



Project Number: **215219**
 Project Acronym: **SOA4All**
 Project Title: **Service Oriented Architectures for All**
 Instrument: **Integrated Project**
 Thematic Priority: **Information and Communication Technologies**

D9.1.1 Future C2C eCommerce Requirements and Scenario Descriptions

Activity N:	Activity 3 - Use Case Activities	
Work Package:	WP9 - C2C eCommerce	
Due Date:	M6	
Submission Date:	01/09/2008 Resubmission: 09/03/2009	
Start Date of Project:	01/03/2008	
Duration of Project:	36 Months	
Organisation Responsible of Deliverable:	Hanival	
Revision:	2.0	
Author(s):	Bernhard Schreder	HANIVAL
	Matteo Villa	TXT
	Sven Abels	TIE
	Michal Zaremba	SEEKDA

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)		
Dissemination Level		
CO	Confidential, only for members of the consortium (including the Commission)	X

Version History

Version	Date	Comments, Changes, Status	Authors, contributors, reviewers
0.1	05/2008	ToC, contents based on D0.2	All
0.2	07/2008	Input from TIE scenario	Sven Abels (TIE)
0.3	08/2008	Updates to TXT scenario	Matteo Villa (TXT)
0.4	08/2008	Updates to Hanival scenario	Bernhard Schreder (Hanival)
0.5	08/2008	Updates to seekda section, Updates to TXT section	Michal Zaremba (seekda), Matteo Villa (TXT)
0.6	08/2008	Updates to TIE section and section 3	Sven Abels (TIE)
0.7	08/2008	Updates to section 2 and exec summary	Bernhard Schreder (Hanival)
0.8	08/2008	Updates to seekda section	Michal Zaremba (seekda)
0.9	08/2008	Conclusion and final updates, final version submitted to reviewers	All
0.9.1	08/2008	Review	Juergen Vogel (SAP)
0.9.2	08/2008	Review	Jacek Kopecký (LFUI)
1.0	08/2008	Final submitted version	Bernhard Schreder (Hanival)
1.1	11/2008	New Draft for Resubmission	Bernhard Schreder (Hanival)
1.2	12/2008	Updated / Commented Version	Sven Abels (TIE)
1.3	12/2008	TXT contributions	Matteo Villa (TXT)
1.4	12/2008	Seekda's marketplace and SaaS	Michal Zaremba (seekda)
1.5	12/2008	TIE contributions and merging	Sven Abels (TIE)
1.6	12/2008	Updates to Section 5 and 6	Bernhard Schreder (Hanival)
1.7	01/2009	Updates to section 5	Matteo Villa (TXT)
1.8	01/2009	Updates to all sections	Bernhard Schreder (Hanival)
1.9.1	02/2009	Review	Juergen Vogel (SAP)

1.9.2	02/2009	Review	Jacek Kopecký (LFUI)
2.0	02/2009	Integration of review comments, Final version for resubmission	All
2.0	03/2009	Overall format and quality revision	Malena Donato (ATOS)

Table of Contents

EXECUTIVE SUMMARY	7
1. INTRODUCTION	9
1.1 INTRODUCTORY EXPLANATION OF THE DELIVERABLE	9
1.2 PURPOSE AND SCOPE	9
1.3 STRUCTURE OF THE DOCUMENT	10
2. CONTEXT AND CURRENT SITUATION	11
2.1 CHILLYDOMAINS PLATFORM DETAILS	12
2.2 SITEBUILDER DETAILS	13
3. BUSINESS SCENARIOS	16
3.1 STORYBOARD SEQUENCE OF ACTIVITIES	16
4. C2C ECOMMERCE FRAMEWORK	19
4.1 HIGH-LEVEL VIEW	19
4.2 USE OF SOA4ALL RESULTS	20
4.3 CORE FRAMEWORK	24
4.3.1 <i>Core eCommerce data model</i>	24
4.3.2 <i>User Interface for C2C eCommerce</i>	24
4.3.3 <i>Fully Integrated Sitebuilder Interface</i>	25
4.3.4 <i>Plugin Registry</i>	25
4.4 DOMAIN SPECIFIC SERVICES	25
4.4.1 <i>Simple Services</i>	26
4.4.2 <i>Collaborative Advertisement services</i>	27
4.5 SERVICE MARKETPLACE & SEARCH	29
4.5.1 <i>Seekda Service Marketplace</i>	29
4.5.2 <i>Service Discovery during Design Time</i>	31
4.5.3 <i>Service Provisioning at Runtime</i>	33
4.6 INVOLVED DATA AND USERS	34
4.6.1 <i>Data</i>	34
4.6.2 <i>Users and Roles</i>	35
5. REQUIREMENTS ANALYSIS	36
5.1 APPROACH	36
5.2 BUSINESS REQUIREMENTS	36
5.3 TECHNICAL REQUIREMENTS	37
5.3.1 <i>Frontend</i>	37
5.3.2 <i>Service Integration and Composition</i>	38
5.3.3 <i>Service Adaptation</i>	38
5.3.4 <i>Service Invocation</i>	38
5.4 REQUIREMENTS LIST	38
5.4.1 <i>Business Requirements</i>	39
5.4.2 <i>Technical Requirements</i>	40
6. CONCLUSIONS	42

List of Figures

Figure 1: chillydomains infrastructure	11
Figure 2: Sitebuilder eShop module interface	15
Figure 3: C2C eCommerce Framework Use Cases.....	18
Figure 4: C2C eCommerce Framework.....	19
Figure 5: Usage of SOA4All WP results in the WP9 scenario	20
Figure 6: Role of the Core framework in the WP9 scenario	24
Figure 7: Domain Specific Services within the eCommerce framework	25
Figure 8: BeanGarden main page	27
Figure 9: BeanGarden typical use-case.....	28
Figure 10: Service Marketplace & Search Engine in the WP9 scenario	29
Figure 11: seekda Web Services search engine (search results page).....	30
Figure 12: seekda Web Services Marketplace - Overall scenario	31
Figure 13: seekda Web Services Marketplace (Web Services Tester).....	32
Figure 14: Service bundle – support for the whole process, not only the single functionality	33
Figure 15: Data Mediation – instantiate different services and mediate data on demand.....	34

Glossary of Acronyms

Acronym	Definition
B2B	Business-to-Business
B2C	Business-to-Consumer
BPM	Business Process Management
C2C	Consumer-to-Consumer
CCBS	Customer Care and Billing System
CRM	Customer Relationship Management
D	Deliverable
DC	Digital Channel
EC	European Commission
jBPM	JBoss process engine
jPDL	jBPM Process Definition Language
QoS	Quality of Service
SOA	Service Oriented Architecture
WP	Work Package

Executive summary

This deliverable describes the results of the first task within work package 9 - the definition of one holistic and real-world oriented C2C eCommerce scenario. This document contains the description of the scenario itself as well as a collection of technical and business requirements for the described C2C eCommerce application framework. Within the scenario description, the existing service and product landscape of Hanival's ISP platform "chillydomains" is explained. The scenario is based on a storyboard involving a customer of the chillydomains webhosting platform who wants to start with a C2C eCommerce web site.

The document describes how SOA4All technology will be used to enable the various functionalities provided by the eCommerce framework and it describes the different roles that are involved in such a C2C scenario, such as service providers, service consumers, and brokers. In addition to this, the document describes the existing landscape of Hanival, which will be used as a basis for the deployment of the eCommerce framework. It also describes technologies that the other WP9 partners contribute to this framework: TXT supplies an collaborative advertisement service, TIE will provide eCommerce services based on their TIE's Kinetix/Mambofive platform, while seekda will supply their knowledge as a service broker, realising the web service search engine and marketplace functionalities with SOA4All technologies. Altogether, the eCommerce framework will thus provide prospective users with basic eCommerce functionalities and the capability of extending their solutions with third party services.

The WP9 Use Case is entirely focused on providing an easy way for end users to use third party services offered through the framework, enabling them to build eCommerce applications. To convert the vision into reality, a C2C service platform under design must be capable to provide discovery facilities and support users with the integration and composition of these services.

Potential users of such a C2C eCommerce platform, such as the customers of an ISP, will normally not search for individual services (e.g. a specific credit card approval service), but for complete solutions to address their business needs (e.g. services to provide a certain kind of payment functionality for the Web shop). That is why the platform must be capable to support very different application scenarios, while at the same time it should remain generic enough so that the same set of common components can be (re-)used to run each of such scenarios.

The platform will allow users to customize these services as per their individual requirements. The execution of (complex) services can be handled by an execution engine provided by the service broker (based on results from WP6), but services themselves might be (and usually will be) provided by third parties remaining independent and out of control from the service broker. Customers can also easily integrate these services into their own already existing environments.

In the future, we would imagine that the end customers offer various solutions (products) to other end customers via the eCommerce framework, consisting and composed of many Web Services. Some examples for possible eCommerce applications include:

- e-Hotel solution allowing to run Web based reservation of rooms in the hotel
- Web shop solution allowing to sell products over the Web,
- e-Conference solution supporting in organization of conferences (e.g. including registration service, payments service, fraud check etc.)

Customers of Hanival's C2C eCommerce framework would be capable, when buying a

typical Web hosting option at Hanival, to extend this product by additional services. End customers are able to use various SOA4All-enhanced tools offered through this framework to build their own end customer applications. While people may use the SOA4All results to build generic applications, the eCommerce framework will provide eCommerce specific functionality and will itself also use the SOA4All services for achieving this.

Main Changes:

Since the initial submission of this deliverable for M6 of the project, the document has seen extensive revision. The partners in WP9 have decided to pool their resources in order to realise a single scenario – the eCommerce framework for ISP customers. While the partners thus concentrate on one of the original three scenarios described in the previous version of this document, they still supply their specific expertise and technology to the scenario, which enables the framework to support the creation of eCommerce applications of various levels of complexity.

1. Introduction

1.1 Introductory explanation of the deliverable

This deliverable describes the results of the first task within work package 9, the definition of a concrete C2C eCommerce scenario. This document contains the description of the C2C eCommerce framework, hosted by an Internet Service Provider. The framework enables the ISP's end users to set up C2C eCommerce applications of various complexities and host these applications on their web sites. The document also describes a collection of technical and business requirements for this C2C eCommerce system.

1.2 Purpose and Scope

The document aims in giving a precise overview about WP9 and its outcomes. It will highlight

- The WP9 scenario in detail
Containing the detailed structure of the C2C eCommerce framework.
- The practical benefits of using SOA4All in this scenario
Showing how SOA4All will ease the development of C2C eCommerce applications
- The requirements for fulfilling the scenario
Summarizing the business requirements as well as the technical requirements

The document will be used as an input to all other WP9 tasks and it will be used as a basis for all future developments. It may also be used by other work packages as a reference for defining their own requirements. This reference may be of value for defining their technical interfaces, architectures and technology selections, and align them to the needs of the use case work packages.

