MSC BUSINESS INFORMATION TECHNOLOGY

# From Cloud to Ground: Assessing Integration Platform-as-a-Service (iPaaS) Against On-Premise Integration Platforms

Wessel Rienstra

Supervision:

| | |
|---|---|
| João Luiz Rebelo Moreira | University of Twente |
| Hans Moonen | University of Twente |
| Niraj Punwani | Deloitte Nederland |
| Daan Noordzij | Deloitte Nederland |

May, 2024

University of Twente
Master Business Information Technology
Faculty of Electrical Engineering,
Mathematics and Computer Science

# Management Summary

As digitalisation progresses, the need for robust systems integration becomes increasingly crucial. Initially, organizations relied on on-premise integration platforms (TIP), licensed by vendors. With the advent of cloud computing, the Integration Platform-as-a-Service (iPaaS) emerged, offering a cloud-based alternative to TIP. Today, vendors offer both platforms, yet their respective advantages and disadvantages remain underexplored in literature. Furthermore, in practice, organizations struggle with the decision between TIP and iPaaS.

This research adopts a Design Science Research Methodology (DSRM) with a problem-centred approach to address these issues and answer the key research question:

*What are the functional and non-functional capabilities of integration platforms, and how do iPaaS (Integration Platform as a Service) and traditional integration platforms differ across these capabilities?*

The DSRM is applied in three phases over six steps. The first phase involves a systematic literature review on iPaaS in Chapter 3, identifying gaps and formulating six problems that establish the design requirements for this study. Based on these requirements, this research proposes a three-component design: a glossary to clarify terms, a Capability Model to understand essential integration platform capabilities, and a Non-Functional Capability (NFC) Tool to compare TIP and iPaaS based on non-functional aspects. Functional capabilities were excluded from the tool as they are too vendor-specific to apply to the general concepts of iPaas and TIP.

The first iteration of applying the methodology in Chapter 4 includes expert reviews and interviews to validate and refine the proposed capabilities, leading to the NFC Tool 1.0 and Capability Model 1.0 development. These tools were further refined in a second iteration in Chapter 5 through additional expert reviews and a case study comparing the iPaaS and TIP offerings of two vendors, Microsoft and SAP. This iterative process resulted in a robust comparison of benefits and drawbacks associated with each platform type.

The findings indicate a general preference for iPaaS over TIP due to its functional superiority and reduced maintenance burden, leading to an 'iPaaS unless' decision scenario. However, reasons such as regulatory compliance and control over infrastructure may lead organisations to opt against iPaaS. In comparing TIP and iPaaS, the focus of this research is directed to less complex integration scenarios including organizations that currently do not have an integration platform, or that have only one integration platform implemented. More complicated, multi-platform environments, which involve a mix of TIP and iPaaS from multiple vendors, are acknowledged but are also recommended as a focus for future research.

In conclusion, this research fills a significant gap by comparing the benefits and drawbacks of iPaaS and TIP and provides practical tools for industry practitioners. The Capability Model and NFC Tool facilitate a thorough comparison of integration platforms, assisting in formulating client requirements. Additionally, the glossary clarifies the distinctions between TIP and iPaaS, serving as a valuable reference for future research.

# Preface

This thesis, titled " From Cloud to Ground: Assessing Integration Platform-as-a-Service (iPaaS) Against On-Premise Integration Platforms", represents the completion of my studies in Business Information Technology at the University of Twente. Completed in partnership with Deloitte Netherlands between September 2023 and May 2024, this work delves into the field of integration platforms.

The journey of crafting this thesis has been challenging yet profoundly rewarding, thanks to the support of numerous individuals. I extend my gratitude to my academic supervisors at the University of Twente, João Luiz Rebelo Moreira and Hans Moonen. Their expert guidance and constructive feedback have been pivotal in elevating the scientific rigor of my research, and their supportive supervision made the experience enjoyable.

I am also appreciative of the support provided by Daan Noordzij, Niraj Punwani and Sophie van Kaam during my tenure at Deloitte. Their readiness to assist during challenging moments and their help in structuring my thoughts were invaluable. I am thankful to the Tech Vision & Architecture team at Deloitte for fostering a stimulating and supportive work environment, and to all the interviewees and case study participants whose cooperation was crucial for this research.

A special thanks goes to my family, whose unwavering support has been my cornerstone. I am particularly grateful to my father, whose inspiration is a constant source of strength, and to my mother, whose positivity continues to empower me.

I invite you to explore this thesis and hope it provides both insight and enjoyment.

Wessel Rienstra

# Contents

# List of Tables

# List of Figures

# List of Abbreviations

| Abbreviation | Meaning |
|---|---|
| **API** | Application Programming Interface |
| **BAM** | Business Activity Monitoring |
| **BPM** | Business Process Management |
| **CRM** | Customer Relationship Management |
| **CSV** | Comma-Separate Values |
| **DSRM** | Design Science Research Methodology |
| **EAI** | Enterprise Application Integration |
| **EDI** | Electronic Data Interchange |
| **ERP** | Enterprise Resource Planning |
| **ESB** | Enterprise Service Bus |
| **ETL** | Extract Transfer Load |
| **FTP** | File Transfer Protocol |
| **HTTP** | Hypertext Transfer Protocol |
| **Azure IS** | Azure Integration Services |
| **ISO** | International Organization of Standardization |
| **IT** | Information Technology |
| **JSON** | JavaScript Object Notation |
| **NFC** | Non-Functional Capability |
| **REST** | Representational state transfer |
| **SCM** | Supply Chain Management |
| **SLA** | Service Level Agreement |
| **SLR** | Systematic Literature Review |
| **SME** | Small and Medium-sized Enterprises |
| **SOA** | Service-Oriented Architecture |
| **SOAP** | Simple Object Access Protocol |
| **TIP** | Traditional Integration Platform |
| **VM** | Virtual Machine |
| **XML** | Extensible Markup Language |
| **XSD** | XML Schema Definition |
| **XSLT** | eXtensible Stylesheet Language |
| **FEDS** | Framework for Evaluation in Design Science |

# List of Definitions

| Concept | Definition used in this research |
|---|---|
| **Enterprise application integration** | The controlled sharing of data and business processes among any connected applications and data sources in the enterprise. |
| **Enterprise service bus** | An architectural pattern (rather than a specific technology) designed to support message-based, distributed integration through open standards. It aims to enable secure and reliable interactions between disparate applications by providing routing, invocation, and mediation services. |
| **Functional capability** | The abilities or features integration platforms possess to perform tasks, processes or functions. |
| **Integration platform** | Vendor-provided middleware enabling development, execution and governance of Integration flow connecting any combination of on-premises and cloud-based premises, services, applications and data within or across organizations. |
| **Integration platform-as-a-service** | An integration platform that is offered through a PaaS cloud service model. Therefore, the hosting is off-premise, deployment is in a public cloud and the management is externally organized. |
| **Non-functional capability** | The attributes or qualities of a system that define its operational performance. |
| **Traditional integration platform** | An integration platform for which the hosting is on-premise, deployment is on-premise and the management is internally organized. |
| **Vendor** | A company that sells integration platform technology and/or integration platform services to other organizations. |

# 1 Introduction

As the world is digitalizing at a fast pace, the number of systems required to be linked together grows and integration is becoming crucial for scalable and resilient businesses [40]. 80% of enterprise business operations leaders identify data and system integration critical to ongoing operations [17]. Despite its relevance, integration is still a big challenge amongst companies as "38% of respondents see the integration of siloed business software applications as their biggest challenge to digital transformation" [1]. The importance of integration can be argued from an intra and inter-organizational viewpoint, so within the organization, and beyond organizational borders.

Intra-organizational integration already "is a daunting task in the on-premise environment, where the full control of resources and services is usually assured" [54]. The introduction of cloud-based applications poses a whole new set of architectural and technological challenges as companies find themselves having a combination of on-premise legacy systems and cloud-based applications that should be able to exchange data [21]. In terms of inter-organizational integration, data and application integration allows companies to become part of a larger ecosystem in which they can cooperate with more companies to increase business value. Consequently, when the number of integrated actors and systems expands, the integrations' complexity increases and the importance of well-managed integrations grows [42].

In scientific literature, Enterprise Application Integration (EAI) is defined as the controlled sharing of data and business processes among any connected applications and data sources in the enterprise [52]. This discipline established itself somewhere around the year 2000. Till then, isolated applications, whether custom-built or off-the-shelf were connected via point-to-point integrations [21]. This approach resulted in complicated integration landscapes, also referred to as a 'spaghetti architecture' where adding new, or modifying existing integrations becomes increasingly difficult. EAI emerged as a solution, proposing to simplify the complicated application network by linking applications through a centralized, application-independent information broker', also referred to in this research as a Traditional Integration Platform (TIP). A TIP is installed, deployed and managed on-premise, by the company using the platform, and uses standardized connectors and defined integration logic to facilitate smoother inter-application communication [52]. Companies like Microsoft, Oracle and IBM are popular vendors that offer these integration platforms.

Currently, more than two decades later, technology is arguably having even more of an impact on businesses. Modern-day developments like the Internet of Things (IoT), Big Data and Cloud Computing are on companies' interest lists. In contrast, large applications like Customer Relationship Management (CRM), Supply Chain Management (SCM) and Enterprise Resource Planning (ERP) have already found their way into the organization. Within the EAI domain, an integration Platform-as-a-Service (iPaaS) emerged as a cloud-based alternative to existing on-premise integration platforms. Boomi launched the first iPaaS in 2008 named AtomSphere [13], but the term was further popularized by Gartner who publishes iPaaS-focused magic quadrant studies [33] in which they compare iPaaS offerings of different vendors. The concept of iPaaS promises both mature EAI functionalities and the benefits of SaaS applications [21]. It is used to facilitate on-premise to on-premise, on-premise to cloud and cloud-to-cloud integration [10]. In 2021, this iPaaS market was valued at 5.32 billion dollars and is expected to reach 61.67 billion dollars by 2023 [44]. It seems that iPaaS is a competitor to the more traditional EAI solutions and that companies face the challenge of selecting the right integration solution for their integration problem [11].

## 1.1 Problem Statement

Scientific literature has identified a lack of comparative analysis between iPaaS and TIP. Hyrynsalmi et al., (2021) concluded that "the adoption of new cloud-based integration platforms is in a maladjusted situation and that the choice between on-premises and cloud-based integration solutions

is not always clear" [42]. Ebert et al., (2017) proposed that future research should focus on “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” [21]. This research infers the following problem statement: There is insufficient knowledge of the advantages and disadvantages of iPaaS compared to TIP for organizations to make informed decisions between them.

This research came across a second problem while constructing the body of knowledge around the iPaaS concept. Current literature has adopted the following iPaaS definition:

*“a suite of cloud services enabling development, execution and governance of integration flows connecting any combination of on-premises and cloud-based premises, services, applications and data within the individual, or across multiple organizations.” [42]*

This research states that this definition is not specific enough to distinguish between TIP and iPaaS. What happens, for example, if a traditional integration platform is being hosted on a cloud machine? The platform's functionalities remain the same but now it becomes a cloud service and perhaps also an iPaaS. The definition does not provide conclusive answers to this. Therefore, to accurately compare iPaaS to TIP, it should be clear what is considered an iPaaS and what is not.

## 1.2 Research Goal

The research goal should logically follow from the problem description. The first goal is to understand what an iPaaS is and to be able to clearly explain when middleware should be referred to as iPaaS and when it should be referred to as a TIP. Only then can this research compare the two.

The second goal of this research is to provide clarity into the benefits and drawbacks of iPaaS compared to traditional integration platforms. Current research has not yet put much effort into how this comparison could be structured. Ebert et al., (2017) state that “research on iPaaS is rare, many interesting business and technical issues on the platforms and use cases have not been addressed” [21]. The distinction between business and technical aspects is a starting point for this comparison. These aspects can be translated into functional capabilities (technical aspects) that describe what functions an integration platform is supposed to perform, and non-functional capabilities (business aspects), the quality attributes of the integration platform [77].

Therefore, this research develops an integration platform capability model that includes these capabilities. This model itself is not necessarily the primary goal of this research. Instead, the goal is to use that model to make the comparison between iPaaS and TIP. It is important to note that this will not be a decision model because that would be too complex due to the large amount of scenario-specific aspects that play a role in this decision. This research primarily aims to shine as much light as possible on the different aspects that might play a role in this decision.

## 1.3 Research Questions

Based on this objective, the main research question of the thesis is formulated as:

*What are the functional and non-functional capabilities of integration platforms, and how do iPaaS (Integration Platform as a Service) and traditional integration platforms differ across these capabilities?*

The research question is divided into multiple sub-research questions. The answers to these sub-research questions combined provide an answer to the main research questions.

*Sub-research questions:*

1. How to distinguish between iPaaS platforms and traditional integration platforms?
    a. What is a traditional integration platform?
    b. What is an iPaaS?

c. How do these concepts differ from each other?

2. What are the capabilities of integration platforms?
    a. Which capabilities for integration platforms are described in the literature?
    b. Which capabilities for integration platforms can be identified by expert validation?

3. What are the use cases of organizations that have to select between a traditional integration platform and iPaaS?

4. How can the capability model be used to compare iPaaS and traditional integration platforms?

## 1.4 Research Methodology

Since this research aims to design and demonstrate a capability model for integration platforms, it adopts the Design Science Research Methodology (DSRM) proposed by Pfeffers et al., (2007) to answer the research questions. This methodology is developed for "carrying out design science research in information systems" [66].

Figure 1 outlines the six steps of this methodology, the different activities that this research performs within each step and the iterations made in this research. A detailed description of these steps is given below where the specific activities listed in the figure are marked *italic.*

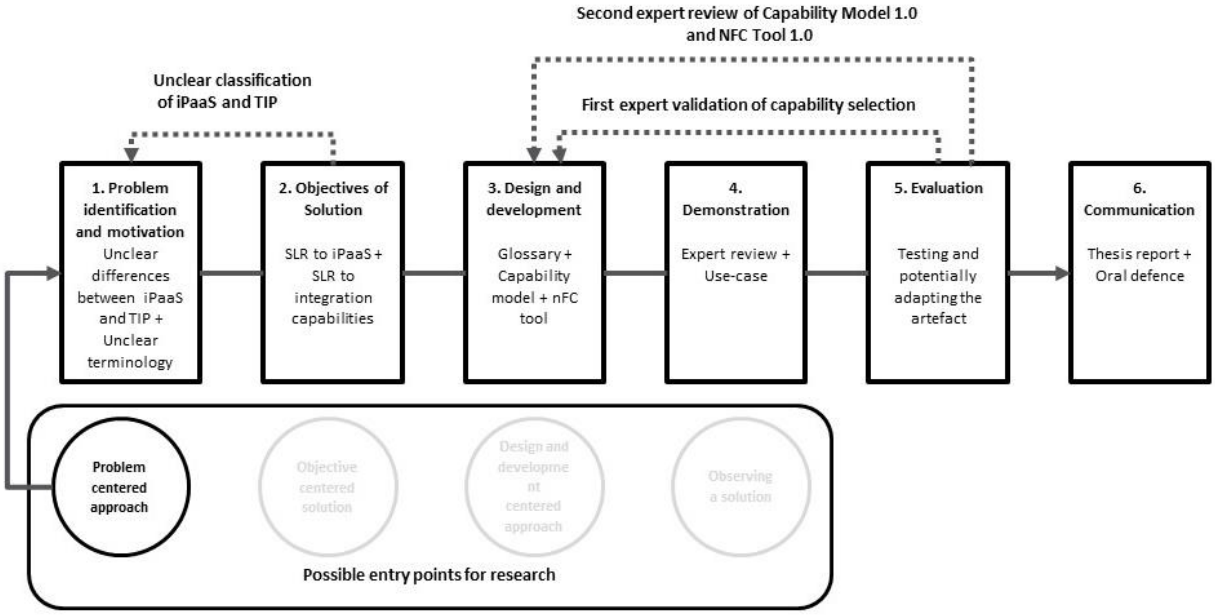



*Figure 1: Design science research methodology[66]*

**Step 1: Problem identification and motivation.** Pfeffers et al., identify four possible entry points to start the research. A problem-centred approach is suggested when the idea for the research results from suggested future research in a paper [66]. Current research proposed future work into the concept of *iPaaS in comparison with TIP*. Section 1.1 further elaborates on these papers. To increase relevance and justify the value of a solution, expert interviews were conducted. These interviews are further described in section 4.3. Nine out of ten interviewees confirmed the need for further research into this direction.

*Unclear terminology* of the concept of iPaaS was identified as a second problem during the Systematic Literature Review in step 2. To be able to compare iPaaS with TIP this research should first make clear what these concepts refer to. This problem is further described in the section 1.1.

**Step 2: Objectives of solution.** "Infer the objectives of a solution from the problem definition and knowledge of what is possible and feasible" [66]. This step includes acquiring "knowledge of the state of the problems" [66]. To do this, this research conducts a *Systematic Literature Review (SLR)*, that follows the methodology of Okoli [65], to the concept of iPaaS to summarize its current body of knowledge. The review identifies missing knowledge in current research on what the term iPaaS refers to as well as its benefits and drawbacks compared to traditional middleware. The review finds several problems that are translated into design requirements. The process and findings of the review are described in Chapter 3

To understand what is possible and feasible in terms of the objectives for this research, a second SLR following the methodology of Okoli [65] is performed that discover current models that include functional and non-functional capabilities related to integration platforms. This review is described in the section 3.1.2 and confirms the feasibility of developing and applying a capability model to compare iPaaS and TIP.

A full description of the objectives of this research is described in section 1.2. It is important to mention here that developing the capability model is not necessarily the main goal of this research. This model is only a means to be used to compare the two types of integration platforms.

**Step 3: Design and development.** This research first *develops a glossary* by synthesizing information from literature on iPaaS with background literature on the domain of EAI. This glossary is described in the section 4.2 and is used for the remainder of this research to distinguish between iPaaS and TIP.

The main artefact of this research is the *capability model.* The selection of capabilities is constructed by synthesizing the functional and non-functional capabilities described in current models (identified in the SLR to integration capabilities). This process is further described in section 3.1.2. "Because design is inherently an iterative and incremental activity, the evaluation phase provides essential feedback to the construction phase as to the quality of the design process and the design product under development" [34]. Therefore, the *first iteration of this capability selection is performed through expert interviews.* This process is described in section 4.3 and resulted in the first version of the capability model (section 4.4).

Lastly, the concept of iPaaS and TIP are further compared on the non-functional capabilities. This comparison is formulated into a Non-Functional Capability (NFC) tool based on expert interviews and background knowledge to show the different aspects on which iPaaS and TIP differ. This process and the results are described in section 4.5.

Table 1 shows the criteria that are defined for the NFC tool. These criteria are used in the demonstration and evaluation step of this research.

| Criteria of the artefacts | Description |
|---|---|
| **Understandability** | The information in the artefact is understandable assuming that the reader has a basic understanding of integration concepts |
| **Sufficiency** | The artefact contains all necessary information sufficiently |
| **Accuracy** | The information in the artefact is correct |
| **Usefulness** | The artefact is useful to compare iPaaS against TIP |
| **Usability** | The artefact is easy to use |

*Table 1: Design criteria of the artefacts*

**Step 4 and 5: Demonstration and evaluation.** The capability model and the NFC tool are demonstrated and evaluated in a *second iteration using a mixed-method validation approach* derived

from the Framework of Evaluation in Design Science Research (FEDS) [87]. Recall that the objective of this research is to use the capability model to compare iPaaS and TIP.

Firstly, the NFC Tool and the Capability Model are demonstrated and evaluated through *expert reviews*. This process includes identifying relevant interview participants, developing a review protocol including evaluation metrics, using a method for qualitative data analysis of the interview and analysing the results on the evaluation criteria. The process is described in detail in section 5.2.

Secondly, the capability model is applied in a case study. Together with an integration expert who knows specific vendors' iPaaS and TIP offerings, the model is used to support and structure the process of identifying how these two different platforms compare on their functional capabilities. The case study description, the findings and the evaluation are described in section 5.5

**Step 6: Communication.** The different steps described in this methodology and their findings are described in this report and presented in a thesis defence. This step concludes the research and finalizes the DSRM.

## 1.5 Thesis Outline

The approach of this thesis consists of three general phases: (1) theoretical background, (2) design and development, and (3) demonstration and evaluation. Figure 2 presents the different steps within these phases, the section where this step can be found and the sub-research question it addresses.

Chapter 2 and Chapter 3 cover the *theoretical background* in three building blocks. Background information on the domain of EAI to contextualize the domain in which integration platforms exist. A systematic literature review (SLR) towards iPaaS is conducted to build the theoretical framework on iPaaS but also to show that current literature does not sufficiently describe iPaaS, let alone understand its benefits and drawbacks compared to TIP. A second literature review is performed to identify models covering capabilities of integration platforms.

Chapter 4 covers the *design and development* phase, which first presents the glossary. The glossary is designed based on the information listed in the theoretical background and used to clearly distinguish between iPaaS and TIP. Secondly, this research synthesizes the different capabilities identified in the SLR into one capability model. This model is validated through expert reviews.

The model consists of functional and non-functional capabilities. The non-functional capabilities are further compared for iPaaS and TIP in the NFC Tool. This tool aids stakeholders, for example, Information Technology (IT) managers in understanding how iPaaS and TIP differ in their non-functional capabilities.

Chapter 5 covers the *demonstration and evaluation* phase. First, expert reviews are conducted to validate the NFC Tool. A review protocol describes the different steps of the interview as well as the evaluation metrics. The review session is also used to validate the capability model. The capability model is applied and evaluated in a use case where it functions as a guiding tool to compare the iPaaS and TIP integration platforms of specific vendors.

Chapter 6 discusses the findings of this research and concludes it by answering the research questions, describing the practical and theoretical contribution and formulating directions for future work.

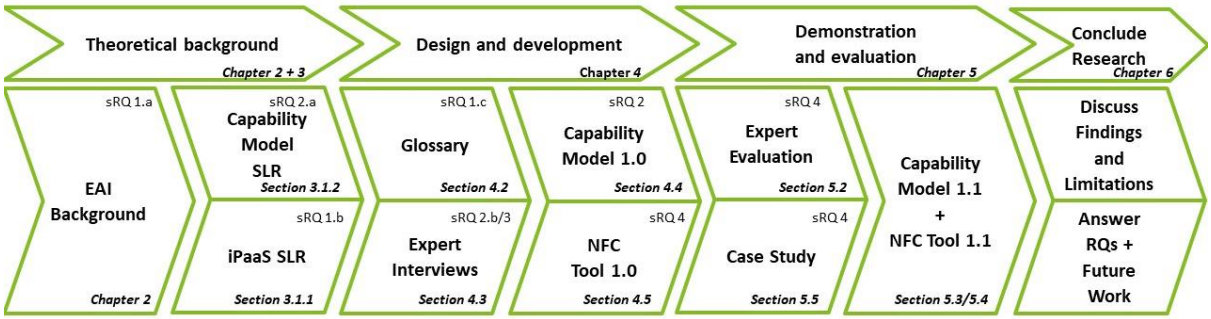



*Figure 2: Thesis Structure Overview*

# 2 Background Enterprise Application Integration

This chapter covers the domain of Enterprise Application Integration (EAI). Section 2.1 defines EAI and explains some terms that are relevant for the remainder of the chapter. Section 2.2 discusses how different EAI styles evolved. Section 2.3 outlines the components that make up for an integration solution. Section 2.4 introduces the different architectures used to develop solutions. The emergence of cloud computing is discussed in section 2.5 Lastly, section 2.6 shows how traditional integration platforms and iPaaS coexist at a vendor.

## 2.1 Definition of EAI

Enterprise Application Integration (EAI) is defined as "the controlled sharing of data and business processes among any connected applications and data sources in the enterprise" [52]. This is realized by building an integration solution that connects applications and data. However, integration is a difficult task as explained by [37] through these four challenges:

- **Networks are unreliable**: Compared to processes running on a single computer, an integration solution has to transport data across the networks between computers. The steps required to do so can cause delays or interruptions.
- **Networks are slow**: "Sending data across a network is multiple orders of magnitude slower than making a local method call" [37].
- **Any two applications are different**: The systems for which the integration solution is designed can use different programming languages, operating platforms and data formats. The solution must be able to interface between them.
- **Change is inevitable**: The applications change over time and an integration solution has to keep up with these changes.

Table 2 displays the four dimensions of EAI. The concept of an integration Platform-as-a-Service (iPaaS) exists in the data and application interface dimensions [21]. Therefore, the remainder of this chapter refers to those two dimensions when considering integration.

| | Dimension | Definition | Example |
|---|---|---|---|
| | User interface | "Bundle applications by using their user interfaces as a common point of integration (also known as screen scraping)" [52]. | Mainframe applications that do not provide database or business process level access |
| | Method | "Sharing of the business logic that may exist within the enterprise" [52]. | Application servers, TP (transaction processing) monitors, and frameworks. |
| **iPaaS focus** | Application interface | Leverage interfaces of applications to access both business processes and simple information which are used to bundle the applications together [52]. | Message broker |
| | Data | "Process, techniques and technology of moving data between data stores" [52]. | Extract Transfer Load (ETL) |

*Table 2: Examples of the four dimensions of EAI [52]*

Several concepts are introduced to assist in explaining the domain of EAI.

**Protocol**: A protocol is a set of rules and regulations that allow two IT components to communicate. Well-known protocols are the Hypertext Transfer Protocol (HTTP) used on the World Wide Web and the Simple Object Access Protocol (SOAP) protocol which provides rules to pass information between applications in an Extensible Markup Language (XML) format. Protocols are layered, the SOAP protocol can use the HTTP protocol to send the messages. A protocol that is also worth

mentioning is the Electronic Data Interchange (EDI) protocol. It facilitates “the intercompany communication of business documents in a standard format” [89]. The protocol includes a message standard which allows two companies to directly exchange messages. Because of this, EDI had a great effect on the Business-to-Business communication [89].

**Asynchronous vs synchronous communication:** Applications communicate either synchronously or asynchronously (Figure 3**)**. The former implies that “the calling process is halted while the sub-process is executing a function” [37]. It can be compared to a phone call between two people where one person listens while the other is talking. Asynchronous communication uses a send-and-forget approach that “allows the process to continue to execute after it sends the message” [37]. Asynchronous has some advantages, the caller can perform other processes in the meantime instead of waiting for the results. The processes can also be executed in any order but this does require the processes to be able to run independently from each other [37].

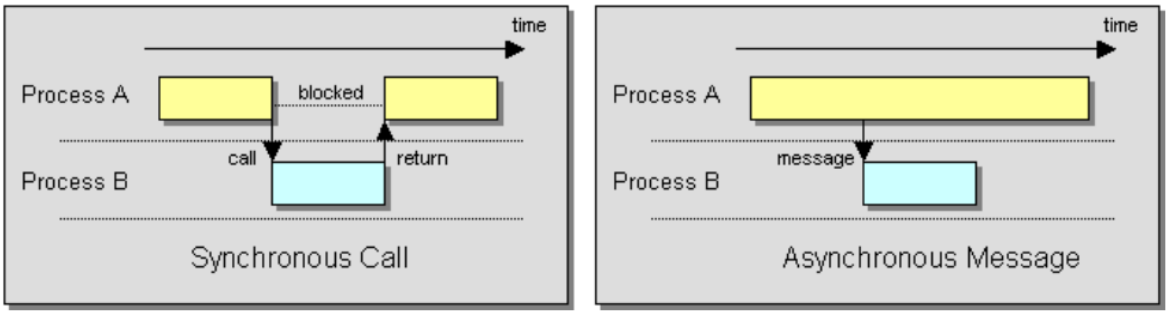



*Figure 3: Synchronous and asynchronous messaging[37]*

## 2.2 Integration Styles

Table 3 outlines the four main approaches that are used to realize an integration solution. Each style was developed in response to its predecessor with messaging being the latest development. In an application-integration solution, multiple styles of integration can be used so that each point of integration takes advantage of the style that is best suited for the problem [37]. Appendix A displays a summary of the benefits and drawbacks of each style.

| Integration style | Description |
|---|---|
| File Transfer | One application writes a file that another later reads. The applications need to agree on the filename and location, the format of the file, the timing of when it will be written and read, and who will delete the file. |
| Shared Database | Multiple applications share the same database schema, located in a single physical database. Because there is no duplicate data storage, no data has to be transferred from one application to the other. |
| Remote Call Invocation (RCI) | One application exposes some of its functionality so that it can be accessed remotely by other applications as a remote procedure. The communication occurs in real-time and synchronously. |
| Messaging | One application publishes a message to a common message channel. Other applications can read the message from the channel at a later time. The applications must agree on a channel as well as the format of the message |

*Table 3: Integration styles [37]*

The section on the Application Programming Interface (API Management Platform describes how Remote Call Invocation is practically implemented in APIs.

## 2.3 Components of an Integration Solution

Using one of the integration styles described in the previous section, integration solutions can be created between two or more applications. To do this, "a number of things have to happen. These things make up what we call middleware – the things that sit between applications" [37]. In literature, the term middleware is also used to describe a software tool used to assist programmers in building integration [8]. However, this term neglects the fact that integrations can be programmed directly without the need for a software tool. The decision of integration style determines the type of middleware that is used: Messaging uses message-oriented middleware whereas RPI uses procedure-oriented middleware. Figure 4 displays the basic elements of an integration solution.

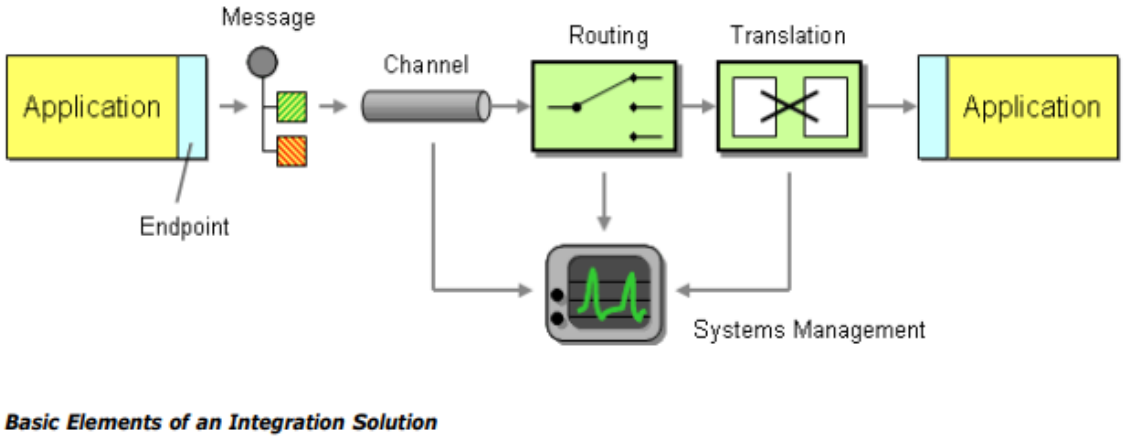



*Figure 4: Abstract process of an integration solution[37]*

A **message** is "an atomic packet of data that can be transmitted on a channel" [37]. It contains of two parts: the body with the actual information of the message and the header with meta-data about the message. A message is represented according to a certain structure called a schema. Examples of schema types are Extensible Markup Language (XML), JavaScript Object Notation (JSON) and Comma-Separate Values (CSV).

Most applications are not prepared to participate in an integration solution. Therefore, an application needs an **endpoint** that allows the application to send and receive messages from other applications [37]. An example of an endpoint is an API which "defines the contract of a software component in terms of the protocol, data format, and the endpoint for two computer applications to communicate with each other over a network" [19]. More information on APIs is provided in the section on the API Management Platform.

**Channels** are the virtual pipes used to connect applications. Two familiar types of channels are the point-to-point channel which ensures only one receiver of a given message, and a publish-subscribe channel allowing multiple receivers to each receive a copy of the message.

If two or more applications are included in the integration solution, the middleware should know to which applications a specific message should be sent. The **routing** component is responsible for orchestrating this. An overview of different types of routers is provided by [37].

Because the internal data format of an application can often not be changed the middleware needs to provide some mechanism to convert one application's data format into the other's, this is called **translation** or transformation. Visual drag-and-drop tools have been developed to aid in transformation. However, these "can become a liability when it comes to debugging or when you need to create complex solutions"[37].

Finally, there should be an overall **systems management function** that monitors the entire integration solution and reports to a central location.

## 2.4 Integration Solution Architectures

The architectural style to implement integration solutions evolved. From pre-mature point-to-point and hub-and-spoke architectures towards more commonly used Enterprise Service Bus (ESB) and API management platform approaches. Be aware that these are broad, architectural concepts in a complicated integration landscape. Even though they are presented as separate concepts, their real-world implementation is complex and multi-modal.

### 2.4.1 Point-to-point

In point-to-point communication architecture, applications interact directly without middleware, as depicted in Figure 5. Integration logic is embedded within the application which requires custom development for each integration. This approach compromises scalability as the number of applications grows. However, for a small number of applications, this approach proves straightforward [50].

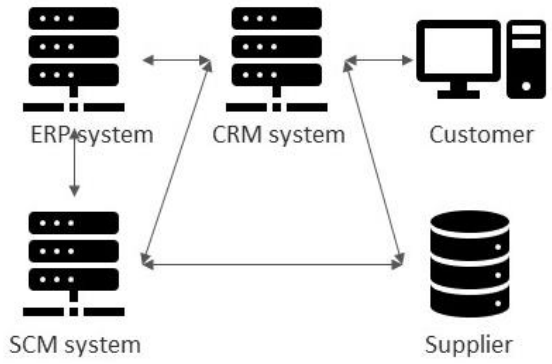



*Figure 5: A point-to-point integration architecture*

### 2.4.2 Hub-and-spoke

The hub-and-spoke model, or message broker [58], centralizes integration logic in a hub, connecting systems (spokes) via bespoke connectors [50], as shown in Figure 6. Its key advantage is fewer necessary connections for integrating multiple systems. However, the hub represents a single point of failure; any malfunction disrupts all integrations, risking total infrastructure collapse. Moreover, the hub's reliance on proprietary protocols and platform-specific interfaces [58] forces organizations into a difficult choice: risk overloading the hub with new applications or creating separate hubs, leading to a fragmented, "islands of hubs" scenario within the organization [58].

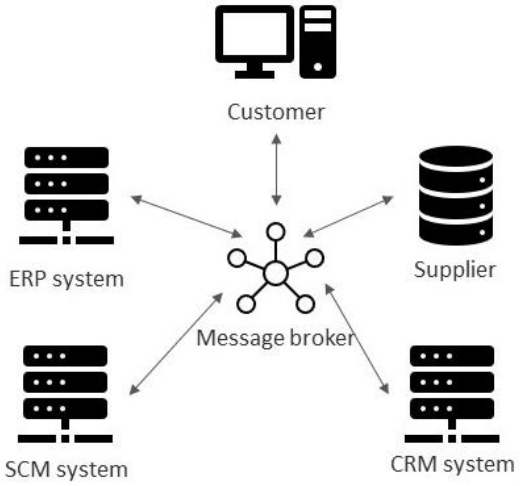



*Figure 6: A hub-and-spoke architecture*

### 2.4.3 Enterprise Service Bus

The concept of the ESB emerged to address the "island problem" seen with Message-Oriented Middleware (MOM) and to support the rise of Service-Oriented Architecture (SOA), which significantly impacted the IT industry in the mid-2000s [78]. SOA is based on the principle that applications should offer their functionalities as reusable services, which are self-contained, stateless business functions accessible via a standardized interface, regardless of the implementation [58]. Services are software components defined by metadata understandable by other programs, encompassing a service implementation (the executing code), a service contract (service parameters), and a service interface (communication protocols and interaction with other services) [6]. In SOA, business processes are designed by orchestrating multiple services based on their metadata, offering advantages like improved scalability, decoupling, and testing and deployment management over traditional monolithic architectures [78]. SOA can be implemented with service-based technologies, typically using web services like SOAP or Representational state transfer (REST) [56]. An ESB is viewed as a practical application of SOA principles to integration, acting as a connecting middleware [15].

Defining an ESB precisely is challenging, as there's no agreed-upon definition and set of features of an ESB. Commercial vendors contribute to this confusion by arbitrarily labelling their products as ESBs [58]. Essentially, an ESB is an architectural pattern rather than a specific technology, designed to support message-based, distributed integration through open standards. It aims to enable secure and reliable interactions between disparate applications by providing routing, invocation, and mediation services [58]. As depicted in Figure 7, an ESB's infrastructure is distributed, consisting of server clusters or hubs that host integration services. While small implementations may have a single hub, the architecture allows for adding more hubs to scale up the system, with the entire network centrally managed for better control and configuration

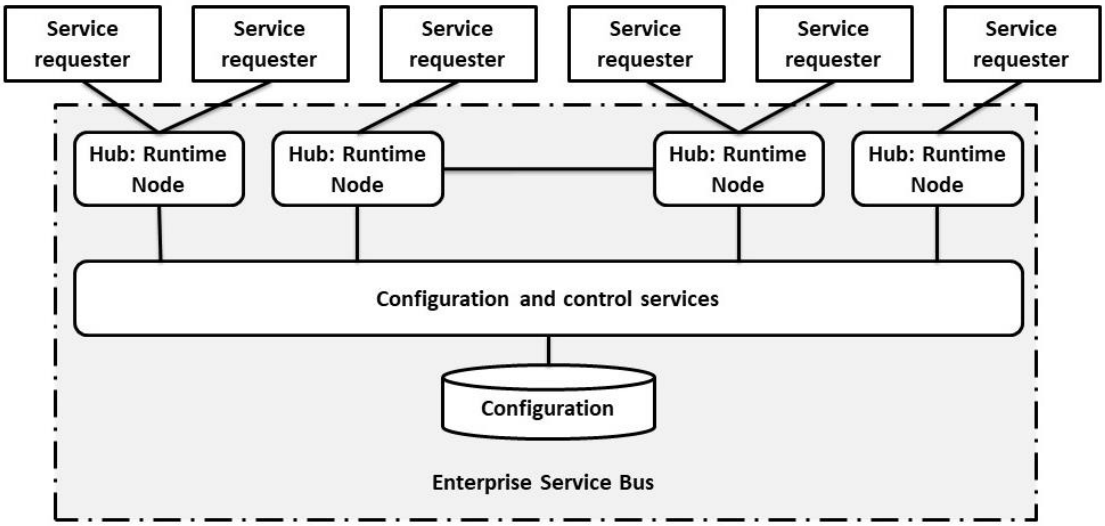



*Figure 7: The ESB as a distributed infrastructure with centralised control [48]*

### 2.4.4 API Management Platform

Where ESB took full "benefit" of SOA, the value of APIs increased with the rise of microservices, a specific type of SOA style. Recall from the previous section that SOA merely means that applications provide their business functionality in the form of reusable services. Hence "more guidelines became necessary in order to achieve performance and fast deployment goals" [75] which resulted in the development of microservices. The microservice architectural style is defined as "an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms , often an HTTP resource API" [26]. This statement shows how API's benefitted from the rise of microservices. Microservices should be thought of "as a specific approach for SOA in the same way that XP or Scrum are specific approaches for Agile software development" [63].

An API defines the contract of a software component in terms of the protocol, data format, and the endpoint for two computer applications to communicate with each other over a network". [19]. In simple terms, it contains the requirements that describe how two applications can communicate. An API can be designed according to several styles including RCI, REST and SOAP [19]. An API provider should provide the following information for a consumer to consume the information of the API [19]:

- The functionality provided by the API
- The URL which can be used to access the API
- The input and output parameters (names, message formats, data types)
- The Service-Level Agreement that the API provider adheres to
- The technical requirements for the rate limits
- Documentation to improve the understanding of the API

APIs need to be managed using an API management platform. "An API management platform enables you to create, analyze, and manage APIs in a secure and scalable environment" [19]. Unfortunately, there is not much research published on the API management platforms. The book of De B. [19] states that the platform should provide capabilities in three types of services

1. **API gateway services:** These services allow the user to "create and manage APIs from existing data and services" [19]. The capabilities are surprisingly similar to the ESB and

include security, data and protocol transformation, routing and orchestration. The way these capabilities are described is also similar to the description of the ESB. An API gateway acts as a single entry point into the system. It is amongst other elements an integral part of the microservice architecture because it allows to management of the different microservices in one central point [29].

2. **Analytics services**: These services provide relevant business metrics and monitor traffic from applications. The platform "should be able to extract and log custom variables from within the message payload for advanced analytics reporting" [19]. Furthermore, "user auditing can help the API administrator review historical information to analyze who accesses an API, when it is accessed, how it is used, and how many calls are made from the various consumers of the API" [19]. Lastly, the platform should monitor if the Service Level Agreement (SLA) conditions are met.
3. **Developer services:** This service should provide the following capabilities. Firstly, developer support which includes correct API documentation and test and development kits to make it easier for developers to adopt APIs. Secondly, community management allows developers to see how other developers are doing—lastly, API catalogue and documentation where API can be publicized [19].

## 2.5 Cloud Computing

Cloud computing is defined as "on-demand access, via the internet, to computing resources, applications, servers (physical servers and virtual servers), data storage, development tools, networking capabilities, and more hosted at a remote data centre managed by a vendor" [88]. Cloud computing technology is at the origin of iPaaS and has a great influence on the IT landscape of today's businesses.

### 2.5.1 Cloud Computing Origin

Before cloud computing and virtualization, organizations managed their entire IT infrastructure (see a list of infrastructure components in Figure 11), including costly servers that hosted multiple applications without isolation, leading to inefficiencies and vulnerabilities to outages [31]. Operating systems were directly installed on the hardware, and applications shared the same system without providing physical or virtual isolation as depicted in Figure 8. This complicated application management and server utilization because applications could not easily be transferred from one server to another.

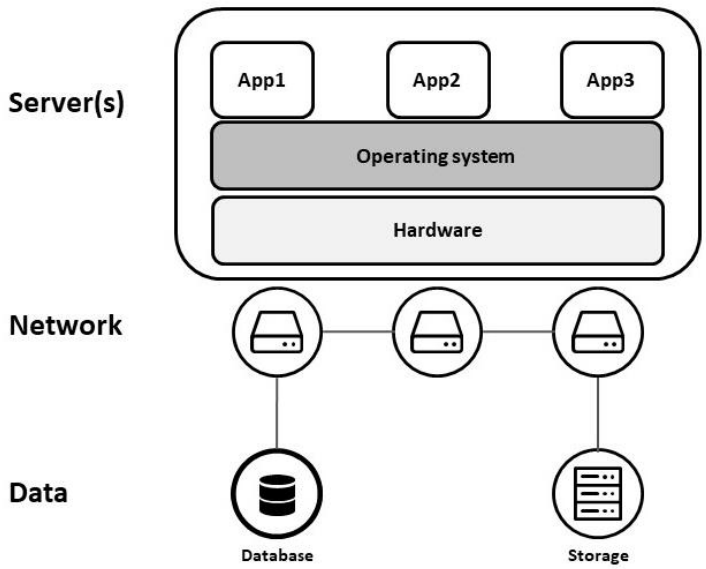



*Figure 8: Servers without virtualisation [31]*

Virtualization provided a solution for this ineffective utilization of resources. “Virtualization uses software to create an abstraction layer over computer hardware that allows the hardware elements of a single computer: processors, memory, storage and more to be divided into multiple virtual computers, commonly called virtual machines (VMs)” [88]. This is important because it “isolates software from hardware and so provides a mechanism to quickly reallocate applications across servers based on computational demands” [31]. This technology improves resource utilization, scalability and flexibility of the IT infrastructure. Figure 9 shows a simplified version of how the IT architecture would look like with virtualized servers.

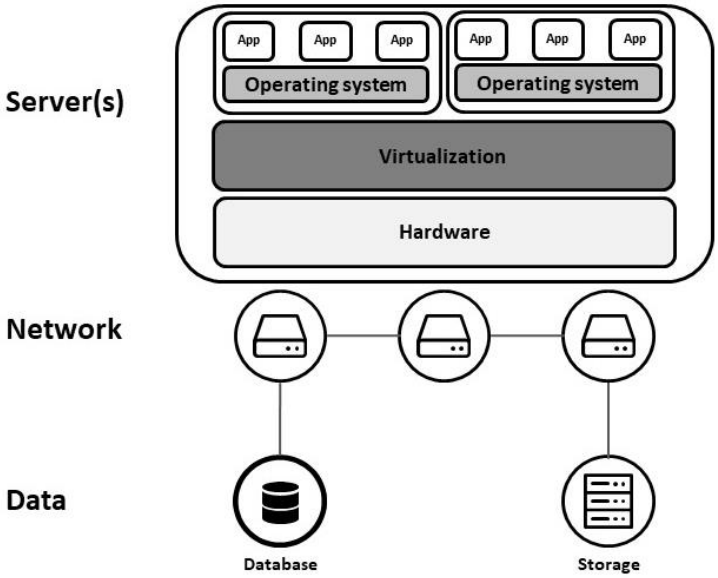



*Figure 9: Virtualized servers [31]*

Virtualization optimized IT architecture, but still within a setup managed by an organization's system administrators. Transitioning to cloud computing requires a service layer and on-demand provision of resources [31] that “enables vendors to serve users with their existing physical computer hardware. It enables cloud users to purchase only the computing resources they need when they need it and to scale those resources cost-effectively as their workloads grow” [88]. This service layer hides the complexity of the infrastructure from the user and presents it with a cloud-management interface. Figure 10 shows how this management layer, depending on the specific implementation, can be accessed through a dashboard, APIs, Web Services and other specific services. The figure is a simplified version visualising that the blue parts are now provided by and under the responsibility of a cloud provider. Section 2.5.3 further explains that this specific set of hardware and software falling under the vendor's responsibility differs per cloud service model.

