Technology Management, Economics, and Policy Discussion Paper Series

**SEOUL NATIONAL UNIVERSITY**
College of Engineering
Technology Management, Economics, and Policy Program

# Role of Platform Providers in Software Ecosystems

**Kibae Kim, Jörn Altmann, Sodam Baek**

**TEMEP Discussion Paper No. 2015:120**

서 울 대 학 교
공 과 대 학, 기술경영경제정책 대학원과정
151-742 서울시 관악구 관악로 599

Technology Management, Economics, and Policy Program
College of Engineering, Seoul National University
599 Gwanak-Ro, Gwanak-Gu, Seoul 151-742, South-Korea
Phone: ++82-2-880-9140, Fax: ++82-2-880-8389

The *Technology Management, Economics, and Policy Program (TEMEP)* is a graduate program at Seoul National University. It includes both M.Sc. and Ph.D. programs, which are integrated into the graduate programs at the College of Engineering.

The *TEMEP Discussion Paper Series* is intended to serve as an outlet for publishing research about theoretical, methodological, and applied aspects of industrial economics, especially those related to the institute's areas of specialization, namely management of technology, information and communication technology, telecommunication, services, health industries, energy industries, as well as infrastructures for development. In particular, paper submissions are welcome, which analyze current technology and industry related issues, discuss their implications, and suggest possible alternative policies. The objective is to gain insights into important policy issues and to acquire a balanced viewpoint of policy making, technology management, and economics. It will enable us to identify the problems in industries accurately and to come up with optimal and effective guidelines. In addition, another important aim of this series is to facilitate communication with external research institutes, individual researchers, and policy makers worldwide.

Research results disseminated by TEMEP may include views on policies for information and communication technologies, technology management, telecommunication, energy, health, and development economics, but the institute itself takes no institutional policy position. If appropriate, the policy views of the institute are disseminated in separate policy briefs. Thus, any opinion expressed in the TEMEP Discussion Paper Series is those of the author(s) and not necessarily the opinion of the institute.

Finally, the current editor of this series would like to thank Prof. Dr. Almas Heshmati for establishing this working paper series in January 2009 and for guiding its development during the course of the first 55 issues. In particular, Prof. Heshmati provided invaluable contributions for setting up the local IT infrastructure, registering the series with RePEc, disseminating the working papers, and setting the quality requirements.

Jörn Altmann, Editor

Office: 37-305
Technology Management, Economics, and Policy Program
College of Engineering
Seoul National University
599 Gwanak-Ro, Gwanak-Gu
Seoul 151-742
South-Korea

Phone: +82-70-7678-6676
Fax: +1-501-641-5384
E-mail: jorn.altmann@acm.org

# Role of Platform Providers in Software Ecosystems

Kibae Kim, Jörn Altmann, Sodam Baek

Technology Management, Economics, and Policy Program
Department of Industrial Engineering
College of Engineering
Seoul National University
Seoul, South-Korea
kibae36@snu.ac.kr, jorn.altmann@acm.org, bsodam@gmail.com

January 2014

**Abstract:** Through the advancement of information technology, a new type of innovation could emerge. This innovation type suggests a leading company to open up its software service platform to customers as well as competitors. Many innovation studies have been performed in the past, in order to understand the structure of this interaction among platform users from a network science perspective. By focusing on the internal mechanisms of software vendors only, these studies missed the role of platform providers in the innovation process though. In this paper, we fill this gap by investigating the impact of platform providers on the structure of a software service network. The empirical data about the software services network has been gathered from Salesforce.com AppExchange. It is used to identify the clusters in the network and the network position of each software service. Using the Kruskal-Wallis test, we infer from those results that the network position of the platform provider's software services is different from the network position of software services of third party vendors. In particular, our results show that the software services released by the platform provider locate at hub positions, while those released by third party vendors locate at positions interconnecting clusters. These results imply a role division between platform providers and third party vendors in this innovation type. The former leads the overall innovation on the platform, while the third party vendors interconnect software services of different categories.

**Keywords:** Open Innovation, Software Ecosystem, Platform-as-a-Service, Software-as-a-Service, Salesforce.com, Platform Leadership, Network Analysis, Kruskal-Wallis Test.

**JEL Classification Numbers:** A12, C12, D85, L86, M15, O32, O35.

## 1. Introduction

One of the most important features of todays software innovation is that software vendors open up their platforms to attract end-users to their innovation environment. By doing so, they benefit from the collective intelligence of the platform users [1, 2]. Collective intelligence is a pool of innovation agents voluntarily engaged in the cooperation with each other for innovation in an open environment, in which the platform provider and the innovation agents share their knowledge and innovation resources [1, 3]. With collective intelligence applied to an open platform, ideas can be reused to create new ideas, which are unable to emerge otherwise. For example, a single software vendor could not perform it due to the limitation of development cost [4].

Since software users interact with each other on an open platform and their interaction leads to innovation, a lot of studies investigated the structure of service networks representing their interactions in a variety of contexts [5, 6, 7]. A research defined a service network as an extension of a value chain model between service providers and users [5]. Henneberg et al. [6] defined a service network, in which new services link existing services through their combination. Previous innovation studies of service networks also investigated the structure and evolutionary pattern of networks [8, 9, 10], and the relationship between network characteristics and innovation performance [11, 12].

In this prior research on service networks, the assumption is hidden that a service network evolves only by the interaction among agents who participate in the innovation through collective intelligence. This assumption is the fundamental idea in network science that a complex network evolves by a simple rule correlating with the interaction among the nodes such as the preferential attachment and random rewiring models [13, 14]. However, the assumption in this prior research is very far from the reality of innovation on platforms. It is common in business that a platform provider strives for motivating the collective intelligence to innovate [15]. For example, Google provides its users with software development tools and an infrastructure to build a community of users, attracting users into its Android ecosystem. In conclusion, previous research needs to be extended to understand the contributions of a platform provider to service networks that evolve through collective intelligence.

The objective of this research is to close this gap between the assumption of previous studies and the real business on the role of platform provider in a software ecosystem. To do so we investigate the innovation in a software service platform, with respect to the structure of a software service network and to identify clusters in the network and the network position of software services in those clusters. A service network is defined as a tuple of a set of nodes and a set of links. While nodes represent software services of the service network, links represent the relationships between software services. For our investigation, we analyze the difference between the network positions of software services released by the platform provider and those by third party vendors. Based on these analysis result, we discuss the effort of the platform provider in stimulating its users (i.e., as a source of collective intelligence) to participate in the innovation. With this discussion, we answer the following two research questions related to the contribution of platform providers to the evolution of software service networks: Where are the services of a platform provider positioned in the software service network? Is the position of software services released by a platform provider different from the location of the software services that have been released by third party providers?

In order to achieve the research objective and answer the research questions, network analysis has been conducted, using empirical data gathered from Force.com, an open platform published by Salesfroce.com (http://www.salesforce.com). It is a customer relationship management system. By analyzing the empirical data gathered from the open platform, we explore what a platform provider endeavors for promoting the growth of its software innovation ecosystem evolving its platform. We identify the clusters through the clustering algorithm on the basis of link similarity [16]. Then, we measure the degree centrality of services within clusters and in the entire network, and the connectivity of a node in a cluster with the nodes outside the cluster to identify the position of nodes over clusters [17]. Finally, we compare the network position of services released by the platform provider with the one released by the third party providers, using the Kruskal-Wallis test, a nonparametric analysis of variance.

Our main findings show that the positions of software services released by the platform provider are different from the positions of software services released by the third party vendors. The degree centrality of software services released by the platform provider in the entire network is significantly higher than the one of the third party vendors. That is, the software services of the platform provider locate at the center of the entire network comparing to the third party vendors. Addition to this, a software service released by the third party is connected with more software services outside its cluster than the software services released by the platform provider. It means that the software services of the third party vendor are likely to be a bridge compared to one of the platform provider. In summary, the results suggest that the platform provider develops the software services leading the innovation in the entire service network, while the third party vendors release the software services at the niche among the software service clusters.

