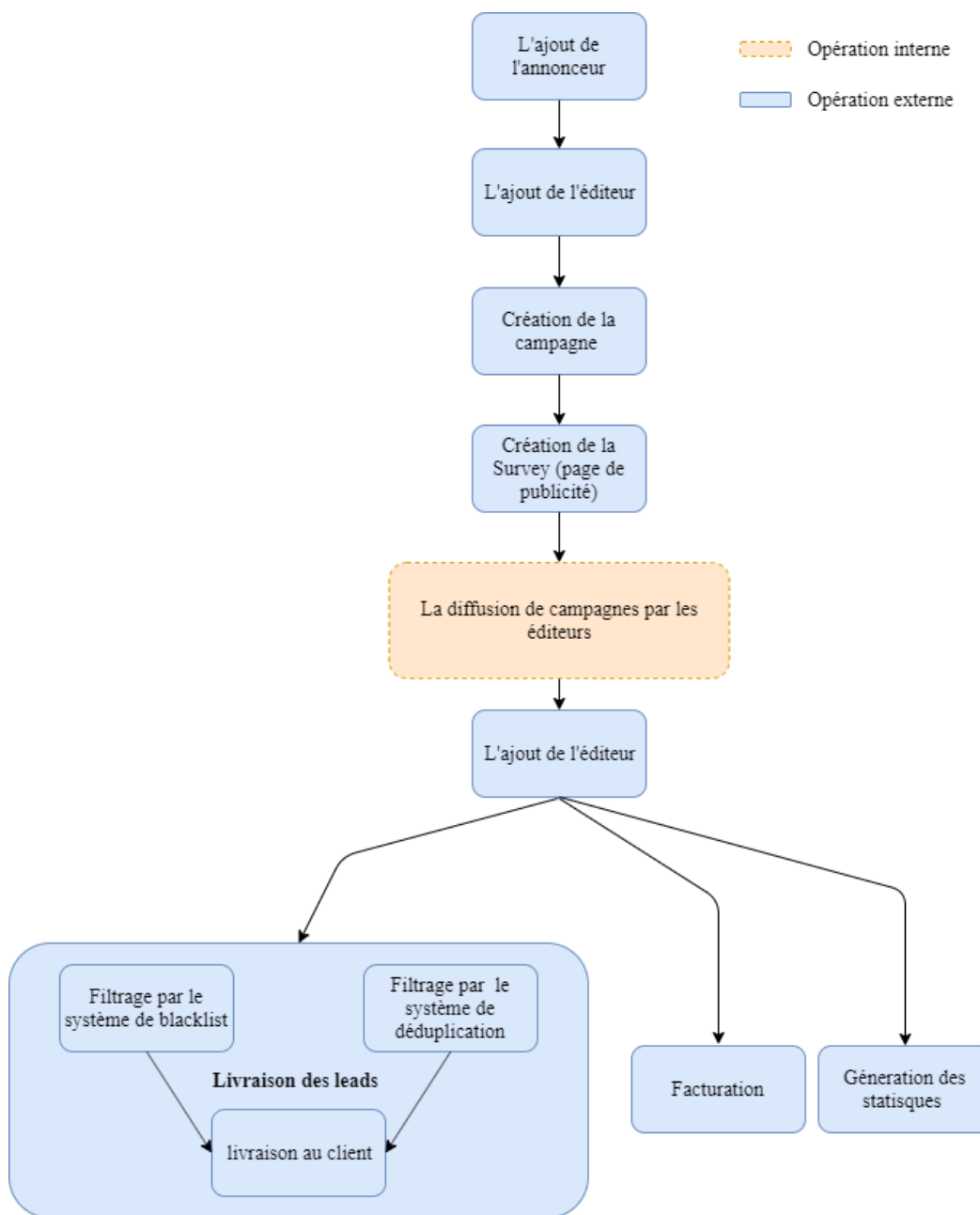


Figure 16 - Diagramme fonctionnel de la collecte

3- Schéma d'architecture technique de la pôle collecte

Le schéma ci-dessous présente un diagramme d'architecture technique qui a pour objectif d'illustrer les différents plateforme(projets) utilisés dans le pôle de la collecte ainsi les bases de données mise en place :

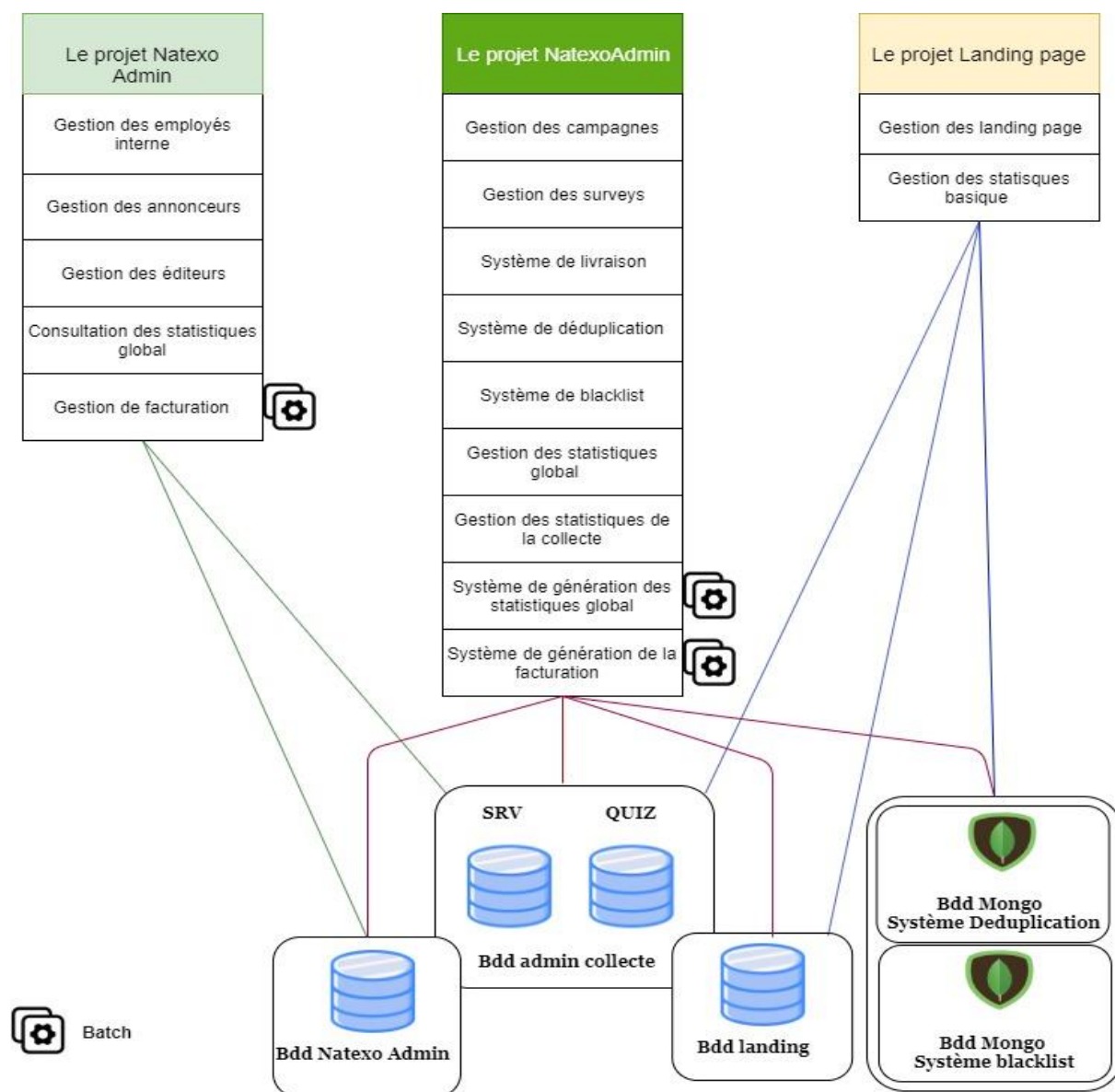


Figure 17 - Schéma d'architecture technique de la collecte

4- Le pôle Leadgen programmatic :

Ce pôle récemment lancé a pour objectif la collecte des leads, son processus est identique à celui de la collecte (voir la section précédente - page 23), la différence entre ces deux processus est la manière de collecter les leads car dans la collecte on utilise principalement les landing page tandis que les ADS sont utilisées dans Leadgen.

5- Le pôle monétisation :

a) L'ajout de l'annonceur et l'éditeur :

Cette étape est semblable à la première étape pour la collecte (Voir la page 22)

b) La création d'une base de données monétisation :

Pendant cette première étape un responsable de monétisation procède la création d'une base données monétisation en précisant la date de début et fin de sa monétisation, dans le cas d'une base de données monétisation interne (la liste des emails a monétisé provient de Natexo) une campagne d'emailing peut-être relier à cette base afin de collecter plus d'information de cet internaute sinon (une base externe) une liste d'email fourni par le client qui sera relie à cette base.

Pour assurer cette étape la société a mis en place une plateforme nommé admin monétisation disponible sur le lien suivant <http://monetization.admin.natexo.com>

La figure ci-dessous présente d'interface graphique pour la création d'une base

The screenshot shows the 'Create a new base' interface in the Natexo admin panel. The top navigation bar includes 'Home', 'Customers', 'Reporting', 'Statistics', 'Synchro.', 'Billing', and 'Tools'. The user is logged in as 'Hello Admin'. The main form is titled 'Create a new base' and contains three main sections: 'Advertiser data' with dropdowns for 'Advertiser's name', 'Contact', and 'Program's name', and a text field for 'Name of the base'; 'Contract data' with date pickers for 'Date of the contract', 'Start of the service', and 'End of the service', input fields for 'Compensation' and 'Marketing pressure' (both followed by a '%' sign), and a 'Jauge' (gauge) set to 0%; and 'Restrictions' which is currently empty. An 'Add contract' button is located next to the contract date fields.

Figure 18 - Interface de création d'une Base

c) Création de la campagne :

Dans cette phase un responsable de monétisation procède à la création de la campagne monétisation relative à la base de monétisation créée dans la première phase

d) La création de kit d'email :

Cette étape consiste à définir le sujet et la mise en page de l'email diffusé dans cette base, elle est assurée par un responsable de la monétisation et la Template d'email doit être validé par le client.

La création de ce kit se fait, en utilisant deux interfaces disponibles sur la plateforme admin monétisation, la figure ci-dessous montre l'interface graphique pour la créer et test un kit

This will appear in the **Subject** field of the recipient.

Aviva SDW

Enter the content of your campaign

```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
2
3 <html>
4 <head>
5   <meta http-equiv="Content-Type" content="text/html; charset=us-ascii">
6   <style type="text/css">
7     /**/
8     .ReadMsgBody { width: 100%; }
9     .ExternalClass { width: 100%; }
10    table { border-collapse: collapse; }
11    /*]]&gt;*/
12  &lt;/style&gt;
13
14  &lt;title&gt;&lt;/title&gt;
15 &lt;/head&gt;
16
17 &lt;body&gt;
18   &lt;div align="center"&gt;
19     &lt;div align="center"&gt;
20       &lt;table width="100%" border="0" cellspacing="0" cellpadding="0"&gt;
21         &lt;tr&gt;
22           &lt;td width="100%" valign="top"&gt;
23             &lt;table width="100%" border="0" cellspacing="0" cellpadding="0" align="center"&gt;
24               &lt;tbody&gt;
25                 &lt;tr&gt;
26                   &lt;td width="100%"&gt;&lt;a href="http://www.couleurverte.fr/tracking/cpc.php?ids=1664&amp;";
27                 &lt;/tr&gt;
28               &lt;/tbody&gt;
29             &lt;/table&gt;
30             &lt;td align="center" style="padding-top:20px"&gt;
31               &lt;h2&gt;&lt;font face="Arial, Helvetica, sans-serif" color="#003366" size="4"&gt;Convention
32               N&amp;deg;1 de l'assurance obs&amp;egrave;ques en direct&lt;/font&gt;&lt;/h2&gt;
33             &lt;/td&gt;
34           &lt;/tr&gt;
35         &lt;/tr&gt;
</pre>
<p><b>Start the test</b></p>
</div>
<div data-bbox="334 484 627 500" data-label="Caption">
<p>Figure 19 - Interface de création d'un kit email</p>
</div>
<div data-bbox="113 538 888 575" data-label="Text">
<p>La figure ci-dessous montre l'interface graphique qui permet au responsable de monétisation d'uploader les images relatives au kit d'email précédemment crée :</p>
</div>
<div data-bbox="115 580 878 872" data-label="Form">
<img alt="Screenshot of the Natexo admin interface showing the 'Ftp in Amazon S3' section. It includes a navigation bar, a header with 'Hello Admin', and a main area with a file upload box and a list of files and folders."/>
<p><b>Natexo</b> Hello Admin</p>
<p>Home Customers Reporting Statistics Synchro. Billing <b>Tools</b> Monetization</p>
<p><b>Ftp in Amazon S3</b></p>
<p>Drop files here or click for upload</p>
<p>+ Create a dir</p>
<p>http://assets.[domain]/OTHER/</p>
<ul>
<li>01.A.Usine</li>
<li>01_Voyance_FR</li>
<li>0_FR</li>
<li>1.jpg</li>
<li>1.png</li>
<li>1.zip</li>
<li>10 avril 2017 reyes</li>
<li>1000montad-es-1018</li>
<li>1001 edmain JB</li>
</ul>
</div>
<div data-bbox="305 890 697 905" data-label="Caption">
<p>Figure 20 - Interface pour uploader les images liées au kit crée</p>
</div>
<div data-bbox="134 936 384 965" data-label="Page-Footer">
<p><img alt="nextformation logo" data-bbox="134 936 175 965"/> nextformation | <img alt="Next group logo" data-bbox="335 936 384 965"/> Next group</p>
</div>
<div data-bbox="853 930 888 947" data-label="Page-Footer">
<p>32</p>
</div>
```


e) Lancement de la campagne :

Les plateformes Expert Sender accessible sur ce lien <https://expertsender.fr/> et mindbaz sont utilisées pendant cette étape afin de diffuser cette campagne par le canal d'email

f) Filtrage par le système de blacklist :

Cette étape est particulière car elle concerne uniquement une campagne d'emailing pour monétiser une base interne (voir définition I page 22).

g) Statistique :

Cette étape constitue un vrai défi pour le responsable de la monétisation ainsi les commerciaux de ce pôle, car leur première tâche chaque matin est de mettre à jour les bases de données Natexo par rapport aux routeurs ou les plateformes d'envoi utilisées pour la diffusion, pour cela un système est mis en place sur le projet admin monétisation afin de répondre ce besoin primordial qui constitue la phase initiale de la facturation de la monétisation.

