Sami Koivisto

Creating Reusable Services for Customized Software Demos

Helsinki Metropolia University of Applied Sciences

Master’s Degree

Business Informatics

Thesis

24.11.2016

Author Title Number of Pages Date	Sami Koivisto Creating Reusable Services for Customized Software Demos 63 24 November 2016
Degree	Master of Engineering
Degree Programme	Business Informatics
Instructor	Thomas Rohweder, Principal Lecturer

The objective of this study was to improve the methods how Technical Sales teams in IBM Finland Hybrid Cloud Integration are building customized demonstrations and how this effort can be reduced by reusable cloud-based services. The goal of the study was to produce a set of reusable services and a related operating model to help fasten the time-to-value and reduce maintenance effort to build meaningful customized demos: showing what the software does best, and winning more business. The thesis was scoped to cover Business Process Management, Operational Decision Management, API Management and Integration services capability areas.

This thesis used Metropolia variation of Stage-Gate system as research construction. The thesis process was sub-divided into five stages, or work stations. Between each stage there is a quality control checkpoint, or gate.

The first phase was the scoping of the thesis. In this phase business problem, objective and outcome was identified and the thesis project was designed. The second phase was Current State Analysis. It played a major part in the whole thesis project, since it gave guidelines to what should actually be improved. Literature search was the third phase, and a lot of theory was mapped to the issues and available best practices and frameworks from literature were found. The conceptual Framework was built using that information. It was then used to build the actual solution in the fourth phase. That was the most iterative part of the study, and contained hands-on solution building. The last phase in the project plan was a sample implementation. In that phase the new demo was built using the implemented assets from the previous phase.

The output of this thesis was a set of software services that are reusable and compatible through the common data model and standard interfaces. Along with the services is the related operating model to describe how the services should be utilized to build demos. This should help fasten the time-to-value and reduce maintenance effort to build meaningful customized demos.

Keywords	Business Process Management, Operational Decision Management, API Management, integration, customized demonstrations, Technical Sales

Contents

1 Introduction

1.1 Case company - IBM

Case company for this thesis is IBM. IBM is a cognitive solutions and cloud platform company that digitally transforms companies and industries. With operations in over 170 countries and innovation at the core of its strategy, IBM delivers by industry cognitive, cloud, big data and analytics, mobile, social business and security solutions.

1.2 Business challenge and objective

Today's high-value clients seek personalized and compelling software demonstrations. Delivering an engaging experience can determine which solution wins (or loses) the business. Sales engineers typically need to build such demonstrations from scratch or use existing demo as a base using Virtual Machines. This is time consuming and generates extra workload in means of maintenance of multiple Virtual Machines.

At the same time Cloud computing has fundamentally changed the nature of how organizations purchase and use technology. International Data Corporation (IDC) is the premier global provider of market intelligence and advisory services, IDC predicts in Special study "Cloud Computing: The Essential Foundation of Industry Digital Transformation — Worldwide and U.S. Cloud Forecast by Vertical, 2015–2019" (Smith, 2015) that those IT suppliers that showcase the long-term benefits and the true value of the cloud as a platform in a given vertical (e.g., efficiency gains in business processes and improvement in customer acquisition/customer experience) will be most successful.

Given this, the objective of this study is to establish a set of reusable services and related operating model that is developed to help fasten the time-to-value and reduce maintenance effort to build meaningful customized demos showing what the software does best

1.3 Scope

The portfolio of IBM hybrid cloud integration consists of typical middleware solutions, application performance monitoring solutions, and application platform solutions. At the planning phase the project was scoped only to cover process centric areas that are separate platforms but typically integrated together using services. Platforms covered are:

- Business Process Management
- Operational Decision Management
- API Management
- Integration services

These platforms offer capabilities that are commonly used in process centric end-to-end solutions. Therefore together they join a functional set of software platforms that can be integrated together so that different use cases and capabilities can be combined in a customized demo scenarios.

Goals for this study is to build a limited set of compatible services that help:

- Limit time spent preparing demonstrations
- Maximize effectiveness of demonstration delivery
- Increase and broaden use of demonstration assets
- Deliver effective, relevant and current demonstrations
- Lower the barrier of entry for delivering demonstrations

2 Research approach

2.1 Research design

This thesis uses Metropolia variation of Stage-Gate system as research construction. Cooper (1990) describes stage-gate models in “Stage-Gate System: A New Tool for Managing New Products” as effective tools to manage, direct, and control product-innovation efforts. The way to improve the production process to manufacture a physical product is to focus on the process itself - to remove variances in the process. A process is sub-divided into a number of stages or work stations. Between each work station, or stage, there is a quality control checkpoint or gate. Stage-gate systems use similar methods to manage the innovation process. They divide the innovation process into a predetermined set of stages, themselves composed of a group of prescribed, related, and often parallel activities. For example, the "Validation" stage might entail a list of mandatory or optional activities such as in-house prototype tests, field tests with customers, pilot or trial production, and test marketing.

Thesis stage-gate model is divided into five stages. The following diagram illustrates those stages and the outputs of that stage.

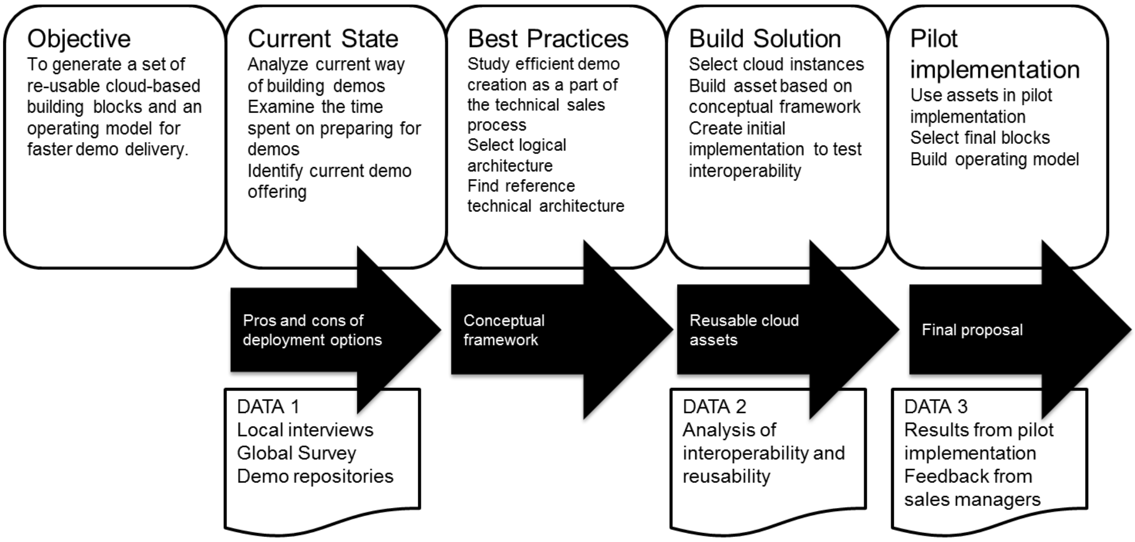

Objective
To generate a set of re-usable cloud-based building blocks and an operating model for faster demo delivery.
Current State
Analyze current way of building demos
Examine the time spent on preparing for demos
Identify current demo offering
Best Practices
Study efficient demo creation as a part of the technical sales process
Select logical architecture
Find reference technical architecture
Build Solution
Select cloud instances
Build asset based on conceptual framework
Create initial implementation to test interoperability
Pilot implementation
Use assets in pilot implementation
Select final blocks
Build operating model
Pros and cons of deployment options
Conceptual framework
Reusable cloud assets
Final proposal
DATA 1
Local interviews
Global Survey
Demo repositories
DATA 2
Analysis of interoperability and reusability
DATA 3
Results from pilot implementation
Feedback from sales managers

Figure 1 Project Plan

First phase in project plan is scoping the thesis. In this phase business problem, objective and outcome was identified. Business problem was related to my personal role in the organization, so it was natural selection as the subject of the thesis. Scoping and objectives were related to the software portfolio of the organizational unit I was working for.

Second phase is Current State Analysis. It played a major part in the whole thesis project, since it gave guidelines to what should actually be improved. The results from this phase were not surprising, but they gave assurance and focus on the scoping.

Literature search was the third phase, and a lot of theory was mapped to the issues and available best practices and frameworks from literature were found. Conceptual Framework was built using that information. It was then used to build actual solution in phase four. That was the most iterative part of the study, and contains hands on solution building and actualized the conceptual framework. Last phase in project plan was a pilot implementation. In that phase the new demo was built using the implemented assets from previous phase. This is the phase where the theory and feasibility of the project was tested. This phase crystallizes the end result and concretizes the value of the exercise.

The project design and methodology follows methods from David Coghlan's book ‘Doing Action Research in Your Own Organisation’. According to Coghlan (2014), several broad characteristics define action research:

- research in action, rather than research about action;
- a collaborative democratic partnership;
- a sequence of events and an approach to problem-solving

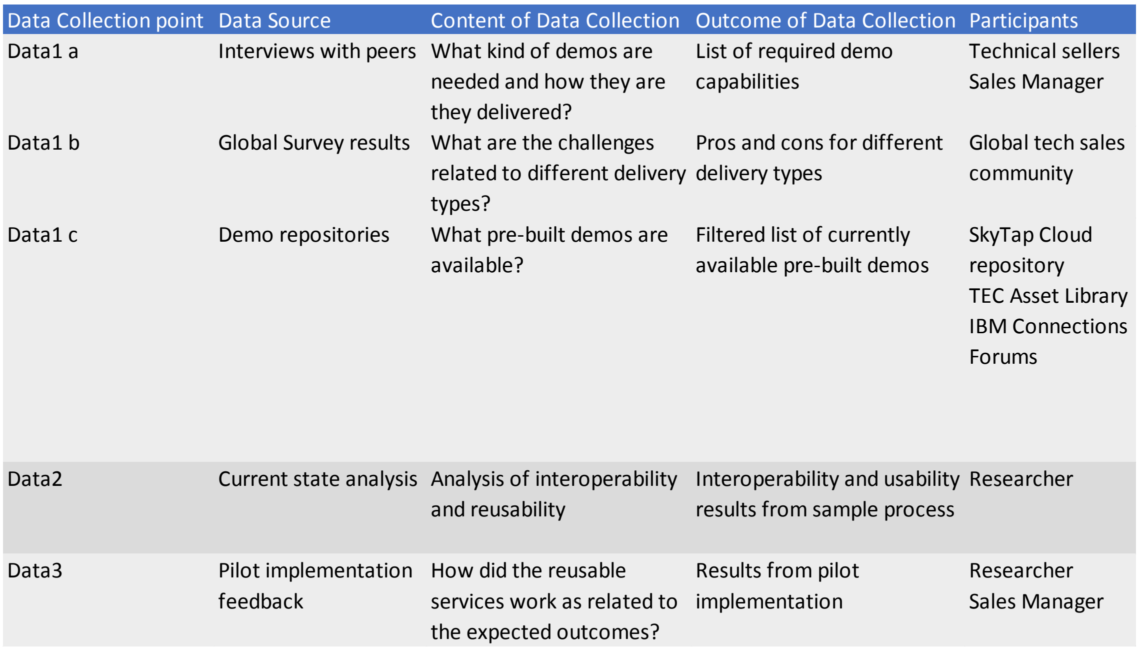
This means in practise that this thesis was focusing on solving the problem whilst simultaneously conducting research. This makes it iterative and inclusive. It also puts action researcher to meet certain challenges:

> “Insider action researchers need to confront the issues pertaining to preunderstanding, role duality and organizational politics. Attention to experience, understanding and judgement which leads to action, provides a methodology through which they can affirm what and how they know. They need to do so in a critical realist approach which challenges them to transcend their own subjectivity through the quality of how they are attentive to the data, intelligent in their understanding, reasonable in their judgements and responsible in their actions.” (Coghlan, 2007)

Since the author of this thesis is directly involved in the action research and implementation of the outcome, the emphasis is put on the approach to overcome the subjectivity and prejudice for the optimal outcome. This topic is discussed more in the chapter “Reliability and validity considerations”.

2.2 Data collection and analysis

Following diagram shows different data collection points for the research and how this data is collected during the research project.



Data Collection point	Data Source	Content of Data Collection	Outcome of Data Collection	Participants
Data1 a	Interviews with peers	What kind of demos are needed and how they are they delivered?	List of required demo capabilities	Technical sellers Sales Manager
Data1 b	Global Survey results	What are the challenges related to different delivery types?	Pros and cons for different delivery types	Global tech sales community
Data1 c	Demo repositories	What pre-built demos are available?	Filtered list of currently available pre-built demos	SkyTap Cloud repository TEC Asset Library IBM Connections Forums
Data2	Current state analysis	Analysis of interoperability and reusability	Interoperability and usability results from sample process	Researcher
Data3	Pilot implementation feedback	How did the reusable services work as related to the expected outcomes?	Results from pilot implementation	Researcher Sales Manager

Figure 2 Data Collection plan

First bit of information that is valuable for this research is what kind of demo's customers typically require in order to progress and eventually close the deal. We try to answer questions what are the typical use cases and what are the golden nuggets that IBM Middleware can demonstrate to beat competitors. This data is collected mostly in informal conversational interviews with peers. According to Michael Quinn Patton (1990) the informal conversational interview relies entirely on the spontaneous generation of questions in the natural flow of an interaction. This is a natural selection for method on how work is conducted currently. Results are recorded in general level for this thesis. Another interesting data set is results from global survey for IBM technical sellers that answers questions on how demos are conducted, what resources are used, what are the reasons why some of the resources are used and how much in average technical sellers are using in preparation for demos. Survey results are analysed by mapping aggregate results to scope of this thesis.

Second bit of data is to search through the repositories for a list available out-of-the-box demos, how they can be deployed and how useful they are for building customized demos. Results from the queries are listed as a part of current state analysis of this study. Only relevant demo offering is included, so all out-of-date versions and non-related product demos are filtered out from the results. These two bits of data constitute first set of data.

From those bits of information it is then easy to collect sample list of assets and services that will be implemented according to the conceptual framework. Together with the operating model they form second set of data.

When implementation phase is over, assets will be tested in real customer case, and feedback from that final phase will be the third set of data that will be analysed. It is based on empirical study on how well new ways to deliver demos help on limit time spent preparing demonstrations, maximize effectiveness of demonstration delivery, increase and broaden use of demonstration assets, deliver effective, relevant and current demonstrations and lower the barrier of entry for delivering demonstrations.

3 Current state analysis

3.1 Scope of the Current State Analysis

The data analysis was done by searching available assets on each delivery model, analysing what are the benefits of delivering demo in that model and what are the problems in that delivery model. There are multiple customer ready demo assets available for IBM technical sales. These include downloadable Virtual Machines that contain pre-built demos, online demos and Virtual Machines running in cloud environment. Results from a global survey of Systems Middleware technical sellers was used to analyse how technical sellers are using these assets, and what are the limitations and problems related to each delivery types.

3.2 Current demo offering

Current demo offering has three distinct delivery models:

- Downloadable VMWare virtual machines
- VMWare virtual machines deployed in cloud environment
- Local software installation

In some cases some demos are delivered using recorded video demos, but in this case they are not relevant since they cannot be customized.

All supported product versions are available to IBM technical sellers for a download. These installations are so called vanilla installations. In information technology, vanilla is an adjective meaning plain or basic. The unfeatured version of a product is sometimes referred to as the vanilla version. The term is based on the fact that vanilla is the most popular or at least the most commonly served flavour of ice cream (techtarget.com, 2005).

Since local installations do not contain pre-built demos, only local VM’s or cloud VM’s are interesting in this scope. This means that there are only three repositories that needs to be analysed. These are Skytap for VM cloud deployments, TEC Asset Library for downloadable Virtual Machines and discussion forums on IBM Connections Communities.

Skytap is a cloud computing platform that enables users to create, access, and share multi-machine virtual environments. Its possible do all of this through a web browser. This means that users can access and control your virtual machines anywhere that has an Internet connection. These environments can be used for pretty much anything that a traditional on-site computing network can be used for, including development, testing, demos, and even production. The focus for IBM sales is on flexibility and ease of use, and ability to manage virtual environments with minimal outside assistance.

IBM’s worldwide Technical Exploration Centres (TECs) offer consistent, high-quality facilities where IBM customers can see, touch and experience IBM software technologies and collaborate with IBM subject matter experts. The worldwide TEC team is a service organization that supports the delivery of technical accelerator events and skills development, including;

- Proofs of Technology (PoTs) and Proofs of Concept (PoCs)
- Technical demonstrations and workshops
- Opportunity pursuit by local Technical Sales teams
- SkillWorks events

TEC Asset Library holds downloadable VM’s containing PoTs and demos.

IBM Connections is a social network platform. At IBM, it contains different discussion forums for different communities. Communities are places where IBMers can work together with people who have common interests, roles and expertise. They also include ideation, media gallery, and bridging capabilities. Some of the available demos are only available through these forums.

3.3 Current way of building customized demos

Based on discussions and interviews with different technical teams that deliver customized demos to prospects and customers, three distinct methods to build demos was identified:

- Use pre-built proof-of-technology materials
- Build from scratch with locally installed software or VM image
- Use cloud based SaaS services to build demo

IBM conducted a global survey (IBM, 2015) to technical sellers around the world and asked about demonstration environments, how were people doing their demos, do they use a cloud environment, and if not, why.

