



# Rapport d'alternance

{ Refonte d'outils de modération }

Entreprise : Atchik

Lieu : Toulouse

Durée : Vingt Mois

École : Epitech

Alternant : HORNEMAN GUTTON Quentin

## Remerciements

Avant tout développement sur cette expérience professionnelle, il semble opportun de commencer ce rapport de stage par des remerciements.

Il n'est jamais facile pour un étudiant de trouver une alternance, aussi je tiens à remercier Atchik de m'avoir accueilli durant ces vingt mois.

J'aimerais tout d'abord remercier M. PITORRE Benjamin, mon tuteur, d'avoir eu confiance en moi lors de notre premier entretien et d'avoir prolongé cette confiance tout au long de mon contrat.

J'adresse ensuite mes remerciements à toute l'équipe qui a su m'intégrer et qui m'a beaucoup aidé, plus particulièrement à M. OUVRIÉ Mathieu pour m'avoir accompagné durant de cette aventure.

# Sommaire

## Table des matières

Remerciements .....	2
Sommaire .....	3
Introduction.....	4
I. CONTEXTE.....	5
A. Description .....	6
B. Services.....	7
C. Equipe et organisation .....	8
II. MA MISSION .....	10
A. Safescreen, NCIS, les outils existants .....	11
B. Tipex .....	13
C. Emploi du temps.....	14
III. RÉALISATION DE MA MISSION .....	15
A. Apprentissage de Big Query.....	16
B. Création d'un outil de statistique.....	22
C. Création de services .....	27
IV. CONCLUSION .....	30
A. Conclusion technique.....	31
B. Bilan personnel.....	32
Documentation.....	33
Outils utilisés .....	34
Langages utilisés.....	36
Glossaire des termes .....	37
Webographie.....	39

# Introduction

Dans le cadre de ma quatrième et cinquième année à l'Institut Européen des Technologies de l'Information de Toulouse, j'ai eu l'opportunité d'effectuer un contrat de professionnalisation chez Atchik, une entreprise spécialisée dans la veille et la modération de contenus et dialogues sur internet.

C'est après quelques mois de recherche et plusieurs entretiens que j'ai découvert cette PME toulousaine. J'ai commencé mon contrat le 2 janvier 2020, ayant pour but de développer Tipex, une refonte d'outils de modération.

Dans un premier temps, je vous présenterai l'entreprise Atchik dans son ensemble, puis je vous expliquerai le rôle des outils de modération existant chez Atchik et la mission qui m'a été confiée. Suivront ensuite les étapes de mon intégration ainsi que mon planning. Enfin, nous verrons les étapes clés de la réalisation de ma mission et les difficultés que j'ai rencontrées lors de la mise en place des nouveaux outils en question.

# I. CONTEXTE

## A. Description

Créée en 1999 à Toulouse, Atchik était spécialiste des conversations mobiles. Par suite d'une baisse d'activité en début d'année 2010 des chat SMS et **WAP** (en anglais : Wireless Application Protocol) et la montée en puissance des réseaux sociaux, l'entreprise a su s'adapter et devenir un acteur historique du web social en France.

Aujourd'hui, cette PME Toulousaine pilote, analyse et valorise des conversations en ligne chaque année. Elle accompagne marques et médias dans la gestion de leurs espaces communautaires. Elle propose des solutions innovantes et personnalisées à leurs clients depuis des années, parmi lesquels nous retrouvons :



FIGURE 1 LISTE DES CLIENTS

## B. Services

Leurs services de modération, veille, community management et prospection sociale, entièrement « made in France », sont opérés en continu, 7j/7, ce qui leur garantit une vision optimale des phénomènes en ligne et une aptitude naturelle à la gestion et à l'anticipation des situations de crise.



### Modération et détection

Ce service assure l'attractivité des espaces du client et fait de leurs conversations en ligne une source de valeur.

### Veille et analyse

La force de l'écoute exhaustive et de l'expérience opérationnelle au service du conseil stratégique le plus pertinent.



### Community management

Un service personnalisé et continu pour porter parole des internautes dans les conversations des réseaux sociaux.

## C. Equipe et organisation

Atchik est actuellement composée d'une trentaine de personnes et organisée par :



**BERTRAND**  
Président



**BRICE**  
Directeur général adjoint



**AMANDINE**  
Responsable  
administrative et RH



**STEVE**  
Directeur communication  
et conseil



**LUCE**  
Référente  
supervision

Nous pouvons identifier trois différents rôles dans l'entreprise parmi lesquelles :

- L'équipe superviseurs qui dirige les modérateurs dans leur planning et tâches quotidiennes
- L'équipe modérateurs qui modère les flux des conversations en ligne

Et enfin l'équipe technique composée de :

- Benjamin, directeur technique, accueilli à Atchik en 2019 dans le but de diriger le nouveau projet de refonte Tipex



- Mathieu, employé depuis 2006 en tant que modérateur et community manager. Il a commencé une reconversion dans l'informatique en 2014 et a commencé à travailler en tant que développeur au début de l'année 2018



**BENJAMIN PITORRE**  
Directeur Technique (CTO)



**MATHIEU OUVRIÉ**  
Développeur FullStack



**QUENTIN HORNEMAN**  
Alternant

L'équipe technique a pour but de :

- Solutionner les complications sur l'utilisation du matériel informatique, rencontrés par les membres d'Atchik (problèmes d'écran, claviers, etc.)
- Résoudre les problèmes logiciels rencontrés par les membres d'Atchik
- Développer Tipex

## II. MA MISSION

## A. Safescreen, NCIS, les outils existants

Ma mission consiste à développer une refonte des outils suivants :

- Safescreen : un outil de gestion de commentaires sur les réseaux sociaux. C'est la sixième version de l'application de modération propriétaire, créée par Atchik en 2010. Elle permet d'agréger des flux de messages afin d'y appliquer des décisions de modération. Ces décisions sont ensuite renvoyées aux plateformes à l'origine de ces messages. Les frameworks utilisés sont **Java**, **Spring**, **Hibernate**, **Google Web Toolkit**, **Flume** et Big Query.

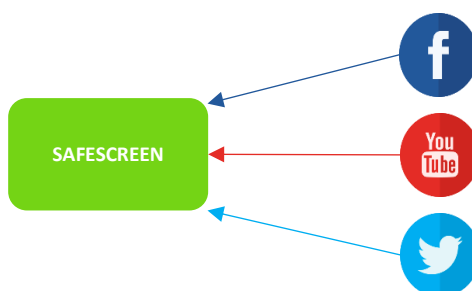


FIGURE 2 RECUPERATION DES MESSAGES

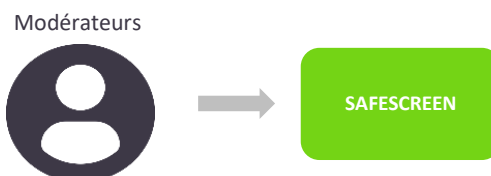


FIGURE 3 DECISION DE MODERATION

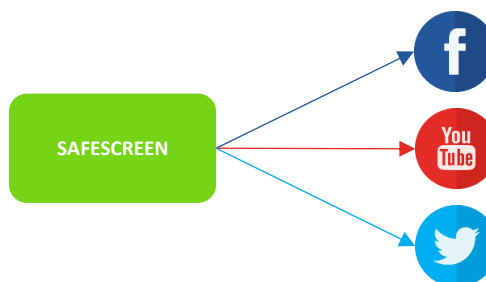


FIGURE 4 ENVOI DES DECISIONS

- NCIS : Développé en 2013-2014, NCIS (New Cool System Investigation), est un outil d'analyse statistique et de post-modération. C'est la première version « extérieure » à Safescreen. Les frameworks utilisés sont **Java**, **Spring**, **Hibernate**, **Google Web Toolkit** et Big Query.

