Decision support model for choosing a cloud solution

Utrecht, August 31st 2015

Author
4094786
Anton Sougalsky
Utrecht University
Master Business Informatics
A.Sougalsky@students.uu.nl

1st Supervisor
dr.ir. Jan Martijn van der Werf (UU)

2nd Supervisor
dr. Fabiano Dalpiaz *(UU)*

Universiteit Utrecht

Abstract

The thesis deals with a process of cloud service selection aiming at presenting a process of rigor systematic selection of possible options starting from a business level perspective and ending with specific implementation tactics. We believe that if choosing a cloud service only gathering requirements such as Quality of Service, in many cases severe architectural problems arise. This is caused by not paying due attention to high-level properties of business processes. Many companies adopt cloud solutions nowadays, and the reasoning behind those decisions is real but not yet formalized in a good level of details. The aim of this work is to demystify this reasoning and structure all the knowledge from academia together with business practices in a single guideline which supports cloud selection process. Current research presents a systematic process of choosing best-fit solution using which less requirements may be missed. The Thesis presents five steps following each other needed to take to formulate all requirements needed for a Request for Proposal to a cloud vendor. Emerging from a rigor literature review, presented model is the best of breed from existing business, Enterprise Architecture and Cloud technologies requirements taxonomies, includes up to date technologies and while being formal keeps place for upcoming technologies and solutions. In the second part of the thesis it is shown how to build a Decision Support System based on the theoretical model proposed, and the possible usefulness of this tool is validated in a series of expert interviews.

Preface and Acknowledgment

The sciences do not try to explain, they hardly even try to interpret, they mainly make models.

— Johann von Neumann, 1903–1957 —

This is a research master's thesis. It does not deal with very practical matters e.g. evaluation study of a new medical app of the company where a student does his internship. Quite in opposite, it deals with a lot of scientific and business knowledge that was put together to broaden my horizon on the subject of cloud computing, which it very topical nowadays. Thus it can help a lot people who are starting to discover the field, and also help professionals in the field due to the fact that this research is rather broad and tries to be unbiased. Simply speaking, while educating myself as a research master, I hope this work will in a way help others interested in the subject.

I would like to use this opportunity to thank the people assisting me during the research. First of all, I would like to thank my first supervisor dr. ir. Jan Martijn van der Werf for accepting the topic that I see relevant for me, providing advice and directing this thesis. I would like to thank my second supervisor dr. Fabiano Dalpiaz for quick responses, strong wish to help students like myself and his encouragement.

Outside of Utrecht University, I would like to thank both of my external advisors dr. Yury Kupriyanov (Higher School of Economics) and Radhika Garg (University of Zurich) for an outsider view on the topic and thus for their invaluable advises. Also, I would like to thank them for their help in validating the results of my scientific study. In addition, I would like to thank questioned field professionals for their time spent, helpful responses and fresh ideas.

Finally, I would like to thank my family and friends for making all possible for me having time and energy to work on my thesis.

Table of Contents

Glossary

Common Used abbreviations

BI .... Business Intelligence

CRM .... Customer Relationship Management

DSM .... Decision Support Model

DSS .... Decision Support System

ERP .... Enterprise Resource Planning

HR .... Human Relations

IoT .... Internet of Things

ISO .... The International Standards Organisation

M2M .... Machine to Machine

NIST .... National Institute of Standards and Technology

SLR .... Systematic Literature Review

Cloud Computing-specific abbreviations

BDaaS .... Big Data as a Service

BPaaS .... Business Process as a Service

DBaaS .... Database as a Service

IaaS .... Infrastructure as a Service

iPaaS .... Integration PaaS

PaaS .... Platform as a Service

SaaS .... Software as a Service

VM .... Virtual Machine

VPN .... Virtual Private Network

XaaS .... Anything as a Service

EA and Business Process Modelling abbreviations

BPM .... Business Process Management

EA .... Enterprise Architecture

QA(s) .... Quality Attribute(s)

QoS ..... Quality of Service

ROI ..... Return on Investment

RTO ..... Recovery Time Objective

SOA ..... Service Oriented Architecture

TCO ..... Total Cost of Ownership

TOGAF ..... The Open Group Architecture Framework

Vendors and Products

AWS ..... Amazon Web Services

EC2 ..... Elastic Computing Cloud {Amazon product}

S3 ..... Simple Storage Service {Amazon product}

MS ..... Microsoft Corporation

Terms used in the paper

Apart from abbreviations mentioned earlier, it is vital to align the reader with the author in specific terms used further on.

To start with, the word "cloud" comes from the fact that many years ago those who built and sold client server applications, software and hardware used to draw a picture with the PC connected to a network and the network connected to a server. Since most people didn't actually understand how the network worked, they drew a cloud and labelled it "network" and left it at that.

Architecture – fundamental concepts or properties of a system in its environment, embodied in its elements, relationships, and in the principles of its design and evolution [1].

Architect – one whose responsibility is the design of an architecture and the creation of an architectural description [2].

Enterprise - a business organization[1].

Enterprise Architecture (EA) – an architecture in which the system in question is the whole enterprise, especially the business processes, technologies, and information systems of the enterprise [2].

Business Function - this is a description of all business elements and structures that are covered by the enterprise [3].

Business Process - a collection of related, structured activities or tasks to achieve corresponding business goals [3]; in this paper, it basically is a part of Business Function.

Business Architecture - an architectural formulation of the Business Function [3].

[1] Merriam-Webster dictionary, available at: http://www.merriam-webster.com/dictionary/enterprise

Service/Application/Solution Function - This is the function that aims at delivering/supplying computerized IT systems required to support the plethora of specific functions needed by the Business Function [3].

Service/Application/Solution Architecture - an architectural definition of the Service/Application/ Solution Function [3].

Technology Infrastructure Function - the complete technology environment required to support the Service/Application/Solution Function [3].

Technology Infrastructure Architecture - an architectural formulation (description) of the Technology Infrastructure Function [3].

View - a representation of a whole system from the perspective of a related set of concerns [2].

By **Quality Attributes (QA)** we mean the properties and requirements of a business function that should be met by EA.

By **Quality of Service (QoS)** which by definition are the performance properties of a network service[4], we mean the EA requirements posed by business to a service.

1. Introduction

Back in 1999 Salesforce.com pioneered the concept of delivering enterprise applications via a simple website. According to Google Trends[2], the term "cloud solution" started becoming popular in the end of 2008 as shown in Figure 1, today yielding 1,810,000 results in Google Scholar.

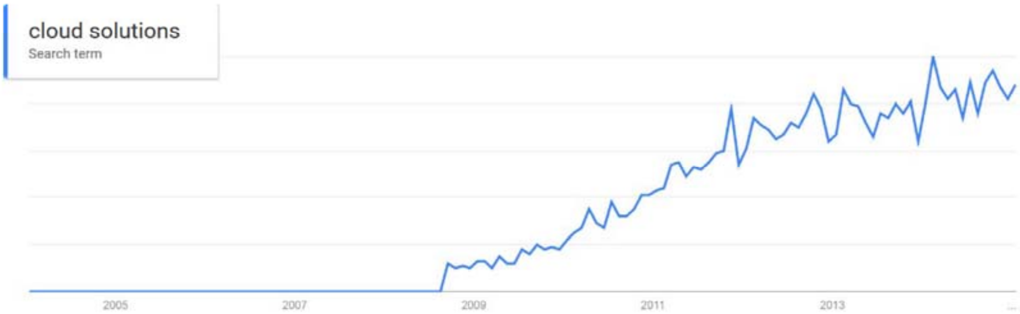


Figure 1 Searches for "cloud solutions" on Google.com, taken from Google Trends

Nowadays, a solid body of research is present in the cloud computing domain. Many papers are focused on showing state-of-the-art [5]–[11], some authors present reviews of relevant literature [12], benchmarking possibilities [13], different architectures [6] and technologies [14] used by cloud solutions.

In 2012, the European Commission adopted a European Cloud Computing Strategy, one of which tasks is to 'cut through the jungle of standards' that have proliferated for cloud computing services.

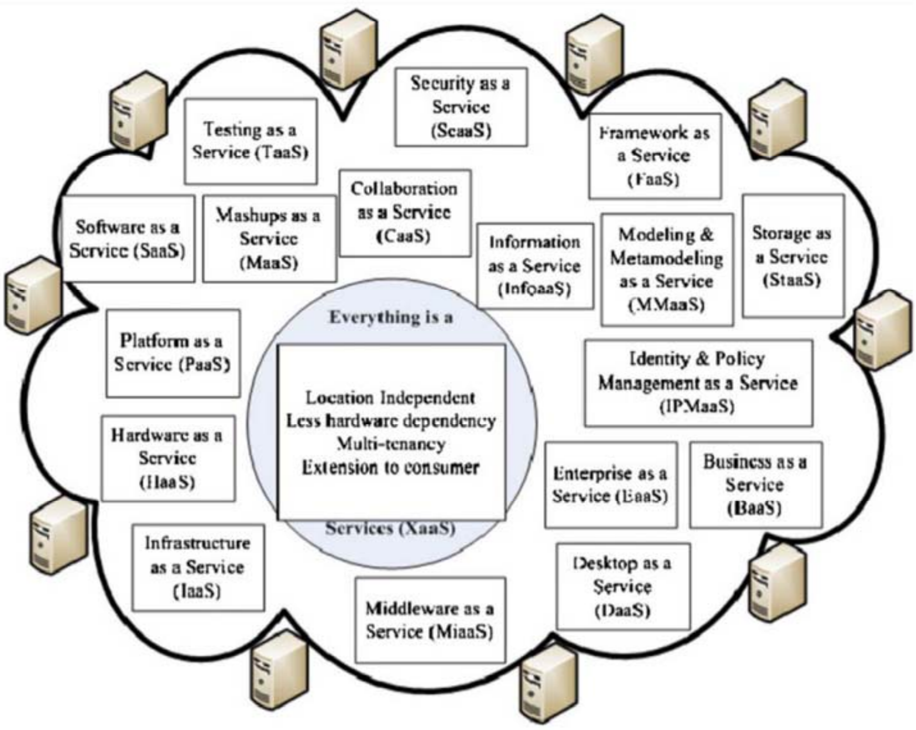


Figure 2 "Jungle of standards" retrieved from [15]

The Pew Research Center's Internet and American Life Project and Elon University recently conducted a survey of 900 Internet practitioners, social analysts, and researchers; their survey results confirm this viewpoint. Specifically, most of the survey respondents believe that Internet users will "live mostly in the

[2] A public web-search analysis facility of Google Inc., based on Google Search: http://www.google.com/trends/

cloud" by 2020 [16]. Thus, it is extremely important both for business and academia to "cut through these jungles" right now.

1.1. Problem definition

It is common to believe that cloud computing reduces costs and increases business agility, but in fact any cloud adoption brings too much questions due to the low maturity of the knowledge in the field. Most of analysts agree that if one thinks a company can move a service to the cloud without re-architecting, he is probably not doing it right. Enterprise-wide cloud architecture is of great importance these days, when cloud technologies are promising to deliver almost all the benefits imagined. Interest and demand for cloud-based solutions has never been higher, and the marketplace has responded with an abundance of solutions that offer every possible permutation of cloud computing architecture: public, private, hybrid, multi-tenant, single-tenant, PaaS, iPaaS, SaaS, BPaaS… the possibilities are seemingly endless. Of course, there are different types of cloud computing taxonomies that exist in the literature: [17]–[46]. Some are wide concepts, some focus on specific aspects such as security, performance, etc., while others present an overview of vendor solutions present in the market. Some authors focus solely on reviewing relevant research done by others [18],[41]–[46], which helps in finding the information needed. Nevertheless, with all of these solutions, how can one possibly determine which cloud architecture approach is right for a specific business case?

Despite dozens of sources touching specific points of cloud selection process, there is still no framework which deals with this process from the very beginning until the very end. Still, due to the fact that many companies adopt cloud solutions, the reasoning behind those decisions is real. We just have to formalize this process. The aim of this work is to demystify this reasoning and structure all the knowledge from academia together with business practices in a single guideline which supports cloud selection process.

Knowing that business application architecture is tightly connected to (best scenario – derived from) Enterprise Architecture (EA), and that EA is in turn tightly connected to business goals – it is logical to start from the top (business architecture) when deciding on the type of application architecture. For instance, The Open Group Architecture Framework[3] (TOGAF) divides an Enterprise Architecture into four categories, as follows:

- Business architecture - describes the processes the business uses to meet its goals;
- Application architecture - describes how specific applications are designed and how they interact with each other;
- Data architecture - describes how the enterprise datastores are organized and accessed;
- Technical architecture - describes the hardware and software infrastructure that supports applications and their interactions.

TOGAF's Architecture Development Method (known as ADM) consists of phases which follow one another, e.g. application architecture follows business architecture and etc. corresponding to the level of granularity.

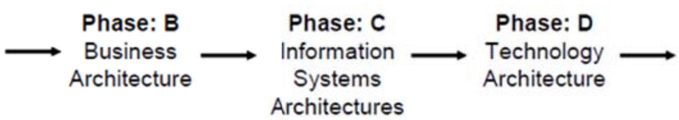


Figure 3 A part of The TOGAF ADM (derived from [47])

Similarly, we propose to derive specific technical requirements step by step from the very beginning of business needs. Figure 4 below illustrates this approach by [3], where TOGAF's four categories are following the business requirements.

---

[3] An enterprise architecture methodology and framework, available at: http://www.opengroup.org/togaf/

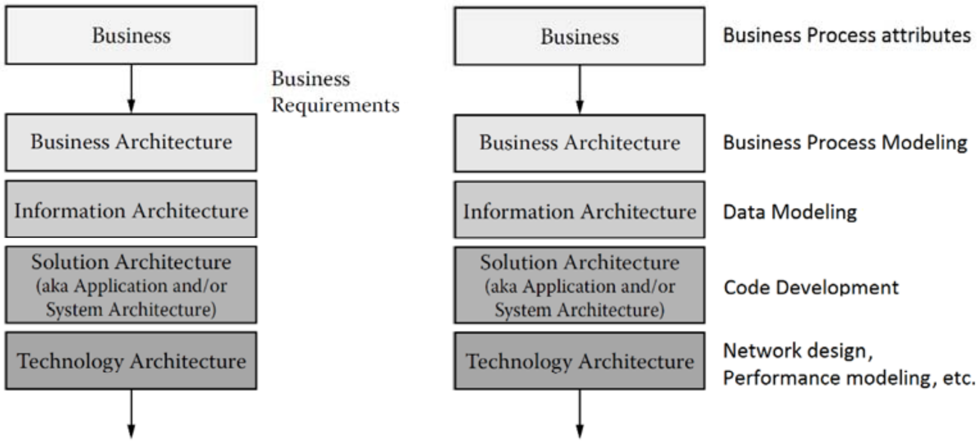


Figure 4 Left: a part of Enterprise Architecture approach by [3], *Right: artefacts examples*

Compliment to the functional requirements and following architectural decisions, there are always non-functional (quality) requirements in place – we call them Quality of Service requirements. Whether it is business architecture view or technology architecture view, quality has an impact on all aspect areas. Thus, an Architect needs to define the quality of all types of service: the quality of business service, the quality of business information service, quality of information system service and the quality of technology infrastructure service [48].

The closest approach to the proposed DMS is formulated by [49]:

"We formulated a three-tier decision model for assessing the types of decisions required to move services to the cloud either wholly or partially or, perhaps, not at all. The model follows our literature analysis and research findings and addresses the requirements for services, commercial offerings (vendor cloud packaging), and architectural decisions (short and long term). In descending order, an organization must decide the importance of numerous factors (not all shown) in how it approaches the cloud adoption decision."

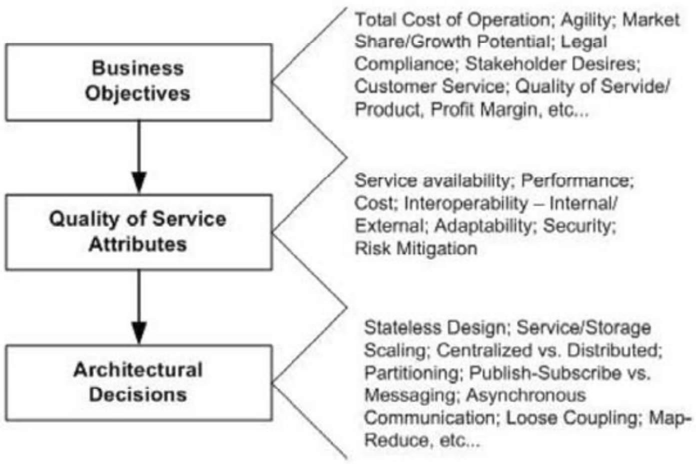


Figure 5 Decision model proposed by [49]

However, this model does not provide the argumentation behind the transition between steps. It only shows the layers of the approach which we believe is accurate. Following this approach (going from high level business objectives to low level service architectural decisions), current work aims to actually develop a decision model with all transitions between steps so no requirements are lost in ideal case. Moreover, it aims to include up to date technologies and cloud deployment scenarios available. All the reasoning behind the model should be documented, so the model could be easily extended for future uses in different business cases.

1.2. Relevance

1.2.1. Scientific Relevance

1. According to the literature review, there is no linkage between Quality of Service attributes and QAs of business functions. Since it is always better to have a reasoning behind service selection decisions, this research aims to provide such for a business so an Architect can derive QoS based on fundamental needs of particular business infrastructure. Proposed DSM shows how business objectives are transformed to EA requirements and then to QoS requirements.

2. According to the literature review, there are dozens of different taxonomical approaches for grouping cloud solutions. This research proposes a taxonomy based on other ones before it, and it is more abstract (concerned with application types, not vendors) and more complete.

What is even more important, there is a place for new technologies in the taxonomy. Vendor solutions are not yet available, but they will emerge soon. That makes this taxonomy last for more years unlike existing ones.

3. According to the literature review, there is no linkage of cloud services deployment models to quality of service attributes. With this research, it will be possible to close those gaps and see how cloud-based Enterprise Architecture interlinks with regular on-premise Enterprise Architecture.

Basically, the current work fills the gap between tacit knowledge of professionals working with cloud solutions and explicit formal method that is documented. Consider the following illustration:

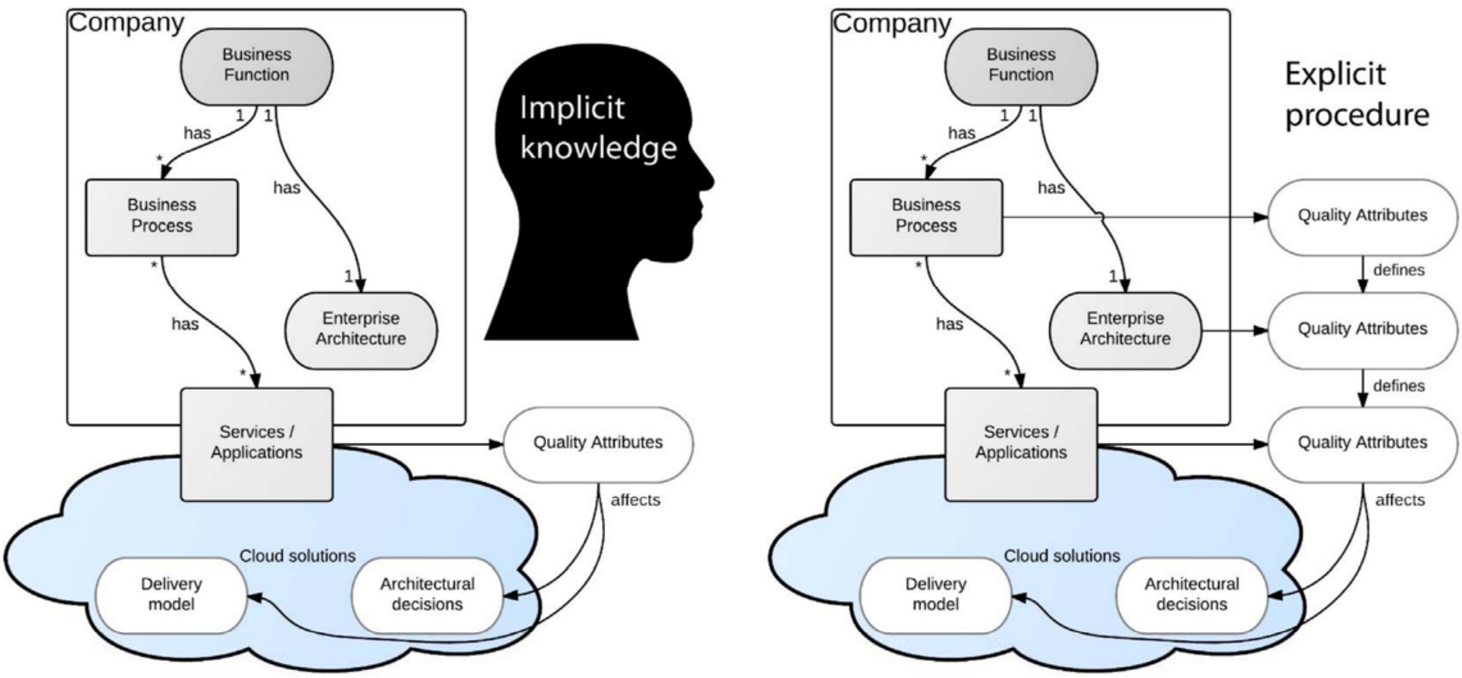


Figure 6 Missing gap in reasoning of cloud adoption

Formalizing the process of gathering necessary requirements for cloud adoption ensures sustainable results and higher level of organization maturity.

1.2.2. Social Relevance

Social (in this case, business) value of this research is the Decision Support System based on proposed concept. There is no such algorithm yet neither a set of described reasoning, and companies rely on gut feeling when choosing the type of cloud solutions they think they should use instead of legacy apps (or new services). After that, they compare solutions within this type among vendors, specs and prices. In some cases - it is ok. But sometimes, they might have chosen better, if having such a solid approach, where a lot of different configurations are taken into account while choosing a service that satisfies necessary QAs.

This research might be used as a tool helping a consultant or an Enterprise Architect to see what kind of cloud fits company's needs and given formulated NFR for it, what deployment models can be used. Given this knowledge, consultant/architect can choose between vendors found knowing the type of cloud, and formulate cloud architecture in RFP.

Apart from bringing structure in business decisions, the model can be used for educational purposes. The model offers a broad perspective on cloud technologies and their adoption process, contains most up to date information about them and is presented as an interactive interface.

1.3. Study outline

Decision Support Model which is the topic of the current work is a model of a Decision Support System (DSS). After the theoretical concept is made - DSM, it is possible to build a tool based on it – DSS. First part of the thesis focuses on a systematic process of choosing a cloud which will be used as a base for DSS. Second part of the thesis presents a DSM syntax, DSS prototype and validation of the theory behind the future DSS.

Three fundamental components of a DSS architecture are[4]:

- the database (or knowledge base),
- the model (i.e., the decision context and user criteria), and
- the user interface.

First two are presented in the thesis as a concept that can be further expanded/elaborated. Therefore, a formal syntax is used to describe the model. The model consists of five steps (five levels or granularity) to address when gathering requirements for a cloud service. Starting from functional requirements (what kind of cloud is it), an Analyst then describes Business Process attributes, then Enterprise Architecture QAs (IT perspective in whole organization), then QoS attributes (for a specific cloud service, and finally the specific architectural decisions (also known as implementation tactics). All these steps forming one DSM are to ensure a thorough requirements gathering procedure, sustainable result, and less alteration as the consequence. Third component, which is a user interface, is presented as a mockup of a DSS tool and partly as a developed prototype of a tool.

Potential usefulness of the tool is validated with expert interviews. Validation procedure is described in Chapter 5. After the successful concept validation phase, a part of DSS tool prototype was developed and a corresponding Chapter 6 presents a mockup of the tool and working examples with its functionality. A running example (a case study with an imaginable company and an Analyst choosing a solution for it) can be seen in Appendix G.

The thesis document is structured in the same way the research process was carried out. To show the steps taken during the research and the order of chapters, we are using a conceptual diagram. A Process-Deliverable Diagram (PDD) is a diagram that describes processes and data that act as output of these processes [50]. Using PDD notation[5], Figure 7 below shows activities performed (left side) during the thesis creation and data artefacts corresponding to them (right side).

[4] https://en.wikipedia.org/wiki/Decision_support_system
[5] PDD notation: https://en.wikipedia.org/wiki/Process-data_diagram

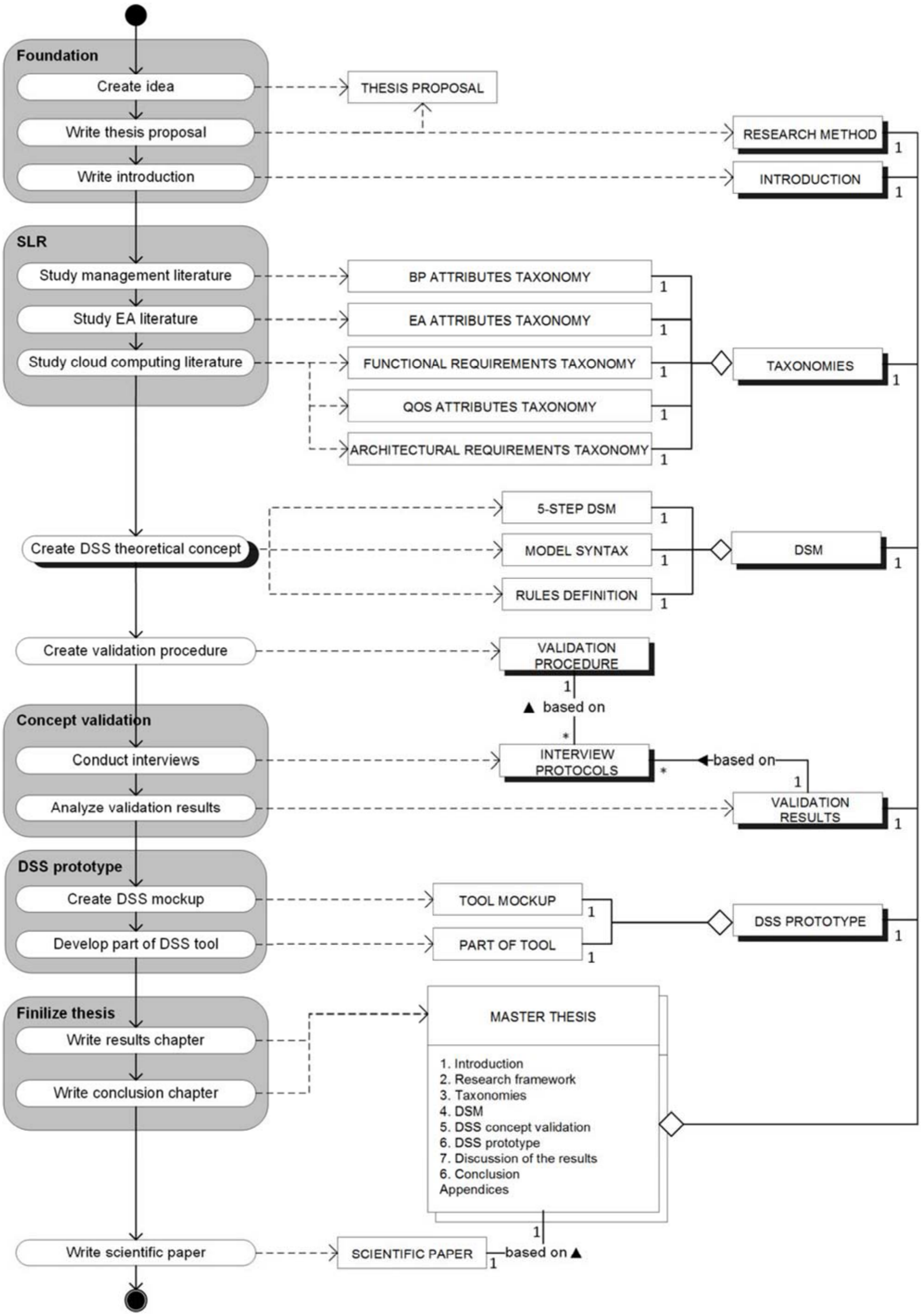


Figure 7 PDD of the current research

2. Research Framework

In this part of the paper one may find the description and the implementation of the specific design methodology model (proposed Decision Support Model) and the corresponding materials, used to study the given case. This section offers a detailed description of what research methods were used and how.

2.1. Research questions

2.1.1. Main research question

As described in previous chapter, there is a substantial amount of studies describing the cloud selection process from different perspectives, different views, and different levels. However, the linking between those levels, a single line of reasoning throughout all the levels which is vital to one's ability to see the full process as whole, is missing.

There is an abundance of business process modelling techniques that capture and address different aspects of a business process [51]. Still, while being present, business side QAs' influence on IT services' QoS attributes is well not documented in the literature. That leads to a more professional's personal experience being used to derive cloud service QoS attributes from Business Process QAs instead of using structured reasoning in help for that. This kind of reasoning is addressed in by this research, as well as a full procedure of selecting a cloud service.

Last but not least, emerging technologies and trends already addressed by the body of literature are still missing from academic papers and dedicated Internet resources dealing with cloud vendors' offers, available deployment techniques etc. Is it possible to include them in a DSM and how to construct such a DSM that is easily extendable?

Considering the scope of this work, it is limited to an enterprise cloud services. That means it only deals with more or less business applications, leaving gaming, health, social media, and other cloud services apart from the scope. Choosing IT solutions for more specific uses e.g. army sector or scientific research require completely different methodology while the needs of organization are completely different. E.g. strategic military objects have a lot more security requirements and this model is too simple on this side, while being too overload with unnecessary attributes on the other – in this case, a very different priorities are set to requirements gathering process. To limit the scope, we focus on business applications. More elaborate description of DSS use cases and scenarios are set in Chapter 6.3 which deals with DSS prototype.

Main Research Question:
What is an effective systematic procedure of selecting a cloud service for an enterprise?

When deciding upon a service migration or a new service adoption, a business case is presented to an Architect. Such business case contains the information concerning service requirements from business level perspective, and they should in turn be translated to EA attributes, then to QoS and consequently to specific architectural decisions of a wished cloud service.

The next step would be in analysing cloud services that fit the formulated requirements/attributes. For this step, a proper overview of cloud services that exist and their properties should be present. There are different views on how to structure types of cloud solutions [18],[41]–[46], their deployment tactics [15], [52], [53], and their QoS attributes [17], [51], [54]–[68].

To answer the main research question of this study, the aim of this work is to bring a unifying structure to all those steps according to the research focus and define a solid Decision Support Model.

It is important to limit the scope of the research. Current work is dedicated to general businesses indifferent of their size but leaving more specific business companies aside. To those specific ones we ascribe ones from health industry, game industry, military or government organizations and those which pose very specific requirements to IT services they use. To give an example, due to unstable political situation in a country a company can only rely on its own developments and cannot choose economically favourable and efficient service. Or an NGO that develops mobile applications for South African countries – it can only build applications on one or two platforms available in that region (in order for them to be reusable), so there is not much choice to be done with elaborate DSM.

2.1.2. Decision Support Model

In order to answer the main research question, the following Decision Support Model was formulated. It consists of 6 different views on cloud service requirements. Starting from high-level business requirements, steps of the DSM descend to specific technologies used to fulfil the requirements. Between the steps of the model, there are information transitions, where the nature of requirements changes i.e. functional requirements affect technical requirements, e.g. if it has to be an iPaaS service, then there should be an API with command line feature. Thus, the following steps depend on preceding ones. This way, a solution definition which is the final result of the process comply with all formulated requirements and truly satisfy business needs.

Based on the model above, architect's task is to assure the QAs of a business function and its processes are satisfied by selected software service with its QoS attributes. That means, that the Architect/Business Analyst should be able to define EA QAs based on BP QAs; then derive QoS from those EA QAs; then based on QoS requirements and functional requirements (type of service needed) select a service or formulate a RFP.

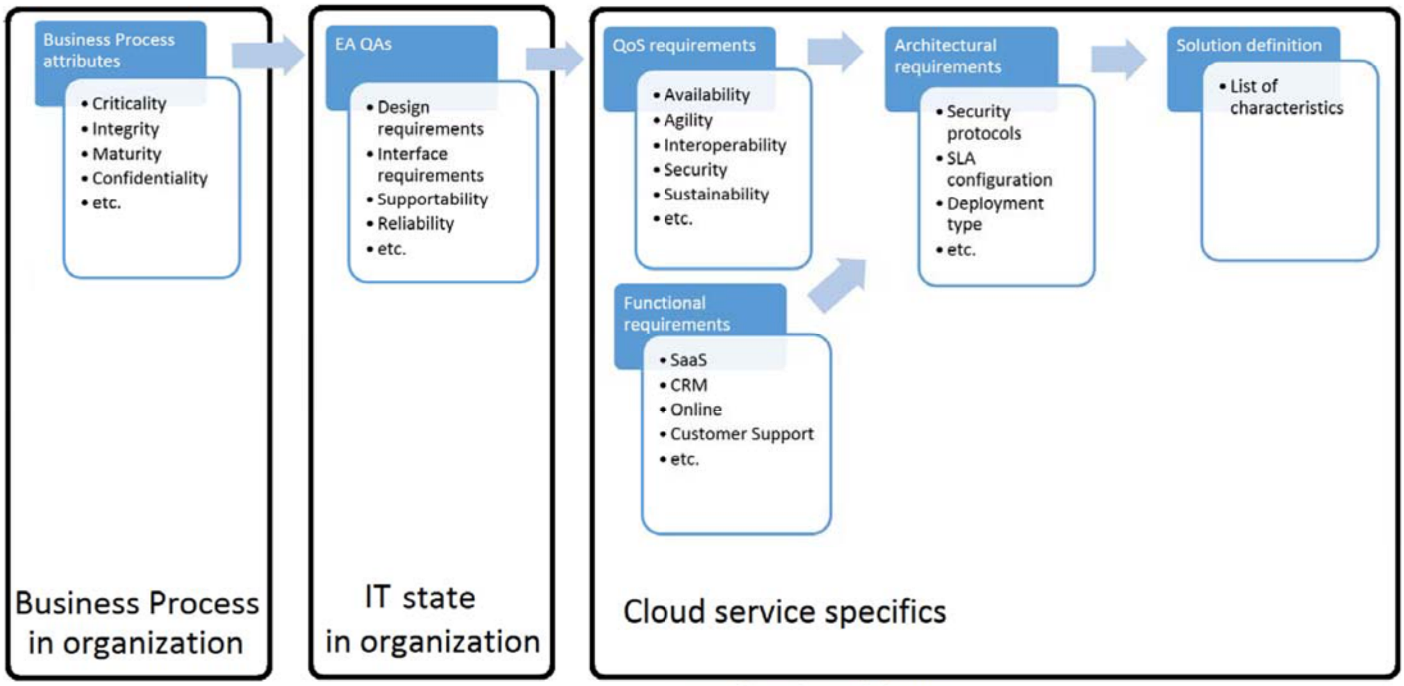


Figure 8 Necessary steps for solution definition

To transfer the requirements from one step to another, attributes of each view should be connected with attributes from another view which are affected by them. Examples of such cross-connections between different views (QoS and Architectural requirements) could be:

- Architectural requirements affected by QoS requirements such as availability or installability (ease of deployment, hardware requirements) [53];
- QoS requirements affected by architectural requirements [15]:

Parameters / Deployment models	Cost	Migration	Client base	Security	Control over the use of cloud resources	Legal issues
Private cloud	Expensive	Standard API is needed between public, private and hybrid cloud	Large enterprises and corporations	High	High	
Public cloud	Less expensive than private cloud	Standard API to make data movement seamless	Large as well as SMEs	– Low, more chances of malicious activities such DDoS attacks – Trusted virtual data center is required	Low	National boundary for data storage
Community cloud	Relatively cheaper than other three cloud models	Standard API is needed to migrate data within specific community	Small SMEs	Low	Low	
Hybrid cloud	Cost savings	Standard API to make data movement seamless	Multiple internal and/or external providers	Application compatibility issues	High	National boundary for data storage

Figure 9 Cloud deployment models and the corresponding architectural requirements [15]

After the solution (DSS tool) to a problem (main RQ) is developed, a look back should be attempted to see whether the problem is solved and how should the solutions be better [69]. However, thesis scope limits the work to developing a DSM as a theoretical concept based on literature and a prototype of a DSS tool based on DSM. Therefore, we will only validate the theoretical concept and extrapolate DSS potential usefulness to DSM success.

2.1.3. Research sub-questions

To build the DSM described above, the main research question should be divided by sub-questions corresponding to the steps of the model proposed.

RsQ1. What are the necessary attributes for each step in the model?

1) **Business Process attributes. Which attributes can be used to assess a business function from business perspective?**

Research task: List most of common QAs, group them in a structured view (from now and further – taxonomy[6]), preferably based on a known standard.

To list some examples of a business function, it can be a core business function (e.g. production) yielding income [70], or a support business function (ancillary activities) that are carried out in order to permit or facilitate the core one (e.g. management functions). Business functions can be divided into mission-critical (vital) (without which a firm cannot operate or remain viable), and not. In turn, criticality falls into several categories: high (most severe), medium, and low (least severe); or: mission-critical, vital, important, minor; etc. depending on the classification. It is relevant for this study, what kind of business functions distinction options are to consider while selecting EA QAs in the next step.

Considering that a business function is a set of business processes, it is logical that business functions properties are strongly influencing business processes attributes. Business Process Management (BPM) is a "hot topic" because it is highly relevant from a practical point of view and motivated by practical challenges and is grounded in both computer science and business administration [71]. BPM's essential concepts are backed up with different process languages (e.g. Petri nets, EPCs, Workflow nets, etc.), while those are characterized and related using metamodels. Main dimensions along which business processes can be classified are investigated for instance by [71]:

- Organizational *vs* operational processes;
- Intraorganizational processes *vs* process choreographies;
- Degree of automation;

[6] Reasoning on why a taxonomical view is chosen to represent gathered information is explained in Chapter 2.2.1.

- Degree of repetition;
- Degree of structuring.

What kind of structure and its properties to use, should be defined while answering this sub-question, along with attributes to use in DSM.

Deliverable: Taxonomy of Business Process attributes.

2) EA QAs. Which QAs exist from the Enterprise Architecture viewpoint?

Research task: Structure EA QAs based on based on a known standard.

When having an existing EA, why move to the cloud? What apps to move, if to? To answer these questions, EA can be viewed from service-oriented perspective - quality attribute requirements drive software architecture design.

Deliverable: Taxonomy of EQ QAs.

3) QoS requirements. Which Quality of Service attributes are used for cloud solutions?

Research task: List most of common QoS attributes for any type of cloud service. Structure them based on based on a known standard.

If there are cloud services that can perform existing on-premise services functions, then it might be wise to compare them both. How to compare them? Compare specs, prices etc. and if a cloud service does the job better or cheaper – then it is time to move. That's not easy, but rather straight-forward and well described so far [17], [51], [54]–[68], [71]. All is needed is to structure those QoS attributes in a way to fit the research purpose.

Deliverable: Taxonomy of cloud service QoS attributes.

4) Functional requirements. What types of cloud solutions exist?

Research task: Develop a super-taxonomy based on different existing standards, also reserving spots for new technologies promising new cloud services types.

The National Institute of Standards and Technology (NIST) [72] has divided the cloud computing model into three service models: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). However, since then new types of cloud services emerged and there is no clear structure of those three [18]. When moving to cloud, what kind of service will fit the purpose?

Deliverable: Taxonomy of available cloud solutions from a functional perspective.

5) Architectural requirements. What cloud service's architectural requirements exist?

Research task: List most of common deployment tactics and other architectural differences for any/specific types of solutions. Structure them based on preferably based on a known standard.

Cloud services of the same type can be deployed differently [15], [52], [53]. Among different payment options [52] and security measures [9], [11], [37], [38], [66], there are even completely new types of deployment tactics [73]–[78]. All these must be structured in a way to provide the base for decision making process.

Deliverable: Taxonomy of cloud service's architectural requirements.

RsQ2. Linking the steps. What requirements are affected by other requirements in the model?

Research task: Link the attributes of each step of the model which are dependent on each other.

For instance, what cloud services satisfy company's EA, or are well aligned with business process attributes? It is the missing gap where most companies rely on "gut feeling" when deciding what services are to be examined.

Deliverable: Rules and dependencies of attributes from each step of DSM.

RsQ3. Does the model work?

1) Proof of concept

Research task: Run a proof of concept with a business case. Show how an imaginary consultant chooses a cloud service for an imaginary company using 5 DSM steps.

Deliverables: Case study.

2) Expert validation

Research task: Construct a validation procedure. Validate the DSM concept with a series of expert interviews to determine potential usefulness of a DSS based on it.

Deliverables: Interview protocols and results.

3) DSS prototype

Research task: Construct a DSS tool mockup based on expert feedback and suggestions. Implement a part of the tool.