Based on the data from the Systems group (that solutions from the scope of this study belong to) 50% of technical sellers are using VMware workstation to deliver their demonstrations. This means that they are relying on the limited hardware on their laptops and that causes performance issues since large Virtual Machines don't perform well on a laptops. It's also limiting the demonstrations that they can deliver because they can't build complex, multiproduct demonstrations. The outcome is that technical sales are not really getting the right visual across to their customers. (IBM, 2015)

Pre-built demos are also available as cloud deployed VM images. They work well for some of the use cases. But since they are pre-built, there is often a need to amend demos for customer cases. It's because the content isn't suitable or can be out of date.

That is typically because the demonstrations made available aren't keeping up with new product releases. This has effect on how much time is really spent for carrying demonstrations. Sixty percent of respondents take between one and five days to prepare for a demo. This means that the average falls somewhere in the middle in the two- to three-day period. About 25% of respondents said they can take them up in two weeks (IBM, 2015). From the business perspective that's a long time preparing for the demonstration and a lot of that comes from preparing the environment.

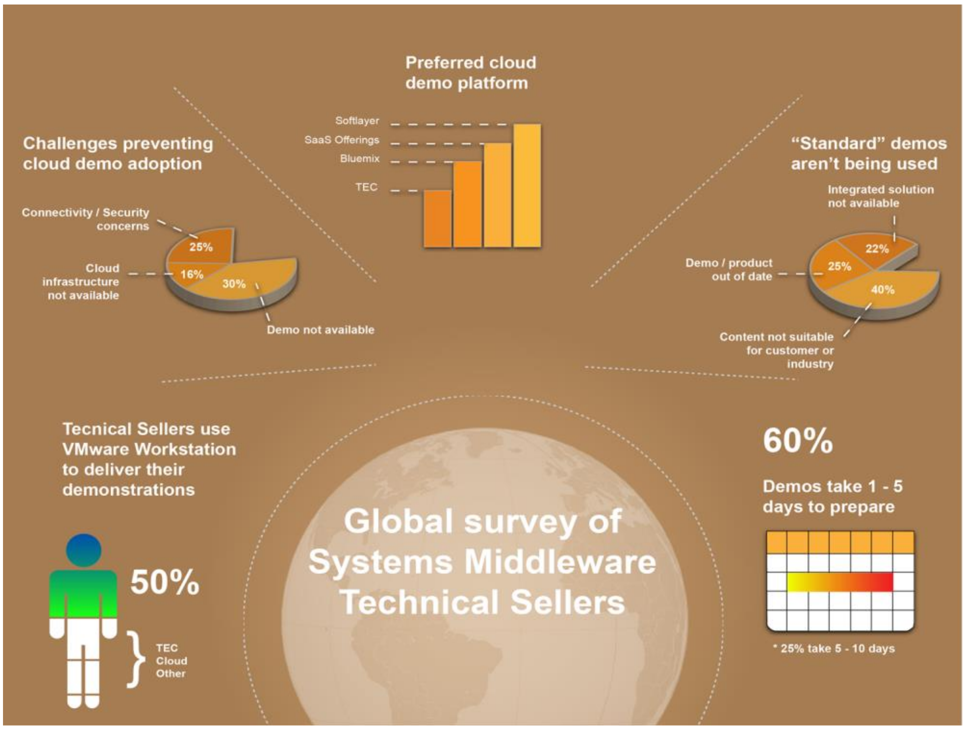


Figure 3 Summary of global survey (IBM, 2015)

Results show that there is a need to cut down on time spent preparing the demonstrations. They also show that it could be done by creating building blocks that are already ready to go and using customization to make any tweaks that are necessary easily so that you’re up and running and delivering demonstration.

3.4 Currently available pre-built demos

This study is only interested in pre-built content for demos. And since demos being examined under this study are related to sales process, we are only interested in latest product versions. This limits the demos that are searched only to latest versions of the products listed in the scope of this thesis and only the demos delivered either as downloadable Virtual Machines or Virtual Machines that are deployable to cloud data centre in EMEA (Europe, the Middle East and Africa) region.

This inventory of assets was carried out on 1.3.2016. Overall there is a broad selection of VM’s available. Ones the ones that are relevant according to the conditions mentioned above were selected. It should also be noted that the change of the location of the TEC Asset library was undergoing, and not all images were available.

Following list describes available content:

SkyTap:

- Smarter Process - Mobile Enabled Coaches (Based on BPM 8.5.6 v2.0) [EMEA]
- IBM Business Process Manager 8.5.6 Proof of Technology [EMEA]
- Smarter Process - Order Fulfilment Advanced Demo (based on BPM 8.5.6 v2.0) [EMEA]
- Smarter Process - Basic Case Manager Demo (based on BPM 8.5.6 v2.0) [EMEA]
- IBM BPM + Digital Experience Demo Image (v1.0) [EMEA]
- IBM Integration Bus v10 PoT VM [EMEA]
- API Management v4 PoT [EMEA]
- IBM ODM Advanced 8.8 PoT [EMEA]
- ODM Advanced DSI 8.7 PoT [EMEA]
- ODM Standard 8.7 PoT [EMEA]

TEC Asset Library:

- Discovering the value of IBM Integration Bus v10
- Discovering the value of IBM Integration for Healthcare
- Discovering the value of IBM Business Process Manager 8.5.5
- Discovering the value of IBM Operational Decision Manager v8.5

IBM Connections Forums:

- API Management POT v4
- ODM Advanced 8.8 PoT
- ODM Standard Rules 8.7 PoT
- ODM Advanced DSI 8.7 PoT

3.5 Data collection topics

The interviews were informal conversational interview with four topics. They were peer-to-peer discussions around current way of working and what are the challenges related to that. Data collection plan is described in the following table:

	Content	Data Source	Outcomes
DATA 1 Current State Analysis	1) What kind of demos are needed? 2) How are they delivered? 3) What pre-built demos are available? 4) What are the challenges related to different delivery types?	1) Interviews with peers -Technical sellers -Sales Manager 2) Global Survey results -IBM Global Survey for Technical Sellers 3) Demo repositories - SkyTap Cloud repository -TEC Asset Library -IBM Connections Forums	1) Filtered list of required demo capabilities. 2) Pros and cons for different delivery types.

Based on the data collected from the interviews, a comparison of different deployment options and their challenges was gathered into the following table. It shows different delivery types and pros and cons for each of them.

Delivery type	Pros	Cons
Local installation	• Full control over installation and versioning • No network latency • Updates can deployed when needed	• Manual updates • Hard clean-up process • No pre-built assets • Performance limited to laptop hardware
Local VM	• Can hold pre-built demos • Snapshots can be used to easy clean-up process • Can easily be copied into new images • Network connection is not mandatory • No network latency	• Long download time • Performance limited to host laptop hardware • Performance overhead of host and guest OS • Difficult to share across the network • Complex multiproduct scenarios can overwhelm performance
Cloud VM	• No download time • Easy to deploy • Does not take laptop resources • Different data centres available for different geolocations	• Network dependency • Network latency • No snapshots available
Video demo	• No setup time • No 'demo effect' risk • Lot of material available from YouTube	• No customization • Fixed time constraint • No multiproduct demos

3.6 Summary of current state analysis

Based on the current state analysis, there are lot of pre-built demos available. There are many technical issues that prevent efficient usage of those assets. Those issues are different for different delivery types, but the common theme is that there is some overhead that extends the time that goes into preparing technical environment before the actual building of the demo can start. In some cases demos are not ready, or pre-built demo is not suitable for the use case. Then technical sellers need to utilize vanilla installations of the products. When that is the case, there are similar issues that needs to be solved before implementation phase.

In all of the cases it is difficult to build multiproduct demos, either because of the integration problems in separate virtual machines, or limited hardware capacity of the single laptop. It seems like by removing some of the technical problems it could be possible to cut down on time spent preparing the demonstrations.

4 Good practices of reusability in software sales

4.1 Software sales process

Since the scope of this thesis is related to the software sales, it’s vital that we understand the role of building demos in software sales. The book ‘Mastering

Technical Sales: The Sales Engineer's Handbook' defines sales life cycle as follows:

Table 1 Sales life cycle (John Care, 2008)

Step	New Product Introduction
1	Market definition
2	Marketing campaign
	Sales Process
3	Lead qualification
4	Request for proposal (RFP) (optional)
5	Needs analysis, discovery, and customer engagement
6	Presentation, demonstration, and proposal
7	Evaluation (optional)
8	Negotiation and closing
	Post-sales Support
9	Account management and add-on sales

This definition is highly generalized, and most of the companies – including IBM – have defined their own unique sales processes. IBM has quite complex and well documented definition of sales process. Part that is most interesting from technical sales perspective is Opportunity to Order sub-process. It holds the part of the sales process where customized demos are typically delivered. Process definition describes roles and responsibilities of delivering demo, under what circumstances it's most suitable and what should be the outcomes of the demo session. But there is very limited help on how to actually build and conduct the demo session.

4.2 Efficient demo creation as a part of the technical sales process

The reason why customized demos are used in sales cycles is typically the need to present solutions rather that features of certain products. Steli Efti highlights this in 'The Ultimate Startup Guide to Outbound Sales' (2015) by writing that your customers aren't interested in your product. They are only interested in what your product will do for them. Always translate your product's features into your customers' benefits. Don't focus on the bells and whistles, but what they do for them.

Study shows that the level of perceived payoffs from IT is directly related to corporate goals for IT (Tallon, 2000). Many companies have a significant opportunity to differentiate themselves from competitors by learning how to create, quantify, communicate and capture customer value by implementing customer value based pricing strategies. (Hinterhuber, 2012).
The disability to communicate value of the solution to the customer often lead to discounting, especially towards the ends of quarters:

> "Salespeople agree to significantly lower pricing in quarters in which they have a financial incentive to close a deal, resulting in mispricing that costs the vendor 6%–8% of revenue." (Larkin, 2014)

Value justification creates compelling reasons to take action. With a compelling value justification, buyers often ask to get started early. In other words, the cost or impact of delay is so overwhelming that they can't afford to wait any longer. (Eades, 2003)

One CEO commented in a study on 'value based pricing' (Hinterhuber, 2012):
"You only need to be brave for one second, and it's when the guy asks for a discount and you say no. And then you justify it. That takes bravery. So how do you get salespeople in a mind set to justify the price? You don't have to go in there and be Superman for two hours. You have to be Superman for one second."

The importance of communicating the value is highly understood and valued among the sales people, but the comprehension how it is accomplished and when it's successful seems to be vaguer:

> "When I ask salespeople and sales executives whether their company provides solutions, they answer yes—virtually every time. Yet when I ask these same salespeople what solution they provided for their last customer, their answers tell a different story. I hear all about their products and services, complete with dazzling brand names and mind-boggling acronyms." (Eades, 2003)

Cakanyildirim and Tevfik (2002) talk about customizing demonstration according to customer needs to reduce standard deviation of customer needs and to increase profits in their study "Using Demonstration to Promote Information Products". They note that information products are becoming complex and this trend is especially pronounced in the software industry. This makes it harder for vendors to communicate the features of the products to potential customers. Two conditions must be satisfied roughly before a purchase: first product awareness and then sufficiently high perceived (by customers) product value.
Latter condition is one of the reasons why customer value proposition has become one of the most widely used terms in business markets in recent years. Yet the management-practice research by Anderson, et al. (2006) reveals that there is no agreement as to what constitutes a customer value proposition - or what makes one persuasive. Their study shows that most value propositions make claims of savings and benefits to the customer without backing them up. According to their study, an offering may actually provide superior value-but if the supplier doesn't demonstrate and document that claim, a customer manager will likely dismiss it as marketing puffery. Customer managers, increasingly held accountable for reducing costs, don't have the luxury of simply believing suppliers' assertions. Simply put, to make customer value propositions persuasive, suppliers must be able to demonstrate and document them. Study shows that best-practice suppliers base their value proposition on the few elements that matter most to target customers, demonstrate the value of this superior performance, and communicate it in a way that conveys a sophisticated understanding of the customer's business priorities.

Next, let’s discuss what different means there are to repeatedly do this in customized and optimized way.

4.3 Modular production logic

One of the theories related to optimizing productivity is modular production logic. Rogers & Bottaci (1997) are discussing production modularity in study "Modular production systems: a new manufacturing paradigm" in low- and high-technology products manufacturers. The building of artefacts from standardized modules, subsystems or components has been common for centuries, but it is becoming strategically important for manufacturers to shorten the `design to market' lead time. They argue that modular production systems enable the use of the simplest integrated combination of processes, machine systems, tooling, people, organizational structures, information flows, and control and computer systems necessary to perform a given task. This idea can easily be adopted to the creation of software demos through modularity. Based on this concept it should be possible to use most suitable platform to build service and then integrate that into the single solution.

4.4 Mass customization logic

As the modularity can be used to structure integrated solutions out of modular blocks, another theory that can be utilized to create customized solutions is a mass customization logic. Idea is to use modularity with ability combine parts into multiple different combinations to produce customization with standardized modules.

While companies everywhere seem to be making great strides in quality by focusing on continuous improvement, many firms are already moving beyond this to mass customization. In this business model, stable but very flexible and responsive processes provide a dynamic flow of goods and services, enabling companies to achieve both low costs and high variety, even individual customization. The organization's primary thrust is to identify and fulfil the individual wants and needs of every customer. Ideally, the product life cycle is one unit: every product is different than the last--and uniquely suited to a particular customer's needs. (Pine, 1993)

This logic translates into building customized demos. Each demo should be unique while still built using as standardized components as possible.

To move to mass customization requires that products be modularized to provide unique combinations for any customer. Further, the tightly coupled processes created through continuous improvement should also be broken apart and modularized, so that, at its ideal, any process can link to any other process to create the unique end-to-end value chain that will best satisfy each individual customer. This results in a dynamic network where linkages between people and processes are loosely coupled to enable the mass customization of goods and services. (Pine, 1993)

4.5 Reusability in software engineering

Modularity and mass customization are themes that directly link into software development. Thomas Erl states in ‘SOA Principles of Service Design’ (2007) that any software program ever built for sale to the general public was designed with reuse in mind. Whether it’s an operating system, a shrink-wrapped accounting product, or an entire middleware platform, in the initial design stages of these programs, the considerations we listed in the previous section were very likely taken into account.

Product cost efficiency concept has been an important characteristic of mass production firms (Nemetz and Fry 1988). A key advantage of new manufacturing approaches such as mass customization is that firms can continue to obtain the cost efficiencies that characterized mass producers (Boynton et al. 1993). In ‘The effects of customizability and reusability on perceived process and competitive performance of software firms’ study Nidumolu and Knotts (1998) define product cost efficiency of a software firm is as its ability, relative to competitors, to produce software products efficiently from a cost perspective.

There is considerable controversy and little agreement in the literature over how to measure reusability. Some of the issues include what kinds of outputs should be considered as valid components for reuse, what kinds of measurement scales could be employed, whether reuse for versions of the same product should be allowed, and whether levels or quality of reuse should be used. In this study, the approach to measuring reusability for the first two issues was kept deliberately broad, i.e., valid reusable components included not just code, but also specifications, data, and documentation. Moreover, both interval and ratio scales for measuring reuse were used to permit cross-validation of results. However, much reuse occurs between versions of the same product, which could misleadingly signify economies of scope for producing a wide range of products (Nidumolu & Knotts, 1998).

In theory, reuse is a pretty straight-forward idea: simply make a software program useful for more than just one single purpose. The reasons for doing so are also quite evident. Whereas something that is useful for a single purpose will provide value, something that is repeatedly useful will provide repeated value and is therefore a more attractive investment. (Erl, 2007)

When comparing the point-of-views about evolution of application and reusability from two major analyst companies, Gartner (Natis & Altman, 2014) and Forrester (Bittner, et al., 2015), it’s evident that whenever a new paradigm is introduced in application development space it’s highly focused on agility through reusability. This is achieved through layering where reusability is present in multiple layers.

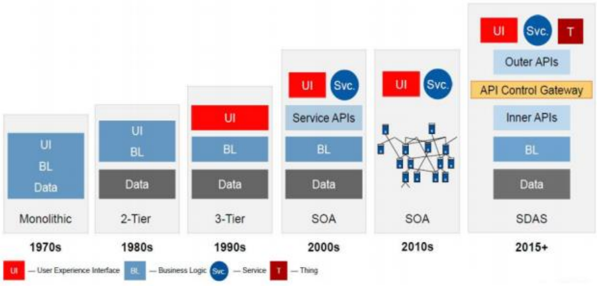


Figure 4 Technology Evolution by Gartner

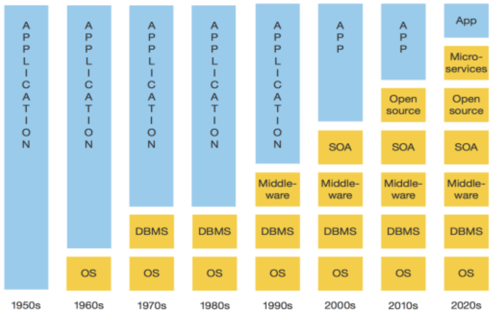


Figure 5 Evolution of Application Architecture for Digital Business by Forrester

One common and pervasive paradigm in information technology is Service Oriented Architecture, or SOA. Thomas Erl makes similar note about history of IT and SOA in his book ‘SOA Principles of Service Design’ (2007). He states that Service-orientation is not a brand new paradigm that aims to replace all that preceded it. It, in fact, incorporates and builds upon proven and successful elements from past paradigms and combines these with design approaches shaped to leverage recent technology innovations. This is why he do not refer to SOA as a revolutionary model in the history of IT. It is simply the next stage in an evolutionary cycle that began with the application of modularity on a small scale (by organizing simple programming routines into shared modules for example) and has now spread to the potential modularization of the enterprise.