The document aims at the following audience:

- Reviewers
To check the overall vision of WP9 and its expected results
- WP9 team members
To act as a guideline for the rest of the work package
- SOA4All project members
To help them defining their requirements and functionalities
- Service and Infrastructure Providers (such as other ISPs)
To show them how SOA4All may be used for improving their products and projects.

1.3 Structure of the document

The document is structured as follows:

Section	Content
3. CONTEXT & CURRENT SITUATION	Description of the current situation of the Internet Service Provider Hanival and describes the context of the WP9 scenario
4. BUSINESS SCENARIO	Simple C2C business scenario that can be created using the eCommerce framework
5. C2C ECOMMERCE FRAMEWORK	Description of the framework that will be created in WP9 in order to realize the business scenario. This section starts with a bird's eye view and afterwards describes the different components.
6. REQUIREMENTS ANALYSIS	Requirements that are specified by the WP9 scenario, including both business and technical requirements
7. CONCLUSIONS	The conclusions and the next steps of WP9

2. Context and Current Situation

Hanival's main line of business is the Web hosting platform chillydomains¹, a fully automated domain registrar and hosting provider. The chillydomains platform is based on the extensive research and development activities of Hanival Internet Services. To succeed in the challenging domain registration market, special attention was paid to provide user-friendly operations, automated business processing and an optimal price-performance ratio.

Figure 1 shows the infrastructure of the chillydomains Web hosting platform, including the Customer Care and Billing system (CCBS), additional systems, such as the Customer Relationship Management (CRM) and localized Web shops, as well as the collection of internal and external services used by the platform.

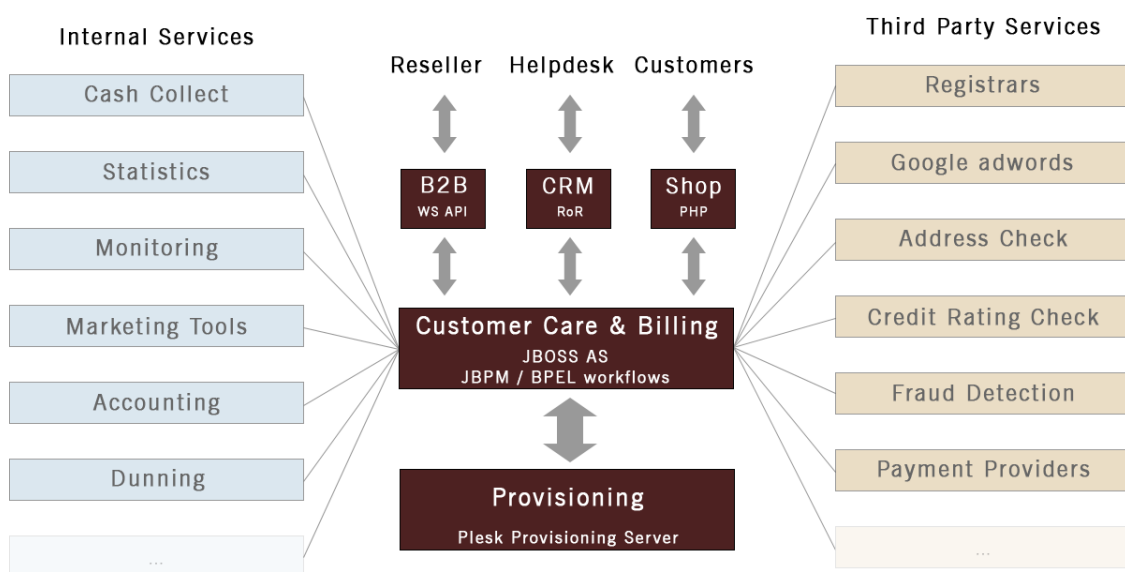


Figure 1: chillydomains infrastructure

Based on the chillydomains platform, WP9 aimed to implement a C2C eCommerce framework, which allows end customers to build their own eCommerce applications from a set of services provided either by Hanival or by third party suppliers. Ideally, a chillydomains customer orders a product bundle including a domain with a hosted Web space and then creates a Web presence using a collection of Web 2.0 applications, such as the dedicated Sitebuilder software currently in use. Integrated with these tools, the chillydomains customers can create an eCommerce application offering their own products or services.

The eCommerce application can consist of various services and components, and can differ in complexity. For example, the user can start with a simple eCommerce component added to an existing web site, or create a complete web shop solution in addition to any other web content on their site. Thus, the customers can directly sell their own products by utilizing an

¹ <http://www.chillydomains.at>

array of services (such as payment providers, fraud detection, delivery and notification services etc.). The user interface for the chillydomains customer should be intuitive, and eCommerce applications should be easy and straightforward to assemble, since we do not expect the average chillydomains customer to be able and willing to create configuration files and utilize scripting languages.

The next sections will explain the chillydomains platform and existing services in more detail.

2.1 chillydomains platform details

The Customer Care and Billing system (CCBS) used by chillydomains has been designed and implemented as a middleware system based on the SOA paradigm, since a quickly growing set of external business services from third party providers, such as payment services and domain registrars, has to be integrated with the core functionalities.

The chillydomains platform includes a wide variety of services, integrated by a Business Process Management (BPM) solution, including:

- Financial Services (Cash Clearing, Credit Card Validation and Payment, Order, etc.)
- Domain Services (Whois, Domain Registration, Domain Transfer, etc.)
- Physical Hosting Services (Webpace, Mail, Provisioning, Maintenance, etc.)
- Retail / Wholesale Services (Order, Upgrade, Trouble Ticketing, etc.)

The core of the chillydomains webhosting platform, the billing and provisioning system, has been realized as a Java EE application, and consists of several business processes (and subprocesses) executed by the jBPM² workflow and BPM engine. jBPM is designed from the ground up to support all kinds of workflow patterns. Benefits of using a dedicated business process management component, such as jBPM, inside the chillydomains system include the increased decoupling of components, necessary due to the application's dependency on communication with external business partners, asynchronous messaging and long-running processes, as well as the flexible adaptation of the states and transitions of the core processes. The workflows for ordering, provisioning and billing are described in the jBPM Process Definition Language (jPDL), a declarative language to describe service orchestrations and business processes.

In addition, as shown in Figure 1, diverse applications involved in the ordering, provisioning and customer relationship management tasks are linked to the central CCBS. The core services are available to applications running on different platforms. These applications include a dedicated CRM system and the multilingual and multinational instances of the chillydomains web shop. The core services include both internal services, deployed on the chillydomains application server and a rapidly growing collection of external services supplied by different third party providers. New services are added whenever additional payment providers are needed, or existing payment providers are exchanged. Also, new domain registration services are added when the chillydomains web shop becomes available for new countries. Currently this results in a growth of available services by about 5% per month. The middleware services realize a number of different communication channels with this set of services by third party providers, and include both synchronous and asynchronous services.

² <http://www.jboss.com/products/jbpm>

As shown in the chillydomains infrastructure figure, the commercial webhosting software Plesk³ is used for the management of the chillydomains customers' domains and webspace. Plesk consists of a control panel for domain owners, enabling them to administrate their domains, as well as to add web pages and applications to their web space. One of the applications integrated with this control panel is the Sitebuilder⁴, an application that should enable webspace owners to construct new web sites quickly, using a collection of templates, page modules and styles. Sitebuilder, respectively any kind of similar web site creation tool, offers an ideal basis for the extension with an eCommerce application framework, as described in the scenario details provided in Section 5.

With the current version of the chillydomains platform new products for end customers can be easily defined, based on a composition of the different internal and external services. Once customers start to configure and set-up their products however, they are bound by the tools and functionalities made available by the webhosting software. A small selection of predefined templates and modules is available for customers, with which they can set up their web site and create specialised pages, like blogs or picture galleries. This is generally an issue for ISPs, as most are dependent on out-of-the-box webhosting software. The only way for their customers to gain a larger degree of flexibility is to extend through their own code, which prevents a large share of potential eCommerce users from realizing their ideas.

The next section explains an example for this kind of commercial web site software, the Sitebuilder application, in more detail, in order to highlight the overall context for the eCommerce framework, and the available integration points.

2.2 Sitebuilder details

Sitebuilder is a browser-based web site creation and editing application. The chillydomains platform currently hosts this application for its end users, but most webhosting companies provide comparable applications. At the time of this writing, Hanival is involved with the development of a new template based website creation tool, which is planned to replace Sitebuilder in the mid-term. The new tool will feature extendable components, which will make the integration of other applications easier. Still, the features of the envisioned web site creation application will be similar to Sitebuilder, so a short overview of the main properties and the web site creation process is given below.

In version 4.2, Sitebuilder enables users to create their own web sites in five steps:

- **Start:** Users select the type of site they would like to create. Possible options include normal web sites, blogs and photo galleries.
- **Choose Design:** After setting the basic type of web site that a user wants to create, the design can be chosen from a selection of different templates. In addition, users can also customize the templates by selecting colours, headers, and logos.
- **Add Pages:** Users can define the site structure and hierarchy, using various page sets and modules. Thus, users can extend their web sites with different sub sites.
- **Edit Pages:** End users create content for the different pages using a WYSIWYG editor. Additional page modules can be dragged into the content pages directly.
- **Publish:** The Sitebuilder users can publish their creations to their own web space

³ <http://www.parallels.com/plesk>

⁴ <http://www.parallels.com/sitebuilder>

and activate the new site.

Currently, Sitebuilder supports a number of special page modules, which are not all used in the chillydomains webhosting platform. The following are the most important modules regarding the context of creating an eCommerce application framework:

- **Common Page:** including common text and figures, with no specific additional functionalities
- **Blog:** The blog module enables to create blog pages with all common functionalities, such as commenting, trackbacks etc.
- **Image Gallery:** The module enables users to upload images from different sources to online galleries.
- **eShop:** this module offers the possibility to create a web shop integrated with the remainder of the customer's web site (see further information below).

Figure 2 shows the administration interface for the eShop module included with Sitebuilder. The eShop owner can manage products and product categories, set prices and stock levels, and can choose from built-in payment systems and shipping methods. The functionalities offered by eShop lead to an eCommerce solution in a straightforward manner, but the possibilities for end users are severely constrained by the built-in services.

The eShop module is not currently supported by the version of Sitebuilder available to customers of the chillydomains webhosting platform. This is due to the fact, that the tool currently still has issues, and is not easily customizable – specifically with regards to the payment services offered by it. Still, the functionalities of this module provide the basic set of requirements to be followed by the envisioned eCommerce application framework, as noted in the technical requirements section below.

The following sections will more clearly define the typical business scenario for an end customer of the chillydomains ISP platform, the components of the envisioned C2C eCommerce framework and finally the needed requirements for the eCommerce application framework, both from a business and a technical perspective. Thus, the requirements are also applicable to the SOA4All components, which should provide the core functionalities of said eCommerce framework.

