Design Options of Store-Oriented Software Ecosystems: An Investigation of Business Decisions *

Bahar Jazayeri[1], Olaf Zimmermann[2], Gregor Engels[1], Jochen Küster[3], Dennis Kundisch[1], Daniel Szopinski[1]

[1] Paderborn University, Germany
{bahar.jazayeri,gregor.engels,dennis.kundisch,daniel.szopinski}@upb.de
[2] University of Applied Sciences of Eastern Switzerland, Switzerland
ozimmerm@hsr.ch
[3] Bielefeld University of Applied Sciences, Germany
jochen.kuester@fh-bielefeld.de

Abstract. Nowadays companies like Apple create ecosystems of third-party providers and users around their software platforms. Often online stores like Apple App Store are created to directly market third-party solutions. We call such ecosystems *store-oriented software ecosystems*. While the architecture of these ecosystems is mainly derived from business decisions of their owners, ecosystems with greatly different architectural designs have been created. This diversity makes it challenging for future ecosystem providers to understand which architectural design is suitable to fulfill certain business decisions. In turn, opening a platform becomes risky while endangering intellectual property or scarifying quality of services. In this paper, we identify three main design options of store-oriented software ecosystems by classifying existing ecosystems based on similarities in their business decisions. We elaborate on the design options, discuss their main contributions, and provide exemplary ecosystems. Our work provides aspiring ecosystem providers with the reusable knowledge of existing ecosystems and helps them to take more informed architectural decisions and reduce risks in future.

Keywords: Software ecosystems, Reusable designs, Variabilities

1 Introduction

Software ecosystems have become an emerging architectural approach for many companies to grow. The term *software ecosystem* is inspired from ecological ecosystems that are the result of an interplay between organisms as well as interactions with a physical environment [1]. Comparably, a software ecosystem consists of a software platform, a set of third-party providers in service to a community of users [2]. For instance, Apple Inc. created an ecosystem of third-party developers in service to the users of mobile Apps around the iOS and MacOS platforms. In practice, many of software ecosystems include online

* This work was partially supported by the German Research Foundation (DFG) within the Collaborative Research Center "On-The-Fly Computing"(CRC 901).

stores such as Apple App Store to make third-party solutions directly available to their users. We call them *store-oriented software ecosystems*.

Recently, such ecosystems have widely been created in different application domains. For instance, while mobile App ecosystems (like Apple and Google) are among the most popular ones with millions of users, ecosystems are created around open source platforms (e.g., Eclipse and Mozilla Firefox[4]) and enterprise applications (e.g., Salesforce[5] and Cloud Foundry[6]). Furthermore, existing ecosystems use different terminologies. For example, third-party providers are known under different terms such as independent developers, committers, and partners. Third-party solutions are also called by plug-ins, Apps, add-ons, etc. In this paper, we refer to third-party providers as *extenders* and third-party solutions as *extensions*. Moreover, in support of different business decisions, ecosystems with diverse architectural designs have been created whereas different ecosystems include very different software features [3]. For instance, in commercial ecosystems like Apple and Salesforce, revenue streams are channeled to the extenders using billing features. However, billing is a negligible feature in open source software ecosystems like Eclipse and Mozilla Firefox.

This diversity makes it challenging for future ecosystem providers to understand which architectural design is suitable to fulfill certain business decisions. In turn, opening a platform becomes a risky while endangering the intellectual property or scarifying quality of products and services [4]. To tackle the diversity, some works [5–7] study variabilities of business decisions and software features in software ecosystems. However, the relation between these two is hardly known. Therefore, it is not clear how to design a store-oriented software ecosystem that fits to an ecosystem provider's needs. This hinders systematic development of customized ecosystems in the future.

