MASTER

A multi-aspect reference architecture for a business process cloud platform

Stoitsev, V.B.

Award date:
2012

Technische Universiteit Eindhoven
University of Technology

Department of Industrial Engineering & Innovation Sciences
Department of Mathematics & Computer Science

A Multi-aspect Reference Architecture for a Business Process Cloud Platform

Vassil Stoitsev

In partial fulfillment of the requirements for the degree of
Master of Science in Business Information Systems

Supervisors: prof. dr. ir. Paul Grefen (TU/e)
dr. ir. Boudewijn van Dongen (TU/e)
Tutor: ir. Erik van der Starre (Capgemini Nederland)

Eindhoven, November 2012

Abstract

The present state of the integration between business process technology and the Cloud is vague and not well-defined. Industry research organizations predict that enterprises will be moving in both these directions in the next few years. This will increase the need for a clear integration between these two areas. Apart from this, many current issues with automated business processes stem from the poor connection between business application systems and the needed business process support, as processes are still hardcoded in application logic. In this study, we define a Business Process Cloud Platform (ProCPlat) and provide a multi-aspect reference architecture for it. We present models in Architecture, Organization and Business aspects. Each of the models is evaluated in a descriptive fashion using scenarios. Additionally, the definition of the platform and the architecture are evaluated using two analytical methods through means of interviews. We believe that our proposal for ProCPlat is a first step towards an easy and manageable platform for business process support in the Cloud.

Acknowledgements

This thesis document is the final result of my graduation project which completes my Master’s degree in Business Information Systems at Eindhoven University of Technology. The research was carried out at the Information Systems group of the Department of Industrial Engineering and Innovation Sciences with the support of Capgemini Nederland B.V. that kindly agreed to help me out in my initiative. Throughout the course of this project I met and collaborated with many professionals from academia and the industry. All of them deserve my appreciation for the help provided.

Firstly, I am thankful to my main thesis supervisor Paul Grefen, who agreed to supervise me as a Master student after long time of working with doctoral students. Also, he agreed to help me realize my own idea which I deeply appreciate. Paul thought me how to put more structure in my thinking which I believe to be something of high value. He always had a good and simple story to explain me how a process should work.

Secondly, I would like to thank my tutor Erik van der Starre who provided me with important feedback and numerous examples from the industry, so that I can better support the designs that I managed to create over the course of this study. After spending more than one year at Capgemini, I know that allocating time for an activity dealing with a student is not the easiest thing to do, and I am grateful that Erik kept his support until the end of the project.

Next, I am thankful to Boudewijn van Dongen for agreeing to participate in the examination committee for my thesis defense. Over the course of my Master program I had the rare chance to be convinced by him multiple times that studying fundamental process modeling principles is important. It took me some time to believe in it though!

I would like to express my gratitude to all the experts who agreed to hear about my idea and give me feedback at different stages of this study: Paul Sijbers, Theo Elzinga, Willem Levelt, Mathieu Diepman, Anton van Weel, Christian Siegers, Ruud Bruls, Hui Li, and Edwin Schouten. Additionally, I am thankful to my manager at Capgemini Wouter van Twillert for giving me the opportunity to conduct my research in collaboration with the company.

During the course of this project I managed to share a lot of lunches with the guys at the Information System group: Heidi, Joel, Zhiqiang, Jana, Egon, Shaya, Shan and Marco. You gave me another perspective on the academia and made me get my daily dose of laughter. Special thanks also to my officemate Rob, who advised me on the way to conduct my evaluation interviews.

Finally, I am thankful to my parents for always giving me freedom and support to pursue my wishes and to my friends for making the seven months of work easier and more entertaining. I hope I have not been a big burden complaining about the writing of this document. Special thanks to Tsetska for reading this document in detail and correcting my grammar, so that it sounds more MIT-like.

Vassil Stoitsev

Eindhoven, November 2012

Management summary

In the past a couple of years the notion of Cloud Computing is being promoted and pushed forward by marketers, even though conceptually it is not something radically new. Together with it process technology has also been steadily gaining more market share as companies realize that it can improve their business. This study intertwines those two concepts and comes up with an innovative design which tackles business problems.

Motivation

Even though business process technology is not present in many software applications, there are business processes which are encapsulated in the logic of the applications. This makes the support of the business processes complicated as applications have to be rebuilt if changes need to be implemented. As more applications get migrated to the Cloud, the built-in processes are also migrated together with them without improving their support. In this study we tackle this issue.

Goal and process

The goal of this study is the design of a reference architecture that supports different activities connected to business processes executed in the Cloud. In order to create the models we go through a research process which is acknowledged by the research community. The process involves a design and evaluation phase that is repeated multiple times as input from literature and industry is acquired. During the repetitions the results of the study are fed back to industry for evaluation. We evaluate our work in two ways: descriptive and analytical. The descriptive approach involves the creation of scenarios using which the platform design is connected with real business issues. The analytical approach involves interviewing industry professionals, which gives us data to perform qualitative analysis. The utilization of such kind of process guarantees the quality of the final product.

Result

The result of this study is a multi-aspect reference architecture for business process in the Cloud with three aspects: Architecture, Organization and Business. Those aspects are part of a three-dimensional thinking framework, in which different architecture models are ordered according to their levels of abstraction and aggregation, and the different realization aspects. Together with those artifacts, we provide a definition for a platform which makes use of the designs: a Business Process Cloud Platform (ProCPlat). We define four different scenarios for the ownership of process definition and process enactment which we use for aligning the proposed platform to the current IT landscape. In the evaluation phase we test the platform design against two scenarios: one connected to the public sector and one connected to the logistics sector. Additionally, the feedback received from the experts during the analytical evaluation method is positive. This implies that the platform design is realistic and it fits the current IT landscape conceptually. Additionally, we come up with points for future development which can make our proposition even stronger and more complete.

Contents

List of figures

1. Introduction

Our introductory chapter starts with setting the context and providing the motivation for this study in *Section 1.1*. In the next three sections we present the problem description, the research objective, and the scope of our research. Then, in *Section 1.4* we describe the methodology that we follow. We conclude this section with an outline for this study.

1.1. Context and motivation

This study is performed with the support of Capgemini, which is a global IT services company that operates in about forty countries around the world and has its headquarters in Paris, France. The Dutch branch Capgemini Nederland B.V. is one of the leading consultancy companies in the fields of Information Technology, Management, and Outsourcing in the Netherlands. The branch employs roughly six thousand people who work closely with the teams in India. The master thesis was designed and developed with the support of the Custom Software Development division which is part of the Application Services sector. The main goal of the division is to design and develop custom solutions in Microsoft .NET, Java, and different Business Process Management Suites.

Nowadays Cloud Computing has been pushed by marketers as one of the most important steps in the IT transformation strategies of medium and large enterprises. It helps companies cut their IT-related costs as it provides cheap IT infrastructure, so that businesses do not have to make capital investments in IT. Cloud Computing utilizes economies of scale, which is the reason why it can offer IT-related services at a low price. Often companies see the value behind the migration of their existing applications to the Cloud, which, unfortunately, does not mean that applications are reengineered and improved to meet the current standards when they are moved to the Cloud. Another concept that is actively promoted on the market is process technology. It is most often advertised with the term Business Process Management (BPM). Even though process technology has been around for about thirty years, applications still encapsulate a lot of business processes. This makes them inflexible and hard to change upon the emergence of new requirements. When the applications get migrated to the Cloud, they carry the problem of encapsulated processes from the on-premise environment to the off-premise one. A possible solution to this problem can be the existence of good business process support from the vendors of Cloud services. This might drive the enterprises in the direction of reengineering their applications during the process of migration to the Cloud and handle the existing business processes in a better way.

1.2. Problem description

Cloud Computing brings in three main service models: Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), and Infrastructure-as-a-Service (IaaS). We elaborate on them later in this study. As applications using PaaS utilize different architectures, our first problem description for this study was that there is only limited process support for PaaS applications. After conducting some initial research, we identified three existing ways to get process support for PaaS applications – integrated workflow engines, on-premise Business Process Management Systems, and Business-Process-as-a-Service (BPaaS) offerings. The later one turns out to be still premature and the number of such offerings on the market is scarce. Additionally, the Cloud vendors do not offer well-integrated process support for PaaS solutions. Based on the last two observations, we decided to redefine our problem description. Our claim is that ***there is a lack of business process support in the Cloud***. For this reason, we decided to work on the extension of the BPaaS concept and make Cloud support for business processes the focal point of our study. We concentrate our efforts in coming up with a better way of handling processes in the Cloud. We already have some BPaaS offerings and BPM-as-a-Service, but this does not cover the whole spectrum of options. Still, we take into consideration what is already on the market and we do not try to propose changes to the existing services as they are already utilized by the business.

1.3. Research goal and objectives

As we already mentioned, there is neither an official standardization document dealing with business processes in the Cloud, nor a reference architecture that provides support for such processes. For this reason, the research goal of this study is ***to design a reference architecture for a Cloud platform that supports various scenarios around the definition, enactment, and integration of business processes***.

As our goal is composite in its nature, we define additionally six objectives that help us in evaluating the success of our final product:

- To make an overview of the research that has already been conducted in the fields of process technology, Cloud Computing, and reference architectures
- To define what a Cloud platform for business processes is and what ownership scenarios are possible
- To design models for the platform in an architectural aspect
- To design models for the platform in an organizational aspect
- To design models for the platform in a business aspect
- To evaluate the models against two practical scenarios and with four experts from Capgemini

Those six objectives are also present in our overall planning for this study, since they serve as milestones in the development process. The first five objectives are directly incorporated as milestones in the planning while the last objective concerning evaluation is spread between the milestones regarding the models. We elaborate on the research methodology in *Section 1.5*.

1.4. Research scope

It is important before proceeding with our study to define what its scope is. Our goal is to develop a generic reference architecture that does not deal with specific technologies on the market. As such, it should be also domain-aspecific, so that it can be applied in various industries. Additionally, we assume that the enterprises that act as parties in the reference architecture are of medium or large size. Such kinds of businesses have clearly defined and formalized processes. This also means that they also heavily rely on large volumes of data, which classifies them as information-intensive.

1.5. Research methodology

In order to obtain better results with this study, we apply a scientific approach that guards the research from inconsistencies and provides different ways to evaluate our findings. We base the methodology of our study on Design Science [1], which offers us a systematic way to approach our research. By applying an established design process, we ensure the quality of the end product by the process of obtaining it. More concretely, we use the very popular Information Systems Research Framework proposed by Hevner et al. [1]. The visual representation of the framework is presented in *Figure 1*. Moreover, the IS Research Framework provides seven Design Science research guidelines that can serve as a way to validate if the research is actually following a Design Science approach.

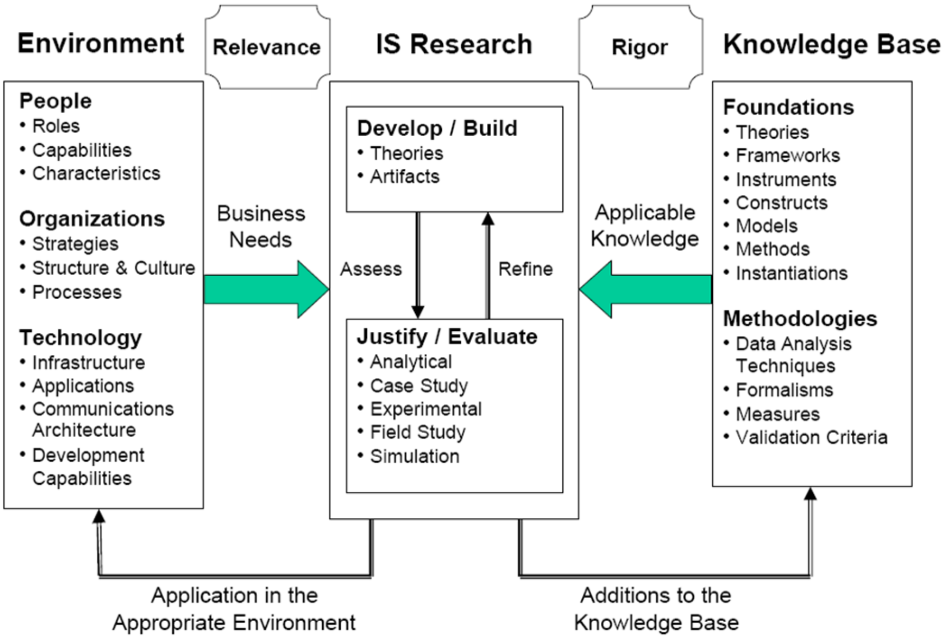

Figure 1: IS Research Framework [1]

Through the course of our study we have numerous transitions between the Environment, the Knowledge Base, and the IS Research. Those are needed in order to design the artifacts related to the study. In our case, we contribute with a new construct and models – a new definition for a Cloud Platform for Business Processes and an underlying reference architecture which supports it. The IS Research framework provides us also with different methods to evaluate our findings. We deal with the design of an innovative artifact new to the field. This limits us to specific methods that we can use for the evaluation, namely one descriptive and two analytical methods.

We define 7 stages which span 23 steps in total that have to be executed in order to reach the final goal of our study. Out of those 7 stages which we can consider as milestones, 5 participate in fulfilling our research objectives, as the first and last one are not directly working towards fulfilling the objectives. For each step we define the way that we are traversing the framework, the deliverable for the step, and the visibility of the deliverable (private, internal, or public). We are providing a table with the complete operationalized project planning for this study in Appendix A.

1.6. Outline

In addition to the complete project planning, in *Figure 2* we provide a flowchart illustrates the order, in which the research is presented in this document. Chapter 1 is naturally concerned with the problem definition and planning. Then, we provide the background information related to this study. In Chapter 3, we proceed with the definition of the platform and the justification for our definition with examples from the industry. In the next three chapters we present the designs concerning the Architecture, the Organization, and the Business aspects. Those models are supported with running scenarios which serve as the first part of our evaluation mechanism. In Chapter 7 we provide the second part of the evaluation where we discuss the conducted interviews with industry professionals. We conclude this study with a chapter for conclusions and future work.

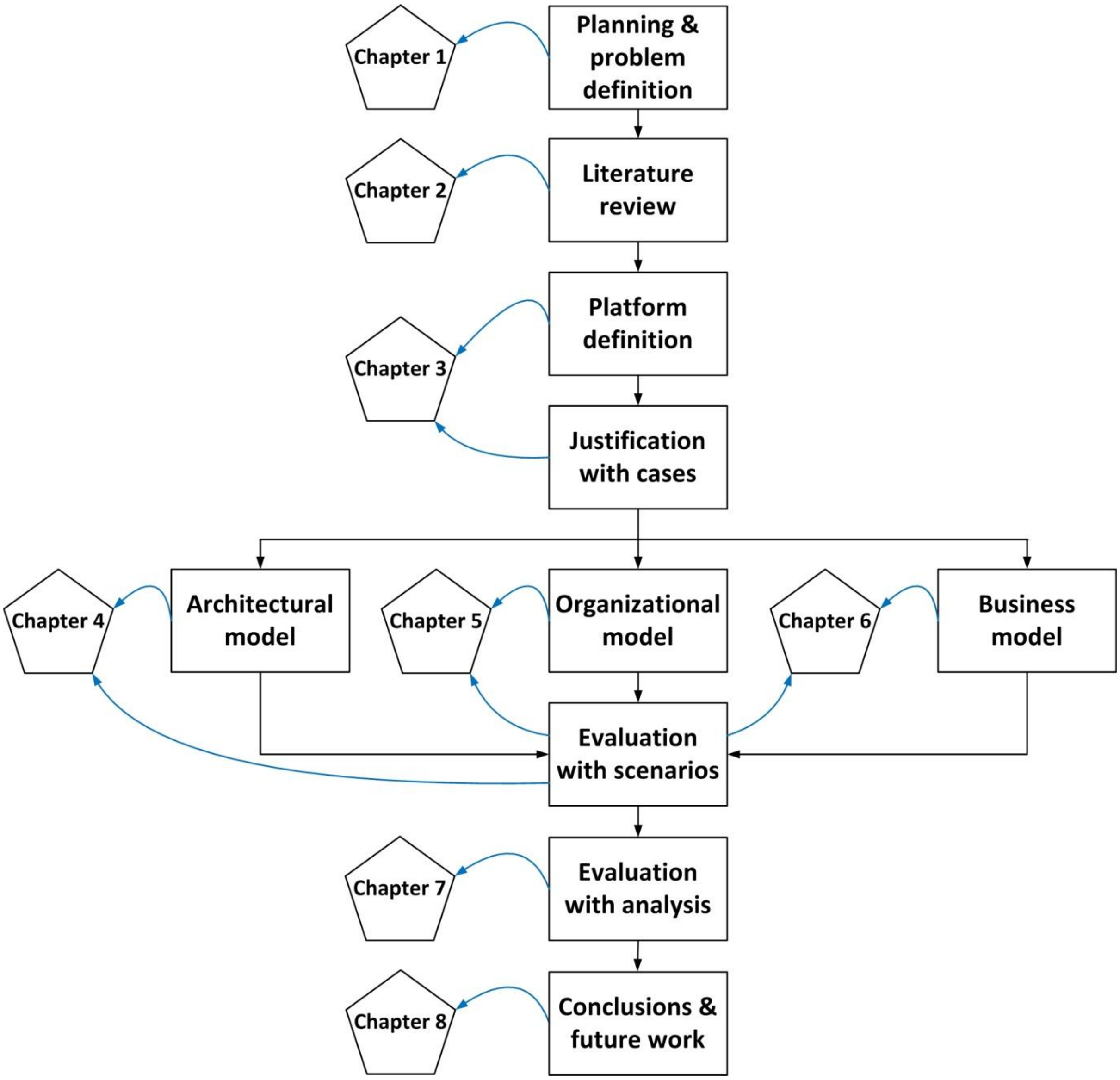

Figure 2: Research Process

2. Background

This chapter presents a summary of all the background information related to process technology, Cloud Computing, and architecture of information systems. We consider the presented information relevant for the later stages of this study, as it serves as a basis for our designs. *Section 2.1* discusses the differences between business processes and workflows, and the supporting software for the two types of entities. In *Section 2.2* we provide an overview of Cloud Computing, and we discuss the underlying service models and two of its important characteristics. Finally, *Section 2.3* presents a summary of the different types of information systems architectures and their application in the field of Cloud Computing.

2.1. Process technology

In this section we discuss the difference between business processes and workflows. We provide a terminology mapping which helps for the understanding of the differences between the two concepts under discussion.

Business processes

One of the most abstract entities in process technology is a business process. It has been well-defined over the years in different publications. Most authors have based their studies on the definition of Davenport and Short [2], which dates back to 1990 and is a cornerstone for process technology in general. According to them, a business process is “a set of logically related tasks performed to achieve a defined business outcome.” The term is defined in similar fashion by the Workflow Management Coalition (WfMC) in 1999 [3].

Workflows

Next, the term workflow needs to be defined. It is common that authors interchange this term with business process. The WfMC defines it very accurately as “the automation of a business process” [3]. It is rather clear that the term workflow is more technology-oriented compared to the term business process. One study even adds upon the technology factor stating that “a workflow comprises cases, resources, and triggers that relate to a particular process” [4]. Cases, resources and triggers stem from the organization and the functioning of an enterprise, and relate to the implementation of process technology. Even more evidence for the difference between the two terms under discussion is the fact that a business process can be complicated enough, so that it involves multiple workflows, which might even be of heterogeneous nature, i.e. being processed by different systems. We visualize the order of exploration of the terminology concerning process technology in *Figure 3*. We differentiate between two conceptual and two technical topics as each of the conceptual topics has a counterpart on the other side of the dashed line.

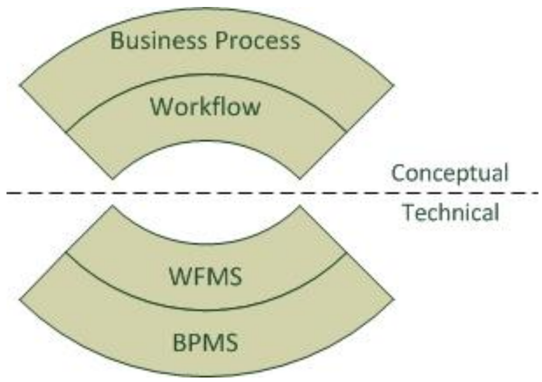

Figure 3: Terminology mapping

Workflow Management Systems (WFMS)

First, it is wise to mention that workflow management does not require a system in all cases. For years, people have executed workflows using their hands and without relying on a smart technology to help them [5]. Actually, a lot of enterprises still rely only on their know-how to manage the execution of workflows manually and WFMSs come to help larger enterprises where the correct management of workflows generates business value. It can be observed that there is a bit of discrepancy on the definition of WFMS. One study defines the terms as a general software package for managing business processes [4]. In this case, the terms workflow and business process are used interchangeably. The WfMC defines WFMS in a more complicated way [3], but it does not mention that WFMS is a general purpose technology. As such it should be able to function properly regardless of the business domain, the type of information flowing and the people involved in the execution of a workflow. The actual WFMS is not performing any work by itself but it is just orchestrating who or which application should process the data and then deliver the result to the next party in the workflow.

