UNIVERSIDAD DEL TURABO
SCHOOL OF BUSINESS AND ENTREPRENEURSHIP

THE IMPORTANCE OF APPLICATION PROGRAMMING INTERFACE (API) AND MASHUPS IN WEB DEVELOPMENT FOR BUSINESS OPPORTUNITIES

by

José Raúl Reyes

Gurabo, Puerto Rico

May 2018

UNIVERSIDAD DEL TURABO

CERTIFICATION OF APPROVAL OF DISSERTATION

The dissertation of José Raúl Reyes was review and approved by the dissertation committee members. The Doctoral Degree Academic Requirements Completion Form with the signature of the members of the committee has been filed in the Register and the Center for Graduate Studies and Research of Universidad del Turabo.

MEMBERS OF THE COMMITTEE

Dr. Ángel Ojeda Castro

Universidad del Turabo

Chair

Dr. Juan Bautista Valera

Universidad del Turabo

Member

Dr. Eulalia Márquez

Universidad del Turabo

THE IMPORTANCE OF APPLICATION PROGRAMMING INTERFACE (API) AND MASHUPS IN WEB DEVELOPMENT FOR BUSINESS OPPORTUNITIES

by

José Raúl Reyes

Dr. Ángel Ojeda Castro

Director of the Dissertation Committee

ABSTRACT

Developers are shifting from technical software creators within a business, into change agents. To succeed in the Internet economy, companies must take in consideration how the API model impacts their business vision, and the actions needed to incorporate an API strategy; where business opportunities are entangled within the Internet economy. Through the analysis of empirical data, a total of 21 categories with 29 types of APIs were collected with their respective developers, mashups, SDKs, articles, libraries, followers and derived company values. The results suggested that APIs and mashups are essential for web development within the API Internet economy and can be traced down to the developer's role and connection with its purpose and functionality. The categorization and relational analysis of APIs can provide a worth of information for companies, developers and the academia to search for possible business opportunities hidden at plain site by the obscurities the Internet entails.

Keywords: APIs, developers, mashups, articles, libraries, Internet economy, SDKs, business entities and category

CURRICULUM VITAE

Master's Degree in Information Systems (MIS)

EDP College, 2003

Professional Certificate in Information Systems Auditing

EDP College, 2003

Bachelor's Degree in Business Administration (BBA)

Major: Economics

University of Puerto Rico, 1994

DEDICATION

To my mother “Jenny” for giving me the best of her years. Her husband. My father. My brothers Andrei, Maritza, Luis and Oscar from heaven. To my son Jonathan for trading adventures for a desk full of papers and books. To my best friend Nelson for his constant support. And my sweetheart “Grace” for giving me strength and encouragement throughout this process.

ACKNOWLEDGMENT

I extend my sincere thanks to the members of my dissertation committee for their patience, guidance, dedication and support, especially Dr. Eulalia Marquez, Dr. Angel Ojeda, Dr. Juan Valera, Dr. Edgar Ferrer and Professor Pablo Colon. Their mentoring and extensive knowledge in research practices made possible the completion of my doctoral dissertation.

TABLE OF CONTENT

Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

CHAPTER I

INTRODUCTION

1.0 Background

In accordance with Dearstyne (2007), Web 2.0 or second-generation web-based services has changed the way people use and share information, by allowing the creation of collaborative tools, such as mashups to serve as platforms on the Internet communication highway. In the words of Marjit and Jana (2009), “As the quality, quantity, and diversity of information grow users long for tools to access and manage this bewildering array of information.”

1.1 Mashup Technology

Mashups can become an alternative tool for end-users possessing little or no programming experience, who intent to create a new application to be use in the Web. Mashup technology serves as a client-centric web-based multipurpose integrator, by assisting computer end-users in the process of combining multiple web content, and in the creation of customized applications, destined to satisfy computing needs. As mentioned by Kendall and Schmidt (2007), computer professionals are currently mixing APIs (A source code interface, designed to support request by end-users, for services generated by the computer software.), to create new applications.

Benslimane, Dustdar and Sheth (2008), addressed that the mashup concept is an action of sharing and aggregating information to deliver publishable content, as a newly generated Web product. Accordingly, computer end-users have phased out as content consumers to evolve into content providers (Thuchinda, Szekely and Knoblock,

2008). Previously, to develop a sophisticated mashup application, required a high level of programming knowledge and expertise from the end-user. However, according to Ariel, Bartolini, Bergman, Mordechai, Peltz, & Tadeski (2010), there are existing mashup platforms which use a Graphical User Interface (GUI) mechanism, to facilitate the mixture of multiple components, to create new composite applications.

Bakalov, Konig-Ries, Nauerz, & Welsch (2009), have developed an ontology-based multidimensional personalization model, capable of supporting the automatic generation of mashups, to assist the end-users in information needs. The technical requirement for creating mashups application is becoming less of a setback to the technical-savvy, due to the availability of numerous tools throughout the Internet supporting the mashup programming environment, such as Dapper, Intel's late MashMaker and Microsoft's Popfly. At a minimum, mashup technology uses publicly available source code, Web feeds and data, where according to Cho (2007), has become a hybrid useful tool formed by a combination of information sources.

Mashup technology according to Hinchcliffe (2009) is the key for future software development, where "A unified mashup model can increase software quality, lower Information Technology (IT) costs and drive choice and innovation". Within a business context, Hinchcliffe's (2009), views the mashup platform, as a time reduction mechanism which permits end-users to tailor Web application to specific needs. Providing an increase in the amount of output per unit of input (Snir and Bader, 2004).

1.1.1 API's

An Application Programming Interface, is the building blocks or source code interface, designed to support request by end-users, for services generated by computer software (Kendall and Schmidt, 2007). Two or more APIs units combined

constitute a mashup. An API module developed by any company will serve as a bridge to their resources, in the interest of the consumers, as visualized by the developer. Based on this programming power augmentation, APIs and mashups constitute a new way of programming, suited for developers with diverse backgrounds.

1.1.2 The API and Mashup User

In consonance with Yue (2010), the average end-user does not possess the necessary advanced programming skills to customize web services on their own. However, for Kendall and Schmidt (2007), many APIs available on the Web are open source, and can be used free of charge making mashup technology a rapid application developing tool.

These APIs provide an opportunity for developers to customized complex applications, without the need of high level computer programming skills as previously required with traditional coding and platforms. The World Wide Web Consortium (www.w3.org, 2018), led by Tim Berners-Lee and CEO Jeffrey Jaffe, visualized a semantic web or the “web of linked data”, destined to facilitate people the creation of stored data on the Web, build vocabularies, and write rules for handling data, where according to Ankolekar, Krotzsch, Tran and Vrandecic (2007), are built on a decentralized and open infrastructure.

1.2 Problem Statement

The API ecosystem paradigm serves as a central point to address the difficulties associated or inherent to the offspring of new technologies and developmental changes within the Internet economy. These challenges are exposed by Bharadwaj, A., El Sawy, O., Pavlou, P., & Venkatraman, N. (2013), businesses in a digitally intensive world operate within ecosystems where their digital business strategy cannot be conceived

outside said environment. Under these conditions, firms should consider the role of networks effects and multisided business models, where complex and dynamic coordination across multiple companies are required to create and capture value in this digital setting.

1.3 Importance of the Study

The development of an API model, designed to demonstrate API tendencies within categories throughout the Internet, can serve as a barometer to measure business opportunities with return on investment (ROI) potential within the Internet economy.

1.4 Objective of the Study

To measure the relationship between APIs, developers, SDKs, mashups business entities (companies) and other related fields, by analyzing secondary data recorded in the specialized public APIs and mashups prime website (www.programmableweb.com). The purpose of APIs and mashups is to enhance user's experience throughout web applications.

1.5 Contribution to the Field of Study

The significance of this dissertation resolves in measuring through secondary data the API and mashup tendencies within the Internet economy under the scope of the developer for the exploitation of business opportunities.

1.6 Research Questions

1. Which are the most prevalent API categories and tendencies?
2. Which APIs are used the most by developers, and which companies predominate?

3. What are the similarities or differences between APIs and SDKs within category? Does any relationship exist between them?
4. What are the similarities or differences between APIs and developers within category? Does any relationship exist between them?
5. What are the similarities or differences between APIs and mashups within category? Does any relationship exist between them?
6. How do APIs, developers, mashups behave within categories under the scope of social network analysis?

1.7 Definitions of Terms

API – Stands for Application Programming Interface, the building blocks or source code interface, designed to support request by end-users, for services generated by computer software (Kendall and Schmidt, 2007). Mulesoft (2018), defines an API as, "a software intermediary that allows two applications to talk to each other."

Categories- A grouping mechanism established by www.programmableweb.com, to denote the different uses the APIs, mashups, SDKs perform.

Developer/experience end-users – In accordance to Merriam-Webster (2018), is a person who creates, elaborate, specifically by deliberated effort new ways of doing business i.e. software. Represents, any individual making use of information technology applications, whose main duty is not computer programming (Zang, 2009).

Internet Economy – In accordance with Barua, A., Whinston, A. B., & Yin, F. (2000), the Internet economy involves the following components: a worldwide IP networks, human capital, a deployed open global network environment, interconnectivity for market exchange, online multiple players, security, a currency mechanism and a

legal/policy frame work; under the infrastructure of global high-speed, IP-based networks, and the human factor of a myriad of services.

Mashups – Term coined from the musical industry, representing the creation of a new song derived from the mixture of others (Merrill, 2006). It is the result of technology compatibility, Web services and data provider's willing participation. The two active ingredients for Web mashup development are the combination of data and APIs, which provide the interface required to gain access and control of data without the need of sophisticated programming expertise (Palfrey and Gasser, 2007).

SDK – Acronym for Software Development Kit.

Web 2.0 – A set of principles and practices that tie together user collaboration, participation, and enable the Web to function as a single universal platform (Kendall and Schmidt, 2007) and (O'Reilly, T., 2005).

Web-Based Environments – A generic Web-based application is comprised of the four types of information technologies, Simple HTML/XML/XSK Platforms, Client-Side Scripting, Server-Side Scripting/Server-Side Processing and Embedded Objects/Controls (Citrix Consulting Services, 2002).

CHAPTER II

LITERATURE REVIEW

2.0 Introduction

Chapter two will discuss relevant academic literature related to the APIs and mashup technology on the Web. Said concepts are guided by scholastic journals in the areas of management information systems, computer science, and computer engineering, including dedicated APIs and mashups research sites.

To address the APIs and mashup phenomena within the Web, it is imperative to abide at proven business and social science concepts specially formulated for information systems studies.

2.1 Applied Information Systems for Business Function

In terms of information systems, the productivity paradox concept was originated in 1987, when Nobel Prize laureate Robert Solow stated, that “computers could be seen everywhere but in productivity statistics”; inferring IT adds no input to productivity (Peslak, 2003). Thereafter, numerous investigations have examined the IT productivity paradox to determine if there is a significant positive effect on the overall business performance and productivity. An empirical research, using a variety of market and financial-based instruments determined, according to Peslak (2003), that “IT expenditures had a significant and positive impact on firm productivity based on a common financial or market-based measure “.

Accordantly, Sircar and Choi (2007), expressed that the productivity paradox has been refuted through research by demonstrating that IT investments do indeed increase various aspects of a firm’s performance. However, they also addressed, that

representing how IT investments account for improvements in various measures of output has been difficult. Same authors determined as well, that the productivity paradox did not hold for IT labor or capital, since both inputs show a significant impact on firm revenues.

To calculate the IT value within the organization, the IT structure must be dismantled into its simplest component within the firms' productivity function. In accordance to Kumar (2004), IT infrastructure contains the following components such as: a collection of technologies, human resources, interoperability, effective data management, integration, connectivity, and security.

Kumar (2004), also addressed, that the effectiveness of an IT infrastructure can be measured using the following criteria: reliability, when IT is operating in low downtime, flexibility which can quickly and economically adapt to business changes, and finally upgradability the stage for adapting, deploying multiple or complex technologies.

The managerial aspect of productivity analysis, in terms of technology, is to answer how input resources are best employed within a process, to determine how changes in technology impact operations through time (Kriebel and Raviv, 1980). In accordance to Yue (2009), mashup technology is interwoven within three major IT trends: Web 2.0, situational software applications and end-user programming. Also, to sustain software development: the creation of feature rich, easy to use and domain specific web-based environments are necessary for mass "*amateurization*" of end-user programming (Yue, 2009). In accordance to Alpar and Kim (1990), the effectiveness of information systems was measured by incorporating the user satisfaction method,

without being able to connect said function with the overall economic firm performance. Snir and Bader (2004), proposed a framework for measuring the output level of high performance computing systems, by using productivity and utility economic theory as their barometer.

Another framework to measure profitability and productivity was designed by Eilon (1985), using numerous performance ratios, such as: the physical volume, revenue, profit, total investment, or a combination of these, depending on the inputs and outputs of the firm.

On previous mashup studies, instruments such as the Unified Theory of Acceptance and Use of Technology (UTAUT) model, which is composed of eight user acceptance elements, could only explain 36.6 percent of the variance in behavioral intentions (Bhattarai, Zhao & Crespi, 2010). In a similar manner, Yue (2009), using the late Yahoo Pipes in a pilot experiment, measured end-users experience in terms of mashup development, where student demonstrated no signs of difficulty when using the visual end-user programming environment on the Web. However, due to the reduce number of participants, the researcher considered the experiment a preliminary investigation.

In accordance with Basole (2016), using a curated dataset of nearly 15,000 APIs, was able to develop an API category ecosystem visualization structure, revealing distinct clusters within it, demonstrating that some categories were more prominently positioned than others in the API environment.

2.2 The Diffusion of Innovations in Society

The diffusion of innovations is interconnected through channels over time within a social setting, were the players have different level of willingness to adopt to a given novelty, which over time resembles a normally distributed curve. A closer look at the

curve's behavior, can provide a spectrum in which the individual innovativeness can be categorized under five segregating values (Rogers, 1995).

These five elements are composed of: innovators, early adopters, early majority, late majority, laggards. Each category possesses inherent values that can designate certain traits. Innovators tend to be venturesome and are educated. Early adopters usually have some leadership role and education. Early majority are more careful and funnel by way of informal social contacts. Late majority are usually skeptical and traditional under a lower socio-economic status. And last, the laggards are encouraged by neighbors and friends whom serve as evangelist (Rogers, 1995).

Based on Rogers research, the rate of adoption of innovations is also impacted by another set of five factors. The first four of these factors are generally positively correlated under: relative advantage, compatibility, trialability and observability. The last element named complexity, is usually negatively correlated (Rogers, 1995).

API developers fall under the first three stages of Rogers' paradigm, since they are willing to try out new technologies with unknown potential of growth. In accordance with Green and Hevner (2000), software developers may be motivated on using IT innovation since it can improve their effectiveness in a measurable way. In the words of Levina, Stephanb and Winkle (2012), "The costs of adopting a new technology depend not only on the price of acquiring the new technology, but also on the complementary investment and learning required to use the new technology."

2.2.1 Transaction Cost Economics Theory

Based on research conducted by Cordella (2006), the elementary unit of analysis as the economic exchange between at least two individuals, shows how the transaction costs model depicts the exchange process with reference to the resources that required to execute this exchange.

Economic agents invest in resources to mitigate the effects of these imperfections in the execution of the exchange. These investments are the costs associated with the transactions and defined as transactions costs. Transactions costs are the consequence of the asymmetrical and incomplete distribution of information among the economic agents involved in the transaction. Alvarez-Suescun (2010) declared that "Assuming similar production costs among firms, managers will choose the governance structure that minimizes transaction costs." In accordance to Chin-Chiung, K. U. O., & Chi-Fang, L. I. U. (2017), the Coase's theory is built on two fundamental assumptions which are the limited rationality on how capabilities are restricted in an environment full of uncertainties and opportunistic behavior dealing with the pursuit of self-interest which can enhance the effect of information asymmetry.

2.2.2 Theory of Technology-Organization-Environment

The Technology-Organization-Environment (TOE) framework conceptualizes the context of adoption and implementation of technological innovations pertaining three elements which consist of constraints and opportunities (DePietro, Wiarda, & Fleischer, 1990). These are (Tornatzky and Fleischer 1990; Thong 1999):

1. Technological – Innovation process were internal and external technologies used are relevant to the firm.

2. Organizational – Organizational profile to which the characteristics and resources of the firm are exposed on: size, centralization, formalization, managerial structure, human resources, slack resources, and linkages among employees.
3. Environmental – Adopted setting of the organization's operative were the environmental context includes the size and structure of the industry, competitors, the macroeconomic context, and the regulatory environment (DePietro, Wiarda, & Fleischer, 1990).

In accordance with Soto-Acosta, Perez-Gonzalez and Popa (2014), "The TOE framework has emerged as the main theoretical framework to analyze the different factors which affect the adoption and use of Internet technologies."

2.3 APIs and Mashups Foundations

Based on Citrix Consulting Services (2002), a generic Web-based application requires a combination of the following four types of information technologies to deliver the final user interface: simple HTML/XML/XSK Platforms, client-Side Scripting, server-Side Scripting/Server-Side Processing and embedded Objects/Controls. The Web-based environment will constitute the playground for APIs and mashups development.

Based on Knobel and Wilber (2009), Web 2.0 ethos promotes three interlocking functions or practices: participation, collaboration and distribution. In addition, innovators can contribute to media production with online services, capable of managing content generation as expressed by same authors.

Taft (2007), addressed that Web 2.0 technologies serves as front ends to Service Oriented Architecture (SOA) back-end systems; in addition, caution must be taken with said services, which are capable of boosting productivity in the one hand, while in the

other creating a problem if not guided with proper governance. The economic value of SOA according to the research performed by Mueller, Viering, Legner and Riempp (2010), is as stated:

> SOA is able to extend a firm's Information Systems capability in terms of both functional range and organizational reach. As external services can be integrated more easily, SOA has a significant potential to reduce vertical integration and improve interorganizational coordination.

SOA comes to play as stated by MacKenzie, Laskey, McCabe, Brown, and Metz (2006) "A paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains". SOA should be flexible during technical execution while creating new interfaces and composite services (Haines & Rothenberger, 2010).

The use of the Web as a platform, based on Web 2.0, has transformed the software life cycle. New applications are created using open source interfaces and content, within a highly interactive environment (Sabbouh, Higginson, Semy and Gagne, 2007).

Guinard, Trifa and Wilde (2010), addressed that the Web 2.0 environment, has significantly open the Web application development spectrum, by lowering its entry barrier, due to the openness and simplicity of network devices and applications.

2.3.1 Software Development Kit (SDK)

Sarrel (2016), determined that providing an SDK benefits the organization because it accelerates deployment, ensures best practices, an increases security and control of the company's brand image.

2.3.2 APIs Building Blocks

The APIs created have favored the simplification of website enhancement. The technical requirement for creating mashups application is becoming less of a setback to the non-technical, due to the availability of numerous tools throughout the Internet supporting the mashup programming environment.

Gamble and Gamble (2008), pointed out that mashups consume reusable resources where developers choose from a variety of APIs, reducing the time needed to assemble a new hybrid application. Rob van den Dam (2017), provided statistics indicating that 90% of Expedia revenues derives from APIs, higher than eBays' 60% and salesforce's 50%, opening a myriad of opportunities for public and private sector to generate income

A compound of Santos' statistical reporting concluded the following (2018): APIs since 2014, have an average of 2000 being added each year into the www.programmableweb site API directory. The top five most clicked APIs in 2017, were Netflix, Bloomberg, Twitter, Wunderground and Facebook (2018). The tracked top five APIs of all times are: Facebook, Google Maps, Twitter, Youtube and Accuweather (2017). The API types leading order, after migration process in 2014 are revealed as: Web/Internet, Product, Standard, Browser and System/Embedded. In terms of architectural style REST reigns in comparison to RPC, Native/Brower, Push/Streaming, Indirect and finally GraphQL (2017). In the same fashion, the top five APIs providers with the most SDKs listings are: Google, Microsoft, Mapbox, Paypal and Amazon. In parallel, the top five providers with the most sample code listings belong to Google, Microsoft, IBM, ESRI and Brightcove, aside from the independent developers (2017).

Invisible for the end-user, the mashup core or API elements are mainly responsible for data mediation, conversion, transformation, and combination of data feeds (Maximilien, E. M., Ranabahu, A., & Gomadam, K. 2008). Based on CA Technologies (2015), "SOA programs are generally driven by the need for IT cost savings, API programs focus on generating new revenue streams. A Web API connects a range of existing business assets to create value in previously unforeseen ways". The API's value as stated by Oracle (2014) is:

> Communications service providers must look beyond just exposing APIs. They are fostering innovation, and new businesses will emerge built on top of the APIs where significant value resides. CSPs are currently making millions of dollars through their APIs, but the potential is significantly greater.

Enterprises are using APIs/mashups applications that are capable of remixing data and information in new ways, to minimize the bottle neck effect, which according to Fichter, D. and Wisniewski, J. (2009), saves time and money.

There are three major barriers withholding the mashup burst in the enterprise, the arena composed of licenses and legalese for commercial use of the APIs, security issues inherent in the creation of new applications from disparate sources and organizational readiness in terms of IT infrastructure capable of fulfilling the data/content needs (Fichter, D. and Wisniewski, J., 2009).

Berlind (2017), explains that the different types of APIs, are labeled to understand their function and usage as follow: The Web/Network: made available to any user, the product: users buy an API where multiple versions will be available through the Web, the Browser: included within the browser functions, the Standard: bodies of standards

established by leading organizations ruling APIs and the System/Embedded: developed hardware open for developers to use proprietary API to access device.

Building API product management capabilities, according to IBM's Institute for Business Value documentation (2016), requires: The application of API standards, the creation of an API coalition within the organization, and the proper delivery and measures of the API benefits.

2.3.3 APIs Genetics

REST - Representational State Transfer as defined Fielding and Taylor (2002), is "a coordinated set of architectural constraints that attempts to minimize latency and network communication, while at the same time maximizing the independence and scalability of component implementations".

SOAP - An XML based protocol for exchange of information in a decentralized distribution environment, consisting of three parts where (Box, Ehnebuske, Kakivaya, Mendelsohn, Nielsen and Winer, 2000), define as a framework for describing what is in a massage and how to process it, a set of encoding rules for expressing instances of application-defined data types, and a convention for representing remote procedure calls and responses.

JavaScript - Dynamic, interpreted language typically used for animation or web page enhancement in a decorative sense (Mikkonen and Taivalsaari,2007).

XML-RPC - Remote Procedure Class, as defined by Allman (2003), "Provides the developer an interface to communications code that is as close as possible to simply making a procedure call.".

The API Architectural Styles defined by Berlind (2017), are comprise of the following:

1. Typical RESTful – Web API
2. RPC (XML-RPC, gRPC…) - SOAP
3. Push/Streaming (publish/subscribe) - Under WebSockets, Webhooks and other players
4. GraphQL- Unique merge between RESTful and RPC
5. Browser/Native – Browser native
6. Indirect – Based on SDKs.

Berlind (2017), also addressed the three functions within the API scope as, single purpose used for one action such as mapping, aggregate destined to work with a combination of other APIs or aggregated, and microservice which interreacts with multiple APIs within a specific sequential order.

Based on Wodehouse (2016), the API Ecosystem contains some of the following traits: Serves as assets to inherent value of the function or purpose it addresses, the API becomes the point of entry and filter, serves as a doorway, reads like a contract, permits the connections of applications and devices, permits reuse and aggregation, and acts as a generic plug. They are also facing out the monolithic IT infrastructure paradigm.

Wodehouse (2016), also addressed that developers will create new ways of API use as well as mashups, and that these apps will increase the value in user's experience, where at the end, the main beneficiaries are the end-users as stakeholders.

Santos (2017), stated that "APIs have provided tremendous value to countless organizations and developers, which is reflected in their continued growth". Based on Fischer, Bakalov and Nauerz, (2009), mashups permit developers to combine multiple

services from various websites, to create a new tool. Also, companies are making their information available by providing data ready to use formats. In the words of Dorn, C., Schall, D., and Dustdar, S. (2009), mashup technology has increased in popularity, due to the benefit it provides to the end-user by assisting in the composition of services and aggregate data, from multiple sources.

Visa and Apigee's whitepaper Growing your business with APIs (2015), stated that APIs open doors to agile innovation, rapid adaptation to market changes or competitors, and the chance to access invaluable data from this process. Based on Debasis and Biswajit (2015), consumers experience enriches with the use of APIs by expanding their knowledge of products and services. In said process, context-awareness and adaptivity are the major force in the widespread proliferation of service mashups. However, according to Dearstyne (2007), in the Web 2.0 environment, caution should be observed with organic documentation created in business transactions, decisions and legal obligations.

2.3.4 API Strategy and Security In-House Questions

Perficient (2017), addressed that to build an API strategy requires the following imperative questions or concerns: should APIs be private or public, how do APIs foster innovative development, how can we simplify access to business assets using APIs, what core considerations result in a strong API strategy, what will APIs contribute to the organization, and where should the company be in 10 years…

Boyd (2017), addressed that the following elements need to be considered by developers when dealing with API security: evaluate APIs attack risk vectors, when to publish the public APIs, whether API documentation be available, manage authorization and access to users, provide clearance for external developers, recognize security

issues, how to deal with API security risks and vulnerabilities, and how to manage API's product security management.

Within the same mindset, according to 451 Research (2015), the API lifecycle consist of: the designing phase which ensures API quality and consistency, the development phase consistent with the agile development, the testing phase rigorous platform test, the integration phase that deals with connectivity, the deployment phase addressing when the API is release. The management phase which handles the API administration process, the monitor phase overviewing the debugging process, usage and performance standards, the monetize phase which deals how to earn income through the contractual and payment structure, and the archive/retire phase which handle the repository status of the API or passé stage.

2.4 Mashups Building Blocks

Based on Jackson (2009), a mashup is a communication form able to compose, combine, assimilate and appropriate elements from existing works to create an original product. Mashups, according to Marjit and Jana (2009), serve as a client-centric web-based multipurpose integrator, by assisting computer consumers in the process of combining multiple web content, and in the creation of customized applications, destined to satisfy computing needs.

To determine the value mashup technology provides to an organization, in terms of performance and productivity, its functionality must be evaluated by the amount of system requirements it satisfies to said entity. Based on Dennis and Haley (2002), a requirement is a statement of what a system must possess; same authors established two types of requirements: functional and nonfunctional. Functional requirements deal with the type of processes the system will perform or the type of information it needs.

The nonfunctional requirements relate to performance and usability.

The mashup classification framework created by Beemer and Gregg (2009), provides researchers with a comprehensive outlook of mashup technology trends, and provides a structured literature based for future academic research. The following major categories were established by Beemer and Gregg (2009):

1. Access Control and Cross Communication: Studies the connection of disjointed applications to provide unified services.
2. Mashup Integration: Studies the aggregation of various types of data sources.
3. Mashup Agents: Studies the ability to semantically determine information sources that are relevant to the end-user.
4. Mashup Frameworks: Studies the elements needed to develop the best practices on framework creation.
5. Enterprise mashups: Studies how mashup technology relates to the business enterprise in issues such as accountability, design principals, and intranet deployment.
6. End-user programming: Develops end-user programming languages and tools, to enable non-technical end-users to easily create a customized mashup.

Beemer and Gregg (2009), also established two approach differences within the mashup creation: The passive approach is concerned with the creation of Web based browser plug-ins, and the proactive approach which focuses on the creation of advance interface integration. This process permits the generation of complex mashups, without the need of obscure programming technics, previously not accessible to non-programmers. Structured mashups should include in their design data models, views,

and interaction controllers, their proliferation however, may set a challenge in the Web programming future, due to the lack of standardization and compatibility in some created mashup instances.

