NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

EVALUATION OF ENTERPRISE APPLICATION INTEGRATION (EAI) AND WEB SERVICES AT FITTING OUT AND SUPPLY SUPPORT ASSISTANCE CENTER (FOSSAC) UNDER NMCI

by

Jeffrey L. Lark
Mark A. Reyes

September 2002

Thesis Advisor:	Glenn R. Cook
Co-Advisor:	Rex A. Buddenburg

Approved for public release; distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** September 2002	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE: **Evaluation of Enterprise Application Integration (EAI) and Web Services at Fitting Out and Supply Support Assistance Center (FOSSAC) Under NMCI**

5. FUNDING NUMBERS

6. AUTHOR(S)
Lark, Jeffrey L., Reyes, Mark A.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT

Information technology has woven itself into the fabric of every organization. As organizations grow and develop specialized needs, specialized software applications emerge to address the needs. Often the business processes take shape around the capabilities of the software applications and the technology infrastructure, until the two are inseparable from one another. When an organization decides to incorporate new processes or upgrade its information architecture, the new system may lack compatibility with the old system. The old, incompatible software is typically referred to as a "legacy application". In an effort to integrate the old applications with the new, organizations are typically faced with expensive, proprietary Enterprise Application Integration solutions. Fitting Out and Supply Support Assistance Center (FOSSAC) is an organization facing a legacy application integration challenge with the implementation of the Navy-Marine Corps Intranet.

This thesis examines the applicability of traditional Enterprise Application Integration (EAI) methodologies for FOSSAC as way to preserve access to its legacy applications. As an alternative integration solution, this thesis explores the potential of the emerging Web Services architecture. The Web Services architecture employs standard Internet protocols to facilitate application integration and information sharing across a variety of computing-platforms.

14. SUBJECT TERMS Enterprise Application Integration (EAI), Web Services, Middleware, Legacy Applications, Navy -Marine Corps Intranet (NMCI)

15. NUMBER OF PAGES
97

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

THIS PAGE INTENTIONALLY LEFT BLANK

EVALUATION OF ENTERPRISE APPLICATION INTEGRATION (EAI) AND WEB SERVICES AT FITTING OUT AND SUPPLY SUPPORT ASSISTANCE CENTER (FOSSAC) UNDER NMCI

Jeffrey W. Lark
Major, United States Marine Corps
B.S., University of Iowa, 1987

Mark A. Reyes
Lieutenant, United States Navy
B.S., Texas Tech University, 1995

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS TECHNOLOGY

from the

NAVAL POSTGRADUATE SCHOOL
September 2002

Authors: Jeffrey W. Lark

Mark A. Reyes

Approved by:

Glenn R. Cook, Advisor

Rex A. Buddenburg, Co-Advisor

Dan C. Boger, Chairman
Information Sciences Department

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Information technology has woven itself into the fabric of every organization. As organizations grow and develop specialized needs, specialized software applications emerge to address the needs. Often the business processes take shape around the capabilities of the software applications and the technology infrastructure, until the two are inseparable from one another. When an organization decides to incorporate new processes or upgrade its information architecture, the new system may lack compatibility with the old system. The old, incompatible software is typically referred to as a "legacy application". In an effort to integrate the old applications with the new, organizations are typically faced with expensive, proprietary Enterprise Application Integration solutions. Fitting Out and Supply Support Assistance Center (FOSSAC) is an organization facing a legacy application integration challenge with the implementation of the Navy-Marine Corps Intranet.

This thesis examines the applicability of traditional Enterprise Application Integration (EAI) methodologies for FOSSAC as way to preserve access to its legacy applications. As an alternative integration solution, this thesis explores the potential of the emerging Web Services architecture. The Web Services architecture employs standard Internet protocols to facilitate application integration and information sharing across a variety of computing-platforms.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND TERMS

A2A	Application-to-Application
ACID	Atomicity, Consistency, Isolation, and Durability
API	Application Programming Interfaces
B2B	Business-to-Business
B2C	Business-to-Commerce
COTS	Commercial-Off-The-Shelf
CRM	Customer Relationship Management
DoD	Department of Defense
DOT	Distributed Object Technology
DWCF	Defense Working Capital Fund
EAI	Enterprise Architecture Integration
EDS	Electronic Data Systems Corp.
ERP	Enterprise Resource Planning
FISC	Fleet Industrial Supply Centers
FOSSAC	Fitting Out and Supply Support Assistance Center
GOTS	Government-Off-The-Shelf
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IT	Information Technology
LAN	Local Area Network
MCEN	Marine Corps Enterprise Network
MOM	Message-Oriented Middleware
NAVSUP	Navy Supply System Command
NMCI	Navy-Marine Corps Intranet
RPC	Remote Procedure Call
SCM	Supply Chain Management
SIMA	Shore Intermediate Maintenance Activities
SOAP	Simple Object Access Protocol
TPM	Transaction Processing Monitor

TCP/IP	Transport control Protocol/Internet Protocol
UDDI	Universal Description, Discovery, and Integration
WSDL	Web Services Definition Language
WWW	World Wide Web
XML	eXtendible Markup Language
Y2K	Year 2000

I. INTRODUCTION

A. PURPOSE

The purpose of this thesis is to analyze the information technology architecture currently in place at the Fitting Out and Supply Support Assistance Center (FOSSAC), aboard Norfolk Naval Base, Norfolk, VA. FOSSAC is a field activity of the Naval Supply Systems Command (NAVSUP), and as such, it is in the process of migrating to a new network architecture as dictated by the Navy-Maine Corps Intranet (NMCI). This difference between the current network and future network is different enough to pose some significant challenges for continuing with current business processes.

The current "as-is" environment will soon be transformed in accordance with the 5-year $6.9 billion NMCI contract with Electronic Data Systems (EDS). The EDS contract will transform commands throughout the Navy and Marine Corps. The objective of this thesis is to provide the rationale for choosing an Information Technology strategy for FOSSAC to maintain its leadership position. This analysis will provide a recommendation for a strategic business solution, incorporating approaches to improve key processes (i.e.; supply chain, front/back office integration, demand chain) and data integration during the Navy's outsourcing period.

B. BACKGROUND

The mission at FOSSAC is to provide Department of Defense (DoD) and federal agencies the best value in global logistics, engineering and support service solutions. FOSSAC was originally established as Fitting Out Supply Assistance Teams, Atlantic and Pacific (FOSATLANT and FOSATPAC) in September 1966. These early organizations assisted pre-commissioning crews in establishing the supply department of newly constructed ships. In June of 1972, FOSATPAC was disestablished and FOSATLANT became FOSAT, responsible for providing services nationwide. In 1983, as demand for services continued to grow, FOSSAC was established with FOSAT reorganized as subordinate unit. FOSSAC as it exists today, provides three major areas of service support for the fleet: Fitting Out Supply Assistance Team (FOSAT) continues to provide engineering and supply support services for outfitting new ships. Inter-Service Supply Support Operations Program (ISSOP) provides contractual inventory management and logistics support services to Department of Defense (DoD) customers. The Price Fighters Program provides unbiased expertise in value analysis and price validation services.

FOSSAC currently employees some 200 military and civilian personnel along with over 2,100 contractor personnel, provide supply-related engineering, training, and support services to both Fleet units and Navy shore activities worldwide. To manage the internal administration of FOSSAC, each department has military department head and an equal ranking Government Schedule (GS) civilian employee. FOSSAC also maintains an innovative Business Development Group (BDG) to develop strategic plans and growth management.

FOSSAC is a Defense Working Capital Fund (DWCF) activity analogous to a fee-for-service activity; as such, the organization is dependent on the ability to competitively market its services within the Federal Government (DoD). There are other organizations in the market that provide similar "best-value" support services. These include the Cooperative Administrative Support Units (CASU), various Fleet Industrial Supply Centers (FISC) and Shore Intermediate Maintenance Activities (SIMA). However, FOSSAC is recognized as the leader. This competitive advantage gives FOSSAC more flexibility for innovation and expansion.

Information technology resources are an essential part of controlling costs, maintaining connectivity with field (geographically separated) centers, and advertising services to potential customers. Subordinate to NAVSUP, FOSSAC closely monitors the policies and procedures adopted by NAVSUP in an effort to promote commonality within the Navy supply system and other systems associated with its business.

FOSSAC is about to feel the effects of the new standardized Information Technology (IT) infrastructure; the Navy-Marine Corps Intranet (NMCI). NMCI will change the way FOSSAC does business. The short-term goals are to integrate the current FOSSAC business process into the NMCI infrastructure. This process will involve consolidation and migration of current software applications to run under the NMCI infrastructure. The long-term goals are to leverage the standardization and integration that NMCI brings and develop a long-term IT improvement plan. This will ultimately advance the level of customer satisfaction, while reducing the cost of doing business; making FOSSAC the first choice among similar service providers.

C. RESEARCH QUESTIONS

This research addresses the following issues:

- With the current dependency on legacy applications, will the NMCI infrastructure adequately support the business processes currently in use at FOSSAC?
- Do current and accepted Enterprise Architecture Integration (EAI) methods adequately define a transition strategy for FOSSAC?
- Are their any other DoD organizations providing similar services and how does NMCI affect their technology strategy?
- Does existing Commercial/Government Off The Shelf (COTS/GOTS) software provide acceptable integration of legacy applications?
- How does the NMCI infrastructure affect the implementation of any recommended solutions?

D. SCOPE

This thesis is an initial assessment of the business and technology environment at FOSSAC and how to integrate the "old" system with the "new" NMCI system. The focus of this research is on documentation of current hardware and software environment, the design of technology architecture to support future hardware and software requirements and development of a migration plan from the current system to the future system. Specific research issues include mapping the functionality of the legacy applications to an Enterprise Applications Integration (EAI) environment and make recommendations on whether to web enable the applications within FOSSAC.

Additionally, the transition strategy should identify the return of investment (ROI) cash flows associated with each suggested action in order to determine economic feasibility.

The approach to this research focuses on a pragmatic assessment of the current business needs while ensuring that the overall infrastructure is improved as a result of delivering a potential solution or recommendation. It should allow incremental benefit to be achieved, with minimum disruption to the existing organization. Follow on research would focus on evaluating the conclusions of this thesis and actual implementation of an integration solution.

E. METHODOLOGY

The methodology used in this thesis research consists of the following steps:

- Interview FOSSAC personnel to determine their use of desktops applications to conduct day-to-day business and determine the workflow processes within each department.
- Interview the FOSSAC and EDS information technology support staff to determine the configuration of the current and proposed network infrastructure.
- Determine the effect of the technological change imposed by NMCI on FOSSAC employees. Do they feel they can adequately perform their daily tasks?
- Research integration methods used by the commercial sector and their compatibility with the NMC environment at FOSSAC.

F. ORGANIZATION

This thesis is organized to address the objectives of the research involved in five chapters:

Chapter II provides a description of Enterprise Architecture Integration (EAI) methodology and how it integrates a business environment. It includes a detailed definition, the basic building blocks and the different types of EAI. The chapter concludes with a discussion of some issues of concern for planners adopting EAI.

Chapter III provides an overview of NMCI infrastructure and what services it provides to the Navy and Marine Corps. Specifically, how NMCI contract services the current system, addressing hardware, software, peripherals and networking environment support. The chapter continues with a proposed network infrastructure for FOSSAC as it is developed under the EAI methodology. It proposes three platforms options for consideration, and analyzes the advantages and disadvantages of each. Then it describes recommendations and strategies on how a migration plan should be implemented.

Chapter IV discusses change and human factors that must be considered for success when implementing technological changes. This chapter provides models for organizational leaders to frame their approach to organizational change.

Chapter V summarizes the findings, concluding with a recommendation resulting from the analysis and identifies areas of further study.

G. SUMMARY

This chapter identified the major thesis topic, outlined the research methodology, scope of research, and organization of the thesis. The next chapter provides background and supportive information necessary for understanding the concepts of EAI.

THIS PAGE INTENTIONALLY LEFT BLANK

II. ENTERPRISE APPLICATION INTEGRATION METHODOLOGY

Integrating an enterprise is a daunting task. The key to successfully executing enterprise integration is the guidance from management, support from the stakeholders and an understanding of how it supports the "bottom line" of the business. The purpose of this chapter is to provide the reader with a broad overview of the definitions, concepts and methodology used in planning and architecting an Enterprise Application Integration (EAI) solution for the target information system at FOSSAC. The methodology outlined in this chapter is adapted from the EAI methodology used extensively by industry.

A. ENTERPRISE INTEGRATION

Survival in today's global economic environment requires innovative business practices, dynamic management techniques, and clear strategic vision. Information technology is one of the tools that help leaders ensure organizational viability by reducing the time to implement changes. Private and public organizations have been struggling for some time to find better ways to integrate information systems and at the same time achieve portability and platform independence. [Ref.1] Information technology has woven itself into the fabric of organizations and has created a large number of non-integrated legacy applications commonly referred to as "stovepipes". These legacy systems have hindered the organization's ability to scale and maintain compatibility across the enterprise. Efforts to regain control of the IT infrastructure have led some businesses to purchase expensive Enterprise Resource Planning (ERP) solutions. These multi-mode software application packages are not the panacea that some businesses had hoped. Organizations have invested millions of dollars in ERP packages only to find that the organization was incapable of changing its business processes to conform to the ERP package. ERP vendors have noted the reluctance for businesses to adopt expensive, all-in-one ERP solutions. In an effort to increase acceptance, ERP vendors are trying to create modular, interoperable packages. However, many organizations don't need or can't afford a packaged ERP solution. Depending on the level of "dis-integration" among the legacy applications, these organizations may better served by creating an internal EAI solution and incorporating Internet based options.

Organizations are continuously trying to find ways to balance the new business processes and manage data in more useful ways. Stovepipe systems do not provide effective methods of accessing data and processes within their own environments. [Ref.2] The challenge is integrating them within the enterprise. According to the Gartner group, 35-60% of an organization's information technology resources are spent on integration. [Ref.1]. EAI is the methodology of developing an internal ERP solution as opposed to purchasing a costly, all–encompassing external solution.

1. Definition and Components

EAI is a business computing term for the plans, methods, and tools aimed at modernizing, consolidating, and coordinating the computer applications in an enterprise. EAI can be defined as "the ongoing process of putting an infrastructure in place, so that a logical environment is created that allows business people to easily deploy new or changing business processes that rely on IT." [Ref.3]

The following paragraphs provide a more detailed description of EAI and its components.

a. Ongoing

Ongoing implies persistence, referring to how the company evolves from its current IT application environment to an EAI-enabled infrastructure or target environment. This is an iterative process with changes occurring in phases; this is not a one-time exercise and requires a longer-term vision – each step must be consistent

b. Process

Process in EAI refers to a series of actions, changes, or functions to achieve a result. This step is vital because it determines how the business will address needs, priorities, objectives, goals and quality criteria. It also characterizes the business' current and target process that will eventually run the business. Processes are done incrementally and take time; therefore, it is essential that a plan is developed to provide guidance for future requirements.

c. Infrastructure

The EAI process will result in the deployment of an infrastructure that serves the strategic business goals, providing tactical solutions in various phases. Infrastructure is the hardware, software, transmission media, and users that support the flow and processing of information. The infrastructure must be consistent with architecture

d. Logical Environment

The logical environment is an image or behavioral view of the business processes. It is an abstract view of the business without concern for the specifics of the individual technical systems or applications. The logical environment should remain relatively consistent regardless of changes in the underlying technical infrastructure.

e. Business People

Business people are the organization's corporate knowledge and will build the logical environment. They will have the most detailed understanding of the business process and must understand the underlying IT domain. However, they must ensure this domain knowledge is communicated to the IT people who build the technical infrastructure. EAI requires unity of effort between business and technical people from the beginning.

f. New or Changing

New or changing refers to the "to-be" environment as opposed to the “as-is” environment that EAI will create. An EAI strategy imposed on a corporate culture can have major effects on the organization. Top-level management must prepare its people for the change ahead.

g. Business Process

Business process is the final keyword of the EAI definition. EAI seeks to first understand the business. By understanding the function of the business first, the architecture definition is driven by the needs of the business, not by the perceived need to adopt a particular technology.