Nicolai M. Josuttis (2007) defines SOA as not a concrete architecture: it is something that leads to a concrete architecture. Why SOA in is important aspect in this study is its strong convergence with Business Process Management, or BPM. The SOA approach provides huge benefits in BPM.

In older IT architectures, the business processes and applications were not linked together. Business processes were "nice drawings", and applications very complex and resistant to change. If the process needed to change, which was quite often, it took a very long time to adapt all applications. The SOA approach provides huge savings as applications are much better aligned with business processes. This reduces the time required for adoption and makes the IT system more flexible. (Pant & Juric, 2008)

4.6 Cloud computing

There are different definitions of cloud computing. Typically it means that the services are hosted remotely and accessed over a network. National Institute of Standards and Technology (NIST) has published special publication on definition of cloud computing. The NIST Definition of Cloud Computing:

> “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.” (Mell & Grance, 2011)

Cloud computing and SOA are different concepts, but they are related. SOA is a pattern of architecture, whereas cloud computing is an instance of architecture, or an architectural option. SOA is more holistic and strategic, meaning it deals with the complete enterprise including the business drivers, whereas cloud computing is more tactical and is a way of solving a problem. They are linked, and it is difficult to do one without the other if you are looking to solve problems at the enterprise level. (Linthicum, 2009)

"The technological innovations and enabling capabilities unleashed by cloud have fostered new opportunities across the industries," says Eileen Smith, program manager, IDC Global Technology and Research Group. "As a result, it is necessary for both technology vendors and buyers to recognize the industry drivers and barriers of cloud deployment, to understand the business transformation brought by cloud, and to act upon the changes that will shape business and technology strategy in the coming years." (Smith, 2015)

4.7 Maturity Models

A maturity model can be described as a structured collection of elements that describe certain aspects of capability maturity in an organization (Rosing, et al., 2014). “The Complete Business Process Handbook” also notes that maturity models can help organizations assess their current capabilities in a structured way to implement changes and improvements. In this study, the maturity models are used to identify capabilities that are essential to the scope of the conceptual framework, and thus for the outcome of the project. Maturity models are good for identifying what are the main dimensions related to certain functional area, and many times they are used when planning functionality of software products.

4.8 Conceptual Framework

4.8.1 Conceptual Framework building blocks

Conceptual Framework of this study builds on SOA Reference architecture and BPM Maturity model introduced in section “Good practices”. Maturity model describes the capabilities needed in the implementation, and reference architecture describes how these capabilities can be deployed into the technical platform.

We use the BPM maturity model to identify capabilities that are needed in end-to-end scenarios in the scope of this project. Following dimensions by Zheltonogov, et al. (2015) describes the BPM maturity model:

- Knowledge worker processes (less predictable, ad hoc, and case-centric)
- Automated business processes (more predictable, straight through processing (STP), and deterministic)
- Rules and decisions
- Analytics and key performance indicators (KPIs)
- Services and integrations
- API management
- Data or Information Architecture (as it relates to the BPM initiative)
- Infrastructure

As we start to match dimensions from maturity model to technological implementations behind services it’s easier to start bottom-up to define the conceptual framework. This is particularly practical since the top levels are related to the process level, and it is the customization layer in this context.

In the next chapter it is described how dimensions in BPM maturity model are taken into consideration when defining the conceptual framework.

4.8.2 Infrastructure

Building the framework from the bottom-up, the first layer in BPM maturity model is infrastructure. We defined as the premise for this project to utilize cloud as a technical deployment platform. Cloud will hide technical implementations from the users, and therefore the conceptual framework does not need to define the infrastructure that the solutions will run on. This also means that solution can focus on the functionality as opposed to worrying about how some technology works. Cloud computing has matured to a stage where basically anybody can get access to technologies and services in the cloud that were only available to the biggest firms a mere five years ago.

In a nutshell, cloud computing has the potential to revolutionise the way that businesses procure and use IT. Rather than managing IT systems internally, as has been the traditional approach, the cloud allows companies to reduce costs by outsourcing the management to an off-site cloud service provider and accessing infrastructure, storage and applications securely over the web. The benefits are obvious: with no physical ties to premises, the organisation is freed from geographical ties. (BCS The Chartered Institute for IT, 2012)

4.8.3 Information architecture

Next layer is the data or information architecture. Since the problem for information architecture in this context is to minimize dependencies when integrating applications that use different data formats, we need to define a Data Model that is independent from any specific application. Bobby Woolf and Gregor Hohpe note in their book “Enterprise Integration Patterns” (2004), that the easiest solution from a technical perspective may seem to dictate a uniform format on all participants. This allows to define a data format for this specific scope as a Canonical Data Model. It’s independent from any specific application and provides an additional level of indirection between application's individual data formats. In contrast, Stefan Tilkov talks about canonical data models in his blog and states: “In fact, it’s a horrible, horrible idea. Don’t do it” (Tilkov, 2015). But he also refers to book “Domain Driven Design” by Eric Evans where Evans gave a name to a concept called The Bounded Context. It is a structuring mechanism that avoids having a single huge model for all of your application, simply because that (a) becomes unmanageable and (b) makes no sense to begin with (Evans, 2003). This leads us to use data model that is limited to the bounded context of few integrated services.

There is plethora of industry specific data formats. One organization that is developing specifications for different domains is The Open Applications Group that is a not-for-profit open standards development organization. The Open Applications Group Integration Specification (OAGIS) is an effort to provide a canonical business language for information integration. It uses XML as the common alphabet for defining business messages, and for identifying business processes (scenarios) that allow businesses and business applications to communicate. (Open Applications Group Corporation, 2016)
They have defined a sales orders schema that can be utilized as a canonical data model in this bounded context use case. Typical problem using canonical data formats is their complexity and often deep structures that are result from data normalization. In this case the aim is to make integrations as easy as possible with possibility to ignore some requirements in real life implementations. One method for simplifying the data structure is to flatten, or to denormalize the data structure.

Data denormalization exploits the fact that an application’s queries and transactions usually target few data columns. This, combined with classical database denormalization techniques such as query rewriting and column replication, allows us to cluster the data into disjoint data services. Although this property was verified in all applications that we examined, one cannot exclude the possible existence of applications with sufficient data overlap to prevent any service-oriented denormalization. (Wei, et al., 2008)

Idea behind denormalized unified data model is to create such a model that can be technically integrated and utilized by all components that contribute to end-to-end solution. Such data model should be solution free.

4.8.4 API management

Although the integration-centric elements of the original SOA reference model remain important, those elements are not sufficient to address the needs that are arising from the API and service economy (IBM Redbooks, 2014). That is why in addition to integration dimension discussed later in chapter “Services and integrations”, it is important to understand API Management dimension and its position in solution stack.

Publishing services through API management is not mandatory for reusability or customization purposes, but it helps to demonstrate also how services can be reused and published using such capabilities. API Management allows to separate services from the back end by creating inner API and outer API.
APIs are needed for many reasons, including the following reasons:

- Expose services for more flexibility.
- Easier access to customers and partners.
- Need to reach more channels and devices.
- Make your place in the market.

There is nothing magical or mysterious about APIs. They are simply the latest technological advancements for driving successful business models. (Shetty, et al., 2014)

APIs can help companies make data available to the outside world or to selected business partners. These APIs can be used to create applications to market a company's products and develop new markets and opportunities. After APIs are established, they can be used to develop brand awareness and increase profit. Most importantly the APIs, which are a now a core part of the business also need to be treated as a product. Whether you or your company are considering using APIs, it is likely one of your competitors are. In our highly competitive world, this is a significant reason to consider adopting an API strategy. (IBM Redbooks, 2014)

To hide some of the complexity related to SOA and virtualize the publication of the service assets, API Management is considered to be used as a service interface. This API-centric, as-a-service delivery, is disrupting the consumption of business services as cloud disrupts the IT consumption model. Embracing the API Economy allows companies to both prepare and take advantage of the next generation platform, building apps at the edge of the enterprise, and positioning companies to open up possibilities.
This approach enables companies to take advantage of present and future developments in social platforms, wearable computing, mobility, and an ever-increasing shared economy. (Redguides for Business Leaders, 2014)

4.8.5 Services and integrations

When considered the basic premise of this project and ability to compose new logic from multiple pre-build components, it is evident that integrations and interoperability is at the core of the solution.

Web services based on the service-oriented architecture framework provide a suitable technical foundation for making business processes accessible within enterprises and across enterprises. But to appropriately support dynamic business processes and their management, more is needed, namely, the ability to prescribe how Web services are used to implement activities within a business process, how business processes are represented as Web services, and also which business partners perform what parts of the actual business process. (Leymann, 2002)

SOA Reference architecture is a good fit for the scope of this project, since interoperability between different platforms is achieved by exposing functionality as services. Following figure by The Open Group (2011) illustrates layers, especially related to choreography layer of process execution and services layer of re-usable components.

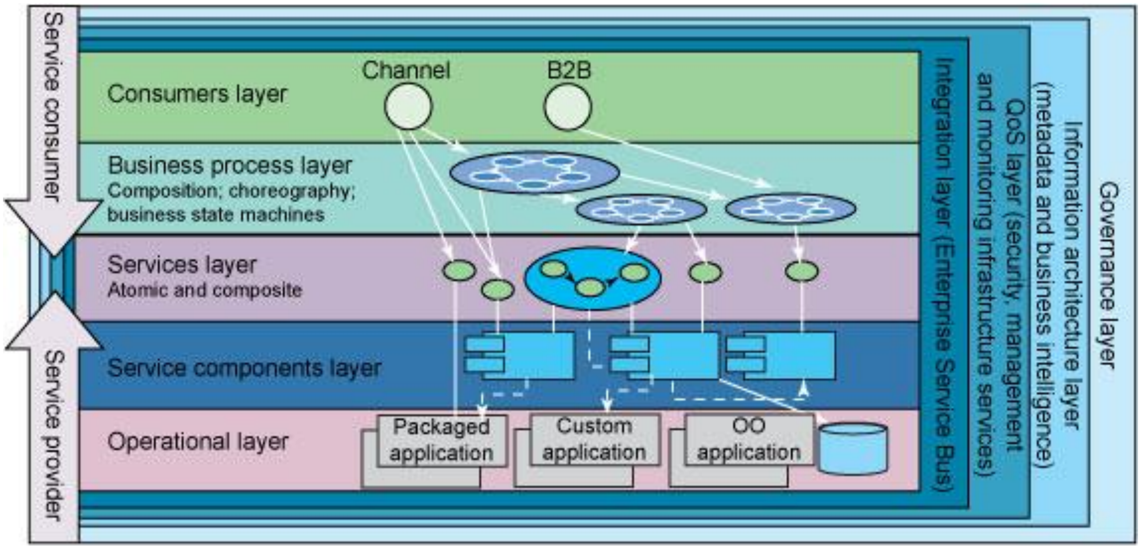

Service consumer
Service provider
Consumers layer
Channel
B2B
Business process layer
Composition; choreography; business state machines
Services layer
Atomic and composite
Service components layer
Operational layer
Packaged application
Custom application
OO application
Integration layer (Enterprise Service Bus)
QoS layer (security, management and monitoring infrastructure services)
Information architecture layer (metadata and business intelligence)
Governance layer

Figure 6 SOA Reference Architecture by the Open Group

SOA technology is known for its large number of existing standards and specifications that are still evolving and maturing. Apart from new standards and specifications, SOA also represents a software architecture shift which is a steep learning curve in itself. These perceptions of SOA complexity might have contributed to the fact that the respondents perceptions about complexity of SOA in relation to SOA use do not differ. While awareness of SOA complexity does not prevent organizations from initiating their SOA projects, it, nonetheless affects their perceptions about the success of those projects. (Maclennan & Van Belle, 2014)

4.8.6 Analytics and key performance indicators (KPIs)

Business process monitoring provides real-time access to critical performance indicators of ongoing processes and supports decision making and risk management during execution of the processes (Kang, et al., 2011). Typical Business Process Management Systems provide analytics and KPI capabilities as a core functionality on top of the process engine.

The process engine provides an environment for the execution of process instances, based on existing process definitions. The process logging component offers functions to track the progress of single process instances and to log corresponding audit data including data on occurred state transitions of process-relevant objects (e.g. processes, activities, resources, work items). The resulting audit data constitutes a rich source of low-level process events for subsequent analysis. (Janiesch, et al., 2012)

4.8.7 Rules and decisions

Tony Morgan defines a business rule in the book “Business Rules and Information Systems: Aligning IT with Business Goals” (Morgan, 2002) as a compact statement about an aspect of a business. The rule can be expressed in terms that can be directly related to the business, using simple, unambiguous language that’s accessible to all interested parties: business owner, business analyst, technical architect, and so on. He also refers to widely quoted definition of a business rule by GUIDE (an IBM-oriented industry user group): “a statement that defines or constrains some aspect of the business. It is intended to assert business structure or to control or influence the behaviour of the business” (Business Rules Group, 2001).

4.8.8 Business Processes

A business process is a collection of tasks and activities (business operations and actions) consisting of employees, materials, machines, systems, and methods that are being structured in such a way as to design, create, and deliver a product or a service to the consumer.

As such, a business process can be understood in the following way:

- It is a placeholder for the action (process area).
- An action is taking place (process group).
- A business task is taking place (business process).
- The location of the business task in the sequence (process step).
- The way the business task is carried out (process activity).

(Rosing, et al., 2014)

The main idea behind process management is to find out how a certain application should be performed. The assumption is that this application is composed of single activities which have to be executed in a certain order. A process describing this application will be defined and then implemented. (Cardoso & Aalst, 2009)

4.8.9 Introduction to used technologies

IBM BlueWorks Live

IBM Blueworks Live is a cloud-based business process modelling tool that lets you discover and model business processes and decisions for your organization. Using a web browser, you can collaborate with local and distributed teams and access the tool from anywhere.

With a Blueworks Live account, your organization can:

- Discover and document business processes and decisions using an easy-to-use interface.
- Use industry standards such as BPMN 2.0 (Business Process Modelling and Notation 2.0) for business processes and DMN (Decision Modelling and Notation) for decisions.
- Add custom attributes to the standard set of properties to document details that are specific to your organization.
- Create accessed-controlled spaces to collaborate within teams before publishing artefacts to the wider organization.
- Automate workflow and checklist style processes that are currently performed through email.
- Define and refer to a set of glossary item that represent the approved terms and definitions used in your organization.
- Analyse processes to discover areas for improvement.
- Use playbacks to step stakeholders through process flows, generating discussion and feedback on the process. (IBM, 2015)

IBM BPM on Cloud

IBM Business Process Manager is a comprehensive business process management platform. It provides a robust set of tools to author, test, and deploy business processes, as well as full visibility and insight to managing those business processes.

The components of IBM Business Process Manager provide a unified BPM repository to manage the business processes and their associated artefacts, tools for authors, administrators, and users, and a runtime platform. (IBM, 2015)

IBM BPM on Cloud provides a dedicated instance for your enterprise and makes your business processes visible in a cloud environment that is configured and ready to use. Logging in to IBM BPM on Cloud provides access to the IBM Business Process Manager application development tools for development, test, and business user environments.

IBM BPM on Cloud is licensed on a user-subscription basis. A user is someone who has access to the IBM BPM on Cloud service. A user can be a developer, tester, administrator, or a business user who uses process applications that are deployed and running on the service. (IBM, 2015)

IBM ODM on Cloud

IBM Operational Decision Manager is a full-featured, easy-to-use platform for capturing, automating and governing frequent, repeatable business decisions. It consists of two components, IBM Decision Center and IBM Decision Server. They form the platform for managing and executing business rules and business events to help you make decisions faster, improve responsiveness, minimize risks and seize opportunities.

IBM Operational Decision Manager improves the quality of transaction and process-related decisions and helps determine the appropriate course of action for customers, partners and internal interactions. It improves business insight and outcomes and helps you detect opportunities and risks. It also enables you to implement, test and deploy decision changes and understand how decisions are made and apply them consistently across processes and applications. (IBM, 2016)

IBM IIB on Cloud

IBM Integration Bus on Cloud offers deployment, visibility and management of integrations in a cloud environment.

It enables you to get started with IT asset integration quickly with a readily available, cloud-based environment hosted in IBM cloud data centres and managed by IBM.

IBM Integration Bus on Cloud delivers an environment that is both highly scalable and protected by IBM cloud security strategies to enable you to develop and deploy integrations that are ready when you are. (IBM, 2016)

IBM API Connect on Bluemix

IBM API Connect is a full-featured, streamlined and seamless management solution that addresses all aspects of the application programming interface (API) lifecycle for both on-premises and cloud environments. It offers capabilities to create, run, manage, secure and monetize APIs and microservices. Delivering an integrated user experience, it enables rapid deployment and simplified administration of APIs. (Hoover, 2016)

4.9 Summary of Conceptual Framework

The conceptual framework that builds on the maturity models and reference architecture utilizes the concept of Dynamic Model Types and Static Model Types. This allows customized demos to be built as dynamic and changing process models that are utilizing pre-built services that are considered static. This separation is based on the academic research of August-Wilhelm Scheer in the 1990s. In his book “ARIS - Business Process Frameworks” (1999) Scheer introduces the concept of ARIS house of business engineering (HOBE).