The remainder of this paper is structured as follows. Section 2 describes the conceptual background of network analysis. The collection of the empirical data and the process of analysis are described in Section 3. Section 4 and Section 5 lay out the analysis results and a detailed discussion of the analysis results, respectively. Section 6 concludes this paper with a summary and outlook.

## 2. Theoretical Background

### 2.1. Software Service Platform as a Service

Through the advancement of IT technology, an IT service user can utilize a computing service provided by a software vendor over the Internet. This type of computing delivery is called cloud computing [18, 19]. A cloud computing provider achieves economy of scale by operating a large data center [20]. And, with cloud computing, a software user can reduce the cost of computing by renting the data storage, computation power, and network capacity from the cloud provider [21]. Moreover, a user is able to aggregate existing services to create new services provided by several cloud computing providers [22, 23].

Cloud computing can be classified into three types [24]: Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), and Infrastructure-as-a-Service (IaaS). In detail, in the PaaS model, a cloud computing vendor provides a platform and environment for software development. The platform is represented on a browser and the complexity of the virtualized infrastructure is hidden. Using this platform, the platform users can concentrate on developing and commercializing their software services [25, 26]. By using the online resource furnished by the platform provider, platform users reduce the costs of software development and system maintenance. And a platform provider gains its profit from the activity of the platform users [27].

Salesforce.com, which is a SaaS provider and PaaS provider, is a company that provides a Customer Relationship Management (CRM) system. It provides its applications as SaaS services as well as a platform for software service development and customization [28]. In other words, Salesforce.coms users participate in the innovation and utilize the output of the innovation from the open platform provided by Salesforce.com. Therefore, Salesforce.com and its users form a service ecosystem ("software ecosystem" as defined in [29], or "digital ecosystem" as defined in [30]). In this service ecosystem, service innovation is performed by cooperation and interaction among users and the platform provider.

Figure 1 illustrates the software service ecosystem of Salesforce.com. The ecosystem comprises a platform provider (Salesforce.com) and users. A user can further be categorized as a software service vendor (i.e., a user that develops software services) and a customer (i.e., a user that consumes software services created by software service vendors). A user can be both a software service vendor and a customer. Customers are charged based on the usage of the software service platform. Vendors can offer their software services to users free-of-charge or for a fee on the platform. A vendor can also utilize the software services created by other vendors. If a vendor charges a fee to customers, Salesforce.com requests a security check fee from the vendor. Salesforce.com also creates software services, using the name Salesforce Lab (SFL).

### 2.2. Platform-Related Innovation

The innovation in the service ecosystem of Salesforce.com is an example of "open innovation", a new trend of innovation identified in the early 2000s [3]. In open innovation, a company, which wants innovation, invites third-party companies in its innovation process to aggregate the innovation resources [3]. Contrarily to the classical innovation process, in which a company invests in and commercializes a product, third-party vendors on software service platforms utilize the resources published by the platform provider for their innovation. Leading companies in information technology (IT) already apply this open innovation process for achieving their innovation. For example, while Intel continues focuses on developing and up-

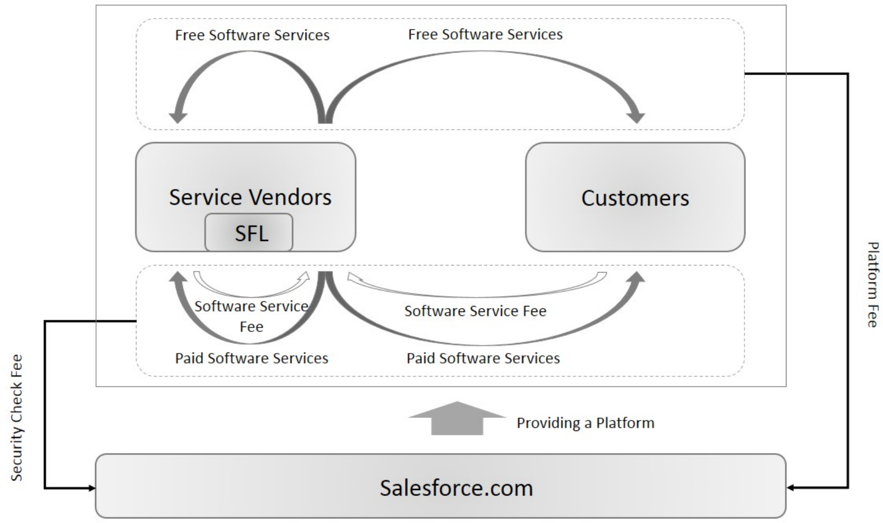


**Figure 1. The Software Service Ecosystem of Salesforce.com AppExchange**

grading CPUs, Intel also opened up its interface to their hardware so that third party innovators can get access to develop supporting technology quickly [15].

The success of open innovation for a platform relies on how vigorous the third party vendors participate in the innovation [31]. The platform provider opens the access to its innovation resources so that users can utilize it for free (i.e., can reuse the resources for innovation and collaboration) [15]. In this way, the platform provider harnesses the collective intelligence of third parties, to achieve its innovation with low cost [1]. On the other hand, the third-party vendors might collaborate with each other without any benefit to the platform provider [30]. To address this, some platform providers separate their platform into a commercial version and a free-of-charge version. This allows them to gain benefit from the commercial version without limiting the innovation through collective intelligence [19].

It is notable that the prior research divides the roles of the stakeholders in the open innovation process. That is, a platform provider provides the platform, on which the third party vendors work, while the third party vendors achieve innovation on the platform provided by the platform provider. However, the division of role is difficult in the software services market. In the software services market, a platform provider can also participate in the open innovation through the development of software services. For example, Salesforce.com, which owns the software services platform, also participates as a vendor in the open innovation (Figure 1).

The platform provider has two incentives to serve as a software service vendor in its platform. First, the platform provider can generate revenue through the software service that it develops on the platform. Second, the platform provider needs to create a critical mass of software services and customers, in order to incentivize third party vendors to participate in the open innovation platform. Without this critical mass, third party vendors cannot generate sufficient revenue to cover their development cost through their innovation [32]. Therefore, the platform provider releases useful software services on the platform to attract customers and third party vendors. The question that rises is whether the output of the platform provider is identical to or different from the third party vendors.

### 2.3. Position of Nodes in a Network

Innovation studies indicated that the network position could affect the innovation performance, if the network structure is inhomogeneous [33]. The most popular techniques of measuring network position is using centralities, e.g., degree centrality [34]. Centralities measures determine how central a node is embedded in a network. They imply the strength (e.g., social power, influence, advantage in information flow) of each node in the network according to its own connectivity (node degree).

The network position of a node depends on the attribute of the node. For example, looking at the pattern of interlocked directorates among the top 50 ecommerce firms, it can be identified that their network position is different from the network position of other dominant firms [35]. Moreover, the network position of a node affects the innovation performance of the node. The links connecting nodes with low degree centrality play an important role in innovation and are called weak ties [11]. The effect of network position is also dependent on the context of innovation. For example, a central position enables a high performance, if the innovator has good capacity for absorbing innovation resources [12, 14].

Some previous studies examined the software service ecosystem from a network perspective [36, 37]. Defining a node and a link to represent a software service and a new service co-development, respectively, the authors described the topology of an entire software service network and traced the change of positions of nodes in the network. Their results show that software service networks show a power-law topology like other real complex networks [38], and that the topology is invariant as time goes on [37]. However, the position of a node changes within the topology [36].

For the research presented here, we consider the effect of node attributes on network positions, which has not been considered in this area of innovation research before. The nodes of the software service network are distinguished according to their ownership. One of the two groups of nodes consists of the software services that have been released by the platform provider. The other group comprises software services that have been released by third party vendors.