Here you can manage your eShop products. Already a user? Log in here... Support Help

Site map

- Home
- About Me
- My Family
- Photos
- Resume

Page title*
eShop

Page file name*
page8

Page keywords

Page description

Show this page in site map

* Required fields

eShop

Products Categories Payment Systems Shipping Methods Notifications Settings Orders

Tools

Add New Product

Products

Remove Selected

All Search [Show All](#)

2 records total Number of entries per page: [5](#) [10](#) [25](#) [100](#)

	P	Name	Categories	Price	Stock level	Edit
<input type="checkbox"/>		Product1	Category1	\$ 99.00	0	
<input type="checkbox"/>		Product2	Uncategorized (shown on front page)	\$ 49.00	0	

2 records total Number of entries per page: [5](#) [10](#) [25](#) [100](#)

Figure 2: Sitebuilder eShop module interface

3. Business Scenarios

This section details a typical business scenario for the eCommerce framework deployed on a webhosting platform. A customer of Hanival's ISP platform chillydomains has concluded that they want basic eCommerce functionalities added to their web page, as they are already actively participating in eBay for selling some small items every now and then. However, they are not comfortable with the low flexibility of this solution, and about being bound to one specific payment provider and one specific rating system, etc. This user has a good technical understanding of basic eCommerce functionalities and the creation of web pages, but they are certainly not an SOA expert.

Below, we define a sequence of activities of this potential user in setting up their own eCommerce solution. Furthermore it is described how SOA4All, and specifically the WP9 eCommerce framework, can help her to achieve her goals and to enrich their site with eCommerce functionalities based on a smart composition of Web Services.

As a prerequisite, it is assumed that all services that are used in this scenario are SOA4All "ready" meaning that they should be annotated with MicroWSMO, respectively WSMO Lite. Furthermore, this means that external services have to be available through discovery and the service broker functionalities of SOA4All. In an extended usage scenario, it should also be possible to easily integrate suitable web services without annotations, that is, services described only in WSDL. With the SOA4All Studio, users should be able to quickly annotate found services.

3.1 Storyboard sequence of activities

With the help of the WP9 eCommerce framework, users are able to simplify their efforts. They can perform the following steps to add C2C eCommerce functionalities to their website:

1. The users choose to create their own web shop, after buying a suitable product from the chillydomains ISP platform. In the Sitebuilder application, which is available to all customers of chillydomains, users see several page modules, which they can use to put together their web sites. Choosing the eCommerce module, they can select from various templates for eCommerce applications with different functionalities. The users select the "basic web shop" template, which has a number of mandatory elements a web shop should consist of, but also offers the possibility to add additional functionality through both internal and third party services.
2. The user starts with something simple: Adding shopping cart functionality. This functionality is already provided by an internal service in the eCommerce framework. The user selects this option to add the shopping cart functionality to the website with just copying and pasting some lines of code and with 5 minutes of configuration.
3. The user then decides to add payment functionality and does the same steps again. This time the eCommerce framework allows users to choose between a set of known Payment providers and the possibility to add new payment services through a service broker. As it happens, other users have used some of the services in the past and have provided recommendations and rankings. Based on this additional information, users can select from a set of payment services, which should be available to customers of their web shop.
4. Now having a shopping cart and payment services, the user wants to connect those services. This functionality is made possible by the SOA4All functionalities included in

the eCommerce framework. A graphical user interface allows the user to orchestrate and connect the 2 services. The possibilities offered to combine the services should be based on simple workflow templates offered by the tool. These templates should reflect a typical order process, and include extension points for additional activities (respectively services added to provide functionalities for these activities).

5. Finally, the user now wants to add a service for converting currencies in order to be able to make the webshop accessible for consumers of other countries. Such a service is not provided in the eCommerce framework, so the user clicks on the “search for 3rd party service” button and uses the search engine to find a currency conversion service. The search engine utilizes the SOA4All discovery components to find matching services. Again, the user can then use the composition functionalities to orchestrate all services and to connect the new service to those that have already been selected previously. Based on end user context, e.g., geographical information, the composite service would use the currency conversion service to adapt the shown prices of products.
6. Having completed the mandatory steps in the creation of the web shop application, users can then publish the shop to their web site and activates the application.
7. The webshop is visited by other end users (customers), which can buy products by adding them to a shopping cart and using a payment service. In addition, the original creator of the webshop application can use a monitoring tool provided by the eCommerce framework to analyze statistics of customers who visited the webshop, including data on service invocation. The users could then use the gathered information to adapt their eCommerce solutions, and also to rate and recommend services to other users.
8. Some months later, users might want to include advertising in their site and at the same time, want to incentivise shoppers’ participation in that site. An advertising service is the ideal solution for this. Thanks to SOA4All functionalities, the users find and integrate a “collaborative advertising” solution with their existing eCommerce application, as described in Section 4.4.2.

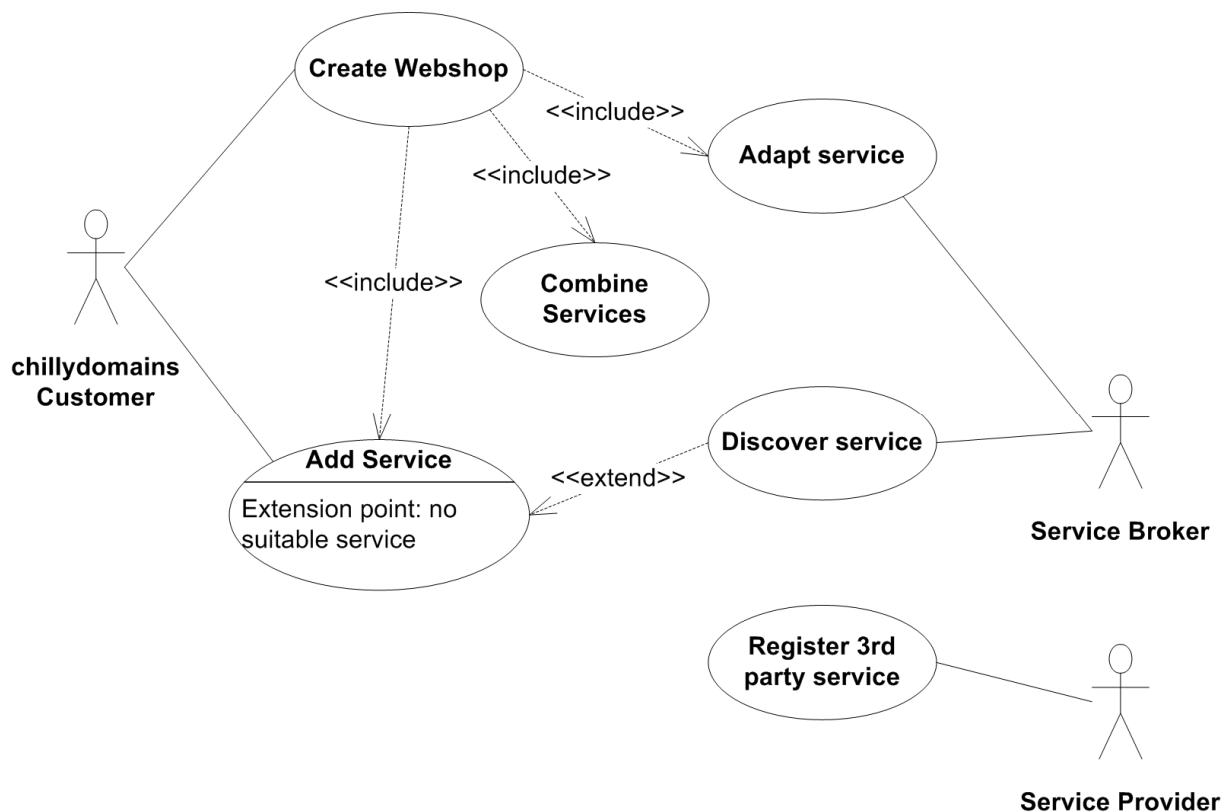


Figure 3: C2C eCommerce Framework Use Cases

In Figure 3, is depicted the major use cases and actors in this business scenario. The chillydomains customer creates a new web shop, by combining services, which are either available from the eCommerce framework itself, or added by using the capabilities of the service broker. In addition, the chillydomains customer is able to adapt the services to a certain context, which can depend on user requirements (such as the location of a given user) or to regulatory requirements (e.g., the need to add suitable credit rating checks for certain countries).

Service brokers provide the means to discover suitable services, and can offer “service bundles”, that is services which are already adapted to a specific setting and might combine several smaller services to one added-value building block. An example for such a service is a payment service from one provider combined with a credit rating check service offered by another provider (more details for this combined service are described in Section 5.5.3). Finally, service providers offer various services for inclusion in an eCommerce application, and can register those services with the service marketplace offered by the service broker. Registering a service in this manner enables other consumers to discover more easily those services, and to provide valuable feedback and ratings.

4. C2C eCommerce Framework

Users may realize a large portion of the business scenario that has been described in the last section by directly using the SOA4All services provided as a result by the different work packages. However, the SOA4All results are completely independent of a specific use case and therefore certain use case related services will be necessary in order to really apply all results in a domain specific way. This allows WP9 to add scenario specific functionality (i.e. C2C eCommerce specific functionality) and to lift the SOA4All results to a level where it is easy to use for non-technical experts coming from the C2C eCommerce domain.

4.1 High-Level View

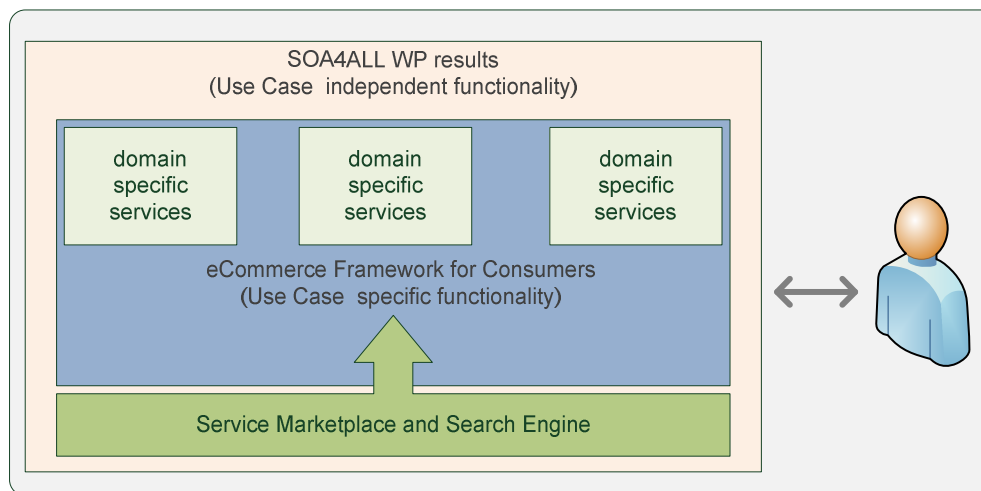


Figure 4: C2C eCommerce Framework

The WP9 Use Case scenario centers on providing end customers the means to create flexible and highly customizable C2C eCommerce websites. In order to do this, users will use a C2C eCommerce framework, which is aimed to operate on an ISP's infrastructure and is accessible to its customers. This C2C eCommerce framework will add eCommerce specific functionality such as a product catalogue, order management and invoicing and basic payment services. Other functionalities will be more generic and may be directly taken from the SOA4All results such as e.g. finding a service and composing a process. External services, available from third party providers, will also be useful to add additional functionalities to the eCommerce solutions. A dedicated service broker, offering a search engine and a service marketplace for value-added (composed) services, will provide access to these external services.

