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

D9.2.1 eCommerce framework infrastructure Design

Activity:	Activity 3 - Use Case Activities
Work Package:	WP9 - C2C eCommerce
Due Date:	M18
Submission Date:	31/08/2009
Start Date of Project:	01/03/2008
Duration of Project:	36 Months
Organisation Responsible of Deliverable:	TIE
Revision:	1.0
Author(s):	SvenAbels TIE Bernhard Schreder Hanival Matteo Villa TXT Michal Zaremba SEEKDA Hamzeh Sheikhhasan TIE Shiva Puram TIE
Reviewers:	Maria Maleshkova OU Gianluca Ripa Cefriel Dario Cerizza Cefriel

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)

Dissemination Level		
PU	Public	**X**

Version History

Version	Date	Comments, Changes, Status	Authors, contributors, reviewers
0.1	06/2009	ToC; first sections	S. Abels (TIE)
0.2	06/2009	Weekly Iteration	H. Sheikhhasan (TIE) S. Abels (TIE)
0.3	07/2009	Contributions to chapter 6	M. Villa (TXT)
0.4	07/2009	Added picture to sect 6.2, updated table in sect. 6.2	M. Villa (TXT)
0.5	07/2009	Added comments Did small corrections in Annex text Changed name of product provider #3	Sven Abels (TIE)
0.6	07/2009	Contributions to chapter 7	Bernhard Schreder (Hanival)
0.7	07/2009	Contributions to chapter 6	M. Villa (TXT)
0.8	07/2009	Integration, small typos fixed	Sven Abels (TIE)
0.9	08/2009	Section 3 updated	Sven Abels (TIE)
0.9	08/2009	Introductions to all sections added	Shiva Puram (TIE)
0.10	08/2009	Section 7 updated	Bernhard Schreder (Hanival)
1.0	08/2009	All sections updated based on internal reviews	All

Table of Contents

List of Figures

List of Tables

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
QoS	Quality of Service
SOA	Service Oriented Architecture
SWS	Semantic Web Services
WP	Work Package

1. Executive summary

The work package 9 scenario shows how the SOA4All results may be applied in the eCommerce domain. Since the M6 review, the WP9 scenario has been changed and extended significantly. WP9 has been changed to present an eCommerce framework. This framework has been initially introduced by an extra deliverable in M12. For M18 the WP9 eCommerce scenario has been strongly extended in order to reflect the suggestions of the reviewers and in order to increase the innovative aspects of the use case.

WP9 now goes clearly beyond a classical use case in a way that it does not only use and apply the results provided by the project. Instead of this, it also adds own ideas and developments to SOA4All, allowing the use case to show the innovation that SOA4All brings to eCommerce in a future looking and highly flexible web 2.0 environment. The purpose of this scenario is to demonstrate the SOA4All vision by telling a real-world story around the complete set of SOA4All components in a highly practical way so showing the usefulness of the project results. The scenario takes place in the eCommerce and social networking domains. It is about utilizing the cloud to expose services and product information from sellers but allowing the reseller to expose product offers to multiple platforms including utilizing various distribution channels including Facebook and Twitter. As such, the name of the scenario has been internally adapted to the **One Stop Cloud Shop**.

This deliverable describes the scenario in detail and it afterwards describes the infrastructure of the WP9 approach. At a project level the practical results of all RTD work packages, tasks and deliverables are involved. As such, the use case can be used to show an overall scenario to users involving all SOA4All parts. The ‘glue’ of this scenario will be implemented by WP9, which will use its resources as much as possible to develop the missing elements for demonstrating the results in a web 2.0 eCommerce domain.

This deliverable also provides an infrastructure description showing both, the SOA4All component involvement and the WP9 framework components. Based on this it performs a technology selection and an initial technical design which will be used as a base for the upcoming prototype implementation. It should be noted that WP9 has another timeline than the other use case work packages (7 and 8). As such this official M18 deliverable represents the eCommerce framework infrastructure design. WP9 does not have an official prototype at M18. However, due to the strong changes in the scenario and in order to provide a sophisticated prototype implementation by the end of the project, the WP9 team has already started with the implementations and will be able to present a prototype on a voluntary base. As such, the section 8 of this document gives an overview about the current prototype implementation status.

2. Introduction

2.1 Introductory explanation of the deliverable

This deliverable presents the design of the WP9 eCommerce framework infrastructure, based on the tools developed by all other work packages of SOA4All. The infrastructure is based on an extended scenario description, which will be given in the first sections of this deliverable.

The scenario defined in this deliverable is targeting to demonstrate the applicability of SOA4All in the eCommerce domain. It highlights the innovative aspects showing how product resellers can combine services of multiple vendors into one unique set of products, which is exposed into various web 2.0 platforms including Facebook and Twitter.

Based on the scenario description, the deliverable will give an infrastructure description highlighting what components of SOA4All are used and also describing which type of components exist in the WP9 framework.

A technology selection will be performed in this deliverable and it will form the base for the implementation of WP9 during the creation of prototypes.

The deliverable will also describe the evaluation planning by giving a brief overview about the planned activities.

2.2 Purpose and Scope

The purpose of this deliverable is to define the design for an eCommerce framework infrastructure, based on the tools developed by other work packages. The deliverable starts with a detailed description of the extended WP9 scenario with a special focus on innovative aspects. Afterwards it describes the infrastructure and performs a technology selection. The results of this deliverable will be used as an input during the rest of the project and it will act as a guideline for all implementation efforts of WP9.

This deliverable

- describes the extended scenario including different actors,
- summarizes the innovative aspects of the extended scenario,
- provides an infrastructure description showing both, the SOA4All component involvement and the WP9 framework components,
- performs a technology selection and an initial technical design, which will be used as a base for the upcoming prototype implementation,
- gives an overview about the current status of the prototype.

It should be noted that WP9 has another timeline than the other use case work packages (7 and 8). As such this official M18 deliverable represents the eCommerce framework infrastructure design. WP9 does not have an official prototype at M18. However, due to the strong changes in the scenario and in order to provide a sophisticated prototype implementation by the end of the project, the WP9 team has already started with the implementations and will be able to present a prototype on a voluntary base. As such, the section 8 of this document gives an overview about the current prototype implementation status.

2.3 Structure of the document

The document is structured as follows:

- The following section 3 will perform a clarification and refinement of the scenario scope. This will consider the feedback received in the last project review and it will describe how the work package has reacted in order to meet the review comments.
- Section 4 will describe the redesigned scenario including new aspects and describing the different roles and actors. The new scenario will cover all parts of SOA4All and will also be used as the overall project scenario for M18.
- The upcoming section 5 highlights the innovative aspects of the scenario and it also describes the added value of SOA4All to eCommerce.
- Section 6 gives a description of the infrastructure including a description of the SOA4All component involvement and the WP9 framework components.
- Within section 7 a brief technical specification is given. This section describes the technical design used in the prototype implementations.
- Although the first prototype of WP9 is scheduled for M24, the project partners have already started with the realization. As such, section 8 gives an overview about the status of the implementation efforts.
- Section 9 defines evaluation criteria by defining when the different parts of WP9 are expected to be realized and by describing which aspects of them will be tested. Please note that an own evaluation deliverable is planned in WP9 for this at a later stage. As such, this section only gives a very brief overview about the timeline.

3. Clarification and Refinement of the Scenario Scope

Please note: This section will not be included in the public version of the document

The WP9 use case has been significantly changed and extended within the last months. This has been performed in order to meet the reviewer's comments given to the last scenario description. In essence the core comment could be summarized as follows:

eCommerce scenario is still basic, without any significant functional innovation as compared to existing eCommerce platforms

It is clearly stated by the consortium that WP9 is a use case which should therefore – similar to WP7 and WP8 – not create additional new technology in any way. Instead of this it should *use* the SOA4All results in order to demonstrate how the project results can be used in a specific case (i.e. in eCommerce).

However, the consortium understands that the use case should be changed and in fact the consortium believes that adding new and innovative aspects to the use case would make the SOA4All results more attractive in eCommerce. As such the consortium has strongly changed the use case with the following aspects:

- The use case is not focusing on a holistic view of concrete actors. This allows the use case to tell a story around all parts of the scenario with concrete examples.
- The use case has removed the goal of integrating the technology of the WP9 partners. Instead of this, the partners technologies and platforms will not be part of the core scenario any more. They have either been removed or minimized. However, it must be stated that this will obviously have negative consequences for the exploitation of the SOA4All technology. The exploitation deliverable of WP9 will therefore have to address this issue.
- The new scenario focuses on modern web 2.0 platforms allowing SOA4All users to apply platforms such as Facebook and Twitter for eCommerce, based on the SOA4All results.
- The extended scenario handles all product information dynamically. Product information is combined on demand and is therefore stored "in the cloud".
- Several aspects can now be demonstrated in the extended scenario which have not been part of the old scenario such as the recommendation system and the sales optimization process.
- The 2.6 process composition editor is now an essential part of the scenario and will be used much more in comparison with the old scenario.
- The extended use case shows how product information can be exposed to multiple platforms at the same time and it shows how information can be combined from different sources. The scenario will combine three different data sources and will expose information to different target platforms after applying a new mediation service which will be developed by the WP9 team.

The following section will describe the redesigned scenario in detail.

4. Redesigned Scenario Description

This section describes the redefined WP9 scenario, which will also act as an overall scenario for the SOA4All project in M18. The scenario involves several roles, namely Buyer, Seller and Reseller.

4.1 Roles

The scenario involves several people:

Name	Role	Summary
Arian	Buyer	Visits Nadas Facebook page, sees some personalized products and decides to buy one of them.
Nada	Reseller	Wants to generate income on various popular web 2.0 sites. Uses SOA4All to connect services of various partners in order to expose product information to Facebook, Twitter, her own Webshop, etc.
Theodore, **Esteban,** **Claus**	Sellers	Want to increase their sales by offering web services, allowing resellers to retrieve a product list and to order a specific product.

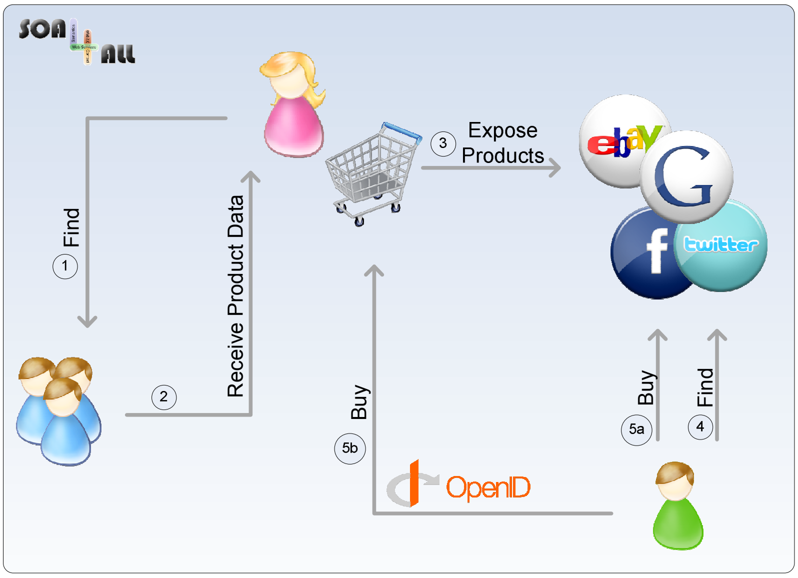


Figure 1: Roles and Process Steps

4.2 Theodore, Esteban and Claus

Theodore, Esteban and Claus are from different companies, which are all producing products from the textile industry, reaching from footwear to T-Shirts for all seasons and covering both, male and female clothes. They are responsible for the eCommerce part of their companies.

