Smilehouse Workspace 1.12.2

Whitepaper

Secure, scalable and platform independent e-commerce software for integrated e-business solutions

Document Info

Document type:	Technical document
Creator:	Smilehouse Workspace Development Team
Last updated:	17.08.2009

Address
Itälahdenkatu 22 A
00210 Helsinki
Finland, Europe

Phone & Fax
+358 – 9 – 25 122 10
+358 – 9 – 25 122 119

Internet
web: www.smilehouse.com
e-mail: info@smilehouse.com

Table of Contents

1. Introduction

1.1. Overview

Workspace is a packaged e-commerce software for integrated e-business solutions. It includes all the basic features required to build and run professional webshops, online stores and extranets. Workspace is a suitable platform for both business-to-consumer and business-to-business solutions.

Workspace is easy to install with the installation wizard and can run on a 100% open source platform. Using for example Linux, Tomcat and MySQL enables building e-commerce solutions with low total cost of ownership (TCO). Also commercial platforms are supported. Workspace is based on J2EE architecture and is continuously developed and supported by Smilehouse.

The Workspace administration interface is easy to use. Through the web-based administration interface the merchant can manage content, configure the web shop layout and functionality and run the day-to-day business activities. Workspace administration Interface offers a language choice of English, Finnish, Estonian, German, Dutch, Russian, Swedish, Spanish and French.

Workspace can be integrated to back-end systems by using its Web Services API. Workspace also has a ready-made connection to OpenSyncro (www.opensyncro.com) integration toolkit to further ease developing integrations. Also developing custom functionality is possible using Workspace SDK.

Security plays an important role within e-business. Workspace has been thoroughly tested by independent auditors to ensure a high level of application security. Security testing is also a standard part of our software development process.

This document is targeted for partners, developers and CTO´s interested in Workspace technology, architecture, security and other technical issues.

1.2. Necessary skills

For managing Webshop content, product data, customer data, order information, payment settings and delivery options no special skills are needed.

For configuring Webshop graphics and features you don't need any special skills since Workspace provides easy-to-use tools for managing the shop's features and layout. However, some experience in web design may turn out to be useful.

For developing real-time integrations using the provided Web Services API, basic knowledge about XML and integration tools are required. Integration capabilities of Workspace are discussed in chapter 6.

For installation of Workspace, including the other required software (**see Requirements in section 3.2**), you need basic system administration skills. It should take about 1-2 hours to install and configure the database (MySQL), Java 2 Standard Edition, a servlet container (Tomcat) and Workspace.

1.3. Different Editions

There are four different Workspace Editions serving the needs of different user groups. In addition there is Workspace Developer Edition (Free download) for testing and evaluation purposes.

WORKSPACE Small Business Edition

Workspace Small Business Edition is free. It is targeted for small companies starting their e-commerce projects. It is suitable for building a basic consumer webshop. It is also available as a free hosted service at http://wosbee.com/.

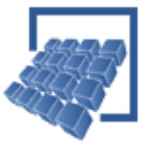
WORKSPACE Standard Edition

Workspace Standard Edition is the software platform for advanced and integrated e-business solutions. It enables building both B2B and B2C solutions. There is also multi-language support and support for international VAT handling.

WORKSPACE Enterprise Edition

Workspace Enterprise Edition offers additional features especially for large enterprises. It enables connecting multiple Standard Editions into one entity. For hosting there is support for commercial platforms and clustered environments.

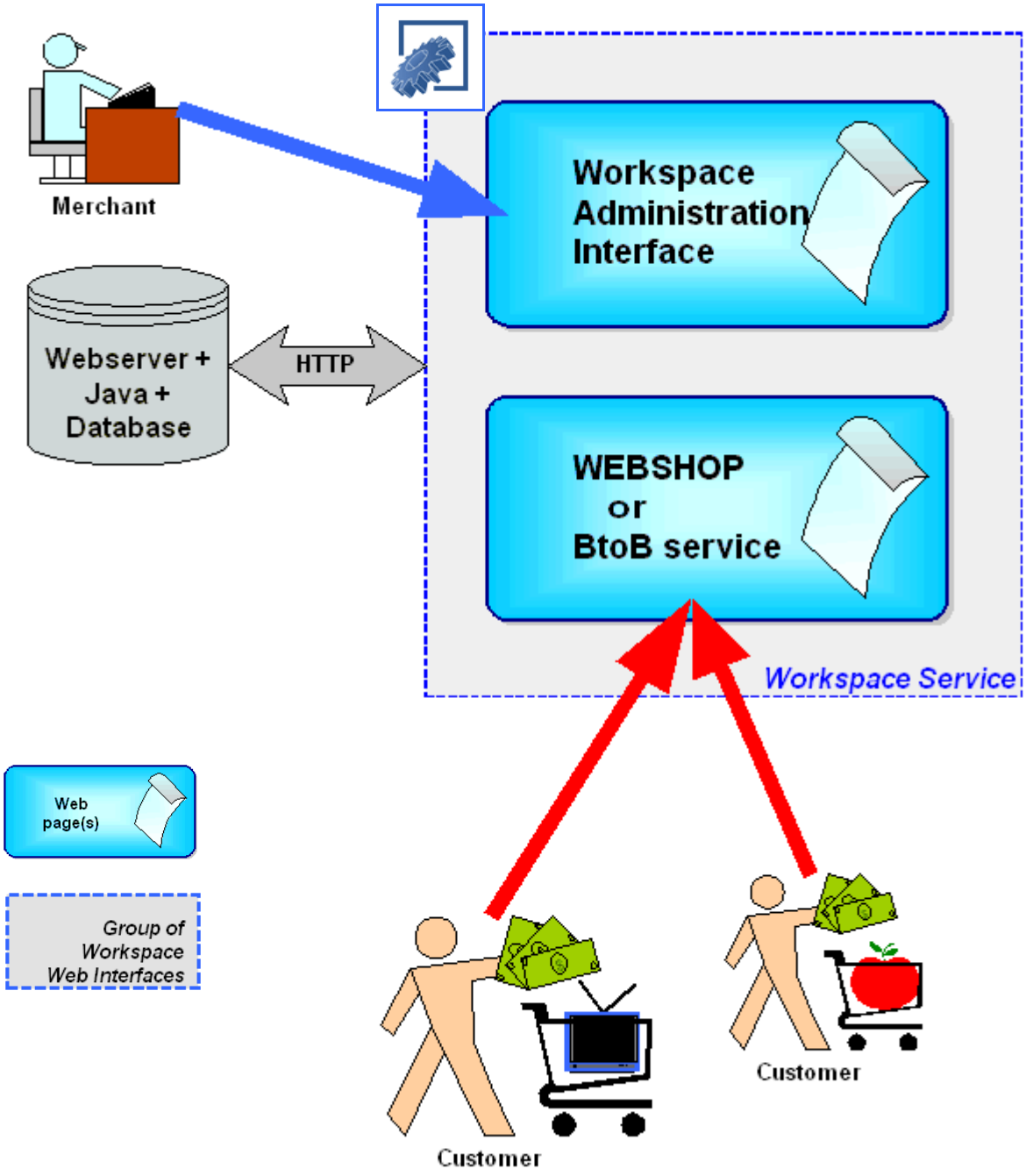
WORKSPACE Operator Edition



Workspace can also be used in Operator mode. Operator Edition is targeted for hosting providers and enables hosting multiple Workspace Small Business and Standard Editions in one server environment.

In Standard and Small Business Editions merchants take care of sales related processes, such as handling orders, updating product information, etc. Customers use the Webshop for purchasing products from the merchant.

Figure 1. Standard and Small Business Editions (simplified picture)



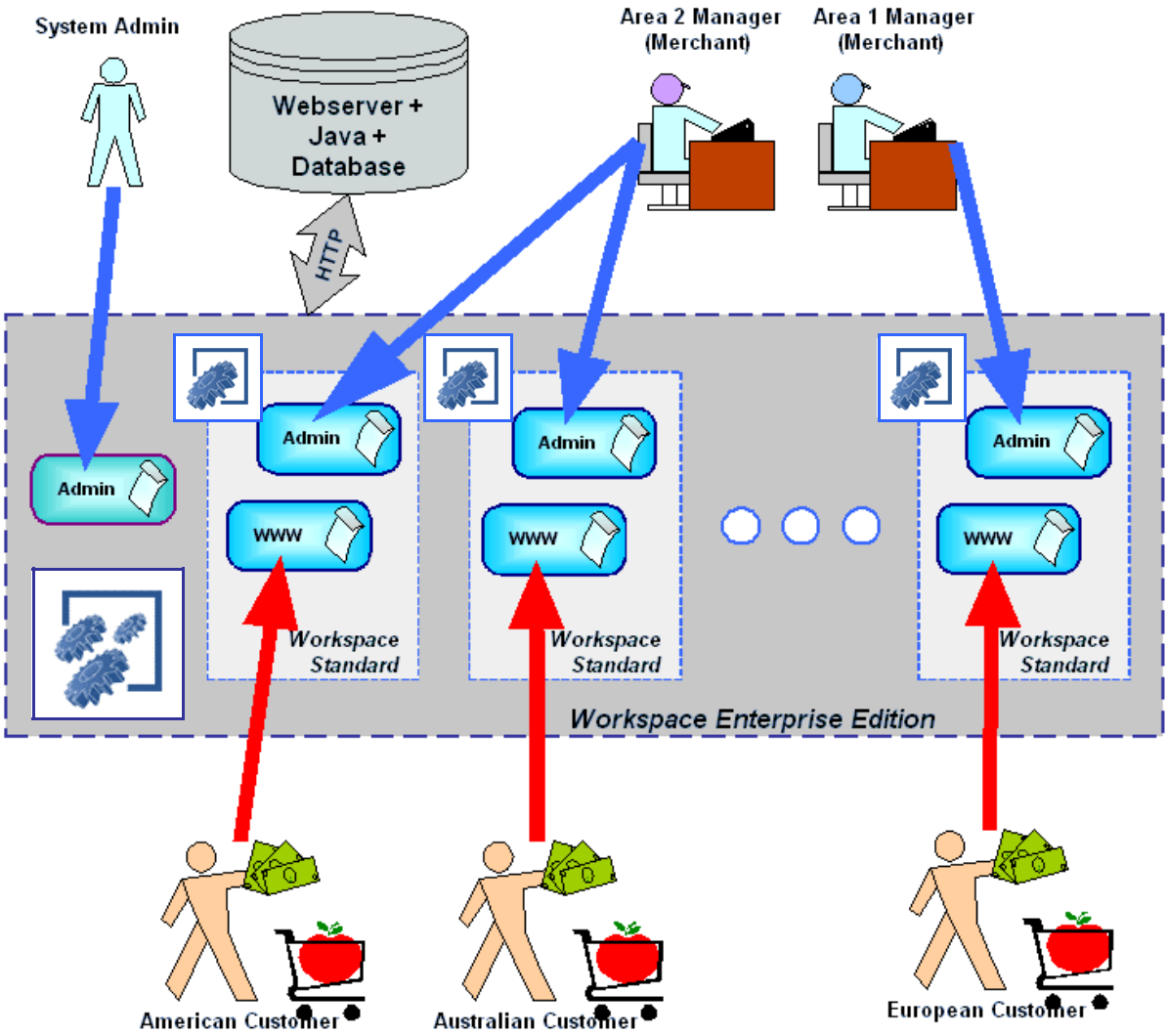
Address
Itälahdenkatu 22 A
00210 Helsinki
Finland, Europe

Phone & Fax
+358 – 9 – 25 122 10
+358 – 9 – 25 122 119

Internet
web: www.smilehouse.com
e-mail: info@smilehouse.com

In Enterprise Edition multiple Standard Edition installations can be connected to a single interface. That interface is used by the Merchant for example for handling orders. See figure 2 below.