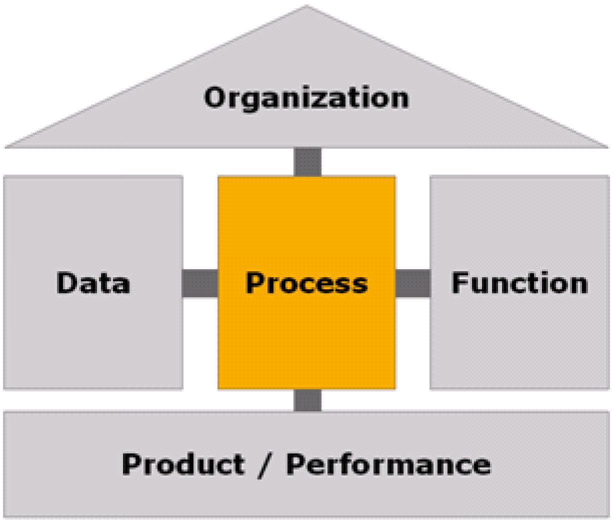


Figure 7 ARIS House of Business Engineering

The ARIS House concept clearly defines dynamic and static model types. All model types that belong to the center of the ARIS house (also known as Process-View) represent dynamic model types. All other views contain static model types (Data-, Organization-, Function-, and Product/Performance-View). (Schneider, 2010)

When building customized demos, the customization according the customer needs is done using process modelling. Customer process will show relevant information that can be put in a chronological, time dependent manner. This means that this model type is referred to as a "dynamic model type".

Static model types represent structures that do not provide time dependency. This includes the canonical data model and services that are pre-build. They are re-used and glued together in the process model.

Technically different components are linked together using assets built on IBM cloud tooling. Process diagram and executable process are dynamic process types. Business modelling will be done in BlueWorks Live SaaS tool. It holds capabilities to do process discovery and business process modelling. That is where the initial business process is modelled.

When business level of process modelling is final, the process model is imported into IBM BPM on Cloud tool. The final technical modelling and linking of dynamic process model to static services and data models is done in that tool. Integration services are exposed through the IBM API Connect –API management tool inside Bluemix iPaaS Platform. Business rules are modelled and executed in IBM ODM on Cloud platform.

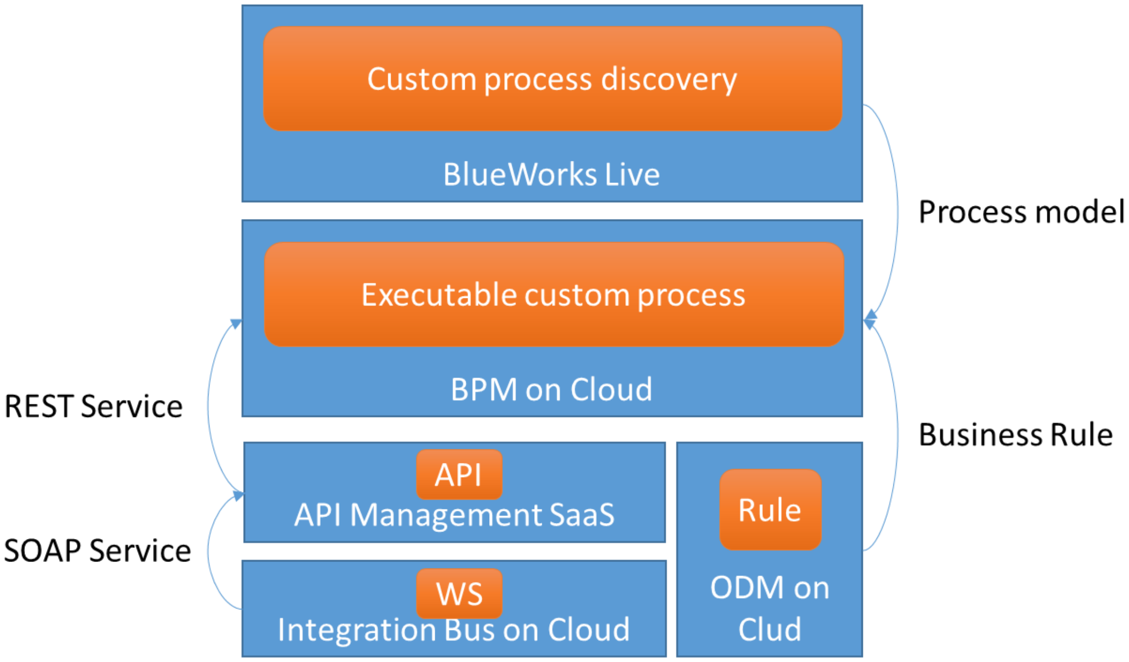


Figure 8 Technical interactions between solution artefacts

Different cloud based components are holding different types of artefacts that constitute fairly complex end-to-end solution. One of the successful approaches to manage complexity is separation of concerns (C. Ghezzi, 1991). Services expose standard interfaces to processes and hide unnecessary details of implementation. Each asset type is maintained in separate cloud based platform and linked to dynamic models in loosely coupled manner.

The process model is what is customized according to customer needs to obtain customer centric solution. Customized process model can be imported into execution environment. All necessary services and data models can be packaged into IBM BPM Toolkit. Toolkit is a container where artefacts can be stored for reuse by process applications or other toolkits. Toolkits can also share components with other toolkits. Diagram below outlines these relationships:

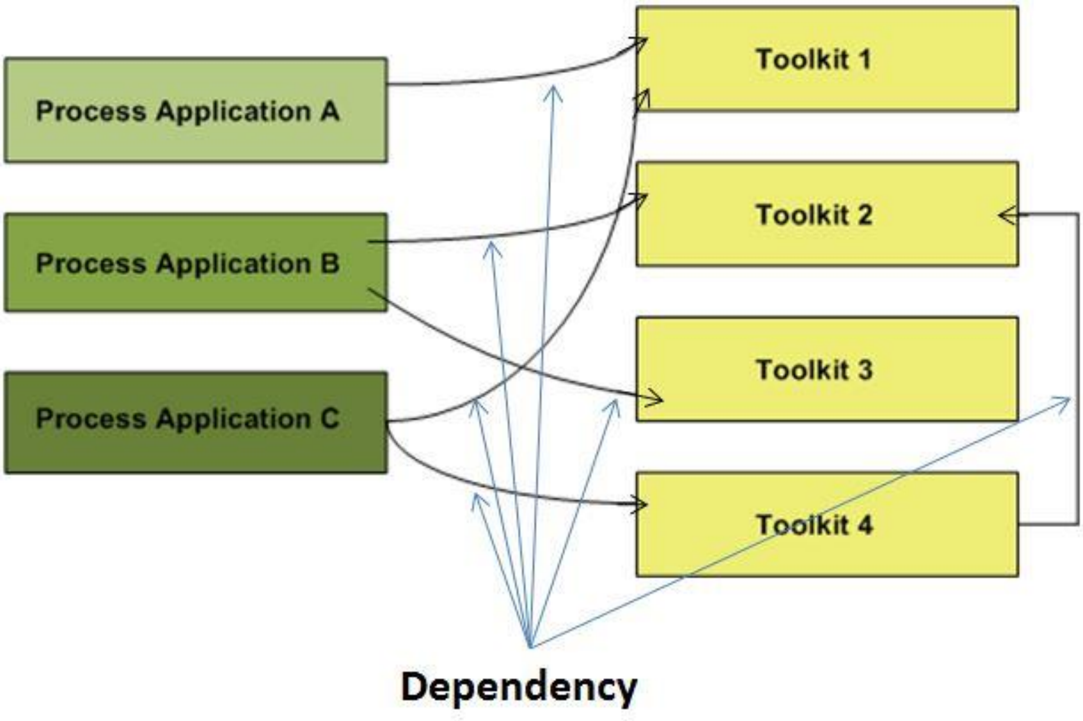
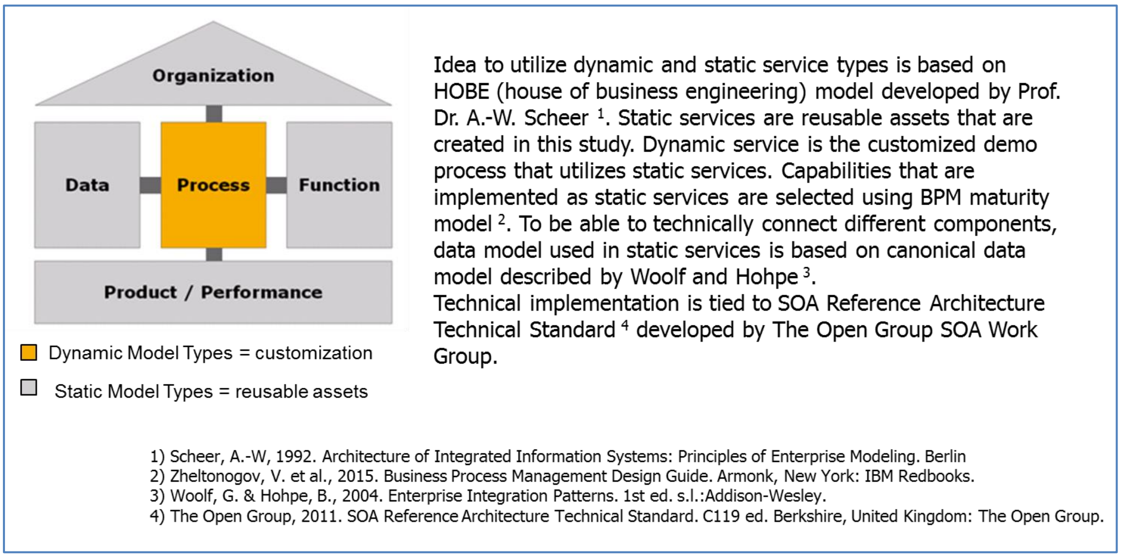


Figure 9 Toolkit dependency

This allows to quickly build highly customized demos by importing toolkit holding connectors for services and needed data models. Using this kind of approach the customization is done purely on the process modelling level.

Following picture illustrates the summary of conceptual framework and the key references.


Organization
Data
Process
Function
Product / Performance

🟧 Dynamic Model Types = customization

□ Static Model Types = reusable assets

Idea to utilize dynamic and static service types is based on HOBE (house of business engineering) model developed by Prof. Dr. A.-W. Scheer [1]. Static services are reusable assets that are created in this study. Dynamic service is the customized demo process that utilizes static services. Capabilities that are implemented as static services are selected using BPM maturity model [2]. To be able to technically connect different components, data model used in static services is based on canonical data model described by Woolf and Hohpe [3].
Technical implementation is tied to SOA Reference Architecture Technical Standard [4] developed by The Open Group SOA Work Group.

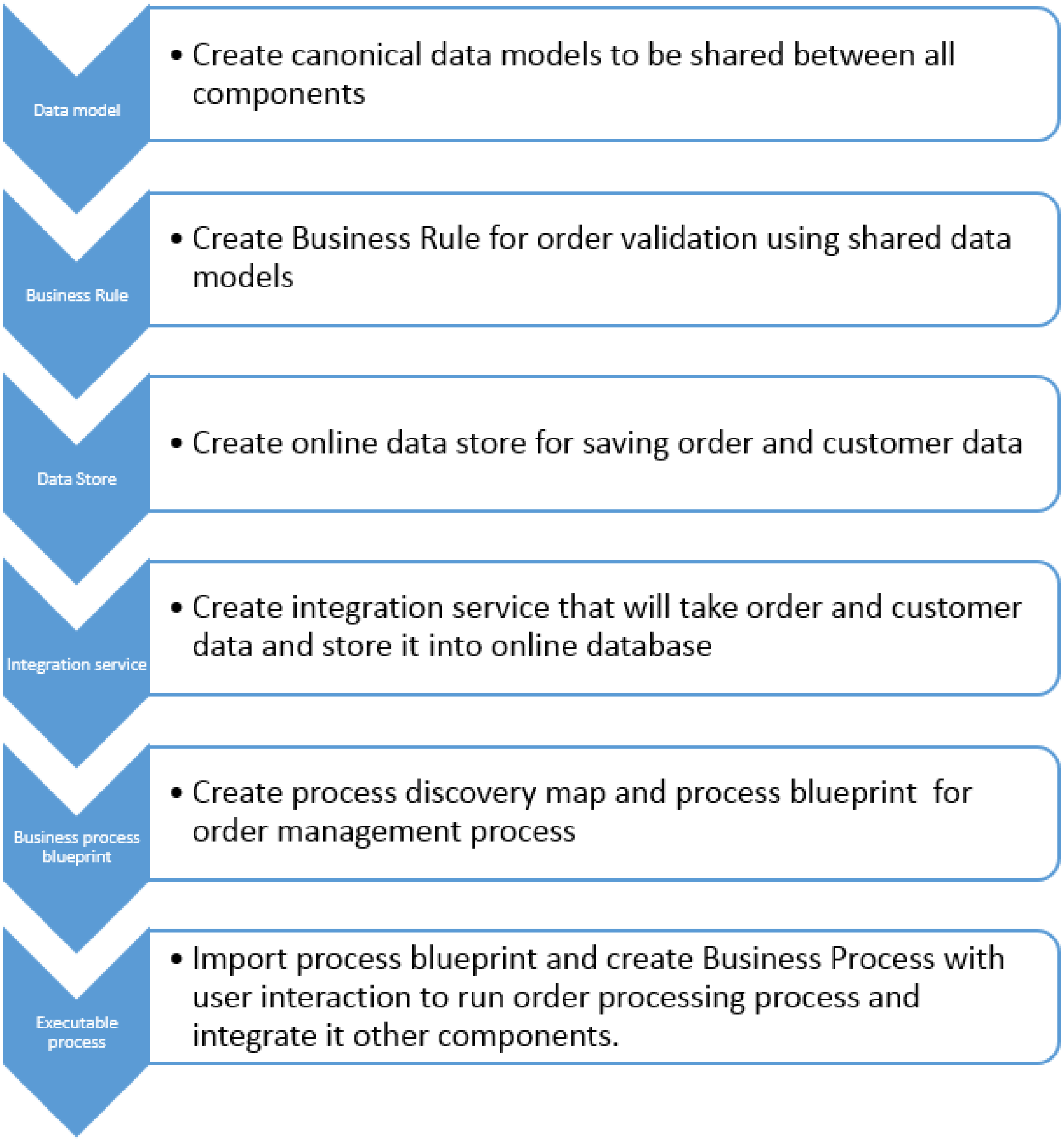
1) Scheer, A.-W, 1992. Architecture of Integrated Information Systems: Principles of Enterprise Modeling. Berlin
2) Zheltonogov, V. et al., 2015. Business Process Management Design Guide. Armonk, New York: IBM Redbooks.
3) Woolf, G. & Hohpe, B., 2004. Enterprise Integration Patterns. 1st ed. s.l.:Addison-Wesley.
4) The Open Group, 2011. SOA Reference Architecture Technical Standard. C119 ed. Berkshire, United Kingdom: The Open Group.

Figure 10 Summary slide of the conceptual framework

5 Establishing re-usable services and related operating model

5.1 Implementation plan

As described in Introduction, the output of this study will be a set of reusable services and related operating model. Before it's possible to build the services, it's necessary to create an implementation plan to improve interoperability between different components. It starts with designing a data model that is the key aspect in communications between different tools and components. By using canonical data model in all components the need for doing transformations and data mapping in interactions. It also allows to create and update components independently. Using standards in communications allows the full solution to be loosely coupled and thus to be more robust. It also allows to change each components easily with only minor changes into process.



Data model

- Create canonical data models to be shared between all components

Business Rule

- Create Business Rule for order validation using shared data models

Data Store

- Create online data store for saving order and customer data

Integration service

- Create integration service that will take order and customer data and store it into online database

Business process blueprint

- Create process discovery map and process blueprint for order management process

Executable process

- Import process blueprint and create Business Process with user interaction to run order processing process and integrate it other components.

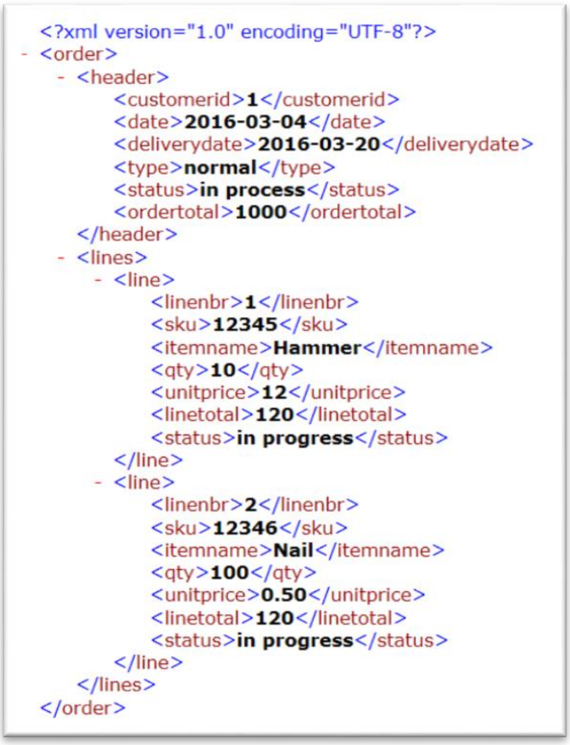
5.2 Creating the common data models

The aim for this thesis was to build a functional set of reusable software services that can be integrated together in executable business process. To make all of these services to function together, the first thing to define is the common data model. The decision that data model would contain customer and order object was made on the assumption that majority of customized demos would be covered by variating and expanding these data models.

After going through the existing common data models for needed objects, the decision to simplify and flatten those models was made. By only leaving the needed data fields and flattening the structure, data models were made more simple and utilizing them in implementations easier.

Data models were created in xml schema format. That is a typical data description format and highly interoperable between different systems.