In this paper, we investigate 111 existing store-oriented software ecosystems and classify them based on similarities in their business decisions. To do so, we use a variability model identified by Jazayeri et al. [6, 8] for variable design decisions of store-oriented software ecosystems. Modeling recurring architectural decisions for knowledge reuse has been proposed by Zimmermann et al. in [9]. We notice that similar business decisions result in similar software features. Our work abstracts from this knowledge and suggests an architectural landscape for each group of ecosystems. Such an architectural landscape resembles a macro view to the system and its interaction with an environment [10, p. 254]. The contribution of our work is twofold: a) Our results show that three design options of store-oriented software ecosystems are frequently applied in practice, which we call *resale software ecosystem*, *partner-based software ecosystem*, and *open source software-based ecosystem* (shortly *OSS-based software ecosystem*). We discuss each design option in detail, elaborate on its main contribution in terms of a business goal, and provide real-world examples. b) Our work provides a practical mean to aspiring ecosystem providers by sharing the knowledge on the reusable architectural designs to perform more informed decision-making while creating their own ecosystems. Specifically, each design option contributes to a main business goal. This knowledge can also be used by existing ecosystem provider to assess their own architectural designs

[4] www.mozilla.org/ [5] www.salesforce.com/ [6] www.cloudfoundry.org/

with respect to the business goals discussed in this paper. In the following, we investigate and classify the existing ecosystems in Section 2. Section 3 presents the frequently applied design options whereas Section 4 elaborates on the overall contributions of the design options. The paper is concluded in Section 5 and future directions are addressed.

2 Classifying Store-Oriented Software Ecosystems based on Variabilities of Business Decisions

We aim at identifying the main design options of store-oriented software ecosystems. We take real-world ecosystems as the basis of our investigation. Initially, we collect a list of ecosystems by first defining our search terms and then performing web searches. To capture as many ecosystems in as diverse application domains as possible, we derive the search terms by using a taxonomy for software ecosystem introduced by Bosch [1]. The author classifies open software platforms into three types, i.e., desktop, web, and mobile. Whereas, each type appears in three dimensions, i.e., operating system, application, and end-user programming. Examples of our search terms are "top operating systems", "top end-user programming software", and "top cloud computing platforms".

Afterwards, we inspect the ecosystems with respect to their business decisions based on a variability model in [6, 8]. We use the variability model, because it provides a concrete set of variabilities that we can use to compare and group the ecosystems. Decisions on such variabilities derive the main architectural landscape of an ecosystem. Table 1 shows the variabilities in form of *variation points* and *variants*. A variation point is the subject of a variability whereas a variant is the object of the variability, i.e., a concrete business decision. Accordingly, **extender** defines who extends a platform's functionality. **Openness** specifies whether a platform is open source. **Fee** defines the main costs of participating in an ecosystem. **Feedback Loop** determines software features that enable a positive feedback loop between users and extenders. In the context of markets, this happens when more users use the platform, and thereby, the number of extenders increases [11]. Finally, **Knowledge Sharing** defines software features that enable knowledge-sharing among users or extenders to communicate and generate common values [12].

We consider whether and how the ecosystems in our list realize every variation point in Table 1. Zimmermann et al. [9] propose an approach to capture and model recurring architectural design decisions for the purpose of knowledge reuse. This allows us to detect three groups of ecosystems that provide similar sets of instances for the variation points. In the next section, we discuss each group of ecosystems in detail. A complete list of search terms and the list of ecosystems can be found in our dataset [13].

3 Design Options of Store-Oriented Software Ecosystems

Three major design options of store-oriented software ecosystems emerge from our investigation in Section 2. In the following, we present the design options in order of their popularity among our list of ecosystems. We describe each design

