Program Name:	**Department of Energy iManage Program**		
Project ID:	**iManage Program STRIPES Project**		
Project Manager:	**Mathew Sparks**		
Program Mgr:	**Lajos Grof-Tisza**	**Doc ID:**	

U.S. Department of Energy

Office of the Chief Financial Officer

iManage Strategic Integrated Procurement Enterprise System STRIPES

Technical Architecture Overview

Status of Document

☒ **Draft**

☐ **Delivered**

☐ **Accepted**

Program Name:	**Department of Energy iManage Program**		
Project ID:	**iManage Program STRIPES Project**		
Project Manager:	**Mathew Sparks**		
Program Mgr:	**Lajos Grof-Tisza**	**Doc ID:**	

Title Page

Document Name: Strategic Integrated Procurement Enterprise System (STRIPES) Technical Architecture Overview

Publication Date: July 2012

Document Author: Babu Ganesan / Michael Somerville

OCFO / CF-40

Program Name:	**Department of Energy iManage Program**		
Project ID:	**iManage Program STRIPES Project**		
Project Manager:	**Mathew Sparks**		
Program Mgr:	**Lajos Grof-Tisza**	**Doc ID:**	

Document information – Technical Architecture for STRIPES

Document source

This document is maintained as an online document. Contact the author for the latest version.

Revision history

Version	Date	Summary of changes	Revised By
1.0	7/25/2012	Initial Release	Michael Somerville

Approvals

The following people have approved this document. (Sign below name)

Name	Function
Lajos Grof-Tisza	Director, Corporate Information Systems
Signature:	Date:

Name	Function
Mathew Sparks	STRIPES Project Manager
Signature:	Date:

Program Name:	Department of Energy iManage Program		
Project ID:	iManage Program STRIPES Project		
Project Manager:	Mathew Sparks		
Program Mgr:	Lajos Grof-Tisza	Doc ID:	

1 Introduction

1.1 Background

The Integrated Management Navigation System (iManage) Program is the Department's solution for managing enterprise-wide systems initiatives to achieve improved financial and business efficiencies, integrated budget and performance, and expanded electronic government in support of the President's Management Agenda. The iManage Program is a collaborative Departmental effort to define and provide a modern, integrated corporate business system for the Department of Energy. The Project Portfolio is comprised of enterprise-wide systems initiatives to include: the Standard Accounting and Reporting System (STARS), Integrated Data Warehouse (IDW)/iPortal, iManage iBudget, Strategic Integrated Procurement Enterprise System (STRIPES), Corporate Human Resource Information System (CHRIS), and the E-Travel System (eTS).

The Strategic Integrated Procurement Enterprise System (STRIPES) is the procurement and contracts management component of the iManage program and encompasses both acquisition and financial assistance. STRIPES is based upon the Compusearch PRISM software. The STRIPES initiative is an important component of the overall vision of the iManage program. The STRIPES project reduces the number of procurement-related systems across the Department. This document provides a description of the technical architecture of STRIPES.

1.2 Purpose

The purpose of this document is to articulate the STRIPES Technical Architecture. This document provides an overview of the services, products, and infrastructure that will collectively define the technical architecture.

2 Major Architectural Elements

2.1 High-level Technical Architecture Model

The STRIPES utilizes the standard n-tier model used for most modern web-based applications. Contained within this n-tier model are several distinct layers which fulfill a distinct functional purpose. The n-tier application model relies heavily on functionally independent components working together in a loosely coupled manner.

The four layers contained within the n-tier architectural model are the:

- End User
- Presentation
- Business Integration
- Data Layer

Each layer is responsible for fulfilling distinct functions within the STRIPES architecture. Details about each of these layers follow.

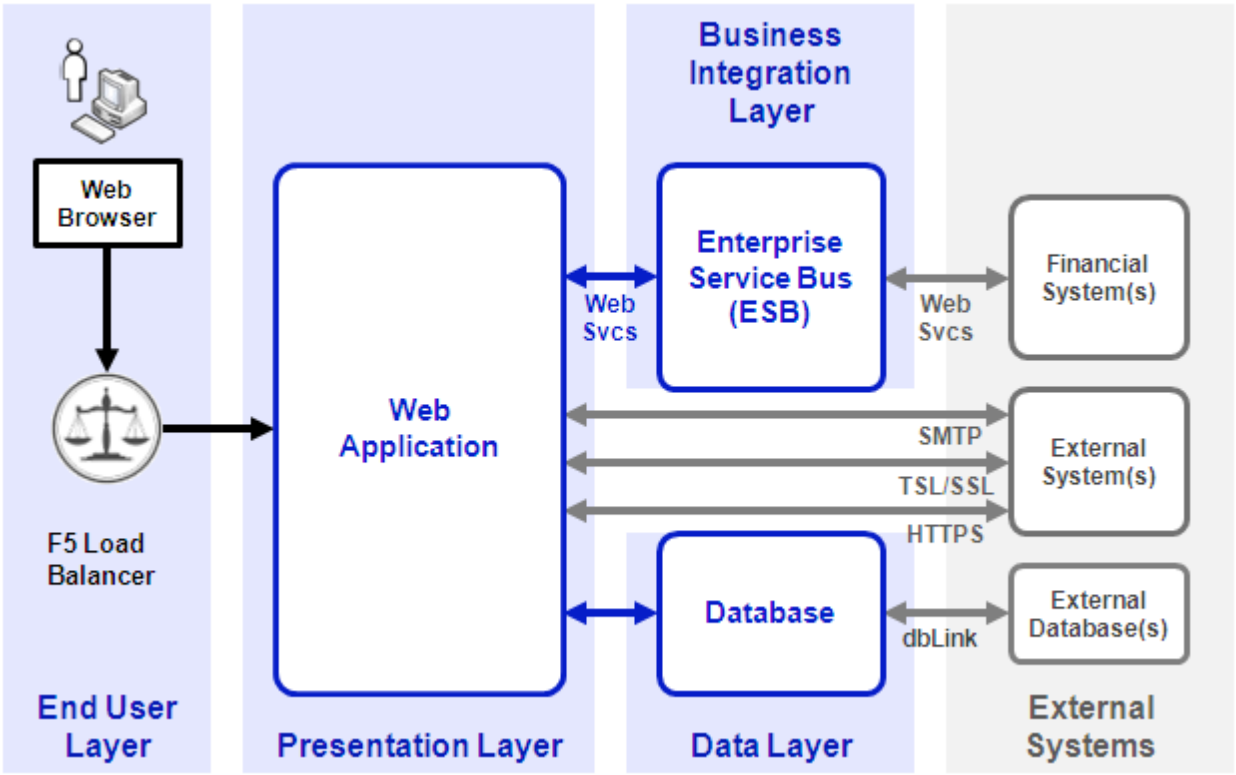

Figure 2.1-1 Generic STRIPES Architecture

2.2 End User Layer

The End User Layer is the user's web browser and any network infrastructure between the End User and the STRIPES server(s). This End User Layer is not provided directly by STRIPES and will not be described in detail.

It is important to note that web browser configuration will need to be actively managed to ensure that each site has the necessary configuration settings to properly run STRIPES. All of the sites will need to ensure that VBScript is enabled since much of the application utilizes client-side scripting to provide an interactive user interface. Additionally, all pop-up blockers will either need to be disabled or will need to add the STRIPES web site as an 'Allowed' site. It is also important to ensure that all cookies are accepted from the web server.

Note that even though VBScript is required, ActiveX, Java, or other third-party plug-ins do not need to be installed or enabled to use the system. However, some of the system utilizes Rich Text Format (RTF) and Adobe Portable Document Format (PDF) for the generation of certain document files.

In order for the end user to view these documents, a standard word processing application and Adobe Acrobat Reader v5.0 or above should be installed prior to using STRIPES. Agency-wide surveys have indicated that Adobe Acrobat Reader and Microsoft Word are standard applications available to all users, thus reducing concerns of their availability.

