ÅTAGANDE

Upphandling	Datum
TrAF-Cloud Technical Report on Platform Ecosystem	2021-01-25

Upphandlingens referensnr	Informationsklass	Anbudets bilaga nr

TrAF-Cloud

Technical Report on Platform Ecosystem

February 2021

Amir Zadeh, Reza Esmaeili & Salah Hashemi

1 Platform definition

The notion of platform has emerged as a concrete, yet ambiguous focal point in the discourse on digitalization (Gillespie, 2010). Concrete in the sense that various aspects of functionality and characteristics are described in literature, while the very nature of platforms remains ambiguous floating between being a piece of software, an organization and a multi sided market (Constantinides, Henfridsson, & Parker, 2018; de Reuver, Sørensen, & Basole, 2018) (Sutherland & Jarrahi, 2018). Platform theory is based on the network economic theory, which maintains that a platform is the fundamental technology or service that spans more than single firm and is subject to network effects (Cusumano, 2011), and hence the value of the network is dependent upon the number of users participating in the network 1973 (Artle & Averous, 1973); (Rohlfs, 1974); (Shapiro & Varian, 1999). The network effect takes place as users take part in the network due to other users on the network or installed base (Shapiro & Varian, 1999). As a result of which several users and markets are served by few platforms (Cusumano, 2011; Eisenmann, 2006). This can lead markets to tip to a single platform or an oligopoly of dominant networks (Eisenmann, 2006; Shapiro & Varian, 1999), which can lead the platforms to provide a set of components and rules that mediate the transactions among users (Boudreau & Hagiu, 2009) on their own terms. Consequently, the third-party actors are subject to power relations within the software platform.

2 Platform types

In academia, the concept of platform has been mattered of discussion and investigation within different fields such as product development, industrial economy or technological strategy. Researchers in different fields explored and develop this concepts overtime from different angles.

As an example, “Platform” in product development, refers to families of products intended to meet the demands of core customers and at the same time easy to modify by changing, adding or removing some components (Wheelwright and Clark, 1992). Platform thinking, platform technologies and platform planning are the other derivatives from this concept.

Despite, product development researchers, Technology strategists focus on controlling aspect of the platforms. According to the strategists, platforms can be seen as control gates within industries. Given the various strategies, different approaches in leveraging platforms have been seen among the firms. However, to achieve larger market share and leadership in the competitive market is one of the main reasons that firms approach platform thinking. That’s why Baldwin and Woodard (2008) argued that in industry level, platforms play significant roles in failure or success of the company (Baldwin & Woodard, 2008).

The third aspect of platforms that has been studied and explored drew industrial economists’ attention where platforms are defined as two-sided (or multi-sided) markets (Rochet & Tirole, 2003).

Acknowledging similarities such as common base in engineering design between different definition on platforms, Gawer (2009) has categorized platforms according to their types to illustrate characteristics of each and facilitate understanding this concept (Gawer & innovation, 2009). According to Gawer (2009) three types of platform can be discussed; Internal platforms, supply chain

platforms and industry platforms (Gawer & innovation, 2009). Internal platforms are mainly leveraged in product development filed. In this field, platforms accelerate and facilitate production of a family of products with a set of common parts and modules. The second type is supply chain platform where the product can be seen as the output of a supply chain rather than a firm. A number of partners within a supply chain collaborate with each other and add one part, module or component to develop a product. In contrast with internal platforms, the third type of platforms are introduced as external platforms or industry platforms. This type of platform represents the "products, services or technologies that are developed by one or several firms, and that serve as foundations upon which other firms can build complementary products, services or technologies" (Gawer 2009, p.54).

3 Digital Platform

Digital platforms mediate activities between buyers and sellers (Transaction platform) and/or provide techniques, technologies, and interfaces to third parties to help them build their products and services (Innovation Platform) (Baldwin & Woodard, 2008) Kenney & Zysman 2016). In a sense, the notion of 'digital platform' points to a constellation of digital arrangements (infrastructures, data and algorithms) that serves to arrange and organize social and economic activity. Thus, the platform, and platform thinking, plays a crucial role in the transformation where incumbent firms search to replace their older systems with more flexible and innovative alternatives (Byggstad & Hanseth, 2018). Tiwana, et al. (2010) define digital platform as "a software-based platform as the extensible codebase of a software-based system that provides core functionality shared by the modules that interoperate with it and the interfaces through which they interoperate (e.g., Apple's iOS and Mozilla's Firefox browser)" where

module is “an add-on software subsystem that connects to the platform to add functionality to it (e.g., iPhone apps and Firefox extensions)”. Platforms are seen as a significant mode of organizing distributed innovation processes among heterogeneous actors.

