RESEARCH REPORT

VTT-R-01500-09

SOMAF Platform and Pilots Report

Authors: Renne Tergujeff
Timo Toivanen
Juha Leppänen
Jyrki Haajanen
Tuukka Laakso

Confidentiality: Public

Report's title	
SOMAF Platform and Pilots Report	
Customer, contact person, address	**Order reference**
Finnish Funding Agency for Technology and Innovation (Tekes) Keijo Silventoinen Kyllikinportti 2, P.O. Box 69, FI-00101 Helsinki	40172/07
Project name	**Project number/Short name**
Service Oriented Mobile Application Framework	20175 / SOMAF
Author(s)	**Pages**
Renne Tergujeff, Timo Toivanen, Juha Leppänen, Jyrki Haajanen, Tuukka Laakso	47
Keywords	**Report identification code**
supply chain, mobile, service-oriented architecture, SOA	VTT-R-01500-09

Summary

This report describes the work and results of the Service Oriented Mobile Application Framework (SOMAF) project in the "VAMOS – Value Added Mobile Solutions" programme of Tekes, the Finnish Funding Agency for Technology and Innovation. The project ran from June 2007 until December 2008, and it focused on service-oriented approaches for supporting flexible information exchange, in business environments that often incorporate also mobile terminals.

Companies participating in the project offered three cases, related to transport logistics and information exchange in business between companies in Finland and Russia. For each case, a requirement specification was made, a pilot solution defined, and a prototype implementation developed. The *EU Entry Summary Declaration* pilot prepared for the forthcoming new requirements for electronic procedures with European customs authorities. In the *Fleet Management* pilot, a lightweight solution for exchanging transportation status information was produced, by exploiting ordinary mobile phones. In the *Trade* pilot, a technical solution was produced for enhancing information delivery between supply chain participants in border-crossing transportations.

As defined as another main goal of the project, a software platform was designed and developed, to support the development of the above and similar business cases. The principles of service-oriented architecture (SOA) were widely applied. The SOMAF platform succeeded in providing a basis for the pilot solutions, and is able to support similar cases in the future. At the end of the report, some conclusions and recommendations for future development are presented.

Confidentiality	Public

Espoo 9 March 2009

Written by	Reviewed by	Accepted by	
Renne Tergujeff	Joni Niemi	Markus Tallgren	Jussi Paakkari
Research Scientist	Team Leader	Technology Manager	Vice President, R&D, ICT

VTT's contact address

VTT, PL 1000, 02044 VTT, Finland

Distribution (customer and VTT)

Ministry of Transport and Communications of Finland, Canon North-East Oy, UPM-Kymmene Wood Oy, UPM-Kymmene Oyj, Russian Cargo Service Oy, Finnish Funding Agency for Technology and Innovation (Tekes), VTT Technical Research Centre of Finland

Raportin nimi SOMAF Platform and Pilots Report	
Asiakkaan nimi, yhteyshenkilö ja yhteystiedot Tekes - teknologian ja innovaatioiden kehittämiskeskus Keijo Silventoinen Kyllikinportti 2, PL 69, 00101 Helsinki	Asiakkaan viite 40172/07
Projektin nimi Service Oriented Mobile Application Framework	Projektin numero/lyhytnimi 20175 / SOMAF
Raportin laatija(t) Renne Tergujeff, Timo Toivanen, Juha Leppänen, Jyrki Haajanen, Tuukka Laakso	Sivujen lukumäärä 47
Avainsanat toimitusketju, mobiili, palvelukeskeinen arkkitehtuuri, SOA	Raportin numero VTT-R-01500-09

Tiivistelmä

Tämä raportti kuvaa Tekesin ”VAMOS – Liiketoiminnan mobiilit ratkaisut” -ohjelmaan kuuluneen Service Oriented Mobile Application Framework (SOMAF) -hankkeen työn ja tulokset. Projekti kesti kesäkuusta 2007 joulukuuhun 2008, ja se keskittyi palvelukeskeisiin lähestymistapoihin joustavan tiedonvälityksen tukemisessa, liiketoimintaympäristöissä, joissa usein hyödynnetään myös mobiileja päätelaitteita.

Projektiin osallistuneet yritykset tarjosivat kolme case-kohdetta, jotka liittyivät kuljetusten logistiikkaan ja tiedonsiirtoon Suomessa ja Venäjällä toimivien yritysten välisessä liiketoiminnassa. Kunkin case-kohteen osalta tehtiin tarvekartoitus, määriteltiin pilottiratkaisu ja kehitettiin prototyyppitoteutus. *EU Entry Summary Declaration* -pilotissa valmistauduttiin tuleviin uusiin sähköisen ilmoittamisen vaatimuksiin Euroopan tullialueella. *Fleet Management* -pilotissa tuotettiin kevyt ratkaisu kuljetusten tilannetiedon välittämiseen, hyödyntämällä tavallisia matkapuhelimia. *Trade*-pilotissa tuotettiin tekninen ratkaisu tiedonvälityksen tehostamiseen rajan ylittäviin kuljetuksiin liittyvissä toimitusketjuissa.

Projektin toisen päätavoitteen mukaisesti suunniteltiin ja toteutettiin ohjelmistoalusta tukemaan edellä mainittujen ja muiden vastaavien liiketoiminnallisten case-kohteiden kehitystä. Palvelukeskeisen arkkitehtuurin (SOA) periaatteita hyödynnettiin laajalti. SOMAF-alusta onnistui tarjoamaan perustan pilottiratkaisuille, ja pystyy tukemaan vastaavien kohteiden kehitystä jatkossakin. Raportin lopuksi esitetään eräitä johtopäätöksiä ja suosituksia jatkokehitykselle.

Luottamuksellisuus	julkinen

Espoo 27.2.2009

Laatija	Tarkastaja	Hyväksyjä	
Renne Tergujeff Tutkija	Joni Niemi Tiiminvetäjä	Markus Tallgren Teknologiapäällikkö	Jussi Paakkari Teknologiajohtaja

VTT:n yhteystiedot
VTT, PL 1000, 02044 VTT, Finland

Jakelu (asiakkaat ja VTT)
Liikenne- ja viestintäministeriö, Canon North-East Oy, UPM-Kymmene Wood Oy, UPM-Kymmene Oyj, Russian Cargo Service Oy, Tekes – teknologian ja innovaatioiden kehittämiskeskus, VTT

Preface

This report describes the work and results of the Service Oriented Mobile Application Framework (SOMAF) project in the VAMOS – Value Added Mobile Solutions (VAMOS) programme of Tekes, the Finnish Funding Agency for Technology and Innovation. The project ran from June 2007 until December 2008, and it focused on service-oriented approaches for supporting flexible information exchange in business environments that often incorporate also mobile terminals.

Partnering companies in the project offered three cases, related to transport logistics in business between companies in Finland and Russia. These partners were UPM-Kymmene Wood Oy (plywood and veneer), UPM-Kymmene Oyj (sawmilling), Canon North-East Oy, and Russian Cargo Service Oy. Also participating and funding was the Ministry of Transport and Communications of Finland. The involved research party was VTT Technical Research Centre of Finland, while the main financier was Tekes, the Finnish Funding Agency for Technology and Innovation.

The steering group of the project had a representative and a deputy member from each of the participating organizations:

organization	primary member	deputy member
VTT Technical Research Centre of Finland	Markus Tallgren	Jyrki Haajanen (until 1 Nov 2008)
Ministry of Transport and Communications of Finland	Lassi Hilska	Tuija Maanoja
Canon North-East Oy	Pentti Nelimarkka	Lasse Fagerström
UPM-Kymmene Wood Oy / UPM-Kymmene Oyj	Erkki Oikarinen	Erkki Pietikäinen
Russian Cargo Service Oy	Lina Solovieva (until 1 May 2008)	Elias Heikari (until 1 May 2008)
Tekes, the Finnish Funding Agency for Technology and Innovation	Jorma Julku	Keijo Silventoinen

We want to thank the members of the steering group, and all the people at the companies who participated in case development, for their contribution and input to the project.

Espoo 27 February 2009

Authors

Contents

Preface 3

1 Introduction 5
1.1 The SOMAF Project 5
1.2 Research Background 5
1.3 Objectives and Scope 6
1.4 Research Methods 7

2 SOMAF Platform 8
2.1 Generic Requirements 8
2.2 Technical Background 9
2.2.1 SOA and Web services 10
2.2.2 Application Layer Frameworks 10
2.2.3 Mobile Device Environment 11
2.2.4 Security 11
2.3 Platform Architecture 12
2.3.1 Platform Services 13
2.3.2 Service Implementations 14
2.3.3 Web Service Handler 15
2.4 Results 16

3 EU Entry Summary Declaration Pilot 17
3.1 Requirements 17
3.2 Business Background 18
3.3 Design 19
3.4 Implementation 20
3.5 Results 21

4 Fleet Management Pilot 23
4.1 Requirements 23
4.2 Design 24
4.3 Implementation 24
4.4 Results 27

5 Trade Pilot 28
5.1 Requirements 28
5.2 Design 30
5.3 Information Exchange Process 31
5.4 Implementation 33
5.5 Results 36

6 Conclusions 37

References 39

Appendices 41

1 Introduction

This report describes the platform development and piloting work performed from June 2007 until December 2008 in the SOMAF project (Service Oriented Mobile Application Framework). The project was part of the Tekes technology program “VAMOS – Mobile Business Solutions”.

This first chapter introduces the project and relevant background research, while the second chapter describes the developed software platform. Chapters 3 to 5 describe the requirements, design and implementation of three pilot solutions, each developed to answer the challenges presented by a business case from a partner company in the project. Finally, some conclusions are presented.

1.1 The SOMAF Project

The goal of the SOMAF project was to investigate and develop methods for supporting flexible information exchange, also in such business environments that incorporate and take use of mobile terminals. The project was set to produce a software platform with complementing service components, suitable for supporting various business cases. The partnering companies in the project offered three cases, related to transport logistics. Pilot implementations were planned for these cases, and they were expected to help confirming the general applicability of the platform solution.

One case dealt with import of wood products from Russia to Finland; the corresponding *EU Entry Summary Declaration* pilot focused on preparing for the forthcoming new requirements for electronic procedures with European customs authorities. The case for the *Fleet Management* pilot focused on information exchange with vehicles of Russian transport companies, who provide services to a Finnish forwarding company. Here, the main goal was to allow fleet management and tracking and tracing of the shipments from Finland to Russia. Third case concerned enhancing information delivery between logistics chain participants in the export of electronics products from Finland to Russia. The *Trade* pilot mainly aimed for achieving more efficient generation of required export/import documentation for individual goods shipments.

1.2 Research Background

In 2006, VTT performed the project “Suomen ja Venäjän välisten logistiikan tietojärjestelmien yhteensopivuus” (FINRUSLOGICT), funded by the Finnish Ministry of Transport and Communications The project studied compatibility issues in logistics information systems between Finland and Russia, and identified several problems that logistics chain stakeholders meet in their daily operations. A set of piloting proposals for selected problems was created. In 2007, the Ministry launched a call for tenders for coordination of these piloting efforts. A consortium led by VTT was selected for this task, and the project FINRUS2 ran through to November 2008. These pilot proposals have great economic and environmental potential, approaching the problems from the logistics perspective. Enabling realization of the pilots requires a coordinated ICT platform development work, which was the target of the SOMAF project described in this document. Benefit was expected from collaboration between the projects throughout their execution.

The SOAMeS project (Service Oriented Architecture in Multichannel e-Services), which ran from 2006 to 2008, studied the present state and the future of service-oriented architecture (SOA) and the applicable technologies. The research partners in the project were University of Helsinki and VTT, while main funders included Tekes and the participating companies. Main research themes of SOAMeS included creating and managing dynamic collaborations in an open networked business environment, and the required middleware infrastructure. Interoperability management at process and pragmatic level, and non-functional requirements emerging from business strategy needs were investigated as well. The project also aimed to produce a roadmap for enterprises for adopting SOA and the necessary tools into enterprise computing and inter-enterprise computing. Industry-partner case projects studied applicable methodology for SOA-based management of business processes and platforms.

In 2006, a technology development project called Mobile SOA was run internally at VTT, to evaluate the potential of applying service-oriented architecture (SOA) in mobile devices (phones). In the project, a design concept and a software solution was drafted, for enabling flexible integration of everyday mobile phones to systems designed with service-oriented approach. A general-purpose solution, based on well defined service interfaces, was sought, freeing from the underlying implementing technologies. The prototype solution is usable on nearly all currently available mobile phones, making it well suited for developing enterprise level, small business as well as consumer applications. Investments can be kept to a minimum. As an example of the opportunities, information exchange between companies and their mobile clients or employees can be easily improved, through a light software solution for communication between the enterprise systems and applications run in small user terminals. The project produced promising results, showing that real interoperability potential exists from wide range of terminals to unified service oriented access points. The experiences prepared VTT for developing the planned technology platform.

1.3 Objectives and Scope

ICT enhancements can provide significant benefits in logistics chains, via better and timely information delivery among the stakeholders. This project aimed to provide technological means for flexible and transparent information flow in the logistics chain, taking into account different requirements of various parties. Such requirements often include the ability to tie mobile parties into the information exchange.

Solutions developed in this project were intended to profit organizations participating in the import/export logistics chain between Russia and Finland. Benefits were expected in preparation and delivery of border crossing documents, and in the information exchange between logistics operators (especially Russian operators operating in Finland) and logistics hot-spots, such as warehouses, plants and terminals.

The scope of the project covered design and implementation of a generally applicable technological platform that enables development of solutions for the above and similar needs. Also in the scope of the project were planning, implementation and piloting of several individual cases, answering to the specific needs of the project participants. Transformation of successful pilots to operative systems was considered outside of the project scope.

1.4 Research Methods

The project work covered two main subjects, supporting each other: on one hand, the design and development of a generally applicable software platform for flexible information exchange, and on the other hand, the planning and implementation of three pilot systems, originating from cases provided by the participating companies.

The work was planned to be carried out in two phases, each running from investigating business requirements, through to pilot implementations. The results of the first phase of the project were to be used in further planning of the second phase. Eventually, the work was mainly performed in a single pass throughout the duration of the project. This was mainly due to the long period required for producing a focused specification for each of the pilots, which in turn delayed completion of the pilot implementations.

Design of the software platform was, for a large part, carried out and documented as a Master's Thesis work of Timo Toivanen, called "Developing a Service-Oriented Platform for Information Sharing" [Toivanen08]. The planning process involved an analysis of the state-of-the-art of current similar software platforms. Work for each of the pilots included several interviews with the representatives of the participating companies and some of their business partners involved in the cases, both in Finland and in Russia. Acquiring understanding of the business cases also required investigating some orders of the authorities. For each pilot, a Software Requirements Specification was produced, helping to confirm the common understanding of the required functionality.

2 SOMAF Platform

This chapter describes the software platform developed in the project, covering requirements, technical choices, design and implementation. The main objective for the platform development was to provide basic means for flexible information exchange between business partners easily and efficiently. An important goal was to support several types of business cases, freeing from the need to design and implement a solution for each new case from scratch.

2.1 Generic Requirements

The business cases in the project originated from logistics chains between Russia and Finland. These chains as environments provide a rich set of development and research challenges. The most important features to be studied were information delivery among distant actors and information transparency within the chain of actors. All the main stakeholders were covered on some level, including trading parties, forwarding agents, store & handling services providers, hauliers, truck drivers, brokers, and the authorities in both countries.

