Chapter 13
Architecture for the Reengineering of Legacy Point of Sale Terminals through Web Services for the Reduction of Transaction Fees

Erik-Jan Monshouwer
Yacht, The Netherlands

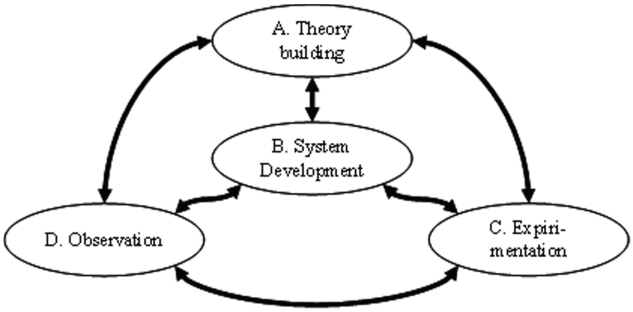
Raul Valverde
Concordia University, Canada

ABSTRACT

With the conventional legacy POS payment transaction method, vendors are bound to a payment institute in their region and can only use relatively expensive dedicated or slow dial-up lines to their financial institute. This chapter covers the work to produce an architecture that shows how to reengineer traditional point of sales terminal payments in order to adapt these for payment over the Internet through Web services. With the use of Web services for payment transactions, vendors will get more freedom to choose their provider and the services they take without having to throw away their legacy applications. Given the globalization of the economy, vendors can negotiate services and fees with payment providers all over the world. Literature research and prototype tests and evaluation in this project show that transactions fees and performance of POS terminal payments transactions trough Web services can be competitive to conventional payment transactions methods and create flexibility for vendors POS terminal application. Vendor's available Internet connections and the Web services standards in the market can be used for payment transactions. With Web services, the system can be created and changed relatively quickly and simply if the right skills are available.

DOI: 10.4018/978-1-4666-0155-0.ch013

Figure 1. Information System research to phenomenon of interest (source: Burstein)



1. INTRODUCTION

This research delivers a web services architecture for the integration of legacy point of sale (POS) terminals with financial institutions. Businesses are interested in the freedom that this may offer because vendors can select payment providers from all over the world to find the best fee and services.

With the conventional legacy POS payment transaction method, vendors are bound to a payment institute in their region and can only use relatively expensive dedicated or slow dial-up lines to their financial institute. To design a web service, structure is needed to implement and support the system. For this, an architecture will be provided in this chapter. Different sources describe the needs of business for integration and the use of an architecture. OMG (2009) describes the needs in a RFI (Request for Information), especially the endorsements in the document. Via the RFI, OMG asks their working group what they like to see in the standard architecture.

The next section will describe the research methodology of the project. After, the literature review describes the conventionally payment processing method, web services and architecture frameworks with significant attention. Following good information systems practices, this paper will describe the architecture and the prototype with system's specification and design, implementation, test and evaluation. The last section expresses the conclusions and future research.

2. RESEARCH METHODOLOGY

Various approaches can be used for the research. The advantages of each methodology depend on the environment of the application and organization (different domain and focus). A variety of research methods are used in most information system research projects. The information system research approach used in this project is based on the method described by Burstein and Gregor (1999) because it presents a group of research methods. They demonstrated the importance of recognizing the "System Development" approach and relevant criteria for guiding the validity and worth of such work. Following the "System Development" approach form Burstein and Gregor (1999), Figure 1 (Nunamaker et al.1990, cited by Burstein et al. 1999) is used.

This form of research can be regarded as an action research which is suitable for this project because "System Development" recognizes other research fields next to system development, supports rapidly changing environments, the use of the prototype and the natural way of approach. The term "action research" is used because the researcher is an active participant in the SD process. Burstein and Gregor (1999) point out that the result of action research is usually associated

with the creation of knowledge about the system, while at the same time attempting to change it.

The process is iterative, i.e., the cycle of action and reflection continually generates new insights. Unified process, is used for development of the system. Linda Knight (et al. 2001) has examined the relative strengths and limitations of existing system development methodologies, from the traditional Waterfall to Rapid Application Development and some of the newest rapid response methods. These methods are focused on the system design.

Following the System Development approach, experimentation and observation in the project is to approve the literature research, design and architecture. Evaluation of the design and architecture is done by the use of the use cases. By building a prototype the integration of the particular reviewed technologies are proven and the system can be tested by novice users.

The suggested criteria of the System Development approach described by Burstein and Gregor (1999) are used for evaluation of the system development work. The issues used are; significance, internal validity, external validity, objectivity and reliability.

3. LITERATURE REVIEW

3.1 Introduction

World wide there are a lot of different standards for electronic payments. These differences consist of communication protocols, currencies and bank-organization standards which are historically formed in small regions. With globalization of economies, borderline payments are more common and will probably grow in the future. Other worldwide movements, like the implementation of the euro, also stimulate exchanging money amongst worldwide regions.

This section of the chapter describes the conventional way of vendor's POS-payment transactions that is common in legacy systems and how POS-payments could be done through web services. The literature review is focused on the world-wide standards and the architecture framework. Other business issues, like costs are also mentioned to make clear what the considerations are when web services are investigated.

