How do Software Ecosystems Co-Evolve?

A view from OpenStack and beyond

Working paper

Open-access at http://www.jteixeira.eu/pub/wp/ICSOB2017SEESNA.pdf

José Apolinário Teixeira
Åbo Akademi University
and University of Turku
Finland
jteixeira@abo.fi

Sami Hyrynsalmi
Tampere University of Technology
Finland

---

As accepted for presentation at the 8th International Conference on Software Business (ICSOB 2017), held in Essen, Germany, June, 12-13, 2017. The official conference proceedings are available at www.springerlink.com.
This research paper was granted as the **Best Paper Award** at the 8th **International Conference on Software Business** (ICSOB 2017). The Program Chairs based on reviews and discussions among Program Committee members and International Software Product Management Association (ISPMA) board members, selected this paper as the best one presented at the conference. On behalf of the ISPMA, the award was handed over by the ISPMA Board member Prof. Samuel Fricke. The announcement is available at [1]. A brief summary of the paper market by ISPMA at their website (www.ispma.org) is also available at [2]. The authors are very grateful to a fellowship from ISPMA that will support further research efforts.

[1] http://community.ispma.org/icsob-2017-best-paper-award/
[2] http://web.abo.fi/projekt/diwil/icsob_best_paper_award_2017.html

---

**Figure 1**: Best paper award certificate at side of the Software Product Management companion book gently offered by ISPMA.
Copyright notice

The same article is available at Springer’s Lecture Notes in Business Information Processing series (vol. 304). See [http://www.springer.com/gp/book/9783319691909](http://www.springer.com/gp/book/9783319691909) with permission from the authors. The copyright was transferred to Springer right after being accepted as a full research paper for ICSOB 2017. The authors retain however some rights:

- The right to communicate the content of the contribution to other scientists, to share the contribution with them in manuscript form, to perform or present the contribution or to use the content for non-commercial internal and educational purposes, provided that the Springer publication is mentioned as the original source of publication in any printed or electronic materials.

- The right to republish the contribution in any collection consisting solely of authors own works without charge, but must ensure that the publication by Springer is properly credited and that the relevant copyright notice is repeated verbatim.

- The right to self-archive an author-created version at its own website, institutional repository, as well as on a non-commercial archival repository such as ArXiv/CoRR and HAL.

- The right to deposit this version on funder’s designated repositories.

- The right to use, without having to pay a fee and without having to inform the publisher, parts of the contribution (e.g. illustrations) for inclusion in future work, and to publish a substantially revised version (at least 30% new content) elsewhere, provided that the original contribution is properly cited.

Archiving information

The article was self-archived by the first author at its own personal website [http://www.jteixeira.eu/pub/wp/ICSOB2017SEESNA.pdf](http://www.jteixeira.eu/pub/wp/ICSOB2017SEESNA.pdf).
Funding and Acknowledgements

Besides the three anonymous reviewers for ICSOB 2018, we would like to thank participants from the 1st European Conference on Social Networks (IN-SNA EUSN 2014), the 10th International Symposium on Open Collaboration (ACM OpenSym 2014), the 2nd IEEE International Conference Big Data (IEEE Big Data 2014), the 38th Information Systems Research Conference in Scandinavia (IRIS 2015), and the 37th International Conference on Information Systems (AIS ICIS 2016) for early feedback on our research efforts.

This research was partially funded by the Fundação para a Ciência e a Tecnologia (grant SFRHBD615612009), Liikenesivistysrahasto (grants 5-3076 and 8-4499), Marcus Wallenberg Säätiö (grant “open-coopetition R&D management strategy”) and Academy of Finland (decision no. 295743). We thank also the University of Turku, the Tampere University of Technology and Åbo Akademi for providing much of the needed research infrastructure. Finally, we would like to thank the R, Gource, Gephi, Tulip, Blender, and the OpenStack open-source communities for developing cool and research-friendly software.
This working paper was formatted in Latex and its available in PDF v.1.5 with rich metadata. It is based on the 'working papers' template from the first author. Special thanks to Vytas Statulevicius from VTeX and Jacky Lee from the Chinese University of Hong Kong for ‘prior art’ on the SpringerOpen BMC template.

\textsuperscript{3}Version details: pdfTeX, Version 3.14159265-2.6-1.40.15 (TeX Live 2015/dev/Debian)
Abstract

Much research that analyzes the evolution of a software ecosystem is confined to its own boundaries. Evidence shows, however, that software ecosystems co-evolve independently with other software ecosystems. In other words, understanding the evolution of a software ecosystem requires an especially astute awareness of its competitive landscape and much consideration for other software ecosystems in related markets. A software ecosystem does not evolve in insulation but with other software ecosystems. In this research, we analyzed the OpenStack software ecosystem with a focal perspective that attempted to understand its evolution as a function of other software ecosystems. We attempted to understand and explain the evolution of OpenStack in relation to other software ecosystems in the cloud computing market. Our findings add to theoretical knowledge in software ecosystems by identifying and discussing seven different mechanisms by which software ecosystems mutually influence each other: sedimentation and embeddedness of business relationships, strategic management of the portfolio of business relationships, firms values and reputation as a partner, core technological architecture, design of the APIs, competitive replication of functionality and multi-homing. Research addressing the evolution of software ecosystem should, therefore, acknowledge that software ecosystems entangle with other software ecosystems in multiple ways, even with competing ones. A rigorous analysis of the evolution of a software ecosystem should not be solely confined to its inner boundaries.
Resumo