Deliverables: DSS tool mockup and a part of developed DSS tool.

2.2. Research methods

To answer the main research question of this study, we aim to bring a unifying structure to all those steps according to the research focus and define a solid decision support model.

Hevner et al. provide a set guidelines which help information systems researchers conduct, evaluate and present design-science research [79]. These guidelines are elaborated in their later work called "Information Systems Research Framework" [80] and applied to current thesis shown below:

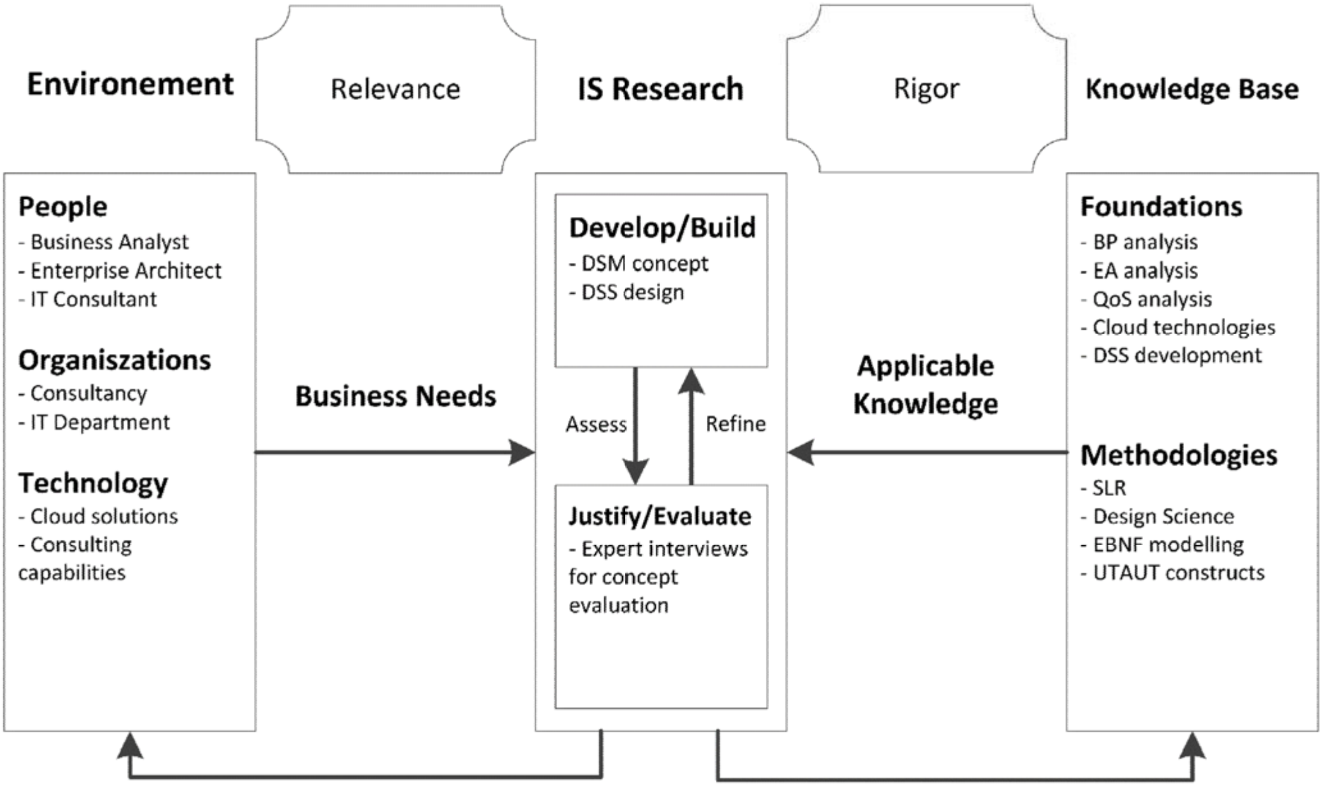


Figure 10 Research framework of current thesis (based on Hevner et al. [80]*)*

Following subchapters describe main research procedures – SLR used for making a DSM based on scientific knowledge, and validation procedure used for validating the concept. Specific methods used for DSM syntax definition and DSS prototype development are described in corresponding chapters.

2.2.1. Literature review procedure

It was decided to structure the findings as taxonomy trees for all models in SLR. The term taxonomy is widely known from biology as the science of defining groups of biological organisms on the basis of shared characteristics and giving names to those groups. However, it is more and more used in other areas of science such as information technology to deal with the same problem – define groups and categories in order to bring structure, clarity and usability to high numbers of objects. In the very first chapter of this work, the task is to bring structure to all existing (and promising to be) cloud solutions. In order to come up with a unified taxonomy for each group, a solid SLRs will be performed on different approaches that exist.

In this section the six steps of the SLR procedure are described: (i) strategy, (ii) identification phase, (iii), screening phase, (iv) reference snowballing, (v) eligibility procedure, and (vi) what is included. Figure 11 gives a graphical overview of this. This procedure is followed each time an SLR is performed in current paper. To illustrate the procedure, the following example shows SLR on cloud solutions functional taxonomy. To see SLR details (i.e. key words, screening protocols etc.) for other taxonomies please refer to Appendix A.

2.2.1.1. Search strategy and process

The rigor of the search process is one of the distinctive characteristics of systematic reviews [81]. To try to implement an unbiased and strict search, it was decided to set a precise publication time span, employ popular literature libraries, alternatively use a set of short search strings, and supplement a manual search to compensate the automated search for the lack of typical search keywords. As the term "Cloud

Computing" started to gain popularity in 2006 [10], the publication time span was restricted since the year 2006 until present.

With reference to the existing SLR protocols and reports for referential experiences, as well as the statistics of the literature search engines [81], it is believed that the following five electronic libraries give a broad enough coverage of relevant primary studies:

- Google Scholar (http://scholar.google.com)
- ACM Digital Library (http://dl.acm.org/)
- IEEE Xplore (http://ieeexplore.ieee.org)
- ScienceDirect (http://www.sciencedirect.com)
- SpringerLink (http://www.springer.com)

According to [82], these five databases cover most of the ISWorld's top fifty IS journals.

2.2.1.2. Identification

In Table 1, the used search keys and their relationships can be found. These search keys are selected based on their relevance to the subject. Some of the other search terms were also tried but with less success and are not worth mentioning. For each search key the databases were scanned. These search keys yielded a total of 1,584,856 articles (some of which of course were duplicates).

		Taxonomy					
Cloud solutions		Categories		XaaS			
Cloud services		Categorization		SaaS		Taxonomy	
Cloud computing	**x**	Benchmark	**+**	PaaS	**x**	Classification	**= 40 search keys**
Cloud technologies		Clustering		IaaS		Types	
		Classification					
		Types					

Table 1 RsQ1.4 search keys

Example search query: ("cloud solutions" OR "cloud technologies" OR "cloud services" OR "cloud computing") AND (taxonomy OR categories OR benchmark OR types OR classification OR clustering OR categorization) = 16,800 in Google Scholar from 2006 until 2015.

2.2.1.3. Screening

Of the roughly 1,5 million results in total, about 1,290 records were screened. This is done by reading the titles and when deemed possibly relevant for this research, also the abstract. Per relevant keyword the first 10 to 60 results were searched through until nothing close to a good match had been found. The following Table 2 illustrates this procedure.

Search keys:				Results:	Irrelevant after:
Cloud solutions	x	Taxonomy	=	16900	50
		Categories	=	43400	40
		Categorization	=	12900	40
		Benchmark	=	20300	40
		Clustering	=	20000	30
		Classification	=	37600	50
		Types	=	233000	30
Cloud services	x	Taxonomy	=	17100	60
		Categories	=	90100	40
		Categorization	=	17100	10

		Benchmark	=	19100	40
		Clustering	=	17800	30
		Classification	=	83700	60
		Types	=	285000	40
Cloud computing	x	Taxonomy	=	17900	30
		Categories	=	73800	20
		Categorization	=	13800	10
		Benchmark	=	21400	10
		Clustering	=	22600	30
		Classification	=	65900	50
		Types	=	198000	20
Cloud technologies	x	Taxonomy	=	17300	40
		Categories	=	44200	10
		Categorization	=	17200	10
		Benchmark	=	18800	20
		Clustering	=	19000	10
		Classification	=	31600	40
		Types	=	20500	50
XaaS	x	Taxonomy	=	293	40
		Types	=	866	20
		Classification	=	587	30
SaaS	x	Taxonomy	=	4200	60
		Types	=	19100	20
		Classification	=	10100	20
PaaS	x	Taxonomy	=	3810	60
		Types	=	17700	10
		Classification	=	7730	20
IaaS	x	Taxonomy	=	3400	50
		Types	=	15300	20
		Classification	=	5770	30

Table 2 RsQ1.4 SLR screening procedure protocol

Based on the title and abstract screening of these 1,290 results, 89 articles are deemed interesting for this research. Consequently, 1,201 screened articles are deemed not interesting enough and are dismissed.

2.2.1.4. Reference snowballing

In turn, SLRs on cloud computing subject such as [4] and [5] as well as some of most relevant articles were analysed in references or active directories in order to ensure that the most of relevant sources are reached in SLR procedure. This yielded 17 articles more.

2.2.1.5. Eligibility

In order to assess the eligibility of the remaining 106 articles, the full texts of these articles were screened. The articles are divided into four different categories:

1. Addressing cloud solutions taxonomies;
2. Too specific, e.g. addressing types of IaaS solutions only. Those were stored for further use in next chapters if needed, but not included here.
3. Too broad, e.g. addressing generalities only or summing up other articles without sufficient level of details. Those were excluded from the review.
4. Not relevant, e.g. addressing mobile apps or big data analysis in bioinformatics. A huge number of articles screened reviled a surprising thing. Only one paper was actually addressing real clouds!

This honourable paper is worth mentioning: Martínez et al. (2009). "Effects of land use change on biodiversity and ecosystem services in tropical montane cloud forests of Mexico." *Forest Ecology and Management*, *258*(9), 1856-1863.

2.2.1.6. Included

A total of 63 articles were included in the literature review on cloud solutions taxonomy. The whole SLR procedure for RsQ4 is shown in Figure 11 below:

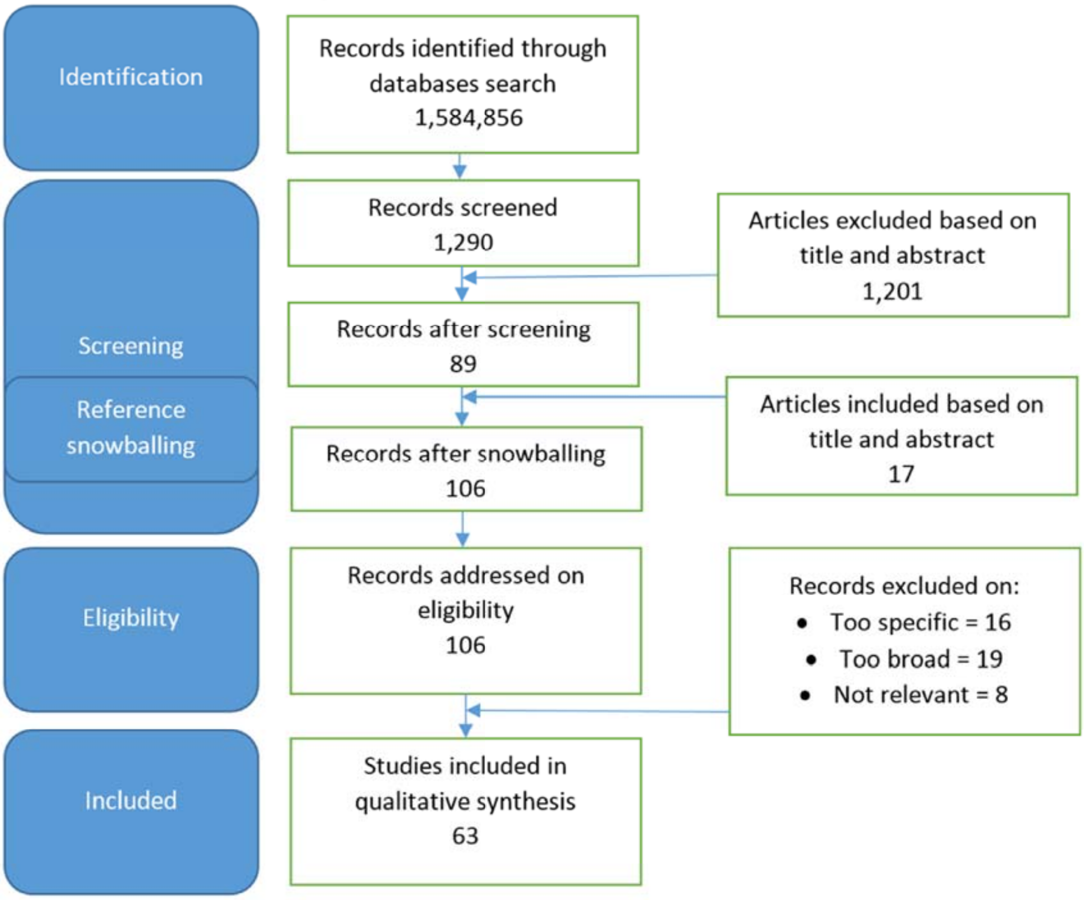


Figure 11 Four phases of RsQ1.4 SLR

2.2.2. DSS concept validation

DSM, which basically is the theoretical concept behind DSS, is evaluated in expert interviews. It is believed that experts that work in the field of Consultancy or Enterprise Architecture and that deal with "clouditization" have a great insight into which business process properties or EA attributes influence a process of choosing a cloud service. This knowledge can help in expanding attributes cross-linkage in the DSM.

The second aim of these interviews is to check whether the concept of proposed DSM is efficient in solving the goal – choosing a cloud service for a given company. Interviewees can tell whether the process used in DSM does not contradict with their experience (does not miss vital steps), and whether the prototype is efficient in use (an expert system can save time of one having a goal to pick a cloud).

To evaluate potential usefulness of a DSS tool, we will use a technology acceptance model by [83] which helps to identify usage intention and behaviour. If the professionals from the field are willing to use such a system in their daily business, we might also ask for any improvement tips or suggestions on functionality of the tool.

2.2.2.1. UTAUT constructs

DSS concept is evaluated using unified theory of acceptance and use of technology (UTAUT) constructs [83]. In expert interviews three constructs are used to evaluate the concept of DSS - performance expectancy, effort expectancy and facilitating conditions. This evaluation procedure will help to understand whether potential users of the DSS intend to use it in their work. Also, these interviews will help to gather more preferences on DSS functionality and design.

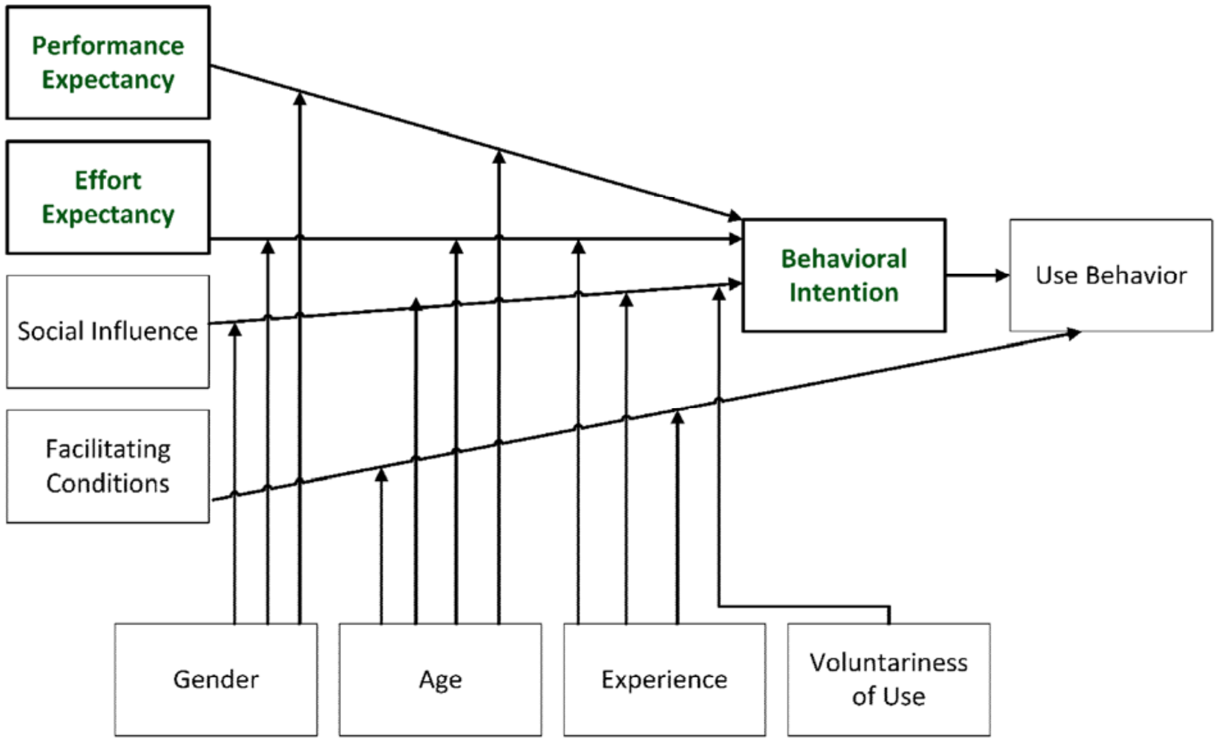


Figure 12 UTAUT Research model [83]

The objective of the evaluation is to find out how experts in the field of Cloud Computing perceive DSM concept and whether they are willing to use potential DSS in their work. Semi-structured interviews are based on UTAUT model constructs:

- Performance Expectancy,
- Effort Expectancy, and
- Behavioural Intention.

We don't apply Social Influence, Facilitating Conditions and Use Behaviour constructs to this research, while we only evaluate a concept instead of a fully developed tool or its prototype which interviewees can actually use. Also, we assume that interviewees experience is rather same while they all are experts in the field of Cloud Computing which by now is not as complex as might be in future. Voluntariness of Use does not apply to current research, while interviewees are not forced to use the tool by employer or anybody else. According to UTAUT, the influence of both performance expectancy and effort expectancy on behavioural intention is be moderated by gender and age at the same time. That means, that gender does not influence behavioural intention if the age is the same – which is exactly our case. All interviewees are of about the same age, therefore we exclude these factors as well.

The important question that arise is how many interviews would lead to a sufficient theoretical saturation [84]? Consensus theory developed by [85] is based on the principle that experts tend to agree more with each other (with respect to their particular domain of expertise) than do novices and uses a mathematical proof to make its case. Even small samples can be quite sufficient in providing complete and accurate information as long as the participants possess a certain degree of expertise about the domain of inquiry [86]. Romney et al. [85] calculated that samples as small as four individuals can render extremely accurate information with a high confidence level (.999) if they possess a high degree of competence for the domain of inquiry in question. However, most of the researchers agree that six to twelve expert interviews be enough to achieve a desired research objective, while a selected group is relatively homogeneous (Enterprise Architects), the data quality is fair (competence level), and the domain of inquiry is not vague (almost a daily routine of Enterprise Architects).

Fontana et al. [87] suggests stopping such interviews when the data desired are elicited. Therefore, considering our individual interviews, it is decided to use a semi-structured procedure – first asking UTAUT questions, and then having a discussion until the most valuable thoughts if the interviewee are expressed.

3. Taxonomies

After a series of five rigor literature reviews, five taxonomies are formed to further be used in DSM. This chapter is structured as follows – first, an introduction to a taxonomy topic is given by a literature base, then a taxonomy is presented. This is the same for all five taxonomies. Each taxonomy is described from the higher level of abstraction to the lowest one.

First taxonomy is targeted on functional requirements for the potential cloud service. A user selects what type of cloud is required – IaaS or SaaS, and if SaaS then which one, CRM, ERP, BPM etc.? This taxonomy is not accumulating all possible cloud services, but it presents a structured taxonomical approach on possible functionality. It includes up to date technologies and while being formal keeps place for upcoming technologies and solutions. Second one is focusing Business Process attributes, or the attributes of a set of BP. It allows a Business Analyst to describe BP attributes both from descriptive perspective and from enactment perspective. Third taxonomy is focusing on Enterprise Architecture attributes. This is a lower level of granularity after BP attributes, and it deals with IT from the company-wide perspective. Fourth taxonomy is focusing on Quality of Service attributes. This is a well-known step in choosing any service. Again, in this taxonomy we gathered most of the important attributes for cloud services from respected literature sources. Fifth taxonomy is about specific architectural requirements to be implemented when adopting a service. All five taxonomies can be accessed here: http://gocloud.ga/DSM.html.

All taxonomies are constructed using competent literature sources that offer overviews on corresponding topics. By comparing those, it is possible to unite and structure the terms and concepts that form the taxonomy. Some of the figures illustrating taxonomical branches include cloud services examples – those were put in by the Author to increase the understandability of the corresponding cloud terms.

3.1. Functional requirements taxonomy

There is one important fact to be mentioned before we proceed to the taxonomy description. While cloud service is a special method of service delivery, variety of functions performed by cloud solutions are called "delivery models" i.e. how they deliver a service implies what they deliver. Thus, functional taxonomy is commonly called the taxonomy of service delivery models.

3.1.1. Literature base

There are different types of cloud computing taxonomies that exist in the literature: [17]–[46] Some are accentuating concepts, some focus on specific aspects such as security, performance, etc., while others present an overview of vendor solutions present in the market. Some authors focus solely on reviewing taxonomy research done by others: [18],[41]–[46]. In terms proposed by [18] we will focus on conceptual taxonomies leaving performance and security ones aside.

The National Institute of Standards and Technology (NIST) [72] has divided the cloud computing model into three service models:

- *Software as a Service (SaaS).* The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure2. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.
- *Platform as a Service (PaaS).* The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but

has control over the deployed applications and possibly configuration settings for the application-hosting environment.

- *Infrastructure as a Service (IaaS).* The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls).

As proposed by [42], the first-level classification criterion is the type of resource offered to the cloud tenant (following NIST classification), and the second level is comprised of the abstractions of a dedicated type which are gathered from specific literature sources. First of all, current taxonomy is the best of breed from existing taxonomies by NIST [88], Intel [89], Bhaskar Rimal [22], Anton Beloglazov [90], Forrester [91], OpenCrowd [92], web-published Peter Laird's taxonomy, and a few others. Second, it includes up to date technologies and while being formal keeps place for upcoming technologies and solutions.

Figure 13 Functional requirements taxonomy high-level view

Full taxonomy can be viewed here: http://gocloud.ga/DSM.html#first

Now, we are going to discuss each solution's type in particular. Starting from the simplest one in terms of offered service (IaaS) the following subchapters present all five types ending with the very new one – Edge Computing.

3.1.2. Infrastructure as a Service (IaaS)

It is proposed by most of the authors [30], [36], [41], [43], [45], [93], [94] to differentiate IaaS services by their usage type – *Computation* (e.g. Virtual Machine) or *Storage* purposes.

Figure 14 Functional taxonomy: IaaS

[42] proposes the differentiation of computation IaaS solutions based on two different layers of abstraction - *Hardware* only where a tenant is free to install and configure arbitrary software including the OS, and a fully managed *OS* including underlying hardware resources. They also propose to divide *Storage* solutions based on the view a tenant has on the data – raw *disk space*, *file system* or *database* structure. Different notions of Storage as a Service presented in [20] include: DaaS - Data-Storage as a Service; StaaS - Storage as a Service; dSaaS - data Storage as a Service. As indicated by [20], [22], [26], raw disk space may also be considered as a type of HaaS solution (Hardware as a Service), but since we already have divided IaaS into Computation and Storage it is logical to address it as a Storage.

As for the Hardware, we do not put any subgroups there because possible hardware configurations are too specific. Thus it is considered fine to specify that aside from the taxonomy, e.g. cars can be grouped in different categories, but it is redundant to keep all the specifications for each car in the taxonomy. A dedicated taxonomy may address it.

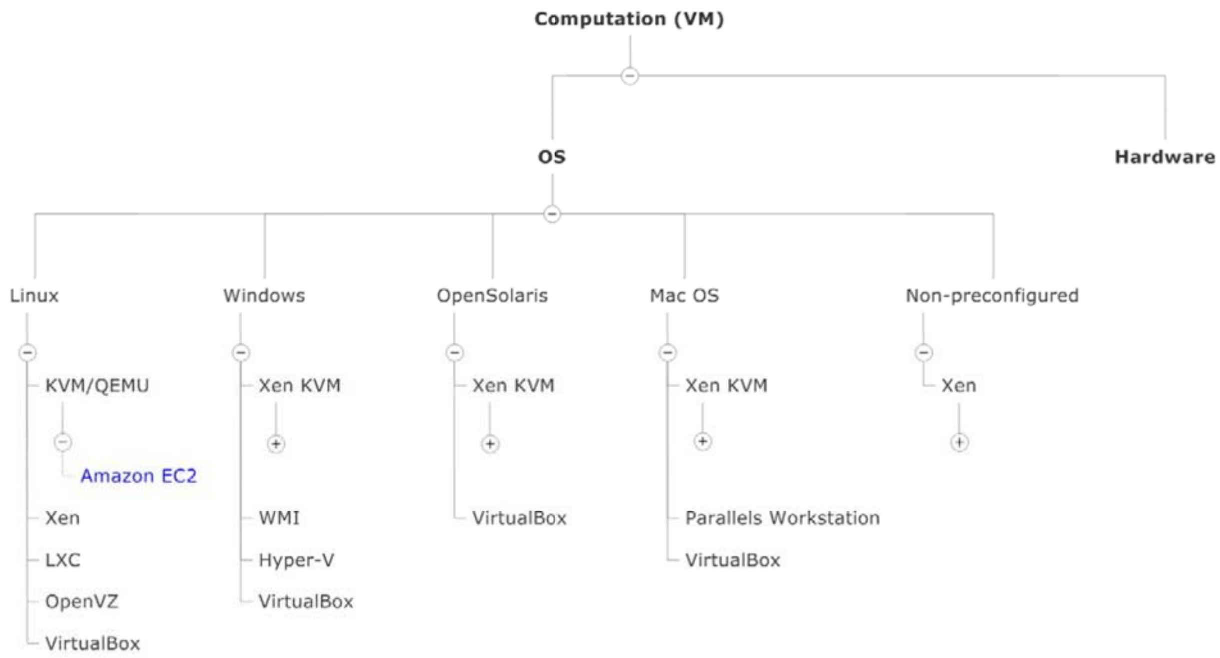


Figure 15 Functional taxonomy: IaaS - Computation

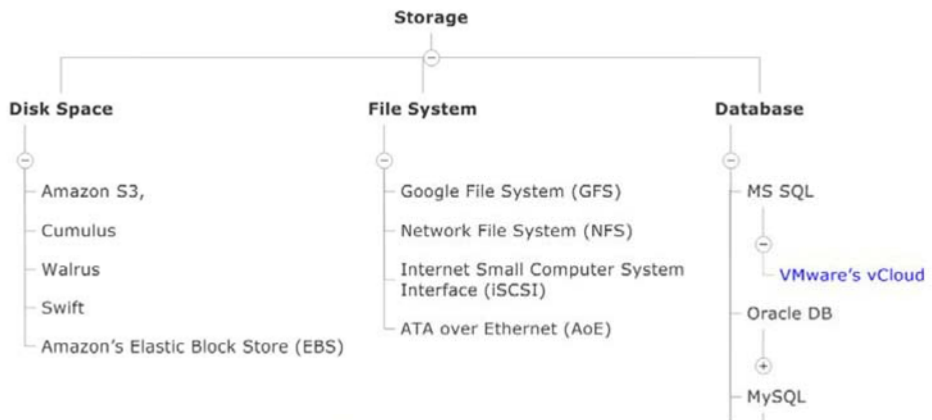


Figure 16 Functional taxonomy: IaaS - Storage

3.1.3. Platform as a Service (PaaS)

[43] propose to differentiate PaaS offerings into *Business* and *Development* ones. This differentiation is used here because there are many kinds of PaaS offerings and any of them can be put in one of these two groups.

Figure 17 Functional taxonomy: PaaS

As proposed by [42], Development PaaS offers are divided into *Runtime Environments* (e.g. Java SE) where a tenant can deploy his own applications via cloud API, and *Software Frameworks* where a tenant can deploy his application in a cloud software platform tailored towards a dedicated application type or use case (e.g. Servlet, Bean). Again, some examples are put here to make the understanding of the grouping better, but the runtime environment can break into very specific offers. Therefore, following [42] the current taxonomy stops within the bold letters in the Figure 18:

Figure 18 Functional taxonomy: PaaS - Development

As a part of Software Framework it is worth mentioning *Mobile Backend* (*MBaaS* or "backend as a service"), it is a subset of rather new and fast emerging services. Such services provide web and mobile applications developers with a way to link their applications to backend cloud storage and APIs exposed by back end applications [95].

Then, as for PaaS offers for business, there is a great number of them providing different kinds of services. We will include most common ones using the grouping structure of existing vendor solutions from a large cloud solutions catalogue [92].

Monitoring and Reporting – such services allow a tenant to be aware of several things while his application is running on PaaS: resources utilization, system performance, restarts, log files, application metrics, etc. These stats (such as log data) can also be reported for instance via a web console with real-time statistics or aggregated to another application or service.
By a *Cloud Broker* we mean a service that provides tools to manage services on more than one cloud infrastructure platform.

Backup & Recovery – such services allow a tenant to secure and manage backups not only of his file systems and raw data stores on servers and desktop systems but also of his application running anywhere. Such backups offer any type of recovery granularity (from a single email to the complete database) and scalability (from single instances to large physical or virtual implementations) based on user requirements.

Integration Platform as a Service (iPaaS) – such services are used for integrating applications ranging from cloud-to-cloud integration to custom application integration. Also, such solutions make possible to integrate existing enterprise software with cloud infrastructure servers from external providers.

Big Data – such services are designed especially for the analysis of large or complex data sets that require high scalability. These Big Data solutions are fed with massive amounts of raw Big Data from tenant's legacy databases, data feeds, web resources etc. which is then used for sophisticated analytics. This approach allows to deliver high performance on off-the-shelf hardware.
This PaaS branch is divided in two to provide the distinction between *Big Data Analysis (BDaaS)* and *Big Data Intelligence (BDIaaS).* While BDaaS is a platform providing any services for Big Data Analysis, BDIaaS are the platforms for creation of business intelligence applications such as dashboards, reporting systems concerned with Big Data analysis.

According to the concepts found in the literature, we can separate and group them as a Business PaaS branch:

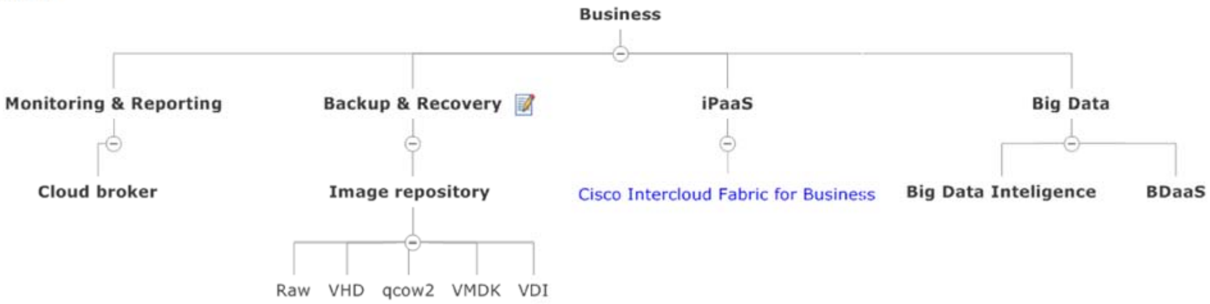


Figure 19 Functional taxonomy: PaaS - Business

3.1.4. Software as a Service (SaaS)

This branch of the taxonomy is mainly based on market analysis. The relevant SaaS solutions categories were gathered from the biggest web resources containing vendor solutions. The examples of those are: CloudBook.net, CloudShowPlace.com, and CloudTaxonomy.Opencrowd.com.

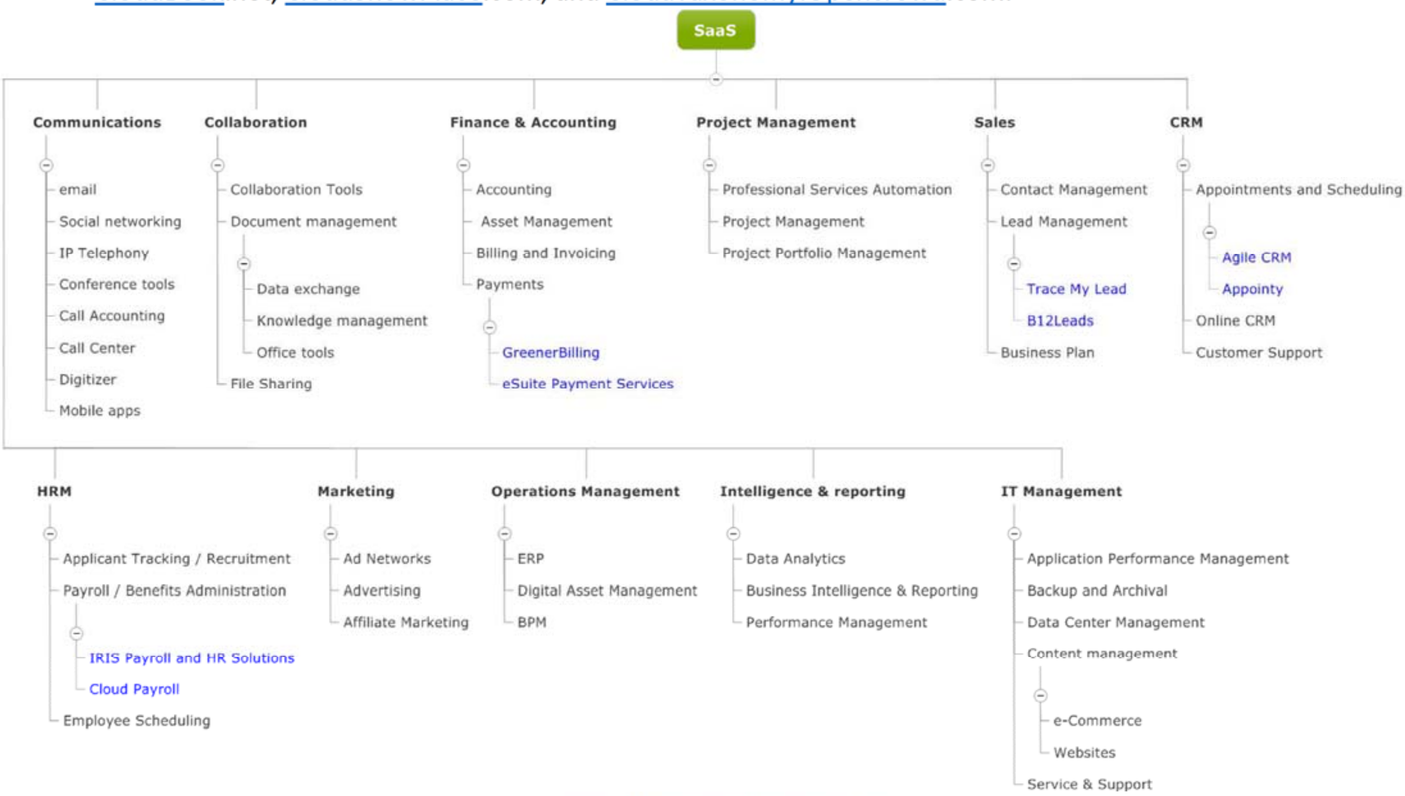


Figure 20 Functional taxonomy: SaaS

As it can be seen from the model above, most of categories relate to dedicated business functions or services. However, there are more categorization options (such as grouping solutions by industry or vendors). It is believed that the chosen approach ensures more efficient fit to the research needs than the others.

3.1.5. Extending service delivery models - Business Process as a Service (BPaaS)

However, these three main service delivery models can be extended. Some authors [20], [38]–[40] together with IBM and Gartner extend the basic model with *BPaaS – Business Process as a Service* which is meant as the delivery of whole business process outsourcing. We place BPaaS on top of the three previous service delivery model levels, while BPaaS is a more complex model composed of SaaS solutions. [38] concludes that "we have been seen the trend of Business Process Outsourcing (BPO) for last few decades". Gartner predicts that BPaaS market will grow from $84.1B in 2012 to $144.7B in 2016, generating a global compound annual growth rate of 15%, which makes this cloud service delivery model very promising.

As for the BPaaS taxonomy, it mainly depends on present market offerings. That is because BPaaS, in simple words, consists of SaaS offerings linked together to produce synergy and ease of use. So, it depends on a particular company, what IT (parts of -) processes goes to outsourcing – it may be HR or it may be HR + Finance + Accounting and so on. This trend can be scaled up to full business outsourcing, e.g. when a franchise is bought and all the IT services are provided to the shareholders as one service package.

There are some services which were usually referred to SaaS ones, but are more complex in their delivered value indeed. Thus, they deserve to be put in BPaaS category because they are delivering a set of auxiliary business process services.

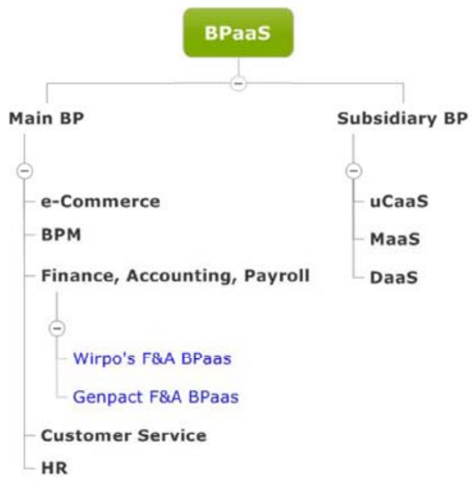


Figure 21 Functional taxonomy: BPaaS

Communication as a Service (CaaS/uCaaS for unified ...) includes enterprise level VoIP, VPNs, PBX, Unified Communications, etc.

Mobility as a Service (MaaS) provides ubiquitous access and connectivity to enterprise data and applications from any device and any place

Desktop as a service (DaaS) is a form of virtual desktop infrastructure (VDI) in which the VDI is outsourced and handled by a third party. Also called hosted desktop services, desktop-as-a-service is frequently delivered as a cloud service along with the apps needed for use on the virtual desktop.

3.1.6. Extending service delivery models - Edge/Fog computing

There are two terms used for this same technology. The term edge based cloud computing refers to a network of edge systems that provide the services currently provided by data center clouds [96]. Cisco promotes the term Fog computing, which extends cloud computing to the edge of the network to provide compute, storage, and networking services between end devices and data centers [97], [98]. This technology thus enabling a new breed of applications and services [99].

Edge computing breaks up to personal usage services and business ones. There is also a branch of this technology dealing with intelligent environment creation (e.g. smart cities), but since an end-user is a nation itself we cannot talk about it as a delivered cloud service.

Personal use is known as Internet of Things (IoT) – e.g. a smart house where many devices are connected to each other building one ecosystem, cars that interact with each other providing benefits for a driver etc. This trend is heading to the Internet of Everything (IoE), where all electronic devices are interconnected [100]. It is worth mentioning that IoT now starts bringing the power of device ecosystems to enterprises, and there are some ways to architect the IoT implementations for enterprises already[7].

Enterprise solutions represent mainly a network (fog) with M2M interactions, where data is analysed at the edge and information or a service is delivered to the end-user. This type of service delivery provides the end-user with end-to-end M2M solutions including connectivity, integrated terminals, and applications (possibly in partnership with other vendors of that ecosystem). Smart machines build on the foundation of big data, advanced analytics, context-rich systems, and ongoing advances in artificial intelligence and robotics hardware[8].

The Machina Research M2M Forecast Database provide a comprehensive guide to the global M2M market opportunity [101]. Their segmentation of the market is illustrated in Figure 22.

