INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




Centralized third-party services to improve customer relationship management

JOSÉ LUIS DE CASTRO MARQUES DOS SANTOS
Outubro de 2020

POLITÉCNICO
DO PORTO

Instituto Superior de Engenharia do Porto

Centralized third-party services to improve customer relationship management

José Luis de Castro Marques dos Santos

Dissertation for obtaining the master degree on Informatics Engineering, on Graphic System and Multimedia field

Advisor: Luis Lino Ferreira

Co-advisor: João Granja

Porto October 2020

Abstract

Customer communication has a crucial role in assuring a trustful and reliable relationship with a brand. In many cases, customer communication is held by many different service providers in order to use different expertise.

Such services channel communication, segment targets, improve recommendations, among others. However, having different services on separate platforms may lead to some issues:

- the campaign creation and management must be done on different platforms;

- the learning curve will be different for each service;

- the more services we have, the more resources we need to manage and use them;

All the mentions points create different limitations for large companies. CRM software providers are reluctant to offer third party services integrations to their clients, considering that they would rather sell the full feature package. To address this issue in the industry, we propose working to find a way to integrate third-party services on one leading service.

Keywords: CRM, integration, API, Salesforce Marketing Cloud, middleware

Resumo

A comunicação com os clientes desempenha um papel crucial na criação de uma relação de confiança com uma marca. Em muitos casos, a comunicação com os clientes é assegurada por diferentes serviços, por forma a tirar partido da especificidade de cada um. Esses serviços podem ser usados para diversificar os canais de comunicação, segmentar clientes, recomendar produtos ou serviços personalizados, entre outros. No entanto, quando em comunicações de grande escala, o uso de diferentes serviços pode trazer alguns desafios, nomeadamente:

- a criação de campanhas tem de ser reproduzida em várias plataformas;

- a curva de aprendizagem será diferente para cada serviço;

- a quantidade de recursos alocados para gestão de uma campanha será maior quanto maior for o numero de serviços utilizados;

Os motivos referidos acima comportam limitações para empresas que precisam de comunicar em grande escala.

Ao mesmo tempo, as empresas que comercializam serviços de comunicação com o cliente e de CRM hesitam em disponibilizar integrações entre os vários serviços externos, promovendo assim a contratação de um pacote mais completo dos seus serviços.

Na perspectiva de resolver o problema descrito, propomos uma solução que visa permitir a integração de serviços externos, a partir de uma aplicação principal.

Keywords: CRM, integrações, API, Salesforce Marketing Cloud, middleware

Index

List of figures

List of Tables

Acronyms

List of Acronyms

API	Application Programm Interface
BI	Business Intelligence
CRM	Customer Relationship Management
CSV	Comma Separated Values
CEO	Chief Executive Officer
CMO	Chief Marketing Officer
CTO	Chief Technical Officer
FFE	Fuzzy Front End
GDPR	General Data Protection Regulation
HTTPS	HyperText Transfer Protocol Secure
IP	Internet Protocol
JS	Javascript
NCD	New Concept Development
NMV	Net Merchandise Value
NPD	New Product Development
QFD	Quality Function Deployment
REST	Representational State Transfer
SFTP	Secure File Transfer Protocol
SFMC	Salesforce Marketing Cloud
SMART	Specific, Measurable, Achievable, Realistic, and Timely
SSJS	Server Side Javascript
URL	Uniform Resource Locator

1 Introduction

In this chapter, we frame the problem tackled by this thesis and provide an overview of its context. The solution and the goals of this project are also succinctly described.

1.1 Context

For online retail companies, the relationship between the customer and a brand is an important asset. Establishing trustful and reliable relations with its customers often leads to more sales and promotes loyalty from its customers[1]. To have a good relationship with customers, a company must manage customer's expectations while delivering an enjoyable user experience[2].

Having excellent customer support and keeping a close and paced communication with customers is also vital to achieve the best possible customer experience and keep the customer engaged. Therefore, customer experience is a genuine concern of the customer relationship management (CRM) team of Jumia, where this project takes place.

1.1.1 Jumia

Jumia[3] is an online retail company that operates mainly in the African continent, having its most important markets in Nigeria, Egypt, Morocco, Kenya, and Ivory Coast. It is also present in countries like Ghana, Tunisia, Algeria, Senegal, Uganda, and South Africa. Jumia's goals are to grant access to new services for its customers and support African companies growth while creating a sustainable impact in Africa [3]. It was created in 2014 and has become a public company since 2019[4].

For online retail companies like Jumia, communication with users and customers plays a crucial role in its success[5]. Opposite to traditional markets, online markets entirely depend on communicating and displaying information to its users, ultimately converting them into consumers[6]. The way one brand interacts with its customers is critical to its success [2]. Therefore it is required to plan all the campaigns and strategies to achieve both company and

customer expectations. To do so, Jumia built a CRM team that uses marketing software solutions to organize data and communicate with its customers.

1.1.2 CRM

CRM stands for Customer Relationship Management. It is essential for companies in order to manage, maintain, and acquire new customers. Its role is to understand customer needs[7] and manage every interaction between the customer and the company.

In Jumia, the CRM team's core responsibility is to communicate company products and values to its customers and clients**.**

This CRM communications usually takes the form of a campaign - a message sent to an audience with a specific and measurable goal. It uses a specific target audience, where specific customer attributes are used to filter the communication(for example, target female gender with skirts, or french speaking user with french content). Currently, campaigns are sent using three different channels: SMS, email, and push notifications. Push notifications are the notifications that mobile and web apps can send to user's devices.

1.1.3 CRM Team

Operating in different countries, with different languages and cultures, demands Jumia flexibility and ease of deploying its communications. Therefore, the CRM team comprises a central team settled in Portugal and Dubai – to create and deploy the campaigns – and local teams to build and write content specific for each country.

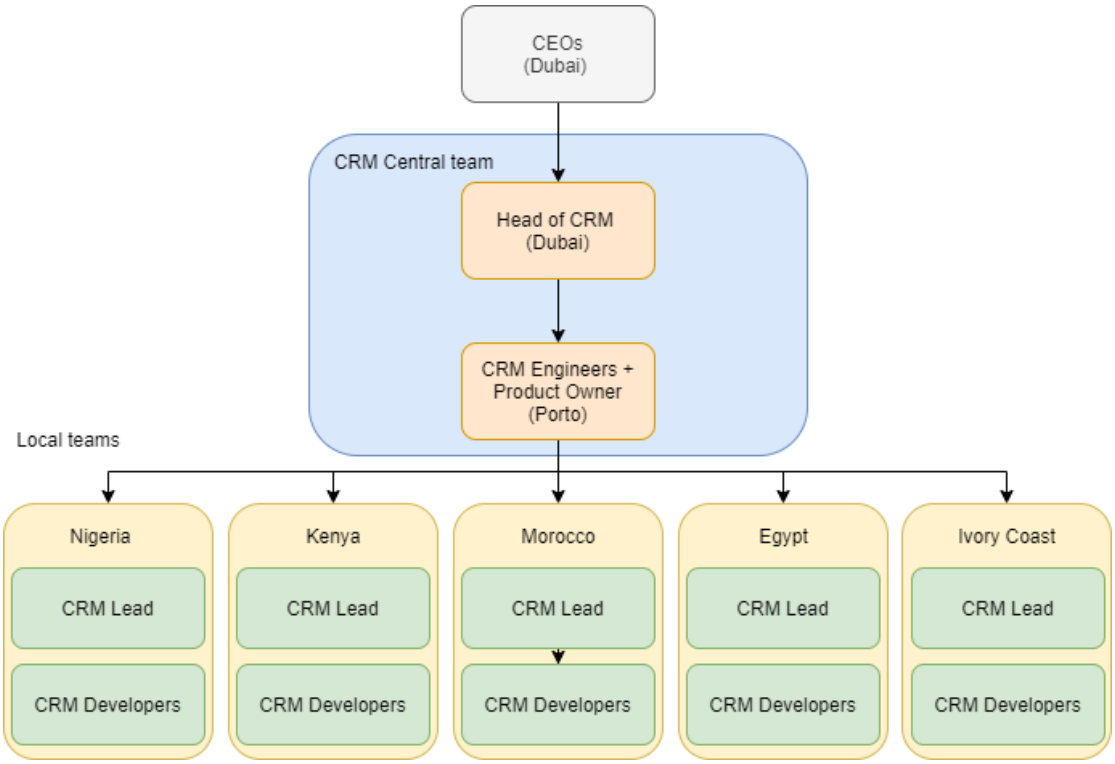


Figure 1 - CRM team organization

The technical team is composed of team managers and CRM engineers. Local teams are represented by a CRM lead and usually three persons for content writing and manage campaigns, as illustrated in figure 1. CRM leads reports to CRM central team manager and aligns ahead altogether the campaigns to launch on a given season. Often, people hierarchically above the Head of CRM request campaigns to the CRM team. They are referred to in this document as stakeholders.

The CRM team sends an average of 5 million emails and 10 million push notification every day, across all countries it operates.

The creation and scheduling of campaigns are set up and managed by CRM central team. Besides the campaign creation, the CRM team is responsible for managing a user lifecycle journey in the Jumia app and website. A user lifecycle journey encompasses all the interactions a user has with a brand, since the day it first visits the website or app – whether browser, mobile or physical store – until he converts to a customer(makes its first purchase), along with all the interactions in between, and further purchases.

To maintain a paced interaction with its customer demands from a company to reach them(customers) through many different channels - emails, push notifications, SMS, and commercials. To do so, a company usually makes use of marketing software to store and process user's data, as well as to communicate with its customers.

1.2 The Problem

Stakeholders defined a new Jumia's strategy, and the need to scale up business arise. Jumia stakeholders defined a plan to reach a bigger audience to increase customer orders and attract new ones. Therefore, a solution is needed to re-engage existing customers and reach new segments of users.

This challenge would increase CRM team work, as more campaigns need to be built, new segments need to be made available, and content created to fulfill such demands. Hence, it became urgent to optimize communication with customers, and if possible, to integrate all services on a single platform.

CRM team uses Salesforce Marketing Cloud (SFMC) to send emails and store and manage customer data.

1.2.1 Users and data gathering

Users and costumers could be differentiated by those who have made one purchase – costumers and those who have never made any purchase – users (figure 2). In case a user wants to receive a Jumia newsletter or any marketing communication, he is considered a subscriber.

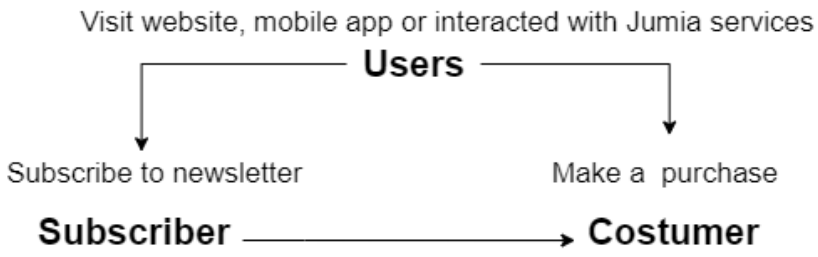


Figure 2 - Diagram for Jumia users naming

Users data is processed as follows:

- Firstly, data is gathered via each reporting service API (Google Analytics or Adjust);

- Then it is sent to and processed on Google big query;

- Finally, the data is sent to Jumia Dataware house;

After user data is processed and made available on a database, the local team can request data to data engineers to provide it via a comma-separated values(CSV) format file. The

collected data will allow local teams to launch campaigns on each channel/service requested and is usually made available on SFMC SFTP.

1.2.2 The current campaign creation process

The first step of a campaign creation is to define its target and its goal. Usually, stakeholders and the Head of CRM meet to request and discuss the campaign details and timeline to roll out. After this first moment, the technical deployment takes place involving both central and local teams.

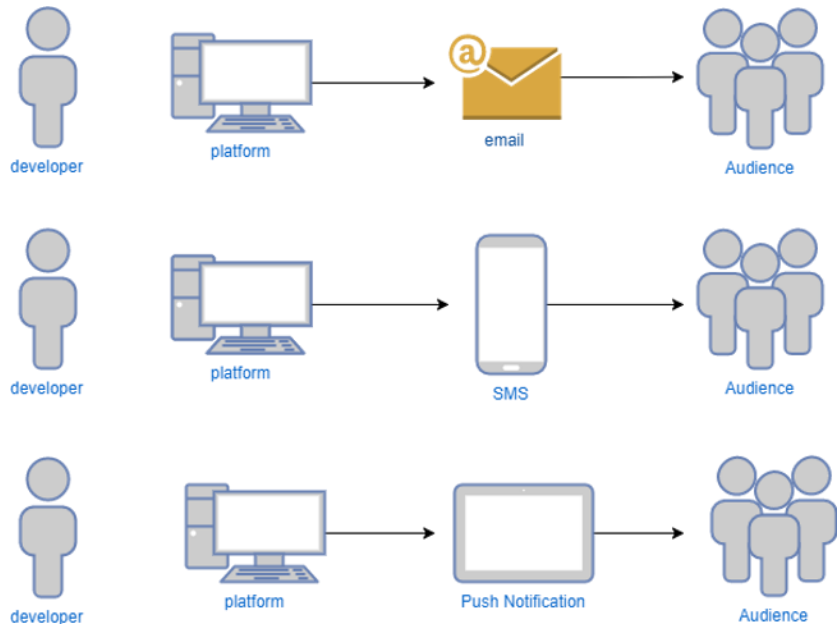


Figure 3 - Scheme for a campaign set up on the CRM team

The local team creates the content for a campaign on each channel platform and define the best timeline to send them (figure 3). The content is then validated by the Head of CRM to make sure it reflects both the campaign's goal and the company values as well. The CRM central team creates the campaign on each platform and automates all the processes to run on the desired date with specified cadence (daily, weekly, monthly, one-time send). This process requires that the central team create different campaigns per country and channel, but one person in every country needs to create content and update it for every service/channel in use.

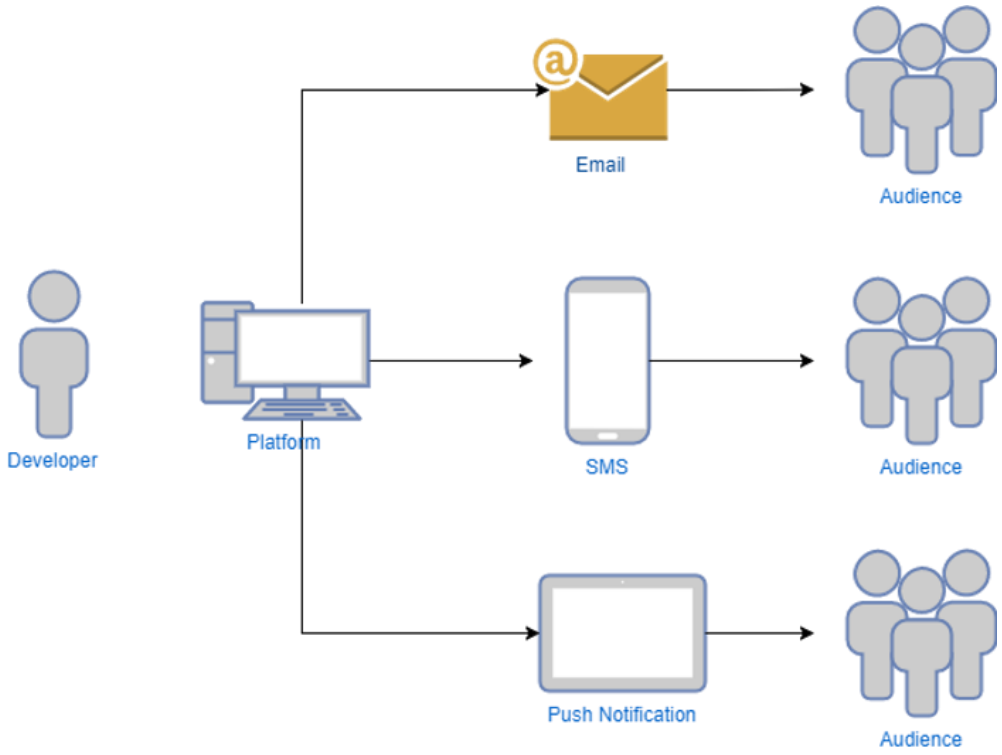


Figure 4 - Proposed solution process

The mentioned process is time-consuming and involves many people. It has proven to be error-prone and not much efficient.

For that reason, we propose a solution where we set up one campaign only and dispatch it to all countries and channels. To do so, we need to create the message in all languages in use for Jumia operating countries and integrate one system with different services to send campaigns.

