UNIVERSIDAD NACIONAL DE SAN MARTIN

RCSI
https://revistas.unsm.edu.pe/index.php/rcsi
e-ISSN: 2709-992X

Original article
Artículo original
Jan-Jun, 2025

Web API Ecosystems: Socio-Technical Analysis and Characterization

Ecosistemas de APIs Web: Análisis y Caracterización Socio-Técnica

Casas, Sandra[1*]
Constanzo, Marcela[1]
Vidal, Graciela[1]
Cruz, Diana[1]

[1]Universidad Nacional de la Patagonia Austral. Instituto de Tecnologia Aplicada, Grupo de I+D en Ingeniería de Software Pragmática (GISP). Río Gallegos, Argentina

Received: 06 Aug. 2024 | **Accepted:** 30 Oct. 2024 | **Published:** 20 Jan. 2025

Corresponding author*: sicasas@uarg.unpa.edu.ar

How to cite this article: Casas, S., Constanzo, M., Vidal, G., & Cruz, D. (2025). Web API Ecosystems: Socio-Technical Analysis and Characterization. *Revista Científica de Sistemas e Informática*, *5*(1), e757. https://doi.org/10.51252/rcsi.v5i1.757

ABSTRACT

Digital transformation and the API economy have given rise to digital ecosystems, where Web APIs are the main assets. Web API Ecosystems (WEAs) bring together people, companies, services, resources, tools, and innovative development activities and processes on a platform. In the field of Software Engineering, WEAs are Software Ecosystems (SECOs). One way to examine and understand complex systems is the Socio-Technical (S-T) approach, which reduces the risk of systems not meeting organizational objectives. According to various published review studies, there are no S-T studies reported in relation to Web APIs as in SECO. In order to identify categories of analysis, we performed an exploratory and descriptive case study applied to four current WEAs, applying an S-T approach. We found and classified the components of the social, infrastructure, and standards dimensions into five main categories, which also include subcategories. Relationships with the context have been classified into three categories. Finally, we discuss and compare our observations with the most widely disseminated literature regarding API Management and Software Ecosystems.

Keywords: API management; Web API; software ecosystem; socio-technical system

RESUMEN

La transformación digital y la economía API han dado origen a ecosistemas digitales, donde las APIs Web son los activos principales. Los Ecosistemas de APIs Web (EAW) reúnen en una plataforma a personas, empresas, servicios, recursos, herramientas, innovadoras actividades y procesos de desarrollo. En el ámbito de la Ingeniería de Software, los EAW son Ecosistemas de Software (SECO). Una forma de examinar y entender los sistemas complejos es el enfoque Socio-Técnico (S-T), el cual disminuye el riesgo de que los sistemas no cumplan los objetivos organizacionales. Según diversos estudios de revisión publicados, no se registran estudios S-T en relación con las API web como en SECO. Con el objetivo de identificar categorías de análisis, ejecutamos un caso de estudio exploratorio y descriptivo aplicado a cuatro EAW actuales, aplicando un enfoque S-T. Encontramos y clasificamos los componentes de las dimensiones social, infraestructura y normas en cinco categorías principales, las cuales además incluyen subcategorías. Las relaciones con el contexto se han clasificado en tres categorías. Finalmente, discutimos y comparamos nuestras observaciones con la literatura más difundida respecto a la gestión de API y los Ecosistemas de Software

Palabras clave: administración de API; API Web; ecosistema de software; sistema socio-técnico

1. INTRODUCTION

Since the creation of the Representational State Transfer (REST) architecture in 2000 (Fielding & Taylor, 2000), 'apification' (the adoption of Web APIs) has been ongoing. Web APIs are the backbone of modern software, enabling software applications to connect via the web. Organizations of all types (governments, corporations and businesses) digitally connect their applications to user interfaces that are accessible to customers, employees and remote users as the primary means of interaction. Web APIs clearly exemplify the extent of globalization in software development.

API Economy (Tan et al., 2016) and Digital Transformation have given rise to a market for digital products and an industry that tracks innovation and progress in leaps and bounds around Web APIs. This sector (market+industry) includes products ranging from APIs themselves to tools that support the API lifecycle (design, testing, etc.), API management solutions, API documentation platforms, API security tools, API dashboards, among others. The industry has evolved from API products to platforms and from companies of all sizes to digital ecosystems (Apidays, 2022).

According to Ofoeda et al. (2019) scientific research on Web APIs in the fields of software development and distributed systems has focused on technical and/or technological problems and solutions. They point out that the research is atheoretical and recommend that Web API research should also be considered from a socio-technical (S-T) perspective. On the other hand Raatikainen et al. (2021) point out that scientific literature still does not sufficiently distinguish between static and Web APIs, and criticize that research often proposes guidelines or practices instead of characterizing the state of industrial practice in terms of challenges and proven solutions.

Due to the impact of Digital Platforms or Platform Ecosystems (PE) on the supply of products and services, competition between firms, and the generation of new business from the economy and innovation, studying PE from different perspectives is considered to be a topic of multidisciplinary interest (Kapoor et al., 2021). In turn, Software Ecosystems (SECO) is an emerging topic of study in Software Engineering (Manikas & Hansen, 2013). SECO research faces various challenges, since technical aspects of software development are mixed with social issues. Therefore, S-T modelling is also needed in SECO and is still a subject of inquiry (Jansen et al., 2015).

