HADRIEN HEURTEL

BUILDING A MOBILE AD EXCHANGE: HOW REAL TIME BIDDING IS REVOLUTIONIZING THE MOBILE ADVERTISING

Trabalho de Formatura apresentado à Escola Politécnica da Universidade de São Paulo para a obtenção do diploma de Engenheiro de Produção.

São Paulo
2015

HADRIEN HEURTEL

BUILDING A MOBILE AD EXCHANGE: HOW REAL TIME BIDDING IS REVOLUTIONIZING THE MOBILE ADVERTISING

Trabalho de Formatura apresentado à Escola Politécnica da Universidade de São Paulo para a obtenção do diploma de Engenheiro de Produção.

Orientador: Prof Livre-docente André Leme Fleury

São Paulo
2015

FICHA BIBLIOGRAFICA

Heurtel, Hadrien
Building a Mobile Ad Exchange: how Real Time Bidding is revolutionizing the Mobile Advertising / H. Heurtel. -- São Paulo, 2015

Trabalho de Formatura - Escola Politecnica da Universidade de São Paulo. Departamento de Engenharia de Produção.

1. Marketing digital 2. Tecnologias 3. Start-up 4. Celulares I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Produção II. t.

Acknowledgments

I would like to thank my professor André Leme Fleury for the support during the realization of this work.

I also would like to thank the École Centrale de Nantes (France) and the Escola Politécnica da Universidade de São Paulo (Brazil), where I had the opportunity to study during the last five years.

Thanks to Olivier, François-Xavier and Patoche for their friendship.

Resumo

Este trabalho apresenta a transformação tecnológica de uma agencia de propaganda digital para móvel (Mobile Ad Network) em uma plataforma aberta de livre troca de propagandas digitais (Mobile Ad Exchange). Esse trabalho integra-se na estratégia da empresa onde ele foi realizado de adaptar-se à revolução da indústria da propaganda digital lidada pelo "Real Time Bidding."

Palavras-chave: marketing digital, tecnologia, propaganda, celulares, real time bidding

Abstract

This study presents the technological transformation of a Brazilian Mobile Ad Network into a Mobile Ad Exchange. The present work aims to provide strategic insights to the company in its will to adapt to the fast changing digital advertising industry and to enter the Real Time Bidding revolution.

Key words: digital marketing, mobile advertising, ad network, ad exchange, real time bidding, technology

FIGURES

TABLES

ABBREVIATIONS

DSP - Demand Side Platform

SSP - Supply Side Platform

SOA - Service Oriented Architecture

UML - Unified Modelling Language

SDK - Software Development Kit

TABLE OF CONTENTS

1. Introduction

1.1. Context

The company, as a Mobile Advertising Network, works to establish a connecting bridge between advertisers - who are looking for advertising space to buy - and publishers - who are the owners of mobile applications and website that are looking for selling advertising space on their medias.

With the Internet and computer technologies booming, the digital marketing and digital advertisements have radically changed during the last ten years. In this new and modern world are emerging huge technology-driven platforms that, on the demand side, gather advertisers (Demand Side Platforms or DSP), and on the supply side, gather publishers (Supply Side Platforms or SSP). The advertisements themselves also changed. If ten years ago the digital advertising space was sold as to incrust links, banners and pop-ups, today the most diverse forms of contents are being auctioned, even in the mobile advertising world, from the full screen banner to the video and rich media.

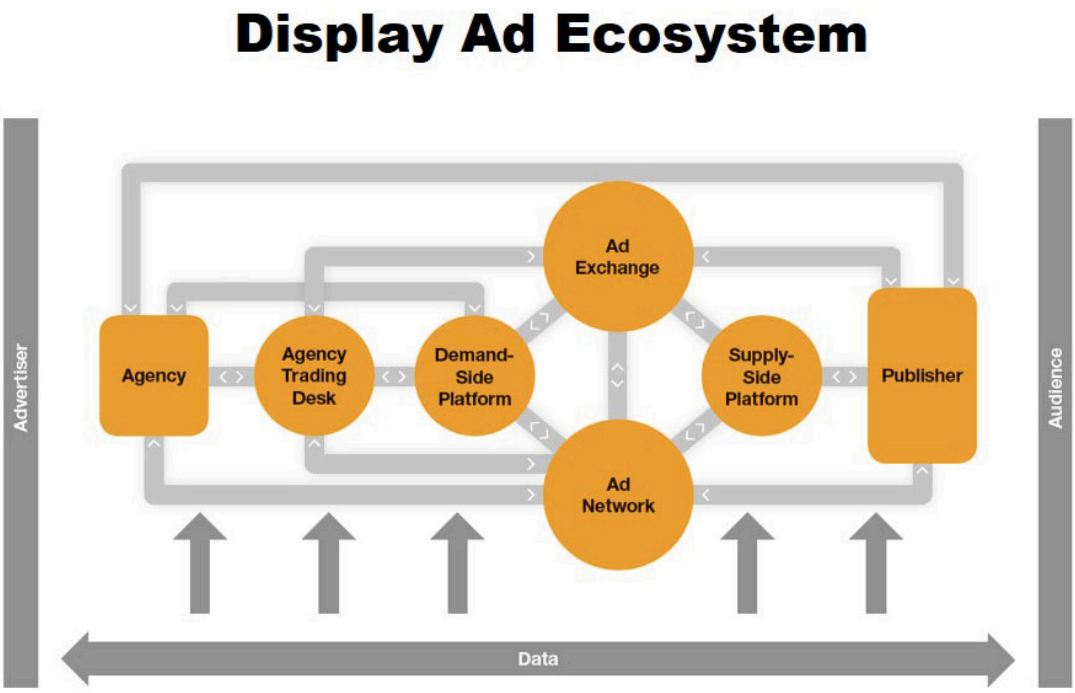

Figure 1 - The Digital Advertising Ecosystem.

Source: CIAOBASTA, 2013.

The Figure 1 shows how intricate is the digital advertising ecosystem nowadays. The left side portrays the demand, represented by the Advertisers and the different intermediates, including digital agencies, trading desks and DSPs, that help them to access audience and to buy digital traffic. The right side portrays the supply side. The Audience generates the traffic that the publishers – owners of websites and mobile applications - sell to the SSP. Right at the intersection, connecting the demand to the supply, are the Ad Exchanges and Ad Networks.

The way digital advertising space is being sold and bought was also revolutionized. Traditionally the mobile advertising space - or impressions - was being sold at fixed negotiated prices. It was the role of the Advertising Agencies to negotiate the contracts directly with the publishers to create some sort of exclusive partnerships and have the opportunity to buy their traffic at fixed rates. As we can understand, those negotiation processes were long and fastidious making the overall selling process inefficient. Over the last years, new technologies emerged and are now being used to increase the efficiency of the selling process. Among all those new tools, one really led to a technologic breakdown: the Real Time Bidding (RTB) technology. Indeed, RTB makes it now possible for the Demand Side Platforms to buy the impressions in real time, each single impression being negotiated programmatically as it is becoming available to the end user mobile device. The overall process of RTB can be illustrated with the following example. An individual opens a gaming application on his smartphone and this application has been programmed to show an advertisement at its opening. At the same instant an ad request is sent from the smartphone to the SSP that registers this gaming application. The ad request will contain some strategic information about the user, its mobile device and the application being used. The SSP instantly forwards the ad request to an Ad Exchange that will auction the impression among all of its DSPs. The winning DSP will have the right to show an advertisement and an ad response is sent back from the winning DSP to the Ad Exchange which forwards it back to the SSP which forwards it back to the cell phone that will display the advertiser's marketing campaign. Modern technologies allow this process to happen in split second.

1.2. The company

The company was born in 2012 from the passion for technology of its founder and will to play in the "tech" field. Between 2007 and 2010, the company's founder was building some casual mobile applications, mainly gaming applications. One of those casual games, Ant Smasher, became a global success with millions of downloads and daily users around the world. The company's founder, at that time, was monetizing his mobile apps thanks to a mobile ad network from which he had installed the Software Development Kit (SDK), little piece of software that allow the mobile app to receive ads. When he really realized the proportion of user reach he had gained thanks to his apps, he started to think of creating his own mobile ad network: the idea of the company was born.

In 2012, the company's founder gathered a team of young engineers and developers to create the server of the future ad network. After a few weeks of hard working, they started to look for advertisers interested in reaching their users by buying their related mobile traffic. The company started to grow integrating major ad platforms, selling them the traffic of its founder's mobile apps.

It quickly appeared to the company's first team that if they wanted to sustain their growth they would have to increase their traffic, diversify their app portfolio and find new publishers. The team focused on building a high performance Software Development Kit (SDK) that could easily spread and be installed by any mobile app developer around the world. They also focused on the company's sales and marketing strategy that would be responsible for the company getting known among the mobile app developer community.

1.3. Problem

At the beginning of 2015, the company is a successful Mobile Ad Network with a dozen of advertisement partners and daily ad volume of sixty millions impressions. The company is selling mobile traffic to advertisers at fix rate, regarding special criteria such as the geo location of the traffic, the sites or apps it comes from etc. In this model, blocks of impressions are negotiated between the company's sales people and the advertisers. The business is strong

and stable but to stay on top in the "tech world", companies have to adapt themselves continuously to the exponential growth of technologies. This is why after having spent two years as a regular Ad Network player, we took the decision to get the company into the technologic revolution of Real Time Bidding by making it a true Mobile Ad Exchange. An Ad Exchange offers to the advertisers the possibility to trade and buy available impressions individually. An Exchange is a trading platform for digital advertisement where advertisers compete in real time for individual digital impressions.

In this journey, the company will have to face new technologic and business challenges since new strategic partners (DSPs and SSPs) have to be found and new processes (Real Time Bidding) have to be implemented. The goals of the company Exchange are:

- For the DSPs: bring high quality, highly targeted inventory
- For the SSPs: bring the highest paying demand to buy their inventory

During the development of this Exchange platform, some challenges are being faced:

Business Development:

The company has to find the best Demand Side and Supply Side Partners. Indeed, in the highly competitive mobile advertising world, it is all about volume. The more volume of traffic an Ad Exchange has to offer, the more interesting it becomes for Demand Side Platforms in term of user reach. The more Demand Side Platforms an Ad Exchange connects to, the more competitive it becomes as a trading platform thus allowing increasing financial gains. In order to become a true player as an Ad Exchange, the company has to connect to the major sources of demand and inventory supply. This is possible only if the company manages to understand perfectly the business models of the Demand and Supply Side Platforms in order to offer them a high performance exchange service.

In order to provide a high quality service as a Mobile Ad Exchange, we need to provide accurate and relevant information to the buyers on the product they are willing to pay for. For an Exchange, the buyers are the DSPs and the product is mobile traffic. Thus we need to understand what information about an available impression on a mobile device really

matters for a DSP to bid on it. The more information we provide, the higher the bids because the competition increases.

Technical Development:

Going real-time and becoming a high performance Ad Exchange requires from the company to fill some important technological gaps. The main challenge is to understand, implement and efficiently use high performing programmatic technologies that are essential to process the extremely high quantity of data an Ad Exchange deals constantly with.

The company also has to fulfil the technical challenge of building the network infrastructure capable of handling hundreds of requests and responses per second between dozens of Demand and Supply Side Platforms. Once this network is set up, we need to understand how we can access and provide the information on the mobile traffic that is required by the DSPs to bid.

Once the network is built and the information about the traffic required by the DSPs is defined, we need to specifically design the processes and tasks that the future Ad Exchange will achieve. In other words, how the Ad Exchange will work.

1.4. Objectives

In this study, we focused on how a Mobile Ad Network - that sells mobile traffic at fixed negotiated price - can become a true Mobile Ad Exchange.

A Mobile Ad Exchange can be seen as an open platform where mobile traffic is being traded in real time between major advertising platforms called Demand Side Platforms. The Mobile Ad Exchange is exactly where the demand -advertisers looking for mobile traffic to buy- meets the supply -publishers looking for monetizing their traffic.

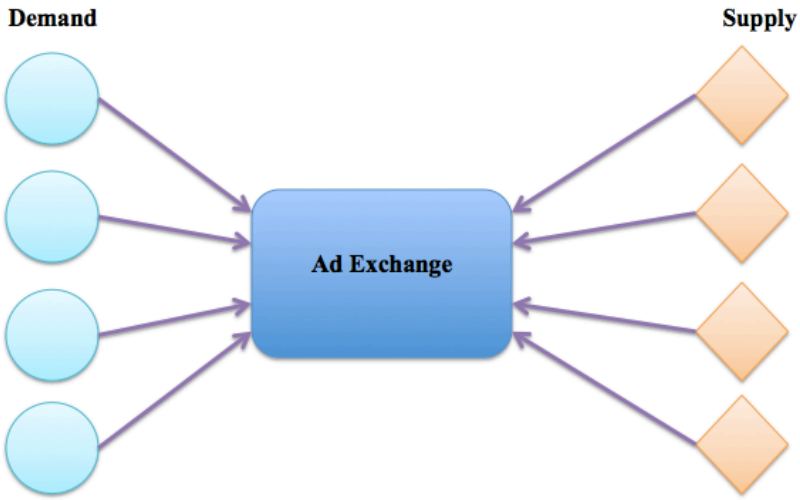

Figure 2 - The Mobile Ad Exchange as the place where demand and supply meets. Source: the author.

In order to build an efficient Mobile Ad Exchange, we first need to understand how it integrates itself into the complex landscape of modern mobile advertising. Therefore, the first objective of the present work is to define how the future Mobile Ad Exchange will connect to its partners, particularly on the demand side. This objective can be split into two specific objectives:

- How the Exchange will be connected to the DSPs on the functional aspect: what specific information do they need about the traffic in order to buy it?
- How the Exchange will be connected to the DSPs on the technical aspect: what are the technical characteristics of the network that supports the communication between the Exchange and the DSPs?

Once we defined the functional and technical aspects of the communication between the Mobile Ad Exchange and its Demand Side Partners, we focused on drawing the architecture of the Exchange itself. We did this by defining exactly the overall process of auctioning and selling a single impression on the Exchange platform. Again, this objective can be split into two specific objectives:

- Map the overall service of the Mobile Ad Exchange,
- Draw the task flows of each identified service in the previous part.

1.5. Justification

The results of this work are critical for the building of a competitive and reliable Mobile Ad Exchange. This study sets the conditions required for building a network capable of handling the high density of communication between the Exchange and its Demand Side Platforms. It also puts on the table what information on the mobile traffic really matters for a DSP when it comes to buying huge quantity of available mobile impressions.

In a second time, this study focused on the company's functional and technical strategy for defining how the Mobile Ad Exchange will work. The content of this study clearly draws the framework that will determine all the building steps of the future Exchange.

1.6. Document Structure

This work is structured as follows: the first chapter introduces some theoretical concepts in a literature review; the second chapters presents the methodology that the author followed during the realization of this work; the third chapter exposes the results of the study, as to say, how the Ad Exchange was built.

2. Literature Review

In the first part of this literature review we explain the main technical and business concepts that have been used to realize the study. Today, digital advertisement, as the core of

every digital marketing campaign, represents a complex science at the cutting edge between marketing and information technologies. The following bibliographic review is divided into four singular but nonetheless intricate parts. First, we briefly review the concept of Marketing and its evolution to Digital Advertising, in order to focus on the different marketing strategies that apply to our modern and connected world. In a second part, we investigate the historical and technological aspects of the concepts lying behind the Data Warehouse and Big Data notions. Indeed, digital and mobile advertising technologies rely on processing huge quantity of data to improve the performance of the overall marketing campaigns. In a third part, we gather insights concerning the core subject of the study by analyzing the today's Mobile Advertising Landscape. We describe the different parties that act in this sophisticate industry and visualize its main work and taskflows. Finally, in the Communication Protocols part, we take a deep look at the technologies that are used to build the complex network upon which rely every ad transaction.

All the concepts that are exposed in this theoric part of the study are based on the works of world known professionals and academics, specialized in their respective fields.

2.1. Marketing

2.1.1. Marketing and Traditional Advertising

Marketing, on a general spectrum, can be defined as the science and art of exploring, creating, and delivering value to satisfy the needs of a target market at a profit. Marketing identifies unfulfilled needs and desires. It defines, measures and quantifies the size of the identified market and the profit potential. It pinpoints which segments the company is capable of serving best and it designs and promotes the appropriate products and services (KOTLER, 1999).

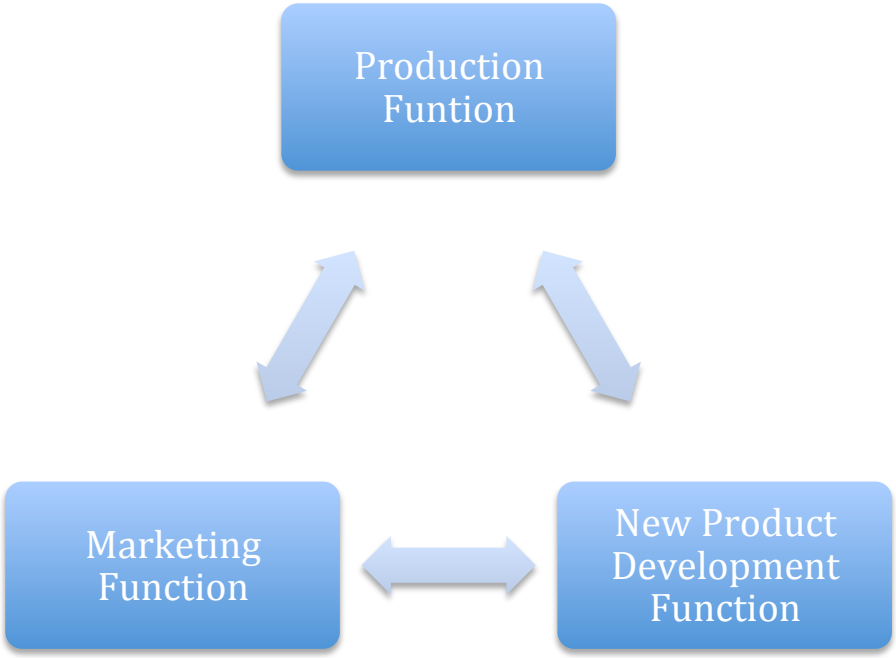

Figure 3 - The core functions of the organization.
Source: Slack et al., 2007

Marketing is one of the three essential functions of an organization, along with the Production Function and Product Development Function as shown in the Figure 3. It is the function responsible for bringing to the market the products and services of the organization with the aim of creating demand. It is responsible of the organization's administrators to maintain unity and integrity amongst all of the organization's departments so that everybody is working together, focusing its efforts to reach a common goal (SLACK ET AL., 2007).

2.1.2. Web Marketing and Online Advertising

The Online Marketing strategy of a company can be described as a set of marketing tools and marketing strategies that can be used on the Internet (GRAPPONE; COUZIN, 2006).

The Online Marketing business and area have been thriving during the last years as a result of the Internet's exponential growth. As the Online Marketing developed, it brought to

the market new assets and radically changed the relationship between companies and consumers on the one hand, and the way consumers behave on the other hand. The target audience evolved to become not only a demographic group but, thanks to the search tools, actually everyone who is looking for a specific need. (GRAPPONE; COUZIN, 2006). And it is the company's duty to ensure this need is easy for the potential consumer to be found (FOX, 2010).

Online Marketing also presents distinct characteristics. In Traditional Marketing, one of the things that have to be considered by the marketers is the effect of the excess of information and advertisement on the individuals' minds. Due to the extremely high density of advertisement and information a single person is exposed to everyday, it is absolutely common that a single advertisement, even useful for the consumer, might get lost in the information ocean (FOX, 2010). In order to compete, companies are using artifacts every time more showy with the hope that it will retain the consumer attention. If this excess of advertisements is also extremely present in the online world, a good use can be done of the Internet by making easier for the user to find what he is really looking for. If done bad, the use of the Internet as a marketing strategy can lead to prejudicial loss of money for the company and can generate frustration amongst the consumers for not finding what they are looking for (FOX, 2010).

The Online Marketing differentiates itself from the Traditional Marketing since the Internet is in its essence a totally different media from TV or radio. A common approach of marketing campaigns using the TV and radio medias would be to emphasize a product or service's relevancy, create a need into the consumer's mind, or increase the brand's appealing. On the Internet, this approach has to be reviewed since the consumer using the Internet already knows what he is looking for. The problem that has to be solved by the marketers is how to understand the consumer's needs and most of all how to bring him the adequate solution (FOX, 2010). Building an effective campaign can be done by identifying a consumer's need and creating an advertisement that effectively presents a response to it. This has shown to be far more efficient than simply drowning the consumer with irrelevant advertisements that will be ignored. Even if it is possible to create needs using the Traditional

Marketing approach on the Internet, the direct contact with the consumer that exists on the Internet make it more valuable for the marketers to think of new strategies.

The raise of search engines and social medias on the Internet drastically reduced the average time to find and access the information relating to the needs and desires of the targeting audience of the companies (FOX, 2010). This phenomenon allows the companies to respond more competitively to the emergence of new market tendencies and to improve the overall quality of their products and services. Thanks to the Internet, it has also become possible for the companies to access strategic information related to their audience targeting without the cost of organizing test groups and field researches (FOX, 2010).

2.1.2.1. The Free Content Model

With the expansion of the Internet, we saw the emergence of a new form of distributing contents. The companies started to provide free services to their consumers and develop new strategies to make profit out of it. Anderson (2009) describes the following business plans of the Free Content Distribution:

2.1.2.1.1. Usual Model

Figure 4 - Usual Model of Selling Products and Services.
Source: Anderson, 2009

The figure 4 represents the most classical model of selling goods and services in which the seller is providing a value, meets some of the consumer's needs and in return the buyer pays the seller (ANDERSON, 2009).