A centralized service is then proposed in this master thesis project to solve the problem mentioned above, as illustrated in figure 4. The goal is to simplify this process and centralize the campaign set up in one service that would dispatch all communications to all countries and channels.

1.3 Goals

The goal of the proposed solution is to optimize the Jumia communication process with customers. By doing so, we expect to achieve the following objectives.

1.3.1 Reduce time to market for CRM campaigns

Currently, setting up a campaign requires users from every country to create content on every channel platform. The solution must allow for one single campaign to be set up and work across all countries and channels.

1.3.2 Resource optimization

Currently, a central team's CRM engineer is in charge of creating and automate every campaign. The content creation and update is done by every local team, across all countries where Jumia is operating, demanding around 14 people involved in creating one campaign through its major markets. Integrating all services in one single platform should allow automation to launch everyday campaigns autonomously to all countries, reducing the number of people involved, freeing them for other tasks.

1.3.3 Autonomous and future proof

The system proposed should integrate existing marketing services with third-party services while allowing for automating the campaign launch in different channels. Though, it must be independent enough to permit connections from other services, other then-current CRM stacks - SFMC, in this case.

1.3.4 SMART objectives

For an easy understanding, we will adopt the SMART objective framework.

SMART objectives seek to simplify and support measuring the results of projects. SMART stands for Specific, Measurable, Achievable, Realistic, and Timely [8]. According to its definition [8], it should:

"(…)outline in a clear statement precisely what is required,

(…) include a measure to enable to monitor progress and to know when the objective has been achieved,

(…), - focused on outcomes rather than the means of achieving them,

(…) agree the date by which the outcome must be achieved(…");

last but not least, all objectives should be achievable.

Considering this, we have framed our goals to deliver a service that integrates different communication services(push notification and SMS) to one central service (SFMC), with all the features to be detailed ahead in this document. Also, we will consider the timeframe of a semester to deliver this service.

1.4 Opportunity and value

The communication process was both a need and an opportunity as the company needed to reach more targets to grow. At the same time, this permits the CRM team to improve its work while helping Jumia grow. Since this request came ahead Jumia yearly roadmap planning, we have had time to survey all options and present alternatives.

The final solution should reduce the amount of time spent on each campaign's launch and setup while scaling the communications. The person-hour per campaign should significantly be reduced by the integration offered.

In the end, we expect to deploy an integration between SFMC and third party services, namely Infobib and Accengage. Below we detail the problem and survey for all available options to achieve the goals mentioned above.

1.5 Thesis structure

After an initial overview of the problem in the first chapter, we will describe it in a more detailed perspective on the second chapter. The third chapter describes the value analysis of the solution. In the fourth chapter, we discuss the available options. Next, solution design and evaluation are detailed, in the fifth and sixth chapters, respectively. The seventh and last chapter of this document presents the conclusion of the solution and the project.

2 Context and state of the art

2.1 Problem and context details

The following section seeks to explain the problem from a more detailed perspective. The concepts for better describe a problem bases on the book Engineering Reasoning [9].

The purpose of this solution is to integrate all external marketing services within a single platform. Doing so is expected to optimize the communication process, reducing the time to set and maintain campaigns.

The objectives of this project also present an opportunity to improve CRM efficiency. Integrating services within one core platform will enable to automate a more significant part of CRM communications. Furthermore, it will spare time and reduce error-prone related to manual writing content.

The scope of this project, its challenges and benefits were primarily defined by Jumia stakeholders – CEO, CMO – and then analyzed and discussed with the Head of CRM. Together they have established the goals and functional requirements to accomplish.

Considering this project is developed within Jumia, stakeholders and head of CRM could be seen as customers of the product as local teams. They will use the integration in their daily work. The solution must allow easy integration to third party services on marketing tools (existing or not), like Accengage and Infobip. It also must be compatible with the existing CRM setup, ideally keeping existing services and campaigns. In case an inhouse solution is adopted, the system should be independent of any platform or service, not to be tied to any service in the future.

From the stakeholder's standpoint, this solution's goal will be achieved if we could reduce the timing to market for the CRM campaign due to third-party tools integration. This project will be developt using SCRUM and agile methodology[10]; hence we will split the project into tasks and then detail each of them into subtasks and add it to the project roadmap. It is

essential to detail the tasks, as it impacts the correct development/prioritization and delivery time for deployments.

It is expected that fully detailed documentation is written and made available to all parts along with development. SFMC does not provide very extensive documentation related to the integration of other tools with SFMC. For that reason, we will need to use available documentation and share our findings internally in Jumia.

2.1.1 Ecommerce tracking and communication

Every action taken by Jumia users on both the website or the app is measured. When a user logs in, makes an order, searches for a product, or clicks on a banner, Jumia receives data that bind that user to an action. Similarly, all user data, such as gender, age, or amount of money spent on orders, is also stored in each user's record.

Such a large amount of data are continuously collected and processed by Business Intelligence (BI) automation processes. According to each campaign's goal, the processed data is later made available to the CRM team to define the best users to target. The data is made available to the CRM team via CSV files, shared on the SFMC SFTP. This data allows the CRM team to fine-tune the products and optimize results.

2.1.2 CRM team and work processes

In Jumia, communications with customers are divided into two types: transactional - order notifications, payment receipts, among others - and commercial - newsletters, marketing campaigns, among others.

The CRM team is responsible for commercial communications, being the transactional communications automated by operational teams.

The chief marketing officer (CMO) defines the company strategy for a specific period – usually one year. Then, the Head of CRM defines with all teams the result each team and country should reach and how communications will perform successfully.

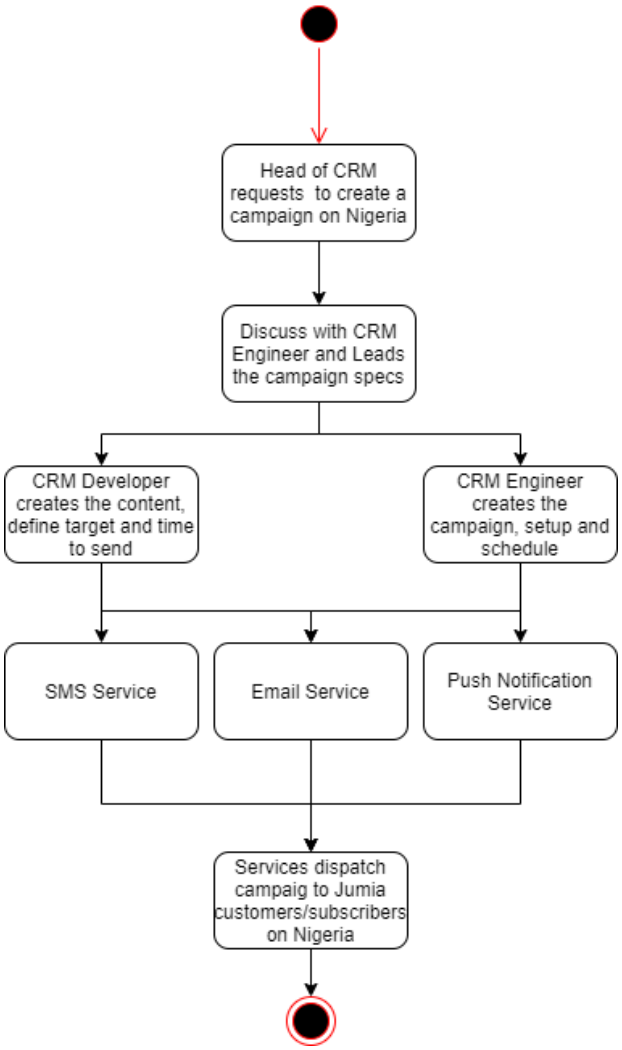


Figure 5 - UML for the current process for campaign creation

After this high-level definition by the management team, a set of campaigns is discussed with CRM central team. Ended this process, the content is created in every country by local teams and aligned on dates to dispatch the campaigns. Both central and local teams are in constant contact during this process. Figure 5 illustrates the process since the request of the Head of CRM.

Overall, the content creation process is as follows:

- Head of CRM team define, with CRM lead from countries, and the CRM engineers, the campaign to be created;

- CRM engineer create the campaign on every service needed (SFMC, Accengage, Infobip), according to the channel to use;

- local teams write the messages and create the images to be spread on every channel (SMS, Email, push notification);

- after approval from the CRM lead of the country, the content is updated on the campaign created previously by CRM engineers on the specific platform for the campaign;

- local teams define the audience and the time to send the message, usually considering analytics tools like Google Analytics or Adjust, in order to measure the best time to send the campaigns;

- the campaign is sent.

As this process showcase, one message is sent to a broad audience, leaving no space for customization on the content[11]. Such customization is crucial for customer engagement[12], as the message will be more interesting for them[13].

2.1.3 CRM tools for communication

CRM team has currentty three tools for marketing communication: Infobip for SMS, Accengage for sending push notifications, and Salesforce Marketing Cloud for sending emails.

Infobip is a service that allows sending SMS to users. It stores the data to send to users in CSV files but does not provide any data storage features. It also has a feature to track the opened messages and clicked URLs(for messages with links). It has features reports and A/B tests for campaigns.

Accengage is the service hired to dispatch push notifications. This service uses the deviceid and UUID from the app and desktop users and displays push notifications, rich push notifications(images), alerts, and web notifications. This service needs campaigns to be created for both operative systems(iOS and Android). To use this feature, it is required that the user deliberately choose to receive notifications. This also allows us to store user data – such as las product view, last product search, etc – and display that same data on a message.

Besides messaging, Accengage also provides A/B test feature and the ability to track each campaign performance with the platform: open rate, bounce rate and click rate.

Salesforce Marketing Cloud is a cloud-based service from Salesforce that provides marketing tools, namely the communication to customers and CRM data management. It aggregates data from customers and subscribers, being that subscribers represents every user that requested to receive commercial communications from Jumia, and allows one to send communications to its clients, design user journeys, and manage automation. As referred to in the previous tools, SFMC also provides a tool to measure the campaign engagement, such as email open rate, bounced rate and click rate.

All the above-mentioned tools were in use in Jumia for some time; hence, most developers and senior members are experienced and comfortable with its features.

Campaign creation is very time consuming and low efficiency since the same message could be written in 3 different services.

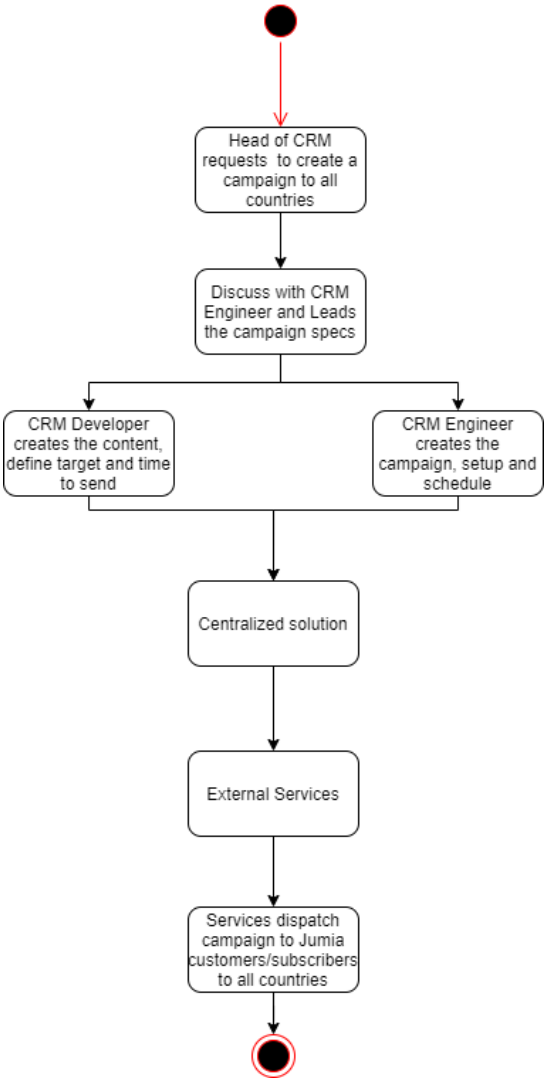


Figure 6 - UML for the new campaign creation process

This project will mostly impact central and local teams' setup, as it would materialize into a software integration. We aim to integrate all channels – namely SMS, Push notifications, and emails - into one platform, as illustrated in figure 6. In the end, we expect to deliver a solution where one same campaign can be set up in one service, and work for all countries and channels.

Below we will look into more detail on the possible solutions to proceed with the integration.

2.1.4 Communication channels

The communication channels are the means to reach users from a CRM standpoint. As stated before, CRM deploys communications in three different channels: SMS, emails, and push notifications.

SMS is the least used channel. Despite a high open-rate(number of people who open the message versus the amount of SMS sent) due to the communication costs and deliverability limitations on some countries and limitations on the content itself(160 characters), this channel is rarely used.

Emails are a quite used channel to reach users. Since the user login requires an email, we use it to measure and track its behavior on the website and app. Also, SFMC provides reporting on how users interact with emails(Open rate, click rate, and bounces). Jumia dispatch around 5 million emails per day.

Last but not least, push notifications. Despite its recent usage, push notifications are massively used in Jumia. CRM team dispatch around 13 million push notification per day. This channel takes advantage of Jumia mobile and app users and has a higher conversion and open rate, comparing with email and sms. Therefore this is the channel where CRM campaigns must focus.

2.2 Available solutions

This survey intends to gather solutions that allow for creating and dispatching campaigns using at least three channels: SMS, push notifications, and email. Their price must be competitive, and it should allow for integrations with external services. Not to exclude any possible solution, we were looking for existing solutions on the market and the possibility of building our solution using existing systems. Worth to mention that this project was started on mid 2018, henceforward, the features and available solutions were considered for that specific time frame.

2.2.1 Marketing Solutions

2.2.1.1 Adobe Marketing Cloud

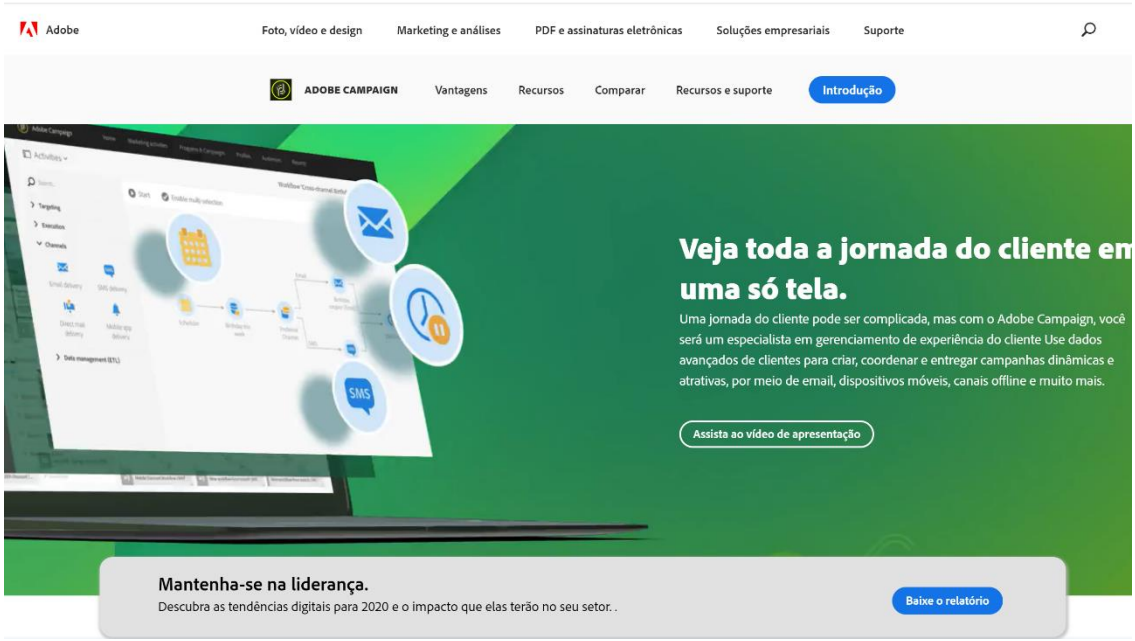


Figure 7 - Adobe marketing cloud

Adobe marketing cloud is a solution for marketing, similar to SFMC, as it offers a bundle of services within the same platform, allowing to add/remove services along the way. Besides providing services for targeting users, tracking the campaign performance, and multichannel communication, its most vital point is integrating with content creation tools, such as Adobe creative cloud.