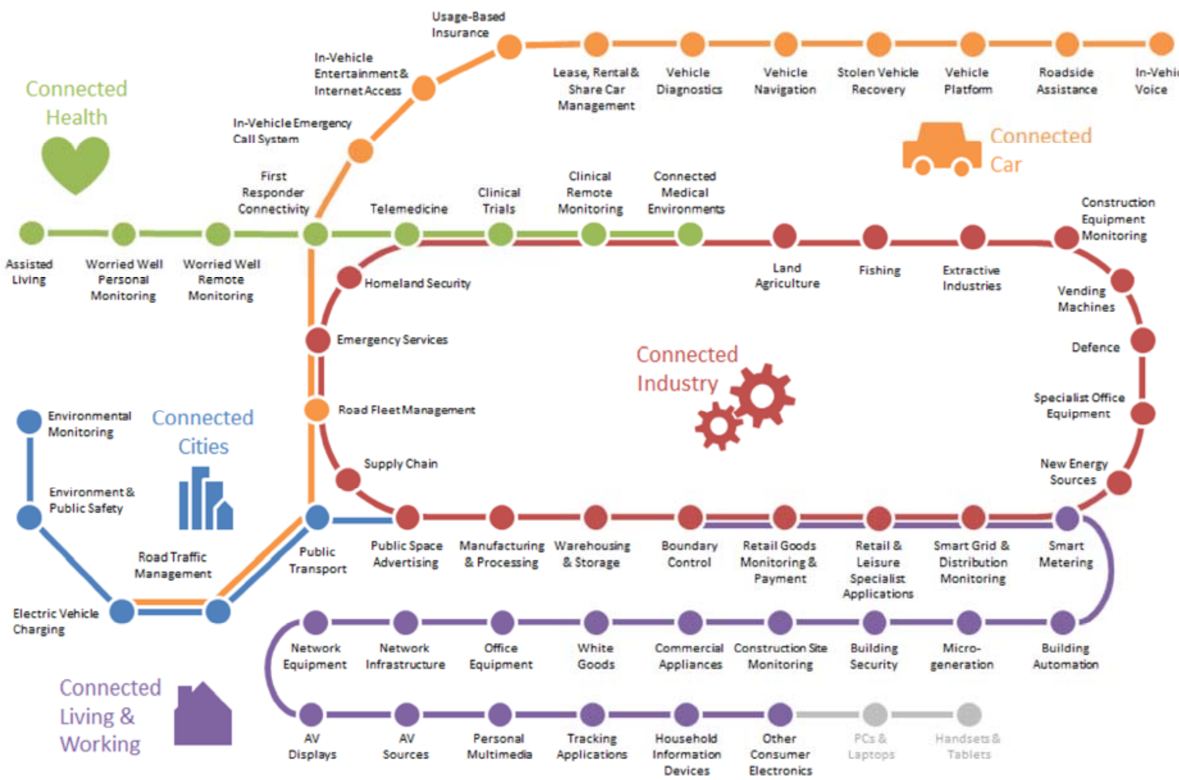


Figure 22 Application groups across M2M sectors (retrieved from [101])

Structuring all mentioned above, this branch of functional taxonomy is presented in Figure 23 below:

[7] Gartner report on five available IoT styles in EA: https://www.gartner.com/doc/2854218?ref=ddisp
[8] Gartner report on smart machine technology landscape: https://www.gartner.com/doc/2849320?ref=ddisp

Figure 23 Functional taxonomy: Edge/Fog computing

Summing the first taxonomy up, it consists of five functional groups:

- Infrastructure as a Service, which provides processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run own software;
- Platform as a Service, which provides the capability to deploy and control the deployed applications;
- Software as a Service, which provides running applications;
- Business Process as a Service, which outsources the whole business process (in terms of software support);
- Fog computing (edge computing) networks, where some computation is performed within the network of devices instead of a data center.

To this data, any cloud service can be put into one of those categories, which makes the taxonomy up to date. Still, the reader should bear in mind that during next 5 – 10 years, new types of services will emerge creating the need to extend the taxonomy.

3.2. Business Process attributes taxonomy

Second taxonomy is mainly based on business and management literature. The grouping of concepts in it follows understandable patterns which are now new due to the maturity of the topic from scientific perspective. The aim of this taxonomy is to be useful in describing business requirements to the next steps of the model – EA requirements and QoS requirements. Therefore, it contains only the attributes which are the most relevant to the DSM for choosing a cloud service.

3.2.1. Literature base

Business process quality include all components of a business process such as activities of the process, the actors performing these activities, the objects handled and created by the process as well as the resources necessary for execution [102]. Great body of literature is focused on BP modelling. Among these sources, some were extended to support QA descriptions. Different BP modelling techniques and their possible extensions with QA attributes were scanned to find out what are the most important QAs modelled during BP analysis. The overview of those techniques is presented by [51]:

Business process modelling techniques	Modelling set(s)	Selected references
Flowcharts	Diagrammatic models	- (Knuth, 1963) - (ChapinN., 1971) - (Chapin, 1974) - (Feldman, 1998) - (Lakin et al., 1996)
IDEF	Diagrammatic models	- (Mayer et al, 1994) - (Menzel and Mayer, 1998) - (Peters and Peters, 1997) - (Zakarian and Kusiak, 2001) - (Zakarian and Kusiak, 2000) - (Zakarian, 2001) - (Badica et al., 2003) - (Shimizu and Sahara, 2000) - (Zhou and Chen, 2002)
RADs	Diagrammatic models	- (Ould, 1995) - (Holt, 2000) - (Phalp and Shepperd, 2000) - (Badica et al., 2003)
UML	Diagrammatic models, Business process languages	- (Quatrani, 2001) - (Kim et al, 2003) - (Wohed et al., 2004)
Petri-nets	Diagrammatic models, Formal / mathematical models	- (van der Aalst, 1998) - (Li et al., 2004b) - (Donatelli et al., 1995) - (Raposo et al., 2000) - (Peters and Peters. 1997)
Business process models based on mathematical or algorithmic models	Formal / mathematical models	- (Hofacker and Vetschera, 2001) - (Powell et al., 2001), - (Valiris and Glykas, 1999)
BPEL -BPML	Business process language	- (Reimer et al., 2000) - (Havey,2005) - (Grigori et al., 2004) - (Smith, 2003)
jPDL (jBPM)	Diagrammatic models, Business process languages	- (Koenig, 2004)

Table 3 Main BP modelling techniques, based on [51]

After a rigor literature study on the subject, a model that fits current research was found. Lohrmann et al. define Business Process Quality based on organizational targets which correspond to business process efficacy and business process efficiency respectively [103]. Based on these organizational targets and on the ISO quality definition they derive a definition framework for business process quality. It consists of four dimensions:

> "*Business process efficacy* means the effectiveness of a business process with respect to achieving its business objective.
>
> *Business process efficiency* means the effectiveness of a business process with respect to limiting its impact on resources.
>
> *Business process design & implementation quality* is the degree to which an actual business process model enables business process efficacy, achieves business process

efficiency during design & implementation, and enables business process efficiency during its enactment.

Business process enactment quality is the degree to which a set of business process instances achieves business process efficacy and business process efficiency."[103]

For purpose of this model, only enactment quality will reside as quality attribute leaving other three attributes from management field aside. Descriptive attributes which influence EA and thus all IT decisions are added to the model as well.

Figure 24 BP attributes taxonomy high-level view

Full taxonomy can be viewed here: http://gocloud.ga/DSM.html#second.

Further sub chapters are describing BP descriptive attributes and enactment quality attributes.

3.2.2. BP descriptive attributes

Descriptive attributes that characterise a business process presented in Figure 25 were found in literature from fields such as Management, Business Process modelling, IT security, Performance engineering, etc. The sources were scanned for formulations specifying a causal or definitional relation related to one of the sub-characteristics: criticality, maturity, availability, integrity, and confidentiality. The grouping structure was made by the Author.

Figure 25 BP attributes taxonomy: Descriptive attributes

BP are called mission-critical if its failure will result in the failure of all business operations. That may be company's core process or one of subsidiary processes without which the core process cannot be carried out. Business critical processes are considered to be non-vital but utmost important to normal business functioning. These processes can be disrupted such that they will not strongly impact the ability of the organization to achieve its minimum business continuity objective. The rest of ancillary processes that are carried out in order to permit or facilitate the critical ones, are labelled as "Low"-criticality in our taxonomy.

Moving to BP maturity levels, Macintosh (1993) defines five levels of process maturity [104]:
1. Initial - setting up of processes,
2. Repeatable - repeatable processes,
3. Defined - documented processes standardised throughout an organisation,
4. Managed - measured and controlled processes, and
5. Optimising - continuous process improvement.

Availability, in our case, refers to the time when the BP needs IS support. E.g. support desk that requires a corresponding software works on office hours, while a website must be available 24/7.

Integrity (also wholeness, continuity, consistency etc.) usually refers to a business process carried out as whole and not being split by parts.

Confidentiality of a BP refers not only to data that BP operates with, but also to any other artefacts of it such as, information, employees or physical devices.

3.2.3. BP enactment quality attributes

Enactment quality refers to how efficiently the process is carried out. Lohrmann et al. [103] define four dimensions of BP enactment quality attributes shown below:

Figure 26 BP enactment quality attributes

Human and non-human resources are rather straight forward to understand. A function is a basic building block, part, step or one of activities within a BP. Based on related work from software engineering Lohrmann et al. identify eight attributes that are applied to BP context: "We took the ISO/IEC 9126 software product quality characteristics for resources and also adapted them for activities. For information objects we took the ISO/IEC 25012 data quality characteristics. The actor characteristics we developed based on QI from practice." [103] The first three enactment quality groups of attributes are shown below as they were formulated by Lohrmann et al.

Figure 27 BP quality attributes for human resource

Figure 28 BP quality attributes for non-human resource

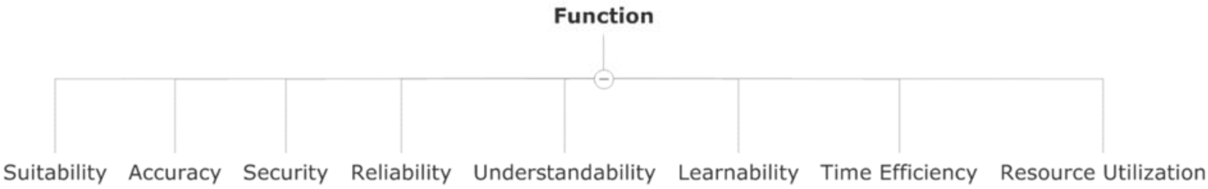


Figure 29 BP quality attributes for BP function

Attributes of information quality are better elaborated in [105] and therefore included here instead of QoBP attributes:

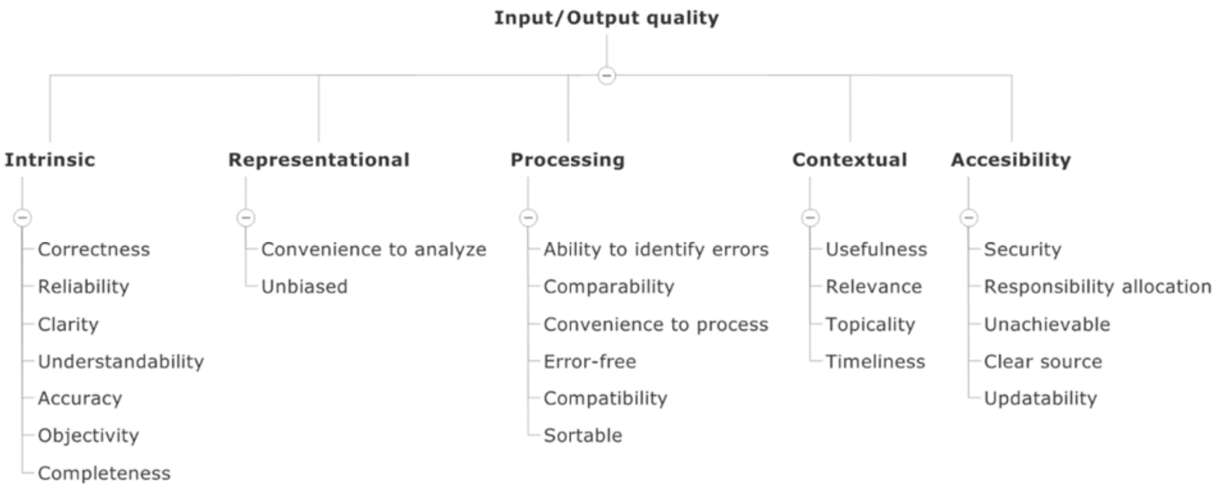


Figure 30 BP input/output quality attributes

To conclude this sub chapter, BP attributes taxonomy divides into:

- BP descriptive attributes – criticality, maturity, confidentiality, etc., and
- BP enactment quality, which is about how the process is carried out – human and non-human resources, information quality etc.

This is a general BP attributes taxonomy, but it is vital not to miss high-level business requirements during the requirements gathering procedure.

3.3. EA Quality attributes taxonomy

In this part of DSM, Enterprise Architecture is used for enterprise analysis on a lower level of granularity after the previous step – BP attributes assessment. The taxonomy is also mainly based on the literature thanks to a considerable level of maturity of the scientific knowledge in the field.

3.3.1. Literature base

The International Standards Organization (ISO) has done an effort to unite the different views on quality in a general definition stating that quality is "the total of properties and characteristics of a product or service that are relevant for satisfying specific requirements and obvious necessities" However, Enterprise Architecture is not a single service or a product. Therefore, most authors working on EA quality subdivide it in a number of quality properties that each address a particular EA view [60], [59], [61].

Figure 31 Four common views on EA

Following this widely accepted paradigm of four views on EA, proposed taxonomy includes four architectural components. To address EA maintainability, four components derived from Doudi [62] are included as shown in the bottom of Figure 32. Physical requirements such as in FURP quality metrics [106] are not in place due to the nature of cloud architecture – hardware is at the cloud vendor.

Figure 32 EA QAs taxonomy high-level view

Full taxonomy can be viewed here: http://gocloud.ga/DSM.html#third.

The following subchapters describe each of EA QA groups one by one starting from BP architecture quality and ending with supporting documentation quality.

3.3.2. Business Architecture quality

In current case, business architecture indicates whether EA consists of processes and thus services highly connected to each other or not connected to each other. One the one side, relations between BP indicate the extent EA services are interconnected to each other, e.g. shade data, rely on each other etc. That kind of knowledge dictates further QoS requirements to a system such as interoperability or throughput. One the other side, traceability between EA components shows how well those links between different EA systems are documented. Therefore we propose the following structure:

Figure 33 EA QAs taxonomy: Business architecture

3.3.3. Technology Architecture quality

A technology architecture is a set of guidelines for technology integration within an enterprise. Essentially, technology architecture is driven by the need to bridge the gap between IT and business people. It provides business people with a clear view on IT assets such as infrastructure, applications, and data.

Enterprise technical infrastructure must be scalable by either adding components or by replacing certain components with large or smaller scale components. Components should allow for a wide variety of configurations by organizing, classifying, and categorizing existing technology investments. While the number of technologies supported constrain operating costs, technology architecture is important both to controlling costs on the one hand and allowing integration with other services on the other. What is beneficial in cloud technologies is that the mentioned trade-off between interoperability of platforms and system portability is less critical. We propose to include the following groups of attributes in technology architecture taxonomy branch:

Figure 34 EA QAs taxonomy: Technology architecture

3.3.4. Software Architecture quality

Software quality is defined within the context of a situation – in each enterprise there are different attributes and requirements to software quality. Thus, it is logical to define high-level attributes such as size, age, and level of interaction with between components/services. To those mentioned attributes we also add quality of source code for projects where a new system is built and consequently development requirements are posed to its source code. We propose the following grouping, where the quality of source code attributes are shown as possible examples:

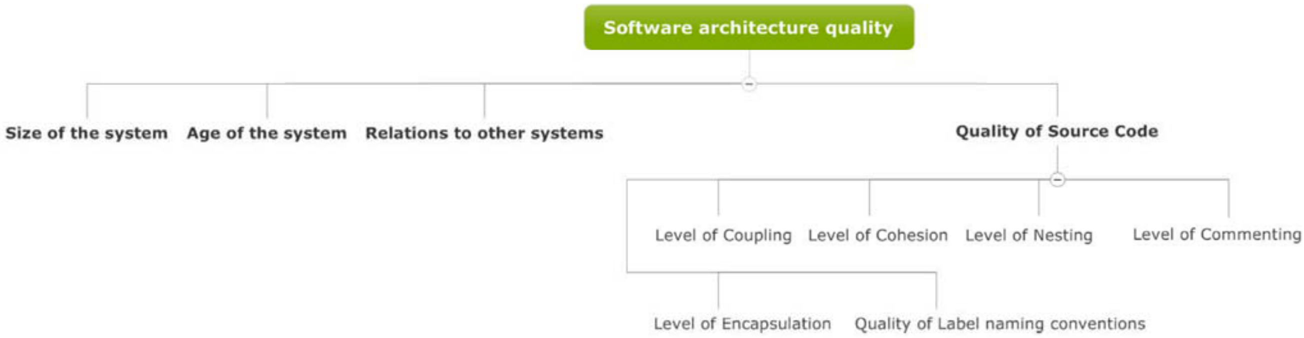


Figure 35 EA QAs taxonomy: Software architecture

3.3.5. Data Architecture quality

There always is a clear extinction between processes and data which these processes operate with. In this specific domain, *entity-relationship diagrams* and *flowcharts are used to model data, and, h*istorically, the best indicator of an IT organization's overall success was not how good its process models were, but rather, how good the data models were [107]. Thus, it is utmost important to focus on data architecture quality when assessing EA quality.

Along with level of normalization and data redundancy attributes retained from [62] EA quality attributes, a new attribute was added – quality of data of a business process itself. These data quality attributes are derived from [108] where authors propose BPMN notation extension for business process data. Requirements assembled and structured in this paper represent most of data quality requirements that are important for a business process.

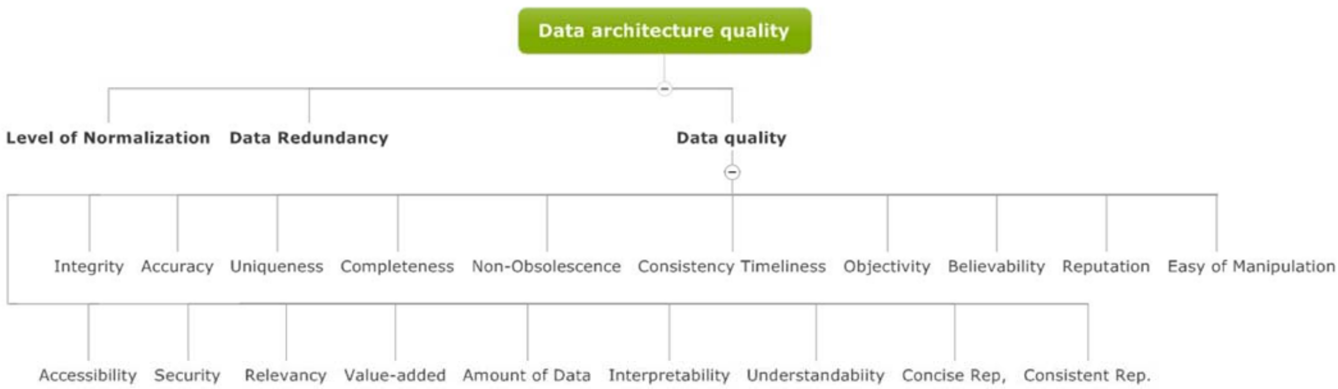


Figure 36 EA QAs taxonomy: Data architecture

3.3.6. Maturity of development & maintenance processes

Interface requirements, Implementation requirements, and Design requirements are added after FURPS+ Architecture Requirements [106]. While first two are intuitive to understand, a design requirement is simply a constraint on the refinement of an analysis mechanism, such as those listed in Functionality branch of QoS further in the paper. For example, a requirement for a relational database is a constraint on the refinement of a persistence analysis mechanism [106]. Originating from FURPS+, this branch of taxonomy structures as follows:

Figure 37 EA QAs taxonomy: Development & maintenance processes

3.3.7. Quality of maintenance policy

Quality of maintenance policy contain such attributes as cost of it, its management value and the risks associated with neglecting it. These attributes are tightly connected to BP attributes (such as criticality) on one side, and to the cost of proposed cloud service on the other. The second part is of course vital to consider when dealing with a trade-off between QoS attributes and the pricing of required QoS fulfilment by a vendor. We propose a simple but sufficient attributes division such as:

Figure 38 EA QAs taxonomy: Maintenance policy

3.3.8. Maturity of development & maintenance personnel

IT personnel knowledge and experience levels are opposed to such QoS attributes within integration or service support. This step is highly important while it helps to set the level of vendor or outsourcing partner involvement during service integration, use or modification. Thus, this taxonomy branch is as follows:

Figure 39 EA QAs taxonomy: Development & maintenance personnel

3.3.9. Quality of supporting documentation

As [109] indicate, enterprise architecture documentation was identified as one core cause of the non-compliancy of projects and the excessive amount of project supervision needed. If the set of design decisions for a potential cloud service are made incorrectly because of inconsistency or simply a lack of EA documentation, it may cause implementation issues or even redoing all the steps again. Therefore, the analysis of the EA documentation quality is an important step towards successful service integration without missing important requirements that were not properly documented. Mainly based on [109], excluding unnecessary attributes, this taxonomy branch is as follows:

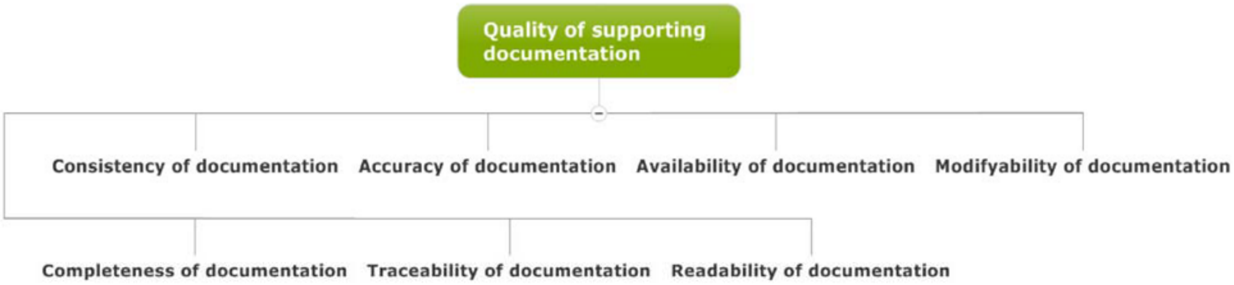


Figure 40 EA QAs taxonomy: Supporting documentation

Mainly based on FURPS+ quality metrics [106] and supplemented with more literature sources, EA quality attributes taxonomy consists of:

- Data architecture quality,
- Software architecture quality,
- Technology architecture quality,
- Quality of supporting documentation,
- Development and maintenance processes,
- Quality of maintenance policy,
- Maturity of development and maintenance personnel, and
- Business architecture quality.

Next step of the DSM will transform requirements derived from this step into one that address a cloud service. Thus, the chosen service will be aligned with EA of the company.

3.4. Cloud QoS attributes taxonomy

Quality of a cloud Service address specific requirements for a cloud service to be chosen. This is the last level of QAs granularity after business level and EA level. This is as well the most common QAs source for a service selection process. In the current paper, the Author tried to merge most up to date QoS taxonomical views and pick only the effective and important ones. The following literature overview presents the sources use so the reader can easily track the attributes origin.

3.4.1. Literature base

Several frameworks for evaluation of the quality of information systems have been proposed, see for instance McCall [110], Boehm [111] or FURP [112]. One of the most commonly cited frameworks is the ISO 9126-1 standard for software quality measurements [113]. ISO 9126-1 defines software quality in terms of six quality characteristics: functionality, reliability, usability, efficiency maintainability and portability. However, metrics associated with mentioned quality characteristics are intended for testing in conjunction with software development, rather than for enterprise architecture analysis, and are as such not directly applicable for our system quality analysis framework [59].

The current situation is that each cloud service provider has predefined measures to fulfil SLAs, and these measures often exclude critical attributes that the clients need to measure, e.g., initial cost of acquisition/transition, control of access and employee privileges, or scalability and flexibility of the provider to increase/decrease service provisions [64]. To address this situation and to ensure that measures are developed to be globally appropriate and to meet the needs of both public and private sector clients, the Cloud Services Measurement Initiative Consortium (CSMIC) was launched by Carnegie

Mellon University to develop the Service Measurement Index (SMI)[9]. SMI is based on a framework of critical QoS attributes and their measures to provide a standardized method for measuring and comparing any cloud-based services (IaaS, PaaS, SaaS, BPaaS etc.) with non-cloud services regardless of whether that service is internally provided or outsourced.

Can we count on the provider organization?	Can it be changed and how quickly?	How likely is it that the service will work as expected?	How much is it?	Does it do what we need?	Is the service safe and privacy protected?	Is it easy to learn and to use?
Accountability	**Agility**	**Assurance**	**Financial**	**Performance**	**Security & Privacy**	**Usability**
Auditability	Adaptability	Availability	Billing process	Accuracy	Access control & privilege management	Accessibility
Compliance	Elasticity	Maintainability	Cost	Functionality	Data geographic/political	Client personnel requirements
Contracting experience	Extensibility	Recoverability	Financial agility	Interoperability	Data integrity	Installability
Ease of doing business	Flexibility	Reliability	Financial structure	Service response time	Data privacy & data loss	Learnability
Governance	Portability	Resiliency/fault tolerance		Suitability	Physical & environmental security	Operability
Ownership	Scalability	Service stability			Proactive threat &vulnerability management	Transparency
Provider business stability		Serviceability			Retention/disposition	Understandability
Provider certifications					Security management	
Provider contract/SLA verification						
Provider ethicality						
Provider personnel requirements						
Provider supply chain						
Provider support						
Sustainability						

Figure 41 SMI - CSMIC Framework, v2.1 from 07/2014 (retrieved from CSMIC website[5])

Current taxonomy follows SMI structure adding standardization attributes and leaving provider accountability in grey while there is no connection from business level and following EA requirements.

Figure 42 QoS taxonomy high-level view

The rest of SMI's QoS attributes taxonomy is extended with up-to date attributes derived from literature [54]–[57], [61], [63], [65], [67], as well as more service-specific ones e.g. detailed IaaS' QoS attributes [17], [54] or security-specific QoS attributes [114]–[121]. One classification that is necessary to mention is the FURPS+ System for Classifying Requirements from Hewlett-Packard [106]. It contains nearly the same categories in classification, but the attributes are very application-specific e.g. printing capability in Performance category. Some of them that can be generalized to most of cloud services were used in current taxonomy. A good example of that is Aesthetics attribute – an attribute missing in all other examined sources – which is important for many users which deal with a service via an interface.

The full taxonomy can be viewed here: http://gocloud.ga/DSM.html#fourth.

As usual, further sub chapters describe the taxonomy per group starting from Agility and ending with Standardization. Provider accountability (trustworthiness) is also described in the last sub chapter of this taxonomy, but it is not included in DSM due to the reasons specified further.

3.4.2. Agility

Agility is considered the most important advantage of cloud computing, identified in science and industry, while it adds more to the agility of an organization than traditional solutions. It means that the company receives the ability to respond quickly to changing capacity requirements. Resources, for instance, can be allocated and de-allocated as required, whereas requirements can sometimes vary greatly [17]. In the current taxonomy, three metric were added from ISO 9126-1 [113] (they are highlighted in green), and the rest were grouped in categories.

[9] The Service Measurement Index official webpage: http://csmic.org/

Figure 43 QoS taxonomy: Agility

3.4.3. Assurance

Assurance - this characteristic indicates the likelihood of a cloud service performing as expected or promised in the SLA [59]. Again, the metrics were grouped according to ISO standard [113]. Also, an example of metrics was included (see Availability in Figure 44).

Availability is usually measured by uptime and downtime in SLA agreement. For example, SLA level of 99.9 % uptime/availability gives following periods of potential downtime/unavailability:

- Daily: 1m 26.4s
- Weekly: 10m 4.8s
- Monthly: 43m 49.7s
- Yearly: 8h 45m 57.0s

Such SLA calculations[10] assume a requirement of continuous uptime (i.e. 24/7 all year long) with following additional approximations: 365.243 days per year; 52.178 weeks per year; 30.437 days per month; and 4.348 weeks per month.

Figure 44 QoS taxonomy: Assurance

3.4.4. Financial

Financial reasonability is without doubts the first - the first question that arises in the mind of organizations when switching to a cloud service. Looking at whether it is cost-effective or not tends to be the single most quantifiable metric today. While the different metrics exist along with different payment methods, schemes etc., most common in the literature financial Quality of Cloud Service attributes are presented in the QoS taxonomy. And of course, those should be derived from business level and EA perspective consequently.

Figure 45 QoS taxonomy: Financial

[10] SLA levels calculator available at: http://uptime.is/

3.4.5. Performance

One the one hand, there are many different solutions offered by cloud providers addressing the IT needs an organization, and each cloud solution has different performance in terms of functionality, service response time and accuracy one the other [59]. If a company wishes to deploy its own application in a cloud or a set of them, it needs to know whether this new environment will provide wished performance. Same as in previous subchapters, one attribute was derived from ISO 9126-1 [113], and an example metric was shown (IaaS platforms compatibility). Extensive interoperability QAs based on [62], [122]–[124] were added to enrich the QoS attributes taxonomy with more specific metrics. These are necessary to take into account when comparing solutions from different vendors that look alike at first glance. Functionality efficiency-specific attributes marked as brown were taken from FURPS+ classification [106].

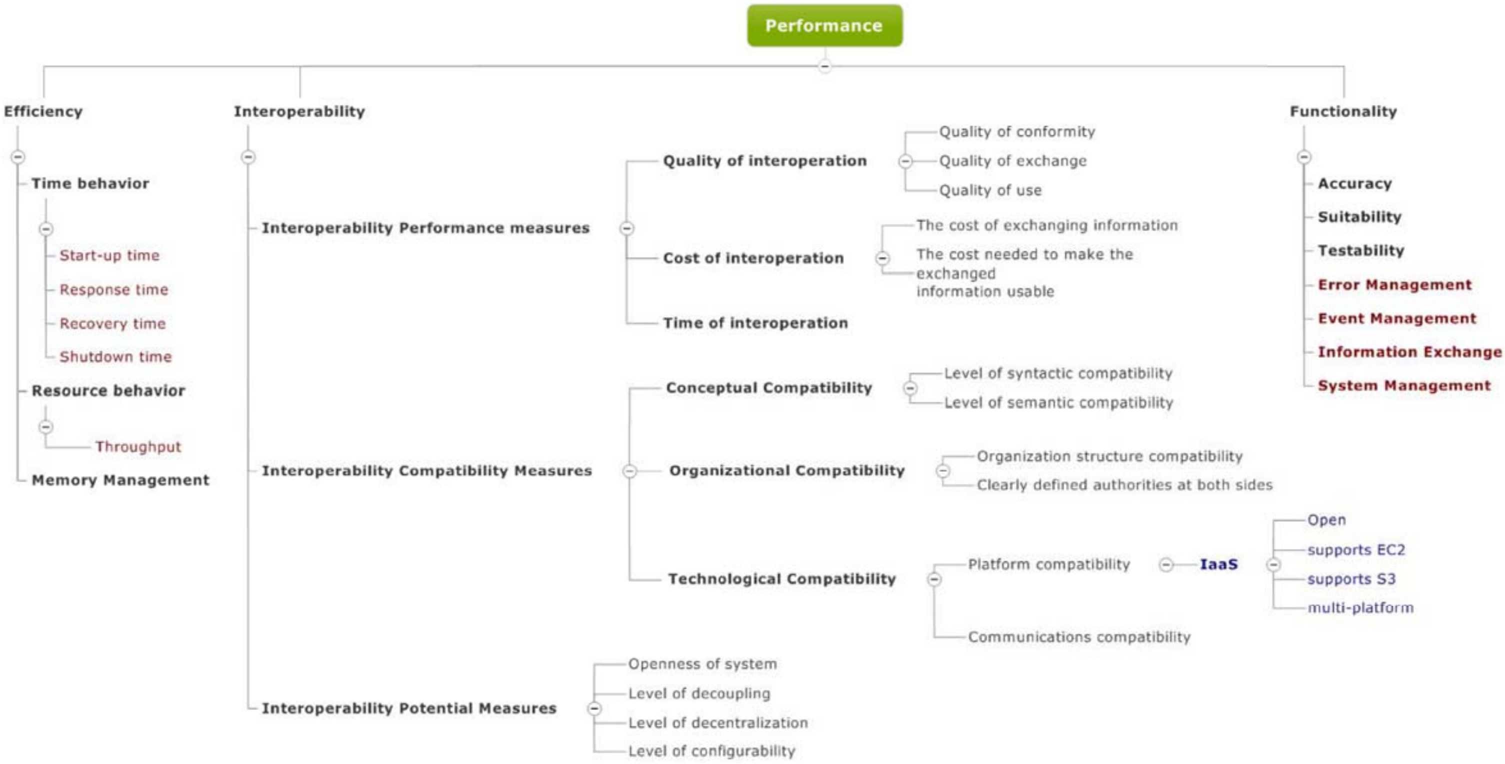


Figure 46 QoS taxonomy: Performance

3.4.6. Security and Privacy

For most of business as well as scientific cloud services, security is of great importance due to the nature of data stored in the cloud. These data might be private, vital for competitiveness or even of national safety importance. All that implies that security requirements must be set as clear as possible in order to choose the right tactics to fulfil them. To do that, current state of research was consulted to reveal the variety of security threats, attributes of security QoS. By a security attribute we mean the way by which unauthorized and intentional harm may be caused to valuable system assets. Security-specific QoS attributes were derived from [114]–[121] and presented in a QoS taxonomy branch below:

Figure 47 QoS taxonomy: Security and Privacy

3.4.7. Usability

When switching to a new service, it is vital for users to adopt quickly so the company will start to benefit from its adoption as fast as it can. That implies that higher the degree of usability, faster the ROI is. The usability of a Cloud service can depend on multiple factors. Five of them presented in grey in Figure 48 were derived from SMI, four in green – from ISO 9126-1 [113], and tree in brown – from FURPS+ [106]. Usability branch of QoS attributes taxonomy is presented in Figure 48 below:

Figure 48 QoS taxonomy: Usability

3.4.8. Standardization

One way to avoid a vendor lock-in is not to stick to vendor-specific technologies, but choose a standardized offering. This way it is possible to flexibly switch a provider without losing the advantages of cloud service. Moreover, for most enterprises interoperability among other applications is vital, and if the exchange mechanisms and interfaces are unique – that might bring some problems. To address these issues and to facilitate interoperability, standardization efforts have to take place in order to support further developments in the clouds [124].

Transparency and auditability are important security requirements for the cloud used being able to verify that the cloud provider is adhering to the standards, processes, and policies he follows.

Ease-of-use and standardization caused web services to be used in ever-increasing numbers and critical functions than ever before. On today's scale, security mechanisms used for protecting data and BP flow are vital and thus need to be audited on a regular basis. Security auditing is about enabling the security personnel to audit the status and use of security mechanisms by analyzing security- related events. This is also often done for the purpose of achieving compliance to laws and regulations or accountability and control. Specific security protocols used in different services and possible certifications are well described in literature base, e.g. security auditing procedures for IaaS [125], PaaS [126], and SaaS [127]. This branch may be customized by the user to fit the case e.g. economy sector or country regulations. Most common attributes were put in the branch by the Author with a few examples in blue:

Figure 49 QoS taxonomy: Standardization

3.4.9. Provider Accountability

No organization will want to deploy its applications and store their critical data in a place where there is no accountability of security exposures and compliance [54].

Provider accountability metrics are supposed to measure how well a provider ensures that a service is up and available for use according to the SLA. The reliability of whether provider's commitments such as guaranteed availability are kept is very important for a cloud user. Be it a sound certificate, vendor's reputation, or infrastructural features, such attributes may be the evidence of a high reliability [17].

Still, while being utmost important, provider accountability metrics are not included in DSM while they are not connected to the company's specific BP and EA attributes. However, they are presented here to show the importance of accounting for them when choosing a cloud vendor of a cloud service.

Figure 50 QoS taxonomy: Provider Accountability

To sum up, the fourth taxonomy which is about Quality of Service attributes has high-level structure of Service Measurement Index (SMI)[11], and the attributes in it are carefully gathered from corresponding literature sources: [17], [61], [65], [67], [56], [57], [63], [55], [54][17], [54] [106], [114]–[121]. This way, the taxonomy is effective for choosing a ***cloud*** service and perfectly fits the DSM. The QoS attributes brake into following categories:

- Agility attributes,
- Assurance attributes,
- Financial attributes,
- Performance attributes,
- Security and privacy attributes,
- Usability attributes, and
- Standardization attributes.

3.5. Architectural requirements taxonomy

This is the last taxonomy which also means that the level of granularity is the lowest. It concerns specific implementation tactics of a potential cloud service e.g. license type or payment options. This step is important when quality requirements are already set and the user can see which service features are available or not due to the preselected QAs.

3.5.1. Literature base

Several frameworks and a great amount of papers are focused on comparing cloud services functionality, QoS measurement and vendor offerings. All these metrics are possible because of the architectural differences that cloud solutions have between each other. Whether to choose one security protocol or another depends on the requirements set in previous set of our taxonomy, and this step is finally focused on specific implementation tactics.

High level grouping of architectural requirements is based on several works in this area [57], [93], [94], [92], [52] and is shown in Figure 51 below:

[11] The Service Measurement Index official webpage: http://csmic.org/

Figure 51 Architecture requirements taxonomy high-level view

Full taxonomy can be viewed here: http://gocloud.ga/DSM.html#fifth.

This taxonomy does not pretend to encompass all possible cloud service features or architectural options and not even most of them – it is not feasible. This taxonomy contains most important ones that depend on previous QA requirements, so an Analyst can see how those QAs and QoS requirements affect service implementation. Of course, there are more options to specify for a cloud solution, but this phase is more technical and is a next step in cloud (and vendor) selection process which is out of scope of this thesis. The following subchapters present a fine overview of most common architectural requirements found in the literature.

3.5.2. Management

Cloud management assumes all means by which data, services and applications residing in the cloud are accessed, monitored, controlled etc. Cloud management tools help ensure the resources are working optimally and properly interact with users and other services. Which tools are needed/used is heavily dependent on the type of cloud service, e.g. IaaS may only have a command-line interface, while an extensive BPaaS solution requires an elaborate dashboard and sophisticated management capabilities. [27], [45] formulate two approaches on management layer formalizing possible architectural features, which are jointly used in current taxonomy shown in Figure 52 and Figure 53 below:

Figure 52 Management features taxonomy

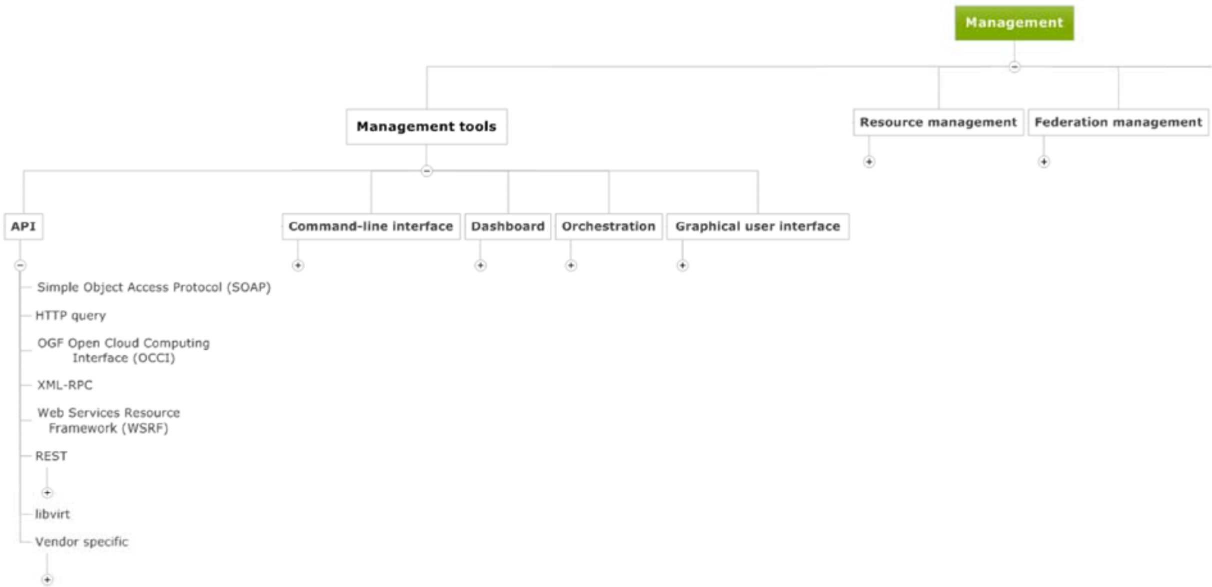


Figure 53 Extended view on management tools

Again, not all management tools are possible within some delivery models and service types. Chapter 0 deals with interconnection of those features and cloud services taxonomy.

3.5.3. Running time

Runtime tuning refers to the capabilities of a cloud environment to dynamically adjust the size of the provisioned resources to one application according to the user load, and distribute the load so that the necessary QoS parameters are fulfilled [52]. Corresponding taxonomy branch splits this category into several elements as proposed by [52].

Figure 54 Runtime tuning branch of the taxonomy

A broad overview of current elasticity (resizing) means is done by [128]. It is included in the taxonomy, and some of the examples of resizing options are presented in the Table 4 below.