Creating an industry-wide abstraction for WFMS is a task that was already handled by the WfMC in 1995. The coalition came with a reference model which defines what basic features a WFMS should incorporate [6]. In total five additional components connected with interfaces to the workflow engine are proposed as we can see in *Figure 4*.

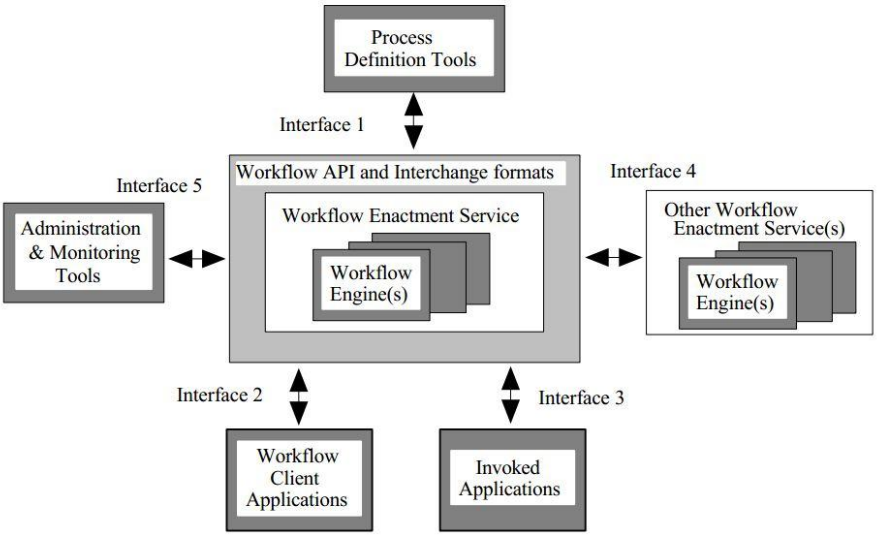

Figure 4: WfMC reference model

The WfMC defines a Workflow API and Interchange formats which are important as they define the way the workflow engine is accessed and fed with data. Interesting for our further research is to find out how this reference model best integrates with the architecture of an application. In addition to the WfMC reference model, which is very generic, there are other models as the Mercurius architecture [7], which are more detailed. The Mercurius architecture even provides a second layer of models where some of the components from the first layer are additionally detailed. We use both the WfMC reference model and the Mercurius architecture as inspiration in the design phase of this study.

An extension to typical WFMS's is Case Management systems, or also known as Case Handling systems. Those systems have some significant differences compared to WFMS that enable the knowledge workers to have a better view on what happens with the case data. Case handling provides all information available, enables process activities on the basis of data, separates authorization and distribution, and allows for data modification even after some activities have been executed [8]. In short, it provides a lot more flexibility compared to WFMS, as the case data is the primary driver and actions like *Skip* and *Redo* are enabled [8].

Business Process Management Systems (BPMS)

Exploring the last sector of *Figure 3* brings us to Business Process Management Systems (BPMS). As a higher abstraction layer, BPMS encompass WFMS. Nowadays businesses are talking on a daily basis about BPMS, which are also named Business Process Management Suites. This is a natural evolution as the development of technology is enabling the support of more complicated systems. One study looks into BPMS in 1995, [9] at which point it was hard to make a distinction between BPMS and WFMS. The author claims that WFMS at this time were "the first Generation of BPMS" and he was right as nowadays the term WFMS is not popular. Another study claims that at a certain point of the development of technology many vendor companies just rebranded their WFMS into BPMS [10]. What BPMS are actually adding on top of the existing technology is more sophisticated monitoring and diagnostic activities. Another goal of BPMS is to enable the improvement of the business activities by integrating heterogeneous application environments. Nowadays, enterprises would like to directly connect to their business partners (those can be part of the same company) and to do this in a seamless fashion. This integration is interesting to be investigated further together with finding the best scenario to achieve it.

2.2. Cloud Computing

In this section we provide an overview of what Cloud Computing is and what are underlying service models for it. Further, we explore two of the characteristics of Cloud Computing which we consider important – elasticity and resource pooling.

Cloud Computing definition

Many scholars have tried to provide a definition for Cloud Computing and to describe its scope. All [11], [12] and [13], which were published in the period 2008-2009, provide good definitions for Cloud Computing. Nevertheless, in this study we use the definition recently provided by NIST [14], which is also acknowledged and used by the Open Group [15]:

> *"Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage,*

> *applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."*

This definition is generic and does not go either in too many technical details, or in explanations about business relevance. In addition to the definition, NIST provides five important characteristics (on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service), three service models (Software-as-a-Service, Platform-as-a-Service, and Infrastructure-as-a-Service), and four deployment models (Private cloud, Community cloud, Public cloud, and Hybrid cloud). Of those, the service models and two of the characteristics are interesting to be investigated, thus we explore them in detail in the following sections.

Reasons for emergence

The emergence of Cloud Computing was not random in any case and has its pure business reasons. The big vendors like Amazon, Microsoft, and Google invested into the Cloud because they saw the business value. In the industry there is always a comparison made between capital expenditures and operational expenditures. This demonstrates how traditional environments require stepwise provisioning of computational power that takes time and financial resources. On the other hand the Cloud can always follow the load of the deployed system by the customer and provide exactly what is needed. This makes the environment efficient and does not waste energy to run unused computational power. Another important business reason for the emergence of the Cloud is that companies do not have any upfront costs in order to start a new business or a new service. This inevitably creates potential users of the Cloud Computing offerings as enterprises are willing to cut on their constantly rising costs for Information Technology. This also means that businesses can focus on their core competences instead of dealing with menial tasks like provisioning servers, and investing money in upgrades of hardware and software.

Everything-as-a-Service (XaaS)

As we already mentioned, some stress should be put on the service models that come together with the concept of Cloud Computing. NIST recognizes three service models [14]: Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), and Infrastructure-as-a-Service (IaaS). Those three are also the most cited ones at different technical conferences, but in addition to them there are numerous other models defined by different scholars. The fourth most important one that we are going to investigate in depth later in this study is Business-Process-as-a-Service (BPaaS). Recently it is generating more hype as it is slowly emerging on the market for services. Moreover there are numerous others offering which can be marketed as a service. Commonly they are referred to as Everything-as-a-Service (XaaS) [16]. Other examples of XaaS are Communication-as-a-Service, Monitoring-as-a-Service, Desktop-as-a-Service and even Human-as-a-Service [16].

The most common type of service is definitely Software-as-a-Service. It has been around for about twenty years already with the emergence of the Web in the early 1990's. This is the easiest model for a company to provide its customers with some productivity such as e-mail service or file management, and it is quite common concept. For this reason, it would be more interesting to discuss Platform-as-a-Service. According to a research made by Gartner [17], PaaS will be gaining more power over the years until 2015. Not only big companies will be entering the PaaS market but also small ones that provide developers with innovative platforms to build applications for. Most of the big players in the market already have some kind of offering around PaaS, as Microsoft appears to be the leader in it with almost a complete development platform providing elaborate mechanisms for integration and security. PaaS is also interesting because it enables the companies to focus their resources on building smartly-architected solutions at the level of conceptual software architecture instead of planning the provisioning of servers or the configuration of out-of-the-box online software. People like to be unique and building custom software, which is integrated with different platforms, will always stay in the software business.

Elasticity

The most often mentioned feature of the Cloud is undoubtedly its characteristic of rapid elasticity. This refers to the ability of the system to scale out horizontally and provision more nodes which are mirrored images of the one(s) already running. This concept is often confused with scaling up vertically – an action that involves the increase of computational power of a single node. In order for the scaling to happen in the most rational way, the architecture must be designed in a way so that the coupling is as loose as possible [18]. This means that each single piece of functionality can be scaled out individually. Having separate components in the different layers of an application architecture implies that those components must communicate asynchronously between each other in order to achieve independence from the rest of the components [19]. The asynchronous communication must be guaranteed by a durable information exchange mechanism such as message queues. The queues are fed with information from one component which is later consumed by another component depending on its availability. In this way the components do not have to wait for the completion of a task by another component. Decoupling of traditional applications also enables scenarios involving hybrid Cloud solutions, in which some of the functionality is still kept in the on-premise datacenter [18]. In those scenarios a Cloud service bus is helpful for a secure message interexchange. Another important factor for the proper usage of elasticity is the implementation of scalability mechanisms which can be classified in three groups: cycling scaling, event-based scaling, and auto-scaling [18][1]. The last method is the most interesting one and it can be achieved by implementing rules that are supplied with diagnostics data from the application in order to make a scaling decision. Finally, elasticity is also an important property for the Cloud data stores. Relational databases are proven to scale out well. Nevertheless, technologies like key-value stores and document stores also scale out quite well horizontally. At the same time those technologies are a lot cheaper compared to traditional relational databases. Key-value stores do not have fixed schemas like relational databases do and do not guarantee full ACID (atomicity, consistency, isolation, durability). Instead they provide BASE (basically available, soft state, eventually consistent) [20]. In many scenarios relational databases are not really needed, so the optimal solution is to find the balance between relational databases and key-value stores.

Resource pooling (Multi-tenancy)

The second important characteristic, resource pooling, this is also known as multi-tenancy, deals with the pooling of resources in a way that those resources can serve multiple consumers of a service. Multi-tenancy commonly occurs in scenarios where Small and Medium Businesses (SMB) are sharing computational costs. This concept is not new as services like web e-mail are around since the Internet has emerged in the 1990's.

[1] If the application is not designed to be scalable, there is no point in implementing scalability mechanisms.

Multi-tenancy is closely related to scalability - the property of a service to be scalable (handling an increasing or decreasing number of tenants) and make use of the Cloud elasticity. If we want to achieve better scalability, we have to consider a tradeoff with the level of isolation of service tenants [21], so that no tenant is impacting the others. There exist different dimensions of isolation like security, performance, availability, data, and execution [21], [22]. In order for a service to provide decent quality, those dimensions should be taken into consideration for solving issues that might arise as a result of multiple parties sharing the same resources.

In addition to these dimensions, patterns for multi-tenancy can also be defined. In *Figure 5* we provide four patterns which are based on the combination of two existing studies [21], [23].

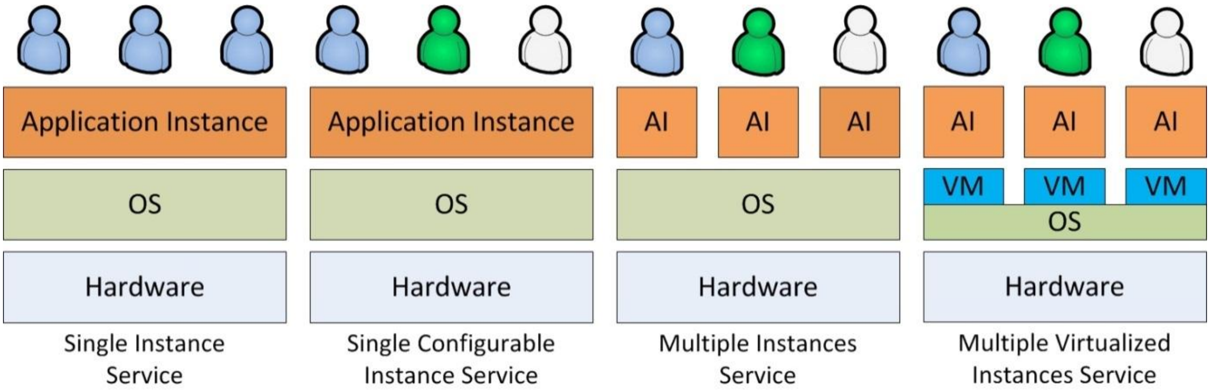

Figure 5: Multi-tenancy patterns

The first pattern, *Single Instance Service*, provides all tenants with the same service as they share the same application instance. An example for this is Amazon, as all tenants get exactly the same behavior of the web shop. In the second case of *Single Configurable Instance Service*, the different tenants get the requested service depending on the configuration parameters that they have defined. This can be illustrated with Google Mail as the tenants can configure how exactly the service looks like and which modules of the service are in use. The next pattern, *Multiple Instance Service*, separates the tenants, as they are sharing only the underlying operating system, and provides them with their own application instances. Services specified in WS-BPEL do not support the sharing of application instances on a per tenant basis, so this pattern is suitable in such scenarios [23]. The last pattern, *Multiple Virtualized Instances Service*, provides a greater level of isolation of the tenants as they share only the host operating system. The tenants use their own operating system and application instance on top of the host with the help of virtualization technology. An example for this pattern can be a corporate datacenter where different departments run their similar CRM systems using a powerful server.

2.3. Architecture of information systems

In this section we discuss the definition of architecture and its different types. We also provide a 3D thinking space, which we use over the course of this study. Finally we make a categorization of the different Cloud Computing reference architecture and present our viewpoint on the differences between Cloud Computing architectures and Cloud Computing application architectures.

Architecture definition

Over the years the term architecture has been defined in various domains such as building, computer science, and information systems. For the purpose of our study we are interested only in the architecture of information systems from product perspective. In 2000 a generally-accepted definition of architecture of software-intensive systems was provided by the IEEE [24]:

> *"The fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution."*

This definition is concise and clear but it is also too generic. As we are specifically interested in the architecture of information systems, we consider that we must make use of a more concrete and relevant definition to our research. Throughout the course of our study we adhere to the following definition [25]:

> *"The architecture of a (corporate) information system defines that system in terms of functional components and interactions between those components, from the viewpoint of specific aspects of that system, possibly organized into multiple levels, and based on specific structuring principles."*

This definition includes characteristics that we are interested in, such as organization into multiple levels. In our study we use three different levels (also called dimensions [25]): abstraction, aggregation and realization. The abstraction dimension defines how concrete an architecture model is with respect to the software building blocks. At its highest level of abstraction an architecture model provides no information about types of systems and software vendors, while at the lower level of abstraction even the product versions should be explicitly mentioned. The aggregation dimension defines how detailed an architecture model is with respect to the number of components. At the highest level of aggregation a system is only a single black box, while at the lowest level all small subsystems are identified. The realization dimension defines how close a model is to the technological implementation of a system. The architectural descriptions along this dimension can be categorized in the range from very business-orientated to very technology-oriented models. This dimension is based on the BOAT framework [26]. When combined with the other two dimensions – abstraction and aggregation, we end up with a three-dimensional design space presented in *Figure 6*, which is presented in [25].

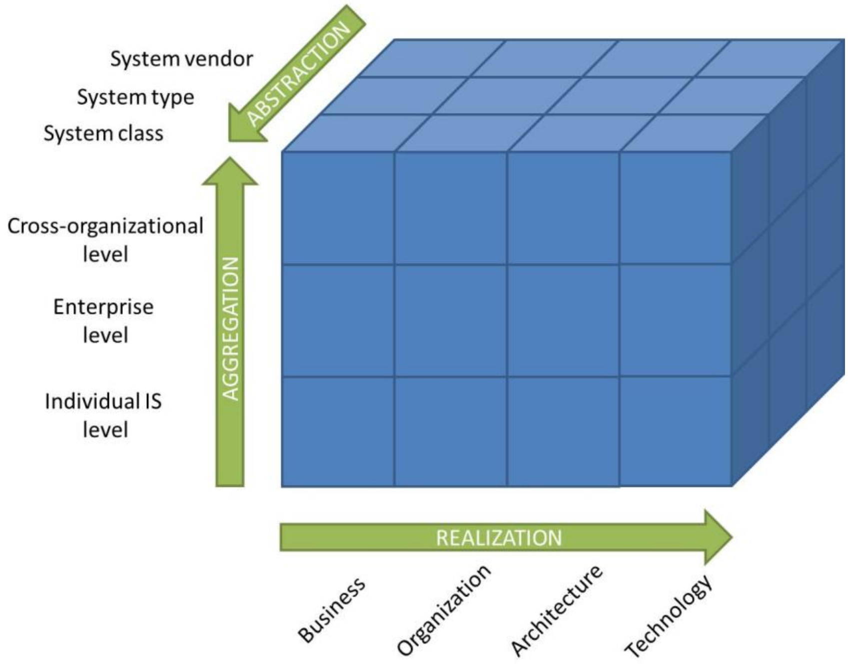

Figure 6: 3D Design Space

Architecture types

We identify several types of architectures that provide different levels of abstraction (concreteness) and aggregation (granularity). The first flavor of architecture we call nowadays *concrete architecture* [25] or *solution architecture* [27]. Both terms refer to the underlying structure of a software system that offers a defined set of functionalities [27] which meet the business requirements in a specific situation. Over the years there were other terms for the same type of architecture as software architecture[2] [28] or capability architecture [29] which are still valid terms. When a type of application is often encountered, it is wise to create high-level designs that can be reused in multiple situations concerning this type of application. Such designs are classified as *reference architectures* [25]. In most cases reference architectures are based on accumulated best practices in an industry domain [27]. Still, some reference architectures are designed as a result of research in a new area of knowledge, so some differentiation can be made [30]. Those two different types of reference architectures are classified as Practice Reference Architectures and Futuristic Reference Architectures [30]. Most reference architectures are either generic like CORA [31], domain-specific like NORA [32] or technology-specific like Microsoft .NET [33]. A specialization of reference architecture is *standard architecture* which is specific for a single organization [25]. This type of architecture could be found in large enterprises or governmental institutions [34]. *Figure 7* provides the relationships between the different types of architectures, which can be also seen as traversing the abstraction level of the different architecture types.

[2] In reality, software architecture can be regarded as umbrella term for most of the terminology that we discuss in this chapter.

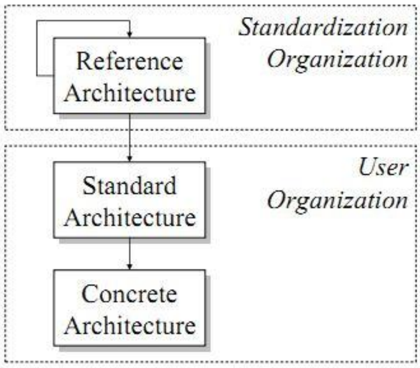

Figure 7: Reference and Standard architectures [25]

An important topic for industrial engineers nowadays is the concept of enterprise architecture. It can be simply defined as the architecture of an enterprise which provides some principles sufficient to meet the requirements of a business [27]. This definition is quite generic and does not show any relation to the technological and business process aspect of an enterprise. For this reason, we consider the following definition [35]:

> *Enterprise architecture: a coherent whole of principles, methods, and models that are used in the design and realization of an enterprise's organizational structure, business processes, information systems, and infrastructure.*

It is important to notice that enterprise architecture provides not only blueprints (or models) but also principles and methods for the development of the architecture which is important when designing a large enterprise. There are various frameworks which can be used to develop the enterprise architecture of a company. Some of the frameworks like Zachman Framework are more general [36], others like TOGAF [37] and IAF [38] touch only a single aspect of enterprise architecture like the methods or the models. It is common that a method framework like TOGAF is combined with different reference architectures in order to achieve the overall enterprise architecture. In this light, the main product of our research, which is a reference architecture, complements *Phase C: Information Systems Architecture* of the TOGAF framework [37]. In this phase of the architecting process, the application and data architectures are defined. Our reference architecture for business processes in the Cloud has a strong connection with the design of an application architecture as it provides the blueprints for the interaction between the Cloud platform and the on-premise resources of the customer.

Cloud Computing reference architectures

During the last three years both academia and industry have been active in defining reference architectures for Cloud Computing. In order to discuss the different propositions in an easy to understand way, we categorize them according to their levels of abstraction and aggregation [25]. We plot their relative values and discuss them along the aggregation axis in *Figure 8*.

