CrossMark

# Integration Platform as a Service

**Nico Ebert · Kristin Weber · Stefan Koruna**

## 1 Introduction

Around 2000, the term "enterprise application integration" (EAI) became popular with IT departments. Until then, silo applications – custom-developed or packaged, host- or client–server-based – had been connected with numerous point-to-point interfaces. The resulting complexity of the application landscape affected the implementation of additional business requirements such as electronic marketplaces, e-commerce, and supply chain management. Commonly used middleware techniques such as message queues, object request brokers, or remote procedure calls required significant and costly alterations of source and target systems. Instead, EAI promised to reduce the

Dr. N. Ebert (✉)
Institute of Business Information Technology, ZHAW Zurich
University of Applied Sciences, St.-Georgen-Platz 2,
8401 Winterthur, Switzerland
e-mail: nico.ebert@zhaw.ch

Prof. Dr. K. Weber
Faculty of Computer Science and Business Information Systems,
University of Applied Sciences Würzburg-Schweinfurt,
Sanderheinrichsleitenweg 20, 97074 Würzburg, Germany
e-mail: kristin.weber@fhws.de

Dr. S. Koruna
ZHAW Zurich University of Applied Sciences, St.-Georgen-
Platz 2, 8401 Winterthur, Switzerland

application landscape complexity by connecting applications with a centralized, application-independent "information broker" by means of standardized connectors and a definable integration logic (Linthicum 2000, pp. 10ff.). EAI has since been adopted in many large enterprises (e.g., banks).

Today, nearly two decades later, firms are faced with new integration challenges. Increasingly, enterprise IT landscapes include best-of-breed business applications, platforms, and infrastructures sourced from cloud service providers (Kleeberg et al. 2014, p. 39). Some of the newly founded companies solely depend on so-called "software as a service" (SaaS). While future "Internet of Things" (IoT) platforms may fully reside in the cloud, they will, nevertheless, still need to exchange data with legacy systems (Wortmann and Flüchter 2015). And again, the question arises of how to integrate these new cloud-based applications.

A relatively new alternative to point-to-point integration and existing EAI middleware is the concept of "integration platform as a service" (IPaaS) (Potočnik and Juric 2012). IPaaS can be considered as the cloud-based equivalent to EAI. IPaaS promises both mature EAI functionalities and the benefits of SaaS applications such as high productivity and predictable costs. While EAI has been technically very complex, has required specialized skills and has been very expensive to maintain, IPaaS are less complex and easier to use. Therefore, the initial integration of a new application is faster and maintenance costs for changes of existing integrations are lower.

For example, the Swiss-based logistics enterprise Kardex successfully uses an IPaaS to synchronize customer and order data across its mobile field service app, a SaaS resource planning tool, and on-premise enterprise resource planning (ERP) (Boillat and Legner 2014).

🖄 Springer

## 2 Key Characteristics and Benefits of IPaaS

According to Serrano et al. (2014) "iPaaS is a suite of cloud services that enable users to create, manage, and govern integration flows connecting a wide range of applications or data sources without installing or managing any hardware or middleware". Terms used interchangeably to IPaaS are "cloud integration platform", "cloud-based integration", "cloud platform integration", and "integration cloud". An IPaaS is a multi-tenant system (i.e., multiple clients share the same instance) on a cloud infrastructure managed by a service provider. According to the classification created by Ruh et al. (2000, p. 19), it addresses the lower levels of data and functional integration within an organization (see Fig. 1).

The main components of an IPaaS are very similar to EAI (cp. Ring 2000):

1. Integration processes that specify the logic of how and when data is exchanged between applications (e.g., event-based synchronization of multiple applications when a new customer record is created in a leading application).
2. Data mappings between the attributes of source and destination data objects (e.g., customer object of one application to client object of another application).
3. Pre-built adapters to connect to different types of applications (e.g., ERP, CRM, etc.)
4. Functionalities to support the development of components 1–3 (e.g., visual process modeling and data mapping tools or software development kits for adapters) and the execution of the integration processes (e.g., e-mail alerts if execution was not successful).