2.2.1.2 Airship

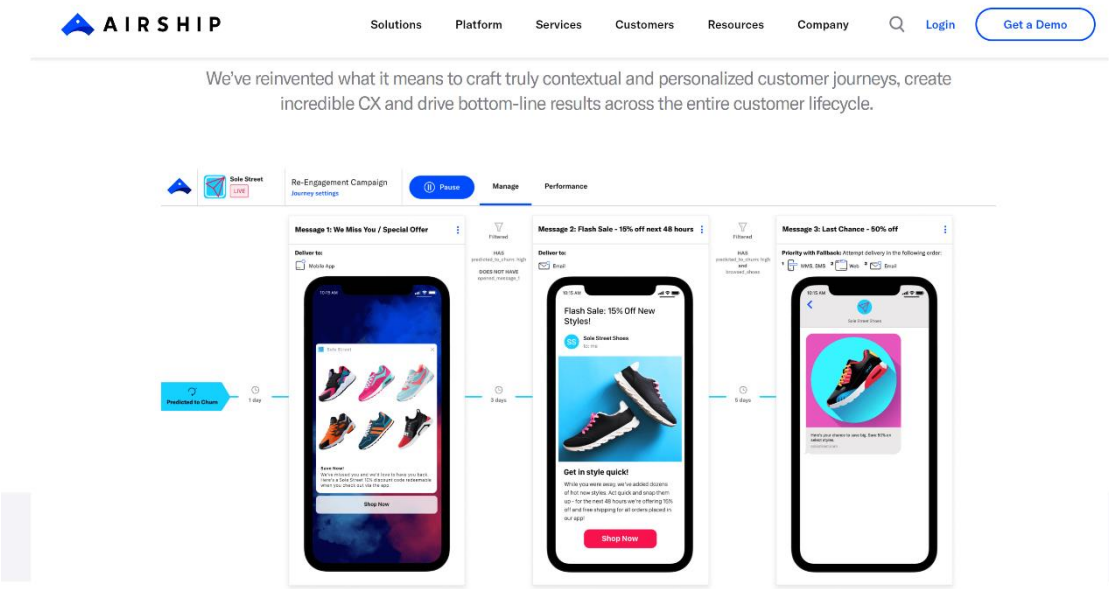


Figure 8 - Airship marketing

Urban Airship is a mobile-oriented platform that allows multichannel communication and orchestration for campaigns. It is very oriented to mobile communications, as it is proven by acquiring Accengage, a push notification platform, in 2019.

2.2.1.3 Braze

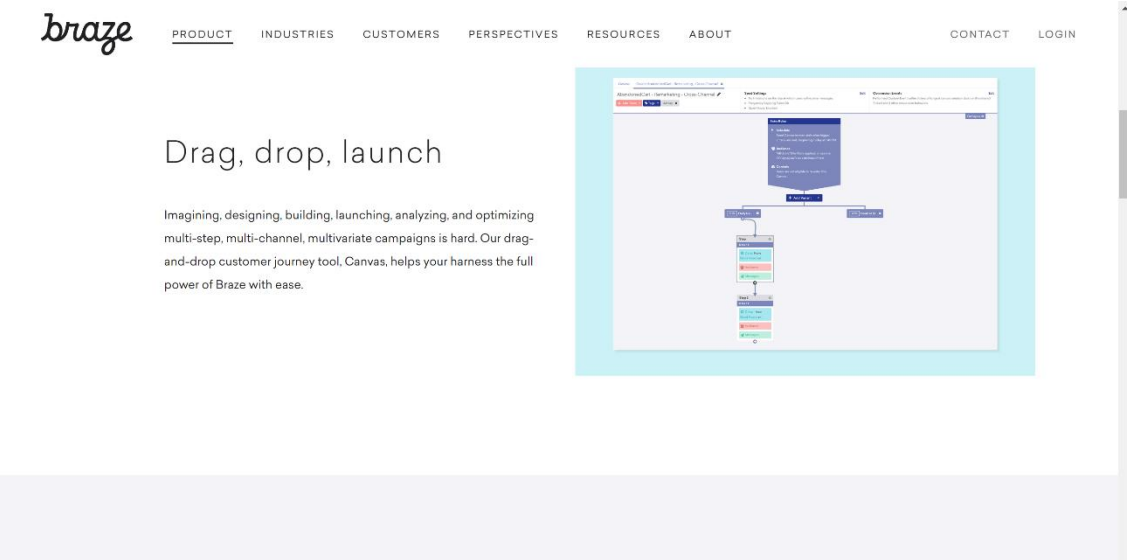


Figure 9 - Braze

Braze, formerly Appboy, is a mobile-first marketing tool. It provides both multichannel and campaign orchestration within its platform. It provides detailed and extended marketing tools – target, tracking, and automation –especially for mobile devices.

2.2.1.4 Oracle Responsys

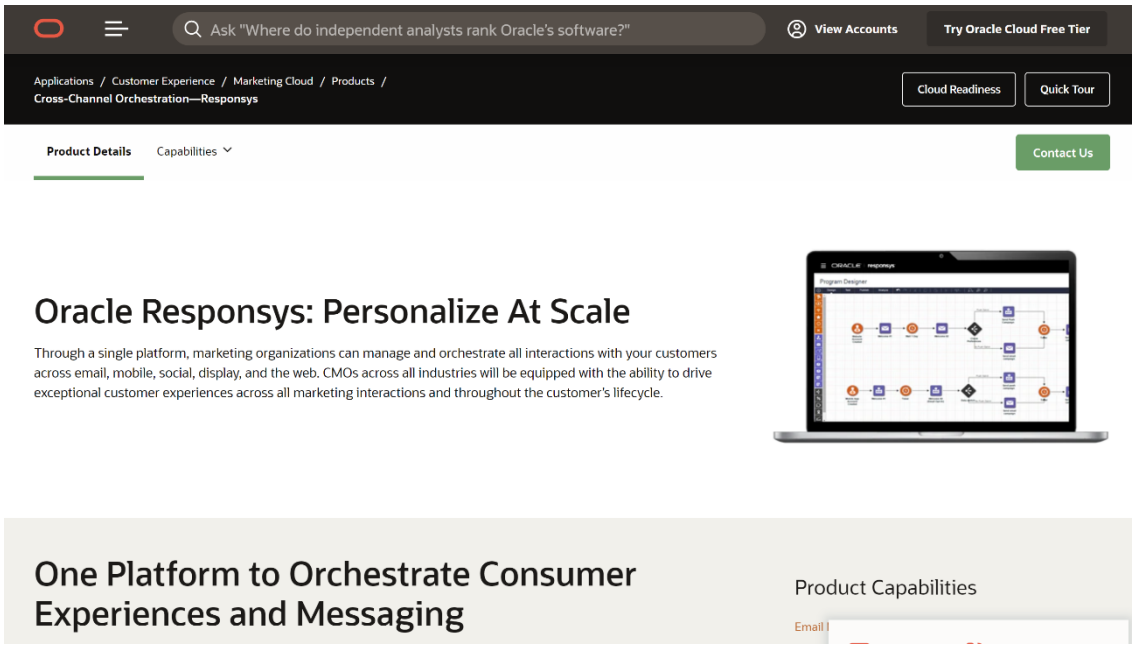


Figure 10 - Oracle Responsys

The marketing bundle from Oracle provides a complete set of features. The downside is that the user experience on the platform is not as intuitive as the other services.

2.2.1.5 Salesforce Marketing Cloud

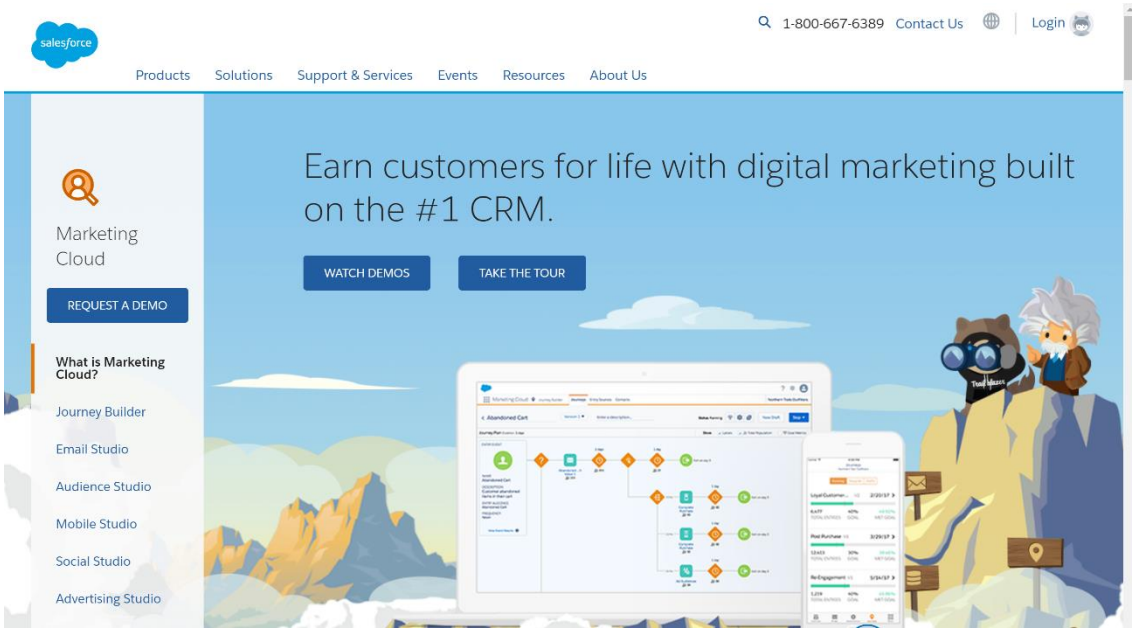


Figure 11 - Salesforce Marketing Cloud

Salesforce Marketing Cloud - a rebrand of the former ExactTarget, meanwhile acquired by Salesforce - is a marketing communication tool. It provides a wide range of tools, namely for sending emails, SMS, and push, store users CRM data, and provide reports on the performance of campaigns and communications.

According to the Salesforce description, "Marketing Cloud is a unified engagement platform for developing personal relationships with consumers." In Jumia, SFMC is used as the core platform for behavioral communications with customers. All marketing campaigns or communications are a trigger and send through it.

Because an integration with Salesforce Marketing Cloud is a possible approach, it is essential to have a ground knowledge of Salesforce Marketing Cloud - hereafter referred to as SFMC - features, and goals.

To have an overview of the services we have available on SFMC, we will list the most important:

- **Email studio** - where all the emails are sent. From here, we can define the Data Extension - like database tables, set a campaign, create the content, and dispatch emails;

- **Analytics Builder** - where we build the campaign reports.

- **Automation Studio** – a module to setup automation-related with processing data – i.e., segment data from users, append data to list, or filter lists with specific criteria, import data from users, execute scripts, send emails, among other tasks. As the name implies, automation

is a set of tasks that we can design to run on a specific schedule or trigger. Within Automation Studio, we have different types of activities that represent each task on the automation.

- **Journey Builder** – Allows for designing user lifecycle journeys. A user lifecycle journey is a set of actions between users and a company with a defined goal. For instance, one journey could be designed for the users birthday[fig2]:

    - from user's data stored on a data extension with their birth date, we start a journey and send a birthday email.

    - after three days, check if the user has made any purchase on the website, using data stored on the data extension;

    - if he did, then exits the journey, achieving the journey's goal;

    - if the user did not purchase, then send an email reminder, wait two days, and then leave the journey;

Journey Builder allows for automating these interactions and provided insights about customer engagement thoroughly. It also allows for setting goals – like make the customer visit a specific page of convert into a purchase.

- **Content Builder** - where we design the content to dispatch.

An overview of Automation Studio and Journey Builder's most used activities is useful to frame the possible options to integrate third party services.

2.2.1.5.1 Automation Studio Activities

Import Activity – Import data from a specific file, placed on SFMC SFTP, to a specified data extension.

Send email – dispatch an email to an audience/segment. The content for the email must be defined before sending, along with the target audience.

SQL Activity – runs a specific query on a specific list/data extension to append, update, or remove data.

Wait – As the name suggests, it is an event that holds the automation for a given time.

Script Activity – Automation Studio allows for running code on automation. The code run on a script activity uses SFMC Server Side Javascript and does not allow any external library to be imported or used.

Data Extract – To extract data from a data extension or list to an XML or CSV file.

File Transfer – to transfer data from an XML or CSV file to a location on SFMC SFTP.

Worth to mention that Script Activities and SQL activities have returned a time out an exception when running for more than 30 minutes.

2.2.1.5.2 Journey Builder Activities

We will aggregate the activities by type to make this point more comprehensive.

Entry events – The events that will trigger a specific row/user to start the journey. It could start from a data extension, an API event call, or other entry event.,

Messages – Define a message to deliver to users defined on the entry source. The content must be assigned ahead of this process.

Flow Control – provides logical operators (if, then) that will trigger actions based on defined conditions.

2.2.2 SFMC Integrations

SFMC is a platform develop and own by Salesforce, a company that provides a vast range of services. Therefore, it is natural that most of the built-in integrations on SFMC apply to Salesforce services. However, despite the recent acquisition of Mulesoft in 2018[14], SFMC still provides some options.

After gathering all requirements, we have reviewed documentation and stated the available options for integrating external services with SFMC:

- Use of third party APIs;

- Develop a middleware to be used by SFMC script activity;

- Create a custom activity to run on Journey Builder;

Below we will detail all the available options.

2.2.2.1 Custom Application on Journey Builder

Custom activities are components of code that run within the journey builder. It allows a developer to build activity to trigger a specific piece of code or interaction within the journey builder.[15]

To do so, we need to host a service accessible to the journey builder, on a public endpoint, to process any data and return the action or data required. This solution allows API calls to any service, every time we run a journey for every user.

2.2.2.2 Middleware

Middleware is a common solution when integrating different software systems. It can be described as software that mediates between two pieces of software[16]. It allows for integrating different systems that, for some reason, cannot be connected otherwise - lack of API, legacy systems, authentication blockers, or others.[17][18]

On SFMC, a middleware could be used from any component that allows for running code: either a script activity on Automation Studio or a custom activity on Journey Builder.

2.2.2.3 MuleSOFT integration API

Mulesoft was a company that provided integration services. Salesforce acquired it in 2018, and it has now its services integrated with most of Salesforce services. It integrates services using APIs and provides a thorough integration with SFMC to manage objects and calling services provided by the Salesforce Exact Target API.[19]

2.2.2.4 Third-party Service API

API stands for an Application programming interface. It is "(…) a way to share information, specifying what information can be requested, how requests are made, and what is returned (…)" [20]. An API usually demands a call to be made to do a service that will return a message, and proceed or not with action. An API contract defines the format of a payload, and it is the object/data structure the system expects to receive, therefore allowing the service to work. Both call and return data format must be specified on API documentation.

Often cloud-based software provides an API so that it can be connected to its customers existing platforms.Given that many people in Jumia were already using SFMC for years, and the change to another tool will have a learning curve.Also, due to the signed contract with Salesforce Marketing Cloud and eventual subscription plan for the upcoming years, marketing stakeholders are reluctant about looking for other options other than the Salesforce Marketing Cloud. Therefore, other players were excluded, remaining the options that integrate within SFMC.

3 Value analysis

Value analysis is a methodology whose goal is to bring the optimal value of a product, good, or service. Such methodology comprises a set of tools and techniques that deeply understand the company and customers' needs and requirements.

It is an essential step towards delivering a successful product. It is meant to survey all the customer requirements and deliver a product that brings value to its customers.

According to David Hugues, "In such a fast-paced environment, product development must be transformed into a continuous, iterative, learning process focused on customer value" [21].

Below is described the process and decisions taken on the analysis of this project.

3.1.1 The new concept development model

Peter Koen proposed a model for the innovation process [22], as per Figure 12. Three phases compound it:

- Fuzzy front end (FFE);

- New product development(NPD) ;

- Commercialization;

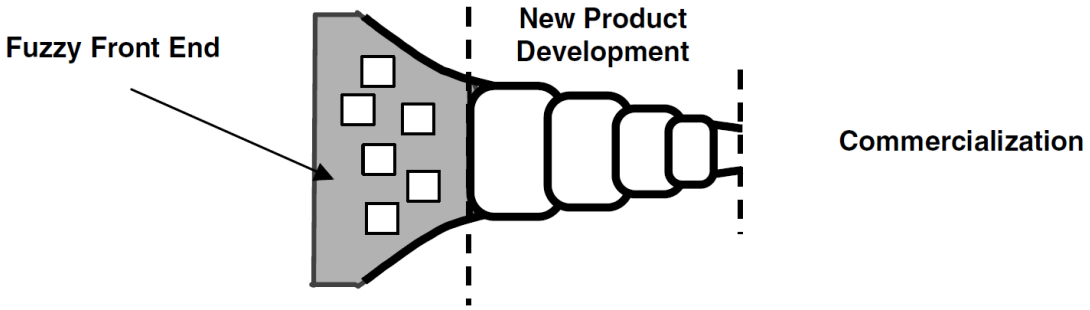


Figure 12 - Fuzzy Front end, NPD, and commercialization scheme [22]

Fuzzy Front End(FFE) is the first stage of this innovation process [5]. It comprises the definition of the problem and surveys new ideas, features, and opportunities for the project. The decision-making process is not structured and often chaotic at this phase, focused only on the client's needs.