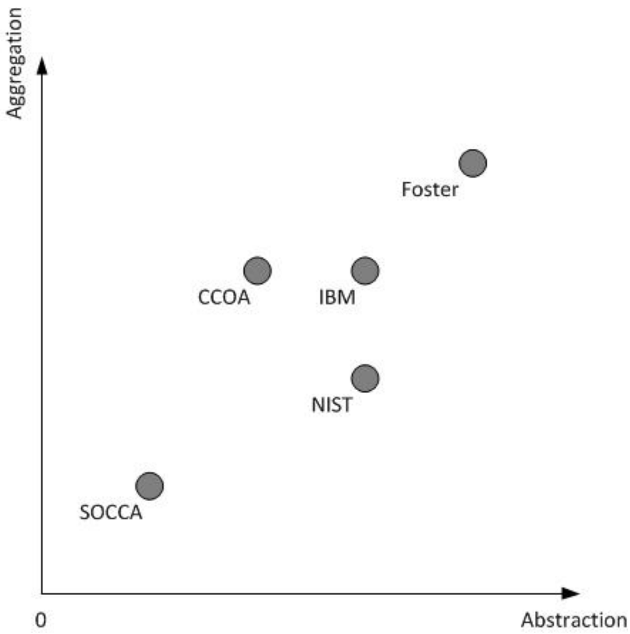

Figure 8: Plotting Cloud Computing reference architectures

The simplest model, labeled as *Foster* in *Figure 5*, has high levels of abstraction and aggregation [13]. The authors provide four-layer architecture without showing any details about the layers. The top three layers of the model can be correlated with the three basic service models that were discusses in the previous chapter - IaaS, PaaS, and SaaS. The fourth layer presented is the actual hardware but there is no service model which directly relates to using hardware. A bit more granular model is provided by IBM [39]. The model is defined in terms of modules which are connected to three main roles: Cloud Service Consumer, Cloud Service Provider, and Cloud Service Creator. The reference architecture model of IBM provides also drilled-down models for the modules in the main model. Thus, it is possible to slide the reference architecture along the aggregation axis in *Figure 5* in a downward direction but for our comparison we take into consideration only the base models presented. Another reference architecture developed by IBM is the Cloud Computing Open Architecture (CCOA) [40]. It was developed 2 years earlier than the IBM reference architecture. It has the same level of aggregation but the level of abstraction is lower as the model uses more technology-specific terminology. The architecture is based on modules which are encapsulated in a holistic ecosystem. A downside of the model is the omission of a module that handles security. The authors of the architecture provide a drilled-down model which is even more concrete from technological point of view. Similar to the first reference architecture of IBM is the one defined by NIST [41]. It is a vendor-neutral architecture that is consistent with the NIST definition for Cloud Computing [11]. According to NIST, the architecture is designed in a way that it does not impair the innovation in Cloud Computing at the technology level [41]. This architecture is also role-based but this time there are five different roles identified: consumer, provider, auditor, broker, and carrier. The background colors used for the components of the model make it ambiguous as they do not fully correspond to what is described in the specification document. The most low-level architecture in terms of aggregation and abstraction is the Service-Oriented Cloud Computing Architecture (SOCCA) [42]. It is a layered architecture which is used to aggregate services from different Cloud Computing providers using ontologies. The authors of the architecture have the opinion that at the current state of Cloud Computing services there are no Service Level Agreements or support for multi-tenancy. Even at the time of publication some of the vendors were providing such kind of features. In addition to those generic reference architectures for Cloud Computing there are publications concerning architectures that have a more specific purpose like federation of Cloud Computing services [43], building applications for the .NET framework [44], or achieving market equilibrium through regulation supply and demand of Cloud Computing services [12].

Cloud Computing architectures versus Cloud Computing application architectures

Before starting our discussion of the application architectures that are used in Cloud environments we should state the difference between "Cloud Computing architectures" and "Cloud Computing application architectures." The first term refers to the conceptual, high-level organization of a Cloud Computing platform without taking into consideration the underlying technologies. Some significant research has been conducted in this area and different reference architectures have been defined as described in the previous section. The second term under discussion refers to the structure of an application that is running in a PaaS environment. The limitation to PaaS comes from the fact that the IaaS model provides empty virtual machines which can be configured in a way that mimics a traditional on-premise computing environment, so there is nothing new and specific in architecting those applications. Furthermore, if we look into SaaS, it is possible to design something similar to an application architecture but there are limitations. There exist cases when the SaaS solutions are mash-up applications that integrate services from different providers, so some kind of architecting can be done. Nevertheless, it is limited to the SaaS solution and the level of architectural creativity that it allows.

3. Defining a Business Processes Cloud Platform (ProCPlat)

As we already mentioned the Cloud has become one of big hypes in the IT industry recently. If we complement this hype with more buzz words as process outsourcing and process orientation, we will reach a more specific aspect of the Cloud which is often named Business-Process-as-a-Service (BPaaS). BPaaS is widely regarded [39], [45] as the fourth service model on top of the three models proposed by NIST [14]. It is still underdeveloped and there is a lack of clear definition what exactly it encompasses. In this chapter we extend on the current concept of BPaaS presented by the industry and academia, and formulate a new definition for a Business Process Cloud Platform (ProCPlat). This chapter presents the second step towards fulfilling our goal. *Section 3.1* provides our definition of ProCPlat, extending on one of the already existing definitions. In *Section 3.2* we briefly describe the reasons for the emergence of the BPaaS Cloud service model which are also relevant for our platform. *Section 3.3* describes the different ownership scenarios of processes depending on the parties responsible for definition and enactment and *Section 3.4* maps those scenarios to the existing Cloud Computing stack. In the last two sections we describe some additional consideration and the present industry state for business processes in the Cloud.

3.1. Defining ProCPlat

Even though a lot of research is currently focused on the Cloud, the business process area inside it is still vaguely defined. One of the reasons for this is its industrial immaturity. We must make it clear from the beginning that BPaaS, which is marketed as the service model dealing with business processes, is not just another fancy word for SaaS that deals with processes, even though BPaaS tends to be considered as a special form of SaaS in some cases [46]. For example, typical CRM solutions in the Cloud, which involve some business processes, would continue to be part of SaaS, while BPaaS would give more flexibility and agility to the end user in configuring predefined processes that are executed in the Cloud. Later in this chapter we describe how business processes relate also to SaaS, and even to PaaS, as our platform covers a wider spectrum of scenarios involving business processes. Currently the best definition of BPaaS is provided by IBM as they merge information regarding the Cloud market presented by two of the industrial research companies – Gartner and IDC [39]:

> *Business process services are any business process (horizontal or vertical) delivered through the Cloud service model (Multi-tenant, self-service provisioning, elastic scaling and usage metering or pricing) via the Internet with access via Web-centric interfaces and exploiting Web-oriented cloud architecture.*

This definition is not further elaborated upon by IBM and it does not explicitly specify whether the processes can be custom-defined, even though that can be implicitly concluded from placing BPaaS on the top of the Cloud services stack. In addition, the definition does not include some characteristics that we consider important for running business processes in the Cloud. Those characteristics can be found in different publications and at the end of this section we provide a new definition for a Business Process Cloud Platform which includes some of the characteristics. The Open Group provides an explanation of BPaaS [15] which extends the NIST definition of Cloud Computing [14] and gives an important addition to the definition of IBM – the processes running in the Cloud can be partiality supported by people. In addition, a lot of processes which are a subject of Business Process Outsourcing (BPO) like payroll, HR, CRM, etc. are mentioned. They form the vertical implementation orientation and one half of the processes that can be executed in the Cloud according to us. The functionality in this half would be categorized as out-of-the-box functionality as enterprises prefer to get the whole stack of processes related to non-core business activities. The other half, which is a subject of the Business Operations Platform concept [45], is connected to the creation of custom processes defined by the application developers [47]. This could be regarded as the horizontal implementation orientation which in other words is BPMS functionality in the Cloud. The functionality of this half would be then both out-of-the-box and custom as this implementation orientation provides greater flexibility. *Figure 9* provides a graphical representation of the distinction between vertical and horizontal implementation orientation:

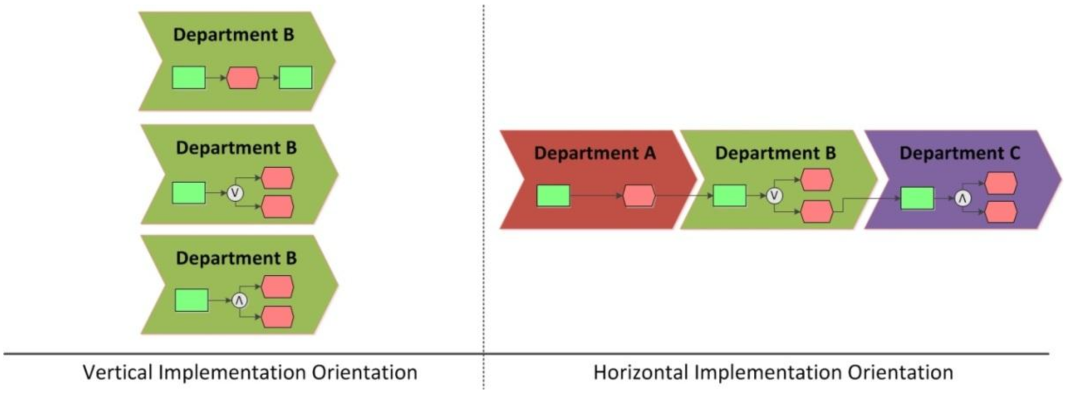

Figure 9: Vertical vs. Horizontal Orientation

Different authors also propose a form of syndication[3] of processes and their brokerage inside a BPaaS platform [47], [48], [49]. It is also claimed that the current Cloud service stack is rather restrictive which makes the development of process-centric applications complicated [47]. Thus, the syndication could be even elevated to the level of dynamic creation of business processes, similar to the concept of instant virtual enterprises (IVE) [50] and involving the concept of Universal Description, Discovery and Integration (UDDI), which is part of the Web Services protocol stack [51]. Based on those additional characteristics we form our definition of ProCPlat:

> ***Business Processes Cloud Platform (ProCPlat) is a platform composed of Cloud services that enables the definition, enactment, and syndication of business processes (fully automated or partially supported by people) in both horizontal and vertical fashion via Web-centric and/or programming interfaces.***

This definition uses as a basis the IBM interpretation of BPaaS [39] but it adds some properties that we regard as important. We consider the syndication (and at a more generic level the integration) of custom processes a factor that will drive the emergence of ProCPlat in the future together with the partial support of processes by people.

3.2. Reasons for emergence

After extending on the BPaaS concept we provide some reasons for the forecasted emergence of this additional Cloud Computing platform as we see them at the current moment. First, the most obvious reason is financial and it coincides with the reasons for the emergence of the Cloud in general – economy of scale. Additionally, implementing an in-house BPMS solution is expensive not only because of the initial charges for consultancy, but also because of the licensing fees. In this case a reasonable question to be asked would be if the BPMS companies would agree to lose part of their

[3] Syndication can be regarded as a more specialized form of integration

revenue as the Cloud uses the pay-per-use principle. The answer is that ProCPlat would allow more companies to use process-oriented technology which might actually increase the profit of the vendors at a later stage. The second reason for emergence that we identify is connected to agility. Currently, most of the SaaS solutions are canned, so only cosmetic configuration is possible [45]. ProCPlat would enable the customization of the process-aware systems and syndication of processes with other business partners which would give ProCPlat an advantage in Business-to-Business scenarios.

3.3. Ownership model in ProCPlat

Another important characteristic that requires discussion is the governance model around ProCPlat – who is the owner of the process definitions, who is allowed to enact them, and who is actually executing them. Currently, most of the business processes run inside corporate datacenters and the interaction with external parties is achieved through complicated integration schemas. Bringing the processes or parts of them into the Cloud enables the emergence of new ownership scenarios and we identify four of them, adding on top of some already identified [46]. The four scenarios are mapped in *Figure 10* according to the multiplicity of the process definition owner and the multiplicity of the process enactment owner. We consider that in all scenarios the party responsible for the execution of the process is the ProCPlat vendor.

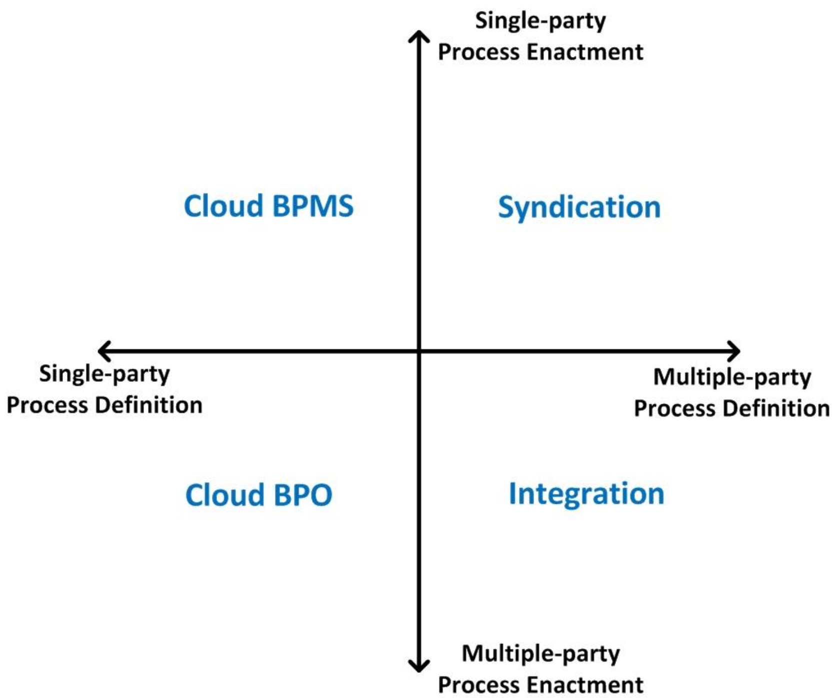

Figure 10: Mapping ownership scenarios

Cloud BPO

In the first scenario the owner of the process definition is the actual provider of the ProCPlat or a third party that provides a process definition as an intangible product. In this case the customer is adding only configuration changes to the process like setting parameters and enabling/disabling certain steps in the process. The same process definition can be enacted by different parties as it is usually a standardized process for a specific domain like human resources, procurement, etc. This scenario resembles a traditional BPO where the processes for a specific part of the company are run by a third party. Such kind of service already exists in the industry as we describe later in this chapter. A similar concept is also present in the academia under the name *Configurable process models* [52]. Those models are non-executable but they serve as a meta-model for similar process models. After configuration they can be converted to executable models. The concept touches also on our next scenario too as the actual executable definition is finalized by the client of the platform.

Cloud BPMS

In the second scenario the sole owner of the process is the consumer of the ProCPlat service. The consumer also enacts the process. This is similar to porting processes from a traditional BPMS solution into the Cloud (also named BPM-as-a-Service). Such kinds of solutions are already present at the market and they do not bring a lot of innovation to the BPM field. Instead, there is a lot of marketing buzz created around simply moving a system into the Cloud and labeling it "as-a-Service".

Syndication

In the third scenario the customer is creating a *syndication* of two or more processes or process chunks coming from different providers. There exist an intermediate process definition created by the customer of the platform, but still the actual sub-process definitions may have a different owner. The client of the platform is responsible for the enactment of the process. In this category also fall processes which belong to the Dynamic Case Management (DCM) field, where the sub-processes are defined by a single party. In that case, the user of the DCM system can rearrange the steps to be executed depending on the case data. Currently, similar systems are emerging at the market as they provide more flexibility.

Integration

In the fourth and most interesting scenario we have a joint ownership of a process by two or more consumers of the BPaaS platform. This case provides a seamless *integration* of processes of two enterprises which is important for Business-to-Business (B2B) relationships. Additionally, the scenario relates to the concept of business network process (BNP) [53] which helps for the realization of collaborative processes in B2B environments. The *integration* quadrant of *Figure 10* should be sufficient to cover an architecture supporting BNP's [53]. This scenario shall also involve scripting and programming in order to achieve the data integration between the parties involved in a common process.

3.4. Positioning of ProCPlat in the Cloud stack

After defining four scenarios for process ownership, we consider important to map those to the existing Cloud Computing stack with IaaS on the bottom and SaaS on the top. In this way, the scenarios are presented in a more realistic and easier to understand way. We present our addition to the current stack in two ways – a detailed model and a general model.

Detailed model

The model presented in *Figure 11* positions the four ownership scenarios in the Cloud Computing stack. The *Cloud BPO* scenario directly maps to the layer defined as BPaaS by IBM [39] as it provides ready-made processes to the platform clients. This scenario goes directly above SaaS. Then, the *Cloud*

BPMS can be considered a type of SaaS as it provides BPM functionality in the Cloud. Such kinds of solutions already exist and are commercially available. Next, the *Syndication* scenario has two sides – combining processes/process chucks from different providers or using self-made process chunks as a Dynamic Case Management (DCM) system. This can be also seen as a type of SaaS as the users of the platform do not have to configure complicated integration patterns. The last scenario of *Integration* maps to the level of PaaS as it requires more fine tuning in order to be successfully used by the platform clients. Additionally, we expect that some scripting and programing would be needed in order to have a smooth integration of different business entities.

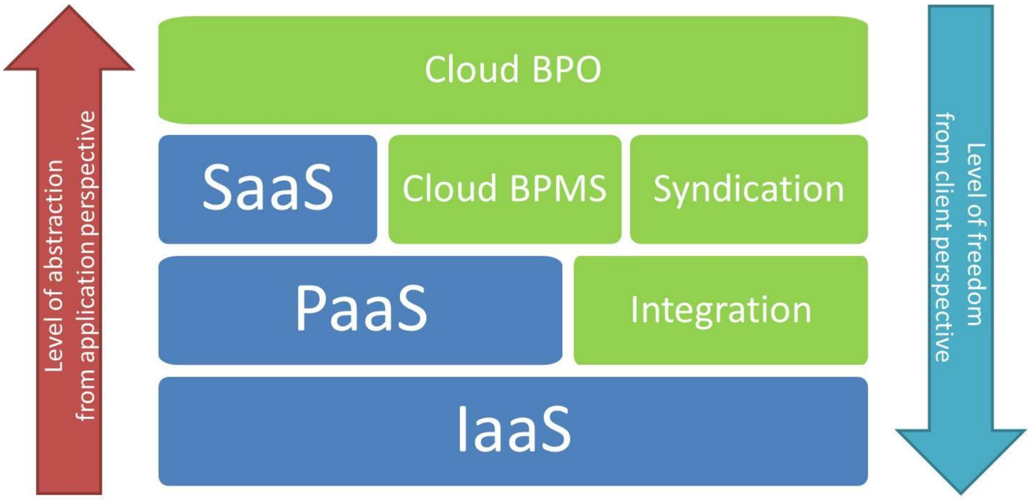

Figure 11: ProCPlat in the Cloud stack - detailed model

On the two sides of the model we present also the levels of abstraction and freedom. They are not the two counterparts of the same concept, but just a good fit to describe the model in both top-down and bottom-up fashion. The top-down approach follows how the freedom from clients perspective changes. At the top layer the customer has almost no freedom, since only configuration changes to the processes can be added. At PaaS layer the customer has greater level of freedom as more complicated integration mechanisms can be implemented. The bottom-up approach illustrates how the level of abstraction from application perspective is changing. At the *Cloud BPO* level the service provided is very generic and can be used by many different enterprises. Some services in the SaaS layer like e-mail and office productivity also have high level of abstraction but this layer also contains services as BPM in the Cloud, CRM, ERP, etc. which are a bit more specific. At the PaaS layer there exists a high level of customization for the user needs, so the level of abstraction is low.

General model

The following model, presented in *Figure 12*, shows the position of ProCPlat in the Cloud stack in general. As the four scenarios that we define are different in their nature, the ProCPlat spans more than one layer of the stack as we already showed in the detailed model. This is also the reason why we define our product as a platform rather than as a service. A platform encapsulates different functionalities and allows for different scenarios to be run on top of it. The services included in our platform also have different levels of freedom and abstraction, so generalizing them under the “as-a-Service” label is impossible. Similar examples can be found also in industry. Recently, Microsoft created features on its Azure PaaS that are typical for IaaS. Similarly, Amazon went from IaaS towards PaaS with the introduction of programming framework in its AWS platform. These two examples can serve as a proof that vendors are actually going in the direction of providing one platform that spans different layers of the established Cloud stack.