Below is a sample message of order data in XML format after the data model is simplified and flattened.



```
  <?xml version="1.0" encoding="UTF-8"?>
- <order>
  - <header>
        <customerid>1</customerid>
        <date>2016-03-04</date>
        <deliverydate>2016-03-20</deliverydate>
        <type>normal</type>
        <status>in process</status>
        <ordertotal>1000</ordertotal>
    </header>
  - <lines>
    - <line>
          <linenbr>1</linenbr>
          <sku>12345</sku>
          <itemname>Hammer</itemname>
          <qty>10</qty>
          <unitprice>12</unitprice>
          <linetotal>120</linetotal>
          <status>in progress</status>
      </line>
    - <line>
          <linenbr>2</linenbr>
          <sku>12346</sku>
          <itemname>Nail</itemname>
          <qty>100</qty>
          <unitprice>0.50</unitprice>
          <linetotal>120</linetotal>
          <status>in progress</status>
      </line>
    </lines>
  </order>
```

Figure 11 Order message sample

5.3 Authoring Business Rules

When developing business rule application, first I created a rule project that contains rule artefacts, Business Object Model (BOM), vocabulary, and reference to the execution object model (XOM).

Tool used for developing these artefacts is Rule Designer that is Eclipse based development tool. It has a Decision Service Map view that guides the user through different tasks to develop Decision Service.

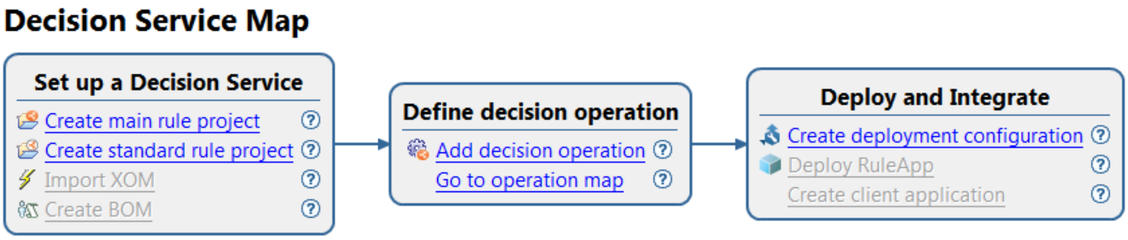


Figure 12 Decision Service Map in Rule Designer

Based on the data models created, execution object models (XOM) was created by importing XML schema files into the Rule Designer.

Next, a business object model (BOM) was defined. In this case it was automatically created by parsing the execution object model (XOM). After the creation it was possible to write rules from the vocabulary terms that are contained in the BOM.

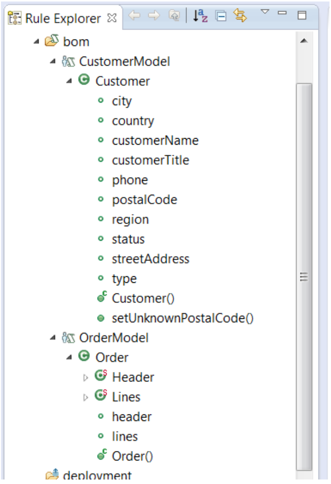
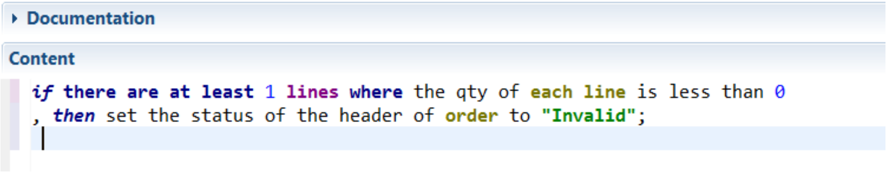


Figure 13 BOM's in Rule Designer

Next two simple rules was created to demonstrate how simple it is to author rules in Business Rule Language. The other rule is used to check customer type, and the other to validate that there are no lines that have order quantity less than zero. This is how the rule looks in Rule Designer.



```
if there are at least 1 lines where the qty of each line is less than 0
, then set the status of the header of order to "Invalid";
```

Figure 14 Order validation rule in Rule Designer

After the rules were authored, they were deployed to the rule execution server for runtime execution.

5.4 Implementing Data Storage

The Cloudant NoSQL Database service was deployed into IBM Bluemix and used to store JSON data accessible via easy-to-use RESTful HTTP/S APIs. Orders database was created to store order data in JSON format.

Graphical Node-RED was used for wiring together APIs and online Cloudant Database.

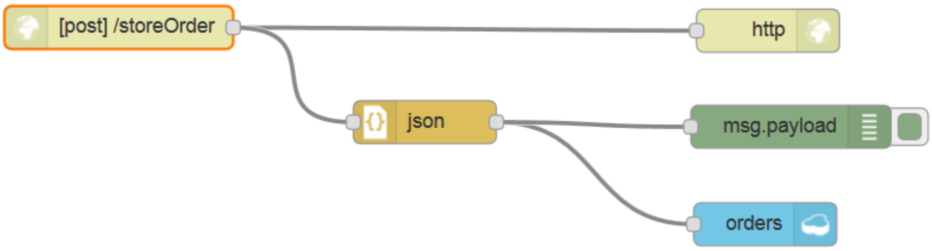
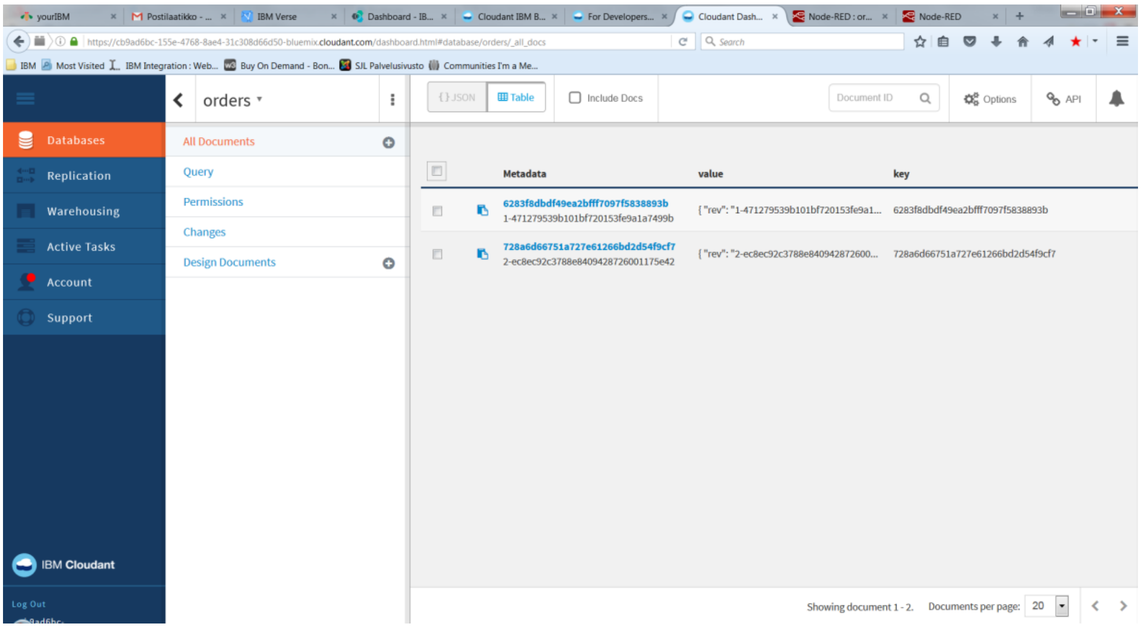


Figure 15 Node-Red application in Bluemix

The Node-RED application receives message in HTTP POST payload. Then, payload is parsed into JSON object and inserted into Cloudant NoSQL Database. Data can be accessed using web-UI.



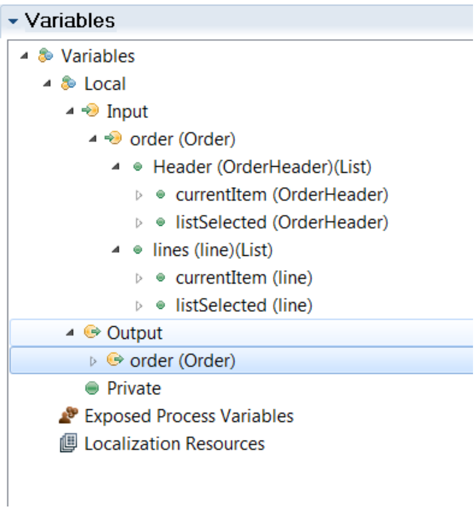
Metadata	value	key
6283f8dbdf49ea2bfff7097f5838893b 1-471279539b101bf720153fe9a1a7499b	{ "rev": "1-471279539b101bf720153fe9a1...	6283f8dbdf49ea2bfff7097f5838893b
728a6d66751a727e61266bd2d54f9cf7 2-ec8ec92c3788e8409428726001175e42	{ "rev": "2-ec8ec92c3788e840942872600...	728a6d66751a727e61266bd2d54f9cf7

Figure 16 Cloudant UI showing order documents

5.5 Creating Integration service

Next, integration service was created. For simple creation of the service, the stub implementation of the process was created to be imported into IBM Integration

Toolkit. This empty implementation with parameter definitions helps to create service signature automatically. This process definition is added to toolkit package and exported. When imported into IBM Integration Toolkit developer tool, this information can be used to create integration service.



Variables

- Variables
  - Local
    - Input
      - order (Order)
        - Header (OrderHeader)(List)
          - currentItem (OrderHeader)
          - listSelected (OrderHeader)
        - lines (line)(List)
          - currentItem (line)
          - listSelected (line)
    - Output
      - order (Order)
    - Private
  - Exposed Process Variables
  - Localization Resources

Figure 17 Integration Service variables in Process Designer

The integration service interface was automatically created using Integration Toolkit. What remained to do, was to implement integration logic inside the message flow. A simple integration was build using standard capabilities of the IBM Integration Bus. Integration service will utilize SOAP Web Service capabilities as an interface.

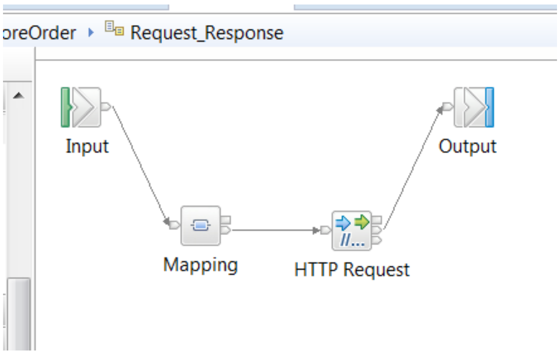
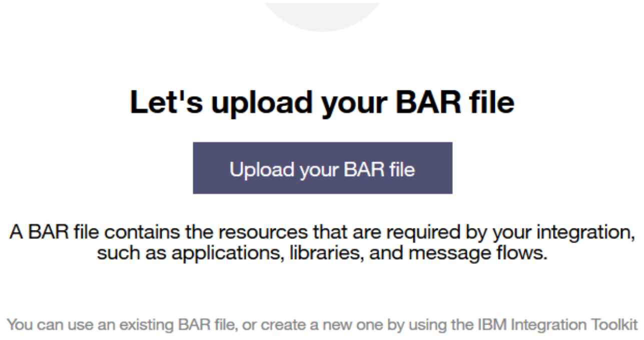


Figure 18 Integration Service implementation in Integration Bus

Message is first parsed from Input node, then it’s converted into JSON object. That object is then passed in REST call to Node-RED application that will store data as JSON document into Cloudant Database.

When implementation of the integration service is final, the next step is to deploy the service to IBM Integration Bus on Cloud. This is done by uploading the deployment archive file called BAR-file to the cloud platform.



Let's upload your BAR file

A BAR file contains the resources that are required by your integration, such as applications, libraries, and message flows.

You can use an existing BAR file, or create a new one by using the IBM Integration Toolkit

Figure 19 Uploading the BAR file to IIB on Cloud

When BAR-file is uploaded, it's possible to manage integration service in the IIB on Cloud. The public endpoints are shown in the portal view, and it is possible to start and stop integrations from the UI.

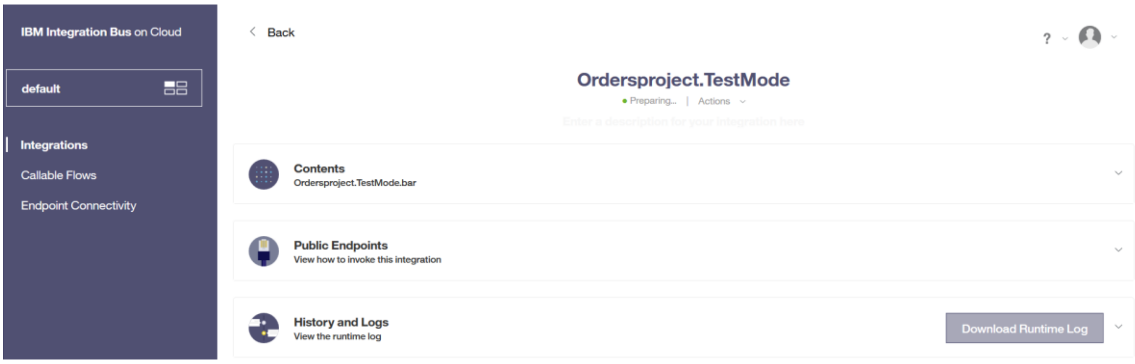


Figure 20 Integration project in IIB on Cloud

When integration service is available in the IIB on Cloud platform, it's possible import service to API Connect platform, so it's possible to add management layer for access control, rate limiting and security enforcement. This is done in Bluemix platform that contains IBM API Connect service.

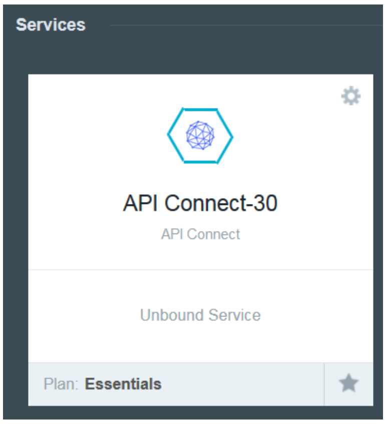
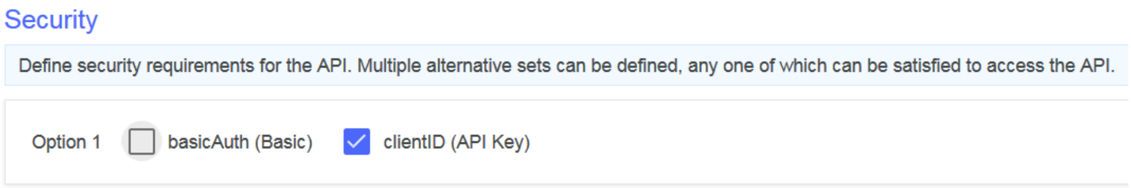


Figure 21 API Connect service in IBM Bluemix

When we configure settings for integration service API, it's possible to define security definitions that are enforced in runtime. In this case we only want to use API keys to authenticate calling application, as configured in Figure 22 Security settings for accessing API.



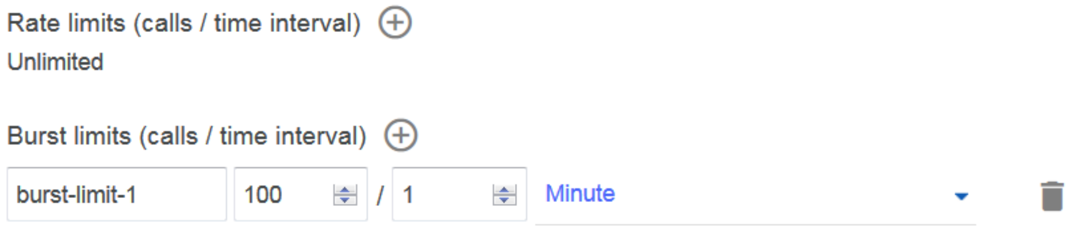
Security

Define security requirements for the API. Multiple alternative sets can be defined, any one of which can be satisfied to access the API.

Option 1 ☐ basicAuth (Basic) ☑ clientID (API Key)

Figure 22 Security settings for accessing API

In API Connect it's also possible to define rate limits to limit number of calls to the API. It's not needed to limit calls for this API, but just to secure API from the Denial-of-Service type of attacks, the burst limit is set to 100 calls / minute.



Rate limits (calls / time interval)

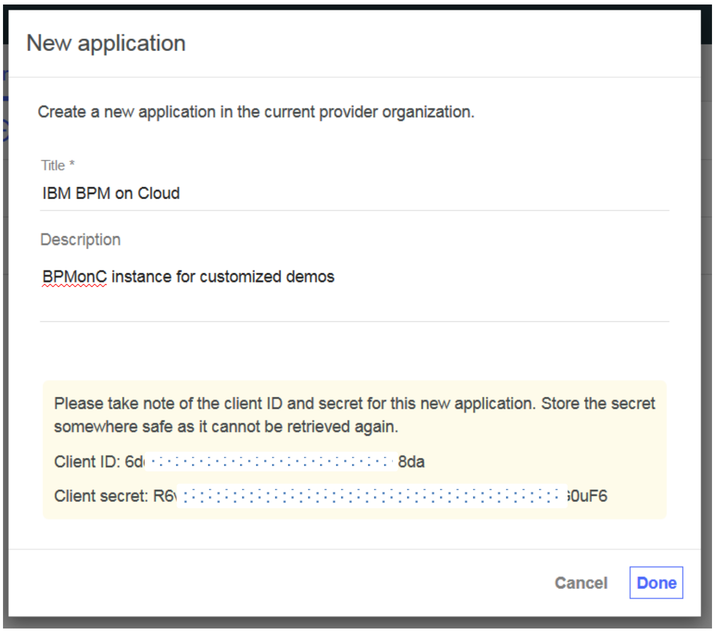
Unlimited

Burst limits (calls / time interval)

burst-limit-1 | 100 / 1 | Minute

Figure 23 Burst limit settings in API Connect

Since the settings requite API Keys (clientID) to be used to access the integration service interface through the API Connect, new application needs to be registered. Typically it's done using developer portal where the information about the API is published. In this case we can register new application using API Connect interface, as shown in Figure 24 Registering new application in API Connect.