System	Scope	Policy	Method	Purpose
Amazon WS	infrastructure	automatic reactive	replication	performance
Amazon WS – Spot-Instances	infrastructure	automatic reactive	replication	cost
CHP	infrastructure	automatic reactive	replication	increase infrastructure capacity
PRESS	infrastructure	automatic predictive	resizing	performance
Elastic Site	infrastructure	automatic reactive	replication	increase infrastructure capacity
Kingfisher	infrastructure	automatic predictive	replication migration	performance cost
CloudScale	infrastructure	automatic predictive	resizing	performance energy
PaaS Clouds	platform	automatic reactive	replication	performance
Aneka	platform	automatic reactive	replication	increase infrastructure capacity
Azure	platform	manual	replication	performance
Knauth & Fetzer	application	automatic reactive	migration	performance
Elastin	application	manual	resizing	performance
Zhang et al.	application	manual	replication migration	performance energy

Table 4 Examples of elasticity solutions derived from [128]

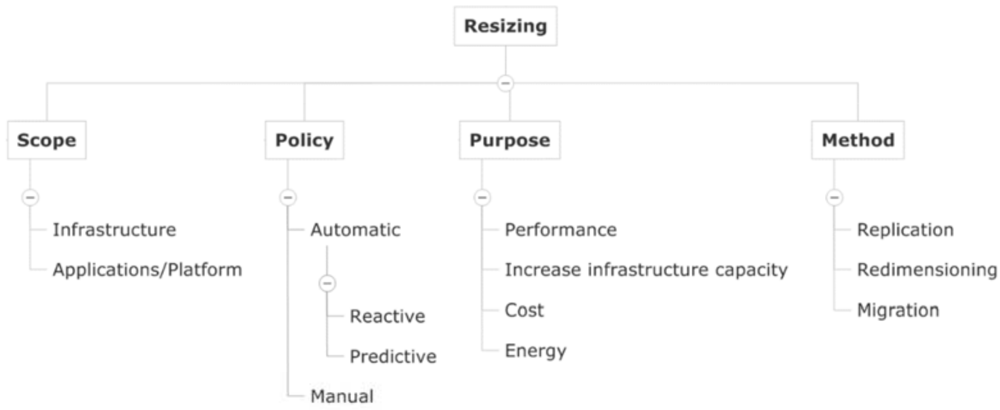


Figure 55 Cloud service resizing options taxonomy

Checkpointing (also called snapshot or cloning) is a feature offered by very few cloud providers and refers to the capability of saving a snapshot of the running VDS (including all applications, data, configuration files, etc.) at any time instance [52]. The benefit is an enhanced fault tolerance by restarting a failed VDS from the checkpoint, or by replicating an existing one to another server to prevent failures through load balancing. Yi et al. [129] observe that checkpointing can significantly affect both the task completion time and the total monetary cost such as that using hourly checkpointing can reduce costs significantly in the presence of failures, and that simple policies (hour and rising edge) work well in very few instance types. Full comparison with vendor examples and tests on them can be found in [129].

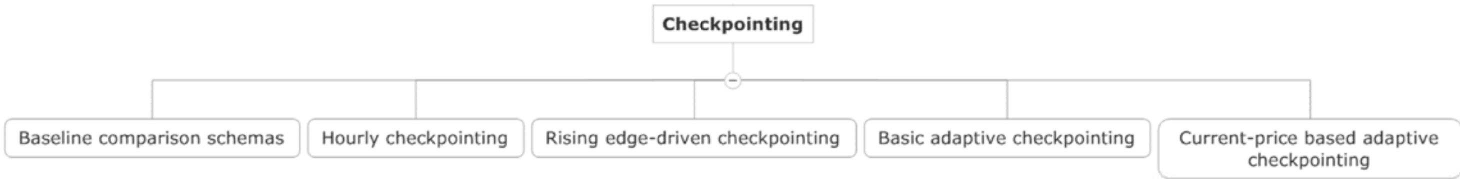


Figure 56 Cloud service checkpointing taxonomy

Existing load balancing techniques are derived from [130]. Examples of possible techniques used are also shown here for the sake of understandability:

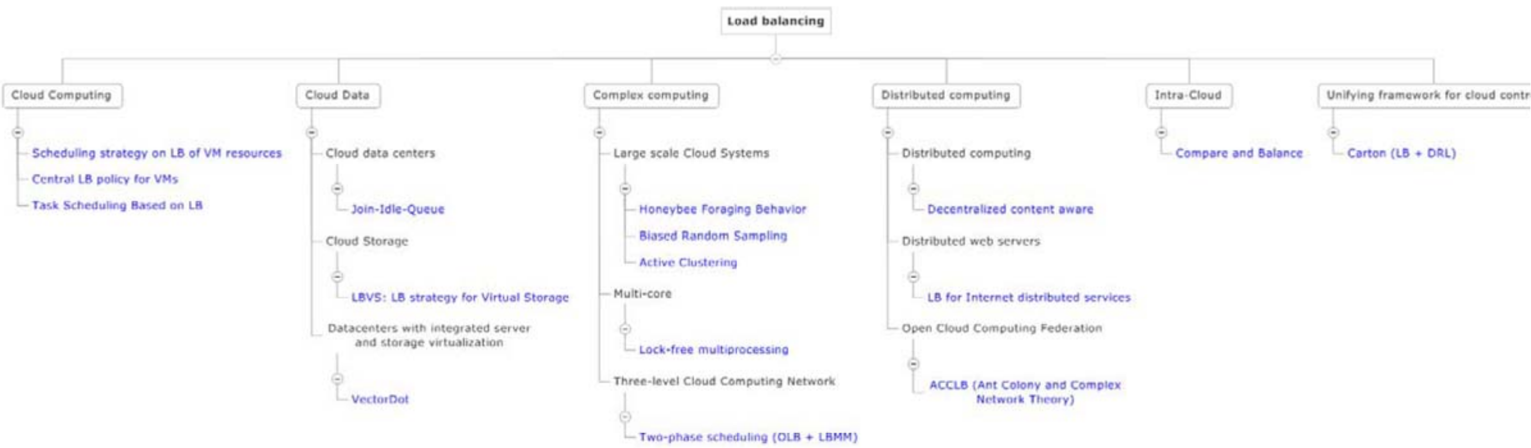


Figure 57 Cloud service load balancing taxonomy

3.5.4. Access

Cloud services are increasingly getting accessed through browsers and thin mobile devices running new set of applications like HTML 5 [131]. Multi-touch surface devices are now gaining popularity as devices for collaborative use. Enterprise wide initiatives such as Bring Your Own Device or Mobility as a Service imply cloud services to be accessed from a variety of thin clients.

It is also vital to see whether a service it able to support special devices like those included in taxonomy. Examples of such devices are: TouchShare[12] for geospatial collaboration solutions; Multi-Touch Collaboration Walls[13]; DepthTouch which makes use of a depth-sensing camera, a projector, desktop computer, and a vertical screen for the user to interact with [132]; Sprout[14] by HP where content can be digitally captured and manipulated in 2D or 3D directly on the TouchMat interface; variety of 3D collaborative devices with infrared or thermographic cameras used to facilitate gestural detection. All those are the instruments used more and more due to emerging technologies, and thus should be considered when thinking of cloud service device supportability. One of the logical grouping is proposed by the Author and is used for the taxonomy:

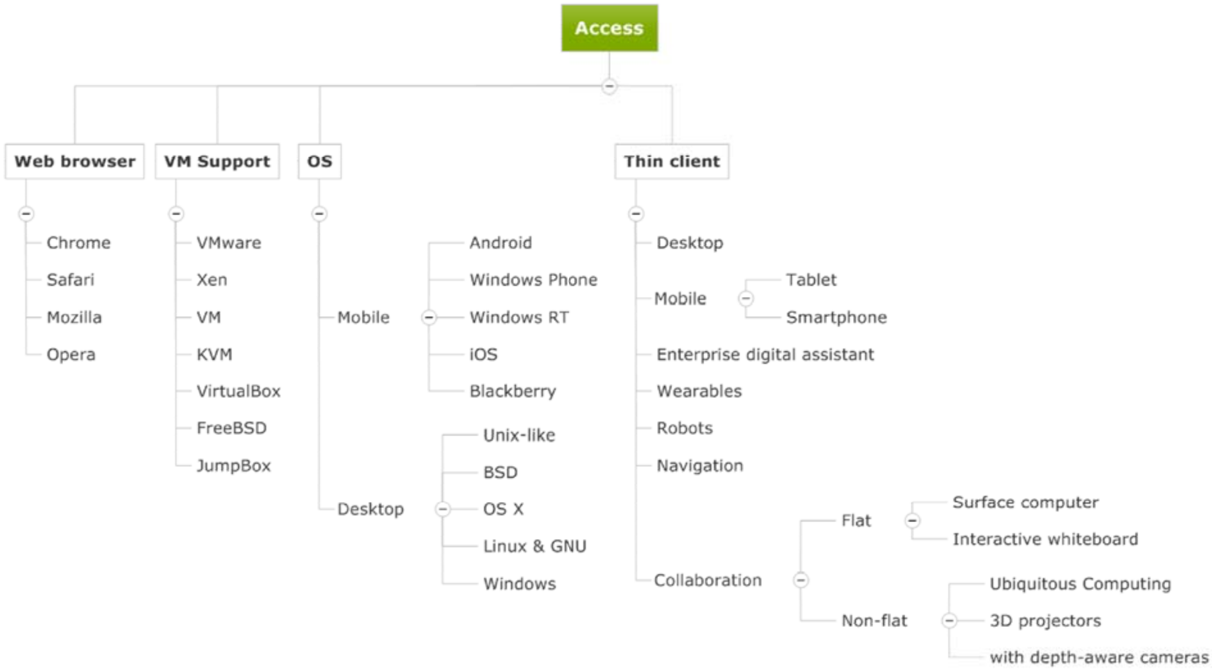


Figure 58 Taxonomy of the means of access to a cloud service

3.5.5. Security taxonomy

To show how different can the views on security taxonomies be, most relevant are listed below as examples to that. Security branch of proposed taxonomy is developed mainly based on those.

Abuhussein et al. [115] have proposed a security attributes taxonomy which consists of several levels: level-0 includes different types of stakeholders, level-1 comprises cloud computing deployment methods, level-2 includes different types of service models, level-3 defines corresponding issues and level-4 determines the attributes. With the help of this taxonomy, the consumers can compare different cloud providers based on security issues.

Prodan et al. [52] proposed a taxonomy that identifies a common terminology and architectural and functional similarities among the cloud providers. It consists of eight elements: service type, resource deployment, hardware, runtime tuning, security, business model, middleware and performance.

Gonzalez et al. [117] have identified the major cloud security issues and grouped them into seven categories: Network Security, Interfaces, Data security, Virtualization, Governance, Compliance and Legal issues. The authors have taken into consideration of all efforts made by the European Union Agency for

[12] http://touchshare.com/

[13] http://ideum.com/touch-walls/, http://multitouch.com/multi-touch-wall.html, http://www.microsoft.com/microsoft-surface-hub/en-us

[14] http://sprout.hp.com/us/en/

Network and Information Security (ENISA), CSA and NIST. Thus, the proposed taxonomy helps to identify how much research efforts have already been made in those dimensions and which area needs to be focused on.

Gruschka et al. [118] proposed a taxonomy of cloud computing attacks and classified them into three entities: service users, service instances and cloud providers. The attacks are classified based on which entity is being attacked. The six categories of attacks among the three entities are: service-to-user, user-to-service, cloud-to-service, service-to-cloud, cloud-to-user and user-to-cloud. So, with the help of this taxonomy, any entity can be aware of the risks and security threats from other entities.

Habib et al. [119] have proposed a framework for assessing the trust-worthiness of the cloud providers' claim of satisfying the controls of Consensus Assessments Initiative Questionnaire (CAIQ). They have mapped the controls of CAIQ into trusted properties and stated who is going to validate the properties.

Mainly based on previously listed security taxonomies [52], [114]–[121], possible security mechanisms used to access a cloud service are listed in Figure 59:

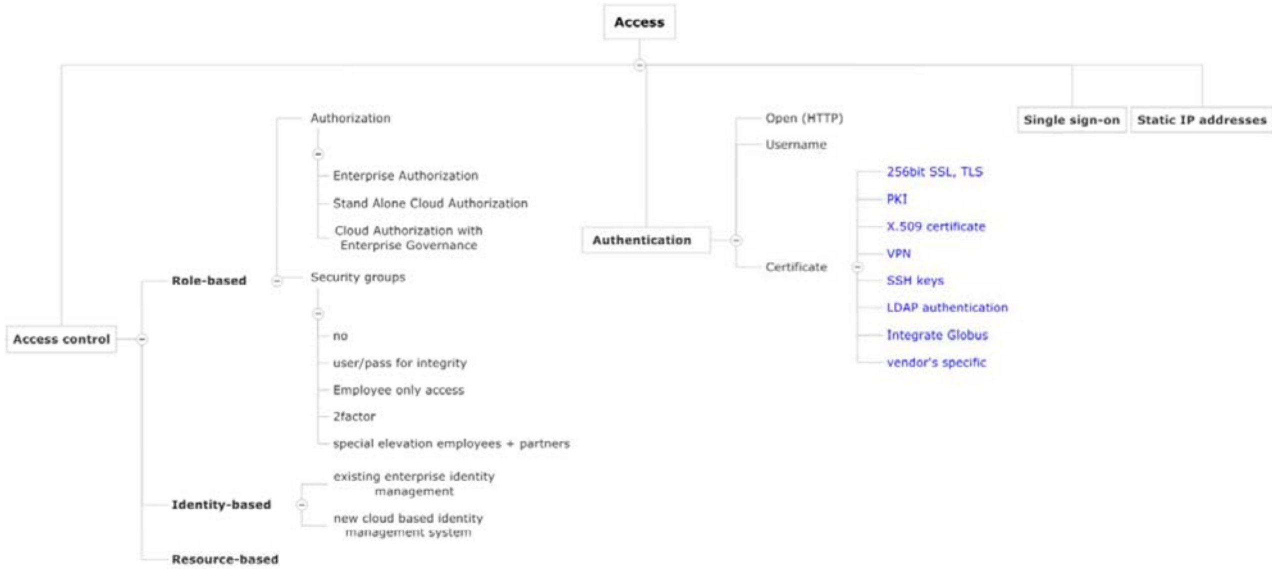


Figure 59 Access branch of security taxonomy

Extensive overview of available technologies used against security threats and risks is presented in [121] and is shown in Figure 60. Browsers of mobile devices do not have direct means of handling XML signatures and XML encryption, and rely on the underlying SSL layer for handshake. Hence this channel may become a potential threat if not secured properly. This may push enterprises to use VPNs while communicating to the cloud.

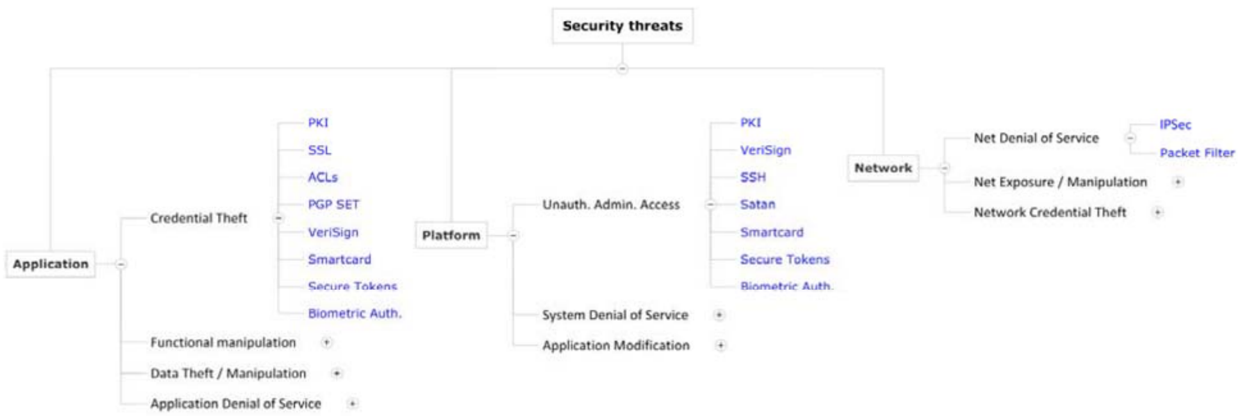


Figure 60 Security threats branch of security taxonomy

Rest of the security mechanisms mentioned in most of literature sources [52], [114], [116], [120], [121] are also included in security branch of the taxonomy.

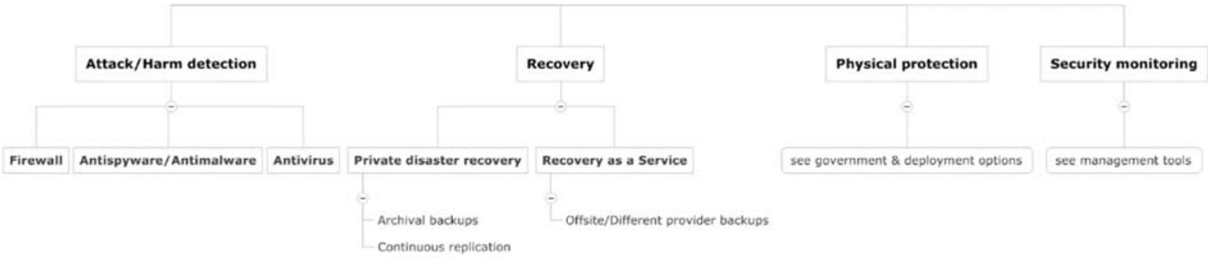


Figure 61 Rest of the branches of security taxonomy

3.5.6. License type taxonomy

Whether to choose open source solution or not, is heavily investigated by literature base [6], [10], [12]–[14], [17]–[20], [25], [26], [32], [36], [42], [44]–[46], [51], [52], [65], [94], [133], [134]. Types of a license provided by a provider vary much from one provider to another, so it is not possible to scope them in the model with a decent level of formalization. Therefore, licensing is very important architectural aspect when implementing a cloud service, but remains a rather small branch of the taxonomy.

Figure 62 License types

3.5.7. Formal agreements taxonomy

Service Level Agreements (SLA) is usually done by mapping between business processes and IT infrastructure. Prodan and Ostermann [52] present a survey where more than half of the surveyed IaaS providers do not explicitly declare any kind of support for SLA. Those who do claim to offer SLA, limit their QoS terms to server uptime with a confidence ranging between 99.95% to 100%, while some providers market even 100% SLA guarantee without giving any concrete details [52].

Figure 63 Basic taxonomy of formal agreements

There are several works in the current body of literature on the negotiation and the enforcement of SLAs and therefore focus on their formalization. E.g. [68], [135] present their schemes which can be used to describe further aspects of SLAs, e.g., design-time related non-functional properties (both technical and human, as well as their combinations), such as cultural (e.g. language), legal (e.g. Sarbanes-Oxley-Compliance, Basel-II-Compliance), organizational (partner list), service usage-related (e.g. GUI simplicity) and trust-related (e.g. Customer rating, experience of provider).

3.5.8. Payment system taxonomy

Prodan and Ostermann [52] present a broad perspective on different payment options available from current vendors. As it is the most extensive and formal found in the literature study, it is used as a base of payment options in the model.

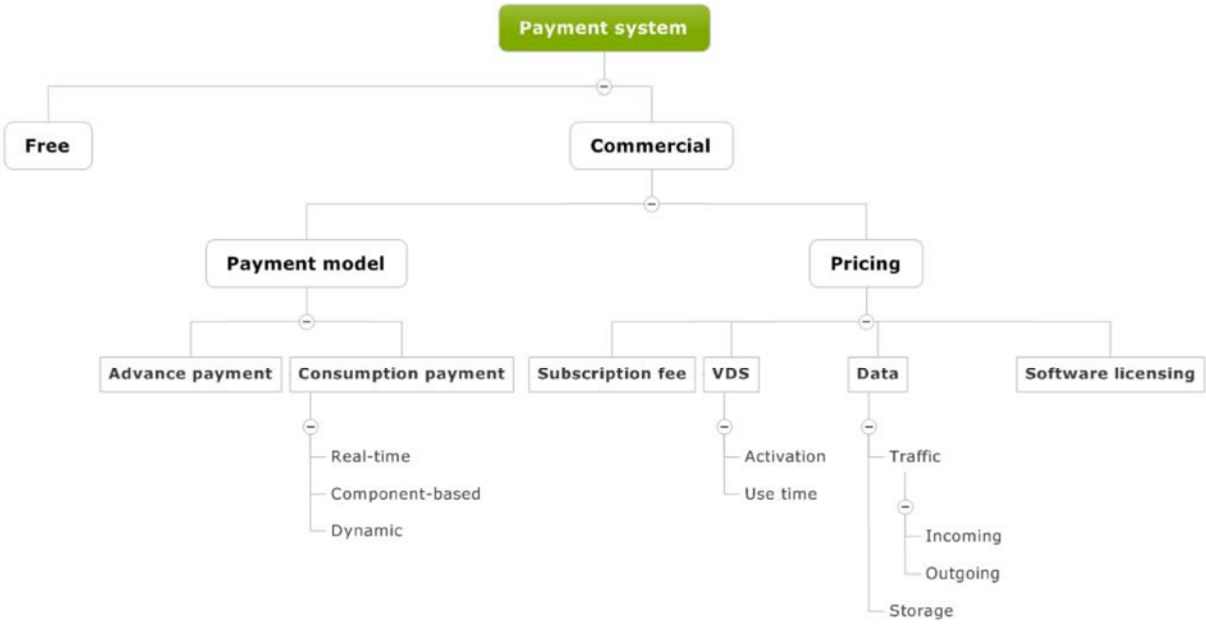


Figure 64 Payment options taxonomy (adapted from [52])

3.5.9. Governance taxonomy

Geographical and political issues are sometimes the key architectural requirements a cloud service. When data begins to move out of the organizations, it is vulnerable to disclosure, or loss. The act of moving sensitive data outside the organizational boundary may violate national regulations for privacy [15]. For instance, due to the German data protection law called "Informationelle Selbstbestimmung" passing data across German territory can be a federal offense. Chapter 4 is dealing with such situations, where e.g. mission-critical process requires a cloud service while the data involved poses strict governance requirements. Avoiding "data lock-in" is also considered as a data governance issue.

Aside from various data privacy and juridical issues, company's internal cultural environment and political situation where cloud provider operates must be taken into account.

Internal politics: internal developers that do not want to give up their software; users resisting cloud computing implementation; changes in internal structure of IT departments; etc.

External, sometimes international politics: data filtering and throttling by telecommunications companies; Internet access availability, scalability and information flow issues; some countries' laws preventing global cloud development; differences in laws between countries; low predictability of law changes tied to political situation; etc. Big cloud vendors have already developed local infrastructures that permit users to choose a desired jurisdiction.

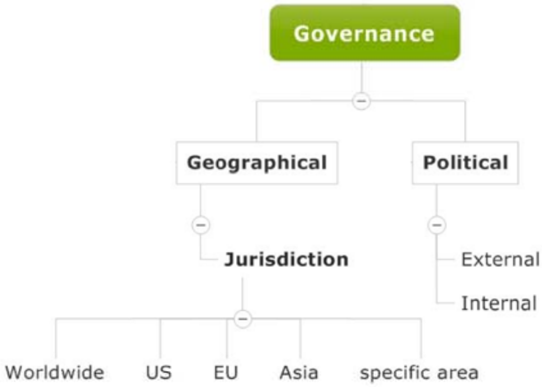


Figure 65 Governance taxonomy

3.5.10. Deployment taxonomy

Common used terms for cloud service deployment model include private/public/hybrid options [53]. To extend those with more sophisticated options, this trinity model is extended by literature review to include more terms and community cloud model [15], [25], [31], [34], [46]:

Figure 66 Common used terms for cloud deployment models

Some authors propose more specific deployment models (e.g. company internal cloud [9]), thus emerges the need to formalize the notions. It was decided to use more formal notions giving the taxonomy a possibility to include both abundant and rare deployment models. More formal model shown in Figure 67 on the right is used in the taxonomy.

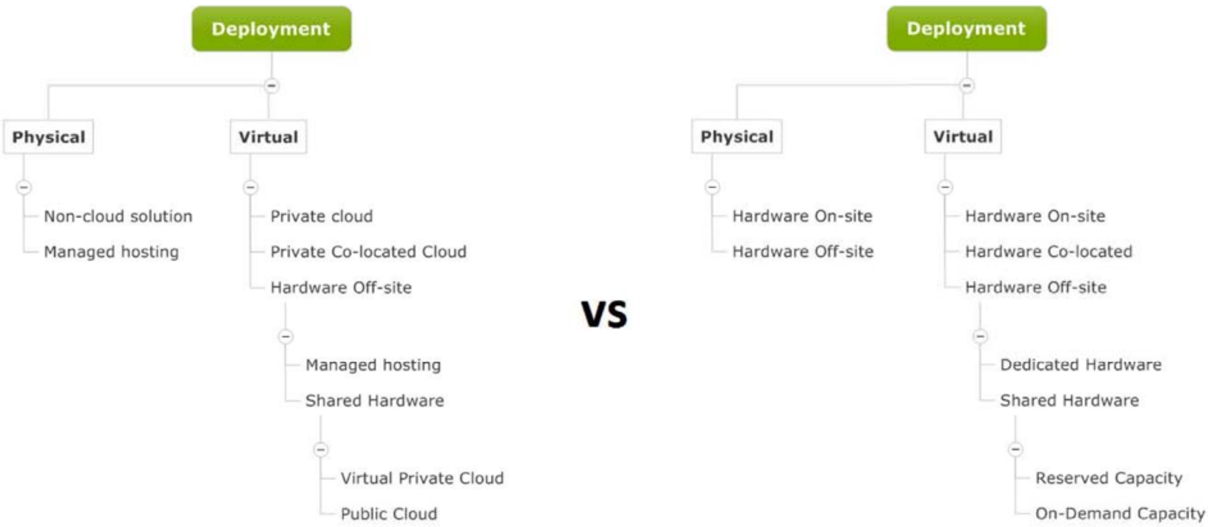


Figure 67 Informal vs formal deployment models notations

3.5.11. New technology – Intercloud

The standard cloud deployment model when utilizing a single cloud results in less reliability then when using multiple ones and less responsiveness and usability to clients distributed worldwide. To achieve better QoS, reliability and flexibility different cloud providers connect their resources into a federation of multiple clouds (i.e. an Inter-Cloud or Multi-Cloud) [78]. Thus an Intercloud provisions on-demand multi-provider multi-domain heterogeneous cloud infrastructure services. Such a service provides Better application resilience while cloud providers can expand on demand with more ease, diverse geographical data locations for multinational customers, and of course avoidance of vendor lock-in with better SLAs at the same time [78].

State of the Art in described in [73]–[78], while [75], [78] contain up to date lists of Intercloud providers. Based on those literature sources, Intercloud taxonomy is presented in Figure 68. As it can be expected, Intercloud QoS very from ordinary cloud service's ones. E.g. IEEE P2302 Working Group (Intercloud) is now developing the Standard for Intercloud Interoperability and Federation (SIIF). This standard defines topology, functions, and governance for cloud-to-cloud interoperability and federation. Such specific architectural requirements are addressed in Chapter 4.

Speaking about 'inter cloud' hand-offs and federated identity management (C3), it can possibly be done through assertion tokens like Security Assertion Markup Language (SAML) or privilege management infrastructure based on x.509 certificates [131].

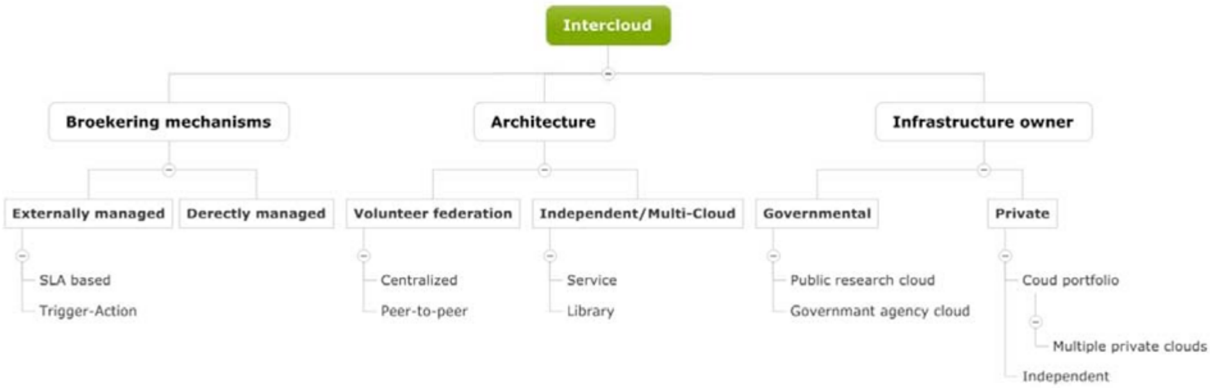


Figure 68 Intercloud taxonomy

Figure 69 Intercloud-specific QoS

This was the last but not least taxonomy. It was focused on main architectural options available when choosing a cloud service.

To sum up all five taxonomies described in this chapter, we can state that:

- Functional requirements, the first step in DSM, is a general taxonomy that groups most of current cloud services into one taxonomy. It can be used to select a type of cloud solution needed. The taxonomy itself is mainly based on highly recognized sources (such as NIST), and partially constructed by the Author (for new types of solutions which lack the literature base).
- Business process attributes taxonomy describes the process (or a set of them), that shall be supported by a cloud service, both from descriptive perspective and from enactment perspective. This taxonomy is critical in spotting high level business requirements, which is utmost important for a well fit cloud solution architecture.

- Enterprise Architecture attributes is a lower level of granularity after BP attributes. This taxonomy deals with IT from the company-wide perspective. By selecting EA QAs, an Analyst ensures that all the systems within the company are well aligned with a new cloud service, all interfaces are compatible, data is accessible etc.
- Quality of Service attributes form the fourth taxonomy. While this step is often used in the cloud section process, this taxonomy gathers most of the important attributes for cloud services from respected literature sources. It is not surprising that most of the QoS attributes presented are the same as in other similar (not only cloud) taxonomies.
- Architectural requirements taxonomy plays a prominent role in service selection. However, it is not feasible to list all possible implementation tactics. Therefore, the most important and common ones were grouped to form the taxonomy which can be further elaborated in a specific business case with a specific client needs.

The next chapter deals with grouping all these five taxonomies into one model, also governing interaction between these taxonomies.

4. Decision Support Model

Five steps of Decision Support Model described in the previous chapter are grouped together to form one model. All five steps combined in a DSM can be viewed here: http://gocloud.ga/DSM.html#DSM.

All leaf nodes of each taxonomy have values in the model. They can be either of a sequence type (e.g. from low to high) or of a set type (e.g. VMware, Xen, FreeBSD etc.). Those values are do not pretend to be unambiguous, rather the opposite of it – subjective to a user. It is not possible to define all possible values for an attribute, but if a user selects a value “High”, he can specify this value afterwards in RFP. To provide an example, an Analyst operates with his own logic when using the DSS tool and after the list of requirements is generated, he can elaborate these requirements, e.g. instead of “High” availability put “99.99 uptime”. This provides the necessary flexibility to the model, to the contrast with tools where a user is limited in choices and often lacks something. The Author believes that this method of using “subjective” values and their clarification and interpretation afterwards is more suitable for a Decision *Support* tool, which is not a Decision *Making* tool.

Also, it is possible to modify the DSM to a certain extent to tailor it to an industry sector, country, or juridical regulations. This can be done in a DSS tool interface. However, the DSM model as a methodology defines what cannot be changed in the model, what are the pillars that should not change. These are all the levels of all taxonomies based on the literature review. All specific instances of cloud solutions or end nodes can be changes, added or deleted. Knowing that there are more low-level attributes that can ever been put in these five taxonomies, the DSS tool allows the user to perform this customization to fit his needs.

Using the DSS tool, a user goes through all five DSM steps gathering more and more specific requirements. To make his task easier, there are certain relationships between the steps – rules of the model. They govern the restrictions to the following steps posed by selecting requirements on the previous steps. E.g. is one specifies that a Business Process operates with top-secret data on the second step of the model (the attribute of BP data privacy is “High”), then in the fourth step of setting QoS attributes a user cannot select “Open (HTTP)” value for authentication attribute.

This sub chapter describes the model structure, its formal description (syntax), and the relationships within (rules between nodes).

4.1. Model description

Each step being one of five previously described taxonomies is represented as a tree, abstract data type with a hierarchical structure. Each of five trees corresponding to respective step consist of nodes (internal nodes or inner nodes or branch nodes are nodes of a tree that have child nodes) and leaves (leaf nodes or external nodes or outer nodes or terminal nodes are nodes of a tree that does not have child nodes). Every node has children, only leaves do not have any. The topmost node in a tree is called the root (the root node).

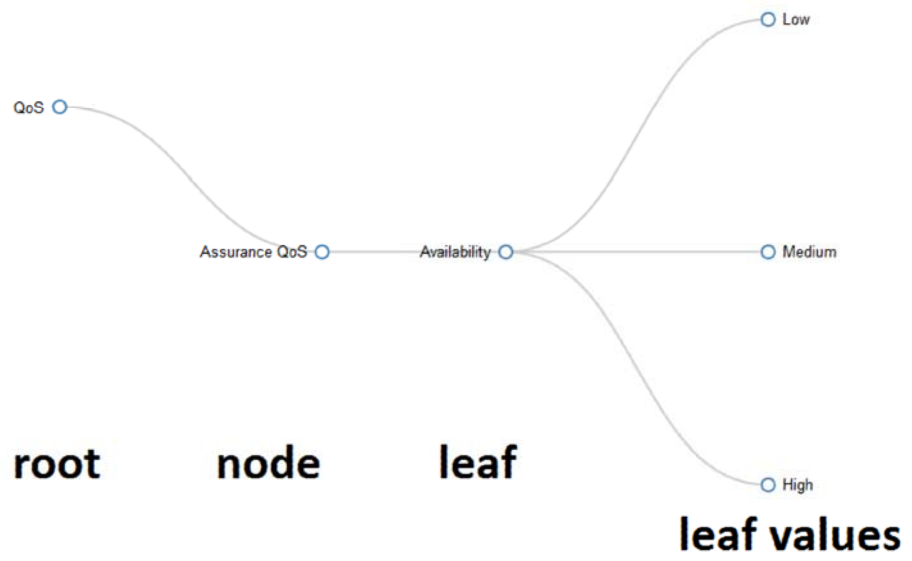


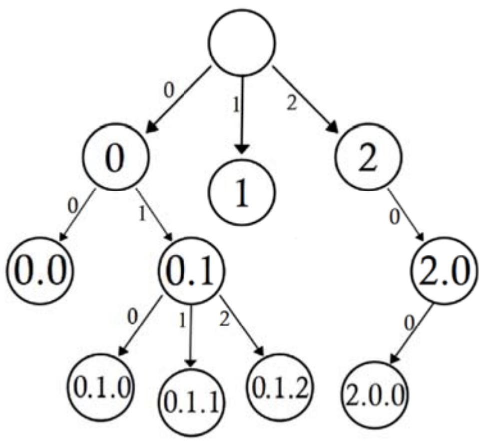
Figure 70 Tree structure representing taxonomies

For the DSM presented we set the following properties:

- Nodes have attributes: unique id, name, description.
- Every node has one or more children.
- Leaves have attributes: unique id, name, description, type, value.
- Every leaf has one or more values.
- Leaf type can be: sequence or set.
- Sequence type stands for a number of values that are ranked among each other, e.g. “Low”, “Med”, “High”.
- Set type stands for a set of values that are not related to each other, e.g. “Oracle DB”, “MySQL”, “sqlite3”.

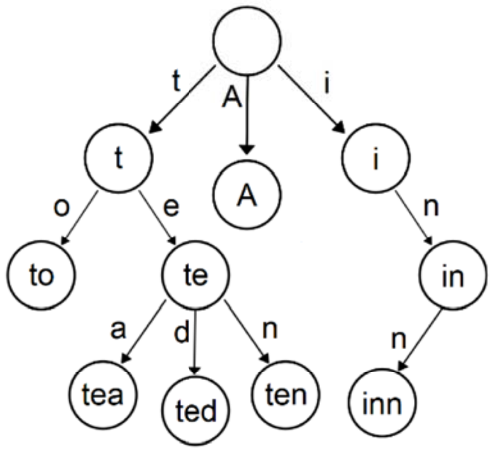
To be able to address each node without mistaking it with other ones, unique ids are used as a usual practice in IS. Unique node ids for this kind of tree structure can be generated using hierarchical ids or trie structure:

1) Empty string means the root, 0.1.2 stands for the third child of the second child of the first child of the root:



To shorten these ids for ease of use, they can be coded with Base 36 to avoid dots and two-digit numbers. E.g. 0.1.1.1 would be 0111, and 0.10.11.12 would be 0ABC with same four characters.

2) Trie (digital tree or radix tree or prefix tree) is an ordered tree data structure where the keys are strings as in the following example (0.1.2 is now “ten”):



To give these kind of ids a consistent length, they can be hashed using MD5 algorithm:

- t = e358efa489f58062f10dd7316b65649e,
- te = 569ef72642be0fadd711d6a468d68ee1,
- ten = b1b9a972ccd8c962a473909b97007eb4.

As for the current state of DSS tool development, hierarchical ids with Base 36 encoding are used while they are simply short and understandable. However, on the later stages trie ids with MD5 can be used if a need to store large amount of nodes/leaves will arise.

Leaf value rules are the constraints on leaf values or on the (sub)set of leaf values. A rule is described as follows:

If {set of statements} then {set of statements}, where a statement can be

- a value equals/not equals a value for sets,
- more or equal/less or equal for sequence types.

Second part of a leaf rule can only include tree(s) with number not less then the number of tree in the first part of the condition. E.g. rule between 1st and 2nd tree leaves is valid, while a rule where a 2nd tree leaf's value governs 1st tree leaf's value is not valid.

To be able to easily modify tree nodes or rules between them for specific business case purposes, a formal syntax for the DSM was formulated. It is presented in the next subsection.

4.2. Model syntax

Since it was decided to use ECMAScript language (JavaScript) for DSM prototype development, it is logical to use JavaScript Object Notation (JSON) for data objects (trees). It is considered to be a good choice for an extendable and reusable data for DSM since it is 1) an open standard format, 2) it uses human-readable text easy to understand, and 3) it is a language-independent data format and can be easily parsed in many programming languages.

To define relations between nodes that actually transfer requirements from one step of the model to another and related to that constraints, JSON notation is extended with "rules". Those rules syntax is derived from JavaScript logical conditions syntax. Syntax of the full model is described further. Extended Backus–Naur Form (EBNF) will be used to make a formal description of a formal language which is used in the DSM. EBNF is a metasyntax notation (also known as ISO/IEC 14977). Comments following the EBNF descriptions are to guide the reader into better understanding of the syntax.

<u>MetaSyntax:</u>

[X] 0 or 1 Occurrences of X

{ X } No or any Number of X

X | Y   X or Y

“x”   x is character in source

(comment *)*

<u>Syntax:</u>

letter = "A" | "B" | "C" | "D" | "E" | "F" | "G” | "H" | "I" | "J" | "K" | "L" | "M" | "N" | "O" | "P" | "Q" | "R" | "S" | "T" | "U" | "V" | "W" | "X" | "Y" | "Z" | "a" | "b" | "c" | "d" | "e" | "f" | "g” | "h" | "i" | "j" | "k" | "l" | "m" | "n" | "o" | "p" | "q" | "r" | "s" | "t" | "u" | "v" | "w" | "x" | "y" | "z" ;

digit = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" ;

symbol = “ ” | "(" | ")" | "<" | ">" | "=" | “≤” | “≥” | "." | "," | ";" | "!" ;   *(* Symbol “ is not allowed *)*

*(*Insignificant whitespace is allowed before or after any attributes or values *)*

char = letter | digit | symbol ;

name = { letter } ;

description = { char } ;

value = “”” , { char } , “”” ; (* e.g. “Low” *)

type = “set” | “sequence” ;

leafvalue = “[” , value , “,” , value , { “,” , value } , “]” ;

(each leaf has two or more values, e.g. [“Low”,”Med”,”High”] *)*

treenumber = “0” | “1” | “2” | “3” | “4” ;

id = treenumber , { digit | letter } ;   *(* assuming Base 36 encoding, e.g. 001A *)*

attribute = id | name | description | type | leafvalue ;

children = “[“ , { node , { “ , “ , node } , “]” ;   *(* e.g. “children” : [{node}, {node}, {node}] *)*

attr_value = attribute | children ;   *(* attribute in a node description can be either a simple attribute like a name, or an array of attributes - children *)*

pair = “”“ , attribute , “”:”“ , attr_value , “”“ ;   *(* a pair example: “name”: “SaaS” *)*

node = “{“ , pair , { “,” , pair } “}” ;   *(* node definition is a set of pairs (name, description, children etc.) *)*

statement = id , (“ == “ | “ != “ | “ >= “ | “ <= “ | “ > “ | “ < “) , attr_value ;   *(* e.g. 001A >= “Low” *)*

rule = “if (“ , statement , { (“ && “ | “ || “ | “ ! “) , statement } , ") {" , statement , { (“ && “ | “ || “ | “ ! “) , statement } , "}" ;

(rule example: if (001A = “Low”) { 2004 = “Low” && 333B = “High” } *)*

In fact, the rules are set in JavaScript in the following manner:

```
if ( act_on_node (last_clicked, [0,0,0]) && act_on_node (d, [1,0,2]) ) return "red";
```

The script looks whether a value is clicked or not, and if yes, then another value "goes" red.