Ces outils doivent être renouvelés, majoritairement parce qu'ils utilisent **Google Web Toolkit**, framework qui n'est plus mis à jour depuis le début de l'année 2020.

## B. Tipex

Tipex est le projet destiné à remplacer les outils Safescreen et NCIS.

D'une part, il garantira une plus grande adaptabilité aux typologies de ressources devenues plus diverses, comme les sites médias et les réseaux sociaux avec Facebook, Instagram, TikTok, etc.

D'autre part, il offrira des prestations plus diversifiées qu'Atchik vend tels que de la modération pure ou de la gestion de la relation client.

Enfin, il proposera une évolution des technologies utilisées pour l'infrastructure, pour davantage correspondre aux standards du marché : **conteneurs, big data** grâce à l'utilisation des frameworks suivant :

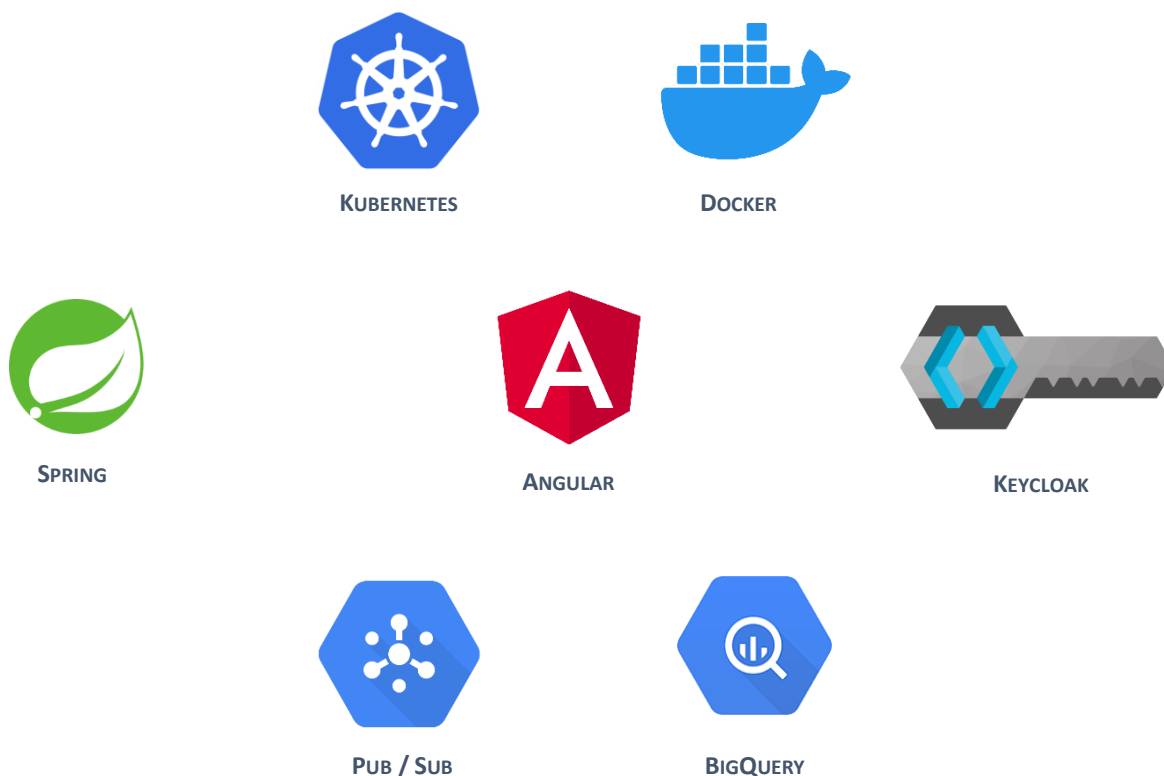


FIGURE 5 FRAMEWORKS  
TIPEX

## C. Emploi du temps

A mon arrivée chez Atchik, on m'a tout de suite partagé un planning des trois premiers mois détaillant les objectifs de chaque semaine.

	Activités	Compétences
Janvier		
S1	Présentation du métier, installation de l'environnement de dev	IntelliJ + git
S2	Exercice Java	Java
S3	Formation sur Safescreen (notre outil de modération) + les App Facebook	Facebook graph api + bootstrap
S4	Formation sur le YouTube manager	Java spring, Angular IO
Février		
S1	Présentation de notre outil de stat + Formation sur le cloud google	Bigquery, Data Studio
S2	Refonte de notre outil de stat avec google data studio	Bigquery, Data Studio
S3	Refonte de notre outil de stat avec google data studio	Bigquery, Data Studio
S4	Refonte de notre outil de stat avec google data studio	Bigquery, Data Studio
Mars		
S1	Mise en place de notre PIC (plateforme d'intégration continue)	Bitbucket, Jenkins
S2	Mise en place de notre PIC (plateforme d'intégration continue)	Bitbucket, Jenkins
S3	Déploiement en preprod de l'architecture micro-service Tipex (refonte de safescreen)	Kubernetes, Keycloak
S4	Test de performance de l'architecture Tipex	Cucumber, Gatling

FIGURE 6 PLANNING D'ACCUEIL

Lors de ma première semaine, Mathieu m'a présenté au métier de modérateur et j'ai mis en place mon **environnement de développement**.

Dans l'entreprise Atchik, l'horaire hebdomadaire moyen des alternants est de 35 heures, soit un forfait mensuel de 151,67 heures. Ainsi la durée quotidienne de travail est de 7 heures. Mes horaires étaient :

- De 9 h 30 jusqu'à 12 h 30
- De 13 h 30 jusqu'à 17 h 30

Tous les jours à 10 heures, tous les développeurs se réunissent pour un « daily meeting ». C'est un point sur les tâches que nous avons réalisées la veille, sur les difficultés que nous avons pu rencontrer et sur les tâches que nous allons aborder dans la journée.

Tous les jeudis, une réunion est organisée dans le but de présenter l'avancée du projet en compagnie de Brice (directeur général adjoint) mais aussi pour définir les différentes spécifications avec l'aide d'intervenants internes, superviseurs et modérateurs.

# III. RÉALISATION DE MA MISSION

## A. Apprentissage de Big Query

À partir du second mois de ma formation, j'ai pu découvrir l'outil de Google, Big Query. Le projet de refonte utilisant principalement ce service, il était important que j'aie toutes les connaissances nécessaires à sa maîtrise.

Rendu globalement disponible en novembre 2011 lors de la conférence Google Atmosphère, Big Query est un service web qui permet l'analyse interactive massive de grands ensembles de données. C'est une solution d'entrepôt de données d'entreprise qui ne requiert pas de serveur, hautement évolutive, qui ne nécessite pas de gestion d'infrastructure et qui supporte des analyses sur de très grandes volumétries de données.