According to Baxter & Sommerville (2011), the motivation for applying the S-T perspective stems from the idea that not doing so increases the risk that a system will not make the expected contributions to organizational goals. A technical view provides information on how a system meets technical requirements but does not take into account the complex relationships between an organization and the actors that undertake and support the various business processes. In order to gain a coherent understanding of how the social system (network actors) adapts/re-aligns objectives to support technical aspects (technical changes and innovations within the PE), an S-T approach is deemed appropriate. S-T systems are inherently dynamic and evolve through the recursive configuration of technical infrastructure and social constructs, which is reflected in actions that change entities at the technical, task, structure and actor levels(Dremel et al., 2020).

Web API Ecosystems (EAWs) are complex and dynamic networks of social and technical components -actors and tools, respectively- that evolve under a system of rules with innovations for software development. In this paper, we present an exploratory and descriptive case study of four current EAWs, with the aim of identifying categories of analysis that deepen our previous findings (Casas et al., 2023), from an S-T approach. Finally, we discuss the categories of analysis in relation to the available literature on API management and possible contributions to SECO modelling.

Related work

This study is framed within several theoretical contexts: Web APIs and API management, Software Ecosystems, Platform Ecosystems and S-T systems.

From the fields of software engineering and distributed systems, Web APIs have been the subject of research for two decades. According to several reviews, the dominant research topics have focused on technological aspects such as usability (Mosqueira-Rey et al., 2018), evolution (Koci et al., 2019), documentation (Nybom et al., 2017; Cummaudo et al., 2019), specification (Casas et al., 2021) and development (Beaulieu et al., 2022). In particular, the systematic review of the literature published between 2010 and 2018 carried out by Ofoeda et al. (2019), comprising 104 articles, provides an overview of API research, synthesizes the research of this period and describes the areas that require further attention. The findings suggest that the dominant themes in Web API research relate to design and usability, focusing on the technical domain of the classification scheme. The study also suggests that API research is largely atheoretical. Most studies tend to use experiments and code samples as research methods, as opposed to the dominant qualitative and quantitative methods. The authors identify the approach from an S-T perspective as a challenge for API research, as suggested in Casas et al. (2021).

The administration or management of APIs is also an emerging topic. A classic reference for Web API practitioners and researchers is De (2017), which deals with API administration or management. The author generally refers to an API service management platform or an API enterprise platform. He provides the following concepts: "An API management platform helps an organization publish APIs to internal, partner, and external developers and unlock the potential of its assets. It provides the core capabilities to ensure a successful API program through developer engagement, business insight, analytics, security and safety. An API management platform helps companies accelerate reach across digital channels, drive partner adoption, monetize digital assets and provide analytics to optimize Digital Transformation investments. The text provides a complete description and analysis from a technical/technological perspective (design, documentation, testing, security, API version control, API analytics and governance, monetization, developer portal).

Mathijssen et al. (2020) present a systematic literature review aimed at compiling API management practices and capabilities and proposing a comprehensive definition of the topic. The authors compile 24 unique definitions for the topic, 114 practices and 39 capabilities.

With an orientation towards the disciplines of business, economics and entrepreneurship, Kapoor et al. (2021) reviews and analyses the literature published between 1999 and 2019 on platform ecosystems (PE) from an S-T approach, which is in agreement with our line of enquiry. One of the main findings is that existing studies are mainly interested in the technical aspect of platforms and other tangible aspects (components and resources) and less in the social aspect.

Manikas & Hansen (2013) reviewed the literature on Software Ecosystems. They found that, apart from the technical perspective, studies with social and business perspectives were beginning to appear. A first definition that emerged from a technical perspective was formulated in Messerschmitt & Szyperski (2003), as a set of companies that function as a unit and interact with a common market of software and services and relate to each other.

The complexity of the relationships between the elements of the Software Ecosystem requires analytical techniques to predict the impact of strategic decisions on the network of suppliers, customers, partners and the ecosystem (McGregor, 2012). These challenges need the analysis of the field of Software Ecosystems from a different perspective. In this sense, the study of Software Ecosystems is an emerging topic that examines the relationships between firms in the software industry (Santos et al., 2012). Typically, these relationships are underpinned by a common technology platform and operate through the exchange of information, resources and artefacts.

Despite initial progress in SECO research -as evidenced by various mapping or systematic review studies, according to Manikas (2016), analytical models, case studies with real data and integrated tool support are yet to be thoroughly examined. A major obstacle to progress with concrete inputs for real industry is the lack of modelling support from SECO. According to Jansen et al. (2015), SECO modelling is important to provide information from representations that make analysis and comparison of “static” ecosystems possible, based on key concepts (e.g. organizations, relationships and flows) and existing methods (e.g. socio-technical networks and software supply networks). Some suggestions for modelling SECO are those presented by Campbell & Ahmed (2010), Boucharas et al. (2009) and Santos et al. (2012) detailed below.

