UNIVERSITÄT KOBLENZ · LANDAU

IWVI

Department of Computer Science, Institute for Information Systems Research

A Service-oriented Approach for the Virtual Company Dossier

Master Thesis

To obtain the degree Master of Science in Information Management

submitted by
Daniel Reiser

First supervisor: Prof. Dr. Maria A. Wimmer
Research Group eGovernment

Second supervisor: Ansgar Mondorf
Research Group eGovernment

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Die Richtlinien der Forschungsgruppe für Qualifikationsarbeiten habe ich gelesen und anerkannt, insbesondere die Regelung des Nutzungsrechts.

Mit der Einstellung dieser Arbeit in die Bibliothek bin ich einverstanden.	Ja □	Nein □
Mit der Veröffentlichung dieser Arbeit im Internet bin ich einverstanden.	Ja □	Nein □

Koblenz - 23. Juni 2010

ABSTRACT

The development of a pan-European public E-Procurement system is an important target of the European Union to enhance the efficiency, transparency and competitiveness of public procurement procedures conducted within the European single market. A great obstacle for cross-border electronic procurement is the heterogeneity of national procurement systems in terms of technical, organizational and legal differences. To overcome this obstacle the European Commission funds several initiatives that contribute to the aim of achieving interoperability for pan-European public procurement. Pan European Public Procurement OnLine (PEPPOL) is one of these initiatives that aims at piloting an interoperable pan-European E-Procurement solution to support businesses and public purchasing entities from different member states to conduct their procurement processes electronically. Its second work package develops a solution to achieve cross-border data exchange of business documents between the actors of the public procurement process. This includes the development of a common data format, the so called Virtual Company Dossier (VCD), that serves as an electronic document container, as well as a supporting IT system. With the need for an interoperable IT architecture, the question for suitable implementation technologies and infrastructure components arises. As interoperability and inter-connection of distributed heterogeneous information systems are the major requirements in the European procurement domain, and the VCD sub-domain in particular, service-oriented architecture (SOA) seems to provide a promising approach to realize such an architecture, as it promotes loose coupling and interoperability. This master thesis therefore discusses the SOA approach and how its concepts, methodologies and technologies can be used for the development of interoperable IT systems for electronic public procurement. This discussion is enhanced through a practical application of the discussed SOA methodologies by conceptualizing and prototyping of a sub-system derived from the overall system domain of the Virtual Company Dossier. For that purpose, important aspects of interoperability and related standards and technologies will be examined and put into the context of public electronic procurement. Furthermore, the paradigm behind SOA will be discussed, including the derivation of a top-down development methodology for service-oriented systems. During the practical part, this top-down approach will be applied, including an analysis and design of the target domain as well as the implementation and deployment of corresponding software applications and technical infrastructure components. The results are primarily relevant for the ongoing work of the PEPPOL project, but can be transferred to any similar problem domain that deals with the development of distributed IT systems.

Zusammenfassung

Die Entwicklung eines IT-gestützten europäischen Systems zur öffentlichen Auftragsvergabe ist ein wichtiges Ziel der EU um die Effizienz, Transparenz und Wettbewerbsfähigkeit im öffentlichen Beschaffungswesen des europäischen Binnenmarktes zu verbessern. Ein großes Hindernis für die grenzüberschreitende elektronische Abwicklung der Beschaffungsprozesse ist die Heterogenität der nationalen Beschaffungssysteme in Bezug auf technische, organisatorische und rechtliche Unterschiede. Zur Überwindung dieses Hindernisses finanziert die Europäische Kommission einige Initiativen, um grenzüberschreitend Interoperabilität der öffentlichen Beschaffungssysteme zu erreichen. Pan European Public Procurement OnLine (PEPPOL) ist ein solches Projekt, das die Pilotierung eines interoperablen pan-europäischen Systems zur elektronischen Unterstützung der öffentlichen Beschaffungsprozesse als Ziel hat. Unternehmen und öffentliche Beschaffungsorgane aus verschiedenen Mitgliedstaaten sollen dabei unterstützt werden, ihre Beschaffungsprozesse elektronisch durchzuführen. Arbeitspaket 2 des PEPPOL Projekts entwickelt eine Lösung für den grenzüberschreitenden Datenaustausch von Geschäftsdokumenten zwischen den Akteuren im öffentlichen Beschaffungsprozess. Dies beinhaltet die Entwicklung eines gemeinsamen Datenformats, dem so genannten Virtual Company Dossier (VCD), sowie einem unterstützenden IT-System. Mit der Notwendigkeit einer interoperablen IT-Architektur stellt sich die Frage nach Technologien und Infrastruktur-Komponenten, die zu deren Umsetzung geeignet sind. Interoperabilität und Vernetzung von verteilten, heterogenen IT-Systemen sind zwei wichtige Anforderungen im Kontext der europäischen Beschaffung und des VCD. Service-orientierte Architekturen (SOA) scheinen einen vielversprechenden Ansatz für die Realisierung solch verteilter Architekturen zu liefern, da sie die lose Kopplung und Interoperabilität zwischen Systemkomponenten fördert. Diese Masterarbeit diskutiert daher das SOA-Konzept und die Fragestellung, wie seine Konzepte, Methoden und Technologien für die Entwicklung von interoperablen IT-Systemen für die elektronische Vergabe öffentlicher Aufträge verwendet werden können. Diese Diskussion wird durch eine praktische Anwendung des diskutierten service-orientierten Ansatzes erweitert, in dem die Konzeption und prototypische Implementierung eines Teilsystems der VCD Domäne durchgeführt wird. Zu diesem Zweck werden wichtige Aspekte bezüglich Interoperabilität und damit verbundener Standards und Technologien untersucht und im Kontext der öffentlichen Auftragsvergabe betrachtet. Darüber hinaus wird der SOA Ansatz diskutiert und ein top-down-Ansatz für die Entwicklung service-orientierter Systeme abgeleitet. Im praktischen Teil wird dieser top-down-Ansatz angewandt, einschließlich der Analyse und Modellierung von Komponenten der Zieldomäne. In einem weiteren

Schritt wird die Installation der technischen Infrastruktur und die Implementierung der entsprechenden Software-Anwendungen durchgeführt. Die abgeleiteten Ergebnisse sind in erster Linie für die laufenden Arbeiten des Projekts PEPPOL geeignet, können aber auf ähnliche Problemstellungen, die sich mit der Entwicklung von verteilten IT-Systemen beschäftigen, übertragen werden.

CONTENTS

List of Figures

LIST OF TABLES

List of Abbreviations

AI	Application Integration
AJAX	Asynchronous JavaScript and XML
B2BAI	Business to Business Application Integration
BII	Business Interoperability Interfaces
BPEL	Business Process Execution Language
BPEL4People	WS-BPEL Extension for People
BPM	Business Process Management
BPMN	Business Process Modeling Notation
CEN/ISSS	European Committee for Standardization/Information Society Standardisation System
CEN/ISSS WS/BII	CEN/ISSS workshop on Business Interoperability Interfaces
CIP	Competitiveness and Innovation Framework Programme
CPA	Collaboration Protocol Agreement
CPP	Collaboration Protocol Profile
DPS	Dynamic Purchasing System
EA	Enterprise Architecture
EAI	Enterprise Application Integration
ebXML	Electronic Business Extensible Markup Language
EDI	Electronic Data Interchange
EIF	European Interoperability Framework
EIS	European Interoperability Strategy
EJB	Enterprise Java Beans
EPC	Event-driven Process Chains
ESB	Enterprise Service Bus
ESP	European Service Provider
EU	European Union
FTP	File Transfer Protocol

G2B	Government to Business
G2C	Government to Citizen
G2G	Government to Government
GUI	Graphical User Interface
GWB	Gesetz gegen Wettbewerbsbeschränkungen
HTTP	Hypertext Transfer Protocol
IT	Information Technology
HTML	Hypertext Markup Language
ICT	Information and Communication Technology
IDABC	The Interoperable Delivery of European eGovernment Services to public Administrations, Businesses and Citizens
JAX-WS	Java API for XML Web Services
JEE	Java Platform, Enterprise Edition
JSF	JavaServer Faces
JPA	Java Persistence API
NSP	National Service Provider
OASIS	Organization for the Advancement of Structured Information Standards
OJEU	Official Journal of the European Union
OMG	Object Management Group
PDF	Portable Document Format
PEDRI	PEPPOL e-business document transport infrastructure
PEGS	Pan-European eGovernment Services
PEPPOL	Pan European Public Procurement OnLine
SA	Software Architecture
SektVO	Sektorenverordnung
SOA	Service-oriented Architecture
SoaML	Service oriented architecture Modeling Language
SOAP	Simple Object Access Protocol
TED	Tenders Electronic Daily
UBL	Universal Business Language

UDDI . . . . Universal Description, Discovery and Integration

UML . . . . Unified Modeling Language

UN/CEFACT . . . . United Nations Centre for Trade Facilitation and Electronic Business

VCD . . . . Virtual Company Dossier

VgV . . . . Vergabeverordnung

VOB/A . . . . Vergabe- und Vertragsordnung für Bauleistungen

VOF . . . . Verdingungsordnung für freiberufliche Leistungen

VOL/A . . . . Verdingungsordnung für Leistungen

W3C . . . . World Wide Web Consortium

WS-BPEL . . . . Web Services Business Process Execution Language

WSDL . . . . Web Service Description Language

XML . . . . Extensible Markup Language

XSL . . . . Extensible Stylesheet Language

XSLT . . . . XSL Transformation

Chapter 1

INTRODUCTION

The modernization of the European government sector through a sophisticated use of information and communication technology (ICT) for the provision of Pan-European eGovernment Services (PEGS) is an important target of the European Union in order to enhance government services provided to both citizens and enterprises throughout the European single market. With governments being the largest buyer in the EU, the electronic support of pan-European public procurement procedures is of great interest to increase "Europe's competitiveness" and create "new opportunities for EU businesses" [European Commission, 2009a].

Driven by the opportunities arising from the electronic support of procurement processes, such as cost savings as well as improved competition and efficiency, the European Commission promotes further development of public electronic procurement in the European Union. With the Directives 2004/17/EC and 2004/18/EC, the legal framework for public procurement has been modernized in 2004, which also includes regulations on the usage of electronic means in procurement procedures. Accompanied by an action plan for the implementation of electronic procurement, the Commission provides a framework for electronic procurement in the EU that is conducted in an open, transparent and non-discriminatory way [European Commission, 2004]. Furthermore, several initiatives funded by the European Commission are concerned with the development of frameworks, standards and pilot solutions to achieve interoperability between national procurement systems. The project Pan European Public Procurement OnLine (PEPPOL) is one of these initiatives targeting the development of an interoperable pan-European e-procurement solution. This shall support enterprises and public purchasing bodies from different member states to conduct their procurement processes electronically.

1.1 Problem Scope

A key aspect of the PEPPOL project is to improve the electronic exchange of documents within the procurement processes between enterprises and government agencies from different EU member states. For this purpose, work package 2 of the PEPPOL project deals with the development of a Virtual Company Dossier (VCD) that serves as an electronic container for documents being exchanged within the awarding process of public contracts. Companies participating in this process are obliged to provide various attestations and certifications in order to prove their compliance with certain qualification and selection criteria defined by public procurement agencies. The Directive 2004/18/EC provides a legal framework at European level defining a number of such criteria. This directive has been implemented by the different member states in accordance with their national legislation. As a result, differences between individual member states both in terms of required certificates as well as their national public procurement systems exist. For this reason, there is a need to create a unified system that integrates the diverse national systems at European level to achieve cross-border interoperability. Besides the VCD, which provides a common data format for electronic document exchange, the overall VCD system has to identify and specify a technical infrastructure consisting of several technical and software components that support companies and public agencies to conduct their procurement procedures electronically and to achieve pan-European interoperability.

1.2 Objectives

The overall objective of this master thesis is to examine and analyze the suitability of Service-oriented Architecture (SOA) for the previously described problem domain. On the one hand, this requires an analysis of relevant aspects related to the public procurement domain and interoperability. On the other hand, the underlying approach of SOA as well as related standards, methodologies and technologies have to be examined. A sub-system of the aforementioned VCD system will be identified, for which the service-oriented approach discussed in the foregoing theoretical analysis is applied in a practical context. The main outcome will be the identification of a set of modeling notations, technical standards, implementation technologies and infrastructure components suitable for the development of a service-oriented architecture and its transfer to the VCD domain. The conceptualization and development of a prototype for the identified sub-system following the service-oriented approach will serve as the proof of concept.

1.3 Structure

The content of this text is grouped into two parts: **Part I** (chapters 2-5) provides the theoretical foundation covering important and relevant topics for this master thesis. Starting point is Chapter 2 which will provide a discussion on electronic procurement and its role for electronic government. The main focus is on the legislative framework of public E-Procurement in the European Union and the procedures of public contract awarding in particular. With interoperability being a major requirement in the context of public E-Procurement, chapter 3 will examine important interoperability issues and deals with important standards as well as technologies and describes relevant activities in the field of European E-Government and electronic procurement. Chapter 4 comprises the main part of this master thesis. Besides a general introduction to provide a basic understanding of service-oriented architectures, this chapter will identify and describe important tasks of the design and implementation of SOA solutions while taking relevant standards into consideration. A discussion about benefits, challenges and the role of SOA in the context of public electronic procurement will complete this chapter. Chapter 5 provides important and relevant information about the PEPPOL project and work package 2. It is intended to be the transition to the practical part, which is conducted in the context of work package 2. **Part II** (chapters 6-10) describes the practical work conducted in the course of this master thesis. Chapter 6 first introduces the project scope, implementation objectives and describes the implementation methodology. Chapter 7 describes the evaluation and selection of a set of implementation technologies, tools and technical components used for the development of the prototype. Chapters 8 and 9 describe the main outcomes of the analysis and design as well as the implementation phase. A critical review, recommendations for further work and an examination of related work are part of chapter 10.

Part I

Theoretical Foundation

Chapter 2

ELECTRONIC PUBLIC PROCUREMENT

The aim of this chapter is first of all to provide an introduction about electronic public procurement. This will comprise a definition of the term itself as well as a description of important aspects of electronic procurement. Furthermore, it will highlight relevant parts of the European policy program that aims at facilitating cross-border electronic procurement within the European single market. As public procurement in the European Union is regulated by a legislative framework, this chapter will furthermore discuss important legal issues. Finally, different public procurement procedures and the use of electronic means within these will be examined.

2.1 Introduction

2.1.1 Defining Electronic Public Procurement

The following definition of the term *Electronic Public Procurement* will be conducted by stepwise defining its three elements. First of all, **procurement** can be defined as one of the core functions of an enterprise comprising

> *all measures to supply a company with production factors that are not created by the company itself.* [Grün et al., 2009, page 90][1]

Procurement accordingly means the purchasing of production factors including any material, goods or services from suppliers. Concerning to Stoll [2007], procurement is part of the enterprises' supply system. Purchasing in this supply system includes several activities like demand planning, tendering, selection of suppliers, contract awarding, ordering, distribution and payment.

[1] Own translation from German.

This definition can now be extended by the second element: **Public procurement** accordingly is

> *"the supply chain system for the acquisition of all necessary goods, works and services by the state and its organs when acting in pursuit of public interest."* [Bovis, 2005, page 14]

This illustrates that the core procurement activities are also applied in the public sector where governmental institutions purchase goods, works or services from suppliers.

By taking the third element into consideration, the term **electronic procurement** can be defined as

> *"the connective processes between companies and suppliers that are enabled by electronic communication networks."* [Meier and Stormer, 2009, page 50]

Correspondingly, this means that the aforementioned activities between buyers and suppliers are completed by using electronic means (e.g. Internet, personal computer, software, etc.). Hence, the relevant activities depicted in figure 2.1 comprise the electronically supported public procurement process [European Dynamics S.A., 2004, 2007a].

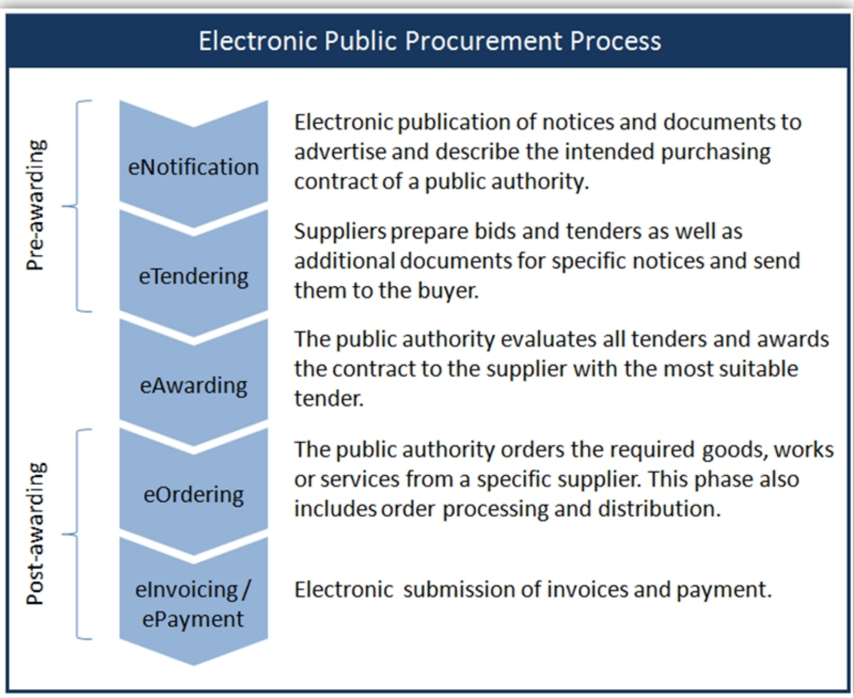

Figure 2.1: Activities in electronic public procurement

The activities completed before the awarding phase are referred to as *pre-awarding* phases, those completed thereafter *post-awarding*.

2.1.2 Main Drivers of E-Procurement

Several issues drive the implementation of electronic procurement. This section will therefor focus on the question why companies and public authorities are concerned with electronically supporting their procurement activities.

Basically, the drivers for conducting procurement electronically is increasing efficiency and effectiveness of purchasing activities. This includes on the one hand the reduction of costs (e.g. process, administrative and transaction costs) and on the other hand increasing the quality of the procurement process by avoiding the traditional paper-based manual processing which is time-consuming, cost-intensive and error-prone. Many activities can be performed faster or even can be automated by using electronic means, such as the electronic exchange and processing of orders and invoices. This also lowers the administrative efforts and reduces errors as data can be processed easier with support of suitable software (cf. [Baily et al., 2008; Stoll, 2007]).

Further reduction of transaction costs as well as process automation can be achieved by connecting the information systems of buyers with those of suppliers. These connections can vary in their intensity depending on the frequency of purchasing or the desired degree of process automation. In loose connections a buyer can simply purchase goods via online product catalogs or shops. A more sophisticated use of information technology may include the electronic exchange of information (e.g. order information, electronic invoices, etc.) by integrating the information systems (cf. [Schubert, 2002] and [Chaffey, 2002, chapter 7]).[2]

Considering the public sector, additional political, legal, social and economic factors affect the implementation of electronic procurement (see [Thai, 2009]). Hence enabling and modernizing cross-border and barrier-free procurement supported by electronic means within the single market of the EU shall increase "Europe's competitiveness" and create "new opportunities for EU businesses" [European Commission, 2009a].

Driven by the aforementioned benefits, a policy program targeting the usage and successful implementation of Information and Communication Technology (ICT) in public services and particularly in public procurement has been developed by the European Commission. This ICT strategy will be part of the next section.

---

[2] This is referred to as Enterprise Application Integration. Important aspects will be further discussed in section 3.1.3 on page 22.

2.1.3 E-Procurement within the European E-Government Strategies

The implementation of electronic public procurement in the European Union is part of an overall strategy concerning the use of Information and Communication Technology (ICT) in government and public services (i.e. electronic government):

> *"eGovernment is about using the tools and systems made possible by Information and Communication Technologies (ICTs) to provide better public services to citizens and businesses. [...] Implemented well, eGovernment enables all citizens, enterprises and organisations to carry out their business with government more easily, more quickly and at lower cost."* [European Commission, 2010a]

These objectives are incorporated into the EU policy framework *i2010 eGovernment Action Plan* to modernize the European public sector. One major target of the action plan is the implementation of electronic public procurement in order to reduce overall costs, improve efficiency and remove trade barriers. The overall target aims at making 100% of procurement activities electronically available within Member States by 2010 [European Commission, 2009a; European Communities, 2006].

As for the overall E-Government agenda, the strategy for electronic procurement is also described in an action plan, which is topic of section 2.1.4 on the next page. The Commission funds several initiatives as well as research and implementation projects that contribute to the electronic public procurement strategy. The following list introduces three initiatives and activities that are relevant for this master thesis as they deal with certain aspects that will be recaptured and further discussed later [European Commission, 2008, 2009b]:

- The **IDABC** programme[3] is concerned with achieving interoperability in the delivery of pan-European E-Government services including electronic public procurement by developing framework specifications and guidelines. The *European Interoperability Framework (EIF)* is part of a set of activities within the IDABC programme that provides recommendations for the implementation of electronic public procurement within the EU.
- The **CIP ICT Policy Support Programme**[4] is concerned with implementing eGovernment solutions to support the eGovernment ICT policy of the i2010 action plan and

[3] The Interoperable Delivery of European eGovernment Services to public Administrations, Businesses and Citizens (IDABC). Website: http://ec.europa.eu/idabc/en/home

[4] Competitiveness and Innovation Framework Programme (CIP). Website: http://ec.europa.eu/cip/ict-psp/index_en.htm

funds several projects that aim at developing pilot solutions. One of these is the *PEPPOL* project[5] that is concerned with developing a pilot solution for European cross-border electronic procurement.

- The **CEN/ISSS workshop on Business Interoperability Interfaces (CEN/ISSS WS/BII)**[6] aims at developing a framework for interoperability for electronic transactions in European cross-border public procurement. The *Business Interoperability Interfaces (BII)* provide a set of specifications of business transactions together with associated data models that occur in electronic procurement.

2.1.4 The European Action Plan on E-Procurement

As mentioned in the previous section, the *E-Procurement Action Plan* proposed by the Commission of the EU specifies the strategy for implementing electronic procurement. It addresses several important challenges related to electronic public procurement that have to be realized and compliant with the European Procurement Directives [European Commission, 2004].[7]

The objectives pursued by the action plan first of all concern the basic principles of non-discrimination, equal treatment and transparency. Put simply, this means that the implementation of electronic public procurement in the European Union shall lead to solutions that are generally available, interoperable and barrier-free leading to an increased and fairer competition in the European market. Additionally, potentials of electronic procurement shall be achieved leading to and increased efficiency through automation, accelerated procurement procedures and cost reductions.

To achieve these objectives in the Internal Market it is necessary that Member States implement the Directives correctly in order to avoid market fragmentation and barriers in cross-border procurement. This includes that appropriate electronic tools are used throughout the stages of the procurements process. An example in this context is the electronic submission and publication of notices to the *Official Journal of the European Union (OJEU)* in order to reduce costs and avoid publication delays and errors of the traditional paper based publication. In case of the electronic submission of official documents, electronic signatures and qualified electronic certificates have to be used and accepted by Member States. Another important issue addressed by the action plan is standardization and interoperability: In order

[5] Pan European Public Procurement OnLine (PEPPOL). Website: http://www.peppol.eu/

[6] European Committee for Standardization/Information Society Standardisation System (CEN/ISSS). Website: http://www.cen.eu/cen/Sectors/Sectors/ISSS/Activity/Pages/Ws_BII.aspx

[7] The legal framework will be discussed in more detail in the following section 2.2 on the following page.

to conduct procurement activities across Europe, it is necessary that national procurement systems are compatible with each other on technical, organizational and legal levels.

2.2 Legislative Framework of Public Procurement

This section will now introduce the legal framework of procurement in the public sector by describing the European procurement legislation as well as general principles and rules applied in public procurement.

Part Three of the Treaty on the Functioning of the European Union (formerly Treaty establishing the European Community) defines, among other things, the establishment as well as general principles (e.g. free movement of goods, capital, services and labor) of the European internal market [European Union, 2008]. The completion of the European single market includes the development of transnational business within the European Union (see [European Commission, 2010b]). Hence, this Treaty is the foundation for the regulation of public procurement within member states of the EU. Based on the Treaty's fundamental principles, the two Public Procurement Directives specifically define rules concerning procurement procedures and the award of public contracts (compare [Directive 2004/17/EC, 2004, recital 1 et seq.] and [Directive 2004/18/EC, 2004, recital 1 et seq.]). An overview on these two directives will be described in the following section. The main focus will be on Directive 2004/18/EC due to its relevance in the context of this master thesis (see section 1.1 on page 1).

2.2.1 European Public Procurement Directives

Originally, the rules for European public procurement have been defined by four directives. In order to modernize, simplify and harmonize the existing public procurement legislation, a revision came into force in 2004. This revision merges the former directives into two new directives:[8]

1. Public Sector Directive 2004/18/EC for public works contracts, public supply contracts and public service contracts, replacing the Public Supplies Directive (93/36/EEC), Public Works Directive (93/37/EEC) and Public Services Directive (92/50/EEC),

2. Utilities Directive 2004/17/EC on the sectors of water, energy, transport and postal services, replacing the Utilities Directive (93/38/EEC).

[8] Detailed information about these directives and their development can be found in [Bovis, 2005].

While Directive 2004/17/EC defines rules for procurement procedures in the water, energy, transport and postal services sectors, Directive 2004/18/EC concentrates on the rules for the award of public works, supply and service contracts. Concerning the intended modernization, the directive defines a new awarding procedure, the so called competitive dialog and furthermore promotes the use of information technology in electronic procurement procedures.

All Member States had to implement the Directives according to national legal constraints by January 31st 2006 [Directive 2004/18/EC, 2004, Article 80]. The German legislation of the awarding of public contracts is currently defined by several laws and regulations as depicted in figure 2.2 [Leinemann, 2010].

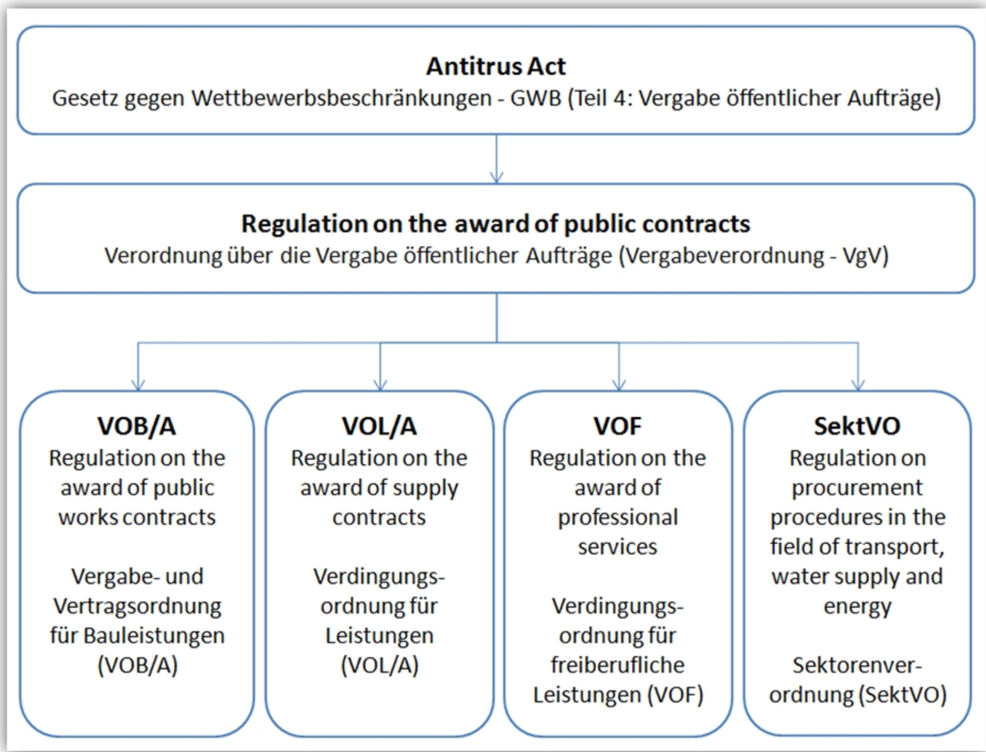

Figure 2.2: Current German legislation of the awarding of public contracts

The arrows illustrate the hierarchical relationship between the different laws and regulations. The German antitrust act (Gesetz gegen Wettbewerbsbeschränkungen (GWB)) contains overall principles and definitions concerning public procurement. § 97 (6) enables the government to further specify certain aspects such as awarding procedures in another enactment, which is realized with the regulation on the award of public contracts (Vergabeverordnung (VgV)). This in turn refers in §§ 4 - 6 to the three regulations on the award of public works contracts (Vergabe- und Vertragsordnung für Bauleistungen (VOB/A)), on the award of supply contracts (Verdingungsordnung für Leistungen (VOL/A)) and on the award of professional services (Verdingungsordnung für freiberufliche Leistungen (VOF)), which define rules for different works, supply and services contracts. The regulation on procurement procedures in the field of transport, water supply and energy (Sektorenverordnung (SektVO)) additionally defines rules for public contracts in the specific sectors of water, energy and transportation (the so called "Sektorentätigkeit").

In order to provide an overview about the important and relevant aspects of the European directive, the following sections will deal with some provisions and rules in more detail.

2.2.2 Public Contracts

As mentioned before the topic of interest of the Public Sector Directive is the regulation of the awarding of public contracts. It only applies to contracts that exceed certain thresholds as defined in Article 7, which are revised every two years by the Commission [Bovis, 2005]. Public contracts that are subject of the Directive are generally defined as contracts between economic operators and contracting authorities. The objective of such contracts is either the execution of work, the supply of products or the provision of services in return for a payment. As the definition is very general, Annex I and II further specify the characteristics of contracts being classified as public contracts.

2.2.3 Contracting Authorities and Economic Operators

As previously mentioned, the Public Sector Directive defines two main parties that are involved in the awarding of public contracts: *Economic operators* representing the supply side and *contracting authorities* representing the buyer side.

With respect to simplification, the term economic operator is used to cover every supplier or service provider that offers the execution of work, products or services on the market. An economic operator which was invited by a contracting authority to submit a tender is called a *candidate*, one that already submitted a tender is called *tenderer* [Directive 2004/18/EC, 2004, Article I (8)].

Contracting authorities can be any governmental procurement entity (e.g. procurement offices), regional or local authorities (e.g. municipal authorities) as well as bodies governed by public law. The latter one is used as a general clause for entities that are influenced by the government, that have a legal personality or that are financed by the state or other public authorities (see [Directive 2004/18/EC, 2004, Article I (9)] and [Arrowsmith, 2009, p. 260]).