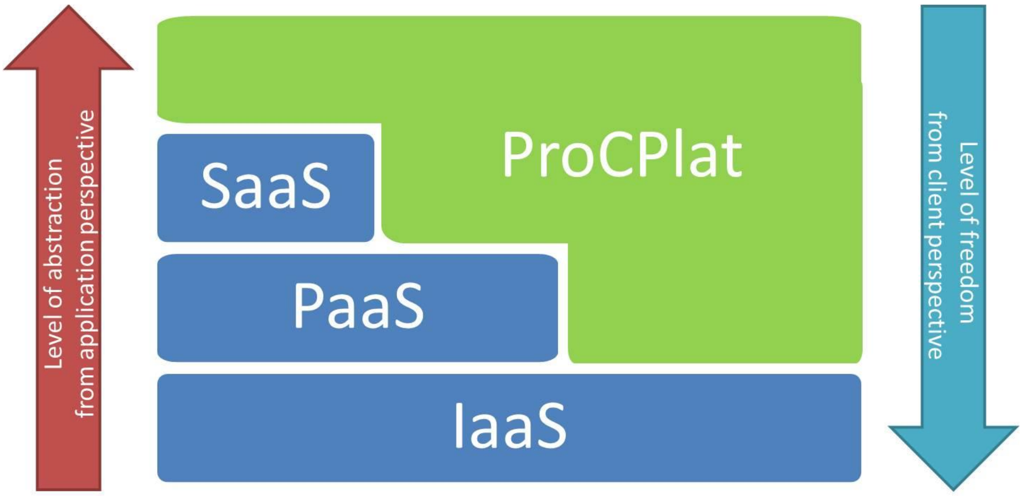

Figure 12: ProCPlat in the Cloud stack - generic model

3.5. Additional considerations

In order to conclude with the specifics of ProCPlat we need to mention several additional factors that might bring some concerns to the market. In general the business process area in the Cloud should bring more innovation and be one of the significant areas for development in the future. The proposed ProCPlat is expected to bring wider reach for BPMS technology, especially in SMB's. Also on the other side of the Cloud boundary, the ProCPlat might create a more fragmented market for processes as some new specialized companies start selling their know-how for specific processes. The ProCPlat should help resolving the vendor lock-in [47] to which Cloud customers are bound, since moving solutions between different platforms is hard to be achieved. This is due to the differences in the programming models of the Cloud vendors. ProCPlat would enable a dynamic collaboration between parties with smooth integration. Nevertheless all this might come with its price – companies being unwilling to put their processes in the Cloud, which in most cases represent the businesses' core competences. This would create a good amount of work for the legal departments as sensitive information would go beyond the corporate firewall. An additional concern would be the ownership of the business process data at any given moment in time. Still, the ProCPlat providers are expected to be flexible enough in order to fulfill the expectations of the market, so we consider that this issue would find its reasonable solution.

3.6. Industry state

At the current moment no one of the vendors labels its service offerings officially as BPaaS, nor is a platform similar to ProCPlat proposed. Still there are services that match up to a point part of our definition for ProCPlat. The closest match comes in the BPaaS field from IBM with IBM Global Expense Reporting Solution [54] which provides out-of-the-box configurable processes that deal with handling expenses. Additionally, IBM offers BlueWorks Live for Cloud process modeling. Three of the other big vendors that have some offerings are Cordys, Pega, and SAP. Cordys provides a service called Process Factory which touches on the topic of syndication which we identified already as important feature of ProCPlat. Pega and SAP market their solutions as Cloud-based but it is harder to specifically relate them to BPaaS as they seem more similar to BPMS in the Cloud, which is common for the SaaS service model. There are also two consultancy companies which are mentioning BPaaS in their service offerings – BeyondCore and Genpact. As a whole the market for business processes in the Cloud is not yet matured, so making a deep analysis at this stage is difficult.

3.7. Case studies (ownership models justification)

In order to justify the choices made for our ownership model described in *Section 3.3*, we are going to mention some out-of-the-box examples from the industry (for Cloud BPO and Cloud BPMS) and describe two projects executed within Capgemini (for Syndication and Integration). Syndication and Integration solutions as we perceive them are currently missing from the market. We are not providing concrete names for the parties involved in the projects, so that we keep the confidentiality of the work. For both of the Capgemini cases we follow the same structure in order to analyze them: explanation of the industry, definition of the objective, details for processes, and some insights on the technological implementation.

Cloud BPO

As we already mentioned in the previous section, there are services offered, which fall into the category of Cloud BPO like IBM Global Expense Reporting Solution. We expect that there are other similar services to the one of IBM which can be categorized as Cloud BPO. Those are hard to be distinguished as they are not explicitly labeled as BPO in the Cloud. We expect that in the future more services will get labeled as Cloud BPO or BPaaS. This thought is also shared by one of the industry research companies [55] which predicts a significant growth in the BPaaS market between 2015 and 2020.

Cloud BPMS

Getting the standard on-premise BPMS functionality in the Cloud is already common. Most of the big BPM software vendors offer Cloud BPMS (also labeled as BPM-as-a-Service). Some examples of companies having such kind of solutions are Pega, SAP, Appian, Cordys, Perceptive Software, and Software AG. According to us, a push to the promotion of Cloud-based BPMS was given initially by the successful emergence of CRM solutions offered as a service such as Salesforce.

Syndication (Project A)

Industry: Project A is taken from the public sector. It deals with the processing of cases, which involves heavily the participation of citizens and numerous third parties.

Objective: The project aims at replacing the old information system (mixture of paper-based and digital systems) that helps in the resolution of cases and introducing business process management technology, which is specifically oriented towards case management.

Process details: The nature of the business processes involved in the project is such that every case in the system has a different execution path. Since COTS (commercial off-the-shelf) case management systems do not offer sufficient flexibility for the business process execution, a custom case management system is implemented. As all the process execution paths share a common set of execution blocks (small process chunks), a way for their syndication (ad-hoc composition) is needed based on the needs of a single case. At the end of the project the administrative personnel is able to control what execution blocks are needed for a specific case, and add, remove, or repeat execution blocks throughout the lifetime of a case.

Technological implementation: The architecture of this project is based on different Microsoft products such as Dynamics CRM, BizTalk, SharePoint and SQL Server. In order to achieve the custom case management functionality, the workflow capabilities of Dynamics CRM are utilized. The product provides a way to define many small workflows that can later be executed in the order that is required by the specific case. This way, there is no need for big amount of different process models, which can be configured depending on a specific case.

Integration (Project B)

Industry: Project B is taken from the private sector for utilities. It deals with the interexchange of data between the different parties joined in a union. There is no designated leader of the union and parties form committees which work on the specification of standards.

Objective: The project aims at facilitating the execution of business process definitions, which are agreed upon in between the parties participating in the ecosystem, through the definition of standard interfaces for data interexchange and of message formats.

Process details: The business processes involved in this project handle the interaction between the utility providers and the clients and in between the providers themselves. The defined processes facilitate, for example, the smooth transition when a client changes its provider as the actual switch is handled by the client's new provider. There are different processes defined for private and enterprise clients. Parts of the processes are inherited from legacy systems that were used to handle the interaction between the different utility providers. The system which results from this project acts as an orchestrator for the different utility providers involved in the ecosystem.

Technological implementation: There is no specific vendor choice for this project. The different parties which take part in it have the freedom to choose the vendor for their own platform. Nevertheless, there are standards set. The data interexchange uses SOAP Web Service and XML schemas in order to achieve the desired functionality. As those technologies are defined with standards, they can be implemented by different platforms.

4. Architecture aspect

In this chapter, we start with the elaboration of the models that present our reference architecture for ProCPlat. We begin with a model describing the Architecture aspect that is highly abstract and highly aggregated. That is a non-standard approach as usually the design starts from the Business toward the Technology aspect. The motivation for this choice originates from the idea that Architecture has a pivotal role in between Business and Organization, and Technology [26]. As we propose an innovative platform that captures a lot of business cases, starting from the Business aspect would limit us in a specific thinking frame. Throughout the next two chapters we also describe the Organization and Business aspects. In *Section 4.1* we present the global architecture model. In *Section 4.2* we provide a detailed model as we zoom into some of the modules defined in the global model. We conclude the chapter with two running cases that help for the understanding of our models. The same cases are used in the following two chapters as well.

4.1. Global model

In this section we present the global model for ProCPlat at the Architecture aspect.

Positioning into the 3D design space

In *Section 2.3* we introduced a three-dimensional design space, which helps for understanding the relations between different models that are needed for the design and development of a software system. In *Figure 13* we highlight with orange color the starting point for our design process.

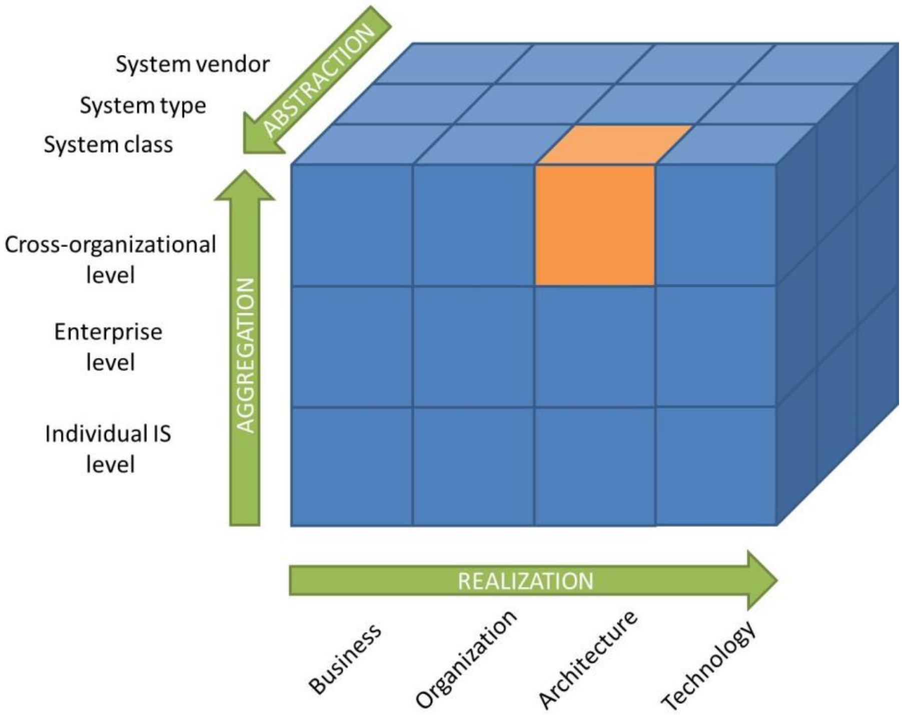

Figure 13: Positioning the global architecture model

Global architecture

The ProCPlat global architecture model is shown in *Figure 14*. The two most important modules are situated in the center of the model: the Business Process Execution Module and the Communication

Interface. They provide the means to execute a business process, and connect the execution module to the rest of the subsystems that depend on it and which feed it with the needed data. In order to reach this state of the model we had a look at other WFMS artifacts: the WfMC reference model [6] and the Mercurius architecture [7]. Additionally, we reasoned what modules are needed, based on the four ownership scenarios that we already defined in *Section 3.3*. Underneath the presented model, we elaborate on each of the modules illustrated. The modules that are colored in blue on the diagram are part of the actual Cloud platform for business processes.

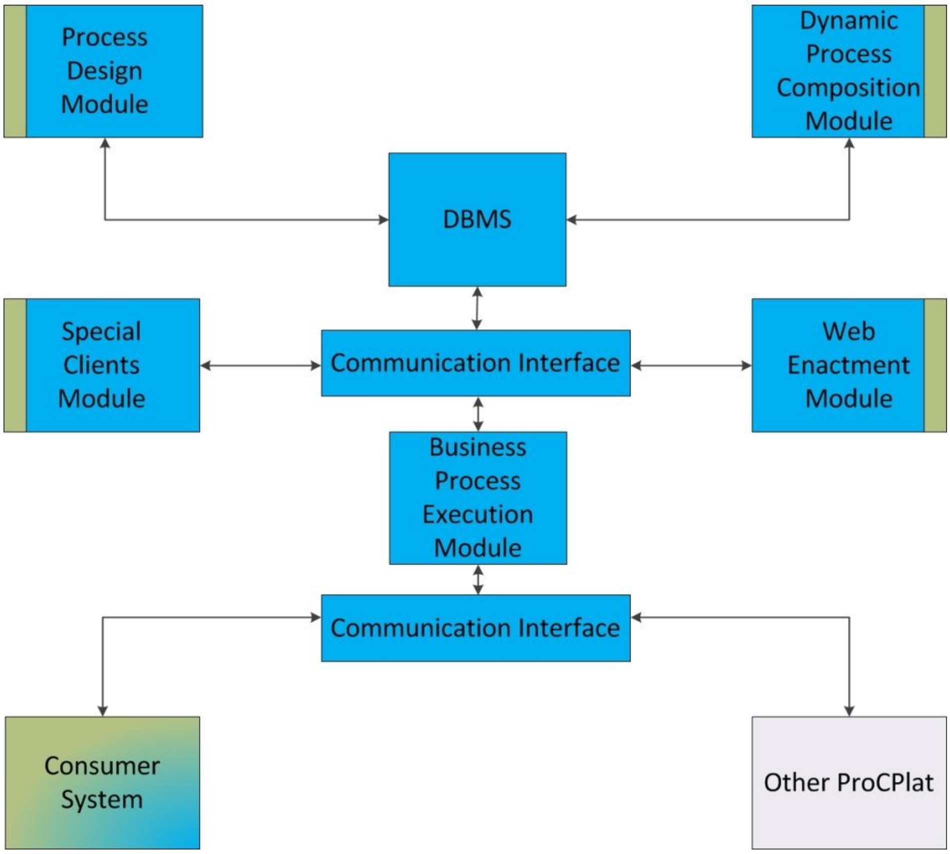

Figure 14: ProCPlat global architecture model

Business Process Execution Module. The business process execution module provides the functionality needed for the execution of business processes. Through the communication interface, this module receives the input data from different sources and sends the output result of the execution to the designated module. This module is further elaborated on in *Section 4.2*.

Communication Interface. The communication interface serves as a gateway for data interexchange between the business process execution module and suppliers and consumers of business process data. It facilitates the communication between the ProCPlat and the outside world, as well as the communication between the internal modules of the Cloud platform. This module is also further elaborated on in *Section 4.2*. For better understandability of our architectural model we have split the Communication Interface into two blocks. This can be seen as a logical division between the internal and external communication. In a Technology aspect the two Communication Interfaces can be supported by the same software module.

DBMS. The database management system serves as a bridge between four modules of our platform. It is the single point of entry to a data bank, which holds all the process definition and process execution data. Additionally, it connects the communication interface with the process design module and the dynamic process composition module which support the consumer of the platform in the definition of their business processes.

In our perception under the DBMS should reside a data bank which serves as a repository for the process data. We do not want to discuss in detail how exactly the data is stored or how the data bank is portioned, so that it can handle the different types of data needed for the operation of ProCPlat. The Mercurius reference architecture already provides some insight on what kind of data stores are needed to run a WFMS [7]. We consider that the way process data is stored logically has not changed. As for the physical storage, process data might be stored not only in relation databases but also in the emerging non-relation databases such as key-value stores.

Process Design Module. The process design module provides the means for the design of business processes in ProCPlat. It is relevant for all four ownership scenarios that we already refined. In the Cloud BPO and Syndication scenarios the module can be used by external process definition providers for the definition of whole processes or process chunks that are later sold as an intangible good to the customers of the platform. In the Cloud BPMS scenario one customer of the platform is defining its own processes, while in Integration scenario multiple parties are working together on a common process definition. There is already a commercial solution proposed by IBM [56], which enables businesses to collaborate on a common process definition.

Dynamic Process Composition Module. The dynamic process composition module is intended to be used in the Syndication scenario as a tool where the customer can dynamically assemble processes or of process chunks based on case-specific data. Those chunks can be either define by the customer itself or bought from an external process definition party.

Web Enactment Module. The web enactment module provides basic functionality to enact a business process manually. It can be seen as a standard client, typical for every BPMS system with the difference that it is ported to the Cloud. For this reason it is a web-based client.

Special Clients Module. This module is intended to handle special purpose clients that are not intended to directly interfere in the enactment of a process. We have three functionalities in mind that we consider special. Firstly, a monitoring functionality should be present, so that the customer can observe the performance of its business processes based on different metrics. Secondly, depending on the monitoring data, the customer should be able to change the number of physical nodes serving its processes in a manual way. Finally, a billing functionality should be present. ProCPlat creates a new ecosystem where external parties can offer process definitions at a marketplace, thus the customers for those processes should be internally billed. Additionally, the actual owner of the ProCPlat should be able to bill the customers of the platform.

Each of the last four modules has a small part that is colored in green in *Figure 14*. By doing this we want to indicate that those modules receive input from the outside world, i.e. the customers of ProCPlat, which is vital for the actual execution of the business processes. We can also consider those green parts as a form of web thin clients.

On the bottom of our model we present two modules that can collaborate with ProCPlat in order to realize a common business goal. ProCPlat should be able to communicate with a Consumer System and receive the data which needs to be handled as a part of a business process. Additionally, ProCPlat should be able to communicate with other similar platforms, so that companies using different Cloud vendors to run their business processes can still communicate and exchange data at the Cloud level, instead of going into more complicated integration patterns involving enterprise services buses.

After discussing all the modules of the reference architecture in detail, it is easy to draw a connection to the WfMC reference model presented in *Figure 4*. The Workflow Engine in the WfMC model coincides with the Business Process Execution Module in our ProCPlat reference architecture and the Workflow API maps to our Communication Interface. The five interfaces defined by the WfMC are related to ProCPlat as follows: IF1 is the interface between the Process Design Module and the DBMS, IF2 is the interface between the Web Enactment Module and the Communication Interface, IF3 connects the Communication Interface to the Consumer System, IF4 is the interface between the Other ProCPlat Engine and the Communication Interface, and IF5 is connecting the Special Clients Module to the Communication Interface. Additionally, there are two more modules in the ProCPlat architecture that do not have a direct mapping the WfMC reference model: the Dynamic Process Composition Module which is responsible for the syndication of processes and the DBMS. The reasons for this difference are that the composition module is a newer concept that was not existent at the time of the WfMC model creation, while the DBMS was abstracted out of the WfMC model. We consider that both modules are important for such an architectural model, and thus our design decision to include them in the ProCPlat architecture.

4.2. Detailed model

In this section we present the detailed model for ProCPlat from the Architecture aspect.

Positioning into the 3D design space

In *Figure 15* we show the place of the detailed model in our 3D design space.

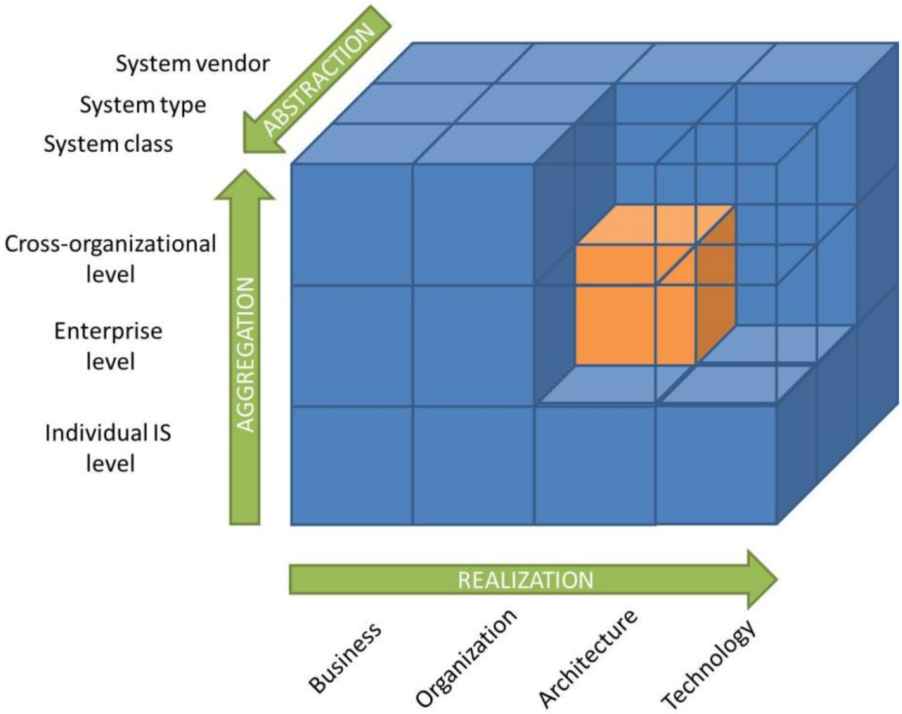

Figure 15: Positioning the detailed architecture model

This new model is not only less aggregated but also less abstract. From realization point of view the model still stays in the Architecture layer.

Detailed architecture

The ProCPlat detailed architecture model is shown in *Figure 16* (the background coloring of the Business Process Execution Module is only meant for visibility). We provide a lot more details about the Business Process Execution Module. We also provide a less aggregated view on the four modules that can have interaction with the outside world. In order to reach this model, we took a look at the Cloud Computing architectures of Microsoft Azure and Amazon Web Services. Additionally, the modules that are surrounded by a thick black border are considered as black boxes. We do not provide further details about them as this is out of the scope of our Cloud platform for business processes. Underneath the presented model, we elaborate on specific modules that are presented in a detailed fashion.