Individual requirements for the pilot implementations in each case were collected in interviews with pilot stakeholders. Furthermore, the aim was to identify and report more general requirements and features that may have interest to other companies and organizations, working on similar development projects related to border crossing, or otherwise dealing with logistics between Finland and Russia.

This report collects issues identified during the requirements gathering early in the project. The reported requirements can roughly be divided in three categories. Business requirements cover business goals, and requirements of the case company and the stakeholders. Technical requirements consider infrastructure, competence and ability to execute of the case company and the stakeholders. Other, miscellaneous requirements include legal and other legislative requirements and possible restrictions. Although this is a requirements gathering report, its subjects address a broader scope in all the aforementioned categories. The purpose here is more to introduce aspects to consider in development work, rather than to define exact requirements for any single system implementation.

Trust & security: Special characteristics identified in the business culture in Russia need to be considered. Generally, business needs to be done in a secure and information-safe manner, without unnecessarily exposing the identity of business partners, or the scale of the collaboration to parties with potentially conflicting interests. This can be a significant safety issue in doing business, applying also to the security requirements of information systems. Classification: business requirement.

Scalability: The size of the companies in logistics business varies a lot in Russia – small companies are dominant as in Finnish markets, although some major players exist too. The SME’s are typically not combined by larger logistics service provider companies like in Finland. Process design and information system development may have to consider layered approach towards different parties. Classification: business requirement.

Adaptability: In Russia there is a wide range of different information system solutions, being very often tailored and based on office applications such as Microsoft Access or Excel. This applies also to data presentation formats. Companies need to ensure the availability and arrangement of transformation services, or provide a solution replacing the need to link with a large group of existing systems. Classification: technical requirement.

Cyrillic: Often Cyrillic alphabet seems to provide problems for software such as ERP systems. Russian counterparties are often working on tightly closed IT environments, and integration to such systems can be very complicated, due to the lack of interface and other information from the providers. Classification: technical requirement, with business process features.

Logistics hardware gadgets: Transport vehicles in Russia are typically not equipped with modern ICT infrastructure; however, many tasks can be handled by utilising the drivers' mobile phones. Maintenance problems and investment safety are risks when considering vehicle dependent equipment investments. Classification: technical requirement.

National customs tariff systems: The tariff systems used in EU and Russia are partially but not fully compatible. Changes in the tariff systems are also often introduced relatively rapidly in certain product classifications. Classification: miscellaneous requirement.

Paper oriented processes: Many processes related to transport logistics in Russia are "stamp-oriented"; frequently documents need to be stamped for authorization at several process phases. This may pose a major hindrance to e-oriented approaches. Paper reporting will probably remain important in the close future. Classification: miscellaneous technical requirement; process and technical aspects need to be considered.

Legal issues: Differences in legislation need to be carefully considered. Classification: miscellaneous requirement.

Communications: Technical feasibility issues related to roaming etc. need to be considered when developing mobile solutions based on communication via the mobile phone network. Classification: technical requirement.

Accelerated information exchange: Exchanging information automatically assures that the business partners always have the same information available, and that it is available as quickly as possible. Efficient information sharing and business process automation can help enterprises to improve their productivity and increase customer responsiveness. Classification: business requirement.

2.2 Technical Background

As discussed in detail by Toivanen in his Master's Thesis [Toivanen08], the requirements for flexible and accelerated information sharing between business partners encourage adoption of service-oriented architecture (SOA) and Web service technologies, which are briefly presented in the following.

2.2.1 SOA and Web services

SOA is an architectural design style for complex, distributed software systems, where functionalities encapsulating business logic are described and offered as services. The core underlying principles include autonomy of services, loose coupling between components, formal contracts, and abstraction. Solutions based on SOA can offer increased agility, enabling faster adaptation to new requirements from changes in business environment or by new technology.

Solutions based on SOA provide more flexibility for information sharing than traditional enterprise integration methods like EDI (Electronic Data Interchange). One problem in using EDI for business connections is that it does not scale very well. [Singh05] When the number of partner enterprises increases, managing linkages becomes increasingly difficult. Adopting EDI in an enterprise is a complicated process where every new connection has to be created separately, increasing the adoption costs. EDI integration can still be useful for large businesses with well-defined trading relationships, but is less suited for the rapidly growing and changing global marketplace.

The term “Web services” refers to a set of technologies most commonly used for realizing systems with a SOA approach. Existing services can be reused when developing new Web services. This reduces development costs and increases the speed of development. Most commonly, Web services are described using the standard Web Service Description Language (WSDL). Communication with services is handled by requesting execution of their operations that are described in their published interfaces.

Because the interfaces are platform independent, Web services don’t impose any requirements to implementation methods for client applications. SOAP with HTTP provides a platform independent communication protocol, where one advantage is that it allows communication also through proxies and firewalls. Alternatively, a lighter but more restricted approach to Web services, based on Representational State Transfer (REST) [Fielding00], can be taken. As a supporting technology, Extensible Stylesheet Language Transformations (XSLT) can be used for easy automation of information transformations to suitable formats for requestors.

Unlike many traditional technologies, Web service technologies are based on open standards that are freely adoptable without expensive royalties or license fees. The use of Web service technologies can help lowering costs, offering a more affordable technology for enterprises of different sizes, including the SME’s. [Toivanen08] Most Web service standards are generally accepted and vendor neutral, and used by many industrial parties.

2.2.2 Application Layer Frameworks

The essential frameworks for implementing Web services are .NET Framework and Java. Java is a programming language developed by Sun Microsystems. It is platform independent, and a program written in Java runs similarly on any Java supported hardware and operating system. Java is divided into three major development and runtime platforms, each addressing different types of solution. Java Platform, Standard Edition (Java SE) is designed to support creation of desktop applications. Java Platform, Enterprise Edition (Java EE) is built to support large-scale, distributed solutions. Java Platform, Micro Edition (Java ME) enables developing and running programs in mobile and other embedded devices.

.NET is Microsoft's framework for programming applications and Web services on Windows clients, servers, and mobile or embedded devices. [Singh05] The lack of free development tools for .NET applications is a disadvantage. On the other hand, the commercial development tools by Microsoft are considered to be very good. .NET also has support for different programming languages, like COBOL, C#, and Visual Basic. An open source implementation of .NET called Mono has been created for UNIX, Linux and Mac OS X architectures.

All of Java's core code is open-source. Many additional free libraries and APIs are also available, as well as lots of free and open-source tools for developing Web services. In addition, many Java-based open-source web servers and databases exist, usable for free in commercial environment. With .NET, options are mostly restricted to Microsoft's products that can require expensive licenses, increasing the development costs too high for many SMEs. Overall, Java platform can be considered advantageous over .NET in creating and deploying Web services. It has better support for different hardware and operating systems, while .NET can be fully exploited only in Microsoft Windows environment. In a Windows environment, both are viable options and choosing between them is a difficult decision.

2.2.3 Mobile Device Environment

The most widely spread runtime environment for mobile devices is Java™ Platform, Micro Edition (Java ME), more specifically the Mobile Information Device Profile (MIDP) version 2.0 [MIDP2]. This platform is found in great majority of all mobile devices published in recent years, including not only the high-end smart phones, but also the more low-priced devices with limited resources. The mobile Java platform enjoys dominant market penetration globally, and it has sustained its popularity over the years. With some limitations, standard Web Services can be utilized, based on specifications such as WSDL for service interface definitions, and SOAP and HTTP protocols for communicating over the mobile network. These specifications should be used as guided by the WS-I Basic Profile [WS-IBP], published by the Web Services Interoperability Organization.

2.2.4 Security

Standards and measures for security should be implemented at every leg of a distributed application, and it needs to be done consistently across all of the organizations where the application is used. Security of information systems can be assessed on several levels. On a transport layer, security can be ensured by using Secure Socket Layer (SSL). Combined with HTTP basic authentication, it provides generally sufficient security level for confidentiality, authentication, and integrity.

For more complex applications, involving multiple parties and services, messages among end points should be encrypted and signed to protect their confidentiality and integrity. When communicating using Web service standards, combining WS-Security with XML Signature and XML Encryption standards is a common method to secure Web services. They can be used to fulfil all the typical requirements for secure information sharing. [Toivanen08]

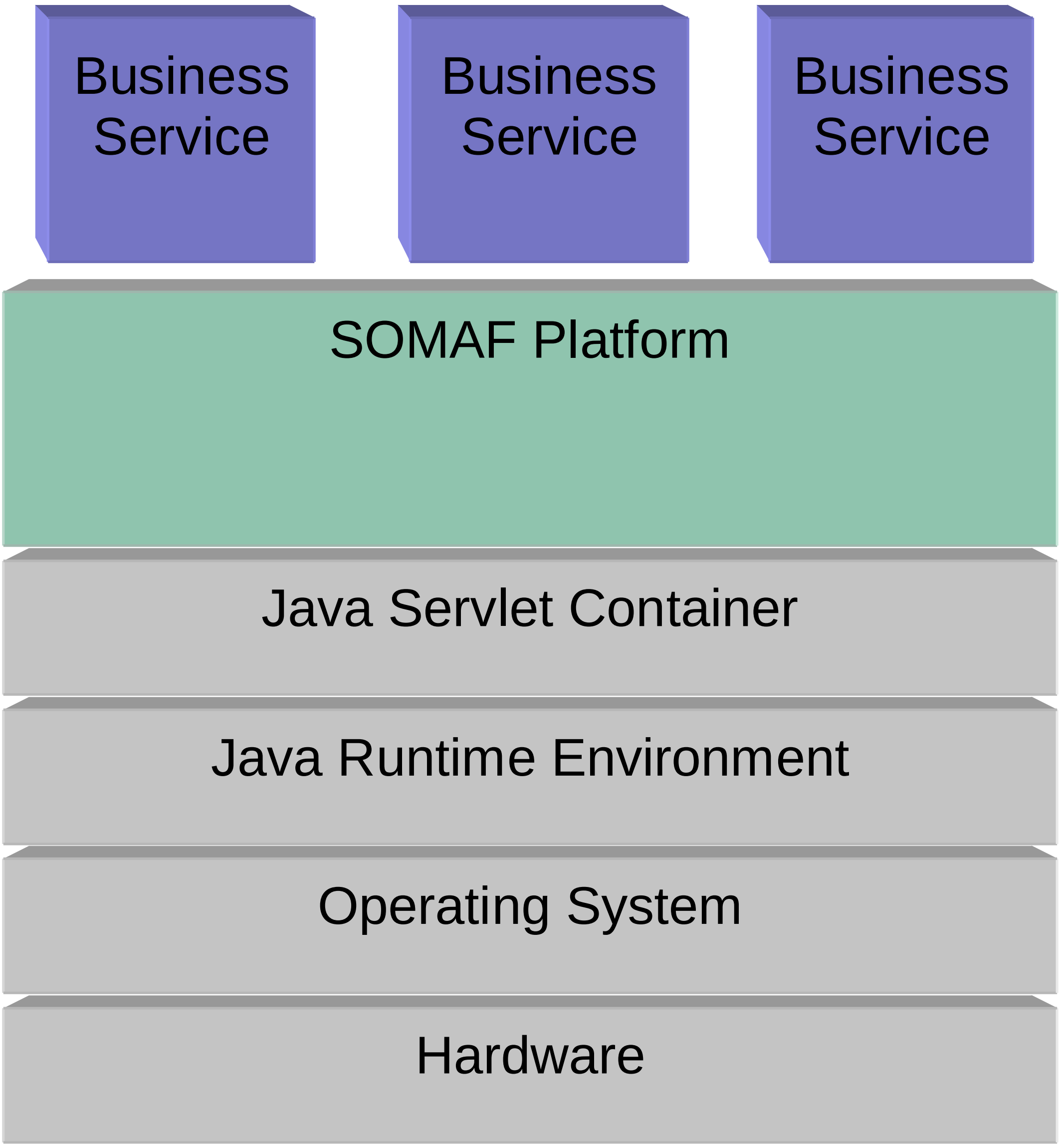


Figure 1. Layered view of the SOMAF platform installation with several Business Services.

2.3 Platform Architecture

The SOMAF platform is designed using the principles of service-oriented architecture. The platform contains several Web services, offering various functionalities. It is designed to be as generic as possible, so that it could be used in different business environments and processes. Enterprise solutions are created by developing various business services on top of the platform. These business services take use of the existing services available on the platform, and provide themselves some specific functionality that is needed by the enterprise. It is also possible for an enterprise to open the business service interfaces to its business partners, who can then plug into the services as well.

There are several ways to interact with the business services. A service client can be a standalone application, another Web service, or a web client, used through a web browser. Optimally, to minimize the need for manual operations, the client functionality can be integrated into the enterprise's ICT systems. In addition to the business services, clients can also interact directly with services inside the SOMAF platform, if the services are configured to be accessible from the outside.

Figure 1 illustrates the layered view of the SOMAF platform installation and the position of business services in the architecture, on top of the platform. Figure 2 presents the general internal architecture of the SOMAF platform, consisting of a Web service handler, set of Web services inside the handler, a transactions log, an SMS sender/receiver, and databases, where information can be stored. The only technical requirements for deploying the platform are that the target computer has to have the Java runtime environment and a web server with Java Servlet container installed. Also, the target computer has to be connected to a network so that services offered by the platform can be consumed over the Internet. Otherwise, the SOMAF platform is hardware and operating system independent.

2.3.1 Platform Services

The core set of implemented services on the SOMAF platform offer basic functionalities that are needed in several business scenarios. Additionally, the architecture allows creating high-level services, which offer more specialized functionalities. These take use and extend the functionalities provided by the core services. Service interfaces are described using WSDL, and service requesters can interact with services using SOAP messages.

- *FTP (File Transfer Protocol) and SFTP (Secure Shell FTP) Services* offer client functionality for file transfers, enabling sharing of information with various legacy systems.
- *Database Service* provides a high-level interface for database operations.
- *Email Service* is used for programmatic sending and receiving of emails, e.g. in sending automatic notification emails based on defined events.
- *Transformation Services* handle transformations between different data formats.
- *Authentication & Authorization Service* is used to authenticate incoming Web service requests and to define what rights service requester has on the target service.
- *Mobile Management Service* is a high-level service, used for sending and receiving SMS messages, and for managing mobile clients that can interact with the services of the SOMAF platform.