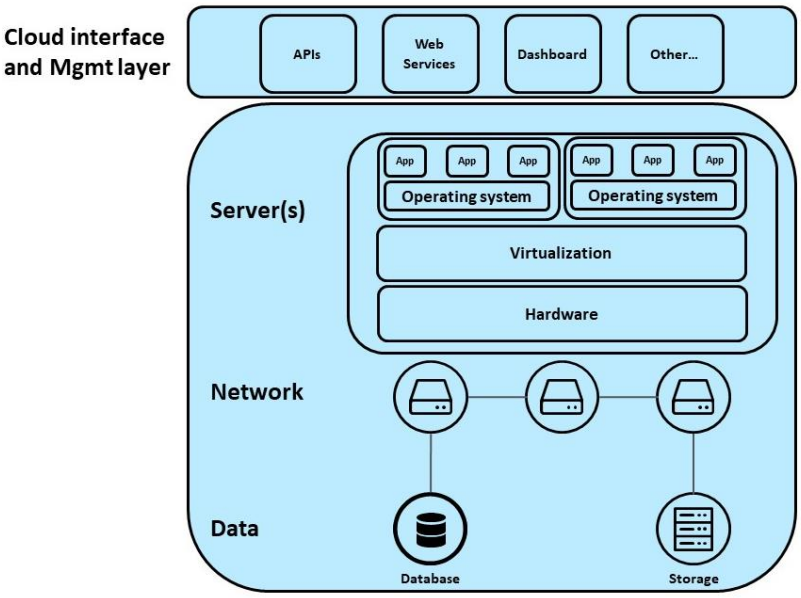



*Figure 10: Simplified cloud infrastructure [31]*

### 2.5.2 Cloud Migration Incentives

Organizations that adopt cloud computing services enjoy scalability and flexibility, allowing them to dynamically adjust the required resources to meet (changes in) demand [56]. Furthermore, cloud computing allows for cost efficiency through the pay-as-you-go model, eliminating the need for significant upfront investments in hardware and software as shown in Figure 11. This reduced responsibility not only eliminates upfront investment but also reduces the costs that would normally be incurred in maintaining these different hardware and software components.

Cloud computing also allows organizations to only pay for the resources they use, leading to potential cost savings compared to software packages that need to be bought in their entirety, even though an organization might only want to use a specific subset of the package [55]. The cloud enables unparalleled accessibility and collaboration, allowing users to access data and applications from anywhere, fostering productivity and flexibility in today's increasingly remote work environment [84]. Additionally, the cloud accelerates innovation and agility, providing consumers with the tools to rapidly deploy and test new applications without substantial investment, thereby lowering the barrier to innovation [22].

Vendors had several reasons to start using and offering cloud computing products. Section 2.5.1 explained that the introduction of virtualization was the big realisation of cloud computing. Besides that, vendors desired the shift from traditional upfront purchase models to subscription-based revenue models. The move to a service-oriented model emphasizes recurring revenue over one-time sales, promoting customer retention and long-term relationships [3]. Also, the vendor now takes responsibility for a large part of the infrastructure that would normally be the responsibility of an individual organization. Vendors can achieve economies of scale by organizing very large data centres as it is more cost-efficient to buy and configure hardware resources in larger volumes [31]. Also, the larger the data centre becomes, the easier it becomes to maximize the amount of work per dollar spent: Components can be shared more efficiently, server idle times are reduced and physical and virtual server density is improved [31].

### 2.5.3 Cloud Service Models

Section 2.5.1 explained that cloud computing involves a virtualized environment that is offered on-demand through a service layer to the customer. The service model describes how the service layer is made available to the customers [31]. The three most common models are SaaS, PaaS and IaaS [52].

"These service models may have synergies between each other and be interdependent - for example, PaaS is dependent on IaaS because application platforms require physical infrastructure" [31]. Figure 11 shows the differences in responsibilities for hardware and software between on-premise and the different service models.

*Software as a Service (SaaS)*
The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings. For example, a vendor might allow a company to use its logo inside a SaaS application [52].

*Platform as a Service (PaaS)*
The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment [52]. Therefore, the consumer can deploy applications without incurring the cost and complexity of buying and managing the underlying hardware, software and hosting capabilities [72]. "The difference in PaaS from SaaS is that SaaS only hosts completed cloud applications whereas PaaS Offer a development platform for both completed and in progress cloud application" [72]

*Infrastructure as a Service (IaaS)*
The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls) [52]. The advantage of IaaS is that it provides the ability to customers "to access a virtual server in few minutes and pay only for the resource they use" [72].

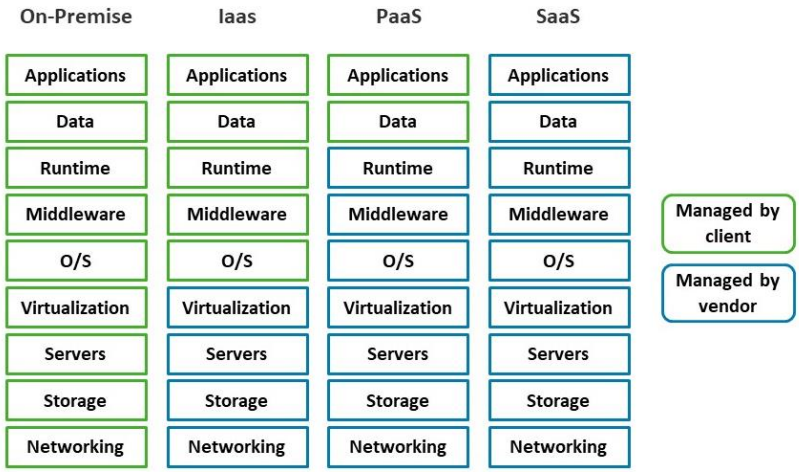



*Figure 11: On-premise, IaaS, PaaS and SaaS [73]*

### 2.5.4 Cloud Architectures
There are three different cloud computing architectures: Public cloud, private cloud and hybrid cloud. A public cloud "is a multi-tenant cloud environment, where the same computing resources are shared

among multiple customers" [90]. This is the standard cloud computing model [5]. In this case, the vendor owns and maintains the infrastructure just as stated in the cloud computing definition at the beginning of this section. The vendor then makes these resources available to the customers and businesses over the internet through virtualization [5]. "The customer has limited visibility into or control over where the computing infrastructure is hosted." [5]. The advantages of a public cloud are "greater scalability, lower cost of entry and faster access to the latest technologies" [90].

In contrast to public clouds, private clouds represent a distinct cloud computing model characterized by their single-tenant architecture, where resources are exclusively allocated to a single customer [90]. This differentiates private clouds from their public counterparts by offering dedicated services and infrastructure to one organization, enhancing privacy and control over the cloud environment. Private clouds can be hosted in various ways: on-premise by the organization, by the cloud vendor, or by a third party. When hosted on-premise, organizations assume full responsibility as depicted in Figure 11 which could limit the actual cloud advantages like scalability and flexibility. Alternatively, vendor-managed private clouds offer dedicated resources at a higher cost compared to public clouds [5]. Furthermore, management of this type of private cloud can vary, ranging from complete control by the organization to a shared responsibility model with the cloud service provider. An overview of the deployment, hosting and management options is given in the glossary presented in section 4.2.

Lastly, a hybrid cloud architecture "integrates public and private cloud infrastructures. In this model, the two types of cloud are joined together into a single, flexible infrastructure, and the enterprise can choose the optimal cloud environment for each individual application or workload" [90]. By leveraging a hybrid cloud, organizations can move workloads between cloud solutions as needs and costs fluctuate, providing a balance between on-premises infrastructure and the public cloud [16]. This is the most flexible infrastructure. However, the complexity increases because the cloud infrastructures that are combined, and their interfaces "should be competent enough while permitting to transfer data from one cloud to another" [30].

## 2.6 The Vendor

The previous sections in this chapter delved into EAI's theory, highlighting the shift towards messaging and the evolution of ESBs and API management platforms. The upcoming chapter introduces iPaaS, the latest advancement in EAI. In practice, these integration styles (section 2.2,) components (section 2.3) and architectures (section 2.4) are combined into a vendor's integration platform offerings. This section briefly outlines the current state of these vendors.

EAI is essentially a strategy for managing business applications [64], not a one-size-fits-all solution but rather achieved through various methods. Companies can build their middleware or adopt a third-party product, typically from leading IT vendors [50][64]. These vendors offer integration platforms, defined as "specialized software tools that help to design, implement, run and monitor integration solutions that orchestrate a set of applications to keep their data synchronized or to develop new functionalities on top of the current ones" [28]. Key players include IBM, Oracle, Microsoft, and Dell Boomi [64], providing comprehensive solutions that encompass all necessary EAI products.

Traditionally, these platforms were on-premise solutions that required an upfront license fee and client-side management (see Figure 11). The advent of cloud computing opened up opportunities for vendors to offer iPaaS, a cloud-based integration service, while still supporting traditional platforms. Microsoft with BizTalk and Azure Integration Services, IBM with the Integration Bus and App Connect, Oracle with SOA Suite and Integration Cloud Services, and SAP with Process Orchestration and Integration Suite exemplify this dual offering. This allows businesses to choose between cloud-based or on-premise solutions. This study aims to explore the differences between these platform types.

# 3 Literature Review

This chapter describes the literature review to the concept of integration Platform-as-a-Service (iPaaS) and the literature review to discover integration capabilities. Section 3.1 presents the method for both systematic literature review and their protocols. Section 3.2 presents and analyses the iPaaS definition. Section 3.3 introduces iPaaS components and functionalities as described in the literature. Section 3.4 describes the benefits and drawbacks of iPaaS as an integration platform. Lastly, section 3.5 concludes this chapter.

## 3.1 Method

Two systematic literature reviews are conducted in this thesis. Firstly, to construct the current body of knowledge around the concept of iPaaS. This exposes the state of problems that serve as input for the design recommendations of this research [66]. Secondly, to identify integration capabilities which reveal the theory that is operationalized in the solution design of this research, allowing this research to move from the objectives of the solution to the design and development phase [66].

For both reviews, the Systematic Literature Review (SLR) methodology of Okoli [65] is chosen as it is specifically developed for the information system research stream. “Information systems combines social science, business, and computing science, whose research methods are different from those of the health sciences”. The methodology consists of seven steps. Section 3.1.1 explains the steps for the SLR to iPaaS. Section 3.1.2 explains the steps for the SLR to discover integration capabilities.

### 3.1.1 Systematic Literature Review to iPaaS

#### 1. Identify the purpose

The iPaaS definition introduced in section 1.1 states that it can be used to create, manage and govern integration flows to connect applications, data and services across different organizations. These are essential components of any integration platform, and not just iPaaS. Therefore, this systematic literature review aims to answer research question 1.b by identifying the components, capabilities and features belonging to an iPaaS that enable it to create, manage and govern integration flows.

#### 2. Draft protocol and train the team

This research uses a research protocol to improve its replicability. This protocol is a plan that describes the conduct of the proposed systematic literature review [65]. This protocol is further described in Appendix B. Given the recent development of scientific content on the topic of iPaaS and the limited number of papers on iPaaS, articles that are not cited by others are included. This increases the number of sources and improves the generalizability of the results. Scopus is used as the database to query as it covers a wide variety of scientific articles. Training the team is not needed in this case as the SLR will be performed by a single person.

#### 3. Search for literature

The following search query is used resulting in 53 articles: “*TITLE-ABS-KEY( 'ipaas' OR 'integration platform as a service' OR 'integration-platform-as-a-service')”*.

#### 4. Practical screening and quality appraisal

The practical screening and quality appraisal of the articles is done according to the inclusion and exclusion criteria from Table 4.

| Inclusion criteria | Exclusion criteria |
|---|---|
| • Publicly available<br>• English or Dutch articles<br>• Discusses iPaaS | • Duplicates<br>• Medical papers (see description below)<br>• Different meaning of iPaaS |

*Table 4: Inclusion and Exclusion criteria*

Articles are considered publicly available if they can be accessed through the publishers' website, via a semantic scholar like Google Scholar or through the license of the University of Twente without the need to make a payment. Poorly written articles are assessed by checking spelling and grammar and assessing the referencing. Medical papers are excluded because they refer to another, medical definition of iPaaS. Finally, some papers referred to iPaaS as an "intelligent paging-as-a-service" or "input parameter analysis system" which are not relevant to this research. The process of applying these criteria is visualized in Appendix B.

### 5. Data extraction

The inclusion and exclusion criteria are used to extract the data and narrow the results. Applying the criteria removed 37 of the 53 included articles. Consequently, 16 papers are included for further analysis in the SLR. The complete extraction protocol is described in Appendix B

### 6. Synthesis of studies

Table 5 shows that the included papers in the SLR range from papers publicized in 2012 to 2023. The largest number of citations belongs to the paper of Ebert et al., (2017). Also, the earlier publicized papers generally have more citations than the more recent publications. It stands out that Hyrynsalmi, with four papers ([40][41][42][43]), is responsible for 25% of the total number of papers. A further description of the papers and the used methods is found in Appendix B. Notably, only a small number of papers validated their findings and some papers did not describe a research method.

| Ref | Author | Year | Citations |
|---|---|---|---|
| **[10]** | Bolloju and Murugesan | 2012 | 15 |
| **[54]** | Marian | 2012 | 21 |
| **[69]** | Potocnik | 2012 | 20 |
| **[68]** | Phan | 2013 | 22 |
| **[21]** | Ebert et al. | 2017 | 47 |
| **[85]** | Theilig | 2018 | 3 |
| **[83]** | Srimathi | 2019 | 3 |
| **[14]** | Cestari et al. | 2020 | 5 |
| **[93]** | Zhang and Yue | 2020 | 12 |
| **[42]** | Hyrynsalmi | 2021 | 3 |
| **[62]** | Neifer et al. | 2021 | 11 |
| **[27]** | Frantz et al. | 2021 | 16 |
| **[40]** | Hyrynsalmi | 2022 | 7 |
| **[41]** | Hyrynsalmi | 2022 | 1 |
| **[80]** | Sänger and Abeck | 2022 | 1 |
| **[43]** | Hyrynsalmi | 2023 | 0 |

*Table 5: Number of citations per paper*

### 7. Writing the review

The results of this review are part of a larger research project, which also encompasses a background information investigation. In this comprehensive research endeavour, the SLR is treated as an independent study. Consequently, the SLR findings are discussed in section 3.2 section 3.3 and section 3.4 and used alongside other research outcomes, collectively shaping the research's conclusions.

### 3.1.2 Systematic Literature Review to Integration Capabilities

#### 1. Identify the purpose

The goal of this review is to identify the functional and non-functional capabilities of integration platforms. These can function as a starting point for a structural model to describe integration platforms in the design and development phase of this research.

#### 2. Search

The following data sources were selected to extract information from in the subject fields of information systems and computer science: Scopus, Scholar and Gartner. The latter is a grey literature source which is included because of the limited availability of scientific sources that cover the technical capabilities of integration platforms.

The following keywords were used to discover sources: "(EAI OR enterprise application integration) framework", "(EAI OR enterprise application integration) model", "(EAI OR enterprise application integration) conceptual model", "SaaS model", "SaaS adoption", "Integration capabilities", "Integration platform evaluation".

An iterative strategy was used that resulted in this long list of keywords. It became clear the sources covering the functional capabilities were scarce and that the sources that did cover that part were frameworks. Regarding the non-functional capabilities, no models could be found that covered these for integration platforms specifically. Therefore, the decision was made to include models that covered the non-functional capabilities of Enterprise Application Integration (EAI) adoption and Cloud adoption.

#### 3. Practical screening and quality appraisal

The inclusion criteria are that sources can be of any document type (not only journals) that are written in Dutch or English and cover either functional or non-functional capabilities related to EAI, integration platforms or cloud adoption. By not adopting very strict inclusion criteria, this review aims to limit the risk of not including certain integration capabilities.

#### 4. Data extraction

Using the previously described search strategy, this literature search discovered three models that covered the functional capabilities of an integration platform, six models on non-functional capabilities influencing EAI adoption and four models on non-functional capabilities influencing SaaS adoption. Table 6 shows the included sources per category.

| Category | Ref | Author | Citations |
|---|---|---|---|
| **EAI models on functional capabilities** | [32] | Guttridge, Comes, Ray (2023) | 0 |
| | [60] | Bahreininejad and Moradi (2013) | 15 |
| | [70] | Pushmann and Alt (2001) | 106 |
| **Factors influencing EAI adoption** | [37] | Hung et al., (2015) | 32 |
| | [2] | Al-Balushi et al., (2015) | 8 |
| | [86] | Themistocleous (2004) | 167 |
| | [46] | Themistocleous and Kamal (2006) | 54 |
| | [4] | Aserey, N., & Alshawi, S. N. (2013) | 4 |
| | [53] | Mantzana, V., & Themistocleous, M. (2005) | 17 |
| **Factors influencing Cloud adoption** | [92] | Yang, Z et al., (2010). | 355 |
| | [79] | Safari, F., Safari, N., & Hasanzadeh, A. (2015) | 151 |
| | [36] | Hsu, C. L., & Lin, J. C. C. (2016) | 153 |
| | [73] | R. D. Raut et al., (2017) | 120 |

*Table 6: Included papers in the literature review towards conceptual models*

### 5. Synthesis of studies

The selection of models is synthesized to construct a first selection of functional and non-functional capabilities. Two different synthesizes are performed, one for the functional capabilities and one for the non-functional capabilities. The former included models from the category "EAI functional capabilities", and the latter included the models from the category "factors influencing EAI adoption" and "factors influencing cloud adoption" (see Table 6).

The synthesis is performed according to the following steps:

1. List all factors/capabilities and their description
2. Group factors/capabilities that are the same or have a similar description
3. If a factor is only mentioned in one paper, remove it from the list
4. Map the non-functional factors to the International Organization of Standardization (ISO) 25010 framework (step is based on [71])

*(Steps 3, and 4 are not for functional capabilities synthesis)*

Steps three and four do not apply to the synthesis of the functional capabilities because they include only three models so the decision is made to make the list as complete as possible.

Figure 12 shows a representation of how the discovered models are combined into one list.

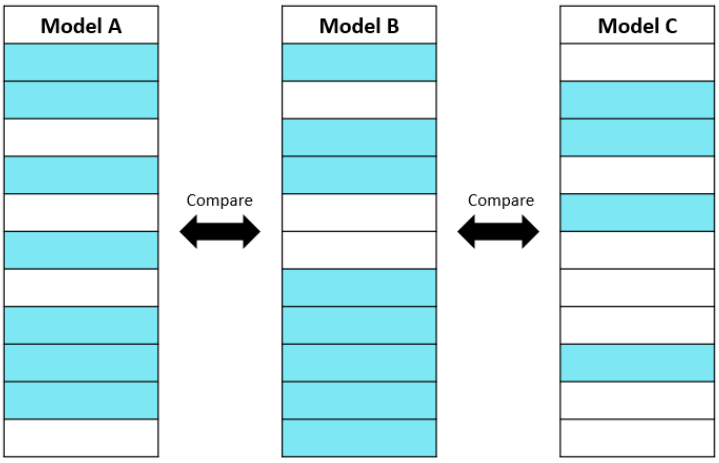



*Figure 12: Schematic overview of model construction*

### 6. Writing the review

The results of this literature review consist of applying the synthesis protocol described in the previous step to the identified models. The synthesis for the functional capabilities and the resulting list of capabilities is described in Appendix C. There was a large amount of overlap, and only a small number of functional capabilities were described in only one of the three models. Due to the small number of included models, the decision was made to include these in the final list. The synthesis resulted in a list of 17 functional capabilities related to an integration platform.

The model synthesis for the non-functional capabilities can be found in Appendix D. Ten models were included, six covering the aspects that influence the adoption of EAI and four covering the aspects that influence the adoption of cloud within organizations. Due to this larger amount of models, the process of synthesis was more difficult. Several aspects were excluded as they were only present in one of the ten models and overlapping descriptions were combined resulting in a total of 20 business factors influencing the adoption of EAI/SaaS. To shorten the list of non-functional capabilities, the approach of Rahman and Reza [71] is used. They mapped non-functional capabilities

on the ISO 25010 framework [45] which is ISO's latest system and software quality model. The mapping resulted in a selection of eleven non-functional capabilities that are listed in Appendix D.

## 3.2 iPaaS Definition

iPaaS is part of the cloud computing "family": Platform-as-a-Service. In short, this means that the responsibility for the hardware and software required to use the integration platform lies on the vendor side, and not on the client side. A more detailed explanation of the as-a-service solutions is given in section 2.5.3. Literature is in agreement with how an iPaaS is defined. This definition is provided by Pezzini and Lheureux (2011) [67] and formulated as:

> *"a suite of cloud services enabling development, execution and governance of integration flows connecting any combination of on-premises and cloud-based premises, services, applications and data within the individual, or across multiple organizations."*

Deriving from this definition, an iPaaS is "**a suite of cloud services**". The papers in the SLR do not explain what a cloud service actually is. However, Theilig et al., (2018) [85] describe a Cloud Service Provider as a party that facilitates easy access to remote resources. Hence, this research assumes that a cloud service is a remote resource that can be accessed by companies.

Furthermore, an iPaaS uses **integration flows** to connect applications. The papers do not define the integration flow. Therefore, this research assumes an integration flow to be everything that is responsible for the integration between two applications. The term 'integration flow' can then be used to refer to a specific integration between two applications. An iPaaS should enable the **development**, **execution** and **governance** of these integration flows. These are three distinct activities and section 3.3 summarizes through which components and functionalities these activities are realized.

An iPaaS should be able to connect both **on-premise** and **cloud-based** applications. Figure 13 shows how an iPaaS platform mediates between these environments.

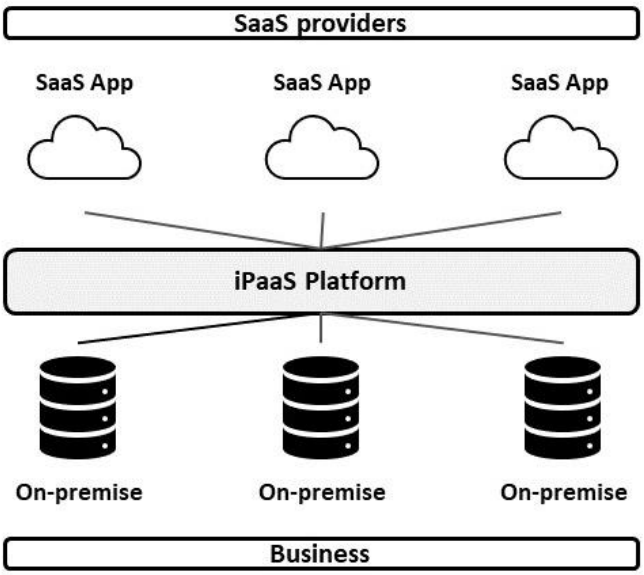



*Figure 13: Global relationship structure of the iPaaS market [62]*

Lastly, an iPaaS should connect **services, applications and data** referring to the lower two dimensions in Figure 14.

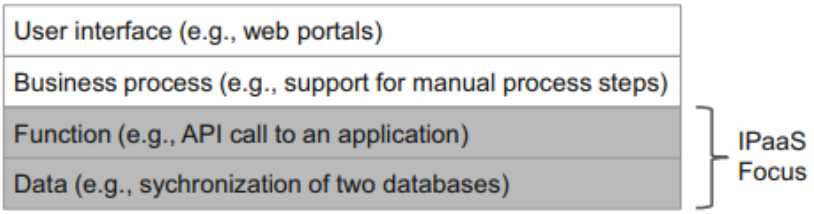



*Figure 14: Levels of integration addressed by iPaaS [21]*

One of the older papers on iPaaS was written by Potočnik and Juric (2012) [69]. They provide some general capabilities of an iPaaS which are presented in Table 7.

| Capabilities | Description |
|---|---|
| **Data integrity and security** | Data should be complete and consistent via data synchronization mechanisms to provide automatic transformation and migration, and data should be secure. |
| **Data transformation and migration** | Transform data between different storage types and formats |
| **Connectivity** | Connect different systems using their native interfaces |
| **Governance and management** | Mechanisms to support governance and management of integration services |
| **Orchestration** | Enable service orchestration into business processes |
| **Monitoring** | Offer insights into the performance of the integrations |

*Table 7: Responsibilities of an iPaaS [69]*

Despite being general, these capabilities are still true. However, Potočnik and Juric's works lack detail on the implementation of these responsibilities. Consequently, it can be argued that these capabilities apply to any integration platform, encompassing both iPaaS and on-premise platforms. Given this context, this research seeks to extract further details on the components and functionalities of an iPaaS that enable it to fulfil these capabilities. These are examined and summarized in section 3.3.

## 3.3 iPaaS Components and Functionalities

According to the iPaaS definition, it should enable the development, execution and governance of integration flows. To further elaborate on these distinct activities (development, execution and governance), the different findings of the papers included in the SLR are summarized per activity.

### 3.3.1 Development of Integration Flows

iPaaS should allow the development of (highly complex) integration flows [10][14][41][62] using visual design tools to describe integration flows that include logic-based branches, process hierarchies and complex transformation operators [21]. Visual design tools are described as a business process modelling language by [14]. Message transformation capabilities are also mentioned by [54] but not further exemplified how iPaaS offers this capability. Additional data integration concepts include data synchronization, data format, data migration, structure transformation and data replication [10][69]. iPaaS should enable service orchestration into business processes composed of diverse SaaS services or applications [54][69]. Again, no further details are provided in the papers as to how that is reached. Furthermore, all of the above-described functionalities are regular integration platform functionalities, not specific to iPaaS.

iPaaS should include development methods [10][14][27][41][42][69]. An example of such a method is the “drag and drop” or no-code/low-code development feature within the integration flows [27][42]. This approach allows users with limited coding skills to employ pre-built components for creating functional integration flows [57]. It is important to mention that low-code development features

existed before the origin of iPaaS. A traditional integration platform like BizTalk also has a low-code development tool. Key elements in low/code development are pre-built adapters/connectors that make for smoother database and application integration and reduced development time [14][21][40]. A pre-built adapter is a piece of code that allows connection and communication with a specific type of application or data source.. These connectors expose the fields and entities from the configuration of an endpoint of the application such that an iPaaS user can connect their applications [57]. This is not a feature distinct to iPaaS as pre-built adapters are also available in TIPs. They can range from file-based Hypertext Transfer Protocol (HTTP) adapters to more complex applications or Electronic Data Interchange (EDI) adapters [21]. The concept of HTTP and EDI was explained in section 2.1. The user should also be able to develop adapters for enterprise-specific applications [21][80]. The development of these adapters is facilitated by the iPaaS vendors' software development kits. Since iPaaS is a Platform-as-a-Service, users do not have to install any hardware or middleware to make use of its functionalities [14][21][39][40][42]. Till now, the last statement on hardware and middleware instalment is the only concrete difference between iPaaS and traditional integration platforms.

Table 8 summarizes all described development features in iPaaS literature and classifies if they are specific to iPaaS or if they are general integration platform concepts.

| Described in iPaaS literature | Reference | Specific for iPaaS? | Argument |
|---|---|---|---|
| **Develop complex integration flows** | [10][14] [41][62] | No | This is the essence of any integration platform |
| **Visual design tools/business process modelling language** | [21][14] | No | Existed before iPaaS in TIP offerings, see the 'TIP: BizTalk' section for a TIP example |
| **Message Transformation** | [21][54] | No | An essential capability of any integration platform as described by [35]. |
| **Service Orchestration of SaaS services** | [54][69] | Maybe | iPaaS papers do not further elaborate. Service orchestration can be done through a visual design tool. If iPaaS has more native adapters to SaaS services, then it could be better suited to orchestrate these services. |
| **Drag and Drop/Low-code** | [27][42] | No | Existed before iPaaS in TIP offerings. See the 'TIP: BizTalk' section for a TIP example |
| **Pre-built adapters** | [14][21][40] | No | Existed before iPaaS in TIP offerings. See the 'TIP: BizTalk' section for a TIP example |
| **PaaS service model** | [14][21][39] [40][42]. | Yes | TIP was never offered 'as-a-service' so correct. |

*Table 8: Classifying if iPaaS development functionalities are specific to iPaaS*

### 3.3.2 Execution of Integration Flows

The platforms support advanced EAI concepts such as message queues or transaction processing. Synchronous and asynchronous communication mechanisms are available as well as single/batch and scheduled/event-based execution of the integration flows [21]. Evaluating received data and outcomes, and proposing or implementing solutions to control the managed process automatically is summarized as monitoring and analysis [14]. They also describe that iPaaS should guarantee smooth context adaption and system evolution. However, the paper lacks further explanation as to how these aspects are realized in practice. Furthermore, this section is again not unique for iPaaS as all the mentioned concepts are present in traditional integration platforms as well.

In terms of execution, iPaaS should facilitate routing and protocol conversions. Users develop, test, deploy and monitor the execution of their integration flows "in the cloud" and pay the provider for a "service." At onboarding time, iPaaS clients implement the local agent software appliance, possibly in multiple instances for high availability and scalability, which will be under the full control of the iPaaS provider. As explained earlier, integration flows are often designed using low-code development tools which provide a graphical interface to the user. However, once the flow is implemented, hosted and triggered, the iPaaS transforms the low-code integration flow into actual executable code and metadata. These are then transparently deployed on the proper local agents to execute the functionality of the integration flow. At runtime, the local agents report performance and statistical data to the iPaaS cloud, so that users can track and manage their integration flows from the iPaaS management console [54].

Table 9 summarizes all described execution features in iPaaS literature and classifies if they are specific to iPaaS or if they are general integration platform concepts.

| Described in iPaaS literature | Reference | Specific for iPaaS | Argument |
|---|---|---|---|
| Message Queues | [21] | No | Introduced by [35] as a general integration feature. |
| Synchronous/Asynchronous communication | [21] | No | Introduced by [35] as a general integration feature. |
| Routing | [54] | No | Introduced by [35] as a general integration feature. |
| Protocol conversion | [54] | No | Introduced by [35] as a general integration feature. |
| Single/Batch execution | [21] | No | The concept existed before the origin of iPaaS |
| Scheduled/Event-based execution | [21] | No | The concept existed before the origin of iPaaS |
| Transaction processing | [21] | No | The concept existed before the origin of iPaaS |

*Table 9: Classifying if iPaaS execution functionalities are specific to iPaaS*

Figure 15 shows that according to Ebert et al., (2017), there are three different variants of iPaaS architectures depending on where the integration system is developed and where it is executed, local (on-premise) or in the cloud [21]:

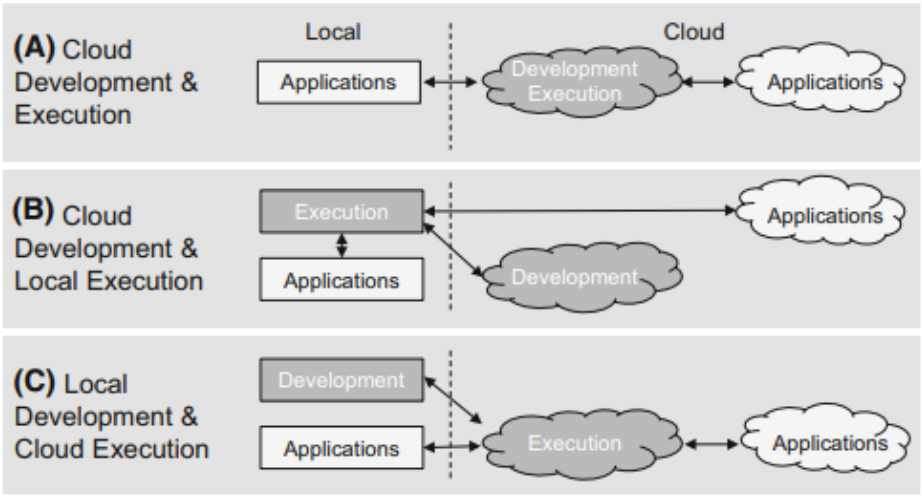



*Figure 15: Architecture variants of iPaaS [21]*

In Architecture A, the development is supported by web-based design tools storing meta-data, such as data mappings and integration flow definitions, in the cloud. During the execution of the integration flows, 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 [21]. This seems to be the traditional public cloud iPaaS environment as discussed in the section 2.5.

While the development in Architecture B is web-based, the integration flows are executed locally. Integration flows are deployed to a local execution environment fully managed by the user. If only on-premise applications are integrated, no application data needs 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. This option might be preferred when a company wants to keep its data inside but the iPaaS vendor offers more development features in their cloud-based environment to develop integration flow [21]. Instinctively, this architecture variant should have huge implications on the way the integration platform is being set up and managed. However, the authors, Ebert et al., [21] provides no further information on this architecture variant which leaves room for many questions. For example, if a local execution environment has to be set up by the company using the platform, can it still be referred to as an iPaaS? Recall from section 2.5.3 that within the Platform-as-a-Service, a company user does not have to manage any infrastructure.

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 the second step, processes are deployed to a vendor's cloud-based execution environment similar to Architecture A. Compared to Architecture B, the user neither needs to set up nor manage the execution environment but can simply scale the environment according to resource demands [21]. Similar to architecture B, too little information is provided on this architecture variant. What is meant by the vendor's cloud-based execution environment? Is this a public cloud and if so, is this option possible for all vendors to develop locally and execute their public cloud? Or could it be that the execution environment is a private cloud hosted at the vendor? If that is the case, the company using the platform could (partially) be responsible for setting up and managing these resources (section 2.5.4).

To summarize, even though these architectural variants seem like important distinctions, the paper does not further elaborate on these different options.

### 3.3.3 Governance of Integration Flows

iPaaS should have real-time process management and monitoring [14] [62] [68] [69]. “It should be able to detect the quality of service (QoS) and service level agreement (SLA) violations and inform users about the problems. On the other hand, it should provide cloud-oriented Business Activity Monitoring (BAM) with metrics defined across multiple SaaS applications” [69]. Furthermore, several papers include the topic of governance within iPaaS and conclude that an iPaaS should have well-defined governance [10][42][62][69]. “iPaaS should provide mechanisms and functionalities to support governance and effective management and provisions of integration services” [69]. “The governance for SaaS should be extended Service Oriented Architecture (SOA) governance as it should cover SaaS application performance, backward compatibility, continuous support, security” [69]. However, [42] merely mentions it as a benefit compared to on-premise data integration without further investigating the governance in an iPaaS. [62] states that governance can reduce errors and increase data security and integrity in an iPaaS. The relationship between governance and security is well established but their statement does not say anything about the governance implementation in an iPaaS. Marian (2012) [54] mentions the following governance concepts: security federation, usage tracking, administration, registry/repository, artefact life cycle management, and policy

management/enforcement. The iPaaS governance platform services can potentially be used independently from the integration platform services.

Table 10 summarizes all described governance features in iPaaS literature and classifies if they are specific to iPaaS or if they are general integration platform concepts.

| Described in iPaaS literature | Reference | Specific for iPaaS | Argument |
|---|---|---|---|
| Real-time management and monitoring | [14] [62] [68] [69] | No | Existed before iPaaS in TIP offerings. See the 'TIP: BizTalk' section for a TIP example |
| Business Activity Monitoring | [69] | No | Existed before iPaaS in TIP offerings. See the 'TIP: BizTalk' section for a TIP example |
| Extended SOA governance | [69] | No/Maybe | Unclear what extended SOA governance is and no further description is given in the literature. However, section 2.4.3 introduced SOA and Enterprise Service Bus (ESB) which existed before the iPaaS introduction |
| Security federation, Usage tracking, Administration, registry/repository, Artefact life cycle management, Policy management and enforcement | [54] | No | The author provides no further information on these concepts. Without context, these concepts are very broad and apply also to TIP. |

*Table 10: Classifying if iPaaS governance functionalities are specific to iPaaS*

## 3.4 iPaaS Benefits and Drawbacks

Several papers introduced benefits and/or drawbacks associated with iPaaS.

### 3.4.1 Benefits

Compared to traditional EAI methods, iPaaS is easy to use and adopt [21][42]. The following reasons for this are given: iPaaS was developed later with an increased focus on usability, drag and drop integration building methods were used as well as pre-built adapters and reused data mapping templates [21]. Section 3.3 already explained that these features are not exclusive to iPaaS. Regarding the statement that iPaaS was developed later with an increased focus on usability. This statement is very speculative as one could also argue that the traditional integration platform has proof of concept whereas the iPaaS platform is still being tweaked with different services being tried and adjusted along the way.

iPaaS is a developer-friendly integration platform that needs little set-up time to build integrations [42]. Therefore, integration of new applications is faster and maintenance costs for changing existing integrations are lower [21]. The aspect of cost savings was also mentioned by Bolloju and Murugesan (2012) [10] but they related cost savings to procuring, deploying and managing the required integration infrastructure. It is again unfortunate that the papers do not dive into the cost aspect of iPaaS compared to traditional integration platforms. Indeed, a company does not have to manage or deploy the required infrastructure for integrations but this research does not agree that one can argue based on that statement that it is also cheaper then.

A central advantage of the iPaaS solution for companies and SaaS providers is a reduction in the number of connectors required which is made possible by the platform structure [62]. This statement is true when comparing iPaaS to traditional point-to-point integration because iPaaS offers a central point of communication meaning that an extra application only has to make one connection to the

iPaaS platform to be connected to all other applications. Section 2.4 further explains this and also shows that this is not a specific iPaaS benefit but instead, a benefit of each integration platform as traditional EAI middleware provides the same advantage.

The scalability of the platform was mentioned as a benefit by [40] and [42]. This is a very true statement. Furthermore, iPaaS allows the creation of efficient and cohesive integration systems that can automate activities, and conduct data evaluation and decision-making [40]. In line with efficiency and cohesiveness, the high productivity of iPaaS was mentioned as a benefit [21]. They also include the predictability of costs as a benefit. It remains unknown why iPaaS provides a better predictability of costs than traditional integration platforms. The entire structure of the payment model is different for iPaaS than for traditional integration platforms. If anything, traditional integration platforms are producing more predictable costs because a company generally pays a perpetual licensing fee whereas iPaaS often operates a subscription-based model that uses a recurring fee based on usage, traffic and volume. Finally, in line with the definition of iPaaS, there is no need to acquire, initiate and manage hardware and application infrastructure software in their data centres [54].

### 3.4.2 Drawbacks

The most often mentioned drawback was related to data security challenges as enterprise application data and meta-data is transferred through the internet [42][21][62][69][54]. Also, this data could potentially be shared with a third-party provider which poses additional risks [21].

The choice of selecting the right platform is crucial [10][21][42]. Users do not have the same kind of control over upcoming updates compared to on-premise integrations which limits their flexibility [69]. Therefore, choosing the platform vendor influences the ability to control the kind of integration solutions a company can develop [14][42]. Also, the iPaaS vendor decides which pre-built Application Programming Interface (APIs) and adapters are available. A lack of sufficient pre-defined APIs was identified by [69] as a drawback. Bolloju and Murugesan (2012) [10] state some requirements to consider when selecting a platform:

- The investments made in the existing IT environment
- Type, volumes and complexity of B2B interaction
- Available mechanisms for interaction with different types of customers, suppliers and partners

Not being able to control upcoming updates contributes to the risk of vendor lock-in [21][40]. Limited portability further increases this risk [21]. Limited portability means that integration solutions developed in one iPaaS solution are difficult to transfer to another iPaaS solution. As a result, it becomes less attractive to switch from iPaaS providers contributing to vendor lock-in. Furthermore, compliance checks with laws and internal regulations as well as technical challenges were identified as factors that might complicate and slow down the deployment of iPaaS in contrast to EAI [21][40]. These technical challenges can include the need for an appropriate firewall setup from the company side, but also performance issues from the iPaaS vendor or the internet connection which affects the message and data exchange between applications [21]. Lastly, iPaaS vendors do not provide a clear cost-benefit ratio compared to other integration methods [62]. This could be explained by the complexity of providing a quantitative number in terms of perceived benefit due to the specific integration architecture and needs of each company. However, such insights would make the decisions for companies more straightforward.

## 3.5 Conclusion Literature Review

According to Pfeffer's, the resources required to define the objectives of a solution include knowledge of the state of problems [66]. Current literature lacks a structural approach to describe iPaaS. Each paper contributes with a small number of iPaaS functionalities but none of the papers provide an overall model used to describe iPaaS. Furthermore, almost all functionalities prescribed to iPaaS are

similar to/present in traditional integration platforms. Similarly, some described benefits and drawbacks are not exclusively to iPaaS but are valid for traditional integration platforms as well. Others are presented with little context which makes it impossible to assess if true. The review also showed different architectural variants of iPaaS which confused as to whether some of these variants can be referred to as iPaaS. This Systematic Literature Review confirms the idea that more in-depth research is needed on the differences between iPaaS and traditional integration platforms. Table 11 summarizes the current state of the problems, which are used to formulate requirements in the design and development phase.

| Ref | Problem |
|---|---|
| **P1** | There is no structural model to describe the integration platform in its entirety. |
| **P2** | The described iPaaS functionalities are often general integration platform functionalities. They do not mark notable differences between iPaaS and Traditional Integration Platforms (TIP). |
| **P3** | Some described iPaaS benefits and drawbacks apply to integration platforms in general. |
| **P4** | Some described iPaaS benefits and drawbacks are not provided with enough context to assess if true. |
| **P5** | Different architectural variants are presented by [21] that raise the question if these variants can still be referred to as iPaaS. |
| **P6** | Current literature does not reflect on the decision-making process for organizations that have to select an integration platform. |

*Table 11: The identified problems during the Systematic Literature Review*

The second literature review (section 3.1.2) provided this research with a first list of functional and non-functional integration capabilities derived from the literature. This list is operationalized and serves as input in chapter 4 to solve P1.

# 4 Design and Development

This chapter describes the design and development of the different artefacts of this research. First, section 4.1 presents the design requirements and the design approach. Section 4.2 introduces a glossary that is used to clearly distinguish between Traditional Integration Platforms (TIP) and integration Platform-as-a-Service (iPaaS). Section 4.3 presents the first iteration of expert reviews. Section 4.4 presents the Capability Model 1.0. Lastly, section 4.5 introduces the Non-Functional Capability (NFC) Tool 1.0.

## 4.1 Design Requirements and Approach

The problems identified in Chapter 3 and described in Table 11 correspond to requirements that should be addressed in this design and development phase. Table 12 shows the formulated design requirements for this research. The last column maps the requirements to the corresponding problem.

| Requirement ID | Requirement | Relates to the problem: |
|---|---|---|
| **R1** | This research must clearly and completely distinguish between iPaaS and TIP. | P5 |
| **R2** | This research must present a structural model to describe integration platforms. | P1, P2 |
| **R3** | This research must present use cases of integration platform decision processes that include TIP and iPaaS | P6 |
| **R4** | This research must present a comparison of iPaaS and TIP, their benefits and drawbacks, based on the structural framework. | P2, P3, P4 |

*Table 12: Design requirement and their mapping to the research problems*

To address the aforementioned requirements, this research proposes a design of three components: Component one is a glossary that aims at addressing R1; Component two is a Capability Model that aims at addressing R2; component three is a Non-Functional Capability (NFC) Tool that aims at addressing R4. The last requirement, R3, is addressed by conducting expert interviews to provide practical input on the decisions that organizations face when choosing for TIP or iPaaS.

Figure 16 shows a Business Process Modelling Notation model of the different steps taken in this research to develop the design components. Given the iterative nature of the Design Science Research Methodology (DSRM), the figure presents the Design & Development and the Demonstration & Evaluation phase. In total, this research conducts two iterations of the DSRM to improve the artefacts. The bottom lane contains the findings of the literature review that are used as input for this phase. The top lane contains the artefacts of this research. Red artefacts are final and yellow artefacts are improved in an iteration. The activities are marked with numbers and briefly explained below.

