Comparing extranet solutions for an ERP integration

Bill Kåla

Bachelor's thesis in Information Technology

Vaasa 2013

BACHELOR'S THESIS

Author: Bill Kåla

Degree Programme: Information Technology, Vaasa

Supervisor: Susanne Österholm

Title: *Comparing extranet solutions for an ERP integration*

Date 2.9.2013 Number of pages 30 Appendices 2

Summary

This thesis was commissioned by Fouga IT and it is about researching and testing possible solutions to integrate an extranet with the ERP system Lemonsoft. The extranet will work as a web shop for retailers only and all information has to go through Lemonsoft. A few open source e-commerce platforms were researched and compared to a solution created from scratch.

The biggest problem was how the extranet works together with Lemonsoft. Requirements for the extranet were simple and almost any open source e-commerce platform filled them. Mainly one e-commerce platform was researched and tested in this thesis.

The result is a suggestion that is efficient and easy to implement.

Language: English Key words: e-commerce, Lemonsoft, ERP

EXAMENSARBETE

Författare: Bill Kåla
Utbildningsprogram och ort: Informationsteknik, Vasa
Handledare: Susanne Österholm

Titel: *Jämförelse av extranetlösningar för en ERP-integration*

Datum 2.9.2013 Sidantal 30 Bilagor 2

Abstrakt

Detta examensarbete gjordes på uppdrag av Fouga IT och handlar om jämförelse av möjliga lösningar för att integrera ett extranet med ERP-systemet Lemonsoft. Extranätet skall fungera som en webbshop för återförsäljare och all information skall överföras via Lemonsoft. Några e-handelsplattformar med öppen källkod har undersökts och jämförts med en lösning gjord från noll.

Största problemet med dessa olika lösningar var hur de fungerade hand i hand med Lemonsoft. Kraven för extranätet var enkla och nästan alla e-handelsplattformar uppfyllde dem. Det var huvudsakligen en e-handelsplattform som undersöktes och testades i detta examensarbete.

Resultatet är ett förslag som är effektivt och lätt att implementera.

Språk: Svenska Nyckelord: e-handel, Lemonsoft, ERP

OPINNÄYTETYÖ

Tekijä:	Bill Kåla
Koulutusohjelma ja paikkakunta:	Tietotekniikka, Vaasa
Ohjaaja:	Susanne Österholm

Nimike: *Extranet ratkaisujen vertailu ERP-integrointia varten*

Päivämäärä 2.9.2013 Sivumäärä 30 Liitteet 2

Tiivistelmä

Tämän opinnäytetyön toimeksiantaja on Fouga IT ja työ käsittää mahdollisten ratkaisujen vertailua extranetin liittämisessä osaksi Lemonsoft toiminnanohjausjärjestelmään. Extranetin tarkoituksena on toimia verkkokauppana jälleenmyyjille ja tiedonkulku tapahtuu Lemonsoftin kautta.

Muutamaa avoimen lähdekoodin verkkokauppajärjestelmää on tutkittu ja vertailtu itse räätälöityyn ratkaisuun. Suurin ongelma oli kuinka eri ratkaisut toimivat yhdessä Lemonsoftin kanssa. Extranetille asetetut vaatimukset olivat yksinkertaiset ja lähes jokainen verkkokauppaan perustuva ratkaisu täytti ne. Pääasiassa yhtä verkkokauppaa ja räätälöityä ratkaisua on tutkittu tässä työssä.

Tuloksena on ehdotus Fouga IT:lle helposti toteutettavasta ja tehokkaasta ratkaisusta.

Kieli: Suomi Avainsanat: sähköinen kaupankäynti, Lemonsoft, ERP

Table of Contents

ABBREVIATIONS

API	Application Programming Interface
ASP.NET	Web Application Framework
CRM	Customer Relationship Management
EAV	Entity-Attribute-Value database model
E-Commerce	Electronical Commerce
EDI	Electronic Data Interchange
ERP	Enterprise Resource Planning
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
ID	Identifier
IIS	Internet Information Services
MS	Microsoft
MVC	Model-View-Controller software architecture
MySQL	Relational Database Management System
PHP	Hypertext Preprocessor
SGS	Company offering certificate services
SME	Small-Medium Enterprise
SOAP	Simple Object Access Protocol
UI	User Interface
WSDL	Web Services Description Language
XAMPP	Cross-Platform Web Server Solution Stack
XML	Extensive Markup Language

1 Introduction

1.1 Employer

Fouga IT Ltd, which is a company owned by professionals located in Kokkola and Kaustinen, implements IT-services that support the customers' businesses. The most important services provided are:

- work management applications for location-based services.
- data management planning and implementation.
- office environment infrastructure.
- LAN technology, server and storage solutions.
- remote access and communication solutions.

Fouga IT has a total of 12 employees divided between Kokkola and Kaustinen. The company represents IBM, Lenovo and Toshiba in devices, and Microsoft, M-Files and Lemonsoft in software. A budgeted turnover of one million euros was planned for the year 2013.

Fouga IT is certified by SGS, meaning that SGS has granted Fouga IT the ISO9001 and ISO14001 certificates.

The company's goal is to be able to offer anything IT related to their customers, meaning that Fouga IT strives to be the best in the area. /18/

1.2 Task

The task was to research and test different E-commerce platforms which then should be integrated with an Enterprise Resource Planning system (ERP) called Lemonsoft. A customer wanted Fouga IT to create an extranet for retailers with the ERP Lemonsoft as the primary system. This means that an extranet solution could be created using an open-source E-commerce platform or that a solution could be made from scratch.

In figure 1 you can see that Lemonsoft is the primary system in this solution containing all data.

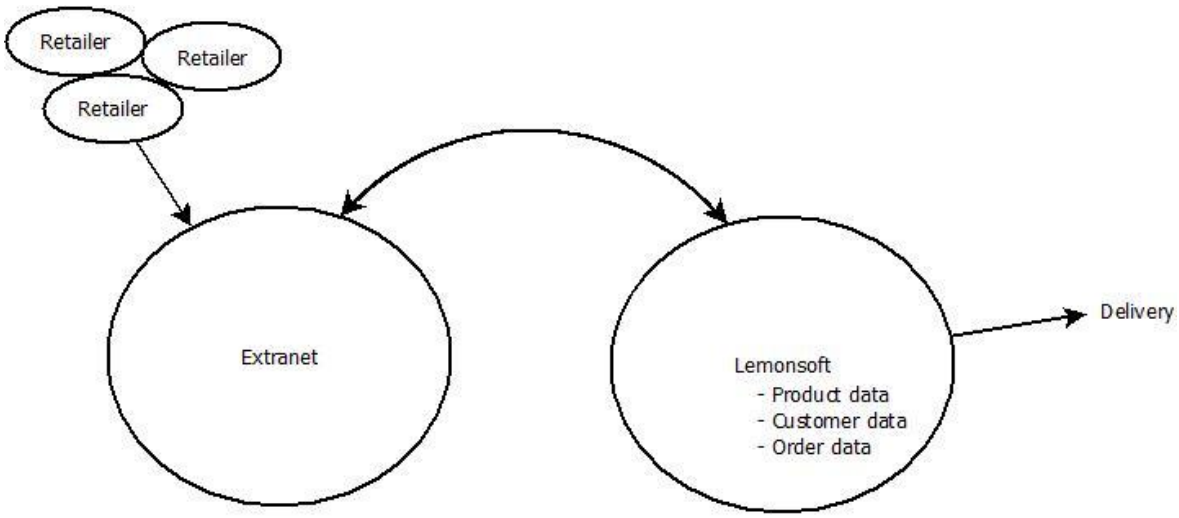

Figure 1. Extranet scheme

This may seem like an easy task at first but it certainly was not because using Lemonsoft as the primary system means that data must be exchanged automatically between the extranet and Lemonsoft. To achieve this the different E-commerce platforms had to be researched and tested thoroughly and also the possibility that the extranet would be created as a tailored solution from scratch had to be researched and tested.

The customer required the extranet to be stylish and easy to use for the retailers. This means that categorization of products and simple but informative product pages were important. The retailers that order the products through the extranet need to identify themselves, meaning that some kind of login is also of importance. After each new order an e-mail has to be sent to the sales administrator (customer from Fouga IT point of view).

