

MESG
MESTRADO EM ENGENHARIA
DE SERVIÇOS E GESTÃO

**From Legacy System into SaaS-based System:
A Public Transport Document Generator Case Study**

Muhammad Ghufron Mahfudhi

Master Dissertation

Supervisor in FEUP: Prof. Maria Teresa Galvão Dias

Supervisor in OPT, SA: Luís Filipe Ferreira



Faculdade de Engenharia da Universidade do Porto

2015-10-08

To this small city and all of the families that I met and cherish in my life.

Abstract

The evolution of internet has changed the way businesses deliver their services to public. Internet creates new business opportunities and service offers with the support of technology, such as virtualization, cloud computing, and Service Oriented Architecture (SOA). Nowadays, most of businesses are performed through internet using the software as a service (SaaS) business model, one of the delivery model through cloud computing, by charging the services through subscription and pay-as-you-go.

The internet trend also draws an interest to OPT who wants to reach international market using the opportunity of a new business model by modernizing the Infopub system into a service-oriented SaaS model of public transport document generator. However, the issues arise since the current research is still lack of discussion of modernization into SaaS using a full service engineering concept.

Therefore, this research proposed a Service-Oriented Modernization Framework (SOMF) to provide a general support for the modernization process of legacy system into a service-oriented SaaS system. The framework was developed by integrating several disciplinary approaches and consist of the maturity model description for evolution roadmap identification, the service-oriented SaaS reference architecture, and the User-Centered Service-Oriented Software Reengineering (UCSOSR) methodology to guide the modernization process through several methods and techniques. The framework then was applied to the OPT's case study to develop the public transport document generator as a service. The modernization process produced a design specification of the system in service oriented architecture and development plan for implementation and deployment.

The combination of service design and SOA towards the alignment in business and technical aspects of service engineering which promotes the business agility from customer experience and information system. Service orientation, supported by virtualization, cloud computing, and SOA, hold an important role to achieve a competitive advantage by creating an agile business model, thus raising more business opportunities.

Resumo

A evolução da internet mudou a forma como as empresas entregam os seus serviços ao público. A Internet criou novas oportunidades de negócios e oferta de serviços com o apoio da tecnologia, tais como virtualizações, computação em nuvem e Arquitetura Orientada a Serviços (SOA). A maioria dos negócios das empresas hoje em dia são realizados através da Internet, utilizando um modelo de negócio de software como serviço (SaaS), usando um modelo de entrega através de computação em nuvem, pela cobrança dos serviços por meio de subscrição e pagamento conforme utilização.

A tendência da internet também atrai os interesses de OPT que querem alcançar mercados internacional usando a oportunidade de um novo modelo de negócios através da modernização do sistema Infopub em um modelo SaaS orientada para os serviços de transporte público gerador de documentos. No entanto, os problemas surgem uma vez que a pesquisa atual ainda apresenta falta de discussão da modernização em SaaS recorrendo ao conceito de engenharia de serviço completo.

Portanto, esta pesquisa propôs uma Framework de Modernização Orientada a Serviços (SOMF) para fornecer um apoio geral para o processo de modernização do sistema legado em um sistema SaaS orientada a serviços. O quadro foi desenvolvido integrando várias abordagens disciplinares e consistem na descrição do modelo de maturidade para identificação, roteiro de evolução, arquitetura de referência SaaS orientando os serviços, e metodologia de reengenharia de software orientada a serviços centrado no usuário (UCSOSR) para orientar o processo de modernização através de vários métodos e técnicas. A framework foi então aplicada no estudo de caso da OPT para desenvolver o transporte público gerador de documentos como um serviço. O processo de modernização produz uma especificação de conceção do sistema em arquitetura orientada a serviços e plano de desenvolvimento para implementação e implantação.

A combinação do design de serviço e arquitetura orientada a serviços (SOA) para o alinhamento dos aspetos técnicos e de negócio através de engenharia de serviços promovem a agilidade dos negócios a partir da experiência do cliente e sistema de informação. A orientação a serviços, apoiada pela virtualização, computação em nuvem e SOA, mantém um papel importante para alcançar uma vantagem competitiva através da criação de um modelo de negócios ágil, aumentado mais oportunidades de negócios.

From Legacy System into SaaS-based System: Public Transport Document Generator Case Study

Acknowledgments

I would like to express my special gratefulness for all of the people who support and help me in this research development and give the best two experience in my life in this beloved city, thus gaining a lot of oPORTOnities.

Above all, I am particularly grateful for my supervisor, Professor Teresa Galvão, for providing the opportunity to get the internship and research project in OPT, and for patiently guiding me and provide inputs, assistances, inspirations throughout the research project.

Secondly, I would like to thank my supervisor in OPT, Mr. Luís Filipe Ferreira, for providing a lot of inputs for the research project and brainstorming ideas of the SaaS model for Infopub.t I also would like to thank Mr. Fernando Vieira for giving me the opportunity to have an internship in OPT, although not much that I can do for the company.

I would like to thank Professors Lia Patrício and João Falcão e Cunha, as director of MESH FEUP, for providing me the opportunity to study in MESH and supporting me in my study in MESH. Assistance provided by Ms. Isabel Ribeiro for all of the administrative support during the research was also greatly appreciated.

I also would thank all of the friends and families that I made in Porto and made me ‘survive’ in this city, especially during the research period. I would like to thank my Portuguese family that I made from MESH program and Orfeão Universitário do Porto that welcomed me and taught me to be a half Portuguese. I also would like to thank my Indonesian family that I met in Indonesian Student Association in Portugal that make me feel so close from home, especially with the Indonesian foods. I could not forget all of the memories of the families that I created in Porto.

Finally, I express my deep gratitude for my parents and siblings in Indonesia which always support me and give me motivation for this research. Thank you very much for your love, support, and pray. Despite the long distance between us, our hearts would still be connected as one.

List of Contents

1	Introduction	1
1.1	Optimização e Planeamento de Transportes, SA	2
1.2	Research Project.....	3
1.3	Research Objectives and Questions	3
1.4	Research Methodology	4
1.5	Outline.....	5
2	Literature Review	7
2.1	Cloud Computing	7
2.2	Software as a Service	7
2.3	Service Engineering	10
2.3.1	Service in Business Front	10
2.3.2	Service in Technological Front.....	11
2.3.3	Service Concept Application in SaaS.....	16
2.4	Legacy and Modernization	17
2.5	Summary.....	18
3	Service-Oriented Modernization Framework	19
3.1	SaaS and SOA Maturity Model	20
3.2	Service-oriented SaaS Reference Architecture.....	22
3.2.1	SaaS Platform Side	23
3.2.2	Service Provider Side	24
3.2.3	Service Consumer Side	24
3.3	User-Centered Service-Oriented Software Reengineering.....	24
3.3.1	Business Modeling and Transformation.....	26
3.3.2	Project Initiation and Planning	28
3.3.3	Stakeholder Requirements Specification	28
3.3.4	Service-Oriented Reverse Software Engineering	29
3.3.5	Service-Oriented Forward Software Engineering.....	29
3.3.6	SaaS Application Development	34
3.4	Service Development Tool	34
3.5	Demonstration and Evaluation Plan	35
3.6	Summary.....	36
4	SaaS Services Development	37
4.1	Public Transport Document Generator.....	37
4.2	Business Modeling and Transformation	38
4.2.1	Business Architecture and Model	38
4.2.2	Situational Analysis.....	40
4.2.3	Strategy Plan	42
4.3	Project Initiation and Planning.....	43
4.4	Stakeholder Requirements.....	44
4.5	Service-Oriented Reverse Software Engineering.....	44
4.6	Service-Oriented Forward Software Engineering.....	46
4.6.1	Service Identification.....	46
4.6.2	Service Specification.....	48

4.6.3	Service Realization	49
4.6.4	Service Implementation and Deployment	50
5	SaaS Application Development.....	52
5.1	Prototype Development.....	52
5.2	Cloud Platform Evaluation.....	53
5.3	SaaS Implementation	54
5.4	SaaS Operation.....	55
5.5	Summary.....	55
6	Evaluation and Discussion	56
7	Conclusions and Future Works	58
	References	60
APPENDIX A:	Business Motivation Model and Business Model of OPT and Infopub.....	63
APPENDIX B:	Maturity Model Analysis of OPT and Infopub	65
APPENDIX C:	Detailed Situational Analysis	73
APPENDIX D:	Modernization Requirement Specifications	75
APPENDIX E:	The Artifacts of Legacy System.....	77
APPENDIX F:	User Interfaces of Legacy Infopub System	79
APPENDIX G:	Service Development Artifacts	81
APPENDIX H:	Prototype of Public Transport Document Generator	85
APPENDIX I:	Comparison between Microsoft Azure and Google Cloud Platform.....	86
APPENDIX J:	Modernization Framework Evaluation	88

Index of Figures

Figure 1. Design Science Research Methodology (DSRM) Process Model (Peffer et al. 2008)4

Figure 2. SaaS Maturity Level (Kang et al. 2010)..... 8

Figure 3. Key Success Factors of SaaS for Continuous Improvement (Hagins 2008).....9

Figure 4. SaaS Development Life Cycle (Kommalapati and Zack 2011)9

Figure 5. Components of Multilevel Service Design (Patricio et al. 2011) 10

Figure 6. The Open Group Service Integration Maturity Model (The Open Group 2011a) 13

Figure 7. SOA Reference Architecture (The Open Group 2011c) 14

Figure 8. Web Service Building Blocks (Hauch 2005) 15

Figure 9. SOCCI Architecture (The Open Group 2011b) 16

Figure 10. Service-Oriented SaaS Reference Architecture22

Figure 11. User-Centered Service-Oriented Software Reengineering Methodology Process..25

Figure 12. Business Motivation Model (OMG 2015)26

Figure 13. Evaluation Criteria for Modernization Process and SaaS Quality Model (Khadka et al. 2012, Pang Xiong and Li 2013)..... 36

Figure 14. Infopub System Architecture44

Figure 15. Customer Journey of Public Transport Document Generator.....47

Figure 16. The Candidate Services of Public Transport Document Generator47

Figure 17. Exposed Service Interfaces of Public Transport Document Generator48

Figure 18. Message Data Structure of Generation Process48

Figure 19. Service Architecture of Public Transport Document Generator49

Figure 20. The Completed Information Production Structure in Participant Diagram49

Figure 21. Generated Web Services and WSDL example of Document Generator Interface .50

Figure 22. Prototype of Public Transport Document Generator52

Figure A-1. Business Model Canvas of Infopub Service from OPT.....64

Figure B-1. Evolution Roadmap of Maturity Model for OPT and Infopub 65

Figure C-1. Competitive Force Analysis of Infopub in SaaS Model 74

Figure E-1. Database Schema of Infopub (Meneses 2009).....77

Figure E-2. Use Case Diagram of Infopub78

Figure F-1. Data Manager of InfoPub79

Figure F-2. BusMap Generator of InfoPub79

Figure F-3. BusSched Generator of InfoPub80

Figure F-4. MultiDoc Generator of InfoPub	80
Figure G-1. Goal-Service Model of Infopub in SaaS	81
Figure G-2. Service Blueprint of Infopub in BPMN	81
Figure G-3. Service Development Packages	81
Figure G-4. Service Architecture of Public Transport Document Generator	82
Figure G-5. Sequence Diagram for Service Contract of Information Production Service	82
Figure G-6. The Identified Participants of Public Transport Document Generator	83
Figure G-7. Building Block Layer Architecture of Public Transport Document Generator ...	84
Figure H-1. Prototype of Public Transport Document Generator	85

Index of Tables

Table 1. Characteristics Comparison of Software Architectures (Alkazemi, Baz, and Grami 2012)..... 12

Table 2. CASE Tool Review for Development of Service-Oriented SaaS System 34

Table 3. SWOT Analysis of OPT and Infopub 42

Table 4. List of Sources of Requirements 44

Table A-1. Ends and Means of OPT's Business Motivation Model..... 63

Table B-1. SaaS Maturity Model Analysis of OPT and Infopub 66

Table B-2. OSIMM Analysis of OPT and Infopub 68

Table C-1. PESTEL Analysis of Infopub in SaaS Model 73

Table D-1. Modernization Requirements Specifications 75

Table I-1. Comparison between Microsoft Azure and Google Cloud Platform..... 86

Table J-1. Modernization Framework Evaluation..... 88

1 Introduction

Technology has evolved rapidly these days to support the businesses in delivering the services to customers. The internet of things, which connects products and services to the internet, has changed the way businesses deliver their services, even for traditional industries. In 13 countries with several levels of development, internet contributed on an average of 3.4% to GDP, and accounted for 10% of GDP growth in the past 15 years. Moreover, 75% of internet impact arose from small and medium business in traditional industries which were not the sole internet players, and the internet could increase their productivity by 10% (Rausas et al. 2011). Internet is a critical element to grow as it affects the economic condition by creating economic modernization, creating business opportunity and service offer.

InfoPub, a desktop-based public transport document generator from *Optimização e Planeamento de Transportes, SA (OPT)* which allows publics to access the transport information, also uses internet as support in the system infrastructure as a service application provider. The clients, which consist of transport companies, can use InfoPub to provide the transport information to publics in the form of document. However, there was a problem regarding the flexibility of the service delivery, caused by the limitation of desktop environment, as they had to provide a configured service application separately for each client's needs and install it in the clients' sites. The application was not interoperable enough that it still needed developers' intervention for configuration and redevelopment, thus increasing cost for OPT and inefficiency for clients. Therefore, an interoperable and flexible system would be needed to give competitive advantage for OPT and more satisfaction to clients.

Meneses (2009) proposed a solution to develop InfoPub in a web-based environment. A functional prototype was developed as a result of the modernization process using several web-based technology. The solution allowed clients to access the information and generation service in a fast and effective way. It gave the benefits of cost reduction for both clients and OPT because software maintenance and user assistance process could be done easier. However, the proposal only focused on the web-based software development and was limited in the term of software delivery as a web application. In the business aspect, this proposal could be enriched by developing a complete service concept, not just delivering a software through web, but also delivering complete and integrated services to satisfy the customers' needs.

With the growth of internet, several supporting technology emerges, promoting cloud computing to support service delivery. Cloud computing promotes the use of cloud service and service oriented architecture (SOA) which promote cost reduction and agility, allowing business to implement service oriented system in the cloud, giving easier accessibility and flexibility to customers (Khanjani, Rahman, and Ghani 2014). One type of the common implementation of cloud services is software as a service (SaaS) which allows access to software features and functionality in the form of web services, thus giving access of services for business process support without installation. Moreover, as SaaS technologies are growing faster, SaaS software revenues are predicted to represent the 25% of overall software market in the next five years, thus giving a promising results (Kaszniak 2014). This opportunity will give more added value for business companies to compete, creating a service innovation to give better experience to customers, and achieve business goals.

The possibility of SaaS application for the public transport will give a great benefit. OPT can provide one integrated platform which can be subscribed and customized based on what clients need. The platform will be centralized so that the management process can be done easier with less cost. Clients will also feel a better experience with the system of use based on what they pay and direct usage through web browser, without any need for installation. This implementation will give easier access for getting customer acquisition and retention, thus increasing the revenues and market coverage.

Developing an SaaS application needs an investigation to find the best suited SaaS model to the provision of every stakeholders. Most studies found on literatures still focused only on each particular aspect of SaaS development, such as reverse engineering, software engineering, service oriented analysis and design, and user centered design. However, SaaS is an integrated model which is connected to all of those aspects. Besides, SaaS development also depends on the company's particular business model. Therefore, this arose an issue on how to integrate all of the aspects within SaaS to develop a comprehensive SaaS development model and particularly in the case of public transport document generator.

1.1 Optimização e Planeamento de Transportes, SA

Optimização e Planeamento de Transportes, SA (OPT) is the first company in Portugal to develop intelligent and automatic solutions for decision support in transport planning. Carrying the vision towards excellent service in innovative and optimized systems for transport planning, management, and public information, OPT has started its activity in 1997 as a result of a research and development project involving 2 universities and 5 of the largest Portuguese bus transport companies. Its objectives is to keep the business level and quality of its products in a strong growth, to maintain the leadership in national level and aim a high level of market diversification. It invests in innovation, research, and technological development, and actively collaborates with partners by joined COTEC Innovation Network and Microsoft Partner, receiving a certification in ISO 90001 and NP4457 IDI (OPT 2015).

OPT has a core activity in operational management of urban transportation. It is responsible for GIST, a decision support system for optimized operational planning in transport companies. It also focuses in products and services for improving passenger information by developing system that automate the production of the stops information and producing information maps for different areas for different cities, such as SpiderMaps, InfoPub, SMSBus, Infoboard, and Move-me. It also carries out consultancy work in transport operational planning by reengineering the operational planning processes, changing the studies of personnel policies, etc. It has expertise in complex operational problems: transport planning, systems integration, intelligent transportation systems, passenger information, and demand responsive transport.

OPT is involved in a European project, called CIVITAS ELAN, promoted by a consortium which consisted of several cities. The mission is to 'mobilize' the citizens by developing clean mobility support solutions for vital cities, ensuring health and access for all of them. In this project, OPT integrated real time information of several Portuguese transport operators and keep the consistency of several geo-referenced information. The system assured an optimized information to support users' needs and specifications, adapting the usage of planning algorithms to a dynamic situation in real time. One of the product, the Move-me system, was the winner for Technical Innovation Award on September 2012.

To leverage its market, OPT focused by adapting their solutions to smaller companies and growing into bigger markets and promoting internalization strategy. The future prospects of technology development in quantitative methods and information and telecommunication technology enable OPT to offer creative solutions, creating different level of services for publics. The quality is also always being improved in all of its solutions as support for the needs in demanding and dynamic markets, using adaptation and promoting both innovation and national technological research.

1.2 Research Project

The project was mainly to perform a modernization of a legacy system into web-based environment delivered in SaaS model using service concept and SOA. It was firstly proposed as a mean to explore the integration of service engineering in both business front and technological front by combining the service design concept and service oriented architecture. Then it was expanded for the domain of OPT as the company also considered the idea to develop InfoPub on web-based environment using SaaS delivery model.

InfoPub is an application service provider platform to generate a static or dynamic information about public transport. It produces a graphic documentation with advanced design which adapts to the data contents by generator application such as BusMap and BusSched. InfoPub is developed in desktop-based environment using client-server architecture. As an on-premise application, it is required to be installed on the client side and can be different for each of client, which resulting difficulties and high cost in maintenance and configuration.

The need to reduce the cost and to offer services that can be scaled up/down based on varying demand motivates the modernization process. By transforming InfoPub platform into cloud, it can provide better support of document generation of the stops for public transport companies through web. Moreover, it will not be just hosting the application in the cloud server as a web application, but also applying the multi-tenancy solution which can give the benefit of scalability and customization, thus creating a new business model for InfoPub and allowing to reach a bigger market.

The project was performed mainly based on the analysis of the InfoPub platform and requirements defined by OPT. The project was more focused on the architecture reengineering of InfoPub from legacy client-server architecture into service oriented architecture, thus motivating the need to develop the modernization framework and methodology which was suitable for this case.

1.3 Research Objectives and Questions

Based on the motivation stated in the background and project description, this research aimed to investigate the modernization process of a legacy system, particularly in case of public transport document generator, into SaaS-based system using service concept through multidisciplinary approach. Within this scope, this research gives contribution in two areas: academic and practical. The objective in the academic area is to design the modernization framework, including the integrated reference architecture, development methodology, and the artifacts in each process. Whereas in the practical area, the objective is to design and develop the architecture model and prototype as requirement and design specification for modernization of InfoPub into Public Transport Document Generator as a Service.

From the identified objectives, several research questions which support the contribution in academic and practical area were defined:

1. What is the correlation of characteristic between service design and SOA in SaaS modernization?
2. How to find and develop the key aspects that should be considered to design a solution of suitable and comprehensive SaaS modernization framework with multidisciplinary approach, especially within the case of public transport document generator?
3. How can the framework be developed and applied to an industrial case, with concern of the decision of cloud services outsource (public cloud) or in-house built (private cloud)?
4. What is the impact of the application of service concept within modernization process for the business and technological aspects and changes of an organization?

1.4 Research Methodology

This research used the methodology of Design Science Research Methodology process model by Peffers et al. (2008) as shown in Figure 1. The entry point came from demonstration, as OPT aimed to develop the SaaS-based system of InfoPub. As mentioned before, the initial objective was combining service design in software engineering using SaaS and SOA, thus resulting a final research objective to design and evaluate SaaS development model and architecture of public transport document generator using multidisciplinary approach.

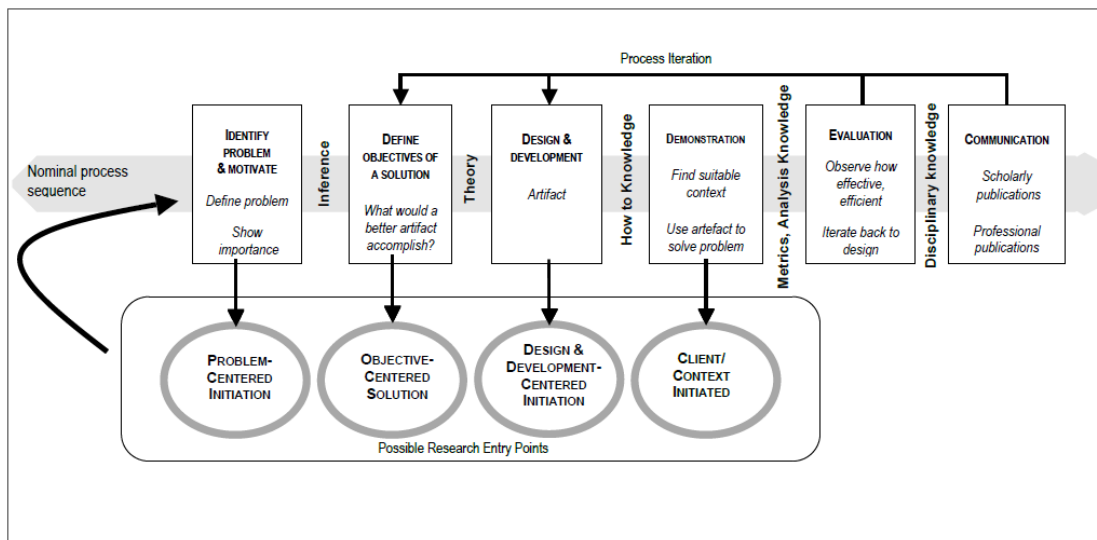


Figure 1. Design Science Research Methodology (DSRM) Process Model (Peffers et al. 2008)

Using demonstration as entry point, the detailed processes consist of:

1. Identify problem and motivate: to design a comprehensive SaaS model, design and develop it for public transport document generator, and analyze the impact.
2. Define objective of a solution: to demonstrate the design and development process of SaaS application development with multidisciplinary approaches in an industry setting.

3. Design and development: to integrate several development aspects of SaaS to build a comprehensive model and particularly for public transport document generator using multidiscipline approach:
 - a. Cloud computing, as SaaS is part of cloud computing which software is delivered as a service.
 - b. Service engineering, as the context is to develop service-orientation both in business front and technical front.
 - c. Software reengineering, as SaaS is still an application software which is delivered through web browser and the context is mainly about modernization from legacy system into service-oriented SaaS system.
 - d. User centered design, as service is close to customer experience and SaaS model is close to usability and user needs, by eliciting the requirements from stakeholders.
4. Demonstration: to demonstrate the comprehensive modernization framework into public transport document generator, from stakeholders' needs into prototype.
5. Evaluation: to evaluate the framework and platform development using evaluation criteria based on the model, analysis of the changing business model, and stakeholders' acceptance.
6. Communication: to publish the research in a thesis document and possible journals or conferences.

1.5 Outline

In this report, the discussion is presented in the following topics:

1. Chapter 1 – Introduction, which describes the background of the thesis research project followed by the description of the company (OPT, SA), the research project description, the research objectives and questions, the research methodology and the report outline.
2. Chapter 2 – Literature Review, which describes the explanation of several existing theories and latest research which were used as foundation for this project, including legacy and modernization, cloud computing, SaaS, service engineering, software reengineering, and user-centered design.
3. Chapter 3 – Service-Oriented Modernization Framework, which describes design process and discussion of the framework development from multidisciplinary approach. This chapter shows the correlation of each discipline to support the modernization process in resulting the integrated reference architecture, the user-centered service-oriented software reengineering methodology, the evaluation model, and the demonstration plan.
4. Chapter 4 – SaaS Services Development, which describes the first part of the modernization process which is more focused to identify the needed services from several approaches and to develop into web service in cloud. It starts from the system scope description, business modeling and transformation as envisioning foundation, project initiation and planning, stakeholder requirements, the service-oriented reverse

software engineering to analyze the legacy system, and the service-oriented forward software engineering to develop the services.

5. Chapter 5 – SaaS Application Development, which describes the second part of the modernization process which more focused on the SaaS application as a whole system which was built based on the identified services. It starts from the prototype development, cloud platform evaluation to analyze the cloud implementation strategy, and followed with the SaaS implementation and operation description.
6. Chapter 6 – Evaluation and Discussion, which describes the evaluation from the thesis research development and discussion for several found important points throughout the research development.
7. Chapter 7 – Conclusions and Future Works, which describes the overall main points of the result of the research and recommendation for possible future research which comes from this research for improvement purpose both in this research scope and general academic and practical purpose.

2 Literature Review

This chapter reviews the basic theory of several research literatures based on in the disciplinary areas related to cloud computing, service engineering, and modernization. The discussion is started with cloud computing and Software as a Service (SaaS), and followed with service engineering, within business aspect, technological aspect, and their application for SaaS. Then, it focus on legacy and modernization, specifically from legacy into service system, as the main object in this research. Finally, the correlation between the concepts found in current literature is reviewed to find the research novelty as foundation of the next steps of the research.

2.1 Cloud Computing

Cloud computing is a computing paradigm which connects information through cloud, which is a scalable network of nodes (Letaifa et al. 2010). It provides dynamic resources, virtualization, and high availability through service delivery model based on internet. Using cloud computing, resources could be shared through all stakeholders using service offers which provided by the virtualization. The uniqueness of cloud services is that it provides more flexible service offers with ubiquitous network access, location independent resource pooling, self-service and instant-used, and pay-as-you-go elasticity.

There are several types of cloud developed in the market. Cloud computing use the concept of Everything as a Service (EaaS), providing component functions in the cloud service, including infrastructure, platform, database, and software. The architecture consists of six layers: cloud clients, applications, services, platform, storage, and infrastructure, thus promoting the development of the whole integrated service delivery system. The service delivery itself comprises of three main categories: public cloud which services are delivered through a third party service provider, private cloud which services are managed and provided within organization, and hybrid cloud which is combination of public and private cloud. Several cloud solution providers are also available in the market such as Eucalyptus, Open Nebula, Nimbus, AbiCloud, Windows Azure, and Google AppEngine (Letaifa et al. 2010, Nodehi 2014).

2.2 Software as a Service

Software as a Service (SaaS) is a part of cloud computing which software application is delivered as a service through internet so that customers could access the service via web browser. SaaS providers manage the remotely application and customers can consume it on a pay-for-use or certain subscription based on usage metrics (Kang et al. 2010). SaaS is often compared with hosted traditional software or on-premise Application Service Providers (ASP), but unlike them, SaaS is a service-oriented framework with high deployment efficiency and supportable platform. The main difference within SaaS is that it uses multi-tenancy approach and on-demand delivery model which provide economies of scale to offer the software-cost effectively based on the available capabilities. NIST (National Institute of Standard and Terminology) defines that SaaS's main characteristics are identified that it is an on-demand self-service with a broad network access, resource pooling based on multi-tenancy model, rapid capabilities elasticity, and service measuring system. In brief, SaaS could provide a more flexible way of business to grow with a bigger opportunity and agility.

SaaS characteristics can be defined from the maturity model, which shows leveled comparison benchmark based on defined criteria, as parameter for SaaS development. Kang et al. (2010) pointed out several SaaS functions from technical and business aspects which define its main characteristics such as multitenant, configuration, scalability, integration, and security, thus contributing to the architecture development. From this characteristics, he developed a maturity model as combination of the ones developed by Microsoft and Forrester which shows the condition of infrastructure level based on the data, system, service, and business architecture. It consists of four levels as shown in Figure 2:

1. Ad hoc level, which means that the organization still uses an ad hoc system with simple Application Service Provider (ASP) business model.
2. Standardization level, which means that the organization develops a standard SaaS application with shared services and configurable single instance/tenant, but still does not support multi-tenancy.
3. Integration level, which means that the organization is focus in actualizing multi-tenant service process with service connection as key factor.
4. Virtualization level, which means that the organization develop an optimized multi-tenant system with service modulation and encapsulation which applicate the concept of cloud computing and Service Oriented Architecture (SOA).

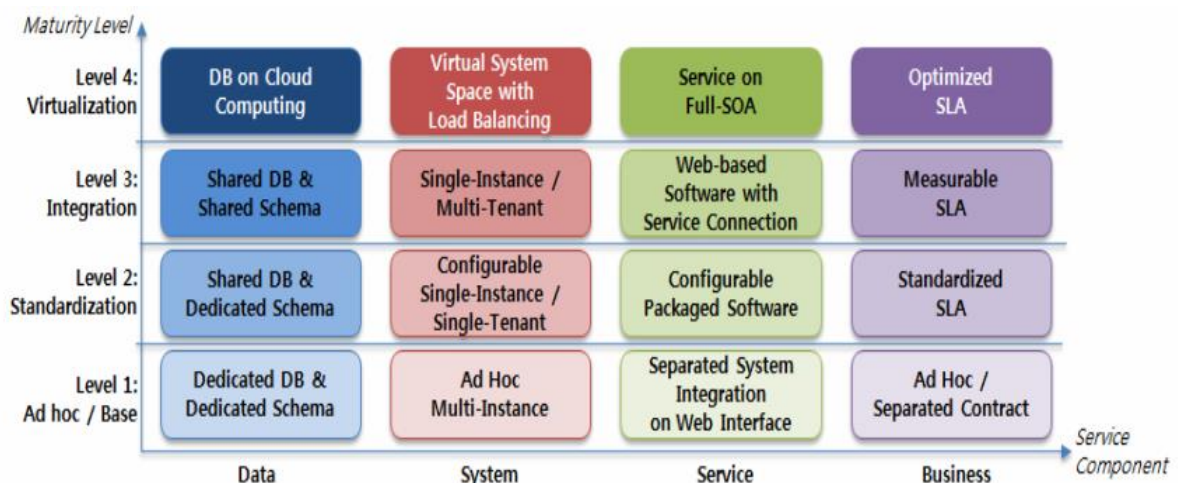


Figure 2. SaaS Maturity Level (Kang et al. 2010)

Based on the defined activities and components, the reference architecture then could be defined to show the main standard components for SaaS system and its connection between the activities on the application provider side and service consumer side (Kang et al. 2010, Pang Xiong and Li 2013). Additionally, Pang Xiong and Li (2013) developed an SaaS quality model to evaluate the SaaS development level based on the defined quality-based metrics. These researches show that SaaS is different than traditional software as it turns out to provide the benefit of mixing several paradigms and technologies.