Vous pouvez effectuer l'intégralité de l'interaction avec BigQuery de plusieurs manières :

- Utilisation de la console **Google Cloud Platform** (GCP)
- Utilisation des commandes Command-line interface (CLI)
- Bibliothèques clientes pour le langage de programmation de votre choix

J'ai tout d'abord commencé par utiliser la console **GCP** via un compte créé par l'entreprise. Pour ce faire, je me suis rendu sur le site <https://cloud.google.com> et j'ai accédé à la console.

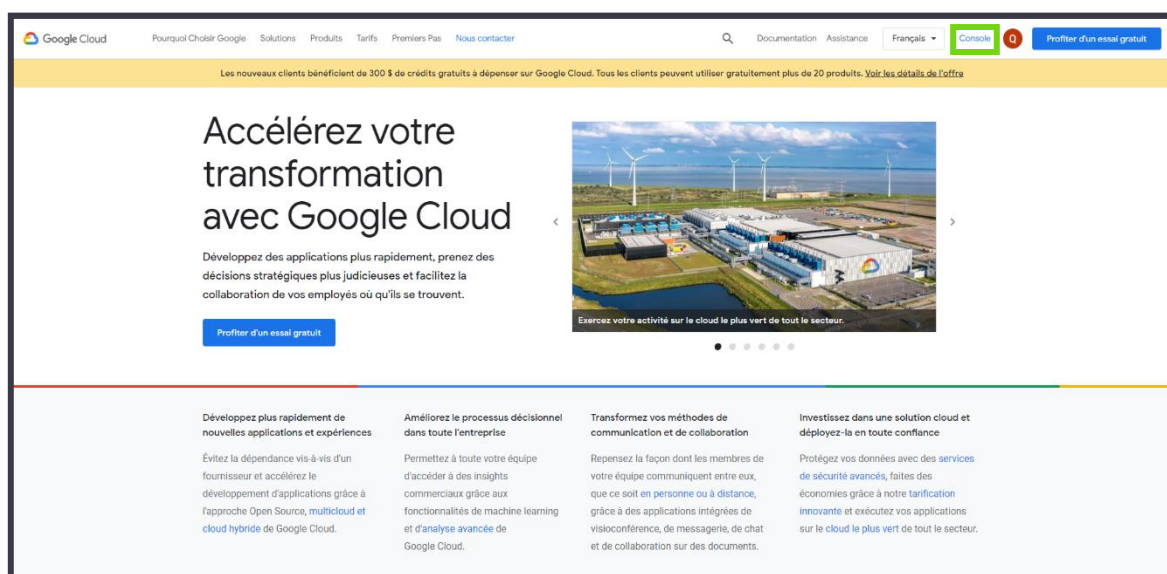


FIGURE 7 CLOUD GOOGLE



Ceci fait, j'ai sélectionné le projet de l'organisation et eu accès aux paramètres du projet, de l'état de **GCP**, l'état actuel de la facturation dans le but de monitorer au mieux le projet.

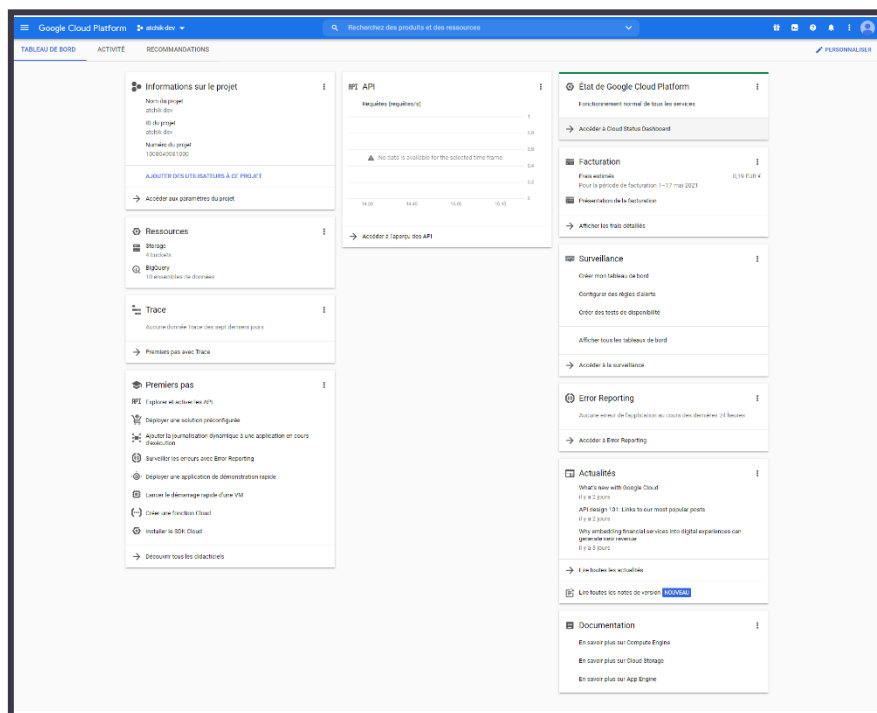


FIGURE 8 CONSOLE CLOUD GOOGLE

Sur la partie haut gauche de la console, j'ai accès au menu de navigation qui me permet de parcourir les différents produits et services de **Google Cloud** dont Big Query dans la partie Big Data.

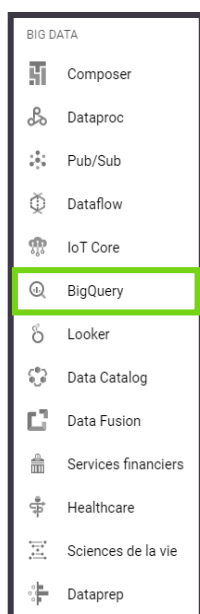


FIGURE 9 MENU DE NAVIGATION

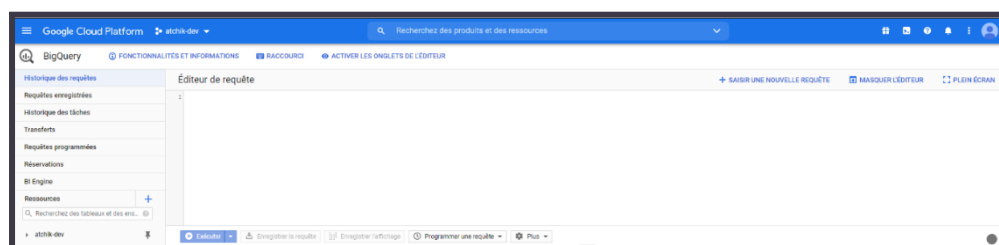


FIGURE 10 INTERFACE BIG QUERY

À partir du projet, j'ai créé un ensemble de données contenant plusieurs tables. Fondamentalement, une table est un compartiment de **base de données**, similaire à une feuille de calcul Excel, avec un nombre fixe de colonnes nommées (les attributs ou propriétés de la table) et un nombre quelconque de lignes de données.

