jamk.fi

Enterprise architecture

Case Pyhäsalmi Mine Oy

Antti Pasanen

Bachelor's thesis
May 2016
Technology, communication and transport
Degree Programme in Information Technology (Data Network Technology)

Jyväskylän ammattikorkeakoulu
JAMK University of Applied Sciences

jamk.fi

Description

Author(s)	Type of publication	Date
Pasanen, Antti	Bachelor's thesis	May 2016
	Number of pages 52	Language of publication: English
		Permission for web publication: (x)

Title of publication
Enterprise architecture
Case Pyhäsalmi Mine Oy

Degree programme
Degree Programme in Information Technology (Data Network Technology)

Supervisor(s)
Sampo Kotikoski, Mika Rantonen

Assigned by
Pyhäsalmi Mine Oy/IT-department

Abstract

This Bachelor's thesis project was assigned by the Pyhäsalmi Mine Oy, copper- and zinc mine located in Northern Ostrobothnia. The company did not have any previous existing Enterprise architecture, and they needed help in showing the benefits of Enterprise architecture to the company management.

For this specific reason the study focuses on a presentation of few selected Enterprise architecture frameworks, an analysis of their contents in terms of the required scope and a thorough review of the sections expected to bring the most benefit to the company, leading to high-level guidelines of the most essential issues to be considered in architectural work. The research was carried out as a qualitative study since it was a natural first option in this kind of case study.

As a result, the company's IT department has a way to faster rationalize the need for Enterprise architecture to the higher management and thus gain both the authorization to get all the employees to participate as well as funding for carrying out the entire project is easier. Because of the extent of Enterprise architecture, especially in terms of mapping out all the business processes in place, the support of the highest management is crucial.

Even though the selected frameworks had some partially overlapping sections and modes of operation, eventually they formed a unified body. Each one presented their strengths that were worth becoming a part of the final product, and also more generalized definitions left out were selected from the other frameworks.

Keywords/tags
Enterprise architecture, TOGAF, COBIT, ITIL, FiSMA

Miscellaneous

jamk.fi

Kuvailulehti

Tekijä(t)	Julkaisun laji	Päivämäärä
Pasanen, Antti	Opinnäytetyö, AMK	Toukokuu 2016
		Julkaisun kieli: Englanti
	Sivumäärä 52	Verkkojulkaisulupa myönnetty: (x)

Työn nimi
Enterprise architecture
Case Pyhäsalmi Mine Oy

Tutkinto-ohjelma
Insinööri (AMK), tietotekniikan koulutusohjelma (tietoverkot)

Työn ohjaaja(t)
Sampo Kotikoski, Mika Rantonen

Toimeksiantaja(t)
Pyhäsalmi Mine Oy/IT-osasto

Tiivistelmä

Opinnäytetyön toimeksiantajana toimi Pyhäjärvellä Pohjois-Pohjanmaalla sijaitseva kupari- ja sinkkikaivos Pyhäsalmi Mine Oy. Yrityksellä ei ollut ennestään olemassa olevaa kokonaisarkkitehtuuria, ja he halusivat apua kokonaisarkkitehtuurin hyötyjen todentamiseen yrityksen toimivalle johdolle.

Tästä syystä tutkimuksen keskiöön valittiin muutamien kokonaisarkkitehtuurin viitekehyksiä. Työ sisältää niiden esittelyn, analysoinnin työn vaatimassa laajuudessa sekä kokonaisvaltaisen tarkastelu osa-alueista, joissa kokonaisarkkitehtuurista ajateltiin saatavan yritykseen suurin hyöty. Tämän pohjalta luotiin yleisiä korkean tason suuntaviivoja huomioon otettavista asioista arkkitehtuurityössä. Tutkimus toteutettiin laadullisena tutkimuksena, koska se valikoitui luontevasti vaihtoehdoksi tämän kaltaisessa tapaustutkimuksessa.

Tuloksena yrityksen IT-osaston osalta arkkitehtuuriprojektin aloitus on helpompi, koska arkkitehtuurityön tarpeellisuuden perustelu yrityksen johdolle onnistuu nopeammin ja näin saadaan sekä valtuutus, jolla kaikki yrityksen työntekijät saadaan osallistumaan työhön, että rahoitus koko projektin toteuttamiseksi. Koska kokonaisarkkitehtuurin on erittäin laaja ja aikaa vievä projekti, johdon tuki on elintärkeä sen onnistumisen kannalta. Erityisesti kaikkien yrityksen toimintaprosessien kartoitus ei onnistu ilman ylemmän tason valtuutusta.

Vaikkakin tarkasteltavaksi valituissa viitekehyksissä oli osittain päällekkäin asettuvia osioita ja toimintatapoja, loppujen lopuksi ne kuitenkin nivoutuivat hyvin yhtenäiseksi kokonaisuudeksi. Jokaisesta löytyi omat vahvuusalueet, joita kannatti hyödyntää, ja hieman yleisemmin määritellyt osat, jotka oli järkevämpää täydentää muista valituista viitekehyksistä.

Avainsanat
kokonaisarkkitehtuuri, TOGAF, COBIT, ITIL, FiSMA

Muut tiedot

Content

Figures

Abbreviations

ABB	Architecture Building Block
ADM	Architecture Development Method
AVI	Aluehallintovirasto (Regional State Administrative Agencies)
COBIT	Control Objectives of Information and related Technology
EA	Enterprise architecture
ERP	Enterprise Resource Planning
FEEM	Federation of European Explosives Manufacturers
FQML	First Quantum Mining Ltd.
FPA	Functional Point Analysis
ISACA	Information Systems Audit and Control Association
ITIL	Information Technology Infrastructure Library
KaTTi	Kaivoksen Tuotannon Tietojärjestelmä (Mines production data warehouse)
SBB	Solution Building Block
SOA	Service-Oriented Architecture
SOTE	Sosiaali- ja Terveysministeriö (Ministry of Social Affairs and Health)
TOGAF	The Open Group Architecture Framework
TRM	Technical Reference Model
VAHTI	Valtionhallinnon Tietoturva- ja Kyberturvallisuuden johtoryhmä (State Administration Management Group of Information- and Cyber Security)

1 About the customer

Pyhäsalmi Mine Oy is a mining company located in Pyhäjärvi, Northern Ostrobothnia. The premise for the mine came to life in 1958, when local resident Erkki Ruotanen happened to find strange looking pieces of rock in his property after digging a well. The samples were sent to be analyzed by geology experts and minerals were found in high enough concentrations to enable profitable mine to be built. The actual mining activity began four years later, in 1962. The founder of the mine was a Finnish mining company Outokumpu Oy. Today, Pyhäsalmi mine is the biggest individual mine in Finland as well as the deepest in all Europe.

Pyhäsalmi Mine Oy employs about 200 people in-house. (Vuorimiesyhdistys, 8) The most all mining activity today takes place in depth of 1050-1440 meters from the surface. The main source of revenue for the mine is sales of copper, pyrite and zinc concentrates dressed in their own processing plant. In 2001, Outokumpu Oy sold the mine to Canadian Inmet Mining Corporation. Inmet Mining Corporation was acquired by another Canadian company, First Quantum Minerals Ltd. in 2013, and the ownership of Pyhäsalmi Mine also changed at that time. (First Quantum Minerals history)

2 Project background

The motivation for this project, from the company's perspective, derived from the need for better management and easing the planning of modifications, maintenance and repairs to their existing infrastructure, in other words a better governance and management overall. Ill-planned projects tend to hike up the total costs for a very simple reason: figuring out the issues on the go can very easily result in unplanned service outages, delays and those often equal to extra man hours which means additional costs. In order to carry out projects effectively and in timely manner, an up-to-date model of the infrastructure is crucial. That model would work as a blueprint for project teams, making it much easier to assess the effects the proposed changes have on the systems based on how they interact with each other and minimize the

possible risks before they turn into actual problems. Building a complete Enterprise architecture (EA); however, is an incredibly vast project, even in company of this size, and thus it is not very suitable for a thesis project. The client wanted to have a document that could be used to rationalize the benefits over the expense of such an effort to the management group of the company, analyzing different EA-frameworks, detailing what kind of EA solution would be good for Pyhäsalmi Mine Oy and how implementing EA would benefit the top management, governance and the company as a whole.

3 Enterprise architecture

3.1 What is an enterprise?

When discussing Enterprise architecture, it should first be figured out what is the definition of enterprise. According to TOGAF, an enterprise is simply a group of organizations that all ultimately work towards a common goal. Enterprises come in multiple sizes: from something as small as a department inside a company to the whole company or even group of companies with a common owner or are linked tightly in some other way. In terms of Enterprise architecture, the term enterprise can be used to describe an entire enterprise, including all of its IT services possibly spread around multiple locations, processes and infrastructure in varying degree, or smaller section of the whole. (TOGAF 9.1, 2011, 5)

COBIT definition agrees with the statement of varying enterprise sizes; however, it also adds to that definition: division of different enterprises is based on the type of business. They are differentiated by the method of receiving funding: commercial, non-profit and public sector enterprises. While TOGAF also states that enterprise is a group of organizations, COBIT view includes singular entities to qualify. (COBIT 5, 2012, 13)

Thus, an enterprise could also be seen as a team. A group of companies or people that form one work together in order to achieve a common goal that benefits them as a whole. Because the size varies, the leadership requirements can be expected to

vary as well. It is not possible to lead a department of a company, consisting only of a few people, in exactly the same way as a huge multinational corporation with thousands of employees. Keeping track of everything that happens in latter case would be virtually impossible. However, these two cases have common aspects and one on the top of that list that most people can agree on: thorough knowledge about how the gears keep on turning is crucial for keeping the whole complex in a working order. It is crucial in identifying possible risks that may be present. When new risks are found, anticipating their effect and decreasing the chance for occurrence, in best case negating the risk completely, makes the organization work in an efficient way. In this project three major frameworks dealing with EA and IT Service Management were selected due to the requirements of the employer and their relevance to the case itself.

3.2 TOGAF

The Open Group Architecture Framework (TOGAF for short) is all about Enterprise architecture. It is a framework that gives detailed instructions about creating EA-implementation and adjusting it to be specific to an organization. It is open source; however, the commercial use requires purchasing a license from Open Group organization. The usage inside an organization is not seen as commercial use, as long as the EA is developed for the organization creating it and not for the needs of some third party. (TOGAF Architecture Forum, 2011)

The development of TOGAF standard is created by The Open Group organization and its multiple member organizations. Originally the development process was started in 1995, using an already existing Technical Architecture Framework for Information Management (TAFIM). TAFIM was a creation of United States Department Of Defense (DOD) and with their blessing it could be used as a foundation for this new framework. By allowing it, DOD gained more out of all the government investments already spent on TAFIM at that point. Multiple versions of the framework exist, most

recent being TOGAF 9.1 and they have all been made publicly available via The Open Group website. (TOGAF 9.1, 2011, 3)

3.3 COBIT

3.3.1 History

The development of COBIT framework is similar to TOGAF: ISACA, nowadays a global organization behind the work, is the top authority in COBIT. ISACA provides COBIT-certifications to professionals and is in charge of improving the framework. The seed for ISACA was planted in 1967 by a small group of individuals sharing a similar line of work: auditors with concerns about the critical computer system controls inside their own organizations. The group saw a need for centralized source of knowledge and guidance in the field, and in 1969, they incorporated as the EDP Auditors Association. Later on, the name changed to Information Systems Audit and Control Association and finally, just the acronym to reflect the broadness of the professional members of ISACA. The currently used revision of COBIT has a revision number 5. (ISACA history, ISACA guidance)

3.3.2 COBIT way

COBIT framework offers a set of instructions for companies wishing to govern and manage their IT in more comprehensive manner, allowing them to better serve their own individual business needs. Thus, COBIT is another alternative framework for building an EA-model. However, TOGAF and COBIT are not outright copies of each other and for that reason they do not both include exactly the same things. The two take different approaches on the subject: where TOGAF covers Enterprise architecture in multiple topics and offers a more general and overall aspect, COBIT approach emphasizes more the governance of enterprise and IT specifically. COBIT as business-driven philosophy can be seen as one of the contributors to this characteristic feature and as the major benefit for picking this particular framework. COBIT is also highly scalable: it works in enterprises of all sizes which puts the concerns of the company outgrowing the framework itself to the rest. (COBIT 5, 13-14)

Business executives would most likely see COBIT as the easiest to approach of the three, due to the familiar-feeling perspective it offers. They would be already familiar with some of the terminology and definitions it offers. In a probable case where an enterprise would want to implement only one framework and the background work made by the IT-department would be minimal to non-existent, maybe because of the governing group of the enterprise not being very favorable towards this kind of architectural work in the first place, COBIT would be a strong candidate. Figure 1 illustrates the five principles of COBIT framework that are the foundation and therefore crucial for anyone who wishes to understand how to apply COBIT in practice.