SaaS development requires concern on business logic, application servers and virtualizations, multitenant data management (Rico et al. 2014), and cloud database management (Candan et al. 2009). The key success factors of an SaaS system relies on the five key process areas shown in Figure 3 with the red ones which indicate the user experiences (Hagins 2008). It shows that user experience hold an important role for an SaaS system as it determines how customers access, interact with, and assess the offered services.

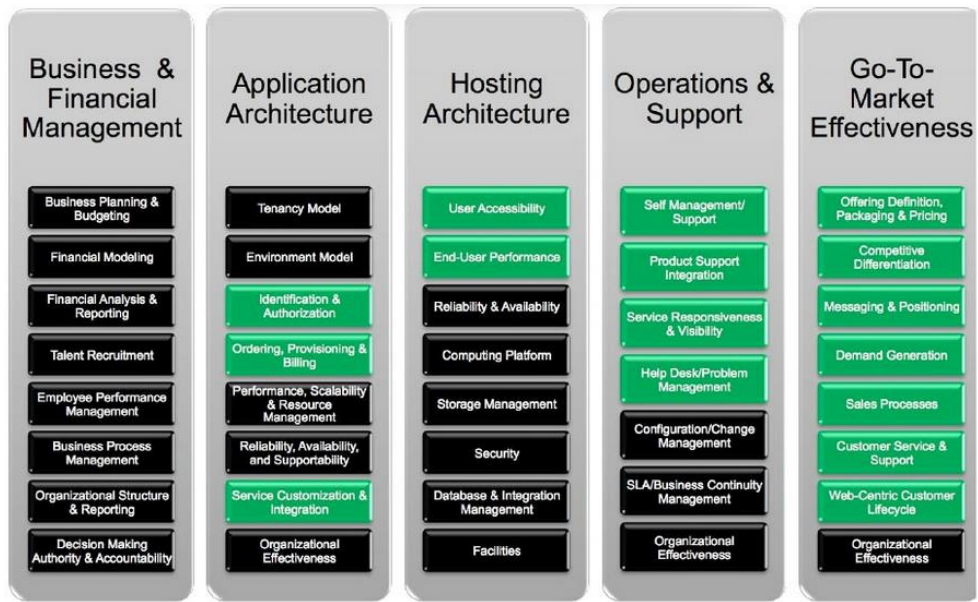


Figure 3. Key Success Factors of SaaS for Continuous Improvement (Hagins 2008)

The development process follows the SaaS development life cycle as shown in Figure 4. These factors are applied into development process of SaaS system using the SaaS development life cycle as shown in Figure 4 (Kommalapati and Zack 2011). It consists six main steps:

1. Envisioning, to define the vision and scope based on the strategy and opportunity.
2. Platform evaluation, to select the cloud provider and specific platform which are feasible to be used.
3. Planning, to organize and manage actions to be performed in development process.
4. Subscribing, to negotiate the service level agreement for needed components and infrastructure within SaaS system.
5. Developing, to execute the service development process from analysis and design, implementation, testing, and deployment.
6. Operations, to operate as basis of SaaS system evaluation to give insights for improvement.



Figure 4. SaaS Development Life Cycle (Kommalapati and Zack 2011)

2.3 Service Engineering

Service can be defined in two kinds of area: business front and technological front. In business front, service refers to business service which is offered by companies to customers. While in technological front, service refers to the software component in Service Oriented Architecture (Chen 2008). Combining both definitions, service engineering could be defined as engineering process of a whole service concept, from the business value proposition into business service model and implemented into service oriented architecture and service technology.

There are several topics circling around service development and delivery, supported by software technology. The whole service development process is a multidisciplinary approach and not just limited in the business aspect or technological aspect. The correlation between several aspects, such as service design, SOA, the role of user-centered design, in supporting SaaS service development process, is discussed in the next sections.

2.3.1 Service in Business Front

Service from business perspective is the economic activities offered by providers to customers through interaction and evidence (Lovelock and Wirtz 2011). Service sector is different with product sector as it has its distinctive characteristics such as intangibility, simultaneity, heterogeneity, perishability, and customer participation in the service process. Service is delivered to customers through a business and service delivery model so that they can perceive the value experience. Business model represents how companies create, deliver, and capture value through the nine building blocks of business model canvas (Osterwalder and Pigneur 2010). This value then would be delivered to customers through a service delivery system which involves people, processes, technology, and physical evidence so that the service offer would meet the customers' expectations.

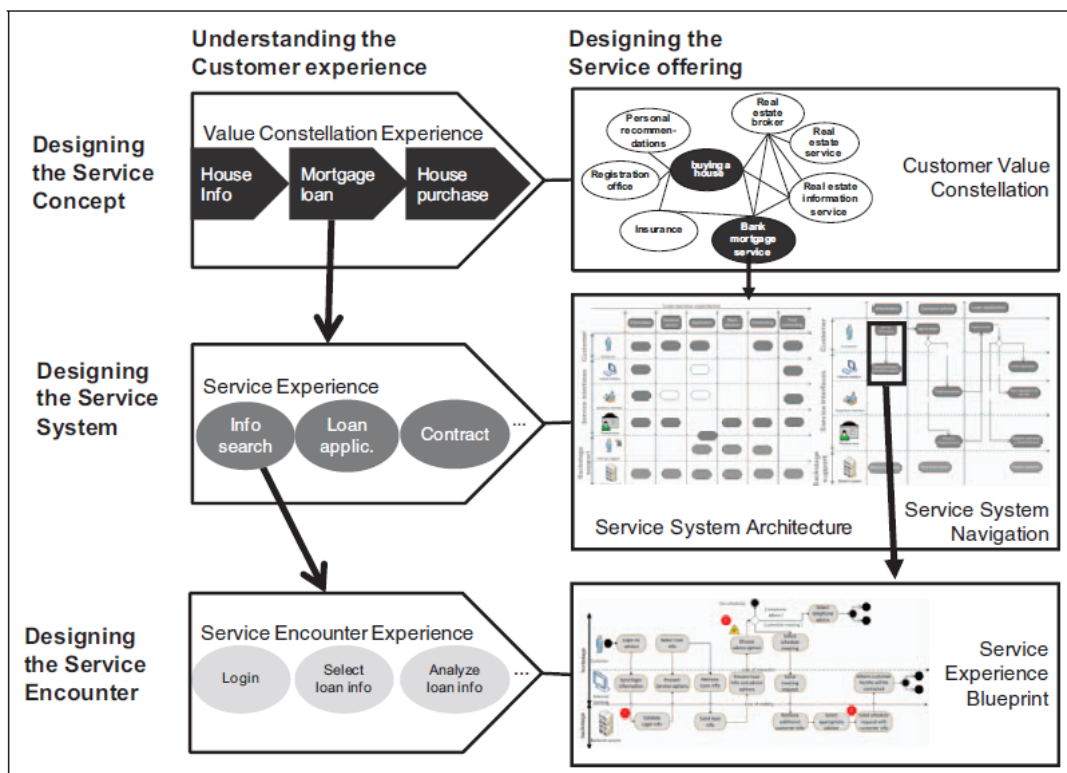


Figure 5. Components of Multilevel Service Design (Patricio et al. 2011)

The study of service and service design is a multidisciplinary approach which collaborate several elements. It involves understanding users and their context, service providers and social practices, and evidence and interaction development. Patricio et al. (2011) develop a comprehensive methodology based on multidisciplinary approach for designing service system, which is called Multilevel Service Design (MSD). This method promotes customer experience as key point for the service design process for innovation. Therefore, service offers could be developed from understanding the customer experience through three hierarchical levels as can be seen in Figure 5. The first level is to design the service concept using customer value constellation based on value constellation experience. The second level is to design the service system into service system architecture and navigation based on service experience. Finally, the service encounter is designed using service experience blueprint based on service encounter experience.

2.3.2 Service in Technological Front

Service from technical perspective is a form of logic encapsulation of a business process which consist of activities in a particular business entity or other logics. Service in this context is the main component which is independent and can be reused overtime in Service Oriented Architecture (SOA). In creating a service, there are several principals which need to be concerned such as reusability, service contract, loose coupling, abstraction, composability, autonomy, statelessness, and discoverability, which make service different that the other software and business components (Erl 2005).

SOA is an architectural pattern which represents an abstraction between business logic and technology which are loosely-coupled, promoting independence, flexibility, and agility in system organization. SOA approach to business focuses on agility and collaborates business entities in the service system, while SOA approach to technology gives support for service implementation and realization through an agile, interoperable, and modular technology base. SOA consists of three main elements: service provider, service consumer, and service registry; and also divided into three types of services: orchestration service, business service, and application service. In a wider perspective, SOA is not just a software design pattern, but also an enterprise IT architecture which promotes the development of enterprise agility and interoperability.

As service could be defined in business and technical term, SOA could a bridge between business and technology, thus creating an enterprise SOA system. SOA approach to business focuses on agility and collaborates business entities in the service system. SOA approach to technology give support for service implementation and realization through an agile, interoperable, and modular technology base.

From the history of SOA, the concept was developed from Client-Server Architecture (CSA), which is a distributed system model where data and processing are distributed through several components. Table 1 describes the comparison between CSA, multi-tiers architecture, and SOA, which shows that SOA is the best in term of flexibility and independency. The main different between SOA and CSA is the coupling system between tiers. CSA is more coupled and very dependent between server and client, without planning for reuse. Companies need to use many different applications by different vendors with different business and technical logic. With SOA, those boundaries are disappear because it is more vendor and platform independent, using reuse concept in mind. Client does not need to know where each service is

provided as the services are no longer bounded to specific implementation, thus promoting plug and play concept. Therefore, it would ease the system integration process.

Table 1. Characteristics Comparison of Software Architectures (Alkazemi, Baz, and Grami 2012)

Characteristics	Client-Server	N-Tier	SOA
Language/Technology Dependency	Dependent	Dependent	Independent
Extendibility	Hard to Extend	Hard to Extend	Fully Extendable
Modifiability	Major changes to source code	Major changes to source code	Support Plug-&Play effectively
Re-use Legacy system	Requires an Adapter	Requires an Adapter	Full support of Re-using legacy systems
Data Exchange	Internally	Internally	Internally/Externally
System Integration	Homogeneous	Homogeneous	Heterogeneous
Maintenance	Affect all parts of the system	Affect most of the system parts	Only a single service is affected
Separation of Concerns	Low	Medium	High

A SOA based solution is analyzed based on the organization's position. The Open Group Service Integration Maturity Model (OSIMM) is used as a roadmap model as it shows the degree of service integration of an organization. OSIMM defines several parameters from business and architectural views of an organization which consist of business, governance and organization, methods, applications, architecture, information, and infrastructure and management. As a baseline for a SOA transformation process, organization can use OSIMM to assess its dimension to position the service condition and develop plan and roadmap to achieve the service-matured position for future improvement. From the defined dimensions, there are seven identified levels as shown in Figure 6:

1. Silo level, which shows that the components of the organization are still separately developed without any integration.
2. Integrated level, which shows that the organization uses technology to integrate the IT system for better communication without extension for standards in data or business process, resulting the need of conversion for their connection. The evolution from silo level into integrated level could reduce operational and maintenance cost, and leverage reusability through object-oriented development.
3. Componentized level, which shows that the organization uses components-based development which is better in case of functional granularity and reusability, but still not loosely-coupled which creates limitation in agility and interoperability.
4. Services level, which shows that the organization, as a service provider, applies the basic concept of service-orientation with reduced cost in infrastructure and maintenance, but still has limitation of agility for development of new business process as services.
5. Composite services level, which shows that the organization is able to construct business process for service interactions which reduces cost for business agility, to change the business model based on business strategy and transformation.

6. Virtualized services level, which shows that the organization uses façade design pattern by creating a virtual or abstract service to manage the access between service consumer and service provider, thus resulting a higher level of flexibility and interoperability.
7. Dynamically re-configured services level, which shows that the organization is able to assemble business process at design time to match process flows with the defined SLA, thus offering high organization competitiveness, service availability, and scalability.

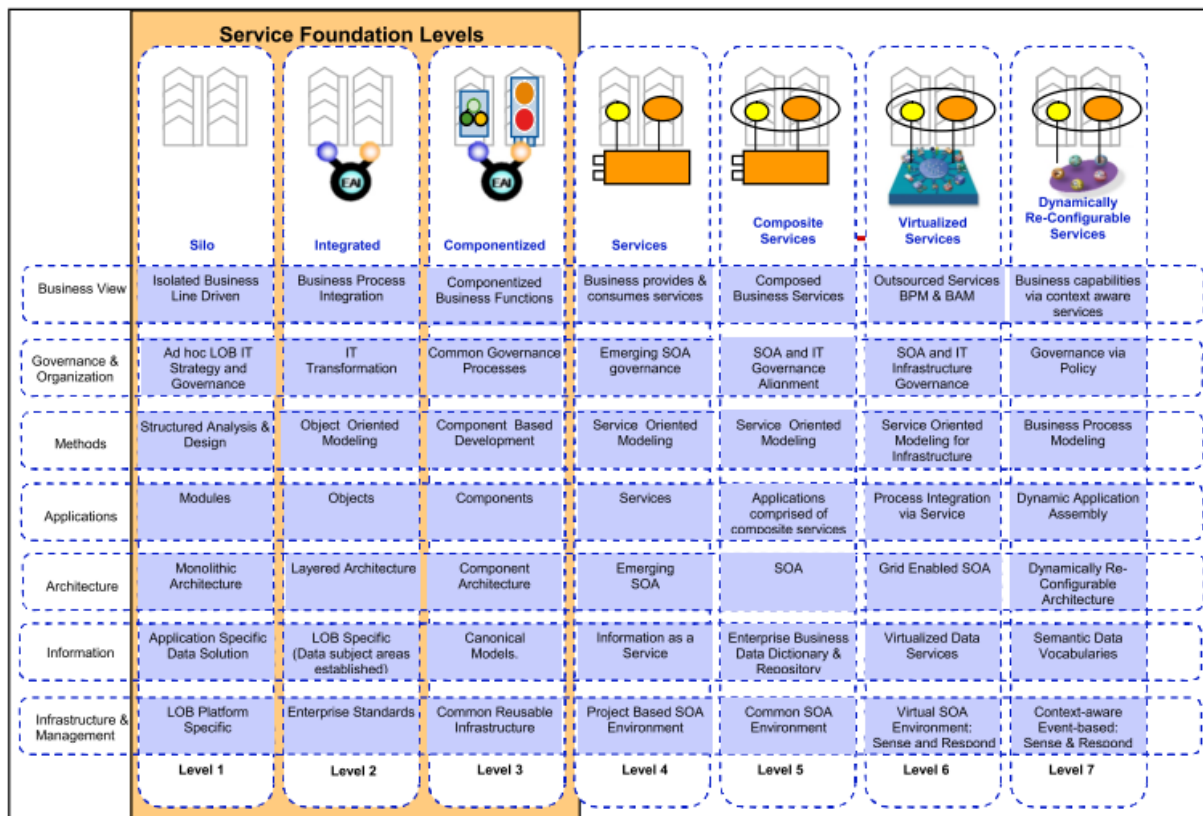


Figure 6. The Open Group Service Integration Maturity Model (The Open Group 2011a)

Based on the roadmap plan, a reference architecture then can be designed as logical view to represent the components of the SOA system in a high level model based on Figure 7 (The Open Group 2011c). The reference architecture can be used as a guideline for designing SOA architecture as it provides blueprint for creating, integrating, and evaluating the architecture. It shows the architectural building blocks in each layer which connects the consumer perspective into implementation by service providers. It consist of three five functional layers of SOA solution and four supporting layers. Operational system layer, service components layer, and services layer address the implementation and interface with a service, while business process layer, consumer layer, and integration layer support the consumption of services, and the rests support cross-cutting concern for non-functional nature of the solution. Those layers are the general logical view of SOA solution to give representation of the needed components to develop a service system.

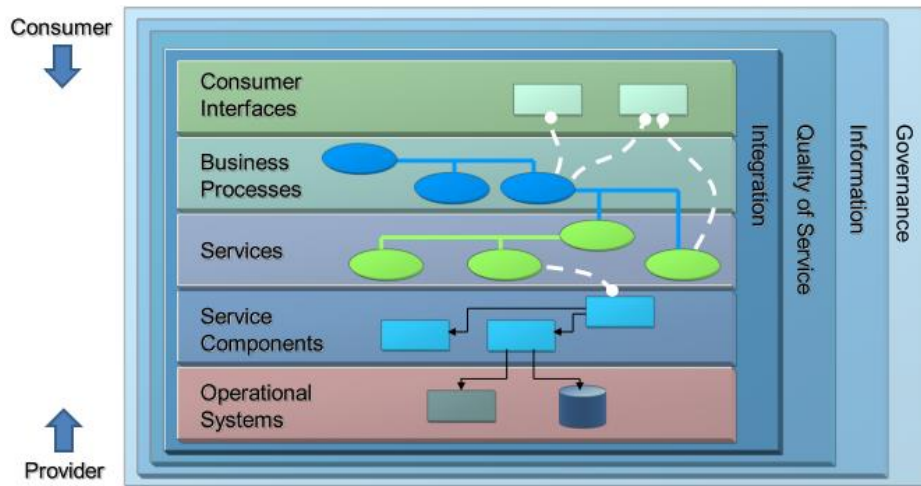


Figure 7. SOA Reference Architecture (The Open Group 2011c)

The SOA-based solution is then developed using SOA methodology. Mohammadi and Mukhtar (2013) reviewed several SOA methodologies and modeling and stated that the combination of Service-Oriented Modeling and Architecture (SOMA) and Service Oriented Architecture Modeling Language (SoaML) allows a service development process in a detailed and systematic way. SoaML (OMG 2012) provides the foundation of the service modeling which is based on UML with extension, while SOMA (Arsanjani et al. 2008) provides the methodology which consist of service identification, specification, realization, and reuse. SoaML is used to identify, model, and specify the services based on these elements:

1. Participant (service provider or consumer)
2. Port (center of interaction between participant and the service)
3. Service description (interaction description which can be indicated through service interface and/or contract)
4. Capabilities (participant's features which can be delivered through service).

SoaML is chosen as service modeling language because it is an extensible language. The advantage of using SoaML is that it can be integrated with Business Process Management (BPM) technology such as Business Process Model and Notation (BPMN) so that the services can be identified from business process and capability. Furthermore, the combination of Model Driven Architecture (MDA) inside SOA development promotes interoperability between application and service using model transformation and automation in each model layer. Likewise, the combination of Business Motivational Model (BMM) also support services identification based on business motivations and promotes organizational interoperability using organization participation in a higher level of abstraction (Nodehi 2014, OMG 2015).

SOA can be implemented using several available service technologies, but it has a close connection with web service along its concept development. Web service is a type of Application Programming Interface (API) which use a specific web service protocol. As defined by W3C (2004), web service is a software system designed to support interoperable machine-to-machine interaction over a network. Under SOA development, web service, namely Web Service Description Language (WSDL), XML Schema Definition (XSD), and Business Process Execution Language (BPEL), can be generated automatically from SoaML using MDA tool. Web service is structured as in Figure 8 and mainly involves three elements:

service interface, interaction mechanism, and service directory. Using the W3C standard for web service, these elements are described by these technologies:

1. Web Service Description Language (WSDL), which is an interface description to define the identity and calling mechanism of a service.
2. Simple Access Object Protocol, which is a communication mechanism between each service or a service with other system which uses XML message and HTTP protocol.
3. Universal Description, Discovery, and Integration (UDDI), which is a service registry, as a central control to publish and locate service.

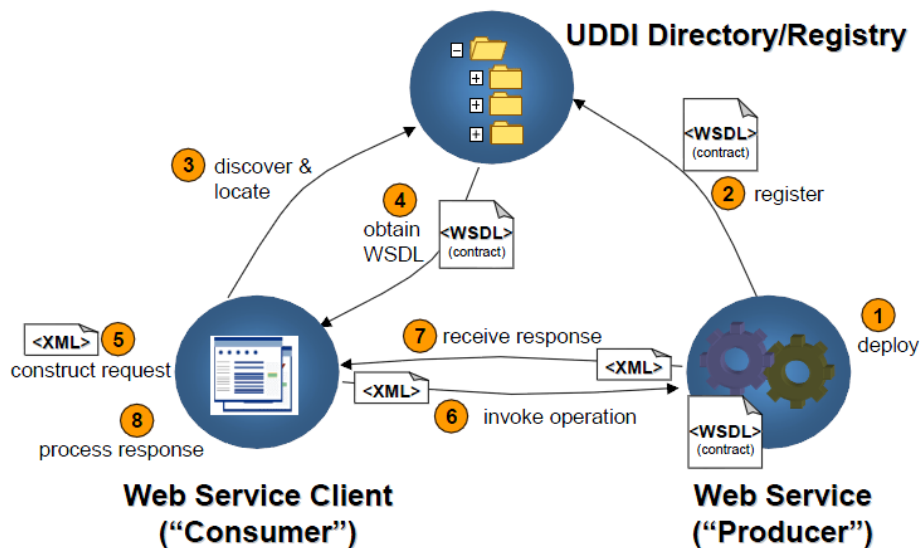


Figure 8. Web Service Building Blocks (Hauch 2005)

After being developed using defined technology, services within SOA then could be orchestrated into a software under cloud-based infrastructure. It could be developed based on Service-Oriented Cloud Computing Infrastructure (SOCCI) which is essential for implementing and managing an IaaS (Infrastructure as a Service) environment. SOCCI is a service-oriented, utility-based, manageable, scalable on-demand infrastructure that support essential cloud characteristics, service, and deployment models. SOCCI is about leveraging the virtualization technology to expose infrastructure of cloud into SOA development. As shown in Figure 9, SOCCI's elements of infrastructure consists of compute, storage, network, and facilities services, while the management building blocks consist of business and operational functions (The Open Group 2011b). In brief, the application of SOCCI into SOA-based software enables the benefits from both SOA and cloud computing.

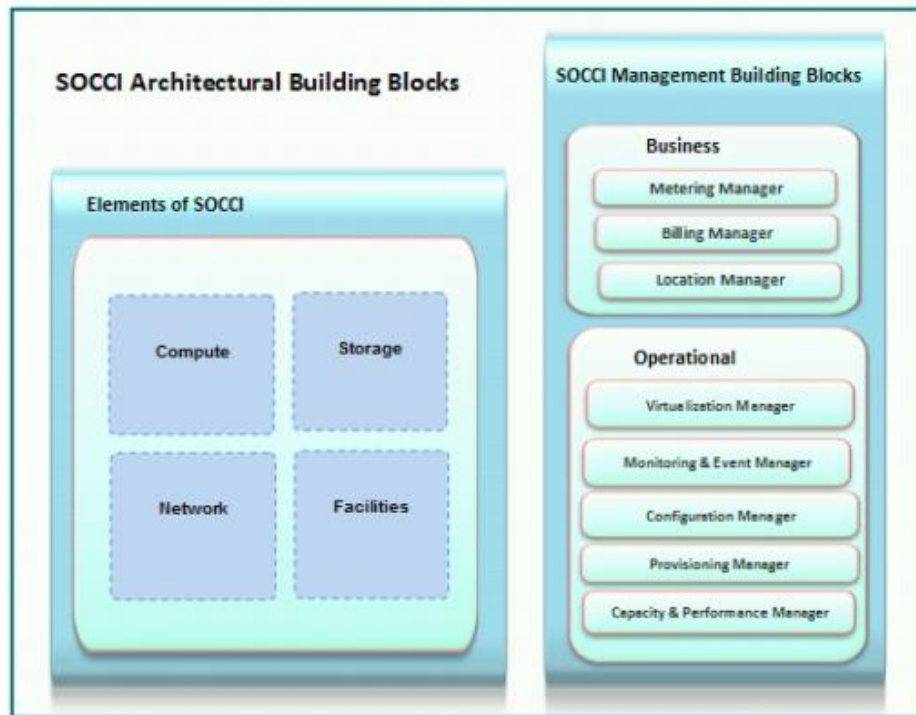


Figure 9. SOCCI Architecture (The Open Group 2011b)

2.3.3 Service Concept Application in SaaS

The combination of service in business aspect and technological aspect is important to create the true service-orientation concept. The possibility of service in business aspect, it does not always involve technology, but more concern on the customer interaction and physical evidence, while in the technology aspect, it is more concern on the technological based interaction support using SOA. However, service is delivered through experience which means that customer still has an important role in service delivery. Therefore, user-centered design, including usability design perspective with ethnomethodology, and service design concept need to be integrated to identify the related software services which align the needs of customer in SOA (Saini, Nanchen, and Evequoz 2013). Then, it will result a service system development which focuses on customer experience.

The combination of service system described with SOA and SaaS will enable the development of a loosely coupled SOA-based SaaS system. SOA provides loose coupling between applications and cloud computing provides a loose coupling between applications and hardware (Letaifa et al. 2010). Besides, SOA-based application leverages the orchestration of service components, and by adding with SaaS model, it will create a new model which enables the access of an on-demand software through pay-per-use or other business model and pricing strategy (Deeter and Jung 2013, Xiaoye, Xianhui, and Weidong 2013). This can be done because SaaS and SOA give similarities in term of reducing cost, provide more agility, and both are service IT solution. SaaS, as a business model, provides the service system delivery model as it describes how to deliver software to provide business service to customers. On the other hand, SOA, as an architectural strategy, provides the service system design model as it describes how to develop software structure by providing small isolated process as service component which give service to other application (Khanjani, Rahman, and Ghani 2014).

2.4 Legacy and Modernization

Legacy system is an existing system with outdated technology (Bjeljac et al. 2014) which needs modernization because of several reasons. Legacy system usually is expensive to support and maintain or use an expensive infrastructure that it needs other solution to make it more efficient. Several legacy systems are also proprietary and on-premise which means that they use single vendor tools and are too dependent with the used technology. If the system is big, there would be too many applications and documentations to maintain and organize. Evermore, for competition purpose, business also need to use modern technology and business process to be able to survive in the market. With the opportunity to use the emerging technology such as cloud and open source, the legacy system would inevitably need to be modernized into an up-to-date system which could be easier to manage and maintain.

Seacord et al. as mentioned by Chung, Chul An, and Davalos (2007) introduced a risk-managed modernization approach to reengineer a legacy system. The step is started by selecting the candidate legacy system and identify the stakeholders. Then, the requirements are specified by eliciting the stakeholders and analyzing the legacy system. Furthermore, the target technology also needs to be analyzed and evaluated to be able to define the target architecture and modernization strategy. Afterwards, the requirements and the modernization strategy need to be reconciled to find a suitable solution. Finally, the modern system can be developed and deployed.

In modernization, reverse engineering and forward engineering are the main important factors, as part of the reengineering process to find the best solution. To understand the legacy system, reverse engineering is needed to analyze the high level abstraction from a defined system as an as-is model. This abstraction contributes to the model-based understanding process so that the legacy system could be restructured, redesign, and reimplemented into a suitable solution. Then, forward engineering is performed to analyze and design the new form of solution as in to-be model.

In case of modernization from legacy to SOA-based system, the evolution can be categorized into two in general: redevelopment, to rewrite the existing applications, and wrapping, to wrap an existing component into a new and more accessible component. From these two categories, there are also variant strategies which come their combinations, which consist of migration, to transform the system into a more flexible one with the original data and functionality, and replacement, to retire the application and replace it with packages from out-source or in-house development.

For the modernization process, most of the found research literatures always shows the application of cloud computing as foundation inside SOA infrastructure as main context. However, this is contrary to the context of this research which is to use SOA concept as foundation to develop an SaaS architecture model. As an illustration, most of the literatures show how to use cloud service such as storage service from Google or Amazon to develop SOA. Otherwise, the desired mean is to show how to develop and orchestrate services based on SOA model, while it could also use cloud-based technology infrastructure, into a software system and then deploy it into SaaS model. Therefore, a careful selection is needed to find out the perfect example as means to cover the foundation of this research.

Apparently, Chauhan and Babar (2011) shows the suitable core mean in question in this research context. They extended the SOA-based software into SaaS model by executing evaluation of components for scalability, evaluation for orchestration, identification of the

components for refactoring, and evaluation of the solution against the target cloud environment. However, the research describe the process in a practical way without general and clear description as shown by Arsanjani et al. (2008). One solution for this is to describe the mean context of the methodology in the clearest way by combining several approaches.

One other best practice, as described by Ionita (2013), is to evolve legacy system into service-oriented system is by transforming into SOA environment, and then transforming into cloud-based SaaS environment. For evolution into SOA, an integrated service-oriented software reengineering could be used which consist of service oriented reverse engineering by understanding the legacy system, target system, and the evolution feasibility, and also service oriented forward engineering by identify service candidate, implementation, and deployment and provisioning (Khadka et al. 2012, Chung, Chul An, and Davalos 2007). Then, the deployed services can be orchestrated as SaaS system components and the platform can be developed using PaaS and/or IaaS cloud providers, either with public, private, or hybrid cloud, with concern to its principles, especially, multi-tenancy, scalability, and security.

2.5 Summary

Service-orientation concept is very vast as it concern on multidisciplinary approaches and emerged to be the key success factor for companies to achieve competitive advantage. As companies need to have agility to adapt with changes, they can not just stay with what the legacy system provides. Modernization became a need for companies as it offers new opportunities for business growth, with a care execution. Several service concepts and technologies are available to support modernization process and bring different roles, such as service design for as service development method, SaaS as service system delivery model, and SOA as service system design model. Through innovation from market and technology development, companies can explore a new business model which can grow their competitiveness.

Based on the findings from this literature review, it is found out that the presentations of framework and methodology for legacy system modernization into SOA-based SaaS system are still separated apart based on different aspects. The discussions about modernization are still limited in the term of legacy into SOA or legacy into SaaS, and only a little discussed about the modernization process for SOA-based system, although the combination of SOA and SaaS is quite common for service system development from scratch. Most of the found literatures described the development and evolution from legacy to SOA using SaaS components from vendors and service providers, which in the end the software itself was not delivered in SaaS model.

However, the SaaS maturity model and SOA maturity model (OSIMM) show that there is a connection between the highest level of both models within their characteristics for improvement from the lower level. This could be a starting point to analyze the key points of modernization to service-based SOA system. Additionally, as service-orientation covers several disciplinaries and approaches, the modernization process also needs to be holistic, including business aspects and technological aspects, and also close to customer point of view. Therefore, this research could give a novel contribution in both academical and practical area by proposing the holistic and multidisciplinary modernization process from legacy system into SOA-based system and demonstrate it to the public transport document generator system created by OPT.

3 Service-Oriented Modernization Framework

This research investigates the issue of modernization of a legacy system, particularly in case of public transport document generator, into SaaS-based system. The purpose is to apply the service-orientation concept, both in business aspect and technological aspect, as an integrated approach. This way, the service provider can apply the benefit of reuse, interoperability, and organizational agility to allow the company to adapt with the competition needs with a flexible business model. However, previous studies has showed that there still lack of comprehensiveness in case of service-orientation methodology, with discussion only limited to the combination of SOA and SaaS, modernization only for SOA, or modernization only for SaaS. As it is a multidisciplinary subject, modernization into a service-oriented system should include several related approaches such as service design, user-centered design, software reengineering, SOA, SaaS, and even cloud computing itself. Therefore, there would need a suitable approach to guide the modernization process from legacy system into service-oriented-based system.