New application

Create a new application in the current provider organization.

Title *

IBM BPM on Cloud

Description

BPMonC instance for customized demos

Please take note of the client ID and secret for this new application. Store the secret somewhere safe as it cannot be retrieved again.

Client ID: 6d[redacted]8da

Client secret: R6[redacted]0uF6

Figure 24 Registering new application in API Connect

When registering new application, the platform creates API Keys for that application (hidden in screenshot). Those are needed when calling service in API Connect.

5.6 Sample Process

As described in the conceptual framework, process is a dynamic type that will make use of static types like data models and integration services. Therefore, a simple process was designed for testing the static components. It contained discovery map and process diagram for the sales order process. Discovery map describes milestones and activities contained in those milestones.

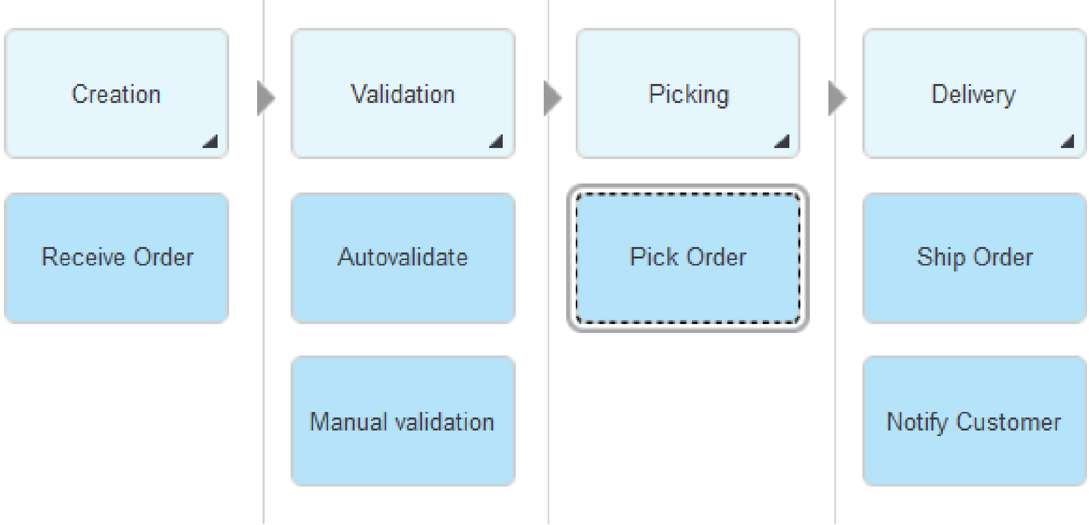


Figure 25 Sales Order discovery map in Blueworks Live

A process diagram is created based on the contents of the discovery map, including flow lines, swim lanes, and start and end events. It displays more details about the flow of the process, including alternate paths and swim lanes, in BPMN notation.

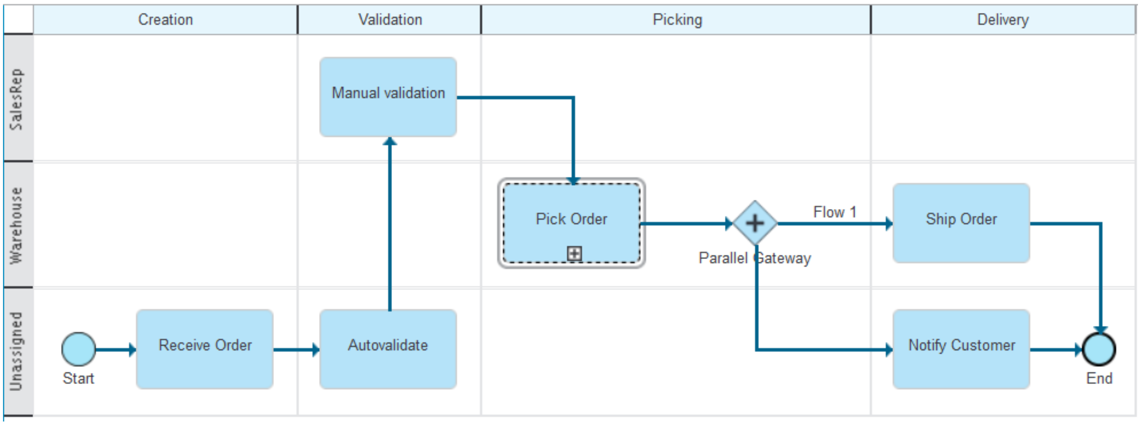


Figure 26 BPMN notation of the process in Blueworks Live

This process diagram was then imported into IBM Process Designer tool for building the actual executable process model. In the import the layout of the process is slightly changed, so first thing was to change layout of the process to more user friendly. Next some changes and additions needed to be made. The data models created for customer and order were defined. Those data models was then used as process variables that will carry runtime information across the execution of a process instance at execution time.

Those variables can then be used to call business rule in IBM ODM on Cloud and integration service in IBM Integration Bus on Cloud (proxy in API Connect). In the process model in IBM BPM those re-usable components are marked in green just to highlight integration points to external systems.

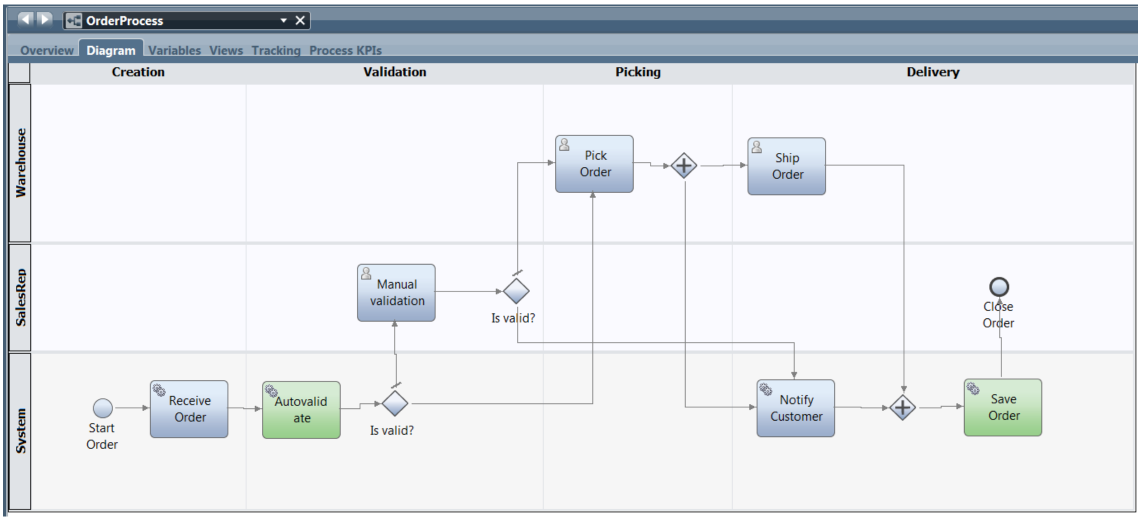
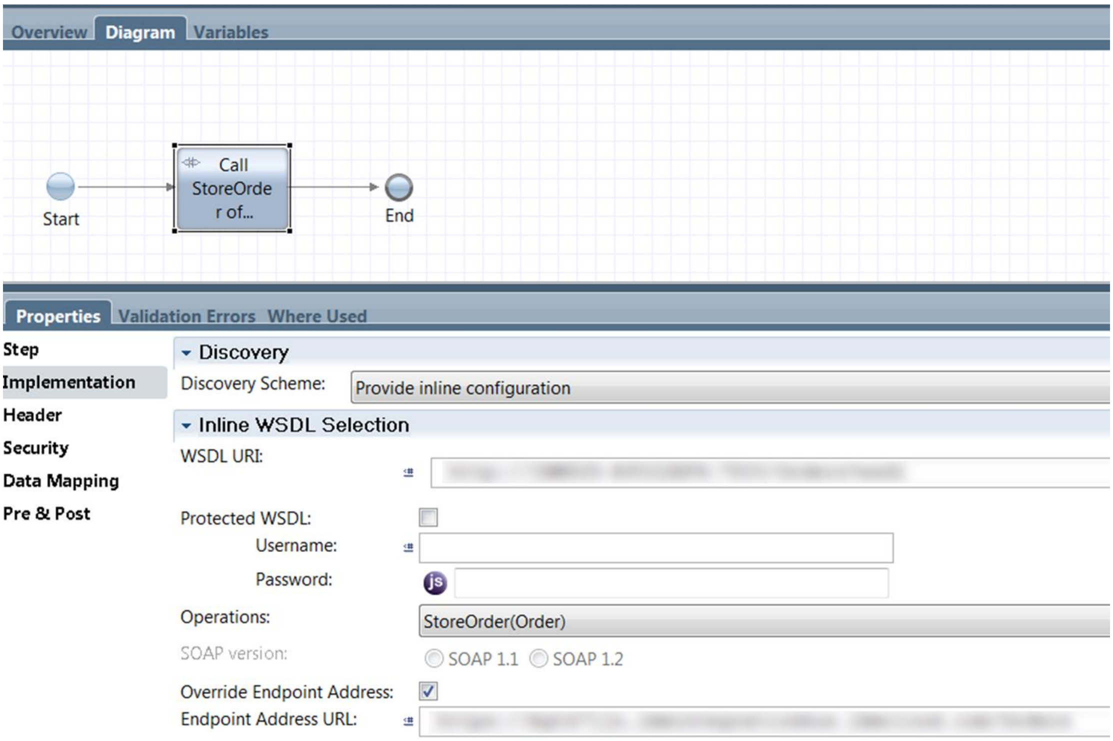


Figure 27 Business Process definition in Process Designer

Integrations are packed into system tasks as activities, where the implementation information is already entered. It describes how to connect to services that are pre-built and re-usable. This also allows to package service definitions into the Process Designer Toolkit and that Toolkit to be imported into the Process Designer when customer specific process implementation is built.



Discovery

Discovery Scheme: Provide inline configuration

Inline WSDL Selection

WSDL URI: [redacted]

Protected WSDL: □

Username:

Password:

Operations: StoreOrder(Order)

SOAP version: ○ SOAP 1.1 ○ SOAP 1.2

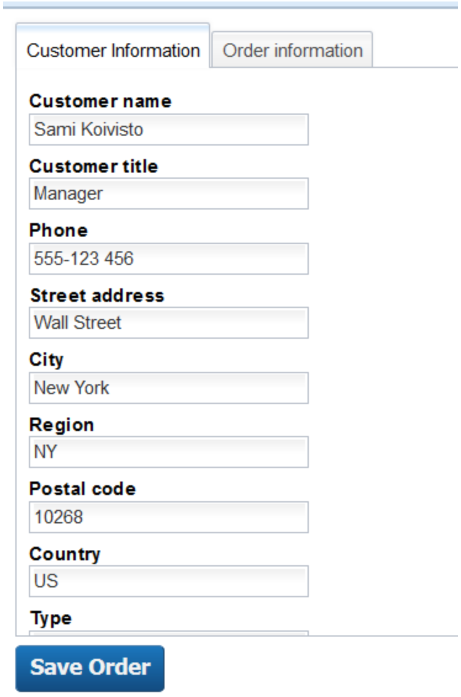
Override Endpoint Address: ☑

Endpoint Address URL: [redacted]

Figure 28 Integration service activity

When building user interfaces for process executions, IBM Coach Framework is used.

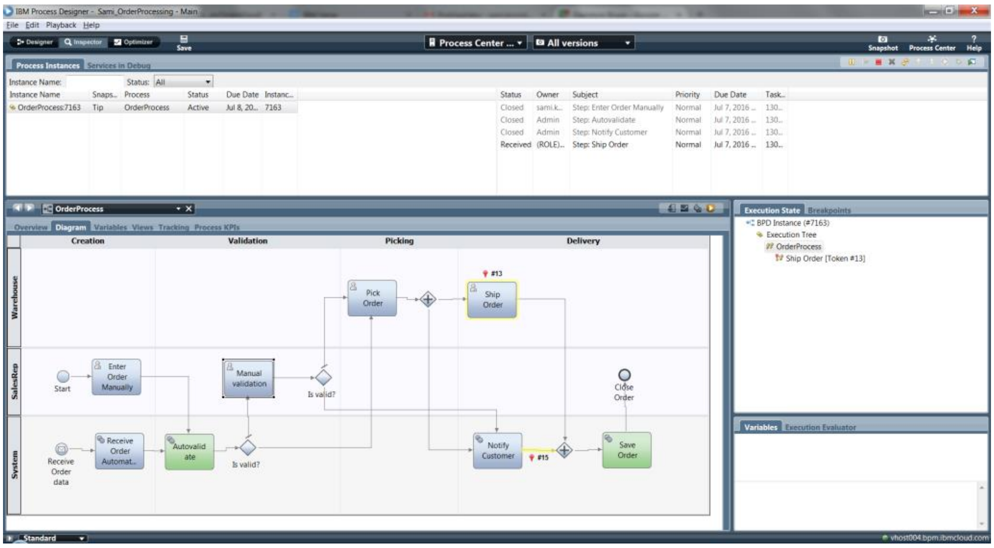
The IBM Coach Framework is a key element of the IBM Business Process Manager (BPM) product suite. With the Coach Framework, process authors can create and maintain custom web-based user interfaces that are embedded within their business process solutions. (Reynolds, et al., 2014)



Customer name	Sami Koivisto
Customer title	Manager
Phone	555-123 456
Street address	Wall Street
City	New York
Region	NY
Postal code	10268
Country	US
Type	

Figure 29 Sample UI build using Coach Framework

The Inspector in IBM Process Designer can be used for an iterative approach to process development. The idea is to use the Inspector to demonstrate current process design and implementation in playback sessions. Playback sessions are designed to capture important information from different stakeholders in a process, such as management, end users, and business analysts. Taking an iterative approach to process development ensures that your process applications meet the goals set for the customized demos.



Instance Name: Status: All

Instance Name	Snaps...	Process	Status	Due Date	Instanc...
OrderProcess:7163	Tip	OrderProcess	Active	Jul 8, 20...	7163

Status	Owner	Subject	Priority	Due Date	Task...
Closed	sami.k...	Step: Enter Order Manually	Normal	Jul 7, 2016 ...	130...
Closed	Admin	Step: Autovalidate	Normal	Jul 7, 2016 ...	130...
Closed	Admin	Step: Notify Customer	Normal	Jul 7, 2016 ...	130...
Received	(ROLE)...	Step: Ship Order	Normal	Jul 7, 2016 ...	130...

BPD Instance (#7163)
Execution Tree
OrderProcess
Ship Order [Token #13]

Figure 30 running a sample process in Inspector view

At this point all initial technical components are built and deployed. Next phase is to create related operating model how to deliver customized demonstrations using these components.

5.7 Related operating model

As one of the outputs of this study is related operating model. Operating model describes how reusable assets that we built and are described in previous chapter can be re-used in building a customized demo. Operating model assumes understanding on process modelling theory and notations, as well as technical knowledge of the related components. This is meant to guide subject matter experts and technical sellers, so it’s not meant to be step-by-step guide for building demos.

The technical implementation of the components allow them to be separately maintained. This was one of the theoretical findings in conceptual framework. Data models and service connectors are packed into toolkit in IBM BPM. Toolkits enable Process Designer users to share library items across process applications. Process applications can share library items from one or more toolkits, and toolkits can share library items from other toolkits.

When creating new customized demo that will use these shared components, users who have access to the toolkit can create a dependency on the toolkit and use the library items within it for their process development efforts.

By using pre-defined data models and services user can ensure that the interoperability between components is seamless. It also reduces the time to develop process application and therefore highlights the value of business process management as agile discipline for building business applications. When packaged into toolkits, data models and service integrations already implicitly utilize same data format.

The process of building customized demo with re-usable component starts from the process discovery. Typically this is done with the customer using Blueworks Live and its process discovery map. First the milestones and process activities are planned and modelled with the customer. When all steps are identified and relevant meta-data is inserted into the discovery map, it can be generated into process diagram in BPMN notation. At this point the decisions and additional routing of process execution is added. There are some best practices how to model your process for execution. It makes sense at this point to pay attention to these points, so the transition from modelling to actual execution is smoother.

For good process modelling practices the idea is to model business process for “good fit” and follow the “Rule of Seven” at each level. The “rule of seven” in the context of process modelling is defined in the book “Scaling BPM Adoption: From Project to Program with IBM Business Process Manager” (Dyer, et al., 2012) is defined in simple guidelines: keep the number of activities on your process diagram at any particular level of granularity down to seven or less. Beyond seven distinct activities on a process diagram, a business user loses the ability to understand the diagram or its intent.

It’s particularly important in these demonstration scenarios that we are able to comprehend and communicate each level in 5 minutes or less. Each activity should be able to summarize “work performed” as a goal with an outcome.