3.2 Conventional POS Terminal Communication

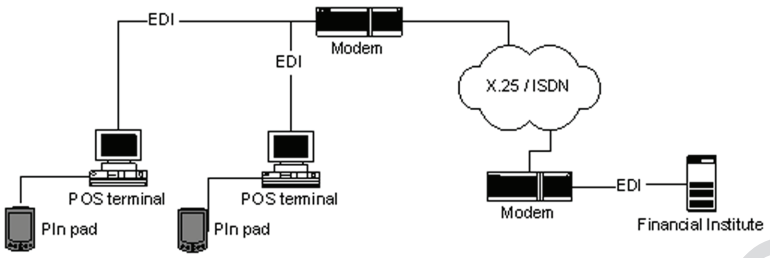
The conventional electronic payment transaction method is used worldwide from the early eighties. POS terminals communicate via dedicated communication lines to one specific financial institute. All transactions are done via directly coupled connection lines. These financial institutions are mostly large players in a country or region. For example, Interpay in the Netherlands is originally founded by Dutch banks in 1994 (Interpay, 2009) to set up, manage and develop an efficient payment infrastructure.

Almost all of the Dutch payment transactions are done through Interpay so you could call it a monopolist. This also applies to some other European countries (like Banksys in Belgium and Cetrel in Luxemburg). In Germany, more financial Institutions are active on payment transaction market, each operating its own network.

During the last years, governments are trying to make the payment transactions market more competitive and the old monopolists have now to deal with other payment institutions in their region.

Most conventional electronic payment Institutions are region-bound and settled prominently in their area. Technologies used for payment transactions thereby are different in origin. When vendors want to switch to another institute (read network), interfaces have to be changed. Changing interfaces is only useful when the service is comparable and total costs are lower after changes in software and infrastructure are made.

Figure 2. General network infrastructure (based on information of Interpay)



3.3 The Legacy Network Infrastructure

The network infrastructure will be different for each vendor but the basics are the same. Transactions are done on the POS terminal or through the electronic card reader. Via EDI, communication is set-up over a dedicated line to the financial institute which will respond if the transaction is successfully completed or not. In large stores communication can be managed via a central host which handles communication form all POS terminals in store to the financial institute. Figure 2 shows a typical network infrastructure with 2 POS terminals.

3.3.1 Security

Before the use of electronic payments, customers handle over a credit or debit-card and the vendor collect the card number and card expiration date. Now a day, most vendors have moved to electronic card readers linked by communication lines directly to their financial institute. Card details are collected from the magnetic strip when the card is "swiped" through a card reader machine. The shop assistant types in the sales amount and details are passed to the financial institute for approval. Card readers are normally certified by the financial Institute to minimize fraud on the store side.

Actually, the customer is not always actually present in a shop or at the POS terminal for a transaction. For example when a transaction is made by placing an order over the phone or by internet-order, the customer does not need to be present at the POS terminal. This situation is known as a "customer not present" transaction. Financial institutions normally make a distinction between "customer present" and "customer not present" transactions, as there is a potential increase in fraud when the customer does not present the card or sign a sales voucher at the point of sale. This is why most institutions ask for more commission to carry out these transactions. This problem non-existent in the case of POS terminal payments in stores because the customer is always present.

With a dedicated connection line security is guaranteed with the use of a list of telephone numbers at the receiver site (financial institute) which are authenticated to make a connection. For extra security this is commonly used in combination with a token which is given with the payment transaction(s).

3.3. 2 Transactions Fees

The costs of the transactions strongly depend on a few variables. For a shop that sells a large amount of relatively cheap articles the average transaction costs are not the same for a shop that sells less expensive products (with the same total amount of business). Financial Institutions do service a lot of different options which could fit vendor's needs. To get a clear view of the total costs one has to look at transactions volumes and their costs

Table 1. Credit card fees (Source DNB 2009)

Country	Costs debit card transaction in 2009
Australia	0,30 euro
Belgium	0,06 euro
France	0,9%
Germany	0,3% (minimum of 0,08 euro)
Italy	0,9%
Netherlands	0,07 euro
United Kingdom	0,08 euro

and recurring costs. This will also influence the need of the connection to be leased to the financial institute. To get an impression of payments costs for POS terminal debit card transactions, an overview is given in Table 1 (Source DNB 2009). These costs are the transactions costs, which is without vendor's costs of communication lines, card reader, POS terminal, human resources, et cetera.

As shown in Table 1 fees issued differ over the countries. Although the average transaction charge is lowest in Belgium, this does not imply that all transactions processed in Belgium are the cheapest. The fees also depend on variables like the amount of money involved and the number of transactions processed in each period. In Germany, average POS charges clearly exceed those in the Netherlands, indicating that more competition in debit card infrastructure does not automatically result in lower tariffs. Vendors do have to find out which combination of services is the best choice for their situation. The final solution depends on used payment systems in store, possibilities of connection lines between store and financial institute and the alternative services which the financial institutions offer (rent of hardware, fees, maintenance, et cetera).

As a general rule of thumb, an electronic payment costs only from one-third to one-half as much as a paper-based payment. If a country moves from a wholly paper-based payment system to an nearly all electronic system, it may save 1% or more of its GDP (Gross Domestic Product) annually once the transition costs are absorbed. While economies of scale exist for paper-based payments, they are small compared to electronic payment methods (Humphrey et al. 2003).

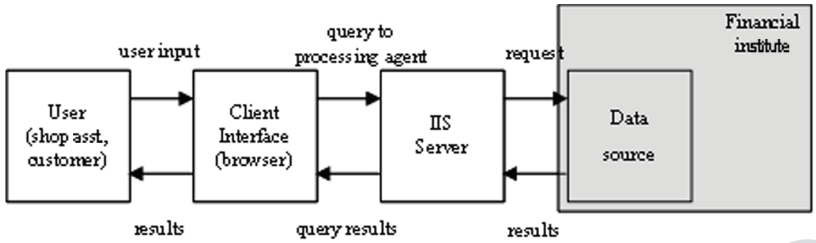
3.4 POS Terminal Communication through Web Services

This paragraph explains how payment transactions could be done via web services. Through integration of POS terminals with financial institutions via web services, transfers of money could take place to any part of the world. Providing these services, financial institutions become Internet Payment Service Providers (PSP). Vendors will have more freedom about the choice of financial institutions and in finding the best rate from the best financial supplier to reduce their payment transaction costs.