	Variation Point	Variant	Resale Software Ecosystems	Partner-Based Software Ecosystems	OSS-Based Software Ecosystems
Determinant Design Decisions of Business Architecture	**Extender** Who extends the platform's functionality?	**Trusted Partner**	Hardware / software suppliers, strategic partners, system integrator, & resellers	Hardware /software suppliers, strategic partners, system integrator, & resellers	–
		Independent Developer	High number of independent developers	–	High number of independent developers
	Openness Is the source code open?	**Open**	Fully or partially open source libraries	Access to the source code and API Reference for partners	Source code on a public repository, e.g., GitHub or SourceForge
		Closed	Fully or partially closed source libraries	No access for independent developers	–
	Fee What are the main costs of participating in the ecosystem?	**Entrance Fee**	Same payment for everyone / One time payment / periodic payment	Membership in partner programs / different payments for different partners	–
		Platform Fee	Usually the same for everyone	Different payments for different platform editions, e.g., standard & enterprise	–
	Feedback Loop Facilitator Which software features enable a positive feedback loop between the users and extenders?	**Rating & Reviewing**	Number of downloads, likes. Star system, reviewing	–	Number of forks / sharing / likes (bookmarks, favourites)
		Ranking	List of featured / popular / new extensions	–	–
		Market Analytics	–	Statistics on user visits (No. of clicks, time spent on a webpage). Consulting services on market trends. Mail server to keep users in contact with partners	–
		Version Control Management	–	–	E.g., SVN, Git, Mercurial, Perforce, TortoiseCVS, TeamForge, etc.
		Ticket and Issue Tracking System	–	–	Test planing, bug tracking, notification interfaces, e.g., Jira, Bugzilla, Apache Bloodhound, and Roundup
	Knowledge Sharing Which software features enable knowledge sharing inside the communities of users or extenders?	**Documentation Framework**	Developer manuals, wiki	Partner portal: Documentation and guidelines on software development frameworks (requiring authorized access)	Developer manuals, wiki, public resources from open source software community
		Q&A Forums	Developer forum, user forum, idea forum	–	Public developer forum, e.g., StackOverflow

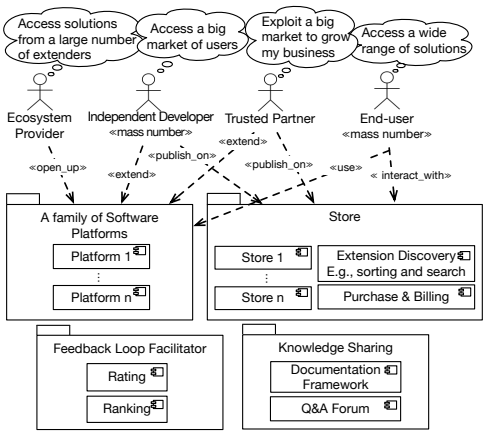
Table 1. Variabilities of business decisions and real-world instances from each group of ecosystems ("–" means that a variant is not realized in the architecture)

option by answering two questions: a) What is the architectural landscape of the design option? We abstract from the knowledge of variabilities and suggest an architectural landscape for the design option using the UML notation. As a part of it, thought bubbles are used to refer to the actors' business goals of participating in an ecosystem, e.g., "*I want to access a big market of users*". b) What are exemplary real-world ecosystems applying the design option? While naming exemplary ecosystems, we elaborate on one ecosystem in detail.

3.1 Resale Software Ecosystem

37% of ecosystems in our list provide software products and services to a mass number of end-users. In addition, a mass number of extenders develops extensions on top of the platforms. After the extensions are developed, they are sold several times. We call this group of ecosystems *resale software ecosystems*.

a) *Architectural landscape*: Table 1 provides details on how existing ecosystems realize the variants of the variability model. Figure 1(a) outlines the architectural landscape of resale software ecosystems. **Extender**: Both trusted



(a) Architectural Landscape

Variation Point	Variant	Apple
Extender	Trusted Partner	IBM, salesforce, Samsung, etc
	Independent Developer	Mass number of independent developers
Openness	Open	Few libraries like WebKit
	Closed	iOS, MacOS
Fee	Entrance Fee	Annual payment
	Platform Fee	Indirectly through purchasing devices
Feedback Loop Facilitator	Rating & Reviewing	Star rating system, Reviewing
	Ranking	Lists of featured, popular, & new Apps
Knowledge Sharing	Documentation Framework	Apple Developer Documentation
	Q&A Forums	Apple Developer Forum, Apple Support Communities

(b) Exemplary Ecosystem

Fig. 1. Resale software ecosystem: An architectural solution to business scalability

partners and independent developers extend the platforms and publish their extensions on the stores, where a mass number of end-users can access them. **Fee**: In general, resale software ecosystems can be commercial or non-commercial. A majority of 82% in our list is commercial, while the users have to pay for the extensions. There might be an entrance fee and platform fee, which varies based on an ecosystem provider's strategies. **Feedback Loop Facilitator**: In order to create a positive feedback loop between the mass number of extenders and end-users, software components like rating and ranking are employed. **Knowledge Sharing**: Q&A forums such as developer and user forums support knowledge sharing and social interaction between the end-users and developers.

