

ECOLE PRATIQUE DES HAUTES ETUDES COMMERCIALES



AVENUE DU CISEAU, 15 1348 LOUVAIN-LA-NEUVE

Monitoring de l'utilisation d'une application de "Rich Communications" à l'aide du Web Analytics

Travail de fin d'étude présenté en vue d'obtention du diplôme de bachelier en
Informatique et Systèmes orientation Technologie de l'Informatique

Steve Henriquet



Prometheus



Grafana

Rapportrice :
Virginie VAN DEN SCHRIECK

Client :
Escaux SA

Année Académique 2018 - 2019

Remerciements

En préambule de ce document, je tiens à remercier toutes les personnes qui ont contribué au succès de mon TFE et qui m'ont aidé lors de la rédaction de ce rapport.

Tout d'abord, j'adresse mes remerciements à mon client, Escaux représenté par Mr Joachim Lonfils, R&D manager au sein de l'entreprise Escaux, pour le sujet proposé et le temps qu'il m'a consacré pour effectuer le suivi de ce travail. Grâce aussi à sa confiance, j'ai pu accomplir totalement ma mission et j'ai énormément appris.

Je tiens à remercier mon professeur, Mme Virginie Van Den Schrieck qui m'a suivi tout au long de la réalisation de ce TFE et qui m'a aidé lors de la rédaction de ce présent rapport par ses conseils. Je remercie également l'EPHEC de nous donner l'opportunité de réaliser un tel travail.

Je remercie également l'équipe de développement d'Escaux qui s'occupe de l'application Connect Me et surtout Mr Gerik Bonaert qui m'a beaucoup aidé dans la compréhension du fonctionnement de l'application Connect Me d'un point de vue programmation.

Enfin, je tiens à remercier Pauline Hinnekens qui m'a aidé dans la rédaction de ce rapport de TFE par sa relecture et ses conseils pour l'écriture.

Table des matières

1. Introduction

La société Escaux, spécialisée dans le domaine de la télécommunication, en particulier dans les services de téléphonie IP (VoIP) pour les entreprises, a développé une application multiplateforme de Rich Communications appelée Connect Me. Afin de mieux comprendre l'utilisation de l'application par ses clients, Escaux souhaitait établir une solution pour récupérer les informations de Connect Me et permettre la visualisation de ces informations à l'aide d'un des outils de visualisation déjà implantés et utilisés au sein de la société. Ayant eu l'opportunité de réaliser mon stage au sein de l'équipe de développement d'Escaux, le responsable de cette équipe m'a proposé de travailler sur l'élaboration de la solution en question dans le cadre de mon travail de fin d'études (TFE).

Pour mon TFE, je souhaitais m'orienter vers la possibilité de réaliser un travail pour un client réel, c'est-à-dire de pouvoir utiliser mon TFE comme une opportunité de créer un outil ou une solution pour une entreprise. Le travail proposé par Escaux correspondait donc en tout point à cette attente, me permettant de proposer une solution qui répondrait à leur problématique. Cela m'a également contraint à travailler dans l'environnement en place chez Escaux, une contrainte qui me semblait très importante car il est essentiel de pouvoir y faire face et de savoir structurer son travail en fonction d'un cadre donné et fixe. En outre, la nature-même du sujet proposé par Escaux a attiré mon attention car elle nécessitait d'aborder des aspects divers de l'informatique, que ce soit de la programmation, de l'architecture ou encore du développement. Étant de nature polyvalente, je me suis naturellement tourné vers ce genre de challenge. Le travail proposé faisait également appel à un aspect d'engineering, étant donné qu'il fallait élaborer une solution de A à Z, en passant par la phase de documentation et de recherche de pistes potentielles pour la solution demandée, tout en se pliant aux contraintes présentes. Enfin, travailler pour un client réel m'a confronté à une contrainte supplémentaire, celle du temps. En effet, être son propre client laisse la liberté de ce que l'on souhaite élaborer et du temps consacré à cette élaboration. Par contre, avoir un client réel et la nature du travail à réaliser impliquait de devoir travailler au sein de l'infrastructure de la société, nécessitant une plus grande efficacité pour tirer profit au maximum des heures dédiées à la réalisation du TFE chez Escaux. Du temps était également nécessaire pour discuter avec le client afin de suivre l'orientation du travail, de préciser les caractéristiques au fur et à mesure de l'avancée et de poser certains choix. Une fois toutes mes attentes, celles du client et les contraintes mises ensemble, le sujet de TFE proposé par Escaux constituait un véritable challenge auquel je souhaitais m'atteler.

Le présent document a pour objectif de présenter la réflexion sous-jacente à la solution proposée pour répondre à la problématique d'Escaux, menant à l'explication de la solution mise en place. Pour ce faire, une première partie présente la problématique traitée, le cahier des charges à respecter, la méthodologie suivie et fournit également un contexte technique nécessaire à la compréhension du travail. Une deuxième section regroupe ensuite les informations recueillies dans la documentation en vue de comprendre le fonctionnement des différentes solutions envisagées au cours du projet. Une troisième section concerne les tests effectués ainsi que l'analyse des différentes solutions qui se présentaient aux différentes étapes du projet. Une quatrième et dernière partie traite de l'implémentation de la solution au sein du serveur d'Escaux, décrit l'architecture de la solution proposée et détaille la documentation nécessaire à l'utilisation ultérieure de la solution par la société.

2. Problématique générale et cahier des charges

Au sein dEscaux, récolter des informations au sujet de l'utilisation de Connect Me est une demande provenant à la fois d'un client dEscaux et d'une nécessité interne à la société. En effet, savoir si l'application Connect Me est utilisée et comprendre comment elle est utilisée par les clients permettrait de mieux orienter le développement de l'application. Les informations à récolter sont, entre autres, la navigation dans le site, les fonctionnalités utilisées et les changements de paramétrage. La société avait proposé d'utiliser une solution de Web Analytics afin de récupérer des informations sur l'utilisation de l'application et de créer un logiciel permettant de récupérer les informations collectées par cette solution de Web Analytics. Ce logiciel permettra ensuite d'afficher les informations récoltées sous différentes formes à l'aide d'un des outils de visualisation utilisés au sein de la société, Grafana ou Kibana.

En amont du travail réalisé pour proposer une solution adéquate au client, un cahier des charges a été établi afin de mettre sur papier la demande exacte du client, d'évaluer les points importants et ceux moins importants au sein de cette demande et d'établir la liste des critères que la solution devrait remplir. Cette section a pour but de décrire l'infrastructure de la société cliente (Figure ??), de pointer les différentes contraintes liées à cette infrastructure et d'explicitier les grandes lignes du cahier des charges.

La Figure ?? représente une version très schématique de l'infrastructure dEscaux par rapport à sa solution Connect Me. Tout d'abord, l'environnement est divisé en deux parties, à gauche est schématisé le côté client, avec les échanges d'informations via HTTPS vers un serveur proxy, alors qu'à droite est représenté l'environnement interne dEscaux, accessible en HTTP. La connexion entre les serveurs internes et le serveur proxy se fait via des tunnels SSH, permettant une scalabilité des connexions utilisées par Connect Me pour les appels téléphoniques. Cela permet également de chiffrer les données échangées entre les serveurs, ce qui renforce la protection de ces dernières en cas d'intrusion dans le réseau. Grâce à ce type d'architecture, la gestion des certificats SSL/TLS se déroule seulement sur le serveur proxy. Le fonctionnement de cette partie est assez simple : l'application Connect Me utilisée par les clients communique avec le serveur proxy qui retransmet les informations au serveur contenant la partie serveur (Backend) de Connect Me et les renvoie vers le bon client.

Pour alléger le schéma et éviter toute confusion, le monitoring des serveurs se faisant sur le backend de Connect Me n'a pas été représenté, évitant ainsi toute méprise avec la solution à implémenter en réponse à la demande du client. La solution à implémenter, représentée par le cadre contenant le point d'interrogation, dépendra des deux possibilités suivantes. Soit la solution de Web Analytics est hébergée dans le cloud de l'entreprise fournissant ce service, soit la solution de Web Analytics est hébergée sur un serveur dEscaux.

Dans le premier cas, les informations d'utilisation de Connect Me seraient envoyées à la solution hébergée dans une autre société. Une méthode de récupération est alors généralement fournie sous la forme d'une API (Application Programming Interface). Dès lors, la solution à implémenter serait une solution permettant de questionner cette API afin de récupérer les informations souhaitées, pour ensuite les formater afin qu'elles soient utilisables par la solution de visualisation. Dans le second cas, la solution de Web Analytics serait hébergée sur un serveur dEscaux et les informations d'utilisation seraient envoyées à ce serveur via le serveur proxy. Malgré l'installation au sein de l'entreprise, il faudrait comme dans le cas précédent créer un logiciel permettant de récupérer les informations et de les formater pour la solution de visualisation.

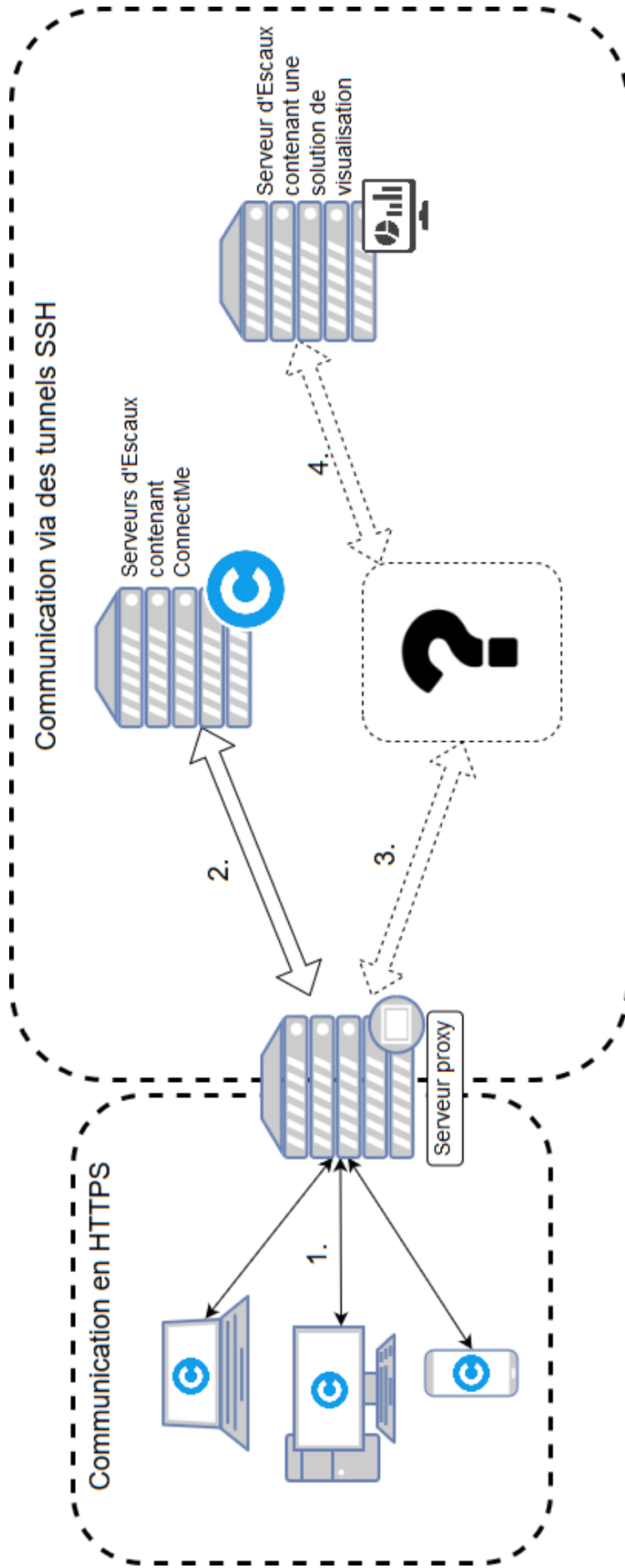


FIGURE I – Schéma de l'infrastructure d'Escaux avant implémentation de la solution proposée à la société. Figure originale, élaborée à l'aide de <https://www.draw.io/>

1 : Échange d'informations entre le serveur proxy et les différents périphériques dans lesquels peut fonctionner Connect Me. Ces informations regroupent les messages envoyés et reçus, les appels et autres types de communication ainsi que le code pour l'analytique et l'envoi des différents événements à Matomo.