2.1.2.1.2. The Three Parties Model

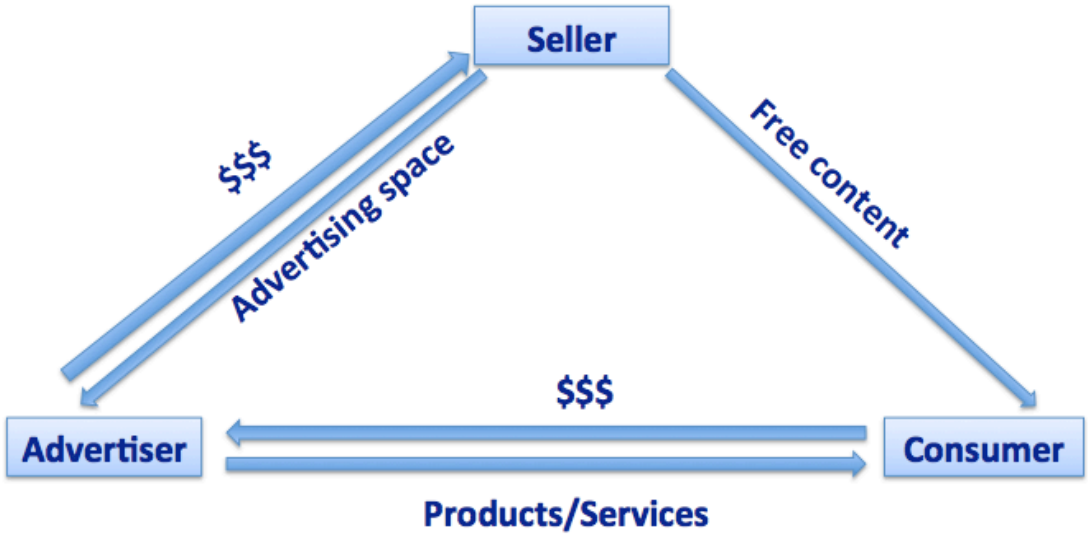

Figure 5 - Three Parties Model (media selling).
Source: Anderson 2009

This model is including a new party, the advertiser. In this case, the provider provides a free content to the consumer. The profit of the provider comes from the advertising space that is being sold in parallel of the content that is being provided for free. The advertiser is paying the provider according to his hope that the consumer will see his advertisement and eventually pay for the products and services that he is providing. In this model, now extremely common on the Internet, the provider is becoming part of the promoting campaign of the product/service that the advertiser is offering (ANDERSON, 2009).

2.1.2.1.3. The Freemium Model

Figure 6 - The Freemium Model.

Source: Anderson 2009

The Freemium Model as presented in the Figure 6 has become a quite natural model on the Internet. In this model, the organization provides for free a product or a service that includes the main basic functionalities and will be enjoyed by most of its consumers. The company manages to make profit out of the minor part of its consumers that accepts to pay for a richer version of the product or the service. The Internet makes it possible for some industries (the software industry for instance) to offer their products and services to wide range of consumers with negligible costs of distribution and production and to make profit out of the selling of a Premium version to a group of specific clients (ANDERSON, 2009).

2.1.2.1.4. The Non Profit Market

Figure 7 - The Non Profit Market.

Source: Anderson 2009

This model is used by organizations or individuals that instead of making profit by offering a value to the consumer are seeking the attention and/or the acknowledgment of a specific audience (ANDERSON, 2009).

2.1.3. The Mobile Marketing and Advertising

Mobile marketing is marketing on or with a mobile device, such as a smart phone (HEIKKI; MATTI, 2005). Mobile marketing can provide customers with time and location sensitive, personalized information that promotes goods, services and ideas. In a more theoretical manner, academic Andreas Kaplan defines mobile marketing as any marketing activity conducted through a ubiquitous network to which consumers are constantly connected using a personal mobile device (KAPLAN, 2012).

2.1.3.1. Push notifications

A push notification is a small alert type message related to the installation of a mobile app. The particularity of those notifications is that they are shown to the user even if the related app is closed. Apple was the first company to introduce the push notifications in 2007 in its operational system and Android popularized them a couple of years later. Push notifications allow the mobile app developers and owners to communicate directly with the end users. They represent an efficient way to encourage the use of a mobile app but can be perceived as intrusive and can alienate the users if not used carefully (SMITH, 2014).

If the relatively high cost of development makes a marketing strategy based on push notifications quite expensive in the short term, in the long term this can become much cheaper in comparison to SMS marketing. An efficient inbound mobile marketing strategy can be developed by using wisely SMS and push notifications (SMITH, 2014).

2.1.3.2. App-based marketing

With the increasingly widespread use of smartphones, app usage has also greatly increased. Therefore, mobile marketers have increasingly taken advantage of smartphone apps as a marketing resource. This allows for direct engagement, payment, and targeted advertising (GADZOOG, 2013).

There is a lot of competition in this field as well. However, just like other services, it is not easy anymore to rule the mobile application market.

2.2. Data Warehouse and Big Data

2.2.1. Historical considerations

The concept of Data Warehouse goes back to the end of the 1980's, when the IBM research scientists Barry Devlin and Paul Murphy first used the term "business Data Warehouse" to qualify their own integrated information system (HAYES, 2002).
In their paper published in the IBM Systems Journal, Devlin and Murphy (1988) present a model of information system that is consistent and integrated with the goal of providing relevant information to the decision takers. Called "Business Data Warehouse", this system aims to focus on the final user with the accent on how to use the information extracted rather than on how to access the information.

Two years later, in 1990, W. H. Inmon published his first book entitled "Building the Data Warehouse". Since this book has been the first to be published regarding this brand new concept, W. H. Inmon is often referenced as the father of the Data Warehouse expression (BRESLIN, 2004), although this is still an open issue, discussed among scientists (DEVLIN, 2009).

The researches on the Data Warehouse concept have been motivated by the growing need for a database system that would not only keep and register operational data but also organize it efficiently and make it available and easy to be accessed.

Machado (2006) affirms that the technology lying behind the Data Warehouse concept can be considered as the natural evolution of the Direction Support Systems (DSS). Its growing use among organizations has been motivated by two important factors:

- The need for accessing strategic information
- The use of this strategic information to take decisions in competitive and ever changing market

Inmon (1996) is the first to bring out the main issues related to the presence of diverse database systems not integrated inside a single organization:

- Lack of credibility of the data: Each individual system can have its own unique data, collected in different moments and different ends of the overall system.
- Productivity issues: The lack of data unity due to the huge number of sources of different layouts and configuration increase the work for building efficient reports.
- Incapacity of transforming data into information: The database systems of the organization are isolated one from another and usually don't have a complete data history, preventing the user from realizing complex studies and analysis.

According to Kimball (2002), the information is one the most important asset of an organization. This asset is commonly stored in two distinct ways: into transactional systems and into the Data Warehouse. Kimball presents the following analogy to explain these different roles: the user of the transactional system is the one that makes the wheel rotate while the user of the Data Warehouse is the one that observes the wheel rotating.

The user of the transactional system will for instance register the clients into the database, receive the orders and register the possible complaints. This type of user usually works with one data type at a time. On the other hand, the user of the Data Warehouse will create statistics based on the weekly number of orders and will try to find out where the complaints are coming from and why the number of clients changed between two distinct periods. This type of user usually works with different data types at the same time.

It is important to understand the differences between the two systems since they aim to attend different needs and are accessed by distinct users who will prioritize the information differently. A Data Warehouse cannot be a simple copy of a transactional system.

Transactional System	**Decision Support System (Data Warehouse**
Application-oriented	Subject-oriented
Extended data	Extended and summarized data

Updated data	Data of long time periods
Responds to functional needs	Responds to decision-making needs
Static structure	Flexible structure
Data can be updated/modified	Data cannot be updated/modified

Table 1 - Transactional System versus Decision Support System.

Source: Inmon, 1996

Table 1 presents a synthesis of the main characteristics of a transactional system versus a decision support system (DSS/Data Warehouse).

2.2.2. Definition

A Data Warehouse acts like a warehouse of an organization's data, storing it in an organized and efficient way, from diverse data sources that can be both intern and extern to the organization. It allows all the other organization's systems and tools to access efficiently the collected data, providing important assets to the decision making process.

Kimball (2002) defines the expression Data Warehouse as a database in which the operational labelled data is structured in a way that it can be easily accessed and analysed.

The classical definition given by theme's pioneer, Inmon (1996), is "a Data Warehouse is a subject-oriented, integrated, time-variant, and non-volatile collection of data in a support of Management's Decision-Making Process". The fundamental purpose of a Data Warehouse is to empower the business staff with information that allows making decisions based on consolidated information. In essence, a Data Warehouse is in a continuous process of transformation as regards information and business rules; both of them must be considered at design time to assure increase robustness and flexibility of the system.

The definition made by Inmon is still the most popular and influential. Even if not perfect, Inmon's definition lists all the core functionalities that a Data Warehouse has to supply to an organization (JIANG, 2012).

In order to clarify his definition, Inmon (1996) explains each Data Warehouse characteristic:

- Subject-oriented: the Data Warehouse is subject-oriented in order to reflect the real subjects of matter of the organization. This is different from a transactional system that will be oriented on the daily transactions and operations of the organization.

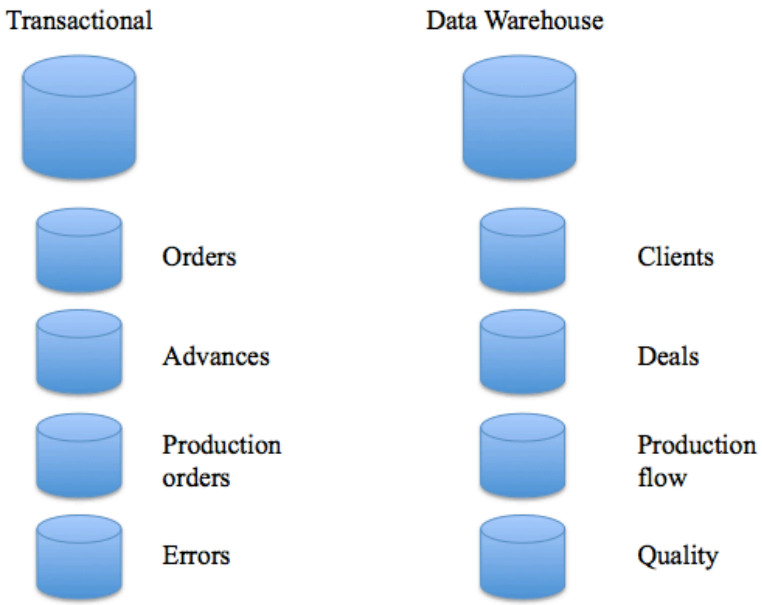

Figure 8 - Subject-oriented Characteristic.
Source: Machado, 2006

- Integrated: this the most important characteristic of a Data Warehouse. When migrating from an application-oriented environment to a the Data Warehouse environment, the data is processed in order to first, filter possible inconsistencies between the different applications and second, to obtain standardized data.

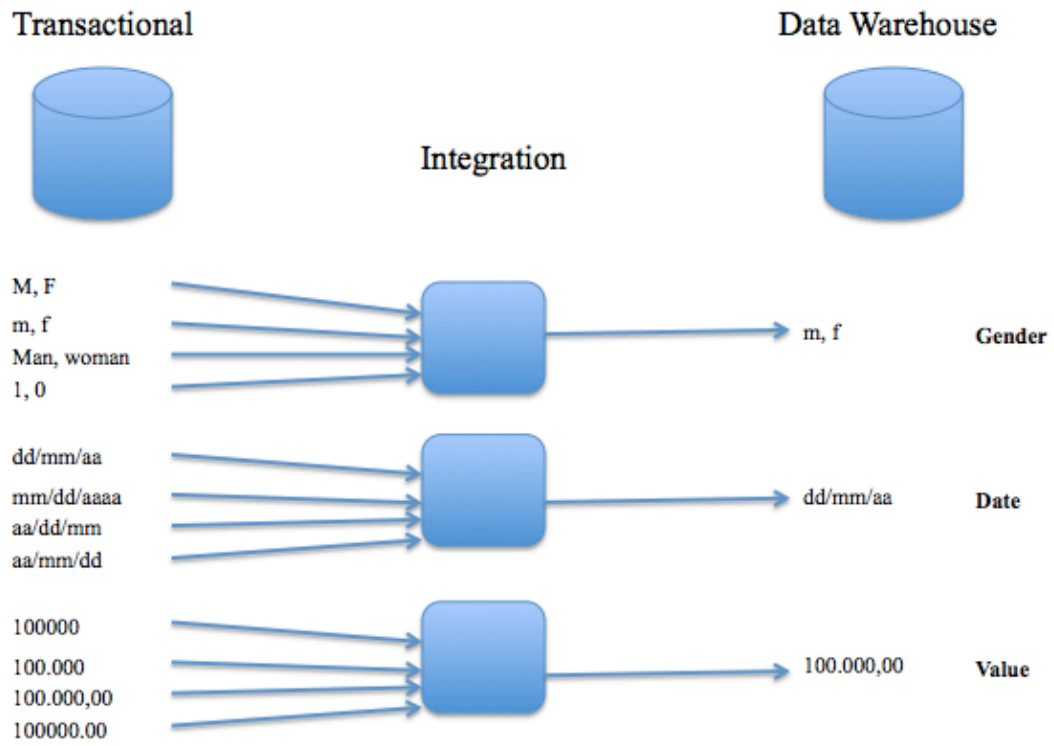

Figure 9 - Integrating data into a Data Warehouse.

Source: Kimball, 2002.

- Non-volatile: Only two different operations are allowed on the data stored in a Data Warehouse: consulting and uploading, without alteration of the existing data. In a transactional system, data is constantly being manipulated.

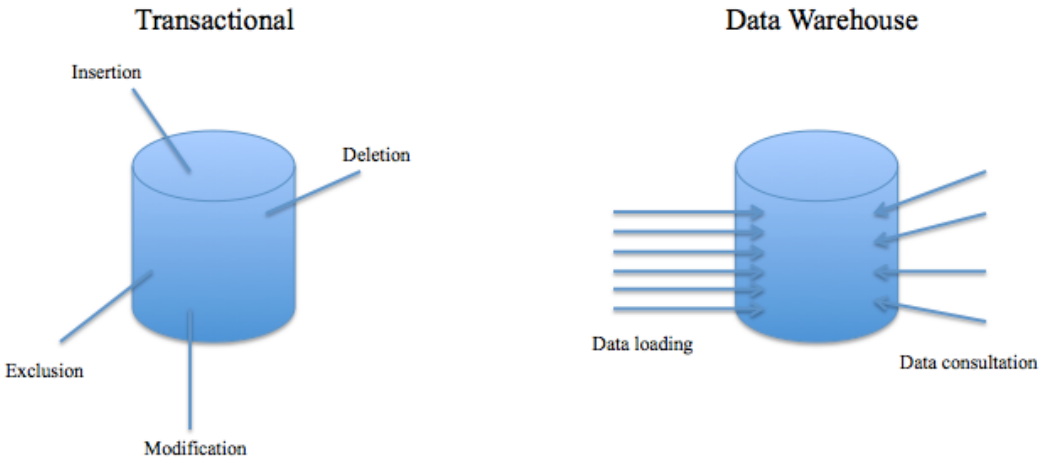

Figure 10 - Non-volatile Characteristic of a Data Warehouse.

Source: Inmon e Hackathorn, 1994

- Time-variant: It is not possible to alter the data present in a Data Warehouse and the data always refers to a certain instant or period of time. Data in a Data Warehouse can last five to ten years, when in a transactional system it usually lasts from sixty to ninety days.

2.2.3. The Goals of a Data Warehouse

According to Kimball (2002), an efficient Data Warehouse has to:

- Make information easily accessible: The data has to be understandable, meaning that it has to have intuitive meaning for the user.
- Present the organization's information consistently: The data has to be consistent, meaning for example that two different values cannot be attributed to the same piece of data.
- Be adaptive and resilient to change: Even if the data stored in a Data Warehouse cannot be modified, the changes brought to the infrastructure cannot invalidate existent data.
- Protect information: The data stored in a Data Warehouse usually refers to strategic pieces of information such as what is being sold, to whom, at what price, etc. Such data could ruin the business of the organization if held by wrong persons.
- Serve as the foundation for improved decision making: The Data Warehouse has to act like a real Decision Support System (DSS) helping improve the Decision-Making process of the organization.

To implement a successful Data Warehouse architecture in an organization it is important to deeply link the IT strategy to the business strategy that will effectively make good use of the Data Warehouse.

2.2.4. Measure the quality of the use of the Data Warehouse

Ensuring the quality of the stored data has become a critical success factor for any organization, furthermore when it comes to Data Warehouse since it is the system that will truly give the insights to make the right strategic decisions. A wrong decision made based on incorrect information can prorogue serious damages to the organization, including financial losses and the organization's withdrawal from its market.

One of the big issues that bad quality data can provoke to an organization is the intern and/or extern discredit of the organization's decisional systems. On the other hand, good data quality can become a strong competitive advantage for the company. When the company managed to have good quality data, it becomes possible to identify quickly new business or marketing opportunities from relevant data analysis and forecasts (MATTIODA, FAVARETTO, 2006).

According to Orr (1998), six generic rules can be enunciated to improve the data quality stored in information systems:

- Unused data cannot remain stored for a long time;
- The data quality of a system is a function related to the use that is being done of it, not of the collection of it;
- The data quality will improve with a more rigorous use of it;
- Issues linked to bad data quality tend to increase as the system gets older;
- The less is the probability of a data attribute change, the more traumatic it will be when it is actually going to change;
- These rules about the data quality also apply to metadata.

In addition to these six last rules, Almeida Jr (2011) enunciated the main reasons for a bad quality data stored in information systems:

- Creation of new unbound data;
- Data deterioration, data becoming obsolete;
- Wrong or irrelevant use of the data;
- High rate of change and restructuration of the data base;
- Low level of data monitoring.

Almeida Jr (2011) also underlines the idea that the quality of data can be clustered in a few representative dimensions: accuracy, availability, updates, reliability and credibility. Evaluating the Data Warehouse of an organization by taking into account these dimensions will help the organization understand and improve the quality of the stored data.

Almeida Jr (2011) affirms that the quality of the data stored in a Data Warehouse can be improved by encouraging the decision makers and the members of the business area of the organization to effectively use the data.

Spreading the same idea of Almeida Jr, Orr (1998) ensures that improving the use of the data is the only way to improve the quality of the data. It is critical to focus on defining the best use that will be done of the data in order to increase the participation of the users to the whole process. This way, the users will develop a certain sense of responsibility regarding the maintenance of the quality of the data.

2.3. Mobile Advertising Landscape

2.3.1. Mobile Advertising Network

The role of a mobile advertising network is to connect the advertisers to the mobile applications and mobile web sites owners that want to sell the advertising space of their medias. A successful ad network aggregates the maximum of advertising space from publishers and then sells this traffic to the advertisers at the best highest price. The key function of an ad network is to match the advertising space supply with the demand regarding the advertisers' targeting criteria (demographic, geographic...) (CRISTAL, 2014).

2.3.2. Exchanges

An Ad Exchange is a technology-driven platform that aims to ease the buying and selling of online advertising space. The Ad Exchange automates the whole buying and selling process by using programmatic technologies so that a marketing campaign can be implemented without any direct contact between the buyer and the sellers (CRISTAL, 2014).

The vocation of an Ad Exchange is to reduce the costs related to the traditional online advertising market.

Usually, the sellers registered on an Ad Exchange choose a minimum bid price for their advertising space available while the buyers upload their marketing campaigns in association with their bids. Afterwards the Ad Exchange compares in real time the demand and the supply in order to efficiently spread the marketing campaigns (CRISTAL, 2014).

2.3.3. Differences between an Ad Network and an Ad Exchange

Even if the key function of an Ad Network and an Ad Exchange is almost the same, as to say facilitate the buying and selling of online advertising space, there is a subtle difference. In the Ad Network model, all the different aspects of the buying and selling process are controlled by the Ad Network itself while in the Ad Exchange model, the platform is more transparent and the buyers and sellers have the control on different aspects such as the price, to whom the impression is being sold or from which media the impression is being bought (CRISTAL, 2014). The majority of Ad Networks rely on a model with three different actors for each transaction, a seller, a buyer and the network itself, but the reality of today's online advertising is far more complex than that.

The Ad Exchange model can be seen as the evolution of the Ad Network model in a way that it is a technology-driven platform upon which many actors of different kind (publishers, advertisers and Ad Networks or Ad Exchange included) can buy and sell advertising space. In other terms, this results in the fact that each ad traded can have anywhere from zero to many middlemen (CRISTAL, 2014).

2.3.4. Demand Side Platforms (DSP)

The key function of a Demand Side Platform (DSP) is to enable the advertisers to buy advertising space at scale, via Ad Exchanges, across a wide range of publishers. In this model, the Ad Exchange represents a single buying point for huge numbers of impressions globally (CRISTAL, 2014).

DSPs, just as the Ad Exchanges, were created in order to improve the traditional way of selling and buying digital advertisement, which was done by human ad buyers and salespeople. By removing humans from parts of the process, DSPs help make that costly and unreliable process cheaper and more efficient (SMITH, 2014). DSPs are part of the programmatic revolution of the online advertisement that tries to eliminate all manual actions of the buying and selling ads process.

The reason why an advertiser is buying online inventory through a DSP is that it provides the intelligence of buying targeted impressions across publishers (SMITH, 2014). This means that the DSP has access to strategic information regarding the end user such as:

- Its location
- The device that is being used
- The media that will display the ad

Publishers make their inventory available for sale on Ad Exchanges and a DSP can evaluate the quality of the inventory regarding the targeting criteria of its advertisers, deciding programmatically which of those impressions it makes the most sense for an advertiser to buy. This type of buying is called programmatic buying because prices are not being negotiated by human salespeople. The price of those impressions is determined in real time by the DSP through an automatic process known as Real Time Bidding (SMITH, 2014). It is called "Real Time" because the price evaluation is done in split second.