To overcome the issues, a framework for service-oriented modernization process was proposed. Service-Oriented Modernization Framework (SOMF) provides a structured way guideline to perform the modernization process. It uses a multidisciplinary approaches from several existing researches to capture all of the benefits of the comprehensive service-orientation concept. Mainly, the approach consists of reverse engineering to analyze potential functionalities as services from legacy system understanding, service design as forward engineering to design the target architecture and service orchestration, and also stakeholder involvement by taking into account usability and customer experience. Moreover, the framework was design based on model-based development, as model is easy to understand by human and computer, and it allows transformation from model to other model or model to code with the available CASE tools.

The context of SOMF is not just technical modernization, but also business modernization. The context of technical modernization is defined from the system code program to be reversed into architecture and high level abstraction model so that it can be developed into the new cloud service model. However, by adding with the business concern, the modernization process also includes the concern of business changing and customer experience improvement by seeing the business process and interaction between customer and provider. Overall, it provides a general way of thinking which can be applied a company who wants to perform a comprehensive modernization process to achieve competitive advantage.

As a basic conceptual structure, this framework can be viewed as a blueprint, which includes templates and guidelines, for a holistic modernization process from legacy system into service-oriented SaaS system. It provides consultants, business, and IT practitioners with guidelines and options for making architectural, design, and implementation decisions in the development of modernization solution. It defines a roadmap to perform modernization and followed with artifacts and methodology which can be followed as representation of a holistic reengineering process.

This chapter presents the result of the analysis and development of SOMF. This framework was designed based on the interconnection between SaaS Maturity Model and The Open Group Service Integration Maturity Model. Through the highest level of both maturity level models, it was shown that SOA system and SaaS system actually complements each other

through the similar concept of development. From this concept, the characteristics of a system based on SOA and SaaS could be derived to develop the general reference architecture which can be developed with a methodology using multidisciplinary approaches, namely User-Centered Service-Oriented Software Reengineering (UCSOSR). Then, an evaluation model was identified to validate the feasibility of using the framework within a case study of public transport document generator. Finally, the demonstration plan was identified as preparation and initial step for the system development.

3.1 SaaS and SOA Maturity Model

Maturity model is a model to assess the level of maturity of an object based on defined parameters. The model consists of several defined parameter and maturity levels which are defined by the satisfied condition. The assessment is used to show the current position and set out the opportunities of improvement. It serves as a benchmark analysis tool to structure the problem of a company by analyzing the position of a company and develop a development strategy plan based on the improvement roadmap. The assessment method is started by identifying the business objectives relevant to the target assessment. Then, the model is extended by adding desired maturity indicator, in accordance with the business capabilities. The model is used to assess the current maturity model and determine the goal state maturity levels. Finally, the current and target level maturity can be compared to identify gaps and determine the organization's transformation roadmap from current maturity to the desired target. In brief, maturity model is a core guide for organizations for transformation towards improvement targets.

For development of service-oriented SaaS system, SaaS maturity model from Kang et al. (2010) and The Open Group Service Integration Maturity Model (OSIMM) from The Open Group (2011a) were being used. Both models were chosen because they covers the improvement roadmap for both SOA system and SaaS system. As can be seen in Figure 2 and Figure 6, the models show that there is connectivity between SOA and SaaS towards an on-demand service-oriented system. From these models, the capabilities needed for modernization from legacy system into service-oriented SaaS system can be identified to form a single point of view. As a result, these capabilities can then be formed as key aspect of components of the framework which is suitable in this research case.

Based on those transformation characteristics, SOMF facilitates the transformation from legacy into the basic SaaS and SOA, and even enable a further evolution to the highest maturity level. The context of modernization in an organization can be vary depends on its starting position level and the additional expectation for business and technical capabilities. Wherever the position within SaaS maturity model, the goal is to achieve the virtualization level to leverage the use of service-orientation to its extent. This level can be achieved by implementing the multi-tenant database and scheme with cloud computing to create virtual system space with load balancing so that service providers can allocate the computing power dynamically based on the measurement of quantity of service used. Moreover, the services also needs to be developed into SOA with the Service-Level Agreement (SLA) policy optimization to customer by using flexible and dynamic methods for measuring the amount of services used.

As a consequence for evolution into virtualization level of SaaS maturity model, the organization needs to apply SOA principle and at least achieve the services level in SOA maturity model to embark the early phase of SOA application. The organization needs to align

business and IT by using enterprise architecture to model business drivers, enterprise mission, and business architecture into SOA application based on SOA strategy and vision towards SOA governance. The application can be built out of loosely coupled services across the whole organization using SOA pattern of separation of concern and service integration using Enterprise Service Bus (ESB) with common business data vocabulary as data model for the information architecture. The services then are invoked using open standards and first contracts (SLA), and are defined using service description language to describe the services operations. As a result, with the evolution into services level, the basic concept of service-orientation within SOA can be applied throughout organization.

In addition to the characteristics from the application of virtualization level of SaaS maturity model, the organization also can easily achieve virtualized services level of OSIMM with the use of the concept of virtualization and cloud computing. However, the organization is required to achieve services composite level first by taking account to the service ecosystem to align services and business processes. Organization can use the support of business process management to align the organization strategies, businesses, and IT models with the support of enterprise SOA, service management, and SOA governance for integration across enterprise-wide. Information flow and control flow between services can be defined using a composition language, such as Business Process Execution Language (BPEL) to allow a faster and more flexible design of business processes. Moreover, organization also can use outsource cloud services between business partners with the support of Business Activity Monitoring to watch the internal processes and outsourced processes. The virtual services are created by introducing façades to invoke the underlying business and IT services thus tremendously increasing service reusability. In doing so, organization needs to apply SOA governance entirely towards organizational culture and treats SOA service as enterprise assets with SOA metrics and KPI which are integrated into virtualized SOA with ESB and registry. Therefore, the services virtualization can be achieved through the support of concept of service orientation and cloud computing.

Even furthermore, the organization also can extend its evolution into the dynamically re-configurable services level by automatically managing the services based on configuration information to provide flexible transformation. Business processes can be constructed at runtime instead of being assembled by a developer at design time using business process modeling. The services also can be managed automatically based on SLAs and requirements. If for instance the availability of one service is below the required availability, the service can be exchanged as a reaction to the changing circumstances in real time. Besides, the service architecture also can be dynamically reconfigured using Service Component Architecture (SCA), such as FRASCATI framework proposed by Seinturier et al. (2012), to develop distributed SOA system. As the system would be more complex with virtualized ecosystem, a policy-driven governance needs to be applied as cloud platform operators to control the quality and standards of software offered by enforcing different kinds of policy (Bratanis and Kourtesis 2014, Kourtesis, Paraskakis, and Simons 2012). Accordingly, the implementations of this level has already provided in the available cloud service providers in the market and these implementations will ultimately lead to a very loose-coupling system as well as higher fault tolerance with the application of enterprise service on-demand.

Overall, SOMF facilitates a flexible evolution process with a general modernization framework to support organizations' needs based on their business targets. It enables the organization to improve the service application even into the highest level. However, the framework only provides guideline for modernization using several development methods and

technical recommendations. As it is still limited to recommendations of best practices, practitioners can either follow the recommendations or use other methods and decisions. Therefore, the application of the framework and the system implementation can be vary and can affect the evolution roadmap, depends on the practitioners' decisions and the organization's business strategy.

3.2 Service-oriented SaaS Reference Architecture

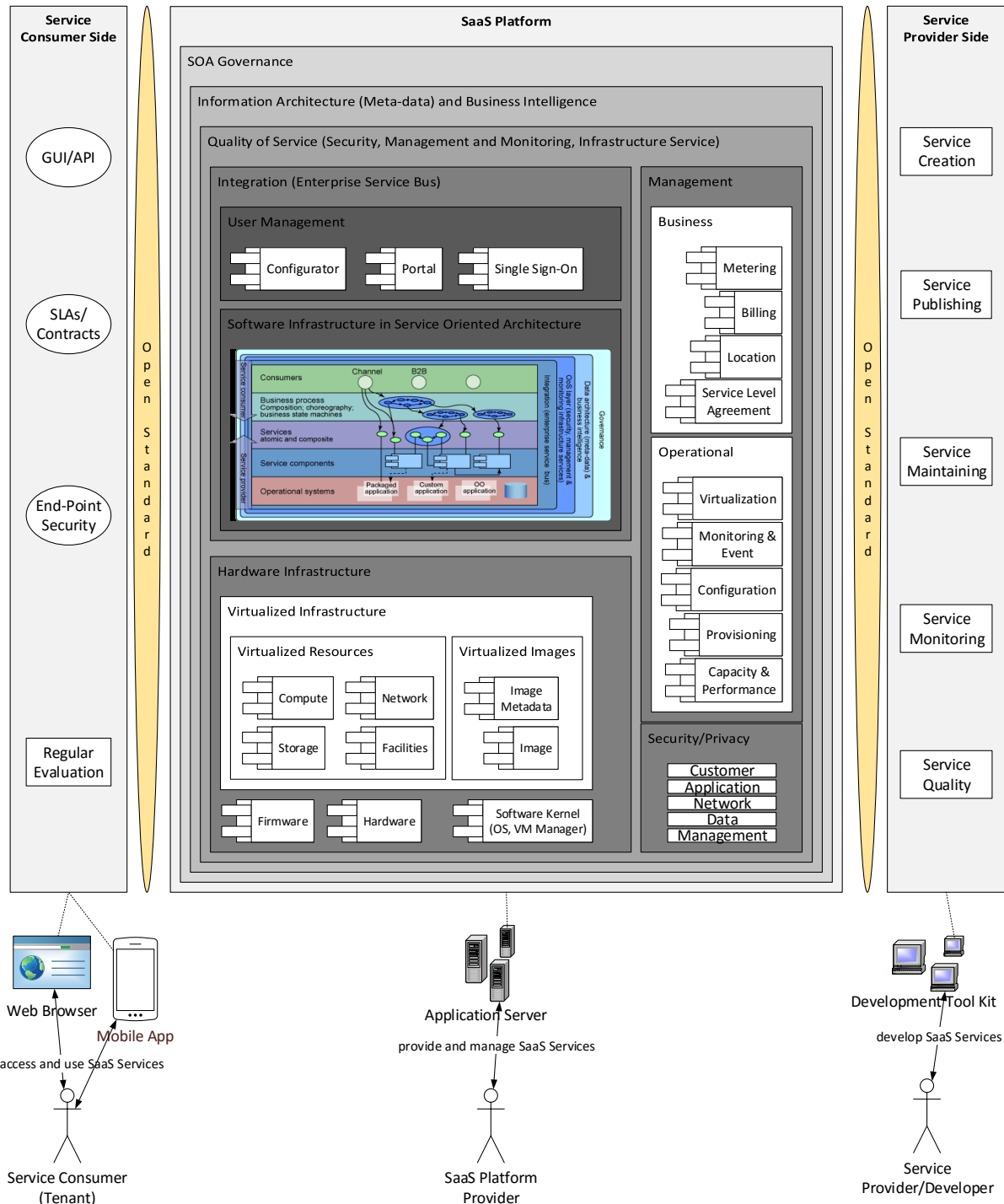


Figure 10. Service-Oriented SaaS Reference Architecture

Reference architecture is one of the key points for delivering a development solution. Reference architecture provides a template solution for designing a solution system

architecture. It shows how to construct components and structure them with interactions between each other into a complete system architecture. The use of reference architecture promotes acceleration of solution delivery by reusing an effective solution from experience and ensuring the consistency and applicability within the problem context.

As any other type of solutions, a reference architecture is also important in the context of service-oriented SaaS system development. The reference architecture serves as a guideline in determining the necessary components to develop the service-oriented SaaS system, which is in line with this research context. Therefore, the architecture needs to be applied with service-orientation concept, such as separation of concern between business and technology, and enables the delivery of the provided services on-demand in a cloud-based environment.

To achieve those condition, the reference architecture used for SOMF was designed by integrating several approaches to describe the necessary components for service-oriented SaaS system. The service-oriented reference architecture was developed based on the reference architectures described in Chapter 2 into an integrated architecture shown in Figure 10. The reference architecture covers several important characteristics of service orientation and SaaS. It shows a high level abstraction of a SOA-based SaaS system composition and can be used as a general template structure for developing an SaaS system using SOA.

The reference architecture includes an integration of SaaS reference architecture and SOA reference architecture. It uses the SaaS reference architecture (Kang et al. 2010, Pang Xiong and Li 2013) as the foundation to show the main necessary components for an SaaS system. Additionally, SOA reference architecture (The Open Group 2011c) is also integrated into the SaaS reference architecture to apply the service-orientation concept into SaaS platform architecture. Therefore, it results an architecture with three main sides: SaaS platform side, service developer side, and customer side. The next subsections present the discussion of each of the three sides.

3.2.1 SaaS Platform Side

The SaaS platform is the main element of the SaaS ecosystem. It consists of the main general components needed to develop an SaaS platform system. SaaS platform provider, as the main actor in this element, can design and develop the architecture and the platform by in-house development or outsourcing from the service providers who provide the service components. As the reference architecture provides the general needed components, the SaaS platform provider can design only using the required components to build the platform, but still needs to take into account to the service orientation and the basic SaaS system requirements. Besides developing the platform, the platform provider also needs to manage the SaaS platform provider to monitor the service usage and the system performance based on the defined SLA to make sure that the system has high reliability and availability. Those properties are managed through the management and security layers which fulfill the non-functional requirements and the basic function of the platform. The management layer is responsible for the platform management in business and operational area, while the security layer is responsible for the platform security with concern on the trust issues since the customers' application and data are served through third-party services. Finally, the SaaS platform can then be deployed into an in-house developed application server or cloud server from PaaS provider so that customers can access and use the service.

As concern in this research, the SaaS platform should apply the service orientation concept with SOA. SOA reference architecture from The Open Group (2011c) enables the

orchestration of services as the SaaS platform's components, and also deliver the SaaS platform application into an integrated service. SOA reference architecture takes the role under software architecture layer by dividing the structure into consumer layer, business process composition layer, service layer, service component layer, and operational system layer. Besides, the cross-cutting layers is also used to integrate all of the service components within the software architecture layer and within the whole SaaS platform through the integration layer, Quality of Service (QoS) layer, information architecture layer, and SOA governance layer. Additionally, the SOCCI architecture from The Open Group (2011b) is also integrated to provide the general structure of cloud based service-oriented infrastructure within SOA structure. SOCCI describes the cloud infrastructure elements and the service management layer which consist of business-related and operational-related. In brief, SOA reference architecture and SOCCI architecture provide the basic principle of service-orientation into SaaS platform towards on-demand application delivery.

3.2.2 Service Provider Side

The service provider side is responsible for the development of services as components to be used in SaaS platform. The service provider can be the internal division of the organization or outsource from third-party provider. Using the development kit, service provider can design and develop the services through the process of service creation, service publishing, service maintaining, service monitoring, and service quality. The developed services can be published into web-based API with a defined interface description so that the service consumer, which is the SaaS platform, can use the service components. Therefore, service provider should focus on providing the best quality service components for the development of the reliable SaaS platform.

3.2.3 Service Consumer Side

The consumer side describes the end-user or enterprise that actually the use the services from the SaaS application in this ecosystem. By applying virtualization and multi-tenancy concept, several service consumers can access the SaaS application as different tenants in the form of web-based GUI, either in web browser or data-generated desktop application, or API to be used as component for their system. The consumers need to work with SLAs and contract which are negotiated between the consumers and SaaS platform providers to equally agree on the service offers and usages. The consumers also should regularly evaluate the SaaS application to ensure that the services meet the SLA requirements. Besides, consumers should also be responsible for the end-point security to ensure the security of the usage media, such as desktop and mobile-based devices and web browser. In brief, the consumers should be the main focus on the SaaS application development through user-centered development as the main actors who will access and use the offered services.

3.3 User-Centered Service-Oriented Software Reengineering

The User-Centered Service-Oriented Software Reengineering (UCSOSR) was developed as a reliable systematic approach for the modernization process from legacy into service-oriented SaaS system. As stated in the introduction of this research, the unique characteristics of service orientation requires a multi-disciplinary approaches methodology, integrating service-oriented analysis and design with user-centered design and software reengineering. The methodology is general and flexible, as it was designed based on service, software, and SaaS

development life cycles, and utilizes a model-driven development process to promote visualization and traceability which leads to better productivity, quality, and alignment between business and IT.

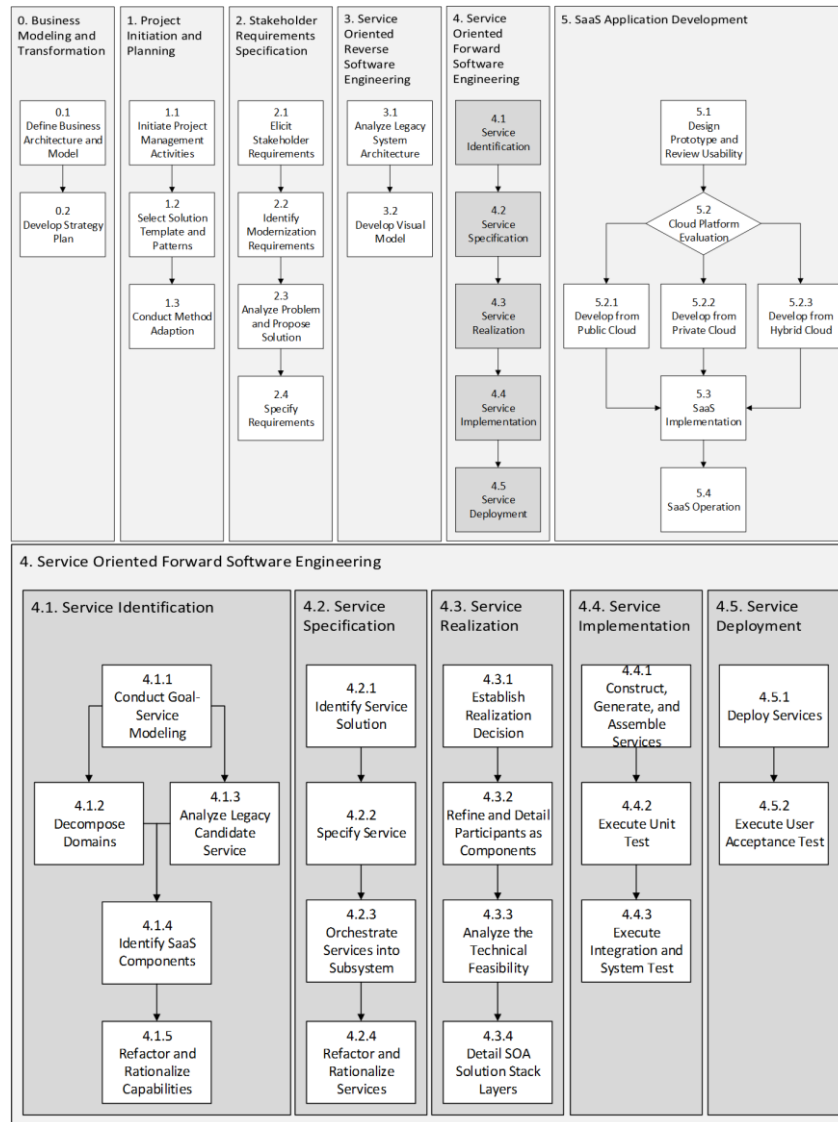


Figure 11. User-Centered Service-Oriented Software Reengineering Methodology Process

This proposed methodology is not a new one, instead it is a form of integration of several best practice approaches towards service orientation which consists of several supporting methods for development in each step. Figure 11 illustrates the process of UCSOSR methodology. The bigger picture of the process is based on SOMA methodology (Arsanjani et al. 2008) and Service-Oriented Software Reengineering (Khadka et al. 2012, Chung, Chul An, and Davalos 2007) for service development and integrated with SaaS development life cycle (Kommalapati and Zack 2011). The service development process also use several methods from Multilevel Service Design (Patricio et al. 2011) to perceive customer experience and design the service system in detail. The concern of user-centered design for SOA (Saini, Nanchen, and Evequoz 2013) is also need to be applied so that the system will be rich in user experience and customer will be satisfied with the offered service interaction.

The UCSOSR methodology is composed mainly of transformation provisioning, service development, and SaaS application development. It consists of five main phases: business

modeling and transformation, project initiation and planning, stakeholder requirements specification, service-oriented reverse software engineering, service-oriented forward software engineering, and SaaS application development. The next subsections present the description and methods used in the five phases.

3.3.1 Business Modeling and Transformation

The UCSOSR methodology is started with modernization envisioning using business modeling. The purpose is to identify the business opportunities of the transformation process by analyzing the business architecture and strategy. The business analysis is important to do before starting the transformation because modernization is not just the development of new system, but also includes business and customer perspective, thus promoting business improvement based on strategy and customer experience focus.

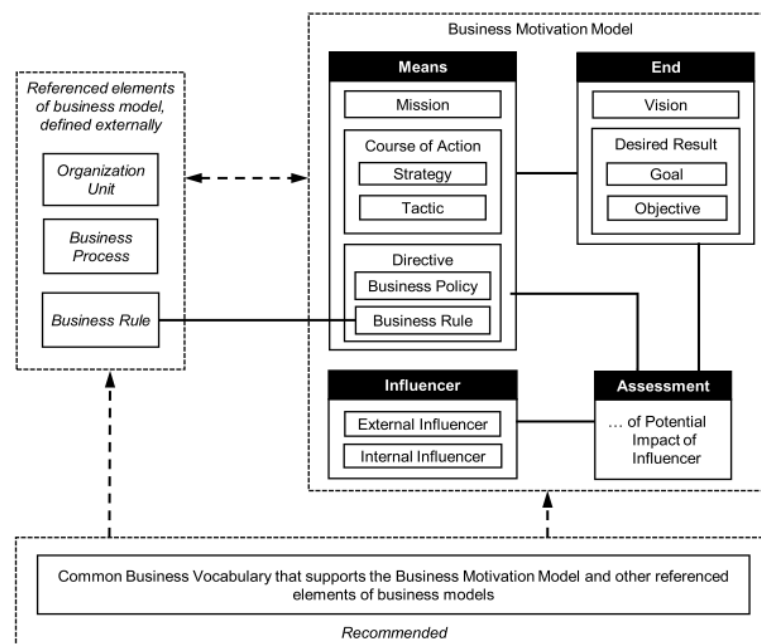


Figure 12. Business Motivation Model (OMG 2015)

The business analysis follows the best practice from several integrated methods. This phase aims to analyze the business architecture and model as a reference to develop the strategy plan. The business architecture and model is developed to analyze the business structure and position as consideration for the strategy plan. The first artifact to be developed is the business architecture which describes the motivation of the organization and the business plan to reach the motivation. The business architecture is developed based on Business Motivation Model (BMM) from OMG (2015) since the model provides a structure for developing, communicating, and managing business plan in an organized manner. As can be seen in Figure 12, the motivation model is mainly determined through the Ends and Means elements of the organization. The Ends determine the things that the organization wants to achieve through the big picture vision and the desired results in the form of goals and objectives. Furthermore, the means determine the things the organization will do to achieve those Ends through its mission and course of actions, consisting of strategies, tactics, and the directives to define how to perform the course of actions.

Additionally to the Ends and Means, the Influencers determine the things that can affect the elements of the business plans. The Influencers can affect the organization both in its

employment of the Means and achievement of the Ends. The Influencers elements is really close-related to the strategy plan definition since the strategy of the organization can be determined based on the position of the organization within the analyzed environment and its assessment (Johnson, Scholes, and Whittington 2008). The Influencers can be defined through situational analysis which consist of:

1. External analysis, to view the condition of the market and competition which come from macro environment, micro environment, and critical success factors of the organization. The macro environment is defined through PESTEL (Political, Economic, Social, Technological, Environmental, and Legal) framework to analyze the broader society that influence the industry within it. Then, the micro environment is defined through Porter's competitive forces framework (threat of new entrants, power of suppliers, power of buyers, product/service substitutes, and intensity of rivalry) to analyze the attractiveness and competition in industry environment. Finally, the customer value can be defined through critical success factors to gain the industry's competitive advantages.
2. Internal analysis, to view the internal condition of the organization to diagnose the strategic capability through its resources, capabilities, and core competences. Firstly, the resources of the organization are defined to determine its assets, which consist of tangible (financial, physical, technological, and organizational) and intangible (human, innovation, and reputation) resources. Then, the capabilities are analyzed to define the capacity to deploy the resources to achieve the desired Ends. Finally, the core competences are analyzed to define the strategic capability using the sustainable competitive advantage criteria (VRIN –Value, Rarity, Inimitability, and Non-substitutability) and value chain analysis.
3. The SWOT (Strengths, Weaknesses, Opportunities, and Threats) analysis, to assess the impacts of the Influencers, both external environment and internal environment, on Ends and Means.

After defining business architecture, the analyst needs to determine the organization's business model. Business model helps developing and managing values proposed by organization based on the market needs to be delivered to customers (Osterwalder and Pigneur 2010). Business model is illustrated through nine building blocks which build the Business Model Canvas (BMC), which consist of customer segments, value propositions, channels, customer relationships, revenue streams, key resources, key partnerships, and cost structures. In the scope of modernization, BMC can be used as a tool to view a company as a service-oriented business by illustrating the as-is and to-be model. The evaluation of the BMC provides the strategy to discover new opportunities and options to create, deliver, and manage the new value to customers.

Furthermore in regard to modernization into service-oriented business system, the maturity model of the organization is analyzed to develop the transformation roadmap plan. As already described in the beginning of this chapter, the maturity model is used to analyze the position of the organization regarding to the SaaS and SOA dimensions. The analyst can determine the current and target maturity level based on the modernization objective, then provide the recommendation to reach the target level. Based on the recommendation, the analyst can identify the modernization roadmap and the strategy execute the roadmap.

Finally, the organization can identify and develop the strategy plan to transform the legacy system into a new system. The identification is put based on the analysis of the business

architecture, business model, maturity model, and the business case or commercial benefit from the modernization. The type of the strategy is aligned into three level: corporate level, business level, and operational level. The corporate level describes the business plan of the modernization project in perspective of variety of markets and distribution of resources among the business, which can be defined with diversification, internationalization, vertical integration or outsourcing, or networks. The business level strategy describes how the organization compete in market segment with the new business model, which can be defined with cost-leadership, differentiation, or focus based on Porter's strategy model. The operational level strategy describes the modernization method that will be applied, either redevelopment, wrapping, migration, or replacement. With the definition of the strategy, the organization then can initiate the modernization process to develop the new business system based on the developed plan.

3.3.2 Project Initiation and Planning

The project initiation and planning phase is the starting process of the modernization project development. The purpose is to plan the project development in case of the scope, time, and cost, based on the modernization goal and the strategy plan defined in business modeling and transformation. The inputs then are processed into project charter, stakeholder definition, project plan, and tailored delivery process. The phase is divided into three steps:

1. Initiate project management activities, to formally start and manage the project by identifying the project description and stakeholders into project charter and plan based on the Project Management Body of Knowledge (PMBOK) guideline (PMI 2013).
2. Select solution template and patterns, to select the suitable custom SOA and SaaS solution template, such as specific methods, techniques, tools, etc., to create a tailored project delivery process based on the identified plan.
3. Conduct method adaptation workshop, to inform the plan and tailored delivery process into all of the project stakeholders so that they can have the same perception about the project development.

3.3.3 Stakeholder Requirements Specification

After defining the modernization objective and the project plan, the organization then needs to specify the modernization requirements from the stakeholders. The requirements determine the needed capabilities to modernize the legacy system into a service-oriented system (Pohl 2010). The source of the requirements can be identified from the stakeholder needs, legacy system, documents, standards and regulation, and domain information, to be compiled into a formal specification document. The phase consists of four steps:

1. Elicit stakeholder requirements, to identify the requirements from the sources regarding to the modernization project and get better understanding of the legacy system. The elicitation can use conversational methods (interviews, workshops, brainstorming), observational methods (protocol analysis, ethnographic study), analytical methods (requirement reuse, cards sorting, laddering, repertory grid), and/or synthetic methods (JAD/RAD, scenarios, prototyping). Within user-centered service design, the elicitation process needs to be done using the combination of those methods with the concern of customer and user experience to enrich the experience of accessing the system.

2. Identify modernization requirements, to analyze the requirements defined from elicitation to be developed into innovative requirements for modernizing the legacy into new system.
3. Analyze problem and propose solution, to analyze the issues and problems identified in the requirement definition to develop a solution proposal into the requirement specification. This step identifies the already existing elements within the legacy system to determine the required elements to develop the proposed system.
4. Specify requirements, to develop the formal specification document of the proposed solution regarding the modernization project.

3.3.4 Service-Oriented Reverse Software Engineering

As mentioned in literature study, reverse engineering is one of the important aspect in modernization. To complement the requirement elicitation, the legacy system needs to be analyzed to get a better understanding about the system. The analysis is done through experimenting the application, examine the source code, and also investigate the related documents. Then, the legacy system is presented through visual model so that it can be used together with the modernization requirement specification to identify the candidate services. The legacy system is visually modelled in four perspective:

1. Use case view: to show the use case of the system by each type of users.
2. Entities view: to show the information and data which are used within the system.
3. Behavior view: to show the process and flow within the system.
4. Architecture view: to show the structure, design, implementation, and deployment model of the system.

3.3.5 Service-Oriented Forward Software Engineering

After the modernization requirements is specified and the legacy system is analyzed, the service then can be developed using service-oriented forward software engineering. This phase aims identify the possible candidate services and develop them using service technology as components for the SaaS system. The service development process comprises of five detailed steps: service identification, specification, realization, implementation, and deployment. The execution of this phase is flexible based on the defined custom solution template for the particular project, although the purpose is still the same. In case of modernization, this phase needs to take into accounts the identification of the candidate services from the legacy system and the needed services which satisfy the needs from the business process and SaaS features. The candidate services then are selected and specified to define the suitable SLA. Then, the services are implemented, either by in-house development or outsourcing, and deployed into cloud services using service technology. Furthermore, the detail explanation of each step is described in the next subsections.

3.3.5.1 Service Identification

The service identification process serves to identify needed elements of SOA development. Since the source of services may be vary and different, this process needs to use multidisciplinary approaches to identify the services to meet the development requirements.

The methods will complement each other to be able to identify a full set of the needed services. The candidate services are identified from the modernization requirements and legacy system analysis to be assessed and rationalized to find out the needed services which are aligned with the business needs and capabilities. Then, this process can produce SOA Business Architecture Model which consist of goal-service model, service concept model, business process model, and capability model.

Since services are aligned with the business goals and strategies, the first step of service identification is to conduct goal-service modeling. A high-level goal which describes the general statement of business goal related to the project is decomposed into subgoals which comprise the detailed level which need to be achieved first. This hierarchy structure leads to a set of actionable goals which are aligned with the services needed to achieve the goals. The model can be represented using a matrix table which describes the goal and subgoals based on the identified key performance indicators (KPIs), metrics, and the candidate services. Another model representation also can be illustrated using visual model such as KAOS diagram which shows the hierarchy of the goals in AND/OR abstractions (Lapouchnian 2005, Respect-IT 2007). By using the business goals, the service identification is focused based on business process, resulting in identification of those services which would have the greatest business impact and creating traceability between business goals and services.