2.3 Awarding of Contracts in Public Procurement

2.3.1 Fundamental Principles of Contract Awarding

The awarding of public contracts is bound to general principles. Article 2 defines the principles of non-discrimination, equal treatment and transparency. Non-discrimination and equal treatment refer to the obligation that all economic operators participating in the awarding of public contracts shall have the same chances. This means for example that a contracting authority has to act on the basis of defined selection and qualification criteria which have to be applied equally to all participating economic operators (see following section). The principle of transparency is achieved by provisions on the publication of notices, through which the awarding of public contracts above certain financial thresholds is advertised (see Articles 35 et seq.) [Arrowsmith, 2009; Bovis, 2007]. Further details about the qualitative selection criteria and publication of notices will be described later in this section.

For the awarding of public contracts above the financial thresholds, contracting authorities have to use specific award procedures. As the Directive is concerned with the award of public contracts, the regulations cover the pre-awarding phases of the procurement process (see section 2.1 on page 6). In general the so called *Open Procedure* or the *Restricted Procedure* have to be applied. Under specific circumstances contracting authorities may also use the *Competitive Dialogue* or the *Negotiated Procedure* [Directive 2004/18/EC, 2004, Article 28]. The following sections will first introduce these four types of awarding procedures as defined by the Procurement Directive. Afterwards, certain aspects will be described in more detail.

2.3.2 Different Types of Award Procedures

Open Procedure Public contracts awarded in an open procedure are addressed to an unlimited number of economic operators. The procedure's starting point is the publication of a contract notice in the Official Journal of the European Union by a contracting authority (notification phase). Any interested economic operator may then prepare and submit a tender regarding the specifications of the contract notice and contract documents (tendering phase). After the deadline to submit tenders is reached the contracting authority evaluates and checks all tenders and finally awards the contract to the selected economic operator (awarding phase) [Arrowsmith, 2009; Directive 2004/18/EC, 2004].

Restricted Procedures are similar to open procedures except that only a limited number of economic operators is invited to submit a tender. Any interested economic operator may request to participate but only those invited by the contracting authority may submit a tender.

For this first selection the contracting authority applies selection criteria as indicated in the contract notice. The number of invited candidates generally shall be "sufficient to ensure genuine competition", the minimum number shall be at least five (see [Directive 2004/18/EC, 2004, Article 44(3)] and [Arrowsmith, 2009]).

The **Competitive Dialogue** has been introduced in 2004 with the new Public Sector Directive. The competitive dialog can only be used in specific cases (see [Directive 2004/18/EC, 2004, Article 29]), for example for complex contracts, in which the contracting authority has difficulties in defining technical, financial or legal aspects (e.g. for an information technology system). This procedure also begins with the publication of a contract notice, which contains information about the needs and requirements. After selecting candidates that have requested to participate the contracting authority conducts discussions with invited candidates through dialogs in order to develop solution alternatives that meet the requirements. In the last stage, candidates submit their final tender and the contracting authority selects the award winner [Arrowsmith, 2009].

Negotiated Procedures may be used by contracting authorities in special cases (see [Directive 2004/18/EC, 2004, Articles 30, 31]) to negotiate the terms of a contract with economic operators. There are two types of negotiated procedures, one with publication of a contract notice and one without. In the first case, after the publication of a contract notice, the contracting authority invites economic operators, which have requested to participate, to negotiations and finally selects the award winner. As in the other procedures a candidate is selected according to the selection criteria as defined in the contract notice. In the specific cases of Article 31 a contract authority may renounce to publish a contract notice and only negotiate with a selected economic operator [Arrowsmith, 2009].

2.3.3 Publication of Contract Information

As means of advertising the procurement activities of contracting authorities four types of notifications may be published in the *Official Journal of the European Union (OJEU)* respectively its supplement for public procurement notices [Bovis, 2007].

The *Contract Notice* serves as an advertisement to announce a procurement project and is published by a publication body like the Official Journal of the European Union (OJEU). It contains general information about the contract and specifies the criteria of qualitative selection as well as award criteria that will be applied by the contracting authority.[9] The *Call for Tender* relates to a specific Contract Notice and serves as an invitation for an economic

[9] See next section for more information about these criteria.

operator to participate in the awarding procedure. Additionally to the information from the Contract Notice it contains further technical or legal contract documents. Once a contract has been awarded a *Contract Award Notice* containing information about the selected tenderer and the reason for its selection is published. The third type of notice is a *prior information notice* which contains estimations about the total value of contracts intended to be awarded by a contracting authority during a specific time period[10]

2.3.4 Criteria for Qualitative Selection

The "criteria for qualitative selection" are defined in Articles 45-50 of the Public Sector Directive. Economic operators that do not meet these legal and professional requirements are excluded from the awarding of a public contract. These criteria include for example requirements on the personal situation of an economic operator such as not being convicted by legal judgment as well as its technical and professional ability to pursue the work and its financial standing. Every economic operator participating in an awarding procedure has to provide the corresponding official documents and attestations for the aforementioned criteria. According to Article 51, economic operators may be asked to provide additional documents or information to supplement or clarify the submitted certificates [Frenz, 2007].[11]

In order to support contracting authorities in the evaluation of these selection criteria, member states may establish official lists of approved economic operators or certification of economic operators by certification bodies (Article 52 (1)). Such lists or certificates prove that economic operators meet the criteria defined in Articles 45-50 [Bovis, 2007].

The aforementioned criteria are used for a preselection of economic operators within the award procedure. The final awarding of a contract is based on certain award criteria defined in Article 53 of Directive 2004/18/EC. Accordingly, a contracting authority may either choose the cheapest or the economically most advantageous tender. In case of the latter one, each tender will be evaluated on the basis of specific weighted criteria (e.g. quality, price, costs) [Frenz, 2007].

[10] Annex VII A of the Public Sector Directive contains further information about the contents of the different notices.

[11] See pages 867 et seq. for more information of the different selection criteria as well as the corresponding certificates and attestations.

2.3.5 Electronic Means Used in the Pre-Award Procedures

Electronic means can be generally understood as the use of information technology (e.g. personal computer, software, Internet) to store, process and transmit data (cf. [European Commission, 2005] and [Directive 2004/18/EC, 2004, Article 1 (13)]).

The main aspects of the usage of electronic means covered by the Directives concern on the one hand general obligations on electronic communication and exchange of information. On the other hand, the *Dynamic Purchasing System (DPS)* and *electronic auction* are introduced as specific electronic purchasing techniques. Additionally, several electronic tools exist that support the different phases of the awarding process. As this master thesis mainly focuses on the pre-awarding phases and the awarding itself, the following list contains important tools used in the notification, tendering and awarding phases.

eNotification refers to electronically supporting the first phase of the procurement process, in which contracting authorities advertise their purchasing activities by publishing contract notices. All notices and additional contract documents are published in the Official Journal of the European Union respectively its electronic version *Tenders Electronic Daily (TED)*.[12] To support the publication of notices in TED, two services have been established:[13] *eNotices* provides an online tool supporting the preparation of notices, *eSenders* allows direct submission of notices in a structured format such as XML [Anghelakis et al., 2007]. Additionally to TED, equivalent national procurement platforms exist in different member states, for example the German platform *e-Vergabe* through which the submission of tenders can be carried out electronically.[14]

eCatalogues support the electronic tendering phase of the procurement process. Electronic catalogues basically are documents prepared by an economic operator describing its products and prices as well as additional tender information. Thus, eCatalogues may constitute tenders and support a contracting authority in the collection and evaluation during the pre-awarding phase. However, eCatalogues can furthermore be used to support the electronic ordering and invoicing during the post-awarding phases. To support electronic processing of eCatalogues, they can be provided in a standardized Extensible Markup Language (XML) based format such as Universal Business Language (UBL)[15] [European Commission, 2005; European Dynamics S.A., 2007a].

---

[12] Url: http://ted.europa.eu/

[13] See also: SIMAP-Portal, Url: http://simap.europa.eu/index_en.htm

[14] Url: http://www.evergabe-online.de/

[15] For more information on UBL see section 3.2.1 on page 25. Detailed information on eCatalogue standards can be found in [European Dynamics S.A., 2007b].

eCertificates or **eAttestations** refers on the one hand to the electronic submission of certificates and attestations by economic operators that contracting authorities use to evaluate the criteria of qualitative selection. On the other hand it includes the electronic issuing and collection of these attestations from authorized certification bodies. This is a very challenging aspect in European cross-border procurement as it is influenced by various legal, organizational and technical issues (e.g. mutual recognition of certificates, interoperable data formats, data security and protection) [16] [Siemens and Time.Lex, 2008].

Electronic auctions can be used in the awarding phase after candidates have already been selected. Throughout the auction, candidates are asked to submit values for defined criteria such as price or other quantifiable variables depending on the award criteria applied. Candidates are then automatically ranked based on their submitted values. The best tender in this final ranking will be the award winner [European Commission, 2005; European Dynamics S.A., 2005].

A **Dynamic Purchasing System (DPS)** may be established by contracting authorities for repetitive purchasing over a certain time period (usually four years). The awarding of contracts under this system is solely based on electronic means [Directive 2004/18/EC, 2004, Article 1(6), 33(2)]. Using a DPS leads to a simplified award procedure: On the one hand, contracting authorities only need to publish a simplified contract notice. On the other hand, the qualification of economic operators is omitted in case they are already registered in the DPS [European Commission, 2005; European Dynamics S.A., 2005].

2.4 Summary

Figure 2.3[17] visually summarizes the key aspects discussed in the previous sections. The implementation of the EU Procurement Directives in the different Member States lead to a heterogeneous legal and technical procurement environment in the Single Market.

Although the Directives aim at harmonizing public procurement in Europe by defining common rules for cross border procurement, the transformation into national law had to be in accordance with the legal systems and individual requirements of each Member State. Hence, achieving interoperability between the different national systems is a great challenge in realizing European-wide electronic procurement. As mentioned in section 2.1.4 a major issue in this context is the electronic exchange of attestations and certificates which economic

[16] Work package 2 of the PEPPOL project develops a digital document container, the so called Virtual Company Dossier (VCD), that enables the electronic collection and submission of certificates. Hence, chapter 5 on page 60 will provide further information.

[17] Own figure.

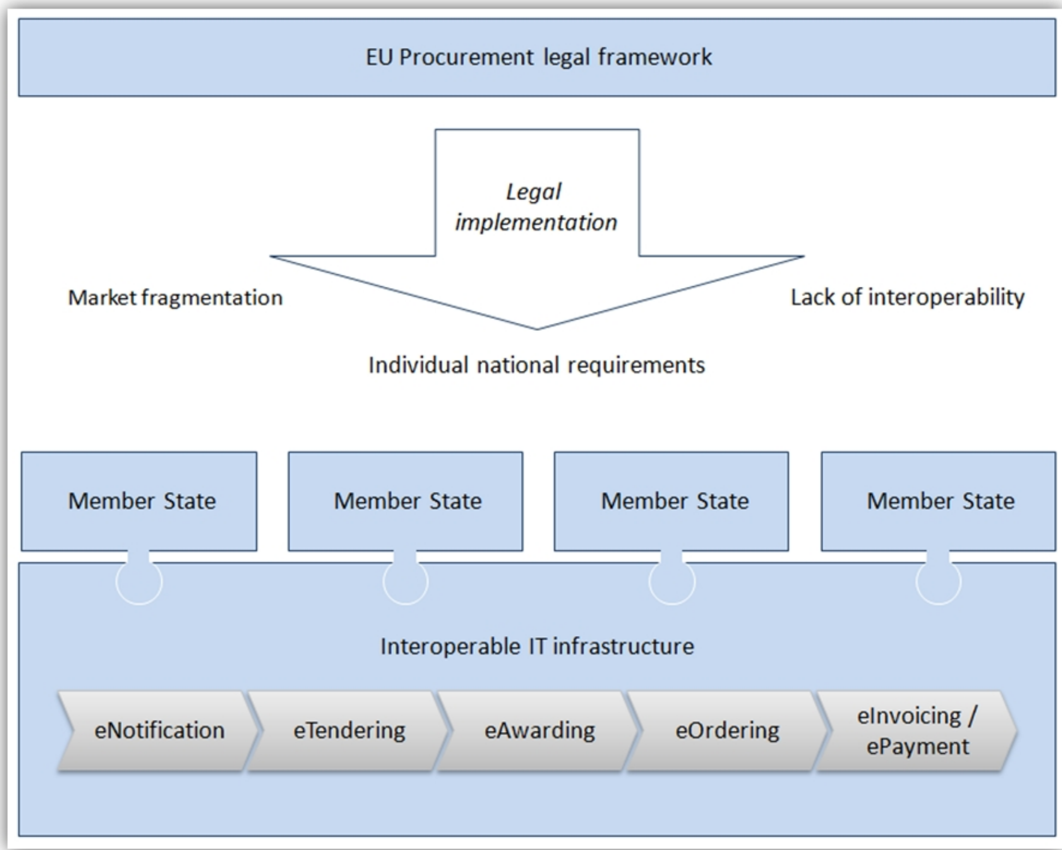

Figure 2.3: Implementation of the legal framework for public procurement

operators have to collect and submit within the tendering phase. In order to facilitate an interoperable IT infrastructure for electronic public procurement processes, the European Commission funds several initiatives and projects that are concerned with elaborating specifications and technical solutions. Section 2.1.3 introduced three important initiatives: The European Interoperability Framework (EIF), the CEN/ISSS workshop on Business Interoperability Interfaces (CEN/ISSS WS/BII) and the PEPPOL project.

Hence, the two main topics of interest for this master thesis are derived from the aforementioned discussion:

1. Interoperability in European electronic public procurement.
2. Electronic collection and submission of certificates.

The electronic collection and submission of certificates is subject of the development of the Virtual Company Dossier (VCD) within work package 2 of the PEPPOL project and will be part of chapter 5. Important issues about interoperability including the EIF and BII will be part of the next chapter.

Chapter 3

INTEROPERABILITY

As interoperability has been identified as a major topic relevant for this master thesis, this chapter will now focus on important related issues. Starting with a definition of the term, the following introduction will identify important aspects of interoperability and drivers for achieving it. Afterwards, several technologies and standards will be described that support the implementation of interoperable software systems. Finally, a discussion about two important initiatives driven by the European E-Government strategies that are concerned with the provision of so called Pan-European eGovernment Services (PEGS) and the standardization of business transactions within public procurement will be conducted.

3.1 Introduction

3.1.1 Defining Interoperability

At first it is necessary to highlight what the term *interoperability* means. When talking about interoperability of IT systems, the term should not be confused with Application Integration (AI). Differences and similarities between integration and interoperability will therefore be discussed in the following section.

Kosanke [2006] as well as Gottschalk and Solli-Sæther [2009] provide an overview of several definitions of *interoperability*. They all agree upon basic characteristics of interoperability, which is generally described as the ability of different ICT systems to communicate and interact with each other as well as to exchange information.

A suitable definition is provided by Sturm [2007]:

> *"Interoperability is the ability of independent, heterogeneous information and communication systems to co-operate seamlessly and without format discontinuities. Two inter-operable systems are also called compatible."* [Sturm, 2007, p. 123] [1]

Mertins et al. [2008] extend this rather technical view on the level of ICT systems by a more general organizational viewpoint. The authors state that interoperability is

> *the ability of a system or an organisation to work seamless with other systems or organisation without any special effort.* [Mertins et al., 2008, page v]

In the narrow sense, interoperability accordingly is a technical characteristic or ability of information systems to exchange information with each other seamlessly and not requiring any special efforts or adjustments of the involved systems. In a broader sense, interoperability also refers to the ability of organizations to co-operate, i.e. by means of inter-operable processes, business rules and agreements. A fundamental condition thereby is that the interconnected systems may maintain their independence from each other.

3.1.2 Levels of Interoperability

According to the common understanding, interoperability can generally be achieved on three levels (cf. [BMI, 2008; IDABC, 2004; Laskaridis et al., 2008; Mondorf and Wimmer, 2008; Sturm, 2007]): On the **technical level** it is necessary for the correct data interchange between systems that common protocols (e.g. SOAP, HTTP, FTP) and data formats (e.g. XML) are supported. The protocols on the one hand ensure that data can be exchanged over the network that links the systems (e.g. the Internet). Data formats provide a common syntax that both systems understand. Technical interoperability also refers to automatic invocation or sharing of specific software functionality between different software applications. On the **semantic level** it has to be ensured that the data being exchanged is understandable so that it can be processed correctly within different information systems. Hence, a commonly defined semantic is required additionally to a common syntax in order to specify the meaning of the

[1] Own translation from German: "Interoperabilität ist die Fähigkeit von unabhängigen, heterogenen Informations- und Kommunikationssystemen zur möglichst nahtlosen, medienbruchfreien Zusammenarbeit. Zwei miteinander interoperable Systeme werden auch als kompatibel bezeichnet."

data. Finally, on the **organizational level** business goals and processes have to be interoperable between entities (e.g. enterprises, government agencies) so that information can be exchanged, understood and used within business processes across organizational boundaries.

According to IDABC [2008], the organizational level can be extended by the **legal level**, especially in the government context. Legal interoperability refers to the ability of legal systems of different countries (e.g. Member States of the EU) to inter-operate. One example in this context is the Public Procurement Directive 2004/18/EC that has been implemented according to national laws and requirements in each Member State leading to a heterogeneous legal environment for pan-European public procurement. As a result, achieving interoperability on this level between different national legislations requires, for instance, mutual legal recognition of information and documents being exchanged. All of these levels are embedded in the **political context**, which means that different Member States agree in their visions, intended goals and strategies to develop interoperable systems [IDABC, 2008]. The Procurement Directives and the E-Government action plan are examples for such common EU policies (cf. section 2 on page 5).

Figure 3.1 summarizes the aforementioned interoperability levels[2].

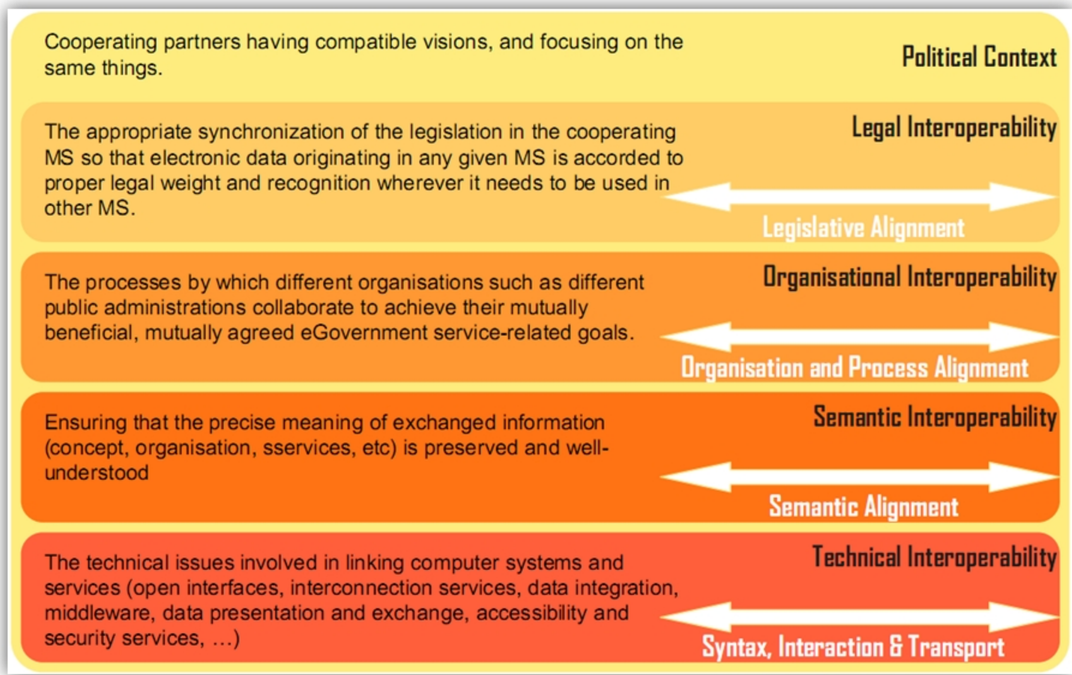

Figure 3.1: Interoperability levels of the European Interoperability Framework (EIF)

[2] Graphic taken from [IDABC, 2008, p. 32].

3.1.3 Interoperability, Integration and Coupling

As mentioned in the previous section, interoperability and integration should not be used synonymously. As stated in IDABC [2008], "Interoperability is not Integration, which is a means of changing loosely coupled systems to make them into more tightly coupled systems" [IDABC, 2008, p. 5]. This already clarifies the main difference: Integration of IT systems requires adjustments of the involved loosely coupled IT systems in order to create a more tightly coupled composite IT system. As the previous section showed, interoperability means the ability of IT systems to interact without such additional effort. However, integration also aims at connecting different IT systems or software applications (referred to as Application Integration (AI)), that need to be inter-operable. Similarly, Scholl and Klischewski [2007] state that integration is "the forming of a larger unit [...], temporary or permanent, for the purpose of merging processes and/or sharing information." [Scholl and Klischewski, 2007, p. 897], whereas interoperability is defined by the authors as the technical capability for the interoperation of independent or heterogeneous information systems that are controlled by different entities in a predefined way.

But what exactly is integration, and where are the connecting points to interoperability? According to the numerous definitions that exist, Application Integration can be described as follows (cf. BSI [2004]; Linthicum [2001, 2003]; Ruh et al. [2001]; Schubert [2003]): AI is a strategic approach to bind different heterogeneous application systems together so that information can be exchanged and shared seamlessly between them. Conducted internally within an organization, this is referred to as Enterprise Application Integration (EAI). External integration moreover aims at enabling efficient inter-operation and collaboration between several business partners, e.g. a company and its suppliers, customers and partners across organizational boundaries, which is referred to as Business to Business Application Integration (B2BAI). Hence, achieving interoperability on the technical as well as semantic and organizational levels is an integral part of Application Integration. Whereas interoperability enables the inter-operation of independent software systems, integration leads to composite systems with stronger dependencies and inter-relations.

The term coupling has already been used at the beginning of this section. In the context of AI it "defines the degree of integration" and "measures the level of interdependency" of the connected software and information systems, ranging from loose to tight coupling [Ruh et al., 2001, p. 21]. Tight coupling of different software applications leads to a composite overall software system with strong interdependencies. Loose coupling of software systems may increase flexibility and autonomy of the involved systems as changes in one system have a minor impact on the connected systems. On the other hand, loose coupling may require the usage of additional software components that mediate and establish communication links between the connected software components (so called middleware). This may increase maintenance and implementation efforts as well as costs for the technical infrastructure. Hence, finding an optimal balance between loose and tight coupling is an important issue in Application Integration projects (cf. Roshen [2009]; Schwinn and Winter [2007]).

Summing up, Application Integration and especially B2BAI is a rather holistic strategic and operational approach to bind several existing information systems and business processes together, thus forming a more tightly coupled consolidated system, which requires adjustments and refinements of systems in order to make them interoperable. Two application systems that are already interoperable usually do not require greater additional integration effort. Furthermore, the degree of integration and interdependencies between different software applications may vary from tight to loose coupling. Concerning the technical and semantic level (data exchange, data formats, software applications, etc.) several standards and technologies exist that support interoperability and integration of software applications, some of them promoting loose coupling of information systems. Some important of these standards will be discussed in section 3.2 on the following page.

3.1.4 Drivers for Achieving Interoperability

The overall driver for achieving interoperability is to ensure efficient collaboration among organizations or business partners, both within the enterprise and government sectors. A major challenge arises from the fact that different IT systems are characterized by heterogeneity because they have been developed to match individual requirements of an organization or specific functional domains. Efficient collaboration between organizations requires connectivity of distributed heterogeneous information systems and seamless information flow between them [Kalfoglou, 2010; Lam and Shankararaman, 2007].

Application Integration is an approach to bind existing stand-alone information systems together internally (EAI) as well as externally (B2BAI), which requires interoperability on the previously described levels. This enables companies to provide for example increased customer services as well es efficient collaborations with business partners. By automating interconnected business processes, processing times, processing errors and costs can be reduced. Through automated data exchange it is furthermore possible to realize a consolidate access to information that can be used and processed throughout the connected information systems (cf. [Lam and Shankararaman, 2007; Schwinn and Winter, 2007]).

The drivers for achieving interoperability in the government sector are quite similar to those of the enterprise sector, aiming at realizing the same benefits such as increasing efficiency of service delivery, speeding up processes and reducing costs through interoperable processes and information systems [Sturm, 2007]. In the EU, the need for interoperability is manifested in the European ICT-Strategy of developing efficient E-Government services (cf.

section 2.1.3 on page 8 and European Interoperability Framework (EIF) in section 3.3.1 on page 28). As stated in [Commission, 2003], interoperability is

> *"a fundamental requirement, from both economic and technical perspectives, for the development of efficient and effective eGovernment services at both the national and pan-European levels, including the regional and local ones."* [Commission, 2003, p. 5]

The development and provision of E-Government services covers all the relationships in which government entities interact with citizens (Government to Citizen (G2C)), businesses (Government to Business (G2B)) as well as other governmental agencies (Government to Government (G2G)), both at national and international levels. The aforementioned heterogeneity is a great challenge in these wide ranging relationships: Many differences between information systems as well as semantic and organizational disparities in terms of national laws, regulations, citizen services, processes and languages exist [Klischewski, 2003]. In the context of electronic public procurement, an interoperable E-Procurement infrastructure is expected to guarantee equal treatment, non-discrimination and transparency and facilitate fairer and more effective competition in the European single market. Currently, achieving these goals is challenging as E-Procurement infrastructures in the Member States of the EU are characterized by a high market fragmentation and heterogeneity [Mondorf and Wimmer, 2008].

Achieving interoperability on the five aforementioned levels is therefore essential to overcome this heterogeneity and fragmentation of national and international government systems, and E-Procurement systems in particular. Thus, sharing of information, cooperation among administrative processes as well as provision of government services can be ensured in the G2B, G2C and G2G sectors [Commission, 2003].

3.2 Interoperability Standards and Technology

Several technologies and standards that support and enable interoperability between organizations, business processes and the supporting software applications have been developed. Many of these standards and technology are used within EAI or B2BAI projects to achieve ex-post interoperability of existing software systems. However, some of these can be used to achieve ex-ante interoperability when developing ICT-systems. Due to the scope of this master thesis, this section will only deal with the most relevant standards and technologies that support inter-connecting software systems and data interchange. The underlying concept of electronic exchange of documents and information is Electronic Data Interchange (EDI), that refers to the exchange and processing of documents in a standardized format between IT systems [Zimmermann, 2007]. Some aspects concerning standardized data formats are relevant in this context and will be introduced in the following section.[3]

3.2.1 Data Formats

An important prerequisite for the exchange of data between different software applications is that it needs to be in an standardized and structured format that is commonly understood and can be automatically processed by software systems. Some of these data standards will be described in the following.

Extensible Markup Language (XML)

The Extensible Markup Language (XML) provides a standardized and platform-independent syntax to structure and exchange data between software applications. Due to its simplicity and flexibility, XML has become very popular and is a frequently accepted data format in the context of electronic data exchange between software applications (cf. [Ciganek et al., 2007; Linthicum, 2003; Ruh et al., 2001]). The basic syntactical component of an XML document is an *element* that represents a specific data field in the document and can be defined both in a flat and hierarchical structure. Additional metadata can be defined for an element by using *attributes*.

XML documents can be created and accessed within software applications by using *XML processors*, that provide means to validate and parse XML documents in order to edit or read the contained data elements. Processing XML documents may furthermore require transformation from one XML document to another. This is necessary especially if two applications need to exchange data that exist in different XML based data formats. XSL Transformation (XSLT) processors provide required means to transform XML documents to other XML documents or to create PDF or HTML documents (cf. [Harold and Means, 2007; Linthicum, 2003]).

Due to the popularity of XML, many other data formats that are based on XML have been developed by multinational consortia to provide exchangeable data formats and define commonly agreed semantics of XML data elements. Two of them, the Electronic Business Extensible Markup Language (ebXML) and Universal Business Language (UBL), will be described in the following.

[3] For further information on EDI see [Zimmermann, 2007] and [Stoll, 2007, chapter 4].

Electronic Business Extensible Markup Language (ebXML)

ebXML is more than just a data format. It is an initiative of the United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) and the Organization for the Advancement of Structured Information Standards (OASIS) to "provide an XML-based open technical framework to enable XML to be utilized in a consistent and uniform manner for the exchange of electronic business (eb) data in application to application, application to human, and human to application environment" [UN/CEFACT and OASIS, 2001c, p. 7]. ebXML provides a holistic framework to facilitate electronic data exchange between ebXML-compliant trading partners.

The standard defines three main components that address organizational and process issues between different trading partners: The so called *Collaboration Protocol Profile (CPP)*, *Collaboration Protocol Agreement (CPA)* and the *Business Process and Information Meta Model*. A CPP is a document that contains information about a company such as supported business processes and details about technological capabilities of exchanging business information. A CPA is the "intersection of two CPP's" and serves as an agreement for a specific business collaboration between several companies [UN/CEFACT and OASIS, 2001d, p. 16] [4]. The ebXML Business Process and Information Meta Model provides a modeling methodology that allows business partners to specify their collaboration scenarios and business transactions.

The *ebXML Core Components* represent information objects and are the basic building blocks of information used in business processes and can be expressed in an XML syntax. ebXML provides a hierarchical approach that aggregates several Core Components of a specific *Context* into a *Business Document* [UN/CEFACT and OASIS, 2001b,d] [5]. A concrete implementation of these core components is the Universal Business Language (UBL), which will be described in the following section.

Universal Business Language (UBL)

The UBL is an XML-based document standard developed by the Organization for the Advancement of Structured Information Standards (OASIS) and defines XML representations of business documents (e.g. purchase orders or invoices). Therefore, it provides data schemata and defines a data dictionary containing several data elements to facilitate a common syntax for electronically exchanged business documents. It is furthermore designed for usage within

[4] XML example documents for both CPP and CPA are available at http://www.ebxml.org/specs/cpp-example.xml and http://www.ebxml.org/specs/cpa-example.xml.

[5] A set of Core Components is provided in [UN/CEFACT and OASIS, 2001a].

a business framework such as ebXML and is an implementation of the previously mentioned ebXML Core Components Specification. The basic information components of the UBL are so called *Business Information Entities*, which comprise specific documents such as *Orders* or *Invoices* [OASIS, 2006].

UBL was developed to solve the disadvantages resulting from the existence of multiple industry-specific data formats. For instance, adapters and tools that support all the various formats are required in order to enable the electronic exchange of documents leading to greater efforts and higher implementation costs to conduct business relationships. UBL is designed to overcome these disadvantages by defining a common vocabulary and generic business documents that can be used in multiple business contexts and are not tied to specific industries or sectors.