La figure ci-dessous présente un exemple d'une interface graphique avec laquelle un commercial de monétisation mis à jour les bases de données interne par rapport à la base de données de nos partenaires

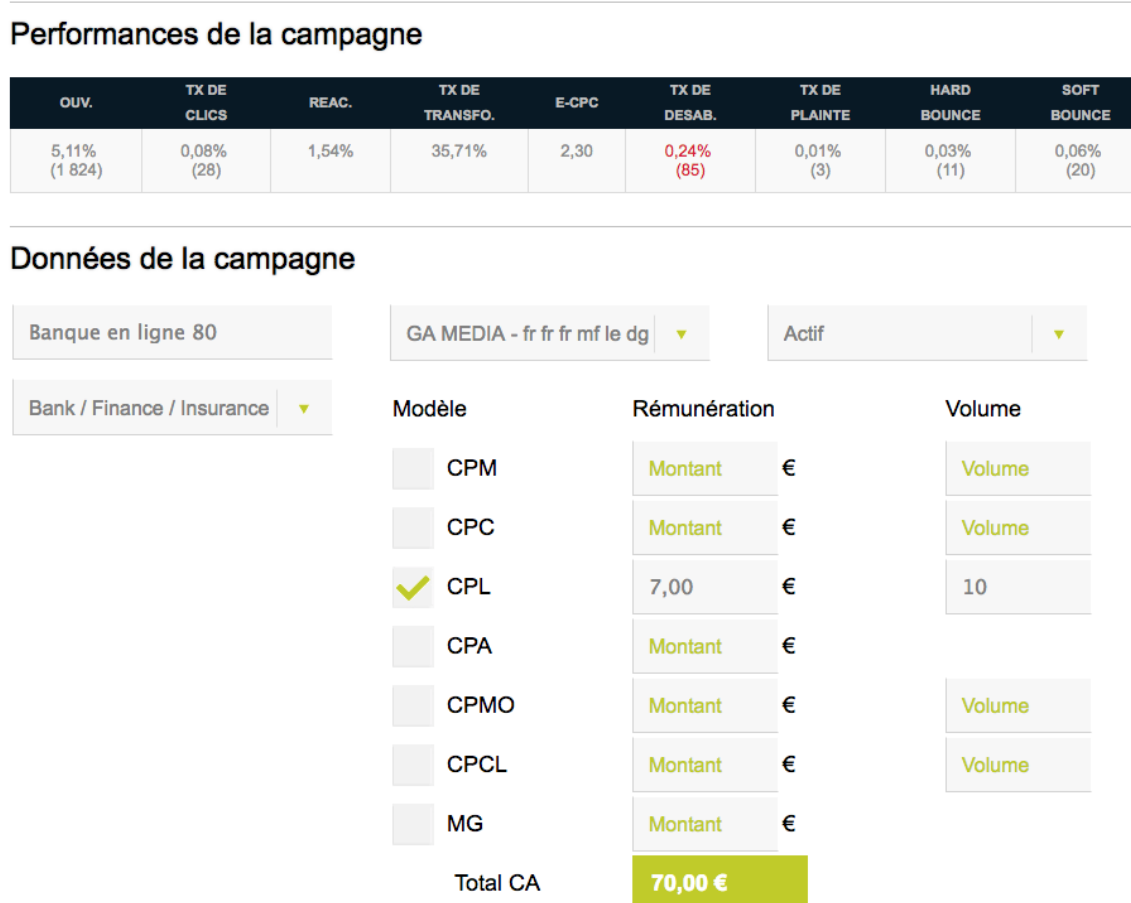


Figure 21 - Interface pour mettre à jour des bases de données local

h) Facturation :

Après avoir mis à jour les bases de données de la société par rapport au celles de nos partenaires, un système composé de plusieurs batch, tourne quotidiennement de manière continue procède à la génération des factures des clients ainsi la génération des AAF.

6- Diagramme fonctionnel de la Monétisation

La figure ci-dessous présente le diagramme fonctionnel de pole monétisation

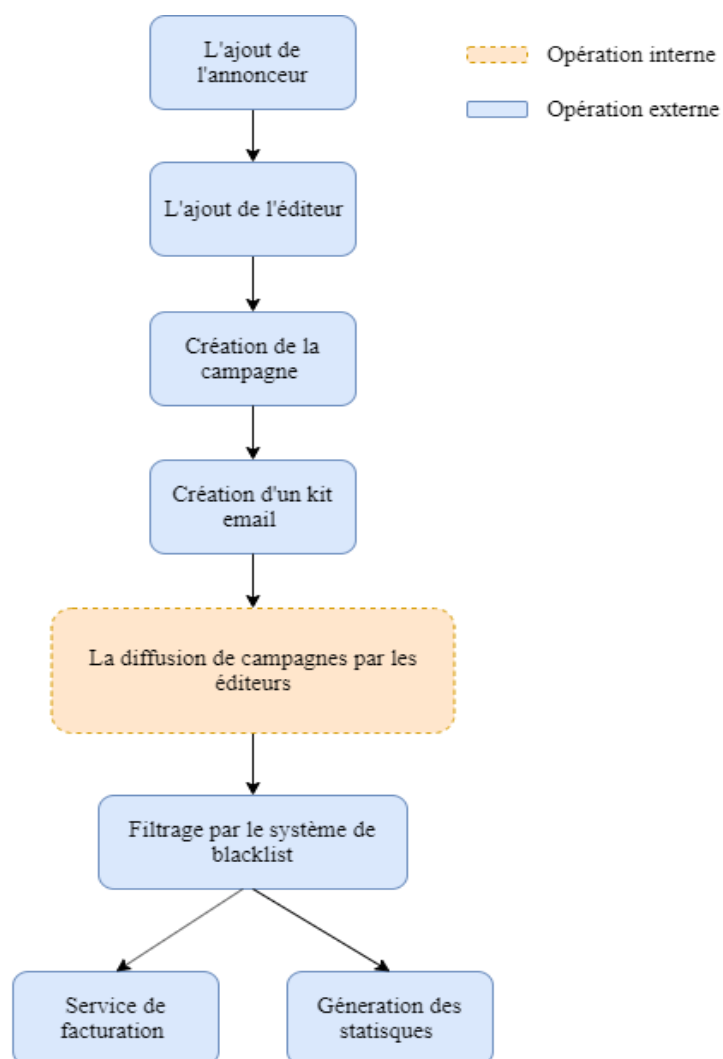


Figure 22 - Diagramme fonctionnel de la monétisation

7- Schéma d'architecture technique de la monétisation

Le schéma suivant présente une architecture technique de la monétisation

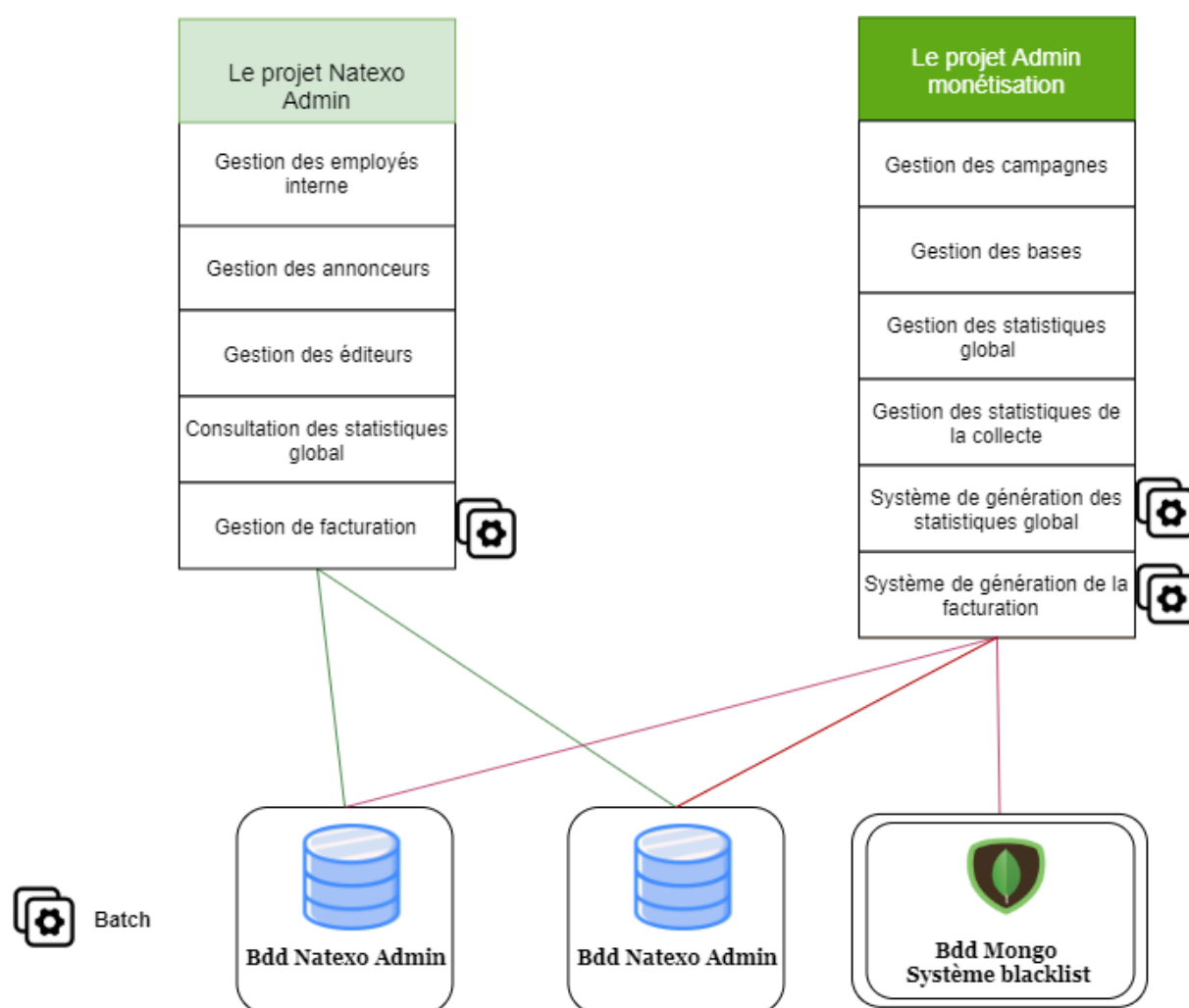


Figure 23 - Diagramme technique de la monétisation

C. Autres produits de la société

En plus des quatre plateformes principales de la société qui sont Natexo admin, admin collecte, landing page, et admin monétisation, l'entreprise possède d'autres produits. Nous présentons ceux sur lesquelles notre solution a un impact.

1- La plateforme des éditeur (Publisher) :

Disponible sur <http://publisher.natexo.com> cette plateforme a été développée pour améliorer le service de la collecte, elle permet à l'éditeur de choisir et de valider des campagnes parmi une liste les campagnes qui leur convient pour les diffusés ainsi les statistiques relatives à ses campagnes, un anglet de facturation et présent aussi cette plateforme.

D. Analyse de l'architecture existante

A l'heure actuelle Natexo possède plusieurs projets pour ses divers produits, depuis sa création en 2008 l'entreprise a opté pour PHP comme langage de programmation à utiliser ainsi le Framework Symfony, et le choix de l'architecture depuis sa création jusqu'à fin de l'année 2017 été une architecture monolithique.

Dans la suite de cette section sur l'analyse de l'architecture existence, nous allons aborder en détail les causes de choix de Natexo de ce monolithe, ses avantages pendant le lancement de cette société, on terminera par les inconvénients et les problèmes liés à cette architecture actuellement.

E. L'architecture monolithique

1- Définition

Une architecture monolithique est le modèle unifié traditionnel de conception d'un logiciel, la Monolithique , dans ce contexte, signifie composé en un seul morceau, un logiciel qui conçu dans cette architecture monolithique est fait pour être autonome, les composants du programme sont interconnectés et interdépendants plutôt que faiblement couplés

Voici un schéma explicatif d'une architecture monolithique :

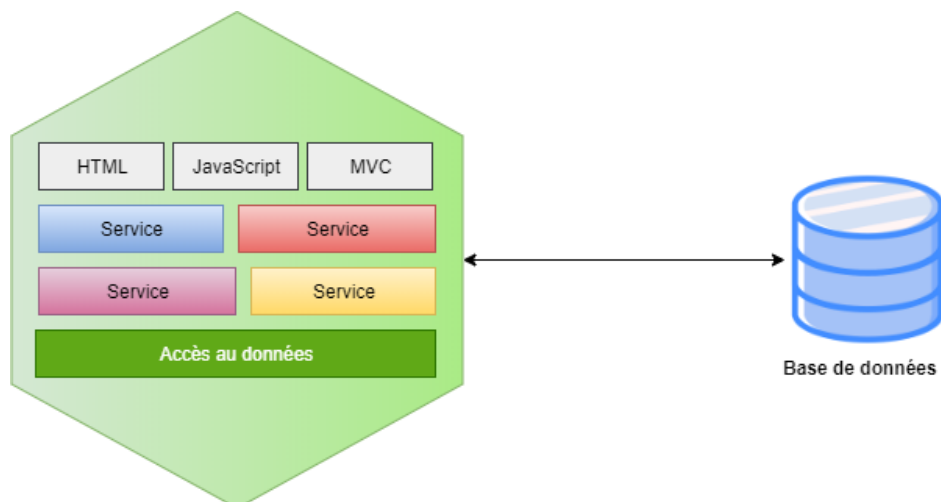


Figure 24 - Schéma de présentation d'une application monolithique

2- Avantages de cette architecture

Contrairement à ce que l'on pourrait penser, un monolithe n'est pas une architecture datée que nous devons laisser dans le passé. Dans certaines circonstances, une architecture monolithique peut être idéal pour répondre aux besoins des clients.