Muitas pesquisas que analisam a evolução de um ecossistema de software estão confinadas a seus próprios limites. As evidências mostram, no entanto, que os ecossistemas de software coevoluem em função de outros ecossistemas de software. Em outras palavras, entender a evolução de um ecossistema de software requer uma percepção especialmente astuta de seu cenário competitivo e muita consideração por outros ecossistemas presentes no mercado. Um ecossistema de software não evolui em isolamento, mas com outros ecossistemas de software. Nesta nossa pesquisa, nos analisamos o ecossistema do software OpenStack com uma perspectiva focal que tentou entender sua evolução em função de outros ecossistemas de software. Tentamos entender e explicar a evolução do OpenStack em relação a outros ecossistemas de software no mercado de computação em nuvem. As nossas descobertas extendem teoria em ecossistemas de software, identificando e discutindo sete mecanismos pelos quais os ecossistemas de software se influenciam mutuamente: sedimentação e imersão nas relações de negócios, gestão estratégica do portfólio de relações comerciais, ideais e reputação como parceiro, arquitetura tecnológica, design das interfaces de programação, replicação competitiva de funcionalidade e multi-homing. Futuras pesquisas que abordem a evolução do ecossistema de software devem portanto reconhecer que os ecossistemas de software se envolvem com outros ecossistemas de software de várias maneiras, mesmo com os ecossistemas concorrentes. Uma análise rigorosa da evolução de um ecossistema de software não deve ser confinada apenas a seus próprios limites.
How do Software Ecosystems Co-Evolve?
A view from OpenStack and beyond

Apolinário Teixeira and Hyrynsalmi

Abstract

Much research that analyzes the evolution of a software ecosystem is confined to its own boundaries. Evidence shows, however, that software ecosystems co-evolve independently with other software ecosystems. In other words, understanding the evolution of a software ecosystem requires an especially astute awareness of its competitive landscape and much consideration for other software ecosystems in related markets. A software ecosystem does not evolve in insulation but with other software ecosystems. In this research, we analyzed the OpenStack software ecosystem with a focal perspective that attempted to understand its evolution as a function of other software ecosystems. We attempted to understand and explain the evolution of OpenStack in relation to other software ecosystems in the cloud computing market. Our findings add to theoretical knowledge in software ecosystems by identifying and discussing seven different mechanisms by which software ecosystems mutually influence each other: sedimentation and embeddedness of business relationships, strategic management of the portfolio of business relationships, firms values and reputation as a partner, core technological architecture, design of the APIs, competitive replication of functionality and multi-homing. Research addressing the evolution of software ecosystem should, therefore, acknowledge that software ecosystems entangle with other software ecosystems in multiple ways, even with competing ones. A rigorous analysis of the evolution of a software ecosystem should not be solely confined to its inner boundaries.

Keywords: alliances; business ecosystem; software ecosystem; evolution; coopetition; open-coopetition; open-source; OpenStack

I. Introduction

In the so-called ‘Information Age’ companies and organizations do not live in isolation; instead, business activities of modern companies are highly interwoven with other companies. Furthermore, the fate of a company nowadays depends on its connections and environment where they are working — not anymore solely on the company itself. From these observations, James F. Moore [1] built his theory and concept of ‘business ecosystem’. According to Moore [2], a business ecosystem consists of a set of companies working on a shared innovation. The companies work together, cooperatively and competitively, for

[1] As accepted for presentation at the 8th International Conference on Software Business (ICSOB 2017), held in Essen, Germany, June, 12-13, 2017. The official conference proceedings are available at www.springerlink.com.
creating value for customers; the ecosystem advances as companies and the innovation co-evolve together.

Since Moore’s seminal article, a plethora of different kinds of artificial ecosystems has been defined and used [3]. One of the most important is ‘software ecosystem’ as software is pervasive and ubiquitous by its nature — there hardly is any industrial domain where software would not be a part of. That is, software is available nowadays everywhere and it is rarely built on isolation. Software ecosystems have become also an important research field and there are hundreds of studies addressing different kinds of software ecosystems [c.f. 4, 5].

As proposed by Jansen et al. [6], a software ecosystem consists of “the set of businesses functioning as a unit and interacting with a shared market for software and services, together with the relationships among them. These relationships are frequently underpinned by a common technological platform or market and operate through the exchange of information, resources and artefacts.” From the definition, it seems clear that there are a relationship between software ecosystem and business ecosystem conceptualizations. However, there is one major caveat: Whereas Moore’s view on business ecosystems focused on co-evolution, the definition of software ecosystem does not cover this aspect.

Since the ecosystem concept has been accepted as a perspective for business development, management and governance, it is also necessary to discuss the interactions between ecosystems. Whether and how ecosystems influence each other and to what degree, are intriguing questions. That is, it is essential for both scholars and practitioners to analyze the competitive landscape, in order to better understand software ecosystem evolution. This study focuses on the relatively uncovered area in the field of software ecosystems: the co-evolution of them. Specifically, we focus on how ecosystems influence each other evolution (i.e. co-evolution of ecosystems). The starting research hypothesis is that an ecosystem does not evolve in isolation; instead, the ecosystems are interwoven with each other (e.g., characteristics of one can affect the other or change in one can also affect the other). That is, this study seeks to answer a question

**RQ** How do software ecosystems co-evolve?

To answer the presented research question, we performed a case study by taking the OpenStack software ecosystem as our focal unit of analysis. By analyzing OpenStack in relation to other software ecosystems in the industry (e.g., CloudStack) we identified and explored seven different ways in which software ecosystems are interwoven with other software ecosystems:

- a) **Sedimentation and embeddedness of business relationships**,  
  b) **Strategic management of the portfolio of business relationships**,  
  c) **Firm’s values and reputation as a partner**,  
  d) **Core technological architecture**,  
  e) **Design of external APIs**,  
  f) **Technological replication of new functionalities**, and  
  g) **Complementors’ multi-homing**.

The remaining of this paper is structured as follow. The next section will briefly present the related work whereas Section III views on the empirical background of our study subject.
Section IV presents the research approach used, Section V results and Section VI their implications and limitations. The final section concludes the study.

II. Related literature

Moore [1], in his seminal essay on the new ecology of competition, defined that business ecosystems evolve through distinct phases. He identified and named four stages which are:

1. **Birth** where companies define value propositions of a seed innovation.
2. **Expansion** where the ecosystem seeks to expand to new territories.
3. **Leadership** where participating companies start to struggle for a leadership. As an example, Moore used Microsoft’s and Intel campaign against IBM during the “clone wars” of personal computers.
4. **Self-Renewal or Death** where an ecosystem faces an external threat and it is forced to either renew itself or cease to exist.

As software ecosystem share distinct similarity with the older ecosystem concept, it is surprisingly how little have been written on the evolution of different kinds of ecosystems. However, there are a few prior studies existing. Similarly, as have entire software ecosystem literature diverged into communities [3], also existing studies can be categorized into two groups with a remarkable different basis.

