

**WING ELECTRONIC DATA INTERCHANGE (EDI)  
TRANSLATOR FOR HIPAA (EDITH)**

Presented By

Williams Information Network Group Inc. (WING)



## **ABSTRACT**

This document will outline the development of a WING Electronic Data Interchange (EDI) Translator for HIPAA (referred to as EDITH). It will illustrate the planning, development and implementation of EDITH.

# TABLE OF CONTENTS

1.	System Overview .....	3
2.	Functional Process Flow for an EDI Transaction .....	3
2.1.	Inbound Transaction Process .....	3
2.2.	Outbound Transaction Process .....	4
3.	Software Technology Implementation .....	6
3.1.	Operating System .....	6
3.2.	Processing Software .....	6
3.2.1.	BizTalk .....	6
3.2.2.	SQL Server .....	7
3.2.3.	Communication Software .....	7
3.2.4.	Web Services .....	8
3.2.5.	Message Queuing .....	8
3.2.6.	Microsoft Operation Manager .....	9
3.2.7.	Microsoft Application Center .....	9
4.	Hardware Technology Implementation .....	10
4.1.	BizTalk Server Group .....	10
4.2.	SQL Server Cluster .....	11
4.3.	Storage Area Network .....	12
4.4.	Proxy Server .....	12
4.5.	Web Servers .....	13
4.6.	Tape Library .....	14
5.	Architecture Diagrams .....	16
5.1.	High Level System and Interfaces .....	16
5.2.	Hardware/Software Correlation .....	17
5.3.	Hardware Interconnectivity .....	18
5.4.	Physical Hardware Locations .....	19
5.5.	EDITH Communications Layers .....	20
5.5.1.	Trading Partner Communication .....	20
5.5.2.	The MMIS Communication .....	22
6.	Architectural Process Flow .....	22
6.1.	Trading Partner File Submission .....	23
6.2.	Processing of Inbound X12 Documents .....	24
6.3.	MMIS Communication .....	25
6.4.	Processing of Outbound X12 Documents .....	26
6.5.	Outbound Trading Partner Communication .....	28
6.6.	Processing of Non-X12 Inbound Documents .....	28

## LIST OF FIGURES

<b>Figure</b>	<b>Page</b>
Figure 1: Inbound Transaction Process.....	4
Figure 2: Outbound Transaction Process.....	5
Figure 3: High Level System and Interface.....	16
Figure 4: Hardware and Software Correlation.....	17
Figure 5: Hardware Interconnectivity.....	18
Figure 6: Physical Hardware Location.....	19
Figure 7: Trading Partner Communication.....	20
Figure 8: Client Network Communication.....	22
Figure 9: Inbound Transaction Process Flow.....	25
Figure 10: Outbound Transaction Process Flow.....	27

## **1. System Overview**

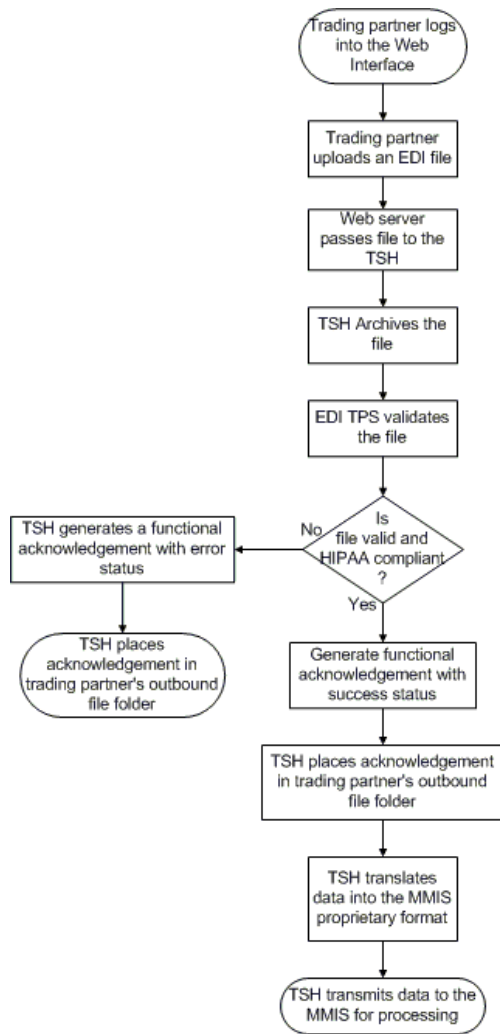
WING EDITH is used to achieve HIPAA compliance, while minimizing the impact to the client's internal system. EDITH acts as an intermediary to accept and send HIPAA compliant data to external systems while sending proprietary formatted data to the Medicaid Management Information System (MMIS). The benefit of this architecture is that it limits the modifications to the current processing logic on the internal systems; thus allowing interfaces for outside organizations to be left intact. This will also allow outside organizations to adopt support for HIPAA on different schedules and still be able to integrate with MMIS.

## **2. Functional Process Flow for an EDI Transaction**

For a basic understanding of EDITH and its functionality, the following flow charts illustrate a high-level inbound and outbound EDI transaction.

### **2.1. Inbound Transaction Process**

A trading partner will access EDITH through a Web Interface, which they will log in using a unique id and password. Once logged in the trading partner can upload their X-12N formatted file, which is passed onto EDITH by the Web server. Every file received by EDITH is first archived in its original form. The information sender's identification is then verified as being a valid and certified trading partner. If the information is valid a functional acknowledgement with a success status is generated, if not, the system generate a functional acknowledge with an error status. These acknowledgements are then place into the trading partner's outbound folder. Once a trading partner has been validated successfully the THS processes the transaction and send it to the MMIS for adjudication.

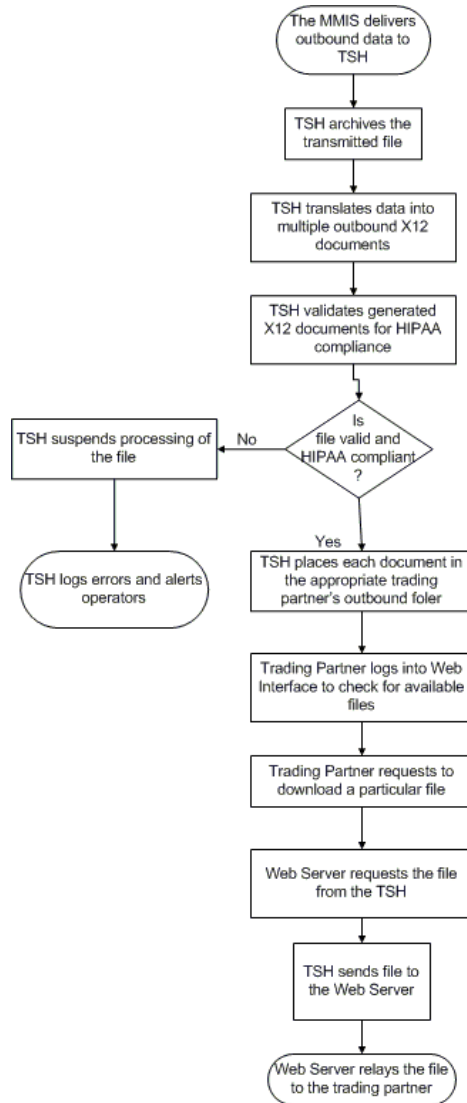


**Figure 1: Inbound Transaction Process**

## 2.2. Outbound Transaction Process

When EDITH receives and outbound file from the MMIS, the file is archived in its original form. The file is then translated into the X-12N format and is then validated for HIPAA compliance. If the file is found to be non-compliant EDITH suspends the processing and operations are alerted. If the file is valid EDITH places the documents within the

appropriate trading partner's outbound folder. The trading partner will then log in, at their convenience, into the Web interface to check for available file. If files are available, the trading partner can request for that file to be downloaded. The Web server will send that request to EDITH which will then forward the particular file to the Web server for downloading.