Figure 2. Enterprise Edition (simplified picture)



Red = Local Customers visiting their local shop.
Blue = Area/Country Managers (merchant) conducting local business actions.
Blue = System Administrator doing various system operations.

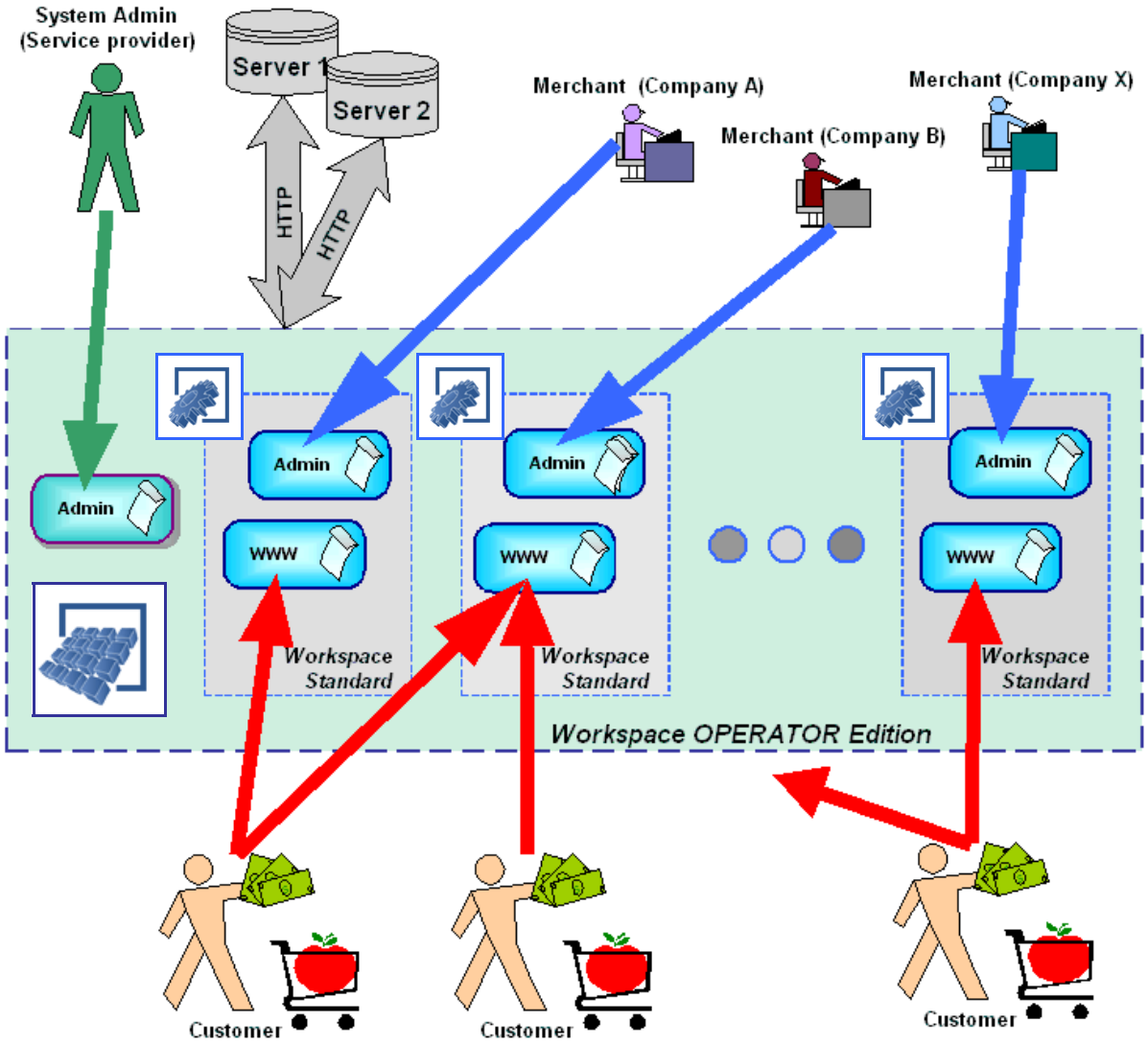
Address
Itälahdenkatu 22 A
00210 Helsinki
Finland, Europe

Phone & Fax
+358 – 9 – 25 122 10
+358 – 9 – 25 122 119

Internet
web: www.smilehouse.com
e-mail: info@smilehouse.com

In Operator Edition you are able to host multiple Workspace Standard and Small Business Editions on a single server. The Operator Toolkit server tools make it easy to create, update and remove shops. It is also possible to automatize the creation of the shop, so you can create it directly from your web browser. See figure 3 below.

Figure 3. Operator Edition (simplified picture)



Green = Service provider's administrator using Operator Toolkit.
Red = Customers visiting shops.
Blue = Merchants doing various managing operations in their shops.

Address
Itälahdenkatu 22 A
00210 Helsinki
Finland, Europe

Phone & Fax
+358 – 9 – 25 122 10
+358 – 9 – 25 122 119

Internet
web: www.smilehouse.com
e-mail: info@smilehouse.com

2. Technical overview

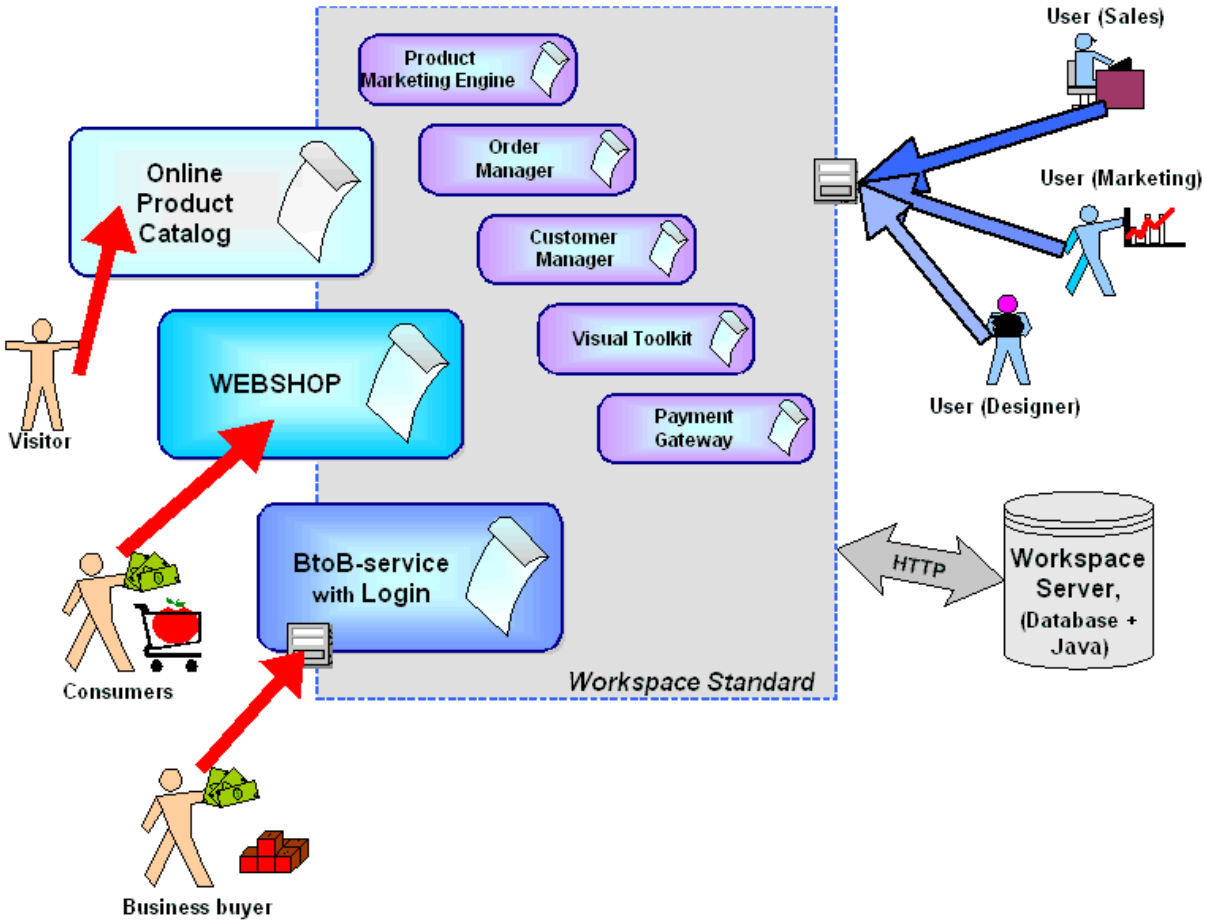
2.1. Architecture

Workspace software has four main interfaces. Two browser-based interfaces for users and two software API's for back-end and front-end integrations.

- For users, the application uses browser-based interfaces to enable secure use of the software through the internet, regardless of the user's location. The application modules and the basic architecture are shown in Figure 4 and discussed in chapter 2.4. The browser-based interfaces in Workspace are separated into two main parts:
  - *Web shop* itself which can be either open to all visitors or require customers registration. A single web shop can have multiple interfaces for customers with different functionality and content.
  - *Workspace Administration Interface* can be accessed only by authorized personnel (the Merchant) and is used for building and managing the Web shop. The Administration Interface is divided into multiple application modules which can be independently activated based on the company needs.