Evolutions in the markets for banking services are part of a strategic trajectory in the sector which started long before the advent of Internet-based e-commerce and in which the Internet is more of a catalyst than the prime driver (Munck et al. 2001). Munck also stated that much of this involves the consolidation of financial institutions, spurred on by a variety of factors like the liberalization of national markets, the internationalization of banks and of the banking client-base, and the harmonization of the European market. Along with consolidation, boundaries between financial institutions and specific products have blurred. Major Banks have now positioned themselves as integrated suppliers of a full range of financial services, usually in several national markets.

A simplified process overview is given in 3. The end users (vendor and customer) add data in the system via a user interface. The user interface will ask the web server to request a query at the financial institute through a web service. The financial institute will process the query and responses via the web service to the client.

Figure 3. Simplified process overview (based on information of EFSnet 2004)



There are few more sophisticated web services models for payment systems such as the Cohen et al. (2005) and the Dahlberg et.al. (2008) models.

Cohen et al. (2005) proposed a web services model for the implementation of web based payment systems. The model makes few contributions to the basic model discussed in figure 3 by allowing the implementation of chargeable Web Services by payment providers and by providing the infrastructure to negotiate service rates in real time over the web.

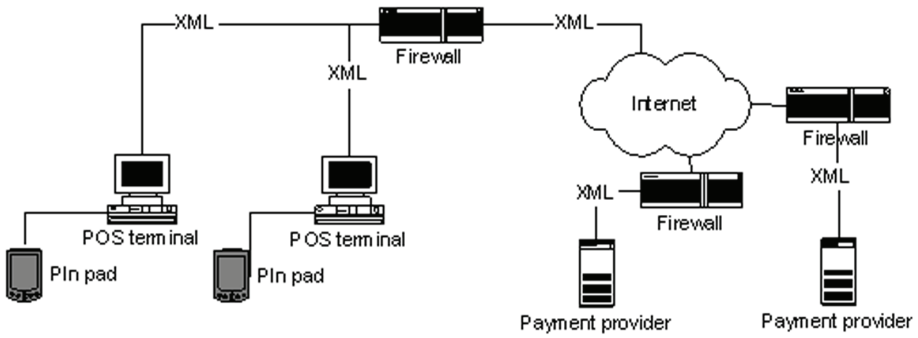
Dahlberg et al. (2008) proposed a more complex web services payment model that includes the implementation of a mobile of point sales services with the use of web services. The mobile point of sales can be used to support an M-commerce application.

Although the Dahlberg et al. (2008) and Cohen et al. (2005) models make excellent contributions to the basic web based payment model discussed previously, they will not be used for the prototype of this research as they cover features outside of the scope of this research. Nevertheless, they can be considered for future enhancements of the proposed architecture.

3.4.1 Network Infrastructure

As in the traditional legacy EDI payment method the network infrastructure for payments through web services will differ for each vendor. Payment transactions can be done on the POS terminal or directly through the electronic card reader. Via XML communication will be set up over the Internet to the payment provider. The payment provider will respond if the transaction is successfully completed or not. At both sides of the internet connections security measures are needed. Processing could also be handled by a central host for all communication to the payment provider. Figure 4 shows general network infrastructure with XML communication. It also shows that the

Figure 4. General network infrastructure with XML communication



Internet is used for communication where vendors can connect to different providers.

3.4.2 Transactions Fees

For transaction over the Internet some additional costs have to be made. A secure Internet connection should be available and POS terminal or card readers need to be upgraded or replaced to make XML transactions possible. In some situations investments in extra security will be needed like firewall software. This will depend on the type of Internet connection that is already used by the vendor.

As in the conventional method, the Payment Service Provider will ask a fee from their customers for payment processing. The more transactions a vendor makes the less it will cost per transaction. The choice of Payment Service Provider will depend on its cost and compatibility with the chosen e-commerce software solution. To give an impression: there are options available starting as low as 0.19 euro per transaction.

It is beyond the scope of this project to compare Payment Service Providers fees. The fees depend on the services taken from the Provider and will change rather rapidly with new market developments and techniques. There are several sites on the Internet, by example DTI & Scottish Enterprise (address is in the bibliography), which are specialised in comparing Payment Server Providers on their options. The Payment Server Providers will generally accept most types of business with a business bank account and an address that confirms the identity of the business. However, Payment Server Providers take sometimes 30-60 days to clear merchant's funds. Generic costs for acquiring are 1–5% per credit card transaction and 0,19 – 0,55 euro for debit cards.

3.4.3 Architecture Framework

When the complexity of a web system grows an architecture framework is useful to implement and support the system for planning, designing, building, deploying and maintaining. Points of reference are usually called frameworks. Architecture frameworks typically serve a strategic guide for the development of a system.

The Open Group Architecture Framework (TOGAF, 2009), defines an architecture framework as follows: "An architecture framework is a tool which can be used for developing a broad range of different architectures. It should describe a method for designing an information system in terms of a set of building blocks, and for showing how the building blocks fit together. It should contain a set of tools and provide a common vocabulary. It should also include a list of recommended standards and compliant products that can be used to implement the building blocks."