The next stage, New product development, is a structured process, and outcomes are clearly defined. New Product development encompasses the following steps:

- Influence factors:

- Engine: It is the part responsible for supporting all the other parts of the process. It represents the company leadership, culture, and business strategy of the company.

- Five elements are the FFE activities - idea generation and enrichment, idea selection, concept definition, opportunity identification, and opportunity analysis.

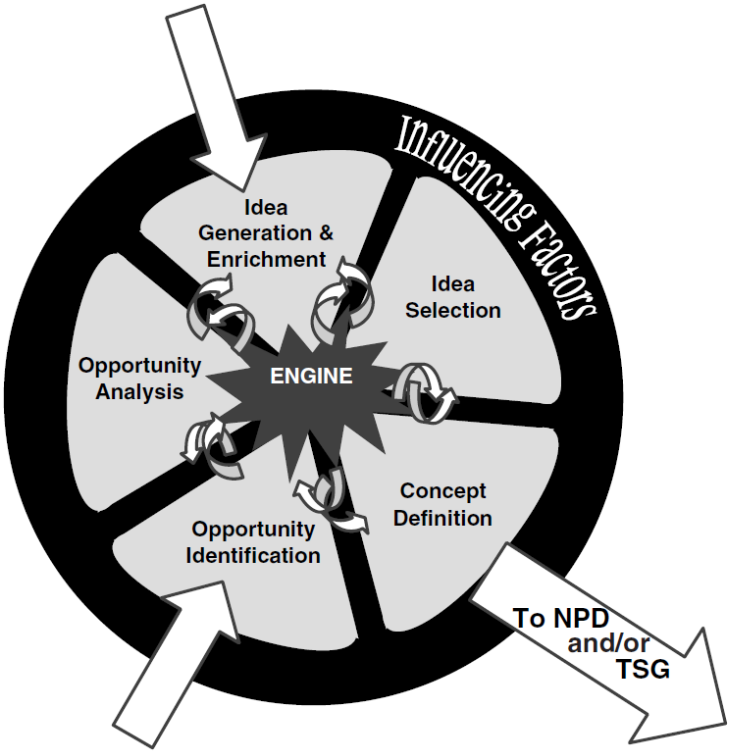


Figure 13 - The new concept development (NCD) [22]

The two inner pointing arrows indicate that projects begin at either opportunity identification or idea generation and enrichment. The outer pointing arrow represents this process's exiting, entering the new product development (NPD).

3.1.1.1 Opportunity identification

"In this element, the organization identifies opportunities that it might want to pursue(…)".[22]

To validate the opportunity, we use the following techniques:

- technology trend analysis ;

- customer trend analysis ;

- market research;

In the end, we have concluded that improving processes are an increasing concern among companies, usually linked to a need to scale.

We have met with CRM team stakeholders for customer assessment. We have noticed that the campaign creation processes were very time and resource consuming. Hence they were

looking for some way to optimize while scaling the communication with users and customers. After defining the roadmap, a specific bandwidth was assessed to this task and assigned to the CRM team.

3.1.1.2 Opportunity analysis

"(…). In this element, an opportunity is assessed to confirm that it is worth pursuing. Additional information is needed for translating opportunity identification into specific business and technology opportunities. This involves making early and often uncertain technology and market assessments. (…)Opportunity analysis may be part of a formal process or may occur iteratively".[22]

In this step, the following tools were used: strategic framing and market assessment.

Strategic framing defines how an opportunity fits within its market and its technology strengths, gaps, and threats. Market segment assessment describes the market segment, showing why it represents a great opportunity.

This integration would increase value to the company and could be useful in other contexts and systems in the future.

3.1.1.3 Idea Generation & Enrichment

"The element of idea generation and enrichment concerns the birth, development, and maturation of a concrete idea. (…)[6]

An idea may go through many iterations and changes, as it is examined, studied, discussed, and developed in conjunction with other elements of the N.C.D. model. (…)" [22]

To make sure all options and approaches were taken into account, we met with different teams and product owners from different projects and backgrounds to validate the idea, enrich it, avoid blind spots on deployment, and avoid other design flaws.

3.1.1.4 Idea Selection

"(…)Idea selection involves an iterative series of activities that are likely to include multiple phases through opportunity identification, opportunity analysis, and idea generation, and enrichment, often with new insights from the influencing factors and new directives from the engine. (…)" [22]

The techniques used were portfolio methodologies based on multiple factors, namely technical success probability and commercial success probability, and strategic fit.

3.1.1.5 Concept Definition

"(…)Concept definition is the final element of the new concept development model. This element provides the only exit to the NPD or technology stage-gate(T.S.G.). (…)" [22]

In the end, a concept was defined, and CRM had the sponsorship by a receiving-group champion, and a project plan, including resources and timing, was outlined. Given that there were business concerns about signed services, it was decided to integrate all third party services within SFMC.

3.1.2 Value, Value for the customer, and perceived value

Value is a concept that can have different definitions. It could be defined as the reason between what a person gets - whether there are tangible or intangible benefits - and what a person gives, meaning, the costs for that acquisition, where the costs could be monetary, timely, behavioral, or other(Walters & Lancaster,1999).

Alternately, value creation could be seen as "(…)a trade-off between benefits and sacrifices perceived by customers during a supplier's offering" (Geoff Lancaster 2000).

Being that the solution was built within the company, it is not easy to frame value. Optimization would be the value to deliver in this solution, as it encompasses many advantages and features. Ultimately, optimization for the customer communication process would undoubtedly be the value for this project stakeholders.

According to Woodall [23], value for the customer is defined as "(...)any demand-side, personal perception of advantage arising out of a customer's association with an organization's offering, and can occur as a reduction in sacrifice; the presence of benefit (perceived as either attributes or outcomes); the results of any weighted combination of sacrifice and benefit (determined and expressed either rationally or intuitively); or an aggregation, over time, of any or all of these". From this perspective, sacrifice is the resources allocated to this project and the roadmap's development time. Those resources weigh in with benefits - the ability to interact with other platforms in the future, and assure an easy way to integrate third-party services (table 1). It does not commit to any service signing by not depending on a specific service or platform, being that another advantage.

	Product	Relationship
Gains	Integrate with other services Automate campaigns on different channels	Flexibility to be used with other services in the future, Reliability
Pains	Resources allocated to the project	Time to develop

Table 1 – Sacrifices and benefits from the proposed solution

Zeithaml (1988, p14) proposed that "perceived value is the consumer's overall assessment of the utility of a product based on perceptions of what is received and what is given."

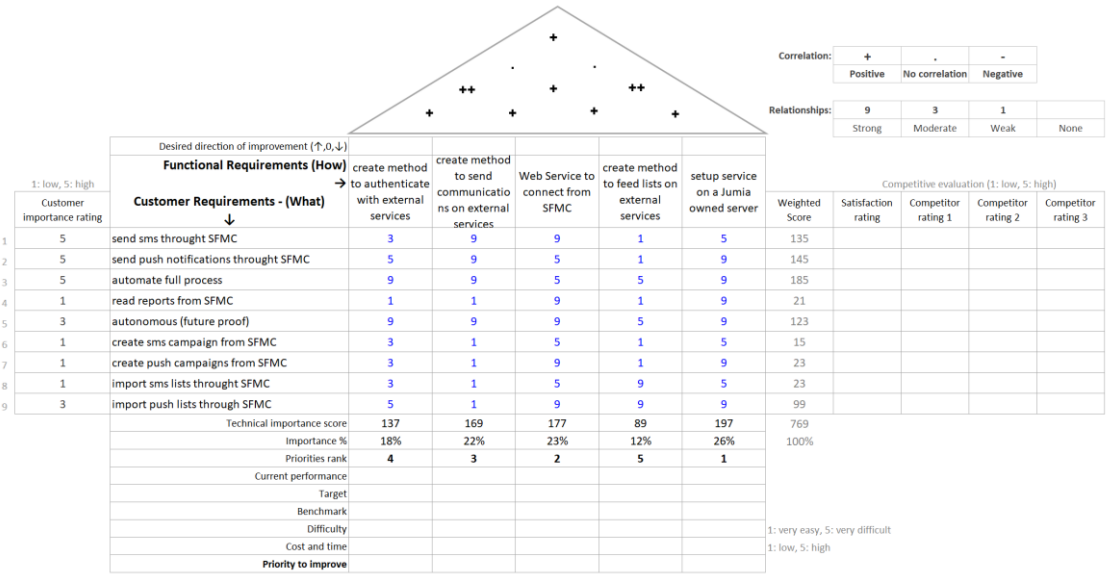
The creation of a solution within the CRM team also brings ownership to the team. Ownership is seen as an advantage for managers, as it could widen the team's scope from the leadership point of view. At the same time, the solution brings autonomy so that we have full control of the service.

As previously referred, perceived value is here defined in a company's context, so its framing is somehow subjective.

3.1.3 Quality Function Deployment

Quality Function Deployment(QFD) is a technique designed to ensure that customer needs are focused on throughout the new product project[24]. This technique allows for matching product requirements with technical deployments. The resulting of QFD is to prioritize technical tasks in regards to customers/product priorities.

From the customer(stakeholder) point of view, this document provides a clear view on the feature to deliver from one project. On the other hand, this technique eases the roll out of the project, as it details every technical task needed to be deployed in order to achieve each feature requested. From this macro tasks, developers can use the SCRUM methodology and detail and plan all the development.



								Correlation:	+	.	-	
									Positive	No correlation	Negative	
								Relationships:	9	3	1	
									Strong	Moderate	Weak	None
		Desired direction of improvement (↑,0,↓)										
	1: low, 5: high	**Functional Requirements (How)** →	create method to authenticate with external services	create method to send communications on external services	Web Service to connect from SFMC	create method to feed lists on external services	setup service on a Jumia owned server		Competitive evaluation (1: low, 5: high)			
	Customer importance rating	**Customer Requirements - (What)** ↓						Weighted Score	Satisfaction rating	Competitor rating 1	Competitor rating 2	Competitor rating 3
1	5	send sms throught SFMC	3	9	9	1	5	135				
2	5	send push notifications throught SFMC	5	9	5	1	9	145				
3	5	automate full process	9	9	5	5	9	185				
4	1	read reports from SFMC	1	1	9	1	9	21				
5	3	autonomous (future proof)	9	9	9	5	9	123				
6	1	create sms campaign from SFMC	3	1	5	1	5	15				
7	1	create push campaigns from SFMC	3	1	9	1	9	23				
8	1	import sms lists throught SFMC	3	1	5	9	5	23				
9	3	import push lists through SFMC	5	1	9	9	9	99				
		Technical importance score	137	169	177	89	197	769				
		Importance %	18%	22%	23%	12%	26%	100%				
		Priorities rank	**4**	**3**	**2**	**5**	**1**					
		Current performance										
		Target										
		Benchmark										
		Difficulty						1: very easy, 5: very difficult				
		Cost and time						1: low, 5: high				
		Priority to improve										

Figure 14 - Quality function deployment

Using a SCRUM methodology, we have defined the necessary tasks to start developing our solution. Making use of the defined tasks, we have designed a QFD (figure 14) to preview what would be the most critical tasks to be developed. The setup of a server, creating a web service, and developing methods to send communications were the most prioritized.

3.1.4 Value proposition

According to Osterwalder, the value proposition can be described as "(…)the definition of how items of value, such as products and services as well as complimentary value-added services, are packaged and offered to fulfill customer needs (…)" (Alexander Osterwalder Yves Pigneur 2000)[25]. It defines the "(...)specific strategy to compete for new customers" (Jalili and Rezaie, 2010).

The costs and the solution's autonomy are also a pivotal factor to bear in mind, especially if compared with other market solutions(as mentioned earlier on services survey) and future proof deployments. Autonomy here is our service's ability to connect to and be requested by any service in a perspective that the current marketing service changes, our integration must be fully compatible.

Our solution is based on the premise that it will improve efficiency in the communication process, reducing the amount of time spent setting up and maintaining a campaign.

3.1.5 The business model of canvas

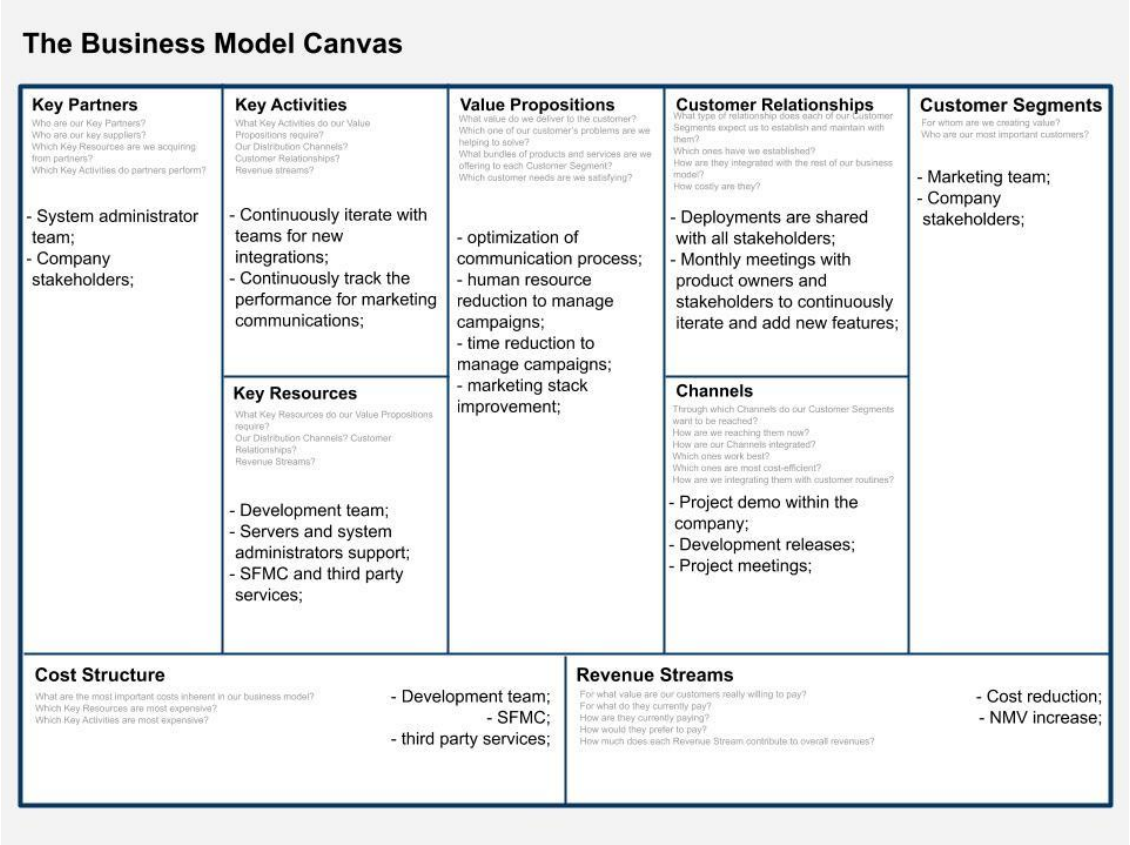


Figure 15 - Business Model of canvas

The business model of the canvas is a tool that provides a clear view of the business strategy of a project. It is represented by the figure 15. Below we have detailed each point.

Key Partners

Partners and suppliers to take into consideration our proposal, as well as its activities and resources. In our case, we will need the system administrator team to set up the server to host our project. Also, company stakeholders would act as partners to define the path to move as the product is developed.

Key Activities

Answers the questions about our distribution channels and leading activities our value proposition requires. It also refers to any revenue stream or customer relationship from our project. In this project, the one principal activity is the continuous iteration with stakeholders

for new services to integrate and add to our solution. Jumia stakeholders would act as customers from our project, so it is essential to keep track of our product deliveries' impact.

Value Propositions

The value we deliver to the customer, the problems our solution will solve. As stated previously in this chapter, the value our project delivers is process optimization, therefore reducing the time and resources allocated to marketing campaigns creation.