Voici quelques avantages de cette architecture :

- Moins de préoccupations transversales : Le principal avantage de l'architecture monolithique est que la plupart des applications ont généralement un grand nombre de préoccupations transversales, telles que l'enregistrement, la limitation du débit et les caractéristiques de sécurité telles que les pistes de vérification et la protection DOS. Lorsque tout fonctionne avec la même application, il est facile de connecter des composants à ces préoccupations transversales.
- Moins de frais généraux opérationnels : Avoir une application signifie qu'il n'y a qu'une seule application pour laquelle vous devez configurer la journalisation, la surveillance et les tests. De plus, il est généralement moins complexe à déployer.
- Performances : il peut également y avoir des avantages en termes de performances, car l'accès en mémoire partagée est plus rapide que la communication inter-processus (IPC). Actuellement ses systèmes isolés construire un vrai challenge pour l'équipe R&D (recherche et développement) de l'entreprise pour une extensibilité partielle d'un produit ou global des produits de Natexo, la partie suivante nous allons aborder les problèmes à cette architecture

3- Les problèmes de cette architecture

La section suivante présente les principales raisons pour changer et de s'orienter vers une architecture découpée facilement extensible. Alors voici les limites et les inconvénients de cette architecture existante :

- La répartition des services d'un pôle d'emailing de la société sur plusieurs plateformes (interface graphique) rend la tâche des commerciaux pour la gestion d'une campagne d'emailing plus complexe.
- Le problème de synchronisation et la mise à jour des bases de données lié aux dysfonctionnements des batchs mis en place (ex : génération des statistiques quotidiennes)
- La baisse de vitesse (le but initial de choisir cette architecture) à cause de la complexité,
- La dette technique en croissance.
- L'augmentation de la complexité de l'infrastructure
- La duplication de certains services sur plusieurs projets (service de facturation, service de déduplication etc.)
- Un inconvénient majeur de l'architecture et notamment pour Natexo qui possède actuellement un volume de données important repartir de ses diverses bases de données, donc on dit par là un traitement de données important et essentiel est de rendre une

application dépendante à certaines technologies, obligatoirement compatibles entre elles. Choisir Java limitera les évolutions futures aux technologies compatibles avec la JVM, C# aux technologies Microsoft, etc. Si un Framework utilisé par l'application devient obsolète, la migration vers un nouveau peut nécessiter la réécriture complète de l'application avec les risques que cela comporte.

- Les modifications sur l'application peuvent impacter d'autres modules
- Lorsque la charge d'une application augmente, l'extensibilité de l'application devient un point critique. Une application monolithique n'ayant qu'une capacité assez limitée on ne peut avoir que des copies de l'application entière. Or les différents composants de l'application n'ont vraisemblablement pas les mêmes besoins en ressource. Certains vont faire un usage intensif du processeur, d'autres de la mémoire. Une application monolithique peut entraîner une augmentation de la consommation des ressources conséquentes de manière inutile.
- Concernant l'évolutivité et fiabilité plus le temps passe, plus les nouvelles fonctionnalités métier deviennent complexes, et plus les différentes briques ont d'interactions. On a beau organiser le code en couches et en composants, il y a toujours des cas particuliers et des rustines qui rendent les choses plus floues. Au-delà d'un certain seuil, il devient impossible d'avoir en tête un modèle global du projet. Même avec une base de tests solide, la multiplication des effets de bord de chaque action rend le système moins fiable, et il devient alors plus difficile d'ajouter proprement des nouvelles fonctionnalités et d'effectuer des re-factorings (factorisation).
- Si en début de projet, la simplicité du modèle monolithique favorise grandement la compréhension du code, ceci devient de moins en moins vrai au fur et à mesure que le volume de code augmente, et ce, même avec une organisation rigoureuse de ce dernier. Un nouveau venu mettra davantage de temps à être pleinement opérationnel. Parce qu'il n'y a pas de frontières fortes entre les modules d'une application monolithique, elles auront tendance à s'effacer, affaiblissant la modularité, complexifiant les relations de dépendance. Le code devient de moins en moins lisible et testable, la productivité et la qualité baisse et enfin la dette technique s'accumule.
- Pour satisfaire l'innovation technologique et pour capitaliser les investissements et faciliter la gestion des personnes, il est normal de vouloir avoir une cohérence entre les différents projets d'une entreprise : même manière de travailler, mêmes langages de programmation, mêmes outils. Chaque projet est invité à suivre des choix transverses, et peut s'en écarter en fonction de ses spécificités, à condition de le justifier. Pour les gros projets, la même tension a lieu à l'intérieur même des projets : pour éviter la fragmentation, chaque évolution technique doit pouvoir être propagée à l'intégralité du code. Avec le temps, les modifications deviennent donc plus coûteuses, et il est plus difficile d'introduire de nouveaux outils pour des besoins spécifiques.

- Concernant innovation métier et pour répondre aux nouveaux besoins métier, il faut être capable de ménager une zone d'innovation à l'intérieur des projets. Car si certaines nouveautés sont mises en œuvre par de nouveaux projets, la plupart se font sur des projets existants. Or plus un projet est gros, plus il est critique pour l'entreprise, moins on va prendre de risques de le modifier pour tester de nouveaux produits ou de nouveaux marchés, et petit à petit les enjeux de stabilité vont prendre le pas sur la capacité d'innovation.

A. Analyse des besoin métier de changement du d'architecture

En plus de ces dénombrable inconvénients de l'architecture actuellement, depuis fin de l'année 2017 Natexo a décidé :

- Au long terme de fournir à ses employés interne une nouvelle application englobante de tous les services de la société avec un menu variable selon les droits d'accès de chaque utilisateur, ce nouveau projet est appelé Natexo App, à la fin de ce projets la société fermera les projets suivants :
 - Natexo admin : l'application web <http://admin.natexo.com>,
 - Admin collecte : qui correspond à la première application web de la collecte pour enquêtes et jeux (<http://collecte.admin.natexo.com>),
 - Admin monétisation : la plateforme de pôle monétisation (monetization.admin.natexo.com).
- Au moyen terme d'enrichir le pôle de la collecte par un nouveau produit appelé Leadgen programmatic,
- Au moyen terme aussi de fournir une plateforme à l'annonceur ainsi à l'éditeur afin qu'ils puissent gérer leur campagne eu même

B. Conclusion

Pour Natexo qui est une société en pleine croissance et en constante évolution et avec la sortie des nouvelles produits, le choix de passer par cette transition est été absolue et éminente vue au nombreux projet envisagé par cette entreprise,

Dans la section suivante (solution) nous allons voir en détail la définition de cette architecture, les technologies utilisés et les bonnes pratiques afin de la mettre en place et la planification suivi par l'équipe R&D de la société pour faire cette transition, en finissant par les tests unitaires et d'intégrations implémenté

VI. SOLUTIONS UTILISEES ACTUELLEMENT

Dans cette section, nous allons aborder deux types d'architecture logicielle les plus utilisées et susceptibles de remplacer l'architecture monolithique existante.

A. Architecture microservices

1- Définition de l'architecture

L'architecture microservices désigne une approche architecturale du développement d'applications. Une architecture de microservices se différencie d'une approche monolithique classique par le fait qu'elle décompose une application pour en isoler les fonctions clés. Chacune de ces fonctions est appelée « service » et ces services peuvent être développés et déployés indépendamment les uns des autres. Ainsi, chacun peut fonctionner sans affecter les autres.

2- Définition d'un microservice

C'est un service unique conçu pour prendre en charge une fonctionnalité de l'application. Dans le modèle de microservices, l'application représente l'agrégation de plusieurs services, chacun ayant un objectif spécifique. Par exemple, pour une application marchande, des services différents peuvent gérer respectivement le panier de clients, le service de paiement et l'interface avec une application backend pour le stock. Ces microservices doivent être faiblement couplés et s'interfacer via des API bien définies. Ils peuvent être écrits dans différents langages et frameworks, et comporter des cycles de vie différents.

La figure 25 présente un schéma explicatif de cette architecture.

MICROSERVICES

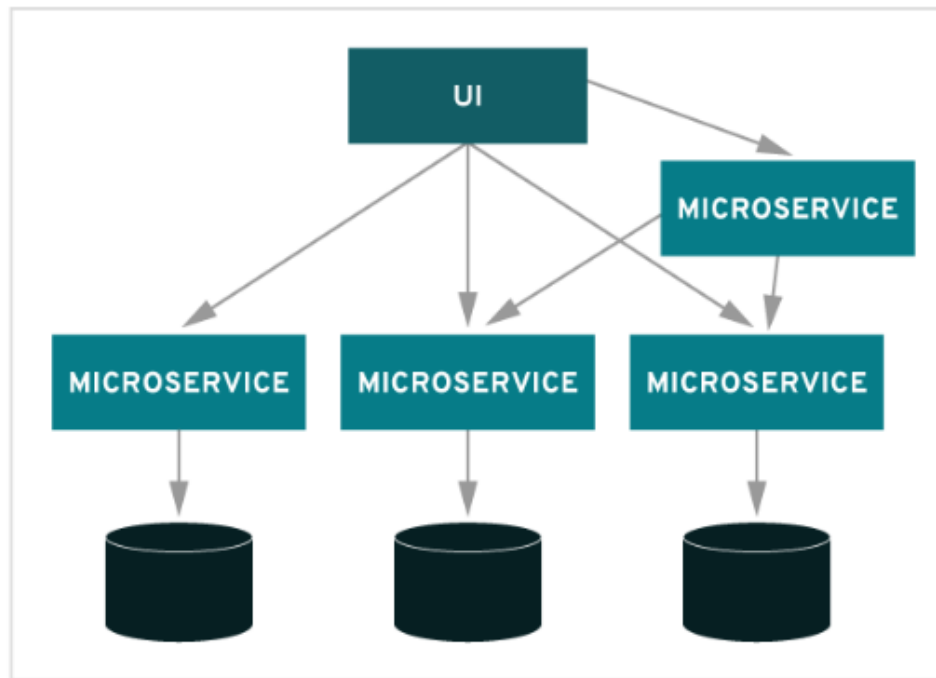


Figure 25 - Schéma explicatif de l'architecture microservices

3- Les avantages de MSA

Voici quelques avantages de cette architecture :

- **Agilité** : en décomposant les fonctionnalités au niveau le plus élémentaire, puis en résumant les services associés, il suffit de modifier et de redéployer les applications lorsqu'une modification est nécessaire dans l'ensemble de l'application.
- **Efficacité** : l'exploitation d'une architecture basée sur microservices permet une utilisation beaucoup plus efficace du code et de l'infrastructure sous-jacente, avec la possibilité de développer et de déployer des services de manière indépendante, sans attendre les décisions relatives à l'ensemble de l'application.
- **Résilience** : en répartissant les fonctionnalités sur plusieurs services, les applications ne sont plus exposées à un seul point de défaillance. Si l'un des services échoue, les autres continueront à fonctionner.

L'architecture des microservices semble être la solution idéale à suivre en raison de la décomposition de l'architecture en services à composants multiples, de sorte que chacun de ces services puisse être déployé puis redéployé de manière indépendante. Mais cela peut s'avérer trop complexe pour les besoins de certaines entreprises. Dans un autre cadre simplifié une architecture plus simple avec moins de contraintes et de règles à respecter et plus économique et plus adéquate à mettre en œuvre.

4- Les préoccupations de la migration vers MSA

Les investissements nécessaires à la migration de services existants et les risques potentiels font hésiter les entreprises plus modestes, qui n'ont pas forcément l'expertise et les connaissances à disposition.

Pourtant, les avantages des applications développées en microservices sont nombreux, notamment en termes de stabilité, de flexibilité et de maintenance. Décomposer des applications monolithiques lourdes en différents services métier, répondant chacun à des tâches bien distinctes, garantit un haut niveau de disponibilité des services aux utilisateurs, en revanche pour assurer la réussite de cette migration il faut que l'investisseur envisage des multiples changements sur différentes échelles de la société

Voici quelques adaptations organisationnelles et autres qui doivent accompagner cette transition :

- Les investissements sont lourds d'un point de vue humain comme financier. Découper une application monolithique en différents microservices services peut se mesurer en mois, à raison d'une ou plusieurs équipes de DEVOPS et de développeurs. Il faut pouvoir assumer cet investissement. Mais les bénéfices peuvent être nombreux sur le long terme : les temps de développement s'en trouvent réduits, il est plus simple de corriger les bugs et surtout, en cas de problèmes sur un service, il est possible de programmer à l'avance le basculement sur une version alternative afin que cela n'impacte pas le parcours client.
- Il est essentiel de former son équipe R&D ou de choisir un expert qui connaît et comprend les enjeux des différents métiers, leurs limites ainsi que les liens entre eux. Il lui sera ainsi plus facile d'extraire le code lié aux fonctionnalités à développer. Techniquement, il devra bien connaître les API et leur architecture, et être soucieux de bien tester son code. En termes de technologies, il devra maîtriser Kubernetes et surtout Docker, AWS et Google Cloud, ainsi que les technologies de monitoring, de log et de tracing.

B. Architecture SOA

1- Définition de cette architecture

Une Architecture Orientée Services est une architecture logicielle s'appuyant sur un ensemble de services simples. Ils sont développés en s'inspirant des processus métier de l'entreprise.

Un service est un composant fonctionnant de manière autonome et offrant des fonctionnalités métiers à d'autres applications ou d'autres services. Ces services représentent les fonctions basiques des fonctionnalités des entreprises. Ils dialoguent entre eux au travers de bus ou par Internet, on parle alors de Web service (WSOA). Les échanges peuvent se faire de manière synchrone ou asynchrone. L'entreprise s'enrichit de services mutualisables permettant de répondre rapidement et avec souplesse aux demandes du marché. En effet, ils correspondent à un processus métier mutualisable au niveau de l'entreprise. Cela permet les changements au niveau informatique des décisions stratégiques et tactiques de l'entreprise.