Digital platforms such as Facebook, Google Play, iOS etc. stimulate Internet users to participate through various software applications that support multimedia such as music, videos, and games. Many of these software applications are third party applications, that is, these are written, designed and owned by third party developers. Thus, platforms serve as passage points for third party developers which, trigger platform owners to increasingly recognize the importance of supporting them to build and maintain platforms (Evans, Hagiu, & Schmalensee, 2008; Messerschmitt & Szyperski, 2005).

4 Platform Ecosystem

A platform based ecosystem includes two main building blocks: a platform and complementary applications (Tiwana, 2013). Tiwana’s define the core elements of a platform ecosystem as such (Tiwana 2013, p.13):

- Platform owners
- Third-party developers (complementary partners)
- Apps (Complementary services/products)
- Users
- Interfaces

By emergence of platform ecosystems, relocates the centre of innovation from the firm to a distributed network outside the firm. Instead of being the only responsible of innovation, the firm’s role also changed to facilitate external innovation. The platform owner in one hand should stimulate innovation through

engaging third-party development and on the other hand should sustain existing governance and control mechanism over the platform ecosystem.
Addressing this dilemma in orchestrating platform ecosystem, Literature suggests a couple of concepts: Trading zone and Boundary Objects.

5 Ecosystem Maturity Model

Jansen (2020) propose a model to investigate ecosystem maturity model. The model suggests detailed instruction on how to evaluate a firm's maturity in terms of shaping and governing its ecosystem. Jensen (2020) also provides 8 different levels of ambition maturity that companies may see themselves in. Table (xx) shows these levels:

Table 1. Ambition Maturity Level provided by Jansen (2020)

Level	Name	Description	Examples
0	No ecosystem	Products are budding open, first lists of partners are being created, but the ecosystems are unstructured and the coordinating organization is immature.	NetCompP3, NetCompP1
1	Extensible open product	Products are opened up for multi-layer extension, but very little attention is paid to ecosystem coordination. Partner management receives little attention.	NetCompP2, ERPCompP1
2	Extensible open platform	The product is increasingly seen as a platform. Third parties approach the organization with feature requests about the platform.	XBMC platform
3	Robust platform ecosystem	Partners increasingly base their business on the platform. The platform is leading in some niches. The supporting organization can fully support all partners. Some certification takes place.	Eclipse platform
4	Leading ecosystem	Partners are benefiting greatly from the platform. Customers are increasingly seeing the value of the platform and creating extensions themselves. The platform is challenging the status quo in some industries.	SAP Hana Platform
5	Reigning ecosystem	The ecosystem is at full strength and growing rapidly. Partners are experiencing strong connections with the coordinating firm. The coordinating party is strategically focusing on the platform and decreasing its efforts on customers and end-users. It is on top in many industries and seen as a market leader. Others say they want to have an ecosystem such as the coordinating party.	Steam platform
6	Absorbing ecosystem	The ecosystem is leading and also absorbing other ecosystems, such as surrounding hardware and software ecosystems. Patents and mergers have become strategic instruments for increasing business. New niches are introduced regularly.	Apple iOS Platform
7	Ecosystem of ecosystems	The ecosystem is absorbing other ecosystems and creating new ones in its wake. Third parties can create markets in markets. The coordinating firm needs to maintain an open strategy for fear of monopoly.	Google Android Platform, RedHat Linux platform

The ecosystem maturity model provides 7 focus areas. These focus areas should be investigated to define the maturity of a firm's ecosystem. Figure 1 shows these 7 focus areas:

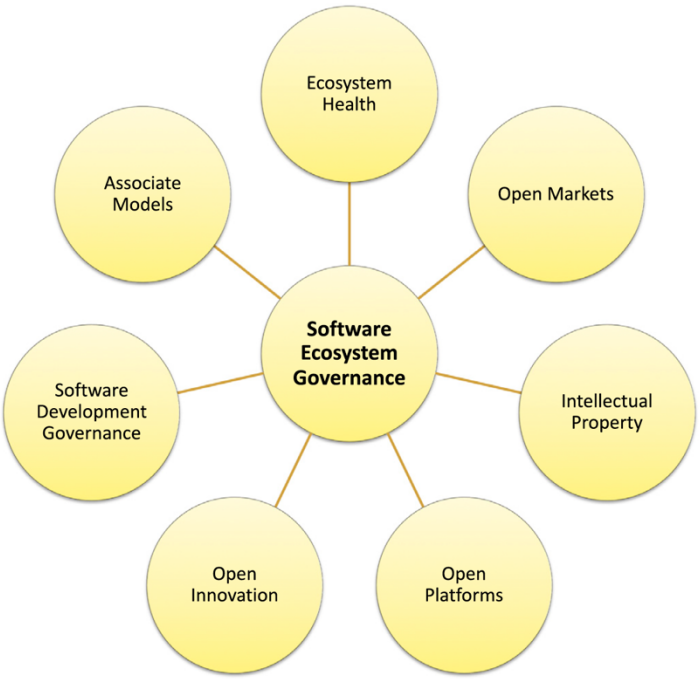


Figure 1. Focus areas of ecosystem maturity model

Each one of these 7 focus areas includes several subcategories. As an example, while investigating the aspects of open platforms in an ecosystem, the firm should study it from 1) platform hardening, 2) platform extensibility, 3) software operation knowledge, 4) Platform documentation, 5) security and 6) platform evolution.

Sub-categories of the seven focus areas are provided in the figure 2:

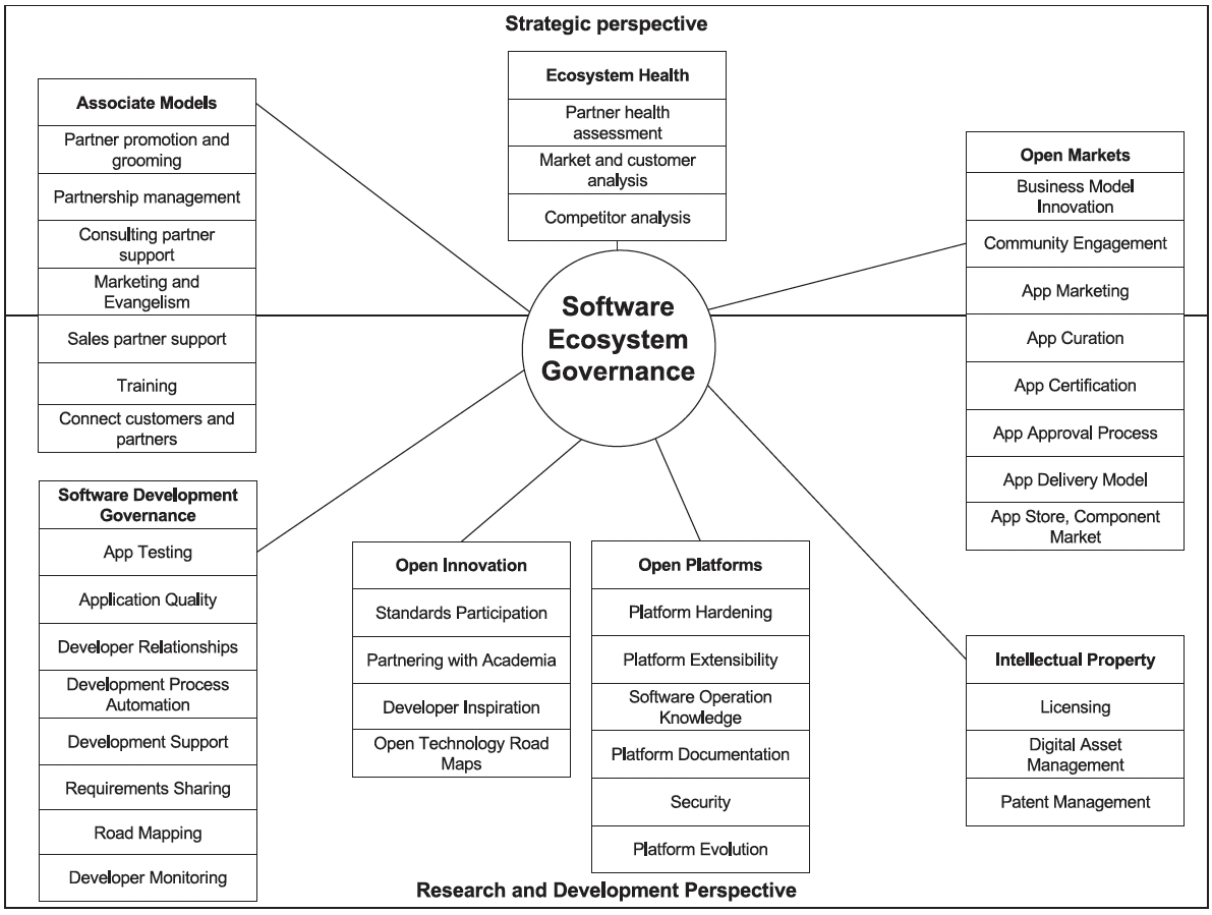


Figure 2 Sub-categories of the focus areas

Ambition level (provided in the table 1) guides the firm to understand the status of the ecosystem in any of these sub-categories of the focus areas.

As an example, in open platform focus areas, when it comes to platform documentation, the firm should study the current status. If there is evidence for documentation with getting started, then it is the first level of ambition. If there is documentation with examples, then the ambition level is 2. The ambition level will be 3 if there is an evidence to show that documentation is generated from codes. Level 4 is interactive documentation and level 5 is when feedbacks are gathered. The ultimate level of maturity in platform documentation is level 6 when

the prioritizing is done based on knowledge needs. The article defines each sub-categories and relevant ambition level.

There are several interesting points about this model that can be applied in ecosystem shaping and governance. The first point is that there is a model to study maturity of an ecosystem. If a company thinks that there is an ecosystem around its platform, this model can help to find out how mature it is. If there is no ecosystem around the platform, then this model is a guideline to consider different aspects to shape and grow an ecosystem.

6 Ecosystem and Platform Boundary Resources

6.1 Platform Boundary Resources

The boundary resources model draws on the notion of boundary objects (Star & Griesemer, 1989) as applied to software platforms and their role in stimulating third party development as an alternative form of system development. Boundary objects are ‘common objects (that) form the boundaries between groups through flexibility and shared structure’ (Star, 2010; p.603), and potentially engage in power relations (Boland Jr & Tenkasi, 1995). The flexibility embedded in the boundary object allows enough room for affordances of various interpretations and usage in different contexts by diverse actors (Hutchby, 2001). The boundary objects in the context of BRM are labelled as boundary resources, which are software tools and regulations designed by platform owners or in-house developers to facilitate the development of new contexts, defined by third-party applications. The actors using boundary resources to develop third party applications are called third party developers.

Boundary resources facilitate the interaction between platform and third-party applications and are “plastic enough to adapt to local needs and constraints of the several parties employing them, yet robust enough to maintain a common identity across sites” (Star and Griesemer, 1989, p. 393). As such, they are designed in a way that although construed in a particular context they are useful in various contexts that may not exist at the time of its conception. Such boundary resources are designed so as to facilitate the development of new contexts defined by innovative third-party applications. Examples of boundary resources include Application Programming Interfaces (APIs), documentation, and legal documents. The role of third party developers is an inevitable one when it comes to platform success and survival (Taudes, Feurstein, & Mild, 2000). Third party developers contribute to designing the platform ecosystem by entering into contractual agreements offered by the platform. These agreements are considered as boundary resources.