The eCommerce application framework will enable consumers to conduct business with other customers. This approach is similar to popular Web 2.0 platforms (e.g. Facebook or Google Checkout) where the company provides a framework that allows customers to create solutions for C2C business. What is different in the WP9 scenario from existing solutions is that the WP9 scenario will entirely focus on providing an easy way of using third party services (solutions) offered through the framework. This is performed by using the SOA4All results of the other work packages and will therefore involve all major parts of SOA4All.

End customers are able to use various SOA4All-enhanced tools offered through this framework to build their own end customer applications. While people may use the SOA4All results to build generic applications, the eCommerce framework will add eCommerce specific functionality and will itself also use the SOA4All services for achieving this. For example, it

will provide typical Web Shop functionalities such as a shopping cart feature and an access to payment providers using the SOA4All service orchestration and communication facilities. Figure 4 shows the overall building blocks of the WP9 use case, centred on the joint eCommerce framework.

Since “end customer eCommerce applications” is a generic description, WP9 will provide “*focus areas*” that extend the generic C2C eCommerce framework. Those focus areas are described as “domain specific services” in Figure 4. While the eCommerce framework itself can be seen as the basic framework that interacts with the ISPs infrastructure (in SOA4Alls case Hanival’s chillydomains platform), the domain specific services are concrete sets of APIs for achieving a specific functionality such as a shopping cart or a product catalogue. Other services and APIs enable the users of the framework to extend their eCommerce applications in specific directions, such as advertisement and marketing management. The following sections will describe the different parts of the framework in more detail. The different parts of the scenario will be highlighted based on Figure 4 at the beginning of each subsection.

4.2 Use of SOA4All Results

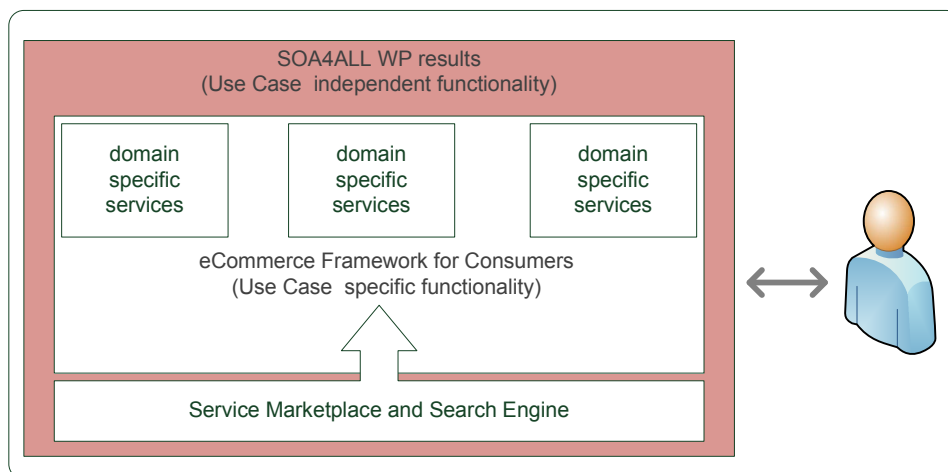


Figure 5: Usage of SOA4All WP results in the WP9 scenario

WP9 will completely rely on the SOA4All results of the different work packages. All functionality will be provided by the C2C eCommerce framework will also rely on the results although they might add some eCommerce specific logic to them.

The following work packages and concrete activities play a major role in the C2C eCommerce scenario of WP9:


















- Annotation and Reasoning (WP3): WSMO-Lite and MicroWSMO will be used for the description and annotation of third party services. In addition, a scalable reasoner is needed to work with possibly large sets of services.
- Service Context information (WP2 and WP6): services to be used in the eCommerce applications will feature different contexts, for example, often there will be several user views (e.g. using a service in a localized manner) and provider views (constraints on payment services usable by specific users, mandatory credit rating checks etc.).



- Service Discovery and Adaptation (WP5): service adaptation that is needed for aspects such as security, Quality of Service parameters etc. In addition, ranking and selection play roles when similar services are available from several providers.
- Service Construction (WP2 and WP6): An adaptive way to compose services, usable in the ISP's infrastructure, is needed. For Hanival, the tool and service for the composition needs to be integrated with the Sitebuilder application explained in Section 3.2.

The following table shows which tasks from the technical SOA4All work packages will be used in the course of this scenario, concentrating on the most relevant upcoming tasks. The following symbols indicate the usage:

	Yes Work package outcome will heavily be used
	Partly/Maybe Work package outcome will partly be applied depending on the actual progress of implementation
	No Work package outcome will not be used directly within this scenario

Task	Description	Use
WP1: Service Web Architecture		
T1.1	Web Principles and Fundamentals	
T1.3	Semantic Spaces	
T1.4	Reference Architecture, Integration and Implementation	
T1.5	Testbeds for SOA4All	
WP2: Service Deployment and Use		
T2.1	Service Provisioning	

T2.2	Service Consumption (Personalisation)	
T2.3	Service Analysis	
T2.4	SOA4All Studio UI and Infrastructure Services	
T2.5	Evaluation	
T2.6	SOA4All Process Editor	
T2.7	Recommendation based on User and Usage Data	
WP3: Service Annotations and Reasoning		
T3.1	Semantic Service Annotation	
T3.2	Reasoning with Semantic Service Descriptions	
T3.3	Ontology Instantiation and Ontology Tag Clouds	
T3.4	Semantic Service Description	
WP5: Service Location		
T5.1	Service Crawling	
T5.2	Service Indexing	
T5.3	Service Discovery	
T5.4	Service Ranking and Selection	
T5.5	Service Adaptation	
T5.6	User and Usage Data	
WP6: Service Constructions		
T6.3	Lightweight, Context-aware Process Modelling Language	

T6.4	Context-aware Service Composition and Adaptation	
T6.5	Adaptive Service Compositions Execution	

As shown above, the results of all tasks will influence the development of the eCommerce framework, its internal services, and the functionalities offered by the service broker to a certain extent. From work package 1, the architecture and various other results are implicitly used, as most other SOA4All components will depend on these. The testbed infrastructure will provide a concrete testing environment, where the various services and tools for the eCommerce framework are deployed.

Regarding WP2, the eCommerce framework will need to cover the whole lifecycle of services. Service Provisioning (and accordingly the SOA4All Process Editor) is used partly to allow the users of the eCommerce framework to set up simple composite services, based on templates. More complex process composition will not usually be needed, as useful composite services will be provided by the framework itself, or by the service marketplace (e.g., see the notion of service bundles introduced in Section 5.5.1).

End users of the eCommerce framework are going to use the consumption platform as well, for which WP9 would provide a specialised GUI in order to integrate the functionalities provided with the overall process of setting up an eCommerce application. Additionally, the service broker role provides us with the means of searching for Web Service, both with and without semantic annotations. The services, respectively APIs provided by Service Consumption can thus be used partly by the base WP9 eCommerce framework (as a GUI for end users), and partly by the service broker (i.e., by adapting their discovery and adaptation process used by the service broker).

Furthermore, Service Analysis will provide useful functionalities for the framework's end users, allowing them to gain statistical information on service usage, leading to adaptations to the services, which they have included in their eCommerce applications. As an example, users will be able to monitor how customers of their eCommerce applications use services, and identify unreliable services or other important quality of service parameters.

In order to annotate third party services and enable their discovery, the service broker in the eCommerce framework will utilize results from WP3 and WP5. Also, service crawling, indexing and adaptation is used to identify and index suitable services for eCommerce applications, and to adapt them to user needs.

Finally, the results from the Service Constructions workpackage, WP6, will be used both by the eCommerce framework itself – to set up templates for typical workflows in eCommerce applications – as well as by the service broker, in order to set up service bundles, as mentioned above.

4.3 Core Framework

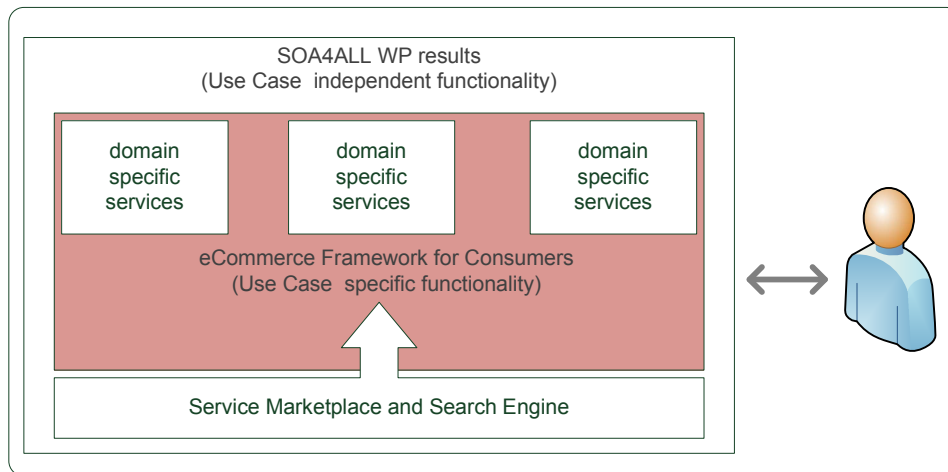


Figure 6: Role of the Core framework in the WP9 scenario

The results that are developed by SOA4All will be independent from the actual scenario and domain. They are used for developing any kind of application or system by using services or creating a process of services. However, they obviously are not focused on the C2C eCommerce domain and therefore do not provide any functionality that is needed in the domain. Succinctly, the WP results provide the tools that are required to create a web of billions of services but they do not provide the services themselves. The Core framework interacts with the SOA4All WP results, providing the following:

- A core eCommerce data model
- A user interface for C2C eCommerce
- An integration into Hanival's ISP platform, respectively the Sitebuilder application
- A set of functionalities that allow developers to add domain specific services and to access them (plugin approach)

Those four core elements are described in more detail in the following subsections. In order to ease the developments, some components and libraries may be reused coming from Hanival's ISP platform and TIE's Kinetix/Mambofive platform.