The customer should never have to do anything to the extranet. Everything that the customer does regarding products and customers in Lemonsoft should be updated automatically to the extranet because the ERP works as the primary system.

The most important part of this task was to solve the data exchange issue. During the research and testing phases several problems were found that were not thought of in the beginning. Several of these problems, with possible solutions, are discussed in chapter 4.

In this thesis the different E-commerce platform possibilities and a solution made from scratch are researched and tested to find out how they would fit as an extranet with integrated ERP.

As a result a suggestion was made for a solution that would be efficient and easy to implement and that would fulfill all the requirements of this task.

2 E-commerce

E-commerce or electronic commerce is a term for business, or commercial action, that involves some kind of transfer of information across the Internet. It is one of the most important things to emerge from the Internet. E-commerce allows consumers to exchange goods and services from anywhere in the world irrespective of time or distance.

Electronic transactions have existed since 1988 in the form of Electronic Data Interchange (EDI). Unfortunately when using EDI you have to have set up a dedicated data link. /19/

E-commerce itself is nowadays more and more important to a company's image. A website and good design help businesses and individuals to achieve their objectives. /20/

Due to the easiness in shipping worldwide today a change from conventional stores to electronic ones is seen in the near future. Although conventional stores are needed, E-commerce does increase company sales since you can sell and ship worldwide with ease. For example platforms like *Big cartel* have made selling products through the Internet so easy. *Big cartel* does everything for you except shipping. All you have to do is to ship the products/services. /1/

2.1 E-commerce selection process

During the first stage of the E-commerce selection process the task was to pick two to three E-commerce platforms and compare them and their different attributes and possibilities. After a couple of Google searches it was noticed that on almost every top 10 list Magento was controlling the number one spot. Without any further thought the decision to choose Magento was made. After that the selection process got a bit tricky because there was no clear number two in the reviews. The different platform documentations were studied but no platform really stood out except Magento and its API.

The decision was made to choose Magento, Woo commerce and Open Cart. Magento was chosen because it was the clear list winner, Woo commerce because it is different than the others and Open Cart because of its placement in the reviews.

Magento is the market leading platform trusted by more than 150,000 businesses, including world-leading brands such as Samsung, Nike, Gant and Olympus. Magento is an open source E-commerce platform owned by eBay Inc. which was developed by Varien using the Zend Framework. Magento was launched on March 31, 2008 and offers one free version, which is the community edition. Magento uses the entity-attribute-value (EAV) database model. Magento's model is different from the traditional relational model. Magento is written in PHP and uses MySQL as its database. /21/ /22/ /28/

Woo commerce is a WordPress extension based on PHP and MySQL running on a web hosting service. Both Woo commerce and WordPress are free with open source code. Of the top 10 million websites WordPress is used by over 18.9 %, powering over 60 million

websites worldwide. WordPress was released on May 27, 2003 by the founders Matt Mullenweg and Mike Little. Version 3.5 has been downloaded 18 million times as of April 2013. Woo commerce has been downloaded 1.3 million times as of September 2013. /23/

Open Cart is a free open-source E-commerce platform based on the MVC-Framework. Open Cart was founded by Daniel Kerr. This is a cross-platform, meaning it can be run from Linux, Windows or other systems. As for the programming language, it is written in PHP and uses MySQL as the database. /24/

Shortly after these platform selections documentation research began. Nothing extra was found in there except that Magento had a great API that could be used to get and set data. All of the platforms were installed on an XAMPP test server locally and test data was inserted in all of them to see how they worked. After that Woo commerce stood out by its simplicity and ease of use. Unfortunately that was not a criterion.

Magento proved to be very time-consuming compared to the other platforms. Because of this there was not enough time for Woo commerce, nor Open Cart but this was actually a good thing since neither Woo commerce nor Open Cart had an API that could be used. The data transfers would have been hard to implement because of that.

3 Lemonsoft

Lemonsoft Oy is a company founded in 2006 that is located in Finland. Lemonsoft employs 33 employees across Finland. Offices are found in Vaasa, Joensuu and Helsinki.

Lemonsoft ERP is a software program meant for Small-Medium Enterprises (SMEs) as a total solution for financial management, human resource management, logistics, production and project management needs. It consists of the following parts:

- Financial Management.
- Customer Relationship Management (CRM).
- Human resources.
- Logistics.
- Production Control.
- Management Tools.
- Project Management.
- Document Management.

Lemonsoft itself is really easy to use and the only requirements are that the user knows Windows basics. The program is a database-based software, using the standard MS SQL Server. Although the program is based on a database they can be separate, meaning that the database can be running on a server and Lemonsoft software is run from a local workstation. In a nutshell this means that Lemonsoft is a client-server architecture-based software. /14/

Lemonsoft trains one or more persons in a company, depending on its size, so they can teach others. After a person has been trained he/she will then work as a contact person within the company if any problems emerge with Lemonsoft.

Lemonsoft is the most important part in this project. It will work as the primary system for the solution. All data will be transferred through Lemonsoft. This is because the customer wants to use only one program and everything should then be updated to the extranet automatically. /2/ /3/

A company using an ERP often only uses one software instead of several that do not allow interaction with each other. Constantly flowing information that is registered in a centralized database allows the users to follow different processes at any moment. An ERP integrates the company's departments and functions into one single computer allowing all data to be accessible from one place. /26/

In this thesis only the sales section of the ERP is used. The customer will probably implement other features into the ERP system like Human Resources etc.

Figure 2 gives a closer look at Lemonsoft and what it offers to companies:

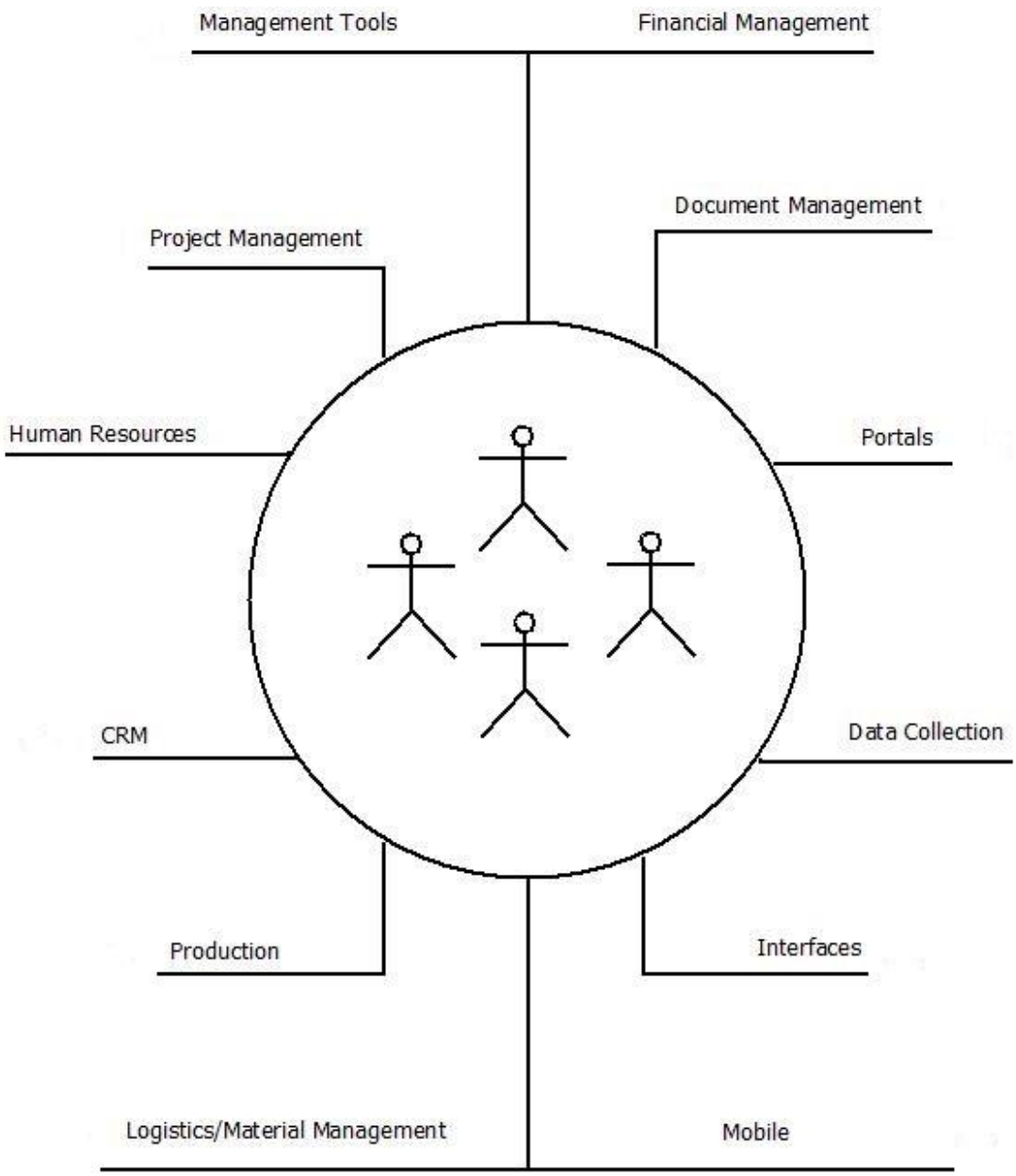

Figure 2. Lemonsoft and all its different use possibilities /4/

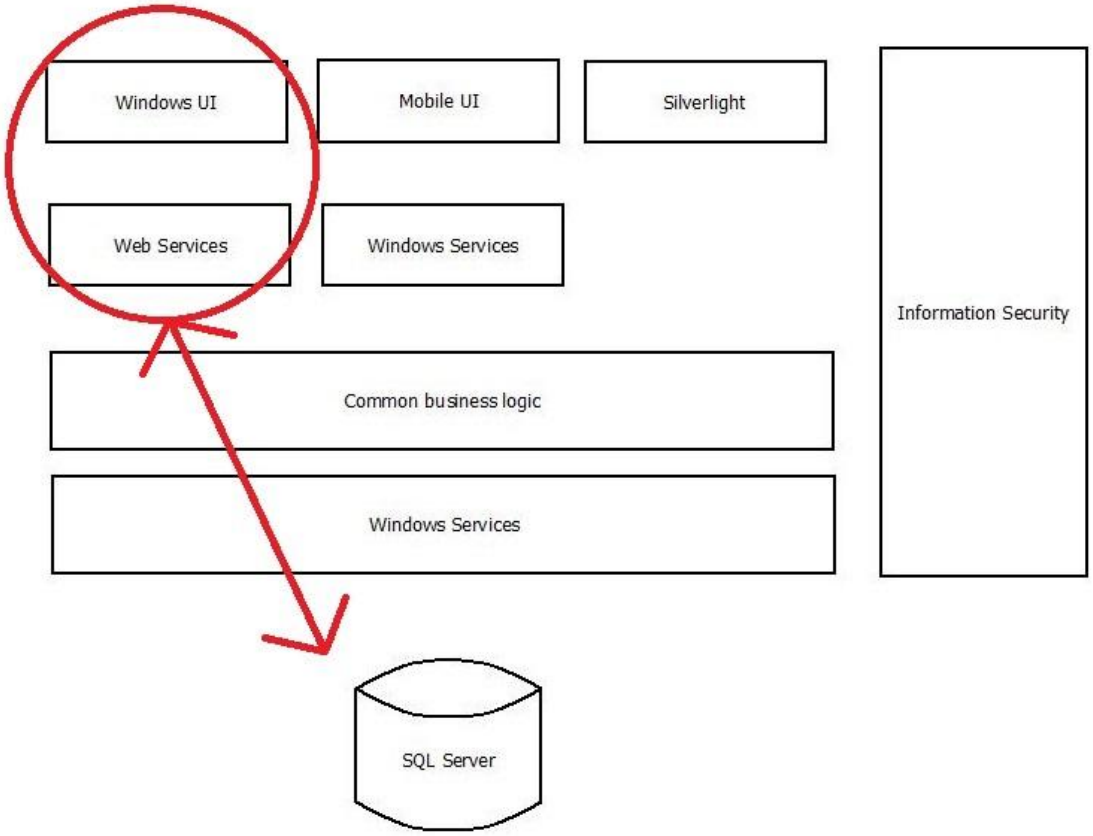

Figure 3. Lemonsoft architecture /5/

Figure 3 shows the Lemonsoft architecture. As you can see only the Windows User Interface (UI) and Web Services are used in this project. It is possible to implement Mobile UI, Silverlight and Windows Services after the initial extranet has been made.

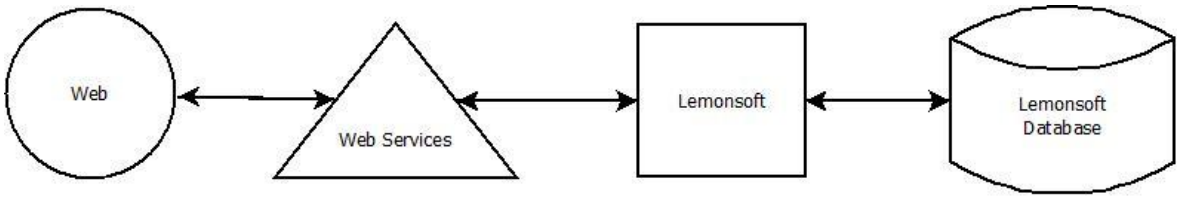

Figure 4. Lemonsoft service interface /6/

The Lemonsoft interface looks like figure 4. To get data from Lemonsoft, web services are used. Figure 4 shows how data is transferred using web services irrespective of using an E-commerce platform or creating a tailored solution.

4 Problems and possible solutions

This project contains three big problems and a handful of smaller ones. The big problems were known from the start but the smaller ones were found during our testing phase of each solution. The three big problems are:

- How to exchange product data between Lemonsoft and Magento.
- How to exchange customer data between Lemonsoft and Magento.
- How to exchange order data between Lemonsoft and Magento.

When a user (customer) inserts data into Lemonsoft, it needs to be displayed in the extranet. This also applies in the opposite direction, for example a placed order needs to make its way to Lemonsoft to be processed. Mainly all three problems are the same but they come with their own traits.

4.1 Magento

Although Lemonsoft is the primary system in this integration, Magento still serves an important role. It should handle all the data sent by Lemonsoft and make the data visible for the retailers and also send data back. Its importance is not just limited to handling data, but it shall also categorize the products, create different size product images and send an e-mail of ordered products to the sales administrator.

The problems described in chapter 4 needed to be solved for the extranet to work as intended. When data like a new or modified product is inserted into Lemonsoft it needs to make its way to Magento. A possible solution was to use the Magento API and Lemonsoft web services.

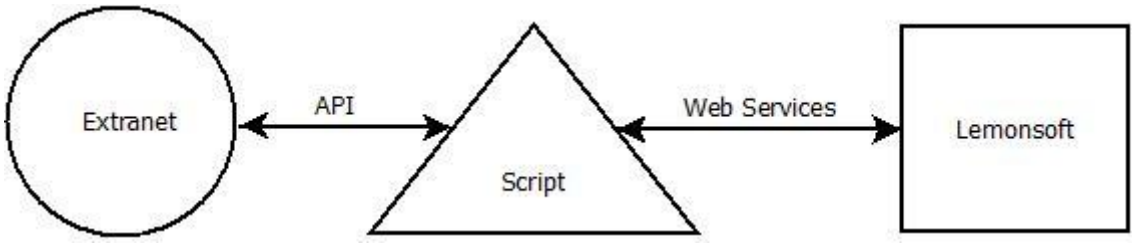

Figure 5. Data transfer scheme

Figure 5 shows how data could be transferred between Lemonsoft and Magento. Lemonsoft web services could be used to get data from Lemonsoft and the Magento API could be used to insert/update the product data. Unfortunately, the data could not be transferred in real time because the script needs to be executed for the data to move. After some discussion within the team the decision was made to execute the script each night using Windows Scheduled Tasks. Nightly executions were selected because if any errors happen they will not interfere with the possible ongoing orders by retailers.

Also customer (retailer) data needed to be exchanged but, if the customer is new, an e-mail needed to be sent to the retailer with login information. To achieve this there were two possible solutions. The first one was to modify Magento to send an e-mail with an auto-generated password and the second was to send the e-mail using the script that is executed at night. It seemed easier and safer to send the e-mail via Magento and also prompt the customer to change the auto-generated password at the first login instead of sending the login information as it is with the script because the e-mail is sent in the Magento end after data has been successfully exchanged.