2 : Échange des informations propres à Connect Me entre le serveur proxy et l'un des serveurs de Connect Me via des tunnels SSH.

3 : Récupération des informations d'utilisation de Connect Me

4 : Récupération des données formatées pour l'affichage dans une solution de visualisation

Par conséquent, qu'importe la situation envisagée, la solution doit pouvoir communiquer aussi bien vers l'extérieur de l'infrastructure qu'en interne. Afin de permettre les communications vers l'extérieur, il faudra éviter le blocage par le firewall installé au sein de la société et pouvoir traverser le serveur proxy. Pour la communication en interne, il faudra un tunnel SSH vers le serveur proxy et un autre vers le serveur contenant la solution de visualisation afin de permettre la récupération d'informations.

Pour ce qui est des informations à collecter, le client voulait récupérer des informations relatives aux points suivants :

- Appel
- Vidéo conférence
- Conversation par message
- Utilisation d'un casque audio
- Navigation à travers l'application
- Personnalisation de l'application
- Fonctionnalité par rapport à l'utilisateur
- Méthode de recherche de contact
- Informations générales d'utilisation

Pour la visualisation des informations collectées, le client souhaitait voir ces informations de manière précise, c'est-à-dire connaître exactement le nombre de messages envoyés, par exemple, mais aussi avoir une visualisation des tendances générales d'utilisation, comme la tendance par rapport à l'envoi de messages au cours de la journée.

Dans les annexes se trouvent toutes les informations d'utilisation de l'application récupérées par la solution proposée, ainsi que le nom de tous les graphiques créés pour la visualisation.

3. Méthodologie

L'élaboration de la solution pour Escaux s'est présentée sous forme d'une problématique à deux axes. D'une part, il fallait récolter les informations en provenance de différents serveurs impliqués dans les communications entre les utilisateurs. D'autre part, il était nécessaire de récupérer les données d'utilisation de la Web Application Connect Me. Pour répondre à la problématique, un grand nombre de solutions souffraient à moi pour la réflexion. Toutefois, ne connaissant pas ces différentes solutions, j'ai décidé de travailler pas à pas, en me basant sur les informations présentes dans la documentation des différentes solutions possibles et en collaboration avec mon client afin d'être à l'écoute des demandes spécifiques et des critères précis à remplir. La réflexion sur la solution proposée pour répondre à la problématique initiale s'est axée sur les quatre grandes étapes suivantes : le choix de la solution de Web Analytics, le choix de la base de données, le choix du serveur web et le choix de la suite logiciel pour l'exportation et la visualisation des données.

Voici un descriptif détaillé de la méthodologie suivie pour l'élaboration de la solution demandée par Escaux. À chaque étape du projet, différentes solutions se présentaient. Je me suis donc tout d'abord renseigné sur chacune des solutions (cf. les sections 4 Contexte technique 5 et 6 Tests et analyse 7) pour en connaître l'utilisation et les spécificités. Cependant, la documentation n'étant pas toujours suffisamment explicite, j'ai effectué un certain nombre de tests avec les différentes solutions afin d'expérimenter et de comprendre leur fonctionnement. Ensuite, le choix de la solution à conserver se faisait en concertation avec le client en comparant les avantages et les inconvénients que j'avais relevés pour chaque possibilité. Dans cet objectif, une réunion hebdomadaire était fixée avec le responsable de l'équipe de développement d'Escaux afin de présenter les avancées du projet au fur et à mesure et de discuter des différentes solutions. Ces réunions m'ont permis de vérifier que la solution que j'élaborais répondait bien à la demande du client, de m'assurer de la bonne direction prise par le projet et d'affiner de plus en plus les objectifs à atteindre.

4. Contexte technique

La solution mise en place dans le cadre de ce TFE a recours à différents outils et concerne l'application Connect Me développée par Escaux. Afin de mieux comprendre le choix de certains outils par rapport à d'autres lors de l'élaboration de la solution, une présentation technique de ces différents outils et de Connect Me s'impose et est reprise dans cette section.

4.1 L'application Connect Me

Connect Me est une application multiplateforme de Rich Communications. Une application de Rich Communications consiste en un softphone supportant également un service de messagerie (Figure ??). Connect Me prend en charge les appels audio et vidéo, la messagerie, l'annuaire personnel et professionnel des utilisateurs. Elle est basée sur le standard ouvert SIP (Session Initiation Protocol) et peut être utilisée en combinaison avec n'importe quel système de communication cloud ou d'entreprise. Cette application permet de passer du mode privé au mode professionnel via l'interface de l'application. Lorsque le mode privé est activé, toute communication professionnelle est coupée, alors qu'en mode professionnel, les communications privées sont maintenues. Connect Me peut aussi être liée à un agenda, tel que Google Calendar, pour se mettre automatiquement à jour en fonction de l'emploi du temps de l'utilisateur.

D'un point de vue technique, Connect Me est une web App écrite en AngularJS, lui permettant d'être facilement multiplateforme grâce à des WebViews. Les WebViews sont des objets en programmation Android, IOS et Windows permettant de visualiser et d'interagir avec du contenu normalement prévu pour être affiché dans un navigateur web mais sans contenir toutes les fonctions d'un navigateur[?, ?]. L'application a également été écrite de manière modulaire, chaque module ayant une fonction propre. Par exemple, les conversations, les utilisateurs, les paramètres, sont pris en charge par différents modules de l'application. Cette modularité est justement facilitée par l'utilisation d'AngularJS.

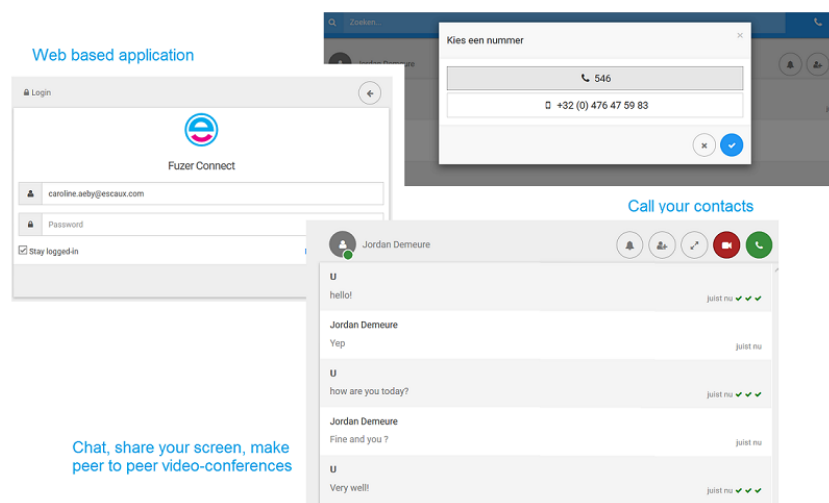


FIGURE II – Interface de la Web Application Connect Me (Image provenant de la documentation officiel d'Escaux <https://www.escaux.com/docs/ConnectMe.html>).

4.2 Les solutions de Web Analytics

Une solution de Web Analytics (ou solution d'audience d'un site web) est une solution serveur permettant de récolter des informations sur l'utilisation d'une application web par l'ajout d'un script Javascript sur cette dernière. Ce script Javascript permet de demander au serveur de la solution de Web Analytics d'envoyer un fichier. Ce fichier contient un autre code Javascript qui comprend des fonctions propres à la solution de Web Analytics et induit également la génération d'un cookie sur la page web, permettant de distinguer de manière anonyme les différents utilisateurs. Une solution de Web Analytics possède aussi une interface web de visualisation pour l'affichage des informations collectées et peut générer des rapports sous divers formats, entre autres PDF, HTML, CSV.

Pour récupérer des informations en provenance d'une application web avec un solution de Web Analytics, il faut ajouter du code Javascript propre à chaque solution de Web Analytics. Ce code permettra d'envoyer des informations au serveur contenant la solution de Web Analytics. Ces informations sont des informations en rapport avec l'utilisation que les visiteurs ont du site web. D'autre information tel que l'heure de visite, le navigateur utilisé, le type de périphérique (ordinateur, smartphone,) sont déjà collectées par défaut par les solutions de Web Analytics.

Plus précisément, les solutions de Web Analytics récupèrent des informations contenant quatre variables pour les définir, dont deux de ces variables sont optionnelles. Ces quatre variables sont la catégorie de l'évènement, l'action de l'évènement, le nom de l'évènement et la valeur de l'évènement qui doit être une valeur numérique, le nom et la valeur étant les valeurs optionnelles. Le code Javascript ajouté dans l'application pour suivre l'utilisation doit donc systématiquement envoyer au serveur au moins deux de ces quatre valeurs pour chaque évènement que l'on veut suivre. Le fait de devoir ajouter toutes ces variables directement dans le code de l'application web impose de connaître à l'avance les valeurs à associer à ces variables, puisqu'une fois définies, ces valeurs apparaîtront dans les rapports générés par la solution de Web Analytics. En cas de changement d'une de ces valeurs ou si une de ces valeurs est fausse, il faut retourner dans le code de l'application pour le mettre à jour. Or, on ne peut pas se permettre de redéployer certaines solutions pour un aussi petit changement, il faudrait donc attendre un prochain déploiement pour que la modification soit prise en compte. Cela entraînera des erreurs dans les informations récupérées pendant toutes ces périodes de latence, faussant les analyses que l'on aurait pu en tirer. En plus de cet inconvénient, un autre problème est présent : l'exécution du code qui envoie les informations est synchrone, ce qui engendre un ralentissement et peut parfois mener à des blocages en cas de problèmes de connexions avec le serveur contenant la solution de Web Analytics. Il est possible de rendre ce code asynchrone, mais cela alourdit alors le code de la page et demande une révision complète de l'application pour éviter tout oubli.

Certaines solutions de Web Analytics ont ajouté un module complémentaire nommé un Tag Manager (gestionnaire de balises). Ce Tag Manager s'ajoute au site web de la même manière que la solution d'analytique classique. Il faut d'ailleurs supprimer le code de la solution d'analytique classique en cas d'ajout du code du Tag Manager. Le Tag Manager offre toute une interface de gestion des valeurs envoyées pour l'analytique. Il est subdivisé en trois catégories, les variables, les déclencheurs et les balises. Cette solution offre un outil de prévisualisation qui une fois démarrée permet de voir ce que chaque action déclenche. Cela permet de pouvoir bien gérer les différents déclencheurs, variables et balise à créer.

Les variables peuvent contenir plusieurs types d'informations, comme des constantes permettant une homogénéité dans certains cas de valeurs, du code Javascript personnalisé, la configuration pour communiquer avec la solution d'analytique et des variables préconfigurées permettant de récupérer des informations sur l'élément cliqué, l'url de la page, etc.

Le déclencheur permet de choisir quand un évènement sera déclenché, il contient par défaut des déclenchements par rapport au clic, vue de page, minuteur. Ces déclencheurs peuvent être configurés pour ne se déclencher que sous certaines conditions. Un déclencheur avec beaucoup

d'intérêt est le déclencheur pour des événements personnalisés. Ce dernier permet de capturer un événement codé dans la page de la même manière que précédemment mais l'événement ne possède alors qu'un nom et une valeur, ce qui apporte une plus grande flexibilité (on y reviendra quand on parlera des balises).