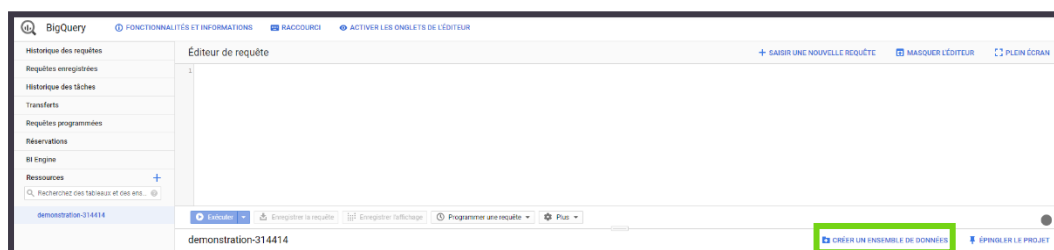


FIGURE 11 BOUTON CREATION ENSEMBLE DE DONNEES

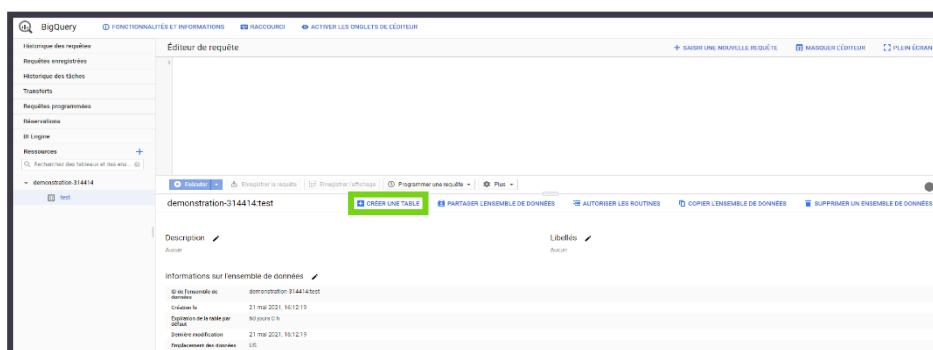


FIGURE 12 BOUTON CREATION D'UNE TABLE

 The image shows the 'Créer une table' (Create table) form. It has two main sections: 'Source' and 'Destination'. Under 'Source', there is a dropdown menu 'Créer une table à partir de :' with 'Table vide' selected. Under 'Destination', there are radio buttons for 'Rechercher un projet' (selected) and 'Saisir un nom de projet'. Below these are three input fields: 'Nom du projet' (containing 'Demonstration'), 'Nom de l'ensemble de données' (containing 'test'), and 'Type de table' (containing 'Table native'). At the bottom, there is a text input field for 'Nom de la table' containing 'message'.

FIGURE 13 CREATION D'UNE TABLE

C'est sur cet outil qu'Atchik stocke toutes les informations des messages modérés, comme la plateforme sur laquelle le message a été posté, le lien vers le profil du commentateur, etc. En règle avec la **RGPD**, les données sont stockées sur une durée maximale d'une année. Ci-dessous vous trouverez le schéma simplifié de la table.

id	STRING	Id du message
content	STRING	Contenu du message
date	TIMESTAMP	Date du message
link	STRING	Link du message
initialStatus	STRING	Status initial du message
clientName	STRING	Nom du client
resourceInternalId	STRING	Id en interne de la ressource
resourceInternalName	STRING	Nom en interne de la ressource
resourcePlatform	STRING	Nom de la plateforme
resourceIsPrivate	BOOLEAN	Boolean représentant si la ressource est privée ou pas
userId	STRING	Id de l'utilisateur
userName	STRING	Nom de l'utilisateur
userLink	STRING	Link du profil utilisateur
parentId	STRING	Id du commentaire parent
parentContent	STRING	Contenu du commentaire parent
parentLink	STRING	Link du commentaire parent
topicId	STRING	Id de la publication parent du commentaire
topicContent	STRING	Contenu de la publication parent du commentaire
topicLink	STRING	Lien de la publication parent du commentaire
processTime	INTEGER	Temps de modération du message
actionBan	BOOLEAN	Boolean représentant si l'utilisateur a été banni à la suite du message
actionLike	BOOLEAN	Boolean représentant si le message a été liker par le modérateur
actionResponse	BOOLEAN	Boolean représentant si le modérateur a répondu au message
decision	STRING	Décision prise par le modérateur (PUBLISH ou HIDE)
decisionModerator	STRING	Nom du modérateur
decisionDate	TIMESTAMP	Date de la décision
decisionStatus	STRING	Status de la décision
label	RECORD	Liste de labels assignés au message
postModerated	BOOLEAN	Boolean représentant si le message a été post modérer

**FIGURE 14 SCHEMA SIMPLIFIE DE LA TABLE MESSAGES**

STRING	Données de type caractères de longueur variable (Unicode)
INTEGER	Valeurs numériques sans composants fractionnaires
TIMESTAMP	Point absolu dans le temps avec spécification de la microseconde
BOOLEAN	Type de variable à deux états (généralement notés vrai et faux)
RECORD	Conteneur de champs triés ayant chacun un type (obligatoire) et un nom de champ (facultatif)

**FIGURE 15 TYPES DE DONNEES DU LANGAGE SQL STANDARD**

BigQuery accepte les types de données simples tels que les nombres entiers, ainsi que les types plus complexes tels que ARRAY (tableau) et STRUCT (structure) .

Pour manipuler ces tables, BigQuery est compatible avec deux dialectes SQL :

- Le SQL standard
- L'ancien SQL (SQL Legacy)

SQL est l'abréviation de Structured Query Language (langage de requête structuré). Il est utilisé pour communiquer avec une base de données. Les instructions SQL sont utilisées pour effectuer des tâches telles que la mise à jour ou la récupération de données dans une base de données. Bien que la plupart des systèmes de bases de données utilisent SQL, la plupart d'entre eux ont également leurs propres extensions propriétaires supplémentaires qui ne sont généralement utilisées que sur leur système. Tout au long de ma mission, j'ai principalement utilisé le SQL standard qui a plus de fonctionnalités que le Legacy.

Pour extraire des données d'une base de données SQL, nous devons écrire des requêtes. Une requête est qu'une instruction qui déclare quelles données nous recherchons, où les trouver dans la base de données et, comment les transformer avant de les renvoyer. La plus basique que nous pourrions écrire serait une requête qui sélectionnerait quelques colonnes (propriétés) de la table avec toutes les lignes (instances).

```
SELECT
id,
content,
date
FROM `demonstration-314414.test.message`
```

FIGURE 16 EXEMPLE DE REQUETE

Cette requête SQL va sélectionner (**SELECT**) le champ “**id**”, “**content**” et “**date**” provenant (**FROM**) du tableau appelé “**demonstration-314414.test.message**”.

Il existe des fonctions d'agrégations qui récapitulent les lignes d'un groupe en une valeur unique. COUNT, MIN et MAX sont des exemples de fonctions d'agrégation.

```
SELECT
resourcePlatform,
COUNT(DISTINCT id) AS nbMessages
FROM `demonstration-314414.test.message`
GROUP BY resourcePlatform
```

FIGURE 17 EXEMPLE DE REQUETE

Cette requête SQL va sélectionner (**SELECT**), le nom de la plateforme **resourcePlatform**, le nombre de lignes dans l'entrée "**id**" distinct (**DISTINCT**) provenant (**FROM**) du tableau appelé "**demonstration-314414.test.message**". Le résultat est ensuite regroupé (**GROUP BY**) par le nom de la plateforme **resourcePlatform**.

Ma principale priorité dans cet apprentissage était d'optimiser au maximum les requêtes SQL afin de minimiser le post-traitement des données, à la différence des premiers outils qui génèrent des requêtes particulièrement longues et coûteuses.