In the first group, there are studies addressing software ecosystem as a business network and previous studies have addressed how the relationships between the firms have developed. For example, Basole [7] studied the convergence of entire mobile ecosystem—including software and hardware vendors as well as network operators—however, it focused on the interfirm relationships and visualization of cooperative networks. Basole and Karla [8] studied the evolution of mobile platform ecosystems. However, also their focus was on the visualization and on the interfirm relationships inside an ecosystem. Hanssen [9] followed a transformation of a product line organization to an emerging software ecosystem and focused on why and how the transformation was done.

In the second group, there are studies addressing software ecosystems as a collection of interdependent projects and these studies on the evolution of software ecosystem’s codebase over time [e.g. 10, 11]. Already 2007, Yu and Bush [12] noted that software projects evolve and there are certain types of relationships between the actors. Later, Yu et al. [13] adapted different symbiosis types, that might affect the evolution of projects, from biology and applied them to relationships between software projects. Furthermore, Scacchi and Alspaugh [14] studied how different licenses affect on the ecosystem evolution.

To summarize, while there are few studies addressing the evolution of software ecosystems, they represent different ends of the spectrum: On one corner, there are studies on the relationships and evolution of software code base; and on the other corner, studies have focused on visualizing interfirm relationships of companies with Social Network Analysis. To the best of the authors’ knowledge, this study is unprecedented as it combines both of the existing schools of thought in the study of ecosystem evolution: We study software ecosystems as a business network construction but acknowledge the importance of source code and address the evolution of the ecosystem and interfirm relationships through the developments in the shared codebase. In addition, we specifically focus on the co-evolution of competing...
software ecosystems. Generally, *co-evolution* refers to cases where two entities affects to each others’ evolution. Here, the entities are ecosystems and their actors.

### III. Empirical background

The cloud computing business is dominated by a relatively small number of players, including 1) Amazon, a pioneer in cloud computing services selling the Amazon EC2; 2) Google, selling services around its Compute Engine (Google Compute); and 3) Microsoft, heavily marketing cloud strategies based on its Azure cloud computing infrastructure (Microsoft Azure). The entrance costs for building and providing a public cloud computing infrastructure are very high as they often require global-distributed data-centers, fast and large accesses to the Internet backbone, much computing and storage power. Public cloud providers must provide very low latency – after all, they are convincing its enterprise customer to move from self-managed in-house computing infrastructures to vendor-managed computing infrastructures out there.

The leader of the cloud computing industry (i.e., Amazon, Google, and Microsoft) do not provide cloud infrastructure products, merely computing services. In practice and if there were no alternatives, all cloud computation would run in hardware and software infrastructures controlled by very few players. Such control from the cloud computing service provider locks-in its customers[15]. Surprisingly, the leading product alternatives to Amazon EC2, Google Compute and Microsoft Azure are not commercial but rather four open-source projects. They include: 1) OpenStack, our unit of analysis; 2) CloudStack, backed by Citrix and the Apache Software Foundation; 3) Eucalyptus, a system that is compatible with Amazon EC2 services and backed by many IT consulting firms; and 4) OpenNebula, more present in the European markets and backed by C12G, a Spanish company. During our research, we perceived that many cloud computing vendors associated with the leading open-source cloud computing ecosystems to ease the pain of “selling cloud computing services that are famous and infamous for their single-vendor locking mechanisms”.

OpenStack is a software cloud computing infrastructure capable of handling big data. It is often offered as an IaaS (Infrastructure-as-a-Service) solution. The development of this open-source software involves private companies (such as AT&T, Canonical, Ericsson, IBM, Intel *etc.*), public organizations (such as NASA, CERN, Johns Hopkins University *etc.* ) as well as independent, non-affiliated individuals.

We selected OpenStack as our case study subject due to four main factors. First, it is truly heterogeneous software ecosystem including start ups, high-tech corporate giants, non-profit and public organizations as well as individual software developers. Second, it is highly inter-networked. That is, there are several companies and individual contributors working together. Thus, there is a rich data available for co-evolution. Third, its size is large enough for a meaningful study (more than 70,000 individual contributors and more than 600 supporting companies from 185 countries that have contributed with more than 20 million lines of source-code[2]). Finally, it is well-studied [see e.g. 16] and, therefore, there is a good amount of scholarly information published.

---

[12] See [https://www.openstack.org/community/](https://www.openstack.org/community/).
IV. Method

In this section, we present our research design. Given the multidisciplinary nature of our research approach which borrowed significantly across disciplines, many interwoven methodological issues are disclosed. We employed a case study research strategy [17] that relied on naturally occurring data which emerged *per se* on the Internet. Such data (e.g., web pages, wikis, blogs, public announcements, market-research reports, technical documentation, the software, source code repositories, videos broadcasted from the OpenStack summits, among many others data sources) are not a consequence of researchers’ own actions, but rather are developed by the OpenStack community in their own pursuits of developing an open-source infrastructure for handling and storing big amount of data.

Given the open-source nature of our focal unit of analysis, many but heterogeneous data regarding OpenStack is available. Therefore we have selected a novel approach by combining three well-known technique: mining software repositories (MSR) of OpenStack repository, Social Network Analysis (SNA) of the contributing developers, and qualitative analysis of archival data (QA). All within a mixed methods design, that reconstruct as well as visualize the evolution of the software ecosystem as a sequence of networks connectiong firms and individuals that jointly develop the OpenStack ecosystem.

We started our efforts qualitatively by searching publicly available data sources such as news articles, public announcements by companies, financial figures as well as press reports. Those helped us to create a picture of the cloud computing industry where OpenStack is a part of. In addition, we went through OpenStack documentation regarding how the software ecosystem is developed (i.e. the technical information) and governed (e.g., structures, policies, and procedures). While keeping in mind the limitations on the use of archival data [17], we gained valuable insights from OpenStack community and its surrounding industrial environment. After gaining an understanding of the surrounding industrial dynamics and understanding of how OpenStack software is developed, we extracted the developer and affiliation information from the publicly-available OpenStack Nova repository. Then, we created and analyzed the social network of the project by using the SNA guidelines given in [18].