La balise, quant à elle, permet de formater les informations à envoyer à la solution d'analytique suivant le déclencheur ainsi que le nombre de fois que l'on souhaite ajouter cette valeur dans l'analytique. Par exemple, s'il y a eu ajout d'un composant dans la web app et que celui-ci a été vu au moins une fois, il y a eu déclenchement une fois par utilisateur. Cette balise contient donc ce qui définit un événement, soit sa catégorie, l'action et l'objet de l'action, conférant l'avantage de pouvoir séparer l'action effectuée dans le code de la page web et ce qui est envoyé à l'analytique.

Reprenons l'exemple des événements en utilisation classique. Maintenant, certains des événements écrits en dur comme des clics sur des éléments peuvent, suivant certaines conditions, être configurés uniquement dans le Tag Manager. Le suivi de navigation dans le site est aussi plus aisé car il est également configurable dans le Tag Manager. Pour les événements personnalisés, il est possible, comme précédemment, de les ajouter dans le code mais il n'est alors plus nécessaire d'ajouter toutes les informations (catégorie, action et nom de l'événement) dans le code, il suffit de lui attribuer un nom et une valeur (souvent un prix, d'où la solution de marketing) et de choisir dans le Tag Manager les valeurs pour la catégorie, l'action et le nom de l'événement. Le Tag Manager permet une réelle supervision du code sans devoir prendre en compte l'implémentation dans le code. Tout comme le Web Analytics, le code de suivi ajouté dans la page du site web ajoute un fichier contenant le code Javascript permettant la récupération des éléments avec le Tag Manager.

4.2.1 Google Analytics (GA) et Google Tag Manager (GTM)

4.2.1.1 Google Analytics

GA est un produit développé par Google, offrant donc les avantages et les inconvénients classiques des produits Google. Un de ses avantages est d'être fortement utilisé et donc d'être maintenu à jour, ce qui en fait un allié fiable sur le long terme. Cependant, vu qu'il appartient à Google, cela veut dire que toutes les données qui transitent par GA sont analysées par Google et que toutes les informations sont stockées dans le cloud Google. À la suite de la mise en place du règlement général sur la protection des données (RGPD, ou plus connu sous l'abréviation anglaise GDPR), il faut faire attention à ce point très important qu'est la protection des données et ce même si, selon les informations fournies par Google, leurs produits sont en parfait accord avec le RGPD depuis le 25 mai 2018[?].

GA permet dans sa conception une très grande précision et personnalisation de ce que l'on collecte, mais peut représenter un travail conséquent pour l'équipe de développement puisqu'il faut ajouter des bouts de codes à chaque action sur la page. Il est aussi nécessaire de bien choisir le nom ainsi que les propriétés associées aux actions de l'utilisateur. Un changement dans ces noms peut engendrer une inconsistance voire une incohérence des données récupérées.

Cependant, il existe un utilitaire, nommé Google Tag Manager (GTM), permettant de gérer des tags sur un site sans devoir implémenter de nouveau code dans ce dernier. Avec GTM, il semblerait que l'implémentation de GA soit plus aisée car il permettrait d'alléger la partie intégration ainsi que la qualité des noms utilisés pour les événements.

4.2.1.2 Google Tag Manager

GTM est une solution qui facilite la gestion des données récupérées pour un site web. Il est basé sur une gestion en trois niveaux, la création de variables, de déclencheurs et enfin de balises (en anglais tags). Comme expliqué précédemment, les variables sont les informations que l'on peut récupérer de la page web ainsi que des données utilitaires, par exemple l'identificateur

de suivi de GA ou du code personnalisé pour récupérer une information spécifique d'une page. Une bonne partie des éléments d'un site web possède déjà une variable associée créée par GTM, mais il faut les activer pour y avoir accès. Les déclencheurs sont tous gérés par GTM mais on peut choisir qu'ils ne se déclenchent que lorsque certaines conditions sont remplies. Ils peuvent aussi être déclenchés par des événements personnalisés écrits dans le code mais dont il suffit simplement de préciser un nom[?].

Dans le cas de GTM, les seules balises qui nous intéressent sont celles incluant à Google Analytics - Universal Analytics à qui demandent le type de suivi souhaité. Le type de suivi contient les informations que GA récupérera. Ces informations seront envoyées à GA une fois que les déclencheurs associés s'activeront quand un utilisateur effectuera une action remplissant les conditions prédéfinies.

4.2.1.3 Couplage de GA avec GTM

L'utilisation des deux solutions GA et GTM couplées permet de séparer les événements de la page des informations envoyées dans GA. Cela permet de n'ajouter dans l'application que le code nécessaire aux événements qui ne peuvent être récupérés d'une autre manière. La conséquence directe est que le code à ajouter dans la solution est grandement réduit et, en cas de changement de nom de métrique ou d'abandon de la métrique, il suffit juste de supprimer la balise et non de devoir remodifier le code.

4.2.2 Matomo

Matomo offre des services similaires à ceux offerts par Google Analytics, à la différence que ceux de Matomo sont open-source et doivent être hébergés au sein de la société pour la version gratuite. Néanmoins, cela confère l'avantage de posséder les informations en local et offre la possibilité de récupérer les informations via la base de données en plus de l'API déjà proposée par Matomo.

4.3 Transformation d'informations

4.3.1 Exporter Prometheus, Prometheus et Grafana

L'outil de supervision Prometheus travaille avec des exporters Prometheus et l'outil de visualisation Grafana. Ces outils ont été développés pour être utilisés ensemble et forment un système cohérent de monitoring et d'alerting (Figure ??). Dans la figure ??), la partie inférieure est celle qui nous intéresse le plus. En effet, c'est dans celle-ci que l'on retrouve les exporters, le serveur Prometheus et la visualisation via Grafana. Comme on peut le voir, les exporters Prometheus qui sont des ETL (Extract-Transform-Load) permettent de récolter des données provenant d'une cible définie. De plus, ils formatent et exposent ces données de telle sorte qu'elles puissent être récupérées par Prometheus. Ces exporters sont des logiciels qui peuvent être écrits dans différents langages de programmation, l'exporter qui nous intéresse est un exporter créé par Escaux, comprenant une requête SQL et formatant ensuite les données en regroupant ensemble les données d'intérêt. Les données exposées sont ensuite récupérées par Prometheus à intervalle de temps régulier, représentée par la partie à Retrieval du schéma, et stockées dans une base de données spécifique, la partie à TSDB du schéma, avec un format en séries temporelles, c'est-à-dire un format basé sur un classement chronologique. Ce format diffère de celui des bases de données classiques dans la mesure où les données récoltées ont des caractéristiques spécifiques connues. Ainsi, Prometheus est capable de compresser les données, optimisant donc l'utilisation des ressources mémoire, processeur et disque¹[?, ?].

1. Pour en savoir plus sur Prometheus :
<https://linuxfr.org/news/decouverte-de-l-outil-de-supervision-prometheus>

En ce qui concerne Grafana, cette solution permet de créer plusieurs tableaux d'affichage afin de regrouper de manière logique les informations à afficher. De plus, ces tableaux permettent à Grafana de ne devoir envoyer à Prometheus que des requêtes en rapport avec ce qui est nécessaire pour l'affichage et quand un utilisateur s'en sert. Les requêtes ne sont effectuées que dans le cas de rechargement de la page ou de changement d'intervalle de temps pour lequel l'utilisateur veut voir les métriques ou encore quand l'intervalle de temps sélectionné est du type les trente dernières minutes, auquel cas l'affichage doit se rafraîchir afin de présenter des informations cohérentes. Les requêtes effectuées par Grafana sont écrites en PromQL, langage permettant de faire des requêtes à Prometheus[?].

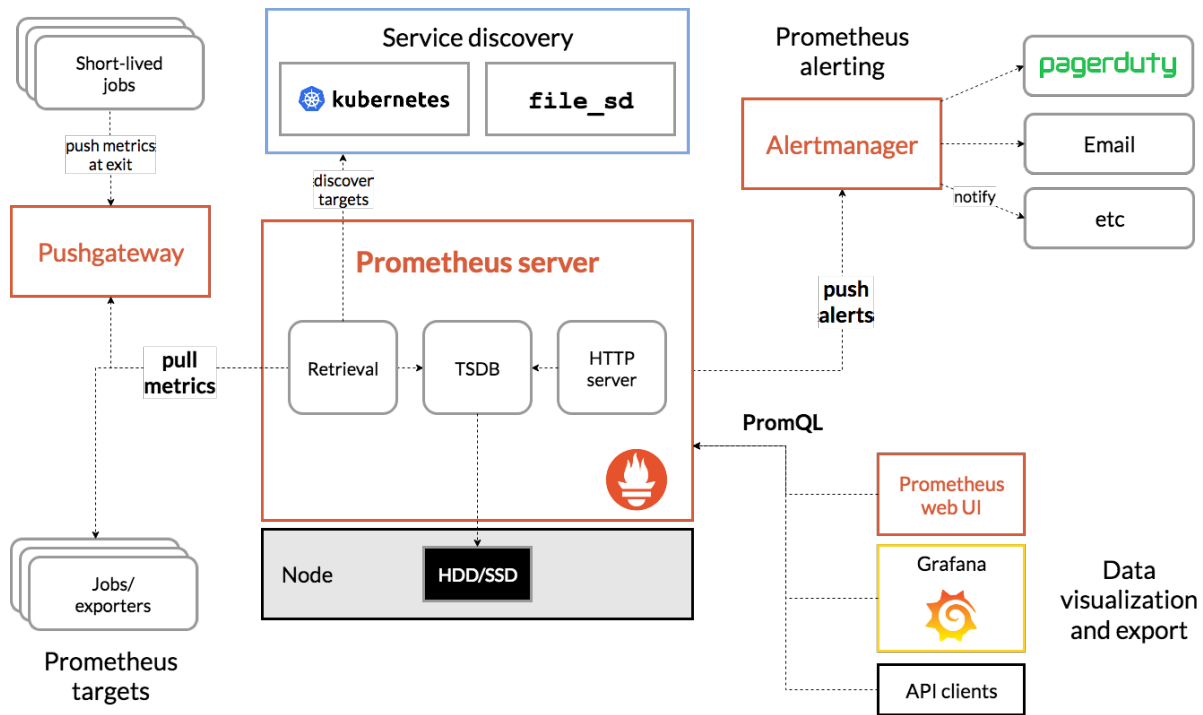


FIGURE III – Architecture de Prometheus et de certains composants associés.
<https://prometheus.io/docs/introduction/overview/>

4.3.2 Logstash, Elasticsearch et Kibana

Elasticsearch est un outil d'indexation et de stockage conçu pour fournir des résultats rapides. Il travaille avec Kibana, un outil de visualisation, et Logstash, un ETL (Figure ??). Logstash a d'abord été conçu pour parser des logs mais ensuite, ses fonctionnalités se sont étendues afin de permettre la récupération d'informations dans une base de données. Logstash, à l'instar de l'exporter Prometheus mentionné ci-avant, peut également travailler avec plusieurs requêtes SQL suivant la manière dont on veut récupérer et surtout formater les informations dans Elasticsearch. Comme pour l'exporter Prometheus, cette tâche se réalise à intervalles de temps réguliers. Cependant, Kibana possède un avantage intéressant par rapport à Grafana. En effet, si les données d'une colonne de la base de données d'Elasticsearch sont des timestamps, il est capable d'utiliser cette colonne comme vecteur temporel pour afficher les données. Cela permettrait d'effectuer moins de requêtes et par conséquent, de séparer sur la base de données les moments des requêtes entre la récupération des informations de l'analytique et l'extraction de ces données pour la visualisation dans Kibana[?, ?, ?].