One day they read about SOA4All and they understand that SOA4All makes it simple to provide and consume services. They then decide to make their offers available to everyone via SOA4All.

They create a small service that allows people to:

- Get a list of their products and product descriptions,
- Request the price and availability of a product,
- Place a product order.

They use the SOA4All studio to add their service to SOA4All. Two of them have used web services before and already have suitable product services they want to expose to potential resellers. The third one decides to create a RESTful service.

They use the SOA4All tools to annotate their services and to describe them graphically with some semantic information based on common eCommerce ontologies, like eCl@ss[1] for product categorization and the GoodRelations[2] for details on their product definitions and order services. They can do all of this using the graphical components of SOA4All – without any knowledge of the technologies behind this – just by using drag & drop to annotate their service elements. Afterwards they click the save button to add the service to SOA4All.

[1] http://www.eclass-online.com

[2] http://purl.org/goodrelations

4.3 Nada

4.3.1 Nada's Background

Nada wants to sell some goods to generate some side income. She has registered a business allowing her to buy and sell products. Nada is skilled in IT: she uses a lot of web 2.0 platforms including Facebook, Twitter and even an alpha version of Google Wave and she even owns a small webshop where she adds textile products manually from time to time. However, via her webshop she is only making a small number of sales and the product descriptions are usually outdated. She spends most of her time that she would like to instead invest into her webshop, updating prices and availabilities or to remove or add products. Also she has no way of automatically aligning her offered products with the Web 2.0 platforms she is using. For example, she also has created an eBay shop to sell and auction some of her products, but needs to manually synchronize the two shops.

Nada wants to change this and to do this more efficiently, thus saving time and being able to spend it making more sales opportunities. She wants to be able to automate product listings and she is convinced that adding her products to her web 2.0 sites would significantly increase her sales. However, automating things is a highly technical work and even if she develops a piece of software to connect her shop to the product data of her supplier, then she would make herself bound to this supplier and wouldn't be as flexible, any more as things tend to be highly connected.

4.3.2 SOA4All

Nada chats with some friends about her problem and one of them recommends SOA4All identifying the following benefits to her:

- It would help her to easily discover supplier services that she can use as a product data source.

- It would allow her to stay flexible as she can model complex processes easily using a graphical process composer.

- It would allow her to do a fully automatic integration by directly connecting all services starting at the supplier's product catalog and ending with her product presentation. – No more manual updates of product data.

- It would allow her to optimize the overall quality of connected services. For instance, this can be achieved by minimizing their prices and ensuring seamless connections between services as well.

- It would allow her to 'send' her data everywhere as long as services permit this: to her webshop, to Facebook, to Twitter and to virtually any other web 2.0 platform that she already uses or will become available in the future. Unlike RSS feeds her product information would be seen as part of the platforms.

- She doesn't have to deal with storing outdated product data and worrying about the best description terms. She can simply rely on getting product descriptions from the cloud the moment she needs it. No need to store the description at her server if she does not want.

- She can consider context information for her sales items. For example, could be considered (based on the privacy settings of the visitor):
    - o the profile of the site visitor (gender, education, age, country, language, …),
    - o the profiles of his/her contacts/friends (e.g. birthday dates, current locations, …),
    - o the device currently used by the site visitor (e.g. an iPhone, a car navigation system, a PC, a kiosk in an hotel/shop/airport),
    - o the current location of the site visitor and eventually the weather and traffic conditions at the location,
    - o the current activity of the user (e.g. travelling, working, shopping, …),
    - o the scheduled activities of the site visitor,
    - o …

Although it sounds like a dream to her, it can all be realized with the SOA4All enabled 'One Stop Cloud Shop'

4.3.3 Discovery, Recommendation and Selection

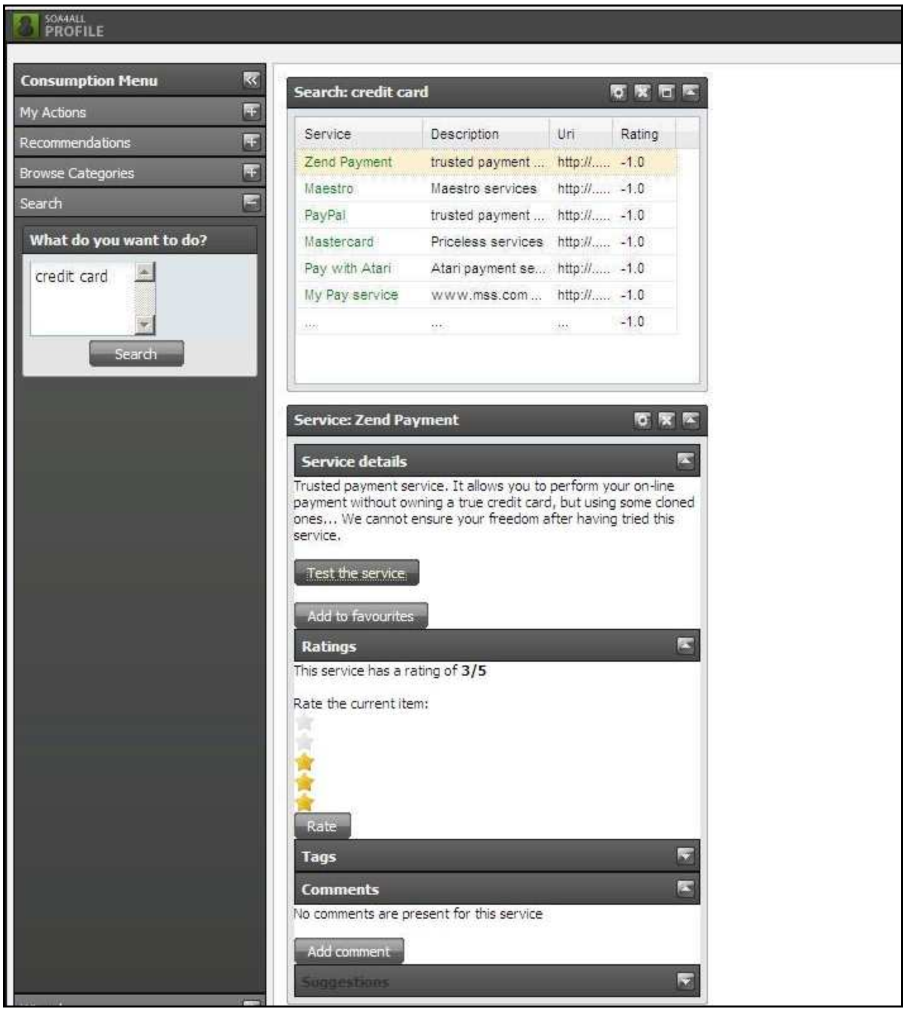
Nada visits the SOA4All Studio and creates a user profile with the SOA4All Profile Editor. Surprisingly, she notices that she can even reuse her OpenID for logging in which she uses on many other websites as well.

She then starts searching for suitable services using the SOA4All Discovery Platform and finds many services related to products and product catalogs such as a service from Amazon and other sellers.

She has the chance to refine her search and to filter the result set to those services related to the textile domain. Nada also has the chance to see comments and feedback left by other users who tried them before.

Anyway, she is still not sure about which services she really could use and thinks about it for a day or two. Fortunately the SOA4All Recommendation System is now capable of automatically recognizing her interests by analyzing her past behaviour and relating it to other SOA4All users within the Consumption Platform. Because of this SOA4All is able to "recommend" to her services that are appreciated by users with a similar profile.

She looks at the product service result list and gets an ordered list of four different services, in which the fourth one was rated not so positively, while the first three have received good feedback. She finally decides to use the three different services coming from sellers Claus, Theodore and Esteban and bookmarks it to her favourite list.



Service	Description	Uri	Rating
Zend Payment	trusted payment ...	http://.....	-1.0
Maestro	Maestro services	http://.....	-1.0
PayPal	trusted payment ...	http://.....	-1.0
Mastercard	Priceless services	http://.....	-1.0
Pay with Atari	Atari payment se...	http://.....	-1.0
My Pay service	www.mss.com ...	http://.....	-1.0
...	...	...	-1.0

Trusted payment service. It allows you to perform your on-line payment without owning a true credit card, but using some cloned ones... We cannot ensure your freedom after having tried this service.

This service has a rating of 3/5

Rate the current item:

No comments are present for this service

Figure 2: Search results and ratings

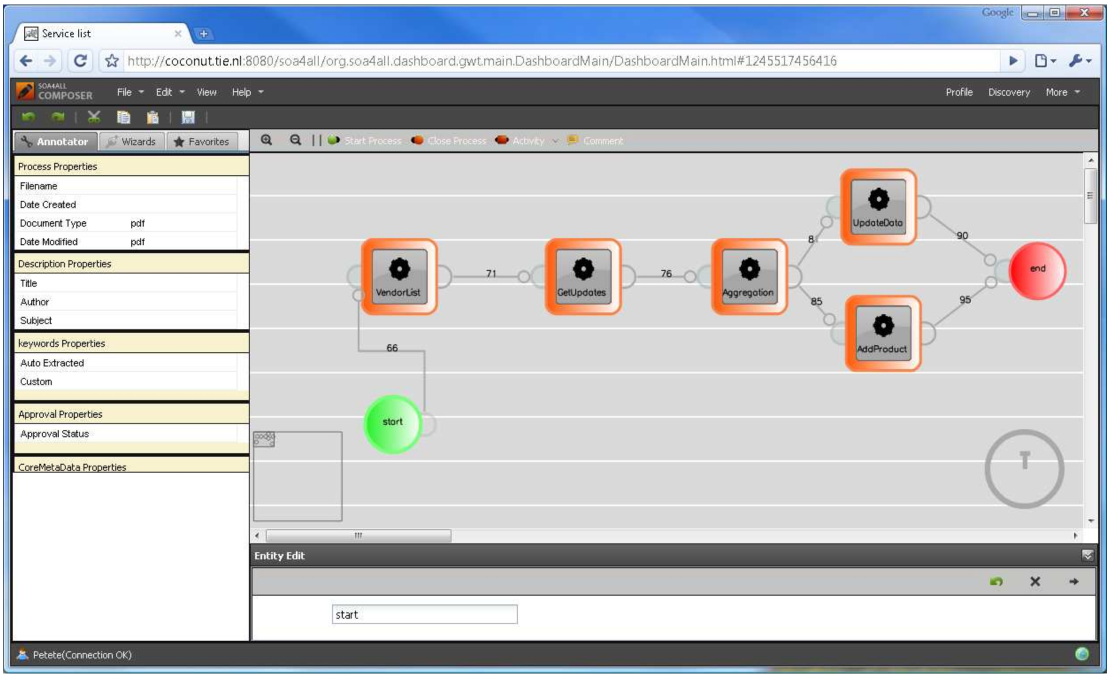
4.3.4 Integration

Having selected some services she wants to achieve, Nada wants to connect them to her webshop. For the service of Theodore she sees that there are more end points available. She selects all possible services at design time and leaves it to SOA4All to select the best one based on the current context. For example, SOA4All will select one of the service endpoints of Theodore’s service automatically during runtime.

Alternatively, Nada can also select goals instead of services. Goals are defining what Nada wants to achieve and they will be concretized on the fly by the most appropriate services given some services’ price constraints and some User preferences. For example, in terms of payment services, Nada might select speed as the main criteria meaning that SOA4All will select the fastest payment service that is able to fulfil Nadas goal.