1) **Formulate design requirements:** The research problems formulated based on the systematic literature review are transformed into design requirements for this research.
2) **Develop glossary:** This is the first artefact to be developed. It allows this research to be clear on what is considered an iPaaS and what is not (R1). Understanding this is necessary to conduct the expert reviews.
3) **First iteration of expert reviews:** The first expert review iterations has two goals:
    1. Identify use cases for integration platform decision processes, and the important aspects that play a role in this process (R3).
    2. Practical validation and improvement of the capability selection from theory. The theory selection functioned as a starting point. Through practical validation, this research improves the rigour. The input of the experts is used to construct a Capability Model 1.0 (R2)

4) **Determine how to address R4:** Steps two and three addressed the first three design requirements. Using the available information and constructed model, this research faces the decision of how it should address R4. This research decides to develop a Non-Functional Capability tool 1.0 as a separate artefact that compares iPaaS and TIP on their non-functional capabilities. Section 4.5 further explains why this decision was made.
5) **Construct NFC Tool 1.0:** TIP and iPaaS are compared on the non-functional capabilities (R4). The information is presented in a usable tool to enhance practical usability.
6) **Second iteration of expert reviews**: The Capability Model 1.0 and Non-Functional Capability Tool 1.0 are subjected to a second iteration of expert reviews. Resulting in the improved versions 1.1.
7) **Case study**: The Capability Model 1.1 is applied in a case study that compares the functional capabilities of the TIP and iPaas offerings of two vendors different vendors.
8) **Ex-Post evaluation:** The benefits and drawbacks of iPaaS and TIP are listed per identified use case of R3

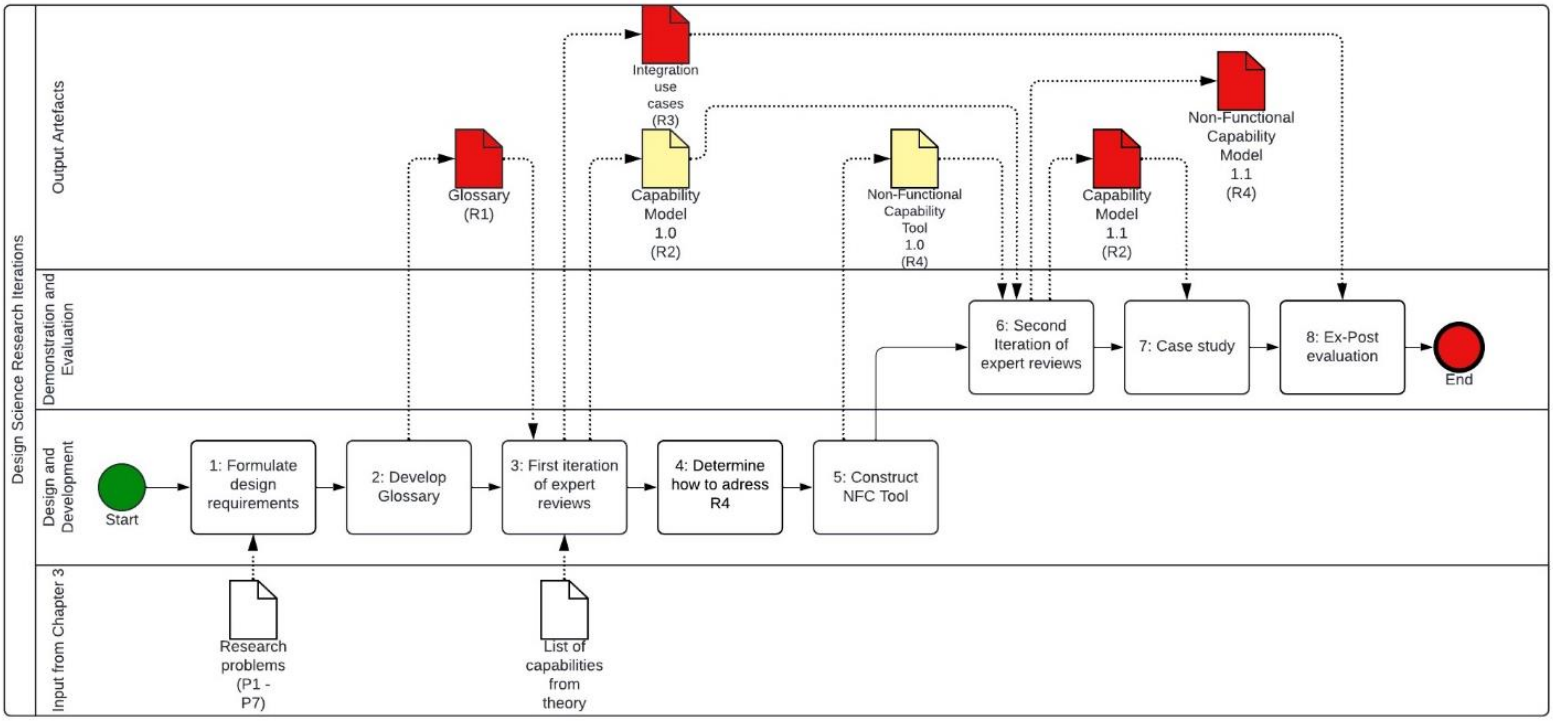



*Figure 16: Design and Evaluation Iterations of the DSRM*

## 4.2 Integration Platform Glossary

One of the research goals is to clearly describe what is considered to be an iPaaS, what is a TIP and how those two concepts differ from each other. Section 3.2 presented the iPaaS definition. However, this definition is similar to the definition given on integration platforms in section 2.6.

iPaaS definition:

> *"a suite of cloud services enabling development, execution and governance of integration flows connecting any combination of on-premises and cloud-based premises, services, applications and data within the individual, or across multiple organizations." [42]*

TIP definition:

> *"specialized software tools that help to design, implement, run and monitor integration solutions that orchestrate a set of applications to keep their data synchronized or to develop new functionalities on top of the current ones" [16].*

The main distinction is that iPaaS is offered as a Platform-as-a-Service model (section 2.5.3). Hence, a company does not have to acquire and/or configure hardware and software before using the platform and is also in no way responsible for the management of the infrastructure. Therefore, this research considers iPaaS to be a platform where the hosting is off-premise, deployment is in a public cloud and the management is externally organized. This research considers a traditional integration platform to be hosted on-premise, with on-premise deployment and internally organized management. Table 13 shows a glossary where the TIP and iPaaS definitions are extended, indicated with the '*'. Words in *italics* refer to other concepts described in the glossary.

A concept is rigid "if it is essential to all of its instances. For example, the concept animal is rigid because everything that is an animal, must be an animal and is an animal for as long as it exists. It cannot cease being animal and change into, for example, a plant" [9]. Following this classification, iPaaS and TIP are not rigid concepts. For example, Microsoft's integration platform, BizTalk, is typically offered as a TIP as described in the section 2.6 because organization A purchases a license to install BizTalk on its own infrastructure. Now consider a hypothetical scenario where Organization Z takes responsibility for the infrastructure that hosts BizTalk platforms for multiple entities (e.g., Organizations A, B, and C). This setup would resemble a public cloud, according to the definitions provided in the glossary. Thus, from Organization A's perspective, it is utilizing BizTalk through off-premise hosting and external infrastructure management, akin to a public cloud deployment. As per the glossary's definition, BizTalk, originally a TIP, effectively transforms into an iPaaS under these circumstances.

As always, there exists a grey area of possibilities where it is difficult to classify an integration platform as one or the other. However, this line of questioning quickly becomes very complex and is subjected to a great lot of subjectivism in the classification. For example, a company takes an iPaaS platform but hosts it off-premise, at some third-party vendor in a private cloud. Then it "can be hosted on an independent cloud provider's infrastructure or built on rented infrastructure housed in an offsite data centre" [90]. One could argue already that this is not an iPaaS anymore as the company needs to take some precautions to realize this and therefore the Platform-as-a-Service model is not valid anymore. However, assuming that we are still considering an iPaaS, it then becomes a question of how the management is organized at that third party where the hosting is fixed. These "management models also vary—the customer can manage everything itself or outsource partial or full management to a service provider" [90]. If a customer has to manage everything itself, it would be counterintuitive to label that as an iPaaS. To overcome this unclearness, this research for now only considers iPaaS and TIP forms as described in the glossary.

| Concept | Description | Ref |
|---|---|---|
| **Middleware** | Software tools or services that assist an application in interacting and communicating with other applications, networks, hardware, and/or operating systems. | [8] |
| **Integration platform** | Vendor-provided *middleware* enabling development, execution and governance of *Integration flow* connecting any combination of on-premises and cloud-based premises, services, applications and data within the individual, or across multiple organizations. | [42] |
| **TIP** | An *integration platform* *for which the *hosting* is on-premise, *deployment* is on-premise and the *management* is internally organized. | [42] |
| **iPaaS** | An *integration platform* *that is offered through a PaaS *cloud service model*. Therefore, the *hosting* is off-premise, *deployment* is in a public cloud and the *management* is externally organized. | [42] |
| **Integration flow** | An automated workflow used to synchronize data between multiple cloud-based and/or on-premise applications and services. | [52] |
| **Cloud Service Provider** | An IT company that provides cloud services which are on-demand, scalable computing resources like computing power, data storage, or applications over the internet. The provider offers its services according to a defined *cloud service model* | [88] |
| **Cloud Service Model** | Describes how the service layer is made available to the customer. The most popular service models are IaaS, PaaS and SaaS (section 2.5.3) | [31] |
| **Hosting location** | Refers to the provision of infrastructure and computing resources (like servers, storage, and network capabilities) on which software applications, websites, and services run. Hosting can be performed:<br>- **On-premise:** When the resources are located physically within the organisation's facilities.<br>- **Off-premise:** When the resources are physically located at a provider facility | [21] |
| **Deployment architecture** | The process of distributing and installing a software application or a service on a hosting platform, making it available for use. The deployment options are:<br>- **Public cloud:** Sharing of the same computing resources (hardware, storage, and network devices) with other organizations or cloud "tenants". access to services and management of an account goes through a web browser.<br>- **Private cloud/on-premise:** The computing resources are used exclusively by one business or organization. The resources can be physically located at your organization's on-site datacentre (the option is then on-premise or private cloud deployment) or can be located at a third-party service provider (private cloud deployment).<br>- **Hybrid:** A hybrid cloud is a type of deployment option that combines on-premises infrastructure—or a private cloud—with a public cloud. | [90] |
| **Infrastructure Management** | The responsibility for technical support, patch management, system management, monitoring and updates. The management options are:<br>- **Internal:** The organization is responsible<br>- **External:** A provider is responsible | [52] |

*Table 13: A Glossary for Integration Platforms*

## 4.3 Expert Interviews

The next step in the design and development phase is conducting semi-structured interviews. Table 14 shows the three goals of the interviews. The following questions are used for the semi-structured interviews to reach these goals:

1. In the literature, it seems that the decision for organizations to choose between an iPaaS platform and a traditional integration platform to develop and manage their integrations is unclear. What is your view on this matter?
2. In your experience, if and how do organizations struggle with the decision between different integration vendors?
3. Would you agree that the list of capabilities is complete and that it can be used to assess an integration platform?
4. Would you prioritise the capabilities and if so, how would you prioritize them?
5. Can you describe an integration project(s) that involved the decision between TIP and iPaaS or the migration from one to the other?

| Goal of interview | Rationale |
|---|---|
| Experts' views on whether the decision between on-premise and iPaaS is unclear. | To retrieve insights from the business on what is being stated in theory (see section 1.1). |
| Identify integration use cases. | So they can be used to demonstrate the capability list on them |
| Evaluate the selection of capabilities that resulted from the model synthesised in section 3.1.2 | To use that selection to analyse and compare TIP and iPaaS. |

*Table 14: Goals of the Interview*

Table 15 shows the ten experts that were interviewed, eight Deloitte experts from different departments. One expert from eMagiz, a small iPaaS provider, and one expert from Livense, a company specialising in Microsoft integration software, both BizTalk and Azure Integration Services.

| Ref | Role | Organization | Team | Date consulted |
|---|---|---|---|---|
| **1** | Senior Manager | Deloitte | Digital Customer Integration | 18-12-2023 |
| **2** | Senior Consultant | Deloitte | Digital Customer Integration | 20-12-2023 |
| **3** | Manager | Deloitte | Digital Customer Integration | 13-12-2023 |
| **4** | Integration specialist | Livense | N.A. | 12-12-2023 |
| **5** | Senior Manager | Deloitte | Cloud Engineer and Integration | 19-12-2023 |
| **6** | Senior Consultant | Deloitte | Digital Customer Integration | 20-12-2023 |
| **7** | Manager – Oracle | Deloitte | Oracle | 07-12-2023 |
| **8** | Product Manager | eMagiz | N.A. | 18-12-2023 |
| **9** | Senior Manager | Deloitte | Digital Customer Integration | 08-12-2024 |
| **10** | Manager | Deloitte | Human Resources Transformation | 13-12-2023 |

*Table 15: Reference table for first validation interviews*

The transcript of the interview is provided in Appendix E. The most important findings of the interviews are discussed per the goal described in Table 14 using the references from Table 15.

### 4.3.1 Practical Validation of Research Relevance

Current iPaaS literature advised future research work directed to the benefits and drawback of iPaaS and TIP. The practical relevance of this advice is validated by experts. Nine out of ten experts (2-10) agreed that this research direction is practically relevant explaining that “organizations are struggling with the decision between a traditional integration platform and iPaaS” (2). Furthermore, within this

decision, “there are different motives that contradict each other which makes it very complicated as there are pros and cons for applying an iPaaS” (4). There was one respondent who argued that iPaaS is always the best option because “it offers the latest greatest version and it fits in the cloud movement that companies experience when adopting SaaS” (1).

### 4.3.2 Identify Integration Use Cases

The second goal of the interview was to identify integration use cases from practice that can be used as a starting point to compare iPaaS against TIP. The experts were asked about their experiences regarding integration platform selection processes for clients to understand these different use cases. These decision processes are complex and specific to each organization because many different organization-specific aspects can play a role in this process. Therefore, formulating concrete and differentiating use cases that are both general enough to be relatable but also specific enough to make accurate statements is difficult. A distinction can be made regarding the initial situation at an organization that proved to apply to each process. Either, the organization does not have an integration platform already implemented, called a greenfield situation. Or they already have an integration platform implemented, which means that they have to decide if they want to migrate from that platform.

Figure 17 shows the steps in a greenfield scenario. The use-case that this research derives is in step 2. Hence, it assumes that the organization has already made the decision to adopt an integration platform and currently faces the decision between TIP and iPaaS.

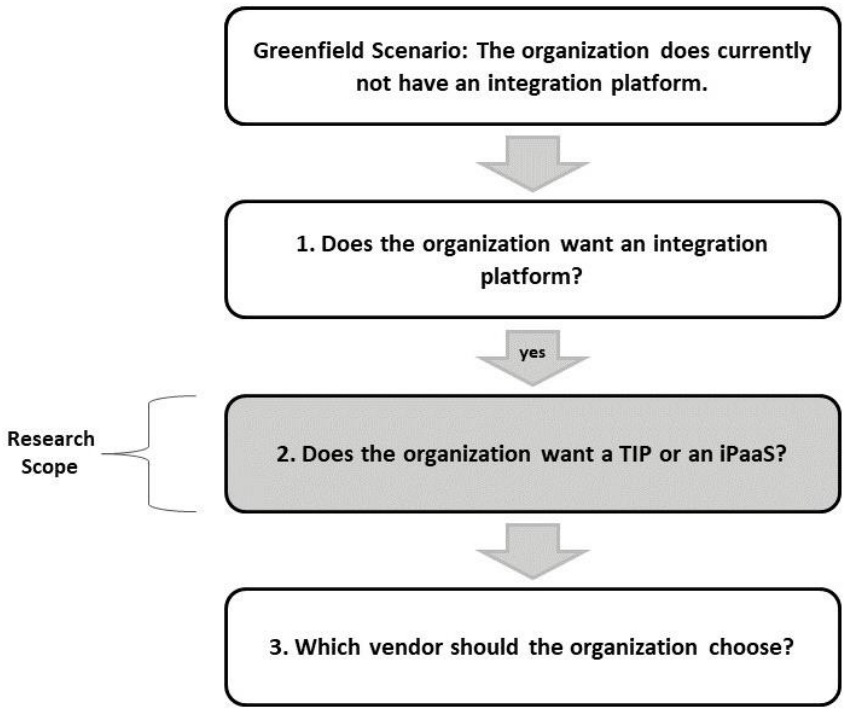



*Figure 17: Research Scope of Greenfield Scenario*

Figure 18 shows the steps in an existing integration platform scenario. The use-case that this research derives is in step 1. Hence, it assumes that an organization that has a TIP faces the decision to migrate to iPaaS or an organization that has an iPaaS and faces the decision to migrate to TIP. Based on the expert review, the latter option seems to occur very little. When organizations decide to adopt a cloud strategy they are less likely to return to the on-premise implementation. However, to complete this research this situation is also included.

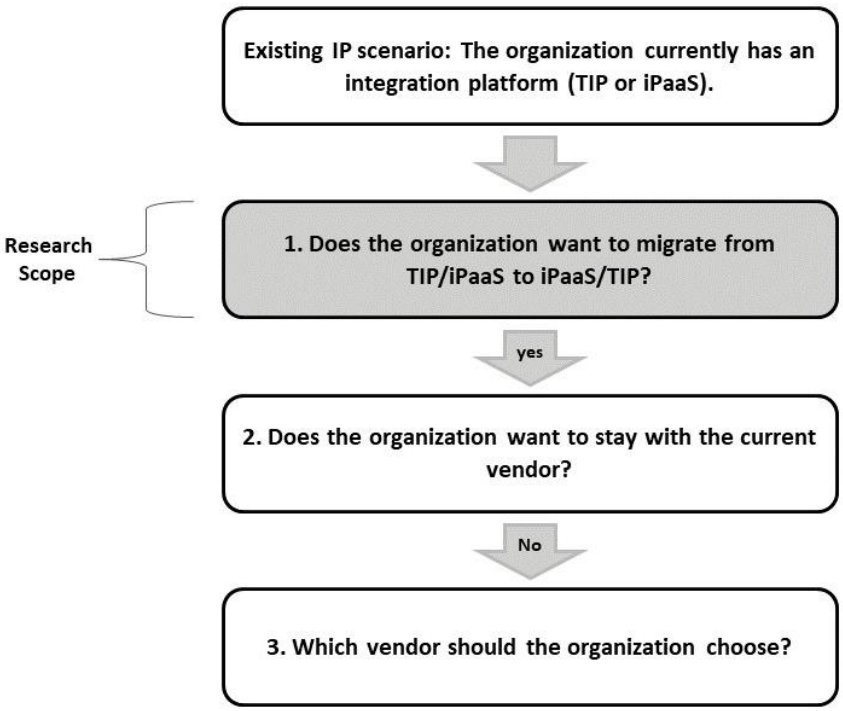



*Figure 18: Research Scope of Existing Integration Platform Scenario*

This research acknowledges that there exist more complex scenarios in which organizations have more than one integration platform implemented, possibly both TIP and iPaaS and possibly from different vendors. Due to the added complexity of these scenarios, and the scope of this research which aims to compare TIP and iPaaS, these scenarios are not included in the remainder of this research.

Table 16 outlines the different aspects that play a role in the decision between TIP and iPaaS as mentioned by the experts. These aspects are divided into general aspects that apply to both integration platforms and specific aspects that benefit iPaaS or TIP. The last column refers to the referencing as stated in Table 15.

| General aspects | Ref |
|---|---|
| **IT vision and strategy:** Does the organization have a plan to move its application to the cloud/keep its applications on-premise? This influences the decision of where to host the actual integration platform. | 5, 6, 7 |
| **Resource Extensive Migration Process:** Migrating from TIP to iPaaS or the other way around is costly. | 5, 7 |
| **No incentive to migrate:** It can be difficult to formulate a business case to invest time and resources into the migration of an integration platform. | 4, 6, 8 |
| **Future prediction:** It is difficult to assess where an integration platform will be in 3, 5, or 10 years and how that impacts the choice that organizations are making now for their integration platform. | 2 |
| **TIP Benefits** | |
| **Increased influence:** TIP offers more freedom to organizations to determine when they want to scale their resource. In iPaaS, organizations are bound by an agreement made with the vendor. | 2 |
| **Data Compliance Issues:** TIP allows the organization to enforce their internal data compliance rules. These can determine that data cannot be sent across the internet, or reside in servers located across the border. | 2, 3, 5, 7 |
| **iPaaS Benefits** | |
| **Lower setup costs:** In the greenfield scenario, the investment to realize a TIP is considerably larger than to realize an iPaaS | 1, 2 |
| **Flexible scalability:** iPaaS is designed with more features and options to realize scalability | 1 |
| **Changing business requirements:** iPaaS responds better to changes in business requirements like scalability, API-led connectivity, B2B integration and security | 1 |
| **Latest greatest:** the vendor is pushing the organization to the iPaaS platform as it offers the latest greatest there. | 1, 4 |

*Table 16: Factors influencing the choice between TIP and iPaaS from expert interviews. (N=10)*

The earlier described scenarios and the aspects listed in Table 16 are further applied in the ex-post evaluation (section 5.6) of this research. The first and second iterations are used to improve the understanding of how TIP and iPaaS compare to each other on the functional and non-functional capabilities without taking the different scenarios into account. In the ex-post evaluation, this knowledge is combined with the information described in this section to make accurate statements on the reasons to choose iPaaS or TIP in any of the scenarios.

### *4.3.3* Capability Selection Validation

The experts agreed with the division between functional and non-functional capabilities to include both technical aspects and business-intensive factors in the analysis. Regarding the functional capabilities, several experts deemed the list to be complete (1, 4, 7, 10). Other experts proposed adjustments for the selection of functional capabilities which are listed in Table 17. The referencing corresponds to Table 15.

| Proposed Adjustments | Ref |
|---|---|
| Divide the API Management capability into the following sub-capabilities: Authentication and authorisation, Traffic management, Orchestration and routing, Developer portal, Analytics, and API lifecycle management [19]. | 9 |
| Split Electronic Data Interchange (EDI) support and Partner Management into two separate capabilities. Partner Management should be described separately because it concerns a dashboard that can be used to monitor business partner integrations. EDI support falls under Adapters. | 2, 5 |
| Add Encryption and Trading Partner Agreements to the capability selection. These are important B2B capabilities that should be included within the integration platform. | 2, 8 |
| Add Integration Flow Development Tool as a capability. | 3 |

| | |
|---|---|
| Add data-related capabilities including Master and Meta Data Management, Data Virtualization and Data Synchronization | 5, 8, 9 |
| Group the capabilities in higher groups that contain sub-capabilities. Upon further discussion with the respondent, these groups were determined to be Messaging, B2B integration, File Transfer, Data Management and API Management. Some of these capabilities were already included in the model but the respondent indicated that they can be further worked out into sub-capabilities. | 9 |
| Change Connectivity to Synchronous and Asynchronous Communication because it is better recognized in the industry. | 6 |

*Table 17: Functional capability adjustments based on expert interviews (N=10)*

Several experts also proposed adjustments for the selection of non-functional capabilities. These are listed in Table 18. The referencing corresponds to Table 15.

| **Proposed Adjustments** | **Ref** |
|---|---|
| Split resource utilization into the costs of licensing, and actual resource utilization like personnel and materials. Because often, people do not associate licensing costs with resource utilization. | 5 |
| Combine availability and recoverability into one ‘availability’ capability. Often, aspects like recovery planning and disaster planning are part of availability requirements. | 4 |
| Group accessibility, learnability and operability into usability (this is also done in the International Organization of Standardization model), because they share a lot of overlap. | 5 |

*Table 18: Non-functional capability adjustments based on expert interviews (N=10)*

In terms of prioritization, it became clear that “it is quite difficult to prioritize because different priorities apply in a specific use case” (4). Within a company, it comes down to personnel preference as well as available technical expertise (5). Furthermore, it depends on who is assessing the platform, a financial person will not care about recoverability but business people do” (3). Therefore, this research does not prioritize the capabilities but instead, provides insight into each capability.

## 4.4 Capability Model 1.0

In preparing to construct a capability model, it is crucial to distinguish between 'capabilities' and 'requirements.' According to IBM, requirements are defined as the functionalities or features that organizational units need, whereas capabilities refer to the functionalities or services that these units can provide to others [76]. Thus, a requirement is a necessity or demand from a unit, whereas a capability is an ability or capacity that an entity possesses and can offer to other units. In this context, the integration platform itself is viewed as the 'providing unit' as it offers specific capabilities to other units within the organization. The aim here is to define what the platform can effectively provide (capabilities) through a capability model.

Five guidelines are stated by Lankhorst (2016) that help to define capabilities [51]:

1. Capabilities define what an entity does, not how it does it or who is doing it;
2. Capabilities may be composite, consisting of sub-capabilities. A capability may also use other capabilities;
3. Capabilities are defined in understandable domain terms using nouns;
4. Capabilities can be organized in a capability model that provides an overview of the integration platform;
5. A capability model consists of unique capabilities.

Capabilities can be functional or non-functional. Functional capabilities refer to the functions that a unit is supposed to perform. Non-functional capabilities help in assessing how well the unit performs this set of functions [77]. Non-functional capabilities are often referred to as quality attributes of a unit. In the first iteration, experts confirmed that dividing capabilities into functional and non-functional allows the research to cover the entire spectrum of integration platforms. Therefore, a

capability model can be the structural model that allows this research to describe and compare TIP and iPaas.

This section scrutinizes the design decision of the capability model. Recall that the following steps have been taken till now:

1. This research assessed that a capability model can be used to model an integration platform and validated this approach with experts;
2. This research constructed a first selection of functional and non-functional capabilities based on current literature (section 3.1.2);
3. This research validated the selection of capabilities with experts which resulted in a list of suggested improvements described in Table 17 and Table 18.

This research chooses to accept the proposed adjustments of experts (step 3) for the following reasons:

1. Current literature is scarce on integration capabilities so practical input is required;
2. The practical validation did not propose major adjustments that contradicted the literature. Instead, some refinements were proposed to improve the selection.
3. The entire selection will be subjected to expert validation for a second time.

Figure 19 shows the first version of the capability model with the suggested improvements. Appendix F shows if the capability was derived from the literature review (indicated with the corresponding source [*]. Or, added based on the expert validation (indicated with the expert reference  of Table 15 (*)).

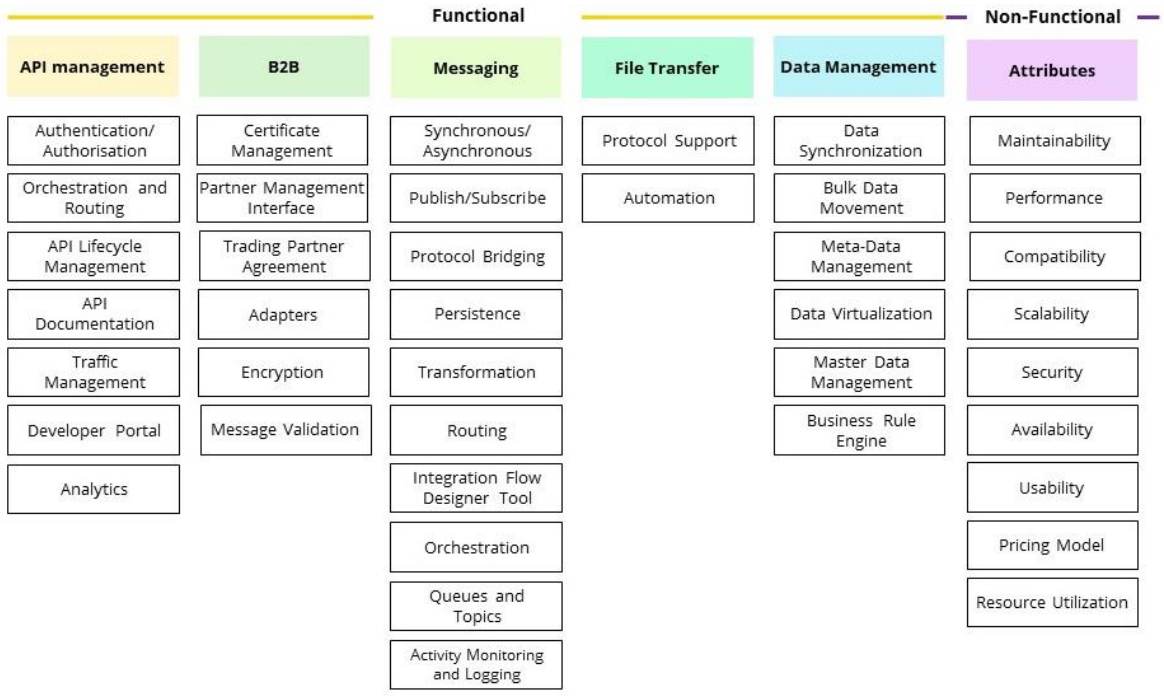



*Figure 19: The Capability Model 1.0*

The main reason to group the functional capabilities was it provides structure in the model and it acknowledges that certain capabilities are often used in combination. Expert validation confirmed the choice. Upon further inspection, it also relates to the Enterprise Application Integration (EAI) background chapter. Three of the four integration styles are present in the capability groups. These are Messaging, File Transfer and Application Programming Interface (API) management. Shared

Database was not mentioned by any of the experts as a missing capability and is also not present in the theoretical models.

Creating a distinct group for B2B capabilities reinforces the difference between inter-organizational and intra-organizational integration. To achieve B2B integration, it is likely that other capability groups will be utilized as well, which aligns with Lankhorst's second guideline. Moreover, the capabilities within the B2B group are specifically tailored for B2B integration, ensuring adherence to Lankhorst's guideline two.

In the literature, bulk data movement was identified as the primary data management-related integration capability. However, several experts recommended establishing data management as a separate capability group with additional sub-capabilities. This advice stems from the fact that many organizations seek integration platforms that not only support but also enhance data management capabilities.

It is important to mention that this capability model is not exhaustive. More sub-capabilities can probably be added. However, the model in itself is good enough for this research because it gives the different aspects to consider when deciding between TIP and iPaaS. The completeness of these different aspects is validated in the expert review.

The capability model applies to all integration platforms, both iPaaS and TIP. The next step in this research is to apply the model and compare iPaaS and TIP. The model is demonstrated and evaluated in a case study (section 5.5) to compare TIP and iPaaS within a specific vendor. Furthermore, the non-functional capabilities are further worked out into a separate tool in this design phase. Section 4.5 further elaborates on this process.

## 4.5 Non-Functional Capability Tool 1.0

The Capability Model fulfils the second design requirement (see Table 12) by defining the functionalities of an integration platform through the functional capabilities, and the quality attributed through the non-functional capabilities. To fulfil the third design requirement, this research needs to investigate how TIP and iPaaS compare on these capabilities. Ideally, the entire capability model can be used to analyse and compare the concepts of iPaaS and traditional integration platforms as described in the research objectives in section 1.2. Unfortunately, the functional capabilities cannot easily be generalized to the broader concepts of TIP and iPaaS because they rely too much on specific vendor offerings. To clarify, the iPaaS of Microsoft (Azure Integration Services) can realize its functional capabilities completely different from the SAP iPaaS (Business Technology Platform Integration Suite). That is why developers working with Microsoft iPaaS cannot directly work with the SAP iPaaS. For example, the data transformation tool of SAP iPaaS is different from the tool of Microsoft iPaaS. Consequently, comparing TIP and iPaaS on their functional capabilities requires this research to focus on vendor-specific iPaaS offerings. This research does this in the case study in section 5.5. However, for the design and development phase, this research focuses on the entire concept of iPaaS and TIP.

The non-functional capabilities are better suited in this regard. As described in the previous section, non-functional capabilities are quality attributes of the entire platform. Therefore, this research develops an information tool which further compares iPaaS and TIP on the non-functional capabilities. The tool is designed in PowerPoint because it is widely used in practice and allows for a straightforward and clear way of structuring the information. The information in the PowerPoint is derived from the literature on EAI, iPaaS and integration capabilities, and the expert interviews. Figure 20 shows an example slide on the NFC's availability. The other capabilities are structured similarly with a TIP part (part 4) and an iPaaS part (part 5) resulting in 18 pieces of information (9 capabilities, each containing an iPaaS and a TIP part). Appendix G lists all 18 pieces of information with corresponding references.

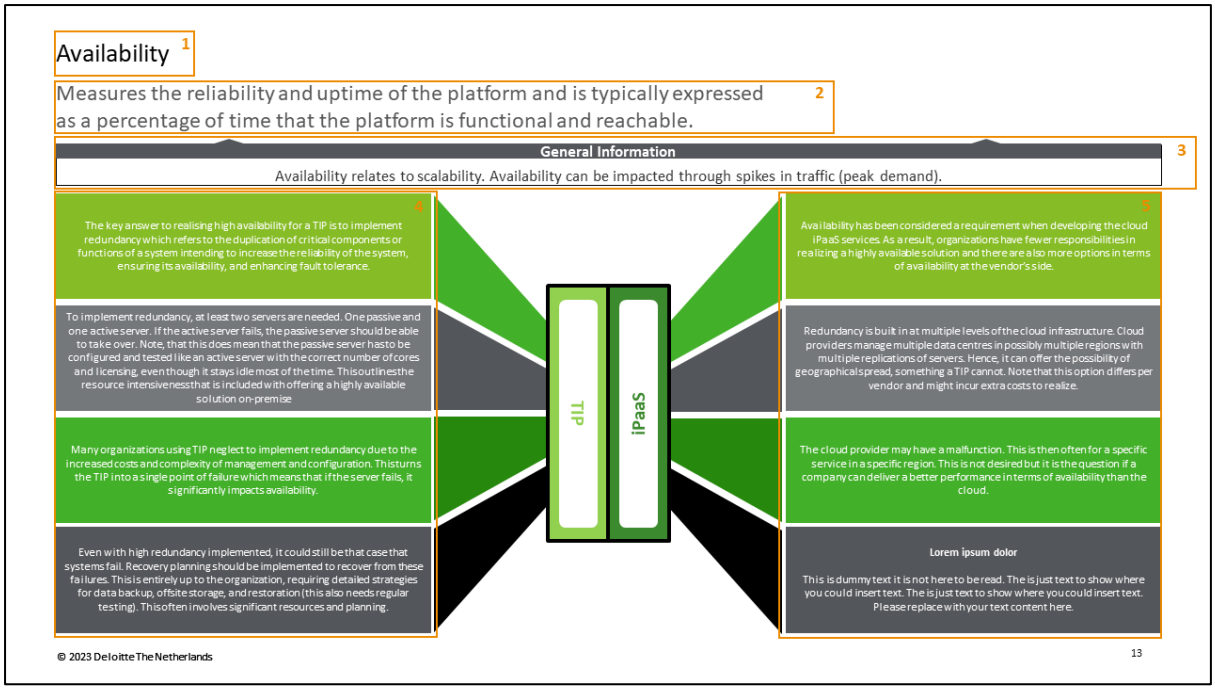



*Figure 20: The slide structure of the NFC Tool 1.0*

Table 19 describes the different components of the slide.

| Number | Description |
|---|---|
| **1** | The non-functional capability of this slide |
| **2** | A description of the non-functional capability |
| **3** | Information related either to the non-functional capability or to integration platforms in general. So not necessarily to iPaaS or TIP |
| **4** | The information on TIP for this specific non-functional capability |
| **5** | The information iPaaS for this specific non-functional capability |

*Table 19: The slide components explained*

# 5 Demonstration and Evaluation

This chapter covers the demonstration and evaluation of the Capability Model and the Non-Functional Capability (NFC) Tool and presents the changes towards the second version. Section 5.1 outlines the approach this research takes. Section 5.2 presents the expert review used to validate the NFC Tool and the Capability Model. Section 5.3 presents the Capability Model 1.1 which processed the feedback of the expert review. Section 5.4 presents the NFC Tool 1.1 with processed feedback from the expert review. Section 5.5 outlines a use case in which the Capability Model 1.1 is demonstrated on the Traditional Integration Platform (TIP) and Integration Platform-as-a-Service (iPaaS) offerings of two different vendors. Lastly, section 5.6 presents an ex-post evaluation in which this research reflects on the reasons for organizations to choose iPaaS or TIP based on the earlier defined scenarios.

## 5.1 Mixed-Method Validation Strategy

This research adopts a mixed-method validation strategy that follows the Framework for Evaluation in Design Science (FEDS) evaluation design process [87]. This process consists of four steps: (1) explicate the goals of the evaluation, (2) choose the evaluation strategy or strategies, (3) determine the properties to evaluate, and (4) design the individual evaluation episode(s). These steps are described in this section.

The goal of the first iteration of expert reviews in section 4.3 was formative. That is, the purpose of the evaluation was to improve the outcomes of the artefacts. In this second iteration, the goal shifts more to a combination of a summative approach, to judge the extent to which the outcomes match expectations and a formative approach. The aim is to improve the capability model and the NFC tool but also to test if they match the expected outcome. The strategy chosen is the Technical Risk & Efficacy strategy because it “emphasises artificial formative evaluations iteratively early in the process, but progressively moving towards summative artificial evaluations” [87]. The properties to evaluate the NFC tool are the evaluation criteria introduced in section 1.4. The capability model is evaluated on its completeness.

In total, this research designed three individual evaluation episodes concerning stakeholders that interact with the artefacts. First, the expert review of the first iterations is described in section 4.3. Those results have been worked out in the remaining sections of Chapter 4. The second and third episodes are described in this chapter and consist of a second iteration based on expert reviews and a case study. Figure 21 places these evaluation episodes on the FEDS graph [87].

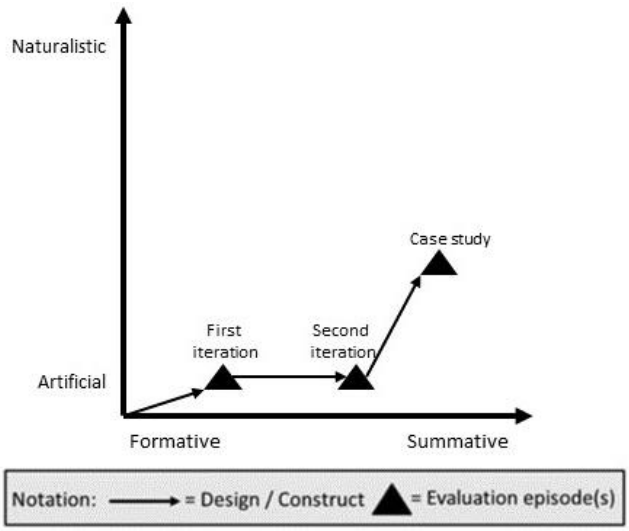



*Figure 21: Evaluation Episodes of this research [87]*

The second evaluation episode evaluates both the Capability Model 1.0 and the NFC Tool 1.0 through expert reviews. Aiming to constitute a diverse group of experts including Deloitte Consultants, integration platform users and integration platform vendors to improve the validity and generalizability of the findings. The purpose of the evaluation is partly formative and partly summative. The artefacts are improved based on the provided feedback but they are also evaluated on the evaluation criteria defined in the section 1.4. The evaluation paradigm is artificial because the experts assume that they apply the artefacts in a real-world context.

The third evaluation episode demonstrates the Capability Model 1.1 in a case study that compares the iPaaS and TIP offerings of a specific vendor on their functional capabilities. This case study is performed together with an integration expert with a technical understanding of the integration offerings of that vendor. The purpose of this evaluation is more summative as it aims to test if the capability model can efficiently be used to structure the comparison. The evaluation paradigm leans more to the naturalistic side as the capability model is used in the real-world process together with an integration expert. However, the researcher plays a large role in the case study making it also an artificial setting.

Table 20 summarizes the demonstration and evaluation approach used in this research.

| Method | Section | Input artefacts (to be demonstrated/evaluated) | Outcome | Section |
|---|---|---|---|---|
| **Expert Review** | 0 | Capability Model 1.0 | The final version of the Capability Model (version 1.1) | 5.3 |
| | | NFC Tool 1.0 | The final version of the NFC Tool (version 1.1) | 5.4 |
| **Case Study** | 5.5 | Capability Model 1.1 | Demonstrate that the capability model can be used to compare TIP and iPaaS | 5.5 |

*Table 20: Overview of demonstration and evaluation approach*

## 5.2 Expert Review

### 5.2.1 Review Participants

The initial phase of validation involves conducting interviews with integration experts. Given their direct relevance, their insights into the model's understandability, usability, and practical application are crucial. Table 21 outlines the experts who take part in the evaluation, with those who participated in the initial interview round (section 4.3) highlighted in bold. Engaging this particular group is critical as it ensures the model is reflective of their profound knowledge, a process also referred to as member-checking [91]. A unique reference number was assigned to each participant, which is utilized in sections 5.3 and section 5.4 to document any modifications to the artefacts. Appendix H presents the transcription of the interviews.

| Ref | Role | Organization | Team | Date consulted |
|---|---|---|---|---|
| **1** | Senior Manager | Deloitte | Digital Customer Integration | 26-02-2024 |
| **2** | Senior Consultant | Deloitte | Business Operations | 27-02-2024 |
| **3** | Specialist Lead Delivery | Deloitte | Oracle | 28-02-2024 |
| **4** | **Senior Manager** | **Deloitte** | **Cloud Engineer and Integration** | **28-02-2024** |
| **5** | **Product Manager** | **eMagiz** | **N.A.** | **28-02-2024** |
| **6** | Specialist Lead | Deloitte | Oracle | 01-03-2024 |
| **7** | Senior Consultant | Deloitte | Technology Vision and Architecture | 13-03-2024 |
| **8** | Enterprise Application Architect | BDR Thermea Group | N.A. | 22-03-2024 |

*Table 21: Review Participants*

### 5.2.2 Review Protocol

The NFC tool is validated based on the requirements presented in the section 1.4. These requirements are scored on a 5-point Likert scale. The expert interviews are also used to provide a second validation of the selection of capabilities in the capability model (section 4.3.3)

The review is structured according to the following protocol:

1. **Preparation:** The participant receives the NFC tool in advance and is asked to read through it and evaluate the slides in the tool based on the evaluation criteria described in step 2 in Table 22.
2. **Evaluating the individual slides:** During the interview, the participant is asked to go through each slide of the tool and to provide feedback on every aspect using the questions described in step 2 in Table 22.
3. **Evaluating the complete tool:** The participant is asked to provide feedback on the entire tool based on the questions described in step 3 in Table 22.
4. **Evaluating the capability model:** The participant is presented with the capability model and is asked the questions described in step 4 in Table 22.

Table 22 outlines the questions per step in the protocol:

| Step in protocol | Questions | Answer format |
|---|---|---|
| 2: Evaluating the Slides | Would you update any TIP or iPaaS information description? If so, please explain what and why. | Yes/No + comment |
| | The information in the slide is understandable assuming that the reader has an understanding of basic integration concepts. (Understandability) | Likert scale (1-5) |
| | The information in the slide covers the non-functional capability sufficiently. (Sufficiency) | Likert scale (1-5) |
| | The information in the slide is correct. (Accuracy) | Likert scale (1-5) |
| | The information in the slide is useful for understanding the different aspects of iPaaS and TIP. (Usefulness) | Likert scale (1-5) |
| | | |
| 3: Evaluating the Tool | The slide deck tool is easy to use. (Usability) | Likert scale (1-5) |
| | This model improves understanding of the differences between iPaaS and on-premise integration platforms better than the currently available model | Likert scale (1-5) |
| | Do you feel like important topics have not been covered in the model? If so, could you explain which topics? | Yes/No + comment |
| | Would you add/change any non-functional capabilities? If so, please explain what and why. | Yes/No + comment |
| | | |
| 4: Evaluating the Capability Model | Would you add/change any of the capabilities in the model? If so, please explain why. | Yes/No + comment |
| | Can you based on your knowledge, identify functional (sub)-capabilities that only belong to iPaaS or only to traditional integration platforms? | Yes/No + comment |

*Table 22: Interview questions per step in the protocol*

### 5.2.3 Qualitative Data Analysis

The transcriptions of the expert interviews are analyzed to extract relevant insights from the data. Utilizing the qualitative data analysis methodology outlined by Dey [20] ensures a structured and insightful evaluation of the feedback. The complete transcriptions are available in Appendix H. The analysis is conducted in three primary steps:

1. **Reading and Annotating:** This involves a thorough review of the transcriptions to identify and note down sections that are particularly relevant to the NFC tool, any open-ended questions, or the established evaluation criteria.
2. **Categorizing Data:** This step requires organizing the annotated data into specific categories. These categories are detailed in the appendices of each transcription, facilitating a systematic approach to data analysis.
3. **Corroborating Evidence:** Finally, data is synthesized based on its categorization and context. For example, the specific capabilities or the comparison between TIP and iPaaS within a capability. This consolidation aids in reinforcing the findings and ensuring they are backed by the data.

### 5.2.4 Evaluation Criteria

In the second and third steps of the interview protocol (Table 22), participants evaluate the NFC tool using a series of statements based on the assessment criteria formulated in section 1.4. Table 23 displays the findings per criteria per slide. The lowest, median and average scores are used to reflect the distribution of the results.

In this evaluation, which utilized a Likert scale from 1 to 5, the lowest average score recorded was 3.7. Overall, the results indicate satisfaction with the evaluation, though there were significant insights gained during the process. Notably, 'understandability' and 'accuracy' received the highest average scores. This outcome can be attributed to the comprehensive use of text to facilitate a detailed comparison between TIP and iPaaS.

However, the extensive text also diminished the tool's practicality; several experts reported that the slide deck could not be directly utilized with clients due to its verbosity. Moreover, the 'sufficiency' of the tool scored slightly lower, as some experts pointed out the necessity of considering various organization-specific aspects when comparing TIP and iPaaS. Thus, while the tool serves as a useful starting point, it alone is not adequate for a complete analysis.

Additionally, none of the researchers were aware of any existing models before this, and they expressed high satisfaction with the development of a new one. Consequently, the potential for improving current practices was rated highly in the survey results.

Other feedback and proposed adjustments are discussed in section 4.4 and section 4.5.

| Criteria | Score | Pricing model | Resource Utilization | Security | Scalability | Availability | Performance | Compatibility | Usability | Maintainability |
|---|---|---|---|---|---|---|---|---|---|---|
| **Understandability** | Min. | 4 | 4 | 4 | 4 | 4 | 4 | 3 | 4 | 5 |
| | Med. | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| | Avg. | 4,7 | 4,7 | 4,9 | 4,7 | 4,6 | 4,7 | 4,7 | 4,6 | 5 |
| **Sufficiency** | Min. | 3 | 2 | 3 | 4 | 2 | 3 | 4 | 3 | 4 |
| | Med. | 4 | 4 | 4 | 5 | 5 | 4,5 | 4 | 5 | 5 |
| | Avg. | 4,1 | 3,9 | 4,3 | 4,7 | 4,4 | 3,7 | 4,4 | 4,4 | 4,0 |
| **Accuracy** | Min. | 4 | 4 | 4 | 3 | 3 | 3 | 4 | 2 | 3 |
| | Med. | 4 | 4 | 4 | 5 | 5 | 4,5 | 5 | 5 | 5 |
| | Avg. | 4,4 | 4,4 | 4,4 | 4,4 | 4,7 | 4,4 | 4,6 | 4,3 | 4,7 |
| **Usefulness** | Min. | 2 | 3 | 3 | 4 | 3 | 3 | 4 | 4 | 4 |
| | Med. | 5 | 5 | 5 | 5 | 5 | 4 | 5 | 5 | 5 |
| | Avg. | 4,6 | 3,9 | 3,9 | 4,1 | 4,4 | 4,0 | 4,1 | 4,1 | 4,0 |

| | | Rate Entire Tool |
|---|---|---|
| **Usability** | Min. | 2 |
| | Med. | 4 |
| | Avg. | 4,4 |
| **Improves current practices** | Min. | 4 |
| | Med. | 4.5 |
| | Avg. | 4.5 |

*Table 23: Evaluation Criteria Scores from Expert Reviews (N=8)*

## 5.3 Capability Model 1.1

This section introduces the final version of the capability model based on the expert review. Table 24 displays the results of the expert review. The last column refers to the experts introduced in Table 21.

| Capability | Comment | Adjustment | Ref |
|---|---|---|---|
| **Data management** | Within data management, only data synchronization and bulk data movement are integration capabilities. | Include Data synchronization and Bulk Data Movement as Data Management Capabilities | 1, 3 |
| **Encryption** | Encryption is not only a B2B capability. It is also used for internal traffic. | Place Encryption outside the B2B capability group | 6 |
| **Activity monitoring and Logging** | Activity monitoring and Logging are done for the entire integration platform. It is now only listed as a Messaging capability but it is also done for Application Programming Interface (API) management and B2B. | Place Activity monitoring and Logging outside the Messaging capability group | 2, 6 |
| **Adapters and Pre-Built Integration Content** | Adapters in B2B and pre-packaged integration content in Messaging show some overlap. Adapters are pre-built integration pieces that can readily be used. | Combine Adapters and Pre-Packaged Integration Content and place it in the general capability section | 2, 5 |

*Table 24: Capability model expert review comments and adjustments*

Figure 22 shows the final version of the capability model with the processed adjustments. A description of all the capabilities can be found in Appendix F.