3.2.2 Web Services

The World Wide Web Consortium (W3C) defines web services as "a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards." [W3C WSA Working Group, 2004a]. The web service technology provides platform independent means based on open standards to expose and use system functionality over the Internet. Furthermore it achieves interoperability and loose coupling between service providers and consumers [McGovern et al., 2006]. As web service are the core components of a Service-oriented Architecture (SOA), this topic will be recaptured and further discussed in chapter 4 on page 34. At this point it is sufficient to state that SOA and web services together enable loosely coupled connections between software systems and thus "achieve interoperability without forcing integration" [Gottschalk and Solli-Sæther, 2009, p. 139].

3.2.3 Middleware

If software applications are not able to communicate directly with each other, for example by using web service interfaces or a common data format, additional software is required that establishes connections among several software applications or databases. Such software is referred to as *middleware*, which is independent software that routes data from one application or database to another and is responsible for transforming different data formats where required [Ruh et al., 2001]. Concerning to this understanding, middleware may be any software establishing links between software systems. Hence, many different types of middleware exist that can be grouped into two main categories: *Point-to-point* middleware establishes connection between two applications, *many-to-many* middleware inter-connects several applications [Linthicum, 2003]. With the number of applications being linked together, the implementation effort of establishing point-to-point connections rises, as specific middleware components are required for every link. Many-to-many middleware reduces the number of single connections by providing a communication gateway to establish links between several applications. Due to the higher complexity, such middleware has to provide additional services to ensure that information can be exchanged correctly between target and destination applications. This includes for example handling and routing of messages from a source to the desired destination or transformation between different data formats if necessary [Ruh et al., 2001; Serain, 2002].

The need for more flexible and easier ways to inter-connect several software systems through middleware lead to the development of middleware types that promote loose coupling. Assembling a so called Enterprise Service Bus (ESB) is such an approach [Chappell, 2004], that provides a platform for connecting a large number of distributed software systems. Similar to the aforementioned middleware, an ESB also supports messaging, routing and data transformation. But it adds value by being a standards-based approach that includes web services and provides the implementation backbone for an SOA, thus promoting loose coupling of connected software systems.

3.3 Interoperability in European E-Government

After the discussion of important aspects of interoperability, this section will now focus on two concrete artifacts that have been elaborated within the European E-Government strategies to facilitate interoperability (cf. 2.1.3 on page 8): The European Interoperability Framework (EIF), which provides a multilateral European framework for interoperability, and the Business Interoperability Interfaces (BII), which provide specifications of business processes and information content for the awarding processes of European public procurement.

3.3.1 The European Interoperability Framework

Background The European Interoperability Framework (EIF) is being developed within the IDABC program. Its first version was published in 2004. Due to continuous refinements and extensions, a draft for version 2 of the EIF has been published for external review in 2008 [IDABC, 2004, 2008].

The EIF is one of a set of documents, studies and activities conducted within the IDABC program to support and establish interoperability, such as the European Interoperability Strategy (EIS), the specification of architecture guidelines and the promotion of open document formats.[6] As a multilateral European framework, the EIF is intended to specify policies, guidelines and recommendations for the delivery of E-Government services in the European Union, so called Pan-European eGovernment Services (PEGS). The provided guidelines and specifications aim at facilitating cross-border interoperability of PEGS within the European Union, between government entities (G2G) as well as between the government and citizens (G2C) or businesses (G2B). Member States should adapt the guidelines of the EIF for their national interoperability frameworks according to national requirements and circumstances in order to make them compatible, thus facilitating interoperability both on the national as well as the European level.

Proposed recommendations for the interoperability levels The EIF proposes several guidelines and recommendations for the implementation of PEGS in the EU according to each of the interoperability levels described in section 3.1.2 on page 20 (legal, organizational, semantic and technical interoperability) [IDABC, 2004, 2008].

On the **legal level**, achieving cross-border interoperability faces the challenge of heterogeneity and that PEGS provided in one country have to be compliant with legislations of other countries. European as well as national legislation therefore should support interoperability and provide a common legal basis accepted in all member states. Examples therefore are harmonizing the legislation of specific sectors (e.g. Public E-Procurement sector, cf. section 2.2.1 on page 10) or establishment of frameworks for specific domains or sectors that contain sets of the strictest legal provisions that are applicable across all member states.

The main interoperability aspect on the **organizational level** is the adjustment and harmonization of business processes between actors within and across country borders. To achieve interoperability on this level, cooperating partners should specify agreements on the interaction within business processes that have to be interconnected. Adjusting and harmonizing business processes should furthermore be realized by analyzing and establishing a common understanding of the relevant processes in order to decompose them and to identify common sub-processes that can be interconnected. Additionally, the EIF proposes agreements on necessary Business Interoperability Interfaces (BII) that support the inter-operation of business processes at pan-European level.[7]

[6] See http://ec.europa.eu/idabc/en/document/5317/5883 for a list of IDABC studies and documents.

[7] Such BII have been established by the European Committee for Standardization/Information Society Standardisation System (CEN/ISSS) to facilitate interoperability in public procurement processes, which will be discussed in section 3.3.2 on the next page.

The **semantic level** of interoperability refers to the establishment of a common understanding of the meaning of data being exchanged. As proposed in the EIF, this can be achieved by establishing sectoral communities, e.g. for the E-Procurement sector, that create and agree upon common sets of data structures, data elements as well as the corresponding meaning of information. Multilateral mapping tables can furthermore be used to transform certain data elements used in one domain or country to the corresponding data elements of other domains.

In order to resolve the issues of interoperability on the **technical level** like connecting software systems as well as data presentation and exchange, the recommendations of the EIF foresee a distinction between the presentation of data and the interconnection of software and data exchange between information systems. Data presentation should be realized through portals that are accessible to citizens and businesses over the Internet and provide an aggregated view on the data that may have been collected from several sources. An important aspect for data presentation is the design of user interfaces to enable accessibility, usability and multilingualism of portals. In order to collect and share data among several sources, software systems have to be interconnected and data interchange between applications has to be established. Correct exchange of data should be realized by using common XML or EDI based data formats. To establish links between loosely coupled software applications, appropriate middleware that connects and mediates between software systems may be used. Web services are furthermore a promising way to make distributed software applications interoperable and to share data and functionality between them.

3.3.2 Business Interoperability Interfaces

Background The Business Interoperability Interfaces (BII) have been developed by the European Committee for Standardization/Information Society Standardisation System (CEN/ ISSS), in particular by the workshop on Business Interoperability Interfaces (CEN/ISSS WS/BII). Driven by the existence of different standards for electronic information exchange in public procurement processes, the BII aim at supporting the implementation of interoperable public procurement systems in member states of the European Union by standardizing the electronic business transactions between business partners [CEN/ISSS WS/BII, 2009b]. The core elements are so called *profile descriptions*, each being a "specification of how one or more Business Processes are executed by specifying the business rules governing its business collaborations and the information content (data model) of the electronic business transactions exchanged" [CEN/ISSS WS/BII, 2009a, p. 5].

The overall goal is reducing implementation costs of procurement systems and minimizing the effort of concluding business agreements between trading partners. The profiles of the

BII act as agreements that organizations can implement in order to be compliant to them, thus minimizing the need for further business agreements. Compliance to a profile is thereby expressed by requirements for software solutions implementing a profile. Software solutions compliant to a profile therefor need to support the business rules, the business transactions as well as the data content specified by a profile [CEN/ISSS WS/BII, 2009c].

Profile descriptions As previously mentioned, each profile description consists of a specification focusing on the following aspects [CEN/ISSS WS/BII, 2009a]:[8]

- **Choreography of business processes**, specifying the roles of different participants, tasks and activities of a specific business collaboration.
- **Electronic business transactions**, specifying the sequence of electronic transactions (i.e. the exchange of business documents) that are part of a business process.
- **Business rules**, specifying rules and constraints that govern the execution of a business process, business transactions and information elements.
- **Information content**, specifying the content and data model of business documents exchanged in electronic business transactions.

It should be noted, that the specification primarily address the organizational and semantic interoperability levels. The specification of data models is syntax-neutral only focusing on defining necessary data elements as well as their meaning and cardinality that comprise a specific business document. However, the CEN/ISSS WS/BII also provides a specification of how specific data elements may be mapped to existing syntax standards such as UBL [CEN/ISSS WS/BII, 2009a].

The profile descriptions are aligned to the different awarding processes of public procurement as defined by the Public Procurement Directives and are separated into pre-award and post-award profiles [CEN/ISSS WS/BII, 2009c]. Each of these two main areas of procurement processes is further broken down into specific business processes that consist of different business collaborations. The corresponding profiles are then defined for each business collaboration.[9]

[8] All profile descriptions and corresponding transaction data models specified by CEN/ISSS WS/BII are available at http://spec.cenbii.eu/Profiles/IndexWG1.html.

[9] An overview on the hierarchical relationships is illustrated in [CEN/ISSS WS/BII, 2009c, pp. 16, 34].

3.4 Summary

This chapter defined the term interoperability and discussed relevant issues related to the role of interoperability in the enterprise and public sector as well as major technical aspects. Interoperability accordingly basically is a characteristic of software systems being able to exchange information without the need for further refinements. As expressed by the different interoperability levels, achieving interoperability requires that issues on the technical, semantic, organizational, legal and political levels have to be considered.

Interoperability on these levels is crucial in order to realize efficient and effective collaborations between business partners. In the public sector, interoperability is seen as a fundamental requirement for the development and provision of E-Government services in relationships between different government institutions (G2G) as well between government institutions and businesses (G2B) or citizens (G2C). A great challenge in the context of pan-European E-Procurement is the heterogeneity of national procurement systems throughout the European Union, which currently hinders achieving the goals of fairer competition, equal treatment, transparency and non-discrimination.

Enterprise Application Integration (EAI) and Business to Business Application Integration (B2BAI) have been introduced as holistic approaches to bind existing information systems together internally and externally in order to achieve ex-post interoperability. Both are related to a high effort as existing organizational structures, business processes and software applications have to be adjusted. Integration projects usually lead to tightly coupled composite IT systems of several collaborating business partners, resulting in a high degree of dependencies of the connected software systems.

Several technologies and standards exist that support interoperability of software systems, some of them promoting loose coupling. Applied in the development of IT system to achieve ex-ante interoperability, these technologies and standards help minimizing the need for further refinements and adjustments. Standardized data formats like XML or the XML based Universal Business Language (UBL) provide means for technical and semantic interoperability. The Electronic Business Extensible Markup Language (ebXML) furthermore provides a framework supporting technical, semantic and organizational interoperability between trading partners. Web services are a promising technology to exchange information and share functionality between different information systems based on open standards (HTTP, SOAP, WSDL). As the core building block of an Service-oriented Architecture (SOA), loosely coupled connections between several software systems can be achieved without the need of high integration efforts. Middleware is another technology used when connecting IT systems with each other in order to achieve electronic exchange of information. An Enterprise Service Bus (ESB) is a specific kind of middleware providing an approach based on web services and open standards that can be used as the implementation backbone for an SOA, thus promoting loose coupling of connected software systems.

In the context of E-Government and public E-Procurement, two initiatives exist that address the implementation of interoperable E-Government services and E-Procurement systems: The European Interoperability Framework (EIF) is an overall framework of the European Commission providing a set of guidelines and recommendations for the aforementioned interoperability levels in order to support the implementation of Pan-European eGovernment Services (PEGS) in the European Union. The CEN/ISSS workshop on Business Interoperability Interfaces (CEN/ISSS WS/BII) specifically addresses interoperability in European public procurement by providing specifications of business processes, business transactions as well as corresponding data models expressed in so called profile descriptions. These profiles aim at standardizing the exchange of business documents in European public E-Procurement thus supporting the implementation of interoperable E-Procurement systems.

Chapter 4

SERVICE-ORIENTED ARCHITECTURE

After the discussions about electronic procurement and interoperability, this chapter will describe the paradigm of *service-oriented architectures* that provides an approach to design and realize IT system architectures through which the requirement of interoperability and loose coupling of information systems can be achieved. After a comparison of existing definitions of the term SOA, the key building blocks of a service-oriented architectures will be identified and described. On that basis, a general discussion about important phases of the SOA implementation process will be conducted. Finally, the last section will provide an examination of benefits and challenges, and furthermore put SOA in context with the foregoing discussions about interoperability and public E-Procurement.

4.1 Introduction

4.1.1 Defining Service-oriented Architecture

Because one single definition of a *Service-oriented Architecture (SOA)* does not exist, this section starts with a comparison of three exemplary definition approaches that cover the main characteristics and concepts of an SOA mentioned in literature. The following paragraphs highlight the core elements as well as differences and similarities of the chosen definitions.

SOA as a paradigm for organizing distributed capabilities The *OASIS SOA Reference Model Technical Committee* [1] has developed a reference model for service-oriented architectures [2], that aims at defining "the essence of service oriented architecture, and emerge with a vocabulary and a common understanding of SOA" [OASIS, 2006, page 4]. In this reference model, service-oriented architecture is defined as

[1] Established by the Organization for the Advancement of Structured Information Standards (OASIS)

[2] Approved as an OASIS Standard in October 2006

> *"a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains."* [OASIS, 2006, page 8]

This abstract definition generally describes SOA as an organizational paradigm. The central concept is a capability, which is basically the ability to solve a business problem. Such capabilities may be distributed among different ownership domains. Necessary means to organize and utilize these capabilities are provided by the SOA paradigm. The reference model uses concepts to specify these means. The core concept of an SOA is a *service*, through which a capability is made accessible. Within an SOA, services are provided by *service providers* to *service consumers*. An important concept of a service is a *service description*. Such a description provides, for example, information about the service location, the functionality it provides and the interface through which the service can be accessed.

SOA as a software architecture based on key components Krafzig et al. [2005] follow a different approach to defining what a service-oriented architecture is. They provide a rather detailed definition, stating that a service-oriented Architecture is

> *"a software architecture that is based on the key concepts of an application front-end, service, service repository, and service bus. A service consists of a contract, one or more interfaces, and an implementation."* [Krafzig et al., 2005, page 56 ff.]

Contrarily to the rather abstract OASIS definition, this definition introduces SOA as a concrete software architecture that is based on the mentioned components. However, both definitions agree upon the role of services as being the fundamental building blocks which provide a solution for a specific business problem. The service contract specifies the purpose, functionality and interfaces of the service. Such a contract comprises a service along with its implementation and physical interfaces through which service consumers can access the functionality. Additionally to services, the authors introduce further crucial components of an SOA, which are application frontend, service repository and service bus. Application frontends enable users to interact with the system through graphical user interfaces. A service repository is described as a central database holding information required to discover and use services, such as the physical location of a service, the provider of a service, usage fees or technical constraints. Finally the service bus establishes connections between the different components of an SOA, for example the invocation of a service by other services or application frontends.

SOA as a model to decompose automation logic Erl [2005] provides another definition that will be used to add certain aspects appropriate to extend the comparison of different SOA definitions conducted so far. The author introduces Service-oriented architecture as

> *"a term that represents a model in which automation logic is decomposed into smaller, distinct units of logic. Collectively, these units comprise a larger piece of business automation logic. Individually, these units can be distributed."* [Erl, 2005, chapter 3.1]

Similar to the paradigm described in the OASIS definition, the author considers SOA to be a model rather than a concrete architecture or technology. The general approach in this model is to decompose business automation logic into smaller units of logic which are called *services*. Within SOA, services exist autonomously and can evolve independently from each other. Although being distributed and independent, they are not isolated from each other because several services are intended to comprise a larger piece of business logic. The author further describes that the aforementioned business automation logic is typically the implementation of a specific business process. Decomposed into smaller units, each service then encapsulates logic for a specific task of the process. Several of such services may interact with each other thus comprising other services. The enabling concepts to achieve service interaction are *service descriptions* and *messages* (cf. OASIS [2006] and Krafzig et al. [2005]). Service descriptions are required for services to be aware of and interact with each other. The interaction and communication between services is in turn achieved through the exchange of messages. Concerning the implementation the author specifically introduces the *Web Service technology* as a suitable technology platform to build service-oriented architectures (Web Services will be described in more detail in section 4.1.3 on page 39).

All in all, some aspects are described in this definition which are not specifically covered by OASIS [2006] and Krafzig et al. [2005]: The decomposition of logic of business processes into smaller units leading to services that encapsulate logic for single tasks or the composition of several services that encapsulate logic of a sub-process. Furthermore, web services are an implementation technology suitable to build an SOA.

Baseline for understanding SOA As a baseline for understanding service-oriented architectures, the three definitions described before can be summarized as follows: First of all, SOA is not a specific technology but rather a design paradigm to realize distributed information systems and to reach a better alignment of business processes and supporting software systems. The fundamental building block of an SOA is a *service*. Such services encapsulate logic of business processes that is made available through a standardized *interface*. Furthermore, *service descriptions* are required to discover and access services. The interaction and communication with and between services is achieved through the exchange of *messages*. A key characteristic of SOA identified in the definitions is that services may be distributed. So called *service consumers* can access any service provided by a *service provider*. Concerning the implementation of services, a suitable technology is the Web Service technology.[3] An SOA furthermore consists of important additional components: *Application frontends* enable user interaction, for example to start or interact with a business process. *Service repository* and *service bus* provide technical means to discover, access and connect services.

According to (Open Group SOA working paper, page 8) the "principle of service orientation can apply throughout the enterprise architecture, but is most commonly applied to the organization of the software that supports the enterprise's business operations." This statement clarifies that SOA is both Enterprise Architecture (EA) and Software Architecture (SA). Hence, these topics will be discussed in the following section.

4.1.2 SOA, SA and EA

As the term Service-oriented Architecture implies and as the previous definition shows, SOA is an approach to realize a specific kind of software architecture. However, SOA also addresses issues of enterprise architecture. This section therefore will put SOA in the context of Software Architecture and Enterprise Architecture.

According to [Hofmeister et al., 2000], a software architecture describes the structure of the elements and components of a software system as well as their functionality and interconnections, hence representing a design model of the system. Classical architectures decompose a system into the main components *servers*, which provide specific functionality through interfaces, and *clients*, which access these interfaces. The system is additionally sliced into three main layers: The *presentation layer* providing the interface to system users, the *application logic layer* provides the business logic and functionality and the *database layer* finally consists of the systems data storage (cf. [Dunkel et al., 2008; Lassmann, 2006]).

The basic two-tier client/server architecture basically consists of clients, that provide the user interfaces and implement all the application logic, which are connected to a single or several different servers. This architecture can be extended by one or more application servers that provide some or all of the business logic as well as additional functionality such as load balancing and queuing of database requests. (cf. [Dunkel et al., 2008]) This leads to 3- or N-tier software architectures. A basic 3-Tier architecture is depicted in figure 4.1.[4]

[3] More information about web services and service-orientation principles are part of section 4.1.3 on page 39

[4] Own figure

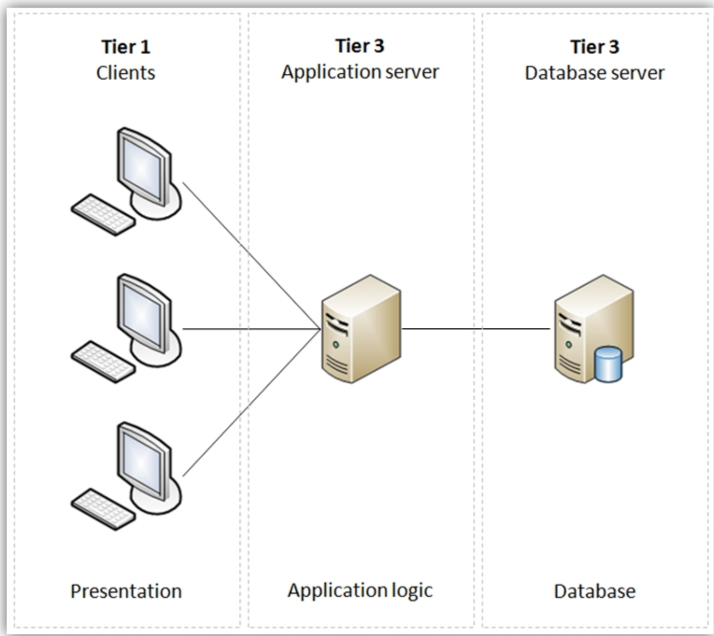

Figure 4.1: Basic 3-tier software architecture

The Internet and web services enable an evolution and further extension to this architecture, leading to what Erl [2005] calls a *Distributed Internet architecture*. The core aspect of this architecture type is that business logic can be shared among several distributed servers by exposing functionality through web service interfaces. This enables software to access and reuse functionality that is distributed among several servers. Such point-to-point connections between different application servers are shown in figure 4.2.[5]

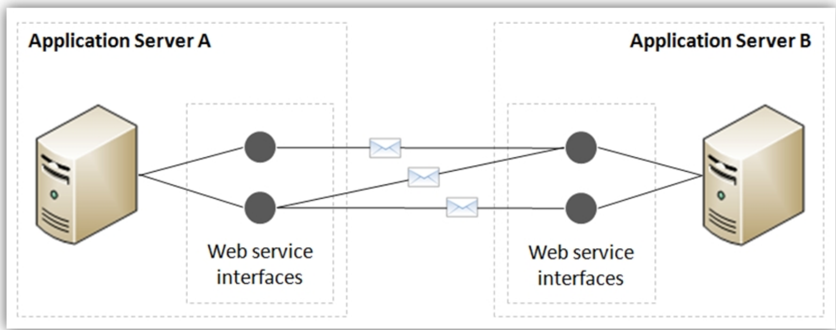

Figure 4.2: Sharing business logic through web services

As the author states this is simply "a distributed Internet architecture that uses Web services" and "in this manner does not qualify as a true SOA" [Erl, 2005, ch. 4.3.4.]. Hence, SOA is not only the sharing and reusing of distributed business logic but moreover aims at loosely

[5] Own figure

coupling of software systems and composing services into business processes. Rosen et al. [2008] clarify: "The real value of SOA comes when reusable services are combined to create agile, flexible, business processes." [Rosen et al., 2008, p. 33].

This is where the alignment of IT and business processes, and business goals respectively, comes into play, implying that the SOA approach is furthermore an issue of the overall Enterprise Architecture. According to Minoli [2008], Enterprise Architecture (EA) is a "blueprint for the optimal and target-conformant placement of resources in the IT environment for the ultimate support of the business function" [Minoli, 2008, p. 9]. The purpose of an EA is to describes an enterprise and its IT environment (hardware and software systems) that supports the conducted business and the overall business vision in terms of components and their relationships among each other. It also provides a strategical planning for the development and evolution of the IT environment. As a part of the overall EA, the IT systems architecture provides a view on the software systems as well as a strategy for the development of software systems within the organization to support the overall business functions and processes [Barry, 2003; Lawler and Howell-Barber, 2008; Minoli, 2008]. A Service-oriented Architecture is an approach to realize such an IT systems architecture, that consists of services as the core building blocks as well as further architectural components.[6]

After defining SOA and putting in into the context of the enterprise IT strategy, the next section will now provide a closer discussion about services, which are the fundamental building blocks of an SOA.

4.1.3 Introducing Services

According to the previous definition of an SOA, the core component is a service that encapsulates business functionality implemented as a specific software component. As summarized in the following list, a service consists of three basic sub-components (cf. [Brown, 2008; Krafzig et al., 2005; OASIS, 2006; Rosen et al., 2008]).

Implementation The business functionality the service provides. It consists of a set of operations that are executed when the service is invoked through its interface.

Interface Enable service consumers to access the functionality and operations provided by the service.

Contract Description of the service specifying its purpose, functionality and interfaces. In addition to these more technical parts, a contract can furthermore contain organiza-

[6] More information on SOA components will be given in section 4.2.1 on page 43.

tional, legal and commercial aspects, such as quality of service, access constraints or usage billing.

Concerning to the context of a service, several service types can be identified. Independent of its type, the idea behind a service is exposing its business functionality that can be accessed in standardized ways over a network (e.g. Internet). The functionality thus can be reused within automated business processes or by other services. The different types of services will be discussed in the following section. Afterwards an overview on different characteristics and service design principles will be provided, followed by an introduction of web services, which comprise a suitable technology platform to implement services in an SOA because they embody the aforementioned design principles.

Service types

A general classification distinguishes between *atomic* and *composite* services [Newcomer and Lomow, 2004]. Atomic services do not use other services whereas composite services rely on other services by accessing their functionality. These can be other atomic or composite services that are used within business processes, hence providing more value by combining different functionalities. Both [Krafzig et al., 2005] and [Rosen et al., 2008] undertake a more detailed classification that additionally to atomic and composite services distinguishes between the following types:

Basic / foundation services provide basic functionality similar to the above mentioned atomic services. In terms of *utility services* they provide functionality that can be used across an organization whereas the functionality of *domain services* is only relevant within a specific business domain.

Intermediate / business services are composite services that utilize the functionality of other services, hence providing a higher business value.

Process-centric services are similar to intermediate services, with the difference that they take control of a business process and are able to maintain the state of a process.

Public enterprise / external services are implemented to provide interfaces to share functionality across organizational boundaries.

These different classifications show, that services can be used within different contexts, either to provide basic functionality or to add additional value by combining other services. Hence they can be used to share functionality between different software systems or utilize it within business processes both within a single organization or across organizational boundaries.

Service design principles and characteristics

[Bean, 2010; Newcomer and Lomow, 2004] both define a list of service characteristics describing important aspects of services in an SOA. [Erl, 2008] also provides such a list, calling it service design principles. All of them describe the foundational concepts of services and service-oriented architectures through which flexibility and interoperability of IT systems can be achieved, hence the following list will summarizes these principles and characteristics [Bean, 2010; Erl, 2008; Newcomer and Lomow, 2004].

Standards-based refers to the use of open standards, thus promoting interoperability and independence of specific technologies.

Service contracts are formal definitions of a service and its functional purpose as well as interfaces through which the functionality can be accessed.

Loose coupling refers to the degree of inter-dependencies between services. In an SOA, services are built to interact without tight dependencies.

Reusability is the ability to reuse the functionality that a service implements.

Composability is related to reusability. It refers to the composition of services to form a composite service that provides higher business value.

Discoverability is the ability to find services by their descriptions allowing service consumers to utilize a service.

Autonomy refers to the independent development and maintenance of services.

Abstraction means that services hide their internal implementation logic from the outside world and only exposing the provided functionality through the service contracts.

In order to support the understanding of service-oriented architectures and to complete this introductory section, the next section will now give a brief overview on SOA maturity models as they identify important characteristics of SOA solutions.

4.1.4 SOA Maturity Models

Generally speaking, maturity models intend to provide a structural model to identify incremental stages of maturity of a concrete SOA solution. These models have been developed with a focus on the transformation of existing systems towards sophisticated service-orientated architectures and help to identify general requirements for the further development of an SOA, starting from its current maturity towards higher maturity levels [IBM, 2010].

Important aspects of three exemplary maturity models will be described in the following (cf. [IBM, 2010; Rathfelder and Groenda, 2008; Sonic Software Corp. et al., 2005; The Open Group, 2009]).

Although the examined maturity models differ in the number and names of their levels, they all have in common that the maturity of an SOA ranges from a very basic and experimental service-orientation to more sophisticated levels that promote service composition and orchestration, finally leading to flexible IT systems that support dynamic business processes and a high degree of interoperability. Along with the maturity levels, the models define several dimensions in order to describe important characteristics of each level, covering for example technical, architectural and organizational issues. The Open Group [2009] provide the most detailed model consisting of seven levels. The lowest maturity level describes silos of monolithic IT systems limiting interoperability and flexibility of business processes and supporting software systems. Loosely coupled software components and services increase flexibility and interoperability on higher levels. The greatest amount of agility and interoperability is achieved by the ability to orchestrate and automate business processes, and to discover and access services dynamically, in order to provide composed business services. Sonic Software Corp. et al. [2005] as well as Rathfelder and Groenda [2008] both define five maturity levels starting with initial services that provide basic service orientation and integration of legacy applications. Greater alignment of these services to business processes is achieved through service orchestration on the more mature SOA levels. Further support for interoperability, loose coupling and automation of collaborative business processes is achieved on the highest maturity stages with the provision of business-centric services.

In conclusion, SOA solutions can range from basic service orientation with a focus on interoperability and reusability of business logic within organizations, to sophisticated SOAs that allow orchestration and reorganization of business processes within and across organizational boundaries and support both flexible EAI and B2BAI.

4.2 Designing and Implementing an SOA

After elaborating a basic understanding of what SOA is, this section will now focus on the general development process of a service-oriented architecture, thereby concentrating on service design and implementation aspects. Issues related to enterprise architecture are omitted as they are beyond the scope of this master thesis. After identifying the core components of an SOA, a top-down approach to identify and implement services will be described, also taking important modeling and implementation standards into consideration.

4.2.1 Components of an SOA

As mentioned before, an SOA is neither a single technology nor a product, but a paradigm to design and implement the architecture of an IT system. To identify different components of an SOA, it is useful to draw a conceptual architecture model. Liebhart [2007] provides a layered model of an SOA that will be taken as the basis for a further discussion of important components, also including further literature that provide similar models [Bieberstein et al., 2006, 2008; Krafzig et al., 2005; McGovern et al., 2006; Minoli, 2008; Rosen et al., 2008; Woods and Mattern, 2003]. Figure 4.3 illustrates this model and the major technical components that are attached to each layer.[7]

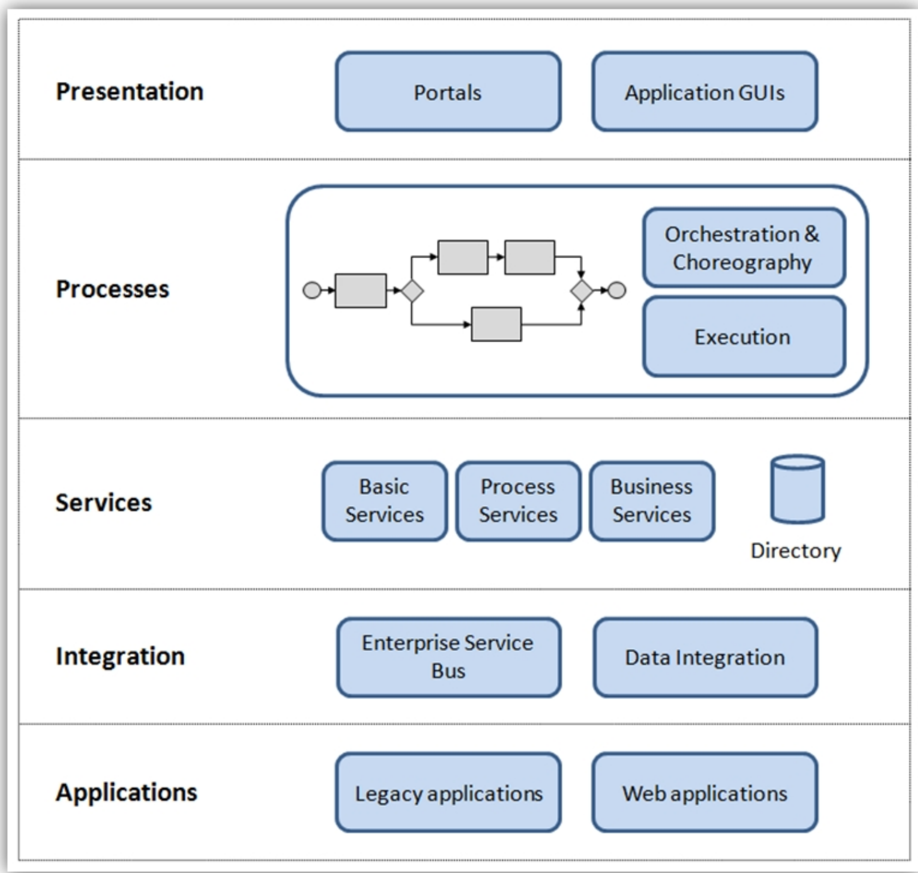

Figure 4.3: Conceptual model of an SOA

The **presentation layer** enables system users to access the system functionality, for example to start or interact with business processes or to access business services that expose their functionality to users. This is achieved through Graphical User Interfaces (GUI) or portals.