Nada uses a pre-defined template from another SOA4All user, which allows her to take advantage of existing process templates without having to start from scratch.

She is recommended to use some mediation & aggregation service for merging and aggregating the three product services including some descriptions.



Process Properties

Filename

Date Created

Document Type pdf

Date Modified pdf

Description Properties

Title

Author

Subject

keywords Properties

Auto Extracted

Custom

Approval Properties

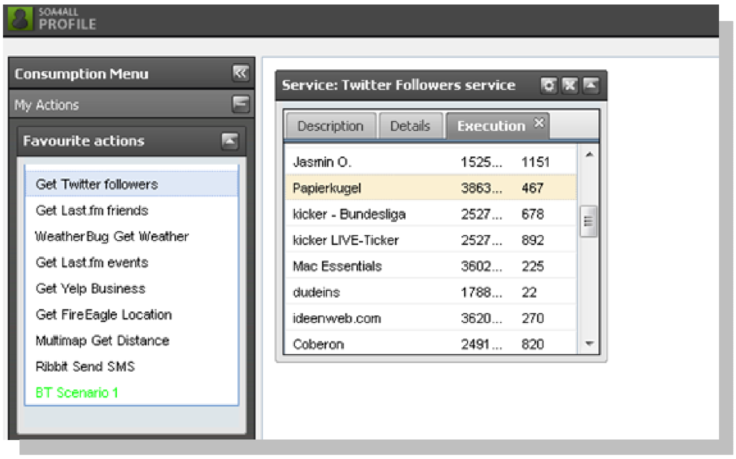
Approval Status

CoreMetaData Properties

start

Figure 3: Nada's first process

Once she has done so she uses the process editor again to create a second process, which forwards an order from her webshop to the product manufacturer service, as soon as an order arrives. She executes her process to test it and she can directly see the results of her execution.



Jasmin O.	1525...	1151
Papierkugel	3863...	467
kicker - Bundesliga	2527...	678
kicker LIVE-Ticker	2527...	892
Mac Essentials	3602...	225
dudeins	1788...	22
ideenweb.com	3620...	270
Coberon	2491...	820

Figure 4: Service execution and result

4.4 Nada (2)

4.4.1 Integrating web 2.0 platforms

Nada is happy with the things SOA4All has done for her: She does not have to care about manually updating product data or cleaning the product catalog any more. However, although this has saved Nada a lot of time, her sales are still almost the same. She did not yet expose her product data to any other platform than her own webshop. However, the saved time has enabled her to think about other marketing channels and she wonders if she can use SOA4All for bringing her sales to the next level.

She wants to use SOA4All to show her product information on some well known web 2.0 platforms so presenting her products to a large number of people. She also wants to ensure that people that visit her Facebook profile see the products belonging to their current context. For example, she does not want her Facebook profile to advertise a Winter Jacket in the summer time but she might want to advertise clothes with special Facebook logos.

Nada visits the SOA4All studio and extends the process, which gathers the product definitions from external suppliers by adding additional distribution channels. For this purpose, Nada creates a new process in the process editor, which combines her webshop and the service outputs from Claus, Theodore and Esteban and feeds them into a syndication service, which acts as a mediator for her product data. From this syndication service, Nada feeds the data via SOA4All to different platforms. She starts with Facebook and decides to extend it with Twitter, eBay and the newly announced Google Wave beta.

This process will actually lead to some important benefits for Nada:

- No more manual synchronization or conflict resolving- changes in product data, new products and outdated products are all handled automatically by SOA4All. As such, all data will be up to date including the latest descriptions, product pictures, availability dates, etc.

- No product information needs to be stored in her webshop any more, which means she does not need to maintain it. All product information is essentially “in the cloud”. Whenever her Webshop needs product data, it will request the data directly and all enabled via SOA4All. Nevertheless, Nada would, of course, have the possibilities to store some or all product data in her shop, if she wants in order to e.g. filter products or specify price information.

- Nada does not have to care about the technical details. - Using the processes that Nada has defined in the SOA4All Process Editor, SOA4All will automatically invoke the services of Claus, Theodore and Esteban, without having to bother Nada for any manual integration work.

- Context Information - Some platforms even allow Nada to get some information about the visitors and to therefore use this information when displaying products. Nada can also consider things such as the weather or the gender of a visitor.

For payment and order handling, Nada either forwards users to her webshop or uses the built-in functionality of platforms such as Facebook Credits. She starts creating another service, which takes the payment notification as an input and invokes a fraud detection service, which she found using the SOA4All service discovery functionality. Once the fraud detection service has passed, she wants to invoke the corresponding order services provided by Claus, Theodore and Esteban. However, when starting with the process creation she realizes that someone might have done such a payment process already. So she uses the search facilities of the SOA4All process editor and quickly finds an existing process that she can reuse and which only needs some minor customization. The SOA4All process composer even guides her through this modification using easy to use wizards. Thanks to SOA4All she can even define how the process will behave in case that a service is unavailable or in case of any other error that needs to be handled automatically. In case that one of the process gets too complex she can breakdown her processes into sub-processes or reuse an existing sub process thanks to the SOA4All template functionality.

Physical Goods

Using SOA4All allows Nada to do an on-the-fly integration with all data being requested on demand from the services of Claus, Theodore and Esteban. As such, Nada does not need to store data locally in her webshop. However, in some cases, Nada might actually want to either store additional product data in her webshop or to combine the data of Claus, Theodore and Esteban with information from her webshop. This integration can be done by adopting the process with the SOA4All composer. For example, in case of physical goods Nada might want to store additional information about e.g. delivery time, delivery responsibilities and number of goods available, etc.

4.4.2 Maintenance

Some days ago she has found that Theodore is offering a new interface to his service as an alternative access. She can benefit from this new service by defining it as an alternative.

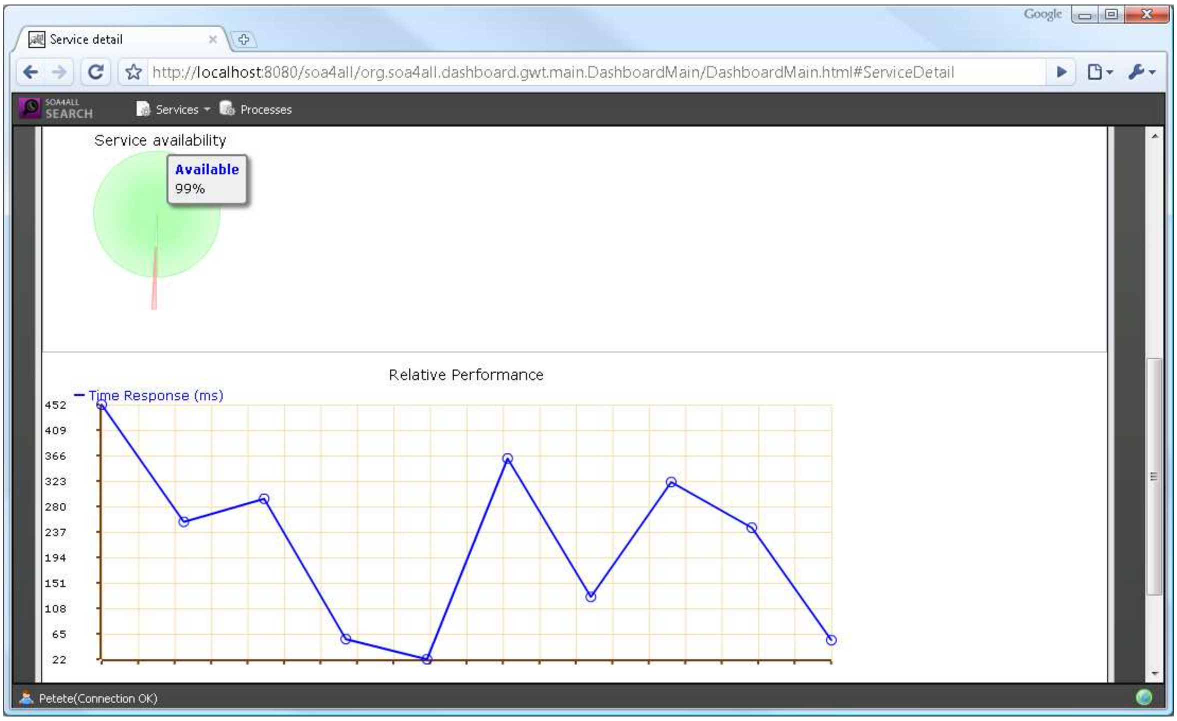


Figure 5: Performance of Theodore's service

Later, Nada wants to inspect the services she is currently using. She visits the SOA4All Studio again, logs in with her OpenID user credentials and invokes the Monitoring functionality of SOA4All. This shows her the performance of her services. She wants to avoid slow processes, so she defines a non functional requirement in the process composer about activity performance. The SOA4All composition optimizer will determine a proper service and bind it. During runtime, the SOA4All Execution Engine will monitor the service performance and replace an underperforming or failing service with another one.

Nada afterwards uses the SOA4All Studio to do an own rating of the services that she uses and to add comments. As Nada is exposing her product information to very popular sites, the number of her visitors is very high. She soon realizes from the SOA4All Monitoring information that her sites are visited million times a year but thanks to the SOA4All infrastructure she does not have any scalability problems.

4.4.3 Sales Optimization

After two months of usage, Nada feels that it is time to have a look at what she could do to optimizing her sales. Therefore, she visits the SOA4All Studio again.

Collaborative Advertising:

Nada is informed by the SOA4All Recommendation System that users with similar profiles and who were interested in the services coming from Claus, Theodore and Esteban were also very interested in a certain collaborative advertising service. Nada was not aware of this possibility, so she clicks on this service in order to get more information on it.

The collaborative advertising service incentivizes users to click on banners, as they will get more credits. This means users will be keener to visit sites offering this type of banners, and given the community incentive features of this service (where a user who brings a new member will get extra credits), the number of potential new visitors will be even greater. Nada realises that all this will bring additional revenues to her site, and will be more "friendly-seen" to shop visitors, so she decides to include it in her web-shop.

Business Intelligence:

Nada is also very interested to understand what her shop visitors are actually doing while they are interacting with her site, and to discover if there is some typical behaviour they are following, so to understand if there is some way to optimize her selling processes.

The SOA4All Template Generator is able to analyze visitors past actions, and to derive some general schemas (at different levels of abstraction and complexity) aiming at representing their processes. It has found out that 96% of her customers are directly going to the checkout process after clicking a specific product on her Facebook profile. This fact therefore suggests Nada to shorten her Process by leading people directly to the checkout page instead of showing them a confirmation page first. Nada only has to click one button in the SOA4All Process Editor to update her existing process with the more effective one.

Process Optimization:

Many new services with common functionality come up every day. So Nada was concerned about maximizing her profits. To this end she needs i) to make sure that the services she uses are the most appropriate (e.g., low price, low execution time, good availability, seamless connections between services) and ii) to satisfy end users satisfaction as well.

The SOA4All composition optimizer is able to select the most appropriate services given some Nada's constraints and preferences (e.g., a connection of services that costs less than 10 Euros, an execution time less than 4 seconds, no manual integration). Nada only has to enter her constraints and preference and then has to tick the "Optimal" box in the SOA4All Process Editor to update her existing process with the more effective one. In this way, the SOA4All composition optimizer can support her to increase her profits by making sure the services she used are the best ones for her needs.

