Electronic Invoice Broker

Bruno Ricardo Vala Sequeira
Create It, Lisboa. Portugal, brvs@mega.ist.utl.pt

Abstract. The electronic mail allowed a profound transformation on the traditional commerce techniques, where information and communication technologies are important for the new ways of doing business. A currently expanding area, is the electronic billing, that emerges from the e-commerce, as an essential condition for its development. Nowadays, it is already possible to ensure the trustworthiness and integrity of electronic documents and therefore it makes no sense to demand that invoices, as billing documents, are turned into paper and then filed. The use of electronic invoice brings several benefits for the agents of a trade, such as, reduction of costs and the simplification of the financial processes thus optimizing productivity. This work will approach the problems resulting from the exchange of electronic invoices and the legal requirements that support its implementation and will also present an integrated solution that allows the transaction of these documents between different organisational information systems. Amongst the main challenges we find the absence of international standards for the documents' format making its integration problematic. The importance of invoice validation as an essential document for the success of a transaction makes it necessary to ensure its authenticity, integrity and confidentiality. For the implementation of the solution the "Message Broker" model was adopted to allow an easier exchange of electronic documents and solve operational incompatibilities.

Keyword: Electronic Invoice, Interoperability, Integration, Business Process

1 Introduction

The efficiency and the productivity of the administrative and financial areas of a company depend, mainly, on the time and resources spent in its tasks. Betting in solutions that make these task automatic and digital, is therefore, a growing necessity in organizations that want to minimize time and cost of these processes and bring added value to their business.

Part of the financial expenditure of organizations is related with the administrative processes that deal with invoice processing. The bigger the companies get, the more serious this problem tends to be, once they have a large number of invoices and thus a huge amount of costs spent in paper, envelopes, mail taxes, processing and time in these tasks, goes sky high, as well as its later income control. The current invoice system makes printing the invoice necessary, as well as mailing it in order to conclude a transaction.

The use of electronic invoice has, therefore, a very high importance, in the way that it allows companies to turn its invoice processes automatic and increase its productivity and internal efficiency.

This work will try to present the current technologies and models in the scope of interoperability and company's internal efficiency, and to make an approach on the standards and platforms that allow the development of an electronic invoice solution. It consists on the development of an application for invoice mediation and of a portal which allows the configuration and management of the business elements that intervene in the documents' exchange.

The invoice broker has the following operational goals:

- Create conditions for interoperability that allow its usage by different platforms;
- Follow the rules related to electronic invoice;
- Define a business model to grant a service that allows the users to exchange invoices as business partners;
- Define the work flow for the emission, reception and storage of electronic invoices;
- Define the canonic scheme for the invoice, in order to make the data integration with the users' applications easier.

The development of the Broker application is based on BizTalk 2006 R2 integration technology from Microsoft, that offers transactional support to the sending and receiving of messages. The Portal allows the creating of agreements among clients who wish to use the system, as well as visualization of the messages that run through the broker.

2 State-of-the-Art

2.1. E-Commerce integration

The business-to-business (B2B) integration can be defined as the set of business activities where an electronic exchange of message between one company and its business partners occurs [1]. It is then possible to characterize it as the connection between certain business processes of a company. In other words, between the processes with higher external importance for the company, and the business processes of its partners.

B2B integration has had a significant impact in information technologies industry. One integration solution may offer, not only, advanced middleware between companies and businesses, allowing a starting point for companies to develop interfaces faster, as well as to allow the automation and management of the business processes. Basically, an integrated business solution combines technologies and the processes that allow the optimized business application construction, for the exchange of message at business level, in formats and contexts that each person will understand. Or rather, it allows the offering of a uniform insight of the electronic business system for the entities involved in the interaction (both selling and buying parts). [2]

2.1.1. System integration challeges

Several problems must be taken into consideration, during the integrated system solution implementation, in the cooperative relation between partners.

On B2B relationships, data exchange between different systems occur, creating incompatibilities between data input from one system and the output of another, its formats and even its transport mechanisms. In other words, the companies share heterogeneous information resultant from the activities involved in their inter and intra-company processes.

The interoperability between heterogeneous and shared systems has to be made not only at a technical level but also at the information level itself. From this point of view, a B2B relationship implies the communication between heterogeneous information sources that belong to different companies with its own cultures and conventions. To deal with this heterogeneity, it's necessary to have pre established agreements in which a semantic is defined to allow a formal sharing on a determined application domain, in a way that the meaning of the information exchanged is understood. [3]

2.1.2. Interaction between business partners

In an electronic information exchange system, the applications interactions occur in 3 layers: communication, content and business processes. In a real situation, for two companies to establish an interaction, it's necessary that an agreement on the union of their business processes is made; they need to understand the contents of the sent messages and have to agree on a communication protocol for the messages exchange.