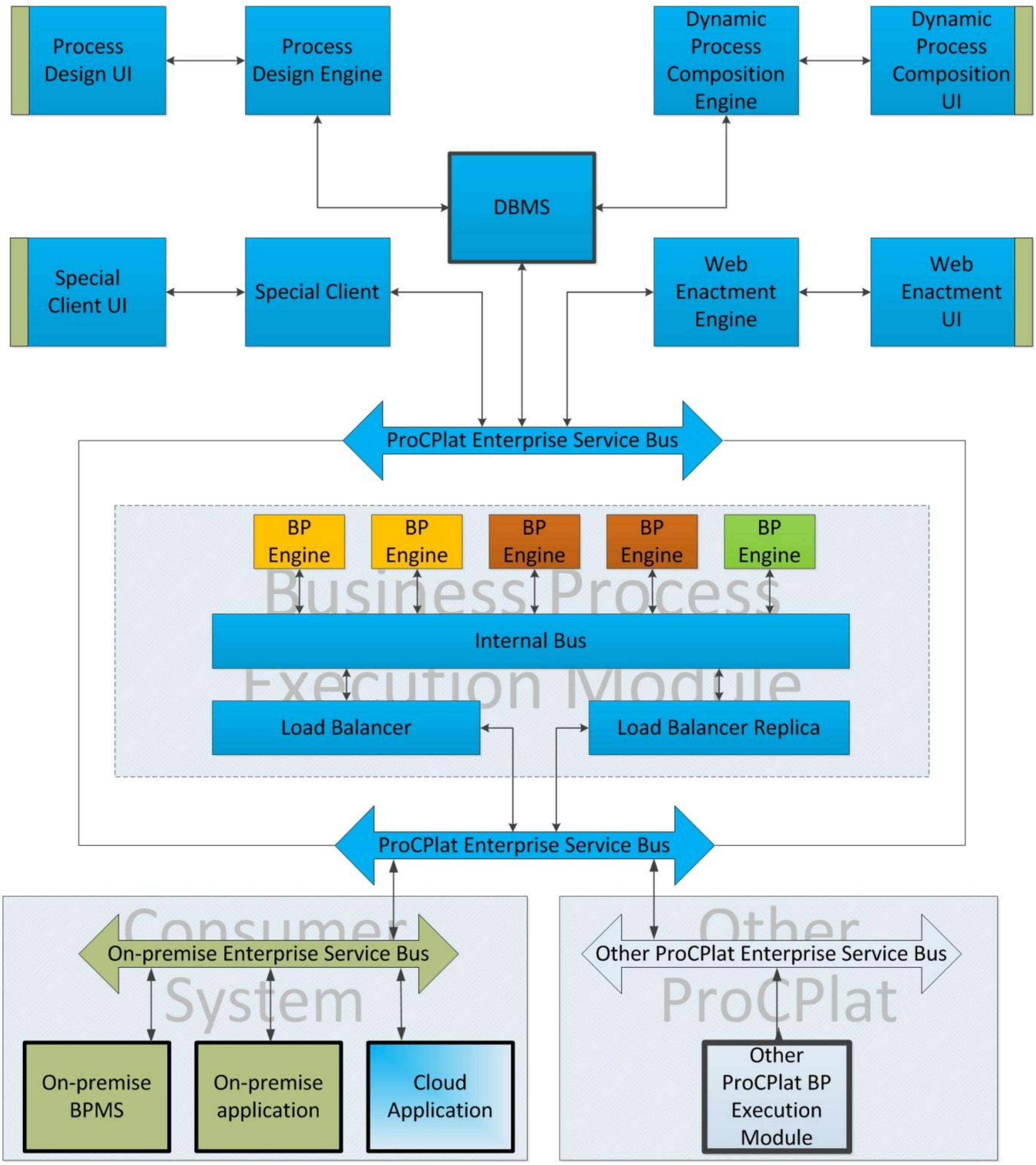

Figure 16: ProCPlat detailed architecture model

Business Process Execution Module

Load Balancer. The load balancer module operates in a way that it distributes the load between the different nodes in an even fashion. For nodes running the same process definition the load balancer can choose a new node to spin up a new process instance on round-robin basis which is a typical approach in Cloud services. Additionally, the load balancer always forwards the data for a specific process instance to the node that has already served it. This way the security is enhanced as the instance data can be kept local and only replicated on an external storage for safety purposes instead of having the completely centralized instance data storage. Another concern is the performance of a node that has to load different process definition every time it is serving a request. Some process definitions are quite large in size, so it would be better to stick to serving the same process instances with the same physical node. In order to handle a big load of the Cloud platform in general, we introduce a *Load Balancer Replica*. A replica is a mirrored image in this case of the load balancer, and it contains the exact same routing tables as in the load balancer. It should handle requests if the utilization of the system goes beyond a certain threshold. There could be also multiple replicas (this is not shown in *Figure 16*).

Internal Bus. The internal bus module plays the role of an internal service bus that delivers the messages from the load balancer module to the right node. The purpose of using such kind of communication is to reduce the number of connections between the numerous amounts of nodes and the load balancer that are dispatching the messages.

BP (Business Process) Engine. The nodes containing the business process engines in the Business Process Execution module are actually responsible for the execution of the business processes. They contain an engine that can handle a request and respond with a message. We present engines with three different colors in order to illustrate that the nodes might have different physical resources. This is important for clients that are aware of the fact of scaling up and scaling out. Currently, it is still impossible to split one ready-made process definition between different physical nodes. A recent study suggest that there are some workarounds for it [22], but we do not consider them a feasible solution for industrial business processes. For this reason, we propose that heavy process definitions are run on more powerful physical nodes. In cases when a process is not running according to the expectation it should be possible to seamlessly scale up and move the process to a more powerful machine. If there are too many instance of the same process we run into the case of scaling out. In this situation more similar machines running the same process definition should be spin up, so that the load could be handled without compromising the performance for the client. In a case of a node failure, the virtual instances running on the problematic node are migrated to a healthy one. This is a common pattern that current Cloud implementations handle out-of-the-box.

Communication Interface

The Communication Interface is composed of only one module which is the *ProCPlat Enterprise Service Bus* (ESB). In *Figure 16* there are two graphical representations of the ProCPlat ESB that are connected together. With this we want to illustrate that the ESB's for the internal and external communication are separated from one another at a logical level. The reason for this design choice is to improve security. The function of the ESB is connected with the data conversion between different formats, i.e., converting client data to data that can be used by the actual business process engine in the Cloud. Our choice for such type of technology is based on the fact ESB is common software for data interexchange in medium and large enterprises, which actually have implemented processes technology in place.

Consumer System & Other ProCPlat

We consider that ProCPlat should be able to seamlessly communicate with on-premise BPMS and legacy applications, so that the Business Process Management is partially migrated into the Cloud. Additionally, newly-created Cloud Application should be able to feed data to and receive data from ProCPlat. In some cases the vendor of ProCPlat might be the same as the one running Cloud applications for the consumer. Similarly, a connection with different SaaS applications like CRM or ERP in the Cloud should be possible.

We envision that the ProCPlat ESB should be communicating with other ESB's that are either hosted by a client of the ProCPlat or by another ProCPlat vendor. In *Figure 16* we label those two ESB's as *On-premise Enterprise Service Bus* and *Other ProCPlat Enterprise Service Bus*. The ProCPlat ESB should provide a data format specification to which the parties communicating with it should adhere to. For example, a client of the ProCPlat would create transformation schemas in their ESB, so that the corporate data is converted into the format required by the ProCPlat ESB. Then, the ProCPlat ESB would convert the data in a way that it can be fed into the process execution module. Businesses that do not operate using an ESB can still feed the ProCPlat with data. They can provide the formatted data to the ProCPlat ESB using a custom-built endpoint.

Another important point that needs to be mentioned is that exchanging data between the different Cloud providers is still a problem that has not been resolved. The IEEE has a draft standard for communication between different Cloud platforms [57] which sheds some light how inter-Cloud communication can be achieved. Some of the mentioned technology standards are XMPP and SAML. Those developments should contribute to the removal of the vendor lock-in which is currently present in the Cloud world. Additionally, a serious contribution could be the definition of entities that can be exchanged between parties. In 1996, NIST has defined the Electronic Data Interexchange (EDI) standard which has a lot of sub-standards and technology formats that can be used in the B2B world. The same concept can be reused in between Clouds. For example, there could be defined standard entities like order, bill of materials, invoice, etc. or a global schema for message formats. Such a standardization step would need time to be accepted by the industry because of the huge amount of custom-build system. Still, it would improve the ecosystem as a whole. The actual technical implementation of the data transformation between the ESB's and data interexchange is out of the scope of this project. For this reason, we only provide an idea how this can be actually realized in the Technology aspect.

Trying to give an idea about the realization in the Technology aspect, the most common approach would be the use of XML-formatted data as that is an approved corporate standard, and ESB software has capabilities for transformations into this format. Unfortunately, the XML standard has quite a big overhead. As we expect that the channel for data communication would be heavily utilized, the XML format might not be the most appropriate choice. Another viable option in our opinion is the use of the lightweight JSON format, which has a more concise syntax. The language also supports schemas which is vital for the Business-to-Business data interexchange.

4.3. Evaluation through scenarios

We demonstrate our models with two ongoing detailed scenarios. This approach is suggested in the IS Research Framework [1] as the right approach to evaluate innovate designs that do not have precedents. For the first scenario we use the public (governmental) sector as a target domain, where a municipality is the process owner. Often different municipalities have similar business processes which differ slightly depending on some region specifics and local legislation. Also, it is common that the business processes depend on the data connected to a specific business process instance. This implied that case management can be a more suitable technology compared to workflow management which does not provide process flexibility. This scenario is related to our Syndication scenario defined earlier in this study. For the second case we use the logistics sector. We relate the case to an imaginary set of companies that are supposed to ship containers with goods between different endpoints. As multiple companies are involved in the transportation of the same container, the companies develop a joint strategy and utilize a common process model for which they collaborate with each other. This scenario coincides with our Integration scenario. We are interested in demonstrating that our architectural design fits well with the scenarios already described.

Government

In this scenario the business processes running inside a municipality, which have something to do with the citizens, often depend on case data. For this reason the Dynamic Process Composition module from the global architectural model of ProCPlat is very important in this scenario. Using it, the local municipality officials would be able to adjust a global process defined by an advisory body, which is directly connected to the central government. The executive agency would provide the building blocks for the process in the form of process chunks that can be used by the local municipalities depending on their local legislation. Those process chunks can be defined or uploaded to the ProCPlat using the Process Definition Module. Logically, the Web Enactment and Special Clients module would also be part of this scenario as they provide basic functionality like manual process enactment and user management. On the other side of the Communication Interface the ProCPlat would connect to the on-premise systems run by the municipality. They could be normal BPMS as well as legacy custom software systems running as a single application. It is common nowadays, that municipalities provide digital web services to the citizens. We consider it normal that some of those web applications are actually running in the Cloud. As different municipalities have different budgets and often the similar software is not deployed in a centralized fashion, we believe that municipalities might use different vendors of a ProCPlat. In this case a connection between two municipalities could be made via the two different ProCPlat directly.

Logistics

In this scenario the business processes are connected with the transportation of goods, scheduling, payments, etc. In other words, the complexity of the business processes is high since there is a complicated supply chain. Shipping a single container by multiple parties is a common case in this industry. For this reason, the different parties shipping the same container should communicate in an organized fashion, so that the deadlines connected with the shipment are fulfilled. Currently, there are complicated mechanisms for data interexchange as companies run their local, often legacy, systems. Bringing the parties together and giving them the opportunity to collaborate on specifying a single process definition is vital. In ProCPlat this ability is archived with the Process Definition module that allows for multi-party process definition. Additionally, the Web Enactment and Special Clients modules take part in the process, as they provide common feature for business process management. Similar to the Government scenario, the Logistics scenario needs to utilize different on-premise systems like BPMS and traditional, in-house developed software applications. In this scenario the interconnection with different Cloud application and Other ProCPlat engines is even more vital as the logistics business in heavily dependent on Information Technology.

5. Organizational aspect

In this chapter we continue with the presentation of the models that we propose. We make a transition to the Organization aspect of the Realization dimension. In *Section 5.1* we provide the global organizational model. In *Section 5.2* we focus on the organizational model of each of the four scenarios already defined in *Section 3.3*, so that it is clear which parties can be potential participants in each of the scenarios. As in the previous chapter, we conclude this part of our study with the two running cases that help for the understanding of our models.

5.1. Global model

In this section we present the global model for ProCPlat in the Organizational aspect.

Positioning into the 3D design space

We continue using the three-dimensional design space presented earlier in this study. In *Figure 17* we highlight with orange the position of the model that we discuss in this section.

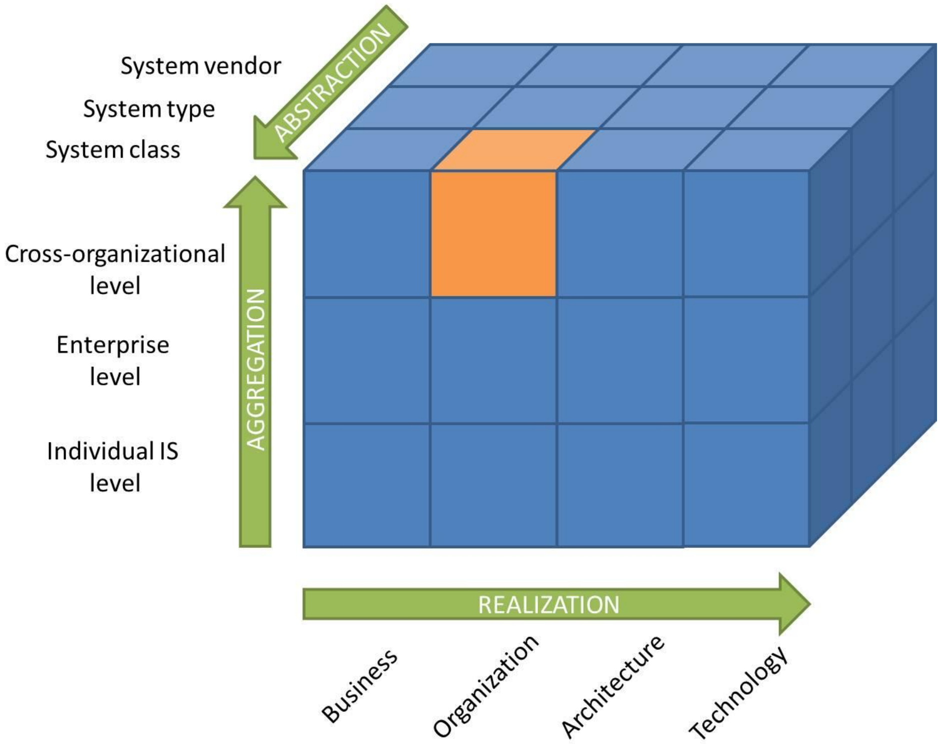

Figure 17: Positioning the global organizational model

We present a model in the Organizational aspect that is again highly abstract and highly aggregated. For this Realization dimension we do not provide other models as we want to focus only on a model with a high level of aggregation and a high level of abstraction.

Global organizational architecture

The ProCPlat global organizational architecture model is shown in *Figure 18*. The two most important and always required components of this model are the *Provider (Intermediary)* and *Consumer 1* (in some scenarios more consumer parties can be required, so we enumerate them from 1 to N). Those two parties form the minimum requirement for a business relationship, where the provider is supplying the consumer with digital services in the Cloud. All other components have a dashed line as a border which represents that they are optional, depending on the scenario that we have at hand. In order to reach this model we had a look at similar artifacts [26] which describe e-Business relationships. If we divide the model vertically, on the left side we position the consumers of services while on the right side we place the providers of services. In our case there can be also provider(s) of knowledge goods as the *Process Definition Providers* at the bottom of our model.

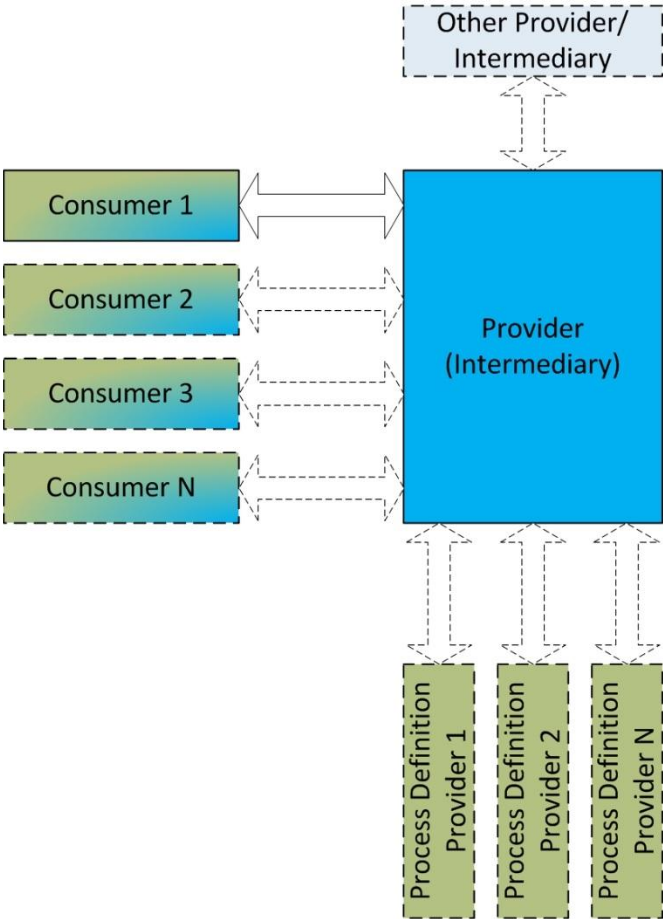

Figure 18: ProCPlat global organizational model

Consumer 1-N. The parties that can act as consumers of the collection of Cloud services offered by the Provider can be of different nature. In the simplest case a company uses the web enactment interface of the ProCPlat to enact existing business processes. Since we target more business-to-business scenarios though, this can be also different on-premise systems like BPMS or generic application software or off-premise Cloud applications. The current Cloud landscape for PaaS provides capabilities for the creation of more complicated data interexchange, so integration with other Cloud services seems normal. Additionally, many enterprises are outsourcing their CRM or ERP systems into the Cloud and the success of Salesforce is good example of it.

Provider (Intermediary). The provider in our organizational model is a single party which in our case is the ProCPlat vendor. It supplies the consumer(s) with Cloud services that help for defining and enacting business processes. In some of the cases the provider can also act as an intermediary when other parties provide process definitions for the ProCPlat or when the platform has to communicate with similar external platforms. In the first case the provider is acting as an enabler of virtual economies, as it creates an ecosystem for exchange and consumption of digital goods. Generally, the Cloud platform provider does not always have to be also a provider of business services. In some cases the provider of business service can be a company which is utilizing a Cloud platform but branding the service as its own (e.g. Pega uses Amazon Web Services to deliver its business services). In our case though, we consider that the two functions under discussion overlap. Still, if ProCPlat is realized in practice and proven to be a success, a scenario where companies use ProCPlat to provide their own business services is also possible.

Process Definition Provider 1-N. This party in our model might be part of a scenario where it provides process definitions as intangible goods. Companies that have in-depth domain knowledge about business processes can create generic process definitions that can help different enterprises to achieve their business goals. The process definitions should only need to be configured in order to fit the needs of the companies. A common example for such a party is PayPal, which provides a way for digital payments in different web platforms.

Other Provider/Intermediary. This party in the organizational model can participate in a scenario where there is an interaction between two Cloud platforms handling business processes. As a lot of research is performed towards the standardization of Cloud Computing, we believe that it would be normal to have a simplified data interexchange between different platforms. This adds even more abstraction as business processes would have to be run using different process execution modules. As we already mentioned IEEE has published a draft specification for Cloud interoperability [57]. Additionally, academic publications also propose ways for Cloud harmonization [47]. Still, solving the security and data governance challenges seems even harder in a heterogeneous environment.

5.2. Relation to the ProCPlat ownership model

In this section we present each of the four scenarios that we formulated in *Section 3.3* in terms of the parties that interact in a specific scenario. Only the components that are needed for the defined scenario are left highlighted. We believe that in reality the general model can be modified to fit also a mixed scenario, which is based on specific business requirements.

Cloud BPO

In *Figure 19* we present the organizational model from the perspective of the Cloud BPO scenario. In this scenario the consumer of the Cloud service is utilizing ready-made business process definitions which can be parameterized depending on the specific business requirements.