The **process layer** consists of the business processes an organization executes internally or in collaboration with external business partners. This includes on the one hand the automation

[7] Own figure

and execution of business processes with a business process runtime engine. On the other hand, the analysis, design and modeling of business processes as well as the identification of relevant services provides a detailed understanding of the different processes. In this context, *orchestration* and *choreography* are two important concepts: Orchestration refers to the ordering and arrangement of process tasks that are executed from an initiating to a terminating event. Choreography refers to the sequence and constraints of interactions (e.g. exchange of messages) between collaborating peer services [OASIS, 2006; W3C WSA Working Group, 2004b].

The **services layer** comprises all services that have been identified and implemented to provide a specific functionality. Composite services may be composed of several basic services in order to provide a desired business value or may be orchestrated to a sequence of tasks within a business process. Other services may be used by applications, or the other way around, may expose specific application logic for external usage. In order to manage and find services, a service registry contains information (i.e. the service contracts) of all services that may be discovered and accessed by service consumers.

The **integration layer** serves the need for integrating software components and data within an organization or with external business partners. These software components may be existing legacy applications or services. An Enterprise Service Bus provides necessary means on this layer, for example message queuing and routing, adapters to integrate legacy applications, integration of process engines to orchestrate services, data format transformations or monitoring tools (cf. [Chappell, 2004]). An ESB is primarily suitable for large SOA systems. In smaller environments it may be sufficient to renounce such integration middleware so that services interact directly with each other.

The **application layer** includes all software applications, for example existing legacy applications or new web or other applications. These applications may access other services or expose internal logic through services.

4.2.2 A Top-Down Approach to Identify and Implement Services

This section will now expand the previous discussions about aspects of an SOA by providing a general approach for the development and realization of a service-oriented architecture, focusing on the tasks required to design and implement services. The basis is a top-down approach starting from the analysis and design of the business domain and processes that is decomposed into smaller sub-domains and components. The aim is to identify reusable software components that can be implemented as services for later orchestration within new business functions and processes. Figure 4.4[8] illustrates the main tasks to identify, implement and assemble services within a service-oriented architecture and outlines some important artifacts and standards that are produced and used during the phases.

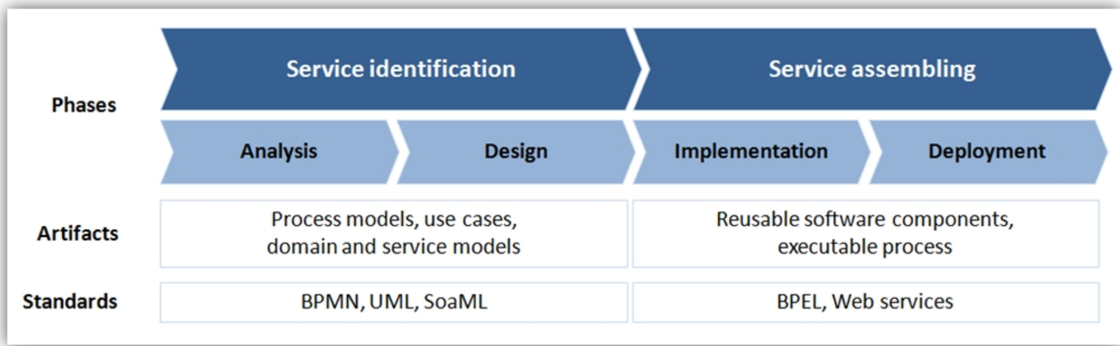

Figure 4.4: General Tasks for service identification and implementation

It should be noted, that a standardized or common development process for service-oriented architectures does not exist. However, the development of SOA solutions also requires techniques and methodologies known from software engineering and software development, such as requirements engineering, analysis, design and implementation. A crucial task in the context of SOA is the identification of services, based on the analysis and design of the target domain. Hence, this section will concentrate on the identification of services followed by an introduction of how the identified services can be assembled. An elaboration of a complete SOA development process is out of scope of this master thesis.

The following sections will describe the general tasks depicted in figure 4.4, introducing a set of modeling notations and standards that can be used for the identification and implementation of services. The purpose of this section is not to provide an overview of existing modeling languages, standards or products, but rather to examine a subset to form a suitable and consistent package of modeling and implementation standards focusing on the processes and services layer depicted in figure 4.3 on page 43.[9]

4.2.3 Service Identification

The phase of service identification mainly consists of an analysis of the business domain and the modeling of business processes in order to decompose them into sub-processes, tasks and business functionality that can be encapsulated by services. Artifacts of this phase are

---

[8] Own figure

[9] References to further information on other standards will be given when appropriate.

domain as well as business process models (cf. [Bieberstein et al., 2008; Brown, 2008; Deeg, 2009; Erl, 2005; Hoernes, 2007]). These describe

- business processes, workflows as well as relevant data objects,
- business use cases,
- participants and their relationships (i.e. possible service consumers and providers),
- functional components, services, service interfaces and
- information objects.

In the SOA context, a set of standards that is suitable to specify the above mentioned modeling artifacts consists of the Business Process Modeling Notation (BPMN), Unified Modeling Language (UML) and Service oriented architecture Modeling Language (SoaML). The following sections will discuss these notations, highlighting important aspects as well as advantages of their usage in the context of SOA design and analysis.

Business Process Modeling Notation (BPMN)

The Business Process Modeling Notation (BPMN), currently under the responsibility of the Object Management Group (OMG), is a standard modeling notation for business processes. The current version 1.2 has been published in January 2009. A beta version of BPMN 2.0 has been published in August 2009 and is currently in the finalization process and extends the former specification with several new notational and structural elements [Object Management Group, 2009a]. It should be noted, that although several other process modeling notations such as *Event-driven Process Chains (EPC)*, *Petri Nets* or *Workflow Nets* exist, BPMN has become a widely accepted standard for business process modeling and SOA. One major advantage of BPMN is the ability to model both high-level organizational as well as detailed technical processes, thus enabling analysts and developers use and understand BPMN (cf. [Brown, 2008; Cummins, 2009; Weske, 2007]). BPMN furthermore provides a mapping to the Business Process Execution Language (BPEL), enabling process models being transferred into an executable format. The following sections will introduce important aspects of both BPMN versions, a further discussion about BPEL will follow later.[10]

[10] For more information on the other notations see for example [Weske, 2007].

Use of BPMN The Business Process Modeling Notation allows modeling of both intra-organizational and inter-organizational interactions (processes and collaborations). The process diagrams are intended to provide means of communicating information about processes while being easily understandable by various stakeholders, like business analysts, technical developers and business users. Primarily, BPMN is a graphical modeling notation, however, in order to support both business process design and process implementation, BPMN provides means to map the design elements of a diagram to an execution format such as the Business Process Execution Language (BPEL).[11]

BPMN elements The notation provides four categories of graphical elements, which will be described in the following. The basic elements are illustrated in figure 4.5[12].

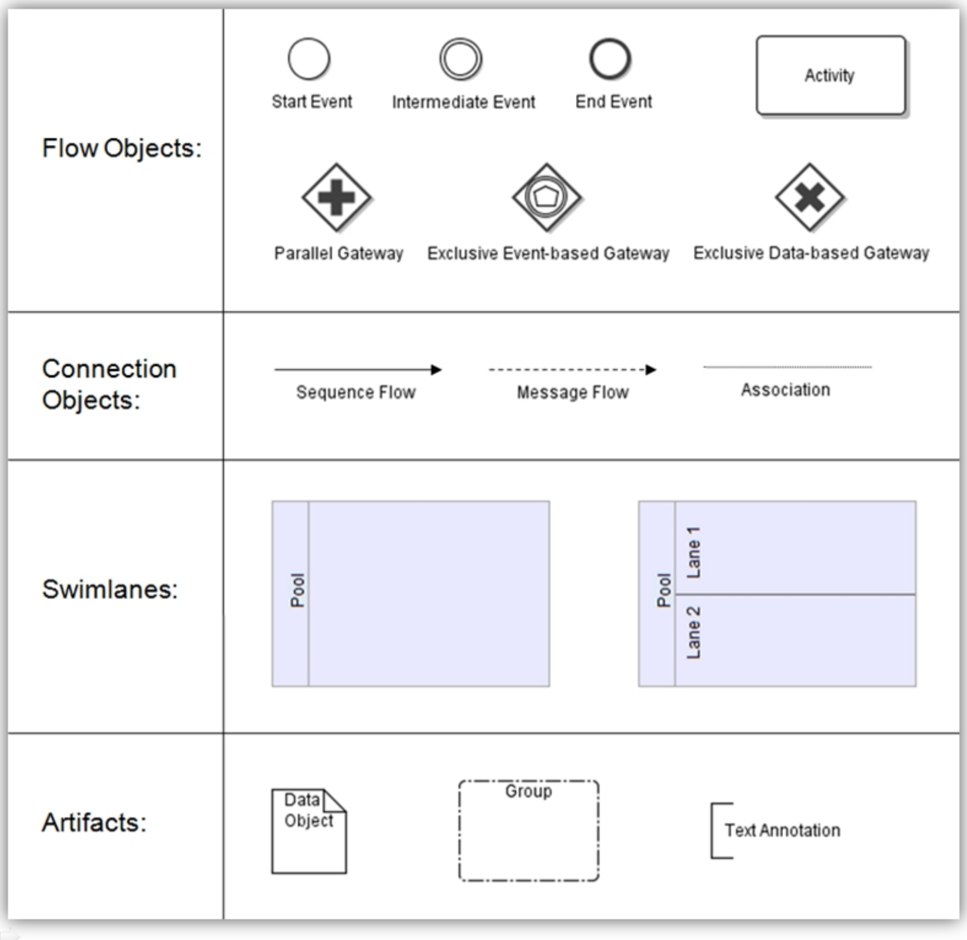

Figure 4.5: Basic elements of BPMN

Flow Objects define the behavior of a business process: *Events* happen during a business

[11] Detailed information and examples on the mapping of BPMN to BPEL can be found in the specification documents as well as in [White, 2005] and [Ouyang et al., 2009]

[12] Own figure

process and can have a trigger and a result. *Activities* are used to visualize the work that is performed within a process, whereas *Gateways* split or merge the flow of a process based on the split type and condition. *Sequence Flows* are used to model the process flow whereas *Message Flows* are used to visualize message exchanges between two participants of a process. *Associations* are used to attach Artifacts to Flow Objects. *Pools* are graphical containers which represent actors and participants of collaborative business process. Pools can only be connected to each other by Message Flows. A Pool can be further partitioned into several *Lanes*, each representing a specific department of the organization. *Data Objects* visualize additional information such as documents or messages that are among flow objects. To improve readability, *Group Boxes* can be used to logically group several objects, whereas *Text Annotations* can express additional information.

Extensions of BPMN 2.0 Version 2.0 of the BPMN specification defines several extensions of the current notation. The major novelties concern graphical elements and diagram types [Object Management Group, 2009a]: Extensions of graphical elements, for example, allow modeling of specific task types, which further specify the action that is performed, such as Manual Tasks or Service Tasks, which can be used to model execution of web services or automated applications. Extensions of diagram types are used to visualize so called Choreographies and Conversations. Choreography Diagrams specifically allow visualizing details about the coordination of interactions and information exchanged between organizations. The basic shape used in Choreography Diagrams is a Choreography Task as illustrated in figure 4.6[13].

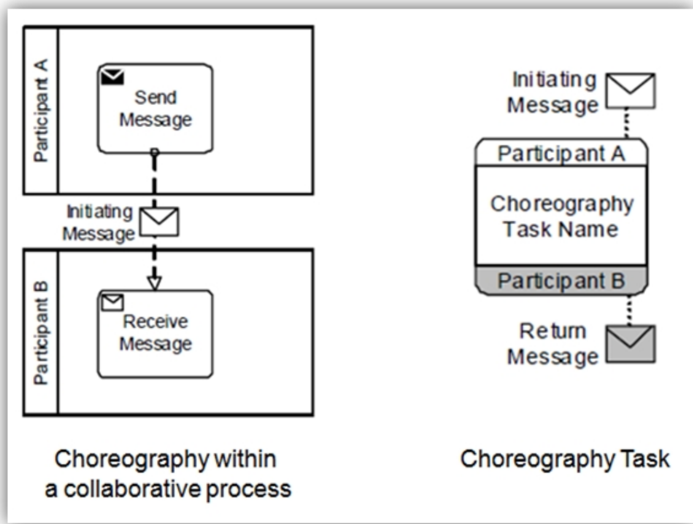

Figure 4.6: Modeling choreographies with BPMN

[13] Own figure

A Choreography can be modeled as a stand-alone diagram or in between two pools of a collaboration process. It consists of several Choreography Tasks as well as Events, Gateways and Artifacts. Conversation diagrams provide an abstract view of the conversations illustrating communications and message exchanges between several participants. Hence, a Conversation Diagram provides a general overview about participants an communication links between them in a specific domain. Whereas a Choreography Diagram can be used to model the detailed behavior and message exchange sequences of a Conversation. The general flow of tasks within a collaborative business process can be modeled with the common process diagrams.

Unified Modeling Language (UML)

The Unified Modeling Language (UML) is a specification developed by the Object Management Group providing several kinds of diagrams to visualize the architecture, structure and behavior of software systems. In the context of SOA design, some of these diagrams a suitable to model both structural and behavioral system aspects such as use cases, interactions between participants as well as internal structures of system components. The following will give a brief overview of these diagrams and how they can be used for SOA design.[14]

Use case diagrams allow modeling of the usage of a system and the involved actors that use the system's functionality. Use case diagrams are suitable to identify and provide an overview of the functionality of a specific business domain, including the participants that provide and use the functionality.

Activity diagrams are used to model the behavior of a system focusing on the execution and flow of activities and performed actions. Activity diagrams are suitable to orchestrate and model the control flow of tasks within business processes or detailed workflows, either performed by one actor or by several participants.

Sequence diagrams visualize the sequence of messages exchanged between different participants. Messages represent the invocation of specific functionality along with business objects and data. Hence sequence diagrams allow the specification of interactions and choreographies expressed as a sequence of transactions between the participants.

Composite structures diagrams provide a high level structural model of a system and system parts that are connected to each other. Furthermore, it can be used to model the roles of system parts in specific collaborations to accomplish specific tasks.

---

[14] For a complete listing and further details see [Grässle et al., 2005; Object Management Group, 2009c; Pilone and Pitman, 2007].

Component diagrams enable the decomposition of software systems into smaller subsystems, so called *components* as well as their relationships and interfaces through which internal component functionality can be accessed. Component diagrams therefore provide means to model reusable software components that can be implemented as services.

Class diagrams are used to model the structure and relationships of object classes and their interfaces in a system. In the context of SOA design, class diagrams can be used for the structural modeling of business objects, their provided interfaces and relationships.

Service oriented architecture Modeling Language (SoaML)

SoaML has originally been developed by a consortium of several companies and organizations, until it has been adopted by the Object Management Group (OMG) in 2009. Based on the view on SOA as being an architectural paradigm, the Service oriented architecture Modeling Language "provides a standard way to architect and model SOA solutions using the Unified Modeling Language (UML)" [Object Management Group, 2009b, p. 14]. *Profiles* are part of the UML specification enabling the development of UML based stereotypes for the use in specific contexts - within SoaML, this context is the modeling of services and service-oriented architectures. In particular, SoaML specifies notational elements that allow analysts to model services, service providers and consumers as well as service contracts. The focus thereby is on the business and architectural view of an SOA, implementation details of services are out of scope [Casanave, 2009; Deeg, 2009]. An overview of the major stereotypes defined in SoaML will be given in the following, briefly describing the specific purposes of each stereotype.[15]

Participants are the actors (e.g. people, organizations, software components), either providing services as *providers* or consuming services as *consumers*. **ServicePoints** and **RequestPoints** specify that a participant provides a specific service (*ServicePoint*) or accesses a service provided by another participant (*RequestPoint*). The functionality and ability of services is specified by **Capabilities**. Rules on how service consumers and providers have to interact are described by **ServiceInterfaces**. A ServiceInterface specifies for example the service name and its functional purpose, information exchange and protocols or rules for using a service. Two participants that interact have to agree on **ServiceContracts**, that define terms and conditions as well as interfaces required to enact a service. The aforementioned ServiceInterafaces thereby specify the role of each participant. The **ServicesArchitecture** is used to model the relationships among participants that provide and consume services thus

[15] The full specification as well as examples can be found in [Casanave, 2009; Object Management Group, 2009b].

providing a high-level view on the scoped business domain.

4.2.4 Service Assembling

Based on the developed design models, this phase consists of the implementation and deployment of services, their composition into software components as well as orchestration and execution of business processes (cf. [Bieberstein et al., 2008; Brown, 2008; Erl, 2005; Margolis and Sharpe, 2007]). Accordingly, this section will focus on the implementation of services as web services and the orchestration and execution of business processes using the Web Services Business Process Execution Language (WS-BPEL) and will introduce some important related standards.

Service implementation with Web services

As already mentioned, a promising and common implementation technology for services in an SOA is provided by *web services*. Section 3.2 on page 24 already introduced web services. The following list shall recapture the basic definition of a web service. Afterwards, the important standards on which web services base will be introduced. According to section 3.2 (cf. [Sherif, 2010; W3C WSA Working Group, 2004a]), a web service

- is a software system;
- supports interoperable machine-to-machine interaction over a network;
- has an interface that is described in a machine-readable format (WSDL);
- interacts with other systems by exchanging SOAP messages over HTTP.

Web services are based on three important standards: Web Service Description Language (WSDL), Simple Object Access Protocol (SOAP) and Universal Description, Discovery and Integration (UDDI) (cf. [Cerami, 2003; McGovern et al., 2006; W3C WSD Working Group, 2007]). **WSDL** provides an XML grammar to describe the interface of a web service. It contains information about the operations the service provides as well as data types and message formats of service requests and responses. Furthermore, information required to access the service, such as location and transport protocols, is provided by a web service description. **SOAP** is a protocol for exchanging XML messages that can be delivered via a network protocol (e.g. HTTP). SOAP messages are used to deliver the request and response messages when communicating with a web service. **UDDI** provides a standard for registering and searching web services in a service directory. It provides information about registered web services such as the service provider, the web service description, location and interface. Service consumers can search in a UDDI directory and locate and access appropriate services based on the aforementioned information. These three core components are part of a basic web service architecture as depicted in figure 4.7.[16]

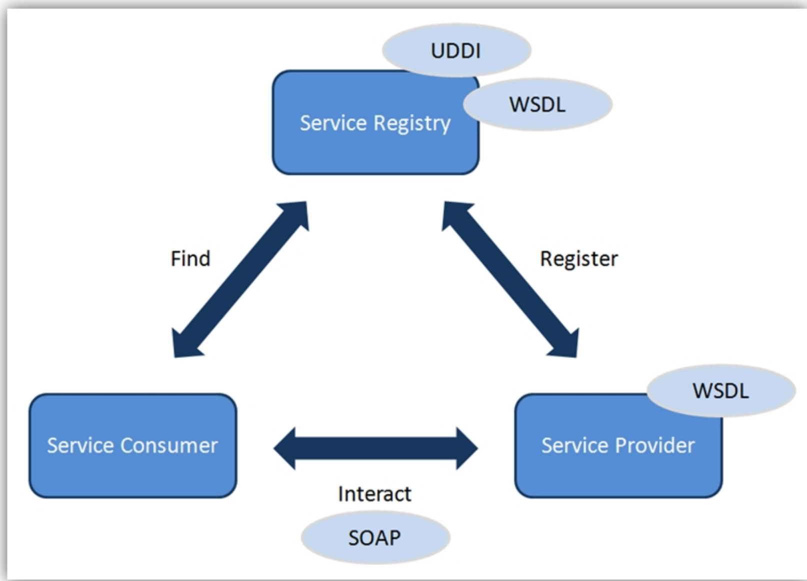

Figure 4.7: Basic web service architecture

This basic web service architecture illustrates that it embodies several of the service design principles and characteristics described in section 4.1.3 on page 39, such as reusability, composability and discoverability. Moreover, many standards have been developed around web services that address specific purposes and extend the basic purpose of SOAP and WSDL. Some of these cover aspects like security, reliable messaging or service discovery. One such standard, Web Services Business Process Execution Language (WS-BPEL), will be described in the following section, because it addresses the orchestration of web services within business processes.[17]

Process orchestration with the Web Services Business Process Execution Language (WS-BPEL)

As mentioned before, one purpose of identifying and implementing services is the reuse and orchestration to flexibly develop and execute business processes. As the previous section already stated, web services are a common and suitable implementation technology for ser-

[16] Own figure, based on http://www.w3.org/TR/2002/WD-ws-arch-20021114/Triangle.png

[17] For further information about different web service standards see http://www.soaspecs.com/ws.php, http://www.oasis-open.org/specs/ and http://www.w3.org/2002/ws/

vices in an SOA. This section will now focus on the Web Services Business Process Execution Language (WS-BPEL) as it provides a standardized XML based language to orchestrate web services within business processes. The major advantage of WS-BPEL is its support for web services and the relationship to BPMN, as this standard provides the ability to translate BPMN processes models into WS-BPEL [Chang, 2006].

The Web Services Business Process Execution Language (WS-BPEL), or BPEL in short, developed by the Organization for the Advancement of Structured Information Standards (OASIS), provides a standardized process orchestration language to describe executable business processes based on web services: "WS-BPEL defines a model and a grammar for describing the behavior of a business process based on interactions between the process and its partners. The interaction with each partner occurs through Web Service interfaces" [OASIS, 2007, p. 8]. Accordingly, a WS-BPEL process composes several partner services and furthermore acts as a composite process service itself, as it is started by invocation through another external service, that may be part of another BPEL process itself [Juric, 2006]. A process defined in WS-BPEL consists of several files: On the one hand, the BPEL file defines the activity sequences, partner links, variables as well as fault and event handlers. On the other hand, WSDL documents are required to specify the web service interfaces and the message exchanges between partners. [Havey, 2007]. To execute a process, the process definition file and the WSDL documents are deployed on a BPEL execution engine that is capable of managing the activity invocations and message exchanges of the process [Vasiliev, 2006].

As WS-BPEL processes consist of message transactions between web services it is not possible to include human interaction into a BPEL process. The OASIS WS-BPEL Extension for People (BPEL4People) specification addresses exactly this need and provides an extension to WS-BPEL to support involvement of people and human interaction within business processes [OASIS, 2010b]. Many of the concepts of BPEL4People are derived from the WS-HumanTask specification, for example tasks, notifications and roles.[18]

4.3 The use of SOA for Public E-Procurement

This section will now examine the use of service-oriented architectures in the domain of public E-Procurement. Based on the previous definition and discussion about important aspects of an SOA, the following section will first examine general benefits and challenges related to the SOA approach. In the second section, a closer look on the suitability of service-

[18] WS-HumanTask is another OASIS specification providing an abstract definition of human interaction with business processes. For more information see [OASIS, 2010a].

oriented architectures in the government and public procurement domain will be provided, while also taking existing research in this field into consideration.

4.3.1 SOA Benefits and Challenges

The SOA-approach promises a set of benefits through which organizations can improve their business operations and supporting Information Technology. However, these benefits are related to several challenges that have to be addressed in order to achieve the desired improvements. Major benefits and challenges will be discussed in the following, summarizing important issues according to discussions in [Bean, 2010; Beinhauer et al., 2007; Bieberstein et al., 2008; Endrei et al., 2004; Juric et al., 2007; Richter et al., 2005; Rosen et al., 2008; The Open Group, 2007].

Leveraging the potentials of a service-oriented architecture enables organizations to benefit through a set of different technical, organizational and economic advantages. Some important and major benefits are illustrated in figure 4.8.

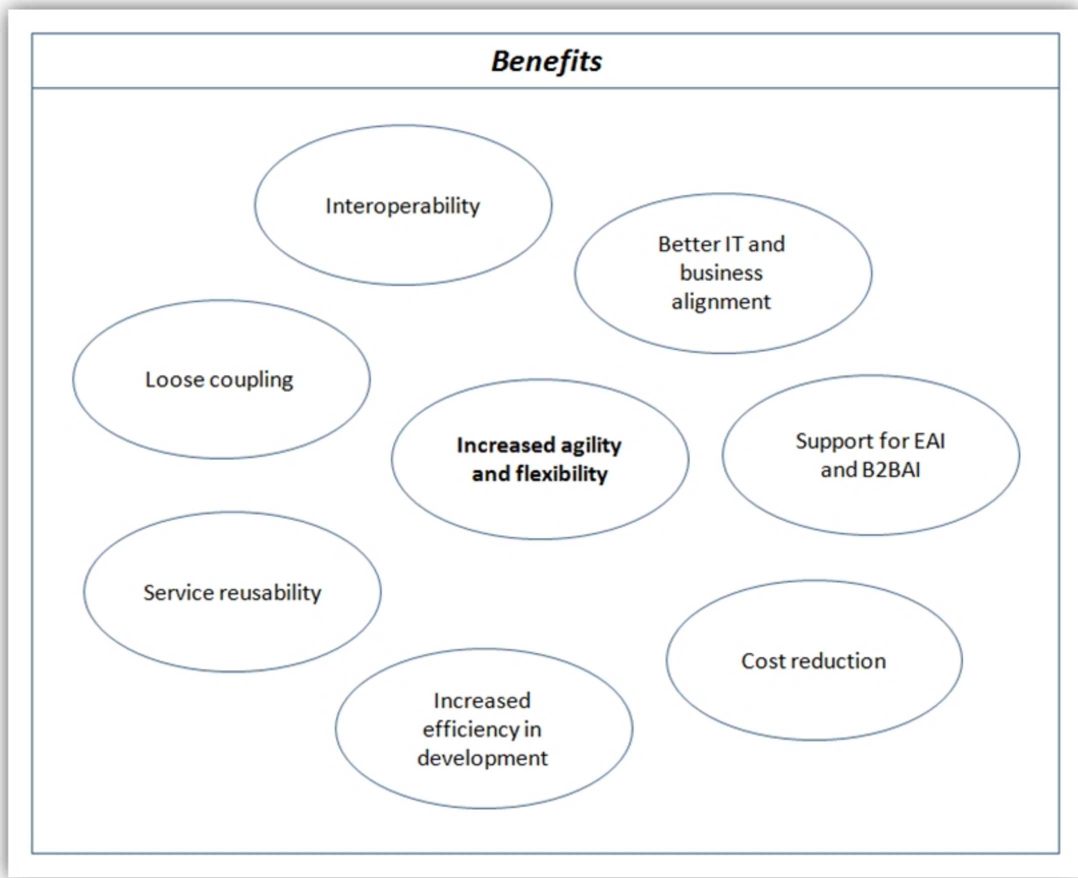

Figure 4.8: Benefits of Service-oriented Architecture

The core benefit is that SOA enables organizations to build an agile and flexible IT architecture so that software applications as well as business processes can be reorganized and adapted to changing environmental conditions. Enabling aspects are interoperability, loose coupling and service reusability: Interoperability refers to the platform independence of SOA and the use of several standards, such as web services and related standards, enabling data exchange between independent IT systems. Loose coupling and reusability both enable the flexible orchestration and composition of existing services into new business services and processes. To achieve this, the SOA approach strictly promotes a better alignment of the business operations of an organization to the supporting information technology as reusable software components are orchestrated to provide greater business value. With a higher SOA maturity, the interoperability spreads beyond organizational boundaries enabling the creation of distributed IT systems that support collaborative business operations. This is supported by an Enterprise Service Bus as the integration platform of an SOA, that enables integration of services and legacy applications both internal and external to an organization (EAI and B2BAI). Finally, cost reduction and increased efficiencies in the development of services and the IT infrastructure can be achieved by loose coupling and reusing existing services as well as a common SOA infrastructure through which services can be provided and accessed.

To achieve these benefits, several challenges have to be resolved in order to leverage the potentials of an SOA. Major challenges are illustrated in figure 4.9.