Il n'existe pas spécifications officielles pour l'architecture d'une SOA. Il peut être décrit par la notion, la description, la publication et l'invocation de services. La notion de service est une fonction encapsulée dans un composant que l'on peut interroger à l'aide d'une requête composée d'un ou plusieurs paramètres et fournissant une ou plusieurs réponses. Idéalement chaque service doit être indépendant des autres afin de garantir sa réutilisabilité et son interopérabilité. La description de service est la manière de décrire les paramètres d'entrée du service, le format et le type des données retournées. Le principal format de description de services est WSDL (Web Services Description Language), normalisé par le W3C. Ensuite la publication consiste à publier dans un registre (en anglais registry ou repository) les services disponibles aux utilisateurs, tandis que la notion de découverte recouvre la possibilité de rechercher un service parmi ceux qui ont été publiés.

La figure ci-dessous illustre le concept de l'architecture orientée service

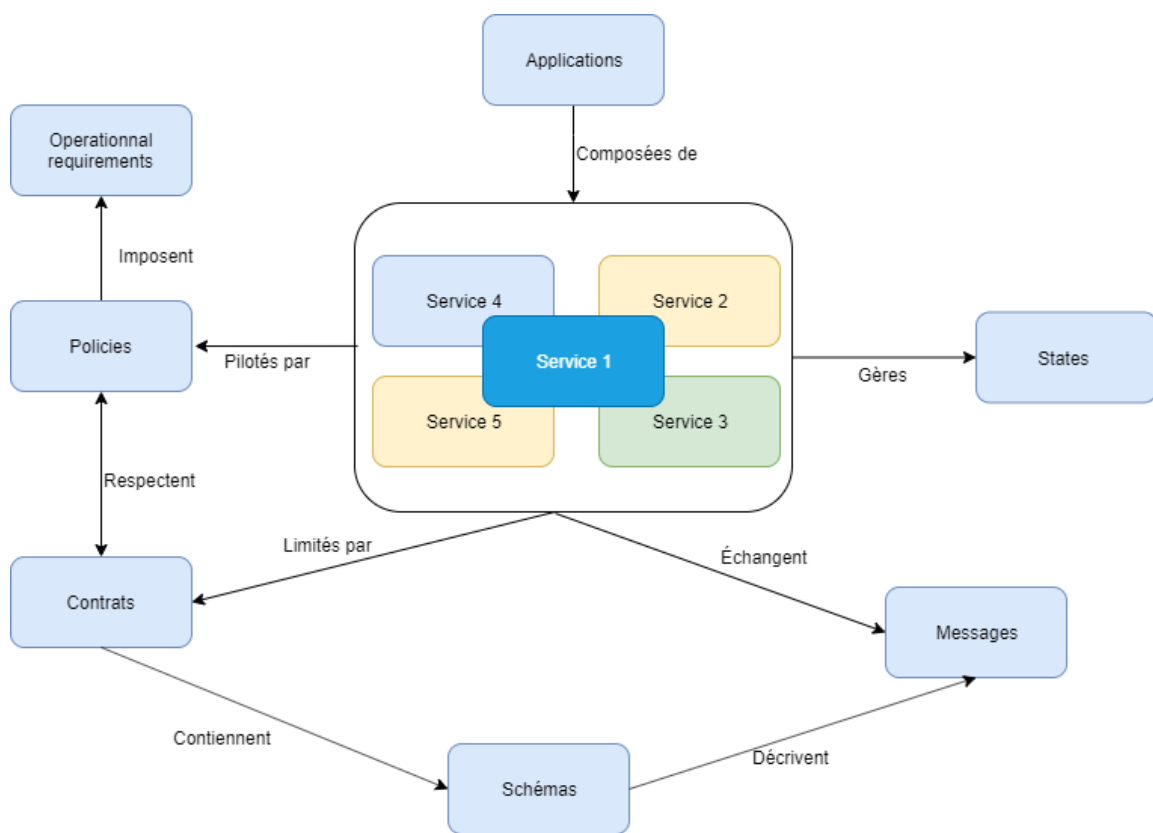


Figure 26 - Schéma expliquant le concept de SOA

Les services au centre de ce schéma (figure 19) sont les unités atomiques d'une architecture orientée services. Ces services communiquent entre eux par le biais de messages. Ces communications peuvent être synchrones ou asynchrones. Ces services sont indépendants du langage de programmation, de la plateforme. L'objectif étant l'interopérabilité. Ces services respectent des contrats (spécifiés parfois en WSDL) portant essentiellement sur les formats de données et des règles de composition.

2- Définition d'un service

Le service est un composant clé de l'architecture orientée services. Il consiste en une fonction ou fonctionnalité bien définie. C'est aussi un composant autonome qui ne dépend d'aucun contexte ou service externe. Il propose diverses opérations qui constituent autant d'actions spécifiques que le service peut réaliser. On peut faire un parallèle entre opérations et services d'une part, et méthodes et classes dans le mode orienté objet d'autre part.

Une architecture orientée services consiste une collection de services qui interagissent et communiquent entre eux. Cette communication peut consister en un simple retour de données ou en une activité (coordination de plusieurs services).

Un service est une entité de traitement qui respecte les caractéristiques suivantes :

- **Large granularité (coarse-grained)** : les opérations proposées par un service encapsulent plusieurs fonctions et opèrent sur un périmètre de données large au contraire de la notion de composant technique.
- **Interface** : un service peut implémenter plusieurs interfaces, et aussi plusieurs services peuvent implémenter une interface commune.
- **Couplage faible (loosely-coupled)** : les services sont connectés aux clients et aux autres services via des standards. Ces standards assurent le découplage, c'est-à-dire la réduction des dépendances. Ces standards sont des documents XML comme dans les web services.
- Synchrones ou asynchrones.

C. Comparaison entre les deux architectures

Dans cette section, nous allons voir les différences entre l'architecture microservices et architecture SOA. Nous débuterons par un schéma qui montrera le critère de granularité qui constitue un facteur majeur à prendre en compte lors de la décomposition d'un système existant en monolithe vers un ensemble de service. Pour cela, la figure 27 montre la manière de décomposition de chaque architecture. Nous terminerons cette comparaison par une analyse qui se fera sur les trois principaux de la gestion d'un projet informatique :

- La qualité et l'analyse technique de produit final,
- Les coûts
- Le délai de la réalisation de cette migration

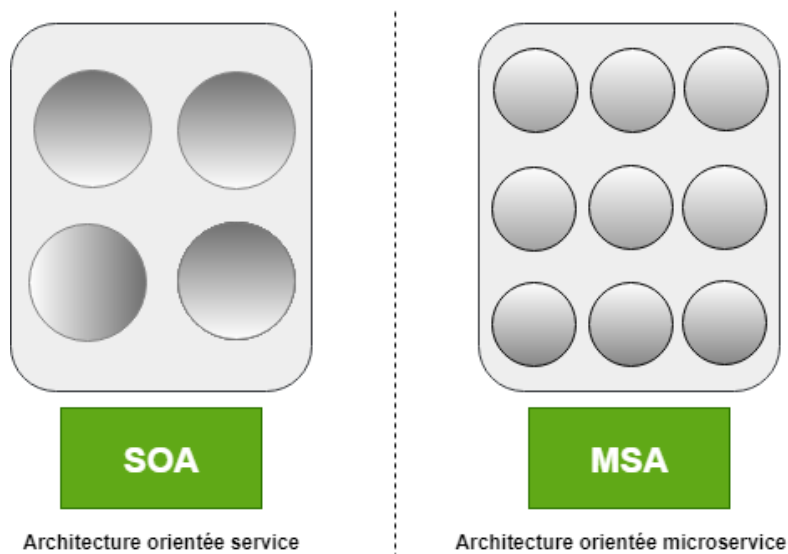


Figure 27 - Décomposition dans SOA et MSA

1- L'analyse de qualité de produit finale

La première colonne du tableau suivant correspond aux critères de qualité attendus de nouveau système d'information à la fin de cette migration et qui répond au problème présent sur l'architecture existante :

Analyse de qualité		
Critère	MSA	SOA
Partage du stockage de données	Une base de données est requise pour chaque microservices	Les services partagent le stockage de données
Taille et la portée du service	MSA est limitée par son objectif de responsabilité unique par service.	Une SOA est libre de prendre en charge plusieurs responsabilités au sein d'un service
Orchestrateur (exemple : Natexo app)	Dans une MSA, les services ne peuvent pas être composés d'autres services en raison de l'exigence d'indépendance.	La SOA complète permet de créer des services par le biais d'orchestrations.
Relation entre service	Un service doit être indépendant des autres services.	Aucune exigence d'indépendance n'est requise.

Réutilisation des services	Met l'accent sur le couplage faible	Maximise la réutilisabilité du service d'application
Système de communication entre services	Ils utilisent uniquement les protocoles légers tels que HTTP/REST	Ils supportent plusieurs protocoles de messagerie
Taille du projet	Efficace pour des projets de moyenne taille. Devient difficile à appréhender si l'on ne connaît le code	Permet de centraliser des briques métiers. Facilite l'appréhension et la communication entre les différentes briques

Tableau 1 - Comparaison entre MSA et SOA selon la qualité de produit attendu

2- L'analyse des coûts de la réalisation

MSA	SOA
<ul style="list-style-type: none"> - Pour Natexo un investissement au point de vue humain est essentiel pour assurer une meilleure réalisation 	<ul style="list-style-type: none"> - L'équipe R&D est la seule dans la société et elle est composée de quatre développeurs

Tableau 2 - Comparaison entre SOA et MSA selon les coûts de la réalisation

3- L'analyse des délais de la réalisation

MSA	SOA
<ul style="list-style-type: none"> - Mise en place de l'environnement de production peut être lent et complexe - Évolutivité plus rapide - La limitation de la portée des services peut être coûteuse en termes de temps car le découpage en plusieurs microservices peut s'étaler sur plusieurs mois - Former son équipe R&D pour la maîtrise théorique et pratique de cette architecture pourra prolonger les délais de livraison 	<ul style="list-style-type: none"> - Mise en place de l'environnement de production peut être rapide et facile - Évolutivité plus lente et dépendra de la complexité des services existant - La souplesse lors de la décomposition peut être facile et le « re-factoring » peu prendre moins de temps - Formation non requise pour cette architecture

Tableau 3 - Comparaison entre SOA et MSA selon les délais

D. Conclusion

Finalement, il n'est pas aussi simple de dire quelle architecture est meilleure qu'une autre. Cela dépend principalement du but de l'application à construire et des besoins. La SOA convient mieux aux environnements d'applications d'entreprise **complexes** et de **grande taille**

nécessitant l'intégration de nombreuses applications hétérogènes, et la migration vers SOA depuis de telle système complexe est moins difficile en revanche les petites applications ne conviennent pas à la SOA car elles ne nécessitent pas de composant middleware. Les microservices, en revanche, conviennent mieux aux systèmes Web plus petits et bien partitionnés, dans lesquels les microservices offrent un contrôle beaucoup plus important en tant que développeur.

La conclusion est que, comme ils ont tous les deux des caractéristiques d'architecture différentes, cela dépend principalement de la fonction de l'application créée.

VII. ARCHITECTURE CIBLE

A. La transition

Afin de comprendre cette transition, nous allons commencer par la présentation fonctionnelle de nouveau produit Leadgen programmatic, car cela a permis au Product owner et son équipe ainsi l'équipe R&D d'identifier, et de fixer des priorités sur l'ensemble des systèmes qui sont en commun entre l'ancien produit de la collecte Survey et jeux et le nouveau projet Leadgen programmatic, le tableau présent dans la section suivante (voir la page 52) montre en détail ses services ainsi leur priorité.

1- Leadgen Programmatic

Le fonctionnement de ce nouveau produit est semblable l'ancien produit existant la collecte de Survey et jeux, ce nouveau produit est composé de deux systèmes distingue :

- Le premier système d'information englobe l'application back office de ce produit ainsi une plateforme pour les éditeurs et les annonceurs, afin de diffuser une campagne emailing en premier lieu un responsable de la collecte possède à l'ajout de l'éditeur ainsi l'annonceur dans nos bases de données après une demande de ce dernier, ensuite cet annonceur procéder à la création de son compte pour qu'il puisse ajouter des *ads* (voir la page 22) et des campagnes, ces campagne sont suggérer aux éditeurs, le schéma ci-dessous

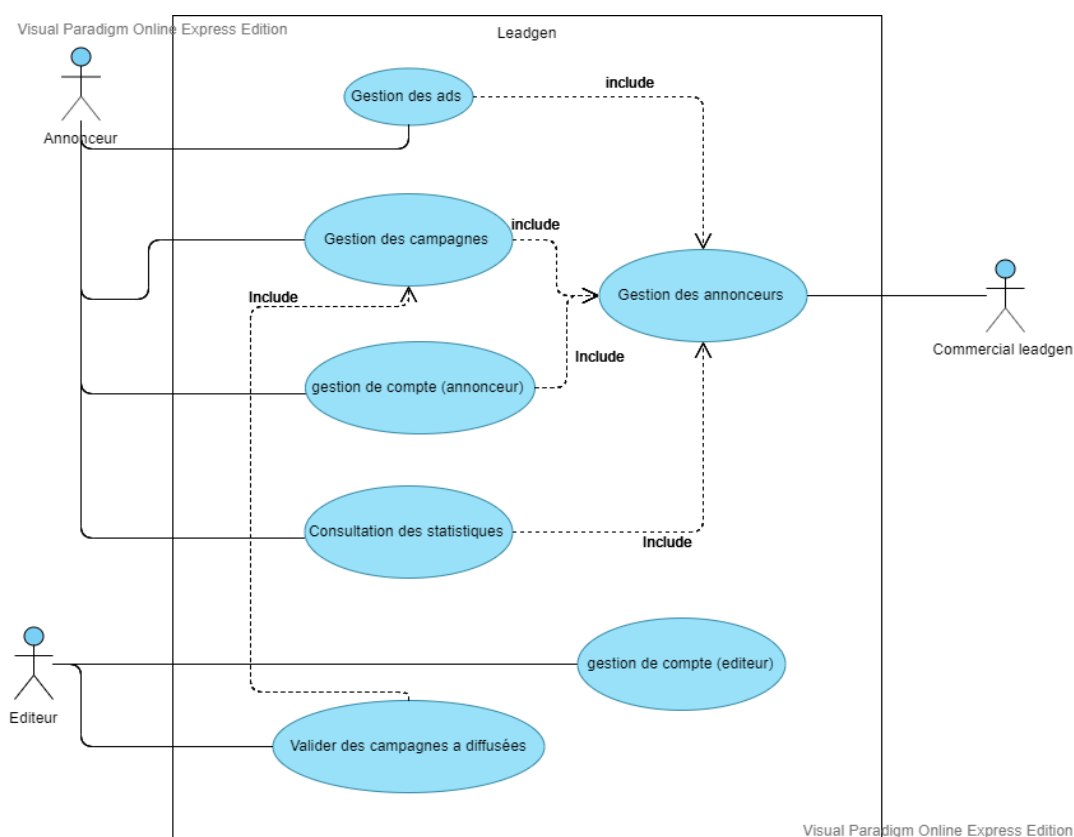


Figure 28 - Diagramme de cas d'utilisation de Leadgen

- Le second système constitue la plateforme de génération d'Ad ainsi le processus de la collecte, la figure suivante présente le diagramme fonctionnel de ce système.