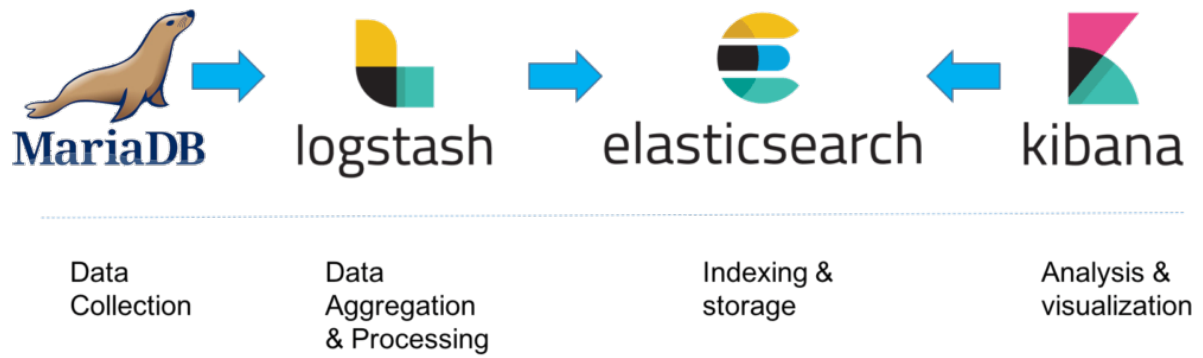


FIGURE IV – Architecture de la suite appelée ELK regroupant Logstash, Elasticsearch et Kibana et associée à MariaDB (Image modifiée provenant de <https://logz.io/learn/complete-guide-elk-stack>).

5. Tests et analyse

5.1 Choix de la solution de Web Analytics

La solution demandée par le client comportait du Web Analytics et le client souhaitait utiliser Google Analytics puisqu'un grand nombre de solutions proposées par Google sont déjà en place chez lui. Il a d'abord fallu me renseigner sur la solution de Google pour le Web Analytics. Après avoir lu la documentation, je me suis rendu compte que, malgré les exemples donnés, cela restait vague sur la manière de l'implémenter dans un site web. J'ai donc décidé de l'implémenter sur un site que je possédais pour effectuer des tests et voir comment se passe l'intégration de l'analytique dans le site.

Après avoir réussi à faire fonctionner la récupération de données dans la page web, je me suis rendu compte que ce processus était long, fastidieux et assez lourd d'implémentation. En effet, les valeurs qui allaient être présentes sur l'interface web de Google Analytics devaient également être présentes dans le code de l'application. Par conséquent, il fallait être sûr dès le départ de la manière dont on voulait structurer les informations à envoyer sachant que celles-ci comportent : une catégorie, une action et un nom d'événement[?]. Cela implique un retour obligatoire dans le code lorsqu'il est nécessaire de changer les valeurs des événements après une erreur éventuelle, rendant le code très difficile à maintenir. Le code est fortement alourdi par cette méthode de fonctionnement puisque du code doit être ajouté pour chaque action dont on veut récupérer l'information. De plus, les envois des données vers le service de GA se font de manière synchrone, ce qui ralentit la page et pourrait la bloquer lors d'un problème de communication avec GA ou au sien de la page elle-même. Il y a une possibilité de contourner ce problème de synchronisme en rendant le code asynchrone mais cela alourdit davantage la partie à ajouter pour l'analytique.

J'ai donc cherché une autre solution pour rendre cette implémentation plus légère et j'ai trouvé Google Tag Manager proposé par Google pour rendre l'implémentation de Google Analytics plus facile et surtout plus légère au niveau du code à ajouter dans l'application.

La solution GTM possède déjà un système embarqué pour récupérer des informations rien que par sa présence sur la page. Comme pour GA, il est possible d'ajouter des événements dans la page mais contrairement à ceux précédemment explicités, ceux-ci ne possèdent qu'un nom et une valeur. Tout le reste se fait via une interface de gestion en ligne qui permet d'ajouter les tags qui contiennent les mêmes valeurs que les événements de Google Analytics et ce sont justement ces valeurs qui sont envoyées pour l'analytique[?].

Ce système permet une séparation très intéressante des informations présentes dans l'application web et dans ce que l'analytique récupère. Il est également possible de créer du code JavaScript personnalisé sans non plus devoir l'ajouter dans l'application, ce code permettant par exemple de récupérer les changements de l'URL par rapport à un événement préexistant contenu dans le Tag Manager, outil très utile dans une Single Page App. Ce genre de page ne change pas réellement d'URL mais change des informations après un #, impliquant qu'il n'y a pas de changement réel de page. L'application d'Escaux étant une Single Page App, une telle personnalisation est plus intéressante. Ce simple exemple prouve que ce système est indispensable car sans cela, il faudrait ajouter ce code à chaque changement de page dans l'application.

Le Tag Manager permet l'asynchronisme des requêtes et l'injection de code personnalisé, il possède des déclencheurs très puissants et sert de tampon entre le code de l'application et les informations pour l'analytique.

Un détail se présente cependant opposé à l'utilisation des services Google, le RGPD (plus connu sous l'abréviation anglaise GDPR) ou Règlement Général sur la Protection des Données. Malgré la conformité au RGPD, certaines options nécessaires à cette conformité n'étaient pas activées par défaut. En effet, Google ainsi que les autres membres du GAFAM ne brillent pas toujours par rapport à l'utilisation qu'ils font des données de leurs utilisateurs. Par conséquent, j'ai envisagé d'utiliser d'autres solutions de Web Analytics, Google ne devant pas être le seul à proposer ce type de solutions. Parmi les autres solutions envisageables trouvées, beaucoup d'entre elles étaient payantes, sauf deux : Matomo, formellement appelé Piwik, et Open Web Analytics. J'ai informé le client de ce fait et demandé par conséquent si une solution payante était envisageable ou s'il préférerait les solutions gratuites comme Google Analytics.

L'intérêt du client se porte sur les solutions gratuites, non pas pour leur gratuité mais pour leur caractère open source, ce qui donnait accès au code et permettait l'installation en local, permettant ainsi de connaître où et comment sont stockées les informations. Cependant, la condition pour utiliser ces solutions était que celles-ci soient au moins aussi puissantes que la solution proposée par Google puisque, comme souvent, Google propose des solutions très complètes gratuitement mais avec un défaut en termes de confidentialité des données.

Concernant les deux solutions open source, j'ai rapidement écarté Open Web Analytics car il ne possédait pas de Tag Manager et est moins suivi que Matomo, à la limite même de l'abandon (Figure ?? et Figure ??). Au moment de la réflexion entre Matomo et Open Web Analytics, la dernière mise à jour de ce dernier datait du 11 juin 2018. Le quasi-abandon de cette solution ainsi que la différence de contributeur entre les deux solutions envisageables à orienter mon choix vers Matomo.

Matomo est une solution gratuite si elle est hébergée en local. Pour effectuer les tests, je l'ai utilisée avec l'image Docker fournie par la société elle-même pour avoir un premier aperçu de la solution[?]. Matomo se comporte comme Google Analytics et possède également un Tag Manager. Au vu des inconvénients que possèdent Google Analytics, j'ai directement effectué mes tests avec le Tag Manager de Matomo car la documentation de Matomo sans utilisation du Tag Manager était similaire à celle de Google Analytics. Les tests effectués indiquaient que les deux solutions (GA et Matomo) étaient très semblables, mais le client voulait être absolument certain de cette similitude. Par conséquent, j'ai implémenté la récupération des données via Google Tag Manager dans l'application dEscaux pour juger son comportement et ensuite, j'ai fait de même avec Matomo, toujours dans un conteneur Docker. J'ai dû reconfigurer Matomo à la suite d'un problème avec la solution Docker pour Windows qui présentait un défaut avec les volumes partagés, faisant que la configuration avait été supprimée.

Après la démonstration de ces résultats auprès dEscaux, Matomo fut accepté du fait qu'il était Open source et conforme par défaut au RGPD. Sa conformité au RGPD ne fut pas une surprise étant donné que Matomo possède comme client la Commission Européenne et qu'il fournit toute une interface de gestion par rapport au RGPD, dont des recommandations pour respecter ce dernier.

Vu que Matomo nécessitait d'être installé sur un serveur de la société, j'ai dû attendre que la société me mette un à disposition. J'ai entre-temps commencé à chercher une solution pour récupérer les données de Matomo en vue de les coupler avec la solution de visualisation que voulait utiliser la société. Il y a deux manières de procéder : soit récupérer l'information directement dans la base de données où sont stockées toutes les informations, soit récupérer l'information via l'API fournie par Matomo.



FIGURE V – Information par rapport au succès de Matomo (Capture écran de GitHub <https://github.com/matomo-org/matomo>, réalisée le 04/05/2019).

FIGURE VI – Information par rapport au succès dOpen Web Analytics (Capture écran de GitHub <https://github.com/padams/Open-Web-Analytics>, réalisée le 04/05/2019).

5.2 InfluxDB : une fausse bonne idée

Le client connaissait une solution de base de données orientée séries chronologiques, InfluxDB, et souhaitait savoir s'il était envisageable de l'utiliser comme base de données. InfluxDB possède l'avantage d'être directement associable avec les solutions que la société désire employer pour la visualisation des données. Après recherche, la littérature sur Matomo ne mentionnait aucune compatibilité avec cette base de données. Après un rapide test pour essayer de lancer Matomo avec InfluxDB, j'ai bien eu confirmation qu'il était impossible de les utiliser directement ensemble. Une solution semblait présente pour exporter les données provenant de MySQL vers InfluxDB, mais je n'ai pas pu la tester correctement car j'effectuais mes tests avec les applications dockerisées, la gestion de l'architecture dans le cas d'utilisation de script étant parfois assez difficile. De plus, cette solution aurait fait qualourdir l'architecture en possédant deux bases de données à la place d'une seule, juste pour avoir un miroir de la première avec un format différent. Cette solution a donc été écartée car trop lourde au niveau de l'implémentation et engendrant une surcharge pour l'architecture sans apporter un réel intérêt si non directement utilisable.

5.3 Transformation/formatage de données

L'accès direct à la base de données étant possible, il était plus intéressant de récupérer les informations par ce biais. L'autre solution aurait été de récupérer les informations présentes dans l'API mise à disposition par Matomo mais les API manquent parfois d'une certaine granularité pour bien filtrer ce que l'on souhaite réellement recevoir ou alors demandent de nombreuses requêtes différentes pour récupérer une donnée précise.

À première vue, la base de données de Matomo semblait confuse et la documentation en rapport avec son contenu était incomplète car elle présentait la base de données par rapport à ce qu'il est possible de récupérer via l'API. Il m'a donc fallu comprendre la logique sous-jacente de la base de données et j'ai ainsi pu en extraire un schéma qui ne contient que les informations en rapport avec les métriques intéressantes (Figure 6), le reste de la base de données servant à la gestion propre de Matomo[?].

La Figure ?? représente les tables de la base de données en rapport avec les visites et les événements déclenchés par les utilisateurs. La table `matomo_log_visit` contient toutes les visites effectuées sur les sites contenant le code de suivi de Matomo, certaines informations de cette table sont mises à jour tant que la visite n'est pas terminée, d'autres sont définies dès le début de la visite. C'est ainsi que tous les champs faisant mention de `id`, `first`, `entry`, `référer`, `visitor` et `config` sont définis dès la première action d'un utilisateur sur le site. Les champs référés représentent la manière dont un utilisateur entre sur le site et l'URL de la page précédente s'il était sur un autre site web avant d'arriver sur le site suivi. Les champs contenant les mentions `total`, `last` et `exit` se modifient au fur et à mesure des actions effectuées par l'utilisateur puisqu'ils représentent des informations dynamiques. La table `matomo_log_link_visit_action` contient les différents événements effectués au cours d'une visite par un utilisateur sur un site donné. La table `matomo_log_action` contient l'identificateur et le nom qui peut, bien que son nom indique le contraire, être le nom d'une action, d'un événement, d'une catégorie d'événement, d'une URL, etc. Les liens entre la table `matomo_log_action` et `matomo_log_link_visit_action` et les liens entre `matomo_log_action` et `matomo_log_visit` sont des liens possédant une référence aux valeurs de l'autre table partant du principe clé valeur.