After identifying the business goals, the focus of the service identification process is placed in domain decomposition. The scope consists of business domain, to analyze the functional area of the organization as service components, and business process modeling, to identify the organization's business process as service flows for orchestration. The functional area of business domain represents the owners of the services (participant) and is partitioned into the responsible for the business functions, business entities included in business functions, and the associated policies, rules, and subsystems. Furthermore, the business process represents the dynamic view of the organization and the service flows which is identified using Multilevel Service Design (MSD) into customer value constellation, service system architecture, and service experience blueprint (Patricio et al. 2011). The framework put the focus of the process in customers' point of view to achieve the purpose of service design through user-centered and customer-experience-focused. Then the final defined process is modeled using Business Process Model and Notation (BPMN) to be able to integrate with SoaML as candidate services (OMG 2011). Finally, the integration of these methods can help better understanding of how the business works in the organization as part of key elements in service identification.

While domain decomposition represents a top-down method, legacy system analysis is performed as a bottom-up method for service identification. The analysis is made based on the reverse engineering done in the previous phase. The purpose is to identify the existing assets which can support the business needs realized in the service development. In case for modernization from legacy system, the functionalities of the legacy system can be wrapped in service interface so that it can be deployed as a service with communication between service consumer and the legacy system's services. This services then can be incorporated as candidate services within the new system.

Moreover as the context is modernization into SaaS system, the next step is to consider the needed services which can meet the SaaS characteristics. While the previous steps concern on the business-focused services, this step focuses on the identification of services needed to deliver the system within SaaS environment. These services support the business and

operational management of the system, the security of the system access, and other nonfunctional requirements which are required in SaaS system. While this step is only focus on the identification of the candidate services, the implementation decision for these services will be executed in the next phase.

Finally, the identified candidate services from those previous step are refactored and rationalized to develop an integrated service portfolio which is suitable for the project scope. Service litmus test is performed to the identified candidate services to determine the capabilities which should be exposed through service interface which then can be modeled into capability model. The capabilities need to be reviewed to identify redundancies and opportunities for refactoring. The capabilities then are refactored and rationalized based on the service granularity which is determined by the functional affinity and cohesiveness of the candidate services. This granularity determines the candidate service refactoring so that lower-level services can be grouped under a higher-level service. This review process needs to be done continuously to verify the candidate service model with business stakeholders in order to maintain the level of the relevance of the services.

3.3.5.2 Service Specification

After identifying the final candidate services, the services then are specified in detail to elaborate the design view of the service model. The design focus is to identify the composition of services, the dependencies of services on other services, components, or applications, the flow among services, and the nonfunctional requirements to enable the business functions and processes. The output from this step is SOA Logical Architecture Model which consist of the high-level service architecture model, service contract model, service interface model, and service data model.

The first step in this phase is to develop the service solution based on the candidate services into service model. The identified candidate services need to be reviewed to define the final services. The objective is to ensure that the services are business aligned, composable, reusable, technically feasible, and has external description. The final identified services are then arranged into service architecture model which shows the network of participants to define their requirements and interactions. This diagram show the context view of the service architecture to determine how the participants work together for the organization in a big picture without required process management

The next step is to specify the detail context of the services, which is the core process of service modeling. The service specification decision will define the realization and choreography of the services to be able to work with other services to execute the business processes. The services specifications then are modeled in SoaML using these approaches:

1. Simple interface approach, which focus on the one-way interaction from the participant through port which is represented as UML interface.
2. Service interface approach, which focus on the two-ways interaction between services through interface description provided by the particular service.
3. Service contract approach, which focus on service specification definition to manage the connection of value exchange between service provider and consumer.

The SoaML development combines both service interface and service contract approaches. The service contract specifies the roles played by the interacting participants, while the

service interface defines the interface between participant components for a particular service. The role type defined in a service contract is the service interface which describes the responsibility of the participants. As service interfaces are defined from the exposed capabilities, they may define the Quality of Service (QoS) policies and link to the goals and value propositions defined in business model. In short, service interface describes the provided and required service operation and can fulfill the service contracts or realize use cases.

The service interfaces use the input parameter from defined from the service data model. The domain business entities identified in service identification represent the information in the form of data model used to implement the services. Service data is the abstraction of this information exchanged between participants. This exchanged messages use value objects or data transfer objects to reduce coupling with support of both document-centered and RPC parameter style as message type.

After the service elements have been specified, the services are orchestrated into subsystems. The subsystems represent the logical IT boundaries for business functionality and are identified by the functional decomposition. A subsystem may be identified by service components which provides the relevant functional and business services. This step is the beginning of the transition from service solution specification into service realization design to identify the needed components that underpin a service.

Finally, the specified services are refactored and rationalized to define the final version of the services specifications. The services specifications are reviewed to avoid redundancies and to find opportunities for refactoring. The review process will result the final version of the detailed service model.

3.3.5.3 Service Realization

After the services are already specified, the next phase is to realize the services to design how the services are provided. The objective is to decide which participants will provide and use what services, since this decision implies to service availability, distribution, security, transaction scopes, and coupling. The services components provided by the participants then are arranged based on the service-oriented SaaS reference architecture to design the complete view of the SaaS system. Then, the result is the SOA Solution Architecture Model which consists of participant model, service solution stack layer architecture, and the decisions regarding service realization.

The first step in this phase is to establish the realization decision. The decision is made based on the technical feasibility of the architectural decision and the risk factors through the extensible prototypes. The architectural decision can be defined based on the selection and instantiation of service patterns for information realization, Enterprise Service Bus (ESB) patterns for integration scenarios, and rule patterns for rule realization. The realization decision is also defined based on components realization from different kind of sources, either developing in-house, purchasing already-built services, transforming existing services, subscribing cloud-provided services, integrating with legacy's functionality, and outsourcing from partner providers. The decisions made in this step will affect the service implementation and deployment in the next phases.

After making the realization decision, the services components are designed by identifying the participants who provide and/or use the specified services. The participants are detailed

further to introduce the technical details of service providers and consumers. The participants are instantiated and assembled using the ports and provided interfaces. The assembly process aims to orchestrate the participants' services to minimize coupling of the architecture by:

1. Avoid having components realize and use interfaces directly, instead through the service and request points.
2. Avoid direct dependencies between components by using service channel connection between request and service points.
3. Avoid dependencies between component ports.
4. Avoid subsystem components that contain other components by putting participants in packages.
5. Separate component specification from realization

Finally, the defined service components are arranged into detailed SOA solution stack layer based on the reference architecture. The components of the solution architecture represent the instance of the reference architecture's layers. The architecture facilitates the communication of the solution as representation of the system evolution which is in accordance with the stakeholders' needs.

3.3.5.4 Service Implementation

The service implementation phase aims to construct the designed service architecture using service technology. The specified service architecture can be implemented using several technology such as CORBA, REST, Web Service, etc. However, Web Service technology is used in this research since the technology also grew along with the development of SOA. The objective of this phase is to implement the design service model into Web Service using the model transformation and generation. Using the available SOA and MDA tools, it is possible to generate the Web Service artifacts from SoaML and BPMN into XSD, WSDL, and BPEL. The XSD describes the service data, while the WSDL describes the service interface description and BPEL describes the business process execution for the service choreography. After the generation, developers still needs to develop the services code implementation to the generated artifacts to construct the well-built services. After the implementation, the developed services are then tested using unit testing to check the individual level of the service and also integration and system testing to check the whole level of the service assembly. Finally, the well verified and validated services are developed and ready to be deployed.

3.3.5.5 Service Deployment

In service deployment, the developed services are deployed in the defined servers, either in-house built servers or cloud-provided servers. The deployment on cloud aims to make the services availability high so that they can be used in other application development, improvement, and integration, promoting the service reusability. This phase focuses on packaging, provisioning, executing user-acceptance testing, and deployment of services in the production environment, along with monitoring and management of services. The deployed services need to be continuously maintained to keep the availability and reliability of the services in accordance with the user needs.

3.3.6 SaaS Application Development

After developing the service components, the next process is to develop the SaaS application which can be accessed by the customers based on the deployed services. The process is started by designing the prototype of the platform to test the interaction that will be experienced by the customers based on the defined usability criteria. Then, the usability testing is performed to the developed prototype to get the customers' feedback for experience improvement. Besides, the market-available cloud platforms also need to be evaluated to make the decision of the platform development and deployment. The decision is made regarding either the SaaS system will be hosted in public, private, or hybrid cloud platform based on the considerations in economic, capability, supportability, security, compliance, and operation (reliability, availability, scalability, performance, and disaster recovery) perspectives. After the prototype has been finalized and the cloud platform has been decided, the SaaS application then is implemented in accordance to the technological environment supported by the cloud platform. The implementation process is executed iteratively and aligned with the system and alpha testing to ensure that the system will work in accordance with the requirement specification. After the SaaS system has been developed, it is then deployed in the defined cloud platform and launched in two phases. The first phase aims to perform beta testing to get the evaluation and feedback of the SaaS from the customers to check the acceptance of the services. After applying the improvement, the final version of the SaaS is then launched and operated with continuous monitoring, performance evaluation, and tuning. The insights from the evaluation, subscription, and usage process are always captured to improve the quality of the offered services in accordance to the SaaS operational characteristics to develop strong customer relationships.

3.4 Service Development Tool

As the development of service technology is growing, the framework is also supported with the computer-aided software engineering (CASE) tools which are available in the market. The CASE tools support the service development for SaaS system, starting from requirement capturing, goal-service modeling, business process modeling, and service architecture modeling. The use of CASE tool also serves to keep the traceability of the artifacts developed in each phase so that the development process can be performed in an integrated way. The review of the available tools can be seen in Table 2.

Table 2. CASE Tool Review for Development of Service-Oriented SaaS System

No.	Tool	Version	Advantages	Disadvantages
1	IBM Rational Software Architecture 7.5.4 or latest ¹	Paid and free trial for 30 days	RSA provides SOMA method for service solution design, which is referenced with the proposed framework, and full integrated support with designed template for service development and generation into Web Service artifacts with MDD. The software is still continuously supported at the present time. Besides there is a possibility for access to the free version using IBM Academic	The access of the full version software is still limited to the paid version, using the free trial, or registering in IBM Academic Initiative.

¹ <http://www-03.ibm.com/software/products/en/ratisoftarch>

No.	Tool	Version	Advantages	Disadvantages
			Initiative, despite that it is not the latest version, but still suitable for the project scope.	
2	ModelDriven.org	Open source	It supports for SoaML modeling and production of executable Web Service implementation with Eclipse and JEE technology.	It is not currently maintained as the latest software and the project stopped in 2009.
3	NoMagic MagicDraw bundled with Cameo SOA+ from ModelDriven.org ²	Paid and free trial for 30 days	It integrates the main features of SoaML development from ModelDriven.org, and provides the other modules needed for service development.	The modules of each modeling tools (business process, SOA, etc.) are separately sold.
4	Modeliosoft ³	Paid with 10 days of trial and open source	It provides a free open source tool for SoaML designer which supports business process modeling and service modeling. The paid version provides an integrated support from requirement, goal modeling, business process modeling, service modeling, and generation into Web Service.	The SoaML Engine module, which provides the generation of Web Service implementation from service architecture, is only supported in paid version of 1.2, and is not added in the latest version.
5	Sparx Enterprise Architect ⁴	Paid and free trial for 30 days	It provides SoaML modeling with Service Oriented Modeling Framework and is integrated with other modules from requirement into Web Service generation.	The access of full version is limited on paid and trial version, and the provided framework is not referenced in the proposed framework.

3.5 Demonstration and Evaluation Plan

The proposed framework needs to be validated in a demonstration of case study to show that the framework can be used in modernization project of legacy system into service-oriented SaaS system. The need of the Service-Oriented Modernization Framework (SOMF) arose when an industry case appeared for modernizing the legacy system into service-oriented system. Furthermore, the instrumental case study performed in the industry environment aimed as an opportunity to learn further and gain insight from the exploration and demonstration of the proposed framework.

As described in introduction chapter, the demonstration was performed in OPT. The project aimed to modernize the Infopub system into public transport document generator as a service in the cloud environment. The demonstration illustrated how the phases in SOMF can be carried out using the described methods related to modernization process, from envisioning using business modeling and transformation, into project initiation and planning, requirement specification, reverse engineering, service development, and SaaS application development. All of the required data for the development process were obtained from the exploration of the existing system and the target system through interview with the stakeholders, legacy system application exploration, document review, and research through internet. The data then were

² <http://www.nomagic.com/products/magicdraw-addons/cameo-soa.html>

³ <https://www.modeliosoft.com/>

⁴ <http://www.sparxsystems.com/>

analyzed and put into artifacts development described in the framework to design the target system. Finally, the demonstration would result the structure of the Web Service implementation and the design specification of the SaaS application which could be developed further into a complete service system.

The development of service and SaaS architecture needs an evaluation model to ensure that the framework demonstration could be validated and verified. The evaluation process was intended for both the framework and the demonstration. Scientific-experimental model was used as an evaluation model since it takes values and methods of evaluation from the existing research to check the desirability of impartiality, accuracy, objectivity and the validity of the information generated. The evaluation was performed using the evaluation criteria for modernization and SaaS quality model as shown in Figure 13.

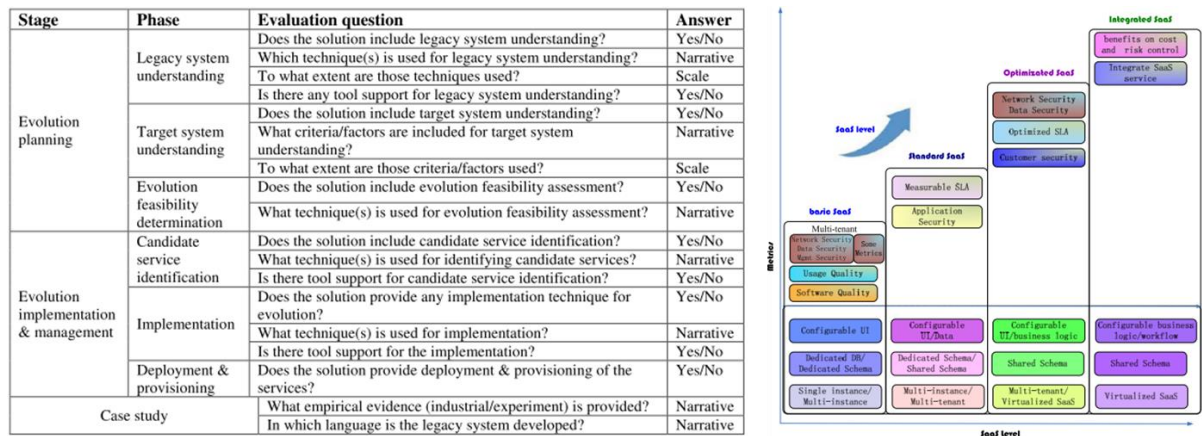


Figure 13. Evaluation Criteria for Modernization Process and SaaS Quality Model (Khadka et al. 2012, Pang Xiong and Li 2013)

3.6 Summary

This chapter describes the solution of the modernization process through the proposed Service-Oriented Modernization Framework (SOMF). SOMF aims to guide the process of modernization from legacy system into service-oriented SaaS system based on the custom transformation roadmap defined from the business targets to achieve the business and IT alignment towards business agility. The framework describes the SaaS and SOA maturity model as assessment for improvement roadmap plan, service-oriented SaaS reference architecture as general architecture for component guideline, and user-centered service-oriented service reengineering as modernization process methodology which integrates several methods from multidisciplinary approaches. Several market-available development tools was also presented to support the modernization process in models and artifacts development. Then, the demonstration and evaluation plan was also described to present how the requirement for the demonstration and evaluation would be gathered and processed to show how to use the framework in real industry case study. This framework was expected to be a suitable solution for the need of multidisciplinary approaches for service modernization which can be widely used. Since the solution has already described in this chapter, the demonstration of the framework finally could be performed and discussed in the next development and discussion chapters.

4 SaaS Services Development

The process of modernization from a legacy system into a service-oriented SaaS system is described using a case study of public transport document generator or Infopub system. The case study was used to demonstrate the application of the framework into an industrial environment. The detailed description of how the public transport document generator system works is explored to represent the case study to test and evaluate the framework. The case study is presented based on the structure of User-Centered Service-Oriented Software Reengineering (UCSOSR) methodology and divided into two chapters. This chapter focuses on SaaS services development, while the next chapter focuses on SaaS application development. This chapter explores the initiation and services development process, from business modeling and transformation, into project initiation and planning, stakeholder requirements specifications, service-oriented reverse software engineering, and service-oriented forward software engineering.

4.1 Public Transport Document Generator

Public transport document generator is an information system which automates the production of public transport information. This system is part of public information system which serves to provide transport information to public travelers (IMTT 2011). The public transport document generation enable an automatic generation process which contributes to decrease the cost from human resource and time to analyze network data into graphic documents, thus ensuring normalization. The information could directly reflect the image of the public transport companies, contributing the service and quality improvement for the customers. It uses an innovative technology which allows public transport companies to automatically create a complete public information with several features for management and configuration. It can be easily adapted to client's needs with several data sources and design graphics in the format of XML (Extensible Markup Language) document and SVG (Scalable Vector Graphics) which allows high level of configurations.

Infopub is a public transport document generator platform which was developed by OPT in a desktop-based client-server architecture. This application service provider produces a graphic documentation with advanced design which adapts to the data contents. Infopub was developed with the concern on integration with other system for solving issues on public transport information and rapid production of advanced graphics layouts so that the content could adjust dynamically based on the needs, form, situation, and object references. It is also connected to several external system, such as the real time schedules systems based on the vehicle's position, or supportive dynamic system, such as Infoboard and Move-me.

The automatic production of public information will create a standardization of format and media, thus giving several advantages. This information provides better visibility and accessibility of the companies' offers to customer so that they will feel more secure of the transportation option. The automatic process will give cost reduction from the layout production and effort of the document generation which usually requires manual editing and resource allocation for graphic design. It also can speed up the production process of mass documents in a shorter period of time. Besides, the standardization of information will reduce the error which usually produced from information normalization. Finally, this system gives

an ease of configuration of the graphic generated style as the documents are produced in editable vector format.

4.2 Business Modeling and Transformation

In this phase, the business scope of OPT, especially regarding Infopub service, was modeled and analyzed to identify the optimization and transformation roadmap and plan. The analysis could be the trigger to decide the decision for the whole modernization process of Infopub service. It took account the OPT and Infopub's business architecture and model, situational analysis of external and internal condition, and the maturity model, which resulting the strategy definition for modernization process. Those models take an important role for the evolution process because modernization is not just transforming the platform into web environment, but also transforming the business model of OPT and Infopub service in accordance to the modernization envisioning.

4.2.1 Business Architecture and Model

Business architecture and model were used to inspect OPT's blueprint to better understand how the company works, especially in the scope of Infopub service. They served to connect the business objective with the business strategy and capability which could act as service candidate. They were developed based on the organization's structure from its website and description from the director. Then, they were developed and analyzed using Business Motivation Model (BMM), Business Model Canvas (BMC), and maturity model.

Business Motivation Model (BMM) was used to determine OPT's business plan. The model describes what OPT wants to achieve, how to achieve them as an entire enterprise, and how it relates specifically through Infopub service. This section investigate the Ends, which were described through vision and goal, and Means, which were described through mission and strategy of OPT. The Ends and Means of OPT as an enterprise is presented in Table A-1, which shows that Infopub is a product service which were developed in accordance to the vision of the public information. OPT needs to evolve the public transport document automatic generation platform into web in accordance to the vision and goal of leadership for support in public information service. Therefore, the Infopub service can grow stronger so that it can attract a bigger market and opportunities with a flexible service offer.

Business Model Canvas (BMC) describes the general view of OPT's business model, showing the products/services that the company offers and the strategy to deliver them. The model is used to identify and create the strategy blueprint for Infopub service in accordance to the evolution plan. As shown in Figure A-1 in the blue boxes which represent the current business model (as-is), the Infopub service was firstly developed to provide a platform to produce graphical transport information to public for Portuguese transport companies. However, the business model can be expanded with more variant of value propositions and customer segments as shown in the green boxes which represent the business model after the SaaS application.

As a company with focus in IT-based solution development for transport system and public information, OPT bases its value through the delivery of the solution products. As complementary of the local solution deployment, OPT can provide an easy document generation of every kind of transport-related information based on the provided networks and schedule using web and cloud technology. Therefore, the solution can be easily accessible

through web browser and web-based desktop/mobile application. Moreover with the on-demand and flexible characteristics from SaaS, OPT also can deliver the services based on the customers' demands by considering the solution features and price options, including a free trial version. As a result, the customers can choose the variety of the services based on their solution needs with the offered capacity and feature-based freemium pricing strategy:

1. Free version with several limitations, such as only allowing customer to change the logo, using OPT's watermarks for the output in PNG, limited data store, and limited time period.
2. Premium version with flexible pricing package based on defined data store capacity and feature, such as paid service per rental contract or per document generation, or different price based on variables:
 - a. The output choice, either just in PNG, or in PNG and SVG.
 - b. Access to configuration of templates and design.
 - c. Access to BusMap and/or BusSched generator.
 - d. Request for custom generators and layouts.

By expanding the service offers, OPT can gain a wider variety of customer segment. From becoming the leader of the associated service business, the company can reach the international market with different kind of customers, both individuals and transport companies. The group of transport companies also can combine their resources to provide information about public intermodal. If the customers are still in doubt with the provided services, they can access the trial version for evaluation purpose before deciding to use the full offered services. This strategy will attract new markets as customers from anywhere can access the platform and try the provided service, thus increasing the customer acquisition. Moreover with the suitable of sales and marketing promotion strategy, the company can increase the customer retention so that the customers will move to the premium service and stay to use the offered services.

As a consequence for the expanded services through cloud, the company also needs to improve the customer experience through channel and developing customer relationship. A reliable integrated multiplatform is needed as main interaction medium for customers to access the offered service. As the platform becomes the main channel, OPT also need to concern on sales, marketing, and customer service delivery strategy to support the customers in using the offered services so that OPT can gain their trust, thus improving customer relationship. OPT needs to develop the platform with a full complete contents so that the customers can find every needed information and access in self-service. If customers need any support, OPT needs to give a dedicated personal assistance which could be different for every customers based on their needs. OPT also needs to develop an Infopub community in the form of news which is already exist, and added with blog and forum so that OPT can share new and interesting information about the services usages and customers can discuss with other customers, thus creating a co-creation service between customers. In brief, the customer support services take an important role so that the customers can perceive a good physical evidence through online experience.

Finally, the maturity model analysis was used to assess OPT's capability in general and Infopub in particular to develop a modernization roadmap plan. Based on the business objective of the project, the current condition of OPT and Infopub system were assessed based

on the maturity dimensions. From the overall assessment, the goal state maturity level could be determined based on the minimum criteria of maturity model of an SOA-based SaaS system discussed in Chapter 3. The SOA maturity model is analyzed thoroughly to achieve the optimized SaaS application and the OSIMM is analyzed based on the business dimension to achieve the evolution which complies with business objective. Then, the comparison between the current and target level could be defined, resulting in a plan recommendation of transformation to achieve the desired target.

OPT envisioned a service-oriented SaaS system for public transport document generator with the application of SOA and cloud computing to improve business model and gain business agility and flexibility aligned with the service IT implementation. Overall, the current Infopub system was a desktop-based application service provider which connects to server through client-server architecture. The focus of the company was still application oriented, with little consideration of services as company assets, thus there was still a little alignment between business and IT. Within SaaS maturity model, Infopub currently placed at level 2 (standardization level) since it provided a standard data structure and multi-instance application system configurable based on the customers' needs. While for service orientation aspect analysis using OSIMM, Infopub currently placed level 2 (integrated level) based on the business view dimension since the business sees IT as application platform for line of business in automatic generation rather than business composite services that can be created from other services.

By considering the characteristics of both SaaS and SOA system, the Infopub needed to be evolved to realize the modernization envisioning. To achieve the full characteristics of optimized SaaS system, Infopub needed to reach the level 4 (virtualization level) to achieve an optimized multi-tenant environment and scalability based on business capability with the support of cloud computing and service orientation. Furthermore by considering the characteristics of SaaS, the business dimension needs to be evolved into level 7 (dynamically re-configurable services) to enable the business capability-driven service development with on-demand SLA deployed into cloud. The roadmap showed the need to go from level 2 of SaaS maturity model into level 4, supported with service integration evolution in business dimension from level 2 into level 7. The detailed evolution roadmap is described in Appendix B, showing the recommendations from the current into target maturity level for each dimension. The overall suggestion gained from the roadmap is by introducing business process and service to structure the functionality of Infopub with deployment support with virtualization and cloud computing. The developed architecture needs to support service orientation with an extension to virtualization and dynamically reconfigurable services using the cloud services provided by SOA middleware components and PaaS providers.

4.2.2 Situational Analysis

Situational analysis aims to check the conditions that can affect the organization to execute the Means and achieve the Ends. As the modernization is a critical process, the causes of the changes need to be analyzed to set the best strategy for the organization. The analysis consists of external analysis, internal analysis, and SWOT analysis. The analysis was performed based on the condition of the modernization project of OPT from Infopub into public transport document generator as a service.

The first analysis, the external analysis, aims to assess the influence from the external factors. The external environment of OPT comes from the macro environment and micro

environment. The analysis would result critical success factors that the company could apply to get the best advantages. The external analysis was performed using PESTEL framework for macro environment, competitive forces framework for micro environment, and critical success factor analysis, with the detail further described in Appendix C.

PESTEL analysis is used to analyze the macro environment factors related to the company and specifically to the Infopub service. Macro environment refers to the broader society which affect the industry and the organizations within it. The factors consist of political, economic, social, technological, environmental, and legal aspects which come from the market of Portugal, European Union, and internet. As those factors could affect business strategy, PESTEL analysis is needed to assess how the factors influence business performance to ease up the decision making process. The analysis showed that the deployment of Infopub system using virtualization, SOA, and cloud computing will greatly increase the business opportunities of OPT since the current trend of industry is related with internet which increase the service delivery effectivity and efficiency. However, the SaaS providers still need to concern with the political and legal issues because the policies and legislations regarding security (sensitive data), ownership and location of data, confidentiality, and intellectual property are still not well defined and integrated throughout the world.

The competitive force analysis was used to check the influence from the industry environment. The competitive force framework was used to determine the OPT's profit potential regarding Infopub service from the competition between firms in the market in the terms of threat of new entrants, power of suppliers, power of buyers, product/service substitutes, and intensity of rivalry. The industry related to Infopub service was identified as outsource consultancy for providing the generation engine for public transport document generation service. The example of the identified competitors consisted of:

1. The IT department of public transport companies.
2. Tecmic⁵, which is a similar consultancy company based in Leiria.
3. FWT⁶, which is OPT's partner company based in UK that provides planning tools development for producing maps and diagrams for public.
4. CHK America⁷, which provides customer information solution for public transit in USA.
5. TransitEditor⁸, which provides transportation data management tools for delivering information to public based in Spain.

The competitive force analysis for Infopub system in SaaS model showed that the competition in industry of transport information for public is very challenging. Even though becoming the leader in Portuguese market, OPT needs to expand the offered services using the latest technologies to increase business opportunities. It would be easy for OPT to enter the SaaS business model because of the economies of scale, low switching cost, and low product differentiation. However, it would be a tough challenge because the competition is very high with medium supplier power and high level of threat from rivalry, product/service substitutes,

⁵ <http://www.tecmic.pt/en/>

⁶ <http://www.fwt.co.uk/>

⁷ <http://chkamerica.com/>

⁸ <http://www.transiteditor.com/>

and buyer power. The SaaS-based industry is an unattractive industry since SaaS model becomes a mandatory strategy for IT-based company to survive. OPT would be the first company in Portugal who applies SaaS model for public transport document generator, but will compete with the already exist companies in the international market since the services can be easily accessed through internet. Therefore, OPT needs to join in the competition by applying the SaaS model with unique strategy definition with focus on customer/buyer experience to gain competitive advantage and intensify the rivalry.

The key success factors of Infopub system rise from the ability to keep up-to-date with standardization and request from customers. OPT needs to develop the standard document template and transport network metadata which can satisfy the customer needs in international market. Besides, the SaaS platform and application also need to be updated with the new features which can engage new customers, and are implemented with the latest technology.

While within internal analysis, OPT has enough resources which leads to enterprise capability. In finance, OPT get the research funding from development project both in academic area and governmental development program, such as European Union projects, while acquire new customer and achieve customer retention through its product-service system. OPT also develop its own server in-house which is important to store critical data and developed with the latest technology implementation from its partnership with academics. OPT also hires talented human resource in information system and technology which leads to the environment of innovation within the company. Using the resources, OPT performed its business mainly in consultancy in transportation planning and public information with support of technology and innovation. From these conditions, OPT promotes its main capability and core competence in expert human resources in information system and technology and also transportation consultation which leads to excellence in its services.

Furthermore, the SWOT analysis can be derived from the situational analysis as can be seen in Table 3. OPT needs to make use of its strengths in innovation combined with the opportunity of internet and service technology, while still needs to improve its services as company's assets and develop new values to create a competitive advantage in the bigger market.

Table 3. SWOT Analysis of OPT and Infopub

<p><u>Strengths:</u> - OPT promotes its excellence resources in human resource and technology, promoting innovation of its product and service delivery.</p>	<p><u>Weaknesses:</u> - OPT's mindset is still focus in product-oriented, while there is still no standard documentation and development process within the company.</p>
<p><u>Opportunities:</u> - The closeness with technology in its business create an opportunity of continuous innovation of product and services, which can grow in a bigger market with the help of internet and service technology.</p>	<p><u>Threats:</u> - Even though becoming a leader in Portugal for its business, there is a tough competitions in a bigger market since there are several companies with the same industry already execute their business in different region.</p>

4.2.3 Strategy Plan

Based on the defined business architecture, evolution roadmap, and situational analysis, the strategy plan can be identified for the enterprise ecosystem. The strategy was defined in three level of managements:

1. Corporate level: OPT needs to extend the Infopub system using the SaaS business model by aiming global internalization, providing lower cost solution throughout internet, and outsourcing, using cloud services for service components and hosting.
2. Business level: OPT needs to apply cost leadership strategy by providing standard SaaS-based generation platform with competitive price, added with differentiation for particular customers per request. The platform should be able to use network automation of labor-intensive generation services and business process with economies of scales from customers' aggregation and network effects of customer co-creation.
3. Operational level: The Infopub system needs to be evolved into SaaS model to add new values and achieve a flexible business model using virtualization, SOA, and cloud computing technology. The modernization of the system was performed by wrapping the existing components of Infopub system into Web Service and developing the integrated and service-oriented SaaS application which promotes service reusability and interoperability.

4.3 Project Initiation and Planning

The modernization project aimed to satisfy the needs defined in the business goals and strategies. The need for modernization was identified because the Infopub system was still deployed in desktop environment which caused a limitation of service delivery with high cost of maintenance and configuration. With the situation of industry competition, OPT envisioned the need to modernize the Infopub system into SaaS model using the latest supporting technology to achieve high accessibility, cost reduction, and business agility. Therefore, the scope of the project is to evolve the legacy Infopub system into public transport document generator as a service using the best practices describes in the Service Oriented Modernization Framework (SOMF). The project plan was to performed each phases of the framework to develop the requirement and design specification of the target system which could be further implemented into the full system. Thus, the discussion of the phase execution was limited until the development of service realization into service technology and also SaaS system prototype, with planning description for service implementation and deployment, and SaaS system implementation and operation. Furthermore, the tools used for supporting the project development consist of:

1. MS Office, for development of reports and presentations used in the project.
2. Modelling tools, which consist of:
 - a. MS Visio, for general diagram and model development.
 - b. The trial version of Objectiver 3.0.0, for development of KAOS diagram for goal-service modeling.
 - c. IBM's Rational Software Architect (RSA) for WebSphere Software Version 9.1.2 with license from IBM Academic Initiative Membership, for development of business process model in BPMN, UML models (use case, class, component, deployment), service model in SoaML, and generation of Web Service from SoaML into XSD, WSDL, and BPEL.