Le schéma ci-dessous présente un diagramme de cas d'utilisation de ce nouveau produit avec les fonctionnalités principale

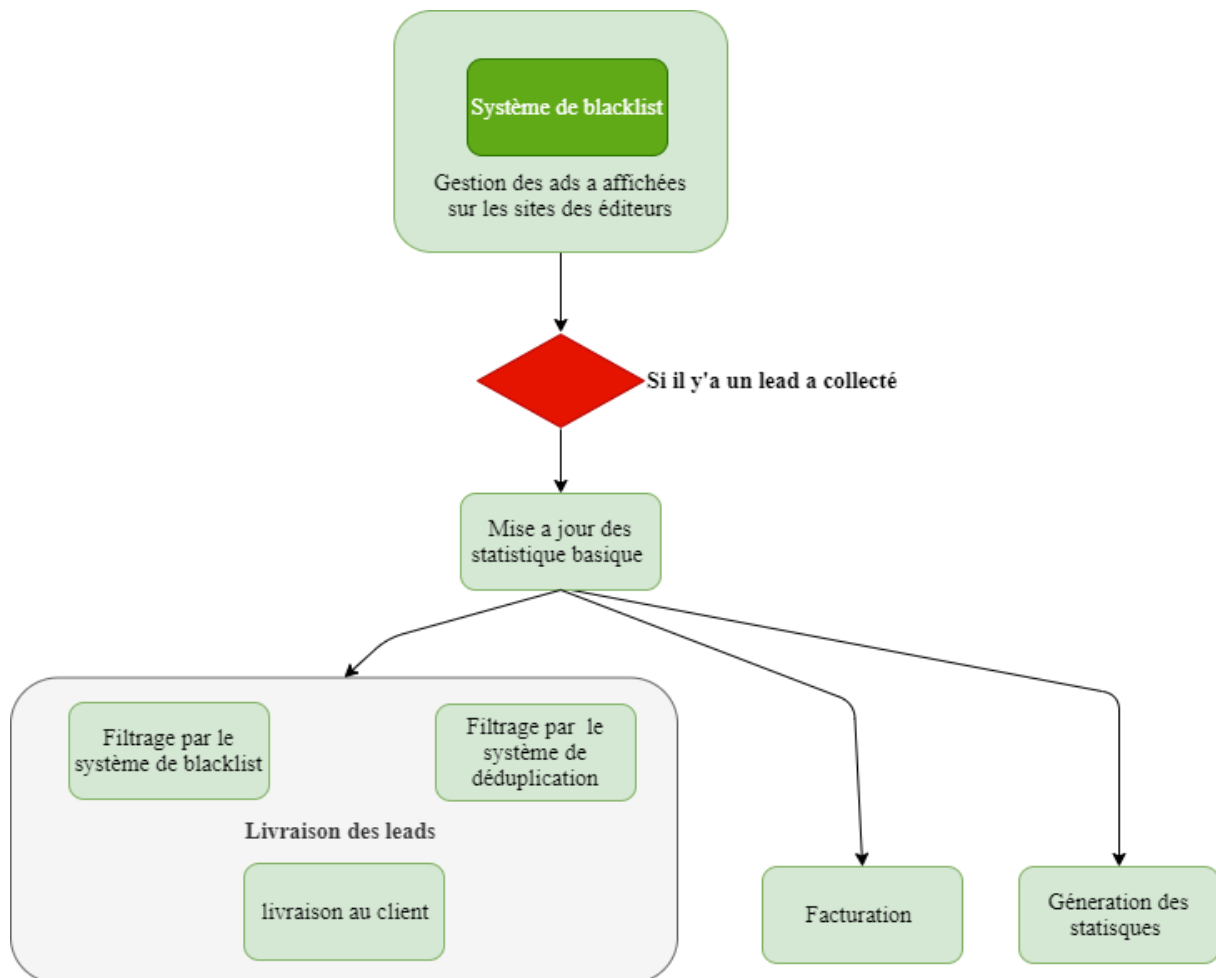


Figure 29 - Diagramme fonctionnel de la plateforme collecte de Leadgen

La première étape dans ce schéma consiste à gérer l'apparition des *ads* sur un site d'un éditeur selon un internaute et les critères définis lors de la création de la campagne correspondante, cette étape est assurée par le système.

1- Définition d'un web service

Appelé WSOA pour (Webservices Oriented Architecture), les Webservices sont des services disponibles via Internet par le biais d'URL. Pour les WSOA, les données sont stockées sur le serveur. On peut se connecter aux Webservices à l'aide de terminaux mobiles. L'utilisateur dispose donc à partir d'un client léger la puissance de calcul du serveur. La connexion à distance est réalisée par protocole HTTP, on a donc une indépendance de tout langage de programmation

et plateformes. La communication entre les différents acteurs du système est réalisée par la technologie XML ou par JSON

2- Définition de REST

La conception de l' API REST ou RESTful (Representational State Transfer) est conçue pour tirer parti des protocoles existants. Bien que REST puisse être utilisé sur presque tous les protocoles, il tire généralement parti de HTTP lorsqu'il est utilisé pour les API Web. Cela signifie que les développeurs n'ont pas besoin d'installer de bibliothèques ou de logiciels supplémentaires pour tirer parti d'une conception d'API REST. REST API Design a été défini par le Dr. Roy Fielding. Il est remarquable pour son incroyable niveau de flexibilité. Étant donné que les données ne sont pas liées à des méthodes ni à des ressources, REST peut gérer plusieurs types d'appels, renvoyer différents formats de données et même modifier la structure de manière structurée avec la mise en œuvre correcte de l'hypermédia.

Cette liberté et cette souplesse inhérentes à la conception des API REST vous permettent de créer une API répondant à vos besoins tout en répondant aux besoins de clients très divers. Contrairement à SOAP , REST n'est pas contraint au XML, mais peut renvoyer XML, JSON, YAML ou tout autre format, en fonction de la demande du client.

La figure 30 présente un schéma explicatif d'un service web

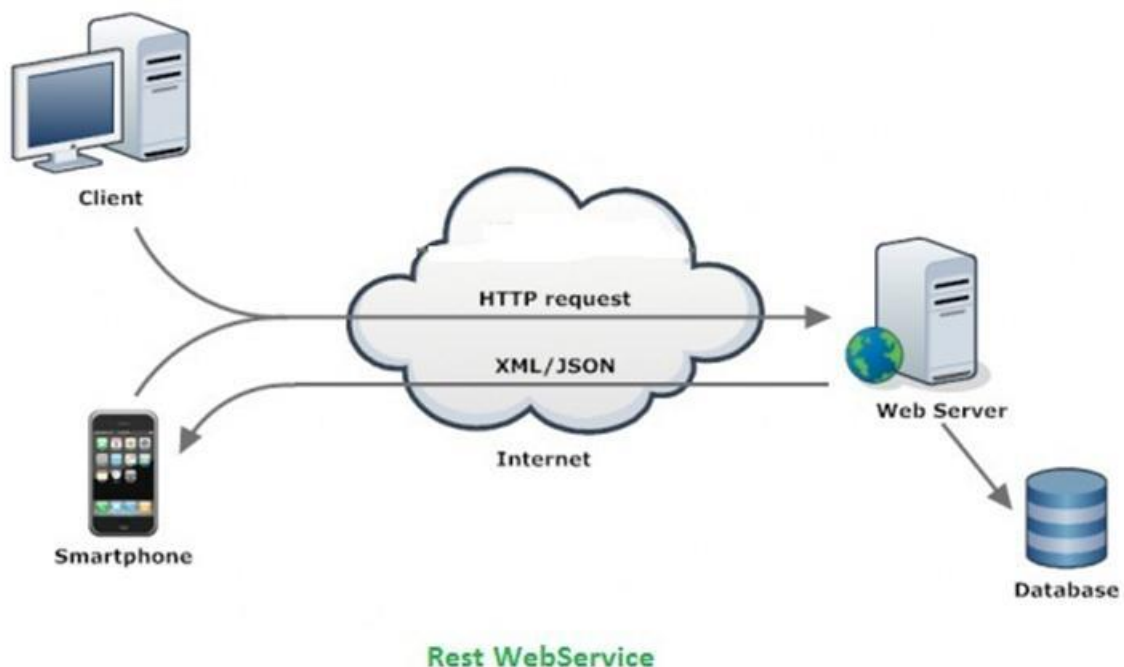


Figure 30 - Schéma explicatif de concept des webservices REST

3- La planification

Avant d'entamer les détails techniques de cette transition, nous allons tout d'abord commencer par détailler la planification mise en place par le Product owner et son équipe en coordination avec l'équipe R&D et le PDG de la société ainsi les responsables des deux pôles existants, à la fin cette planification initiale deux cahiers des changes ont été rédigés, le premier concerne la nouvelle application web nommé NatexoAPP qui remplacera au long terme le projet de la société Natexo , le second cahier des charges concerne le nouveau produit Leadgen programmatic, présenté en détail dans la section précédente.

Grace a cette analyse de besoin et ses cahiers des charges réalisés, un ensemble de priorité ont été fixe sur chaque fonctionnalité a réalisé, le tableau suivant illustre la fonctionnalité a réalisé ainsi leur priorité

Tableau 4 - Liste des services a implémenté

Catégorie	Service	Produit concerné	Priorité
Services en commun entre les produits Collecte, Leadgen et monétisation	Service d'authentification avec OAuth	Concerné tous les produits	Elevé
	Service de déduplication	Leadgen	Elevé
		Collecte	Moyen
	Service de livraison de lead	Leadgen	Elevé
		Collecte	Moyen
	Service de facturation	Leadgen	Elevé
		Collecte	Moyen
		Monétisation	Moyen
Nouveaux services (Tâches personnel)	Service de blacklist	Concerné tous les produits	Elevée
	Service de statistiques	Leadgen	Elevé
	Service des traces	Leadgen	Moyen

B. Réalisation

1- Système de facturation

Ce système a été transformé vers un webservice REST après avoir été un ensemble d'actions et de méthodes repartis sur les anciennes applications. Ce nouveau système est composé de deux services :

- Un premier service chargé de générer les différents types de facture (facture client ou Aaf) pour le nouveau produit (Leadgen programmatic) et le pôle collecte (Survey et jeux),
- Un second service chargé de générer la facturation de la monétisation.

La figure ci-dessous présente un schéma technique de ce système :

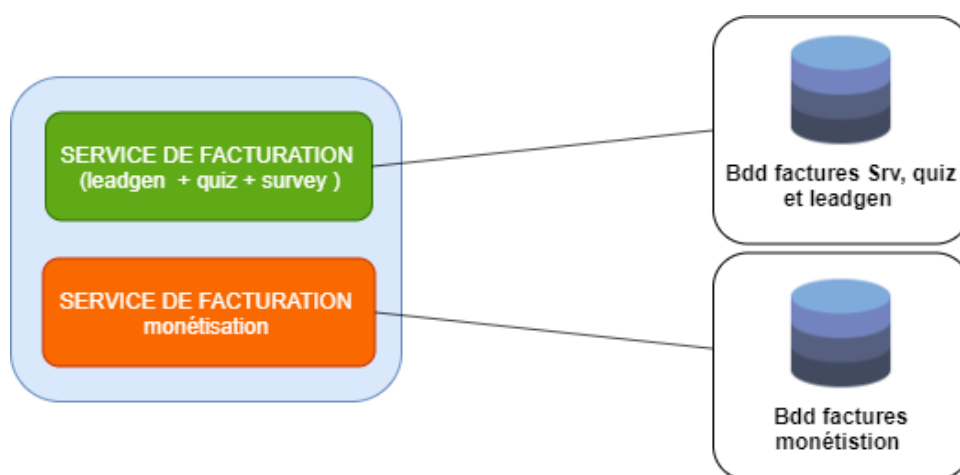


Figure 31 - Schéma technique de système de facturation

2- Système de déduplication et livraison des leads

Pour rappel le système de déduplication a pour objectif de garantir l'unicité des leads vendu pour certains de nos clients (Annonceur qui ne possède pas de système de vérification), et le système de livraison assure l'envoi des leads à nos clients, ces deux services ont été développés séparément et regroupés dans le même service web avec celui de la facturation

La figure 32 présente un schéma technique de service de déduplication ainsi le service de livraison

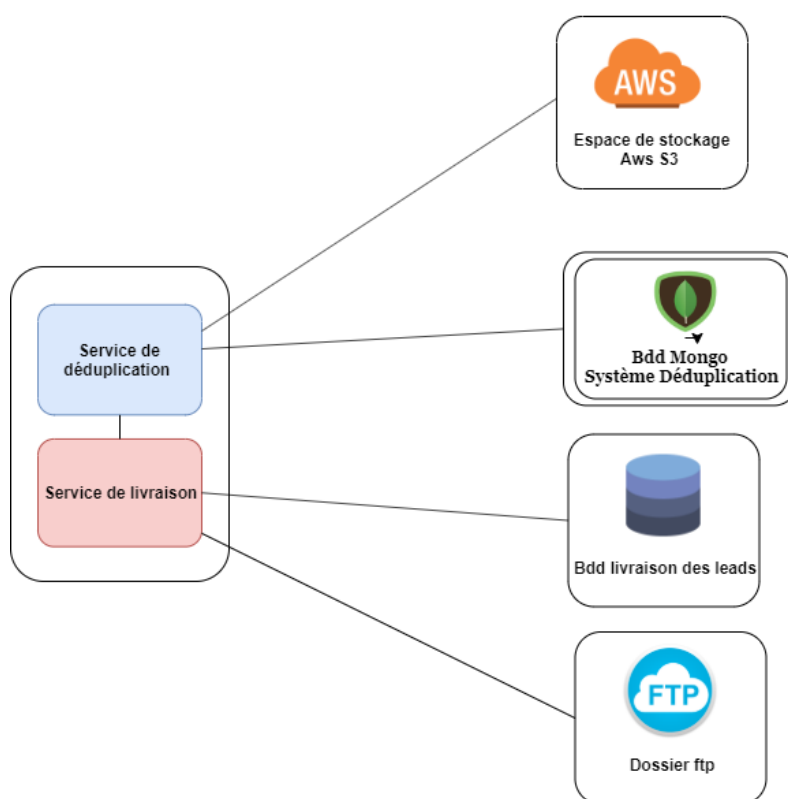


Figure 32 - Diagramme technique de Service de livraison

3- Système de blacklist

Avant cette transition, ce service été dupliqué sur les trois projets différents (les projets landing, admin collecte et admin monétisation). Actuellement, toutes les fonctionnalités de ce service ont été regroupée dans le même projet transformé en librairie PHP. Le but est que ce package soit réutilisé dans les anciens et les nouveaux produits. La figure ci-dessous présente un schéma technique de cette librairie.