Transport Layer: This layer focus on the exchange of messages between parties, remotely connected, supplying protocols that allow their interaction. The interoperability objective of this layer is to supply an independence of these protocols and platforms through a translation and conversion of messages between heterogeneous protocols, obtaining a transparent integration. [4]

Content Layer (Information): This layer solves heterogeneous, semantic and structure related problems, supplying languages and models to describe and organize the information in a way that it can be understood and used. The content interactions need that the involved systems understand the content semantics and the business document types**.** [4]

Business Process Layer (Workflows): This layer solves the semantic interaction between business processes. It is necessary to understand the semantic used by each economic partner's business processes. [4]

2.2 Electronic Invoice

The electronic invoice is a commercial document, similar to the conventional, but in electronic format, possessing the same fiscal/legal value as the paper version, as long as it contains the mandatory statements of all invoices and fulfil certain conditions demanded by law. These conditions go through maintaining the authenticity of the source and the integrability of its content. But when we talk about electronic invoices we aren't talking about sending invoices by email. It its a modern method, safe and efficient to process and manage assets, services and other expenses minimizing human intervention.

The de-materialization of the invoice brings several benefits for the parties involved in an electronic collaboration: Cost reduction associated with the processing, emission, distribution and conservation of the paper invoices; Optimization and simplification of the administrative and financial processes, bringing a reduction in time and space spent (since it's not necessary the invoice printing for legal/fiscal purposes), with clear benefits for the collaborators' productivity; It allows the integration with partners' systems, for example the accounting system, making it easier and increasing the data process; Improved service quality, waiting time and delay reduction; Fewer fails while sending invoices and abolishing the chances of the document getting stolen/lost; Management optimization and transaction control; Easy to read and store; Better security and transparency in the transactions.

2.2.1 Electronic Invoice Models

There are two alternatives to ensure the conditions demanded by law, and that correspond to two distinct electronic invoice models [9]:

- Electronic invoices exchange on an electronic data interchange, or EDI (Electronic Data Interchange);
- Electronic invoices exchange, ensuring the origin and integrity by the use of an advanced electronic signature.

The EDI is a way to transmit messages between two computer applications, in a commercial context, without human intervention, in which these messages obey to a pre-defined and structured grammar. With EDI, documents are exchanged through peer-to-peer connections, private networks, VANs and internet. Its goal is to allow that two commercial partners with different information systems and therefore with different internal representations of data can exchange business information in a normalized format which is compatible for both ends. In this mechanism, both parts determine the initial business conditions, techniques, legal and communication safety, through an initial data interchange contract. [5]

The digital signature is a type of advanced electronic signature based on a cryptographic asymmetric system made by an algorithm, from which it is generated a couple of exclusive asymmetric keys that are independent being one considered private and the other public and that allows the owner to use a private key to declare the ownership of the electronic invoice, and the receiver to use the public key to verify if the signature was created from the correspondent private key, and if the document was altered after the signature. [5]

2.2.2 Electronic Invoice Techniques

i. Electronic Invoice with Web Forms

This option involves one web application that allows the insertion of invoices and its posterior emission in an electronic form. This solution consists on a service, where a supplier fills up a web form that is sent electronically to the client in a specific format. Usually, the web site offers additional services such as storage and printing of invoice copies. The solution is better for the buyer whose manual work is diminished. [6]

ii. Electronic Invoice by mediation (intermediary)

This solution uses an intermediary that can be an ASP operator (Application Service Provider) and that allows the electronic invoice transaction, converting the data from the invoice between the supplier and the receiver.

This solution allows us to solve the problem of the existence of several message and protocol formats of communication for the exchange of documents. Many companies use this solution to exchange electronic documents with business partners, independently from the data standard that they came to choose. Instead of managing each peer-to-peer combination with the business partner, the documents that are to be sent and the standards to use, the transactional hub allows to interconnect the different data standards and communication mechanisms used by each business partner, translating the format used by a certain partner to the format used by another. [6]

iii. Auto-Invoicing

The process of auto-invoicing can be described as being the client creating the invoice, instead of the traditional way where the supplier is the one that generates the invoice for the product to be sold. The auto-invoicing is used in acquirements where the prices are agreed and altered in the buyer's systems, which makes its use difficult for all the invoices. [6]

iv. Invoice Digitalization

The invoice digitalization is nothing more then an application that makes the creation of electronic invoices possible, and instead of printing the invoice, its sent to a receiver by email (in format like Adobe PDF). This

solution is different from the others in a way that the client ends up printing the invoice himself. The potential benefits of this solution are the reduction of paper and envelop costs for the seller. There are some risks in this solution that come from the possibility of the documents being altered.[6]