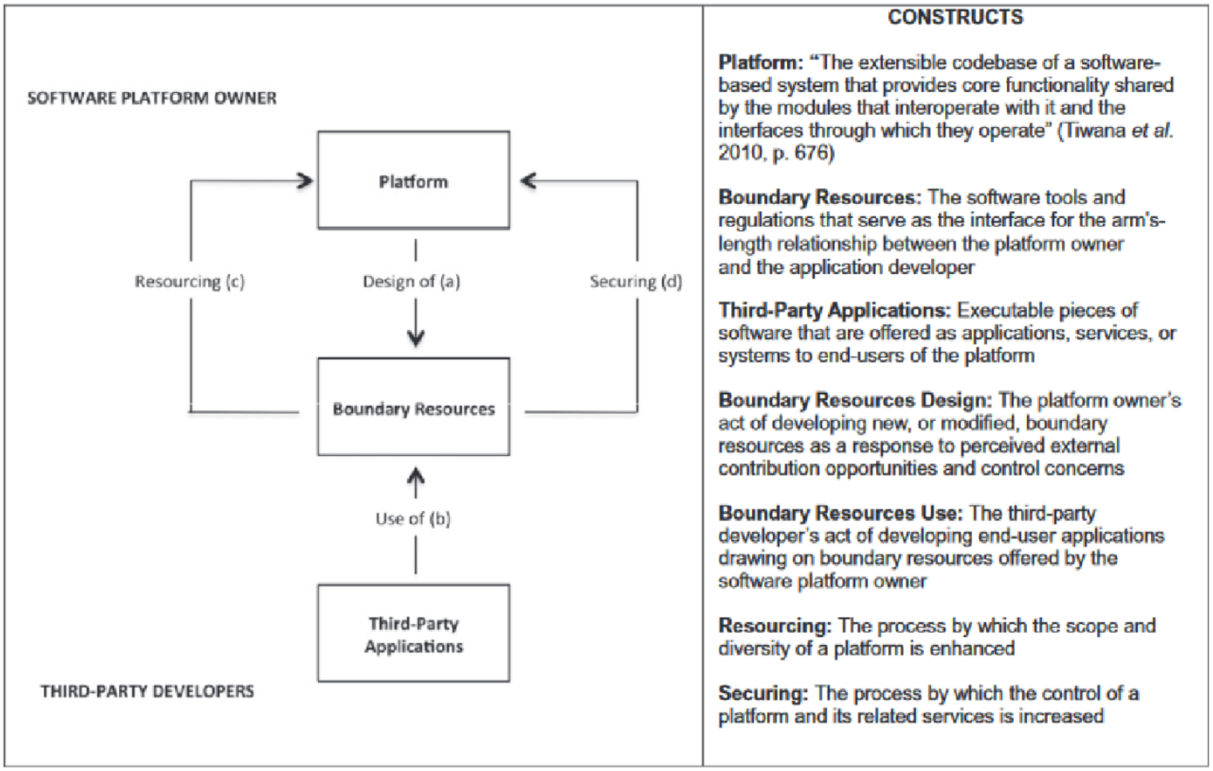
6.2 Platform Boundary Resources in Ecosystem Maturity Model

Ghazawneh (2012) defines two major types of platform boundary resources: technical and social. Technical boundary resources are tools such as APIs and SDKs and social Boundary resources are guidelines, documentation and other incentives to guid or motivate the third-party developers. In Ecosystem Maturity Model, Technical boundary resources play significant role in approaching open platform. In other words, if the incumbent aim for the open platform, then one of the aspects that they should consider is to design the platform extensible. To do so, platform boundary resources should be provided to the third parties.

While the ecosystem maturity model describes the importance of PBRs in extensibility of the open platform, it does not provide further information on what PBRs should be design or how they should be designed. The authors of this this report suggest further studies to define the right approach for designing PBRs using other frameworks and tools.

7 Designing Platform Boundary Resources

A boundary resource model has been developed by Ghazawneh and Henfridsson (2013) to illustrate the arm's-length relationships between platform owners and third-party developers through the boundary resources (Ghazawneh & Henfridsson, 2013). In this model, third-party developers can contribute to the platform ecosystem by developing "executable pieces of software that are offered as applications, services, or systems" (Ghazawneh and Henfridsson, 2013, p. 174). To do so, the developers use the boundary resources designed by the platform owner. This model also includes boundary resource design. When it comes to designing boundary resources, Ghazawneh and Henfridsson's (2013) model answers three questions: 1. Who designs the boundary resources? 2. What approaches can be taken? 3. What is the purpose of developing boundary resources?



