Platform as a Service (PaaS)

DOI 10.1007/s12599-011-0183-3

The Authors

Dr. Daniel Beimborn (✉)
Thomas Miletzki, B.Sc.
Faculty of Information Systems
and Applied Computer Sciences
University Bamberg
Feldkirchenstr. 21
96045 Bamberg
Germany
daniel.beimborn@uni-bamberg.de
thomas.miletzki@uni-bamberg.de

Dipl.-Wirt.Inf. Stefan Wenzel
SAP Store & Commercial
Infrastructure Solution Management
SAP AG
Dietmar-Hopp-Allee 16
69190 Walldorf
Germany
stefan.wenzel@sap.com

Received: 2011-01-16
Accepted: 2011-09-12
Accepted after four revisions
by Prof. Dr. Sinz.
Published online: 2011-10-27

This article is also available in German in print and via http://www.wirtschaftsinformatik.de: Beimborn D, Miletzki T, Wenzel S (2011) Platform as a Service (PaaS). WIRTSCHAFTSINFORMATIK. doi: 10.1007/s11576-011-0294-y.

1 PaaS as an Important Part of the Cloud Economy

To provide and consume software and hardware as services, labeled under the umbrella term "cloud computing", has proved to be a sustainable trend, both in the B2C and in the B2B market (Bandulet et al. 2010). Usually, we distinguish between the concepts of Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). While IaaS provides hardware virtually and "on demand", SaaS offers software applications which can be used via the Internet or other networks. The concept of Platform-as-a-Service often remains invisible to the user as it provides the necessary operating platforms for the virtually provided applications. Nevertheless, the PaaS concept has successfully shown to be a service model that can be offered independently, as the Google App Engine or Force.com of Salesforce.com exemplify. Thus, PaaS extends the role model of the SaaS ecosystem to the platform provider as an additional actor and thus offers interesting new aspects to Information Systems research and the software industry.

Grohmann (2009, p. 60–61) defines PaaS as the "provision of a complete platform, i.e. hardware AND software, as service" in order to give independent software vendors (ISVs) the opportunity to "develop and to provide SaaS solutions or to integrate them with traditional software applications". The platform provides the ISVs with all functionalities which are needed during the lifecycle of an application, from development via testing to deployment and operations (Mitchell 2008).

From an economic perspective, the introduction of PaaS as a business model of its own represents a shift in the three roles of the software market. While the customer/user still uses the software on demand via the network (as in a pure SaaS model), the relation between software developer and software provider changes. In the ASP (application service providing) model, it was mainly the platform provider's responsibility to implement the software and to provide an adequate infrastructure, while, in the PaaS context, the platform, including the development environment, is pre-determined and has to be used by the ISV.

2 Basics of the PaaS Model

PaaS offers a complete set of technologies which are required to develop and to operate SaaS applications. Additionally, many platforms provide marketing and sales opportunities (e.g., Google App Marketplace) and other services along the software value chain. **Figure 1** shows the core components and optional elements of a PaaS platform.

PaaS's most central component is the Application Runtime Environment (ARE), which has to fulfill the typical requirements such as scalability, reliability, and security. Often, the ARE supports a multi-tenancy architecture which allows multiple users to share a single instance of a SaaS application.

Beside the ARE, the PaaS provider usually provides an Integrated Development Environment (IDE), which normally supports the use of multiple programming languages and offers a wide variety of libraries and tools for modeling, implementing, testing, and versioning. Depending on the application domain, various database systems are also provided. Moreover, external data sources can be integrated, e.g., via web service interfaces.

Basically, two forms of PaaS offers can be distinguished, depending on if they include the component "SaaS core application" or not (cf. **Fig. 1**). The components described so far form so-called *pure PaaS* offers, such as the Google App Engine. On the other hand, some of the large software firms also provide platforms which allow ISVs to develop extensions or "add-ons" for the software firm's core application. An example for this *application-based PaaS* (aPaaS) is Force.com with its core application Salesforce.com. The core application has to be seen as a necessary part of the platform because any third-party software running on this platform only makes sense as an add-on to the core application but not as a stand-alone software.