In accordance with Liu, X., Hui, Y., Sun, W., and Liang, H. (2007), mashups are an ad hoc composition which has become a hallmark of Web 2.0, attracting industry and academia, which can be created using a web browser in a "drag and drop" fashion. The components related to mashup technology architecture have been defined as follow: the API/content providers which deal with the content provisions, the mashup hosting site specific location, and the consumer's Web browser (Liu, X., et al.,2007).

2.5 APIs and Mashups: Plural Singularity

APIs and mashups coexist with cloud computing, which is defined by National Institute of Standards and Technology (Mell and Grance, 2011), "As a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources, that can be rapidly provisioned and released with minimal management effort or service provider interaction".

APIs and mashups relate to the cloud computing model, through the process of creating new and distinct web services previously not available in the Web (Marston, Li, Bandyopadhyay, Zhang and Ghalsasi, 2011).

2.6 APIs/Mashup Technology

In the words of Jhingran (2018), ecosystems represent a future in which companies leverage their core strengths, while using assets from other providers to drive growth. The developer community assets that must be in place according to Boyd (2017) are: to first clear terms of service for the API, engage in proper documentation, code snippets and code for a sample application, create a self-serve API registration, responsive error

messages in place, maintain an API specification format, a sandbox environment and finally an API uptime status page.

The steps the most successful companies follow when implementing an API strategy according to Digital McKinsey's report (2017), are: identification, prioritizing its value, manage monetization actively, create a centralized governance and organizational model, and drive usage and adoption to gain scale standing.

Mashups was a term coined from the musical industry as mentioned earlier, representing the creation of a new song derived from the mixture of others (Merrill, 2006). It is the result of technology compatibility, Web services and data provider's willing participation. The two active ingredients for Web mashup development are the combination of data (content) and APIs. It is also a communication form to compose, combine, assimilate and appropriate elements from existing works to create an original product (Jackson, 2009). A Web application that combines information from numerous publicly available sources using web services, RSS feeds or other means (Marjit and Jana, 2009).

Grammel & Storey (2008), developed structure themes to measure consumer development, which are: the levels of abstraction, computer knowledge requirements, reducing the learning curve requirements, community support to mashup developers, the re-use of mashup application components, appropriate user interfaces creation, security and correctness awareness.

2.7 Conceptual Framework

The conceptual framework of this study describes the process involved in the analysis of the content dimensions adhered to APIs within the interaction of companies, SDKs, mashups and developers and its effect on business functions delineated by

categories under the mixed method research. Through content analysis, these dimensions adhere to the value the data collected provides and are expected to underline the behavior and tendencies of the APIs, SDKs, mashups and developers. (Johnson and Christensen, 2000).

2.8 Research Model

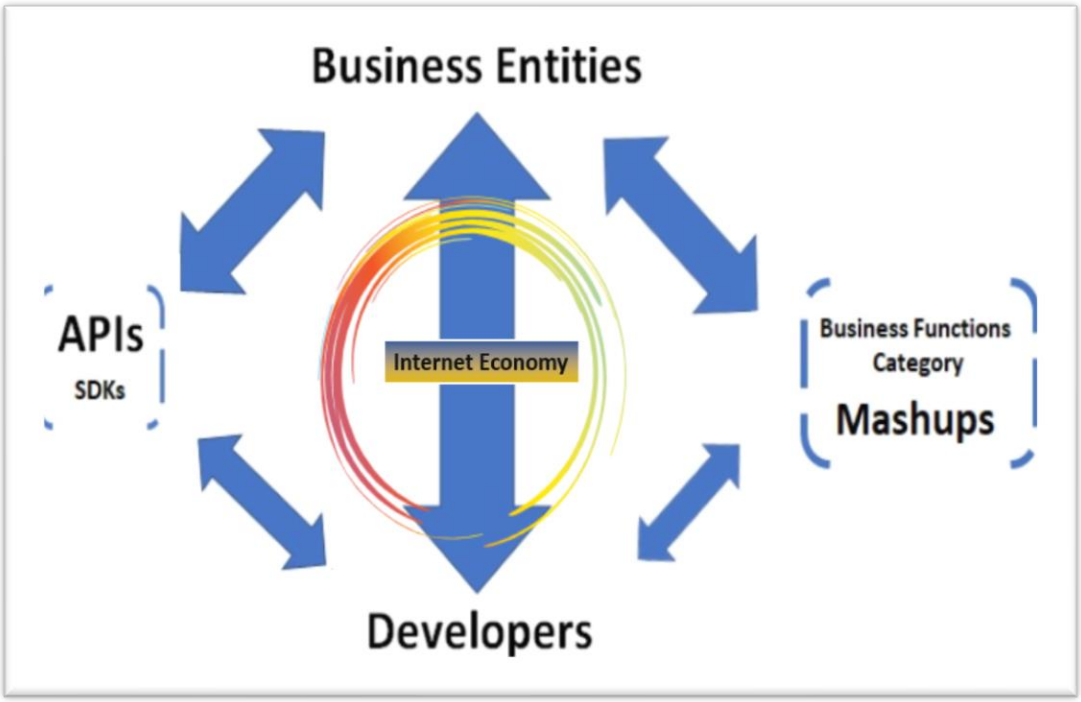


Figure 1. APIs Players and Dynamics Model

APIs Players and Dynamics Model (figure 1): Business entities contribute with the creation of APIs and business functions with the participation of developers. APIs are the raw materials needed as building blocks for the Internet economy, with the complementary SDK element. Developers are the driving force that convey the business functions needs through innovative programing and apps combinations. As a result, business entities receive feedback and evaluate realized business functions through mashups and other apps configurations.

2.8.1 Variables

APIs - Represent the public API units available which are maintained by each organization and are the legal owners of said intellectual property.

SDKs -This resource toolkit provides the key elements for developers to take advantage of the API's capability.

Business Entities – Companies that are maintaining active API presence within the Web-environment.

Developers - Persons whom possess the necessary skill to produce a mashups composition regardless of its simplicity.

Business Functions or Categories - The categories selected for this study have been established by www.programmableweb.com, which structures the different type of functions the APIs, mashups, SDKs perform. The value of the category component relies on how it divides the business environment over the Internet.

Related Fields - Articles, libraries and followers were not included in the research model.

2.9 Summary

APIs and mashups are driving the Internet economy by way of interconnecting content with functionality. New instruments designed by companies are facilitating and enhancing user's experience, to include the opportunity to develop and share new applications throughout the Web, as described throughout this literature review.

CHAPTER III

METHODOLOGY

3.0 Introduction

APIs serve developers in the creation of mashups using the combination of existing disparate Web sources such as content and functionality. This investigation focused on how developers used existing APIs for mashup creation, and how this action impacted the API developing businesses.

The APIs and mashup technology trend has been evaluated through the lens of the mixed method research framework. This mechanism was ideal, to analyze the impact the APIs and mashups had over the business entities based on their categories (Schumacher, S., and McMillan, J. H. , 2006).

3.1 Purpose of the Study

While evaluating the innovative web-based services used in the Internet, researching the benefits provided by APIs, was the logical path to follow. Throughout the software development history, IT systems for businesses fail at times to satisfy consumer needs and expectations, providing therefore, an open door for rejection and dissatisfaction.

Developers have been empowered by APIs, by facilitating the creation of sophisticated Web-based processes or functions (category type), without the need of specialized programming knowledge, to create simple or complex mashups. This study focused on said interactions.

3.2 Scope of the Study

This study concentrated on the use of publicly available information from the Internet in relation to APIs and mashup technology, guided under secondary data, content tabulation and social network analysis. Also, it addressed how developers advance the Internet economy by using APIs for mashups creation within established categories.

3.3 Limitations of the study

Due to the novelty of mashup technology, there were limited academic published references addressing the developer's behavior, performance or satisfaction aspects in relation to APIs or mashup within Web-based environment. There was however, a high volume of technical papers, addressing plug-ins technicalities and proprietary APIs platforms. Despite this drawback, the academia, media and private sectors have created API related documentation, which was used a reference for this study.

3.4 Research Design

The mixed method research framework was used by incorporating secondary data in tabulated, curated and standardized form. In the curating process, the data was configurated into fields where repeated items were deleted. These steps served as anchors for the preservation of internal and external validity (Sekaran, 2003).

This study abided to the collection of publicly available data throughout the Internet to include: website traffic values and ratings (Neale, Thapa and Boyce, 2006). These techniques eliminated the need of questionnaires for this research, external data from www.programmableweb.com was used to measure the relationships and tendencies between API categories, SDKs, developers and mashups (Johnson and Christensen, 2000).

In accordance to Schumacher, S., and McMillan, J. H. (2006), quantitative research maximizes objectivity through statistical modeling, controls and structure. Qualitative research designs, however, emphasize on data gathering on natural occurring phenomena usually in words forms instead of numerical values.

Schumacher, S., and McMillan, J. H. (2006), resolve the mixed method research design by combining quantitative and qualitative methods. The mixed method research for this study consisted of the following instruments:

3.4.1 Secondary Data Analysis

In accordance to Schumacher, S., and McMillan, J. H. (2006). secondary data analysis is based on the use of gathered data created by government and private entities, available throughout the Internet or through a formal written request (Johnson and Christensen, 2000). Website www.programmableweb.com, was used as our primary repository. This website has maintained a database of created APIs and mashups for a period of over 5 years. The segment studied for this research was selected from the category index provide by the website.

3.4.2 Content Analysis

In accordance to the Colorado State University's Writing Guide, content analysis is a tool used for determining the presence of certain words or concepts within a set of texts, to establish relationships and inferences. Similarly, content analysis can be used to analyze entity documentation to determine regulatory compliance, trends, and group differences (Texas State Auditor's Office Methodology Manual, 1995).

Based on the Texas State Auditor's Office Methodology Manual (1995), the following coding units are commonly used for content analysis:

Words: The smallest possible unit.

Themes: A simple sentence or single idea.

Character: Division of content by subject or object.

Item: Whole unit of the original content.

Space-and-time: Metric of space/time displaced by content.

This research followed the practical approach towards content analysis, established by Jones (2009) aside from the others aforementioned methods.

3.5 Data Collection

The data collected for this investigation was obtained from www.programmableweb.com, under the API directory. The featured APIs, categories, developers, mashups and SDKs and other related fields were selected for this study.

3.6 Data Analysis

The data collected was analyzed by dividing the components into the following fields: categories, APIs, developers, mashups and SDKs; the company field was derived from extracted data, and related fields, articles, libraries and followers used as well. To handle such data, Microsoft Excel Pivot function was used to group and explore indirect values of entities, Microsoft Access facilitated the relationship process between the groups, Tableau enhanced the visualization experience and furthered the exploration and evaluation of the data collected in conjunction with the numerical values.

Node XL Social Network Analysis plug-in displayed the associations and clusters between them with graphical and model presentations (Hansen, D., Shneiderman, et al., 2010).

3.7 Summary

This study followed the mixed method research framework, using secondary data with content analysis to understand the behavior of the APIs, mashup and developers under established category role, using historical data on the Web (Schumacher, S., & McMillan, J. H. , 2006).

A total of 4,633 instances were collected and divided into APIs, developers, mashups, categories and derived companies. Other pertinent data such as: SDK's, articles, libraries and followers were used. These instances represent the tabulated data entries stored at www.programmableweb.com.

Dedicated mashup repository website www.programmableweb.com, was used for this investigation's secondary data, in consonance with academic literature and procedures. Collected instances was filtered using content tabulation and social network analysis. No interviews or questionnaires were administered during this investigation.

CHAPTER IV

DATA ANALYSIS

4.0 Introduction

The API infrastructure is divided by public release (APIs of diverse functionalities given by major companies such as Google and Facebook), partners and proprietary software developed for internal use within the private sector (Boyd, 2014). This study used public API data from www.programmableweb.com, which has contributed with the advancement of the API knowledge. This site has been used by businesses and the academia a like to study the growth of APIs, ecosystems, and to measure the impact of APIs strategies among other investigations. The available data and history of APIs' behavior within this website has been invaluable for this research. That said, this single source provided all the necessary data needed to understand the APIs, developers, mashups and business entities behavior within categories.

4.1 Description of the Population

To analyze the API environment, it was necessary to focus on each component to determine their participation and behavior within the ecosystem. The players involved in this ecosystem were limited to categories, APIs, companies (derived value), developers, mashups and SDKs. Other related fields were collected to establish tendencies within the API phenomenon, these were: articles, libraries and followers. However, there were not considered main targets for this investigation. The time frame of this study was encapsulated in the year 2018. Historical data was evaluated and curated to avoid duplicates.

4.1.1 APIs Population Phase I

APIs were first selected by category. The categories were evaluated and measured in terms of the quantity and quality of the data it represented. The categories with the highest levels of participation in the API ecosystem were selected, provided they addressed with empirical data the values needed for the sample. Figure 2 displays the manual process of gathering data by way of content analysis.

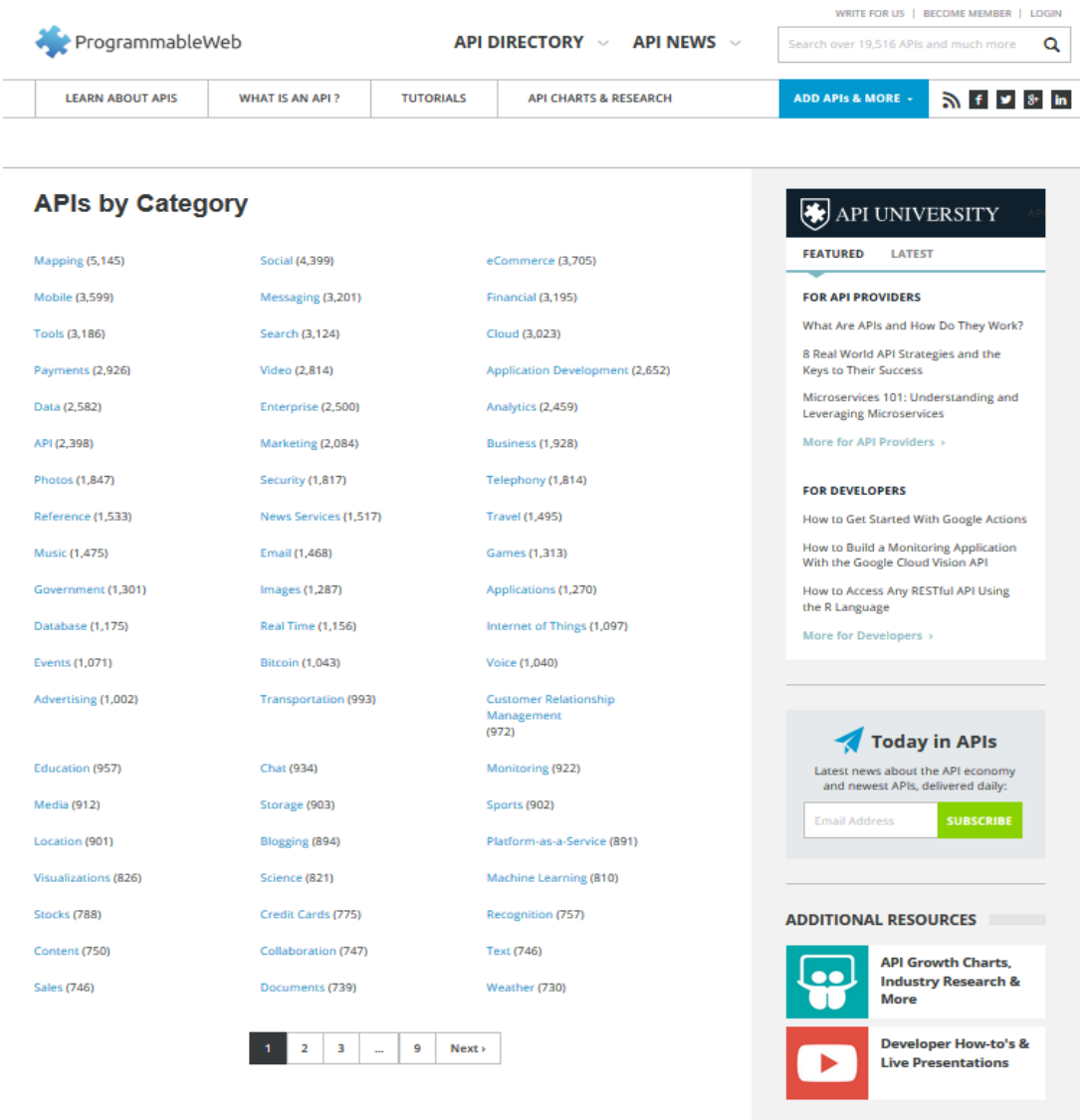


Figure 2. APIs Population Phase I

4.1.2 APIs Population Phase II

Top APIs were selected within the directory provided in the website menu. After gathering the main categories, the next step was to tally the APIs that were active or provided a significate historical participation within the APIs' ecosystem that could explain tendencies or relationships. Figure 3 depicts this content analysis process.

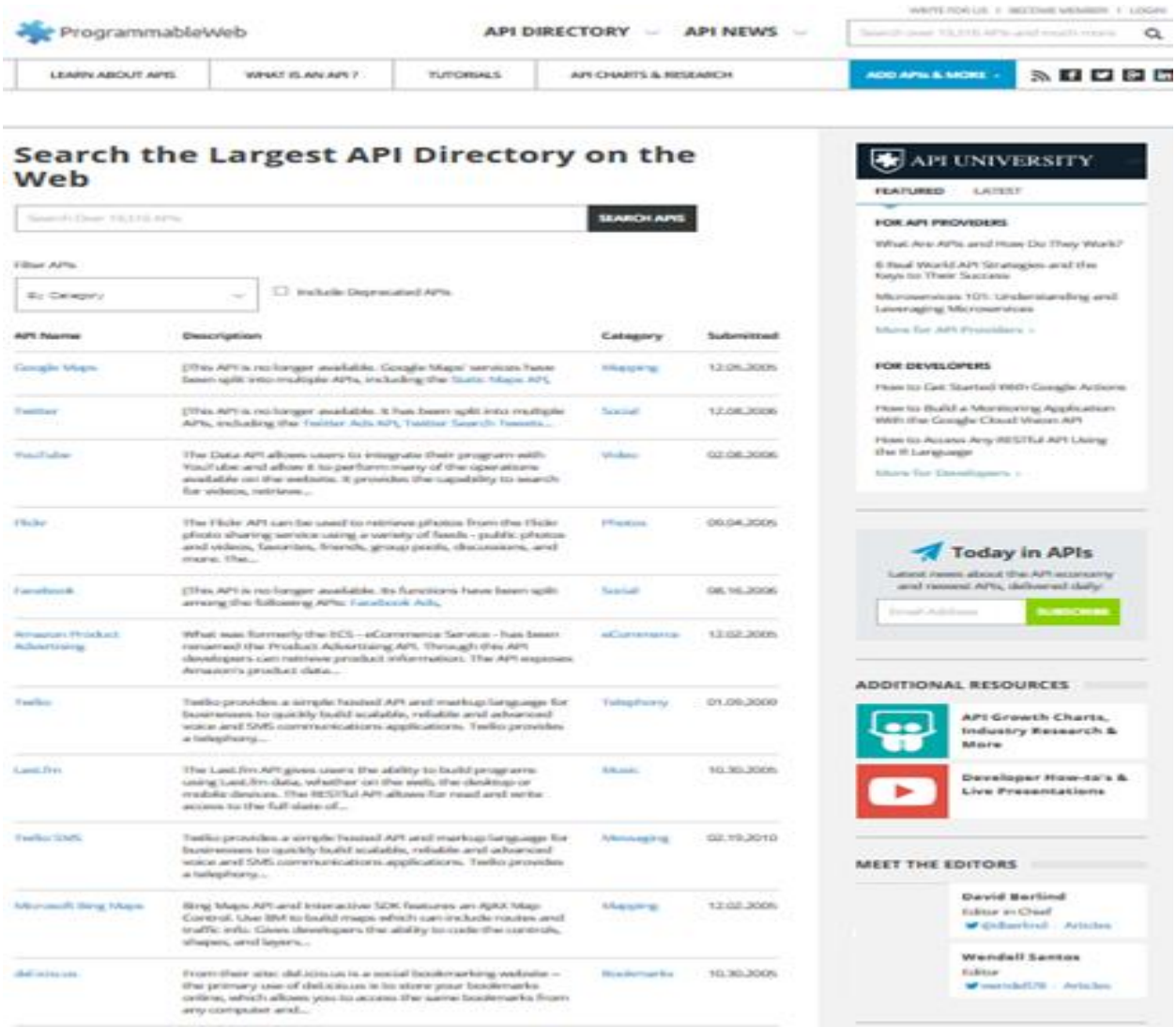


Figure 3. APIs Population Phase II

4.1.3 APIs Population Phase III

After selecting the top APIs, they were individually evaluated based on their statistical performance. The particularities of each API used in this study is addressed in the Appendix section. The APIs selected were filtered by the statistical data bank provided in their dossier at the time of the survey. Figure 4 illustrates the fields associated with the number of SDKs, articles and the like that constitute the APIs' DNA.

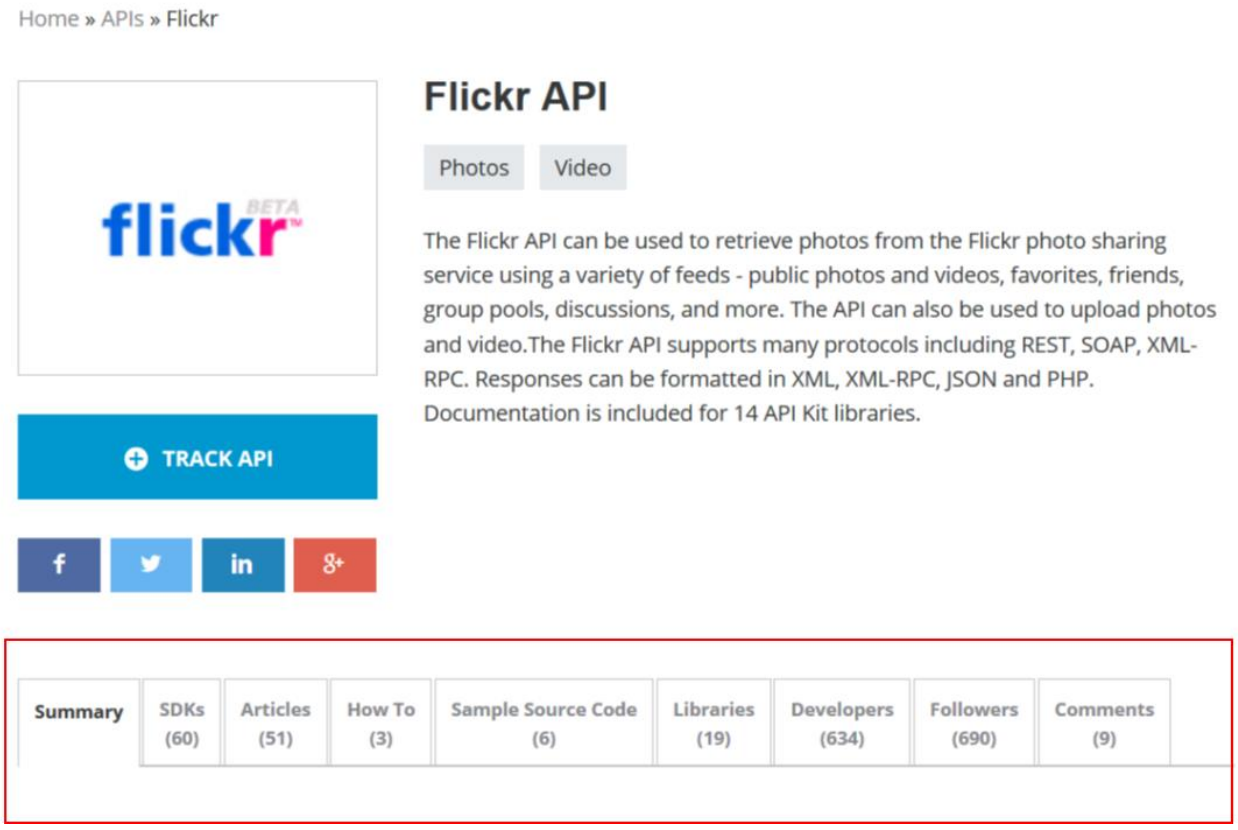


Figure 4. APIs Population Phase III

4.1.4 APIs Population Phase IV

With content analysis, it was discovered that the developers and mashups were linked together as part of the indexed designed by the website. After surfing each dimension, the elements involved were tabulated, organized, curated and process for sampling purposes. Pictures, usernames or names of persons were omitted. Comments associated with used website blog, were not considered for this survey. Under the selected API statistical bank, developers and mashups were found and collected using the website's section as depicted in figure 5.

Figure 5. APIs Population Phase IV

4.1.5 APIs Population Phase V

The SDKs values were obtained by selecting the category criteria in the website menu. The SDKs values were collected as part of the overall picture within the API ecosystem, by using the APIs' specific SDKs values attributed in their portfolio, as shown in figure 6.

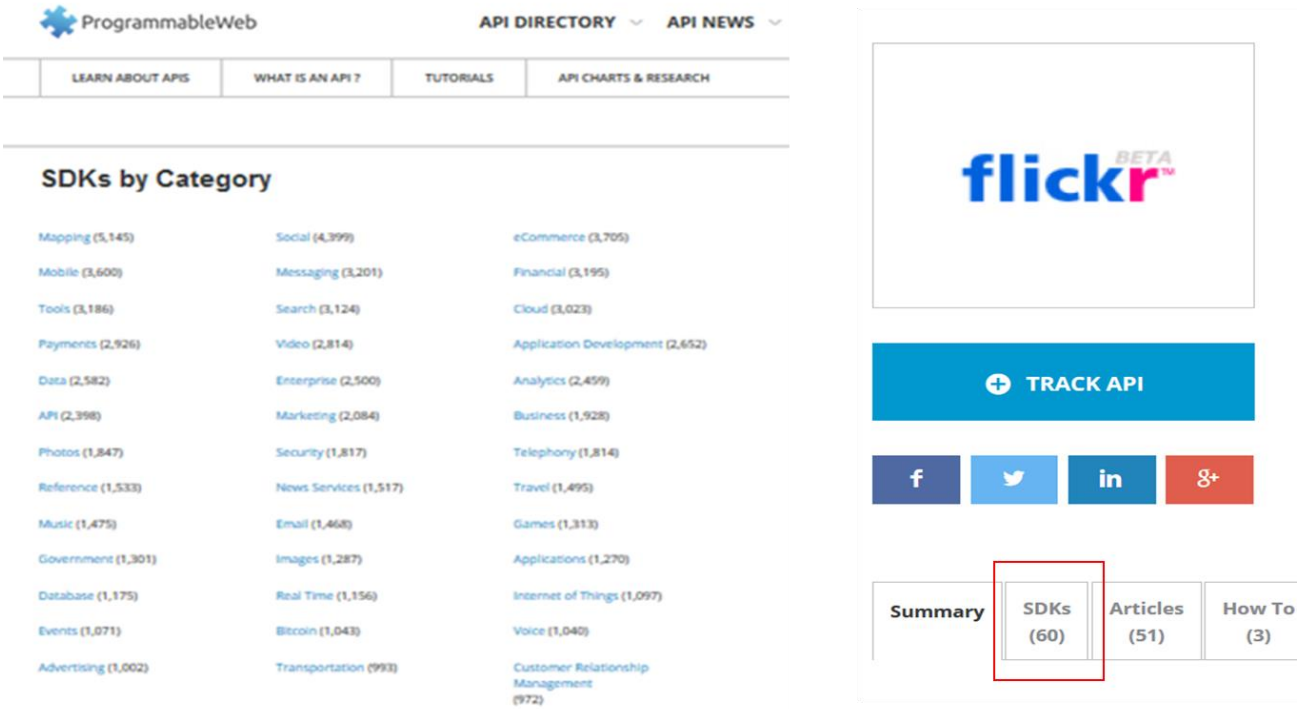


Figure 6. APIs Population Phase V

4.1.6 APIs Population Phase VI

To capitalize the data provided by the website, other variables were considered for referential purposes such as, followers, articles and libraries using the API's statistical bank, created by the website as shown in figure 7.