As opposed to the original EAI platforms, in IPaaS the above-mentioned components are often easier to use because they have been developed more than a decade later and usability has been more strongly emphasized. Furthermore, the time from development to execution of an integration process is reduced due to the fact that many steps previously requiring programming can simply be done using "drag and drop" and configuration. To reduce development time further, IPaaS vendors offer web-based market places for pre-built adapters as well as integration process and data mapping templates. Because multiple clients use the same platform, some IPaaS even
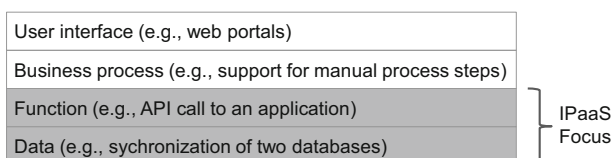
automatically suggest potentially applicable data mappings to the user based on similar mappings created by other users (e.g., Dell 2016). However, compared to EAI new questions must be addressed using IPaaS, too. The use of a platform in the cloud might require compliance checks with laws and internal regulations as well as create technical challenges such as an appropriate firewall setup. These factors may complicate and slow down the deployment in contrast to EAI.

An IPaaS can cover multiple integration scenarios (cf. Table 1). If an enterprise wants to integrate multiple cloud-based applications, it most likely does not want to deploy any on-premise integration infrastructure at all, preferring instead to benefit from a cloud-based IPaaS (Scenario 1). The Swiss pharmaceutical company Novartis uses an IPaaS to integrate cloud-based and on-premise applications (Scenario 2) because in Novartis' case integrating a new cloud-based application has proven dramatically quicker and cheaper than using a traditional EAI platform (Ovum 2013). However, even if enterprises want to focus on integration between on-premise applications, an IPaaS might be the preferred choice compared to an EAI platform (Scenario 3). For example, Oneworld, an airline alliance with members such as American Airlines, British Airways, and Air Berlin, uses an IPaaS as an IT hub to simplify integration among the heterogeneous IT architectures of its member airlines (Oneworld 2012).

## 3 Classification of Existing Platforms

For a better understanding, the existing IPaaS landscape can be divided in two major segments: Private user and small enterprise platforms as well as enterprise platforms.
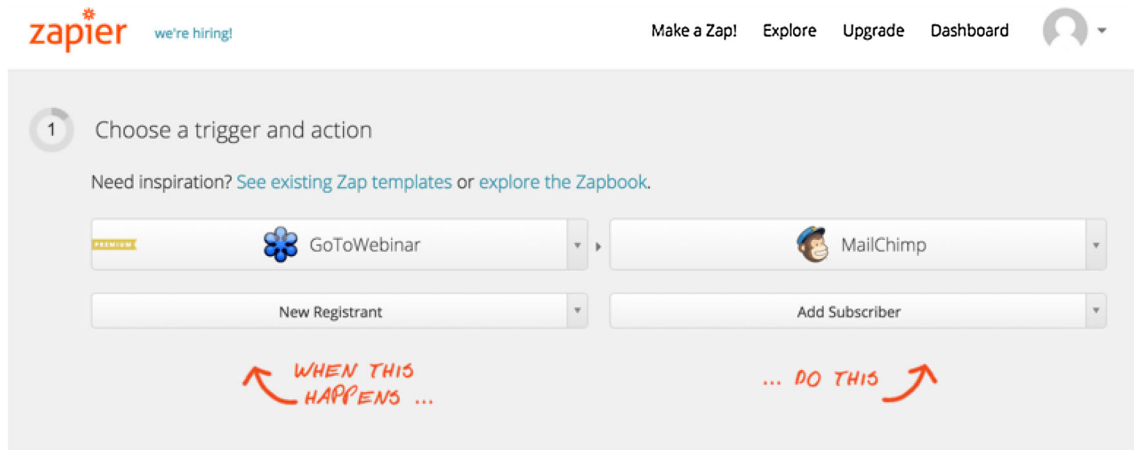
### 3.1 Private-User and Small Enterprise Platforms