Regarding the first question, the platform owner designs and develops the boundary resources (Ghazawneh & Henfridsson, 2013). Depending on the developing purpose, the platform owner can initiate the process by designing completely new resources or by modifying existing resources. They argued that "boundary resources design is typically initiated when a platform owner recognizes that existing boundary resources are insufficient for developing the platform, including its applications, in a favourable way." According to Ghazawneh and Henfridsson (2013), modification or re-design of boundary resources is typically an answer to control concerns like "when third-party developers launch, or announce the intention to launch, applications that represent potential threats to the platform." The platform owner designs new boundary resources to facilitate third-party development and the possibility to develop new applications.

They also argued that sometimes, the boundary resources are not what third-party developers need or want. In this case, the developers can create new boundary resources. This type of contribution is defined as "self-resourcing" by Ghazawneh and Henfridsson (2013). However, even in the case of self-resourcing, the third party's role is simple dialectic negotiations with the owner of the platform. In other words, the primary designer of the boundary resource is the platform owner (Ghazawneh & Henfridsson, 2013).

8 Practical Contribution

In this technical report we are trying to highlight the contribution of this report to industrial practices. This chapter of the report will be focused on:

1) Applicable lessons from this report for the TrAF-Cloud project and

2) Combiteh's cybersecurity experts' review on the ecosystem maturity model.

8.1 Lessons for the TrAF-Cloud Project

There are several key messages in this report that can contribute to the TrAF Cloud Project.

The first message is that designing and developing digital platform does not necessary lead into shaping powerful ecosystem. Shaping, empowering and governing an ecosystem requires a coherent roadmap. This report tries to introduce one of the (if not the only) existing roadmap for shaping an ecosystem.

The second message is that even if there is an ecosystem, a well-defined framework is needed to assess and evaluate the maturity level of the ecosystem. This framework should describe different ambition level and criteria for

evaluating the ecosystem status in each focus area. In that sense the suggested ecosystem maturity model is trying to guide us in this report.

This report also tries to illustrate the difference between two main important concepts in this project: Ecosystem and Platform. This report tries to explain these concepts, categories of them and show the differences between them. This will help the readers to understand the relation and correlation between these two main building blocks.

Finally, as an important bridge between the concept of platform and ecosystem, this report reviews the concept of platform boundary resources (PBRs). Platform owners may use the PBRs to govern the ecosystem. Providing right and useful tools, APIs, SDKs, guidelines and documentation can encourage and empower third-party developers. In this way PBRs play significant role in nurturing the ecosystem. Beside this, PBRs has been seen as the main governance mechanism that platform owner has to govern the ecosystem. The PBRs design model in this report tries to explain this dual function.

8.2 Cybersecurity aspect of the Ecosystem Maturity Model

Cybersecurity is an inseparable part of every system working with product development. The need for security becomes even more critical when it comes to software products. Thus, Software Ecosystem Governance is no exception to the rule since it encircles a broader range of activities, and software development governance is only one of them.

Software Ecosystem Governance Maturity Model (SEG-M^2) practices are classified into seven different categories each of which encompassing both

technical and managerial aspects to some extent. The domains that these focus areas are covering are not of the same size. In other words, the requirements under each category may differ in type and size depending on what the category encloses. Although security is briefly mentioned under few of these focus areas such as software development governance or open platforms, yet there are multiple security checks and controls that shall be considered in almost all categories. It should also be noted that the existing security practices in SEG-M^2 focus more on how security-related practices shall be governed rather than what security practices are actually required in an ecosystem surrounding software products.

Table below represents a summary of required security components under each focus area.