4.3.1 Core eCommerce data model

In order to create a C2C eCommerce use case, a solid data model is needed that covers typical parts of eCommerce application and that provides data structures for e.g. products and catalogues. In addition to the data model itself, a lightweight API will be created that allows accessing and managing data that is based on the data model. In addition, some kind of data mediation will be needed, to adapt the data model for use with external services, which will almost certainly use different models.

4.3.2 User Interface for C2C eCommerce

The core framework will provide an eCommerce oriented user interface. This user interface will allow users to have a shop-like view of SOA4All allowing the ISP's customers to specify how a web shop looks and which elements should be contained. It also allows users to invoke the process designer and the seekda search engine to select specific features for their C2C eCommerce system. This user interface is the main part that is accessed by users to setup their webshop. A template mechanism will allow users to create eCommerce

applications of various complexities. These Templates should also provide the technical means for the integration of the Web Services in, i.e. by providing the necessary glue code, which will be needed to combine the various services on different web pages. This will be investigated in the next deliverable D9.2.1, but depending on the results attained from the SOA4All Studio platform, this procedure will likely require some manual integrations in the context of the prototype development for the Use Case.

In addition to this, the User interface also allows users to define how the end user shop looks like. In this aspect, the UI will also define what visitors will see when they use the webshop.

4.3.3 Fully Integrated Sitebuilder Interface

The Core framework will provide a strong integration into the Hanival Sitebuilder application, respectively the website template system currently under development at Hanival. The users will still be able to define the different parts of their web presence, from static web pages to blog and image galleries, and will be able to link these pages to the eCommerce applications, which can be built through the eCommerce framework. This will allow users to start using SOA4All seamlessly, without having to switch to a separated system.

4.3.4 Plugin Registry

In addition to the points above, the Core framework will also allow developers to add APIs for developing C2C eCommerce applications and to make them available for shop owners. This is based on a plugin approach in order to make this set of services as extendible as possible. The WP9 team will provide some basic domain specific services, which are described in the next section. Plugins will provide additional services and might therefore be implemented as Web services. A detailed specification of this will be defined in deliverable 9.2.1.

4.4 Domain Specific Services

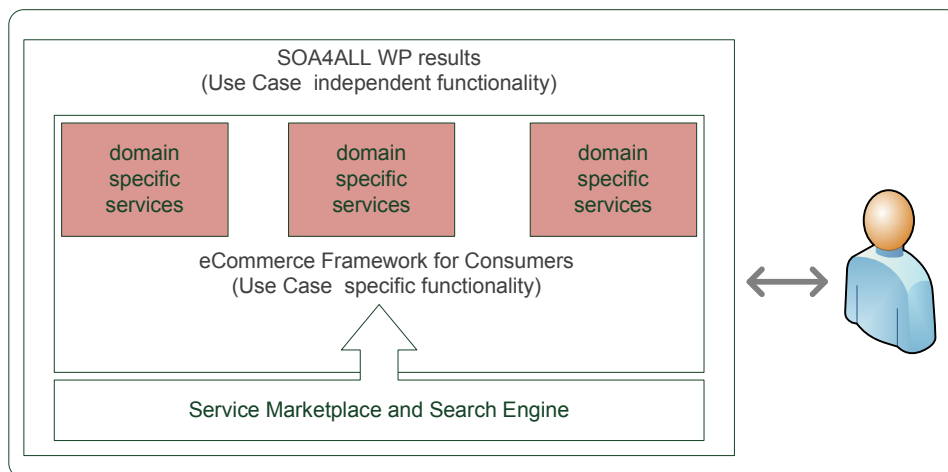


Figure 7: Domain Specific Services within the eCommerce framework

The Domain Specific Services will provide additional eCommerce functionality to users, which is neither covered by the overall SOA4All WP results nor available through the core functionalities of the eCommerce framework. This functionality is a set of different services that may be combined. As described in the last subsection, users are free to add more services using the plugin functionality of the C2C eCommerce Core framework. The WP9 team will provide the following services as a starting point. The provided services is a good mix between necessary basic eCommerce services such as product management

functionality and some examples for more advanced services used to demonstrate the possibility of handling more complex parts of the scenario (such as collaborative advertising).

4.4.1 Simple Services

4.4.1.1 *Product management*

This service will provide basic product management functionality. It will allow users to define products and to arrange products into product groups (taxonomies). It will allow specifying typical product properties such as the price or the product description. The product management service, respectively the user's view of the service, will be adapted according to the user's needs. A simple eCommerce feature, where users want to sell a couple of products on their existing web page, will not necessitate a complete product catalogue system. Still, users should be able to build on existing eCommerce applications they have already created with the eCommerce framework, i.e. they should be able to extend the functionalities and add more services in the future. Thus, the perspective of the product management service might change as well, in order to adapt to the changing requirements.

4.4.1.2 *Payment services*

This collection of services will connect the eCommerce application to real-world payment services such as PayPal or Google Checkout. The users can add additional payment services through the Service Broker. The basic payment services should enable users to start quickly with their own eCommerce applications, without the need to search for external service providers. After the user has extended their eCommerce application, the need for additional payment services is fulfilled by using the service broker. The internal services will stay useful, as we expect to provide services with a greater degree of reliability. Thus, the internal services can still act as a fallback solution, if external services are unavailable.

4.4.1.3 *Buyer Account Services*

Unlike the SOA4All user management, this service will manage user profiles of end customers (buyers) that are using the shop. It will be connected to the Hanival chillydomains user management and allow an access to the buyer history or the usernames, etc. The chillydomains platform already offers extensive services for customer management, which are used by a dedicated Customer Relationship Management system. Many of these services would be useful for shop owners, which could leverage the existing functionalities for their own customer care tasks. Buyer accounts would of course be kept separate between the eCommerce applications created by customers of the eCommerce framework.

4.4.1.4 *Order Management and Invoicing Services*

Additional services can be provided to complete intermediate steps in the order fulfilment process of an eCommerce application. An example for such a service would be the creation of invoices for an order. Invoices can be generated in various formats, such as PDF, and forwarded to the end customer via email. Currently the chillydomains platform uses a stylesheet (XSLT) based approach to generate various documents, invoices and mail notifications for its customers. As described above, shop owners will be able to reuse some of these services, for example by providing their own stylesheets as templates for the invoice generation. Other possibilities, such as offering pregenerated templates (with exchangeable branding information, like logos), will be investigated, as the current approach involves too much technical effort for the end customers.

4.4.2 Collaborative Advertisement services

4.4.2.1 TXT BeanGarden

The TXT BeanGarden is a component (part of the TXT Polymedia Platform), aiming at incentivising Internet users' participation in online advertising, thanks to a credits mechanism. Every time users visit an on-line advertising, which is BeanGarden-enabled, they accumulate credits ("beans"). Such credits can be spent later in various ways – i.e. in the eCommerce applications created by users of the eCommerce framework. BeanGarden also offers an affiliation program to third party sites – such sites will get part of the revenues generated by BeanGarden users who registered through them.

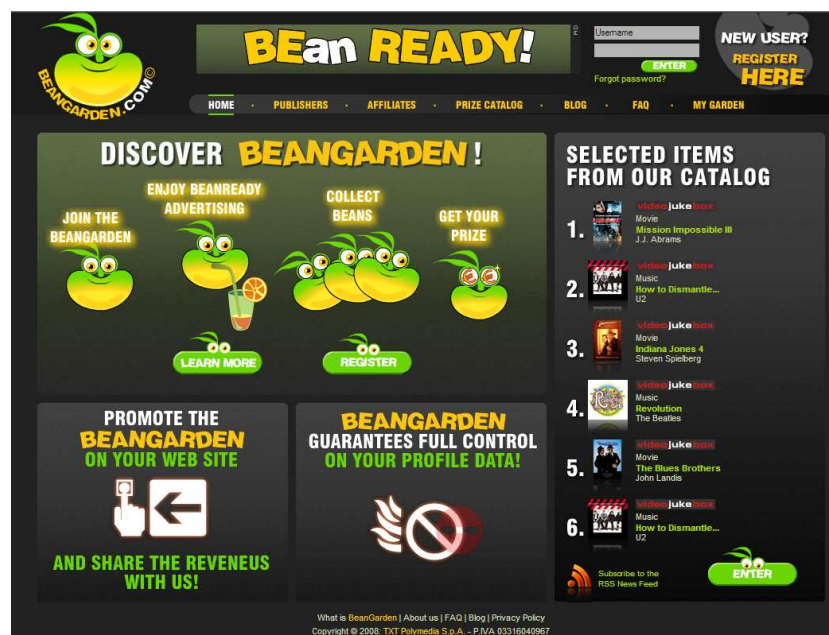


Figure 8: BeanGarden main page

Figure 8 shows the main page of the BeanGarden collaborative advertisement service, while the following figure provides more detail on how this component works:



Figure 9: BeanGarden typical use-case

A typical sequence of steps in the collaborative advertising scenario, as shown in Figure 9, is described below:

1. A User is visiting a web site embedding the BeanGarden-enabled advertising.
2. By clicking on it, they are redirected to the advertiser's site, where he finds a link to BeanGarden. If the user is already registered, he will accumulate credits.
3. Users can register with BeanGarden services by providing their data in the BeanGarden site page. If they choose to share such data, they will be granted extra-credits.
4. Users' information are received by the BeanGarden service provider who informs the site editor and/or the advertising agency (5), and the user about the total credits he accumulated

Moreover, BeanGarden incentivises the creation of user communities: users can invite friends to subscribe. In this case, such users will get part of the revenues generated by their BeanGarden friends.

4.4.2.2 Integration with the eCommerce framework

BeanGarden is a SOA application, composed by a set of user-interfaces (i.e the BeanGarden page, the banners), and a server-side set of web services, aiming at:

- Shopper management
- Publisher management
- Credits management
- Additional services

BeanGarden represents a benefit to C2C eCommerce applications: in fact, it provides benefits both for site visitors and for site owners, and allows involving advertising agents. C2C site functionalities are enhanced thanks to the advertisement services offered by

BeanGarden. SOA4All technology is required to fully exploit these. In particular:

- BeanGarden services must be made available to non-expert users: it should be possible to easily find and combine these into the web shop site.
- Most of advertisers are not using BeanGarden: it is therefore necessary to combine and orchestrate them with existing advertising services.

Therefore, it is expected that SOA4All will make it possible to integrate collaborative advertising services within the eCommerce framework and to create an added-value environment to end-users.

On the other hand, SOA4All, and the eCommerce framework created within WP9 will allow TXT BeanGarden to broaden its adoption to a much wider market, thanks to an easier way to combine it with third-party services.