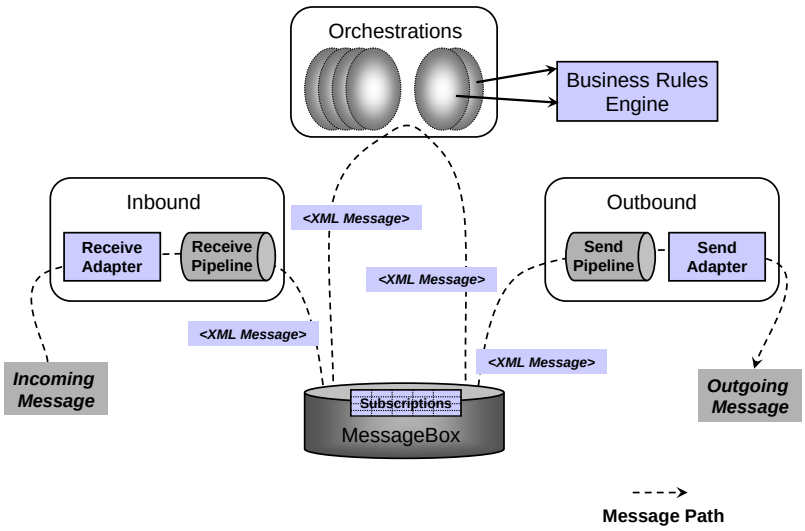
3 Technological Platform

The BizTalk 2006 server is a product from Microsoft, known for managing business processes. The objective of this product is to allow the automation and optimization of a company's business processes, using for that goal, several integration tools. The BizTalk server was designed specifically to integrate heterogeneous systems in a model called "loosely coupled" and allows a fast creation of integration solutions. The approach of these tools for B2B interactions is based in an investment in several standards and technologies that include SOAP (Simple Object Access Protocol), XML and MIME (Multipurpose Internet Mail Extensions). [7]

3.1 Main Components

The picture found bellow illustrates the main components of the BizTalk engine that allows the message trading mechanism to work and the logical processing of these messages. As it's found in the illustration, a message is received through a receive adapter. Different adaptors supply different communication mechanisms, so that a message can be acquired through a Web Service or the reading of a file. The message is then processed on the receive pipeline, that allows the conversion of this message from its native format to a XML document. The message is then delivered to a database, called MessageBox, implemented on the SQL Server.

The logic that executes the business process is implemented in one or more orchestrations. Each orchestration creates substitutions to indicate the types of messages that it will receive. When a message enters the MessageBox, this message is forwarded to the correct orchestration. The result produced by the orchestration is another message that is stored in the MessageBox. This message is then processed by the send pipeline that can convert to the correct format and then sent through the send adapter, that uses the right mechanism to communicate with the application to which the message is destined.[8]



Picture 1 - BizTalk architecture

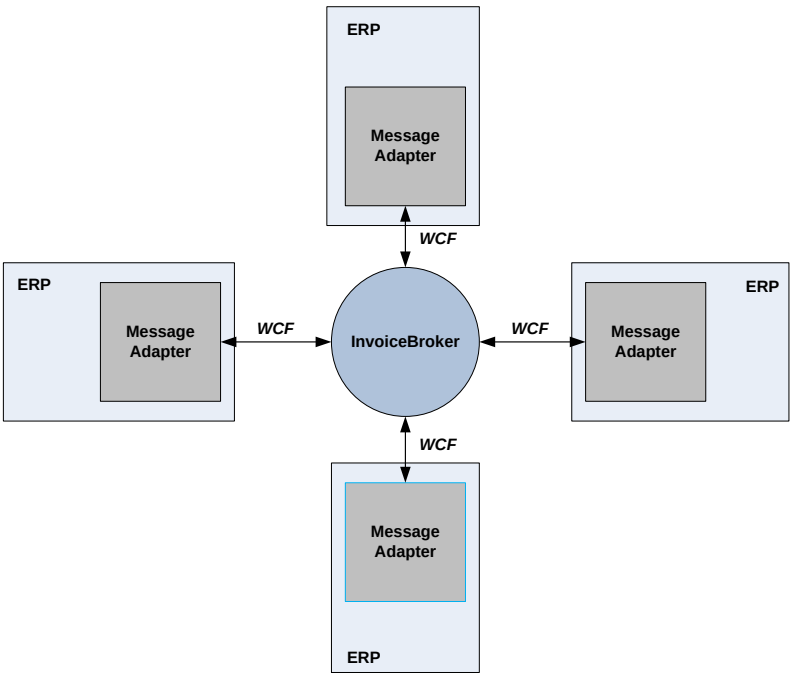
The following list shows the several components that make the BizTalk: Ports and Adapters, Business Activity Monitoring and Business Activity Services, Pipelines, Orchestrations, Transformations, Messaging Engine, Business Rule Engine.