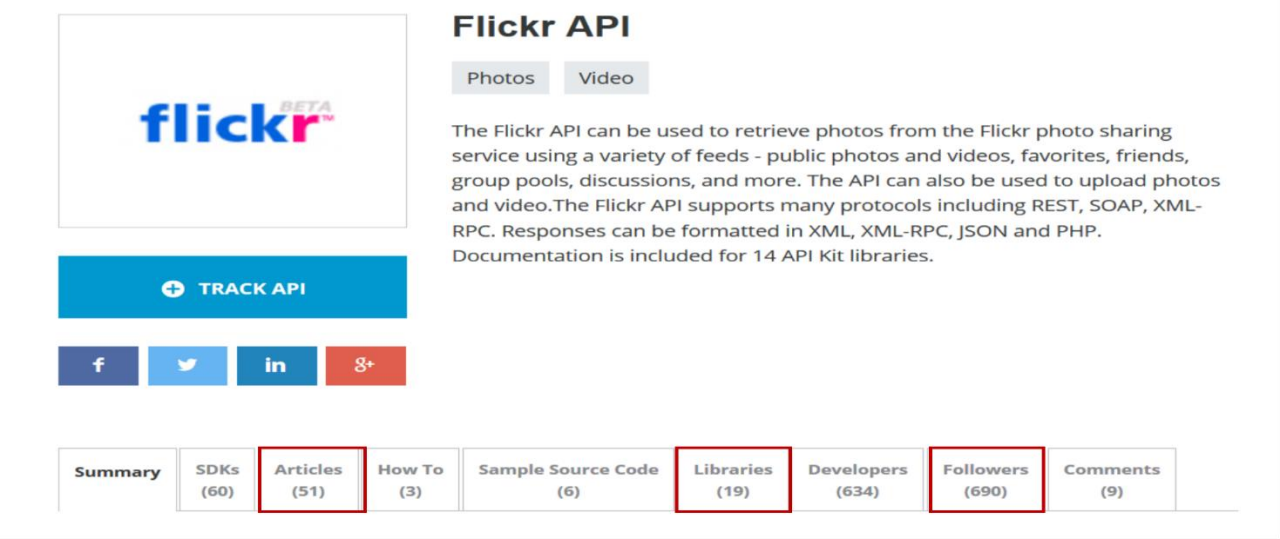


Figure 7. APIs Population Phase VI

4.1.7 APIs Population Statistics

The following information was obtained through the website regarding the population used: A total of 482 categories were counted, with (19,279) APIs, developers (-), 7,909 mashups, 11,417 SDKs, articles (-), libraries (1,676), followers (-) and companies (-) as shown in table 1.

Table 1. Population Values

Type	Quantity
Company	-
Category	482
APIs	19,279
Developers	-
Mashups	7,909
SDKs	11,417
Followers	-
Libraries	1,676
Articles	-

4.2 Sample

The sample data values collected and used to analyze the APIs' behavior was in terms of categories and business type. Content and clustering analysis was the exploratory method used in conjunction with a correlation test. The data was collected as factual and historical, as previous published academic articles have done, including the written permission for use from the editor in chief. The webpage www.programmableweb.com has maintained high standards and integrity in terms of the capturing of APIs' functionality and related information throughout the years. The data collection processed included the digital extraction of tabulated instances, picked

by way of searching multiple webpages, analyzing their components to maintain consistency. organizing and structuring the elements involved to gather all the necessary fields for this research.

A total of 21 categories (curated), with 29 types of APIs (curated), followed by 2,434 developers (curated), a sum of 4,608 mashups (curated), 274 SDKs pertaining each API was captured, a complementary set of articles (769), libraries (241), followers (11,403) and derived companies (19) values were used. The sample collected is distributed in the following manner: the APIs variables are categorical in nature, developers address 12%, mashups or resources 23%, SDKs 2%, articles with 4%, libraries 1% and followers with 58%, as shown in table 2.

Table 2. Sample

Type	Quantity
Company	19
Category	21
APIs	29
Developers	2,434
Mashups	4,608
SDKs	274
Followers	11,403
Libraries	241
Articles	769

4.3 Data Collection

To discover new insights within the APIs' ecosystem, it was necessary to accept the established categories marked by the www.programmableweb.com. The data collected was filter and curated using MS Excel, Notepad, MS Access, NodeXL Pro plug-in, Tableau and other website statistical tools. The personal data provided by captioned website regarding the name or username of the developers and mashups have been omitted. The data collected for this research did not required any type of human intervention, beyond the information provided by the entities or individuals who voluntary submitted the same through said website channels.

The business entity, category, SDKs and other values are exposed in full within the API it represents. The data was tabulated to incorporate the body of the variables involved to include relationships, statistical values and the framework needed for mathematical derivative analysis.

4.4 Data Analysis

Category Values: There are 21 categories in total, 15% are destined for video and photos, 11% for eCommerce and 8% for social and telephony. The remaining categories are shared with 6% or less in an even fashion.

APIs by Category: The API data collected shows a median of 91, a mean of 159 and a standard deviation of 168. The mayor players in terms of businesses are Google with 26%, flickr with 14%, Amazon with 13% and Twilio with 12%. However, Google leads is based on 24% of all the APIs evaluated, in comparison with flickr which only counts a merely 3%. Amazon shares 10% of the APIs studied.

The explanation of this outcome can be traced to the creation phase of the APIs' function and purpose, which cannot be overlook by any business entity, to avoid any stagnate effect on the market share of the APIs ecosystem. A Shapiro-Wilk normality test was performed resulting in data not normally distributed. The Kurtosis value is 4.6, in addition to the Skewness with a value of 2.26 as shown in figure 8.

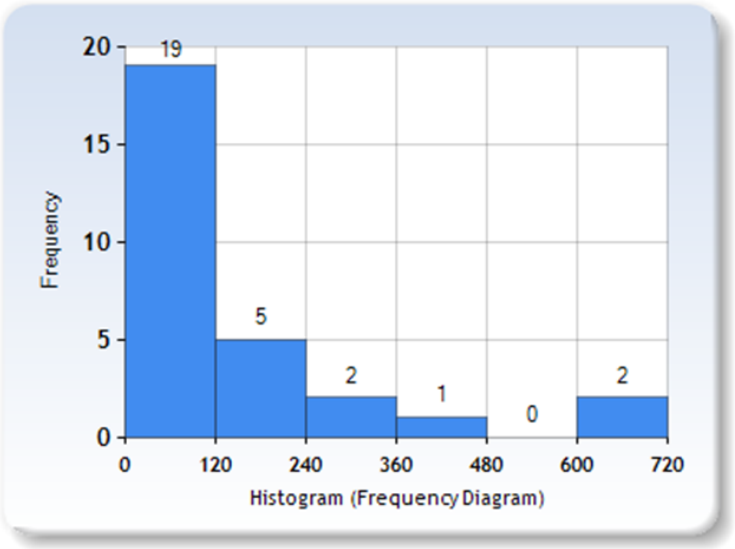


Figure 8. Distribution of API Data

Developers by Category: A derivative approach had to be implemented to access the relative values the developers contained, and their connection to the category field. Through content analysis, the developers were incorporated within each of the 29 APIs entailed in the collected sample. All duplicates were eliminated to avoid unnecessary redundancy within the same API and mashup. In terms of the data value the developers represent Google shares 31%, Amazon 14%, flicker 13% and Last.fm 7%. A Shapiro-Wilk normality test was performed resulting in data not normally distributed. The developer's data mean falls under 83, a median of 55, and a standard deviation of

94. The Kurtosis value is 7.6, in addition to the Skewness with a value of 2.74 as shown in figure 9.

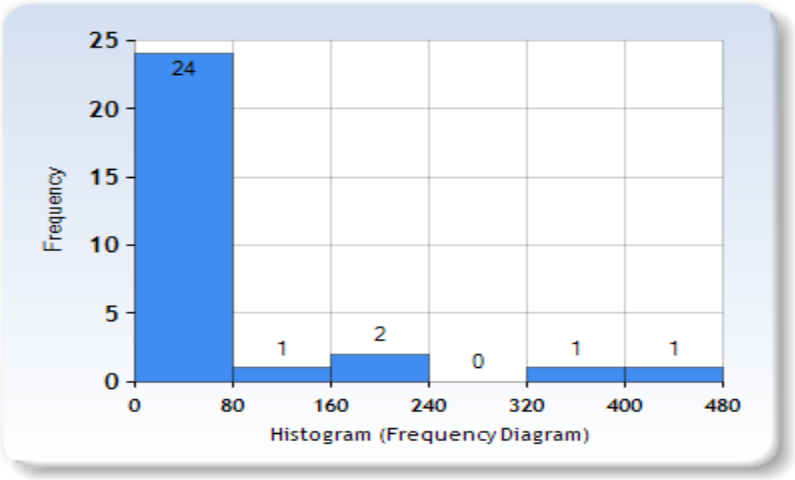


Figure 9. Distribution of Developers Data

Mashups by Category: To associate the mashups with the respective category, the values were derived using the developer's API link to the resource intervened or created. Only .0054 of the mashups data collected was removed due to duplication. Google's participation reaches 26%, flickr 14%, Amazon 13%, Twilio 11%, Microsoft, eBay and Last.fm 5%. A Shapiro-Wilk normality test was performed resulting in data not normally distributed.

The mashup's mean falls under 158, with a median of 88, and a standard deviation of 167. The Kurtosis value is 4.6, in addition to the Skewness with a value of 2.26 as shown in figure 10.

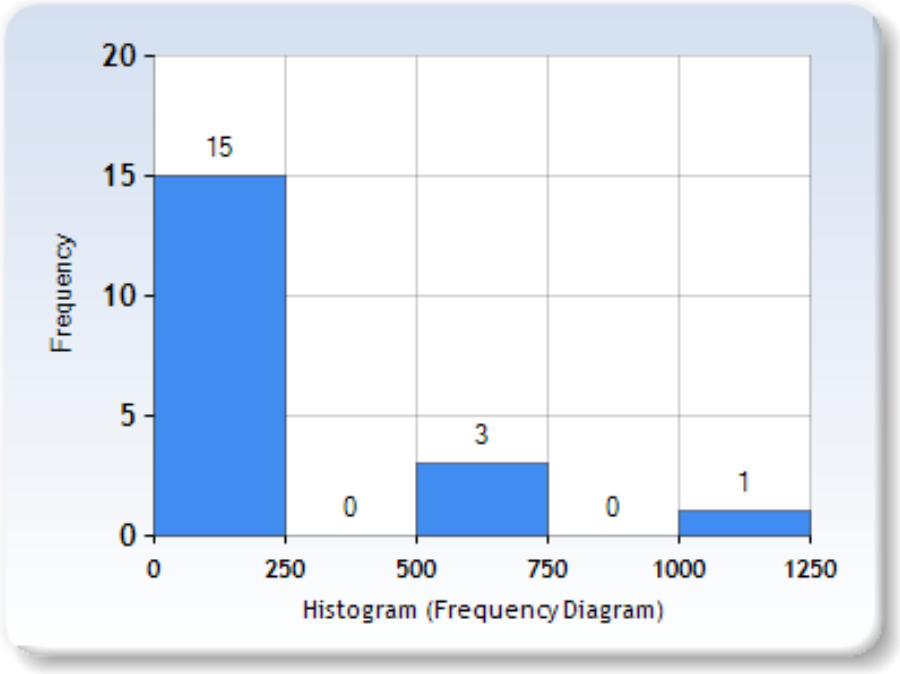


Figure 10. Distribution of Mashups Data

SDKs by Category: SDKs' data was obtained through the APIs' statistical bank. The values were accepted without any data cleansing necessary. In terms of the data values, flicker obtained 22%, Facebook Graph 19%, Google 12% and Yelp 8% among the top shares for the SDKs enclosure.

The mean was established as 9.44, a median of 4 and a standard deviation of 14.33. A Shapiro-Wilk normality test was performed resulting in data not normally distributed. The Kurtosis value is 7, in addition to the Skewness with a value of 2.61 as shown in figure 11.

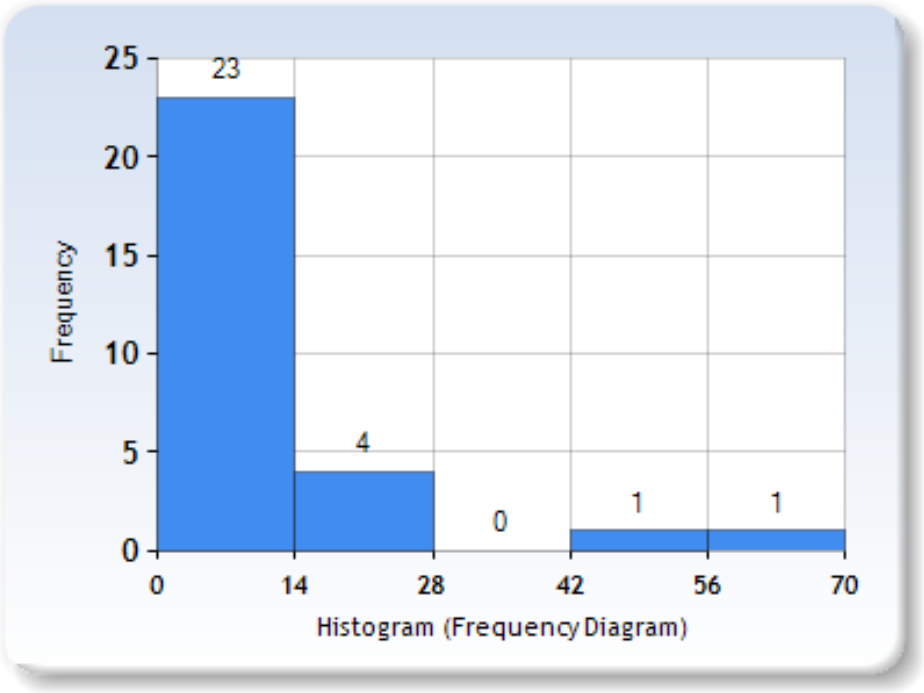


Figure 11. Distribution of SDKs Data

Articles, Libraries and Followers by Category: The data pertaining articles, libraries and followers were obtained by the APIs' statistical bank as is, with no further data processing required beyond standard tabulation. The values associated with the extracted articles data are comprise of Google 21%, Twilio 16%, Amazon 10%, Foursquare 9% and Facebook Graph 8% as the main top percentages. The mean was established as 26, a median of 15 and a standard deviation of 25. A Shapiro-Wilk normality test was performed resulting in data not normally distributed. The Kurtosis value is 1.49 in addition to the Skewness with a value of 1.32 as shown in figure 12.

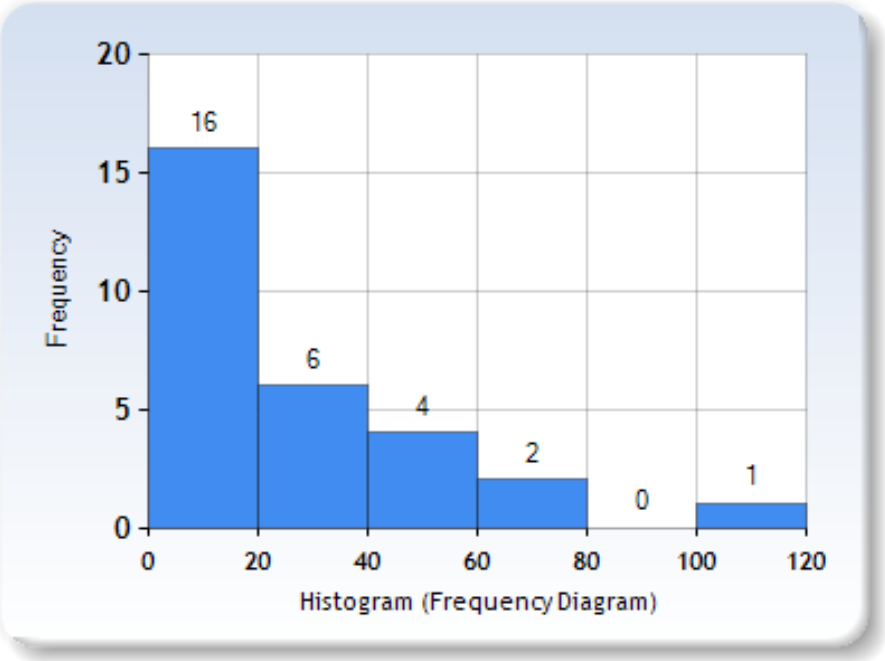


Figure 12. Distribution of Articles Data

The values pertaining the data extracted for libraries falls under Google 15%, Twilio 14%, Foursquare and Amazon with 10%, Flickr and Last.fm with 8% among the top values. The Libraries mean value was established as 8.3, a median of 6 and a standard deviation of 8.75. A Shapiro-Wilk normality test was performed resulting in data not normally distributed. The Kurtosis value is 1.18, in addition to the Skewness with a value of 1.21 as shown in figure 13.

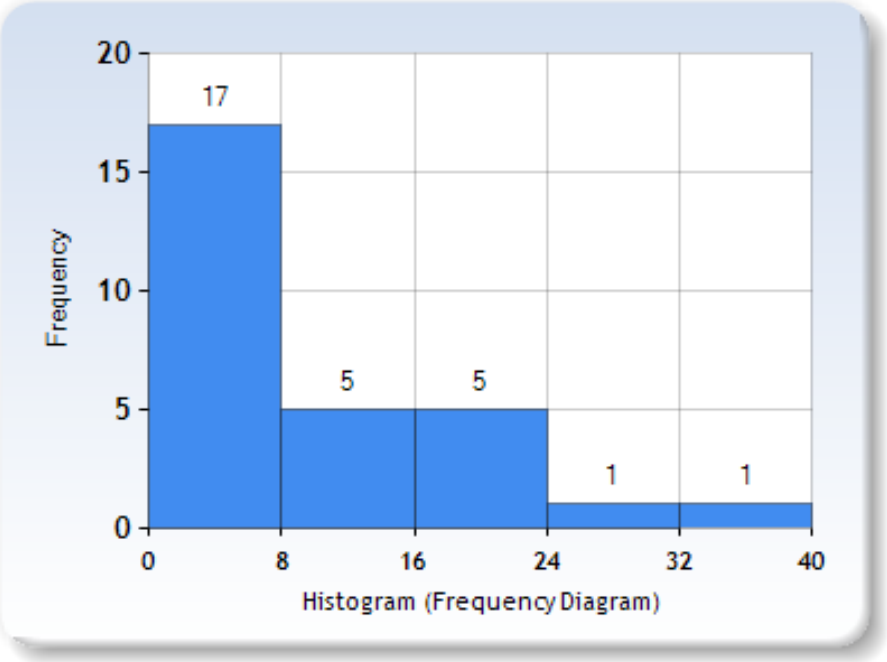


Figure 13. Distribution of Libraries Data

The values connected to followers fall under Google 22%, Amazon 12%, Linkedln 11% with the top percentage. The followers' field is considered important cause it reflects the interest of users and developers a like within the APIs ecosystem. The mean was established as 393, a median of 246 and a standard deviation of 336. A Shapiro-Wilk normality test was performed resulting in data not normally distributed. The Kurtosis value is 3.47, in addition to the Skewness with a value of 1.9 as shown in figure 14.

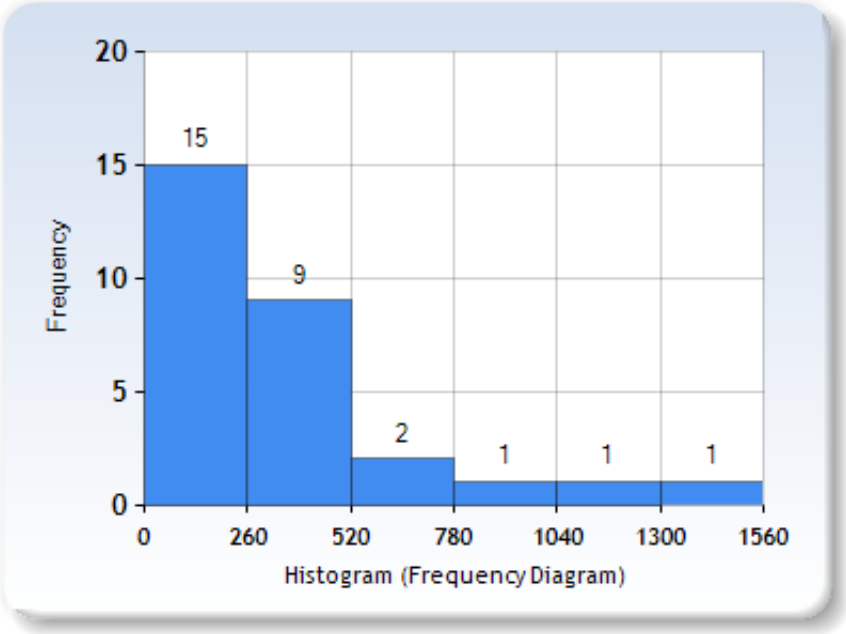


Figure 14. Distribution of Followers Data

4.4.1 Content Analysis

Content analysis was the catalyst instrument used to evaluate, gather and organized the data for this investigation. This technique was performed by systematically analyzing text within the subscribed webpage and incorporated the same into a coded form, feasible for computation and statistical scrutiny. After structuring the data to its main components, it was necessary to conduct reliability and validity statistical test.

This mixed method research used spelling and grammar check protocols, redundancy evaluations, mathematical processing for the conversion of qualitative values into numerical form, the Shapiro-Wilk Normality test and data transformation methods to soften the curve properties of data sets when not in compliance with the normal distribution form.

4.4.2 Correlation

There is a strong correlation between APIs/developers, APIs/mashups, and APIs/followers. There is a strong correlation between developers/mashups and developers/followers. There is also a strong correlation between mashups/follows and mashups/articles. And there is a strong correlation between libraries/articles, no correlation was found for the SDKs variable shown in table 3.

Table 3. Correlation of Variables

Correlations								
		API	Developers	Mashups	SDKs	Followers	Libraries	Articles
API	Pearson Correlation	1	.929**	1.000**	.336	.646**	.467*	.465*
	Sig. (2-tailed)		.000	.000	.075	.000	.011	.011
	N	29	29	29	29	29	29	29
Developers	Pearson Correlation	.929**	1	.929**	.354	.713**	.366	.347
	Sig. (2-tailed)	.000		.000	.060	.000	.051	.065
	N	29	29	29	29	29	29	29
Mashups	Pearson Correlation	1.000**	.929**	1	.335	.646**	.466*	.466*
	Sig. (2-tailed)	.000	.000		.076	.000	.011	.011
	N	29	29	29	29	29	29	29
SDKs	Pearson Correlation	.336	.354	.335	1	.196	.160	.264
	Sig. (2-tailed)	.075	.060	.076		.307	.408	.166
	N	29	29	29	29	29	29	29
Followers	Pearson Correlation	.646**	.713**	.646**	.196	1	.504**	.390*
	Sig. (2-tailed)	.000	.000	.000	.307		.005	.036
	N	29	29	29	29	29	29	29
Libraries	Pearson Correlation	.467*	.366	.466*	.160	.504**	1	.691**
	Sig. (2-tailed)	.011	.051	.011	.408	.005		.000
	N	29	29	29	29	29	29	29
Articles	Pearson Correlation	.465*	.347	.466*	.264	.390*	.691**	1
	Sig. (2-tailed)	.011	.065	.011	.166	.036	.000	
	N	29	29	29	29	29	29	29
**. Correlation is significant at the 0.01 level (2-tailed).								
*. Correlation is significant at the 0.05 level (2-tailed).								

4.4.3 Cluster Analysis

Cluster analysis was performed using the Excel NodeXL template to determine if any grouping existed within the acquired data. For each data group the vertices and edges were defined to connect the network structures for the visualization process. The Harel-Koren Fast Multiscale and Grid layouts were used to display the results. Figure 15 and 16 respectively demonstrates the connection between parent companies and APIs in addition to categories.