When an order has been placed in the extranet by a retailer the data need to be moved from extranet to Lemonsoft. A possible solution to achieve this is to use the Magento API to get the order data and then use the Lemonsoft web service to insert the order into Lemonsoft. As you may have noticed now the data is moved the opposite way from Magento to Lemonsoft. To send an e-mail regarding the newly created order to the sales administrator has a fairly easy solution since Magento has a built-in function to send e-mail to the administrator after each new order. This was tested in chapter 6.1.

It was found out that no one had thought of product deletion. If a product is removed from Lemonsoft, the same product should be removed from the extranet. To solve this problem a script has to be created and executed using scheduled tasks. There are also other possibilities to solve this problem, e.g. hiding product/customer data using a module that could be created in Magento. In this way the products are still there if they need to be visible.

To test these solutions a script written in PHP was made. More details about the scripts will be given in chapter 6.1.1.

4.2 ASP.NET Solution

The same problems persist even if a tailored extranet is created, meaning that data still needs to be exchanged between the extranet and Lemonsoft. Two programming languages were possible when creating a tailored solution. The extranet could be created using PHP and ASP.NET. The decision to create the extranet using ASP.NET was taken because it seemed useless to create it using PHP since Magento is written in PHP and its technical contents would need to be done by hand.

To exchange data between the ASP.NET extranet and Lemonsoft was a lot easier compared to the Magento extranet. Figure 6 shows you why.

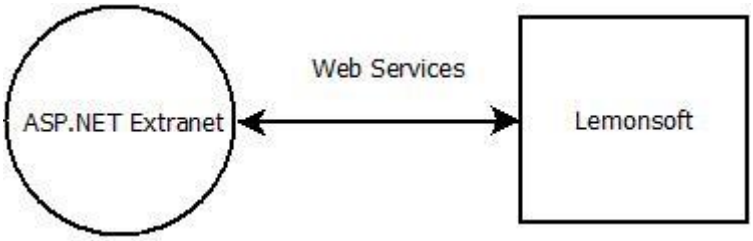

Figure 6. ASP.NET extranet data transfer scheme

As you may have noticed the script between the extranet and Lemonsoft in figure 5 is gone in figure 6. This is because a possibility to use real time data using the Lemonsoft web services was available. In the ASP.NET code behind *Page_Load* section, web services could be used. This means that data is fetched from Lemonsoft each time the page loads using web services. Another possibility could be connecting the extranet to the Lemonsoft database. Connecting the extranet to the database results in one problem. If Lemonsoft changes their database structure, the whole extranet may become unavailable for use and it needs to be rewritten again to match the database fields. It is almost certain that the web services will not be modified in such a way that the extranet could become unusable. This is why using the Lemonsoft web services with each *Page_Load* is recommended.

Product images are a problem that no one had thought about. Since the product images are stored on the customer's network drive they need to be somehow retrieved from there. They can't be retrieved every page load because that is too heavy on the server. It would be possible if the images were not full resolution images. A possible solution to this problem could be to base_64 encode the images and save them to a separate database and use them that way. Other possibilities could be caching the pages or similar methods. Images are not a problem when using Magento because of its built-in image handler.

An extranet (or in our case a web shop for retailers) always requires a shopping cart, and this could be done with session IDs. Lemonsoft has session IDs available for use but creating sessions by hand could also be possible.

Read more about testing these solutions in chapter 6.2.

5 Comparison of Magento and an ASP.NET solution

When comparing Magento and an ASP.NET solution several things were found that stand out in each of the solutions. All of these things were found after the testing phase. In this chapter the positive and negative aspects of each solution are discussed.

5.1 Magento

There are several good things about Magento, for example these:

- It is a complete and ready package.
- Categories, images, tags and search functions are built-in.
- A lot of other beneficial properties are available for use.
- An API to exchange data exists.
- Experience with E-commerce platforms is not required.
- A Login to identify the retailer is built-in.
- Extensions created by Magento users are found.
- Support / Forums with active users are available for help requests.
- Modules are available to re-write almost anything.

Magento as a whole is really easy to install and deploy. There are only a few hosting server requirements that are often met. Magento itself is a complete package with everything required by the customer, and even more. Product pages with categorization, tags and images to support the looks and easiness of browsing through products are found. It has an API that can do almost anything required by the user and there is also a possibility to create your own API. Since the extranet shall be behind a login or the retailer must be recognized before ordering products, Magento is a good choice.

With its extensions found at *www.magentoconnect.com,* possibilities to extend the extranet in the future are almost limitless. The extensions are created by regular Magento users and can be downloaded for free. In this case a Finnish language pack could be downloaded. Magento also has a great support forum where a reply from a Magento employee or a community member is guaranteed.

If any modifications need to be made to the Magento E-commerce platform, they can be made using modules. If you edit the source code in Magento, new updates could ruin the whole extranet. With modules it is possible to rewrite the code keeping the original source code intact.

After modifying Magento to your needs you have a ready package with all the required features already in place. Only styling is left and then the extranet Lemonsoft integration is ready. Unfortunately as good as Magento may seem it comes with some negatives. A lot must be modified and remodeled to work with the Lemonsoft integration. Since a checkout

process with payment is not needed, this has to be removed. To do this a module has to be made to keep the original source code intact and to prevent Magento updates from clashing with the modifications. To be able to order products without payment may not seem safe to you but the customer itself inserts the retailers into Lemonsoft and only they can login to the extranet. This means that the Magento registration must be removed.

It was also found out that when an order has been placed Magento requires the shop administrator to check two checkboxes. Shipping and invoice checkboxes need to be checked so that the order is in the right state. The customer did not want to do anything to the extranet. This is why the checkout needs to be modified.

Magento comes with a lot of unnecessary extra features. These are the administrator panel, graphs of sales and data exchange between Lemonsoft and the Magento extranet. Also, a search function is probably not needed since categories are used. Lemonsoft takes care of the graphs.

Magento uses a MySQL database as its information storage. Using two databases can cause a strain on the server. Preferably only one should be used but that is not possible with the Magento solution.

To run Magento on a server does not require much. Almost any provider meets the requirements which are:

- Apache 2.2.x.
- PHP compatibility 5.2.13 – 5.3.24.
- SOAP extension.
- MySQL 5.0.2 or newer.
- Availability to run scheduled tasks (crontabs) with PHP 5.

These requirements are more easily met than the requirements of a solution made from scratch. More details about this will be given in chapter 5.2. /16/

5.2 ASP.NET solution

The positive aspects of an ASP.NET solution are that no scheduled tasks are needed to exchange data because the solution could use the Lemonsoft web service to get and set data. In a nutshell this means that the data is in real time and no data deletion is needed. All of this heavily favors a tailor-made ASP.NET solution.

Since the extranet could be totally custom-made with no extra features, only filling the requirements set by the customer, this could lead to a more effective solution with numerous styling possibilities. Also a tailored solution could open up possibilities as a product that could be sold forward. Since Fouga IT is a company that supports Lemonsoft,

this tailored solution is a good way of to learning how Lemonsoft really works and what its possibilities are. Therefore offering other customers tailor-made solutions using Lemonsoft as the ERP is a possibility.

Even with full controllability of the extranet solution it requires a lot of work to make it from scratch since everything required has to be “hand” made. This means that login, shopping cart, categorization and checkout need to be “hand” made. This could prove to be very time consuming compared to a solution made using the Magento platform.

Keep in mind that the site needs to be easy to use. If retailers can’t find the products, they aren’t going to buy them. These 4 things are good to keep in mind when creating a web application:

- **Navigation:** What is found on the site and where it is located.
- **Graphics:** Keep the site attractive but effective.
- **Access:** The site should be usable by all users in spite of design decisions.
- **Browsers:** Browsers may interpret the HTML code differently.

/25/

To host an ASP.NET website seems to be fairly easy. A server computer with Windows Server, IIS (Internet Information Services) and Microsoft SQL Server for the databases are required. Since the Lemonsoft requires a pretty huge database it could possibly be hosted on a separate computer than the website.