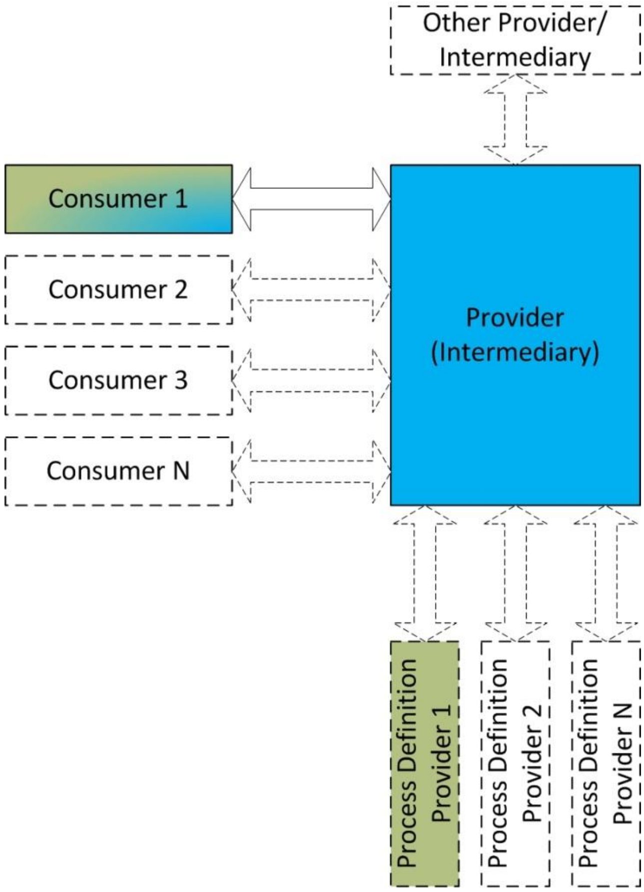

Figure 19: Organizational model Cloud BPO

There are two options for the ownership of the process definitions. First, they can be directly provided by the vendor of the ProCPlat. An example for this is IBM, which offers such kind of Cloud BPO services. Second, the process definitions can be provided by an external party whose main objective is to monetize on their know-how for business processes in a specific domain. In that case the ProCPlat vendor has the function of an intermediary, allowing for the creation of virtual economies. In general, the creation of such ecosystems with clear cost-benefit models provides for a clear and open business relationship between the parties. In this scenario the ProCPlat vendor can have two revenue streams - a bigger one coming from the consumer and a smaller one coming from the process definition providers. After the business process definitions are supplied to the ProCPlat, they are always enacted by the consumer and executed by the vendor of the ProCPlat.

Cloud BPMS

In *Figure 20* the organizational model for the Cloud BPMS scenario is shown. This is the simplest scenario out of the four that we propose as there is only one possible option involved - the consumer is defining or uploading its own business process definitions.

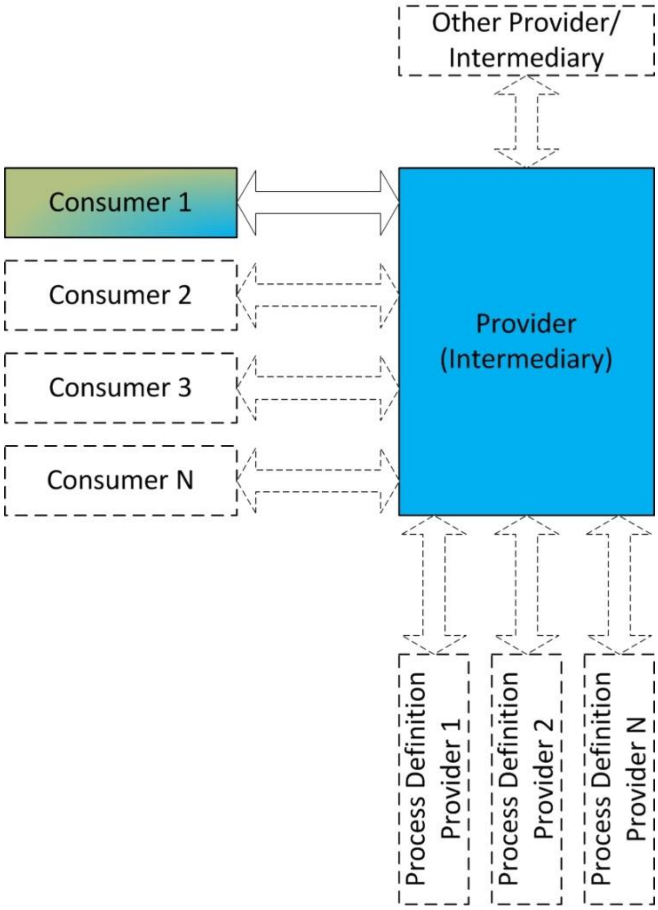

Figure 20: Organizational model Cloud BPMS

As a result the consumer gets standard BPMS functionality which is provided by the ProCPlat. This way the consumer cuts on its capital costs as it follows the most popular SaaS model. For the ProCPlat vendor there is only one stream of revenue as there is only one party involved. We already discussed in *Section 3.7* some of the vendor which provides BPMS as a Cloud service.

Syndication

In *Figure 21* we present the organizational model from the viewpoint of the Syndication scenario. This is one of the two scenarios that currently do not have a commercial solution on the market. It enables a consumer to syndicate (or mash-up) chunks of business processes, so that a business process with custom flow can be enacted at the end.

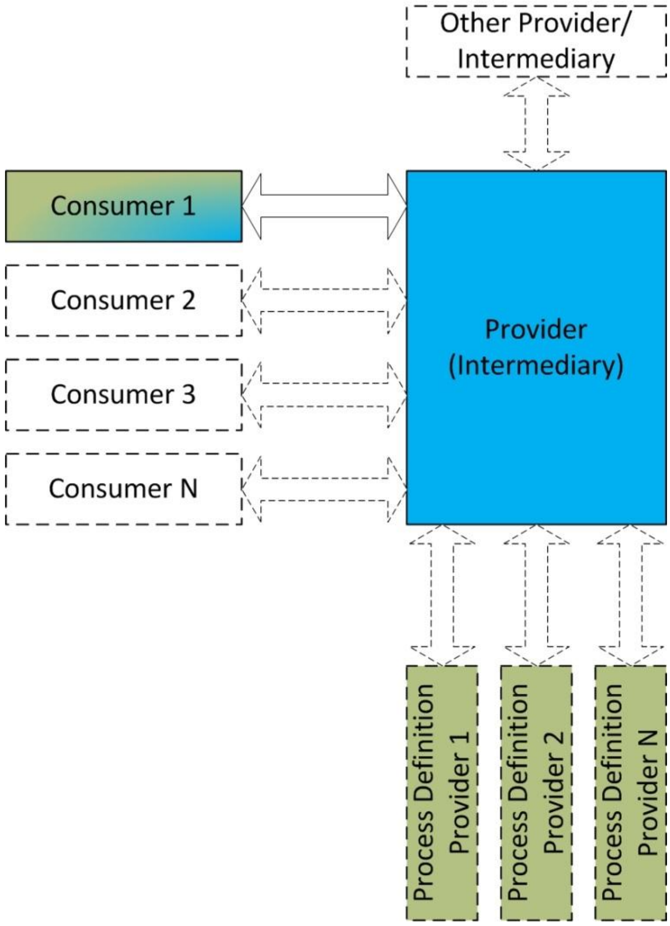

Figure 21: Organizational model Syndication

Similar to the Cloud BPO scenario external parties are able to provide their domain know-how and offer process definitions on demand. Logically, the ProCPlat provider can also offer such process definitions. The difference with the Cloud BPO scenario comes from the fact that the consumer has more freedom in the customization of business processes as new ones can be created in a semi-dynamic fashion. The Syndication scenario is suitable for enterprises which deal with cases which do not have a fixed business process supporting them. Again, as in the first scenario the ProCPlat vendor might create a virtual economy and get two streams of revenue.

Integration

Finally, in *Figure 22* we demonstrate the organizational model regarding the Integration scenario. In this scenario there are multiple consumers of the ProCPlat as the platform is used as tool that orchestrates cross-organizational business processes.

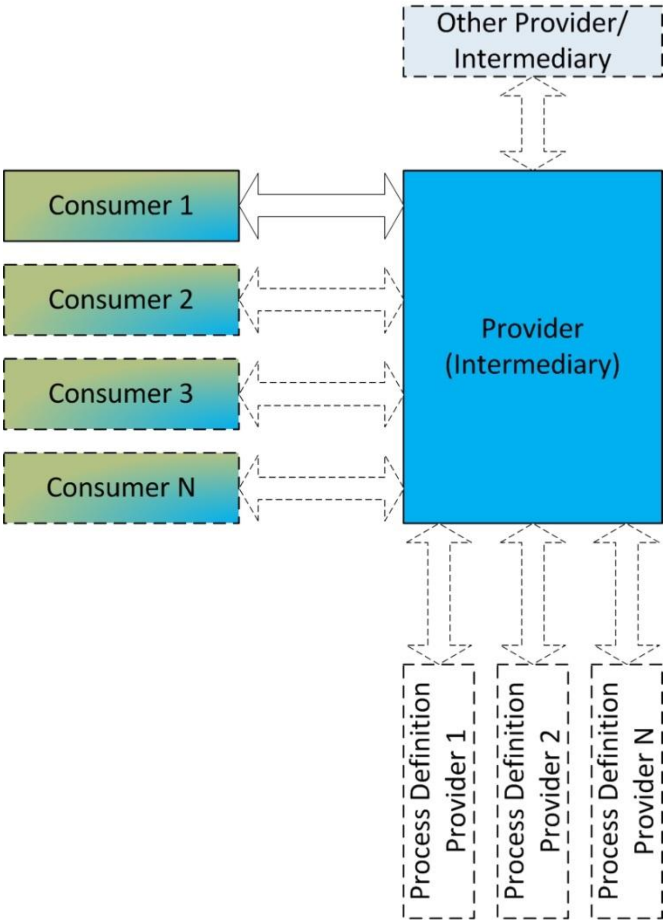

Figure 22: Organizational model Integration

In this scenario parties collaborate and define common business processes that are also connected to the parties' on-premise systems. As we illustrate in the model, there are multiple consumers which can range from on-premise BPMS system to Cloud applications and personnel using the web enactment interface of ProCPlat. Additionally, we envision cases when those consumers would be using ProCPlat as a gateway to connect to business partners which use other platforms similar to ProCPlat. The vendor of the ProCPlat creates a different type of ecosystem where companies get the possibility to collaborate and work together. In the long-run other businesses might see the potential of working in collaboration with other business too.

5.3. Mapping the Architecture aspect to the Organization aspect

In order to produce models of good quality that fit well in our 3D design space, we should be able to map one model to another. A consistent mapping can serve as verification that we model the same concept right but from a different viewpoint. Elements, present in two models that are being compared, which cannot be mapped, imply that there is a problem with one or both of the models. We provide the mapping between the Architecture and the Organization aspects in *Figure 23*.

We have already described the cross-organizational models at the Architecture and Organization aspects. We consider them to have a good match as the elements of the Architecture model (*Figure 14*) can easily be spotted in the Organizational model (*Figure 18*). We keep the coloring of the models consistent, so that the mapping is more easily observable. All modules colored in blue in the

Architecture model are part of the Cloud vendor platform, which handles the execution of business processes. All those modules are mapped to the *Provider* element in our Organizational model. Then, all modules or part of modules that are connected to the consumers of the platform are colored in green. In the Architecture model we have some thin clients which we also consider to be part of the consumer side, so they also translate to the elements on the left-hand side of *Figure 18*. The single modules *Cloud Application* and *Other ProCPlat Engine* in the Architecture model are trivially spot in the Organizational model as they come without any significant change (we consider the *Cloud Application* as a *Consumer* in the Organizational model).

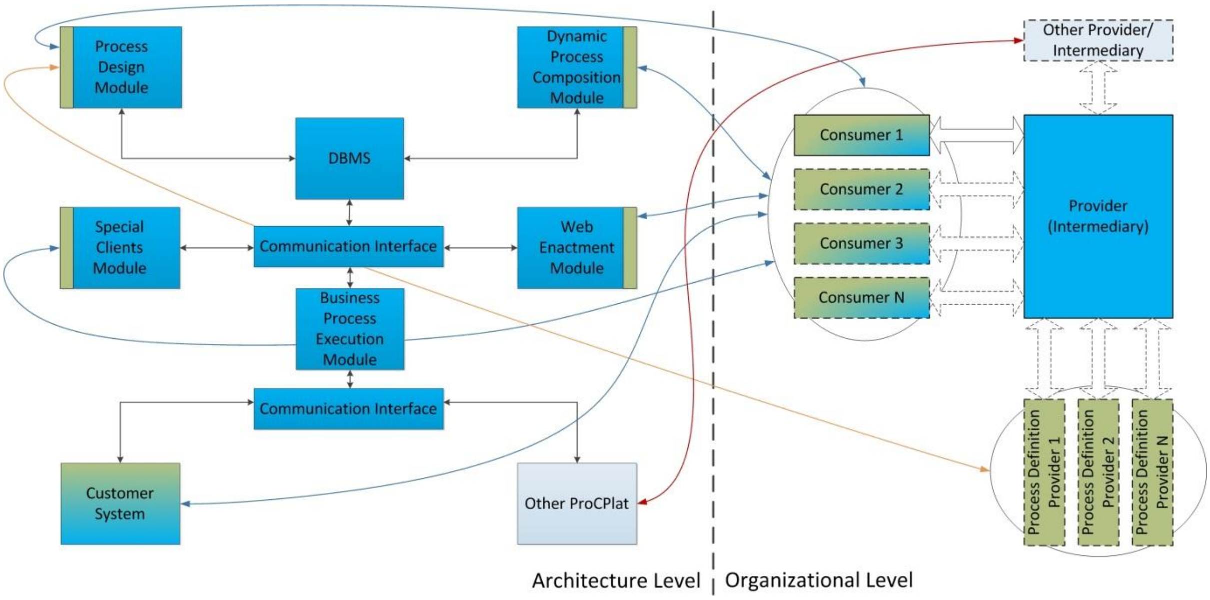

Figure 23: Mapping between Architecture and Organization aspects

5.4. Evaluation through scenarios

In this section we continue with the evaluation through scenarios as we discuss the Organizational model.

Government

As we already mentioned the Government scenario deals with business processes running in local municipalities. As those processes are both specific and confidential, we do not expect that a ProCPlat vendor would generally offer them. This implies that the business processes or actually the business process chunks will be specified by some governmental institutions. This means that they are not always specified by the local municipalities themselves. We envision that in the best case business process chunks are specified at higher level and then parameterized at local level. Still, there are services like garbage collection and cleaning which are done through partnerships with external companies. Those companies can also participate in the definition of processes. A project that is currently running called CoSeLoG (http://www.win.tue.nl/coselog) deals with such kind of municipal processes. Usually municipalities also deal with the issue of permits for different purposes. In order for such permits to be issued other agencies might also participate. Those agencies can provide their own process chunks that interconnect with their own systems when needed. Overall, handling a personal case by the municipality involves the dynamic or semi-dynamic composition of a full-fledged business process out of smaller process chunks, which might be defined by different parties. As this scenario corresponds to the Syndication scenario defined in Section 3.3, there could be multiple process definition providers and one consumer of the ProCPlat.

Logistics

Our second scenario deals with the Logistics sector and the collaboration of companies on the delivery of shipping containers. As we have indicated in *Figure 22* we have multiple parties labeled as consumers that are connected to the ProCPlat. Those parties use the platform as an orchestration mechanism instead of having orchestration and data interexchange between each single party. The consumers do not have to be of similar nature. We expect that there is human input via the web interface of ProCPlat but also other Cloud applications, as well as BPM, ERP and CRM systems. Using the ProCPlat the involved businesses are also focusing less on implementing complicated IT infrastructures and integration mechanisms. In this way they can outsource partially their IT tasks in the Cloud. The companies can then focus on their actual business of shipping goods and shift their capital investments into activities that bring a direct return on investment. This scenario corresponds to the already-defined Integration scenario which implies that there should be multiple consumers of the services provided by ProCPlat. Additionally, connections to other ProCPlat implementations should be possible.

6. Business aspect

In this chapter we make the transition from the Organizational aspect to the Business aspect of the Realization dimension. In *Section 6.1* we present the relation between four basic business-relevant qualities and the Cloud Computing characteristics defined by NIST [14]. Then, we use those qualities to make a comparison between on-premise BPMS with ProCPlat. In *Section 6.3* we provide the mappings between the Organizational and the Business aspects. We conclude the chapter with the evaluation scenarios for the government and logistics sectors.

6.1. Business-relevant quality dimensions

In this section we present the business-relevant qualities that are important to be discussed in terms of their relation to the standard Cloud Computing characteristics defined by NIST [14].

Positioning into the 3D design space

In this chapter we continue using the three-dimensional design space presented earlier in this study. In *Figure 24* we highlight with orange the position of the artifacts that we discuss in this section.

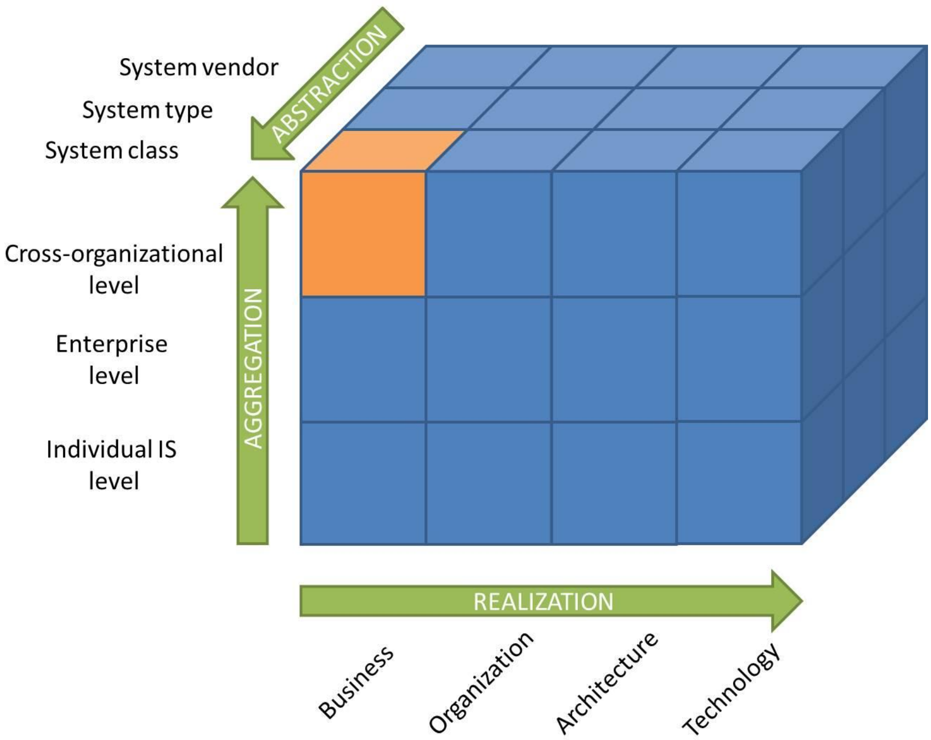

Figure 24: Positioning the business model

We consider that it is hard to provide a generalized model for the Business aspect which fits all existing industries and types of business. For this reason, we provide a relation matrix which is relevant for the analysis of the Business aspect. We consider the most interesting thing to discuss to be a collection of business-relevant qualities and we link those to the Organization aspect.

Linking business-relevant qualities with Cloud Computing characteristics

In order to discuss the Business aspect in terms of cross-organizational behavior, we use three elements: business chains, business drivers [26] and business objectives. Two of those elements are taken from literature and we consider them to be suitable for our study too. We regard the third element – business objectives –as important because it is connected to the strategy of an enterprise which is connected to its business models.

We can identify two forms of restructuring the business chains between organizations - disintermediation and reintermediation [26]. Those two forms are concerned with either removing or adding parties from a business chain. Both forms are applicable to our proposition for ProCPlat. In the one case the introduction of ProCPlat is reducing the number of parties connected in a business chain or eliminating parties as in the Integration scenario (*Figure 22*). In such a case we can consider ProCPlat to be a disintermediator. At the same time, ProCPlat can be considered as a new party to a supply chain. In the Syndication scenario (*Figure 21*) ProCPlat acts as a link between consumers of business processes and providers of business process definitions. This can be viewed as reintermediation and it fits in the typical example of creating a virtual marketplace [26]. In our case the intangible products at the marketplace are business process definitions. This could be regarded as an important factor for the actual building of ProCPlat, as this would allow a lot of companies to consume non-standard business processes.