An overview of most used architectures frameworks are given by Xiaoying Kong (2004) and Frank Goethals (2004). They both showed clearly that an architecture is not just one model. It exists of architecture descriptions and dimensions. The architecture frameworks described by Kong and Goethals are useful for web services to separate the concerns of different participants in a development project. The perspectives of architecture frameworks are often classified in categories like structure (what), behaviour (how), location (where) and pattern.

Framework architectures with significant attention in the literature are "Framework for information system architecture" from John Zachman (1987), "The 4+1 view model of software architecture" from Philippe Kruchten (1995), "Recommended practice for architectural description of software-intensive systems" (IEEE147, 2000) from IEEE and the "Model Driven Architecture" (MDA) from OMG (2009).

Web service systems are not tightly coupled as conventional systems. This is why the architecture frameworks of Zachman, Kruchten and IEEE are not the best choice for web services. Other web systems natures, beside payment web services issues, also differ from conventional software

Table 2. WAAF matrix

	Structure (What?)	Behaviour (How?)	Location (Where?)	Pattern
Planning Architecture (Planner's Perspective)	List of things important to the business	List of processes the business performs	List of locations in which the business operates	-
Business Architecture (Business Owner's Perspective)	e.g. Business Entity Relationship Model	e.g. Business Process Model	e.g. Business Entity Location Model	e.g. Business Model Patterns
User Interface Architecture (User's Perspective)	e.g. User Interface Structure Model	e.g. User Interface Flow Model	e.g. User Site Map Model	e.g. Interface Templates, Navigation Patterns
Information Architecture (Information Architect's Perspective)	e.g. Information Dictionary	e.g. Information Flow Model	e.g. Information Node Location Model	e.g. Information Scheme Patterns
System Architecture (System Architect's Perspective)	e.g. System Functioning Module/ Sub-Module/ Server Page Structure	e.g. Workflow Model of Module/ Sub-Module/ Server Page	e.g. Site Mapping Model of Module/ Sub-Module/ Server Page	e.g. Design Patterns, Presentation styles
Web Object Architecture (Developers' Perspective)	e.g. Physical Object Relationship	e.g. Algorithms in Source Code	e.g. Network Deployment Model	e.g. COTS, Components, Code Library
Test Architecture (Tester's Perspective)	e.g. Test Configuration	e.g. Test Procedure	e.g. Test Deployment	e.g. Templates, Standards of Test Document

systems. These are described by Lowe (2004b). Lowe and Eklund (2002) stated that specifications for web systems are consequently very different from those for more conventional software systems. This is also the view of Kong. Kong (2004) stated that for web services architectures "Model Driven Architecture" or "Web Application Architecture Framework" (WAAF) could be used for web services because these architecture frameworks recognise the open and more modularised approach of web services.

The WAAF Matrix (2004) is presented in Table 2. Another web service issue which is recognised by WAAF is the short delivery timeframe. For small projects, like POS payment web services, a light weight matrix of WAAF or MDA can be used.

MDA is based on UML which seems insufficient to model user interface and some business software and UML and MDA are not fully understood by the IT community (Kong 2004). WAAF is not bound to any method or language.

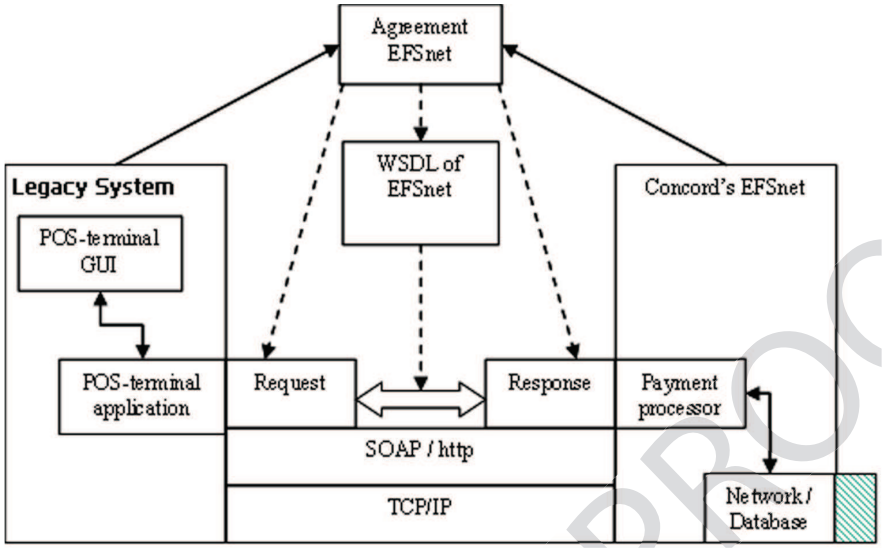
This project is focused on the vendor's side. To accomplish the integration of IFX, OFX and SWIFT standards the Context Interchange (COIN) framework could be used on the payment provider side. Jayasena et al. (2004) stated that the COIN architecture leverages a model of sources and receiver's contexts in reference to a rich domain model or ontology for the description and resolution of semantic heterogeneity.

4. DESIGN AND PROTOTYPE

4.1 Introduction

The prototype is deployed for POS terminals communication with EFSnet through web services. EFSnet is the payment service of Concord EFS which is a large commercial electronic transaction processor in the USA. They acquire, route, authorize, capture, and settle virtually all types of electronic payment, including debit transac-

Figure 5. Specific component architecture



tions for financial institutions and merchants. This choice means that all payment transactions the services and description files of the financial institute EFSnet are used.

A web page is used as the user-interface to simulate payments via a legacy POS terminal. The financial side will communicate with a financial institute via SOAP or XML files.