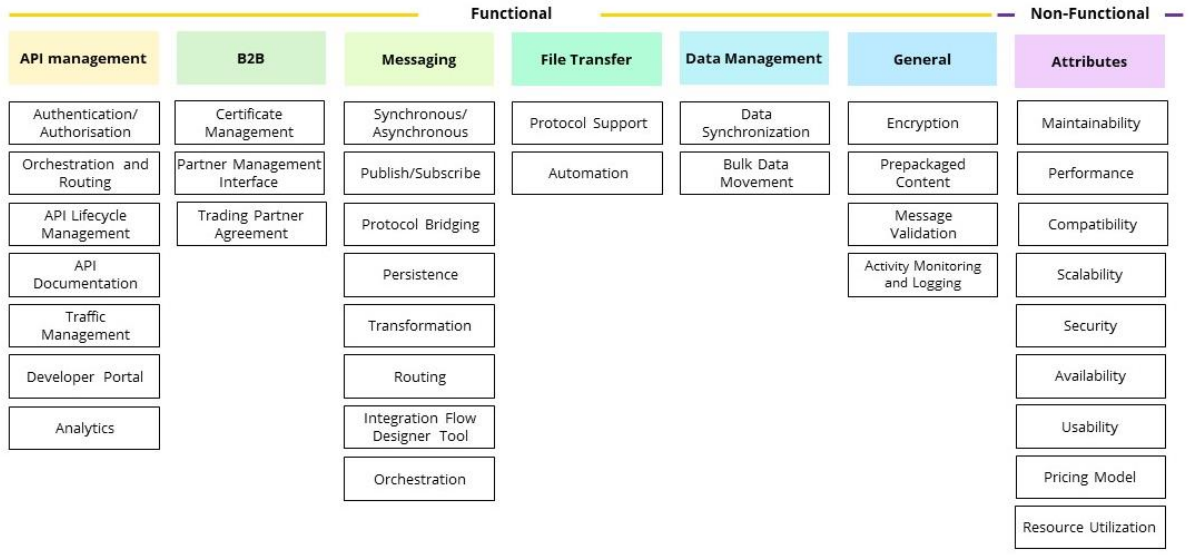



*Figure 22: Capability Model 1.1*

## 5.4 NFC Tool 1.1

This section introduces the final version of the NFC tool which includes the processed feedback that was received during the expert review. For each non-functional capability, the changes and statements from the expert's review are described with corresponding references. These references point to the experts described in Table 21. The slides of the final NFC tool can be found in Appendix I.

***Pricing Model***

Table 25 presents an overview of the adjustments made in the pricing model slide based on the expert reviews. The second column refers to the experts introduced in Table 21.

| Adjustment | Ref |
|---|---|
| Add that, depending on the vendor, prices are determined based on different aspects including a per-core licensing model or a fixed price independent of traffic | 4 |
| Remove the pricing example of Tibco from the slide because the given price is incorrect and the correct price is not published. | 4, 5 |
| Add this comment in the iPaaS section: 'It is only difficult to estimate costs in the initial phase. The moment you run an iPaaS, after 3,4 months it is pretty easy to estimate the costs | 6 |
| Add this comment to the TIP section: "With TIP, a company becomes owner of the license". On the other side with iPaaS, this is not the case so add the following text to the iPaaS section: "Organizations do not become the owner of the iPaaS services". | 7 |

*Table 25: Processed feedback on the pricing model slide*

***Resource Utilisation***

Table 26 presents an overview of the adjustments made in the pricing model slide based on the expert reviews. The second column refers to the experts introduced in Table 21.

| Adjustment | Ref |
|---|---|
| Split the columns up into separate 'needed for TIP/iPaaS' columns and 'comment' columns | 1 |
| Add an extra 'Configuration' aspect under implementation costs that explains the added complexity of TIP regarding the configuration of the extra hardware and software required for the platform. | 1 |
| Change the table header 'needed for' to 'prerequisite for' to show that servers are needed for iPaaS also but under the responsibility of the vendor | 5 |

*Table 26: Processed feedback on the resource utilisation slide*

***Security***

Table 27 presents an overview of the adjustments made in the pricing model slide based on the expert reviews. The second column refers to the experts introduced in Table 21.

| Adjustment | Ref |
|---|---|
| Add this to the iPaaS text: 'It is important to understand that even though using iPaaS transfers part of the responsibility for security to the vendor, the integrations that are developed and hosted on the platform will always remain under the organization's responsibility. That means that one unsecured developed API can harm the entire security, even if the vendor has their security part managed perfectly.' | 1 |
| Indicate that refusing to send data across the internet is a compliance matter. Also, give Brexit as a compliance example. | 3, 4 |
| Add this to the iPaaS text: 'Sometimes, action from the organization is required to implement new security updates.' | 5 |

*Table 27: Processed feedback on the security slide*

***Scalability***

Table 28 presents an overview of the adjustments made in the pricing model slide based on the expert reviews. The second column refers to the experts introduced in Table 21.

| Adjustment | Ref |
|---|---|
| Add this text to the general part: 'Whether it's TIP or iPaaS, platform expertise is crucial for scalability. The main difference is that TIP also requires infrastructure alignment, needing broader expertise. While both need platform experts, iPaaS simplifies managing scalability components.' | 1 |
| Add this to the iPaaS text: 'So the ease of scalability depends on this specific agreement.' | 3 |
| Add this section to the TIP text: "A key advantage of TIP is that it allows organizations complete control over scaling, despite its resource demands. Unlike iPaaS, where bureaucratic hurdles can delay scaling, TIP enables organizations to scale on their terms". | 7 |

*Table 28: Processed feedback on the scalability slide*

***Availability***

Table 29 presents an overview of the adjustments made in the pricing model slide based on the expert reviews. The second column refers to the experts introduced in Table 21.

| Adjustment | Ref |
|---|---|
| The point of criticism on geographical availability seems to be in the question where this is automatically offered. Since eMagiz also does not offer it automatically but on request. The following text is added to the part on geographical availability: 'Note that this option of geographical availability differs per vendor and might incur extra costs to realize.' | 4, 5 |
| Add this section to TIP: "Many organizations using TIP neglect to implement redundancy due to the increased costs and complexity of management and configuration. This turns the TIP into a single point of failure which means that if the server fails, it significantly impacts availability". | 7 |

*Table 29: Processed feedback on the availability slide*

***Performance***

Table 30 presents an overview of the adjustments made in the pricing model slide based on the expert reviews. The second column refers to the experts introduced in Table 21.

| Adjustment | Ref |
|---|---|
| Add to the text on TIP that higher performance can only be achieved when 'the connected applications are also hosted on-premise.'<br><br>Also added the following text to the iPaaS part: 'Can theoretically achieve higher performance compared to TIP when applications are hosted in the same data centre where the iPaaS platform is hosted.' | 1 |
| Add this example to the TIP text: 'One of the use cases where this advantage might be important is IoT applications within an on-premise production environment. Take company X which uses robots that cut crops within greenhouses. This requires the processing of a lot of traffic where an increase in performance can make a substantial difference in efficiency | 1, 4 |
| Create a general text box and add this text:<br>'Two important aspects that play a role in the performance of integrations that do not rely on whether TIP or iPaaS is used. First, the performance of an integration is affected by how efficiently that integration is developed. Unnecessary lines of code will increase latency. Second, many integration platforms are part of a vendor's wider product offering. Often, performance is increased when the integration platform integrates other products of this vendor because tailor-made adapters are used that integrate those offerings very efficiently.' | 2, 6 |
| Include this text in iPaaS after SLA information: 'Performance is often constated after the fact. It all depends if you make hard agreements with the vendor.' | 3 |

*Table 30: Processed feedback on the performance slide*

***Compatibility***

Table 31 presents an overview of the adjustments made in the pricing model slide based on the expert reviews. The second column refers to the experts introduced in Table 21.

| Adjustment | Ref |
|---|---|
| Added two sources in the slide to the statement: TIP "is promoted as better suited to handle legacy system integrations". | 1, 3, 4 |
| Add a grey text box that states this message: 'The toolkit of a platform that is, all the specific features that are included to allow a user to develop integrations, largely determines how compatible that integration platform is within a specific application landscape. For example, applications that use a lot of EDIFACT messages are easier to integrate with a platform that supports many versions of this format. Hence, compatibility should be assessed per platform and the decision between TIP or iPaaS does not play a large role in this discussion.' | 2, 3, 5 |

*Table 31: Processed feedback on the compatibility slide*

***Usability***
Table 32 presents an overview of the adjustments made in the pricing model slide based on the expert reviews. The second column refers to the experts introduced in Table 21.

| Adjustment | Ref |
|---|---|
| Add the following text to the TIP part: 'Furthermore, TIP is often not designed to be handled by business people and almost always requires technical experts to develop the integrations.' | 1, 4, 5 |
| Add the following text to the iPaaS toolkit story: 'However, this toolkit builds over time and in the beginning, iPaaS is often less feature-rich than the TIP offering of a vendor.' | 3 |
| Do not agree with business people being able to develop integrations. It remains a technical endeavour. Therefore, add the following text to the iPaaS section: ". In practice, this seems unwise because integration, even a simple one, operates in a larger complex IT landscape which always requires some technical thinking". | 7 |
| Add this section to iPaaS: "iPaaS also offers a marketplace where users can offer pre-built integrations. This greatly enhances the usability of the iPaaS platform". | 7 |

*Table 32: Processed feedback on the usability slide*

***Maintainability***
Table 33 presents an overview of the adjustments made in the pricing model slide based on expert reviews. The second column refers to the experts introduced in Table 21.

| Adjustment | Ref |
|---|---|
| Add the following text to the under-construction part: 'This does depend on the specific vendor as was mentioned in an expert review. Google and Azure often do this but platforms like Boomi, Workato and Mulesoft never retract their features.' | 1 |
| Add this comment to the iPaaS text: With iPaaS, a vendor has more control over the enforcement of accepting updates. Whereas with TIP, it depends on whether companies accept the update' | 4 |
| Both iPaaS texts cover the industry terms: Vendor-managed updates, and continuous development. Therefore, add these terms to the text. | 7 |

*Table 33: Processed feedback on the maintainability slide*

***General Adjustments***
Table 34 presents an overview of the adjustments made in the pricing model slide based on the expert reviews. The second column refers to the experts introduced in Table 21.

| Adjustment | Ref |
|---|---|
| Added two slides of Management Summary which contained the most important points per capability for TIP and iPaaS (see Appendix I). This provides a nice overview that makes the tool more practically usable. | 7 |
| Different comments were made on the concept of automatic scaling. One expert (2) stated that it is often included in most iPaaS offerings, sometimes even free of charge. Another expert (4) stated that it is often not included in the offerings. The decision was made to follow the feedback of the iPaaS provider (5) who stated that it is available for several services but it comes at extra costs. | 2, 4, 5 |

*Table 34: Processed general feedback*

## 5.5 Case Study Capability Model

The second validation phase consists of a case study that applies the capability model. As described earlier, the functional capabilities are difficult to generalize to the concept of TIP and iPaaS because they are closely connected to a specific vendor. Therefore, this case study aims to simulate the scenario where an organization want to compare the iPaaS and TIP offerings of a vendor on their functional capabilities to guide them in their decision-making process. The non-functional capabilities are excluded from this case study for the following reasons: Firstly, The case study takes a

considerable amount of time when only the functional capabilities are used. Including the non-functional capabilities makes the case study too lengthy. Secondly, the non-functional capabilities are less connected to a specific vendor and already handled in the NFC Tool.

Within this case study the capability model functions as a tool that structures the process of comparing TIP and iPaaS by comparing these platforms per capability.

### 5.5.1 Case Study Participants

Since this case study investigates the integration platform offerings of a specific vendor, participants are required to have extensive technical expertise in both the TIP and iPaaS offerings of the vendor. The vendors Microsoft and SAP are included in the case study. Each case study consisted of one session with the expert, these sessions per participant are presented in Table 35. Section 5.5.2 describes the case study protocol.

| Ref | Position | Vendor expertise | Data consulted |
|---|---|---|---|
| **1** | Integration Expert | Microsoft | 06-03-2024 |
| **2** | Senior Manager Digital Transformation | SAP | 14-03-2024 |

*Table 35: Case study participants*

The other participant is the researcher himself, taking the role of the organization that wants to compare the platform offerings.

### 5.5.2 Case Study Protocol

During the case study, the capability model is applied to understand the offerings of TIP and iPaaS of a specific vendor. The following protocol is used to apply the capability model.

1. **Selection:** The participant is invited with a detailed overview of the case study that clearly explains what is expected from the participant in terms of the expertise of the TIP and iPaaS platform of a specific vendor.
2. **Preparation:** Upon agreement, the participant is sent the capability model and asked to think about how the TIP and iPaaS offerings of the vendor differ per capability.
3. **Introducing the platforms**: Within this report, the platforms are briefly introduced based on publicly available documentation info from the vendor's website. This increases the reader's understanding of the platforms.
4. **Applying the capability model:** In one session, the participant is asked to walk through the capability model and describe/compare the TIP and iPaaS offerings on that vendor per capability. The researcher asks follow-up questions to ensure an as detailed as possible comparison.
5. **Processing Results:** The results of the sessions are processed into a readable comparison per capability.
6. **Evaluate the process:** The process of the case study is evaluated with the participant, and by the researcher himself.

### 5.5.3 Case Study Evaluation

The comparison of the platforms is described in Appendix J. It stood out that iPaaS was functionally superior to TIP. A reason for this is that iPaaS is built with the lessons learned from TIP, allowing the vendors to build a better product. Furthermore, in both cases, the TIP offering of the vendor did not have clear API management capabilities. The expert indicated that these capabilities had to be realized through another, third-party, solution. They also indicated that other TIP offerings can have an API management solution included.

Besides the results of the actual comparison, the case study is more focused on the evaluation of how well the capability model functions as a structured tool for a platform comparison. The participants were asked two open questions that are listed in *italics*. The referencing described in Table 35 is used to identify the answers of each participant:

*Do you agree that utilizing the capability model is a good way to structure this comparison?*

“Yes, I believe that the functional capabilities can help clarify the choices that organizations have to make. Thus, I think that using that capability model could have helped companies to make better decisions beforehand, which might have resulted in fewer surprises later on. Looking at my last two clients, I can say that after making their choice for an iPaaS, they still faced some surprises. These surprises could have been minimized if they first made a similar assessment like the one in this case study” (1).

”Yes, actually it is good. A similar method is used for other vendor products by colleagues of mine. However, it does highly depend on the expert used in this process. For example, I know both SAP platforms on a relatively technical level. However, if you want a strong technical analysis, developers should be brought in. It might be difficult to find one developer that has that deep technical knowledge of both platforms. So then, multiple people are required. So besides this being a good approach, other driving points influence the decision for a platform. Many customers want to go with a platform of choice on the cloud solution they are already on. So if they for example already use Azure, they will be more likely to adopt Azure Integration Services” (2).

*How would you further improve this process?*

“Normally, the comparison of platforms needs to align with an organization's specific wishes or requirements. In this case study, these wishes were not incorporated into the evaluation. This was a deliberate choice because the aim was to only compare TIP and iPaaS. That said, I see potential in using the capability model not just as a comparison tool, but also as a framework to help organizations clarify their requirements. Hence, the process should be standardized to also include the organizational requirements in the process. This would make it a more powerful tool, in its entirety, for tailored platform comparisons." (1)

“If you only want to assess the two platform offerings, this method works well. You could maybe include more people in the use case together to have a discussion. Preferably, those people have different levels of technical and business experience with the platforms” (2).

***Researcher Evaluation***

In this case study, the researcher took the role of an organization that wishes to compare two platform offerings. Hence, the evaluation of the research is also valuable to take into account.

The researcher agrees with the feedback given by the participants. Indeed, the capability model functions well to structure the comparison. However, besides the different capabilities that should be discussed, it does not provide any other structure or guidance on the comparison process. Thus, the researcher had to have the technical knowledge of the capabilities and integration platforms, in general, to guide the case study and ask the correct follow-up questions.

Per the vendor, the case study took about two hours with one expert. To improve the results of the comparison, more time per session is needed. Also, the comparison would benefit from a diverse group of experts including technical developers more business-oriented experts to have the complete picture and to foster discussion.

## 5.6 Ex-Post Evaluation

According to Venable et al., (2016) an ex-post evaluation “regards a chosen and developed system or technology after it has been acquired, designed, constructed, or implemented” [87]. In section 4.3.2, this research formulated starting scenarios which could serve as a basis to provide recommendations to organizations in terms of choosing between TIP and iPaaS. These scenarios were organizations in the greenfield (1), organizations currently having a TIP and considering switching to iPaaS (2), and organizations currently having an iPaaS and considering switching top TIP (3). This research attempts to combine the information derived throughout this research process to formulate hypotheses on the

benefits/drawbacks of TIP and iPaaS per scenario. It thereby further improves the information listed in Table 16

### 5.6.1 Protocol

1. **Formulate hypotheses for the scenarios:** Based on the knowledge acquired throughout this research. Hypotheses on TIP and iPaaS benefits are formulated per scenario. These hypotheses can be found in Appendix K;
2. **Develop survey**: The hypotheses are formulated in a survey that asks participants if they agree with the hypotheses and if they would make any modifications or additions;
3. **Select participants:** Throughout this research, a considerable number of integration experts have been contacted. The survey is sent out to all these experts to achieve a high response number;
4. **Process results:** The results are processed in the following section.

### 5.6.2 Survey Results

In total, the survey received eight responses. There was an overall agreement with the hypotheses with several proposed modifications. Several experts, understandable, noticed that the hypotheses aim to reduce a very complex decision process down to a view key point. Consequently, the list of benefits is broad and the actual decision between TIP and iPaaS will largely depend on specific organizations needs. However, the list does provide a brief overview per scenario on the benefits and drawbacks of iPaaS compared to TIP.

Based on the received feedback and formulated hypotheses, this research argues that the decision between TIP and iPaaS is an 'iPaaS unless' decision. This means that iPaaS is generally favoured unless some specific reason(s) exist to choose TIP. The following reasons are given for iPaaS' superiority over TIP:

**Organizations are unburdened:** An iPaaS vendor takes responsibility for the hardware, operating system and security software which unburdens organizations from this responsibility resulting in the following advantages over TIP:

1. **Decreased Complexity**: iPaaS, being cloud-based, removes the necessity for organizations to buy, install, and upkeep physical hardware and infrastructure. This simplifies resource management. Still, organizations require technical integration expertise and understanding of cloud services to effectively use iPaaS (2)organization's
2. **Vendor-managed updates:** Automatic updates require minimal effort from organizations.
3. **More possibilities:** iPaaS has built-in features and options to realize scalability, availability and performance of integrations. However, contractual negotiations with the vendor determine the scalability, availability and performance options. For example, a discount on purchasing a fixed amount of resources can take a lot away from the freedom to change scalability, availability, and performance.
4. **Potential Cost Savings:** Firstly, a reduction of staff that are responsible for the infrastructure. Secondly, organizations with fluctuating peak demands may find iPaaS more economical. Unlike TIP, where hardware must be scaled for peak loads, often sitting underutilized during quieter periods, iPaaS allows for more flexible scaling, potentially lowering costs. However, extensive TCO assessments are required to determine actual cost savings for specific organizations' needs.

**Vendors' latest greatest**: iPaaS is a vendor's latest greatest providing:

1. **Access to the latest features/functionalities**: New features are always launched on the iPaaS. They are not necessarily launched on the TIP.
2. **Comprehensive Toolkits:** Similar or a more diverse toolkit than TIP including extensive API management capabilities. This toolkit is also developed with the lessons learned from the TIP platform of the vendor.

3. **Expertise availability:** As integration experts gravitate towards the latest technologies, iPaaS platforms benefit from a larger pool of skilled professionals.

**Easier integration with external parties:** iPaaS platforms facilitate easier integration with external parties by providing secure gateways, simplifying connections with diverse user environments. In contrast, Tip often requires a buffer zone for secure data exchanges, adding complexity and overhead.

**Alignment with Cloud Strategies:** As organizations increasingly adopt cloud computing to fulfil their IT strategies, iPaaS emerges as a compatible cloud-based integration solution

In a greenfield scenario (scenario 1), there are three reasons for organizations to adopt a TIP over an iPaaS:

1. **Organizations' on-premise IT vision and strategy:** If an organization's strategy and plans are firmly rooted in on-premise solutions, without any inclination towards cloud migration, TIP could align better with their needs.
2. **Regulation and compliance issues:** For organizations with highly specific security requirements, TIP allows for complete control over security settings. Regulations and compliance standards, especially those that restrict data transmission over the Internet or data storage in specific locations, can necessitate the use of TIP over iPaaS. It should be noted that this only applies to some very specific scenarios. For most organizations, the iPaaS security features are sufficient, if not better, than the organization's security provisioning.
3. **Control over scalability and performance configurations:** While iPaaS offers advanced features for integration scalability and performance, these are often governed by vendor agreements. TIP offers organizations unrestricted control over these configurations, free from any third-party agreements. Hence, there are edge cases where iPaaS solutions are too inefficient and TIP is superior.

When organizations already have an existing TIP platform (scenario 2). There are two more reasons added to the previous three that make organizations warry of migrating to an iPaaS:

4. **Resource extensive migration process:**
    - **a.** Migrating involves a period in which both TIP and iPaaS have to run beside each other to facilitate undisrupted operations.
    - **b.** Migrating requires redevelopment of existing integrations on the iPaaS platform. Unless the organization stays with the same vendor and that vendor offers migration tools to automate the activity.
    - **c.** Migrating requires new integration experts who can work with the iPaaS. TIP integration experts become less relevant, if not useless. Or they should be trained to use the iPaaS.
5. **The absence of innovation need that covers the entire scope of the organization:** As long as the TIP still functions as an integration platform. It can be difficult to make a business case to invest time and resources into the migration to iPaaS. Often, the reason to reassess the integration platform is when a big application migrates to the cloud. If such a big project remains absent, the urge to migrate is likely to remain absent as well.

The last scenario refers to organizations that have an iPaaS and are considering switching to TIP. Experts indicated that this is a very unlikely scenario and organizations will never completely switch back to TIP but will keep a combination of TIP and iPaaS. The following two reasons are mentioned why an organization would move back from iPaaS to TIP:

1. **Application infrastructure remains on-premise:** The integration platform was the first to migrate to the cloud with the expectation that other applications would follow. This did not happen and now the on-premise infrastructure is integrated with a cloud solution.
2. **False promise**: iPaaS is actively praised by vendors for its low-code solution that should allow citizen integrators to develop integration. In practice, this is not so straightforward. If organizations migrated from TIP to iPaaS with the idea of developing integration with citizen integrators, they might want to switch back to TIP because they had built up expertise there.

# 6 Discussion and Conclusion

This chapter discusses the research and concludes by answering the sub-research questions that contribute to answering the main research question. Section 6.1 discusses notable findings during the research process. Section 6.2 answers the sub and main research questions. Section 6.3 and 6.4 describe the practical and theoretical contribution. Section 6.5 details the limitations of this research. Finally, the recommendations for further research are described in section 6.6.

## 6.1 Discussion

While Enterprise Application Integration (EAI) is a well-studied topic, the sub-field of Integration Platform as a Service (iPaaS) has received comparatively less scholarly attention. The Systematic Literature Review (SLR) described in Section 3.1.1 revealed a sparse body of knowledge on iPaaS, indicating that current research does not comprehensively cover its full spectrum. Since its emergence in 2008, iPaaS has evolved rapidly, potentially outpacing academic analysis. One reason for this might be the commercial origins of iPaaS development, which often limits the public availability of detailed technical information. Another factor contributing to the limited identified research in the SLR could be the use of outdated or incorrect search terms during the literature review, leading to the exclusion of relevant studies. The terminology in IT evolves and terms previously used like "hybrid integration platform" or "cloud integration platform" may now fall under the umbrella of iPaaS. Research that used those terms instead of iPaaS was not included in the SLR due to the used query. Given the sparse knowledge body of knowledge, engagement with industry experts provided deeper insights into iPaaS, underscoring the value of such interactions. In hindsight, this research would have benefited from expert interaction before conducting the literature review. This would have sped up the researcher's knowledge of the topic.

To compare iPaaS and Traditional Integration Platforms (TIP), this research had to make a clear distinction between them through hosting, deployment and management options. However, the lines between these concepts are increasingly blurring, a trend expected to continue. iPaaS offerings are evolving to support on-premise hosting and deployment, enabling organizations to leverage iPaaS capabilities while complying with organizations' TIP-centric requirements, such as data regulations and specific security concerns. As iPaaS is the vendors' latest and greatest, it will continue to outperform TIP in terms of functionalities thereby gaining dominance in the market. Consequently, it seems to be inevitable that iPaaS will increase its presence in the future and for TIP to become less relevant in time.

The expert reviews revealed that there are multiple types of iPaaS offerings. Experts identified two main types: hyper-scalers like Azure Integration Services, Amazon Web Services, and Google Cloud, offering integration services as part of their broader cloud platforms, including Infrastructure as a Service (IaaS) used for comprehensive infrastructure needs. In contrast, vendors like Oracle and SAP provide iPaaS solutions within their suite of applications. Additionally, the study examined a smaller iPaaS provider focusing on smaller organizations and greenfield scenarios. This distinction highlights the variety of iPaaS offerings available, something that was not described in earlier literature.

The practical world of integration does not have a 'one-size-fits-all-all' approach. Solutions that work for one organization do not necessarily work for another. This depends on complex contextual aspects. This also became evident during the expert reviews where, sometimes, contradicting statements were made by different experts on the efficiency of certain solutions. An example of this is the concept of a private cloud architecture (introduced in section 2.5.4). This architecture was excluded from the scope of this research but was mentioned by some experts as a popular approach to use an iPaaS and benefit from the cloud advantages while limiting the cloud disadvantages that are described in section 5.6.2. However, other experts would disagree due to management complexity and

substantial costs incurred with setting up and maintaining a private cloud. Consequently, this research recommends future work into the concept of a private cloud and its influence on iPaaS adoption. This is further described in section 6.6.

Currently, iPaaS is a vendor's most recent evolution of integration product offerings. They actively promote this product and its functionalities and benefits. Furthermore, Gartner introduced the role of a citizen integrator: "business-oriented roles with limited to no IT experience" [12]. iPaaS is promoted as suited to be used by citizen integrators to develop integration through the low-code features. However, all experts included in this research agreed that developing integrations, even with iPaaS, requires technical expertise. They argued, contrary to Gartner, that it is generally not a good idea to let citizen integrators develop integrations. Integration flows that are relatively easy to develop still operate in an organizations infrastructure, together with other integration flows. Therefore, technical expertise in the integration landscape is required to understand the consequences of newly implemented integrations. The low-code features do make the process of integration experts more efficient.

This research aimed to cover the entire spectrum of iPaaS through the functional and non-functional capabilities. While this enables a comprehensive overview, it also simplifies the complex nuances that exist in the practical applications integration platforms. Hence, the capability model and the NFC Tool should function as a starting point for practical users in their decision-making process and for scholars' research as a base for their future research (section 6.6).

## 6.2 Conclusion

### 6.2.1 Sub-Research Questions

1. **How to distinguish between iPaaS platforms and traditional integration platforms?**
    a. *What is a traditional integration platform?*

A TIP is defined as "specialized software tools that help to design, implement, run and monitor integration solutions that orchestrate a set of applications to keep their data synchronized or to develop new functionalities on top of the current ones" [28]. As described in section 4.2, this research adds to this definition that, for a TIP, the *hosting location* is on-premise, *deployment architecture* is on-premise and the *management* is internally organized. These concepts are further detailed in the glossary described in Table 13. The platform functions as a central point of communication within an organizations application landscape. Its functionalities are used by integration developers to develop integration flows that connect an organisation's different applications. Often, this platform is offered by a vendor through a commercialized product. Organizations can acquire a license from the vendor that allows them to deploy the integration platform within their hosting environment.

    b. *What is an iPaaS?*

iPaaS is defined as "a suite of cloud services enabling development, execution and governance of integration flows connecting any combination of on-premises and cloud-based premises, services, applications and data within the individual, or across multiple organizations" [33]. As described in section 4.2, this research adds to this definition that an iPaaS is offered through a PaaS *cloud service model*, the *hosting location* is off-premise, the *deployment architecture* is in a public cloud and the *management* is externally organized. These concepts are further detailed in the glossary described in Table 13. Similar to a traditional integration platform, it is a central point of communication used to develop integration flows. Current literature prescribes several functionalities and benefits to the concept of iPaaS which are summarized in section 3.3 and 3.4. However, this research showed that these are often applicable to integration platforms in general, as described in Table 8, Table 9 and Table 10. Hence, they provide little help in clearly identifying the differences between iPaaS and TIP. This observation was part of the motivation for this research.

c. *How do these concepts differ from each other?*

This research found that iPaaS architectures described by [21] and introduced in section 3.3.2 cause difficulties when classifying integration platforms as iPaaS or TIP. Classification questions arise in, for example, scenarios where a traditional integration platform is hosted on a virtual machine or an iPaaS platform is deployed in an on-premise private cloud. It is debatable in these situations whether platforms should be considered iPaaS or TIP. There is however an important implication of iPaaS that is currently not addressed in both its definition and in iPaaS literature: the Platform-as-a-Service model. Therefore, organizations do not have to *manage* the *hosting* environment in which the iPaaS platform is *deployed.* To summarize this service model implies specific options for the hosting location, deployment architecture and management organization of the integration platform. Therefore, this research distinguishes between TIP and iPaaS based on those aspects listed in Table 36.

| | **TIP** | **iPaaS** |
|---|---|---|
| ***Hosting location*** | On-premise | Off-premise |
| ***Deployment Architecture*** | On-premise | Public cloud |
| ***Infrastructure Management*** | Internally organized | Externally organized |

*Table 36: Distinguishing aspects of TIP and iPaaS*

Section 4.2 covers the design of a glossary that adds these options to the existing definition of TIP and iPaaS. Explicitly defining what constitutes an iPaaS and a TIP within this research context, paves the way for a systematic and insightful analysis of the two concepts. Furthermore, the concepts of TIP and iPaaS are not rigid as a TIP can effectively transform into an iPaaS. An example to clarify this statement is given in section 4.2.

**2. What are the capabilities of integration platforms?**

a. *Which capabilities for integration platforms are described in the literature?*

Capabilities can be split up into two groups, functional and non-functional capabilities [77]. Through a systematic literature review, this research aimed to constitute a first selection of integration platform capabilities of these two groups (section 3.1.2). The review included papers that described functional capabilities and factors influencing Enterprise Application Integration (EAI) adoption in organizations. The former were synthesised to construct a first selection of functional capabilities which is displayed in Appendix C. The latter were synthesised into a selection of factors influencing EAI adoption. Following the method [71], this selection was mapped to the non-functional capabilities in the ISO 25010 framework [45]. The synthesis and mapping resulted in a first selection of seventeen functional and twelve non-functional capabilities that can be found in Appendix C and Appendix D.

b. *Which capabilities for integration platforms can be identified by expert validation?*

The development of the capability model underwent rigorous evaluation through two rounds of expert reviews, details of which are found in Sections 4.3 and 5.2 respectively. Initially, while experts did not object to the capability selection derived from the literature, they recommended additions and structural refinements that are detailed in Table 17 and Table 18. While refining the selection of capabilities, this research adhered to the capability guidelines described by [51]. For the functional capabilities, the primary modification involved categorizing them into six groups: API Management, Messaging, File Transfer, B2B, Data Management, and General. Although these capabilities were all sourced from literature, experts proposed incorporating additional sub-capabilities. Following this feedback, nine sub-capabilities were added, bringing the total to 26 functional capabilities. For non-functional capabilities, the key adjustment was the consolidation of several capabilities into more

cohesive groups. After these modifications were validated by experts, the total count of non-functional capabilities stood at nine.

The first comprehensive draft of the capability model emerged from the initial review, as documented in Section 4.4. Subsequent minor adjustments during the second round of reviews culminated in unanimous expert agreement on the model's completeness, as documented in Section 5.3. This research recognizes that the capability model is not definitive; further sub-capabilities could be integrated. However, for this study, the model provides a thorough framework for comparing TIP and iPaaS technologies.

3. **What are the use cases of organizations that have to select between a traditional integration platform and iPaaS?**

When selecting an integration platform, organizations participate in a decision-making process. Expert reviews described in section 4.3.2 concluded that this process is complex and requires analysis of the organization's current, and future needs and resources. This study refrained from attempting to formulate detailed use cases of how organizations make these decisions. However, based on the expert interviews, this research found a primary differentiator: whether an organization already possesses an integration platform.

If organizations have an existing integration platform, the consideration becomes whether they should migrate to a different type of platform, be it from TIP to iPaaS or vice versa. Hence the first scenario constructed is when organizations already have an integration platform. If the organization is without a pre-existing integration platform, it first decides if it wants to have an integration platform at all. Upon establishing this need, the question then shifts to selecting between TIP and iPaaS. This decision is the second scenario used as a basis for evaluating the relative advantages and disadvantages of TIP versus iPaaS. These scenarios are further described in section 4.3.2.

It is also crucial to acknowledge that some organizations may deploy multiple integration platforms, potentially including both TIP and iPaaS from various vendors. These scenarios, involving multi-platform integration strategies and the potential for conflicts or synergies between platforms, are inherently complex and were not explicitly covered in this study due to their intricate nature. Nevertheless, the insights gained from this research can still be applied to these more complex scenarios as an initial step. A comprehensive comparison in such cases, however, would require further exploration of additional factors. By recognizing these complexities, this study contributes to a broader understanding of integration technology deployment, even though such detailed scenarios are beyond its current scope beyond its current scope.

4. **How can the capability model be used to compare iPaaS and traditional integration platforms?**

The capability model is applied in a case study that assesses and compares the TIP and iPaaS offerings of a specific vendor. This case study, described in section 5.5, is performed with an integration expert who has extensive technical knowledge of the TIP and iPaaS offerings of the vendor. In one session, the expert was asked to walk through the functional capabilities of the Capability Model and compare the TIP and iPaaS offerings for each capability. As such, the Capability Model structured this comparison. The case study was performed for two different vendors, Microsoft and SAP. To improve the process of applying the capability model this way, it would be helpful to have a more diverse group of experts to foster discussion and to have access to the actual integration platform so real-life examples can be shown.

Where the functional capabilities are connected to a specific vendor, the non-functional capabilities are better suited to actually compare the concept of iPaaS and TIP generally. Therefore, these non-functional capabilities are further worked out in the NFC Tool. This is a PowerPoint deck that

compares iPaaS and TIP per non-functional capability. The textual content of the tool is displayed in Appendix G. The information in the deck is validated in section 5.4 and adjusted based on expert reviews that rated the tool on defined criteria from 1 (strongly disagree) to 5 (strongly agree). During this evaluation, integration experts rated the information in the tool. Based on the results it can be concluded that the information is understandable, sufficient, accurate, useful and usable. Furthermore, the following statement was evaluated with an average score of 4,5. Experts agreed that his model improves the understanding of the differences between iPaaS and on-premise integration platforms.

In an Ex Post Evaluation, the gathered information from the expert review is applied to the scenarios formulated through sub-research question three. To provide recommendations to organizations on the decision between TIP and iPaaS. This research concludes that it is an 'iPaaS unless' decision. This means that iPaaS is generally favoured unless organizations have some specific reason to choose TIP. Section 5.6.2 described the reasons for iPaaS to generally be superior over TIP and the specific reasons per scenarios for TIP to possibly be favoured over iPaaS. It's important to note that the choice between TIP and iPaaS should be based on a thorough analysis of the organization's specific needs, resources, and strategic objectives. The Capability Model and NFC Tool can provide a first direction into formulating these needs of organizations.

### 6.2.2 Main Research Questions

The previously described answers to the sub-research questions contribute to answering the main research question that this research aims to answer:

*What are the functional and non-functional capabilities required for integration platforms and how do iPaaS (Integration Platform as a Service) and traditional integration platforms differ from each other in these capabilities?*

Based on the results obtained in this research, it can be concluded that this research succeeded in identifying the functional and non-functional capabilities and developing them into a capability model. In the second expert review, all experts agreed that the list of capabilities was complete. This capability model was the starting point to further compare iPaaS and TIP. This comparison is performed on several different levels.

First, the capability model itself proved an effective starting point to assess the integration platform offerings of a certain vendor. Together with experts, the functional capabilities of the capability model were worked through, providing an overview of how those two platforms compare. For the two platforms included, it can be concluded that for almost all capabilities, iPaaS provides a similar or better feature than TIP. Hence, iPaaS is superior to TIP from a functional perspective.

Secondly, the NFC tool further compares iPaaS and TIP on their non-functional capabilities. The comparison shows that iPaaS has several advantages over TIP but also highlights certain aspects that organizations should take into account when adopting an iPaaS. Some examples are the importance of the specific vendor agreement and the inability to develop flows by citizen integrators even though that is promoted by iPaaS vendors. The tool also shows the added complexity of setting up and maintaining a TIP. The tool was rated by experts on different evaluation criteria. The rating was based on a 5-point Likert scale and averaged between 3,7 and 4,9.

## 6.3 Contribution to Practice

The contribution of this research to practice is twofold. First, the thesis presents a validated capability model for integration platforms that includes both functional and non-functional capabilities. This model was developed with extensive input from industry practice across all phases of the research, ensuring its completeness and applicability in real-world settings. Practitioners can use this model to structure assessments and comparisons of integration platforms, as demonstrated during a case study.

It also supports Deloitte consultants in understanding and structuring their clients' requirements for integration platforms.

Secondly, the NFC tool offers practitioners a quick and effective way to compare the most critical aspects when deciding between TIP and iPaaS. This tool, also developed and refined through practical input, guides organizations in their decision-making process for choosing an integration platform and is useful for Deloitte consultants advising these organizations. The NFC tool has been evaluated by integration experts, receiving high scores for its accuracy, usability, and understandability, further indicating its value in supporting informed decision-making processes.

## 6.4 Contribution to Research

This research makes several significant contributions to the scientific community. Firstly, it addresses the recognized need by Ebert [21] and Hyrynsalmi [42] for a structured comparison between Traditional Integration Platforms (TIP) and Integration Platforms as a Service (iPaaS). This study uniquely structures this comparison through a detailed research design (outlined in section 1.4), thus filling a notable gap in the existing literature.

Secondly, while Ebert [21] highlighted the architectural variants of iPaaS (outlined in the section 3.3.2), there was confusion about whether these variants qualify as iPaaS. This research clarifies these ambiguities by extending the definition of iPaaS to include specific aspects of hosting, deployment, and management. These refined definitions are consolidated into a comprehensive glossary (outlined in section 4.2), providing a clear framework that future research can utilize to distinguish between iPaaS and TIP more effectively.

Thirdly, as concluded after the literature review on iPaaS (outlined in section 3.5), the existing literature on iPaaS lacked a structured model to thoroughly describe and analyze iPaaS, particularly in detailing its functionalities and differentiating its benefits and drawbacks compared to TIP. This research developed a capability model (outlined in section 5.3) that not only describes the structural components of iPaaS but also introduces a Non-Functional Capability Tool (outlined in section 5.4) that compares iPaaS with TIP. These tools were rigorously tested and validated across two iterations through expert reviews, marking the first known study to comprehensively compare both the functional and non-functional capabilities of these platforms. Through two expert reviews, this research infused practical insights into the academic discourse. This approach has effectively bridged the theoretical and practical realms within the domain of Enterprise Application Integration (EAI), enriching the academic literature with grounded, actionable knowledge.

## 6.5 Limitations

The validity of the systematic literature review may be influenced by the exclusion of potentially relevant papers, a common challenge in any literature review. To mitigate this threat, multiple databases were utilized to ensure a more comprehensive coverage of available literature. A notable limitation of the initial review was the exclusion of potential iPaaS synonyms such as 'hybrid integration platform' or 'cloud integration platform' from the search queries. While the inclusion of such synonyms might have brought additional sources, these are likely rather older sources. This is because the term 'iPaaS' has become more standardized which means that the most current and applicable insights are likely to be found in works that explicitly use this term. Thus, while the exclusion of synonyms could be seen as a limitation, its impact on the applicability of the research findings may be minimal.

The synthesis and the mapping of the capability models in the design and development phase both pose threats to the validity of the research because of the subjectivism that is included in the mapping of constructs. This threat is addressed as best as possible by adhering to a standard protocol and

having the resulting selection of capabilities validated by different panels of experts twice in this research. Furthermore, the mapping choices are described allowing for reflection on these decisions.

This research relied heavily on expert reviews and interviews to validate the NFC tool and capability model. This method introduces potential biases. Efforts were made to select a diverse panel, of experts, but there is always a risk that these experts are biased and that their experiences influenced their objective judgement and therefore the findings of this research. This research acknowledges that a different selection of experts could give different insights.

This research was not able to demonstrate the NFC Tool and the Capability Model in a real-life application. Preferably, the NFC Tool and/or the capability model were used in an organization that was in a platform migration process. Currently, the artefacts were only demonstrated in an artificial setting. This was more straightforward as a demonstration in a real-world setting would require specific organizations that were in the midst of an integration platform decision process. Furthermore, this research did not have enough time to prepare and conduct that type of validation. Acknowledging this limitation underscores the need for further research focusing on practical implementations.

The researcher's involvement in the case study introduces potential bias and limitations. The primary limitation lies in the dependency on the participating expert's knowledge and the researcher's ability to follow up on made inquiries. This research acknowledges that other pairings of experts or researchers might provide different outcomes when comparing the TIP and iPaaS of a certain vendor. Moreover, the direct involvement of the researcher with the capability model ensured a clear understanding of its application, eliminating any confusion regarding its use. This familiarity poses a challenge for individuals less engaged with the model, who may wish to replicate the case study. For such individuals, navigating the capability model without the benefit of deep involvement in its development could introduce difficulties in its application and interpretation.

One of the limitations of this research is the generalizability of the findings over time. Advancements in the world of integration go fast and the information derived from experts as well as the comparison made between TIP and iPaaS are a snapshot of the current situation. If this comparison had been made when iPaaS was first launched in 2008, the results would have likely been different due to the immaturity of the platform. Following this line of reasoning, the results of a future comparison will likely be different than the conclusions of this research as a result of the fast development of iPaaS.

## 6.6 Recommendations for Future Work

This research suggests several promising directions for future research related to this study conducted on the concept of TIP and iPaaS and the developed artefacts.’

First and foremost, future research should apply the NFC Tool and Capability Model in real-life scenarios. The former could be applied to organizations that are in the greenfield and starting the decision-making process for their integration platform. It is difficult to quantify the effect of the tool on this process. A suggestion is to use consultants who are guiding organizations in their decisions for an integration platform. They could perform these projects with the tool and without to see how they experience the effectiveness of the tool. One point of attention that was noticed during expert reviews in this research is that the tool is lengthy and text-heavy. Hence, future research should aim to increase the practical usability by making the tool easier to use. The latter can be used within a platform comparison, similar to the case study described in section 5.5. This research did not develop a standardized process to apply the Capability Model as a structure tool so future attention should be directed to how this tool can more effectively be used in a platform comparison.

This research explained the greenfield scenario (section 4.3.2) in which organizations do not yet have an integration platform. To make these scenarios relevant for this research, the assumption was made that organizations in these scenarios have already decided that they want to have an integration platform. However, in practice, this is not likely to be the case. Therefore, future research should be

directed to the business benefits that justify purchasing/hiring an integration platform. Possibly, research has been conducted in the earlier period of the EAI domain but this research did not consider the availability of iPaaS.

Future research should explore scenarios where organizations employ multiple integration platforms, such as a combination of TIP and iPaaS, potentially from different vendors. As described in section 4.3.2, this study identified these scenarios as too complex to analyze due to the need for a detailed examination of multi-platform integration strategies. However, investigating these complexities is important for a comprehensive understanding of integration platform deployment. Future work could provide valuable insights by focusing on how different integration platforms interact within diverse organizational contexts, and what the added benefits and drawbacks are of these hybrid integration platform scenarios.

Future research should further investigate the concept of a private cloud (section 2.5.4) and how that influences iPaaS adoption. This research did not include this concept in its comparison. Furthermore, experts had disagreeing statements on the practical applicability of the private cloud. Hence, it is an interesting direction for future research to investigate this concept as it can bridge the functionality of the iPaaS platform with the security requirements of a TIP platform.

The current decision-making process for organizations to choose an integration platform is, as described by experts (section 4.3), complex and highly dependent on specific aspects. Hence, it would be valuable to conduct research on this specific process to investigate why this process is complex. Possibly, succeeding in developing a more standardized approach to deciding between integration platforms. It could use this research as a source of information for important aspects to consider when deciding between TIP and iPaaS.

This research showed that iPaaS is very likely to be dominant compared to TIP. This is the ‘why’ reason for organizations to migrate from TIP to iPaaS. Future research should be conducted to investigate ‘how’ organizations should migrate to iPaaS and ‘when’ the timing is good to make this migration. This costly migration process is identified as one of the reasons for organizations do not migrate to a potentially better platform. By investigating this process, organizations are supported in this process, making it less of an obstacle to migrate to another platform.

# Bibliography


[1] *2023 Digital Transformation - Connectivity Benchmark Report. (n.d.).* Available at: *https://resources.Mulesoft.com/ty-report-connectivity-benchmark.html*

[2] Al-Balushi, F. M., Bahari, M., & Rahman, A. A. (2015). A conceptual framework for the implementation process of Enterprise Application Integration (EAI) in government. *ARPN Journal of Engineering and Applied Sciences*, *10*(23), 17871-17880, doi: 10(23):17871-17880

[3] Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., ... & Zaharia, M. (2010). A view of cloud computing. *Communications of the ACM*, *53*(4), 50-58, doi: 10.1145/1721654.1721672.

[4] Aserey, N., & Alshawi, S. N. (2013). A conceptual model of enterprise application integration in higher education institutions.

[5] Balasubramanian, R., & Aramudhan, M. (2012). Security issues: public vs private vs hybrid cloud computing. *International Journal of Computer Applications*, *55*(13).