Process Self-Adaptation

At execution time, it may happen that a service becomes unavailable or lowers its performances. In this situation it is not needed to come back to the SOA4All studio and change the design of the process. The Execution Engine is able to replace the failing or ineffective service with another one according to the same constraints and preferences expressed by Nada for optimizing the process. It is possible, due to the high dynamicity of the businesses, that a new service is deployed and becomes available for usage later when Nada has already deployed the optimized process. The Execution Engine is able to consider also the new published services, without the need of a human intervention.

Customized UI

Nada is very impressed by SOA4All and therefore recommends SOA4All to her friends. Two months later her good friend Adan uses SOA4All but meanwhile, the SOA4All WP9 team has created a user interface, which is specifically targeting eCommerce scenarios. As such, he sees a set of standard links and actions in the eCommerce Studio (which is a customized version of the normal SOA4All studio). This customized UI contains direct links to search eCommerce services and it also creates a wizard with typical eCommerce activities that are performed in modern eCommerce such as searching for payment providers, etc.

4.5 Arian

Arian is a friend of Nada but has not seen her for 6 months, when they met at a social event and they talked about her desire to set up a webshop. He asks himself what Nada is doing at the moment and finally finds Nada’s Twitter page and her Facebook profile. He sees that Nada has been pretty active and is now selling clothes via her Facebook page and surprisingly these clothes even fit to his profile and are perfect summer clothes. He decides to buy one of them. He clicks on the product, pays with his Facebook credits and happily receives the products a few days later.

Summarizing, Arian never notices that he actually deals with SOA4All. From his viewpoint everything is happening in the background but all courtesy of the project.

5. Innovative Aspects and Added Value

This section highlights the innovative aspects of the scenario and explains the added by SOA4All in this context. For this purpose, the scenario is described in two ways: The first subsection describes how the scenario would look like today using today's eCommerce software. The second subsection then describes the scenario as it is today with the help of SOA4All.

5.1 The Scenario today (i.e. without SOA4All)

Coming exactly from this domain, the WP9 partners have a very good insight on how most people would try to realize the process today. So let us imagine how the process for Nada would look like without SOA4All, or more precisely, how the process looks like today in most real world installations.

5.1.1 Theodore, Esteban and Claus

As described in the scenario above, Theodore, Esteban and Claus still want to sell products that they are manufacturing. They have a website, which they use to offer their products and they offer resellers to exchange data via CSV files, sometimes XML and often EDI.

However, as either of them aresemantic experts, they have no easy possibility to add semantic descriptions to their services. However, even if they would be able to semantically annotate their services, they would have no idea what to do with them.

So at the end of the day, they use plain old Search Engine Marketing to find business partners and they therefore have to spend about 20% of their yearly turnover in e.g. Google Adwords or Yahoo Search Marketing. They consequently have to increase the prices for their customers in order to pay for the partner-finding advertisement.

5.1.2 Nada

We assume the same situation for Nada where she wants to sell products and we assume that she has already installed a webshop that she found somewhere on the web. As a matter of fact, Nada has to deal with many things in parallel:

- Finding Partners
  Finding suitable and reliable business partners is a tough job. Nada will use Google to find business partners. After 2 days of searching she has found 50 entries and decides to send an email to them asking for the possibility to act as a reseller. She gets a reply of 25 companies with 15 companies giving her some positive feedback.
  However, along the way she will later notice that only 10 of them are reliable and honest partners but as there is no rating facility that she could use to find out this. The other 5 partners will turn our to be unreliable and will later even lead to some customer losses.

- Integrating Product Data
  Nada now wants to integrate the products into her webshop. Her webshop offers a XML import interface. However, unfortunately it is not in the same XML format as for any business partner. Some business partners offer EDI but her new webshop does not support this one.
  Her new shop even has a web service interface but unfortunately, it is not configurable enough to connect it with the service interfaces of the business partners.
  However, she finally finds a CSV interface that allows her to import the data of some

business partners and she has to accept that she has to type in the product data of the other business partners manually, which turns out to be a long lasting and ineffective task.

- Keeping Product Data up-to-date
  What turns out to be even more problematic is the process of keeping her data up-to-date. For those partners that offer CSV files, she can import and update data by manually uploading a new catalog file, which she receives via email once in a week – in case that it is not filtered by her spam filter. However, unfortunately, this import does not remove old data that was removed. It only imports new data and adds it to the existing catalog.
  For business partners that do not provide a suitable format, she has to update all products manually by visiting the website of her resellers. She does this each Monday for 2 months and then she gives up as it takes too much time.

- Connecting to web 2.0 platforms
  Nada is very active in web 2.0 communities. She would really like to add her product information to her Facebook profile or to the new Google Wave, which she is trying out. However, Nada is very skilled but she is not a developer. She can see that there is an API for many web 2.0 platforms and she sees that some people also have created different examples and services that she could use. However, none of them integrated with her proprietary webshop.
  Nada gives up but 2 months later she 'solves' this problem by manually adding a link to her profile page that brings people to her webshop including product pictures of 5 hard coded products.

- Synchronization
  Nada notices that synchronization takes a lot of time. For example, after six weeks the 5 hard coded products in Facebook are already outdated. Even her webshop contains products that are not available any more as she only gets updated once a week (see bullet "Keeping webshop up-to-date"). Synchronizing between the partners and between the platform seems to be an unsolvable task at the moment.

- Process Changes
  As everything is done manually, she has to do each step on her own, which often leads to errors and mistakes and overall does not make her happy as she feels that she could automate the full process, if she would have the right platform.

- Monitoring
  Nada received some complaints of her visitors that her webshop is sometimes down. However, as she is not capable of monitoring it, she has no idea if this is true or where it might come from.

- Sales Optimization
  Nada does have a lot of possibilities to optimize the sales process. However, she has no tool support for this and hence is bound to a long lasting and laborious try-and-error approach. In addition, she cannot easily discover new business opportunities because she cannot rely on a Recommender System.

- Execution
  If a service becomes unavailable or fails it may cause the failure of the whole process. If a new service becomes available for use Nada should develop a proper adapter in order to use it inside her process.

5.1.3 Arian

Arian will not directly notice any differences to the scenario with SOA4All technology support, as everything is happening in the background. However, Arian will notices that many products are out of date, since Nada has to synchronize all data manually.

He will also see non-personalized products meaning that he might see a winter jacket for women instead of some summer clothes for men.

Moreover, the payment and shopping processes will not be optimized, forcing him to use much more of his time than he actually wanted.

However, another problem is that Arian will basically not see any offer of Nada at all when visiting her Facebook profile unless Nada solved her Facebook integration problem. Afterwards he can see her products but when clicking on them he is only redirected to Nadas webshop start page and has to search the product manually. After 5 minutes he gives up and turns to good old Amazon not thinking about Nadas webshop any more.

5.2 Benefits, Innovations and Added value because of SOA4All

With the help of SOA4All, all actors can benefit from the SOA4All infrastructure and the SOA4All Studio and the underlying concepts and ideas bring to them.

5.2.1 Theodore, Esteban and Claus

Theodore, Esteban and Claus can use the SOA4All Studio directly without any installation. Doing so will provide them some very important advantages:

- Preparation
  They can add their services directly to SOA4All using the SOA4All Studio and they can easily annotate them using the tools annotation tools of the SOA4All Studio, which can be launched with a single mouse click from the SOA4All Studio. This will provide them with an easy way to make their product catalog semantically annotated without having to know anything about ontologies.

- Getting Business Partners
  Theodore, Esteban and Claus strongly benefit from the SOA4All discovery functionalities. Thanks to their semantic annotations, they will be found easily if someone is searching for an eCommerce service.
  Moreover, they even automatically benefit from the SOA4All recommendation system where their services will be recommended to their resellers – without spending a single €/$ to search engine marketing.

- Integration with Partners
  Of course Theodore, Esteban and Claus can still do business in the ‘old’ way but with SOA4All they benefit from an additional sales possibility and from 0 integration effort as

SOA4All will care about the integration. Theodore, Esteban and Claus will never have to worry again on how to integrate their product data with the system of their resellers.

5.2.2 Nada

Nada will benefit from all parts of SOA4All in her process. Some of them are even hidden and fully automatic so that she will not even know the technical details but will only benefit from the results:

- Finding Partners
  Finding suitable business partners is easy now. SOA4All provides a very simple search and unlike Google, Yahoo and Bing the search results will not point to webpages that might have nothing to do with what Nada wants: The SOA4All discovery functionality will list precisely those services that are matching her goals, thanks to the semantic annotations.

- Reliability (Social)
  Nada can have a look at the rating and commenting facilities of SOA4All, allowing her to benefit from the SOA4All social network. This feature will allow her to easily detect services that are bad or not reliable.

- Reliability (Technical)
  Apart from the rating facilities, Nada can also have a look at the technical reliability meaning the average response time or the availability of a service. Thanks to the SOA4All monitoring facilities, this information is only one click away and is presented to her graphically.

- Live Testing
  Nada does not need to request example data and wait for the delivery in order to test if the product data exchange really works. She can simply launch the service (consuming functionality) in the SOA4All Studio and watch the results as soon as they come in.

- Integrating Product Data
  Integrating data from Theodore, Esteban and Claus into her webshop or even into other platforms is an easy task now: She can launch the SOA4All process composer and drag and drop the services to a process desktop. Afterwards she can simply connect the services. Thanks to the semantics, SOA4All will automatically connect many input and output parameters of the services. For others, she can use a graphical editor to connect services. No need to know any details about process modeling, no need to know anything about WSDL and no need to deal with obscure formats such as CSV, EDI, etc.

- Keeping webshop data up-to-date
  As everything is based on services, all product data will always be up-to-date. Product data is directly fed from the services of Theodore, Esteban and Claus into her webshop and into other platforms. No need for manual updates or for manually deleting old products.

- Connecting to web 2.0 platforms
  Nada is very active in web 2.0 communities. With SOA4All this is much easier. She can reuse the existing SOA4All connectivity to add her data to many well-known web 2.0 communities including Facebook, Twitter, eBay and even the new Google Wave. She can simply create a new process that routes her data from her webshop or even directly from the services of Theodore, Esteban and Claus to her favorite web 2.0 platforms.

- Synchronization
  No need to care about synchronization of the different platforms. With SOA4All this will all happen in the background as all product data is coming from the cloud. In this sense, SOA4All essentially acts as a master data management for her.

- Process Changes
  Again, Nadas process is very 'flexible' now – but this time because she uses SOA4All. Every process that she has defined can be changed and extended within minutes using the SOA4All process editor. Adding new service providers or new process steps can be done by everyone without needing training in processes modelling. Furthermore, the process is able to self-adapt by reacting to some situation that may happen at runtime.

- Monitoring
  Nada received some complaints of her visitors that her webshop is sometimes down. She can handle this easily by using the SOA4All monitoring functionality. This shows her that one of the product sellers does not offer a stable process. She can either contact him or simply modify her process to get rid of the problematic service.

- Sales Optimization
  Nada does have a lot of possibilities to optimize the sales process. SOA4All gives her the right tools to do so. For example, the recommendation system gives her direct links to other interesting services such as a collaborative online advertising service. A second example is the template generator, which will analyze her process and give her concrete tips on how to improve her process. A third example is the composition optimizer that will optimize her process (by maximizing and minimizing some constraints given by Nada) and, in cooperation with the execution engine, will increase her profits by using the most appropriate services in real time.

5.2.3 Arian

Although Arian does not realize that SOA4All is used under the hood; he still benefits a lot from the new functionalities of SOA4All.

- Context Awareness
  Arial will see products that are 'made for him' or more precisely are matching his profile. He will not see any winter dresses for women in the summer time.