But, it is easier for the end-user to specify the rules as it is stated in EBNF notation, then the DSS tool transforms them into JavaScript notation automatically. In other words, there is transition between DSM syntax and DSS tool syntax. DSS can be implemented in any possible way different from the current one, but having a formal DSM syntax notation makes it possible to bridge those two easily.

4.2.1. Nodes definition

Nodes of which the trees consist of are defined using JSON notation and are stored in JSON files. This way, the trees are easily accessible, editable and reusable while JSON files can be accessed by most programming languages. To show how nodes are defined within trees, the following example defines a part of Business Process attributes tree:

```
{"id": "10",
 "name": "Business Process attributes",
 "description": "Set of attributes characterising a business process or a set of processes to be supported by cloud service",
 "children": [
  {"id": "100",
   "name": "Descriptive attributes",
   "description": "Descriptive attributes describe a business process or a set of processes to be supported by cloud service",
   "children": [
    {"id": "1000",
     "name": "Criticality",
     "description": "BP criticality tells how crucial a BP is for functioning of the company",
     "type": "sequence",
     "leafvalue": [ "Low", "Business-critical", "Mission-critical" ] } ] } ] }
```

All five trees are listed in full in online Appendix B[15].

4.3. Rules in the Model

4.3.1. Rules definition

Transitions between steps of DSM are governed by predefined rules. Most of BP, EA, and QoS attributes have a scale such as from Low to High, e.g. if an automated business process is mission critical – reliability QoS attribute will be high i.e. from high to high transition. As for Functional requirements and Architectural requirements - they either exist or not for each specific solution, e.g. a service should be IaaS (then it is not PaaS, SaaS, etc.) and have .Net command-line interface (some solutions do not provide it). First of all, when one option in the Functional requirements is selected, most of other options are disabled. E.g. if an IaaS solution is selected, PaaS, SaaS and others are disabled. However, some functionalities can be selected at once, e.g. different PaaS components such as languages or OS.

[15] http://gocloud.ga/appendixB.html

Further subchapters describe each step of the model with its connections. Current model does not pretend to encompass all possible relations – set of rules can be extended to fit the consultant using the model or a specific business case. Not all nodes have relations with other nodes for the same reason. In some cases, more nodes can be added to fit the case, or leaf values can be added/modified. Rules presented below are based on literature review and sometimes just on common sense.

To show the relationships between leaf nodes of the trees, next subchapters will demonstrate the rules that originate from Business Process attributes tree and the logic they are based on. For the sake of readability of this subchapter, unique node ids are substituted to node full names.

4.3.2. Logic behind rules

An analysis of business requirements combined with an understanding of the level of investment required to implement different solutions enables the development of a desired-level architecture that will achieve both business and technical objectives. As an indication of the level of investment, Total Cost of Ownership (TCO) will be used. It includes all costs such as acquisition, implementation, systems, networks, facilities, staff, training, and support, over the useful life of the solution chosen. We assume that there is always a trade-off between high Return on Investment (ROI) posing strict requirements and high TCO if those are met. This way, TCO is acting as an intermediate measure between different steps of the model and their requirements. E.g. when making a bridge between BP criticality and Performance QoS attribute we don't choose high performance when an automated process is not business or mission-critical due to high TCO associated with it. If not keeping in mind the TCO, we could set maximum performance, maximum security and so on for all attributes in spite of BP characteristics.

Any enterprise that is designing and implementing a high availability strategy must begin by performing a thorough analysis of the business drivers that require high availability.

Geographical and political issues are often posing strict data governance requirements for mission-critical processes. What will happen if a country changes its laws unexpectedly? The missing control could easily bring a business down when relying on cloud services that do not deliver anymore. This is a serious issue if mission critical data is inaccessible for a longer period. Therefore, many government agencies do not want to move their mission critical data like defense data, government strategy and certain important policy data [15].

Failure of the applications supporting mission-critical process have a direct impact on financial results of the company. Thus, cost of Downtime of such applications is very high.

The Recovery Time Objective (RTO) requirements are driven by the mission-critical nature of the business. Thus, for a system running a company's core process, the RTO is zero or very near to zero.

BP Criticality	Downtime	RTO		**Availability**	**Governance**	TCO
Mission-critical	Low	Low	→	High	High	High
Business-critical	Med	Med		Med	Med	Med
Low	High	High		Low	Low	Low

Here and after, green means good, red means bad

BP Availability		**Availability**	TCO
24 / 7		High	High
Office hours	→	Med	Med
Best effort		Low	Low

Business Process maturity defines the level of detail to what the BP is annotated, and thus ease of service implementation. And of course, the more mature BP is it requires more sophisticated IT service to handle it implying higher cost of running such system.

BP Maturity		**Performance & Accountability**	**Assurance & Standardization**	**Usability**	TCO
Optimizing / Managed		High	High	High	High
Defined	→	Med	Med	Med	Med
Repeatable / Initial		Low	Low	Low	Low

High BP integrity possesses service requirements to ensure that the process is performing its intended functions without being degraded or impaired by changes or disruptions in its internal or external environments. Such requirements arise in standardization challenges, business rules transparency, security restrictions and of course agility issues.

BP Integrity		**Data Integrity & Data Security**	**Assurance & Standardization**	**Agility**	TCO
Vital		High	High	Low	High
Important	→	Med	Med	Med	Med
Non vital / Unimportant		Low	Low	High	Low

These attributes are rather straightforward in posing architectural requirements.

BP Confidentiality		**Security & Privacy**	**Accountability**	TCO
Top secret		High	High	High
Limited restrictions	→	Med	Med	Med
Public		Low	Low	Low

First of all, Business Process input/output quality relates to data quality restrictions. To achieve integrity, accuracy, completeness, non-obsolescence, accessibility, etc. of data the requirements should be posed to quality of service architecture.

Amount of Data	**Accessibility**	**Accuracy & Completeness**		**Interoperability**	**Accountability & Assurance**	**Data Security**	TCO
Big	High	Vital		High	High	High	High
Med	Med	Important	→	Med	Med	Med	Med
Low	Low	Non vital / Unimportant		Low	Low	Low	Low

Degree of personnel involvement into Business Process execution affects the requirements for the service used to support or automate it. So, the higher involvement is, the stricter the requirements are.

HR Involvement				**Service Quality**	TCO
Domain Knowledge				Usability	
Qualification				Maintainability	
Certification	High		High	Communication	High
Experience	Med	→	Med	Accessibility	Med
Time Management	Low		Low	Learnability	Low
Communication Skills				Operability	
				Transparency	

Understandability

4.3.3. Rules

These are the rules defined for BP attributes tree:

BP. Descriptive Attributes. Criticality → BP. Descriptive Attributes

if (Criticality = "Low") **{** Availability ≤ "Office hours (08.00–18.00, weekdays)" **}**
if (Criticality ≥ "Business-critical" **)** **{** Availability ≥ "Office hours (08.00–18.00, weekdays)" **}**

BP. Descriptive Attributes. Criticality → EA. Quality of maintenance policy

if (Criticality = "Low" **)** **{** Management Value ≤ "Medium" **}**
if (Criticality ≥ "Business-critical" **)** **{** Management Value ≥ "Medium" **}**

BP. Descriptive Attributes. Criticality → EA. Quality of maintenance policy

if (Criticality ≥ "Business-critical" **)** **{** Management Value ≤ "Moderate" **}**

BP. Descriptive Attributes. Criticality → QoS. Assurance

if (Criticality = "Low" **)** **{** Stability ≤ "Medium" **}**
if (Criticality ≥ "Business-critical" **)** **{** Stability ≥ "Medium" **}**

BP. Descriptive Attributes. Maturity → EA. Business architecture quality

if (Maturity = "Initial" **)** **{** Relations between Business Processes ≤ "Medium" **}**
if (Maturity ≥ "Repeatable" **)** **{** Relations between Business Processes ≥ "Medium" **}**

BP. Descriptive Attributes. Availability → EA. Quality of maintenance policy

if (Availability ≤ "Office hours" **)** **{** Cost of maintenance policy ≤ "Medium" **}**
if (Availability ≥ "Extended of.h." **)** **{** Cost of maintenance policy ≥ "Low" **}**

BP. Descriptive Attributes. Availability → QoS. Assurance

if (Availability ≤ "Office hours" **)** **{** Availability ≤ "Medium" **}**
if (Availability ≥ "Extended of.h." **)** **{** Availability ≥ "Medium" **}**

BP. Descriptive Attributes. Availability → QoS. Financial

if (Availability ≤ "Office hours" **)** **{** On-going cost ≤ "Medium" **}**
if (Availability ≥ "Extended of.h." **)** **{** On-going cost ≥ "Medium" **}**

BP. Descriptive Attributes. Confidentiality → QoS. Security and privacy. Privacy and Confidentiality

if (Availability ≤ "Company conf." **)** **{** Access control & privilege management ≤ "Medium" **}**
if (Availability ≥ "Restricted" **)** **{** Access control & privilege management ≥ "Medium" **}**

BP. Descriptive Attributes. Relations with other BP → EA. Business architecture quality

if (Relations with other BP = "Independent" **)** **{** Relations between BP ≤ "Medium" **}**
if (Relations with other BP ≥ "Related" **)** **{** Relations between BP ≥ "Medium" **}**

BP. Enactment quality. Function → QOS. Performance. Functionality

if (Suitability = "Low" **)** **{** Suitability ≤ "Medium" **}**
if (Suitability ≥ "Medium" **)** **{** Suitability ≥ "Medium" **}**

<u>BP. Enactment quality. Function → QOS. Performance. Functionality</u>

if (Accuracy = "Low" **)** **{** Accuracy ≤ "Medium" **}**
if (Accuracy ≥ "Medium" **)** **{** Accuracy ≥ "Medium" **}**

<u>BP. Enactment quality. Function → QOS. Security and privacy. Privacy and Confidentiality</u>

if (Security = "Low" **)** **{** Governance ≤ "Medium" **}**
if (Security ≥ "Medium" **)** **{** Governance ≥ "Medium" **}**

<u>BP. Enactment quality. Function → QOS. Assurance. Reliability</u>

if (Reliability = "Low" **)** **{** Maturity ≤ "Medium" &&
Fault tolerance ≤ "Medium" &&
Recoverability ≤ "Medium" **}**
if (Reliability ≥ "Medium" **)** **{** Maturity ≥ "Medium" &&
Fault tolerance ≥ "Medium" &&
Recoverability ≥ "Medium" **}**

<u>BP. Enactment quality. Function → QOS. Performance. Efficiency. Resource behaviour</u>

if (Resource Utilization = "Low" **)** **{** Throughput ≤ "Medium" **}**
if (Resource Utilization ≥ "Medium" **)** **{** Throughput ≥ "Medium" **}**

<u>BP. Enactment quality. Function → QOS. Usability</u>

if (Understandability = "Low" **)** **{** Understandability ≤ "Medium" **}**
if (Understandability ≥ "Medium" **)** **{** Understandability ≥ "Medium" **}**

<u>BP. Enactment quality. Function → QOS. Usability</u>

if (Learnability = "Low" **)** **{** Learnability ≤ "Medium" **}**
if (Learnability ≥ "Medium" **)** **{** Learnability ≥ "Medium" **}**

<u>BP. Enactment quality. Function → QOS. Performance. Efficiency. Time behaviour</u>

if (Time Efficiency = "Low" **)** **{** Start-up time ≤ "Medium" &&
Response time ≤ "Medium" &&
Recovery time ≤ "Medium" &&
Shutdown time ≤ "Medium" **}**
if (Time Efficiency ≥ "Medium" **)** **{** Start-up time ≥ "Medium" &&
Response time ≥ "Medium" &&
Recovery time ≥ "Medium" &&
Shutdown time ≥ "Medium" **}**

<u>BP. Enactment quality. Non-Human Resource → QOS. Performance. Functionality</u>

if (Suitability = "Low" **)** **{** Suitability ≤ "Medium" **}**
if (Suitability ≥ "Medium" **)** **{** Suitability ≥ "Medium" **}**

<u>BP. Enactment quality. Non-Human Resource → QOS. Performance. Functionality</u>

if (Accuracy = "Low" **)** **{** Accuracy ≤ "Medium" **}**
if (Accuracy ≥ "Medium" **)** **{** Accuracy ≥ "Medium" **}**

<u>BP. Enactment quality. Non-Human Resource → QOS. Performance. Functionality</u>

if (Security = "Low" **)** **{** Governance ≤ "Medium" **}**
if (Security ≥ "Medium" **)** **{** Governance ≥ "Medium" **}**

<u>BP. Enactment quality. Non-Human Resource → QOS. Assurance</u>

if (Reliability = "Low" **)** **{** Reliability ≤ "Medium" **}**
if (Reliability ≥ "Medium" **)** **{** Reliability ≥ "Medium" **}**

<u>BP. Enactment quality. Non-Human Resource → EA. Quality of maintenance policy</u>

if (Reliability = "Low" **)** **{** Risk ≤ "Low" **}**
if (Reliability ≥ "Medium" **)** **{** Risk ≥ "Moderate" **}**

<u>BP. Enactment quality. Non-Human Resource → QOS. Performance. Efficiency. Time behaviour</u>

if (Time Efficiency = "Low" **)** **{** Start-up time ≤ "Medium" &&
Response time ≤ "Medium" &&
Recovery time ≤ "Medium" &&
Shutdown time ≤ "Medium" **}**
if (Time Efficiency ≥ "Medium" **)** **{** Start-up time ≥ "Medium" &&
Response time ≥ "Medium" &&
Recovery time ≥ "Medium" &&
Shutdown time ≥ "Medium" **}**

<u>BP. Enactment quality. Non-Human Resource → QoS. Performance. Efficiency. Resource behaviour</u>

if (Resource Utilization = "Low" **)** **{** Throughput ≤ "Medium" **}**
if (Resource Utilization ≥ "Medium" **)** **{** Throughput ≥ "Medium" **}**

<u>BP. Enactment quality. Non-Human Resource → EA. Technology architecture quality</u>

if (Robustness = "Low" **)** **{** Level of Standardization of Platform Technology Used ≤ "Medium" **}**
if (Robustness ≥ "Medium" **)** **{** Level of Standardization of Platform Technology Used ≥ "Medium" **}**

<u>BP. Enactment quality. Non-Human Resource → QoS. Assurance</u>

if (Availability = "Low" **)** **{** Availability ≤ "Medium" **}**
if (Availability ≥ "Medium" **)** **{** Availability ≥ "Medium" **}**

All rules between all trees are listed in full in online Appendix C[16].

In this chapter we have seen how five taxonomies derived from the literature form a Decision Support Model by linking to each other with rules. Now all the leaf nodes have values either of a sequence type (e.g. low, med, high) or of a set type (e.g. VMware, Xen, FreeBSD). The rules govern the relationships between the steps in this way: if a value is selected on one tree, then one or more values are restricted to select in the same or following trees.

This model of step-by-step selecting requirements and eliminating unnecessary ones is used in a DSS tool described in the next chapter.

[16] http://gocloud.ga/appendixC.html

5. DSS concept validation

In this phase of the research, the theoretical concept of DSM is validated with expert interviews. Based on their opinion whether the DSS tool based on proposed DSM concept will be useful in a job of Business Analyst / IT Consultant / EA Architect. The interview structure and protocol are based on UTAUT constructs as described in Chapter 2. A list of 6 experts participated in the interview is given in Appendix D. For the sake of privacy, their names are not shown in the list, but their professional background casted in it, speaks for itself. The results of the interview presented as a grouping summary of all interviews is following the interview structure subchapter.

5.1. Interview structure

Before the actual interview, the author sent the invitation emails containing a presentation[17] of the research. The presentation can be read in 2-4 minutes and it gives an impression on what the research is about. First, the expert can decide whether it is in his interest to participate in the interview concerned with presented topic. Second, the interviewees that agreed to participate already knew the topic, what the research is about, and how the DSM concept work. Of course, that superficial understanding is further elaborated in the beginning of the interview.

The semi-structured interviews used for theoretical concept validation contains four sections. In the first section the author explains his background and asks contextual questions about the interviewee, e.g.:

- Can you please tell me a bit more about your background?
- How many years of experience do you have with cloud solutions and what kind?
- How do you estimate the difficulty of choosing one?

The second part contains the explanation of the DSM concept. First, a webpage containing a presentation is shown to an interviewee. The presentation which lasts for around 4 minutes includes the research question, the scope of research, target user audience, the idea of five consecutive steps for choosing a cloud solution, and the logic between requirements transition from one step to another. After that, an interactive webpage with five separate trees and a complete DSM is shown to an interviewee. While an interviewee can explore the steps and see how requirements are grouped in taxonomies, interviewer explains the DSM concept. Each interview was individually different, but the main narrative is formulated as follows:

- Today we will not explore the tool, but only the theoretical concept behind it. After validating the concept and gathering remarks, a tool will be developed.
- We are now looking at steps of the model, the whole model will follow after them. To proceed to the next step just scroll down or click the on the next step in the navigation bar on top of the page. In each taxonomy, you can click on plus signs to expand branches of the tree. Sometimes there are examples of specific cloud services or mechanisms and they are shown in blue.
- First step is about choosing the functionality of the cloud…
- Second step is about choosing the attributes that the Business Process(es) that needs to be automated has(ve). This step is critical to have in the requirements gathering process while most any software that facilitates business needs to be aligned with BP goals.
- Third step concerns Enterprise Architecture. This is the lower granularity view on the company, than the previous step. EA basically stands for all IT in the organization and how it is organized.
- Fourth step is about Quality attributes of a future cloud service…

[17] Presentation can be accessed at: http://gocloud.ga/presentation.html

- The last step concerns specific implementation tactics – which security protocol to use, what type of web-interface the service should have and so on.
- Now, all the steps are combined together to form a DSM. Nodes with text in bold are nodes, while leaves are with normal text - these leaves are node values. The links between nodes represent rules between them. To give an example, in a developed DSS, if on the BP attributes tree privacy attribute marked as high, such option as low security in QoS tree disappears from the user view. Some attributes are simply restricting the choice of others. As for the leaf values, low/med/high or interface 1/interface 2 values are to be elaborated after the full list of the requirements is set. Due to unfeasibility of putting all possible values in the DSS, it is left to the expert to fill the necessary details when a preliminary RFP is listed after the final DSM step.

This interactive introduction to the DSM take around 15 minutes.

Third part of the interview includes UTAUT questions which are to measure corresponding UTAUT constructs. These questions are used not as a quantitative survey with loadings per question, rather as a guideline for a semi-structured interview. They were reformulated to fit the current research. Also, some of them were dropped or grouped to save valuable time of interviewees, since it is not a questionnaire but a semi-structured interview. Each question is to be discussed with interviewer. This part consumes around 15 minutes. These are the UTAUT questions for corresponding constructs and sub constructs explained by UTAUT definitions:

<u>Performance expectancy</u>

Perceived usefulness - the degree to which a person believes that using a particular system would enhance his or her job [83]:
<u>U6.</u> Do you think that using such DSS would be useful in for choosing a cloud service?

<u>U1,5.</u> Do you think that using such DSS would enable choosing a cloud solution more quickly or would make it easier?

<u>U2,3,4.</u> Do you think that using such DSS would increase Analyst/Consultant's job performance, productivity or effectiveness?

Job-fit - how the capabilities of a system enhance an individual's job performance [83]:

<u>JF5.</u> Do you think that use of such DSS can increase the quantity of output for the same amount of effort?

<u>JF6.</u> What can be a general extent to which use of the DSS could assist on the Analyst/Consultant's job?

Outcome expectations relate to the consequences of the behaviour:

<u>OE2.</u> Do you think that using such DSS one will spend less time on requirements analysis?

<u>Effort expectancy</u>

Perceived Ease of Use - The degree to which a person believes that using a system would be free of effort [83].

<u>EOU1.</u> Learning to operate the DSS based on this concept should be easy in your opinion?

<u>EOU2.</u> If using such DSS, would you find it easy to get the system to help you in choosing a cloud service?

<u>EOU4,6.</u> Do you think that a DSS based on this concept will flexible/modifiable and thus easily adoptable for a specific business case?

Complexity - the degree to which a system is perceived as relatively difficult to understand and use [83].

CO1. Using the system takes too much time from normal duties.

CO2. Working with such DSS might be so complicated, that is difficult to understand what is going on.

CO3. Using such DSS might involve too much time doing mechanical operations (e.g., data input).

The fourth part of the interview is a free chat with an interviewee. Interviewer asks interviewee what are his/her general thought about the concept and future DSS, can he/she bring up any other applications for this DSM, and any other questions he thinks may be relevant. According to the experience, the whole interview lasts approximately 47 minutes (from 34 to 63 minutes).

5.2. Interview results

Interview transcripts are represent as notes taken during the interviews. Full collection of them is present in Appendix E. A summary of all findings derived from the interview is presented below structured in a way the asked questions correspond to UTAUT sub-constructs.

5.2.1. Performance usefulness

All experts agreed that such a DSS would be useful in choosing a cloud service. Such a DSS might not speed the process up, but it will formalize it, perhaps make it more efficient by suggesting requirements that might be skipped otherwise. Most of the interviewees agreed that a standardized method is usually more effective and what is most important, is that:

- a standardised method ensures the result;
- explicit procedure lowers the number of alterations needed;
- formalized / structured process increases the maturity of organization as whole;
- such a detailed RFP with all the requirements written down explicitly is an insurance both for a client and for a vendor.

However, being a support only tool, the DSS is not sufficient for choosing a cloud. Having such a checklist of high-level requirements doesn't exclude all the work that should be done when deciding upon a cloud vendor. It only helps in saving time during requirements gathering part, not contracting or implementation phases.

Writing a proposal also takes a lot of time, and following this rigor procedure will not make it more. If a goal is to solve a partial problem (not a complete proposal, but a part of requirements list), then yes – it will consume more time than needed.

5.2.2. Job-fit

Knowledge is needed to be able to use the DSS - rules and taxonomies require deep knowledge of the concepts behind them. However, with adoption of such a formalized approach, it is easier to transfer knowledge and less experienced Analysts are insured. Thus, such a formal procedure supports upcoming experts. For them, starting professionals, the system may be too complicated (requires a lot of knowledge behind all the concepts), but still will eventually increase job performance. On the contrary, for a high-level expert, following this procedure might be unnecessary.

The fact that sufficient amount of knowledge is required to use the tool is not a drawback while DSS tool is not a black box, but just a support tool. If the tool doesn't suggest anything (specific solutions) – it is good, because it leaves critical decisions to be made by humans.

Important aspect mentioned by one of the interviewees is that the tool will be useful only for high level consultants, not technical people implementing the solution. It covers the general approach, but does not go deep into specifics. As interviewee 4 indicated, there is always a gap between high-level part and low-level part – people from business and technicians. This tool doesn't solve that gap problem, and after these high level requirements are gathered, developers have to see whether the solution is really feasible.

5.2.3. Perceived ease of use

All DSS systems are alike, it is logical. One tricky thing is that one step is dependent on other steps – not everyone is familiar with this this concept. It is more than just a checklist with five levels. Therefore, most of the interviewees agreed that a briefing is necessary to learn how to operate the tool. Two to three days seminar and an audit afterwards will settle the tool as a company-wide practice. Also, might be useful to have at least one person who has an experience in using this DSS and is able to walk through new users. It is very important to always have an expert that is able to explain all the process to others.

The DSS tool will be very useful if BP and EA trees will be highly and easily modifiable and can be tailored to specific company completely. Only QoS tree is universal, the first two should be at least modified per sector. The flexibility is ensured by the tool development methods (and ideally by a good interface), but how easy it is? Interviewees note that an interactive interface is better for a user, and different language support is very important as well.

If different consultants can collaborate on one requirements document – that would unite people working on one project from different units e.g. functional designers, developers, security specialists.

5.2.4. Complexity

All interviewees conclude that using DSS tool will consume more time because an Analyst is following a thorough procedure. Also, using such DSS might involve more time as usual in the beginning as any new procedure adopted by the company. However, as soon as it is part of the duties, it is ok to use any method and spend time on it as soon as the result meets expectations. The time spent on requirements gathering may increase, but the number of alterations will decrease. Interviewee 3 suggests that it is also possible to share data input with a client using surveys for data collection etc. Interviewee 5 argues that, when gathering requirements, an Analyst needs to type a lot anyway, and if the time on the input increases than it simply means that before the requirements were gathered worse. Moreover, with such a tool with an elaborate checklist and relationships that rule out some options, the process will take less time on the same level of precision.

The amount of effort put into following the prescribed steps increases, but traceability of requirements gathering process also increases – and that is more important. Moreover, double-checking takes even more time – an Analyst should always verify his assumptions with a client. Again, interviewees agree that more effort increases quality of the result.

Interviewee 1 argues that the terms used in the taxonomies don't mean a lot to clients, but having a glossary to align the experts with each other would be great. The rest of interviewees agree that a glossary has different meanings between client and experts – e.g. the meaning put in these terms by BA is not the same as the client perceive. But, these terms are a good anchor between experts when they want to demonstrate the logic they use to each other. Other example is that for a provider receiving such an elaborate RFP it is easy to go through requirements and conclude whether it can fulfil them or not. Still, from a client perspective, it is not easy to choose a cloud based on this list – it is a list for a professional e.g. Consultant. Summing all that, the tool is dedicated to use by professionals that operate the same terms – it might be one company, one sector or one region. Otherwise, a thorough glossary alignment should be made before the tool is used.

5.2.5. Alternative applications

- The process of choosing a cloud service is described well and can be used in educational purposes.
- Governmental organizations, NGOs etc. have specific prescribed rules for tendering software. If it will be possible to put weights on the requirements in DSS tool, it will be a great tool for specifying a

complete RFP with weightings for a tender. Thus the vendors will have sufficient information for applying to it.

- Basically, many instances of IT (e.g. outsourcing), logistics, or other RFP generating tools can use the same concept.
- The DSS can be used for an analysis of the system which is in place in an organization. With doing that it is possible to see whether it is possible to add new functionality or compare a legacy system with a newer one, or non-cloud to a cloud one.
- Such a concept of going through a checklist of requirements is wise, but it will be even better for an enterprise to have a methodology of making such checklists for other processes, a method/system according to which the model can change as well.

5.2.6. Discussion of the interview results

The theoretical concept of DSM was validated with 6 expert interviews. The interview structure and protocol are based on UTAUT constructs as described in Chapter 2. A list of experts participated in the interview is given in Appendix D. All the 6 experts are working with cloud solutions either from academia or business perspective point of view, and their considerable experience insures the high level of expertise. After the first three-four interviews the saturation was reached – all interviewees were stressing the same positive points, but all interviews had interesting points e.g. possible uses or tool functionality.

All experts agreed that such a DSS would be useful in choosing a cloud service. Such a DSS might not speed the process up, but it will formalize it and make it more rigor. Most of the interviewees agreed that a standardized method is usually more effective, it ensures the result, increases the maturity of an organization, and that such a detailed RFP with all the requirements written down explicitly is an insurance both for a client and for a vendor. The time spent on requirements gathering may increase, but the number of alterations will decrease. Writing a proposal also takes a lot of time, so this will not make it more – for complete proposal. If a goal is to solve a partial problem, then yes – it will consume more time than needed.

Based on these conclusions, the DSS tool based on proposed DSM concept is promising to be useful in a job of Business Analyst / IT Consultant / EA Architect. Of course, if developed with a proper interface. Interviewees note that an interactive interface is better for a user, and different language support is very important as well.

6. DSS approach

The following section describes how the concept of a DSM can be used in DSS tool. The DSS tool is partially implemented using development techniques described further. The part of the tool can be examined at the website http://www.gocloud.ga. The overall design of the tool is presented as a mockup. While a tool is developed only partially, the tool's mockup includes the features that were suggested by experts in the series of interviews for concept validation. The first subchapter presents the use cases of proposed DSS – how can it be used, for what purpose, and by whom. The second one – DSS tool interface mockup, and the third one – what was implemented to the date of finishing the paper.

6.1. DSS use case scenarios

In this subchapter, few possible DSS tool use scenarios are shown to illustrate who can use the tool and for what purposes. As it was already described in the second chapter, target user audience is searching for clouds to support business products and services. It is not aimed at choosing a research, science, governmental, defence, or any other special types of clouds. Necessary modifications to the tool needed to adopt it in a consultancy company are made before the use cases below. These alterations can vary from a general remarks to align the tool with company procedures to a thorough customization for different industries. That is up to a particular consultancy company. Minor alterations can be made during requirements gathering from a client and performed by an Analyst.

Scenario 1

First scenario deals with an Analyst performing a requirements gathering procedure. He or she is going through DSM model steps, finding out how the business process works at the client, what is the EA state, what are the requirements for the cloud service. After that, he extends the generated list with specific requirements and sends the list to the client for the approval. After client validated the list, an RFP with it can be distributed to preselected vendors to see if they can fulfil stated requirements. If the solution is not feasible, then some requirements from the list are relaxed, and the RFP is sent again.

To give an example of relevant resources where cloud vendors and their offerings are presented, they are gathered here:

1. CloudBook.net,
2. CloudShowPlace.com, and
3. CloudTaxonomy.Opencrowd.com.

Also, such tool as SMI index evaluation can be used to compare vendors based on QoS rankings. Further description of this method can be found here: SMI framework.

Imagine an Analyst X, who has a task to analyse company Y for the possible migration of their ERP solution to the cloud.

He first goes to the company and clarifies what type of functionality should the software have. He concludes that the current system is an ERP software with PDM, CRP, MRP, and work-order management tools in it. The client also wants forecasting possibility and shop-floor control in the new cloud system. At this stage he selects SaaS ERP in the DSS tool, nothing more. This results in eliminating options in further steps such as IaaS and PaaS management tools and access interfaces.

On the second step of the selections process, he gathers information about business processes of the client company. He sets values in the second step of the DSS tool, client company employees may help him with that by answering short questionnaires. These selections mainly affect EA and QoS attributes restricting some choices.

On the third step, he talks with IT department of the client company – EA Architect, IT director, CIO etc.

The result of that is a completed third step about IT in the client company. Again, this restricts some attributes on further steps to insure that all software systems, data formats, interfaces, thin clients etc. will be aligned with a new cloud ERP.

On the forth step an Analyst selects required QoS attributes for the system. Again, this is done by asking the client about how the system should run and what the boundaries of the QoS attributes are. An Analyst does not fill specific QoS values, e.g. in Availability QoS he puts high.

Fifth step is to show the client what are the possible options that may be asked from the cloud vendor.

After all five steps are ahead, an Analyst presses the "Generate result" button in the DSS tool and receives a list with requirements that he has set. This is the time to configure the list so it will be used as a base for RFP document. In functional requirements he states SaaS ERP with PDM, CRP, MRP, forecasting, MPS, work-order management, and shop-floor control. In interface requirements (from EA step) he states compatibility with external system SAP e-Invoicing, virtualization platforms Microsoft Hyper-V, Citrix Xen Apps, VMware ESX, ect. In assurance QoS he specifies the availability value high with 99.999% uptime and so on. All this is done *together* with the client! This insures that the details are correct. The fact that important requirements are not missed is granted by the step-by-step procedure. Both high-level DSM procedure and low-level requirements specification allow following a thorough procedure and saving time at the same time. Simple divide and conquer method. This way, finally, the requirements list is ready.

Good practice is to loop the process after the provider selection. It means, that after the requirements are formulated and the vendor offers a solution, the Analyst should go through the requirements again to see whether the solution meets them indeed.

To see an example of this scenario, a sample case study is presented in Appendix G.

Scenario 2

One of the interviewees argued that the DSM model too general to support the full process of cloud selection. It is useful for business people (consultants), but there is still a gap between them and technical people. And choosing a particular solution is technical, and here it doesn't help. However, he doesn't say that the DSS can handle all the process – quite in opposite, there are too any specific technical details at the later phases. Still, he doesn't like that there is no solution to the gap between technical level and abstract business level. Perhaps, bridging them at least for some industry sectors might work. Therefore, a use case scenario can be one about industry tailored DSS tool.

A consultancy company has different modifications of the tool. E.g. one for manufacturing industries (with more accent on SaaS clouds), one for IT industry cloud software (more focused on IaaS and PaaS clouds), etc.

Scenario 3

A scenario which is not yet possible, but is an interesting one to implement in future – reverse engineering from vendor solution. If a vendor can specify the requirements (features) list in the same format as the DSS tool export format, it can be imported in a tool. After that, it is possible to see how organizational changes affect the solution, or what are the properties of an enterprise that should permit the adoption of this system.

The input functionality provides more possibilities, such as if a Client specifies the requirements first, then they are imported in the tool and then it is possible to see what functionality can be added/modified, what QoS may be top up or is the migration to another service is feasible (by comparing two inputs).

6.2. DSS tool interface mockup

It was decided to represent the taxonomies in the DSS tool as tree structures. Reingold Tilford Tree is a type of visualization, named after the two scientists who first formalized it in 1981, which is suitable to provide a light and pleasant representation of large trees of information with several layers of depth. As shown in Figure 71, when the next level is expanding – the previous one is shrinking enabling the whole tree to maintain its size within a restricted area. This way of representing trees was used also because of the ease of its implementation as well as the ease of reusability. This, in turn, is due to the use of Java Script library d3.js - Data-Driven Documents[18]. It combines powerful visualization components with a data-driven approach needed to handle taxonomies' data. Thus, the JSON database files are used to visualize the taxonomies, while they can be easily modified, ported or reused afterwards. The developed part of the DSS tool is available at http://www.gocloud.ga.

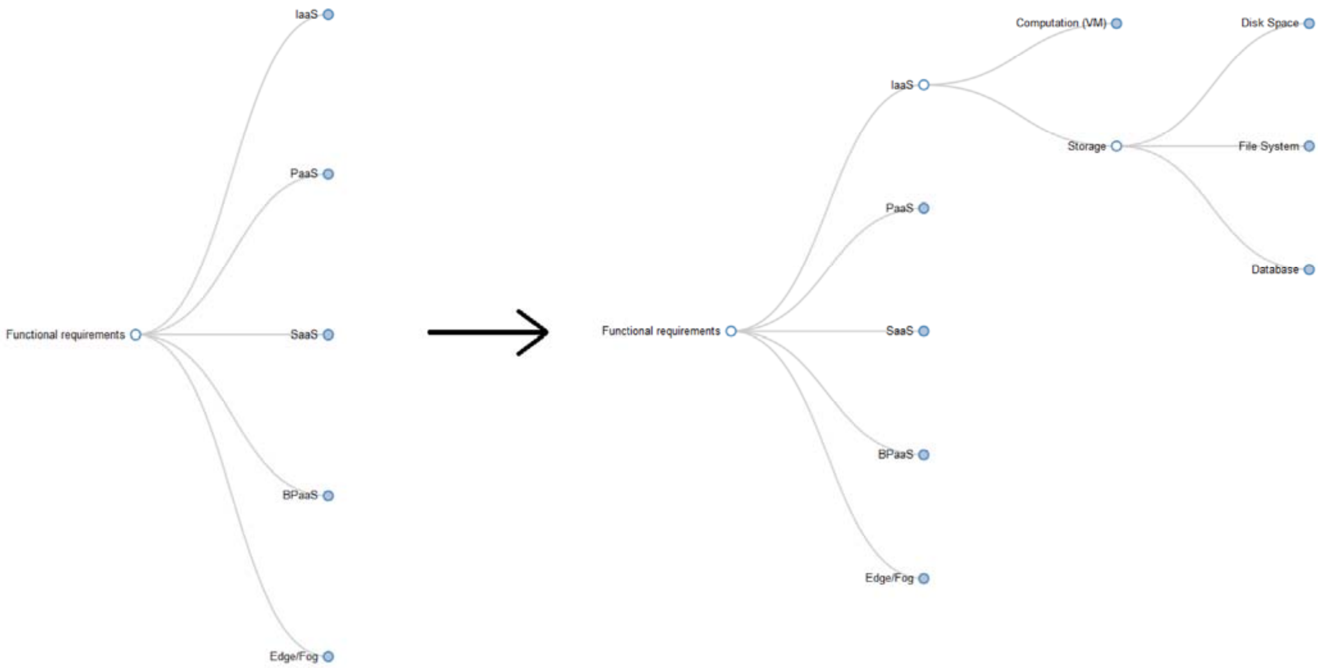


Figure 71 Reingold Tilford Tree automatic resizing feature

The DSS tool can be accessed with any desktop or mobile browser capable of handling a JavaScript query. It consists of taxonomies visualisations and a menu:

[18] More information is available at: http://d3js.org/

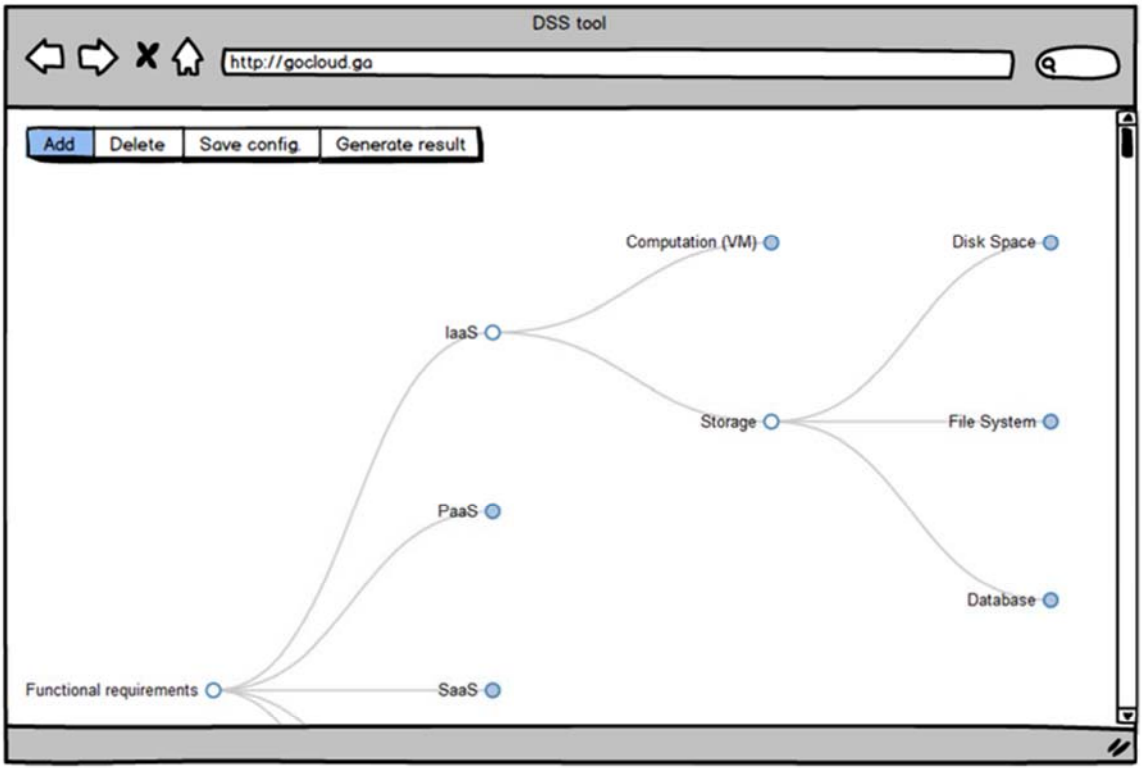


Figure 72 DSS tool interface mockup

When selecting a value of an attribute (leaf node or several leaf nodes), the preconfigured rules apply. That results in disappearance of other attributes' values in other trees. Double click on the node/leaf value opens a full list of properties of the node in a popup floating box. It contains id, name, description, and two lists of rules – influencing rules from other nodes and rules triggered by setting a value to this attribute. Modifiable fields are all except id (which is assigned automatically) and incoming rules – they have to be edited in the nodes of origin. Of course, these lists of rules only appear if the node is a leaf node.