As a platform provider is familiar with its own platform and can easily gather information on the innovation trend within its platform, the platform provider can create software services of high value for customers easier than third party vendors. As a result, software services released by the platform provider might be used by customers more often than those developed by third party vendors. Consequently, it is expected that the software services developed by platform providers are probably located more central than those of third party vendors. Based on this, we establish the first hypothesis:

**H 1.** *The software services released by the platform provider locate at more central position in a software service network than those released by third party vendors.*

### 2.4. Network Position within a Cluster

Complex network science has shown that some nodes in a network are connected more densely than other nodes. These groups of densely connected nodes are called clusters [14, 39, 40]. The process of forming clusters is affected by the preferences of people for similarity. Examples of clusters can be found in networks about the physical distance between houses [41], academic background of researchers [39], and demographic properties (e.g., gender and race) of people [42, 43].

Innovation studies focus on the process through which knowledge is created within and between communities [44, 45, 46]. Knowledge is not only constructed within a community (cluster), in which agents share their academic background, norm, perspective, and methodology [47] but also between clusters. For example, innovation is achieved, if a community with a problem hard to solve finds the solution with the knowledge from another community [45]. In this process, an agent with low connectivity within a community but reach to another community has the chance to obtain the knowledge and disseminate it to its own cluster [11, 44]. This agent locates at a position that mediates several communities (clusters) in a network. It is called structural hole according to its feature of the position in the network structure [44, 48].

Similar to those social networks, software services tend to be connected with other software services that share identical attributes [49]. The software services within the same service category are more likely to be combined than those in different service categories. Furthermore, software services are likely to be connected with those software services released by the same vendor. According to Kim et al. [49], we assume that a software service network also contains several clusters, in which software services are closer than other software services. Innovation is mainly achieved within the clusters of the software service network, as the innovation does in a social network. If a platform provider intends to promote third party vendors to innovate on its platform, one of the best strategies is seed the software services that build their own communities. In this case, the software services released by the platform provider would be at the center of the clusters. Therefore, we establish the following hypothesis:

**H 2.** *The software services released by the platform provider locate at more central position within a cluster than those released by the third party vendors.*

A network evolves as clusters emerge and decline. That is, an entire network survives in a changing environment, while old clusters disappear and new clusters appear [46]. The survival of an entire network is also dependent on the interaction between the clusters. An empirical study shows that the probability of an organizations survival increases as subgroups in the organization interacts with each other [50]. This correlates with the innovation process. Innovation is normally achieved within clusters, but a solution to a hard problem of a cluster could be found in another cluster [45]. The interaction between clusters creates more innovation, resulting in a high probability of survival [50].

An issue to the interaction between clusters is to identify the nodes that mediate. It is impossible for a platform provider to catch each innovation trend in detail and to invest in all potential services. This is why the platform provider invites third party vendors in its innovation environment [3, 15]. In this open environment, the platform leaders strategy is to develop core software services running on the platform and to allow third party vendors to develop software services complementary to its core services [15, 31]. In this role, third party vendors release these new complementary software services, which could also mediate between several clusters designed by the platform provider. This could lead to novel innovation based on the existing software service, fitting to the market demand. Based on this discussion, we suggest the following hypothesis:

**H 3.** *The software services released by third party vendors are connected with more nodes outside their clusters than those software services released by the platform provider.*

## 3. Methodology

### 3.1. Data Collection

The data for the empirical analysis was collected from AppExchange, which is the open software service platform of Salesforce.com (http://appexchange.salesforce.com) for buying and selling software services. A customer can purchase and run services that are needed. A vendor uploads and offers its software services to customers. The marketplace lists software services offered. Each software service contains information about the provider of the service, the software service name, the release date, the pricing plan, the service categories, the reviews, and the rating of the software service. The information that is used for the analysis includes only the software service name, the provider of the service, and the release date. The information about the software service co-installations that is used for the analysis is the information that has been made available on the AppExchange platform by software service vendors. The data comprises the name of the user (i.e., the Salesforce.com account) and the list of services that the user has installed.

Based on the collected data, the software service network has been constructed. The software service network is defined as a tuple of a set of nodes and a set of links. While the nodes represent the software services operating on the Salesforce.com platform, the links represent the software service co-installations. Figure 2 illustrates an example of the software service network. The example shows Salesforce Lab, operated by Salesforce.com (left hand side of Figure 2). It released two services (Software Service 1 and Software Service 2). The third party software service vendors released two more software services (Software Service 3 and Software Service 4). User A installed Software Service 1 and Software Service 2, and User B installed Software Service 2, Software Service 3, and Software Service 4. Then,

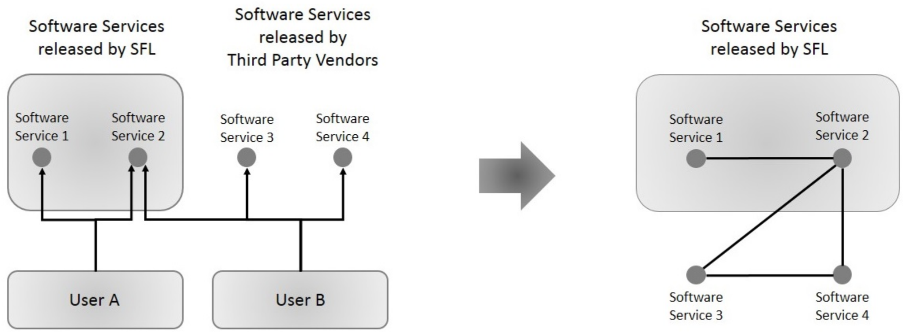


**Figure 2. Example of a Software Service Network at AppExchange of Salesforce.com**

the example leads to a software service network with 4 nodes and 4 links (i.e., links between Software Service 1 and Software Service 2, Software Service 2 and Software Service 3, Software Service 3 and Software Service 4, as well as Software Service 4 and Software Service 2 (right hand side of Figure 2)).

### 3.2. Identifying Clusters in a Network

A cluster is a subset of a network, in which nodes are more similar than with nodes outside the cluster [16, 39, 40, 51]. We use a hierarchical clustering approach on the basis of the similarity of links [16]. This approach resolves the vagueness of the affiliation of a node densely connected with nodes in different clusters.

The process of identifying clusters starts with calculating the similarity between a pair of adjacent links on the basis of the number of their common neighbors. Then, links are agglomerated, if they have a similarity greater than a certain threshold. These two steps are repeated for all pairs of links that have a common node. Afterwards, all links have been assigned to a cluster. As a consequence of using links is that a node could belong to multiple clusters. For finding the optimum threshold, or the threshold that generates clusters, in which nodes are most densely connected, we use the partition density recommended by Ahn et al. (2010). That is, the optimum threshold of similarity is the value that maximizes the partition density.

### 3.3. Measuring the Network Position

Centralities measure the network position of nodes. There are a variety of centralities, including degree centrality [34]. As the degree centrality of a node measures the number of nodes connected with the node, it is most intuitive and simple way to measure how deep the node is embedded in its network. The degree centrality is normalized by the maximum number of links $(g-1)$ that it can possibly have in a network of size $g$. Assuming that $a_{ij}$ represents the link between two nodes $i$ and $j$, then the normalized degree centrality $d_i$ of node $i$ in a network of size $g$ is defined as:

$$d_i = \sum_{j=1}^{g} a_{ij}/(g-1) \tag{1}$$

### 3.4. Nonparametric Method of Sample Comparison

If node $i$ and node $j$ are connected, $a_{ij} = 1$, otherwise $a_{ij} = 0$. If a node is isolated from any other nodes, $d_i = 0$.