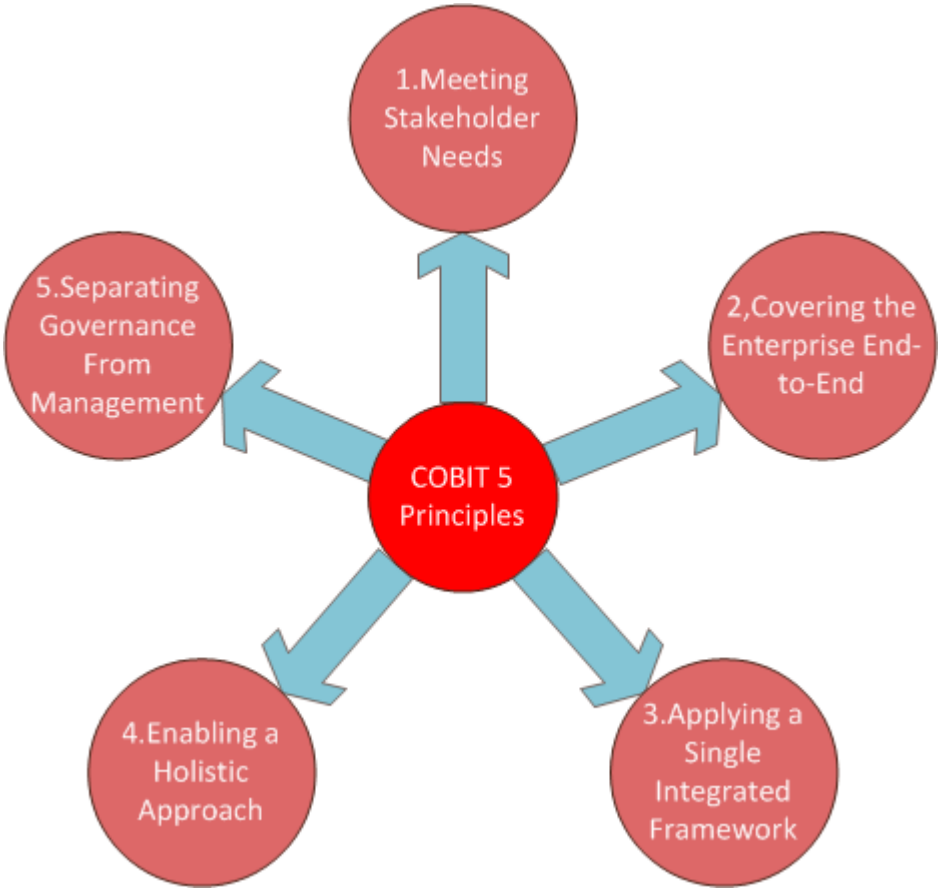


Figure 1 COBIT Principles.

COBIT framework is based on five base principles describing the most essential guidelines. The reason why the principles are illustrated in this way is most likely because all five are interconnected: all of them are to be considered equally and simultaneously when architecture is built. It's not possible to detach some of them and just throw them away, all five need to be part of the process. First one is meeting stakeholder needs, and it is undeniably the most important one when general principles of EA are considered. Enterprise creates value to the stakeholders by keeping a balance between the awareness of the benefits and keeping the risk and resource use at optimal level. COBIT processes to maintain this balance can be customized for the needs of specific enterprise by using the goal cascade technique included in COBIT. This way broad enterprise goals can be translated into very specific IT-related goals which can be in turn mapped to other processes when needed. (COBIT 5, 14)

The second principle deals with end-to-end enterprise coverage. According to COBIT, governance of IT in enterprise should be injected to the enterprise governance itself. Information is seen as an asset and thus it is to be treated as any other asset would be by every employee. End-to-end means that all the functions and processes inside the enterprise, no matter if they are IT-related or not, fall under COBIT domain. The third principle calls for applying a single integrated framework. This does not, however, mean that COBIT should necessarily be the only framework that the enterprise is using. Instead, COBIT is compatible with many other industry-standard frameworks and can be used in cooperation as a unifying solution that knits all the used ones together. Since its major benefits come from the management and governance functions, it fits very well to manage the whole. (COBIT 5, 14)

Two remaining basic principles deal with holistic approach and the separation of governance and management from each other, and these principles are interconnected. Holistic approach is a cornerstone in turning both governance and management of enterprise IT into effective and efficient processes and during the work, architecture team needs to take into account all interacting components to know, what kind of waterfall effect changing, modifying or removing a single component have on the rest of complex. (COBIT 5, 14)

COBIT 5 includes a definition for a set of enablers that support implementation of this holistic approach. Enablers are simply put anything that can help in achieving the

objectives of the enterprise. They are divided into 7 categories: PPF (Principles, Policies and Frameworks), Processes, Organizational Structures, CEB (Culture, Ethics and Behavior), Information, SIA (Services, Infrastructure and Applications) and PSC (People, Skills and Competences). Final principle calls for the importance of separating governance from management. Since they require divergent activities, organizational structures and serve differing purposes, COBIT indicates that they should function on their own, still remembering that the enterprise management has to act as a support for the business plan, most likely put into effect by the governing body, such as a board of directors led by a chairperson. Management, on the other hand is often carried out by the executive manager, working under company CEO. (COBIT 5, 14)

3.4 ITIL

3.4.1 Origins

Thirdly, there is the ITIL framework. ITIL development was started in the 1980s, and published by the UK Government department, known as Central Computer and Telecommunication Agency (CCTA for short) at the time. Later CCTA was absorbed by Office of Government Commerce and finally, OGC was joined into the Cabinet Office. CCTA was tasked to invent a way to improve the quality of IT services for the public sector in the UK. This was done by creating a framework to enable efficiency, financially stable and responsible use of their IT resources. The government had deemed that the service quality in place was not up to the desired requirements prior to the birth of ITIL. ITIL has 4 major revisions, first three known with their version numbers 1-3 and the fourth one as ITIL v3 2011 edition. The 2011 edition is the one that is widely in use today, due to major differences with versions 1 and 2. The original version 3, from 2007, was seen as too diffused and it was updated in the 2011 edition to have better consistency. From this point on, reference to ITIL can be understood to mean the 2011 revision. (Farenden, 2012.)

3.4.2 Mechanics of ITIL

ITIL models IT service management, i.e. that its aim is to ease governance of IT-services while making the development of new services and their lifecycle management as effective as possible. Scalability is one of the best aspects of ITIL: it works in companies of various sizes and will grow with the company when needed. Following some of the general EA guidelines, ITIL includes best practice models to help determine how to plan, implement, deliver and manage IT in support of business needs. Therefore, ITIL is not intended to be a set of instructions to be followed blindly to the letter, but rather guidelines to offer general direction for the development because in most cases they need to be adjusted to work in the most effective way for a specific implementation. (Farenden, 2012.)

Figure 2 presents the order in which the phases of ITIL process should be carried out. Service Strategies in the middle means the specific plan that company has for a service that is the subject of an ITIL-based project. Company wants this desired service to perform a certain function in support of an underlying business process and thus Service Strategy needs to be kept in mind at all times. Three arrows with the names of three phases; Service Design, Service Transition and Service Operation, signifies the order that they are carried out on. This cycle is will also be re-entered if service needs redesigned or replaced by something totally different. Lastly the outer ring, Continuous Service Improvement. It is shaped like a ring because it is included in the background of three previously mentioned phases that are the arrows. All people involved in any ITIL-process need to constantly be thinking on how to increase the efficiency of said service. (Farenden, 2012. Chapter 9.)

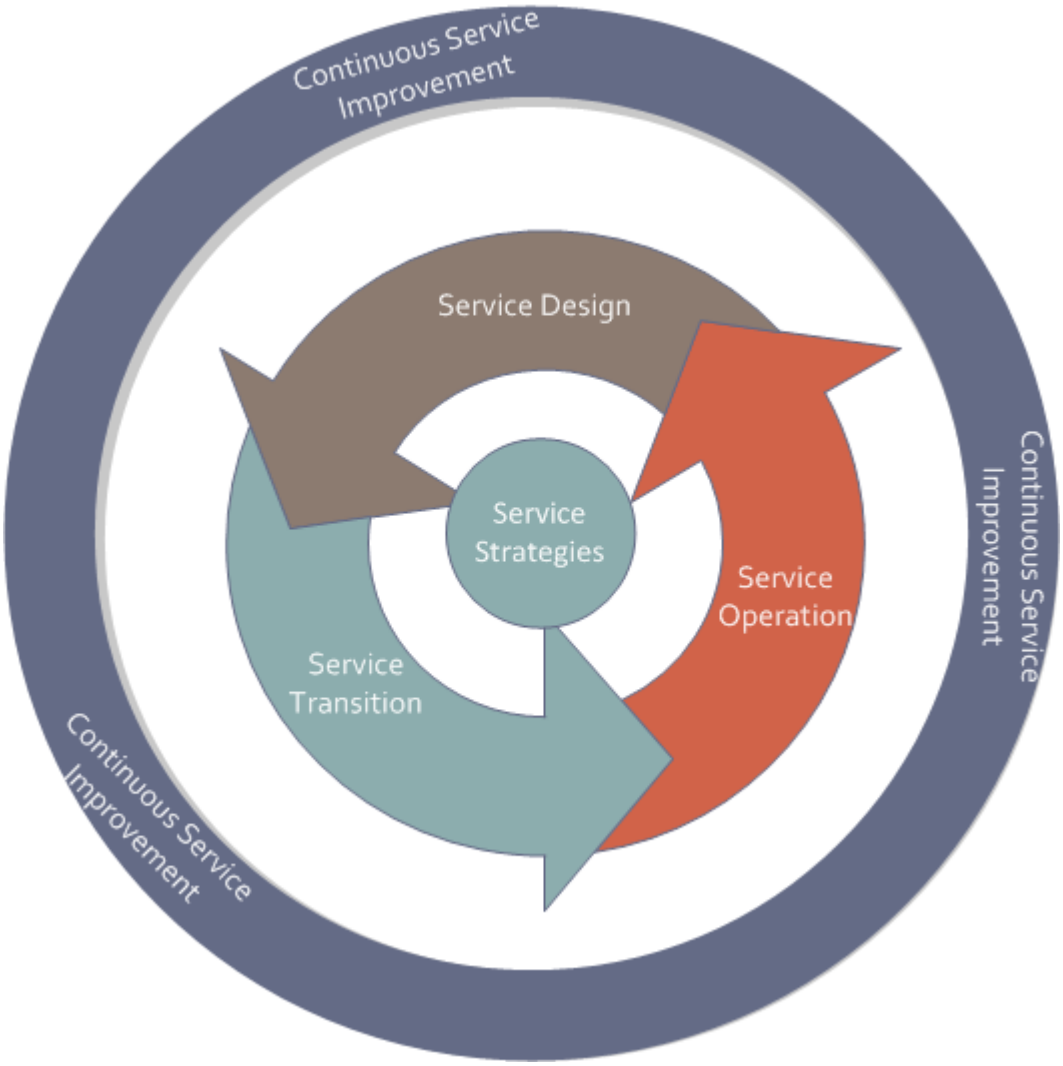


Figure 2. ITIL wheel

ITIL is divided into five different main books (core publications) and to an undefined amount of complementary guidance publications. Complementary guidance publications provide specific guidance to a specified industry sector or type so they can be very useful to an IT -service manager whose company has never before applied service management or EA to their environment by providing already discovered unique specifics that the said type of industry should take into account while planning to implement EA. The core books, on the other hand, are the basis that the essential processes and best practices of ITIL consist of. These can be divided by the phase of service management that they focus on. The books are named after those phases, as follows: Service Strategy, Service Design, Service Transition, Service Operation and

lastly Continual Service Improvement. The interconnected nature of the first three is pictured in Figure 2 above, while Continuous Service Improvement is seen as working in the background of all of them because is there not always room for improvement? (Farenden, 2012.)

Service Strategy goes through how IT -services can create value in reality and how they can be linked to support the business processes of the company and how to create and implement a clearly designed strategy for the services IT-department is producing. Service Design focuses on describing the services themselves: defining the elements that make said service and the requirements a specific service has to meet to be deemed successful. The descriptions created in this phase use the specification created in Service Strategy as a starting point so the requirements are met. The proposed service can be completely new or modified, improved version of pre-existing service. While Service Strategy state focuses on high-level business goals that drive the change, Service Design state is where those proposals get more concrete, palpable forms. In this phase the person/team in charge of the service creates plans and conducts resource estimations in order to understand what is required from IT personnel to reach the business goals. The research also includes putting together specific service requirements, determining whether they are achievable and finally designing the service based on those requirements. (Farenden, 2012. Chapter 4.)

When a service is designed, the work should always be initiated by some business need. A new or modified service needs to serve a business goal so it can increase the efficiency of organization and take the most out of IT-employees, creating services merely for the sake of technology alone does not make sense from the aspect of the business goals. How is the level of service that the customer receives, measured and managed? ITIL takes a very easily adoptable approach on determining the successfulness of a service with different kind of contracts. There are three types: Service Level Agreements (SLAs), Operational Level Agreements (OLAs) and Underpinning Contracts (UCs). The difference between the three is the parties of the agreement/contract, SLA is established between IT service provider and the customer, who could be an organization or another company. OLA, on the other hand, exists in between IT department and some other department inside the same company. Underpinning

Contract differs from the two previously mentioned types rather significantly: it exists between the service provider and a third-party supplier. The idea is that the third party supplier produces goods or services that support the delivery of said IT service to a customer. So basically a section of the service is outsourced. When IT service is defined with these type of contracts, the customer and service provider both have a clear understanding on what they can expect to get and what kind of sanctions they should expect in case contract terms aren't getting filled. SLAs can be roughly divided based on the services they cover and so-called customer base. Service-based SLA means that it defines only one service that is used by multiple customers/departments/entities. The other suggestion is customer-based SLA that defines all the services that a single customer uses. These two methods can be used to simplify keeping track of SLA requirements because they can decrease the amount of needed SLAs significantly. If there are multiple customers and multiple services are offered by an IT department in a situation where every customer uses every single one of our services, IT could end up with the amount of SLAs that would be product of the amount of customers and the amount of our offered services. (Farenden, 2012. Chapters 4-6.)

Service Transition deals with the creation of service, implementation plans and the actual implementation of a service as well. The service itself and practical plans are built based on the documentation created in Service Design phase to make sure that the service fulfills the set specification. Service is tested in action during Service Transition phase. Usually a service does not go to production environment right away: it requires careful plans on how to carry out possible migration of data from the old system if the new service is to replace something that already exists. Some form of testing is most always done as well: smaller scale experiment on how the new service works in specific situations such as interacting with other systems in place or getting an input of data from some other system or a user that is not precisely in expected format. These are all good notions to make and test, rather in Service Transition phase than in Service Operation phase with service already fully in production environment. Most important output of Service Transition phase is a new or modified service that can then be managed in the next phase, Service Operation. (Farenden, 2012. Chapter 7.)