B. BASIC BUILDING BLOCKS OF EAI

In order to implement EAI successfully both methodology and technology must be integrated. Businesses today are constantly seeking ways to conduct commerce more efficiently and more profitably. Some succeed, others fail, but the fact is technology enables a business rapidly apply changes. As quoted by Fingar "In the interlocked cycles of technology and business advances, the issue companies face are not just about business, not just about technology. They are inseparably about both."[Ref.4] An EAI architecture is the combination of technologies brought together in a structured manner, based on four basic building blocks. [Ref.1] The four building blocks are: communications model, method of integration, middleware and services. This section provides an overview of the EAI framework.

1. Communications Model

The communications model describes the manner in which systems can interact. This is critical in maintaining flexibility, scalability, and interoperability. There are two basic types: synchronous and asynchronous.

a. Synchronous Communication

Synchronous communication is a form of communication that requires the sending and receiving application to running concurrently. This form of communication is tightly coupled, meaning that an application issues a request and waits until it receives a response from the other application before continuing.. Request/reply, one-way, and polling are three poplar types of synchronous communication.

b. Asynchronous Communication

Asynchronous communication is a form of communication by which sending and receiving application can operate independently. This is a loosely coupled form of communication; the applications do not have to be running or available simultaneously. An application sends a request and may or may not wait for a response. Message passing, publish/subscribe, and broadcast are three popular types of asynchronous communication.

2. Methods of Integration

The method of integration refers to the approach used to construct a request from a sender to a receiver. The request is constructed thought the use of connectors or adapters. Connectors and adapters are access points. The access point allows either a message or invocation on an interface to be passed into the application. These are required to create the “plug" into the application through which requests are transmitted. Two primary ways of integration are messaging and interface definitions.

a. Messaging

Messaging is the creation, storage, exchange, and management of text, images, voice, telex, fax, e-mail, paging, and Electronic Data Interchange (EDI) over a communications network.

b. Interface Definitions

The sender communicates through an interface, which defines the actions that can be invoked by an application. Any data to be processed is sent through the interface. The interface must be associated with an application in order for any integration to be successful.

3. Middleware

Middleware is used as a mechanism to move information and share business logic between existing applications. [Ref.1] Middleware can also be defined as a layer of utility software that sits between application and systems software to transparently integrate differing technologies to provide interoperability. [Ref.5] This allows disparate technology-based systems to interconnect. EAI architectures are based on middleware. There are five basic types of middleware in the market today.

a. Remote Procedure Calls (RPC)

RPC is the oldest type of middleware. It is a form of application-to-application communication that hides the intricacies of the network by using an ordinary procedure call mechanism. It is a complex, tightly coupled process that is losing favor to more modern, object oriented methods.

b. Database Access Middleware (DAM)

Database access middleware is any middleware that facilitates communications with a database. It allows access to distributed data whether it is from an application or between databases. DAM is the most common middleware but also the most limited in functionality and is usually combined with other forms of middleware. DAM can only respond to externally generated requests (i.e. a client requesting data from a server).

c. Message Oriented Middleware (MOM)

MOM provides the ability to integrate diverse applications through the use of messages, most commonly through the use of message queuing. It is a loosely coupled asynchronous process that provides the ability to create, manipulate, store, and communicate the messages. Message-oriented middleware takes care of relaying data from one application to another by "wrapping" that data in a message format, similar to the way e-mail works.

d. Distributed Object Technology (DOT)

DOT facilitates inter-application communications. This type of middleware can be classified as a set of small application programs that utilize standard interfaces and protocols to communicate with one another. An example of this is the Common Object Request Broker Architecture (CORBA). CORBA is one of a group of protocols for communicating in an object-oriented architecture. It extends the concepts of object-oriented technology to distributed processing.

e. Transaction Processing Monitors (TPM) and Object Monitors

TPM is the most complex of the middleware options. TPMs sit between front-end applications and back-end databases to manage the writing and reading of transactional data. TPM middleware preserves the integrity of a transaction. It allows a transaction to be formed by a sender and then ensure that it gets to the right place, at the right time, and completed in the right order. Object Monitors are advanced forms of TPMs providing TPM functionality but constructed according to object-oriented specifications.[Ref.1] [Ref.5]

Middleware is the software technology that integrates applications together in an enterprise. There is no magical "plug-and-play" or "one-middleware fits all" solution that will address all of the needs of a business. But for this to be successful, the business people and the IT staff must choose the right mix EAI tools in order to apply them to the right type of business process.

4. Service Building Blocks

Service building blocks are functional extensions to basic communication model. In the simplest implementation of an EAI, these building blocks provide reusable communication services to provide the message broker with the necessary information for inter-process communications. The services are intended to reduce the burden of applying the core technology.

a. Directory Service

EAI involves communication between distributed systems. A directory service is necessary to track all the components and key information about the system. It is used to automate the action of locating any element.

b. Lifecycle Management

This service provides the ability to monitor the overall integrity of inter-process communication. This service aids the EAI developer by automating the creation of objects or messages as well as ensuring that they are properly managed and disposed of on completion of a task.

c. Security

The security service ensures confidentiality, integrity, authenticity and availability of network resources. This service is analogous to an access control list for distributed EAI applications.

d. Conversion and Transformation

Data exist in many formats with different definitions. It is necessary to be able to convert and transform data into the correct format to properly complete any integration. This service, also known as an a adapter, is analogous to a set of libraries that map the difference between the target and source applications.

e. Persistence

This service provides the capability to save information and data by ensuring that state information and data are safely stored; this is critical to ensuring that information is not lost. A persistence service should be included to provide a consistent interface and an orderly method to manage data transactions.

f. Events

Event tracking is also known as an exception service provides the ability to identify the occurrence of a specific problem or other unique events. Examples of exception events are the occurrence of a business rule violation or improper termination of a transaction (a persistence violation).

g. Notification

A notification service Once an event is detected the notification service will alert the interested component that the event has occurred. [Ref.1]

An adequate EAI solution should support most, if not all, of these services and should be considered during EAI tool selection. As the business grows, leaders must consider adaptive and flexible plans to support all these services and to handle the different levels of integration.

C. TYPES OF EAI

When contemplating EAI in an organization, the business must first understand the sum and content of the business processes and data in that organization. Both business people and IT must work together to select which processes and data elements require integration. This process of integration can occur at three points in an application: the presentation, functional, and data layer. [Ref.1]

1. Presentation Integration Model

The presentation integration model, also called the user-interface model, is based on the concept of accessing the legacy application through its existing presentation logic. [Ref.1] The requirement for improved access included the ability to integrate with multiple application as well as added business logic related to the management of the interface, such as validation, error checking, and calculation. By presentation we are referring to the user interface that provides access to an application. Figure 1 shows how the presentation integration model integrates through the user interface of applications.

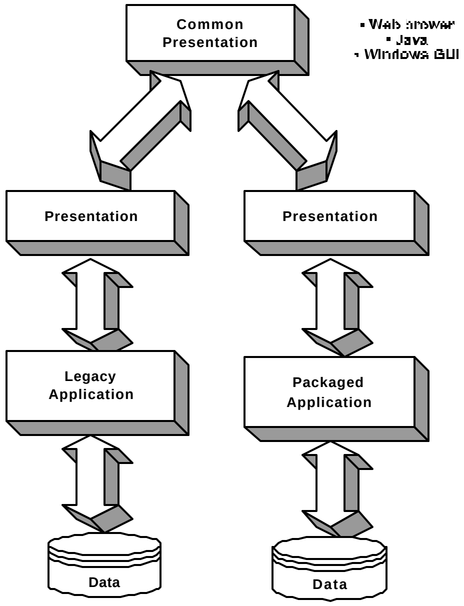


Figure 1. Presentation Integration Model (After: Ref.1)

Architects and developers are able to bundle applications by using their user interfaces as a common point of integration known as "screen scraping". [Ref.2] Screen scraping uses screen-based data capture and mapping or advanced terminal emulation to translate between legacy application programs and new user interfaces (i.e.; a Web browser). This allows continued access to the logic and data associated with the legacy programs. However, this method of integration results in a tightly coupled system. This means that if a change occurs in the presentation (user interface), the underlying application must be re-mapped to the user interface. Although limited in flexibility, this method is a commonly used technology for integrating systems.

a. Why Use the Presentation Integration Model

A business would use the presentation model when they need the following:

- PC-based user interface on an existing terminal-based application in order to provide an easier-to-use interface for an end user
- Present an interface that the user perceives to be a single application but is, in fact, the composite of several applications

- Integrate with an application whose only useful and implemental integration point is through its presentation [Ref.1]

This form of integration is useful only when the integration can be accomplished using the user interface or presentation level of the legacy applications. It is a simple form of integration requiring limited expertise in the integration tool, and therefore it has lower cost to implement. Reusability across application is limited, however, there are a limited number of features and functions. For instance, presentation integration can be used to improve a user's experience by reducing the complexity of accessing multiple applications. [Ref. 1]

b. Pros and Cons of the Presentation Integration Model

Pros include:

- Easy to accomplish and done relatively quickly
- Less complex than either data or functional logic
- When tools work together they do most of the work necessary to create the integration

Cons include:

- Most limiting of the three models
- Integration occurs only at the user interface level
- Can have performance bottlenecks because it adds an extra layer of software to the existing application
- Not well suited for Internet since any changes in the user interface require re-mapping

2. Data Integration Model

The Data Integration Model is based on the middleware integrating the data components at the lowest level, bypassing the application and presentation layers. The system allows the sharing of information via its middleware. Once integrated, it may be used by an application or may require more sophisticated integration with custom databases/files managed by the application. [Ref.1] Figure 2 depicts this model.

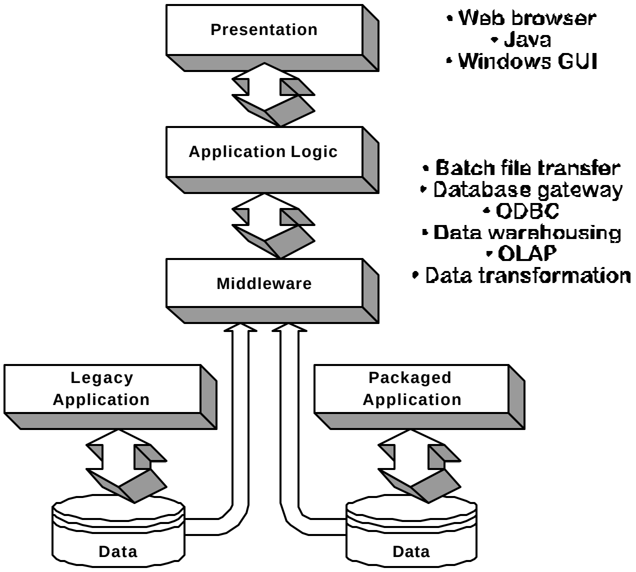


Figure 2. Data Integration Model (After: Ref.1)

a. Why Use the Data Integration Model

A business would use the data integration model when they need the following:

- Combine data from multiple sources for analysis and decision-making
- Provide multiple applications with read access to a common source of information
- Allow data to be extracted from one source and reformatted and updated in another

b. Pros and Cons of the Data Integration Model

Pros include:

- Provides greater flexibility than the presentation integration model
- Allows access to either a complete set of data or subsets depending on the need of the new application
- Simplifies access to data sources, which promotes rapid integration
- Allows data to be reused across other applications

Cons include:

- A possible need to rewrite the business logic
- Each integration is tied to a data model, if a data model changes, the integration may break, making data integration sensitive to change.

3. Functional Integration Model

The functional integration model, which is also called the business-process integration or application interface-level model, is based on integration of software at the code level. Software invokes existing functionally from other new or existing applications (i.e.; SAP, PeopleSoft, or Baan), therefore, the integration is done through interfaces to the software. [Ref.1]

The functional integration model integrates at the business logic level, as opposed to the presentation or data levels. Figure 3 depicts the functional integration model.

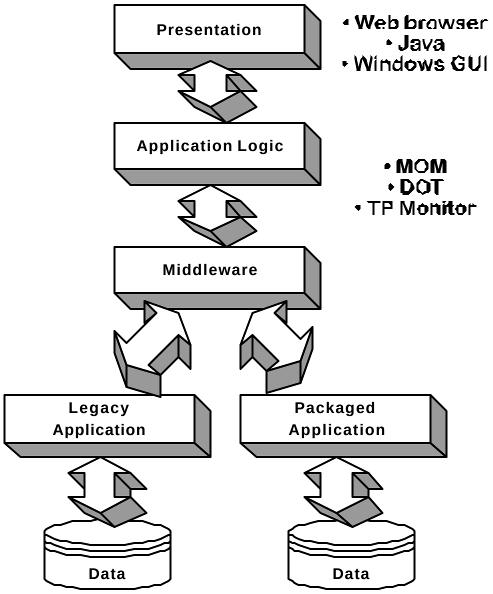


Figure 3. Functional Integration Model (After: Ref.1)

Functional integration is more flexible than data and presentation integration. It can be broadly applied using three different approaches. Each approach has different characteristics and is used to solve a different type of functional integraton problems. These functions include data consistency, multi-step processes, and plug-and-play components. Data consistency is the coordination of information update from one or more sources across integrated applications. Multi-step process is a coordinated set of actions executed across integrated applications. And finally, plug-and-play components are the creation of reusable interfaces across applications that simplify construction of new applications.

a. Why Use the Functional Integration Model

A business would use the function integration model when it needs the following:

- PC-based user interface to access an existing terminal-based application in order to provide an easier-to-use application for an end user by replacing the existing terminal interface and directly accessing the code from the new interface
- Present an interface that the user perceives to be a single application but is, in fact, the composite of several applications by accessing the functionality of each application and integrating through the new use interface
- Combine data from multiple sources for analysis and decision making
- Provide multiple applications with read access to a common source of information
- Allow data to be extracted from one source and reformatted and updated in another

b. Pros and Cons of the Functional Integration Model

Pros include:

- Provides the most robust integraton capabilities of all the models
- It is the most flexible in the problems it can solve and be used to solve presentation or data integration problems

- When properly applied, it provides a higher degree of component reuse than the other two integration models
- Easer to web-enable than the other two models

Cons include:

- More complex when it deals with integrating at the business logic level
- Steeper learning curve required for the software coding
- Accessing business logic may be difficult because the source code may not exist or there may be no API's
- Does not facilitate incremental implementation. Tends to be an enterprise-specific solution.

In summary, the system infrastructure will dictate the integration model best suited to support the business goals. Table 1 is a summary of integration requirements of each type.

Type of Integration	**Requirement of use**
Presentation	Shared front-end
	Need to update different data sources from single front-end
Functionality	Application processing logic required interpreting data from different applications.
	Addition of processing logic required to integrate functionality from different applications
	Transactional integrity between application required
Data	Need to update date from multiple sources
	Data needs to be synchronized between databases

Table 1. Types of Integration (After:Ref.1)

D. MARKET DRIVERS OF EAI

There are many trends in the information market of today; five significant business trends stand out.

First, is the proliferation of specialized packaged applications. This is the largest factor driving the market. Within the last decade, organizations were forced to make large investments to deal with the Y2K problem. Packaged applications (ERP's) were viewed as a quick fix for a multitude of problems. Implementing ERP was a huge undertaking and many companies were persuaded into making large investments. Although the Y2K problem no longer exists, many organizations are still tied to their ERP investments and have not realized significant gains in productivity or profitability.

Second, are mergers and acquisitions. When companies are bought out or merge into one company, integration efforts become a problem. Discrete and new systems may have problems migrating because skilled IT labor or a communication infrastructure has not been established.