Key Resources

The critical resources, distribution channels, revenue streams, or customer relationships in use on our solution. Our solution will require a development team to design and create the integration service. System administrator teams will also be vital to provide and maintain the server to host our project. Finally, an active license in SFMC is needed to be used as the core platform to connect to our service and integrate with third party services.

Customer Relationships

Customer relationships refer to the costs and dynamics established with our customers. In this project, Jumia stakeholders and local teams will be our customers. Stakeholders will define the product - services to integrate with campaigns to automate - whereas local teams will use our service to automate their campaigns. There would be biweekly meetings with stakeholders to evaluate the features deployed and to define the next tasks. It is expected less contact from local teams, as they will define each campaign once(time to dispatch, routines, and content).

Channels

The channels in use to promote and communicate the product. In this case, there will be a project demo to showcase its features. Development releases will also be communicated to all managers and product owners via email for visibility. Finally, all new and upcoming features will be communicated on the project meetings.

Customer segments

The segment of customers we are creating value and targeting. Concerning this point, the marketing team and its stakeholders are the segments we are targeting.

Cost structure

The cost structure responds to the costs inherent to the development of this project. Our project costs are mainly development team time cost, SFMC service license cost, and any third-party service we use to communicate with customers.

Revenue stream

This point represents the revenue impacts of our solution. Namely, our customers are willing to pay, how much a similar solution currently costs, and how much our customers are paying now. There is no service to integrate all communications within the marketing team; therefore, it is impossible to compare. However, it is expected to spare less by reducing the resources allocated to communication processes. Furthermore, connecting with other tools will allow Jumia to send recommendation campaigns and increase the net merchandise value(NMV).

4 Solution Design

4.1 Solutions assessment

Our approach was to search on the webto gather available market solutions and request demos from its CRM communication products. The comparison was based on key features that the service must deliver: allow for multichannel communication, allow for integration with Jumia communication channels, and availability of an API. Another condition for the listed solution was sending the number of daily messages needed by Jumia, namely 13 million push notifications and 10 million emails. Other functionalities were considered as good to have, though not essential.

Worth to mention that this survey took place in mid-2018, henceforward all the features available refer to that date.

The selected tools and features are presented in Table 2 to ease comparison among them.

	Integration				**Other Features**			
Tool	**email**	**push**	**SMS**	**Facebook(social)**	**tracking**	**user-triggered interaction**	**API**	**cross channel**
Braze	x	x			x	x	x	x
SFMC	x	x	x	x		x	x	x
Oracle Responsys	x	x				x	x	x
Urban airship	x	x	x	x	x	x	x	x
Adobe Toolsuite	x	x	x	x	x	x	x	x
Middleware	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)

Table 2 – Comparison between available solutions

In the end, the products were very similar in their offer. Multichannel, user-triggered interactions and API availability were available for all the services, except Braze and Oracle, that lack the African continent's SMS channel.

As mentioned earlier in this document, SFMC was in use for some years in Jumia. Local teams had already been trained to use it, so change to another service would carry a higher cost to the company. Given that situation, and due to signing constraints, stakeholders decided to keep all processes within SFMC.

4.2 Domain Modelling

A domain model is a representation of the various concepts inherent to a given solution. It includes not only their attributes but also the relationships with each other.

It can be defined as "(...)a rigorously organized and selective abstraction of the knowledge." [26].

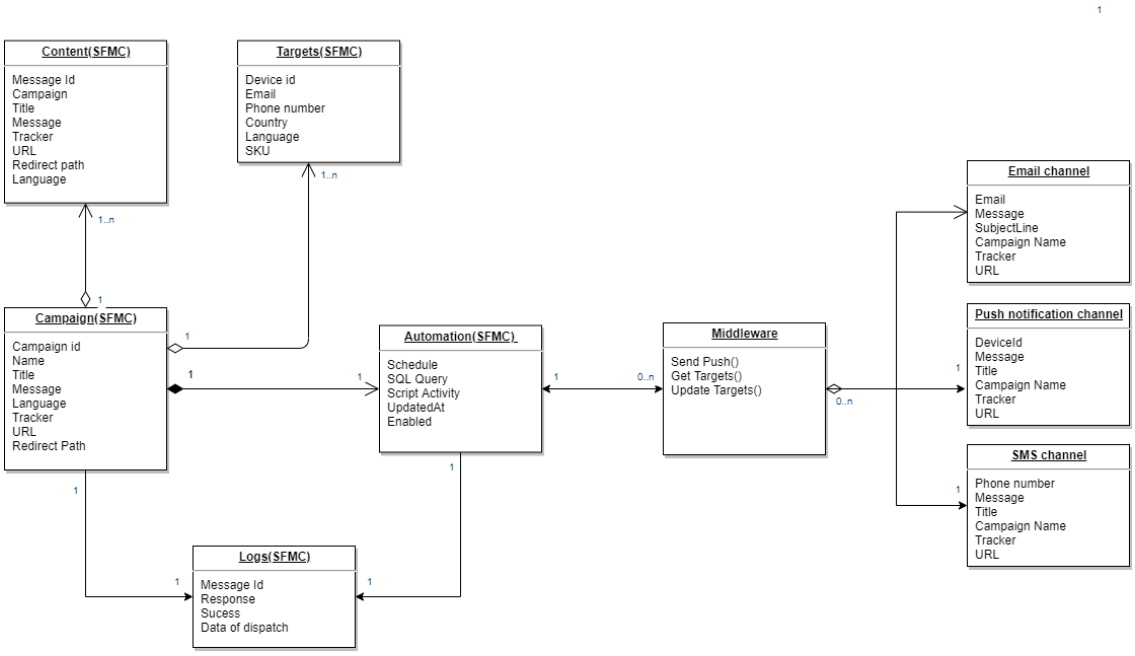


Figure 16 - Domain model for our solution

In our project, the domain can be summarized into three essential entities:

- campaign;

- middleware;

- channels(third party services).

Its relations and subdomains are illustrated in Figure 16.

4.3 SFMC integration assessment

4.3.1 API integration with third-party services

One possible approach to integrate external services to SFMC was to use the available service APIs [27].

Modern APIs require authentication to access it and often use a hashing algorithm to process these payloads. Hashing algorithms are mathematical functions that condense data to a fixed size [28].

When connecting to a service, we have noticed that the hashing function of SHA1 was not available from the SSJS Core library.

The first approach to overcome this problem was to define the function within the activity and reference it. However, it took too much processing when running the activity, and the activity throws an error. We have investigated parts of the code and found out that the issue was related to a bitwise operator on a floating-point function, returning unexpected results.

The effort to perform these changes accessed by the SFMC backend software team, and after some discussions, they have shared with our team that such change will impact few customers, and the number of changes and work to be done will be significant. They will ask for us to request a quotation from our commercial agent, share the timings and next steps.

From CRM stakeholders, a decision was made to avoid this path, as it would take much time to get done, and also it would carry more costs.

As a consequence of this limitation, some APIs, like Accengage Rest API, were not able to authenticate with SFMC. This reason leads us to exclude the direct API connection from SFMC to external services.

4.3.2 Custom activities from Journey Builder

Custom activities are a type of activity in use within SFMC Journey builder. It allows for running external code within a journey, by using an iframe and passing and retrieving data.

Despite allowing a seamless integration, this solution would not suit our use case, as it would run only within Journey Builder, and the CRM team does not use user journeys. Instead, our team uses the SFMC Automation studio for campaign launch and data management, and for that case, custom activities were not available.

Such limitations made us opt to develop our connector/middleware to ensure no significant technical constraints.

4.3.3 Middleware

As previously defined, middleware is a service that processes data from one service to another, allowing them to communicate [29][30]. In our case, this service will allow SFMC requests to communicate seamlessly with any third party service. As future proof, a configuration file should take part in the service architecture to ensure the compatibility with different systems.

Despite the effort in development and service maintenance, a middleware allows us to customize the solution and grant all the features required in this project. Furthermore, it assures that new features could be added without any constraint if the service evolves to a product.

Therefore this was considered to be the most reliable and scalable solution to deploy.

4.4 Solution requirements

The essential requirements for our solution are as listed:

- Allow for integrating all campaigns for all countries into one tool;
- Allow to connect SFMC with different services;
- Deploy an agnostic solution in what concerns technology and platform;

Given this project will take place on a marketing related team, we do not own or maintain any server or database. For that reason, and not to commit to security concerns, we will want this system to be as straightforward as possible. We decided to store all data on SFMC data extensions, leaving servers managed and set up by system administration teams.

4.4.1 Current integrations

As of this point of the project, the systems in need to be integrated with are:

- Accengage;

-Infobip.

The former is a service to create and dispatch push notifications, while the ladder is a service to send SMS to customers of Jumia.

Both services were hired before this project started, and due to signing constraints, they were kept. Therefore, the choice of services to integrate was not on this project's scope, as those decisions were taken previously by the stakeholders. During this project's deployment, the push notifications were prioritized, and the Infobip service was kept on hold. For that reason, we will focus on push notification integration.

4.5 SFMC technical limitations

SFMC has inherited ExactTarget architecture (some old URL domains are still in use, namely documentation).

During this project's development, we have noticed some limitations that could impact our solution, namely the hashing problem described in the previous chapter in point 3.2.

The script activity runs server-side javascript, without allowing any other library, besides SFMC "Core" library [31]. Also, every code has its scope, not allowing to reference other activities or store any environment variables. That library provides a basic set of functions, which limits our ability to compute code. SFMC also limits the number of rows on an HTTP request payload to 2000.

Like script activities, query activities on automation studio limits to 30 minutes per process before throwing a timeout error. Moreover, SFMC provides no error logging besides the front end displayed – "Error on script activity." Not having clear error logging or a debug console has proven to be a blocker to understanding errors. We had to create specific data extensions to write any existing errors from call back functions written on our code.

4.6 Dependencies from other teams

The CRM team scope is essential to highlight here. Given that this team's focus is marketing communication and CRM, we do not own or maintain any server or database. Henceforward, if any infrastructure or database is needed, we need to make a formal request to stakeholders to request the concerned team. In this case, we had decided to develop a middleware, so we requested the system administrators team the setup for a server to run our project. It is also required to set a development environment (production and development) and software versioning (using git); we had requested system administrator team support.

We have met with senior developers from teams that had previously designed similar projects to advise on the best architectures and technologies to use. The security team was also consulted to assure there was no blind spot on the solution architecture.

4.7 Architecture overview

The solution must allow for seamlessly connect SFMC to an external service. Middleware must then receive the requests, handle it, and perform the desired action on the external service.

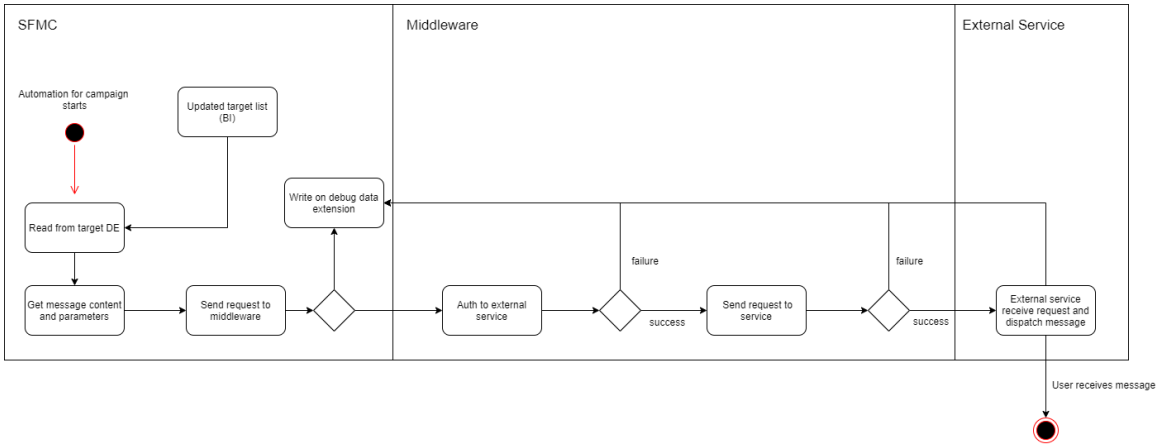


Figure 17 - Activity UML for the proposed solution

Figure 16 presents an example of a possible solution for integrating SFMC with third party services. Middleware would process all requests to push notifications and SMS, via SFMC automation studio SSJS activities. Using the script activity, we request our middleware via HTTPS, which would process it and redirect the concerned service with the requested action.

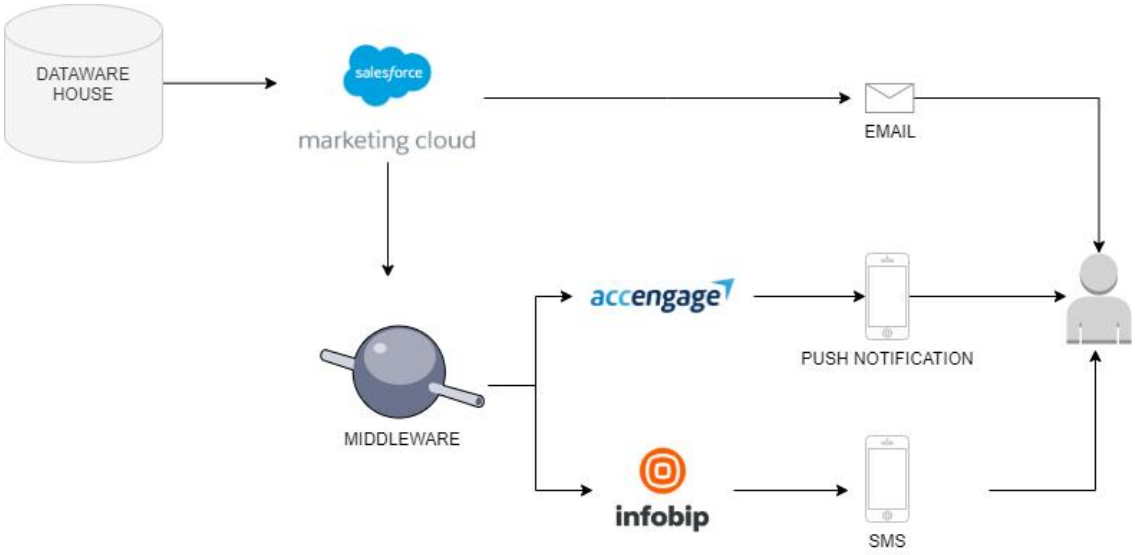


Figure 18 - Scheme for middleware usage on CRM setup

The server to deploy the middleware must require a user/password to authenticate. It will then set the user credentials stored within a JSON file to authenticate the external service. It receives a payload that should follow the format as detailed on documentation[32]. Then it will run the code to proceed with the desired action (send push, send list, send SMS) on the desired service, and return the third party API response to SFMC, which will store it on a logging Data Extension. If succeeded, the message is sent to Jumia customers/subscribers. In figure 17, we illustrate this setup.

4.8 Technologies and frameworks

To ensure we do not have any duplicated work, we have met with senior developers from teams developing APIs. We have aligned on languages and frameworks to use. Ultimately, they will provide enough insight on what to use for a specific case – an integration middleware in our case - and which frameworks were suitable.

In the end, we decided to use Node js as a framework to use. We have also decided to use the web application framework, Express JS, already used for other backend services in Jumia. The server was deployed on Amazon Web Services(AWS).

4.8.1 Node JS

Node JS is a Javascript library that is light and stable enough to use in this project[33]. It was used on other projects on Jumia, so we count on senior developers' expertise from other teams, in case needed. We should install Node and Node Package Manager (NPM) on our machine to make all packages available. All libraries would be installed using NPM.

4.8.2 Express JS

Express JS is a Node lightweight framework, specialized for routing and back end services[34]. According to documentation, routing "(...)refers to determining how an application responds to a client request to a particular endpoint(...)" [35].

```
app.METHOD(PATH, HANDLER);
```

Code 1 – express JS route definition [35]

The above code exemplifies a routing definition structure. Where:

- app is an instance of express;
- METHOD is an HTTP request method, in lowercase;
- PATH is a path on the server;
- HANDLER is the function executed when the route is matched.

These concepts and detailed explanations of available methods can be found on the Express documentation page[35].

5 Solution deployment

In this chapter, we will thoroughly describe the steps taken to deploy our solution.

After reviewing the available options, and for the reasons we have disclosed in the previous chapter, we decided to move towards middleware deployment that would connect SFMC to other services. This system should assure that all communications could be set up on SFMC and dispatched through external services.

When the decision to deploy a middleware was made, we have started by setting up the server to deploy the code and set up the development environment and the Github page. With this, some security concerns were raised, which led us to visit Jumia's security team.