**Figure 2: Outbound Transaction Process**

### **3. Software Technology Implementation**

EDITH integrates commercial off-the-shelf software (COTS) for EDI processing, custom developed processing software, custom communications software, and enterprise-class servers and storage devices. The products used to build EDITH are discussed below.

#### **3.1. Operating System**

The entire EDITH system is Microsoft-based. It is running both Windows 2000 Server and Advanced Server. Advance Server is used on the database server for its clustering capabilities when integrated with Microsoft Cluster Services. EDITH is built on the Microsoft Active/Active Cluster technology, which allows for two or more service to share a work load under normal circumstances and to maintain operation with a reduced number of servers in the event that any of the servers fail. Even though during a fail-over, the performance of the system will degrade, due to the higher workload of the remaining server, the functionality of the system can continue to run.

#### **3.2. Processing Software**

EDITH employs two major COTS products for the processing of EDI transactions. These products are Microsoft BizTalk Server with BizTalk Accelerator for HIPAA 2.0 and Microsoft SQL Server. These products perform the translation, HIPAA validation and data processing and storage for EDITH.

##### **3.2.1. BizTalk**

Microsoft BizTalk Server 2002 is the core workhorse that enables EDITH to integrate with trading partners and MMIS through five key processes:

- document transport and routing,
- data transformation,
- application integration,
- process automation, and
- scalability and manageability services.

Simply stated, these processes provide prepackaged functionality that accepts, translates, and appropriately routes the EDI transactions that pass through EDITH. Each process is achieved through receive functions and channels. Receive functions are the starting point for documents submitted to BizTalk Server. They monitor specific file folders and trigger the translation process when new files are detected. Channels modify the structure of these documents to make them easier to translate, as well as manage encryption, digital signatures, and logging.

BizTalk is augmented with the BizTalk Accelerator for HIPAA 2.0, a powerful combination of product enhancements, simple-to-use tools, documentation, and samples that are developed specifically for BizTalk. The HIPAA accelerator provides up-to-date HIPAA schemas, and even incorporates the latest implementation guide addenda. Using this COTS plug-in – which leverages the direct cooperation of WPC, the publishers of the HIPAA implementation guides – vastly reduces the time, effort, and costs associated with solution development, deployment, and management.

### **3.2.2. SQL Server**

SQL Server 2000 is an enterprise-class database and data analysis package. SQL Server provides core support for XML and the ability to query from any server on the network, and across a firewall. SQL Server is the primary data storage system for EDITH. BizTalk uses it as the central repository for in-process data, as the container for its work queues and status information, and as a central storage location for state and configuration data. Much of the custom code for EDITH, including the Web interface, relies heavily on SQL Server to store, manage, and retrieve data quickly and powerfully.

### **3.2.3. Communication Software**

WING EDITH employs both COTS and custom software to communicate with its interfacing systems. Primarily, these external systems are the Communications Handler

(a public interface to the trading partners), and the MMIS. The details of the communication processes are explained further in the Architectural Process Flow section.

#### **3.2.4. Web Services**

A web service is a technology that allows applications to communicate with each other in a platform and programming language-independent manner. A Web service is a software interface that can be accessed over a network through standardized XML messaging. It uses protocols based on the XML language to execute operations or to exchange data with another Web service.

EDITH hosts Web service methods, or Web methods, with which the Communications Handler interacts to send and receive data. The two main functions of the Web methods are to query EDITH file system for a list of files that are available for a given trading partner to download, and to request that a particular file is transmitted to the Web server so that the Web server can relay it to the trading partner.

#### **3.2.5. Message Queuing**

Message Queuing is another technology employed along side of the Web services to get inbound EDI documents over the network from the Web server to EDITH. Message queuing is a method by which a process or program can exchange data with an operating system-managed queue of messages. A message queue is an ordered repository of messages sent by one or many source applications, and read by the same or other applications.

EDITH houses a message queue to which the Web server sends messages which includes the inbound EDI documents. The documents are queued in the order of arrival, and then removed from the queue by a custom application that writes them to the file system. The folders to which the files are written is monitored by a BizTalk receive function, so the documents' queuing and subsequent writing to disk is what triggers the entire EDITH process.

### **3.2.6. Microsoft Operation Manager**

Microsoft Operations Manager 2000 (MOM) provides a comprehensive event collection, performance monitoring and alerting, and reporting enterprise services in the Windows 2000 environment. The most important feature of MOM is its collection of predefined rules designed to monitor various metrics and indicators for Windows and for several major applications. MOM collects this information from all parts of EDITH and aggregates them into a central event repository, which may be managed via a console application or a Web interface. Some of the relevant items MOM can monitor out-of-box include:

- All Critical Windows 2000 Services
- BizTalk Server 2002
- SQL Server 2000
- Internet Information Service (IIS)
- Active Directory Service
- Microsoft Message Queuing
- Microsoft Operating System Log
- Domain Name Service (DNS)
- Microsoft Terminal Server

MOM also allows the system administrator to create custom monitors and processing rules, including triggers to send alerts through e-mails, by paging personnel or take prescribed actions (run scripts, launch applications, disable a device or application). Such custom rules can also link to resources such as Microsoft Knowledge Base articles and EDITH white papers, so that when a particular event occurs, the operator can have instant access to related knowledge and information for solving the issue at hand.

### **3.2.7. Microsoft Application Center**

Microsoft Application Center provides all the services and features one need for the creating, deploying, and management of enterprise applications. Together with Microsoft Operations Manager, an operation-orientated monitoring and management tool, MAC

provides a powerful foundation for administering BizTalk solutions in an enterprise environment.

There are three major layers in MAC:

- User Interface – A windows application, Web interface, or command line utility that provide system health monitoring, Web server administration, and component management. Users may also define their own interfaces and controls.
- Feature Set – Utilities to automatically manage, adjust, and operate system functions including Windows cluster services, server load balancing, and synchronization and deployment of code and software upgrades.
- Operating System Enhancements – Hooks and add-ins to enhance existing Windows features, such as the Microsoft Management Console (MMC), Internet Information Server (IIS), COM+ component support, Network Load Balancing (NLB), and others.