- For system integrations, Workspace has two interfaces. The interfaces are discussed more thoroughly in chapter 6.
  - The Back-end API (Open Interface) is a standard web services API and is used to connect Workspace databased to any back-end systems or 3rd party services. It enables transferring product, order, customer and pricing data.
  - Tho Front-end API (WebAPI) is an REST API using JSON notation. It enables both adding Workspace functionality (e.g. shopping cart, product data, order process) to other web sites and managing web shop UI with Ajax.

Figure 4. Application modules and simplified architectural use-case diagram



2.2. Workspace and corporate IT architecture

By using the built in integration API's Workspace integrates well with both existing and future corporate IT architectures.

- Back-end integrations can be developed both as point-to-point integrations using timed batch transfer and more advanced real-time integrations using separate EAI or MQ layers. In typical heterogeneous IT systems, both types are usually used based on need and back-end capability.

- Web shop and its UI can be implemented either using the included web shop interface or adding Workspace e-commerce functionality to separate portal layer using the Front-end API.

- In additions, the platform independence of Workspace enables using commercial or open source platforms for hosting infrastructure.

Address
Itälahdenkatu 22 A
00210 Helsinki
Finland, Europe

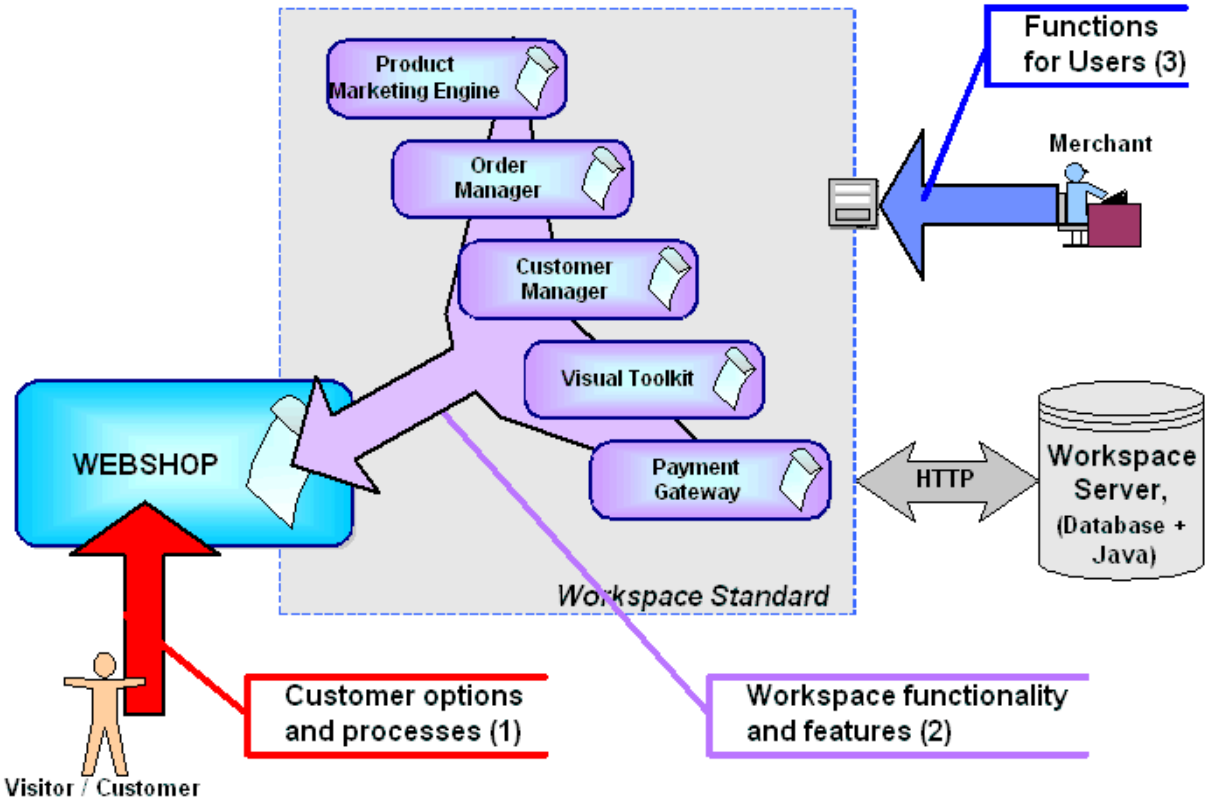
Phone & Fax
+358 – 9 – 25 122 10
+358 – 9 – 25 122 119

Internet
web: www.smilehouse.com
e-mail: info@smilehouse.com

2.3. Functional overview

Below is a conceptual Figure of the application modules and user interfaces in Workspace. It also shows the general information flow in Workspace.

Figure 5. Module information flow diagram with typical actions/features



1. Customers can...

- search, sort and view products and categories
- get specific product information
- check inventory levels
- add products to shopping basket
- place and pay orders
- check order status (if integrated to the delivery systems)
- save shopping baskets for re-use
- change his/her personal information in the system

2. Merchants can...

- check and confirm orders
- manage product database and visibility
- manage customer data
- define and edit service outlook and functionality.

3. Workspace functionality includes...

- product data management
- product data publishing
- shopping basket functions
- order and customer handling
- status based order automation
- service web-outlook management
- file management
- payment systems
- integration interface support

2.4. Functional modules in administration interface

2.4.1. System

The users of the Administration Interface can be managed in the System module. The user who has access to this module can set the access rights of all the other users. A user can be given rights just for some parts of the software and only these parts will be visible to him when he logs into the Administration Interface.

Also some other system functions such as setting the overall information of your company and clearing the customers, products, orders and statistics from database are located in the System module.

2.4.2. Products

The management of product information takes place in the Products module. It offers powerful and easy to use editing capabilities for products and product group hierarchies. There is also possibility to define different VAT settings allowing to list and update VAT rates specific to regions and categories.

2.4.3. Orders

Orders module contains a searchable list of all the orders made either in the Webshop or created manually in the Orders module itself. Most of the information contained in an order including the info given by the customer and the ordered products can be edited here.

Also a fully customizable status describing the current state of the order can be set. Statuses can be configured to be set automatically triggered by various events such as receiving a confirmation of a successful payment from an online payment system. Also sending of customizable email messages to the customer or the merchant can be set to be triggered by these events. There is possibility to use Prinetti integration.

2.4.4.Customers

The customer database can be created and extended automatically by customers registering themselves, importing customer records from an existing system or by managing the database through the Customers module.

Customers can be arranged into customer groups and mass emails can be sent to chosen groups. Also the Webshop appearance, content and pricing can be customized for different customer groups.

2.4.5.Marketing

With the Marketing module you can see web shop statistics, create discount coupons and send group e-mails.

2.4.6.Visual

With the Visual module you can edit the Webshop themes. A theme consists of pages and settings that contain all the information related to the Web shop appearance and structure.

The pages are constructed from customizable ready-made *Workspace Action Elements (WAEs). i.e. Search, Shopping Basket, Order Form, etc.* Visual module also has a FileManager and a feature for importing and exporting themes.

You can have up to ten different themes and while some themes are in use, others can safely be edited without affecting the Webshop.

2.4.7.Payment & Shipping

Payment module consists of multiple ready-made implementations for different ePayment vendors. These include many Nordic and Estonian banks as well as some international payment methods such as PayPal, Google Checkout, Click&Buy, Itransact, Ogone, DIBS, 2CheckOut and ChronoPay.

Detailed information about payment transactions are included. Every transaction is connected to an order made in Webshop. Also Identification Gateway settings and ready-made implementations for Finnish TUPAS standard based identification vendors are included here.

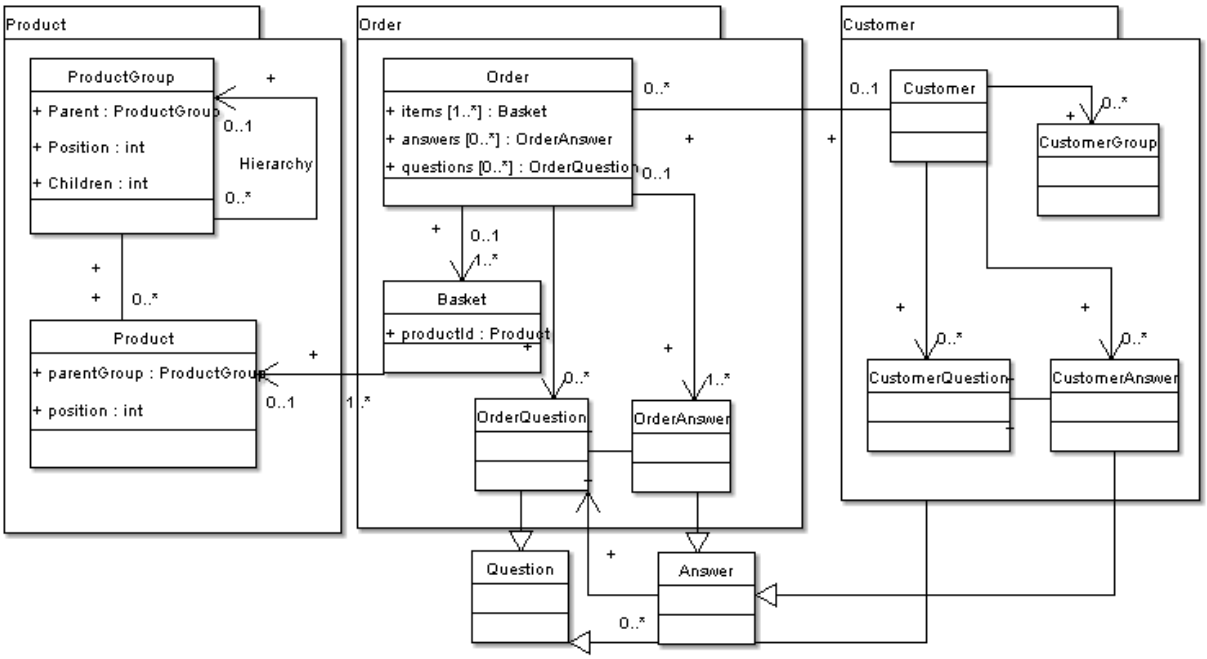
Payment module also contains fully customizableShipping and delivery options with individual costs. Delivery costs are fully customizable and allow to create rules based on order value and/or weight.