[6] Barney, N., & Nolle, T. *(2023, May 2).* service-oriented architecture (SOA). App Architecture, Available at: *https://www.techtarget.com/searchapparchitecture/definition/service-oriented-architecture-SOA*

[7] Bigelow, S.J. (2023) *What is ipaas? guide to integration platform as a service*, *Cloud Computing*. Available at: https://www.techtarget.com/searchcloudcomputing/definition/iPaaS-integration-platform-as-a-service (Accessed: 01 February 2024).

[8] Bishop, T. A., & Karne, R. K. (2003, March). A survey of middleware. In *CATA* (pp. 254-258).

[9] Blanchard, E. G., & Allard, D. L. (Eds.). (2010). Handbook of Research on Culturally-Aware Information Technology: Perspectives and Models: Perspectives and Models. *IGI Global*, doi: 10.4018/978-1-61520-883-8.

[10] Bolloju, N., & Murugesan, S. (2012, August). Cloud-based B2B systems integration for small and medium-sized enterprises. In *Proceedings of the International Conference on Advances in Computing, Communications and Informatics* (pp. 477-480), doi: 10.1145/2345396.2345475.

[11] Brasier, M. and Olliffe, G. (2021). Choosing Application Integration Platform Technology *Gartner RAS Core Research*, available at: https://www.gartner.com/en/documents/3999171.

[12] Thoo, E. Pezzini, M. Golluscio, E. (2018). Citizen Integrators Bring Application and Data Integration Into a Common Focus. *Gartner RAS Corse Research*, available at: https://www.gartner.com/en/documents/3725217

[13] Boomi announces AtomSphereSM: Industry's first integration platform-as-a-service. Boomi. (2016, October 4), available at: *https://boomi.com/press_release/boomi-announces-atomsphere%e2%84%a0-industrys-first-integration-platform-service/*

[14] Cestari, R. H., Ducos, S., & Exposito, E. (2020, September). IPaaS in agriculture 4.0: an industrial case. In *2020 IEEE 29th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)* (pp. 48-53). IEEE, doi: 10.1109/WETICE49692.2020.00018.

[15] Chappell, D. A. (2004). Enterprise service bus: Theory in practice. *" O'Reilly Media, Inc."*.

[16] Cloud, H. (2011). The first definition of cloud computing. *National Institute of Science and Technology, Special Publication*, *800*(2011), 145, available at: https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-145.pdf.

[17] Columbus, L. *(2020, March 29). The State Of Enterprise Data Integration, 2020. Forbes. https://www.forbes.com/sites/louiscolumbus/2020/03/29/the-state-of-enterprise-data-integration-2020/?sh=7c26875c4665*

[18] Cybersecurity, Peris. (2023) *Exploring contrasts between on-premises and Cloud Cybersecurity*, *LinkedIn*. Available at: https://www.linkedin.com/pulse/exploring-contrasts-between-on-premises-cloud-cybersecurity/ (Accessed: 27 February 2024).

[19] De, B., & De, B. (2017). *API management* (pp. 15-28). Apress. Doi: 10.1007/979-8-8688-0054-2.

[20] Dey, I. (2003). Qualitative data analysis: A user friendly guide for social scientists. *Routledge*.

[21] Ebert, N., Weber, K., & Koruna, S. (2017). Integration platform as a service. *Business & Information Systems Engineering*, *59*, 375-379, doi: 10.1007/s12599-017-0486-0.

[22] Etro, F. (2009). The economic impact of cloud computing on business creation, employment and output in Europe. Review of Business and Economics, *54(2), 179-208.*

[23] *Explaining SAP process orchestration*. Learn SAP skills. (n.d.). https://learning.sap.com/learning-journeys/discovering-sap-process-orchestration/explaining-sap-process-orchestration_f714636a-a6dd-40c0-a60e-71786eaf3616

[24] *Exploring SAP integration capabilities - zero to one* (2024) *SAP Community*. Available at: https://community.sap.com/t5/technology-blogs-by-sap/exploring-sap-integration-capabilities-zero-to-one/ba-p/13566475 (Accessed: 20 March 2024).

[25] Flower, Z. (2024, January 19). *On premises vs. Cloud Pros and Cons, key differences: TechTarget*. Cloud Computing. https://www.techtarget.com/searchcloudcomputing/tip/Evaluate-on-premises-vs-cloud-computing-pros-and-cons

[26] Fowler, M. (2014). Microservices, martinfowler.com. Available at: https://martinfowler.com/articles/microservices.html (Accessed: 20 February 2024).

[27] Frantz, R. Z., Corchuelo, R., Basto-Fernandes, V., Rosa-Sequeira, F., Roos-Frantz, F., & L. Arjona, J. (2021). A cloud-based integration platform for enterprise application integration: A Model-Driven Engineering approach. *Software: Practice and Experience*, *51*(4), 824-847, https://doi.org/10.1002/spe.2916

[28] Freire, D. L., Frantz, R. Z., & Roos-Frantz, F. (2019). Ranking enterprise application integration platforms from a performance perspective: An experience report. *Software: Practice and Experience*, *49*(5), 921-941, https://doi.org/10.1002/spe.2679

[29] Gadge, S., & Kotwani, V. (2018). Microservice architecture: API gateway considerations. *GlobalLogic Organisations, Aug-2017, 11*. Available at: https://www.globallogic.com/wp-content/uploads/2017/08/Microservice-Architecture-API-Gateway-Considerations.pdf

[30] Gopalakrishnan, P., & Maheswari, B. U. (2019). Research on enterprise public and private cloud service. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, *8*(6S4), 1453-1459, doi: 10.35940/ijitee.f1296.0486s419

[31] Gorelik, E. (2013). *Cloud computing models* (Doctoral dissertation, Massachusetts Institute of Technology), available at: https://web.mit.edu/smadnick/www/wp/2013-01.pdf

[32] Guttridge, K., Comes, A., Ray, S. (2023). Critical Capabilities for Integration Platform as a Service, Worldwide, available at: https://www.gartner.com/en/documents/4142799.

[33] Guttridge, K., Comes, A., Ray, S. (2023). Magic quadrant for Integration Platform as a Service, Worldwide, available at: https://www.gartner.com/en/documents/5198963.

[34] Hevner, A., & Chatterjee, S. (2010). *Design research in information systems: theory and practice* (Vol. 22). Springer Science & Business Media, doi: 10.5555/1859261.

[35] Hohpe, G., & Woolf, B. *(2004*). Enterprise integration patterns: Designing, building, and deploying messaging solutions. *Addison-Wesley Professional*, doi: 10.5555/940308.

[36] Hsu, C. L., & Lin, J. C. C. (2016). Factors affecting the adoption of cloud services in enterprises. *Information Systems and e-Business Management*, *14*, 791-822, doi: 10.1007/s10257-015-0300-9

[37] Hung, W. H., Chang, I., Yen, D. C., & Lee, C. M. (2015). Critical factors of adopting enterprise application integration technology: an empirical study on larger hospitals. *Communications of the Association for Information Systems*, *36*(1), 31, doi: 10.17705/1CAIS.03631

[38] Hwang, K., Shi, Y., & Bai, X. (2014, December). Scale-out vs. scale-up techniques for cloud performance and productivity. In *2014 IEEE 6th International Conference on Cloud Computing Technology and Science* (pp. 763-768). IEEE, doi: 10.1109/CloudCom.2014.66.

[39] Hyrynsalmi, S. (2022). Pathway to the successful integration platform management, doi: 10024/165600.

[40] Hyrynsalmi, S. M. (2022, May). The State-of-the-Art of the Integration Platforms as a Service research. In *Proceedings of the 5th International Workshop on Software-intensive Business: Towards Sustainable Software Business* (pp. 17-22), doi: 10.1145/3524614.3528634.

[41] Hyrynsalmi, S. M. (2022, October). Definition of the Enterprise Integration Platforms as a Service—Towards a Common Understanding. In *International Conference on Software Business* (pp. 167-181). Cham: Springer International Publishing, doi: 10.1007/978-3-031-20706-8_12

[42] Hyrynsalmi, S. M., Koskinen, K. M., Rossi, M., & Smolander, K. (2021, June). Towards the utilization of Cloud-based Integration Platforms. In *2021 ieee international conference on engineering, technology and innovation (ice/itmc)* (pp. 1-8). IEEE, doi: 10.1109/ICE/ITMC52061.2021.9570235.

[43] Hyrynsalmi, S., & Smolander, K. (2023). Between the Rock and the Hard Place-Conflicts in Implementing Integration Platforms.

[44] *Integration Platform As A Service (IPAAS) Market Size, Scope & Forecast. (2023, August 18). Verified Market Research. https://www.verifiedmarketresearch.com/product/integration-platform-as-a-service-iPaaS-market/*

[45] *International Organization for Standardization. (2011). Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models (ISO/IEC 25010:2011). https://www.iso.org/standard/78176.html*

[46] Kamal, M., & Themistocleous, M. (2006). A conceptual model for EAI adoption in an e-government environment.

[47] Karsh, P. (2023) *Comparing load balancer configurations: Active-passive vs. active-active, Medium*. Available at: https://patrickkarsh.medium.com/comparing-load-balancer-configurations-active-passive-vs-active-active-4900acb7af87 (Accessed: 27 February 2024).

[48] Keen, M., Bishop, S., Hopkins, A., Milinski, S., Nott, C., Robinson, R., ... & Acharya, A. (2020). Patterns: Implementing an SOA Using an Enterprise Service Bus, IBM Redbooks (2004). *URL http://ibm. com/redbooks*.

[49] Kewear. *(2023, January 12). Migration approaches: BizTalk Server to Azure Integration Services - Azure Logic Apps. Microsoft Learn. https://learn.microsoft.com/en-us/azure/logic-apps/BizTalk-server-azure-integration-services-migration-approaches*

[50] Kusák, D. (2010). Comparison of Enterprise Application Integration Platforms, doi: 20.500.11956/33964.

[51] Lankhorst, M. (2016 ). *blog/archimate-3-0-capability-mapping/. Available at: bizzdesign.com: https://bizzdesign.com/blog/archimate-3-0-capability-mapping/ (Accessed: 17 April 2024).*

[52] Linthicum, D. S. (2000). *Enterprise application integration*. Addison-Wesley Professional, doi: 0201615835

[53] Mantzana, V., & Themistocleous, M. (2005). Towards a conceptual framework of actors and factors affecting the EAI adoption in healthcare organizations, available at: https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1100&context=ecis2005

[54] Marian, M. (2012). iPaaS: Different ways of thinking. *Procedia Economics and Finance*, *3*, 1093-1098, doi: https://doi.org/10.1016/S2212-5671(12)00279-1.

[55] Marston, S., Li, Z., Bandyopadhyay, S., Zhang, J., & Ghalsasi, A. (2011). *Cloud computing — The business perspective. Decision Support Systems, 51(1), 176-189. https://doi.org/10.1016/j.dss.2010.12.006*

[56] Mell, P., & Grance, T. (2011). *The NIST definition of cloud computing. National Institute of Standards and Technology*.

[57] *Mendix. (2023, August 31). What is Low-Code Development? | Mendix.*

[58] Menge, F. (2007, August). Enterprise service bus. In *Free and open source software conference* (Vol. 2, pp. 1-6), available at: https://www.semanticscholar.org/paper/Enterprise-Service-Bus-Menge/75393c46ab62d8c25d13f79d68ef42e232474b53?utm_source=direct_link

[59] Microsoft Azure Integration Services. (2018). *Azure Integration Services.* https://azure.microsoft.com/mediahandler/files/resourcefiles/azure-integration-services/Azure-Integration-Services-Whitepaper-v1-0.pdf

[60] Moradi, H., & Bahreininejad, A. (2013). Toward a Comprehensive Framework for Evaluating the Core Integration Features of Enterprise Integration Middleware Technologies. *Journal of Systems Integration (1804-2724)*, *4*(1), 10.20470/JSI.V4I1.150.

[61] Narayan, P. (2021, March 31). *Consumption vs fixed pricing: What's it (really) going to cost to move your data and apps?* LinkedIn. https://www.linkedin.com/pulse/consumption-vs-fixed-pricing-whats-really-going-cost-move-narayan/

[62] Neifer, T., Lawo, D., Bossauer, P., & Gadatsch, A. (2021, July). Decoding IPaaS: Investigation of User Requirements for Integration Platforms as a Service. In *ICE-B* (pp. 47-55), doi: 10.5220/0010626000002997.

[63] Newman, S. (2021). Building microservices. " O'Reilly Media, Inc.. doi: 10.5555/2904388

[64] Nolle, T. *(2015, August 7). Your key products for enterprise application integration. App Architecture. https://www.techtarget.com/searchapparchitecture/feature/Your-key-products-for-enterprise-application-integration*

[65] Okoli, C., Schabram, H. (2010). A guide to conducting a systematic literature review of information systems research. Doi: 10.2139/ssrn.1954824.

[66] Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of management information systems*, *24*(3), 45-77. Doi: https://doi.org/10.2753/MIS0742-1222240302

[67] Pezzini, M. and Lheureux, B. (2011). Integration platform as a service: moving integration to the cloud. *Gartner RAS Core Research Note G*, *210747:7.* Available at: https://www.gartner.com/en/documents/1575414

[68] Phan, D. H., Suzuki, J., Omura, S., & Oba, K. (2013, September). Toward sensor-cloud integration as a service: Optimizing three-tier communication in cloud-integrated sensor networks. In *Proceedings of the 8th International Conference on Body Area Networks* (pp. 355-362). https://doi.org/10.4108/icst.bodynets.2013.253639

[69] Potočnik, M., & Juric, M. B. (2012, October). Integration of SaaS using iPaaS. In *Proceedings of the 1st International Conference on CLoud Assisted ServiceS* (pp. 35-51).

[70] Puschmann, T., & Alt, R. (2001, January). Enterprise application integration-the case of the Robert Bosch Group. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences* (pp. 10-pp). IEEE. Doi: 10.1109/HICSS.2001.927242.

[71] Rahman, M. S., & Reza, H. (2020, July). Systematic mapping study of non-functional requirements in big data system. In *2020 IEEE International Conference on Electro Information Technology (EIT)* (pp. 025-031). IEEE. Doi: 10.1109/EIT48999.2020.9208288

[72] Rani, D., & Ranjan, R. K. (2014). A comparative study of SaaS, PaaS and IaaS in cloud computing. *International Journal of Advanced Research in Computer Science and Software Engineering*, *4*(6).

[73] Raut, R. D., Gardas, B. B., Jha, M. K., & Priyadarshinee, P. (2017). Examining the critical success factors of cloud computing adoption in the MSMEs by using ISM model. *The Journal of High Technology Management Research*, *28*(2), 125-141. Doi: https://doi.org/10.1016/j.hitech.2017.10.004.

[74] Rebelo Moreira, J.L. (2022). Architecture of Information Systems (AIS) [slide 6]. Faculty of EEMCS, University of Twente. https://www.utwente.nl/en/ces/sal/exams/architecture-of-Information-systems/ais-workshop1-nfreq.pdf

[75] Rebelo Moreira, J.L. (2022). Microservices [Slide 4]. Faculty of EEMCS, University of Twente.

[76] Requirements and capabilities (no date) *IBM. Available at: https://www.ibm.com/docs/en/rsas/7.5.0?topic=overview-requirements-capabilities (Accessed: 17 April 2024).*

[77] Reiff-Marganiec, S. (Ed.). (2011). Handbook of Research on Service-Oriented Systems and Non-Functional Properties: Future Directions. IGI Global.

[78] Richards, M. (2015). Microservices vs. service-oriented architecture (pp. 22-24). Sebastopol: O'Reilly Media. Available at: https://www.oreilly.com/radar/microservices-vs-service-oriented-architecture/

[79] Safari, F., Safari, N., & Hasanzadeh, A. (2015). The adoption of software-as-a-service (SaaS): ranking the determinants. *Journal of Enterprise Information Management*, *28*(3), 400-422, doi: 10.1108/JEIM-02-2014-0017

[80] Sänger, N., & Abeck, S. (2022). Authentication and Authorization in Microservice-Based Applications. *INFORMATIK 2022*. Doi: 10.18420/inf2022_19

[81] *SAP help portal*. (*n.d.)* Available at: https://help.sap.com/docs/integration-suite/sap-integration-suite/working-with-integration-suite-home (Accessed: 20 March 2024).

[82] SAP help portal. (n.d.). https://help.sap.com/doc/saphelp_nw73ehp1/7.31.19/en-us/f1/24e6e6f548480b85197bde372d13c9/frameset.htm

[83] Srimathi, H. (2019). Integration Of Student System Using iPaaS. *INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 8, ISSUE 09, SEPTEMBER 2019*. Available at: https://www.semanticscholar.org/paper/Integration-Of-Student-System-Using-iPaaS-Srimathi-Krishnamoorthy/cb8b26f8324e169a7cd037696e3c0cb3df69b981

[84] *Sultan, N. (2011). Reaching for the "cloud": How SMEs can manage. International Journal of Information Management, 31(3), 272-278. https://doi.org/10.1016/j.ijinfomgt.2010.08.001*

[85] Theilig, M. M., Pröhl, T., & Zarnekow, R. (2018). Requirements Analysis for an Open iPaaS: Exploring the CSP, ISP, and SME View. Doi: not available.

[86] Themistocleous, M. (2004). Justifying the decisions for EAI implementations: a validated proposition of influential factors. *Journal of Enterprise Information Management*, *17*(2), 85-104. Doi: 10.1108/17410390410518745.

[87] Venable, J., Pries-Heje, J., & Baskerville, R. (2016). FEDS: a framework for evaluation in design science research. *European journal of information systems*, *25*, 77-89. Doi: 10.1057/ejis.2014.3.

[88] *What is cloud computing? | IBM. (n.d.). https://www.ibm.com/topics/cloud-computing*

[89] *What is EDI: Electronic Data Interchange? (no date) IBM. Available at: https://www.ibm.com/topics/edi-electronic-data-interchange (Accessed: 15 November 2023).*

[90] *What is private cloud?* (no date) *IBM*. Available at: https://www.ibm.com/topics/private-cloud (Accessed: 24 January 2024).

[91] Wieringa, R. J. (2014). Design science methodology for information systems and software engineering. Springer.Microservices. In *2016 4th International Conference on Enterprise Systems (ES)* (pp. 60-67). IEEE. Doi: 10.1007/978-3-662-43839-8

[92] Yang, Z., Sun, J., Zhang, Y., & Wang, Y. (2015). Understanding SaaS adoption from the perspective of organizational users: A tripod readiness model. *Computers in Human Behavior*, *45*, 254-264. Doi: https://doi.org/10.1016/j.chb.2014.12.022

[93] Zhang, X., & Yue, W. T. (2020). Integration of on-premises and cloud-based software: the product bundling perspective. *Forthcoming in Journal of the Association for Information Systems*. Doi: 10.17705/1jais.00645

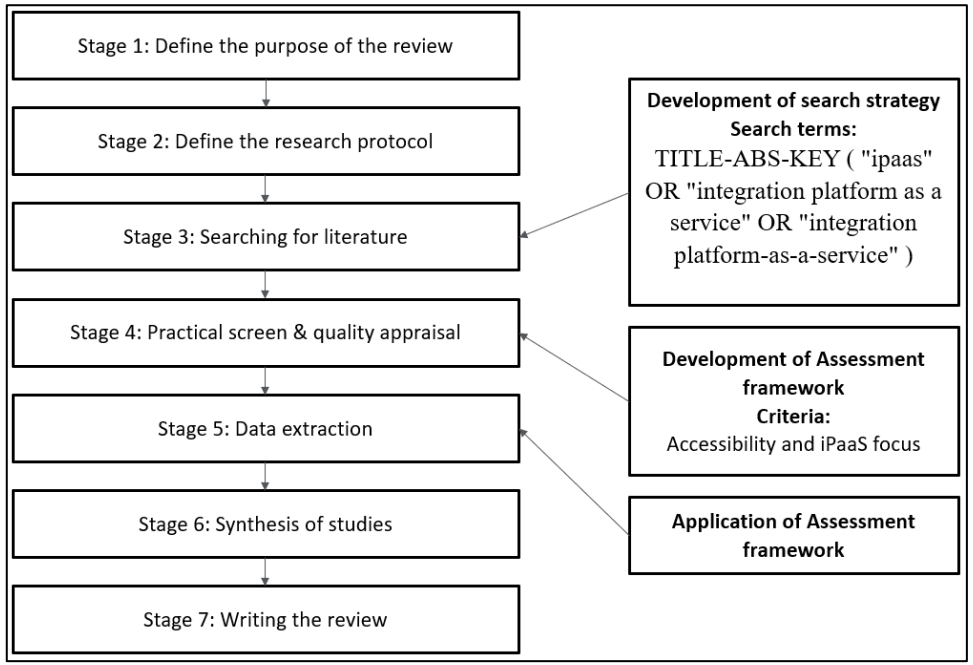
# Appendices

## Appendix A The Benefits and Drawbacks of the Different Integration Styles

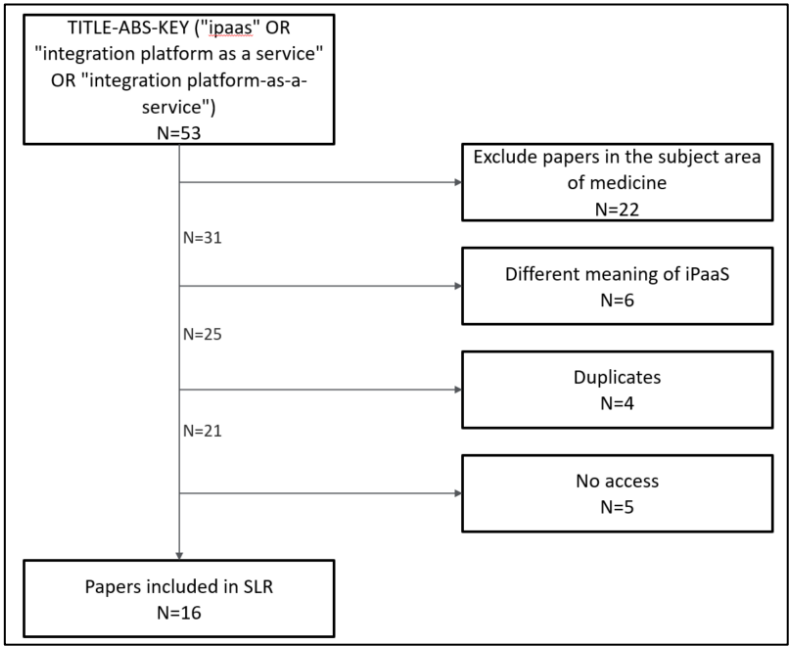
| *Approach* | *Description* | *Benefits* | *Drawbacks* |
|---|---|---|---|
| *File/data transfer* | Use a common data transfer mechanism that can be used by any language and platform. | Integrators need no knowledge of applications<br><br>Applications are decoupled from each other<br>Simple, no extra packages are needed | Decide what format to use<br><br>Manual file processing by developers<br><br>When files be produced and consumed?<br><br>Updates occur infrequently resulting in stale information |
| *Shared database* | A central, agreed-upon datastore that all of the applications share, so that any of them have access to any of the shared data whenever they need it. | Data consistency<br><br>No timeliness | Developing a suitable design for the database (both technical and political)<br><br>External packages that often do not work<br><br>Performance issues when multiple applications read and modify the same data |
| *Remote procedure invocation* | Applies the principle of encapsulation. Applications make direct calls to each other when they need data to be sent or modified. | Applications maintain data integrity<br><br>Applications can alter internal data without other applications being affected | Tightly coupled applications<br><br>Remote calls can fail which can affect the entire system |
| *Messaging* | Transfer packets of data frequently, immediately, reliably, and asynchronously, using customizable formats | Allows for decoupling<br><br>Asynchronous messaging<br><br>Collaborative behaviour of applications | Semantic dissonance occurs (can be overcome with transformations but still applies)<br><br>Less fast dan RPI but still relatively fast<br><br>Inconsistency problems can still occur but are reduced substantially compared to file transfer<br><br>Increased development effort to design transformations |

# Appendix B Systematic Literature to iPaaS

**The Systematic Literature Review Protocol:**



**Data Extraction Steps:**



**iPaaS summary of included sources:**

| Ref | Author | Theme | Industry | Method | Results validation |
|---|---|---|---|---|---|
| **[10]** | Bolloju and Murugesan (2012) | They are describing that Small and Medium-sized Enterprises (SMEs) are falling behind in adopting a cloud-based approach for the integration of their application. Therefore, they argue that SMEs should invest more in the cloud and they outline several brief approaches and recommendations for adopting an iPaaS in an SME. | SMEs | Literature review | No |
| **[54]** | Marian (2012) | Combining several grey literature sources into an analysis of the concept of iPaaS. However, few scientific sources are presented and a proper research method is lacking. | NA | No method presented | No |
| **[69]** | Potocnik (2012) | Arguing that integration problems arise from the absence of proper mechanisms to integrate SaaS applications that are conceptually different from on-premise applications. They present iPaaS as the solution and highlight several requirements that belong to iPaaS. | SaaS users | No method presented | No |
| **[68]** | Phan (2013) | Developing a sensor-cloud iPaaS that can be used to host, operate and integrate sensors. | IoT | No method presented | Yes |
| **[21]** | Ebert et al. (2017) | Combines existing literature on the topic of iPaaS and presents statements regarding the architectural types of iPaaS and the future challenges that are present in the research domain of iPaaS. | NA | Literature review | No |
| **[85]** | Theilig (2018) | Investigating the concept of an open iPaaS by designing and validating a list of requirements and developing a target architecture of the underlying open-source project. | SMEs | Design Science Research | No |
| **[83]** | Srimathi (2019) | They apply an iPaaS to integrate heterogeneous school systems and present some visual integration architecture models. | Education | Case study | Yes |
| **[14]** | Cestari et al. (2020) | Designing an open, generic and extensible iPaaS platform that can be used within the agriculture sector. | Agriculture | Literature review, Case study | Yes |
| **[93]** | Zhang and Yue (2020) | Comparing ESB and iPaaS from a product bundling expertise. They investigate the optimal pricing and bundling decisions and from that, derive the best integration choice. | NA | Design Science Research | No |
| **[42]** | Hyrynsalmi et al. (2021) | Conducting interviews with software companies to understand how integration problems are solved using integration platforms (both on-premise and iPaaS) | Software | Semi-structured interview and thematical analysis | No |

| | | | | | |
|---|---|---|---|---|---|
| **[62]** | Neifer (2021) | Focussing on deriving integration practices and design-related barriers and drivers regarding the adoption of IPaaS. They conducted 10 interviews with experts from different Software-as-a-Services vendors. They conclude that the main factors regarding the adoption of IPaaS are the standardization of data models, the usability and variety of connectors provided, and the issues regarding data privacy, security, and transparency. | SaaS vendors | Literature review and semi-structured interviews | No |
| **[27]** | Frantz et al. (2021) | Presenting an open source iPaaS platform named The Guarana which is demonstrated in a use case | Software | Literature review, case study | No |
| **[41]** | Hyrynsalmi (2022) | Arguing that the definition of an Enterprise Integration Platform as s Service (EiPaaS) is outdated. Therefore, the research aims to formulate a new definition and also highlight several capabilities associated with EiPaaS | NA | A scoping review of academic work and popular tools | No |
| **[40]** | Hyrynsalmi (2022) | Summarizing the state of the art of iPaaS by conducting a systematic literature review. | NA | Systematic literature review | No |
| **[80]** | Sänger and Abeck (2022) | Developing a software solution to integrate authentication and authorization among different systems. The Mulesoft iPaaS is used to develop business logic. | Software | No method presented | No |
| **[43]** | Hyrynsalmi (2023) | Studying the experiences of the professionals, who have gone through an integration platform adoption project in their company recently. | Cross industry | Semi-structured interviews with IT managers and professionals | No |

# Appendix C Synthesis of Functional Capability Models

| Functional capability | Capability description | Ref |
|---|---|---|
| **Connectivity** | The ability to support the physical transport of data and messages between distributed applications through middleware by using synchronous and asynchronous communication mechanisms. Resulting in the ability to tightly or loosely coupled applications | [32] [60] [70] |
| **Protocol bridging** | The ability to seamlessly connect applications that use different communication protocols. | [60] |
| **Message/event Broker** | The ability to operate as a message/event broker to enable topic-based publish-subscribe and queue-based point-to-point distribution patterns. | [32] [60] [70] |
| **Routing** | The ability to direct messages to different destinations, even via different connectivity mechanisms. To support the specification of routing rules, the EAI system must have functionality for the execution of procedural logic, accessibility to the content of individual messages and dynamic routing of outgoing messages based on their content. | [60] [70] |
| **Persistence** | The ability to reliably store messages until they are successfully delivered to their intended destination. | [60] |
| **Transformation** | The ability to transform a message format from a source application into a format of the destination(s) application. | [60] [70] |
| **Message identification and validation** | The ability to identify an incoming message through the metadata of that message. Upon identification, the integration solution should be able to validate the message against a pre-defined message definition to ensure that applications receive correct messages. | [32] [70] |
| **Business Rule Management** | The ability to support the management of business rules which can be used to automate decisions | [32] [60] |
| **Process Management/Orchestration** | The ability to orchestrate the flow of information between applications, services and data sources that includes a sequence of transformations as specified by a pre-defined process model. Each transformation expects a source message to be available, and so the process model coordinates every inter-resource message and request within a complete integration scenario. | [32] [60] [70] |
| **Bulk Data Movement** | The ability to move bulk data between two data sources. Examples of this type of integration are ETL and ELT | [60] |
| **Activity Monitoring and Logging** | The ability to monitor the availability and performance of the integrations and to collect information about these integrations. This information can be used to detect errors and to provide insights into the currently deployed integrations | [60] [70] |
| **Partner Management/media support** | The ability to integrate with B2B-specific data formats and protocols, including the interpretation and creation of message formats and flows applicable to electronic data interchange (EDI). It covers the common administrative transactions of inter-enterprise data sharing and EDI process management | [32] [60] |
| **Prepackaged integration content** | The ability to deliver prebuilt integration solutions, including application and technology adapters, cloud streams, integration templates, patterns and other predefined (including industry-specific) integration content that users can configure, customize, extend and deploy to address common integration requirements for packaged applications | [32] [60] [70] |
| **File Transfer** | The ability to be used as a file transfer solution between file systems, data stores and document stores on a triggered, scheduled or manual basis. Also, the ability to provide pre- and post-transfer processing functions, and to provide operational reporting on file transfers. | [32] |
| **Authentication and authorisation** | The ability to support data and application security through access provisioning, authentication and authorization | [32] [70] |
| **API management** | The ability to define, manage and enforce policies like metering, data masking, traffic shaping and throttling. This includes the provision of developer portals to enable API discovery and manage access requests. | [32] |

# Appendix D Synthesis of EAI and Cloud Factors

**Synthesis of Factors:**

| Non-functional capability | Description | Ref |
|---|---|---|
| Commitment by management | The criticality in the implementation of EAI studies is interlinked with the extended commitment by the management | [37][2][46][92][73] |
| internal pressure | Employees have been working with a traditional manual system for ages and they refuse to move to new systems, which makes the implementation difficult | [53][2] [4][37] [36][46] [92] |
| Centralisation | Centralisation refers to the decision-making authority and encompasses participation in the decision-making and authority hierarchy | [2] [4][46] |
| ROI | measures the profitability of an investment, calculated as the net profit from the investment divided by its cost | [2] [4][46] |
| Costs | The implementation of new technology depends on its cost. For EAI implementation considering the cost is indispensable for procuring and developing adequate levels of hardware and software and training end-users as needed | [53][2] [4] [36][86] |
| Managerial capability | The ability to identify problems of the current systems, and to develop and evaluate alternatives to improve the IT capacity of the organisation appears to be a decisive factor influencing EAI adoption. | [2] [4] [46] |
| Size | The advantages of scaling flexibility and pay-as-you-go cost structure may make cloud services are attractive to SMEs, while large firms may be reluctant to adopt such services because it entails an effective loss of self-control over their IT/IS infrastructure. | [2][37] [73] |
| Personnel IT knowledge | The available skill set of the personnel is an important factor that constrains the introduction of EAI technologies | [2] [46][79] [92] [73] |
| IT Infrastructure | This relates to the heterogeneity of the platforms and the networking capabilities of the organizations that are planning for EAI implementation | [53][2] [4][37] [46] [79][86] [92] |
| IT sophistication | Organizations with sophisticated IT resources are likely to be implementers of integration technology because they are ready to implement new EAI technologies | [53][2] [4] [86] |
| Technological risks | The risk and uncertainty associated with EAI can make risk-averse managers require higher, not lower, rates of return before they invest | [2] [4][46] |
| security and privacy | The security and privacy of citizens' data have always been important. Trust and confidence between users and the government are recognized as critical success factors in the implementation of EAI | [2] [4][37] [36][46] [79] [73] |
| Compatibility | Once the technology is compatible with existing systems and working norms, the organization will tend to adopt it. When a hospital integrates a new information system that is more compatible with it. | [37] [36] [79] [73] |
| Simplicity/ease of use | organizational users need to find out how simple the solution is and judge this new technology's potential benefits | [79] [36][92] [73] |
| business partner satisfaction | Business partner satisfaction has a significant impact on the performance of enterprises. The accomplishment of this factor will not be attained without integrating services needed by business partners and EAI technology usage | [53][2] [4][46] |
| Market knowledge | The majority of successful IT implementation cases refer to the recognition of demands in the market which supports stakeholders with the knowledge needed to be updated about what is the trend in the market regarding EAI technologies | [2] [46] |
| External support | The system implementation can never be successful without external support and it is also an important factor. EAI distributors have to arrange this support implicitly (consultancy, maintenance and updates) | [53][2][37] [46][86] [73] |
| External Pressure | Companies face external pressures in adopting innovative technologies in the form of governments, stakeholders and competitors. Competition increases environmental uncertainty and the demands for and the adoption of innovative technology. Increased | [53][4][37] [46] [79] [92] [36] [73] |

| | | |
|---|---|---|
| | external pressure results in circumstances where a higher level of information system integration is necessary to respond to various external pressures. | |
| Regulatory environment | We defined the regulatory environment as the degree to which the cloud service is affected by government or company rules. Government policy and related legislation play a critical role in affecting the development and diffusion of innovation | [36][73] |

## Mapping from factors to ISO capabilities

| **Factors influencing EAI adoption** | **Description** | **Mapping to ISO 25010 component(s)** |
|---|---|---|
| Internal pressure | Internal pressure is another important aspect of the influence of IT adoption. Employees have been working with a traditional manual system for ages and they refuse to move to new systems, which makes the implementation difficult | Learnability, operability |
| Centralisation | EAI projects will increase the efficiency and effectiveness of companies and save money through increased centralisation of resources. Centralisation refers to the decision-making authority and encompasses participation in the decision-making and authority hierarchy | Resource utilization (being the main driver for higher, managerial personnel) |
| ROI | The management layer of companies is often only interested in the ROI of the EAI project R | Resource utilisation |
| Costs | The implementation of new technology depends on its cost. For EAI implementation considering the cost is indispensable for procuring and developing adequate levels of hardware and software and training end-users as needed | Resource utilisation |
| Size | The advantages of scaling flexibility and pay-as-you-go cost structure may make cloud services are attractive to SMEs, while large firms may be reluctant to adopt such services because it entails an effective loss of self-control over their IT/IS infrastructure. | Scalability |
| Personnel IT knowledge | The available skill set of the personnel is an important factor that constrains the introduction of EAI technologies | Accessibility, learnability, operability |
| IT Infrastructure | This relates to the heterogeneity of the platforms and the networking capabilities of the organizations that are planning for EAI implementation | Compatibility |
| IT sophistication | Organizations with sophisticated IT resources are likely to be implementers of integration technology because they are ready to implement new EAI technologies | Compatibility, |
| Technological risks | The risk and uncertainty associated with EAI can make risk-averse managers require higher, not lower, rates of return before they invest | Recoverability, security, availability |
| security and privacy | The security and privacy of data have always been important. Trust and confidence within the company but also between its business partners are recognized as critical success factors in the implementation of EAI | Security |
| Compatibility | Past studies have found that, when innovative technology is incompatible with users' existing values and jobs in an organization, obstacles to its adoption will arise. In other words, once the technology is compatible with existing systems and working norms, the organization will tend to adopt it. When an organization integrates a new information system that is more compatible with it, the cost and difficulty level of the integration will be relatively lower. In other words, when the information system can be integrated relatively easily, there will be an increased willingness to seek it | Compatibility, |
| Simplicity/ease of use | Organizational users need to find out how simple the solution is and judge this new technology's potential benefits | Accessibility |

| | | |
|---|---|---|
| business partner satisfaction | Business partner satisfaction has a significant impact on the performance of enterprises. The accomplishment of this factor will not be attained without integrating services needed by business partners and EAI technology usage | Performance, availability, security |
| External support | The system implementation can never be successful without external support and it is also an important factor. EAI distributors have to arrange this support implicitly (consultancy, maintenance and updates) | Learnability |
| Regulatory environment | We defined the regulatory environment as the degree to which the cloud service is affected by government or company rules. Government policy and related legislation play a critical role in affecting the development and diffusion of innovation | Compatibility |

**The first selection of Non-Functional Capabilities**

| Non-functional capability | Description |
|---|---|
| **Resource utilisation** | The degree to which the amounts and types of resources used by an integration platform, when performing its functions, meet requirements. |
| **Availability** | The degree to which the integration platform is operational and accessible when required for use. |
| **Security** | The degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization. |
| **Scalability** | The ability for the integration platform to scale to meet increasing demands; for example, at peak times or as the system becomes more widely adopted. |
| **Performance** | The ability of the integration platform to always run acceptably. In mission-critical systems, even the smallest delay in processing data can be unacceptable. |
| **Compatibility** | The degree to which an integration platform can exchange information with other products, systems or components, and/or perform its required functions while sharing the same hardware or software environment |
| **Operability** | The degree to which the integration platform has attributes that make it easy to operate and control. |
| **Maintainability** | The degree of effectiveness and efficiency with which the integration platform can be modified by the intended maintainers |
| **Learnability** | The degree to which an integration platform can be used by specified users to achieve specified goals of learning to use the integration platform with effectiveness, efficiency, freedom from risk and satisfaction in a specified context of use. |

| | |
|---|---|
| **Accessibility** | The degree to which the integration platform can be used by people with the widest range of characteristics and capabilities to achieve a specified goal in a specified context of use. |
| **Recoverability** | The degree to which, in the event of an interruption or a failure, the integration platform can recover the data directly affected and re-establish the desired state of the system. |

## Appendix E First Expert Review Transcription

| | |
|---|---|
| | |
| | Permission to record is asked before the starts |
| | |
| | Q: Could you introduce yourself |
| 1 | I started another company 27 years ago that included custom applications. It also included CRM, which was sold to Deloitte in 1998. Then we went along because Deloitte took over the company and its IT people. At the time when we were part of Deloitte, we still did some customization for a while. In the end, mainly customer applications between CRM. I have always been involved in CRM and integrations. Siebel was acquired by Oracle so then Oracle CRM and Oracle integrations were done. That became less in the CRM part because Oracle CRM wasn't selling well and salesforce came up. So then I did a lot of integration between different Oracle cloud products. I missed the CRM part a bit because that's where the innovation is. So I switched to digital customer a few years ago. I was asked to build an integration team for a year. Now I am the business lead of this team. A year ago we made a big acquisition, so now we have about 40 men. We are getting broader as a team. Because we have such broad expertise, we are also involved with the customer at an early stage because we can also advise them in their choice of a particular vendor. In this, we overlap with the enterprise architecture team. We also regularly receive requests from customers to help them with this |
| 2 | I have been working in the integration world since 2017. I started as an intern at another company that switched to Deloitte and now we are part of Deloitte Digital. During my time I worked as a system administrator, and then I ended up as a platform engineer. That's when I was asked to become a platform engineer at Company X. I have a wide range of expertise considering running everything on-prem to everything in the cloud and everything in between. In my time as a system administrator, I was still busy setting up servers and patching switches. |
| 3 | I rolled into integration 20 years ago, at one of my previous employers felt the need to connect financial processes (SAP and invoice scanners) and to build business logic into them. I started there in operations, supporting developed applications and those were all on-premise environments in Tibco (2000s). That was technical and functional management. I did that at two companies, from 2003 to 2010 and from 2010 to 2017 at a bank. That was all on-premise Tibco middleware integration platform. At the time I left, the migration from on-premise to the cloud was underway, so I didn't experience that anymore. The speciality of the company was integration software (Tibco, Mulesoft and other platforms) Mulesoft is a modern integration platform where you can have a mix of on-prem, iPaaS or both. That's where I started with Mulesoft development 6 years ago. My current client at Deloitte is company X, there I do Mulesoft development all as iPaaS. Development consists of analysis, design and building integrations. |
| 4 | I have been working in IT for a while now, for over 30 years. And in the last 20 years, I have been working quite intensively on the subject of integration and came into it a bit by chance. I worked at Microsoft and there I specialized in BizTalk, Microsoft's on-premise integration platform. I worked with me for 7 years. Yes, after that I started my own business. I think I find it especially interesting to work on the content of the integration. But I have had various roles: at company X I worked as an architect of the integration team for a while. Now I'm with two customers: Company Y, which is a cheese supplier and Company Z, a slaughterhouse and food services. That is a slaughterhouse and also a meat processing company. |
| 5 | I'm studying computer science and started at a software company that Deloitte bought around 2000. At Deloitte, I have always been involved in software development with Oracle and Microsoft. After that, more with integration. |
| 7 | I've been working for Deloitte since 2017 in the Oracle offering, so I do everything but in the context of Oracle-related projects. We do large ERP implementations based on our offerings. Oracle currently has the latest greats of product portfolio being offered in the cloud as a SaaS solution. For ERP, it is |

| | |
|---|---|
| | interesting for a lot of companies to go from an on-prem to a cloud ERP package. We support large projects of cloud transitions. This can be all kinds of different flavours. Sometimes for a customer in the Netherlands, sometimes for large multinationals with a large number of countries. The scope of that context also determines how complex, large and long it takes. ERP is at the heart of business operations, which means that it often requires a multitude of integrations with peripheral applications. Transactions have to come from the outside in or vice versa. To facilitate this as much as possible, Oracle has its integration product: Oracle Integration Cloud. That is the product that is usually used as a middleware product to facilitate the middleware. |
| 8 | I've been working at eMagiz for 13 years. First started as an application and integration consultant. From there, we built up expertise in integrations. eMagiz is a spin-off of Cape Group. Because there was so much demand within the cape group for an integration platform, we developed it separately. When it got bigger, I took the step to become a product manager there. I then managed the development team for several years. Had multiple roles at Emagiz since 2018. However, always in the integration branch. When the company was just starting, we were already dealing with integration tooling. Then we used open source. What you're saying, it used to be middleware and since 2015, 2016 cloud computing has been called iPaaS. You have parties such as Gartner and Forrester that throw a new name at it once in a while. Then they come up with a new name every 5 years and then they start interviewing parties and they have to meet their standards (that's their marketing model). iPaaS is a term, but a hybrid integration platform was used for it. We have continued the ideal image of a hybrid integration platform. We used to be on-premise, but when the cloud became hip, we fully focused on it. In that capacity, our software has been transformed in such a way that it can run in the cloud and on-premise. In the cloud, it's fully automated and you don't have to do certain installation steps. On-premise, you have to ask the customer for a server (windows Linux). Someone on-site to do the initial setup. In the cloud, you press one button and you have something. |
| 9 | I have been working in integration for a little over 10 years. Currently working for Deloitte for almost 2 years. In this capacity, I have often advised on IP selection processes for large organizations in various industries. I have developed several methodologies for this type of advisory process. |
| 10 | I write a functional design about what an integration should do. I also write a technical design in which there are specifications without details. E.g. endpoints that are used. I can also think along to a certain extent in solutions for, for example, a message attachment needs to be used. I don't program in middleware. |
| Int | Q: In the literature, it seems that the decision for companies to choose between an iPaaS platform and an on-premise platform to develop and manage its integrations is unclear. What is your view on this matter? |
| 1 | Looks like they have to. First of all, because the latest versions are all cloud products or hybrid iPaaS. Secondly, it fits in the movement that companies already have towards the cloud for their other applications (SaaS) Also, for greenfield customers, the investment in a traditional integration platform is too big, the vendor knowledge offered for iPaaS is better, and you get Kubernetes hosting, and you do not need maintenance and service teams to manage your network so you are partially unburdened in the cloud. Compared to traditional IP, you see a step up based on the business requirements to which iPaaS responds better: Scalability, API-led connectivity, low-code orchestration tools (of which I am not a particular fan), B2B integration, security and flexibility The reasons for an on-premise platform almost always has to do with regulations. A couple of years ago, speed was an aspect but now you can host a private cloud with servers located next to each other so the latency is reduced. |
| 2 | Yes, you see companies struggling with the decision between running a traditional integration platform or going to an iPaaS or SaaS platform. Running a traditional integration platform on the production level requires a lot of people which is difficult, especially for smaller companies. Those smaller companies tend to move towards an iPaaS or SaaS platform because it unburdens them. However, for iPaaS, you will still need the expertise to set up and manage the platform which might be difficult to find in the market. Then SaaS is a good option because it unburdens completely. At company X, on-premise never goes away as a result of sensitive data that cannot leave the company's premises (also not private cloud). However, the data that is not sensitive is integrated through iPaaS so we use a hybrid approach |