Popular examples of platforms within this segment are "IFTTT"[1] (if this then that) and "Zapier".[2] These platforms allow users to easily connect different web applications to automate tasks in cloud-to-cloud integration scenarios (Ng 2015). Designing these integration processes does not require special technical knowledge. By using a web-based tool, the user visually connects different pre-built adapters of the applications involved and then selects basic triggers and actions. Figure 2 provides an example in which new subscribers of a company webinar are automatically registered for the company newsletter. A small HR department could integrate Gmail and Dropbox to transfer incoming job application files automatically to a



Fig. 1 Levels of integration addressed by IPaaS

---

[1] https://ifttt.com.

[2] https://zapier.com.

**Table 1** Scenarios for the usage of IPaaS

| Scenario | Description | Example |
| --- | --- | --- |
| 1. Cloud to cloud | Integration purely between cloud-based applications | User profiles in different social networks are synchronized with contact data of a cloud-based CRM |
| 2. Cloud to on-premise | Integration of cloud-based applications with existing on-premise applications | A cloud-based CRM system is connected to a legacy ERP system to synchronize customer data |
| 3. On-premise to on-premise | Integration solely between on-premise application | Airlines of an alliance synchronize their passenger information systems |



**Fig. 2** Integrating a webinar and newsletter tool in Zapier (https://zapier.com)

shared Dropbox team folder whenever a new application arrives at jobs@company.com. Another example is a private user who connects his or her cloud-based car information system to a cloud-based garage information system to open the garage door whenever the car enters the driveway.

Pre-built adapters exist mainly for private-user and small enterprise applications (e.g., small CRM systems). Most of these adapters rely on simple stateless RESTful APIs based on HTTP, which are provided by most web applications. The user is not able to custom-develop adapters, which limits the platforms' connectivity.

Once the integration process has been designed, it is executed in the cloud-based environment. The trigger for execution can be an event (e.g., new order arrived) or a predefined schedule (e.g., once per hour). The user can monitor its successful execution. To sum up, these platforms allow an easy and quick integration of predefined cloud-based applications without any need for programing skills.
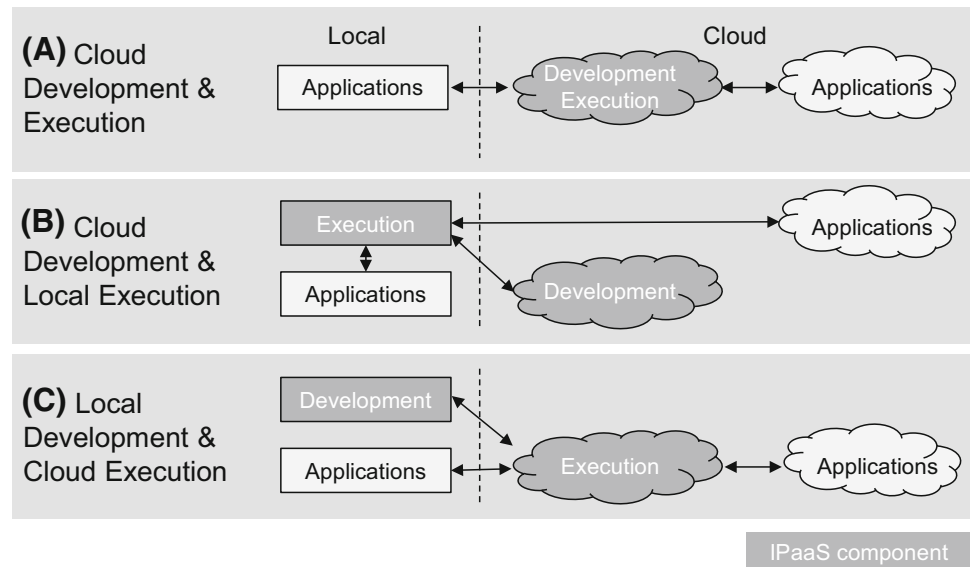
### 3.2 Enterprise Platforms

Enterprise class integration platforms support larger enterprises in all of the three integration scenarios described in Table 1. These platforms can also be combined with existing on-premise EAI platforms so that established connections to the EAI platform do not have to be replaced. Popular enterprise platforms are Dell Boomi, Informatica Cloud, Mulesoft, and SAP HCI.