4.4 Stakeholder Requirements

The requirements describe the list of conditions that need to be have by the system. The requirements were elicited from different sources as seen in Table 4. The main elicited stakeholders were only from OPT because of the project and language limitation, which made the project to be more focused on requirements from the developer side, not the end users. Furthermore, the modernization requirements were compiled and specified as can be seen in Appendix D.

Table 4. List of Sources of Requirements

No.	Source	Description	Priority
1.	Stakeholders	People involved in the project and system which consist of the thesis author, the OPT’s director, and two Infopub developers from OPT.	High
2.	Infopub documentations	Documents related to Infopub system which mainly consist of Infopub’s architecture description, the envisioning of Infopub in web, user experience in Infopub and Move-me system, and the modules manual document of Data Manager and BusMap.	High
3.	Infopub system	The existing deployed Infopub system with dummy database for system usage exploration and experience.	High
4.	Domain literature study	Study from wide-range articles about state of the art regarding enterprise architecture, SOA, and SaaS, and also service of transport information for public.	Medium

4.5 Service-Oriented Reverse Software Engineering

The Infopub system consists of three main elements: data manager, generation engine, and document, as can be seen in Figure 14. The main component lies on the data manager information system with an integrated database to store network data, schedule, and graphics (routes, paths, stops, landmarks, zoning, colors, and pictograms). The database is developed with Oracle database which is also connected with clients’ database and can be accessed to the information system using ORM (Object Relational Mapping) framework. The database schema, as analyzed by Meneses (2009), can be seen in Figure E-1. The information system is then connected to graphic generation engine, as second component, for rapid development of configurable document templates generation. The engine will automatically generate the documents, which is the third component, to be put as displays in the stops or clients’ website. The example of the interface and generation result is shown in Appendix F.

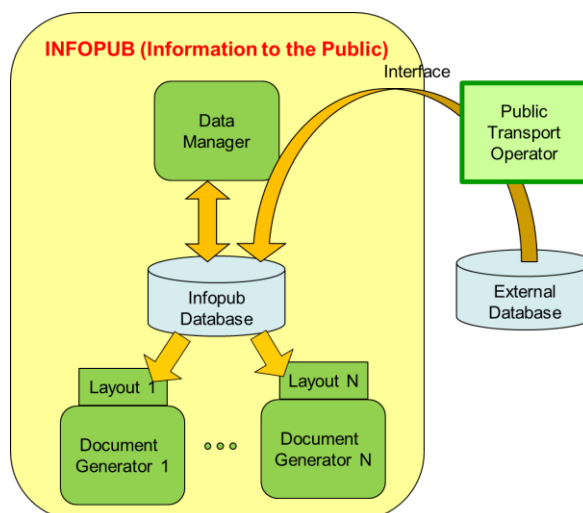


Figure 14. Infopub System Architecture

The first component, the data manager, serves to gather all of the necessary information to generate the documents by importing external data sources, connected with public transport companies, and then synchronize with the existing data. It allows configuration of network data, schedules, and other graphical elements, and also allows to export data to GTFS format (General Transit Feed Specification), allowing quick posting of the network and schedules of transportation on Google Maps. It also exports and synchronizes the data on other services, such as Move-me and GIST Light, to maintain the timeliness of the documents, including routes and stops schedule, and export them to other system.

Another component, the document generator engine, is used to produce a document type using a wizard type mechanism. It was developed on a graphical engine which allowed layout customization with several styles. The resulting files are in SVG (Scalable Vector Graphics) format, which allows to be edited with format supported application, such as Inkscape, and then synchronized with the client's website. The configuration regarding the generation format, such as layout and style sheet, was already defined earlier based on the company client request and stored in external file of the Infopub system on the customer side. Currently there are three types of document generator which have been developed: BusMap (the schematic map generator), BusSched (the schedule generator), and MultiDoc which combines both BusMap and BusSched.

BusMap is a module to generate schematic maps which show the travel possibilities related to a stop. It allows to generate documents for a stop or in batches of several stops. The representation of various routes preserves its geographic orientation, containing the information of the transfer connections and average travel times.

Similar with BusMap, BusSched is also a module to generate the document which shows the schedule of a particular transport vehicles stopping by each particular stop in a particular route for a particular period of time of the year. These schedules may be accompanied with a spine which represents the line of stops with particular reference to some emphasis and connection to other means of transport. One variation of BusSched is MultiSched which generate the schedule of several type of time into a document.

As the combination of BusMap and BusSched, MultiDoc generator produces an integrated composition of various types of documents which are already generated. It analyzes all of the generated documents in a location whose names follow a standard and assembles them in other documents. They are usually posted in stops and may contain schematic maps, schedules of several routes, and related fixed pictures of the selected stop.

As desktop-based application, Infopub is deployed in the customer side with the network connection to OPT's servers. The deployed package consists of the Infopub system modules, including data manager, BusMap, and BusSched as different executed files. Customer can use the application based on the use case diagram shown in Figure E-2. With data manager, customers can input the network and operator data with the option of importing of network data from their own database. Besides, they also can export the network data to other format and other system, and set the parameters related to the data management. Using the document generator, both BusSched and BusMap, customers can generate the document based on the defined document template configuration, layout, and style sheet, and the identified generation parameters.

The current Infopub system is only the software product which provides the generation service. However, the support services of Infopub is still being managed manually by contacting the company, thus creating a dedicated-personalized service agreement. For

service provision, the customer needs to contact OPT to create a service agreement, purchase the service, implement the Infopub system in the customer's system, and the service support itself. This condition will be difficult if the target customer segment is extended in international range since OPT will manage a bigger size of market. Therefore, the process of customer relation needs to be performed automatically by providing every information and interaction needed by customers in the platform. The interaction would be one-way, from customer into the platform, while OPT only needs to focus on maintenance and innovation of the document generator system, providing the best services to customer with every needed insight. Therefore, the components of service provisioning, customer relation, and support also need to be put in the platform since the focus of SaaS implementation is providing all of the needed services for customers, including the business services and management services.

4.6 Service-Oriented Forward Software Engineering

Based on the previous phases, the requirements were then developed into service based on several consideration. The service development process is structured as: service identification, service specification, service realization, service implementation, and service deployment.

4.6.1 Service Identification

The first step in this phase, service identification, was perform to identify the candidate services which are required for the target system development. The candidate services were identified from goal-service modeling, domain decomposition from business process, legacy Infopub system, and the needed SaaS components. The detailed artifacts of this phase are presented in Appendix G.

The goal-service model of Infopub in SaaS can be seen in Figure G-1. The super goal is to achieve the efficient public transport document generation with cost reduction, high accessibility, and business agility. The subgoal is to provide easy and effective public transport document generation platform which can be performed with Infopub's existing components: data manager and document generator. Then, the goal is extended with low cost solution of the target system by using service-orientation which promotes flexibility and loosely coupled service architecture which can be easily maintained, delivered by cloud service providers. The target system also needs to be highly accessible through cloud and SaaS, promoting flexibility, scalability, and multi-tenant by deploying in web and mobile environment under PaaS provider. Finally, the business agility also can be achieved by implementing new business model with on-demand pricing model and market and offered value extension to acquire new customer and achieve customer retention.

Under domain decomposition, the business aspect of Infopub was identified with MSD based on the customer experience. The starting point was from customer journey described in Figure 15. The customers of Infopub, which are transport companies and operators, use the Infopub service from the need of public information updating. After making contract and SLA, the customers can use the Infopub system from inputting data, setup the template customization, and generate the document. Then they can review the generated documents and publish into bus stops and/or the company's website. The customer journey was then detailed further into service blueprint which can be seen in Figure G-2. By using the RSA, the candidate services can be generated from the business process blueprint. The identified candidate services from this blueprints consist of Instruction Processing, Infopub, CRM, and Platform Management.

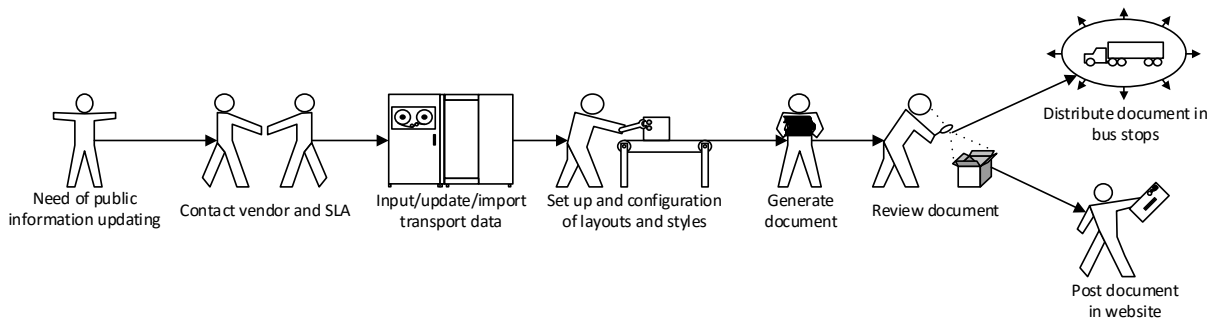


Figure 15. Customer Journey of Public Transport Document Generator

Furthermore, the candidate services was also identified in the legacy Infopub system analysis and the SaaS components analysis. The candidate services from the legacy system were derived from the Infopub’s components which consist of data manager and document generator (BusMap and BusSched). The candidate services were also added with the components defined in the reference architecture as required components for SaaS. Most of the required services for an SaaS application can be provided by the PaaS provider, such as infrastructure, virtualization, data storage, integration, service management, and security. Therefore, the services mentioned in the modeling process are the ones which are relevant to the identified customer journey.

Then, the candidate services were identified from the four approaches. The candidate services mainly came from the Infopub’s services of data management, template customization, and document generation. Moreover, there was also CRM services which handle sales (SLA) and customer service. Besides, there was also platform management services from PaaS provider which was able to list the available services, create multitenant database account, process the payment, authenticate and authorize the account, monitor the service usage, and process the service configuration. Furthermore, instruction processing service was also added to act as intermediate service which connect to other services, thus creating interoperability. Finally, the candidate services were refactored for naming convention of the process operation as shown in Figure 16.

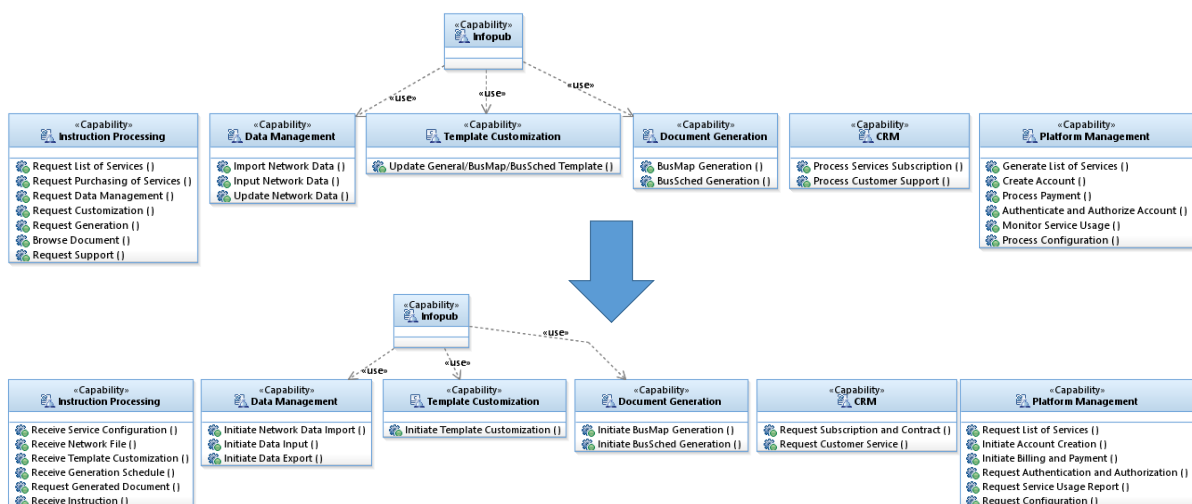


Figure 16. The Candidate Services of Public Transport Document Generator

4.6.2 Service Specification

In specification phase, the identified candidate services were exposed as services, as shown in Figure 17. Each capability represents each service, except for the platform management service which was exposed into payment, user management, and service management services. Furthermore, as services communicate to each other through messaging protocol, the message structure used for service communication was also specified. Moreover, Figure 18 illustrates the message structure which was used for generation process. Each messages and data types described in the diagram represent the parameter for the service interfaces. The generation message consisted of operator’s profile data, network data (the network structure described in Figure E-1), and template data. The generation process would be executed based on specific generation schedule of the provider and would be charged, which depended on the chosen contract, and generated into target document. However, since the research focused to demonstrate the service development process, the robust service data analysis and modeling was outside of the research scope. The message diagram was developed to illustrate the message used by the service interface for completeness purpose since it is a critical part of the service development process.

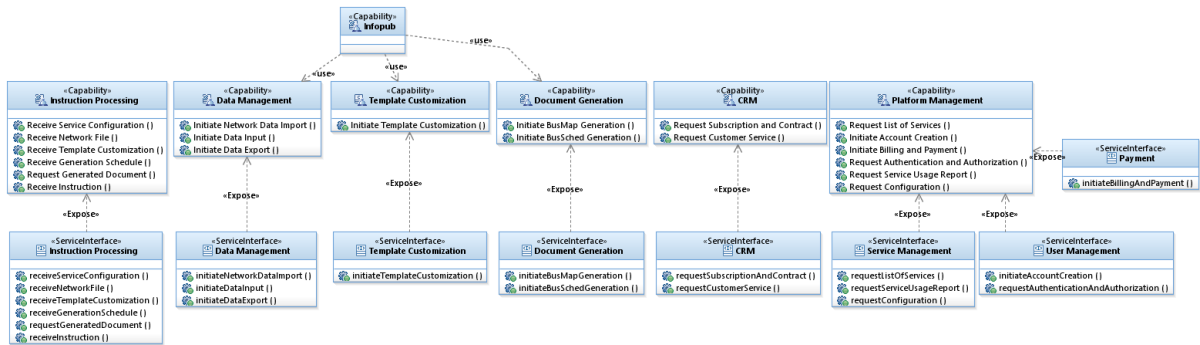


Figure 17. Exposed Service Interfaces of Public Transport Document Generator

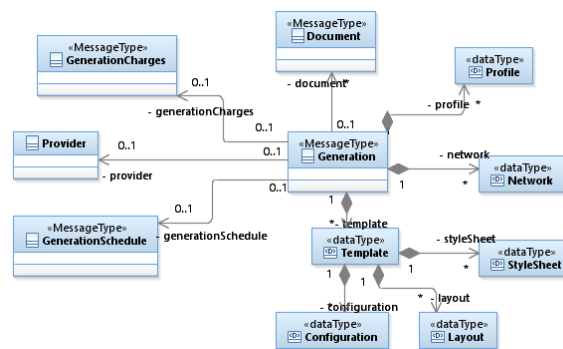


Figure 18. Message Data Structure of Generation Process

The exposed service interfaces then were composed into general service architecture model as shown in Figure 19. The architecture (on the left) illustrate the connection of service consumption and provision between participants through service contract. The identified participants consisted of customer, instruction handler using ESB component, components from Infopub (data manager and document generator), CRM for creating customer relation and loyalty, PaaS provider for deploying the application, and bank for payment service. Moreover, the service contract was detailed further into sequence diagram. Figure G-5 illustrate the sequence diagram of the service contract for document generation process, from inputting the operator data into data manager, customizing the document template, and

generate the document. Furthermore, another architecture (on the right) illustrate the structure of the generation process from the defined databases which was designed by OPT.

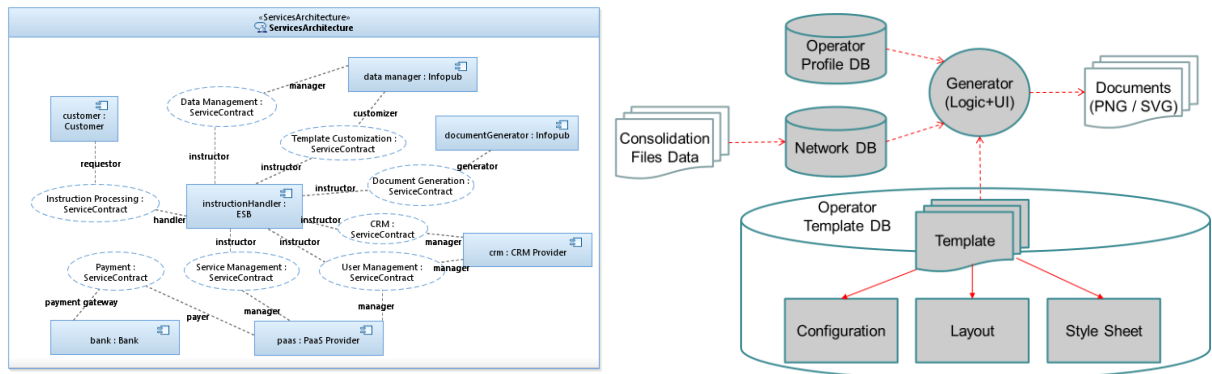


Figure 19. Service Architecture of Public Transport Document Generator

4.6.3 Service Realization

In this phase, the services were realized into detailed design specification. The purpose was to realize the service components which were required for the system development. The service components were represented by the identified participants, as shown in Figure G-6. Each participants had service and request ports which represents the service interfaces to interact between the services through service channel. The participants were composed by connecting each services ports, as shown in Figure 20. The participants structure could also be represented into Service Component Architecture (SCA) which could be further developed into Web Services. Furthermore, the service architecture was composed based on the defined reference architecture, shown in Figure G-7, to illustrate the needed components in service orientation in each building block. The software architecture layer was presented with the developed service architecture.

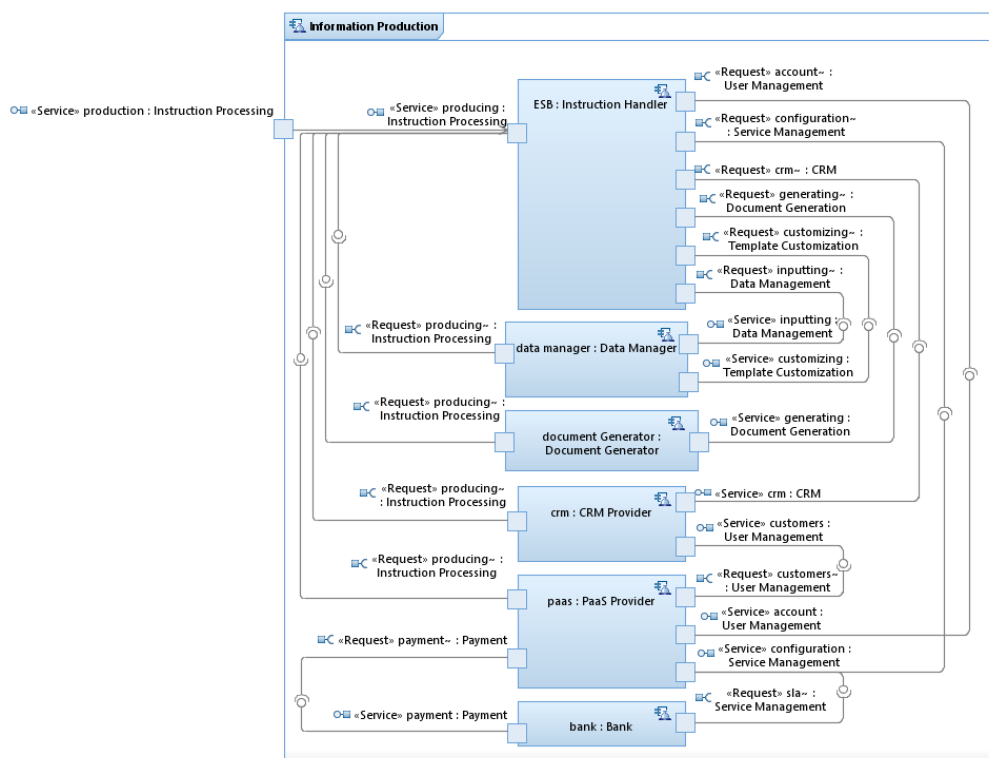


Figure 20. The Completed Information Production Structure in Participant Diagram

4.6.4 Service Implementation and Deployment

After designing the service architecture, the application can be implemented into Web Service and the IBM SOA programming model. RSA provides model transformation tool to generate the Web Service implementation using the UML-to-SOA transformation which is based on Service Component Architecture (SCA) and uses Service Component Definition Language (SCDL) as metadata definition. Using this transformation, the service architecture model from SoaML could be generated into Web Service implementation, including business objects (XSD), interfaces (WSDL), module assemblies (SDLC, an IBM predecessor of Open SCA or “Classic SCA”), and processes (BPEL4WS). The XSD could be generated from the service message diagram, while the WSDL could be generated from service interface, and BPEL4WS could be generated from the BPMN model of business process or activity diagram.

In the context of public transport document generator, the service architecture diagram was generated into Web Service implementation using the UML-to-SOA transformation. The architecture illustrated in Figure 20 was generated into Web Service as illustrated in Figure 21. The generation produced the WSDL files from the identified service interfaces and the components definition (SCA) files from the specified participants. The generated Web Services then could be implemented more detailed by developing the implementation codes, which associated the service and component interface with the access to the Infopub’s API and the other required outsourced components. In short, this phase could show the generation feature of model-driven development (MDD) for service oriented architecture (SOA) which could facilitate the implementation phase, supported by the available tools.

<ul style="list-style-type: none"> ▼ Web Service Generation <ul style="list-style-type: none"> ▼ PTDG <ul style="list-style-type: none"> > META-INF ▼ Specification <ul style="list-style-type: none"> CRM.wSDL DataManagement.wSDL DocumentGeneration.wSDL InstructionProcessing.wSDL Payment.wSDL ServiceManagement.wSDL TemplateCustomization.wSDL UserManagement.wSDL > Realization.Bank > Realization.CRMProvider > Realization.DataManager ▼ Realization.DocumentGenerator <ul style="list-style-type: none"> > META-INF <ul style="list-style-type: none"> DocumentGenerator.component generatingExport.export producingExport.export producingImport.import sca.module > Realization.InstructionHandler > Realization.PaaSProvider 	<pre> <?xml version="1.0" encoding="UTF-8"?><wso:definitions xmlns:wso="http://schemas.xmlsoap.org/wsdl/" name="DocumentGeneration" targetNamespace="http://Specification/DocumentGeneration/" xmlns:tns="http://Specification/DocumentGeneration/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <wso:types> <xsd:schema targetNamespace="http://Specification/DocumentGeneration/"> <xsd:element name="initiateBusMapGeneration"> <xsd:complexType> <xsd:sequence/> </xsd:complexType> </xsd:element> <xsd:element name="initiateBusSchedGeneration"> <xsd:complexType> <xsd:sequence/> </xsd:complexType> </xsd:element> </xsd:schema> </wso:types> <wso:message name="initiateBusMapGenerationRequest"> <wso:part element="tns:initiateBusMapGeneration" name="initiateBusMapGenerationParameters"/> </wso:message> <wso:message name="initiateBusSchedGenerationRequest"> <wso:part element="tns:initiateBusSchedGeneration" name="initiateBusSchedGenerationParameters"/> </wso:message> <wso:portType name="DocumentGeneration"> <wso:operation name="initiateBusMapGeneration"> <wso:input message="tns:initiateBusMapGenerationRequest"/> </wso:operation> <wso:operation name="initiateBusSchedGeneration"> <wso:input message="tns:initiateBusSchedGenerationRequest"/> </wso:operation> </wso:portType> </wso:definitions> </pre>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Figure 21. Generated Web Services and WSDL example of Document Generator Interface

Finally, the implemented Web Services could be deployed in a web server, which could be decided using public, private, or hybrid cloud. In case of the Infopub system, the developed Web Services could be deployed in the private server which had already been developed by OPT. This application promoted more security of the code implementation of the Infopub system, as the efficiency of the software processing could affect the competitive advantage of the similar products. Moreover, OPT could use the outsourced resources from the public cloud server to increase the software processing. Furthermore, OPT also needed to concern on the registry of the outsourced services to be able to use the services into SaaS application. As a result, the SaaS application could then be developed based on the defined Web Services.

5 SaaS Application Development

This chapter is the continuation of the case study described in Chapter 4. While the previous chapter focuses on the service development, this chapter focuses on the application development in SaaS application. As part of the UCSOSR methodology, the development is discussed from prototype design and development, cloud platform evaluation for choosing the deployment plan, and plan for platform implementation and operation.

5.1 Prototype Development

Prototype illustrates how the user can access the service system using the identified user interface. Prototype serves to evaluate the interaction between user and system to improve the experience and usability. Prototype can be developed in many ways depend on the complexity level that the developer want to show. For continuous and early phase of agile development, low fidelity prototype, such as mock up and storyboard, can be used since it uses low cost and less resource. Therefore, the focus of the development can be put on higher level, such as service concept, and develop the detailed prototype which can be continued into implementation after the usability requirements are already well defined. Additionally, high fidelity prototype also can be used to show how the system works in the real application form using the designated technology. The high fidelity prototype is developed like the real application to show the application interaction in the detail way with a higher degree of resources needed.

There were two prototype that were developed for Infopub in web project as shown in Figure 22 with a detailed ones in Appendix H. The low fidelity prototype was developed in a mock up design and was presented from the presentation proposal of Luís Filipe Ferreira to develop the Infopub system in Web. Moreover, the high fidelity prototype of Infopub in Web was also presented in the dissertation of Meneses (2009) using the Microsoft environment technology, such as MS Silverlight and WCF, which was in line with the technology used by OPT. Furthermore, the mock up and the prototype could be evaluated on several users using usability testing for further improvement on customer experience.

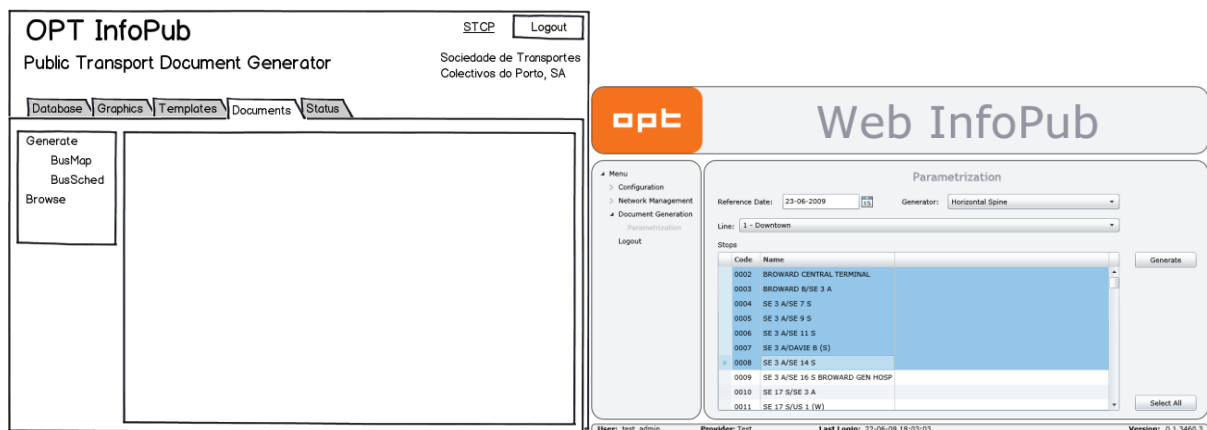


Figure 22. Prototype of Public Transport Document Generator

5.2 Cloud Platform Evaluation

After designing the prototype, the cloud platform providers need to be evaluated to decide the development and deployment strategy. For SaaS application development, the software application needs to be deployed in a network infrastructure to be accessed widely. There are three choices of deployment: public cloud, private cloud, and hybrid cloud. Public cloud refers to the outsource cloud infrastructure provider which can be subscribed to deploy the application. Private cloud refers to deployment in-house, by using the owned server. Hybrid cloud combines both public and private deployment to get the maximum benefit from the two models (Cloudyn 2014).

OPT could deploy the public transport document generator as service system using a public cloud or hybrid cloud. By using public cloud, OPT could just focus on the service development and improvement, without concern on the infrastructure and platform system. However, this would raise an issue on security and control since the application system would be hosted in external entity. Since there is already a server infrastructure as resource, OPT could also develop the SaaS application using hybrid cloud. One example of the solution is to deploy the Web Service implementation of the Infopub system, which consist of the data manager and document manager (BusMap and BusSched) in the company's server. The private deployment aimed to maintain the efficient maintenance and configuration of the data manager and document generator for further improvement. It also aimed to keep the security by applying a firewall and more trusted security policy. Furthermore, the application can be deployed in outsourced PaaS provider which already provide the required services for database management system and service management. The combination of both deployment style into hybrid cloud would give benefit of security with additional resources on the go by renting computing and storage requirements. However, managing both clouds would be difficult since the security on the private cloud needs to be extended to be accessed by public cloud and there might be different protocol and policy. OPT needs a third party management tools to facilitate the control and integration between both of the clouds. Therefore, the choice between public cloud or hybrid cloud needs to be decided carefully since it would affect the business strategy of OPT in SaaS area.

OPT considered two PaaS providers to be used in for the deployment of the system in the cloud: Microsoft Azure⁹ and Google Cloud Platform¹⁰. As cloud provider, both Microsoft Azure and Google App Engine provide the standard module for cloud computing solution, such as PaaS, IaaS, analytics, computing, database, mobile, networking, storage, and web. However, there are some differences, advantages, and disadvantages between both which are described in Appendix I. Overall, OPT would be more suitable to use Microsoft Azure since OPT join the Microsoft Partner Network (MPN) and Infopub system was developed in Microsoft environment using MS Visual Studio, .NET, and Oracle RDBMS. There would be kind of promotion for member of MPN and it would be easier to adjust the Infopub's code implementation to be deployed in the public cloud. Furthermore, Microsoft Azure also support hybrid connection services which integrate the private cloud of OPT and the Azure public cloud. However by considering the pricing policy, Microsoft Azure is more expensive than Google Cloud Platform, and even its App Engine provides free package which would be useful for early operation evaluation of the SaaS application. Even by considering the return

⁹ <https://azure.microsoft.com/>

¹⁰ <https://cloud.google.com/>

of investment (ROI), the cost of using the Google Cloud Platform would be added on the time and effort cost for porting the Web Services into the available environment, and ROI would cover the expensive price after several times. Therefore, OPT should decide to use the Microsoft Azure for easy and flexible implementation of the public transport document generator.