Campbell & Ahmed (2010) proposed to analyze SECOs in three dimensions: business dimension (knowledge of the market and the key role of decision makers who identify SECO needs and platform extensions and maintain the product portfolio), architectural dimension (the definition and maintenance of platform technologies, i.e. what is needed to improve quality, interoperability between systems and performance evaluation as some of the aspects commonly analyzed in the platform) and social dimension (knowledge of the different actors playing different roles such as users, customers, suppliers and developers interacting with each other). Boucharas et al. (2009) proposed a SECO modelling perspective based on three levels. The First level is that of organizational scope, in which the objects of study are the actors and their relationships within the context of the organization belonging to a SECO. The Second Level is the Software Supply Networks (SSNs). The object of the study are their different relationships, and the focus is on the links between software, hardware and organizational services. The Third level is made up by the SECOs themselves, including the relationships among them. It is important that SECOs have well-defined boundaries, such as a technology, platforms, a market or a company.

In addition to considering the life cycle of ecosystems, Pereira dos Santos & Werner (2011, 2012) divide the activities that constitute the process of evolution from an application to an ecosystem into '3+1' dimensions: (i) 'architectural (or technical) dimension' - focused on the SECO platform (i.e., market, technology, infrastructure or organization) - (ii) 'business (or transactional) dimension' - focused on the SECO knowledge flow (i.e., artefacts, resources and information); (iii) 'social dimension' - focused on the SECO stakeholders and (iv) 'engineering and management dimension in SECO', which combines the three basic dimensions, and includes activities related to business.

The contribution by Lima et al. (2015) represents a Software Ecosystem as an environment where component repositories bring together stakeholders and software components. These repositories require S-T resources to support stakeholder relationships.

Casas et al. (2023) present the first S-T study of Web API ecosystems, which aims at understanding the structure and components of Web APIs. It identifies the preconditions, activities, dimensions and context of EAWs. The study involved the construction of an S-T framework to analyze three EAWs. This tool was developed based on the specific S-T systems approaches and contributions of Kingdon (1995), Baxter & Sommerville (2011) y Elatlassi et al. (2022).

2. MATERIALS AND METHODS

The qualitative study presented here consists of an exploratory and a descriptive case study of four current EAWs, following the guidelines of Creswell (2014), Fabbri (2020) and Runeson et al. (2012). Based on the formulation of categories of analysis of S-T elements, the aim is to deepen the understanding, conceptualization and characterization of the S-T components of EAWs. The question that we aim at answering is how the S-T components of EAWs may be categorized. Figure 1 shows the applied methodological design. In what follows, we describe the framework and then how its application was organized.

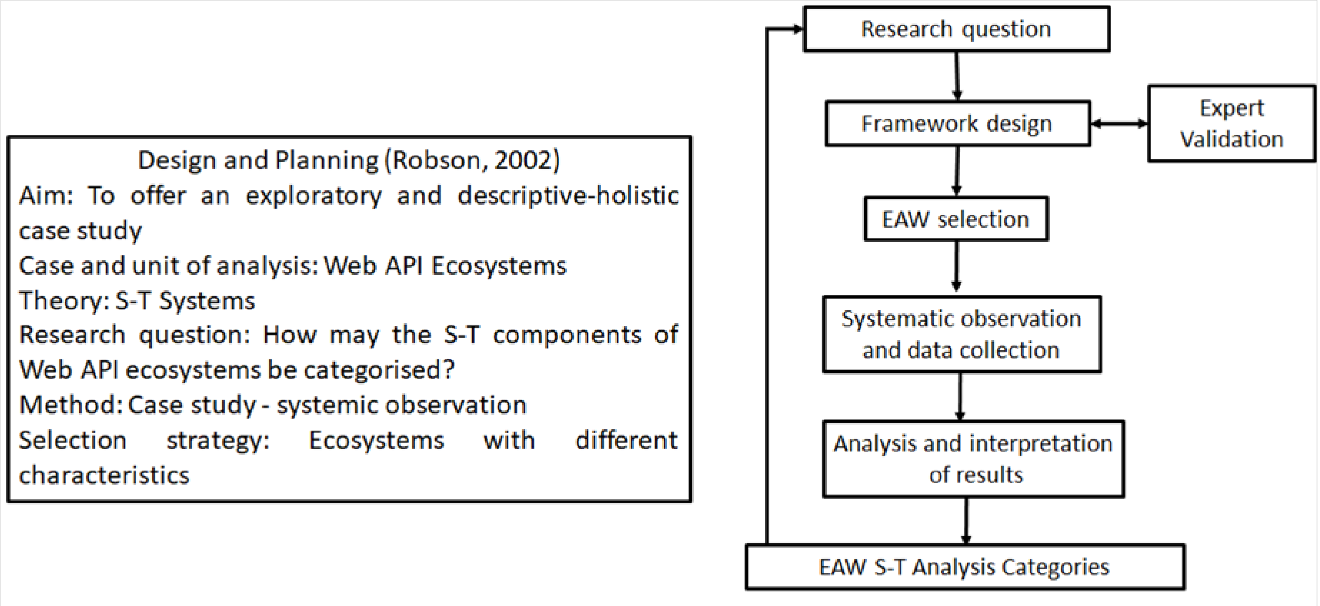

Figure 1. Methodological design

2.1. Framework

This tool has been developed and presented in Casas et al. (2023). Figure 2 shows the general structure, which is briefly described below.