2.5. Data model

2.5.1. Main business objects

Workspace contains three main business objects: Products, Orders and Customers. Of course there are many supporting objects relating to these concepts and many other objects relating to other system services. Here only the main objects are introduced to illustrate business information stored by the system. Below is a figure that shows the main objects and their associations.

Figure 6. Workspace business objects and associations diagram



Products

Products are contained in ProductGroups which are organized in a hierarchy. Each Product has its own unique ItemCode.

The analogy of a file system is pretty close, ProductGroups are folders and Products are files. Products and ProductGroups can be copied or transferred to different ProductGroups. Also links to Products can be created in multiple ProductGroups.

One difference to a regular file system is that Products and ProductGroups always have a definitive order specified by a position attribute. The order can be changed manually or by tools that sort the positions according to some attribute, e.g. a product's price.

ProductGroups hierarchy is based on a Parent-Child relationship where each ProductGroup contains a list of its children (other ProductGroup objects) and possibly a parent. In this hierarchy leaf nodes don't have any children and root nodes don't have a Parent. Products can belong to multiple ProductGroups and a ProductGroup can have only a single parent. Thus in the file system analogy the Product package supports links.

Orders
Order contains items which represent the ordered Products. An item is represented by a Basket object. A Basket object is created when a customer adds a product to the shopping cart in the Webshop.

The Product's most important information like itemcode, product name, chosen product variation and determined price are copied to the Basket object on creation. These details are omitted from the diagram and are better explained in the Product Pricing Scheme chapter.

An Order also contains information asked from the buyer during the order process. This information is contained in a group of related Question and Answer objects. The Questions are specified in the Administration interface and contain predefined Question types like the 'Payment method'-question. Also custom Question types are supported.

This type of data model was chosen to implement the order information in a very customizable and extensible way.

An Order can also contain a reference to a Customer who made the order. This reference is created automatically if the Customer was logged in to the shop when he/she made the Order. The reference can also be added manually in the Administration Interface.

Each Order is connected to Payment Transaction, created during the ordering process.

Customers
Customers represent identified users of the Webshop. Customers have usernames and passwords by which they identify themselves in the shop. Customer information is represented in the same way as Order information by Question and Answer pairs. The Customer and Order Questions can also overlap so when the Customer creates an Order, his/her Order information is automatically filled.

Customers can belong to multiple CustomerGroups but only one of those groups is the customer's primary CustomerGroup. CustomerGroup can specify a shop theme that will be used for the Customer (and is accessible only to the Customers belonging to the CustomerGroup if the theme is set as password protected).

CustomerGroup can also specify a PriceGroup that will be used when determining the Product prices for a customer that belongs to the CustomerGroup. PriceGroups are used to specify discounts to a set of Customers. More information about this is in the Product Pricing Scheme chapter.

2.5.2.Product Pricing Scheme

Products have a price attribute that defines the Product's default price. The default price is used in situations when there are no PriceRules to apply. PriceRules are a way to implement different pricing of Products in different situations.

A PriceRule can change the default price by:

1) Overriding the default price with a new price. e.g. 154

2) Adding or subtracting a static amount from the default price. e.g. -15, +34

3) Adding or subtracting a percentage of the default price. e.g. -10%

ProductVariator-object is responsible for determining which PriceRules apply and therefore determines the Product's price in different situations. The input used by the ProductVariator is the Product Ordered Amount, possibly a Customer object and possibly the selected options from the Product.

Customer object can belong to a CustomerGroup and the CustomerGroup can specify a PriceGroup that should be used with the Customer. If a customer is known the ProductVariator applies the specified PriceGroup's PriceRule to the Product's default price.

Also the chosen options can effect the price when the user is adding the Product to the Basket. Products can have ProductOptions (such as size) and each ProductOption can have multiple predefined Choices (like S,M,L,XL). Each of these Choices is represented by a ProductOptionChoice object. A choice can specify a PriceRule, for example size S jeans could be given a 10% discount.

Contract pricing feature makes it possible to set overriding price rules for customers or customer groups on a product basis. It is possible to e.g. set different product prices or discounts for each customer.

Also quantity based PriceRules can be appended. The PriceRule is applied, when the amount of products ordered is equal to or greater than the quantity threshold.
On an addToBasketAction the ProductVariator uses the chosenOptions as input to possibly apply the PriceRules related to the choices.

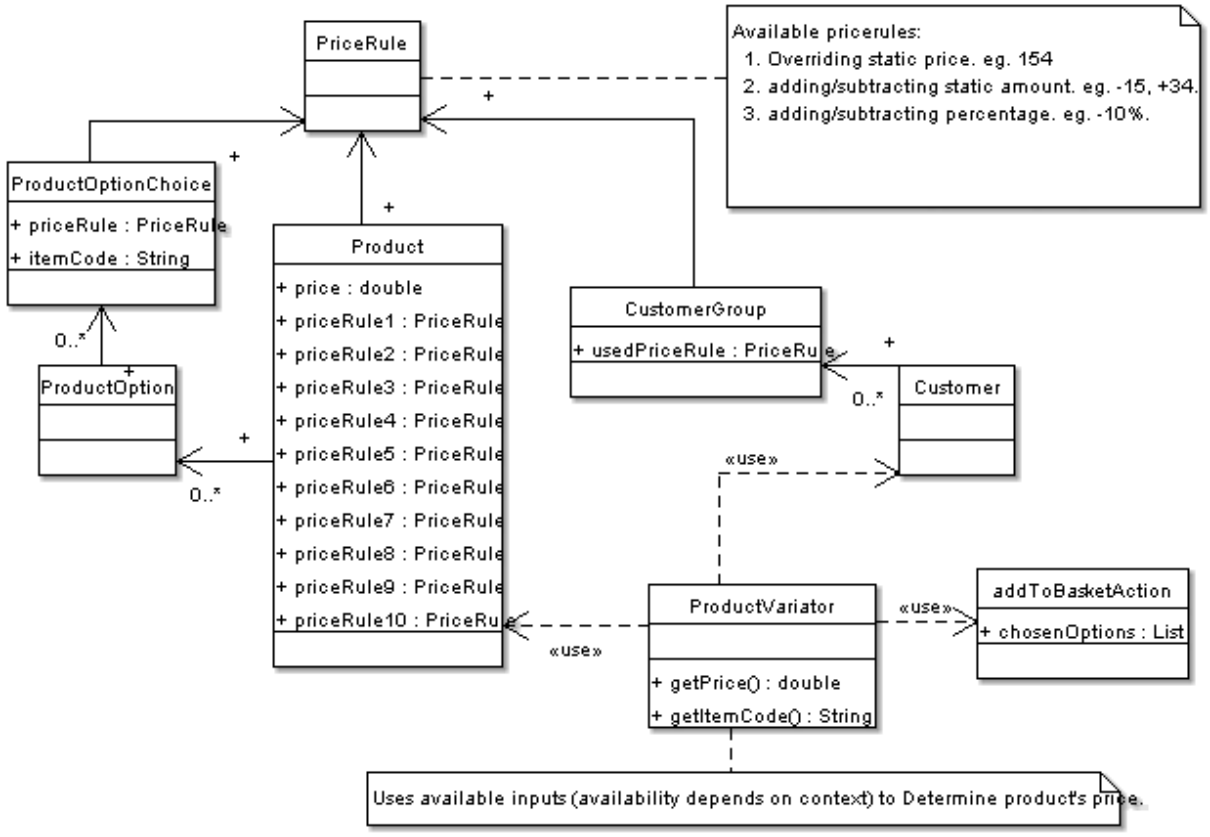
A ProductOptionChoice can also override the Product's itemcode and contain a separate inventoryAmount. This way a group of products can be represented by a single Product-object in the Webshop.

The order in which pricing rules are appended is following:

1. Contract pricing rules
2. Product Pricegroup rules
3. Product option PriceRules
4. Quantity based PriceRules

Below is a diagram showing objects and associations related to the pricing of Products.

Figure 7. Product Pricing Scheme diagram

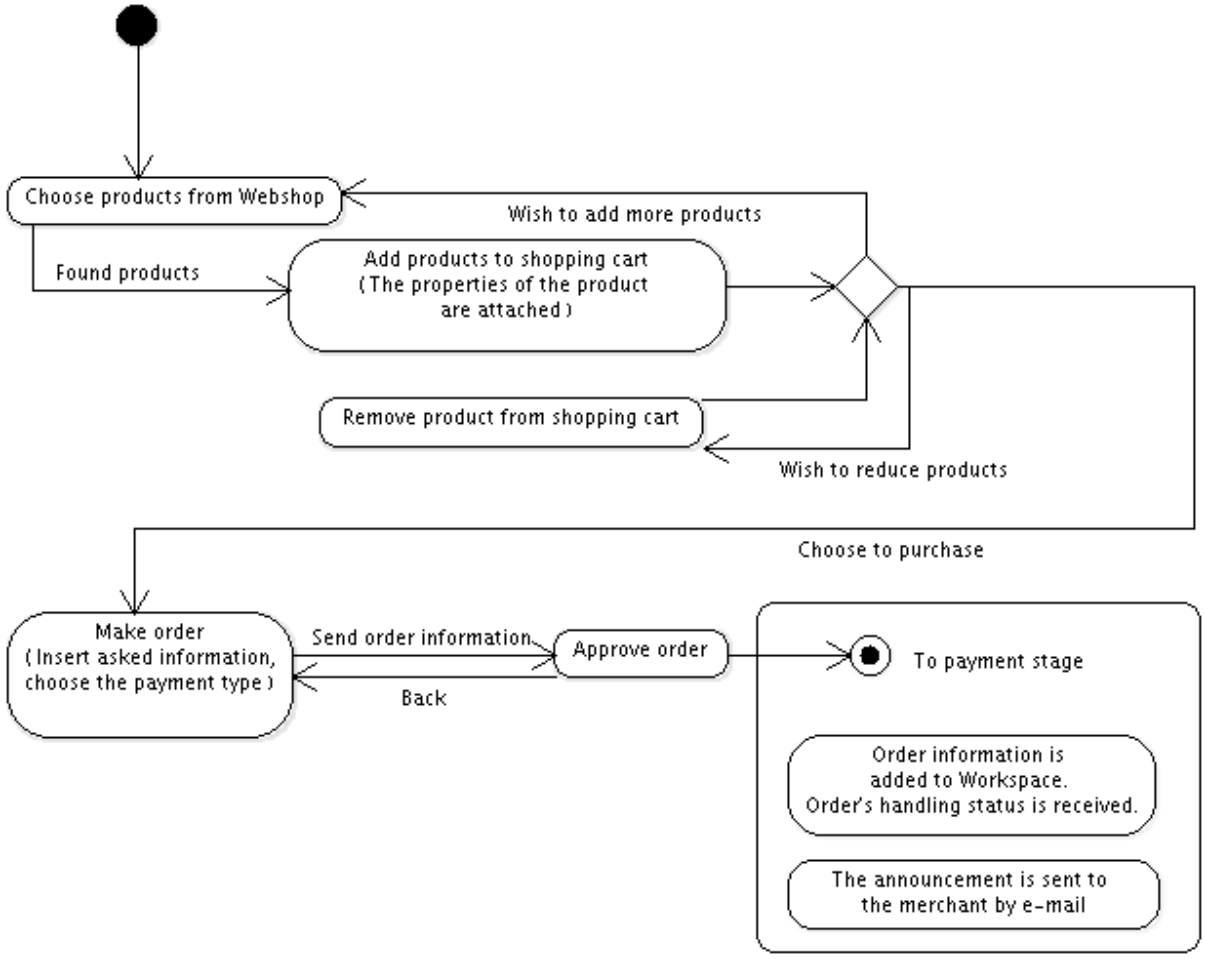


2.6. Ordering process in the Workspace system

Below is an UML Activity diagram of the ordering process in the Workspace system. Note that order's handling and payment statuses are fully customizable and their functionality described here is only the default functionality used in a fresh Workspace installation.