Based on this definition of degree centrality, we define the degree centrality within clusters and the external connectivity of nodes within clusters. The degree centrality within a cluster is the degree centrality of the network sub-graph that is determined by the cluster. It is normalized by the maximal possible degree $(c-1)$ for a cluster $C$ of size $c$. Expressing this cluster degree centrality with a mathematical formula, the normalized degree centrality $d_{ic}$ of node $i$ in cluster $C$ can be defined as:

$$d_i^C = \sum_{j \in C} a_{ij}/(c-1) \tag{2}$$

In addition to this, the external connectivity of node $i$ belonging to cluster $C$ measures the number of links to nodes outside the cluster. It is normalized by the maximal possible number of links $(g-c-1)$ for a cluster $C$ with size $c$ in a network of size $g$. In detail, the normalized external connectivity $d_i^{\neg C}$ of node $i$ belonging to cluster $C$ is defined as:

$$d_i^{\neg C} = \sum_{j \in \neg C} a_{ij}/(g-c-1) \tag{3}$$

Analysis of variance (ANOVA) compares multiple samples according to their means and variances for determining whether they are statistically different [52]. It assumes that the distribution of samples are normal (normality), and their variances are identical (homoscedasticity). Therefore, the Kolmogrov-Smirnov test [53] and Levenes test [54] should precede ANOVA for determining the samples normality and homoscedasticity, respectively. In some cases, the samples do not satisfy the normality and homoscedasticity assumptions, and non-parametric methods should be applied. The Kruskal-Wallis test compares the sum of ranks of cases instead of comparing the means and the variances of samples [53]. This test can be used for testing the difference between two independent samples. The null hypothesis of this test is that the two samples are identical and, therefore, the alternative hypothesis is that the two are different with a certain significance probability.

### 3.5. Procedure

For our analysis, we create the software service network from an empirical data set and, then, identify clusters in the network according to Ahn et al. [16] method. We measure three indicators about network position: the degree centrality in the entire network, the degree centrality in the cluster, and the external connectivity to nodes outside the cluster. As the clustering procedure based on Ahn et al. [16] agglomerate similar links, a node could belong to multiple clusters. To deal with this, we calculate the degree centralities in the clusters that the node belongs to as well as the external connectivity for each clusters that the node belongs to. Then, for both measures, we calculate the average. Furthermore, we split the data set into two sets of nodes: a set of nodes, which consists of software service released by the platform provider, and a set of nodes that represents software services released by third party vendors. Using the same three measures as before, we compare the two sets of nodes. If the test of normality and homoscedasticity of the two sets fails, we apply the Kruskal-Wallis test for comparing the two sets.

## 4. Analysis Results

### 4.1. Data Description

The data collected shows that 1000 vendors registered on the platform and 74 of those vendors published their information about their software service installations. The 74 vendors utilized 217 software services. These 217 software services were created by 99 vendors (i.e., they participated in the innovation). Among the 217 software services, 88 software services were released by Salesforce Lab, the brand of Salesforce.com. The remaining 129 software services were released by third party software service vendors. However, the rate of contribution of Salesforce Lab (i.e., the number of software services released compared to the total number of software services) decreased gradually over time. The total number of new software services per month, however, increased steadily during the study period from July 2005 to April 2013.

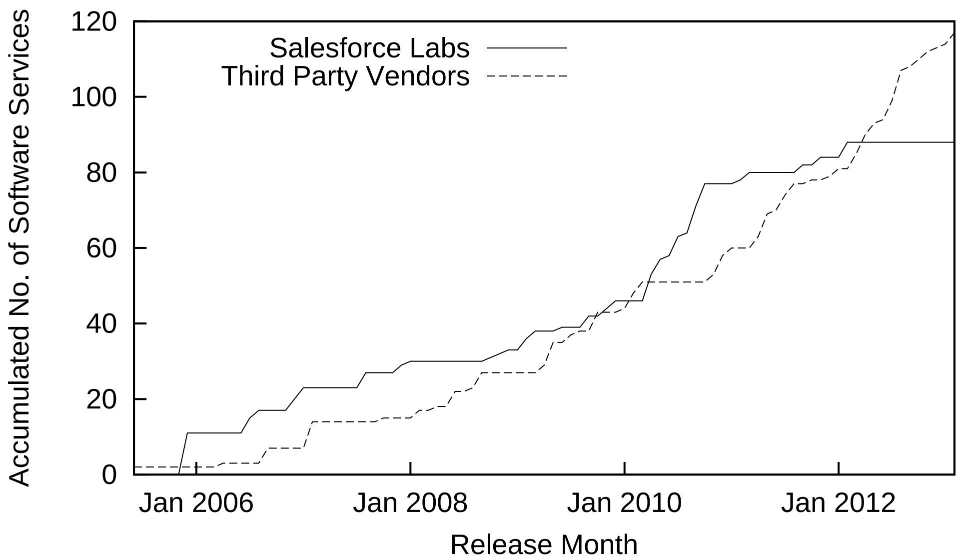


**Figure 3. Growth of the Software Service Network of Salesforce.com AppExchange**

Figure 3 depicts the trend of the number of software service releases of Salesforce

Lab (SFL) and the trend of the number of software service releases of third-party vendors (TPV). The number of software services, which have been released by Salesforce Labs, steadily increased from 11 software services in January 2006 to 77 software services in November 2010, but slowed down since then. At the end of the study period, only 88 software services have been released in total. Compared to this, the increase in the number of software services, which have been released by the third party vendors, has been little during these periods. The number of software services released by third-party vendors was 2 software services in January 2006 and 51 software services in November 2010. Afterwards, it increased faster than the number of software services released by Salesforce Labs. The third party vendors contributions surpassed the number of software services released by Salesforce Labs in May 2012. The number of software services of third-party vendors is 129 at the end of study period.

### 4.2. Network Topology

A software service network is formed with the data described above according to the definition given in the previous section. Figure 4 shows the cumulative distribution of the node degree in log-linear scales. The cumulative degree distribution decays by a linear function, not considering the five outliers, which have a degree centrality that is larger than 55. The results of the OLS regression of the distribution (excluding the five outliers) in log-linear scales depicts a slope of 0.0347. The t-test statistic of the slope is 0.000. The adjusted $R^2$ is 0.993.

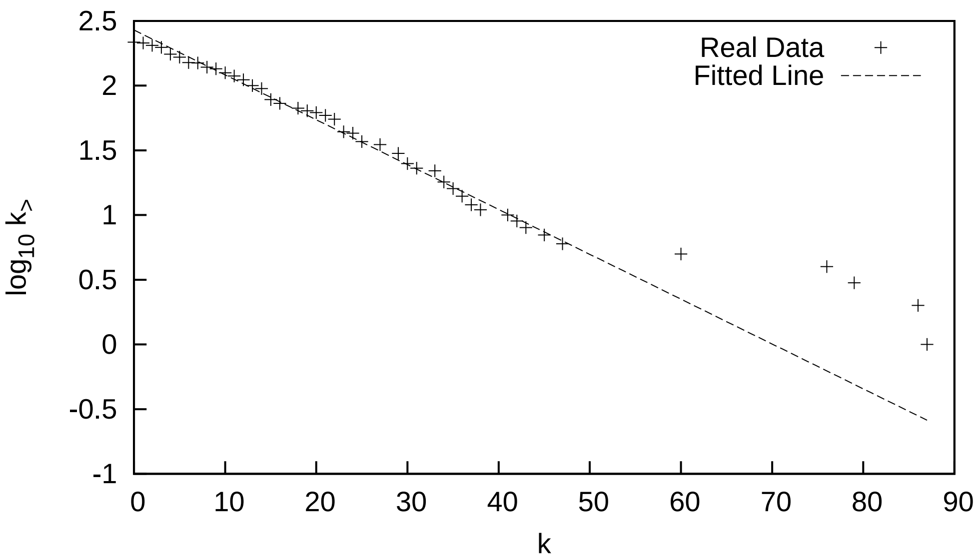


**Figure 4. The Cumulative Degree Distribution in Log-Linear Scales**

The topology results suggest that the degree distribution of the service network fits to an exponential function with higher significance and explanation power than a power function, if the five outliers are ignored. The results are extraordinary compared to the traditional discussion of complex network analysis. Prior research normally distinguishes scale-free networks, which consist of a few hubs connecting all nodes, and random networks, which do not have hubs at all [13, 14]. However, our results show that the degree distribution in the service network looks mostly exponential but involves a few outlier nodes with an extraordinarily large number of links. Our conjecture on this topology is that the rule of evolution of the network except the outliers is different from the one of the outliers. That is, the network expect the outliers evolved by random selection like the model of Watts and Strogatz [14], while there was a positive feedback for the outliers like the preferential attachment model [55].