After clearing all security questions, we have started developing the service on the provided server. The development of the middleware and the setup on SFMC were the most time-consuming part of this deployment. Expectedly, they are the core of this solution.
All of these steps are detailed below.

5.1 Server Setup

The server configuration and availability lie out of the CRM team scope; therefore, the SRE Team met with the infrastructure setup.
After sharing the language and technologies in use in the project with the team, the team deployed the infrastructure, hosted on A.W.S. System administration also configured the development environment and Github repository. We have used one development branch (Dev) to test and develop the service and production to deploy the live code. Given that only one person would be in charge of this project, there was no big concern over merging branches or overriding code.

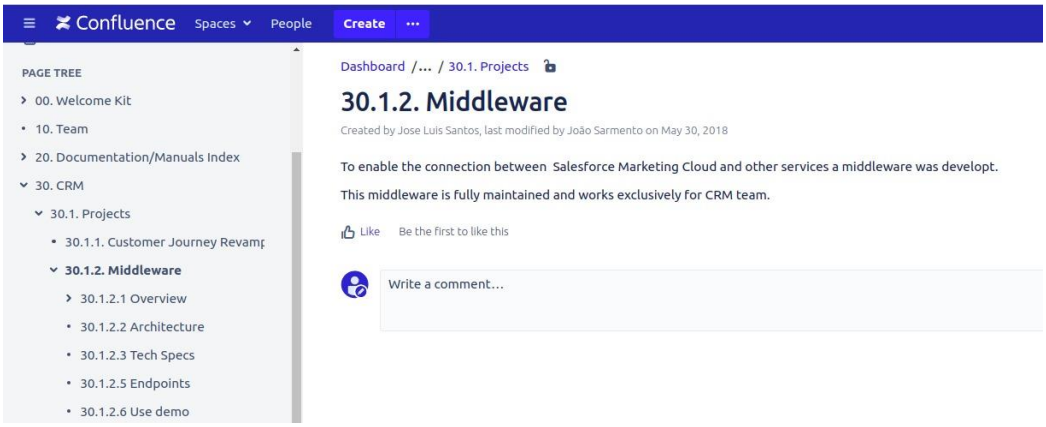


Figure 19 - Documentation page for the new service

At the same time, we have created a Confluence page(see appendix A) - to store all related documentation.

5.2 Security considerations

In this stage of the project, some security questions arise. We were concerned about user's data compliance with General Data Protection Regulation (GDPR), eventual flaws on the transport layer, and unintended access to this service. For that reason, we have decided to reach the Jumia security team for best practices on security. We have shared the project purposes, and some decisions were made:

- The server should be private, with user-password authentication;
- The API transfer protocol should be HTTPS, to assure all data encryption;
- Whitelist our SFMC IP address to limit the request source to our service(Salesforce provide a range of IPs for each of its server instances);

Considering all the security-related challenges to data storage, we have decided that all data should be stored within SFMC. This decision removed all the necessary setup for any database and security concerns with data; inclusively, GDPR. Not having databases on the solution meant that all the content to be sent, execution logs, and parameters to be sent are stored on SFMC data extensions.

5.3 Scalability considerations

Due to the need to increase the number of targets to reach more people, we need to ensure that our service can scale. As shared before on this document, CRM team sends 13 million push notifications every day. So this is the minimum amount of messages that our system must deliver.

The first campaign to deploy using the middleware was a challenge from a scalability point of view. Until now, the most significant campaign using push notification in Nigeria dispatched nearly 2 million messages from Accengage for android devices. The campaign to test using middleware would start on around 2 million, but predicted to scale up 8 million messages. Therefore we could gradually assess the effort and manage the audience's growth accordingly.

The first limitation we face was the higher number of rows(2 million) on SFMC. SQL and script activities throw a timeout error as soon as a process takes more than 30 minutes running. For that reason, we needed to create auxiliary data extensions with less number of rows, and use SQL query activities to split the master data extension into smaller ones.

On the service provider end, Accengage API had an API that limited the number of messages to send to 100 users. This limitation has proven a blocker for scaling communications due to the high amount of time needed to dispatch around two million messages(more than 12 hours). To address this problem, two actions were taken: in one front, we cycled through the data extension to assure that only 100 users were sent per request(code 5); on the other hand, we reached Accengage support team and shared the problem. In the end support team could develop a fix and increased the amount of messages to 2000 users per request. This impacted the performance of our service and reduced the time of dispatching the campaign considerably.

The code to overcome this limitation was developt under SFMC and not the middleware, as we consider this would be a fix for one specific provider.

5.4 Middleware service

The first step towards service development was installing NPM. NPM stands for Node Package Manager, and as the name implies, it is the engine that allows for Node js code compilation and debugs. As previously decided, after setting up NPM, we have installed the following packages: express js, require, and Newrelic.
The first is the framework in use for the middleware, require is a package needed to call other files, and the latter is an extension for monitoring our API calls on an external service.

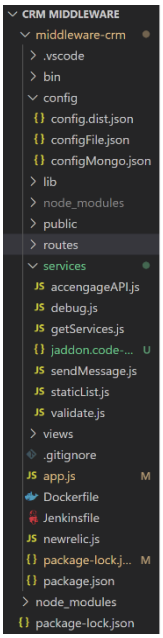


Figure 20 - Folder structure for the project

After this initial setup, our project folder structure looked like the image above.

5.5 Middleware code overview

One of the advantages of using express js is that it provides the barebone of service on the initial project. Hence, we can reuse that structure and build our solution easier and faster. Below are shared relevant parts of the code to understand how the different parts of the code interact.

```
var nr = require('newrelic');
var createError = require('http-errors');
var express = require('express');
var path = require('path');
var cookieParser = require('cookie-parser');
var logger = require('morgan');

var indexRouter = require('./routes/index');

var app = express();

// view engine setup
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'jade');

app.use(logger('combined'));
app.use(express.json({limit: '50mb'}));
app.use(express.urlencoded({ limit: '50mb',extended: false }));
```

```
app.use(cookieParser());
app.use(express.static(path.join(__dirname, 'public')));

app.use('/', indexRouter);
```

Code 2 – part of code from app.js file

Express JS is a lightweight framework based on node js and is a simple routing solution for HTTP. The endpoints definition and all controllers are bonded on the app.js file, as per the code above.

```
//post request with params to send PNs
router.post('/', function(req, res, next) {

    getServices.getToken().then(function(data){

        //Send PN
        sendMessage.sendPushToUsers(data, req).then(function(dt){
            res.send(dt);
        })
        .catch((e)=>{
            res.send(e);
        });
    })
    .catch((e)=>{
        res.send('erro a criar o token');
    });

  });

module.exports = router;
```

Code 3 – part of code from router file indexRouter.js

Then the router defines the services to be called to each endpoint and the response handling(code 3). Endpoints here were added as per the product features defined ahead(send push and update segment).

```
       //set title field specific for each OS
        if(param.body.OS=="iOS"){
            bodyPN.messageData.subtitle = param.body.content[0].subtitle;
        }else if(param.body.OS=="Android"){
            bodyPN.messageData.shortText = param.body.content[0].shortText;
        }else{
            reject("no OS found");
        }

        bodyPN =JSON.stringify(bodyPN);

        var strToHash = bodyPN + apiKey + timeFinal;
        var accSign = crypto.createHash('sha1').update(strToHash);
        var accDigest =accSign.digest('hex');

        //api call
```

```
        try {
            request.post({
                headers:{'Authorization': accToken, 'Accengage-Signature':accDigest,
'Accengage-Time': timeFinal, 'Content-Type':'application/json;charset=utf-8'},
                url:'https://api.accengage.com/v1/me/apps/'+partnerId+'/send',
                body:bodyPN,
            }, function (error, response, body){
                resolve(response);
            });
        }
        catch(e){
            reject(e);
        }
```

Code 4 – part of code from service file sendMessage.js

Finally, the services would process the data received, encapsulate it on a JSON object - alongside the config file values - and request the desired service(code 4).
Besides the service to send and update segments, we have written a class for authentication. This class generates a token that is stored on the configuration file. Authenticating on the external service happens before every request. The service checks if the token is valid; if not, it generates a new token. As an example, for Accengage service, each security token lasts approximately 24 hours.

```
{
    "apiKey":"test",
    "partnerIdiOS":"test",
    "partnerIdAndroid":"test",
    "NLT_AND_CUSTOM":"test",
    "NLT_AND":"test",
    "NLT_IOS_CUSTOM":"test",
    "NLT_IOS":"test",
    "serverPort": 8080,
    "newrelic_enable": false,
    "newrelic_license_key": "random_key",
    "newrelic_app_name": "CRM MDWR",
    "staticList":"false"
}
```

Code 5 – part of the configuration file options

The configuration file is where all data relative to an external service is stored. Authentication values and optional parameters like enabled features are stored in the file, as shared on the code example above. This level of abstraction allows us to make each service generic and assure that each new service can be added with ease. in the future

Figure 21 - New relic page for monitor requests

After testing the server communication with external service, we received a 200 response, ensuring the correct payload and access to the service. The logs were collected on both the monitoring service Newrelic and SFMC data extensions . In Figure 20, we share an example of a page on the Newrelic service to monitor our requests. In the second case, the response comprises both the result from middleware and the external service we are reaching.

5.6 SFMC Setup

As per the goal of this project, we have centralized all the communications via SFMC. Therefore, all data must be available there. The data includes message content, target definition, execution logs, and service requests. Also, all automation run and is set up on SFMC.

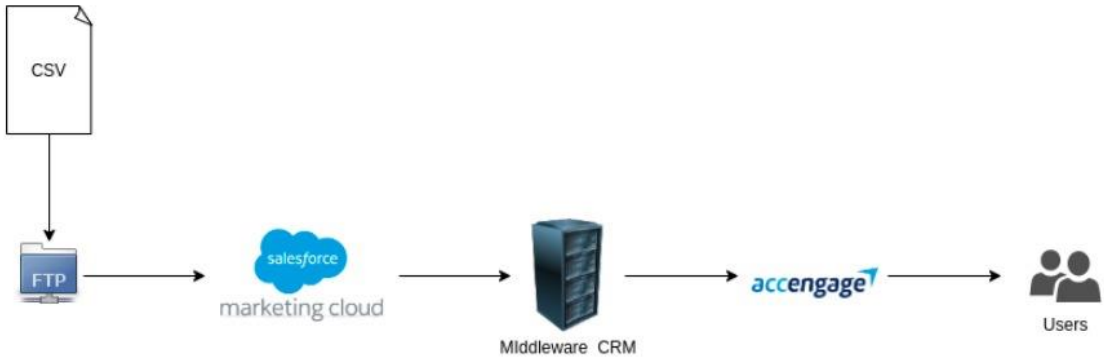


Figure 22 - Data process for update targets

Due to its volatility and constant updating, the target data is refreshed daily from BI team and made available on SFMC through an automated process. BI team updates all the user's data drop a CSV file on an SFTP server connected to SFMC, which will trigger a process to update this data on the concerned data extension. Those targets will then be used for our service to send communications. An example of this process is illustrated in Figure 21.

For the campaign and message content, it should not change much. Having all services connected to SFMC allows the same campaign to be set once. Using available placeholders(customer name, favorite product, for example), the message would fit a broader audience.

For a more detailed view of the setup, the data and automation configuration will be described separately.

5.6.1 Data Storing

Data extensions are a type of list within SFMC It allows one to define its field name and type, and can be used in any process within SFMC - either email sending, reports, scripts, among others.
In our project, we will use it for three purposes:

- Store campaign information;
- Store list of targets;
- Log all useful information from each request;

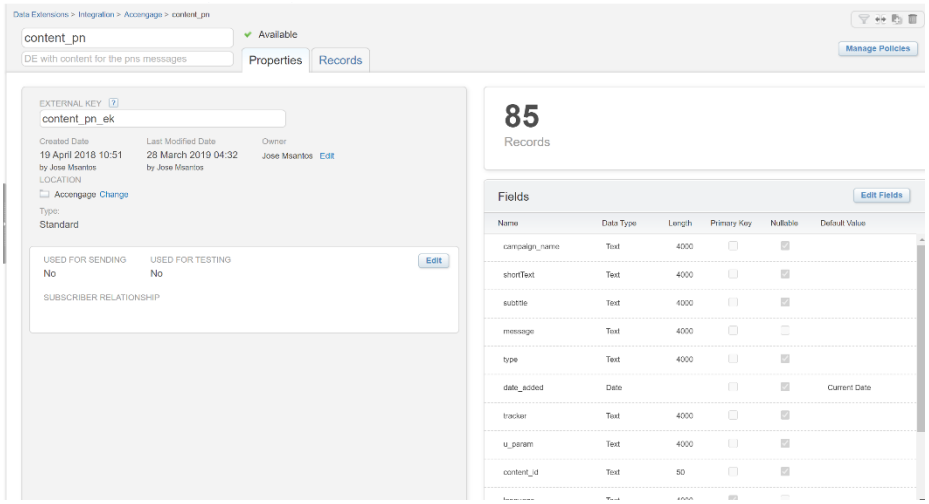


Figure 23 - Data extension to store messages content and parameters on SFMC

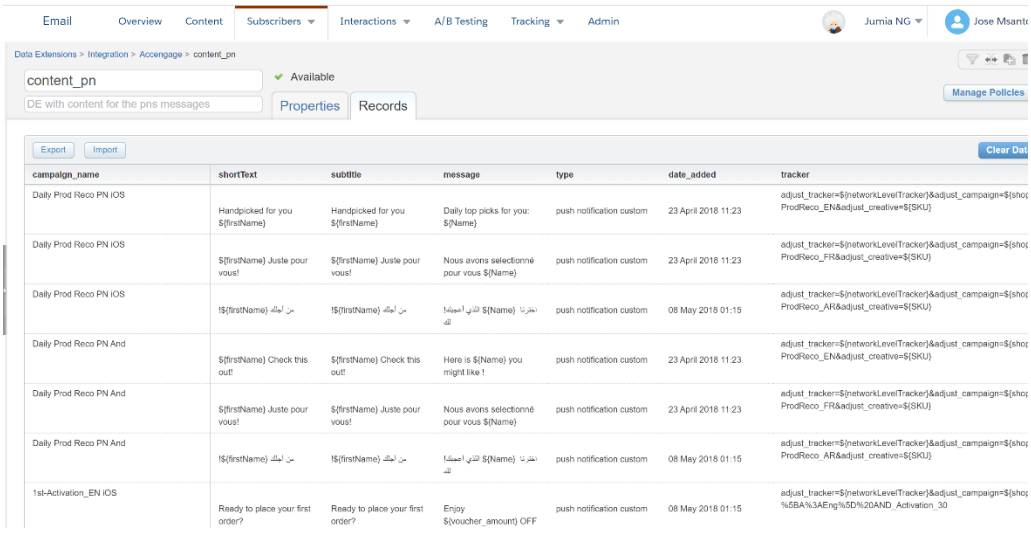


Figure 24 - Content data extension with message data on SFMC

For every campaign we have, there is some critical information to store, namely:

- the name of the campaign,
- the message to send,
- the id of the campaign on the external service(Accengage in this case),
- the target to send the message,
- the language to display the message,
- the operative system to send the message,
- the tracker to append the message,
- the URL to redirect the message

We have created a data extension with these fields and update it with the needed content. This data extension structure and content are shared on the figures above(22 and 23).

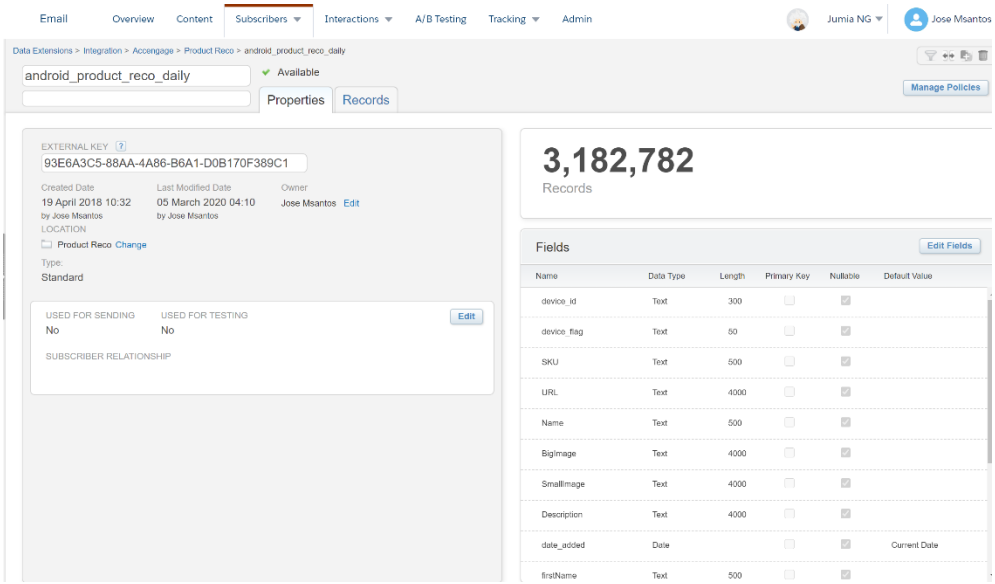


Figure 25 - Target data extension on SFMC

Another data extension was created to store target data with the fields, as shown in the image above. In our case, the field in use were phone numbers, or, more often, deviceids. The device id is a field that allows some services to target a smartphone that has a specific app installed; in this case, the Jumia app. For every campaign to be created, a new data extension must be created for all necessary languages with its targets. Because the goal is to centralize communications, we need to have each data extension with users from a specific language. In dispatching time, we target the users with the correct language. Target data is imported as automation that reads a CSV file from SFMC SFTP and feeds it to data extension. An example of a target data extension is shared above in figure 21.