Apart from these core components of PaaS, many platform providers also offer additional services, which are important for marketing, distribution, and operations of the applications. These include support (e.g., ticket systems), quality reviews, certification of applications, as well as monitoring functionalities which allow the localization of bottlenecks, errors, and optimization potentials. Finally, PaaS providers often provide an online marketplace which supports the ISVs' sales activities.

In order to distinguish themselves from competitors, PaaS providers often develop extra value-added services which

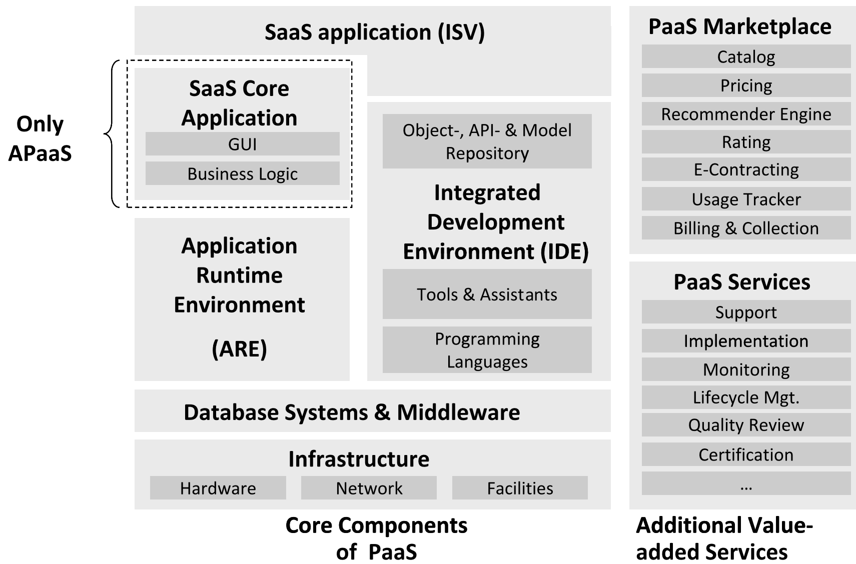

Fig. 1 The PaaS stack (based on Dubey et al. 2008; Gillett 2008)

increase the attractiveness of the platform and tie the ISVs to this particular platform. One example is the value-added service of billing&collection. This service contains all activities which are needed to transfer the payment for using the software from the consumer to the ISV. Comprehensive billing&collection services cover the generation and delivery of invoices, money transfer via various payment infrastructures, and dunning and collection.

3 Roles in the PaaS Ecosystem

The PaaS ecosystem consists of three groups of actors: the PaaS provider, the ISV, and the SaaS customer. Their roles will be explained in the following.

In the case of aPaaS (application based PaaS), a software firm offers a core application (usually as a service, SaaS). In order to increase the scope (and thus the revenues) of this product, this firm also provides a platform which enables ISVs to develop add-ons for the core application. Thus, the core application is extended by additional functionality and can cover more customer segments and satisfy more individual customer needs. By contrast, pure PaaS does not show comparable dependencies between the ISVs' applications and a core product.

The second group consists of ISVs who develop and deploy (stand-alone) applications (in case of pure PaaS) or add-ons for the core application (aPaaS) on the PaaS platform. They do not need to worry about the infrastructure and can solely focus on the development and marketing/sales activities regarding their product.

The customers or users of the software, as the third actor in this environment, often do not even realize the PaaS nature of the scenario. They receive the application developed by the ISV as SaaS service from the platform provided by the PaaS provider. However, from a contractual perspective, the PaaS scenario can imply that the customer has to engage in multiple contractual arrangements with different parties (e.g., one contract with the PaaS provider and multiple contracts with the add-on developers). Ideally, the PaaS provider tries at least to reduce this complexity during later steps of the software lifecycle. The software customer should receive consistent and transparent bills and should be offered one single support channel for all applications running on the same platform. **Figure 2** shows a typical interaction model between the parties involved in an aPaaS scenario.