Further characteristics: A majority of providers of resale software ecosystems, i.e., 84%, offers a family of software products. In support of multiple platforms, the stores are divided to sub-stores, each for extensions of one platform. Since the developers work highly independent of the ecosystem providers, another concern is to separate their revenue streams. This is achieved by including billing and purchase features on the stores.

b) *Exemplary real-world ecosystems*: Apple Inc. is the provider of a resale software ecosystem while the iOS and MacOS are the software platforms. There are App Store and Mac App Store, where third-party software and digital products are published. Figure 1(b) provides concrete instances of the variants in the Apple ecosystem. IBM, Salesforce, and Samsung are examples of Apple's trusted partners. The source code is mainly closed. Developers pay an annual entrance fee and entering the ecosystem as a user requires purchasing Apple devices. In addition, a feedback loop is created using a star system and different ranking categories like featured Apps. Apple Developer Forum[7] and Support Communities[8] are examples of Q&A forums. Further examples of the resale software ecosystems are Adobe, Salesforce, Facebook, Esri, Informatica, Autodesk, and FreshDesk.

[7] `developer.apple.com/devforums/` [8] `discussions.apple.com/`

3.2 Partner-Based Software Ecosystem

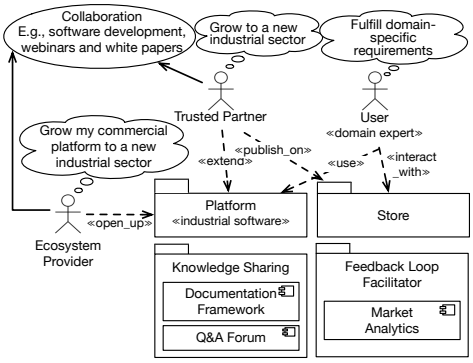
Another group of ecosystems is the ecosystems of carefully selected partners. This includes 35% of our list. We call them *partner-based software ecosystems*. Providers of these ecosystems are companies offering complex software solutions for industrial sectors. Supply chains, warehouse operation, and aerospace are examples of such industrial sectors. Moreover, users in the partner-based software ecosystems are also companies that we call *domain experts*, because they are companies with high knowledge and expertise in certain domains. These domain experts offer the solutions to further end-users.

a) *Architectural landscape*: Table 1 provides details on how the existing partner-based software ecosystems realize the variants of the variability model. Furthermore, Figure 2(a) sketches the architectural landscape of the partner-based software ecosystems. **Extender**: Trusted partners are the only extenders in the ecosystems. The ecosystem providers collaborate with the partners to develop joint solutions in new industrial sectors or for new application use cases. The collaborations come in form of software development, road map sessions, and webinars as well as marketing the joint solutions. To achieve different intensities of collaborations, the ecosystem providers often offer different tiers of partnerships like platinum or gold partners. Joint solutions resulting from a closer collaboration are eligible for certain promotions such as getting listed as featured products on the stores. **Openness**: Source code is closed in general. The platform APIs are monetized through an API management system, which requires authentication and defines a partner's access permissions to the code or APIs. **Fee**: To enter the ecosystems as a partner, candidates need to fulfill certain requirements, e.g., having certain annual revenue. Such requirements vary depending on ecosystem's policies. Prospective partners usually need to periodically pay to participate in the ecosystems or to use the platforms.

Feeback Loop Facilitator: In contrary to resale software ecosystems, rating features are not widely used in partner-based software ecosystems. But, the extensions are rather marketed in the ecosystems. Market analytics features inform the partners about quality of user experience. Examples of the market analytics are customer relationship management (CRM) features and repository mining. Thereby, the partners can stay connected with their customers and track their market growth. **Knowledge Sharing**: Documentation frameworks are provided as a part of partner programs. Partners are given access to partner portals. Due to protection of intellectual property, such portals are not accessible to anyone and require access permission. However, Q&A forums are accessible to the current and prospective users and partners.