Finally, we will query the records on the campaign data extension and break it in auxiliary data extensions, to ensure no data extension has more than 160 000 records. To split big data extensions into smaller ones comes as a workaround for two limitations pointed previously on point 4.4- SFMC technical limitations – query timeout and limit on HTTP payload. After some tests, we have noticed the script activities timed out due to long requests(more than 30 minutes). We have removed the size of data extensions to ensure it will not reach its limit. Then we verify that 160 000 records would fix this problem.

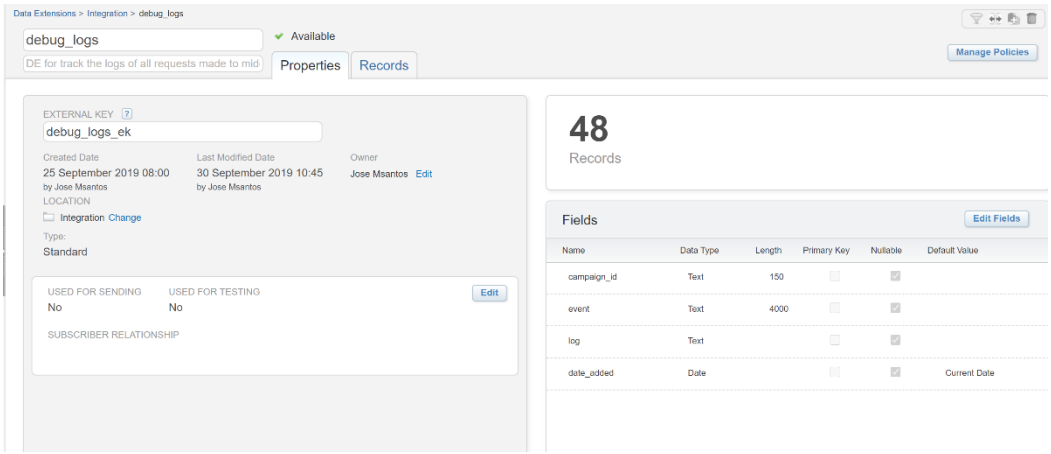


Figure 26 - Data extension for store middleware and external service logs on SFMC

Lastly, we will define a data extension for the execution logs as per the image above. The fields store the outcome of the request: success/failure, the error message in case of failure, and the execution time and date. Each campaign should have one debug data extension.

5.6.2 Content setup

In order to optimize the communication process, we have made changes to the content creation process. To simplify the process, and given that all messages were always created in three languages: arabic, french, and english, we realize there was room for improvement. Therefore, instead of creating content for each country, we have created content in each of the abovementioned languages – arabic, french, and english – to spread on the concerned countries. Reducing the number of messages eased the content creation process, as one message would suit many countries. Simultaneously, it simplifies the campaign creation, as fewer campaigns were needed to be created.

The content data extension was designed to allow such logic to work: the language field, alongside with query activities, permitted to filter the right content for the right countries.

In the end, both the targets and content data extensions rely on the language field to filter the correct message to users when sending the campaigns.

5.6.3 Automating process

SFMC has a tool for automate actions named Automation Studio. It allows for automate major actions such as email sending, querying data and run server side javascript. Below we detail the steps done to setup the automation of our service on Automation Studio.

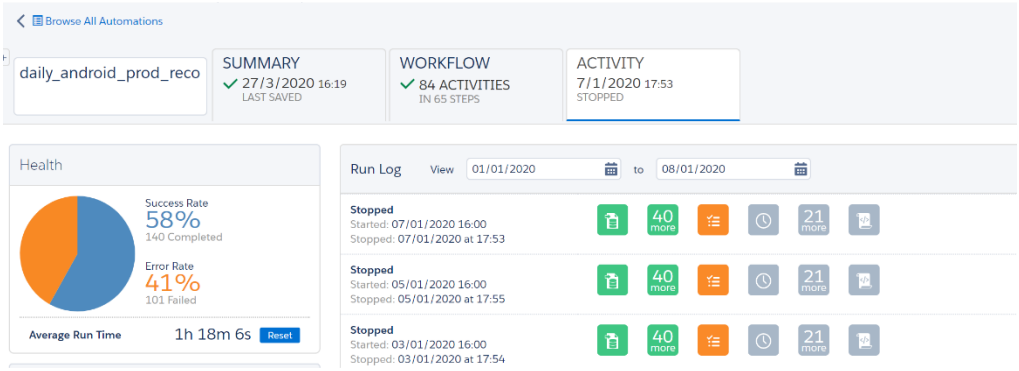


Figure 27 - Automation for dispatching push notifications via middleware

The communication process uses an automation studio to automate the sending of campaigns, as presented in figure 23. For each campaign, we must have automation created with all the necessary activities. This automation would do the following tasks in the following order:

- Update the target segment list;
- Check for message content;
- Request the campaign to send via middleware service;
- Record all execution logs to data extension;

Each automation can be either scheduled or triggered by an event. In the case of campaigns, all automations were scheduled to reach the target audience at the best possible time.

```
var content = content_de.Rows.Retrieve(filter);
var url = "https://crm-mds.jumia.com/sendPushNotification";
var contentType = 'application/json; charset=UTF-8';
var headerName =["Authorization"];
var headerValues =[test'];

do{
    var results = [0,0];
    targets = Platform.Function.InvokeRetrieve(rr, results);

    var runstatus = results[0];
    var requestId = results[1];

    if (targets != null) {

        debug_de.Rows.Add({event:"arr size"+targets.length});

        var pns=[];

        for(var i =0; i < targets.length;i++){

            pns.push(targets[i]);
```

```
            //processar dados de targets de 1000 em 1000 linhas
            if(i % 1000 == 0) {

                try{
                    var payload = Platform.Function.Stringify({"content":content,
"target":pns, "OS":"Android"});
                    var response = HTTP.Post(url, contentType, payload, headerName,
headerValues);
                    var r = Platform.Function.Stringify(response);
                    debug_de.Rows.Add({event:"sucesso1"+r});

                }
                catch(e)
                {
                    debug_de.Rows.Add({event:'auth1',log:'erro: '+e});
                }

                pns = [];
            }
        }
```

Code 5 – Script activity piece of code with connection and log records

The script activity is a crucial step in automation. It will request our middleware using the data stored on SFMC. An example of code executed on that script activity is shared above(code 5). This code will connect the two services, and ultimately, perform the integration with third party service.

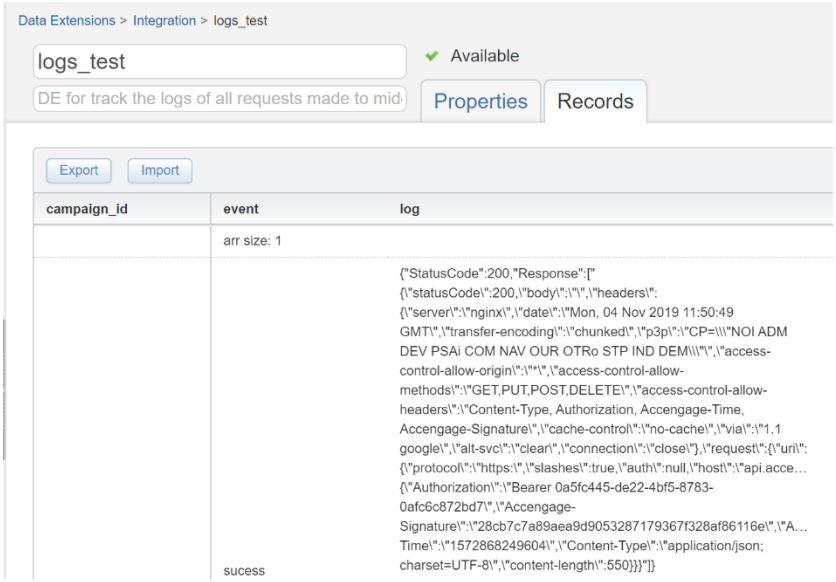


Figure 28 - Logs stored on SFMC from middleware and external service

Before running the automation, we have run the script activity to ensure the proper communication between SFMC and our middleware service. After testing the connection on Postman, we tested it on SFMC, and it successfully reached our service, as shared in Figure 27.

```
{
"serviceType":"Push",
"language":"EN",
"countryCode":"NG",
"data":{
       "deviceId": ["device1", "device2", "device3"],
       "message": "Welcome to Jumia",
       "imageURL": "http://jumia.com/logo.png"
 }
}
```

Code 6 – Example of a Postman payload

Postman was used to testing and debug any payload errors by monitoring server response. In code 6 we have an example of a payload to test simple push notification send to devices.

5.7 Deployment overview

After the data extensions were setup, automations were running, and the service is live, our setup was completed. We tested the service end to end, by creating a new message on SFMC and dispatch it via push notifications to a test device.

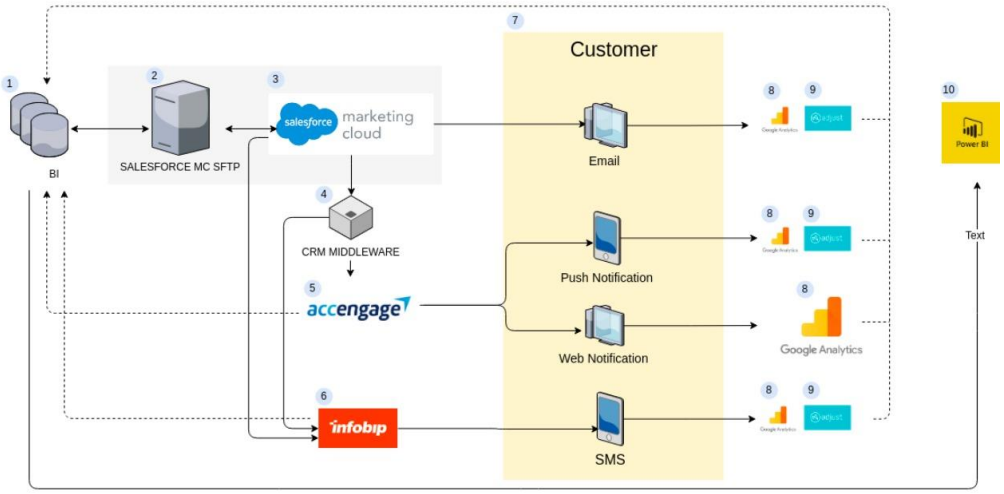


Figure 29 - New CRM stack

In the end, the new architecture for the CRM stack is represented in the image above.

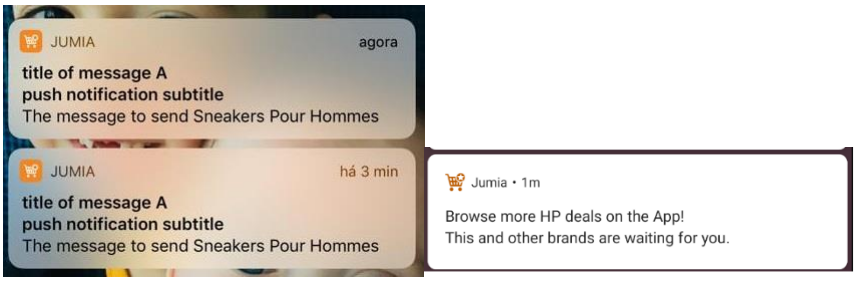


Figure 30 - push notification sent to users on iOS(left) and Android OS(right)

The automation had successfully run, and the messages were sent to the desired audience. An example of a push notification sent to mobile devices can be seen in figure 29 for both iOS and Android OS.

6 Solution evaluation

This section describes the assessment of our proposal with all its intervenients - stakeholders and local teams. After the solution was deployed, we seek to assess whether the solution has met its initial goals.

Our project would use three evaluation methodologies: customer inquiry, to assure the project was aligned with customer(stakeholders in our case) expectations, A/B test, to weight our solution against the former process, and Quantity Evaluation Framework, to have measured the deployment of the functional requirement of the provided solution.

Below are detailed the methodologies and their data sources.

6.1 Qualitative Evaluation Framework

Qualitative Evaluation Framework is a tool that allows us to measure the project's overall goal, weighing all requirements. In Appendix B, we have the requirements for this project.

Given all the features expected, we have detailed all the necessary tasks to deliver, we groomed it, and add to our sprint backlog. In the end, all the requested features were delivered: send push, get data, update data.

6.2 Users feedback

The user's feedback is important here to have a transparent perspective from the end-user of the solution. Due to this, this step will only be possible after the solution is deployed.

We have requested key users(users that make the most use of the tool) for its feedback that we will share below.

6.3 A/B Tests

Our hypothesis relies heavily on stakeholder's decisions. For this reason, we will use A/B tests to assess the value of our solution.

A/B tests use similar conditions – in this case, targets and campaign structure – with different setups and compare final results. One solution A will be set up with an older setup for campaign creation(with previous CRM team processes), whereas solution B will be using the new system for campaign setup. The same audience, content, and- if possible – dates will be used on both A and B campaigns to avoid biased results.

In this case, we had used the 'product recommendation' campaign to A/B test using the push notification channel. To do so, we have defined an audience to target. After that, we had split the audience, where 50% of it would be used on A test, while the other 50% would be used on the B test.

A test will use the old process: we have requested all countries to deliver the message on a given day.

B test will use the newly created setup, using the proposed solution of this thesis. Hence, the CRM engineer had to setup SFMC to deliver the campaign on the same day as A test.

In the end, A test had involved 14 people: Head of CRM for content approval and review, CRM engineer for setup campaign on Accengage platform and SFMC, and CRM developers to create the content and update on the Accengage platform. B test required five people to deploy: Head of CRM for approval and review, CRM engineer for deploy SFMC setup, and one local team member for writing English, arab, and french content.

Regarding the time to deploy, A test took around 51 hours, combined all tasks: four hours for content creation for each country, 2 hours to create a campaign and automate on the Accengage platform, and one hour to review the content. B test had taken the same amount of time; however, given that only one language was needed, it has taken 15 hours.

6.4 Hypothesis

The hypothesis will define the goals of this thesis. It will define whether the provided solution was or was not valid.

Our goal with this project is to integrate third-party services within one core platform, hence improving the CRM communication process.

Therefore we will define two hypotheses: the null hypothesis and alternative hypothesis.

Null hypothesis (H0): the integration does not improve the communication process.

A hypothesis is null if all the conditions are met:

- there is not a decrease in time to set up and maintain new campaigns;

- there is not a decrease in resources used to set up and maintain new campaigns;

- integration cannot connect to external services;

The alternative hypothesis (H1): integration does improve the communication process.

An alternative hypothesis is valid if all of the conditions are met:

- there is a decrease in time to set up and maintain new campaigns;

- there is a decrease in resources used to set up and maintain new campaigns;

- the integration allows to connect SFMC to external service;

6.5 Metrics and data sources

To assess the hypothesis mentioned above, the following metrics should be considered: Time to market for new campaigns, resources allocated to each campaign setup, and the connection to third party services.

Time to market

To measure time to market, we will use the JIRA ticketing system to manage our team's tasks and time log. Given that the CRM team uses SCRUM methodologies in its projects, we can measure the time to deploy campaigns, from our ticketing system time log to the JIRA platform.

Allocated resources

Given that the Jira platform does not log all people accountable for the communication process, we will use the stakeholder's roadmap planning documents and the people allocated to all tasks.

Connectors to third party solution

We will use the external platform logs and match them with a campaign set from SFMC.

6.6 Technical evaluation

To evaluate SFMC middleware, we will validate the coding efficiency.

6.6.1 Unit tests and integration tests

Unit tests are used to evaluate and ensure the effectiveness of a function to work as expected. Therefore we will cover core classes on middleware with unit testing.