Enterprise platforms allow the development of a highly complex integration process. The visual design tools use (often proprietary) process modeling languages to describe data flows between applications including logic-based branches, process hierarchies and complex data transformation operators. Some platforms even allow freely definable data transformation rules based on scripting or programming languages (e.g., Java). Furthermore, the platforms support advanced EAI concepts such as message queues or transaction processing. Synchronous and asynchronous coupling mechanisms are available as well as single/batch and scheduled/event-based execution of the integration processes. Numerous pre-built application adapters for professional business applications are also available. These adapters range from file-based and HTTP adapters to more complex business application adapters (e.g., SOAP adapters for the salesforce) and EDI adapters for inter-enterprise communication. The user can custom-develop adapters for enterprise-specific applications facilitated by the platform vendors' software development kits.

**Fig. 3** Architecture variants of enterprise IPaaS



In contrast to the original definition of IPaaS created by Serrano et al. (2014), enterprise platforms do not necessarily have to be fully cloud-based. Instead, three different architectural variants can be distinguished, two of which use the cloud only partially (see Fig. 3).

In Architecture A, the development is supported by web-based design tools storing meta-data, such as data mappings and integration process definitions, in the cloud. During the execution of the integration processes, actual application data are transferred through the IPaaS cloud. Platform users can flexibly scale the platform size according to the data traffic and do not need to manage an integration infrastructure. Compared to the other Architecture variants, Architecture A requires the least amount of time from the development to the execution of integration processes. Examples of enterprise platforms supporting such an architecture are SAP HCI and Dell Boomi.

While the development in Architecture B is web-based, the integration processes are executed locally. Integration processes are deployed to a local execution environment fully managed by the user. Vendors typically provide these environments, e.g., for Windows or Linux. If only on-premise applications are integrated, no application data need to leave the enterprise. Compared to the cloud-based execution environment of Architecture A and C, the local execution environment of Architecture B requires a ramp-up phase for its setup and might be less easy to scale. Examples of enterprise platforms supporting this architecture are Informatica Cloud and Dell Boomi.

In Architecture C, the development relies on on-premise tools (e.g., based on Eclipse), which are typically more powerful than their web-based counterparts and allow custom programming. In a second step, processes are deployed to a service provider's cloud-based execution

environment similar to Architecture A. Compared to Architecture B, the user neither needs to setup nor manage the execution environment but can simply scale the environment according to resource demands. Informatica Cloud and Mulesoft are examples of platforms supporting this architecture.

## 4 Challenges and Future Directions

The few case studies on IPaaS not published by vendors conclude that the platforms can be effective alternatives to classical on-premise integration tools or point-to-point integration (e.g., Janković et al. 2011; Boillat and Legner 2014). However, as the market for platforms is young and very fragmented (cp., Guttridge et al. 2016), users will have to be careful in selecting a platform matching their demands.

Data security and privacy are among the major critical challenges in cloud integration (Kleeberg et al. 2014). One critical aspect is that application data is transferred through the Internet, with all the potential risks that this entails. Another is the fact that meta-data (e.g., passwords to access integrated applications) and application data are shared not only with the platform provider but also, potentially, with a third-party provider of the underlying cloud infrastructure, such as Amazon. Vendors address these aspects, for example, with different security standards (e.g., HTTPS) and internal information security management systems such as ISO 27001. However, the transparency of many vendors with regard to information security is limited and some security measures seem to be inappropriate (Ebert and Weber 2016). Other common challenges of cloud-based applications are performance and portability

(Moreno-Vozmediano et al. 2013). A lack of performance (or a performance unpredictability) of either the Internet connection or the cloud platform itself can negatively affect the message and data exchange between enterprise applications. A limited degree of portability among integration platforms of different vendors may lead to a strong vendor lock-in.