Further characteristics: The extensions on the stores are often labeled as "tested" or "validated". The partner-based ecosystems are highly commercial whereas 96% of the ecosystems in our list are for profit.

b) *Exemplary real-world ecosystems*: Citrix is a provider of cloud computing services. Among others, the services include server and desktop virtualization and cloud computing. Figure 2(b) provides the variants of the Citrix ecosystem. Citrix establishes long-term collaborations with strategic partners like Microsoft, Cisco, and Google. The Citrix platforms are closed source. Platform fee is defined as a pay-as-you-go model that allows the partners to pay on-demand



(a) Architectural Landscape

Variation Point	Variant	Citrix
Extender	Trusted Partner	Citrix strategic partners: Microsoft, Cisco, Google, etc. Citrix partner programs: Service providers, System Integrators, SaaS Referral Partner, etc.
Openness	Open	Partially only to the partners
	Closed	Platforms are generally closed
Fee	Entrance Fee	Periodic payment depending on partner type
	Platform Fee	On-demand pay-as-you-go
Feedback Loop Facilitator	Market Analytics	Citrix Marketing Concierge
Knowledge Sharing	Documentation Framework	Citrix Partner Central
	Q&A Forums	Citrix Discussions, Citrix User Group Community (CUGC)

(b) Exemplary Ecosystem

Fig. 2. Partner-based ecosystem: An architectural solution to strategically grow a commercial platform

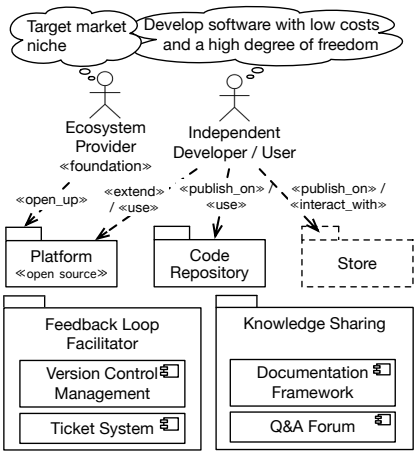
for the Citrix services. Citrix Ready Marketplace[9] is the store, where the partners' extensions are published. As a part of market analytics, Citrix mailing servers support communication between the users and partners. Another example is Citrix Marketing Concierge, which is to manage email campaigns, webinars, and roadshows. In addition, knowledge sharing is enabled by Citrix Product Documentation[10] and Partner Central[11]. Using the partner central, the partners access different partner programs. Furthermore, Citrix Discussions[12] and User Group Community[13] are the forums respectively used by the partners and users. Further examples of the partner-based software ecosystems are SAP, VMware Vsphere, Symantec, IFTTT, ExtendSim, and SolidWork.

3.3 OSS-Based Software Ecosystem

28% of ecosystems in our list have grown around open source software platforms. We call them *open source software-based* or shortly *OSS-based software ecosystems*. Ecosystem providers are in form of foundations or consortia. The members of such foundations are software companies or individuals, which unify to create an ecosystem around open source software. Mozilla and Eclipse foundations are examples of such providers.

a) *Architectural landscape*: Table 1 summarizes the way that the OSS-based ecosystems realize the variants. Figure 3(a) illustrates the architectural landscape of OSS-based software ecosystems. **Extender**: Independent developers extend the platforms by directly accessing the source code. Such developers play both roles of extender and user, because the extensions are pieces of code that are collaboratively developed by them. **Openness**: A direct access to the source code gives the developers a high degree of freedom. This mostly releases them from certain business and technical requirements imposed by the ecosystem providers. **Fee**: No entrance or platforms fee is usually demanded. In

[9] citrixready.citrix.com/ [10] docs.citrix.com/ [11] www.citrix.com/partnercentral/
[12] discussions.citrix.com/ [13] www.mycugc.org/