The prototype is developed for payment transactions with a test environment of the Concord's EFSnet. No real payment transactions will be transmitted within the prototype. Fixed credit card data, provided by the financial institute, is used for testing payment transactions over the Internet. The actors in the system are vendor, customer and financial institute (payment provider).

Features of the system are:

- Customer pays with credit card after vendor presents the amount of money to be paid and a reference number.
- In the test environment fixed Credit card data is used for testing purposes (test credit card number and expiration Year/Month).
- Internet Explorer 6 is used as user interface for the POS terminal prototype.

The features of the system are not:

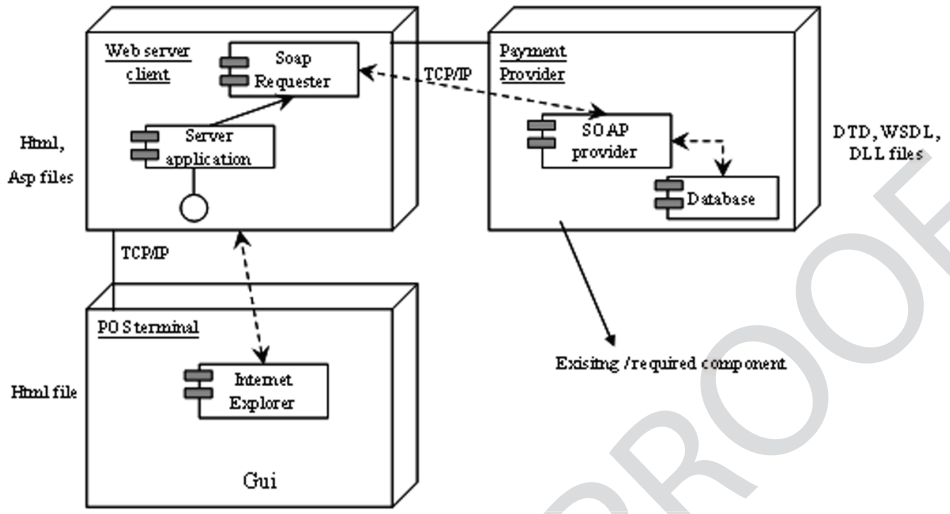
- Advanced POS terminal application: payments will be done at a prototype web page.
- Advanced payment security
- Create management information
- For use in other browser as Internet Explorer 6

4.2 Design and Web Service Architecture for Legacy POS Terminals

For the prototype, the WAAF architecture is used. The EFSnet Web services Interface specification (Concord, 2009) is used and the input and output parameters (WSDL file) are based on the specific environment of Concord's EFSnet. To make use of the EFSnet test environment, a test account at Merchant Services Login at EFSnet is applied. With this account unique ID's are created which are needed for communication with EFSnet.

The architecture consists of several components (Figure 5). In the agreement between the vendor legacy system and the payment provider

Figure 6. UML deployment diagram



is told that interaction will be done as described in EFSnet's WSDL file. Now the transport protocol and SOAP/XML vocabulary is agreed. The components are described in the next sub paragraphs.

4.3. Network Infrastructure and Security

The deployment diagram is illustrated in Figure 6. It shows what components are placed on which node. This structure has impact on the infrastructure and make visible what communication lines are needed

Vendor's legacy POS terminal is integrated with a POS-service that is placed on a web server with a dedicated Internet connection. The firewall and router are configured to communicate over the needed TCP ports. In principle the end user node (legacy POS terminal) can be anywhere as long as it can reach the web service server and it is trusted by the web service environment. The payment provider can be considered as a black box where only the interface is known. In reality, the payment provider will have a complex environment of systems, databases and networks (which are connected to other financial institutes).

SSL is used for secure communication between vendor and EFSnet. Here for a third party as VeriSign can be used. For security reasons, domain registration is needed in combination with SSL security.

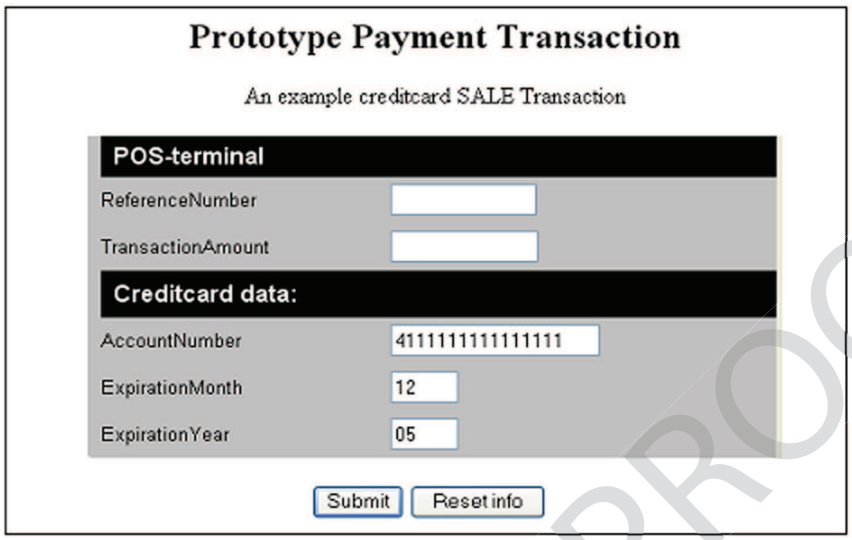
4.4 POS Services (Request)

Communication to EFSnet is set up as described in de WSDL file of EFSnet. An ASP file is used to process XML or SOAP files from vendor's web server to the financial institute. Also the processing of the responses of the requests to the POS terminal user interface is done by the same ASP page. The HTTP protocol is used as the transport protocol which is prescribed by agreement and the description file.