From a research perspective, not only security-related and technical questions such as how to improve security or performance are of interest. As empirical research on IPaaS is currently rare, many interesting business and technical issues on the platforms and use cases have not been addressed. For example, critical success factors for using IPaaS as well as advantages and disadvantages compared to classical EAI tools have not been investigated in detail. Furthermore, new cloud-based business process management systems are evolving (Schulte et al. 2015) and the boundaries to IPaaS are not yet clear.

## References

Boillat T, Legner C (2014) Why do companies migrate towards cloud enterprise systems? A post-implementation perspective. In: 2014 IEEE 16th conference on business informatics, 1:102–109. doi:10.1109/CBI.2014.46

Dell (2016) Boomi suggest and your privacy – Dell Boomi. https://www.boomi.com/privacy/suggest/. Accessed 30 Dec 2016

Ebert N, Weber K (2016) Sicherheit von Cloud-basierten Plattformen zur Anwendungsintegration: eine Bewertung aktueller Angebote. FHWS Sci J 3:10–22

Guttridge K, Pezzini M, Malinverno P, Iijima K, Thompson J, Thoo E, Golluscio E (2016) Magic quadrant for enterprise integration platform as a service, Worldwide. https://www.gartner.com/doc/3263719/magic-quadrant-enterprise-integration-platform. Accessed 6 Jan 2017

Janković S, Mladenović S, Radonjić V, Kostić-Ljubisavljević A, Uzelac A (2011) Integration platform-as-a-service in the traffic safety area. In: MIC-CNIT2011, Mosharaka international conference on communications, networking and information technology, Dubai, UAE, pp 70–75

Kleeberg M, Zirpins C, Kirchner H (2014) Information systems integration in the cloud: scenarios, challenges and technology trends. In: Brunetti G, Feld T, Heuser L, Schnitter J, Webel C (eds) Future business software. Springer, Heidelberg, pp 39–54

Linthicum DS (2000) Enterprise application integration. Addison-Wesley, Boston

Moreno-Vozmediano R, Montero RS, Llorente IM (2013) Key challenges in cloud computing: enabling the future internet of services. IEEE Int Comput 17:18–25. doi:10.1109/MIC.2012.69

Ng JW (2015) Task as a service: extending cloud from an application development platform to a tasking platform. In: Proceedings – 2015 IEEE world congress on services, SERVICES 2015, pp 294–301

Oneworld (2012) Oneworld IT hub in the clouds smooths airberlin's addition – Asset Publisher. https://www.oneworld.com/news-information/oneworldnews/-/asset_publisher/QtTQ7EuCzxhd/content/oneworld-it-hub-in-the-clouds-smooths-airberlin-s-addition/pop_up?_101_INSTANCE_QtTQ7EuCzxhd_viewMode=print. Accessed 6 Jan 2017

Ovum (2013) OvumCase study: adoption of Dell Boomi AtomSphere iPaaS by Novartis – Ovum. https://www.ovum.com/research/case-study-adoption-of-dell-boomi-atomsphere-ipaas-by-novartis/. Accessed 6 Jan 2017

Potočnik M, Juric MB (2012) Integration of SaaS using IPaaS. In: The 1st international conference on CLoud Assisted ServiceS, pp 35–51

Ring K (2000) EAI: making the right connections. Ovum Reports, Boston

Ruh WA, Maginnis FX, Brown WJ (2000) Enterprise application integration: a Wiley tech brief. Wiley, New York

Schulte S, Janiesch C, Venugopal S, Weber I, Hoenisch P (2015) Elastic business process management: state of the art and open challenges for BPM in the cloud. Fut Gen Comput Syst 46:36–50. doi:10.1016/j.future.2014.09.005

Serrano N, Hernantes J, Gallardo G (2014) Service-oriented architecture and legacy systems. IEEE Softw 31:15–19. doi:10.1109/MS.2014.125

Wortmann F, Flüchter K (2015) Internet of things: technology and value added. Bus Inf Syst Eng 57:221–224