As in [16], we took advantage of naturally occurring digital trace data (i.e., the OpenStack Nova project repository and its *changelog*) and built cooperative social networks that were analyzed using a variety of tools: *Gephi, Visone*, and the *sna* and *statnet* statistical modules for *R*. To better explore cooperation at the ecosystems level, we also modeled cooperative relationships in the tri-dimensional (3D) space using *Blender*. We mined evidence of cooperation from the source code and by visualized the social structures with SNA. This revealed the cooperation in the OpenStack ecosystem and we later enrichment this data with qualitative information from the public sources used in QA. The use of all these methods were helpful in terms that they both showed the social structures as well as helped to explain them.

We highlight the visualization of the collaboration network. The changes in this network, over time, show the dynamics among the OpenStack ecosystem. We aim to understand the visualized networks with the information gathered from the industry in previous steps. In this, we follow prior work [e.g. 7, 16] done in multi-disciplinary settings.
V. Results

We present our results in a chronological narrative format. The textual narrative is complemented with visualizations that capture the evolution of the OpenStack ecosystem. Besides richly describing the evolution of the OpenStack ecosystem, we also attempt to interpret such evolution and explain it by employing multiple theoretical lenses. Our analysis aggregates both empirical and theoretical issues that are later addressed in the discussion section.

We start with the words of, at that time Senior Vice President and General Manager of Rackspace, Jim Curry. In this, the first public disclosure of the OpenStack project, Curry emphasizes the roles of NASA’s and Rackspace’s roles as initial contributors to the project – that is, it is built with experienced partners and the project did not start from scratch.

“Our mission statement says this: To produce the ubiquitous Open Source Cloud Computing platform that will meet the needs of public and private clouds regardless of size, by being simple to implement and massively scalable. That is a big ambition. The good news is that OpenStack is starting with code contributions from two organizations that know how to build and run massively scalable clouds – Rackspace and NASA.” — Jim Curry, founder of OpenStack on behalf of Rackspace, 19 July 2010[3]

The footsteps of Rackspace in NASA started as a supplier of Anso Labs. A startup company which was later acquired by Rackspace on February 9, 2011[4]. Before OpenStack, Anso Labs and Rackspace have been working in Nebula – a Federal cloud computing platform. Nebula emerged at NASA Ames Research Center at Moffett Field, California in 2008. It allowed NASA researchers to manage the computation of data-intensive research projects in a cloud computing way. The design of Nebula reflected the growing popularity of the Amazon Web Services (AWS) cloud computing environments.

“Nebula’s architecture is designed from the ground up for interoperability with commercial cloud service providers such as Amazon Web Services, offering NASA researchers the ability to easily port data sets and code to run on commercial clouds.” — NASA under the Open Government Initiative, 7 April 2010[5]

The NASA Nebula team started by adopting the Eucalyptus open-source cloud computing infrastructure (now a competitor of OpenStack), as it resembled the EC2 compute cloud and S3 storage cloud technologies from Amazon. However, NASA faced scalability issues. After all, NASA demands computing and storage were very high. Nebula could accommodate files as large as eight terabytes. Furthermore, Nebula could support only an individual file system of 100 terabytes. As an example, the maximum for Amazon EC2 file size was just one terabyte and for file system size was also one terabyte[6].

In addition of scaling requirements to handle big data, NASA engineers were not happy with the ‘open-core’ business model strategy of Eucalyptus Systems Inc to monetize its

[3]See https://www.openstack.org/blog/2010/07/introducing-openstack/.
[4]See https://gigaom.com/2012/05/24/nasa-backs-off-openstack-development/.
[5]See https://www.nasa.gov/pdf/440932main_Nebula.pdf.
[6]See https://www.nasa.gov/open/nebula.html.
cloud computing software ecosystem. According to NASA, Eucalyptus-based clouds were not entirely open-source.

“NASA engineers attempted to contribute additional Eucalyptus code to improve its ability to scale, they were unable to do so because some of the platform’s code is open and some isn’t. Their attempted contributions conflicted with code that was only available in a partially closed version of platform maintained by Eucalyptus Systems Inc., the commercial outfit run by the project’s founders.” — Chris Kemp, NASA chief technology officer, 20 July 2010[7].

As argued in prior related research [see 16, 19], the visualizations in Figures 1 to 3 helps us to understand how the cloud industry’s actors cooperate in OpenStack. Such visualizations, obtained with combining MSR and SNA, helps us to visualize the evolution of the software ecosystem as an evolving complex network of companies and individuals interacting with each other to develop complex[9] software. The diameter of a node reflects its degree-centrality – in other words, a large node depicts a well-connected developer. The value of degree-centrality is a sum of the number of adjacent nodes with which a focus node is connected to. Thus, a high degree-centrality value, the more likely the developer is to be cooperating with other developers.

To start with, Figure 1 captures the cooperation in the OpenStack Nova project from the Austin (October 21st 2010) to the Bexar (February 3rd 2011) release. The figure illustrates the cooperation between individual software engineers and their affiliated companies. For example, as shown by the figure, Citrix had three developers working on the project together with Rackspace.

Citrix’s, who had worked before with Rackspace in Desktop visualization technologies[10], aim was to ensure that their XenServer platform would be included in OpenStack’s future plans.

“As a longtime technology partner with Rackspace, Citrix will cooperate closely with the community to provide full support for the XenServer platform and our other cloud-enabling products.” — Peter Levine, SVP and GM, Citrix, 19 July 2010[11].

Our second visualization, in Figure 2, captures the cooperation from the Bexar (February 3rd 2011) to the Cactus (April 15th 2011) release. The figure illustrates the entrance of a new actor, a developer from the company Cloudscaling.