The End User Layer also enables reporting through integration with other parts of the overall iManage system. Specific STRIPES data is transferred to the iPortal/IDW tool. iPortal/IDW provides ad-hoc reporting capabilities through the Business Intelligence (BI) tool, in addition to providing standard reports that can be invoked by users.

Shared network infrastructure includes the F5 Networks Load Balancer. The Office of the Chief Information Officer Application Hosting Environment, (AHE) uses F5 Networks load balancing content aware switches to do active-active server configurations and for balancing workloads dynamically across multiple systems. The F5 Networks load balancer can also handle SSL encryption, off loading that workload from the server. The load balancing will take incoming SSL encrypted requests from user clients, will decrypt the SSL encryption, and using an algorithm that tracks current usage balance, it will forward the cleared request to the least used web server. The load balancer will make sure and route all requests from a current session to the same web server, to maintain the 'state' of transactions and workflow from the user's perspective.

STRIPES will be protected by the AHE enterprise firewalls which are based on Cisco PIX firewall technology. These firewalls are maintained by the AHE networking group. The ports and protocols that STRIPES plans on using for both user interface and systems interface are registered with the firewall services. All of the external systems interface connections will be initiated by PRISM, so all traffic to PRISM servers from external sites will be blocked, except any users coming in via approved DOE VPN tunnels.

The deployment of the STRIPES system will deal with site firewalls that are not enterprise managed. Generally it will be a simple matter of registering the correct ports (80,443) and endpoint servers with each site's firewall support group, and ensuring through deployment testing that the traffic is allowed.

The End User Layer is a key part of the overall STRIPES Technical Environment, but falls outside of the direct control of the STRIPES team.

2.3 Presentation Layer

The Presentation Layer provides the user interface as a web application. The Presentation Layer is the interface through which users manage documents throughout the process.

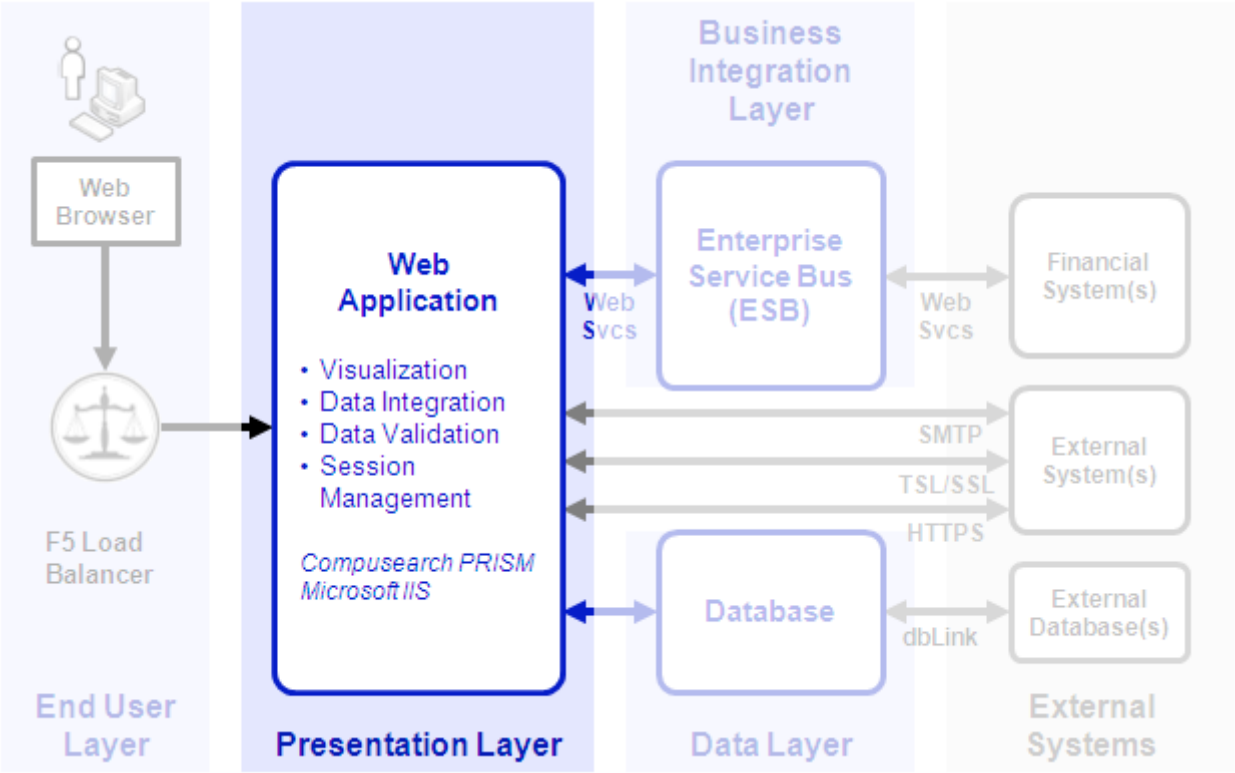

Figure 2.3-1 STRIPES Architecture – Presentation Layer

2.3.1 Presentation Layer Services

Visualization: The Presentation Layer displays the user interface. Navigation menus, input forms, lists, pages and messages are examples of the objects managed in this tier.

Data Integration: The Presentation Layer is responsible for mapping the output/results of the Business Integration Layer and the Data Layer into something that is usable by the end user.

Data Validation: The Presentation Layer manages the bulk of the application logic. This is where input validation, business rules, workflow and data manipulation occur.

Session Management: The Presentation Layer manages the user session from authentication through termination of a particular session.

2.3.2 Presentation Layer COTS Products

The Presentation Layer builds on the following Commercial Off-The-Shelf (COTS) products.

2.3.2.1 Compusearch PRISM

PRISM, the COTS procurement package from Compusearch, is configured using built-in tailoring mechanisms to meet many of the OCFO functional requirements. It is configured to meet agency-specific requirements for integration with the agency's financial system, STARS.

Since PRISM is a web-based application, the primary end-user interface is the common web browser. A web browser is the only application required to be installed on the client machines in order to run the PRISM software. All of the other layers within the PRISM software architecture model are located at the server level.

The physical layout of the PRISM software will reside on multiple virtual servers. Distinct components of the physical architecture will reside on separate servers in order to increase performance, stability, scalability, and maintainability. The presentation layer uses Microsoft Windows Server 2003 Enterprise Edition (32bit). Windows Server 2003 is one of the standard server operating systems currently utilized by the DOE AHE.

The PRISM software allows agency specific customization through the large number of configurable settings extended to the system administrator. Most system configuration will be achieved through the system administrator screens contained within the PRISM user interface. The other means of configuring the PRISM software are listed in the table below**Error! Reference source not found.**. This table describes the extensibility mechanisms available to the system administrator. This table also provides a brief description of these mechanisms and their general usage.

Table - PRISM Extensibility Mechanisms

Mechanism	Description
User Views	A set of database views that extract information from various internal tables that underlie the PRISM software, (e.g., that store requisition information) and make them easier to decipher. Some views also reflect business rules and attributes such that analysts and report designers do not need to know how various flags or statuses are set. For example, one view shows only those requests that have yet to be awarded.
Application Programmer Interface	A set of software routines, written in PL/SQL but accessible through web services. These APIs act as an alternative to user entry and allow for the insertion or extraction of data to and from the PRISM application. Their capabilities include the creation of vendor records, awards, users, etc., in PRISM, and extract data from external systems like STARS.
Web Services Interface	A service that allows offering of web services as defined by the W3C group

2.3.2.2 Microsoft IIS

STRIPES requires a web application container that can run ASP code. The PRISM software was written using Microsoft's ASP and .NET platform which can only run using Microsoft's Internet

Information Server (IIS) on the Windows Platform. IIS is a standard component within Microsoft Windows Server 2003, provided by AHE.