Third, is the Supply-Chain Management (SCM) and Customer Relationship Management (CRM) aspect. In an effort to integrate suppliers, distributors and customers, businesses strive to achieve high levels of SCM and CRM. Enterprises must realize that they have to extend their processes out to partners and customers by providing access to business data and process flows.

Fourth, is streamlining the processes linked to e-business. By exposing portions of the front and back office systems the organizations can establish a direct link to the information systems of business partners. This is commonly referred to as Business-to-Business (B2B). Figure 4 shows this linking.

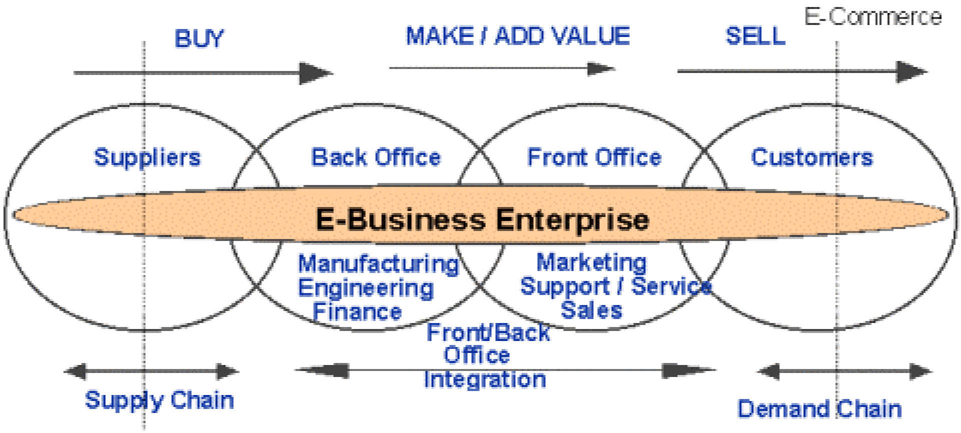


Figure 4. E-Business Enterprise

Last, is technology wealth. Message queuing, data transformation, business-process modeling, and middleware are technology advancements that business environments are using for competitive advantage. As mentioned before, the challenge is to tie the array of processes and data together to conform to their environment. Integration tools, such as EAI, should make that goal less imposing.

E. EAI CHALLENGES

When implementing an EAI solution, organizations must look to the future to avoid creating new stovepipes. The following is discussion of the EAI challenges followed by a look at complimentary technology that can help organizations implement web-based solutions.

1. Inconsistent Approach

IT architecture results from an amalgamation of technology implemented over a period of time. The diverse applications and technologies of existing applications, combined with new applications that are continually being introduced can lead to chaos in the system. Most business and government agencies started their information technology architecture as a means of automating manual processes. As time passed, and technology changed, that architecture became more complex, more ad-hoc and less integrated. Most organizations worked in an environment where each department or business unit developed their own applications, databases and processes with little concern for integration across the enterprise. This ad-hoc nature of the information systems has hurt the competitive standing of many companies. Rather than channel the innovative efforts of the various business groups, most IT departments wrestled for control over application development. The end result was that most of these organizations had lost control over the development and fielding of new applications. Add to that the various hardware platforms that have grown over time, it was clear that there was no coherent architecture.

2. Cost Versus Money to Implement

In today’s economy timely, accurate information is the foundation upon which a business must compete. The level and complexity of integration problems were demonstrated by the Year 2000 (Y2K) problem. Billions of dollars and man-hours were spent trying to “fix” a problem that was magnified by the multitude of complex interrelationships between application programs and databases. In the internal, B2B and B2C arenas of business it is imperative that a business present a united front to the customer. The best way that that can be achieved is through a process of Enterprise Application Integration.

EAI as mentioned before can be loosely defined as the creation of business solutions by combining applications using common middleware. As an organization moves from an ad-hoc, functional viewpoint to an integrated enterprise viewpoint, the transition will likely be chaotic as the organization fine-tunes the implementation. ERP is not a viable solution for many organizations because the business processes incorporated in an ERP package may introduce more risk than is tolerable for a particular business or industry. Until ERP solutions become modular or allow incremental implementation, EAI solutions will provide a lower risk, lower cost solution.

3. Staff and Skill Shortages

A shortage of skilled personnel is a major barrier to successfully implement EAI. The integration of business people and IT planners is crucial to EAI’s success. As mentioned before, IT planners must understand the core business, and the goals for successfully reaching those core business strategies. Integrating the business people is just as important as integrating the business process. However, in some organizations, the IT departments possess an intuitional arrogance regarding their role in automating the business process. The EAI is a methodology as been around for years, but is just now getting internal recognition by middle management many companies. It is critical that all people involved understand the methodology before attempting to implement technological or business process changes, or the creditability of the entire effort will be at risk. An aggressive education program, using a variety of methods (seminars, training classes, and outside consultants) will mitigate the risk.

4. Organizational Structure

Support of management is the key to any successful change effort, and EAI is no exception. The key decision makers must be educated about EAI, and realize that the

EAI effort is key to reaching their business goals. Managing this change effort will require an organized effort, patience, and training. Successful integration of an enterprise requires a high level of cooperation between the business staff and the technical staff; perhaps more than either side is willing to admit. Organizational leaders may be faced with restructuring the organization to facilitate the increased level of cooperation.

5. Securing Applications

EAI applications require a consistent and coherent security architecture that will fit into the enterprise. In every business, the security problem is real. When addressing EAI security, these systems may require a more comprehensive and integrated approach than security for more traditional application. Therefore, the IT department must be properly staffed to handle the changes in security policies and guidelines to support the business goals associated with the EAI enabled environment. The technical aspects of securing an enterprise can be difficult to articulate to management. The return on investment (ROI) for IT staffing is difficult to quantify, as are the losses in the event of a security breach. Along with the regular enforcement of IT security policies, the IT staff must educate management on the organizational vulnerabilities and the consequences of not protecting the enterprise. Vulnerabilities minus protections equals residual risk; management needs to decide what residual risk is tolerable.

F. IMPLEMENTING EAI

The goal of implementing an EAI-structured architecture is to enable critical new solutions for the enterprise. First, it improves relationships with customers. The bottom line is to please the customer no matter what the business (i.e.; product(s) or services). Second, it supports strengthened supply chains. Traditional supply chains are linear. In order to develop a coherent business process, these supply chains must be reengineered for integration into the business. This refers to the supply-chain partners and others outside organizations. Third, it helps to streamline internal process. Outside organizations must work in unison with the enterprise's internal process. EAI techniques can be used to simplify information flow between departments and divisions of the enterprise. Lastly, it helps bring new applications online more quickly. By leveraging the capabilities of existing applications, the work is half-way done. Using current functionality, the enterprise can create front-end channels like the Web or new, composite applications. Also through the use of middleware, the development of an integrated process will focus on the business aspects of the application, not the “plumbing”. [Ref. 1] Implementing an EAI strategy should be based on a pragmatic assessment of current business needs while ensuring that the overall architecture is improved as a result of delivering the solution. It should allow incremental benefits to be achieved, with minimum disruption to the existing organization.

G. COMPLIMENTARY TECHNOLOGY

When considering an EAI implementation, one must consider the size of the organization and the level of "dis-integration" that currently exists. EAI has a reputation as an expensive, long-term process of building middleware applications to perform the integration. As in the case of an ERP solution, an EAI solution may also be cost prohibitive. A complimentary technology called Web Services can be used in parallel and in some cases can be used in place of a traditional EAI approach. Depending on the size and data structure of the legacy systems, Web Services can provide a lower cost alternative to the application integration problem. Concurrent implementation of Web Services and EAI may offer the best long-term integration solution.

The concept of Web services is often positioned as a replacement for EAI solutions. However, from the definitions below, there is quite a difference in the scope of these two approaches.

- Web Services are modular applications that can be accessed by a network through a standard eXtensible Markup Language (XML) format interface.
- "EAI is a concept that groups a set of methods, technologies ad tools to consolidate and coordinate different applications, leading to the urbanization of the enterprise's information system.” [Ref.6]

Web services are not a replacement for EAI. In reference to Figure 5, Web services are providing integration from the "middleware" layer up to the "presentation" layer.

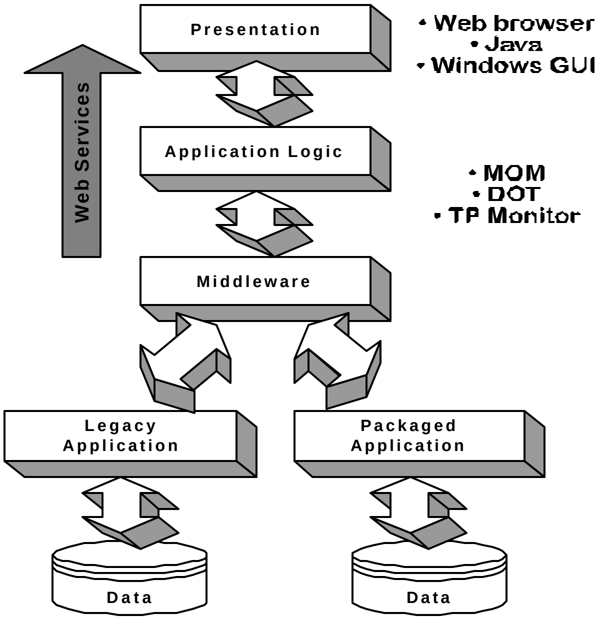


Figure 5. Web Services Integration (After: Ref.1)

The Web services technology is not mature enough to handle the complexity of integrating the lower layers. "Twenty percent of the integration problems will remain complex, requiring expensive proprietary solutions. But the vast majority of integration is going to be achieved by Web services."[Ref.7]

Before continuing, it is necessary to define of some key terms to aid in a more detailed discussion of Web services:

- DTD - Document Type Definition (DTD) is a specific definition that follows the rules of the Standard Generalized Markup Language (SGML). A DTD accompanies the XML document to identify how each document is to be processed. By sending a DTD with an XML document, any location that has a XML "reader or "SGML compiler") can process the document to display it as intended by the document creator. Creating a document with XML following the SGML rules allows for a single standard SGML compiler to display the document. In the case of a web page, the "compiler" or document handler is the Web browser. [Ref.8]
- HTML - HyperText Markup Language is the language of the World-Wide Web (WWW). Web pages are text documents written in HTML, a

Document Type Definition (DTD that is a set of tagging instructions to describe the appearance of the document in a web browser.

- XML – eXtensible Markup Language is also a DTD and is HTML compatible. XML is differs from HTML in that it is capable of providing context for a document. It is a set of tagging instructions to define and format a document in a web-browser-compatible manner. The tags describe the hierarchical *structure* of a document as opposed to just the on-screen *appearance* as HTML does.
- HTTP Hypertext Transfer Protocol is the communication standard that governs the transfer of Hypertext between client and server computers. IT is the standard for document exchange on the WWW.
- SOAP - Simple Object Access Protocol is an XML-based protocol that allows activation of applications or objects within an application across the Internet. It defines the use of XML and HTTP to access services, objects, and servers independent of the computing platform. SOAP defines the practice of using XML and HTTP to invoke methods across networks and computer platforms.
- WSDL - Web Services Definition Language defines the grammar used by XML to describe network services as collections of communication endpoints capable of exchanging messages – it specifies the public interface for a Web service.
- UDDI - Universal Description, Discovery, and Integration is the framework for defining a data model for XML and SOAP. This data is used to describe a distributed directory of businesses and Web services. When a query is sent to find a web service, UDDI returns a pointer to the target. It is analogous to a registry containing the location information.
-

H. WEB SERVICES VERSUS EAI

1. A Review of EAI

The earlier in this chapter, a detailed description was provided of the components that make up the EAI architecture. Figure 6 is a pictorial summary of these components but with the added layer called the public interface. The three components shown in the public interface are communication/presentation agents that retrieve the desired information from the lower levels of the architecture.

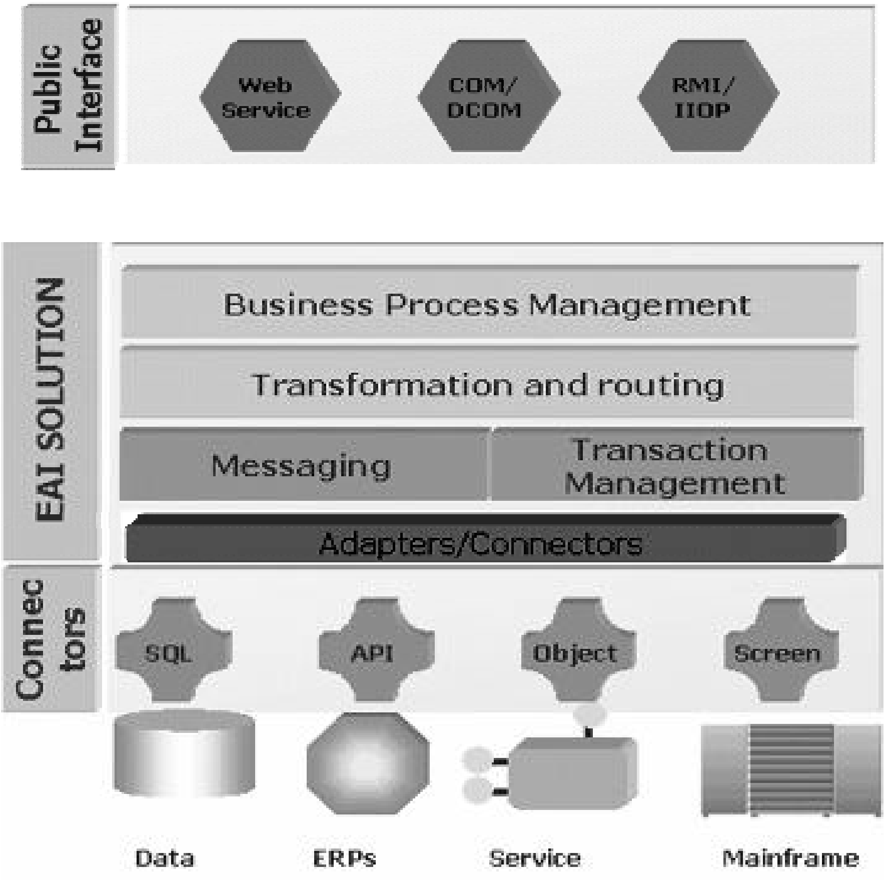


Figure 6. EAI Review (After:Ref.6)

2. Web Services

Web Services are "self-contained, self-describing, modular applications that can be published, located and invoked across the Web". [Ref. 9] The Web Services platform makes use of standard XML protocols making it platform, language and vendor independent and an ideal candidate for use in EAI solutions. The goal of Web Services is to provide functional integration of business logic at the application interface. This is done by creating interfaces to invoke existing functionality using XML and the associated protocol set. [Ref.2] The key features of the Web Services approach are:

a. Interoperability

Any Web service can interact with any other Web service. SOAP is a standard protocol that will provide this interaction.

b. Ubiquity

Web services communicate using HTTP and XML. Any device, supporting these technologies, can both host and access Web services.

c. Low Barrier to Entry

The concepts behind Web services are easy to understand and free toolkits from vendors like IBM and Microsoft allow developers to quickly create and deploy Web services.

d. Industry Support

All of the major vendors (Sun, IBM, Oracle, BEA and Microsoft) are supporting SOAP and the surrounding Web services technology. The Microsoft .NET (pronounced dot net) platform is built around Web services. Microsoft, in particular, hopes to capitalize on the popularity of Visual Basic and the ease of deploying these applications as an integrated part of Web services.[Ref.2]

A general representation of the Web services architecture is shown in Figure 7.