#### **4. Hardware Technology Implementation**

The physical equipment housing EDITH software is equally as advanced and capable. The devices were chosen to obtain a data throughput level high enough to process the estimated future volume of transactions in the same amount of time as the current volume is processed by the MMIS. The hardware is grouped by function and is described in the following paragraphs to provide an understanding of how each group helps to achieve this goal.

##### **4.1. BizTalk Server Group**

One of BizTalk's greatest advantages is its ability to work in "groups." A BizTalk group shares the work load by supplying multiple servers with tasks from a single work queue. EDITH employs 4 BizTalk servers using this model. One server, the BizTalk Receive server, accepts incoming documents and adds them to the work queue. Three servers, called BizTalk Processing servers, work in parallel to remove and process items from the work queue. Each processing server has a copy of the BizTalk Accelerator for HIPAA 2.0 installed, whereas the Accelerator is not required for the Receive server.

Each server is a Hewlett Packard DL380 with the following specifications:

<b>Processors</b>	Dual Intel Xeon 2.80 GHZ
<b>Cache</b>	512 KB Level 2 ECC
<b>Memory</b>	2 GB ECC PC2100 DDR SDRAM
<b>Controller</b>	Smart Array 5i Plus
<b>Internal Storage</b>	3 36.4 GB Ultra320 SCSI 10,000 rpm, RAID 5 configuration
<b>Networking</b>	2 PCI-X embedded 10/100/1000 WOL
<b>Options</b>	Battery Backed Write Cache Enabler for Smart Array 5i
<b>Common Names</b>	BizTalk Receive Server, a.k.a. BRSSVR1 BizTalk Processing Servers, a.k.a. BPSVR2, BPSVR3, BPSVR4
<b>NetBIOS Names</b>	BTSVR1, BTSVR2, BTSVR3, BTSVR4

#### 4.2. SQL Server Cluster

Supporting the storage and data access needs of the BizTalk server are two clustered SQL Servers. Since BizTalk is a database-intensive application, the machines running SQL Server need to be rather substantial. They also require high levels of built-in redundancy, very fast data access, and superior throughput.

To achieve these goals EDITH contains 2 Hewlett Packard ML570s with the following specifications:

<b>Processors</b>	Quad Intel Xeon 2.0 GHZ
<b>Cache</b>	2 MB Integrated Level 3
<b>Memory</b>	6 GB ECC PC1600 DDR SDRAM
<b>Controller</b>	Smart Array 532 dual channel PCI
<b>Internal Storage</b>	3 36.4 GB Ultra320 SCSI 10,000 rpm, RAID 5 configuration
<b>External Storage</b>	2 Fibre Channel Host Bust Adapters tied to the SAN volumes
<b>Networking</b>	2 PCI 10/100 WOL

<b>Options</b>	3 600W Power Supplies
<b>Common Names</b>	SQLSVR1, SQLSVR2
<b>NetBIOS Names</b>	SQLSVR1, SQLSVR2, EDITHSQLSVR1, EDITHSQLSVR2

### 4.3. Storage Area Network

As mentioned in the SQL Server Cluster description above, BizTalk requires fast, reliable access to large amounts of data. This is a perfect job for a Fibre Channel Storage Area Network (SAN). The SAN is a set of external hard drive bays (or enclosures) controlled by a redundant pair of hot-pluggable hardware RAID controllers. The controllers have a SCSI interface to the drive enclosures and a Fibre Channel interface to three of EDITH servers through a redundant pair of Fibre Channel switches mounted in the SAN.

The SAN has the following specifications:

<b>Hard Drives</b>	28 72.8 GB Hot-Pluggable Ultra320 SCSI 10,000 rpm
<b>Raw Capacity</b>	2.04 TB
<b>RAID Setup</b>	4 Arrays, each using RAID 1 + 0 (striped mirrors)
<b>Effective Capacity</b>	903 GB
<b>Controllers</b>	2 redundant Modular Storage Array 1000 controllers
<b>Enclosures</b>	1 integrated 14-drive enclosure, 2 additional 14-drive enclosures
<b>Interconnect</b>	MSA 1000 6-port, 2GB/s Fabric switch
<b>Options</b>	Battery Backed cache
<b>Management</b>	hp Array Configuration Utility software

### 4.4. Proxy Server

The primary purpose of the Proxy Server is to offload disk access tasks to a server that is not responsible for document processing. The Proxy Server has a Fibre Channel connection to the SAN controllers and acts as a relay for servers that do not have Fibre

Channel connectivity, namely the BizTalk servers. When the BizTalk servers need access to data on the SAN, they access it via a mapped network drive to the Proxy. To the Storage Proxy, the SAN drives appear as local drives.

The tape library (described below) is also mounted on the Proxy Server, via a SCSI connection. The server runs Backup software and manages all backup jobs. Through Fibre Channel and network connections, the Proxy can access all of the hard drives in the system in order to facilitate backing up all data to tape.

An additional benefit of having a server that does not perform processor-intensive translation or database tasks, is that it can house any software that could affect system throughput if it were to run on a BizTalk or SQL server. Software that falls under this category includes Backup software, MAC, MOM, and all communication software.

The Proxy Server is a Hewlett Packard DL380 with the following specifications:

<b>Processors</b>	Dual Intel Xeon 2.80 GHZ
<b>Cache</b>	512 KB Level 2 ECC
<b>Memory</b>	2 GB ECC PC2100 DDR SDRAM
<b>Controller</b>	Smart Array 5i Plus
<b>Internal Storage</b>	2 36.4 GB Ultra320 SCSI 10,000 rpm, RAID 1+ 0 configuration
<b>Networking</b>	2 PCI-X embedded 10/100/1000 WOL
<b>Common Name</b>	Proxy Server
<b>NetBIOS Name</b>	ProxySVR

#### **4.5. Web Servers**

There are two Web servers tasked with handling all interaction with the public so that users do not maliciously or accidentally access to protected data on EDITH. Sometimes, the term “Communications Handler” is used to describe the Web servers because they

handle various forms of communication with the Client and trading partners. HTTP and Network routing are the two supported communications methodologies.

A firewall is located between the Web Server and the internal network on which EDITH is located. The firewall permits communication between the IP addresses of the Web servers and two IP addresses of EDITH. Only one or two ports are open for each of those internal IP addresses, so nearly all public access to the internal system is blocked. The open ports are specific to SQL Server access, message queuing, and Web services. See the interconnectivity diagram in the Architectural Diagrams section for details on these firewall policies.

Each Web Server is Hewlett Packard DL380 with the following specifications:

<b>Processors</b>	Dual Intel Xeon 2.80 GHZ
<b>Cache</b>	512 KB Level 2 ECC
<b>Memory</b>	1 GB ECC PC2100 DDR SDRAM
<b>Controller</b>	Smart Array 5i Plus
<b>Internal Storage</b>	4 72.8 GB Ultra320 SCSI 10,000 rpm, RAID 5 configuration
<b>Networking</b>	2 PCI-X embedded 10/100/1000 WOL
<b>Common Names</b>	Web Servers
<b>NetBIOS Names</b>	EDITHWEB1, EDITHWEB2

#### **4.6. Tape Library**

EDITH is a rather extensive system that handles a substantial amount of data. Therefore, it requires significant backup capabilities. To ensure that the system could recover from a disaster, the server's base configuration, all of the SQL Server data, and all of the file system data that accumulates within the day, must be backed up. With a terabyte of data to back up, there is a clear need for a tape library containing many backup tapes to provide the necessary space.