2.4 Business Integration Layer

The Business Integration Layer provides the interface to financial systems using the Enterprise Service Bus (ESB). The ESB components handle incoming requests from STRIPES and perform required transformations and interaction with the external components.

In the interest of understanding the bigger system environment in which the iManage application portfolio currently exists and how changes will occur to both the iManage portfolio and its environment, the STRIPES project chose to align its decision making with the Federal Enterprise Architecture (FEA) Service Component Reference Model.

After examining the FEA Service Component Reference Model, it appears that the iManage application portfolio spans across multiple business areas and domains, and this portfolio represents a small yet significant set of services that are part of a bigger portfolio of enterprise level services that will eventually grow as DOE will use information technology to implement the additional services listed in the FEA Service Component Reference Model.

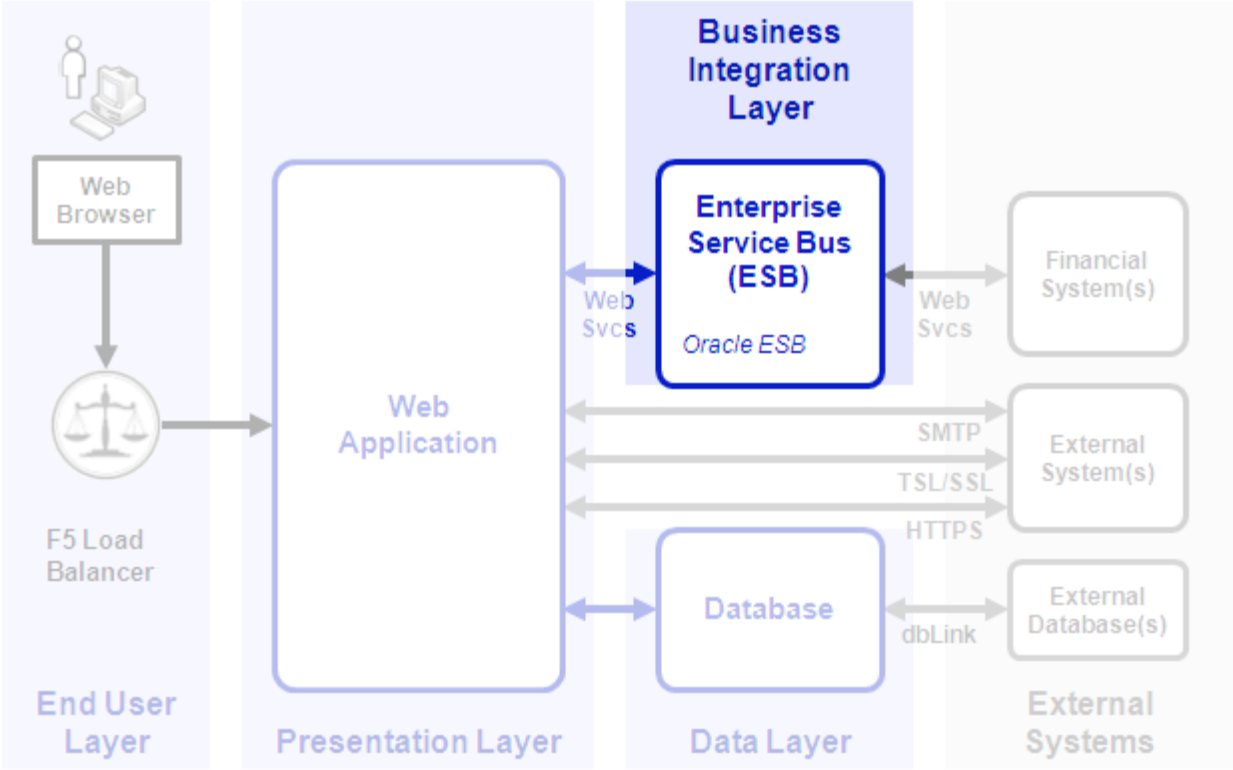

Figure 2.4-1 STRIPES Architecture – Business Integration Layer

2.4.1 Business Integration Layer Services

The Business Integration Layer provides the services that enable STRIPES to efficiently and securely communicate with external financial systems (STARS). The following transactions are supported between STRIPES and STARS.

Table – Transaction Details

Transaction	Invocation Flow	Synchronicity (Synchronous / Asynchronous)	STARS Interface	STRIPES Interface
Validate PR	Outbound	Synchronous	Validate PR	n/a
Commit Funds De-commit Funds	Outbound	Asynchronous	Commitment	n/a
Validate PO	Outbound	Synchronous	Validate PO	n/a
Obligate Funds De-obligate Funds	Outbound	Asynchronous	Obligation	n/a
Release PR	Inbound	Batch	n/a	Approve Req.
Release PO	Inbound	Batch	n/a	Approve Award
Sync AFF Segments (1-4,6-10)	Inbound	Batch	n/a	Create Segment
Sync Approver Data	Inbound	Batch	n/a	Create Header Level Field

Table – Transaction / Messaging Style Matrix

Interaction	Interface	Messaging Style
STRIPES to STARS	Accounting (Validation, Commitment, Obligation)	Request / Response
STARS to STRIPES	Approval / Data Updates	Send-and-Forget (1-to-1)

Figure 2.4.1-1 shows the integration view between STRIPES and STARS. All synchronous services implemented by STARS are exposed using existing technologies due to potential constraints of not being able to support Web Services given its existing runtime platform, but then these services are further exposed using Web Services via the Enterprise Service Bus. The exposure of these services using open standards will enable faster coordination and communication of integration semantics across application groups, and easily allow those services to be reused more readily with little coordination effort.

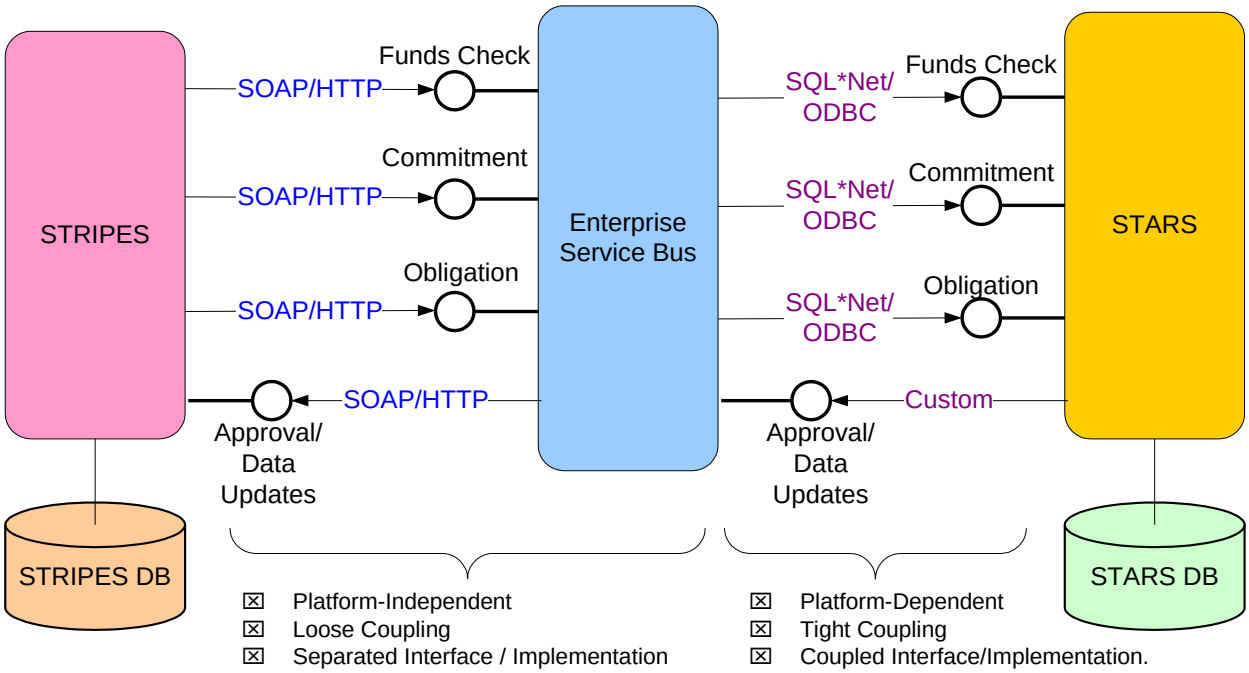

Figure 2.4.1-1 STRIPES/STARS Physical Connectivity using an ESB

2.4.2 Business Integration Layer COTS Products

2.4.2.1 Oracle Enterprise Service Bus (ESB)

The middleware component of STRIPES/STARS integration is developed using Oracle Enterprise Service Bus architecture (ESB). While the Oracle ESB product was used for the STRIPES implementation, the BPEL Process Manager was not used.