### 4.3. Clusters in the Software Service Network

Figure 5 shows the partition density of clusters changing according to the threshold of similarity defined in Ahn et al. [16]. The partition density roughly increases from 0.0034 to 0.0201 as the similarity threshold increases from 0.00 to 0.50, and decreases to 0.0137 as the thresholds further increases to 1.00. That is, the partition density peaks at 0.0201, if the threshold is 0.50. The results suggest that the optimum threshold is 0.50 to identify clusters in the software service network. The method of Ahn et al. [16] identifies 327 clusters in the network. The number of cluster is larger than the number of nodes in the network. This is because the clustering method we used agglomerates similar links instead of nodes. The size of a cluster is 4.3 software services in average, and the largest cluster consists of 20 nodes and 19 links. Two hundred and fifty one clusters involve the software services released by Salesforce Labs. The number of software services of Salesforce Labs in a cluster is 1.661 in average. The average of the ratio of the number of software services released by Salesforce Labs in a cluster to the cluster size is 0.400. That is, 40% of software services are released by SalesForce Labs. Because the algorithm of Ahn et al. [16] identifies the clusters on the basis of links, a node could be involved in multiple clusters.

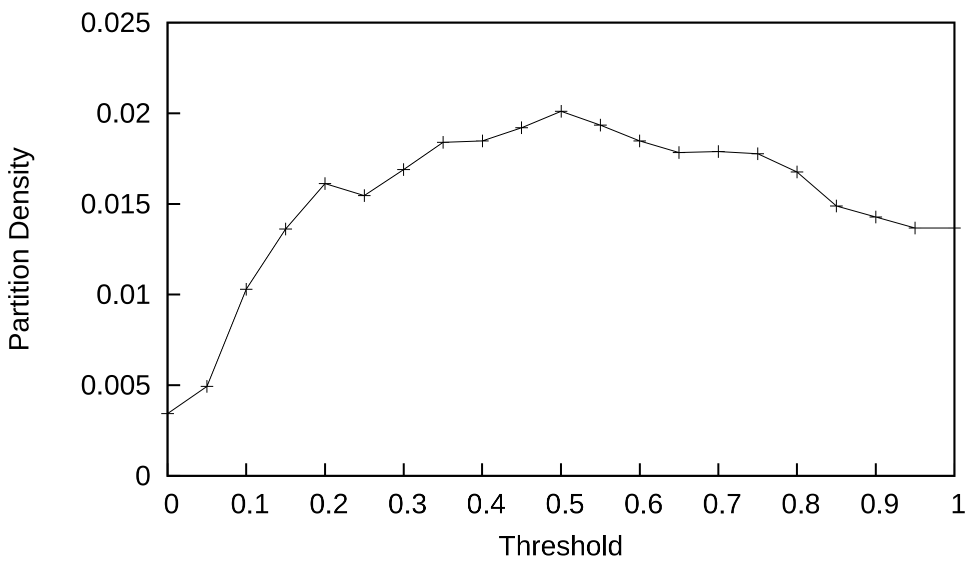


**Figure 5. Partition Density for Each Threshold of Similarity**

### 4.4. Hypothesis Test Results

The network position of software services released by the platform provider (group SFL) is compared with those of the third party vendors (group TPV). The cumulative degree distribution is linear in log-linear scale (Figure 4), and it means that it is highly skewed in linear-linear scale. Therefore, we use the logarithm of the normalized indicator in the test. The degree centrality in a cluster and the external connectivity are also normalized and transformed by a logarithm function due to the same reason.

Table 1 presents the summary of Kolmogrov-Smirnov test results [53] and Levenes test results [54]. The Kolmogrov-Smirnov test results show that the significance probability of two indicators are less than 0.05 for both groups SFL and TPV, and the degree centrality in the entire network is less than 0.10. These mean that the distributions of indicators do not follow a normal distribution. The Levenes test results show that the significance probabilities of the indicators are all larger than 0.10. These results mean that the distributions of indicators for group SFL are not different from those for group TPV, therefore fulfilling the homoscedasticity assumption of ANOVA. Although the homoscedasticity assumption works, as the normality assumption is not given, the parametric ANOVA is not recommended for our inference.

**Table 1. Kolmogorov-Smirnov Test Results**

| Variable | Group | Kolmogorov-Smirnov Statistic[a] | Levene Statistic[b] |
|---|---|---|---|
| $\log d_i$ | SFL | $0.127^{*}$ | 0.414 |
| | TPV | $0.093^{*}$ | |
| $\log d_i^C$ | SFL | $0.207^{***}$ | 0.173 |
| | TPV | $0.178^{***}$ | |
| $\log d_i^{\neg C}$ | SFL | $0.281^{***}$ | 0.118 |
| | TPV | $0.193^{***}$ | |

[a] Degree of freedom is 73 for SFL and 139 for TPV.
[b] Degree of freedom 1 and degree of freedom 2 are 1 and 210, respectively.
$^{*}$ significant at 10% level
$^{***}$ significant at 1% level

Therefore, we use the Kruskal-Wallis Test for testing the difference in the network position of software services between the platform provider (SFL) and the third party vendors (TPV). The dependent variables in the analysis are the logarithms of the normalized degree centrality in the entire network, the logarithms of the normalized degree centrality in a cluster, and logarithms of the normalized external connectivity out of a cluster.

Table 2 shows the analysis results for the three dependent variables. First, the degree centrality in an entire network of software services belonging to group SFL is significantly larger than those to group TPV. The average rank of the group SFL is 118.56, while the average rank of group TPV is 100.17. Their difference is statistically significant. It is smaller than 5%. This means that the network position of software services released by the platform provider is more central in the entire network than the one by the third party vendors. That is, Hypothesis 1 can be accepted.

**Table 2. Kruskal-Wallis Test Results**

| Variable | Group | Average Rank | Kruskal-Wallis Statistic[a] |
|---|---|---|---|
| $\log d_i$ | SFL | 118.56 | 4.317$^{**}$ |
| | TPV | 100.17 | |
| $\log d_i^C$ | SFL | 98.05 | 2.111 |
| | TPV | 110.94 | |
| $\log d_i^{\neg C}$ | SFL | 94.58 | 0.040$^{**}$ |
| | TPV | 112.76 | |

[a] The statistic follows $\chi^2$ distribution.
$^{**}$ significant at 5% level

Second, the degree centrality belonging to group TPV is significantly larger than those to group SFL. The difference in the degree centrality in a cluster is not statistically significant between groups SFL and TPV in 10% of significance. Its significance probability is only 0.146. Therefore, Hypothesis 2 is not accepted. On the other hand, the averaged ranks of the external connectivity to nodes outside of clusters in group SFL and TPV are 94.58 and 112.76, respectively. Their significance probability is smaller than 5%. This means that the external connectivity of group TPV is significantly higher than the one of group SFL. That is, Hypothesis 3 is accepted.

## 5. Discussion

### 5.1. Releasing the Role of Platform Provider

While previous theoretical and empirical studies on open innovation concentrate on opening innovation resources and on the environment of the platform leader and users, we emphasized that the role of platform provider comprises more than providing the platform.

From the traditional view of open innovation, a platform leader could achieve sustainability, if it develops an innovation ecosystem through opening up its innovation resources to third party agents, inviting them to its innovation process [3, 15]. In this model, a platform provider allows its platform users to utilize its innovation resources, and the users develop goods complementary to the providers goods. Moreover, the platform provider can increase the quality of its goods effectively by concentrating on the innovation of its core technologies and leaving the complementary goods development to platform users. Then, the quality and complementarity of the group of goods on the platform attract more customers than when they are provided separately [15]. The key to this mechanism of open innovation is making profit as well as attracting users. Therefore, the platform provider divides its platform into a commercial version and free-of-charge version, which is the strategy called two-sided market [19, 32].