Service Operation consists of the day-to-day tasks of service lifecycle: events and issues are managed, problems solved, users get support when needed and access to services is also supervised. Service Operation phase and Continual Service Improvement phase are closely related but they do have one distinctive difference between them: where Service operation manages help desk and guides users in problematic situations, Continual Service Improvement will try to find ways to better the service in more permanent manner. Practical example is in order here: when user encounters an minor error or glitch that makes the job that needs to be done harder or impossible to perform, it would most likely fall under Service Operation and help desk would find a solution that's eases the situation, or in best case, remove the problem completely. If this same error would be wide-scale and happen on multiple users, possibly even in multiple offices, Continual Service Improvement would step in. With huge problems re-designing the whole service instead of fixing it multiple times in exact the same way would become more rational solution. (Farenden, 2012. Chapter 8-9.)

3.5 The need for Enterprise architecture

3.5.1 Benefits by TOGAF

TOGAF states that the reasoning behind developing good, detailed Enterprise architecture is a sum of multiple beneficiary factors, and all of them can be seen directly in the overall profit or unplanned expenditures of the company in question:

- Efficiency of business operation
    - Lower business operation costs
    - Increased agility of the organization
    - Making the talent inside the organization known, more effective use of Human Resources
    - Lower change management costs
    - Flexible employees
    - Productivity increases
- Enhanced IT-operation
    - Decreased software development, support and maintenance costs
    - Better application portability
    - Easier operability between systems and easier management of system and network
    - Easier to handle crucial enterprise-wide issues like security

    - Upgrading and exchanging system components becomes a simpler task
- Protection of investments with increased return of investment and lower risk in the future investments
    - Overall complexity reduced
    - Return of Investment maximized
    - Company has a clear image of how different IT- and the business solutions overall should be carried out (in- development, buying from contractor, complete out-sourcing)
    - With the thorough knowledge, risks associated with new investments are reduced
- Better procurement
    - Unified view provided by TOGAF makes buying decisions simpler
    - Speed-up of procurement without losing the overall architectural view
    - Diverse, multi-vendor systems become a viable option

(TOGAF 9.1, 2011, 6-7)

While considering these statements, one might wonder why all companies are not working on their own EA? The benefits surely look great and well worth the investment. The interest in EA has grown steadily and the trend seems to be on-going. The extensive effort required to build comprehensive Enterprise architecture is still an indisputable fact and most probable deterrent for the deciding body of a company. Other reasons for the unfriendly reception can be though as well: maybe a prior bad experience with a similar project exists? That kind of memory sure has potential to cause prejudice towards any architecture work since the bad experiences tend to linger on and be remembered, while good ones might not get spread around enough. That would be reasonable assumption because if the enterprise using EA is working like a well-oiled machine, it would just keep on going on that governance path, which is why consultant companies offering EA architects and tools for building architecture present their satisfied clients as case studies, in order to show that working EA is worth pursuing.

The first three points under the Efficiency of business operation are rather self-explanatory: knowing the skills employees possess enables those skills to be benefited in full for the good of the company. Employees are also prone to be more productive and flexible with their work when they get an impression of appreciation from the employer, resulting in increased productivity of Human Resources. Thorough knowledge brought by EA also increases agility since the ability to react on changes

and problems increases because of it at the same time while reaction times take a dive. Enterprise architecture also helps to avoid the great unknown: there is always a possibility for a hidden risk that no-one has thought of. Because the risk is hidden, its impact on project timeline and costs can't be assessed and this is a huge problem for an enterprise that does not know what makes the enterprise, consisting of all the little pieces, tick. Having properly made EA, an enterprise can then minimize the found risks and lower the possibility of unknown risks to the lowest possible point. All of the risks can never be fully invalidated because chance is always present, but with EA it is possible to lower the chance to the smallest possible value, and that kind of state of being is undeniably something that seems worth pursuing. (TOGAF 9.1, 2011, 313-317)

Making the IT-operations more effective fall back to the thorough knowledge. Seeing the holistic picture of the whole enterprise gives person looking at the EA very detailed description that could be described as a spider web: all the pieces and processes that make the enterprise are interconnected. Some of them are connected directly, others via one or more adjacent pieces, represented by the knots of the web. No matter which type of connection exists, they are all part of the same complex and a change in one might have secondary effects on others. When EA is complete, not only does a person looking at it know all the pieces (people, infrastructure, all different kind of software in use), but also how they operate and interact with each other, exchange information and simply put: function. Once that knowledge exists on a document and not just in the head of employee or employees, the benefits present themselves. By looking at the document, only the effects of a change to specific part of the enterprise can be figured out, what is needed to take into account concerning the other pieces it will affect. The amount of unknown problems plummets when this kind of planning can be done at an early state of change management. With the same knowledge it is possible to assess all the IT service management and monitoring processes in place inside the enterprise. That way the IT –department can determine whether they are sufficient so that the desired level of service is being reached and in case this is not happening, the processes can be tweaked to do a better job or even add more oversight if there is not enough of it. Application portability increases

also, because knowing all the components upside-down makes transferring an application very much simpler task, making other research of it completely unnecessary. This will of course be applicable only if EA documentation is up-to-date.

3.5.2 Software size assessment

To truly understand benefits posed by TOGAF, or any other EA-framework, we should use some other tools should be considered in planning as well. Very few managers would just blindly believe that all the statements of benefits given by TOGAF are true in 100 percent of situations and reserve company resources for the project with that justification, especially because they are given by one single organization, most likely promoting their own view. Therefore, some other ways are needed along with the frameworks themselves to back up the posed claims. An objective way to measure the complexity and based on analyzed complexity, assessing whether a need for implementing EA would be in order to manage it better sounds like a working starting point.

One possibility for conducting the needed research of a specific company is Finnish Software Measurement Association developed FiSMA 1.1-method. It allows the assessment of the functional size of a software in a very logical and quantifiable matter: FiSMA produces a simple numerical value that reveals the comparable representation of size and complexity of the software. Knowing this value provides a group of other possible uses. Although the primary uses are for estimating costs and effort of a software development project, it has also proved its usefulness as a part of project planning process. That aspect makes it applicable for EA purposes as well, for the EA implementation requires a very thorough analysis as a part of it. FiSMA approach fills the voids left after researching the three previously mentioned architecture frameworks, for FiSMA is ready to be used after the building of Enterprise architecture is ready for it. While others offer rather wide-ranging concepts for the architecture team to decide and design, functional size measurement has the basic building blocks laid out, making size measurement and software complexity analysis very straightforward. (SFS-ISO/IEC 29881, 2013, 7)

FiSMA has been primarily developed to support software development planning projects and assessing the functional size of an already existing piece of software. The predecessor of FiSMA, Experience 2.0, was in use from 1997 until 2003 and for that reason FiSMA has significant similarities with it. Because of the relationship between these two methods, FPA results gotten from older Experience 2.0-based analysis can be converted to match with FiSMA-based work and vice versa. The only requirement is that the originally used source data has been collected at the recommended detail level or above it. FiSMA's value constraints are derivatives of the ones Experience 2.0 used and have also been proved to be correct statistically. (SFS-ISO/IEC 29881, 2013, 7)

3.5.3 Complexity

Managing complexity is crucial in maintaining the company costs and maintenance of different IT systems at an acceptable level. If overlapping pieces of software exist, an assessment should be carried out to find out if some of them could be replaced to create a more homogenous environment, the variety of systems would decrease and IT could focus their expertise on maintaining the remaining systems and becoming more specialized in them. Having multiple solutions for the same purpose drive up the costs because some applications might have specific requirements in terms of information security or management of said software and overlapping is only increased in that case. This is one of the ways to also lighten the administrative burden and give IT a chance to focus on the inner mechanics of the systems themselves and furthering their targeted expertise for the good of the company instead of administrative duties, such as keeping track of the differences of different software/platform in a general level. (Hausman, Cook, 2011)

A generally acceptable way to calculate increases of cost and support is mathematically squaring the amount of equivalent technological solutions in use, which means that the use of two different desktop computer operating systems would roughly mean four times the cost and support effort (2 x 2) compared to sticking to an environment where users have to use a single operating system. This claim can be applied to everything calculable under IT-department: applications, programing languages, even hardware is included. (Hausman, Cook, 2011)

3.6 TOGAF structure

3.6.1 Architectural domain overview

TOGAF Enterprise architecture has four so-called subdomains, divided by the scope of architectural aspects they focus on. The first one, Business Architecture, describes business aspects such as goals, the governance of the enterprise, organizational structure and crucial business processes. The second one, Data Architecture, is meant to describe how data and devices that manage and store it in the organization are placed and governed, both logically and physically. The third section, Application Architecture, includes instructions for deployment of each application that is planned to be used. Detailed descriptions of their interaction and how they link with the core business processes of organization in question is also key part of this section. The fourth and final section, Technology Architecture, is a specification sheet of the first three sections and is meant to give description of logical software and hardware requirements that are needed when implementing business, data, and application aspects of EA. Laying out the infrastructure, needed middleware, and standards are good examples of information that would be included in Technology Architecture. At the end of each of these stages the architecture team holds a stakeholder review so that all the parties are involved in the proposed work and possible changes and stakeholder concerns can still be taken into account by modifying the architecture before finalizing it. (TOGAF 9.1, 2011, 10)

3.6.2 Business Architecture

Business Architecture is the one that all the other three architectures in TOGAF builds on. Since the EA methodology overall aims to integrate business goals more closely with IT, business needs are the starting point and foundation for the technological solutions. Hence, no other architecture work can begin before the Business Architecture has been finished. Some enterprises might have less work in Business Architecture phase than others though, other prior organizational processes may already include ground work, such as strategic business planning or re-engineering of business process for example. Based on pre-existing business process and other related descriptions a scope for this phase is determined. Business processes are, in

most cases, described in some manner; however, scope definition also includes getting familiar with what exists already, to see if some sections need updated or adjusted to fit in EA process. An enterprise might even have some other business planning process already in place that has its own individual lifecycle. In that kind of situation an assessment is needed to determine if some parts of the Business Architecture could be carried out in the process that is already in place, or if a transition to TOGAF-based Business Architecture development would produce better results. Because business strategy typically defines what the enterprise should achieve – goals and the drivers that state why those goals exist and also metrics for measuring when goals are met. It does not, however, usually explain how those goals can be achieved, the means for success, which is exactly where Business Architecture steps in. (TOGAF 9.1, 2011, 80)

If the enterprise has little or no prior work done on business matters falling under Business Architecture, the architecture team needs to research them, verify their findings and finally gain approval from key stakeholders, like high executives that their mapping of the needed business processes and goals includes the ones intended. The business scenario technique from TOGAF ADM can be used for this kind of work. (TOGAF 9.1, 2011, 80)

According to TOGAF it can also be concluded that Business Architecture, if done properly, is a major help in getting the stakeholders (a person or a group of people who are in some way involved in the mapped process) to approve the architectural work and participate in it willingly. If common sense and logic are used here, fulfilling business goals is what gives an enterprise the best chance of staying in business, growing and overall being healthy. From that alone a conclusion can be drawn that business goals should be the first and foremost matters that every employee aims towards with their personal effort, which is why presenting the Business Architecture, and how their work contributes to it helps. Gaining approval of the stakeholders makes cooperation and collaboration in architecture projects an easier task when the benefits of knowing a bigger process, seemingly insignificant to their daily job duties, can be shown to have an effect on them specifically. This could mean for example bonuses that are related to production records or having zero accidents.

3.6.3 Data Architecture

Data Architecture requires major planning effort considering data management in an enterprise. The posed approach should be constructed to have a well-defined structure and holistic approach on data management since effective use of data is what enables an enterprise to capitalize its competitive advantages. Turning knowledge to profit can only be succeeded if data is available quickly and in organized manner. (TOGAF 9.1, 2011, 97)

Data Architecture also includes classifying application components into one of the three defined categories: system, record and finally reference. The architecture team should also consider whether the enterprise has a need for enterprise-wide standard that every single application component should comply with, since this might not be a possible or even preferable outcome in all cases. However, probably the most essential part of Data Architecture is the data itself, meaning the definitions about how and when enterprise data entities are created, stored, transported and also reported. To elaborate this even further, Data Architecture should also describe clearly the level and complexity of data transformations that enable information exchange needed between different applications. (TOGAF 9.1, 2011, 97)

The major benefits of good Data Architecture are gained when a migration process needs to be carried out to replace an existing application. Well-done Data Architecture defines the specifications that need to be met for the migration to be successful and how the migration of application data specifically (master data, transactional data and metadata). This data migration is delicate since different applications handle data differently, Data Architecture should also have instructions on how that data needs to be parsed so the new application can identify it and is able to process the data properly. Application usually needs data to be in very specific format in order to understand it as intended. The ease on drafting procurement requirements can also benefit from Data Architecture, for both internal applications and application interfaces in between the enterprise and its partners. (TOGAF 9.1, 2011, 97)

Simply put, Data Architecture maps out the lifecycle and various transformations that data is required to go through in an enterprise environment when transmitted be-

tween different applications. Handling of data through different systems, data formats that systems require to be able to use it and classification of different types of existing data have strong presence.