In order to achieve such new business chains, enterprises need to have reasons to do such kind of a switch to a Cloud-based platform for their business processes. For this reason we introduce different business drivers that can lead to this switch. Usually the business drivers are connected to strategy, finance, marketing, etc. Nevertheless, we choose for an approach in which we also add technology-related drivers to the business ones. Since we design a system, we use the term 'quality' to represent the different business drivers as we do not regard the term 'driver' to be normally coupled with system abilities. As technology drivers we use three widely-accepted system qualities [58] in addition to two Cloud-related capabilities that we consider important. Separately, we use four non-system qualities which are vital for conducting any type of business with services. We choose those based on our experience. Finally, we relate the whole set of nine qualities to the five NIST Cloud Computing characteristics [14]. We visualize this relation in *Figure 25* where we use the symbol "X" whenever there is a positive relation between a business-relevant quality and a Cloud Computing characteristic, i.e., the characteristic is enabling or improving the quality.

Business-relevant qualities		Cloud Computing Characteristics [14]				
		Self-service	Network Access	Resource Pooling	Rapid Elasticity	Measured Service
System Qualities	Interoperability		X			
	Security		X	X		
	Maintainability	X				
	Scalability			X	X	X
	Accessibility		X			
Non-System Qualities	Core Competency	X	X			
	Financial Efficiency	X		X	X	X
	Speed-to-Market				X	
	Business Flexibility	X			X	

Figure 25: Relating business-relevant quality dimensions to NIST Cloud characteristics

As we investigate the cross-organizational business aspect, we need to mention that this relation is important for the parties acting as consumers of ProCPlat. We consider that the non-system business-relevant qualities also hold for the actual vendors of ProCPlat. In order to illustrate how the relation between business-relevant qualities and Cloud Computing characteristics work, we provide two examples: one for the system qualities and one for the non-system qualities from the perspective of the consumer.

Our first example is concerned with scalability. According to our matrix in *Figure 25* it relates to resource pooling, rapid elasticity and measured service. We already described in depth what elasticity and resource pooling (multi-tenancy) are in *Section 2.2*. In order for a business to utilize the elasticity of a Cloud platform there need to be scalability mechanisms in place. Additionally, multi-tenancy helps scalability in a way that when one consumer releases computational resources, another one can take over the same resources. Finally, measuring the service utilization by the platform vendor and providing this information to the consumer is important because decision concerning scalability can be taken based on the service utilization data.

Our second example goes over one of the non-system qualities - speed-to-market. We relate this quality to only one Cloud characteristic, namely the rapid elasticity. We consider that the relation is obvious but it is very important from business point of view. As some enterprises are growing faster, they want to be able to deliver their products or services in an ample amount of time. In a traditional scenario buying and provisioning new hardware can take months, while scaling out a Cloud service is possible in some minutes. This property is very important for service start-ups which experience a rapid growth of user and need to handle the heavy load of the system. Additionally, start-ups can improve their speed-to-market by the actual utilization and syndication of ready processes from the marketplace. We regard this perspective as important from operational point of view.

After discussing the business drivers, we make another small step towards business objectives which is slightly touching on another aspect that we do not explicitly define - the Strategy aspect. We propose two objectives for both the vendors and the consumers of ProCPlat. First and most common objective for a vendor is to make revenue. This is easily linked to the strategic goal of distributing dividends among the shareholders of the Cloud provider. A second business objective is the creation of an ecosystem of consumers. Such an objective is related at the strategic level to the achievement of long-term growth. From the perspective of a consumer, one business objective is to increase the automation of the business which we relate to the enabling and improvement of collaboration with channel partners. A second business objective for the consumers of ProCPlat is to improve the service of customers, for example resolving a case with fewer documents. This objective can be related at the strategic level to the establishment of long-term relationship with the clients of the company.

6.2. Comparison between on-premise BPMS and ProCPlat

We consider that it is important to extend on the business-relevant qualities presented in the previous section as they are a key element in the actual decision making process for a business when the business considers the utilization of a Cloud platform like ProCPlat. In order for such decision to be taken, it is usual that feasibility studies are performed. In the case we are discussing, it is most reasonable to compare ProCPlat with a standard on-premise BPMS. In order to provide a good visual representation for the comparison between on-premise BPMS and ProCPlat, we use the radar chart shown in *Figure 26*.

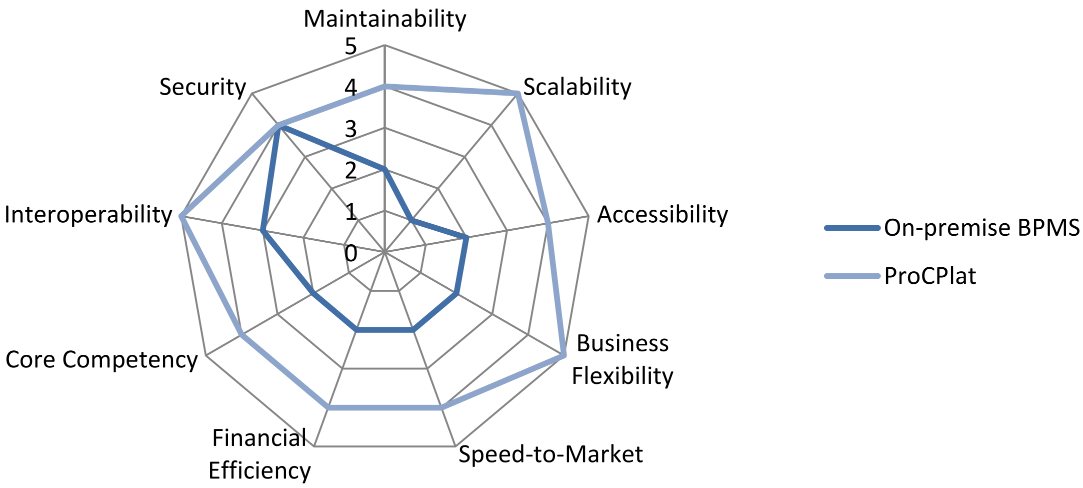

Figure 26: Business-relevant dimensions comparison

We rate each of the nine qualities defined in the previous section from zero to five. Our rating is not based on statistical data or another kind of empirical research since such a procedure is lengthy and requires a lot of resource. The rating is based on our practical experience with Cloud platforms and BPM. The big gaps between on-premise BPMS and ProCPlat indicate the place where we consider that ProCPlat has substantial advantages over the usage of established systems. We illustrate this with two examples - one from the upper half of the radar chart concerning the system qualities and one from the bottom half dealing with non-system qualities.

Our first example is concerned with security, which is often the most important system quality for big enterprises. We consider that migrating the execution of business processes into the Cloud should neither impair nor improve the security aspect of a system. We already discussed in *Section 2.2* the multi-tenancy characteristic of Clouds which is usually considered cumbersome by business. Still, the new technologies that are developed provide sufficient level of isolation between the different tenants of a Cloud platform.

The second example is connected to the financial efficiency. We consider that migrating the execution of business processes into the Cloud can severely impact the spending of companies. Firstly, the capital investments in hardware which concern the execution of business processes can be eliminated since the companies does not need to provision expensive servers by itself. Secondly, the cost on personnel involved in the deployment and administration of BPMS should be reduced. This spending bucket is not supposed to disappear completely as staff can be retrained for tasks which can be performed in the Cloud.

6.3. Mapping the Organization aspect to the Business aspect

After discussing the mapping between the Architecture and the Organization aspects in the previous chapter, we move on to the next possible mapping - between the Organization and Business aspects. For this mapping we provide a visual representation in *Figure 27*.

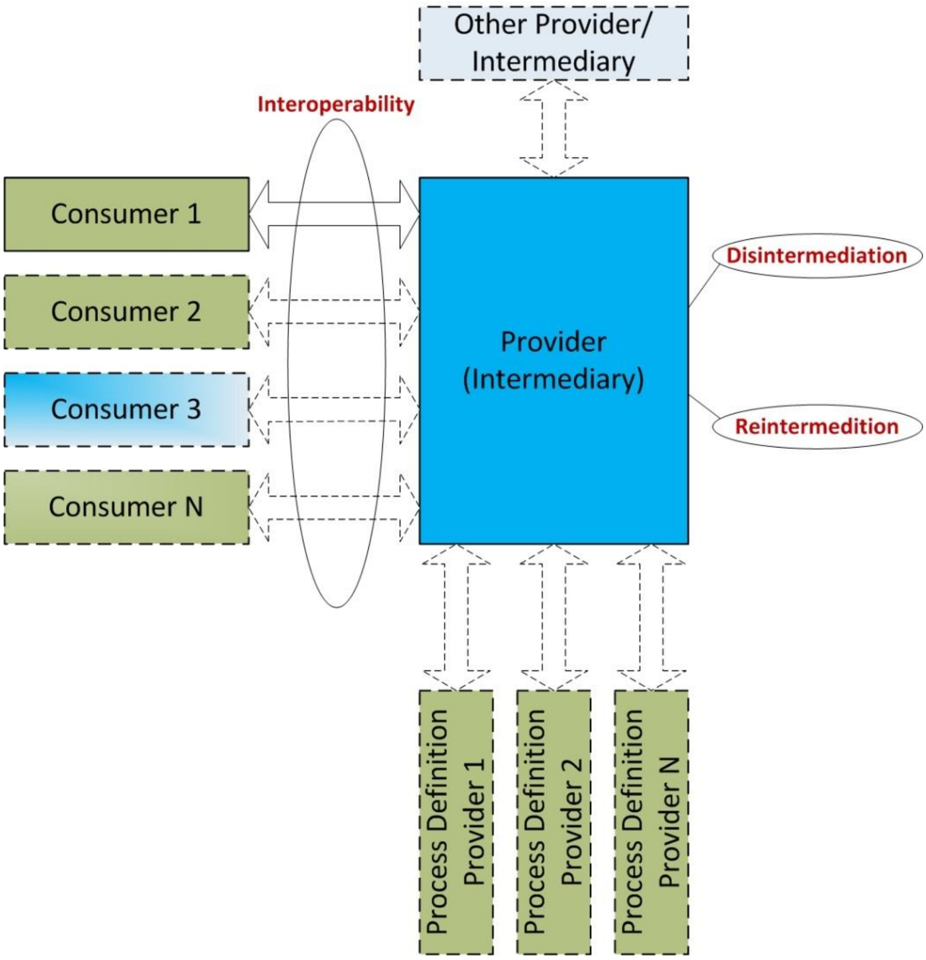

Figure 27: Mapping between Organization and Business aspects

As a basis we use the global organizational model that we described in *Section 5.1*. We relate some of its elements to our discussion about the business-relevant quality dimensions in this chapter. We do not cover all elements from the two aspects as this is not feasible. Models for a specific case are needed in order to make a concrete mapping.

Firstly, we can relate the global organizational model to interoperability, which is one of the system business-relevant qualities. As ProCPlat enables the collaboration between parties in a Cloud ecosystem, the connections created in between the platform and the parties enable interoperability. Through the definition of standard schemas at the Cloud vendor side, parties are able to communicate in a simplified way.

Secondly, we can associate the *Provider* element from the organizational model to disintermediation and reintermediation, which are the two forms of restructuring of the business chains between organizations. As we already mentioned, the ProCPlat vendor can eliminate some systems on the one hand, or serve as an additional node in the business chain on another hand.

6.4. Evaluation through scenarios

In this section we present the evaluation through scenarios and logically the target aspect is the one that we discuss in the current chapter – the Business aspect.

Government

The scenario concerning governmental institutions deals with the processing of cases at the local municipalities. Business models for those institutions can be specified both at local and at regional or national level. The business chains in this scenario might involve the collaboration between different governmental agencies. The utilization of ProCPlat would be a new way of mediation between those parties which adds a new party to the ecosystem, so this can be categorized as reintermediation. We would like to discuss three of the presented business drivers in relation to this scenario. Firstly, the scalability is important as the government tries to provide more electronic services. As the older population is getting experienced with Information Technologies, more citizens should use the digital municipal services. Secondly, the interoperability driver is a good reason for a government to utilize a platform like ProCPlat. Interconnecting services is a costly procedure and a common platform will bring benefits. Finally, the financial efficiency should be the most important for the central government and local municipalities. ProCPlat should substantially reduce the costs for providing digital services.

Logistics

The second evaluation scenario takes the logistics sector into consideration. The different companies which take part in this industry all have different business models, since they do not have a supervisory body. The usage of ProCPlat should enable both types of restructuring of business chain – disintermediation and reintermediation. Since ProCPlat can act as an orchestrator of business processes, a lot of the one-to-one interconnections between the different companies should disappear. Still, the ProCPlat itself is another party added to the business chain. In general, the companies should benefit of ProCPlat as it would greatly reduce number of interconnections in the complicated business network of the logistics business. This is also related to one of the business drivers – interoperability. In addition to it, accessibility is another driver that is vital to this industry sector. As transportation companies use many portable devices and travel in different locations, the platform which serves them should be easily accessible from everywhere. The Cloud is a perfect solution for getting access to a resource in almost all physical locations around the world. Finally, the adoption of ProCPlat can be driven by the opportunity to focus on core competences. As logistic companies ship goods, Information Technology should not be their main responsibility. Outsourcing this task to the Cloud should create more time to focus on the core business.

7. Evaluation through analysis

In addition to the two running cases that revolve around the government and logistics scenarios (descriptive evaluation), we decided to utilize a second approach for evaluation in this chapter – analytical evaluation. In our initial planning this step was not defined. Still, we consider that it brings more consistency to our research, as it tests the artifacts produced of study against the opinions of industry specialists.

7.1. Procedure

This evaluation approach is also proposed in the IS Research Framework [1]. It is referred to as an analytical method with four options (static analysis, architecture analysis, optimization, dynamic analysis). For our study we choose two of those options, namely static analysis and architecture analysis. The way of performing this analysis is through interviews with experts. We spoke to four architects working within Capgemini, who we interviewed separately for 45 minutes per person. Our goal was to get a general idea if our constructs and models are complete and if they fit well into the current IT landscape. We do not target to make a thorough quantitative analysis as this would go out of the timeframe of the study; therefore we perform a concise qualitative analysis.

In order to utilize the analytical evaluation method, we had to choose some system qualities for the static analysis part. To achieve this, we surveyed the literature for taxonomy of different systems qualities. Substantial research has been performed in this area [58], [59] and we managed to select a group of system qualities. We reduced this set of qualities, so that we produce a feasible-for-an-interview questionnaire. We use the static analysis with the qualities of completeness and interoperability over the construct and models that we have created. For the architecture analysis we use the fit quality to evaluate how good is the relationship between ProCPlat and both the organization and the IT landscape.

We formulated five questions which target the evaluation of our design artifacts, which are listed in Appendix B. In order to get a better feedback, we utilized a blind evaluation approach for the static analysis part (the first three questions of the questionnaire) which means that instead of presenting the ready designs, we asked about the expectations of the experts for such designs. In this way we can compare the expectations of industry professionals with the ProCPlat design and see where the overlaps are. Before proceeding with the architecture analysis, we presented the construct and the designs in the Architecture and the Organizational aspects, so that we get more relevant answers to questions concerning fit. In the following section we discuss the results of the interviews.

7.2. Discussion of the results

We performed four interviews with experts from Capgemini. This accounts for about 35% of the total interviews that we requested. We expected a low turnover in the beginning and for this reason we choose to contact larger amount of people. We provide the raw results of the interview in Appendix C. For each of the interviews we provide the functional position of the interviewed expert. Overall, we are quite satisfied with the interviews, which confirm big part of the research performed. We find some points which have not been explored but those are not concerned with any critical modules. The possible points for improvement, that we consider relevant, are discussed in the section of our study which deals with future work at the end of the next chapter. As we pointed out already we conducted the first part of the interview in a way that we did not provide our respondents with an idea regarding our design, so that we do not frame their thinking. This turned out to be a successful way to conduct the interviews.

Question 1: Completeness of the construct (ProCPlat definition in *Section 3.1*)

Our definition coincides with the opinions of the respondents if combined together. A point that nobody mentions is concerned with explicitly stating that people should have the ability to participate in the business processes. We believe that this is due to the fact that the respondents think in a technical direction when they reason about architecture. A point which we considered to be a good compliment to the definition is the ability of the platform to support testing and deployment of business processes in addition to the definition and enactment. Two of the respondents also mention that the platform should support end-to-end business processes which we cover with the addition of horizontal and vertical orientation of the business processes inside the enterprise.

Question 2: Completeness of the architecture

Again, as in the previous question, the combined set of responses coincides well with our architectural design. The modules that we have identified are well-supported by the interviews. One possible point of extension which we also find interesting is the inclusion of a module or sub-module which deals explicitly with business rules. This module could be either an additional module which serves a couple of business process engine instances at a time or it could be integrated with the actual business process engine. A module that is not mentioned by any of the respondents is a module dealing with the billing. Such a module is not typical for an on-premise implementation of a BPM system, but in our opinion it is a vital part of a Cloud platform not only for the standard users of the platform but also for the ones that are exploring the virtual economy created in the Cloud.

Question 3: Interoperability of the architecture

The feedback received regarding this question clearly indicated that the set of external systems to which ProCPlat should connect is exhaustive. All respondents mentioned that ProCPlat should be able to connect to on-premise applications such as CRM and ERP systems or legacy applications. The interconnection using Enterprise Service Bus was also pointed out, so we can claim that the selected integration mechanism is a reasonable choice. The integration with other platforms similar to ProCPlat was not explicitly mentioned but we do not consider this a big drawback, since currently there is no integration between existing Cloud platforms.

Question 4: Fit with the organization

The reactions to the questions connected to the organizational fit of ProCPlat were less positive compared to the previous three questions. In general, the respondents see a fit between the proposed platform and the business requirements. Unfortunately, the predominance of the businesses is still reluctant to accept innovations when it comes to the usage of the Cloud. This fear is mostly connected to securing data which goes out of the corporate firewall. Additionally, the platform requires the presence of well-structured data which is not always available. Some significant resources should be invested in order to structure the company data and its processes.

Question 5: Fit with the IT landscape

The question about the fit with the current landscape is most important when it comes to the actual implementation of a platform like ProCPlat. According to our respondents there is a fit with the current IT market since the technology to solve business problem in such a way is present. Still, the overall opinion is that it is still too soon for the introduction of such a platform. The Cloud market is still quite fragmented and making interconnections at business process level is unrealistic. Based on those opinions we can claim that conceptually the proposition for ProCPlat is viable but the problems lay in the making of a business decision to invest in the development of such a platform.

8. Conclusions and future work

In this chapter, we summarize the results of the work accomplished during the study. *Section 8.1* summarizes the findings in view of the research goal that was set out in the introduction chapter. *Section 8.2* proposed some possible directions for future work. We conclude the chapter with a reflection on the product of this study and the process utilized during the study.

8.1. Conclusions

The concept of Cloud Computing has been seriously promoted during the last few years. Together with it, different flavors of process technologies have been pushed to the market too, although processes have been present in the application logic for many years. Unfortunately, the application logic is still full of custom business processes that do not get the support which BPM technology provides. When such applications are ported to the Cloud the problems persist instead of being solved.

The goal of this study was to design an architecture that could support business processes in the Cloud. Based on our literature review we can claim that there are already multiple ways already to get such kind of business process support but it is still somehow limited. We based our definition for a Business Process Cloud Platform (ProCPlat) on the existing definition for Business-Process-as-a-Service (BPaaS), which we first considered for extension. Then, we made the decision to categorize business processes according to the parties which can define and enact them. This turned out to be of vital importance for our study, as it changed the position of our platform in the Cloud stack and from there the actual definition. Still, BPaaS played a significant role as a driver which brought us to our final definition for ProCPlat. We managed to find industry examples to support our categorization of business processes too. We believe that supporting our models with scenarios and cases from the industry makes our claims more relevant for the actual business world.

Upon the definition of ProCPlat we designed a multi-aspect reference architecture with three aspects: Architecture, Organization and Business. In order to achieve this we used existing reference architectures to get inspiration from and then tuned the already-defined modules according to the business requirements at hand. Additionally, we added new modules which have not been observed before as we believe that they can bring significant advantages from using our ProCPlat design. Again, all designs were supported by running scenarios, so that we show the relevance of our work. We cannot claim that our architecture considers all possible cases connected to business process in the Cloud. We categorized processes according to the owner of the definition and enactment and we believe to have covered well the process landscape. A proof for this is the second evaluation approach that we used which was based on interviews with professionals. Based on their expertise, they were suggesting small improvements which can definitely improve our work. Still, their feedback was positive which can indicate that our designs can be easily understood and the business value of such a platform is visible. The way in which we conducted the interviews also had an impact on the results. We did not frame the thinking of our respondents with the design that we made, and this turned out to be successful.