- Always up-to-date
  There is nothing worse than ordering a product, paying and then getting a notification that the product is not available any more. With SOA4All, Arian can be sure that all product data is up-to-date and available.

- Process Optimization
  Arian indirectly benefits from the SOA4All template generator, which will optimize the order process over time to match the requirements of the users. This will shorten the time for him to buy products and will increase his shopping experience.

- Reliability and Trust
  As Nada used the rating facilities of SOA4All, she only added product sellers that are reliable. Furthermore, in case the services fail for some reasons, the Execution Engine is able to substitute them to complete the execution. Arian therefore directly benefits from this increased reliability.

6. Infrastructure Description

6.1 WP9 Framework

In this section we provide more insight on how the different SOA4All tools will be used in order to support the WP9 scenario actors to implement their tasks. While SOA4All tools can cover most of the actions they need, it is essential that these tools are pre-configured and complemented in order to target the specific needs of the C2C scenario. It is up to the WP9 Framework to provide all the data, schemas, information and interfaces that are required to perform such pre-configurations.

The following table shows in detail for each step of the scenario the SOA4All tool to be used, the information WP9 is required to provide and the information that will be created by the actor himself

Pre-requirements – outside of WP9 scope

- **Service Product Providers:** we assume that service product providers (Theodore, Esteban and Claus) have their services already available and deployed as Web Services (including WSDL description) at their sites. In order to achieve this, the WP9 team has compiled a set of services that can be used during the scenario.**Third Party Providers:** we assume the availability of their part service providers in the area of e-payments and order processing (e.g. Paypal, etc..)

WP9 Scenario Process Step	Actor	SOA4All main tool	WP9 Framework	Information created / edited by end-user
Annotate Product Services	Product Service Providers (Theodore, Esteban, Claus)	Provisioning Platform / WSMO-Lite Editor	Pre-Created set of services	Semantic WebServices (SWS)
Have annotated services available for discovery	Product Service Providers	Distributed Service Bus Storage Services	n/a	SWS indexed into SOA4All
Register in SOA4All Studio	Nada	SOA4All studio	n/a	User's (static) profile. Optionally her OpenID
Search for Product Services	Nada	Consumption Platform	Pre-defined set of goals to search for product services	Goal or natural language query. Optionally she can edit a parametric goal
Receive suggestions about services	Nada	Recommendation System / Consumption Platform	n/a	Suggested services
Select & Bookmark	Nada	Consumption	n/a	n/a

Services		Platform		
Build Distributed Catalog Process	Nada	Process Editor	Library of Process Templates Mediation services	Process schema
Build a web-shop GUI	Nada	n/a	Access to eCommerce platform, pre-configured with some GUI templates	Web-shop design
Export catalog to multiple platforms	Nada	n/a	Interface to Facebook Interface to Twitter	n/a
Build order forwarding process	Nada	Process Editor	Library of Process Templates / Set of processes	Process schema
Build order processing and payment process	Nada	Process Editor	Library of Process Templates Set of existing payment & order processing services	Process schema
Monitor Processes	Nada	Monitoring Platform	n/a	Monitoring data
Optimise Processes	Nada	Composition Optimiser	n/a	Optimised process schema
Rate Services	Nada	Consumption Platform	n/a	Service ratings
Integrate Collaborative Advertising Services	Nada Advertisers	Provisioning Platform, Consumption Platform	Collaborative Advertising Services	Updated process schemas
Visit & Purchase on the web-shop	Arian	Execution Engine	Web-shop ready	Generate logs of eCommerce activities
Use Customized UI	All	n/a	eCommerce customization	n/a

Table 1: SOA4All tools and WP9 Framework

We can derive the list of information **WP9 Framework** is required to provide out of this table:

1. **Services:** WP9 will provide a link to some example services that can be used in SOA4All. Those service providers are not part of WP9. However, the WP9 team will provide some pre-defined services that can be used for the purpose of the scenario:

   a. A real world web service coming from an existing webshop. This service will deliver real world data and will be connected to a real data source.

   b. A real world web service from a webshop hosted on the chillydomains[3] infrastructure. This service will wrap the interfaces of a real world webshop.

   c. A simple product service, providing data for Theodore.

   Please note that those services are only examples that will be used within the project scope. However, the WP9 solutions will be able to use any other services as well. This means that we are recommending them to our end-users, but it will of course always be possible to select other alternative equivalent providers.

2. **Pre-defined set of goals:** these goals will help end-users to identify service providers in the area of product catalogs. Due to the high number of services expected, it is essential that these goals allow to focus the search by means of parameters end-users can exploit.

3. **Library of Process Templates**: templates will help end-users to compose their processes in a faster and easier way than starting from scratch. It is necessary to have a set (library) of templates focussed on the eCommerce area, more specifically about Distributed catalog process, Order forwarding, Order processing & payment. Process templates may refer to well-known third party service providers (i.e. for e-Payment, and so on), or even find most suitable providers automatically (on-the-fly).

4. **Support services:** due to the nature of the eCommerce processes, we foresee the need of additional services required to complete and to optimise the scenario. In particular, the following support services will be provided by WP9 (indeed end-users will still be able to search and to include services from third parties instead):

   a. Mediation Service: a mediation service will be required in order to merge information coming from different catalog providers (i.e. product category names, unit of measures, etc..).

   b. Collaborative Advertising Services: such services (described into D9.1.1) will incentivise the use of advertising in the web-shop, by supporting a collaborative paradigm.

5. **Multi-Channel Export Services:** these services will enable Nada to export her catalog to multiple platforms such as Facebook and Twitter. Further platforms may be taken into account if required.

6. **Web-Shop setup:** this webshop will be used out of the box without any changed in the existing webshop. Exposed web services and existing integration features will be used to show that the products selected by Arian are connected to the shop.

7. **Customized UI:** A customized UI will be added on top of SOA4All, making the access for SOA4All users in the C2C domain easier and demonstrating the flexibility.

The following picture provides a graphical representation of the scenario described in Table 1 – colours correspond to the ones used in the table, in order to show which information will be created by the end-users and which information should be provided by WP9 framework. Arrows show which information is created by whom.

[3] http://www.chillydomains.at

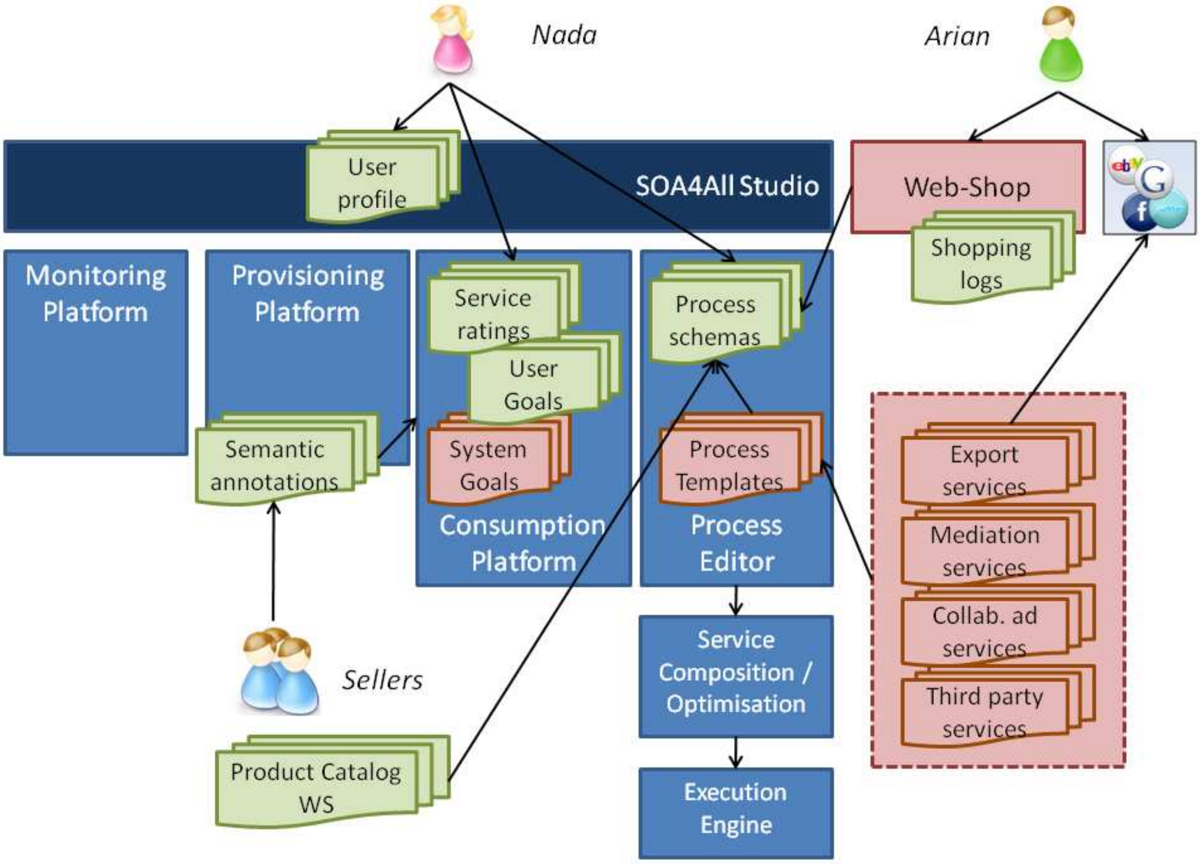


Figure 6: Graphical Representation of the Scenario

6.2 SOA4All Component Involvement

This summary contains two tables showing the usage of SOA4All project results in the WP9 ‘One Stop Cloud Shop’ scenario. Table 2 shows all the tasks for the project, and describes their involvement with the WP9 scenario. Table 3 then provides a similar overview from the perspectives of components, tools and platform services.

Task	Title	Where is it involved? How will it help?
1.1	Web Principles and Fundamentals	Input (use case requirements) provided.
1.2	Web Grounding	Continued as **T3.4** Web Grounding.
1.3	Semantic Spaces	Used by the runtime and platform components, thus indirectly used by the Use Case.
1.4	Reference Architecture, Integration and Implementation	SOA4All runtime used by all other components.
1.5	Testbeds for SOA4All	Use case requirements and service templates

		(based on Use Case services). Used for UC prototype testing.
1.6	Monitoring and Execution Context Management Infrastructure	Monitoring and context information is used in various ways. (see "Maintenance").
2.1	Service Provisioning	Used by Service Providers (Sellers) to create and annotate their services.
2.2	Service Consumption	Used for discovery of services, also for (context-based) consumption (see "Discovery, Recommendation and Selection").
2.3	Service Analysis	Monitoring and Management Tool Suite, presents useful information for sellers (who used my services how), and for the reseller to analyze the suitability of her processes.
2.4	SOA4All Studio UI and Infrastructure Services	Entry-point for the seller and reseller roles. All tools will be made available through an (adapted) dashboard.
2.5	Evaluation	Adaptation of interfaces based on usability evaluation. Improves ease-of-use of the tools, facilitating their use by non-technical users.
2.6	SOA4All Process Editor	Main tool to enable the reseller to set up the scenario of combining external product information and distributing to different channels.
2.7	Recommendation based on User and Usage Data	Recommendation data is useful for the reseller to select "fitting" services (see "Discovery, Recommendation and Selection") and easily discover new business opportunities (see paragraph "Collaborative Advertising" in section "Sales Optimization").
3.1	Semantic Service Annotation	Essential for both the discovery and composition of services in this scenario.
3.2	Reasoning with Semantic Service Descriptions	Used internally by the tools and platform services.
3.3	Ontology Instantiation and Ontology Tag Clouds	Mainly used to facilitate the annotation of services by the sellers.
3.4	Semantic Service Description	Essential for annotation and grounding.
5.1	Service Crawling	Used by the reseller to add additional services to her processes (e.g. payment, credit rating and fraud detection services).
5.2	Service Indexing	Merged with T5.1.
5.3	Service Discovery	Underlying mechanisms for discovery.