Focus area	**Security components**

Associate Models	• Security governance
	• Security policies, procedures and guidelines
	  ○ Privacy protection
	  ○ Asset classification
	  ○ …
	• Risk management
	• Security training and awareness
	• Security incident response

Software Development Governance	• Secure Software Development Life Cycle (SSDLC)
	  ○ Security requirement management
	  ○ Secure design
	  ○ Secure coding
	  ○ Security test
	  ▪ SAST
	  ▪ DAST
	  ▪ Pen-testing
	  ▪ Fuzz testing
	  ○ Secure implementation
	  ▪ Secure configuration
	  ▪ Secure installation
	  ○ Secure maintenance
	  ▪ Secure update

Open Markets	• Version handling • Code signing • Secure update/ patch management • Over-the-air update • API security management
Intellectual Property	• Ecosystem IP protection • Forensic investigation • Non-Disclosure Agreement (NDA)
Open Platforms	• Platform hardening, server hardening, DB hardening, OS hardening, network and network hardening, etc. • Vulnerability management • SOC and monitoring
Ecosystem Health	• Security health assessment of the partners • Secure communication and data transfer
Open Innovation	• Secure communication and data transfer • Non-Disclosure Agreement (NDA)

Associate models

As this area focuses on coordination and managing business partners as well as defining partnership models, the security requirements in this category are mainly from a higher-level perspective. In other words, the main security governance plan shall be defined and then maintained in this area. The management level shall define proper security policies, procedures and guidelines covering all security needs of any partnership in the ecosystem. These policies and guidelines shall address topics such as preservation of privacy in all communications within the ecosystem, identification of valuable assets that shall be protected, and the like.

A helpful tool the management team can utilize in order to establish well-defined policies and procedures is Risk Assessment. Performing a high-level risk assessment will assist them with the identification of potential threats and risks against the ecosystem and help them define clear security goals to be followed. Furthermore, the security guidelines shall include a plan for organizing a security incident response procedure, so that in case of a security incident, each partner knows how to involve in order to handle the incident.

Eventually, security training and awareness should be included into the policies and procedures such that each party receives necessary information and training about different threats against the ecosystem and how such threats can be prevented, detected and recovered from.

Software development governance

Software development has immensely evolved in the recent past. Prior to the modern models used nowadays, security as an important part of software development was either ignored or postponed to the end of the development lifecycle when there was limited to no time and budget. The emergence of Continuous Integration and Continuous Delivery (CI/CD) made it possible for

security to be an integrated part of the software development lifecycle and granted it a new name: Secure Software Development Life Cycle (SSDLC).

This new method consider security from the starting steps of the development, namely requirement engineering and design all the way to implementation and nowadays even in the operation and maintenance. In the requirement elicitation phase, in addition to the functional requirement, the security requirements, which are mostly derived from the security goals defined by the management team, shall also be considered. In the design phase, utilization of security design principles such as defence in depth, fail-safe, etc. shall be encouraged throughout the design phase. Once the software product is designed, in the coding and implementation phases multiple types of testing shall be performed on the product to ensure that it is clear of all known vulnerabilities as well as that the product meets the pre-defined expectations. Various types of testing such as Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST), penetration testing, and fuzz testing shall be performed in order to assure the security of the product from an implementation perspective. Eventually, since new vulnerabilities are discovered almost on daily basis, once the software is deployed in its operational environment, regular patches and updates need to be applied on the software in a secure manner.

Open Markets

SEG-M^2 discusses the possibility of creating an open market for services and applications in the ecosystem. Although this openness makes delivery of services, applications and extensions easier, it also raises the need for secure boundaries. Issues like handling consecutive versions of a service, managing of updates in a secure manner and the integrity of the products or services by means of signing them shall all be addressed in this category.

The most important issue for which security requirements shall be defined is the security of the APIs existing in the ecosystem. A well-defined security procedure for how different partners collaborate and deliver their services using such APIs shall be followed by every partner.

Intellectual Property (IP)

As mentioned before, the assets or properties of the ecosystem requiring protection shall be identified by the higher levels in the ecosystem. The security practices under this category shall ensure the protection of all intangible properties such as data, patents and ideas of the ecosystem. In addition, well-defined documents and contracts such as None Disclosure Agreements (NDA) shall be used as a part of IP protection.

Once all the preventive mechanisms are in place regarding the protection of the intellectual properties, one or more units shall be organized for any probable needs for future forensic investigation.

Open platforms

Creation of a solid, stable open platform makes it easier for future service providers to join the ecosystem and be able to register their products using standardized procedures. However, the hardening of such platform is of great importance. Ecosystem hardening shall encompass all participating platforms, repositories, servers and markets existing in the ecosystem. On the other hand, managing vulnerabilities by means of iterative vulnerability assessments and constant monitoring by a Security Operation Centre (SOC) should be considered as an essential need for the ecosystem.