3.7 Application Architecture

Data Architecture is followed by Application Architecture. Application Architecture focuses on mapping existing applications in the enterprise. Complex applications can be broken into two or more smaller ones for the sake of this phase, making it easier to understand their operation. Relationships between applications should also be defined, for example by creating matrices that map applications back to the business functions and processes so that logical grouping is possible. Having all the applications in one big pile does not help in managing the architecture for an employee who is not already familiar with the whole, grouping is the key of managing them. Bringing in new employees or even switching software developer contractors becomes a lighter task with detailed Application Architecture. (TOGAF 9.1, 2011, 112-113)

If multiple applications serve the same purpose one adjacent application, duplicates should be removed, or in case of it being a part of a bigger application, disabled in some way. Application Architecture phase also includes mapping out concerns that rise from integration, migration and development. Defining how a change in an application affects its functions, what kind of possible changes does that involve in interacting applications, how can application migration be carried out with minimal downtime and make sure that the risks involved are minimized? These could be questions usually asked when going through this phase. User base and organizational dependencies of applications need to be figured out. These will be of utmost importance when a change of some sort is being planned and the impact on users and departments needs to be figured out in order to continue the change planning. From everything defined in Application Architecture phase, a target architecture will be formed. (TOGAF 9.1, 2011, 112-113)

3.8 Technology Architecture

The last of the architecture domain phases is Technology Architecture. Technology Architecture phase is meant to unify all the architectures and all the information that was built and gathered in the first three architecture phases considering Business, Data and Applications in both Baseline and Target Architectures. After all the information has been gathered, Target Technology Architecture is derived from it. In addition to this data and architecture models alone, the architecture team also has to consider how to best enable the cooperation between logical aspect like applications with the physical aspect such as servers and network infrastructure. Already existing IT services of the company probably have some existing documentation considering their operation and those are important inputs for Technology Architecture as well. Generic technology models relevant to the needed service and/or specific industry can also be beneficial in this phase. The usual gap analysis and stakeholder review follow after Baseline and Target Technology Architecture definitions are done. Finalizing the Technology Architecture-section is of utmost importance because everything the team builds in Technology Architecture phase needs to be finished in that section of this phase and that includes all of created architecture scopes since after this phase they should not be modified anymore. All the building blocks of this completed architecture are fully described according to the desired scope of the whole Enterprise architecture building effort. After this phase the described target architecture is pursued and implementation work begins. (TOGAF 9.1, 2011, 120-128)

3.9 Process for Architecture phases

Each of the previously mentioned four architecture development phases is divided into smaller steps to make it easier to understand how they function. They all start with picking out the reference models, viewpoints and tools fit for the phase in question and the specific EA case. Reference models are especially useful if the enterprise does not have an existing EA, and for that reason, the work has to start from scratch. After all the initial steps have been gone through, the description of the Baseline Architecture is developed. Baseline Architecture pictures the current situation of the said architecture domain, as the name suggests, so the person or people responsible

of EA development can understand what their starting point is. Next up comes Target Architecture phase. In this phase the desired state of the specific architecture domain is outlined and described in great detail. At the end of each iteration of EA development Target Architecture is the goal that the whole team aims to achieve. (TOGAF 9.1, 2011, 48)

After forming Baseline and Target Architectures, gap analysis would follow up. Gaps mark the figurative distance between Baseline and Target Architectures defining the amount of work needed to achieve the state that the latter describes and the size of gaps can vary among each other significantly, even inside a single phase. The gap analysis is followed by defining candidate roadmap components, which means creating a roadmap guiding the whole ADM process by defining how different activities should be prioritized when going through it: the roadmap is being initially defined in each of the three architectural phases (Business-, Information Systems- and Technology Architecture). (TOGAF 9.1, 2011, 48)

Next up the resolving impacts on Architecture Landscape are dealt with. In short, this means finding solutions for possible challenges that arose after previous states, making proposed architecture presentable to the stakeholders that would not appreciate this architecture building effort if it created high number of problems when it may be still hard for them to fully understand the benefits of it. Keeping stakeholder opinions pro-architecture will makes working on it so much easier. After impact resolving comes formal stakeholder reviews. Here their concerns would be addressed and their approval sought. The reviews are followed by the last two stages: Finalizing the Architecture and Creating Architecture Definition Document. Finalizing is self-explanatory; however, the following stage needs elaboration. Architecture Definition Document contains the planned Architecture and it is published for the whole company. (TOGAF 9.1, 2011, 48)

Since this thesis project is aimed to produce a document describing how Pyhäsalmi Mine Oy specifically would benefit of implementing EA, and the project was proposed by the IT department in the first place, Business Architecture will gain less attention in the analysis itself. The analysis also has a main goal of combining a working model of the different frameworks so sections of each one may have been omitted

on needed basis presenting and explaining it was deemed unnecessary. Full framework documentations are publicly available online on websites of Open Group for TOGAF and ICANA for COBIT, ITIL being available only in form of published core books and supporting publications.

3.10 Architecture Development Method

Figure 3 defines the structure of full ADM cycle, instructing on the order in which all of the phases are carried out and how all the decisions that are made in every phase have to be based on the business requirements. Business requirements are to be monitored throughout architecture work through constantly present Requirement Management phase.

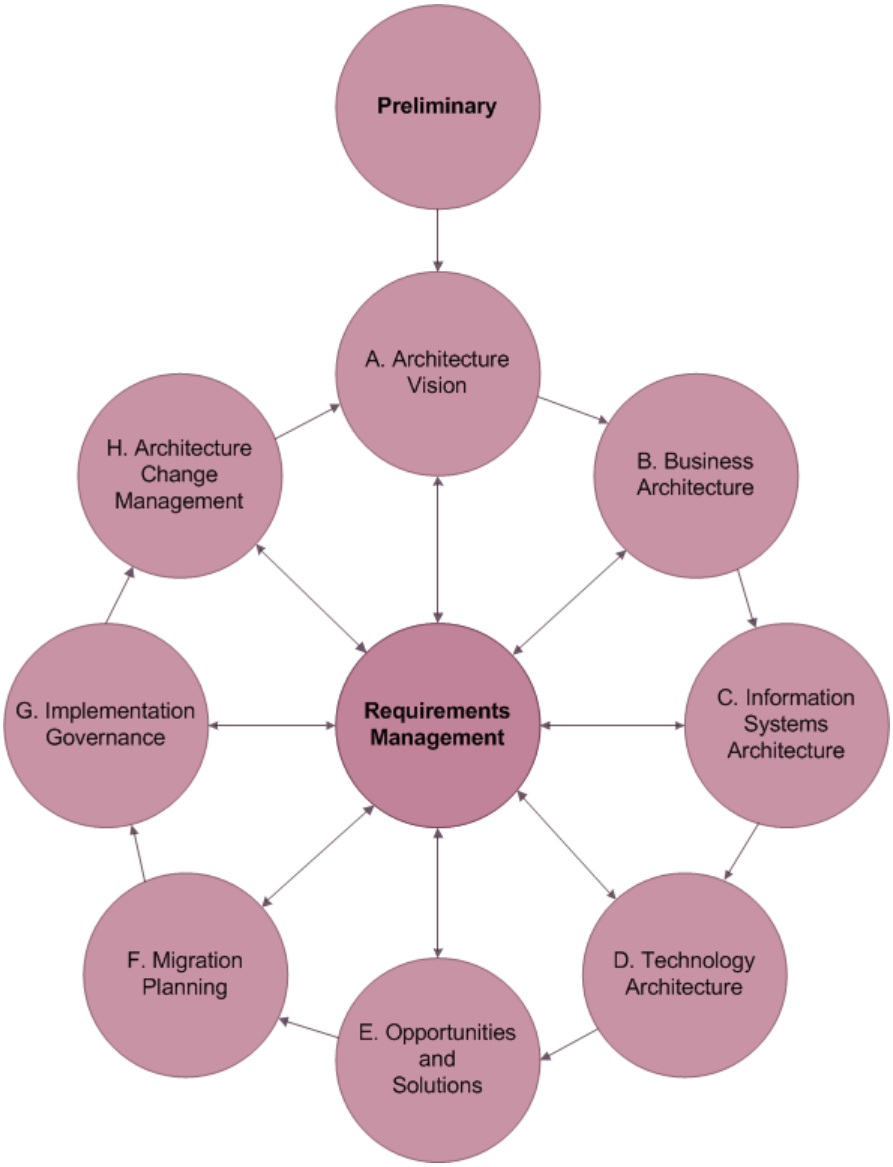


Figure 3 Architecture Development Method

ADM is one of the most important components of TOGAF process. All four previously mentioned subsets of TOGAF included in Architecture Development Method (ADM). ADM is a key process of TOGAF framework since the creation and major modifications of the EA are handled by going through an ADM cycle. In ADM these four previously mentioned phases are merged slightly: Data and Application Architectures are joined together under the Information Systems Architectures phase. This change is only superficial since both architectures will be created individually inside this said

phase so beyond that, it has no practical impact. In addition to the three architectural domain phases, ADM cycle also includes five other phases plus the preliminary phase. Prior to Business Architecture phase there is the Architecture Vision phase, which defines very broadly the scope of the architectural work and constraints needed to be taken into account. TOGAF uses version numbering with two values to signify the state of specific subset architecture. The four different subset architecture descriptions formed in Architecture Vision phase all carry the first value, version number of 0.1, which means that the formed architectures are high-level outlines instead of highly detailed descriptions. In three subset scope phases (Business-, Information Systems- and Technology Architecture) that are specific for each architecture subset, new architectures are formed and they all receive version number of 1.0. This version number means that they are on the detailed level. (TOGAF 9.1, 2011, 45, 70, 161)

After all the subset architectures are finished, Opportunities and Solutions phase is in turn. This phase consists of two objectives: generating the initial complete version or the Architecture Roadmap and determining the need for incremental approach in reaching the target architecture. Architecture Roadmap is basically the way how architecture team reached the state described by Target Architecture. It is a very practical plan that explains all the work which results in changes that are part of building the desired architecture, neatly divided into smaller steps. Architecture Roadmap is built based on gap analysis and candidate components that were defined in all of prior subset architecture phases B, C and D shown in Figure 3 above. An incremental approach provides a whole new concept, Transition Architecture. In some cases, the gaps between Baseline and Target Architectures are just too wide to reach on one "jump" which means that team would need to define Transition Architecture. It acts as a midpoint in reaching the Target Architecture state, so implementation will be done twice if there is one Transition Architecture planned. More than one is a possibility. The team needs to be careful when planning Transition Architecture because the creation of business value needs to remain continuous even in Transition Architecture. (TOGAF 9.1, 2011, 132)

Migration Planning phase is next up. It focuses on finalizing Architecture Roadmap and Implementation and Migration Plan that is meant to support it. Migration Plan

will also be checked to make sure that it follows the guidelines of the change management process that the specific enterprise uses. The architecture team should also make sure that key stakeholders understand the business value and cost associated with Transition Architectures and measures to be taken to make the transitions between different architectures. Implementation Governance phase follows Migration Planning and is the second-to-last phase of the ADM cycle. In Implementation Governance phase team makes sure that the planned implementation projects are eventually going to reach Target Architecture and any possible Transition Architectures prior to it. Incremental approach is recommended in the standard; breaking down the job into smaller tasks makes the overall process easier to manage. In this phase measures of effectiveness are also agreed on in cooperation between architecture and implementation organization if they are not one and same. (TOGAF 9.1, 2011, 132)

4 Adaptations/Adjusting EA

Successful use of Enterprise architecture does not work just by following what the frameworks instruct to do; like it would be the only proper way to create an architecture. Building an architecture includes adapting the used framework or frameworks to the specific company because every company has some unique characteristics to be taken into account, and they are very unlikely to repeat themselves exactly as the same kind of mixture in any other company alike. Ideas about what an architecture team should consider can come from the industry and should be analyzed, as long as the balance between taking advice from what came before and thinking on one's own is preserved. After all, the employees of a company are the most likely people to have the first-hand knowledge when it comes to the uniqueness of their employer.

4.1 JHS 179

Public administration of Finland has created their own adaptation of the Enterprise architecture, going by the name JHS 179 (Julkisen Hallinnon Suositus 179, Public Administration recommendation number 179). JHS 179 is a recommendation that gives directions to public entities in Finland, such as health care and Local Register Offices,

about how they should be building their Enterprise architecture descriptions, creating models based on those descriptions in order to streamline their processes, standardizing the way technical solutions are built so compatibility between systems of different government bureaus increases, interfacing becomes a simpler, cheaper task and therefore the availability of said services grows. Services in question are used by the agencies themselves as well as by the Finnish public. The reason for creating architecture guidelines specific to Finland is most likely a product of few issues: legal requirements for steering the development of IT systems in public bureaus to be highly cost-effective while also demanding ease-of-use of their services, followed by developing thorough architecture descriptions to ease governance, management and development of those complexes because of their massive size. Language was surely one of the drivers too, since TOGAF does not exist in Finnish. (JHS 179)

JHS 179 is largely based on TOGAF approach as can be deducted from structure it uses, as illustrated in figure 4. JHS uses the same division to Baseline and Target Architectures to signify the current state and the goals we want to reach with EA methods. The iterative architecture process is also present, which is very close to TOGAF Architecture Development Model.

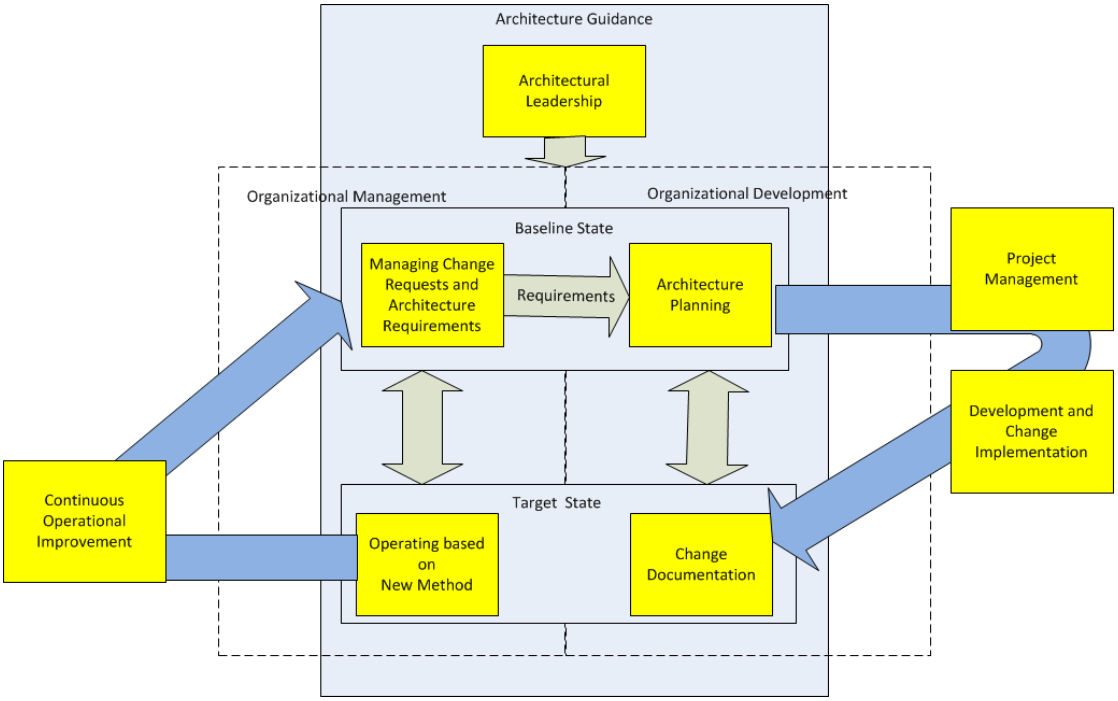


Figure 4. Organizational development in JHS EA

Figure 5 illustrates JHS 179 division of Architecture scopes and development of Baseline and Target Architectures (Business, Data, Application and Technology sub scopes), as well as how organizational strategy, also called Business Strategy, drives the direction of Enterprise architecture. The similarities to TOGAF are undeniable, as they should be when TOGAF is the framework that JHS 179 is built on. Terms used in JHS are in Finnish since JHS has only been designed to be used in Finnish Government and therefore translations shown in these figures are work of the thesis writer, not from the original document. This is why there might be some minor changes in wording that are merely a result of possible gaps in writer's interpretation of English vocabulary and should not be deemed as flaws in the JHS recommendation documentation itself.