4.5 POS-Terminal (end User Client)

Presenting the form and validating the data fields is done with JavaScript in a HTML page which is placed on the web server (this is showed in the deployment diagram in Figure 5). On the back ground data is processed at the web server after submitting, with the use of an ASP page. The POS

Figure 7. POS terminal prototype user interface



services asp file is placed on the same server as the POS terminal user interface file. Advanced security measures can be added as when the specific environment is known.

The web interface can be used form every location as long there is connection with the Internet and the environment is trusted. A screenshot of the user interface is showed in Figure 7.

In fact, vendor's business application could be any application. The prototype will use a web-based interface. This way functionality can be added relative simple by adding fields and validation codes in the ASP file.

4.6 Financial Institute (Response)

The financial institute WSDL description file is placed on EFSnet web servers. Each SOAP message is checked by the WSDL file before it is transmitted. The data format is, before it is submitted, also validated in the user interface application. With the processing of the ASP file, an SOAP message is created and transmitted to the financial institute. Files received by the financial institute are transformed to the back end systems for each transaction.

If connection can be made and the financial institute authenticates the vendor successfully, the institute will try to process the data. If problems are encountered a response code is given back to the system. Input and output parameters are described in interface specification (Table 3) and can be used by vendor's application. These parameters will also be used as response code in the user interface

4.7 Implementation

In Table 4, all used software and standard protocols are presented. Vendors can chose other standards or tools which fit their organisation. The choice of tools will normally not influence the architecture. Hardware used for the implementation is mentioned in the sub paragraphs.

4.8 System, Infrastructure, and Security

Web server is a Microsoft Internet Information Server which is standard delivered with Windows XP. The system holds the latest service packs and security patches. All web service files are placed

on this single server which also will hold the User Interface files which are presented on the prototype POS-terminal system.

Normally domain registration is needed in combination with SSL. In the prototype a temporary SSL environment of VeriSign is tested. With a SSL test environment, all end users do have to import "Test CA Root" into their browser. The link to the instruction is added in the prototype user interface. When a regular SSL Certificate is purchased this "Test CA Root" procedure is not needed. For the test environment of EFSnet, secure transactions are not obliged.

For the prototype no advanced router or firewall is used and Internet connection is limited to 800 kbit/sec from Internet to web server and 245 kbit/sec from server to Internet. To make Internet-communication possible the firewall and the web server are configured to pass HTTP (TCP port 80) and HTTPS (standard TCP port 443). The router is configured that a virtual server is configuration to get request from the Internet to the web server.

4.9 Web Services via SOAP

An ASP file in the root directory of the web server processes the data to the Financial Institute after submitting the form data. A part of the content of the ASP file is presented in Algorithm 1. The Web address and ID's are left away for security reasons..

SOAP consists of three elements: an envelope that defines a framework for describing message contents and processing instructions, a set of encoding rules and a convention for representing responses. The use of SOAP reduces the code needed to request a credit card authorization or charge to a few lines (See Algorithm 1). Static data is written in the ASP file. The dynamical data needed is presented to the end user with the Prototype POS terminal interface.

Table 3. Input and output parameters CreditCard-VoiceAuthorize method

CreditCardAuthorize	
Input Parameters	**Output Parameters**
*StoreID	ResponseCode
*StoreKey	ResultCode
*ApplicationID	ResultMessage
TerminalID	TransactionID
*ReferenceNumber	AVSResponseCode
CashierNumber	CVVResponseCode
*TransactionAmount	ApprovalNumber
SalesTaxAmount	AuthorizationNumber
Currency	TransactionDate
*AccountNumber	TransactionTime
*ExpirationMonth	ReferenceNumber
*ExpirationYear	AccountNumber
CardVerificationValue	TransactionAmount
Track1	
Track2	
BillingName	
BillingAddress	
BillingCity	
BillingState	
BillingPostalCode	
BillingCountry	
BillingPhone	
BillingEmail	
ShippingName	
ShippingAddress	
ShippingCity	
ShippingState	
ShippingPostalCode	
ShippingCountry	
ShippingPhone	
ShippingEmail	
ClientIPAddress	
* Required field	

Table 4. Software used

Software function / protocol	Tool / standard / language
Web service protocol / language	Http(s), XML, SOAP, ASP
Scripting languages	Javascript
Operating system	Windows XP Professional SP2
Web server	IIS 5.0
Html editor	FrontPage 2000 / notepad / Crimson editor
XML-editor	XML Spy 2.5 / notepad
UML diagrams	MS Visio / Ms PowerPoint
Pictures editing	Ms PowerPoint / Ms Visio
Browser	Internet Explorer 6

4.10 Web Services via XML

An alternative is processing transactions by XML files. EFSnet communicates using XML as the native message format. Advantage of XML is that optional parameters can be used and that is faster. It is faster because there is no check on a description file. Instead of a WSDL file a DTD file is used to check the format but a XML request is directly send to EFSnet's DLL file

Next SOAP version used by EFSnet will also support optional parameters. An example of XML request code is presented in Algorithm 2. To view a request, an option (VIEW_XML_REQUEST = true) can be set in the asp file.

The prototype meets the objectives and the requirements. The prototype system is flexible in use: new functionality can be added easily and grow is relatively simple to realize. The used standards methods make it easy to adopt the design in other environments. SOAP is the preferred method because the minimal needed code but XML can also be used. Vendor's choice will depend on the preferences of the business application.

An architecture framework should be used for design, planning and maintenance. An architectural framework, like WAAF, is not fully required because of simplicity of the prototype and limited views. For complex environments the architecture for payment web services can be easily worked out in more views.