5.4 Comparaison entre l'exporter Prometheus et Logstash

Une fois le fonctionnement de la base de données compris, il restait encore à en extraire les données, d'autant plus que les rapports automatiques générés par Matomo n'étaient pas assez

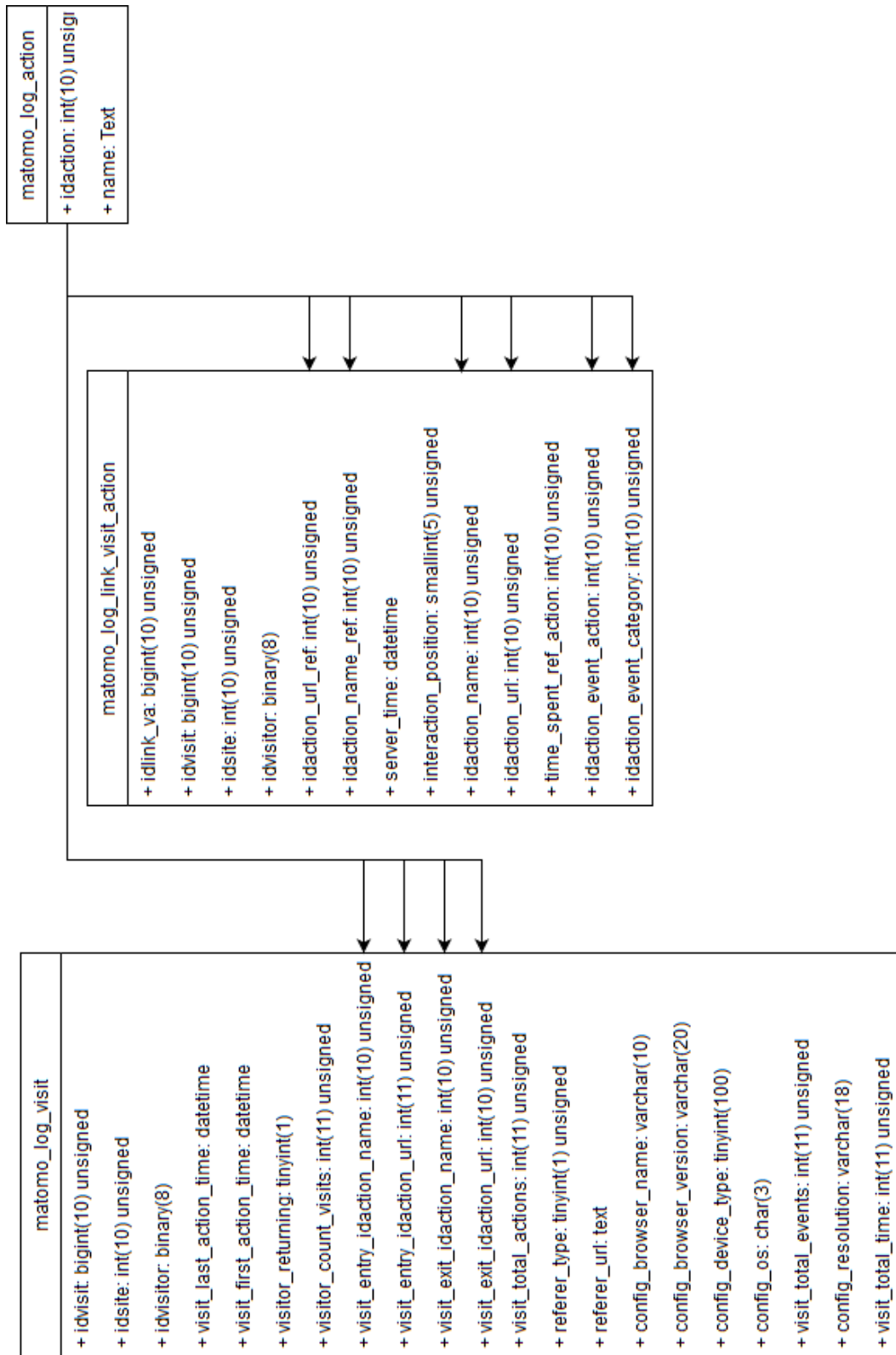


FIGURE VII – UML de la base de données de Matomo pour les informations intéressantes. Figure originale, élaborée à l'aide de <https://www.draw.io/>.

clairs et compréhensibles pour le client. Le client travaillant déjà avec Grafana et Kibana, il était intéressant d'utiliser l'une de ces solutions pour afficher l'information.

Pour ce faire, il fallait utiliser soit un exporter pour Prometheus, très utilisé avec Grafana et dont le client possédait déjà un équivalent pour monitorer une de leurs solutions, soit Logstash. Ce dernier permet également de récupérer des informations depuis une base de données pour les inscrire dans Elasticsearch qui travaille en accord avec Kibana, de la même manière que Prometheus travaille avec Grafana.

L'exporter Prometheus permettait de travailler avec une seule requête SQL et de spécifier les informations à exposer en vue d'être collectées par Prometheus via un dictionnaire en Python dans lequel les informations sont formatées. Logstash travaille avec plusieurs requêtes SQL afin de stocker dans Elasticsearch une nouvelle table contenant les informations stockées de manières chronologiques par rapport au timestamp présent dans la base de données d'origine. Bien que l'exporter Prometheus était la première solution proposée et quelle convenait, j'ai décidé d'explorer Logstash car elle permettait de regrouper l'information une fois par jour, ce qui n'est pas problématique dans le cas de l'analyse de l'utilisation d'une solution et aurait permis d'alléger le serveur en ne lui demandant pas de travailler en même temps à la récupération d'informations pour l'analytique et à l'affichage de ces valeurs dans un autre logiciel. L'exporter Prometheus, quant à lui, oblige une récupération plus fréquente des données pour garder une cohérence par rapport à la chronologie des données.

Toutefois, après présentation des deux solutions, le client a fait appel à un de ses collaborateurs pour lui demander son avis et ce dernier m'a appris que je métais mépris au sujet de Logstash et d'Elasticsearch. Les membres de l'équipe de développement préféraient travailler avec Prometheus qu'avec Elasticsearch et, s'ils sont amenés à utiliser ce dernier, ils emploient un exporter pour remettre les informations dans Prometheus. Logstash n'a pas été utilisé et je me suis concentré sur l'exporter Prometheus. Par conséquent, la solution de visualisation utilisée est Grafana. Cette dernière utilise le langage PromQL de Prometheus pour effectuer les requêtes quand elle interagit avec celui-ci. Il est facile de créer l'affichage des métriques une fois que l'on connaît les informations puisque l'on peut tester au préalable les requêtes dans l'interface de Prometheus. Cette interface présente les métriques exposées par l'exporter Prometheus, ce qui en facilite l'identification. Une fois les métriques correctement identifiées, différentes opérations peuvent être testées sur celles-ci via l'interface de Prometheus. Ensuite, ces opérations peuvent être sauvegardées dans Grafana.

6. Implémentation de la solution

6.1 Installation sur le serveur et choix des technologies associées

Une fois le serveur mis à disposition par le client, j'ai tout d'abord installé toutes les dépendances requises pour Matomo, lesquelles comprennent un serveur Web, PHP et une base de données. Pour le serveur web, je pouvais choisir entre Apache et Nginx. J'ai choisi Nginx parce qu'il est plus léger qu'Apache et que j'avais déjà travaillé avec cette solution. De plus, Nginx gère mieux la concurrence et les multiples connexions qu'Apache, ce qui est très intéressant pour mon utilisation[?]. En effet, cela évite d'utiliser trop de ressources puisque le serveur va surtout être utilisé pour traiter beaucoup de petites informations[?].

Une base de données est nécessaire à la solution de Web Analytics pour stocker les données d'utilisation du site web suivi ainsi que des informations sur la configuration de l'outil en lui-même, tel que les utilisateurs enregistrés, les balises, déclencheurs et variables utilisées pour le suivi. Par conséquent, j'ai dû trouver une solution pour la base de données et je me suis orienté vers MariaDB. MariaDB n'est pas détenue par une société et n'est donc pas en concurrence avec des solutions payantes équivalentes, telles que MySQL, un système de base de données détenu par Oracle[?, ?, ?]. De plus, MariaDB est plus performante et possède une meilleure gestion de la concurrence, ce qui est intéressant étant donné que les données de suivi peuvent arriver par paquets. Par ses bonnes performances, la vitesse de stockage de MariaDB est augmentée, ce qui est un avantage intéressant et non négligeable pour les solutions analytiques. PostgreSQL n'a pas été une solution envisagée car selon la documentation de Matomo, Piwik (la société travaillant sur Matomo) ne supporte officiellement que les solutions de type MySQL. Les demandes pour que PostgreSQL soit supportée n'ont jamais abouti[?, ?].

Ensuite, une fois les dépendances installées, j'ai poursuivi avec l'installation de Matomo et sa configuration via l'interface en ligne. Il a fallu vérifier le bon fonctionnement après installation car la société utilise un serveur proxy pour éviter de devoir sécuriser chaque serveur utilisé. Dès lors, les connexions externes se font via HTTPS mais, en interne, toutes les connexions sont en HTTP. Cependant, le trafic passe par des tunnels SSH, ce qui permet une gestion plus légère des certificats SSL/TLS utilisés. En prévision de l'utilisation en production de Matomo, j'ai ajouté un archivage des données via Cron, comme conseillé dans la documentation[?]. Cron est un outil présent sur Linux qui permet d'automatiser des tâches, telles que des scripts, et de les exécuter à un intervalle de temps régulier et/ou à une ou des heures précises. J'ai également utilisé Cron pour créer un backup de la configuration de Matomo et des plugins installés. Le client utilisant une solution nommée Backuppc pour effectuer un backup automatisé de ses différents serveurs, une sauvegarde de la base de données se effectuait déjà par défaut grâce à cette solution. Ainsi, j'ai créé un script qui permet de sauvegarder la configuration de Matomo et des plugins installés et non pour la base de données.

Le premier test consistait en de simples captures de clics, sans modification du code de Connect Me, afin de tenter d'obtenir des informations rapides. Cette version de l'application a ensuite été déployée sur le serveur de développement. L'équipe de développement a alors pu utiliser la version de Connect Me hébergée sur ce serveur, permettant d'obtenir des métriques et un premier aperçu des rapports générés par Matomo.

Pendant ce premier test, j'ai remarqué que la génération de données par les clics n'était pas

possible à cause de l'imprécision des éléments à cibler et de la non-utilisation d'identifiant dans l'application. En effet, le clic sur un élément précis, comme une icône, ainsi que son parent direct déclenche la même action mais n'est pas pris en compte par Matomo. Il faudrait donc doubler tous les déclencheurs, ce qui pose des problèmes en cas de changement du rendu graphique de l'application et, vu que ces rendus diffèrent suivant la taille de l'appareil du fait du caractère responsive design de l'application, le sélecteur CSS change. Cependant d'autres éléments, dont notamment la navigation à travers le site, sont très facilement gérés via l'interface de Matomo, il n'a donc pas fallu ajouter de code pour cela dans l'application.

6.2 Matomo

Pour être en accord avec le RGPD (GDPR), j'ai ajouté le code de tracking de Matomo dans un module séparé, dans lequel l'utilisateur choisit d'accepter ou non le tracking d'utilisation lors de sa première connexion avec un compte. Ce choix est lié au compte de l'utilisateur, donc il ne lui sera plus demandé d'accepter ou non le tracking, peu importe ses changements de périphérique. De plus, si un second utilisateur utilise le même périphérique que le premier utilisateur mais qu'il [le second utilisateur] avait refusé le suivi, le script est directement supprimé. L'intégralité du code de tracking ainsi que les codes pour les événements personnalisés de Matomo se trouvent dans un module complètement séparé du reste du code. Cela permet de simplement supprimer le chargement de ce module si l'on souhaite le désactiver voire le supprimer. Il n'existera alors plus de tracking d'utilisation pour l'application. Certes, certains événements ont été émis uniquement pour être utilisés par l'analytique, mais ces événements ne bloquent en rien l'application si l'analytique ne fonctionne pas ou a été supprimé. L'utilisateur peut aussi changer à tout moment son choix d'être suivi ou non dans les paramètres et ce choix s'applique à l'ensemble de ses périphériques. Ainsi, si un utilisateur avait précédemment accepté d'être suivi mais qu'il souhaite ne plus l'être à présent, la modification de ce choix dans les paramètres entraîne un rechargement de la page afin de supprimer le script de suivi.