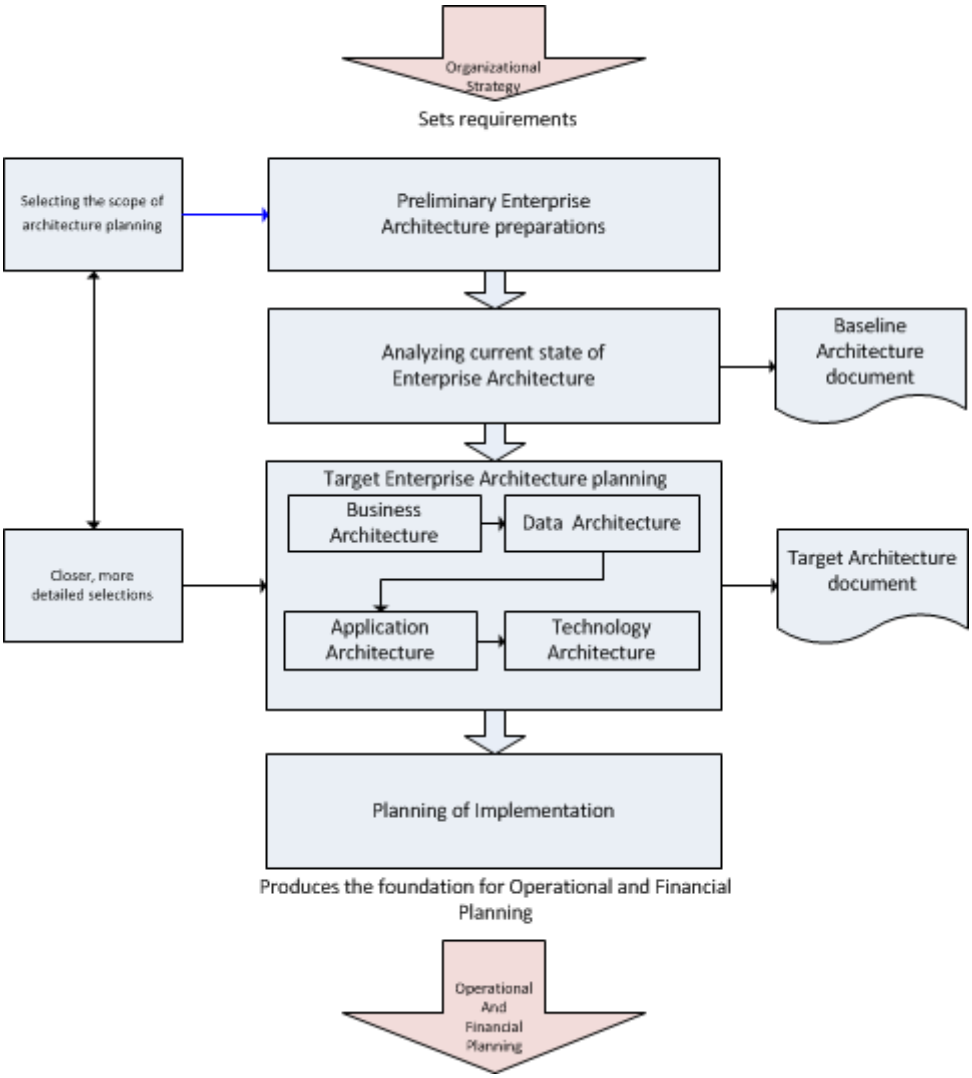


Figure 5. Division to Architecture scopes

4.2 Local Registers Offices EA

Local Registers Offices of Finland have adopted JHS implementation of Enterprise architecture for their entire organization, and as a result of this work regional State Administrative agency of Eastern Finland (Itä-Suomen aluehallintovirasto) published a document describing how JHS was used and adapted to serve their needs. The regional State Administrative agency of Eastern Finland was the one to publish it because that specific office is known as national governance and development unit and

therefore high-level Enterprise architecture falls under its jurisdiction. The document does not describe how individual administrative courts implement EA but is more a high-level description of agency-wide goals and challenges needed to be considered in between 2012 and 2015. (Local Registers Offices Enterprise architecture, 6)

Local Registers Offices are a good example of how Enterprise architecture can help in managing a large ecosystem that Local Registers Offices definitely form. Even though the amount of offices is not of grand scale with 41 total units under 11 different Local Registers Offices, the set of services the agency is responsible for maintaining is vast. (Local Registers Offices Enterprise architecture-document, 19)

Local Registers Offices are integrated into the local state government and their main responsibilities include maintaining population registers and handling custody processing. Records of marital status and name changes, aquatic vehicles, civil union processing, public notary service, prenuptial agreements - and donation matters as well as administrative process of a people relocating are other services that the agency has to offer. (Local Registers Offices Enterprise architecture, 18-19)

When dealing with the kind of sensitive information such as these, information security is also an aspect needing a great deal of work. Defining the types of encryptions used for documents and data transfers between offices via public internet are crucial in ensuring that all that information ends up only to the wanted recipient and it does not end up to wrong hands. Other aspects of cyber security are defined in the document also; such as personnel security about vacation notification policies outside of the organization and infiltrator posing as organization member with measures that are to be taken in that situation, physical security such as storing of devices and definitions of authorized access to the premises, hardware security such as locking of devices when leaving office/post and remote access to them in case of theft. These are examples of the cyber security aspects included in the document and unlike Enterprise architecture itself, are mostly based on VAHTI- and AVI-information security directives set by the regional State Administrative agency of Eastern Finland. (Local Registers Offices Enterprise architecture, 41-42)

The document lists important strategic aspects affecting the most regarding to Enterprise architecture efforts, affecting either directly or indirectly the development of

computer-based services or their constraints. This is one of the cornerstones in EA. Strategic goals have been gathered from multiple sources: the government platform of the former Finland's Prime Minister Jyrki Katainen, the strategic memo of Local Registers Offices for 2012-2015-time period and the computer-based services-strategy memo of Local Registers Offices 2012-2015. (Local Registers Offices Enterprise architecture, 43)

Aside from general points considering maximizing productivity and efficiency throughout the agency, the list talks about using open source interfaces and standards in public administration systems so that there is clear consistency, which in turn will ensure compatibility and increase cooperation between systems of different government agencies, basically picturing even broader Enterprise architecture scope consisting of the whole public administration of Finland and all its entities. Then again, that is what JHS 179 was created for in the first place, standardizing all that in order to tune the efficiency and usability of all computer-based services of public administration into their maximum value. This makes sense, since those agencies are funded by taxes and general public is not interested in wasting those scarce tax euros. (Local Registers Offices Enterprise architecture, 43)

The three illustrations below (Figure 6, Figure 7 and Figure 8) give a clearly-defined notion of how the Baseline and Target states of the process flow when a customer fills out an application for any of the services they can get through Local Register Offices. In the Target state a service does not use separate systems for identifying a customer and submitting the forms. The customers can simply log into the system and have all the services behind one login, in that one single user environment. The automation percentage of the whole process has also been increased to 100 percent concerning computer-based services in the the Target state. The increase in the amount of fully computer-based services where customer would not have to be physically present in the office is. The increase in the desired outcome. amount of fully computer-based services where customer would not have to be physically present in the office is the desired outcome. In the Baseline state, only the change of family name can be handled as a truly remote service at the moment. (Local Registers Offices Enterprise architecture, 89, 93, 94)