Managed by the Backup software on the Proxy server, the tape library automatically loads and unloads its tapes as it runs through the various backup jobs on a scheduled basis. The tapes are broken into groups dedicated to a particular backup job. For example, 3 tapes may be dedicated to BizTalk server images, 10 may be dedicated to SQL Server backups, and so on. The tape drive contains one “mail-slot” access tape that can be removed without removing an entire magazine of tapes. This slot is ideal for jobs that involve small, one-time backups, or very long term backups containing data that the operator wishes to purge from the system and store elsewhere.

The tape library is a StorageWorks MSL 5026 with the following specifications:

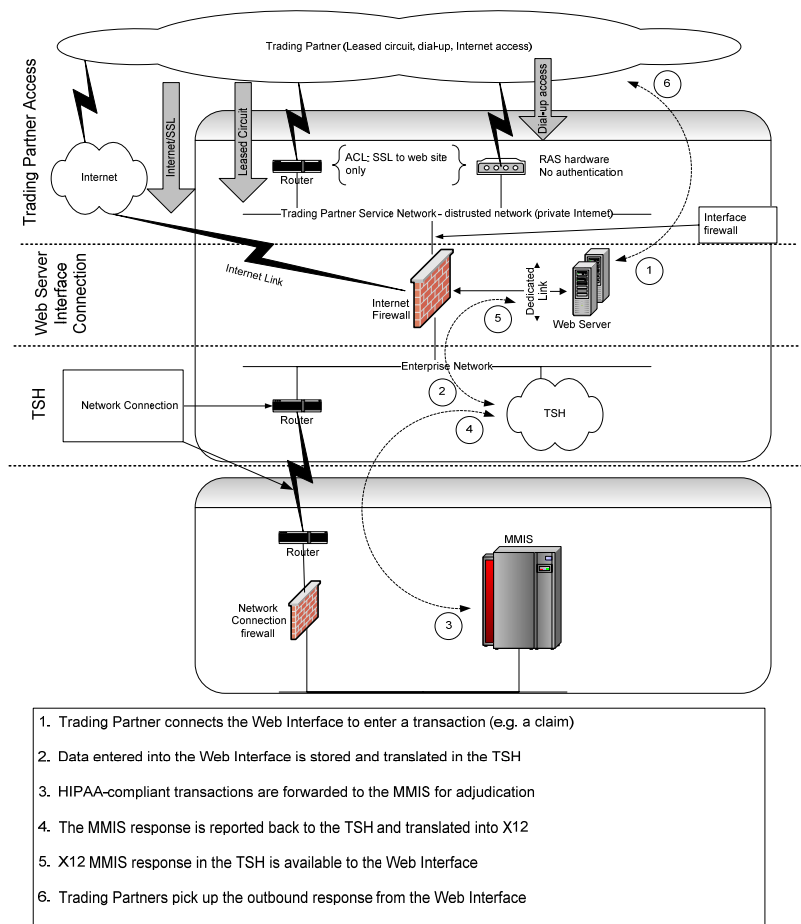
<b>Tape Drive</b>	1 SDLT 160/320 (expandable to 2 drives)
<b>Tape Size</b>	160 GB uncompressed, 320 GB compressed
<b>Tape Capacity</b>	25 internal tapes, 1 mail-slot tape
<b>Total Capacity</b>	4.2 TB uncompressed, 8.3 TB compressed
<b>Interconnect</b>	Dual Channel Wide Ultra3 SCSI adapter
<b>Features</b>	Robotic arm and bar code system for automatic tape loading

## 5. Architecture Diagrams

The lists of hardware and software do not fully explain how all of the components work together. The following diagrams aim to complete the picture by illustrating the logical and physical relationships among the interdependent systems.

### 5.1. High Level System and Interfaces

EDITH interacts with the MMIS and with many trading partners. This diagram shows how each system is related and how they interact. The following sections describe each system in detail. EDITH Communications Layers section describes the interconnectivity between the systems.



**Figure 3: High Level System and Interface**

## 5.2. Hardware/Software Correlation

Below is an illustration of all of the hardware in EDITH and the software that is installed on that hardware. Each list of installed software also shows the number of licenses required for the server in question.

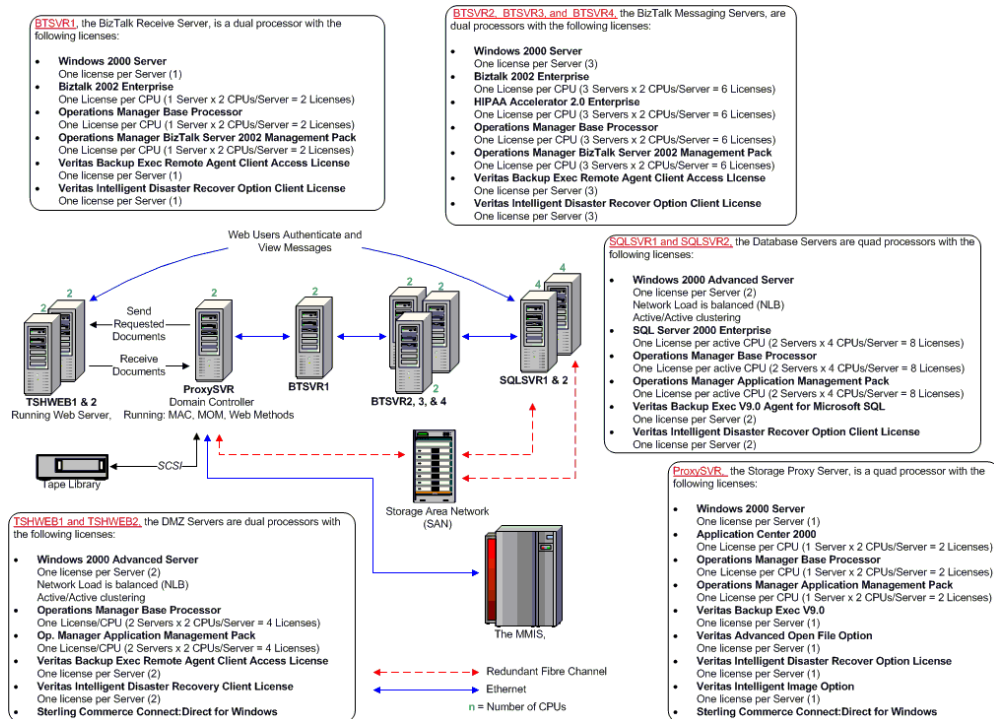


Figure 4: Hardware and Software Correlation

### 5.3. Hardware Interconnectivity

The following diagram illustrates the physical interconnectivity of each device in EDITH. The layout of the diagram is logical. Refer to the Physical Hardware Locations diagrams for the literal placement of each server within the racks.