Given that our solution must allow different systems to communicate seamlessly, we found the need to proceed with integration tests. To do so, we used the postman tool to test the HTTP request to all endpoints on our service.

6.7 Results assessment

In the end, we have successfully deployed an integration that fulfilled the alternative hypothesis (H1). By the roadmap shared by stakeholders, and because campaigns are now created once, and not every day, our solution made possible the deployment of a higher number of campaigns. A process that initially required one person per country to deploy, on a daily routine, the same campaign on different platforms or services, is now set up once on SFMC, using middleware, and dispatch the same campaign to all countries' languages. The setup is now deployed by one member of the CRM team, hence the resources allocated are way less then it was with the previous setup.

Also, some users feedback were shared:

> **CTO** (uses the tool to demo possible solution on other projects)
>
> "The tool enabled us to deploy more communications with fewer costs and resources."
>
> **Head of CRM**(uses the tool to plan campaigns)
>
> "This project increased CRM team relevance, as it optimized a critical process for the company."
>
> **CRM developer** (uses the tool daily, to deploy campaigns)
>
> "This integration simplified the campaign setup, as it reduced the redundancy and repetitive work of creating contents all over different services."

In the end, the possibility to connect to external service was achieved, as exemplified in the previous chapter.

7 Conclusion

The development of this project was a long journey. Either by the lack of proper documentation, fragile development infrastructure on SFMC, or occasional business constraints, we have set a path for improving the Jumia communication process.

The solution proved very efficient, in concern of the number of people involved and communications sent. We have successfully deployed a solution that reduces the time and the number of people involved in the communication process.

The first campaign to use this system, the *product recommendation campaign,* showcased the solution's potential. It required the setup to be made on SFMC by myself, one person to create the content, and the Head of CRM to monitor the results. The campaign uses a push notification channel and is sent daily to around 8 million users. Centralizing the data, and using SFMC as a data source, permit BI team to produce a customized recommendation for each user. Our service reads it from a data extension and reaches 8 million users with content that is now more customized then the previously defined messages. Despite its not a direct, rather indirect consequence, this recommendation campaigns were enabled by our system and generated an expressive increase in user's conversion rate and NMV per campaign.

In the end, we had deployed a solution that will allow the CRM team to centralize its communications, hence optimizing the campaign creation process. Moreover, by allowing for any external service to be connected to SFMC, the middleware can either improve the marketing stack by integrating new services; or permit other Jumia teams to use communication services in the future without the need of SFMC.

8 Bibliography

[1] D. D. Raditha Hapsari, Michael D. Clemes, "The impact of service quality, customer engagement and selected marketing constructs on airline passenger loyalty," *Int. J. Qual. Serv. Sci.*, 2017.

[2] M. Ioannou, "Customer Relationship Management (CRM)," *Cust. Mark. Strateg.*, pp. 150–170, 2012.

[3] "Homepage | Jumia Group | Jumia Expand Your Horizons." [Online]. Available: https://group.jumia.com/. [Accessed: 25-Feb-2018].

[4] "Jumia shares fall below IPO price on New York Stock Exchange — Quartz Africa." [Online]. Available: https://qz.com/africa/1681874/jumia-shares-fall-below-ipo-price-on-new-york-stock-exchange/. [Accessed: 21-Feb-2020].

[5] P. Raman, C. M. Wittmann, and N. A. Rauseo, "Leveraging CRM for Sales: The Role of Organizational Capabilities in Successful CRM Implementation," *J. Pers. Sell. Sales Manag.*, vol. 26, no. 1, pp. 39–53, Dec. 2006.

[6] M. P. Bolos Cheristena, Idemudia Efosa C., "Conceptual Models on the Effectiveness of E-Marketing Strategies in Engaging Customers," *J. Int. Technol. Inf. Manag.*, vol. 24, no. 4, pp. 37–50, 2016.

[7] C. Homburg, D. Jozić, and C. Kuehnl, "Customer experience management: toward implementing an evolving marketing concept," *J. Acad. Mark. Sci.*, vol. 45, no. 3, pp. 377–401, May 2017.

[8] P. Drucker and S. Objectives, "Setting SMART Objectives Checklist 231," 1991.

[9] R. Paul and L. Elder, "Engineering Reasoning."

[10] K. Schwaber and J. Sutherland, "The Scrum Guide: The Definitive The Rules of the Game," *Scrum.Org and ScrumInc*, no. November, p. 19, 2017.

[11] "7 Long-Term Benefits of Content Marketing | by Carol Forden | Medium." [Online]. Available: https://medium.com/@CarolForden/7-long-term-benefits-of-content-marketing-f84f7230ffd3. [Accessed: 24-Aug-2020].

[12] “Personalized Content and Product Recommendation Engine.” [Online]. Available: https://www.dynamicyield.com/recommendations/. [Accessed: 25-Feb-2018].

[13] D. Das, L. Sahoo, and S. Datta, "A Survey on Recommendation System," *Int. J. Comput. Appl.*, vol. 160, no. 7, pp. 6–10, 2017.

[14] “Salesforce Completes Acquisition of MuleSoft | MuleSoft.” [Online]. Available: https://www.mulesoft.com/press-center/salesforce-acquisition-completed. [Accessed: 22-Feb-2020].

[15] “Custom Activity Configuration | Marketing Cloud Package Development | Salesforce Developers.” [Online]. Available: https://developer.salesforce.com/docs/atlas.en-us.noversion.mc-app-development.meta/mc-app-development/custom-activity-config.htm. [Accessed: 22-Feb-2020].

[16] “What Is Middleware?” [Online]. Available: https://web.archive.org/web/20120629211518/http://www.middleware.org/whatis.html. [Accessed: 22-Feb-2020].

[17] “Visualizing Integration Applications – Red Hat Middleware Blog.” [Online]. Available: https://middlewareblog.redhat.com/2016/07/11/visualizing-integration-applications/. [Accessed: 22-Feb-2020].

[18] Rafael Z.FrantzaRafaelCorchuelobFabriciaRoos-Frantza, "On the design of a maintainable software development kit to implement integration solutions," *J. Syst. Softw.*, vol. 111, 2016.

[19] “MuleSoft | Integration Platform for Connecting SaaS and Enterprise Applications.” [Online]. Available: https://www.mulesoft.com/pt/. [Accessed: 22-Feb-2020].

[20] “Terminology | IEEE Xplore.” [Online]. Available: https://developer.ieee.org/Terminology. [Accessed: 22-Feb-2020].

[21] G. D. Hughes and D. C. Chafin, "Turning new product development into a continuous learning process," *J. Prod. Innov. Manag.*, vol. 13, no. 2, pp. 89–104, Mar. 1996.

[22] P. A. Koen *et al.*, "Fuzzy Front End : and Techniques," *Ind. Res.*, vol. pp, pp. 5–35, 1996.

[23] T. Woodall, "Conceptualising ‘value for the customer’: an attributional, structural and dispositional analysis,” *Acad. Mark. Sci. Rev.*, vol. 2003, no. 12, 2003.

[24] W. M. Group, “QFD: Product Excellence using Six Sigma: QFD,” pp. 1–61, 2013.

[25] B. A. Osterwalder, Y. Pigneur, G. Bernarda, A. Smith, T. Papadakos, and J. Wiley, “Value Proposition Design: How to create products and services customers want,” *J. Bus. Model.*, vol. 3, no. 1, pp. 81–89, 2015.

[26] E. Evans, “Domain-Driven Design: Tackling Complexity in the Heart of Software: Amazon.de: Eric J. Evans: Fremdsprachige Bücher,” vol. 7873, no. 415, p. 529, 2003.

[27] C. Business, “The Evolution of Integration : A Comprehensive Platform for a Connected Business White Paper.”

[28] “The Difference Between SHA-1, SHA-2 and SHA-256 Hash Algorithms.” [Online]. Available: https://www.thesslstore.com/blog/difference-sha-1-sha-2-sha-256-hash-algorithms/. [Accessed: 22-Feb-2020].

[29] “What is Web Services Middleware? - Definition from Techopedia.” [Online]. Available: https://www.techopedia.com/definition/30632/web-services-middleware. [Accessed: 25-Feb-2018].

[30] M. A. Razzaque, M. Milojevic-Jevric, A. Palade, and S. Cla, “Middleware for internet of things: A survey,” *IEEE Internet Things J.*, vol. 3, no. 1, pp. 70–95, 2016.

[31] “Server-Side JavaScript | Programmatic Marketing Content | Salesforce Developers.” [Online]. Available: https://developer.salesforce.com/docs/atlas.en-us.noversion.mc-programmatic-content.meta/mc-programmatic-content/ssjs_serverSideJavaScript.htm. [Accessed: 22-Feb-2020].

[32] “30.1.2. Middleware - AFRCRM - Confluence.” [Online]. Available: https://confluence.jumia.com/display/AFRCRM/30.1.2.+Middleware. [Accessed: 22-Feb-2020].

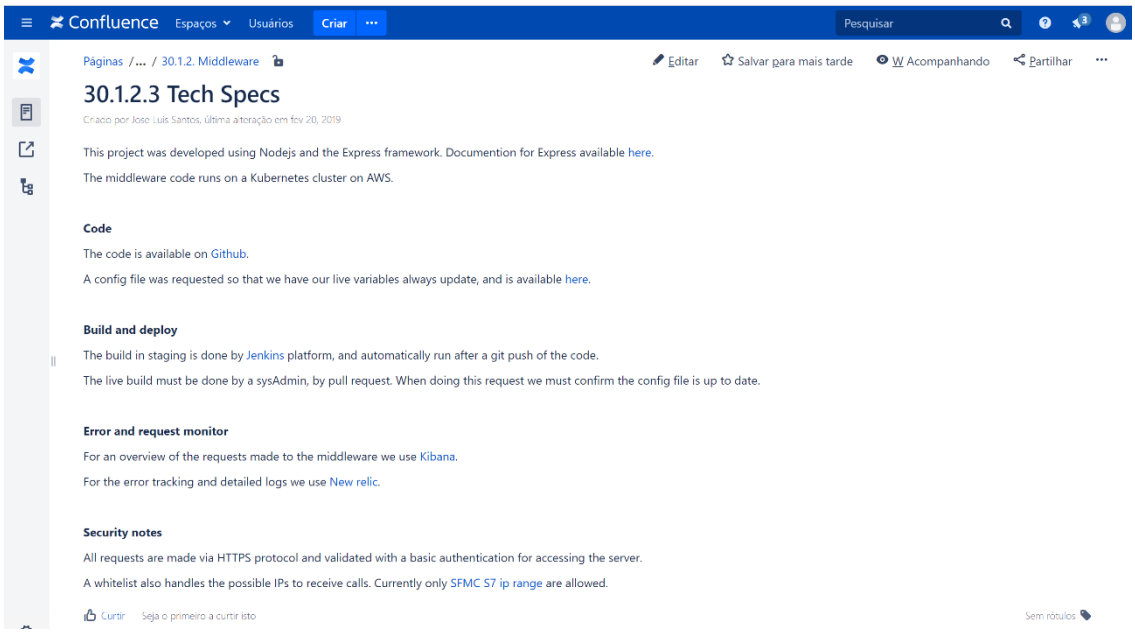
[33] “Node JS.” [Online]. Available: https://nodejs.org/.

[34] “Express4.17.1 Fast, unopinionated, minimalist web framework for Node.js.” [Online]. Available: https://expressjs.com/.

[35] “Basic routing.” [Online]. Available: https://expressjs.com/en/starter/basic-routing.html.

9 Appendix

Appendix A – Solution documentation, available on Jira platform



Confluence Espaços Usuários Criar ··· Pesquisar

Páginas / ... / 30.1.2. Middleware

Editar Salvar para mais tarde W Acompanhando Partilhar ···

30.1.2.3 Tech Specs

Criado por Jose Luis Santos, última alteração em fev 20, 2019

This project was developed using Nodejs and the Express framework. Documention for Express available here.

The middleware code runs on a Kubernetes cluster on AWS.

Code

The code is available on Github.

A config file was requested so that we have our live variables always update, and is available here.

Build and deploy

The build in staging is done by Jenkins platform, and automatically run after a git push of the code.

The live build must be done by a sysAdmin, by pull request. When doing this request we must confirm the config file is up to date.

Error and request monitor

For an overview of the requests made to the middleware we use Kibana.

For the error tracking and detailed logs we use New relic.

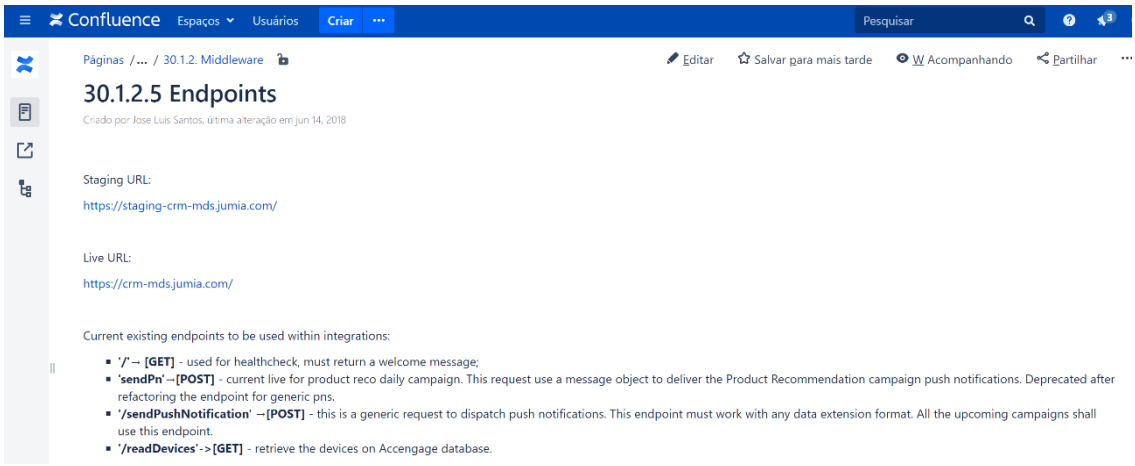
Security notes

All requests are made via HTTPS protocol and validated with a basic authentication for accessing the server.

A whitelist also handles the possible IPs to receive calls. Currently only SFMC S7 ip range are allowed.

Curtir Seja o primeiro a curtir isto

Sem rótulos



Confluence Espaços Usuários Criar ··· Pesquisar

Páginas / ... / 30.1.2. Middleware

Editar Salvar para mais tarde W Acompanhando Partilhar ···

30.1.2.5 Endpoints

Criado por Jose Luis Santos, última alteração em jun 14, 2018

Staging URL:

https://staging-crm-mds.jumia.com/

Live URL:

https://crm-mds.jumia.com/

Current existing endpoints to be used within integrations:

- **'/'→ [GET]** - used for healthcheck, must return a welcome message;
- **'sendPn'→[POST]** - current live for product reco daily campaign. This request use a message object to deliver the Product Recommendation campaign push notifications. Deprecated after refactoring the endpoint for generic pns.
- **'/sendPushNotification' →[POST]** - this is a generic request to dispatch push notifications. This endpoint must work with any data extension format. All the upcoming campaigns shall use this endpoint.
- **'/readDevices'->[GET]** - retrieve the devices on Accengage database.

Appendix B – Qualitative Evaluation Framework

q	D	Qi	Dimension	Q_j	W_{ij} (Factor Weight *j* in Dim *i*) [0,1]	Factor	rw_{jk} (requirement weight *k* in Factor *j*) {2, 4, 6, 8, 10}	Requirement	wf_k % requirement fulfillment *k*) [0,1
98%	0,06	93,5142	Functionality	95,1724	0,53	Functional (Refering Use Cases)	10	FF01 - Connect to infobib	100
							10	FF02 - Connect to Accengage	100
							10	FF03 - Send SMS	100
							10	FF04 - Send push notification messages	100
							6	FF05 - Send push notification campaigns	100
							8	FF06 -Send segments to accengage	100
							2	FF07 - Read data from Accengage	60
							2	FF08 - Send lists to infobip	0
				85,3333	0,27	User Interaction	6	FUI01 - Provide insightfull synchronous messages	100
							10	FUI02 - Provide insightfull logging	80
							8	FUI03 - Provide complete documentation	70
							6	FUI04 - Logging on data extensions	100
				100	0,20	Security	10	FCQ01 -HTTPS protocol on endpoint	100
							10	FCQ02 - Server authentication	100
							10	FCQ03 - Limit access to whitelisted ips	100
		100	Autonomy	100	1,00	Abstraction	8	AV01 - send methods read from configuration file	100
							8	AV02 - authorization parameters reading from a configuration file	100