Toutes les informations de suivi à envoyer vers le serveur contenant Matomo sont envoyées vers l'URL `https://webstat.connect.fuzer.net/` que le proxy de la société redirige ensuite comme il se doit vers le serveur de Matomo.

6.3 Instrumentalisation de Connect Me

La majorité des événements et des actions effectués sur Connect Me ne pouvaient pas être récupérée via l'utilisation des déclencheurs préprogrammés proposés par le Tag Manager de Matomo. Par conséquent, j'ai dû comprendre la logique de programmation de Connect Me afin de créer un nouveau module dans le code de l'application. Ce module, appelé par abus de langage module Matomo, comprend tout le code en rapport avec le Web Analytics, tels que le code gérant l'acceptation du suivi implémenté pour la conformité au RGPD, le chargement et la suppression du script fourni par le Tag Manager et tous les événements et valeurs à envoyer au serveur hébergeant Matomo.

L'instrumentalisation de Connect Me est simple. Lorsqu'un utilisateur effectue certaines actions, typiquement les actions dont on veut récupérer des métriques, un événement AngularJS est déclenché. Le module Matomo contient des listeners par rapport à ces événements et en réaction à l'émission de ces événements, ils envoient des informations au serveur contenant Matomo. Ces informations peuvent être des événements personnalisés, des variables ou une combinaison des deux.

Cette tâche d'instrumentalisation de Connect Me fut longue et difficile car certaines parties du code de Connect Me ont dû être réécrites pour pouvoir émettre un événement et, pour chaque ajout de listener, je devais créer dans le Tag Manager la balise associée, le déclencheur en rapport

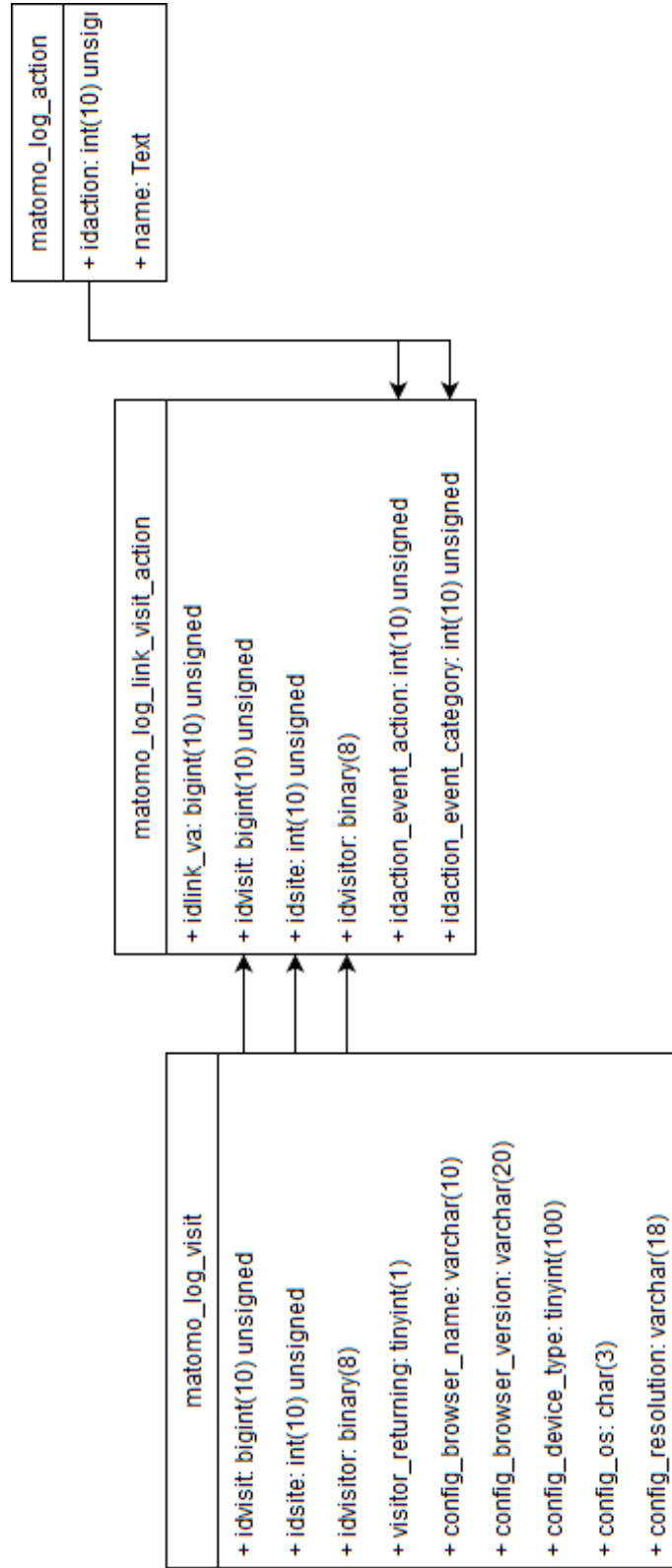


FIGURE VIII – UML de la base de données de Matomo pour les informations utilisées dans le monitoring. Figure originale, élaborée à laide de <https://www.draw.io/>.

avec l'évènement personnalisé et la variable si la valeur de celle-ci provient du code. Après chaque ajout, j'ai vérifié que tout était bien récupéré par la solution de Web Analytics.

6.4 Installation de l'exporter Prometheus

L'exporter Prometheus utilisé est un logiciel écrit en Python, permettant de faire des requêtes sur une base de données et de les exposer pour Prometheus. Pour faciliter le déploiement de la solution proposée, le programme écrit est archivé dans un package Debian (.deb) permettant l'automatisation de son installation sur un système Linux compatible avec le format du paquet créé. Escaux utilise GitLab, une plateforme de versioning, embarquant des outils permettant également l'archivage en package Debian. En créant un repository sur GitLab, si GitLab a été configuré en vue d'une telle utilisation, l'archivage s'effectue automatiquement à chaque publication du code sur la plateforme. Cela permet de tester si le code écrit respecte bien les règles pour un package Debian et de récupérer facilement le package une fois ce dernier archivé. L'archive contient les instructions nécessaires pour configurer l'exporter Prometheus en tant que service système. Ainsi, la distribution du logiciel se fait de manière plug-and-play et sa gestion en production en est facilitée.

Après avoir pu utiliser cet exporteur en production et vérifier les données nécessaires pour le monitoring de Connect Me, les champs de la base de données qui ont été utilisés ont été réduits (Figure ??). Cette réduction est aussi due à la contrainte de récupération d'informations qui ne permettait pas d'utiliser certaines valeurs variant au cours du temps, telles que la quantité d'actions effectuées lors d'une visite.

6.5 Visualisation des données d'utilisation

Pour permettre la visualisation des données d'utilisation de Connect Me, il a fallu modifier la configuration de Prometheus installé sur un serveur d'Escaux pour que celui-ci aille récupérer les informations exposées par l'exporter et créer un tableau d'affichage dans Grafana, aussi installé sur un serveur de la société pour récupérer les données et les afficher dans des graphes. Ces graphes montrent surtout les tendances par rapport aux actions effectuées par les utilisateurs. L'interface web de Matomo a aussi été conservée comme affichage car contrairement à Grafana, elle permet d'afficher de manière précise la quantité d'actions effectuées.

6.6 Architecture de la solution

Comme présenté sur la Figure ??, les serveurs de la société ne sont pas directement exposés mais passent par un serveur proxy. La connexion entre les serveurs et le proxy s'effectue via des tunnels SSH permettant ainsi une scalabilité des connexions, non nécessaire pour la solution de Web Analytique, mais utilisée par Connect Me pour les appels téléphoniques. Néanmoins, cela permet aussi de chiffrer les données, ce qui renforce la protection de ces dernières en cas d'intrusion dans le réseau. Sur le serveur analytique, j'ai donc installé Matomo, un serveur web Nginx avec les dépendances PHP nécessaires au bon fonctionnement de Matomo, la base de données MariaDB et l'exporter Prometheus exposant les données pour le serveur contenant Prometheus et Grafana. Ce dernier récupère les informations du serveur analytique correctement formatées et en permet la visualisation.

La Figure ?? représente l'architecture d'Escaux (Figure ??) au sein de laquelle la solution élaborée a été déployée. Cette nouvelle partie du schéma représente la solution de Web Analytics choisie, Matomo avec sa base de données, MariaDB, ainsi que l'exporter Prometheus associé. La solution de visualisation utilisée est donc Grafana qui récupère les informations à afficher via Prometheus. La partie de gauche représente toujours le côté client. Cependant, en plus d'envoyer des informations vers la partie serveur de l'application Connect Me, des informations de suivi

sont maintenant envoyées vers `https://webstat.connect.fuzer.net/` qui pointe aussi vers le serveur proxy dEscaux et le proxy redirige ces requêtes vers le serveur contenant la solution Matomo.

La partie de droite du schéma représente toujours la partie interne de la société avec, dans la partie supérieure, les échanges d'informations propres à l'application Connect Me entre le proxy et les serveurs contenant la partie backend de Connect Me. La partie inférieure quant à elle représente la deuxième solution mentionnée au début de ce présent travail, c'est-à-dire l'utilisation d'une solution de Web Analytics hébergée au sein de l'infrastructure de la société et comportant une manière de récupérer les informations en vue de leur récupération pour la solution de visualisation. Par conséquent, la partie inférieure comprend un serveur sur lequel est installé la solution de Web Analytics, Matomo avec sa base de données et l'exporter Prometheus. La solution de Web Analytics communique avec le proxy pour envoyer, lors du démarrage de l'application Connect Me chez un client, le code contenant tout le script permettant la récupération des données. Ensuite, le proxy, en plus de transférer toutes les informations propres à Connect Me, renvoie aussi les informations en rapport avec le Web Analytics vers le serveur contenant Matomo. Matomo va alors récupérer ces informations et les ajouter dans sa base de données. En parallèle, l'exporter Prometheus récupère les nouvelles informations présentes dans la base de données pour les exposer sur un port accessible par Prometheus uniquement via le tunnel SSH. Prometheus vient récupérer ses informations par défaut toutes les minutes, les traite et les ajoute dans sa base de données. Enfin, Grafana effectue des requêtes vers Prometheus pour récupérer des informations en rapport avec le tableau d'affichage consulté par un utilisateur.

6.7 Documentation

Une documentation a été laissée aux développeurs dEscaux travaillant sur Connect Me. Cette documentation contient les informations nécessaires pour être capable d'ajouter et de modifier des événements dans Matomo ainsi que dans Connect Me. Elle contient aussi les instructions et les liens vers la documentation en cas de nécessité de migrer Matomo ou en cas de réinstallation du serveur. Cette documentation est rudimentaire et contient le strict minimum pour pouvoir vite prendre en main l'ajout des éléments précédemment cités. Pour une documentation plus complète, il reste conseillé d'aller lire la documentation officielle de Matomo contenant un descriptif complet sur les diverses possibilités qu'offre cet outil de Web Analytics. Le lecteur intéressé pourra trouver une copie de la documentation laissée à l'entreprise en Annexe de ce document.