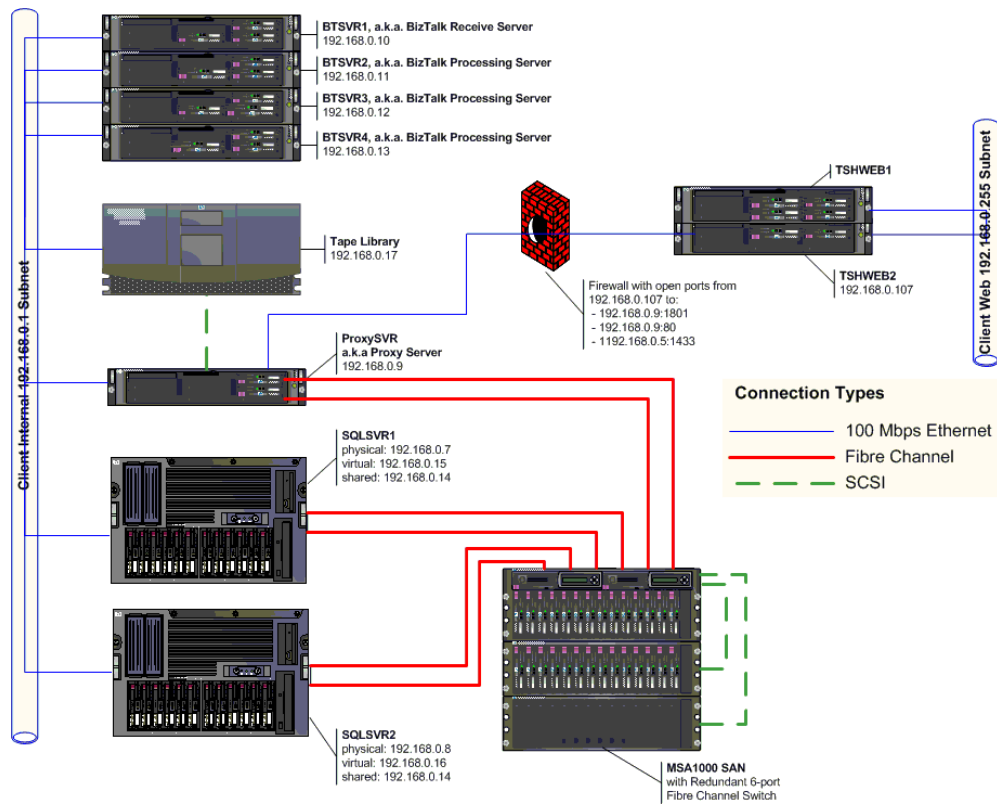


Figure 5: Hardware Interconnectivity

## 5.4. Physical Hardware Locations

These two diagrams illustrate the actual locations of EDITH hardware devices within the server racks, and the location of the racks within the server room. The diagrams can help an administrator locate a particular device to facilitate maintenance or upgrade.

Depicted below are the two server racks that house EDITH. Servers that are not part of the *Production* EDITH equipment are not labeled.