To host a website with IIS does not require a lot of knowledge from before. Several good tutorials are found on the internet. /17/

6 Testing

In the testing phase of this project an attempt was made to solve the problems mentioned in chapter 4. This phase started off by testing how the Lemonsoft web service works using the code in code example 1 to see if data could be retrieved or inserted. Also, the same thing was done with the Magento API. After a successful test with both Lemonsoft and Magento the real testing was started. Code example 1 shows how to use the Lemonsoft web services.

```
// Muodostetaan tietoluokka
$asiakashaku = new stdClass();
$asiakashaku->iCustomer = 2;

// Haetaan asiakasyrityksen tiedot jos mahdollista
try
{
$soap = new soapClient( constWSDL, array(”trace” => 1)); //Keskusteluyhteys
$asiakas = $soap->GetCustomer($asiakashaku);             // Haetaan asiakastiedot
}
catch(SoapFault $fault)
{
echo ”Tietojenhakuvirhe: KOODI: ” . $fault->faultcode ”;
}
// Otetan oliosta nimi muuttujaan
$nimi = htmlentities ($asiakas->GetCustomerResult->Customer_name1, ENT_COMPAT,
”UTF-8”);
```

Code example 1. Lemonsoft example code of how to use the Lemonsoft web services /10/

6.1 Magento

All tests regarding the Magento solution were written in PHP, a language used by more than 20 million domains. /25/

Magento was tested out locally on an XAMPP platform. XAMPP is an open-source cross-platform web-server-solution stack package. It consists of an Apache HTTP Server, a MySQL database and interpreters for scripts in PHP and Perl programming languages. The acronym XAMPP is from these:

- X (meaning cross-platform).
- Apache HTTP Server.
- MySQL.
- PHP.
- Perl.

XAMPP is really easy to install since it requires only a .zip, .tar, .7z or .exe file to be downloaded and run. It also has a few good features that are not always needed. These are

phpMyAdmin that works as a user interface for your database. With phpMyAdmin you can access the database and do what you want to do. With XAMPP also comes the FileZilla FTP Server that comes in handy when FTP (File Transfer Protocol) traffic is required. /13/

Magento's SOAP API was tested in order to find out whether it could be used to insert and retrieve data to and from Magento. The abbreviation stands for Application Programming Interface. Basically an API is a set of programming instructions to be used when accessing different software applications. Often a company releases its API so that software developers can implement products that work hand-in-hand with their product. An API is a software-to-software interface. As a user of an application using API you only see one interface while behind it the API is working with other programs to make them work together.

This type of integration is called *seamless*, because a user never knows when data is being transferred from one application to another. A web service is a method of communication between two electronic devices over the World Wide Web. They are a collection of technological standards and protocols, like XML (Extensive Markup Language). The API itself is software code written as a series of XML messages. Each XML message has a corresponding function on the server. SOAP is responsible for translating the messages to make them understandable to and receivable by any operating system or network protocol.

One of the biggest users of API is Facebook. All data that is transferred between different sites and applications for example sharing photos, liking pages, logging in to applications with Facebook and a lot of more are established by using an API. Code example 2 shows you how to use the Magento API. /7/ /8/

```
$client = new SoapClient('http://magentohost/soap/api/?wsdl');

// If somestuff requires api authentification,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

$result = $client->call($session, 'somestuff.method');
$result = $client->call($session, 'somestuff.method', 'arg1');
$result = $client->call($session, 'somestuff.method', array('arg1', 'arg2',
'arg3'));
$result = $client->multiCall($session, array(
     array('somestuff.method'),
     array('somestuff.method', 'arg1'),
     array('somestuff.method', array('arg1', 'arg2'))
));


// If you don't need the session anymore
$client->endSession($session);
```

Code example 2. How to use Magento API /9/

6.1.1 Scripts

To solve the data exchange problems some scripts needed to be developed. The scripts were created according to figure 5 in chapter 4, which means that they are between the ERP and the extranet. The scripts will get data from Lemonsoft using its web services and set data to the extranet using the Magento API. The same methods are used when the script needs to fetch data from the extranet and insert it into Lemonsoft. Three test scripts were developed for each data exchange, meaning that the nightly data exchange scripts are executed separately to minimize the amount of data transferred. All of the scripts were developed using a modified version of code examples 1 and 2. Every script starts off by setting up a connection to the Magento API and the Lemonsoft web service as shown in code example 3. After a connection has been established several API/web service specific functions and calls can be executed like for example creating new products and modifying customers.

```
// Lemonsoft
$soap = new soapClient(
"http://192.168.150.12/LemonsoftWebServiceSetup/Services.svc?singleWsdl",
array("trace" => 1) );
                $products = $soap->GetProductList_Changed($productsearch);
// Magento
$client = new SoapClient('http://localhost/magento/api/soap/?wsdl');
```

Code example 3. Connection string to both Lemonsoft and Magento

When retrieving data with the Lemonsoft web service a list is returned containing data as seen in figure 7. The data is stored within objects that can be accessed as shown in code example 1.

```
object(stdClass)#5 (56) { ["Codelist_driver"]=> int(0) ["Sales_order_date"]=>
string(25) "2013-08-15T00:00:00+03:00"
["Sales_order_delivery_customer_address3"]=> NULL
["Sales_order_delivery_customer_name1"]=> NULL ["Sales_order_delivery_date"]=>
string(32) "2013-08-15T13:05:10.210127+03:00" ["Sales_order_delivery_text"]=>
NULL ["Sales_order_description"]=> NULL ["Sales_order_id"]=> int(0)
["Sales_order_number"]=> int(0) ["Sales_order_ordermark"]=> NULL
["Sales_order_state"]=> int(0) ["Sales_order_totalsum"]=> int(15)
["Company_location_id"]=> int(1) ["Currency_code"]=> string(3) "EUR"
["Delivery_method"]=> int(0) ["Language_code"]=> string(3) "FIN" ["OrderRows"]=>
object(stdClass)#6 (4) { ["Row_productcode"]=> string(6) "SARJA5"
["Row_amount"]=> int(1) ["Row_Total"]=> int(15) ["Row_Unitprice"]=> int(15) }
["Payment_term"]=> int(0) ["Person_invoice_res_person"]=> int(0)
["Person_seller_number"]=> int(0) ["RowCount"]=> int(0)
["Sales_order_confirmed_bit"]=> bool(false) ["Sales_order_currency_rate"]=>
string(1) "1" ["Sales_order_customer_address1"]=> string(11) "Koulukatu 1"
["Sales_order_customer_address2"]=> NULL ["Sales_order_customer_address3"]=> NULL
["Sales_order_customer_contact"]=> NULL ["Sales_order_customer_country"]=> NULL
["Sales_order_customer_name1"]=> NULL ["Sales_order_customer_name2"]=> NULL
["Sales_order_customer_number"]=> int(0) ["Sales_order_customer_ordernumber"]=>
NULL ["Sales_order_customer_reference"]=> NULL ["Sales_order_delivery_code"]=>
NULL ["Sales_order_delivery_customer_address1"]=> NULL
["Sales_order_delivery_customer_address2"]=> NULL
["Sales_order_delivery_customer_contact"]=> NULL
["Sales_order_delivery_customer_country"]=> NULL
["Sales_order_delivery_customer_name2"]=> NULL
["Sales_order_delivery_customer_number"]=> int(0) ["Sales_order_delivery_term"]=>
int(0) ["Sales_order_material_bit"]=> bool(false) ["Sales_order_note"]=>
string(0) "" ["Sales_order_orderer_customer_address1"]=> string(13) "LÃ¤nsikatu
15" ["Sales_order_orderer_customer_address2"]=> NULL
["Sales_order_orderer_customer_address3"]=> NULL
["Sales_order_orderer_customer_contact"]=> NULL
["Sales_order_orderer_customer_country"]=> NULL
["Sales_order_orderer_customer_name1"]=> NULL
["Sales_order_orderer_customer_name2"]=> NULL
["Sales_order_orderer_customer_number"]=> int(0) ["Sales_order_our_reference"]=>
NULL ["Sales_order_packlist_bit"]=> bool(false) ["Sales_order_project_number"]=>
int(0) ["Sales_order_taxtype_bit"]=> bool(false) ["Sales_order_type"]=> int(0) }
```