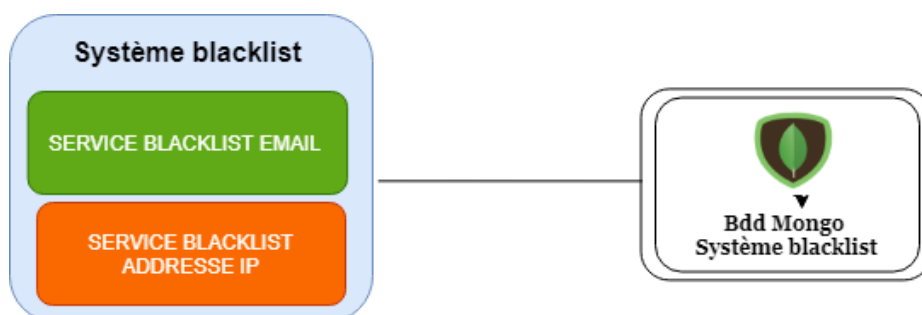


Figure 33 - Diagramme technique de nouveau système blacklist

4- Service d'authentification avec OAuth

Afin de garantir une communication sécurisée entre les webservice implémentés, un service d'authentification basé sur le protocole OAuth a été mis en place dans le webservice orchestrateur dans le projet Natexo App qui est censé remplacer au long terme l'ancien projet « Natexo admin ».

L'autorisation via mot de passe (Resource owner Password Credentials Grant) est le principe sur le quel ce service a été fait car un utilisateur (utilisateur interne) doit s'authentifier pour avoir un token d'accès (Access Token).et cet utilisateur peut utiliser ce token d'accès auprès du serveur de ressources qui correspond aux autres webservices.

5- Service des traces

Ce service implémenté en tant qu'une librairie PHP a pour objectif d'enregistrer les tâches importantes effectuées par les différents utilisateurs (utilisateur interne ou annonceur ou éditeur).

6- Service de statistiques (Leadgen)

Ce service est composé d'un ensemble de worker asynchrone ou batch qui calculer à partir de statistiques basiques des nouvelles statistiques complexes de différentes formules qui seront par la suite stocker dans des collections MongoDB.

Ces statistiques sont visualisées sur le projet front de Natexo app.

C. Choix des technologies

a) Le langage programmation PHP

- Définition :

Le PHP, pour HyperText Preprocessor, désigne un langage informatique, ou un langage de script, utilisé principalement pour la conception de sites web dynamiques. Il s'agit d'un langage de programmation sous licence libre qui peut donc être utilisé par n'importe qui de façon totalement gratuite.

Créé au début des années 1990 par le Canadien et Groenlandais Rasmus Lerdorf, le langage PHP est souvent associé au serveur de base de données MySQL et au serveur Apache

- Fonctionnement :

D'un point de vue exécution, PHP a besoin d'un serveur Web pour fonctionner. Toutes les pages demandées par un client seront construites par le serveur Web, en fonction des paramètres transmis, avant d'être retournées au client. Le schéma suivant illustre le principe de fonctionnement de PHP.

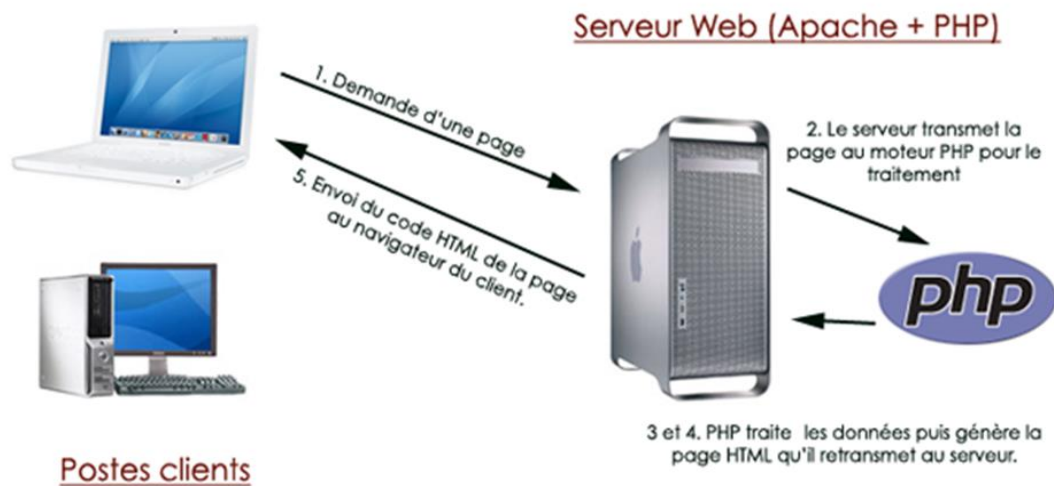


Figure 34 - Schéma de fonctionnement de PHP

- Les raisons pour choisir PHP :

- Par rapport à Natexo, PHP est un langage créé pour le développement web contrairement à C/C++ ou JAVA, il est entièrement web, en 2002, PHP est utilisé par plus de 8 millions de sites Web à travers le monde, en 2007 par plus de 20 millions¹² et en 2013 par plus de 244 millions, une raison valable pour la société de choisir ce langage puisque ses services sont fournis par le web

La figure ci-dessus présente la répartition des langages de programmation côté serveur, des sites Internet en 2016



Figure 35 - Répartition des langages de programmation côté serveur web en 2016

- La gratuité de PHP.
- Une communauté de développeurs partageant sans cesse des milliers de scripts de ce langage, cela permet aux développeurs de rester concentrés sur le développement des fonctionnalités.
- Les scripts en PHP sont très simples à comprendre même pour les développeurs les plus novices, alors cela permet de gagner en termes de temps pour maintenir l'existant et répondre aux nouvelles demandes et satisfaire le cahier des charges constamment variable et instable.
- Flexibilité des bases de données car PHP est flexible pour la connectivité des bases de données. Il peut se connecter à plusieurs bases de données dont la plus utilisée est la base de données MySQL, et ce dernier peut être utilisé gratuitement. Si ASP est utilisé, MS-SQL, un produit Microsoft doit être acheté.
- L'existence de nombreux frameworks avec une communauté forte (Symfony, Laravel). Différentes fonctionnalités sont disponibles sous forme de package ou peuvent être facilement assemblées en combinant les bibliothèques disponibles.
- Disponibilité de l'hébergement car presque tous les fournisseurs d'hébergement prennent en charge PHP, tandis que la prise en charge d'autres langages de programmation n'est pas souvent trouvée sur tous les hébergements.

b) Le Framework Symfony

- Définition :

Symfony est un framework français open source pour PHP qui représente un ensemble de composants (aussi appelés bibliothèques) PHP autonomes qui peuvent être utilisés dans des projets web privé ou open source. Mais c'est également un puissant Framework PHP édité par l'agence web SensioLabs et dont le créateur est Fabien Potencier. Il a également été écrit par toute une communauté. Sa première version est sortie en 2005 et est aujourd'hui toujours utilisée. La deuxième version est sortie en août 2011 et bien que différente dans sa conception, elle est plus rapide et plus souple que la première. Une troisième version de Symfony est sortie le 30 Novembre 2015. Il permet de réaliser des sites internet dynamiques de manière rapide, structurée, et avec un développement clair. Les développeurs peuvent travailler sur ce Framework très facilement, seuls ou en équipe, grâce à la facilité de prise en main.

- Les avantages de Symfony :

Depuis sa première version Symfony propose :

- Une séparation du code en trois couches, selon le modèle MVC, pour une plus grande maintenabilité et évolutivité
- Des performances optimisées et un système de cache afin d'assurer des temps de réponse optimaux
- Une gestion des url parlante, permettant à une page d'avoir une URL distincte de sa position dans l'arborescence
- Un système de configuration en cascade utilisant pleinement le langage YAML
- Persistance de données simplifiée avec l'ORM Doctrine
- Un générateur de back-office et un lanceur de module (scaffolding)
- L'internationalisation native
- L'utilisation des différents de design patterns comme MVC et singleton
- Une architecture extensible permettant créations et utilisations de plugin
- Symfony est maintenant un écosystème stable qui est à la fois bien connu et reconnu par les professionnels du monde entier il possède une grande communauté active de passionnés. Développeurs, tous participent à l'enrichissement de l'outil. C'est un véritable écosystème qui s'est développé autour de ce Framework depuis son arrivée
- L'organisme propriétaire de ce Framework (SensioLabs) donne une grande importance à l'évolution et l'intégration de nouvelles fonctionnalités de façon à répondre aux

besoins des marchés. Sans oublier la grande communauté de développeurs qui entoure l'application.

- Symfony respecte les normes existantes de PHP et PHP Unit, Dans ce sens, elle permet aux développeurs d'utiliser des segments de son Framework et de les intégrer dans d'autres écosystèmes de développement sans que cela ne pose des problèmes de compatibilité.

La figure ci-dessous présente d'un diagramme d'évolution des framework PHP ainsi l'utilisation de Symfony depuis 2013.

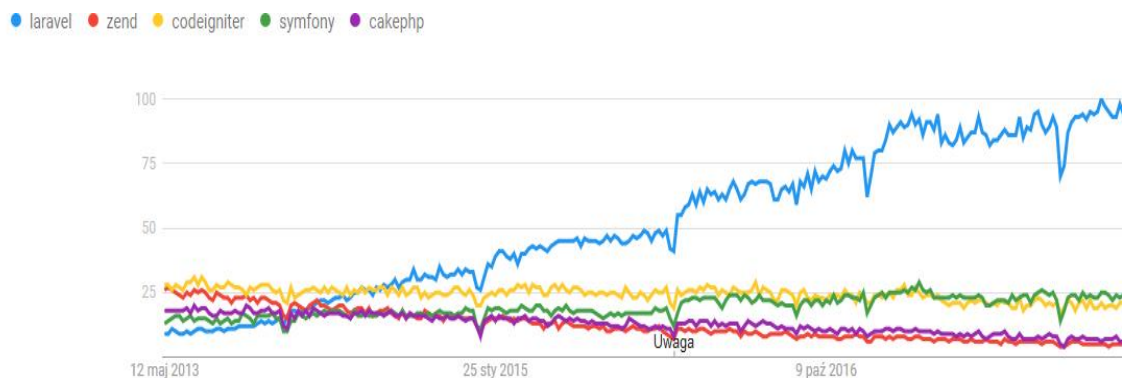


Figure 36 - Diagramme d'évolution des framework PHP

c) FOSRestBundle

FOSRestBundle fournit est un bundle tiers qui permet de créer facilement des applications REST et il permet aussi de générer les routes automatiquement. La Documentation est présente ici <https://github.com/FriendsOfSymfony/FOSRestBundle>

d) JMSSerializerBundle

C'est un bundle qui permet de Sérialiser et désérialiser facilement des données de toute complexité, il support trois format différent XML, JSON et YAML, la documentation officiel est accessible dans le lien suivant <https://github.com/schmittjoh/JMSSerializerBundle>

e) NelmioApiBundle

NelmioApiDocBundle permet de générer de la documentation au format Open API (Swagger) et fournit des interfaces graphiques permettant d'expérimenter de manière interactive avec l'API, sa documentation officielle est présente ici <https://github.com/nelmio/NelmioApiDocBund>

f) MySQL

- Définition :

MySQL est un système de gestion de base de données (SGBD). Issu du monde libre, il est l'un des logiciels de gestion de base de données le plus utilisé au monde.

MySQL est un serveur de base de données relationnelles SQL, il est multithread (peut exécuter plusieurs processus en même temps) et multi-utilisateur qui fonctionne aussi bien sur Windows que sur Linux ou Mac OS. Les bases de données sont accessibles en utilisant de nombreux langages serveur.

- Les avantages de MySQL :

MySQL est une base de données open-source gratuite à utiliser qui facilite la gestion efficace des bases de données en les connectant au logiciel. C'est une solution stable, fiable et puissante avec des fonctionnalités avancées telles que :

- En migrant les applications de base de données actuelles vers MySQL, les entreprises réalisent d'importantes économies de coûts sur les nouveaux projets. La fiabilité et la facilité de gestion qui accompagnent MySQL vous permettent de gagner du temps en matière de dépannage, ce qui est une perte de temps pour la résolution des problèmes de temps d'arrêt et de performances.
- Le temps moyen de téléchargement et d'installation étant inférieur à 30 minutes, MySQL signifie une facilité d'utilisation dès le premier jour. Que votre plate-forme soit Linux, Microsoft, Macintosh ou UNIX, MySQL est une solution complète dotée de fonctionnalités d'autogestion permettant d'automatiser tout, de l'extension et la configuration de l'espace à la conception de données et à l'administration de bases de données.
- Toutes les craintes et tous les soucis liés à une solution open source peuvent être résolus grâce au support 24 heures sur 24 et à l'indemnisation des entreprises de MySQL. Le traitement sécurisé et le logiciel sécurisé de MySQL se combinent pour fournir des transactions efficaces pour les projets de grande envergure. Il facilite et accélère la maintenance, le débogage et les mises à niveau tout en améliorant l'expérience de l'utilisateur final
- MySQL est mondialement reconnu pour être le système de gestion de base de données le plus sécurisé et le plus fiable utilisé dans les applications Web populaires telles que WordPress, Drupal, Joomla, Facebook et Twitter. La sécurité des données et la prise en charge du traitement transactionnel, qui accompagne la version récente de MySQL, peuvent grandement bénéficier à toute entreprise, en particulier s'il s'agit d'une entreprise de commerce électronique qui implique des transferts d'argent fréquents.
- MySQL propose une structure de moteur de stockage distincte qui permet aux administrateurs système de configurer le serveur de base de données MySQL pour des

performances sans faille. Que ce soit un site Web de commerce électronique qui reçoit un million de requêtes chaque jour ou un système de traitement transactionnel à grande vitesse, MySQL est conçu pour répondre aux applications les plus exigeantes tout en garantissant une vitesse optimale, des index de texte intégral et des caches de mémoire uniques pour des performances améliorées.

g) MongoDB

MongoDB est une base de données open source qui utilise un modèle de données orienté document.