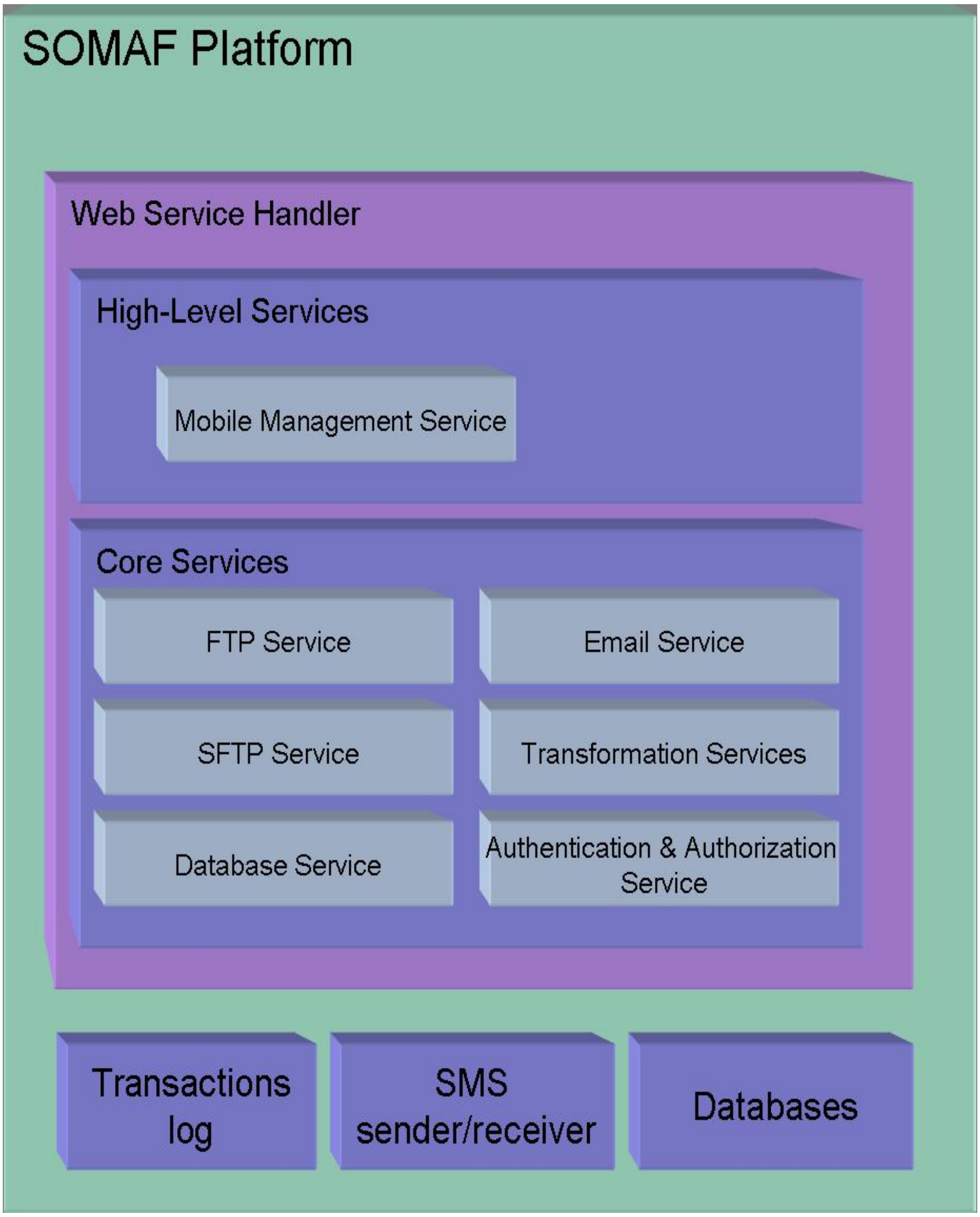


Figure 2. General architecture of SOMAF platform

2.3.2 Service Implementations

All the services offered by the SOMAF platform are implemented using Java. However, as the services are accessed through their interfaces, the chosen implementation techniques are not visible to the service clients. The services can be consumed through both SOAP and REST protocols. This allows utilising the services from clients that are developed using any programming language that has support for Web services, or the HTTP methods used in REST.

With the help of FTP and SFTP services, files can be downloaded from, and uploaded to FTP servers. SFTP service provides secure connections using SSH, and should be used when data security is required in the information exchange.

This protects the transferred information by encrypting it, so that potential eavesdroppers can not exploit the information.

The Database Service provides a higher-level interface to basic SQL (Structured Query Language) commands; insert, select, delete, and update. As input parameters, the service takes information like the database to use, table name, SQL command parameters, and the username and password. The service returns the results in XML format.

The Transformation Services, based on XSLT, enable transforming XML documents to various data formats, like PDF or plain text. Several transformations can be defined for producing documents in different formats.

Mobile Management Service uses the database service for managing information about mobile devices. The service provides operations to add new device information, and to update, delete and search existing device information (see Appendix 3). It receives SMS messages from a physical SMS receiver device and transforms them into a suitable format based on the phone number of the SMS sender. Then it forwards the messages to correct target services. The mobile management service can also be used to send SMS messages to mobile clients. As input, the service takes a phone number and the message to be sent.

The SOMAF platform services can be consumed over secured SSL connections, and service requesters are authenticated using HTTP basic authentication when accessing the services. Additionally, all the messages inside the platform are sent using SSL and HTTP basic authentication. This combination provides security for end-to-end connections. SSL with basic authentication provides secure enough environment for piloting the case implementations. When using the SOMAF platform in a production environment, additional security measures are required to provide message non-repudiation and authorization. For example, the WS-Security specification can provide for all typical security requirements needed in production environment.

The lowest layer of the SOMAF platform architecture contains some underlying components. Information is stored in MySQL databases, which can be located on the same computer as the rest of the SOMAF platform, or they can be accessed over the network. SMS sender/receiver is used for mobile communication using SMS messages. This functionality is achieved by connecting a normal mobile phone to the server computer, and using appropriate driver software. It enables for mobile devices without support for Web services to interact with services offered by the SOMAF platform. Additionally, the transactions log records information about the actions inside the platform. This information can be used for monitoring the activities in the platform.

2.3.3 Web Service Handler

The services of SOMAF platform are deployed inside Apache Software Foundation’s Web service handler called Axis2, which provides an implementation of the SOAP protocol. [Axis2] The basic functionality of the software is routing incoming SOAP requests to correct services, and outgoing SOAP responses to their right destinations. Axis2 is open-source software, available without costs. It is licensed under the Apache License v2.0, which allows free commercial use, and also enables making modifications to Axis2 without having to make it freely available.

Implementations of Axis2 are available in Java and C programming languages, which are also the languages for creating the service implementations. SOMAF Platform is built using the Java-based version. Axis2 can act as a standalone server or it can be deployed to Java Servlet containers like Apache Tomcat and Glassfish, the latter being the typical scenario. Axis2 also offers support for RESTful Web services, which refers to a lightweight architectural style for designing simple Web services. However, this method of offering services has limitations that prevent its use in complex enterprise application integration scenarios. [Pautasso08]

Services for Axis2 can be developed using several different data binding techniques. Data binding is the term used for the technique that handles the conversion between XML and application data structures. Services in the SOMAF platform are created using Axis2 Data Binding (ADB), which is a lightweight and simple data binding technique. [Sosnoski07] It has also good support for code generation from schema. The limitation of ADB is that it is specific to the Axis2 environment. XMLBeans [XMLBeans], another data binding technique, provides the most complete support for modelling XML Schema structures in generated Java code. Unfortunately, it creates the most complex Java model for a given schema. JiXB [Sosnoski08], on the other hand, is mainly focused on working with existing Java classes, but it has limited support for generating Java classes from XML Schema.

2.4 Results

The evaluation of the developed platform was based on the experiences gained from implementing the three different pilot implementations. In the pilot implementations, several business services were created on top of the SOMAF platform, and used to support various business processes.

The SOMAF platform came off well in implementing these pilots. Based on the pilot experiences, the platform can be effectively used to accelerate and automate the information flow between different interest groups. It is easy to define transformations of information to suitable formats for information receivers. In addition, new services can be created quickly, and modifications to existing services are convenient to make. The platform uses generally accepted standards for communication and data formats, which facilitates information sharing between various applications and information systems.

3 EU Entry Summary Declaration Pilot

The “EU Entry Summary Declaration” pilot investigated solutions for a case provided by UPM-Kymmene Oyj and UPM-Kymmene Wood Oy (UPM). The focus of the pilot solution was to facilitate sending electronic advance notification – entry summary declaration (ENS) – of goods arriving to the EU area, as required by customs regulations starting from July 2009.

While the first meetings with the company representatives were held in the autumn of 2007, the work for this pilot was deferred until early 2008. Due to the partly unclear and continuously developing business requirements set by the Finnish customs, specifying the actual case and the possible solutions took a very long time. This work phase included also a meeting with a transportation/logistics operator company in Moscow. Several possible directions for development were investigated and considered, before deciding on the approach to take in the pilot. In the autumn of 2008 the case was adequately clarified to proceed to implementation work, which was then progressed rapidly, with two near-complete implementation cycles completed before the end of the year.

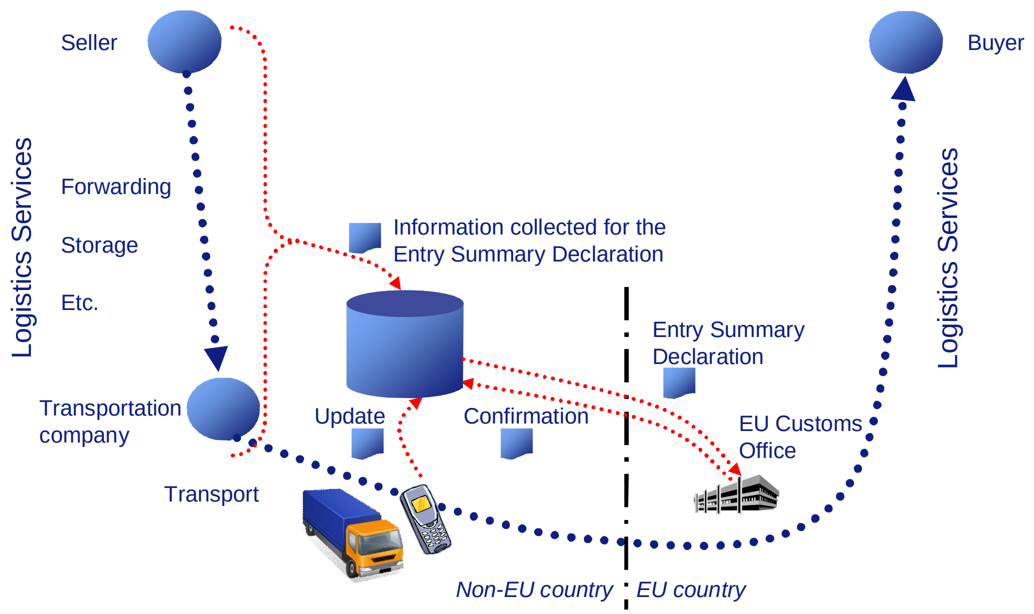


Figure 3. Advance notification (entry summary declaration) will be required of all goods transported from outside the EU to the union’s territory. The required information is created in the processes of seller, transporter and other logistics service providers. The declaration can be updated with changed data; in the model investigated in the pilot, to some extent also via the driver’s mobile phone.

3.1 Requirements

The commission regulation 1875/2006 [EC06] of the European Council (EC) requires an electronic entry summary declaration from all goods arriving to the territory of EU, starting from July 2009 (Figure 3). The transporter of the goods is responsible for making the declaration; however, it is expected that in many cases the importer (buyer) of the goods will arrange the declaration. In the case of Finland, the declaration is made to the forthcoming AREX system of Finnish Customs, either through a message based interaction of IT systems, which would enable efficiently automating processes, or manually, using a web form. For the case company, the goal is to fulfil these new external requirements, and at the same time seek efficiency through enhanced flow of information within the enterprise and its partner network. [Tulli08]

3.2 Business Background

The partnering companies in this pilot were two UPM-Kymmene units: sawmilling (from UPM-Kymmene Oyj) and veneer and plywood business (UPM-Kymmene Wood Oy). The production plants are located in Russia: *Sawmill and Planing mill* in Pestovo, and *Plywood mill* and *Veneer mill* in Chudovo. The total volume of production is about 400 000 m^3 per year, equivalent of about 10 000 lorry loads. Major share of this volume is transported on land routes to Finland, with some of the volume carried on to third countries.

In the import of wood products, UPM uses a few Russian subcontractors, responsible for transporting the goods from the production plants to Finland. Some of these are purely transport companies, some work as logistics operators. Some of the goods are transported by sea. The company uses periodic clearance, where aggregated customs declarations are delivered monthly. Some of the transported goods are customs cleared to Finland; some are transit freight being carried elsewhere.

In the veneer and plywood production plants in Chudovo, several parties participate in the preparation of documents: production planning, warehouse, and import-export department, which plans and orders transportation from forwarders and finally delivers the documents and their attachments to the local customs office. The attachments of export declaration include, among others, CMR waybill and pro forma. The documents are finalised only when loading the cargo, when the warehouse registers the loaded batches to the information system.

The production plants of Chudovo and Pestovo have differing information systems, which are relatively troublesome to develop further. Move to a newer WSS system is ongoing in Pestovo, while in Chudovo a SAP-based system is planned, approximately from 2010 onwards. Inter-enterprise connections have been implemented to the systems in corporate group level, and to the systems of Russian production plants and subsidiaries. Outside connections are available to customs and to the logistics service providers. The implemented connections are for a large part EDI based. Little to none mobile systems are in use.

UPM is participating in the partnership program of Finnish customs, where the goal is to ease the procedures in the interactions between the customs and trusted partners. Through the partnership the company is kept well up-to-date in the developments in customs procedures.

The wood product units of UPM are planning to take customs transactions in their own hands, giving up using a separate customs representative. In the first phase this requires acquiring and implementing a separate application for handling the customs communication. In a later phase, communication with the customs could possibly be handled using the new enterprise information systems.

3.3 Design

At present, the production plants in Russia already need to gather extensive information about transports for making the export declaration. This same information set can be used for preparing the entry summary declaration, in the format required by the Finnish customs. The planned solution (Figure 4) involves electronic sending of the information set to the company's customs communication system in Finland. This system can be an outside purchased solution, or it can be an integrated part of the company's own systems.

At the sending factory, information is entered either to a web form provided by the company's customs communication system, or to a separate locally run application, from where the information is transferred to the system. In either case, a correctly formulated entry summary declaration message is sent to the system of Finnish Customs, from where the information is forwarded also to the customs station at the border. If the declaration is correctly formulated, the customs will answer with a Movement Reference Number (MRN), which the truck driver will need to present at the border crossing customs station.

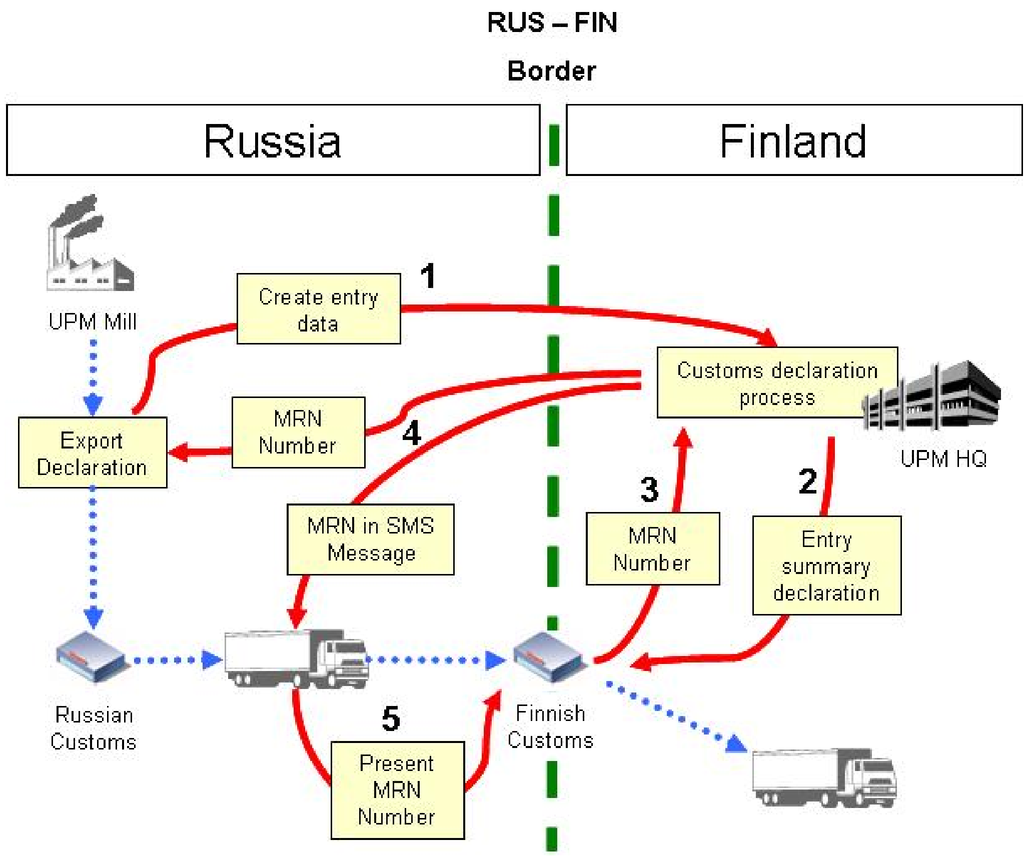


Figure 4. The planned process for handling the process of entry summary declaration. The dotted line refers to moving of the goods and solid lines to information flow. Based on presentation by Erkki Oikarinen and Erkki Pietikäinen in St. Petersburg in 12 November 2008.