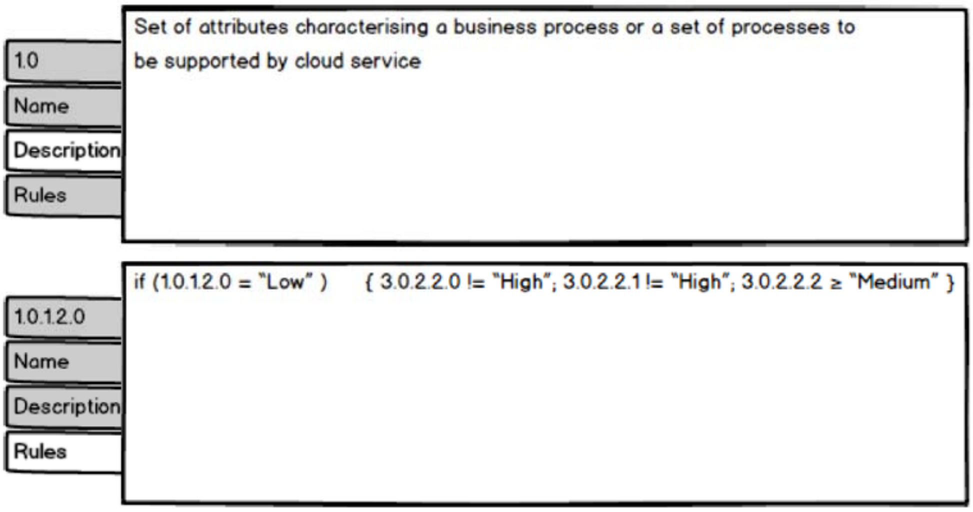


Figure 73 Double click on a node action - node properties menu

Adding, editing, and deleting nodes is possible, but not for model-critical nodes. Nodes that cannot be changed are simply restricted to change by the developer. Less important nodes, and all the leaf values are possible to adopt to situational needs. Adding a node is done by clicking on the "Add" button, deleting one – by first clicking on the node and then clicking on "Delete" button. Editing is possible within the menu toggled by double-click. Part of this functionality – adding and dragging nodes - is already made available using d3.js library thanks to Rob Schmuecker[19]. Drag and drop functionality part adopted to the current DSS prototype can be examined at the following link: http://gocloud.ga/drag&drop.html.

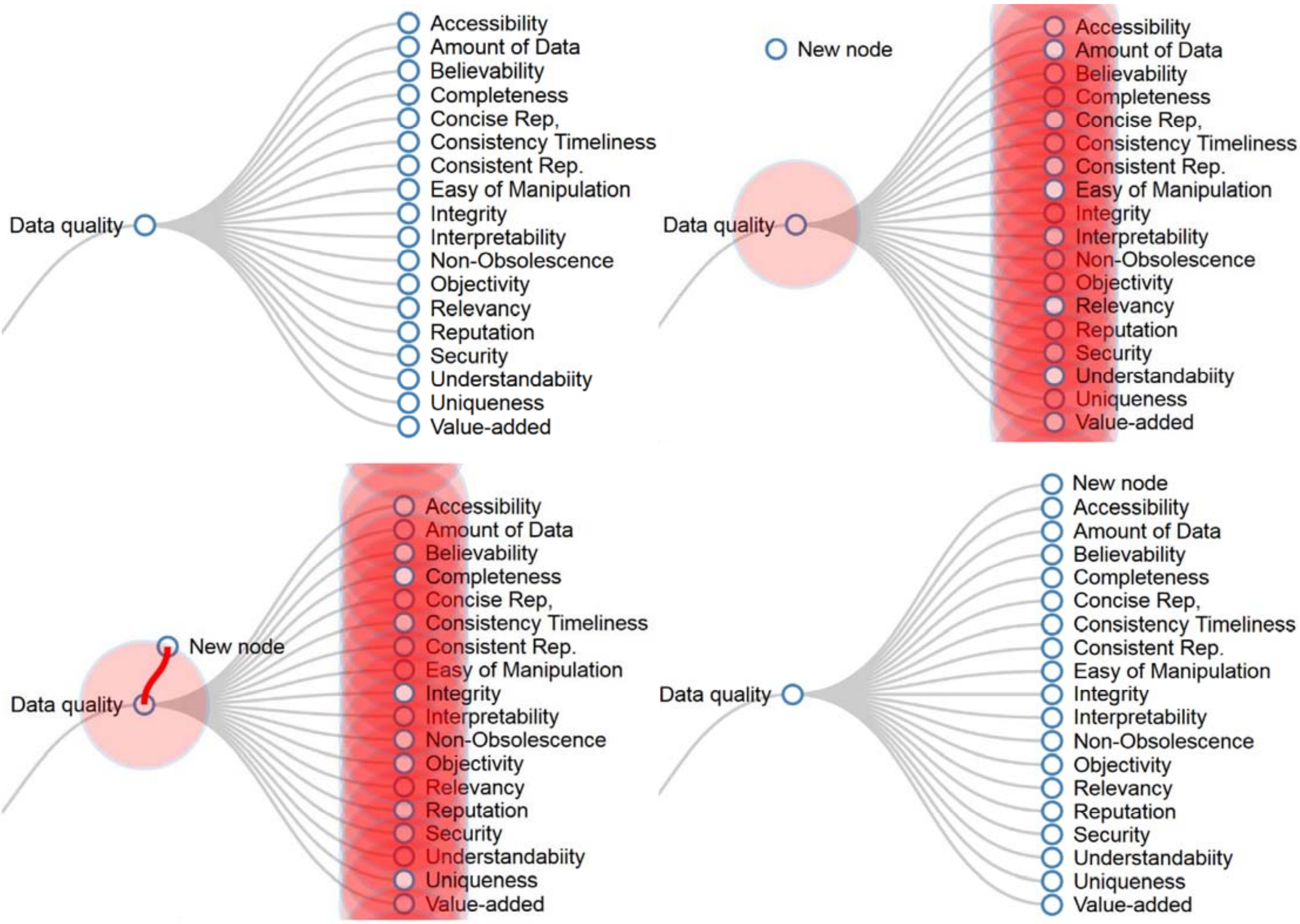


Figure 74 Drag and drop functionality

After adding a new node, the user can add the name, description, and rules by clicking on the node and working with properties tab. The current user-configuration of the taxonomies can be saved into a new JSON file using "Save config." button:

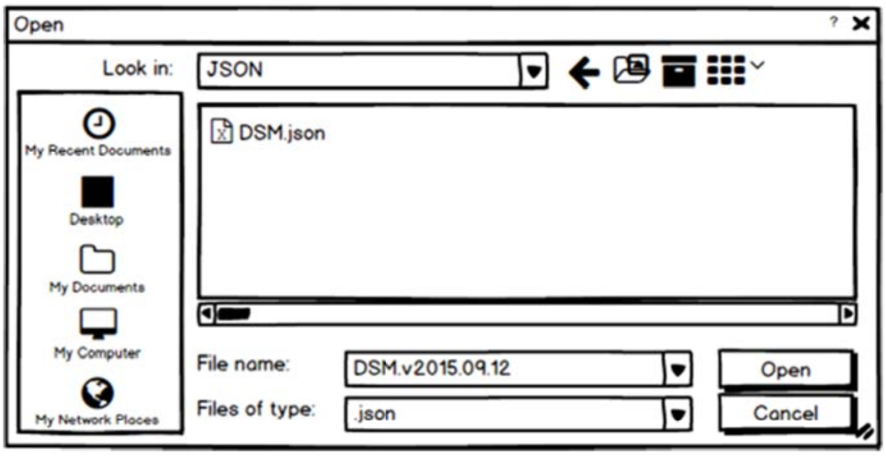


Figure 75 DSM configuration save dialogue

[19] http://stackoverflow.com/users/1672632/rob-schmuecker

Pointing to a node results in displaying the description of this node:

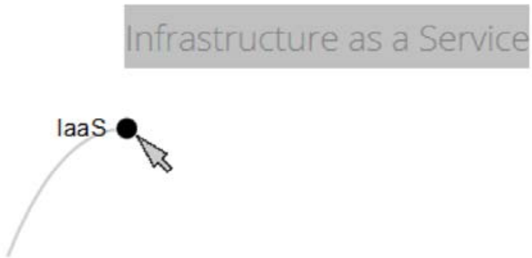


Figure 76 Hover over a node action

After all the steps are done, user can click on "Generate result" button. After that, the tool exports the list of selected requirements as an xls/xml file. After the list is generated, it is time for Analyst to refine it elaborating the requirements. A short sketch of that is shown in Figure 77, where the generated list is on the left and elaborated one is on the right:

Functional requirements	**Functional requirements**
SaaS ERP: to fill in	SaaS ERP with PDM, CRP, MRP, forecasting, MPS, work-order management, and shop-floor control.
Interface requirements	**Interface requirements**
External systems: to fill in	External systems: SAP e-Invoicing
Virtualization Platform: to fill in	Virtualization Platform: Microsoft Hyper-V, Citrix Xen Apps, VMware ESX
Web browser: to fill in	Web browser: Internet Explorer 9 and late; Google Chrome 3.1 and later
Assurance QoS	**Assurance QoS**
Availability: high	Availability: 99.999% uptime
Performance QoS	**Performance QoS**
Start-up time: low	Start-up time: less than a minute.

Figure 77 Elaborating generated requirements list

In future development of the tool, it may be a good addition to the functionality, if it would be possible to generate different versions of those requirements lists e.g. one for RFP to be sent to a vendor, second one for a client, and the third one for analyst's own use. They could differ in the level of details and also the format.

6.3. Developed part of DSS tool

A part of the DSS tool was developed by the author, however due to the time constraints it lacks full functionality. Tool prototype can be examined at http://gocloud.ga. The source code is available in online Appendix F.

Interactive part of the tool in which the predefined rules apply was also developed partially. At this link http://gocloud.ga/rules.html it is possible to examine the interactivity, but not all the rules are integrated into DSS tool prototype. In the mentioned example two rules apply - when Low is selected at one branch, high is not possible at another, and vice versa. Selecting the middle option does not restrict any further choice. These three scenarios are shown in Figure 78 below:

Figure 78 Interactive rules example

7. Results and conclusion

To better interpret the results and to reach a logical conclusion about the thesis, the research question and its sub questions are restated below.

Main research question:

What is an effective systematic procedure of selecting a cloud service for an enterprise?

The main research question is answered fully, the effective systematic procedure of selecting a cloud service was found. Further development of the DSS tool prototype may bring this method of cloud service selection to the market.

During the concept creation phase, it was decided that on a high level of abstraction, there are four levels of view on IT: business level perspective; Enterprise Architecture level; QoS requirements for *a cloud* service; specific features (implementation level). Of course, there is the most important step before going through those levels – what kind of system should it be – functional requirements. This way, the concept of a DSM for choosing a cloud service was formed to allow an Analyst not to miss any requirements at any level of abstraction, or at least to minimize this number.

To answer the main research question and develop a DSM consisting of these five steps, three sub questions were formulated: what is in each step; how the steps relate to each other to form a cloud selection process; and does this concept work?

Research sub questions:

RsQ1. What are the necessary attributes for each step in the model?

This sub question was answered with the production of five taxonomies.

First one is focusing functional requirements. Emerging from NIST classification, presented functional taxonomy of cloud services is the best of breed from existing taxonomies, includes up to date technologies and while being formal keeps place for upcoming technologies and solutions. The Author is very proud of it, while it is the best cloud taxonomy to this date.

Second one is focusing Business Process attributes, or the attributes of a set of Business Processes. It allows a Business Analyst to describe BP attributes both from descriptive perspective and from enactment perspective. This taxonomy is critical in spotting high level business requirements, which is utmost important for a well fit cloud solution architecture.

Third taxonomy is focusing on Enterprise Architecture attributes. This is a lower level of granularity after BP attributes, and it deals with IT from the company-wide perspective. By selecting EA QAs, an Analyst ensures that all the systems within the company are well aligned with new cloud service, all interfaces are compatible, data is accessible etc.

Fourth taxonomy is focusing on Quality of Service attributes. This is a well-known step in choosing any service. Again, in this taxonomy we gathered most of the important attributes for cloud services from respected literature sources.

Fifth taxonomy sets up specific architectural requirements to be implemented when adopting a service. It plays a prominent role in service selection, therefore this step is highly described in literature and there is an abundance of knowledge within consultancy companies. Current research presents the top of the iceberg while any consultancy can tailor it to a sector or to a specific cloud type or even to a client.

RsQ2. Linking the steps. What requirements are affected by other requirements in the model?

The proposed DSM model unites all five taxonomies to make a step by step model with predefined rules between steps. The model syntax is explicitly explained using EBNF notation, the database containing taxonomies has JSON format which is accessible by most of current programming languages. Interaction between trees is formed by a set of rules expressed in JavaScript. Not all the nodes can be modified due to the fact that the model has its pillars and is based on a rigor literature review. Still, minor modification may be made using a web-interface (which should be user-friendly, but is not yet fully developed).

Mentioned rules act as follows, e.g. if node A has a value of “Low”, then node B cannot have a value of “High”. Simply speaking, if a Business Process operates with top-secret data, and the attribute of its data privacy is “High”, then in further step of setting QoS attributes a user cannot select “Open (HTTP)” value for authentication attribute.

Making selections step by step a Business Analyst restricts more and more choices on the following steps. After finishing the procedure, the DSS tool exports a list of requirements that he selected. Then he can elaborate them, e.g. instead of “High” availability put “99.99 uptime”. After that, he can use this list as a base for a Request for Proposal to a cloud vendor.

RsQ3. Does the model work?

The theoretical concept (DSM) and a potentially developed tool based on it were validated in a series of expert interviews. All the 6 experts are working with cloud solutions either from academia or business perspective point of view, and their considerable experience insures a very high level of expertise. After the first three-four interviews the saturation was reached – all interviewees were stressing the same positive points. Romney et al. [85] calculated that samples as small as four individuals can render extremely accurate information with a high confidence level (.999) if they possess a high degree of competence for the domain of inquiry in question. Therefore, following [85] and [86], we can conclude that even such a small sample is sufficient in providing complete and accurate information while the participants possess a certain degree of expertise about the domain of inquiry.

All experts agreed that such a DSS would be useful in choosing a cloud service. Such a DSS might not speed the process up, but it will formalize it and make it more rigor. Or, if a rigorous process is a practice, then DSS will only speed it up. Most of the interviewees agreed that a standardized method is usually more effective, it ensures the result, increases the maturity of an organization, and that such a detailed RFP with all the requirements written down explicitly is an insurance both for a client and for a vendor. Mentioned interview were based on scientific concepts of measuring potential usefulness of a tool (UTAUT constructs).

To show that the process really works, the proof of concept is showing an imaginable company and in this case study, an Analyst follows all the steps to gather requirements.

As for the tool prototype, it was not fully developed, but the main functionality is shown as a workable software. The rest of the functionality is presented in the Thesis as a mockup of the tool.

7.1. Discussion of the results

It comes as no surprise that overall experts opinion is that the tool is perceived to be useful, and while it structures the process there is a positive effect on the process of choosing a cloud solution. It should be tailored per sector/company/client which should be easy using a proper interactive interface. Adopting such a DSS tool will increase effort/complexity/time spent, but as any formalized process will ensure the result and help in avoiding alterations. This only applies for companies that does not have a thorough requirements gathering procedure in place, otherwise the tool will only make this process faster. Also, an explicit logic behind the formulated requirements and a final vendor solution acts as an insurance both for Analyst, Vendor and a Client.

The most prevalent cloud computing myth is that a cloud service always saves money comparing to anything else. To actually save money, a cloud selection process should represent a thorough procedure that aligns business goals with potential benefits. Those goals and benefits are different in various use cases and should be the driving force for businesses, rather than any attempts to standardize on one offering or strategy [136]. To give an example, if hosting an application in the cloud (IaaS) instead of using a cloud service (SaaS) there are immediate benefits such as no need in hardware, but the company is still responsible for service functionality and maintainability. It can be labelled as a trade-off between full migration to a cloud and a half step which for many companies is the best first step to try out.

7.2. Further development of the concept

The prototype, when fully developed, may increase business maturity of any consultancy organisation, and also help cloud vendors to assess their offerings. Pursuing a career in this field, the Author plans to develop the fully-functional prototype for the company that he will be employed after Master Thesis defense. An important aspect that shall be mentioned is that the model is freely available for anyone, only consultancy-tailored versions with companies sets of rules will be restricted for free access while being a valuable company asset.

One of the interviewees introduced the idea of putting weight on some requirements so cloud vendors might better assess their tendering position when dealing with governmental institutes. This kind of DSM modification can also be a promising future development of the model. However, [49] argues that quantifying the basis for these decisions will likely be a unique activity for many businesses. Still, if those quantified decisions are developed per sector or tailored per governmental institute, benefits of such a tool are seen with a naked eye.

Another idea is to assign price and risk values to most of attributes, so it will be seen how the total price differs according to attributes selections and the tool also will act as a risk inventory tool.

Last but not least, the Author wants to formulate overall trend in cloud adoption for the nearest future. Almost everybody now is certain that a cloud service is more adaptable and easier to manage than any traditional one. Still, more issues arise such as where the companies critical data is stored, what rules and regulation apply to data centres, what are the provider guaranties on the service QoS, etc. Therefore, following the trend of cloud computing is not enough nowadays - IT professionals are now obliged to become cloud specialists as well. From now on, new trends are based on cloud computing. As new illumination techniques in our homes are based on electricity supply and electricity suppliers, - BPaaS solution, Cloud Brokers, household IoT devices, Fog computing systems, M2M interaction cars, smart grids, wearables – all these new trends are based on cloud technologies. Choosing a cloud solution will not be an option in nearest future, it will be by default while outsourcing hardware and massive computations is economically beneficial for any company – it is as simple as economy of scale. However, we now still need to become familiar with cloud opportunities, differences and potential migration issues. Therefore, a thorough cloud selection process such as presented DSM tool fits the demand at the right time.

References

[1] ISO/IEC/IEEE 42010, "Systems and software engineering - Architecture description," *International Organization for Standardization, Geneva*. p. 37, 2011.

[2] IEEE, "IEEE Recommended Practice for Architectural Description of Software Intensive Systems (IEEE Std. 1471-2000)," 2000.

[3] D. Minoli, *Enterprise architecture A to Z: frameworks, business process modeling, SOA, and infrastructure technology*. 2008.

[4] D. Howe, "'Quality of Service,'" *The Free On-line Dictionary of Computing*. 2015.

[5] W. Kim and S. Korea, "Cloud Computing: Today and Tomorrow," *Engineering*, vol. 8, no. 1, pp. 65–72, 2009.

[6] A. Lenk, M. Klems, J. Nimis, S. Tai, T. Sandholm, and P. Alto, "What ' s Inside the Cloud ? An Architectural Map of the Cloud Landscape," in *ICSE Workshop on Software Engineering Challenges of Cloud Computing*, 2009, pp. 23–31.

[7] L. Wang, G. Von Laszewski, M. Kunze, and J. Tao, "Cloud computing: A Perspective study," *New Gener. Comput.*, vol. 28, no. 2, pp. 137–146, 2010.

[8] D. M. Wink and Pallis G, "Cloud computing," *Internet Comput. IEEE*, vol. 14, no. 5, pp. 70–73, 2010.

[9] B. Young, D. Heney, and M. Dixon-Woods, "Cloud Computing," *Internet Protoc. J.*, vol. 12, no. 3, pp. 2–29, 2009.

[10] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: State-of-the-art and research challenges," *J. Internet Serv. Appl.*, vol. 1, pp. 7–18, 2010.

[11] M. Janakiram, *Demystifying the cloud*. e-Book, 2012.

[12] H. Yang and M. Tate, "A Descriptive Literature Review and Classification of Cloud Computing Research.," *Commun. Assoc. Inf. Syst.*, vol. 31, pp. 35–60, 2012.

[13] M. Rak and G. Aversano, "Benchmarks in the cloud: The mOSAIC benchmarking framework," in *4th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2012*, 2012, pp. 415–422.

[14] M. Mahjoub, A. Mdhaffar, R. B. Halima, and M. Jmaiel, "A Comparative Study of the Current Cloud Computing Technologies and Offers," *Netw. Cloud Comput. Appl. (NCCA), 2011 First Int. Symp.*, pp. 131–134, 2011.

[15] B. P. Rimal, A. Jukan, D. Katsaros, and Y. Goeleven, "Architectural Requirements for Cloud Computing Systems: An Enterprise Cloud Approach," *J. Grid Comput.*, vol. 9, pp. 3–26, 2011.

[16] J. Q. Anderson, "The Future of the Internet," *Internet Futur.*, vol. 5733, no. February, pp. 1–48, 2010.

[17] J. Repschlaeger, S. Wind, R. Zarnekow, and K. Turowski, “A reference guide to Cloud computing dimensions: Infrastructure as a service classification framework,” in *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2011, pp. 2178–2188.

[18] F. Polash, A. Abuhussein, and S. Shiva, “A Survey of Cloud Computing Taxonomies: Rationale and Overview,” in *Internet Technology And Secured Transactions, 2014 International Conference For IEEE., ICITST 2014*, 2014.

[19] P. T. Endo, G. E. Gonçalves, J. Kelner, and D. Sadok, “A Survey on Open-source Cloud Computing Solutions,” in *VIII Workshop em Clouds, Grids e Aplicações*, 2009, vol. 508, pp. 3–16.

[20] R. Esteves, “A Taxonomic Analysis of Cloud Computing,” *1st Dr. Work. Complex. Sci. ISCTE-IUL/FCUL*, pp. 1–12, 2011.

[21] H. Kamal Idrissi, A. Kartit, and M. El Marraki, “A taxonomy and survey of Cloud computing,” in *2013 National Security Days (JNS3)*, 2013, pp. 1–5.

[22] B. P. Rimal, E. Choi, and I. Lumb, “A taxonomy and survey of cloud computing systems,” in *NCM 2009 - 5th International Joint Conference on INC, IMS, and IDC*, 2009, pp. 44–51.

[23] R. Dukarić and M. B. Jurič, “A Taxonomy and Survey of Infrastructure-as-a-Service Systems,” *Lect. Notes Inf. Theory*, vol. 1, no. 1, pp. 29–33, 2013.

[24] S. Strauch, O. Kopp, F. Leymann, and T. Unger, “A taxonomy for cloud data hosting solutions,” *Proc. - IEEE 9th Int. Conf. Dependable, Auton. Secur. Comput. DASC 2011*, pp. 577–584, 2011.

[25] V. Jain and A. Sharma, “A Taxonomy on Cloud Computing,” *Int. J. Emerg. Technol. Adv. Eng.*, vol. 4, no. 3, pp. 149–153, 2014.

[26] B. Rimal, E. Choi, and I. Lumb, “A Taxonomy, Survey, and Issues of Cloud Computing Ecosystems,” in *Cloud Computing SE - 2*, N. Antonopoulos and L. Gillam, Eds. Springer London, 2010, pp. 21–46.

[27] C. N. Höfer and G. Karagiannis, “Cloud computing services: Taxonomy and comparison,” *J. Internet Serv. Appl.*, vol. 2, pp. 81–94, 2011.

[28] N. Antonopoulos and L. Gillam, *Cloud Computing: Principles, Systems and Applications*, vol. 54. 2010.

[29] W. Zeng, Y. Zhao, and J. Zeng, “Cloud service and service selection algorithm research,” in *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*, 2009, pp. 1045–1048.

[30] A. Milenkoski, A. Iosup, S. Kounev, K. Sachs, P. Rygielski, J. Ding, W. Cirne, and F. Rosenberg, “Cloud Usage Patterns: A Formalism for Description of Cloud Usage Scenarios,” 2014.

[31] B. Furht, A. Escalante, H. Jin, S. Ibrahim, T. Bell, W. Gao, D. Huang, and S. Wu, “Handbook of cloud computing,” in *Handbook of Cloud Computing*, 2010, pp. 3–19.

[32] A. Iosup, R. Prodan, and D. Epema, “IaaS Cloud Benchmarking : Approaches , Challenges , and Experience,” in *Cloud Computing for Data-Intensive Applications*, Springer New York, 2014, pp. 83–104.

[33] R. Ranjan, L. Zhao, X. Wu, and A. Liu, *Peer-to-Peer Cloud Provisioning: Service Discovery and Load-Balancing*. 2009.

[34] A. Kaur and J. Kaur, "Review of Cloud Computing Environment," *Int. J. Emerg. Trends Technol. Comput. Sci.*, vol. 3, no. 2, pp. 2–7, 2014.

[35] A. Reuther and J. Kepner, "Supercomputing: SaaS, PaaS, or IaaS?," in *High Performance Computing Modernization Program*, 2011, p. 526.

[36] C. N. Hoefer and G. Karagiannis, "Taxonomy of cloud computing services," *2010 IEEE Globecom Work.*, pp. 1345–1350, 2010.

[37] B. P. Rimal and E. Choi, "A service-oriented taxonomical spectrum, cloudy challenges and opportunities of cloud computing," *Int. J. Commun. Syst.*, vol. 25, no. 6, pp. 796–819, Jun. 2012.

[38] A. M. Bento and A. K. Aggarwal, *Cloud Computing Service and Deployment Models*. Hershey, PA: IGI Global, 2013.

[39] A. Mulholland, J. Pyke, and P. Fingar, "Enterprise Cloud Computing: A Strategy Guide for Business and Technology Leaders," Apr. 2010.

[40] F. van der Molen, *Get Ready for Cloud Computing: A Comprehensive Guide to Virtualization and Cloud Computing*. Van Haren Publishing, 2010.

[41] R. Mittal and K. Soni, "Analysis of Cloud Computing Architectures," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 2, no. 5, pp. 2087–2091, 2013.

[42] S. Kachele, C. Spann, F. J. Hauck, and J. Domaschka, "Beyond IaaS and PaaS: An Extended Cloud Taxonomy for Computation, Storage and Networking," in *2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing*, 2013, pp. 75–82.

[43] C. Weinhardt, A. Anandasivam, B. Blau, N. Borissov, T. Meinl, W. Michalk, and J. Stößer, "Cloud Computing – A Classification, Business Models, and Research Directions," *Bus. Inf. Syst. Eng.*, vol. 1, pp. 391–399, 2009.

[44] C. Binnig, D. Kossmann, T. Kraska, and S. Loesing, "How is the weather tomorrow? Towards a benchmark for the cloud," in *Proceedings of the Second International Workshop on Testing Database Systems*, 2009, pp. 91–96.

[45] L. Youseff, M. Butrico, and D. Da Silva, "Toward a unified ontology of cloud computing," in *Grid Computing Environments Workshop, GCE 2008*, 2008.

[46] T. F. Fortiş, V. I. Munteanu, and V. Negru, "Towards an ontology for cloud services," in *2012 6th International Conference on Complex, Intelligent, and Software Intensive Systems, CISIS 2012*, 2012, pp. 787–792.

[47] The Open Group, *TOGAF Version 9*. 2009.

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

[49] S. Kaisler, W. H. Money, and S. J. Cohen, "A Decision Framework for Cloud Computing," *25th Hawaii Int. Conf. Syst. Sci.*, pp. 1553–1562, 2012.

[50] I. Van De Weerd, S. Brinkkemper, J. Souer, and J. Versendaal, "A situational implementation method for web-based content management system-applications: method engineering and validation in practice," *Softw. Process Improv. Pract.*, vol. 11, no. 5, pp. 521–538, 2006.

[51] K. Vergidis, A. Tiwari, and B. Majeed, "Business Process Analysis and Optimization: Beyond Reengineering.," *IEEE Trans. Syst. Man, Cybern. Part C*, vol. 38, no. 1, pp. 69–82, 2008.

[52] R. Prodan and S. Ostermann, "A survey and taxonomy of infrastructure as a service and web hosting cloud providers," in *2009 10th IEEE/ACM International Conference on Grid Computing*, 2009, pp. 17–25.

[53] P. C. Heckel, "Hybrid Clouds: Comparing Cloud Toolkits," 2010.

[54] S. K. Garg, S. Versteeg, and R. Buyya, "A framework for ranking of cloud computing services," *Futur. Gener. Comput. Syst.*, vol. 29, no. 4, pp. 1012–1023, 2012.

[55] Z. Li, L. OBrien, R. Cai, and H. Zhang, "Towards a Taxonomy of Performance Evaluation of Commercial Cloud Services," in *2012 IEEE Fifth International Conference on Cloud Computing*, 2012, pp. 344–351.

[56] Z. Li, L. O'Brien, H. Zhang, and R. Cai, "On a catalogue of metrics for evaluating commercial cloud services," in *IEEE/ACM International Workshop on Grid Computing*, 2012, pp. 164–173.

[57] Z. Li, H. Zhang, L. O'Brien, R. Cai, and S. Flint, "On evaluating commercial Cloud services: A systematic review," *J. Syst. Softw.*, vol. 86, pp. 2371–2393, 2013.

[58] N. Lim, T. G. Lee, and S. G. Park, "A comparative analysis of enterprise architecture frameworks based on EA quality attributes," *10th ACIS Conf. Softw. Eng. Artif. Intell. Netw. Parallel/Distributed Comput. SNPD 2009, conjunction with IWEA 2009 WEACR 2009*, pp. 283–288, 2009.

[59] P. Narman, P. Johnson, and L. Nordstrom, "Enterprise Architecture: A Framework Supporting System Quality Analysis," in *11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007)*, 2007.

[60] E. Niemi and S. Pekkola, "Enterprise Architecture Quality Attributes: A Case Study," in *2013 46th Hawaii International Conference on System Sciences*, 2013, vol. 9, pp. 3878–3887.

[61] L. O'Brien, P. Merson, and L. Bass, "Quality Attributes and Service-Oriented Architectures," in *International Workshop on Systems Development in SOA Environments (SDSOA'07: ICSE Workshops 2007)*, 2007, pp. 3–3.

[62] M. Razavi Doudi, F. Schams Aliee, and K. Badie, "An AHP-based approach toward enterprise architecture analysis based on enterprise architecture quality attributes," *Knowl. Inf. Syst.*, vol. 28, pp. 449–472, 2011.

[63] J. Repschlaeger, R. Zarnekow, S. Wind, and T. Klaus, "Cloud Requirement Framework: Requirements and Evaluation Criteria to adopt Cloud Solutions," in *20th European Conference on Information Systems*, 2012, pp. 1–12.

[64] J. Siegel and J. Perdue, “Cloud services measures for global use: The Service Measurement Index (SMI),” in *Annual SRII Global Conference, SRII*, 2012, pp. 411–415.

[65] I. Voras, B. Mihaljevic, and M. Orlic, “Criteria for evaluation of open source cloud computing solutions,” *Proc. ITI 2011, 33rd Int. Conf. Inf. Technol. Interfaces*, pp. 137–142, 2011.

[66] E. Folkerts, A. Alexandrov, K. Sachs, A. Iosup, V. Markl, and C. Tosun, “Benchmarking in the cloud: What it should, can, and cannot be,” in *Selected Topics in Performance Evaluation and Benchmarking*, vol. 7755 LNCS, no. August, Springer Berlin Heidelberg, 2013, pp. 173–188.

[67] M. A. Chauhan and M. A. Babar, “Cloud Infrastructure for Providing Tools As a Service: Quality Attributes and Potential Solutions,” in *WICSA/ECSA 2012 Companion Volume*, 2012, pp. 5–13.

[68] M. Alhamad, T. Dillon, and E. Chang, “Conceptual SLA framework for cloud computing,” in *4th IEEE International Conference on Digital Ecosystems and Technologies - Conference Proceedings of IEEE-DEST 2010, DEST 2010*, 2010, pp. 606–610.

[69] G. Polya, “How to Solve It: A New Aspect of Mathematical Method,” *American Mathematical Monthly*, vol. 52. p. 575, 1945.

[70] P. B. Nielsen, “Towards a Definition of Business Functions,” in *UNSD/ESCWA Regional Seminar on International Trade Statistics*.

[71] M. Weske, *Business Process Management: Concepts, Languages, Architectures*, vol. 54. 2012.

[72] P. Mell and T. Grance, “The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology,” *Nist Spec. Publ.*, vol. 145, p. 7, 2011.

[73] D. Bernstein, “IEEE Intercloud Testbed: Techniical Overviiew and Engiineeriing Plan,” 2013.

[74] D. Bernstein and Y. Demchenko, “The IEEE intercloud testbed - Creating the global cloud of clouds,” in *Proceedings of the International Conference on Cloud Computing Technology and Science, CloudCom*, 2013, vol. 2, pp. 45–50.

[75] R. Buyya, R. Ranjan, and R. N. Calheiros, “InterCloud: Utility-oriented federation of cloud computing environments for scaling of application services,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2010, vol. 6081 LNCS, pp. 13–31.

[76] Y. Demchenko, M. X. Makkes, R. Strijkers, and C. De Laat, “Intercloud Architecture for interoperability and integration,” in *CloudCom 2012 - Proceedings: 2012 4th IEEE International Conference on Cloud Computing Technology and Science*, 2012, pp. 666–674.

[77] Y. Demchenko, C. Ngo, C. De Laat, J. A. Garcia-Espin, S. Figuerola, J. Rodriguez, L. M. Contreras, G. Landi, and N. Ciulli, “Intercloud architecture framework for heterogeneous cloud based infrastructure services provisioning on-demand,” in *Proceedings - 27th International Conference on Advanced Information Networking and Applications Workshops, WAINA 2013*, 2013, pp. 777–784.

[78] N. Grozev and R. Buyya, “Inter-Cloud architectures and application brokering: Taxonomy and survey,” *Softw. - Pract. Exp.*, vol. 44, pp. 369–390, 2014.

[79] A. R. Hevner, S. T. March, J. Park, and S. Ram, “Design Science in Information Systems Research,” *MIS Q.*, vol. 28, no. 1, pp. 75–105, 2004.

[80] A. Hevner and S. Chatterjee, “Design Science Research Frameworks,” in *Design Research in Information Systems*, vol. 22, 2010, pp. 23–31.

[81] H. Zhang, M. A. Babar, and P. Tell, “Identifying relevant studies in software engineering,” *Inf. Softw. Technol.*, vol. 53, pp. 625–637, 2011.

[82] Y. Levy and T. J. Ellis, “A Systems Approach to Conduct an Effective Literature Review in Support of Information Systems Research,” *Informing Sci. J.*, vol. 9, pp. 181–212, 2006.

[83] V. Venkatesh, M. G. Morris, G. B. Davis, and F. D. Davis, “User acceptance of information technology: Toward a unified view,” *MIS Q.*, vol. 27, no. 3, pp. 425–478, 2003.

[84] S. E. Baker and R. Edwards, “How many qualitative interviews is enough ?,” 2012.

[85] A. K. Romney, S. C. Weller, and W. H. Batchelder, “Culture as Consensus: A Theory of Culture and Informant Accuracy,” *Am. Anthropol.*, vol. 88, no. 2, pp. 313–338, 1986.

[86] G. Guest, A. Bunce, and L. Johnson, “How Many Interviews Are Enough?: An Experiment with Data Saturation and Variability,” *Field methods*, vol. 18, no. 1, pp. 59–82, 2006.

[87] A. Fontana and H. F. Wolcott, “The Art of Fieldwork,” *Contemporary Sociology*, vol. 28, no. 5. p. 632, 1999.

[88] F. Liu, J. Tong, J. Mao, R. Bohn, J. Messina, L. Badger, and D. Leaf, “NIST Cloud Computing Reference Architecture Recommendations of the National Institute of Standards and,” *Nist Spec. Publ.*, vol. 292, no. 9, p. 35, 2011.

[89] Intel Corporation, “Cloud Computing Taxonomy and Ecosystem Analysis,” *Intel Inf. Technol.*, 2010.

[90] A. Beloglazov, R. Buyya, Y. C. Lee, and A. Zomaya, *A Taxonomy and Survey of Energy-Efficient Data Centers and Cloud Computing Systems*, vol. 82. 2011.

[91] S. Ried, H. Kisker, and P. Matzke, “www.forrester.com/go?docid=57232,” 2010.

[92] OpenCrowd, “Cloud Computing Vendors Taxonomy.” [Online]. Available: http://cloudtaxonomy.opencrowd.com/. [Accessed: 16-May-2015].

[93] L. Colombo-Mendoza and M. Salas-Zárate, “Cloud Computing: A review of PaaS, IaaS, SaaS Service and Providers,” *Ed. Responsab. las Fac. Ing. Frente al Desarro. del País*, pp. 47–57, 2012.

[94] D. Hilley, “Cloud Computing : A Taxonomy of Platform and Infrastructure-level Offerings,” *Technology*, no. April, pp. 1–37, 2009.

[95] M. Heller, “MBaaS shoot-out: 5 clouds for building mobile apps,” *InfoWorld*, 2014. [Online]. Available: http://www.infoworld.com/article/2842791/application-development/mbaas-shoot-out-5-cloud-platforms-for-building-mobile-apps.html.

[96] J. Elizondo and S. Palmer, "Edge-Based Cloud Computing as a Feasible Network Paradigm," Austin, 2009.

[97] Cisco Systems Inc., "Cisco RFP-2013-078. Fog Computing, Ecosystem, Architecture and Applications." .

[98] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog Computing and Its Role in the Internet of Things," *Proc. first Ed. MCC Work. Mob. cloud Comput.*, pp. 13–16, 2012.

[99] J. Apostolopoulos, "FOG Computing," 2014.

[100] Cisco Systems Inc., "The Internet of Everything." [Online]. Available: http://www.cisco.com/c/r/en/us/internet-of-everything-ioe/index.html. [Accessed: 05-Mar-2015].

[101] M. Research, "The Machina Research M2M Forecast Database." [Online]. Available: https://machinaresearch.com/what-we-do/advisory-service/m2m-forecast-database/. [Accessed: 05-Mar-2015].

[102] R. Heinrich, A. Kappe, and B. Paech, "Modeling Quality Information within Business Process Models."

[103] M. Lohrmann and M. Reichert, "Understanding Business Process Quality," *Bus. Process Manag.*, vol. 444, pp. 41–73, 2013.

[104] R. S. Aguilar-Savén, "Business process modelling: Review and framework," *Int. J. Prod. Econ.*, vol. 90, no. 2, pp. 129–149, 2004.

[105] K. Kundeliene, "The assessment of business processes accounting quality attributes," Kaunas University of Technology, 2009.

[106] P. Eeles, "Capturing Architectural Requirements," *http://www.ibm.com/developerworks/rational/library/4706.html*, 2005. [Online]. Available: http://www.ibm.com/developerworks/rational/library/4706.html.

[107] J. Tash, "Enterprise Architecture vs. Technology Architecture," *Flashmap Systems, Inc.*, 2006. [Online]. Available: http://www.flashmapsystems.com/resources/articles/articles29.htm. [Accessed: 23-Jun-2015].

[108] A. Rodríguez, A. Caro, C. Cappiello, and I. Caballero, "A BPMN extension for including data quality requirements in business process modeling," *Lect. Notes Bus. Inf. Process.*, vol. 125 LNBIP, pp. 116–125, 2012.

[109] E. Eltikangas, "Enterprise Architecture Documentation and Representation," Helsinki University of Technology, 2006.

[110] J. A. McCall, P. K. Richards, and G. F. Walters, "Factors in Software Quality," *Nat'l Tech. Inf. Serv.*, vol. 1, 2 and 3, no. ADA049055, 1977.

[111] B. W. Boehm, J. R. Brown, and H. Kaspar, *Characteristics of software quality*. Amsterdam, 1978.

[112] M. Joseph, "Software metrics: Establishing a company-wide program," *Information and Software Technology*, vol. 30, no. 1. p. 58, 1988.

[113] International Organization for Standardization and International Electrotechnical Commission, "ISO/IEC 9126-1 Software engineering - Product quality - Part 1: Quality model," 2001.

[114] T. Baars and M. Spruit, "The SeCA Model," in *Security Engineering for Cloud Computing: Approaches and Tools*, D. G. Rosado, D. Mellado, E. Fernandez-Medina, and M. G. Piattini, Eds. Hershey, PA: IGI Global, 2013, p. 257.

[115] A. Abuhussein, H. Bedi, and S. Shiva, "Towards a Stakeholder-Oriented Taxonomical Approach for Secure Cloud Computing," in *2013 IEEE Sixth International Conference on Cloud Computing*, 2013, pp. 958–959.

[116] D. Firesmith, "Specifying reusable security requirements," *Journal of Object Technology*, vol. 3, no. 1. pp. 61–75, 2004.

[117] N. Gonzalez, C. Miers, F. Redígolo, M. Simplício, T. Carvalho, M. Näslund, and M. Pourzandi, "A quantitative analysis of current security concerns and solutions for cloud computing," *J. Cloud Comput. Adv. Syst. Appl.*, vol. 1, no. 1, p. 11, 2012.

[118] N. Gruschka and M. Jensen, "Attack surfaces: A taxonomy for attacks on cloud services," in *Proceedings - 2010 IEEE 3rd International Conference on Cloud Computing, CLOUD 2010*, 2010, pp. 276–279.

[119] M. Habib, S. M., Varadharajan, V., Muhlhauser, "A Trust-aware Framework for Evaluating Security Controls of Service Providers in Cloud Marketplaces," in *12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2013, pp. 459–468.

