Electronic Invoice and Payments

David Paisana

Computer Science Engineering, Instituto Superior Técnico, Lisboa 2008
David Paisana, david.paisana@ist.utl.pt

Abstract Aiming to reduce costs, companies are trying to find ways of automate business process like invoice processing. This document gives an image of the state of the UE and Portuguese regulations, along with many different standards that can be used to implement an electronic invoice solution. This thesis was conducted in Link Consulting and is a proposal for an electronic invoice solution using a service-oriented architecture, with the goal of being flexible enough to adapt to different scenarios. Another objective was to make software that could be extensible in the future and with small development could support more formats and protocols of communication. This project solves two different scenarios. The first one is when a company buys a system to use together with an ERP. The second one is when a company or a person hires a invoice service that can be used or not with an ERP. In this last case the software is installed in a company that offers the service remotely to its subscribers. This way solves the problem of individuals and companies that do not want to buy a system like this. This solution uses Microsoft technologies like BizTalk and takes advantage of standards like AS2 and UBL 2.0. All the work was made considering the Portuguese law and the recommendation made by the Portuguese Agency for the Technological Program (UMIC).

Keywords: Electronic Invoice, EDI, UBL, BizTalk, AS2

1 Introduction

An invoice is the proof that a payment and a receiving took place. Besides that, this official document is mandatory to audit the payment of taxes.

In the last years, there has been an effort to modernize this process, making it electronic and automating some of its actions, renaming it electronic invoice.

The “Nordic eInvoice consortium” describes it as a modern, reliable and economic method proper to big and small companies, and also to private consumers. It’s specially efficient when there are many receivers. It can be showed graphically in a computer in the same way as the traditional one, along with facilitate the process of archiving, distributing and approval.

The companies can receive the electronic invoice directly through its financial information system while private consumers can receive invoices using systems hired to specialized companies. (1)

In 2001 the European Union adopted the Directive 2001/115/EC that harmonize, modernize and simplifies invoicing in matter of VAT, along with other aspects related with transmission and conservation of the invoices in communitarian territory using electronic means. The member states compromised to transpose this Directive until the 1st of January of 2004, this means that after this date all member states recognize electronic invoices as valid. In Portugal this happened in 2003 through the law Decreto-Lei 256/2003. (2)
The Directive defines that:

The invoices or equivalent documents can be issued by electronic means, if the receiver accepts it and if the authenticity of the origin and integrity of the contents be guaranteed using advanced electronic signature or electronic data interchange (EDI). (3)

The rules about the issue of the invoice are the same as the ones applied to the paper invoices. Although the law allows auto invoicing and the outsourcing of the invoicing when electronic invoice is used.

As it happens with the paper invoices, it is necessary to store the copies of the electronic invoices sent and received. In this case it’s only allowed to store over electronic means the electronic invoices, and only if it can be guaranteed complete and online access to the data by the authorities. It is also necessary guarantee the authenticity of the origin and integrity of the contents for a stipulated period. Each member state determines this period, which in Portugal is 10 years. The way the authorities access the invoices is not described.

With the objective of stimulate the electronic invoice in Portugal and adopt a standard to the Portuguese case, UMIC (Portuguese Agency for the Technological Plan) has organized meetings with electronic invoice providers, institutes and governmental offices.

In April 2007, UMIC started recommending the adoption of the following standards (4):

Sintax: UBL 2.0
Document Signature: S/MIME
Transmission Protocol: AS 2
Coding of the Data: Base64

2 Defining the Problem

The technological evolutions forced the software developing methodologies suffer constant changes. Nowadays the industry is following the Service Oriented Architecture (SOA) paradigm and adopting open standards that can be supported by any manufacture, making easier the extensibility and interoperability.

Thus the objective of this work is to build a solution that follows the SOA paradigm, and at the same time use open standards that are recommend by the authorities, like AS2 and UBL. Due to the fact the types of the files change during the years, it is important to add support to new file types and protocols. This way the system can interact with other electronic invoice systems and make the connection with systems that doesn’t support the adopted standards.

In what concerns security it is necessary to consider some restrictions as it happens when you send important files over an open network.

Privacy
Authentication
Non repudiation by the issuer and receiver
Integrity of the invoice
Availability of the system
Integrity of the sequence of invoice numbers

The system needs also to support different working scenarios.