En général, les requêtes qui impliquent le moins d'opérations sont plus performantes. Lorsque j'évalue les performances d'une requête dans BigQuery, la quantité de travail nécessaire dépend de plusieurs facteurs :

- Données d'entrée et sources de données : combien d'octets ma requête lit-elle ?
- Communication entre les nœuds (brassage) : combien d'octets ma requête passe-t-elle à l'étape suivante ? Combien d'octets ma requête transmet-elle à chaque emplacement ?
- Calcul : quelle est la capacité du processeur nécessaire pour ma requête ?
- Sorties (matérialisation) : combien d'octets ma requête écrit-elle ?
- Anti-modèles de requêtes : mes requêtes suivent-elles les bonnes pratiques relatives aux instructions SQL ?

Un avantage supplémentaire de Big Query est qu'il s'intègre sans effort avec d'autres produits **Google Cloud Platform**. En outre, vous pouvez facilement visualiser les données de Big Query dans Data Studio de Google.

## B. Création d'un outil de statistique

Après avoir appris à manipuler Big Query, j'ai eu pour mission de créer un outil de statistique, nommé StatAktiv, pour remplacer celui existant. L'outil n'étant pas reprenable en l'état, il était impossible d'en assurer la maintenance.

Les utilisateurs se servent de l'outil de statistique pour facturer le client, pour anticiper une prochaine facturation mais aussi pour vérifier le flux entrant du trafic. Pour cela, ils utilisent deux tableaux :

- « Tableau de bord Mois en cours » qui affiche pour chaque client son nom, le total et la projection à partir du premier jour du mois en cours
- « Tableau de bord Mensuel » qui affiche pour chaque client, son nom, le total et le total manuel du mois précédent

Grâce à Google Data Studio, j'ai dû mettre en place plusieurs tableaux de bord répondant aux besoins des superviseurs.

Data Studio est un outil de visualisation de données qui vous permet de créer des tableaux de bord interactifs et des rapports personnalisés. Il permet d'importer des informations à partir de différents systèmes.

Pour commencer, je me suis rendu sur <https://datastudio.google.com>. Une fois connecté à mon compte Google, j'ai sélectionné le modèle « Rapport Vide ».

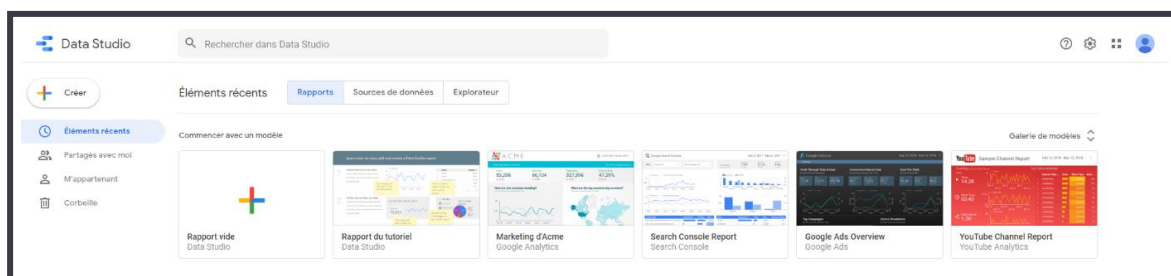


FIGURE 18 ACCUEIL DE L'INTERFACE UTILISATEUR WEB DE DATA STUDIO

Une fois le rapport créé, une page est directement ajoutée à celui-ci et j'ai dû lui associer des données provenant d'une ou plusieurs sources.

À ce stade-là, j'avais seulement ma table messages. Il était donc nécessaire de mettre en place une source de données contenant les statistiques demandées et calculées depuis la table messages. Pour ce faire, j'ai créé sur Big Query une seconde table qui a pour but de contenir les statistiques. Cette base de données doit se mettre à jour quotidiennement afin d'avoir à chaque instant, les statistiques de la veille. Grâce à Big Query, j'ai pu créer des requêtes programmées, qui vont s'exécuter toutes les n heures et vont insérer les statistiques dans la table.

Une fois la requête saisie, j'ai pu la paramétrer en ajoutant une date de début, une date de fin, une périodicité et la table de destination des résultats de la requête.

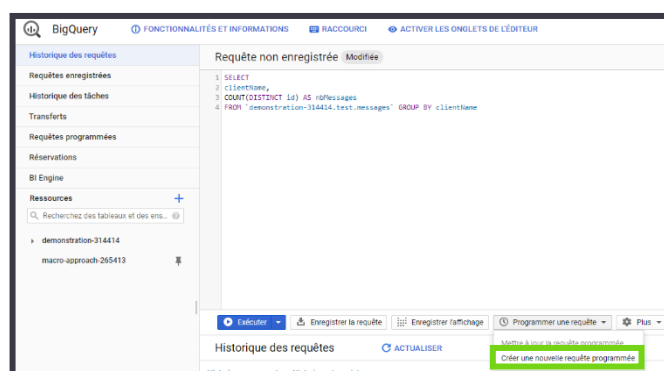


FIGURE 19 BIG QUERY, REQUETE PROGRAMMEE

Nouvelle requête programmée

Détails et programmation

Nom de la requête programmée

Requête programmée

Options de programmation

Choisissez la fréquence, l'heure et le fuseau horaire (par défaut, le fuseau horaire local est sélectionné). BigQuery convertira et planifiera la requête en heure UTC.

Périodicité

Tous les jours

Commencer

Programmer une heure de début

Date de début et heure d'exécution

01/06/2021 01:00 HAEC

Durée illimitée

Définir une heure de fin

Cette programmation exécutera Tous les jours at 23:00 UTC à partir du lun. mai 31 2021

Destination des résultats de la requête

Veillez indiquer une table de destination pour enregistrer les options des requêtes programmées.

Nom du projet

Demonstration

Nom de l'ensemble de données

test

Nom de la table

statistiques

FIGURE 20 CREATION DE REQUETE

Une fois que la table « statistiques » a été mise en place, j'ai pu l'associer au rapport Data Studio.

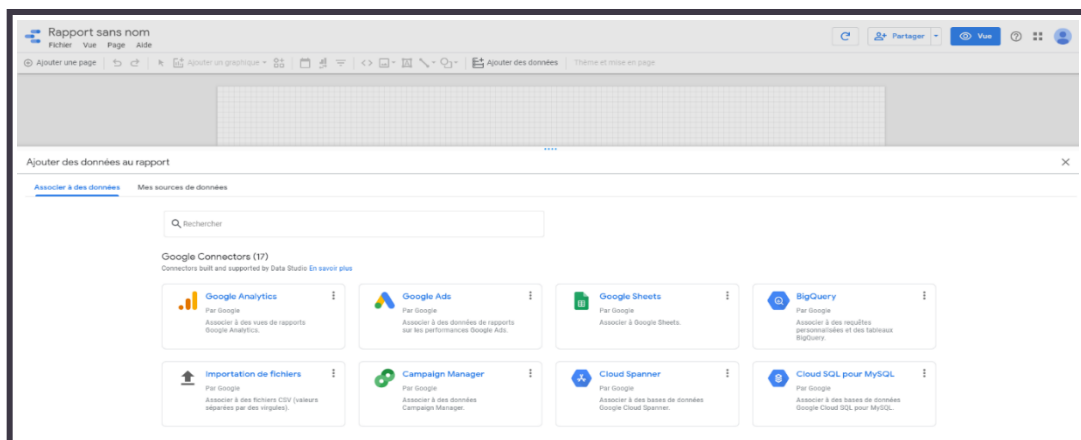


FIGURE 21 ASSOCIATION D'UNE SOURCE DE DONNEES AU RAPPORT

Par défaut, Data Studio crée une page contenant un tableau. Il est cependant possible d'ajouter des graphiques de séries temporelles, des graphiques à colonnes ou encore des cartes à bulles.