5.3 SaaS Implementation

This phase aims to implement the designed prototype into SaaS application based on the defined cloud platform environment. First of all, the developer needs to identify the needed development tool kits and technology needed for implementation in the defined environment. In case of OPT, the development needs to be executed in Microsoft environment, thus the developer needs to prepare the development tool kits such as MS Visual Studio and MS Silverlight for web application development. The SaaS application is implemented using the Web Services as main components. The Web Services serve as control components which will be put for the specific part of the user interfaces.

The developer firstly develop the graphical user interface of the application in web-based technology. The development in web-based presentation layer aims to generate multiplatform interfaces so that it can be deployed in several platforms, such as web, desktop, and mobile, with the specific configuration for the specific platform. Service-orientation enables multichannel delivery through the possibility of multiplatform. Services take an important role as mediator since with the use of middle logic layer, all of the logic processing doesn't need to be done in the client side, but just need to be put in the server side, thus decreasing the access load. As complementary with the multiplatform delivery, the web-based presentation layer can be used as it can generate the similar multiplatform interface with specific configuration. The client side only needs to show the contents and the process result, leaving the functional processing within logic components and data processing within data stores. Thus, it enables the expansion of the platform and to keep the client's behavior consistent and up-to-date. The local data is only for cache or temporary data so that the processing within presentation layer is not slowed down. One example of this implementation is to use Chromium framework which allows to develop a browser-based application in web technology, thus the content of the browser can be generated from the Web Services and the multiplatform could be achieved.

Furthermore, the implementation could be extended to create a community for co-creation. The services development needed to be well documented to provide the information about the API for clients to integrate within their server. Therefore, the service ecosystem could be developed, not just the application, but also the co-creation system between customers by creating the community.

After developing the application, then the system is migrated into the targeted server with change management, including the database and application. Pre-migration needs to be concerned to avoid extra costs and delays. Developers need to check the inventory of all assets to be migrated and the SLA between the company and the service providers to keep everything under control and avoid vendor lock-in from the defined exit strategy. Backup plan also needs to be created in case of failed or delayed migration. Furthermore, developers also need to understand the user interaction so that migration to the cloud would help business cuts the cost and increase users' expectations. Finally the dependencies need to be identified to

make sure that the migration does not interfere with the business activities by using live migration and cease the business activity during the migration process to ensure the process.

After preparing the pre-migration process, then the system is ready to be migrated into cloud services. The best practice to perform the migration is to run the test migration with a re-architected application to ensure that it runs properly and not to upgrade during migration process to prevent an error. Then data could be moved incrementally to analyze the advantages to the cloud before becoming completely committed.

5.4 SaaS Operation

After system has been developed and the deployment plan has been identified, the system then be deployed and operator firstly to get insight about the cloud operation. Evaluation is needed to see how the developed application is working in the cloud platform. The evaluation and operation can be planned using alpha and beta testing. Firstly using alpha testing, the SaaS application is tested by several identified users as test drive to check the first feedbacks. Then, the SaaS application is launched with the status of beta testing to check the feedback from real users. The beta testing can be performed through users' invitation using the free modules of the generation application. The application can be released as a preview version for a period of time until it is improved into the final version. The insights can be acquired from evaluation, subscription, and the business process of the application usages to be improved continuously. This way, the system can be verified based on customers' needs and preferences, thus creating a more reliable and trustworthy application ecosystem. After making sure that the business process can be well performed, the application can be launched and operated with continuous monitoring, performance evaluation, and tuning to keep serving the best product and services to customers.

5.5 Summary

The last two chapters describes the demonstration of the defined framework into a case study of public transport document generator as a service. The demonstration was performed by following the User-Centered Service-Oriented Software Reengineering methodology from envisioning of the transformation project until the SaaS application development. The business envision of the system modernization has been developed as strategy for evolving the legacy system. The requirements and the legacy system were analyzed, along with the goal-service model, domain specification, and SaaS components analysis to identify the candidate services for development. The services then were specified and realized into service architecture with participants as representation of service components, which then were generated into Web Service implementation. Finally the SaaS application development plan has been presented with prototype suggestion from the existing models and the SaaS development and migration plan to be deployed into cloud services. Several artifacts were developed using the proposed methods used in the framework. The demonstration resulted a system design specification in the form of service architecture and further implementation and deployment plan and proposal. Even though the service architecture has been designed, the development is still limited as the implementation phase only covered the generation of service architecture into Web Services, without further code and system implementation. Moreover, the identified services was defined as the primary components which can be used in the SaaS application development. Furthermore, the result of the demonstration and how it relates to the proposed framework is discussed in the next chapters.

6 Evaluation and Discussion

This chapter evaluates the demonstration that has been executed and relates it with the literatures' theories developed in the proposed framework. The evaluation aims to find a lesson learned from this research aligning with the topic and the defined research questions. The framework evaluation was performed by using verification and validation based on the legacy evolution evaluation criteria (Khadka et al. 2012) and the SaaS quality model (Pang Xiong and Li 2013). However, the SaaS quality model analysis could not be performed since the described quality depends on the implementation decision of the SaaS system. Then, the case study was also evaluated based on the conformity with the framework. Finally, the research result would be discussed from both the framework and case study development.

The framework evaluation based on the identified criteria is presented in Appendix J. The evaluation shows that the proposed framework covers most of the criteria, except for the detail on implementation and deployment. The framework provides the best practices of the services development from legacy system understanding, target system understanding, modernization feasibility determination, candidate service identification, implementation until the service realization, and also the SaaS system development process. The full implementation and deployment were not described completely using the implementation and deployment technique, instead the proposal and the recommendation were presented. Therefore, there are still rooms of improvement for providing a comprehensive framework.

Similarly, the case study was also evaluated based on the execution of each phases of the methodology. The case study was well executed for all of the phases except for the implementation and deployment process. The final service architecture was designed in service realization phase and could be used as design specification for service implementation. The service architecture then could be generated into Web Service implementation to ease the implementation process. The utilization of model in service and software development is very useful since model provides the same perception for both human (analyst) and machine. However, the drawback from model development is that model sometimes is complicated to be designed and developed. The analyst needs to carefully follow the modeling best practice to develop a verified and validated model so that it can be further processed. Therefore, the model can be verified and validated to follow each steps of the generation process.

Furthermore with the implementation, the generated Web Services then could be developed to be used as components for the SaaS system. OPT could develop the Web Services using web-based technology mentioned by Meneses (2009), such as .NET framework and Windows Communication Framework (WCF), which were in line with the development environment used by OPT. The implementation could be executed by developing the WSDL to describe the service interface as a wrapper of the Infopub services. Based on the service architecture, the Infopub services could be wrapped into two interfaces: the data manager and document manager. Since service interfaces acted as mediators or façade component for other system to access the provided services, the internal code of the Infopub system would not need to be change a lot. Furthermore, the other components, such as CRM service, payment service, integration hub service, and PaaS, could be outsourced from external partners with defined service contracts and subscriptions. As outsourced cloud services promoted the payment based on the service usage and were available through internet connection, the needed

implementation strategy was to integrate the identified services through their interfaces. With the current available technology, the service implementation should be able to apply the service characteristics described in the level 7 of the OSIMM. Therefore, the service orientation implementation promoted interoperability and loose couple between its elements

Moreover, the user centered part also could not be performed since the project was performed only in the scope of OPT. With user-centered design, the usability could be enriched by performing the ethnographical study to real customers, such as public transport companies and operators, to reiterate the development of the prototype and perform the usability testing to evaluate the prototype. With user-centered design, the developed application could have a rich user interaction, and integrated with service design, it would promote user engagement and retention.

Despite of the scope limitation, the framework provides a structured guideline which integrates the business aspect into technical aspects. Since service development is a business oriented process, the case study showed that the business strategy and services could be identified and extended into implementation of services. The service components represent the business services and technical services which together serves the purpose of the system. Moreover, the integration with several approaches could enrich the development process with a wider point of view. The development of SaaS and SOA is different than the usual software development, since they concern to develop a service ecosystem which includes business model and the strategy to deliver the services to customer. Therefore, the business envisioning and goal-service modeling hold an important role for the modernization process to bring the maximum competences in delivering of business.

7 Conclusions and Future Works

This research proposed a framework for modernization process from legacy system into service-oriented SaaS system. The framework integrated several disciplinary approaches such as service engineering both in business and technical aspects, software engineering, reverse engineering, user-centered design, and cloud computing. The framework was then demonstrated in a case study of development of public transport document generator as modernization process of the Infopub system from OPT. The project aimed to evolve the Infopub system into SaaS model using the service orientation concept which resulted a service oriented architecture of the system with implementation and development plan of the Web Services and SaaS system.

Triggered by the identified research questions, this project develops several key points. First of all, the combination of service design and SOA develops an alignment of an integrated enterprise, from the business aspects and the technical aspects. Service design serves as domain analysis method which focus on customer experience to perform user-centered study using ethnographical methods, thus identifying new services and value propositions for the company from the perspective of the customer. Moreover, the SOA serves to implement the identified services into software system with a reusable, flexible, and loosely-coupled architecture which allows separation of concern between business and technical implementation. Therefore, the combination of both results a business alignment and agility from customer experience into information system which is able to create competitive advantage and strong customer relations.

In case of modernization and development process in general, the key aspect is the envisioning of the project and the traceability throughout the execution of the project. Since service and SaaS concept involve multidisciplinary approaches, the framework needs to ensure that they project can be developed in line with the business strategy and still be maintained until the final result is produced.

The framework also can be applied in the wide-range of industrial cases as it serves as general guideline of modernization and development process in general. The application of each phases depends on the project decision and goal, until which level that the maturity want to be achieved, what technology that the system want to implement, which tools and methods that will support the best in the project development, and which type of cloud services that want to be used in system. Every considerations need to take into account of the business strategy.

Since service is the moving point of business nowadays, it holds an important role on the enterprise business. Service has change the product-oriented mindset so that the business will focus more on customer experience. The modernization serves as a trigger for business evolution to be agile, to change and adapt the service innovation based on the external and internal conditions to achieve competitive advantages. The service and modernization concept result an agile alignment between business and technology in two way around, the business improvement can enrich the technology implementation and technology improvement can extend the business model and opportunity.

For future development, there are several issues which raise to improve the quality of research in this topic. First of all, the framework can be further improved with a more thorough and deep exploration and analysis even until implementation and deployment with the support of methods, techniques, and tools to create a comprehensive guideline with the latest research

and technology development. Since model-driven development is currently growing with the advantages of using model as representation to get better understanding by human and computer, development of supporting tools needs to be continuously improved to support and provide a basic template for promoting reuse and resulting an efficient development process. The framework also needs to be more reliably evaluated by asking the employers of OPT to apply the framework in the development process to examine if the framework can be widely used. The framework also needs to be evaluated using benchmark method by comparing the framework with the similar ones to identify the advantages, disadvantages, and evaluate the application into general case study. The issues related to the methods used in the framework also need to be examined to review the benefits and drawbacks, and how to resolve them. For instance, the decision of using SaaS will create an issue in data privacy and security, thus the organization needs to be careful to choose the service vendors. Moreover, modernization process needs a concern of change management to keep the system reliable despite of the migration process since modernization is not just migration of an application, but also involving bureaucracy, administration, and management issues from the organization.

Furthermore in term of the development of public transport document generator, the application quality can be improved to enrich the system. The design architecture and generated Web Services need to be evaluated by employers of OPT using user acceptance testing to verify and validate the result to be applied in the real system. Furthermore, since the context of development in this research only limited to the design specification of service oriented architecture, the development of the service and SaaS system should be continued into full system implementation and deployment. The system operation and evaluation need to be performed to get insights and feedback from customer experience. Moreover, the application of the service orientation concept can be extended into the whole services aspect of OPT as an enterprise to show the real advantages of SOA for SaaS system. This could use the SOA Enterprise methodology with the context of business and technical aspects, since SOA acts an integrator of all of the elements in the enterprise. As a result, business agility could be achieved in a larger means and covers all of OPT's services and capabilities.

References

- Alkazemi, Basem Y., Abdullah Baz, and Grami M. Grami. 2012. "Refactoring Legacy Software Systems into SOA Compatible Style to Support e-Business Development in Enterprise Organizations: The Case of Umm Al-Qura University's Systems." *International Journal of e-Education, e-Business, e-Management and e-Learning* 2 (6):465-472. doi: 10.7763/ijeeee.2012.v2.165.
- Arsanjani, A., S. Ghosh, A. Allam, T. Abdollah, S. Ganapathy, and K. Holley. 2008. "SOMA: A method for developing service-oriented solutions." *IBM Systems Journal* 47 (3):377-396.
- Bjeljac, Petar, Branko Perišić, Igor Zečević, and Danijel Venus. 2014. "Refactoring Legacy Enterprise Information Systems to Service Oriented Architecture - The Faculty of Technical Sciences Case Study." *INFOTEH-JAHORINA* 13:869-873.
- Bratanis, Konstantinos, and Dimitrios Kourtesis. 2014. "Introducing Policy-Driven Governance and Service Level Failure Mitigation in Cloud Service Brokers: Challenges Ahead." 8377:177-191. doi: 10.1007/978-3-319-06859-6_16.
- Candan, K. Selçuk, Wen-Syan Li, Thomas Phan, and Minqi Zhou. 2009. "Frontiers in Information and Software as Services."1761-1768. doi: 10.1109/icde.2009.168.
- Chauhan, Muhammad Afeef, and Muhammad Ali Babar. 2011. "Migrating Service-Oriented System to Cloud Computing: An Experience Report."404-411. doi: 10.1109/cloud.2011.46.
- Chen, Hong-Mei. 2008. "Towards Service Engineering: Service Orientation and Business-IT Alignment." Proceedings of the 41st Annual Hawaii International Conference on System Sciences.
- Chung, Sam, Joseph Chul An, and Sergio Davalos. 2007. "Service-Oriented Software Reengineering: SoSR."172c-172c. doi: 10.1109/hicss.2007.479.
- Clouduyn. 2014. "Who moved my cloud? Part I: Introduction to Private, Public and Hybrid clouds and smooth migration." Clouduyn Accessed 28 August 2015. <http://learn.clouduyn.com/who-moved-my-cloud/>.
- Deeter, Byron, and Ryan Jung. 2013. "Software as a Service Pricing Strategies." Accessed February 1st, 2015. <https://bvp.box.com/shared/static/05d7zb2zi64q7rbv1opl.pdf>.
- Erl, Thomas. 2005. *Service-Oriented Architecture: Concepts, Technology, and Design*. NJ, USA: Prentice Hall PTR.
- Hagins, Jeff. 2008. "Making the SaaS Decision: The Threats, Opportunities, and Challenges of Moving to SaaS." Mural Consulting Corporation Accessed 20 May 2015. http://www.softsummit.com/library/presentations/2008/jeffhagins_muralventures_makingthesaasdecision.pdf.
- Hauch, Randall M. 2005. "Using MDA to Integrate Corporate Data into an SOA." Object Management Group Accessed 4 June 2015. http://www.omg.org/news/meetings/workshops/MDA-SOA-WS_Manual/02-4_Hauch.pdf.
- IMTT. 2011. "COLEÇÃO DE BROCHURAS TÉCNICAS / TEMÁTICAS: Sistemas de Informação ao Público." Pacote de Mobilidade: Conferência em Território, Acessibilidade e Gestão de Mobilidade, March 2011.
- Ionita, Anca Daniela. 2013. "Introduction to the Migration from Legacy Applications to Service Provisioning." In *Migrating Legacy Applications: Challenges in Service Oriented Architecture and Cloud Computing Environments*, edited by A. D. Ionita, M. Litoiu and G. Lewis, 1-11. IGI Global.
- Johnson, Gerry, Kevan Scholes, and Richard Whittington. 2008. *Exploring Corporate Strategy*. 8th ed. Essex, England: Prentice Hall.

- Kang, Seungseok, Jaeseok Myung, Jongheum Yeon, Seong-wook Ha, Taehyung Cho, Ji-man Chung, and Sang-goo Lee. 2010. "A General Maturity Model and Reference Architecture for SaaS Service." *Lecture Notes in Computer Science* 5982:337-346. doi: 10.1007/978-3-642-12098-5_28.
- Kaszniak, Efrat. 2014. "How the SaaS business model affects software company valuations." *The Business Journals*, Advision of ACBJ. Accessed February 1st, 2015. <http://www.bizjournals.com/bizjournals/how-to/funding/2014/12/how-the-saas-business-model-affects-valuations.html?page=all>.
- Khadka, Ravi, Amir Saeidi, Andrei Idu, Jurriaan Hage, and Slinger Jansen. 2012. "Legacy to SOA Evolution: A Systematic Literature Review." In *Migrating Legacy Applications: Challenges in Service Oriented Architecture and Cloud Computing Environments*, edited by A. D. Ionita, M. Litoiu and G. Lewis, 40-71. IGI Global.
- Khanjani, Atieh, Wan Nurhayati Wan Ab. Rahman, and Abdul Azim Abd Ghani. 2014. "Feature-based Analysis into the Trend of Software Technologies from Traditional to Service Oriented Architecture and SaaS Cloud." *Journal of Computer Science* 10 (12):2408-2414. doi: 10.3844/jcssp.2014.2408.2414.
- Kommalapati, Hanu, and William H. Zack. 2011. "The SaaS Development Lifecycle." InfoQ. Accessed 29 May 2015. <http://www.infoq.com/articles/SaaS-Lifecycle>.
- Kourtesis, Dimitrios, Iraklis Paraskakis, and Anthony J.H. Simons. 2012. "Policy-driven governance in cloud application platforms: an ontology-based approach." Proc. 4th. Int. Workshop on Ontology-Driven Information Systems Engineering.
- Lapouchnian, Alexei. 2005. "Goal-Oriented Requirements Engineering: An Overview of the Current Research." University Of Toronto, Last Modified 28 June 2005 Accessed 3 June 2015. <http://www.cs.utoronto.ca/~alexei/pub/Lapouchnian-Depth.pdf>.
- Letaifa, Asma Ben, Amel Haji, Maha Jebalia, and Sami Tabbane. 2010. "State of the Art and Research Challenges of new services architecture technologies : Virtualization , SOA and Cloud Computing." *International Journal of Grid and Distributed Computing* 3 (4).
- Lovelock, Christopher, and Jochen Wirtz. 2011. *Services Marketing: People, Technology, Strategy*. 7th ed. ed. NJ, USA: Prentice Hall.
- Meneses, André da Costa. 2009. "InfoPub na web." Departamento de Engenharia Informática, Faculdade de Engenharia da Universidade do Porto.
- Mohammadi, Mohsen, and Muriati Mukhtar. 2013. "A Review of SOA Modeling Approaches for Enterprise Information Systems." *Procedia Technology* 11:794-800. doi: 10.1016/j.protcy.2013.12.260.
- Nodehi, Tahereh. 2014. "A new MDA-SOA based Framework for Intercloud Interoperability." Departamento de Engenharia Electrotécnica, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa.
- OMG. 2011. "Business Process Model and Notation (BPMN) Version 2.0." [Specification]. Object Management Group, Inc. (OMG), Last Modified January 2011. <http://www.omg.org/spec/BPMN/2.0>.
- OMG. 2012. "Service oriented architecture Modeling Language (SoaML) Specification Version 1.0.1." [Specification]. Object Management Group, Inc. (OMG), Last Modified May 2012. <http://www.omg.org/spec/SoaML/1.0.1>.
- OMG. 2015. "Business Motivation Model Version 1.3." [Specification]. Object Management Group, Inc. (OMG), Last Modified May 2015. <http://www.omg.org/spec/BMM/1.3/>.
- OPT. 2015. "Empresa-OPT." Accessed 28 May 2015. <http://www.opt.pt/empresa.asp>.
- Osterwalder, Alexander, and Yves Pigneur. 2010. *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers*. Edited by Tim Clark: Self Published.

- Pang Xiong, Wen, and Dong Li. 2013. "Quality Model for Evaluating SaaS Service." Emerging Intelligent Data and Web Technologies (EIDWT), 2013 Fourth International Conference on, 9-11 Sept. 2013.
- Patricio, L., R. P. Fisk, J. Falcao e Cunha, and L. Constantine. 2011. "Multilevel Service Design: From Customer Value Constellation to Service Experience Blueprinting." *Journal of Service Research* 14 (2):180-200. doi: 10.1177/1094670511401901.
- Peppers, Ken, Tuure Tuunanen, Marcus A. Rothenberger, and Samir Chatterjee. 2008. "A Design Science Research Methodology for Information Systems Research." *Journal of Management Information Systems* 24 (3):45-78.
- PMI. 2013. *A guide to the project management body of knowledge : (PMBOK® guide)*. 5th ed. Newtown Square, Pa: Project management institute.
- Pohl, Klaus. 2010. *Requirements Engineering: Fundamentals, Principles, and Techniques*. 1st ed: Springer Publishing Company, Incorporated.
- Rausas, Matthieu Péliissié du, James Manyka, Eric Hazan, Jacques Bughin, Michael Chui, and Rémi Said. 2011. Internet matters: The net's sweeping impact on growth, jobs, and prosperity. McKinsey Global Institute.
- Respect-IT. 2007. "A KAOS Tutorial." [Tutorial]. Respect-IT, Last Modified 18 October 2007 Accessed 10 August 2015. <http://www.objectiver.com/fileadmin/download/documents/KaosTutorial.pdf>.
- Rico, Antonio, Manuel Noguera, José Luis Garrido, Kawtar Benghazi, and Joseph Barjis. 2014. "Extending multi-tenant architectures: a database model for a multi-target support in SaaS applications." *Enterprise Information Systems*:1-22. doi: 10.1080/17517575.2014.947636.
- Saini, Arnita, Benjamin Nanchen, and Florian Evequoz. 2013. "Putting the Customer Back in the Center of SOA with Service Design and User-Centered Design." 8135:94-103. doi: 10.1007/978-3-642-40651-5_8.
- Seinturier, Lionel, Philippe Merle, Romain Rouvoy, Daniel Romero, Valerio Schiavoni, and Jean-Bernard Stefani. 2012. "A component-based middleware platform for reconfigurable service-oriented architectures." *Software: Practice and Experience* 42 (5):559-583. doi: 10.1002/spe.1077.
- The Open Group. 2011a. *The Open Group Service Integration Maturity Model (OSIMM), Version 2*. Reading, UK: The Open Group. Technical Standard.
- The Open Group. 2011b. *Service-Oriented Cloud Computing Infrastructure (SOCCI) Framework*. Reading, UK: The Open Group. Technical Standard.
- The Open Group. 2011c. *SOA Reference Architecture*. Reading, UK: The Open Group. Technical Standard.
- W3C. 2004. "Web Services Architecture." Accessed 4 June 2015. <http://www.w3.org/TR/ws-arch/>.
- Xiaoye, Jin, Liu Xianhui, and Zhao Weidong. 2013. "The convergence of SaaS and SOA magnify effect." Proceedings of the 2nd International Symposium on Computer, Communication, Control and Automation (ISCCCA-13), Paris.

APPENDIX A: Business Motivation Model and Business Model of OPT and Infopub

Table A-1. Ends and Means of OPT's Business Motivation Model

Component	Description
Ends:	
Vision	Towards excellent service in innovative and optimized systems for transport planning, management, and public information .
Goal	To grow strongly to maintain the leadership in the national level to aim the international market diversification by continuously investing in innovation.
Means:	
Mission	Maintain the solution based on the up-to-date and competitive products, through regular introduction of new features and increase the adequacy to the client companies' needs, to be adapted from small dimension companies and leveraged into international.
Strategy	<ul style="list-style-type: none"> - Providing national transport companies with the best products and services in support of operational management and resource optimization; - Providing products and innovative applications designed to automate and increase the quality of public information; - Promoting sustainable and innovative research and development in the most relevant scientific fields by using the national scientific and technological system; - Evolving the development platforms and deliver the solutions via the web; - Making the products and services which are competitively developed in nearby markets, particularly targeting small and medium sized companies and international markets; - Adapting the solution systems to new markets, such as metros, railways, air transport and rostering management in hospitals; - Exploring the interactions with new complementary business areas such as ticketing, point systems, and analysis of supply and demand.

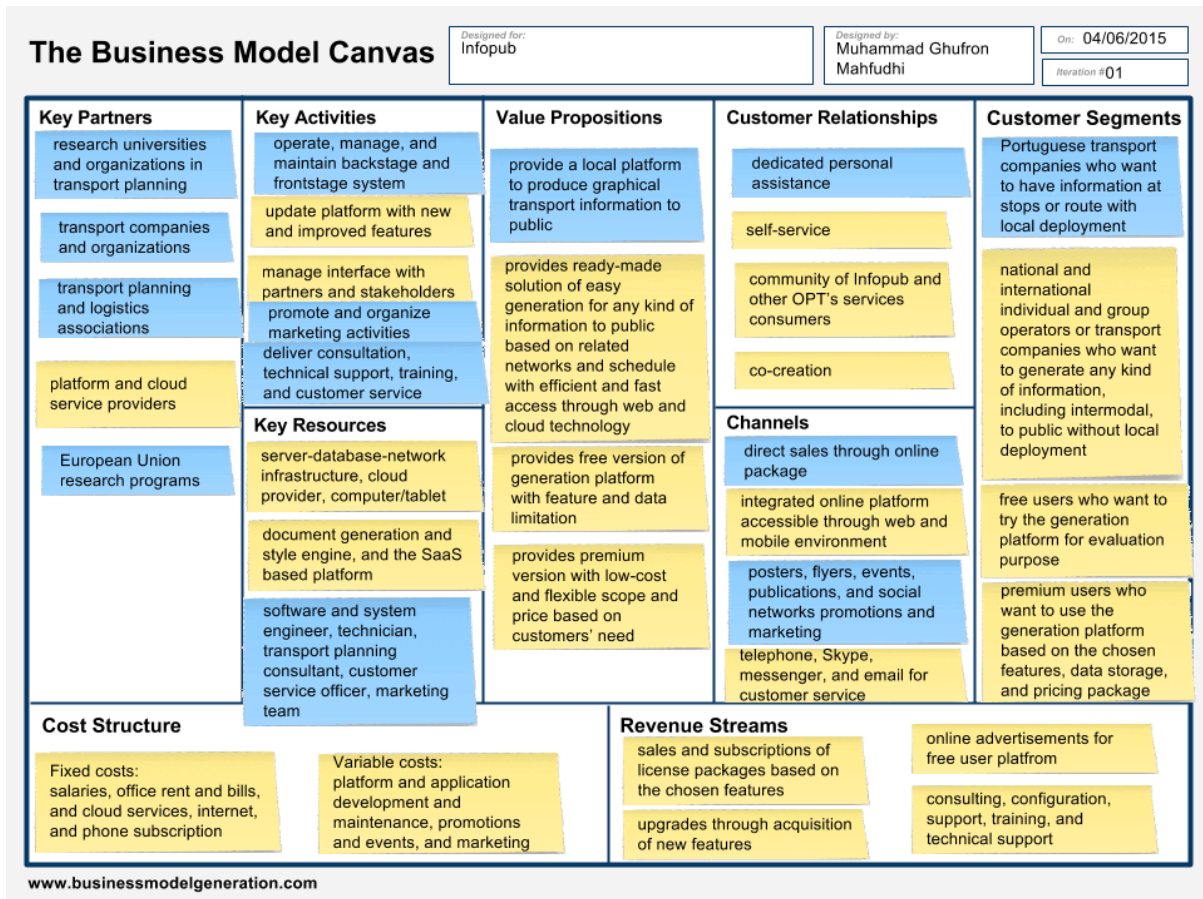


Figure A-1. Business Model Canvas of Infopub Service from OPT

APPENDIX B: Maturity Model Analysis of OPT and Infopub

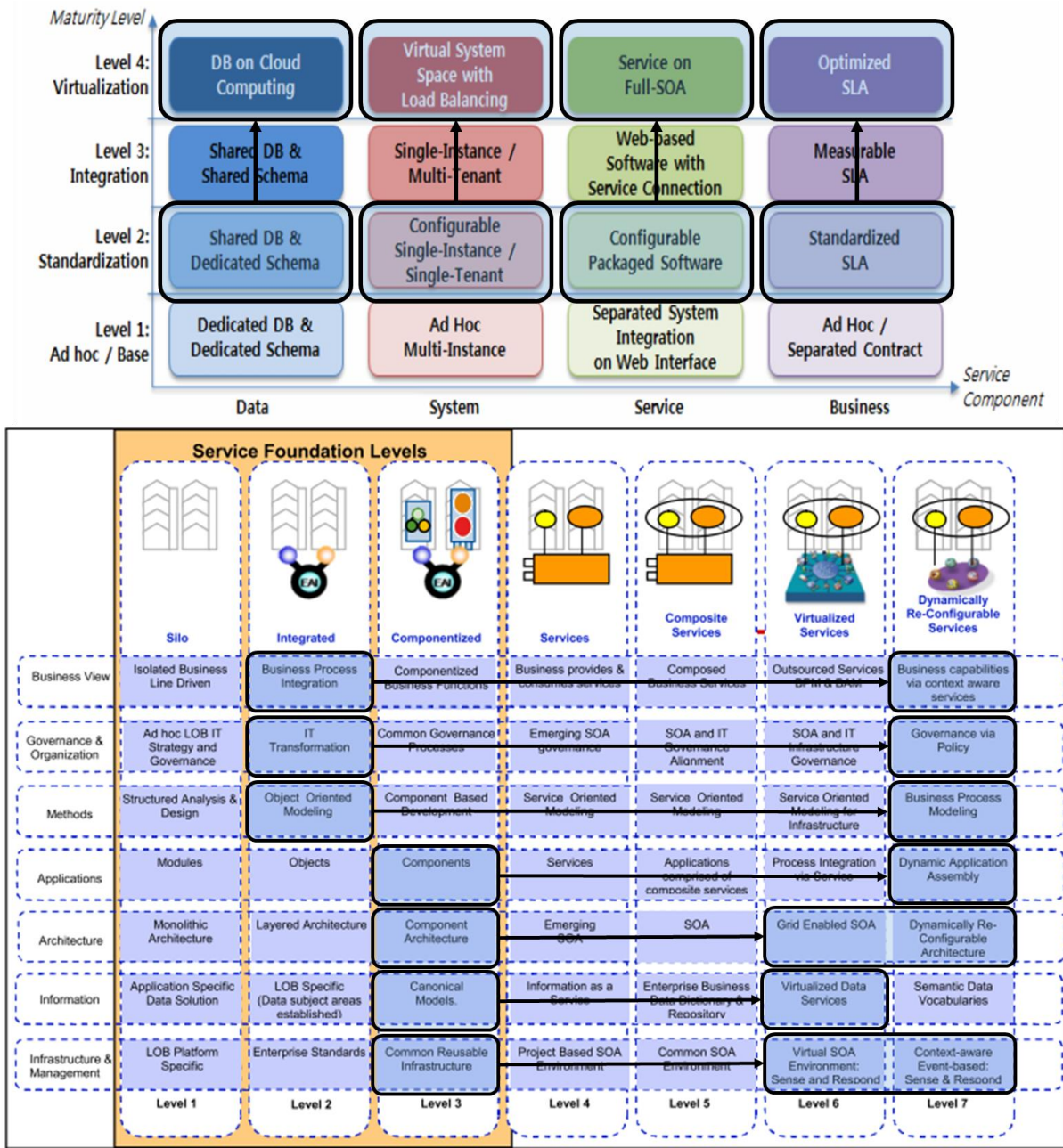


Figure B-1. Evolution Roadmap of Maturity Model for OPT and Infopub

Table B-1. SaaS Maturity Model Analysis of OPT and Infopub

SaaSMM Dimension	Current Maturity Level (1-4)	Summarized Assessment	Target Maturity Level (1-4)	Recommendations
Data	2	<p>Strengths: all data is collectively stored in OPT’s database server with different database access per user based on server and database location and credentials, except for the images and styles data which are stored locally, and customers are allowed to import the networks data from their database.</p> <p>Weaknesses: the data is hard to manage since it’s possible to be deployed in a different server location so that customers need to memorize the server location, and it does not support multi-tenancy.</p>	4	Deploy shared and distributed database and schema with cloud computing under cloud-based infrastructure to optimize the multi-tenant environment through well-defined set of metadata with high security management and different access based on the authentication and authorization methods.
System	2	<p>Strengths: customers use a predefined application instance with particular configuration so that they can have different solution application with more freedom of management.</p> <p>Weaknesses: still use the multi-instance applications which are installed per user, thus hard to be upgraded, managed, and maintained. Besides, the design components</p>	4	Deploy the system in single instance for easy management and maintenance with multi-tenant supporting platform. Use of virtualization in the cloud load-balancing system with virtual instance for each user with dynamic resource management based on the quantity of service used to achieve high availability and scalability.