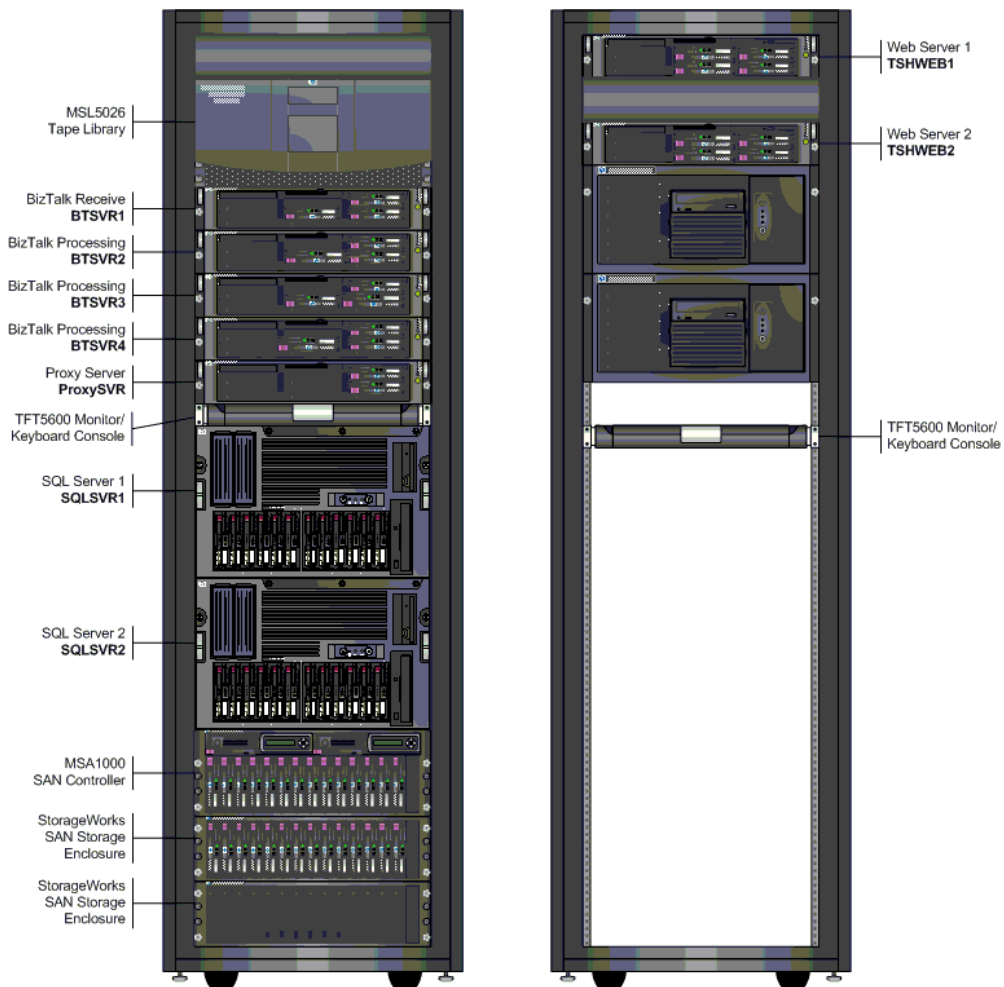


Figure 6: Physical Hardware Location

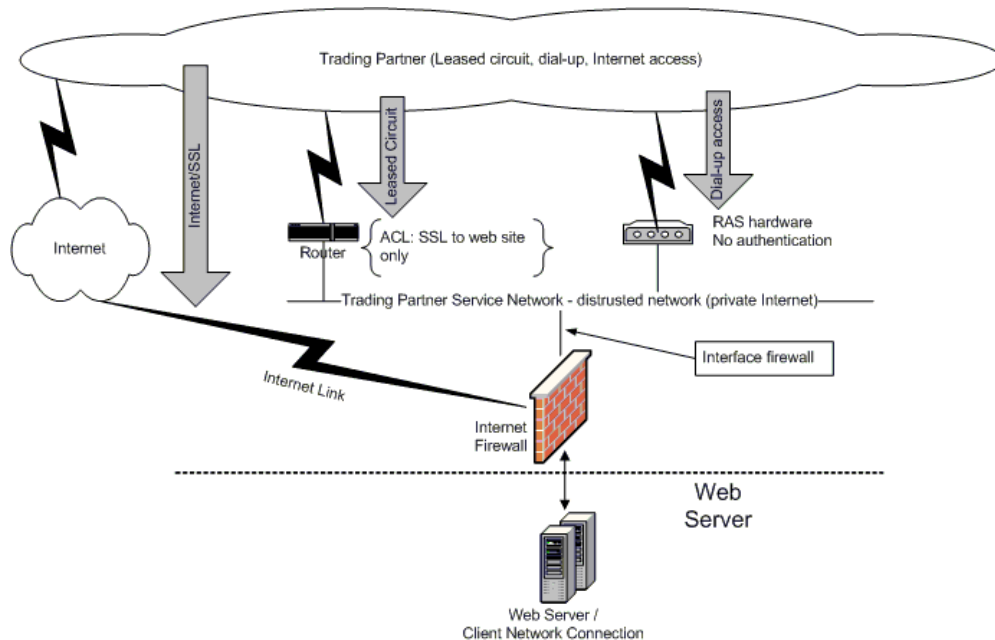
## 5.5. EDITH Communications Layers

EDITH operates in a reactionary mode, responding to input from either a trading partner or the MMIS. On the external side of the system, trading partners initiate EDITH process through an HTTP interface hosted by the Web server.

Communication with the MMIS is handled through a secure network connection.

### 5.5.1. Trading Partner Communication

Both HTTP and Network users access the Web server via TCP/IP using one of three possible connection vehicles: Internet, Dial-up, or leased circuit. See Figure 7 for a graphical representation. Trading partners who have internet access through an Internet Service Provider (ISP) simply connect to our URL or IP address as they would to access any other online resource.



**Figure 7: Trading Partner Communication**

The Web server supports a graphical user interface which acts as a front door to the system that provides an extra layer of security between EDITH and the rest of the world. Trading partners interact with the Web server and the Web server interacts with EDITH through another firewall that restricts access even more. Therefore, when a trading partner transmits a document to EDITH, the Web server relays it to EDITH, and no direct trading partner to EDITH interaction occurs. Likewise, when a trading partner requests an outbound document, the Web server turns the request around to EDITH, and then forwards EDITH's response back to the trading partner.

The firewall between the Web Server and EDITH allows inbound communication to 3 ports only:

- Port 1433 of SQLSVR1 (192.168.0.15) for database access
- Port 80 of the Proxy Server (192.168.0.9) for Web Services access
- Port 1801 of the Proxy Server (192.168.0.9) for MSMQ messages

### 5.5.2. The MMIS Communication

EDITH and the MMIS reside on an internal data network and communicate via TCP/IP. See Figure 8 for a graphical representation of how the two systems are interconnected.

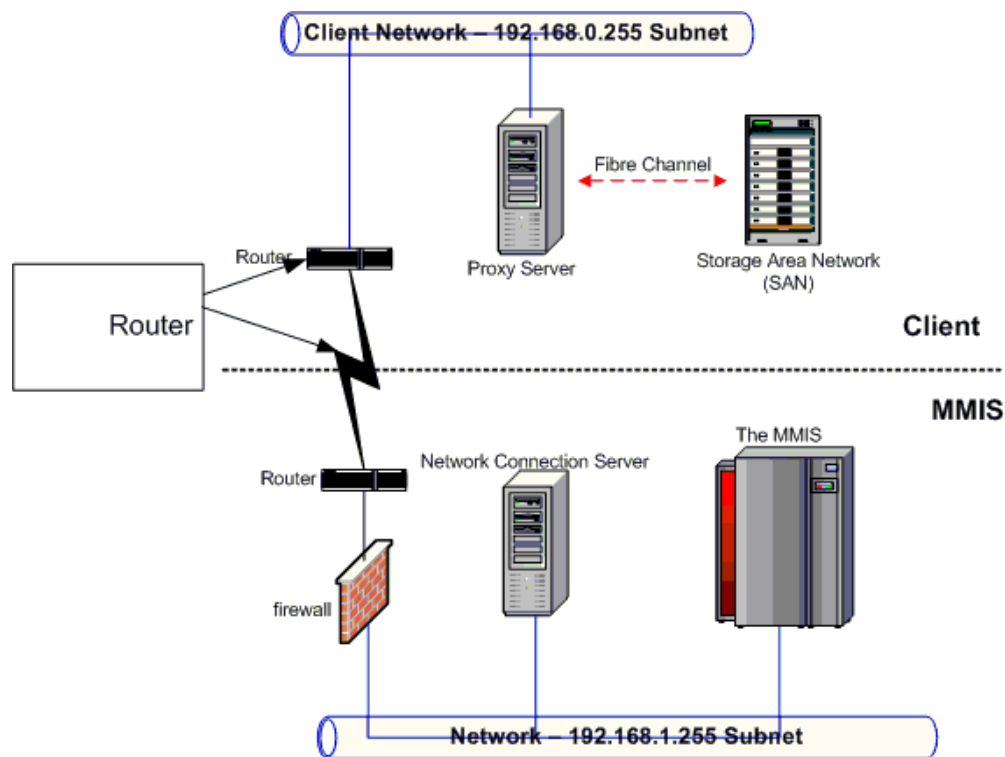


Figure 8: Client Network Communication

## 6. Architectural Process Flow

The flow charts in the Functional Process Flow section describe what EDITH does, but do not explain how each software and hardware component is involved in the process. The diagrams and descriptions below provide detailed illustrations of the steps and the roles each system component has in processing an X12 EDI transmission.

## 6.1. Trading Partner File Submission

Upon connecting to the Web server via any of the supported technologies described in the Trading Partner Communication subsection, users use a Web interface and URL provided by the Client to login to their account. Upon successful authentication the main “welcome” page is displayed. This page offers various options, including uploading X12 transactions, uploading non-X12 documents, and checking for available outbound transmissions to download. Account management and reporting features are also available through the Web interface. After a trading partner submits a document to the Web server, minor syntax checks are performed on the document, which are then passed onto EDITH. The syntax checks are designed to determine whether the document has X12 header information, the proprietary MMIS format header information, or neither. It also looks for an embedded data element that indicates if the file is a test transmission or if it contains real data bound for the production system. In addition, zip files are unzipped and their contents are processed as individual documents. If the inbound document fails the syntax checks or is not a readable ASCII or zip file, the Web server marks the document as a “bad file” with a code in the filename before sending it to EDITH. The Web server then sends inbound documents to EDITH via Microsoft Message Queuing (MSMQ). This is a technology that allows servers to communicate over networks with a minimum amount of security risk. To implement MSMQ, the Proxy Server in EDITH hosts a message queue. The Web server sends documents in the form of MSMQ messages to the Proxy Server through a hole in the firewall, and then the Proxy Server adds them to its queue. If the message is larger than 4MB, a performance barrier for MSMQ, the Web server first breaks the message into multiple 4MB messages and sends each individual message to Proxy Server in an ordered group called a MSMQ Transaction.