A mobile DSP is similar to a DSP for the mobile world. It allows advertisers to buy mobile inventory from the most appropriate mobile sites and applications for their marketing campaigns. In order to generate its buying intelligence, the mobile DSP gathers data from different sources such as publishers, advertisers and third party data provider companies. The collected data is then efficiently used in its Real Time Bidding process to give the advertisers the best opportunity to buy the right mobile inventory for their clients, thus maximizing ROI (SMITH, 2014).

Instead of buying blocks of online inventory, as it was historically done, through its Real Time Bidding intelligence, the DSP evaluate each single impression as the end user opens a mobile web page or application. If the generated impression is positively evaluated by the DSP and matches a client's campaign, it is purchased and the corresponding ad will be displayed on the mobile device (SMITH, 2014).

Through a mobile DSP, advertisers can monitor the performance of their campaigns in relation to the ad impressions they buy. They can adapt their marketing strategy regarding the performance, engagement and audience reach of their campaigns (SMITH, 2014). The most significant difference between a DSP and an Ad Network is that it is more transparent for the advertiser to buy through a DSP. Indeed, when buying inventory through an Ad Network, advertisers will not benefit from a complete monitoring control and might not know where their ads have been served or the real cost/value of the purchased inventory. Ad networks operate in an opaque way: they try to buy the inventory at the lowest price in order to sell it to advertisers at the highest (CRISTAL, 2014).

2.3.5. Real Time Bidding

Real Time Bidding is the process used by Demand Side Platforms to value and bid in real time for an advertising space impression. These impressions are being auctioned in Ad Exchanges where advertisers, DSPs and publishers gather to buy and sell online inventory (CRISTAL, 2014). An analogy can be done between the stock market and these online media marketplaces that are the Ad Exchanges. In the stock market shares become available and buyers have to compete for them, with the stock being sold to the highest bidder. Similarly, in online Ad Exchanges, an impression becomes available for sale and the advertisers and DSPs have to bid for it. At the end, the advertising space goes to the highest bidder and the process is restarted.

Real Time Bidding technology appeared in 2009 and is getting every time more popular. Selling and buying online advertising space has proved to be more effective than the traditional non-programmatic process since it is more transparent and does not involve human sales people to negotiate the prices of the impressions. The technology that lies behind Real

Time Bidding is called Programmatic Buying. This technology allows the DSPs to inspect, value and eventually bid through proprietary algorithms on a single impression as it is becoming available on the mobile user's device (SMITH, 2014). The overall buying process becomes more efficient since it allows the buyers to value impressions individually. Traditionally, blocks of impressions were negotiated at fixed prices. Through Real Time Bidding, the buyers can monitor in real time the performances of their marketing campaigns and adjust their bids in order to remain competitive. In a similar way, the sellers can monitor in real time the revenue of their inventory and adjust their asks to make more profit. Another great advantage of Real Time trading is that it improves the ad targeting and the efficiency of marketing campaigns because it uses end user's data (ANKIT, 2014).

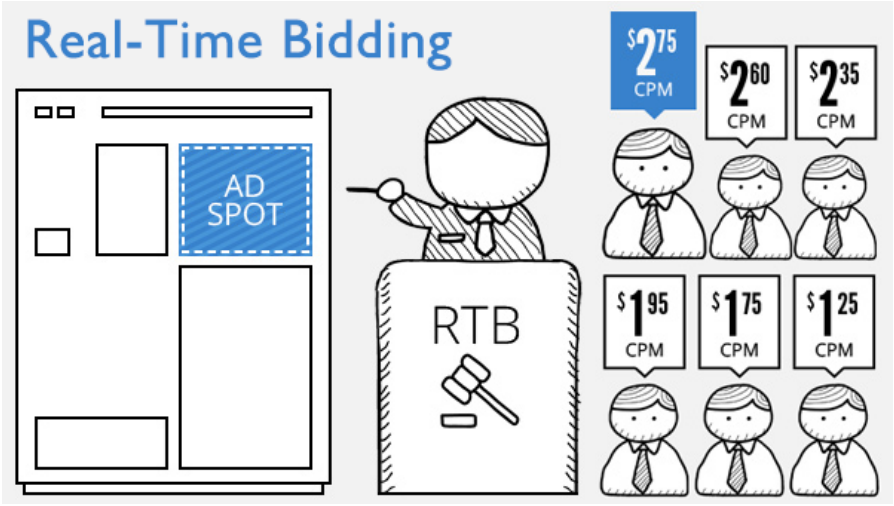

Figure 11 - Real-Time Bidding explained.

Source: Ankit, 2014

With the use of Demand Side Platforms and Ad Exchanges, the buyers are not paying in advance anymore for available advertising space and have to bid in real time for each ad impression. This process allows the advertiser to get rid of inefficient negotiation processes - usually done by human sales people - and make better use of the available technological resources (ANKIT, 2014). Also, the Ad Exchanges and DSPs manage to gather always more information about the users through demographic and behavioural data, which provides insights for the advertiser to choose the right targets for the advertisement (SMITH, 2014).

Let us illustrate the process of Real Time Bidding with a common day-to-day example. X is a user who spends a lot of time on cooking mobile applications and websites. While surfing, X arrives on a mobile application that uses RTB ads. Meanwhile, a kitchen equipment company and an editor of recipe books have both signalled interest in this kind of user. The system recognizes the compatibility between the user's profile and the requests of the advertisers and they both bid on the impression. The advertiser that places the highest bid wins and he gets to display the ad to the mobile application (ANKIT, 2014).

The protocol to build a Real Time Bidder is the following:

1. Handling the request: By the time an impression is becoming available, a call is sent from the publisher to the advertiser's bidding system with the characteristics of the impression. The advertiser's bidding system is generally the Ad Exchange that can interpret the call. In the meantime, the publisher will wait for a pre-established period of time for a response. If some advertisers do not send back a response to the bid request, the Ad Exchange will continue the auction using only the bids that have been received.
2. Building the response: When the advertiser receives a bid request from the Ad Exchange, its system will build a bid response to send back. This response needs to contain details and descriptions of the bid, such as, the bid price, the creative that will be eventually displayed, etc. The advertiser's system that is able to create and send back the response is generally handled by the DSP (ANKIT, 2014).

When compared to the traditional ways of selling, buying and displaying online advertising, Real Time Bidding presents numerous advantages. By integrating and automating the different processes, the Real Time Bidding helps improve the disjoint, manual and inefficient traditional value chain (ANKIT, 2014). Consequently, the process of running a campaign is being facilitated for both advertisers and publishers. This can greatly help to improve the efficiency of the ads since the process is accelerated and becomes less labour intensive. More efficiency means more return on investment for the parties involved in the advertising process.

Historically, advertising was done by buying large number of impressions, often not very targeted. As a consequence, it was common to see impressions from different websites and mobile applications traded and priced to the same CPM (Cost Per Mille/Cost for a thousand impressions). Those prices were subject to negotiation between sales people and the role of the market forces in the pricing was marginal. This traditional process of purchasing advertising is characterized by a lack of transparency that also affects the monitoring of the marketing campaigns' performances (ANKIT, 2014).

The advent of Real Time Bidding has allowed the publishers and the advertisers to interact directly on the online Ad Exchanges, meaning that both supply and demand forces are interacting freely to determine the price of the impression (ANKIT, 2014). The free market rules are applied and help remove the negative impact of negotiation and lack of transparency in the process. If the only forces driving the price are the market forces, both advertisers and publishers are winning: advertisers are protected of being over charged and publishers are realizing the true potential of their inventory. Moreover, since impressions are now being traded individually, the market is being fragmented which in the long run leads to equilibrium prices (ANKIT, 2014).

Furthermore, Real Time Bidding allows advertisers to improve their targeting by concentrating their advertisement budget on the impressions that really target their core market. This helps the advertisers save resources that were previously spent on untargeted impressions and can now be spent to increase their targeted bids and win more auctions. In the long term, this strategy will help reach higher conversion rates and improve the efficiency of the marketing campaigns. Also, with more transparency due to the use of Real Time Bidding, the advertisers have more control on the websites and mobile applications where their advertisements are being displayed which insures more brand safety (ANKIT, 2014).

For the supply side, Real Time Bidding can help publishers to increase the selling rate of their ads and the price at which the impressions are being sold since the advertising is better targeted. Less human labour is needed to bulk impressions, which directly lead to reducing the cost of trading inventory. Even if this cost advantage is counterbalanced by the high cost of the technology lying behind Real Time Bidding, it is expected to decrease in the

future. For the demand side, Real Time Bidding can help the advertisers access more targeted inventory which helps improve the efficiency of the marketing strategy. Therefore, even though the Cost Per Mille can be higher with Real Time Bidding, campaigns will be generating higher return on investment, increasing the profitability of the advertisers (ANKIT, 2014).

2.3.6. Ad Transaction Stream

The ad transaction starts at the same instant that a user clicks on a link to a mobile web page or opens a mobile application. An ad request is automatically built and sent to the Ad Exchange on the basis of information regarding: the impression and page environment (location of the impression, size, type of ad...), the website or application (URL, contextual category) and the user and its device (time zone, device model, behavioural and demographic data) (ANKIT, 2014).

Once the ad request is received by the Ad Exchange, it will programmatically build a corresponding bid request. This request contains all the information transmitted in the ad request and is sent by the exchange to the advertisers. All the advertisers' systems will then value the impression regarding to the information of the bid request. If the impression is positively valued, the advertiser's programmatic system places a bid on the auction and builds a bid response that is sent back to the Ad Exchange. The Ad Exchange evaluates all the bid responses and the highest bidder gets the impression, able to place its creative in it (ANKIT, 2014).

Once the Ad Exchange notified the DSP that offered the best price that it won the auction, the process of serving an ad takes place as follows:

- The advertiser sets up a campaign corresponding to the impression
- The Ad Exchange transmits the monitoring data to the advertiser (data of clicks, views...)
- The Advertisers monitors the performance of its campaign

The main specificity of this process is that everything from the ad request building, to the bid response building, with advertisers setting budgets and bids, happens in a split second. Sometimes setting a bid is a very intricate process, taking in consideration various types of data from the behavioural profile of the user to performance data of the campaign (ANKIT, 2014).

2.4. Communication Protocols

2.4.1. Concept of Server-To-Server Communication

The Server-to-Server communication model works as the Client-to-Server model with servers acting like clients. The Client-To-Server model is a computing model of a distributed application structure. This means that tasks and workloads are partitioned between the servers -which provide a resource or a service- and clients -which are requesting the corresponding service or resource (DUSTDAR; SCHREINDER, 2005).

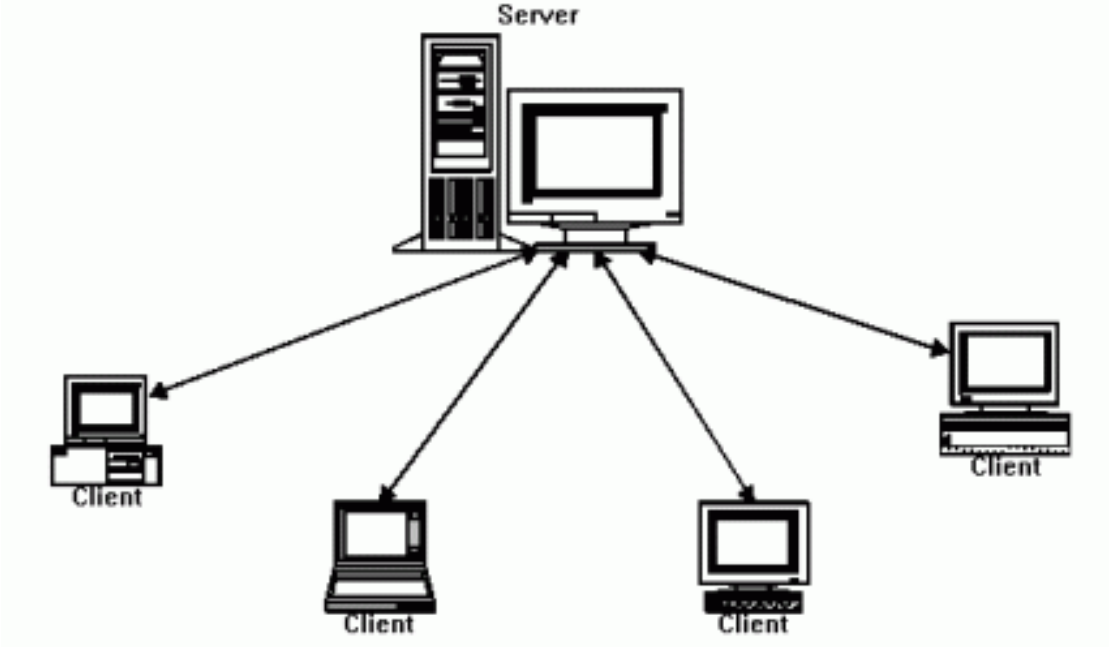

Figure 12 - Server and Requesting Clients.

Source: Harper, Douglas, 2013

In the Client-Server architecture, programs from different computers cooperate in a network. The client sends a request to the server in order to initiate the service or function that it is aiming to provide. Servers are commonly classified by the services they provide (HARPER; DOUGLAS, 2013). For example, a web server serves web pages.

A server and its clients communicate on top of a simple request-response messaging template: the client sends a request to the server, which returns a corresponding response. This communication relies on a common language shared by the computers of the network. As a human language, computers' language has to follow specific rules and patterns, which are defined in a communication protocol (FARREL, 2004). In our model, both requests and responses will be sent on the Internet. Consequently, we will use one of the protocols of the Internet Protocol Suite.

2.4.2. The Internet Suite Major Protocols

The Internet Protocol Suite sets the protocols of communication used by computers organized in networks on the Internet. The suite provides a pattern of end-to-end connection between the computers on the network and sets the rules to packetize the data, transmit and route it to its final destination (BRADEN, 1989).

The Internet Protocols are constructed on top of four abstraction layers, each layer providing tools to build the communication on the network. From lowest to highest, the layers are the link layer, which contains the communication technologies for a single network segment (link); the internet layer, which helps connect hosts across independent networks, thus establishing internetworking; the transport layer which handles the host-to-host communication; and the application layer, providing process-to-process application data exchange (FARREL, 2004).

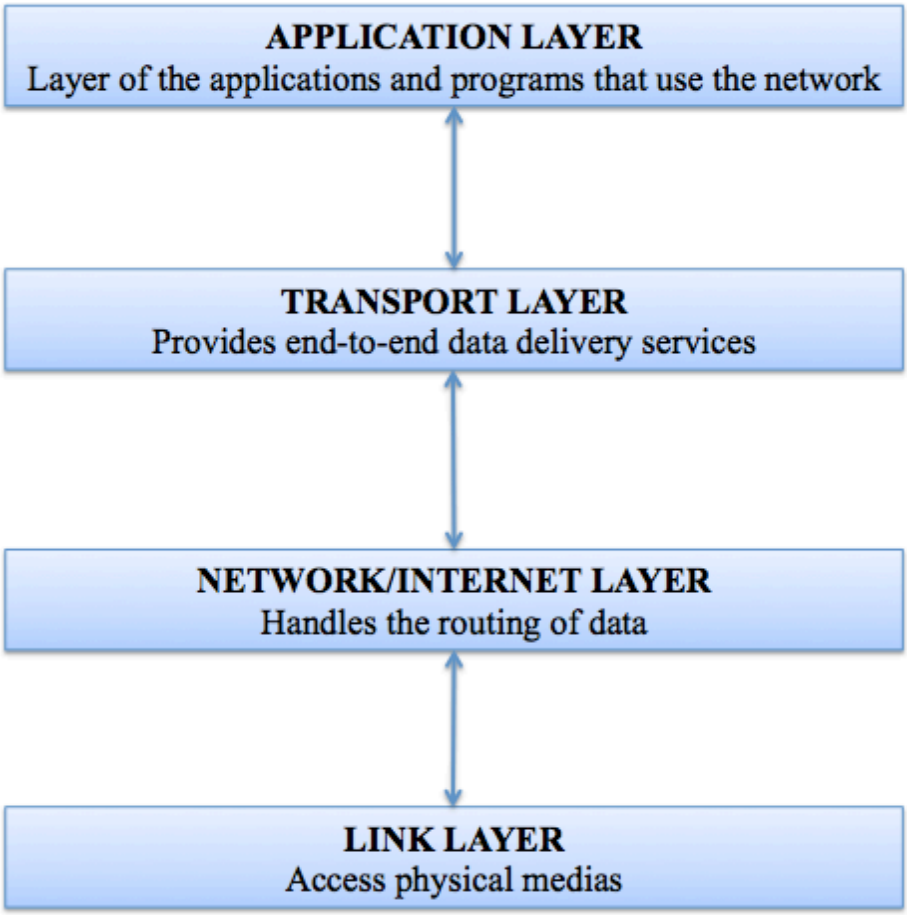

Figure 13 - The Four Layers of the Internet Protocol Suite

Source: Farrel, 2004

In order to simplify the study, we will focus on the two upper layers: Application and Transport. We will rely on the standard protocols used by our communication client for the other layers.

2.4.3. At the Application Layer

The Application Layer lies on top of the Internet Protocol Suite. It aims to set the application protocols and how the host programs should interface and interact with the services provided by the Transport Layer - which set the rules on how to use the network (FARREL, 2014).

Among the protocols working at the Application Layer are:

1. Hypertext Transfer Protocol (HTTP)
2. Simple Mail Transfer Protocol (SMTP)
3. Dynamic Host Configuration Protocol (DHCP)
4. Domain Name System (DNS)
5. Simple Network Management Protocol (SNMP)
6. File Transfer Protocol (FTP)
7. Trivial File Transfer Protocol (TFTP)

The role of the protocols at the Application Layer is to set the interfaces and consequently they are not concerned with the network infrastructure devices. The role of the protocols at the other layers is to format, packetize and transmit the data on the network and consequently interact with the network hardware (FARREL, 2014).

2.4.3.1. Hyper Text Transfer Protocol (HTTP)

The Hypertext Transfer Protocol (HTTP) sets the pattern to implement the connection between a client and a web server. HTTP is mainly responsible for the information and the data transmitted on the World Wide Web. The port number 80 of a computer has been reserved for the HTTP connections. Among the most known HTTP servers, we can name the Apache Web Server or the Internet Information Server (IIS). HTTP clients are called Web Browser some of them known as Mozilla Firefox, Google Chrome, etc. In the HTTP model, a Web Browser requires a file or web page and if available, the server will respond with the appropriate message. A big asset of the HTTP model is that its messages are English-based and flexible (FARREL, 2014).

2.4.3.2. File Transfer Protocol (FTP)

The File Transfer Protocol (FTP) is a standard end-to-end protocol that helps transfer a computer file on a network of computers, such as Internet. FTP includes specific controls of the data connections between the client and the server. Users must follow an authenticating

procedure by following a sign-in protocol, which is often presented as a username and password combination. The server can also be configured to allow anonymous connections (FARREL, 2014).

2.4.3.3. Dynamic Host Configuration Protocol (DHCP)

The Dynamic Host Configuration Protocol (DHCP) allows automatically assign new parameters of configuration a computer on the network. Some of the parameters that can be modified are:

- The IP address
- The Subnet Mask
- The DNS Server etc.

When a device on the network gets its configuration parameters by using DHCP, it is called a DHCP client. Those DHCP clients communicate and exchange information with a DHCP server in order to obtain the related configuration information (FARREL, 2014).

2.4.3.4. Simple Mail Transfer Protocol (SMTP)

The Simple Mail Transfer Protocol (SMTP) has been created to set a standard of the transmission of e-mails on the Internet. The port number 25 of a computer has been reserved for SMTP connections. It is interesting to note that if the SMTP is used by most of the electronic mail servers to send and receive email messages, at a user level, the application of the client usually uses the SMTP only for sending messages to the mail server. The client application will preferentially use either the POP3 or the IMAP protocols for receiving messages (FARREL, 2014). The SMTP is also used by the main proprietary mailing systems -such as Gmail or Hotmail- to send and receive emails from outside their own systems even if most of them use their own non-standard protocols to access their own mail box accounts.

2.4.4. At the Transport Layer

The Transport layer includes two major protocols, Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). The TCP and the UDP protocols at the Transport Layer help in data communication as explained below:

1. Create an interface to help the network applications access the physical network (FARREL, 2014).
2. Provide means to help the devices on the network accept data from different applications and direct that data to the recipient application on the receiving device (Multiplexing). Similarly, the receiving device has to redirect the data to the correct application (De-multiplexing) (FARREL, 2014).
3. Error checking, flow control, and verification (FARREL, 2014).

2.4.4.1. Transmission Control Protocol (TCP)

The Transmission Control Protocol (TCP) has the characteristic to ensure that the order of the information that is being transmitted and therefore is considered as a reliable protocol. The data - from the Application Layer - to be sent is first broken into TCP segments and then reassembled at the receiving side. This property is essential to preserve the order of the information since it is not sure that the data reaching at the receiving device is in the same order as it was at the sending device because of problems in network or different paths packets can take (FARREL, 2014).

Main characteristics of TCP are:

- Stream Data transfer: The information exchanged by the applications at the Application Layer is transmitted to the bottom layers as a contiguous stream of bytes. The role of the TCP protocol is to pack this byte streams into what is called TCP segments, and then to pass them to the IP layer for transmission to the destination computer (FARREL, 2014).
- Reliability: Is the most important aspect of the TCP protocol. TCP provides reliable data delivery by recovering from data that has been accidentally damaged, altered or lost by the Network Layer. The recovery of the data is possible because the TCP protocols assigns a sequence number to each byte that will be transmitted and expects a positive acknowledgment from the receiving TCP layer. If this acknowledgment is

not received within a determined interval, the byte is retransmitted. On the receiving side, the TCP protocol uses the same sequence numbers to rearrange the TCP segments and reorder the information (FARREL, 2014).