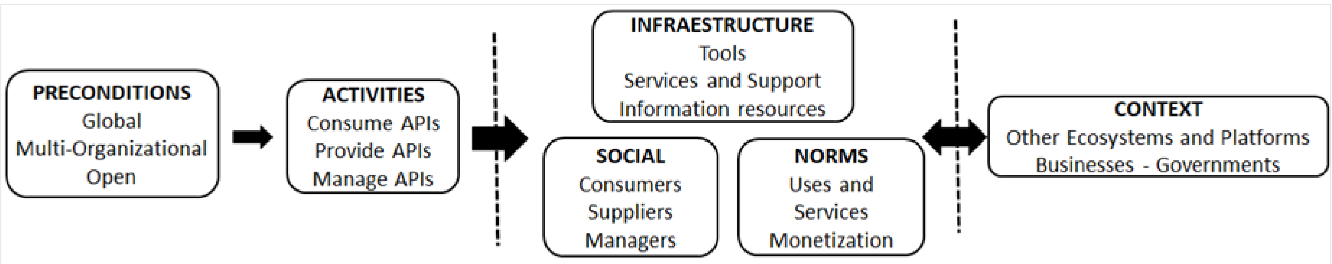

Figure 2. S-T Framework

a) Preconditions: A set of distinguishing characteristics that a system or PE must meet to be considered a Web API ecosystem.
   - Global: It is built around a digital platform.
   - Multi-organization: A variety of organizations (business, government, etc.) actively participate in the ecosystem.
   - Open: Any person, team or organization can participate in the ecosystem as long as they accept and abide by the established rules.

b) Activities: Activities are the consumption and production of Web APIs and their management on the platform.

c) Social dimension: people or companies that participate in or are part of the ecosystem.

d) Infrastructure dimension: set of tools, resources and services provided by the platform to consume or provide Web APIs.

e) Norms dimension: set of norms (rules, guidelines and policies) established to regulate the relationships and actions of the different actors in the ecosystem. They refer to the commitments made by consumers, providers and managers.

f) Context: Ecosystems, platforms, systems and/or companies with which the EAW establishes relationships and which are not part of the set of consumers or API providers.

2.2. Data collection

The EAWs selected were RapidAPI, APILayer, Nubentos and Apilanding. They have well differentiated characteristics in terms of their permanence and position in the market (level of maturity and consolidation), number of users (consumers and providers), number of APIs and domain. The process of applying the framework to the EAWs selected was carried out during the years 2022 (RapidAPI, APILayer, Nubentos) and 2023 (Apilanding).

The systematic or structured observation techniques offered by Creswell (2014) and Fabbri (2020) were implemented for data collection. A form based on the components of the framework was used to record observations and experiences. All content (documentation, videos, forms, templates, blogs, etc.) of each platform corresponding to the EAWs were examined and inspected. Based on the registration as a user, tools and wizards were provided, among others, to collect all the data from the EAWs. Once this data was obtained, the information was unified, processed and analyzed manually in order to define the categories of analysis.

The guidelines of Creswell (2014) were followed in the design and pursuit of the case study in order to minimize threats to validity, bias and limitations in terms of descriptive (internal) validity, external validity and generalizability of the findings.

3. RESULTS

3.1 Categories of analysis of Web API ecosystems

Figure 3 outlines the categories of analysis identified in this case study. Details of each analysis element are presented below (for a more thorough description, see Appendix).