In the pilot solution (Figure 5), the truck driver is integrated to the process of information transfer, via his/her mobile phone. The MRN received from the Customs is forwarded to the driver's phone, where it is readily available to present to the customs official at the border station. In addition, with his mobile phone the driver is able to make minor updates to the given declaration. These changes automatically trigger the sending of an updated entry summary declaration to the Customs, which again results in a received MRN number that is also forwarded to the driver.

The pilot solution supports manually entering, modifying and retrieving the declaration data. Importantly, it also offers direct support for integration with other IT systems of the enterprise, both for entering the data from other systems and for extracting data to other systems.

3.4 Implementation

The pilot design and implementation work was carried out in two phases, both based on the open service architecture. In the first phase implementation, Customs Notification Pilot System (CNPS) was created on top of the SOMAF platform. For demonstration purposes, a mock-up "Enterprise System" was also created, providing tools for entering and managing the actual entry summary declarations. Additionally, a mock-up of Finnish customs interface was created, to receive the entry summary declarations and to send appropriate replies. For this phase, proprietary document models developed by VTT had to be used.

CNPS is used to manage driver's data, and to handle routing and transforming the ENS trigger and response messages to the right destination in the correct format. The database service is used to store data about truck drivers and their cargos.

The mobile management service is used to send information as SMS messages to truck drivers, and to receive SMS messages from the drivers. Incoming update messages from drivers, containing information about the customs station and/or the estimated arrival time to the border, are forwarded to the mobile management service. Mobile management service transmits the information to CNPS, which triggers the sending of ENS by sending to the enterprise ICT system. Later, the response messages from the enterprise ICT system are again sent to drivers. If the ENS request was accepted by the customs, the MRN number is forwarded to the driver. Else, information about the ENS rejection is sent to the driver.

The "Enterprise System" allows for entering, modifying, saving, retrieving and sending of declarations. The system also allows monitoring the produced declarations. Responses from the customs system, and the updates provided by drivers on the road are updated dynamically. The mock-up of Finnish customs interface receives and processes the entry summary declaration, creates a MRN number and returns it back to the sender. A simple user interface was also created for observing the incoming declarations and the outgoing reply messages.

In the second phase implementation, CPS (Customs Pilot System) was created on top of the SOMAF platform. The objective was to integrate the functionalities provided by the CNPS with utilities for managing the information required to send ENS to the customs. Web service interfaces are offered for managing the information and sending the ENS to the customs. This second implementation was able to exploit the actual XML schemas published by the Finnish customs. The schemas define the required data structure for entry summary declaration. The user interface was also enhanced in the second phase, redesigned to be more compact and easier to use than the initial interface (see Appendix 1).

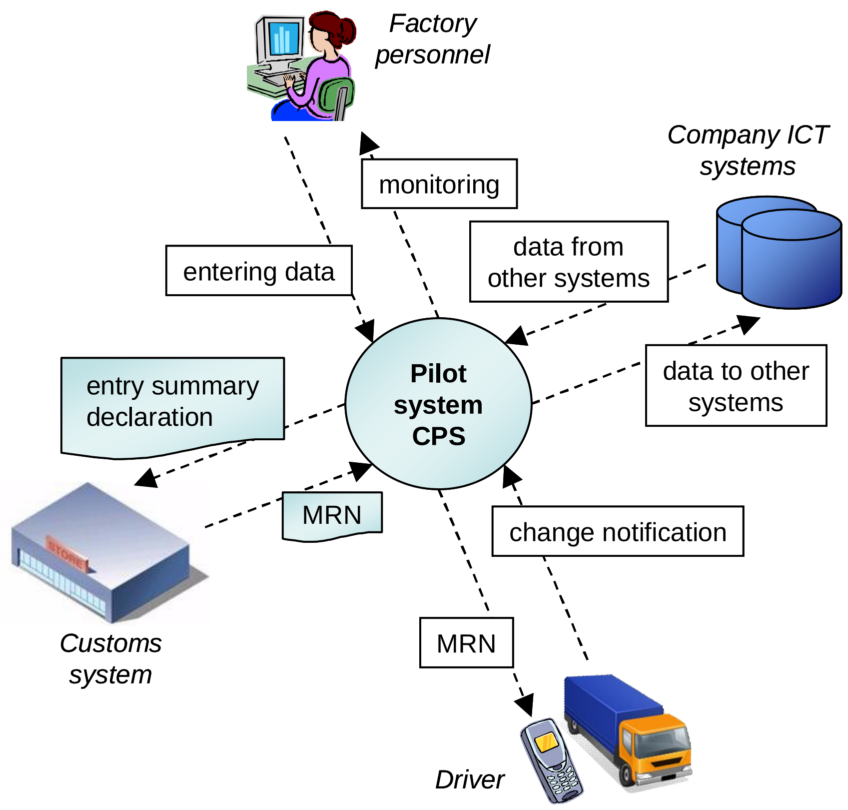


Figure 5. In the modelled solution a pilot system CPS was implemented, for producing the entry summary declaration to the customs. The necessary data can be entered at the factory, or it can be transferred from other systems to the interface provided by CPS. The data can also be forwarded to other systems, such as forwarder or the receiving factory. The customs replies with a MRN number, which is automatically delivered also to the transporter of the goods. The declaration can be updated with changed data, partly also from the driver's mobile phone. CPS enables monitoring and archival of the declarations, as well as user interfaces for different actors.

3.5 Results

The primary benefit from the piloted solution is the capability to answer to the requirements brought by new legislation. The piloted concepts enable both in-house implementations and solutions from service providers. With the methods used in the pilot solution, information collected for the purpose of advance notification can also be directly utilised in enterprise resource planning, including management of warehouses and transportations. In this context, it is possible to achieve significant gains, for example through more efficient resourcing.

The first phase implementation was demonstrated, although not piloted in working environment. The latter phase implementation could not be completed because of the time constraints set by the late publishing date of document models by the Finnish customs. However, it was demonstrated in a reduced form.

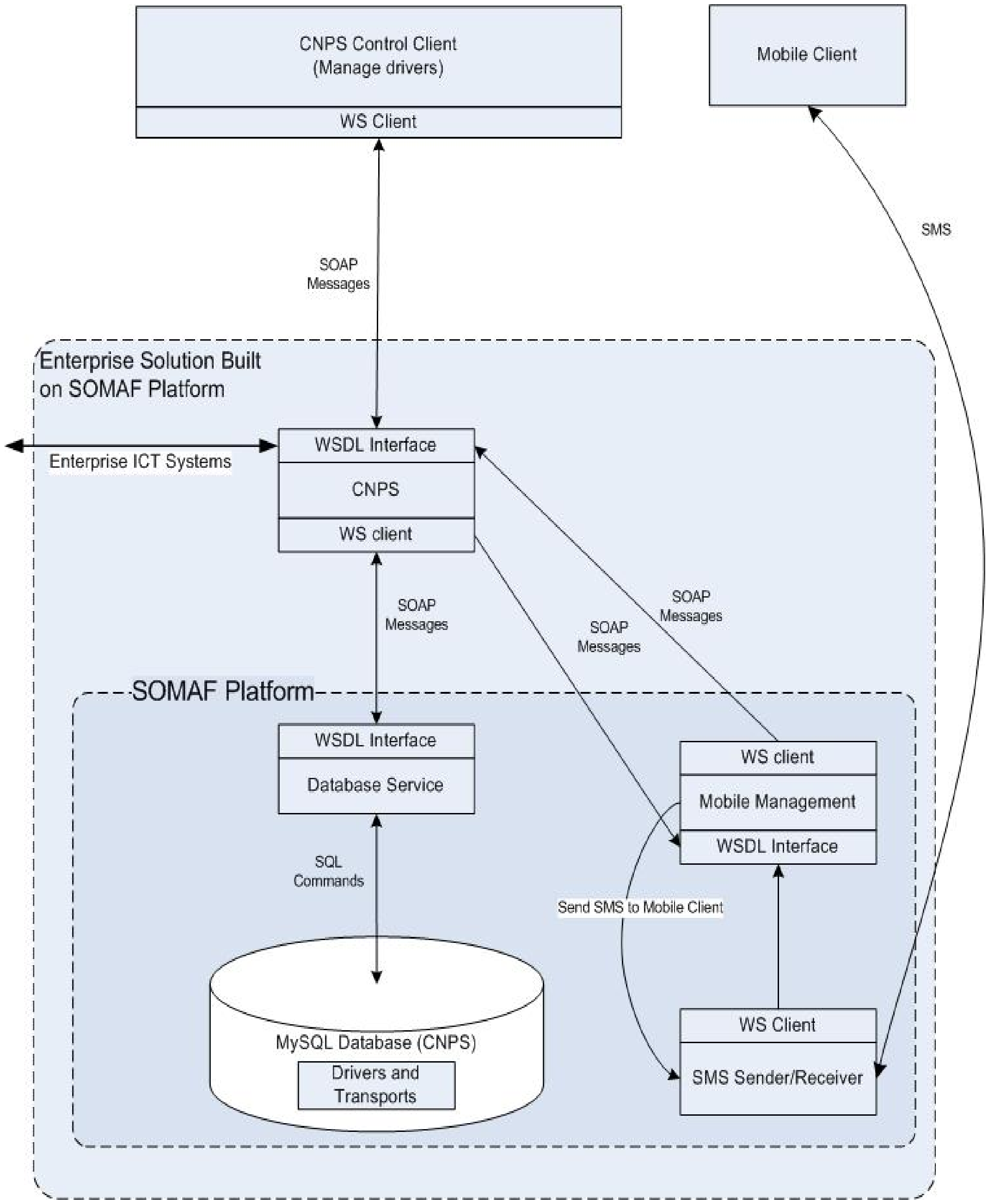


Figure 6. Implementation architecture of the EU Entry Summary Declaration pilot. Built on the SOMAF platform, the CNPS service handles the driver and transport information, utilizing the Database and Mobile Management services. A separate CNPS control client is used to manage driver information; in the pilot demonstration, a simple web browser interface was created for this purpose. Mobile client communicates with the platform using SMS messages.

4 Fleet Management Pilot

The Fleet Management pilot is based on a case provided by Russian Cargo Service Oy, a logistics service provider focusing on delivery of collected/combined cargo loads from Finland to Russia. The case aimed to pilot means for collecting tracing information of the transportation process, and providing more accurate information, enabling fleet management by exploiting the already available communications equipment on board the vehicles – the driver's mobile phone. The mobile phones can be used to provide the drivers a process interface for tracing purposes (Figure 7).

Work for this pilot was started in the autumn of 2007, when the first meetings with the company representatives were held. The requirement specification and the implementation work were carried out during the following winter and spring. In April 2008, the first pilot solution was completed successfully. However, major organizational change in the case company followed soon after, which eventually lead for cancellation of any plans for enhanced follow-up versions.

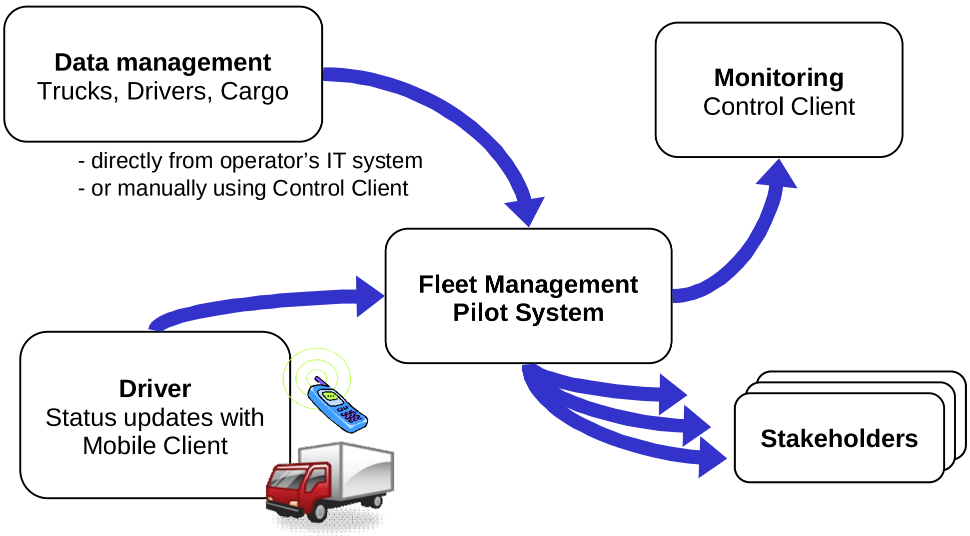


Figure 7. Overall view of the pilot solution, where transportation status information is collected for monitoring purposes and for transport chain stakeholders.

4.1 Requirements

The operations of the company strongly rely on the use of the services of Russian transportation companies. However, fleet management and tracing facilities currently available for such a combination of collaborating parties are not currently feasible. Savings in operations and business model development could be possible if more accurate tracing information on the transportations would be available.

On the company side, an open solution for managing the cargo data and the truck drivers, and monitoring the cargo data transportation status, is needed. The solution would need to primarily enable managing and monitoring the transportations with an ad hoc (stand alone) web solution, and secondarily enable integration of the features with the company's existing framework.

On the truck driver side, the main requirement is for an easy to use solution for status reporting with a mobile phone. The mobile application should guide the phone user in providing the appropriate and correct information about the transportation phases or problems encountered.

4.2 Design

The design concept (Figure 8) was to implement a solution based on a web service database for storing and managing information on cargo data and its transportation status, and on mobile phone applications for providing cargo status data for monitoring purposes. Several options exist for implementing the mobile client and the communication channel to databases.

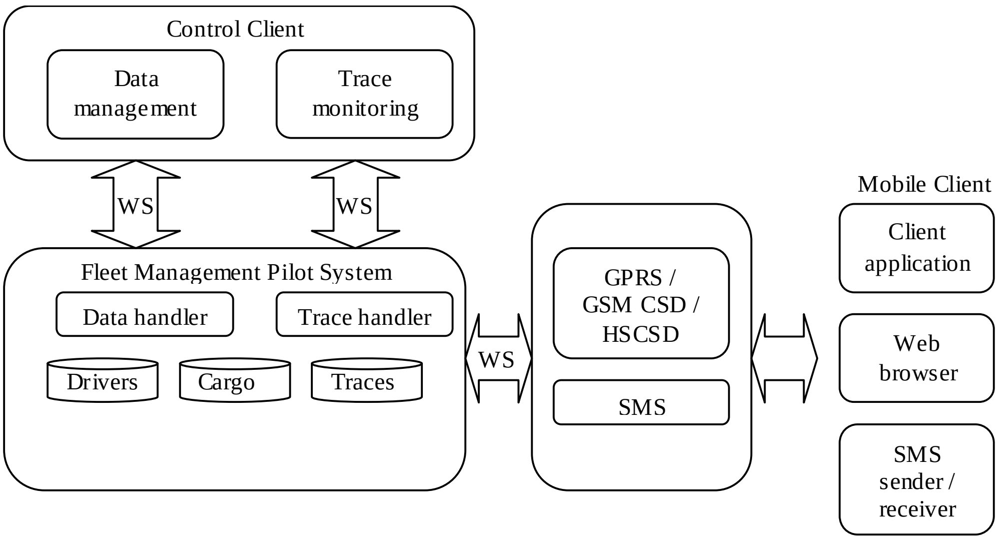


Figure 8. Design concept of the Fleet Management pilot.