5.4	Service Ranking and Selection	Supports the selection process of the reseller to find suitable services (based on contextual info).
5.5	Service Adaptation	Necessary for selection of discovered services.
5.6	User and Usage Data	Used for ranking (comments, ratings and tags on services). Enables the reseller to get feedback from other users.
6.1	Lightweight Composition	Requirements and technological commitments for T6.3.
6.2	Adaptive Dynamic Composition	Requirements and technological commitments for T6.4 and T6.5.
6.3	Lightweight, Context-aware Process Modelling Language	Essential for the scenario, in order to provide the reseller (a non-technical user) the means to create the processes described above.
6.4	Context-aware Service Composition and Adaptation	Adaptation and composition techniques needed to create processes from the templates and user requirements.
6.5	Adaptive Service Compositions Execution	Execution infrastructure for the processes created by the reseller.

Table 2: Task Involvement

WP	**Component**	**Where is it involved?** **How will it help?**
1	Semantic Spaces	Communication and coordination infrastructure, integral part of the SOA4All service infrastructure (e.g. used for storage by Service Registry).
2	Profile Editing	Used by the reseller (the profile will then be used for recommendation and discovery purposes).
2	Dashboard	Main entry point for the sellers (to annotate their services) and the reseller (to discover and compose).
2	Monitoring	For maintenance and analysis tasks (for both the sellers and the reseller roles).
2	Annotation (WSDL)	For adding semantics to WSDL based web services.
2	Annotation (REST)	For adding semantics to RESTful services.
3	WSML Reasoning	Mainly used for discovery and composition

	Framework	tasks.
3	WSMO Data Grounding	Used by the Provisioning Platform at design time and the Execution Engine at runtime, essential for interoperability between WSML and XML.
5	Crawler	Finds additional services needed for eCommerce activities (payment, credit rating, fraud detection).
5	Service Registry	Stores and manages service descriptions, both for services provided (and annotated) by the sellers, and for additional services found by the crawler (i.e. payment and credit rating services).
5	Discovery	Find suitable product catalogue, ordering and payment services.
5	Ranking and Selection	Ranking services according to user preferences on the non functional properties of the services (e.g., for geographical context, or other user data).
6	Design-time Composer	Service Composer used by the reseller to define the processes described above, Service Adapter for context dependant configuration of these processes.
6	Template Generator	Generates process templates in order to support end-users in the selection of the most suitable one.
6	Composition Optimizer	Used to create an executable lightweight process of Semantic Web services.
6	Execution Engine	Used to execute the reseller processes.

Table 3: Component involvement

7. Technology Selection & Technical Design

In Section 6.2 several aspects of the WP9 Framework have been highlighted, which need to be provided in order to realise the full scope of the WP9 scenario. While most of the contributions from WP9 include configurations, Web Service annotations and goal and process templates, the mentioned components need technical design and development. The following section thus clarifies the necessary functionalities of these components and explicates the selected technologies, respectively, the design for the prototypical implementation of the particular components.

The current implementation status of these e-Commerce framework services will then be discussed in Section 8.

7.1 Support Services

Support services aim to facilitate the creation of eCommerce processes, mainly for the reseller role, as defined previously in Section 4.1. Two concrete services are mentioned below, though the pool of available support services is likely to grow in future iterations of the WP9 scenario as more required functionalities are discovered. These services differ from third-party services in the sense that they fulfil specific functions in the WP9 scenario and are unlikely to be offered by external service providers.

7.1.1 Mediation Service

Mediation services will be required for different purposes, though in the business scenario described in this document, the obvious role of such a service would be the aggregation of product data from different providers. The seller roles (Theodore, Esteban and Claus) have potentially different ways to define their products (i.e. concerning categorisation, price structure, etc.). In addition, they might even have annotated their product services with different domain ontologies (e.g., while Theodore has used a fragment of eCl@ss OWL to define his products, Esteban might have used and ontologised version of the UNSPSC product category taxonomy).

Either a mediation service has to align two different ontologies, or – given the same domain ontology – it might have to transform the different representations (in XML format) of product data returned from the product services. Ontology alignment is a vast area of research and has been the focus for various projects, but is not within the scope of SOA4All. For the intents and purposes of the WP9 scenario, we thus concentrate on the transformation of various XML schemata, which are annotated with the same ontologies.

Nada has to annotate her own product catalogue schema – of her selected web shop solution – with the selected domain ontology. The XML product data from the product providers is then transformed, i.e. via XSLT, to her chosen form of representation. For the WP9 scenario, the GoodRelations ontology[4] will be used to annotate the product offers in Nada's web shop. The same ontology will be used to annotate the product services of Theodore, Esteban and Claus. The Mediation Service to be implemented by WP9 will then align the XML data returned from the product providers with the format needed by Nada's shop solution.

7.1.2 Collaborative Advertising Service

Deliverable D9.1.1 has previously discussed specialised advertising solutions for webshop owners, and the refined scenario presented in this deliverable can still use the collaborative

[4] Available at http://www.heppnetz.de/ontologies/goodrelations/v1

advertisement paradigm.

The following table shows the steps that have to be taken by the different stakeholders in order to enable collaborative advertising.

Table 4: Collaborative Advertising Service

Role	Actions
Advertiser	Advertiser logs into the eCommerce framework
	Advertiser chooses to register as a "BeanGarden-enabled" advertiser
	The WP9 eCommerce framework provides him with the BeanGarden code (i.e. HTML code) to include into his banner, and the BeanGarden logo
	Advertiser is also given the user affiliation banner to include into his web-site
	The advertiser semantically annotates the advertiser's service (i.e. HTML banner, RESTful service), by using the SOA4All Studio
Shop Owner	Shop Owner logs into the eCommerce framework
	Shop Owner chooses to look for "collaborative adversting services" using SOA4All functionalities (consumption platform)
	The shop owner is presented with a list of possible banners to include into her webshop and selects one to include in the webshop
Shop Visitor	Webshop visitor finds a BeanGarden-enabled banner in the web shop and clicks on it
	Shopper can log into BeanGarden, he is given extra "beans", he is redirected to the advertiser's site

The service to provide these functionalities will be developed by WP9 and will enable the shop owner (Nada) to add collaborative advertising to her webshop.

7.2 Multi-Channel Export Services

Export services are created for the WP9 scenario, in order for the shop owner – Nada – to distribute her product data, offers and other information to different distribution channels. These services make use of existing APIs of the various targeted platforms to either directly post data to the platform (e.g., see the Twitter export service), or they transform the data and insert it to a specific repository, which is in turn accessible by the respective platform (again as an example, the Facebook Service has been designed in this manner).

7.2.1 Facebook

If Nada wants to publish her product data on Facebook, two different things have to be available to her, as previously explained. A Facebook application has to be deployed to her Facebook profile, and an export service has to be linked to the product syndication part of the process.

Facebook offers two different ways for application developers to work with Facebook data and API calls: the IFrame approach, shown in Figure 7 below, and the FBML solution, depicted in Figure 8.

IFrame solution: When the user loads an application's start page (called a canvas page on Facebook) Facebook renders a Web page that contains the Facebook chrome surrounding the IFrame. The IFrame has a URL to the application's callback URL, and Facebook appends a number of parameters, which provide more information about the current users and so forth. The content of the IFrame is completely up to the application designer and will be rendered inside the Facebook chrome. Facebook social content from the Facebook API can be used to determine which content to show inside the application, but the data returned from API calls has to be rendered by the application itself.

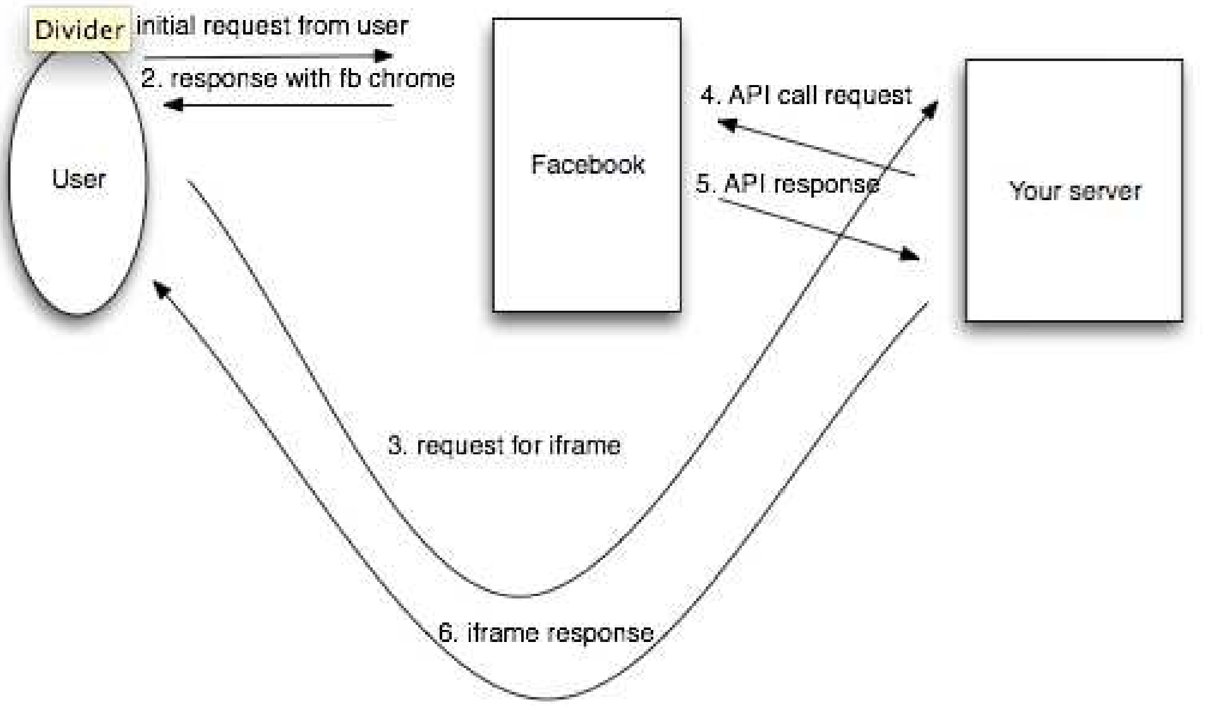


Figure 7: Traditional IFrame Canvas Page [1]

FBML solution: In contrast, Figure 8 shows the FBML solution for interacting with a custom Facebook application: When the user requests the canvas page, Facebook does not send back a response immediately; instead, Facebook sends an HTTP POST to a callback URL on the application's server. Facebook then expects the server to return FBML (Facebook Markup Language), and then converts that FBML into HTML and sends it back to the user's browser. Facebook data, like names and pictures, can be added directly in FBML, instead of going through the Facebook API (by using tags like fb:name and fb:profile-pic to reference data directly).