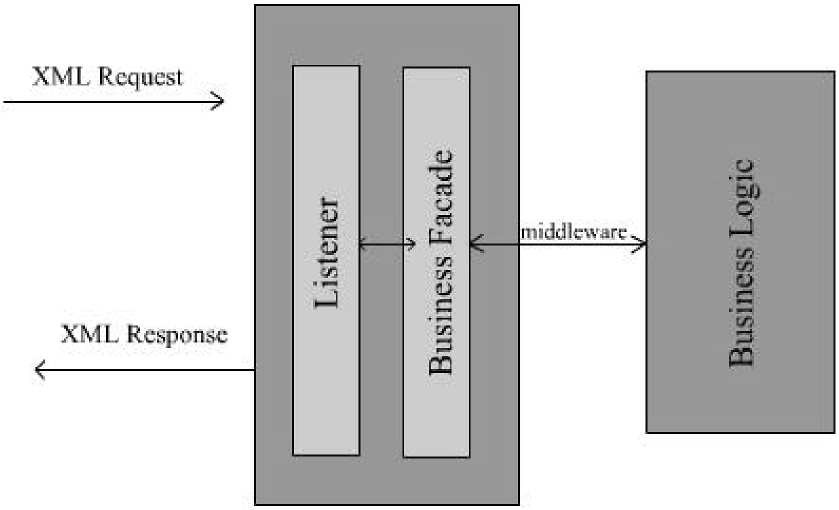


Figure 7. Generic Web Service Architecture.(After:Ref.10)

The block on the left is the Web services portion and needs a form of middleware to connect it to the logical representation of the business. The Listener is a platform-independent, presentation layer receiver. The listener passes the XML request (via HTTP) to the Business Façade. The Business Façade represents the application layer containing the business rules. In this general architecture, the Middleware represents the traditional (proprietary) EAI software linking the lower levels of the architecture.

Figure 8 shows a more detailed representation of the Web services architecture and the linking between the XML-based Web services layer and the proprietary middleware, depicted here as the Service Wrapper.

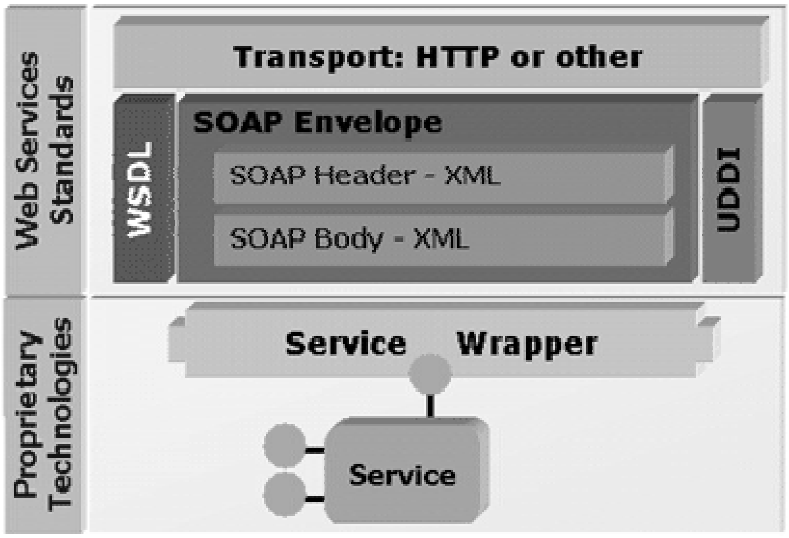


Figure 8. Web Service Architecture (After:Ref.6)

The basic Web Services platform uses XML to define and describe the data for interpretation and display in a web browser via HTTP. SOAP is the communication protocol, defining the format for exchanging data between computing platforms. Figure 7 highlights the role middleware plays in integrating (wrapping) the proprietary legacy applications before the Web Services can access them.

In an effort to summarize the above discussion of Web services, the following is offered: The computers on an enterprise network are linked via the Internet transport protocols (TCP/IP) and the documents on the network are created/defined via the "markup" languages (HTML and XML) and accompanied by a DTD to ensure proper translation. SOAP, WDSL and UDDI are the linking mechanisms for finding and retrieving the desired information.

- XML is used to describe the data
- UDDI is used to advertise or find desired services
- WDSL describes Web services
- SOAP is used to manage the asynchronous messaging services and the synchronous remote procedure calls for executing Web services. [Ref.6]

I. INTEGRATING EAI AND WEB SERVICES

The key to integrating EAI and Web services is to determine the proper mix of each integrator. As mentioned earlier, Web services are typically an "eighty-percent solution" to the integration problem. Some mix of traditional EAI services will be required depending on the age and size of the organization. Older organizations that were once dependent on "mainframe" computers may have their business logic embedded in COBOL coding. Large organizations will have to consider how to overcome the inertia to integrate the entire origination, consider smaller scale integration, or choose a total replacement of older legacy systems. Organization will have to weigh the cost of the three different EAI integrations strategies: Presentation, Data, or Functional integration or make the switch to Web based services

The hype about the capabilities of Web services has generated offerings form various vendors claiming to have an all-encompassing solution. These all-in-one (EAI and Web services) solutions are commonly referred to as Business Process Management Systems (BPMS). The vendors in this category treat the middleware as a component in the overall architecture. This approach will typically be a proprietary solution but accomplished with a mix of the platform independent, XML-based Web services and the more complex, Application Program Interface (API) and CORBA connector/adapter based technology. This total integration approach is being offered by industry leaders like IBM (WebSphere), Oracle (9iAS), BEA (WebLogic), and Sun Microsystems (SunONE).

In some cases, an organization can pick individual pieces of the various integration options using in-house personnel to execute the implementation. In the case of FOSSAC where the primary issue is data integration, combining data integration EAI with a Web services implementation is a viable option. FOSSAC has been a client-server (as opposed to mainframe) based computing environment from its inception. Analysis of FOSSAC's business process, their computing environment, and the transition to NMCI indicates that this may offer the best Return On Investment (ROI).

III. OVERVIEW OF CURRENT AND TARGET SYSTEMS

This chapter is intended to provide background information on the research sponsor and its current primary information system. It begins with a brief history of the Navy-Marine Corps Intranet and what effects it will have to these armed services. It briefly describes the history of FOSSAC's network architecture and the continuing development of their information system. It then examines in the composition of the existing information system, the incorporation of the Navy-Marine Corps Intranet (NMCI) concluding with recommended technology to facilitate integration of the current legacy application into the NMCI environment, including the potential to serve in a post-MNCI environment.

Overall, NMCI will apply the speed and might of world-class Internet technology to help the Navy and Marine Corps meet these critical objectives:

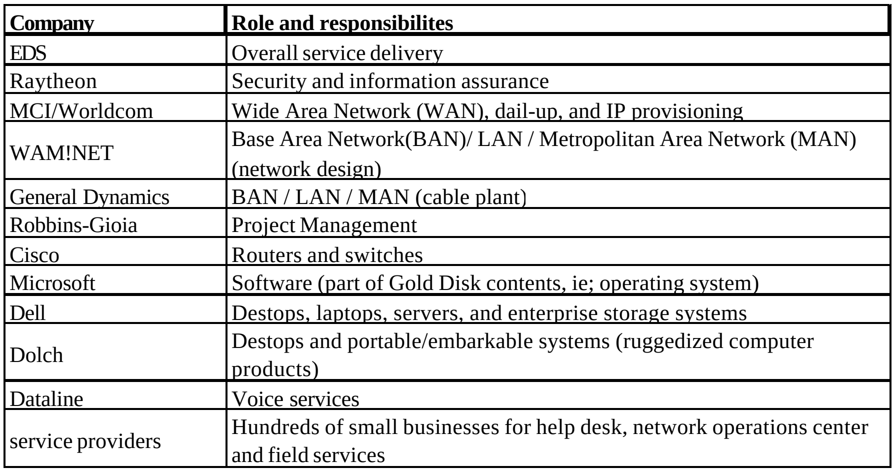
- Enhanced network security
- Interoperability among Combatant Commanders and with other Services
- Knowledge sharing across the globe
- Increased productivity
- Improved system reliability and quality of service
- Reduced cost of voice, video and data services

A. WHAT IS NMCI?

The Navy Marine Corps Intranet (NMCI) is a comprehensive, enterprise-wide outsourcing initiative that will make the full range of network-based information services available to Sailors and Marines, increasing combat readiness and effectiveness. The scope of the NMCI program is to provide value for the Navy and Marine Corps by proving secure, universal access to integrated voice, video and data communications services for a lower cost that the Department of the Navy is paying today. This outsourcing initiative includes hardware, software and physical infrastructure upgrades necessary to meet quality of service requirements. This contract also includes maintenance, training and operational support required to maintain the capital infrastructure. NMCI will link more than 360,000 desktops across the United States as well as sites in Puerto Rico, Iceland and Cuba and will afford pier-side connectivity to Navy vessels in port.

The NMCI contract was awarded on October 6, 2000 and since it inception has undergone several amendments and modifications. The last reported contract revision dated 24 July 2002, consisted of eighty-six Contract Line Item Numbers (CLINs) and thirty-seven Service Level Agreements (SLAs). The CLINs specify, in detail, the minimum required deliverables for the contract. The SLAs specify monetary awards (or penalties) for service related items such as network availability (up-time) and network defense against DoD-initiated vulnerability checks.

In an effort to coordinate the delivery of assets and services, EDS formed the Information Strike Force (ISF) to manage the transition to NMCI. The ISF represents the collaborative effort of the NMCI contract partners responsible for delivering NMCI functionality to the end user. Members of the Information Strike Force team are listed in Table 2.



Company	**Role and responsibilites**
EDS	Overall service delivery
Raytheon	Security and information assurance
MCI/Worldcom	Wide Area Network (WAN), dail-up, and IP provisioning
WAM!NET	Base Area Network(BAN)/ LAN / Metropolitan Area Network (MAN) (network design)
General Dynamics	BAN / LAN / MAN (cable plant)
Robbins-Gioia	Project Management
Cisco	Routers and switches
Microsoft	Software (part of Gold Disk contents, ie; operating system)
Dell	Destops, laptops, servers, and enterprise storage systems
Dolch	Destops and portable/embarkable systems (ruggedized computer products)
Dataline	Voice services
service providers	Hundreds of small businesses for help desk, network operations center and field services

Table 2. Company roles and responsibilities

B. LEGACY APPLICATION RATIONALIZATION

A formidable obstacle in implementing the NMCI, the magnitude of which was significantly underestimated, is the issue of "legacy applications". As of 24 July 2002, there are 31,287 legacy applications under evaluation within the DON; this is down from over 96,025 applications in February 2002. The goal is to reduce this number below 10,000.[Ref. 11] A significant reduction resulted from finding suitable alternative applications or eliminating the redundancy from different versions of the same application. Other applications were eliminated from consideration due to incompatibility with the Microsoft Windows 2000 operating system or failing to meet DoD and Navy security requirements.

The question of how to handle the remaining legacy applications has caused significant delays in the scheduled implementation of NMCI. The implementation at FOSSAC has been delayed in excess of four months. The certification process has not been able to keep pace with the Requests For Service (RFS) to evaluate the legacy applications. As a result, many organizations are still dependent on maintaining the pre-NMCI computers and applications. These organizations have provided rational arguments for maintaining access to the legacy applications while the certification process continues. Because these older applications have not passed certification or are still waiting for approval, they are not allowed to interact with the NMCI network. The applications on these separate, local networks are "quarantined" until receiving certification. Along with maintaining this separate, parallel network, local IT staffs are working to find certified alternatives or to convert/rewrite applications to comply with NMCI and DoD requirements. A more detailed description of the rationalization/certification process is shown below in Figure 9.

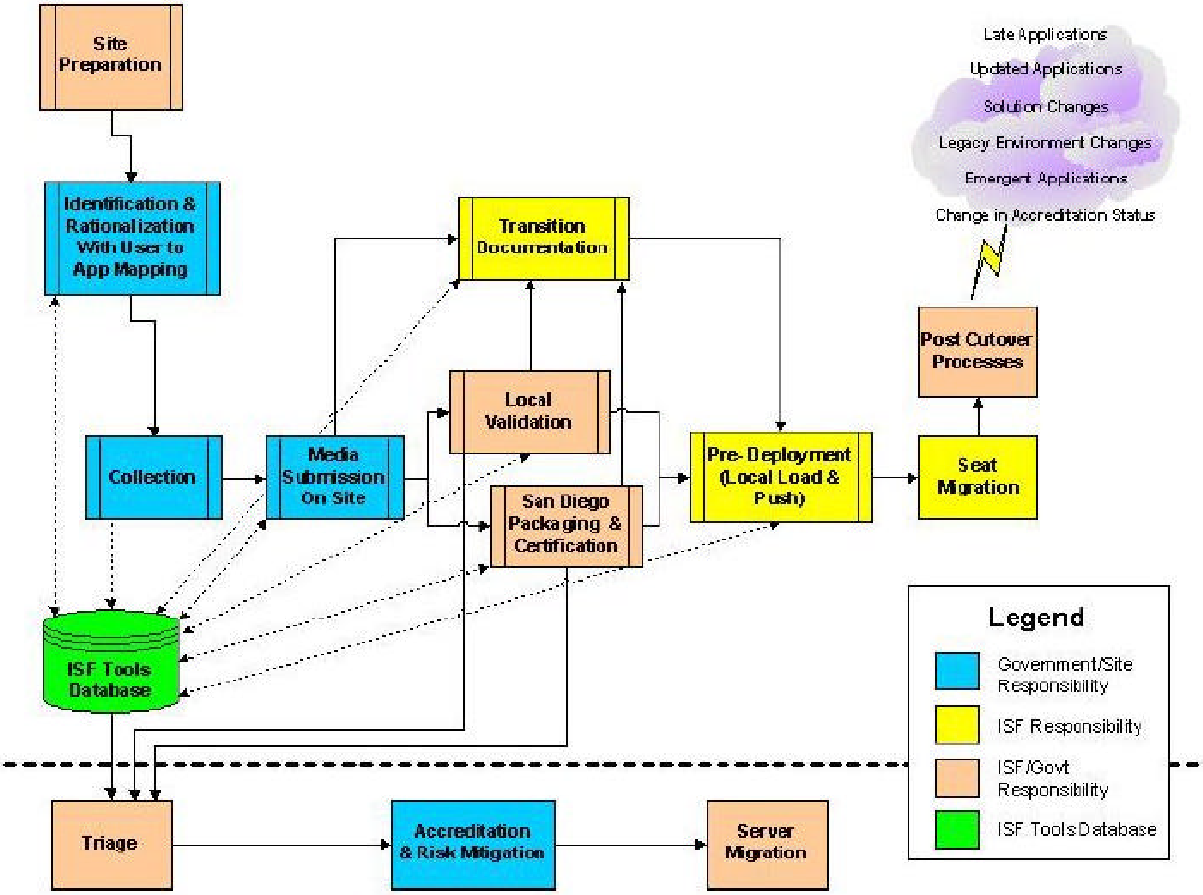


Figure 9. Rapid Certification Phase Process (After: Ref.12)

The certification process begins with identification of the software application and mapping it to an actual user. This mapping ensures that there is a bonafide user and a legitimate need to evaluate the application. The "owner" of the software fills out a Certification Phase Engineering Review Questionnaire (CPERQ) describing the functionality of the software and how it is used (client or server based). The CPERQ and an RFS accompany the software to Space and Naval Warfare Command (SPAWAR) San Diego for testing in the certification lab. Some software can be certified locally through a certification-by-association procedure. This reduces the time and labor to certify software if a determination is made that it is similar to an application already certified. A subset of the certification process is conducted on-site using what is called Point of presence In A Box (PIAB) also know as "POP in a Box". This local evaluation is done on an NMCI Windows 2000 workstation, to allow testing of the local configuration settings. These local procedures have been helpful in reducing the burden on SPAWAR for testing.

The overall goal of the certification process is twofold; to ensure Windows 2000 compatibility and to comply with the DoD Information Technology Security Classification and Accreditation Process (DITSCAP). DITSCAP is an all-encompassing security evaluation, which includes Security Penetration Testing, Monitoring Compliance, Risk-Based Management Review System Operation and Change Management Review. In terms of software certification the evaluation focuses on the Application Program Interfaces (APIs) and how programs interact with the Windows 2000 operating system. If the APIs are unstable or not documented in sufficient detail, the program may be an unacceptable risk to the network.