4.5 Service Marketplace & Search

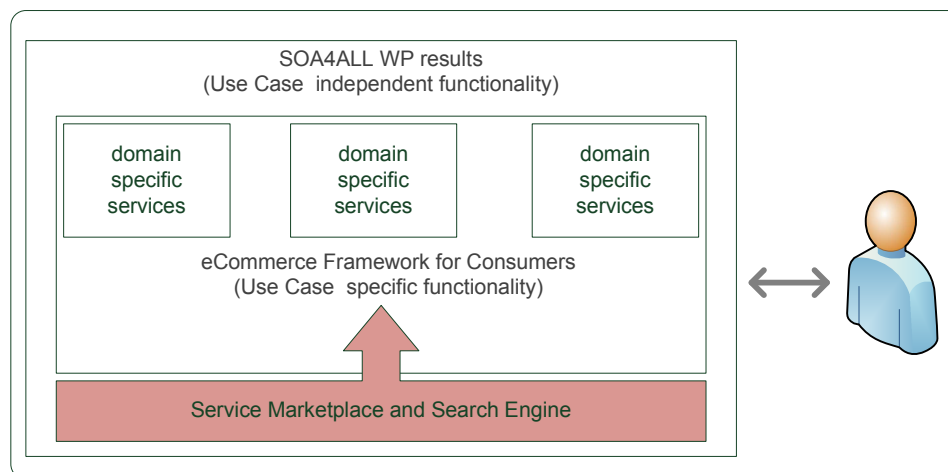


Figure 10: Service Marketplace & Search Engine in the WP9 scenario

4.5.1 Seekda Service Marketplace

The emergence of Web Services paradigm allows for a completely new e-Business software to be sold and operated in a dynamic and ad hoc fashion.

Seekda's mission is to facilitate on-demand use of services over the Web. As a first step towards its mission, seekda is operating a search engine providing access to publicly available Web Services (see Figure 11). In the future, the company also aims to simplify purchases across different providers and unify the use of services in bundles by offering a service marketplace. Our existing understanding of the Web Services market indicates that the potential users of C2C platforms will often search for complete solutions to address their business needs, rather than individual services (e.g. services to run reservation process at hotelier website, services to run registration and payment process at conference website etc.). The role of seekda in SOA4All WP9 is to provide support in integrating SOA4All results into the C2C eCommerce framework to ensure that Web Services provided by independent providers can be discovered in an ad-hoc fashion, based on requirements of a particular scenario. After discovering the right service (set of services), users of the eCommerce framework will be assembling them into coherent processes in a semi-automated and

automated fashion.

The screenshot shows the 'seekda!' web service search engine interface. At the top, there is a navigation bar with 'Seek Services', 'News', 'Consumers', 'Providers', and 'About'. A search bar contains the query 'fax' and a 'Search' button. Below the search bar, the 'Search Results' section displays 'results 1 to 10 of 552' and a list of search results. The first result is 'Fax' by 'acrosscommunications.com', which sends text document notifications to a fax machine. The second result is 'FaxService' by 'venali.net', providing fax solutions via desktop software, email, and APIs. The third result is 'fax' by 'webservicex.com', offering free fax coverage to any country. On the left side, there are sections for 'Recent News' with articles like 'Commitment to Research' and 'Improved Web Services Invoker'. On the right side, there are sections for 'Get in Touch', 'Related Searches', 'Ads by Google' (including 'Free SOA Testing Book' and 'Simulation in VB or Java?'), and 'Data Integration'.

Figure 11: seekda Web Services search engine (search results page)

The current Web Service market is far from transparent and service consumers have to spend extensive resources to locate and procure services. The search engine offered through the C2C eCommerce Framework aims to improve the transparency and accessibility of the Web Services market. The marketplace, which will be built as a part of the C2C framework, will facilitate the trade of Web Services usage in a one-stop-shopping manner - reducing effort of integrating services and costs of their procurement. Also interoperability issues between different providers are going to be handled by this platform. Figure 12 presents our initial vision for a service marketplace offered through C2C platform, showing an example of a very simple SMS service being sold directly within the C2C Platform.

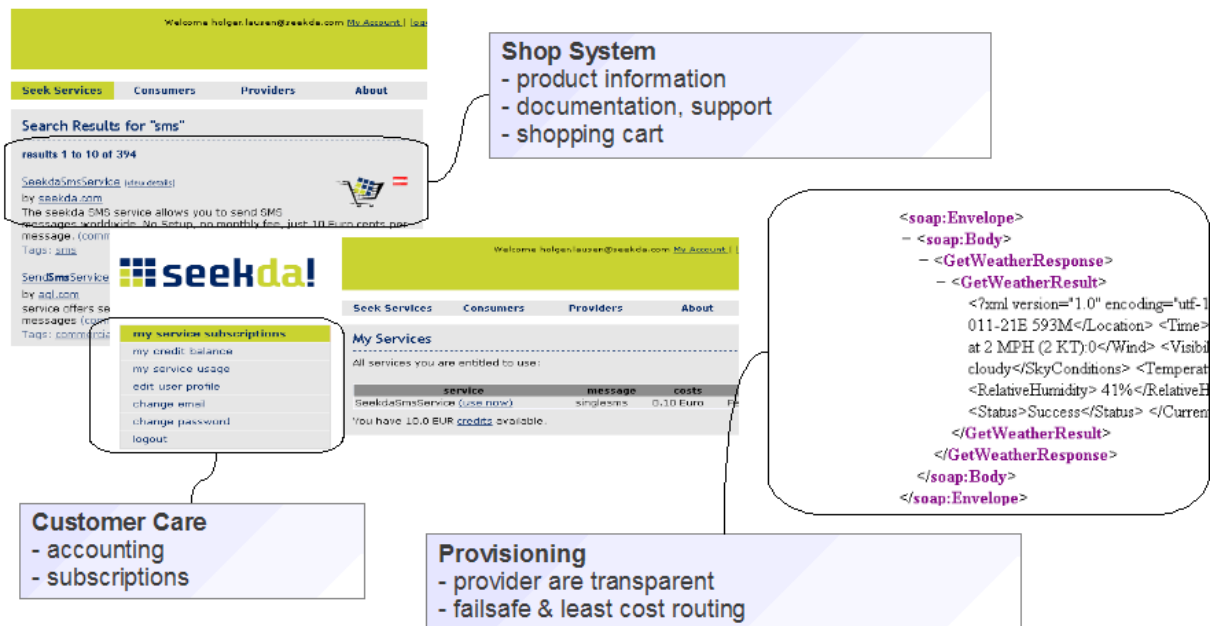


Figure 12: seekda Web Services Marketplace - Overall scenario

Using the results of the technical workpackages of the project, especially WP5, and seekda's own already existing technologies, we will add annotated Web Services to the C2C eCommerce Framework in order to further simplify the discovery process. The results of WP6 will be used to study the assembly process of complex processes to deliver more accurate and sophisticated services adjusted to specific needs. Ultimately, we foresee to use SOA4All ontologies to define all the additional properties not captured by the current syntactical descriptions.

4.5.2 Service Discovery during Design Time

In order to facilitate the use of the SOA4All framework, the application designer needs to be aware of the services available on the public Web. Web Services search engine provides currently access to the biggest set of publicly accessible Web Services (at the time of writing this deliverable seekda has indexed almost 28.000 public Web Services coming from 7.000 providers; the number of publically known WSDL files has reached a level of over 130.000) and in the future also REST services. SOA4All technology, especially the results of WP5 (Service Location), will improve the means available to explore this set of Web Services.

If an application designer needs, for example, to create a payment solution, they will need to choose a certain payment provider. This might be a service capable of processing credit card payments or transferring using some online payment provider such as paypal⁵ or clickAndBuy⁶. In addition the application designer might consider including fraud prevention services, such as offered by fraudlabs⁷ which provides an assessment whether a transaction is fraudulent or not based on several criteria (e.g., such as the location of the person

⁵ <http://www.paypal.com/>

⁶ <http://www.clickandbuy.com/>

⁷ <http://www.fraudlabs.com/>

performing an online-purchase in relation to their home address and whether an anonymous mail address or some proxy server is used).

In addition to those services that are clearly related to the domain of payment processing, the application designers might use the Web Services search engine to discover that services that are intended for completely different application domains are relevant for their applications. For example, if the task is to provide an online registration and payment facility for a scientific conference, a different kind of service might be suitable to assess fraudulent transactions. Generally, a participant of such a conference should have already published several scientific papers and articles. A search service like Alexa Web Search⁸ could then be used to check whether publications exist for a particular name. These results may be then utilized as an additional scoring criterion to prevent fraud.

Finally, the application designer might wish to quickly test discovered services, which are going to be used during the assembly process. Figure 13 presents a technology developed within seekda allowing an ad-hoc testing of public Web Services.

