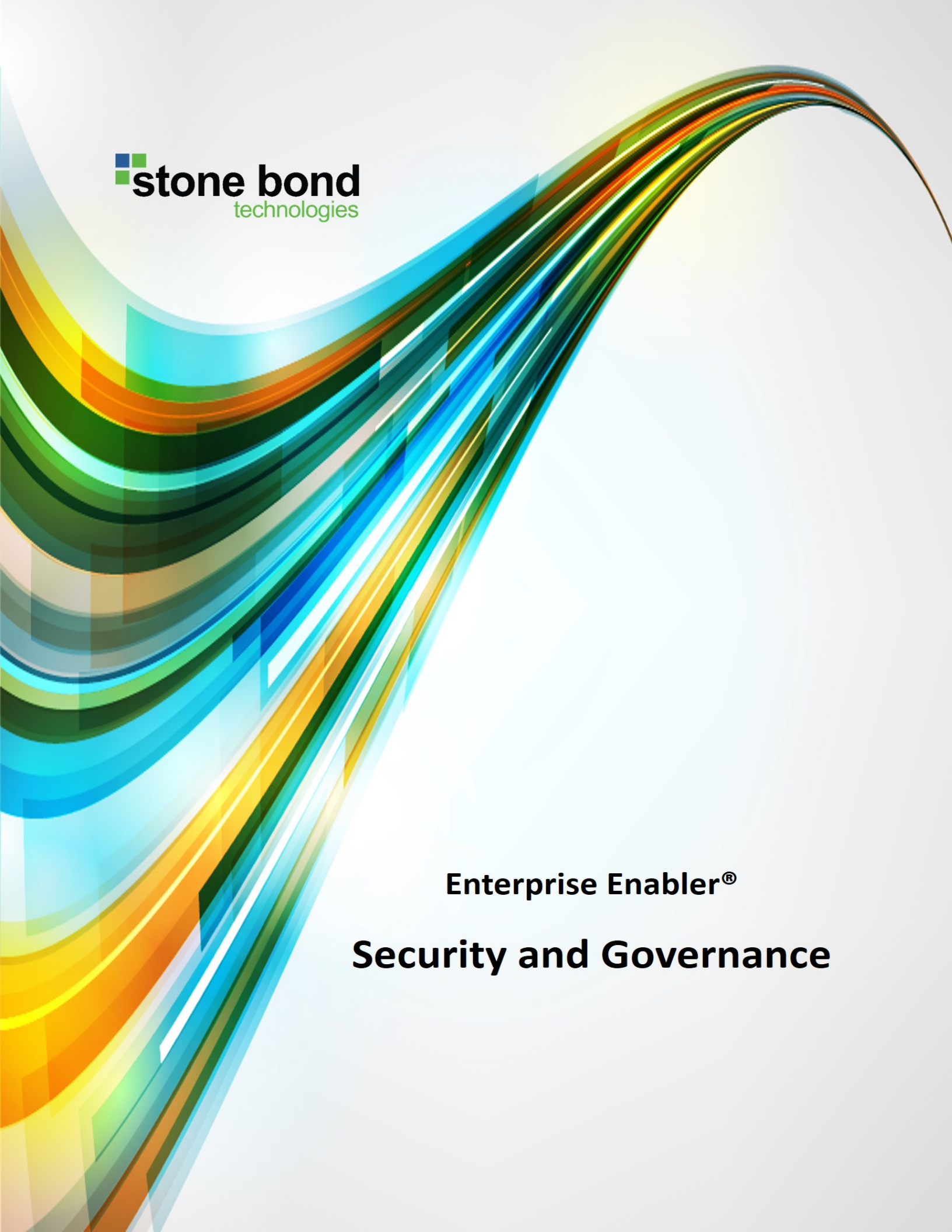

Enterprise Enabler®

Security and Governance

Stone Bond Technologies, L.P.
1021 Main Street Suite 1550
Houston, TX 77002
713-622-8798
www.stonebond.com

Contents

1 OVERVIEW

A holistic view of enterprise data security must include careful scrutiny of all the mechanisms and tools used to move data into, throughout, and outside to business partners of the company. This puts a strong focus on the enterprise middleware products integration architectures, and solutions that deal with sensitive data. Stone Bond Technologies' Enterprise Enabler® (EE) platform provides a rich system of security and assurance across the integrated environment that extends to data exchanges with outside entities.