The first operation scenario is when a company hires a service to add functionalities to an information system that already exists (e.g. ERP). This system aims to be the connection between the ERP of a company and other electronic invoice brokers (providers), fulfilling all the needed requirements and when necessary converting files to different types and protocols making the integration fast and easy.

This scenario can be split into two sub scenarios that differentiate itself by the fact that one owns the system and the other one do not.

- The client possesses the electronic invoice system, has the responsibility of managing the system, and maintains the invoices in the system during the obligatory period.
- The client hires an electronic invoice service, this means that the system sends and receives the invoices from its ERP; although the electronic invoice system belongs and is managed by a broker.

The second operation model that the system supports is addressed to the SME’s that don’t have a managing system that supports invoicing, but still wants to use electronic invoice. In this case the objective is not to integrate with an system, but offer an interface that offers the possibility of sending and receiving invoices.

3 Proposal

Functional Architecture

The systems consists of the following modules

![Diagram of the functional architecture of the system](image)
Responsible for keeping the invoices and guarantee the security restrictions like:

- Data integrity
- Holding period of 10 years
- Integrity of the invoice number
- Contributes for non repudiation, because it keeps the digital signature
- Guarantee the confidentiality of the archive data
- Guarantee that only access information who has authorization
- Log of unwanted changes in the archive

This module tries to be the most abstract as possible in order to offer generic operations like `get` and `set` and not be stuck to only one file type.

**Audit and Security**

Responsible for all security functions of the system, accomplishing the defined requirements for the Portuguese case. The supported features include:

- Sign invoices using S/MIME
- Functions for validating the signatures of the invoices
- Generate symmetric keys
- Functions to encrypt and decrypt using symmetric keys
- Functions to encrypt and decrypt using asymmetric keys

The symmetric keys are encrypted and stored in the data base.

**Tax information**

Allows subscribers to visualize information about the sent and received invoices (e.g. invoices total, VAT total, etc). This module should take in account that each subscriber should only see its own invoices and that the tax authorities should have access to all.

**Entities Managing**

This module needs to keep information about subscribers, its clients and its suppliers including the authorization that each entity have on the system and the associated certificates. Should also keep the information about each user and manage the changes on the details of each entity. In cooperation with audit module should manage the certificates of the system.

**Invoice Issue**

It is the way in for the invoices that are going to be issued, supporting different syntax and protocols. The received documents don’t have any legal value since they are not signed, however this module needs to fulfill security restrictions between the information system of the subscriber and this electronic invoice system such as integrity of the information, authentication of the sender and privacy of the channel. This module should also assure the conversion of the files in case the document is sent in a different format.

**Invoice Reception**
It’s the way in to the received invoices. It should support the standards recommended by the authorities.

**Processing**

It’s where is made the forwarding of the invoices after it’s received by the invoice issue module or by the Invoice Reception module. Takes care of the invoice workflow, calling all the activities that are part of the business process. From here the document can continue or be rejected, depending on the established criteria (e.g. if the invoice number already existo on the system)

**Delivery**

It is the final module of the business process, where the invoice is delivered to its receiver. The same way as it happens in invoices issue module, this module supports different protocols and formats.

**WEB Interface**

It is one of the ways entities interact with the system. To the subscribers that do not have an information system connected to this, they will use this interface to send and receive invoices. On the other hand due to the fact that tax authorities haven’t defined the access to the invoices, this system offers a very friendly interface able to generate reports and show the details of the invoices.

**Implementation Architecture**

In what concerns implementation it was used a three-layer model (presentation, business and data)

The presentation layer was implemented using the .NET 2.0 framework.

In the business layer has a business workflow module composed by four orchestrations that match the three business process of the system. This was implemented using Microsoft Biztalk 2006 R2:

- Invoice Issue
- Invoice Reception
- Invoice Delivery

An application can expose some functionalities as a service that other application can use. The aim of this is to hide the implementation details and expose only an interface of the business. As it is usual, this interface is implemented using XML Web Services.

This work will dispose three services:

- Invoice management that handles all the issues related with the archiving and treatment of the invoice.
- Entity management that gives information about the entities registered on the system, along with it access rights.
- Security management that provides methods to encrypt, decrypt, sign and verify signatures.
Finally the data layer is composed by a relational database with eight tables. The table Entidades keeps the data of each entity that interacts with the system, independently of being or not a subscriber.