- Flow control: The TCP protocol helps fix the issues related to the different data rates at which the computer can operate on the network. Because of factors like the available bandwidth or the use of the Central Processing Unit (CPU), it can happen that a sending computer sends the data at a much faster rate than the receiver computer can handle. The TCP protocol at the receiving computer uses the same acknowledgement used to assure reliability to inform the sending computer of the number of bytes it can receive without causing serious problems to its internal buffers (FARREL, 2014).
- Multiplexing: Multitasking achieved through the use of port numbers (FARREL, 2014).
- Connections: TCP is a Connection Oriented Protocol. This characteristic reflects the fact that before any can be sent, the computers must establish a connection between some of their respective port numbers (FARREL, 2014).

2.4.4.2. User Datagram Protocol (UDP)

Unlike the TCP Protocol, the User Datagram Protocol (UDP) is an unreliable and Connection Less protocol of the Transport Layer. The UDP does not perform all the complex functionalities of the TCP because it is a much simpler protocol. The UDP protocol only serves as a multiplexer and demultiplexer using port numbers and provides no flow control or error recovery (FARREL, 2014).

Main characteristics of UDP are:

- Connectionless: UDP protocol does not check the connection between the receiving and the sending devices before sending the data. UDP is not a reliable protocol when compared with TCP (FARREL, 2014).
- Error Checking: UDP protocol is capable of performing fundamental error checking. The UDP header includes a checksum field that the destination device can use to check the integrity of the data (FARREL, 2014).

- Data Sequencing: UDP protocol does not provide sequencing of data. This means that the data may arrive at the destination device in a different order from which it was sent. This may happen in large networks, such as the Internet, where the Datagrams might take different paths to the destination and experience delays in different routers (FARREL, 2014).

Description of a UDP Header:

- Source Port Number: The first 16 bits of the UDP header are defined by the port number of the application that is sending the data (FARREL, 2014).
- Destination Port Number: The following 16 bits are defined by the port number of the application that is receiving the data (FARREL, 2014).
- Length: The next 16 bits identify how long the datagram is in bits (FARREL, 2014).
- Checksum: The last 16 bits of the UDP header are reserved for the error-detection mechanism, called the checksum value. The sending computer runs a mathematical algorithm on the datagram. The receiving computer will run the same mathematical algorithm on the datagram. If the both values match then it roves that the datagram wasn't damaged while its journey (FARREL, 2014).

Some well-known protocols of the Application Layer using the UDP protocol at the Transport Layer are:

- Trivial File Transfer Protocol (TFTP),
- Domain Name System (DNS),
- Remote Procedure Call (RPC) used by the Network File System (NFS),
- Simple Network Management Protocol (SNMP),
- Lightweight Directory Access Protocol (LDAP).

2.4.4.3. Differences between TCP and UDP

The TCP and the UDP are the two main protocols of the Transport Layer. Both of them have a singular way of operating and they have to be chosen wisely depending on one's requirements. The TCP, as a reliable protocol, guarantees that the data is delivered in the

same order than it has been sent. TCP features a broad error checking mechanism, which includes flow control and acknowledgment of the data. It is a connection-oriented protocol that establishes a connection between the computers before any data can be sent. The UDP operates in a datagram mode and is a connectionless protocol. It only has a basic error checking mechanism that uses checksums (FARREL, 2014).

TCP characteristics summarized:

1. TCP is a connection-oriented protocol, which means the devices first have to open a connection before any data can be transmitted and have to close the connection afterwards.
2. TCP assures reliable delivery of data.
3. TCP provides extensive error checking mechanisms such as flow control and acknowledgment of data.
4. TCP sequences the data before sending it.
5. TCP guarantees the delivery of data.
6. TCP is rather slow because of these extensive error-checking mechanisms.
7. Using port numbers, TCP allows multiplexing and demultiplexing.
8. TCP resends the data packets that are being lost or damaged.

UDP characteristics summarized:

1. UDP is a datagram-oriented protocol that does not check for an opened connection.
2. UDP is efficient for broadcast and multicast type of network transmission.
3. UDP has only the basic error-checking mechanism using checksums.
4. UDP does not sequence the data before sending it.
5. UDP does not guarantee the delivery of data.
6. UDP is faster, simpler and more efficient than TCP. However, it is less robust.
7. Using port numbers, UDP allows multiplexing and demultiplexing.
8. UDP does not resend the data packets that are being lost or damaged.

2.4.5. The Secure Protocols

Protocols at the Application Layer level offer means to secure the connection by encrypting the data. The secure versions of those protocols are commonly built upon the traditional version of the protocol, adding the security capabilities of a specified protocol (FARREL, 2014).

- HTTP Secure (HTTPS): HTTPS benefited of a wide deployment on the Internet. Technically, it is not a protocol of itself. HTTPS is the result of layering the HTTP protocol on top of the SSL or TLS protocol, thus adding the security capabilities of SSL/TLS to standard HTTP communications. The main motivation for HTTPS is to provide authentication of the requested server and prevent wiretapping and man-in-the-middle attacks.
- FTP Secure (FTPS): FTPS is an extension to the commonly used File Transfer Protocol (FTP) that adds support for the Transport Layer Security (TLS) and the Secure Sockets Layer (SSL) cryptographic protocols.
- DHCP: provides no secure means
- SMTP Secure (SMTPS): SMTPS is the result of securing SMTP with a transport layer security. It is intended to provide authentication of the communication partners, as well as data integrity and confidentiality. SMTPS is not a proprietary protocol and not an extension of SMTP. It is just a way to secure SMTP at the transport layer. This means that the client and server speak normal SMTP at the application layer, but the connection is secured by SSL or TLS. This happens when the connection is established before any mail data has been exchanged.

2.5. Unified Modeling Language (UML) Diagrams

The Unified Modelling Language is a language used in graphic modelling, which has been created to provide a normalized method for visualizing the conception of a system. The UML is often used in software engineering and during the development of object oriented systems (RUMBAUGH, JACOBSON, BOOCH, 2010).

The UML offers fourteen different diagrams listed in the following table:

Name	Brief description
Class Diagram	Represents the classes of the system
Object Diagram	Represents the instances (objects) of the classes used in the system.
Component Diagram	Represents the physical components of the system.
Deployment Diagram	Represents the physical elements and the way the components of the system are distributed among those elements.
Package Diagram	An UML package is a logical container. This diagram represents the dependencies between packages.
Composite Structure Diagram	Represents the relations between the components of a single class.
Profile Diagram	Allows for a particular field of expertise to adapt an UML meta model of reference.
Use Case Diagram	Identifies all the possible relations between an external user and the system.
State Machine Diagram	Describes the behaviour of the system. Requires that the system described is composed of a finite number of states.
Activity Diagram	Describes the behaviour of the system as a succession of activities.
Sequence Diagram	Represents the sequence of interactions between the components of the system.
Communication Diagram	Simplified representation of a Sequence Diagram. Focuses on the communication between objects.
Interaction Overview Diagram	Describes the relations between the different scenarii identified in the

	Sequence Diagram.
Timing Diagram	Describes the time variations of a variable.

Table 2 - Overview of the UML Diagrams.
Source: Rumbaugh, Jacobson, Booch, 2010

In the following sections we take a deeper look at the UML diagrams used in the present work.

2.5.1. UML Class Diagram

A Class Diagram is a UML graphic representation of the different classes and interfaces of a system, and the relations between them. It is a static diagram because it does not take into account the temporal and dynamic aspects of the system (RUMBAUGH, JACOBSON, BOOCH, 2010).

A class describes the type and particular behaviour of a group of objects. The different variations of a class are called the instances. A class is represented as a combination of functions (methods) and data (attributes). Classes are used in object-oriented programming and help to modelling a program (RUMBAUGH, JACOBSON, BOOCH, 2010).

2.5.1.1. Representation of a class

A class is represented by three slots rectangle:

- First slot represents the name of the class,
- Second slot represents the attributes of the class,
- Third slot represents the methods of the class. They are the function the class can execute.

The second and third slots represent the behaviour the class.

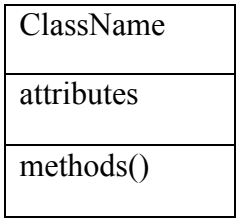

Figure 14 - Representation of an UML class. Source: Rumbaugh, Jacobson, Booch, 2010

2.5.1.2. Relations between classes

The following relations are not exclusive to the Class Diagram and are used in every static diagram.

- An Association: represents a family of links. It represents the static relationship among the object of two classes (RUMBAUGH, JACOBSON, BOOCH, 2010).
- An Aggregation: is an association that represents a part-whole or part-of relationship (RUMBAUGH, JACOBSON, BOOCH, 2010).
- A Composition: is more specific than aggregation. It means that if the container class is destroyed, every instance it contains is destroyed as well (RUMBAUGH, JACOBSON, BOOCH, 2010).
- A Generalization: indicates that one of the two related class (subclass) is considered to be a specialized form of the other (the super type) and the superclass is considered a generalization of the subclass (RUMBAUGH, JACOBSON, BOOCH, 2010).

2.5.2. UML Sequence Diagram

The Sequence Diagram chronologically represents the interactions between the components of the system. The goal of the sequence diagram is to describe how the actions take place between the different objects of the system (RUMBAUGH, JACOBSON, BOOCH, 2010).

2.6. Service Oriented Architecture (SOA)

SOA is an architecture based upon the use of services. Each of the service of the architecture realizes a small basic function such as validating inputs, looking for specific data etc (ROUSE, 2014).

We call a service a function that is well defined, self-contained and does not depend on the context or state of other services. The services communicate with each other by passing simple data. The following figure represents a simple communication pattern between two services of a SOA:

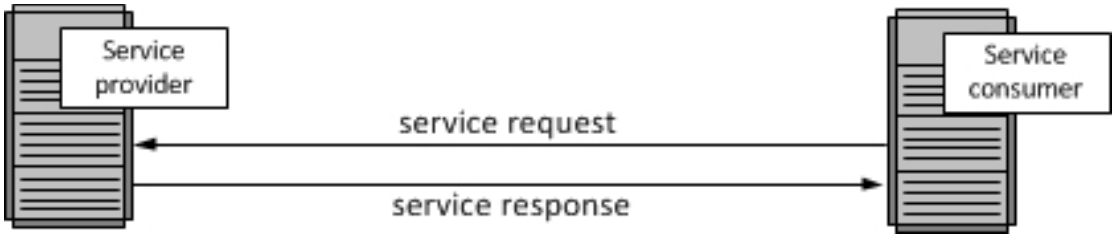

Figure 15 - Communication Patter between two Services.

Source: Barry, 2014.

At the right, a service-consumer sends a request message to a service-provider. The service-provider processes the request and returns the appropriate response to the service-consumer. In order to make the communication possible, the requests and responses must be defined so that they are understandable by both the service-consumer and service-provider. It is important to note that a service-provider can also be a service-consumer (BARRY, 2014).

In order to harmonize the communication, the entry of each service is defined by an Application Program Interface (API). An API creates a minimal interface that allowing to control the pattern of the requesting messages. A service-consumer only accesses the system of the service-provider through its API (BARRY, 2014).

3. Methodology

In this section we present the methodology that was applied all the way from the beginning to the end of the present study. By taking into consideration the defined objectives and the bibliographic study, we went through the process of developing a methodology of study that allows the realization of a quality, structured and logically organized work.

The study is organized in two major parts, each one presenting two sub parts and a partial conclusion. In the first part, we focused on understanding the bidding model of a Demand Side Partner and the requirements related to the building of a reliable network between an Ad Exchange and its Demand Side Partners. This helped to build efficient and sustainable relations between the company Ad Exchange and its Demand Side Partners. In a second part, we drew the Service Oriented Architecture of the company Ad Exchange and studied each identified service by building its class diagram.

The following figure illustrates the structure of the work that has been described in the chapter. First, the author focused on the relation between the Ad Exchange and the DSPs. The information that a DSP needs to know in order to bid on an impression is explained, and the characteristics of the network are detailed. Second, the author described the structure of the Ad Exchange itself by drawing its Service Oriented Architecture and the class diagram of each identified service of the architecture.

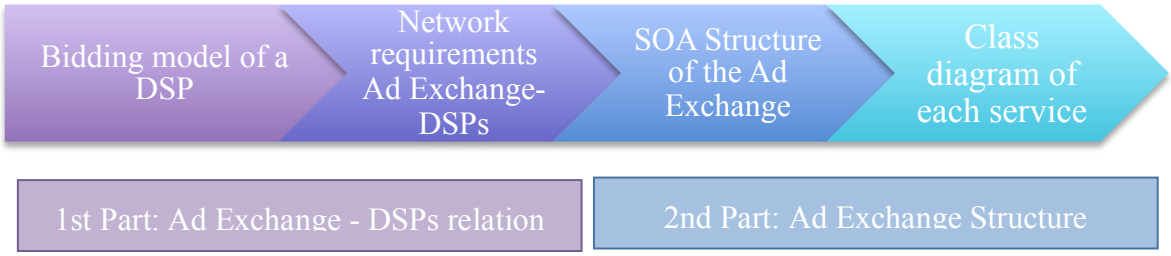

Figure 16 - The Methodology Chronogram.

Source: the author

3.1. Understand the relationship between an Ad Exchange and the Demand Side Platforms

In this section, we understood what is the information required by a Demand Side Platform to place a bid on an available impression. We will also identified the best way of setting a communication pattern between our Demand Side Platform partners and the future the company Exchange.

3.1.1. The bidding process of a Demand Side Platform

The profit of the company Exchange is directly linked to its DSP partners' bids. It is critical for the company Exchange to make sure that its DSP partners are bidding on the maximum number of Bid Requests. Since a DSP places a bid based on the information present on the Bid Request, we first need to understand what information is essential for a DSP to bid.

Before the present study, the system of the company as a Mobile Ad Network was able to send to the advertisers some pieces of information about an available impression on a mobile device. By becoming a Mobile Ad Exchange, the company felt the need to check the relevancy of this data and verify if it really met the needs of the Demand Side Platforms. Based on the work of SMITH (2014), we determined all the pieces of information that can be accessed today about a mobile website or mobile application that is being used, the end user and its device.

Therefore, we built a survey to understand once and for all what information is critical, important, facultative or useless for a DSP to place a bid. The survey was sent to all the DSP partners of the company and the answers helped us understand whether the information we were already providing was relevant. It also gave us insights to improve our SDK in order to be able to pass strategic pieces of information that are missing today.

3.1.1.1. Building the survey: understand a Request-Response pattern

In order to help the DSPs evaluate the quality of an impression and bid on it, the Ad Exchange needs to pass a certain quantity of information. It appears that the information that can be extracted from a mobile device is limited in comparison to the one that can be extracted from a computer browsing a page on the Internet. Indeed, on a mobile device (smartphone, tablet…) the user's profile cannot be identified accurately since cookies are not present. Cookies are small files, which are stored on a user's computer. They are designed to hold a modest amount of data specific to a particular client and website, and can be accessed either by the web server or the client computer. This allows the server to deliver a page

tailored to a particular user, or the page itself can contain some script which is aware of the data in the cookie and so is able to carry information from one visit to the website (or related site) to the next. However, in the mobile world, some pieces of information can be passed regarding:

- The mobile website or application the user is currently browsing
- The nature of the impression that is being sold
- The mobile device that the user is currently using
- The nature of the user

The following survey was made by taking in account the technology level of the company's SDK (Software Development Kit) and a list of all information that can be possibly accessed today on a mobile device (smartphone or tablet). The company's SDK is the technology developed by the company that allows us to access this strategic information about the site/app/impression/device/end user.

The answers to this survey allowed us to evaluate the importance of each piece of information that can be extracted of the mobile website or application, the impression, the device and the end user. The Demand Side Partners of the company were asked to send the survey back as soon as possible since it was also in their interest to help us identify the required field of a complete bid request.

3.1.1.2. Building survey: its different parts

In the following paragraph, we broke down the survey into four major parts, each part gathering a specific and distinct type of information. The four parts are:

1. Information on the browsed Mobile Website or the Mobile Application that is being used
2. Information on the nature of the Impression that is available
3. Information on the Device that is being used
4. Information on the end User

The survey will also include a blank section for the DSPs to add some relevant information fields that we could have forgotten.

3.1.1.2.1. About the Mobile Website/Mobile Application

A Site object should be included into the bid request if the ad is to be displayed on a mobile website. Similarly, an App object should be included if the ad is to be displayed on a mobile application. The fields of the survey that are common to both a mobile website and a mobile application and are of interest for this research are the following:

identifier: This field makes reference to a unique identifier of the site or app where the impression is being available. This identifier is provided by the Ad Exchange.

name: This field makes reference to the site or application name as it has been defined by the publisher (cf. legal owner of the site or application). It is important to note that the publisher have the right to choose to hide the app/site name and that the exchange responsible for serving ads to this site/app must respect this privacy setting.

Domain: this field makes reference to the domain of the site or of the application (e.g. angrybirds.com).

categories: This field makes reference to the content categories of the overall site or application. Examples of common content categories are:

- Arts & Entertainment
- Automotive
- Business
- Careers
- Education
- Family & Parenting
- Health & Fitness
- Food & Drink
- Hobbies & Interests

- Home & Garden
- Law, Government & Politics
- News
- Personal Finance
- Society
- Science
- Pets
- Sports
- Style & Fashion
- Technology & Computing
- Travel
- Real Estate
- Shopping
- Religion & Spirituality
- Non-Standard Content
- Illegal Content

The only field that is specific to a mobile website is the following:

page: This field makes reference to the URL of the page where the impression is being available.

3.1.1.2.2. About the Impression

The impression object must describe the ad position or impression that is being auctioned. This piece of information can be critical for the DSPs to make sure they return a compatible ad. An impression can be of a banner, which is basically a static picture, or of a video. In both cases, the nature of the information to be passed is similar. The corresponding fields of the survey are the following:

Banner/Video

width: This field makes reference to the exact width of the impression being auctioned. A common way to measure it is using pixel unity, which means passing the exact number of pixels the width of the impression will occupy on the screen.

height: This field makes reference to the exact height of the impression being auctioned. A common way to measure it is using pixel unity, which means passing the exact number of pixels the width of the impression will occupy on the screen.

Position: This field makes reference to the exact position of the ad that will be displayed. Some common ad positions are:

- Above the fold: it is the portion of a web page that is visible in a browser window when the page first loads.
- Below the fold: it is the portion of the page that requires scrolling in order to see content.
- Header
- Footer
- Sidebar
- Full screen

mimes: This field makes reference to a whitelist of content MIME types supported. MIME stands for Multi-purpose Internet Mail Extensions. MIME types form a standard way of classifying file types on the Internet. Internet programs such as web servers and browsers all have a list of MIME types, so that they can transfer files of the same type in the same way, no matter what operating system they are working in. Popular MIME types include, but are not limited to “image/jpg”, “image/gif” and “application/x-shockwave-flash”.

3.1.1.2.3. About the Device

The device object provides information pertaining to the device including its hardware, platform, location, and carrier. In the mobile world, devices can be of a smartphone or of a tablet. The corresponding fields of the survey are the following:

ip: This field makes reference to the IPv4 address of the device.

geoLocation: This field makes reference to the geography as derived from the device's location services (e.g., cell tower triangulation, GPS) or IP address.

maker: This field makes reference to the manufacturer of the device (e.g: "Apple").

model: This field makes reference to the device model (e.g.: "iPhone").

os: This field makes reference to the device operating system (e.g.: "is").

osVersion: This field makes reference to the device operating system version (e.g.: "8.1.2").

connectionType: This field makes reference to the detected data connection type for the device. Some values for this field can be of the following:

- Ethernet
- Wifi
- Cellular data - Unknown generation
- Cellular data - 2G
- Cellular data - 3G
- Cellular data - 4G

deviceType: This field makes reference to the type of the device that is being used. Some values for this field can be of the following:

1. Mobile
2. Tablet
3. Personal computer
4. Connected TV
5. Connected device

3.1.1.2.4. About the User

The user object contains information known or derived about the human user of the device. The corresponding fields of the survey are the following:

birthdate: This field makes reference to the birthdate of the user.

gender: This field makes reference to the gender of the user.

identifier: This field makes reference to a unique consumer ID of the user on the exchange. It is important to note that the user ID is an exchange artifact and may be subject to rotation policies.

geoHome: This field refers to the home geographic position of the user. Where the users comes from.

3.1.1.3. Building the survey: the template

In this survey we asked the company's Demand Side Partners to evaluate the importance of each previously identified piece of information in their bidding process. Each field of the survey represents a piece of information that can be provided about the media, device or end user. The DSPs were asked to choose for each field if they consider it as "required", "recommended", "optional" or "not relevant" to bid on an impression.

Each object of the survey presents itself as followed:

Information field	Description	Required	Recommended	Optional	Not Relevant
<name of the field in	<description of the information passed through				

the bid request>	this field>				

In this model the DSPs were asked to tick a single degree of importance for each field of each object of the survey. Following is the "banner" object as it was presented in the survey:

Nature of the Information	**Description**	**Required**	**Recommende**	**Optional**	**Not Relevant**
w	Width of the impression in pixels.				
h	Height of the impression in pixels.				
pos	Position of the ad.				
mimes	Whitelist of content MIME types supported by the impression: “jpg”, “png”, “gif”...				

3.1.1.4. Consolidation of the results

Based on the company DSP partners' responses to this survey and in regard to the company's SDK technology we were be able to identify what information we were already able to provide and among the pieces of information we were not already able to provide, what were the critical ones and the less important ones. This analysis has been of great help for the team responsible of the company's SDK to prioritize the development of the next SDK's features.