Although interoperability is one benefit of an SOA, this is mainly achieved on the technical level. Semantic interoperability still remains a challenge due to the different vocabularies that are used within business units or collaborating organizations. The same challenge exists with organizational interoperability. Hence, the development of a common semantic model and adjustments to business processes are required to realize the full interoperability potentials of and SOA. This can be achieved for example by using standards like ebXML, UBL or, in the public procurement context, the CEN/ISSS BII that both provide means for achieving semantic and organizational as well as process interoperability. In the government context, further legal and political issues affect the interoperability of IT systems. As discussed in chapters 2 and 3, differences between national legislations intensify the heterogeneity of government services, hence impeding interconnections and data exchange of systems that need to be interconnected across borders. Political and legal frameworks such as the European Interoperability Framework and the European public procurement directive are two examples of additional prerequisites for a successful SOA implementation. Another aspect is the need of a correct understanding of what SOA is and how it can be implemented. A common definition of SOA, services, service characteristics and a standardized development approach is still missing, which may lead to misunderstandings of the value and capabilities of the SOA paradigm. Therefore it is necessary to understand both business and technology dimensions (enterprise architecture and software architecture). This furthermore enables an

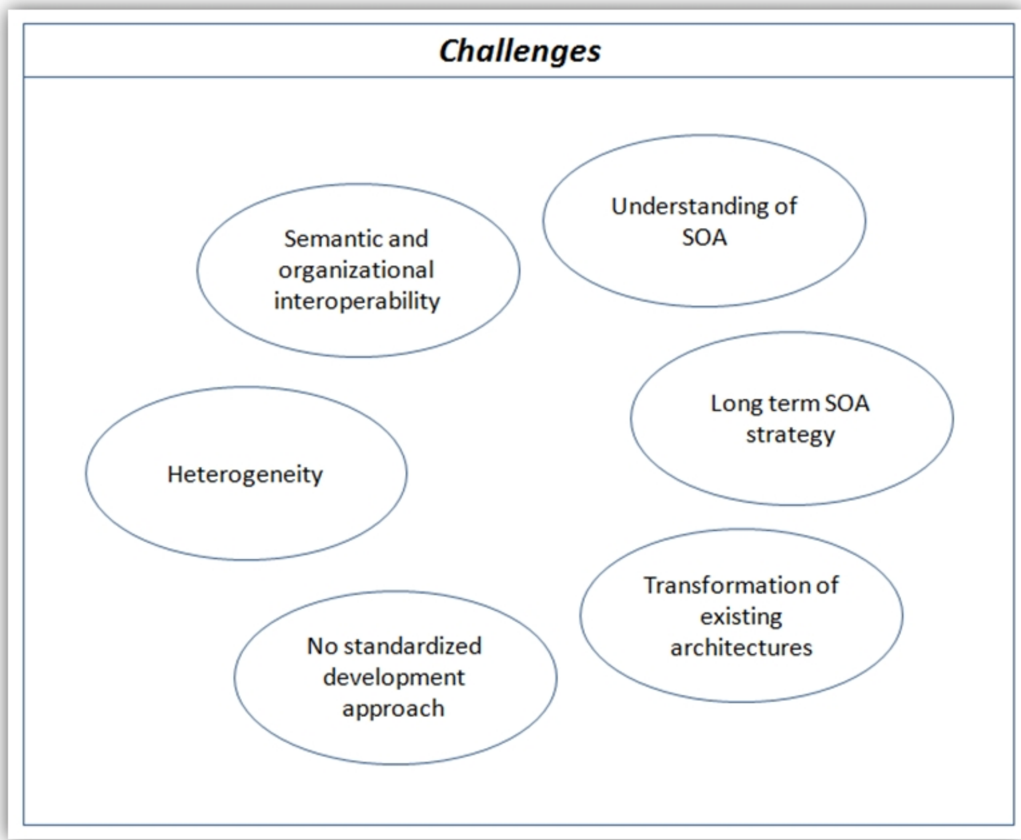

Figure 4.9: Challenges of Service-oriented Architecture

organization to develop an overall long term SOA strategy. Such a strategy has to provide a view on all levels of an SOA, such as business, processes, information, services, applications and technology. Additionally, the SOA strategy has to specify a concrete implementation strategy that follows a top-down approach starting form modeling business requirements and processes and decomposing them into services. This can be supported by the selection of modeling techniques such as SoaML or BPMN. The technical basis has to be provided by an appropriate selection of an implementation methodology and technical infrastructure components, such as web services, Enterprise Service Bus and business process engines. Hence, a sufficient alignment of businesses and IT can be achieved. However, the implementation of an SOA strategy usually requires a transformation of existing architectures which implies wide ranging changes and adjustments of an organization's existing IT and business architecture. SOA maturity models support organizations in identifying and specifying required transformation phases to realize sophisticated SOA systems.

4.3.2 SOA in the Context of Public E-Procurement

As discussed at the beginning of this master thesis (see chapters 2 and 3), interoperability is a fundamental requirement in the context of electronic government and in the provision of Pan-European eGovernment Services (PEGS). This spreads across several domains in which G2G, G2C and G2B collaborations and interactions occur. As this chapter showed, service-oriented architecture is a paradigm to build interoperable and flexible IT infrastructures, making it a suitable approach to achieve interoperability for the provision of PEGS, and cross-border electronic public procurement in particular.

Web services as the implementation platform for services have already been declared suitable for interoperable processes and electronic document exchange (cf. [Anghelakis et al., 2007; European Dynamics S.A., 2007b; Klischewski, 2004]). According to [European Dynamics S.A., 2005], SOA is a potential strategy to realize interoperable E-Procurement systems, as independent services can be combined to form higher level business processes. Laskaridis et al. [2008] furthermore specify the so called "CCIGOV"[19] generic interoperability platform that is achieved with the SOA approach in order to enable government organizations deliver e-services to international businesses, with public procurement being a relevant domain of G2B interactions within the CCIGOV platform.

In their report for the IDABC Programme "European Interoperable Infrastructure Services: Study on potential reuse of system components", Declercq et al. [2009] specifically state that the SOA paradigm is suitable in the context of reusable components and for the development of interoperable IT infrastructures and that it furthermore addresses the need for interoperability in heterogeneous E-Government environments. Based on a maturity model consisting of five stages and ranging from initial services to optimized business services, the authors present implementation phases for the development of a technical SOA infrastructure to reach the highest maturity level. The core infrastructure component is an ESB that connects the different information systems, services and data sources with each other. Services are incrementally integrated into the infrastructure, ranging from basic authentication services and service registry to collaborative and optimized business services such as document management and orchestration services.

In conclusion, the characteristics and benefits of service-oriented architectures address the needs for interoperability, loosely coupled integration and reusability in government and public procurement domains. The aforementioned examples show the awareness of the potentials and benefits of the SOA approach and emphasize its application in the context of electronic government and electronic public procurement.

[19] CCI = Chambers of Commerce and Industry; GOV = Government

4.4 Summary

This chapter defined service-oriented architecture as an approach to develop distributed IT system architectures. The core concept in this approach is the provision of capabilities encapsulated by services, that describe their purpose in a service contract and expose their functionality to consumers by service interfaces. Each service within an SOA is based on specific principles and characteristics, that promote for example loose coupling, interoperability, reusability, composability and autonomy. Hence, service-oriented architectures enable the composition of existing services into new business services and processes and facilitate interoperability of distributed and heterogeneous software systems that retain their autonomy and independence. Another important aspect is the alignment of business processes and software systems. This refers on the one hand to the identification of services based on an analysis of the targeted business domain and processes. On the other hand, existing services which provide specific units of business functionality can be composed and orchestrated into new business processes.

A set of standards has been identified for the identification and assembly of services. These provide means for business analysts and developers to model the business domain, identify services and implement these as reusable software components that can be orchestrated within executable business processes. Although the SOA approach is independent to a concrete implementation technology, Web services provide a suitable implementation platform for services. Additionally to services and processes, an SOA consists of several other components: Application GUIs providing access to software applications, that in turn provide the business functionality. An Enterprise Service Bus represents the core technology backbone of large SOA systems, enabling the integration and interconnection of several software systems.

The discussion about benefits and challenges of SOA showed that organizations can enhance their flexibility and agility as business processes can be adjusted to changing requirements. Moreover, SOA supports the integration of software systems within an organization (EAI) and between several collaborating organizations (B2BAI). However, despite all promises about potential benefits, a successful implementation of an SOA is related to a set of challenges, including the need of a common understanding of SOA and the definition of a long term strategy. In order to leverage its full interoperability potential, a great challenge is related to the heterogeneity of organizational structures, requiring a great effort to achieve semantic and organizational interoperability. To overcome this challenge, existing data standards and frameworks have been developed to realize interoperability on the technical, semantic and organizational level (e.g. UBL, ebXML, CENN/ISSS BII).

The promises of SOA concerning interoperability and flexibility makes this architecture interesting for large heterogeneous and distributed system environments. In the E-Government context, and in the provision of PEGS in particular, as well as public E-Procurement, several initiatives and research is conducted concerning the application of the SOA approach, each stating that it is suitable to realize interoperability, which is the most important requirement in the E-Government domain.

Chapter 5

PEPPOL WORK PACKAGE 2: THE VIRTUAL COMPANY DOSSIER

The project Pan European Public Procurement OnLine (PEPPOL) provides the practical background for this master thesis. In particular, the conceptualization and development of the prototype conducted within this master thesis is based on and related to the ongoing specifications of work package 2 of the PEPPOL project, that is concerned with the development of a so called Virtual Company Dossier (VCD). The next section will first provide a brief introduction to PEPPOL and its work packages. Afterwards, work package 2 and relevant aspects for the implemented prototype will be described in more detail.

5.1 The PEPPOL Project

5.1.1 Motivation, Vision and Objectives

The motivation of the PEPPOL project is to contribute to the targets of the eGovernment Action Plan of making 100% of procurement activities electronically available within the Member States of the European Union by 2010 (cf. sections 2.1.3 on page 8 and 2.1.4 on page 9). The main objective of the PEPPOL project thereby is the development of an interoperable environment for public E-Procurement to achieve electronic data exchange between companies and governmental procurement institutions throughout the entire procurement process. The pilot solution of the PEPPOL project will be built upon the existing national procurement systems and provides a European layer that facilitates interoperability between the national systems throughout the European Union. In result, this pilot will be a pan European electronic procurement system that supports companies to conduct procurement processes with any government in the European Union electronically.[1]

[1] See http://www.peppol.eu/About_PEPPOL (11.06.2010)

5.1.2 Overview of the PEPPOL Work Packages

The project is divided into eight work packages, each addressing specific domains of the public procurement domain as well as tasks of the project work. Figure 5.1 illustrates the main work packages (highlighted by a red border) that are directly concerned with public procurement domains.[2] Each work package, including the two not depicted in the figure will be briefly described in the following.

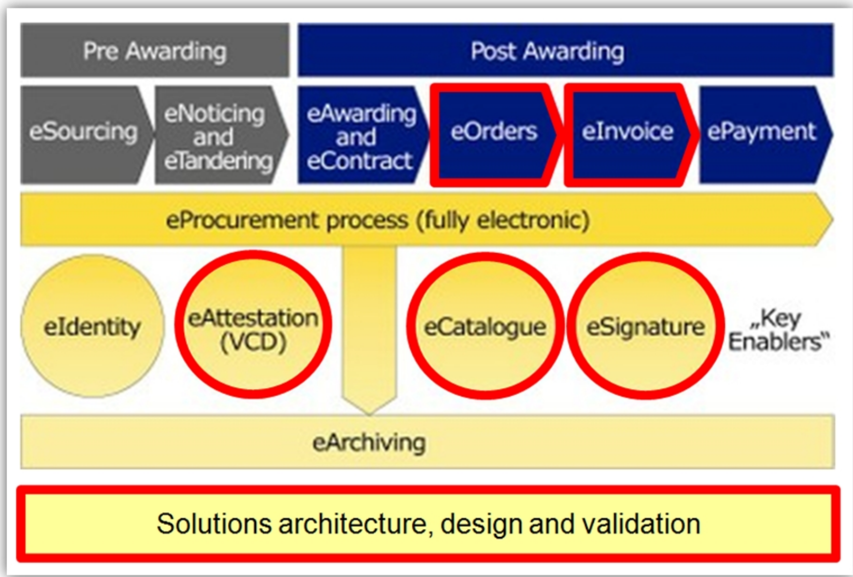

Figure 5.1: PEPPOL work packages within the E-Procurement process

WP1 - eSignature is concerned with specifying and piloting an interoperable eSignature solution that allows economic operators from any Member State to electronically sign procurement documents when submitting them to a contracting authority.[3] **WP2 - Virtual Company Dossier** is responsible for the development of a digital document container (i.e. the Virtual Company Dossier (VCD)) that enables economic operators to electronically collect certificates and attestations from corresponding issuing bodies and to submit them as a VCD package to any public awarding entity in the European Union.[4] **WP3 - eCatalogue** envisions a solution to use electronic catalogues in procurement procedures. Economic operators will be able to provide information about their products and prices that can be utilized in the pre- and post-awarding phase.[5] **WP4 - eOrdering** aims at implementing a so called European Ordering (EOI) framework facilitating electronic exchange of order documents (e.g. order

[2] Graphic adapted from http://www.peppol.eu/About_PEPPOL

[3] WP1 description: http://www.peppol.eu/work_in_progress/wp-1-esignature (11.06.2010)

[4] WP2 description: http://www.peppol.eu/work_in_progress/wp2-virtual-company-dossier (11.06.2010)

[5] WP3 description: http://www.peppol.eu/work_in_progress/wp3-eCatalogue (11.06.2010)

notification, delivery information) across EU member states between awarding entities and economic operators.[6] **WP5 - eInvoicing** is working on a solution that facilitates electronic invoicing, so that economic operators are able to send invoices electronically to any awarding entity throughout the European Union.[7] **WP6 - Project Management** is responsible for administrating, managing and coordinating the PEPPOL project and its work packages.[8] **WP7 - Dissemination, awareness and consensus building** is responsible for project marketing. This includes informing key stakeholders about the project and its progress as well as encouraging Member States to become members of the PEPPOL consortium.[9] **WP8 - Solutions architecture, design and validation** is piloting a technical infrastructure for the electronic exchange of documents between companies and governmental institutions involved in the European public procurement process.[10] This work package includes among other things the specification of the technical infrastructure for transport of business documents (PEPPOL e-business document transport infrastructure (PEDRI)).[11]

Because this master thesis is written in context of work package 2, the next section will provide more details about the Virtual Company Dossier by highlighting several issues that are relevant for the practical work of this master thesis.

5.2 WP2: The Virtual Company Dossier

5.2.1 Introduction

Work package 2 of the PEPPOL project targets the development of a digital document container, the so called Virtual Company Dossier (VCD) [Mondorf and Wimmer, 2009]. The idea behind the VCD is the creation of an electronic information package that contains all the documents an economic operator has to provide in the tendering phase to prove the economic operator's compliance with the qualification and selection criteria (see section 2.3.4 on page 15) claimed by a contracting authority. Furthermore, the VCD is accompanied by a supporting IT system that enables the collection of these evidence documents from corre-

[6] WP4 description: http://www.peppol.eu/work_in_progress/wp4-eordering (11.06.2010)

[7] WP5 description: http://www.peppol.eu/work_in_progress/wp5-einvoicing (11.06.2010)

[8] WP6 description: http://www.peppol.eu/work_in_progress/project-management (11.06.2010)

[9] WP7 description: http://www.peppol.eu/work_in_progress/consensus-and-awareness-building (11.06.2010)

[10] WP8 description: http://www.peppol.eu/work_in_progress/wp8-Solutions% 20architecture% 2C% 20design% 20and% 20validation (11.06.2010).

[11] Because it is related to the work conducted in this master thesis, a discussion about PEDRI will be part of section 10.3 on page 98 about related work.

sponding issuing bodies as well as electronic processing and exchange of a VCD.

The status quo of public tendering procedures that has been elaborated in [Mondorf and Wimmer, 2009] is depicted in figure 5.2 taken from the specification document. A brief description of the most relevant aspects will follow afterwards.

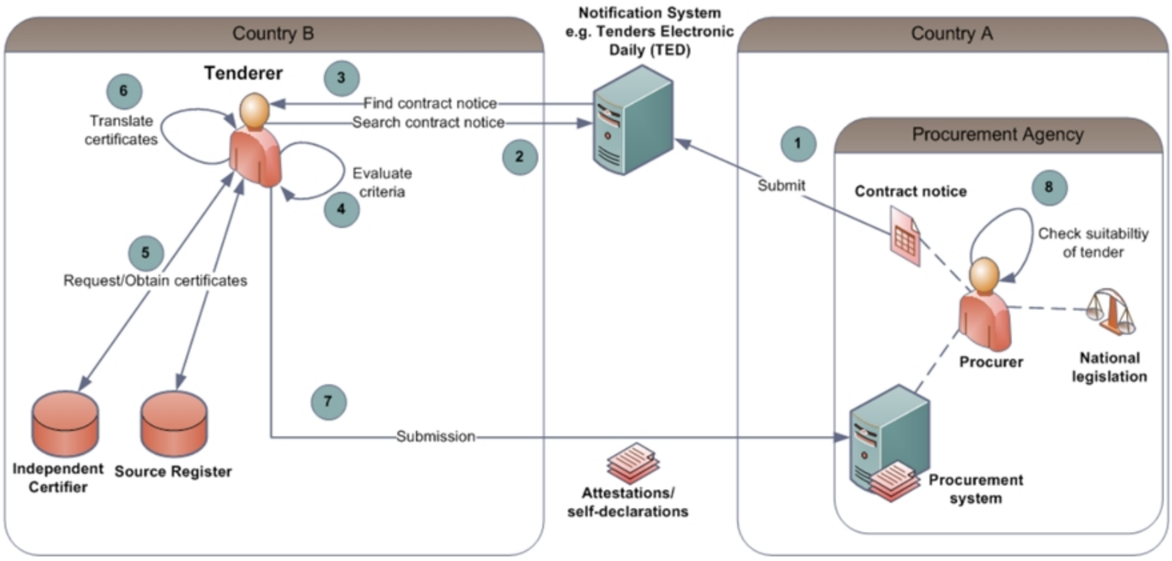

Figure 5.2: Status quo of public tendering procedures

A tendering procedure starts with the preparation of a call for tender by a contracting authority (CA). This call for tender consists of a contract notice published in a notification system such as Tenders Electronic Daily (TED), contract documents, technical specifications about the project that is subject of the contract as well as additional documents. Economic operators (EO) can search in the notification system for contract notices. Once decided to participate in the awarding of a specific contract, the EO can extract required information from the notification system, or request them from the CA. The most important information in the context of the VCD are the criteria of qualitative selection specified in the call for tender. These are the basis for EOs to manually collect evidence documents that prove compliance with the required criteria from corresponding issuing entities. The collection of evidences includes the identification of evidences issued in the EO's home country that are suitable to prove compliance with the criteria of the contracting authority located in a foreign country. After collecting the evidences (and corresponding translations) the economic operator submits the documents to the contracting authority, which can then check the suitability of the submitted tender documents.

The envisioned VCD solution will provide necessary means to allow the electronic exchange of evidence documents among member states, thereby fulfilling the need of contracting authorities to evaluate the compliance of economic operators based on the criteria of qualitative selection. As the European public procurement domain is characterized by a high complexity and heterogeneity, a staged maturity model has been proposed that allows a step-wise implementation of the VCD, ranging from basic interoperability and document exchange on lower levels to direct electronic exchange on the higher maturity stages [Mondorf and Wimmer, 2009]. A brief overview of these stages will be given in the next section.

5.2.2 The VCD Maturity Model

On **stage one (pre-VCD mapping)**, support for the collection of evidence documents is introduced. This is achieved by a pre-VCD mapping tool that allows an economic operator to automatically identify which evidences issued in his country are suitable to prove compliance to the criteria claimed by a contracting authority located in another country. Hence, this stage provides basic interoperability through a mapping of national criteria of one member state to corresponding national evidences of another member state. The mapping tool is hosted and provided by a European Service Provider (ESP). The corresponding extension of the above status-quo picture in figure 5.2 on the preceding page is illustrated in figure 5.3.

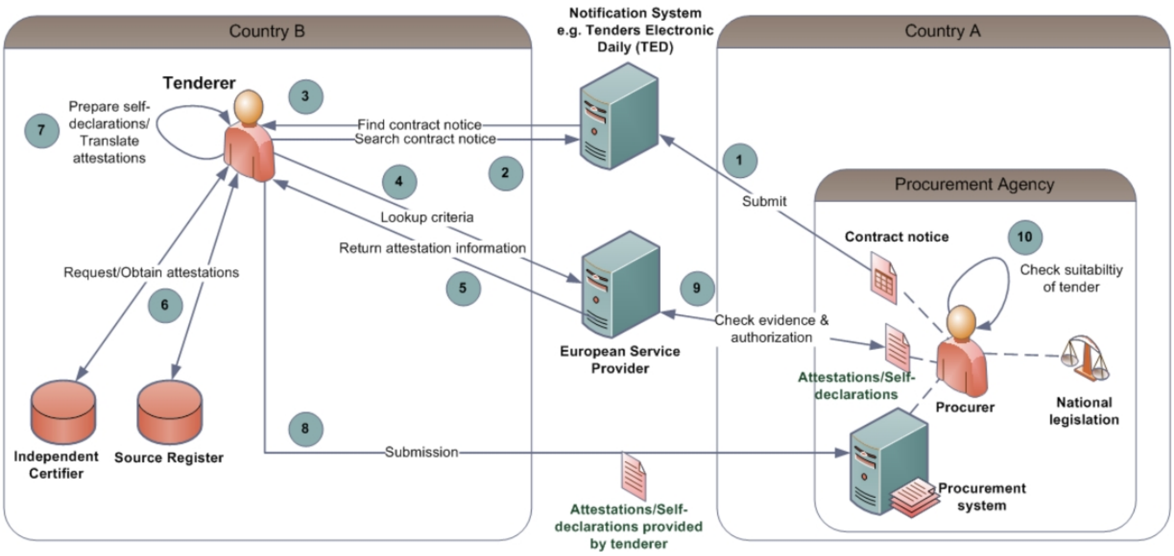

Figure 5.3: Stage 1 - Pre-VCD mapping tool

Steps 4 and 5 in this model illustrate the mapping: An economic operator requests a criteria lookup from the ESP which returns information about the evidences that have to be provided to prove compliance to these criteria. The collection and packaging of these evidences still has to be conducted manually by the economic operator. Another extension is represented by step 9, allowing contracting authorities to check evidences received from an economic operator via the VCD mapping tool (cf. [Mondorf et al., 2010; Mondorf and Wimmer, 2009]).

Stage 2 (VCD simple Package) further extends stage 1 by introducing a National Service Provider (NSP) that provides the service of automatically retrieve evidence documents from directly connected issuing bodies. Being the single point of access for an economic operator, the national VCD system takes the request to map criteria from an economic operator, forwards it to the ESP and receives the returned VCD Skeleton Container. The required evidences are then collected by the NSP by requesting issuing bodies to create and deliver them. Missing evidences that cannot be collected automatically, for example because a corresponding issuing body is not directly connected to the NSP, have to be collected manually by the economic operator. Once the NSP has received all evidences, either automatically or manually collected by the economic operator, the national VCD system can compile a VCD Package (i.e. packaging of evidences into the VCD Skeleton Container).

Stage 3 (VCD advanced Package) adds three capabilities: The first enables re-compiling of existing VCD Packages, for example to change existing evidences in a VCD Package. The second enables combining several VCD Packages of different economic operators in case they form a bidding consortium. In this case relationships among the economic operators have to be retained in the package structure, for example a company having several subcontractors. The third extension allows the inclusion of so called context-specific data that is extracted from evidences, thus enabling quick access to the most important information provided by an evidence document.

Stage 4 (VCD network Package) provides ideas about possible enhancements of the VCD system to create additional value. This includes for example the creation of a VCD network in that evidences and attestations can be directly accessed, implying that a VCD Package contains references to evidence documents instead of binary data. This stage proposes the scenario of storing all required documents in a secure data base hosted by the National Service Provider. All documents can then be accessed on demand by the involved actors and system components. Another enhancement would be the implementation of Article 52 of the Public Sector Directive (see section 2.3.4 on page 15) as an official list of approved economic operators may be maintained by the National Service Provider that enables the inclusion of corresponding validation data into a VCD Package. Although such a scenario would provide the most sophisticated stage of maturity, it raises the complexity and costs of the overall VCD system while providing limited added value, resulting in a negative cost-benefit ratio [Mondorf et al., 2010].

After this basic introduction to the VCD system, the next part of this thesis will describe the developed prototype, including technical details about components based on the VCD system specification. Inter-relations to the specification elaborated within work package 2 will be highlighted at the appropriate locations.

PART II

PRACTICAL WORK

Chapter 6

INTRODUCTION

Before detailing technical issues concerning the infrastructure and the prototype implementation, this chapter will provide an introduction of underlying issues of the development work. First, the scope of this project as well as implementation objectives will be defined. Afterwards, the development methodology providing a process model of the development tasks being conducted will be described. Finally, a basic scenario that defines a practical use case for the European public procurement domain will be elaborated.

6.1 Project Scope

The overall scope of the implementation work conducted within this master thesis is to apply the aspects discussed in part one of this text. This means in particular the application of an approach following the SOA paradigm as discussed in chapter 4. The background is provided by the PEPPOL project and work package 2 concerning the development of the Virtual Company Dossier. As a result, the project work and the deliverables (see [Mondorf et al., 2010; Mondorf and Wimmer, 2009]) of this work package provided the information input about the target domain, business processes, system components and requirements. Based on the overall VCD system domain, a sub-system shall be identified for which the service-oriented approach will be applied. Besides the identification, design and implementation of sub-system components and services, this includes a brief evaluation and selection of technical infrastructure components. The resulting prototype shall simulate the interaction of system users and components within a sub-domain of the overall VCD public procurement domain.

6.2 Implementation Objectives

Because the aim of this prototype is primarily the identification of different components of the scoped sub-system and the simulation of their interactions, the implemented components shall (and can) not encapsulate detailed business functionality. They rather shall provide basic functionality so that a user of the system is able to provide relevant input data and receive meaningful output data. Furthermore, the components can be used to implement additional components that are composed and orchestrated of other components.

In accordance to the aforementioned project scope and based on the SOA concepts and components identified in chapter 4.2, table 6.1 defines the set of implementation objectives for the prototype implementation.

No.	Description
OBJ-1	Based on the WP2 deliverables, a sub-domain shall be identified as the target domain for the prototype.
OBJ-2	The system components as well as their functional and non-functional requirements shall be derived from the project deliverables.
OBJ-3	The top-down approach to identify and assemble services for that target domain shall be applied.
OBJ-4	A set of components, implementation technologies and existing standards suitable for the development of service-oriented architectures shall be used.
OBJ-5	The implemented system components shall not provide the full business functionality as specified in the project deliverables.
OBJ-6	The system components shall simulate basic functionality allowing users to provide input data and receive meaningful output data.
OBJ-7	A use case scenario derived from the European public procurement domain shall be developed in order to simulate the system behavior.

Table 6.1: Prototype implementation objectives

6.3 Development Methodology

The methodology to develop the prototype follows the general top-down approach of service identification and assembling as described in section 4.2.2 on page 44. The process model depicting the development tasks is illustrated in figure 6.1.

The development process starts with an identification of services based on an analysis and design of the target domain, scoped business processes and use cases. Input for this phase is provided by the deliverables of work package 2 ([Mondorf et al., 2010; Mondorf and Wimmer, 2009]), from which a sub-domain will be derived. The identification of services in that target domain will be based on developed models using the modeling standards BPMN,

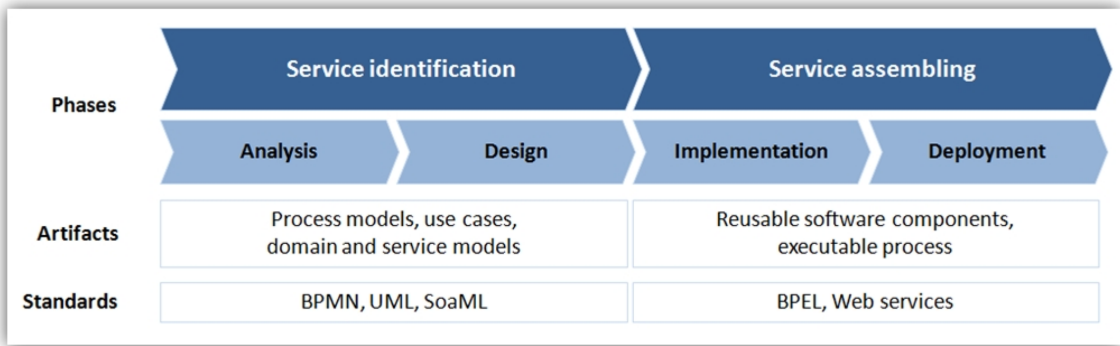

Figure 6.1: Development Tasks

UML and SoaML followed by a specification of the software components, requirements and data models as well as an identification of the required implementation technology and technical infrastructure (i.e. programming language, software tools, application server, business process runtime engine etc.). This specification will be input for the service assembling phase. This consists of the implementation, deployment and testing of the identified software components (i.e. web services, web applications, service orchestration and execution, other software components). The result of this phase will be the running prototype system.

6.4 Use Case Scenario

In order to provide meaningful input and output data, a basic use case scenario has been elaborated[1]: It defines a simplified copy of a realistic procurement use case and is derived from the WP2 specification of the Virtual Company Dossier. It defines main actors, data models and concrete document objects that are created and used within the VCD system.

The contracting authority "Landesbetrieb Straßenbau NRW Planungs- und Baucenter" (Address: Henri Dunant Straße 9 - 45131 Essen - Germany) publishes a contract notice on the TED platform.[2] The bidding consortium consisting of the following three (fictive) economic operators participates in the award procedure:

- Leader tenderer: HochTief Bau AG, Geothstraße 10a - 45131 Essen - Germany
- Other tenderer 1: Costruzione di strade S.R.L, Via Milano 87 - 20100 Milan - Italy
- Other tenderer 2: Veibygging BA, Tollbugata 4 - 0155 Oslo - Norway

[1] More information can be found in the specification document on the enclosed DVD.

[2] See http://ted.europa.eu/udl?uri=TED:NOTICE:153423-2010:TEXT:EN:HTML&tabId=0

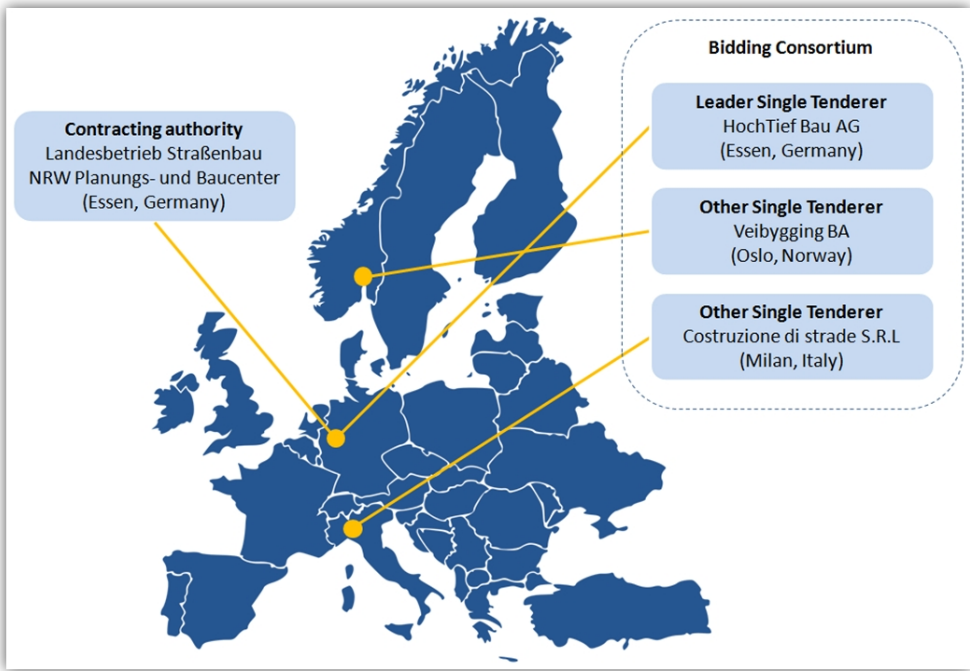

Figure 6.2: Geographical scenario overview

According to the contract notice, the conditions for participation (qualification and selection criteria) concerning the legislation of the contracting authority's home country are:

- Personal situation of economic operators, including requirements relating to enrolment on professional or trade registers Information and formalities necessary for evaluating if requirements are met: Gemäß VOB/A §8 Nr. 3 (1) Buchstabe: f.
- Economic and financial capacity Information and formalities necessary for evaluating if requirements are met: Gemäß VOB/A §8 Nr. 3 (1) Buchstabe: a,b,c.
- Personal situation / absence of conviction (own criterion added).