Security needs to be added over the system as extra layer over all components. SSL is a method to secure the transaction during transport over the Internet. Also other security measurements should be taken by the vendor (firewall, server security, logging and et cetera).

4.11 Practical Applications of the Proposed Architecture

The proposed architecture can have several applications in industry. One possible application is the integration of existing legacy POS terminals with a POS service that is capable of detecting in real time the different payment processing fees of multiple financial institutions, the terminal could send the payment to the financial institution with the lowest processing fee that is detected at the time the transaction is made. This application would allow an organization to strengthen its bargaining power for payment processing rates as financial institutions would need to compete for the transaction in real time and will be forced to offer competitive rates.

Another possible application is the reengineering of POS legacy systems in order to create a global point of sale terminal capable of processing payments in multiple currencies; the terminal

Algorithm 1. ASP and SOAP code

```
<%Option Explicit%>
<%
Dim objEFSnet  ' Declare SOAP client object
Dim WsdlUrl, StoreID, StoreKey, ApplicationID, 'etc. Declare Input Variables
Const TEST_MODE = True ' Make it easy to switch between Test or Production
Servers
ApplicationID = "Prototype"
ReferenceNumber = Request.Form("ReferenceNumber")
TransactionAmount = Request.Form("TransactionAmount")
AccountNumber = Request.Form("AccountNumber")
ExpirationMonth = Request.Form("ExpirationMonth")
ExpirationYear = Request.Form("ExpirationYear")
If (TEST_MODE) Then
    WsdlUrl = "https://testefsnet.concordebiz.com/EFSnet2.wsdl" 'Internet
    StoreID = "StoreID"
    StoreKey = "StoreKey"
Else ' Live Mode
    WsdlUrl = "https://efsnet.concordebiz.com/EFSnet2.wsdl"
    StoreID = "Issued Live Account StoreID Goes Here"
    StoreKey = "Issued Live Account StoreKey Goes Here"
End If
' Create the SOAP client object
Set objEFSnet = Server.CreateObject("MSSOAP.SoapClient")
' Setting this property is required before calling MS SOAP From Server
objEFSnet.ClientProperty("ServerHTTPRequest") = True
' Initialize the SOAP client
objEFSnet.mssoapinit WsdlUrl
' Call EFSnet Web Service Method: CreditCardAuthorize
ResponseCode = objEFSnet.CreditCardAuthorize(StoreID, StoreKey, ApplicationID,
etc)
' Handle the response
Response.Write "ResponseCode = " & ResponseCode & "<br>ResultMessage = " & Re-
sultMessage
%>
```

could get exchange rates in real time from different financial institutions and decide where to send the transaction based on the exchange rate of the option that represents the lowest cost for the transaction being processed. As exchange rates are dynamic and change in real time, a smart system that could compare multiple options could be advantageous from the cost point of view.

A taxation aware point of sales component that could be integrated with existing POS terminals could be also implemented with the proposed architecture, the terminal could decide among dif-

Algorithm 2. XML request code

```
- <EFSnet:Request xmlns:EFSnet=https://testefsnet.concordebiz.com/EFSnet.dll
   StoreID="StoreID"
   StoreKey="StoreKey"
   ApplicationID="My Test App Ver 1.1.0">
- <EFSnet:CreditCardAuthorize>
      <ReferenceNumber>12</ReferenceNumber>
      <TransactionAmount>100</TransactionAmount>
      <AccountNumber>4111111111111111</AccountNumber>
      <ExpirationMonth>12</ExpirationMonth>
      <ExpirationYear>05</ExpirationYear>
   </EFSnet:CreditCardAuthorize>
  </EFSnet:Request>
```

ferent financial institutions alternatives based on a taxation strategy that could take into consideration different taxation policies in different countries.

5. CONCLUSION

The prototype and architecture for the reengineering of existing legacy point of sale terminals for integration with financial institutions through web services has been successfully developed. The tests have demonstrated that payments can be made with the developed prototype with a good performance.

The architecture shows that web services will integrate legacy POS-terminal and payment providers across business and countries boundaries which make vendors much more flexible without having to throw away their legacy applications. Whit this freedom, vendors can negotiate about services and fees all over the world.

Standard permanent internet connections can be used by web services for payment transactions instead of legacy dedicated communication lines to financial institutes as in the conventional method. The used design methodology in the report is a good approach to design, develop, plan and maintain a POS-terminal web service system. The approach is iterative, generic usable and can be extended to organizations needs. The provided methods (OO, UP) and languages (UML, HTML, ASP, JavaScript) are worldwide standards which makes it relatively easily to build and change the system if the right skills are available

Future enhancements of the model would be the ability to negotiate in real time payment processing rates by incorporating the model proposed by Cohen et al. (2005) and by being able to extend the model to mobile point sales terminal in order to cope with modern trends towards M-commerce.

REFERENCES

Banksys, B. (2009). *Website available in Dutch/ French*. Retrieved from http://www.banksys.be

Burstein, F., & Gregor, S. (1999). The systems development or engineering approach to research in Information Systems: An action research perspective. *Proceedings of the 10th Australasian Conference on Information Systems,* (pp. 122-34).