Whether driven by specific government regulations, privacy restrictions, or self-imposed to maintain the competitive advantage or the confidence of its customers, every company has security requirements.

A unique aspect of Enterprise Enabler in comparison to other middleware products is it is a "closed system." That is, it is a single platform for development, testing, deployment, and monitoring of integrations. All integration components are 100% metadata-driven, where the metadata is generated from within the secure User Interface and executed by run time engines. A developer, or DBA, configures complex integrations without ever having to leave the platform in order to use a separate code development environment. This means that all integration is managed and contained within the single metadata stack, eliminating the typical breakpoints of stepping from one tool to another, both at development time and at run time. Each of those points becomes a potential point for breach of security.

Many perspectives of data security and infrastructure integrity are handled within the Enterprise Enabler system. At configuration, only users with correct, granular permissions are able to view, implement, or modify integration metadata objects. At run-time, a single control point executes the metadata instructions. Data can be encrypted in *situ* or in transit, moved with security such as SSL, and end user security can be implemented with SSS or Claims Authentication, among others.

The overarching benefit that all integration with Enterprise Enabler has live federation (sometimes called "data virtualization") available means that staging databases that exist for the purpose of aligning disparate data simply are no longer needed. EE federates across different sources (e.g., databases, applications, services, electronic devices, etc.) not only for data virtualization on-demand such as ODBC, JDBC, Web services, OData, and others, but also for ETL and EAI oriented integration patterns. Every database that is eliminated reduces the security risk.

Loosely speaking, data security and governance cover a very broad range of topics which include access, visibility, delivery assurance, failover, validation, loss, consistency across data stores and applications, and data breaches, all of which are integration break points where hackers can easily play.

This paper is intended to provide an overview of security and risk mitigation features available in Enterprise Enabler, showing specifically how some are configured within the Integrated Development Environment (IDE).

2 EE SECURITY FEATURES

EE provides a number of security features out-of-the-box that are easily configurable to ensure that the organization's data is stored and transported securely. Features such as Encryption, Authentication, SSL Transport, and Data Packaging, are all built in to the EE environment and accessible through a single unified platform that can be completely managed within the network of the organization. These features can also be used to build processes that enable the business to adhere to the security requirements specific to the industry that the organization operates in.

2.1 SERVER SECURITY THROUGH SECURE SOCKETS LAYER (SSL)

SSL is a secure data exchange standard commonly used when exchanging data with a business partner, as well as among remote servers. Enterprise Enabler supports SSL communication between the EE Client and EE Server. All services exposed by the server can be enabled to use SSL as the communication channel. It is easily configurable by setting up the Server & Studio Communication. Below is the Enterprise Enabler Security Manager Tool's configuration screen for enabling Https security using SSL. Enabling this type of security is simple. The administrator imports a certificate into a secure store on the server that runs the EE Service. Then the Server and Studio manager tool can be used to enable Https on the Server.