The mapping of these criteria and their corresponding required national evidences to the European legislation leads to the following atomic European criteria:[3]

- Criterion "has been convicted by final judgement of the participation in a criminal organization".
- Criterion "has official enrollment on one of the professional or trade registers in its member state".
- Criterion "has balance sheet or extracts from it".

[3] More information on the underlying criteria mapping can be found in the specification document.

Chapter 7

IMPLEMENTATION TECHNOLOGIES AND TOOLS

For the prototype design and implementation according to the aforementioned process model, several software tools, modeling notations and implementation technologies have been used. The modeling notations Business Process Modeling Notation (BPMN), Unified Modeling Language (UML) and Service oriented architecture Modeling Language (SoaML) used within the service identification phase have already been classified as suitable in section 4.2.3 on page 45. This section will now focus on the implementation tools and technologies, including a brief evaluation of alternatives as well as reasons for the selection of some of these.

7.1 Identification of Technologies and Tools

Based on a conducted web search, a set of possibly suitable technologies and tools for the development of SOA components as described in section 4.2.1 on page 43 has been identified. Table 7.1 contains general requirements for the selection of technologies that has been applied during the search.

Although technologies from different vendors may be combined, RE-5 has been defined to minimize the effort of installing and configuring the used technologies. This means, that choosing a set of technologies that are delivered as bundles or independent components from the same vendor allows the integration and efficient inter-operation of the installed components. All of these requirements allowed an early selection of three major product bundles:

- The SOPERA Advanced Service Factory (ASF),
- The JBoss Community Projects and
- The Intalio Business Process Management Suite (BPMS).

No.	Description
RE-1	The implementation platform of choice for this prototype is Java Platform, Enterprise Edition (JEE), therefor further technologies and tools must support the development of Java enterprise applications.
RE-2	The implementation technologies must allow the development, implementation and deployment of web services, their orchestration into executable processes as well as web applications that provide business functionality and graphical user interfaces.
RE-3	An Enterprise Service Bus (ESB) shall be installed to provide the integration layer.
RE-4	Sufficient documentation and online learning material must be available for each product.
RE-5	The tools and technologies shall be available as product bundles or a set of independent products that can be used together allowing their integration.
RE-6	All tools and technologies must be available as free or open source software.

Table 7.1: Requirements for the selection of implementation technologies

A brief feature overview of the selected candidate products will be part of the following sections.

SOPERA Advanced Service Factory (ASF)

The SOPERA Advanced Service Factory (ASF) is a platform for service-oriented architectures providing features such as an Enterprise Service Bus, service development, business process management, application and data integration as well as service and system management.[1] The ASF is delivered as an open source *Community Edition* as well as a commercial *Enterprise Edition*. The Community Edition can be downloaded from the vendor's website as an "all in one" bundle.

The major components are SOPERA ESB[2] and SOPERA ToolSuite[3]. The SOPERA ESB provides the technological backbone that allows for example messaging and communication between services as well as process, application and data integration.

---

1 SOPERA website available at http://www.sopera.de/.

2 See http://www.sopera.de/produkte/sopera-features/enterprise-service-bus/ for more information.

3 See http://www.sopera.de/downloads/sopera-asf-33/toolsuite-33-allinone-binaries-fuer-windows/ for more information

The SOPERA ToolSuite is an Eclipse-based software application which provides as the main features the development, deployment and management of services. The development of services is supported by components that allow the graphical modeling of web service interfaces by using the Eclipse integrated WSDL editor. Based on such a web service description, the SOPERA Code Generator allows the automatic generation of Java source code and automatic installation of services in the SOPERA ESB. The ToolSuite's administration user interface furthermore provides means to configure and manage the technical SOPERA infrastructure, such as the service registry. Adapters allow connections to other third party software. Interesting in this context is the Intalio Process Adapter that enables the orchestration of SOPERA services within business processes modeled with the Intalio Process Designer.[4]

All in all, the SOPERA ASF provides a full set of technical components to implement and manage a service-oriented architecture bundled in one product. The SOPERA ToolSuite furthermore provides an Eclipse-based software application for the development of Java-based services as well as infrastructure management.

JBoss Community Projects

The JBoss Community Projects provide a large set of open source technologies for the development of applications on the Java Platform, Enterprise Edition (JEE)[5]. A set of these projects provides the following technologies relevant in the context of service-oriented architectures that are suitable for the development of the targeted prototype:

JBoss Application Server [6] is an open source application server to develop and deploy Java enterprise and web applications. It provides the technical platform to assemble other JBoss components and technologies, such as Web Services, Seam, Richfaces, Enterprise Service Bus and Business Process Management.

JBoss Seam [7] is a framework for the development of web applications. It allows the implementation of rich internet applications by integrating other technologies such as Asynchronous JavaScript and XML (AJAX), JavaServer Faces (JSF), Java Persistence API (JPA), Enterprise Java Beans (EJB) 3.0 and jBPM.

JBoss RichFaces [8] allows the development of graphical user interfaces for Seam web applications. It provides a library of user interface components such as complex tables,

---

[4] More information about the Intalio Business Process Management Suite will follow later.

[5] See http://www.jboss.org/projects for more information.

[6] Website available at http://www.jboss.org/jbossas.

[7] Website available at http://seamframework.org/.

data grids and AJAX components. Together with JBoss Seam and the integration of JSF and AJAX it allows the development of rich internet applications.

JBoss Web Services [9] provides a Java API for XML Web Services (JAX-WS) framework for the JBoss Application Server. It enables the development of web services and their deployment in the JBoss Application Server. JAX-WS allows the convenient development of Java web services by using annotations within simple Java classes. Furthermore, Apache jUDDI[10] is integrated into the JBoss Web Services distribution.

JBoss jBPM [11] is a business process management suite consisting of a graphical process modeler and a process runtime engine. Business processes can be modeled with the process definition language jPDL. Through the integration of jBPM and JBoss Seam it is possible to develop web applications that allow interactions with a process instance. Furthermore, it is possible to attach user interface forms to specific tasks of a process, through which a user can access and edit process data. Additionally, BPEL support is achieved through a jBPM-BPEL add-on allowing orchestration of web services.

JBoss Enterprise Service Bus [12] is an open source Enterprise Service Bus. The core components in that ESB are services, that encapsulate business logic. The internal and external interaction with the ESB is achieved through the exchange of messages (e.g. via HTTP, FTP or SOAP). Event listeners and actions thereby manage the transport of these messages. Furthermore, it allows the integration of other JBoss components, such as jBPM or JBoss web services.

Intalio Business Process Management Suite (BPMS)

The Intalio Business Process Management Suite[13] consists of two components: The Intalio Designer, which is an Eclipse-based software application that allows the graphical modeling of business processes using BPMN, and the Intalio Process Server, which is the process runtime engine on which the modeled processes can be deployed as WS-BPEL processes. The transformation from BPMN to BPEL thereby is performed automatically during deployment.

[8] Website available at http://www.jboss.org/richfaces.

[9] Website available at http://www.jboss.org/jbossws.

[10] Apache jUDDI is an open source Java implementation of the Universal Description, Discovery, and Integration specification for Web Services. More information can be found at http://ws.apache.org/juddi/

[11] Website available at http://www.jboss.org/jbpm.

[12] Website available at http://www.jboss.org/jbossesb.html. Further information can be found at http://www.jboss.org/jbossesb/presentations.html

[13] See http://community.intalio.com/ for more information

The Intalio Designer allows orchestration of web services by attaching operations defined in a WSDL document to specific tasks of a BPMN process model. A Data Mapper also allows the mapping of data elements (e.g. request and response data of web service calls) that are attached to message flow connectors. User interaction with a process can be achieved through the design of task forms that allow a user to edit data of a specific process instance.

The different products and technologies described above form the final set of candidates for the selection of the development toolkit. An evaluation based on a set of criteria will be conducted in the following section.

7.2 Evaluation and Selection

Based on the general requirements for the selection of candidate technologies and tools in table 7.1 on page 72, a set of technical and qualitative evaluation criteria has been defined. Table 7.2 describes these criteria and their meaning.

The technical criteria are only expressed in terms of "yes" and "no" referring to the existence or non-existence of each criterion. For the qualitative criteria, a simple rating scale ranging from 1 to 3 has been used to rate each product by the arithmetic mean of the ratings for each criterion (the higher the final mean value, the better the rating).

All candidate technologies have been checked against these criteria by testing the products and their features. It should be noticed, that a detailed evaluation of each product could not be conducted due to limited time constraints. However, the product tests included the installation of the required product components, a basic analysis of the major product features, an implementation of simple software components following user tutorials that exist for each product as well as an analysis of existing product documentation. The evaluation results are shown in figure 7.1 on page 78.

The SOPERA ASF has been excluded because of its complexity and the missing support to develop rich internet applications. The support for BPM capabilities such as process modeling and execution is not integrated into the SOPERA Community Edition, but can easily be achieved through the installation of the Intalio Process Adapter. SOPERA is primarily useful to implement the technological backbone of an SOA, including an ESB as well as the development, deployment and management of services. A positive product feature is the automatic code generation for SOPERA services from WSDL documents and their deployment in the SOPERA service registry. With the Intalio adapter installed, the SOPERA service registry can be accessed within the Intalio Designer. For the development of rich internet applications, additional software and technologies are required. All in all, SOPERA provides a sophisticated product bundle for the installation, development and management

Criterion	Description
Technical evaluation criteria	
BPMN support	The support of the tool / technology to model business processes in BPMN.
WS-BPEL support	The support of the tool to orchestrate web services within executable BPEL processes.
RIA development	Ability to develop rich internet applications.
ESB	Ability to install an Enterprise Service Bus (ESB).
Qualitative evaluation criteria	
Learning effort	Refers to the amount of work related to learning how to use, implement, install or configure the technology or tool. Rating scale: 1 = high, 2 = medium, 3 = low effort.
Installation and configuration effort	The amount of work necessary for installation and configuration. Rating scale: 1 = high, 2 = medium, 3 = low effort.
Complexity	The complexity of the technology or tool (e.g. in terms of the number of features, sub-components, required technologies, need to install other components). Rating scale: 1 = high, 2 = medium, 3 = low complexity.
Usability	Related to complexity; Refers to the usability of the user interface of a tool. Rating scale: 1 = low, 2 = medium, 3 = high usability.
Documentation and support	Availability of information, documentation, tutorials, online communities and forums. Rating scale: 1 = low, 2 = medium, 3 = high support

Table 7.2: Technical and qualitative evaluation criteria

of a SOA infrastructure consisting of an ESB, services and business process modeling and execution capabilities. However, the high complexity of this SOA bundle requires a large amount of learning how to use all of the product features, which finally was the reason for rejecting the SOPERA ASF.

The JBoss Community Projects provides a large set of SOA related products that can be combined with each other and generally provide capabilities similar to the SOPERA ASF. One major benefit and advantage compared to SOPERA is the possibility to develop rich internet applications by combining the JBoss Application Server, JBoss Seam and JBoss Richfaces. Furthermore, JBoss Web service provides means for the convenient implementation and deployment of web services based on JAX-WS. Business process modeling and execution capabilities can be integrated through jBPM, and the JBoss Enterprise Service Bus provides the core integration and communication component. Hence, this set of JBoss products covers the main SOA components that can be combined with each other flexibly: ESB, business process modeling and execution, web services and rich internet applications. For the implementation of the prototype, not all components have been chosen for the following reasons: The (currently) missing capability to model business processes with BPMN and comfortable orchestration of web services within these processes lead to the exclusion of JBoss jBPM. Especially in comparison to the Intalio BPMS, jBPM was classified as less suitable primarily due to the convenience and features the Intalio BPMS provides. The complexity of the JBoss ESB in terms of development of ESB services and integration of JBoss Web services have been the main reasons to reject the installation of this component. JBoss ESB services follow, similar to the SOPERA services, a different development approach as the development and deployment of JAX-WS based web services. Although the corresponding service development documentation provides the required knowledge, learning how to develop such services is related to a higher effort as for the JAX-WS based approach. A second reason why the JBoss ESB has not been chosen is related to the scenario and scope of the developed prototype, which simply does not require an Enterprise Service Bus due to the SOA maturity of the corresponding IT system.

The Intalio BPMS has been chosen as it provides a sophisticated and easy to use business process modeling and execution solution. The major features providing benefits in comparison to jBPM are the graphical process designer and process engine, BPMN support, automatic conversion into WS-BPEL processes and convenient means to orchestrate web services. Furthermore, a large set of documentation sources and online support are available via the Intalio Community.

Finally, the set of technologies and tools chosen to develop the prototype consists of the following products:

- JBoss Application Server (Version 4.2.3),
- JBoss Seam (Version 2.2.0),
- JBoss Richfaces (Version 3.3.3),
- JBoss Web Services (Version jbossws-native-3.1.1) and
- Intalio Business Process Management Suite (Version 6.0.3).

	Technical criteria				Qualitative criteria					
	BPMN support	WS-BPEL support	RIA development	ESB	Learning effort	Installation / configuration effort	Complexity	Usability	Documentation / Support	Qual. rating
SOPERA ASF	Yes (with Intalio Adapter)	Yes (with Intalio Adapater)	No	Yes	1	2	1	3	2	**1,8**
Jboss	No	No	Yes	Yes	2,6	2,8	2,3	2	3	**2,54**
AS	-	-	Yes	-	3	3	3	-	3	3
Seam	-	-	Yes	-	2	3	2	-	3	2,5
Richfaces	-	-	Yes	-	3	3	3	-	3	3
WS	-	-	-	-	3	3	3	-	3	3
jBPM	Not yet supported	Not sufficient, only with BPEL plugin	-	-	2	2	2	2	3	2,2
ESB	-	-	-	Yes	3	3	1	-	3	2,5
Intalio BPMS	Yes	Yes	-	-	2,5	3	3	3	3	**2,9**

Figure 7.1: Evaluation matrix

Chapter 8

ANALYSIS AND DESIGN

This chapter will now provide an overview of the important specifications and results of the analysis and design phase. The focus thereby is only on the most important issues that provide an overview of the system specification as well as additional information about the conducted work that are not part of the specification. The content of this chapter therefore does not substitute the content of the specification document (or vice versa). They rather provide a complementary description of the conducted work and its results. All technical details are described in the specification document on the enclosed DVD.

8.1 System Domain

The overall system domain for the VCD is specified in the deliverables of work package 2 ([Mondorf et al., 2010; Mondorf and Wimmer, 2009]). Starting point for the narrowing of that domain for the targeted prototype was the modeling of the business processes related to the VCD maturity stages as introduced in section 5.2 on page 62. The scoped sub-system of this prototype focuses on the tasks performed by economic operators after receiving a call for tender, including the generation of a VCD package skeleton and the compilation of a VCD package. The sub-system is specified by a general process model including the main tasks performed by an economic operator as well as corresponding use case diagrams and descriptions. The process model is derived from the WP2 specification process model for the VCD maturity stage one. This process contains the interactions between an economic operator and the VCD system to generate a VCD Package that serves as the tender document container. A step independent of this process and therefore not specifically modeled is the visualization of the VCD Package content. Figure 8.1 illustrates the general process model of the VCD sub domain.

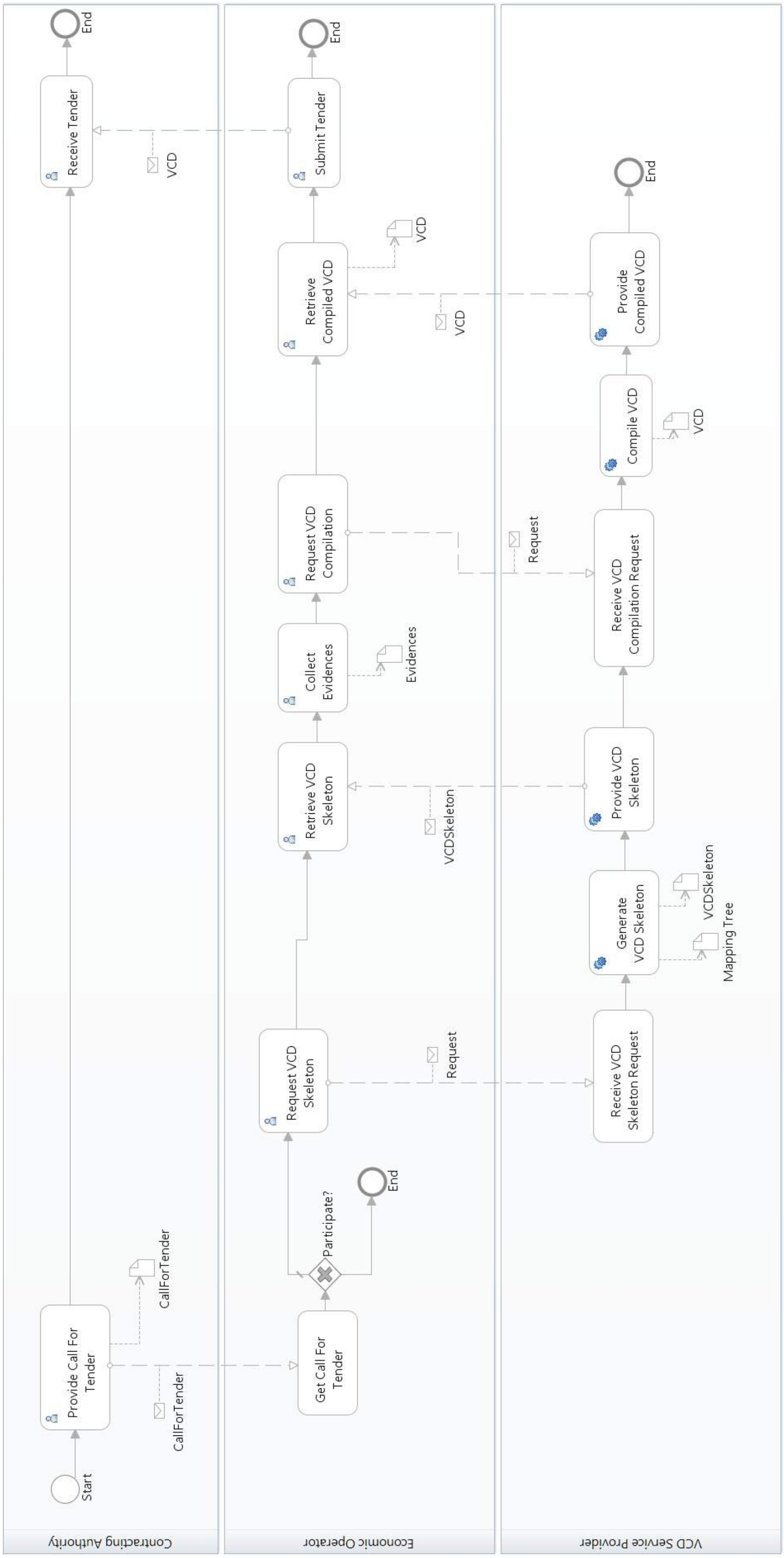

Figure 8.1: General process model of the VCD sub domain

The start task in the focused sub-system is the request to generate a VCD skeleton of an economic operator. The end task is the receipt of a compiled VCD package. Hence, the pool of the contracting authority is out of scope of this sub-system, and only part of this model in order to illustrate the connecting points to the overall tendering process. However, a contracting authority is one actor within the use case to visualize the content of a VCD package.

Based on this general process model, a ServicesArchitecture diagram has been derived as illustrated in figure 8.2. It provides a high-level view on the services and interactions between service providers (VCD system components) and service consumers (economic operators and contracting authorities) for the creation of a Virtual Company Dossier.

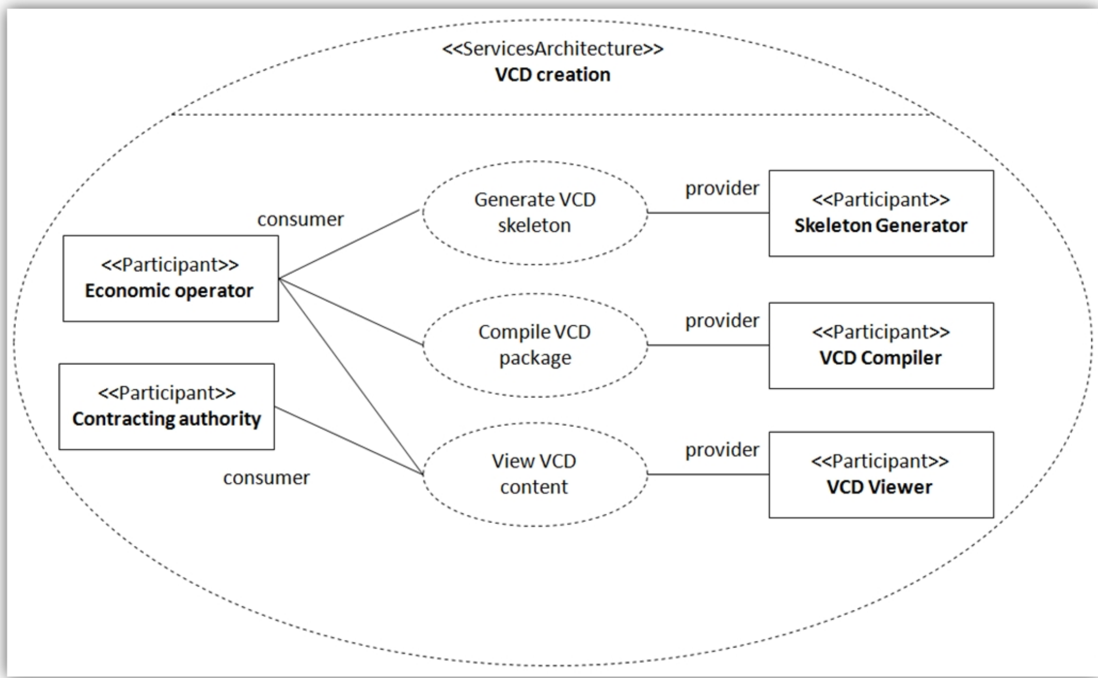

Figure 8.2: ServicesArchitecture diagram of the VCD System

In order to further detail the target domain and to identify system components, several Use Case descriptions and diagrams have been derived from the aforementioned process model and ServicesArchitecture. Table 8.1 provides an overview of these use cases.[1]

[1] The specification document on the enclosed DVD provides further information.

Use case	Description
Generate VCD Package Skeleton	The Skeleton Generator component provides the service to generate a VCD Skeleton Package based on input data provided by an economic operator, such as a Call For Tender document.
Compile VCD Package	After providing a VCD Package Skeleton, an economic operator can upload required evidence documents that are packaged into a VCD Package.
Automatic creation of a VCD Package	This use case combines the two use cases above and provides the capability to automatically generate a VCD Package, including the automatic collection of evidence documents stored in the system data store.
View VCD Package	Economic operators and contracting authorities can visualize the content of a VCD Package. Primarily, this use case addresses contracting authorities to support them during the evaluation of received tenders.

Table 8.1: Overview of the system's use cases

8.2 System Components

The system components derived from the previously described models are divided into three categories:

1. User interface components that provide the graphical user interfaces allowing users to interact with the system,
2. functional components that provide reusable business functionality and
3. utility components that provide basic functionality that can be used by other components.

Table 8.2 provides an overview of the different components according to this classification. The corresponding component diagram is depicted in figure 8.3.

Name	Description
User interface components	
VCD Viewer Application	Provides the graphical user interface and functionality to view the content of VCD Packages.
VCD Skeleton Generator Application	Provides the graphical user interface and functionality to provide required input (such as a Call for Tender Document) and to generate a VCD Package Skeleton from this data.
VCD Compiler Application	Provides the graphical user interface and functionality to upload a VCD Package Skeleton and required evidence documents as specified in the VCD Package Skeleton. The uploaded evidence documents are then packaged into a VCD Package.
Functional components	
VCD Skeleton Generator	Provides the functionality to generate a VCD Skeleton based on provided input data.
VCD Compiler	Provides the functionality to package evidence documents into a VCD Package.
VCD Doc. Manager	Provides the functionality to persist and retrieve documents from the data store.
Utility components	
User management	Provides user management functionality such as storing user data and checking user credentials.
Data store	Provides functionality to store data and files in the system's data store and to retrieve them.

Table 8.2: System components

The user interface components consist of the user interface itself, i.e. the interface between the application and a user, and a controller part that encapsulates the functionality. These controller parts access the functional components and utility components through their interfaces. Both VCDSkeletonGeneratorService and VCDCompilerService use the interface provided by the VCDDocumentManagerService. The utility component User management provides interfaces for the authentication and registration of system users. Both VCDDocumentManagerService and UserManagementService rely on a data store to persist objects and meta data. This functionality is provided by the Data store component, that consists of a MySQL database and a file system. The functional components and the utility component UserManagementService will be implemented as web services, the user interfaces will be provided through web applications and the data store as a MySQL database with an additional file system to store binary data that is related to the objects stored in the database.

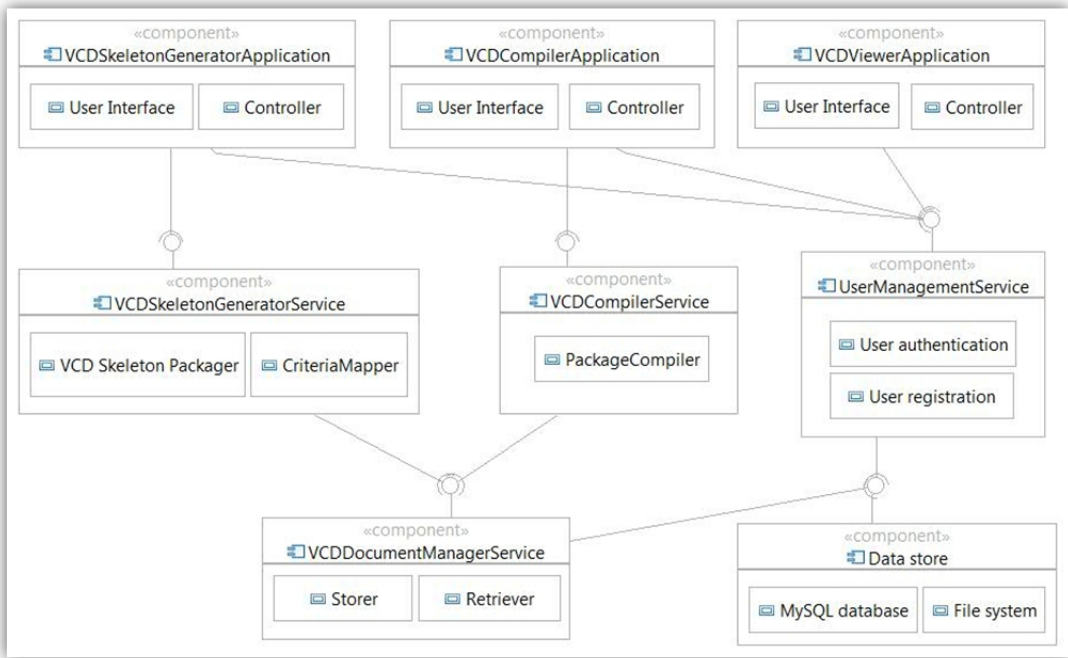

Figure 8.3: Component diagram

8.3 Data Formats

The most important documents exchanged within the prototype system are Call for Tender documents as well as VCDs and VCD Packages.

The **Call for Tender** is the document that a contracting authority sends to economic operators to invite them to participate in an awarding procedure for a specific contract. Hence, it contains information about the procurement project and tendering terms, such as the required qualification and selection criteria. The XML schema[2] is derived from the CEN BII profile *BII22 - Call for Tender*[3] and contains a subset of data elements that are most relevant and important for this prototype. Figure 8.4 illustrates the data format[4].

Besides general meta data about the call for tender, the schema defines data elements to describe the issuing contracting authority as well as the receiving economic operator. The most important data element is the element `TenderingTerms` that contains the set of criteria for qualitative selection (`RequiredCriteria`) for which the receiving economic operator has to provide appropriate evidence documents. A Call for Tender document contains the input data for the generation of a VCD Package Skeleton.

---

[2] Defined in file `CallForTender.xsd` in the folder `prototype/specification/data-schema/` on the enclosed DVD.

[3] As specified in http://spec.cenbii.eu/Profiles/IndexWG1.html.

[4] Graphic taken from Altova XMLSpy.