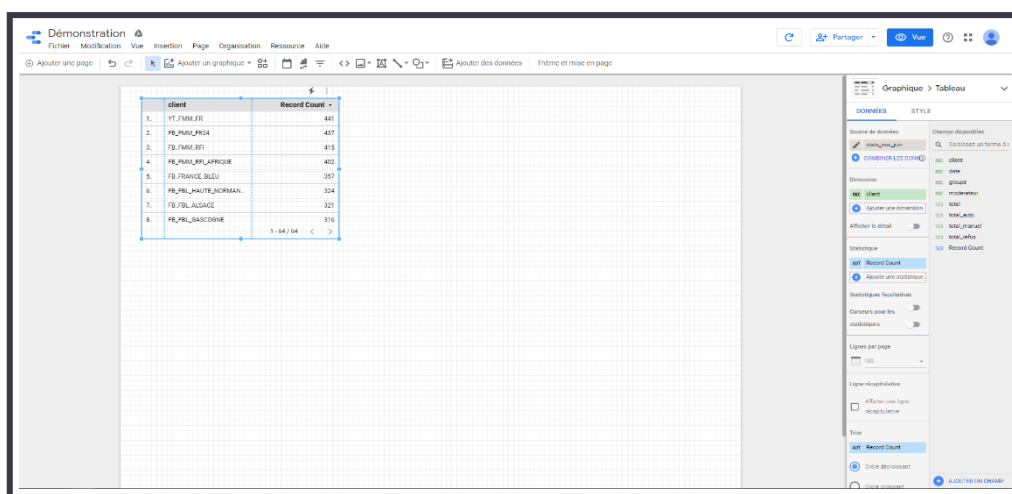


FIGURE 21 PAGE PAR DEFALT LORS DE LA CREATION D'UN RAPPORT

Nous pouvons changer le nom du rapport dans la partie haut gauche de l'interface utilisateur



FIGURE 22 MODIFICATION DU NOM DU RAPPORT



Il est possible de personnaliser les champs contenus dans notre source de données afin que mes rapports affichent les données comme l'utilisateur le souhaite.

Il existe trois types de champs dans Data Studio :

- Les dimensions décrivent ou classent vos données

Je peux regrouper les données en fonction de ces dimensions si je les ajoute à un graphique. Nom du client, date sont des exemples de dimensions permettant de regrouper les informations dans un graphique. N'importe quel type de données peut servir de dimension, y compris une colonne de chiffres non agrégés.

- Les statistiques mesurent vos dimensions

Elles s'obtiennent en appliquant une agrégation à un ensemble de valeurs. Cette agrégation peut provenir de l'ensemble de données sous-jacent, ou résulter de l'application implicite ou explicite d'une fonction d'agrégation telle que COUNT(), SUM() ou AVG(). Ces fonctions sont équivalentes à celles de Big Query. Si je veux compter le nombre de messages total, il me suffit d'utiliser la fonction COUNT() sur la colonne id.

- Les paramètres stockent les données définies par l'utilisateur

Contrairement aux informations des dimensions et des statistiques, qui sont extraites de l'ensemble de données sous-jacent, celles des paramètres proviennent des utilisateurs. Les paramètres peuvent ensuite servir à personnaliser vos rapports et sources de données.

Ci-dessous, une première version que le directeur général adjoint a pu expérimenter pendant quelques mois. Une seconde version est actuellement en développement pour approfondir les besoins de l'entreprise. Dans le futur, il est probable que ces rapports soient rendus disponibles pour les clients.

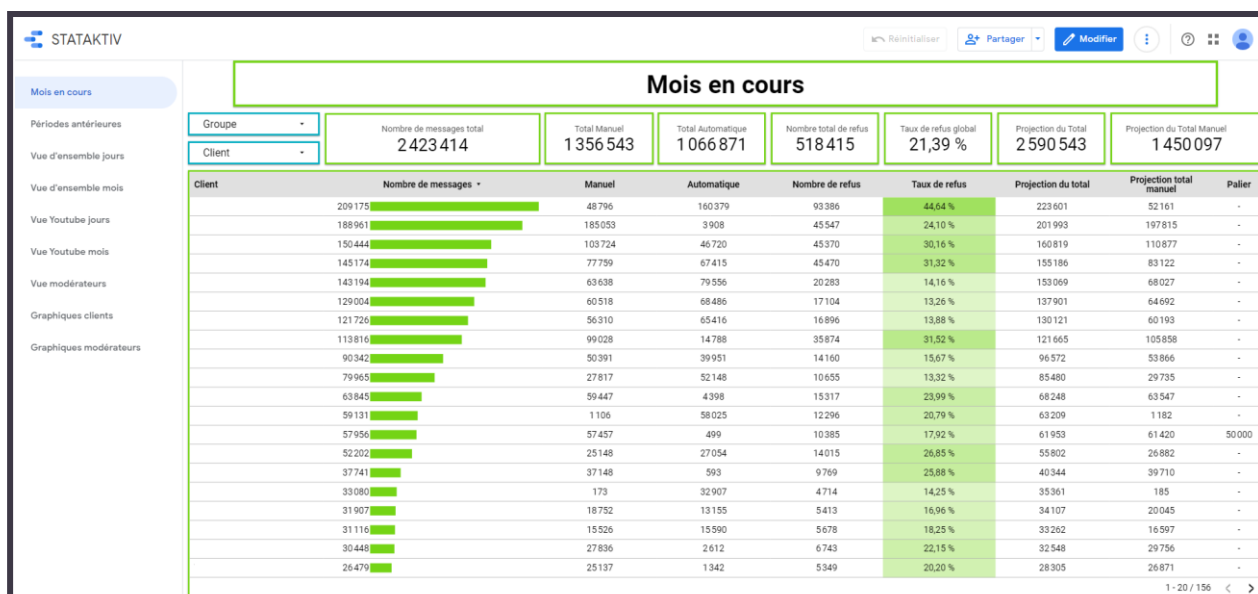


FIGURE 23 STATAKTIV

## C. Création de services

Après avoir élaboré des tableaux de statistiques, j'ai eu la tâche de produire plusieurs fonctionnalités pour l'application Tipex, par le biais de services.