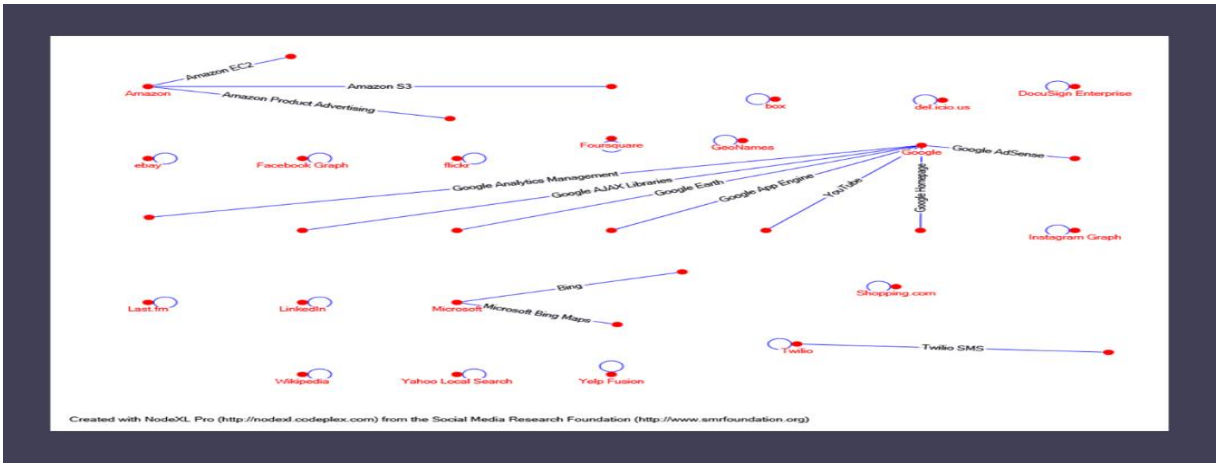


Figure 15. Parent Companies and APIs

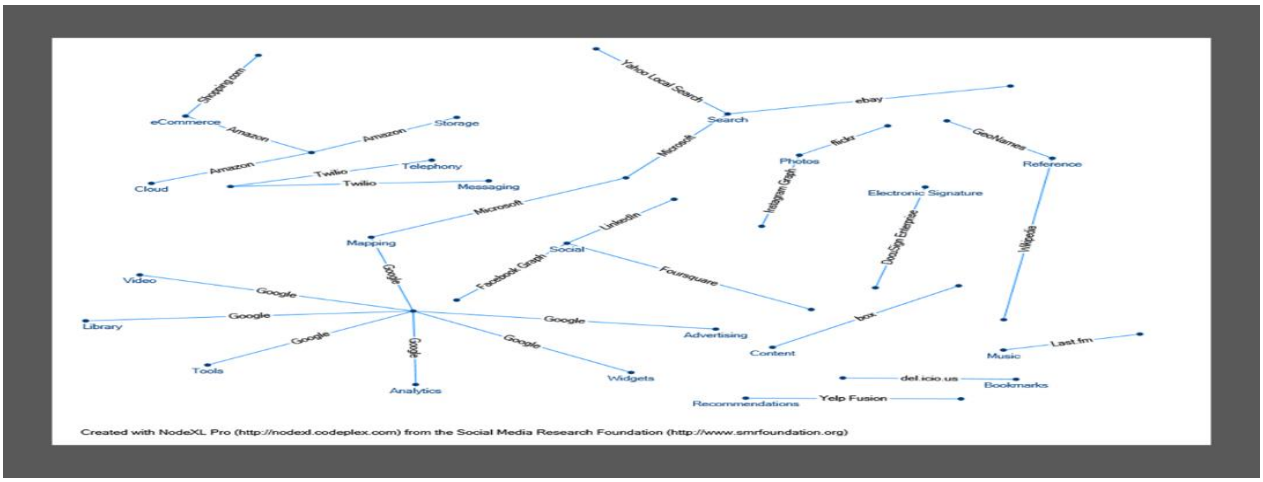


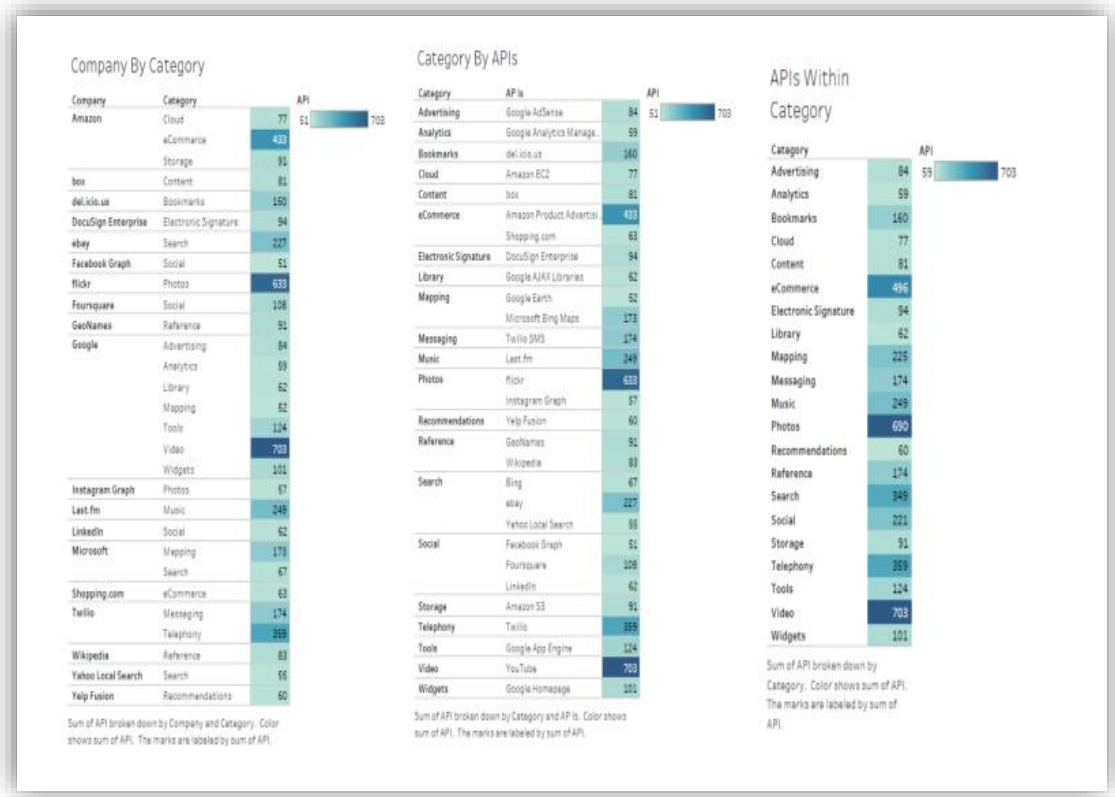
Figure 16. Categories and Parent Companies

4.5 Research Questions

The questions set forward are entangled with the APIs-Category condition, necessary to evaluate from a distance how the API resource leads the Internet economy. The content analysis and mixed method research provided the adequate tools to analyze the behavior of the APIs, within the developers, mashups and SDKs creation process under the category field boundaries, and its impact within the business environment.

1. Which are the most prevalent API categories and tendency?

Table 4. APIs by Company and Category



The most prevalent API categories fall under: Google (Videos) 26%, flickr (Photos) 14%, Amazon (eCommerce) 13%, Twilio (Telephony) 12% shown in table 4.

The tendencies for the APIs categories are moving towards direct user experience instead of procedural modules, which are more suited for developers; values were extrapolated from captioned table 3. The mobility factor cannot be over look since, the APIs under the category of telephony and messaging are becoming more in demand. The solid paced of APIs categories dealing with mapping, search and social are becoming more stable, since these technologies have been in play throughout the Web for a relative long period, as shown in figure 17.

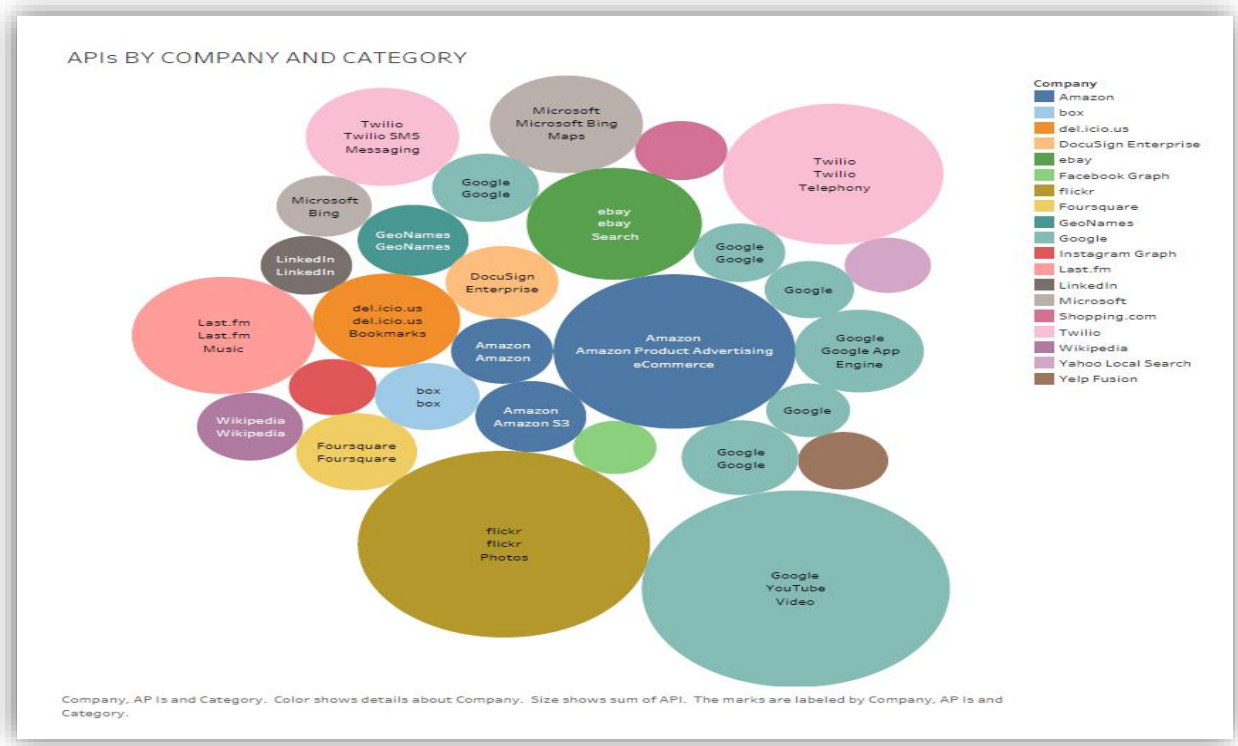
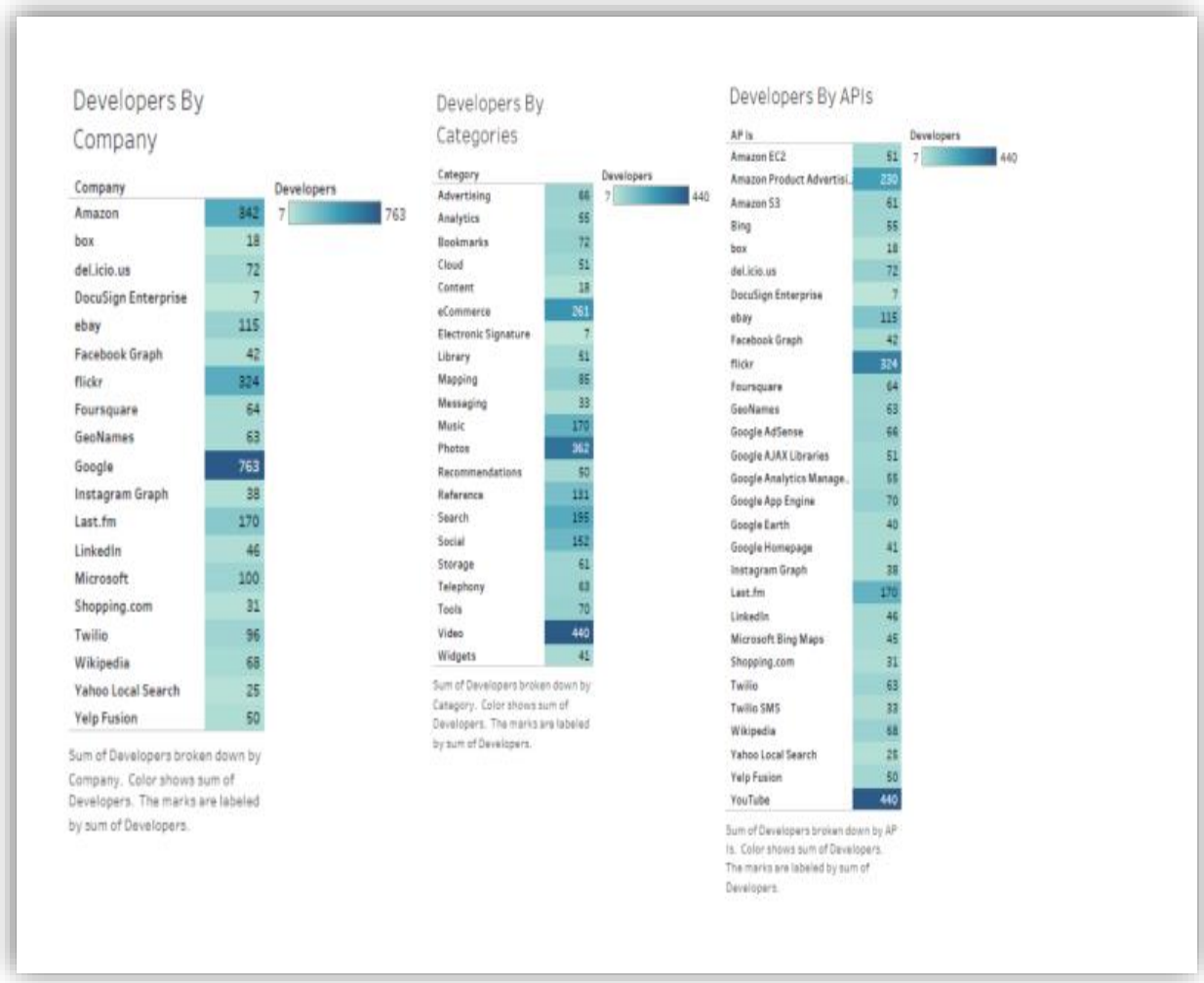


Figure 17. *APIs by Company and Category*

2. Which APIs are used the most by developers, and which companies predominate?

This study directly associated APIs with developers and companies, by calculating each unique instance and curating their values to avoid duplicates. The top APIs used the most by developers, fall under Google (YouTube) 18%, flickr (flickr) 13%, Amazon (Amazon Product Advertising) 9% and Last.fm (Last.fm) 7% as shown in table 5.

Table 5. APIs by Company/Category and Developers



The companies predominating the APIs ecosystem in terms of developers fall under Google 31%, Amazon 14%, flickr 13% and Last.fm 7%. The businesses dealing with social and eCommerce API categories have become favorites for developers. The API Last.fm, under the music category is becoming attractive for developers and users by allowing them to create a personalized listening experience as seen in figure 18.

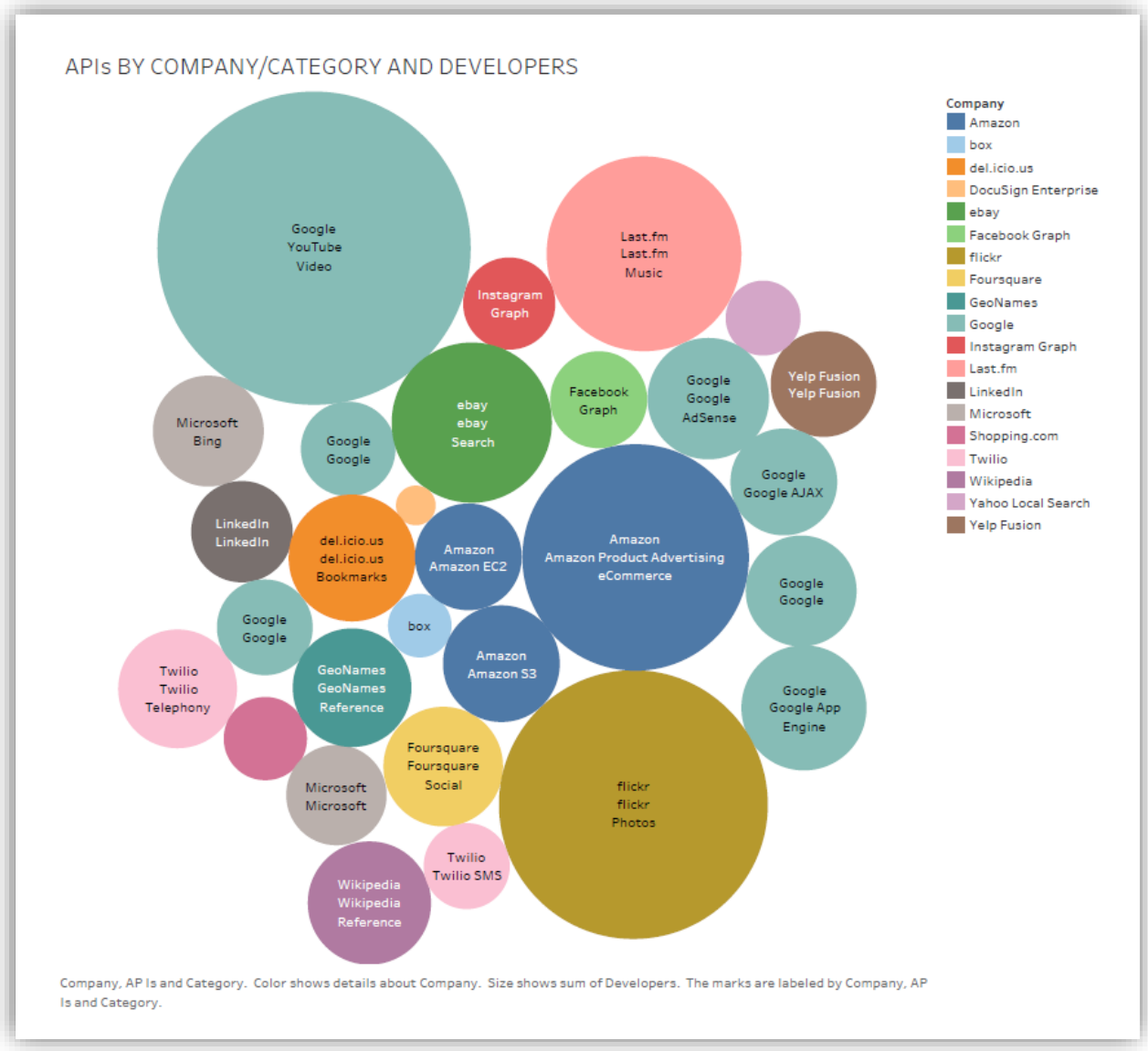


Figure 18. APIs by Company/Category and Developers

Additionally, a Social Network Analysis was conducted and depicted developers connecting to businesses were Google 31%, Amazon 14%, flickr 13% lead in this cluster model as shown in figure 19.

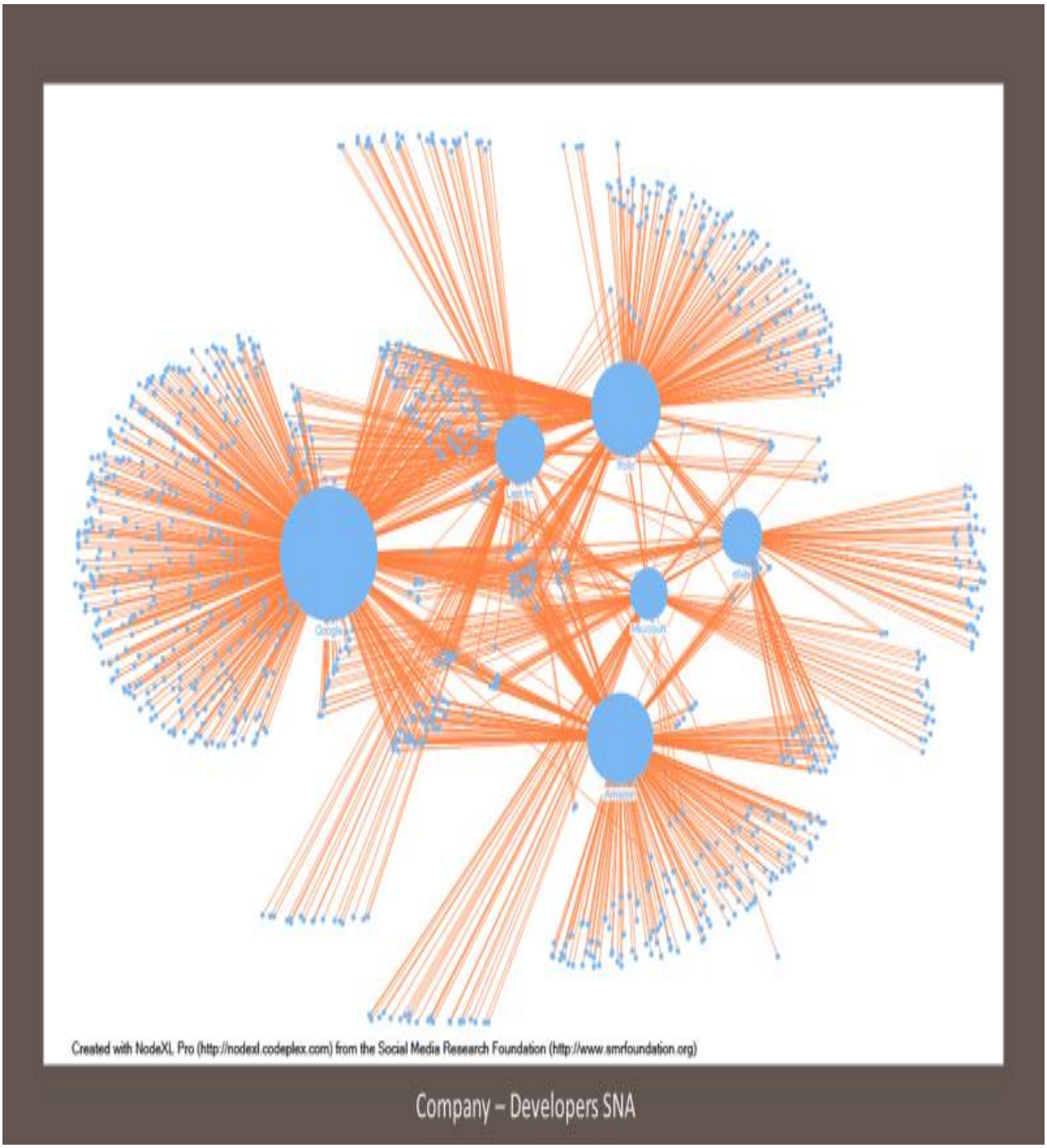


Figure 19. Company-Developers SNA

3. What are the similarities or differences between APIs and SDKs within category? Does any relationship exist between them?

The common categories for APIs and SDKs fall under eCommerce, video, music, reference, social, and photos. The differences between them fall under bookmarks, library, and widgets among others as shown in table 6. This comparison addresses the need for businesses to generate SDKs for the deprived captioned areas, where developers may become interested, especially in the telephony category which carries the element of mobility. No direct relationship exists between the creation of APIs and SDKs for the categories analyzed at this moment beyond the historical captured instances, perhaps in the future business will include as part of their API strategy the creation of SDKs.

Table 6. API/SDK Similarities and Differences

Category	SDKs	Category	API
Photos	75	Video	703
Social	62	Photos	690
Recommendations	22	eCommerce	496
Reference	21	Telephony	359
Content	18	Search	349
eCommerce	11	Music	249
Electronic Signature	10	Mapping	225
Music	10	Social	221
Video	10	Messaging	174
Advertising	9	Reference	174
Tools	8	Bookmarks	160
Mapping	5	Tools	124
Messaging	4	Widgets	101
Search	4	Electronic Signature	94
Analytics	2	Storage	91
Storage	2	Advertising	84
Cloud	1	Content	81
Bookmarks	0	Cloud	77
Library	0	Library	62
Telephony	0	Recommendations	60
Widgets	0	Analytics	59

4. What are the similarities or differences between APIs and developers within category? Does any relationship exist between them?

The main similarities among APIs and developers fall under the categories of video (15%), photos (15%), and eCommerce (11%). The main differences reside in the categories of advertising (2%), analytics (1%), library (1%) and recommendations (1%) to name a few as shown in table 7.

A strong relationship exists between APIs and developers in the areas associated with the user's experience, instead of mechanized processes. This can be explained by the amount of developers, well of 50%, carrying out projects in captioned main categories.

Table 7. APIs/Developers Comparison

Category	Developers	Category	API
Video	440	Video	703
Photos	362	Photos	690
eCommerce	261	eCommerce	496
Search	195	Telephony	359
Music	170	Search	349
Social	152	Music	249
Reference	131	Mapping	225
Mapping	85	Social	221
Bookmarks	72	Messaging	174
Tools	70	Reference	174
Advertising	66	Bookmarks	160
Telephony	63	Tools	124
Storage	61	Widgets	101
Analytics	55	Electronic Signature	94
Cloud	51	Storage	91
Library	51	Advertising	84
Recommendations	50	Content	81
Widgets	41	Cloud	77
Messaging	33	Library	62
Content	18	Recommendations	60
Electronic Signature	7	Analytics	59

5. What are the similarities or differences between APIs and mashups within category? Does any relationship exist between them?

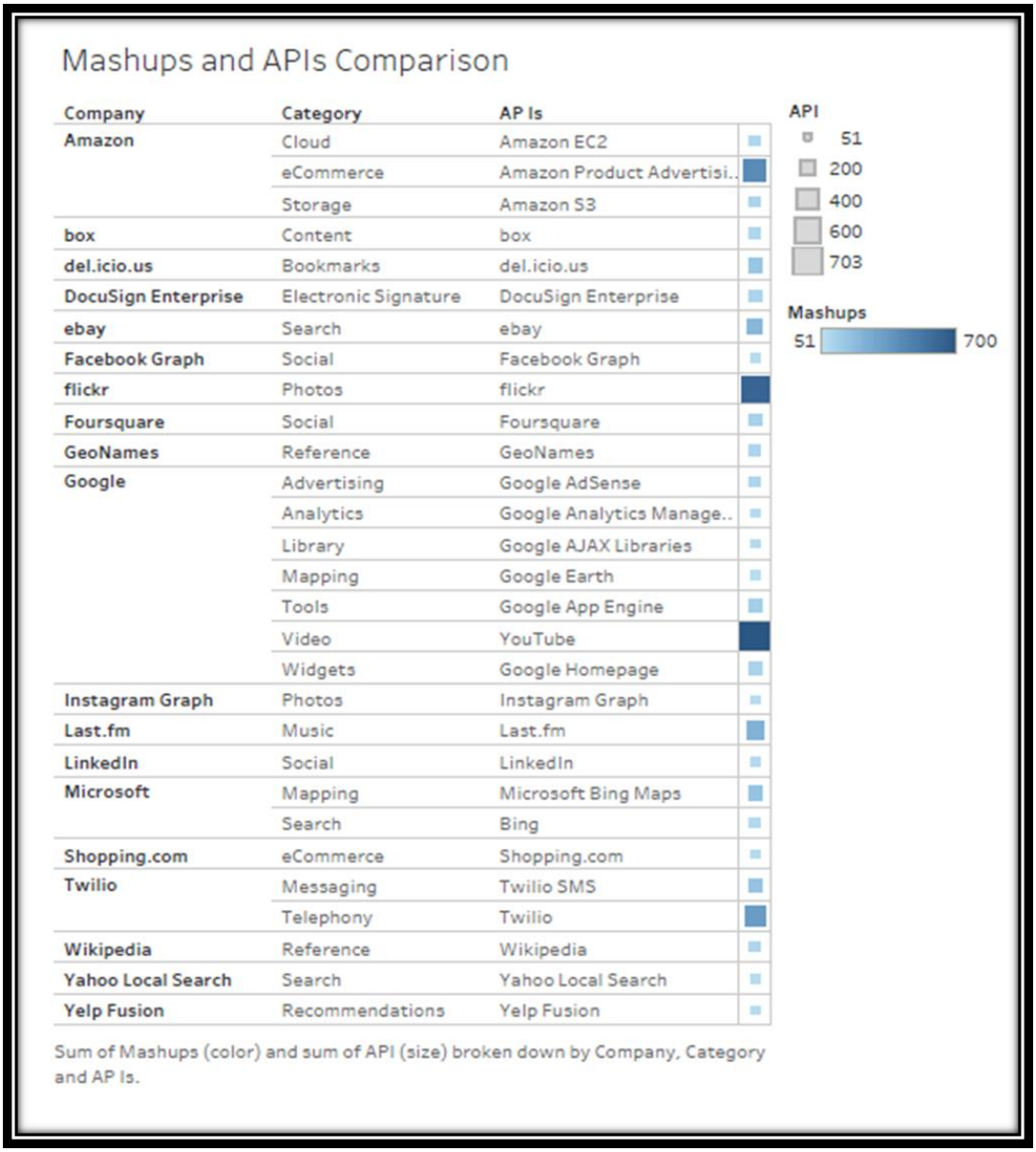
After analyzing the gathered data, the APIs and mashups were directly linked by way of the developers' mashups creating process. Except for a few repeated items, the data is practically the same as shown in table 8.

Table 8. Mashups/APIs Comparison

Category	Mashups	Category	API
Video	700	Video	703
Photos	686	Photos	690
eCommerce	494	eCommerce	496
Telephony	357	Telephony	359
Search	348	Search	349
Music	246	Music	249
Mapping	224	Mapping	225
Social	221	Social	221
Messaging	172	Messaging	174
Reference	171	Reference	174
Bookmarks	160	Bookmarks	160
Tools	124	Tools	124
Widgets	100	Widgets	101
Electronic Signature	94	Electronic Signature	94
Storage	91	Storage	91
Advertising	84	Advertising	84
Content	80	Content	81
Cloud	77	Cloud	77
Library	62	Library	62
Recommendations	59	Recommendations	60
Analytics	58	Analytics	59

To extract the detailed information describing the tendencies of the singular API within the respective company, an alternative tool was created composed of companies, categories and APIs. This action revealed that Amazon's (eCommerce), flickr (photos), Google (videos), Twilio (telephony) and Last.fm (music) were among the most used APIs within mashups, as shown in table 9.

Table 9. Mashups/APIs Comparison Under Company



6. How do APIs, developers, mashups behave within categories under the scope of social network analysis (SNA)?

The main three players within the APIs category SNA scenario are video, photos and eCommerce. Figure 20 shows the SNA depiction of category and API relationship.