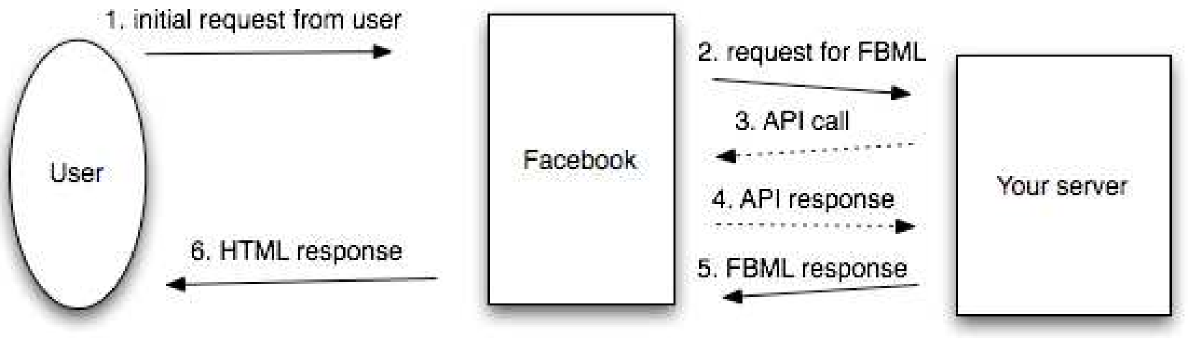


Figure 8: FBML Canvas Page [1]

For the purposes of the Facebook shop application built in WP9, the IFrame solution will be used, as it provides a more fine-grained approach to social data, and allows the Facebook application to query additional data from visiting users.

As show above, when creating a Facebook application, it is often necessary to access the social context by utilizing profile, friend, Page, group, photo, and event data. The Facebook API has been created to make these functionalities available to application programmers. The API uses a REST-like interface, and exposes a large number of Web Services. As an example for a typical Facebook API method, Table 5 shows the method used to get standard information about users. This method can be used by Nada's Facebook application to collect useful information on visiting Facebook users (provided the permission settings allow this kind of information gathering). The user data can then be used by the Facebook application to customize the offered products.

Table 5: Querying a Facebook User

Facebook API Method	***Users. getStandardInfo***
Description:	Returns an array of user-specific information for use by a Facebook application itself. Returned user data includes: # first_name # last_name # timezone # birthday # sex # affiliations (regional type only) # locale
URL:	http://api.facebook.com/restserver.php
Formats:	xml
HTTP Method(s):	GET
Requires Authentication:	true

As a special feature of the Facebook API, a dedicated query language is available, called FQL (Facebook Query Language), which allows user to create SQL-style queries for the Facebook social data. An example FQL query is shown below. This query will return results with similar information like those, which are delivered by invoking the *getInfo* method of the Facebook API:

SELECT uid, birthday FROM standard_user_info WHERE uid IN (uids)

The Facebook export service will uniformely use the Web Service interface. A dedicated Java library will be used to interact with Facebook inside the eCommerce application, called facebook-java-api[5]. The library has been made available under an MIT licence.

7.2.2 Twitter

The Twitter micro-blogging service provides two discrete APIs: a REST API for the core functionalities, including updating one's status and getting personal and public timelines, and a Search API to look for specific tweets with certain keywords or by certain users.

The Twitter API attempts to conform to the design principles of Representational State Transfer (REST). As usual for RESTful Web Services, methods to retrieve data from the Twitter API require a GET request, while methods that submit, change, or destroy data require a POST, though a DELETE request is also accepted for methods that destroy data. The Twitter APIs return meaningful HTTP Response Codes, that need to be interpreted by invoking parties.

Each method of the Twitter APIs supports a different set of formats for the returned results. The actual returned format can be selected by extending a request with a suitable format extension (see the example for a specific Twitter API call below). Usually supported data formats include XML, JSON, and the RSS and Atom syndication formats.

Numerous libraries exist for various languages to address the Twitter APIs programmatically. For the purposes of the WP9 eCommerce export services, the Twitter4J[6] library has been selected. This open source library (released under a BSD-style license) is a pure Java library, which includes built-in OAuth support and requires no additional libraries.

For the Twitter export service made available by the eCommerce framework, the following main API method is supported:

Table 6: Twitter Status Update

Twitter REST API Method	***statuses/update***
Description:	Updates the authenticating user's status. Requires the status parameter specified below. Request must be a POST. A status update with text identical to the authenticating user's

[5] available at http://code.google.com/p/facebook-java-api/

[6] available at http://yusuke.homeip.net/twitter4j/en/index.html

	current status will be ignored to prevent duplicates.
URL:	http://twitter.com/statuses/update.*format*
Formats:	xml, json, rss, atom
HTTP Method(s):	POST
Requires Authentication:	true

7.2.3 eBay

eBay provides an exhaustive set of API methods to developers, and thus provides a fertile ground for third party providers to create their applications and websites interactive with the auction house. Generally, the available methods are collected in large sets of functionally related APIs, including the Finding API, Shopping API, Merchandising API, Trading API and others.

For the purposes of creating an export service for SOA4All WP9, the Trading API was selected, and suitable methods to post product data and auctions have been identified. One of the most important functionalities is to actually list an item (i.e., a product or offer) on eBay. To list an item on eBay, the AddItem method is used (as shown in Table 7). Doing this involves three general steps: setting up the execution environment, instantiating an ItemType object (and setting its properties with values that define the new item), and making the API call. Generally, eBay methods are available both as WSDL based and RESTful services.

Table 7: Selling Items with the eBay Trading API

eBay Trading API method	***AddItem***
Description:	Use AddItem to define a single new item and list it on a specified eBay site. To list multiple new items in an application, execute AddItem once for each item, with a new item definition each time. AddItem returns the item ID for the new listing plus the estimated fees for listing the item.
URL:	https://api.ebay.com/wsapi
Formats:	SOAP
HTTP Method(s):	POST

A sample request to list an item on eBay is shown below:

```
<AddItemRequest xmlns="urn:ebay:apis:eBLBaseComponents">
      <RequesterCredentials>
            <eBayAuthToken>ABC...123</eBayAuthToken>
      </RequesterCredentials>
      <WarningLevel>High</WarningLevel>
      <Item>
```

```
            <CategoryMappingAllowed>true</CategoryMappingAllowed>
            <Country>US</Country>
            <Currency>USD</Currency>
            <Description>Store Inventory format listing sample with a
quantity of 10.</Description>
            <ListingDuration>Days_30</ListingDuration>
            <ListingType>StoresFixedPrice</ListingType>
            <Location>San Jose, CA</Location>
            <PaymentMethods>AmEx</PaymentMethods>
            <PaymentMethods>VisaMC</PaymentMethods>
            <PaymentMethods>PayPal</PaymentMethods>
            <PayPalEmailAddress>test@ebay.com</PayPalEmailAddress>
            <PrimaryCategory>
                  <CategoryID>307</CategoryID>
            </PrimaryCategory>
            <Quantity>10</Quantity>
            <Site>US</Site>
            <StartPrice currencyID="USD">1.00</StartPrice>
            <Storefront>
                  <StoreCategoryID>1</StoreCategoryID>
                  <StoreCategory2ID>20</StoreCategory2ID>
            </Storefront>
            <Title>Pink Floyd Dark Side of the Moon</Title>
      </Item>
</AddItemRequest>
```

7.2.4 Google Wave

The Google Wave[7] is currently in an early stage of development – at the time of this writing, it has just been made available to beta phase developers. Still, it provides a new communication paradigm that can be leveraged for eCommerce platforms. The next version of the WP9 prototype deliverable, due at M24, will contain updated information on the planned version of an export service, which can link to a Google Wave.

7.3 Web-shop Building Services

Finally, as part of the eCommerce framework and services that should be provided to the users of the reseller role in WP9's scenario, the means to work with existing webshop and eCommerce solutions have to be provided. The following example highlights the way in which SOA4All results will be linked to existing solutions.

7.3.1 Mambofive

Mambofive is an eCommerce solution providing all modern Webshop functionalities. It provides a service interface allowing technical experts to access the webshop data using RESTful services. Within the past, this interface has only been usable by experts. With the help of SOA4All, the interface can now be used by anyone that uses the SOA4All results. As such, the interface will be used to receive product data from Theodore.

[7] The API documentation is available at http://code.google.com/intl/de-AT/apis/wave/

8. Prototype Status

This section gives an overview of the prototype status as of month 18. This task was scheduled originally for month 24 but for giving a complete picture of the scenario, the team decided to start already with its implementation and prototype effort. As such, this section will give a status overview about the ongoing activities and it should be noted that all implementation effort that has been made is ahead of the original time planning.

The prototype is divided into three parts based on the people involved in the scenario described earlier in this deliverable. Eeach part explains the tasks and their status (see table below).

▨	Implemented / finished
□	New / unimplemented
B: Buyer, **R:** Reseller**,** **S:** Seller	

The progress in the table shows the completion of the work in context of the WP9 scenario. In most cases, the scenario relies of the results of the other SOA4All work packages. As such, the following percentage values can be interpreted as follows:

- If an element is marked as 100% then it means that it has been implemented and also fully integrated into the scenario in a perfect and final way.
- If an element is marked as 0% then this functionality is either not yet provided by the corresponding work package or it is available but not yet integrated into the WP9 scenario.
- Any other values mean that an element is already part of the WP9 scenario implementation but it might be only partly realized. For example, some services might still be hard-wired in the first prototype version.

8.1 Theodore

The following table shows the status of each task related to Theodore who wants to increase the sales of his company by providing services.

ID	Name	Progress	
S1	Register in SOA4All using Profile Editor	▨▨▨▨▨□□□□□	**50%**
S2	Service annotation using Good Relations[8]	▨▨▨▨□□□□□□	**40%**
S3	Store service in SOA4All	▨▨▨▨▨▨□□□□	**60%**

[8] http://purl.org/goodrelations

S4	Monitor service usage	□□□□□□□□□□ **0%**
S5	Update sales plan and product definitions	□□□□□□□□□□ **0%**
S6	Service annotations using additional ontologies (in comparison to S2)	□□□□□□□□□□ **0%**
S7	Update service in SOA4All	□□□□□□□□□□ **0%**

8.2 Nada

The following table shows the status of each task related to Nada, who wants to generate more income by connecting services throughSOA4All in order to expose information to various popular web 2.0 sites such as Facebook.

ID	Name	Progress
R1	Login using OpenID	■■■□□□□□□□ **30%**
R2	Search for services	■■■■■■■□□□ **70%**
R3	Check ratings and comments for each service	■■■■□□□□□□ **40%**
R4	Check recommendations	■■■□□□□□□□ **30%**
R5	Test Theodore’s service	■■■■■■□□□□ **60%**
R6	Add services including Facebook service to bookmark	■■■■■■■□□□ **70%**
R7	Check pre-defined templates to Model	□□□□□□□□□□ **0%**
R8	Model process using Composition Editor	■■■■□□□□□□ **40%**
R9	Add or Merge services, parameters based on an automatic approach	■□□□□□□□□□ **10%**

R10	Monitor service usage	▨▨□□□□□□□□ **20%**
R11	Rate and recommend services	▨▨□□□□□□□□ **20%**
R12	Improve marketing channels based on context	□□□□□□□□□□ **0%**
R13	Adjust and extend the Model process	□□□□□□□□□□ **0%**
R14	Add a syndication service to mediate between web 2.0 sites and Nada’s product data	□□□□□□□□□□ **0%**
R15	Add a service to handle payment issues	□□□□□□□□□□ **0%**
R16	Replace services and add rating	□□□□□□□□□□ **0%**
R17	Execute complex scenario	□□□□□□□□□□ **0%**
R18	Optimize sales and extend partners	□□□□□□□□□□ **0%**
R19	Make use of the collaborative advertising service	□□□□□□□□□□ **0%**
R20	Analyse webshop visitors to update process	□□□□□□□□□□ **0%**
R21	Automatic process optimization	□□□□□□□□□□ **0%**
R22	Improve process self-Adaptation	□□□□□□□□□□ **0%**
R23	Customise User Interface	□□□□□□□□□□ **0%**
R24	Use Advanced web 2.0 platforms: Twitter, eBay ..etc	□□□□□□□□□□ **0%**

R25	Monitor the process and compare results	□□□□□□□□□□ **0%**
R26	Install a webshop	▨▨□□□□□□□□ **20%**
R27	Integrate the webshop with the process model	□□□□□□□□□□ **0%**

8.3 Arian

The following table shows the status of each task related to Arian, who wants to buy some products already seen in Nada's Facebook page.