Figure 7. Lemonsoft data as an object

A customer script was the first one to be created. It was called *LemonGetChangedCustomerList.php* because of the function *GetChangedCustomerList($date)*. The function returns a list of customers that is modified based on the date parameter. Also new customers are considered "modified". When a script is executed it needs to have a correct date as the parameter. To achieve this, another script had to be developed that gets the real time date and subtracts one day to make it possible to fetch the latest modifications. Code example 4 shows how the date parameter is created and one day is subtracted.

```
include 'LemonCreateOrdersTest.php';
include 'LemonGetChangedProductsTest.php';
include 'LemonGetChangedCustomersTest.php';
$year = date("Y");
$month = date("m");
$day = date("d");
$hour = "00";
$minute = "00";
$second = "00";

$date = new DateTime($year . "-" . $month . "-" . ($day) . " " . $hour . ":" .
$minute . ":" . $second);
date_modify($date, '-1 day');
// var_dump(date_format($date, 'Y-m-d H:i:s'));


GetChangedProductsByDate(date_format($date, 'Y-m-d'));
GetChangedCustomersByDate(date_format($date, 'Y-m-d'));
CreateOrders(date_format($date, 'Y-m-d H:i:s'));
```

Code example 4. Magento date script

After a list of customers has been returned a check has to be made to see if some of the returned and required values are null. If that is the case, the data cannot be inserted into Magento because of an error message appearing during execution. This is why the Lemonsoft users need to be trained to insert the right and required data.

After the values have been checked, each variable that needs to be inserted into Magento from the object is separated and saved in a more logical variable. For example the *Product_group_code* field from Lemonsoft is saved in a variable called *product_category* as shown in code example 5. This is done to ease the understanding of the code.

```
$customer_email = $contact[$j]->Customer_contact_email;
```

Code example 5. How variables were saved

After a new customer has been inserted into Magento, an e-mail should be sent containing his/her login information. This was achieved by using Magento's own source code created from a user submitted example. The code is found in appendix 2. When creating new customers, a modified version of code example 6 was used.

```
$client = new SoapClient('http://magentohost/api/soap/?wsdl');

// If some stuff requires api authentification,
// then get a session token
$session = $client->login('apiUser', 'apiKey');

// get attribute set
$attributeSets = $client->call($session, 'product_attribute_set.list');
$attributeSet = current($attributeSets);


$result = $client->call($session, 'catalog_product.create', array('simple',
$attributeSet['set_id'], 'product_sku', array(
    'categories' => array(2),
    'websites' => array(1),
    'name' => 'Product name',
    'description' => 'Product description',
    'short_description' => 'Product short description',
    'weight' => '10',
    'status' => '1',
    'url_key' => 'product-url-key',
    'url_path' => 'product-url-path',
    'visibility' => '4',
    'price' => '100',
    'tax_class_id' => 1,
    'meta_title' => 'Product meta title',
    'meta_keyword' => 'Product meta keyword',
    'meta_description' => 'Product meta description'
)));

var_dump ($result);
```

Code example 6. Magento product creation code example /11/

The product data exchange script was almost the same as the customer script but also images need to be inserted into Magento. To insert images with a new product an API specific call named *product_media.create* was used. In this script a multicall version of the SOAP API was used since there were two API calls (*catalog_product.create* and *product_media.create)* used at the same time. The code of the multicall & image creation is found in appendix 1.

When the order script was created, problems arose. First of all the scripts start by fetching order data from Magento and then the data needs to be inserted into Lemonsoft. When trying to save the created order, a fatal error arose. Different solutions were tried but unfortunately the problem never got solved using PHP. A solution to this problem was to create the script using Visual Basic (VB) as the programming language. A code example of how to create the script was found in Lemonsoft documentation which was executed. In code example 7 you can see the code.

```
Dim oService As New Lemonsoft.SalesOrderServiceClient
Dim oSalesOrder = oService.NewSalesOrder(0)

oSalesOrder.Sales_order_ordered_customer_address1 = "Länsikatu 15"
oSalesOrder.Sales_order_customer_address1 = "Koulukatu 1"

Dim oRow = oService.NewSalesOrderRow(oSalesOrder)

ReDim oSalesOrder.OrderRows(1)

oRow.Row_productcode = "Lippa lakki"

oSalesOrder.OrderRows(oSalesOrder.OrderRows.Length - 1) = oRow

oService.SaveSalesOrder(oSalesOrder)
```

Code example 7. Lemonsoft VB code example /15/

Since the Magento API can also be used with VB, the problem could be solved by just creating a script written in VB, C# or PHP. These languages can be used by scheduled tasks to execute scripts.

After the order script had been executed an order was created in Lemonsoft, but with a wrong *sales_order_state*. This was because the order had no amount in the example code and Lemonsoft throws the orders straight through the system, which means that the order is at that point invoiced, collected and shipped. The *sales_order*_state should be 0 instead of 4 so this was solved by adding some more lines to the code, most importantly the amount of products and the total cost. This caused the order to be moved from Lemonsoft's *myyntitilaukset* (sales orders) section to the *myyntitilauskeskus* (sales order center) where the orders can be processed. /12/

The e-mail containing the order data that needed to be sent after each new order was achieved by enabling a Magento built-in function that sends an e-mail to the sales administrator using a pre-defined address. A successful test was made using Mercury Mail, which comes with XAMPP.

6.2 ASP.NET solution

The ASP.NET extranet solution was tested to see how time consuming it really is. To create the extranet from scratch was always a possibility.

The tests were done by connecting Visual Studio 2012 straight to the Lemonsoft database but this proved to be a bad idea since there is a possibility that the database will be changed by Lemonsoft updates in the future, which could mean that the extranet will not work properly. Instead of connecting straight to the database, the tests should have been done by using the Lemonsoft web service, which is probably not going to change.

Visual Studio 2012 was connected to the Lemonsoft interface by adding it as a service reference to be able to use its functions during the tests as seen in figure 8.

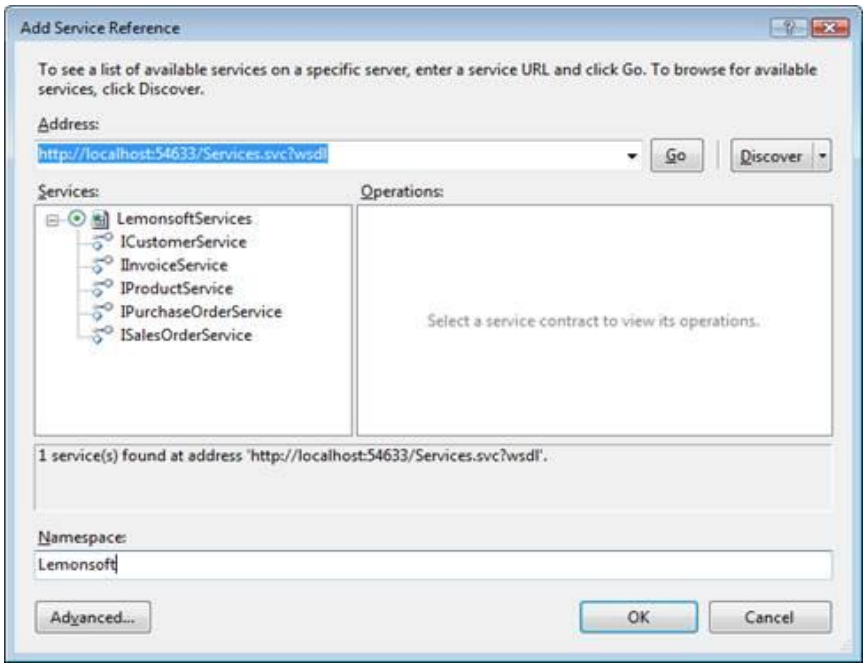

Figure 8. Lemonsoft service reference example /27/

The plan was to use the Lemonsoft web services in the code behind section of ASP.NET for each page load instead of fetching the data straight from the database. Although the usage of the web services was not tested in a larger test solution, it is possible to successfully implement them.

When testing the database version of the above mentioned solutions, an ASP.NET master page was used so the layout would always be the same, only the content section would change with the requested information.