For each field of the survey we analyzed the driver lying behind the answer of the DSPs. We tried to understand the reason why the DSPs were considering a piece of information as more important than another. This study allowed us to form hypothesis about how a Demand Side Platform is bidding against a single impression.

3.1.2. The Network Infrastructure between a Mobile Ad Exchange and Demand Side Platforms

In order to send Bid Requests to the DSPs and receive their Bid Responses, it is essential for the company to make sure it has built a reliable network structure. On the Internet network, different protocols help send and receive information, each one having its own specificities and responding to certain needs. These protocols are gathered in the Internet Protocols Suite.

Based on the work of FARREL (2004) that explains the specificities of each protocol, we built a technical survey that we sent to the company's DSP partners. The answers to this survey helped us choose the right protocols to use when exchanging information with the DSPs in order to build a reliable and efficient network that would be able to handle high-density communication traffic.

3.1.2.1. Building a survey: origin of each question

In the following section we explained the origin of each of the eight questions of the survey.

- **Question 1: Which way is the best to pass the required information in the bid request?**

All the protocols at the Application Layer of the Internet Suite have their own single way of conveying the information. The following table summarizes, for each studied protocol, its way of sending information.

Protocol	How it conveys information
File Transfer Protocol (FTP)	Information is written in a computer file that is sent on the network.
Hyper Text Transfer Protocol (HTTP)	Information is encapsulated into a

	JavaScript Object (JSON) that is appended to a POST request.
Dynamic Host Configuration Protocol (DHCP)	Information is passed in the request in the form of configuration information.

Table 3 - Ways of conveying information of the main protocols of the Internet Suite
Source: the author

The first question of this survey was presented as followed:
"Which way is the best to pass the required information in the bid request?"

In a file	In a JavaScript Object	In Configuration information	Other

- **Question 2: Which way is the best to pass the information related to the bid in the bid response?**

The exact same justification as for the first question can be applied for the second question.

The second question of this survey was presented as followed:
"Which way is the best to pass the required information in the bid response?"

In a file	In a JavaScript Object	In Configuration information	Other

- **Question 3: Does the information passed need to be encrypted?**

Most of the protocols of the Internet Suite present a secured version that offers means to secure the connection by encrypting the data being exchanged. The secure versions of those protocols are commonly built upon the traditional version of the protocol, adding the security capabilities of a specified protocol. For instance, the HTTP Secure (HTTPS), FTP Secure (FTPS) and SMTP Secure (SMTPS) are the results of layering the corresponding Application Layer on top of the Secure Sockets Layer (SSL) or Transport Layer Security (TLS) protocol at the Transport Layer.

The answer to this question helped us understand the importance of the risk that could be taken in relation to an unauthorized interception of the communication between the company Exchange and its Demand Side Partners.

The third question of the survey was presented as followed:
"Does the information passed need to be encrypted?"

Yes, always	Yes, sometimes	No, never	It doesn't matter

- **Question 4: Does the connection between servers need to be persistent?**

Some of the protocols of the Internet Protocol Suite allow connections between servers and clients to be persistent. These types of connections are also called "keep-alive connections" or "connections reuse". The idea of a persistent connection is to use a single connection at the Transport Layer to send and receive multiple requests/responses at the Application Layer, as opposed to opening a new connection for every single request/response pair.

This property is not available when using every protocol at the Application Layer of the Internet Suite and can be very useful when a high density of requests/responses are being sent and received between the servers.

The fourth question of the survey was presented as followed:
"Does the connection between servers need to be persistent?"

Yes	No	Better if it is	Better if it is not	It doesn't matter

- **Question 5: How reliable needs to be the transport protocol?**

Some of the protocols at the Transport Layer of the Internet Suite protocol present the property of being reliable. It is the case for the Transport Control Protocol (TCP) and not the case for the User Diagram Protocol (UDP). A protocol is reliable on a network if it provides reliability in the delivery of data to the intended recipient. As a consequence of this property, a reliable transport protocol functions more slowly and with less scalability. In situations where speed is a priority and the loss of little data is not as important -for example because of the transitory nature of the data-, an unreliable protocol might be set up.

In the context of a Mobile Ad Exchange communicating with DSPs, speed might be an issue but the integrity of the transmitted data is essential. The answers to this question helped us prioritize speed or data integrity and decide of the use of a reliable versus an unreliable protocol.

The fifth question of the survey was presented as followed:
"How reliable needs to be the transport protocol (from 5 "reliable", to 1 "not reliable")?

5	4	3	2	1

- **Question 6: How complete needs to be the error checking messages?**

Storing and transmitting data both involve the actions of physical entities in the real world: electrons, photons, atoms, molecules, wires, contacts and more. This means there's always some degree of uncertainty because background noise is ever present in our physical

universe and might alter or corrupt any given data bit. The protocols at the Transport Layer of the Internet Suite present different means to check the possible errors that can occurs during the transmission of data on the network. Some of the protocols' errors checking mechanisms are really complete when others only return basic information.

The answers to this question helped understand the importance of the error checking of the protocol that will be used in the communication between the company Exchange and its Demand Side Partners. Since the protocols of the Internet Suite present different approaches of the problem, it helped us choosing the most adapted one.

The sixth question of the survey was presented as followed:
"How complete needs to be the error checking messages (from 5 "as complete as possible" to 1 "only basic checking")?

5	4	3	2	1

- **Question 7: Does the transmitted data need to be sequenced?**

Some protocols at the Transport Layer of the Internet Suite sequence the transmitted data - for instance the TCP - and others don't -for instance the UDP. When a protocol does not sequence the transmitted data, it means that the data may arrive at the destination in a different order from which it was sent. This may happen in large networks, such as the Internet, where the data might take different paths to the destination and experience delays in different routers.

The responses to this question helped us understand the importance of sequencing the data that is sent to the DSPs.

The seventh question of the survey was presented as followed:
"Does the transmitted data need to be sequenced?"

Yes	No	Better if it is	Better if it is not	It doesn’t matter

- **Question 8: What is the importance of the transport velocity of the transmitted data?**

As we previously mentioned, the transport velocity of the data on a computer network can be affected by multiple causes, the reliability property of the transport protocol being one of them.

The responses to this question helped us understand the importance of transmitting data as fast as possible on the network when communicating with the company's Demand Side Partners.

The eighth question of the survey was presented as followed:
"What is the importance of the transport velocity of the transmitted data (from 5 "as fast as possible", to 1 "it doesn't matter")?

5	4	3	2	1

3.2. Building the Ad Exchange and related services

The results of the previous part of the work allowed us to understand clearly what strategic information needs to be passed to the partners of the future Ad Exchange and how a reliable network could be built between the company and the Demand Side Platforms. Those results set the technical and functional basis upon which the Ad Exchange will be built. This means that we are now able to focus on assembling the service architecture of the future Ad Exchange.

In the following section, we draw the lines of the future Ad Exchange in regard to the bibliography review, the answers to the survey that have been sent to the DSPs and our objective to build an efficient Service Oriented Architecture of the Ad Exchange.

3.2.1. A Service Oriented Architecture (SOA)

Considering the results of the previous surveys, the company's Exchange will be built as a Service Oriented architecture. In a Service Oriented Architecture the company is divided into functional areas, each functional area providing a specific service of the company's overall service. Once the services and their inputs and outputs are defined, they work independently one from another.

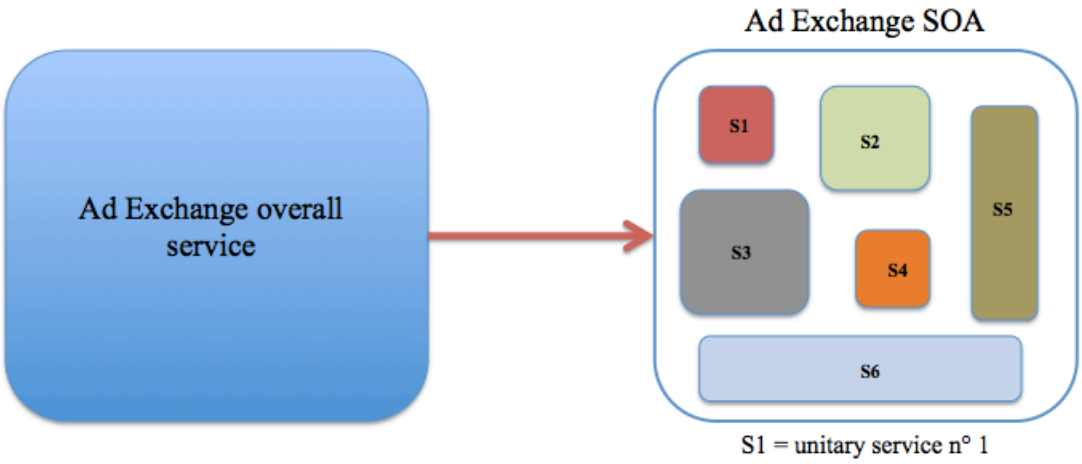

Figure 17 - Breaking the overall service of the company Exchange into Unitary Services.
Source: the author

In order to draw such an infrastructure model, we first broke the Ad Exchange's overall service into unitary services that would communicate between each other through Application Programming Interfaces (API). These APIs work as communication gates between the different services so as to facilitate, order and clarify the information flows inside the Ad Exchange.

A Service Oriented Architecture allows for a better control of the Ad Exchange processes. Once the required inputs and outputs of each service are identified, we build the services independently one from another upon the protocols that have been set up between the

services. This structure forces each team to expose the data and functionalities of their service through API interfaces, these interfaces becoming the only way for team to communicate.

To sum up, the services of the company's SOA will communicate as in a Client-To-Server model: if the service A requires an action to be done by service B, service A will require the service via the service B API and wait for it to respond. The Service B will respond with either:

- The answer expected by the Service A
- An error because:
    - The Service A request did not meet the requirements and standards Service A and Service B agreed on
    - The Service B request did not manage to process the Service A request

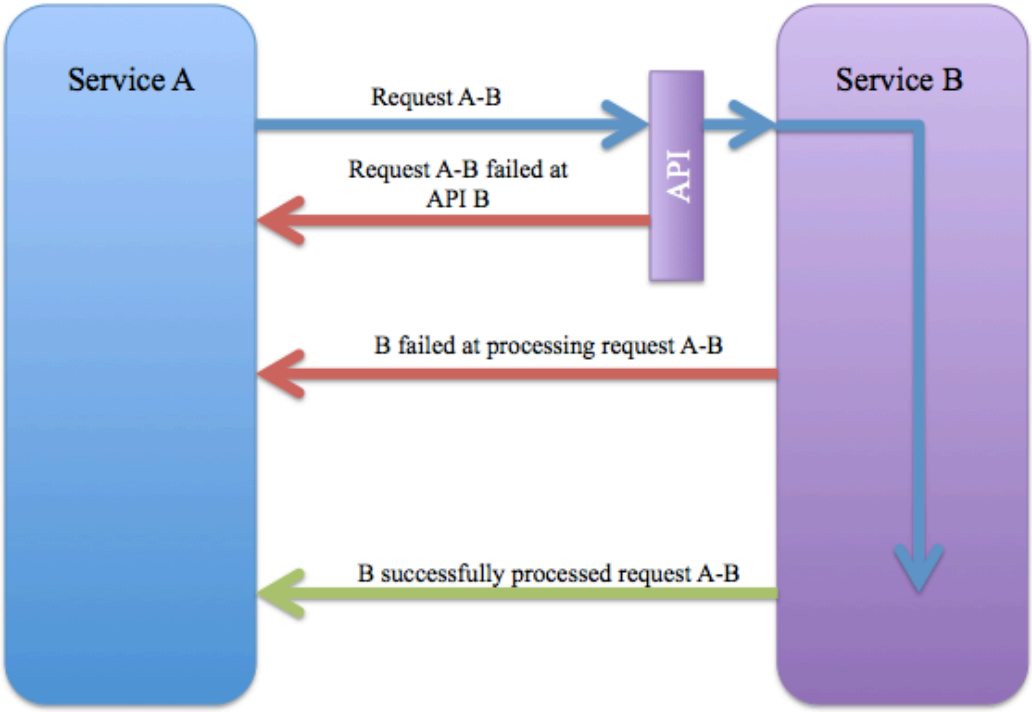

Figure 18 - Two services communicating through API.

Source: the author

We built the SOA Framework of the future company's Ad Exchange with the use of an UML (Unified Modeling Language) sequence diagram in three steps.

3.2.1.1. First Step of Building the SOA Framework

In the first step of the SOA building process, we focused on defining clearly the overall service of the company's Ad Exchange in order to break it into unitary services. We drew the lines of each of those services and determined precisely what tasks they would realize at a functional level.

As the result of this first sub-part, we were able to draw a complete and structured map of the services that constitute the Ad Exchange.

3.2.1.2. Second Step of Building the SOA Framework

In the first step of the SOA building process, we focused on defining clearly the overall service of the company's Ad Exchange in order to break it into unitary services. We drew the lines of each of those services and determined precisely what tasks they would realize at a functional level.

As the result of this first sub-part, we were able to draw a complete and structured map of the services that constitute the Ad Exchange.

3.2.1.3. Third Step of Building the SOA Framework

In the last step of the SOA building process, we defined the API that rule the communications between the unitary services of the company Ad Exchange. We analyzed the flows identified at authorized requests and responses.

As a result of this last sub part, we defined the API that will rule the communications of the different services of the Ad Exchange. Each service will communicate with the rest of

the Exchange through - and only through - its APIs. These communication gates will provide total control to the different services on what kind of requests they can process. It will allow them to work autonomously into the overall structure of the Ad Exchange.

3.2.2. Structure of the SOA services of the Ad Exchange

In the previous section of the Methodology, we explained how we broke the overall structure of the future Ad Exchange of the company into unitary services. Building a Service Oriented Architecture helped us to simplify the structure of the Ad Exchange by dividing this complex system into multiple system, each of them completing one specific task.

In the company Ad Exchange Service Oriented Architecture, each unitary service can be seen as an independent system providing outputs by completing a specific task upon the inputs provided by another unitary service. In order to clearly demonstrate the scope and tasks of the company Exchange, we defined the scope and tasks of each unitary service.

We considered all the identified services and for each of them we represented its structure with a UML class diagram. First, we defined the classes of each service, and then we built the class diagram. In those class diagrams that we built, the classes were organized in groups that share common patterns and characteristics. Each defined class was represented as a box with three different slots:

- Top slot: name of the class.
- Middle slot: attributes of the class. They are the properties of the class and symbolize its characteristics. Each of those attributes is changeable and can be set to different values.
- Bottom slot: methods of the class. They represent the operations that can be made on the class.

Once the classes were defined, we organized them into a flowchart so as to define their relationships. We portrayed those relationships -also called associations- with the use of lines and arrows as defined by the UML standards.

4. Results

The following part presents the results and analyze of the two surveys that has been conceived in the Methodology part, and explains how the Service Oriented Architecture of the Ad Exchange has been designed.

4.1. The Demand Side Platform - Ad Exchange cooperation

The following parts presents the results of the two surveys and from then, identify, first, the information that needs to be passed to the DSPs, second, the characteristics of the network that links the Ad Exchange to its partners.

4.1.1. The Request - Response pattern

The following part focuses on the first survey that aimed to characterize the information that is communicated between the Exchange and its Demand Side Partners when an impression is being auctioned.

4.1.1.1. Results of the survey

The survey was sent by email to the ten Demand Side Partners of the company Ad Exchange the first week of February 2015 and the results were collected one week later. Their answers helped us classify, by their order of importance, the pieces of information that can be extracted and sent by the company's SDK.

In this section, we present the results by following the same structure as the survey itself. For a given field, we show how many DSPs ticked each level of importance. We colored in red the levels of importance that gathered the most DSPs.

Site/App object:

Site sub-object:

Information field	**Description**	**Required**	**Recommended**	**Optional**	**Not Relevant**
id	Site ID provided by the exchange (e.g. the company).	8	2	0	0
name	Site name.	7	1	2	0
domain	Domain of the site.	5	4	1	0
page	URL of the page where the impression will be shown.	5	3	1	1
categories	Array of content categories for the overall site.	4	4	2	0

App sub-object:

Nature of the Information	Description	Required	Recommended	Optional	Not Relevant
id	App ID provided by the exchange (e.g. the company).	9	1	0	0
name	App name.	7	1	2	0
domain	Domain of the app.	5	4	1	0
categories	Array of content categories for the overall app.	5	4	1	0

Impression object:

Banner sub-object:

Nature of the Information	Description	Required	Recommended	Optional	Not Relevant
width	Width of the impression in pixels.	10	0	0	0
height	Height of the impression in pixels.	10	0	0	0
position	Position of the ad.	7	3	0	0
mimes	Whitelist of content MIME types supported by the impression: “jpg”, “png”, “gif”...	6	2	2	0

Video sub-object:

Nature of the Information	Description	Required	Recommended	Optional	Not Relevant
width	Width of the player in pixels.	10	0	0	0
height	Height of the player in pixels.	10	0	0	0
position	Position of the ad.	7	3	0	0
mimes	Whitelist of content MIME types supported by the impression: “mp4”, “3gp”, “mov”...	6	2	2	0

Device object:

Nature of the Information	Description	Required	Recommended	Optional	Not Relevant
ip	IP address of the device.	4	5	1	0
geoLocation	Geographic position of the device.	3	3	4	0
manufacturer	Device manufacturer (e.g. “Apple”).	2	4	4	0
model	Device model (e.g. “iPhone”).	2	3	5	0
os	Device operating system (e.g. “iOS”).	1	3	6	0

osVersion	Device operating system version (e.g. “3.1.2”).	1	2	7	0
connectionType	Return the detected data connection type for the device (e.g. “wifi”, “3G”).	2	3	5	0
deviceType	Return the device type being used (e.g. “cellphone”, “tablet”).	4	4	2	0

User object:

Nature of the Information	**Description**	**Required**	**Recommended**	**Optional**	**Not Relevant**
id	Unique consumer identifier of this user on the exchange.	9	1	0	0
yob	Year of birth.	2	2	6	0
gender	Gender of the user as “M” male, “F” female, “O” other.	2	2	6	0
geoHome	Home geographic position of the user (Where does the user come from)	1	1	8	0

4.1.1.2. Consolidation of the results

The first relevant observation that can be made in relation to the results of this survey is that we can easily identify the level of importance of most of the fields. We also note that none of the DSPs that were interrogated added new fields. This allows us to think that the survey was complete and that no relevant information has been forgotten during its making.

In this section, we tried to understand the result of the answers to each field of the survey. The following analyze was made by the author and some of the company members with the help of the work of SMITH (2014) - cf. the Literature Review. This analyze gives us precious hints on the bidding mechanism of a DSP, helping the company in the building of strong business relationships with its partners.

Site/App object:

"id" field:

The majority of the DSPs indicated this field as a "required" one in both cases. The unique identifier of a mobile website or application represents a critical piece of information since its value can be used easily by the DSPs as a key to identify the site or the application in their database.

"name" field:

The majority of the DSPs indicated this field as a "required" one in both cases. These answers translate the need for an identifier that is easily readable in complement to the "id" field. Indeed, the name of the mobile website or application might not be as practical as the id for buyers to use during database research but still can be really useful for manual and human reporting.

"domain" field:

The answers to this field are divided between "required" and "recommended" in both cases. Knowing the domain of a mobile website or application can be important for the DSP for blocking purpose. An advertiser can use the domain of site or an application to prevent its marketing campaigns to be displayed on the related media.

"categories" field:

As for the "domain" field, the answers here are divided between "required" and "recommended" in both cases. We understand that the content category of a mobile website or application can represent a very useful piece of information for the buyers to target their

audience and displaying more relevant ads. It also can be used for blocking purposes, in order to protect the image of the brands whose ads are being trafficked by the DSPs. Indeed, brand image is critical in marketing (digital or traditional) and brands won't accept to have their ads displayed on sites/apps that display content of certain sensitive categories.

Specific to the Site object:

"page" field:

The answers to this field are a bit divided even if half of the DSPs ticked the "required" field. Similarly to the "domain" field, this piece of information can help the DSPs in targeting better its marketing campaigns. The "page" field informs us of the exact page of the site where the impression is being available.

Impression object:

The impression object presents two sub-objects depending on if the impression is relating to a banner or a video. In both cases (banner and video), the answers of the DSPs were equally allocated.

"width" field:

All of the ten interrogated DSPs are requiring this field. This betrays the fact that an advertiser needs to know precisely the size (width and height) of the available impression in order to choose a campaign with the same size.

"height" field:

Similarly to the "w" field, all of the DSPs ticked this field as "required". This betrays the fact that an advertiser needs to know precisely the size (width and height) of the available impression in order to choose a campaign with the same size.

"position" field:

The majority of the DSPs pined the position of the impression as a required piece of information. We understand that this piece of information is important for the buyers to bid

accurately on an impression since conversion rates and user engagement are function of the position of the ad.

"mimes" field:

The majority of the DSPs indicated this field as a "required" one. This result underlines the importance of the compatibility of the formats between the impression and the ad that will be served. If the impression doe not support the format of the ad, it won't be displayed.

Device object:

"ip" field:

The answers to this field are divided between "required" and "recommended". Since the IPv4 address of a device can be easily linked to the geo localization of the device, we understand that it represents an important piece of information for the DSPs' targeting strategies.

"geoLocation" field:

The answers to this field are divided but the "optional" field gathers the most answers. This information can appear as redundant if the IP address of the device is being passed via the "ip" field but can be relevant in the following cases:

- If the geo localization information passed in this field is not calculated from the IP address of the device, it can be used additionally to determine more accurately the localization.
- If the buyers don’t have the intelligence to calculate the geo localization out of an IP address.

"manufacturer" field:

The answers to this field are mainly divided between "recommended" and "optional". Knowing the manufacturer of the device where the impression is becoming available can help the DSPs targeting better their ads.