[120] I. Iankoulova and M. Daneva, "Cloud computing security requirements: A systematic review," in *2012 Sixth International Conference on Research Challenges in Information Science (RCIS)*, 2012, pp. 1–7.

[121] H. Wang and C. Wang, "Taxonomy of security considerations and software quality," *Communications of the ACM*, vol. 46, no. 6. pp. 75–78, 2003.

[122] D. Chen, B. Vallespir, and N. Daclin, "An approach for enterprise interoperability measurement," in *CEUR Workshop Proceedings*, 2008, vol. 341, pp. 1–12.

[123] N. Daclin, D. Chen, and B. Vallespir, "Enterprise interoperability measurement - Basic concepts," *Proc. EMOI*, no. 1, pp. 1–5, 2006.

[124] R. Teckelmann, C. Reich, and A. Sulistio, "Mapping of cloud standards to the taxonomy of interoperability in IaaS," in *3rd IEEE International Conference on Cloud Computing Technology and Science, CloudCom 2011*, 2011, pp. 522–526.

[125] W. Dawoud, I. Takouna, and C. Meinel, "Infrastructure as a service security: Challenges and solutions," *Informatics Syst. (INFOS), 2010 7th Int. Conf.*, 2010.

[126] S. Bertram, M. Boniface, M. Surridge, N. Briscombe, and M. Hall-May, “On-demand dynamic security for risk-based secure collaboration in clouds,” in *Proceedings - 2010 IEEE 3rd International Conference on Cloud Computing, CLOUD 2010*, 2010, pp. 518–525.

[127] B. Gowrigolla, S. Sivaji, and M. R. Masillamani, “Design and auditing of Cloud computing security,” in *Proceedings of the 2010 5th International Conference on Information and Automation for Sustainability, ICIAfS 2010*, 2010, pp. 292–297.

[128] G. Galante and L. C. E. De Bona, “A survey on cloud computing elasticity,” in *Proceedings - 2012 IEEE/ACM 5th International Conference on Utility and Cloud Computing, UCC 2012*, 2012, pp. 263–270.

[129] S. Yi, A. Andrzejak, and D. Kondo, “Monetary cost-aware checkpointing and migration on amazon cloud spot instances,” *IEEE Trans. Serv. Comput.*, vol. 5, no. 4, pp. 512–524, 2012.

[130] N. J. Kansal and I. Chana, “Cloud Load Balancing Techniques : A Step Towards Green Computing,” *IJCSI Int. J. Comput. Sci. Issues*, vol. 9, no. 1, pp. 238–246, 2012.

[131] K. R. Deepitha and G. Animesh, “Ensuring Correctness and Error Localisation In Cloud,” *Int. J. Comput. Sci. Inf. Technol.*, vol. 4, no. 4, pp. 621–625, 2013.

[132] H. Benko and a Wilson, “DepthTouch: Using depth-sensing camera to enable freehand interactions on and above the interactive surface,” *... Tabletops Interact. Surfaces*, no. March, 2009.

[133] C. Radich, D. Christiansen, D. Zhang, and I. Donald, “Federal CIO : Cloud Selection Toolkit,” 2011.

[134] G. Reese, *Cloud application architectures*. Sebastopol, Calif.: O’Reilly, 2009.

[135] V. Stantchev and C. Schröpfer, “Negotiating and enforcing QoS and SLAs in grid and cloud computing,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2009, vol. 5529, pp. 25–35.

[136] D. M. Smith, “The Top 10 Cloud Myths,” 2014.

Appendices

Appendix A SLR Procedures

Appendix A.1 SLR of BP attributes

This section's SLR procedure started with the same approach as the one described in Chapter **Error! Reference source not found.**. However, after screening of first results, it became clear that there is much less sources to start with. Thus, in this SLR the main work is done by reference snowballing from few found relevant articles.

Appendix A.1.1 Identification

The following Google Scholar query yielded a total of 306,000 results:

['business process' OR 'business process model' OR 'business process diagram' OR 'business function' OR 'business process modelling' OR 'bpmn' OR 'epc' OR 'uml' OR 'bpel'] AND ['quality information' OR 'quality characteristics' OR 'quality requirements' OR 'quality aspects' OR 'quality properties' OR 'quality attributes' OR 'constraints' OR 'requirements' OR 'non-functional requirements' OR 'nfr' OR 'business rules' OR 'quality metrics' OR 'quality measures'].

Appendix A.1.2 Screening

Most of the articles deal with business process modelling technique or a business process model quality. And yet, we need quality attributes of a business process, not its model or modelling techniques.

Based on the titles screening only a few were deemed interesting for this research. These are:

- Article by Heravizadeh et al. entitled "Dimensions of business processes quality (QoBP)", 2009;
- Article by Heinrich et al. entitled "Modeling Quality Information within Business Process Models", 2011;
- Technical Report by Demirors et al. entitled "A Model for Using Software Quality Characteristic to Measure Business Process Quality", 2005.

Appendix A.1.3 Reference snowballing

After following the references from those articles, a few articles were found that broadened the field. Rodríguez et al. [A BPMN Extension for Including Data Quality Requirements in Business Process Modeling] and Vergidis et al. [Business Process Analysis and Optimization: Beyond Reengineering] were citing the most relevant sources of Business Process Quality Attributes in BP modelling. Thus, the reference snowballing came to 13 articles more addressing the topic from different perspectives – from business rules and requirements to BPMN and EPC notations extensions to support quality requirements.

Appendix A.1.4 Eligibility

In order to assess the eligibility of the remaining 106 articles, the full texts of these articles were screened. The articles are divided into four different categories:

1. Addressing cloud solutions taxonomies;
2. Too specific, e.g. addressing BPMN techniques. To some extent they are useful, e.g. when describing extensions of BPMN with certain QoS attributes. Those were stored for further use in next chapters if needed, but not included here.
3. Too broad, e.g. addressing management issues. Those were excluded from the review.
4. Not relevant, e.g. *how to* process business rules etc.

Appendix A.1.5 Included

A total of 22 articles were included in the literature review on cloud solutions taxonomy.

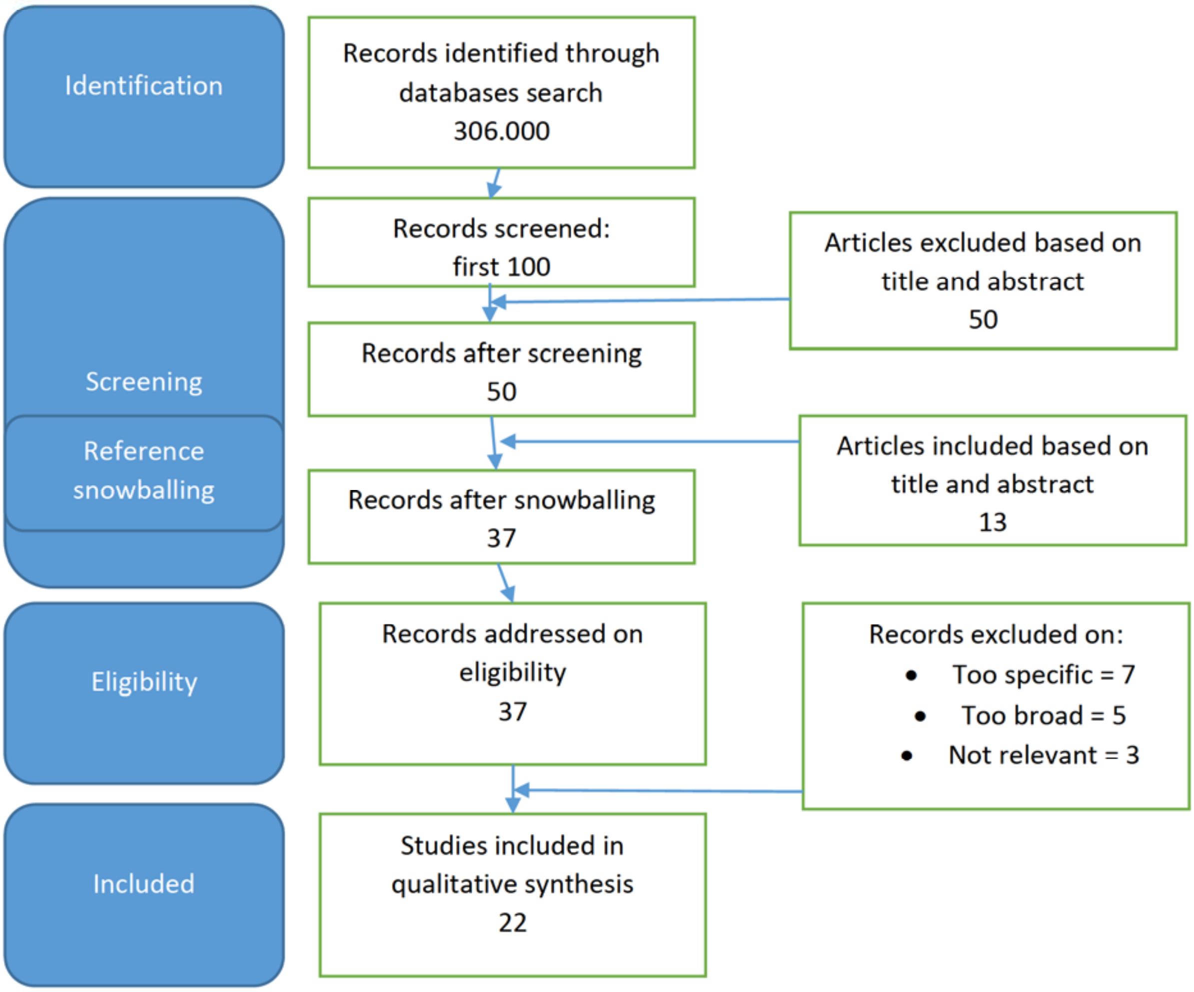


Figure 79 Four phases of RQs1.1 SLR

Appendix A.2 SLR of EA QAs

This section's SLR procedure started with the same approach as the one described in Chapter **Error! Reference source not found.**. However, after screening of first results, it became clear that there is much less sources to start with. Thus, in this SLR the main work is done by reference snowballing from few found relevant articles.

Appendix A.2.1 Identification

Search keys shown in Table 5 yielded a total of 1,020,000 articles (including duplicates).

Enterprise Architecture EA Service Oriented Architecture SOA TOGAF	**x**	Quality attributes Quality	**= 40 search keys**

Table 5 RsQ1.2 search keys

The following search query: ("Enterprise architecture" OR EA OR "Service Oriented Architecture" OR SOA OR TOGAF) AND ("Quality Attributes" OR Quality) = 1,020,000 in Google Scholar from 2006 until 2015.

Appendix A.2.2 Screening

Mentioned query did not provide sufficient material needed to structure EA QAs, but it did yield the main search streams for reference snowballing. These are:

- Article by Lim et al. entitled "A Comparative Analysis of EA Frameworks based on EA QAs", 2009;
- TOGAF by Open Group related papers;
- "The Integrated Architecture Framework" by Capgemini related papers;
- Official Enterprise Architecture Standards e.g. ISO 15704, IEEE 1471–2000, ISA-95.00.01-2000, ENV 13350, IS 15704;
- The Zachman Architectural Framework related papers;
- Service-Oriented Architecture modelling extensions with QAs.

Appendix A.2.3 Reference snowballing

Abovementioned research directions were searched for QAs extensions. Thus, the reference snowballing came to 20 articles more addressing the topic from different standards' perspectives.

Appendix A.2.4 Eligibility

In order to assess the eligibility of the remaining 106 articles, the full texts of these articles were screened. The articles are divided into two categories:

1. Addressing certain EA framework QAs;
2. Not relevant.

Appendix A.2.5 Included

A total of 17 articles were included in the literature review on cloud solutions taxonomy.

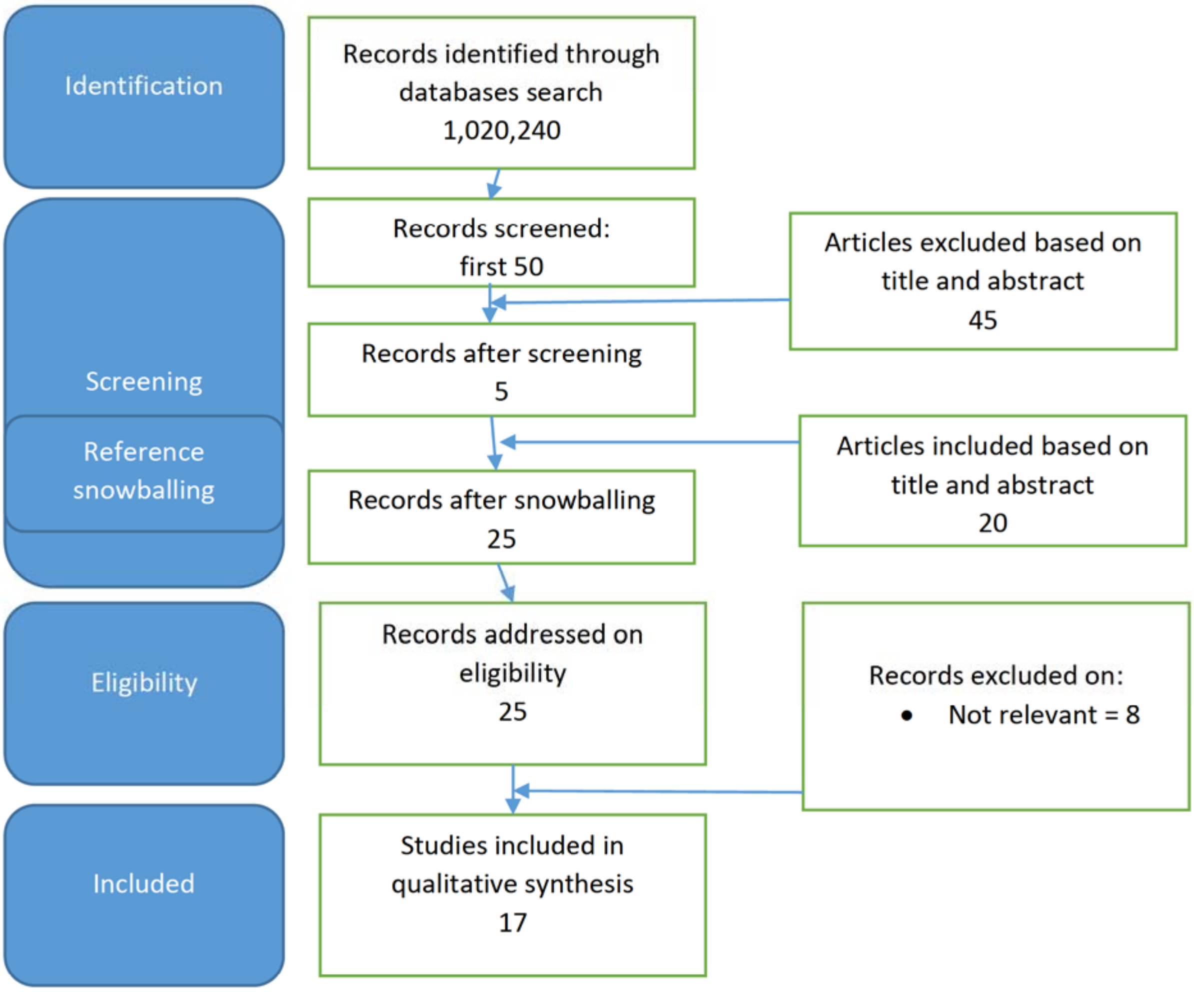


Figure 80 Four phases of RQs1.2 SLR

Appendix A.3 SLR of QoS requirements

Appendix A.3.1 Identification

Search keys shown in Table 6 yielded a total of 948,000 articles (including duplicates).

Cloud solution		Quality Attributes		
Cloud		Service Quality Attributes		
Cloud service	**x**	Quality of Service	**=**	**25 search keys**
Cloud computing		QoS		
Cloud application		Quality		

Table 6 RsQ1.3 search keys

The following search query: ("cloud solution" OR "cloud service" OR "cloud computing" OR "Cloud application" OR Cloud) AND ("Quality Attributes" OR "Service Quality Attributes" OR "Quality of Service" OR QoS OR Quality) = 948,000 in Google Scholar from 2006 until 2015.

Appendix A.3.2 Screening

Of the roughly 1 million results, about 600 records were screened. This is done by reading the titles and when deemed possibly relevant for this research, also the abstract. Per relevant keyword the first 10 to 40 results were searched through until nothing close to a good match had been found.

Search keys:			Results:	Irrelevant after:
Cloud solution	x Quality Attributes	=	21100	30
	Service Quality Attributes	=	15700	30
	Quality of Service	=	30700	40
	QoS	=	16200	10
	Quality	=	145000	40
Cloud service	x Quality Attributes	=	26100	20
	Service Quality Attributes	=	21600	10
	Quality of Service	=	81700	30
	QoS	=	17500	10
	Quality	=	319000	20
Cloud computing	x Quality Attributes	=	19900	30
	Service Quality Attributes	=	15700	30
	Quality of Service	=	29800	10
	QoS	=	16600	10
	Quality	=	208000	10
Cloud application	x Quality Attributes	=	25400	40
	Service Quality Attributes	=	20400	40
	Quality of Service	=	40600	30
	QoS	=	16800	30
	Quality	=	554000	20
Cloud	x Quality Attributes	=	50000	30
	Service Quality Attributes	=	18100	30
	Quality of Service	=	79200	20
	QoS	=	20300	20
	Quality	=	813000	10

Table 7 RsQ1.3 screening procedure

Based on the title and abstract screening of these 600 results, 24 article are deemed interesting for this research. Consequently, 576 screened articles are deemed not interesting enough and are dismissed.

Appendix A.3.3 Reference snowballing

In turn, backward (and then again forward) snowballing on more specific cloud deployment models yielded 10 more articles more, which extended the model in specific aspects. Mainly, they are ISO standards and QoS frameworks.

Appendix A.3.4 Eligibility

In order to assess the eligibility of the remaining 34 articles, the full texts of these articles were screened. Except for irrelevant articles, all others were included in SLR. Some papers are addressing specific aspects e.g. security QoS attributes or IaaS QoS attributes, and they were included in SLR as well.

Appendix A.3.5 Included

A total of 31 articles were included in the literature review on cloud solutions taxonomy.

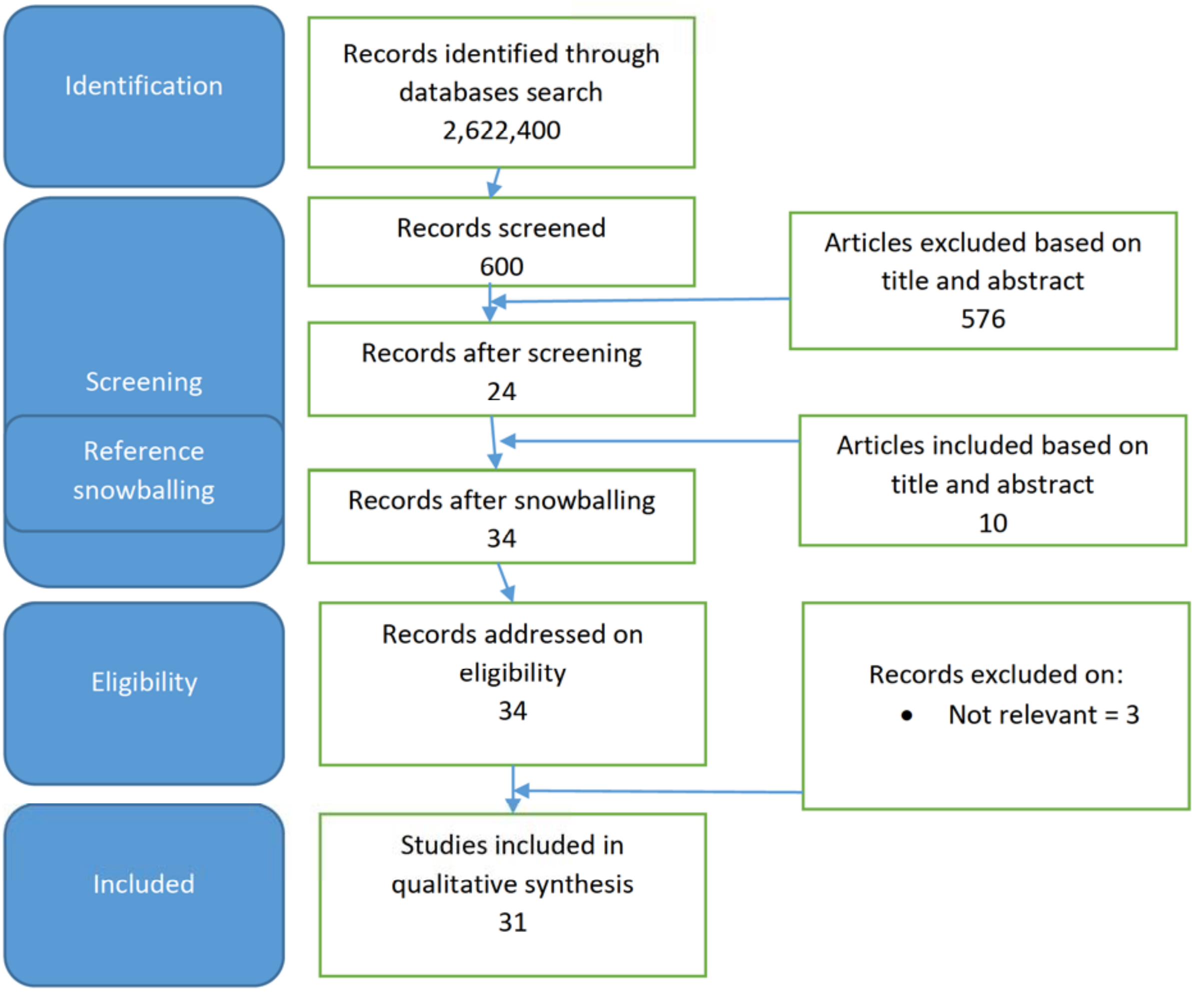


Figure 81 Four phases of RsQ1.3 SLR

Appendix A.4 SLR of Architectural requirements

Appendix A.4.1 Identification

Search keys shown in Table 2 yielded a total of 388,056 articles (including duplicates).

Cloud solution Cloud service Cloud computing	**x**	Deployment Architecture	**x**	Model Tactics		**= 12 search keys**

Table 8 RsQ1.5 search keys

Example search query: ("cloud solutions" OR "cloud service" OR "cloud computing" OR "cloud computing") AND (deployment OR architecture) AND (models OR tactics) = 15,500 in Google Scholar from 2006 until 2015.

Appendix A.4.2 Screening

Of the roughly 400 thousands results, about 470 records were screened. This is done by reading the titles and when deemed possibly relevant for this research, also the abstract. Per relevant keyword the first 20 to 50 results were searched through until nothing close to a good match had been found.

Search keys:							Results:	Irrelevant after:
Cloud solution	x	Deployment	x	Model	=		18100	50
				Tactics	=		16100	40
Cloud service	x	Deployment	x	Model	=		20800	30
				Tactics	=		15500	30
Cloud computing	x	Deployment	x	Model	=		16200	50
				Tactics	=		14200	20
Cloud solution	x	Architecture	x	Model	=		33100	90
				Tactics	=		10600	30
Cloud service	x	Architecture	x	Model	=		56800	30
				Tactics	=		16100	40
Cloud computing	x	Architecture	x	Model	=		66700	30
				Tactics	=		15000	30

Table 9 RsQ1.5 screening procedure

Based on the title and abstract screening of these 470 results only 26 articles were deemed interesting for this research. Consequently, 444 screened articles are deemed not interesting enough and are dismissed.

Appendix A.4.3 Reference snowballing

In turn, backward (and then again forward) snowballing on more specific cloud deployment models yielded 34 articles more, which substantially extended the model in specific aspects.

Appendix A.4.4 Eligibility

In order to assess the eligibility of the remaining 60 articles, the full texts of these articles were screened. The articles are divided into four different categories:

1. Addressing cloud solutions architectural tactics;
2. Too specific, e.g. addressing types of load balancing algorithms;
3. Too broad and obvious, e.g. explaining three main deployment models;
4. Not relevant, e.g. addressing manufacturing or science applications.

Appendix A.4.5 Included

A total of 50 articles were included in the literature review on cloud solutions taxonomy.

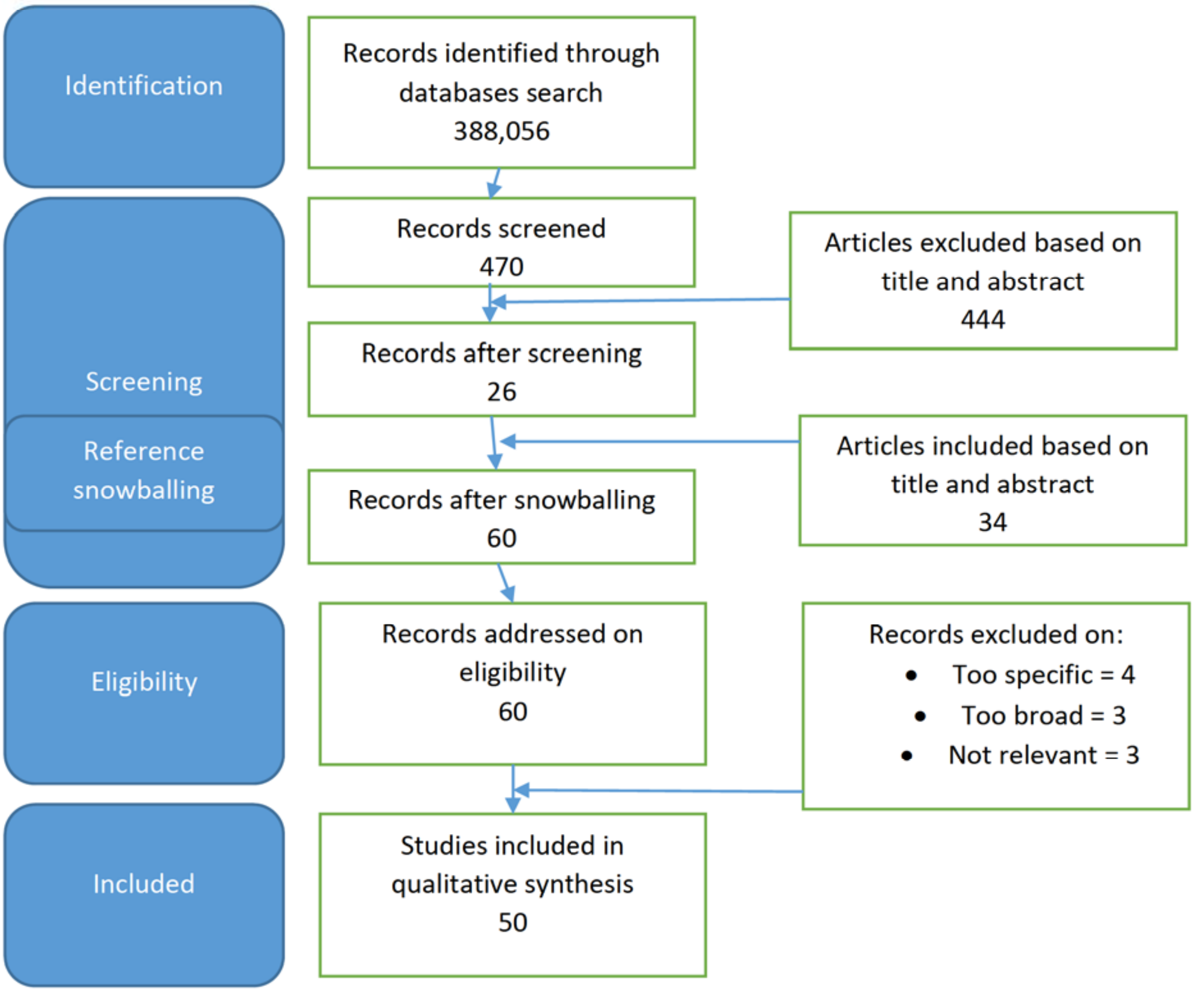


Figure 82 Four phases of RsQ1.5 SLR

Appendix D List of experts

Interviewee 1

Former business Analyst and IT Consultant, is now working at Higher School of Economics in Moscow. While being in a program committee of 2015 Third International Conference on Enterprise Systems, he encountered several works on the close topic. Both business background and academic research make him an ideal candidate for the concept evaluation.

Interviewee 2

He is a consultant at one of the largest professional auditor companies in the world. Currently is working in Shared Services and Outsourcing Advisory department dealing with clients across Europe. Four years of experience with clients from different European countries make his view wide, and therefore the contribution to the research - very valuable.

Interviewee 3

He spent several years working at the biggest software company in CIS countries (1C) as a developer, he is now working at the biggest integrator of 1C products in Russia. Since last two years, main 1C products are cloud solutions. Day to day integration of cloud software in different client-companies and excellent knowledge of the principles behind them make him an ideal candidate for the concept evaluation.

Interviewee 4

Being IT Consultant and Software Engineer for more than five years he is able to testify if the proposed model is going to be useful for his job. He also can provide two different points of view due to his dualistic background – technical view and more general business view.

Interviewee 5

After being a Senior Software Developer for 10 years, 4 years ago he started to work as Chief Technology Officer in a company integrating tendering software. The nature of these products is very alike to the proposed DSM. Moreover, his company is rearchitecting their products to being cloud-based applications.

Interviewee 6

During his carrier, he was a consultant for more than 5 years, Senior Manager for 8 years, and then started to work in more cloud-specific sector. Cloud computing consulting for 10 years, coaching on cloud security, cloud architecture and cloud business, and being a founder and chairman of Dutch Computer Measurement Group division. His experience makes him on of the most appropriate interviewees for our topic in Europe.

Appendix E Interview notes

Appendix E.1 Interview 1

Interview duration: 36 minutes.

Overall: positive. DSS promises to be useful because it will structure the process (and formalization makes a company more mature and efficient).

Relevant points:

U6. Using such DSS would be useful in for choosing a cloud service.

U1,5. While it structures the process, there is a positive effect in the process of choosing a cloud solution.

U2,3,4. For a starting professional it may increase job performance.

JF5. As any formalized/structured process, it increases the maturity of organization as whole.

OE2. The time spent on requirements gathering may increase, but the number of alterations will decrease.

EOU1. Briefing is necessary to learn how to operate the tool. 2-3 days seminar and an audit afterwards will settle the tool as a company-wide practice.

EOU4,6. A concrete methodology is needed to define what can be changed in the model, and what are the pillars that should not change.

CO3. Using such DSS might involve more time as usual in the beginning as any new procedure adopted.

3. Outsourcing can be analyzed with the help of similar concept

The process of choosing a service is described well and can be used in educational purposes.

Appendix E.2 Interview 2

Interview duration: 50 minutes.

Overall: positive. Standardised method ensures the result More effort increases quality of the result. Formal procedure helps to transfer knowledge within the company and supports upcoming experts. DSS should be tailored to industry sectors. An interactive interface is better for a user, language change should be as well.

Relevant points:

U6. Such a DSS would be useful in for choosing a cloud service.

U1,5. Such a DSS might not speed the process up, but it will formalize it, perhaps make it easier by suggesting requirements.

U2,3,4. Standardized method is usually more effective. What is most important, is that a standardised method ensures the result.

JF5. The amount of effort put into following the prescribed steps increases, but traceability of requirements gathering process also increases – and that is more important. More effort increases quality of the result.

JF6. Knowledge is needed to be able to use it. However, with adoption of such a formalized approach, it is easier to transfer knowledge and less experienced Analysts are insured. Formal procedure supports upcoming experts.

OE2. No, I think it will consume more time because you are following a thorough procedure.

EOU1. A person who has an experience in using this DSS should walk through new users. Important thing is to have an expert that is able to explain the process to others.

EOU2. For a provider received an elaborate RFP it is easy to go through requirements and conclude whether it can fulfil them or not. Still, from a client perspective, it is not easy to choose a cloud based on this list – it is a list for a professional e.g. Consultant. Such a detailed RFP with all the requirements written down explicitly is an insurance both for a client and for a vendor. It is also easier for a vendor to have a formal list of specs.

EOU4,6. It is flexible, but how easy it is? An interactive interface is better for a user (language). Standard core that cannot be changed, but the rest should be customizable – 10% is the degree of change, not more.

CO1. Using the system takes too much time from normal duties? It is part of the duties, so it is ok to use a method and spend time on it as soon as the result meets expectations.

CO2. Working with such DSS might be so complicated, that is difficult to understand what is going on? It depends on the knowledge – rules and taxonomies require knowledge. And it is not a drawback while DSS tool is not a black box, but just a support tool.

CO3. Using such DSS might involve too much time doing mechanical operations (e.g., data input). It is possible to share data input with a client - surveys for data collection, etc. Moreover, double-checking takes time – an Analyst should always verify his assumptions with a client.

3. IT, logistics, any RFP generating tools can use the same concept.

Governmental organizations, NGOs etc. have specific prescribed rules for tendering software – if a DSS can put weights on the requirements, it will be a great tool for specifying a complete RFP with weightings for a tender. Thus the vendors will have sufficient information for applying to it.

The DSS can be used for an analysis of the system in place in organization. With doing that it is possible to see whether it is possible to add new functionality or compare a legacy system with a new one, or non-cloud to a cloud one etc.

Appendix E.3 Interview 3

Interview duration: 63 minutes.

Overall: skeptical. Such a tool cannot be used without many modifications to each client, while they will take too much time for an expert who doesn't need the support tool or will be too difficult for a newbie. With a proper interface, DSS might be useful as a checklist guiding experts and ensuring that BP and EA steps are not missed.

Relevant points:

- 30:00 For a beginner, the system is too complicated (requires a lot of knowledge). For a high-level expert, it is a waste of time.

- Glossary – different meanings between client and experts (between experts – ok). The terms used in the taxonomies don't mean a lot to clients usually, but having a glossary to align the experts would be great. The meaning put in these terms by BA is not the same as the client perceive. But, these terms are a good anchor between experts when they want to demonstrate the logic they use to each other.
- 35:03 If the tool doesn't suggest anything (specific solutions) – it's good, so people will use their own head to think
- 39:00 Model is useful, but it is not sufficient. Having such a checklist doesn't exclude all the work that should be done like before having the checklist. It only helps in saving time sometimes.
- 49:00 All DSS systems are always like this, it is logical. One tricky thing is that one step is dependent on the other – not everyone is going to use that. As a checklist with five levels it is quite convenient.
- It is easy to understand how to modify the tool. If there will be a nice interface, the tool will be usable.
- 57:40 The DSS tool will be very useful if BP and EA trees will be highly and easily modifiable and will be tailored to specific company completely. Only tree number 3 is universal, the first two should be at least modified per sector. Still, it is impossible to formalize unique competitive advantage of each company.

Question 3. It is ok as a checklist, but better is a process of making a checklist (the method/system according to which the model can change)

Appendix E.4 Interview 4

Interview duration: 34 minutes.

Overall: positive but argues the scope. The model too general to support the full process. It is useful for business people (consultants), but there is still a gap between them and technical people. And choosing a particular solution is technical, and here it doesn't help. However, he doesn't say that the DSS can handle all the process – quite in opposite, there are too any specific technical details at the later phases. Still, he doesn't like that there is no solution to the gap between technical level and abstract business level. Perhaps, bridging them at least for some industry sectors might work.

Relevant points:

The tool will be useful but only for high level consultants. It covers the general approach, but does not go deep into specifics. There is always a gap between high-level part and low-level part – people from business and technicians. This tool doesn't solve that problem, and after these high level requirements are gathered, developers have to see whether the solution is feasible.

Good tip - loop the model after the provider selection. It means, that after the requirements are formulated, and the vendor offers a solution, the Analyst should go through the requirements again to see whether the solution meets them indeed.

Appendix E.5 Interview 5

Interview duration: 59 minutes.

Overall: positive. Smart rules will make the process of going through the model even quicker. Productivity will increase because an Analyst won't redo the requirements gathering cycle Several times, but will just follow this thorough procedure once (maybe with some minor follow up questions). Writing a proposal also takes a lot of time, so this will not make it more – for complete proposal.

Relevant points:

U6. 27:50 yes, when a lot of requirements are there, it is easy to miss something important e.g. compatibility with other systems.

U1,5. More quick – no, it but will ensure requirements are complete. Also smart relationships will make it quicker.

U2,3,4. 29:10 for sure – productivity will increase because he won't redo the requirements gathering cycle 10 times, but will just follow this thorough procedure once (maybe with some minor follow up questions).

JF5. Complete picture faster

OE2. Less iterations of requirements gathering cycle.

EOU1. Learning could be problematic - a glossary is needed, terms could be interpreted differently.

EOU4,6. 40:30 yes, it is very wide and can be used (if a bit modified) in any business case

CO1. Writing a proposal also takes a lot of time, so this will not make it more – for complete proposal. If a goal is to solve a partial problem, then yes – it will consume more time than needed.

CO2. Working with such DSS might be so complicated, that is difficult to understand what is going on.

CO3. No, you need to type anyway.

Question 3. Tighten attributes to price, so it will be seen how the total price differs according to attributes selections.

Appendix E.6 Interview 5

Interview duration: 37 minutes.

Overall: positive. In the same way as all previous interviewees he said that the DSM is a good structured checklist and having such procedure as an adopted practice makes the company more efficient.

Relevant points:

- If different consultants can collaborate on one requirements document – that would unite people working on one project from different units e.g. functional designers, developers, security specialists.
- Risk inventory tool – every feature is also a risk, and most of attributes can be linked to risk levels.

Appendix G Case study

Marketing & Sales System adoption case study

Background

Consider a fictitious organization, ABC with a not-so-fictitious scenario. The company was found in 2008 in City and specializes in software development.

ABC is a part of a group of companies called "Automation Technology". Areas of activity of this group of companies are:

- development of automated systems for businesses;
- working with governmental agencies;
- delivery of computers, specialized machinery and equipment;
- IT outsourcing and consulting.

The main activities of ABC are:

- development, implementation and technical support of information systems for companies and organizations in different sectors;
- supply of industrial computer technology;
- IT - outsourcing.

ABC is a developer of software product line for environmental situation monitoring. The product line includes: software for environmental monitoring "Agate", weather control software "Amber", and laboratory information software "Onyx".

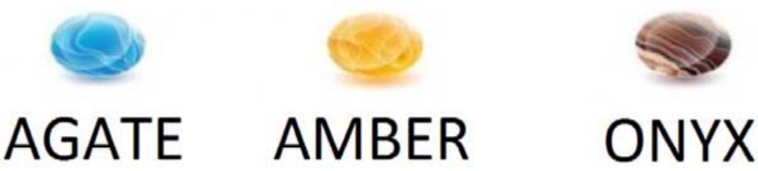


Figure 83 ABS software products line

The flagship product of the company is the environmental monitoring system "Agate", which is currently run in different state and industrial enterprises in different regions of the Country.

Customer situation

Until 2015, ABC was working with a few regular customers. The process of finding and attracting new customers was conducted ineffectively and not systemically. In the beginning of 2015, the company decided to start an active marketing of its services in the market. For these purposes, the company needed a full-featured active sales and marketing department.

Construction of marketing processes by the company's experts did not lead to success. In June 2015, the company management decided to involve external consultants and build sales and marketing department from scratch. By the results of a tender, one consultancy company was chosen. Luckily, the partner of that consultancy company is a friend of Business Owner Bob (BOB). Consultants made a power-point presentation about the strategy of forming a new department and n IT solutions for that as well to Bob and IT Enterprise Architect Euclid (EAE) among others.

What comes to IT part of that strategy, EAE is concerned about the cloud based marketing software component functionality, quality, and maturity since consultants could not provide functionality details and customer references during the presentation. Moreover, EAE feels that interoperability among other systems in place in ABC might be a problem.

BOB has high visibility and political clout in the organization and has funding for the project. He wants to implement this solution recommender by consultants right away. EAE does not feel comfortable approving the solution because of the risks associated (there has been only a power-point presentation which was not very helpful). At the same time he does not want to slow down the business. He wants to do the right thing for ABC but does not want to strain his relationship with BOB either. Therefore, he decided to formulate the requirements for such a solution and pose them to consultants before ABC agrees to implement the solution. For that he will use a new web-based DSM for choosing a cloud service.