The certification process runs concurrent with the Assumption Of Responsibility (AOR) by the ISF in its progress toward seat migration (desktop installations). Software applications that fail certification or require a more detailed evaluation are sent to "triage". These applications are "quarantined" and are only authorized for use on the pre NMCI, legacy LAN. In the case of FOSSAC, there were several "essential", but uncertified applications that require maintaining a separate LAN.

C. STANDARDIAZATION OF ASSETS

1. Hardware

In an effort to reduce costs through hardware standardization, NMCI provides for installing four basic hardware (seat) types plus a variety mobile user seats (laptop computers). The primary desktop computers are referred to as Red, White, Blue, and Developer seats. Of course, since one size does not fit all, there are variations on all of these to account for specialized requirements. FOSSAC has limited special requirements and will be receiving a mix of the four basic seat types for stationary and mobile users. The primary difference between the Red, White, and Blue seat types is processing power, memory, and hard disk size. The developer's seat is a customized configuration based on the specific services (by individual CLINs) desired by the developer. The Developer seat provides designated personnel the ability to make hardware and software changes to the base configuration without intervention by the ISF service staff. Developers will be able to install more sophisticated software for tasks such as writing Visual Basic applications to perform user-specific tasks or to develop Web-based applications

2. Software

In an effort to reduce costs through software standardization, NMCI is installing a collection of applications referred to as the "Gold Disk". This disk contains the Microsoft Office 2000 Suite and other applications standardized across the enterprise (i.e. anti-virus, email, multimedia, web browser, etc.). This software package is described in detail in Appendix A.

D. NMCI IMPACT ON THE NAVY

Outsourcing IT is steadily gaining acceptance as a viable option to reduce management and infrastructure costs. Outsourcing has given the perception of making IT management "somebody else's problem", providing the *guarantee* of reduced costs from economies of scale, increased interoperability though standardization, and increased performance (throughput) by consolidating management and procurement.

Prior to the awarding the NMCI contract in October 2000, the Navy IT procurement process was handled at the individual command level. This resulted in a variety of localized IT infrastructures based on the needs of the local commands. These localized centers often had their own way of doing business and compatibility problems for sharing data became a Navy -wide problem. In an effort to reduce IT costs, reduce the proliferation of these local data centers and networks, and improve knowledge sharing, the Navy embarked on a five-year, 6.9 billion dollar consolidation effort to outsource its IT infrastructure.

Knowledge superiority and network-centric warfare have become necessary to defeat (or defend against) the asynchronous threat in today's environment. Information linking throughout the Department of Defense is essential and NMCI is viewed as an instrumental piece in advancing the Navy effort in network centric warfare.

Implementation of the NMCI contract has already become a culture shock to the end users. Regular correspondence to journals such as *Government Computing News* and *Federal Computer Weekly* tell the stories of the resistance and misunderstanding existing throughout the Navy and Marine Corps. A good portion of this resistance is due to poor management of organizational change. A detailed discussion of managing change is provided in chapter four of this thesis.

E. NMCI IMPACT ON THE MARINE CORPS

The Marine Corps has experienced less of a culture shock than the Navy due to an internal integration effort started almost three years prior to NMCI. The Marine Corps recognized the need for configuration and integration management back in 1997, establishing the Marine Corps Enterprise Network (MCEN). This network integrated the Marine Air Tactical Command and Control System and the Marine Corps Tactical Network. In 1998, one year after establishing the MCEN, the Marine Corps began centralized procurement, buying servers and computers centrally vice the previous practice of each command buying its own. Concurrent with centralized procurement, the Marine Corps began publishing enterprise software standards. Having completed a significant portion of the transition, the Marine Corps is ready to reap the benefits of NMCI.

F. THE FOSSAC NETWORK

Prior to August 2001, FOSSAC activities were geographically dispersed throughout Norfolk Naval Base and used a variety of means to maintain connectivity. In some cases, wireless links (microwave) were the only feasible methods. One thing unique to a naval base compared with a commercial site is the close proximity to ships and their radars. One activity, Price Fighters, was dependent on a wireless link for access to the base network and the Internet. This problem, among others led to the consolidation of FOSSAC on a single floor in a newly constructed building. The consolidated organization enjoyed reliable LAN connectivity to every desktop. However, the client-server arrangement and connectivity to the Base Area Network (BAN) was slow and unreliable.

G. THE CURRENT SYSTEM

FOSSAC's current operating architecture is in a stage of transition referred to Assumption of Responsibility by the ISF. In accordance with the NMCI contract, the ISF assumed the responsibility of maintaining the FOSSAC infrastructure and will begin the process of installing the NMCI certified Windows 2000 servers. Once the server infrastructure is established, the ISF will begin installing the software to manage the NMCI client computers. During this transitory phase, the old FOSSAC LAN is isolated from the new NMCI LAN. However, the software applications that passed the certification testing will be installed on the new NMCI servers and he process of distributing the new NMCI desktop computers begins in anticipation of "cutover".

1. Hardware and Network Plumbing

The desktop hardware environment is a mix of computers varying in processing power from Intel Pentium 166 MHz up to Pentium 850MHz. A pair of Pentium 400 MHz Novell Netware servers provided the network services. This network is relatively unsophisticated providing routine administrative support services (file, print, and email services). The network plumbing is approximately one year old with data and voice ports located at each employee workstation. Upon AOR, maintaining the FOSSAC LAN became the responsibility of the ISF. Unfortunately, the ISF was not sufficiently trained to operate and maintain a Novell Netware-based network, requiring the assistance of the FOSSAC IT staff. While the two networks are physically separated, the FOSSAC and NMCI personnel are cooperating to ensure both networks maintain connections to the Base network without compromising network security.

2. The Software Environment

The current software environment covers a wide range of applications that are essential to the viability of FOSSAC. Nearly all computers at FOSSAC have been upgraded to Windows 98 and are using the Microsoft Office 97 suite. The email service was handled through a Lotus Notes server running Windows NT. Prior to the arrival of

NMCI, each department within FOSSAC was able to purchase and install any software it felt was necessary to perform the daily tasks. In some cases COTS software was customized to perform very specific tasks.

Throughout, FOSSAC has wrestled to support and maintain accountability of these software applications installed in the work centers. Adding to the software management problem, significant income at FOSSAC was generated from partnerships with civilian contractors. These partnerships often required the purchase of additional software to maintain compatibility on certain projects. Lacking the personnel to track all these installations, software management became untenable. As a result, there was no clear accountability of the patches or service packs applied to the operating system or application software. Additionally, oversight was lost on version numbers among similar programs resulting in some incompatibility among the in-house work centers. To FOSSAC's benefit, the IT staff was diligent in maintaining network servers, routers and firewalls preventing any serious security or denial of service incidents.

3. NMCI Hardware Environment

Since the NMCI network has not been deployed to the desktop users, the current hardware environment differs little from the pre-NMCI environment. Part of the NMCI contract entails taking possession of any FOSSAC asset that meets the NMCI standards meaning some users, will have the exact same computer on their desk after cutover.

Once the NMCI assets are deployed to the desktop, the only change for the end user is that now, they will be required to logon to the network before being able to access any applications. Additionally, users will not be able to install or delete applications and they will not be able to modify desktop settings. This effectively locks down the computers as an "official use only" device.

4. NMCI Networking Environment

The original NMCI contract was primarily a "plumbing" contract, to provide guaranteed bandwidth and availability to the wall outlet. The contract was expanded to encompass the desktop environment to include software management and connectivity with peripherals (printers, scanners, etc.). The issue of securing the desktop environment is significantly more complex than just "router-to-router" security in the plumbing behind the walls. The desktop is where the applications and where mail attachments are opened – the most vulnerable part of the network. This vulnerability has been dealt with by drastically reducing the ability to modify the user interface.

5. Security Concerns

The ultimate goal of any information system is to transport data to from the sender to the receiver and to prevent the data from being intercepted or altered en route. This ensures the confidentiality, integrity, and authenticity of the data is protected. In an effort to fulfill this goal, the NMCI infrastructure employs various security tools. The first thing the desktop user notices is that the keyboard has a "card reader". Now, along with providing something the user knows (password), the user must also provide something he/she possesses (smart card/ID card) to access the network. Once the user has successfully logged on to the network, the Windows 2000 operating system limits what portions of the network are accessible based on the Access Control List (ACL) associated with the user's network profile. In the course of performing their work, users must routinely correspond by sending data across the network. To ensure integrity and confidentiality, NMCI uses the Public Key Infrastructure (PKI). This security measure issues digital "keys" referred to as a private key-public key pair to each user. These keys work together to provide a digital signature confirming the identity of the sender of the message. In a simple example, the sender encrypts the message with the private key (known only to the sender). The only way the message can be read is if it is decrypted with the sender's public key (available to the public). To provide confidentiality, the sender would encrypt the message with the receiver's public key. In this case the receiver could only decrypt the message with their private key. Combinations of the sender/receiver keys pairs provide a method for the secure exchange of data.

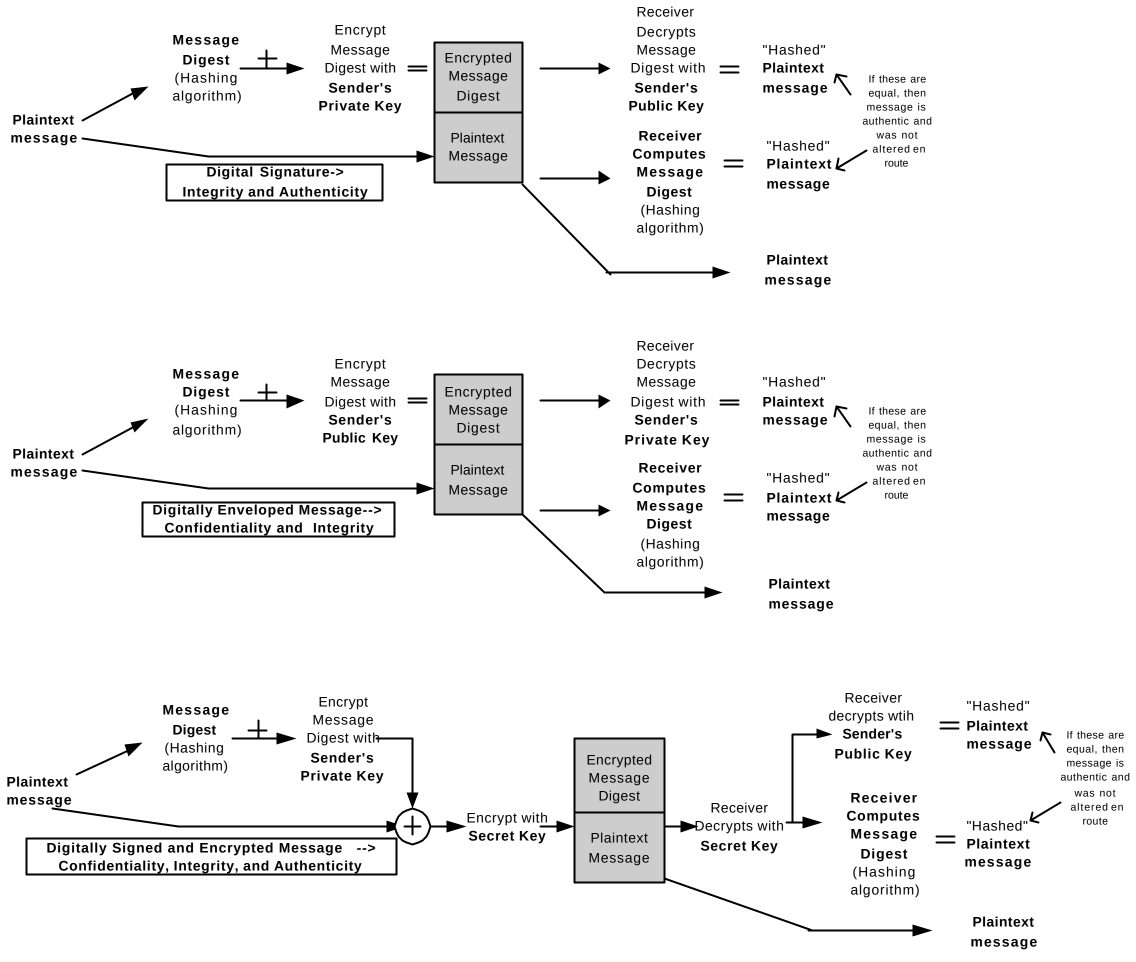
6. Security Implementations

The PKI is an accepted method to exchange the keys used to digitally sign data. However, there is one piece of the puzzle that is missing. The users do not have any tools at their disposal to provide “type 1” encryption of the actual data before the message is digitally signed with the private-public key pair. Type 1 encryption is a Federal

Information Processing Standard (FIPS) for protecting classified information. [Ref.13]

Referring to figure 10, below, the encryption process begins with creation of a digital “thumbprint” of the message. This process is a one-way (non-reversible) encryption algorithm referred to as "hashing"; this creates what is commonly referred to as the message digest. After the data is transmitted, the receiver will unwrap the data using the public/private key pair. The receiver now has two things, the plaintext message and the digest of the message. The receiver "hashes" the plaintext message, using the same process the sender used to create the original message digest. To ensure the message received is the same as that sent, the receiver just needs to make sure the sender and receiver message digests match. The process is described in figure 10, illustrating the benefit gained by encrypting the actual data vice just digitally signing the message.

The top scheme represents the method to digitally sign a message. The middle scheme, referred to as the Diffe-Hellman key exchange, represents the method to distribute a common secret key necessary to implement the bottom scheme. The bottom scheme is the most robust, providing confidentiality, integrity, and authenticity. There are commercially available tools to implement the digital signing plus encryption scheme. At the time of writing this thesis, these tools additional desktop encryption tools had not yet been incorporated in the NMCI contract. While beyond the scope of this thesis, additional information regarding data protection methods, can be found at the RSA Laboratories Web (www.rsasecurity.com).


Plaintext message
Message Digest (Hashing algorithm)
+
Encrypt Message Digest with Sender's Private Key
=
Encrypted Message Digest
Plaintext Message
Digital Signature-> Integrity and Authenticity
Receiver Decrypts Message Digest with Sender's Public Key
=
"Hashed" Plaintext message
Receiver Computes Message Digest (Hashing algorithm)
=
"Hashed" Plaintext message
If these are equal, then message is authentic and was not altered en route
Plaintext message
Plaintext message
Message Digest (Hashing algorithm)
+
Encrypt Message Digest with Sender's Public Key
=
Encrypted Message Digest
Plaintext Message
Digitally Enveloped Message--> Confidentiality and Integrity
Receiver Decrypts Message Digest with Sender's Private Key
=
"Hashed" Plaintext message
Receiver Computes Message Digest (Hashing algorithm)
=
"Hashed" Plaintext message
If these are equal, then message is authentic and was not altered en route
Plaintext message
Plaintext message
Message Digest (Hashing algorithm)
+
Encrypt Message Digest with Sender's Private Key
Digitally Signed and Encrypted Message --> Confidentiality, Integrity, and Authenticity
Encrypt with Secret Key
Encrypted Message Digest
Plaintext Message
Receiver Decrypts with Secret Key
Receiver decrypts wtih Sender's Public Key
=
"Hashed" Plaintext message
Receiver Computes Message Digest (Hashing algorithm)
=
"Hashed" Plaintext message
If these are equal, then message is authentic and was not altered en route
Plaintext message

Shaded boxes depict the "bits" actually transmitted across the network

Figure 10. Data Protection Schemes

H. J2EE AND MICROSOFT .NET

As discussed in Chapter Two, platform-independent computing based on Internet standards and protocols is becoming a reality. The technology is maturing but due to the complexity of integrating and web-enabling an entire enterprise, there is no single-source solution. With the emergence of Web services, application vendors are yielding to market pressure for platform independent solutions. These companies are experimenting with a new business model viewing integration software as a service vice a product, with the distinction between competitors related to the developmental tools support services.