The aPaaS provider operates the platform and, based on this, provides both the core application and the add-ons, developed by the ISV, to the customer. As a rule, he will receive a proportional share of the ISV's revenue (while in a pure PaaS scenario the platform provider usually receives usage based fees for operating the application on the platform, e.g., based on bandwidth or processing/storage usage). Additionally, the platform provider will normally charge fees for additional services, e.g., for certifying the applications/add-ons. The customer pays the fees directly to both the ISV and the aPaaS provider. The latter will provide the complete support for the core application and the first-level support for the add-on. Thus, the provider will become the first contact for the customer since the latter cannot know who is responsible for a certain problem when using the bundled SaaS solution (i.e., core application + add-ons + platform infrastructure). In case of an incident, the PaaS provider will first identify the source of the problem and then, if necessary, involve the ISV to provide second-level and third-level support.

4 Examples

In the following, we will introduce Google App Engine (GAE) as an example for pure PaaS and SAP Business ByDesign Studio as an example for aPaaS.[1]

With its GAE (http://appengine.google.com), Google enables the development and operations of web applications on the basis of the same technical infrastructure which is used by their native Google apps. The ISVs need to install a local GAE-specific SDK or Eclipse-Plugin to develop and test their applications. After publication on the platform, the ISV can monitor usage access and resource consumption. An in-house data store technology enhancing the *Google File System* is deployed as database solution. The GAE is offered for free for a restricted level of usage; for excess resource

[1]A detailed comparison of current PaaS offers is provided at http://isdl.uni-bamberg.de/paas/appendix.pdf.

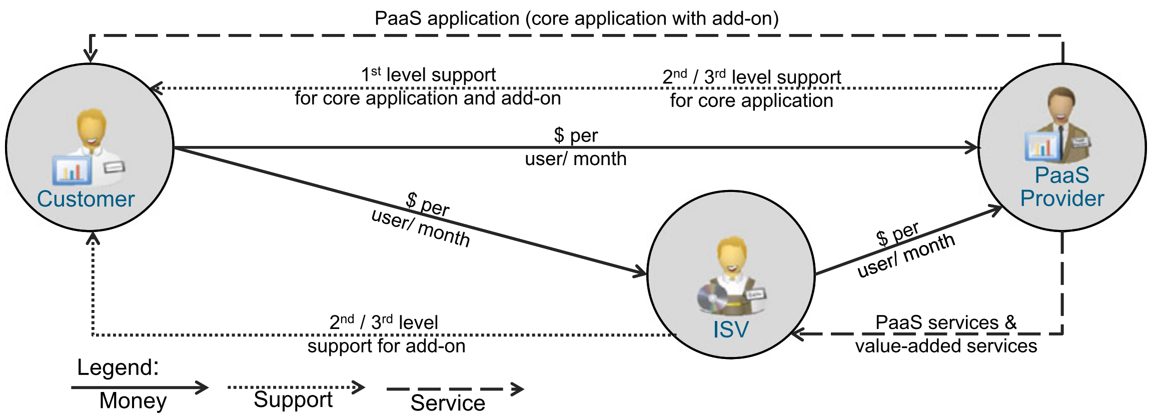

Fig. 2 Flows of money and services in the aPaaS scenario

consumption a usage fee is charged. Optionally, the ISVs can integrate their applications with functionalities offered by the Google Apps (e.g., Gmail, Google Calendar) and use Google Accounts to authenticate their users. To market & sell the applications, the ISVs can use the Google Solutions Marketplace for which Google charges a proportional share of the revenue.

SAP AG offers an aPaaS related to its SaaS solution SAP Business ByDesign (http://www.sap.com/solutions/products/sap-bydesign/). SAP Business ByDesign is an integrated business management solution covering all relevant corporate functions of medium-sized companies. Since 2011, SAP supports ISVs with an SDK, the SAP Business ByDesign Studio which enables them to extend SAP's core application by industry-specific or customer-individual add-ons.

SAP Business ByDesign Studio is based on Microsoft Visual Studio and uses two separate, proprietary programming languages: the Business Object Declaration Language (BODL) to define business objects and the Advanced Business Scripting Language (ABSL) to define business logic. Further, the graphical editor "UI designer" supports developers to quickly model and implement the user interface while a library with reference code illustrates the most common coding patterns. To build the desired functionality, the ISV can reuse the integrated business processes which are already available with SAP Business ByDesign. Further, the lifecycles of the add-ons and of the core application are decoupled. This allows the ISV to supply updates independently from the release cycle of the core application. After the mandatory quality review, the add-ons can be distributed via the online marketplace "SAP Store" (http://store.sap.com). Once the online purchase is completed, the add-ons are instantly deployed to the customer's application environment (see Faisst 2011).