"model" field:

Half of the DSPs pined this field as "optional". Similarly to the "manufacturer" field, knowing the model of the device where the impression is available can help the DSPs targeting better their ads. Nonetheless, we can note that the "model" field tends to be considered as a bit less important than the "make" field.

"os" field:

The majority of the DSPs indicated this field as "optional". Knowing the operational system (OS) of the device can help the advertisers target their marketing campaigns. For instance, if the marketing campaign is related to a mobile application that runs on iOS, the advertiser would not want it to be displayed on a device running Android.

"osVersion" field:

The majority of the DSPs indicated this field as "optional". Even if the version of the OS does not appear as a strategic piece of information, in addition to the "os" field, it can help the DSPs targeting specific audiences.

"connectionType" field:

Half of the DSPs pined this field as "optional". This piece of information does not appear as strategic but can help the DSPs know if the connection of the device is good enough to support relatively heavy ad formats (for instance, videos).

"deviceType" field:

The answers to this field are divided between "required" and "recommended. The type of the device that will display the ad appears to be a strategic piece of information, certainly in the building of targeting strategies.

User object:

"id" field:

Almost all the DSPs pined this field as "required". This result betrays the importance for a DSP to identify each user with a unique id. A DSP can evaluate for each individual user

the results of marketing campaigns and adapt the ads that will be served. This marketing practice is called Retargeting. Retargeting Marketing is a form of online advertising in which marketing campaign is targeted to consumers based on their previous Internet actions, in situations where these actions did not result in a sale or conversion. Storing a unique user ID allows marketers to track the frequency capping of a user on a specific mobile web page or application in order to design more relevant and specific ad campaigns.

"gender" field:

The majority of the DSPs indicated this field as "optional. This result shows that the gender of the user can help the DSPs targeting better specific audience, but since this piece of information cannot be accessed every time, it is not critical.

"yob" field:

The majority of the DSPs indicated this field as "optional". This result shows that the year-of-birth of the user can help the DSPs targeting better specific audience, but since this piece information cannot be accessed every time, it is not critical.

"geoHome" field:

The majority of the DSPs indicated this field as "optional". This result shows that knowing where the end user comes from can help the DSPs targeting better specific audience, but since this piece information cannot be accessed every time, it is not critical.

4.1.2. The Communication Protocol of the Network Infrastructure

The following part focuses on the second survey that aimed at specifying the characteristics of the network upon which will rely the communication between the Ad Exchange and its partners.

4.1.2.1. Results of the survey

The survey was sent by email to the ten Demand Side Partners of the company Ad Exchange the first week of February 2015 and the results were collected one week later. Their answers helped us identify and understand the main technical characteristics of the network that will connect the Ad Exchange to the DSPs.

In the following section, we present the results of the survey. For a given question, we show how many DSPs ticked each possible answer. We colored in red the answers that gathered the most DSPs.

Question 1: Which way is the best to pass the required information in the bid request?

In a file	In a JavaScript Object	In Configuration information	Other
2	8	0	0

Question 2: Which way is the best to pass the information related to the bid in the bid response?

In a file	In a JavaScript Object	In Configuration information	Other
2	8	0	0

Question 3: Does the information passed need to be encrypted?

Yes, always	Yes, sometimes	No, never	It doesn't matter
2	0	1	7

Question 4: Does the connection between servers need to be persistent?

Yes	No	Better if it is	Better if it is not	It doesn't matter
2	0	5	0	3

Question 5: How reliable needs to be the transport protocol (from 5 "reliable", to 1 "not reliable")?

5	4	3	2	1
6	3	1	0	0

Question 6: How complete needs to be the error checking messages (from 5 "as complete as possible", to 1 "only basic checking")?

5	4	3	2	1
1	3	6	0	0

Question 7: Does the transmitted data need to be sequenced?

Yes	No	Better if it is	Better if it is not	It doesn't matter
1	0	4	0	5

Question 8: What is the importance of the transport velocity (from 5 "as fast as possible", to 1 "it doesn't matter")?

5	4	3	2	1
7	1	2	0	0

4.1.2.2. Consolidation of the results

Similarly to the first survey, to most of the questions, the DSPs gathered in the same answers. This allowed to us to identify more accurately the right characteristics of the network that will handle the communications between the company Ad Exchange and its Demand Side Partners.

In this section we will analyze the answers of the DSPs in order to understand how the information will be transmitted and identify the protocols that will be used at the Application and at the Transport Layer of the network.

Question 1: Which way is the best to pass the required information in the bid request?

The majority of the DSPs indicated that using a JavaScript Object (JSON) is the best way to pass the information related to the bid request. A JSON has the following structure:

```
JSON = {
        field1 : value1,
        field2 : value2,
                ...
        fieldN : valueN
}
```

Using a JSON to pass the information presents the advantage to allow the company and its DSP partners to easily standardize the format of the bid requests. Each field of the JSON will correspond to a specific piece of information related to the impression that is being available. The format of the value of each field will also be standardized -string, number, array, JSON...- in order to allow the DSPs to programmatically process the requests.

As we previously saw in the bibliographic study of this work, an easy way to pass a JSON is to send it in a POST HTTP request.

Question 2: Which way is the best to pass the information related to the bid in the bid response?

The majority of the DSPs indicated that using a JavaScript Object (JSON) is the best way to pass the information related to the bid response. Since a bid response can be seen as symmetric in relation to the bid request, we understand why the answers to this question were identical to the first question.

The company Ad Exchange and its Demand Side Partners will define each field of the bid response and the type and format of its associated value. This will allow the company to process programmatically the bid responses and build its Real Time Bidding technology. This will also allow the company Ad Exchange to programmatically build the marketing campaign related to the winning bid responses before sending it back to the displaying device.

As we previously saw in the bibliographic study of this work, the DSPs can easily pass the information related to their bid response in a JSON wrapped in a POST HTTP request to the company's server.

Question 3: Does the information passed need to be encrypted?

The majority of the DSPs indicated that it does not matter whether the information is encrypted or not when transmitted on the network. The answers to this question show us that there is no necessary need for using a secure protocol on the network. Even if some personal information related to the user can be passed - year of birth, geo localization... - the overall information is not considered as sensible and doesn't require protection against possible interception.

In regard to the first three questions of the survey it appears that the communication between the company Ad Exchange and its DSPs could be set up using HTTP POST requests and responses at the Application Layer of the network.

Question 4: Does the connection between servers need to be persistent?

Half of the DSPs answered that it is better if the connections between the company Ad Exchange and their servers were persistent. This result traduces the high-density traffic that

normally exists between a Mobile Ad Exchange and its Demand Side Partners. Using keep-alive connections provides more efficiency and rapidity in the company-DSPs communication pattern. Indeed, since the same connection is used for a high number of request/response pairs, it eliminates the risk related to the possible failure of opening a new connection.

As we saw in the bibliographic revision of the present work, only a few protocols at the Application Layer of the Internet Suite allow the connections to be persistent, among which the Hyper Text Transfer Protocol.

Question 5: How reliable needs to be the transport protocol (from 5 "reliable", to 1 "not reliable")?

The majority of the DSPs indicated the protocol used at the Transport Layer of the network should be reliable. This helps us to understand that no data can be lost in the communications between an Ad Exchange and its Demand Side Partners. The DSPs and the company Exchange cannot process a broken bid request or bid response and the related impression will not be filled with a marketing campaign.

As we saw in the bibliographic study of this work, not all the protocols at the transport layer are reliable. The two major protocols of the Transport Layer are the Transmission Control Protocol - reliable - and the User Diagram Protocol - unreliable. TCP works well with HTTP allowing persistent connections while providing reliability in the data transmission. TCP at the Transport Layer and HTTP at the Application Layer appear as good choices for the network.

Question 6: How complete needs to be the error checking messages (from 5 "as complete as possible", to 1 "only basic checking")?

The majority of the DSPs responded indicating that the error checking messages should not be basic but does not need to be extremely complete. The communications between the company Ad Exchange and its Demand Side Partners are more willing because

of functional issues - broken information in the JSON, campaign that cannot be displayed on the device... - than because of technical issues on the network. This is probably why even if error checking at the Transport Layer is important but not critical for the DSPs.

As we saw in the bibliographic study, the TCP present a more detailed error checking mechanism than the UDP and appears more adapted to the needs of the DSPs in relation to the network infrastructure at the Transport Layer.

Question 7: Does the transmitted data need to be sequenced?

Half of the DSPs considered that this question was not relevant and almost the other half considered that it was better if the transmitted data was sequenced. Even if the result to this question is not teaching us a lot about the importance of sequencing the transmitted data on the network, it goes in the sense of adopting the TCP at the Transport Layer.

Question 8: What is the importance of the transport velocity (from 5 "as fast as possible", to 1 "it doesn't matter")?

The majority of the DSPs indicated that the transport velocity should be as fast as possible. This result betrays the reality of the Real Time Bidding: at the instant the impression is becoming available, the ad must be displayed as soon as possible. The overall process should occur in 400 to 800 milliseconds. Most of this time is spent at the Demand Side when the DSPs process their algorithms to value the impression and at the Exchange when it comes to pick the most profitable marketing campaign. We understand that no time should be wasted in data transmission.

The most efficient protocol in term of velocity at the Transport layer is the UDP.

As regard to the answers to the overall survey, we were able to identify the most adapted protocols of the Internet Suite to use on the network between the company Ad Exchange and its Demand Side Partners.

1. **At the Application Layer:**

The Hyper Text Transfer Protocol imposed it self as the most adapted protocol. Indeed, HTTP POST requests allow to pass the information into a JavaScript Object that can be easily standardized. Once the company Ad Exchange and its Demand Side Partners would have agreed on the standards related to both the bid request and the bid response, they will manage to programmatically process the requests, improving the efficiency of the overall process.

2. **At the Transport Layer:**

Even if the User Diagram Protocol presents the advantage of increasing the velocity of the transmission of the data on the network, it does not allow the connections between the company Ad Exchange and its Demand Side Partners to be persistent. The necessity for opening a new connection for every bid request/bid response pair increases both the latency of the overall process and the risk of failure of the transmission of the data. Moreover the UDP is unreliable and it is critical to maintain the integrity of the transmitted data so that the DSPs can bid on the available impression and the Ad Exchange can effectively display correctly the winning marketing campaign. For all those reasons, the Transmission Control Protocol imposed itself as the most adapted protocol of the Internet Suite at the Transport Layer.

4.2. Building the Service Oriented Architecture of the company Ad Exchange

In the previous part, we defined the request-response pattern of the communication between the Ad Exchange and its Demand Side Partners, and we explained upon which protocols the network has to be built. Now we focus on the service of the Ad Exchange itself and explain clearly its future structure and how it will work.

4.2.1. Mapping the services of the Mobile Ad Exchange

In the following part, we clearly explain how the Ad Exchange processes the requests and deliver the ads to the audience. We define its overall service and break it into unitary services, drawing the map of services of the company.

4.2.1.1. Defining the scope of the overall service of the company Ad Exchange

In order to build a Service Oriented Architecture of the company Ad Exchange we first need to define the overall to understand and define clearly the overall service that it will provide. Basically, the company Ad Exchange will work like this:

- First, the exchange receives a ad request from a mobile device, somewhere around the world,
- Second, the Ad Exchange formats and forward this request to all its Demand Side Partners,
- Third, at the reception of the responses from the DSPs, the Ad Exchange choose the most valuable marketing campaign,
- Finally it formats the winning campaign; sends it back to the mobile device and notifies the winning DSP

This overview of the service provided by the company Ad Exchange is represented in the following figure:

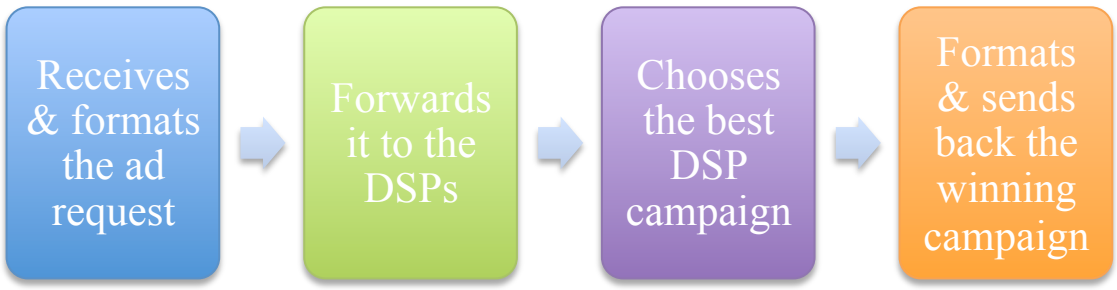

Figure 19 - Basic Process of the company Ad Exchange.

Source: the author.

The previous figure represents only aims at understanding better the over process of serving a mobile ad and each of the identified steps need to be detailed and enhanced. In the present study, we identified six basic steps that completely characterize the overall service of the company Ad Exchange. These steps helped us define the unitary services of the Service Oriented Architecture of the future Ad Exchange. Let us take a deeper look at them:

Step 1: Receive and standardize the Ad Request:

Through its partnerships with Supply Side Platforms - platforms gathering huge numbers of publishers - and its SDK, the company Ad Exchange is receiving ad requests from devices browsing mobile websites and application all around the world. The sending devices and mobile websites/application are extremely diverse and this directly leads to an incredible heterogeneity of the requests that gather at the company platform. In order to provide an efficient service, the company Exchange needs to filter the broken requests that cannot be handled and process all the other ones into a single standard format.

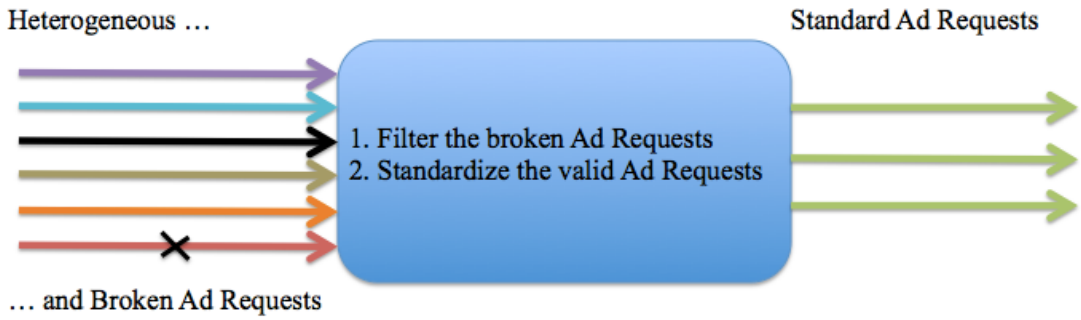

Figure 20 - First step of the overall service of the company Exchange: Receive and Standardize the Ad Request.

Source: the author

Step 2: Fraud control and tracking activities:

Once the ad request coming from a device is standardized, the Ad Exchange can process it. Before going any further, the company Ad Exchange needs to check if the request is not fraudulent. Indeed, in the mobile advertising world, a significant amount of the traffic

is dishonest and results from fake mobile websites or mobile application developed in order to simulate high-density traffic with the intention of earning money from advertising. Any suspicious ad request should be blocked at this step and should not be forwarded to DSPs, at the risk of loosing strategic demand partners.

All the requests that flow through the company system should be tracked. Tracking is important for the following activities:

1. Summarize the company activity,
2. Summarize the traffic generated by each media -mobile application or website- in order to pay the publishers the correct amount.

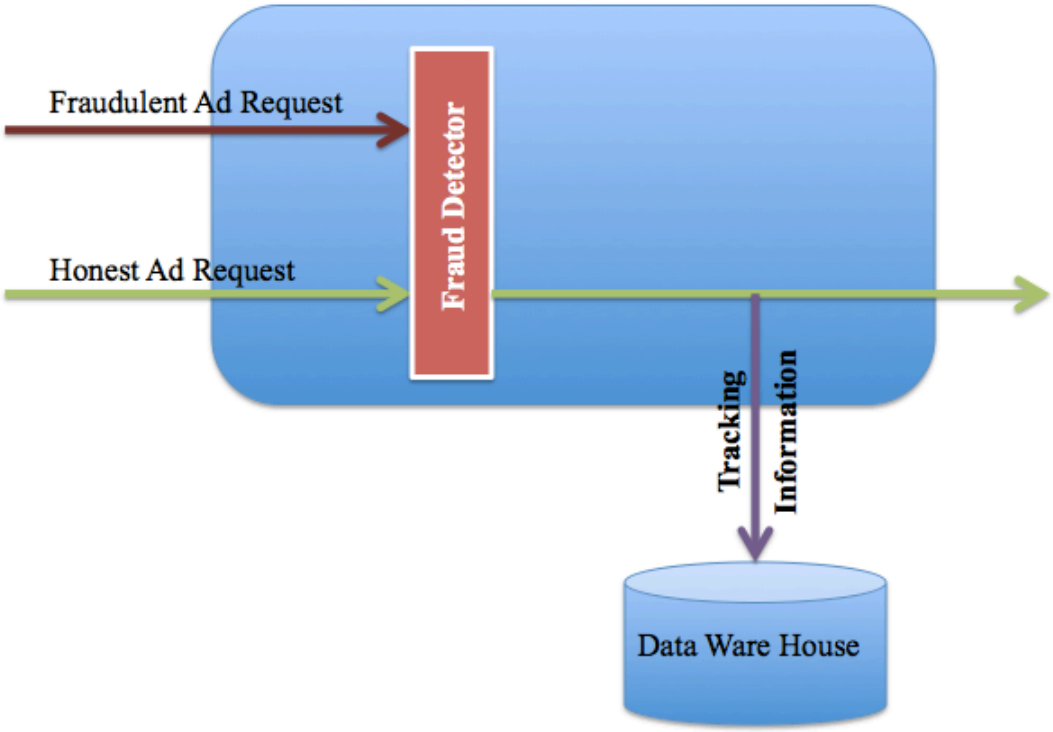

Figure 21 - Second step of the overall service of the company Exchange: Block the fraudulent Ad Requests and store information related to honest Ad Requests.
Source: the author

Step 3: Create the bid requests and call the DSPs:

Once the ad request has been flagged as "not-fraudulent", it can be forwarded to the Demand Side Partners. It is important to note that even if the DSPs asked the same information about the ad request - as seen in the first part of the study -, each of them requires this information in a specific format. Thus, it is the duty of the company Ad Exchange to adapt the ad request and create a different bid request for each of its Demand Side Partners.

When the bid requests are made, they are sent to all the DSPs and the Ad Exchange has to receive the different campaigns that the DSPs send back as bid responses.

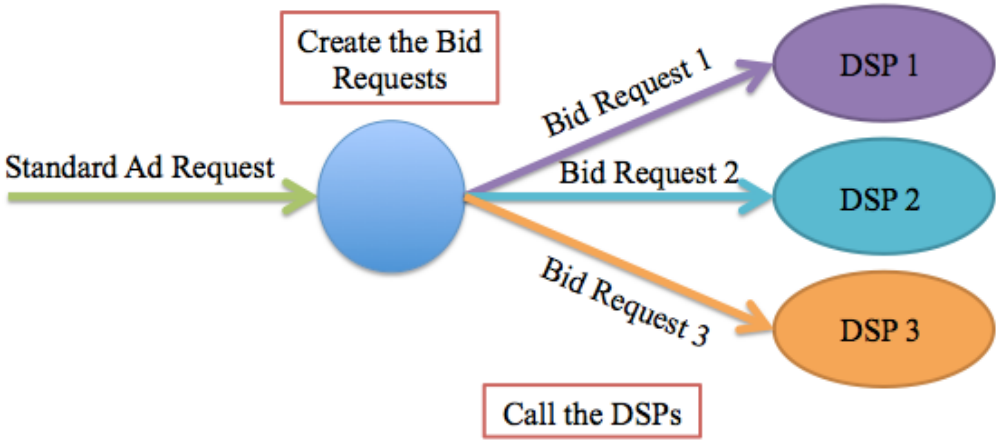

Figure 22 - Third step of the overall service of the company Exchange: Create the Bid Requests and call the DSPs.
Source: the author

Step 4: Identify and pick the most valuable campaign:

After receiving the bid responses from its Demand Side Partners, the company Exchange makes use of its Real Time Bidding technology to evaluate and pick the most suitable campaign regarding the context of the ad request.

Identify the best campaign is not as easy as picking the campaign of the highest bid because each of them might have a different paying model. Some campaigns have a CPM bid (Cost Per Mille, pays upon the impression) when others present a CPC bid (Cost Per Click, pays only if the end user clicks on the ad) or a CPI bid (Cost Per Install, pays only if the end user downloads the mobile application related to the campaign). The Real Time Bidding technology of the Ad Exchange is made of complex algorithms that are able to compare CPM, CPC and CPI campaigns and value them independently of their paying model.

Figure 23 - Fourth step of the overall service of the company Exchange: Identify and pick the best campaign out of the Bid Responses.
Source: the author

Step 5: Standardize the winning campaign:

Once the best marketing campaign has been identified, it still needs to be standardized in order to correctly display on the end device.

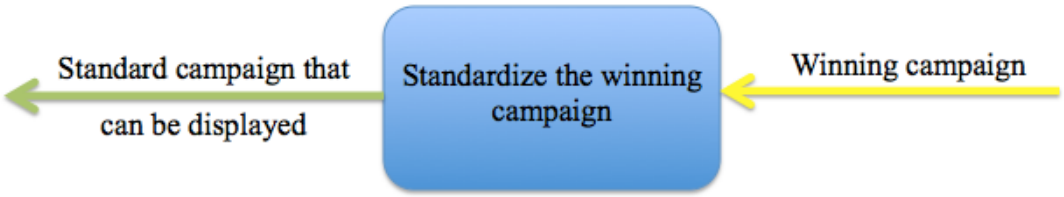

Figure 24 - Fifth step of the overall service of the company Exchange: Standardize the winning campaign.