4.3 Implementation

The pilot was completed in two phases: 1) the preliminary implementation was based on direct access to the platform resources such as database management and communications, 2) the enhanced implementation was based on the open web service architecture on the SOMAF platform.

In general, all software was implemented using Java. In specific, the mobile application was implemented using MIDP 2.0, where the selection was based on the questionnaires about the types of mobile phones that the truck drivers are using. The questionnaire results confirmed that the generic software platform could be exploited for building smart solutions for the mobile phones potentially used by the truck drivers.

Three business services were created on top of the SOMAF platform. Figure 9 shows the implementation architecture of fleet management pilot. Driver management service is used to manage information about the drivers. It offers Web service operations for creating, deleting, updating, and searching driver information. Load management service provides operations for managing loads information. It can be used to create, update, delete, and search load information. Transportation statuses are managed using status management service. Status management service offers also operations for listing statuses, and updating and deleting status information. All the business services exploit database service of the SOMAF platform to store information in database and retrieve information from database. Mobile management service is used to forward status information from the drivers to status management service.

The drivers send their status information using a Java ME application which is installed to their mobile phones (Figure 10). The application has pre-defined status messages from which the driver can choose the message he/she wants to send. The drivers don't have to write anything, they only select a message from menus and then send it. In the case of exceptions, the driver can give some optional descriptive information. The language of the application is English, but it would be technically possible to translate application menus into Russian using Cyrillic alphabets. Also, the drivers can give the descriptive text using Cyrillic alphabet.

Status messages are sent as SMS, because there may be fringe areas in the mobile networks that have support for IP based communication in the Russia and near the border Finland and Russia. The requirements for using the Java ME application is that the mobile phone has support for Java ME, MIDP 2.0, and Java wireless messaging API, which are supported by almost all the fairly new mobile phones in the market. Java wireless messaging API enables applications to send and receive SMS messages.

Incoming status messages from the truck drivers are forwarded from SMS receiver to mobile management service. Status information is forwarded to status management service, which stores the information to the database. The service offers operations for retrieving this information, which can be the used by the enterprise. Status management service provides also interfaces for interest groups to retrieve the status information of transportations or it can be configured to send automatically the status information to the interest groups of the enterprise.

Fleet management control client was created to monitor the status information and to manage the drivers and loads data. Control client can be, for example, a standalone program or a web client. In scenario implementation it was created as web client but it could be also possible to integrate control client to enterprise's ICT systems. The users of the web client control interface are authenticated using HTTP basic authentication over SSL. Functionalities of the control client include managing (add, edit, delete) truck driver information, managing cargo data, assigning truck drivers to cargo, and monitoring transportation status (phase or problems).

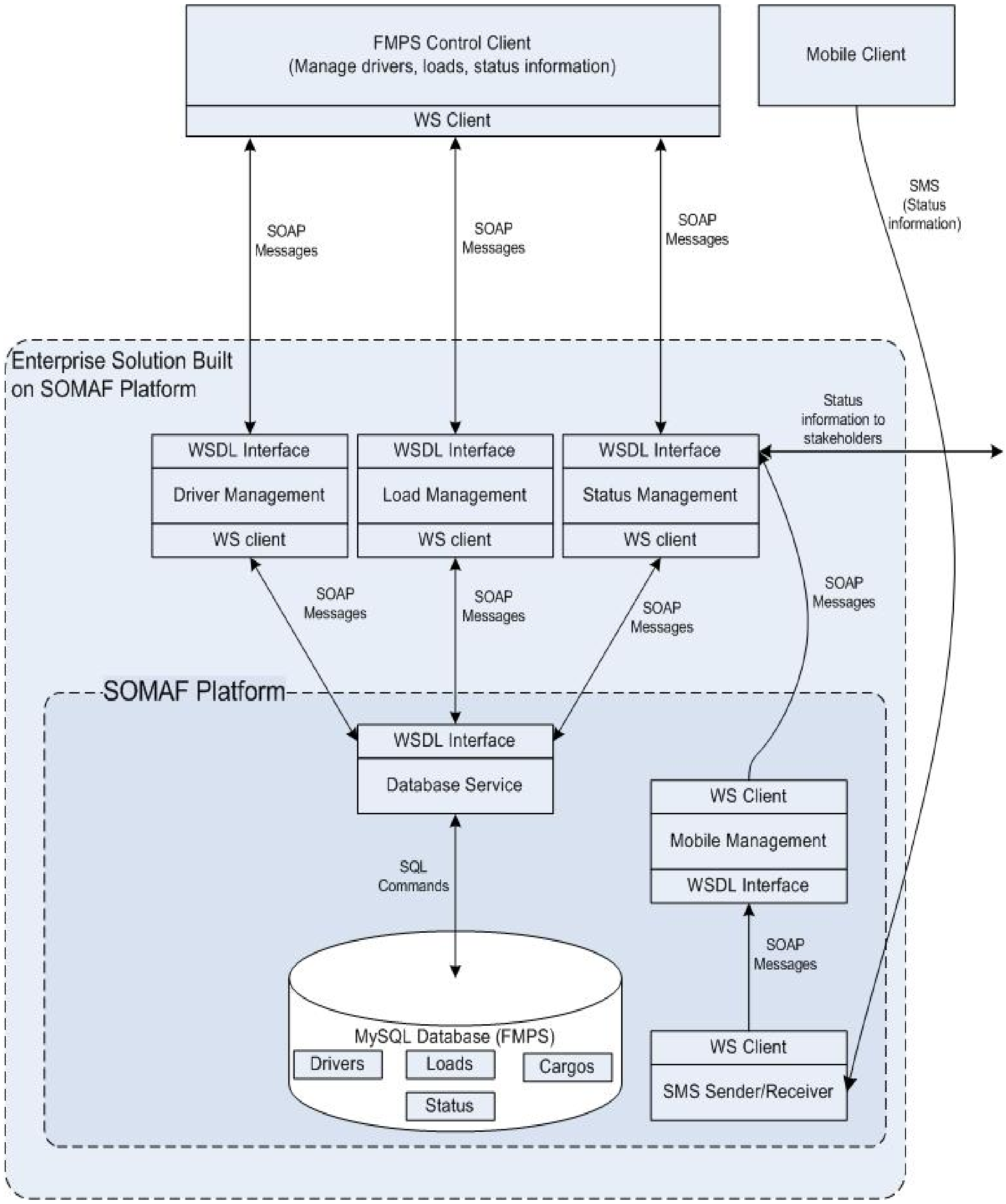


Figure 9. Implementation architecture of Fleet Management pilot.

Figure 10. Mobile phone application for giving status and exception reports.

4.4 Results

The pilot implementation provides a lightweight, simple and easy way to share transportation status information to the enterprise and its interest groups in suitable formats. The enterprise can plan their transportations more efficiently and monitoring the transports is easier than before. The scenario solution also reduces manual work needed to get statuses of transports by calling the drivers and entering status information manually to enterprise's information system.

At the moment, driver provides only status information, but it would be also technically possible to provide information about the location of the transportation. However, this would require acquiring newer and more expensive mobile devices. Nowadays the GPS (Global Positioning System) has been included in many new mobile phones. GPS could be used to provide more accurate information about time of arrival to the loading and unloading destinations. Mobile network based positioning could be used as well. Position of the transportation can be important information to clients who are waiting for delivery of their goods. More accurate arrival information allows more efficient planning of the activities in customer enterprises, e.g. resource allocation of machines and employees for unloading the cargo.

5 Trade Pilot

The Trade pilot is based on a case provided by Canon North-East Oy, who exports its parent company's office machines and consumer electronics from Finland to Russia. For B2B customers the import to Russia is handled through a Russian daughter company Canon Russia. The target for development in the pilot was enhancing the information exchange in the transport chain of the goods delivery.

Work for this pilot was started in the autumn of 2007 in the first meetings with the company representatives. The phases of information sharing and the required data in the delivery process were clarified in meetings with the case company. During the project, some of the company's central partners were also met, including a partnering Finnish-Russian haulage company in Moscow, and with them the case specification was further clarified. In early 2008, the requirement specification was completed. The first, rough concept version of the implementation of the pilot solution was introduced in April.

The development progress was somewhat slowed by partly technical, partly juridical problems concerning the delivery of example business data to the pilot system. Alternative solution was sought out, where the input data was to be received via a logistics partner of the case company. Unfortunately, this too never realized, due to lack of resources at the logistics company for building the needed integration. Regardless, development work was carried on, and near the end of the year an enhanced pilot version was completed. The pilot work was introduced in the second seminar titled *Exchange of transport information in logistics between Finland and Russia*, in St. Petersburg in 12 November 2008.

Testing of the piloted concept was planned to be organized between the participants of the supply chain, using data equivalent to real. Canon Russia (in the role of a distributor) was to be the primary partner for testing, while other expected participants were a domestic warehouse operator, and the aforementioned haulage company, which operates both as a forwarder in Finland and as a customs broker in Russia. However, due to the aforementioned difficulties, proper testing could unfortunately not be realized during the project.

5.1 Requirements

Generally, the case company aims for a direct approach to business, and to maximize performance per human resources. The company strives for efficiency and for accelerated delivery, by enhancing availability and flow of information. Additionally, the aim is for independence from individual actors, and consequently, to minimize risks. The company wants to be able to use the services offered by various actors more flexibly, in accordance to current situation.

More efficient delivery of information required for import customs procedures in Russia was named as the most important concrete need for the pilot work. This information is delivered to customs brokers; this term refers here exclusively to import hauliers operating in Russia, who offer customs declaration handling as a service. Currently, the required information is mainly delivered via email as individual documents that are not, as such, exploitable in information systems. This mode of operation requires manual processing of information, which is slow and, to some extent, prone to errors.

The most important parties in the supply chain are, in Finland, a warehouse operator, who receives, stores and packs the goods to be ready for transportation, and a forwarder, who is responsible for handling the transportation and the export procedures. In Russia, participants in the chain include the distributors of goods, and the customs brokers operating in different customs districts in Moscow. In the future, a Russian warehouse operator may also have a role in the transport process (Figure 11). In the reviewed operations model, the products are mainly sold to a daughter company operating in Moscow. This company handles the distribution of products in Russia, mainly to wholesalers and retailers (in total approx. 30 companies), and partly also directly to end customers.

At the moment, each branch of the Canon main organization has its own ERP or other information systems. Corporate level development plans exist to unify the solutions among daughter companies. Majority of the documents are transferred manually between the enterprises (e.g. using email), and entering this information to different information systems requires a large amount of work and time. The scenario solution should enable automated data transfer between the partner enterprises. The main tasks of the solution are acquiring the order data from Canon’s information system, sending information to business partners, storing the data in a database, and transforming the combined data to suitable output formats.

Because the transferred information contains confidential business information, information security has to be ensured. The information should be sent only through secured channels to prevent potential eavesdropping. Also, the information senders should be authenticated.

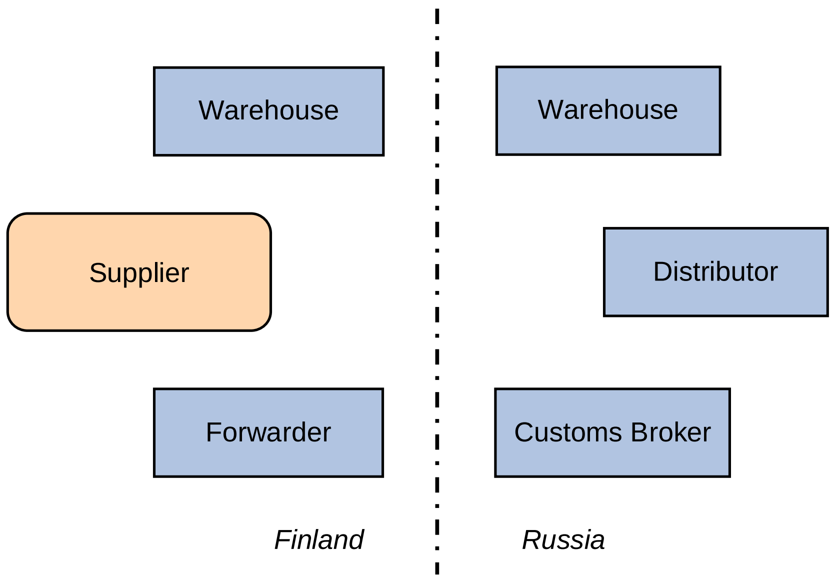


Figure 11. Participants in the collaboration forming the supply chain of the goods export from Finland to Russia. “Customs Broker” refers to a haulier in Russia handling the declaration procedures in the import process.

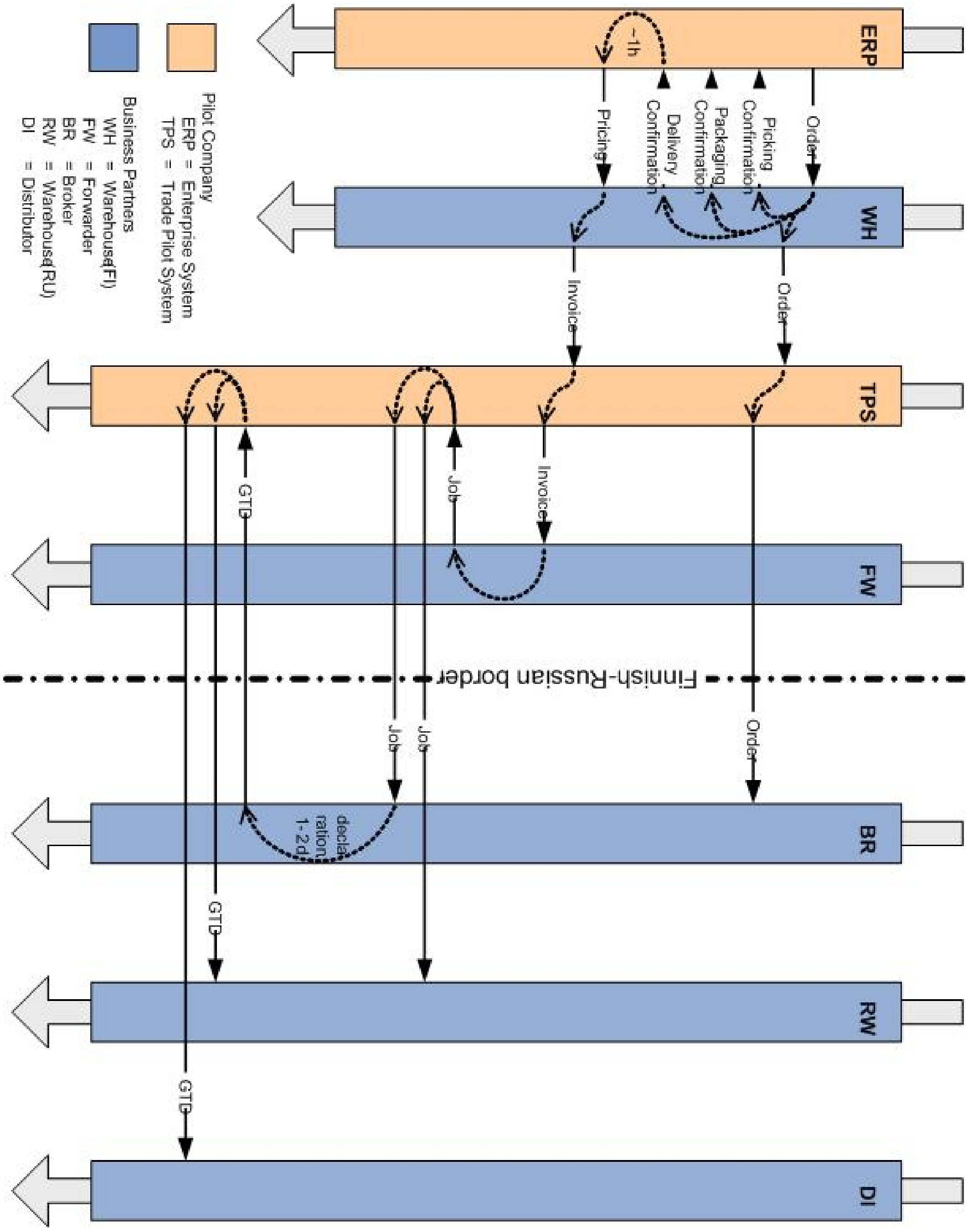


Figure 12. Information exchange process in the pilot solution. The diagram does not cover ordering or invoicing processes. The names of the data sets do not alone describe their contents precisely; for example, ”Invoice” refers here to order information that has been complemented with pricing and packaging data.