5 Value Analysis and Economic Aspects

With the PaaS model, each of the actors pursues their own interests. This section discusses the opportunities and risks for each of the individual roles.

Regarding development, marketing, and sales, the platform services provide substantial cost saving potentials for the *ISVs*, e.g., by using standardized certification and marketing processes of the platform. Especially for aPaaS, ISVs gain access to a previously "closed market" of potential new customers. The usage-based fees for the PaaS services allow a market entry with low startup costs.

However, when using the "shared services" the ISVs run the risk of giving up strategic areas of control within their value chain to the platform provider (e.g., giving away direct access to customers by using centralized marketing and customer service processes). Further, ISVs might get locked into the platform due to proprietary standards, which are common on platforms. To mitigate this risk, ISVs need to develop and maintain their applications on multiple, competing platforms (known as "multihoming", Armstrong and Wright 2007).

From the *user* perspective, the software acquisition and operations model changes from a software license and in-house operations model to an external operations and usage fee model. The usage fee can either be a usage based (e.g., per transaction, i.e., "pay per use") or a usage independent fee (e.g., per user and month, i.e., "pay per period") (Buxmann 2009). Contrary to the traditional software license model, the usage fees for the SaaS application also include operations, maintenance, and support. Additionally, upfront investments for hardware, licenses, and implementation projects are reduced. This does not necessarily lead to lower TCO, but can help limit dependencies in terms of lock-in effects (Bandulet et al. 2010; Buxmann 2009). Particularly for low or highly volatile usage patterns, this more variable cost structure can result in substantial cost advantages.

Due to the virtualization of the service provisioning, safeguarding the infrastructure and technical operations of the application is part of the PaaS provider's duties. The end-to-end warranty of application availability, however, is in both the PaaS provider's and the ISV's responsibility, as the latter accounts for the functionality of the application. This decentralization of duties increases the operational risks of PaaS. In general, Armbrust et al. (2009) see the service availability as the largest hurdle for cloud computing. Consequently it is necessary that the user agrees with the ISV – and the ISV in turn with the PaaS provider – on certain service level agreements which contractually regulate the service availability, their protection, and possible consequences in case of non-compliance.

The economic aspects for the *platform provider* depend on the PaaS model. Whereas pure PaaS mainly aims at transforming existing idle capacities (and fixed costs) into earnings or on generating revenue by renting out resources, the aPaaS provider is primarily interested in increasing the attractiveness and marketability of the core application. The

ISVs' third party solutions can address the specific needs of single companies or small industry sectors and thus increase the market penetration of the core application.

Since the PaaS provider and the ISVs jointly compile a compound service for the end customer, the quality of the ISVs' applications can also affect the reputation of the platform provider – especially in the aPaas model. Therefore, quality assurance of the platform should consider also "soft" criteria like performance, usability, and the application's semantics, besides "hard" criteria like security and correctness.

A key economic aspect of PaaS is the existence of network effects (Arthur 1989) and multi-sided markets (Rochet and Tirole 2003). A platform will only be attractive for an ISV if it can expect many users. This again will only be the case if many ISVs are active on the platform, i.e., an ISV's decision in favor of a certain platform depends on the behavior of other ISVs. To reach a critical mass of positive decisions among the group of relevant ISVs, the platform provider has to develop adequate marketing strategies. Therefore it needs to create expectations about whether a sufficient number of users can be acquired. Consequently, the success of the platform depends on multiple market sides, whose decisions are determined by the behavior of both the same and the other side.

PaaS can be seen as a special configuration of the software value chain which offers the potential to create a sustainable "win/win/win situation" for all relevant stakeholders. The strongly interdependent individual value potentials, however, require to consequently take the risks related with the PaaS model into consideration.