Source: the author

Step 6: Sends the campaign back to the device and track activities:

Once the winning campaign has been standardized, it is sent back to the device where it will display. In this last step of the overall process, some activities need to be tracked, among them:

- Click and Install reports in the case of a CPC or CPI campaign,
- The winning DSP needs to be notified that its campaign is displayed.

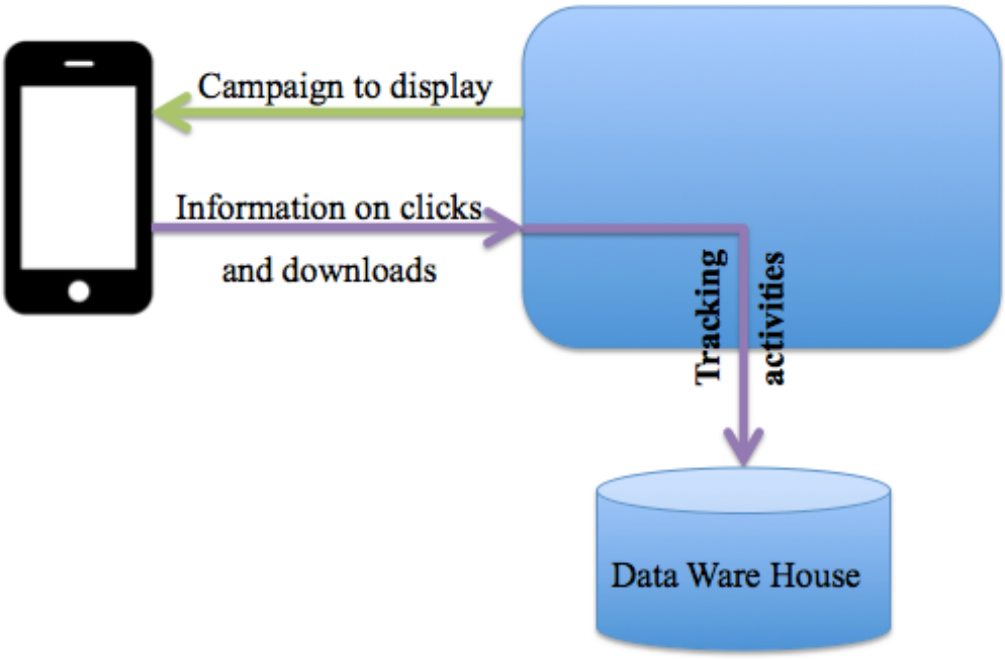

Figure 25 - Sixth step of the overall service of the company Exchange: Display the campaign on the device and track related activities.

Source: the author

4.2.1.2. Defining the unitary services of the SOA

Once the overall process of the company Ad Exchange had been characterized, we were able to define the different unitary services of the new Service Oriented Structure. Each service would be responsible for completing some of the six detailed steps. We identified four unitary services that we called: Traffic Hub, Router, DSP Hub and Recommender. The scope of each of those services is the following:

Traffic Hub: The entry and exit door of all ad requests and built campaigns:

The Traffic Hub is literally the entry and exit door of the company Ad Exchange. It is responsible for processing the following tasks:

- When an Ad Request arrives from a Supply Side Partner or the company SDK:
    - Traffic Hub evaluates the request and filters it if it is broken or incomplete (Step 1),
    - Traffic Hub standardizes the request so the Router can then process it (Step 1).
- When a marketing campaign is built and has to be sent back to the device:
    - Traffic Hub sends the campaign back for the device to display it (Step 6),
    - Traffic Hub tracks the Display, Click and Install data of the campaign for the Router to summarize.

Router: Routing the requests through the Ad Exchange and summarizing the tracking data:

The Router provides a service mainly of routing the requests through the company system and storing the data related to all the ad request/response pairs. The data stored by the Router allows the company to count the traffic related to all of its media and how much the DSPs are paying. These pieces of information are critical for the finance team to charge the DSPs and pay the publishers the correct amount. The Router is responsible for processing the following tasks:

- When a standard Ad Request arrives from the Traffic Hub:
    - Router first evaluates the request and detect possible fraud,

  - If the request is evaluated as fraudulent, it is not forwarded; otherwise Router stores data in order to track this request/response pair and forwards the request to the DSP Hub.
- When a winning campaign has to be sent back to the device:
  - Router stores data in order to track this request/response pair and forwards the campaign for the Traffic Hub to display it on the device.

DSP Hub: Communicating with the DSPs and building the winning campaign

The role of the DSP Hub is to build the bid requests upon the information related to the Ad Request, then send them to the Demand Side Partners. DSP Hub gathers the bid responses and forwards them to the recommender. The DSP Hub is responsible for the following tasks:

- When a standard Ad Request is routed from the Router:
  - DSP Hub builds one bid request per DSP upon the Ad Request and sends it to the related DSP,
  - DSP Hub gathers the bid responses of the DSPs,
  - DSP Hub forwards the bid responses and the Ad Request to the Recommender.
- When the winning campaign is picked by the Recommender:
  - DSP Hub notifies the related DSP that it won the auction,
  - DSP Hub formats the campaign so that the device can displays it correctly.

Recommender: Picking up the best campaign:

The role of the Recommender is to choose the most valuable campaign of all the campaigns that the DSPs sent back. The Recommender disposes of complex algorithms that allow it to pick up the best campaign upon data stored by the Router in the company data warehouse. Some of the data used by the recommender are about the conversion rates and previous results of the campaign, the end user who will see the campaign etc.

4.2.1.3. Defining the flows of information that link the services of the company Ad Exchange

Regarding the results obtained in the previous section, we are now able to clearly identify the way the unitary services communicate between them inside the architecture of the company Ad Exchange. The flows of information that link Traffic Hub, Router, DSP Hub and Recommender are synthesized in the following figure:

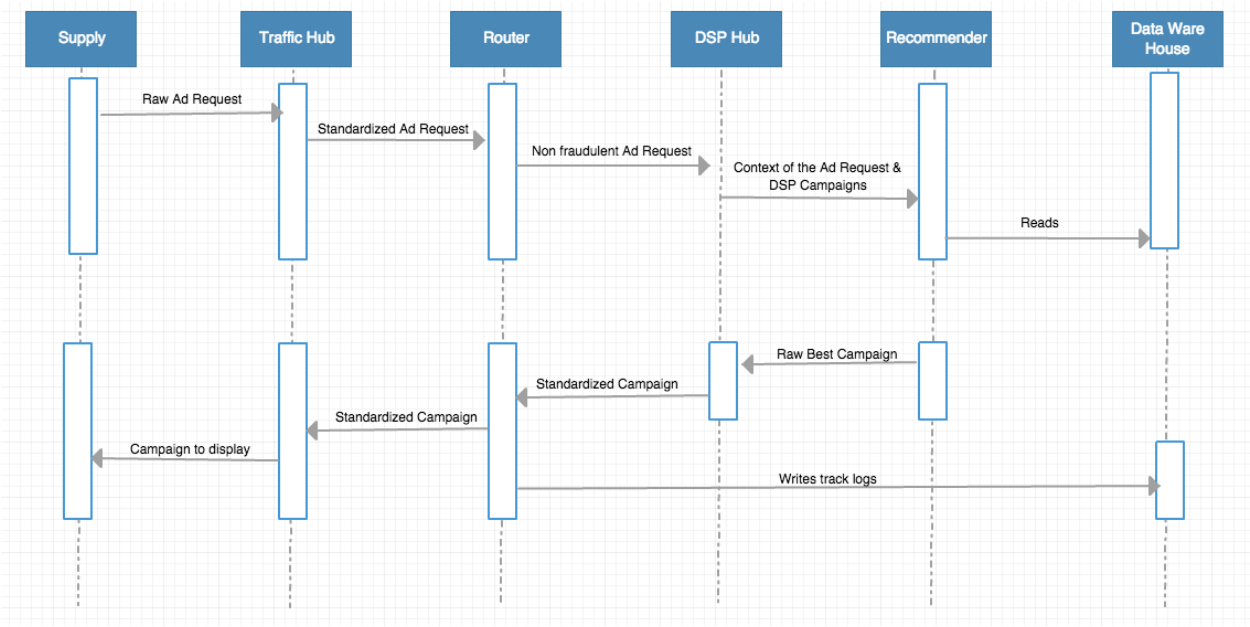

Figure 26 - UML Sequence Diagram of the company Ad Exchange.
Source: the author.

Now that we roughly showed the information flowing in the company Ad Exchange, we take a deeper look at the communications between the different services.

Traffic Hub - Router Communication:

Between the Traffic Hub and the Router, the possible Request/Response pairs are the following:

Situation 1: Traffic Hub receives a broken Ad Request

If Traffic Shied evaluates the Ad Request as broken, it does not request anything to the Router. Thus, no response is expecting from the Router. Nothing happens.

Figure 27 - Broken Ad Request: No Request/Response Pair.
Source: the author.

Situation 2: Traffic Hub receives an appropriate Ad Request

When Traffic Hub receives an appropriate Ad Request, it standardizes it, forwards it to the Router and expects a campaign back. If Traffic Hub fails at processing the Ad Request, no request is sent to the router - no response is expected either - and nothing happens. If Traffic Hub successfully processes the Ad Request, two request/response pairs are possible:

- Request/Response Pair 1: Traffic Hub sends a standardized Ad Request to the Router - Request. The Router responds with a campaign to be displayed -Response. This is the perfect situation where no problem occurs.

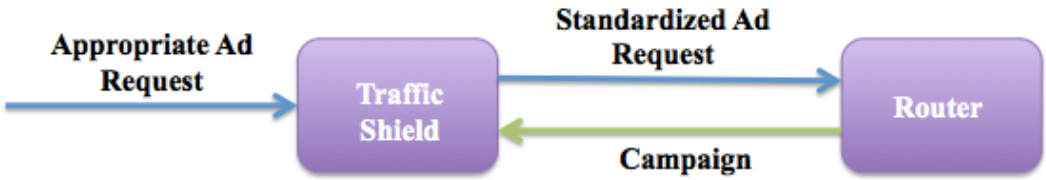

Figure 28 - Appropriate Ad Request: Traffic Hub sends a standardized Ad Request and Router responds with a Campaign to display.
Source: the author.

- Request/Response Pair 2: Traffic Hub sends a standardized Ad Request to the Router - Request. An error occurs somewhere in the rest of the process -Router, DSP Hub or

Recommender - and Router is not able to return a campaign to display so it responds with an appropriate error message.

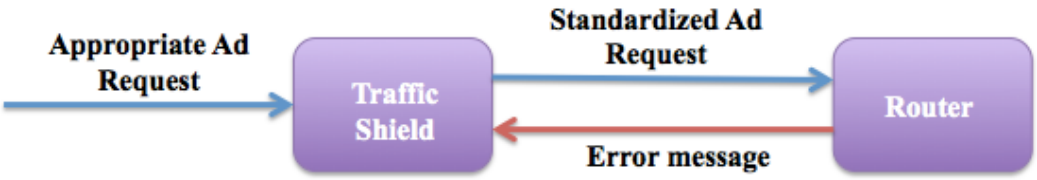

Figure 29 - Appropriate Ad Request: Traffic Hub sends a standardized Ad Request and Router responds with an error message.
Source: the author.

Router - DSP Hub Communication:

Between the Router and the DSP Hub, the possible Request/Response pairs are the following:

Situation 1: No error

The Router forwards to the DSP Hub the standardized Ad Request that it received from the Traffic Hub - request. The DSP Hub responds with a campaign to display - response. In this situation, no error occurs.

Figure 30 - Communication between Router and DSP Hub in situation of no error. Source: the author.

Situation 2: Error

The Router forwards to the DSP Hub the standardized Ad Request that it received from the Traffic Hub -request. An error occurs somewhere in the rest of the process - DSP Hub or Recommender - and DSP Hub is not able to return a campaign to display so it responds with an appropriate error message.

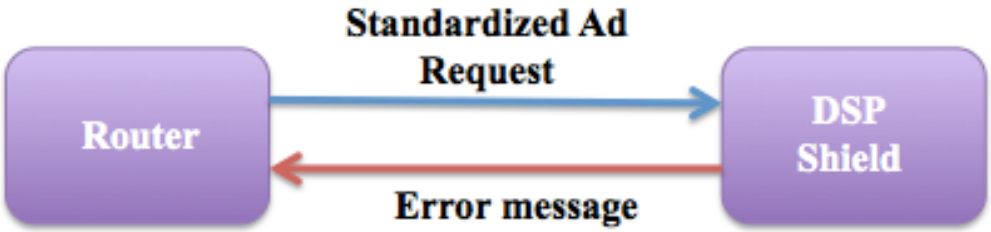

Figure 31 - Communication between Router and DSP Hub in situation of an error.
Source: the author.

DSP Hub - Recommender Communication:

Between the DSP Hub and the Router, the possible Request/Response pairs are the following:

Situation 1: No error

The DSP Hub sends to the Recommender both the information related to the context of the Ad Request and the bid responses from the Demand Side Partners -request. The Recommender responds with the campaign related to the winning bid response.

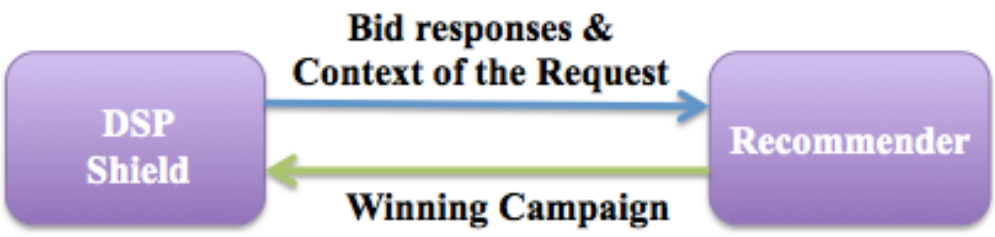

Figure 32 - Communication between DSP Hub and Recommender in situation of no error.
Source: the author.

Situation 2: Error

The DSP Hub sends to the Recommender both the information related to the context of the Ad Request and the bid responses from the Demand Side Partners - request. An error occurs in the Recommender process and Recommender is not able to return the winning campaign so it responds with an appropriate error message.

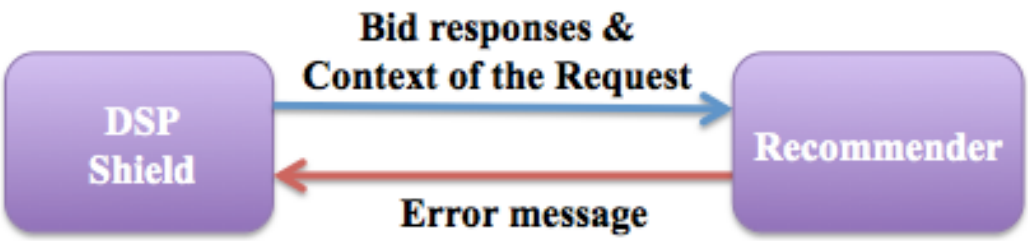

Figure 33 - Communication between DSP Hub and Recommender in situation of an error.
Source: the author.

4.2.1.4. Building the APIs between the different services of the company Ad Exchange

In the previous section, we identified the possible request/response pairs for two communicating services of the company Ad Exchange. In order to build an efficient Service Oriented Architecture, we need to define the "communication gates" - APIs - that will rule the communication between the different services. Each service wrote its own API in order to control the requests that they receive from other services. A request that does not respect the API will be automatically rejected and an appropriate error message will be sent back to the requesting service.

We considered each service separately and built the different APIs that rule its communications with the other services.

Traffic Hub APIs:

Two types of request can solicit the Traffic Hub:

- An Ad Request coming from a Supply Side Platform or the company SDK, this defines the Supply - Traffic Hub SDK,
- A campaign to display coming from the Router, this defines the Router - Traffic Hub API.

Supply-Traffic Hub API:

Considering the pieces of information that the DSP need to bid on an available impression, an ad request will be processed only if it includes the following fields:

- Name of the mobile website or mobile application,
- Domain of the mobile website or mobile application,
- Page of the mobile website - if the request comes from a website,
- Width of the banner - or player if the impression is for a video advertisement,
- Height of the banner - or player if the impression is for a video advertisement,
- Position of the impression,
- File types of the advertisement supported by the impression -.mp4, .jpeg...

All the other identified fields, such as the content categories of the application or website, the information on the user, the operating system of the device... are optional. The more information an Ad Request includes, the better, but event if those fields are missing, the request will be processed.

Router-Traffic Hub API:

The only information that needs the Traffic Hub to display a marketing campaign is the advertisement itself as a file. Traffic Hub is not responsible for serving an inadequate

campaign, so it does not do any check to ensure that the advertisement is compatible with the available impression. A campaign will be display by the Traffic Hub if it includes the following field:

- Advertisement to be displayed.

Other fields such as links to track the installs and clicks on the campaign are optional and do not prevent the Traffic Hub from displaying the campaign.

Router APIs:

Two types of request can solicit the Router:

- A complete Ad Request coming from the Traffic Hub, this defines the Traffic Hub-Router API,
- A campaign to display coming from the DSP Hub, this defines the DSP Hub-Router API.

Traffic Hub-Router API:

The Router is responsible for controlling the origin of the Ad Request and check if the mobile website or application has not been flagged as fraudulent by the company fraud system. To do so, the Router only requires the following field from Traffic Hub:

- Unique media identifier -of the application or website- as provided by the company SDK or the Supply Side Platform.

The Router does not require the presence of the other fields related to the context of the Ad Request -information about the impression, the device, the user... It is not the duty of the Router to check if they are present or not.

DSP Hub-Router API:

The Router is also responsible for tracking the data related to the campaign that will be displayed. This means that the Router requires certain pieces of information about the

campaign in order to count and inform the DSP how many times its campaign has been clicked or downloaded. To do so, the Router requires the following fields from the DSP Hub:

- A Click URL, which is basically an URL that will be pinged any time the end user clicks on the campaign,
- An Install URL, which basically an URL that will be pinged any time the end user downloads the mobile application that the campaign is promoting.

DSP Hub APIs:

Two types of request can solicit the DSP Hub:

- An Ad Request coming from the Router, this defines the Router-DSP Hub API,
- The best campaign from all the DSPs, coming from the Recommender, this defines the Recommender-DSP Hub API.

Router-DSP Hub API:

The DSP Hub is responsible for forwarding the Ad Request to all the Demand Side Partners of the company. Regarding the results of the survey that was sent to all the DSPs in the first part of this work, and similarly to the Supply-Traffic Hub API, the DSP Hub requires the following fields from the Router:

- Name of the mobile website or mobile application,
- Domain of the mobile website or mobile application,
- Page of the mobile website -if the request comes from a website,
- Width of the banner -or player if the impression is for a video advertisement,
- Height of the banner -or player if the impression is for a video advertisement,
- Position of the impression,
- File types of the advertisement supported by the impression -.mp4, .jpeg...

Other information about the available impression is optional. If more information is present, it would help the DSPs evaluate the impression and bid but the Ad Request is processed anyway.

Recommender-DSP Hub API:

The only piece of information that requires the DSP Hub to identify the campaign that will be displayed is the unique identifier of the winning campaign. The DSP Hub requires the following field from the Recommender:

- Unique identifier of the winning campaign, as provided by the DSP.

With this identifier, the DSP Hub is able to pick the winning campaign out of all the campaigns that have been sent by the DSPs.

Recommender API:

The DSP Hub is the only service that needs to request the Recommender. This defines the DSP Hub-Recommender API.

DSP Hub-Recommender API:

The Recommender is responsible for choosing the most valuable campaign regarding the context of the Ad Request. Basically, the work of the Recommender is to pick the marketing campaign that is the most adapted to the end user. To do so, the Recommender compares the previous results of each DSP campaign in regard to the specific context of the given Ad Request. To do so, the Recommender requires the following fields from the DSP Hub:

- For each campaign sent by the DSPs:
    - Unique identifier of the campaign as provided by the DSP. This allows the Recommender to check in the Data Warehouse the previous results of the campaign. Those results were tracked by the Router.
    - The bid -price- of the campaign. This represents how much the DSP is ready to pay to have its campaign displayed.
- About the Ad Request:
    - No information is required to choose the best campaign, however, the more information the DSP Hub can provide, the better.

The Recommender does not require any specific piece of information related to the Ad Request to process the request. Indeed, in the hypothetic case of an Ad Request without information context, the Recommender would pick the campaign with the highest bid as the winning campaign.

4.2.2. Building the class diagram of each identified Service of the company Ad Exchange.

In the previous part, we described the Service Oriented Architecture of the future Ad Exchange. Given those results, the author was able to define the class diagram of each of the four unitary services that form the architecture.

4.2.2.1. Traffic Hub Class Diagram

The classes and their related attributes and methods that the author identified for the Traffic Hub are the following:

Class 1:

Name: AdRequest

Description: An Ad Request coming from a device.

Mother Class: No mother class.

Attributes	Methods
isSite	
mediaName	
mediaDomain	
sitePage	

impressionWidth

Class 2:

Name: API

Description: The interface that rules the communication with other services.

Mother Class: No mother class

Attributes	Methods
	accept()
	reject()
	forward()

Class 3:

Name: Supply-Traffic API

Description: The API that rules the communication between the Traffic Hub and the Supply.

Mother Class: API

Attributes	Methods
	rejectAdRequest()
	acceptAdRequest()
	forwardAdRequestToRouter()

Class 4:

Name: Router-Traffic API

Description: The API that rules the communication between the Traffic Hub and the Router.

Mother Class: API

Attributes	Methods
	rejectCampaign() acceptCampaign() displayCampaignInDevice()

Class 5:

Name: CampaignToDisplay

Description: A campaign to display that comes from the Router.

Mother Class: No mother class.