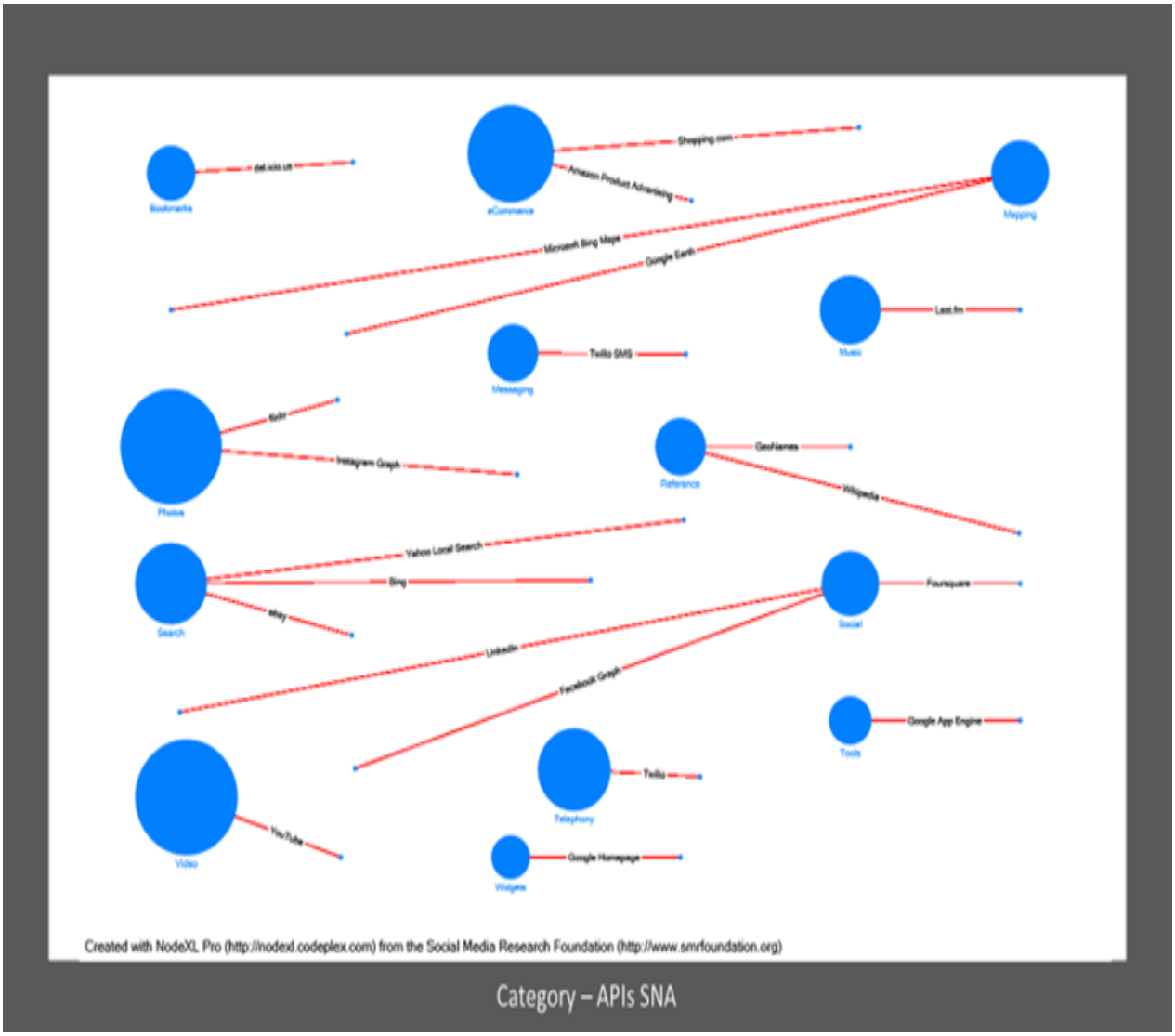


Figure 20. Category - APIs SNA

The developers SNA behavior is tied to the categories of videos, photos, eCommerce, music and search. In congruence with APIs – developers analysis,

captioned categories continue to reflect their value within the API ecosystem as shown in figure 21. This confirms that captioned categories still hold strong within the APIs ecosystem.

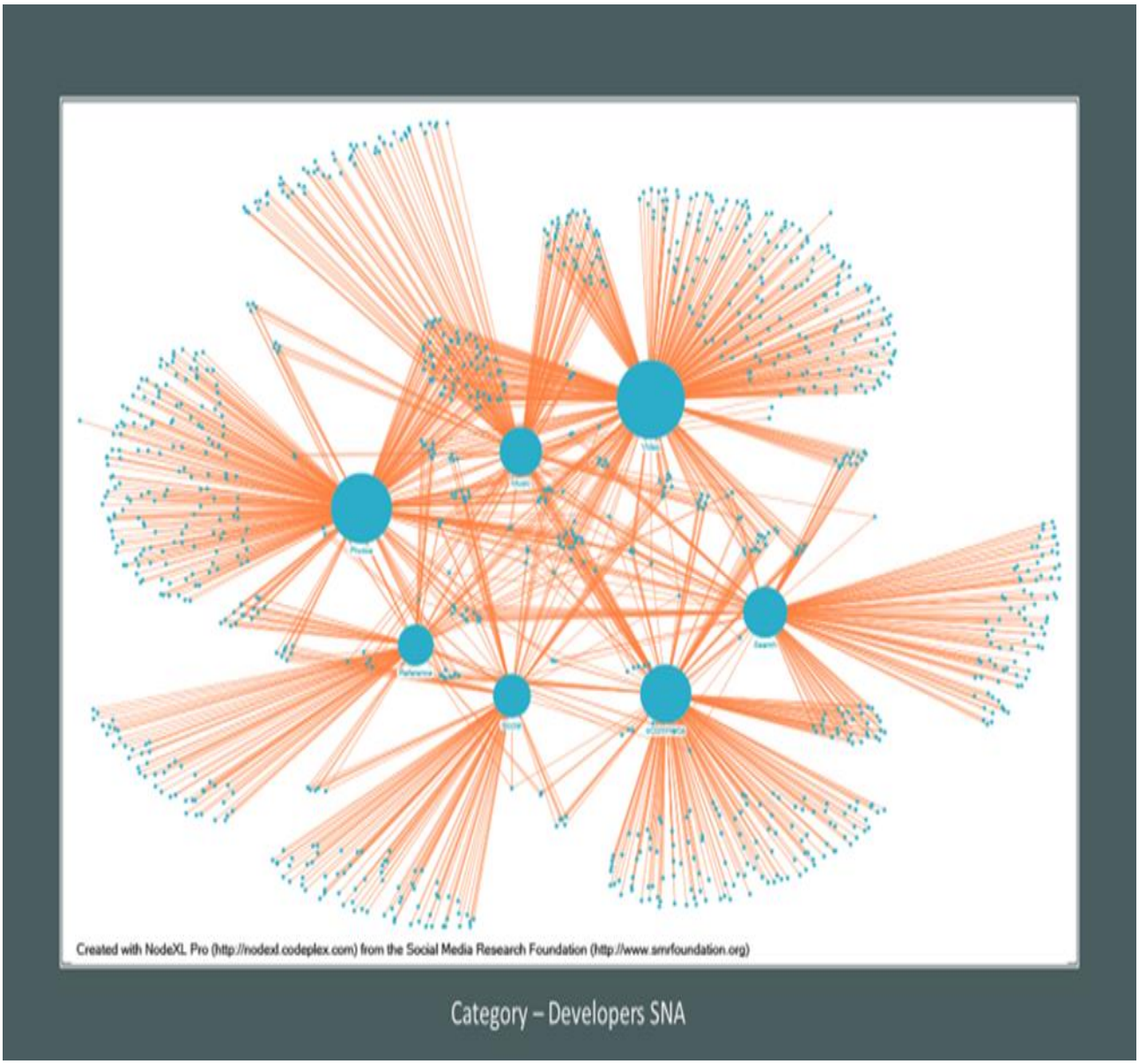


Figure 21. Category -Developers SNA

In a similar fashioned, the mashups SNA scenario revealed that the video category continues to lead, along with photos, eCommerce and a new trend with telephony, as depicted in figure 22.

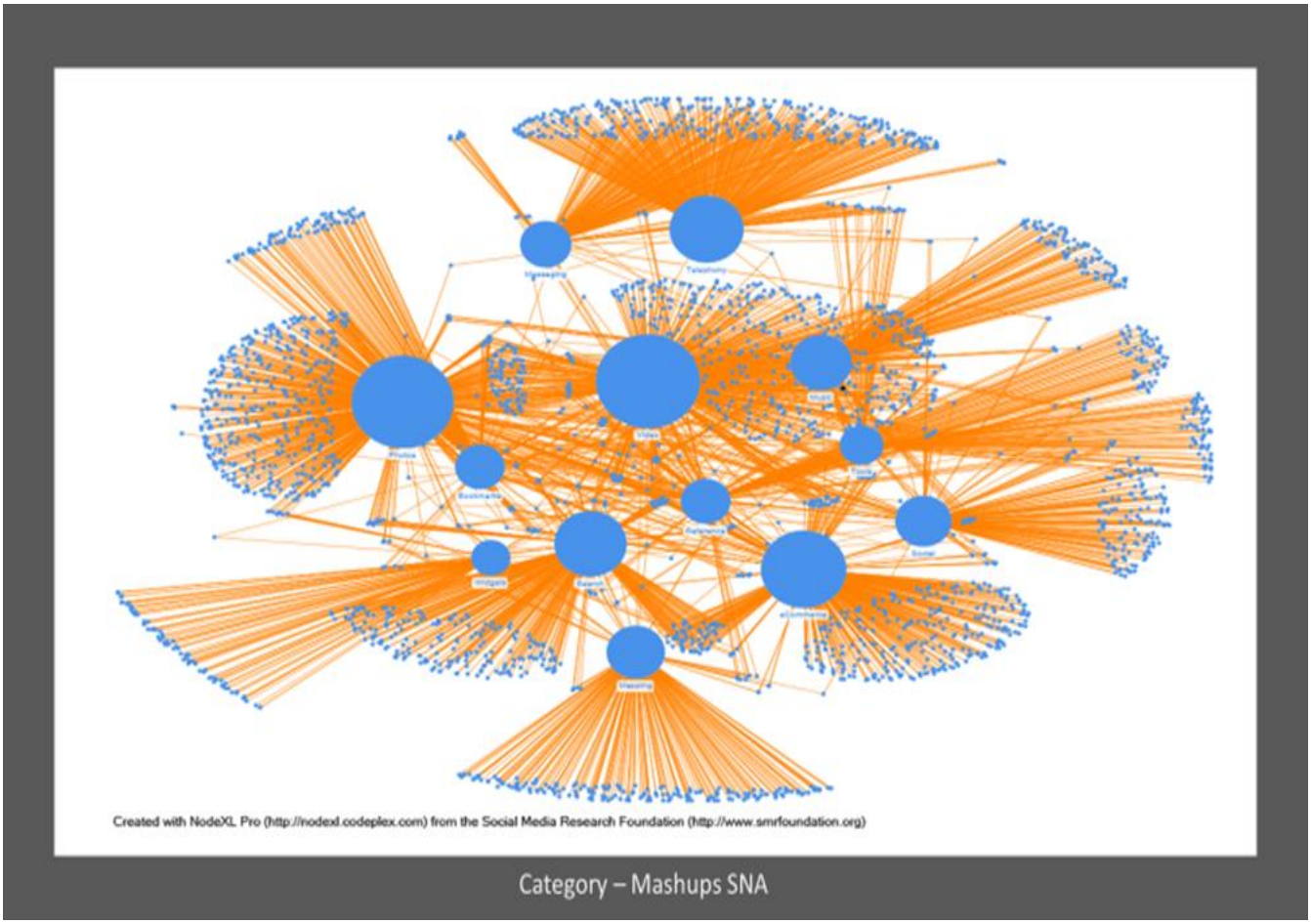


Figure 22. Category -Mashups SNA

4.6 Summary

The data collected regarding the APIs' ecosystem, indicates that a relationship exist between developers and mashups created under the established categories. The businesses creating and establishing API presence in the Web must maintain guard in how these categories are affected by new technology, the arrival of new developers and the obsolescence of mashups and API modules.

Furthermore, businesses must also intake how the APIs are being used, to maintain their presence and brand name, and the incorporation of monetizing methods which enables the API's ROI.

CHAPTER V

CONCLUSIONS AND RECOMMENDATIONS

5.0 Introduction

This chapter details the discussion of the research findings addressed in chapter IV, and how it can contribute to the academic knowledge based on chapter II. This chapter concentrates on the following topics: Discussion of Results, Conclusions, Implications, Limitations and Future Research Opportunities.

5.1 Discussion of Results

Developers are shifting from technical software creators within a business, into change agents. To succeed in the Internet economy, companies must take in consideration how the API model impacts their business vision, and the actions needed to incorporate an API strategy. An API can only be quantified through the developer's active creation process of mashups, website enhancements and the value the API's interface functionality can deliver.

Business opportunities are now entangled within the API model in the Internet economy, where entrepreneurs need to assess the API-developers equation when considering investing in new venues. As stated by Hammer (2016), "APIs used to be developer's tools, but now they are a business model driver". And as addressed by the Jones, B. (2012), to have success in the API world, developers must monitor the usage of the API to adapt to changes.

Academic literature is starting to focus on how APIs are becoming part of the business strategy, as a product and service. In accordance to Hammar (2016), many

businesses have benefit by treating the APIs as new products. While others, can drive revenue growth by reaching new users, channels and markets using their own public API (Sharma, 2017). In the words of Lundquist (2012), the APIs “have become a fast, simple, and effective way for organizations to connect their internal applications, connect with their partners, suppliers, and increasingly with customers and the public at large”.

The relationship between the APIs was measured, by depicting the following variables: APIs, SDKs, developers and mashups as main, libraries, articles and followers as reference. Which demonstrated the level of participation these entities had within the API ecosystem. The exposed API categories were then group by the parent company based on content analysis, to determine which business the API was addressing. API’s created towards developers’ activities were concerned with the production of eCommerce, mobile, messaging, reference and tools; while the top categories inclined towards users were video and photos.

5.1.1 Predominant API categories, and where they headed

The predominant categories hold 65% of the total value of data set collected It resolves into four companies: Google, flickr, Amazon, and Twilio. The user experience and mobility factors appear to have an important role when addressing the API needs. Older players are becoming more stable within the category influx such as mapping and searching. Business opportunities should be tied within the API model into categories that are showing an increase in demand or interest from the developer’s and the user’s point of view.

To understand were the API's categories are headed, can serve as an instrument for entrepreneurs searching business opportunities within companies who are controlling a considerable amount of the public API market share. By integrating a collaboration API strategy, entrepreneurs can gain access to the API market using leading company's API instruments, thru specialized microservices difficult to handle by large corporations due to their economy of scale. Entrepreneurs should also be cognizant on how each companies' API affects the overall category values within the ecosystem.

In congruence with captioned description, business opportunity seekers should be aware of the principles addressed by the Jones (2012), which it is imperative the usage of API statistics, recognize the purpose of the API being used, understand what the API's parameters are and when is the correct time to get involved.

5.1.2 Which are the APIs used the most by developers, and which companies predominate

The APIs used the most by developers resolved in 47% of the data extracted under three major companies resulting in 58% of the developer's realm. While an emerging API from the music category experiences a 7%. Business opportunities are adjacent to public APIs of major parent companies, were collaboration condition may exist for entrepreneurs to integrate their products and services.

In addition, single unit APIs should be evaluated when their function carries a high level of users, as the music related API. By understanding the developer's position between categories and APIs, subtle opportunities that flourish can be seized, when applying the proper API strategy.

Developers more than ever have an important role within the API Internet economy. Independent developers can now acquire access to public APIs and specialized in areas that offer a higher ROI when selecting business opportunities. Developers should consider as well, which APIs are addressing the needs of the users more efficiently and which companies are involved. In accordance to Peppard, J., and Ward, J. (2016) a simplified version of an Information systems and technology history can be outline by the Data Processing stage, where businesses were concerned with automatization, the Management Information Systems epoch, known for using the IT function for decision making and finally, the use of Strategic Information Systems which emphasizes the search for business opportunities to achieve a strategic advantage in this present era.

5.1.3 Similarities or differences between APIs and SDKs within categories, and their relationship

The categories under APIs and SDKs should be monitored more closely due to their proximity with developing tendencies and interest, reflected from companies and developers searching for better ways to do business. The API life cycle is a factor that cannot be underestimated, companies maintaining public APIs need to address this matter by providing new tools and SDKs environment updates. Most of the SDKs are driven towards the photos and social categories, which are inclined more for users than for developers. Without the proper tools, developers may be discouraged to initiate and spend a compromising amount of time, for a technology not offering an easy programming platform.

Despite the unlikeliness of a relationship between SDKs and APIs, their comparison can serve as a barometer when measuring the tendencies discovered

between them. Business opportunities connected with the APIs and companies satisfying this influx, through the creation of SDKs, can benefit, accelerate, provide security and ensure best practices in consonance with Sarrel (2016).

5.1.4 Similarities or differences between APIs and developers within category, and their relationship

Developers are the force drive of the API economy and responsible for the furtherance of new technologies within the Internet economy. This is confirmed when analyzing the values obtained from this study, in relation to developers and APIs within categories. Aside from eCommerce and reference which are the pillars of the developer's domain, the other remaining top categories fall under the user experience territory.

Developers are technical consumers, that need APIs and mashups to address their creation efforts and programming needs. The API ecosystem is still on an early stage, were user requirements are being addressed by developers, setting new ways to enhance the experience and value of created services within a global perspective. The developer's role is becoming more important due to the impact their mashups creations can generate and the use of API type when addressing the user's requirements and needs.

As stated by Wood (2016), the success or failure of an API depends on the quality of the developer; where an excellent developer with the following traits: relies on the ability to be a conduit, practical, visionary, collaborative and able to meet the set requirements from the organization's stakeholders.

5.1.5 Similarities or differences between APIs and mashups within category, and their relationship

The relationship between APIs and mashups within categories behave numerically in a similar fashion. This is due to the linkage between the API and mashups within the matrix extracted from the content analysis data. However, by analyzing the companies, holding the hegemony within the singular API, it can be determined which types are being used with more frequency than others. This action can lead developers to new business opportunities, after reducing the haze intertwined with the similarities shared between each category.

5.1.6 Behavior of APIs, developers, mashups within category, under the scope of social network analysis

The fact a company provides multiple APIs, does not guarantee the acceptance or success within a category. The spread of the APIs is based upon the continuous exchange process of functionality and compound needs expressed by users throughout the Internet. The outcome of the SNA resolves into the ability of locating nodes that can depict new business opportunities where collaboration and data sharing may exist. Clearly the Internet is not an Island, so even successful companies controlling a significant amount of the market share, need to create alliances with external entities such as independent developers, companies capable of responding to specialized requests at a smaller scale, and the public by generating brand presence using their API's instrumental functionality. That is why the API's DNA and strategy is so important, when developers are in the planning and creating stage of the API's functionality. By having a network and collaboration perspective, developers and

companies alike, can funnel their APIs product and services ideas into monetary generating devices.

5.2 Conclusions

APIs and mashups are essential for web development within the API Internet economy and can be traced down to the developer's role and connection with its purpose and functionality. Business entities can part take as API service providers as well as product generators to address the demands and needs of the public within the Internet economic model.

The categorization and relational analysis of APIs can provide a worth of information for companies, developers and the academia to search for possible business opportunities hidden at plain site by the obscurities the Internet entails, under a SNA perspective.

It is imperative for businesses and independent developers to understand the API equation and its significant in terms of business opportunities through collaboration, alliances and brand placement within API's strategic model.

Furthermore, based the results, the APIs categories demonstrating higher tendencies are video, photos, eCommerce and telephony which are inclined towards the user experience type, under Google, Flickr, Amazon, and Twilio. APIs instruments that are driving the developers are YouTube, Flickr, Amazon Product Advertising and Last.fm, inclined towards the user experience type, under the companies Google, Flickr, Amazon and Last.fm.

The most common SDKs and APIs categories are photos, social, eCommerce, music, video and reference; for developers and APIs categories are video, photos, ecommerce, search, music, social, reference and mapping. APIs and mashups

variables share the same categories and order. The fact a company provides multiple APIs, does not guarantee the acceptance or success within a category, like is with the Last.fm API case.

This study focused solemnly on public APIs available throughout the Internet. However, if the same systematic research approach is taken to measure private and partners APIs, results will center on how the categories established are being affected by the use of the APIs within the dimension of their network.

The variables studied are related within APIs, developers, mashups and followers. Libraries and article also take part of the correlation between them. However, the SDKs variable did not meet the statistical requirements to satisfy any significance for this study.

If the model of creating a virtual concentric world for measuring simulated clusters attached to the category tendencies, derived from this research is applied in Puerto Rico, subtle business opportunities can be discovered entangled within the Internet digital economy.

The new ecosystem paradigm shift is headed towards the exploitation of the API function as a monetary collection mechanism. The knowledge of API categories and behavior is instrumental for business entities and independent developers contemplating the creation of digital products or services, while addressing the complexities the multiple company environment entails within the Internet digital economy (Bharadwaj, A., et al., 2013).

5.3 Implications

5.3.1 Academic

The academia may benefit by providing technical courses and breeding new students towards the development of APIs and mashups destined to satisfy the demands of Internet customers worldwide. That said, universities and colleges no longer need to subscribe to a geographical area and can promote a global approach towards the creation of new business entities within their region.

5.3.2 Practical

Independent developers can enhance their possibilities by knowing which API category should be targeted, while having an insight of future instruments that may play a significant role in the Internet economy. Furthermore, by isolating the API's category function into a virtual concentric world, economic clusters (API categories) can be addressed within a global perspective, instead of the traditional geographic cluster analysis, to stimulate regional economies for products and services outside their domain and into the realm of the Internet digital economy.

5.4 Limitations

This research was delimited by three factors, the first one being the API as a new field of study, were academic publications are modest in comparison with corporate advertising pamphlet, which provide highly technical valuable information within the scope of their proprietary business model. In addition, technical driven websites (videos, newsletters, blogs, etc.) offer a myriad of information regarding the API model and economy, but do not conform to the scholastic level required for journal peer-reviewed publication. A collage of academic, technical and business disparate information was required to lessen said limitation.

Secondly, the data available regarding APIs' activity were limited to a few websites. However, this drawback was conquered to a certain degree with the assistance of www.programmableweb.com, the main website dedicated to the study of APIs.

The third factor entailed the percentage of public APIs available in the market compared with the private sector. Based on Business, D. R., & Innovation, U (2016), private APIs comprises 90% of the Web. This obstacle was eliminated by focusing on the APIs' data available through www.programmableweb.com. The dimensions provided by said website, were able to suffice the sample data requirements for this investigation.

5.5 Future Research Opportunities

This research resembles the tip of the iceberg in terms of the possibilities the API function can provided to businesses and the Internet economy. APIs facilitate digital services through the addition of new features, and effectively improve B2B interfacing with website and mobile applications. In addition, the API ecosystem is the new frontier for capturing new revenues, a reduction cost in customer acquisition and the benefit saving time when making the APIs' available to the market (Wordline Corporate, 2018).

This research focused on how the APIs behave within categories, and how developers, users and businesses can benefit by knowing where these categories are headed regarding the API ecosystem. Future academic research must center on how businesses can grasp new opportunities within the API ecosystem and turned them into profitable endeavors. Since the Internet has no boundaries, except the political ones, the creation of APIs to benefit users worldwide is a possibility previously impossible to

accomplish. Therefore, governments, the academia and businesses as full sketch entities, should incorporate the study and implication of the API ecosystem within their domain.

REFERENCES

451 Research (2015). The API Economy: How Bits of Code Empower Business *Relationships Retrieved from http://www.451alliance.com/Portals/5/2015reports /072215_api_economy/api_economy_report.pdf*

Allman, M. (2003). An evaluation of XML-RPC. ACM sigmetrics performance evaluation review, 30(4), 2-11.

Alpar P. and Kim M., (1990), A Microeconomic Approach to the Measurement of Information Technology Value. Journal of Management Information Systems. Vol. 7, No. 2 (Fall 1990), pp. 55-69.

Alvarez-Suescun, E. (2010). Combining transaction cost and resource-based insights to explain IT implementation outsourcing. Information Systems Frontiers, 12(5), 631-645.

Ankolekar, A., Krotzsch, M., Tran, T., and Vrandecic, D. (2007). The two cultures: Mashing up web 2.0 and the semantic web. Web Semantics: Science, Services and Agents on the World Wide Web, 6(1):70–75.

API. (2018). In Mulesoft.com. Retrieved from https://www.mulesoft.com/resources /api/what-is-an-api Retrieved 3/28/2018.

API Directory. (2018). In Programmableweb.com. Retrieved from https://www.programmableweb.com/category/all/apis

Ariel, O., Bartolini, C., Bergman, R., Mordechai, E., Peltz, C., & Tadeski, I. (2010). A Visual Tool for Rapid Integration of Enterprise Software Applications. Building, (3), 430-444. Retrieved from http://www.hpl.hp.com/techreports/2010/HPL-2010-29.html

Bakalov, F., Konig-Ries, B., Nauerz, A., & Welsch, M. (2009). Automating Mashups for Next-Generation Enterprise Portals. *IT professional*, *11*(4).

Barua, A., Whinston, A. B., & Yin, F. (2000). Value and productivity in the Internet economy. Computer, 33(5), 102-105.

Basole, R. C. (2016). Accelerating Digital Transformation: Visual Insights from the API Ecosystem. *IT Professional*, *18*(6), 20-25.

Beemer, B., & Gregg, D. (2009). Mashups: A Literature Review and Classification Framework. Future Internet, 1(1), 59-87. Molecular Diversity Preservation International. Retrieved from http://www.mdpi.com/1999-5903/1/1/59/

Benslimane, D., Dustdar, S., and Sheth, A. (2008). Services mashups: The new generation of web applications. IEEE Internet Computing, 12(5):13–15.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. (A. Gómez-Pérez, Y. Yu, & Y. Ding, Eds.) Scientific American, 284(5), 34

Berlind, D. (2017) APIs 101: What's the Difference Between All of These API Types? Part 6 Retrieved from https://www.youtube.com/watch?v=pjuLNpFHnkU

Bharadwaj, A., El Sawy, O. A., Pavlou, P. A., & Venkatraman, N. (2013). Digital Business Strategy: Toward A Next Generation of Insights. Mis Quarterly, 37(2), 471-482.

Bhattarai, S., Zhao, Z., & Crespi, N. (2010, November). Consumer mashups: end-user perspectives and acceptance model. In *Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services* (pp. 930-933). ACM.

Box, D., Ehnebuske, D., Kakivaya, G., Layman, A., Mendelsohn, N., Nielsen, H. F., ... & Winer, D. (2000). Simple object access protocol (SOAP) 1.1.

Boyd, M. (2017). How to Build Great API Developer Communities Retrieved from https://www.programmableweb.com/news/how-to-build-great-api-developer-communities/analysis/2017/09/27

Boyd. M. (2017). Understanding What it Takes to Secure Your API. API University Retrieved from https://www.programmableweb.com/news/understanding-what-it-takes-to-secure-your-api/analysis/2017/09/27

Business, D. R., & Innovation, U. Winning in the API Economy (2016).

Chin-Chiung, K. U. O., & Chi-Fang, L. I. U. (2017). The Development and Application of Transaction Cost Economics in Taiwan. Turkish Economic Review, 4(2), 205.

Cho, A. (2007). An introduction to mashups for health librarians. Provider, 22(1), 19-22. Retrieved from http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle: An+introduction+to+mashups+for+health+librarians#0

Citrix Consulting Services, (2002). Deploying Web-Based Applications Through Citrix MetaFrame Fort Lauderdale, FL. Retreived from http://www.citrix.com, 5/4/2011

Colorado State University Retreived (2018) from https://writing.colostate.edu/guides/

Cordella, A. (2006). Transaction costs and information systems: does IT add up? Journal of information technology, 21(3), 195-202.