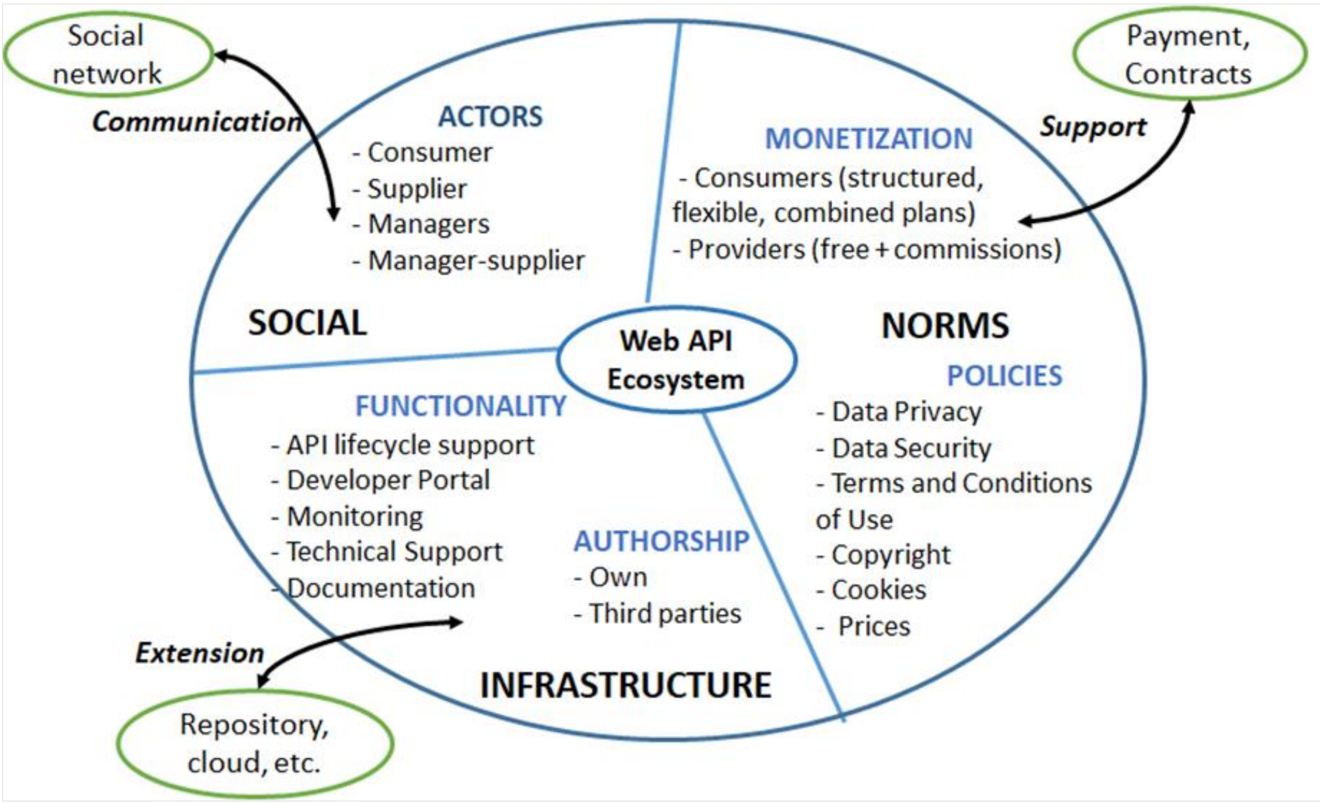

Figure 3. EAW S-T Analysis Categories

- Preconditions: All four EAWs fulfil the requirements set out above. Table 1 shows their general characteristics.

(i) Global: they are established and accessible through digital platforms (https://rapidapi.com/, https://www.nubentos.com/en/apistore, https://apilayer.com/ and https://apilanding.com/web/index.php).
(ii) Multi-organizational: all of them highlight the participation of different companies and individuals (suppliers, consumers, context).
(iii) Open: this was confirmed by the creation of accounts and the use of the different functionalities, both as consumers and as API providers.

Table 1. General characteristics of the EAWs analysed

EAW	Origin	Domain	APIs	Developers
RapidAPI	USA 2015	General	More than 40000	More than 4000000
APILayer	USA	General	More than 80	1200000
Nubentos	España-2018	Health	Mentions 30 suppliers	Mentions 10 consumer companies
Apilanding	Argentina	Finance	15 own + 218 third-parties	Mentions some well-known client companies

- Activities.- All four EAWs offer catalogues of Web APIs from various providers that can be consumed. The management of the APIs and the platform makes this possible.

- Social dimension.- In all EAWs, three actors can be distinguished: consumers, providers and managers. In the case of APILayer and Apilanding, they fulfil two roles, since they are not only managers but also providers. In the case of APILayer, the ecosystem is owned by Idera Inc. and Apilanding is owned by Sysworld Services S.A., while Nubentos and RapidAPI are organizations dedicated to the API business only. We identified four categories of actors: consumers, suppliers, managers and manager-suppliers.

- Infrastructure dimension.- In all EAWs, various tools, services and resources are provided. They have been categorized according to their function and authorship.

- Function.- The tools and resources are divided into five groups.

(i) API lifecycle supports tools, intended for vendors to publish, design, test, and manage versions of APIs.
(ii) API consumption tools, aimed at consumers, generally SDKs, CLIs, online editors, test tools, code generators in various languages that make it possible for APIs to be integrated.
(iii) Support tools and technical services, aimed at both consumers and providers, with facilities such as ticket generators, online chat, problem and/or bug forms.
(iv) Documentation resources, aimed at both consumers and providers, referring to any resource (tutorials, books, articles, blogs, blogs, examples, videos, get started, FAQs, etc.) that guide and helps to consume or provide an API.
(v) Monitoring services, consisting of information (metrics and indicators) related to the performance of the APIs and/or the gateway (platform).

- Authorship.- The development or construction of the main tools may be:

(i) Own: The company that manages and 'owns' the platform is the developer of the platform's main tools. This would be the case for RapidAPI, Apilanding and Nubentos.
(ii) Third party: most of the tools supporting the ecosystem activities are external or belong to third party developments. This would be the case of APILayer.

- Norm dimension.- Like other digital platforms or platform ecosystems, the platform publishes content (text, images, videos) that is the property of managers, providers and/or consumers. The use of this information and resources/services must be legal, authorised and controlled. Terms and conditions of use and service, copyright, privacy, security, etc. follow the scheme of most platforms or websites. The different categories are:

(i) Confidentiality/privacy policies.
(ii) Data security policies.
(iii) Terms of Use (services/platform).
(iv) Copyright.
(v) Cookie Policy.
(vi) Pricing policy.

All EAWs have a monetization model. Contracts with consumers are defined around the services/products offered and a scheme based on plans, according to the needs of the customers. The products and services are a combination of the use of the platform, care and support and the use of the APIs. However, alternatives are offered that make the system more flexible. The following are considered:

(i) Structured plans.
(ii) Flexible plans.
(iii) A combination of plans.

Another problem with the model is the supplier aspect. The publication of APIs in the catalogues is free of charge, and, in some cases, commissions are charged on the consumption of APIs on consumers. In the case of APILayer, the cloud hosting is variable in terms of prices, and in the case of RapidAPI, in the use of the products.