4 Integration Solution

The electronic invoice solution discussed in this work, approaches a system that includes one adaptor, placed on the system users side, and one electronic invoice central mediator, where they are treated and forwarded correctly. The message adapter, from the users side, will allow the exchanged messages integration, with several ERP information systems from the users. The invoice broker is responsible for the secure communication, for the necessary business processes support and forwarding, for the correct receiver, for the exchanged messages between the users. The message adapter is an integrated component in each information system of each user, where the aspects of message integration on the system are treated such as the data conversion that allows the inclusion of these messages. Although the message adapter component is important for the approached business scenario in this work, it is only asked the study and development of the message broker component, for the electronic invoice mediation.

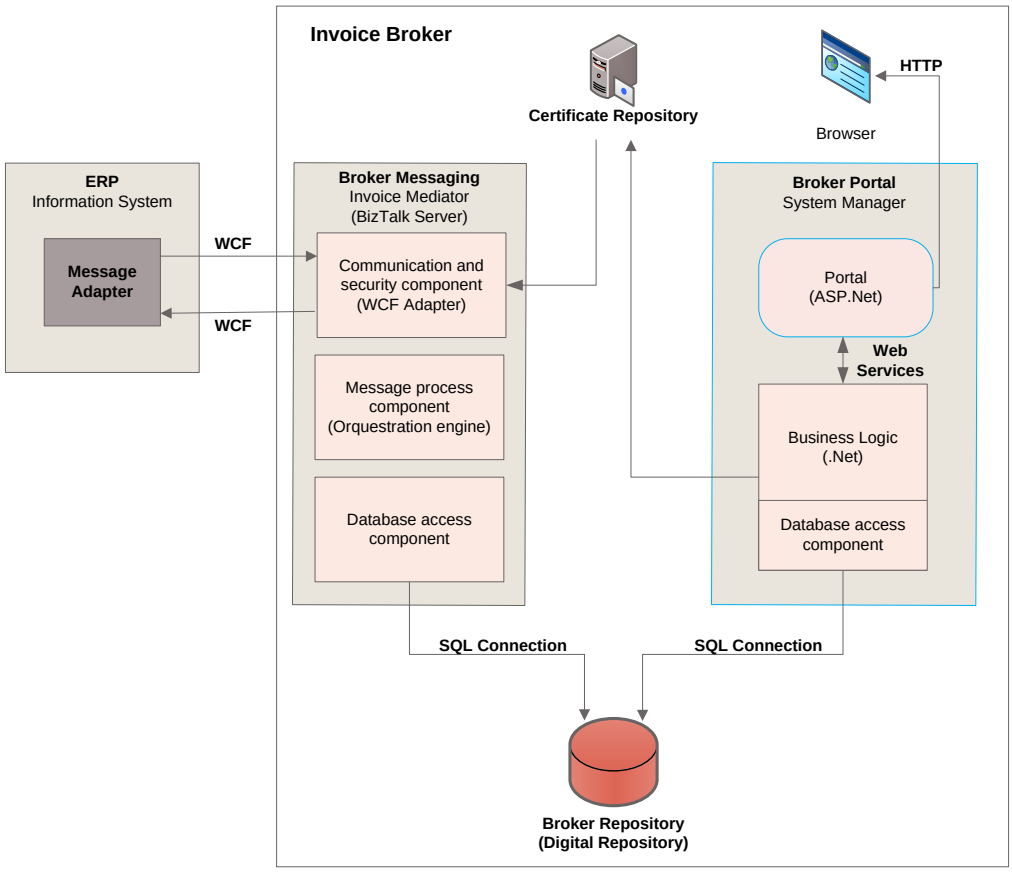
The electronic mediation process consists in emitting invoices by an ERP application for the broker, go through the validation process and forward the invoice for the correct ERP application. Messages that signal the success or error of the processes that work on the documents. The process starts with the arrival of a message on the BizTalk. This message can be an invoice or an acknowledgement, the invoice is sent by the message adapter from the ERP side. The message adapter generates the invoice and sends it to the broker in accordance with the specified scheme. When the message is received, its integrity is verified to guarantee it's complete or if a transition error occurred. If the conditions are not verified, a state message is stored in the monitorization system and the origin system is notified of the existence of an error. If the message is correctly received, the scheme is validated to verify the existence of errors in the message's data.