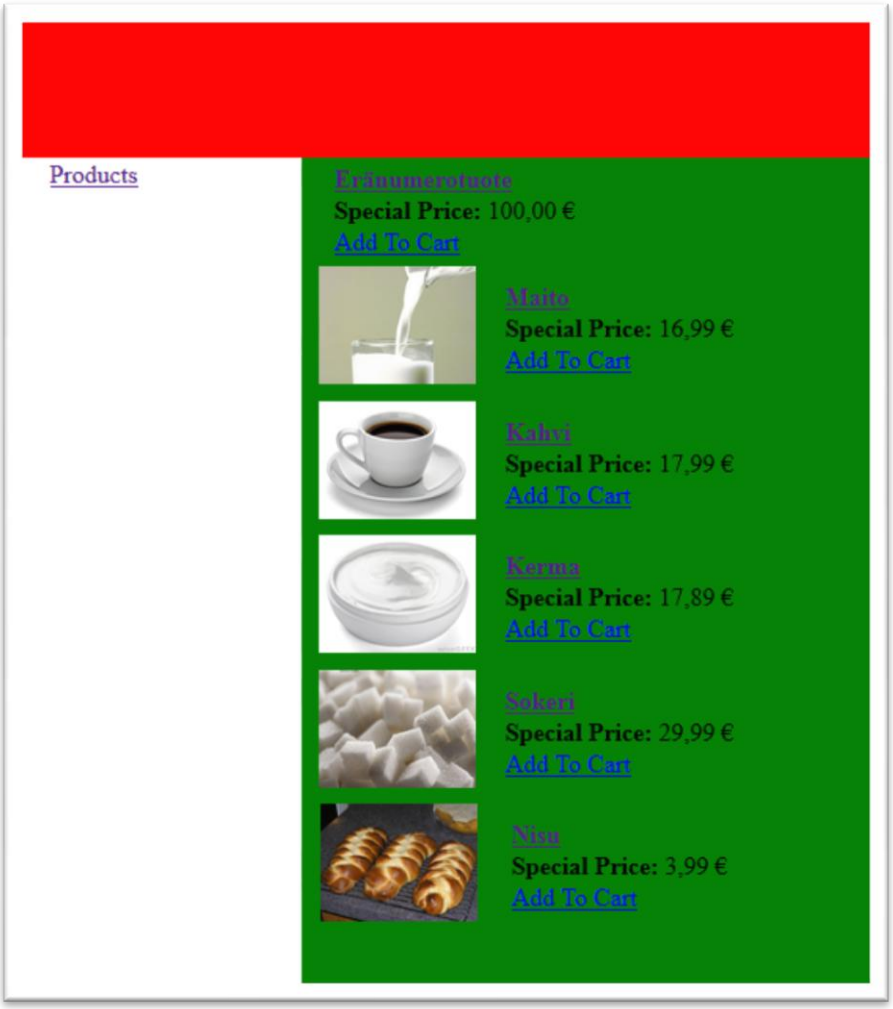

Figure 9. ASP.NET test solution, green color indicates the content part of the page

In figure 9 the test data was fetched using a *SQLDataSource* found in the basic toolbox of an ASP.NET solution. To get everything done was time consuming and required some studying because of the lack of ASP.NET knowledge. Using ASP.NET seemed to be an easy way to create web pages. Other pages than the product pages were also created to test this functionality.

To create the extranet using ASP.NET proved to be a very feasible solution. Unfortunately there was not enough time to test the ASP.NET solution any further.

6.3 Lemonsoft interface

If the customers are to be able to use Lemonsoft, they have to be trained in order to know which fields are required, because of the errors mentioned in chapter 6.1.1. This is vital to ensure the successful use of the ERP extranet integration. Since Lemonsoft offers training to companies, Fouga IT must train their customers on how to use Lemonsoft.

Lemonsoft itself is a very simple and clear software requiring only basic Windows usage knowledge of the user. Lemonsoft was tested by creating dummy data like products and

customers, and also orders, created with web services, were processed. This proved that Lemonsoft was very easy to use and a good useful tool for companies. Although Lemonsoft has a lot of different use possibilities, it was fairly easy to navigate and find what was wanted.

Irrespective of whether Fouga IT chooses Magento or ASP.NET as the platform it is still good that the customers are trained to fill in the data properly, because otherwise Magento will create a fatal error when exchanging data. For the ASP.NET solution the fields are not required but are used by the ASP.NET solution when e.g. products are shown. Although all the data in an ASP.NET solution will be in real time, empty fields could still cause a problem. Luckily they have not so far. /14/

7 Results

After all this testing and researching a conclusion is to be made. It is still hard to say which of these solutions is the better one. Both alternatives should be created to be able to give a recommendation with 100% certainty. The opinions about these two different solutions have varied. Most days Magento seemed to be the best solution, and now in the later stages ASP.NET has shown its possibilities.

Both solutions require a lot of work since Magento needs to be modified according to the needs and the ASP.NET solution is to be created from scratch as a tailored solution. The biggest upside of an ASP.NET solution is not to have nightly scheduled tasks, as all data is in real time straight from Lemonsoft. With the data in real time there is no need for product deletion, no scripts are needed to be run, and the orders require nothing but some coding to send the order straight to Lemonsoft.

After this amount of research a conclusion is to be made. The suggestion is to choose ASP.NET and to create the extranet from scratch because it seems a lot easier, cleaner and smoother to create a tailor-made solution from start to finish compared to Magento and its restrictiveness. This gives Fouga IT a lot of possibilities in terms of layout, styling and flexibility. With no nightly updates the solution is run without much strain on the server because the data is in real time instead of being moved at night. The data size is also smaller, faster and more compact. With an ASP.NET solution it is possible to easily handle the data exchanges and create a login either using Lemonsoft's session IDs or creating them yourself. Categories must be created by hand but the data can be fetched using the Lemonsoft web services. Only the product images may cause some problems but they should be easy to solve. Also a shopping cart is required for a proper extranet. The shopping cart could be implemented creating a separate database.

This solution requires a lot of ASP.NET knowledge because the extranet will be pretty technical in terms of coding, but so is the Magento version and its modules.

It is sad that an E-commerce platform as good as Magento proved to be a bit stuttering when integrated with an ERP. A stripped down Magento is not as good as Magento itself without modifications. The biggest minus with this integration using Magento is the nightly data transfers that need to be done. If they were not needed Magento would probably be the best choice.

Keep in mind that this is only a suggestion. Fouga IT has free hands to choose whichever solution they want, even another platform than the ones researched.

7.1 Discussion

During this process a lot was learned. Since I've used PHP before I had the knowledge required to create and test the scripts. My web design background and studies in ASP.NET also proved to be very helpful during the work on this thesis.

As I already knew the basics in PHP there was not that much to learn since all of the scripting was just basic PHP. The only thing I had to learn was objects in PHP. But after learning how to use objects the testing scripts were easily created. Using the API gave me a lot of knowledge of how to use APIs. Earlier I had not used APIs except once in a school project.

I also learned how to work within a company as I participated in weekly meetings with the staff and did a weekly report of what I had already done and what I still needed to do.

With an ASP.NET solution come a lot of possibilities in the future. ASP.NET is a new style of programming that is not used that much within Fouga IT. As far as I know from speaking with the staff, improvement is always wanted and also the ability to be able to provide anything the customers want.

In the future the customer might want a web shop or something similar. If this is the case we will have to build it separately from the extranet and probably use Magento or something else with a complete checkout process and secure use of credit cards / PayPal.

There were also other possibilities instead of using ASP.NET like creating the extranet in PHP using a framework like *CodeIgniter* or similar. It just seemed illogical to create the extranet in PHP from scratch since the E-commerce platforms were written in the same language. As mentioned earlier the order script could have been written in another language than PHP to solve the problem.

The testing phase took a lot of time and there would have been many other things to test but the most important things were tested. Unfortunately there was not enough time to test everything. Each of these extranet solutions is possible and the problems can be solved. Whether the solutions mentioned are the best ones or not is something that can be discussed further.

During the research phase I found a lot of problems that no one had thought of, some of them are mentioned and solved in chapter 4. Probably the biggest problem was the product deletion from Magento since it needs to be scripted or the products will somehow have to be hidden by creating a module.

Things that could have been tested were the execution of the scripts by Windows Scheduled Tasks, the use of the Lemonsoft web services in the code behind section in the

test extranet, image caching/base_64 encoding and a lot more. The testing was just done in order of importance.