A custom application called the Message Monitor runs on the Proxy Server and monitors the queue for new messages. As they arrive, it removes the file from the queue and writes them to the inbound trading partner folder on the SAN, which is monitored by the BizTalk Receive server to begin the inbound processing for the documents. The message

monitor removes transactional groups of messages together and recombines them into one document before writing them to the BizTalk-monitored folder.

## **6.2. Processing of Inbound X12 Documents**

BizTalk Server uses File Receive Functions, which run on the BizTalk Receive server, to monitor specific folders on the SAN for new documents dropped off by external sources. When the Message Monitor writes a file to the inbound trading partner folder, a receive function that is designated to accept raw transmissions detects it and triggers the custom preprocessor running on the BizTalk Receive server, to begin the inbound document process. Before any parsing or translation takes place, the preprocessor calls EDITH Archiver component, which makes an exact copy of the raw transmission. The copy is stored on the SAN and remains there for 6 months before being backed up to tape for long term storage.

Simultaneously, the preprocessor looks for the “bad file” indicator in the transmission’s filename, and then either writes the bad transmission to a folder designated for files with errors, or passes good X12 transmissions to EDITH Interchange Parser. This common component parses X12 documents into easily digestible pieces. For instance, the parser makes it possible to isolate any given segment or envelope without parsing the file all over again. If the parser is not able to parse the file – because more in-depth validation determines that the file is not an X12 document or has egregious syntactical errors – the system’s standard error handler is invoked and passed a description of the problem. If the parser is successful, it creates a searchable object containing the individual EDI data elements to which the preprocessor can refer as it processes the data.

The Interchange Parser returns control to the preprocessor which then begins looping through each interchange found in the EDI file. The preprocessor calls another common component, EDITH HIPAA Validator, and iteratively passes it individual interchanges.

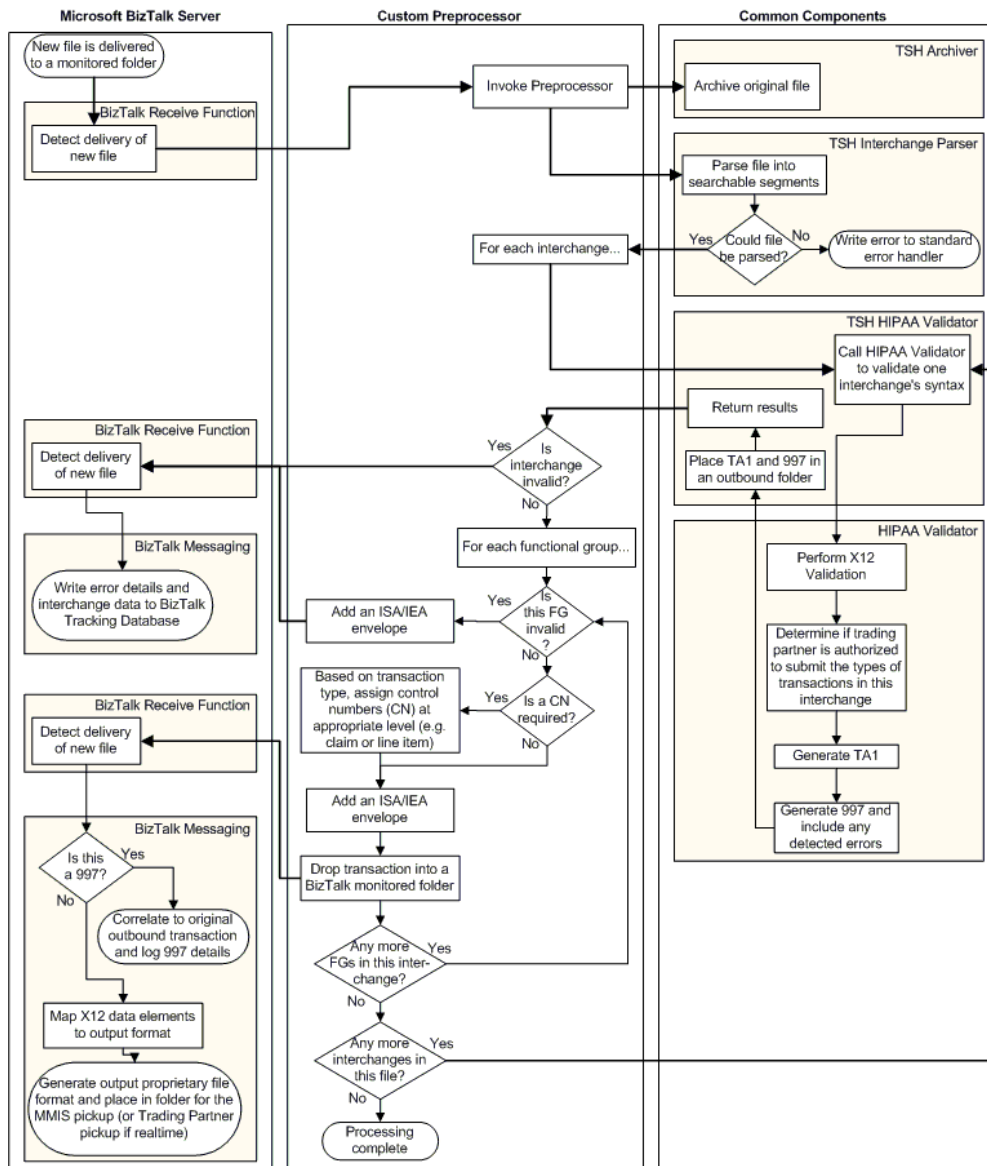


Figure 9: Inbound Transaction Process Flow

### 6.3. MMIS Communication

On a scheduled basis, which varies by transaction type, EDITH transmits translated transactions which are sitting in the MMIS-bound output folder, to the MMIS via a network connection. The transmission process is scripted and runs automatically without operator interaction. The scripts write to particular datasets on the mainframe. These datasets are

designated as the input locations for various MMIS processes. The MMIS retrieves and processes the data that is written to that location during its next processing cycle for each type of transaction.

Once the MMIS has finished processing the inbound data for a particular transaction type, it writes responses and output to another dataset. A scheduled script on the mainframe transmits the outbound data to EDITH in much the same way as the inbound process. The network connection to the Proxy Server receives the data and writes it to a folder on the SAN that the BizTalk Receive server monitors and initiates the process described in the next section.

#### **6.4. Processing of Outbound X12 Documents**

The outbound process begins when a BizTalk File Receive Function – running on the BizTalk Receive server – detects the file that the MMIS delivered to the SAN via network connection. This event triggers the outbound preprocessor which manages the flow of data through EDITH custom components.