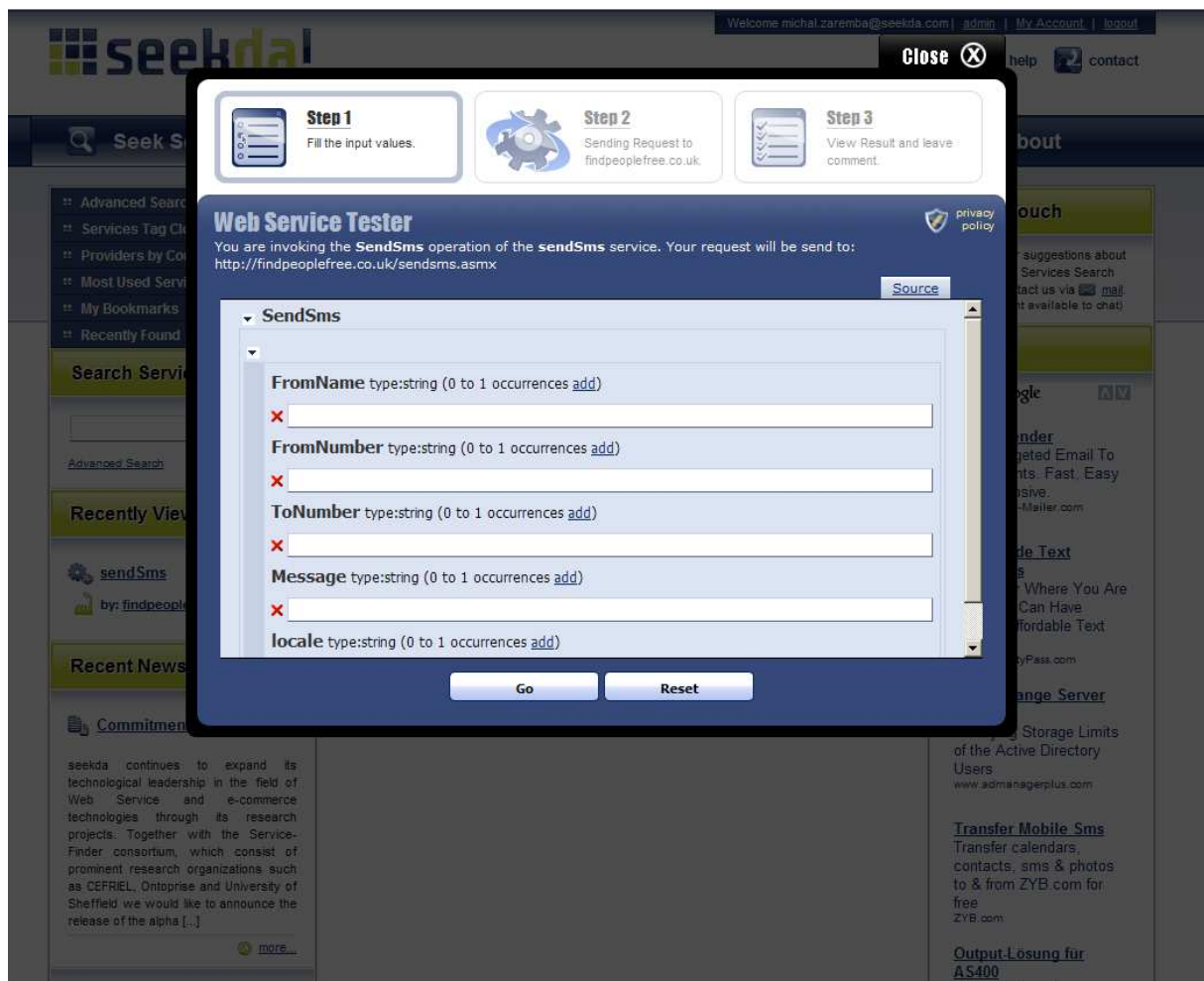


Figure 13: seekda Web Services Marketplace (Web Services Tester)

8 http://www.amazon.com/Alexa-Web-Search/b/ref=sc_fe_l_5?ie=UTF8&node=269962011&no=236156011&me=A36L942TSJ2AJA

4.5.3 Service Provisioning at Runtime

Besides the discovery of Web Services, we should be capable to provide the actual solutions rather than single services for the use case scenarios where applicable. The marketplace aims to help to abstract from the concrete implementation details of a single Web Services provider and to offer a generic service interface for a given service type (see Figures 14 and 15). If for example an eCommerce scenario would require an address validation service, it might choose to implement address validation service proxy service. The interface would be generic and abstract from the concrete provider.

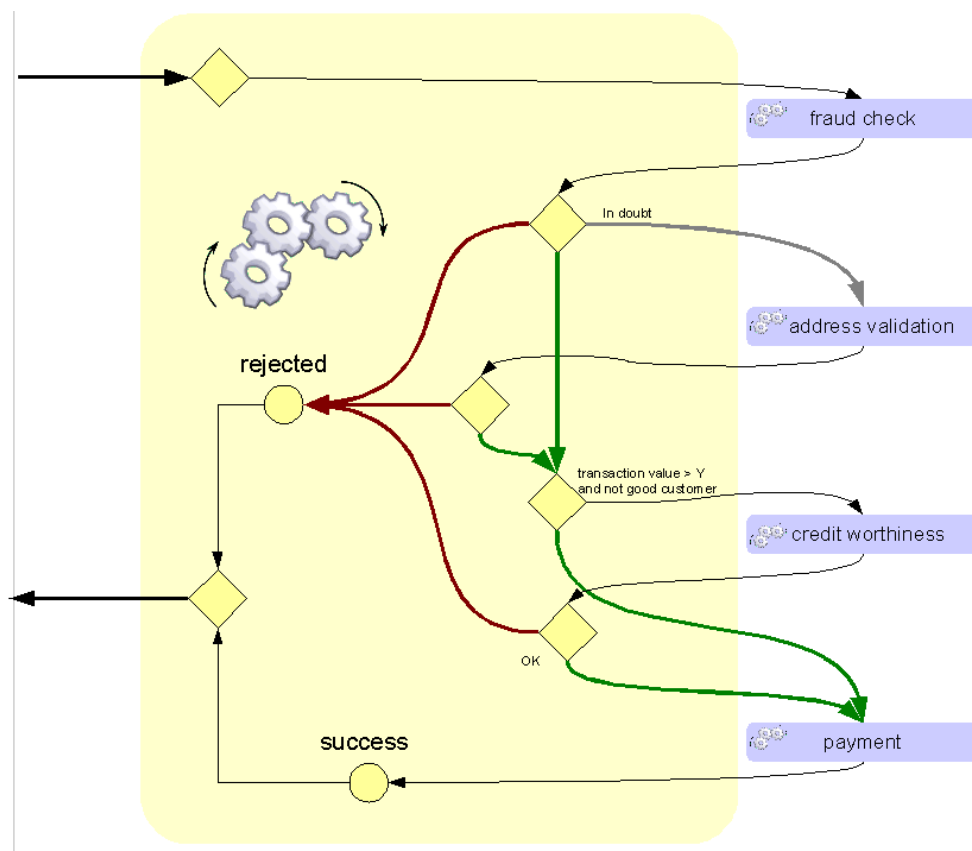


Figure 14: Service bundle – support for the whole process, not only the single functionality

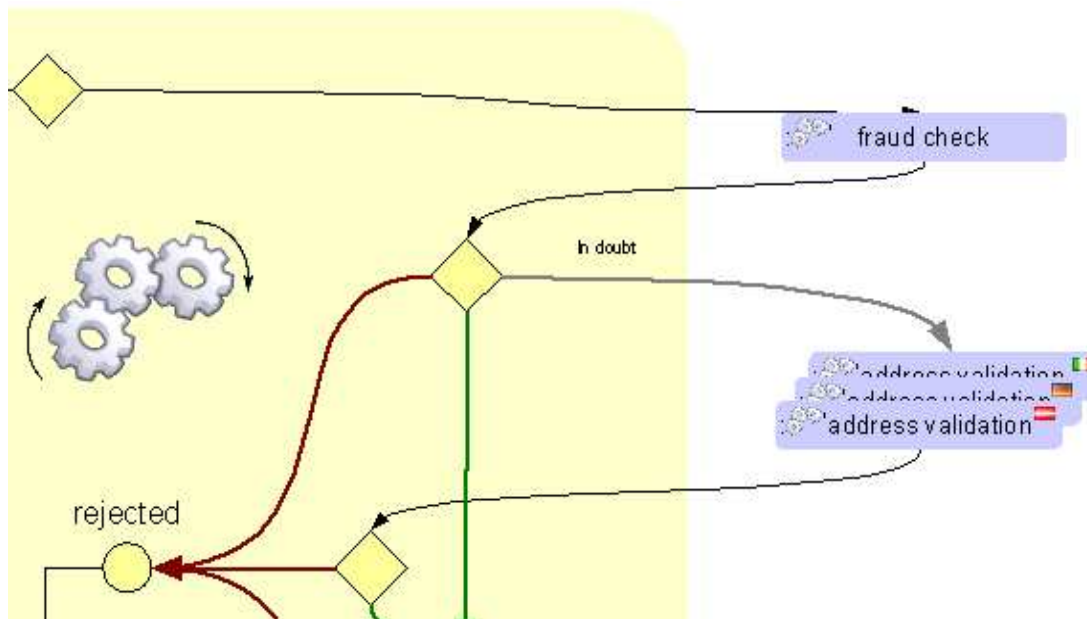


Figure 15: Data Mediation – instantiate different services and mediate data on demand

In such a scenario SOA4All technology would be used to define a generic service type abstracting from a concrete implementation. A semantic model would be used to describe the generic address validation service. Semi-automatic mappings to interfaces of different service providers allow choosing different providers at runtime, based on various criteria. Such criteria can be for example customer preferences on Service Level Agreements or jurisdiction of a provider. Another criterion might be to choose a provider based on a least-cost-routing paradigm, where the contents of a message is analysed first and based on the various pricing models of providers, the cheapest service could be chosen.

4.6 Involved data and users

The following section lists the different data elements and the types of users in the envisioned scenario. As described previously, the eCommerce application should interact with the other products offered to a customer of the chillydomains ISP platform. Many of these products already produce different data artefacts, which are reused for this scenario.

4.6.1 Data

- User generated content (including text, blogs, audio, video, pictures, comments, tags) is produced by chillydomains customers for use on their web sites or within their eCommerce applications. For example, pictures from the chillydomains image galleries can be sold within the eCommerce application. In addition, chillydomains customers could use the same pictures for product images within the product catalogue of their own eCommerce application.
- End customers, using the eCommerce application they have created with the framework, create different products and offers. Product management and catalogue functionalities of the framework will be used to generate the product entries.
- Payment and transaction data is generated during the payment processes. Depending on the offered payment service, this data can either be stored on the chillydomains platform, or directly delegated to an external payment interface.

- Additional audit data is collected during the ordering, provisioning, payment and delivery processes in the customers' web shops. Users can analyze this data to find out how services are used and to better adapt the web shop to their customers' needs.

4.6.2 Users and Roles

- chillydomains customers are customers of the chillydomains webhosting platform. They buy different products from the chillydomains web shops, which enables them to build their own web sites, including eCommerce applications. Based on the bought product, different third party services for payment, delivery etc., will be available to them for the creation of their own applications.
- Shop customers are users who want to buy in a chillydomains customer's shop, but are not necessarily chillydomains customers themselves. Data about these customers needs to be evaluated, as according to their profile, different services should be customized. As an example, if a customer is from a specific country or region, different fraud detection and payment regulation mechanisms need to be enforced during the payment process. The customer's data will also be managed by appropriate services of the platform.
- Affiliates can resell services or service bundles to other customers, but are not necessarily chillydomains customers themselves. Services can be offered through other chillydomains products to the chillydomains customers, i.e., Hanival would add third party services offered by affiliates to the pool of services which are available to chillydomains end customers within their respective chillydomains products.
- Advertisers can also be involved thanks to the Collaborative Advertising services that incentivise shop customers to participate in the site advertising.
- Service Providers register their services with the service marketplace or provide added value services, by combining several services to service bundles.
- The Service Broker offers a search engine to find additional suitable third party services, and a marketplace, which has the benefit of being able to provide service bundles, or services, which abstract from a concrete implementation, thus providing the means to link different services of the same functionality to the eCommerce applications. The services could then be selected according to availability or concrete, context-based requirements.

As part of our goal to provide chillydomains customers with suitable and intuitive tools to realise their own eCommerce applications, and using results from the work on service adaptation, we will analyze the characteristics and behaviours of these user groups. The interfaces and available tools will be adapted to the personal profiles of the current user, including language and skill levels, after classifying him or her in a number of categories.

5. Requirements Analysis

5.1 Approach

This section contains an aggregated list of requirements that are identified for the C2C eCommerce framework. In order to allow a real-world estimation of those requirements, the list classifies each entry into either **must**, **should** or **nice-to-have**.