When modelling the process for customized demo, it’s important to pay attention to certain anti-patterns, like String-of-Pearls Pattern. This is a term used in process modelling context to indicate a sequence of two or more activities in the same swim lane. While an activity may have multiple steps within it, these steps are not process steps. Although performing this activity may include multiple steps and multiple screens, this is a single activity, and should be combined into a single Activity in process diagram. This helps in building the executable process model and makes business model and technical model more consistent.

After the business model in BlueWorks Live is viable, it should be imported into Process Designer tool as a technical model for authoring. In Process Designer, the prepackaged data model and integration activities can be added to process model as toolkit dependency. When you create a dependency on a toolkit, you can use the library items from that toolkit for the implementation of your current project. After you create a dependency on a toolkit that includes services, the Designer view automatically makes those services available when you choose the implementation of activities.

Authoring processes is an iterative process that can involve several playback sessions. Typically some user interfaces are created based on the data model packaged in the toolkit. It is possible to use different data labels in user interface than in the data models. This adds flexibility to the customization of the demo process without the disadvantages it might cause in live production systems.

When customized process is ready to be demonstrated in front of the customer, it’s recommended to use playback session methodology to present the process. You can use the Inspector to demonstrate current process design and implementation in playback sessions. You can run your process or service in the Inspector to show your flow and to demonstrate how individual tasks are running as expected. You can act as different users if you want to show the access that you set up for your users. You can also use Process Portal to work on your assigned tasks and collaborate with others to demonstrate how to complete the work efficiently and use dashboards to view the performance of individuals, teams and processes. In typical demo use cases it’s not needed to deploy processes to run time environment, where they can be used by your customers.

5.8 Summary of implementation project and key outcome

As the solution is composed using multiple different tools, platforms and technologies, it was vital to do proper planning and design before entering actual implementation. Since conceptual framework well defined different technological layers, the most important design pattern was in what order the implementation should be done. Based on the requirement of interoperability, the implementation started from the data model. When data model was designed and implemented, the implementation order of business rules and integration service was irrelevant. When those assets were available, it was necessary to a build simple process model to test how these assets work together. Based on that test the related operating model was created. It contains information and tips on how to best utilise the assets in demo setup. Implementation work itself involved many different tools and platforms and utilised many different platforms. That made it very versatile and thus interesting and very suitable for thesis.

Key outcome for the implementation project is reusable services and related operating model. Following diagram illustrates the key outcomes in component view.

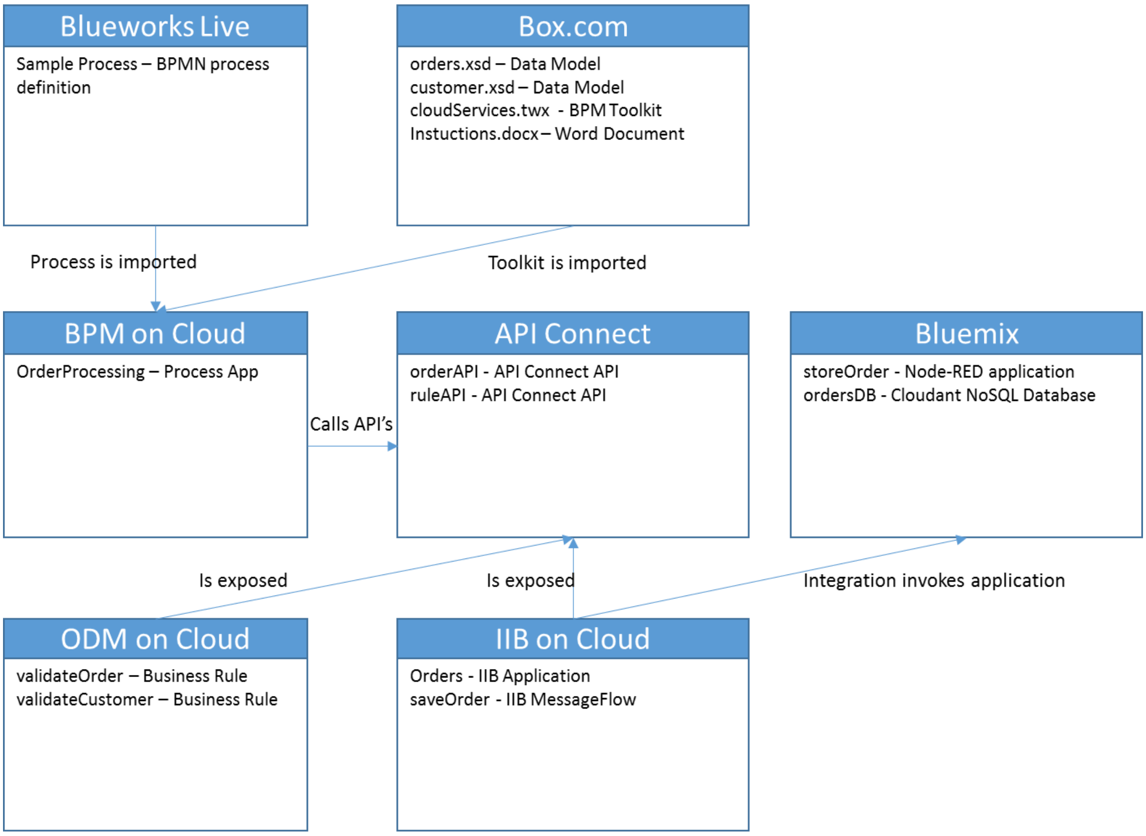

Blueworks Live
Sample Process – BPMN process definition
Box.com
orders.xsd – Data Model
customer.xsd – Data Model
cloudServices.twx - BPM Toolkit
Instuctions.docx – Word Document
Process is imported
Toolkit is imported
BPM on Cloud
OrderProcessing – Process App
Calls API's
API Connect
orderAPI - API Connect API
ruleAPI - API Connect API
Bluemix
storeOrder - Node-RED application
ordersDB - Cloudant NoSQL Database
Is exposed
Is exposed
Integration invokes application
ODM on Cloud
validateOrder – Business Rule
validateCustomer – Business Rule
IIB on Cloud
Orders - IIB Application
saveOrder - IIB MessageFlow

Figure 31 Component view of the reusable services

Diagram displays reusable services and related operating model, where they are deployed and stored and how they are related.

6 Pilot implementation

6.1 Pilot implementation use case

The pilot implementation was done for a customer demo. Due to the schedule of this thesis, the pilot implementation was done for a use case where the requirements were not clearly defined. Customer was looking of ways to improve and fasten the delivery of applications in general. This allowed a simple process implementation similar to where components were tested in previous phase. Only the naming and order of the steps changed.

Method for delivering a customized demo was related to Process Improvement Discovery workshop methodology. It was delivered as a one day solution development exercise led by team of Smarter Process experts with business and IT team from customer. It was used to validate if and how IBM Smarter Process products can solve customer's problems and provide some added-value for the project. After the discovery session IBM provided a summary session and written document of findings and recommendations.

Typical stage in sales cycle for Process Improvement Discovery Workshop is when customer has pains but there is no single use case identified yet or use case needs to be qualified and investigated. Goals for PIDW are to explain and demonstrate IBM capabilities around process improvement, develop an IBM solution and discover business values for the customer, understand the project and the customer environment and to design a high level solution and architecture.

In the discovery session IBM experts first introduce the basic concepts of BPM and process improvement methodology. Then customer introduces the use case or improvement area. Typically in this part of the workshop there is an involvement from the business side from the customer. When use case in understood, the team together starts to sketch a solution and turn that into technical architecture. This stage involves more IT experts from the customer side. Idea is that solution is approved by business and technical architecture is approved by IT, making the outcome comprehensive and tangible.

Outcome of the discovery session contained a written document with key findings: business problem overview, current and desired situation. It also listed prioritized areas of improvement. Another key outcome is to describe a potential benefit of using IBM solutions for these areas. Document also described high level solution architecture overview, recommendations, risks and target architecture.

After the discovery session IBM team wrote the summary document, and also built customized demo to demonstrate high level solution to the business problem. It was not intended to represent the final or comprehensive solution, but rather to show BPM platrorms ability to solve the problem. Customized demo presentation was a part of the summary session. As the design time template for a process, a sample Sales Order process from BlueWorks Live Template repository was used. Business process model from the BlueWorks Live was imported into the Process Designer along with the toolkit that contained implementations for activities to call pre-made services. Those were then linked into the process diagram that was deployed as executable process. Simple UI's were built that utilized the defined data model. Playback methodology was used to demonstrate the implemented process. Process was ran using the Process Designers Inspector view. It allows to follow the flow of the process as it's executed and also to open user tasks implemented in Coach Framework.

6.2 Results from pilot implementation

Results from the pilot implementations were positive. Customized demo was a part of Process Improvement Discovery Workshop that is highly interactive way to discover the process that will be demonstrated. Running processes in feedback sessions using playback methodology enables to discuss other, not yet implemented, capabilities that tooling offers. Due to the limited scope of the pilot project and the fact that the use case for favourable for this type of implementation the ability to utilize pre-built services was high. Another favourable aspect was the fact that the customer had not that specific use case defined. This means that all of the services needed were already build. This gave extra time to focus on the process implementation and user interfaces in the human tasks within the executable process. This makes it more compelling to the customer, when also the look-and-feel of the solution matches their standards.

When considering how technical components worked, it was noticeable that with modular structure that allows to mix multiple languages, development frameworks and data-storage technologies, services were easier to deploy and since they are autonomous, they are less likely to cause full system failures when they go wrong.

Results show certain benefits of the solution:

- Modular structure from service components
- Each service can be organized around needed business capability
- Each service can be implement using best possible tools for the job
- Infrastructure allows decentralized maintenance and governance
- SaaS platforms relieved from infrastructure maintenance

Chapter "Data collection topics" contains a table that is based on the data collected from the interviews, a comparison of different deployment options and their challenges. It shows different delivery types and pros and cons for each of them. Based on the feedback from the pilot implementation the following comparison was made for the delivery type developed in this project.

Delivery type	Pros	Cons
Pre-built cloud services	• No setup time • No download time • No platform maintenance needs • Planned for interoperability • Automatic scaling of services • No need for versioning • No need to store assets • Related operating model • Planned for multiproduct demos • Easy to scale horizontally and vertically • Unlimited possibilities to introduce new services and capabilities	• No control over the platform updates • Out-of-sync version updates • Involves multiple different technologies and tools • Possible inconsistency between artefacts

6.3 Final software services and related operating model

There was some changes to set of services after the feedback from the pilot implementation. The initial set is almost the same that was built in the project phase that is described in chapter 5. Only addition was that in the first phase only integration service was published as an API. But after the feedback from the pilot implementation also the rule services are published as proxy API's in the IBM API Connect tool in Bluemix platform. This allows to maintain the all service interfaces separated from the actual implementations and therefore mitigating the risk of compatibility issues when platforms are updated.

List of implemented assets:

- orders.xsd – Data Model
- customer.xsd – Data Model
- validateOrder – Business Rule
- validateCustomer – Business Rule
- storeOrder - Node-RED application
- Cloudant - NoSQL Database
- Orders - IIB Application
- saveOrder - IIB MessageFlow
- orderAPI - API Connect API
- ruleAPI - API Connect API

The final operating model documentation, toolkit and connection details for services are stored in Box.com. Box.com is an online file sharing and content management service for businesses.

7 Discussion and conclusions

7.1 Summary of the project

The project was launched to find practical solution for common problem of how to fast build customized and compelling software demos. The focus area is around software tooling that is platform type in its nature. This means that these middleware software solutions typically possess a set of capabilities that are used to custom build actual implementation on top of the platform. This makes it difficult to demonstrate the capabilities in an effective way. Delivering an engaging experience early in the sales cycle can determine which solution wins the business. That is why this type of process improvement has direct financial impact. Basic premise was to utilize business process as the customization layer and technology from most typical use cases as integrated prebuilt services.

The project started off by building the understanding how customized demos are built currently, and what technical tools are used to help building them. Before the actual current state analysis, scoping was done to define bounds for the analysis. After the scoping current demo offering data was collected and analysed. Next step was to understand benefits and challenges related to different technical approaches. Main focus was to investigate how fast it is to create customization based on existing assets.

As the literature review best practises for reusability in software sales were investigated. This means that review of literature on software sales cycle was done and how demos are relevant for it was studied. Some basic theory about reusability and mass customization was studied.

Conceptual model was built based on the most suitable theoretical approaches found in literature. It combined data architecture, technical architecture and integration model to define what should be built and how they should interoperate.

As defined in the introductions, the output of this thesis is a set of software services that are reusable and compatible through the common data model and standard interfaces. Along with the services is the related operating model to describe how the services should be utilized to build demos. This should help fasten the time-to-value and reduce maintenance effort to build meaningful customized demos. So before the actual implementation of the services, implementation plan was designed. It helped to build services in such way that they are easy to integrate and from them customized process model can be built.

Output services were tested in a pilot implementation that gave feedback on validity of the model and set of services created. Operating model was slightly adjusted, but otherwise feedback mostly confirmed that this kind of approach works in suitable cases very well.

This thesis project followed Metropolia project methods that guided through the different steps of building the solution. Metropolia research construction is a variation of Stage-Gate system that is sub-divided into a number of stages or work stations. Between each work station, or stage, there is a quality control checkpoint or gate. The project design methods from ‘Doing Action Research in Your Own Organisation’ (Coghlan & Brannick, 2014) were used to ensure the strong theoretical guidance throughout the process.

7.2 Immediate practical next step recommendations

As immediate practical next steps after the introduction of new services and methodology, it is recommended that larger library of services is built. Those services should then be exposed as well defined API’s so they can be easily integrated and reused. It would also be beneficial to investigate how the target group can be expanded and who could benefit from these capabilities.

It is also advisable to further study how to incorporate other pre-built services, like cognitive services provided by IBM Watson that allows you to include wide array of machine learning capabilities, like how to interpret behavioural patterns, understand unstructured data through sensing and interacting, reason by generating hypotheses, considering arguments and recommendations or learn from training by experts, from interactions and from continually ingesting data. This allows to build demo processes and decisions that can sense, respond, and learn.

Combining cognitive capabilities with your BPM and ODM solutions that are already part of this solution, is often called as Cognitive business operations.

In long term, there is a need to revise how well these services maintain their ability to integrate and work after certain iterations of version updates in the platforms. As each service is contained in separate platform, they are in different update cycle and lifecycle stage. At this point it is not clear how this will effect on the interoperability and maintenance of the services.

7.3 Evaluation of the project

7.3.1 Outcome vs objective

The Objective of this study was to improve methods how Technical Sales teams are building customized demonstrations and how this effort can be reduced by reusable cloud based assets.

When evaluating this project, it should be noted that all services and assets planned and implemented for this project are utilizing cloud as their platform. This allows to use so called best-of-breed approach for selecting tools. In that sense, the results are derived from the speed and flexibility offered by cloud computing. There is no big difference in development of services whether it is developed for on-premise or for cloud platform. But the deployment and maintenance flexibility is the key advantage. From the project implementation and the pilot feedback it can be stated that cloud was the right choice as a platform.

Second major theme in objective for this study was reusability. During the course of this study it was only possible to reuse same services twice. This means that repeatability was tested, but more accurate results would need more iterations. The fair amount of time was used to plan and design the solution so that it would have good foundation on reusability. Within the scope of this study also this objective was met in the outcome.

Interoperability was achieved by utilizing standards based interfaces and connectivity. All design was focusing on the fact that every service (static model) can be bind together in the process layer (dynamic model). Since in the two iterations this was easily attained, it's evident that the project succeeded in this objective.

When we compare the objectives to the outcomes, we can see that the outcome was approximately what was stated in chapter 1.2 Business challenge and objective.

Goals for this study was to build a limited set of compatible services that help:

- Limit time spent preparing demonstrations
- Maximize effectiveness of demonstration delivery
- Increase and broaden use of demonstration assets
- Deliver effective, relevant and current demonstrations
- Lower the barrier of entry for delivering demonstrations

From the successful outcome of the project and the positive feedback from the pilot implementation it's fair to state with reasonable confidence that output of this study will help to achieve these goals if used properly, if it will obtain adequate coverage and services are updated sufficiently.

There are some aspects that are affecting on results of future implementations. First is that each customized demo scenario is different, so with current knowledge it's precarious to reflect outcomes of this study to majority of the future implementations. But this study has shown that for scenarios that are close to ones implemented here are good candidates to benefit from the outcome of this study. Secondly, agility and speed of development depends heavily on experience and know-how of the person in question. This makes it really hard to get reliable quantified results for speed and agility, especially when typically each project is time boxed with different limits.

How maintenance of the objects will affect results in the future is obscure. With limited time for this study it was not possible to observe it there will be any kind of erosion of the service that will reduce the benefits gained from the pilot implementation.

This was the analysis of the outcomes of this study. Next, let’s consider how reliable the results are from the research perspective.