We designed our models starting from the Architecture aspect and finishing at the Business aspect. Usually the business requirements are the main driver for the development of a project and everything is aligned with them. Even though we took a reverse approach, we still managed to align the Architecture aspect to the Business aspect via the Organizational aspect. This makes us claim that for an innovation project like ours, the design and modeling can be reversed. At the end we managed to connect non-system-oriented business drivers like business like financial efficiency and core competence to our proposition for a Business Process Cloud Platform.

8.2. Future work

During the course of this study we noticed possible extension points that can be additionally tackled. Also, the input that we received from the Capgemini experts provided us with additional ideas. As we narrowed the scope of the project in the very beginning, we do not consider that there are serious omissions in the project. Also, we are aware that such kind of innovative design project could not be evaluated in a good way using traditional empirical techniques. Underneath we present the possible future work.

During the design process we utilized a strict research methodology which is well-backed by the community. Following it guarantees a decent result if the methodology is executed correctly. Still, a very good extension point is to apply architecture principles for the design of our models. This should act as another safeguard against unreasonable design decisions. Since this study did not involve architecture modeling by more than one person, the utilization of guidelines is not required, but still we considered that it can improve the quality of the end product.

A second possible extension point is to extend our reference architecture to two other modeling aspects – Strategy and Technology. This would make the reference architecture complete as it traces everything from the long-term business ideas to the actual technological implementation. We believe that creating model at Strategy aspect is a hard task as it is not reasonable to model a generic strategy as one size does not fit all businesses and business domains. On the other hand, modeling the architecture in the Technology aspect is very costly as it would involve a lot of research in specific technology implementations. In comparison with the Strategy aspect, it is reasonable to have such an architecture and a possible development of a ProCPlat would require it.

Next, possible future work can be done on the security and data governance of ProCPlat. As those concepts are elaborated enough, similar sized studies can be done on each of them. Both topics are relevant for the actual industry as two of the main issues in the cross-organizational business relations are protection from malicious attacks and access to corporate data. At a more technical level an interesting task is the development of standards for interexchange of process data. As we already mentioned IEEE has proposed a draft standard for Cloud interoperability, so working in this direction is important for the better functioning of the business.

Finally, our work can be additionally evaluated with experts from other companies or scientists from other universities. This would bring more new ideas and possible points for rearchitecting of our proposition.

8.3. Reflection

It is important to reflect on the process that was applied in order to realize this study. The gathered experience can serve as a guideline for better execution of future projects. In the beginning of the study there was no research methodology adopted, so the findings were not following a logical line of thinking. The utilization of the IS Research Framework [1] was of great help for the design phase. It put an order in our work which also made it more logical and well-structure to read.

Another important point is setting the goal of the project and the objectives right from the beginning of the study. We agree that working on an innovative project includes a trial and error process in order to find the right problem, but without setting the right objective at the beginning the search process can take too long time to be performed.

As the project required a lot of collaboration with industry specialists and finding already-developed projects as case studies, it could have been better to perform this process in a faster pace. Getting data from the industry is challenging and waiting for people's input leads to loss of time over the course of the project. Additionally, in the beginning of the modeling phase there were made numerous sketches and brainstorm diagrams which were not modeled and described digitally and in writing. This slowed down the design process as decisions were not taken in ample amount of time, as there were no fixed artifacts to be put through a revision process.

Summarizing our observations regarding the project process, we can claim that setting an objective and choosing a methodology at the beginning of the study is of vital importance. Additionally, improving the communication with the industry and having more confidence during the design process would give some acceleration to such type of a project.

Bibliography

[1] Alan Hevner, Salvatore March, Jinsoo Park, and Sudha Ram, "Design Science in Information Systems Research," *MIS Quarterly*, vol. 28, no. 1, pp. 75-105, March 2004.

[2] T. Davenport and J. Short, "The New Industrial Engineering: Information Technology And Business Process Redesign," *Sloan Management Review*, vol. 31, no. 4, pp. 11-27, Summer 1990.

[3] Workflow Management Coalition, "Terminology & Glossary," Specification WFMC-TC-1011, 1999.

[4] W.M.P. van der Aalst and K.M. van Hee, *Workflow management : models, methods, and systems*.: MIT Press, 2002.

[5] H.A. Reijers, *Design and control of workflow processes : business process management for the service industry*.: Springer, 2003.

[6] Workflow Management Coalition, "The Workflow Reference Model," Specification TC00-1003, 1995.

[7] P. Grefen and R. Remmerts de Vries, "A Reference Architecture for Workflow Management Systems," *Data & Knowledge Engineering*, vol. 27, no. 1, pp. 31-57, 1998.

[8] W.M.P. van der Aalst, M. Weske, and D. Grunbauer, "Case Handling: A New Paradigm for Business Process Support," *Data and Knowledge Engineering*, vol. 53, no. 2, pp. 129-162, 2005.

[9] D. Karagiannis, "BPMS: Business Process Management Systems," *ACM SIGOIS Bulletin*, vol. 16, no. 1, pp. 10-13, 1995.

[10] H.A. Reijers, "Implementing BPM systems: the role of process orientation," *Business Process Management Journal*, vol. 12, no. 4, pp. 389-409, 2006.

[11] University of California at Berkeley, "Above the Clouds: A Berkeley View of Cloud Computing," Department of Electrical Engineering and Computer Sciences, UCB/EECS-2009-28, 2009.

[12] R. Buyya, C.S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for," *Future Generation Computer Systems*, vol. 25, pp. 599-616, 2009.

[13] I. Foster, "Cloud Computing and Grid Computing 360-Degree Compared," in *Grid Computing Environments Workshop*, 2008, pp. 1-10.

[14] National Institute of Standards and Technology, "The NIST definition of Cloud Computing," National Institute of Standards and Technology, Special Publication 2011.

[15] The Open Group, *Cloud Computing for Business*.: Van Haren Publishing, 2011.

[16] A. Lenk, M. Klems, J. Nimis, S. Tai, and T. Sandholm, "What's inside the Cloud? An architectural map of the Cloud landscape," in *CLOUD '09 Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing*, 2009, pp. 23-31.

[17] Gartner, "PaaS Road Map: A Continent Emerging," Gartner, 2011.

[18] Amazon Web Services, "Architecting for the Cloud: Best Practices," Whitepaper 2011.

[19] P. Bell, "Architecting applications for the cloud," IBM Corporation, Whitepaper 2011.

[20] D. Pritchett, "BASE - An ACID Alternative," *ACQ Queue*, pp. 48-55, May/June 2008.

[21] C.J. Guo, W. Sun, Y. Huang, Z.H. Wang, and B. Gao, "A Framework for Native Multi-tenancy Application Development and Management," in *The 9th IEEE Internation Conference on E-Commerce Technology and the 4th IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services*, 2007, pp. 551-558.

[22] M. Pathirage, S. Perera, I. Kumara, and S. Weerawarana, "A Multi-tenant Architecture for Business Process Executions," in *IEEE Conference on Web Services*, 2011, pp. 121-128.

[23] R. Mietzner, T. Unger, R. Titze, and F. Leymann, "Combining Different Multi-Tenancy Patterns in Service-Oriented Applications," in *IEEE International Enterprise Distributed Object*

Computing Conference, 2009, pp. 131-140.

[24] IEEE, "IEEE Recommended Practice for Architectural Description of Software-Intensive Systems," Standard IEEE Std 1471-2000, 2000.

[25] P. Grefen. (2012, Summer) Business Information System Architecture. Document.

[26] P. Grefen, *Mastering e-Business*. New York, USA: Routhledge, 2010.

[27] D. Greefhorst and E. Proper, *Architecture Principles: The Cornerstones of Enterprise Architecture*.: Springer, 2011.

[28] P. Kruchten, "Architectural Blueprints - The "4+1" View Model of Software Architecture," *IEEE Software*, vol. 12, no. 6, pp. 42-50, 1995.

[29] The Open Group. (2009) TOGAF 9.1: Definitions. [Online]. http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap03.html#tag_03_27

[30] S. Angelov, J. Trienekens, and P. Grefen, "Towards a Method for the Evaluation of Reference Architectures: Experiences from a Case," in *2nd European Conference on Software Architecture*, 2008, pp. 225-240.

[31] T. Elzinga, J. van der Vlies, and L. Smiers, *The CORA Model*.: Sdu Uitgevers, 2010.

[32] ICT Uitvoeringsorganisatie. (2012) E-Overheid. [Online]. http://www.e-overheid.nl/onderwerpen/e-overheid/architectuur/nora-familie/nora

[33] Microsoft Corporation. (2009) Microsoft Application Architecture Guide. Document.

[34] Het Waterschapshuis. (2012) E-Overheid. [Online]. http://www.e-overheid.nl/onderwerpen/e-overheid/architectuur/nora-familie/wilma

[35] M. Lankhorst, *Enterprise Architecture at Work*, 2nd ed.: Springer, 2009.

[36] Zachman International. (2008) The Zachman Framework. [Online]. http://www.zachman.com/about-the-zachman-framework

[37] The Open Group. (2012) TOGAF. [Online]. http://www.opengroup.org/togaf/

[38] J. van't Wout, M. Waage, H. Hartman, M. Stahlecker, and A. Hofman, *The Integrated Architecture Framework Explained*.: Springer, 2010.

[39] IBM. (2011, February) IBM Cloud Computing Reference Architecture 2.0. Document.

[40] L. Zhang and Q. Zhou, "CCOA: Cloud Computing Open Architecture," in *2009 IEEE International Conference on Web Services*, 2009, pp. 607-616.

[41] National Institute of Standards and Technology, "NIST Cloud Computing Reference Architecture," Special Publication 2011.

[42] W. Tsai, X. Sun, and J. Balasooriya, "Service-Oriented Cloud Computing Architecture," in *2010 Seventh International Conference on Information Technology*, 2010, pp. 684 - 689.

[43] B. Rochwerger and D. Breitgand, "The reservoir model and architecture for open federated cloud computing," *IBM Journal of Research and Development*, vol. 53, no. 4, pp. 535-545, July 2009.

[44] C Vecchiola, X. Chu, and R. Buyya, "Aneka: A Software Platform for.NET-based Cloud Computing," *CoRR*, July 2009.

[45] P. Fingar. (2010, April) Enterprise as a Service (EaaS) – That's where BPM. Document.

[46] R. Accorsi, "Business Process as a Service: Chances for Remote Auditing," in *Workshop Proceedings of the 35th Annual IEEE International Computer Software and Applications Conference*, 2011, pp. 398-403.

[47] M. Papazoglou and W. van den Heuvel, "Blueprinting the Cloud," *IEEE Internet Computing*, vol. 15, no. 6, pp. 74-79, Nov.-Dec. 2011.

[48] A. Norta, "A Hub Architecture for Service Ecosystems: Towards Business-to-Business Automation with an Ontology-Enabled Collaboration Platform," in *Proceedings of the 6. International Conference on Web Information Systems and Technology (WEBIST) 2010*, 2010, pp. 240-243.

[49] S. Grivas, T. Kumar, and H. Wache, "Cloud Broker: Bringing Intelligence into the Cloud," in *Proceedings of the 2010 IEEE 3rd International Conference on Cloud Computing*, 2010, pp. 544--545.

[50] P. Grefen, R. Eshuis, N. Mehandjiev, and G. Weichhart, "Internet-Based Support for Process-Oriented Instant Virtual Enterprises," *IEEE Internet Computing*, vol. 13, no. 6, pp. 65-73, 2009.

[51] OASIS. OASIS UDDI Specification. [Online]. https://www.oasis-open.org/committees/uddi-spec/faq.php

[52] W.M.P. van der Aalst, "Configurable Services in the Cloud: Supporting Variability While Enabling Cross-Organizational Process Mining," in *OTM Federated Conferences, 18th International Conference on Cooperative Information Systems (CoopIS 2010)*, vol. 6426, 2010, pp. 8-25.

[53] P. Grefen, N. Mehandjiev, G. Kouvas, G. Weichhart, and R. Eshuis, "Dynamic business network process management in instant virtual enterprises," *Computers in Industry*, vol. 60, no. 2, pp. 86-103, 2009.

[54] IBM. (2011) IBM Global Expense Reporting Solution (GERS). Document.

[55] Forrester Research, Inc., "Sizing the Cloud - A BT Futures Report," 2011.

[56] IBM. (2012) IBM BlueWorks Live. Webpage. [Online]. https://www.blueworkslive.com

[57] IEEE, "IEEE P2302™/D0.2 Draft Standard for Intercloud Interoperability and Federation (SIIF)," Draft Standard IEEE P2302/D0.2, 2012.

[58] F. Losavio, L. Chirinos, N. Levy, and A. Ramdane-Cherif, "Quality Characteristics for Software Architecture," *Journal of Object Technology*, vol. 2, no. 2, pp. 133-150, March-April 2003. [Online]. http://www.jot.fm/issues/issue_2003_03/article2/

[59] H.Y. Kim, K. Jerath, and F. Sheldon, "Assessment of High Integrity Software Components for Completeness, Consistency, Fault-Tolerance, and Reliability," in *Component-based Software Quality: Methods and Techniques*.: Springer, 2002.

Appendix A: Detailed project planning

Step	Task	Origin	Destination	Deliverable	Visibility
1	**Problem definition and literature review**				
	a. Problem definition	Environment	Research	Thoughts/ Unstructured list	Internal
	b. Initial review of academic sources (BPMS, Cloud Computing, Reference architectures, Cloud reference architectures)	Knowledge base	Research	Document	Internal
	c. Problem redefinition	Research	Research	Thoughts	Internal
2	**Initial original research**				
	a. Develop a new construct (BPaaS) and design the quadrant model	Develop/Build	Justify/Evaluate	Models	Internal
	b. Evaluation of the constructs/model with industry and academic professionals (3 from Capgemini, 2 from other companies, 2 from academia)	Evaluate	Develop/Build	Interview	Private
	c. Justification using case studies (analytical architecture analysis of 2 cases)	Justify	Develop/Build	Document	Internal
	d. Publication of a working paper	Research	Knowledge base	Document	Public
3	**Design of initial platform architecture**				
	a. Initial architecture design (*Iteration 1*)	Develop/Build	Evaluate	Model	Internal
	b. Evaluation and redefinition of constructs (static analysis)	Evaluate	Develop/Build	Interview	Internal
	c. Survey for additional information (multi-tenancy, scalability)	Knowledge base	Research	Thoughts	Private
	d. Survey for additional information (architectures of real Cloud implementations)	Environment	Research	Thoughts	Private
4	**Design of detailed platform architecture**				
	a. Architecture design - Architecture dimension with 2 models (*Iteration 2*)	Develop/Build	Evaluate	Models	Internal
	b. Evaluation – descriptive method	Evaluate	Develop/Build	Interview	Internal
5	**Design of organizational models**				
	a. Survey the knowledge base	Knowledge base	Research	Thoughts	Private
	b. Architecture design - Organization dimension (*Iteration 3*)	Develop/Build	Evaluate	Model	Internal
	c. Evaluation – descriptive method	Evaluate	Develop/Build	Interview	Private
6	**Design of business models**				
	a. Survey the knowledge base	Knowledge base	Research	Thoughts	Private
	b. Architecture design - Business dimension (*Iteration 4*)	Develop/Build	Evaluate	Model	Internal
	c. Evaluation - descriptive method	Evaluate	Develop/Build	Document	Internal
7	**Final platform redesign**				
	a. Final redesign	Evaluate	Research	Models	Internal
	b. Final paper (discussion, conclusions, future work)	Research	Research	Document	Public
	c. Addition to knowledge base	Research	Knowledge base	Document	Public
	d. Proposition of complete model to industry	Research	Environment	Document	Public

Appendix B: Evaluation interview questionnaire

Evaluation Interview ProCPlat	
Interviewee:	Date:
Position:	Time:
Q1: What would you be willing to do with business processes in the Cloud? (completeness construct)	
Q2: What modules would you expect to see in a business processes Cloud platform architecture model? (completeness model)	
Q3: To what external systems would you expect that the ProCPlat connects and how (using what mechanisms)? (interoperability)	
Q4: Does the proposed platform fit with the possible needs of an enterprise? (fit organization)	
Q5: Does the proposed platform fit with the current Cloud/IT landscape? (fit tech landscape)	

Appendix C: Interview results

In this appendix we present the raw results of the interviews conducted with experts from Capgemini. In order to keep the confidentiality of the interviewees, their names are not mentioned but instead encoded with letters. We provide the function of the expert as well as the date and time when the interview was conducted.

Interviewee A: Enterprise architect (interview conducted on 28.09.2012, 11:15)

Q1: Completeness construct

- Manage processes at different levels in the organization
- Manage end-to-end business processes spanning different business units in the enterprise
- Everything connected to business processes should be bundled in one application

Q2: Completeness model

- The BPM functionality and supporting GUI's should be migrated directly
- Process, business, and application logic should be separated
- Monitoring the execution of the business processes should be possible
- Application to application communication should be seamless, but also human interaction with the platform should be possible
- Decisions for the flow of the business process execution should made according to what the actual data is
- Presence of a module similar to an Enterprise Service Bus is required
- A data repository should be present
- Business rule engines should be in place to steer the end-to-end business process

Q3: Interoperability

- ProCPlat should be able to connect to on-premise BPM and workflow execution engines

Q4: Organizational fit

- It depends on how structured the data of the company is
- Should be used only if there is need to be used
- It is not a silver bullet to solve all problems of the company

Q5: Technological landscape fit

- ProCPlat is rather futuristic
- The size and maturity of the organization is from great importance
- Data with good quality is needed for a fit with the current technology

Interviewee B: Infrastructure architect (interview conducted on 16.10.2012, 10:30)

Q1: Completeness construct

- The system should be flexible enough, so that business processes can be extended
- Utilize the system in a Hybrid mode (reference to the Hybrid Cloud model)
- The system should be executing the business processes in a cheaper way compared to an on-premise system. Also the business processes should be secured as the ones on-premise.

Q2: Completeness model

- The user should be able to create workflows around messaging, purchasing, etc.
- Presence of support for business rules is required

Q3: Interoperability

- The platform should be able to connect to an existing mailing system and the databases of the local ERP systems

Q4: Organizational fit

- There could be a fit if the organization is willing to accept the technology

Q5: Technological landscape fit

- It is a good addition to the current landscape but in order for the vendor to be successful, it should be the first in the market (-60% revenue of second)

Interviewee C: Enterprise architect (interview conducted on 16.10.2012, 15:15)

Q1: Completeness construct

- Everything done on-premise should be possible to be migrated in the Cloud
- Such a platform should be used for standardized processes
- Such a platform would not be that helpful only for the processes of a single enterprise
- The platform should help the collaboration between companies and impose standards for business processes regarding their semantics
- The processes should be editable
- Such a platform should provide way to develop, test, and deploy business processes

Q2: Completeness model

- The scalability should be seamless
- More parties involved in the process would make the architecture more complicated
- Multi-tenancy is not a real issue
- There should be explicitly defined roles depending on who can do what
- The versioning of the processes and messages should be taken into account

Q3: Interoperability

- All the in-house applications and ERP/CRM/BPM systems
- Different Enterprise Services Busses
- Applications in the Cloud

Q4: Organizational fit

- There is a clear fit with the organizational needs. Actually there are a lot of implementations which possess part of the functionality proposed
- Such a platform does not have to be purely in the Cloud

Q5: Technological landscape fit

- The platform fits the current technological landscape and at some point such a platform should be offered to companies
- The current landscape is still a bit fragmented for such kind of initiatives

Interviewee D: Solution architect (interview conducted on 17.10.2012, 13:15)

Q1: Completeness construct

- Business processes running in the Cloud should be automated
- There should be an option for human input and interaction too by adding intelligence or configuration
- Processes should be monitored
- Business processes should be defined in the Cloud and linked to systems on-premise for automation

Q2: Completeness model

- Module for process definition
- Module for process monitoring
- Integration layer
- Data repository

Q3: Interoperability

- Master Data Systems
- Financial and HR systems (SAP)
- All in-house applications
- Client management systems
- External Cloud applications

Q4: Organizational fit

- The platform will serve the organization but there is a “cold water fear” for the companies, since such a product has not been commercially tested
- The idea for such a platform in itself is powerful
- Bringing agility to companies might be painful for them in the beginning

Q5: Technological landscape fit

- There is nothing scary in the platform, so it should fit well
- Existing components should be reused
- Sticky session are possible for Enterprise Service Bus