The customer browses the Web shop products. When the customer finds a product he/she would like to order, he/she can add the product to the shopping cart. If the Product has options, e.g. color, the chosen options (e.g. green) are now attached to the item in the shopping cart and cannot be changed. The Items can be removed from the shopping cart though and their amounts can also be changed.

Figure 8. Choosing the products and ordering



When the customer chooses to purchase the contents of the shopping cart he/she moves to the order form by pressing the button "order products". Here the customer fills the form with the required information. Typically customer's name and postal information is asked and the type of payment and delivery.

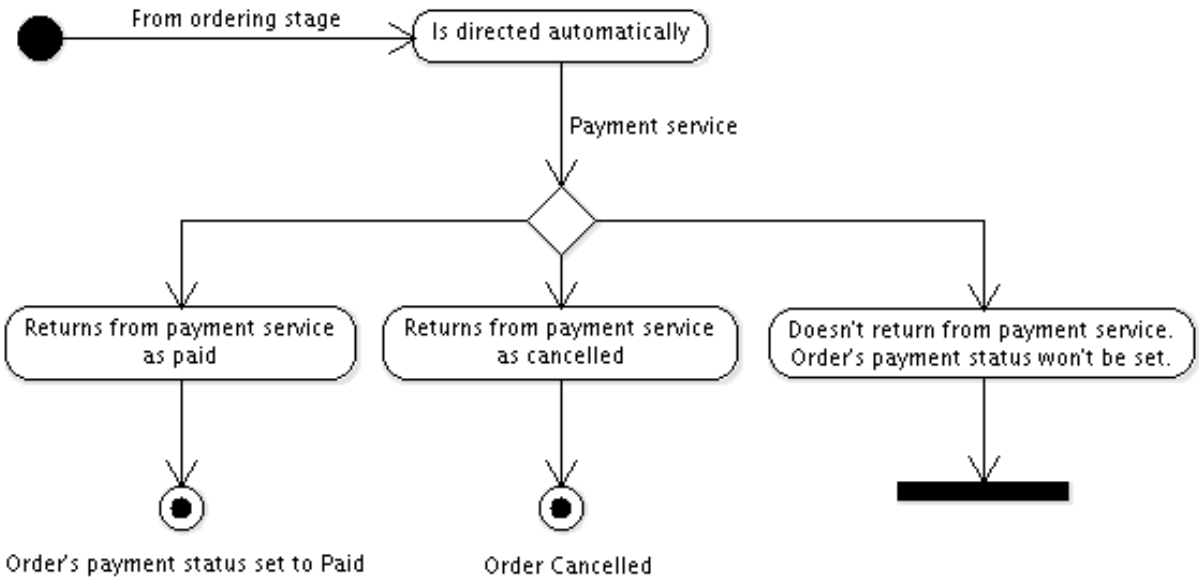
After sending the information asked in the ordering form the customer moves to an approving page. Here the customer can see the information of the order and can return to change that information by pressing the back button of the browser or approve the order and continue to the payment stage.

When the customer approves the order, the system:

1) creates a new order and new payment transaction related to the order.
2) sends an order creation note to the customer's email address (customizable)
3) sends an order notification to the merchant's email address (customizable)
4) redirects the customer to the chosen payment service

The order is generated by using the information given in the order form and the contents of the shopping cart. If the customer was logged in to the shop when the order was created the system also automatically added the reference between the customer and the order. After the order has been created, its initial handling status is 'Received' and the Payment status isn't set at all (default behavior, customizable).

Figure 9. The payment process



After the order is created the customer is automatically directed to the chosen payment service. From the payment service the system can basically receive two kinds of acknowledgments. If the payment is approved the customer will be shown a thank you page and the order's payment status will be set to "paid".

Automatic email messages can also be sent when the payment is accepted. If the payment is canceled the order handling and payment statuses change to "canceled" and the system hides the order. The information of the order is still stored in the system's database for the case a payment interface malfunctions, but usually this information isn't required. The order canceled page is shown to the customer.

Payment Transactions management provides detailed information about every order made, included canceled ones. It is also possible, that the system doesn't receive any acknowledgement from the payment service. This happens if the customer doesn't return from the payment service with the link provided in the payment service.

Reason for this can be customer's low experience related to online payment systems or some other unexpected situation, like the customer's browser crashing. In this case the system doesn't get any information of the payment state, so the payment status stays empty. The Administration interface includes a feature that can be used to check the payment status related to an order with a push of a button (if the order's chosen online payment system supports this feature).

3. Platform compatibility and deployment options

3.1. General

This section describes supported and recommended platforms and browsers. Scalability issues are discussed also.

3.2. Supported platforms

Smilehouse Workspace officially supports the following server side software.

Operating System	Debian GNU/Linux (x86)
JVM	SUN Java 1.6
Application Servers	Apache Tomcat 5.5.x Apache Tomcat 6
Databases	Mysql 5.x PostgreSQL 8.x

Although not officially supported, our clients have also used in production environments the following configurations

Operating Systems	Redhat Linux (x86) SUN Solaris (sparc, x86) Windows 2000, 2003, XP, Vista Mac OS X
JVM	SUN Java 1.5.x
Application Servers	JBoss 4.0 Resin
Databases	Oracle

3.3. Supported browsers

Smilehouse Workspace (Administration Interface and Webshop) has been tested with the following browsers:

Microsoft Internet Explorer Windows (from IE 6)
Mozilla Firefox (any version)

Both interfaces require the following browser options:

Cookies
SSL (if configured in the application server)

Administration Interface requires also the following browser options:

Javascript
SSL (if configured in the application server)
Windows popup blocker should be disabled

3.4. Scalability

Basic server installation includes:

2 x Intel Xeon 3 GHz processor
4 GB DDR RAM
2 x 80 GB SCSI HD (RAID-1)
100 Mbps LAN

Basic server installation supports about:

100+ Workspace webshops
10 000+ products per organization
10 000+ registered customers per organization
50 000+ orders per organization
50 000+ visitors / day (in total)

Naturally Workspace can be installed also to clustered environments for high availability and fault tolerance.

4. Language support

4.1. Language support in the Webshop

Smilehouse Workspace uses the UTF-8 encoding so many languages work fine with it by default. Tested languages are:

- English
- Finnish
- Swedish
- Danish
- Estonian
- Dutch
- Russian
- German
- Italian
- French

By using multiple themes, a single Workspace Standard installation can be used to create a webshop with multiple languages. All webshop themes can use different language.

You can provide for example a Finnish and Swedish view to the same shop with same products. All messages (e.g. "Your shopping cart is empty") are theme based and can be localized.

Usually the need for multiple languages is based on the need to serve customers in different countries. In this case, in addition to content language, you probably need to customize delivery options and charges, payment methods and taxation. You can implement these by configuring different Themes for different target markets. Workspace Standard also offers tools for international VAT handling.

For advanced global e-business we recommend using Workspace Enterprise as it enables connecting multiple Workspace Standard webshosp as a single entity and thus enables more configuration options to each country/area specific webshop.

4.2. Language support in the Administration Interface

Workspace currently supports English, Spanish, Finnish, Estonian, German, Dutch, French, Swedish, Russian and Vietnamese languages in the Administration Interface. Java resource bundles are used for the language support, so adding further languages is possible.

5. Payment interfaces

5.1. General

Workspace has ready-made integrations to many payment systems. New systems are added continuously. The Merchant can manage payment systems used in the webshop through Payment Gateway module in administration interface.

5.2. International payment systems

PayPal

PayPal accepts world's major credit cards such as Visa, MasterCard, American Express, Discover and Amex. Customers do not need active PayPal account, if paying with credit card. Current version of PayPal IPN (Instant Payment Notification) integrated into Workspace is 1.7. IPN uses Workspace order id's to identify the payment request.

Workspace merchants direct their customers to PayPal payment form, where customer confirms payment by giving his/her credit card info. IPN then posts a notification to Workspace about the payment. Notification includes all payment information and an encrypted code.

When Workspace receives the notification, payment information with that code will be sent back to PayPal for integrity check. IPN then authenticates the transaction and sends a confirmation about its validity to Workspace.

Google Checkout

Google Checkout is actually not a payment method but an entire checkout flow. When Google Checkout is used, Workspaces own order form is not used at all. Workspaces order management is integrated with Google Checkout so that all the order management can still be done in Workspace. Currently Google Checkout can only be used by merchants located in United States and all the prices must be in US dollars.

Ogone

Ogone e-Commerce is international payment platform supported in countries like Belgium, France, Netherlands, Austria, Germany, Luxembourg, Switzerland, and United Kingdom. Ogone accepts a large range of methods of payment, including bank cards and NetBanking systems and it is compatible with SSL3, Verified-By-Visa and SecureCode. Ogone offers merchants a management module, with secure access, where they can manage their transactions and configure their account.

DIBS Payment Window

DIBS is a payment service provider in Scandinavia, certified by banks and credit card aquirers. With DIBS you gain access to more than 30 different payment types included all leading credit cards. For example following bank cards and credit cards are supported by DIBS: Dankort, Eurocard/Mastercard, VISA, Diners Club, American Express, Maestro, VISA Electron, JCB, Forbrugsforeningen and many more. DIBS has been PCI certified by VISA and MasterCard since 2005 and it supports 3D Secure transactions.