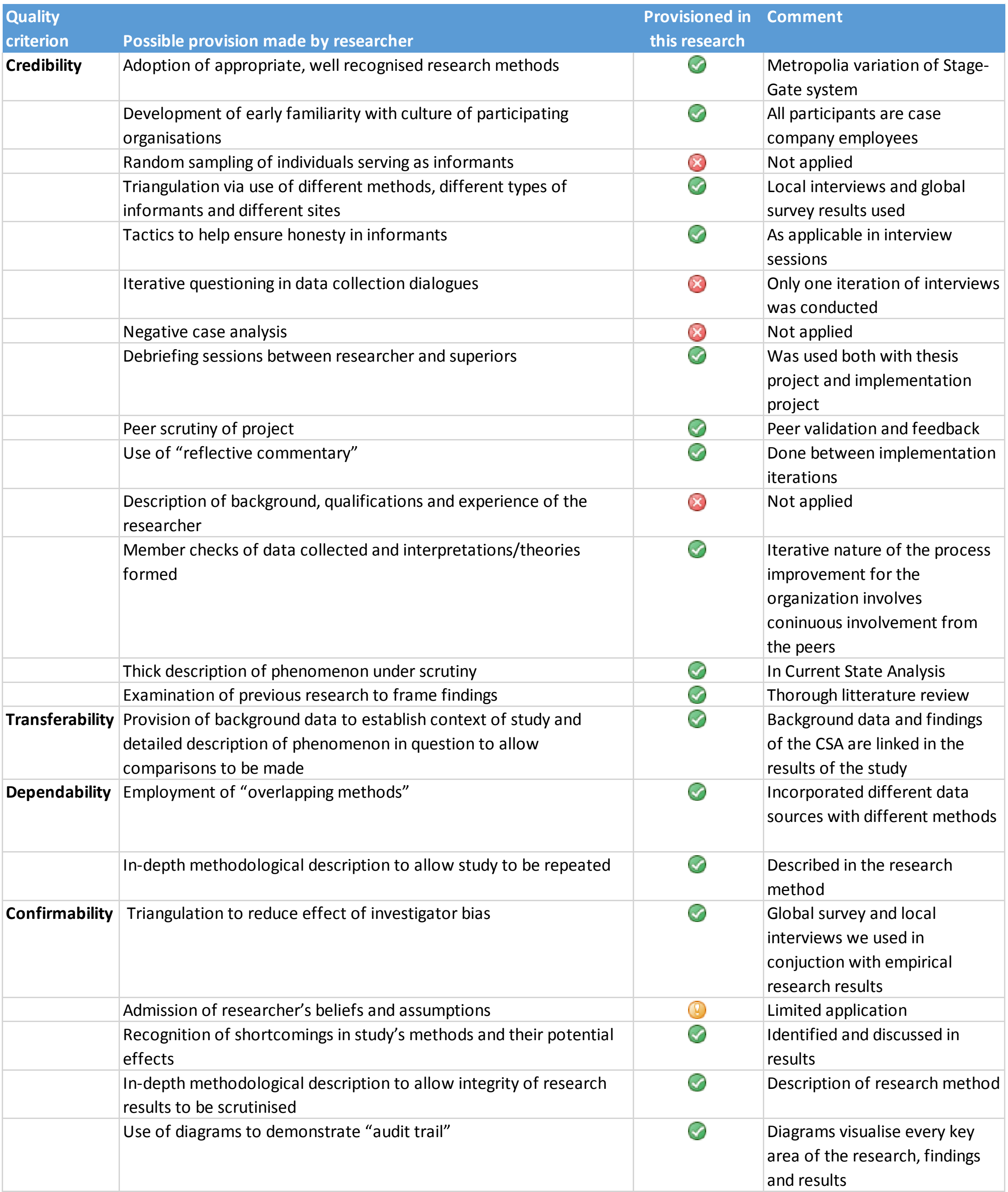
7.3.2 Reliability and validity considerations

Terms like validity and trustworthiness are used to describe quality criteria for quantitative and qualitative academic research (Anderson, et al., 2007). According to Wilson (2013) reliability issues are most of the time closely associated with subjectivity and once a researcher adopts a subjective approach towards the study, then the level of reliability of the work is going to be compromised. Since this study is based on the action research model, it’s possible to test validity of practitioner action research using outcome validity. It’s a term introduced in a book “Studying Your Own School: An Educator's Guide to Practitioner Action Research” (Anderson, et al., 2007). It is assumed that problem solving takes place in the context of the site and is "solved" or "understood" within those parameters, possibilities and limitations. They state that outcome validity is synonymous with the "successful" outcome of the research project, but also note that “if the purpose of action research is to produce knowledge for dissemination in fairly traditional channels (e.g. dissertations, journals) then the criteria for a ‘valid’ or ‘trustworthy’ study may be different from the criteria of practitioners who organize their research around specific problems within an action context and recycle the knowledge back into that context”. This means that action research is less dependent on research method for its trustworthiness criteria.

Lincoln and Guba (1985) posit that trustworthiness of a research study is important to evaluating its worth. Trustworthiness involves establishing:

- **Credibility** - confidence in the 'truth' of the findings
- **Transferability** - showing that the findings have applicablity in other contexts
- **Dependability** - showing that the findings are consistent and could be repeated
- **Confirmability** - a degree of neutraility or the extent to which the findings of a study are shaped by the respondents and not researcher bias, motivation, or interest.

Journal article "Strategies for ensuring trustworthiness in qualitative research projects" by Andrew K. Shenton (2004) considers the criteria in detail and suggests provisions that the qualitative researcher may employ to meet them. Following diagram extends Shentons table “*Provisions that may be Made by a Qualitative Researcher Wishing to Address Guba’s Four Criteria for Trustworthiness*” by mapping provisions to this thesis.



Quality criterion	Possible provision made by researcher	Provisioned in this research	Comment
Credibility	Adoption of appropriate, well recognised research methods	✓	Metropolia variation of Stage-Gate system
	Development of early familiarity with culture of participating organisations	✓	All participants are case company employees
	Random sampling of individuals serving as informants	✗	Not applied
	Triangulation via use of different methods, different types of informants and different sites	✓	Local interviews and global survey results used
	Tactics to help ensure honesty in informants	✓	As applicable in interview sessions
	Iterative questioning in data collection dialogues	✗	Only one iteration of interviews was conducted
	Negative case analysis	✗	Not applied
	Debriefing sessions between researcher and superiors	✓	Was used both with thesis project and implementation project
	Peer scrutiny of project	✓	Peer validation and feedback
	Use of “reflective commentary”	✓	Done between implementation iterations
	Description of background, qualifications and experience of the researcher	✗	Not applied
	Member checks of data collected and interpretations/theories formed	✓	Iterative nature of the process improvement for the organization involves coninuous involvement from the peers
	Thick description of phenomenon under scrutiny	✓	In Current State Analysis
	Examination of previous research to frame findings	✓	Thorough litterature review
Transferability	Provision of background data to establish context of study and detailed description of phenomenon in question to allow comparisons to be made	✓	Background data and findings of the CSA are linked in the results of the study
Dependability	Employment of “overlapping methods”	✓	Incorporated different data sources with different methods
	In-depth methodological description to allow study to be repeated	✓	Described in the research method
Confirmability	Triangulation to reduce effect of investigator bias	✓	Global survey and local interviews we used in conjuction with empirical research results
	Admission of researcher’s beliefs and assumptions	!	Limited application
	Recognition of shortcomings in study’s methods and their potential effects	✓	Identified and discussed in results
	In-depth methodological description to allow integrity of research results to be scrutinised	✓	Description of research method
	Use of diagrams to demonstrate “audit trail”	✓	Diagrams visualise every key area of the research, findings and results

Figure 32 Provisions to Address Guba’s Four Criteria for Trustworthiness applied in this thesis

From this table we can see that the majority of the provisions are addressed during the execution of the thesis project. Thus, it’s fair to state that study has reached an appropriate level of trustworthiness. Also, we can assume that the successful outcome of this thesis is a successful validity test for the research itself. If we break down validity for different aspects other that outcome validity, it’s possible to validate that data collection was democratically valid, since it involved all subject matter experts that are working in this context in Finland, and also collected data from global survey. Process itself can be considered valid, since it is following Metropolia Thesis process that is well defined and closely monitored. Results from this thesis are moved to take actions, so also catalytic validity can be verified. Dialogic validity requires a peer review process, and since it is built-in into action research, because action research involves the researcher in working with members of an organization over a matter which is of genuine concern to them and in which there is an intent by the organization members to take action based on that intervention (Eden & Huxham, 1996).

References

Details of the reference Details of the reference Details of the reference Details of the reference Details of the reference.

Anderson, G. L., Herr, K. & Nihlen, A. S., 2007. Studying Your Own School: An Educator's Guide to Practitioner Action Research. 2nd ed. s.l.:Corwin Press.

Anderson, J., Narus, J. & van Rossum, W., 2006. Customer Value Propositions in Business Markets. Harvard Business Review, 84(3), pp. 90-99.

BCS The Chartered Institute for IT, 2012. Cloud computing - Moving IT out of the office. 1st ed. Swindon, UK: British Informatics Society Limited.

Bittner, K. et al., 2015. Application Modernization, Service By Microservice, s.l.: Forrester Research, Inc.

Business Rules Group, 2001. Defining Business Rules ~ What Are They Really?. [Online]
Available at: http://www.businessrulesgroup.org/first_paper/br01c1.htm#s1e
[Accessed 22 9 2016].

C. Ghezzi, ,. M. J. &. D. M., 1991. Fundamentals of Software Engineering. 1st ed. s.l.:Fundamentals of Software Engineering.

Cakanyildirim, M. &. T. D., 2002. Using Demonstration to Promote Information Products, s.l.: University of Texas at Dallas.

Cardoso, J. & Aalst, W. v. d., 2009. Handbook of Research on Business Process Modeling. 1st ed. s.l.:IGI Global.

Coghlan, D., 2007. Insider action research: opportunities and challenges. Management Research News, Vol. 30(no. 5), pp. 335-343.

Coghlan, D. & Brannick, T., 2014. Doing Action Research in Your Own Organization. 4th ed. s.l.:SAGE Publications Ltd.

Cooper, R., 1990. Stage-Gate System: A New Tool for Managing New Products. Business Horizons, Issue May-June, pp. 44-54.

David Coghlan, T. B., 2014. Doing Action Research in Your Own Organization. 4th ed. s.l.:SAGE.

Dyer, L. et al., 2012. Scaling BPM Adoption From Project to Program with IBM Business Process Manager. 2nd ed. s.l.:IBM Redbooks.

Eades, K. M., 2003. The New Solution Selling. 2nd ed. s.l.:McGraw-Hill.

Eden, C. & Huxham, C., 1996. Action research for the study of organizations. In: Handbook of organization studies. Beverly Hills: Sage, pp. 526-542.

Efti, S., 2015. The Ultimate Startup Guide to Outbound Sales. 1st ed. s.l.:Close.io.

Erl, T., 2007. SOA Principles of Service Design. 1st ed. s.l.:Prentice Hall.

Evans, E., 2003. Domain-Driven Design: Tackling Complexity in the Heart of Software. 3rd ed. s.l.:Addison-Wesley Longman, Inc..

Hinterhuber, A. &. L. S., 2012. Is It Time to Rethink Your Pricing Strategy?. MIT Sloan Management Review, 53(4), pp. 69-77.

Hoover, K., 2016. Announcing IBM API Connect: A complete API lifecycle offering. [Online]
Available at: https://developer.ibm.com/apiconnect/2016/02/18/announcing-ibm-api-connect-a-complete-api-lifecycle-offering/
[Accessed 14 6 2016].

IBM Redbooks, 2014. Integration Throughout and Beyond the Enterprise. 1st ed. Armonk, New York: Vervante.

IBM, 2015. Blueworks Live Overview. [Online]
Available at: http://www-01.ibm.com/support/knowledgecenter/SS2MKC/com.ibm.bwl.doc/topics/bwl_overview_old.html
[Accessed 17 01 2016].

IBM, 2015. Business processes in IBM BPM on Cloud. [Online]
Available at: http://www-01.ibm.com/support/knowledgecenter/SS964W/com.ibm.wbpm.cloud.man.doc/topics/cbpm_manclo_overview.html
[Accessed 17 01 2016].

IBM, 2015. Global survey of Systems Middleware Technical Sellers, Armonk, New York: IBM.

IBM, 2015. IBM Business Process Manager overview. [Online]

Available at: http://www-01.ibm.com/support/knowledge-center/SSFPJS_8.5.6/com.ibm.wbpm.main.doc/topics/ibmbmp_overview.html
[Accessed 17 01 2016].

IBM, 2016. IBM Integration Bus on Cloud. [Online]
Available at: http://www-03.ibm.com/software/products/fi/ibm-integration-bus-on-cloud
[Accessed 14 6 2016].

IBM, 2016. IBM Operational Decision Manager product page. [Online]
Available at: http://www-03.ibm.com/software/products/en/odm/
[Accessed 17 01 2016].

Janiesch, C., Matzner, M. & Müller, O., 2012. Beyond process monitoring: a proof-of-concept of event-driven business activity management. Business Process Management Journal, 18(4), pp. 625-643.

John Care, A. B., 2008. Mastering Technical Sales: The Sales Engineer's Handbook. 2nd ed. s.l.:Artech House Technology Management Library.

Josuttis, N. M., 2007. SOA in Practice. 1 st. ed. s.l.:O'Reilly Media, Inc..

Jun Shena, G. G. Y. Y. ,. M. S. ,. M. S. T. R., 2007. Analysis of business process integration in Web service context. Future Generation Computer Systems, March, 23(2), p. 283–294.

Kang, B., Jae-Yoon, J., Cho, N. W. & Suk-Ho, K., 2011. Real-time business process monitoring using formal concept analysis. Industrial Management & Data Systems, 111(5), pp. 652-674.

Larkin, I., 2014. The Cost of High-Powered Incentives: Employee Gaming in Enterprise Software Sales. Journal of Labor Economics, 32(2).

Leymann, F., 2002 . Web services and business process management. IBM Systems Journal , 41(2), pp. 198 - 211 .

Lincoln, YS. & Guba, EG. (1985). Naturalistic Inquiry. Newbury Park, CA: Sage Publications.

Linthicum, D. S., 2009. Cloud Computing and SOA Convergence in Your Enterprise: A Step-by-Step Guide. s.l.:Addison-Wesley Professional.

Maclennan, E. & Van Belle, J.-P., 2014. Factors affecting the organizational adoption. Information Systems and eBusiness Management, 12(1), pp. 71-100.

Mell, P. & Grance, T., 2011. The NIST Definition of Cloud Computing, Gaithersburg: Computer Security Division, Information Technology Laboratory,National Institute of Standards and Technology.

Morgan, T., 2002. Business Rules and Information Systems: Aligning IT with Business Goals. 1st ed. Boston: Addison-Wesley Professional.

Natis, Y. V. & Altman, R., 2014. Software-Defined Architecture for Applications in Digital Business, s.l.: Gartner.

Nidumolu, S. R. & Knotts, G. W., 1998. The effects of customizability and reusability on perceived process and competitive performance of software firms. MIS Quarterly, 22(2), pp. 105-137.

Open Applications Group Corporation, 2016. Open Applications Group. [Online]
Available at: http://www.oagi.org/dnn2/
[Accessed 3 3 2016].

Pant, K. & Juric, M., 2008. Business Process Driven SOA using BPMN and BPEL. 1st ed. Birmingham: Packt Publishing.

Patton, M. Q., 1990. Qualitative evaluation and research methods. 2nd ed. s.l.:SAGE Publications.

Pine, B. J. I., 1993. Making mass customization happen: Strategies for the new competitive realities. Planning Review, 21(5), p. 23.

Redguides for Business Leaders, 2014. The Power of the API Economy, Armonk, New York: IBM.

Reynolds, J. et al., 2014. Leveraging the IBM BPM Coach Framework in Your Organization. 1st ed. s.l.:IBM Redbooks.

Rogers, G. & Bottaci, L., 1997. Modular production systems: a new manufacturing paradigm. Journal of Intelligent Manufacturing, vol. 8, no. 2, pp. 147-156. , 8(2), pp. 147-156.

Rosing, M. v., Scheer, A.-W. & Scheel, H. v., 2014. The Complete Business Process Handbook. s.l.:Morgan Kaufmann.

Scheer, A.-W., 1999. ARIS - Business Process Frameworks. 1st ed. s.l.:Springer Science & Business Media.

Schneider, G. K., 2010. SAP Modeling Handbook - Modeling Standards. s.l.:SAP.

Shenton, A. K., 2004: Strategies for ensuring trustworthiness in qualitative research projects. Education for Information, Vol. 22, pp. 63-75.

Shetty, D. G. et al., 2014. API concepts, trends, and overview. In: Exposing and Managing Enterprise Services with IBM API Management. s.l.:IBM Redbooks.

Smith, E., 2015. Cloud Computing: The Essential Foundation of Industry Digital Transformation — Worldwide and U.S. Cloud Forecast by Vertical, 2015–2019, Framingham: International Data Corporation.

Tallon, P. K. K. &. G., 2000. Executives' perceptions of the business value of information technology: A process-oriented approach. Journal of Management Information Systems, 16(4), pp. 145-173.

techtarget.com, 2005. What is vanilla?. [Online]
Available at: http://whatis.techtarget.com/definition/vanilla
[Accessed 29 2 2016].

The Open Group, 2011. SOA Reference Architecture Technical Standard. C119 ed. Berkshire, United Kingdom: The Open Group.
Tilkov, S., 2015. Blog: Why You Should Avoid a Canonical Data Model. s.l.:innoQ.

Wei, Z., Dejun, J., Pierre, G. & Chi-Hung Chi, M. v. S., 2008. Service-Oriented Data Denormalization. Beijing, China, Proceedings of the 17th International World Wide Web Conference.

Wilson, J., 2013. Essentials of Business Research: A Guide to Doing Your Research Project. 2nd ed. s.l.:SAGE Publications Ltd.

Woolf, G. & Hohpe, B., 2004. Enterprise Integration Patterns. 1st ed. s.l.:Addison-Wesley.

Zheltonogov, V. et al., 2015. Approaches and process discovery. In: Business Process Management Design Guide: Using IBM Business Process Manager. Armonk, New York: IBM Redbooks.