The picture below illustrates the complete system where the Invoice Broker is inserted, in a scenario of electronic invoice mediation.



Picture 2 - Invoice Broker use scene

The Invoice Broker mediation system is made of four essential components. The message processing module, the digital repository module, the communication and safety module and the system management module. In the picture above, it's illustrated what the invoice broker consists of and how its layers interconnect.



Picture 3 - Invoice Broker Components

The **communication and security component** is responsible for the communication with the adapter. For its configuration it was used the BizTalk's WCF adapter. It is in this component that the signature and cipher of the electronic components that are exchanged with the broker are guaranteed, being used for this purpose the WS-Security protocol.

The **invoice processing component** (BrokerMessaging), that is used for invoice processing and transmission to the proper location, the forwarding of the invoice notification after its properly processed and confirmed by the destiny information systems, the invoice validation and the credentials of the origin. It consists in the business process orchestration that involves the invoice data validation and the intelligent routing to the correct destination. This component uses orchestration of the BizTalk server to process its messages that are received by the broker and the formatting of the messages sent by the broker.

The **digital repository** (BrokerRepository) is responsible for the persistent storing of all system entities and exchanged documents. To interact with the repository the data access layer is used and it allows the storing, editing and selecting operations over the system's information.

The **system management component** (BrokerPortal) is responsible for the administration and management of the system entities that will ensure a good invoice broker system performance. It is used to perform maintenance over the subscribers data sheets and existing agreements among subscribers and also for the exchanged documents search and invoice reporting. In this component we can find several administration services and system user services, allowing the data access that are important for this layer. It is also composed by a portal, where it is implemented the information layout and business logic, and where we can find the available services logic implemented.

4.1 Communication and Security Component

The Invoice Broker uses the WCF protocol for sending and receiving messages to the message adapter, situated in the user endpoint. For the Invoice Broker communication service configuration, it was used BizTalk's WCF adapter.

To configure the WCF adapter to allow receiving clients messages, it was used WSHttp Binding that offers support to the WS-* standards over the Http transport protocol. This adapter allows the access to the SOAP

security, reliability and transaction properties. For the client (message adapter) to access the service, the use of the “ScvUtil” tool was necessary, to allow the creation of the service's metadata files to be used by the client. This tool is executed with the service address and allows us to obtain the “proxy” file that specifies the message type that the service awaits, and an application configuration file that allows the client WCF configuration.

The security used by the WCF service is made at message level. The WSHttpBinding uses the WS-* specifications and from WS-Security to encrypt the message and authenticate the users. This standard (WS-Security) allows the message's SOAP Header definition that gives the security parameters used in the message.

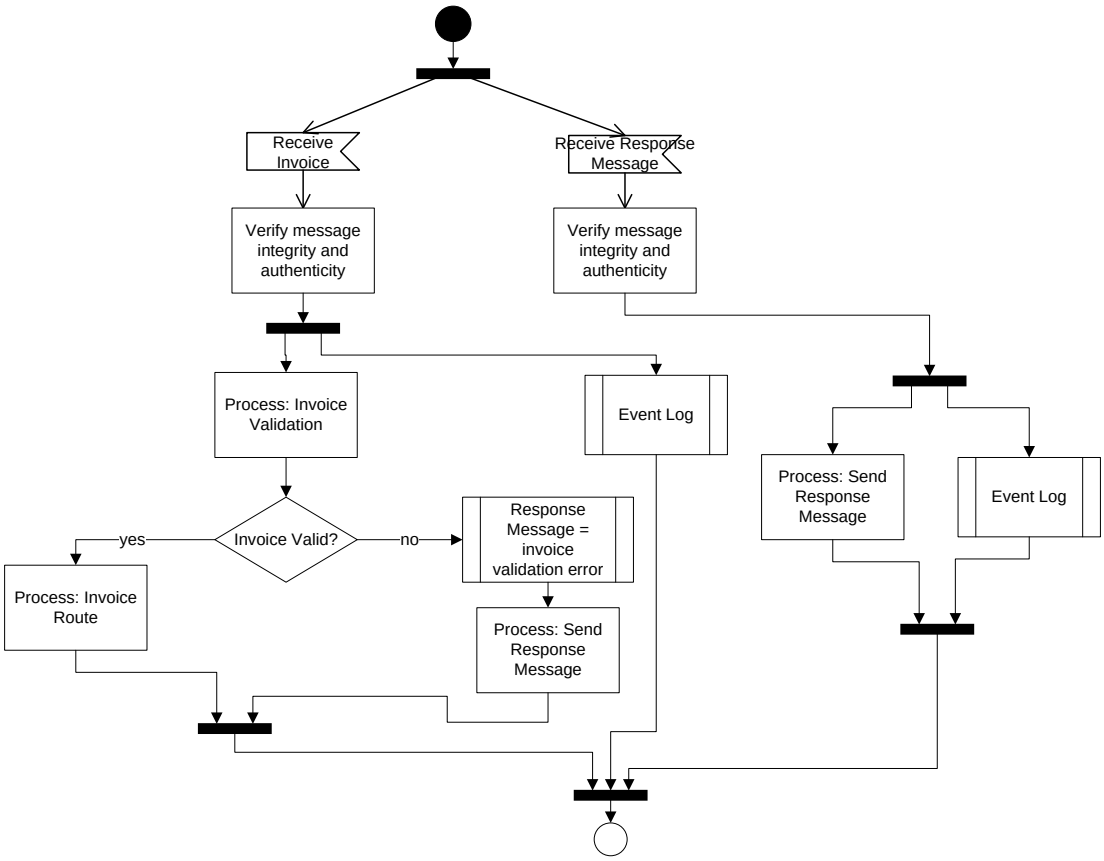
To configure the WCF service that allows the client authentication using the correspondent certificates, it was necessary to follow the following steps:

- The MessageClientCredentialType property is set to Certificate, to allow the use of digital certificates.
- The NegotiateServiceCredential specifies how the client application sends the certificate to the WCF service. If this is set to true, the WCF service waits for the client application to include the certificate with the messages it sends (a series of initial messages occurs while the client and the WCF service exchange certificates). These properties were set to false, to make the client certificate installation to be made manually on the Trusted People certificate Store.
- The CertificateValidationMode property allows the specification on how the WCF service verifies the client certificate confidentiality. This property was set to to PeerOrChainTrust, to allow the service to search on the Trusted People Store for the client certificate; otherwise the client's request is rejected. The Chain Trust service verifies if the CA that emitted the certificate is valid and trustworthy.
- The RevocationMode property specifies if the service verifies if the client's certificate was revocated. The service does not verify this condition.
- The service certificate is obtained given a Root address. The service searches on the Local Machine repositories in order to validate certificates.

4.2 Message Processing Layer

This component was implemented with BizTalk orchestrations and finds it self divided in three parts. Each part is executed sequentially and corresponds to the invoice treatment process that arrive at the invoice mediator.

To start the flow, the application finds its self listening to a logical port, that receives authorized messages from the WCF adapter. The receive WCF adapter validates the message's digital signature and only on success it gets published at the BizTalk’s MessageBox. After the message is published the business orchestration starts, that contains the rules to treat the received message. The next figure illustrates the high level process corresponding to the invoice treatment.



Picture 4 – High level business process

First the information about the running transaction is stored, being indicated the state in which its found. Then the current step is registered at the system's log, and then the message's content validation process starts. This process runs through the validation of the sender, verifying if he exists on the system, then it gets validated if the receiver and sender have and agreement that allows the forwarding of the invoice. It is also verified the invoice's chronology, if the number of the invoice is correct taking into consideration the prior invoices and if the date is also correct (if it is sent with a temporal order superior to its prior invoice). If the invoice number and dispatch data have the correct sequence, the invoice document is stored into the database and the current transaction information gets updated, the current states gets now a “processed” status and it is indicate to which agreement this transaction belongs to. If the invoice number and dispatch data are the same to prior invoice, then its a duplicate document, being only changed the information of the running transaction. If the chronological sequence is invalid, an exception is generated about the invoice treatment. All steps taken during the document's validation process are stored in the system's log.

Depending on the invoice validation state it is created a response message to the sender, stating the success of the invoice treatment or obtained errors, next it is sent to invoice owner.

If an error occurs during the invoice validation the work flow is terminated. Other wise it is stated the start of the process that will forward the invoice to the correct destination. First the existence of the address is verified and then the invoice gets sent to the logical port that will forward the message to the WCF adapter. If the sending of the invoice gets an error, two more tries are made. If a sending error occurs, it is generated an exception and a response message is created with the given error for future delivery to the sender. All steps taken during the invoice forwarding process are registered in the system's log.