(a) Architectural Landscape

Variation Point	Variant	Apache Cordova
Extender	Trusted Partner	Apache Foundation
	Independent Developer	Committers & contributors
Openness	Open	Apache Cordova, Plug-ins
Fee	Entrance Fee	Free of charge
	Platform Fee	Free of charge
Feedback Loop Facilitator	Ticket System	Atlassian Jira
	Version Control Management	Cordova plug-in versioning
Knowledge Sharing	Documentation Framework	Cordova Documentation Framework
	Q&A Forums	Public Q&A Forums

(b) Exemplary Ecosystem

Fig. 3. OSS-based ecosystem: An architectural solution to address market niche

addition, in 64% of the ecosystems, the extensions are for free. In this situation, the developers might be non-commercially motivated and willing to extend an open source platform to fulfill domain-specific requirements [14]. Still the rest of ecosystems, i.e., 36%, generates revenue for their developers.

Feedback Loop Facilitator: A positive feedback loop between the developers, who commit to the code, and the ones, who reuse it, is created using version control management and ticket system features. A version control management like Apache Subversion supports forking, branching, and merging the code. Additionally, a ticket system like Jira[14] helps to track and communicate issues and bugs. **Knowledge Sharing**: Both Q&A forums and documentation frameworks are heavily used in the OSS-based software ecosystems. An example of Q&A forums is Stack Overflow[15] that facilitates questioning and answering on a wide range of topics in programming languages. An example for documentation frameworks is Markdown[16], which is a markup language with syntax formatting to create enhanced documents like wikis.

Further characteristics: Extensions are published on a public code repository, which, in 78% of the cases, GitHub.com is used. In addition, some ecosystems provide online stores. We assume this happens when an ecosystem is mature and offers commercial and ready-to-use pieces of code. In addition, in OSS-based software ecosystems, ranks are generated implicitly once the developers bookmark or fork a project.

b) *Exemplary real-world ecosystems*: Figure 3(b) outlines the variants of the Apache Cordova ecosystem. Apache Cordova[17] is a framework to develop cross-platform mobile applications. Apache Software Foundation[18] is the ecosystem provider. Adobe, IBM, and Mozilla are the exemplary members. Developers are called committers and contributors. They develop applications using CSS3,

[14] www.atlassian.com/software/jira
[15] stackoverflow.com/
[16] daringfireball.net/projects/markdown/
[17] cordova.apache.org/
[18] www.apache.org/

HTML5, and JavaScript rather than relying on platform-specific APIs. Using Apache Cordova, the code is wrapped to be run on different platforms like Android and iOS. While the source code and the extensions reside on GitHub.com, a list of extensions including their metadata like supported platforms is on a store namely Cordova Plug-ins[19]. Moreover, a feedback loop between the developers is facilitated by the Cordova plug-in versioning as a part of the framework. Furthermore, a ticket system is established by using Jira. As a part of knowledge sharing, a documentation framework[20] provides information on the API reference. The ecosystem does not own any Q&A forum, but many related threads can be found in public forums like Microsoft developer network (MSDN) and Stack Overflow. Further examples of OSS-based software ecosystems are Cloud Foundry, Ubuntu, Odoo, OpenFOAM, CTAN: Packages, Mozilla, Zotero, LibreOffice, and Eclipse.

4 Discussion

In this section, we further discuss the design options with respect to their overall contribution. The arrangement of the actors and software features contribute to an outstanding business goal in each architectural landscape. Considering the high number of extenders, users, and extensions on the stores, the resale software ecosystem can be seen as an architectural solution to achieve **business scalability**. Rating, ranking, and billing features support to establish a market between the extenders and users, while making them highly independent of the ecosystem providers. Furthermore, using the partner-based ecosystem design option, ecosystem providers monetize the platforms and extensions by establishing different degrees of collaborations with trusted partners and including the market analytics features to promote the joint solutions. This ultimately addresses the **commerciality** of an ecosystem. Finally, the OSS-based software ecosystem design option degrades the protection of intellectual property by opening the source code to the developers. However, due to the high availability of tools and resouces for free and open-source software (FOSS) community, the **cost** of opening a platform decreses [3]. Moreover, as the degree of openness increases, an ecosystem suceeds to address **innovation** by providing extensions for market niche [15].