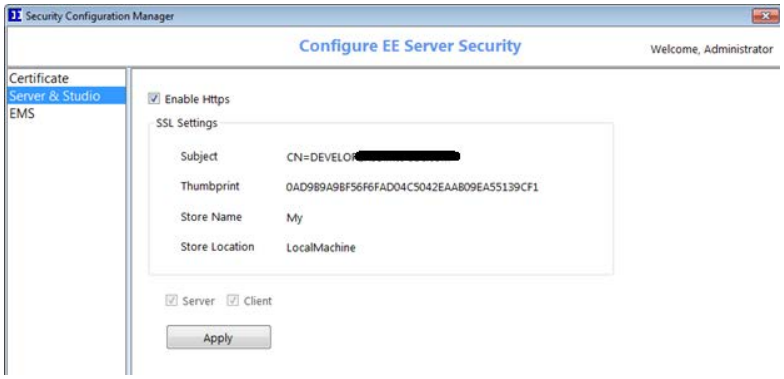

Figure 1: EE Security Manager - Enabling Https on the Server

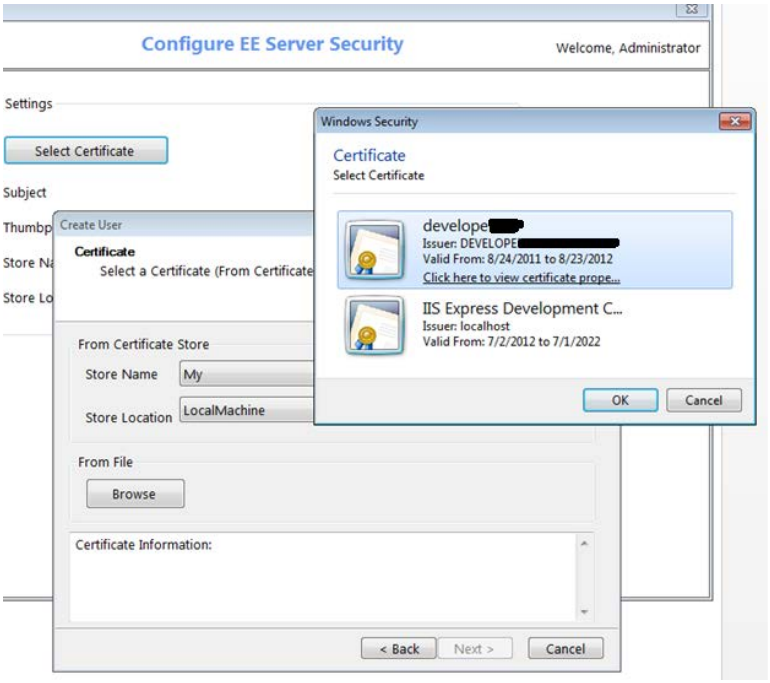

Figure 2 Selecting a Certificate

2.2 EE Web services Security

EE allows securing the web services that are hosted by the EE Server in the Enterprise Master Service's configuration using SSL. The EMS Security Manager allows the administrator to select the EMS Processes that needs to be secured and enable Https security. The administrator can select individual Web Service Methods that need to be secured and enable security using SSL. Underneath the covers, the EE Server generates secure proxy methods that may be invoked separately.

The EMS Services or web service methods can also be secured using Domain authentication for secure access. This feature is mentioned below.

2.3 Data Packaging Security

EE can package data for external consumption using multiple encryption algorithms. For example, if a file has to be sent to an FTP or Http destination, the data can be packaged using the RSA, Rijndael, or Triple DES encryption algorithms. Additional or custom algorithms can also be easily added.

These are easily configurable as shown in the screen below. An example of this function in the oil and gas industry is when they need to send PIDX files to an Http client using secure and encrypted files. The file can be encrypted and packaged using the RosettaNet packaging as part of the template.

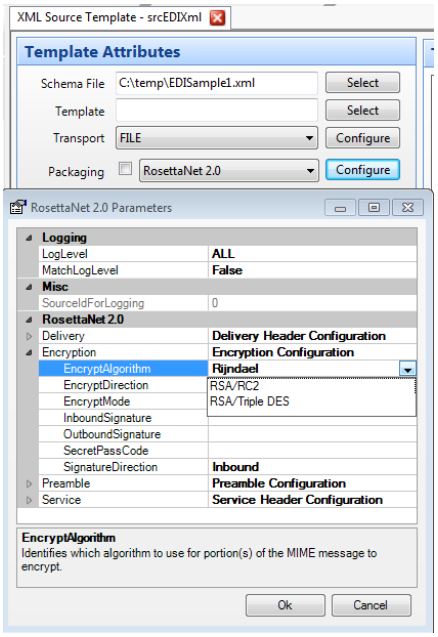

Figure 3: Data Packaging and Encryption Properties

EE also contains the RosettaNet process node which can process the incoming mime message and generates a RosettaNet response mime message if the flag is turned on. This mime message adheres to the Partner Interface Processes (PIPs) specification. PIPs specification is standardized by the RosettaNet Implementation Framework (RNIF) 2.0 specification.