The company started in 2006 with personnel previously working for Amazon and VMWare. It started by selling customized cloud infrastructures for large service providers. For example, Cloudscaling had Korea Telecom as an early customer. In 2010, the company shipped an OpenStack-based storage cloud to Korea Telecom. It was first OpenStack delivery without

[7] See https://www.theregister.co.uk/2010/07/20/why_nasa_is_dropping_Eucalyptus_from_its_nebula_cloud/.
[8] Please note that all figures are encoded as Scalable Vector Graphics, therefore readers can freely zoom in and zoom out for a better visualization of the networks.
[9] Complex as it involves different programming languages, different operating systems, dozens of different hardware configurations, hundreds of firms, thousands of software developers, and over one million of lines of code
[10] See https://ir.rackspace.com/phoenix.zhtml?c=221673p=ir01-newsArticleID=1608440.
[11] See https://www.rackspace.com/blog/newsarticles/rackspace-open-sources-cloud-platform.
Rackspace. Together with Mirantis, they were among the first pure-play OpenStack firms deploying OpenStack-based in-premise private clouds (e.g. Korea Telecom and PayPal). While CloudScaling kept a strategy of compatibility with Amazon EC2 APIs, Mirantis was more on the position that OpenStack should not follow the designs of its competitor but challenge it.

“We are introducing a cloud infrastructure suite of products that essentially delivers an Amazon Web Services-like cloud, but on a customer’s premise.” — Michael Grant, Cloud-scaling’s CEO, 9 February 2012.

Our visualizations in Figures 3 and 4 capture cooperation from the Cactus (April 15th 2011) to the Diablo (September 22nd 2011) release and cooperation from the Diablo (September 22nd 2011) to the Essex (April 5th 2012) release. HP and IBM (large IT companies), Mirantis (an OpenStack pure-play startup), Red Hat (a Linux operating system distribution’s vendor), Canonical (company behind the Ubuntu Linux distribution), VMware (an expert on the virtualization software and services) and Intel (selling CPUs that powered cloud infrastructures) got involved in the coopetitive software project.

[12] See presentation entitled “OpenStack Co-Opetition: A View from Within” from Boris Renski (co-founder and chief marketing officer of Mirantis) presented on 04 Nov 2013 at the OpenStack summit, Hong Kong. Available on youtube at https://www.youtube.com/watch?v=i7Rkx2abfj0.

[13] See Nancy Gohring news article at http://www.infoworld.com/article/2619192/.

[14] Coopetitive as firms within OpenStack cooperate and compete simultaneously. See [16, p. 6] for a relational map of competition among OpenStack firms.
Figure 2 Bexar → Cactus [16].

Figure 3 Cactus → Diablo [16].
Mirantis, founded in 2011, marketed itself as a “pure-play” OpenStack company. The startup started collaboration early with Red Hat. Besides cooperating in the development of OpenStack, both firms partnered in implementation and integration services at common customers\textsuperscript{[15]}. Mirantis was involved in the early deployments of OpenStack at large enterprises such as Paypal, AT&T, Comcast, and Wells Fargo among others.

In the meantime, HP launched an OpenStack-based cloud computing services. The company started marketing itself as the leading organization behind the project. In addition, they marketed OpenStack as free of single-vendor locking as there were a full ecosystem behind the project\textsuperscript{[16]}. At that time, it was the only cloud computing solution with a such promise. After all, cloud computing services are known by locking-in its customers \textsuperscript{[15]}.

With a very good track of contributions to open-source projects IBM (top contributor to the Eclipse IDE project), RedHat (top contributor to the Linux kernel) and Canonical (top contributor to the GNOME Linux Desktop project) also joined OpenStack. In common, all those companies had much expertise on Linux, the host operating system of OpenStack. RedHat and Canonical aimed at being the \textit{de facto} host operating system for OpenStack-based clouds\textsuperscript{[17]}

IBM entered with force in OpenStack and showed much commitment to the platform. Besides contributing with much source-code to the project, it helped many of its customers to

\textsuperscript{[15]}See https://www.redhat.com/en/about/press-releases/red-hat-and-mirantis-partner-across-products-and-services.
\textsuperscript{[16]}See https://www.openstack.org/foundation/companies/profile/hewlett-packard-enterprise.
\textsuperscript{[17]}See https://www.openstack.org/blog/2013/11/openstack-user-survey-october-2013/.
deploy openStack. Moreover, it entered into the public cloud business with OpenStack as well. In the case of IBM, as well with HP and Intel, money could be made by selling complementary hardware optimized for OpenStack. On the space of virtualization technologies, VMware did not want to lose ground to Citrix, and its contributions to OpenStack ensured compatibility with its vSphere, NSX, vSOM and vCloud offering.[18]

VI. Discussion

In this section we discuss our most significant results. The structure of the discussion reflects our mixed-methods analytical approach where we attempted to make sense of the retrieved social network visualizations capturing cooperative relationships within a complex software ecosystem. After all, “the fundamental quest of SNA is to understand the structure of the network” [20, p. 36].

A theoretical and empirical evolutionary approach to software ecosystems

In order to understand and explain why the retrieved social network visualizations took such topology and not other, much theoretical and empirical background knowledge was required. The use of certain theory to understand and explain our results was complemented with our understanding of the competitive cloud computing industry in which OpenStack is embedded, as well as with our understanding of how OpenStack is developed and governed. Besides literature directly addressing software ecosystems [c.f. 6, among many others], our explanation integrated as well with theory on the embeddedness of business relationships [21], management of the portfolio of business relationships [22], cooperation among competitors [23], materiality of technology [24], innovation and intellectual property [25] and multi-homing strategy [26]. Empirical and theoretical knowledge was complementary – we could not explain the complex evolution of the OpenStack ecosystem without much knowledge on the surrounding industrial background of OpenStack or knowledge on the internal socio-technical practices by key actors within the development of OpenStack. Furthermore, a theory was fundamental to derive why the cooperative relationships (captured with SNA from the source-code) within OpenStack evolved in one way and not other.

By tentatively explaining the retrieved networks, and by focusing on OpenStack in relation to other software ecosystems in the industry, we identified seven mechanisms that shaped the evolution of OpenStack. Such mechanisms are not internal to OpenStack, but are rather enacted by other software ecosystems in the industry. In other words, we identified different ways on how do software ecosystems mutually co-evolve. We found seven mechanisms by which software ecosystems mutually influence each other — but we do not reject the existence of others. The mechanisms reported here can be seen as that drive the evolution of a software ecosystem in relation to others. Interesting enough, some of those identified causal mechanisms are enacted by competing software ecosystems.

Mechanisms of co-evolution among software ecosystems

In the following, we will present the identified co-evolution mechanisms. The list is not complete and further work is needed to validate the mechanism. In addition, some of the

[18] See http://www.vmware.com/products/openstack.html.
mechanisms may overlap partially; however, we have decided to present them separately in order to validate or reject them in further studies.