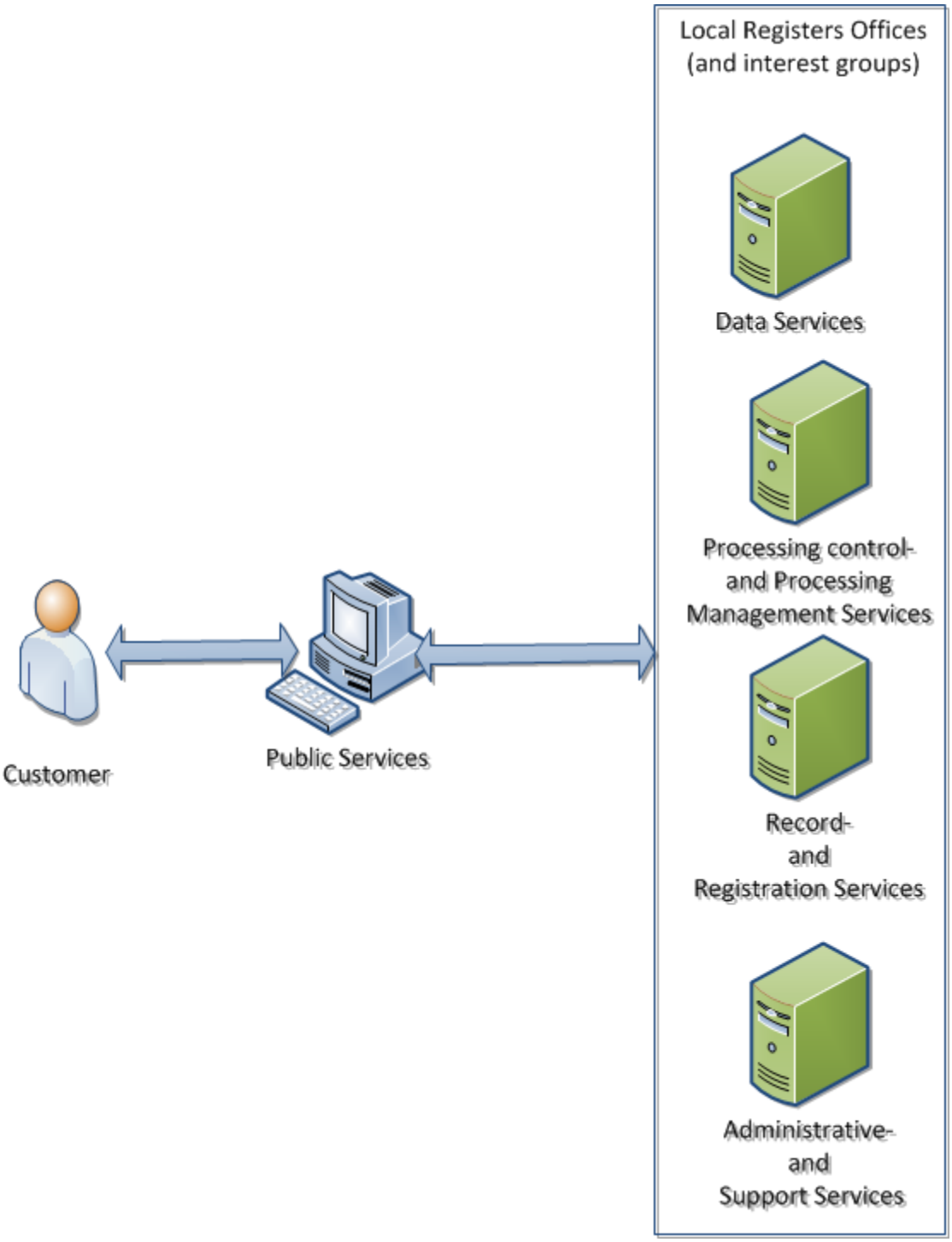


Figure 6. Centralizing Local Registers Offices services

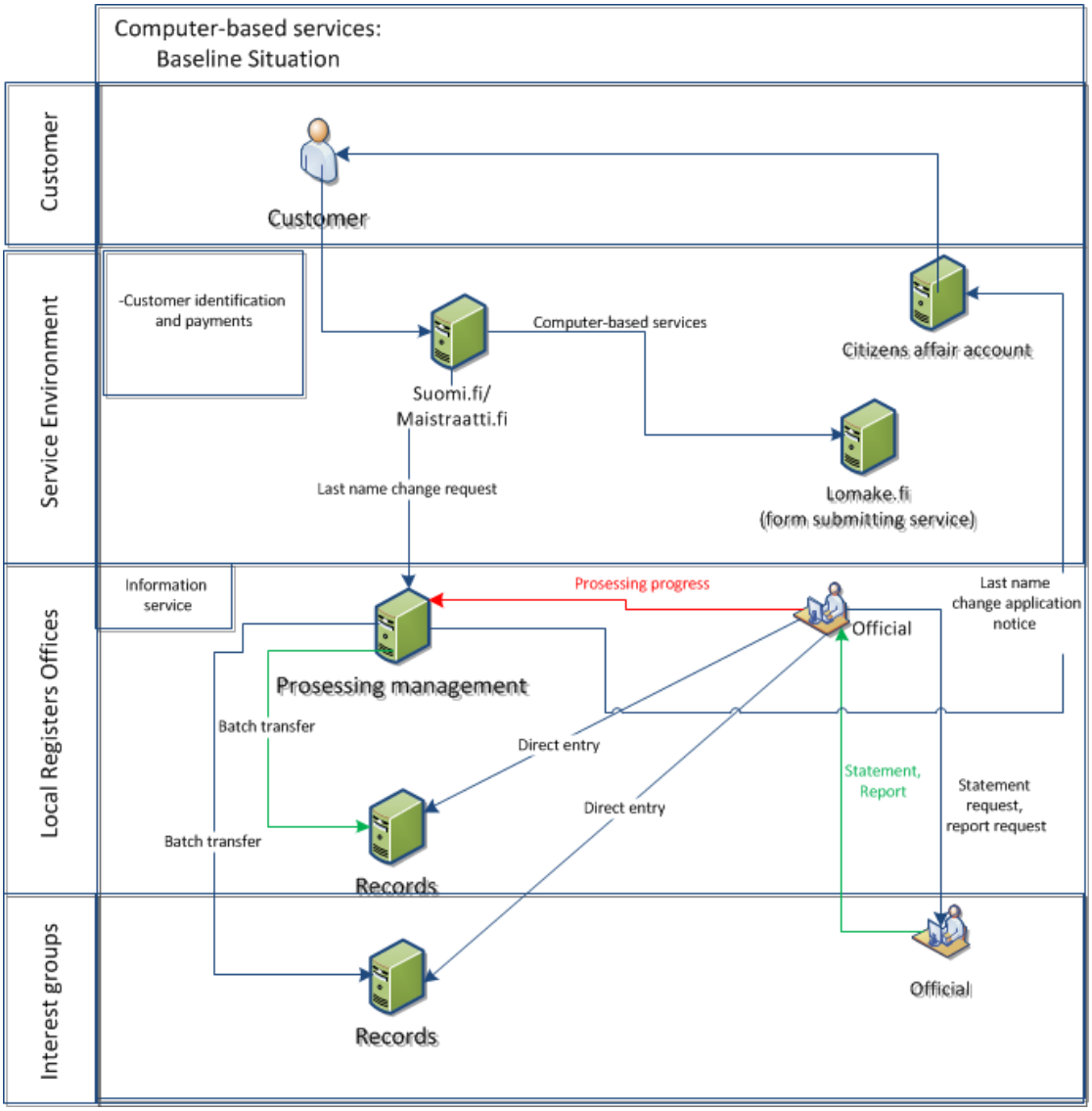


Figure 7. Current state of online services in Local Registers Offices

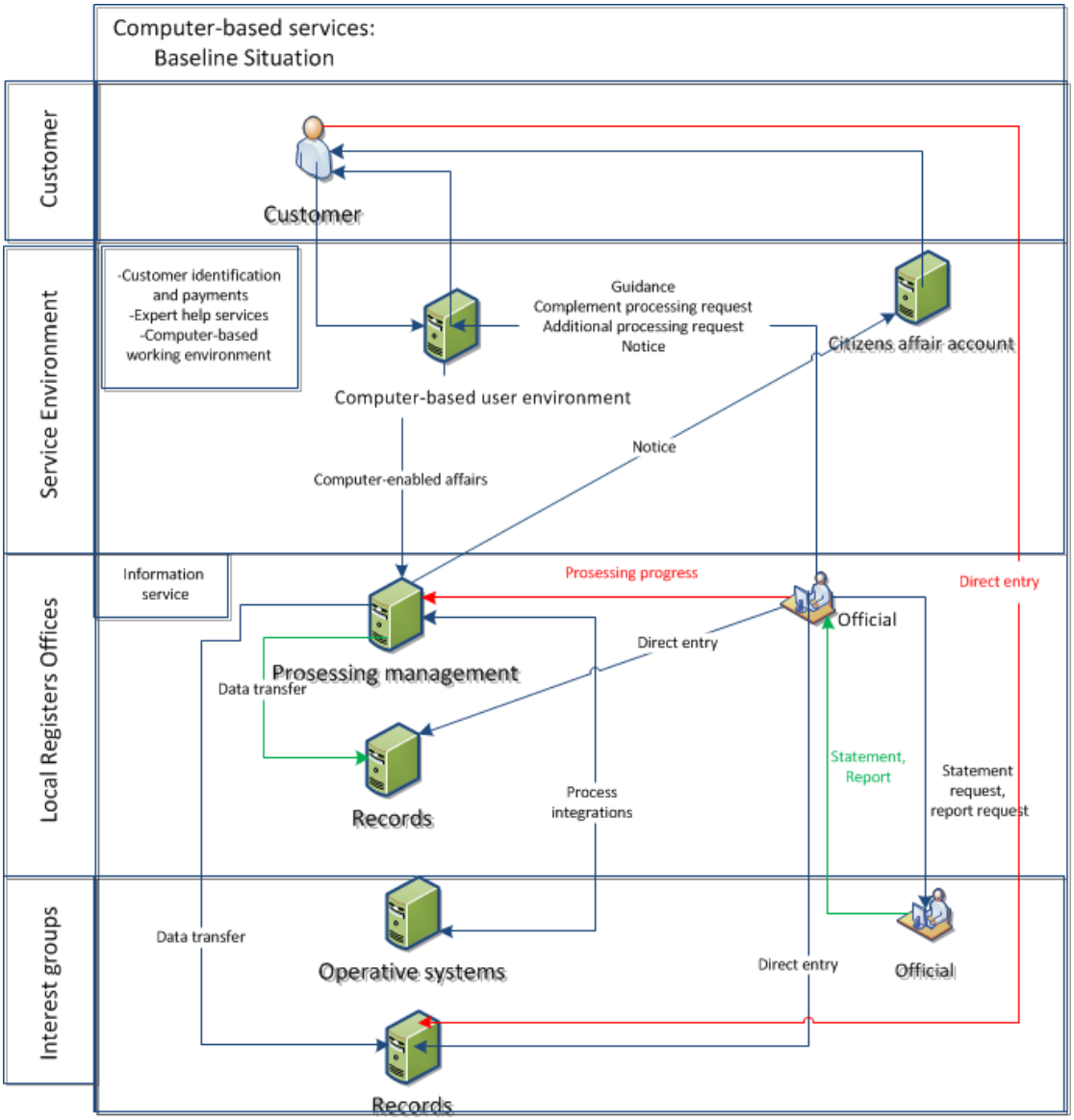


Figure 8. Target state of online service in Local Registers Offices

4.3 EA in public social- and health care services

Another example of JHS implementations is social- and health care services Enterprise architecture recommendation. This recommendation is also a high-level and general recommendation because it comes from the Department of Health and well-being (Terveyden- ja Hyvinvoinnin laitos, THL), the highest authority of health care-related policies of Finland. Central Business goals of THL are nationwide information system services, such as electronic medicine prescriptions, patient file records and

social services client records. In addition to these large-scale services, THL is also responsible of creating a policy that guides the development of all the Finnish hospitals and other health care units, including private clinics. The jurisdiction over private entities is not as comprehensive as it is with public health care; however, safety and patient personal information security regulations defined by appropriate laws apply to both groups the same. In addition to legal requirements, Vision for THL's architecture defines that social services and health care client records should always be available in a location where clients receive such services. Records should always be up-to-date, uniform, their handling needs to take place in secure systems and travel through secure channels as well as in standardized technical format. Structuring of records enables creating smart summaries based on said records. Nationwide information system services act as means for storing and transferring said data, while social services and health care organization client record systems take advantage of the broader nationwide systems. (Huovila, 2015, 3-6)

The upcoming restructuring of public social and health care services into bigger regional units illustrated below in Figure 9, called SOTE restructuring is taken as an example of how EA can simplify the service delivery from the client viewpoint: services are centralized so that clients do not have to travel around over unreasonable distances for different hospitals to receive the care they need and are entitled to. Clients also have the freedom to choose between public and private providers if they so desire. (Huovila, 2015, 11)

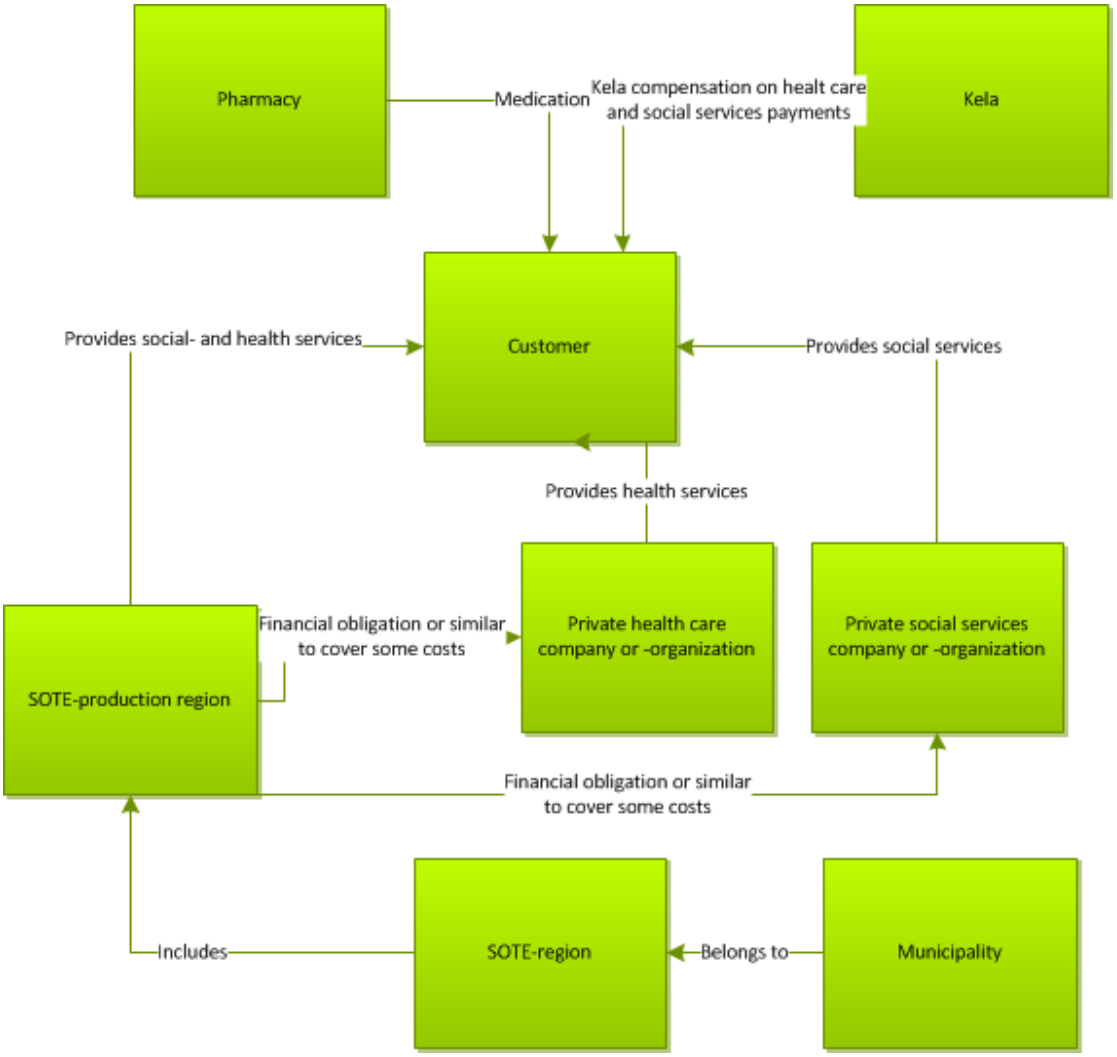


Figure 9. High-level model of health care and social services in Finland after upcoming SOTE-restructuring

5 Tools to help creating EA

Creation of Enterprise architecture is always a major effort, and starting it might seem overwhelming because of the mere size alone, not to mention the complexity found in the existing environment or even the frameworks themselves in some cases. However, it does not have to be that way since many different tools exist to aid the architecture team, speed things up and de-mystifying the whole concept of EA. The features offered vary somewhat between different choices; however, generally these programs are known by the name of Integrated Management System (IMS). In con-

trast to the architecture frameworks presented previously that are somewhat passive ways to work with Enterprise architecture, IMS software allows more reactive ways to unify all the systems and processes with each other in a company, as is one of the basic principles of EA. Therefore, IMS is not completely separate from EA frameworks but rather furthers what can be accomplished with them. IMS is a platform enabling that unifying effort to take place when systems and processes are built and described to be compatible with it. Some IMS software solutions can also offer other tools that ease tasks such as software or database engineering by generating code automatically based on database or software models. Business executives can also manage requirements for software being developed or trace progress of all ongoing different software projects systematically to stay informed about the overall situation. Some examples of these kind of software products are IMS by IMS Business Solutions Oy and Enterprise Architect by Sparx Systems. (Sparx Systems, IMS Oy)

6 Pyhäsalmi Mine Oy upcoming architecture

6.1 Starting point

Pyhäsalmi Mine Oy has not completed EA-documentation of its environment prior to this thesis project. However, when the mine was still owned by Canadian Inmet Mining Corporation in the early 2000s, an architecture project with the whole consolidated corporation as the scope following COBIT framework was initiated on every mine under Inmet ownership back then. This particular effort was never finished and with the new owner, it would have to be redone completely anyway, so that it complies with the business practices defined by the current consolidated corporation. Creating the architecture only for Pyhäsalmi Mine can also give birth to new opportunities: a smaller scale project can act as a pilot and makes it possible to give practical examples and more comprehensible presentations to the business executives of the concern so the benefits of EA can be made clear and with this kind of lobby, general opinion would change to be more favorable towards approving large-scale architecture project in the future.

The reason why TOGAF, COBIT and ITIL frameworks were picked lays partially on the prior track record considering COBIT: IT-employees already have some experience of

it so they have some base to start from. This combination also has a good spread considering viewpoints and focus on different aspects of EA overall. The most important benefit could be the openness of all of them. Open-source like development model will make sure that all the participating people get their ideas peer-reviewed and likely are being more objective in their reviews when compared to that of close-source development model where a single company with common leadership and vision has the danger of missing out on deviant ideas.