Dearstyne, B. W. (2007). Blogs, Mashups, & Wikis Oh, My! Information Management Journal, 41(4), 24-33. Association of Records Managers & Administrators. Retrieved from http://search.ebscohost.com/loginaspx?direct=true&db=aph&AN=25853893

Debasis Roy Choudhuri, Biswajit Mohapatra. Fifty Shades of API for Business in New Networked Economy. Internet of Things and Cloud Computing. Vol. 3, No. 1, 2015, pp. 1-7. doi: 10.11648/j.iotcc.20150301.11

Dennis A. and Wixom H. (2002), Systems Analysis Design 2 edition, Wiley Hoboken, New Jersey, USA.

Depietro, R., Wiarda, E., & Fleischer, M. (1990). The context for change: Organization, technology and environment. The processes of technological innovation, 1990), 151-175.

Developer. (2018). In Merriam-webster.com. Retrieved from https://www.merriam-webster.com/dictionary/developer

Digital Mckinsey (2007, September). What it really takes to capture the value of APIs. Retrieved from https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/what-it-really-takes-to-capture-the-value-of-apis

Dorn, C., Schall, D., & Dustdar, S. (2009, December). Context-aware adaptive service mashups. In Services Computing Conference, 2009. APSCC 2009. IEEE Asia-Pacific (pp. 301-306). IEEE.

Eilon S. (1985). A Framework for Profitability and Productivity The Institute of Management Sciences, Interfaces 15: 3 May-June 1985 (pp. 31-40).

End-user. (2018). In Merriam-webster.com. Retrieved from https://www.merriam-ebster.com/dictionary/end%20users

Fichter, D., & Wisniewski, J. (2009). They grow up so fast: Mashups in the enterprise. Online, 33(3), 54-57.

Fielding, R. T., & Taylor, R. N. (2002). Principled design of the modern Web architecture. ACM Transactions on Internet Technology (TOIT), 2(2), 115-150.

Fischer, T., Bakalov, F., & Nauerz, A. (2009, March). An Overview of Current Approaches to Mashup Generation. In Wissensmanagement (pp. 254-259).

Gamble, M. T., & Gamble, R. (2008). Monoliths to Mashups: Increasing Opportunistic Assets. IEEE Software, 25(6), 71-79. IEEE Computer Society Press. Retrieved from http://ieeexplore.ieee.org/xpl /freeabsalljsp?arnumber=4670717

Grammel, L., & Storey, M. A. (2008). An end user perspective on mashup makers. University of Victoria Technical Report DCS-324-IR.

Green, G. C., & Hevner, A. R. (2000). The successful diffusion of innovations: guidance for software development organizations. IEEE software, 17(6), 96-103.

Guinard, D., Trifa, V. M., & Wilde, E. (2010). Architecting a mashable open world wide web of things. Technical report/Swiss Federal Institute of Technology Zurich, Department of Computer Science, 663.

Haines, M. N., & Rothenberger, M. A. (2010). How a service-oriented architecture may change the software development process. *Communications of the ACM*, *53*(8), 135-140.

Hammar, S. (2016). API economy puts a new spin on solving old problems. Cape Town: SyndiGate Media Inc. Retrieved from http://librarylogin.suagm.edu:84/docview/1830423601?accountid=28867

Hansen, D., Shneiderman, B., & Smith, M. A. (2010). Analyzing social media networks with NodeXL: Insights from a connected world. Morgan Kaufmann.

Hinchcliffe, D. (2009). Creating a unified model for enterprise mashups, |11/24/2009, www.zdnet.com, cbs interative. Retrieved from http://www.zdnet.com /blog/hinchcliffe/creating-a-unified-model-for-enterprise-mashups/872

IBM. (2015), Growing your business with APIs Retrieved from https://usa.visa.com/dam/VCOM/download/partner-with-us/growing-business-api-whitepaper.pdf

Jackson, M. H. (2009). The Mash-Up: A New Archetype for Communication. Journal of Computer-Mediated Communication, 14, 730-734.

Jhingran, A. (2018). How and Why to Transform Your Business into a Digital Ecosystem API University Retrieved from https://www.programmableweb.com/news/how-and-why-to-transform-your-business-digital-ecosystem/analysis/2018/01/11

Johnson, B., & Christensen, L. (2004). Educational research: Quantitative and qualitative approaches. Allyn & Bacon.

Jones, B. (2012). THE API REVOLUTION. Campaign, , 24-25. Retrieved from http://librarylogin.suagm.edu:84/docview/1021241448?accountid=28867

Jones, C. (2009). Content analysis: A Practical approach.

Kendall K. and Schmidt A. (2007). Mashups: The Art of Creating New Applications by Combining Two or More Web Sites Ecommerce: Decision Line, March 2007. Pages: 15-16.

Knobel, M., & Wilbur, D. (2009). Let's talk 2.0. Educational Leadership, 66 (6), 20–24.

Kriebei, C. H., & Raviv, A. (1980). An Economics Approach to Modeling the Productivity of Computer Systems. Management Science, 26(3), 297-311. Retrieved from http://mansci.journal.informs.org/cgi/doi/10.1287/mnsc.26.3.297

Kumar, R. L. (2004). A Framework for Assessing the Business Value of Information Technology Infrastructures. Journal of Management Information Systems, 21(2), 11-32. ME Sharpe. Retrieved from http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=14776227&site=ehost-live&scope=site.

Levin, S. G., Stephan, P. E., & Winkler, A. E. (2012). Innovation in academe: the diffusion of information technologies. Applied Economics, 44(14), 1765-1782.

Liu, X., Hui, Y., Sun, W., & Liang, H. (2007, July). Towards service composition based on mashup. In Services, 2007 IEEE Congress on (pp. 332-339). IEEE.

Lundquist, E. (2012). 3 steps to success in the API economy. Informationweek - Online, Retrieved from http://librarylogin.suagm.edu:84/docview/1112133242?accountid=28867

MacKenzie, C. M., Laskey, K., McCabe, F., Brown, P. F., Metz, R., & Hamilton, B. A. (2006). Reference model for service-oriented architecture 1.0. *OASIS standard*, *12*, 18.

Marjit, U., & Jana, S. (2009). Mashups: An Emerging Content Aggregation Web2. 0 Paradigm.

Marston, S., Li, Z., Bandyopadhyay, S., Zhang, J., & Ghalsasi, A. (2011). Cloud computing—The business perspective. *Decision support systems*, *51*(1), 176-189.

Maximilien, E. M., Ranabahu, A., & Gomadam, K. (2008). An online platform for web APIs and service mashups. IEEE Internet Computing, 12(5).

Mell, P., & Grance, T. (2011). The NIST definition of cloud computing.

Merrill, D. (2006). Mashups: The new breed of Web app. IBM Web Architecture Technical Library, 1-13. IBM Developer Works. Retrieved from http://www.ibm.com/developerworks/xml/library/x-mashups.html

Mikkonen, T., & Taivalsaari, A. (2007). Using JavaScript as a real programming language.

Mueller, B., Viering, G., Legner, C., & Riempp, G. (2010). Understanding the economic potential of service-oriented architecture. Journal of Management Information Systems, 26(4), 145-180.

Neale, P., Thapa, S., & Boyce, C. (2006). Preparing a case study: A guide for designing and conducting a case study for evaluation input. Massachusetts: Pathfinder international.

Neuendorf, K. A. (2016). *The content analysis guidebook*. Sage.

Oracle Communications (September 2014) Making Money Through API Exposure Enabling New Business Models Retrieved from http://www.oracle.com/us/industries/communications/comm-making-money-wp-1696335.pdf

O'Reilly, T. (2005). What Is Web 2.0. Pattern Recognition. San Francisco: Retrieved from http://oreilly.com/web2/archive/what-is-web-20.html

Palfrey, J., & Gasser, U. (2007). Case Study: Mashups Interoperability and Innovation. Social Science Research Network. Retrieved from http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1033232

Peppard, J., & Ward, J. (2016). The strategic management of information systems: Building a digital strategy. John Wiley & Sons.

Perficient.com (2017). Create an API Program to Drive Digital Transformation. Retreived from https://www.perficient.com/-/media/files/guide-pdf-links/digital-transformation-part-8-creating-an-api-program-to-drive-digital-transformation.pdf

Peslak, A. R. (2003). A Firm Level Study of Information Technology Productivity Using Financial and Market Based Measures. Journal of Computer Information Systems, 43(4), 72. Retrieved from http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=10346363&site=ehost-live&scope=site

RM Van Dam (2017, June 5). Digital Trust and the API economy Retrieved from https://dtw.tmforum.org/wp-content/uploads/2017/05/5.-Rob-van-den-Dam-1.pdf

Rogers, E. M. (2010). Diffusion of innovations. Simon and Schuster.

Sabbouh, M., Higginson, J., Semy, S., & Gagne, D. (2007, May). Web mashup scripting language. In Proceedings of the 16th international conference on World Wide Web (pp. 1305-1306). ACM.

Santos, W. (2017, July 7). API Providers with the Most SDKs and Sample Code Listings. Retrieved from https://www.programmableweb.com/news/api-providers-most-sdks-and-sample-code-listings/research/2017/07/13

Santos, W. (2017, September 01). Which Are Developers Favorite APIs? Retrieved from https://www.programmableweb.com/news/which-are-developers-favorite-apis/research/2017/09/01

Santos, W. (2017, November 26). API Types and Architectural Styles are Most Used? Retrieved from https://www.programmableweb.com/news/which-api-types-and-architectural-styles-are-most-used/research/2017/11/26.

Santos, W. (2018, February 21). ProgrammableWeb's Most Popular APIs of 2017 Retrieved from https://www.programmableweb.com/news/programmablewebs-most-popular-apis-2017/research/2018/02/21.

Santos, W. (2018, February 23). Research Shows Interest in Providing APIs Still High. Retrieved from https://www.programmableweb.com/news/research-shows-interest-providing-apis-still-high/research/2018/02/23.

Sarrel, M. (2016, February 24). Why API Providers Should Offer SDKs Retrieved from https://www.programmableweb.com/news/why-api-providers-should-offer-sdks/analysis/2016/02/24

Schumacher, S., & McMillan, J. H. (2006). Research in Education Evidence–Based Inquiry. Boston: Pearson Education.

Sekaran, U. (2003). Research methods for business . Hoboken.

Sircar, S., Choi J., (2007). A study of the impact of information technology on firm performance: a flexible production function approach. Info Systems Journal 2009 19, 313–339.

Snir, M., & Bader, D. A. (2004). A Framework for Measuring Supercomputer Productivity. International Journal of High Performance Computing Applications, 18(4), 417-432. Retrieved from http://hpc.sagepub.com/cgi/content /abstract/18/4/417

Soto-Acosta, P., Perez-Gonzalez, D., & Popa, S. (2014). Determinants of Web 2.0 technologies for knowledge sharing in SMEs. Service Business, 8(3), 425-438.

Taft, D. K. (2007). SOA and Web 2.0 Prepare to Merge-The trend of merging service-oriented architecture and Web 2.0 technologies will address issues normally handled through PC-based software, resulting in faster. *eWeek*, *4*, 7-13.

Texas State Auditor's Office, Methodology Manual (1995) Retreived from www.preciousheart.net/chaplaincy/Auditor_Manual/Metintro.pdf

Thong, J. Y. L. (1999). An integrated model of information systems adoption in small businesses. Journal of Management Information Systems, 15(4), 187–214.

Tornatzky, L. G., Fleischer, M., & Chakrabarti, A. K. (1990). Processes of technological innovation. Lexington books.

Tuchinda, R., Szekely, P., & Knoblock, C. A. (2008). Building Mashups by example. (J. M. Bradshaw, H. Lieberman, & S. Staab, Eds.) Proceedings of the 13th international conference on Intelligent user interfaces IUI 08, (December), 139. ACM Press. Retrieved from http://portal.acm.org/citation.cfm?doid=1378773.1378792

Visa and Apigee (2015). Growing your business with APIs. Retrieved from https://usa.visa.com/dam/VCOM/download/partner-with-us/growing-business-api-whitepaper.pdf

Wacker, J. G., Yang, C., & Sheu, C. (2016). A transaction cost economics model for estimating performance effectiveness of relational and contractual governance: theory and statistical results. International Journal of Operations & Production Management, 36(11), 1551-1575.

What Is Driving the API Economy Growth? (2017, October 10). States News Service. Retrieved from http://link.galegroup.com.librarylogin.suagm.edu:84/apps/doc/A509099123/AONE?u=uturabo&sid=AONE&xid=2d749c2b

Wodehouse, D. (2016). The API Economy: How to Plug in and Build your Business (Upwork company) Retrieved from https://pages.upwork.com/rs/518-RKL-392/images/16-0916-Api-for-Marketers%20FINAL.pdf?mkt_tok=eyJpIjoiTW1FMU9XVTFOVFk1T1RWbSIsInQiOiI2QzZTZlZHOTNpQVhHNk1sN3VuVHF0Q3V1cjBpVVBCQXg4TXUrZGhUVzBDR3BpR3lyTnZkMFRWY0Q3a1VYNzRuZW1rblp2U0hKUEIzalhUQ1NOOTdLbE9DcUI4Z2VyNmhraStJd0wraW9uVUFBajF1bmNKNHlUYVpBMVVKaDMyRCJ9

Wood, C. (2016, May 19), How to Hire a Great API Developer, API University Retrieved from https://www.programmableweb.com/news/how-to-hire-great-api-developer/analysis/2016/05/19

Wordline Corporate (2018, February), Shifting to an API economy. Worldline Corporate Retrieved from https://worldline.com/en/home/blog/2018/february/shifting-to-an-api-economy.html

Yue, K.-bun. (2009). Experience on Mashup Development with End User Programming Environment. Journal of Information Systems, 21(1), 111-120. Retrieved from http://prtl.uhcl.edu/portal/page/portal/SCE/COMPUTING_MATHMATICS_DIV/CM_Documents/YahooPipe_JISE_Paper_Published.pdf

Zang N. Mashups on the Web: End User Programming Opportunities and Challenges. Plateau '09 Workshop Orlando, ACM Press 2009.

APPENDIX A. Authorization email by Editor-in-Chief Mr. David Berlind

Subject: Re: Permission Request
From:david.berlind@programmableweb.com
To: Jose R. Reyes
Cc:david.berlind@programmableweb.com; ut_aojeda@suagm.edu; emarquez@suagm.edu
Date: Monday, April 23, 2018, 10:36:19 AM GMT-4

Hi Jose,

You are welcome to use our publicly published research. Thanks for checking. I'd love to hear more about your dissertation or how we can help.

David Berlind
Editor-in-Chief
ProgrammableWeb.com
david.berlind@programmableweb. com

On Sun, Apr 22, 2018 at 9:37 PM, Jose Reyes < jrcarta@yahoo.com > wrote:

Jose R. Reyes
Candidate Doctor
Business Administration Faculty
University of Turabo
Gurabo, Puerto Rico

Sunday, April 22, 2018

Dear editor-in-chief, Berlind:

As part of my doctoral compliance process, I need an email response granting permission for the use of the Research and Charts section, images and data available throughout the API directory from your website, https://www.programmableweb.com/, which will be incorporated in my dissertation findings.

Once again, thank you.

Sincerely,

Jose R. Reyes

APPENDIX B. IRB Approval Letter Waiver

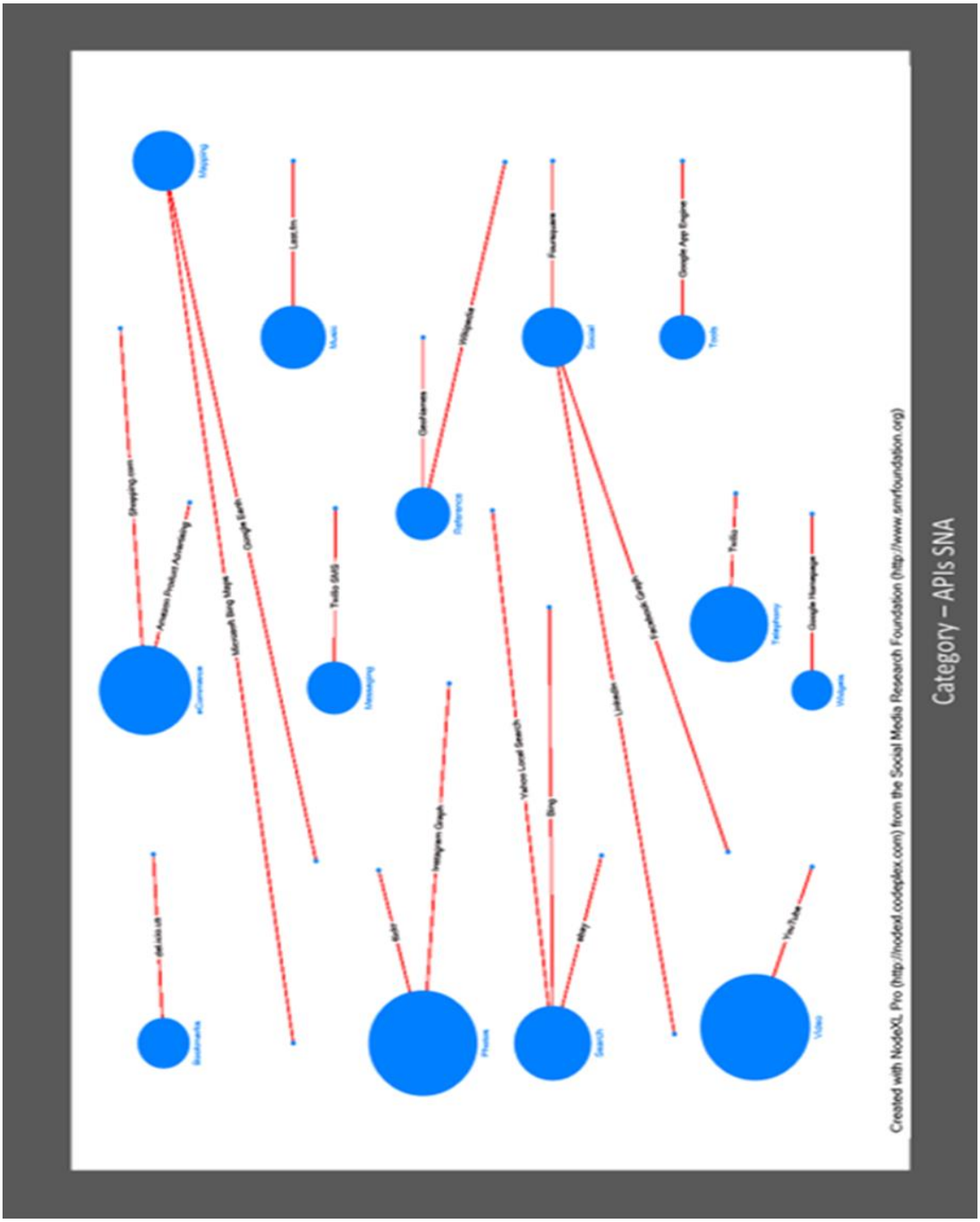
Data Obtained from Publicly Published Research Webpages

from https://www.programmableweb.com/

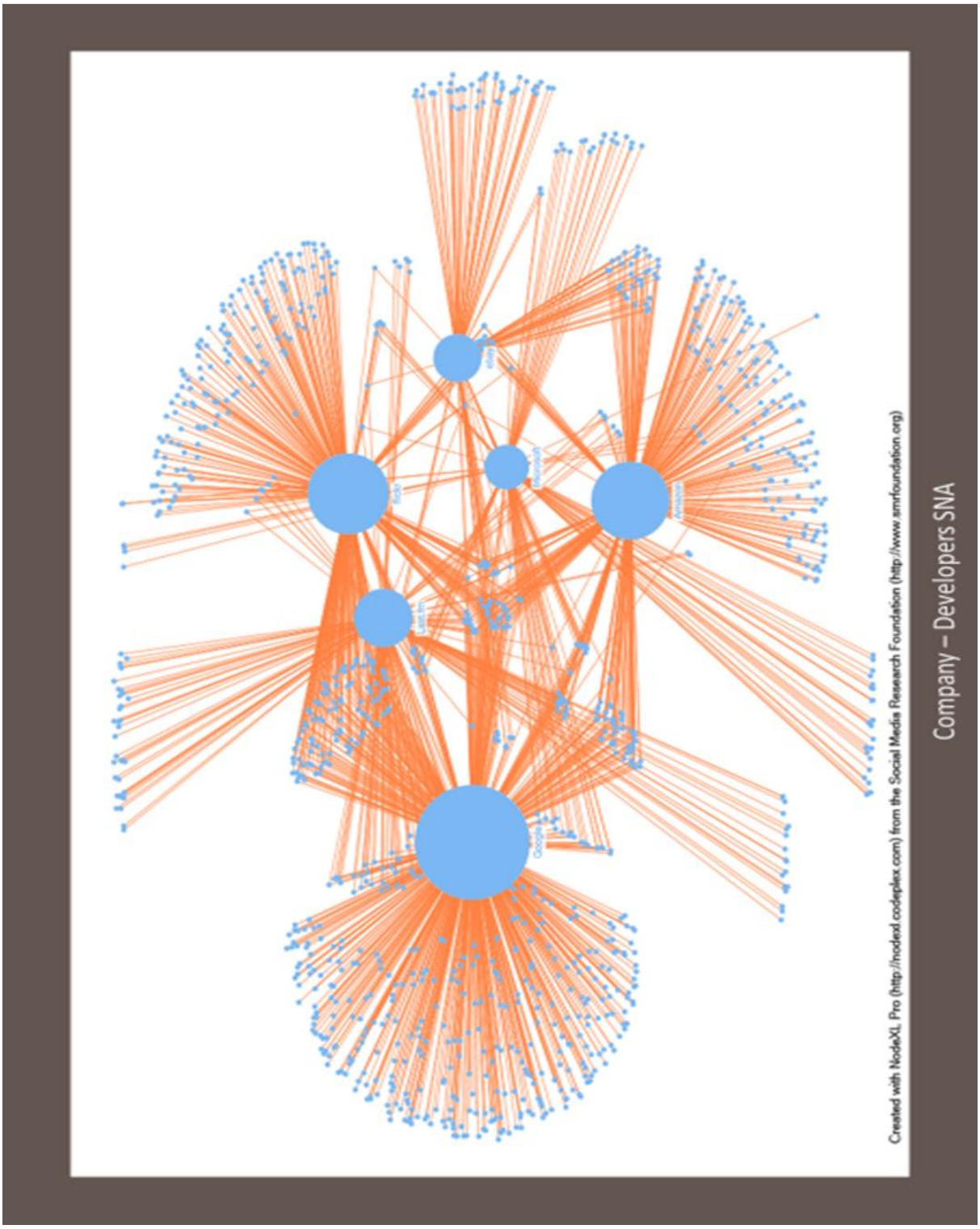
APPENDIX C. Extracted Data Summary

Company	Category	APIs	API	Developers	Mashups	SDKs	Followers	Libraries	Articles
Amazon	Cloud	Amazon EC2	77	51	77	1	111	6	37
Amazon	eCommerce	Amazon Product Advertising	433	230	432	11	971	9	12
Amazon	Storage	Amazon S3	91	61	91	2	236	9	31
box	Content	box	81	18	80	18	197	6	15
del.icio.us	Bookmarks	del.icio.us	160	72	160	0	215	0	8
DocuSign Enterprise	Electronic Signature	DocuSign Enterprise	94	7	94	10	70	11	12
ebay	Search	ebay	227	115	226	4	391	6	45
Facebook Graph	Social	Facebook Graph	51	42	51	52	317	2	59
flickr	Photos	flickr	633	324	630	60	689	19	51
Foursquare	Social	Foursquare	108	64	108	5	450	25	71
GeoNames	Reference	GeoNames	91	63	88	18	364	18	6
Google	Advertising	Google AdSense	84	66	84	9	139	6	5
Google	Analytics	Google Analytics Management	59	55	58	2	197	9	24
Google	Library	Google AJAX Libraries	62	51	62	0	134	0	4
Google	Mapping	Google Earth	52	40	51	3	205	0	10
Google	Tools	Google App Engine	124	70	124	8	217	5	52
Google	Video	YouTube	703	440	700	10	1471	17	68
Google	Widgets	Google Homepage	101	41	100	0	112	0	2
Instagram Graph	Photos	Instagram Graph	57	38	56	15	760	12	37
Last.fm	Music	Last.fm	249	170	246	10	508	19	19
LinkedIn	Social	LinkedIn	62	46	62	5	1211	18	32
Microsoft	Mapping	Microsoft Bing Maps	173	45	173	2	297	2	8
Microsoft	Search	Bing	67	55	67	0	246	3	13
Shopping.com	eCommerce	Shopping.com	63	31	62	0	220	0	3
Twilio	Messaging	Twilio SMS	174	33	172	4	202	0	22
Twilio	Telephony	Twilio	359	63	357	0	472	34	103
Wikipedia	Reference	Wikipedia	83	68	83	3	514	4	3
Yahoo Local Search	Search	Yahoo Local Search	55	25	55	0	197	0	2
Yelp Fusion	Recommendations	Yelp Fusion	60	50	59	22	290	1	15
Values Obtained from from https://www.programmableweb.com/									

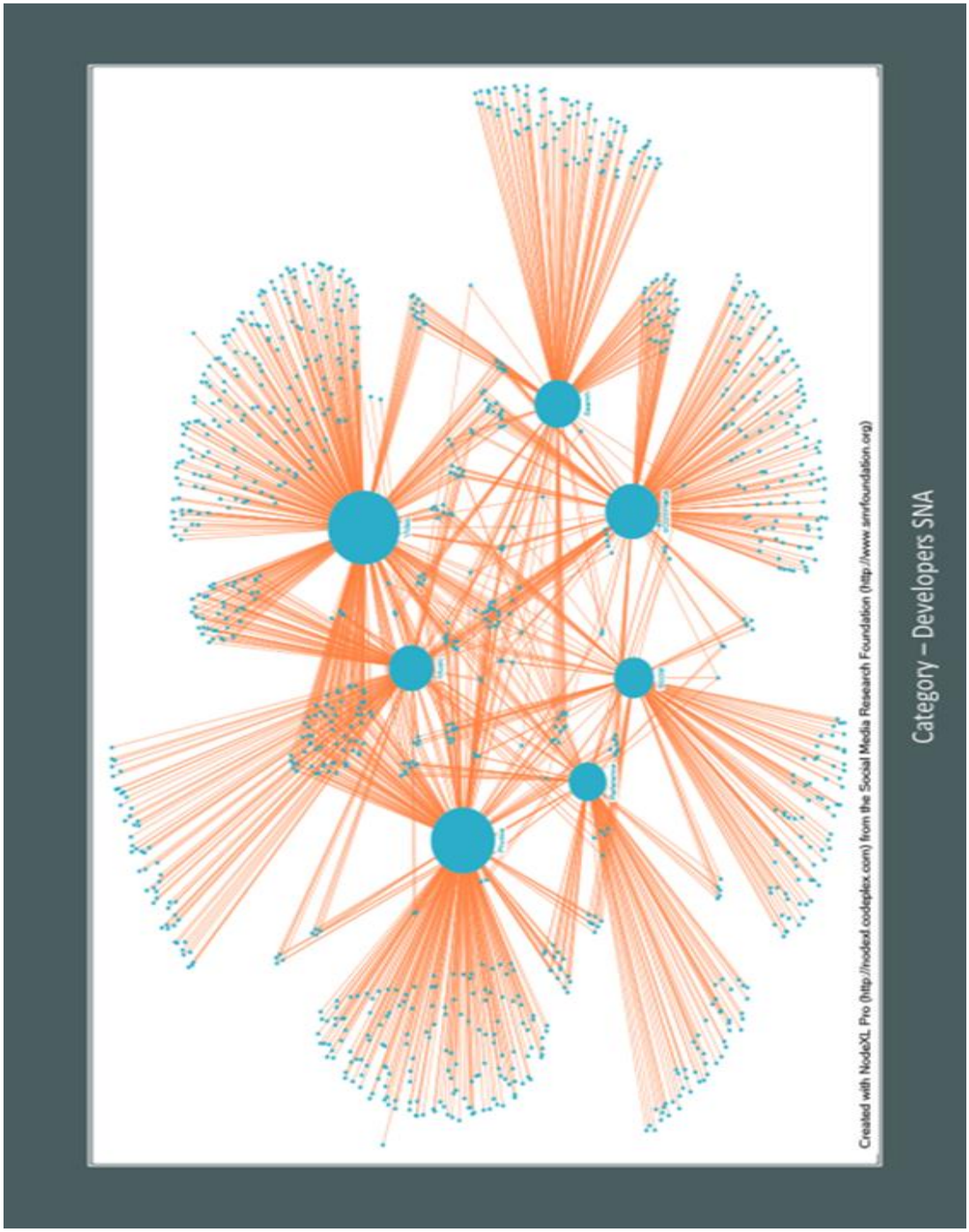
APPENDIX D. Expanded Graphic: Category - APIs SNA