ClickandBuy
ClickandBuy is one of the leading payment systems on the internet. More than 7,000 merchants benefit from this complete service in 26 countries, which includes 21 different currencies and 43 different payment methods. ClickandBuy is used by Apple iTunes, Skype, msn, T-Online, Electronic Arts, Meetic, Playboy, SanDisk, Yamaha, UNICEF and many more.

2CheckOut
2Checkout (2CO) is an authorized online reseller for over 40,000 international web sites offering a wide range of shipped products, downloadable products (such as ebooks and software), and online services (such as web hosting and site memberships). 2CO's proprietary e-commerce technology supports back-office functions including financial reporting, tracking, fraud prevention, customer service and sales tracking.

ChronoPay
ChronoPay is international multi-currency credit card payment platform integrated with reliable European acquiring banks and processors. Chronopay supports all major credit cards and bank ATM Debit cards.

ITransact
ITransact credit card payment doesn't have a version number included in its payment implementation. Credit cards supported by ITransact are Mastercard, Visa, American Express, and Diner's. ITransact credit card CVV Card Verification Value check has been enabled when integrated into Workspace to increase security of payment and reduce possibility of fraud.

In order to use RediEFT (electronic funds transfer), you must have an account with First American Payment Processing (FAPP), a trusted EFT processor. Workspace merchants direct customers to ITransact payment form where credit card information can be submitted.

All connections between ITransact, Workspace and customer are SSL encrypted, and no credit card information is passed back to Workspace. After customer data (credit card number or checking account number) arrives at iTransact's server, it is re-encrypted using PGP (RSA algorithm), which makes the information unreadable.

Order's reference number and timestamp are used to check payment integrity when the customer returns to the Workspace Webshop.

Kreditor
Kreditor is international payment provider. Kreditor supports pay by bill or part payment without taking any risks or increased administration for the Merchant.

Since Kreditor does not provide its own web-based interface, Workspace provides its own web-based proxy for use with Kreditor payments. A proxy is included with the default installation, but any valid Kreditor proxy may be used if available. However, any proxy used should be served via SSL encryption, as the purpose of the proxy is to accept sensitive customer information, such as Social Security Number.

Kreditor is supported in countries like Finland, Sweden, Norway and Denmark.

5.3. Finland

The payer/customer using Finnish payments can be the bank's retail or corporate customer, who has the necessary service keys for the bank's service. In order to use the bank's Web Payment, merchant should have an agreement with the bank.

Account transfer from the customer's account to that of the Workspace merchant's account is immediate and instantly visible among the Workspace merchant's transactions. All Finnish payment interfaces use a 32-character hexadecimal MD5 hash code to check the integrity of payment transactions.

The process of payment in Workspace includes sending a payment request with a payment form to the bank. The request includes all the information concerning the Workspace merchant and the customer's order. Along with the information sent to the bank, is the MD5 hash code, which is in form of 32-char hexadecimals, which the bank uses to check the integrity of the payment request.

If all the information is accurate, bank sends the customer payment request - customer will be redirected to the bank's payment request identification form. After a successful identification and account transfer, the customer and transaction information, protected by a MD5 hash code is transmitted back to Workspace.

Workspace checks the integrity of received information using the MD5 hash code and some additional parameters returned from bank. Order will be cancelled, if payment information, including customer's order, received from the bank doesn't match the order in progress in Workspace, otherwise order status in Workspace set to paid. Within payment process also transaction created for every single order. Transaction gives detailed information about ordering process and order statuses.

Luottokunta credit card payment

Luottokunta accepts MasterCard, Visa, Visa Electron and Amex credit cards. Current Luottokunta payment version supported by Workspace is 1.2. Luottokunta payment has been integrated with Workspace using Luottokunta's dynamic HTML web interface.

Payment form, generated by Workspace includes all order, merchant and customer information. Workspace gives each payment request its unique id, which is generated either from the order id or transaction id, based on settings in administration page. This id connects the order to the payment. Form will be sent to Luottokunta for integrity check.

If the integrity is confirmed, the customer will be redirected to Luottokunta's own payment identification form. The integrity check uses a 32-character hexadecimal MD5 hash code to confirm the payment request. Workspace generates a hash code based on the order information sent to Luottokunta.

No credit card information is passed between Workspace and Luottokunta.
When returning from Luottokunta to Workspace, Luottokunta passes an MD5 hash code (LKMAC) among other order information. Based on that hash code Workspace checks the integrity of the payment information returned and if the payment transaction succeeded.

Nordea e-payment
Current version of Nordea e-Payment integrated with Workspace, is 0003. Security check key used in MD5 hash generation is 32 characters long and its version number given from bank is either 0001 or 0002.

Sampo Web Payment Service
Current version of Sampo Web Payment, integrated with Workspace, is 2.

OKO bank web payment
Currently the OKO bank's web payment version integrated to Workspace is 1 (parameter VERSIO). Invoicer's check version (parameter TARKISTE-VERSIO), is currently 1.

Sp/Pop, Aktia Web payment
Current version of Sp/Pop payment integrated with Workspace is 0001.

Handelsbanken
Current version of payment integrated with Workspace is 001.

Ålandsbanken
Current version payment integrated with Workspace is 0002.

Sonera Web Payment Service
Current version of payment integrated with Workspace is 0.1.

Tapiola
Current version of payment integrated with Workspace is 0002.

Nordea Nettiluotto
Current version of payment integrated with Workspace is 0001.

S-pankki
Current version of payment integrated with Workspace is 0002.

Lindorff
Current version of payment integrated with Workspace. Version is unknown since payment is still under development

5.4. Estonia

Hansabank Banklink
Hansabank Banklink has been integrated to Workspace using Hansabank's own bank link Java API 1.4.1. Workspace merchants direct their customers to hanza.net with prepared payment orders and receive confirmation from the bank concerning the success or failure of the payment order.

The communication between the bank and the customer takes place using SSL encryption. The data presented to the merchant from Hanza.net and vice versa, are electronically signed, which allows the opposite party to verify the correctness of the data presented.

The value of the digital signature used in queries is calculated on the basis of the public key RSA algorithm, VK_VERSION. The version currently used is 008. VK_MAC is submitted as the value of the query parameter in BASE64 encoding. Signature is calculated according to the PKSC1 standard (RFC 2437), using private and public keys. Public keys exchange should be made with the bank.

Order reference number and payment timestamp generated by Workspace are used to identify the payment request. When returning from bank to the Workspace shop the reference number and timestamp of the order should match the ones in process by customer.

SEB Eesti Ühispank
The SEB Eesti Ühispank payment works similarly as the Hansabank Banklink payment.

Sampo Banklink
The Sampo Banklink payment works similarly as the Hansabank Banklink payment.

Krediidipank i-bank
The Krediidipank i-bank payment works similarly as the Hansabank Banklink payment.

Nordea e-payment
Current version of the Nordea e-Payment integrated with Workspace is 0003. Security check key used in MD5 hash generation is 32 characters long and its version number given from bank is either 0001 or 0002.

Estcard
Estcard's E-Commerce Payment Gateway is service provider in electronic banking environment in Estonia. All local banks issuing payment cards are supported and being served by (Card Centre of Banks Ltd or "PKK"): Those banks are Hansabank, Eesti Ühispank, Sampo Estonia, Nordea Bank Finland Plc Estonia Branch, Eesti Krediidipank, Tallinn Business Bank Ltd.

Cards supported are: Visa, Visa Electron, Mastercard, Maestro, Diners, Amex, JCB International. All data presented to the merchant from Estcard and vice versa, are electronically signed, which allows the opposite party to verify the correctness of the data presented. The digital signature is calculated on the basis of the public key RSA algorithm.

5.5. Latvia

Hansabanka
Information for the Estonian Hansapank payment also apply to the Latvian version.

Nordea e-payment
Current version of the Nordea e-Payment integrated with Workspace is 0003.
Security check key used in MD5 hash generation is 32 characters long and its version number given from bank is either 0001 or 0002.

5.6. Sweden

Handelsbanken
Current version of the Handelsbanken integrated with Workspace is 1.

Nordea e-payment
Current version of the Nordea e-Payment integrated with Workspace is 0001.

5.7. Denmark

Nordea e-payment
Current version of the Nordea e-Payment integrated with Workspace is 0002.

Address
Itälahdenkatu 22 A
00210 Helsinki
Finland, Europe

Phone & Fax
+358 – 9 – 25 122 10
+358 – 9 – 25 122 119

Internet
web: www.smilehouse.com
e-mail: info@smilehouse.com

6. Integrations and extensibility

6.1. Back-end API (Open Interface)

Workspace provides a standard Web Services API for importing, updating and exporting data. This enables building customer specific integrations to various back-end systems
(such as logistics, pdm, financial systems and ERP solutions) and 3rd party services.

The Web Services API, **Workspace Open Interface,** supports exporting orders and importing/updating customer and product records in XML format. Web Services API can be accessed using virtually any EAI tool.

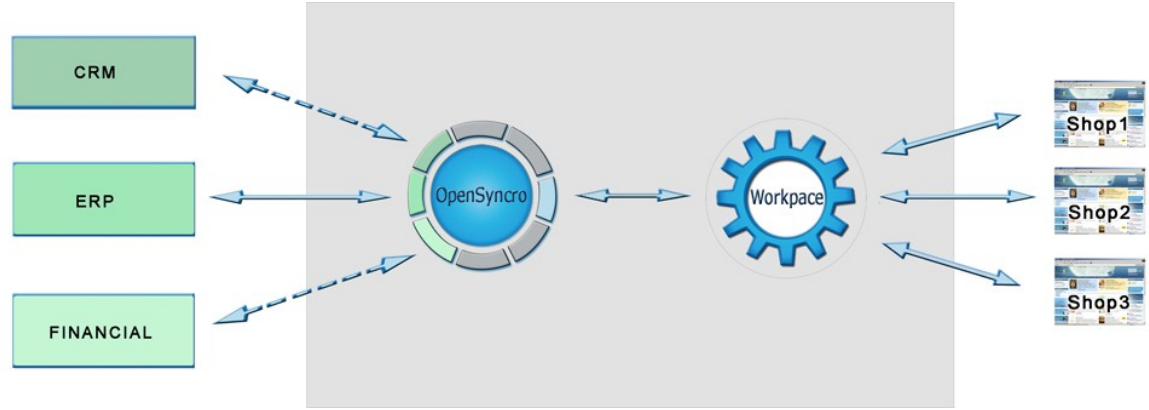
For further information, please refer to the Workspace Open Interface documentation found at workspace.smilehouse.com. Workspace Web Services API contains examples of SOAP queries for retrieving and storing data, the Open Interface WSDL description as well as the XML schemata of orders, customers and products.