6.2 TOGAF

TOGAF framework acts as a base for the EA structure in Pyhäsalmi Mine: sections of TOGAF the most important to be included for the company are the division of EA into four different architecture scopes; Business-, Data-, Application- and Technology Architecture, along with the ADM cycle that includes it. A clear division is very helpful in the future when an architecture exists and employees need to update the architecture in projects including major changes that affect more than one architecture scope. It enables division into smaller, easier handled tasks and eases managing the workflow and the effort put into it, especially for those who are business or IT experts. ADM cycle, on the other hand, is used for the initial EA work for building the architecture when none exists yet.

During this first time of creating EA, Pyhäsalmi Mine goes through one whole ADM cycle and all of its phases. In the future the EA maintenance would not be the sole responsibility of the architecture team simply because IT department consists of less than five regular employees and it would add to their workload so much that they would not be able to handle their regular tasks. Instead, maintaining the EA up-to-date and proposing changes could fall on to the responsibility of different department heads and they could delegate it to appropriate employees even further when needed. This way the workload would be divided more evenly and it would be more manageable for everybody involved. An alternative could be hiring a single person to be in charge of the whole EA, and other employees could go on with their current responsibilities as they used to. A good example of such a company would be M-Solutions Oy, very specialized IT-company that operates locally and for that reason would also be in logistical advantage to any other consultant company. The downside with

this approach is that it would require starting a whole new vacancy, naturally with rather high salary (according to the competence of said new employee) and which is why the first option seems more likely.

6.3 COBIT

Since governance is the major selling point of COBIT, it was decided to adopt that as so it was included in Pyhäsalmi Mine EA scope. COBIT flags for IT governance being integrated into enterprise governance and that was the first thing that was sought, adapted to the specific needs of course. In development projects that include IT-related technological solutions, IT-employees need to be included more closely into the decision-making process: currently their expertise is not being capitalized on to the fullest potential when deciding software platforms that other departments will be using. Since the IT department possesses insight to the industry and has a great deal of knowledge in assessing what kind of costs specific software acquisition adds up to when it is for example interfaced with current systems, these employees should be listened to more closely. After all, that could save thousands upon thousands of euros of company money, and since that is one of the major drivers behind EA as a concept and therefore also behind this EA research project specifically, that is something that the upper management should hear about in order to have all the necessary information when deciding what is best for the company.

6.4 ITIL

ITIL statements about how SLAs, OLAs and Underpinning Contracts are a good way for managing and maintaining the level of service at a wanted level required some thorough consideration. In the end it was decided that the normal agreements and contracts for services produced by the internal IT department, the user base consisting only of company's own employees would be a working option if the company was bigger in size since keeping track of all the details gets harder with a higher amount of customers. OLAs inside the organization have the potential danger of leading to finger pointing and unnecessary accusations creating mistrust among different departments and employees when issues raise. That can also breed smaller groups of employees who start aiming merely for the success of their own department instead

of keeping the overall business goals in mind. Customer-based SLAs would also be out of the question since the ratio of IT employees to other employees is so small, service-based SLAs would be the only possible way for it to work in a reasonable manner. External contractors producing a service for the company are a whole different thing: those need close monitoring and clearly defined SLAs so that their performance can be tracked, and it can be made sure that the level of service stays within acceptable limits. SLA is a standard procedure with outside services; however, they also need to be implemented to company network monitoring software since at least availability could be easily monitored in that way.

Aspects of ITIL that were deemed to be useful were Service Design and Service Transition. Practical approach of ITIL should ease the adoption of the new upcoming Enterprise architecture model while broad business goals act as drivers for everyday tasks. From under Service Design, sections considering availability, capacity as well as IT service continuity were picked out for Pyhäsalmi Mine's EA implementation. Even though SLAs were decided not to be useful inside the organization, services still need to be monitored at some level. Capacity and availability also require planning for the future that takes into account scenarios proposing different amount of growth and maps out expansions to the systems based on the magnitude of growth predictions. IT Service Continuity Management section is a natural extension of the two previously mentioned service level aspects. It consists of management and analysis of risks with the potential of causing service outages, continuing with their impact on business functions.

6.5 Unified and standardized solutions

By creating a solution for EA on one site that can be applied with modifications to the other mines under FQML of the consolidate corporation and showing some real figures about differences of how it works in action before and after the architecture team has completed describing the current architecture, the high-level business executives of consolidated corporation can see how standardization of all the technical solutions in all the companies can benefit them as a whole. With standardization of information systems an improvement invented in one facility can potentially be used in all the others and interfacing it for others should be relatively small task in such

environment. The same works the other way of course: when an issue is found and identified, similar systems in all the other sites can be inspected to make sure they are not in risk of repeating said issue.

Standardization with company hardware has major benefits similar to software ones as well: unified solutions limited to one or two vendors have in general greater compatibility with each other and future expansions, as long as they use the same vendors. That way procurement is a simpler task and keeping tracks of multiple vendors with their different warranty policies and differing contracts does not require major administrative work and IT-employees can focus on the more productive duties. This does not mean that all the hardware has to be the same because it cannot often be standardized all the way. Some employee might need a special kind of ergonomic keyboard or Braille-keyboard. It should, however, be a rule that computers and networking equipment (routers, switches) should be fairly similar, possibly the same model if employee work duties or specific spot of the infrastructure does not specifically require something different. In a differing case, a case-by-case consideration should take place, still trying to keep it close to the de-facto standard of work computers if that is possible and rational. An example follows here: 50 of the exact same computer takes less effort than managing 50 different computer models, the needed knowledge consists of much more narrow section of the field and therefore IT employees can be specialized in it much further. (Hausman and Cook, 2011)

Mining industry has to follow legal safety – and environmental regulations same way as any other industry has to follow theirs. These two are usually the major concerns in the mining industry and so also in Pyhäsalmi Mine Oy, because the enrichment process of minerals is a chemical process needing plenty of water to function, and underground mining of the used minerals requires blasting with explosive material. Process waters are returned to the local lake after extensive purification and this process follows the environmental regulations so no harm is done to the lakes ecosystem. Following regulations is very important from the company perspective since breaking them also often comes with big fines that the violator company has to pay, decreasing its own profit margins and causing worry for its investors, which might in worst case result in the loss of funding and bankruptcy, the reason why those regulations and having them followed has to be a high priority. These regulatory matters

were also included in the EA so that their effect and required changes could be seen in a simpler way from one unified location and their progress and measurement of effectiveness towards staying within regulatory boundaries in all processes tied to Pyhäsalmi Mine could be tracked with less effort. In a project where EA is created, it will take some serious work to input all of them as a part of EA; however, in the future it only needs to be modified and updated instead of being re-designed every time there is a minor regulatory change. (Hausman, Cook, 2011)

6.6 Continuity

As IT management is seen as the risk management for IT, continuity of enterprise is one important aspect; in particular, meaning insight information that employees have about their specific area of responsibility in the company. Information that might not have been written down anywhere so that someone else could have it as well. A not that uncommon scenario would be an employee relocating and therefore, resigning from his current position, taking that know-how with him/her. The person who gets hired as a replacement has a great deal of work in finding out that information on his own if it even is possible in the first place. That kind of behavior would also take attention and time of other duties of the employee and that way cost more money to the company. Secondly, much more unlikely; however, still a possible scenario, death of an employee. In case of death all of the knowledge of an employee is lost once and for all if no documentation of it exists. These two reasons alone should be considered very carefully when deciding if Enterprise architecture implementation is worthwhile. Accidents such as this might never happen but is the remaining possibility worth it if it could be fixed with adopting EA? There is no way to measure the severity of a risk that we have no knowledge of.

6.7 Taking reference model into practice

6.7.1 Mining Industry Reference Model by EMMM Forum

This reference model, as other available models, serves a critical purpose in TOGAF EA process and it is a great help to aid the work. Starting the architecture out of zero

is a very challenging task and something that has the danger of baffling employees for quite some time, if not indefinitely. If the team does not have any kind of pre-existing guidance of that sort, work might just stop completely before it even got started for real. TOGAF reference models are a great starting point: they are widely accepted and constructed by different specialized subsections of The Open Group and for that reason act as a wireframe model, mapping out the Business Capabilities that should be achieved in the specified industry. They are reflected back to the more generalized Business Capabilities that exist in varying degree in almost all types of enterprises. In other words, the reference model provides industry-specific goals to be fulfilled and more generalized business goals that support the specialized ones. It does not explain to you how to reach any of them; however, it will give ideas of most common mining industry business aspects and results in easier building of Enterprise architecture. (The Exploration and Mining Business Capability Reference Map)

All the process descriptions existing in the Pyhäsalmi Mine so far have to be thoroughly processed and if some documentation does not exist yet, it needs to be created. Since concentrate production is the most important part of the business, being the source of profit, it needs to be mapped with meticulous attention to detail. That obviously means also every single system, and process supporting it in any way.

6.7.2 Tracking of explosive materials at Pyhäsalmi Mine Oy

The new and upcoming system to track explosive materials used at Pyhäsalmi Mine in everyday activities offered a good way to bring this EA theory into practice, even more so because EMMM reference model mentions “Handling of hazardous materials” as a core business capability. (The Exploration and Mining Business Capability Map) It is a prime example of a system that needs to be mapped in great detail since the amount of interfaces it will have with other, currently present systems is significant and the complexity of even the smallest change in it or in any related system substantial. The development project is on-going; however, it is expected to be fully in production environment very shortly.

The major driver behind this tracking service is legislation: more specifically European Commission’s Directives 2008/43/EC and 2012/4/EU. These two EU directives

created lawful requirements for a system that enables identification and traceability of explosives for civil uses. Given the dangers of explosives getting lost and/or to the wrong hands, tracking them makes perfect sense. Shortly after the first directive was ratified, Federation of European Explosives Manufacturers (FEEM) followed in 2009 by creating a guidance package to be used by its members. It includes instructions on this required system, although guidance only includes information about what sort of tags should be used and the specific format of identification codes to be used in said tags. The European Union decided to use standard two-dimensional QR-codes, which means that adoption of this code format will not be what stops transitions to the new tracking system from happening: this kind of codes can be produced with basically any regular QR-code generator. What created some challenges was how FEEM guidance does not instruct on how the tracking should be implemented by explosives manufacturers or by their clients.

The explosives supplier of Pyhäsalmi Mine Oy offered their own product as an option to implement tracking all the way from supplier delivery to the individual miner "buying" them out of mines storage. This solution, however, posed a problem: it would have required a secondary handheld computer system to be used in the storage area, which would have raised the overall complexity of the whole environment. Maintaining two different system would have significantly raised the amount of work for IT department. Even bigger problem would have been interfacing a whole new system to the environment of already present systems: namely, a system that the mine would have had little to no control over when it comes to software development and so would have been stuck with this supplier as well. For these reasons a system of their own was decided to be developed. The new system still uses only a single device from the end user perspective and the development is more straightforward without having forced extra entities to take part. These entities can also be predicted to be less willing to make small changes that would make IT department's job easier at the mine, since Pyhäsalmi Mine Oy is not their only customer.

The tracking system of Pyhäsalmi Mine Oy uses two major interfaces to other systems: Enterprise Resource Planning system and proprietary system named KaTTi. ERP system takes care of keeping tracks the amount of supplies, spare parts and other

consumable storage items. KaTTi is a whole other system: it is a data warehouse system and includes a massive amount of different types of crucial business data. KaTTi could be called the heart of the mine operation without a shadow of a doubt. All the process data considering enrichment, geological exploration and analysis, work planning, equipment data and user identification based on their personal ID-tags are few examples for the type of data it stores. Most important types of data stored by KaTTi can be found in Appendix 1.

The principle of the upcoming tracking system is rather straight-forward: when the supplier sends a shipment of explosives to the mine, they add an XML-file (Extensible Markup Language) to their Secure FTP web server. This file includes every single unique identifier of the shipment in text form and these codes are also found in the shipment itself as QR-code stickers. The mine has a service running on their own web server that is set to access suppliers in pre-determined intervals and downloads any files it finds. At the end of this process files are also deleted from supplier server so that the service does not fetch them again during the next cycle. After the shipment has arrived to the mine, warehouse personnel read any code from the shipment in order to approve the whole shipment, which is when it gets added to both ERP and KaTTi, followed by moving all the products to the locked storage area. Designated miners and warehouse personnel are the only employees who have authorized access to the explosives storage: the storage area has electrical locks that are operated with RFID-based identification cards. Once a miner needs to perform blasting, he accesses the storage area with his card, reads the individual codes from the products he wants to use and so they are marked as consumed in KaTTi-system and then company's ERP-system. If explosives are left over, miner returns them back to the storage area and warehouse personnel reads the codes again, thus moving them back to the storage in the eyes of both computer systems. Figure 10 below illustrates this whole process.