6 Research Potential in the Field of PaaS

The PaaS concept as an independent business model adds a new facet to the software value chain and thus implies new important questions for Information Systems research. First of all, it is essential to define a commonly accepted framework or reference model, to better compare and evaluate the highly differing offerings on the market. Further, Tiwana et al. (2010) identify important research questions in the area of platform architecture and platform governance (distribution of decision rights among platform provider and ISVs, design of control mechanisms etc.). Beyond that, the economic aspects are interesting. As described above, the success of PaaS providers depends on the dynamics of two-sided markets. Analyzing these interdependencies via multi-sided adoption models and diffusion models will contribute to better understanding these dynamics, and in turn can help to assess the contribution of value-added services to the economic success of a platform.

From the point of view of welfare economics, the question arises which influence PaaS can and will have on the market efficiency and innovativeness of the entire software industry. For example, PaaS can be beneficial since small ISVs will have increasing opportunities to quickly and successfully implement and market new ideas.

Finally, PaaS is a special configuration class of the software value chain whose existence can be explained by various trends and theories. In this context it is required to understand how the trends of "consumerization", standardization and industrialization of IT will influence PaaS and to what extent PaaS will embrace mass-customization, known from the consumer industry, to satisfy the needs of the "long tail".

7 Outlook

PaaS as a stand-alone business model constitutes a step in the evolution towards the service paradigm and will become an important component in the software value chain. Forrester analysts estimate a market volume of up to 15.2 billion USD for 2016 (Ried et al. 2009). Moreover, PaaS is an advancement with regard to the industrialization of the software industry, as specialization and distribution of work on multiple different actors increases. However, network effects will lead to consolidation tendencies also in this market. In the medium term, smaller platforms will cooperate or merge with larger platforms, as the cooperation between the Intuit platform with Microsoft Azure has recently shown.

References

Armbrust M, Fox A, Griffith R, Joseph A, Katz RL, Konwinski A, Lee G, Patterson D, Rabkin A, Stoica I, Zaharia M (2009) Above the clouds: a Berkeley view of cloud computing. University of California, Berkeley, CA. http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf. Accessed 2011-09-01

Armstrong M, Wright J (2007) Two-sided markets, competitive bottlenecks, and exclusive contracts. Economic Theory 32(2):353–380

Arthur WB (1989) Competing technologies, increasing returns, and lock-in by historical events. Economic Journal 99:116–131

Bandulet F, Faisst W, Eggs H, Otyepka S, Wenzel S (2010) Software-as-a-Service as disruptive innovation in the enterprise application market. In: Benlian A, Hess T, Buxmann P (eds) Software-as-a-Service (SaaS) – Über den Wandel von Anbieterstrategien, Kundenbedürfnissen und Marktstrukturen in der SW-Industrie. Gabler, Wiesbaden

Buxmann P (2009) Software-as-a-Service. Enzyklopädie der Wirtschaftsinformatik – Online. http://www.enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon/is-management/Integration-und-Migration-von-IT-Systemen/Software-as-a-Service/index.html/?searchterm=saas. Accessed 2011-09-01

Dubey A, Mohiuddin J, Baijal A (2008) Emerging platform wars in enterprise software. McKinsey

Faisst W (2011) Die nächste Generation der Unternehmenssoftware am Beispiel von SAP Business ByDesign. Wirtschaftsinformatik und Management 4:24–31

Gillett FE (2008) Future view: the new tech ecosystems of cloud, cloud services, and cloud computing. Forrester

Grohmann W (2009) Von der Software zum Service. ASP – Software on Demand – Software-as-a-Service – Cloud Computing. H.K.P. Consulting, München

Mitchell D (2008) Defining platform-as-a-service or PaaS. http://bungeeconnect.wordpress.com/2008/02/18/defining-platform-as-a-service-or-paas/. Accessed 2011-09-01

Ried S, Garbani JP, Bartels A, Lisserman M (2009) Platform-as-a-service market sizing. Forrester

Rochet JC, Tirole J (2003) Platform competition in two-sided markets. Journal of the European Economic Association 4(1):990–1029

Tiwana A, Konsynski B, Bush A (2010) Platform evolution: coevolution of platform architecture, governance, and environmental dynamics. Information Systems Research 21(4):675–687