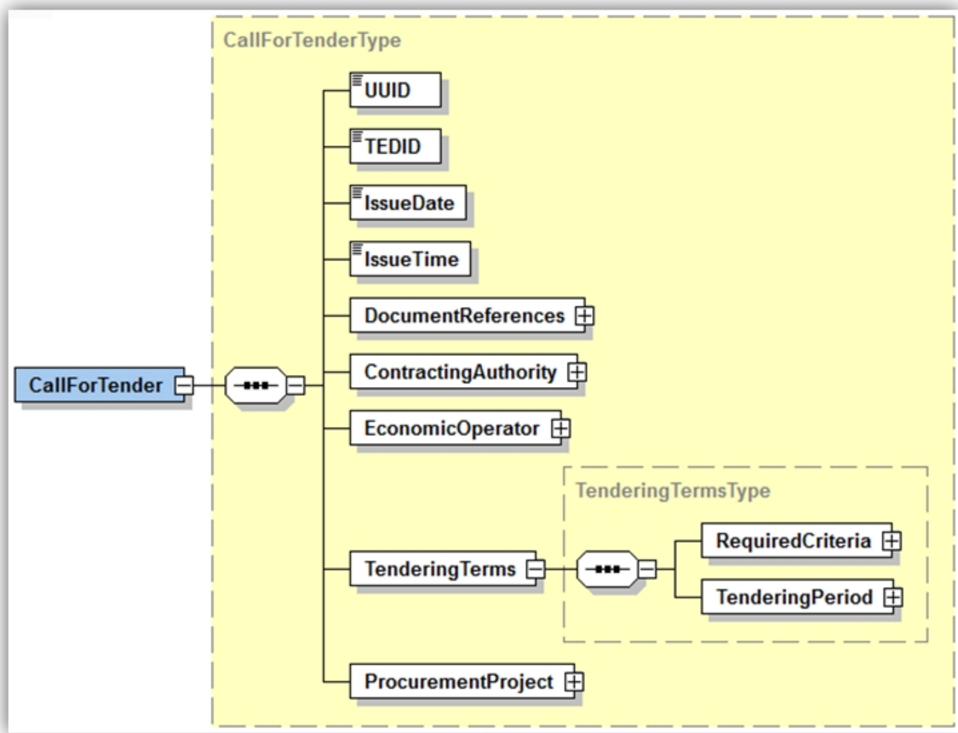

Figure 8.4: Call for Tender XML schema

Both **VCD** and **VCD Package** consist of an XML file holding meta-data that describes the content of a VCD or VCD Package. A VCD furthermore may contain several files (i.e. evidence documents, additional documents, etc.), whereas a VCD Package contains one or more VCDs.

The XML schemes for the XML meta-data files of a VCD and a VCD Package[5] are based on the schemes defined within WP2 and contain some of the data elements that are most important for this prototype. Figures 8.5 and 8.6 illustrate the general data format of a VCD and a VCD Package XML meda-data file.[6]

The VCD XML meta-data document describes one single economic operator, the criteria it has to comply with (sub-element of `EconomicOperator`) as well as corresponding evidence documents, that may be embedded as binary data or referenced by an URI pointing to external files (element `Evidences`). A VCD is delivered in a folder structure that contains the main VCD XML document and may contain further files (e.g. evidence documents that are not embedded as binary data inside the XML meta-data file).

[5] Defined in files `VCD.xsd` and `VCDPackage.xsd` in the folder `prototype/specification/data-schema/`

[6] Graphic taken from Altova XMLSpy.

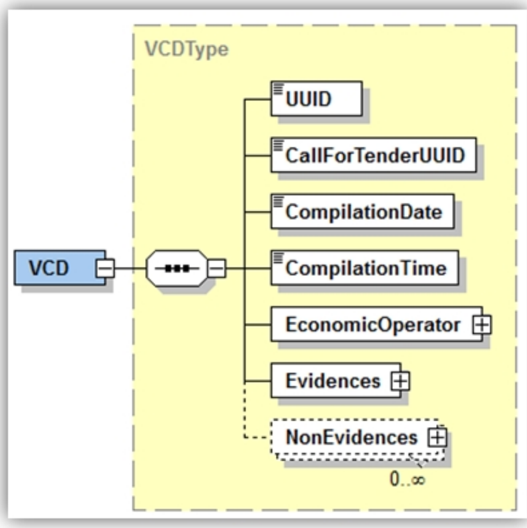

Figure 8.5: VCD XML schema

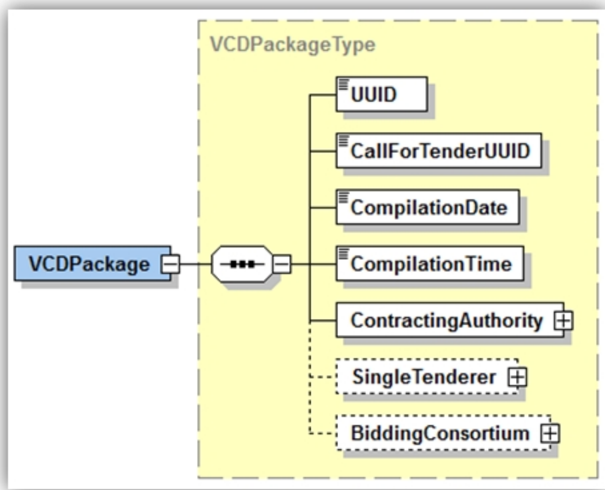

Figure 8.6: VCD Package XML schema

The VCD Package XML meta-data file describes, among other things, the tenderer structure consisting of the number and relationships of economic operators (e.g. tenderer and sub-contractor) preparing a bid (elements `SingleTenderer` and `BiddingConsortium`). A VCD Package is delivered as a zip archive containing the XML meta-data file along with the VCDs of one or more economic operators (depending on the tenderer structure). The general structure of a VCD Package is depicted in figure 8.7.

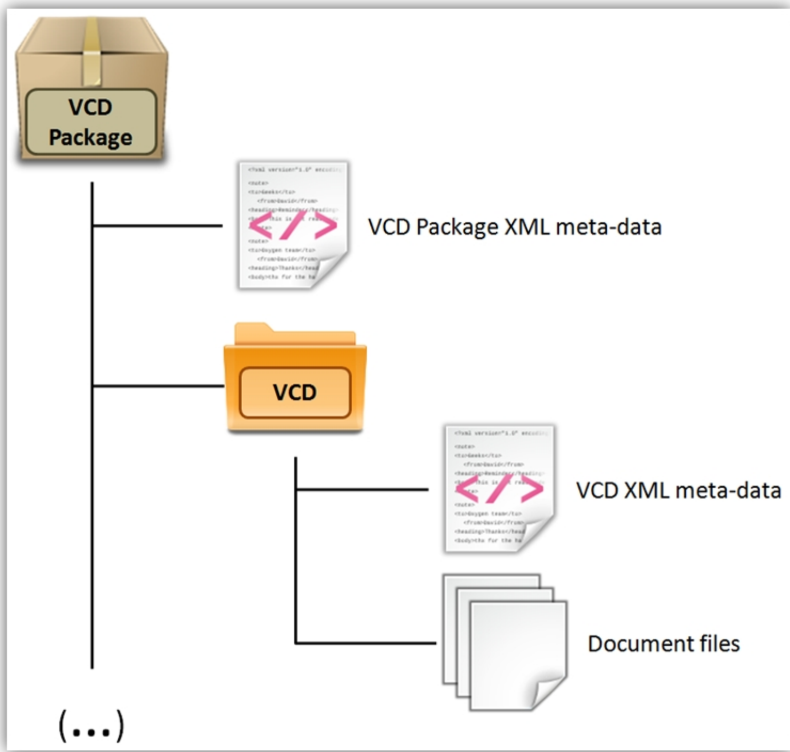

Figure 8.7: VCD Package structure

Two further document types are derived from these data formats: A *VCD Skeleton*, which is a VCD with the difference that no binary data or external document references are included. It serves as a support for economic operators to provide evidences required to prove their compliance with the qualification and selection criteria. A *VCD Package Skeleton*, which is similar to the VCD Skeleton an "empty" VCD Package, contains only VCD Skeletons indicating the required evidences, but no binary document data.

Chapter 9

IMPLEMENTATION AND DEPLOYMENT

After the description of the analysis and design, this chapter will now focus on the implementation and deployment phase of the development process. The following sections will provide an overview about the technical system architecture and infrastructure as well as the implementation of the prototype's components. Similar to the previous chapter, the information in this chapter are intended to provide a complementary view to the specification document on the tasks and results of the service assembling phase. All technical details are described in the specification document on the enclosed DVD.

9.1 System Architecture

The components identified in the previous phases comprise the system architecture as depicted in figure 9.1.

Core components in this architecture are the web services, that can be accessed through their interfaces via the network (i.e. the Internet). These services can either be accessed by applications, that provide the user interfaces as well as additional business logic, or by processes that orchestrate existing web services. Via the network, it is also possible for web services to access other web services. The User Management and Document Manager web services provide the interfaces to persist and retrieve data in the system's data store.

With the Internet as the technical communication layer in this architecture, it is possible to flexibly re-arrange or add further web services, applications and orchestration processes. Because of the prototyping approach applied for this architecture, it only provides a low SOA maturity that renounces the installation of a central integration component (i.e. an Enterprise Service Bus), indicating that the components communicate with each other peer-to-peer and not via a central service bus.

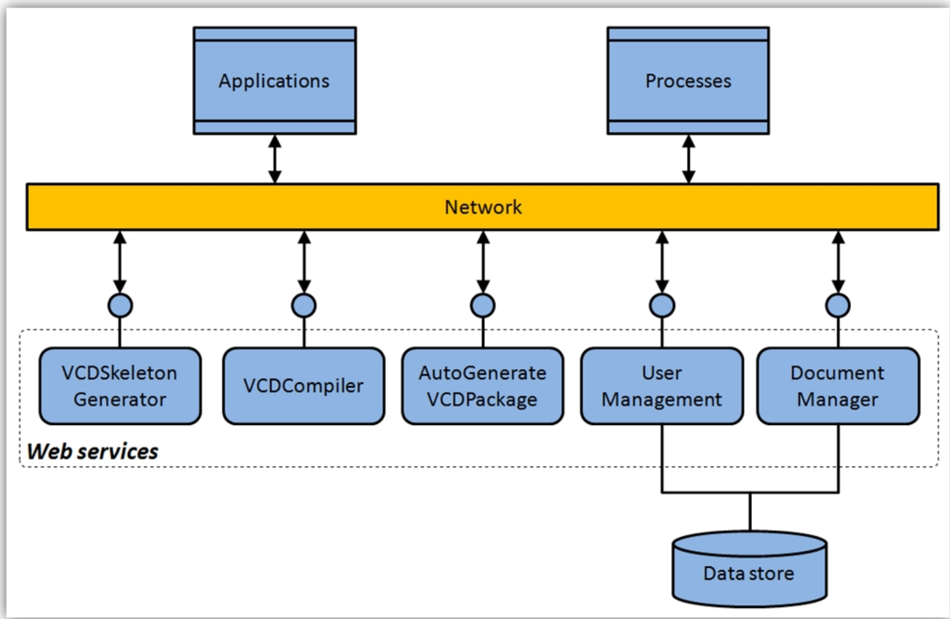

Figure 9.1: System architecture

An overview of the technical infrastructure for this prototype from the deployment viewpoint is illustrated in figure 9.2.

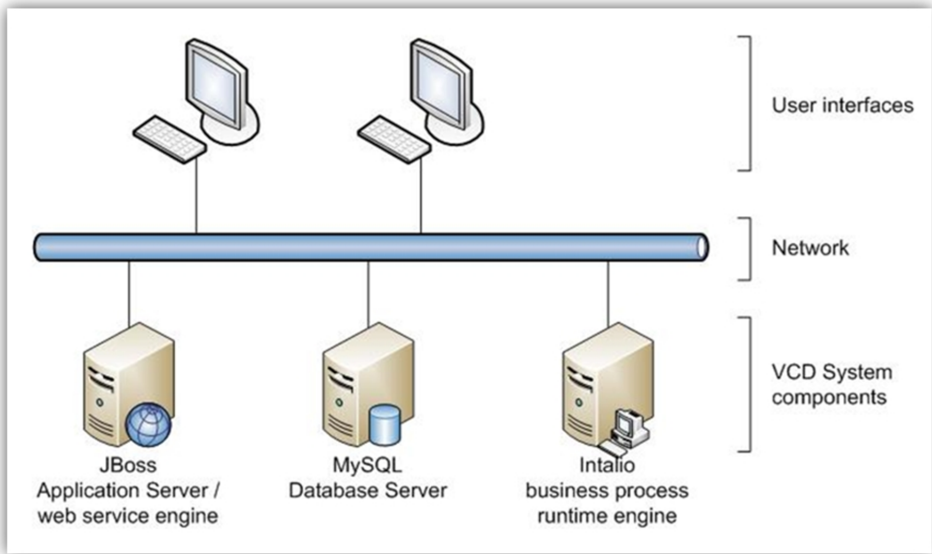

Figure 9.2: Technical system infrastructure

The components of this system architecture are deployed on the corresponding technical infrastructure components as identified in chapter 7. In particular, web applications are deployed on the *JBoss Application Server*, web services on the *JBoss web service engine* integrated in the application server. A *MySQL Database Server* hosts the database used for the system's data store, and executable business processes are deployed on the *Intalio business process runtime engine*.

Implementation issues concerning the web applications and web services of the prototype will be discussed in the following two sections.

9.2 Web Services

This section will give a brief overview of the implemented web services by describing their general purpose and operations. Further details about functional requirements, service contracts, interfaces and message types can be found in the specification document on the enclosed DVD.

VCDSkeletonGeneratorWS

The VCD Skeleton Generator Web service is a service that receives information about qualification and selection criteria (from a Call for Tender document) as well as a tenderer structure (manually entered via a user interface) and generates a VCD Package Skeleton (see section 8.3). The generated VCD Package Skeleton will be returned to the requester and is (optionally) stored in the VCD system data store. The ServiceContract of the VCDSkeleton-GeneratorWS specifies the interaction between service consumers and the service provider through the ServiceInterface. Possible consumers are the VCDCompilerWS (see next section), Web applications or other system components. The provider is the VCD Skeleton Generator web service. The purpose of the provided operations is described in table 9.1.

Operation	Description
GenerateVCDSkeleton	Receives a request to generate a VCD Package Skeleton based on the data specified by the request parameters and returns it to the requester.
AutoGenerateVCD Skeleton	Receives an object id of a GenerateVCDSkeletonRequest object stored in the system data store to generate a VCD Package Skeleton. Instead of returning the created VCD Package Skeleton, this method stores it in the VCD system data store and returns the id of the persisted object.
GetEvidenceList ForCriterion	Returns a list of evidences that prove a criterion under the legislation of a specific target country. This operation simulates the criteria mapping.

Table 9.1: Operations of the VCDSkeletonGeneratorWS

VCDCompilerWS

The VCDCompilerService is a service that receives a VCDPackageSkeleton along with a set of evidences and compiles them into the Skeleton in order to return the compiled VCD to the requester. ServiceContract of the VCDCompilerWS specifies the interaction between service consumers and the service provider through the ServiceInterface. Possible consumers are web applications or other system components. The provider is the VCD Compiler web service. The provided operations are described in table 9.2.

Operation	Description
compileVCDPackage	Receives a VCD Package Skeleton and a list of required evidences. The evidences are stored in the VCD system data store, the VCD package is returned to the requester.
autoCompile VCDPackage	Receives an ID of a VCD Package Skeleton stored in the VCD system data store. It retrieves this package skeleton and collects the required evidences from the VCD System Datastore and compiles the VCD Package, which is then stored in the VCD system data store. The id of the stored object is sent back to the requester.

Table 9.2: Operations of the VCDCompilerWS

VCDDocumentManagerWS

The VCDDocumentManagerWS is a web service that stores and retrieves VCD objects (i.e. VCDs, VCD package skeletons, evidences). The service uses the system data store to store meta data for each object as well as the binary data of an object. The ServiceContract of the VCDDocumentManagerWS specifies the interaction between service consumers and the service provider through the ServiceInterface. Possible consumers are the web applications or other system components. The provider is the Document Manager web service, providing the operations described in table 9.3.

Operation	Description
storeVCDDocument	Receives a VCD Document and stores it in the VCD System data store.
getVCDDocument	Returns a VCD Document, if this exists in the data store.
getEvidence DocumentByCode	Returns an evidence document for a given evidenceType-Code.

Table 9.3: Operations of the VCDDocumentManagerWS

UserManagementWS

The UserManagementWS is a web service responsible for the registration and authentication of users within the VCD System. Its main purpose is to check the login credentials of users trying to access the web application user interfaces. Additional functionality includes the registration of users and organizations. The ServiceContract of the UserManagementWS specifies the interaction between service consumers and the service provider through the ServiceInterface. Possible consumers are the web applications. The provider is the user management web service. The purpose of the provided operations is described in table 9.4.

Operation	Description
isAuthenticated	Checks if a user with a given username and password is registered in the system.

Table 9.4: Operations of the UserManagementWS

AutoCreateVCDPackageWS

The AutoCreateVCDPackageWS invokes a process service that orchestrates the above specified services into a single one. It implements the use case *Automatic creation of a VCD package*. The orchestration is realized with the Intalio BPMS as depicted in figure 9.3. The process service is deployed on the Intalio Process Server as a WS-BPEL process. The process can be invoked through the interface that is represented by the pool "AutoCreateVCDPackage-Client". The orchestrated process itself is represented by the middle pool "AutoCreateVCD-PackageWS". The pool "VCDSystem" contains the operations of the web services that are invoked within the defined process. The icons attached to the message flows refer to the data elements that are exchanged between the tasks. This view on the implementation details is usually not visible to service consumers, who only see the service interface represented by the top most pool. This interface is used by the AutoCreateVCDPackageWS to invoke the process service. The corresponding ServiceContract specifies the provided operation which are described in table 9.5.

Operation	Description
createVCDPackage	Generates a VCD Package based on the given GenerateVCDSkeletonRequest document specified by its ID.

Table 9.5: Operations of the AutoCreateVCDPackageWS

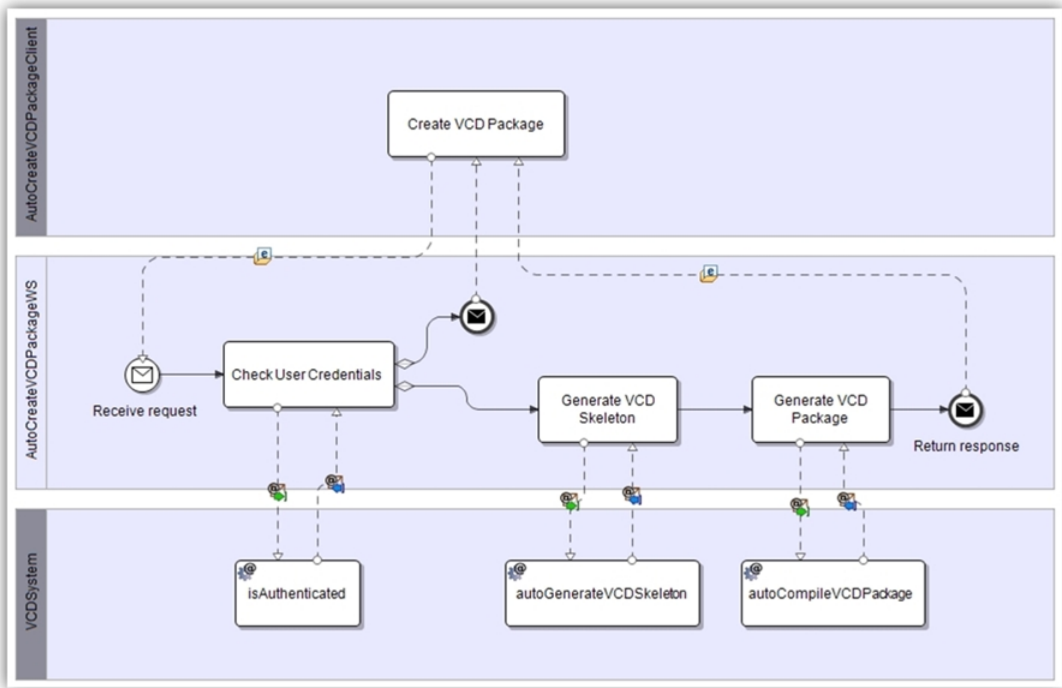

Figure 9.3: AutoCreateVCDPackage process service

9.3 Web Applications

The web applications provide graphical user interfaces allowing users to interact with the system for the use cases described in chapter 8 on page 79. Besides application specific functionality, they invoke operations of the previously described web services and provide GUI elements to allow users specify the required data for the web service's input parameters.

The **VCD Package Skeleton Generator Application** supports economic operators in creating a VCD Package Skeleton. It invokes the VCDSkeletonGeneratorWS and provides required GUI elements to specify the qualification and selection criteria (through an upload of a Call for Tender document) as well as the tenderer structure. It furthermore allows a user to invoke the AutoCreateVCDPackageWS, that automatically generates the VCD Package Skeleton, collects required evidence documents from the system data store and generates the VCD Package.

The **VCD Package Compiler Application** supports economic operators in compiling a VCD Package. It visualizes the list of required evidences extracted from an uploaded VCD Package Skeleton and provides GUI elements allowing the user to upload the corresponding evidence documents. Once all documents are uploaded, the application compiles the data into a VCD Package and provides it to the user.

The **VCD Viewer Application** enables different users (such as economic operators and contracting authorities) view the content of VCD Packages and VCD Package Skeletons. The focus of its implementation within this prototype is to provide a first version as a web application that will be the foundation for the later implementation (see [Mondorf et al., 2010] for a detailed specification). In order to specify the data presentation and user interaction elements, the first step was to design the graphical user interface of the application. The GUI design developed within WP2 is described in detail in the specification document on the enclosed DVD. The implementation of the VCD Viewer is based on this user interface design including the data presentation and user interaction elements, but is adjusted to the data model specification of this prototype and the defined scenario. For the ongoing implementation within WP2, the functionality has to be extended to comply with the full VCD data format specification as defined in [Mondorf et al., 2010; Mondorf and Wimmer, 2009].

Chapter 10

REVIEW OF PRACTICAL WORK

This chapter will now provide a review of the conducted work by first summarizing the main issues and outcomes. After a critical assessment of the prototype implementation, an outlook on further development and improvements will be provided. This also includes a comparison and transfer of the results of this work to other related work.

10.1 Summary and Critical Assessment

The target of the practical part of this master thesis was to apply aspects elaborated within part one of this master thesis, in particular the service-oriented approach described in chapter 4. Based on implementation objectives derived from the theoretical foundation, the project work started with narrowing the overall VCD problem domain and the definition of an exemplary use case scenario for the purpose of a later demonstration of the prototype, its system behavior and functionality. The tasks within the analysis and design phase consisted of modeling the business processes for each VCD maturity stage. The general process model for the scoped sub-domain has been derived from the process model of stage one. Further detailing this domain through the modeling of the overall service architecture and use cases finally allowed the identification and specification of relevant software components. These provide system functionality on the one hand which is implemented as web services. Graphical user interfaces on the other hand provide means for users to interact with the system. A detailed specification of the VCD sub system domain and software components has been elaborated to define the use cases in scope, system components as well as software requirements. This has been the basis for the implementation of the specified software components, which represent technical implementations of the specified use cases. The technical infrastructure components evaluated and selected (i.e. JBoss Application Server, JBoss Web Service Engine, Intalio business process runtime engine, MySQL database) comprise the IT architecture.

All in all, the implementation objectives described in section 6.2 on page 67 have generally been achieved successfully: The sub-domain has been identified for which services and software components have been specified and implemented following a service-oriented approach. The SOA paradigm has been realized by aligning the identified services to the business processes of the target domain, by implementing reusable capabilities as web services and by providing an infrastructure that supports technical interoperability, re-usability and loose coupling while being able to flexibly extend and adapt it to changing conditions. However, the developed service-oriented architecture has a low maturity concerning the SOA maturity levels introduced in section 4.1.4 on page 41. The primary reason for that is the limited size of the target domain that only provides a small set of use cases, services and system components. Hence, the implemented technical system architecture only features basic business process support and does not contain an Enterprise Service Bus as the core integration component.

As the software components are not intended to provide full business logic as specified in the WP2 deliverables, the software functionality only covers a minimal set of requirements sufficient enough to demonstrate the functionality for the prototype's use case scenario. However, the implemented web applications demonstrate a possible solution for providing the user interaction elements within this IT architecture while re-using existing functionality provided by web services.

10.2 Further Work and Development

The most obvious need for improvements concerns the functionality of the implemented software components. Both web applications and web services have been implemented to provide user interaction elements and data processing functionality suitable for the use case scenario. One example for improvements is the mapping component within the VCD Skeleton Generator web service: The current version only provides a simple mapping of criteria and evidences based on a tree data structure. In order to realize the WP2 specification, this functionality needs to be replaced by an ontology reasoning component.[1] Another improvement in this context is the support for the CEN/ISSS specifications on Business Interoperability Interfaces so that software components implement common data formats for the exchange of documents within public procurement processes. The Call for Tender document defined for this prototype is a basic example for a BII based data format. Further modeling and orchestration of the procurement processes of the BII specification with the installed business process management solution would provide additional business value.

Other important and necessary improvements relate to security aspects: Most important in this context is the implementation of a solution that enables the secure exchange of messages

[1] A prototype for this component has been developed in another diploma thesis, see [Müller, 2009].

as well as user authentication and access restrictions. The message exchange between web services and consumers for example is not secure in terms of integrity and encryption. Hence, the implementation of security standards for web services such as the WS-Security standard is required. User authentication requires improvements of the prototype's user management component, that currently only supports basic password authentication for accessing the developed web applications. This also includes means to define and manage access rights which have to guarantee that a user can only read or edit those information for which he has corresponding access rights. This is especially relevant for the highest maturity stages of the VCD maturity model which relies on the electronic on-demand collection of documents that are stored in data stores available within the VCD system.

Further extensions concern the IT infrastructure: In order to support the management and discoverability of deployed web services, a service registry has to be installed within the system's architecture. As the JBoss Application Server in combination with the JBoss web service engine already contains such a service registry, this may be specifically used within this system. In order to install a central integration and messaging component, an Enterprise Service Bus also needs to be installed that allows the communication between connected information systems in larger IT environments. The integration of the developed web applications into a portal solution may furthermore provide an aggregated view on information and data within the system and a single point of access via a fully featured graphical user interface.

A resulting extended architectural model of the prototype's system architecture (see figure 9.1 on page 89) is illustrated in figure 10.1. This system architecture provides the technical means to support higher VCD maturity stages as well as higher SOA maturity levels and allows the integration of different components that communicate via the central enterprise service bus.

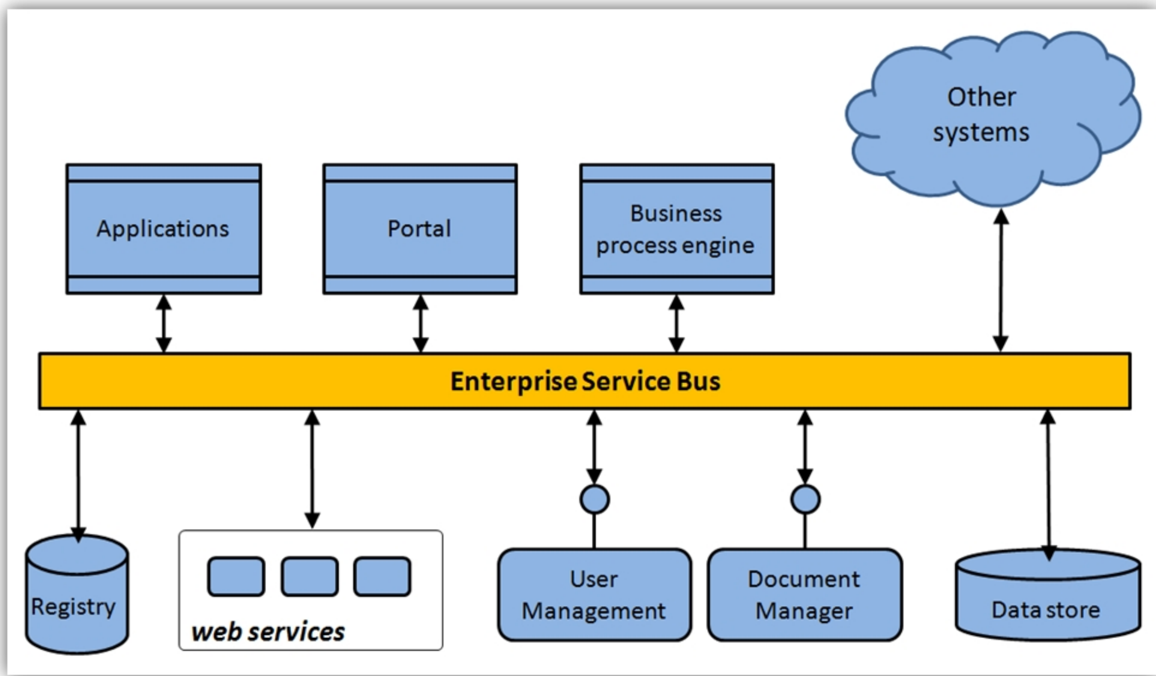

Figure 10.1: Extended system architecture

10.3 Related Work

Related to the work of this master thesis is work package 8 of the PEPPOL project that addresses the need for a technical infrastructure that enables the electronic exchange of documents and the inter-connection of distributed system components. Important aspects of the project results as well as relations to the architecture developed for this master thesis will be briefly discussed in this section.

The **PEPPOL e-business document transport infrastructure (PEDRI)** developed by WP8 is intended to provide the infrastructure for a secure and reliable exchange of electronic documents within the PEPPOL domain. It is furthermore designed to provide means that allow private companies and public sector institutions to access this infrastructure via existing transport infrastructures, such as national value added networks or E-Procurement systems, or directly via the Internet. This allows the technical infrastructure to be built upon the existing national solutions, which is one aim of the overall PEPPOL project (cf. section 5.1). Furthermore, PEDRI will implement the standards and tools compliant to the CEN/ISSS BII.[2] Figure 10.2 provides an overview of the overall PEDRI architecture[3].

[2] See http://docs.google.com/Doc?id=dg8mxnjx_0dck65kgx and for detailed information.