| | |
|---|---|
| 3 | When I get involved, the choice has already fallen for the Mulesoft environment. What I do read is that it often comes down to the following considerations: Do we want to limit the capital expenditure or the operational expenditure? Another aspect is the security of data (GDPR rules) |
| 4 | Companies indeed have to make a choice, but within that choice, different motives contradict each other, and I think that that makes it very complicated. There are many pros and cons for applying an iPaaS making it a difficult trade-off. If you already have a traditional integration platform, investments are already made in that platform, you are used to developing integration with that platform, and integrations are currently running using that platform so it is convenient to keep using it. However, the vendor is pushing towards the iPaaS platform by offering the latest and greatest there. That is however also a point of caution, the iPaaS concept is new and under development so it might be worthwhile to postpone migration projects till the platform is more mature. In the end, cloud iPaaS is the future but it is a question of how and when the integrations are migrated to an iPaaS. |
| 5 | First, the question is whether you want to use a vendor's integration platform at all. That depends on the amount of integrations how technically savaged your team is and what that team prefers. If we then assume that we are considering scenarios in which the IP will be used, then that choice is based on all kinds of arguments. First of all, what does your application landscape look like, is it 90% on-prem<br><br>and 10 in the cloud. Or are there a few large ERP packages in the cloud and the rest on-prem<br>. And what is your expectation for the future when that will change? I think that mainly determines whether you make that choice to do integration in the cloud at some point and when you make that choice. It will never be the case that companies first opt for IP in the cloud and only then start running applications in the cloud, IP follows the rest. The choice for an IP is a choice that often goes along with one of the major applications (CRM, ERP). If it goes to the cloud, they often start thinking about iPaaS. Why is that, because a lot of applications within the landscape use that ERP package, so maybe 50% of your applications have to be changed the moment your ERP package goes to the cloud. |
| 6 | Absolutely, Companies struggle. First of all, in deciding do they even need an integration platform. However, building your integrations point-to-point without IP is not future-proof. and companies will end up spending a lot more time on development and maintenance. However, using an ip costs money so it is a matter of time versus money. Once they are clear that they need an integration platform, the second question is: do they need it on-premise? which is like an enterprise service bus or do they need a properly cloud-hosted iPaaS? That is the second thing that they struggle with. And third thing which I do see a lot, the people struggling a lot these days, is which iPaaS is going to be best suited for them to like know have in their IT architecture. So these three things I think I've seen a lot and the answer to that is it is going to depend on what their needs are, what their current landscape architecture is, what is their future roadmap, what they see in their next fires down the line, how do they see their IT infrastructure. |
| 7 | Depends very much on the situation in the IT landscape that you find at the customer. If you start a cloud transition in a complex IT landscape that might have an integration platform running. Then you have to make the trade-off, are we going to use the existing platform, maybe use both integration platforms (because there are new and existing use cases that need to be maintained) or just migrate entirely to the cloud? It has to do with vision, Is there an IT strategy and IT vision within the company on how a company should deal with integrations and where we want to go on our roadmap to the future? So it has to do with the maturity of implementing a strategy within a company (vision of the future or letting it be ruled by the issues of the day, "just put it to work" and everything in between. That makes what kind of choice is made. It's always about a business case. |
| 8 | In principle, no. Our customers are not concerned with the cloud at all, on-premise, they just have integration issues. In addition, a lot of our customers don't have an existing IP at all when they knock on our door so that demand is not there either |
| 9 | I think the question of on-prem or iPaaS is outdated. There are two scenarios, 1 is greenfield, they have no IP and many p2p integrations and see that they are now stuck. The other is the legacy modernization of their integration capabilities. For example, they have an on-premise ESB installed. But they want to explore |

| | |
|---|---|
| | innovation in that area. With Greenfield, the starting point is immediately an iPaaS or iSaaS, and the choice is between these different forms. In the second scenario, they want to upgrade their integration capabilities, but on-premise is usually still included because they already have an on-premise installation and then they want another on-premise deployment model. The iPaaS is the concept of a distributed IP that can be on-premise or in Azure. So the question between iPaaS and on-premise mixes up certain attributes of such an IP.<br>Greenfield is often already in the cloud because they have new IT infrastructure. If you have a larger company that has been around for a long time, they have an on-premise presence. But they want a combination of both within the same IP (same vendor). |
| 10 | Company X used a traditional integration platform before, as did a large insurer. So it's still going to happen. But they do look for solutions that make it a bit cheaper. It must be said that the SOA suite can do more than OIC. There's more functionality in there than OIC. You can also get SOA suite in the cloud from Oracle on an Oracle server. |
| Int | Q: your experience, if and how do companies struggle with the decision between different integration vendors? |
| 1 | yes, that decision is very relevant for our customers. Important for that decision is that the requirements from the company side are clear. It helps if the company has an experienced IT department because they usually understand the requirements. Otherwise, we sit together with the company to guide them in the process of defining the requirements but that takes more time.<br><br>In terms of differences between platforms. Azure is all-encompassing but difficult for companies that do not have the resources to deal with Azure (because it is such a wide platform). If you are not tied to SAP or Oracle you will probably not use any of those integration platforms. Then often you will go to Boomi or Workato |
| 2 | These are choices that are made at the management level.<br><br>If you look at those different cloud platforms, there are many reasons to choose or not to choose a particular platform. You often see that companies have a high priority on Azure. Especially because Azure is easy to integrate with the customer's current environment because part of the server park is already running on Windows. |
| 3 | Yes, I regularly see these questions in Deloitte team groups. I don't know if it's difficult, but companies do need to determine what are important factors to them. |
| 4 | I profile myself as a Microsoft Integration Specialist. So the choice has already been made when I get to the customer. But there is a choice. Note that you often choose a technology of something that is already running within your company. Gives Eneco an example And so they had different integration platforms. That resulted in a whole battle every time where something had to be implemented. I think that the functional and non-functional requirements can help a lot to provide insight into the choice you are making. So I think that could have helped companies to make that decision better in advance so that they might be faced with fewer surprises afterwards |
| 5 | If they make a choice themselves, it is almost always the choice of the IT department and not a business choice. That's because integration is a technical thing, so the business doesn't think about integration at all and assumes that it's taken care of and so it ends up with IT. If you have an IT department that programs the most in Java, you will look for the solution that best suits you. If they are very involved with Salesforce, they will probably go to Mulesoft. |
| 6 | And third thing which I do see a lot, the people struggling a lot these days, is which iPaaS is going to be best suited for them to like know have in their IT architecture. |
| 8 | Not always, you have two types of businesses: |

| | |
|---|---|
| | - Companies that are looking for an IP<br>- Companies that do not realize they need an IP (the smaller companies that have become bigger and normally never had to deal with this)<br><br>For the first type of company, you come in a selection process where they indeed use those requirements lists to compare the vendors. In the latter type, they are often made aware during a project that an IP might be helpful for them<br><br>A problem we experience is that if a customer already has an IP, they don't just say goodbye to it. There are years of investment in this. Companies will only replace their IP if they are very dissatisfied, or if there is a large party that can afford to have a second IP next to it. The reason for that is that when you switch from platform, you have to redevelop all the integration interfaces which results in a complex migration process + you have to educate people again in using the new platform |
| Int | Q: Would you agree that the list of capabilities is complete and that it can be used to assess an integration platform? |
| 1 | Starting with functional: I understand all the functional capabilities and think that the list is complete<br><br>Non-functional: I miss one thing: Your integration strategy must be linked to your business strategy. This must be reflected in what a company wants to distinguish itself. For example, you can start a new company X, but you won't distinguish yourself with a good portal. So you can buy a portal from the customer, buy a good ERP system, and use standard connectors but that is not differentiated so you don't have to put a super advanced integration platform in between. Or, as a company, you want a customized portal, which means that very specific integrations will be possible, which will make many more demands on your IP. So the question is: Are you dealing with very specific integrations because of your business strategy? What is the impact of your business strategy on the integration strategy? From there come the requirements for your IP. The rest of the NFRs I can place well and sound logical, |
| 2 | Yes I think that both lists are pretty complete |
| 3 | I thought that both were good lists. There is one thing I've written about it:- Development tools: What kind of tools are available and how well do they support the development process you have in mind? |
| 4 | Functional capabilities are complete<br>Non-functional requirements: I would like to add maybe 1 more thing in the choice of an IP, a weird one but does have an influence. What you see is that e.g. Microsoft, which just says Azure is the future. This means that the whole elite corps of integration people is focused on the latest technology. So apart from the fact that as a company you have to choose whether you want to go in that direction. You see that the people who are going to help you build the integrations are already chosen for you. If I'm speaking to myself, I'm not going to do a BizTalk implementation anymore. I have to work on my profile and my profile is not served by the fact that I am going to gain experience with outdated technology |
| 7 | Yes, both lists look complete. I don't think of anything missing |
| 8 | NFR: Non-functional requirements: These are all things that companies expect you to have from a certain limit. (hygiene factors). It has to be able to do this, it has to be able to handle so much, it can cost so much, etc.<br><br>FRs: I don't see anything crazy in the list and I only miss it: Often you need parties for IP that can manage it for you in case you don't have the expertise yourself. I don't see that one again. What professional services are provided? We have sometimes had problems with parties wanting us, but because we are smaller, those |

| | |
|---|---|
| | customers were afraid that we could not provide enough support capacity. In addition, larger companies often do not want to work with smaller companies because they are afraid that those smaller companies will not be able to handle it. |
| 9 | A lot of classic ESB capabilities. I would develop B2B/partner management further into several sub-capabilities because it's an important aspect of an iPaaS. The same applies to API management and maybe even File transfer. Although the latter is used less in reality. I also miss data management capability here, it's not a core capability of integration but it can have a lot of effect on how you set up your integrations. Below I would include data virtualization, master data management, metadata management, and business rule engine. |
| Int | Q: What kind of integration project(s) have you worked on? |
| 1 | I am now with a client where we started 3.5 years ago. They had started with a digital factory and requested an integration platform because they didn't have one there at all. They had a very small IT department which they gradually expanded and which had to gradually start working in the IT landscape. A central organisation is now being set up for this. So we started in the greenfield with an IP, which is nice, there are the typical challenges with on the one hand the latest of the latest and on the other hand the old SAP ERP system from 1990 that has been completely customized. In this way, we come across all kinds of things that need to be tied together. Those are beautiful puzzles. |
| 2 | ASML is a broad company with a lot of data flows, so they had the request for an IP that can operate both on-prem and in the cloud. That's how they ended up with Mulesoft. What we have seen at ASML is that on-prem is difficult to set up, but you have the most influence at times when you need to expand things, you have more solution options. ASML uses a hybrid Mulesoft approach because they have to deal with highly classified data that is not allowed to be sent over the cloud. They do integrate non-classified data through a cloud-based iPaaS.<br><br>One of the difficult things in deciding on a platform is to predict how your integration landscape will look in 3, 5, or 10 years from now. That requires a lot of knowledge and predictive ability.<br><br>if you go for an on-prem platform then you need a lot of resources, you have a team from the network, a team from the server installation, a team that manages software and only then do we come and set up the IP. That makes the TCO very large, where you can remove a number of those layers (teams) with an iPaaS platform. You don't need a networking team, for example. That's all taken care of for you. That does not alter the fact that you do need to have some of that knowledge. ON an iPaaS platform you still have to spin up servers, you just don't have to physically go there. The dividing line between on-prem and iPaaS is there, but it is not very large (he is talking about the dividing line here in terms of necessary knowledge, complexity, etc.). However, the difference between those 2 and SaaS is very big. All you have to do is press deploy and everything will be taken care of for you. If you go to an SME, the choice to spend a little more on a license for SaaS is often smarter than getting a lot of knowledge to set up an iPaaS platform or on-premise platform |
| 3 | It is often about unlocking data from back-end systems and connecting different systems, at different locations, with different protocols, from different vendors. They say: We have a legacy system here, which produces a lot of data about aircraft movements. We want that to go to a database, but not via a p2p script but with a good integration that also facilitates reusability of the data. Then an application rolls into Tibco, Mulesoft that reads, stores, transforms and sends data.<br><br>Another company is a government organization. They all have measurement systems in place in the country that send data to the Directorate-General for Public Works and Water Management. They want to make that available as a service to customers. This is done via APIs that you define at the front end with well-defined resources. That request is routed to the right backend systems, which send the data and combine it into a response. You can think of it as bringing differences between data from systems A to B. For example, synchronizing product data or orders. Another use case is unlocking data to the clients. Those clients |

| | |
|---|---|
| | can be systems within the company (an old legacy mainframe computer, you can build an API in your iPaaS that unlocks the data from that computer) That's the main use case for mule soft |
| 4 | Gasuni had an existing IP so there was not an immediate need to do something else, apart from a healthy interest in what else was happening in the market. At the time, you also saw a lot of immaturity in the Azure platform. It was unclear how the solutions would land there. Some pilot business was done and nothing would be taken away if it had a specific advantage (was easier) to use the iPaaS in the iPaaS. So there was an "on-premise first unless... " mindset. In practice, this meant that several very small projects went to Azure, but those were not even real integrations. There was a pre-emption on the fact that the applications were going to the cloud. That's something many companies embrace: Buy before build before reuse. When you see that all the major software players are betting on the cloud. Then you know that a large part of your IT landscape will run in the cloud. So it's also useful to have technology that can integrate with the cloud.<br><br>With Vergeer, it was very different. They had an on-premise integration platform, but they had it implemented by an external party with whom they no longer had any contact. As a result, their IP was unsustainable. They had it rebuilt by a partner who wanted to do the maintenance but with the latest technology (iPaaS). Then you see that if they had gone through all the capabilities at the time, they would have been more aware of that choice. They also have on-premise software development, but on-prem doesn't match what's happening in Azure. They don't have any cloud expertise themselves, but their IP is now running in the cloud. As a result, everything is now a black box for them in terms of integration. They don't understand Azure and are used to doing on-premise integrations. They also do not transition other IT systems to the cloud. This situation puts pressure on the choice of a cloud integration platform. This transition from on-prem to cloud was easy because there were no other changes in that transition period that allowed them to run the cloud platform as a shadow platform with the existing platform |
| 5 | Consultancy projects about an IP and internal projects for an IP implementation, where we also build integrations. External projects for customers in the healthcare sector where we usually make integrations in Mulesoft. Also built in the Oracle SOA suite. I'm not the one who builds the integrations, but I manage a team that does that. I am functionally responsible for what needs to be built. In the future, this may be done more offshore, where we will work together with Indian colleagues. |
| 6 | So I've worked on a project called Company X. Company X is a competitor of Walmart in the US and I think they chose Mule Soft because it gives a lot of out-of-the-box connectors and they had integrations with Oracle, ERP, Oracle Finance and retail. And I think they saw that mule soft would be the best fit for them. They did not want to invest so much in infrastructure. They did not want to take the pain of handling setting up a server and VMS and all of those things so Mulesoft as cloud is like a pure cloud platform which provides out-of-the-box features for our deployment, hosting infrastructure and everything. So I think that was one of the primary reasons that they went with uh with an iPass mule soft. Apart from that, it gives a lot of flexibility in terms of reusability, high availability, and 0 downtime. All those features I think attracted them to use ipass. I thought they had because they are a retail company, so company X or Walmart, have to talk to a lot of third-party vendors for their wholesale purchases and all of those things, right? So when they want to talk to a lot of third parties where they need us, you know secured layer in between while talking to or while communicating with outside vendors for their information or data or products or and whatnot, they need to have a gateway kind of in between when are you know talking to the outside world that's where an ipass or Mulesoft was helpful to give that extra layer of security to them I would say. |
| 8 | Yes, what I already mentioned is that you have those two types of customers. 1 customer knows he needs an IP. Then you get to the checklists where you can look at what you can do as an IP. This plays a role in, for example, a tender. Those functions that you showed are part of that. The majority (customer 2) don't know that they need an IP. They want to have a problem solved and are much less interested in those functions. They work with problems and that is translated into functional things. All those non-functionals are hygiene factors, if you don't meet them, you won't even get in. The small companies are not asking for this, but the large companies certainly are. Of course, that is of no use to you for your answer. It's just very different from party to party. The big clubs have seen on the |

| | |
|---|---|
| | internet what an IP has to meet and then they ask the vendors for that list. We work a lot with customers in logistics and the question is always about: have you already done this or can you already do this, and then the question arises: how much does it cost? |
| 10 | We mainly have customers with very outdated technology. And then we do a whole implementation. We recently had a client who did everything in Excel. We then roll out Oracle HCM and then Oracle OIC for the integrations. They all have other adjacent systems that have to be integrated via that middleware platform so that all integrations run on 1 platform. If it is spread over all kinds of wood string solutions, it becomes very confusing. So then you get a migration project from those wood string solutions to the Oracle OCI |

# Appendix F Capabilities Descriptions of Capability Model

| Functional Capabilities | Description | Ref |
|---|---|---|
| **Synchronous/asynchronous communication** | The ability to support the physical transport of data and messages between distributed applications through middleware by using synchronous and asynchronous communication mechanisms. Resulting in the ability to tightly or loosely coupled applications | [32] [60][70] |
| **Transformation** | The ability to transform a message format from a source application into a format of the destination(s) application. | [60] [70] |
| **Routing** | The ability to direct messages to different destinations, even via different connectivity mechanisms. To support the specification of routing rules, the EAI system must have functionality for the execution of procedural logic, accessibility to the content of individual messages and dynamic routing of outgoing messages based on their content. | [60] [70] |
| **Integration flow development tool** | The ability to offer (low-code/no-code) tools that support the development of integration flows. | (3) |
| **Queues and topics** | The ability to operate as a message/event broker to enable topic-based publish-subscribe and queue-based point-to-point distribution patterns. | [32] [70] |
| **Publish and subscribe** | The ability to operate as a message/event broker to enable topic-based publish-subscribe and queue-based point-to-point distribution patterns. | [32] [70] |
| **Activity Monitoring and Logging** | The ability to monitor the availability and performance of the integrations and to collect information about these integrations. This information can be used to detect errors and to provide insights into the currently deployed integrations. | [60] [70] |
| **Orchestration** | The ability to orchestrate the flow of information between applications, services and data sources that includes a sequence of transformations as specified by a pre-defined process model. Each transformation expects a source message to be available, and so the process model coordinates every inter-resource message and request within a complete integration scenario. | [32] [70] |
| **Protocol bridging** | The ability to seamlessly connect applications that use different communication protocols. | (4) |
| **Persistence** | The ability to reliably store messages until they are successfully delivered to their intended destination. | [60] |
| **Authentication and authorisation** | The ability to support data and application security through access provisioning, authentication and authorization. | [32] [60] [70] |
| **Traffic management** | The ability to throttle and manage the API traffic to handle unexpected spikes in traffic, limit the number of calls that can be made to an API and prioritize important classes of customers who should be given priority over other calls | [32] |
| **Orchestration and routing** | The ability to route and combine multiple requests from API consumers to the correct back-end services that provide the requested business functionality. | [32] [60] [70] |
| **Developer portal** | The ability to provide a platform that can be used by developers to discover, use and create APIs. The platform should document existing APIs | [32] |
| **Analytics** | The ability to provide information on the traffic of the API platform that can be used as a base for future decision-making | [32] |
| **API lifecycle management** | The ability to manage the already developed APIs. APIs are likely to be updated with business requirements and this capability ensures that these updates are handled correctly | [32] |
| **Adapters** | The ability to offer pre-defined and ready-to-use B2B adapters to connect partner systems to the integration platform | [32] [60] |
| **Partner management interface** | The ability to facilitate event management and reporting, auditing in centralized dashboards that can enable you to extend real-time activity and status information from and to your partners. | (1, 2, 8) |
| **Encryption** | The ability to encrypt and decrypt messages and files to increase security. This capability also includes certificate management including the support for modern ciphers and algorithms | (2, 8) |
| **Trading partner agreement** | The ability to set up, activate and manage trading partner agreements that define how the trading/business process is fulfilled in the given context | (2) |

| | | |
|---|---|---|
| **Message identification and validation** | The ability to identify an incoming message through the metadata of that message. Upon identification, the integration solution should be able to validate the message against a pre-defined message definition to ensure that applications receive correct messages. | [32] [70] |
| **Meta-data management** | The ability to establish policies, procedures, and tools to ensure that metadata is accurately, consistently, and efficiently collected, integrated, accessed, and maintained across an organization. | (4, 9) |
| **Data synchronization** | The ability to ensure that data across multiple devices or systems is consistent and up-to-date. | (4, 9) |
| **Bulk data movement** | The ability to move bulk data between two data sources. Examples of this type of integration are ETL and ELT. | [60] |
| **Business Rule Management** | The ability to support the management of business rules which can be used to automate decisions. | [32] [60] |
| **Data virtualization** | The ability to retrieve and manipulate data without requiring technical details about the data, such as how it is formatted or where it is physically located. It provides a unified, abstracted, and real-time view of data from multiple, disparate sources, enabling users to access and analyze data without needing to know its underlying structure or location. | (4, 9) |
| **Master data management** | The ability to manage an organization's critical data, focusing on creating a single, consistent, and accurate source of truth for shared master data | (4, 9) |
| **Automation** | The ability to schedule and automate repetitive file transfers to reduce the manual work and manual errors | (4) |
| **Protocol support** | The ability to support end-to-end encryption protocols such as SSL, Secure File Transfer Protocol, and File Transfer Protocol Secure (FTPS) for secure content in transit. | [32] [60] |
| **Non-Functional Capabilities** | **Description** | |
| **Pricing Model** | The sum of money paid to the vendor to make use of the platform's functionalities. The approach to calculating this depends on the specific vendors' licensing plan | (5) |
| **Resource utilisation** | The degree to which the amounts and types of resources used by an integration platform, when performing its functions, meet requirements | See Appendix D |
| **Security** | The degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization. | See Appendix D |
| **Scalability** | The ability for the integration platform to scale to meet increasing demands; for example, at peak times or as the system becomes more widely adopted. | See Appendix D |
| **Performance** | The ability of the integration platform to always run acceptably. That is, to run according to the defined Service Level Agreement. This included the performance in terms of latency, the availability of the platform and in case of malfunctioning, the recoverability. | See Appendix D |
| **Compatibility** | The degree to which an integration platform can exchange information with other products, systems or components, and/or perform its required functions while sharing the same hardware or software environment | See Appendix D |
| **Usability** | The degree to which the integration platform can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of the use | See Appendix D |
| **Maintainability** | The degree of effectiveness and efficiency with which the integration platform can be modified by the intended maintainers | See Appendix D |
| **Availability** | The degree to which the integration platform is operational and accessible when required for use. | See Appendix D |

## Appendix G Slide Information Including References

| Slide Title | TIP part | Ref | iPaaS part | Ref |
|---|---|---|---|---|
| Pricing model (1/2) | Generally uses a licensing-based pricing model | NA | Organizations only have to pay for the combination of services that they require (not buy the entire package of a TIP). | [55] |
| | The pricing model is only a small portion of the TCO -> see resource utilization | NA | The pricing model is complex and can be licensing-based or consumption-based or a combination of both. This depends on the specific vendor and the set of services used. The complexity of the pricing model can make it difficult to predict the pricing of a specific vendor and also to compare are pricing of different vendors. | [61] |
| | | | For flexible scalability, a more consumption-based pricing model is favourable.<br>With more constant consumption, a plan-based pricing model is desirable because it is cheaper and more predictable in terms of performance. | [61] (5) |
| | | | | |
| Pricing model (2/2) | BizTalk[1] licenses per core per year:<br>• 10,853 dollars per core per year<br>• Minimum of 4 core licenses required per server resulting in minimum costs of 43,412 dollars per server per year<br>Oracle[2] licenses per processor:<br>• 57,500 dollars per processor per year | See Foot -note | Consumption-based: iPaaS is described in the literature as low start-up costs because it is a pay-as-you-go model. This holds for example for Azure[3]:<br>o Pay-per-execution: Small amount of money every time a resource (integration flow, API connector, Queue/Topic) is executed.<br>o Pay-per-hour/month pricing: Pay per month for a certain amount of CPU and memory. | See Foot -note |

[1] https://www.biztalk360.com/blog/biztalk-server-2020-licensing-cost/#Session%205

[2] https://www.oracle.com/a/ocom/docs/corporate/pricing/technology-price-list-070617.pdf

[3] https://azure.microsoft.com/en-us/pricing/details/logic-apps/

| | | | | |
|---|---|---|---|---|
| | | | Difficult to estimate the total costs as it highly depends on the number of integrations that are to be developed and the amount of traffic that is expected. | |
| | These are the minimum prices to buy the license for these platforms. Be aware that this is for one server only, availability requirements might force a company to have at least two servers up and running -> doubling the price[4]. Furthermore, often companies have multiple environments (dev, acc, prd) which could increase the required computing power and therefore the price. | [47] | **Plan-based:** Other iPaaS platforms have a considerably large upfront investment: SAP integration suite[5] starts at 60,804 dollars/year, Boomi[6] at 48,000 dollars/year and Mulesoft[7] at 80,000 per year. | See Foot -note |
| | | | | |
| Resource utilisation | All the information is derived from this report[8] | | | |
| | | | | |
| Security | Resource utilization showed that a TIP requires a stack of hardware and software for which the organization is responsible. This responsibility includes the security configurations of all the components in the stack. | [73] | Because the platform is automatically updated and the iPaaS is the latest-greatest of vendors (described in maintainability) users have access to the vendor's best security features and protocols. Be aware that this does not mean that iPaaS is always better than TIP as TIP platforms can also have these protocols and features. | [7] |
| | Hence, building and maintaining (patching and updating) a solid security infrastructure requires a lot of resources (both personnel and money). Even then, it is debatable if the security | See Foot -note | Vendors use a shared responsibility approach that outlines which components of security fall under the vendor's responsibility and which under the company's. Hence, the company does not have to dedicate resources to that part of security for which the vendor is responsible. An example of this division is given here. | [69] |

[4] https://www.ibm.com/docs/en/devops-release/6.2.1?topic=release-configuring-servers-high-availability
[5] https://www.sap.com/products/technology-platform/integration-suite/pricing.html
[6] https://www.softwareadvice.com/bi/dell-boomi-profile/
[7] https://www.salesforce.com/editions-pricing/integration/anypoint-platform/
[8] https://www.opentext.com/file_source/OpenText/en_US/PDF/opentext-wp-total-cost-of-ownership-of-enterprise-integration-solutions-en.pdf

| | | | | |
|---|---|---|---|---|
| | infrastructure is better than the security offered by iPaaS vendors[9]. | | | |
| | Also, because of the increased complexity of the stack (different databases, operating systems, different software packages), assessing the entire IT infrastructure on its security level is in itself difficult to do. | [18] | Limited visibility and control in/of the underlying infrastructure can result in specific security and compliance issues (physical location of servers, sharing servers with multiple users). | [21] |
| | One of the advantages of a TIP is the full visibility and control a company has over its infrastructure. This can be critical for highly regulated industries or companies with specific security requirements (physical server location, data protection rules etc.) | [21] | Sending sensitive data across the internet (or buying an expensive data line) can be a downside. | [21] |
| | | | | |
| Scalability | First, understand the concept of scaling up and scaling out. A company is responsible for scaling its TIP according to the traffic. Hence, it needs a team as described in resource utilisation to manage this. It does not only include acquiring more resources but also: Planning, setting up and managing the expansion process. The scaling process could also influence the availability because the server could temporarily be down[10]. The absence of the responsibility for scaling the resources is an important reason for companies to move to iPaaS. | See Foot -note | Scalability is mentioned as a big advantage of iPaaS. However, iPaaS does not endlessly scale with the click of a button. Scalability has been considered in the design of cloud solutions (iPaaS included), which means it has features to scale. However, this does not mean that it scales unlimitedly at all times. This depends on the specific service used. It also does not mean that realizing a scalable solution is easy, this can still require expertise to be performed effectively | [7] |
| | Consequently, one of the problems for TIP is **peak demand.** To promise high availability, the company has to have infrastructure capable of handling the peaks. However, at less busy times, a part of the infrastructure is unused meaning that fulfilling the peak demand is costly. | [47] | Because iPaaS is designed with scalability in mind it does scale more flexibly than TIP. There are also more options to realize scaling. One of the advantages is that specific services can be scaled independently which can result in lower costs. So costs are lower due to flexible scaling in general but also flexible scaling of individual services. | (4) |

[9] https://www.opentext.com/file_source/OpenText/en_US/PDF/opentext-wp-total-cost-of-ownership-of-enterprise-integration-solutions-en.pdf
[10] https://learn.microsoft.com/en-us/biztalk/core/scaling-your-solutions

| | | | | |
|---|---|---|---|---|
| | | | Some providers have the option of automatic scaling[11] (for certain services) where the platform does not need human intervention to scale according to traffic. However, this option will come with extra costs. | |
| | | | | |
| Availability | The key answer to realising high availability for a TIP is to implement r**edundancy** which refers to the duplication of critical components or functions of a system intending to increase the reliability of the system, ensuring its availability, and enhancing fault tolerance. To implement redundancy, at least two servers are needed. One passive and one active server. If the active server fails, the passive server should be able to take over. Note, that this does mean that the passive server has to be configured and tested like an active server with the correct number of cores and licensing, even though it stays idle most of the time. This outlines the resource intensiveness that is included with offering a highly available solution on-premise | [47] | Availability has been considered a requirement when developing the cloud iPaaS services. As a result, organizations have fewer responsibilities in realizing a highly available solution and there are also more options in terms of availability at the vendor's side. | [54] |
| | Even with high redundancy implemented, it could still be that case that systems fail. **Recovery planning** should be implemented to recover from these failures. This is entirely up to the organization, requiring detailed strategies for data backup, offsite storage, and restoration (this also needs regular testing). This often involves significant resources and planning. | [25] | Redundancy is built in at multiple levels of the cloud infrastructure. Cloud providers manage multiple data centres in possibly multiple regions with multiple replications of servers. Hence, it offers geographical spread, something a TIP cannot. | (4) |
| | | | The cloud provider may have a malfunction. This is then often for a specific service in a specific region. This is not desired but it is the question if a company can deliver a better performance in terms of availability than the cloud. | [25] |
| | | | | |

[11] https://learn.microsoft.com/en-us/azure/app-service/manage-automatic-scaling?tabs=azure-portal

| | | | | |
|---|---|---|---|---|
| Performance | Can theoretically, achieve higher performance **for internal communication only** compared to iPaaS because the information does not have to be sent to the cloud servers. However, this only entails synchronous communication and will only apply to a small number of interfaces. | (1) | If a company wants to have performance guarantees, a plan-based pricing model is favoured for iPaaS. Further performance information can often be found in the Service Level Agreement (SLA) of the specific vendor. | [69] |
| | | | | |
| Compatibility | Is promoted as better suited to handle legacy system integrations. However, little evidence is provided for this. This seems to be dependent on the different types of adapters that are included in the platform and the possibility of developing new adapters. These features are often also present in iPaaS | [93] [7] | The decision for iPaaS often follows a more general cloud initiative (cloud strategy). Possibly of one big system moving to the cloud. The reasoning is that the big system moving to the cloud already required a substantial number of integrations to be redeveloped. Therefore, the integration platform is open for change. | (5) |
| | The stack of hardware and software that should be organized to implement TIP does make it more complex to implement the platform within an existing organization infrastructure. | [31] | Vendors have a cloud-first approach which means that new adapters will be launched first on the iPaaS platform making them more compatible with new technologies. It is unsure if these adapters would then also be launched on the TIP. | (4, 5) |
| | | | | |
| Usability | Setting up a TIP from scratch requires hardware and software preparations which iPaaS do not. Hence, it is way more complex and time-consuming to fulfil this. It should not be underestimated how difficult this is. | (6) [73] | Resource utilization showed that there is still a substantial team needed to realize an iPaaS platform. iPaaS does not have the hardware and software requirements but still, setting up the integration platform is complex but less complex than TIP. There will still be difficult choices that have to be made when setting up the platform but less than TIP. | (6) |
| | TIP is often one product, this limits the creativity options in terms of how to solve integration problems but can also be praised for the standardised, simple approach towards developing integrations. Whether this is desired depends on the specific integration scenario. | (4) | In literature, iPaaS is often described as easier to use with a lower entry-level in terms of expertise. The low-code option would perhaps even allow business people to develop integrations. This is true for simpler integrations. When things become complex, it is important to have the correct expertise. However, the low-code options will provide useful tools for experts allowing them to develop integrations faster and easier. | [40] |

| | | | | |
|---|---|---|---|---|
| | | | iPaaS will probably have i more diverse toolkit than TIP, including services that cover API management as well as a more diverse offering of pre-built adapters than TIP. | [40] |
| | | | | |
| Maintainability | For a TIP, platform updates require more responsibility from the company. Take the example of upgrading BizTalk version 2016 to 2020 which is a significant undertaking for a company[12]. It requires substantial planning resource allocation and potential downtime to complete the required steps as well as the need to rigorously test the integrations after the updates are implemented. | See Foot -note | Less action is required from companies when platform updates occur because the vendor manages the entire infrastructure which allows it to implement updates automatically. However, sometimes still upgrade actions are required from companies. Hence, companies have fewer responsibilities because they have a smaller stack to manage. Therefore, less updates and less complexity. | [54] [42] |
| | "in interviews" an often-heard argument is the end-of-life of TIP. This should always be checked per vendor but TIPs like Sap Process Orchestration and Microsoft BizTalk have support till 2030. However, if the end-of-life is close by, it is logically unwise to choose that TIP for integrations. | | iPaaS is considered to be vendors' latest-greatest which is referred to as "the most recent and advanced offering in a vendors' product lineup. This is partly true and requires some further investigation. It is true that vendors adopt a cloud-first initiative where they roll out their latest features and innovations on the iPaaS platform and because updates are handled automatically, users of iPaaS have immediate access to these features. However, this also means that vendors try new approaches and sometimes, these approaches appear to work worse than their TIP counterparts. Then, these features are retracted from the platform. As such, the iPaaS does not have an end-of-life but the features of the iPaaS can have it. Resulting in a platform that is still **under construction** which could impact adoption by companies. | (4) |
| | The end-of-life/latest-greatest discussion also has important implications for the **availability of external expertise**. Often, integration platforms need to be implemented and possibly maintained with the help of external experts. These experts will most likely want to work and develop themselves in the direction of the greatest product. Therefore, it is likely that the | [42] | | |

[12] https://learn.microsoft.com/en-us/biztalk/install-and-config-guides/upgrade-to-biztalk-server-2020

| | | | | |
|---|---|---|---|---|
| | number of experts willing to work with TIPs decreases over time. | | | |

## Appendix H Second Expert Review Transcription

| | |
|---|---|
| | 1: [redacted] 26-02-2024<br>W: A summary of my research. I conduct my research on the concept of iPaaS and how it compares to a traditional integration platform. I approach this comparison from functional and non-functional capabilities and till now, I have developed a full capability model. Then, I continued my research on the nFRs because they are more interesting to compare to iPaaS and TIP.<br>1: I agree<br>W: I developed a slide deck that I would like your feedback on during this session. The goal is to possibly improve the tool and I would like your expert opinion on the content. So can you share your screen and walk through the tool? The tool should be self-explanatory. |
| Pricing structure | 1: interesting, this looks nice.<br>1: This also seems to be correct. The iPaaS pricing model is indeed complex and different options are available. You see that vendors struggle with this.<br>1: yes I also recognize this from Mulesoft, especially on-premise that you have to determine how many cores you have. So very relatable.<br>1: For me, it is very clear. However, I am an expert so I fully understand the different concepts. So from my point of view, it is clear.<br>W: Okay but it would be helpful to have some sort of integration expertise<br>1: Yes, I think so. It is very nice that you use an example |
| Resource utilisation | 1: Yes good, but I think it would more clearer if you would use a separate column for the comments. Because then it is more clear how iPaaS relates to TIP.<br>1: And also, you have the purchase of the software but also the set-up of the software. Here it falls under implementation costs but it should be more clear that everything has to be implemented and configured. So make that more clear under implementation costs. Because that is a project (ordering hardware, software and installing it) requires so many different teams that it is a project of months. Whereas iPaaS only requires a payment and then from day one you can use it. This is a big differentiator Be aware that you do pay for this functionality. |
| **Security**: | 1: I think the model is nice. However, what I am looking for with an integration platform. Look, the application platform is covered on the vendor side and they will have very nice security checks as you described. However, at the platform, I can implement a very insecure API which harms the security. So the integration that you implement (built by developers) is under the responsibility of the company. So make sure that comes into the figure.<br>W: Clear, will update that<br>1: Yes, you can always go into more detail but I think this is a nice summary |
| **Scalability** | 1: Yes true but. I think the following is reflected in your description but maybe you can make it more evident: Does not matter if it is TIP or iPaaS, you need expertise in the specific platform. You need to understand the situation that requires scalability. The big difference between TIP and iPaaS is that with TIP, your infrastructure also needs to be aligned for scalability. So besides being a platform expert, you need to arrange a lot of other aspects that might require expertise from a different team. In both concepts, you need platform experts. But iPaaS does take a lot of responsibility for the other components required for scalability.<br>W: Yes, so important to show that you need knowledge of your IT infrastructure configuration |
| **Availability** | 1: Yes everything is correct |

| | |
|---|---|
| **Performance** | 1: It is important to note that TIPs are faster when the applications are hosted and deployed on-premise. Because you can also host applications in the cloud (private cloud). Then it can become faster to use iPAaS in that same data center as your other applications. So it works two ways, premise applications can have higher performance with TIP, and cloud-hosted applications can have higher performance with iPaaS<br>W: yes and it also depends on the specific use case that determines if latency becomes a big issue<br>1: you could maybe use some examples of why applications have to be on-premise (wet en regelgeving or a production environment (VDL with cutting robots in a production environment) |
| **Compatibility** | 1: The statement that iPaaS is promoted to be better suited to handle legacy system integration. I think that most regular integrations are supported at this moment. I also think that modern integration like streaming API, needs the latest and greatest of integration and I do not know if the TIP has it then but know for sure that iPaaS has it.<br>W: Yes I agree, that is also described in the iPaaS part.<br>1: Regarding the second statement of TIP, I also do not know for sure if that should be compatible because we described this already in resource utilization<br>W: Yes, I think you are correct, will adjust that |
| **Usability** | 1: does this cover the usability of the platform itself or the usability of building the integrations?<br>W: both but maybe good to include more in the description<br>1: what you see with iPaaS is that there is a division into two groups of platforms. One (Boomi and Workato) is more on the citizen integrator focused (more p2p, faster integrations). The other group (Tibco and Mulesoft) is more focused on being used by developers, and more complex integrations. This division is not notable at TIP. TIP platforms always require developers and are always more complex platforms<br>W: Yes good to know. |
| **Maintainability** | 1: The point of iPaaS is “under construction”, do not recognize that with iPaaS. Do you have examples?<br>W: For example Azure<br>1: Ah okay, Google and Azure yes. Mulesoft, Boomi, and Workato do not do this. They launch features and rarely retract these features.<br>1: Yes, in general, you do see that updates are way fast and require less downtime (if needed at all) in iPaaS compared to TIP. This is not black and white but is generally true. |
| **General Comments on the Tool**<br>Completeness | W: Do you think that capabilities are missing in the tool?<br>1: nope, I think everything is included<br><br>W: Does the model improve on the identified model?<br>1: I am not that up-to-date about any models. However, I think it is in line with what we do. We often use a list of requirements that we have used in previous comparisons. This deck is complete and fully up to date. So I would use this tool for a 5 |
| **Capability model** | |

| | |
|---|---|
| Capabilities belonging to iPaaS or TIP<br><br><br>Use pp and capability model to choose integration platforms | 1: It depends per tool how complete they are. If I look at Mulesoft and Oracle. Their TIP and iPaas have both all of the capability groups.<br>1: However, they are less good at data management capabilities but that is also not an integration capability. I think only data synchronization and bulk data movement are integration capabilities.<br><br>1: Yes. However, it becomes very client-specific what kind of requirements they prefer over others. We take an approach to clients in which we discuss all the different options that they want in such a capability model is a nice structured approach to do that.<br>W: And that can in this model be every different combination?<br>1: yes exact<br>1: Many people are enthusiastic about the cloud. I know a lot of big organizations that have good IT departments that work well together. Then you still need to configure more for TIP compared to the cloud. But if that is organized well, then it is not very difficult to add servers. So it is very dependent on how mature your organization is. The same goes for security, if you do not have a good security department then it is an extra pre to move to the cloud. |

| | |
|---|---|
| Pricing structure | 2: [redacted]27-02-2024<br>2: I think people are moving away from the fixed pricing model slowly. Most are moving to consumptions, even tool providers like Oracle, SAP, and Boomi. They are moving away from fixed consumption. Because they want to keep up with what hyperscalers (azure AWS< Google). Nowadays, what we are seeing with the client is, they are heavily investing In hyperscalers. Hyperscalers are intelligent. They have come up with their integration services. Rather than utilizing Mulesoft SAP or Boomi. Customers have already got hyperschalers in their landscape and are therefore more inclined to utilise their services. And their services use consumption-based pricing models. So the competition is high and the ipaas providers are slowly bringing down their licensed-based models and are coming up with very innovative plans for their licensing model.<br>2: I think that vendors are coming up with really good tools to understand the pricing. The only thing they need is the technical details from customers to estimate pricing. Resources like processing capability, and what kind of message transactions are happening. Vendors would need these details in their pricing tools to provide good estimates. |
| Resource utilisation | 2: I am thinking, about what else could be included. What do you mean by external expertise<br>W: External experts like Deloitte consultants or vendors offered expertise<br>2: For ipaas, it is becoming yes or maybe, new vendors are coming up in the market who do not need external expertise to be implemented (Fabric?, Flamengo?). They are built with very intuitive designs). When it comes to SAP, mulesoft etc then technical expertise is needed.<br>2: I would agree with everything else |
| **Security**: | 2: Yes security is a very big headache for the customers when they are using TIP. Even only a few customers are ready to take that headache. Let's say you have the banking sector, that wants to have their integration and data on-premise. However, SAP for example provides agents that can be deployed on-premise. In that way, they can meet their security requirement.<br>W: That would still mean that you are responsible for a stack of the hardware and security. That comes with the entire headache |

| | |
|---|---|
| | 2: I agree with that point. Even iPaaS providers release security patches and upgrades but there will be security architects at the customer who will decide if the security patch or upgrade is installed. Most of the time this is yes, but sometimes they want to test it. Sometimes iPaaS providers even open more gateways for external people to exploit the system. Therefore, the security architects want to test the patching so there will still be some security team needed to assess what patching is provided by the vendor. **(shared responsibility story)** |
| **Scalability** | 2: The costs lay in that you need to set the rules. Some vendors provide autoscaling free of charge<br>W: Could you then name a scenario where it is not wishful to use autoscaling?<br>2: No customers often opt for it. You do need to set a maximum for the scaling. I have not come across any customer who said I do not need autoscaling.<br>2: Maybe add something like business continuity |
| **Availability** | 2: This is a pain for customers when they are using on-premise. Even in iPaaS, if they need high availability, it comes with a price. If they are opting for hyper scalers, and want their solution in different zones deployed, it comes with costs. But it is very easy to manage. iPaaS always have an upper hand in availability than the tip.<br>2: Sometimes, there is a loss of data when systems fail.<br>2: If you use something like Mulesoft or Oracle, availability is high and easy to manage. If you are using hyperscalers then you need to configure availability manually.<br>2: I agree with what you wrote here. |
| **Performance** | 2: Performance also depends on how you are designing your integrations. (on the developers). If they have very complex logic, if they are not efficiently designing the interface it would affect the performance<br>2: Another part of the performance is related to the application landscape of a specific vendor. Take SAP, the performance is better when SAP iPaaS is used to integrate a SAP environment because it is so neatly stitched to each other. This does increase vendor lock-in. + It is not assured that performance is increased. So it is up to the customer if they want to accept the vendor-lock-in |
| **Compatibility** | 2: Both ipaas and TIP are compatible with most products. There is a misconception that Oracle can only to Oracle to Oracle integration. This is not true.<br>2: I don't think both TIP and iPaas have problems in terms of compatibility. |
| **Usability** | 2: When you have a TIP, you need an entirely different team to realize the platform. With iPaas it is very easy to understand. It is not difficult to set up the infrastructure. Everything is in the same UI. When it comes to TIP, they need to set up everything. Need someone who goes into the data centre<br>2: Pre-built adapters are offered by both systems, but it is indeed more in iPAas now. |
| **Maintainability** | 2: yes I agree with these points |
| **General Comments on the Tool**<br>Completeness<br>Does the model improve on the identified model | W: Would you add any NFC?<br>2: I think you have covered most of the commons nfr's<br>W: Does this model improve understanding better than current models?<br>2: The information that is provided. You have put together a lot of information that is widely spread. I think your slide deck provides people with details of what the key differences are. I think this pack helps a lot of people to readily understand the differences. Furthermore, this |

| | |
|---|---|
| Usability | deck would help people like me to convince customers to move from on-premise to ipaas. IT would show which aspects are relieving customers of their duty. |
| **Capability model** | 2: You could add AI capabilities. (he talks hare about an AI mapper)<br>On-premise needs a different API management (external) whereas cloud ipaas have built-in API management capabilities |

| | |
|---|---|
| **Pricing model** | 3 [redacted] 28-02-2024<br>3: No further comments |
| Resource utilisation | 3: This is a very nice overview! Very useful |
| **Security**: | 3: Now it is pretty generic, the questions if you would want to dive into the specific security features. But maybe that is not necessary.<br>3: At Oracle, you have identified management. That gets you a single sign-on in the cloud which is very convenient for security.<br>3: Here comes also the Brexit Russia story that you perhaps could add as an example |
| **Scalability** | 3: What also comes forward with scalability? It has a cost impact but also a turnaround impact. You have to through an administrative cycle with iPaaS when you want to scale. TIP, you can make the decision yourself to scale. With iPAaS, you have to make a scalability request at Oracle, and then you first need to fill out a question list because they want to see if your request is justified. Vendors do not always easily go with these requests. This is a disadvantage of updates<br>W: You hear different sounds on this. I also hear people saying that automatic scaling is possible and sometimes free.<br>3: It depends, the scalability can be asses in different areas. For example, on the IP, they can say that you are allowed, for the pricing agreement, you can send X GB over the line. However, let's say that it gives performance issues. Then it becomes a difficult discussion with the vendor. On one side, you can probably say that you want automatic scaling. When you have a performance problem it can become more difficult. That is a different scalability discussion.<br>W: Good to mention that the scalability agreements can differ per vendor and specific agreement. |
| **Availability** | 3: This is a good example of a real cloud advantage.<br>Everything else is clear |
| **Performance** | 3: Yes it is listed in the SLA but it is often fussy. Performance is often constated after the fact. It all depends if you make hard agreements with the vendor.<br>3: The performance of an integration can also depend on the applications that are being integrated.<br>3: Also add here the different aspects that influence performance |
| **Compatibility** | 3: Compatialbity is good for both types of platforms. It depends more on the features that are present in the cloud. You could try to search for features that are present in the cloud but not in TIP.<br>3: Cloud is more compatible with cloud services. |
| **Usability** | 3: Something we see, at the moment a product is introduced to the cloud for the first time. It is less feature-rich than the on-premise offering. Over time, these features will be added slowly. So in the early life of cloud products, it is less feature-rich. |