The Web services solutions can be divided into two primary categories; Microsoft, and everyone else. However, the non-Microsoft companies all employ some form of the J2EE architecture. The competition, in this case, is related to buying to a totally integrated package of development tools from Microsoft at the expense of vendor lock-in, as opposed to going for flexibility of an open-source solution at the expense of total integration. Although the preceding sentence doesn't sound much different than the "open source versus Windows" argument, the key here is that the Microsoft.NET solution provides multi-platform compatibility based on Internet standards. This means that enterprise integration solutions based on Web services will be able to communicate regardless of the computing platform. The following discussion describes the details of the two architectures along with a comparative analysis of each to assist in choosing a solution for the long-term viability of FOSSAC.

1. J2EE Framework

Java 2 Platform Enterprise Edition architecture is the result of an industry-wide initiative lead by Sun Microsystems. J2EE is a set of standards using the development tools of the Java programming language. This architecture encompasses the Java Virtual Machine (JVM) technology allowing compiled Java programs run unaltered on various machine architectures (CPU instruction sets) as well as the tools to compile, analyze, debug, and deploy Java programs. Sun Microsystems over the last few years has reorganized the Java platform into three profiles:

- The Java 2 Platform, Micro Edition (J2ME), for handheld and other lower-end devices
- The Java 2 Platform, Standard Edition (J2SE), targeted at desktop machines
- The Java 2 Platform, Enterprise Edition (J2EE), installed on servers and responsible for enterprise solutions [Ref. 14]

A common misconception about J2EE is that it is a software product. As stated previously, J2EE is a set of development standards describing agreements between applications and the servers on which they run. J2EE is actually distributed as a set of Adobe Acrobat PDF files.

Sun's marketing strategy is to give customers a choice of products and tools, and to encourage best-of-breed products to emerge through competition. Sun Microsystems has long been recognized as a leader in enterprise computing solutions and companies typically purchased Sun hardware to run the software applications. The only way this could happen is if the industry as a whole were bought-into J2EE. To secure buy-in, Sun collaborated with other vendors of eBusiness solutions, such as BEA, IBM, and Oracle, in defining J2EE. To solicit new ideas and continuously improve J2EE, Sun initiated the Java Community Process (JCP). This community is an open organization of international Java developers and licensees whose charter is to develop and revise Java technology specifications. [Ref.15]

a. Framework and APIs

J2EE is an extension of J2SE, taking advantage of existing J2SE Application Programming Interface (API) services and multiple application program modeling tools. These tools help developers integrate the application framework providing security, scalability, and maintainability. Below is a listing of the APIs that make up the J2EE framework.

- Enterprise JavaBeans (EJB) 2.0
- Java Transaction API (JTA) 1.2
- J2EE Connector Architecture 1.0
- Java Messaging Service (JMS) 1.0.2
- Java Authentication and Authorization Service (JAAS) 1.0
- JDBC (for database connectivity 2.0)
- Java Name and Directory Interface (JNDI) 1.2
- Java Mail 1.1
- Servlet 2.3
- Java RMI 1.0
- Java API and XML Parsing (JAX) 1.1

The J2EE Runtime Environment (JRE) defines four component types that a product must support. The component types are the application container, applet container, Web container, and enterprise bean container. A container provides the runtime support for JSEE application components. Within each container the standard services (APIs) reside. Figure 11 depicts these containers and their relationship to each other. The arrows represent required access to other parts of the J2EE platform.

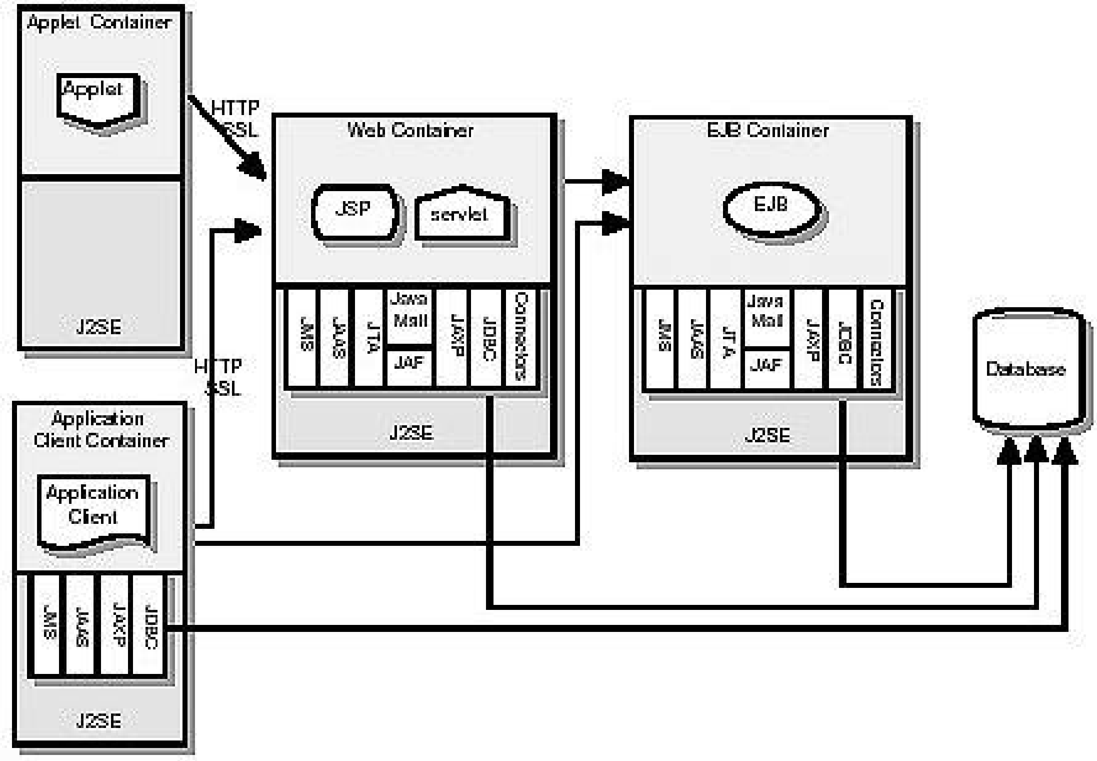


Figure 11. J2EE Framework (After Ref.14)

b. Developer Tools

Since Java is an open source programming language, there are hundreds of Java-related initiatives accessible to the developers in this community. Depending on the task, application developers have a variety of tools to choose from. However, not all these tools conform to the strict guidelines of the J2EE standard, limiting the portability of the applications developed with these tools.

Development tools from industry leaders like Sun, IBM, BEA, and Oracle, have an incentive to follow the J2EE standard since these companies are part of the Java

Community Process. The IBM WebSphere Studio Application Developer is the development tool marketed by IBM. This is an example of an integrated solution that conforms to the J2EE framework. This tool provides an integrated, Web application development environment capable of supporting building, testing, and deploying the J2EE framework. Other application development tools with similar capabilities are the Oracle 9i Application Server, and the BEA WebLogic Application Server.

2. Microsoft .NET Framework

NET is both a business strategy from Microsoft and a programming model that enables developers to build Web-based applications, smart client applications, and XML Web services. The functionality of these applications is accessed over a network using standard protocols such as SOAP and HTTP. The .NET framework manages much of the underlying connectivity, allowing developers to focus on writing the business logic code for their applications.[Ref. 8]

Microsoft.NET evolved from a previous platform called the Distributed interNet Application Architecture (Microsoft DNA). The DNA platform, introduced in January 1999, was Microsoft's platform for enabling modern, scalable, multi-tier business applications for delivery over a network.

The heart of Windows DNA is the integration of Web and client/server application development models through the Component Object Model (COM) and Distributed COM (DCOM). The .NET framework replaces these proprietary (Microsoft) technologies with a Web services based framework. The goal of the Microsoft .NET framework is to make it easy to build XML Web services and applications, but it also has a dramatic effect on every kind of application, from simple client applications to many kinds of distributed applications.

a. Framework and Components

The .NET framework requires a Windows-based computing platform for the servers and the computers running the .NET developer tools. Using the Web services standard protocols, services developed using .NET framework will be accessible to non-Windows computers. The .NET Framework consists of three main parts: the common language runtime, a hierarchical set of unified class libraries, and a component-based version of Microsoft Active Server Pages called Microsoft ASP.NET. [Ref. 8] Together they provide developers a way to create a set of tools and technologies that build complex applications. Like J2EE, it will integrate all facets of the application framework to build requirements into enterprise systems that provide security, scalability, and maintainability. The Common Language Runtime (CLR) consists of the complier, memory manager, and security features. The CLR manages the execution of code and provides services that make the development process easier. The unified classes of the framework consist of Web classes (ASP.NET), Data (ADO.NET), Windows Forms, and XML. The unified programming class provides developers with a unified, object-oriented, hierarchical, and extensible set of class libraries (APIs), which enables cross-language inheritance, error handling, and debugging. ASP.NET consists of Web applications, Web Services, and runtime and infrastructure. ASP.NET builds on the programming classes of the .NET Framework, providing a Web application model with a set of controls and infrastructure that make it simple to build ASP Web applications. Figure 12 depicts the framework relationship.

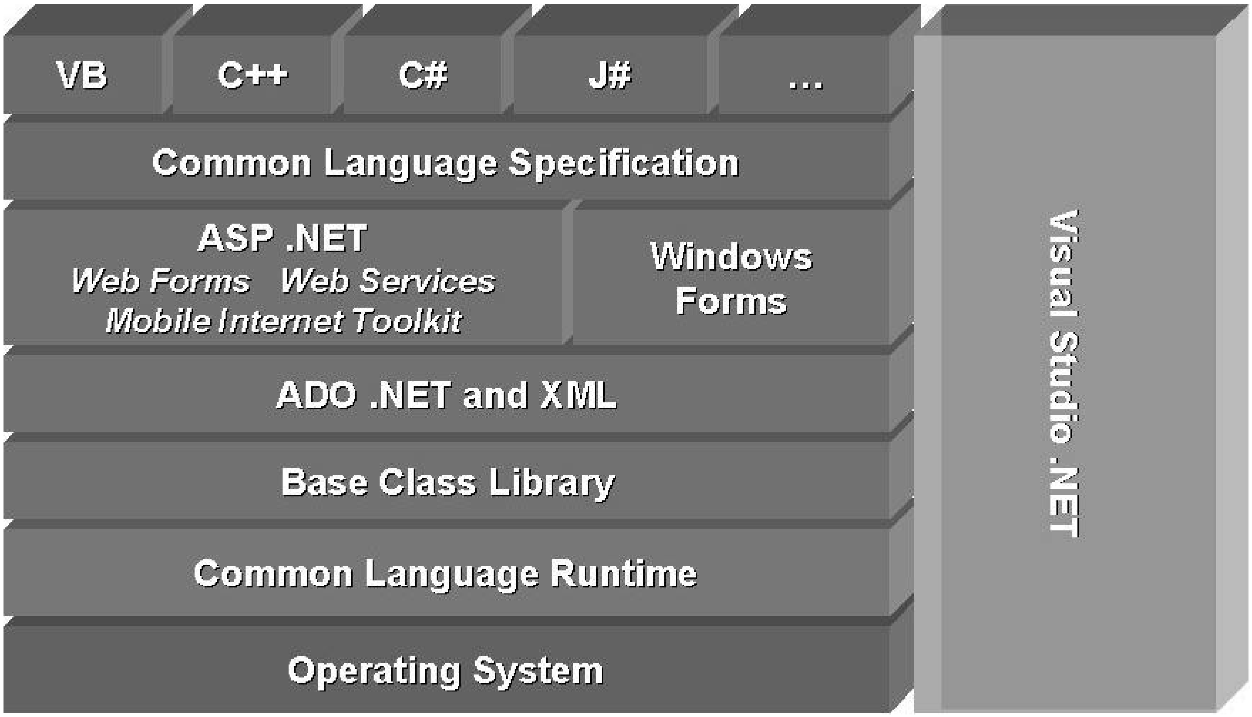


Figure 12. .NET Framework (After: Ref.8)

b. Developer Tools

Visual Studio .NET (Figure 12) is the integrated development environment for .NET. From a developers standpoint it is the comprehensive tool set for rapidly building and integrating XML Web services, Microsoft Windows–based applications, and Web solutions. This allows for applications to share data over the Internet, XML Web services enable developers to assemble applications from new and existing code, regardless of platform, programming language, or object model. Finally, the single, shared Visual Studio .NET integrated development environment (IDE) and a choice of programming languages—including Microsoft Visual Basic, Microsoft Visual C++, and Microsoft Visual C#—allow developers to build applications quickly. [Ref.16]

3. Analogies and Comparisons

a. Analogies

J2EE and .NET solve common issues developers face when building networked applications and architecting a system. Both technologies provide an application and development and deployment platform, combining an object-oriented language with an application-execution component and offer features that support similar functions. Table 3 depicts some of these analogies.

Feature	J2EE	.NET
Type of technology	Standard	Product
Middleware Vendors	30+	Microsoft
Interpreter	JRE (Java Runtime Environment)	CLR (Common Language Runtime)
Dynamic Web Pages	JSP (JavaServer Pages)	ASP. NET
Middle-tier components	EJB (Enterprise JavaBeans)	.NET managed components
Database access	JDBC SQL / J (Java Database Connectivity)	ADO.NET (Active Data Objects)
SOAP, WSDL, UDDI	Yes	Yes
Implicit middleware (load-balancing, etc)	Yes	Yes

Table 3. Analogies between J2EE and .NET technologies (After: Ref. 17)

b. Comparative Analysis

The following table lists various criteria that can be used in differentiating these frameworks. Table 4 is an overview of the J2EE and .NET frameworks.

Criteria	J2EE Framework	.NET Framework
Fundamental Design and Support of Web Services	Support of Web services through a JSP	Web services support is an integral part of the .NET product
Implementations	Implemented through EJBs	Implemented through .NET managed components
Pricing	Expensive as compared to MS .NET, however, if already vested in a J2EE based application, it makes more sense to maintain existing infrastructure	Less expensive than J2EE-based application servers. However, J2EE stil recognized as leader in enterprise-wide solutions
Tools and Servers	There are multiple companies that have built IDEs and application servers based on J2EE. A majority of these companies have already started supporting Web Services creation, deployment and execution within their products. The level of sophistication and support for Web Services standards differs from product to product.	Microsoft's cornerstone IDE for Web Services is Visual Studio .NET. Microsoft Web services are implemented through BizTalk 2002 Server an SQL Server 2000
Promotion Strategy	Multiple, independent companies including IBM, BEA Systems, Oracle, HP, Sun Microsystems are offering support for Web Services in their J2EE-based development tools and application servers. This broad base of development support promotes availability fo "best of breed" tools.	Microsoft promotes a product providing an all-in-one package o interoperable development tools. Support of Web services standards wil allow multi-platform compatibility.
Maturity of Platform	J2EE has proven to be a robust, scalable and a mature platform over the last four years. Addition of support for Web Services is just another feature for this platform.	New product not proven as a serious enterprise-wide solution. However, i maintains familiar development tools like Visual Basic.
Single Vendor Solution	Support from industry leaders such as IBM, Oracle, BEA, and HP, has spawned wide variety of open source tools, products, and applications. Integrated solutions are available as well as the ability to combine individual solutions. However, J2EE tools are not always completely portable between vendors and can limit the ability to mix and match tools without experienced intervention	As the basis of the .NET promotion strategy, Microsoft provides a fairl complete solution. Microsoft lack some of the higher end features that J2E solutions offer like (e-business XML (ebXML). However, with a lack o true standardization in J2EE APIs for Web services, Microsoft.NET has a advantage as an integrated product.
Portability	J2EE will run on a variety of hardware and operating systems. This portability is due to the fact that the Java Runtime Environment (JRE), on which J2EE is based, is available for any platform. However, this portability is only guaranteed if tools are developed in strict compliance to the J2EE standards. To mitigate the compatibility risk, Sun has created a J2EE compatibility test suite.	The .NET product only runs on Windows-based operating systems and i not portable. Microsoft did not intend portability only compatibility wit adherence to Web services standards.
Language Support	J2EE promotes Java-centric computing, and as such all components deployed into a J2EE deployment (such as EJB components and servlets) must be written in the Java language. To use J2EE, you must commit to coding at least some of your eBusiness systems using the Java programming language.	.NET supports development in any language that Microsoft's tools suppor due to the new CLR. With the exception of Java, all major languages will b supported. To facilitate programming in the .NET environment, Microso recently introduced its new C# language. This language is very similar t Java in terms of syntax, design and runtime behavior.
Web services support	J2EE supports web services through the Java API for XML Parsing (JAXP). This API allows developers to perform any web service operation today through manually parsing XML documents. A variety of J2EE-compatible 3rd party tools are available today that enable rapid development of web services.	Web service support is an integral part of the .NET programmin environment. The tools that ship with Microsoft.NET provide rapi application development of web services, with automatic generation of we service wrappers to existing systems.