SaaSMM Dimension	Current Maturity Level (1-4)	Summarized Assessment	Target Maturity Level (1-4)	Recommendations
		are still mainly stored in the application side.		
Service	2	<p>Strengths: the Infopub system is developed in object-oriented paradigm using object-relational mapping (ORM) technique into three main service package modules: data manager, BusMap, and BusSched.</p> <p>Weaknesses: Even though it uses services from other modules, the Infopub system components are not developed in service orientation, so that the same services are delivered through different software package, but with some configurations of modules based on client's requirements</p>	4	Develop and deploy the system's components as services based on the requirements covered in business processes into a full SOA system and deployed in service technology, such as web service.
Business	2	<p>Strengths: customers can contact OPT through telephone, email, social media, and direct contact to perform a service contract and support based on the defined service policy provided by OPT.</p> <p>Weaknesses: the business process for SLA still depends on simple contract and direct agreement which</p>	4	Optimizing a standard SLA policy adaptation to customers by using flexible and dynamic methods for measuring the amount of service used. By deploying the application in a PaaS, the service usage can be tracked using the PaaS provider's platform.

SaaSMM Dimension	Current Maturity Level (1-4)	Summarized Assessment	Target Maturity Level (1-4)	Recommendations
		is suitable for each customers' needs.		

Table B-2. OSIMM Analysis of OPT and Infopub

OSIMM Dimension	Current Maturity Level (1-7)	Summarized Assessment	Target Maturity Level (1-7)	Recommendations
Business View	2	<p>Strengths: business had good understanding of IT capabilities and had already defined service standard with direct SLA agreement.</p> <p>Weaknesses: there was no standard technical documentation for system development, no formal enterprise architecture that align business capabilities and IT, and application interdependencies and complexities which affects business agility.</p>	7	Using context-aware services on demand with a well-defined enterprise architecture and business process management to meet the process with the defined SLAs, so that the business capabilities can be aligned with services assets that transcend the current application-centric views. By using the context-aware cloud services from PaaS providers, this will support service specialization for each customers defined by policy from OPT, such as package recommendation, reminder of data changing (could be from manual editing or integration with GIST system) to suggest new generation, template recommendation based on customer's company profile, personal billing, etc.
Governance & Organization	2	<p>Strengths: use of a standard process for application development</p>	7	Drive changes into adaptive enterprise with concern of service as

OSIMM Dimension	Current Maturity Level (1-7)	Summarized Assessment	Target Maturity Level (1-7)	Recommendations
		<p>and maintenance.</p> <p>Weaknesses: the IT organization is mostly application-centric with specialized skills to manage the development and support for specific applications.</p>		<p>elements of business strategy using a formal SOA governance, including focus on services, business process, and architecture based on the changing business needs with alignment into IT and business capabilities. The use of PaaS with management services for governance control and input to key business decisions.</p>
Methods	2	<p>Strengths: use of defined development process with object-oriented practices and object-relational mapping (ORM) with several defined milestones throughout the development process.</p> <p>Weaknesses: the development process did not support business processes and services modeling and focus on the applications development with traditional software methodology, thus lacking of service reusability.</p>	7	<p>The application of the proposed User-Centered Service-Oriented Software Reengineering which integrate a model-based methods from business aspect into technical aspect using business motivation (BMM), business process modeling (BPMN), service identification and modeling (SoaML), and service technology (web service –WSDL, XSD, BPEL).</p> <p>Promote iterative development and ensure reusability of the components as services, controlled with service governance.</p>
Applications	3	<p>Strengths: Infopub system was developed based on the base of three main components</p>	7	<p>Promote agility by modernizing the Infopub system into service-oriented system using</p>

OSIMM Dimension	Current Maturity Level (1-7)	Summarized Assessment	Target Maturity Level (1-7)	Recommendations
		<p>(data manager, BusMap, and BusSched) with the use WCF service to enable data communication in real time as introduction of service-orientation.</p> <p>Weaknesses: the application was still developed in desktop environment with the minimum use of service-orientation within communication protocol between servers, not apply service concept from business service based on BPM into SOA, and the application was hard to be changed/ upgraded as it is still dependent. Besides, the style configurations was still stored in the application layer in customer side.</p>		<p>loosely-coupled services developed with service-oriented SaaS reference architecture and SOA patterns.</p> <p>As Infopub was developed within Microsoft and Oracle environment, it can use Microsoft Silverlight with implementation of business process as service as and deployment in web environment. The application needs to be designed and developed using a context-aware dynamic component with inference ability to support dynamically reconfigurable business and infrastructure services</p>
Architecture	3	<p>Strengths: the general architecture of Infopub was developed in component-based architecture which consist of data manager and document generator with message-based communication protocol.</p> <p>Weaknesses: the component-based architecture was not loosely-coupled enough</p>	6/7	<p>Develop the IT architecture based on service-oriented SaaS reference architecture with re-usable business services implementation, SOA pattern, and ESB for integration to achieve its goals. Furthermore, the IT provisioning is now at the cloud, which supports a virtualized approach and takes care of service availability, bandwidth</p>

OSIMM Dimension	Current Maturity Level (1-7)	Summarized Assessment	Target Maturity Level (1-7)	Recommendations
		and it consisted of the three main components application that the architecture modules were not flexible and hard to be modified.		and storage concerns and other unpredictable demands on the system. The services will be deployed using web services technology using ESB and SOA middleware components to integrate all of the services. The extension to level 7 can be done by implementing an inference component to dynamically select the suitable architecture based on the condition and context provided by the service request.
Information	3	<p>Strengths: the data stored within Infopub is the standard information of the public transport's network and schedule which is easily managed using Data Manager module and able to be used for other applications. It is able to import/export from/to GTFS, TransXchange, and CSV format.</p> <p>Weaknesses: the database was still specific for Infopub system and need other mechanism to be integrated to other system.</p>	6	<p>Virtualization on data layer to construct database and schema into cloud computing using cloud services so that the system can achieve an optimized multi-tenant environment.</p> <p>It can be improved into level 7 by applying semantic web construct or ontologies from a defined standard to develop a formal business information model.</p>
Infrastructure &	3	Strengths: OPT built its own infrastructure	6/7	Deploy web services management

OSIMM Dimension	Current Maturity Level (1-7)	Summarized Assessment	Target Maturity Level (1-7)	Recommendations
Management		which consist of database server, application server, and network and management server. Weaknesses: the infrastructure was not scalable enough and hard to maintain. SOA-specific infrastructure (services management, BPM) was also absent.		infrastructure to support a scalable enterprise-scale SOA deployment into PaaS provider. Deploy Business Process Management (BPM) infrastructure. Deploy SOA security infrastructure to be able to support security policies defined at the service level.

APPENDIX C: Detailed Situational Analysis

Table C-1. PESTEL Analysis of Infopub in SaaS Model

Factor	Coverage
Political	<ul style="list-style-type: none"> (+) New EU funds in 2020 for the development. (+) Creation of Banco de Fomento to leverage a portion of the targeted European Structural Funds to finance for Small and Medium Enterprises. (-) Portuguese’s policies are still conditioned by the intervention of the Troika. (-) Unintegrated and different governmental policies for services delivery in cloud for global market and issues regarding security, privacy, location and ownership of data.
Economic	<ul style="list-style-type: none"> (+) First signs of improvement in the Portuguese economy result of the intervention of the Portuguese economy by Troika. (+) Need for Portuguese’s economic growth and sustainability through its local business. (+) Cloud computing offers on-demand business delivery with less capital-intensive and promotes rapid time-to-market and flexibility. (-) Consolidation program of public finances still in progress as a result of the intervention of the Portuguese economy by Troika. (-) Portuguese organizations are very indebted unable to access incentive programs.
Social	<ul style="list-style-type: none"> (+) Increased responsiveness to environmental issues and business process efficiency by organizations as a way to differentiate in the marketplace and communities where they operate. (+) Cultural change in management terms as a consequence of the crisis that led to the intervention of the Troika in Portuguese economy. (+) Focus on quality differentiation by Portuguese economic agents at the expense of differentiation by the prices in the markets where they operate respecting environmental issues. (+) Increased use of internet for personal and professional activities.
Technological	<ul style="list-style-type: none"> (+) Portuguese and EU government support for research and development for transport information for public. (+) The trend transformation of standardization and industrialization of IT through services via internet. (+) The emergence of virtualization, SOA, and cloud computing becomes major innovation in IT-based services to enable faster and efficient delivery for economic growth. (-) Technology is constantly changing, thus emerge the need of checking and updating the latest IT technology and support staff training to achieve the optimal result.
Environmental	<ul style="list-style-type: none"> (+) The application of cloud services reduces the electricity use, CO2 emissions, electronic waste, and paper use through the use of IT

Factor	Coverage
	resources outsourcing via online channel.
Legal	(+) As consequence of the crisis and intervention of the troika, the government attempted for simplification of laws. (-) Bureaucracy that comes from the allocation of EU funds is increased due to lack of coordination of all stakeholders and integration throughout the process. (-) Political parties approve overly complex laws. (-) There is no common legal framework for internet business which is necessary for customer protection so that vendors need to face local regulations that often restrict the free flow of information between countries. The legislation is related to security (sensitive data), ownership and location of data, confidentiality, and intellectual property

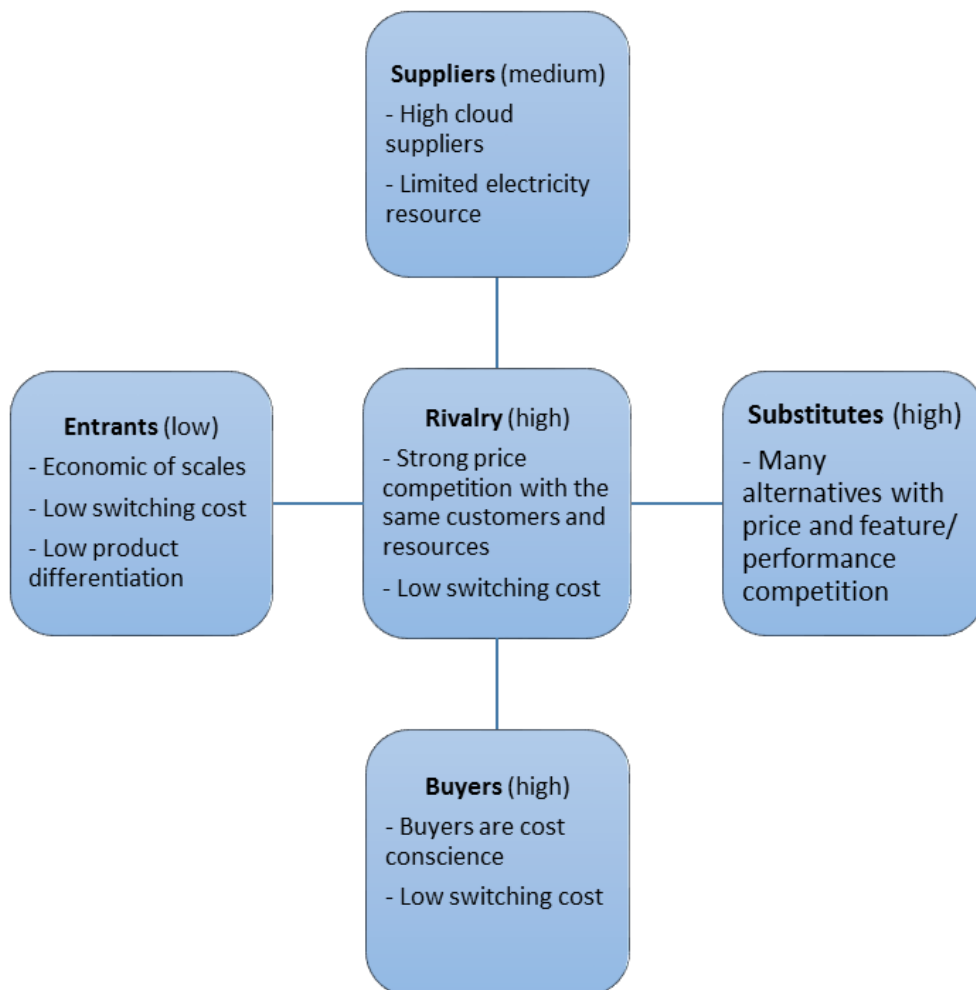


Figure C-1. Competitive Force Analysis of Infopub in SaaS Model

APPENDIX D: Modernization Requirement Specifications**Table D-1. Modernization Requirements Specifications**

No.	Aspect	Description	Priority
1	System	The target system development shall reuse the legacy system (SaaS Infopub = Infopub + IMS + web structure + other composition)	Essential
2		The target system shall support the existing generators (BusMap and BusSched) with generation engine improvement which supports generation for any kind of information related to passenger's public transport, such as intermodal from group of operators	Essential
3		The system shall support configuration by transport operators which consist of design and templates for the generator	Essential
4		The system shall support multilanguages: English, Portuguese, etc.	Essential
5		The system shall provide custom layouts from OPT or from operator, and the use of intermodal/transport connection	Conditional
6	Data	The system shall provide data process and management: file upload (data import), data consolidation, and data editing	Essential
7		The system shall integrate database from Infopub and IMS to preserve the generator's persistence	Essential
8		The system shall be able to input the data of network from external file in CSV or GTFS format and produce SVG and/or PNG format file, depends on the payment profile and need to concern whether it will be for internal or external address.	Essential
9		The system shall allow the customers without the GTFS format data to create the network structure from scratch and export it to GTFS format	Essential
10		The system shall allow customers for template configuration by editing the following data: <ul style="list-style-type: none"> a. Global colors and their tones b. Styles: with or without outline, outline color, outline thickness, color background, font (type, outline, color), rounding or the rectangular corners, etc. c. Global color associated to routes d. Pictograms e. Image 	Essential
11	Design	The system shall introduce the design component directly on the site and without files: by operator (e.g. logo), by generator (e.g. headers, footers, etc.), and others (pictograms interfaces with other transports)	Essential
12		The system should provide basic templates for BusMap and BusSched generators as follows: <ul style="list-style-type: none"> a. BusMap generator: layout, header, footer, header length, body, styles (only some thicknesses, lengths, and colors) b. BusSched generator: layout, header, footer, schedule 	Essential

No.	Aspect	Description	Priority
		<p>header, the schedule, the orientation (vertical/horizontal), styles (several items)</p> <p>The detailed template configuration is described as follows:</p> <ul style="list-style-type: none"> a. General: <ul style="list-style-type: none"> i. Page dimension, margins, header, footer ii. Logo b. BusMap: <ul style="list-style-type: none"> i. Without using style ii. Header and footer type iii. Distance between routes iv. routes format (with or without boundary) v. Waypoints and endings format (rounding corners, with or without contour slide, interface through or below) vi. Permission for corners c. BusSched: <ul style="list-style-type: none"> i. Orientation: vertical or horizontal, partial or complete, up or down or left or right ii. Stops interface (vertical orientation) iii. Header and footer type iv. Sign of exceptions v. Exceptions in schedule or footer vi. Number and types of zoning vii. How to characterize the frequencies 	
13	Technology	<p>The system should be hosted in hosting server with the following options:</p> <ul style="list-style-type: none"> a. Traditional hosting solution: using IIS environment with Oracle or other database system, which is the current development environment of OPT, or using other environments. b. Cloud platform solutions: <ul style="list-style-type: none"> i. Microsoft Azure ii. Google Cloud Platform 	Essential
14	Interaction menu	<p>The system should provide the basic interaction menu which consist of:</p> <ul style="list-style-type: none"> a. Login/Register/Exit b. Profile management c. Transport Network Upload with consolidate and edit menu d. Graphics management: Colors, Images, Pictograms, and Styles e. Configuration management: general and by generator f. Document generator: BusMap and BusSched 	Essential

APPENDIX E: The Artifacts of Legacy System

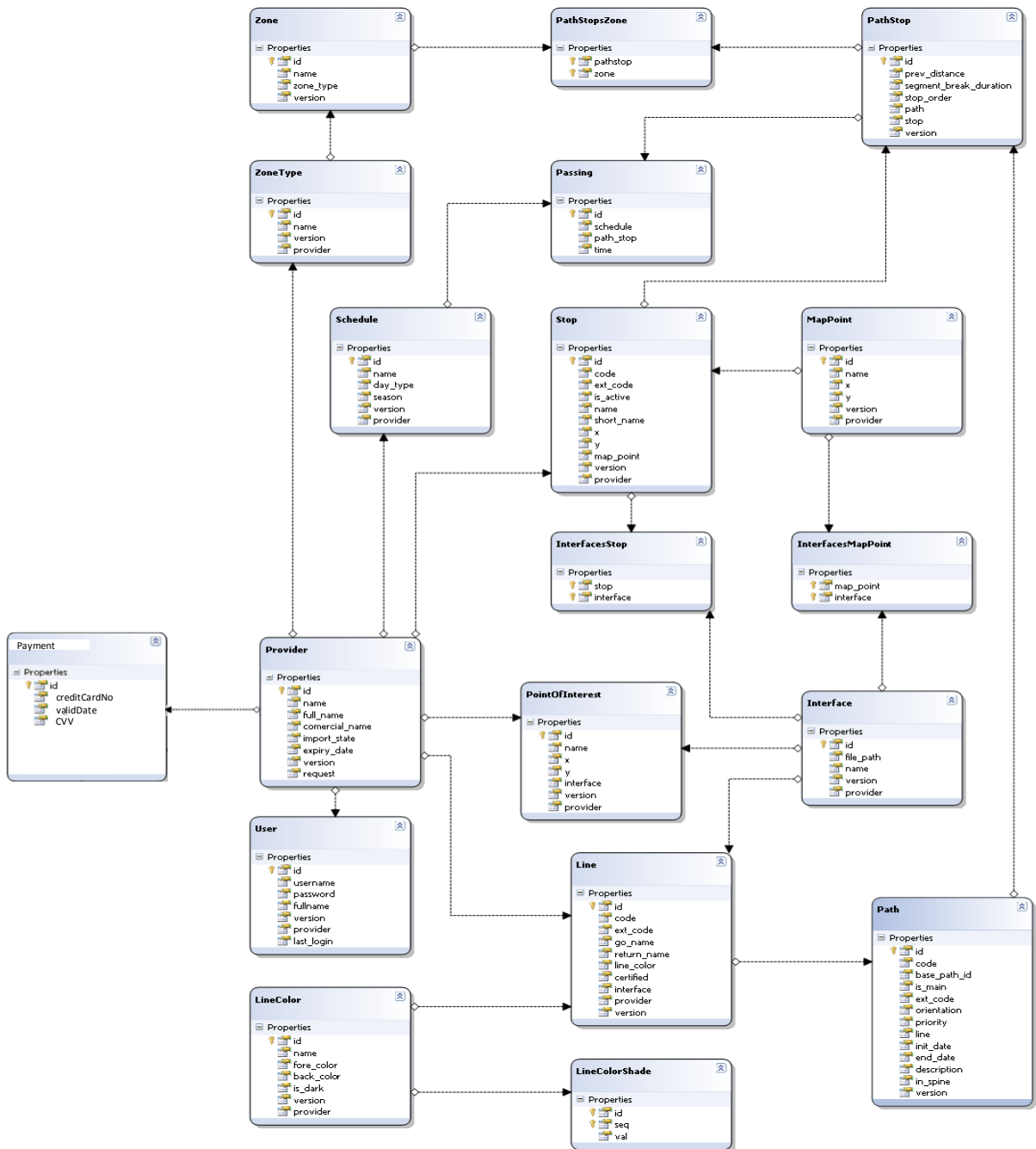


Figure E-1. Database Schema of Infopub (Meneses 2009)