2.4 MASKING

Data Masking or Data Obfuscation is the process of hiding the original data with random characters or other data to hide the original information leaving only the information necessary in the result. For example, social security numbers are usually masked by replacing the first 5 characters: xxx-xx-2345. Another example is for clinical trials to hide the names of subjects by only using first and last initials. While usually masking does not need to be retrievable back to the original source values, sometimes it may be necessary and can also be configured securely if needed. Each industry or organization has its own standards and protocols for masking data.

Enterprise Enabler can be configured easily to implement masking practices that are required by the organization through functions in the map object during ETL. Functions can be created to mask the data and create the desired output at an individual field or column level. For lookups, the map can refer to another source for masking purposes. Encryption of these look-back rules is done automatically within the metadata, and at other levels using various encryption algorithms.

2.5 SSS SECURITY

EE supports third party single sign on, as well as security providers such as the SharePoint Secure Store Service (SSS), for end user security control. Below is a way EE can configure its Templates to use SSS to store the credentials in the SharePoint secure database for Single Sign On capabilities. This means that from SharePoint, Enterprise Enabler ensures that an end user will only see data they are authorized to

see, and can only write back to endpoints when they have the SSS authorization to do so. Enterprise Enabler is designed to support other security models as well.

Security	
GroupName	**Finance**
SecureStore	**SBT.EE.Security.SecureStore.SecureStoreService**
Enabled	**False**
Fields	**(Collection)**
TargetApplication	
SharedWithinGroup	**True**

Figure 4: Enable End User SSS Security

2.6 Two Phase Commit

Two Phase Commit (2PC) is a type of Atomic Commitment Protocol that ensures consistency across multiple endpoints in the case of a failure to write or update any one endpoint. It is a distributed algorithm that coordinates all the processes that participate in a distributed atomic transaction. Ultimately, such algorithm determines whether to commit or abort all transactions.

Enterprise Enabler can be configured for Two Phase Commit using built-in Transaction features at the Enterprise Master Service level or at data base level. These features can keep track of all running maps in an atomic transaction scope and commit if all maps are successful or rollback, if any one of them fails. This assists in keeping the data integrity in-tact and making sure all data sources are accurate and secure.

For example, from SharePoint when a record is updated using an external list, Enterprise Enabler can update multiple data sources, such as Salesforce, SAP, and SQL etc. If any data source update fails when updating, deleting or creating records, the entire transaction is rolled back to its original state.

3 High Availability (Clustering or Failover)

Enterprise Enabler Sever is fully high availability capable. Enterprise Enabler runs as a hosted instance that can be clustered or Network Load Balanced as per the organization's IT needs.

3.1 Database Clustering and Failover

Enterprise Enabler Server databases are housed on a clustered instance of Microsoft SQL Server, which can utilize all the SQL Server Clustering features. In addition, EE has a built-in cache where all data is saved before being saved in the SQL Server. If the database experiences an outage, EE will continue to run normally, storing all the data in memory until the database connectivity is restored. Once the database is back online, the EE server will flush out its cache and resume normal operations.

3.2 Server Clustering and Failover

The EE server can be hosted behind a network load balancer to provide network load balancing and redundancy. EE can also be enabled to store all its transactions in a separate message queue that can be hosted in an SQL Server for clustering purposes. Multiple instances of the EE Server can be deployed simultaneously, all using a shared queue for processing incoming and outgoing messages. For this type of configuration EE has a built-in Message Box Process Node that can be dropped onto the EE Process Designer. This feature is useful for processing incoming messages from an outside party so that no messages are dropped.

4 Reverse Proxy or DMZ

A reverse proxy server is a security device that is usually deployed in a DMZ network to protect Http servers on a corporate intranet. This protection is achieved by performing security functions that shield the internal servers from attacks by users on the Internet. It's a type of proxy server that retrieves resources on behalf of a client from one or more servers. These resources are then returned to the client as though they originated from the proxy server itself. A reverse proxy acts as an intermediary for its associated servers and only returns resources provided by those associated servers. In Enterprise Enabler Server implementations, such a service is typically provided via the IIS Web Server Gateway. The IIS Web Server gateway will need to be routed to the EE Server Host Machine, and the tcp ports 6443 and 6444 or 64445 (if using SSL) will need to be opened.