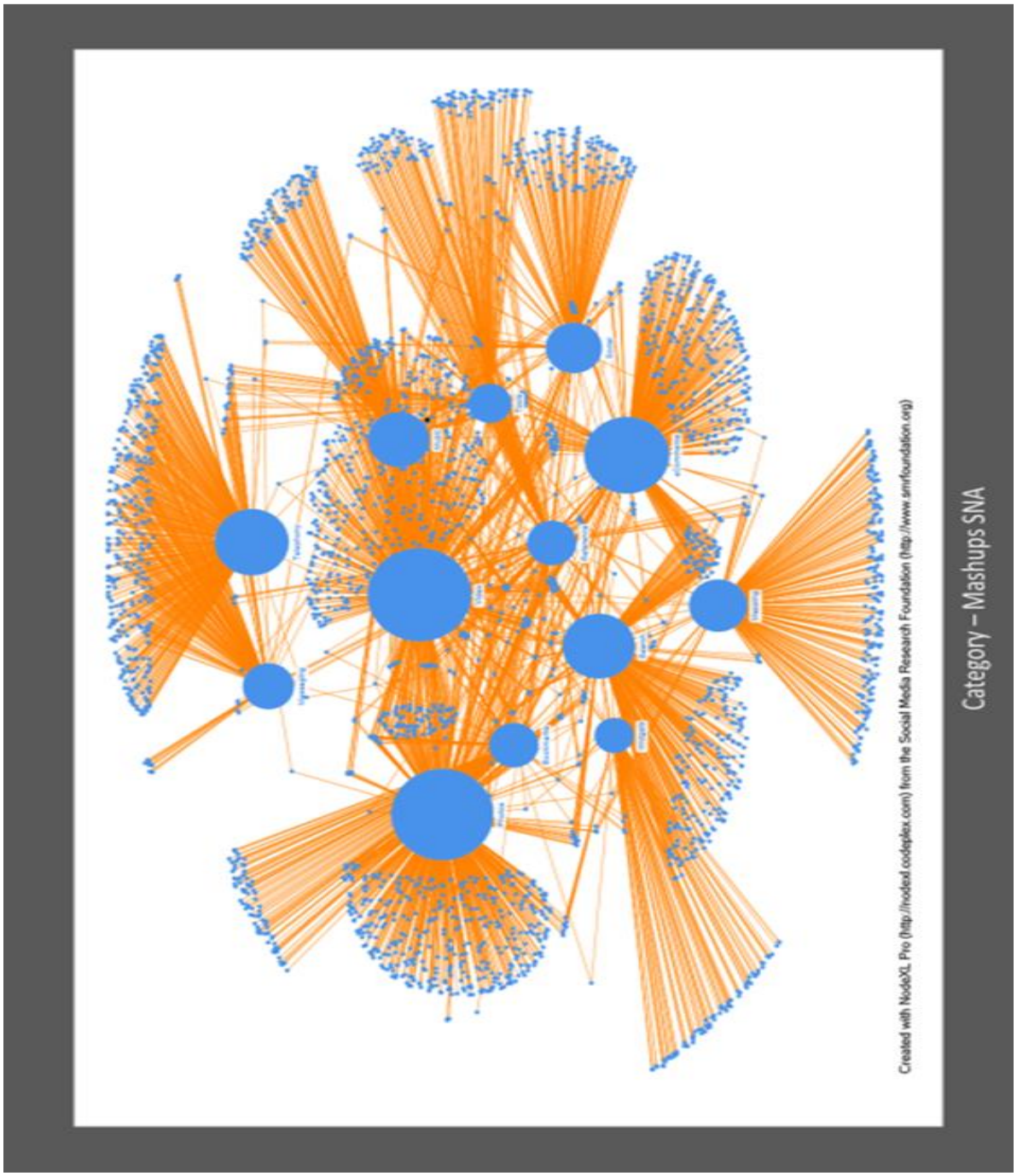
APPENDIX E. Expanded Graphics: Company – Developers SNA



APPENDIX F. Expanded Graphics: Category - Developers SNA



APPENDIX G. Expanded Graphics: Category - Mashups SNA



APPENDIX H. Inclusion and Exclusion Matrix

Include	Exclude
Literature Review	
Literature related to the description and functionality of APIs and mashups.	Technical literature related to the programming or creation process of APIs and mashups.
Common ground theory related to business economics, and the social sciences applied to Information Systems.	Blogs not addressing the APIs and mashups from a business or Information Systems perspective. Comments were omitted.
APIs Data	
Active APIs.	Non active APIs.
Top Values APIs within categories.	Names or usernames of Developers and followers.
Exception for Non active APIs based on historical value.	Names of mashups.
API's name and parent company.	Information related to SDKs, libraries, and articles.

APPENDIX I. APIs by Company Description Index

APIs by Company
Amazon EC2
Amazon Product Advertising
Amazon S3
box
del.icio.us
DocuSign Enterprise
eBay
Facebook Graph
flickr
Foursquare
GeoNames
Google AdSense
Google AJAX Libraries
Google Analytics Management
Google App Engine
Google Earth
Google Homepage
Google YouTube
Instagram Graph
Last.fm
LinkedIn
Microsoft Bing
Microsoft Bing Maps
Shopping.com
Twilio
Twilio SMS
Wikipedia
Yahoo Local Search
Yelp Fusion

Home » APIs » Amazon EC2

TRACK API

Amazon EC2 API

Cloud | Datacenter | Domains | Hosting

The Amazon Elastic Compute Cloud (Amazon EC2) API is a web service that enables you to launch and manage Linux/UNIX and Windows server instances in Amazon's data centers. Giving users the ability to "compute" in the cloud, it provides users with complete control of their computing resources and lets them run on Amazon's computing environment. Amazon EC2 changes the economics of computing by allowing users to pay only for capacity that is actually used. Amazon EC2 provides two APIs: SOAP and REST. The same XML body is returned in both the REST API and SOAP API. The WSDL for each supported API version is available from the following URI: http://ec2.amazonaws.com/doc//AmazonEC2.wsdl The WSDL should be treated as a moving target as it will always map to the latest release of the Amazon EC2 SOAP API.

Summary | SDKs (1) | Articles (37) | How To (0) | Sample Source Code (5) | Libraries (6) | Developers (77) | Followers (111) | Comments (2)

SPECS

API Endpoint	http://ec2.amazonaws.com/doc/2006-06-26/AmazonEC2.wsdl
API Portal / Home Page	http://docs.aws.amazon.com/AWSEC2/latest/APIReference/Welcome.html
Primary Category	Cloud
Secondary Categories	Datacenter, Domains, Hosting
API Provider	Amazon
SSL Support	Yes
API Forum / Message Boards	http://developer.amazonwebservices.com/connect/forum.jspa?forumID=30
Twitter URL	https://twitter.com/AWSSupport
Developer Support URL	http://developer.amazonwebservices.com/connect/kbcategory.jspa?categoryID=59
Interactive Console URL	http://aws.amazon.com/console/
Authentication Model	API Key
Is the API Design/Description Non-Proprietary ?	No
Type	Web/Internet
Scope	Metaservice API
Device Specific	No
Docs Home Page URL	http://docs.aws.amazon.com/AWSEC2/latest/APIReference/Welcome.html
Architectural Style	REST
Supported Request Formats	JSON, SOAP, XML
Supported Response Formats	SOAP, XML
Is This an Unofficial API?	No
Is This a Hypermedia API?	No
Restricted Access (Requires Provider Approval)	No

Home » APIs » Amazon Product Advertising

Amazon Product Advertising API

eCommerce | Advertising

What was formerly the ECS - eCommerce Service - has been renamed the Product Advertising API. Through this API developers can retrieve product information. The API exposes Amazon's product data and e-commerce functionality. This allows developers, web site publishers and others to leverage the Amazon Product Discovery features that Amazon uses to power its own business, and potentially make money as an Amazon affiliate. Additionally, the API has features allowing developers to advertise proucts, let users search for Amazon products and help users discover Amazon products. Both REST and SOAP APIs are provided, this profile is for the REST API.

TRACK API

Summary | SDKs (11) | Articles (12) | How To (0) | Sample Source Code (9) | Libraries (9) | Developers (435) | Followers (978) | Comments (0)

SPECS

API Endpoint	http://webservices.amazon.com/
API Portal / Home Page	https://affiliate-program.amazon.com/gp/advertising/api/detail/main.html
Primary Category	eCommerce
Secondary Categories	Advertising
API Provider	Amazon
SSL Support	No
API Forum / Message Boards	http://developer.amazonwebservices.com/connect/forumindex.jspa
Authentication Model	API Key
Terms Of Service URL	http://www.amazon.com/gp/browse.html/ref=sc_fe_c_2_3434651_3/103-8199824-2730214?_encoding=UTF8&node=3440661&no=3434651&me=A36L942TSJ2AJA
Is the API Design/Description Non-Proprietary ?	Yes
Type	Web/Internet
Scope	Single purpose API
Device Specific	No
Docs Home Page URL	https://docs.aws.amazon.com/AWSECommerceService/latest/DG/Welcome.html
Description File URL (if public)	http://webservices.amazon.com/AWSECommerceService/AWSECommerceService.wsdl
Description File Type	WSDL
Architectural Style	REST
Supported Request Formats	SOAP, XML
Supported Response Formats	SOAP, XML
Is This an Unofficial API?	No
Is This a Hypermedia API?	Yes
Restricted Access (Requires Provider Approval)	Yes

Home » APIs » Amazon S3

TRACK API

Amazon S3 API

Storage Cloud

Since 2006 Amazon Web Services has been offering web services commonly known as cloud computing. AWS Provides a reliable, low cost infrastructure platform that powers hundreds of thousands of businesses. Amazon S3 API, the Simple Storage Service provides a simple web services interface used to store objects using the Amazon online storage infrastructure. Users can then download the data or use the data with other AWS services, such as Amazon Elastic Cloud Computer (EC2). Using this web service, developers can build applications that make use of Internet storage. The API uses Buckets, Objects, Keys and Operations. An object has four parts: value, key, metadata, and an access control policy. Objects are stored in buckets. The API is REST based. Responses are formatted in JSON.

Summary | SDKs (2) | Articles (31) | How To (0) | Sample Source Code (2) | Libraries (9) | Developers (91) | Followers (236) | Comments (2)

SPECS

API Endpoint	http://s3.amazonaws.com/
API Portal / Home Page	http://aws.amazon.com/s3/
Primary Category	Storage
Secondary Categories	Cloud
API Provider	Amazon
SSL Support	Yes
API Forum / Message Boards	http://developer.amazonwebservices.com/connect/forum.jspa?forumID=24
Twitter URL	https://twitter.com/AmazonAppDev
Developer Support URL	http://developer.amazonwebservices.com/connect/forum.jspa?forumID=24
Authentication Model	API Key, Shared Secret
Terms Of Service URL	https://aws.amazon.com/terms/
Is the API Design/Description Non-Proprietary ?	No
Type	Web/Internet
Scope	Single purpose API
Device Specific	No
Docs Home Page URL	https://docs.aws.amazon.com/AmazonS3/latest/API/Welcome.html
Architectural Style	REST
Supported Request Formats	URI Query String/CRUD
Supported Response Formats	JSON, XML
Is This an Unofficial API?	No
Is This a Hypermedia API?	No
Restricted Access (Requires Provider Approval)	Yes

Home » APIs » Box

Box API

Content | Collaboration | Content Management | Enterprise | Security | Storage

Box is a modern content management platform that transforms how organizations work and collaborate to achieve results faster.

Box Platform provides content APIs to build secure content experiences in custom apps. With Box Platform, organizations can bring content management and collaboration services to their own custom apps or build custom integrations with Box.

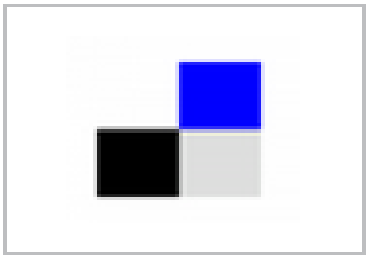
TRACK API

Summary | SDKs (10) | Articles (15) | How To (8) | Sample Source Code (8) | Libraries (6) | Developers (81) | Followers (199) | Comments (8)

SPECS

API Endpoint	https://api.box.com/2.0/
API Portal / Home Page	https://developer.box.com/reference
Primary Category	Content
Secondary Categories	Collaboration, Content Management, Enterprise, Security, Storage
API Provider	Box
SSL Support	Yes
API Forum / Message Boards	https://community.box.com/t5/Developer-Forum/bd-p/DeveloperForum
Twitter URL	https://twitter.com/boxplatform
Support Email Address	platform@box.com
Developer Support URL	platform@box.com
Authentication Model	API Key, OAuth 2, Other/Custom, Token
Version	2.0
Terms Of Service URL	https://www.box.net/static/html/terms.html
Is the API Design/Description Non-Proprietary ?	No
Type	Web/Internet
Scope	Single purpose API
Device Specific	No
Docs Home Page URL	https://developer.box.com/docs
Architectural Style	REST
Supported Request Formats	URI Query String/CRUD
Supported Response Formats	JSON
Is This an Unofficial API?	No
Is This a Hypermedia API?	Yes
Restricted Access (Requires Provider Approval)	No

Home » APIs » del.icio.us



TRACK API

del.icio.us API

Bookmarks | Social

From their site: del.icio.us is a social bookmarking website – the primary use of del.icio.us is to store your bookmarks online, which allows you to access the same bookmarks from any computer and add bookmarks from anywhere, too. On del.icio.us, you can use tags to organize and remember your bookmarks, which is a much more flexible system than folders. The API provides read/write access to Delicious bookmarks and tags via an HTTP-based interface. The RESTful API returns responses in XML.

Summary | SDKs (0) | Articles (0) | How To (0) | Sample Source Code (0) | Libraries (0) | Developers (165) | Followers (215) | Comments (6)

SPECS

API Endpoint	https://api.del.icio.us/v1/
API Portal / Home Page	https://delicious.com/developers#title1
Primary Category	Bookmarks
Secondary Categories	Social
SSL Support	Yes
API Forum / Message Boards	http://support.delicious.com/forum/index.php?CategoryID=0
Support Email Address	https://secure.delicious.com/help/support
Developer Support URL	None
Is the API Design/Description Non-Proprietary ?	No
Scope	Single purpose API
Device Specific	No
Architectural Style	REST
Supported Request Formats	None Specified
Supported Response Formats	XML
Is This an Unofficial API?	No
Is This a Hypermedia API?	No
Restricted Access (Requires Provider Approval)	No

DocuSign Enterprise API

Electronic Signature | Contracts | Documents

DocuSign is a Cloud based legally compliant eSignature service. DocuSign service provides an intuitive web, mobile and an API interfaces. DocuSign API allows an application to connect DocuSign service or embed parts of DocuSign user experience. By connecting to DocuSign web services applications can create electronic signature transactions, retrieve electronically signed documents, get up to date status on outstanding envelopes and do other Create/Read/Update /Delete operations. By embedding DocuSign interface applications can provide single-sign-on capability and insert signing or sending user experience into their applications.

TRACK API

Summary | SDKs (10) | Articles (52) | How To (8) | Sample Source Code (0) | Libraries (15) | Developers (97) | Followers (79) | Comments (1)

SPECS

API Endpoint	https://demo.docusign.net/api/3.0/api.asmx
API Portal / Home Page	https://www.docusign.com/developer-center
Primary Category	Electronic Signature
Secondary Categories	Contracts, Documents
How is this API different ?	+ Carrier-Grade Availability + Bank-Class Security + ISO 27001 Certified
API Provider	DocuSign
SSL Support	Yes
API Forum / Message Boards	http://stackoverflow.com/questions/tagged/docusignapi
Twitter URL	http://twitter.com/DocuSignAPI
Support Email Address	mikeb@docusign.com
Developer Support URL	http://stackoverflow.com/questions/tagged/docusignapi
Interactive Console URL	https://demo.docusign.net/
Authentication Model	API Key, HTTP Basic Auth, OAuth 2, SAML, Token, WS-Security
Version	3.0
Terms Of Service URL	http://www.docusign.com/terms_of_use/
Is the API Design/Description Non-Proprietary ?	Yes
Scope	Single purpose API
Device Specific	No
Docs Home Page URL	https://www.docusign.com/developer-center
Description File URL (if public)	https://raw.githubusercontent.com/docusign/eSign-OpenAPI-Specification/master/esignature.rest.swagger.json
Description File Type	Swagger 2.0 JSON (OAI)
Architectural Style	REST
Supported Request Formats	SOAP, URI Query String/CRUD, XML
Supported Response Formats	JSON, SOAP, XML
Is This an Unofficial API?	No
Is This a Hypermedia API?	No
Restricted Access (Requires Provider Approval)	No

Home » APIs » eBay

TRACK API

eBay API

deprecated

Search | Auctions | eCommerce

[This API is no longer available. It has been replaced by the following APIs: eBay Sell Account, eBay Sell Inventory, eBay Sell Fulfillment, eBay Sell Marketing, eBay Sell Analytics, eBay Sell Metadata, eBay Buy Marketing, eBay Buy Browse, eBay Buy Feed, eBay Buy Order, eBay Taxonomy, eBay Finding, eBay Shopping, eBay Merchandising, eBay Platform Notifications, eBay Product Services, eBay Client Alerts, eBay Post Order, and eBay Feedback.

This profile is maintained for historical and research purposes only.]

eBay's primary web services are found in the three following APIs:

The [[/apis/ebay-trading Trading API]] is designed to provide robust support for transactional activities. The services offered by the Trading API allow developers to list items, manage user information, get item information, and manage eBay sales and purchases.

The [[/apis/ebay-shopping Shopping API]] is a more lightweight service that is optimized for buyer shopping and browsing. The Shopping API lets users take searches off of eBay and put them into their own site or application. This API is designed for the creation of buying applications with very fast response times.

The [[/apis/ebay-finding Finding API]] is eBay's next generation search API. The Finding API provides more relevant search results compared to existing search API's and has an array of search refinement capabilities. Developers can leverage the Finding API to integrate more robust searching and browsing experiences into their buying applications.

Note: As of October 2011, eBay has deprecated the GetSearchResults and GetCategoryListings services from the Trading API, and the FindItemsAdvanced and FindItems services from the Shopping API. This functionality has been migrated to the Finding API.

Summary | SDKs (4) | Articles (45) | How To (8) | Sample Source Code (2) | Libraries (6) | Developers (227) | Followers (382) | Comments (2)

SPECS

Primary Category	Search
Secondary Categories	Auctions, eCommerce
API Provider	eBay
SSL Support	Yes
Supported Request Formats	None Specified
Supported Response Formats	None Specified

Home » APIs » Facebook Graph

Facebook Graph API

Social | Demographics

From documentation: At Facebook's core is the social graph; people and the connections they have to everything they care about. The Graph API presents a simple, consistent view of the facebook social graph, uniformly representing objects in the graph (e.g., people, photos, events, and pages) and the connections between them (e.g., friend relationships, shared content, and photo tags). Public information can be accessed without a developer key/application key, but is required for private data access.

The Facebook Graph API has undergone numerous versioning updates. Launched in November 2017 and scheduled to last until at least November 2019, the newly integrated features of version 2.11 include: updates to the pages API allowing users to know what is happening to private replies, posts an commenting. Updates to groups/videos permissions requiring a user access token to read videos from a group, sunsetting of app links host. All these changes are accessible in the documentation of the API.

TRACK API

Summary	SDKs (52)	Articles (599)	How To (3)	Sample Source Code (0)	Libraries (2)	Developers (51)	Followers (3159)	Comments (1)

SPECS

API Endpoint	https://graph.facebook.com
API Portal / Home Page	https://developers.facebook.com/docs/graph-api
Primary Category	Social
Secondary Categories	Demographics
Is the API related to any other API ?	Facebook public feed API Facebook Keyword Insights API
API Provider	Facebook
SSL Support	Yes
API Forum / Message Boards	https://www.facebook.com/groups/fbdevelopers
Twitter URL	https://twitter.com/facebook
Developer Support URL	https://developers.facebook.com/tools-and-support/
Authentication Model	Token
Version	2.11
Terms Of Service URL	https://developers.facebook.com/policy/
Is the API Design/Description Non-Proprietary ?	No
Type	Web/Internet
Scope	Single purpose API
Device Specific	No
Docs Home Page URL	https://developers.facebook.com/docs/graph-api
Architectural Style	REST
Supported Request Formats	JSON, JSONP, URI Query String/CRUD
Supported Response Formats	JSON, JSONP
Is This an Unofficial API?	No
Is This a Hypermedia API?	No
Restricted Access (Requires Provider Approval)	No

Home » APIs » Flickr

TRACK API

Flickr API

Photos | Video

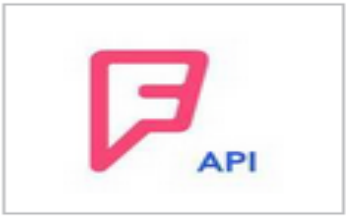
The Flickr API can be used to retrieve photos from the Flickr photo sharing service using a variety of feeds - public photos and videos, favorites, friends, group pools, discussions, and more. The API can also be used to upload photos and video.The Flickr API supports many protocols including REST, SOAP, XML-RPC. Responses can be formatted in XML, XML-RPC, JSON and PHP. Documentation is included for 14 API Kit libraries.

Summary | SDKs (0) | Articles (51) | How To (3) | Sample Source Code (6) | Libraries (19) | Developers (634) | Followers (690) | Comments (9)

SPECS

API Endpoint	https://api.flickr.com/services/
API Portal / Home Page	https://www.flickr.com/services/api/
Primary Category	Photos
Secondary Categories	Video
API Provider	Flickr
SSL Support	Yes
API Forum / Message Boards	https://www.flickr.com/groups/api
Twitter URL	https://twitter.com/flickr
Developer Support URL	https://www.flickr.com/help/forum/en-us/
Authentication Model	OAuth 1
Terms Of Service URL	https://flickr.com/services/api/
Is the API Design/Description Non-Proprietary ?	No
Type	Web/Internet
Scope	Single purpose API
Device Specific	No
Docs Home Page URL	https://www.flickr.com/services/api/
Architectural Style	REST
Supported Request Formats	URI Query String/CRUD, XML, PHP, XML-RPC
Supported Response Formats	JSON, XML, PHP, XML-RPC
Is This an Unofficial API?	No
Is This a Hypermedia API?	Yes
Restricted Access (Requires Provider Approval)	No

Home » APIs » Foursquare



Foursquare API

Social | Mapping | Mobile | Photos | Search

The Foursquare API provides location based experiences with diverse information about venues, users, photos, and check-ins. The API supports real time access to places, Snap-to-Place that assigns users to specific locations, and Geo-tag. Additionally, Foursquare allows developers to build audience segments for analysis and measurement. JSON is the preferred response format.

TRACK API

Summary | SDKs (5) | Articles (72) | How To (0) | Sample Source Code (10) | Libraries (25) | Developers (109) | Followers (453) | Comments (0)

SPECS

API Endpoint	https://api.foursquare.com/v2/
API Portal / Home Page	https://developer.foursquare.com/
Primary Category	Social
Secondary Categories	Mapping, Mobile, Photos, Search
API Provider	Foursquare
SSL Support	Yes
API Forum / Message Boards	https://stackoverflow.com/questions/tagged/foursquare
Twitter URL	http://twitter.com/foursquareapi
Support Email Address	api@foursquare.com
Authentication Model	App ID, OAuth 2, Shared Secret, Token
Version	2
Terms Of Service URL	https://developer.foursquare.com/docs/terms-of-use/overview
Is the API Design/Description Non-Proprietary ?	No
Type	Web/Internet
Scope	Single purpose API
Device Specific	No
Docs Home Page URL	https://developer.foursquare.com/docs
Architectural Style	REST
Supported Request Formats	URI Query String/CRUD
Supported Response Formats	JSON
Is This an Unofficial API?	No
Is This a Hypermedia API?	Yes
Restricted Access (Requires Provider Approval)	Yes

Home » APIs » GeoNames

Geo-Names

TRACK API

GeoNames API

Reference | Database | Mapping | Postal | Weather

Geonames is a geographical database with web services that let users extract useful information about different places, such as weather, timezone and post codes. The GeoNames geographical database covers all countries and contains over eight million placenames that are available for download free of charge. Users can edit, correct and add new names using a wiki interface. Geonames provides a suite of 36 Webservices covering the following categories: Fulltext search Place Hierarchy Wikipedia Postal Codes Reverse Geocoding Earthquakes Weather The APIs use RESTful protocol and responses vary in format between XML, JSON, RDF, CSV, TXT, RSS and KML.

Summary | SDKs (18) | Articles (6) | How To (0) | Sample Source Code (3) | Libraries (18) | Developers (91) | Followers (369) | Comments (2)

SPECS

API Endpoint	http://api.geonames.org/
API Portal / Home Page	http://www.geonames.org/export/
Primary Category	Reference
Secondary Categories	Database, Mapping, Postal, Weather
API Provider	GeoNames
SSL Support	Yes
API Forum / Message Boards	https://groups.google.com/forum/#!forum/geonames
Developer Support URL	http://forum.geonames.org/
Authentication Model	Unspecified
Terms Of Service URL	http://www.geonames.org/export/
Is the API Design/Description Non-Proprietary ?	Yes
Type of License if Non Proprietary	Creative Commons Attributions License
Type	Web/Internet
Scope	Single purpose API
Device Specific	No
Docs Home Page URL	http://www.geonames.org/export/web-services.html
Architectural Style	REST
Supported Request Formats	URI Query String/CRUD
Supported Response Formats	CSV, JSON, KML, RDF, RSS, Text, XML
Is This an Unofficial API?	No
Is This a Hypermedia API?	No
Restricted Access (Requires Provider Approval)	No

TRACK API

Google AdSense API

Advertising SEO

The Google AdSense API is ideal for developers whose users create their own content through blogging, web publishing, forum/wiki/community creation, or any other application where substantial web content is generated. Using the AdSense API, developers can let users sign up for AdSense through their site or program, generate detailed performance reports for users, and choose how the AdSense revenue is shared with our revenue sharing program. Additionally users can create accounts to store publisher website information, then generate ad code snippets and filter out unwanted ads. A previous SOAP version has been deprecated in favor of REST with responses formatted as JSON.