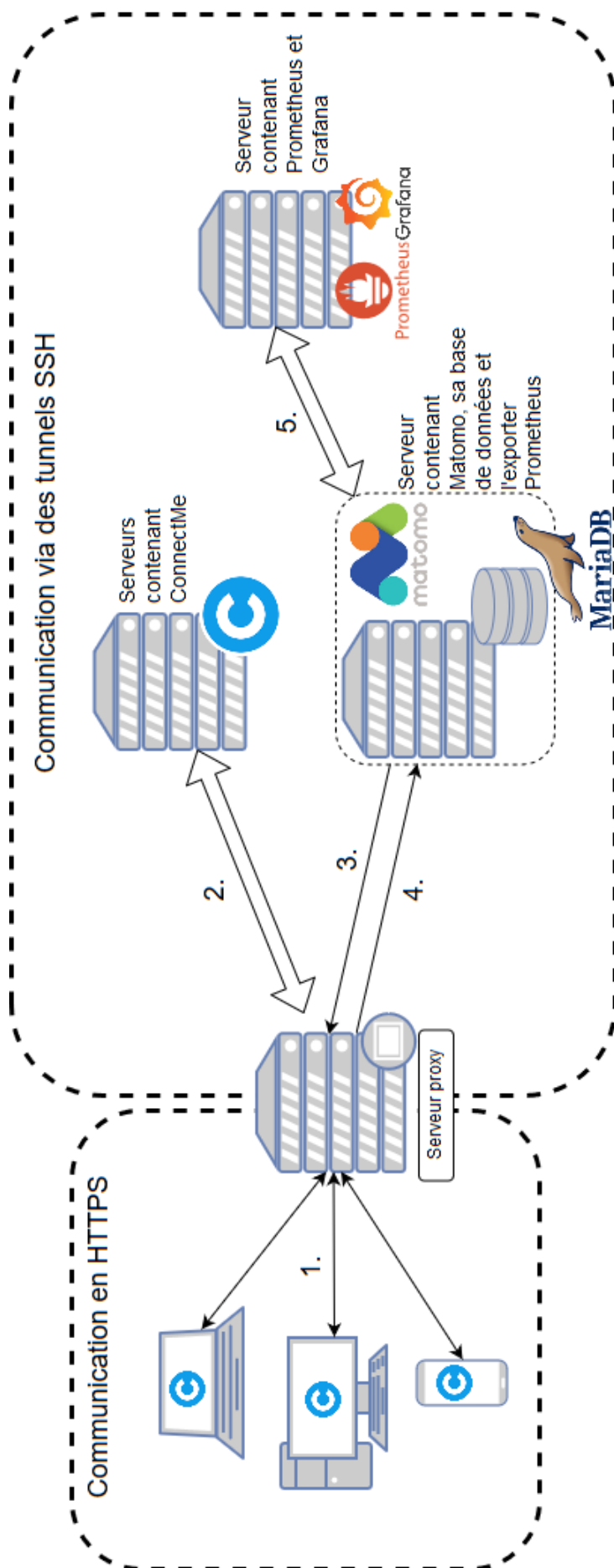


FIGURE IX – Représentation de l'architecture de la solution. Figure originale, élaborée à l'aide de <https://www.draw.io/>

1 : Echange d'informations entre le serveur proxy et les différents périphériques dans lesquels peut fonctionner Connect Me. Ces informations regroupent les messages envoyés et reçus, les appels et autres types de communication ainsi que le code pour l'analytique et l'envoi des différents événements à Matomo.

2 : Echange des informations propres à Connect Me entre le serveur proxy et l'un des serveurs de Connect Me via des tunnels SSH.

3 : Envoi du tracker de l'analytique vers le serveur proxy pour le distribuer au client.

4 : Retour des informations par rapport au suivi des clients ayant accepté le suivi.

5 : Récupération par Prometheus des informations exposées par l'exporter Prometheus présent sur le serveur d'analytique.

7. Conclusion

L'ensemble des solutions finalement proposées se divisent en deux catégories. D'une part, pour le Web Analytics, la solution choisie est Matomo, avec Nginx et MariaDB pour la prise en charge du serveur web et de la base de données, respectivement. D'autre part, pour le monitoring, a été retenue la suite de solutions travaillant avec Prometheus, c'est-à-dire l'exporter Prometheus, Prometheus lui-même et l'outil de visualisation Grafana. Cet ensemble de solutions a permis de garder l'intérêt des rapports générés par Matomo pour avoir des informations précises quant à l'utilisation de Connect Me. Quant à la visualisation fournie par Grafana, elle permet d'avoir un aperçu rapide des tendances d'utilisation de l'application. Cela permettra une compréhension plus claire et plus rapide de l'utilisation de Connect Me, la version instrumentalisée de Connect Me n'étant pas encore en production.

Toutes les solutions proposées ont été intégrées dans l'environnement d'Escaux et l'application Connect Me a pu être instrumentalisée pour permettre l'envoi des données vers Matomo, procurant ainsi la possibilité de récupérer une large quantité de données par rapport à l'utilisation de l'application. Néanmoins, le présent travail ayant été réalisé sur un court intervalle de temps sur le plan pratique, Matomo n'a pas pu être installé dans l'environnement adéquat. Par conséquent, il faudra migrer la solution sur un autre serveur pour s'adapter à la logique de l'infrastructure de la société.

Sur un plan technique, les contraintes liées à l'infrastructure préexistante chez Escaux ont représenté un challenge très instructif en me pliant à certaines règles en place. Par exemple, l'installation des différents modules nécessaires à l'utilisation de PHP avec Nginx demandait certains packages, mais ceux-ci n'étaient pas redistribués par la société car cette dernière possède un serveur miroir pour la distribution des paquets afin de filtrer les paquets utilisés et en gérer la provenance. Ayant recours à Nginx et PHP dans la solution proposée à Escaux, j'ai dû contourner cette contrainte afin de mettre la solution en place.

Sans les différentes contraintes rencontrées, je pense que le travail effectué aurait été, certes tout aussi intéressant, mais moins instructif sur le fonctionnement d'un système en production. En effet, sans l'infrastructure d'Escaux, les contraintes liées à cette infrastructure n'auraient pas été présentes, mais la solution proposée n'aurait alors pas pu être implémentée dans un environnement de production réel. Or, cette implémentation a justement représenté une part importante de ce travail, le principe étant d'intégrer au mieux mon travail dans l'architecture existante de la société afin d'assurer une meilleure maintenabilité de la solution et sa pérennité au sein de la société. C'est d'ailleurs pour cela que l'exporter Prometheus a été utilisé, car un tel exporter est déjà présent et utilisé dans la société, mais je l'ai adapté à mon utilisation avec la base de données de Matomo.

Un autre intérêt de ce travail au sein d'Escaux a été le déploiement de la solution dans un véritable environnement, bien différent du test des solutions depuis des containers Docker préconfigurés. En effet, dans un environnement réel, il faut choisir les différentes solutions à installer et les configurer pour s'assurer d'un fonctionnement correct de l'ensemble du système. Il faut également mettre en place les différentes sécurités, telles que les backups des fichiers de configuration et des données, pour éviter de perdre toute information en cas de problèmes avec le serveur.

Sur un plan organisationnel, les réunions hebdomadaires mises en place ont assuré un suivi régulier et constructif de la progression de l'élaboration de la solution. Elles ont permis, au fur

et à mesure de lavancement du travail, de réorienter la recherche et de revoir les exigences du client par rapport au projet, ainsi que daffiner les tâches à effectuer et leurs importances. Toutefois, malgré ces réunions et une communication régulière au sujet du travail, quelques problèmes de compréhension ont pu avoir lieu entre la demande du client et la traduction de cette demande dans la réalisation du projet. En outre, la demande de la société Escaux na pas toujours été suffisamment claire en termes dobjectifs et de priorités. En effet, des décisions ont été prises tardivement pour certaines parties du projet, ralentissant fortement lavancée de ce dernier. Ensuite, devoir se renseigner sur un grand nombre de solutions qui métaient inconnues auparavant est une activité chronophage. Or, vu le temps imparti pour la réalisation du travail, il était indispensable de rationaliser le temps passé sur la documentation. Jai donc le sentiment davoir parfois dû choisir entre plusieurs solutions sans avoir pu comprendre ces solutions dans leur intégralité. Enfin, la contrainte majeure du projet était de travailler dans lenvironnement de la société. Cela a induit une restriction dans les choix possibles pour les différentes solutions car le projet final devait sintégrer au mieux dans larchitecture de la société. Cette intégration était essentielle pour assurer lexploitation de la solution à la suite de sa mise en place dans le cadre de mon TFE.

Finalement, le présent TFE ma permis dapprendre beaucoup de nouveaux éléments, tels que la création de package Debian, lanalyse et le test de différentes solutions pour essayer de tirer un maximum dinformation en un minimum de temps et surtout la mise en place dune architecture de solution au sein dun système de production contenant des contraintes propres à lentreprise.

Annexes

Annexe I

1 Implémentation de Google Analytics (GA)

Pour implémenter GA dans un site web, il faut tout d'abord se créer un compte ou utiliser un compte gmail pour se connecter sur <https://analytics.google.com/>. Ensuite, il faut créer un compte administrateur sur la plateforme suivi d'une propriété qui représente le site web ou l'application Android/IOS à ajouter. Ici, seul le site nous intéresse. Pour le site, il faut paramétrer un nom, l'URL ainsi que sa catégorie et le fuseau horaire.

Un ID de suivi est alors généré et sera utilisé dans le code de suivi. Enfin, il faut ajouter le code suivant dans la partie `<head>` de toutes les pages du site pour que ce dernier envoie des informations à GA.

```
<!-- Global site tag (gtag.js) - Google Analytics -->
<script async
      src="https://www.googleTag Manager.com/gtag/js?id=UA-135087158-1"
></script>
<script>
window.dataLayer = window.dataLayer || [];
function gtag(){dataLayer.push(arguments);}
gtag('js', new Date());
gtag('config', <IDdeSuivi>);
</script>
```

De plus, pour permettre de récupérer des informations spécifiques via GA, il faut ajouter des codes équivalents à ceux qui suivent et selon les cas d'utilisation :

Pour la vue d'une page, le code suivant est nécessaire dans la balise head.

```
gtag('config', 'GA_MEASUREMENT_ID', {
  'page_title' : 'homepage',
  'page_path' : '/home'
});
```

Pour la gestion des événements, le code doit être associé à chaque fois avec un Onclick ou se trouver dans des fonctions appelées par la page pour récupérer l'information choisie.

```
gtag('event', <action>, {
  'event_category' : <category>,
  'event_label' : <label>,
  'value' : <value>
});
```

2 Implémentation de Google Tag Manager

Pour implémenter GTM, il faut procéder plus ou moins de la même manière que pour GA. Par conséquent, il faut se connecter avec son compte GA sur <https://TagManager.google.com/>,

puis se créer un compte requérant un nom, la localisation, le nom du conteneur qui est le nom du site puis choisir le type de conteneur, dans notre cas Web.

Une fois le compte créé, le site nous propose directement les balises à ajouter. La première balise doit être ajoutée dans le <head> de la page tandis que l'autre doit se trouver juste après l'ouverture de la balise <body>. Codes à implémenter :

```
<!-- Google Tag Manager -->
<script>(function(w,d,s,l,i){w[l]=w[l]||[];
w[l].push({'gtm.start':new Date().getTime(),event:'gtm.js'});
var f=d.getElementsByTagName(s)[0],j=d.createElement(s),
dl=l!='dataLayer'?'&l='+l:''; j.async=true;
j.src='https://www.googleTag Manager.com/gtm.js?id='+i+dl;
f.parentNode.insertBefore(j,f);
})(window,document,'script','dataLayer','GTM_ID');
</script>
<!-- End Google Tag Manager -->

<!-- Google Tag Manager (noscript) -->
<noscript>
<iframe
src="https://www.googleTag Manager.com/ns.html?id=GTM-P5W8JT4"
height="0" width="0" style="display:none;visibility:hidden">
</iframe>
</noscript>
<!-- End Google Tag Manager (noscript) -->
```

Une fois cela implémenté, il est conseillé d'aller dans *Variable* et de créer une constante contenant l'ID de suivi de GA.

Ensuite, allez dans *balise* pour créer les différentes balises qui permettront l'envoi des données à GA.