Pendant la migration et la réalisation de nouveau projet Leadgen programmatic, MongoDB a été utilisé dans plusieurs services à cause de ses divers avantages citant :

- Open source
- MongoDB est une base de données conçue pour le stockage de données volumineuses
- MongoDB tire ses performances principalement de la conception clé basée sur la valeur et facile à déployer, cet avantage a été décisif afin de résoudre les problèmes des pages de statistique lente à charge dans les anciens projets.
- MongoDB n'impose pas de schéma, les formats variés des leads un tel avantage doit être pris en compte pour réaliser le système de blacklist

h) Satis

C'est un produit open source d'hébergement de paquets, ou un générateur de référentiel statique. C'est une version ultralégère et statique de Packagist, basée sur des fichiers, qui peut être utilisée pour héberger les métadonnées des packages privés d'une société. La documentation officielle de satis est disponible sur <https://github.com/composer/satis>

i) Framework Angular

Angular est un framework Javascript côté client qui permet de réaliser des applications de type « Single Page Application ». Il est basé sur le concept de l'architecture MVC (Model View Controller) qui permet de séparer les données, les vues et les différentes actions que l'on peut effectuer.

Dans le cas de Natexo le développeur frontend a choisi la version quatre (4) pour mettre ses applications client.

D. Diagnostic de nouvelle architecture

Dans cette section nous allons aborder les améliorations apportées par la nouvelle architecture, et pour cela on commencera par un schéma technique explicatif de nouveau projet Natexo App (figure 37), qui illustre les interactions entre les différents services qui ont été refaits en respectant l'architecture orientée service,

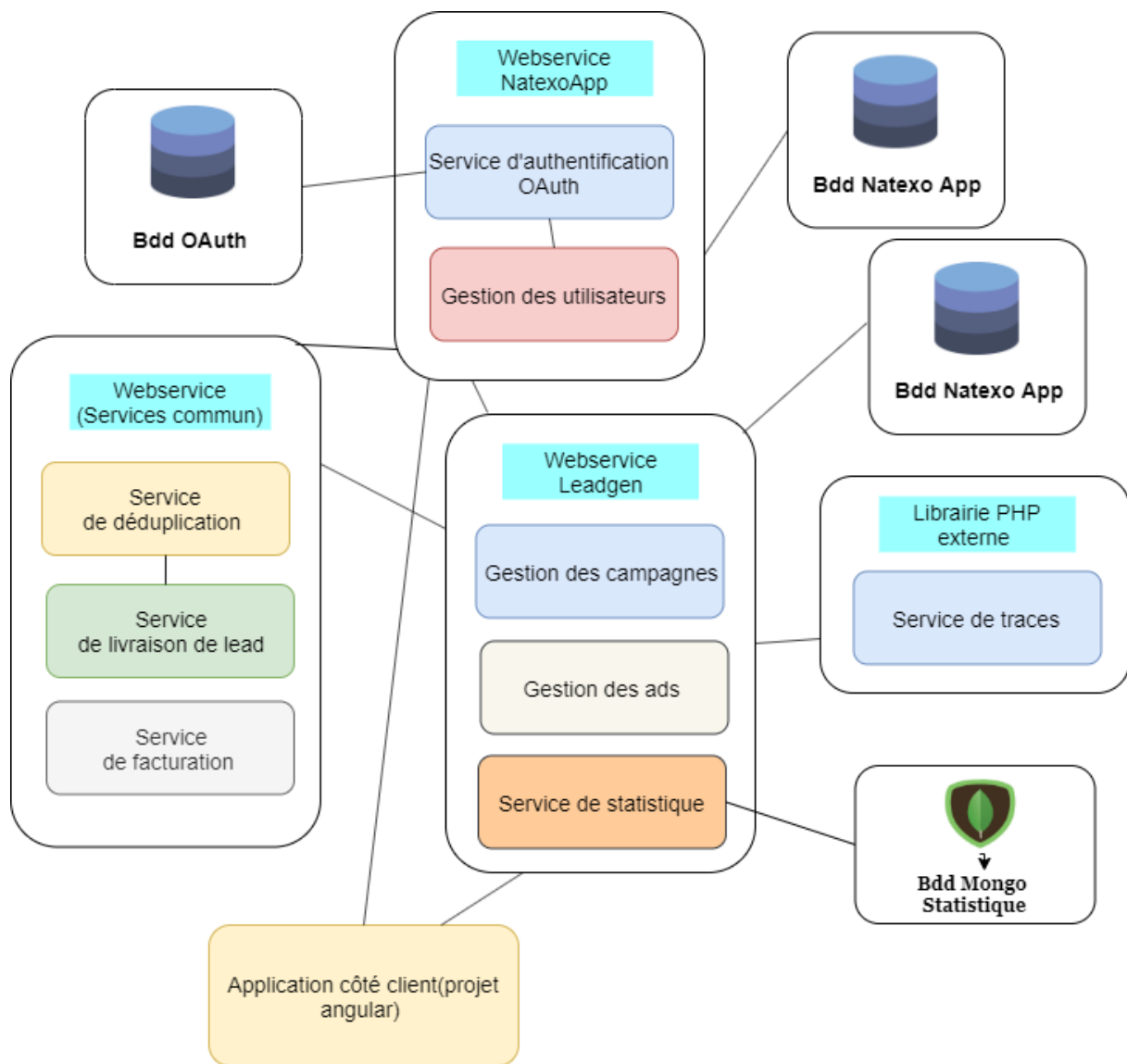


Figure 37 - Schéma technique de la nouvelle architecture

1- Avantages de cette architecture

Cette nouvelle architecture orientée services permet d'obtenir tous les avantages d'une architecture client-serveur et notamment :

- Une maintenance facile grâce à la décomposition de ce système d'information.
- La réutilisabilité possible des différents composants (service)
- Une modularité permettant de remplacer facilement un service
- Une meilleure possibilité d'évolution (il suffit de faire évoluer un service ou d'ajouter un nouveau service)
- Une plus grande tolérance aux pannes
- Code métier bien séparé de reste de code

E. Les tests unitaires et test d'intégration :

Actuellement, la réalisation des projets à Natexo se fait sans tests unitaires ni de tests d'intégration ou fonctionnelle, malgré l'importance de cette phase avec ses drivers niveau ou ses multitude type, cette dernière reste pour l'équipe R&D de l'entreprise une problématique majeur liée à :

- Manque de personnel chargé de lancer les tests unitaires car l'équipe R&D est composée actuellement de 5 développeurs et un Product owner,
- Les délais importants que cette phase pourra prendre pendant la réalisation et qui pourrait atteindre les trente (30 %) de temps de développement.

Afin d'implémenter et de tester les fonctionnalités réalisées :

- Un environnement de préproduction mis en place pour chaque service qui permet au Product owner de faire des tests fonctionnels,
- L'utilisation des autres outils comme Postman,
- Des api doc ajouter pour chaque service (service) web pour des différents tests. Le schéma illustre un exemple d'un api doc



API documentation		body format: Form Data request format: json
<ul style="list-style-type: none">/account/accountSolution/api/blockAccount/commission/contractDelivery/delivery/notification/offerForm/publisher/specialist/url/urlPublisher/article/contactMessage/default		
/account		Show/hide List Operations Expand Operations
GET	/v1/account/forget/password	Send a request to add password
GET	/v1/account/list	The account list
PUT	/v1/account/{piAccountId}/billing	Edit the account billing information
POST	/v1/account/{piAccountId}/contact	Create new contact for account
DELETE	/v1/account/{piAccountId}/contact/{piContactId}	Archive a contact
PUT	/v1/account/{piAccountId}/contact/{piContactId}	Update new contact for account
GET	/v1/account/{piAccountId}/contacts	Get account contacts list
GET	/v1/account/{piAccountId}/edit/billing	The detail billing of account client
GET	/v1/account/{piAccountId}/information	The account information
PUT	/v1/account/{piAccountId}/information	Edit the account information
POST	/v1/account/{piAccountId}/user	Create new user for account

Figure 38 - L'api doc de projet Leadgen (backend)

- Pour effectuer les tests de régression un système a été développé pour cela, Cette librairie PHP intégrer dans chaque Projet (webservice) permet de détecter les erreurs PHP sur des actions lancées sur nos serveur apache et sur les deux environnements, la production et la préproduction et par la suite les envoyer par email à l'équipe R&D

F. Gestion de projet

1- Méthodologies

Pour gérer ses projet, Natexo utilise la méthodologie agile qui se basée sur une approche itérative et incrémentale

Dans cette section, on définira cette approche ainsi la spécification utilisée par Natexo

a) Définition d'agilité :

La méthodologie Agile est une pratique qui aide à itérer en permanence le développement et les tests dans le processus SDLC (Software Development Life Cycle). Agile divise le produit en versions plus petites.

Dans cette méthodologie, les activités de développement et de test sont concurrentes, contrairement aux autres méthodologies de développement de logiciels. Cela encourage également le travail d'équipe et la communication en face à face. Les entreprises, les parties prenantes, les développeurs et les clients doivent travailler

Afin de mettre en pratique cette approche, Natexo utilise Scrum comme un processus de gestion. Agile est convenable à Natexo pour les raisons suivantes :

- Agilité est la mieux adaptée pour un environnement relativement incertain car Il est très difficile, voire impossible, de définir avec précision les exigences et la conception de la solution en détail avant le début du projet. Alors cette approche offre certain La flexibilité et l'adaptabilité
- Cette approche se traduit généralement par une mise sur le marché plus rapide grâce à des temps de démarrage plus courts, Pour Natexo le facteur de délais est un point critique vue aux évolutions apportées au domaine d'emailing
- Agilité permet de réduire significativement les coûts d'une manière indirects résultant de la réduction des exigences de documentation et de contrôle inutiles
- Dans un projet Agile, la qualité fait partie intégrante du processus de développement plutôt qu'une activité séquentielle. Les développeurs savent que la qualité n'est pas la responsabilité de quelqu'un d'autre
- Cette approche devrait entraîner une plus grande satisfaction de la clientèle ainsi celle des employés, car ils sont beaucoup plus impliqués pour assumer la responsabilité de leur propre travail au sein de la société.

2- Définition de Scrum

Scrum est une méthodologie ou un cadre de gestion de projet agile utilisé principalement pour les projets de développement logiciel dans le but de fournir une nouvelle fonctionnalité logicielle toutes les 2 à 4 semaines. C'est l'une des approches qui a influencé le Manifeste Agile , qui articule un ensemble de valeurs et de principes pour guider les décisions sur la manière de développer plus rapidement des logiciels de meilleure qualité.

3- Les outils utilisés

Natexo utilise JIRA comme plateforme de gestion des différents tickets, ces tickets sont repartis selon leur contexte en trois catégories :

- Tickets release (nouvelle fonctionnalité) : c'est l'ensemble des tickets qui correspond à l'ajout d'une fonctionnalité ou une partie de cette fonctionnalité
- Tickets support : cette catégorie correspond à l'ensemble des tickets de maintenance de l'existant
- Tickets production : correspond à l'ensemble des tickets créés par les commerciaux afin de répondre à leur besoin comme l'exportation manuelle des leads depuis nos bases de données ou maintenir les web service qui communique avec nos partenaires lors de la phase de livraison, ses tickets sont à gérer par deux développeurs et valider pendant leur création par le responsable de production qui jouira le Product owner pour cette catégorie

Chez Natexo la durée d'un sprint est de dix (10) jours (2 semaines), le suivi de ce sprint est assuré par des réunions quotidiennes (Daily meeting). La figure ci-dessous présente le cycle de Scrum a Natexo

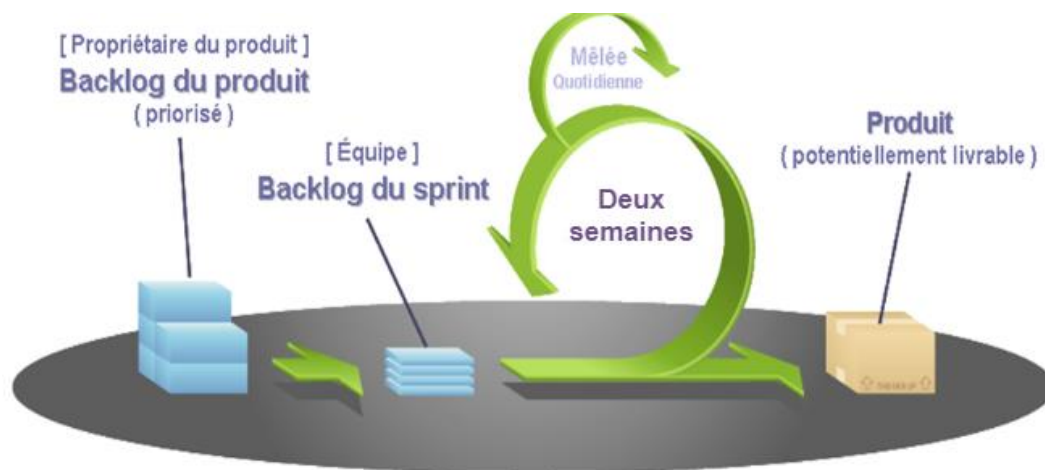


Figure 39 - Le cycle de Scrum

VIII. CONCLUSION

De nos jours, l'évolutivité et la réutilisabilité des composants au sein d'une entreprise ne doit plus constituer un problème majeur car plusieurs architectures pour y remédier peuvent être proposées et implémentées, Avec toutes les architectures existantes aujourd'hui, chaque entreprise peut choisir celle qui lui convient et celle adaptée à ces besoins.