The table ContratosSubscricao contains the registry of the contracts that each entity made on the system, including the actual state in the system. This table has also the certificate of entity, and every time an entity changes its certificate a contract is closed and another one is created, putting the actual date in the End Date field of the contract that ended and in the Begin Date of the contract that is starting.

To store the invoices two tables are used: Facturas and FacturasFicheiro, the first one keeps information that helps classifying and finding, while the second one keeps the encrypted invoice along with the corresponding symmetric key also encrypted.

Two fields stand out, which is ParaApagar, a bit that is activated when the system verifies that an invoice is out of date. When an invoice is out of date it could be deleted, but it was decided to just mark it so the system doesn’t be so vulnerable to attacks (e.g. for example the time of the machine is changed and every invoice become out of date). The other filed is Indiponivel, a bit that is activated when an invoice is changed in the database without permission.

When a change is detected the system logs the change in the table RegistoLogs, registering the previous state and the next state, so the administrator of the system can restore this.

Although the sell-by date of the invoice in Portugal is 10 years, the tax authorities may need to extend this period, so this can be registered in the table Bloqueios, where is registered which was the authority that blocked the invoice and the date it happened.

**Used Tools**

The system lays on Microsoft Windows XP SP2 operating system and uses Microsoft SQL Server 2005 for the persisting store of data.

In order to make the object relation mapper, it was used codesmith with net tiers.

As a development environment it was used Microsoft Visual Studio 2005, with C# language, because it offered good integration with Microsoft Biztalk 2006 R2 that was used for the Business Workflow.

To create the files respecting the UBL rules it was used a library made by a company called ebComposer because was the only one that offered such a tool.

**Application Security**

The application was conceived with a lot of mechanisms that assures all the security restrictions demanded by a system like this. On the internal level i.e. in the archive, the invoices are encrypted to guarantee that it’s only possible of access them thru the application. This way, if the database engine is hacked still is not possible to get the invoices automatically.

After the system receives an invoice verifies its authenticity and integrity using the S/MIME signature attached.
Each subscriber of the system can upload its certificate, so the system uses it to sign the invoices issued by
him in the system and verify the ones received. Every time a subscriber change its certificate, a new virtual
contract is generated, so in the future the system knows, which certificate was used in a given date. This way
when an invoice is signed, the most recent certificate is used, doing the same procedure when receiving an
invoice.

When the system needs to open an old invoice, uses the corresponding certificate. In case there is not a
certificate registered, it system certificate is used. Each certificate is encrypted in the database with a general
asymmetric key of the system.

It was evaluated the need to use a timestamp certificate authority, with the purpose of being less vulnerable to
attacks. If for examples someone manages to change the system date, can turn all the certificates out of date,
stopping the system. If an authority like this is used, the probability of making such an attack is reduced.
Sometimes the certification authority that issued a certificate can disappear, making impossible to assure the
authenticity of a certain certificate. In case a timestamp authority is used, it is possible to be sure, that at the time
the certificate was used, it was authentic and valid.

In a commercial application, the certificate chain should be validated until the root, to guarantee an adequate
level of security. Because this project is academic, such a mechanism is not going to be implemented.

After the integrity and authentication of a received invoice has been confirmed, or after an invoice has been
generated by the system, is encrypted on the archive. The first step of the process is to generate a symmetric key
that encrypts the invoice, i.e. each invoice is encrypted using its own key. After that, the symmetric key is
encrypted using the asymmetric key of the system. This asymmetric key should be kept outside the system, e.g.
smartcard. However in this academic case the key will be in the database.

With the purpose of not storing unnecessary invoices, a schedule trigger verifies which invoices are out of
date and set them to delete. Later a system administrator will be able to confirm the deletion of the invoices using
the back office. The option of not deleting the invoice right after running the trigger happen to avoid that some
invoice may be deleted by mistake if for example the time and date of the system are changed incorrectly.

To lessen the fact of a person gain irregular access to the database and delete some data, it was created a
trigger that logs every change on the database that wasn’t made by the application. Because it is kept the previous
and actual state of the changed column is possible to the administrator to rollback the changes.

Although the access of the tax authorities to the system is not described and the majority of the brokers
present at the UMIC meetings just provide a sql agent, this system offers web interface similar to the ones used by
the subscribers of the system. This access has to be requested by the authority and allows making certain
operations like blocking an invoice so it is not deleted, even if becomes out of date.