In addition to this, the requirements are split into **business** and **technical** requirements. Business requirements are generic requirements, which are – from a business perspective – necessary to conduct successful eCommerce. Technical requirements describe specific functionalities and features that are important to take into account for the technical realisation of the eCommerce framework. This collection of requirements will help WP9 to perform a qualified planning of time and resources in the next phase of the project.

5.2 Business requirements

General business requirements for the Hanival scenario include the following issues:

- **Scalability:** the chillydomains Web hosting platform has a fast growing user basis (currently there are more than 18.000 customers; this user basis is growing by 7% per month). In addition, there is a potentially large number of applicable services for the creation of arbitrary eCommerce applications. While we expect to constrain the customer's option at first – only a defined set of services in combination with the current products will be available – the number of services available to the customers can grow quickly in future.
- **Intuitive composition of services:** As described before, chillydomains customers are generally not IT experts and will expect tools that are convenient to use. These tools should produce workable solutions without a steep learning curve or the need for additional programming and configuration. The usability of the tools and services available to the customers, respectively of the overall eCommerce framework, needs to be investigated and adapted to the needs of the user groups.
- **Security:** eCommerce applications deal with sensitive data - both payment and customer details are submitted to an application, which has been directly generated by an end user of the chillydomains platform. Thus, either additional services have to be added to the applications, which deal with authorization and payment issues, or services need to be wrapped or extended by dedicated security functionalities and other relevant cross-cutting concerns. WP9 will investigate this, but will leverage existing solutions, respectively external services, as it is out of scope for SOA4All.
- **Privacy and trust model:** In a similar vein, trust and privacy are important factors for all relevant actors. Specifically in the case of third party service providers, the end user has to be certain that both their and their customer's data will be safely dealt with. Existing solutions will be investigated by WP9.
- **Reliability:** This includes both the stability and availability of the overall eCommerce application. Since many functionalities depend on third party providers, which might not be available at all times, suitable replacement services providing the same (or at least a subset) functionality as the original service should be available and selected when appropriate. This is especially important for payment providers.
- **Extensibility:** Especially in eCommerce solutions, an extensible framework, where services are exchanged or added to existing applications, is of high importance.

Users of the eCommerce framework should be able to implement new ideas and new approaches, in order to enhance conversion rates and improving the overall experience for their end customers.

- From a business perspective, at least the following functional core building blocks for useful eCommerce solutions need to be supported:
 - Product management services need to be available, whether users want to sell a single product on their web site, or create a full featured web shop solution. This component, which can be provided as a product catalogue administration tool, should be part of the internal services of the eCommerce framework.
 - Payment management is of course of high importance to the success of an eCommerce application. This includes the ability to include several payment providers, supporting different types of payment, but also the secure and reliable transactions with these providers. Payment services are usually provided by third parties, but some default services can be offered by the pool of internal services. These internal payment services can also offer a fallback strategy in the case of unavailable external payment services.
 - Order Processing and Invoicing are also core functionalities, which have to be supported by eCommerce applications, regardless of their overall complexity. Customers which buy products or services should be led through the eCommerce applications based on a configurable order management process, which has been designed by the user of the eCommerce framework. In addition, service(s) to generate invoices after the successful completion of an order need to be provided, and should be available as internal services of the eCommerce framework.

5.3 Technical requirements

The following technical requirements for the WP9 use case scenario can be categorized by distinguishing several components of the envisioned eCommerce application framework.

5.3.1 Frontend

The eCommerce framework should present the end users with an intuitive, yet powerful frontend, which enables the users to both quickly create their own web shops and eCommerce applications and integrate them with their own web sites. The following requirements play a major role for this component.

- The end user front-end **MUST** feature similar functionalities as the commercial software Sitebuilder, respectively its eShop module described above. This includes at least the means to manage products, categories, stock, payment and delivery options and services.
- Templates for the core parts of eCommerce applications (of different complexity) **MUST** be available. Customers should be able to select the degree, or detail level for their eCommerce activities, which can range from simply selling a single product on an already existing web site, to a common web shop with shopping cart and different payment functionalities, to an overall integrated solution, including advertisement, shipment and marketing functionalities.
- Templates **SHOULD** describe the parts of an eCommerce application in the form of typical workflows for their customers.

- Templates SHOULD be presented in a simple composition or workflow notation (e.g., by using a subset of the BPMN).
- Templates for layouting the eCommerce application SHOULD also be available, in order to apply different styles and to present a way to brand the generated application according to the end users wishes.
- The frontend SHOULD allow for integration with the remainder of the web site generation tools used, i.e. the eCommerce application should seamlessly integrate with other Sitebuilder page modules as described above.

5.3.2 Service Integration and Composition

In order to select from a pool of available services, the eCommerce framework has to provide support for both the integration of third party services (based on available service descriptions in different representation languages) and for the composition of said services in order to provide more complex functionalities.

- Service Integration: SOA4All technology MUST provide support for the integration of (known) web services, described using a WSML language family variant. In addition, Web Services described in WSDL SHOULD be useable as well, as it cannot be assumed that every third party provider creates the suitable annotations in WSML.
- Service Composer: Using SOA4All technology, it MUST be possible to orchestrate selected services for the eCommerce application. These orchestrated services enable the end users to create eCommerce applications from building blocks for core functionalities, such as product provisioning and billing. While simple orchestrations will be doable by the actual end user of the eCommerce framework (given suitable template as described above), more complex composite services will be made available by service providers or the service broker.

5.3.3 Service Adaptation

Services need to be adaptable to both end user needs and to regulatory constraints (e.g., country specific regulations, payment specific regulations etc.)

- Services SHOULD be customizable, given their service descriptions and the provided user profiles. Service invocations are constrained according to the current context (both of the end user, and the respective customer).

5.3.4 Service Invocation

Services are consumed by customers of the generated eCommerce application; thus several additional requirements should be taken into account:

- Service Invocation based on SOA4All SHOULD take several non-functional requirements into account, such as requested Quality of Service level (including for example service availability etc).
- Monitoring SHOULD be available, in order to control requested versus actual service consumption and expected quality of service levels. Also, user feedback and ratings should be taken into account, as this data enable the eCommerce application owners to adapt their solutions and select services which better fulfil their customers' needs.

5.4 Requirements List

These tables provide an aggregated list of the major requirements described above, in order to provide a comprehensive overview. They are split into a table for business requirements

and technical requirements. Business requirements mainly focus on those requirements that are needed to fulfil business specific high level tasks such as reliability, usability, product management, etc. Technical requirements define requirements to fulfil tasks that are to be fulfilled at a technical level and are only indirectly noticed by the users such as e.g. service integration or invocation.

5.4.1 Business Requirements

Name	Description	Classification
Reliability	eCommerce applications have to be reliable. This includes the stability and availability of the application.	Must
Trust	Achieving trust is a complex task that is composed of multiple factors. However, it is one of the key factors for successful applications today, specifically when depending on third party providers. Existing solutions would have to be investigated, as this requirement is not within the scope of the project.	Should
Scalability	SOA4All should be applicable for all kinds of users. As such, the scenario requires a certain degree of scalability in order to provide the capability of serving many different users simultaneously.	Should
Security	Security clearly is a main requirement. It is necessary to allow a secure pipeline of services in case that multiple services are composed into complex processes, where payment data is exchanged. External services will be used for this, as it is out of the scope of SOA4All.	Must
Privacy	Whenever sensitive user information is exchanged, it is important to keep this information private and to not allow unauthorized access. Existing solutions will be investigated.	Must
Extensibility	This is especially important in marketing and eCommerce related scenarios. New ideas and new approaches for improving selling experiences may pop up very often. The platform needs to be extensible in order to allow users to react to those new approaches.	Should
Usability	Functionalities such as discovery, service	Should

	integration and composition should be presented in an intuitive manner. End users are not usually IT experts.	
Product Management	Managing and integrating products is a key functionality for eCommerce applications and Web shops.	Must
Payment Management	Similar to product management, the payment process is another core part of Web shops. Hence, the secure and reliable integration of payment providers into the eCommerce application is of utmost importance.	Must
Order Processing & Invoicing	Processing orders and sending out invoices is a key part of eCommerce today and will need to be addressed	Must

5.4.2 Technical Requirements

Name	Description	Classification
Service Integration	Services need to be integrated easily. They need to be integrated in a way that the end users do not notice that a single application is composed of multiple third party services	Must
Legacy Services	Support for the integration of WSDL services without annotations should also be available (e.g. by providing the means to quickly annotate services in a semi-automatic manner)	Should
Service Provisioning	SOA4All should allow third party providers to provide their services directly to the users of the eCommerce framework, or via the service broker	Should
Service Adaptation	(Composite) services should be customizable according to user needs and context, as well as regulatory constraints.	Should
Template Management	Templates in the eCommerce framework allow the users to select suitable services for their own eCommerce solutions of varying complexity, and to define the layout and integration with their web site	Must

Service Composition	Services need to be composed into processes that are more complex in order to cope with the user's requirements. This has to be part of the template concept described above, but should also be available to the service broker, who can combine services to provide added value services to the users of the eCommerce framework	Must
Monitoring	Monitoring Web Services is important in order to react quickly to failures or to performance issues. In addition, QoS behaviour should be monitored. In addition, customer feedback and ratings should be shown and can be used to adapt the eCommerce application.	Should
Service Invocation	Invocation should take several non-functional parameters into account, such as QoS parameters	Should

6. Conclusions

In this deliverable, an eCommerce framework for Internet Service Providers is presented, which enables the end customers of an ISP to set up eCommerce applications on their own web site. The applications can be of various complexities and are based on both internal services and additional third party services integrated via a service broker.

In addition, it describes the various stakeholders in the eCommerce framework and the possibilities to include third party services in the overall eCommerce framework. The WP9 partner seekda will provide its web service search engine and marketplace to offer third party services. Seekda's role, therefore, is the service broker, and its functionalities are enhanced by integrating SOA4All technologies, such as service location and reasoning.

Besides the business scenario descriptions and the description of the main components of the planned eCommerce framework, also established is a first collection of general business requirements for the Use Case, and more concrete technical requirements, which are expected to influence the work conducted in the technical work packages in SOA4All. The list of joint requirements will be provided to the other partners, and form the basis of further refinement and the future design of the eCommerce framework.

In the future, it is imagined that the end customers offer various solutions (products) to other end customers via the eCommerce framework, consisting and composed of many Web Services. Some examples for possible eCommerce applications include:

- e-Hotel solution allowing to run Web based reservation of rooms in the hotel
- Web shop solution allowing to sell products over the Web,
- e-Conference solution supporting in organization of conferences (e.g. including registration service, payments service, fraud check etc.)

Work on the Use Case now continues by refining the concrete ideas, and by providing a design for a C2C eCommerce framework, based on SOA4All technology.