Dans ce document j'ai abordé le sujet d'une migration depuis une architecture existante monolithe vers une nouvelle architecture cible.

Le choix d'une architecture orientée service dans le cas de la société Natexo été un choix correcte, cette déduction été le résultat d'une analyse sur le choix d'une architecture susceptible de remplacer l'ancienne architecture, l'analyse de choix a été basé sur trois éléments principale, le premier élément est la qualité et les critères de nouveau système attendu afin de résoudre les problèmes d'évolutivité de système d'information, pour Natexo, une société active dans le domaine d'emailing inconstant avec l'évolution des techniques d'emailing, ce facteur été un point essentiel a prendre en compte, un système d'information qui n'est pas ouvert à l'extensibilité constitue une vraie menace a la société pour s'adapter, un autre éléments attendu a la fin de cette transition été la réutilisabilité des service qui sont en commun entre les services d'emailing proposé par l'entreprise qui permettra de réduire les délais de la mise en place des nouveaux produit. Cette analyse a été fondé aussi sur les délais de la réalisation de cette migration en parallèle avec la réalisation et lancement de nouveau projet « Leadgen programmatic », le troisième élément dans cette analyse été les coûts et les investissements possible a effectués sur les différents niveaux.

Pendant cette transition, la planification et les priorités fixé pour chaque tâche été un élément clé de la réussite de cette migration, cela nous a permis et en tant que membre de L'équipe R&D de la société de mettre en place et de lancer le nouveau produit Leadgen dans une courte duré par rapport au projet « collecte » qui partage les mêmes fonctionnalités.

Personnellement, malgré une année passée au sein de la société avant d'entamer cette migration, cela n'été pas suffisant de pour moi pour comprendre ce demain complexe.

Au début de cette migration, le majeur problème rencontré été de comprendre le besoin métier depuis le code existant, mais cela est très vite clarifier par l'intervention de notre Product owner pendant des réunions quotidiennes ou par des tickets détaillés de la tâche a effectuée, Ses réunions été une opportunité de dialogue et de partage de savoir technique ou autres.

Ma coopération pendant cette transition, m'a donnée une autre vision sur ce domaine car j'avais la chance de revenir sur les fonctionnalités implémente pour les comprendre et les refaire en respectant les concepts d'une architecture orientée service, cette migration m'a permis aussi d'appliquer mes acquis théorique et technique au sein de mon école NextAdvance et cela m'a permis de monter en compétences.

Avant d'arriver sur le projet, je ne connaissais pas les problèmes ou l'importance d'une architecture choisie au sien une entreprise. Travailler sur une telle migration a été pour moi une expérience très enrichissante car j'ai appris de nouvelles technologies que je ne connaissais pas auparavant. J'ai également pu approfondir mes connaissances dans les technologies comme

PHP et Symfony. Mais le plus important pour moi a été d'avoir pris connaissance des problèmes liées à une architecture existante.

En finale, la rédaction de ce mémoire m'a permis de connaître en détail les concepts de l'architecture orientée (SOA) ainsi l'architecture candidat, l'architecture microservices, et la société Natexo été un exemple réel sur le quelle une analyse a été faite pour déterminer l'architecture le plus adéquate.

IX. DEFINITIONS ET LISTE DES ABREVIATIONS

Kubernetes : est un système open source qui vise à fournir une « plate-forme permettant d'automatiser le déploiement, la montée en charge et la mise en œuvre de conteneurs d'application sur des clusters de serveurs »³. Il fonctionne avec toute une série de technologies de conteneurisation, et est souvent utilisé avec Docker. Il a été conçu à l'origine par Google, puis offert à la Cloud Native Computing Foundation.

<https://fr.wikipedia.org/wiki/Kubernetes>

AWS : est une division du groupe américain de commerce électronique Amazon.com, spécialisée dans les services de cloud computing à la demande pour les entreprises et particuliers

https://fr.wikipedia.org/wiki/Amazon_Web_Services

W3C : (World Wide Web Consortium) est un organisme de standardisation à but non lucratif, fondé en octobre 1994 chargé de promouvoir la compatibilité des technologies du World WideWeb tellesque HTML5, HTML, XHTML, XML, RDF, SPARQL, CSS, XSL, PNG, SVG et SOAP. Fonctionnant comme un consortium international

https://fr.wikipedia.org/wiki/World_Wide_Web_Consortium

WSDL : Web Services Description Language st une grammaire XML permettant de décrire un service web.

https://fr.wikipedia.org/wiki/Web_Services_Description_Language

SOAP : (ancien acronyme de Simple Object Access Protocol) est un protocole d'échange d'information structurée dans l'implémentation de services web bâti sur XML.

<https://fr.wikipedia.org/wiki/SOAP>

YAML : (Yet Another Markup Language) est un format de représentation de données par sérialisation Unicode.

<https://fr.wikipedia.org/wiki/YAML>

FTP : File Transfer Protocol (protocole de transfert de fichier), ou FTP, est un protocole de communication destiné au partage de fichiers sur un réseau TCP/IP.

https://fr.wikipedia.org/wiki/File_Transfer_Protocol

X. BIBIOGRAPHIE

1- Livres

Paul C. Brown / (*Implementing SOA: Total Architecture in Practice*) Edition 1

2- Site internet

L'architecture microservices avec symfony <https://fr.slideshare.net/meeticTech/php-symfony-microservices-migration-meetictech>

Article The soa Source Book , opengroup <https://www.opengroup.org/soa/source-book/intro/index.htm>

Cours « architectures orientée services » , Université de l'Alberta enseigné par kenny wong, <https://www.coursera.org/lecture/service-oriented-architecture/4-1-9-object-brokers-Y2qnQ>

Article Microservices vs. SOA — Is There Any Difference at All? , Kristijan arsov <https://medium.com/@kikchee/microservices-vs-soa-is-there-any-difference-at-all-2a1e3b66e1be>

3- Conférences

SymfonyLive Paris 2017, Etienne Broutin - Micro-Services Symfony chez Meetic : retour d'expérience

Conférence Thomas Lamirault : « Du monolithe à une architecture orientée service »

XI. ANNEXE

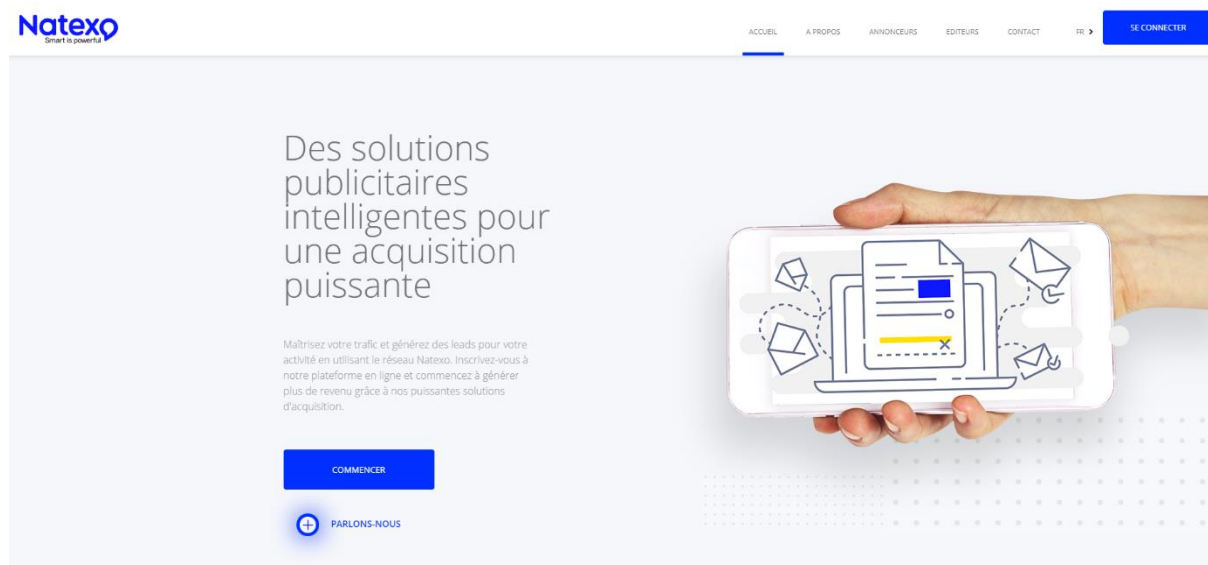


Figure 40 - La page d'accueil public de projet Leadgen

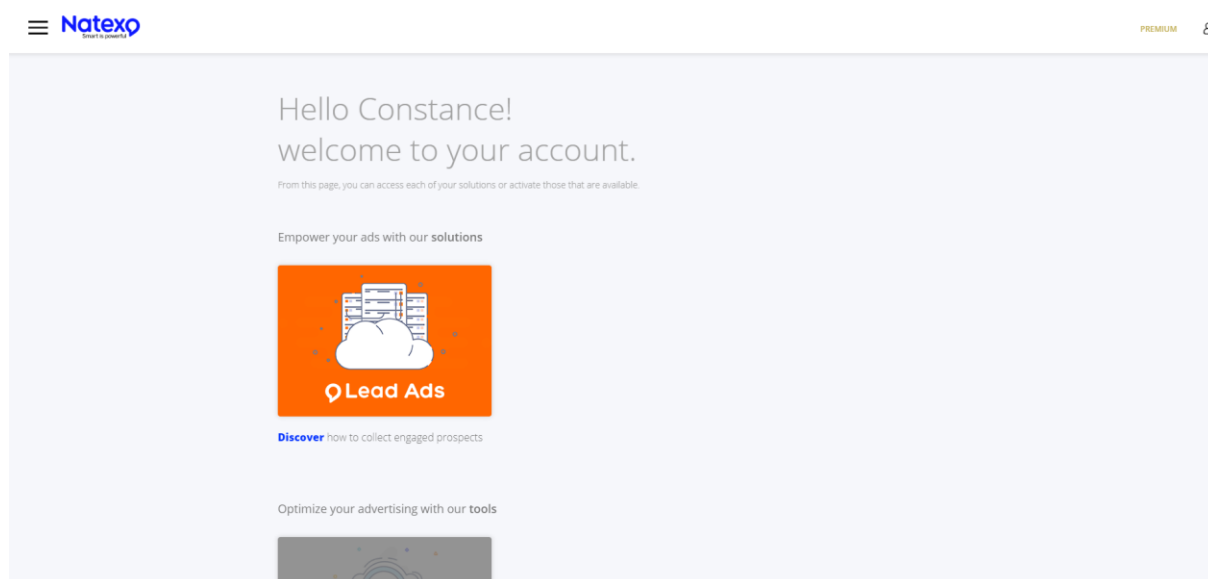


Figure 41 - La page d'accueil pour les annonceurs

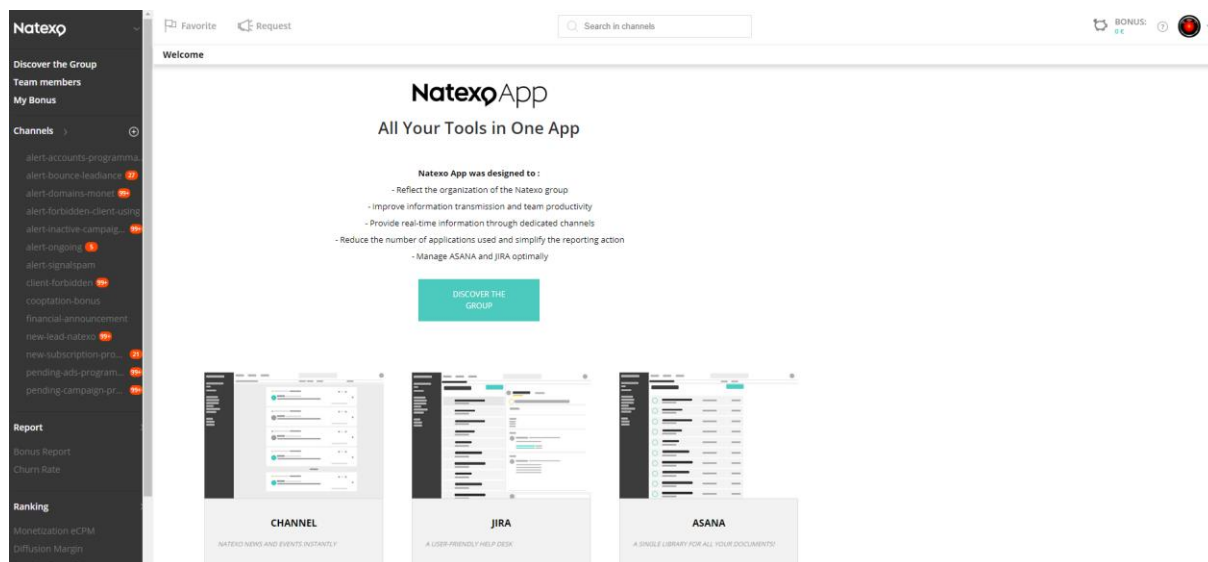


Figure 42 - La page d'accueil de projet Natexo App

PUBLISHER	CA	MARGE	% MARGE	ECPM NATEXO	ECPM P	NOMBRE ADS	BLACKLIST
TOTAL	42 €	19.81 €	47.17 %	28.81	67.63	28	0
Rank me up	0 €	0.00 €	0.00 %	0.00	0.00	6	0
NATEXO LANDING	0 €	0.00 €	0.00 %	0.00	0.00	3	0
MEDAZZEN	18 €	4.80 €	48.00 %	208.00	120.00	9	0
2-30 media	0 €	0.00 €	0.00 %	0.00	0.00	1	0
ADV FR - NATEXO PERFORMANCE	32 €	16.00 €	50.00 %	344.00	172.04	8	0
YourSurfMedFinances.com	0 €	-0.10 €	0.00 %	0.00	1.50	1	0

Figure 43 - Page de statistique de Leadgen selon les éditeurs