- Context.- The four ecosystems relate to the context with different objectives and in relation to the defined dimensions, among which we classify them as:

(i) Establishing communication with current or potential consumers and/or suppliers. All ecosystems are linked to accounts on various social networks.
(ii) Improving or extending infrastructure capabilities. Repositories and tools are used for integration, analytics, cloud hosting, specification, design and testing of APIs.
(iii) Supporting standards. Platforms or tools are used for contract underwriting and/or collections and payment management.

4. DISCUSSION

From the results obtained and our interpretation, we observe that:

a) The categories we have defined are different from other classifications. In the social dimension, four roles have been identified: consumer, supplier, manager and manager-supplier. However, De (2017) only recognizes consumers and manager-suppliers (without the presence of other suppliers). Our classification of actors is similar to that of Kapoor et al. (2021), with different labels (leader for manager, complementor for supplier and end-user for consumer). This difference between the EAWs we have analyzed and De (2017) further defines that they correspond to different types of PE, according to the classification offered by Kapoor et al. (2021). concepts correspond to internal platforms, while the EAWs RapidAPI, APILayer, Nubentos and Apilanding would be external PEs. The design of these categories also has an impact on the interactions between groups of actors, which trigger network externalities (network effects) from the same side or across sides, influencing the overall performance, success and sustainability of the PE when compared to competitors.
b) The 'community' is a social actor for De (2017) and Mathijssen et al. (2020), which is formed by the set of consumers and producers of APIs that interact in forums and blogs on the platform. It includes the 'community' as a social actor and shows that discussion forums and blogs describing developers' experiences contribute to the creation of an engaged developer community. No forums for community

actors were found in the survey of the four EAW platforms. The social actor 'community' is also not identified by Kapoor et al. (2021).

c) Another difference with De (2017) is the monetization model, which is more limited and focuses only on the consumer aspect.

In conclusion, we attribute the differences with De (2017) and Mathijssen et al. (2020) to the fact that API management is conceived as an activity that generates a narrower ecosystem. With less presence of other providers and the centrality of the dual role manager.

In relation to the Software Ecosystem, Santos et al. (2012) suggest studying the relationships between the companies involved in the software industry. In this paper, we have identified some of these relationships. First, the roles of the main actors establish the relationships. Then, when we analyze the context and classify the different companies according to their objectives, they complement the EAW. However, these relationships need to be analyzed in greater depth. In addition, we believe that the results of this study contribute to the construction of some existing SECO models, as some elements/dimensions can be partially mapped. For example, the three dimensions of Campbell & Ahmed (2010) -business, architecture and social-, have points of convergence with our approach. Along the same line, we find some common points with Boucharas et al. (2009), i.e., the definition of levels 1 (organizational) and 3 (relationship with other SECOs), which can benefit from our contributions. Finally, in relation to Santos & Werner (2012), the four dimensions they described (architectural, business, social and engineering and management) are present in our proposal, with the main difference that we do not model the evolution of the SECO, but its most relevant S-T aspects. On the other hand, Lima et al. (2015) have studied the S-T resources to support the relationships of the actors. However, we observe that some of them are present in the infrastructure dimension of our analysis, but at the same time we also observe the absence of other artefacts.

CONCLUSIONS

Based on the knowledge gap regarding the need to study Web APIs with an S-T approach and the importance of applying this perspective on software engineering, we present an S-T characterization based on categories of analysis of EAWs. It is a conceptual deepening of Web API ecosystems, as the categories Function and Tool Authoring have been identified and described for the infrastructure dimension, Actors for the Social dimension, and Policy and Monetization for the Rules dimension. In terms of the context to which an EAW relates, three categories have been distinguished based on the objectives of the EAW, namely, to establish communication with existing or potential stakeholders, to improve or extend infrastructure, and to support standards. This tool facilitates the understanding of these innovations in the software development and software engineering industry. Without abandoning the S-T and multidisciplinary approach, and in direct observation of the Web API industry, we will continue the study and analysis of EAW with a stronger focus on Software Ecosystems.

FINANCING

The authors did not receive any sponsorship to carry out this study-article.

CONFLICT OF INTEREST

The authors declare that they have no conflicts of interest related to the development of the study.

AUTHORSHIP CONTRIBUTION

Conceptualization; Data Curation; Formal Analysis; Research; Methodology; Visualization; Writing - original draft; Writing - revision and editing: Casas, S., Constanzo, M., Vidal, G. and Cruz, D.

REFERENCES

Apidays. (2022). *Platformable API Landscape State of the Market 2022*. The API Landscape. https://apilandscape.apiscene.io/

Baxter, G., & Sommerville, I. (2011). Socio-technical systems: From design methods to systems engineering. *Interacting with Computers*, *23*(1), 4-17. https://doi.org/10.1016/j.intcom.2010.07.003

Beaulieu, N., Dascalu, S. M., & Hand, E. (2022). *API-First Design: A Survey of the State of Academia and Industry* (pp. 73-79). https://doi.org/10.1007/978-3-030-97652-1_10

Boucharas, V., Jansen, S., & Brinkkemper, S. (2009). Formalizing software ecosystem modeling. *Proceedings of the 1st international workshop on Open component ecosystems*, 41-50. https://doi.org/10.1145/1595800.1595807

Campbell, P. R. J., & Ahmed, F. (2010). A three-dimensional view of software ecosyastems. *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume*, 81-84. https://doi.org/10.1145/1842752.1842774

Casas, S., Constanzo, M., Vidal, G., & Cruz, D. (2023). Ecosistemas de APIs web como un sistema socio-técnico: Un caso de estudio. *Ciencia Latina Revista Científica Multidisciplinar*, *7*(1), 9095-9120. https://doi.org/10.37811/cl_rcm.v7i1.5039

Casas, S., Cruz, D., Vidal, G., & Constanzo, M. (2021). Uses and applications of the OpenAPI/Swagger specification: a systematic mapping of the literature. *2021 40th International Conference of the Chilean Computer Science Society (SCCC)*, 1-8. https://doi.org/10.1109/SCCC54552.2021.9650408

Creswell, J. W. (2014). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches* (1.ª ed.). SAGE.