While management science is interested in the strategy of a platform leader for open innovation, network science has investigated the mechanism how the platform users self-organize a software ecosystem. It has been discussed that the evolution of the network follows an internal mechanism (e.g., preferential attachment mechanism or random rewiring [14, 55]. As a consequence of these mechanisms, it must be assumed that a platform provider does not contribute to the network evolution besides providing and managing the platform. The networks resulting from these mechanisms have a complex topology. For example, real networks self-organized through collective intelligence are scale-free as we can see in the co-authorship network (Wagner and Leydesdorff, 2005), WWW [13] and the Internet [56]. Besides, real networks contain clusters, in which nodes are densely connected (Newman, 2001; 39, 51]. It means that agents belonging to the same cluster normally interact more frequently than agents belonging to different clusters. Therefore, significant innovation emerges, if an agent contacts another one in another cluster. In this case, new ideas can flow between different communities [11].

What is interesting in the previous studies is that the research on the platform provider side and the one on the platform user side are not directly connected. No research considered that a stakeholder can take roles on both sides. In management science, platform users are only objects that move according to the strategy of the platform provider. In network science, a platform is already given, on which users play, leaving no room for the provider to interfere. However, the results of our research show that the platform provider also contributes to the network evolution and the initial setting of the software service network. Their contributions motivate third-party vendors to participate in the innovation through the platform. As we can see from our analysis results, although platform providers position their software services at central positions, leading the software ecosystem, they also enables third party vendors to bring innovation through the establishment of links to other software services of other clusters within the software service network.

## 5.2. Making a Software Service Ecosystem Successful

The results discussed are also relevant for software service platform providers from a managerial perspective. In the current models of software innovation through collective intelligence, the role of a platform provider is separated from the roles of third-party vendors. According to the existing models, third-party vendors share their innovation resources, reuse, and recombine them for innovation, while the platform provider simply prepares the environment for the interaction between users. That means, the ecosystem according to existing open innovation theory allows third-party vendors access to core technology of platform providers [15]. The platform providers simply prepare the environment, not necessarily leading to successful innovation through third-party vendors. The platform providers are not considered to take on an active role in the innovation process.

The key to the success of a software ecosystem in a platform is gathering the platform users, i.e. software vendors and software customers [19]. Software customers gather on a platform, if the quality of software services and the variety of complementary services they use on the platform are good. The software vendors come to the platform if there are a critical mass of customers, guaranteeing revenue that cover their development cost [32]. It is based on the indirect network effect (i.e., the profit of software vendors and the utility of software customers [57]. A software ecosystem remains sustainable if a critical mass of users participates. It will fail, if the platform does not find the critical mass of users. It is hard to motivate software vendors, especially in the initial period, because of the lack of existing software services on the platform. The platform does not reduce the innovation cost as open innovation promised [3], and the insufficient amount of software customers does not guarantee the revenue for the third party vendors. However, as this is critical to the success of the platform strategy, it is recommended that the platform provider participates in the innovation process initially in order to attract a critical mass of platform users. This can include the provisioning of some initial software services that have been developed by the platform provider.

Our empirical analysis results indicate that the software service platform provider participated in the innovation process during the early time period of the platform. It provided incentives to platform customers to join and, in turn, through the existence of platform customers, to make the platform attractive to third-party vendors. The results show that the platform provider released more software services than the third party vendors in the beginning (Figure 3). Besides, the software services released by the platform provider are central in the entire network, while those released by the third party vendors connect clusters in the network (Table 2).

In conclusion, the participation of the platform provider is more active than we expected. The platform provider does not just provide the platform, on which third party vendors can offer their services, but also leads the innovation trend in the software ecosystem. With existing software services and customers gathered through the effort of the platform provider, the software vendors achieve innovation through releasing software services complementary to the existing software services.

## 6. Conclusion

In this research, we compared the position of software services of a platform provider in a software service network with the position of the third party vendors. In the analysis, we used the empirical data gathered from Salesforce.com AppExchange marketplace. The analysis results show that the software services released by the platform provider locate at the central position in the entire network, while the software services released by the third party vendors are linked with the software services outside the clusters involving them. There is no difference in the position within clusters between the two providers. The central position in the entire network means that the software services at this position lead the entire software ecosystem. A software service goes to the central position as more customers use the software service. A software service linked through external links to other cluster means that it provides bridges between clusters, allowing new knowledge to flow into the clusters. In summary, out analysis results show that the platform provider leads the software ecosystem and the third party vendors provide complementary goods in the software ecosystem as prior research on platform leadership stated [15].

The novelty of our findings is to demonstrate that the platform provider does not just provide a platform, on which third party vendors can perform innovation but also participate in the innovation process itself. Moreover, the role of the platform provider is to lead the ecosystem, in order to motivate third party vendors to innovate through the provisioning of complementary goods. This is a fact that prior research has missed, and what decision makers are likely to miss.

Without considering the participation of platform providers in the innovation as well as providing a platform, a decision maker may design a wrong strategy, or provides a platform without any complementary goods not motivating the platform users to participate in innovation in the platform. Therefore, our findings are expected to lead the research on innovation studies from a pure analysis of the behavior of people (collective intelligence) in networks towards the analysis of the effort of platform providers.

In the future, we intend to extend our studies. We plan to investigate why there is no significant difference in the position of software services within a cluster with respect to the ownership (i.e., platform provider and third party vendors) of the software service. It is different to our expectation, as we also see that the software service network of Salesforce.com is still growing, and the amount of software services offered by third party vendors has surpassed the one of the platform provider in May 2012 already (Figure 3). A hypothesis is that the strategy of the platform provider on network positioning has changed as the critical mass has reached. The platform provider might have let third-party vendors take lead in certain clusters. To validate this hypothesis, we need to investigate the change of network position of software services of the platform provider and third party vendors within the growing software service network.

## References


[1] P. Lévy, "From social computing to reflexive collective intelligence: The IEML research program," *Information Sciences*, vol. 180, no. 1, pp. 71–94, 2010.

[2] A. Weiss, "The power of collective intelligence," *netWorker*, vol. 9, no. 3, pp. 16–23, 2005.

[3] H. W. Chesbrough, *Open Innovation: The New Imperative for Creating and Profiting from Technology*, first trade paper edition edition ed. Boston, Mass.: Harvard Business Review Press, 2005.

[4] E. Von Hippel, "Innovation by user communities: Learning from opensource software," *MIT Sloan Management Review*, vol. 42, no. 4, pp. 82–86, 2001.

[5] J. Altmann, M. Meschke, and A. B. Mohammed, "A classification scheme for characterizing service networks," Seoul National University; Technology Management, Economics, and Policy Program (TEMEP), TEMEP Discussion

Paper 201286, 2012.

[6] S. C. Henneberg, T. Gruber, and P. Naud, “Services networks: Concept and research agenda,” *Industrial Marketing Management*, vol. 42, no. 1, pp. 3–8, 2013.

[7] P. P. Maglio, S. Srinivasan, J. T. Kreulen, and J. Spohrer, “Service systems, service scientists, SSME, and innovation,” *Commun. ACM*, vol. 49, no. 7, pp. 81–85, 2006.

[8] M. E. J. Newman, “Scientific collaboration networks. II. shortest paths, weighted networks, and centrality,” *Physical Review E*, vol. 64, no. 1, p. 016132, 2001.

[9] S. Valverde and R. V. Solé, “Self-organization versus hierarchy in open-source social networks,” *Physical Review E*, vol. 76, no. 4, p. 046118, 2007.

[10] C. S. Wagner and L. Leydesdorff, “Network structure, self-organization, and the growth of international collaboration in science,” *Research Policy*, vol. 34, no. 10, pp. 1608–1618, 2005.

[11] M. S. Granovetter, “The strength of weak ties,” *American Journal of Sociology*, vol. 78, no. 6, pp. 1360–1380, 1973.

[12] R. Grewal, G. L. Lilien, and G. Mallapragada, “Location, location, location: How network embeddedness affects project success in open source systems,” *Management Science*, vol. 52, no. 7, pp. 1043–1056, 2006.