Table 4. Comparative Analysis (After Ref.18)

I. SYSTEMS UNDER CONSIDERATION

There are three commercial solutions which merit further discussion as a result of this research. The first potential recommendation is the IBM WebSphere Application Server, the second is the BEA Weblogic Application Server, and last is the Microsoft BizTalk Server. The discriminator for choosing these products for evaluation was the market leadership and maturity. IBM and BEA command 31% and 34% respectively and Microsoft, while not a leader in Web services/enterprise integration, is a market leader in operating systems and is seen as a credible of Web service as a integration method.[Ref.19]

1. IBM WebSphere Application Server Version 4.0, Advanced Single Server Edition

IBM WebSphere Application Server is a component of the WebSphere Software platform for e-business. The WebSphere Software platform provides a single solution to support enterprise computing and integration requirements. However, the focus is on the WebSphere Application Server, which is a Java based technology providing integrated support for key Web services, open standards and full J2EE certification. The server is scalable and provides integration for databases, legacy systems, and message exchanging applications. The server will also support a variety of operating environments that include Windows NT and 2000, Sun Solaris, HP-UX, and Linux to name a few. [Ref.20] Depending on the size of the business, three editions (Standard, Advanced, and Enterprise) of IBM WebSphere AS are offered. Below is a listing of services included in the IBM WebSphere AS (Advanced Single) edition:

- J2EE 1.2 compliance with some J2EE 1.3 support (enhanced collaboration tools)
- Messaging services through MQ series Server
- Web services support (SOAP, UDDI, XML, WDSL)
- Multi-platform support (CORBA, COM+, and ActiveX compatibility)
- Robust administrative managing and monitoring system (requires adding IBM DB2fetures)
- Includes an Apache-based Web server
- Security controls (user/group policies and support for third-party authentication techniques)
- Lotus Domino interoperability (enhances distributed content authoring)
- Mapping tools for importing competitor offerings (BEA to IBM)
- EJBs allow method-level security Websphere allowed grant or deny privileges to groups or users [Ref.21]

This server appeals to departments and medium businesses which require a lower cost, fast to get running option that does not require the redundancy, workload distribution, or remote administration associated with multi-sever management.

WebSphere excels when an application requires industrial-strength transaction management, significant scalability, or where business logic is completely encapsulated in distributed components such as servlets or Enterprise Java Beans. IBM WebSphere is an “out of the box’ solution for application integration that doesn’t require any additional components.

2. BEA Weblogic Application Server, Version 6.1

BEA Weblogic Application Server is a component of the BEA Weblogic Integration. The BEA Weblogic Integration provides a single solution to support enterprise computing and integration requirements. The focus is on the BEA Weblogic Application Server, which is a Java-based Web application server provides integrated support for key Web services, open standards, and full J2EE certification. The Weblogic Server consists of three logical supporting application integration, business process management, and B2B standards for integrating applications and enterprises over the Internet. The first layer is the Weblogic Server Web container, which handles client-side presentation logic for browser applications via HTML, XML and Java Server Pages (JSPs). The second layer provides Java components to encapsulate the business logic. The third layer supplies information access using J2EE database, connector and graphical interface services. The Weblogic server supports Unix, Linux, Windows, and mainframe operating systems.[Ref.22] It also integrates with relational databases, message exchanging applications and legacy systems. More importantly, it leverages the J2EE framework providing a set of utilities in support of the following:

- Full J2EE 1.3 compatibility (includes Java Messaging Standard compliance)
- Multi-platform support (CORBA, COM+, and ActiveX compatibility)
- Java adapter for Mainframe computer support
- Enterprise Messaging capabilities
- Third party compatibility with WebLogic Console (light weight management and monitoring)
- Security controls (user/group policies and support for third-party authentication techniques)
- Support for third party development tools (Borland, WebGain, Macromedia)
- Mapping tools for importing competitor offerings (IBM to BEA) [Ref.23]

This server also appeals to small/medium businesses whether developing and deploying new applications, hosting existing applications, or preparing for the future of Web services and distributed computing. The BEA Weblogic Server is an “out of the box “solution that doesn’t require any additional components. Its features support compliance with open standards, multi-tiered architecture; Web services standards (SOAP, WSDL, and UDDI), and support of component-based development.

3. Microsoft BizTalk Server, Standard Edition

Microsoft BizTalk Server is component of the Microsoft .NET Enterprise Server family of products. The BizTalk server is an integral part of Microsoft's move to support Web services with a core built around the Web services protocols (SOAP, HTTP and XML). However, BizTalk is not an out-of-the-box Web services solution. The BizTalk server is described as a manager for Web services; it is more of a flow control and error-tracking manager for XML messaging.[Ref.24]

Microsoft servers are not typically thought of as enterprise application servers by the classical definition, although limited application server functionality is integral to the operating system. Typical application servers are characterized by core transaction management, database access, and business logic functionality. In order to achieve this same functionality and implement Web services at the enterprise level, the BizTalk server must be combined with the Microsoft Host Integration Server, the SQL server and the Application Center 2000. Combined with the .NET framework, the .NET Visual Studio and the Windows 2000 Advanced Server operating system, Microsoft has been able to make inroads as a competitive offering in the EAI/Web services arena.

Below is a listing of some characteristics of the Microsoft's Web services implementation (BizTalk Server 2002 Standard):

- Microsoft Windows-based hardware only
- Proprietary but tightly integrated with COM and DCOM object managers
- Integration with any application or technology
- Support for industry standards (SOAP, XML)
- Reliable document delivery via Message Queuing Server (asynchronous messaging)
- Secure document exchange [Ref.25]

Each of the three products described above have their own strengths and weaknesses. Chapter Five will discuss the products in more detail to suggest the best alternative for FOSSAC.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. COPING WITH THE NMCI TRANSITION

Analysis of human response to technological change has been observed and well documented over the past century. These observations have resulted in several models and methods for implementing and managing organizational change. Organizational change occurs in many forms from minor transitions to transformations and upheavals. Effectively managing change involves different activities depending on the scope of change and the organization's readiness for it. This chapter will discuss techniques for framing a transition strategy, issues for consideration by those leading the change process and the changes taking place at FOSSAC with the implementation of NMCI.

A. RESISTANCE TO CHANGE

In preparing an organization for transition from one state to another, leaders must not underestimate human resistance to change. There are several reasons why individuals resist change, some of which may not be well correlated to the actual change taking place. The resistance by employees typically comes with perceived feeling of losing something. Rational or not, the feeling of loss is often related to one or more of the following factors: [Ref. 26]

- Security – changes in the size of the workforce as the result of "rightsizing" or automating certain processes
- Money – reductions in pay or benefits
- Pride and Satisfaction – reduction in required skill set to perform job, lack of recognition for specialized abilities and a lack of fulfillment from job requirements
- Friends and important contacts – reduced social satisfaction resulting from relocations or reduction in force
- Freedom – increased supervision or less opportunity to make decisions
- Responsibility – closely related to 'pride and satisfaction";
- Authority – loss of optional power from restructuring the organization
- Working conditions – reduction in comfort or physical space often as a result of consolidation

- Status – changes in job title or recognition

Employees are more often satisfied to remain in their comfort zone, questioning the need for any change at all. The employees have difficulty viewing the change objectively; they are preoccupied trying to quantify the impact of the above factors. In rare cases, employees may feel the change is overdue. However, if the change is not what they expected, even those welcoming the change may feel as threatened as those resisting the change.

B. METHODS AND TECHNIQUES

In trying to analyze and predict the effect of organizational change, models provide a starting point. The models help by providing a framework to organize and group the various techniques. One model described by Donald Kirkpatrick consists of seven steps:

- Determine the need for change
- Preparing a tentative plan
- Analyzing probable reaction
- Making a final decision
- Establishing a timetable
- Communicating the change
- Implementing the change [Ref. 26]

This model stresses empathy, communication and participation. Empathy is determining to what extent the change will be accepted or rejected. Communication is more than just informing; it must create understanding. Participation means involvement from those affected by the change

Two other models, one by William Bridges and the other by Kurt Lewin are similar in breaking the transformation process into three steps. The Bridges Model describes the process as Letting Go, the Neutral Zone, and New Beginnings. This roughly parallels the Lewin Model of Unfreeze, Change, and Refreezing.

At a conceptual level, the change problem is a matter of moving from one state to another state. The move is typically accomplished as a result of setting up and achieving three types of goals: *transform*, *reduce*, and *apply*.

- Transform goals are concerned with identifying differences between two states.
- Reduce goals are concerned with determining ways of eliminating these differences.
- Apply goals are concerned with putting into play operators that actually effect the elimination of these differences. [Ref.27]

C. TRANSITION VERSUS CHANGE

The difference between transition and change may appear to be semantics. However, some experts in this field differentiate the two, "Change often starts with a new beginning, but transition must start with an ending – with people letting go of old attitudes and behaviors. The organization will most likely gain from change but the process begins with a feeling of loss".[Ref.28]

The following is a closer look at the transition model described by Bridges – Letting go, the Neutral Zone, and New Beginnings.

- During the Letting go phase, employees need to be allowed to grieve and be acknowledged for their feelings of loss. Leaders need to consider ways to compensate employees and to publicly express their own feelings of loss. Depending on the magnitude of change, employees should be allowed to take a piece of the past with them – perhaps a title, responsibility or status from the pre-change environment. While acknowledging the feeling of loss for the employees, the leaders must act decisively and be able to articulate how the change will benefit the organization. Leaders should seek out "champions of change" – those employees who understand the process and the benefits of the change. The advocates will be critical and the transition enters the second phase; the Neutral Zone.

- The Neutral Zone is the core of the transition process. In his book, Stuart Klein describes this process.

    - "The communications strategy during this phase should have three primary objectives. The first is to provide those who initially are not directly involved with the change with detailed and accurate information of what is happening. Second, those not currently involved should be aware of how they will become engaged in the future; how the change will affect them, their new roles and responsibilities. Third, to challenge whatever misinformation is circulating about the change. This is the time to strengthen intra-group connections and mark the accomplishment of short-tern goal accomplishment." [Ref.29]

    During this period, the employees can become easily overloaded or confused because the organization is in flux. Rumors and misconceptions can generate considerable anxiety. There is the potential for employees to become polarized – those who want to rush forward with the change and those who want to return to the security of the old way. Leaders need to be empathetic, validating the feelings of those who are afraid. This period can also be a time marked by innovation and creativity. Leaders need to recognize these people perhaps with public recognition to gain momentum for the change process. [Ref.29] The ultimate goal in this phase is to reduce confusion through education and communication.

- The final phase, The New Beginning is analogous to refreezing. During this phase, leaders must institutionalize the change and publicize its success. Communication, supervision and feedback from lower levels is essential in advancing the organization's new identity [Ref.30]

D. TRANSITION AT FOSSAC

The ability to predict human response at FOSSAC to the implementation of the Navy Marine Corps Intranet may not be known for certain. However, with the preceding discussion about how change can affect an organization and the techniques available to reduce chaos, leaders have the ability to effectively manage and to some degree control the change. The references used in support of this thesis are a small sample of the research and case studies available describing successful (and unsuccessful) applications of organizational change techniques.

FOSSAC like many other organizations is dependent on information technology to accomplish daily tasks. Because the organization operates in a dynamic environment, occasionally, it must acquire new software applications to accomplish its mission. These applications are often for a specific task and may be purchased from a commercial vendor or developed internally. As the organization expanded, the software became an integral part mission accomplishment. Many of these software applications are considered legacy systems and will be excluded from the desktop operating environment managed by the NMCI contract. This exclusion, combined with a general lack of communication is a significant factor in the current perception of how NMCI will benefit FOSSAC.

As discussed previously, a communication strategy must be adopted to get the word out to all members of the organization. In the day-to-day operations, it is easy to overlook or misunderstand the effect of change on employees. Organizational leaders are typically the first to know of impending changes and tend to focus on the logistics of change as opposed to the psychology of change.

FOSSAC is somewhat unique in that it has two distinct groups working side-by-side. FOSSAC has a military presence, which includes a Commanding Officer, Executive Officer, and approximately twenty-five military personnel. The other group consists of approximately two- hundred civilian employees who are guided by detailed job descriptions. Approximately ten percent of the civilian employees are in excess of the "Full-Time Equivalent" (FTE) authorization. The FTE refers to the number of "permanent" personnel whose jobs are protected as part of the minimum required for FOSSAC to perform its mission. The non-FTE personnel are the ones most likely to fear the change. This fear has generated conversation among the employees at FOSSAC regarding the perceived effect of the NMCI implementation.[Ref.31] Intentionally or not, this conversation had caused a general uneasiness among the FTE employees and their ability to perform their duties under the new system.

NMCI is a high-visibility undertaking with initial implementations being watched closely. Unfortunately, there have been two NMCI implementations that have gained negative publicity and has circulated among FOSSAC employees. Situations like this require intervention and "damage control" to portray these incidents as anomalies. If these incidents are perceived as typical, the change proponents at FOSSAC will lose precious momentum in controlling fear among the employees.

One of the issues for FOSSAC is that like other DOD organizations, the change was imposed by commanders several layers above FOSSAC. The commander responsible for managing the change at the local level was not made fully aware of the transition management plan, let alone the technical details of the change. Additionally, these upper command echelons were unable to grasp the potential affect on the end-users. Imposing changes on the scale of the NMCI, are difficult to manage in large bureaucratic organizations. The change management process used to implement NMCI is an example of how NOT to initiate a technological change.

A fortunate (or unfortunate) consequence of the NMCI implementation is that in order for FOSSAC to remain a viable business, they must have access to many of the legacy software applications not certified for use under NMCI. Delays in the NMCI certification process (certification defined in Chapter 3) resulted in FOSSAC having to maintain a separate, parallel, legacy network. This parallel network will have the effect of delaying the change to the new system since the old system will still be in operation. Originally, the NMCI implementation was to be a "turn-key" event; the old network (non-NMCI compatible hardware and applications) would cease to operate the instant the NMCI network was turned on. The delay with NMCI has effectively allowed employees to hold on the past. A beneficial consequence of the delay is that FOSSAC gained additional time to manage the transition. This may reduce some of the anxiety and chaos associated with the eventual departure of the old network and the legacy applications tied to it. Having access to both the old and the new networks simultaneously, the local commander has regained time to influence the transition using the techniques described in this chapter.