\(\alpha\) Sedimentation and embeddedness of business relationships: When analyzing the complex history of OpenStack, we quickly notice that prior business relationships had much impact on the evolution of OpenStack. For example, Rackspace entered in the NASA Nebula project (the precursor of OpenStack) as a supplier of Anso Labs (a company that it was later acquired by Rackspace). Another early contributor to OpenStack, Citrix, worked before with Rackspace in the development of Desktop visualization technologies before embracing OpenStack. As pointed out by strategic management theory, business partnerships often accumulate in a process of sedimentation [c.f. 27]. Moreover, actors tend to cooperate with actors that they had previously worked with and tend to buy from existing suppliers (over new suppliers) — all in the so-called paradox of the embeddedness of business relationships [c.f. 21].

\(\beta\) Strategic management of the portfolio of business relationships: Many of the firms contributing to OpenStack (e.g., IBM, HP, and RedHat) manage a vast portfolio of business partnerships. However, the value from a stand-alone partnership may not necessarily be value-creating from the overall portfolio perspective. Potential synergies between multiple alliances must be balanced to mitigate conflicts with other alliances [c.f. 22]. Companies such as IBM and HP needed to show commitment to OpenStack as contributions to other cloud computing ecosystems could potentially damage cooperation. Firms that strategically engage with a software ecosystem might not be able to participate in competing others to not damage existing relationships. After all, the “friends of my enemies are my enemies” and the “enemies of my enemies are my friends”.

\(\gamma\) Firms values and reputation as a partner: Another found mechanism with significant impact on the co-evolution of software ecosystems is each firm values and reputation as a good partner. Something especially important in cooperation among competitors [c.f. 23, 19]. In the case of OpenStack, RedHat had a good reputation as a top contributor to Linux; IBM had a good reputation as a top contributor to Eclipse; Mirantis had a good reputation in deploying OpenStack to its customers while contributing back upstream to the community. On the other hand, Eucalyptus had lost some of its reputation of working in a truly open-source way by closing parts of the Eucalyptus cloud computing ecosystem. The evolution of software ecosystems is then a function of the values and reputation of existing and possible participants. With the loss of reputation, a player might disappear from an ecosystem while becoming unwelcome in others as well.

\(\delta\) Core technological architecture: OpenStack is only functional with a “host OS”. At the early days of OpenStack, Citrix and Rackspace welcomed much the expertise on Linux from RedHat and Canonical. In the cooperative side, they could optimize the “host OS” to better run OpenStack. On the other hand, on the competitive side, this two companies competed with others for customers with the sales argument that “we know Linux, we know OpenStack and we are the only ones that can support both”. This reminded us that the stacking of architectural layers influences the software ecosystem evolution. The materiality of a software ecosystem [c.f. 24] influences the materiality of other software ecosystems. In
the case of OpenStack, the architecture of the OpenStack core was very much influenced by the architecture of Linux, the architecture of Eucalyptus (now a competitor) and consequently the architecture of Amazon EC2.

e) Design of external APIs: We also noted that not only the technological architecture of a software ecosystem influences its co-evolution with others, but also the design of its external APIs. At the early days of OpenStack, the design of the APIs from OpenStack, Eucalyptus, and Amazon EC2 converged. As the open-source cloud computing alternatives matured, Eucalyptus pursued compatibility with the established Amazon AWS APIs while OpenStack opted to diverge and provided interfaces to its computing and storage services in a distinct way. At that time, many decided to get away from OpenStack and move to Eucalyptus and CloudStack because its API was very different from the established Amazon AWS APIs. Many customers wanted to easily move applications from Amazon EC2 to their own private clouds and the other way around – Amazon AWS still remains the leader in public cloud services.

ζ) Competitive replication of new functionality: Competition forces players to copy functionality from each other to keep the pace. Besides the existence of many intellectual property protection mechanisms, this happens quite often in the software industry [c.f. 25]. OpenStack started by implementing (and improving) many of the functionalities provided by Eucalyptus and Amazon E2C. Moreover, whenever firms tried to embed OpenStack within a proprietary product (e.g., a new proprietary cloud orchestrator that embedded OpenStack in it), OpenStack implemented a "new, official and open-source version" of it (i.e., OpenStack launched a new sub-project implementing the official orchestrator for OpenStack). In other words, OpenStack expanded its core by replicating complements. After all, copying an idea and making it open-source is often a more powerful tactic than copying open-source software and making it proprietary.

η) Complementors’ multi-homing: Due to the nature of software, the complements of ecosystems are often intangible. This means that these products and services (that add much value to the overall ecosystem) can be transferred or moved to a different ecosystem and setting with a relatively little effort. The phenomenon where a single complementing actor is offering his or her products or services to two or more ecosystems at the same time is called multi-homing [c.f. 28, 29, 30]; the opposite strategy is known as single-homing. For example, due to the success of the ‘Flappy Bird’ mobile game, it was quickly copied into all major mobile application ecosystems, and beyond [31]. Within OpenStack, Xen from Citrix and ESXi from VMWare are Hypervisors[19] that work both in the OpenStack and Eucalyptus software ecosystems. The contributions that shape the ecosystem evolution, are often due to the complementors interest in making their offering available across different ecosystems. Citrix and VMWare wanted to be sure that their hypervisors run on different cloud computing platforms. In the realm of CPUs, Intel and AMD, also contributors to the OpenStack, are also interested in making sure that their CPUs work across different cloud computing platforms — therefore they contribute to most open-source cloud computing

[19]A hypervisor is either a software or a hardware solution that creates, follows and runs virtual machine instances.
software ecosystems. The same happens with Cisco, Juniper Networks, IBM and HP among other vendors of networking technology.

**Implications and limitations of the study**

Besides richly narrating the evolution of a software ecosystem, our focal perspective of attempting to understand the evolution of a software ecosystem as a function of other software ecosystem extends the literate on software ecosystems evolution [7, 9, 13].

Future research towards a deeper understanding of ecosystems’ evolution should acknowledge that ecosystems do not evolve in insulation. Careful analysis of a software ecosystem evolution should take in consideration other software ecosystems as well, including competing ones. We argue then that, in order to understand the evolution of an ecosystem, we need to look way beyond it. New methodologies, capable of capturing inter-ecosystem dependencies, are needed to addresses such findings [c.f. 32, 19, for recent advancements in this direction].