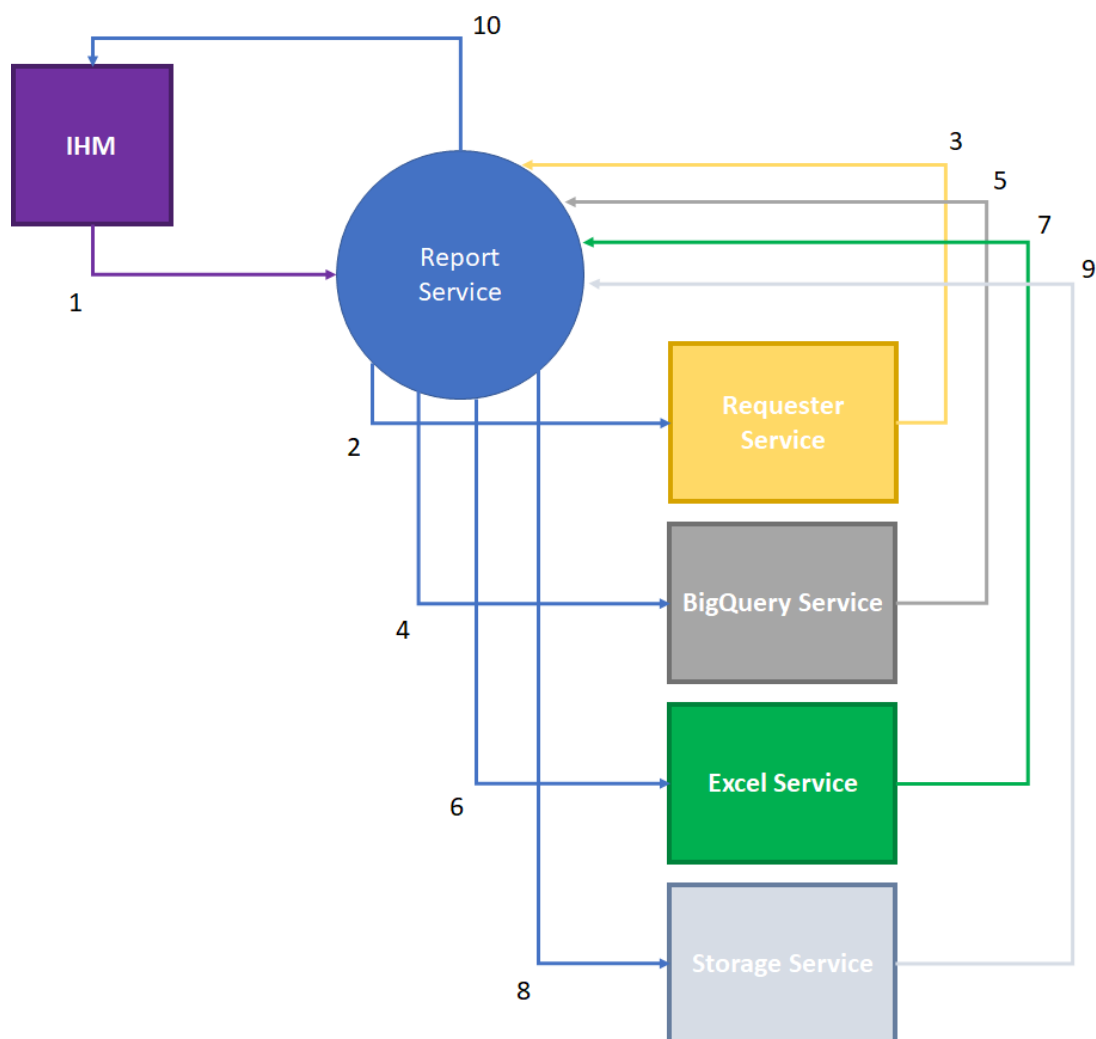
L'une des fonctionnalités était la création de rapport sous forme de fichier Excel. Il permet d'obtenir une liste prédéfinie de rapports statistiques, concernant a minima une ressource d'un client. Un rapport est basé sur 5 critères :

- Une période initiale
- Une période antérieure comparative
- Un client
- Une ou plusieurs ressources
- Un modèle de rapport

Cette fonctionnalité est associée à un service dit de haut niveau nommé « Report Service » et fonctionne grâce à l'intermédiaire de plusieurs services de bas niveau. À la différence des services haut niveau, les services de bas niveau sont autonomes.

La création des rapports se déroule suivant plusieurs étapes :

1. L'**IHM** envoie les paramètres utilisés pour le rapport au Report Service
2. Le Report Service demande pour chaque rapport les requêtes au Requester Service en lui donnant les instructions de la recherche
3. Le Requester Service fournit au Report Service les requêtes
4. Le Report Service demande au Big Query Service de récupérer les données depuis Google Big Query grâce aux requêtes générées précédemment
5. Le Big Query Service envoie au Report Service le résultat des requêtes effectuées
6. Le Report Service fait appel à l'Excel Service pour créer un fichier Excel contenant un onglet pour chaque type de rapport
7. L'Excel Service envoie au Report Service un classeur Excel complété
8. Le Report Service demande au Storage Service de stocker le fichier
9. Le Storage Service fournit le lien du fichier au Report Service
10. Le Report Service renvoie les données et le lien du fichier Excel à l'**IHM**



**FIGURE 24 ETAPES DE FONCTIONNEMENT DU REPORT SERVICE**

Le Report Service fait appel à quatre autres services :

- Requester Service : permet de créer les requêtes
- Big Query Service : permet de récupérer les données de la base de données depuis une requête, il utilise les bibliothèques clientes de Big Query pour interagir avec celui-ci
- Excel Service : permet de générer les fichiers Excel grâce à la librairie Apache POI
- Storage Service : permet d'uploader des fichiers sur le Cloud

Ci-dessous un exemple de rapport de type mensuel.

	A	B	C	D	E	F
1	ClientName	ResourceName	StartDate	EndDate	PreviousPeriodStart	PreviousPeriodEnd
2						
3						
4	Indicateurs					
5	Plateforme	Nombre de messages reçus	Variance			
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						

Global Client | FB Global Plateforme | FB Public RessourceName | FB Privé RessourceName

FIGURE 25 EXEMPLE DE RAPPORT MENSUEL, PAGE GLOBAL CLIENT

	A	B	C	D	E	F	G	H	I	J	K
1	ClientName	ResourceName	StartDate	EndDate	PreviousPeriodStart	PreviousPeriodEnd					
2											
3											
4	Indicateurs		Nombre de messages publics / privés								
5	Nombre de messages reçus			Nombre de messages reçus							
6	Variance			Nombre de messages publics							
7	Nombre d'utilisateurs uniques			Nombre de messages privés							
8	Temps de traitement moyen										
9	Nombre de fans										
10											
11	Messages par jour			Messages par ressource			Nombre de remontées par raison				
12	Date	Nombre de messages publics reçus	Nombre de messages privés reçus	Ressource	Nombre de messages reçus		Raison	Nombre de remontées			
13											
14											
15											
16											
17											
18											
19											

Global Client | FB Global Plateforme | FB Public RessourceName | FB Privé RessourceName

FIGURE 26 EXEMPLE DE RAPPORT MENSUEL, PAGE PLATEFORME

# IV. CONCLUSION

## A. Conclusion technique

Durant cette alternance, j'ai pu participer au développement de plusieurs outils de modération, de leurs créations à leurs premières versions, ceci au sein d'une petite équipe de développeurs et en constante communication avec les utilisateurs en interne.

Lors des tâches à réaliser, j'ai pu débattre et proposer mes idées. Mon avis et mes remarques furent pris en compte par l'équipe, ce que j'ai apprécié.

J'ai eu la possibilité d'approfondir mes connaissances sur le langage Java, SQL et eu la chance de découvrir des outils de la suite Google tels que Big Query, Data Studio ou encore Google Cloud Storage. En effet, étant mis à jour régulièrement et utilisés dans un grand nombre d'entreprises, la connaissance de ces outils me sera fortement utile à l'avenir.

## B. Bilan personnel

Pensant connaître le fonctionnement des réseaux sociaux au début de mon alternance, j'ai été agréablement surpris par la façon dont le contenu en ligne était modéré et aux moyens mis en œuvre pour y arriver.