The ASP.NET solution was not researched as thoroughly as the Magento solution because there were not really any data exchange issues. This is why the Magento solution was discussed more thoroughly than the ASP.NET solution. With no data exchange issues the problems with the ASP.NET solution were not as big as those with Magento.

8 References

1. *What is E-Commerce* (n.d.) http://www.networksolutions.com/education/what-is-ecommerce/ (fetched 18.7.2013 at 12:55)

2. *Johdanto Lemonsoftiin* (n.d.) http://info.lemonsoft.eu/LemonNethelp/#!Documents/johdantolemonsoftiin.htm (fetched 5.8.2013 at 14:34)

3. *Johdanto pääkäyttäjälle* (n.d.) http://info.lemonsoft.eu/LemonNethelp/#!Documents/johdantopkyttjlle.htm (fetched 5.8.2013 at 12:01)

4. *Lemonsoft –ohjelmien käyttätarkoitus* (n.d.) http://info.lemonsoft.eu/LemonNethelp/default.htm#!Documents/lemonsoftohjelmienky.htm (fetched 5.8.2013 at 11:29)

5. *Lemonsoft arkkitehtuuri* (n.d.) http://info.lemonsoft.eu/LemonNethelp/#!Documents/lemonsoftinarkkiteht.htm (fetched 5.8.2013 at 11:29)

6. *Palvelurajapinnat (Web Services)* (n.d.) http://info.lemonsoft.eu/LemonNethelp/#!Documents/palvelurajapinnatweb.htm (fetched 5.8.2013 at 08:09)

7. *Application Programming Interface* (n.d.) http://en.wikipedia.org/wiki/Application_programming_interface (fetched 30.7.2013 at 12:13)

8. Roos, Dave, *How to leverage an API for conferencing* (n.d.) http://money.howstuffworks.com/business-communications/how-to-leverage-an-api-for-conferencing1.htm (fetched 30.7.2013 at 13:59)

9. *Magento API SOAP Introduction* (n.d.) http://www.magentocommerce.com/api/soap/introduction.html (fetched 30.7.2013 at 15:15)

10. *Esimerkkejä eri ohjelmointikielillä* (n.d.) http://info.lemonsoft.eu/LemonNethelp/#!Documents/esimerkkejeriohjelmo.htm (fetched 30.7.2013 at 16:21)

11. *Magento API SOAP Catalog Product* (n.d.) http://www.magentocommerce.com/api/soap/catalog/catalogProduct/catalog_product.create.html (fetched 30.7.2013 at 14:30)

12. *Myyntitilauksen kenttäkuvaukset* (n.d.)
http://info.lemonsoft.eu/LemonNethelp/#!Documents/myyntitilauksenkentt.htm (fetched 31.7.2013 at 09:39)

13. *XAMPP* (2013)
http://en.wikipedia.org/wiki/XAMPP (fetched 31.7.2013 at 10:12)

14. *Perustaidot* (n.d.)
http://info.lemonsoft.eu/LemonNethelp/#!Documents/perustaidot.htm (fetched 5.8.2013 at 14:40)

15. *Osto- ja myyntitilaukset* (n.d.)
http://info.lemonsoft.eu/LemonNethelp/#!Documents/ostojamyyntitilaukse1.htm (fetched 7.8.2013 at 14:11)

16. *System Requirements* (n.d.)
http://www.magentocommerce.com/system-requirements (fetched 8.8.2013 at 10:04)

17. *ASP.NET Hosting Requirements* (n.d.)
http://www.ehow.com/about_5285462_asp-net-hosting-requirements.html (fetched 8.8.2013 at 10:09)

18. *Fouga IT Yritysesitys* (*2013)*
Fouga IT internal documentation

19. Organisaatioiden välinen tiedonsiirto (n.d.)
http://fi.wikipedia.org/wiki/Organisaatioiden_v%C3%A4linen_tiedonsiirto (fetched 12.8.2013 at 10:58)

20. Website – Its importance for business (n.d.)
http://resources.bravenet.com/articles/profit/advertising/website__its_importance_for_business/ (fetched 12.8.2013 at 11:06)

21. *Magento – About us* (n.d.)
http://www.magentocommerce.com/company/ (fetched 12.8.2013 at 11:36)

22. *Magento* (n.d.)
http://en.wikipedia.org/wiki/Magento (fetched 12.8.2013 at 11:36)

23. *WordPress* (n.d.)
http://en.wikipedia.org/wiki/WordPress (fetched 12.8.2013 at 12:51)

24. *OpenCart* (n.d.)
http://de.wikipedia.org/wiki/OpenCart (fetched 12.8.2013 at 13:04)

25. Valade Janet (2010). PHP & MySQL For Dummies, 4th Edition, Wiley Publishing ISBN: 978-0-470-52758-0

26. *Why is ERP important to a company?* (n.d.) http://www.eresourceerp.com/Why-is-ERP-important-to-a-company.html (fetched 20.8.2013 at 10:50)

27. *Yleiset* (n.d.) http://info.lemonsoft.eu/LemonNethelp/#!Documents/yleiset.htm (fetched 20.8.2013 at 11:28)

28. *Trusted by leading brands worldwide* (n.d.) http://www.magentocommerce.com/product/enterprise-whos-using-magento (fetched 26.8.2013 at 09:44)

APPENDICES

I. Image of Magento API using multicall and creating images
II. Sending email with customer info using Magento's own code

```
$newImage = array(
        'file' => array(
                'name' => $product_name,
                'content' =>
base64_encode(file_get_contents('C:/Users/bill.kala/Downloads/xampp-win32-1.8.1-
VC9/xampp/htdocs/magento/media/' . $product_ean . '.jpg')),
                'mime'    => 'image/jpeg'
        ),
        'label'    => 'Test Image 4',
        'position' => 2,
        'types'    => array('small_image', 'image', 'thumbnail'),
        'exclude'  => 0
);

$imageFilename = array('product_media.create', array($product_ean . ' ',
$newImage));

// Newly created image file
var_dump($client->call($session, 'product_media.list', $product_ean . ' '));

$calls = array(
array('catalog_product.create', array('simple', $attributeSet['set_id'],
$product_ean, array(
'categories' => array(2),
'websites' => array(1),
'name' => $product_name,
'description' => $product_description,
'short_description' => $product_description,
'weight' => $product_weight,
'status' => '1',
'url_key' => $product_name,
'url_path' => $product_name,
'visibility' => '4',
'price' => $product_price,
'tax_class_id' => 1,
'meta_title' => $product_name,
'meta_keyword' => $product_name,
'meta_description' => $product_name))),
array('product_stock.update', array($product_ean, array('qty'=>$product_stock,
'is_in_stock'=>1))),
array('product_media.create', array($product_ean . ' ', $newImage)),
array('product_media.update', array(
        $product_ean . ' ',
        $imageFilename,
        array('position' => 2, 'types' => array('small_image', 'image',
'thumbnail'))))
);
$client->multiCall($session, $calls);
```

```
$pwd_length = 7; //auto-generated password length

error_reporting(E_ALL | E_STRICT);
$mageFilename = 'app/Mage.php';
if (!file_exists($mageFilename))
{
        if (is_dir('downloader'))
        {
                header("Location: downloader");
        }
        else
        {
                echo $mageFilename." was not found";
        }
        exit;
}
require_once $mageFilename;
Varien_Profiler::enable();
Mage::setIsDeveloperMode(true);
ini_set('display_errors', 1);
umask(0);
Mage::app('default');

$customer1 = Mage::getModel('customer/customer');
$customer1->setWebsiteId(Mage::app()->getWebsite()->getId());
$customer1->loadByEmail($customer_email);

if(!$customer1->getId())
{
  //We're good to go with customer registration process
  $customer1->setEmail($customer_email);
  $customer1->setFirstname($firstname);
  $customer1->setLastname($surname);
  $customer1->setPassword($customer1->generatePassword($pwd_length));
}

//if process fails, we don't want to break the page
try
{
  $customer1->save();
  $customer1->setConfirmation(null); //confirmation needed to register?
  $customer1->save(); //yes, this is also needed
  $customer1->sendNewAccountEmail(); //send confirmation email to customer?
}
catch(Exception $e)
{
   Mage::log($e->__toString());
}
```