5 Unified Environment

In EE, the security features and processes are configured entirely in a single application environment. All the security settings can be managed within EE without the need for leaving and going to an external tool. Enterprise Enabler has been built from the ground up with security in mind and all the security configuration points are accessible in the appropriate context. For example, a node can be selected, and through its properties all its security configurations can be set. This makes the job of the Administrator easy, with all the configurations in one secure environment.

The unified environment means that every user is known and carries permission controls, every change is time-stamped and versioned, the state of any execution is available throughout the run-time logic, overall integrity can be validated, and logging, error handling, and notifications are universal.

One hundred percent of the configuration is captured as metadata and stored in the end-to-end metadata stack, so it is possible to traverse the metadata in one place for a complete end-to-end picture of all integrations built within the Integrated Development Environment.

5.1 Encrypted Metadata Storage

All the EE Metadata such as template objects, maps, connection information, etc. is stored securely by encrypting sensitive information the EE database. In case of a metadata breach, the information is encrypted so that the data cannot be deciphered and used. For example below is how the connection

information is stored for a particular SQL Server connection with the username, password and data source information all encrypted. This is a default feature and does not need to be configured manually. No actual data involved in an integration is saved.

```
<eecon id="CE66902EBDCF4300A0EC476B06C6C444" name="[redacted]" version="1" cluster="Default"
createdon="2/13/2013 1:48:54 PM" computer="[redacted]" shared="True" ConnObjectType="sqldb"
DataSource="SwdqQEiPqqW3EO4473YtRw==" InitialCatalog="Zn3DbsFyMCV3WcpRgiytyA==" Password="M
+4GqC5XnxSfApDBXjomQw==" UserID="DZSHmQm6tSs=" category="EDI MESSAGES" busowner="{1}"
groupid="{11F6E344-A146-4BE7-A6F5-21AD03CCE744}" type="EECon" author="{1}"
assemblyname="SBT.EE.AppComms.SQL.dll" assemblypath="." assemblyversion="8.0.0.0"
classname="SBT.EE.AppComms.SQL.AppCommConnection.SqlConnectionADONET"
assemblyqualifiedname="SBT.EE.AppComms.SQL.AppCommConnection.SqlConnectionADONET, SBT.EE.AppComms.SQL,
Version=8.0.0.0, Culture=neutral, PublicKeyToken=9b1026809be0a5be">
  <conInfo AppConnSubType="SQL-ADONET" />
</eecon>
```

Figure 3: Connection Information is Encrypted and Stored

5.2 Role Based Security

All integration configuration is done within the Integrated Development Environment (IDE), ensuring that unauthorized users cannot change any aspects of the integration configuration. Historically, this particular risk had no mitigating approach beyond defining business processes and best practices. For example, any programmer could modify the code to divert a data flow to be sent to an additional destination without leaving a discernable footprint. EE supports Role Based Authentication using either the internal EE Security manager, Active Directory, or LDAP Authentication. Roles and Groups can be configured according the organization's requirements, and users can be assigned to those roles. Not only must the user configuring the integration have permissions to make changes, but also every change that is made carries a version number, timestamp, and user's name. Specific privileges can be assigned to each user or group in the Security Manager screen shown below.