The iManage program had already purchased licenses for the Oracle Application Server 10g Enterprise Edition, which included the cost for the Oracle ESB product. Installing the SOA Suite provided the project with the ESB product.

2.4.3 Business Integration Layer Data Stores

ESB Data Store – contains some metadata storage for web service registration.

2.5 Data Layer

The Data Layer provides the authoritative system of records for STRIPES. This STRIPES database will have the ability to store both structured and unstructured data. This section will provide an overview of the two types of data and their usage.

Structured Data: The STRIPES database has several types of structured data which primarily consists of PRISM configuration, reference, and transaction data. Most structure data fields are well defined by PRISM for the purpose of either controlling application functionality or for transaction purposes. The only structured data that is not defined are known as "Flex Fields" which are intended to fulfill agency specific requirements. These fields are undefined in both data type and purpose. All structured data will be maintained in Oracle databases and modified through the STRIPES user interface.

PRISM Implementation of DOE Definable (Flex) Fields: The PRISM software provides 110 undefined fields (40 at the document header, 40 at the line item, and 30 at the funding level) which are available to DOE acquisitions for its own purposes. These fields will need to be defined, labeled, and enabled within PRISM to extend these fields to the end-user. These configuration requirements can be fulfilled once the configuration team has decided how these fields are best allocated within DOE. All field labeling, application of limited business logic, and/or multiple-choice value definitions applicable to each fields are defined within the system administrator's configuration screens. The configuration of these flex fields does not require custom programming or modification to the default PRISM database. Despite the fact that these fields are not predefined, they are completely accessible for reporting or automated data extraction/insertion through system interfaces.

Unstructured Data: In addition to the structured data, PRISM has the ability to store unstructured data. Unstructured data can include any file uploaded using the STRIPES user interface. These files can include statements of work (SOW) in MS Word or PDF formats, budgetary spreadsheets in Excel, charts or diagrams created in PowerPoint or Visio, or any other file type accessible from the end user's machine. These files are stored by saving the binary data directly within the same Oracle database as all of the structured data described previously.

Some of the reference data that needs to be available as a pre-condition for STRIPES processes are as follows:

1. Vendors & Vendor Sites
2. Invoice Approvers
3. Account Codes [Organizations derived from here]

STARS will be the system of record for all the reference data mentioned above. Within STRIPES, this reference data will be kept read-only outside of the system of record.

The STRIPES database is a transactional system. The IDW will serve as the reporting database for all ad-hoc reporting. Raw data will be collected by the IDW and all aggregation and transformation of the raw data will take place within IDW.

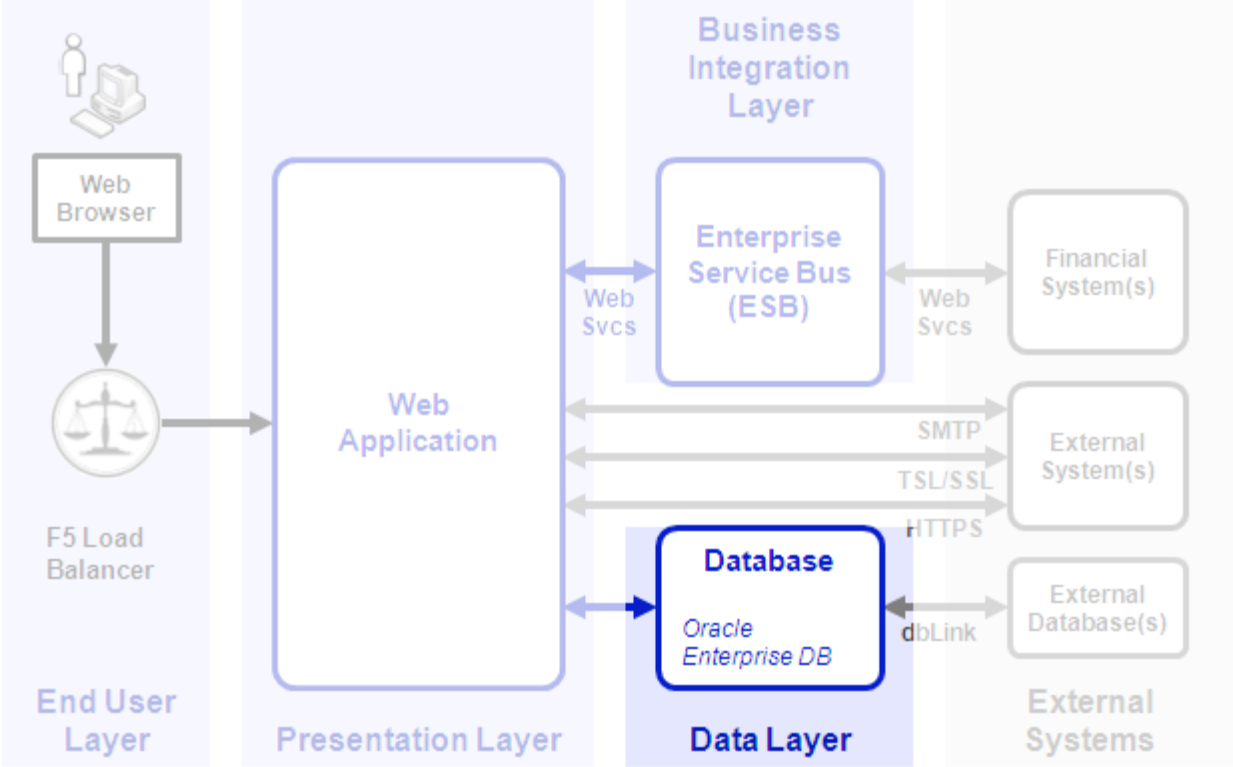

Figure 2.5-1 STRIPES Architecture – Business Integration Layer

2.5.1 Data Layer COTS Products

STRIPES uses Oracle's Database Server to fulfill the data layer architecture. Since Oracle is currently deployed within the DOE AHE, STRIPES benefits from decreased costs and maintenance issues as well as increased reliability.

The current version of Oracle used by the AHE is Oracle Database Server 10g R3.

2.5.2 Data Layer Data Stores

Each instance of the PRISM software requires a single Oracle database capable of storing both structured and binary data. Each environment configuration includes a UNIX database server. The four database instances are named as follows:

- STRPSP – Production
- STRPSQA – QA/Test
- STRPSTR – Training
- STRPSD – Development

3 High Level STRIPES Architecture

3.1 Overview of Architectural Viewpoints

This document describes the STRIPES architecture from several different technical viewpoints. Each emphasizes a different, important aspect of STRIPES. These viewpoints are:

- Basic STRIPES Architecture
- STRIPES Interfaces
- STRIPES Environments
- Key Technical Detail Diagram(s)

3.2 Basic STRIPES Architecture

STRIPES instances follow the same basic pattern. In each environment, shared network equipment balances the requests to the STRIPES web application server(s). STRIPES scales the number of application servers to meet the requirements of each environment.