6.1.1. OpenSyncro support for Back-end API

Workspace has ready-made integrations to major business applications such as SAP Business One, Exact Globe, HansaWorld, Compiere as well as to many in-house systems.

These integrations have been implemented with **Smilehouse OpenSyncro** (www.opensyncro.org) which is an application integration tool with reusable Java components for data transfers and data conversions. OpenSyncro connects to the standard Workspace Web Services API.

OpenSyncro has a Connector Pack for each business application. Due to preconfigured components and tested functionality, a Connector Pack enables integrating a business application to Workspace and other systems very quickly and easily. OpenSyncro Connector Pack for SAP Business One for instance includes all the required components to export product & customer data to Workspace and import order data to SAP Business One.



OpenSyncro comes also with built-in generic data transfer components for Workspace Open Interface, FTP and local file system as well as data conversion components for XML transformations (XSLT), CSV table to XML and a generic ASCII to XML converter based on regular expressions.

This selection of OpenSyncro components makes it pretty easy to get started with integrations; for example importing product data to Workspace from a CSV table may be accomplished in a couple of hours by a person with some experience in XSLT programming.

OpenSyncro business case

OpenSyncro is also very usable when building larger e-commerce solutions. In these solutions OpenSyncro can be used to extract product and customer data from company ERP.

This data is sent to e-commerce site built with Smilehouse Workspace. The e-commerce site is used by consumers and smaller corporate customers. Orders made in this site are integrated back to company ERP.

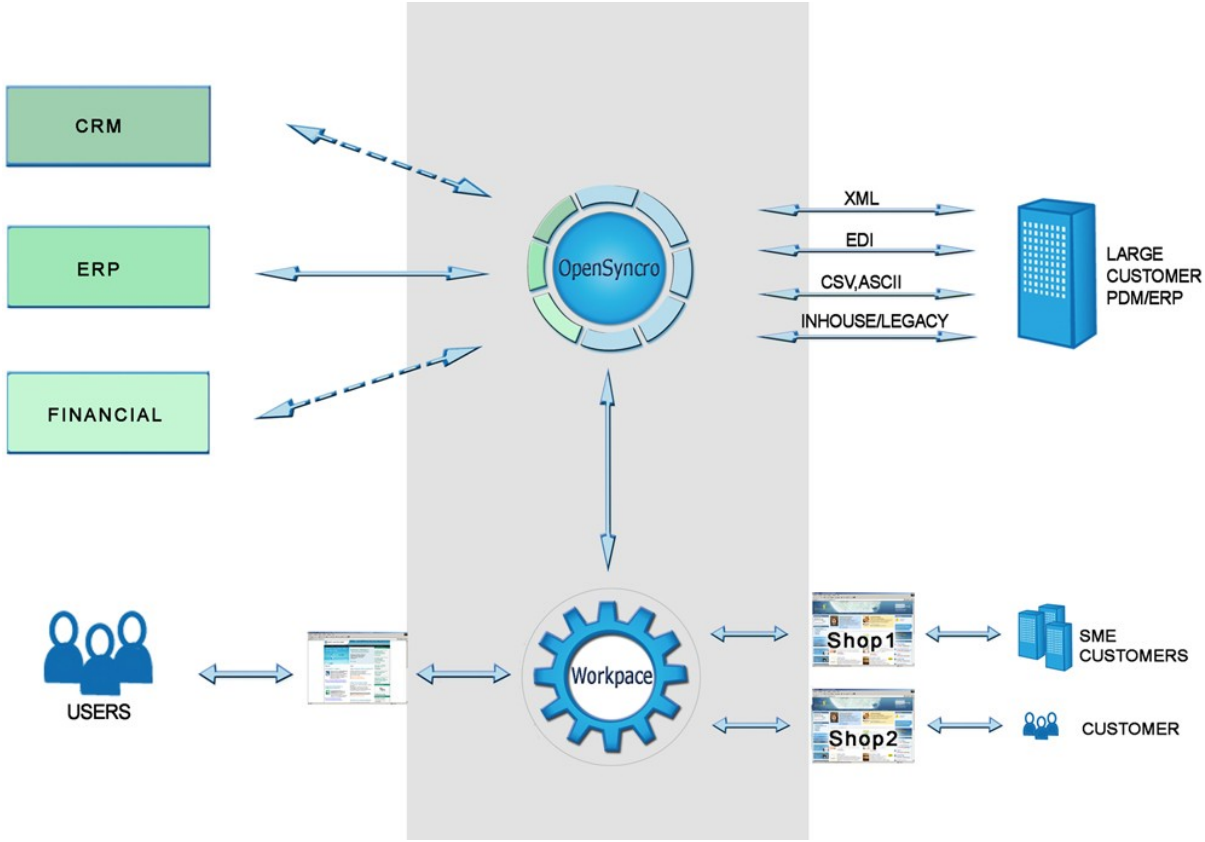
OpenSyncro also sends product data to larger corporate customers using for example XML or EDI formats. Different data formats can be used for different corporate customers.

Corporate customers can also place purchase orders. OpenSyncro can extract the orders for customers´ systems, transform the files to desired format and then load the orders to company ERP.

Address
Itälahdenkatu 22 A
00210 Helsinki
Finland, Europe

Phone & Fax
+358 – 9 – 25 122 10
+358 – 9 – 25 122 119

Internet
web: www.smilehouse.com
e-mail: info@smilehouse.com



For further information about ready-made Workspace integrations, please refer to additional documentation found at workspace.smilehouse.com.

For further information about OpenSyncro and available Connector Packs, please refer to the OpenSyncro documentation found at www.opensyncro.org.

6.2. Front-end API

The role of Workspace Front-end API is to enable both adding Workspace functionality (e.g. shopping cart, product data, order process) to other web sites and/or portal layer and managing web shop UI with Ajax.

Workspace Front-end API is a REST API using JSON notation. As of version 1.12.2 the Front-end API already provides basic functionality for adding products to shopping cart using Ajax. The Front-end API will be expanded in future versions.

6.3. Payment Gateway API

Workspace can also be used as a Payment Gateway. In this case all the built-in Payment system interfaces can be accessed through Payment Gateway API.

This Payment Gateway functionality is useful if you have an existing e-commerce functionality and you want to enable customers to pay using some of the built-in payment systems of Workspace.

For further information, please refer to the Workspace Payment Gateway API found at workspace.smilehouse.com.

6.4. Identification Gateway API

Workspace can also be used as an Identification Gateway. In this case all the built-in identification system interfaces can be accessed through Identification Gateway API.

This Identification Gateway functionality is useful if you have an existing e-commerce functionality and you want to enable reliable customer identification using some of the built-in identifications of Workspace. All built-in Finnish identification systems are based on TUPAS standard of the Finnish Banker's Association.

For further information, please refer to the Workspace Identification Gateway API found at workspace.smilehouse.com.

6.5. Plug-ins / Workspace SDK

No matter how versatile functionality the software has, there will always be customers who want some part of it to work a bit differently. They might for example want the delivery costs of an order to be calculated in some complicated way based on the ordered products or some special actions to be taken on the event of receiving an order etc.

We could of course tailor a completely separate customized versions of Workspace for these customers, but this would be troublesome and lead to serious problems as the software evolves.

To avoid these difficulties the Workspace plug-in framework was designed. Now when a customer wants something in Workspace - let's say the delivery cost calculation - to function differently we turn this part of the software into a pluggable service. Workspace default implementation now becomes the default plug-in which can be replaced by customized functionality simply by installing a custom plug-in.

Creating a new pluggable service and a custom plug-in for it requires of course roughly the same amount of work than simply making a customized version of the software, but after this is done we can make as many different plug-ins for the service as needed and the actual Workspace code won't have to be touched anymore.

If someone else now wants the delivery costs to be calculated differently we can simply make a suitable plug-in, send it to him and tell him to place it in the Workspace plug-in directory and then restart the application server. Currently there are only few pluggable services in Workspace but their number will surely increase in the future.

7. Security considerations

7.1. General

Many security issues have been addressed in the development of Workspace. Sufficient security in web applications is difficult to implement and it is impossible to confirm that all security issues have been properly addressed in any application, including ours.

We have tested for multiple known security issues in Workspace ourselves and implemented protection for these issues. Instead of solely trusting our own judgment we have also evaluated our product with an independent security consulting agency.

7.2. Addressed issues

7.2.1. Session management

Session identifiers are kept in cookies
Session identifiers are always transferred via cookies to prevent them leaking via referrer urls or bookmarks.

Secure session identifiers
Workspace doesn't generate its own session identifiers but delegates that responsibility to its application server. Although the solutions security then depends on the application server's security, it usually provides stronger session identifiers than homegrown solutions because application servers go through much scrutiny by their large user base.

7.2.2. Authentication and authorization

User and Role based authorization
Workspace uses a popular and proven Role based authorization scheme.

Single point of entry
The Workspace Administration Interface authentication implements a single point of entry which enables easier evaluation of authentication's correctness.

Password based authentication
Users are authenticated via passwords.

Optional IP restrictions support
Access to the Administration Interface and to the Webshop can also be restricted via IP restrictions.

7.2.3.Privacy considerations

Caching disabled
Pages served by Workspace have http-headers that tell browsers to disable caching of the pages. This somewhat decreases the risk of having private pages stored on a public computer's browser cache.

Passwords aren't stored, only their checksums
Workspace doesn't store the user's password. It only stores an MD5 checksum of the password that can be used to check that the user has given the same password as previously. The stored checksum generally cannot be used to retrieve or reassemble the password.

Credit card information is not available
Workspace online payment system integrations have been implemented in ways that the buyer's private payment information (credit card numbers etc) is not available to Workspace at any time. The buyer always supplies this kind of information directly to the payment solution.

7.2.4.Platform security support

SSL enabling
SSL should always be enabled for the Administration Interface. See your application server instructions for configuring SSL support. We also recommend using SSL for the webshop too.

Java Sandbox and Workspace security policy
Java Security policy descriptions can be created to externally restrict Workspace's access to the system.

Note that general system security issues should also be addressed, such as updating security fixes, auditing user access, using a well-configured firewall and secure passwords, checking log files, etc.