| | |
|---|---|
| | 3: Why do you say that iPaaS has a more diverse toolkit? Because API management can also be included in the TIP. |
| **Maintainability** | 3: These look all complete |
| **General Comments on the Tool**<br>Completeness<br><br>Does the model improve on the identified model | W: Is the model complete?<br>3: No, you cover the entire spectrum. Very nice.<br>W: Does this model improve understanding better than current models?<br>3: A 5, This is actually, the first model that covers the entire spectrum. Often you see that people speak on NFC. This is a very nice starting point to take the customer on the start of their journey. |
| **Capability model** | 3: Data management is not an integration capability. Only data synchronization, bulk data movement and business rule engine. They are relevant but others are not.<br>3: Theoretically, all of these capabilities can be TIP or iPaaS. However, it can be that some features have to be acquired through other products. |

| | |
|---|---|
| Pricing structure | 4 [redacted] 28-02-2024<br>4: The pricing model of Mulesoft and Boomi is different again so maybe look into that as well. Boomi uses per-core and Mulesoft uses a fixed pricing.<br>4: That is why I said that pricing is difficult because it differs so much per vendor<br>W: Yes so that is why it is said in the beginning that pricing is complex. |
| Resource utilisation | 4: Make a pie chart or something that can be used to effectively see which part of the budget is allocated to which part. (use ballpark figures)<br>4: Is tip not also low-code?<br>W: Yes<br>4: So it indeed clearly shows that there are a lot of things that extra to be managed with TIP.<br>4: I can imagine that iPaaS is cheaper. However, the pricing is often only the starting price and if you want to scale, it will become a lot more expensive. Whereas with TIP it is rather straightforward. You just have to add and configure the hardware. |
| **Security**: | 4: Is it justifiable in these times that companies think that they cannot send their data across the internet under security considerations? It looks in the line “sending data across the internet” that the security of the cloud is not sufficient. Even though it is. So the argument should reflect more that this is because of the rules of a company. |
| **Scalability** | 4: Where did you find automatic scaling?<br>W: Azure and some respondents indicated that most vendors have auto-scaling<br>4: That is bullshit, Mulesoft and Boomi do not have it. You have to say beforehand how many cores you want to utilize. If you exceed this number of cores you have to go through an administrative cycle and connect to Boomi support. So automatic scaling does not exist that often.<br>4: Ask a Mulesoft expert on automatic scaling |

| | |
|---|---|
| **Availability** | 4: Usually, availability is related to servers that fall out, only to scalability with peak demands.<br>4: It is questionable if the vendor automatically offers geographical availability. If you want this as a vendor then you would probably have to configure that specifically and pay more for this. |
| **Performance** | 4: Give an example of internal communication that can be faster. |
| **Compatibility** | 4: Good slide |
| **Usability** | 4: Yes indeed, every time they make a tool meant for citizen integrators, it seems not to work and experts are required pretty soon.<br>4: The development of integration is only a small piece of the puzzle. Testing etc takes often more time for both TIP and iPaaS.<br>4: Yes for tips you always need experts, it is never meant for citizen integrators. |
| **Maintainability** | 4: With iPaaS, a vendor has more control over the enforcement of accepting updates. Whereas with TIP, it depends on whether companies accept the update |
| **General Comments on the Tool** | W: Do you think that there is information missing in the tool?<br>4:You cover the main points, there are however some things that should be included but we have to discuss them now.<br>W: Does this model improve understanding better than current models?<br>4: A 4, I am not aware of other models so it improves the current approach. |

| | |
|---|---|
| Pricing structure | 5: [redacted]28-02-2024<br>5: Yes we also use a plan-based licensing model. I know that other companies use a consumption-based pricing model. We choose a plan-based because then we can ensure customers that they can send unlimited data of their integrations. They do pay if they want to make extra integrations.<br>5: I know from our clients that Mulesoft starts with 100000 and then you do not have everything that the platform offers.<br>5: We start with a small package, (a flow pack of 5 integrations) then If you want to be able to develop more integrations you pay more.<br>5: We also see that is very difficult for us to get into new companies with existing IPs because the vendor lock-in is prominently present. |
| Resource utilisation | 5: Maybe not name it 'needed for' because you do need Hardware for iPaaS. |
| **Security**: | 5: We used to struggle a lot with implementing security features and upgrades because clients did not want to migrate. Now it is easier because we can configure the updates automatically for a large part.<br>5: Important to mention that it is still difficult to enforce the security upgrade on iPaaS. |
| **Scalability** | 5: Very well written. Autoscaling is possible but a lot of our clients do not choose it because you then have to also determine a maximum. It is also only available for a specific part of our services. Furthermore, if you want autoscaling you go to a pay-per-use pricing model and prices will be a lot more expensive compared to the original plan-based pricing model. They would rather buy more servers for a plan-based discount than be surprised by scalable costs.<br>5: It is also very easy to scale with us. You can just add resources which is a manual action. The client then still has to configure how those machines are allocated to their integrations.<br>5: Even a post-NL does not do the auto-scaling option with the us |
| **Availability** | 5: This is the big advantage of cloud (iPaaS) |

| | |
|---|---|
| | 5: With Emagiz, we offer geographical availability. If you want this, we will make sure the cloud provider has two separate data centres in different geographical zones |
| **Performance** | 5: What you can add is. In the cloud, you have limited options. On-prem you can tweak everything yourself but it is very costly. In the cloud, you have fewer of these performance options. |
| **Compatibility** | 5: There is a part of legacy applications that can only communicate in a certain format. Some of the legacy systems require some custom-built software to be able to integrate.<br>5: It does not depend on the TIP or iPaaS type of platform. It depends on the toolkit that is included in that platform that you can use to integrate your specific set of application landscapes. |
| **Usability** | 5: For TIP, you always need integration experts. For iPaaS, you can sometimes use business integrators but when things become more complex, integration experts are required to make the integration. Integration is never easy |
| **Maintainability** | 5: No comment, everything looks fine |
| **General Comments on the Tool**<br>Completeness<br><br>Does the model improve on the identified model? | Yes it is very complete<br><br>Yes definitely, I would give it a 5.<br>5: Complements for how you wrote the tool |

| | |
|---|---|
| **Individual slides of the tool**<br>Pricing structure | 6: [redacted]01-03-2024<br>6: It is only difficult to estimate costs in the initial phase. The moment you run an iPaaS, after 3,4 months it is pretty easy to estimate the costs.<br>6: Also, the experts of Deloitte can help with this kind of cost estimate. |
| Resource utilization<br>Update information: | 6: Good,<br>6: developing integration is not going to be a less of hurdle so you will need the same kind of development tool |
| **Security**: | 6: True, you cover all the points |
| **Scalability** | W: How is scalability organized at Oracle:<br>6: Pretty easy to scale. You increase the components and it is very user-friendly. You do need expertise to understand what is being scaled to do it effectively. |
| **Availability** | 6: Pretty straightforward. You cover everything |
| **Performance** | 6: He agrees with the redesign part of integrations. Performance is more affected by the design of the integrations than by the specific platform |

| | |
|---|---|
| **Compatibility** | 6: It's true, it also depends on the current application cloud and specifically the vendor systems. SAP Oracle etc are more inclined to use that iPaaS platform. |
| **Usability** | 6: Yes, agreed |
| **General Comments on the Tool**<br>Completeness<br>Does the model improve on the identified model<br><br>Usability | s<br>I think that you cover almost everything yeah<br>Yes, I would give it a 4. There is one thing. Comparison is there but I would say that the key differentiator with the clients is when they move one of their big applications to the cloud.<br>Yes, very usable |
| **Capability model**<br>Capabilities belonging to iPaaS or TIP | 6: encryption is not only a B2B capability |

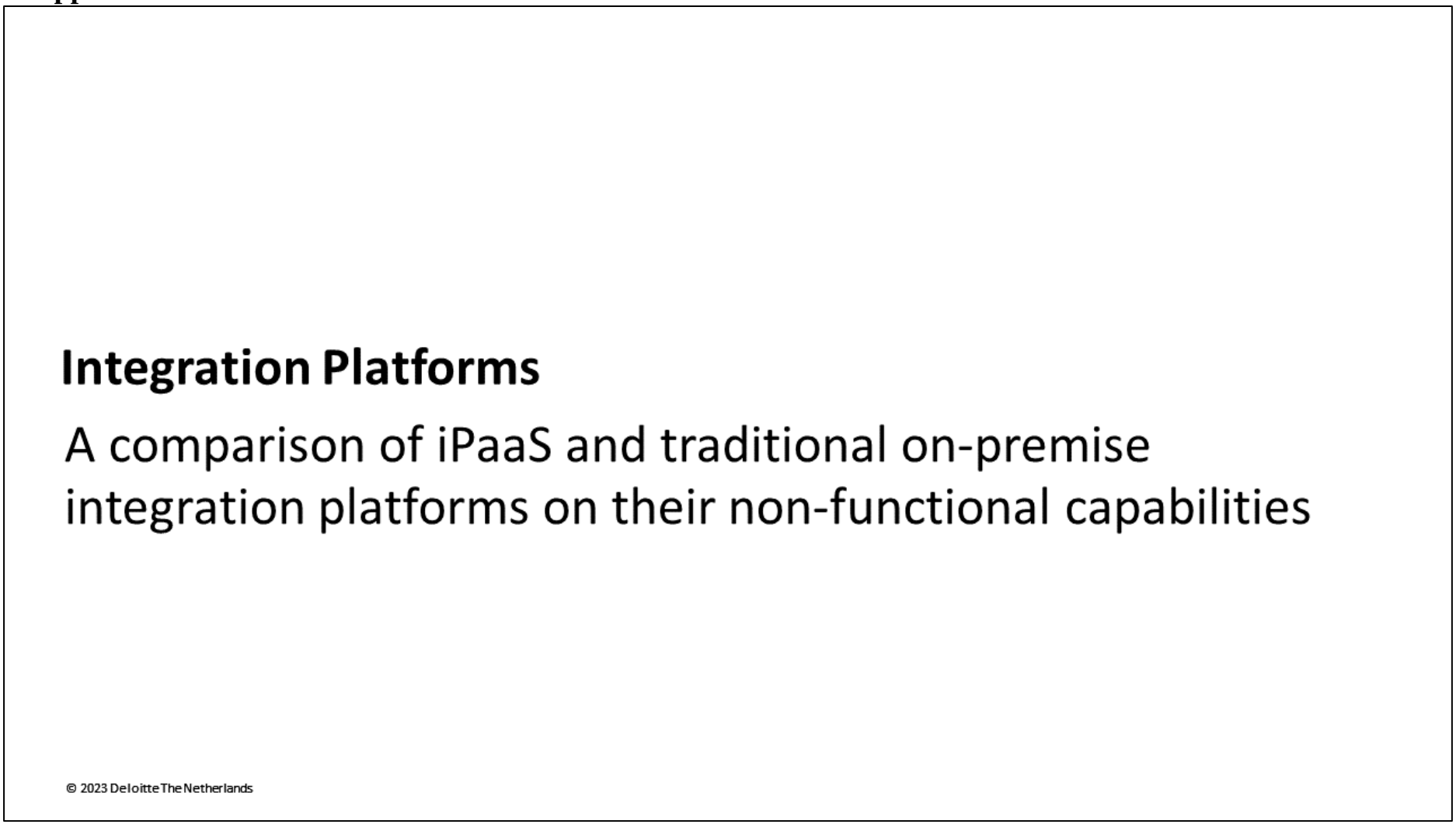
| | |
|---|---|
| Pricing structure | 7: [redacted]13-03-2024<br>7: A traditional one has more upfront pricing (hardware, software etc).<br>7: What is the function of this? I remember, that you wanted to identify the requirements that should be met to go to an iPaaS<br>W: not requirements, a clear overview of how TIP and iPaaS differ from each other<br>7: Often when you look at the license, it is different for a TIP. In an iPAaS, you do not become the owner of the iPaaS. Also, the operational side has a big cost difference (this is in the resource utilization slide) |
| Resource utilisation | 7: It does not compare the cost aspects. It lists the cost aspects but it would be interesting to see some sort of pie chart. You want to see how much each cost aspect weighs.<br>7: With storage, it can be that you have a storage on-premise. So yes, you do not need it but it is possible.<br>7: The dependency on external expertise is larger for TIP than for iPaaS.<br>7: Make it more clear that iPaaS is licensing costs, not integration platform software. |
| **Security**: | 7: To make it more usable, use some sort of tabular form for the comparison because now I have to compare to big laps of text.<br>7: You can also make a kind of shortlist that summarizes the content of each slide. Then for the academic part, you can use the larger text but for a client meeting this shortlist can be used |
| **Scalability** | 7: Maybe look into Serverless iPaaS?<br>7: Also add that you have more control in TIP because you can scale whenever you want. With iPaaS you have to go through a cycle |

| | |
|---|---|
| **Availability** | 7: In practice, implementing redundancy almost does not happen with TIP. They do not have a duplicate server park. So it is a theoretical point to make clear that most companies do not have this resulting in a single point of failure. If servers are offline then this often impacts the availability. – **Single Point of Failure**<br>7: A lot of clients want to have a very high availability without actually considering the need for this availability and the extra incurred resources/costs. |
| **Performance** | 7: Performance is more predictable on the TIP side because it covers a physical park.<br>7: I think this covers everything |
| **Compatibility** | 7: iPaaS are indeed less compatible with really old systems. However, iPaaS does give you more freedom to build custom integrations. |
| **Usability** | 7: Make more nuance on the business people. I do not know one example where it is useful to let business people build integrations. Even for simple integration, they are likely to exist in a complex landscape and it always helps it have technical people building them. I agree that the low-code solution is more user-friendly.<br>7: Besides pre-built adapters, iPaaS has a kind of marketplace where user share their pre-built integrations that can be used by other users. This has a great impact on the usability. |
| **Maintainability** | 7: Single point of knowledge of the expertise. The group of people who know how to build the platform is reducing and also the people who built the IP might leave the company.<br>7 The terms referred to in this piece are **continuous improvements** and **vendor-managed updates.** |

| | |
|---|---|
| Pricing structure | 8: [redacted]**22-03-2024**<br>8: It is more important to offer a pay-per-use/consumption-based pricing model than the actual iPaaS server footprint. For example, SAP offers a charge per transaction which makes it an impeding partner. |
| Resource utilisation | 8: one of the roles I am missing in the integration team is the Architect/design lead. This person will be responsible for the governance of the developed integrations. Hence, code review and code design are under its responsibilities.<br>8: Everything else seems to be covered. |
| **Security**: | 8: This looks complete, nice to make the responsibility distinction between TIP and iPaaS on security. |
| **Scalability** | 8: I miss the concept of scaling down, something very important in iPAaS solutions.<br>W: Yes it is part of scalability in general, it is not covered in TIP because I do not think that scaling down happens often in a physical server park.<br>8: Yes that is correct. |
| **Availability** | 8: iPaaS is the reliability of the internet. If the internet is down then the entire platform malfunctions. However, this can also be the case for TIP if an internet connection is used to transfer data. Examples of these situations are for example in the Brazilian jungle where the Internet is not reliable. |

| | |
|---|---|
| **Performance** | 8: I think that the remaining slides look very complete with nice descriptions. |
| **Compatibility** | 8: Maybe rewrite the first sentence of the iPAaS section because it does not read very easily. |
| **Usability** | 8: looks complete |
| **Maintainability** | **8:** Looks good. |

**Appendix I NFC Tool 1.1 Slides Screenshots**



## Introduction

01 This slide deck compares the concept of traditional on-premise integration platforms with the concept of an integration Platform-as-a-Service (Click here to see what this slide deck identifies as a traditional integration platform (TIP) and an integration Platform-as-a-Service (iPaaS)).

02 It structures this comparison on nine non-functional capabilities identified in an earlier part of this research as important to integration platforms.

03 The goal is to inform users of the different aspects they should consider when deciding on one of the two integration platform concepts.

## Management Summary

The next two slides summarise the content of this comparison between on-premise integration platforms (TIP) and integration Platform-as-a-Service (iPaaS) based on 9 non-functional capabilities

# Management Summary (1/2)

## Comparing TIP and iPaaS

| | TIP | iPaaS |
|---|---|---|
| Pricing Model | • Uses a licensing-based model, leading to upfront costs.<br>• Licensing costs are a small portion of Total Cost of Ownership (TCO)<br>• Licensing costs increase as the number of required servers increases. This could be caused by availability and scalability requirements. | • Specific pricing model differs per vendor and can be plan-based, consumption-based or a combination of both.<br>• iPaaS pricing model allows organizations to only pay for the services they actually require.<br>• Consumption-based pricing model offers low start-up costs.<br>• Costs can vary widely and are difficult to estimate beforehand. |
| Resource Utilization | • The full stack (hardware, software, implementation costs and staffing costs) is under the responsibility of the organization.<br>• Both TIP and iPaaS require similar resources for the integration team.<br>• TIP will have a more costly configuration because of the extra hardware and software that needs to be configured. | • Lifts organizations from the responsibility for hardware and part of the software (operating system and security software).<br>• Both TIP and iPaaS require similar resources for the integration team.<br>• Even though the organization is not responsible for hardware. It still needs to buy hardware artefacts from the vendor. |
| Security | • Requires substantial resources for security infrastructure.<br>• Offers full control over security configurations.<br>• Increased complexity makes security assessment challenging.<br>• Critical for industries with specific security requirements.<br>• An organizations on-premise security infrastructure's effectiveness can be debatable compared to iPaaS vendors' security infrastructure. | • Automatic updates offer the latest security features.<br>• Limited control can introduce specific security and compliance issues.<br>• Utilizes a shared responsibility model for security.<br>• Organization retains accountability for integration security.<br>• Physical server locations, shared environments and the use of internet can be concerns of organizations who want to move to iPaaS. |
| Scalability | • Scaling requires substantial planning and resources.<br>• Infrastructure for peak demand can be costly and underutilized during off-peak times.<br>• Scaling can influence availability due to potential downtime. | • Designed with scalability in mind, offering more flexibility.<br>• Allows for independent scaling of specific services to lower costs.<br>• Platform expertise is still crucial for effective scalability.<br>• The ease of scalability depends on specific vendor agreements and services used. |

# Management Summary (2/2)

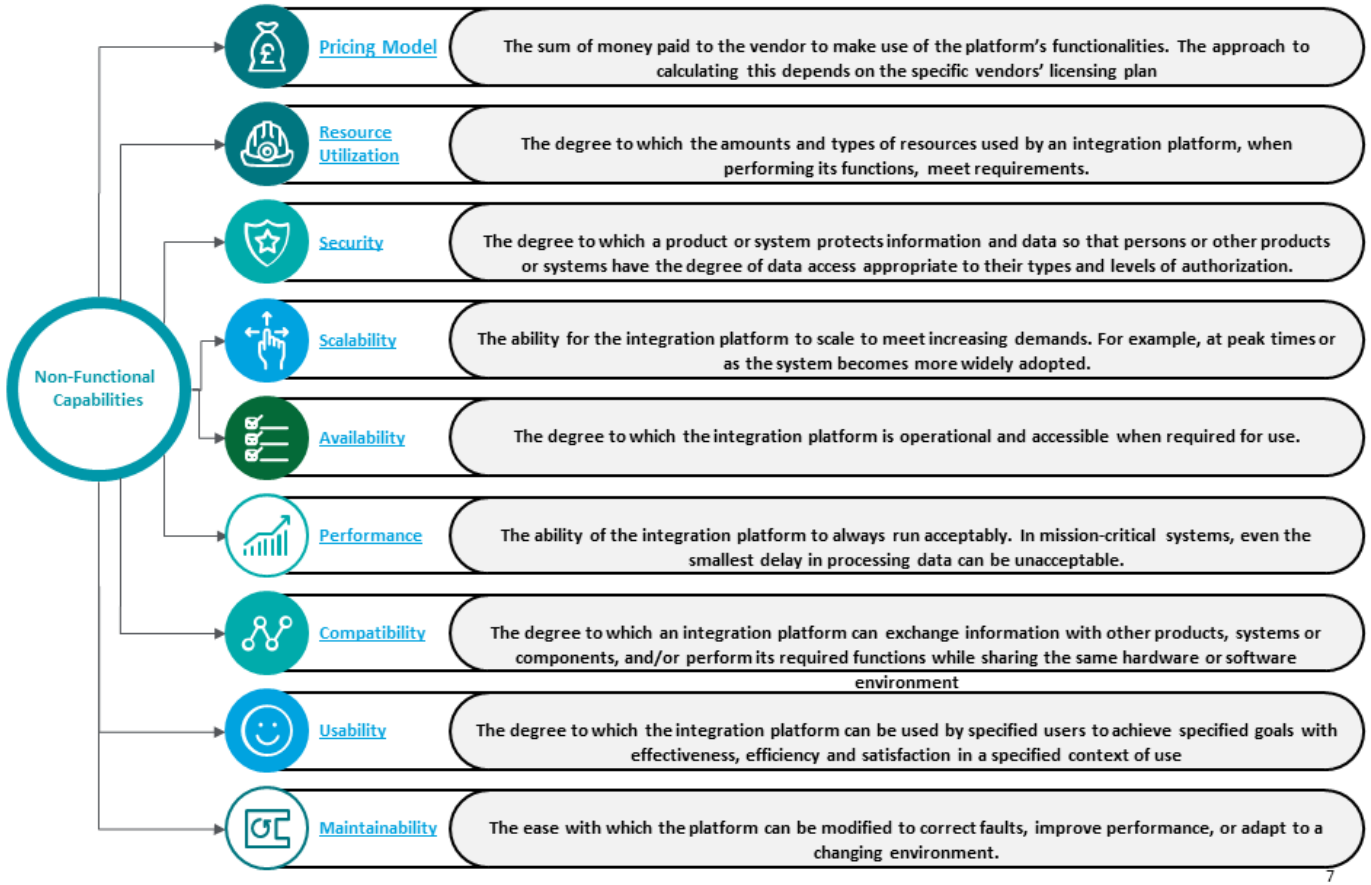
## Comparing TIP and iPaaS

| | TIP | iPaaS |
|---|---|---|
| Availability | • High availability requires redundancy, leading to increased costs and increased management and configuration complexity.<br>• Neglecting redundancy makes TIP a single point of failure.<br>• Detailed recovery planning is required which is resource-intensive. | • Built-in redundancy at multiple cloud infrastructure levels.<br>• Offers options for geographical spread of servers.<br>• Fewer responsibilities for organizations in achieving high availability.<br>• Malfunctions are typically limited to specific services or regions. |
| Performance | • Can theoretically achieve higher performance for internal, synchronous, communication.<br>• The performance of integrations depends on how efficient those integration are developed.<br>• Integration with vendor's other products can enhance performance. | • High performance when applications are hosted in the same data center.<br>• The performance of integrations depends on how efficient those integration are developed.<br>• Service Level Agreements (SLAs) are critical for performance expectations.<br>• Integration with vendor's other products can enhance performance. |
| Compatibility | • Complexity in integrating with existing organization infrastructure.<br>• Compatibility is assessed per platform, not a general TIP vs. iPaaS issue.<br>• Compatibility depends on the specific adapters and development options that are offered within an integration platform. | • Cloud-first approach enhances compatibility with new technologies.<br>• Compatibility is assessed per platform, not a general TIP vs. iPaaS issue.<br>• Compatibility depends on the specific adapters and development options that are offered within an integration platform. |
| Usability | • Setting up a TIP requires many preparations that are time-consuming and difficult.<br>• TIP is often one product which simplifies the approach to developing integration, whereas iPaaS is a combination of different services. If this is desired depends per use case. | • Setting up iPaaS is still complex but less complex than TIP<br>• Increased focus on usability results in faster development of integration by experts<br>• Is less feature rich in the beginning but over time, offers a more divers toolkit and pre-built adapters than TIP. |
| Maintainability | • Requires more responsibility (planning, implementing and testing) and resources from the company for platform updates<br>• End-of-life should be checked per specific vendor<br>• The group of experts willing to work with TIP will become smaller because they want to developed themselves in the direction of the latest-greatest products. | • Has vendor-managed updates, requiring less to no actions from organizations<br>• Is the latest-greatest product of vendors which includes access to the latest features.<br>• Is under continuous development which can result in features that are test and possibly withdrawn from the platform. |

# The non-functional capabilities

## Non-functional capabilities

This research identified 9 non-functional capabilities of integration platforms. The clickable buttons direct you to the information page.





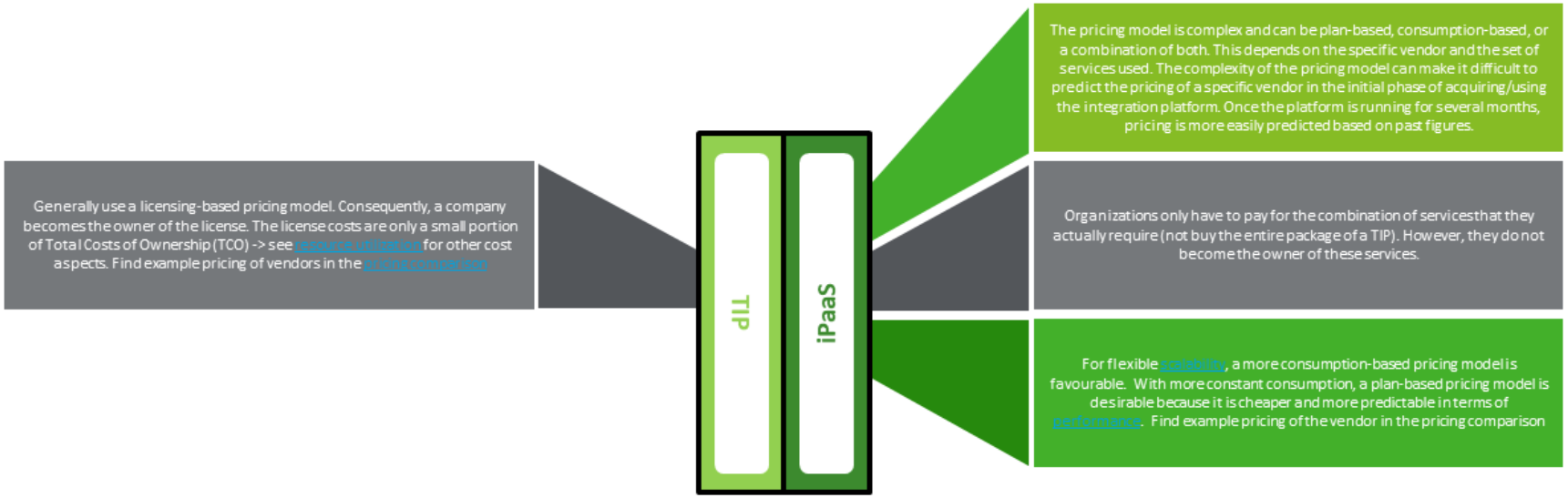

7

# Pricing Model (slide 1 of 2)

## The different pricing structures for integration platforms

**General Information**

Two general types of pricing structures:
**Licensing-based/plan-based:** Pay a fixed, upfront licensing fee based on e.g. the number of cores/processors required in the server(s)
**Consumption-based:** Pay variable costs based on e.g. the amount of traffic or the computing resources that are used.



# Pricing Model (slide 2 of 2)

## Some vendor specific examples

**General Information**

Prices always depend on the vendor. Pricing also reflects the efficiency and scalability of integrations. Inefficient integrations can increase computing resource usage, affecting TIP and iPaaS costs.

BizTalk licenses per core per year:

- 10,853 dollars per core per year
- Minimum of 4 core licenses required per server resulting in minimum costs of 43,412 dollars per server per year

Oracle licenses per processor:

- 57,500 dollars per processor per year

These are the minimum prices to buy the license for these platforms. Be aware that this is for one server only, availability requirements might force a company to have at least two servers up and running -> doubling the price. Furthermore, often companies have multiple environments (dev, acc, prd) which could increase the required computing power and therefore the price.


**Consumption-based:** iPaaS is described in the literature as low start-up costs because it is a pay-as-you-go model. This holds for some iPaaS platforms (Azure and AWS):

- Pay-per-execution: Small amount of money every time a resource (integration flow, API connector, Queue/Topic) is executed.
- Pay-per-hour/month pricing: Pay per month for a certain amount of CPU and memory.

Difficult to estimate the costs beforehand as it highly depends on the number of integrations that are to be developed and the amount of traffic that is expected.

**Plan-based:** Other iPaaS platforms have a considerably large upfront investment: SAP integration suite starts at 60,804 dollars/year, Boomi at 48,000 dollars/year and Mulesoft at 80,000 per year. These are also the base prices and it is unclear how costs increase with increased traffic. Furthermore, the way that these prices are determined differs per vendor. For example, Boomi uses a per-core pricing model whereas Mulesoft handles a fixed price.

## Resource utilization

The table outlines all the different cost aspects of TIP and iPaaS

Organization is responsible

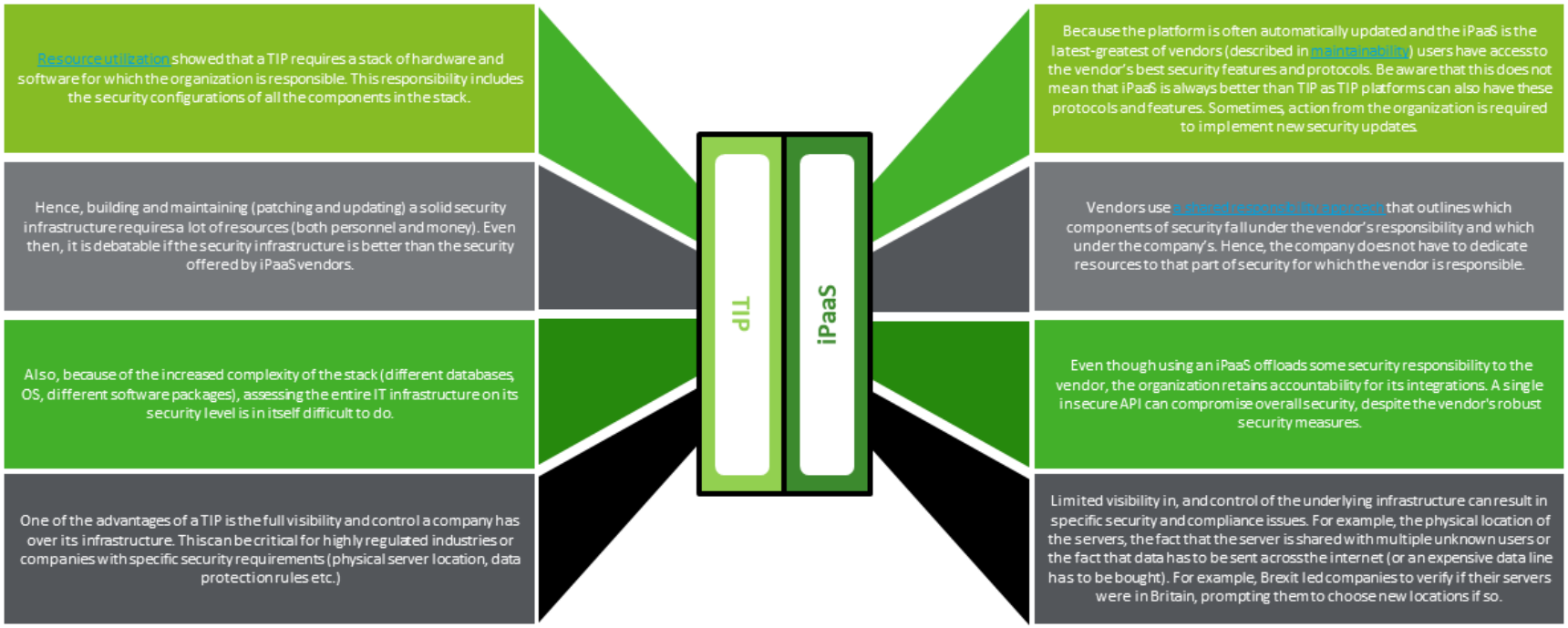
Organization is sometimes responsible

Organization is not responsible

| | | | TIP | iPaaS |
|---|---|---|---|---|
| Major periodic investments | Hardware | Servers (CPU, RAM, Storage) | | Artefacts are bought from the vendor (included in the pricing model) |
| | | Network | | Artefacts are bought from the vendor (included in the pricing model) |
| | | Storage | | Artefacts are bought from the vendor (included in the pricing model) |
| | Software | Integration platform software/License | Price of pricing model | Included in pricing model |
| | | Operating system | | |
| | | Security software | | |
| | | Monitoring and Management tools | Depends on specific TIP platform | Depends on specific iPaaS platform |
| | Implementation costs | Configuration | A lot more than iPaas because the extra required hardware and software needs to be configured. | But less than TIP because only the required software has to be configured. No hardware configurations. |
| | | Internal migration | | |
| | | External expertise | | Perhaps required more because it contains new technology that is still adapting |
| Annual operating expenses | Internal infrastructure IT staff | IT managers, Application managers, Database administrators, Security teams, Netwrok administrators | One person can also assume more roles) | But less than TIP because less infrastructure to manage |
| | Integration team | Program managers, Integration developers, Mappers, User support, Data management teams | Need the roles, amount of people depends on the size/complexity of integration solution. One person can also assume more roles | Need the roles, amount of people depends on the size/complexity of integration solution. One person can also assume more roles |

## Security

Integration security refers to a set of measures and protocols implemented to ensure the secure transmission and processing of data between different platforms, companies, or teams. It makes the data exchanged between your interconnected systems safe and sound.

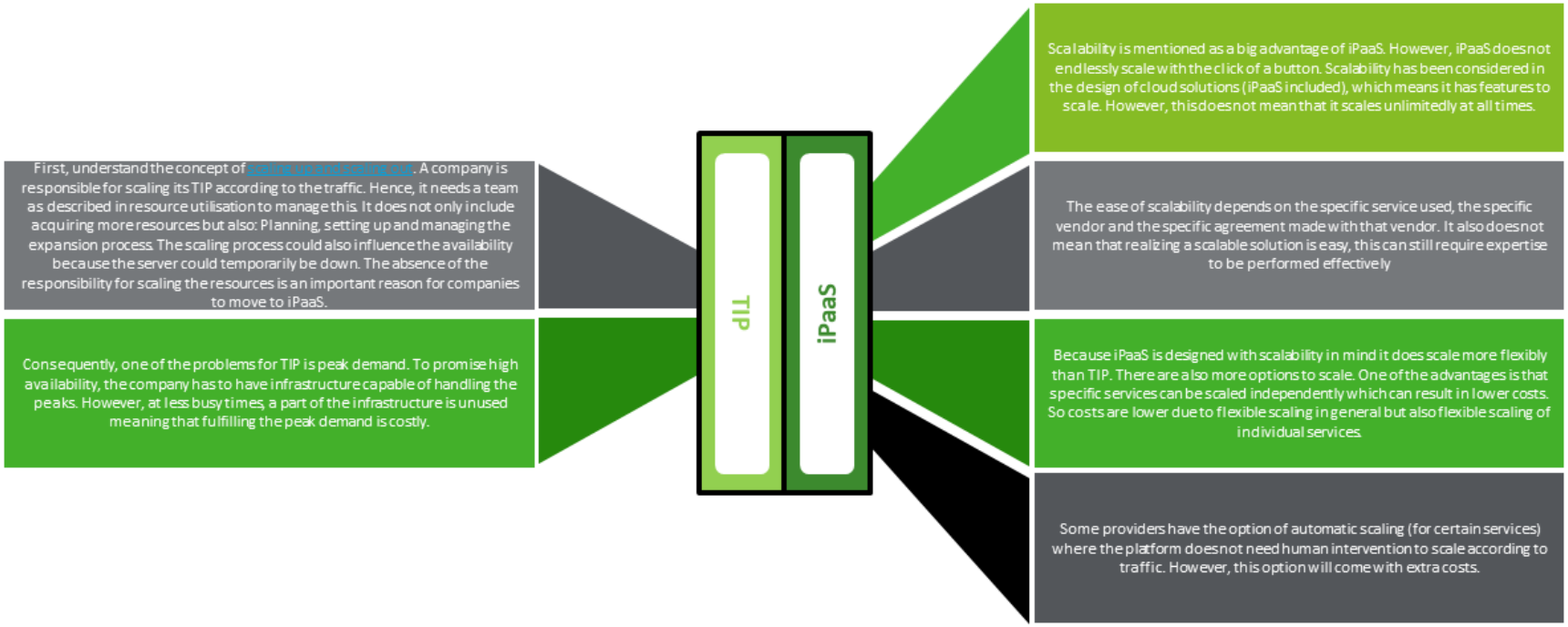


# Scalability

The ability of a system to expand to meet your business needs. You scale a system by adding extra hardware or by upgrading the existing hardware without changing much of the application.

**General Information**

Whether it's TIP or iPaaS, platform expertise is crucial for scalability. The main difference is that TIP also requires infrastructure alignment, needing broader expertise. While both need platform experts, iPaaS simplifies managing scalability components.

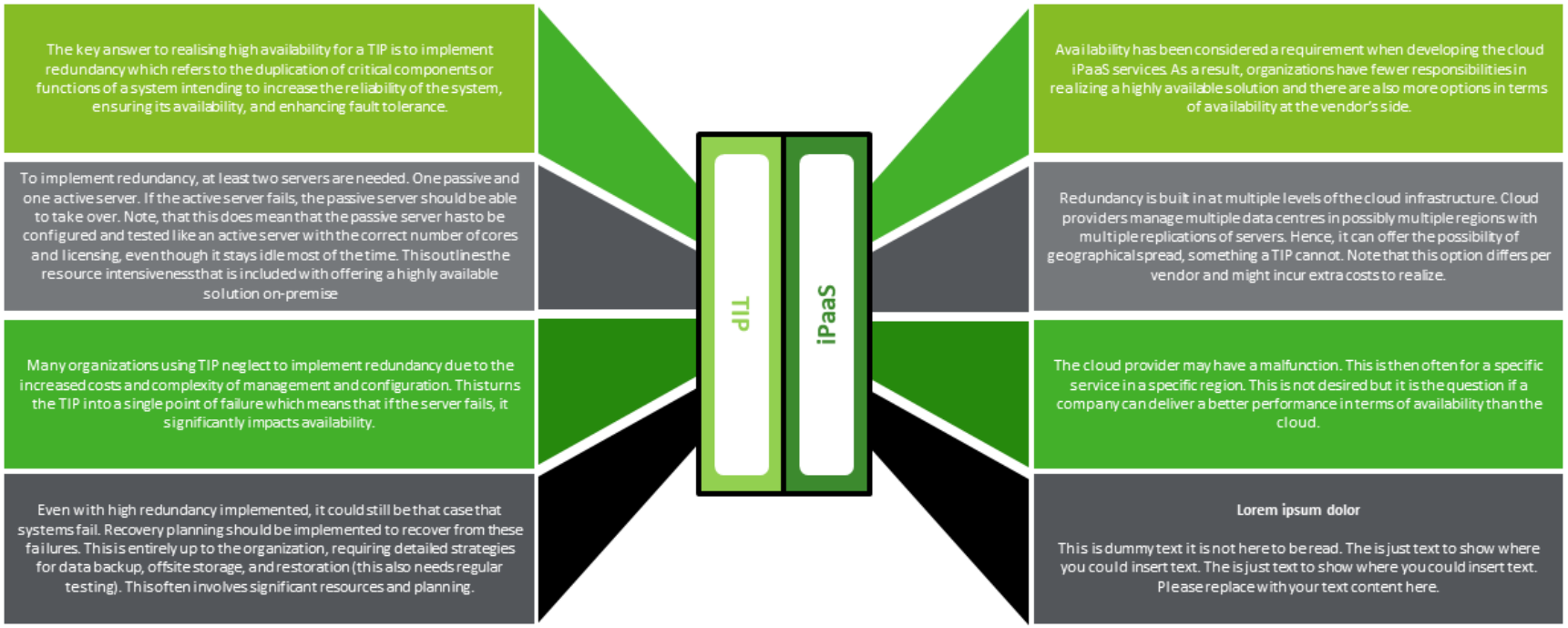


# Availability

Measures the reliability and uptime of the platform and is typically expressed as a percentage of time that the platform is functional and reachable.

**General Information**

Availability relates to scalability. Availability can be impacted through spikes in traffic (peak demand).

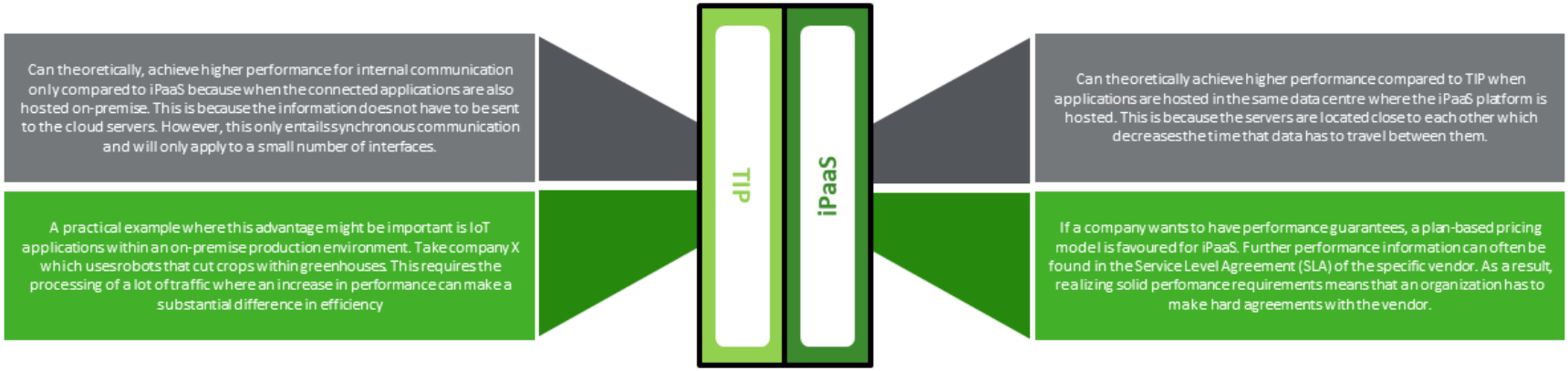


# Performance

The ability of the integration platform to always run acceptably. In mission-critical systems, even the smallest delay (latency) in processing data can be unacceptable.

**General Information**

Two important aspects that play a role in the performance of integrations that do not rely on whether TIP or iPaaS is used. First, the performance of an integration is affected by how efficiently that integration is developed. Unnecessary lines of code will increase latency. Second, many integration platforms are part of a vendor's wider product offering. Often, performance is increased when the integration platform integrates other products of this vendor because tailor-made adapters are used that integrate very efficiently.

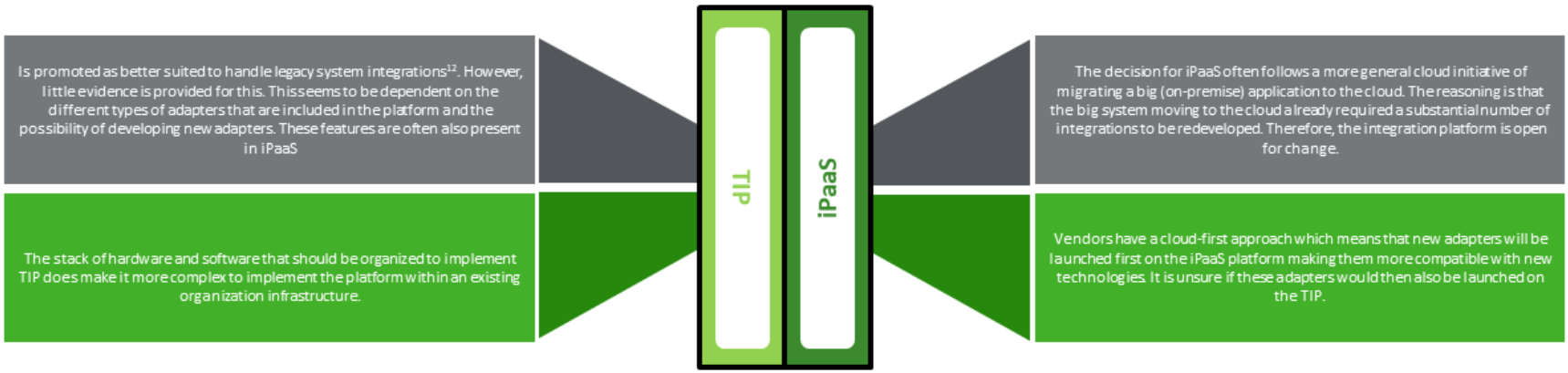


## Compatibility

Refers to the ability of the platform to work seamlessly with various systems, applications, protocols, data formats, and environments without causing conflicts or requiring extensive modifications.

**General Information**

The toolkit of a platform that is, all the specific features that are included to allow a user to develop integrations, largely determines how compatible that integration platform is within a specific application landscape. For example, applications that use a lot of EDIFACT messages are easier to integrate with a platform that supports many versions of this format. Hence, compatibility should be assessed per platform and the decision between TIP or iPaaS does not play a large role in this discussion.