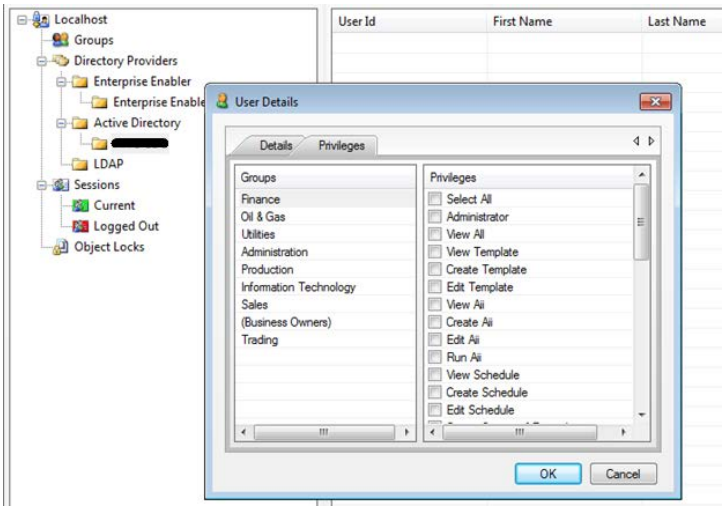

Figure 6: Active Directory Roles

5.3 Integration Integrity Manager

Leveraging the integrated metadata stack, the patented IIM ensures that any changes applied to the integration will not cause a conflict with any other object in the stack. For example, if any of the data being accessed is changed and that template is used in other places, the IIM will validate that the change will not impact another template. If the IIM discovers that it will impact another template, it prohibits saving/deploying the change and presents the information of exactly what would be impacted and how.

The IIM also monitors endpoints for changes at integration touch points. If a change is made to a field data type in a database, for example, it is automatically detected and a notification is sent identifying the change and the impacted integration objects.

6 Logging, Auditing and Reporting Capabilities

6.1 Logging

One of the benefits of EE's single comprehensive environment, in conjunction with the single metadata stack, is an overall consolidated view of the configuration and flow of data throughout the enterprise. Contributing features include:

- Every metadata object is versioned, and each change is marked with what was changed, who applied the last change, a timestamp, and the option to include a comment.
- Logging can be turned on at many levels and set to capture a wide range of events.
- The logs can be captured and reported using many reporting tools.
- It is possible to traverse the stack to discover the path that data follows throughout the organization, because all of the metadata is contained within a single stack.

EE logs all events that occur within the EE Client and Server. These logs provide extensive information about the processes, services, and transactions that are occurring during the integration process. All log events are captured in the EE Database locally. These events can be later viewed and analyzed for reporting within the EE Studio. Some examples of log events that are captured are: Process Execution Details, Transaction Logs, Errors, and Audit Information. This provides extensive logging for performance tuning, fine debugging, and for auditing and reporting to the organization. EE also captures version history of all the EE Objects in the system for version control and roll back.

6.2 Auditing

An important role of logging is to provide information for security auditing to demonstrate the flow paths of data, a record of sources of data that is accessed and moved, business rules, user access, and any other aspect of the integrated environment. Much auditing can be done by looking at the actual metadata from the development interface, since it is easy to see what sources are being used and how the data is aligned and what rules have been built around any data integration. Below is an example of some logged data.