Before any parsing or translation takes place, the preprocessor calls EDITH Archiver component, which makes an exact copy of the raw transmission which is stored on the SAN and remains there for 6 months before being backed up to tape for long term storage. Control is then returned to the outbound preprocessor where the MMIS file is checked for basic structural integrity. If the MMIS file is invalid, then the outbound preprocessor invokes the system's standard error handler, logs the description of the problem and suspends processing of the MMIS file. If the structural integrity check is successful, then the outbound preprocessor will parse groups of records into individual transactions which a BizTalk Processing server will later translate into X12 transactions. The outbound preprocessor writes each individual transaction to the SAN in a folder monitored by a transaction-specific BizTalk receive function running on the BizTalk Receive server. The receive function submits the transaction to the SQL Server-based

work queue. Working in parallel, the BizTalk Processing servers remove transactions from the queue and translate them into the X12 format.

The BizTalk Processing servers invoke a custom Application Integration Component (AIC) to pass the X12 documents to the HIPAA Validator common component which ensures HIPAA compliance.

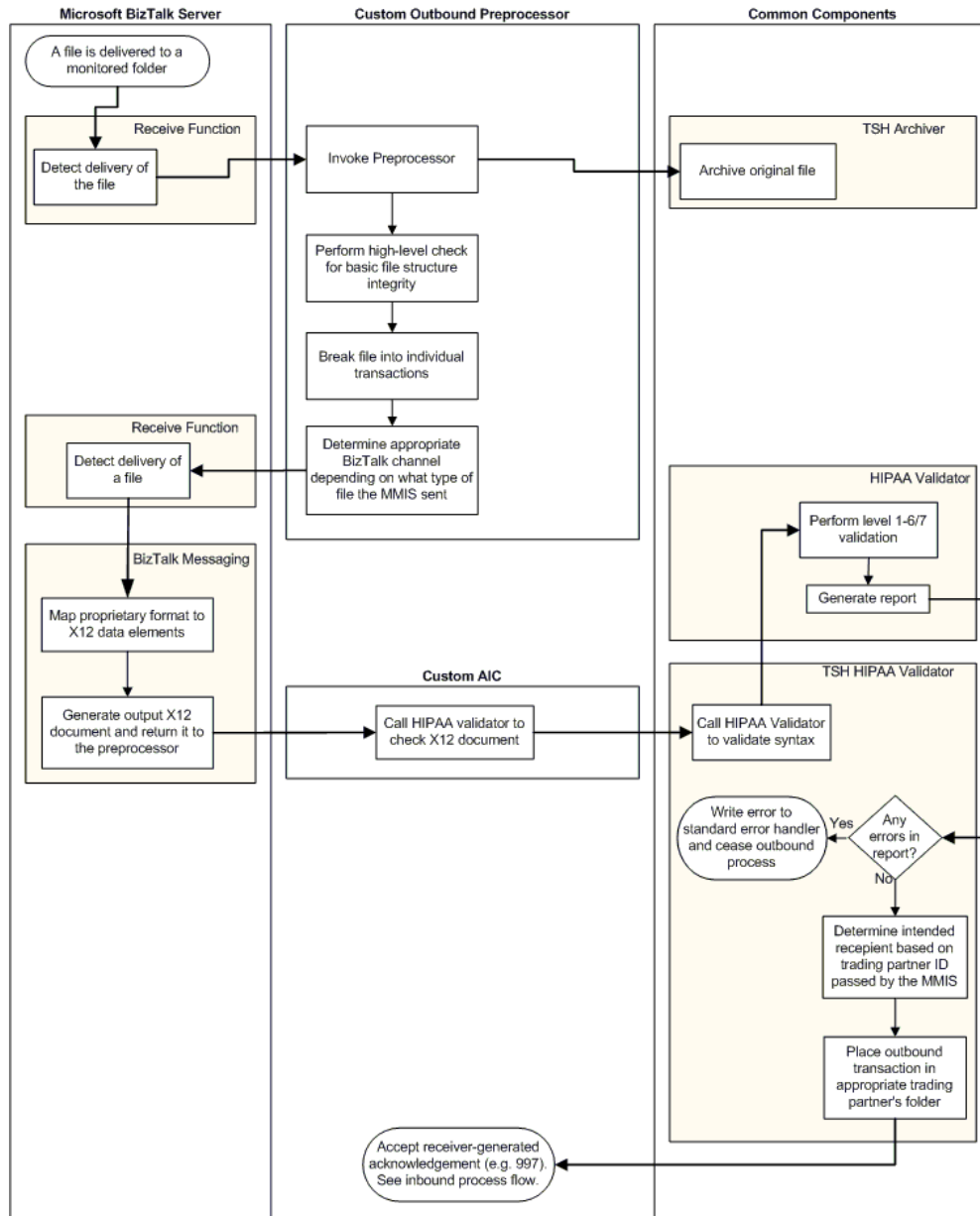


Figure 10: Outbound Transaction Process Flow

## **6.5. Outbound Trading Partner Communication**

EDITH transmits outbound documents to trading partners only when requested to do so. Trading partners must log into the Web interface on the Web server via Internet, dial-up or leased line, and use it to download outbound documents. One of the menu items on the Web application welcome page allows the trading partner to view a list of files that are available for the trading partner to download from EDITH. Each filename is a hyperlink on which the user can click to start the download process to a folder on their system, thus completing the end-to-end process for the transaction.

The primary technology behind the transmission of outbound documents to the trading partner is Web Services. When a user interacts with the Web interface, the Web server calls Web methods. The Proxy Server hosts two main Web methods: one to return a list of files available to download, and one to actually transmit a requested file. The first simply reads the contents of the outbound folder for the trading partner who requested the list. The Web method sends the available files' names, timestamps, and sizes in XML back to the Web interface, which formats and displays the data to the user. The second Web method, receives a request for a particular file and then streams it to the Web interface use Direct Internet Messaging Encapsulation (DIME), a message format protocol designed to simplify the transmission of binary data when using Web services. The Web interface relays the data to the trading partner who sees it as a downloadable file.

## **6.6. Processing of Non-X12 Inbound Documents**

The handling of Non-X12 documents begins when the Web application performs some basic syntax checks on files as trading partners upload them (see the Trading Partner File Submission section). If the Web application determines that a file is a Non-X12 document, it embeds a code in the file name before passing it to EDITH. The inbound receive function on EDITH detects the document and passes it to the preprocessor. The preprocessor looks for the code in the file name and, which allow it to know that is should

handle the file differently than it handles X12 documents. After archiving, the preprocessor launches a custom component that is designed to perform a few front-end edits (validations) on incoming files. If the file passes the edits, it is simply copied to a folder from which through the network connection be sent to the MMIS. If the file does not pass the edits, the custom Non-X12 component generates a detailed error report and writes it to the trading partner's outbound document folder.

The Non-X12 process is for inbound documents only. EDITH is not designed to send outbound documents in any format other than HIPAA-compliant X12.