5.2 Design

An information sharing platform, acting as a data bank, was defined in the pilot to enhance flexible information exchange between the participants of supply chain in Finland and in Russia. Specifically, the pilot is designed to speed up the process of preparing import declaration documents at the Russian customs. The overall architecture of the solution is based on a system titled Trade Pilot System (TPS), which collects and stores information from the supply chain participants, and where the data is available to appropriate parties in various formats. As the information exchange proceeds, the process participants gradually complement the information set, and make it again available in the TPS. History data is collected from each phase, which helps in following the progress of the process. Additionally, optional email alerts can be configured for certain events in the process.

As is the case with all the pilots built on the SOMAF platform, the pilot solution is based on the principles of service-oriented architecture (SOA), by defining the needed functionalities as services that are available for use by higher-level services. Service-oriented design also facilitates further development of the system. The runtime platform of the system is Java, which enables setting it up in different operating system environments.

The companies participating in the supply chain may use the service interfaces by connecting to them from their own information systems, which enables full automation of data transfers between the systems. Alternatively, the companies can build separate applications with appropriate user interfaces that utilise the same interfaces. As part of the pilot work, web-based user interfaces were built for each actor. The user interfaces also allow exporting data from the system as structured XML files, facilitating further automated processing, or as PDF files, suitable for printing as documents. Figure 13 presents a view to information about a single transportation in the Forwarder’s user interface.

The requirement specification also included the possibility for some automatic data transformations, such as converting between the different customs tariff code systems used in Finland and in Russia, and between the Finnish and Russian language product names. Also, support for Russian language in the user interface and the reports produced by the system, was perceived as important. These features were not included in the implementation made during the project, but there is no effective barrier for implementing them.

5.3 Information Exchange Process

In the reviewed operations model (Figure 12) TPS receives data based on orders and invoicing from the company’s own ERP system. For practical reasons, this was organized in the pilot through the domestic warehouse operator. A separate ordering process produces order data, which covers, among other data, information about the seller, the buyer, and the place of delivery, as well as the product numbers and names and the order amounts. The information is first delivered to the warehouse operator, who complements the information set with the customs item names and dimensional data. The data is immediately delivered to TPS, which instantly forwards it to the customs broker, who can then already start preparing the import declaration.

Concurrently, picking and packing is performed at the warehouse, after which the warehouse operator receives pricing information that is based on real amounts that ended up packed for transportation. Together with the packaging data these complement the order data, which are updated to the TPS system. The aggregate

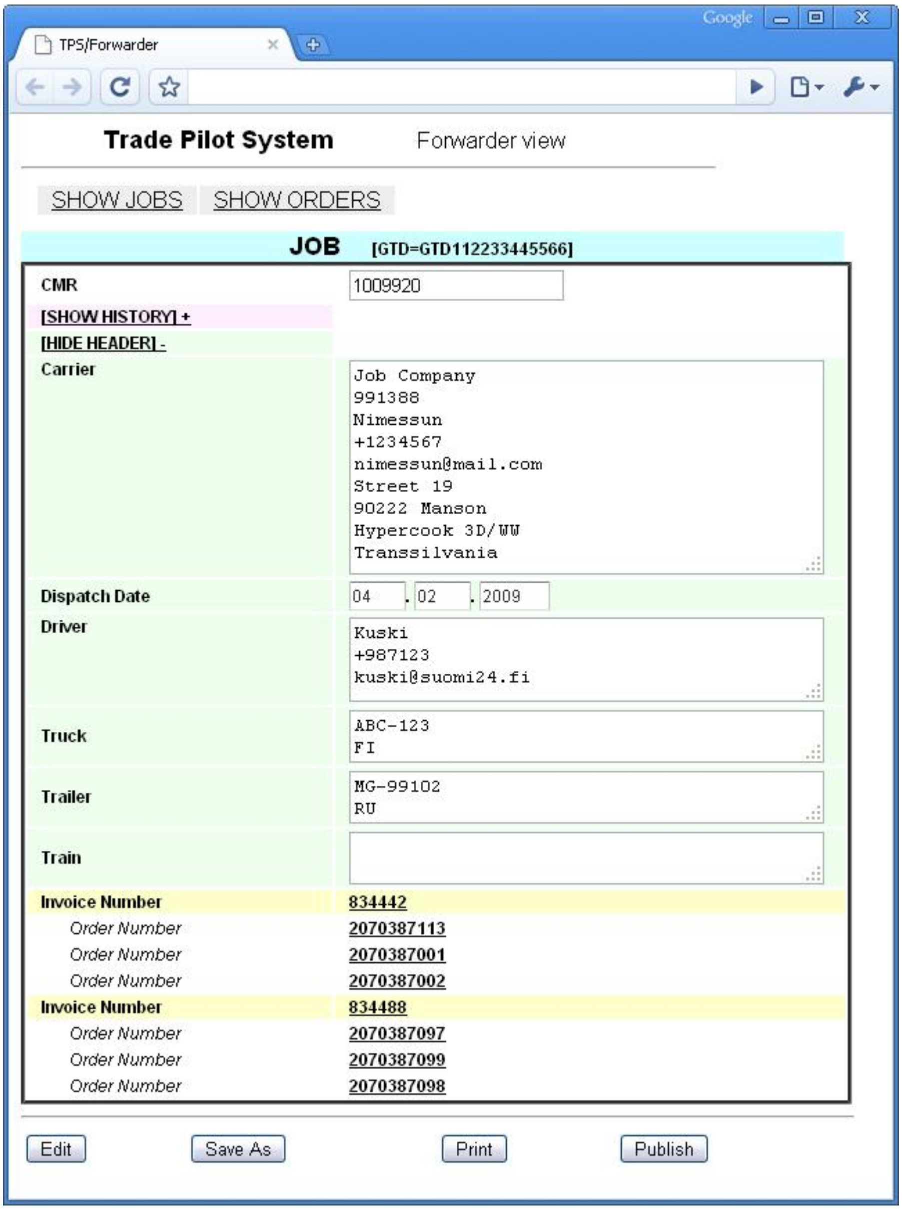


Figure 13. An example view to information about a single transportation, with pointers to detailed order data.

of information created at this point – order data complemented with pricing and packaging data – is marked in the diagram with “Invoice”.

In the next phase the information is dispatched to the forwarder, who uses the data in planning of transportations, by combining several orders to a single “Job” data set. In this phase the data set is complemented by information pertaining to a single transportation; these are the CMR number of the transportation, information about the transport company, and depending on the mode of transport, either the driver, truck and trailer data, or the train and container data.

The complete data complex is immediately forwarded to the Russian customs broker, who uses it for making the import declaration. If needed, the data can also be sent to a Russian warehouse operator. At the end of the process, the customs broker provides the GTD number, received from the declaration. This is forwarded to the warehouse operator and the Russian distributor. In the piloted model, the distributor receives information about the jobs and the orders through other channels, but also those could naturally be delivered via TPS.

5.4 Implementation

The information system interfaces for entering and fetching data were defined using the Web service technology, which allows for loosely-coupled integrations from the partners ICT systems, independent of the used implementation techniques. The interfaces are described using the Web Service Description Language (WSDL), and the data transmission is handled through the Internet using the SOAP protocol. Also, FTP can be used for file transfers. The security solution of the system is using the HTTPS protocol for encrypting the data transfers, coupled user authentication based on roles, user names and passwords.

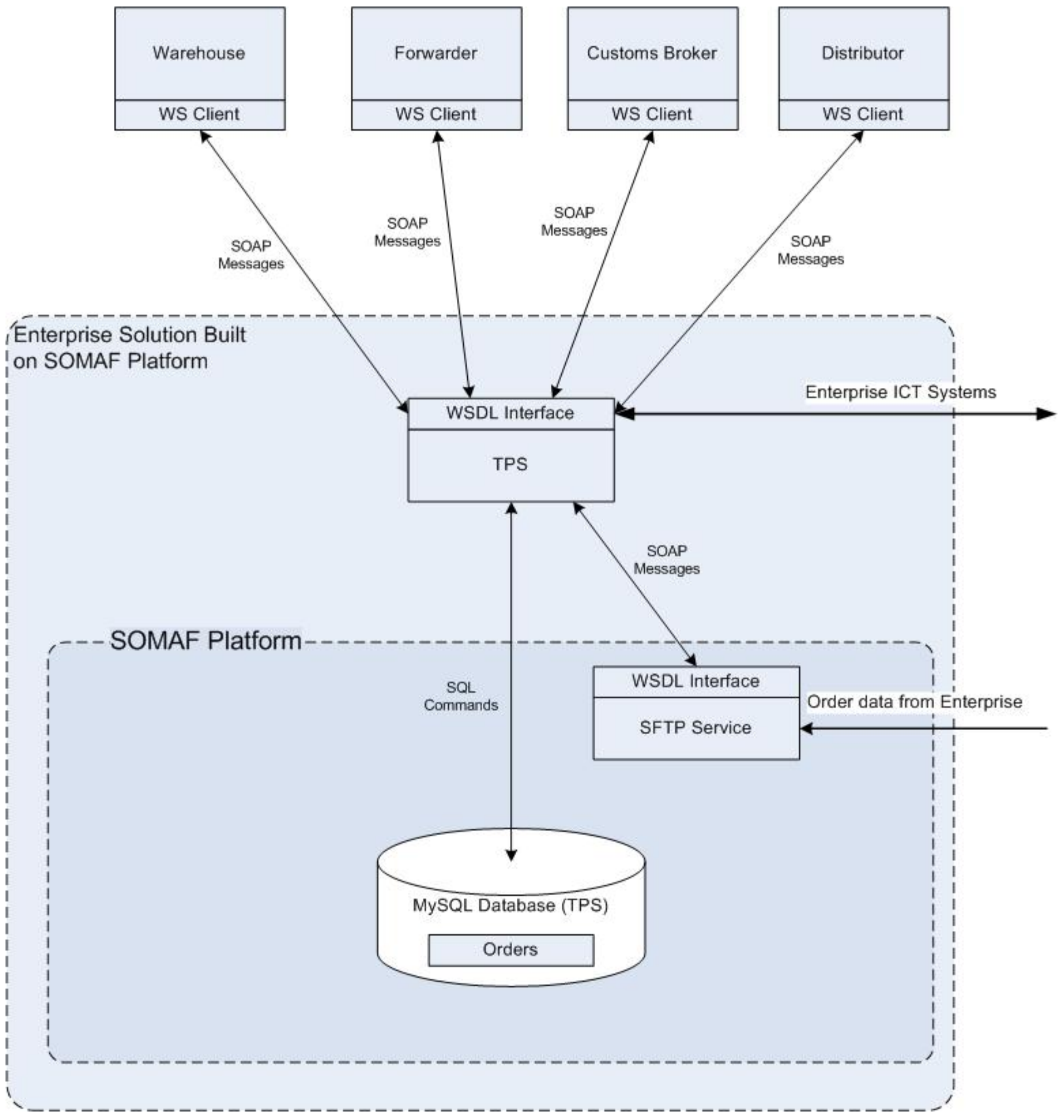


Figure 14. Implementation architecture of the Trade pilot

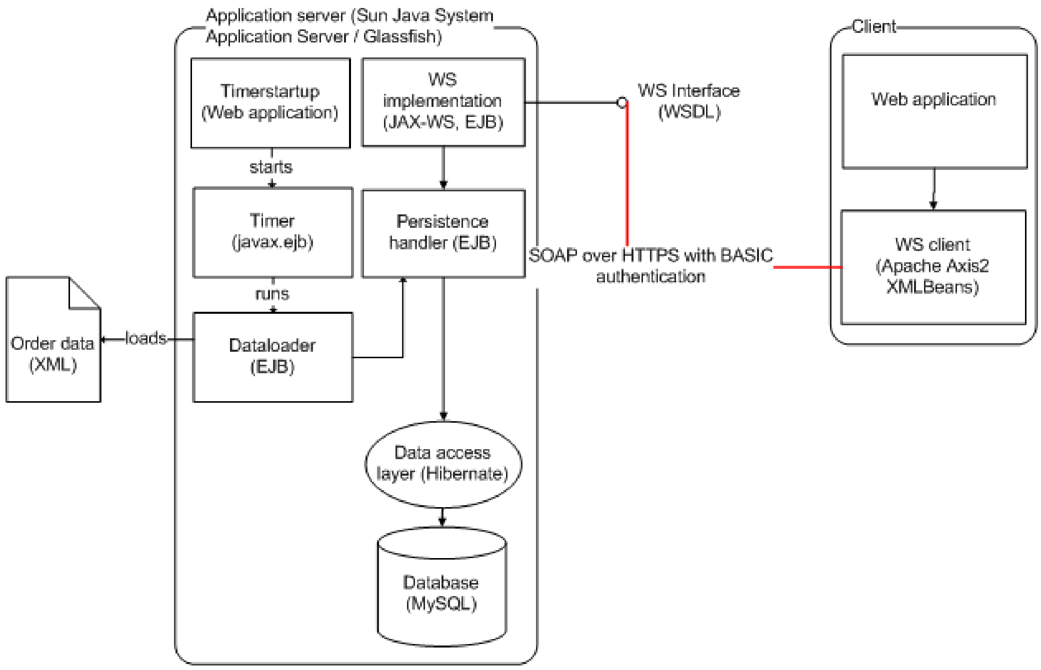


Figure 15. Implementation view (detail).

A business service named Trade Pilot System (TPS) was created on top of the SOMAF platform (Figure 14). TPS uses the SFTP service of the platform to retrieve order data from the warehouse's file server, automatically in regular intervals. The retrieved order data is stored using the database service. Other interest groups can then retrieve and complete this order data through Web service interfaces provided by the TPS. TPS also contains a specialized web application to start timers when the application server starts or when the service is deployed.