ID	Name	Progress
B1	Visits Nada's Facebook public account	▨▨▨▨▨▨▨▨▨▨ **100%**
B2	Chooses a product and pay	▨▨▨□□□□□□□ **30%**

9. Evaluation Planning

We assume that subsequent versions of developed software must be tested and validated. In this section we provide a set of initial ideas for comparison and functional evaluation of e-Commerce infrastructure presented in this document.

For comparative evaluation we will provide a comprehensive description of how the solution worked before SOA4All and after the technology was developed (started already in this document in the section 5). A comprehensive validation/evaluation report (D9.4.2) will be based on a thorough evaluation of the SOA4All tools and technologies used in the context of this scenario. This evaluation will include potential user interviews/surveys, and usability testing.

Within this section, we distinguish between three types of tests that will be performed by WP9: (i) Integration tests, (ii) functional tests and (iii) performance tests. In addition to those, we assume that each work package and each tasks will also perform a solid unit tests independently of WP9, as described in the test framework deliverable D1.5.1 of SOA4All.

Integration tests will be performed whenever a deep technical integration of different components is necessary, which have not been developed by the same work package. For example for creating and executing a process, WP9 has to create a process with the results of task 2.6 (WP2) and need to connect the services registered in the distributed service bus (WP1) in order to use them in the process.

Performance tests will allow WP9 to identify bottle necks of the SOA4All developments. In case that such performance problems are detected, WP9 will give immediate feedback to the corresponding task work package leaders.

For functional evaluation we require a set of tests to be defined for each stakeholder of the scenario, namely buyer (Arian), reseller (Nada) and seller (Theodore, Esteban and Claus). Upcoming sections define procedural guide for testing activities that should be carried out for the developed application. It identifies the tests to be performed, and provides schedules for test activities.

9.1 Tests for Seller

The following table shows tests that have to be preformed by the seller.

Req ID	Name	Integration testing	Functional testing	Performance testing	Schedule
S1	Seller can register in SOA4All using Profile Editor		**Yes**		**M24**
S2, S5 and S6	User can annotate service using Good Relations		**Yes**		**M24**
S3 and S7	User can store and update his/her service in SOA4All	**Yes**	**Yes**	**Yes**	**M24**
S4	User can monitor the usage of his/her	**Yes**	**Yes**		**M30**

	service				

9.2 Tests for Reseller

The following table shows tests that have to be preformed by the reseller.

Req ID	Name	Integration testing	Functional testing	Performance testing	Schedule
R1	Reseller can login using OpenID		**Yes**		**M24**
R2	Reseller can search for Services	**Yes**	**Yes**	**Yes**	**M24**
R3	Reseller can check ratings and comments for each service	**Yes**	**Yes**		**M24**
R4	User can Check recommendations	**Yes**	**Yes**		**M24**
R5	User can test service registered in S3	**Yes**	**Yes**		**M24**
R6	User can add services including Facebook service to bookmark	**Yes**	**Yes**		**M30**
R7	User can check pre-defined templates to Model	**Yes**	**Yes**		**M30**
R8, R13	User can model, adjust and extend process using Composition Editor	**Yes**	**Yes**		**M24**
R9 and R16	User can add or merge services to the process and execute them. Reseller can replace services and add rating	**Yes**	**Yes**		**M30**
R10 and R25	Reseller can monitor service usage and process; compare results	**Yes**	**Yes**	**Yes**	**M30**
R11	Reseller can rate and recommend services	**Yes**	**Yes**		**M30**

R12	Reseller can improve marketing channels based on context	**Yes**	**Yes**	**Yes**	**M36**
R14	User can add a syndication service to mediate between web 2.0 sites and the reseller’s product data		**Yes**		**M36**
R15	Reseller can add a service to handle payment issues		**Yes**		**M36**
R17	Reseller can execute complex scenarios		**Yes**	**Yes**	**M36**
R18	Reseller can optimize sales and extend partners		**Yes**		**M36**
R19	Reseller can make use of the collaborative advertising service		**Yes**		**M36**
R20	User can analyse webshop visitors to update process		**Yes**		**M36**
R21	System can automate process optimization		**Yes**		**M36**
R22	System can improve process self-Adaptation		**Yes**		**M30**
R23	Reseller can customise User Interface		**Yes**		**M30**
R24	Reseller can use Advanced web 2.0 plat-forms: Twitter, eBay etc.		**Yes**		**M30**
R27	Reseller can integrate the webshop with the process model	**Yes**	**Yes**		**M24**

9.3 Tests for Buyer

The following table shows tests that have to be performed by the buyer.

Req	Name	Integration	Functional	Performance	Schedule

ID		testing	testing	testing	
B1	Buyer can visit reseller Facebook public account		**Yes**		**M24**
B2	Buyer can choose a product and pay		**Yes**		**M30**

10. Conclusions

This deliverable presented the design of the WP9 eCommerce framework infrastructure, based on the tools developed by all other work packages of SOA4All. The infrastructure is based on an extended scenario description, which has been given in the first sections of this deliverable. The scenario defined in this deliverable aims to demonstrate the applicability of SOA4All in the eCommerce domain. In order to address the reviewers' comments, the scenario has been strongly extended and has moved its focus to the web 2.0 domain, showing how SOA4All can be used to create new opportunities.

Within the document, we highlighted the innovative aspects showing how product resellers can combine services of multiple vendors into one unique set of products, which is exposed into various web 2.0 platforms including Facebook and Twitter. A brief technology selection has been performed in this deliverable.

The results of this deliverable will serve as a basis for future activities of WP9, including the creation of the first prototype, which will be delivered in M24. The status section of this deliverable shows the current intermediate status of this activity. The evaluation planning described in this deliverable will be used to continuously monitor the progress of the prototype implementation.

11. References

1. Charlie Cheever: Choosing between an FBML or IFrame Application, Facebook Developers Documentation, available at http://wiki.developers.facebook.com/index.php/Choosing_between_an_FBML_or_IFrame_Application, last accessed in August 2009.
2. Schreder, B., Villa, M., Abels, S., Zaremba, M.; Deliverable D9.1.1: Future C2C eCommerce Requirements and Scenario Descriptions, SOA4All: Service Oriented Architectures for All - 215219.
3. eCl@ss. 2004. eCl@ss White Paper, V0.6, 2001, http://www.eclass.de, last access 17.09.2004

12. Annex 1: Installation Guidelines

WP9 does not have to deliver an official prototype at the moment. Nevertheless the WP9 team has decided to integrate its current implementations into the SOA4All Dashboard and to provide it to the public. As such, the installation of the current prototype is very easy. Please ensure that Java 1.6 and Tomcat 6.0 have been downloaded and installed on your system.

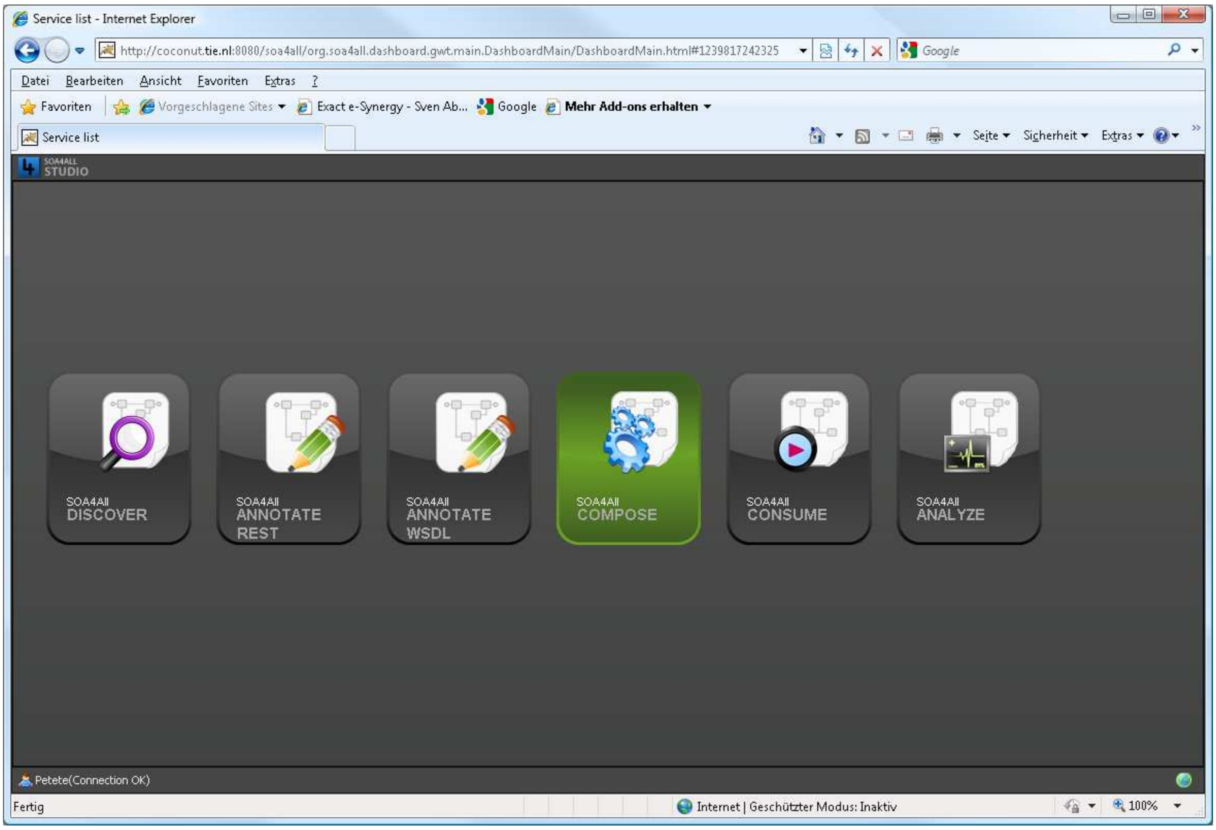
Afterwards, copy the `soa4all-dashboard.war` file, which you find in the `bin` directory of this ZIP file into the folder `webapps` of your Tomcat installation.

Once you have done this, please start Tomcat. This will automatically install all SOA4All files for you. After installing the SOA4All prototype, open a web browser and navigate to the following URL:

http://localhost:8080/soa4all-dashboard

This will show you the welcome screen of the SOA4All Dashboard application, which allows you to access the SOA4All studio.

For any questions, please refer to sven.abels@tieGlobal.com



Service list - Internet Explorer

http://coconut.tie.nl:8080/soa4all/org.soa4all.dashboard.gwt.main.DashboardMain/DashboardMain.html#1239817242325

Petete(Connection OK)

Fertig

Internet | Geschützter Modus: Inaktiv

Figure 9: The D2.4.1 Dashboard