[13] R. Albert, H. Jeong, and A.-L. Barabási, “Internet: Diameter of the world-wide web,” *Nature*, vol. 401, no. 6749, pp. 130–131, 1999.

[14] D. J. Watts and S. H. Strogatz, “Collective dynamics of ‘small-world’ networks,” *Science*, vol. 393, no. 6684, pp. 440–442, 1998.

[15] A. Gawer and M. A. Cusumano, *Platform Leadership: How Intel, Microsoft, and Cisco Drive Industry Innovation*, 1st ed. Boston, Mass: Harvard Business Review Press, 2002.

[16] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann, “Link communities reveal multi-scale complexity in networks,” *Nature*, vol. 466, no. 7307, pp. 761–764, 2010.

[17] L. C. Freeman, “Centrality in social networks conceptual clarification,” *Social Networks*, vol. 1, no. 3, pp. 215–239, 1978.

[18] M. Campbell-Kelly, “Historical reflectionsThe rise, fall, and resurrection of software as a service,” *Communications of the ACM*, vol. 52, no. 5, p. 28, 2009.

[19] M. Cusumano, “Cloud computing and SaaS as new computing platforms,” *Commun. ACM*, vol. 53, no. 4, pp. 27–29, 2010.

[20] L. Youseff, M. Butrico, and D. Da Silva, “Toward a unified ontology of cloud computing,” in *Grid Computing Environments Workshop, 2008. GCE ’08*, 2008, pp. 1–10.

[21] R. Smith, "Computing in the cloud," *Research-Technology Management*, vol. 52, no. 5, p. 65, 2009.

[22] S. Leimeister, M. Bhm, C. Riedl, and H. Krcmar, "The business perspective of cloud computing: Actors, roles and value networks," *ECIS 2010 Proceedings*, 2010.

[23] A. B. Mohammed, J. Altmann, and J. Hwang, "Cloud computing value chains: Understanding businesses and value creation in the cloud," in *Economic Models and Algorithms for Distributed Systems*, ser. Autonomic Systems, D. Neumann, M. Baker, J. Altmann, and O. Rana, Eds. Birkhäuser Basel, 2010, pp. 187–208.

[24] M. Boniface, B. Nasser, J. Papay, S. Phillips, A. Servin, X. Yang, Z. Zlatev, S. Gogouvitis, G. Katsaros, K. Konstanteli, G. Kousiouris, A. Menychtas, and D. Kyriazis, "Platform-as-a-service architecture for real-time quality of service management in clouds," in *2010 Fifth International Conference on Internet and Web Applications and Services (ICIW)*, 2010, pp. 155–160.

[25] V. Gonalves and P. Ballon, "Adding value to the network: Mobile operators experiments with software-as-a-service and platform-as-a-service models," *Telematics and Informatics*, vol. 28, no. 1, pp. 12–21, 2011.

[26] G. Lawton, "Developing software online with platform-as-a-service technology," *Computer*, vol. 41, no. 6, pp. 13–15, 2008.

[27] D. S. Evans, *Platform Economics: Essays on Multi-Sided Businesses.* CreateSpace Independent Publishing Platform, 2011.

[28] R. Qu and Q. Ye, "Service ecosystems of cloud computing." Atlantis Press, 2010.

[29] S. Jansen and S. Brinkkemper, *Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry.* Cheltenham, UK ; Northampton, MA: Edward Elgar Pub, 2013.

[30] K. Karhu, A. Botero, S. Vihavainen, T. Tang, and M. Hmlinen, "A digital ecosystem for co-creating business with people," *Journal of Emerging Technologies in Web Intelligence*, vol. 3, no. 3, pp. 197–205, 2011.

[31] A. Gawer, *Platforms, Markets and Innovation.* Cheltenham, UK ; Northampton, MA: Edward Elgar Publishing Ltd, 2009.

[32] T. R. Eisenmann, G. Parker, and M. W. van Alstyne, "Strategies for two-sided markets," *Harvard Business Review*, vol. 84, no. 10, 2006.

[33] C. Freeman, "Networks of innovators: A synthesis of research issues," *Research Policy*, vol. 20, no. 5, pp. 499–514, 1991.

[34] T. G. Lewis, *Network Science: Theory and Applications*, 1st ed. Hoboken, N.J: Wiley, 2009.

[35] A. Everard and R. Henry, "A social network analysis of interlocked directorates in electronic commerce firms," *Electronic Commerce Research and Applica-*

*tions*, vol. 1, no. 2, pp. 225–234, 2002.

[36] K. Kim, W.-R. Lee, and J. Altmann, "SNA-based innovation trend analysis in software service networks," *Electronic Markets*, pp. 1–12, 2014.

[37] K. Kim and J. Altmann, "Evolution of the software-as-a-service innovation system through collective intelligence," *International Journal of Cooperative Information Systems*, vol. 22, no. 03, p. 1340006, 2013.

[38] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Reviews of Modern Physics*, vol. 74, no. 1, pp. 47–97, 2002.

[39] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[40] J. Scott, *Social Network Analysis: A Handbook*, 2nd ed. London ; Thousands Oaks, Calif: SAGE Publications Ltd, 2000.

[41] L. Festinger, K. W. Back, and S. Schachter, *Social Pressures in Informal Groups: A Study of Human Factors in Housing.* Stanford University Press, 1950.

[42] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annual Review of Sociology*, vol. 27, pp. 415–444.

[43] T. C. Schelling, *Micromotives and Macrobehavior.* New York: w. w. Norton & Company, 2006.

[44] R. S. Burt, *Structural Holes: The Social Structure of Competition.* Cambridge, Mass.: Harvard University Press, 1995.

[45] A. B. Hargadon, "Brokering knowledge: Linking learning and innovation," *Research in Organizational Behavior*, vol. 24, pp. 41–85, 2002.

[46] G. Palla, A.-L. Barabsi, and T. Vicsek, "Quantifying social group evolution," *Nature*, vol. 446, no. 7136, pp. 664–667, 2007.

[47] T. S. Kuhn, *The Structure of Scientific Revolutions, 3rd Edition*, 3rd ed. Chicago, IL: The University of Chicago Press, 1996.

[48] R. S. Burt, "Structural holes and good ideas," *American Journal of Sociology*, vol. 110, no. 2, pp. 349–399, 2004.

[49] K. Kim, J. Altmann, and J. Hwang, "An analysis of the openness of the web2.0 service network based on two sets of indices for measuring the impact of service ownership," in *Proceedings of the 2011 44th Hawaii International Conference on System Sciences*, ser. HICSS '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 1–11.

[50] D. Krackhardt and R. N. Stern, "Informal networks and organizational crises: An experimental simulation," *Social Psychology Quarterly*, vol. 51, no. 2, p. 123, 1988.

[51] G. Palla, I. Dernyi, I. Farkas, and T. Vicsek, "Uncovering the overlapping

community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, 2005.

[52] J. Neter, *Applied Linear Statistical Models*, 5th ed. Boston: McGraw-Hill/Irwin, 2004.

[53] J. D. Gibbons and S. Chakraborti, *Nonparametric Statistical Inference, Fifth Edition*, 5th ed. Boca Raton: Chapman and Hall/CRC, 2010.

[54] I. Olkin, Ed., *Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling*, y first printing edition ed. Stanford Univ Pr, 1960.

[55] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[56] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the internet topology," in *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, ser. SIGCOMM '99. New York, NY, USA: ACM, 1999, pp. 251–262.

[57] N. Haile and J. Altmann, *Estimating the value obtained from using a software service platform.* Springer International Publishing, 2013, pp. 244–255.