Naturally, there are certain limitations for this study. First, this study uses a single case study research design to identify mechanisms of co-evolution. Therefore, it is likely our list of mechanisms is not full and further work is needed in order to validate the identified mechanism as well as to find new ones. In addition, we used one ecosystem as a focal point and studied co-evolution from its point-of-view. While we selected the case study ecosystem carefully, there are threats involved in the single case study research design. In future work, multiple ecosystem point-of-view should be used to validate our results.

Second, we selected an open-source software ecosystem as the case subject and generalizing the results to the other kinds of ecosystems should be done with care. There are some previous discussion on the limits on generalizing results from different kinds of software ecosystems to another kinds [c.f. 31, 3]; however, the results of this study does not heavily rely on a certain ecosystem type. Therefore, they should be generalizable to, at least, open-source software ecosystem and, with some limitations, to general type of software ecosystems.

Third, we used developers’ point-of-view in studying co-evolution of software ecosystems. Another option could be study the business connections between the participating companies [c.f. 8, 32]. However, as software ecosystem are built from the developers’ point-of-view [5], our decision seem to justifiable. Nevertheless, the further studies should pay attention also on other perspectives of ecosystems’ co-evolution.

**VII. Conclusion**

Our findings contribute to a deeper understating of the evolution of software ecosystem. We found that a software ecosystem co-evolve with other software ecosystems in at least seven different ways. Understanding the evolution of a software ecosystem requires an especially astute awareness of its competitive landscape as well as knowledge on its internal socio-technical practices. Research addressing the evolution of software ecosystem should, therefore, acknowledge that software ecosystems entangle with other software ecosystems in multiple ways, even with competing ones.

**Acknowledgements**
Besides the three anonymous reviewers for ICSOB 2018, we would like to thank participants from the 1st European Conference on Social Networks (INSNA EUSN 2014) [33], the 10th International Symposium on Open Collaboration (ACM OpenSym 2014) [34], the 2nd IEEE International Conference Big Data (IEEE Big Data 2014) [35], the 38th Information Systems Research Conference in Scandinavia (IRIS 2015) [36], and the 37th International Conference on Information Systems (AIS ICIS 2016) [19] for early feedback on our research efforts.

This research was partially funded by the Fundação para a Ciência e a Tecnologia (grant SFRHBD615612009), Liikesivistysrahasto (grants 5-3076 and 8-4499), Marcus Wallenberg Säätiö (grant “open-coopeetition R&D management strategy”) and Academy of Finland (decision no. 295743). We thank also the University of Turku, the Tampere University of Technology and Åbo Akademi for providing much of the needed research infrastructure. Finally, we would like to thank the R, Gource, Gephi, Tulip, Blender, and the OpenStack open-source communities for developing cool and research-friendly software.

Competing interests
The authors declare that they have no competing interests.

Author’s contributions
JAT conducted the data collection. SH participated in the analysis and contributed equally to the crafting and writing of this article.

References
[1] James F. Moore. “Predators and Prey: A New Ecology of Competition”. In: Harvard Business Review 71.3 (May 1993), pp. 75–86.
[2] James F. Moore. The Death of Competition: Leadership and Strategy in the Age of Business Ecosystems. New York, NY, USA: Harper Business, 1996.
[3] Arho Suominen, Sami Hyrynsalmi, and Marko Seppänen. “Ecosystems Here, There, and Everywhere — A Barometrical Analysis of the Roots of ‘Software Ecosystem’”. In: Software Business: 7th International Conference, ICSOB 2016, Ljubljana, Slovenia, June 13-14, 2016, Proceedings. Ed. by Andrey Maglyas and Anna-Lena Lamprecht. Springer International Publishing, 2016, pp. 32–46. ISBN: 978-3-319-40515-5. DOI: 10.1007/978-3-319-40515-5_3.
[4] Konstantinos Manikas and Klaus Marius Hansen. “Software ecosystems — A systematic literature review”. In: Journal of Systems and Software 86.5 (May 2013), pp. 1294–1306. ISSN: 0164-1212. DOI: 10.1016/j.jss.2012.12.026.
[5] Konstantinos Manikas. “Revisiting software ecosystems research: A longitudinal literature study”. In: Journal of Systems and Software 117 (July 2016), pp. 84–103. ISSN: 0164-1212. DOI: 10.1016/j.jss.2016.02.003.
[6] Slinger Jansen, Anthony Finkelstein, and Sjaak Brinkkemper. “A sense of community: A research agenda for software ecosystems”. In: 31st International Conference on Software Engineering — Companion Volume, ICSE-Companion 2009. IEEE, May 2009, pp. 187–190. DOI: 10.1109/ICSE-COMPANION.2009.5070978.
[7] Rahul C. Basole. “Visualization of interfirm relations in a converging mobile ecosystem”. In: *Journal of Information Technology* 24.2 (2009), pp. 144–159.

[8] Rahul C. Basole and Jürgen Karla. “On the Evolution of Mobile Platform Ecosystem Structure and Strategy”. English. In: *Business & Information Systems Engineering* 3 (5 2011), pp. 313–322. DOI: 10.1007/s12599-011-0174-4.

[9] Geir K. Hanssen. “A longitudinal case study of an emerging software ecosystem: Implications for practice and theory”. In: *Journal of Systems and Software* 85.7 (July 2012), pp. 1455–1466. ISSN: 0164-1212. DOI: 10.1016/j.jss.2011.04.020.

[10] D. German, B. Adams, and A. Hassan. “The Evolution of the R Software Ecosystem”. In: *17th European Conference on Software Maintenance and Reengineering*. Mar. 2013, pp. 243–252. DOI: 10.1109/CSMR.2013.33.

[11] G. Bavota et al. “The Evolution of Project Inter-dependencies in a Software Ecosystem: The Case of Apache”. In: *2013 IEEE International Conference on Software Maintenance*. Sept. 2013, pp. 280–289. DOI: 10.1109/ICSM.2013.39.