1 https://www.techtarget.com/searchcloudcomputing/definition/iPaaS-integration-platform-as-a-service

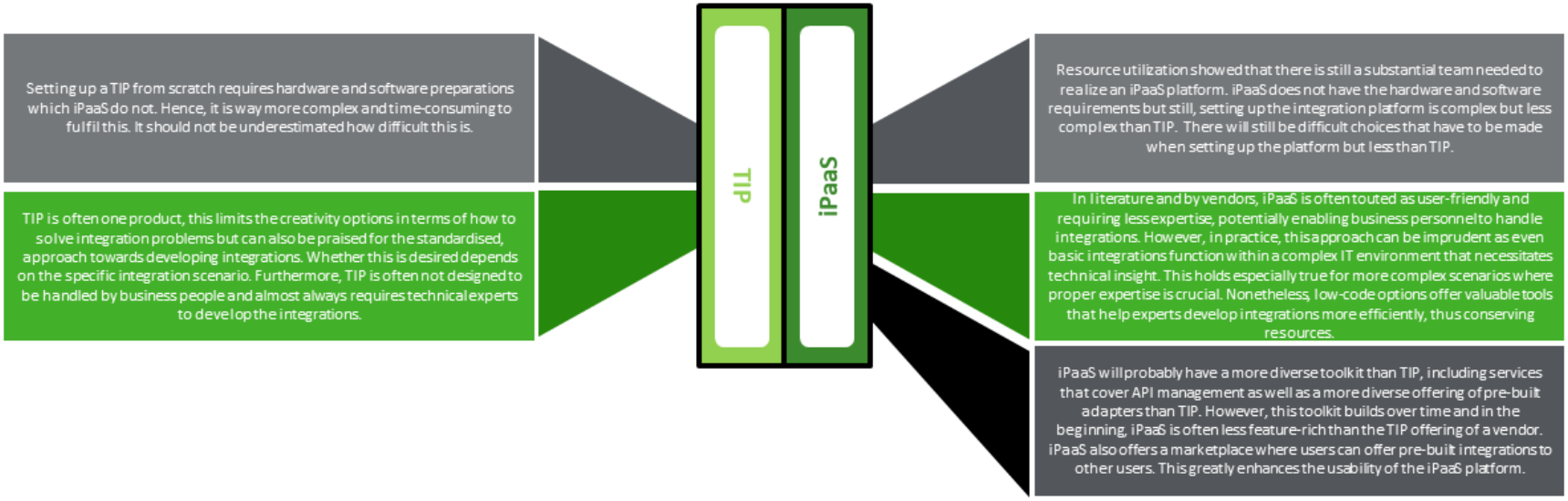
2: Zhang, X., & Yue, W. T. (2020). Integration of on-premises and cloud-based software: the product bundling perspective. Forthcoming in Journal of the Association for Information Systems.

## Usability

The ease with which users can learn to operate, interact with, and achieve their objectives using the platform

**General Information**

There is a trade-off between providing advanced features for highly skilled IT professionals who require the flexibility to develop complex integrations and offering a simplified user experience for less technical users. This goes beyond iPaaS vs TIP but is important to take into account when assessing an integration platform.



## Maintainability

The ease with which the platform can be modified to correct faults, improve performance, or adapt to a changing environment.

# Appendix

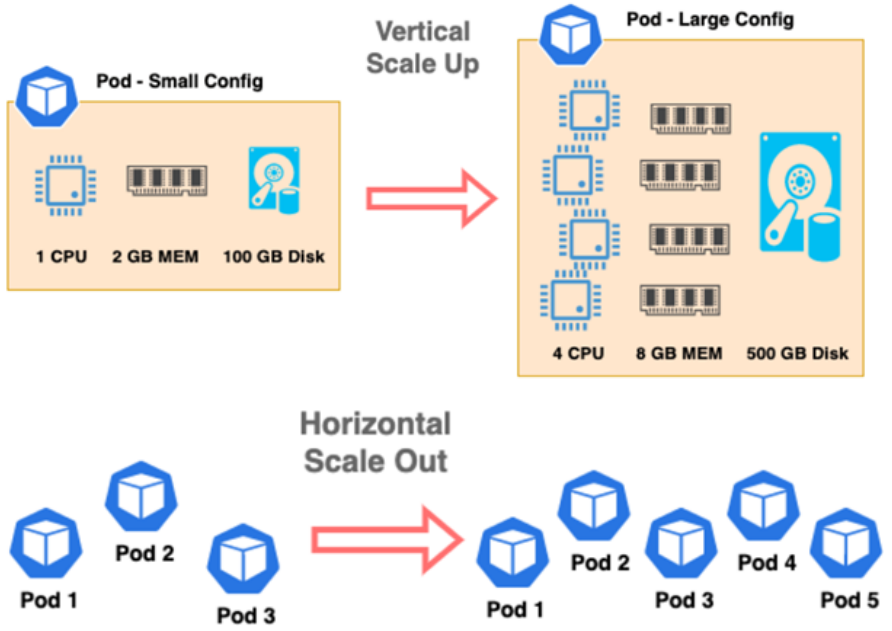
# Scalability

## Scaling-up and scaling-out

To last viewed slide
To capability overview

Scaling-up is adding more resources—like CPU, memory, and disk—to increase more compute power and storage capacity of a single server. The advantage of scaling up is the simplicity compared to scaling out. However, scaling up has limits as one cannot endlessly add computing resources to a server.

Scaling-out: Involves adding more servers or nodes to a pool to work together as a single system. Instead of making a single server more powerful, you distribute the load across multiple servers. The advantage of this option is that scaling can theoretically be endless and scaling back is easy by removing servers. However, it requires the architecting of how to scale out and balance the load.

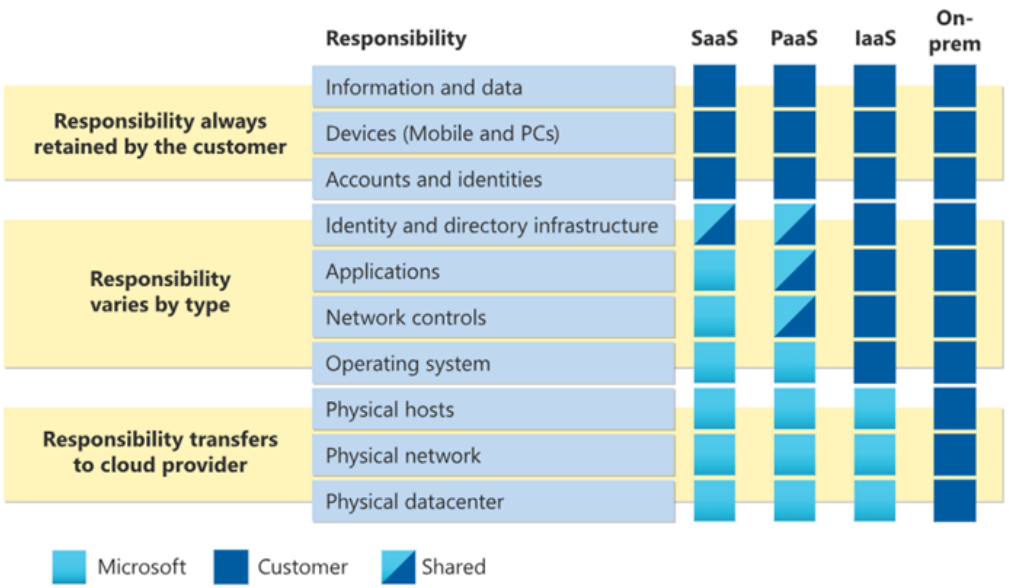


# Security

## Shared responsibility

*To last viewed slide*

*To capability overview*




1: https://exalate.com/blog/integration-security/#:~:text=Integration%20security%20refers%20to%20a,interconnected%20systems%20safe%20and%20sound.

# Definition TIP and iPaaS

## Explanation what this research considered to be a TIP



This PowerPoint defines a **Traditional Integration Platform (TIP)** as: A vendor-provided type of integration platform (can connect across organizations, but does not necessarily have to). A pure traditional integration platform is only when the hosting is on-premise, deployment is on-premise and the management is internally organized.

This PowerPoint defines an **Integration Platform-as-a-Service (iPaaS)** as: A vendor-provided type of integration platform (for on-prem/cloud applications within and across organizations) that is offered through a PaaS cloud service model. Because the PaaS model is used, a pure iPaaS is only when the hosting is off-premise, deployment is in a public
cloud and the management is externally organized.

The coloured words are outlined in the table below:

| Concept | Description |
|---|---|
| Hosting | Refers to the provision of infrastructure and computing resources (like servers, storage, and network capabilities) on which software applications, websites, and services run. Hosting can be performed:<br>- On-premise, when the resources are located physically within the organisation's facilities.<br>- Off-premise, when the resources are physically located at a provided facilities |
| Deployment | The process of distributing and installing a software application or a service on a hosting platform, making it available for use. The deployment options are:<br>- Public cloud: Sharing of the same computing resources (hardware, storage, and network devices) with other organizations or cloud "tenants". access to services and management of an account goes through a web browser.<br>- Private cloud/on-premise: The computing resources are used exclusively by one business or organization. The resources can be physically located at your organization's on-site datacentre (the option is then on-premise or private cloud deployment) or can be located at a third-party service provider (private cloud deployment).<br>- Hybrid: A hybrid cloud is a type of deployment option that combines on-premises infrastructure—or a private cloud—with a public cloud. |
| Management | Refers to the responsibility of technical support, patch management, system management, monitoring and updates. The management options are:<br>- Internal: The organization is responsible<br>- External: A provider is responsible |

## Appendix J Case Study Results

# Case Study Microsoft

Microsoft's TIP offering is BizTalk and its iPaaS is Azure Integration Services. Both are introduced below after which the **Functional Capability Comparison** is described.

## TIP: BizTalk

The BizTalk integration platform launched in 2000[13]. It uses a publish/subscribe architecture that uses adapters to receive and send messages, implements business processes through orchestration, and includes management and tracking of these different parts. Figure 23 shows the interaction between the components. These components are explained below.

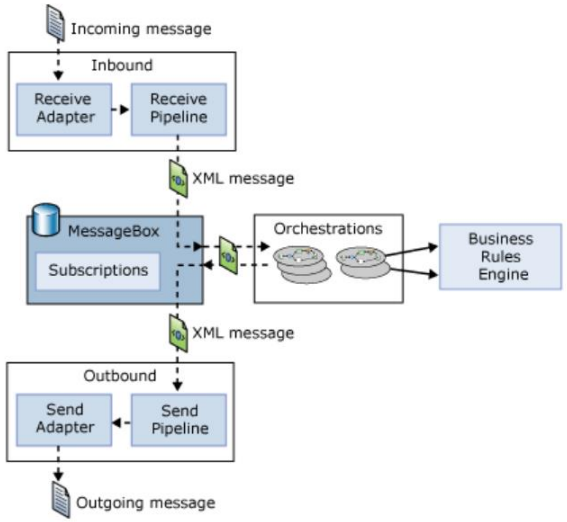



*Figure 23: The main components of BizTalk*[14]

[13] https://news.microsoft.com/2000/12/12/signed-sealed-and-delivered-BizTalk-server-2000-released-to-manufacturing-pricing-and-licensing-detailed/

[14] https://learn.microsoft.com/en-us/biztalk/core/the-biztalk-server-messaging-engine

### Pipelines

Internally, the BizTalk Engine works only with XML messages. Therefore, if needed, a pipeline converts a message from a native format to XML or the other way around in the case of a send pipeline. Furthermore, the receive pipeline adds properties to the messages which are used by the MessageBox to route messages either to the correct orchestration or the send pipeline.

The two main components of BizTalk are shown in the middle of Figure 23: The MessageBox and the Orchestration Engine which work closely together.

### Orchestration Engine[15]

BizTalk is able to define and execute business processes based on applications. This goal is realized with the Orchestration Engine which allows a developer to define the logic of these processes through a graphical tool. This low-code solution is faster, easier to understand and easier to monitor. Creating the logic relies on three primary tools. The BizTalk Editor, the BizTalk mapper and the Orchestration Designer. Firstly, the BizTalk Editor is used to define the message schemas which define the structure and type of a message using the XML Schema Definition language (XSD). Secondly, the BizTalk Mapper is used to define transformations (section 2.3), called a mapping, from one XML message to the other. The Mapper is a graphical tool which can create complex mappings. However, if the boundaries of the low-code tool are reached, code can be used in the form of eXtensible Stylesheet Language (XSLT), a language designed for transforming XML messages. Lastly, the Orchestration Designer is a graphical tool used to define the business logic. Figure 24 shows a flow which uses so-called shapes like “receive”, “send”, “construct” and “transform”. A flow is called an orchestration and each orchestration has a subscription to the MessageBox that indicates the kind of message it wishes to receive. That message functions as a trigger for the orchestration. Often, the last step in the orchestration is to return the (transformed) message to the MessageBox which sends it to the send pipeline.

[15] https://learn.microsoft.com/en-us/biztalk/core/defining-business-processes

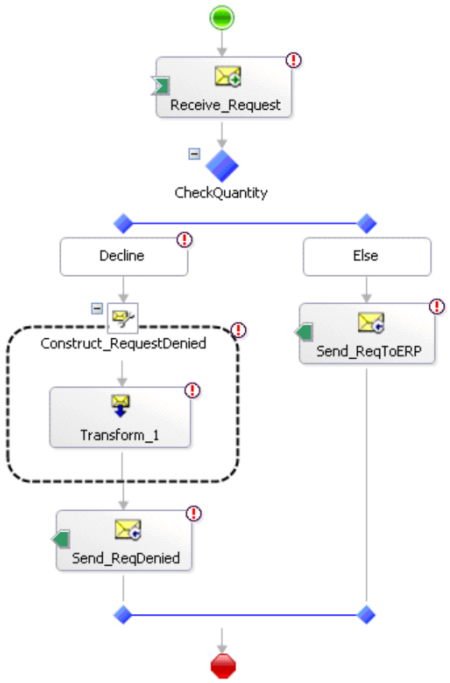



*Figure 24: An example of an integration flow in the orchestration designer*[16]

MessageBox[17]

The MessageBox is the database which is implemented using the Microsoft structured query language server. It functions as the interacting middleman between the receive pipeline, the Orchestrations Engine and the send pipeline. Throughout this process, it stores the different versions of the messages. Upon receiving a message from the receive pipeline, it typically progresses to the Orchestration Engine. Nevertheless, the message can be directly routed to the send pipeline, thereby relegating the BizTalk Server to the role of a messaging system. This routing is performed based on the properties of the message.

## iPaaS: Azure Integration Services (IS)

[16] https://learn.microsoft.com/en-us/biztalk/core/defining-business-processes
[17] https://learn.microsoft.com/en-us/biztalk/core/the-biztalk-server-messaging-engine

The information in this section is summarized from the whitepaper provided by Microsoft on their Azure integration service [59]. Unlike the BizTalk engine, Azure iPaaS is not a fixed package of functionalities. Instead, the concept covers a “set of cloud services for mission-critical enterprise integration”. The main services are:

- **API management**: The API management platform of Azure
- **Logic Apps**: Responsible for the orchestration, adapters and web services
- **Service Bus**: To enable messaging between applications
- **Event Grid**: For event-driven integrations
- **Azure Functions**: To write custom logic tasks

They work together but are sold individually to allow companies to only use the services they need for their integrations. Figure 25 shows an example of how these services are combined in an integration scenario. The individual components of this scenario are explained below.

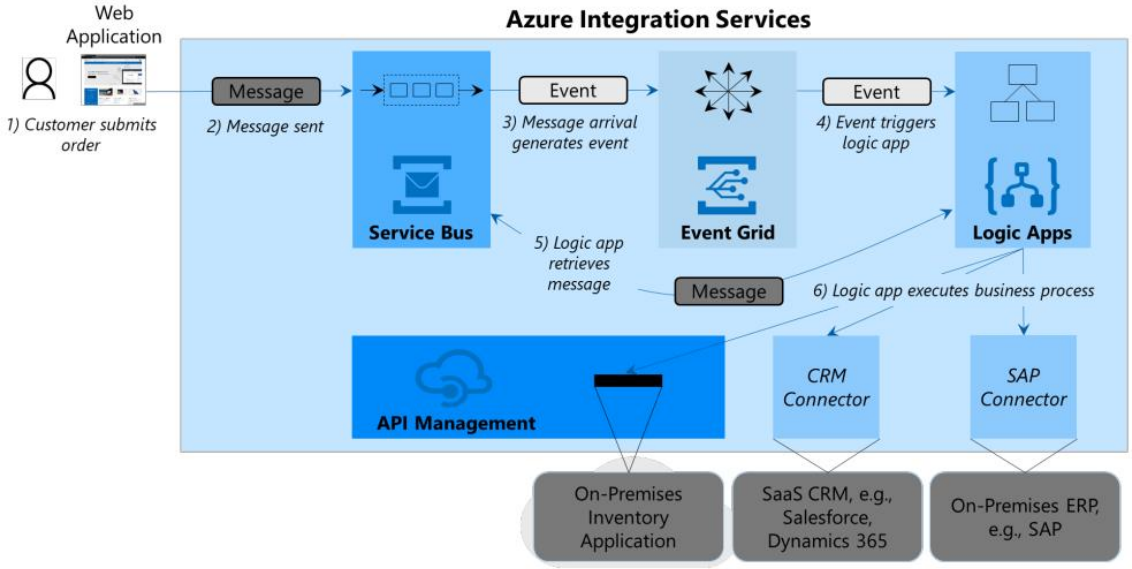



*Figure 25: Overview of an integration scenario using the Azure integration services[59]*

### Logic Apps

Logic Apps are at the heart of Azure's Integration Services as they provide a graphical tool that uses workflow technology to implement integration flows. This process can be system-to-system or user-to-system and is designed using a visual tool shown in Figure 26. The service also contains more than 200 connectors providing straightforward ways to interact with specific external services like Office 365, SAP, Oracle, and Sharepoint.

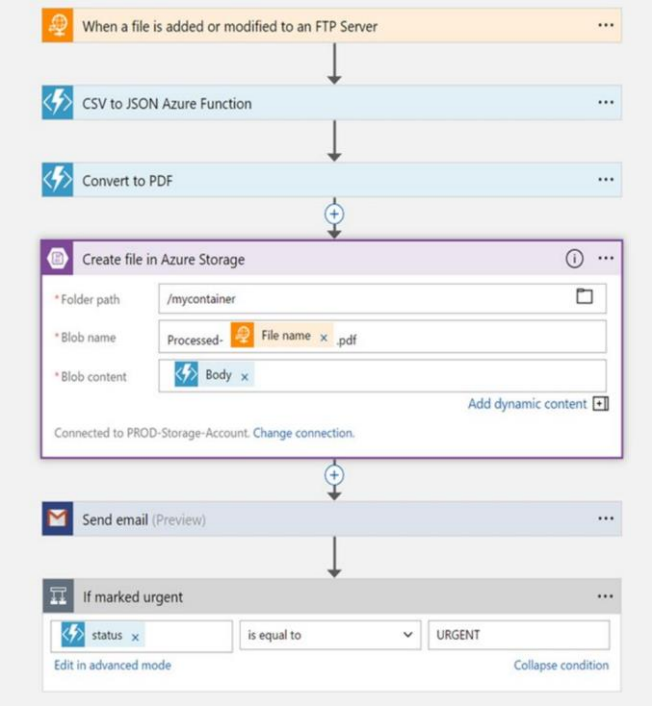



*Figure 26: An overview of a logic app [59]*

The logic app can include APIs exposed through the API management platform, Azure functions to execute code for complex transformations and send and receive messages from the Azure service bus. Furthermore, it can create mappings between messages using the data mapper. A logic app itself can be exposed as an API to be invoked by other services (or logic apps).

### API Management

The Azure API management platform provides the functionalities described in the API Management Platform section. To briefly summarize, the platform provides a layer to expose APIs from backend services such that other (internal and external) developers can consume these APIs securely and reliably to build new applications/processes. Note that section 2.4 explains that an API platform can deliver synchronous communication.

### Service Bus

Where API calls are a synchronous style of communication, the Service Bus allows for asynchronous communication. Despite the name, it is not an actual Enterprise Service Bus as it does not contain many of the features that should be included in an ESB. It should rather be viewed as a messaging system that provides queues and topics which are both types of channels (sections 2.3). A queue can store messages until they are being requested by the consumer of the queue, this is described as a point-to-point channel in the book of Hophe and Woolf (2004) [37]. A topic is a publish/subscribe channel to which message consumers can describe messages based on certain properties of that message.

### Event grid

The event grid has a similar function as the topic in the Service Bus. An event can be created to which consumers can subscribe. If the event is triggered, the consumers will receive whatever is stated in the event. However, the Service Bus is a heavier-weight messaging component whereas the event grid is simpler and faster. The benefits of using the event grid compared to the Service Bus are the speed and the scalability of this service. The event grid delivers 99% of the events in less than a second and it can support 10,000,000 events per second.

### Azure functions

The components introduced till now are all low-code solutions. Some scenarios require custom logic through code scripts. Examples are complex mappings and the inclusion of metadata. To enable this, Azure functions are used which allow developers to create scripts in a variety of languages. These scripts can be executed through a variety of triggers including a time trigger, HTTP trigger or via the queues/topics in the Service Bus.

## Functional Capability Comparison

**API management**

Microsoft does not have a clear on-premise tool to handle API management. Therefore, only the API management capabilities of Azure IS are discussed.

**Authentication and authorisation:** Azure IS functions on an Internet basis and is developed according to the latest technologies. Therefore, a tool like Azure API management originated. This tool has extensive possibilities for authentication and authorisation based on several well-established authentication methods. With API management, it is easy to create an external portal where business partners can register themselves and arrange their authentication.

**Lifecycle management:** The Azure IS API management platform supports multiple versions of an API and has tooling to transform and redirect incoming traffic. This enables flexibility in the development of backend APIs

**Orchestration and routing**: Because API management is a layer of indirection, traffic can be rerouted if systems are changed or replaced.

Azure API management platform allows developers to create and maintain APIs directly in the portal. Even though the participant personally thinks that API development should be seen as normal software development, artefacts should be tested before being brought into production.

**API documentation**: Azure's API management supports the documenting and publishing of the APIs.

**Analytics:** Log analytics and applications insights support monitoring the activity of integrations throughout the platform (not only API management), this way it is possible to follow the complete flow of integration and to produce extensive statistics on the use of these integrations.

**Traffic management**: Azure API management provides features to control the traffic of specific APIs. For instance, the frequency with which a specific business partner can use an API.

### B2B integration:

**Partner Management Interface**: In BizTalk, partner management was included in the platform. Partner-specific configuration could be done for several different communication protocols.

This can be done in API management for general traffic configuration which is something not included in BizTalk. Furthermore, for specific communication, Azure IS has other services to support this. For instance, an integration account can be used to do partner management to support EDIFACT communication.

**Trading partner agreement:** It is not possible to make a general comparison between Azure and BizTalk. The partner agreement is tied to a context/protocol given a certain context/protocol, a comparison is possible.

**Certificate management:** On-premise certificate management is organized on the operating system level so not necessarily included in BizTalk. In Azure IS, most certificate management is done with API management.

### Messaging

**Synchronous communication:** Within BizTalk, all incoming messages are published in the MessageBox which is, in the end, a database. To support the synchronous pattern, only one subscriber can pick up the message and call a backend service. The answer has to follow the same route the other way around. So both request and response are persisted in the database and picked up by a different process. This has negative consequences for the latency and makes BizTalk unsuitable for high-demand traffic.

Within Azure IS, synchronous communication can be configured with API management and for complex scenarios logic apps can be used.

**Asynchronous communication:** For asynchronous communication, BizTalk makes use of the MessageBox which also ensures the **persistence** of messages. All messages are decorated with properties inserted in a central database. Subscribers can, in theory, subscribe to all messages.

Azure IS uses a ServiceBus with queues and topics which also ensures the **persistence** of messages. Messages are published on a queue or topic (see section 2.3). The main difference with BizTalk is that there is only one subscriber to traffic for a specific queue or topic. Again, this is not seen as a disadvantage. Microsoft deliberately left the idea of a central MessageBox.

**Transformation:** In BizTalk, transformation is done with the BizTalk mapper. It supports XML-to-XML transformations and has a graphical tool to develop the transformation (mapping). In real, the BizTalk mapper is an easy tool to create XSLT (a transformation language for XML). For complex scenarios, it is possible to write XSLT directly or to create functionality in .NET

In Azure IS, message transformations can be done with liquid templates. It is far less developer friendly compared to the BizTalk mapper but it supports both JSON and XML format. Recently, Microsoft introduced the data mapper for logic apps. This tool also supports transformation for both JSON and XML but this first version is still less powerful than the BizTalk mapper.

Because message transformation can be an important part of integration development, this comparison is an important difference to consider.

**Orchestration**: BizTalk has a BizTalk orchestration engine (integration flow designer tool) whereas Azure IS uses logic apps. Microsoft has invested heavily in the development of logic apps. It is seen as the low-code development tool for Azure (integration flow designer tool). It can already be seen as far superior to BizTalk orchestration.

**Protocol bridging:** No specific differences other than the differences described above.

## File transfer

BizTalk, each file has to go through the MessageBox. For large files, this can be a problem. In Azure, file transfer can be implemented with a logic app.

Data management: In BizTalk, data management is done through SSIS, part of the structured query language server. In Azure, Azure Data Factory is a service that can be used for data management.

## Other capabilities

**Prepackaged integration content**: The current BizTalk version is based on the BizTalk version 2004. All these years, Microsoft and their partners have created adapters to facilitate easy communication with the product.

Within Azure, the BizTalk adapters could not be used and the integration platform had to start from scratch. At this moment, most adapters and communication protocols which were supported by BizTalk are also supported on Azure IS and because of the open platform, Azure also has adapters to communicate with services which are not available for BizTalk.

**Activity monitoring and logging:** BizTalk has Business Activity Monitoring (BAM), this is a relatively complex solution and therefore often not used. For activity monitoring and logging, most of the time, third-party software is used.

In Azure IS, monitoring and logging are done with application insights. This enables monitoring over a complete solution. This is an area still under development within Azure IS. Microsoft recently introduced Business Process Tracking (BPT). This service is expected to be extended in the coming years.

**Message identification and validation:** Within BizTalk message identification and validation is an important part of the architecture. Incoming messages are identified and validated if needed. BizTalk uses a collection of schemas for this purpose.

Within Azure IS, there is no similar construction. The BizTalk MessageBox is a central point where all messages pass through. This central concept is not used in Azure. Message identification is done in several places. For instance, when a logic app receives messages from SAP, the adapter supports the identification of message types. Even though Azure lacks the central administration of message types, this does not pose a problem in practice. Microsoft

purposely got rid of this central concept because it also had disadvantages. For instance, the central concept made it difficult for integrations not to influence each other.

**Encryption:** Offered through both platforms. No significant difference is noted between Azure and Biztalk.

# Case Study SAP

SAP's TIP is SAP Process Orchestration (PO) and its iPaaS is SAP Business Technology Platform Integration Suite (BTPIS). Both are introduced below after which the **Functional Capability Comparison** is described.

## TIP: Process Orchestration (PO)

The information on SAP PO is summarized from their learning site [23] and documentation site [82]. PO is a package solution combining Business Process Management, Business Rule Management and Process Integration into one solution. Figure 27 shows the main components of PO after which they are briefly introduced.

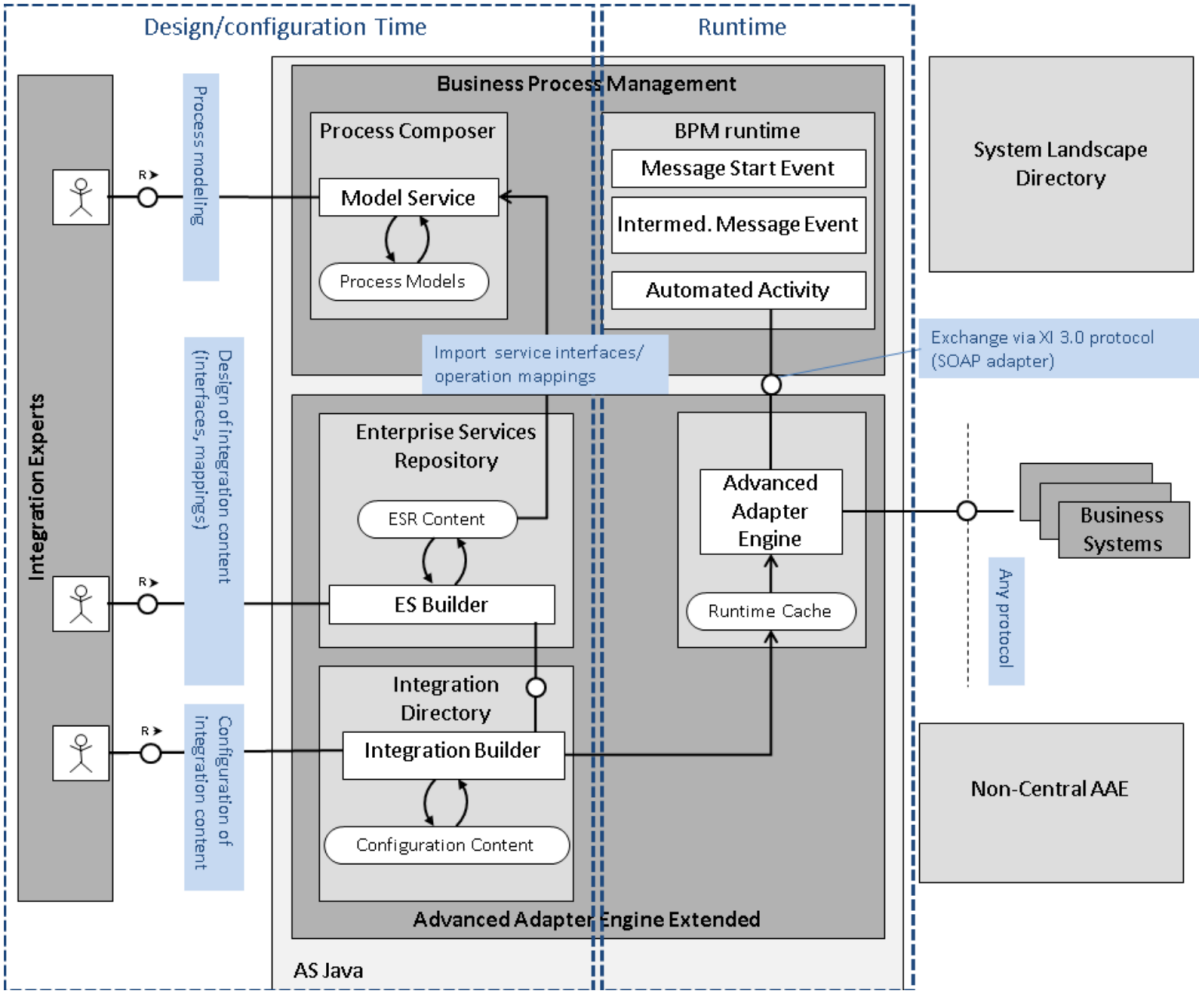



*Figure 27: The main components of SAP Process Orchestration [82]*

As shown in the figure, PO consists of two main components: The Business Process Management (BPM) capabilities and the SAP Advanced Adapter Engine Extended. Furthermore, the figure distinguishes two phases: Design/configuration and Runtime. The components are explained per phase starting with design/configuration.

The BPM capability has the Process Composer that “allows an integration expert to model business processes and to execute and monitor the business processes (BPM-based applications)”. At design time, the **BPM Process Composer** within BPM is used to model the BPM business processes.

**The Enterprise Service Repository** is used by integration experts to develop integration content including data types, service interfaces and mappings. This content can be imported into the **BPM process composer** to make the content available within a **BPM** process. Furthermore, this content is also operationalized in the **Integration Directory** for specific integration scenarios.

**The Integration Directory** is a central tool used to configure BPM processes and integration content to use them in a specific system landscape. It defines communication channels (called integration flows) that specify the connectivity with internal and external systems. This tool also allows for routing, transforming and mediating messages between different systems.

At runtime, the **Advanced Adapter Engine** is used as a runtime engine (message hub) to enable communication between business systems and SAP PO. Therefore, all adapters that are part of the Advanced Adapter Engine can be used. It also involves monitoring and management of processes and integrations via SAP PO's monitoring tools, which ensures monitoring of both integration flows and BPM processes.

The BPM Runtime allows that allows for the definition and management of business rules, which can be integrated into business processes to automate activity. The “internal” communication between the Advanced Adapter Engine and the BPM runtime is based on the XI 3.0 protocol (using a SOAP communication channel).

## iPaaS: Business Technology Platform Integration Suite (BTPIS)

BTPIS is part of the larger SAP Business Technology Platform. The main components are Cloud Integration, API Management, B2B Integration, Event Mesh and the Business Hub. Figure 28 shows a graphical, simplified overview of how these components interact. Each component is explained below using the SAP documentation information [81].

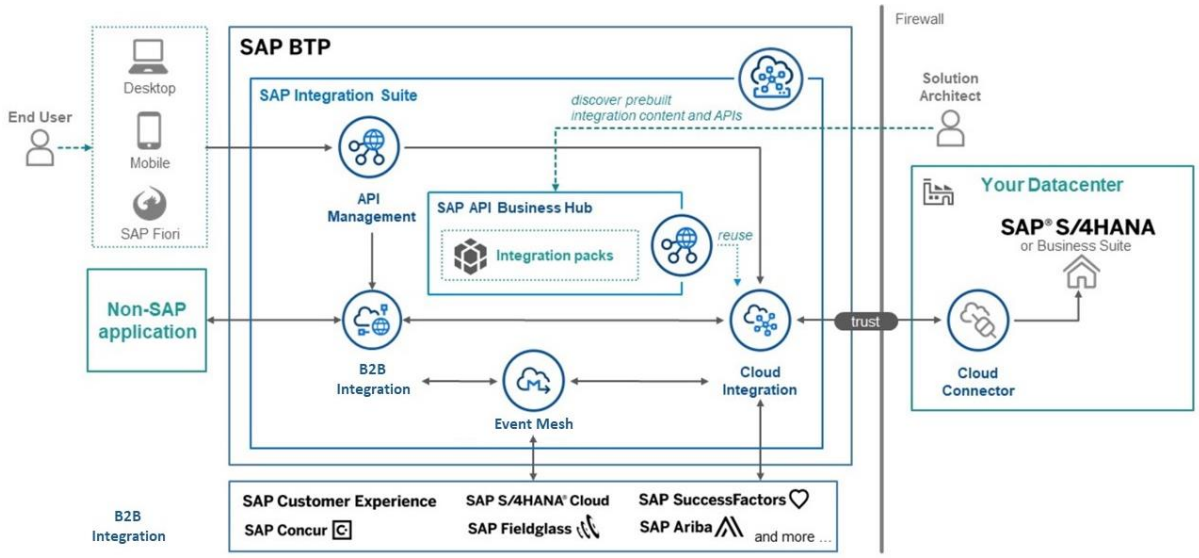



*Figure 28: The main components of the SAP Business Technology Platform Integration Suite [24]*

**Cloud integration:** This is a single end-to-end process integration solution for the execution of internal and external integration scenarios supporting all relevant open standards as well as SAP proprietary connectivity technologies. It provides an integration flow editor which is a low-code, drag-and-drop designer. As shown in Figure 28, it is the central point of communication for all the other components of BTPIS. The integration flow editor allows for the orchestration of these different components and services [81].

**API management**: It "is a complete solution, that addresses all enterprise requirements for API security and governance. It allows users to publish, promote, and oversee APIs in a secure and scalable environment" [81]. It covers all the API management features described in section 2.4.4.

**B2B integration**: It eases the management of B2B relations and speeds up the onboarding process of new partners. It enables secure File Transfer protocols and comprehensive partner management capabilities. Among these is a robust Trading Partner Agreement offering.

**Event Mesh**: a message and event broker supporting the event and messaging technologies for asynchronous and event-based scenarios. It can handle a high volume of data in real time and often triggers events in flows designed in the Cloud integration components [81].

**The SAP Business Accelerator Hub**: A huge number of pre-built integrations, APIs, connectors (for SAP and non-SAP systems), and pre-configured end-to-end business processes are found in the vast SAP Business Accelerator Hub. This abundant resource pool offers ready-to-use components that are easily customizable to match unique business requirements, hence greatly speeding up the implementation of integration projects [81].

# Functional Capability Comparison

### API management

SAP PO does currently not have an on-premise API management platform. Therefore, only the API management capabilities of BTPIS are discussed here.

**Authentication and authorisation:** BTPIS's API management platform offers extensive authentication and authorisation functionalities including the OAuth 2.0. Furthermore, it provides single sign-on capabilities allowing users to log into the different services of SAP with one single ID.

**Lifecycle management:** BTPIS provides extensive offerings to support the lifecycle management of APIs. These include designing API using tools, version management enabling changes in APIs without affecting existing users, and **API documentation** through a portal that functions as a central hub for developers to discover APIs.

**Orchestration and routing**: Since API management acts as an intermediary layer, it enables the redirection of traffic should there be any changes or replacements in the underlying systems.

**Analytics:** BTPIS gives users access to analytics and tools for tracking API performance issues and usage.

**Traffic management**: BTPIS enables the enforcement of security policies and rate limiting to ensure that the APIs are used in line with policies.

### B2B integration:

**Partner Management Interface**: SAP's on-premise offerings have a B2B integration cockpit which is a central application allowing users to monitor its B2B integrations. This is an advanced feature of PO that was successful. The integration cockpit also offers a feature to configure **trading partner agreements** that can be accessed by organizations and their partners.

However, BTPIS has a better alternative than the integration cockpit. It has a specific B2B integration service called Integration Advisor. The tool itself is easier to use than the B2B integration cockpit but both can be used to achieve the same outcome. In this tool, **trading partners' agreements** can be stored and also accessed by organizations and their partners.

**Certificate management**: No specific differences between PO and BTPIS on certificate management. Both have a central repository where certificates are managed and updated. The certificates are used for client authentication.

### Messaging

**Synchronous communication:** In PO, integration flows can be developed within the integration directory. Within an integration flow, you can define a message interface, that connects to the different applications. There you can specify if the integration flow should allow synchronous communication.

In BTPIS, synchronous communication can be done through the API management platform. The API calls can be in REST or SOAP.

**Asynchronous communication:** PO works with the integration directory to create integration flows. Within a flow, it can send messages to a Java Messaging Service Queue, the queuing mechanism of SAP PO. The queue enables asynchronous communication where the target application can retrieve the messages from. The queueing mechanism also realizes **persistence** in PO.

BTPIS is organized based on an event-driven integration architecture. BTPIS has the event mesh which enables applications to communicate with each other through asynchronous events. It allows target systems or integration flows (that are developed using SAP cloud integration) to subscribe to these events. An event can be used to trigger actions without the need for the sender and the receiver to be directly connected in real-time. This does not exist in PO

**Transformation:** In PO there was a very limited way to do transformation. SAP provides standard graphical mapping functionalities and one could also write JAVA scripts (user-defined functions) and import them in PO. Also, XSD mapping was available in PO, so in a way, it satisfied all the mapping requirements that one would encounter in an enterprise. However, the tool was not flexible, especially when mappings became more complex. Then, the tool was difficult to use.

That changed in BTPIS which brings a more powerful mapper. Developers can perform mapping on the incoming messages which is called content modifying. It has a way more powerful mapping option on top of the PO offering that enables it to do JSON-XML and XML-JSON mappings which were not available in PO.

**Protocol bridging:** No specific differences between PO and BTPIS were noticed.

**Integration flow designer tool:** SAP PO does not have a drag-and-drop integration flow designer. The Integration directory does not require the user to write code but the interface uses tables in which developers can configure the integration flows using the integration components developed in the Enterprise Services Repository. **Orchestration** of different services is done in this graphical layout.

BTPIS has cloud integration that uses drag-and-drop features to build the integrations. **Orchestration** is performed in this cloud integration tool. For complex mapping, a script can be written but most use cases can be solved using the drag-and-drop features.

### File transfer

Both PO and BTPIS have File Transfer Protocol (FTP) and Secure Shell File Transfer Protocol adapters. These are used to realize file transfer but no special emphasis is placed by SAP on file transfer in their product offerings.

### Data Management

PO does not have a specific data management feature. However, SAP has an on-premise offering with data management services called SAP BO Data Services which can be used for both **data synchronization** and **bulk data movement** through for example Extract Transfer Load (ETL).

BTPIS has a specific data and analytics group of services including the Datasphere that can be used for **data synchronization** and **bulk data movement**. Unfortunately, the expert had little experience with the data management offerings of SAP so little insights were given as to how these two products compare to each other.

### Other capabilities

**Pre-packaged integration content**: SAP provides a wide variety of pre-built adapters that are all listed on their website. These adapters are available to both PO and BTPIS. These Adapters include File transfer adapters, EDI adapters and partner adapters. The latter are partners of SAP for which a pre-built adapter is available to smoothen integration. All of this is supported by both PO and BTPIS. However, the repository of BTPIS is more frequently updated and some new adapters are not made available in PO.

**Activity monitoring and logging:** in PO, all changes are logged. There is a central interface that shows which developer made changes in the activated version. Monitoring-wise, one of the selling points of PO was the very good monitoring feature. Better than BTPIS currently offers out of the box.

All development is logged. For any integration that is triggered, it is also logged. SAP creates a record of all the changes that have been done that also shows who made the actual change. This logbook can also be retracted from SAP. BTPIS does offer a complete monitoring service called SAP Cloud Application Lifecycle Management to which users can connect their integrations. This service provides full monitoring of all the connected integrations.

**Encryption:** Offered through both platforms. No significant difference is noted between PO and BTPIS.

**Message identification and validation:** In PO, message identification and validation are performed within the mapping tool. This allows the message to be validated against a predefined schemas

In BTPIS, message identification and validation occur mainly at two points. Firstly, in the API management platform where messages can be inspected based on required fields and JSON/XML schemas. Secondly, in the Cloud Integration service which is used to develop integration flows. During the mapping activity, messages can be validated against specific schemas to ensure they conform to the message rules.

## Appendix K Ex-Post Evaluation Hypotheses

This research argues that generally, the decision between TIP and iPaaS is an ‘iPaaS unless’ decision. This means that iPaaS is generally favoured unless some specific reason(s) exist to choose TIP. The following reasons are given for iPaaS’ superiority over TIP:

**Organizations are unburdened:** An iPaaS vendor takes responsibility for the hardware, operating system and security software which unburdens organizations from this responsibility resulting in the following advantages over TIP:

1. **Lower Setup Costs**: Setting up an iPaaS is less costly than TIP because the organization does not have to acquire and configure the hardware, security software and operating system.
2. **Decreased Complexity** iPaaS, being cloud-based, removes the necessity for organizations to buy, install, and upkeep physical hardware and infrastructure. This simplifies resource management.
3. **Vendor-managed updates:** Automatic updates require minimal effort from organizations.
4. **More possibilities:** iPaaS has built-in features and options to realize scalability, availability and performance of integrations.
5. **Potential Cost Savings:** It is not accurate to say that iPaaS is always cheaper than TIP, as costs can vary based on specific integration needs. However, organizations with fluctuating peak demands may find iPaaS more economical. Unlike TIP, where hardware must be scaled for peak loads, often sitting underutilized during quieter periods, iPaaS allows for more flexible scaling, potentially lowering costs.

**Vendors' latest greatest**: iPaaS is a vendor's latest greatest providing:

1. **Access to the latest features/functionalities**: New features are always launched on the iPaaS. They are not necessarily launched on the TIP.
2. **Comprehensive Toolkits:** Similar or a more diverse toolkit than TIP including extensive API management capabilities. This toolkit is also developed with the lessons learned from the TIP platform of the vendor.
3. **Expertise availability:** As integration experts gravitate towards the latest technologies, iPaaS platforms benefit from a larger pool of skilled professionals.

**Alignment with Cloud Strategies:** As organizations increasingly adopt cloud computing to fulfil their IT strategies, iPaaS emerges as a compatible cloud-based integration solution

In a greenfield scenario (scenario 1), this research formulates five reasons for organizations to adopt a TIP over an iPaaS:

1. **Organizations' on-premise IT vision and strategy:** If an organization's strategy and plans are firmly rooted in on-premise solutions, without any inclination towards cloud migration, TIP could align better with their needs.
2. **Reduce the risk of vendor lock-in**: With iPaaS, there's a heightened risk of vendor lock-in, as the vendor is responsible for a larger stack of the platform's operations. Organizations wary of vendor lock-in might lean towards TIP for greater independence.
3. **Specific security requirements**: For organizations with highly specific security requirements, TIP allows for complete control over security settings. This level of control is only achievable by large organizations that can implement superior security measures than those typically provided by iPaaS vendors.
4. **Regulation and compliance issues:** Certain regulations and compliance standards, especially those that restrict data transmission over the Internet or data storage in specific locations, can necessitate the use of TIP over iPaaS.

5. **Control over scalability and performance configurations:** While iPaaS offers advanced features for integration scalability and performance, these are often governed by vendor agreements. TIP offers organizations unrestricted control over these configurations, free from any third-party agreements.

When organizations already have an existing TIP platform (scenario 2). There are two more reasons added to the previous five that make organizations warry of migrating to an iPaaS:

6. **Resource extensive migration process:**
    - **a.** Migrating involves a period in which both TIP and iPaaS have to run beside each other to facilitate undisrupted operations.
    - **b.** Migrating requires redevelopment of existing integrations on the iPaaS platform
    - **c.** Migrating requires new integration experts who can work with the iPaaS. TIP integration experts become less relevant, if not useless. Or they should be trained to use the iPaaS.
7. **The absence of a reason to migrate:** As long as the TIP still functions as an integration platform. It can be difficult to make a business case to invest time and resources into the migration to iPaaS. Often, the reason to reassess the integration platform is when a big application migrates to the cloud. If such a big project remains absent, the urge to migrate is likely to remain absent as well.

The last scenario (scenario 3) refers to organizations that have an iPaaS and consider switching to TIP. This is the least likely scenario given the added benefits of iPaaS and the costly migration process back from iPaaS to TIP. However, there are three reasons why an organization would move back from iPaaS to TIP:

1. **Application infrastructure remains on-premise:** The integration platform was the first to migrate to the cloud with the expectation that other applications would follow. This did not happen and now the on-premise infrastructure is integrated with a cloud solution.
2. **Early adapter of the cloud**: An organization adapted an iPaaS in its early stage. Some vendors made big adjustments to their first version of iPaaS indicating that this first version was not a high-quality product. (this is a somewhat past statement since most vendors have a relatively mature product)
3. **False promise**: iPaaS is actively praised by vendors for its low-code solution that should allow citizen integrators to develop integration. In practice, this is not so straightforward. If organizations migrated from TIP to iPaaS with the idea of developing integration with citizen integrators, they might want to switch back to TIP because they had built up expertise there.