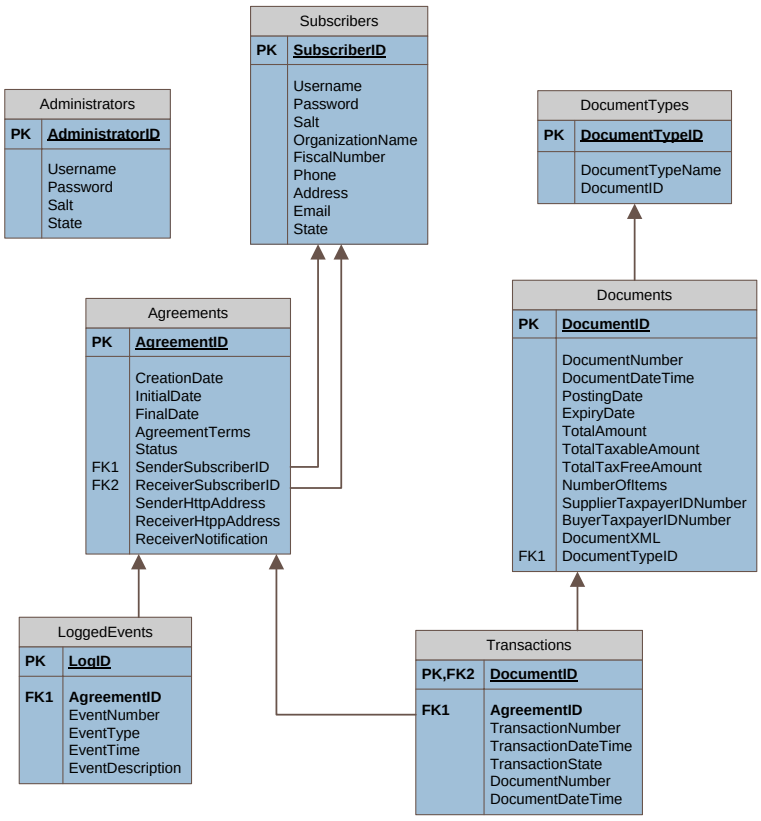
If the invoice is sent successfully, the invoice mediation process ends.

4.3 Digital Repository

The digital repository is responsible for the persistent storage of all system's entities and exchanged documents. To interact with the repository it is used a data access component that allow to operations like saving, editing and selecting system information.

4.3.1 Domain Model

In this section it is presented the domain model that gathers the necessary entities for the correct work of the Broker System. The Administrators table is used to store the information referring to the system's administrator. The Subscribers table is used to store information related to the invoice mediator User. The Transactions table gathers the information about the occurred transactions on the invoice mediator. The Agreements table is used to store agreements made between subscribers. The LoggedEvents is used to store the system's state information. The DocumentTypes table gathers the invoice types that can be exchanged by the system. The Documents table allows the storage of the exchanged invoice documents.



Picture 5 – Domain Model

The figure above describes the system's entities table that are stored in the database. As you can see a subscriber can contain several agreements, being each one composed by two subscribers. The Documents table relates itself with the DocumentTypes table, being the last one only for supported invoice type's information. Each Document has only one DocumentType. The Transaction entity allows us to relate the Document entity with the Agreement, indicating that one transaction occurred for a certain agreement between two subscribers, being the document the content of the exchanged message. The Transactions and LoggedEvents table, although they contain an element that indicates which Agreement they belong to, it was chosen for system performance issues to lack the physical reference to the Agreements table.

This was made because when a message arrives at the invoice mediator module, a Transaction and a LoggedEvent entities are created that inform the occurred step, but it is not possible to determine the agreement to which they belong before the message validation process.

4.3.2 Database Access component

This component is responsible for abstracting the system's database layer. It will allow the system's object to be converted to the relational model, using the ADO.NET technology, and for them to be stored in a persistent way on the system's relational database.

This component gathers a set of services that are used by the user in the Portal component and in the invoice mediator component. It will allow the business entities, configured in the portal, to be stored and accessed just like the invoices that are exchanged between users through the invoice broker.

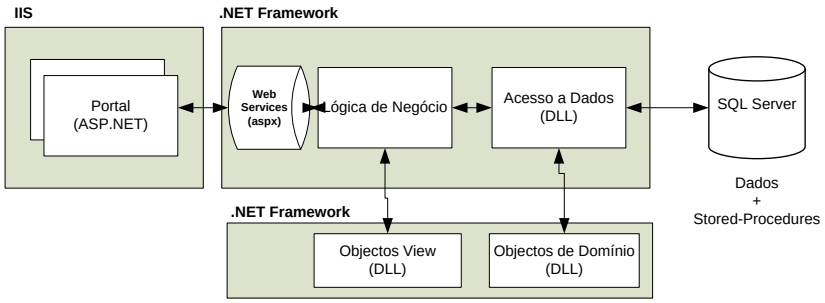
4.4 Portal Component

The BrokerPortal module is a Web application that allows the administration and management of the system through an internet connected browser. The architecture was defined over three layers, like shown in figure 2, for this module.