Attributes	Methods
hasAdvertisement	

Class 6:

Name: Log

Description: A log to be tracked coming from the supply.

Mother Class: No mother class.

Attributes	Methods
hasLogged	

Class 7:

Name: InstallLog

Description: A log tracking if the user has installed the application related to the marketing campaign.

Mother Class: Log

Attributes	Methods
hasInstalled	

Class 8:

Name: ClickLog

Description: A log tracking if the user has clicked on the marketing campaign.

Mother Class: Log

Attributes	Methods
hasClicked	

Class 9:

Name: LogTracker

Description: Is responsible for collecting the logs coming from the supply and forwarding them to the Router.

Mother Class: No mother class.

Attributes	Methods
	collectLogs()

forwardLogsToRouter()

The author drew the following UML class diagram upon the previous listing:

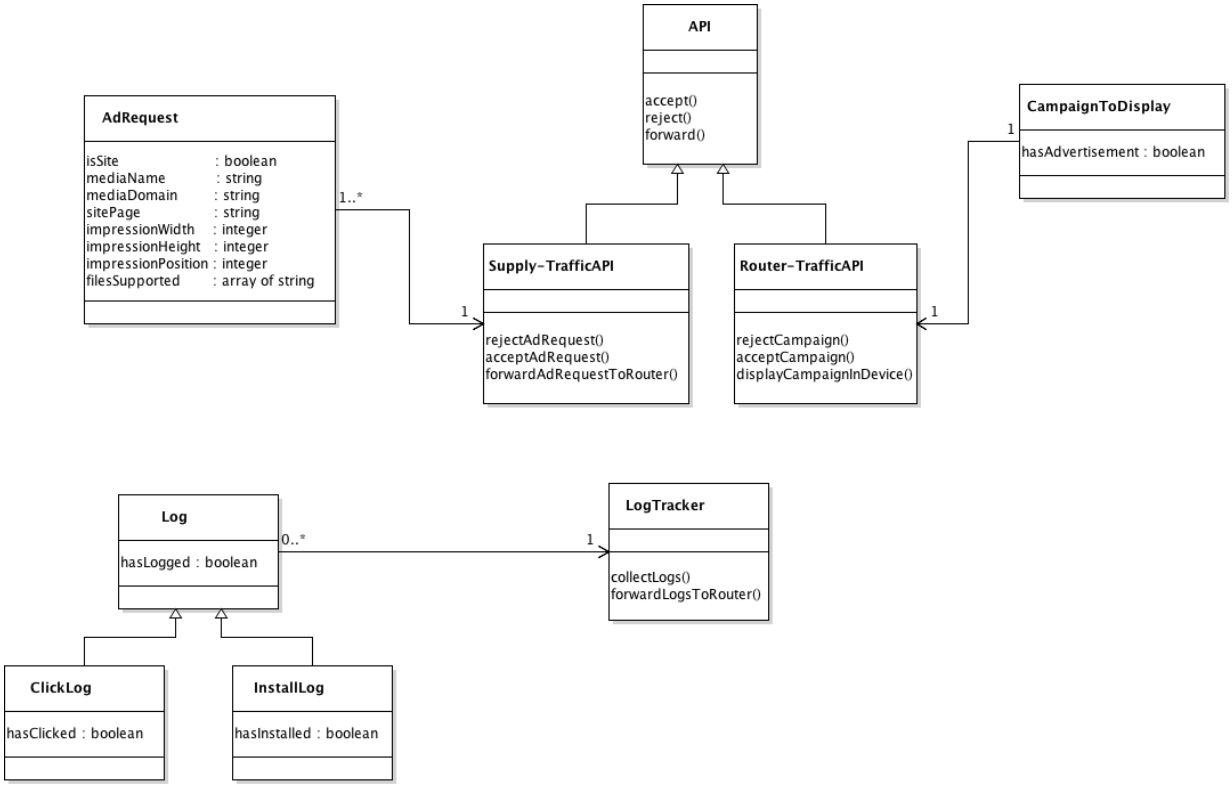

Figure 34 - UML Class Diagram of the Traffic Hub.

Source: the author.

4.2.2.2. Router Class Diagram

The classes and their related attributes and methods that the author identified for the Router are the following:

Class 1:

Name: AdRequest

Description: An Ad Request coming from the Traffic Hub.

Mother Class: No mother class.

Attributes	Methods
mediaIdentifier	

Class 2:

Name: API

Description: The interface that rules the communication with other services.

Mother Class: No mother class.

Attributes	Methods
	accept()
	reject()
	forward()

Class 3:

Name: Traffic-Router API

Description: The API that rules the communication between the Traffic Hub and the Router.

Mother Class: API

Attributes	Methods
	rejectAdRequest()
	acceptAdRequest()
	forwardAdRequestToFraudDetector()

Class 4:

Name: DSPHub-Router API

Description: The API that rules the communication between the DSP Hub and the Router.

Mother Class: API

Attributes	Methods
	rejectCampaign() acceptCampaign() forwardCampaignToTrafficHub()

Class 5:

Name: CampaignToDisplay

Description: A campaign to display that comes from the DSP Hub.

Mother Class: No mother class.

Attributes	Methods
clickUrl install**Url**	

Class 6:

Name: Log

Description: A log to be tracked coming from the Traffic Hub.

Mother Class: No mother class.

Attributes	Methods
hasLogged	

Class 7:

Name: InstallLog

Description: A log tracking if the user has installed the application related to the marketing campaign.

Mother Class: Log

Attributes	Methods
hasInstalled	

Class 8:

Name: ClickLog

Description: A log tracking if the user has clicked on the marketing campaign.

Mother Class: Log

Attributes	Methods
hasClicked	

Class 9:

Name: LogTracker

Description: Is responsible for collecting the logs coming from the Supply Traffic and writing them into the Data Ware House.

Mother Class: No mother class.

Attributes	Methods
	collectLogs()

The author drew the following UML class diagram upon the previous table:

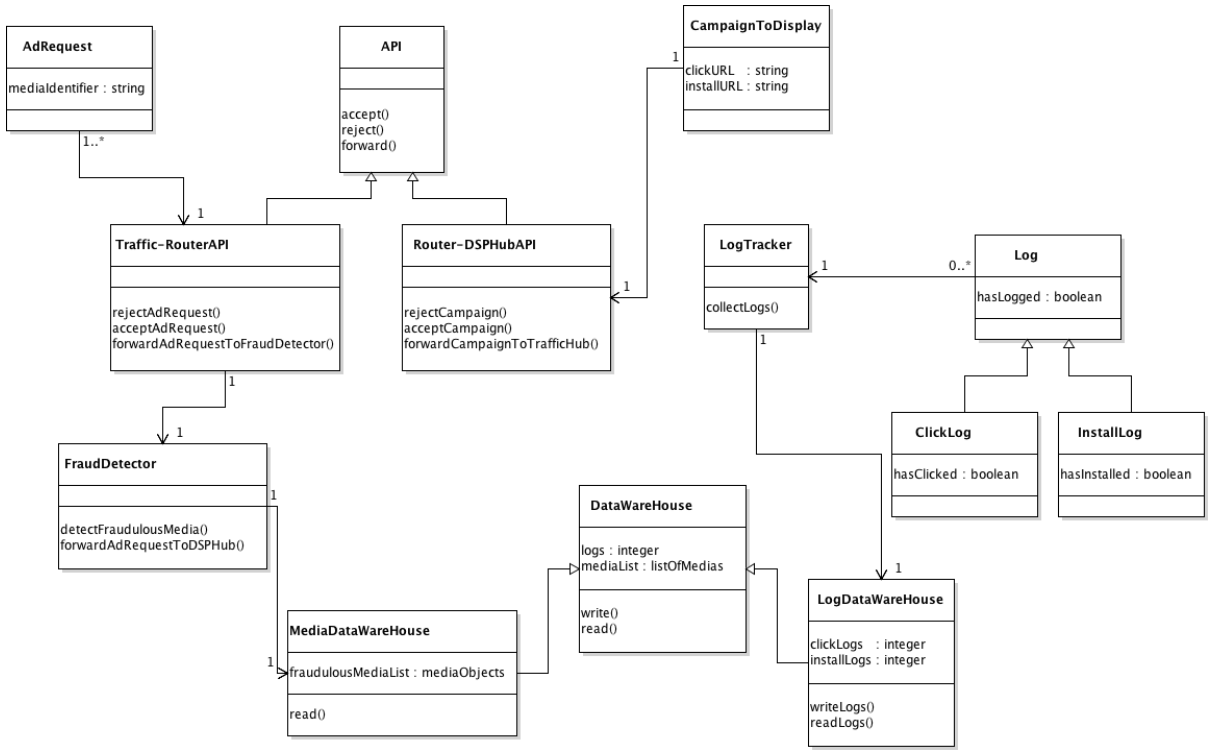

Figure 35 - UML Class Diagram of the Router.

Source: the author.

4.2.2.3. DSP Hub Class Diagram

The classes and their related attributes and methods that the author identified for the DSP Hub are the following:

Class 1:

Name: AdRequest

Description: An Ad Request coming from the Router.

Mother Class: No mother class.

Attributes	Methods
isSite	
mediaName	
mediaDomain	
sitePage	
impressionWidth	
impressionHeight	
impressionPosition	
filesSupported	

Class 2:

Name: API

Description: The interface that rules the communication with other services.

Mother Class: No mother class.

Attributes	Methods
	accept()
	reject()
	forward()

Class 3:

Name: Router-DSPHub API

Description: The API that rules the communication between the Router and the DSP Hub.

Mother Class: API

Attributes	Methods
	rejectAdRequest() acceptAdRequest() forwardAdRequestToBidHub()

Class 4:

Name: Recommender-DSPHub API

Description: The API that rules the communication between the Recommender and the DSP Hub.

Mother Class: API

Attributes	Methods
	rejectCampaign() acceptCampaign() forwardCampaignToCampaignBuilder()

Class 5:

Name: BidHub

Description: Is responsible for calling the DSPs and gathering their bid responses.

Mother Class: No mother class.

Attributes	Methods
	buildBidRequests()
	gatherBidResponses()
	forwardBidResponsesToRecommender()
	forwardAdRequestToRecommender()

Class 6:

Name: DSP

Description: A bid request is sent to all the DSPs and they answer with a bid response.

Mother Class: No mother class.

Attributes	Methods
bidRequestUrl	callForBid()
winNotifyUrl	winNotify()

Class 7:

Name: Campaign

Description: Represents both a marketing campaign included in a bid response or the winning campaign returned by the recommender.

Mother Class: No mother class.

Attributes	Methods
campaignIdentifier	

Class 8:

Name: BidResponseCampaign

Description: A marketing campaign returned by a DSP through a bid response.

Mother Class: Campaign

Attributes	Methods
campaignIdentifier	

Class 9:

Name: WinningCampaign

Description: The best campaign chosen by the Recommender to be displayed.

Mother Class: Campaign

Attributes	Methods
campaignIdentifier	

Class 10:

Name: CampaignBuilder

Description: Is responsible for standardizing the winning campaign and forwarding it to the Router.

Mother Class: No mother class

Attributes	Methods

	formatWinningCampaign() forwardWinningCampaignToRouter()

The author drew the following UML class diagram upon the previous table:

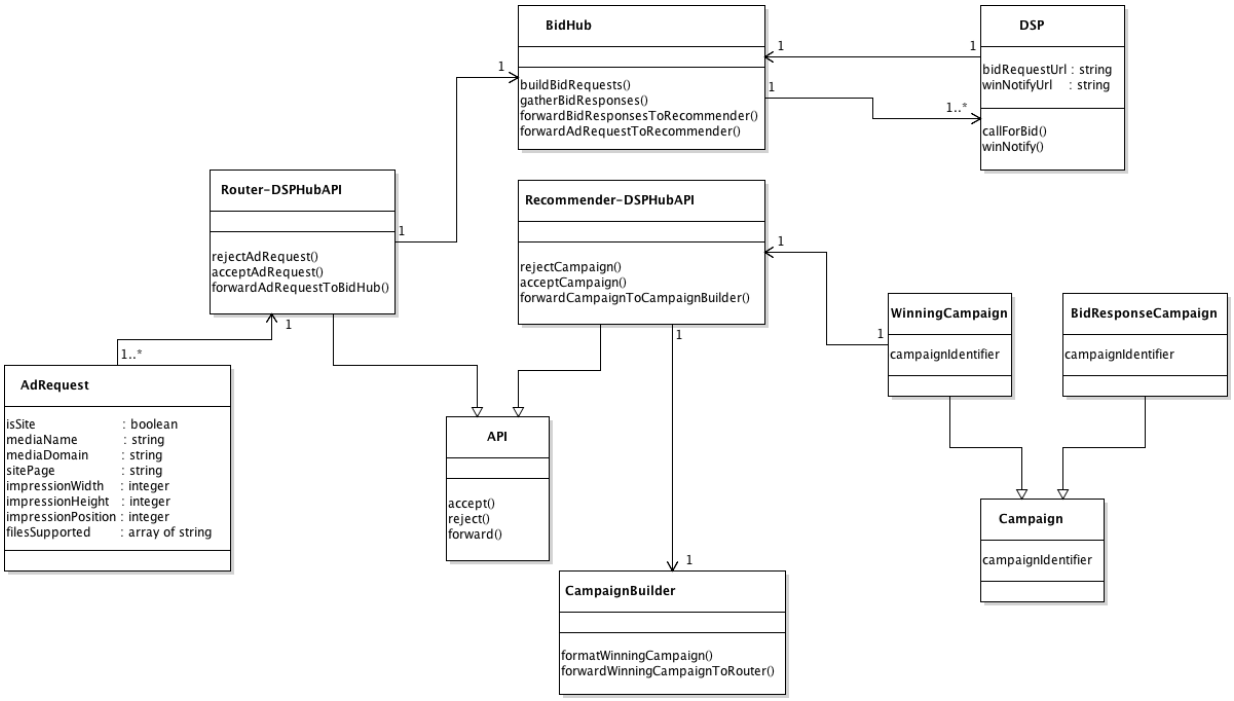

Figure 36 - UML Class Diagram of the DSP Hub.

Source: the author.

4.2.2.4. Recommender Class Diagram

The classes and their related attributes and methods that the author identified for the Recommender are the following:

Class 1:

Name: AdRequest

Description: Information about the context of an Ad Request -user, device, impression- coming from the DSP Hub.

Mother Class: No mother class

Attributes	Methods
adRequestContext	

Class 2:

Name: Campaign

Description: Represents both one of the campaign coming from the DSP Hub and the winning campaign that has been picked.

Mother Class: No mother class

Attributes	Methods
campaignIdentifier bidPrice	

Class 3:

Name: API

Description: The interface that rules the communication with he DSP Hub.

Mother Class: No mother class

Attributes	Methods
	acceptCampaign() rejectCampaign() forwardCampaignToRecommender()

Class 4:

Name: Recommender

Description: Is responsible for picking the best campaign from the DSPs.

Mother Class: No mother class

Attributes	Methods
	selectBestCampaign() forwardBestCampaignToDSPHub()

The author drew the following UML class diagram upon the previous table:

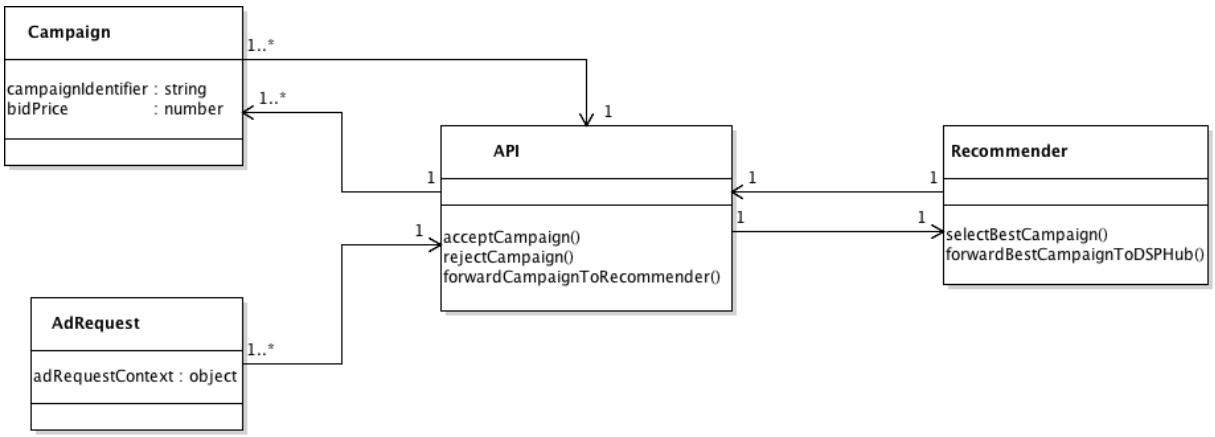

Figure 37 - UML Class Diagram of the Recommender.

Source: the author.

5. Conclusion

The objective of the present work was to show the stakes of building a reliable Exchange platform in the Mobile Advertising world. In this highly competitive and constantly evolving industry, a Mobile Ad Exchange works as a trading platform where the demand for mobile traffic meets the supply that sells impressions. The main characteristic of an Ad Exchange is to allow the impressions to be traded individually. This requires the Ad Exchange to develop the technology capable of processing huge quantity of data in real time.

In this work, we studied the case of the conversion of a Mobile Ad Agency into a Mobile Ad Exchange. First, we asked the company's future Demand Side Partners about the information they need to bid and buy mobile traffic. In parallel we identified the technical characteristics of the network that would handle the high frequency communication between the Ad Exchange and the DSPs. Second, we focused on drawing the service architecture of the Ad Exchange itself. In this part we defined precisely how the platform would process the requests and the flows of information that would run through it.

The present study provided strategic insights for the company to evolve from a common Ad Agency - trading blocks of impression and selling mobile traffic at fix rate - into a modern Mobile Ad Platform able to provide traffic for Real Time Bidding platforms such as DSPs. The study aims to detail the critical aspects to keep in mind when building a digital advertising exchange platform:

- Provide complete and reliable information about the traffic to the Demand Side Partners,
- Build a network capable of handling high frequency communication,
- Provide great control on the data flows inside the Exchange by drawing a Service Oriented Architecture of the platform.

The author wishes that anyone with interest in the Digital and Mobile Advertising Industry could enjoy this work.

6. Bibliography

ANDERSON, C. Free: Free, The Future Of A Radical Price, Hyperion Books, 2009

ANKIT, An In-depth Article on Real Time Bidding. Available at < http://www.adpushup.com/blog/the-ultimate-guide-to-real-time-bidding/>, 2014

BARRY, D. Service-Oriented Architecture, Available at <http://www.service-architecture.com/articles/web-services>, 2014

BRADEN, R. RFC 1123, Requirements for Internet Hosts – Application and Support, (ed.), October 1989

BRESLIN, M. Data Warehousing Battle of the Giants: Comparing the Basics of the Kimball and Inmon Models. Business Intelligence Journal, v. 9, n. 1, p. 6-20, 2004.

CRISTAL, G. Ad Serving Technology: Understand the marketing revelation that commercialized the Internet, January 4, 2014

DEVLIN, B. A.; MURPHY, P. T. An Architecture for a Business and Information System. IBM Systems Journal, v. 27, n.1, p. 60-80, 1988

DUSTDAR, S.; SCHREINDER, W. A survey on web services composition. International Journal of Web and Grid Services, 2005

FARREL, A. The Internet and Its Protocols: A Comparative Approach (The Morgan Kaufmann Series in Networking), May 13, 2004

FOX, V. Marketing in the Age of Google: Your Online Strategy is Your Business Strategy. New Jersey: John Wiley & Sons, 2010.

GADZOOG, Mobile Marketing Tips for Good Branding. Gadzoog Magazine. Retrieved 13 July 2013

GRAPPONE, J.; COUZIN, G. Search Engine Optimization: An Hour a Day. Indianapolis: Wiley Publishing, 2006

HARPER, D. "server". Online Etymology Dictionary. Retrieved 30 November 2013

HAYES, F. The Story So Far. Computerworld.com. 2002. Available at: <https://www.computerworld.com/s/article/70102/The_Story_So_Far?taxonomyId=009>.

INMON, W. H. Building the data warehouse. New York: Wiley Computer Pub., 1996.

INMON, W. H.; HACKATHORN, R. D. Using the data warehouse. New York: Wiley, 1994.

JIANG, B. Is Inmon's Data Warehouse Definition Still Accurate? Available at: <http://www.b-eye-network.com/view/16066>, 2012

KAPLAN, A. If you love something, let it go mobile: Mobile marketing and mobile social media 4x4 Found, Business Horizons, 2012

KARJALUOTO H.; LEPPANIEMI, M. Factors influencing consumers' willingness to accept mobile advertising: a conceptual model, Int. J Mobile Communications, Vol 3, No. 3, 2005

KIMBALL, R. The data warehouse toolkit: the complete guide to dimensional modeling. New York: Wiley, 2002.

KOTLER, P.; ARMSTRONG, G. Marketing An Introduction. Fourth Edition. New Jersey. Prentince Hall International, 1997

MACHADO, F. N. R.Tecnologia e projeto de data warehouse : uma visão multidimensional. São Paulo: Érica, 2006.

ORR, K. Data Quality and Systems. Communications of the ACM. New York, v. 41, n.2, p. 66-71.1998

ROOSE, M. Service Oriented Architecture, available at <http://searchsoa.techtarget.com/definition/service-oriented-architecture>, 2014

RUMBAUGH, J.; JACOBSON, I.; BOOCH, G. The Unified Modeling Language Reference Manual *The Addison-Wesley object technology series. Addison-Wesley, 2010.*

SLACK, N. et Al Operations management. 5th edition. FT Prentice Hall, 2007

SMITH, M. Targeted: How Technology Is Revolutionizing Advertising and the Way Companies Reach Consumers, 2014.