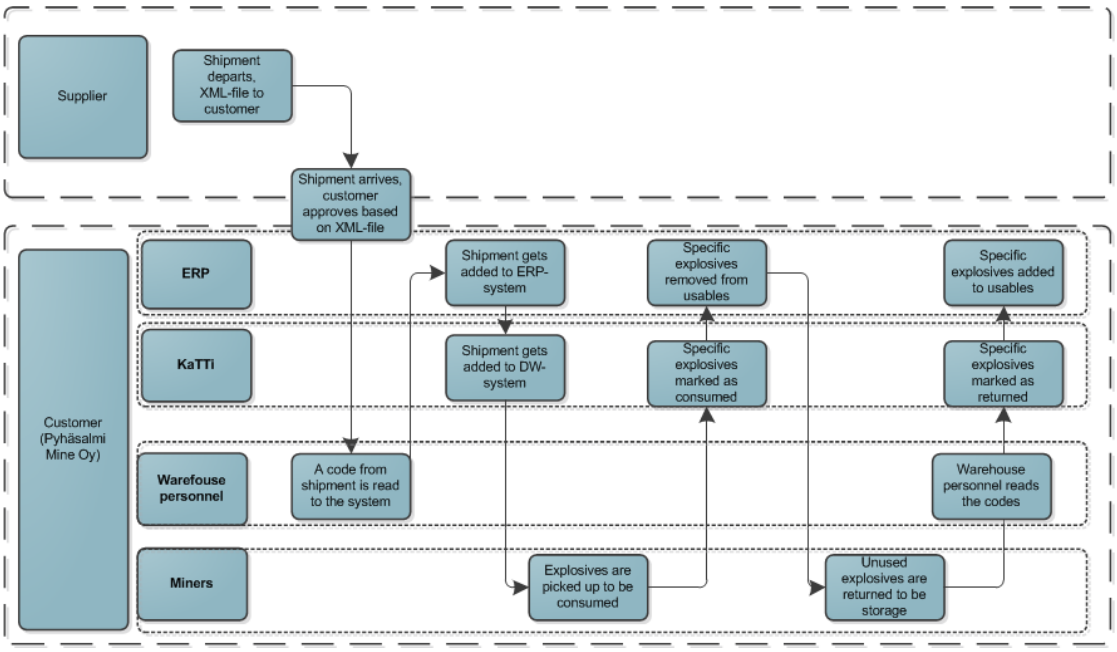


Figure 10. Explosive material delivery and consumption

The tracking is performed very closely because on every step inside the mines gates, the user who interacts with them in any way is identified and logged. The identification is based on the personal identification cards that are required in order for an employee to log in to the system and to perform any action related to explosives. This same card allows authorized personnel access to the explosives storage which will also improve tracking: letting someone unauthorized in and taking the blame for a possible theft is not a very appealing situation to anyone. In case some unknown civil explosives are found or confiscated by police, QR-codes can be read and police can request logs of codes from any companies that they could have originated from. This process is further illustrated in Appendix 2.

The process of an employee accessing the system to consume or return explosives is a good example on how complex interfaces this new system has and how small changes can have a vast effect on the whole environment. This process begins from user identification: identification card is read and queried from KaTTi, since KaTTi includes a list of all the identification cards that ties a specific card to a corresponding person. If a valid card was read (either miners or warehouse employee's), process continues to the next step: product ID checking. Since QR-codes only include the

product codes from the supplier, KaTTi has a reference table that maps specific supplier product code with corresponding in-house product code in ERP-system. For that reason, this phase is marked under the ERP in Figure 11; even though KaTTi helps with the process, majority happens in ERP. After the product code validity is deemed good, the amount check phase follows. In a hypothetical scenario, a miner would want to consume an explosive that has not been returned through the proper leftover process but rather just carried back to the shelf. He would not be able to go through with it, not before the specific explosive has gone through the regular return process. For this reason, a thought about some kind of locked container inside the storage area was brainstormed in the IT department, since it would explicitly eradicate any possibility for such situation. If amounts are valid, storage location phase is initiated. The person inputs the internal name for the explosives storage and system approves it if it is a valid response. After valid response is received, sub-process for either consuming or returning products is gone through. The main phases can again be seen in Figure 10. After either sub-process is done, the program stops. If any of the previously mentioned phases fails, the program does not allow the user to continue to the next phase and only option available is to close the program. The whole process is illustrated in Figure 11 below.

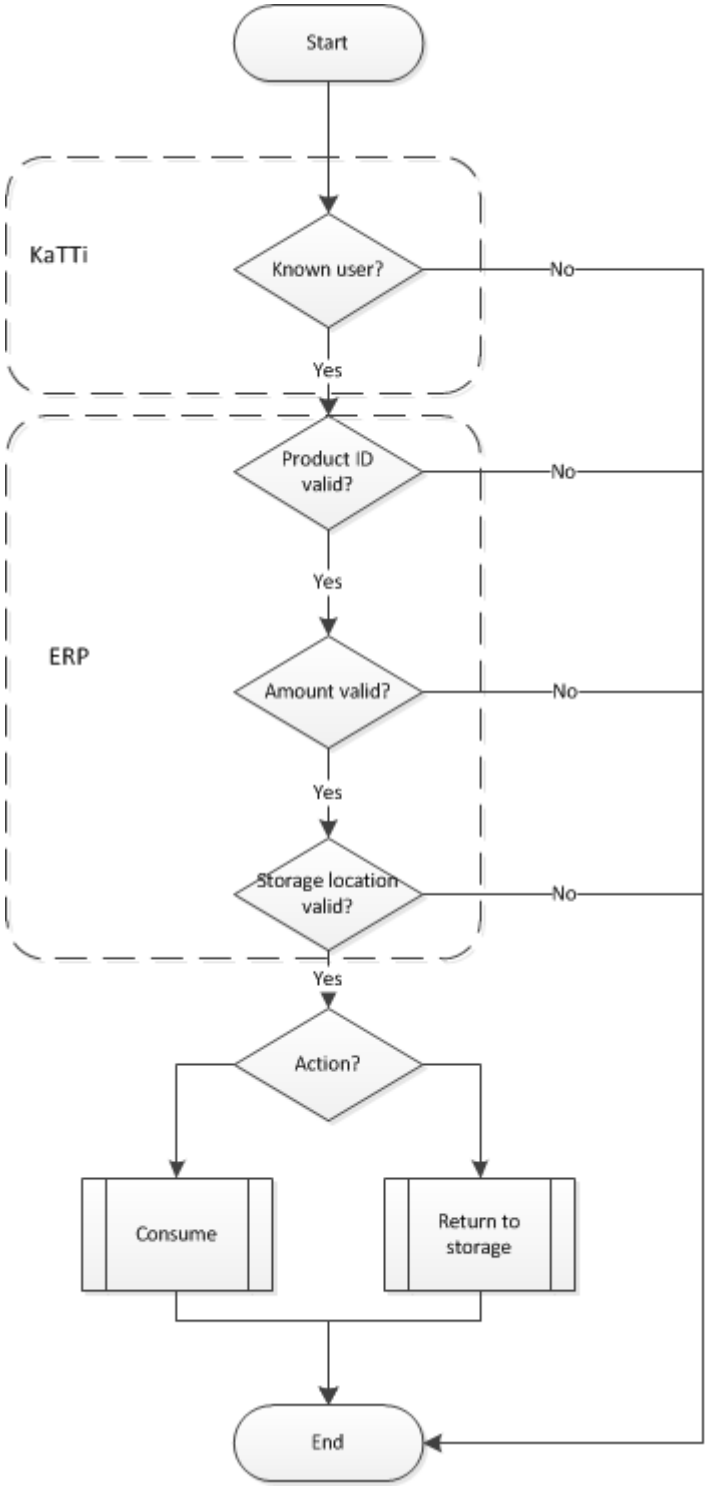


Figure 11. Processing explosives at Pyhäsalmi Mine Oy

7 Discussion

The objective of this thesis was to create a document that would make it easier for the IT department of the company to get the approval of the company management to start building Enterprise architecture for Pyhäsalmi Mine Oy. According to the advisor at the mine, this goal was achieved. The major part of the thesis was theoretical; however, in the end, the theory was brought into the practical application as well. The sheer amount of theoretical material that the selected frameworks included felt slightly overwhelming at times and deeper understanding of all three could not be achieved within the constraints of a thesis project, mostly due to the thesis writer having prior experience only of one. However, since the goal given by the company was met, this aspect is acceptable. Since it was not within the scope of this thesis project, the company obviously still needs to map out the whole enrichment process to be used in Enterprise architecture before it can even begin to be deemed successful. Still, the thesis provides a good basis to continue the started work and gives good advice on methods to be used.

References

Local Registers Offices. List of Local Registers Offices units in Finland. Accessed on 28th April 2016. Retrieved from http://www.maistraatti.fi/fi/kaikki-maistraatit/

Enterprise architecture of Local Registers Offices -workgroup (Maistraattien kokonaisarkkitehtuuri-työryhmä). 19th November 2015. High-level description of how Local Registers Offices will implement Enterprise architecture from 2012 until 2015 v4.0 (Maistraattien kokonaisarkkitehtuuri 2012-2015 v4.0). Accessed on 28th April 2016. Retrieved from https://www.avoindata.fi/data/fi/dataset/mti-ka-2012-2015

The Exploration, Mining, Metals and Minerals (EMMM) Forum. Version 01.12 of the Exploration and Mining Business Capability Map that presents important business capabilities that should be considered when bulding EA for mining industry. Retrieved from https://www2.opengroup.org/ogsys/catalog/C143

Farenden, P. 2012. ITIL for Dummies. Guide about ITIL framework and how to use it [E-book, available in Books24x7]. John Wiley & Sons.

Finnish Standards Association SFS. SFS-ISO/IEC 29881. 21 January 2013. Standard describing FiSMA 1.1 functional size measurement method for a software development and project planning. Accessed on 17th March 2016. Retrieved from SFS Online-service.

First Quantum Minerals Ltd. Company history. Accessed on 22 April 2016. Retrieved from http://www.first-quantum.com/Our-Company/history/default.aspx

Hausman, K. Cook, S. 2011. IT Architecture For Dummies. Guide about planning and implementing IT Architecture [E-book, available in Books24x7]. John Wiley & Sons.

Huovila, M. 2015. 9th March 2015. Examples on how public health care and social services top administrative body in Finland (THL) applies JHS 179 recommendations to better serve health care specific needs. Accessed on 29th April 2016. Retrieved from http://www.sytyke.org/wordpress/wp-content/uploads/2014/01/SOTE-KA-esimerkkej%C3%A4-JHS179-soveltaminen-MikkoHuovila_STM.pdf

IMS Oy. Introduction of Architect software. Accessed on 24th March 2016. Retrieved from http://www.ims.fi/architect-software?language=en

Information Systems Audit and Control Association (ISACA). 2012. COBIT 5 — A Business Framework for the Governance and Management of Enterprise IT [Publication of the ISACA organization, available in Books24x7]. ISACA.

ISACA. Guidance and certifications provided to IT-professionals by ISACA organization. Accessed on 21 April 2016. Retrieved from http://www.isaca.org/About-ISACA/What-We-Offer-Whom-We-Serve/Pages/default.aspx

ISACA. History of ISACA organization. Accessed on 31 March 2016. Retrieved from http://www.isaca.org/About-ISACA/History/Pages/default.aspx

JUHTA (Finnish Public Administration Advisory Board), 2011. 05 October 2012. Documentation of the recommendation JHS 179 that focuses on how the Finnish Public Administration should build Enterprise architecture. Accessed on 25 April 2016. Retrieved from http://docs.jhs-suositukset.fi/jhs-suositukset/JHS179/JHS179.html

Sparx Systems. Introduction of Enterprise Architect software. Accessed on 24th March 2016. Retrieved from http://www.sparxsystems.com.au/products/ea/index.html

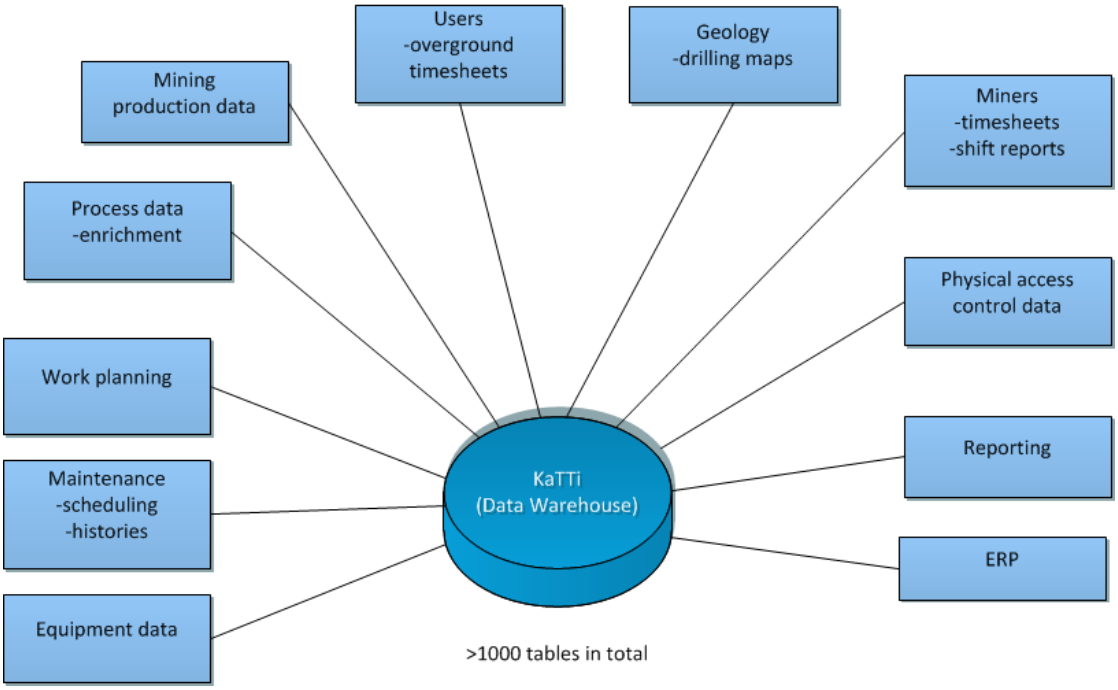
The Open Group Architecture Forum, TOGAF Version 9.1. 2011. Terms of use for TOGAF 9.1 Annual Commercial License. Accessed on 15 February 2016. Retrieved from https://www.opengroup.org/architecture/togaf91/comm_index.htm

The Open Group Standard TOGAF Version 9.1. 2011. The Open Group Architecture Framework and how to create Enterprise architecture following TOGAF guidelines. Accessed on 10 February 2016. Retrieved from http://pubs.opengroup.org/architecture/togaf9-doc/arch/

Vuorimiesyhdistys. 2003. Description about the operation of the Pyhäsalmi Mine in 2003. Retrieved from http://www.Vuorimiesyhdistys.fi/sites/default/files/attachments/Inmet-Pyhasalmi.pdf

Appendixes

Appendix 1. Major data types stored by KaTTi data warehouse system



Appendix 2. Process of tracking unknown explosives origin