7.2.5. Other issues

Cross-site scripting prevention
Cross-site scripting has been identified as a security threat and Workspace has gone through multiple evaluations regarding cross-site scripting in different situations. For example all information given by the user is escaped in order to prevent hijacking of the Administration Interface.

Buffer overflows
Buffer overflows are not likely to happen on the Java platform and Workspace doesn't include any native extensions that would increase the likelihood of buffer overflows.

SQL Injection
SQL Injection has also been identified as a security threat and it is prevented systematically throughout the software, by using PreparedStatements (or an equal method) in most places and by escaping input manually with a library method where the previously mentioned techniques aren't used but user input is involved.

7.3. Security testing

To evaluate the security of Workspace we have employed consulting from an independent security consulting agency which offers both Web application and System level security evaluations.

Security evaluations are a continuous process and we do a lot of co-operation with security experts to ensure that our applications and deployments are secure as possible.

8. 3rd party elements

Workspace distribution contains the following 3rd party libraries.

Sun JavaBeans Activation Framework (JAF)
File: activation-1.0.2.jar
License: Sun Microsystems, Inc. Binary Code License Agreement
Version: 1.0.2
URL: http://java.sun.com/products/javabeans/glasgow/jaf.html
Description: Library required by Sun's JavaMail API.

ANTLR parser generator
File:antlr-2.7.6.jar
License:BSD License
Version:2.7.6
URL:http://www.antlr.org/
Description: Language tool that provides a framework for constructing recognizers, compilers, and translators from grammatical descriptions.

ASM bytecode library
Files: asm-attrs.jar, asm.jar
Licence: BSD Licence
Version: unknown
URL: http://asm.objectweb.org/
Description: Java bytecode manipulation framework required by the Hibernate.

Hansa bank's Banklink API
File: banklink-1.4.1.jar
License:
Version: 1.4.1
URL: http://w.hansa.ee/est/pangalink/banklink_1_4_1.zip
Description: Payment interface for the estonian Hansa bank.

C3P0 JDBC connection pooling library
File: c3p0-0.9.1.jar
License: LGPL
Version: 0.9.1
URL: http://sourceforge.net/projects/c3p0
Description: JDBC Connection pooling library used by the Hibernate

CGLIB Code Generation Library
File: cglib-full-2.1.3.jar
License: Apache License, Version 2.0, January 2004, http://www.apache.org/licenses/
Version: 2.1.3
URL: http://cglib.sourceforge.net
Description: Required by the Hibernate library for dynamic code generation.

Jakarta Commons Collections
File: commons-collections-2.1.1.jar
License: Apache License, Version 2.0, January 2004, http://www.apache.org/licenses/
Version: 2.1.1
URL: http://jakarta.apache.org/commons/collections/
Description: Required by the Hibernate library for efficient handling of data structures.

Jakarta Commons Logging
File: commons-logging-1.0.4.jar
License: Apache License, Version 2.0, January 2004, http://www.apache.org/licenses/
Version: 1.0.4
URL: http://jakarta.apache.org/commons/logging/
Description: Required by the Hibernate library for logging.

Jakarta Commons Codec
File: commons-codec-1.3.jar
License: Apache License, Version 2.0, January 2004, http://www.apache.org/licenses/
Version: 1.3
URL: http://jakarta.apache.org/commons/codec/
Description: Commons Codec provides implementations of common encoders and decoders such as Base64, Hex, Phonetic and URLs.

dom4j
File: dom4j-1.6.1.jar
License: BSD style license, http://dom4j.org/license.html
Version: 1.6.1
URL: http://dom4j.org
Description: Required by the Hibernate library for reading xml configuration files.

DynAPI
File: dynapi-2.5.6.tar.gz
License:
Version: 2.5.6
URL: http://dynapi.sourceforge.net/dynapi/
Description: Used for creating dropdown-menus in the Administration Interface.

Apache ECS Element Construction Set
File: ecs-1.4.1.jar
License: Apache License, Version 2.0, January 2004, http://www.apache.org/licenses/
Version: 1.4.1
URL: http://jakarta.apache.org/ecs/index.html
Description: Used for creating pages in the Administration Interface.

EHCache Easy Hibernate Cache
File: ehcache-1.2.3.jar
License: The Apache Software License, Version 1.1
Version: 1.2.3
URL: http://ehcache.sourceforge.net/
Description: Required by the Hibernate library for caching.

Enterprise Distributed Technologies FTP library
File: ftp.jar
License: LGPL
Version: 1.2.1
URL: http://www.enterprisedt.com/products/edtftpj/
Description: Used for copying files by FTP.

Hibernate
File: hibernate-3.2.5.ga.jar
License: LGPL
Version: 3.2.5.ga
URL: http://www.hibernate.org
Description: Used for persisting java objects to the database.

Apache Logging Services
File:log4j-1.2.11.jar
License: Apache License, Version 2.0, January 2004, http://www.apache.org/licenses/
Version:1.2.11
URL:http://logging.apache.org/log4j/docs/
Description: Log4j is used for creating log information of workspace.

Libraries from Sun Java Web Services Developer Pack (JWSDP) with updated SAAJ library
File: jaxrpc-api.jar, jaxrpc-impl.jar, jaxrpc-spi.jar, jax-qname.jar, FastInfoset.jar, jsr173_api.jar, saaj-api.jar, saaj-impl.jar
Version: JWSDP 1.6, SAAJ 1.3
License: Sun Microsystems, Inc. Binary Code License Agreement. BEA Reference Implementation License (JSR173). SAAJ library has dual license consisting of CDDL v1.0 and GPL v2 (https://glassfish.dev.java.net/public/CDDL+GPL.html).
URL: https://jax-rpc.dev.java.net/, https://saaj.dev.java.net/
Description: Libraries from the Sun's Java Web Services Developer Pack

JFreeChart
File: jfreechart.jar
License: LGPL
Version: 0.5.6
URL: http://www.jfree.org/jfreechart/
Description: Used for drawing statistics charts in the Administration Interface.

Sun's Java Transaction API
File: jta.jar
License: Sun Microsystems, Inc. Binary Code License Agreement
Version: unknown
URL: http://java.sun.com/products/jta/
Description: Required by the Hibernate library for supporting transactions.

Sun's JavaMail API
File: javamail-1.3.2.jar
License: Sun Microsystems, Inc. Binary Code License Agreement
Version: 1.3.2
URL: http://java.sun.com/products/javamail/
Description: Used for sending email.

Apache Lucene
File: lucene-core-1.9.1.jar
License: Apache License, Version 2.0, January 2004, http://www.apache.org/licenses/
Version: 1.9.1
URL: http://lucene.apache.org/java/docs/
Description: High-performance, full-featured text search engine library.

Jason Pell's MultipartRequest library
File: multipartrequest-1.30rc1.jar
License: LGPL
Version: 1.30rc1
URL: http://www.geocities.com/jasonpell/programs.html
Description: Used for uploading files to the Administration Interface.

ODMG API
File: odmg-3.0.jar
License: http://www.odmg.org
Version: 3.0
URL: http://www.odmg.org
Description: Specification used by the Hibernate library for supporting the standard for storing objects.

Tiny Moxiecode Content Editor
File: tinymce_2_0_8.zip
License: LGPL
Version: 2.0.8
URL: http://tinymce.moxiecode.com/
Description: TinyMCE is a platform independent web based Javascript/HTML WYSIWYG editor.

Xerces2 Java Parser
File: xercesImpl.jar, xml-apis.jar
License: Apache License, Version 2.0, January 2004, http://www.apache.org/licenses/
Version: 2.71
URL: http://xml.apache.org/xerces2-j/
Description: Used for XSL transformations.

Prototype JavaScript framework
File: sse.html/lib/prototype/prototype.js
License: MIT-style license, http://www.apache.org/licenses/
Version: 1.5.0_rc0
URL: http://prototype.conio.net/
Description: Used by script.aculo.us javascript package

scriptaculous-js
File: sse.html/lib/scriptaculous/
License: MIT-style license, http://www.apache.org/licenses/
Version: 1.6.1
URL: http://script.aculo.us
Description: Used for drag & drop behavior in administration interface

Axis
File: axis.jar
License: Apache License, Version 2.0, January 2004, http://www.apache.org/licenses/
Version: 1.4
URL: http://ws.apache.org/axis/
Description: Used in SOAP-based payment system integrations.

Commons discovery
File: commons-discovery-0.2.jar
License: Apache License, Version 2.0, January 2004, http://www.apache.org/licenses/
Version: 0.2
URL: http://commons.apache.org/discovery/
Description: library required by the Axis

WSDL4J
File: wsdl4j-1.5.1.jar
License: Common Public Licence
Version: 1.5.1
URL: http://sourceforge.net/projects/wsdl4j
Description: library required by the Axis

EmailAddress
File: sakuracms-emailaddress.jar
License: GNU Lesser General Public License, version 2.1, http://www.gnu.org/licenses/old-licenses/lgpl-2.1.html
Copyright: Les Hazlewood
URL: http://www.leshazlewood.com/?p=23
Description: library required for validating e-mail address format as specified in the RFC 2822.

Apache XML-RPC
Files: xmrlrpc-common-3.1.1.jar, xmlrpc-client-3.1.1.jar
Version: 3.1.1
License: Apache Software License, Version 2.0
Copyright: Apache Software Foundation
URL: http://ws.apache.org/xmlrpc/
Description: Apache XML-RPC is a Java implementation of XML-RPC , a popular protocol that uses XML over HTTP to implement remote procedure calls.

Apache WS Common Utilities
Files: ws-commons-util-1.0.2.jar
Version: 1.0.2
License: Apache Software License, Version 2.0
Copyright: Apache Software Foundation
URL: http://ws.apache.org/commons/util/index.html
Description: This is a small collection of utility classes, that allow high performance XML processing based on SAX.

Liquibase
Files: liquibase-1.9.1.jar
Version: 1.9.1
License: http://www.gnu.org/licenses/old-licenses/lgpl-2.1.html
URL: http://www.liquibase.org/
Description: Liquibase is used for database upgrades.

DbUnit Framework
Files: dbunit-2.4.4.jar
Version: 2.4.4
License: GNU Lesser General Public License, Version 2.1,
http://dbunit.sourceforge.net/license.html
URL: http://dbunit.sourceforge.net/
Description: DbUnit is a JUnit extension (also usable from Ant) targeted for database-driven projects that, among other things, puts your database into a known state between test runs. This is an excellent way to avoid the myriad of problems that can occur when one test case corrupts the database and causes subsequent tests to fail or exacerbate the damage.