The first layer, designated “Portal”, consists on a set of executed controls supported on the IIS server. This layer is responsible for the man-machine interface and allows the interaction of the administrator or subscriber user, with several available services. This component was implemented with ASP.NET technology.

The second layer, designated “Business Logic”, is responsible for the implementation of all the required services for the correct functioning of the business. This component publishes the access to the services using Web Services (.asmx), and these are implemented in .NET platform.

The Data Access layer, consists of a DLL library that gathers the component set responsible for the database accesses. Besides maintaining the data in a persistent way, the database manages a set of routines (stored procedures) responsible for the operations over stored data.



Picture 6 – Portal architecture

4.4.1 Services Architecture

The Portal implements a set of services that will allow the configuration and administration of the system's Invoice Broker. Following is described the functionalities of the available services:

- **Authentication**: This service allows the Administrator and Subscriber entity to authenticate them on the portal, so they can have access to the functions made for them in the system.
- **Create Subscriber**: This service allows the creation of a “Subscriber” user in the database. It is created also a certificate for this user. For that it is called the MakeCert tool and a digital certificate is created on the Windows Certificate Repository.
- **Remove Subscriber**: This service allows the deletion of a “Subscriber”, previously created in the system.
- **Edit Subscriber**: This service allows the altering of data of a “Subscriber”, previously created in the system.
- **Display Subscribers List**: This service allows returns all the “Subscribers” that exist in the database and their correspondent stored data.
- **Create Agreement**: This service allows the creation of a new agreement between two subscribers that exist in the system.
- **Edit Agreement**: This service allows the editing of an agreement that exists in the system's database.
- **Display Agreements List**: This service allows the display of all existing agreements in the database, as well as its stored data.
- **Display System's Log (Alert Messages)**: This service allows the display of all the occurred events on the invoice mediation module. This service allows event filtering by data interval, by occurred events for a certain agreement or return of all events.
- **Exchanged Message Search by Subscriber**: This service allows the monitoring of the messages that a “Subscriber” exchanged by the invoice mediation module. This service allows the search by transactions on a date interval, by send/received messages e by state of transaction.
- **Exchanged Message Search**: This service is similar to the prior one but it allows the display of all the exchanged messages by the invoice mediator. This service allows the search of transactions by date interval, by agreement and state of transaction or return all the transactions made in the system.

- **Display my Agreements**: This service allows the return of all the agreements that exist in the database, belonging to the "Subscriber" user that calls the service. Allows a shortened display of the existing agreements, as well as its stored data.
- **Search my Exchanged Messages**: This service is similar to the "Exchanged Message Search by Subscriber" and allows the display of the messages the "Subscriber" exchanged with another, through the invoice mediator.
- **Search my Exchanged Documents**: This service allows the access to an invoice data that was exchanged in the system and finds itself stored in the database. Each "Subscriber" user has access only to its sent/received invoices.
- **Exchanged Invoices Report**: This service creates a report on all invoices the "Subscriber" user exchanged with other users through the invoice mediator. This service allows the search of exchanged documents by date interval, sent/received invoices for a certain "Subscriber", by total value interval of the invoice, by invoice sequence number interval, by invoice type and finally by the state in which the invoice is found in the system.

5 Conclusion

With this work we wanted to reach two goals: explore the electronic invoice business area, so we could implement a system that supported the business scenario previewed by the legislation, and explore the integration tools that were used as a technological platform to the system's implementation.

These goals were achieved, because since the beginning work was made to couple the business requisites with the system's development. Several technological options were tested and discarded to give place to those that were better suited to the system's performance.

6 Bibliography

[1] Bussler, Christoph - (2003) B2B Integration

[2] Chris O'Brien.: (2006) CEDPM B2Bi Guide: B2B Integration Framework for Backend Provider-oriented IT Applications

[3] Sami Jauhiainen, Olli Lehtonen, Pasi-Pekka Ranta-aho, Nicolas Rogemond .: B2B Integration – past, present, and future.

[4] Feras T. Dabous, Fethi A. Rabhi, Pradeep K. Ray - (2002) Middleware Technologies for B2B Integration

[5] Mira da Silva, Miguel – Integração de Sistemas de Informação, FCA

[6] Sundstrom, Jessica - Master´s Thesis. Adoption of Electronic Invoicing in SMEs

[7] Brahim Medjahed, Boualem Benatallah, Athman Bouguettaya, Anne H. H. Ngu, Ahmed K. Elmagarmid.: (2003) Business-to-business interactions: issues and enabling Technologies

[8] Understanding BizTalk Server 2006, Microsoft Corporation

[9] UMIC.: Guia da factura electrónica, Contributo para a modernização das empresas e do estado