Cummaudo, A., Vasa, R., & Grundy, J. (2019). What should I document? A preliminary systematic mapping study into API documentation knowledge. *2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 1-6. https://doi.org/10.1109/ESEM.2019.8870148

De, B. (2017). *API Management: An Architect's Guide to Developing and Managing APIs for Your Organization* (1.ª ed.). Apress.

Dremel, C., Herterich, M. M., Wulf, J., & vom Brocke, J. (2020). Actualizing big data analytics affordances: A revelatory case study. *Information & Management*, *57*(1), 103121. https://doi.org/10.1016/j.im.2018.10.007

Elatlassi, R., Narwankar, C., & Calvo-Amodio, J. (2022). A Categorization of Socio-Technical Systems Approaches based on Context and Purpose. *International Society for the Systems Sciences*, *16*(1), 1-23. https://journals.isss.org/index.php/proceedings60th/article/view/2982

Fabbri, M. S. (2020). *Las técnicas de investigación: la observación* (pp. 1-10). https://www.academia.edu/36157300/Las_técnicas_de_investigación_la_observación

Fielding, R. T., & Taylor, R. N. (2000). *Architectural styles and the design of network-based software architectures* [University of California, Irvine]. https://dl.acm.org/doi/10.5555/932295#cited-by-sec

Jansen, S., Handoyo, E., & Alves, C. (2015). Scientists' Needs in Modelling Software Ecosystems. *Proceedings of the 2015 European Conference on Software Architecture Workshops*, 1-6. https://doi.org/10.1145/2797433.2797479

Kapoor, K., Ziaee Bigdeli, A., Dwivedi, Y. K., Schroeder, A., Beltagui, A., & Baines, T. (2021). A socio-technical view of platform ecosystems: Systematic review and research agenda. *Journal of Business*

Research, *128*, 94-108. https://doi.org/10.1016/j.jbusres.2021.01.060

Kingdon, J. W. (1995). *Agendas, Alternatives, and Public Policies* (2.ª ed.). HarperCollins College Publishers.

Koci, R., Franch, X., Jovanovic, P., & Abello, A. (2019). Classification of Changes in API Evolution. *2019 IEEE 23rd International Enterprise Distributed Object Computing Conference (EDOC)*, 243-249. https://doi.org/10.1109/EDOC.2019.00037

Lima, T., Pereira dos Santos, R., & Werner, C. (2015). A survey on socio-technical resources for software ecosystems. *Proceedings of the 7th International Conference on Management of computational and collective intElligence in Digital EcoSystems*, 72-79. https://doi.org/10.1145/2857218.2857230

Manikas, K. (2016). Revisiting software ecosystems Research: A longitudinal literature study. *Journal of Systems and Software*, *117*, 84-103. https://doi.org/10.1016/j.jss.2016.02.003

Manikas, K., & Hansen, K. M. (2013). Software ecosystems – A systematic literature review. *Journal of Systems and Software*, *86*(5), 1294-1306. https://doi.org/10.1016/j.jss.2012.12.026

Mathijssen, M., Overeem, M., & Jansen, S. (2020). Identification of Practices and Capabilities in API Management: A Systematic Literature Review. *Software Engineering*, *1*. https://arxiv.org/abs/2006.10481

McGregor, J. D. (2012). Ecosystem modeling and analysis. *Proceedings of the 16th International Software Product Line Conference - Volume 2*, 268-268. https://doi.org/10.1145/2364412.2364463

Messerschmitt, D. G., & Szyperski, C. (2003). *Software Ecosystem: Understanding an Indispensable Technology and Industry*. The MIT Press. https://doi.org/10.7551/mitpress/6323.001.0001

Mosqueira-Rey, E., Alonso-Ríos, D., Moret-Bonillo, V., Fernández-Varela, I., & Álvarez-Estévez, D. (2018). A systematic approach to API usability: Taxonomy-derived criteria and a case study. *Information and Software Technology*, *97*, 46-63. https://doi.org/10.1016/j.infsof.2017.12.010

Nybom, K., Ashraf, A., & Porres, I. (2017). A Systematic Mapping Study on Tools forAPI Documentation Generation. *Turku Centre for Computer Science*, *1180*, 1-22. https://doi.org/10.13140/RG.2.2.31115.49444

Ofoeda, J., Boateng, R., & Effah, J. (2019). Application Programming Interface (API) Research. *International Journal of Enterprise Information Systems*, *15*(3), 76-95. https://doi.org/10.4018/IJEIS.2019070105

Pereira dos Santos, R., & Lima Werner, C. M. (2012). ReuseECOS: An Approach to Support Global Software Development through Software Ecosystems. *2012 IEEE Seventh International Conference on Global Software Engineering Workshops*, 60-65. https://doi.org/10.1109/ICGSEW.2012.16