Allez sur *nouveau*, *configuration* de la balise et sélectionnez *GA - Universal Analytics*.

Ensuite, créez une nouvelle variable pour les paramètres de GA. Pour l'ID de suivi, cliquez sur la brique et choisissez la constante précédemment créée qui contient votre ID de suivi. Pour configurer le type de suivi désiré, si vous choisissez *événements*, il faudra définir une catégorie, une action, un libellé et une valeur soit en dur soit provenant d'un élément récupéré via GTM. Il est à noter que toutes les variables intégrées ne sont pas activées par défaut et que, pour ce faire, il suffit de cliquer sur *Variable intégrée* et choisir la variable que l'on souhaite utiliser.

Après cela, ajoutez le déclencheur de votre choix en accord avec le type de suivi précédemment défini. Par exemple, choisissez *Liens uniquement* avec lequel, si vous le souhaitez, vous pouvez ne déclencher cet événement que lorsque des liens spécifiques sont cliqués.

Il reste la possibilité d'implémenter dans le code des événements personnalisés contenant au minimum un nom et dans lequel l'ajout d'une variable et de sa valeur est possible.

```
dataLayer.push({'event': 'event-custom'});
dataLayer.push({'event': 'event-custom', 'variableName': 'variableValue'});
```

Cela est différent de l'implémentation standard de GA car ce déclencheur sera capturé par le GTM et les métriques envoyées à GA seront formatées par ce dernier. Cela évite, lors de l'implémentation du code, de devoir ajouter toutes les données souhaitées à partir des données que le code nous renvoie lors de ces événements particuliers.

Une fois tout cela fait, il est vivement conseillé d'utiliser la fonction *prévisualiser* pour être sûr que les données et déclencheurs choisis soient bien ceux désirés.

Annexe II

Matomo documentation

Cette documentation explicite la manière d'intégrer les variables pour l'analytique dans une application Web ainsi que la configuration de ces dernières dans l'outil Matomo et présente aussi les choix syntaxiques par rapport à l'écriture des différents éléments de cette configuration.

Matomo est hébergé sur le serveur : `17448.sop.escaux.com`

Il est nécessaire d'avoir le VPN de production activé. Le login et mot de passe pour y accéder se trouvent dans le keypass.

N'hésitez pas à vous référer à la documentation officielle pour avoir plus d'informations sur les différentes possibilités qu'offre Matomo <https://matomo.org/docs/tag-manager>.

1 Informations préliminaires :

Pour utiliser le prévisualiseur, il faut aller dans l'onglet Tag Manager de l'interface de Matomo et cliquer sur l'activation du débogueur/prévisualiseur. Matomo fournit alors un paramètre à ajouter à l'URL de Connect Me pour l'activer dans la page. Une fois ce paramètre ajouté, il faut recharger la page en nettoyant le cache pour s'assurer que la modification soit bien prise en compte et que l'overlay du prévisualiseur apparaisse.

Le système de balise/tag du Tag Manager se divise en quatre valeurs en plus du nom de celui-ci. Les quatre valeurs sont la catégorie de l'évènement, l'action de l'évènement, le nom de l'évènement et une valeur de l'évènement. Les seules valeurs utilisées en vue du tracking et du monitoring sont la catégorie de l'évènement et l'action associée.

Après avoir créé de nouvelles balises, il faut publier une version via l'option publish présente sur le tableau de bord du Tag Manager pour permettre à ces créations/modifications de prendre effet.

2 Syntaxe

Pour assurer une cohérence dans les données stockées par Matomo, j'ai utilisé une convention similaire aux bonnes pratiques en programmation.

Dans les balises pour les catégories, les actions, les événements et pour certaines valeurs d'événements, j'ai utilisé des constantes afin d'être sûr que toutes ces valeurs soient bien identiques, permettant de diminuer la possibilité d'erreur humaine et surtout de garder une unicité pour un même événement. En effet, dans l'exporter, des noms qui diffèrent, même pour une majuscule ou une minuscule, sont traités comme deux informations distinctes. Devoir corriger toutes les erreurs possibles entraînerait un alourdissement du code de l'exporter alors que le problème ne se présente pas en utilisant la méthode choisie.

Au niveau de la syntaxe dans les variables, j'ai utilisé une écriture en majuscule pour toutes les constantes, du CamelCase avec la première lettre en minuscule pour toutes les variables et, pour les déclencheurs, le PascalCase ou UpperCamelCase a été utilisé avec la première lettre en majuscule.

Dans le code de Connect Me, les événements personnalisés sont écrits de la même manière que les déclencheurs. Concernant les valeurs d'événements, la syntaxe est équivalente à celle

des variables afin d'assurer une cohérence entre toutes ces valeurs. Cela permet également de rapidement différencier les différentes valeurs.

J'ai choisi d'employer le CamelCase et non d'autres solutions comme le snake_case utilisant le under score `_` ou encore d'autres solutions car je travaillais sur différents langages de programmation, différents systèmes de stockage et différentes solutions. Dès lors, utiliser des caractères spéciaux pourrait entraîner des erreurs d'interprétation, conduisant à des erreurs de cohérence ou des problèmes dans les solutions utilisées.

3 Choix des variables et implémentation dans le code

Afin d'ajouter des variables pour l'analytique, il faut être assez méthodique pour éviter de casser l'application ou d'avoir des incohérences dans le stockage.

Tout d'abord, il faut définir l'information à récupérer et l'endroit où la récupérer. Comme expliqué précédemment, le Tag Manager possède des variables ainsi que des déclencheurs pré-programmés qui peuvent récupérer certains types de données. Il faut donc vérifier si l'information peut éventuellement être récupérée par ce biais ou lors du déclenchement d'un des déclencheurs. Si cette méthode peut être utilisée, il est bon d'avoir recours au prévisualiseur afin de s'assurer que l'information ou le déclencheur corresponde bien à celui souhaité.

Si l'information ne peut être récupérée via le Tag Manager, il faut alors définir si elle est inhérente à l'utilisateur et non à une action effectuée par lui, comme le serait un fond d'écran pour l'application. Il n'est pas nécessaire d'utiliser ou de créer un événement personnalisé pour définir cela, il suffit d'utiliser un événement préprogrammé et de faire en sorte que l'application envoie cette information lors de l'événement. Pour ce faire, il faut importer la valeur dans le module de Matomo créé dans le code de l'application et y ajouter ladite valeur. Toutefois, si c'est une action que l'on cherche à récupérer, il faut alors créer et émettre un événement dans l'application si aucun n'existe au préalable. Ensuite, dans le module Matomo, il faut ajouter un listener par rapport à cet événement et déclencher un événement personnalisé pour Matomo. Si nécessaire, une valeur peut être ajoutée à l'événement, tout en respectant la syntaxe explicitée ci-avant afin de conserver la cohérence de ce qui a déjà été fait. En outre, il est nécessaire de contrôler le déclenchement de l'événement via le prévisualiseur. En effet, il se pourrait qu'un événement préexistant dans l'application se déclenche plus souvent que prévu.

4 Exemple d'implémentation :

Voir aussi : <https://matomo.org/docs/tag-manager/#3-advanced-settings>
Voici un exemple de l'implémentation de la variable et d'un événement personnalisé.
Le code :

```
import {
  USER_FRONT_END_CHANGE_STATUS_EVENT
} from 'ec-user-status'

import {
  USER_LOGIN_EVENT,
  USER_LOGOUT_EVENT
} from 'ec-user'

export default (userManager, analyticsManager) => {
  'ngInject'

  userManager
    .on(USER_LOGIN_EVENT, (user) => {
      user
```

```

    .on(USER_FRONT_END_CHANGE_STATUS_EVENT, value => {
      analyticsManager
        .pushEventVariableAnalytics('StatusChange', {'statusValue': value})
    })
  })
  .on(USER_LOGOUT_EVENT, () => {
    analyticsManager.pushEventAnalytics('Logout')
  })
}

```

Dans l'interface en ligne de Matomo à Matomo Tag Manager : Onglet : Variables : puis Create New Variable et choisir "Data-layer" (cela référence le fait que la variable vient de Connect Me)

Type

Data-Layer

Name

statusValue

Configure this variable

Data Layer Variable Name

statusValue

Pour une constante similaire à l'ajout d'une variable mais il faut choisir Constant :

Type

Constant

Name

CHANGETO

Configure this variable

Value


changeTo

Onglet : Triggers puis Create New Trigger et choisir Custom Event

Type
Custom Event

Name
StatusChange

Configure this trigger

Event Name
StatusChange 

Onglet : ú Tags ž puis Create New Tag et choisir ú Matomo Analytics ž


Name
User Status Change


Configure what this tag should do

Matomo Configuration
MatomoConfig ▼

[+ CREATE NEW MATOMO CONFIGURATION](#)

Tracking Type
Event ▼

Event Category
{{STATUS}} 

Event Action
{{CHANGETO}} {{statusValue}} 

Configure when the tag should do this

Execute this tag when any of these triggers are triggered.

StatusChange ▼  

▼

Don't execute this tag when any of these triggers have been triggered.

▼

Migration

Faire attention à toutes les documentations ci-après car les fichiers de configuration sont importants.

<https://websiteforstudents.com/setup-matomo-piwik-on-ubuntu-16-04-17-10-18-04-with-nginx-mariadb-and-php-7-2-support/>

https://matomo.org/faq/how-to-install/faq_138/#more-533

<https://matomo.org/docs/setup-auto-archiving/>

Normalement, tout est présent dans le backup du sop 17448. De plus, il faut faire attention, la migration de Matomo peut entraîner un changement dans IID du Tag Manager, ce qui résulte en un changement d'URL pour récupérer le script de suivi. Il faudra donc mettre à jour l'URL présent dans le module Matomo, Services/analytics.js, contenu dans la constante URL_ANALYTICS.

Annexe III

Informations à récupérer depuis Connect Me

- Appel :
 - Entrant/sortant
 - Transfert et type de transfert
 - Coupé et mis en attente
- Vidéo conférence :
 - Démarrage :
 - Utilisation en plein écran ;
 - Partage écran
 - Si non fonctionnel, la cause
- Conversation par message
 - Envoi de message
 - Envoie de fichier
 - Utilisation demojis depuis le menu
- Casque audio :
 - Savoir si les casques de marque Jabra fourni par la société aux clients sont utilisés
- Navigation :
 - Générale dans l'application
 - Dans les paramètres
 - Dans laide
- Personnalisation :
 - Fond écran
 - Thème
 - Utilisation de la mise en favori de contact
 - Changement de paramètres
- Utilisateur :
 - Connexion
 - Déconnexion
 - Identité
 - Statut
 - Retour ou nouvel utilisateur
- Méthode de recherche de contacts
 - Trois méthodes différentes proposées par l'application
- Informations générales d'utilisation
 - Numéro de version et de révision de l'application
 - Périphérique utilisé
 - Navigateur utilisé et sa version
 - Temps passé sur l'application
 - Nombre d'utilisateurs
 - Résolution écran

La visualisation Grafana affiche les tendances de chaque utilisation puisque les métriques proviennent d'une solution de Monitoring

- Tendances Résolution écran
- Tendances nouvel utilisateur et ancien
- Tendances Périphérique utilisé
- Tendances Navigateur et sa version
- Tendances OS du périphérique
- Tendances générales des événements
- Tendances générales des événements par catégorie
- Tendances générales des événements par visite
- Tendances générales des événements par navigateur
- Tendances générales des événements par visiteur
- Tendances du total d'actions par visite
- Tendances du total d'actions par visiteur
- Tendances du total d'actions par navigateur
- Tendances générales des événements pour un périphérique donné
- Tendances générales des événements pour une catégorie d'événements donnée

La visualisation Matomo donne la quantité précise d'information

- Visites par jour
- Navigateur et version
- Périphérique utilisé
- Résolution écran
- OS du périphérique
- Événements regroupés par catégories et affichage dans chaque catégorie des actions
- Durée moyenne des visites
- Nombre d'actions par visite