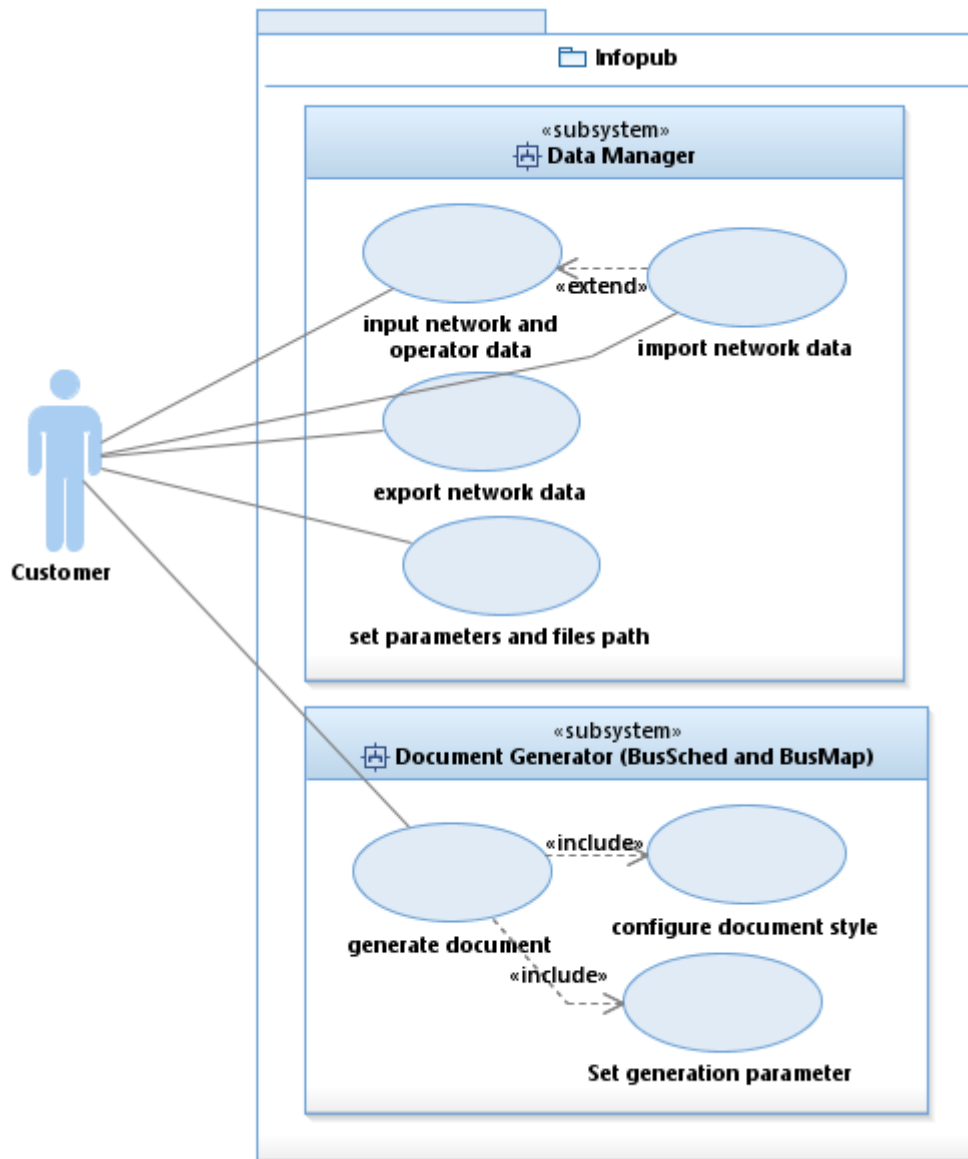


Figure E-2. Use Case Diagram of Infopub

APPENDIX F: User Interfaces of Legacy Infopub System

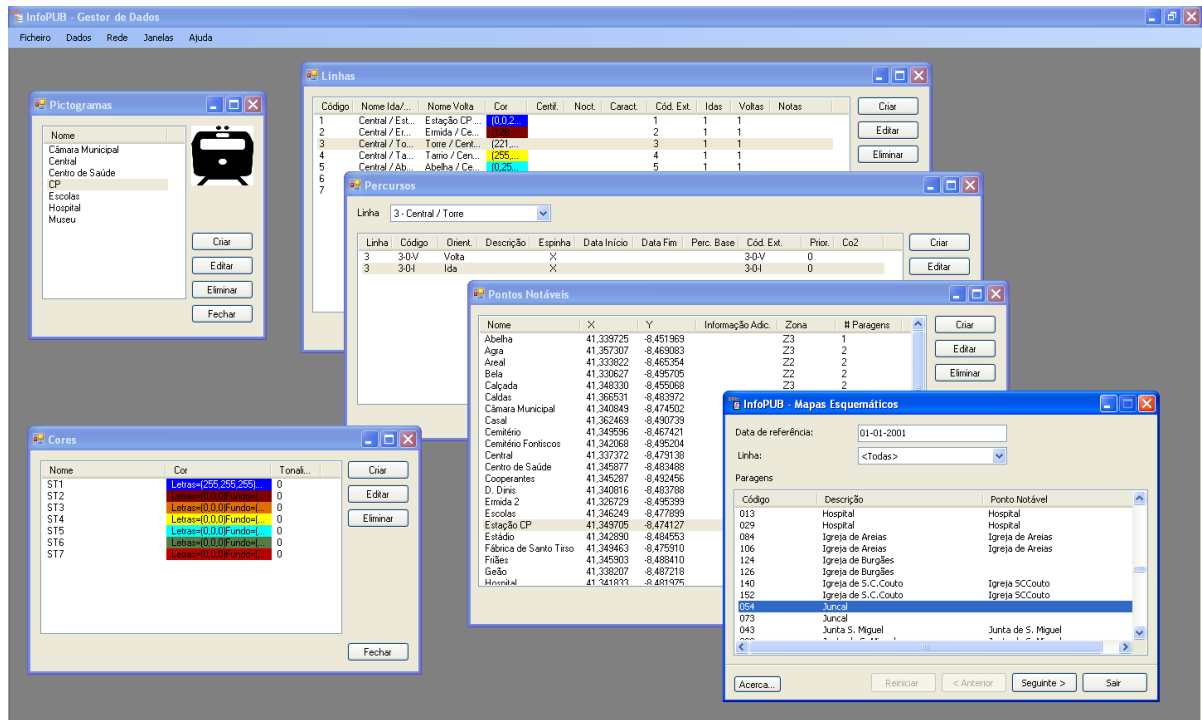


Figure F-1. Data Manager of InfoPub

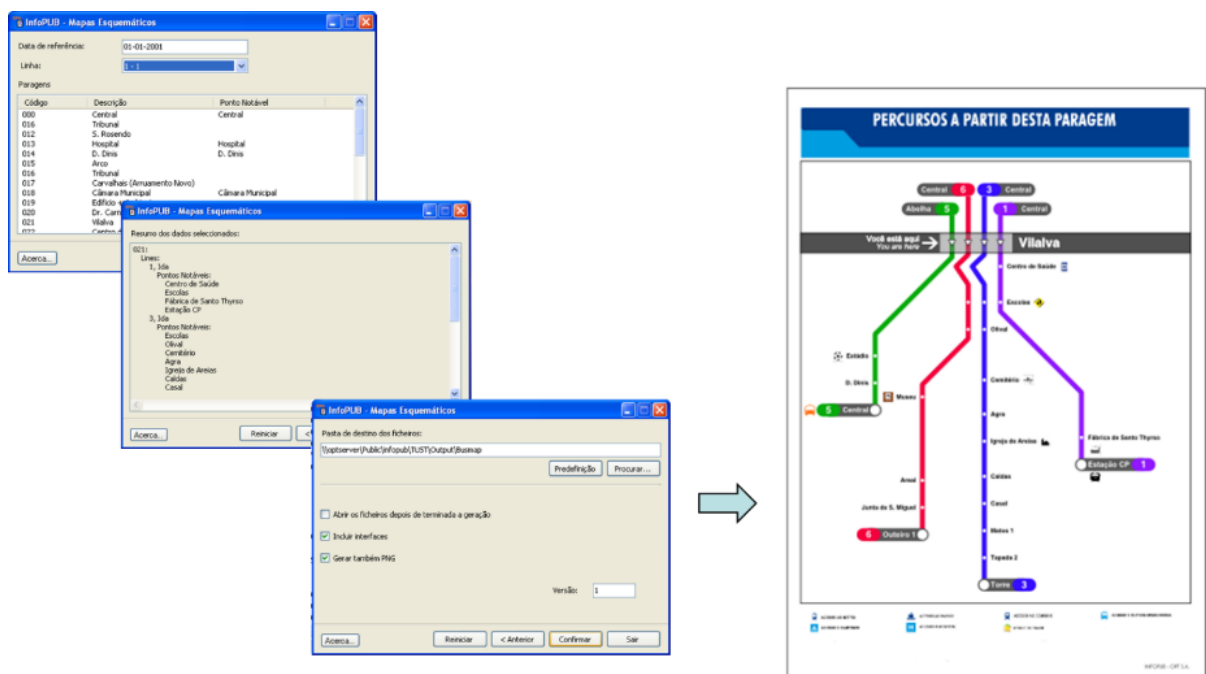


Figure F-2. BusMap Generator of InfoPub

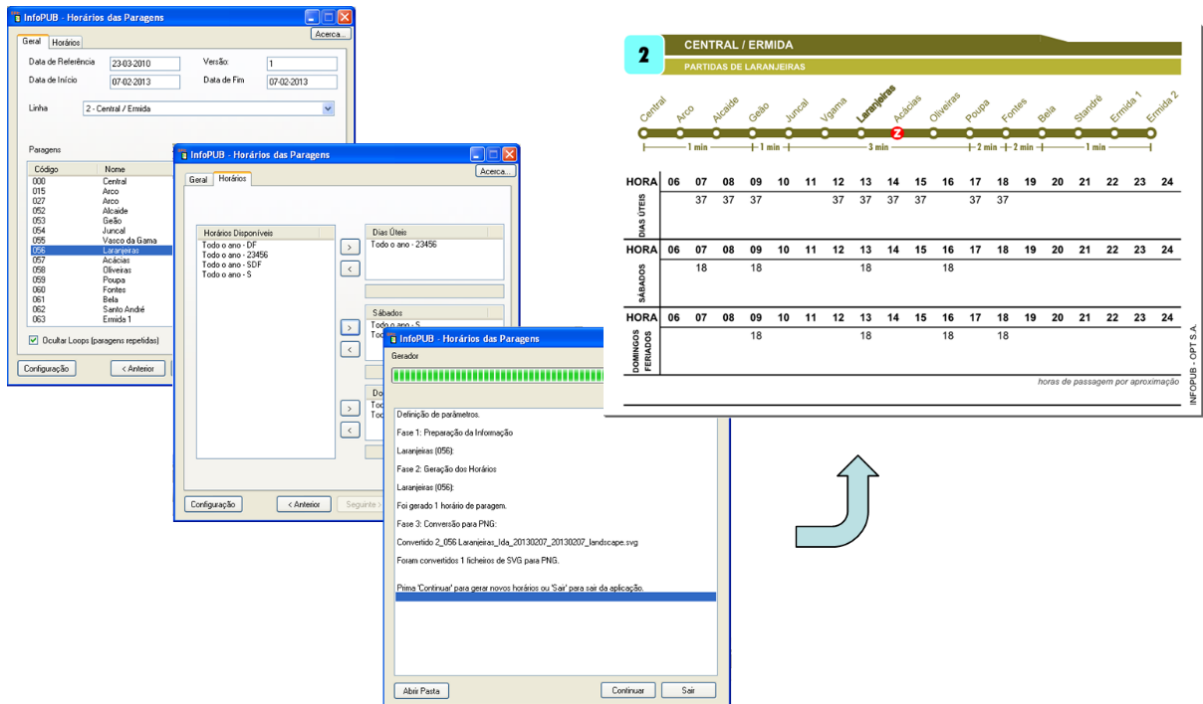


Figure F-3. BusSched Generator of InfoPub

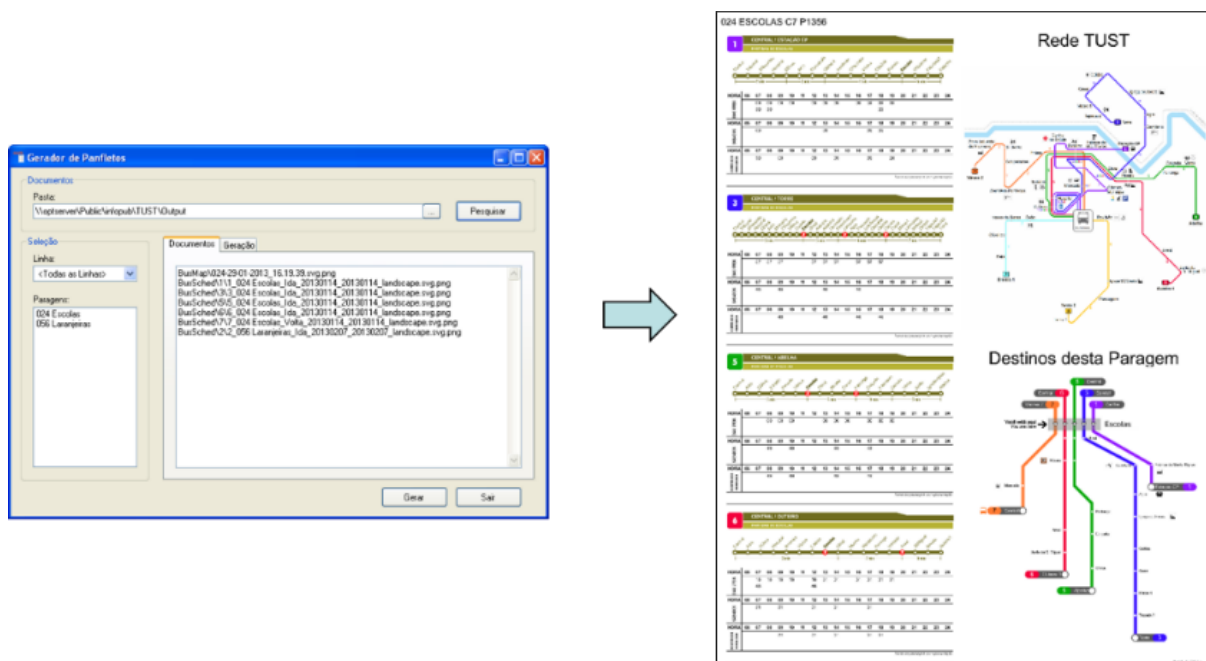


Figure F-4. MultiDoc Generator of InfoPub

APPENDIX G: Service Development Artifacts

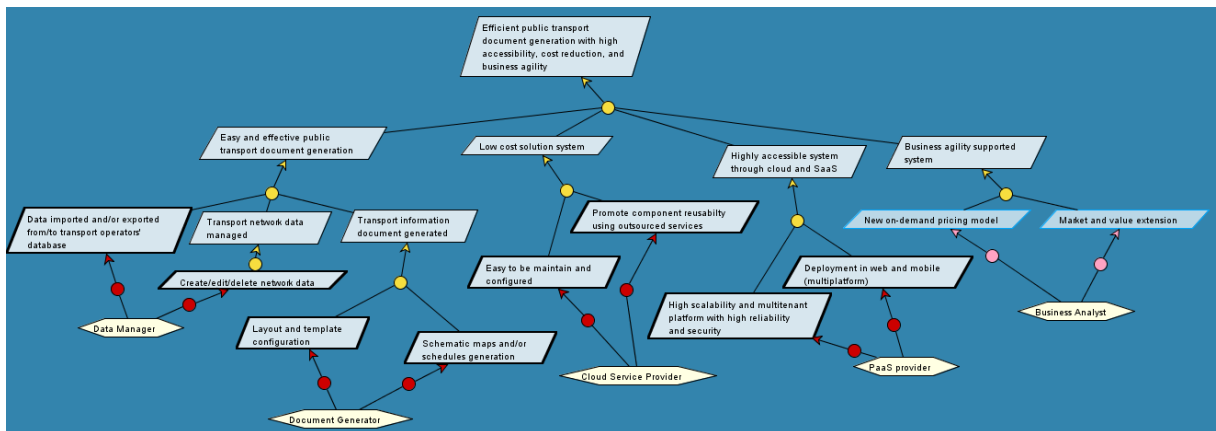


Figure G-1. Goal-Service Model of Infopub in SaaS

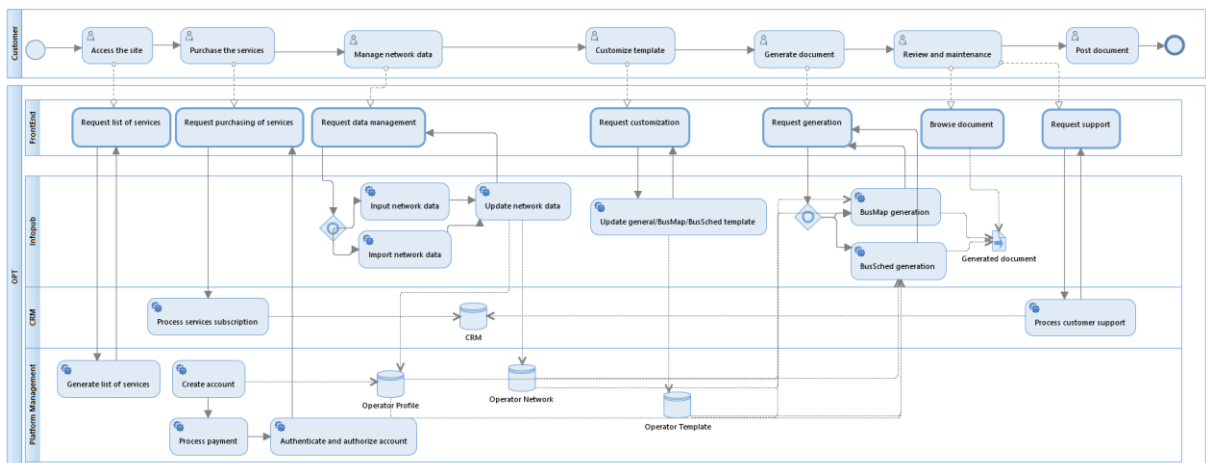


Figure G-2. Service Blueprint of Infopub in BPMN

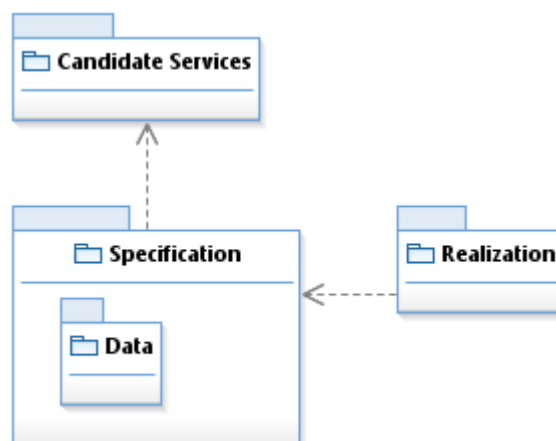


Figure G-3. Service Development Packages

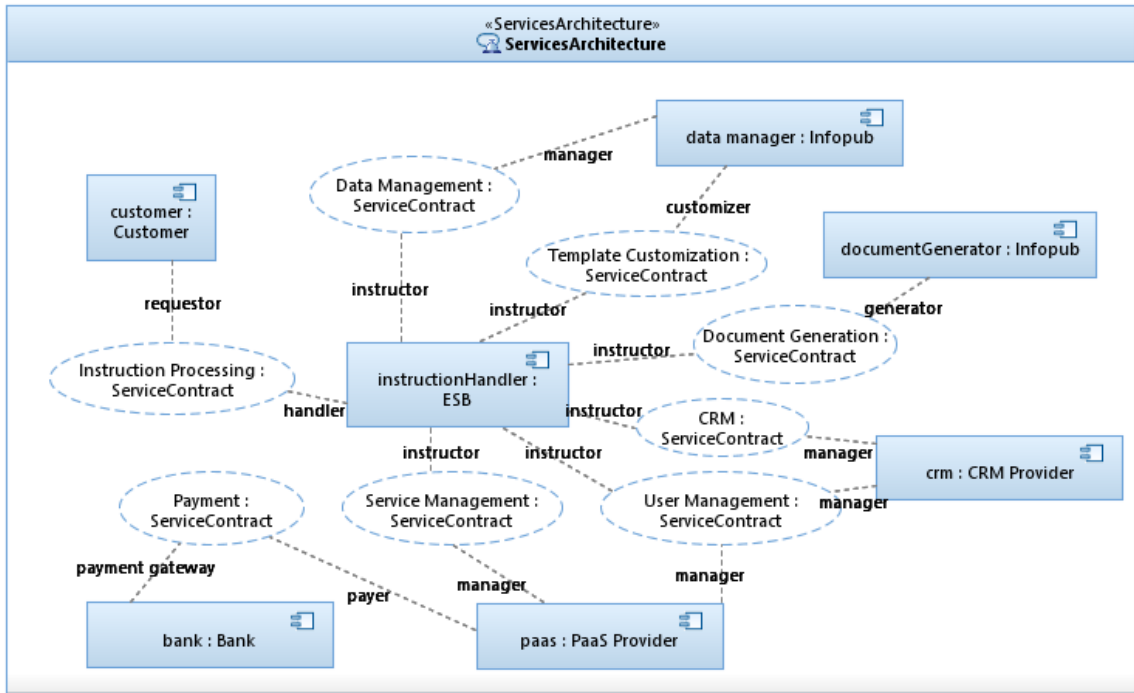


Figure G-4. Service Architecture of Public Transport Document Generator

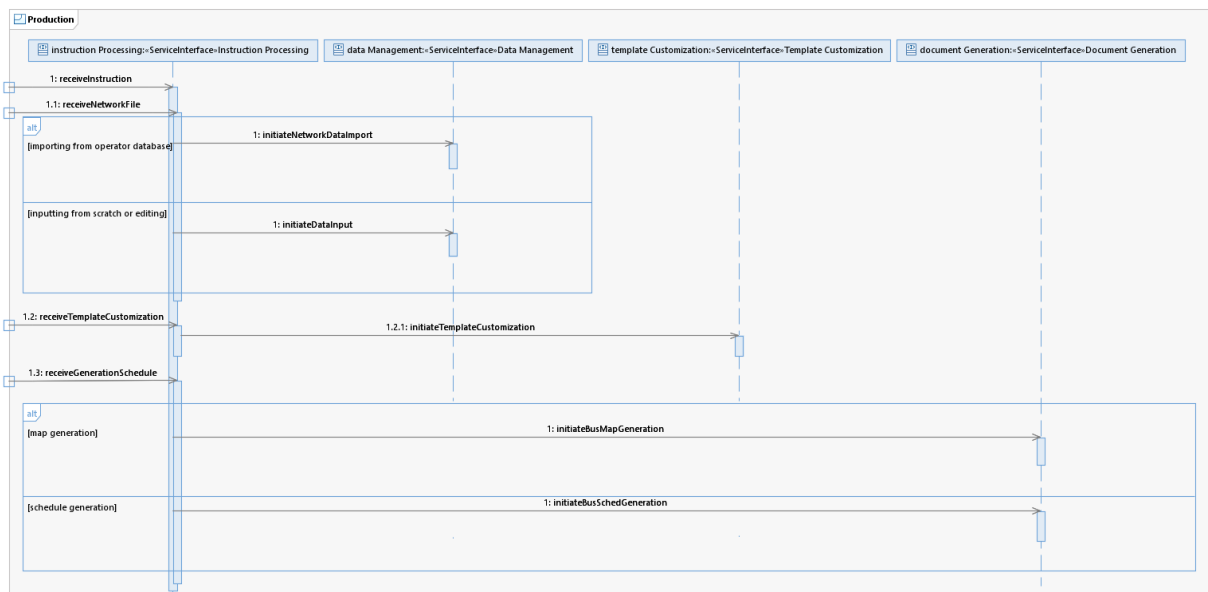


Figure G-5. Sequence Diagram for Service Contract of Information Production Service

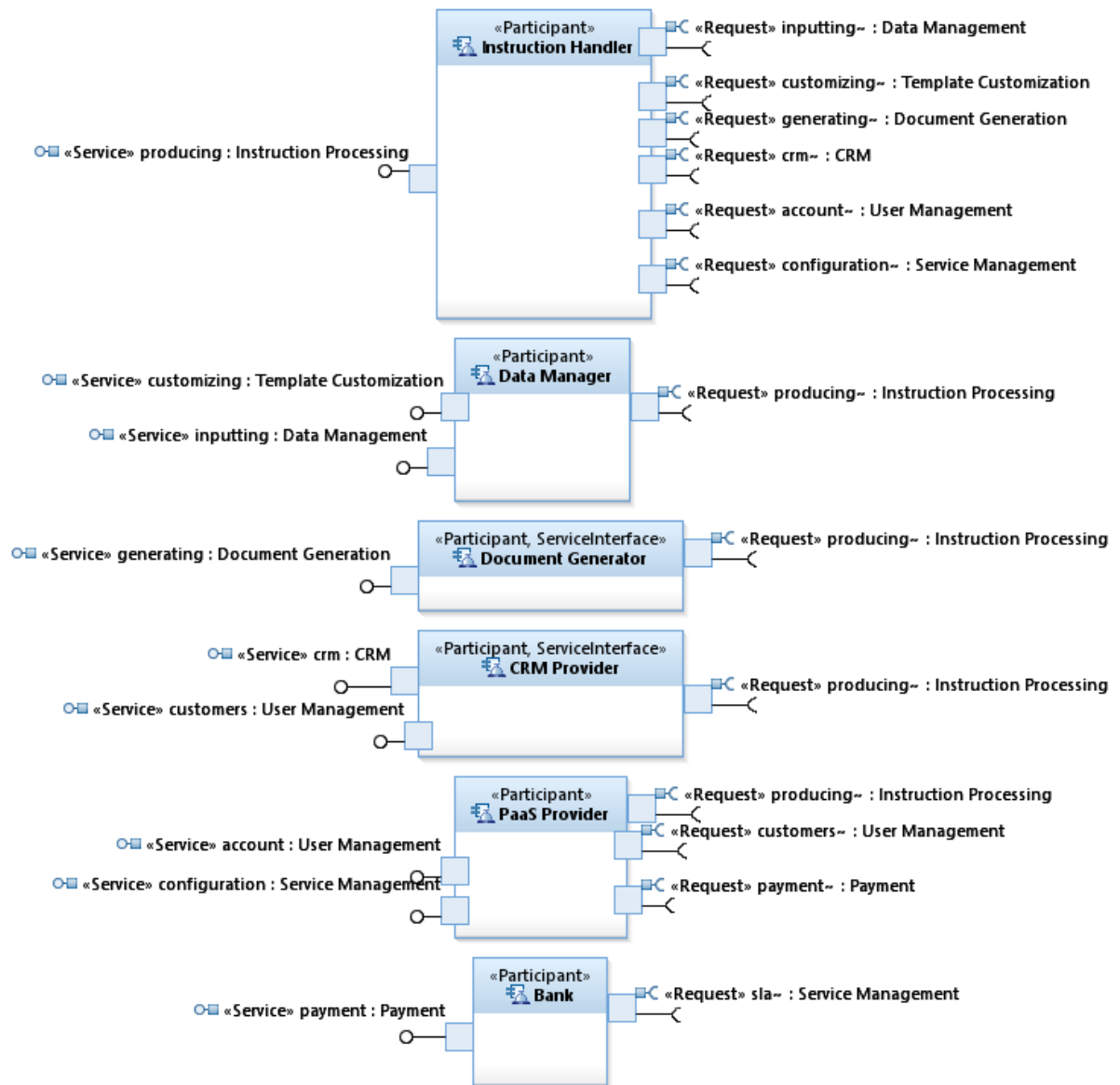


Figure G-6. The Identified Participants of Public Transport Document Generator

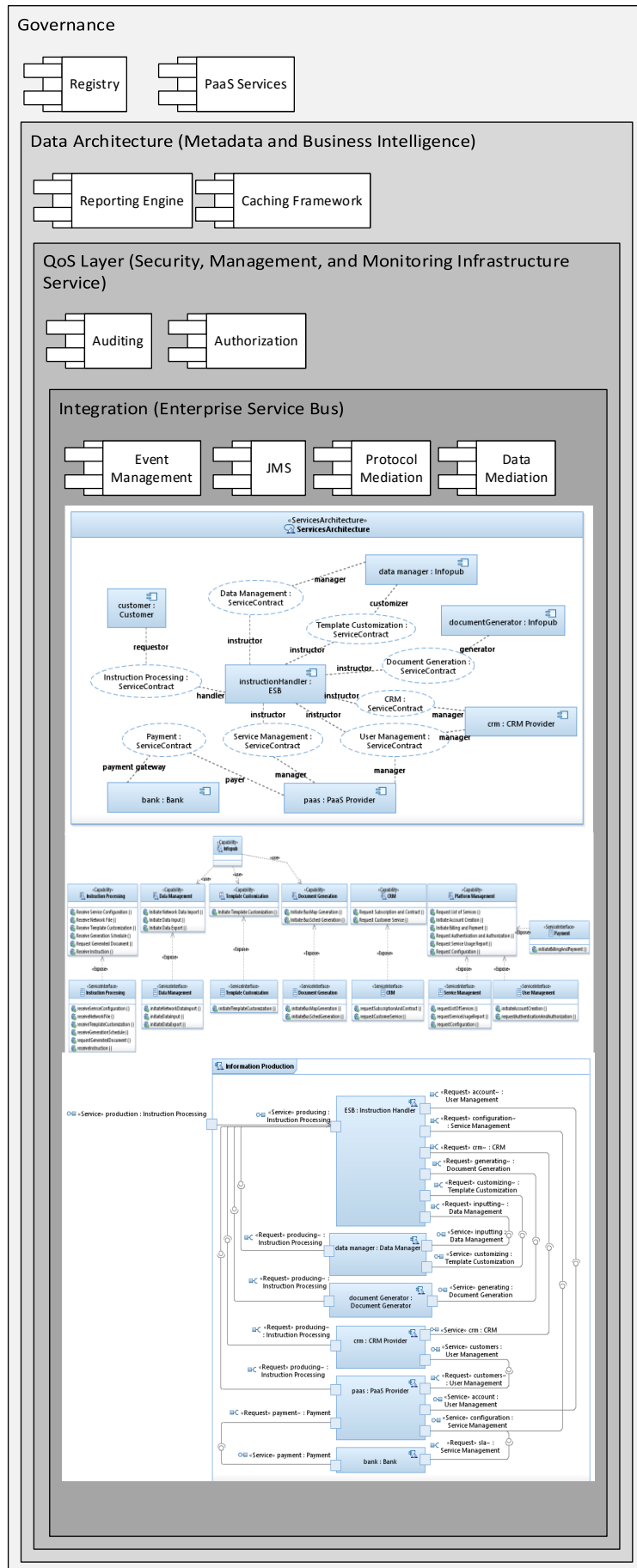
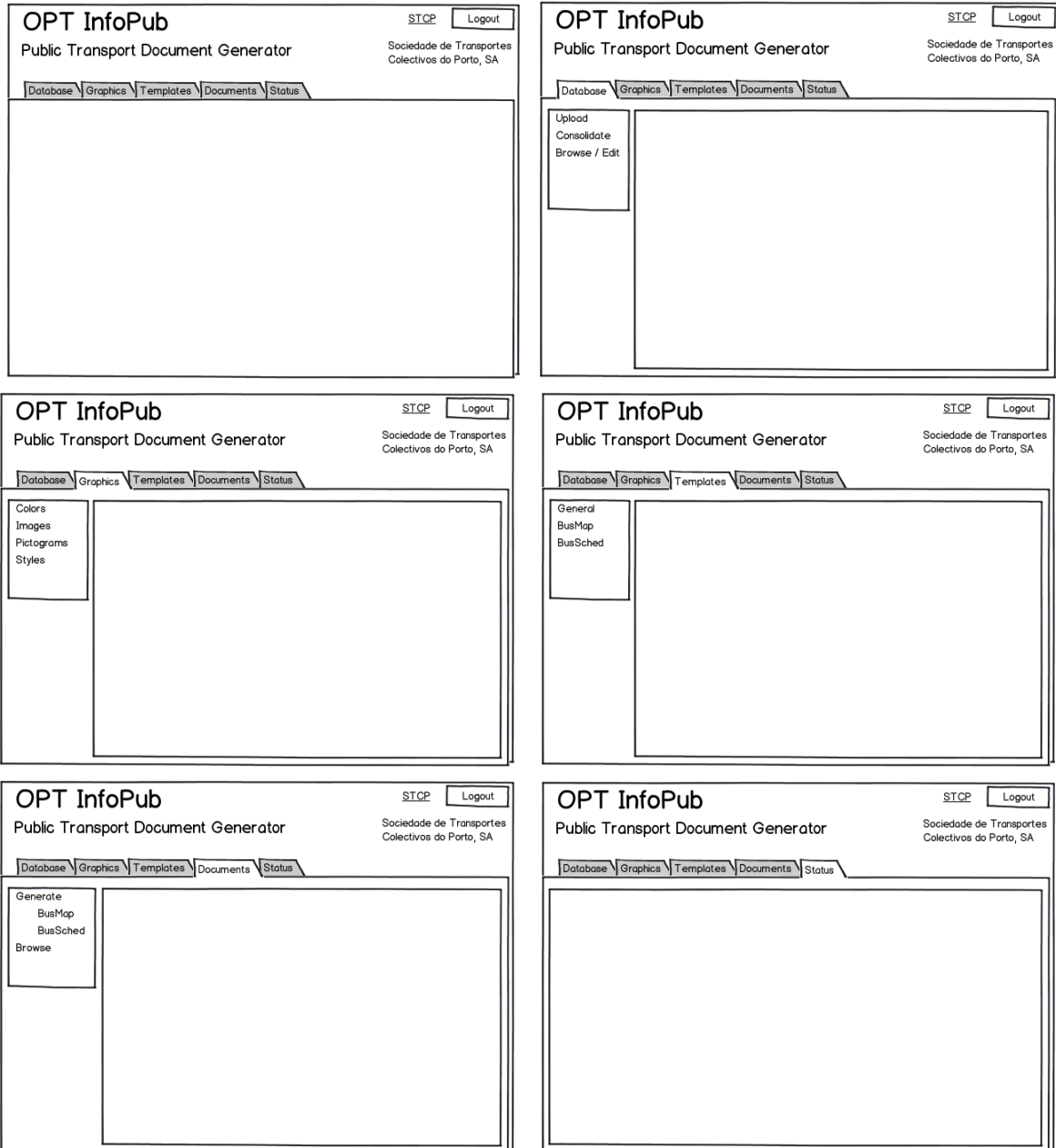


Figure G-7. Building Block Layer Architecture of Public Transport Document Generator

APPENDIX H: Prototype of Public Transport Document Generator

From Luís Filipe Ferreira's proposal presentation for Infopub in Web (2012):



From Meneses (2009):

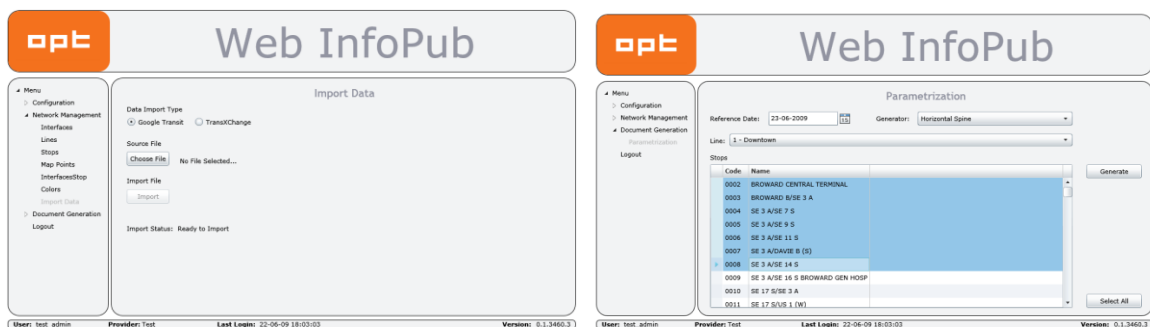


Figure H-1. Prototype of Public Transport Document Generator

APPENDIX I: Comparison between Microsoft Azure and Google Cloud Platform**Table I-1. Comparison between Microsoft Azure and Google Cloud Platform**

Parameters	Microsoft Azure	Google Cloud Platform
Environment	Microsoft Azure is easy to use for people familiar with Microsoft and Windows environment system with the support of .NET, even though it also support for other environment such as Linux, Java, Node, PHP, Python, and Ruby, thus ease up integration. With .NET framework, organization is able to address other instances, use a built-in service bus, which allows batch processing enterprise.	Google Cloud Platform has a bigger advantage for smaller companies since it charges by the minute with a 10-minute usage minimum. It has a strong point in infrastructure and data analytics with support of runtime in Go, Java, PHP, and Python. However, it forces developer to learn too much about Google Cloud and it has limited ability to customize container.
Cloud Deployment	Microsoft Azure consists of PaaS and IaaS with a wide range of platforms, plugins, and features with support of Windows Azure Storage for table-based storage, SQL Azure for SQL-based RDBMS, and Oracle RDBMS. It also support hybrid cloud services with Azure BizTalk Services for integration of private cloud with Azure's public cloud, thus developers only need to worry about maintaining the code, not the infrastructure.	Google Cloud Platform use Google App Engine as PaaS and Google Compute Engine as IaaS with instant scaling and support for Hadoop, BigTable, and BigQuery. However, there is no debug/shell access in standard AppEngine.
Strength Points	<ul style="list-style-type: none"> - Workloads: leading in media and Web2 - Tool: leading in IDE support and target (development, testing, etc.) - Integration: leading in user registry - SLA: leading in availability, reliability, vertical and horizontal scalability, multizones HA, credit - Data stores: leading in SQL - Management: leading in paid monitoring and autoscales - Misc.: leading in real time software patching and load balancing 	<ul style="list-style-type: none"> - SLA: leading in availability, reliability, horizontal scalability, multizones HA, and credit - Data stores: leading in column (table) - Management: leading in autoscale
Offers and Support	There are special offers for Microsoft Partner Network and support from expertise with flexible support plan to leverage the relationship with customers.	Not many offers as Azure.
Data Center	There are 19 data centers regions with the Europe region in Ireland and Netherlands.	There are 3 main regions of data centers: Iowa (US), Belgium (Western Europe), and Taiwan (East Asia).

Parameters	Microsoft Azure	Google Cloud Platform
Pricing	<p>Price is dependent on the support plan selected:</p> <ul style="list-style-type: none"> - Pay As You Go: pricing based on usage per month. - 12-month prepay with 5% discount and minimum purchase of €5,060 - Purchase through reseller or enterprise agreement 	<p>The Google App Engine pricing and product model is all about scalability where users start out with 1GB of free data storage and traffic, and is increased by activating the paid applications. There is a daily limit for extra resources to prevent budget exceeding. Additional resources are billed at a fixed unit rate with exception of instance hours as they can also be purchased on a retainer basis, while there is a discount for buying a set number of instance hours per week. There is also a Premier option, which unlocks all tools and resources and this option starts at \$150 per account per month.</p>
Infrastructure	<p>Azure experienced significantly more downtime than AWS or Google Compute Engine last year at 50.74 hours.</p>	<p>The strong side about Google Cloud Platform is the infrastructure and networking with less significant downtime than Azure, for around 5 hours. With the infrastructure, each instance live on its own network with strong virtualization security tools, including firewalls, routers, and subnets.</p> <p>-</p>

APPENDIX J: Modernization Framework Evaluation**Table J-1. Modernization Framework Evaluation**

Stage	Phase	Evaluation
Modernization planning	Legacy system understanding	The framework includes legacy system understanding by using service-oriented reverse software engineering to analyze the legacy system in use case, business entity, behavior, and architecture views. The analysis depth can be vary, depends on the development strategy, can be into the system model or even the code implementation. The framework focuses more on the system model analysis to generate a general view of the legacy system. This is supported by several modeling tools which support UML modeling.
	Target system understanding	The target system understanding is described in the maturity model and reference architecture. The maturity model describes the characteristics that are required for the target system based on the evolution plan regarding SOA and SaaS, while the reference architecture illustrates the general required components for developing a service-oriented SaaS system.
	Modernization feasibility determination	The framework includes modernization feasibility determination under the envisioning from business model and transformation. The framework considers the analysis of business architecture and model of the organization system with strategy plan definition from situational analysis. The evolution roadmap is described through the maturity analysis in SaaS and SOA implementation to identify the characteristics required for the target system. Several methods and best practices are described to support the analysis, such as BMM, BMC, PESTEL, Competitive Forces Model, resource and capability analysis, SWOT, and maturity model.
Modernization implementation and management	Candidate service identification	Candidate service identification is described in the service identification phase using goal-service modeling, domain decomposition, legacy system analysis, and SaaS components identification. Several tools also described such as KAOS diagram, service journey and service blueprint, business process model, etc.
	Implementation	The framework only provides the development

Stage	Phase	Evaluation
		<p>guideline using SoaML modeling and model generation into service architecture. The detail for implementation process is not described in the framework, thus developers need to identify the best practices in Web Services and SaaS implementation.</p>
	Deployment and provisioning	<p>The framework only suggest the process needed in deployment and provisioning with several recommendation.</p>
Case study		<p>The demonstrated case study is an industrial case study from OPT to modernize the legacy Infopub system into service-oriented SaaS system of public transport document generator. The case study was developed by following the proposed methodology and resulted a design specification and generated Web Services of the system using service-oriented architecture and model-driven development.</p>