The implementation of TPS contains three enterprise Java beans (EJB): one acting as a web service endpoint, one handling the persistence and one loading data from XML files (Figure 15). The web service endpoint is implemented using the Java API for XML Web Services (JAX-WS), which automatically generates the needed WSDL documents from the bean method definitions. The endpoint uses EJB security to authorise clients based on their roles (e.g. forwarder). Every method has a defined set of roles that are authorised to run that method. An exception is thrown if someone tries to run a method without being authorised.

Persistence is achieved using MySQL as a database and Hibernate as an intermediate layer for database access. The persistence bean is implemented using Java Persistence API, which, with Hibernate, automatically generates the needed database tables from the entity classes. These also take care of the transaction details. Queries to the database are done using Hibernate Query Language (HQL), which is more abstract and object-oriented than traditional SQL.

The data loader bean is a timed bean, which on specified intervals polls defined directory for order data XML files, loads the data to the database and archives the XML file to another specified directory. XML data handling is done using Java Architecture for XML Binding (JAXB). Data loader depends on SOMAF platform to handle data loading from enterprise's information system using SFTP.

User management is based on roles defined for users and it is integrated to the admin view of the TPS. The incoming Web service calls to TPS are processed by the Glassfish application server. If the provided username is found from the user database, and the password matches the stored password in database, access to TPS is allowed. Allowed operations are determined by the role assigned to the user. The security architecture is presented in Figure 16. The web user interface for administrators exploits AAS (Authentication and Authorization Service) to store usernames and their passwords, with passwords hashed using SHA.

Database Security Service (DBSS) is used to store information about additional database permissions, i.e. it is possible to define that user is allowed to perform queries only from a certain table of a database. For the incoming order or job queries, TPS checks the query rights of the user from the user database. For example, forwarders can see only the orders assigned to them.

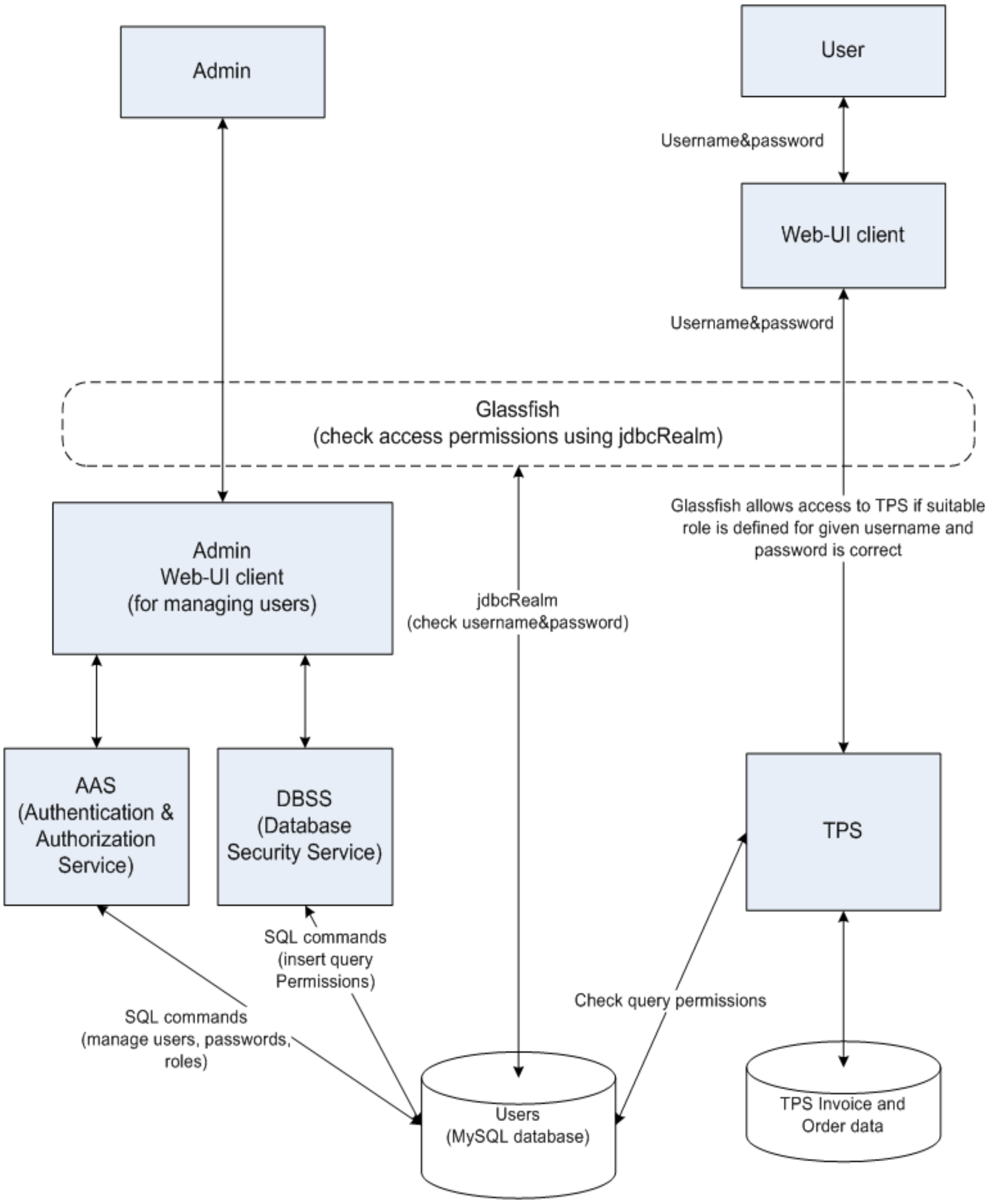


Figure 16. Security architecture in the Trade pilot.

5.5 Results

Several measurable advantages can be reached through implementing the piloted solution. The most essential benefits, applying to all participants of the described supply chain, originate from the electronic transfer of order and transportation data. Automatic integration of the data to enterprise systems leads to enhanced efficiency of work and minimization of errors. Earlier advance notice of transported or arriving goods gives advantages in resource management and planning. Consequently, better service level can be offered to customers.

For the case company, additional benefits can be seen in enhanced efficiency of the whole supply chain, including faster planning of transportations and accelerated customs declarations. Also, the ability to monitor the orders and transportations supports flexibility in business decisions and better timing of invoicing.

6 Conclusions

The project setup was coordinated alongside the FINRUS2 project, which was also the reason for having all the pilot cases related to Finno-Russian business. These cases were very interesting, and although similar in being related to transport logistics, they considered sufficiently different problems to easily warrant separate pilots. This was also extremely useful in architecting and validating the usefulness of the common SOMAF platform in supporting varying business cases.

The project was executed along several parallel development paths: one for the platform, and one for each of the pilots, albeit not all of them concurrently. The individual schedules of the case companies, and the status of their own focusing processes, were the main factors in deciding which pilots to proceed and when. Naturally, in the long term the aim was for equal splitting of effort between the pilots. Several hindrances slowed down the work at various points during the project. Some of them could be expected, others came as big surprises. At VTT, there was unforeseen loss of personnel assigned for the project. Also, reorganization of VTT operations in St. Petersburg at the end of 2007 complicated the plans to use their consultant services for clarifying technical matters directly concerning Russia. These problems were successfully compensated for, by new resourcing and recruitment.

As for the pilots, reaching and fixing focused specifications took longer than expected. Also, as was well known, today's business is dynamic by nature, and this frequently led to changes and refocusing of plans. There were also external drivers for changes; this was especially the case with the customs pilot, where the forthcoming EU and Finnish regulations were still a set of live documents during the project. In the Trade pilot, juridical problems in arranging test data came unexpected, while the sought alternative data source did not realize either due to resourcing problems. Additionally, one case company even experienced a total reorganization of personnel in the middle of the project. These problems each had their part in having the pilot implementations delayed, leaving little time for organizing other activities. Consequently, proper testing of the pilot solutions in real work environments could regrettably not be organized.

In conclusion, any slight problems aside, each of the pilots were successful in producing a working demonstration of a possible solution to the given problem. The SOMAF platform was seen successful in supporting implementation of the pilots. Based on the pilot experiences, the platform can be effectively used to accelerate and automate the information flow in various business setups. The ability to introduce new functionalities as services helped promote reuse of the implementations.

As for the future, all of the pilots, as well as the SOMAF platform itself, would naturally benefit from further work. Valuable insight on the required development would be gained by organizing more extensive pilot testing. Promoting broader application of the results in different industries would be advantageous. Continued development of the SOMAF Platform, by extending it with additional services, would further enhance its ability to support various business cases.

In the Trade pilot, electronic transfer of order and transportation data allows automatic integration to enterprise systems, leading to enhanced efficiency of work and minimization of errors. Consequently, more efficient resource management and planning can be achieved, and better service level offered to customers. The ability to monitor the orders and transportations also supports flexibility in business decisions. Several elements in the defined pilot solution can be identified as being applicable more generally as well. Also, while specifying own data model was seen appropriate during this pilot work, broader concept application could benefit from taking use of, for example, the Universal Business Language (UBL).

The forthcoming requirement for electronic declarations to EU customs is bound to prove troublesome for a large group of SME's, working in the field of border-crossing transportation. New EU initiative is in preparation, delaying the introduction of the regulations until the start of 2011. This would come as positive news for many companies, who could use the extra time for adapting their internal operations to the coming requirements. With the methods piloted in this project, information collected for the purpose of advance notification can also be directly utilised in enterprise resource planning, including management of warehouses and transportations. In this context, it is possible to achieve significant gains, for example through more efficient resourcing.

The mobile solution developed for the Fleet Management pilot in this project, was seen as simple, lightweight but effective. Enterprises can plan their transportations more efficiently, and monitoring the transports is easier, requiring less manual work. In situations where availability of modern mobile devices on board the trucks can be assumed, it would also be technically possible to provide information about the geographical location of the transportation. Either satellite or mobile network based positioning could be used to provide more accurate information about time of arrival to the loading and unloading destinations. This information allows more efficient planning of the activities in customer enterprises, especially in resource allocation.

References

[Axis2] Apache Software Foundation, 2008. Apache Axis2/Java. URL: http://ws.apache.org/axis2/ (ref. Oct 2008)

[EC06] European Concil, 2006. Commission Regulation (EC) No 1875/2006 of 18 December 2006 amending Regulation (EEC) No 2454/93 laying down provisions for the implementation of Council Regulation (EEC) No 2913/92 establishing the Community Customs Code. URL: http://eur-lex.europa.eu/smartapi/cgi/sga_doc?smartapi!celexplus!prod!DocNumber&lg=en&type_doc=Regulation&an_doc=2006&nu_doc=1875

[Fielding00] Fielding, R.T., 2000. *Architectural Styles and the Design of Network-based Software Architectures*. University of California, Irvine. Dissertation, Doctorate of Philosophy in Information and Computer Science. URL: http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm (ref: Sep 2008)

[MIDP2] Motorola, Inc. and Sun Microsystems, Inc., 2002. JSR 118: Mobile Information Device Profile 2.0. URL: http://jcp.org/en/jsr/detail?id=118

[Pautasso08] Pautasso, C., Zimmermann, O., Leymann, F., 2008. RESTful Web Services vs. "Big" Web Services: Making the Right Architectural Decision. *Proceeding of the 17th international conference on World Wide Web. April 2008.*

[Singh05] Singh Munindar P., Huhns Michael N., 2005. *Service-Oriented Computing: Semantics, Processes, Agents*. John Wiley & Sons LTD.

[Sosnoski07] Sosnoski, Dennis, 2007. Java Web Services, Part 3: Axis2 Data Binding. URL: http://www.ibm.com/developerworks/webservices/library/ws-java3/ (ref. Sep 2008)

[Sosnoski08] Sosnoski, Dennis, 2008. JiBX. URL: http://jibx.sourceforge.net/ (ref. Oct 2008)

[Toivanen08] Toivanen, Timo, 2008. *Developing a Service-Oriented Platform for Information Sharing*. Master's Thesis. Lappeenranta University of Technology, Faculty of Information Management, Department of Information Technology. URL: https://oa.doria.fi/handle/10024/42684

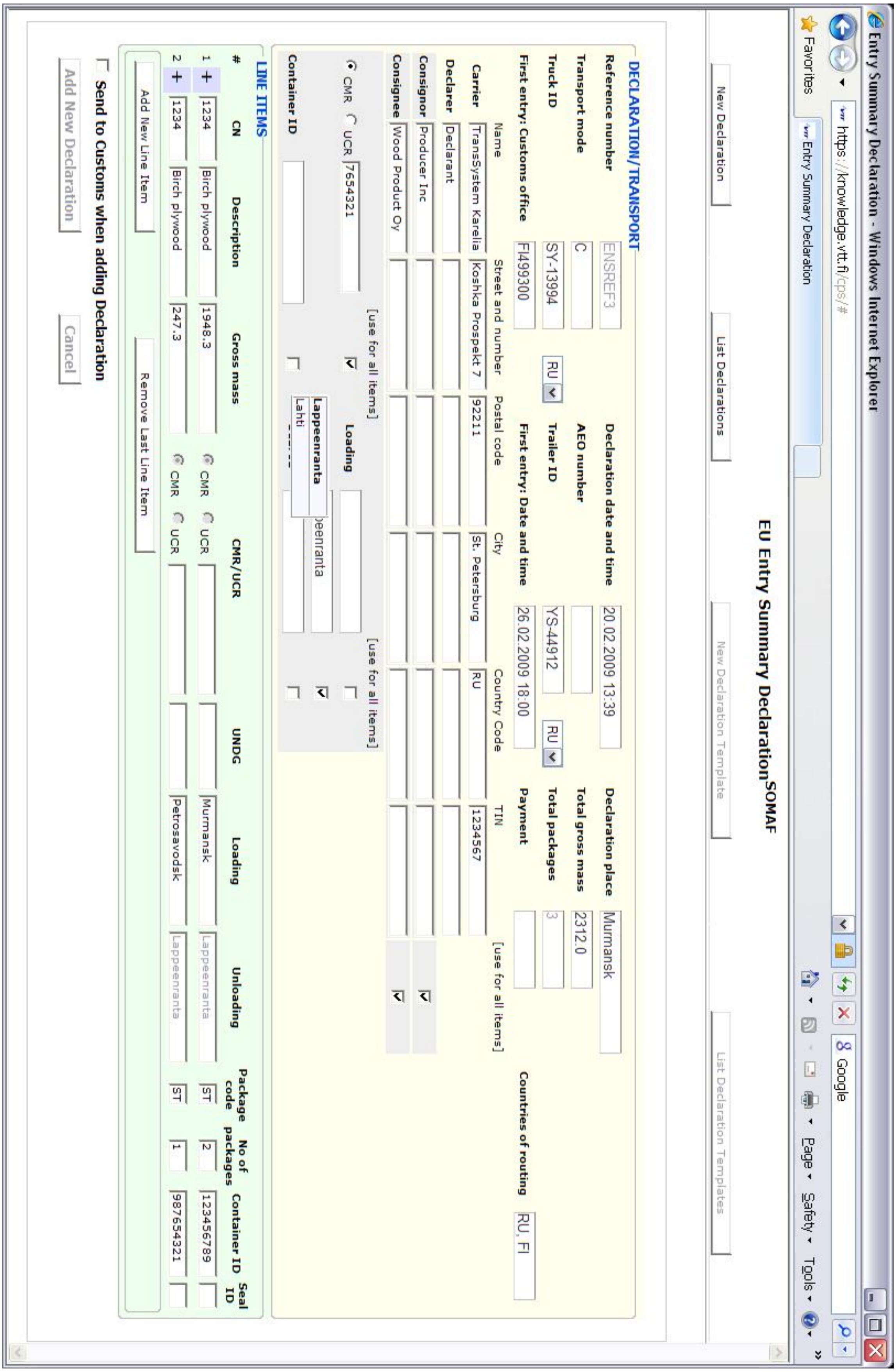
[Tulli08] Tulli, 2008. Turvatiedot pakollisiksi 2009. URL: http://www.tulli.fi/fi/06_Sahkoinen_asiointi/00_Sahkoinen_yleisilmoitus/ (ref. Oct 2008)