To reduce the vulnerability of the system, the connection string to access the database in encrypted in the
configuration file of the application.

To guarantee the uniqueness of the invoice number, the system makes it impossible to register an invoice
number that already exists.

To assure the security of the web portal, out of the box mechanisms supplied by .NET framework are used. In
this case it was chosen Forms authentication, because it does not restrain the universe of user to the ones that use
Windows family OS, like in Windows Authentication. Microsoft Passport could be another option, although it’s difficult to get some information like tax identification number (Número de Identificação Fiscal).

To avoid the vulnerabilities that this option is submitted, like sending the username and password thru the internet, it was used SSL to encrypt the communication. .NET out of date cookie policy assures that a hacker does not access the system using stolen cookies.

**Invoice Issue**

In the issue invoices module it is necessary to guarantee specially two things, the authentication of the issuer and the privacy of the channel. This module supports many protocols and file types (Web Service, directory and back office portal).

In case the subscriber information system is configured to issue invoices thru a directory, this will save a file in a pre-defined directory, using the permission mechanism of the operating system to assure the necessary authorizations.

When using Web Services it is used a secure SSL channel to guarantee the privacy of the channel, the authentication is made by using a token in blank field of the XML invoice file. This token is configured in the back office private area and can be 1024 bits, 2048 bits or 4096 bits long.

Finally, in case the invoices are issued thru the back office application, it is used Forms Authentication mechanism provided by the .NET, where the user is challenged to insert a username and password. The back office application creates an invoice xml file with the corresponding token and calls a web service.

Like it happens in the invoice delivery that supports many file types, also here it’s possible to receive by many means the invoices and easily add more in the future.

**Invoice Reception**

As it was stated before, this project uses UMIC recommendation (UBL 2.0 for syntax, S/MIME for signature and AS2 for the communication protocol). Because AS2 can send signed and encrypted messages, it’s necessary to exchange the certificates between the business partners before start sending the invoices.

Because Biztalk makes the enveloping of the invoice in AS2, the certificates are stored there using the concept of party that Biztalk offers. To verify the signature of the invoice is necessary to use the certificate that goes together with the invoice and verify the certification chain to the root to make sure the certificate is valid.

**Invoice Delivery**

When the subscribers don’t have an information system where the received invoice can be delivered, they can see the invoice details on the portal. The delivery using directory is supported thru this three file types:

- GS1 XML without signature
- UBL 2.0 without signature
- UBL 2.0 with signature

In case the invoice is delivered in another broker it’s used AS2 (over http) in UBL 2.0 with signature. The security mechanism of the AS2 guarantees the safety of the process.
4 Results

The main objective of building a solution that used the industry standards, specially AS2 and UBL was accomplished.

The implementation of web services to issue invoices let that any other system that supports this technology can send invoices, and it is not necessary to take much time in a complex integration. Due to the use of web services is possible to issue invoices from remote systems.

Another protocol that the system support’s is directory. This protocol can be used either to issue invoices or to send received invoices. To send invoices using directory one of three types of files can be used (GS1 XML without signature, UBL 2.0 with signature and UBL 2.0 without signature). This supported file types and protocols are the proof that’s easy to extend the system using Biztalk mappings and dynamic ports.

The web portal offers many options to manage clients and invoices, allowing a company to be a broker that sells electronic invoice as a service, letting each subscriber to configure its clients and subscribers. The clients and suppliers use the system to see its invoices find a user-friendly environment.

The archive shares the same objective of the project, being as flexible as possible, supporting a generic type of invoice. The security issue was not left apart in this component, with many redundancy mechanisms being implemented to detect intrusions.

In this project was achieved a flexible, robust, loosely coupled, service oriented and extensible solution that allows easy integration with different platforms in different locations.

The use of Biztalk allows the reutilization of the system and fast integration with other information systems, due to the fact that it can be easily added new protocols and types of files. Because it promotes the development of a service oriented architecture (SOA) and Business Process Management (BPM) scenarios allow the developer to focus more on what really matters, the business process and less on the implementation. On the other hand Biztalk revealed an obstacle due to the difficulty of learning and some handicap of the tool.

On the financial side, this solution is not accessible to small companies due to its price. If we only consider the price of a license is 9000 dollars. To this value, we need to add the price of developing, and installation. To this kind of companies, the best solution will be to hire this service instead of buying it. (5)
1. Bibliography