The web application server(s) run on Windows Server 2003 installed on virtual machines. Physical resources (e.g. CPU cores, RAM, local disk space) are allocated according to the requirements of each environment.

Each environment has an instance of the Enterprise Service Bus (ESB) and Database.

Each environment has interfaces to External Systems. The specific instances of External Systems vary, depending on the environment. This ensures proper handling of data between systems.

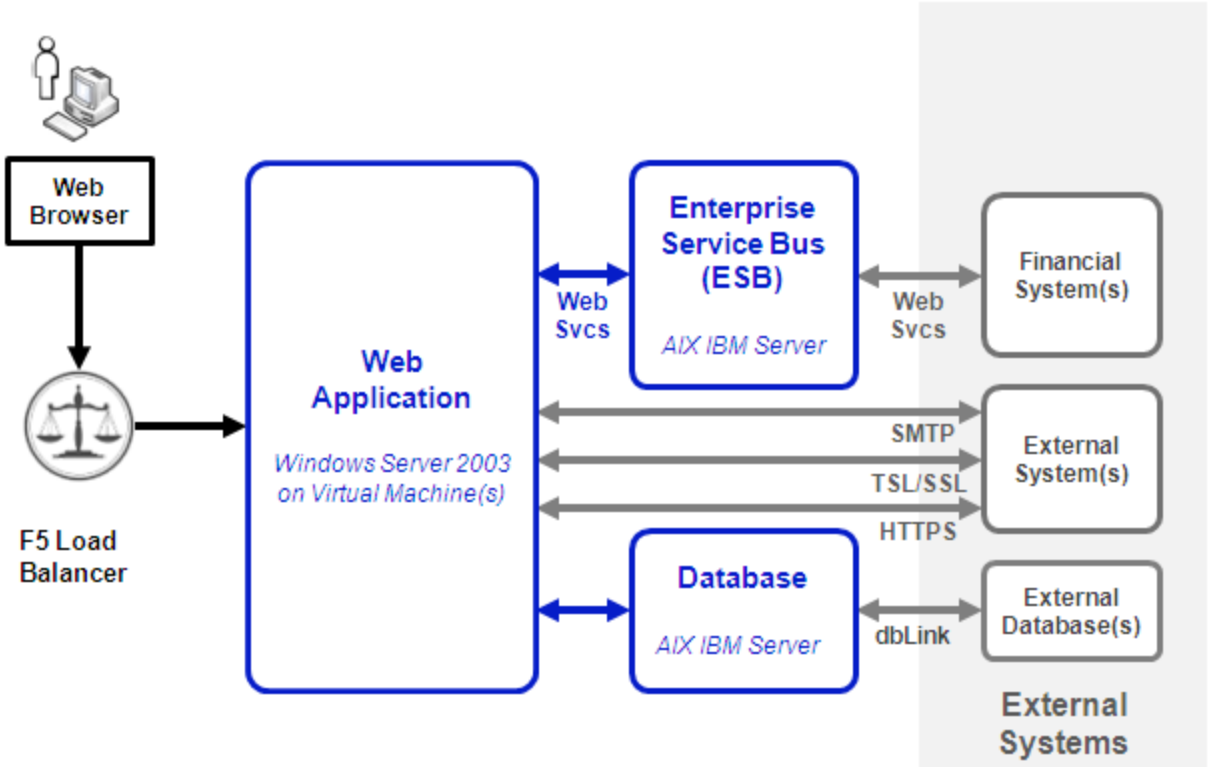

Figure 3.2-1 STRIPES Instance Pattern

3.3 STRIPES Interfaces

The STRIPES application is integrated with several internal DOE systems as well as external federal systems, as shown in the figure below.

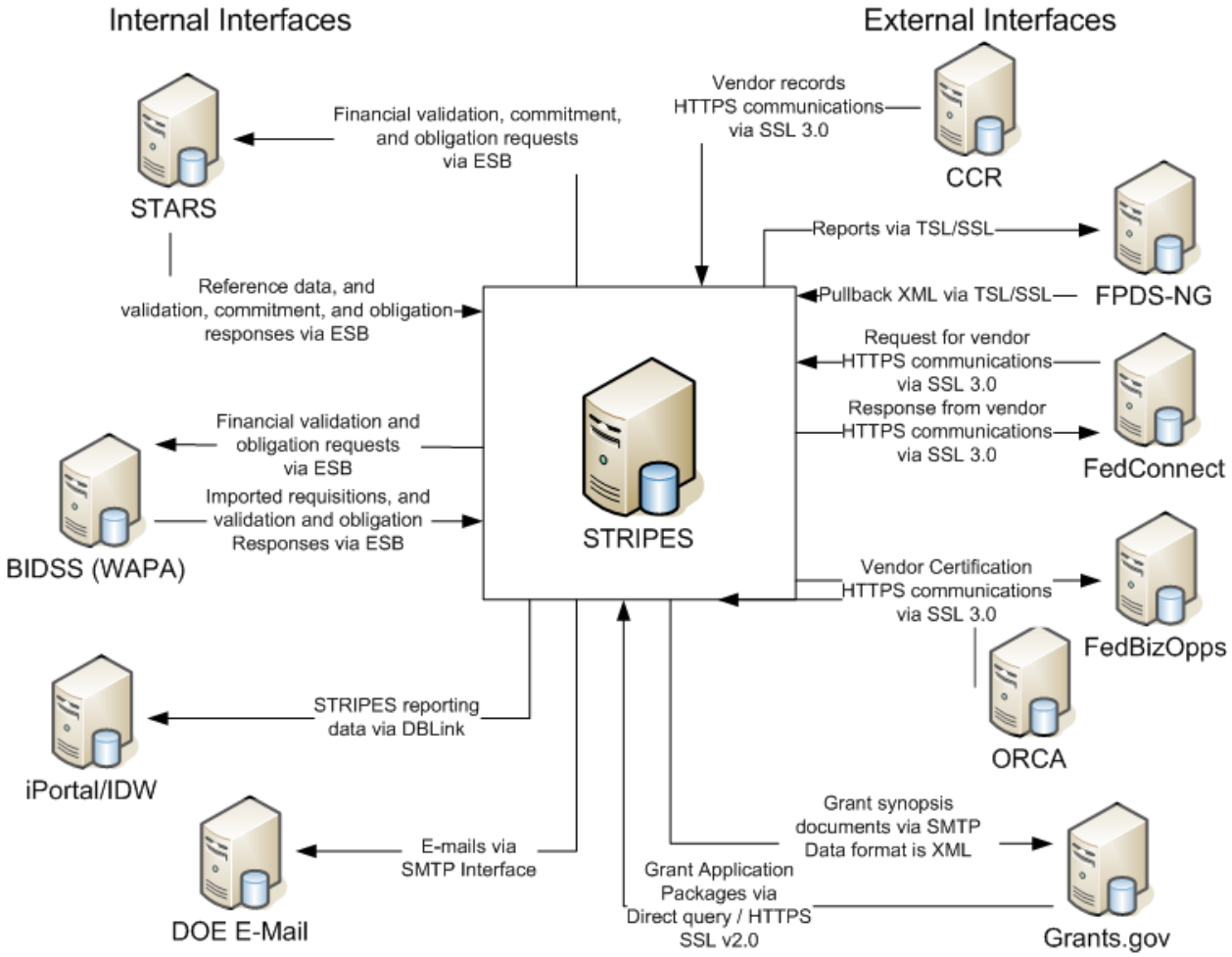

Figure 332-1 STRIPES Interfaces

STRIPES will primarily invoke interfaces on the external federal systems to send or retrieve information. STRIPES will invoke and provide interfaces for integration with internal systems such as STARS.