Cohen, J., Lee, W., & Darlington, J. (2005). *Payment and negotiation for the next generation Grid.* UK e-Science All Hands Meeting 2005, Nottingham, U.K., 19th – 22nd September, 2005

Concord. (2009). *Product information.* EFSnet. Retrieved from http://www.concordefsnet.com/Developers/Documentation.asp

Dahlberg, T., Mallat, N., Ondrus, J., & Zmijewska, A. (2008). Past, present and future of mobile payments research: A literature review. *Electronic Commerce Research and Applications*, *7*(2), 165–181. doi:10.1016/j.elerap.2007.02.001

DNB. (2004). *De Nederlandsche Bank* (National Bank Netherlands). Retrieved from http://www.dnb.nl/dnb/homepage.jsp

DTI & Scottish Enterprise. UK. (2009). *Electronic payments*. Retrieved from http://www.electronic-payments.co.uk

EFSnet. (2004). Concord, EFSnet Web service, interface specifications, version 2.4. Retrieved from http://www.concordefsnet.com/Developers/EfsnetIfaceSpec.pdf

Goethals, F. (2004). *An overview of enterprise architecture framework deliverables, a study of existing literature on architectures*. SAP-leerstoel. Retrieved from http://www.econ.kuleuven.ac.be/leerstoel/sap/downloads/Goethals%20Overview%20existing%20frameworks.pdf

Humphrey, D., Willenson, M., Lindblom, T., & Bergendahl, G. (2003). What does it cost to make a payment? *Review of Network Economics*, *2*(2), 4. Retrieved from http://www.rnejournal.com/articles/humphrey_june03.pdf-doi:10.2202/1446-9022.1024

IEEE1471. (2000). *IEEE recommended practice for architectural description of software-intensive systems*. Software Engineering Standards Committee of the IEEE Computer Society, 21 September 2000. Retrieved from http://www.win.tue.nl/~mchaudro/sa2004/ieee1471.pdf

IFX. (2009). *The Interactive Financial eXchange forum*. Retrieved from http://www.ifxforum.org/

ITAA. (2009). *Web services and how the technology may be used to secure the homeland*. Information Technology Association of America White paper. Retrieved from http://www.qat.com/assets_whitepapers/itaa%20web%20services%20white%20paper.pdf

Knight, L., Steinbach, T., & Kellen, V. (2001). *System development methodologies for Web enabled e business: A customization paradigm*. Retrieved from http://www.kellen.net/SysDev.htm

Kong, X., & Liu, L. (2004). *A Web application architecture framework.* The Tenth Australian World Wide Web Conference, Gold Coast, Australia, 2004. Retrieved from http://services.eng.uts.edu.au/~xiaoying/research/web/paper_0407_Ausweb04_AWebApplicationArchitectureFramework.pdf

Kruchten, P. (1995). Architectural blueprints—The 4+1 view model of architecture. *IEEE Software*, *12*(6), 42–50. Retrieved from http://www.cs.ubc.ca/~gregor/teaching/papers/4+1view-architecture.pdfdoi:10.1109/52.469759

Lowe, D. B., & Eklund, J. (2002). Client needs and the design process in Web projects. *Journal of Web Engineering*, *1*(1). Retrieved from http://services.eng.uts.edu.au/~xiaoying/capstone/readings/DavidLowe_JWE_Paper_WebProcess.pdf

Munck, S., Stroeken, J., & Hawkins, R. (2001). TNO is a knowledge organisation for companies, government bodies and public organisations: E-commerce in the banking sector. *TNO Report*, July 2001, (p 38). Retrieved from http://www.oecd.org/dataoecd/49/3/2072939.pdf

Nunamaker, J., Chen, M., & Purdin, T. (1990). Systems development in Information Systems research. *Journal of Management Information Systems*, *7*(3), 89–106.

OFX. (2009). *Open financial exchange*. Retrieved from http://www.ofx.net/ofx

OMG. (2009). *Object Management Group*. Retrieved from http://www.omg.org/

Planetpayment.com. (n.d.). *Website*. Retrieved from http://www.wilsonweb.com/reviews/planetpayment.htm

SWIFT. (2009). *Society for Worldwide Interbank Financial Telecommunication.* Retrieved from http://www.swift.com

TOGAF. (2009). *Other architectures and architectural frameworks*. Retrieved from http://www.opengroup.org/architecture/togaf7-doc/arch/p4/others/others.htm#CORBA

WAAF, & Kong, X. (2004). *Web application architecture framework*. Retrieved from http://services.eng.uts.edu.au/~xiaoying/research/web/waaf/main_waaf.shtml

Worldpay.com. (n.d.). *World Pay*. Retrieved from http://www.rbsworldpay.com/

Zachman, J. A. (1987). A framework for information systems architecture. *IBM Systems Journal*, *26*(3). Retrieved from http://www.research.ibm.com/journal/sj/263/ibmsj2603E.pddoi:10.1147/sj.263.0276

KEY TERMS AND DEFINITIONS

Electronic Data Interchange: Is the electronic communication of business transactions, such as orders, confirmations and invoices, between organizations. Third parties provide EDI services that enable organizations with different equipment to connect. Although interactive access may be a part of it, EDI implies direct computer-to-computer transactions into vendors' databases and ordering systems.

Extensible Markup Language: Is a metalanguage that allows users to define their own customized markup languages, esp. in order to display documents on the World Wide Web.

Legacy System: Is an old, technology, computer system, or application program that continues to be used, typically because it still functions for the users' needs, even though newer technology or more efficient methods of performing a task are now available.

Reengineering: Is the examination and alteration of a system to reconstitute it in a new form.

Payment System: Is a system for the transfer of money.

Point of Sale (POS): Is the location where a transaction occurs. A "checkout" refers to a POS terminal or more generally to the hardware and software used for checkouts, the equivalent of an electronic cash register.

Web Services: Are typically application programming interfaces (API) or Web APIs that are accessed via Hypertext Transfer Protocol (HTTP) and executed on a remote system hosting the requested services.