Ecosystem health

Since the ecosystem is an open platform working with multiple markets, service providers, developers, or even other ecosystems, the health status of all the

partners shall be constantly analysed a monitored. This includes the security health assessment of the partners themselves as well as mandating partners to provide a security health proof of their own partners. In other words, all parties involved in the ecosystem either tier one (main suppliers) or tier two (suppliers of the suppliers) and so on shall be assessed from a security point of view prior and during their collaboration with the ecosystem.

Open Innovation

The possibility to share knowledge across an ecosystem is a key to build a flourishing ecosystem. However, this openness and sharing shall be securely regulated. Security controls shall be in place to provide a secure way of communication and data transfer. In addition, similar to the preservation of intellectual properties, well-defined contracts and documentation such as NDAs shall be used as a part of procedures concerning open innovations.

9 References

Artle, R., & Averous, C. (1973). The telephone system as a public good: Static and dynamic aspects. *The Bell Journal of Economics and Management Science*, 89-100.

Baldwin, C. Y., & Woodard, C. J. (2008). The architecture of platforms: A unified view. *Harvard Business School Finance Working Paper*(09-034).

Boland Jr, R. J., & Tenkasi, R. V. (1995). Perspective making and perspective taking in communities of knowing. *Organization science, 6*(4), 350-372.

Boudreau, K. J., & Hagiu, A. (2009). Platform rules: Multi-sided platforms as regulators. *Platforms, markets and innovation, 1*, 163-191.

Constantinides, P., Henfridsson, O., & Parker, G. G. (2018). Introduction—platforms and infrastructures in the digital age. In: INFORMS.

Cusumano, M. A. (2011). Platform wars come to social media. *Communications of the ACM, 54*(4), 31-33.

de Reuver, M., Sørensen, C., & Basole, R. C. J. J. o. I. T. (2018). The digital platform: a research agenda. *33*(2), 124-135.

Eisenmann, T. R. (2006). Platform-mediated networks: definitions and core concepts.

Evans, D. S., Hagiu, A., & Schmalensee, R. (2008). *Invisible engines: how software platforms drive innovation and transform industries*: MIT Press.

Gawer, A. J. P., markets, & innovation. (2009). Platform dynamics and strategies: from products to services. *45*, 57.

Ghazawneh, A., & Henfridsson, O. (2013). Balancing platform control and external contribution in third-party development: the boundary resources model. *Information Systems Journal, 23*(2), 173-192.

Gillespie, T. (2010). The politics of 'platforms'. *New media & society, 12*(3), 347-364.

Hutchby, I. (2001). Technologies, texts and affordances. *Sociology, 35*(2), 441-456.

Messerschmitt, D. G., & Szyperski, C. (2005). Software ecosystem: understanding an indispensable technology and industry. *MIT Press Books, 1*.

Rochet, J.-C., & Tirole, J. J. J. o. t. e. e. a. (2003). Platform competition in two-sided markets. *1*(4), 990-1029.

Rohlfs, J. (1974). A theory of interdependent demand for a communications service. *The Bell Journal of Economics and Management Science*, 16-37.

Shapiro, C., & Varian, H. R. (1999). The art of standards wars. *California management review, 41*(2), 8-32.

Star, S. L. (2010). This Is not a Boundary-Object. *Revue d'anthropologie des connaissances, 4*(1), 18-35.

Star, S. L., & Griesemer, J. R. (1989). Institutional ecology,translations' and boundary objects: Amateurs and professionals in Berkeley's Museum of Vertebrate Zoology, 1907-39. *Social studies of science, 19*(3), 387-420.

Sutherland, W., & Jarrahi, M. H. (2018). The sharing economy and digital platforms: A review and research agenda. *International Journal of Information Management, 43*, 328-341.

Taudes, A., Feurstein, M., & Mild, A. (2000). Options analysis of software platform decisions: a case study. *MIS quarterly*, 227-243.

Tiwana, A. (2013). *Platform ecosystems: aligning architecture, governance, and strategy*: Newnes.