5 Conclusion and Future Work

Many modern software companies create store-oriented ecosystems of third-party providers and users on top of their platforms; online stores serve as distribution channels for third-party developments. This architectural approach has been applied widely; however, the diversity of the existing ecosystem designs hinders prospective ecosystem providers to gain a sound overview of the existing designs and to understand how to design a store-oriented software ecosystem that fulfills certain business decisions.

[19] `cordova.apache.org/plugins/` [20] `cordova.apache.org/docs`

In this paper, we provide empirical evidence that three designs options of these ecosystems are frequently applied in practice: 1) Resale software ecosystem, 2) Partner-based ecosystem, and 3) OSS-based ecosystem. We provide insight into their business decisions and the overall contribution in terms of business goals. This knowledge helps future ecosystem providers to decide on when to apply any of these design options according to their needs. In the future, practical effectiveness and possible combinations of the design options can be further studied by architects on real projects while including new business decisions in the design options.

References

1. J. Bosch, "From software product lines to software ecosystems," in *International Software Product Line Conference*. CMU, 2009, pp. 111–119.
2. J. Bosch and P. Bosch-Sijtsema, "From integration to composition: On the impact of software product lines, global development and ecosystems," *Journal of Systems and Software*, vol. 83, no. 1, pp. 67–76, 2010.
3. K. Manikas and K. M. Hansen, "Software ecosystems–a systematic literature review," *Journal of Systems and Software*, vol. 86, no. 5, pp. 1294–1306, 2013.
4. M. H. Sadi and E. Yu, "Modeling and analyzing openness trade-offs in software platforms: A goal-oriented approach," in *International Conference on Requirements Engineering: Foundation for Software Quality*. Springer, 2017, pp. 33–49.
5. T. Berger, R.-H. Pfeiffer, R. Tartler, S. Dienst, K. Czarnecki, A. Wasowski, and S. She, "Variability mechanisms in software ecosystems," *Information and Software Technology*, vol. 56, no. 11, pp. 1520–1535, 2014.
6. B. Jazayeri, O. Zimmermann, G. Engels, and D. Kundisch, "A Variability Model for Store-Oriented Software Ecosystems: An Enterprise Perspective," in *International Conference on Service-Oriented Computing*. Springer, 2017, pp. 573–588.
7. J. Van Angeren, J. Kabbedijk, S. Jansen, and K. M. Popp, "A Survey of Associate Models used within Large Software Ecosystems." in *International Workshop on Software Ecosystems,*. CEUR-WS, 2011, pp. 27–39.
8. B. Jazayeri, M. C. Platenius, G. Engels, and D. Kundisch, "Features of IT Service Markets: A Systematic Literature Review," in *International Conference on Service-Oriented Computing*. Springer, 2016, pp. 301–316.
9. O. Zimmermann, L. Wegmann, H. Koziolek, and T. Goldschmidt, "Architectural decision guidance across projects-problem space modeling, decision backlog management and cloud computing knowledge," in *Working IEEE/IFIP Conference on Software Architecture*. IEEE, 2015, pp. 85–94.
10. N. Rozanski and E. Woods, *Software Systems Architecture: Working with Stakeholders Using Viewpoints and Perspectives*. Addison-Wesley, 2012.
11. A. Holzer and J. Ondrus, "Mobile application market: A developer's perspective," *Telematics and informatics*, vol. 28, no. 1, pp. 22–31, 2011.
12. V. Grover and R. Kohli, "Cocreating IT value: New capabilities and metrics for multifirm environments," *Mis Quarterly*, pp. 225–232, 2012.
13. "Dataset . http://www.mediafire.com/file/nglv8ozn6gs42fo," Tech. Rep., 2018.
14. G. K. Hanssen, "A longitudinal case study of an emerging software ecosystem: Implications for practice and theory," *Journal of Systems and Software*, vol. 85, no. 7, pp. 1455–1466, 2012.
15. A. Gawer and M. A. Cusumano, "Industry platforms and ecosystem innovation," *Journal of Product Innovation Management*, vol. 31, no. 3, pp. 417–433, 2014.