Requirements analysis – DSM in use

Step 1

The DSM is composed of 5 steps needed to analyse the business case and ends up with a list of requirements for a cloud service needed. A screenshot below presents the first step, choosing the service functionality:

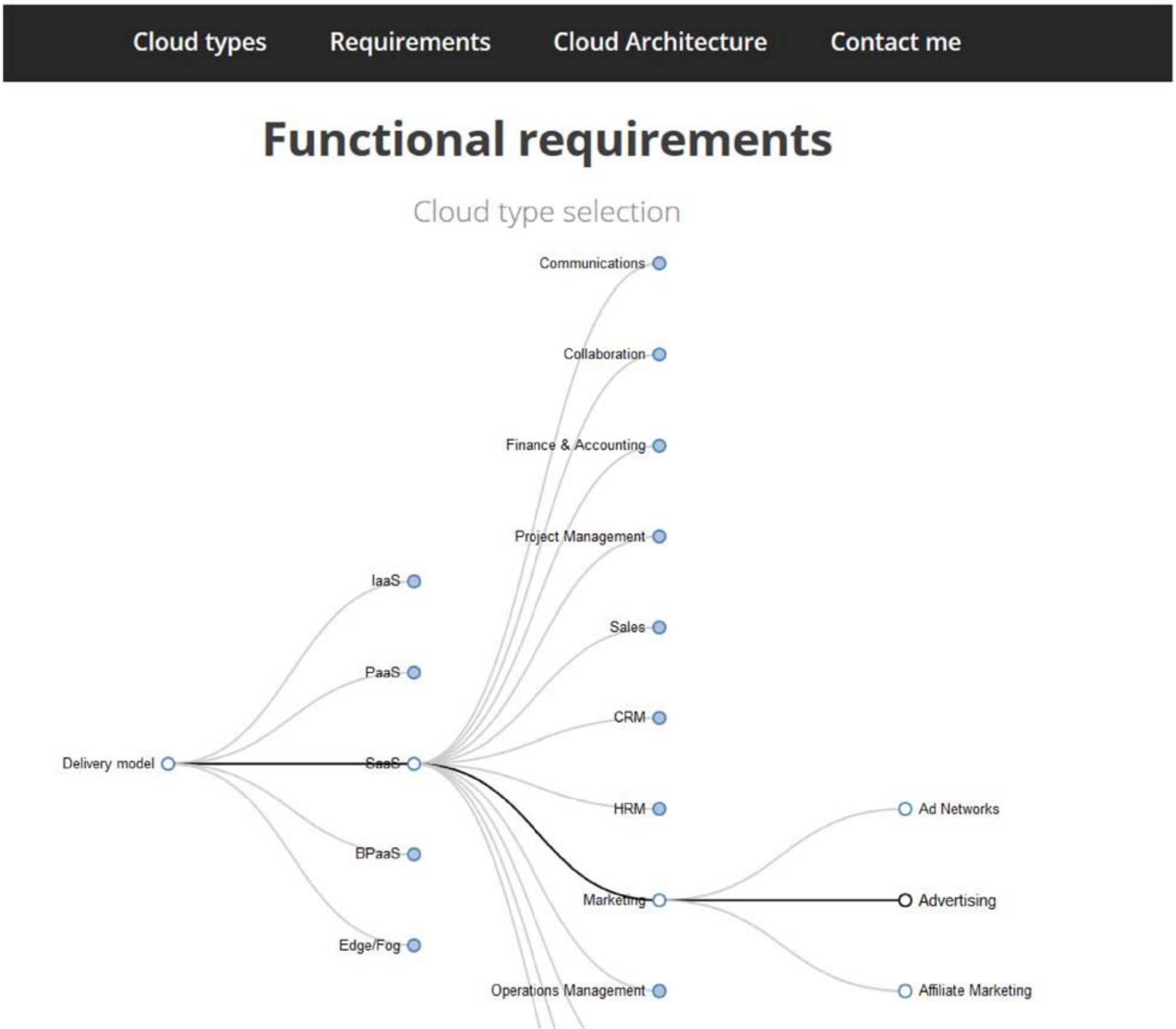
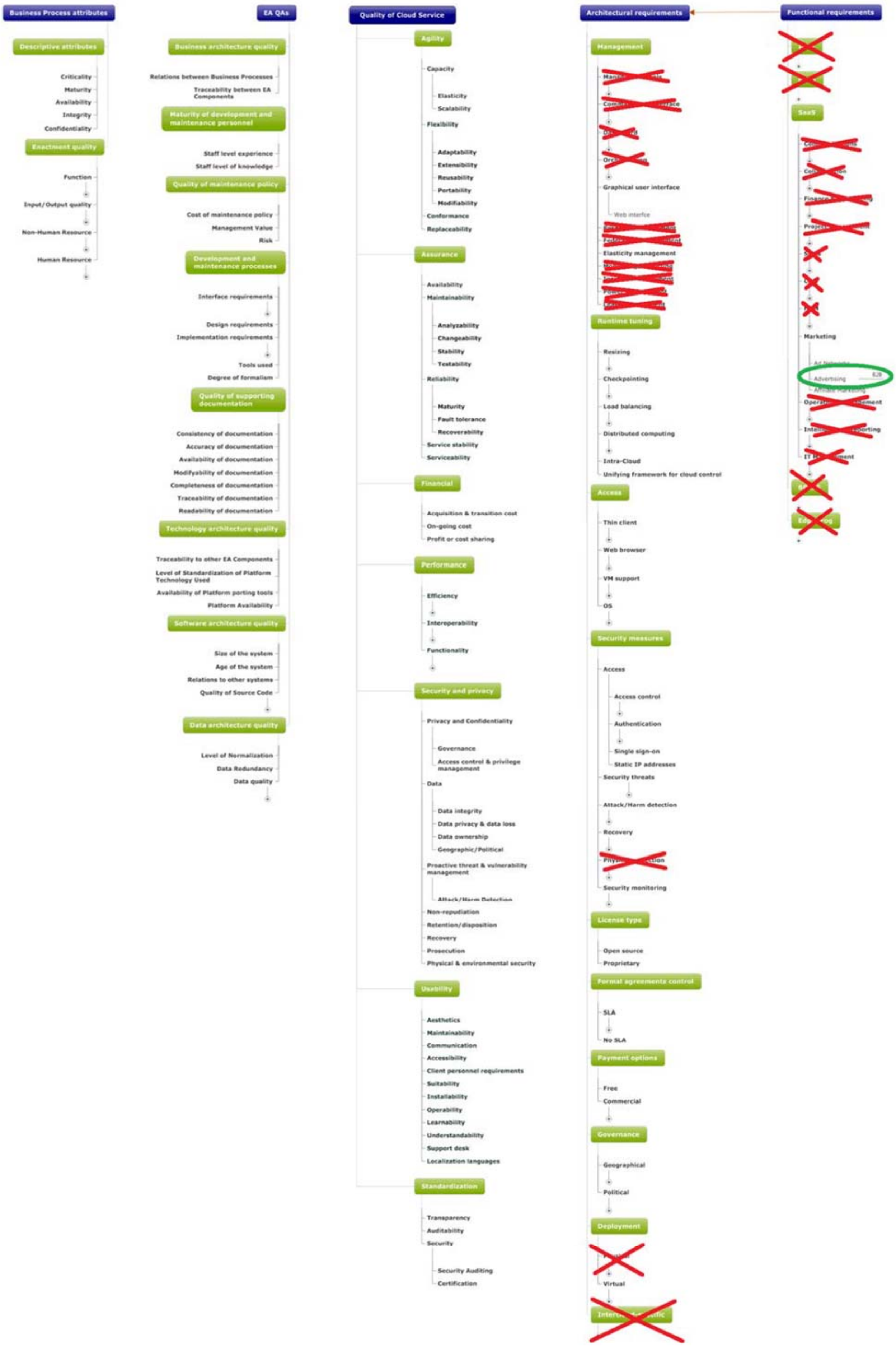


Figure 84 DSM website in operation – step 1

The DSM after the first step is shown on the next page.

Here and after, choices made are shown in green and affected options are shown in red.

EAE is choosing a cloud based service which allows all staff members of ABC to find customers for the products they are developing and engage in sales. This procedure is a part of sales strategy proposed by Consultancy. The motto of this campaign is: "Everybody is engaged in sales". During the pre-project of building a functional sales department, ABC's developers will use that system to find customers using personal marketing techniques. That includes B2B sales in professional communities, linked-in-like social media and personal connections. Appropriate tooling must be in place in that system. That system should be cloud based, interoperable and easily extendable among other companies in the group. Other companies will decide whether to adopt the practice or not and so the costs of making such decision should be minimised (agility without big investments is a cloud benefit).



Step 2

New business process is business critical due to the fact that company profit will depend on future sales. However, it is not mission-critical for ABC due to a large number of sustainable partnerships with long-term customers. Therefore, availability is important, but not crucial to business.

Business Process maturity defines the level of detail to what the BP is annotated, and thus ease of service implementation. And of course, the more mature BP is it requires more sophisticated IT service to handle it implying higher cost of running such system. As we know, current BP is in initial phase, and no sophisticated service is necessary at this stage.

Availability is restricted to office hours. However, different time zones across different cities are to take into account when choosing a provider. This is connected to maintenance quality and costs associated on.

High BP integrity possesses service requirements to ensure that the process is performing its intended functions without being degraded or impaired by changes or disruptions in its internal or external environments.

No confidentiality is required while all business connections are promoted. As all the software is tailored, more public the information about its implementation is available, better is for marketing purposes.

Due to the fact that ABC is a group-owned company, its sales, marketing and business connections should be available to other companies in the group to produce synergy. Thus this BP is related to similar BPs in other companies. In fact, it is something that Consultancy was not taken into account when presenting the strategy to ABC.

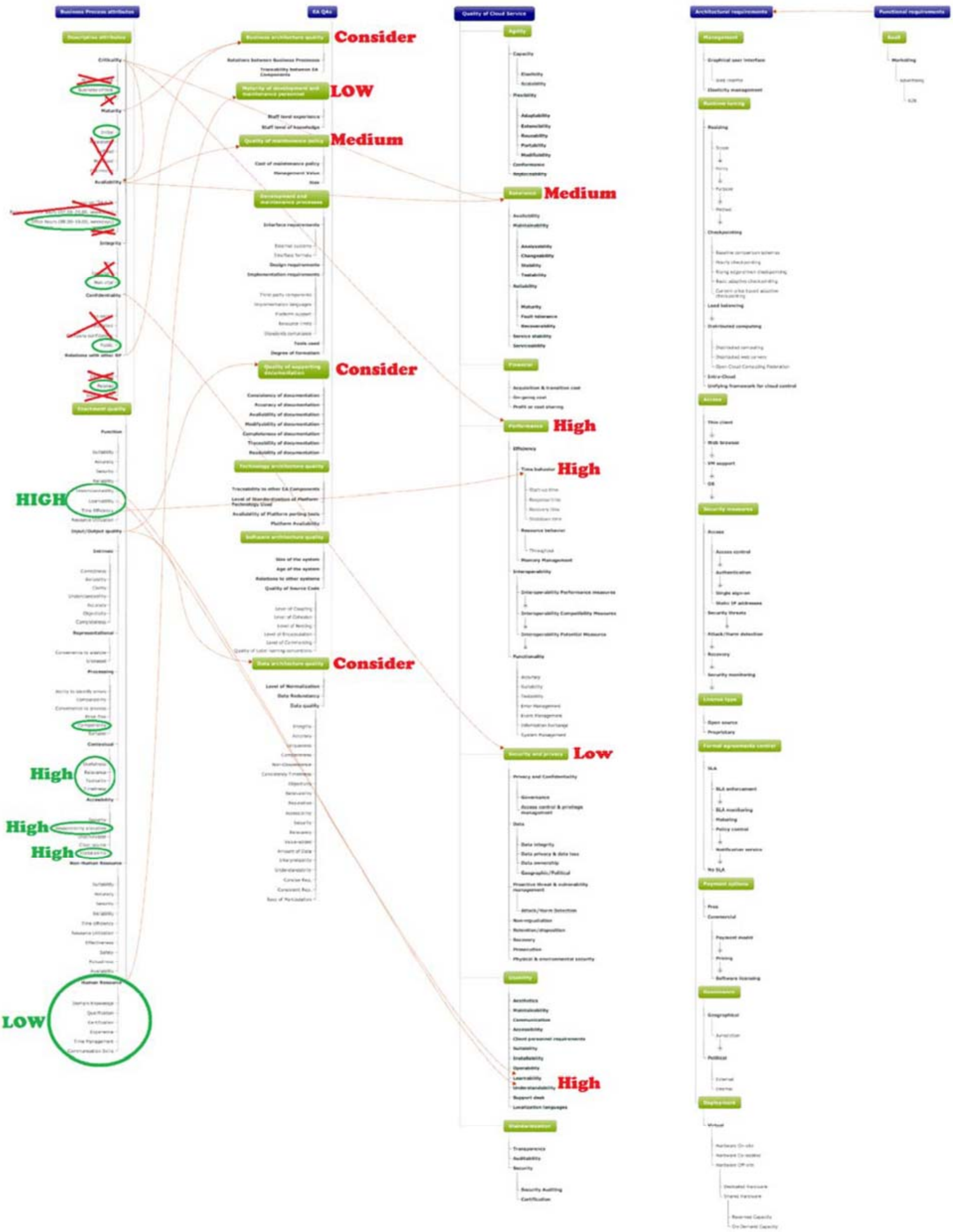
Enactment quality of the new BP should be maximized in the following areas: understandability, learnability, and time efficiency. ABC specialists are mainly busy with development process and time to learn and use new tooling should be minimized.

First of all, Business Process input/output quality relates to data quality restrictions. As it was mentioned before, other companies in the group should be able to access customers database, see the contacts in charge for sales, add contacts they have, etc. Moreover, the development plan includes a project for the implementation of CRM system. It also poses data interoperability requirements.

Non-human resources are not involved in sales and marketing procedures. As for degree of personnel involvement into BP execution, especially in the first phase of the project – it is high and the competence level is low. That affects the requirements for the service used to support this BP very much. A new system should be functional and easy to use without background knowledge at the same time. It is a very hard and strict requirement for ABC.

The DSM after the second step is shown on the next page.

Again, choices made are shown in green and affected options are shown in red.


Consider
LOW
Medium
Medium
Consider
High
High
HIGH
Consider
Low
High
High
High
High
LOW

Step 3

Business architecture specifics affect software architecture (and thus the standardization) capabilities of a new service as well as deployment options. Due to formulated multi-company usage requirements, new cloud should be able to be adopted in all companies of the group. Specific requirements should be formulated in QoS and Architectural decisions in further steps. Also, BA relates to development process QAs:

- External systems: BCD company already has CRM system in place, and ABC business contacts should be able to be exported in it.
- Interface formats: some companies use Open Office, and some MS Office meaning the import/export capabilities of different document formats.
- Platform support: new marketing tool should be able to import National social network contacts and operate in National search engine, while some of the state companies use non-UTF web addresses and are not indexed well in google.
- Standards compliance: The Open Group Data Architecture standard is used in ABC, so it is better to have a cloud data being able to suit company practice.
- Design requirements: EAE knows that a sister-company CDE employs a valuable developer who is blind. Therefore, a new tool should avoid CAPTCHA texts of social engines

Staff level experience and Staff level of knowledge are low as it was commented before. It has an effect on usability QoS (should be easy to learn and use), and on assurance (ABC staff cannot be responsible for service maintenance, so the provider should be highly reliable in all related QoS).

Development quality is tightly connected to future performance QoS. As for the maintenance, we already assume that it will be outsourced to vendor. Still, EA maintenance quality corresponds to costs.

Quality of supporting documentation: due to the nature of ABC business (close collaboration with government institutes) ABC accounting and thus Enterprise Architecture documents should be utmost transparent for audit. Moreover, ABC employees should be able to access guides and FAQs for a new system easily to ensure fast adoption of the service. Therefore, the documentation supporting a new cloud service should be provided by the vendor and contain all the modules of the system well described.

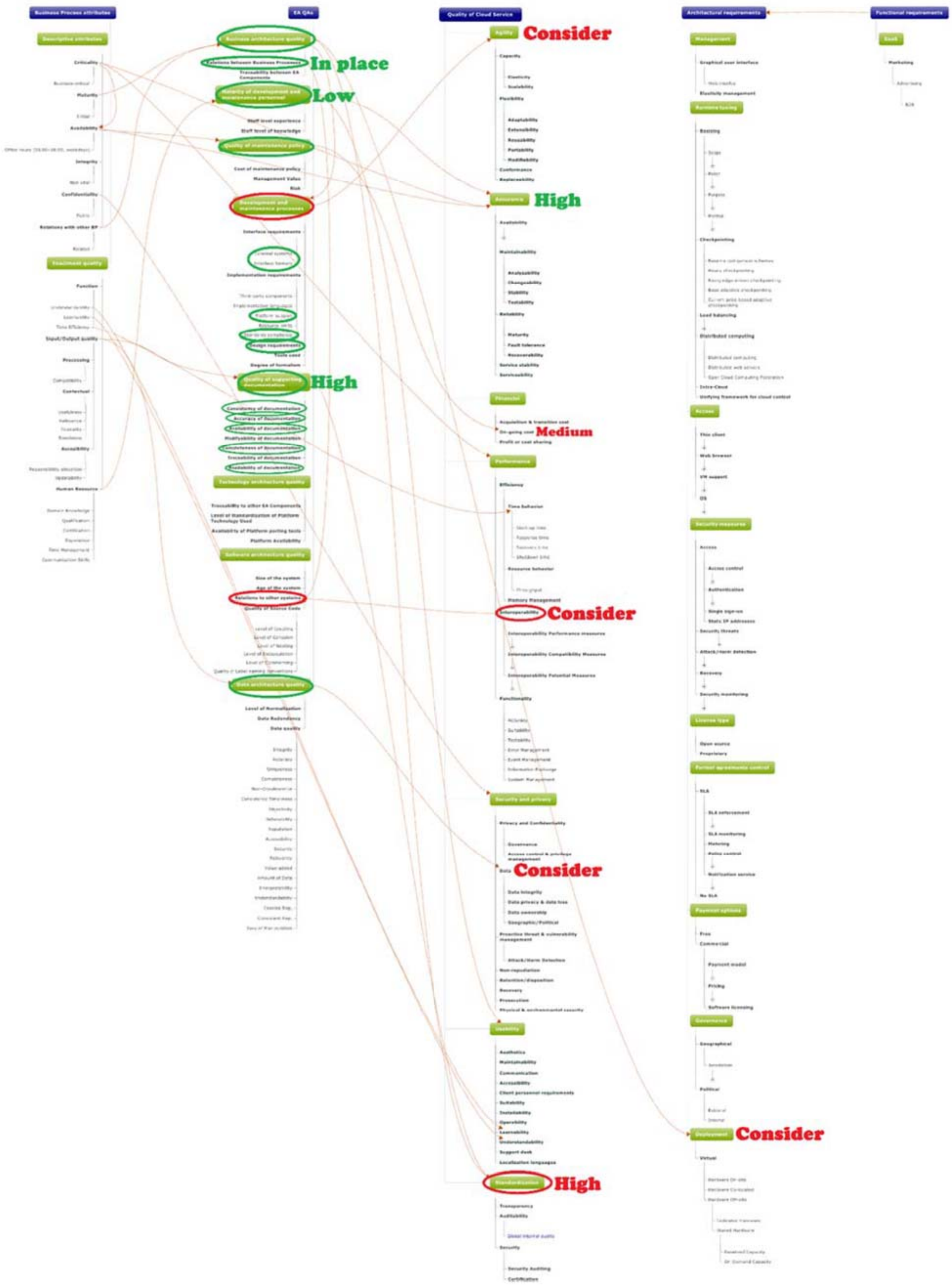
Technology behind SaaS solution is not questioned, while all QoS requirements are fulfilled.

Software architecture quality: relations to other systems mentioned before affect interoperability QoS.

Data architecture quality requirements remains the same from business side.

The DSM after the third step is shown on the next page.

Again, choices made are shown in green and affected options are shown in red.


Consider
In place
Low
High
High
Medium
Consider
Consider
Consider
High

Step 4

Agility.
Knowing that interface requirements are in place and that this system might be used by other companies (not known to which extent), the scalability of a solution should be high. Same goes for adaptability for the other companies of the group. Knowing that this is the first marketing system for ABC and sales practices might change considerably in future, this system should be easily replaced.

Assurance.
12 hours uptime per day on Mondays until Fridays and 0 hours on weekends. On 99.5 SLA level such scheduled downtime gives 18 minutes weekly unavailability comparing to 54 minutes when paying for 24/7 schedule with the same 99.5% level. This is considered an affordable and acceptable level for a new ABC marketing tool.

Maintainability is in vendor's competence, and its stability as well as reliability levels should be prescribed strictly in SLA.

Financial.
EAE recommended the following financial decisions to ABC finance team:

- Acquisition & transition costs should not be high while this is a first try out of a marketing tool;
- On-going cost can be medium comparing to other SLA levels, while all the maintenance is on the vendor and assurance levels are prescribed high in SLA;
- Profit or cost sharing among other companies in the group is possible to the extent they will use the system.

Performance.
Start-up time considered valuable. EAE had some experience with cloud services starting half an hour☺

Interoperability requirements should be formulated in SLA.

Accuracy of service functionality and information exchange reliability should be formulated in SLA.

Security & Privacy.
New service should be able to provide tools to manage service governance, access control, and privilege management.

Data loss should be prevented at least through recovery mechanisms.

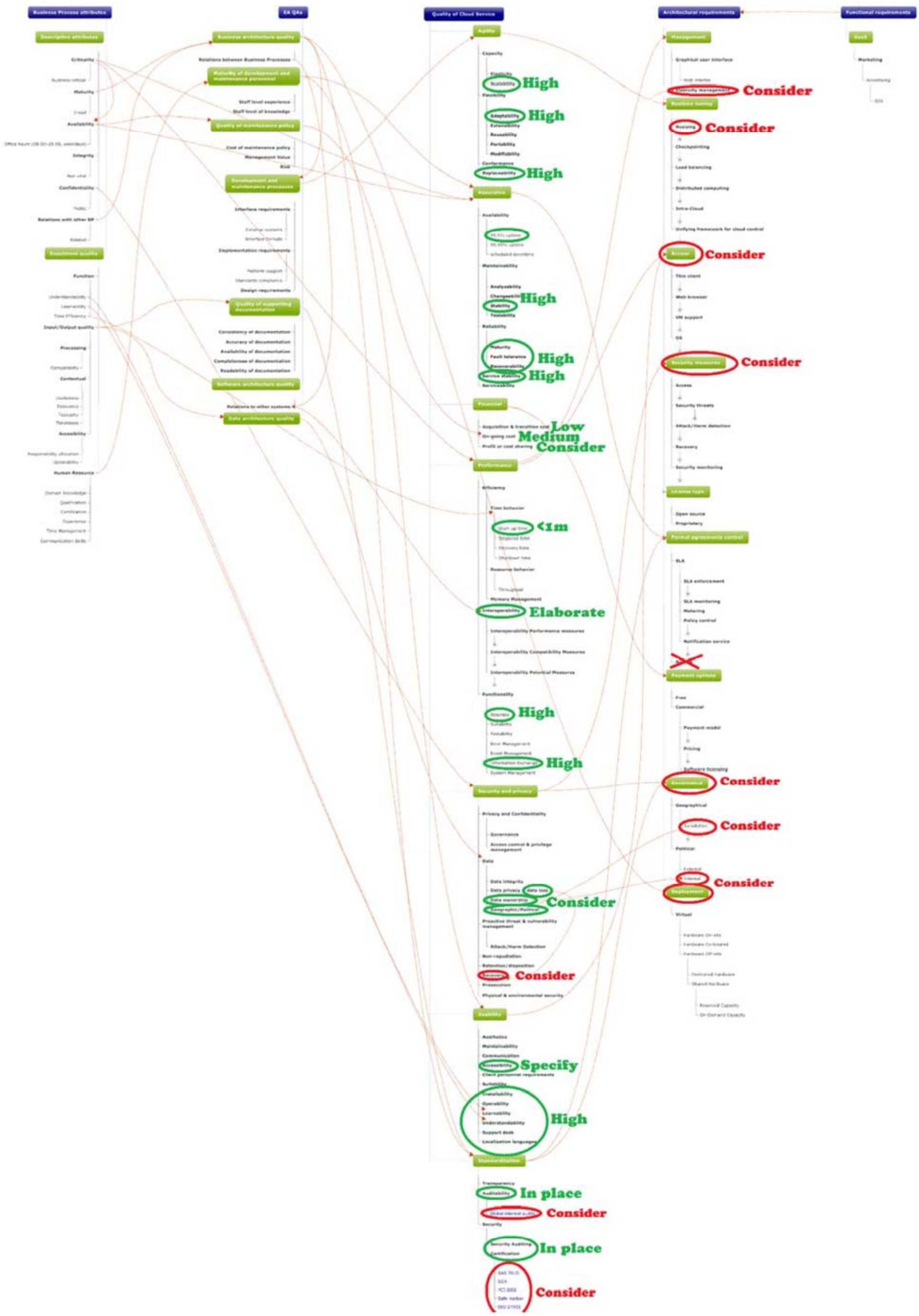
Data ownership should be managed among partner companies

Proactive threat & vulnerability management, and attack/harm detection should be in place in the system.

Usability.
Following features should be in place:

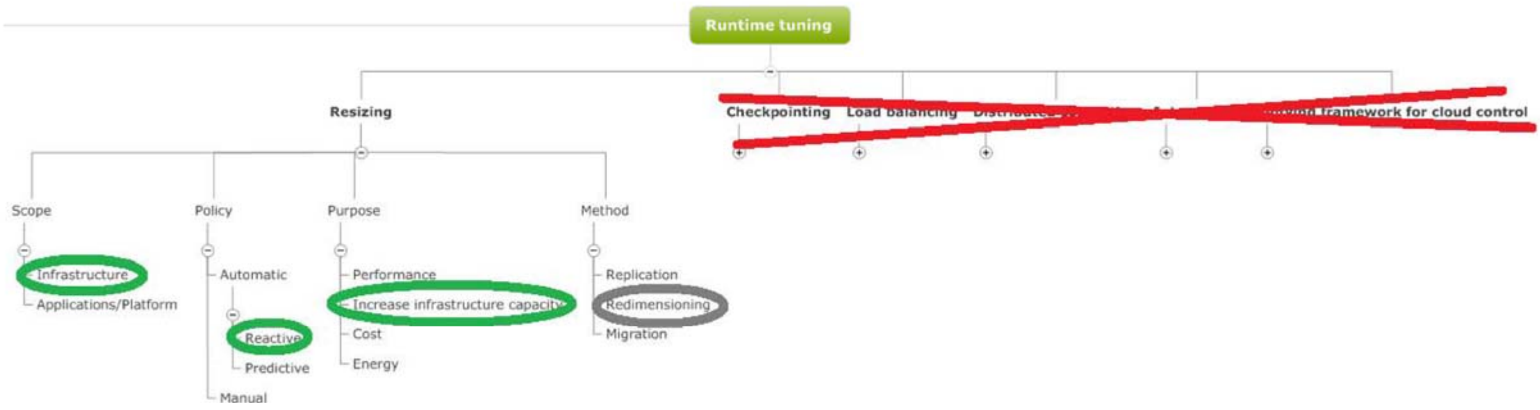
- Accessibility for visually impairment personnel;
- Easy installability, operability and learnability/understandability;
- Vendor's support desk should be available for ABC personnel. It is crucial for non-expert users;
- Local language should be available for all parts of the system.

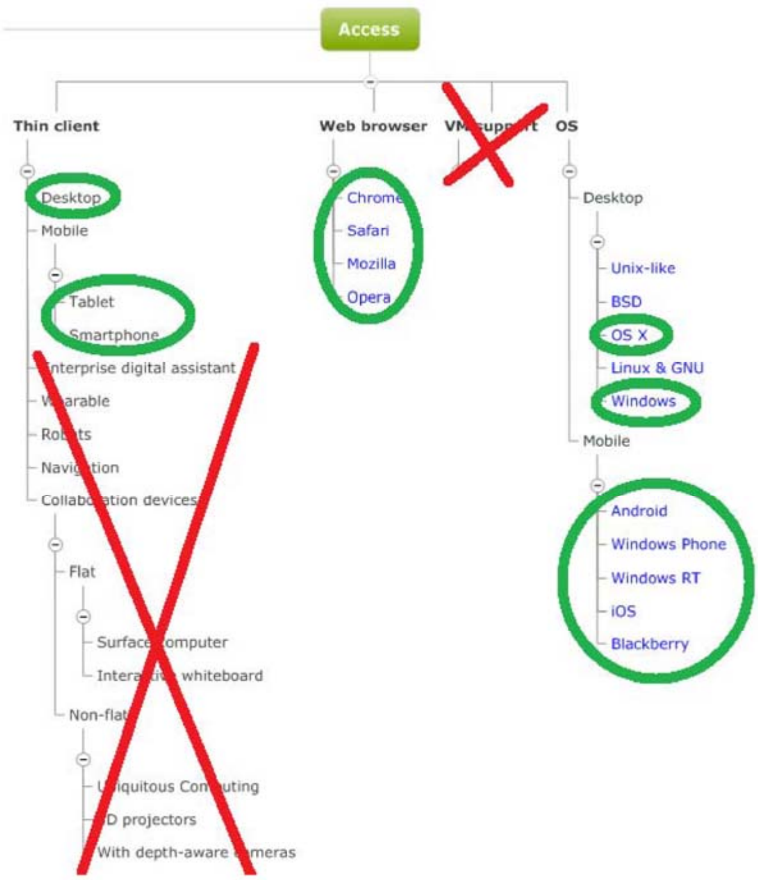
Standardization.
As it was mentioned before, auditability of documentation and service costs is very important for ABC. Therefore cloud vendor should be able to testify its own security procedures (e.g. how and where it handles customer data) to third-party auditor.

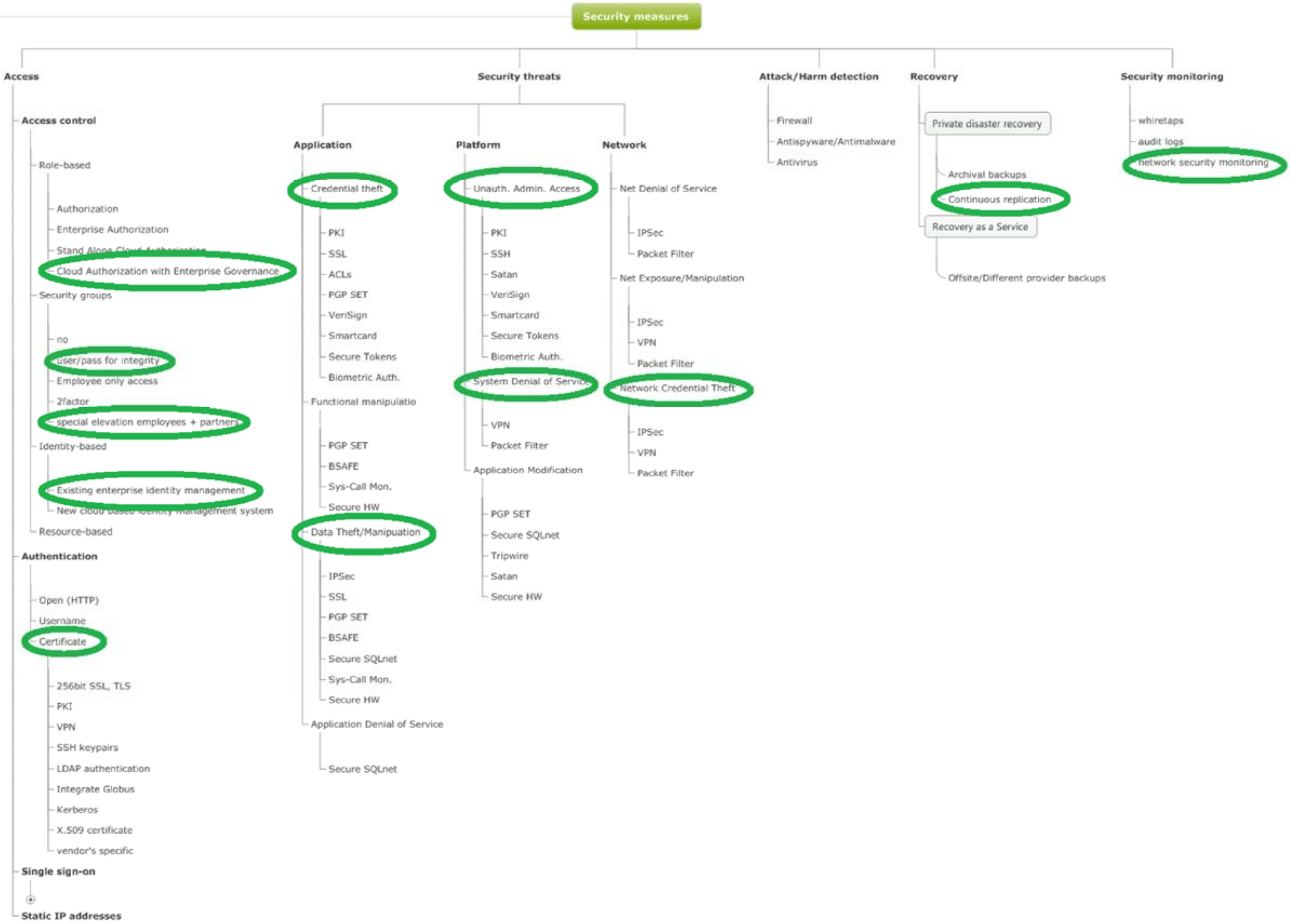

High
High
High
Consider
Consider
Consider
Consider
High
High
High
Low
Medium
Consider
<1m
Elaborate
High
High
Consider
Consider
Consider
Consider
Consider
Specify
High
In place
Consider
In place
Consider

Step 5

Resizing mechanism should be automatic, when new users start using the tool, and based on reactive policy. This is the most user friendly and easy solution acceptable for ABC users.






Security measures
Access
Access control
Role-based
Authorization
Enterprise Authorization
Stand Alone Cloud Authorization
Cloud Authorization with Enterprise Governance
Security groups
no
user/pass for integrity
Employee only access
2factor
special elevation employees + partners
Identity-based
Existing enterprise identity management
New cloud based identity management system
Resource-based
Authentication
Open (HTTP)
Username
Certificate
256bit SSL, TLS
PKI
VPN
SSH keypairs
LDAP authentication
Integrate Globus
Kerberos
X.509 certificate
vendor's specific
Single sign-on
Static IP addresses
Security threats
Application
Credential theft
PKI
SSL
ACLs
PGP SET
VeriSign
Smartcard
Secure Tokens
Biometric Auth.
Functional manipulatio
PGP SET
BSAFE
Sys-Call Mon.
Secure HW
Data Theft/Manipuation
IPSec
SSL
PGP SET
BSAFE
Secure SQLnet
Sys-Call Mon.
Secure HW
Application Denial of Service
Secure SQLnet
Platform
Unauth. Admin. Access
PKI
SSH
Satan
VeriSign
Smartcard
Secure Tokens
Biometric Auth.
System Denial of Service
VPN
Packet Filter
Application Modification
PGP SET
Secure SQLnet
Tripwire
Satan
Secure HW
Network
Net Denial of Service
IPSec
Packet Filter
Net Exposure/Manipulation
IPSec
VPN
Packet Filter
Network Credential Theft
IPSec
VPN
Packet Filter
Attack/Harm detection
Firewall
Antispyware/Antimalware
Antivirus
Recovery
Private disaster recovery
Archival backups
Continuous replication
Recovery as a Service
Offsite/Different provider backups
Security monitoring
whiretaps
audit logs
network security monitoring

And so on...

Result

EAE was able to formulate elaborate requirements document.

Interface requirements

External systems:

Interface formats:

Platforms support:

Standard compliance:

Design requirements:

Quality of supporting documentation: Consistency, Accuracy, Availability, Completeness, Readability

Assurance QoS

Availability: 99.5% uptime

Maintainability: Stability

Reliability: Maturity, Fault tolerance, Recoverability

Service stability

Performance QoS

Start-up time: less than a minute.

Interoperability requirements

Accuracy and information exchange reliability levels

Security & Privacy QoS

Governance, access control, and privilege management, recovery mechanisms, data ownership management, proactive threat & vulnerability management, and attack/harm detection should be in place in the system.

Usability QoS

Accessibility for visually impairment personnel;

Easy installability, operability and learnability/understandability;

Extensive support desk during office hours;

Full localization.

Standardization QoS

Vendor security certification in place;

Readiness for external audit.

Architectural requirements:

...

Outcome

The reasoning behind each decision is present, based on company specifics, accessible, and well documented.

Instead of accepting initial marketing tool (which was based not on ABC and its group of companies specifics but on industry best practices), consultants proposed one which fulfils all posed requirements.

ABC did not receive a large number of complaints from partner companies related to:

- incompatibility with their interfaces such as CRM systems;
- usage difficulties for visually impairment personnel;
- inability to process large number of own clients due to cloud scalability restrictions;

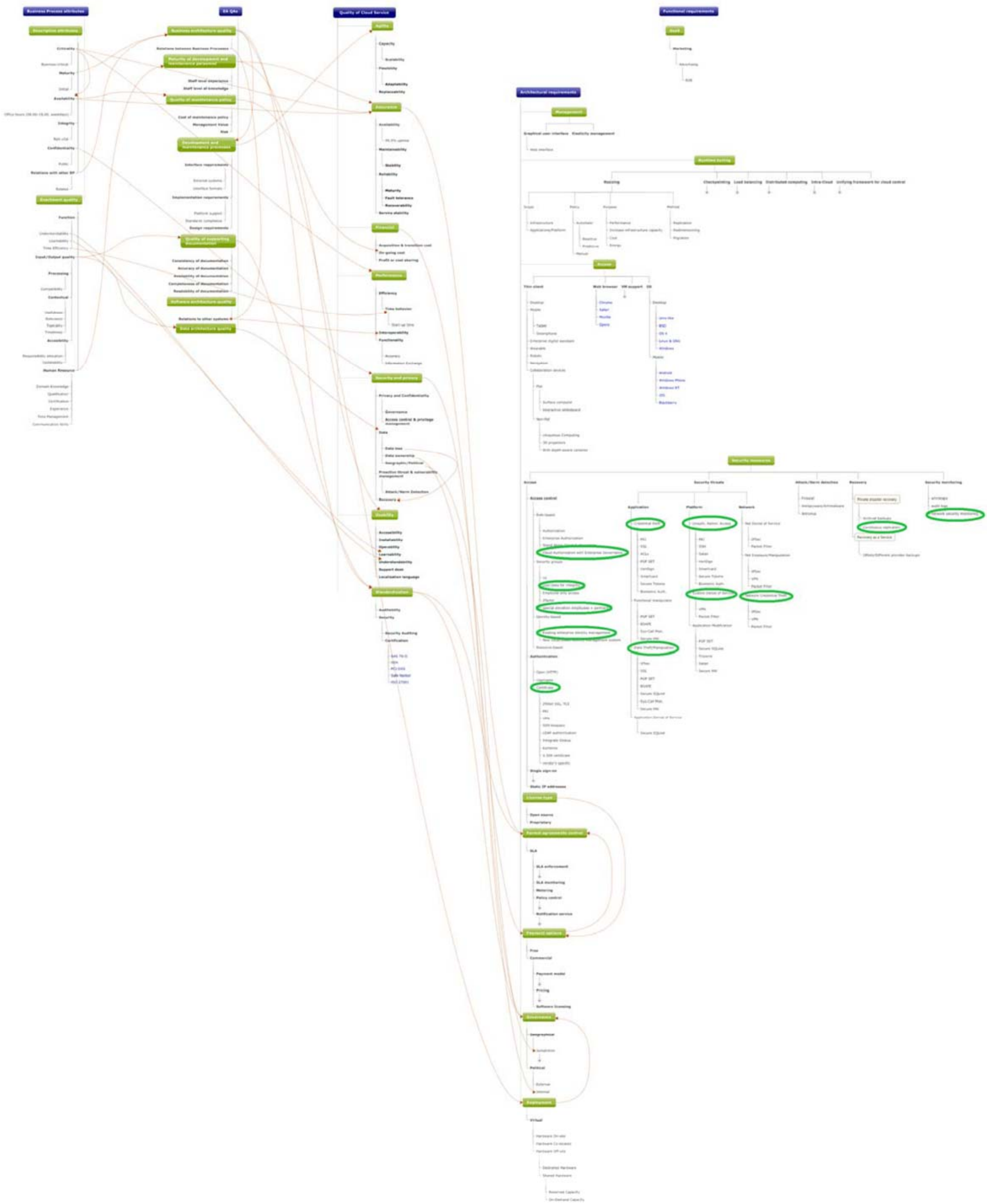
ABC did not pay more for SLA levels including 99.9% uptime, but still managed to retain it during office hours of all partner companies.

ABC accounting team developed dynamic cost sharing for the usage of marketing tool by other partner companies.

ABC avoided vendor lock-in due to EAE posing replicability requirement originated from marketing business process maturity level.

In 2016 the company expected 10-20 new customers, but ended up with nearly 200 new customers. In 2017 ABC replaced marketing solution with BPaaS including sales, marketing and CRM without losing too much time on data migration.

The overview of the DSM is presented on the next page.


Business Process attributes
EA QAs
Quality of Cloud Service
Functional requirements
Architectural requirements