Pereira dos Santos, R., & Werner, C. (2011). Treating business dimension in software ecosystems. *Proceedings of the International Conference on Management of Emergent Digital EcoSystems*, 197-201. https://doi.org/10.1145/2077489.2077526

Raatikainen, M., Kettunen, E., Salonen, A., Komssi, M., Mikkonen, T., & Lehtonen, T. (2021). State of the Practice in Application Programming Interfaces (APIs): A Case Study. En *Software Architecture* (1.ª ed., pp. 191-206). Springer Nature. https://doi.org/10.1007/978-3-030-86044-8_14

Runeson, P., Höst, M., Rainer, A., & Regnell, B. (2012). *Case Study Research in Software Engineering*. Wiley. https://doi.org/10.1002/9781118181034

Santos, R., Werner, C., Barbosa, O., & Alves, C. (2012). Software Ecosystems: Trends and Impacts on Software Engineering. *2012 26th Brazilian Symposium on Software Engineering*, 206-210. https://doi.org/10.1109/SBES.2012.24

Tan, W., Fan, Y., Ghoneim, A., Hossain, M. A., & Dustdar, S. (2016). From the Service-Oriented Architecture to the Web API Economy. *IEEE Internet Computing*, *20*(4), 64-68. https://doi.org/10.1109/MIC.2016.74

APPENDIX

Annex 1. Infraestructure dimension

Tool Service Resource	RapidAPI	APILayer	Nubentos	Apilanding
Tools to support the API lifecycle.	RapidAPI Client RapidAPI for Teams	API provider Postman – OpenApi – Cloud hosting (various options)	Dashboard. API Catalog. User access and pricing plan configuration.	API-File API-DB
Tools to consume the APIs	RapidAPI Hub RapidAPI for Teams RapidAPI Testing	Live Demo (prueba de APIs) y la función Run Code Catalog	10 SDK (integración de APIs) - Catalog	API-Center API-DOC API-Testing API-Cart
Technical Support	Report Issues Form	Klaviyo	Chat with an expert – Help Center	Ticket Generator
Documentation Resources	Guides, tutorials, blogs, Get Started, videos, webiners, ebooks, data sheets, reports and white papers.	Description and examples of each API.	Blog – guides – ebooks – videos – demo reservation.	Documentation, example source codes (for more than 10 languages), a complete tutorial and technical, commercial, descriptive and success case documentation for each API
Monitoring	RapidAPI Testing (global monitoring) RapidAPI Hub: API performance (popularity, average latency, service level) API performance; APIs in use, request limit, by month/day, remaining requests month/day	API performance; APIs in use, request limit, per month/day, remaining requests month/day Datadog (monitoring service for servers, databases, tools and cloud services)	API performance, consumption level per user, usage preferences. Platform performance, % of Gateway, Marketplace and Publisher activity (30 days)	Gateway performance: number of calls received periodically within the API-Cloud, display of all consumption, consumption by type of API and activity dates and indicators (consumption and available credits)
Other	RapidAPI Enterprise Hub	Hubspot (CRM) Stripe (pagos) ReferalCandy		

Annex 2. Norms dimension

Policies	RapidAPI	APILayer	Nubentos	Apilanding
Confidentiality/Data Privacy	X	X	X	X
Data Security				X
Terms and Conditions of Use (Services/Platform)	X	X	X	X
Copyright	X			
Cookie Policy	X	X	X	X
Pricing Policy			X	

Annex 3. Norms dimension. Monetization Model (consumers)

Consumers	RapidAPI	APILayer	Nubentos	Apilanding
Product / services	Products (RapidAPI Hub, RapidAPI Client, etc.).	Platform (technical support, types of use (commercial/personal), email support, and other functionalities) APIs (Number of daily and monthly requests, number of endpoints)	Plataform API consumption	Fixed: Admin Panel, Cloud Support, High Availability, Variable: number of daily calls, number of APIs included registration with a number of credits

Structured plans	Free plans, freemium or payments (with fees and limits). Monthly subscription or pay-as-you-go plans	Plans (free, starter, pro, enterprise y custom)	User Plan + Consumer plan (Free, Freemium, Paid)	Plans (Bronze, Silver, Gold).
Flexible plans	Private plans available by invitation only	Custom plans		

Anexo 4. Norms dimension. Monetization Model (suppliers)

Suppliers	RapidAPI	APILayer	Nubentos	Apilanding
Free	Publication of APIs	Publication of APIs	Publication of APIs	Publication of APIs
Charge	Transaction fee on payments to be made to the provider.	Cloud hosting.		Sales commission on what is contracted by API consumers.

Anexo 5. Context

Objetive	RapidAPI	APILayer	Nubentos	Apilanding
Establish Communication	Facebook, Twitter Linkedin, YouTube	Facebook, Twitter Linkedin, Instagrams - Referal Candy - Hotjar - Klaviyo	Facebook, Twitter Linkedin. YouTube	Facebook, Twitter y Linkedin
Extension/Improvement of Infrastructure	GitHub, PagerDuty, Slack y Twilio.	Postman, GitHub, Cloud services (5 companies), Docker, Google Analytics, Datadog		
Support for Standards	DocuSign (contracts)	Hubspot (CRM), Stripe (payments)		