[12] Liguo Yu, Srini Ramaswamy, and John Bush. “Software Evolvability: An Ecosystem Point of View”. In: *Third International IEEE Workshop on Software Evolvability*. IEEE, 2007, pp. 75–80. DOI: 10.1109/SE.2007.8.

[13] Liguo Yu, Srini Ramaswamy, and John Bush. “Symbiosis and Software Evolvability”. In: *IT Professional* 10.4 (July 2008), pp. 56–62. ISSN: 1520-9202. DOI: 10.1109/MITP.2008.94.

[14] Walt Scacchi and Thomas A. Alspaugh. “Understanding the role of licenses and evolution in open architecture software ecosystems”. In: *Journal of Systems and Software* 85.7 (July 2012), pp. 1479–1494. ISSN: 0164-1212. DOI: 10.1016/j.jss.2012.03.033.

[15] Michael Armbrust et al. “A view of cloud computing”. In: *Communications of the ACM* 53.4 (2010), pp. 50–58.

[16] Jose Teixeira, Gregorio Robles, and Jesús M. González-Barahona. “Lessons learned from applying social network analysis on an industrial Free/Libre/Open Source Software ecosystem”. en. In: *Journal of Internet Services and Applications* 6.1 (July 2015), p. 14. ISSN: 1867-4828. DOI: 10.1186/s13174-015-0028-2. URL: http://www.jisajournal.com/content/6/1/14.

[17] Robert K Yin. *Applications of case study research*. London, UK: Sage, 2011.

[18] Stanley Wasserman and Katherine Faust. *Social Network Analysis: Methods and Applications*. Vol. 8. Cambridge, UK: Cambridge University Press, 1994. ISBN: 0521387078.

[19] Jose Teixeira, Salman Mian, and Ulla Hytti. “Cooperation among competitors in the open-source arena: The case of OpenStack”. In: *Proceedings of the International Conference on Information Systems (ICIS 2016)*. Association for Information Systems, 2016.
[20] Peter J. Carrington. “Social Network Research”. In: *Mixed methods social networks research: Design and applications*. Vol. 36. Cambridge University Press, 2014. Chap. 2.

[21] Brian Uzzi. “Social structure and competition in interfirm networks: The paradox of embeddedness”. In: *Administrative science quarterly* (1997), pp. 35–67.

[22] Werner H Hoffmann. “How to manage a portfolio of alliances”. In: *Long Range Planning* 38.2 (2005), pp. 121–143.

[23] Jean-Charles Rochet and Jean Tirole. “Cooperation among competitors: Some economics of payment card associations”. In: *Rand Journal of economics* (2002), pp. 549–570.

[24] Raymond F. Zammuto et al. “Information Technology and the Changing Fabric of Organization”. In: *Organization Science* 18.5 (2007), pp. 749–762. ISSN: 10477039, 15265455. URL: [http://www.jstor.org/stable/25146136](http://www.jstor.org/stable/25146136).

[25] Brendan Guilden. “App stores: a digital no man’s land or innovation’s bane?” In: *Journal of Intellectual Property Law & Practice* 11.6 (2016), pp. 445–449.

[26] Vardit Landsman and Stefan Stremersch. “Multi-homing in Two-sided Markets: An empirical inquiry in the video game console industry”. In: *Journal of Marketing* 75.6 (2011), pp. 39–54. DOI: [10.1509/jm.09.0199](http://doi.org/10.1509/jm.09.0199).

[27] Ulrich Wassmer, Pierre Dussauge, and Marcel Planellas. “How to manage alliances better than one at a time”. In: *MIT Sloan Management Review* 51.3 (2010), p. 77.

[28] Jean Charles Rochet and Jean Tirole. “Platform competition in two-sided markets”. In: *Journal of the European Economic Association* 1.4 (June 2003), pp. 990–1029. DOI: [10.1162/154247603322493212](http://doi.org/10.1162/154247603322493212).

[29] Sami Hyrynsalmi et al. “App Store, Marketplace, Play! An Analysis of Multi-Homing in Mobile Software Ecosystems”. In: *Proceedings of the Fourth International Workshop on Software Ecosystems*. Ed. by Slinger Jansen, Jan Bosch, and Carina F. Alves. Vol. 879. CEUR Workshop Proceedings. MIT Sloan School of Management, Cambridge, MA, USA: CEUR-WS, June 2012, pp. 55–68. URL: [http://ceur-ws.org/Vol-879/paper5.pdf](http://ceur-ws.org/Vol-879/paper5.pdf).

[30] Sami Hyrynsalmi et al. “Multi-homing in ecosystems and firm performance: Does it improve software companies’ ROA?” In: *International Workshop on Software Ecosystems 2016, Proceedings*. 2016, pp. 1–14.

[31] Sami Hyrynsalmi. “Letters from the War of Ecosystems — An Analysis of Independent Software Vendors in Mobile Application Marketplaces”. TUCS Dissertations No 188. Doctoral dissertation. Turku, Finland: University of Turku, Dec. 2014. DOI: [10.13140/2.1.4076.4484](http://doi.org/10.13140/2.1.4076.4484). URL: [http://urn.fi/URN:ISBN:978-952-12-3144-5](http://urn.fi/URN:ISBN:978-952-12-3144-5).

[32] Rahul C Basole et al. “Understanding business ecosystem dynamics: a data-driven approach”. In: *ACM Transactions on Management Information Systems (TMIS)* 6.2 (2015), p. 6.
[33] Jose Teixeira. “Open-coopetition in the cloud computing Industry: the OpenStack Nova case”. In: Proceedings of the 1st European Social Networks Conference (EUSN 2014). European Social Networks Conference. INSNA, July 2014.

[34] Jose Teixeira. “Understanding coopetition in the open-source arena: The cases of WebKit and OpenStack”. In: Proceedings of the 10th International Symposium on Open Collaboration (OpenSym 2014). ACM, Aug. 2014.

[35] Jose Teixeira. “Developing a cloud computing platform for Big Data: The OpenStack Nova case”. In: Proceedings of the 2nd International Conference on Big Data (IEEE BigData 2014). IEEE, 2014.

[36] Open-coopetition in a cloud infrastructure for big-data: Exploring the dichotomy of startups and giants in the development of OpenStack. Oulu, Finland: Springer, 2015.