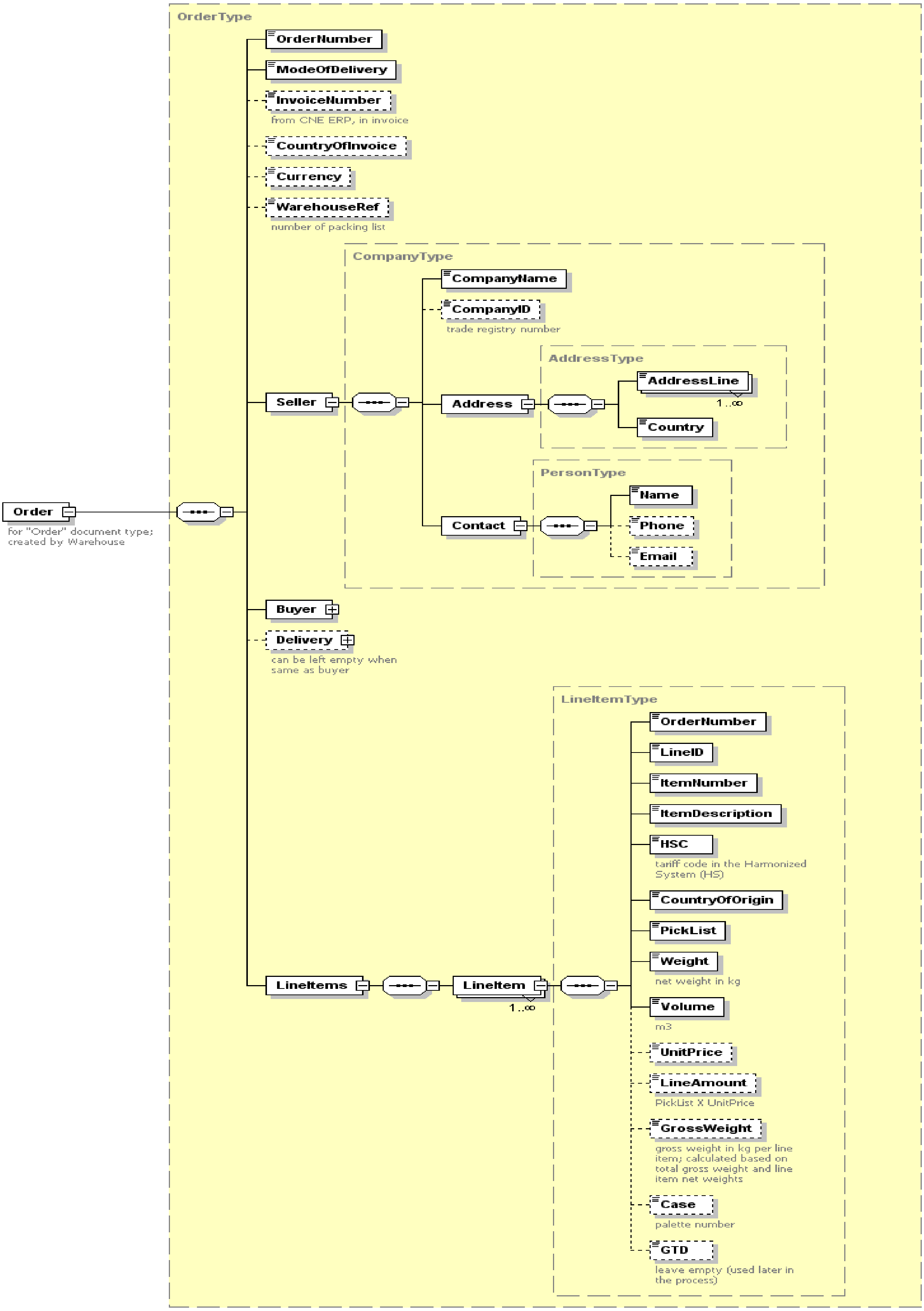
[WS-IBP] Web Services Interoperability Organization (WS-I), 2004. Basic Profile Version 1.0. URL: http://www.ws-i.org/Profiles/BasicProfile-1.0.html (ref. Sep 2008)

[XMLBeans] Apache Software Foundation, 2008. Apache XMLBeans. URL: http://xmlbeans.apache.org/ (ref. Oct 2008)

Appendices

1. Enhanced user interface for the EU Entry Summary Declaration pilot
2. “Order” data model used in the Trade pilot
3. WSDL interface of the Mobile Management service of SOMAF Platform


Entry Summary Declaration - Windows Internet Explorer
https://knowledge.vtt.fi/cps/#
Entry Summary Declaration
EU Entry Summary Declaration
SOMAF
New Declaration
List Declarations
New Declaration Template
List Declaration Templates
DECLARATION/TRANSPORT
Reference number
ENSREF3
Declaration date and time
20.02.2009 13:39
Declaration place
Murmansk
Transport mode
C
AEO number
Total gross mass
2312.0
Truck ID
SY-13994
Trailer ID
YS-44912
Total packages
3
First entry: Customs office
FI499300
First entry: Date and time
26.02.2009 18:00
Payment
Countries of routing
RU, FI
Name
Street and number
Postal code
City
Country Code
TIN
Carrier
TransSystem Karelia
Koshka Prospekt 7
92211
St. Petersburg
RU
1234567
Declarer
Declarant
Consignor
Producer Inc
Consignee
Wood Product Oy
[use for all items]
CMR
UCR
7654321
Loading
Lappeenranta
Lahti
Container ID
LINE ITEMS
#
CN
Description
Gross mass
CMR/UCR
UNDG
Loading
Unloading
Package code
No of packages
Container ID
Seal ID
1
1234
Birch plywood
1948.3
Murmansk
Lappeenranta
ST
2
123456789
2
1234
Birch plywood
247.3
Petrosavodsk
Lappeenranta
ST
1
987654321
Add New Line Item
Remove Last Line Item
Send to Customs when adding Declaration
Add New Declaration
Cancel


OrderType
OrderNumber
ModeOfDelivery
InvoiceNumber
from CNE ERP, in invoice
CountryOfInvoice
Currency
WarehouseRef
number of packing list
CompanyType
CompanyName
CompanyID
trade registry number
AddressType
AddressLine
1..∞
Country
Seller
Address
PersonType
Name
Phone
Email
Contact
Order
for "Order" document type; created by Warehouse
Buyer
Delivery
can be left empty when same as buyer
LineItemType
OrderNumber
LineID
ItemNumber
ItemDescription
HSC
tariff code in the Harmonized System (HS)
CountryOfOrigin
PickList
Weight
net weight in kg
LineItems
LineItem
1..∞
Volume
m3
UnitPrice
LineAmount
PickList X UnitPrice
GrossWeight
gross weight in kg per line item; calculated based on total gross weight and line item net weights
Case
palette number
GTD
leave empty (used later in the process)

```
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="https://vtt.somaf/MobileManagement" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsd1="https://vtt.somaf/MobileManagement" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
targetNamespace="https://vtt.somaf/MobileManagement" name="MobileManagement">
    &lt;types&gt;
        <schema targetNamespace="https://vtt.somaf/MobileManagement“
        xmlns="http://www.w3.org/2001/XMLSchema" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
        elementFormDefault="qualified">
            &lt;xsd:element name="sendSMS"&gt;
                &lt;xsd:complexType&gt;
                    &lt;xsd:sequence&gt;
                        &lt;xsd:element name="phoneNumber" type="xsd:string"/&gt;
                        &lt;xsd:element name="message" type="xsd:string"/&gt;
                    &lt;/xsd:sequence&gt;
                &lt;/xsd:complexType&gt;
            &lt;/xsd:element&gt;
            &lt;xsd:element name="sendResponse"&gt;
                &lt;xsd:complexType&gt;
                    &lt;xsd:sequence&gt;
                        &lt;xsd:element name="return" type="xsd:string"/&gt;
                    &lt;/xsd:sequence&gt;
                &lt;/xsd:complexType&gt;
            &lt;/xsd:element&gt;
            &lt;xsd:element name="receiveSMS"&gt;
                &lt;xsd:complexType&gt;
                    &lt;xsd:sequence&gt;
                        &lt;xsd:element name="phoneNumber" type="xsd:string"/&gt;
                        &lt;xsd:element name="mobileID" type="xsd:string"/&gt;
                        &lt;xsd:element name="message" type="xsd:string"/&gt;
                    &lt;/xsd:sequence&gt;
                &lt;/xsd:complexType&gt;
            &lt;/xsd:element&gt;
            &lt;xsd:element name="receiveResponse"&gt;
                &lt;xsd:complexType&gt;
                    &lt;xsd:sequence&gt;
                        &lt;xsd:element name="return" type="xsd:string"/&gt;
                    &lt;/xsd:sequence&gt;
                &lt;/xsd:complexType&gt;
            &lt;/xsd:element&gt;
            &lt;xsd:element name="addDevice"&gt;
                &lt;xsd:complexType&gt;
                    &lt;xsd:sequence&gt;
                        &lt;xsd:element name="owner" type="xsd:string"/&gt;
                        &lt;xsd:element name="phonenumber" type="xsd:string"/&gt;
                        &lt;xsd:element name="scenario" type="xsd:string"/&gt;
                    &lt;/xsd:sequence&gt;
                &lt;/xsd:complexType&gt;
            &lt;/xsd:element&gt;
            &lt;xsd:element name="addDeviceResponse"&gt;
                &lt;xsd:complexType&gt;
                    &lt;xsd:sequence&gt;
                        &lt;xsd:element name="return" type="xsd:string"/&gt;
                    &lt;/xsd:sequence&gt;
                &lt;/xsd:complexType&gt;
            &lt;/xsd:element&gt;
            &lt;xsd:element name="updateDevice"&gt;
                &lt;xsd:complexType&gt;
                    &lt;xsd:sequence&gt;
                        &lt;xsd:element name="mobileid" type="xsd:string"/&gt;
                        &lt;xsd:element name="owner" type="xsd:string"/&gt;
                        &lt;xsd:element name="phonenumber" type="xsd:string"/&gt;
                        &lt;xsd:element name="scenario" type="xsd:string"/&gt;
                    &lt;/xsd:sequence&gt;
                &lt;/xsd:complexType&gt;
            &lt;/xsd:element&gt;
            &lt;xsd:element name="updateDeviceResponse"&gt;
                &lt;xsd:complexType&gt;
```

```
                <xsd:sequence>
                    <xsd:element name="return" type="xsd:string"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name="deleteDevice">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name="phonenumber" type="xsd:string"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name="deleteDeviceResponse">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name="return" type="xsd:string"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name="listDevices">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name="list" type="xsd:string"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name="listDevicesResponse">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name="device" type="tns:deviceType" maxOccurs="unbounded"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name="searchDevice">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name="phoneNumber" type="xsd:string"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name="searchDeviceResponse">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name="device" type="tns:deviceType" minOccurs="0" maxOccurs="1"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
        <xsd:complexType name="deviceType">
            <xsd:sequence>
                <xsd:element name="mobileid" type="xsd:string"/>
                <xsd:element name="owner" type="xsd:string"/>
                <xsd:element name="phonenumber" type="xsd:string"/>
                <xsd:element name="scenario" type="xsd:string"/>
            </xsd:sequence>
        </xsd:complexType>
    </schema>
</types>
<message name="sendSMS">
    <part name="body" element="xsd1:sendSMS"/>
</message>
<message name="sendResponse">
    <part name="result" element="xsd1:sendResponse"/>
</message>
<message name="receiveSMS">
    <part name="body" element="xsd1:receiveSMS"/>
</message>
<message name="receiveResponse">
    <part name="result" element="xsd1:receiveResponse"/>
</message>
```

```
<message name="addDevice">
    <part name="body" element="xsd1:addDevice"/>
</message>
<message name="addDeviceResponse">
    <part name="result" element="xsd1:addDeviceResponse"/>
</message>
<message name="updateDevice">
    <part name="body" element="xsd1:updateDevice"/>
</message>
<message name="updateDeviceResponse">
    <part name="result" element="xsd1:updateDeviceResponse"/>
</message>
<message name="deleteDevice">
    <part name="body" element="xsd1:deleteDevice"/>
</message>
<message name="deleteDeviceResponse">
    <part name="result" element="xsd1:deleteDeviceResponse"/>
</message>
<message name="listDevices">
    <part name="body" element="xsd1:listDevices"/>
</message>
<message name="listDevicesResponse">
    <part name="result" element="xsd1:listDevicesResponse"/>
</message>
<message name="searchDevice">
    <part name="body" element="xsd1:searchDevice"/>
</message>
<message name="searchDeviceResponse">
    <part name="result" element="xsd1:searchDeviceResponse"/>
</message>
<portType name="MobileManagementPortType">
    <operation name="sendSMS">
        <input name="sendSMS" message="tns:sendSMS"/>
        <output name="sendResponse" message="tns:sendResponse"/>
    </operation>
    <operation name="receiveSMS">
        <input name="receiveSMS" message="tns:receiveSMS"/>
        <output name="receiveResponse" message="tns:receiveResponse"/>
    </operation>
    <operation name="addDevice">
        <input name="addDevice" message="tns:addDevice"/>
        <output name="addDeviceResponse" message="tns:addDeviceResponse"/>
    </operation>
    <operation name="deleteDevice">
        <input name="deleteDevice" message="tns:deleteDevice"/>
        <output name="deleteDeviceResponse" message="tns:deleteDeviceResponse"/>
    </operation>
    <operation name="searchDevice">
        <input name="searchDevice" message="tns:searchDevice"/>
        <output name="searchDeviceResponse" message="tns:searchDeviceResponse"/>
    </operation>
</portType>
<binding name="MobileManagementBinding" type="tns:MobileManagementPortType">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="sendSMS">
        <soap:operation soapAction="sendSMS" style="document"/>
        <input>
            <soap:body use="literal"/>
        </input>
        <output>
            <soap:body use="literal"/>
        </output>
    </operation>
    <operation name="receiveSMS">
        <soap:operation soapAction="receiveSMS" style="document"/>
        <input>
            <soap:body use="literal"/>
        </input>
        <output>
```

```
                <soap:body use="literal"/>
            </output>
        </operation>
        <operation name="addDevice">
            <soap:operation soapAction="addDevice" style="document"/>
            <input>
                <soap:body use="literal"/>
            </input>
            <output>
                <soap:body use="literal"/>
            </output>
        </operation>
        <operation name="deleteDevice">
            <soap:operation soapAction="deleteDevice" style="document"/>
            <input>
                <soap:body use="literal"/>
            </input>
            <output>
                <soap:body use="literal"/>
            </output>
        </operation>
        <operation name="searchDevice">
            <soap:operation soapAction="searchDevice" style="document"/>
            <input>
                <soap:body use="literal"/>
            </input>
            <output>
                <soap:body use="literal"/>
            </output>
        </operation>
    </binding>
    <service name="MobileManagementService">
        <port name="MobileManagementPort" binding="tns:MobileManagementBinding">
            <soap:address location="https://localhost:8181/axis2/services/MobileManagement"/>
        </port>
    </service>
</definitions>
```