**TEMEP Discussion Papers**

2010-69: Michael Maurer, Vincent C. Emeakaroha, Ivona Brandic and Jörn Altmann, "Cost and Benefit of the SLA Mapping Approach for Defining Standardized Goods in Cloud Computing Markets"

2010-70: Khin Swe Latt and Jörn Altmann, "A Cost-Benefit-Based Analytical Model for Finding the Optimal Offering of Software Services"

2010-71: Jung Eun Lee, Younghoon Kim, Yeonbae Kim and Donghyuk Choi, "The Impact of Technology Licensing Payment Mechanisms on Firms' Innovative Performance"

2010-72: Dongook Choi and Yeonbae Kim, "Effects of Piracy and Digital Rights Management on the Online Music Market in Korea"

2011-73: Tai-Yoo Kim, Jihyoun Park, Eungdo Kim and Junseok Hwang, "The Faster-Accelerating Digital Economy"

2011-74: Ivan Breskovic, Michael Maurer, Vincent C. Emeakaroha, Ivona Brandic and Jörn Altmann, "Towards Autonomic Market Management in Cloud Computing Infrastructures"

2011-75: Kiran Rupakhetee and Almas Heshmati, "Rhetorics vs. Realities in Implementation of e-Government Master Plan in Nepal"

2011-76: Alireza Abbasi, Jörn Altmann and Liaquat Hossain, "Identifying the Effects of Co-Authorship Networks on the Performance of Scholars: A Correlation and Regression Analysis of Performance Measures and Social Network Analysis Measures"

2011-77: Ivona Brandic, Michael Maurer, Vincent C. Emeakaroha and Jörn Altmann, "Cost-Benefit Analysis of the SLA Mapping Approach for Defining Standardized Cloud Computing Goods"

2011-78: Kibae Kim and Jörn Altmann, "A Complex Network Analysis of the Weighted Graph of the Web2.0 Service Network"

2011-79: Jörn Altmann, Matthias Hovestadt and Odej Kao, "Business Support Service Platform for Providers in Open Cloud Computing Markets"

2011-80: Ivan Breskovic, Michael Maurer, Vincent C. Emeakaroha, Ivona Brandic and Jörn Altmann, "Achieving Market Liquidity through Autonomic Cloud Market Management"

2011-81: Tai-Yoo Kim, Mi-Ae Jung, Eungdo Kim and Eunnyeong Heo, "The Faster-Accelerating Growth of the Knowledge-Based Society"

2011-82: Mohammad Mahdi Kashef and Jörn Altmann, "A Cost Model for Hybrid Clouds"

2011-83: Daeho Lee, Jungwoo Shin and Junseok Hwang, "Application-Based Quality Assessment of Internet Access Service"

2011-84: Daeho Lee and Junseok Hwang, "The Effect of Network Neutrality on the Incentive to Discriminate, Invest and Innovate: A Literature Review"

2011-85: Romain Lestage and David Flacher, “Access Regulation and Welfare”

2012-86: Juthasit Rohitratana and Jörn Altmann, “Impact of Pricing Schemes on a Market for Software-as-a-Service and Perpetual Software”

2012-87: Bory Seng, “The Introduction of ICT for Sustainable Development of the Tourism Industry in Cambodia”

2012-88: Jörn Altmann, Martina Meschke and Ashraf Bany Mohammed, “A Classification Scheme for Characterizing Service Networks”

2012-89: Nabaz T. Khayyat and Almas Heshmati, “Determinants of Mobile Telecommunication Adoption in the Kurdistan Region of Iraq”

2012-90: Nabaz T. Khayyat and Almas Heshmati, “Determinants of Mobile Phone Customer Satisfaction in the Kurdistan Region of Iraq”

2012-91: Nabaz T. Khayyat and Jeong-Dong Lee, “A New Index Measure of Technological Capabilities for Developing Countries”

2012-92: Juseuk Kim, Jörn Altmann and Lynn Ilon, “Using Smartphone Apps for Learning in a Major Korean University”

2012-93: Lynn Ilon and Jörn Altmann, “Using Collective Adaptive Networks to Solve Education Problems in Poor Countries”

2012-94: Ahmad Mohammad Hassan, Jörn Altmann and Victor Lopez, “Control Plane Framework Emergence and its Deployment Cost Estimation”

2012-95: Selam Abrham Gebregiorgis and Jörn Altmann, “IT Service Platforms: Their Value Creation Model and the Impact of their Level of Openness on their Adoption”

2012-96: Ivan Breskovic, Jörn Altmann and Ivona Brandic, “Creating Standardized Products for Electronic Markets”

2012-97: Netsanet Haile and Jörn Altmann, “Value Creation in IT Service Platforms through Two-Sided Network Effects”

2012-98: Soyoung Kim, Junseok Hwang and Jörn Altmann, “Dynamic Scenarios of Trust Establishment in the Public Cloud Service Market”

2012-99: Lynn Ilon, “How Collective Intelligence Redefines Education”

2013-100: Ivan Breskovic, Ivona Brandic and Jörn Altmann, “Maximizing Liquidity in Cloud Markets through Standardization of Computational Resources”

2013-101: Juseuk Kim, Lynn Ilon and Jörn Altmann, “Adapting Smartphones as Learning Technology in a Korean University”

2013-102: Baseem Al-Athwari, Jörn Altmann and Almas Heshmati, “A Conceptual Model and Methodology for Evaluating E-Infrastructure Deployment and Its Application to OECD and MENA Countries”

2013-103: Mohammad Mahdi Kashef, Azamat Uzbekov, Jörn Altmann and Matthias Hovestadt, “Comparison of Two Yield Management Strategies for Cloud Service Providers”

2013-104: Kibae Kim, Wool-Rim Lee and Jörn Altmann, “Patterns of Innovation in SaaS Networks: Trend Analysis of Node Centralities”

2013-105: Netsanet Haile and Jörn Altmann, “Estimating the Value Obtained from Using a Software Service Platform”

2013-106: A. Mohammad Hassan and Jörn Altmann, “Business Impact Analysis of a Mediator between the Network Management Systems of the IP/MPLS Network and the Transport Network”

2013-107: A. Mohammad Hassan and Jörn Altmann, “Disjoint Paths Pair Computation Procedure for SDH/SONET Networks”

2013-108: Kibae Kim and Jörn Altmann, “Evolution of the Software-as-a-Service Innovation System through Collective Intelligence”

2014-109: Shahrouz Abolhosseini, Almas Heshmati and Jörn Altmann, “The Effect of Renewable Energy Development on Carbon Emission Reduction: An Empirical Analysis for the EU-15 Countries”

2014-110: Somayeh Koohborfardhaghighi and Jörn Altmann, “How Placing Limitations on the Size of Personal Networks Changes the Structural Properties of Complex Networks”

2014-111: Somayeh Koohborfardhaghighi and Jörn Altmann, “How Structural Changes in Complex Networks Impact Organizatioanl Learning Performance”

2014-112: Sodam Baek, Kibae Kim and Jörn Altmann, “Role of Platform Providers in Service Networks: The Case of Salesforce.com App Exchange”

2014-113: Somayeh Koohborfardhaghighi and Jörn Altmann, “A Network Formation Model for Social Object Networks”

2014-114: Somayeh Koohborfardhaghighi and Jörn Altmann, “How Variability in Individual Patterns of Behavior Changes the Structural Properties of Networks”

2014-115: Kibae Kim, Wool-Rim Lee and Jörn Altmann, “SNA-Based Innovation Trend Analysis in Software Service Networks”

2014-116: Jörn Altmann and Mohammad Mahdi Kashef, “Cost Model Based Service Placement in Federated Hybrid Clouds”

2014-117: Kibae Kim, Songhee Kang and Jörn Altmann, “Cloud Goliath Versus a Federation of Cloud Davids: Survey of Economic Theories on Cloud Federation”

2014-118: Stefan Niederhafner, “The Korean Energy and GHG Target Management System: An Alternative to Kyoto-Protocol Emissions Trading Systems?”

2014-119: Gunno Park and Jina Kang, “Effect of Alliance Experience on New Alliance Formations and Internal R&D Capabilities”

2015-120: Kibae Kim, Jörn Altmann and Sodam Baek, “Role of Platform Providers in Software Ecosystems”