Table – STRIPES Interfaces

Type	From	Data & Technology	To
External	CCR	Vendor records HTTPS communications via SSL 3.0	STRIPES
External	STRIPES	Reports via TSL/SSL	FPDS-NG
External	FPDS-NG	Pullback XML via TSL/SSL	STRIPES

External	FedConnect	Request for vendor HTTPS communications via SSL 3.0	STRIPES
External	STRIPES	Response from vendor HTTPS communications via SSL 3.0	FedConnect
External	STRIPES	Synopsis via SMTP Data format is XML	FedBizOpps
External	ORCA	Vendor Certification HTTPS communications via SSL 3.0	STRIPES
External	Grants.gov	Grant synopsis documents via SMTP Data format is XML	STRIPES
External	STRIPES	Grant Application Packages via Direct query / HTTPS SSL v2.0	Grants.gov
Internal	STRIPES	E-mails via SMTP Interface	DOE E-Mail
Internal	STRIPES	STRIPES reporting data via DBLink	iPortal/IDW
Internal	BIDSS (WAPA)	Imported requisitions, and validation and obligation Responses via ESB	STRIPES
Internal	STRIPES	Financial validation and obligation requests via ESB	BIDSS (WAPA)
Internal	STARS	Reference data, and validation, commitment, and obligation responses via ESB	STRIPES
Internal	STRIPES	Financial validation, commitment, and obligation requests via ESB	STARS

3.4 STRIPES Environments

Distinct environments are used to isolate systems serving production needs from other server support (Training, Development, QA/Test) functions.

STRIPES functions can be extended through a Change Management process. The Change Management process requires changes to be tested outside of the Production environment prior to promotion to operational use. To support this process, STRIPES is instantiated in four different environments – Production, Quality Assurance, Training, and Development.

Table – Overview of STRIPES Environments

Logical Environment	Description
Development	Where developers build extensions and do component testing
QA/Test	Used to support regression and system testing, as well as staging, where upgrade procedures can be built and tested.
Sandbox/Training	Used for training activities
Production	The Production environment is where real work is performed by all of the end users.

3.4.1 Production

The Production instance supports operational users. This environment has:

- 6 IIS Virtual Windows Server nodes
- 1 ESB node
- 1 DB node

3.4.1.1 Production System Diagram

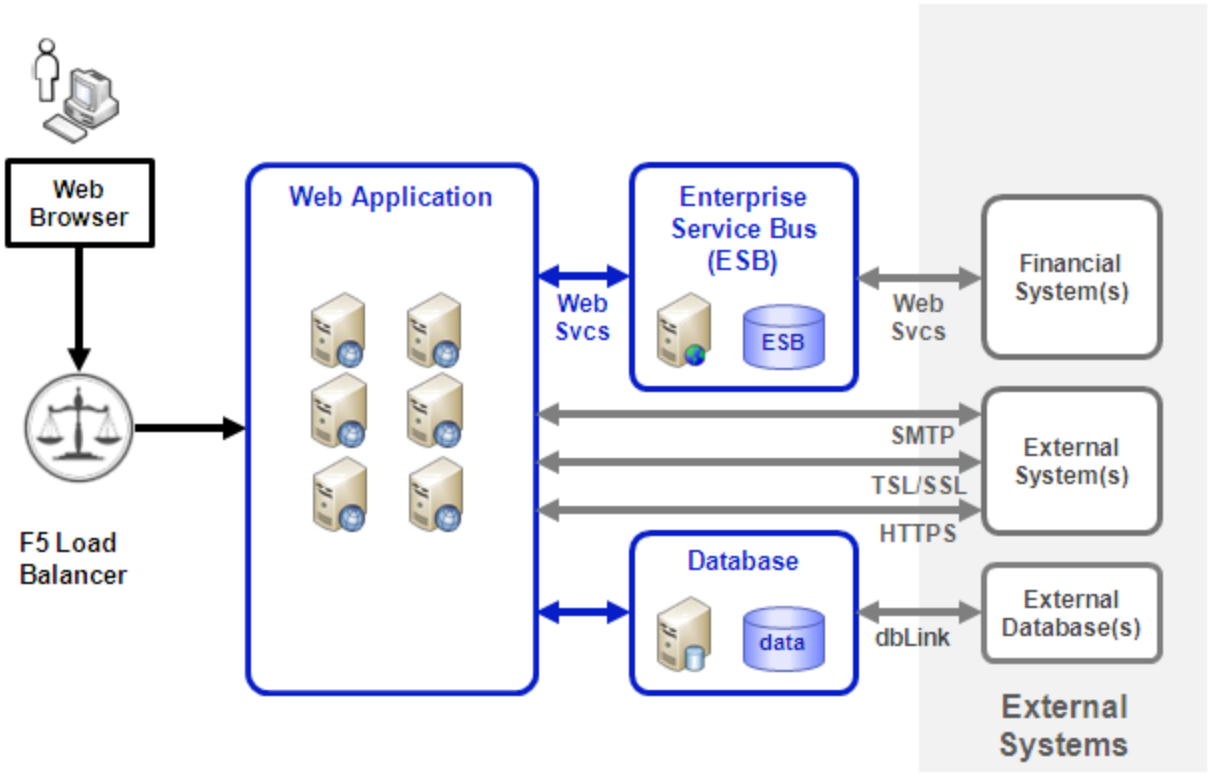

Figure 3.4.1.1-1 STRIPES Production Instance

3.4.1.2 Physical Configuration

Table – STRIPES Production Physical Configuration

Production Server Information	Web Server	ESB Server	Database
Hardware	Virtual Machine	IBM, 9116-561	IBM, 9116-561
OS	Windows 2003 Server	AIX 5.3.0.0 5300-11 (64-Bit)	AIX 5.3.0.0 5300-11 (64-Bit)
CPUs	4 (Intel Xeon X5670 @2.93 GHz)	6	8
Memory	4 GB	16 GB	32 GB
Number of Virtual Machines	6	NA	NA

3.4.2 Quality Assurance (QA)

The Quality Assurance instance ensures that changes are tested in a Production-like environment prior to operational deployment. This environment has:

- 2 IIS Virtual Windows Server nodes
- 1 ESB node
- 1 DB node

3.4.2.1 Quality Assurance (QA) System Diagram

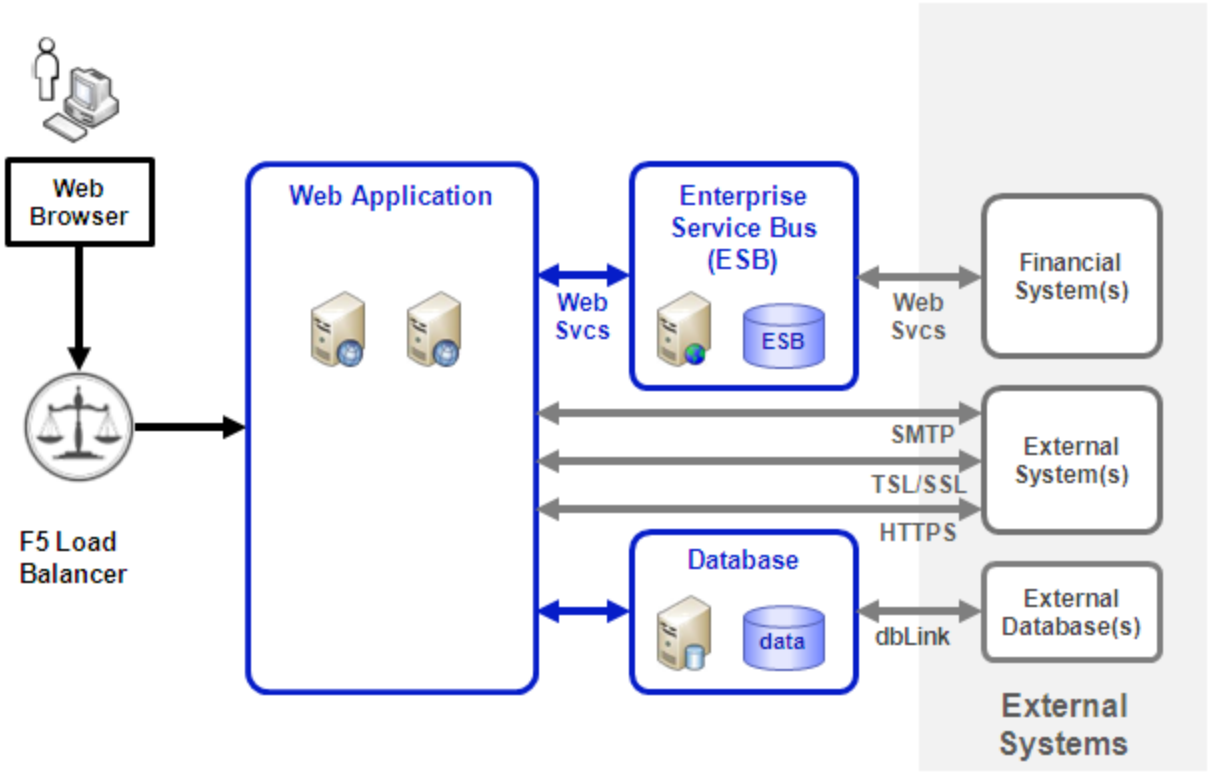

Figure 3.4.2.1-1 STRIPES Quality Assurance (QA) Instance

3.4.2.2 Physical Configuration

Table – STRIPES Quality Assurance (QA) Physical Configuration

Quality Assurance (QA) Server Information	Web Server	ESB Server	Database
Hardware	Virtual Machine	IBM, 9116-561	IBM, 9116-561
OS	Windows 2003 Server	AIX 5.3.0.0 5300-11 (64-Bit)	AIX 5.3.0.0 5300-11 (64-Bit)
CPUs	4 (Intel Xeon X5670 @2.93 GHz)	8	8
Memory	4 GB	16 GB	15 GB

Quality Assurance (QA) Server Information	Web Server	ESB Server	Database
Number of Virtual Machines	2	NA	NA

3.4.3 Training

The Training environment provides an instance where users can create, modify, and delete data without concern for production impact. This environment has:

- 1 IIS Virtual Windows Server node
- 1 ESB node
- 1 DB node

3.4.3.1 Training System Diagram

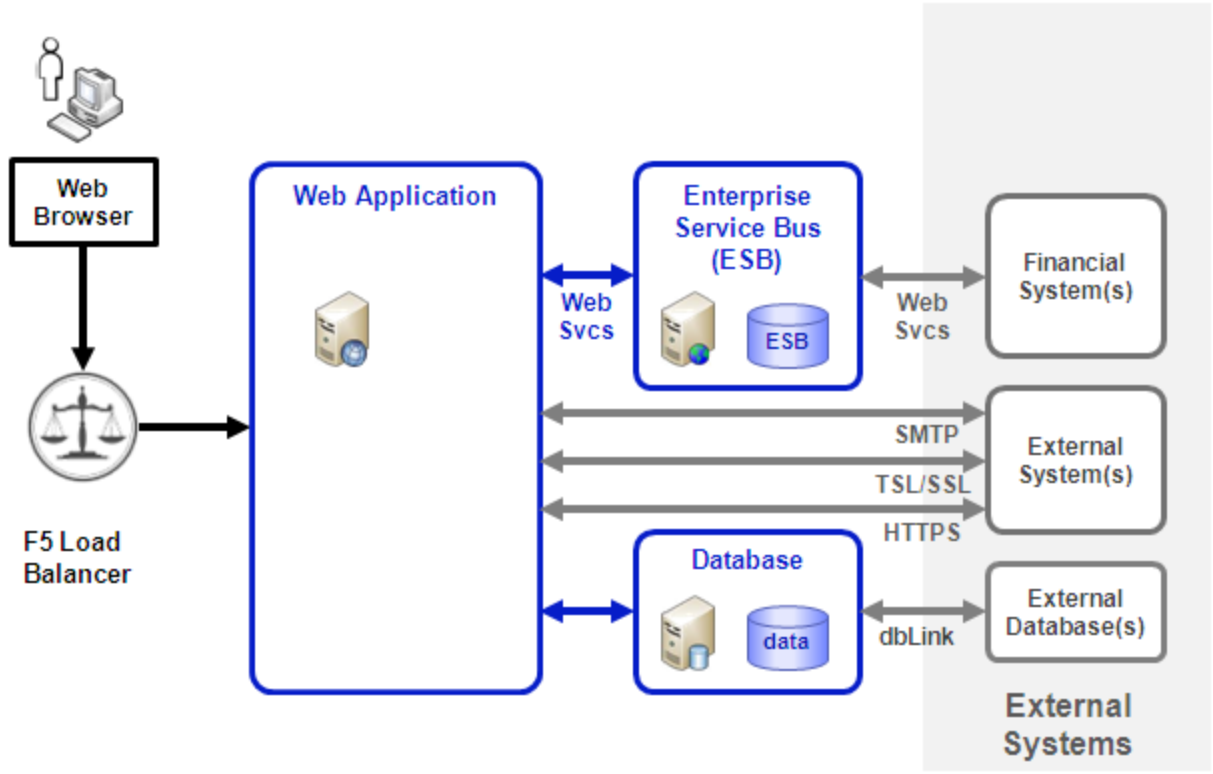

Figure 3.4.3.1-1 STRIPES Training Instance

3.4.3.2 Physical Configuration

Table – STRIPES Training Physical Configuration

Training Server Information	Web Server	ESB Server	Database
Hardware	Virtual Machine	IBM, 9116-561	IBM, 9116-561
OS	Windows 2003 Server	AIX 5.3.0.0 5300-11 (64-Bit)	AIX 5.3.0.0 5300-11 (64-Bit)
CPUs	4 (Intel Xeon X5670 @2.93 GHz)	8	8
Memory	4 GB	21 GB (Shared between 2 Instances)	11 GB

Training Server Information	Web Server	ESB Server	Database
Number of Virtual Machines	1	NA	NA

3.4.4 Development

The Development environment enables STRIPES developers to test patches, custom scripts, and other changes against the same software as found in Production. This environment has:

- 1 IIS Virtual Windows Server node
- 1 ESB node
- 1 DB node

3.4.4.1 Development System Diagram

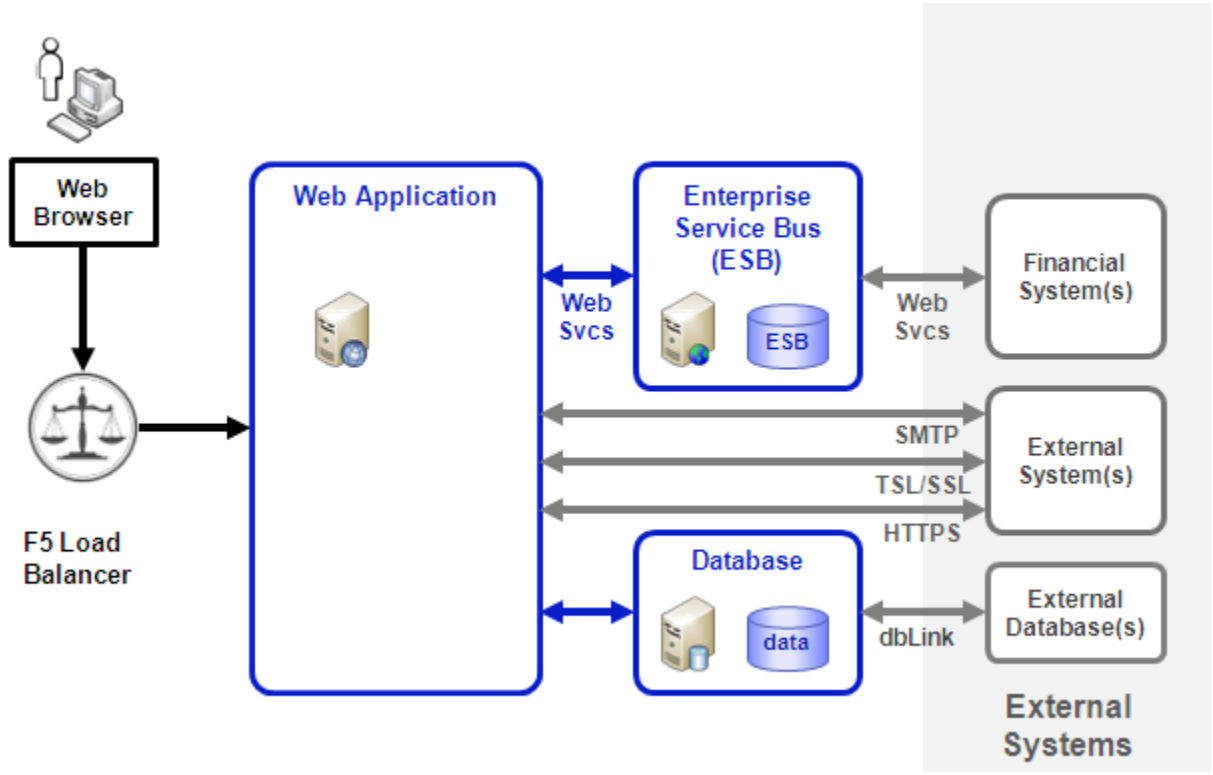

Figure 3.4.4.1-1 STRIPES Development Instance

3.4.4.2 Physical Configuration

Table – STRIPES Development Physical Configuration

Development Server Information	Web Server	ESB Server	Database
Hardware	Virtual Machine	IBM, 9116-561	IBM, 9116-561
OS	Windows 2003 Server	AIX 5.3.0.0 5300-11 (64-Bit)	AIX 5.3.0.0 5300-11

			(64-Bit)
CPUs	4 (Intel Xeon X5670 @2.93 GHz)	8	8
Memory	4 GB	21 GB (Shared between 2 Instances)	15 GB
Number of Virtual Machines	1	NA	NA

3.5 Key Technical Detail Diagram(s)

The following sections describe key technical details that require additional clarification.

3.5.1 Enterprise Service Bus Message Exchanges

The middleware component of STRIPES/STARS integration is developed using Oracle Enterprise Service Bus architecture (ESB). The ESB components will handle incoming requests from STRIPES, perform required transformations and interaction with the external components.

Following figures represent the process flows between ESB and STRIPES:

Commitment

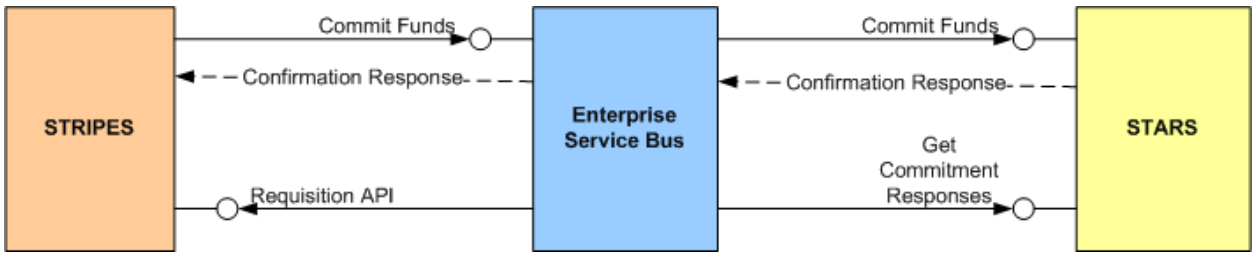

Figure 2.2-2. Commitment Process from/to Financial System

Obligation

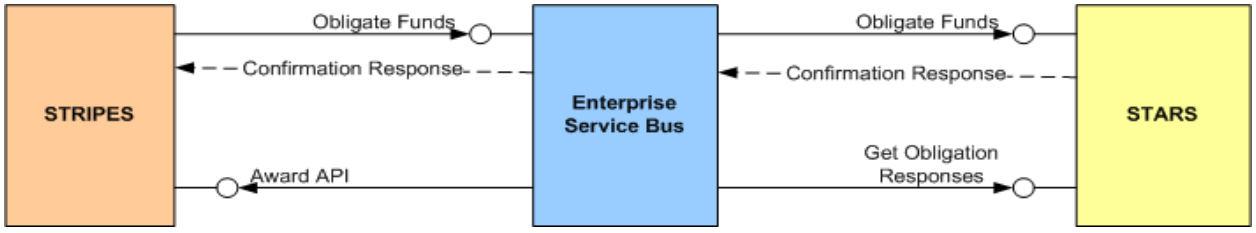

Figure 2.2-3. Obligation Process from/to Financial System

Import Requisitions into STRIPES

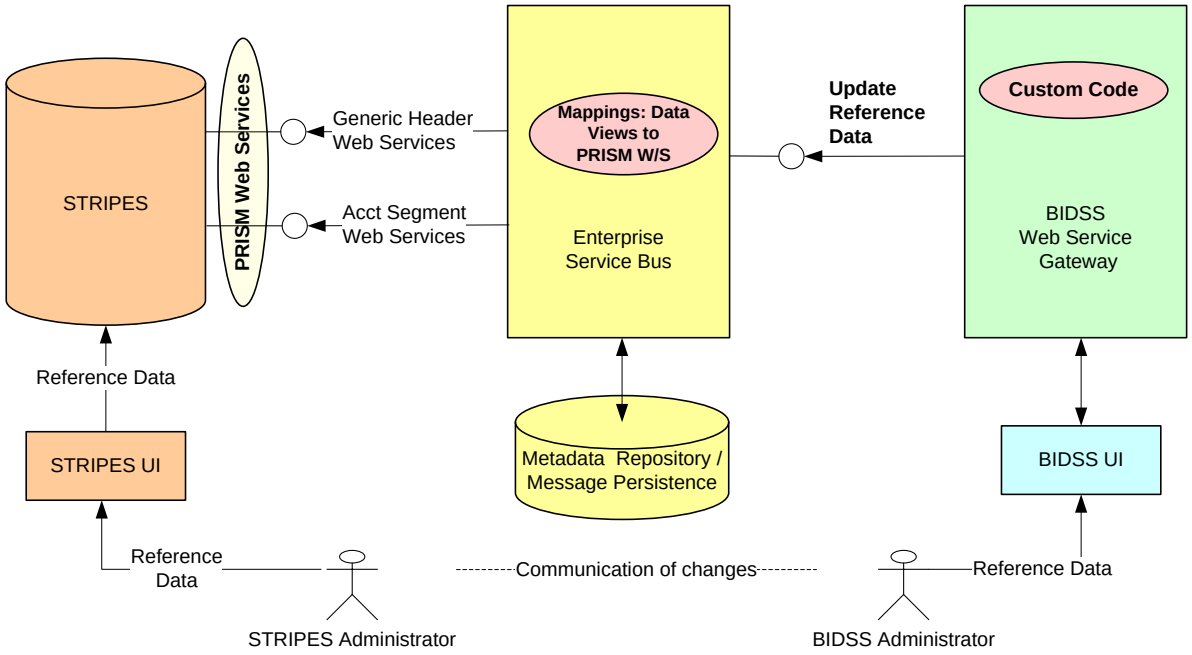

Figure 2.2-4. Requisition Import Process from Financial System