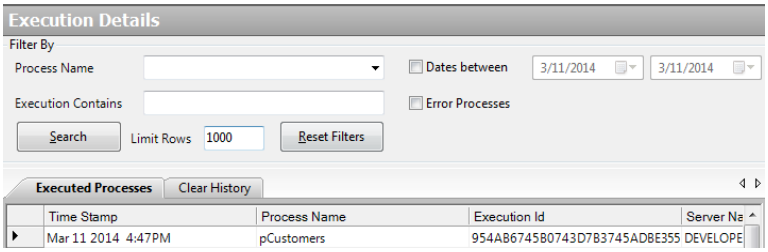

Figure 7: Process Execution Details

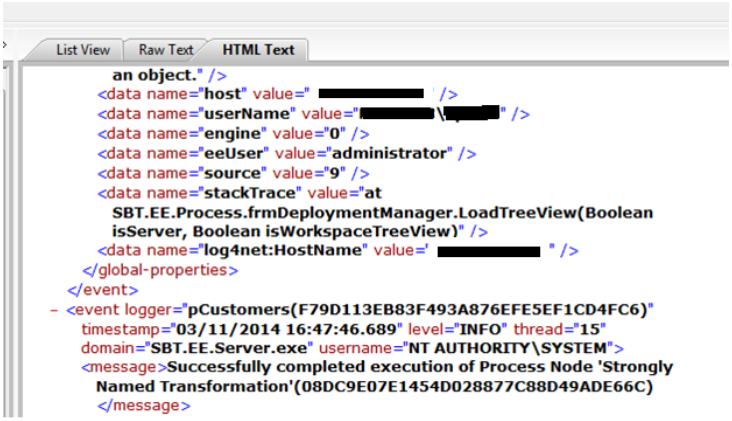

Figure 8: Sample Logged Information

6.3 REPORTING

Enterprise Enabler has a variety of reports that cover different elements within the application or you can utilize Microsoft SQL Server Reporting Services (SSRS). Below is an example of the Enterprise Enabler Session Security Log Report that can be generated from Microsoft SQL SSRS. With this report an administrator can see the log of all the EE Security events that have occurred. Other similar reports such as Process Execution Logs, Event Logs and Error Logs can also be generated with EE log data that is captured in a central SQL Server.

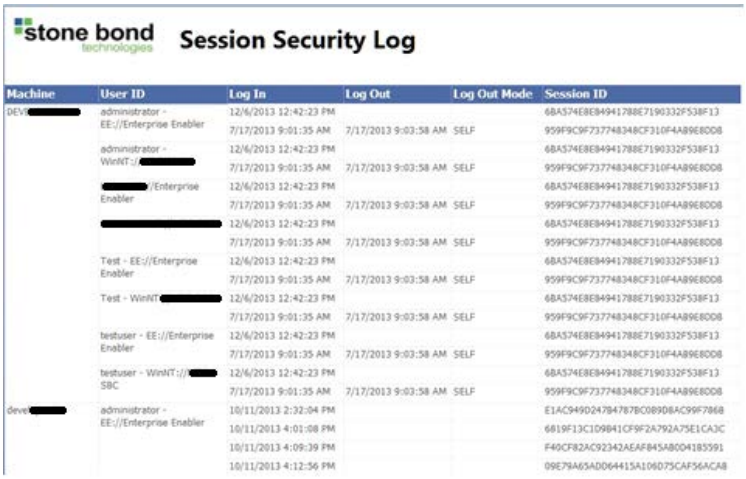

Figure 9: Sample EE Security Log Report showing Machine name, Username, etc.

7 PAYMENT CARD INDUSTRY (PCI) – HEALTH INSURANCE PORTABILITY AND ACCOUNTABILITY ACT (HIPAA) CONSIDERATIONS

The PCI Data Security Standards for payment card data are an example of robust and comprehensive standards for specific industries. For Healthcare there are HIPAA and HL7 standards. EE can enable organizations to comply more easily with standards like PCIS and HIPAA. Here we look at PCIS requirements for enhancing data security to meet Financial Institutions' requirements. Some of the requirements from the PCIS standards and how EE helps achieve those are:

PCI-DSS requirements:

- ***Build and Maintain a Secure Network***
  EE secures the data communication channel using SSL so that data flowing through client and server cannot be compromised.

- ***Protect Customer Data***
  EE does not stage any customer data internally. Data is entirely processed in memory and is not exposed or stored.
  Data can be protected by various encryption mechanisms and packaging.

- ***Implement Strong Access Control Measures***
  EE runs in a controlled environment (behind DMZ and Firewall) by restricting access to only valid users.
  EE validates data using configured rules and takes actions as configured. These actions can be logged if needed.
  EE allows users to manage credentials using their own providers, such as SharePoint SSS.
  EE restricts access to all its objects based on roles.

- ***Regularly Monitor and Test Networks***

  EE tracks and monitors all access to network resources and connections.
  EE maintains a security log of all users who are trying to access and modify objects and data.
  EE keeps up-to-date with the latest security updates and provides regular patches.

- ***Generate Audit Support Documentation***

  As described earlier, EE captures all the configuration information.
  EE captures run-time logs showing data flows, error situations, as well as run time logging of errors and exceptions.
  EE's Integration Integrity Manager (IIM) monitors the data sources for unexpected changes in endpoint applications and databases. It then reports the potential impact of each change detected.