Bien que la crise sanitaire ait chamboulé l'organisation de ces deux années d'études, j'ai tout de même pu mener à termes les tâches que l'on m'avait assignées. Néanmoins, j'ai préféré les périodes de déconfinement, où l'on pouvait travailler en présentiel. En effet, il me semble avantageux d'avoir la possibilité de partager plus facilement mes avancées et difficultés rencontrées.

Aujourd'hui et grâce à Atchik, j'ai la certitude qu'il est préférable pour un jeune développeur de faire ses premiers pas dans une petite entreprise. J'ai eu le sentiment d'avoir plus de responsabilités et plus d'impacts que de travailler pour une grande entreprise (GE) ou une entreprise de taille intermédiaire (ETI).



# Documentation

## Outils utilisés

**MobaXterm** : MobaXterm propose une application logicielle permettant d'utiliser sous Windows les commandes Unix/Linux. Destinée à aider les administrateurs système, les développeurs et les concepteurs de sites Web à effectuer des tâches à distance, celle-ci intègre un ensemble de clients distants (SSH, Telnet, Rlogin, RDP, VNC, XDMCP, FTP, SFTP, série), un serveur X11 et plusieurs commandes Unix.

Version utilisée : Personal Edition v12.4



**IntelliJ Idea** : également appelé « IntelliJ », « IDEA » ou « IDJ » est un environnement de développement intégré de technologie Java destiné au développement de logiciels informatiques. Il est développé par JetBrains (anciennement « IntelliJ ») et disponible en deux versions, l'une communautaire, **open source**, sous licence Apache 2 et l'autre propriétaire, protégée par une licence commerciale. Tous deux supportent les langages de programmation Java, Kotlin, Groovy et Scala.

Version utilisée : Ultimate | Community Edition 2019.3.1 Build 193.5662.53



**Bitbucket** : service web d'hébergement et de gestion de développement logiciel utilisant le logiciel de gestion de versions Git.



**Jenkins** : outil logiciel open source d'intégration continue, développé en JAVA. Il permet de faire des tests et des changements sur du code en temps réel.



## Jenkins

**Google Cloud Platform** (GCP) : plateforme de cloud computing fournie par Google, proposant un hébergement sur la même infrastructure que celle que Google utilise en interne pour des produits tels que son moteur de recherche. Cloud Platform fournit aux développeurs des produits permettant de construire une gamme de programmes allant de simples sites web à des applications complexes.



## Google Cloud Platform

## Langages utilisés

**Java** : Langage de programmation orienté objet, développé par Sun Microsystems. Il permet de créer des logiciels compatibles avec de nombreux systèmes d'exploitation (Windows, Linux, Macintosh, Solaris). Java donne aussi la possibilité de développer des programmes pour téléphones portables et assistants personnels. Enfin, ce langage peut être utilisé sur internet pour des petites applications intégrées à la page web ou encore comme langage serveur.

## Glossaire des termes

**Base de données** : Collection organisée de données, généralement stockées et accessibles électroniquement à partir d'un système informatique. Lorsque les bases de données sont plus complexes, elles sont souvent développées à l'aide de techniques de conception et de modélisation formelles. Le système de gestion de base de données (SGBD) est le logiciel qui interagit avec les utilisateurs finaux, les applications et la base de données elle-même pour capturer et analyser les données. Le logiciel du SGBD englobe en outre les fonctions de base fournies pour administrer la base de données. La somme totale de la base de données, du SGBD et des applications associées peut être désignée comme un "système de base de données". Souvent, le terme "base de données" est également utilisé pour désigner de manière générale le SGBD, le système de base de données ou une application associée à la base de données.

**Big data** : Le big data, les mégadonnées, ou les données massives, désigne les ressources d'informations dont les caractéristiques en termes de volume, de vitesse et de variété imposent l'utilisation de technologies et de méthodes analytiques particulières pour générer de la valeur, et qui dépassent en général les capacités d'une seule et unique machine et nécessitent des traitements parallélisés.

**Conteneurs** : structure de données, une classe, ou un type de données abstrait, dont les instances représentent des collections d'autres objets. Autrement dit, les conteneurs sont utilisés pour stocker des objets sous une forme organisée qui suit des règles d'accès spécifiques.

**Entrepôt de données** : Le terme entrepôt de données ou EDD (ou base de données décisionnelle ; en anglais, data warehouse ou DWH) désigne une base de données utilisée pour collecter, ordonner, journaliser et stocker des informations provenant de base de données opérationnelles et fournir ainsi un socle à l'aide à la décision en entreprise.

**Environnement de développement** : En programmation informatique, un environnement de développement est un ensemble d'outils qui permet d'augmenter la productivité des programmeurs qui développent des logiciels. Il comporte un éditeur de texte destiné à la programmation, des fonctions qui permettent, par pression sur un bouton, de démarrer le compilateur ou l'éditeur de liens ainsi qu'un débogueur en ligne, qui permet d'exécuter ligne par ligne le programme en cours de construction.

**Flume** : logiciel de la fondation Apache destiné à la collecte et à l'analyse de fichiers de log. L'outil est conçu pour fonctionner au sein d'une architecture informatique distribuée et ainsi supporter les pics de charge.

**Google Web Toolkit** : ensemble d'outils logiciels développé par Google, permettant de créer et maintenir des applications web dynamiques mettant en œuvre JavaScript, en utilisant le langage et les outils Java.

**Hibernate** : Framework open source gérant la persistance des objets en base de données relationnelle

**IDE** : Un environnement de développement intégré (IDE) c'est une application logicielle qui fournit aux programmeurs informatiques des outils complets pour le développement de logiciels. Un IDE comprend normalement au moins un éditeur de code source, des outils d'automatisation de la construction et un débogueur. Les environnements de développement intégrés sont conçus pour maximiser la productivité des programmeurs en fournissant des composants étroitement liés avec des interfaces utilisateur similaires. Les IDE présentent un seul programme dans lequel tout le développement est effectué. Ce programme offre généralement de nombreuses fonctionnalités pour créer, modifier, compiler, déployer et déboguer des logiciels. Cela contraste avec le développement de logiciels à l'aide d'outils non liés, tels que vi, GCC ou make.

**Open Source** : la locution open source permet de faire référence à tout logiciel dont les codes sont ouverts gratuitement pour l'utilisation ou la duplication, et qui permet de favoriser le libre échange des savoirs informatiques.

**RGPD** : Le sigle RGPD signifie « Règlement Général sur la Protection des Données » (en anglais « General Data Protection Regulation » ou GDPR). Le RGPD encadre le traitement des données personnelles sur le territoire de l'Union européenne.

**Spring** : Framework open source pour construire et définir l'infrastructure d'une application Java, dont il facilite le développement et les tests.

**WAP** : Le protocole WAP (en anglais : Wireless Application Protocol) est un protocole de communication apparu en France en 1999 qui permettait d'accéder à Internet à partir d'un appareil de transmission sans fil, comme un téléphone mobile ou un assistant personnel.

# Webographie

## Wikipédia

- [https://en.wikipedia.org/wiki/Main\\_Page](https://en.wikipedia.org/wiki/Main_Page)

## Google Cloud Platform

- <https://cloud.google.com>

## Big Query

- <https://cloud.google.com/bigquery/docs>