Summary | SDKs (9) | Articles (5) | How To (0) | Sample Source Code (0) | Libraries (6) | Developers (84) | Followers (139) | Comments (1)

SPECS

API Endpoint	http://code.google.com/apis/adsense/developer/adsense_api_services.html#wsdl_list
API Portal / Home Page	https://developers.google.com/adsense/
Primary Category	Advertising
Secondary Categories	SEO
SSL Support	Yes
API Forum / Message Boards	http://groups.google.com/group/AdSense-API/topics
Developer Support URL	Testing sandbox also available
Is the API Design/Description Non-Proprietary ?	No
Scope	Single purpose API
Device Specific	No
Docs Home Page URL	https://developers.google.com/adsense/
Architectural Style	REST
Supported Request Formats	URI Query String/CRUD
Supported Response Formats	JSON, XML
Is This an Unofficial API?	No
Is This a Hypermedia API?	Yes
Restricted Access (Requires Provider Approval)	No

Home » APIs » Google AJAX Libraries

Google AJAX Libraries API

Library | Application Development

The AJAX Libraries API is a content distribution network and loading architecture for the most popular open source JavaScript libraries. By using the Google AJAX API Loader's google.load() method, the API provides your applications with stable, reliable, high speed, globally available access to all of the most popular, open source JavaScript libraries including: jQuery, jQuery UI, prototype, script.aculo.us, MooTools, dojo, SWFObject, Yahoo! User Interface Library (YUI), Ext Core and Chrome Frame. Google works directly with the key stake holders for each library effort and accept the latest stable versions as they are released.

TRACK API

Summary | SDKs (0) | Articles (4) | How To (0) | Sample Source Code (0) | Libraries (0) | Developers (62) | Followers (135) | Comments (0)

SPECS

API Endpoint	https://ajax.googleapis.com/ajax/libs/
API Portal / Home Page	https://developers.google.com/speed/libraries/devguide
Primary Category	Library
Secondary Categories	Application Development
API Provider	Google
SSL Support	Yes
API Forum / Message Boards	http://groups.google.com/group/Google-AJAX-Search-API
Twitter URL	https://twitter.com/googledevs
Developer Support URL	http://code.google.com/support/bin/topic.py?topic=11330
Interactive Console URL	http://code.google.com/apis/ajax/playground/?exp=libraries#jquery
Terms Of Service URL	https://developers.google.com/speed/libraries/terms
Is the API Design/Description Non-Proprietary ?	No
Type	Web/Internet
Scope	Single purpose API
Device Specific	No
Docs Home Page URL	https://developers.google.com/speed/libraries/devguide
Supported Request Formats	JavaScript
Supported Response Formats	None Specified
Is This an Unofficial API?	No
Is This a Hypermedia API?	No
Restricted Access (Requires Provider Approval)	No

Home » APIs » Google Analytics Managment

Google Analytics Managment API

Analytics

Use the Google Analytics Management API to leverage Google's extensive website analytics tools. The API uses GET requests over HTTPS and JSON for requests/returns. Some of the functions available with the API are: List Account, Property and View (Profile) information for a user; manage properties and goals; manage user permissions; and upload cost data to Google Analytics for non-Google paid campaigns. See the extensive documentation for code samples, detailed method descriptions, and more.

TRACK API

Summary | SDKs (2) | Articles (24) | How To (3) | Sample Source Code (7) | Libraries (9) | Developers (59) | Followers (198) | Comments (0)

SPECS

API Endpoint	https://www.googleapis.com/analytics/v3
API Portal / Home Page	https://developers.google.com/analytics/devguides/config/mgmt/v3/mgmtReference/
Primary Category	Analytics
API Provider	Google
SSL Support	Yes
API Forum / Message Boards	https://developers.google.com/analytics/community/
Authentication Model	OAuth 2
Is the API Design/Description Non-Proprietary ?	No
Scope	Single purpose API
Device Specific	No
Docs Home Page URL	https://developers.google.com/analytics/devguides/config/mgmt/v3/mgmtReference/
Architectural Style	REST
Supported Request Formats	URI Query String/CRUD
Supported Response Formats	JSON, XML
Is This an Unofficial API?	No
Is This a Hypermedia API?	Yes
Restricted Access (Requires Provider Approval)	No

FOLLOWERS (198)

Google App Engine API

Tools | Cloud | Webhooks

Use the Google App Engine platform to run web apps on Google's infrastructure, the same scalable systems that run Google applications. Use the platform to run your apps reliably, even if you have heavy load and data needs. The environment has dynamic web serving, persistent storage with queries, sorting and transactions, automatic scaling and load balancing, APIs for authenticating users and sending email using Google Accounts, and a local testing environment. Google App Engine apps are implemented using the Python programming language, with support for most of the Python standard library. App Engine recently unveiled its second language: Java. This release includes the Java runtime, integration with Google Web Toolkit, and a Google Plugin for Eclipse, giving you an end-to-end Java solution for AJAX web applications.

SPECS

API Portal / Home Page	https://developers.google.com/appengine/
Primary Category	Tools
Secondary Categories	Cloud, Webhooks
SSL Support	Yes
API Forum / Message Boards	http://groups.google.com/group/google-appengine
Twitter URL	http://twitter.com/#!/app_engine
Terms Of Service URL	http://code.google.com/appengine/terms.html
Is the API Design/Description Non-Proprietary ?	No
Scope	Single purpose API
Device Specific	No
Docs Home Page URL	https://developers.google.com/appengine/
Supported Request Formats	None Specified
Supported Response Formats	None Specified
Is This an Unofficial API?	No
Is This a Hypermedia API?	No
Restricted Access (Requires Provider Approval)	No

Home » APIs » Google Earth

Google Earth API

Mapping | 3D

The Google Earth Engine API allows developers to run algorithms on georeferenced imagery and vectors stored on Google's infrastructure. The Google Earth Engine API provides a library of functions which may be applied to imagery for display and analysis. Earth Engine's public data catalog contains a large amount of publicly available imagery.

TRACK API

Summary | SDKs (3) | Articles (10) | How To (0) | Sample Source Code (1) | Libraries (0) | Developers (52) | Followers (285) | Comments (0)

SPECS

API Endpoint	http://www.google.com
API Portal / Home Page	https://developers.google.com/earth-engine/?hl=en
Primary Category	Mapping
Secondary Categories	3D
API Provider	Google
SSL Support	Yes
API Forum / Message Boards	http://groups.google.com/group/kml-support/topics
Twitter URL	https://twitter.com/googledevs
Developer Support URL	https://developers.google.com/earth/forums/index
Authentication Model	API Key
Terms Of Service URL	http://code.google.com/apis/maps/terms.html
Is the API Design/Description Non-Proprietary ?	No
Scope	Single purpose API
Device Specific	No
Docs Home Page URL	https://developers.google.com/earth-engine/?hl=en
Architectural Style	Indirect
Supported Request Formats	JSON, JavaScript
Supported Response Formats	GeoJSON, JSON
Is This an Unofficial API?	No
Is This a Hypermedia API?	No
Restricted Access (Requires Provider Approval)	No

Home » APIs » Google Homepage

Google Homepage API

Widgets

From their site: The Google Gadgets API provides a way to put third-party content onto the Google homepage. The main use of the Google Gadgets API is to create gadgets (mini-applications) that users can add to their Google personalized homepages and other Google properties. For example, you could create a photo album gadget, or a gadget that displays an interactive game. The API is designed to be simple enough so that anyone can turn his or her web page or application into a gadget.

TRACK API

Summary | SDKs (0) | Articles (2) | How To (0) | Sample Source Code (3) | Libraries (0) | Developers (101) | Followers (112) | Comments (3)

SPECS

API Portal / Home Page	https://developers.google.com/gadgets/
Primary Category	Widgets
API Forum / Message Boards	http://groups.google.com/group/Google-Gadgets-API
Is the API Design/Description Non-Proprietary ?	No
Scope	Single purpose API
Device Specific	No
Docs Home Page URL	https://developers.google.com/gadgets/
Architectural Style	REST
Supported Request Formats	JavaScript
Supported Response Formats	XML
Is This an Unofficial API?	No
Is This a Hypermedia API?	No
Restricted Access (Requires Provider Approval)	No

Home » APIs » YouTube

TRACK API

YouTube API

Video | Media

The Data API allows users to integrate their program with YouTube and allow it to perform many of the operations available on the website. It provides the capability to search for videos, retrieve standard feeds, and see related content. A program can also authenticate as a user to upload videos, modify user playlists, and more. This integration can be used for a variety of uses such as developing a web application allowing users to upload video to YouTube, or a device or desktop application that brings the YouTube experience to a new platform. The Data API gives users programmatic access to the video and user information stored on YouTube. This can be used to personalize a web site or application with the user's existing information as well as perform actions like commenting on and rating videos. This RESTful API provides responses in XML format.

Summary | SDKs (10) | Articles (68) | How To (2) | Sample Source Code (100) | Libraries (17) | Developers (786) | Followers (1409) | Comments (23)

SPECS

API Endpoint	https://www.youtube.com/iframe_api
API Portal / Home Page	https://developers.google.com/youtube/
Primary Category	Video
Secondary Categories	Media
API Provider	Google
SSL Support	No
API Forum / Message Boards	https://groups.google.com/group/youtube-api/
Twitter URL	https://twitter.com/YouTubeDev/
Developer Support URL	http://code.google.com/support/bin/topic.py?topic=12357
Interactive Console URL	http://code.google.com/apis/ajax/playground/?exp=youtube#simple_embed
Authentication Model	API Key
Terms Of Service URL	http://code.google.com/apis/youtube/terms.html
Is the API Design/Description Non-Proprietary ?	No
Scope	Single purpose API
Device Specific	No
Docs Home Page URL	https://developers.google.com/youtube/
Architectural Style	REST
Supported Request Formats	Atom, GData, RSS, URI Query String/CRUD, Atom Publishing Protocol (Atom/RSS)
Supported Response Formats	XML, JSON, GData, Atom, RSS
Is This an Unofficial API?	No
Is This a Hypermedia API?	Yes
Restricted Access (Requires Provider Approval)	No

Home » APIs » Instagram Graph

Instagram Graph API

Photos | Mobile | Social

Instagram is a photo sharing iPhone app and service. Users take photos and can share them with Instagram contacts, as well as friends on other social networks like Twitter and Facebook. The Instagram API provides access to user authentication, friend connections, photos and all the other elements of the iPhone app--including uploading new media.

TRACK API

Summary | SDKs (15) | Articles (39) | How To (1) | Sample Source Code (6) | Libraries (12) | Developers (57) | Followers (777) | Comments (0)

SPECS

API Endpoint	graph.facebook.com
API Portal / Home Page	https://developers.facebook.com/docs/instagram-api/v2.11
Primary Category	Photos
Secondary Categories	Mobile, Social
API Provider	Instagram
SSL Support	Yes
API Forum / Message Boards	https://groups.google.com/forum/#!forum/instagram-api-developers
Twitter URL	https://twitter.com/instagram
Developer Support URL	http://getsatisfaction.com/instagram
Interactive Console URL	https://developers.facebook.com/tools/explorer/145634995501895/
Authentication Model	Token
Version	2.11
Terms Of Service URL	https://developers.facebook.com/policy/
Is the API Design/Description Non-Proprietary ?	Yes
Type	Web/Internet
Scope	Single purpose API
Device Specific	No
Docs Home Page URL	https://developers.facebook.com/docs/instagram-api/v2.11
Architectural Style	REST
Supported Request Formats	URI Query String/CRUD
Supported Response Formats	JSON
Is This an Unofficial API?	No
Is This a Hypermedia API?	No
Restricted Access (Requires Provider Approval)	No

Home » APIs » Last.fm

Last.fm API

Music

The Last.fm API gives users the ability to build programs using Last.fm data, whether on the web, the desktop or mobile devices. The RESTful API allows for read and write access to the full slate of last.fm music data resources - albums, artists, playlists, events, users, and more. It allows users to call methods that respond in either XML or JSON.

TRACK API

Summary | SDKs (10) | Articles (19) | How To (0) | Sample Source Code (6) | Libraries (19) | Developers (249) | Followers (511) | Comments (2)

SPECS

API Endpoint	http://ws.audioscrobbler.com/2.0/
API Portal / Home Page	http://www.last.fm/api
Primary Category	Music
API Provider	Last.fm
SSL Support	No
API Forum / Message Boards	http://www.last.fm/group/Last.fm+Web+Services/forum/21604
Support Email Address	office@last.fm
Developer Support URL	http://www.audioscrobbler.net/wiki/
Authentication Model	API Key, Shared Secret, Token
Version	2.0
Terms Of Service URL	http://www.last.fm/api/tos
Is the API Design/Description Non-Proprietary ?	Yes
Type	Web/Internet
Scope	Single purpose API
Device Specific	No
Docs Home Page URL	http://www.last.fm/api
Architectural Style	REST
Supported Request Formats	URI Query String/CRUD
Supported Response Formats	JSON, XML
Is This an Unofficial API?	No
Is This a Hypermedia API?	No
Restricted Access (Requires Provider Approval)	No

Home » APIs » LinkedIn

LinkedIn API

Social | Enterprise

LinkedIn is the world's largest business social networking hub. Launched in 2003, LinkedIn has millions of users and is implemented in over 200 countries. One purpose of the site is to allow registered users to maintain a list of contact details of people with whom they have some level of relationship, called Connections. Users can invite anyone (whether a site user or not) to become a connection. The LinkedIn API is a RESTful platform that provides a simple, consistent representation of people, companies, jobs, and the interactions and relationships between them. Our query language lets you read data in XML and JSON at the granularity and aggregation that you choose. Use OAuth 1.0a to authorize users and begin making REST API calls using any programming language.The API's access is restricted to authorized developers.

TRACK API

Summary | SDKs (5) | Articles (32) | How To (0) | Sample Source Code (1) | Libraries (18) | Developers (62) | Followers (1215) | Comments (25)

SPECS

API Endpoint	http://api.linkedin.com/v1/
API Portal / Home Page	https://developer.linkedin.com/docs
Primary Category	Social
Secondary Categories	Enterprise
API Provider	LinkedIn
SSL Support	Yes
Twitter URL	https://twitter.com/LinkedIn
Support Email Address	atrachtenberg@linkedin.com
Developer Support URL	https://developer.linkedin.com/support
Interactive Console URL	https://developer.linkedin.com/rest-console
Authentication Model	OAuth 2
Version	1
Terms Of Service URL	https://developer.linkedin.com/documents/linkedin-apis-terms-use
Is the API Design/Description Non-Proprietary ?	No
Type	Web/Internet
Scope	Single purpose API
Device Specific	No
Docs Home Page URL	https://developer.linkedin.com/docs/rest-api
Architectural Style	REST
Supported Request Formats	JSON, URI Query String/CRUD, XML
Supported Response Formats	JSON, XML
Is This an Unofficial API?	No
Is This a Hypermedia API?	Yes
Restricted Access (Requires Provider Approval)	Yes

Home » APIs » Bing

TRACK API

Bing API

Search | Machine Learning

[The Bing API is now the Bing Web Search API. This profile is being maintained purely for historical and research purposes.]

The Bing API provides an experience similar to Bing.com/search by returning search results that Bing determines are relevant to a user's query. The results include Web pages and may also include images, videos, and more. Free trial keys are available here: https://azure.microsoft.com/en-us/try/cognitive-services/?api=bing-web-search-api.

Summary | SDKs (0) | Articles (10) | How To (1) | Sample Source Code (16) | Libraries (3) | Developers (67) | Followers (247) | Comments (0)

SPECS

API Endpoint	https://api.cognitive.microsoft.com/bing/v7.0/search[?q]
API Portal / Home Page	https://azure.microsoft.com/en-us/services/cognitive-services/bing-web-search-api/
Primary Category	Search
Secondary Categories	Machine Learning
Is the API related to any other API ?	Bing Custom Search, Bing Image Search, Bing Video Search, Bing News Search, Bing Autosuggest, Bing Spell Check.
API Provider	Microsoft
SSL Support	Yes
API Forum / Message Boards	https://stackoverflow.com/questions/tagged/bing-search
Interactive Console URL	https://azure.microsoft.com/en-us/services/cognitive-services/bing-web-search-api/
Authentication Model	API Key
Version	7
Terms Of Service URL	https://docs.microsoft.com/en-us/azure/cognitive-services/bing-web-search/use-and-display-requirements
Is the API Design/Description Non-Proprietary ?	No
Type	Web/Internet
Scope	Single purpose API
Device Specific	No
Docs Home Page URL	https://docs.microsoft.com/en-us/azure/cognitive-services/bing-web-search/
Description File URL (if public)	https://docs.microsoft.com/en-us/rest/api/cognitiveservices/bing-web-api-v7-reference
Architectural Style	REST
Supported Request Formats	URI Query String/CRUD
Supported Response Formats	JSON
Is This an Unofficial API?	No
Is This a Hypermedia API?	Yes
Restricted Access (Requires Provider Approval)	No

Home » APIs » Microsoft Bing Maps

Microsoft Bing Maps API

Mapping | Viewer

Bing Maps API and Interactive SDK features an AJAX Map Control. Use BM to build maps which can include routes and traffic info. Gives developers the ability to code the controls, shapes, and layers of the maps, and can summon the birds-eye, 3D, and aerial imagery. For commercial applications Bing Maps Web Services is a set of programmable SOAP services that allow users to integrate maps and imagery, driving directions, and other location features into a Web application.

TRACK API

Summary | SDKs (2) | Articles (8) | How To (0) | Sample Source Code (17) | Libraries (2) | Developers (174) | Followers (299) | Comments (0)

SPECS

API Endpoint	See-> http://dev.live.com/virtualearth/sdk/
API Portal / Home Page	http://www.microsoft.com/maps/choose-your-bing-maps-API.aspx
Primary Category	Mapping
Secondary Categories	Viewer
API Provider	Microsoft
SSL Support	Yes
API Forum / Message Boards	http://social.msdn.microsoft.com/forums/en-US/vemapcontroldev/threads/
Developer Support URL	http://msdn2.microsoft.com/en-us/library/bb429619.aspx
Interactive Console URL	http://www.bingmapsportal.com/isdk/ajaxv7
Authentication Model	API Key
Terms Of Service URL	http://www.microsoft.com/virtualearth/product/terms.html
Is the API Design/Description Non-Proprietary ?	No
Scope	Single purpose API
Device Specific	No
Docs Home Page URL	http://www.microsoft.com/maps/choose-your-bing-maps-API.aspx
Architectural Style	Indirect
Supported Request Formats	KML, XML, GeoRSS, JavaScript
Supported Response Formats	KML
Is This an Unofficial API?	No
Is This a Hypermedia API?	No
Restricted Access (Requires Provider Approval)	No

[redacted]

[redacted]

Shopping.com API

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

SPECS

API Endpoint	[redacted]
API Portal / Home Page	[redacted]
Primary Category	[redacted]
SSL Support	Yes
API Forum / Message Boards	[redacted]
Developer Support URL	[redacted]
Terms Of Service URL	[redacted]
Is the API Design/Description Non-Proprietary ?	No
Type	[redacted]
Scope	[redacted]
Device Specific	No
Architectural Style	[redacted]
Supported Request Formats	[redacted]
Supported Response Formats	XML
Is This an Unofficial API?	No
Is This a Hypermedia API?	No
Restricted Access (Requires Provider Approval)	No

Home » APIs » Twilio

TRACK API

Twilio API

Telephony | Cloud | Text-to-Speech | Voice | Webhooks

Twilio provides a simple hosted API and markup language for businesses to quickly build scalable, reliable and advanced voice and SMS communications applications. Twilio provides a telephony infrastructure web service "in the cloud", enabling web programmers to integrate real-time phone calls into their applications. Twilio's simple Pay-As-You-Go pricing model means customers pay for capacity only when they need it, not before. Voice applications written on Twilio's cloud solution scale transparently, eliminating programming, operational and contractual headaches frequently associated with rapid growth or traffic spikes. Twilio provides a cloud API for voice and SMS communications that leverages existing web development skills, resources and infrastructure. It minimizes the learning curve required to build advanced, reliable voice communications applications that solve critical business needs. The syntax and programming model are focused on making application development as close to the request/response model of web application development as possible. The API uses a RESTful interface and responses are formatted in HTML, JSON, XML or CSV.

Summary | SDKs (4) | Articles (103) | How To (4) | Sample Source Code (10) | Libraries (34) | Developers (359) | Followers (475) | Comments (0)

SPECS

API Endpoint	https://api.twilio.com/2010-04-01
API Portal / Home Page	http://www.twilio.com
Primary Category	Telephony
Secondary Categories	Cloud, Text-to-Speech, Voice, Webhooks
API Provider	Twilio
SSL Support	Yes
API Forum / Message Boards	https://www.twilio.com/blog
Twitter URL	https://twitter.com/twilio
Support Email Address	help@twilio.com
Developer Support URL	https://support.twilio.com/hc/en-us
Authentication Model	API Key, HTTP Basic Auth
Terms Of Service URL	http://www.twilio.com/legal/tos
Is the API Design/Description Non-Proprietary ?	No
Type	Web/Internet
Scope	Single purpose API
Device Specific	No
Docs Home Page URL	https://www.twilio.com/docs/api/rest
Architectural Style	REST
Supported Request Formats	URI Query String/CRUD
Supported Response Formats	CSV, HTML, JSON, XML
Is This an Unofficial API?	No
Is This a Hypermedia API?	Yes
Restricted Access (Requires Provider Approval)	No

TRACK API

Twilio SMS API

Messaging | Telephony | USA

Twilio provides a simple hosted API and markup language for businesses to quickly build scalable, reliable and advanced voice and SMS communications applications. Twilio provides a telephony infrastructure web service "in the cloud", enabling web programmers to integrate real-time phone calls into their applications. Twilio's simple Pay-As-You-Go pricing model means customers pay for capacity only when they need it, not before. Voice applications written on Twilio's cloud solution scale transparently, eliminating programming, operational and contractual headaches frequently associated with rapid growth or traffic spikes. The SMS API allows users to send and receive SMS messages from their applications. The API can implement short code technology to allow both faster and higher volume sms messaging. Example uses of the API include sending customers automatic SMS messages, multi-factor authentication and interactive polling. The API uses a RESTful interface and responses are formatted in XML.

Summary | SDKs (0) | Articles (22) | How To (1) | Sample Source Code (5) | Libraries (0) | Developers (174) | Followers (205) | Comments (0)

SPECS

API Portal / Home Page	http://www.twilio.com/sms
Primary Category	Messaging
Secondary Categories	Telephony, USA
SSL Support	Yes
API Forum / Message Boards	http://forum.twilio.com/twilio/?from_gsfn=true
Is the API Design/Description Non-Proprietary ?	No
Scope	Single purpose API
Device Specific	No
Docs Home Page URL	http://www.twilio.com/sms
Architectural Style	REST
Supported Request Formats	None Specified
Supported Response Formats	XML
Is This an Unofficial API?	No
Is This a Hypermedia API?	No
Restricted Access (Requires Provider Approval)	No

Home » APIs » Wikipedia

Wikipedia API

Reference | Wiki

The unofficial Wikipedia API. Because Wikipedia is built using MediaWiki, which in turn supports an API, Wikipedia does as well. This provides developers code-level access to the entire Wikipedia reference. The goal of this API is to provide direct, high-level access to the data contained in the MediaWiki databases. Client programs can use the API to login, get data, and post changes. The API supports thin web-based JavaScript clients, such as Navigation popups or LiveRC, end-user applications (such as vandal fighter), and can be accessed by another web site (tool server's utilities). The API uses RESTful calls and supports a wide variety of formats including XML, JSON, PHP, YAML and others.

TRACK API

Summary | SDKs (3) | Articles (3) | How To (0) | Sample Source Code (0) | Libraries (4) | Developers (83) | Followers (518) | Comments (1)

SPECS

API Endpoint	https://en.wikipedia.org/w/api.php
API Portal / Home Page	http://www.mediawiki.org/wiki/API
Primary Category	Reference
Secondary Categories	Wiki
API Provider	Wikipedia
SSL Support	Yes
Authentication Model	OAuth 1
Is the API Design/Description Non-Proprietary ?	Yes
Type	Web/Internet
Scope	Single purpose API
Device Specific	No
Docs Home Page URL	http://www.mediawiki.org/wiki/API
Architectural Style	REST
Supported Request Formats	URI Query String/CRUD
Supported Response Formats	JSON, XML, JSONFM, PHP
Is This an Unofficial API?	Yes
Is This a Hypermedia API?	Yes
Restricted Access (Requires Provider Approval)	No

Yahoo Local Search API

Search | Mapping | Localization

The Yahoo! Local Search APIs give users access to a comprehensive database of information and content in Yahoo! Local ranging from business information to user comments to ratings. Data includes business address and phone, category, rating, distance, URL and traffic alerts. Yahoo! Local makes a great addition to any mashup, bringing in location-based relevancy and the additional context of what real people have experienced in these places. Version 3 of Yahoo! Local Search adds even more features! Users can now request individual business records by using the Yahoo! Local listing id. Yahoo! supports search categories to help users narrow their queries. Finally, results now return the first few lines of the last user review for each listing. The API uses a RESTful protocol and responses are formatted in either XML, JSON and PHP.

Summary | SDKs (0) | Articles (2) | How To (0) | Sample Source Code (1) | Libraries (0) | Developers (55) | Followers (198) | Comments (0)

SPECS

API Endpoint	http://local.yahooapis.com/LocalSearchService/V3/localSearch
API Portal / Home Page	http://developer.yahoo.com/search/local/localSearch.html
Primary Category	Search
Secondary Categories	Mapping, Localization
SSL Support	No
API Forum / Message Boards	http://groups.yahoo.com/group/yws-search-general/
Terms Of Service URL	http://info.yahoo.com/legal/us/yahoo/api/api-2140.html
Is the API Design/Description Non-Proprietary ?	No
Scope	Single purpose API
Device Specific	No
Docs Home Page URL	http://developer.yahoo.com/search/local/localSearch.html
Architectural Style	REST
Supported Request Formats	URI Query String/CRUD, PHP
Supported Response Formats	JSON, XML
Is This an Unofficial API?	No
Is This a Hypermedia API?	Yes
Restricted Access (Requires Provider Approval)	No

Home » APIs » Yelp Fusion

TRACK API

Yelp Fusion API

Recommendations | Business | Directories | eCommerce | Localization | Search

The Yelp Fusion APIs are RESTful APIs and users can retrieve business review and rating, information for a particular geographic region or location.display review information for a particular business, determine accurate neighborhood name information for a particular location, track recent reviews for a particular business, display pictures of highly rated local businesses and of the top reviewers for that business, determine a particular business' review and rating information based on the phone number for that business. The default output is JSON. This output format was chosen due to the availability of JSON parsers in many languages. The following Yelp Fusion APIs are available: Search, Phone Number Search, Business Search, Transaction, Reviews, and Autocomplete - each API has a separate ProgrammableWeb entry.

Summary | SDKs (22) | Articles (15) | How To (0) | Sample Source Code (7) | Libraries (1) | Developers (60) | Followers (292) | Comments (3)

SPECS

API Endpoint	http://api.yelp.com
API Portal / Home Page	https://www.yelp.com/developers
Primary Category	Recommendations
Secondary Categories	Business, Directories, eCommerce, Localization, Search
API Provider	Yelp
SSL Support	Yes
API Forum / Message Boards	https://github.com/yelp/yelp-fusion
Twitter URL	https://twitter.com/yelpengineering
Support Email Address	api@yelp.com
Authentication Model	API Key
Terms Of Service URL	http://www.yelp.com/developers/getting_started/api_terms
Is the API Design/Description Non-Proprietary ?	No
Type	Web/Internet
Scope	Single purpose API
Device Specific	No
Docs Home Page URL	http://www.yelp.com/developers
Architectural Style	REST
Supported Request Formats	JSON
Supported Response Formats	None Specified
Is This an Unofficial API?	No
Is This a Hypermedia API?	Yes
Restricted Access (Requires Provider Approval)	No