[3] Graphic taken from http://www.peppolinfrastructure.com/20080916FrameworkArchitecture2.0.ppt (19.06.2010)

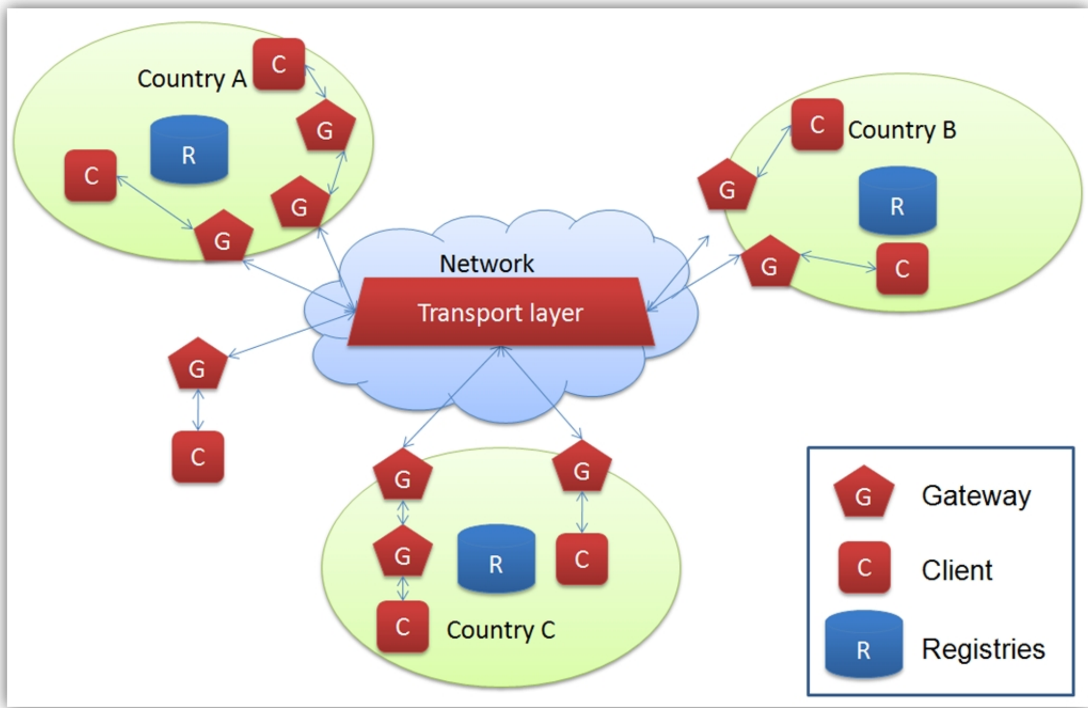

Figure 10.2: Overview of the PEDRI architecture

The core elements within this architecture are distributed national **e-business registries** that store information about the capabilities for exchanging business documents and connected actors from different countries. These information are used for example for the address lookup of other actors required to deliver electronic documents. **Gateways** will achieve inter-connections between actors, shared registries and existing infrastructures. Message oriented middleware will be used for these gateways, providing adapters to PEDRI specific services as well as integrating national infrastructures. Security services provided by the PEDRI infrastructure are for example encryption of documents with *digital certificates* and *certificate validation*. A fundamental challenge within the PEDRI infrastructure is discoverability of registries and recipient endpoints. For the decentralized approach of PEDRI, several types of registries will be used: A European top-level registry (registry of national registries) that contains information of all national registries within the infrastructure. These national registries in turn contain information about registered national e-business registries, which finally contain information about the endpoints of registered actors such as private companies and public procurement institutions.

Taking the extended architectural model from the previous section into consideration, the IT infrastructure implemented for this master thesis provides suitable technologies and technical components to realize the PEDRI architecture through applying the SOA paradigm. In combination with the WP2 specification of the VCD system, an architectural model sketch may consist of the technical components and services as illustrated in figure 10.3.

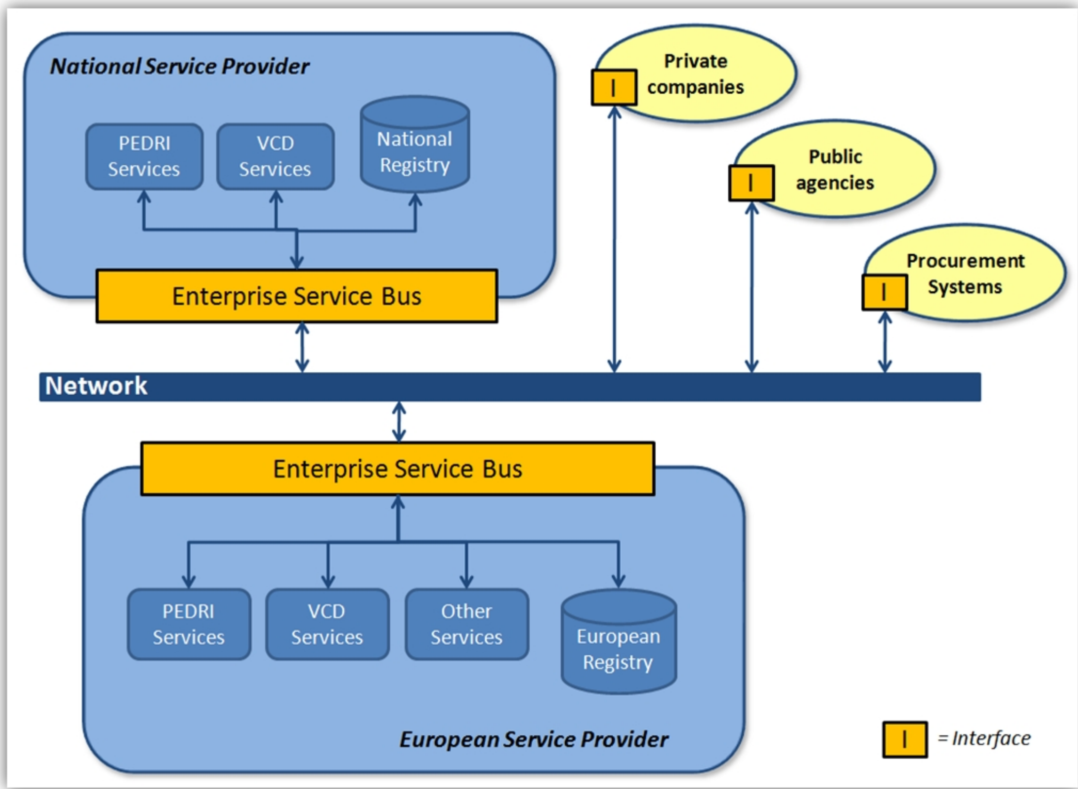

Figure 10.3: Architectural sketch of a SOA for PEDRI and VCD

This architecture foresees the implementation of a European Service Provider that hosts for example the European registry as defined by the PEDRI architecture as well as European VCD services. An ESB provides the integration component and gateway through which the different national systems can be integrated. The national systems implement a National Service Provider that hosts corresponding PEDRI and VCD services with a national scope as well as the national registries. Private companies, public agencies as well as existing procurement systems can be integrated through the ESBs provided by the National Service Providers. The communication between the components and actors is conducted over the Internet by using interfaces, for example PEDRI communication interfaces.

Chapter 11

CONCLUSIONS

Interoperability is one of the most important requirements for European public E-Procurement and the exchange of documents between procuring entities and suppliers. This is caused by a large heterogeneity concerning the technical, organizational and legal systems of different member states. Interoperability is a characteristic of technical systems as well as organizations to be able to exchange and process data electronically. To achieve this, several issues concerning technical, semantic, organizational, legal and political dimensions have to be addressed. Several standardization efforts are being conducted, each targeting interoperability issues of specific dimensions, such as the UBL for the technical and semantic level or the CEN/ISSS BII for the semantic and organizational level. In the European Union, several frameworks furthermore target the legal and political harmonization across member states, such as the Public Sector Directive or the EIF. These standards and frameworks provide important inputs for the system design and development. The PEPPOL project aims at designing and developing an interoperable European public E-Procurement system, hence the work heavily relies on and uses the aforementioned standards. Work package 2 in particular develops data formats based on UBL and the BII specifications. Furthermore, the VCD system requires a technical architecture that supports the electronic exchange of documents as well as interconnections of heterogeneous and distributed information systems.

The SOA paradigm provides a suitable approach to realize such system architectures, in that system components can exist autonomously, are loosely coupled and able to exchange and process information used within business processes across organizational boundaries. The approach, methodologies and technologies behind that paradigm represent a current evolutionary stage of the development of distributed system architectures that promotes the use of existing technologies, standards and methodologies following certain service-orientation principles. These principles facilitate interoperability and loose coupling of autonomous and heterogeneous system components that are technical representations of the SOA services and can be flexibly re-used within business processes. Hence, this approach in combination with the aforementioned interoperability standards provides necessary means for the development of an interoperable technical and organizational environment for European public E-Procurement.

A possible service-oriented development process follows a top-down approach, starting with an analysis and design of the business domain and relevant business processes. These are broken down into sub-components and process tasks that represent the services provided within the domain. Several standards exist such as UML and SoaML for domain and service modeling as well as BPMN for business process modeling. In order to technically realize these services so that they can be re-used within business processes or by other technical components, a suitable implementation technology is the web service platform, as it provides standardized means for discoverability, re-usability, loose coupling and technical interoperability. Accordingly, business processes comprising business services are the organizational components of an SOA, while web services are the technical implementations of SOA services. For the technical infrastructure, further components are required. On the one hand, this includes graphical user interfaces that allow users to interact with system components and services. On the other hand, an Enterprise Service Bus (ESB) provides the core communication and integration component, allowing message exchanges between the system components that are connected via the ESB.

However, a service-oriented architecture cannot be built from scratch. Major challenges arise from the need for achieving interoperability on semantic and organizational levels as the SOA approach merely focuses on the technical interoperability dimensions. Heterogeneity is one great obstacle especially in large scale distributed environments with technical, organizational and legal differences. In combination with existing semantic and organizational interoperability standards and frameworks, SOA provides an approach how these standards and technologies shall be applied in order to achieve interoperable and flexible IT architectures.

Part II of this master thesis identified a set of suitable standards, technologies and tools and applied the developed service-oriented top-down approach to implement a prototype covering a sub-system of the overall VCD domain. The results of this practical work provide a relevant basis for the development of the VCD system infrastructure and its components, and furthermore may be used as an input for the development of the PEDRI infrastructure specified by work package 8.

BIBLIOGRAPHY

Anghelakis, M., Lizardos, E., Valassopoulos, G., Plagiannakou, K., Vrazopoulos, H., Germanou, K., and Tsopelas, N. (2007). *Electronic transmission of public procurement notices for publication. Volume I.* URL: http://ec.europa.eu/internal_market/publicprocurement/docs/eprocurement/feasibility/enot-vol-1_en.pdf, (06.03.2010).

Arrowsmith, S. (2009). EC Regime on Public Procurement. In Thai, K. V., editor, *International Handbook of Public Procurement*, chapter 12, pages 251–289. CRC Press Taylor and Francis Group, Boca Raton, London, New York.

Baily, P., Farmer, D., Crocker, B., and Jessop, D. (2008). *Procurement Principles and Management*. Pearson Education, Harlow.

Barry, D. K. (2003). *Web services and service-oriented architecture: the savvy manager's guide*. Morgan Kaufmann, San Francisco.

Bean, J. (2010). *SOA and Web Service Interface Design: Principles, Techniques, and Standards*. Morgan Kaufmann Publishers, Burlington, MA.

Beinhauer, W., Herr, M., and Schmidt, A. (2007). *SOA für agile Unternehmen: Serviceorientierte Architekturen verstehen, einführen und nutzen*. Symposium Publishing, Düsseldorf.

Bieberstein, N., Laird, R. G., Jones, K., and Mitra, T. (2006). *Planning and Implementing SOA, Section 1 Management Summary*. Butler Group. URL: http://www.butlergroup.com/research/reporthomepages/soa.asp, (19.05.2010).

Bieberstein, N., Laird, R. G., Jones, K., and Mitra, T. (2008). *Executing SOA: A Practical Guide for the Service-Oriented Architect*. IBM Press.

BMI (2008). *Standards und Architekturen für E-Government-Anwendungen*. Bundesministerium des Innern.

Bovis, C. (2005). *Public Procurement in the European Union*. Palgrave Macmillan Ltd., Hampshire.

Bovis, C. (2007). *EU Public Procurement Law*. Edward Elgar Publishing Limited, Cheltenham.

Brown, P. C. (2008). *Implementing SOA: Total Architecture in Practice*. Pearson Education, Inc.

BSI (2004). Sichere Integration von E-Government-Anwendungen. In *E-Government-Handbuch*. Bundesamt für Sicherheit in der Informationstechnik (BSI).

Casanave, C. (2009). *Enterprise Service Oriented Architecture: Using the OMG SoaML Standard*.

CEN/ISSS WS/BII (2009a). *BII Profile Architecture*.

CEN/ISSS WS/BII (2009b). *Business Interoperability Interfaces for Public procurement in Europe – Part 0: Introduction*.

CEN/ISSS WS/BII (2009c). *Business Interoperability Interfaces for Public procurement in Europe – Part 1: Pofile overview*.

Cerami, E. (2003). *Web Services Essentials*. O'Reilly.

Chaffey, D. (2002). *E-Business and E-Commerce Management*. Prentice Hall, London, New York, et al.

Chang, J. F. (2006). *Business Process Management Systems: Strategy and Implementation*. Auerbach Publications.

Chappell, D. (2004). *Enterprise Service, Bus*. O'Reilly, Upper Saddle River, NJ.

Ciganek, A. P., Haines, M. N., Haseman, W., and Ngo-Ye, L. (2007). Using a Standards-Based Integration Platform for Improving B2B Transactions. In Lam, W. and Shankararaman, V., editors, *Enterprise Architecture and Integration : Methods, Implementation, and Technologies*, chapter 6, pages 92–106. Information Science Reference, Hershey, New York.

Commission, E. (2003). *Linking up Europe: the Importance of Interoperability for eGovernment Services*. European Communities.

Cummins, F. A. (2009). *Building The Agile Enterprise: With SOA, BPM and MBM*. Elsevier Inc., Burlington.

Declercq, K., Vincent, A., De Backer, T., and Loterie, H. (2009). *European Interoperable Infrastructure Services: Study on potential reuse of system components*. Report prepared for the IDABC Programme.

Deeg, M. (2009). Modellierung von Services mit SoaML. In *Objektspektrum Online Ausgabe: Service-orientierte Architekturen*. SIGS Datacom. URL: http://www.sigs-datacom.de/fileadmin/user_upload/zeitschriften/os/2009/SOA/deeg_OS_SOA_09.pdf, (23.05.2010).

Directive 2004/17/EC (2004). *Directive 2004/17/EC ofthe European Parliament and of the Council of the European Union coordinating the procurement procedures of entities operating in the water, energy, transport and postal services sectors*.

Directive 2004/18/EC (2004). *Directive 2004/18/EC of the European Parliament and of the Council of the European Union on the coordination of procedures for the award of public works contracts, public supply contracts and public service contracts*.

Dunkel, J., Eberhart, A., Fischer, S., Kleiner, C., and Koschel, A. (2008). *Systemarchitekturen für Verteilte Anwendungen*. Carl Hanser Verlag, München.

Endrei, M., Ang, J., Arsanjani, A., Chua, S., Comte, P., Krogdahl, P., Luo, M., and Newling, T. (2004). *Patterns: Service-Oriented Architecture and Web Services*.

Erl, T. (2005). *Service-Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall, Upper Saddle River, NJ.

Erl, T. (2008). *SOA: Principles of Service Design*. Prentice Hall, Upper Saddle River, NJ.

European Commission (2004). *Action plan for the implementation of the legal framework for electronic public procurement*. URL: http://ec.europa.eu/internal_market/publicprocurement/docs/eprocurement/actionplan/actionplan_en.pdf, (04.03.2010).

European Commission (2005). *Requirements for conducting public procurement using electronic means under the new public procurement Directives 2004/18/EC and 2004/17/EC*. URL: http://ec.europa.eu/internal_market/publicprocurement/docs/eprocurement/sec2005-959_en.pdf, (04.03.2010).

European Commission (2008). *European eProcurement: an overview*.

European Commission (2009a). *Electronic Public Procurement*. URL: http://ec.europa.eu/internal_market/publicprocurement/e-procurement_en.htm, (10.03.2010).

European Commission (2009b). *European eProcurement Map*. URL: http://ec.europa.eu/idabc/servlets/Doc?id=31968, (10.03.2010).

European Commission (2010a). *eGovernment in Europe*. URL: http://ec.europa.eu/information_society/activities/egovernment/index_en.htm, (10.03.2010).

European Commission (2010b). *Summaries of EU legislation: Businesses in the internal market*. URL: http://europa.eu/legislation_summaries/internal_market/businesses/index_en.htm, (17.02.2010).

European Communities (2006). *i2010 eGovernment Action Plan: Accelerating eGovernment in Europe for the Benefit of All*. URL: http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=COM:2006:0173:FIN:EN:PDF, (10.03.2010).

European Dynamics S.A. (2004). *Case Studies on European Electronic Public Procurement Projects, State of the Art Report, Volume I*. URL: http://ec.europa.eu/idabc/servlets/Doc?id=22175, (09.03.2010).

European Dynamics S.A. (2005). *Functional requirements for conducting electronic procurement under the EU framework - Volume I*. URL: http://ec.europa.eu/internal_market/publicprocurement/docs/eprocurement/functional-reguirements-vol1_en.pdf, (26.02.2010).

European Dynamics S.A. (2007a). *eCatalogues in electronic public procurement, Volume I, Report on the state of play*. URL: http://ec.europa.eu/internal_market/publicprocurement/docs/eprocurement/feasibility/ecat-vol-1_en.pdf, (06.03.2010).

European Dynamics S.A. (2007b). *eCatalogues in electronic public procurement, Volume II, Standardisation Initiatives*. URL: http://ec.europa.eu/internal_market/publicprocurement/docs/eprocurement/feasibility/ecat-vol-2_en.pdf, (06.03.2010).

European Union (2008). *The Treaty on the Functioning of the European Union*. Official Journal of the European Union, URL: http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:C:2008:115:0047:0199:EN:PDF, (17.02.2010).

Frenz, W. (2007). *Handbuch Europarecht: Beihilfe- und Vergaberecht*. Springer-Verlag, Berlin, Heidelberg, New York.

Gottschalk, P. and Solli-Sæther, H. (2009). *E-Government Interoperability and Information Resource Integration: Frameworks for Aligned Development*. Information Science Reference, Hershey, New York.

Grün, O., Jammernegg, W., and Kummer, S. (2009). *Grundzüge der Beschaffung, Produktion und Logistik*. Pearson Studium, München.

Grässle, P., Baumann, H., and Baumann, P. (2005). *UML 2.0 in Action*. Packt Publishing, Birmingham, Mumbai.

Harold, E. and Means, W. S. (2007). *XML in a Nutshell*. O'Reilly Media.

Havey, M. (2007). *Essential Business Process Modeling*. O'Reilly Media.

Hoernes, P. (2007). Business Service Modellierung. In *Objektspektrum Online Ausgabe: Serviceorientierte Architektur*, pages 36–39. SIGS Datacom. URL:

http://www.sigs-datacom.de/fachzeitschriften/objektspektrum/online-themenspecial/artikelansicht.html?show=2154, (23.05.2010).

Hofmeister, C., Nord, R., and Soni, D. (2000). *Applied Software Architecture*. Addison Wesley, Reading.

IBM (2010). *Service Oriented Architecture - Open for business: The Open Group Service Integration Maturity Model (OSIMM)*. IBM. URL: http://www-01.ibm.com/software/solutions/soa/newsletter/apr07/article_OSIMM.html, (21.05.2010).

IDABC (2004). *European Interoperability Framework for Pan-European eGovernment Services, Version 1*. European Communities.

IDABC (2008). *European Interoperability Framework for Pan-European eGovernment Services, Version 2 Draft*. European Communities.

Juric, M. B. (2006). *Business Process Execution Language for Web Services*. Packt Publishing.

Juric, M. B., Loganathan, R., Sarang, P., and Jennings, F. (2007). *SOA Approach to Integration*. Packt Publishing, Birmingham.

Kalfoglou, Y. (2010). *Cases on Semantic Interoperability for Information Systems Integration: Practices and Applications*. Information Science Reference, Hershey, New York.

Klischewski, R. (2003). Top down or bottom up? how to establish a common ground for semantic interoperability within e-government communities. In Traunmüller, R. and Palmirani, M., editors, *E-Government: Modelling Norms and Concepts as Key Issues. Proceedings of 1st International Workshop on E-Government at ICAIL 2003*, pages 17–26.

Klischewski, R. (2004). Information integration or process integration? how to achieve interoperability in administration. In Traunmüller, R., editor, *Proceedings of EGOV 2004*, pages 57–65, Berlin. Springer Verlag.

Kosanke, K. (2006). ISO Standards for Interoperability: A Comparison. In Konstantas, D., Bourrières, J.-P., Léonard, M., and Boudjlida, N., editors, *Interoperability of Enterprise Software and Applications*, pages 55–64, London. Springer-Verlag.

Krafzig, D., Banke, K., and Slama, D. (2005). *Enterprise SOA: Service-Oriented Architecture Best Practices*. Prentice Hall.

Lam, W. and Shankararaman, V. (2007). Dissolving Organisational and Technological Silos: An Overview of Enterprise Integration Concepts. In Lam, W. and Shankararaman, V., editors, *Enterprise Architecture and Integration : Methods, Implementation, and Technologies*, chapter 1, pages 1–22. Information Science Reference, Hershey, New York.

Laskaridis, G., Markellou, P., Panayiotaki, A., and Tsakalidis, A. (2008). Interoperability on the road to enhance government-to-business. In Anttiroiko, A.-V., editor, *Electronic Government: Concepts, Methodologies, Tools, and Applications*, Hershey, New York. InformatIon Science Reference.

Lassmann, W. (2006). *Wirtschaftsinformatik: Nachschlagwerk für Studium und Praxis*. Gabler Verlag, Wiesbaden.

Lawler, J. and Howell-Barber, H. (2008). *Service-oriented architecture : SOA Strategy, Methodology, and Technology*. Auerbach Publications, Boca Raton.

Leinemann, R. (2010). *Das neue Vergaberecht*. Werner Verlag, Köln.

Liebhart, D. (2007). *SOA goeas reals*. Carl Hanser Verlag, München, Wien.

Linthicum, D. (2001). *B2B Application Integration*. Addison Wesley, Boston.

Linthicum, D. (2003). *Next Generation Application Integration: From Simple Information to Web Services*. Addison Wesley, Boston.

Margolis, B. and Sharpe, J. (2007). *SOA for the Business Developer: Concepts, BPEL, and SCA*. MC Press Online, Lewisville.

McGovern, J., Sims, O., Jain, A., and Little, M. (2006). *Enterprise Service Oriented Architectures: Concepts, Challenges, Recommendations*. Springer, Dordrecht.

Meier, A. and Stormer, H. (2009). *eBusiness and eCommerce - Managing the Digital Value Chain*. Springer-Verlag, Berlin, Heidelberg.

Mertins, K., Ruggaber, R., Popplewell, K., and Xu, X. (2008). *Enterprise Interoperability III*. Springer-Verlag.

Minoli, D. (2008). *Enterprise Architecture A to Z: Frameworks, Business Process Modeling, SOA, and Infrastructure Technology*. Taylor & Francis Group, Boca Raton.

Müller, M. (2009). *Conceptualizing and Prototyping a Pre-VCD Mapping Tool for Pan-European Public Procurement Online*. Universität Koblenz-Landau, Koblenz.

Mondorf, A., Schmidt, D., Wimmer, M., Groiß, W., and Ipsmiller, D. (2010). *PEPPOL Deliverable D2.2 Specification of architecture and components enabling crossborder VCD, Version 0.8*. Universität Koblenz-Landau, Koblenz.

Mondorf, A. and Wimmer, M. (2008). Interoperability in e-tendering: The case of the virtual company dossier. In *ICEGOV 2008: Proceedings of the 2nd International Conference on Theory and Practice of Electronic Governance*, pages 110–116, Cairo, Egypt. ACM International Conference Proceedings Series.

Mondorf, A. and Wimmer, M. (2009). *PEPPOL Deliverable D2.1 Functional and non-functional requirements specification for the VCD, including critical synthesis, comparison and assessment of national vs. pan-European needs, Version 1.0*. Universität Koblenz-Landau, Koblenz.

Newcomer, E. and Lomow, G. (2004). *Understanding SOA with Web Services*. Addison Wesley Professional.

OASIS (2006). *Reference Model for Service Oriented Architecture 1.0: OASIS Standard*. URL: http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf, (06.11.2010).

OASIS (2006). *Universal Business Language, Version 2.0*. URL: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ubl, (29.04.2010).

OASIS (2007). *Web Services Business Process Execution Language Version 2.0*. OASIS Standard.

OASIS (2010a). *Web Services – Human Task (WS-HumanTask) Specification Version 1.1*. Committee Draft 09 / Public Review Draft 03.

OASIS (2010b). *WS-BPEL Extension for People (BPEL4People) Specification Version 1.1*. Committee Draft 07 / Public Review Draft 02.

Object Management Group (2009a). *Business Process Model and Notation Version 2 Beta 1*. URL: http://www.omg.org/cgi-bin/doc?dtc/09-08-14.pdf, (04.01.2010).

Object Management Group (2009b). *Service oriented architecture Modeling Language (SoaML) - Specification for the UML Profile and Metamodel for Services (UPMS)*. OMG Adopted Specification.

Object Management Group (2009c). *Unified Modeling Language (UML)*. URL: http://www.omg.org/spec/UML/2.2/Superstructure/PDF/, (23.05.2010).

Ouyang, C., Dumas, M., van der Aalst, W. M. P., ter Hofstede, A. H. M., and Mendling, J. (2009). From Business Process Models to Process-oriented Software Systems. In *ACM Transactions on Software Engineering and Methodology*, volume 19, Issue 1, pages 1–37. ACM, New York. URL: http://is.tm.tue.nl/staff/wvdaalst/publications/p526.pdf, (04.01.2010).

Pilone, D. and Pitman, N. (2007). *UML 2.0 in a Nutshell*. O'Reilly Media, Sebastopol.

Rathfelder, C. and Groenda, H. (2008). iSOAMM: An Independent SOA Maturity Model. In Meier, R. and Terzis, S., editors, *Distributed Applications and Interoperable Systems: Proceedings of the 8th IFIP WG 6.1 International Conference*, pages 1–15, Oslo. Springer-Verlag. Lecture Notes in Computer Science.

Richter, J.-P., Haller, H., and Schrey, P. (2005). *Serviceorientierte Architektur*, volume 28, Number 5, pages 413–416. Springer Verlag, Berlin, Heidelberg.

Rosen, M., Lublinsky, B., Smith, K. T., and Balcer, M. J. (2008). *Applied SOA: Service-Oriented Architecture and Design Strategies*. Wiley Publishing, Inc., Indianapolis.

Roshen, W. (2009). *SOA-Based Enterprise Integration*. McGraw-Hill, New York.

Ruh, W. A., Maginnis, F. X., and Brown, W. J. (2001). *Enterprise Application Integration*. John Wiley & Sons, Inc., New York, Chichester, Weinheim, Brisbane, Singapore, Toronto.

Scholl, J. and Klischewski, R. (2007). E-Government Integration and Interoperability: Framing the Research Agenda. In *International Journal of Public Administration (IJPA)*, volume 30, issue 8-9, pages 889–920.

Schubert, P. (2002). E-Procurement: Elektronische Unterstützung der Beschaffungsprozesse in Unternehmen. In Schubert, P., Wölfle, R., and Dettling, W., editors, *Procurement im E-Business - Einkaufs- und Verkaufsprozesse elektronisch optimieren*, chapter 1, pages 1–28. Carl Hanser Verlag, München, Wien.

Schubert, P. (2003). E-Business Integration. In Schubert, P., Wölfle, R., and Dettling, W., editors, *E-Business Integration*, chapter 1, pages 1–21. Carl Hanser Verlag, München, Wien.

Schwinn, A. and Winter, R. (2007). Success Factors and Performance Indicators for Enterprise Application Integration. In Lam, W. and Shankararaman, V., editors, *Enterprise Architecture and Integration : Methods, Implementation, and Technologies*, chapter 2, pages 23–39. Information Science Reference, Hershey, New York.

Serain, D. (2002). *Middleware and Enterprise Application Integration*. Springer-Verlag, London.

Sherif, M. H. (2010). Standards for Business-to-Business Electronic Commerce. In Sherif, M. H., editor, *Handbook of Enterprise Integration*, chapter 23. CRC Press, München, Wien.

Siemens and Time.Lex (2008). *Preliminary study on the electronic provision of certificates and attestations usually required in public procurement procedures*. URL: http://ec.europa.eu/internal_market/publicprocurement/docs/eprocurement/ecertificates-study_en.pdf, (06.03.2010).

Sonic Software Corp., AmberPoint Inc., BearingPoint Inc., and Systinet Corp. (2005). *A new SOA Maturity Model*.

Stoll, P. P. (2007). *E-Procurement - Grundlagen, Standards und Situation am Markt*. Friedrich Vieweg & Sohn Verlag, Wiesbaden.

Sturm, J. (2007). Interoperabilität und Standards. In Zechner, A., editor, *Handbuch E-Government : Strategien, Lösungen und Wirtschaftlichkeit*, pages 123–132. Fraunhofer-IRB-Verlag, Stuttgart.

Thai, K. V. (2009). International Public Procurement: Concepts and Practices. In Thai, K. V., editor, *International Handbook of Public Procurement*, chapter 1, pages 1–24. CRC Press Taylor and Francis Group, Boca Raton, London, New York.

The Open Group (2007). *Service-oriented Architecture (SOA)*. San Francisco.

The Open Group (2009). *OSIMM - The Open Group Service Integration Maturity Model, Draft Standard*. URL: https://www.opengroup.org/projects/osimm/, (21.05.2010).

UN/CEFACT and OASIS (2001a). *Core Component Dictionary, Version 1.0.4*. URL: http://www.ebxml.org/specs/ccDICT.pdf, (29.04.2010).

UN/CEFACT and OASIS (2001b). *Core Component Overview, Version 1.0.5*. URL: http://www.ebxml.org/specs/ccOVER.pdf, (29.04.2010).

UN/CEFACT and OASIS (2001c). *ebXML Requirements Specification, Version 1.06*. URL: http://www.ebxml.org/specs/ebREQ.pdf, (29.04.2010).

UN/CEFACT and OASIS (2001d). *ebXML Technical Architecture Specification, Version 1.0.4*. URL: http://www.ebxml.org/specs/ebTA.pdf, (29.04.2010).

Vasiliev, Y. (2006). *SOA and WS-BPEL*. Packt Publishing.

W3C WSA Working Group (2004a). *Web Services Architecture*. World Wide Web Consortium (W3C). URL: http://www.w3.org/TR/ws-arch/, (01.05.2010).

W3C WSA Working Group (2004b). *Web Services Glossary*. World Wide Web Consortium (W3C). URL: http://www.w3.org/TR/ws-gloss/, (19/05/2010).

W3C WSD Working Group (2007). *Web Service Description Language (WSDL)*. World Wide Web Consortium (W3C). URL: http://www.w3.org/TR/wsdl20/, (10.05.2010).

Weske, M. (2007). *Business Process Management: Concepts, Languages, Architectures*. Springer-Verlag.

White, S. A. (2005). *Using BPMN to Model a BPEL Process*. URL: http://www.bpmn.org/Documents/Mapping_BPMN_to_BPEL_Example.pdf, (04.01.2010).

Woods, D. and Mattern, T. (2003). *Enterprise SOA: designing IT for business innovation*. O'Reilly Media Inc., Sebastopol.

Zimmermann, H.-D. (2007). Elektronischer Dokumentenaustausch zwischen Unternehmen. In Schubert, P. and Wölfle, R., editors, *Business Collaboration: Standortübergreifende Prozesse mit Business Software, Praxislösungen im Detail*, chapter 11, pages 127–134. Carl Hanser Verlag, München, Wien.