V. CONCLUSIONS AND RECOMMENDATIONS

In Chapter Two, there was a discussion about the technology available to integrate the business processes in an organization. In Chapter Three, there was a discussion of NMCI and the commercial vendors offering integration solutions with the potential to aid FOSSAC with their integration challenges. Two of the proposed solutions were Java/J2EE standards based and the other was a Microsoft, proprietary solution. This discussion will focus not so much on which of the three products to choose, but more on the differences between these products and the recommended criteria guide FOSSAC in making a choice.

A. RESEARCH QUESTIONS REVISITED

1. With the current dependency on legacy applications, will the NMCI infrastructure adequately support the business processes currently in use at FOSSAC?

The answer to this question is a qualified "yes". In accordance with the NMCI contract, the ISF is to provide maintenance and support for the legacy LAN (quarantined LAN) while the organization transitions to NMCI-compatible applications. However, because of FOSSAC's dependence on these applications, the decision was made by the FOSSAC IT staff to take over maintenance because the eight-hour response time was insufficient to support the business processes.

2. Do current and accepted Enterprise Architecture Integration (EAI) methods adequately define a transition strategy for FOSSAC?

The answer here is yes. The methods outlined in Chapter Two, in particular, the data integration model and the Web Services model describe a suitable integration methodology for FOSSAC.

3. Are their any other DoD organizations providing similar services and how does NMCI affect their technology strategy?

In the conduct of this research, no other DoD activities were found providing the unique mix of quality services provided by FOSSAC and also subject to the NMCI constraints.

4. Do existing Commercial/Government Off The Shelf (COTS/GOTS) software applications provide acceptable integration of legacy applications?

The answer here is a qualified yes. There are many application integration vendors offering Business Process Management Systems. The best path for FOSSAC to take is to continue the internally initiated process of porting non-NMCI compliant applications to run in the NMCI environment. The IT personnel at FOSSAC are highly skilled and have been successful (with the help of the ISF) in implementing an interoperability plan. This plan provides access from the NMCI LAN to the applications and data on the legacy LAN while simultaneously migrating data and applications to the NMCI LAN.

5. How does the NMCI infrastructure affect the implementation of any recommended solutions?

The standardization and rules imposed under NMCI provide strong motivation for application developers to focus on web-enabled, server based, platform-independent applications that conform to the Web Services protocols (SOAP, XML, UDDI, WDSL).

B. ISSUES ON IMPLEMENTING EAI

FOSSAC's current environment (on the legacy LAN) employs "thick clients" meaning that most of the software application functionality resides on the client (desktop) computers. The Novell NetWare server handles the network communication and administration including access to a Windows NT/SQL server for database management. FOSSAC has been using Windows-based client computers and has already initiated action to rewrite some of its local legacy applications to be compatible or at least accessible from the NMCI environment. The internal actions at FOSSAC appear to be reducing, possibly eliminating the need for any proprietary EAI solution. If one is needed, it will most likely involve some low-level mapping of the data structures (data integration EAI methodology) to allow access to legacy data from the NMCI environment. Unfortunately, in the conduct of this research, there was insufficient access to the detailed data and application functionality at FOSSAC to permit recommending a specific EAI solution. However, looking toward the future and the constraints imposed by the NMCI environment there is the potential for implementing a Web Services solution.

C. ISSUES ON IMPLEMENTING WEB SERVICES

When considering solutions for an organization like FOSSAC, with its relatively small size and limited budget, there is always the possibility that doing nothing is a realistic option. However, based on this research, the best option is to begin implementing Web Services as they continue to do their own internal application integration. Web Services are rapidly gaining acceptance as a viable integration methodology, while providing operating system and platform independence. Rapid acceptance of Web Services among commercial organizations will accelerate the maturation of the Web Services standards, as well as provide a base of skilled Web Services developers. The following is a discussion of details that must be addressed in deciding on a Web Services implementation.

1. Security and Authentication

"Of all the objections to Web Services, these two [security and authentication] get raised earliest and most often." [Ref. 33] Confidential information is vulnerable to hackers or malicious employees and databases managed by Web Services are often unencrypted, making for easy targets. Fortunately, when dealing with sensitive data, Secure Socket Layer (SSL), a protocol using public/private key encryption, works to prevent interception of XML messages. However, authenticating XML documents on the server is still an issue. The critical problem is how to solve the security problems without affecting the underlying Web Services architecture. Draft protocols for XML encryption, key management, and signatures have been submitted to the World Wide Web Consortium (W3C), the standards body, and are under review. [Ref.34] On June 22, 2002, the Organization for the Advancement of Structured Information Standard

(OASIS) received the latest version of the Web Services Security (WSS) specification submitted by IBM, Microsoft, and VeriSign. The WSS specification is a standard to support and integrate multiple security models and technologies. The standard provides for a range of different systems to operate in a platform-independent, language-neutral manner. [Ref. 35] Along with Microsoft and IBM, BEA Systems, Cisco, Sun and other Web Services companies are in support of the WSS. The immaturity of Web Services has made potential users cautious; this industry-wide support should provide for a more rapid maturation and acceptance of Web Services.

Until the security issues are resolved, regardless of a specific vendor choice, all transactions should be authenticated with digital signatures; both intra-FOSSAC and between FOSSAC's DoD customers and commercial suppliers.

2. Computing Platform

As mentioned in Chapter Three, the J2EE based solutions will run on a variety of server operating systems to include Windows. Microsoft.NET will only run on Windows-based servers and clients. Under the NMCI contract, all servers and clients will be Windows based, therefore, the computing platform is not a discriminator for choosing either the J2EE or Microsoft implementation.

3. Cost to Implement

Evaluating these products from a cost perspective is difficult. The systems will require a minimum cost to have advertised functionality on a single CPU. However, comparative analysis of each offering is dependent on the *actual* needs of the organization. Additionally, it is not known where the "cost spikes" occur when the number of users or functional capabilities require follow-on scalability investments. Knowledge of FOSSAC's operational needs is not known with enough detail to make a determination based on cost. What is known is that the initial operational cost for the BEA Weblogic server is approximately $57,000 per CPU, offering stand-alone, and out-of-the-box capability. The IBM WebSphere requires $64,000 for a similar implementation of Web Services. Microsoft can provide Web Services for $6000 but this cost may be several thousands dollars higher depending on the number of Client Access Licenses (CALs) and development tools licenses.

In the case of FOSSAC, the high price of the BEA and IBM solutions begs the question of what comes with the extra up-front costs (as opposed to the Microsoft solution) and is it necessary? Additionally, from a cost perspective, one must consider the cost to develop applications and maintain the systems. Microsoft advertises that the Visual Studio.NET is similar to previous development tools, promising cost savings based on higher productivity in a shorter period from developers already familiar with the Microsoft environment.[Ref.20]

All of these offerings will have recurring costs associated with upgrades and license refresh rates, particularly with the Microsoft option. Additionally, with Microsoft, one must consider the intangible cost of vendor lock-in. The prices quoted above are only an estimate and long-tern costs are difficult to calculate without configuring for a specific purpose and anticipated growth. These prices come from corporate sales literature and attempt to show each as the best offering at the lowest price.

4. Maintenance and optimization

One of the big steps forward with web browsers and the Java Virtual Machine was the ability to "smarten up" the client without installing additional software on it. This allows development and administration to take place at the server vice loading client-specific applications. This significantly reduces the maintenance requirements of the client computers.

Both IBM WebSphere Application Server and BEA WebLogic Application Server are leading, high-end application servers, and both adhere to the J2EE specification. However, they differ in the way they implement some features of the specification, and in their feature extensions. These differences can make it difficult to migrate enterprise applications from one application server to another. The migration encompasses more than just installing new software and reinstalling applications. It also involves issues such as education, skills, code, run time, deployment, tooling, etc. While FOSSAC does not employ any J2EE specific applications, should it choose a J2EE-based solution, it must consider the cost and availability of programmers to develop and maintain a J2EE-based implementation. [Ref. 32]

5. NMCI and Java

Under the NMCI contract "both ActiveX and Java Script are blocked at Boundary 1 (the interface to NIPRNET -- Non-Secure IP Router Network)" to prevent the potential of malicious code getting through the firewall.[Ref. 36]

The NMCI environment is not "java friendly", making J2EE-based Web Services difficult to implement. While the NMCI contract states that Java script is not allowed in or out at the Boundary 1 interface, movement of Java Servlets is allowed but restricted. "The firewall will only be configured to allow this [Java] protocol if there is a validated operational requirement for its use. These will not be set by default. This requires user(s) to have a web browser that supports restricting to trusted Web sites with Java only allowed at those trusted sites. The only authorized wild card in the trusted Web sites list is*.mil" [Ref. 36]

Figure 13 below illustrates a typical commercial implementation of Web services and how businesses and suppliers would typically interact across the Internet and across security boundaries. Since the FOSSAC implementation of Web Services would be an internal integration vice publishing for access by those outside the Boundary 1 firewall, there would be no need for any Java Servlets or Java Script to enter or leave the FOSSAC enclave. The implication here is that the J2EE based Web Services would be internal and thus from a "trusted" site. Firewall permissions would only have to be altered if FOSSAC was to make its Web Services available outside (across) the Boundary 1 firewall perimeter.

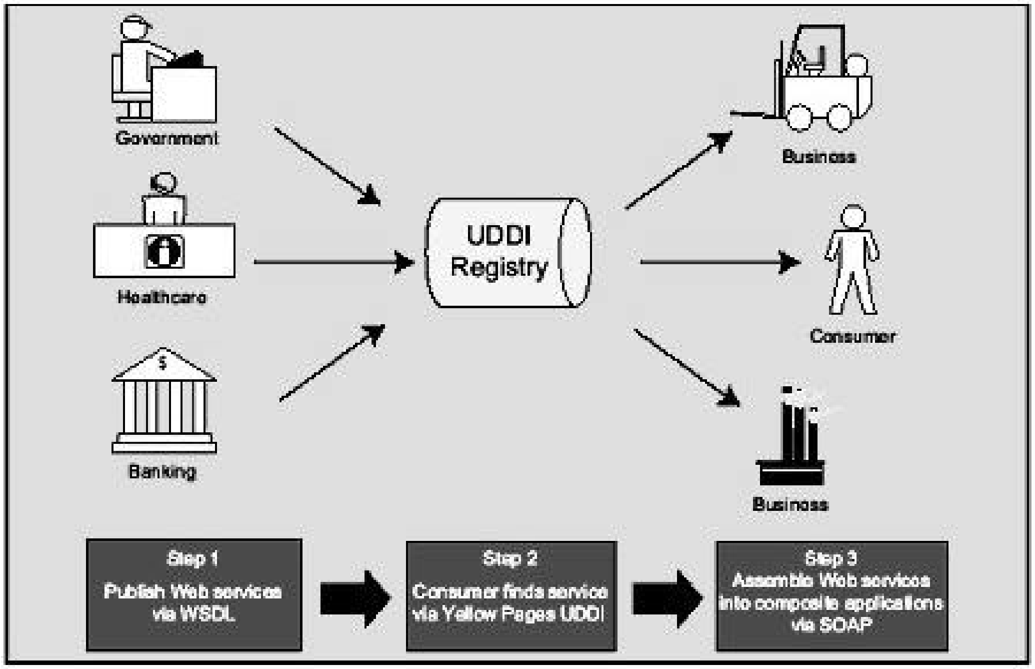

Government
Healthcare
Banking
UDDI Registry
Business
Consumer
Business
Step 1
Publish Web services via WSDL
Step 2
Consumer finds service via Yellow Pages UDDI
Step 3
Assemble Web services into composite applications via SOAP

Figure 13. Typical Commercial Implementation of Web Services

D. SUMMARY OF RESEARCH

Having looked at J2EE and .NET implementations of Web Services, there is not clear decision as to which version to implement. From a purely technical viewpoint, each method has advantages and disadvantages. The key advantage of using .NET approach is that it has been designed specifically for the purpose of implementing Web Services; J2EE has been retrofitted by the addition of APIs.

"One advantage of using J2EE as a base for your system is that you have a much wider choice of vendor for your pre-built software (application servers mostly), including numerous open source projects. In many ways, open source J2EE application servers are closer to the standard laid down by Sun, because they don't add proprietary extensions to overcome problems. Ultimately, unless you are starting your system from the bottom up, your choice of Web Services implementation is more than likely going to be influenced by your present system. If you have a team of skilled programmers, with an existing business system, realistically you'll want to continue using that system, be it J2EE based or Microsoft based." [Ref. 37]

Our recommendation is that FOSSAC implement the Microsoft-based Web services for the following reasons

- The primary skill set of the IT staff is Microsoft based. The developers are well versed in Microsoft Visual Basic for Applications and have started familiarizing themselves with the primary development tool for Microsoft Web Services (Visual Studio.NET)
- FOSSAC is a relatively small organization and does not have the funding to support the cost of a BEA Weblogic or an IBM WebSphere implementation.
- The NMCI environment is Java restrictive. Should FOSSAC want to offer its Web Services outside the Boundary 1 firewall, there are additional (and uncertain) fees associated with modifying the NMCI firewall.

APPENDIX A

GOLD DISK CONTENTS

SERVICE	SOFTWARE DESCRIPTION (MINIMUM VERSION)	VENDOR
Basic		
Operating System	MS Windows 2000 Build 2195 SP2/SRP1	Microsoft
Office Suite	Standard Office Automation Software Included on the Gold Disk • MS Word • MS Excel • MS PowerPoint • MS Access	Microsoft
Email Client	MS Outlook 2000	Microsoft
Internet Browser	Internet Explorer MS 5.5 SP-2 128bit	Microsoft
Virus Protection	Norton A/V Corp Edition v7.5	Symantec
PDF Viewer	Acrobat Reader v.5.05	Adobe
Terminal Emulator - Host (TN3270, VT100, X-Terminal)	Reflection 8.0.5 – Web Launch Utility	WRQ
Compression Tool	Winzip v.8.1	Winzip
Collaboration Tool	Net Meeting v3.01 (4.4.3385)	Microsoft
MultiMedia	RealPlayer 8 (6.0.9.450)	RealNetworks
MultiMedia	Windows Media Player v7.01.00.3055	Microsoft
Internet Browser	Communicator 4.76	Netscape
Electronic Records Mgmt	Trim Captura v4.3	Tower Software
Plug-ins		
Web Controls	MacroMedia Shockwave v 8.0	MacroMedia
Web Controls	Flash Player 5.0	MacroMedia
Web Controls	Apple Quicktime Movie and Audio Viewer v 5.0	Apple
Web Controls	IPIX v6,2,0,5	Internet Pictures
Security Apps		
Security	Intruder Alert v3.5	Axent
Security	ESM v5.1	Axent
Agents		
Software Management	Radia Client Connect v.2.1	Novadigm
Inventory, Remote control	Tivoli TMA v 3.71	IBM/Tivoli
Remote Connectivity (Notebooks)		
Dial-up connectivity	PAL v4.1.1.306	MCI/Worldcom
VPN	VPN Client v.3.0	Alcatel

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

1. Ruh, William A., *Enterprise Application Integration*, John Wiley & Sons, Inc., New York, New York. 2001.

2. Linthicum, David S., *Enterprise Application Integration*, Addison-Wesley, Boston, Massachusetts, 2000.

3. Buyens, Marc. "Enterprise Application Integration" September 2000, March 2002 http://eai.ittoolbox.com/documents/document.asp?i=571

4. Fingar, Peter, *Death of "e" and the Birth of the Real New Economy*, Meghan-Kiffer Press, July 2001.

5. Slater, Derek.. "Middleware Demystified", *CIO Magazine* 15 May 2000. 01 June 2002 http://www.cio.com/archive/051500_middle.html

6. Cimetiere, Jean-Christophe. "Web Services and EAI – Partners or Rivals", *Trendmarkers* June 2002. July 10, 2002 http://www.techmetrix.com/trendmarkers/publi.php?C=22O6O&currentLanguage=E

7. Vizard, Michael. "EAI is dead; long live Web services" *InfoWorld* 29 March 2002. 05 June 2002 http://www.infoworld.com/articles/hn/xml/02/03/29/020329hncapeclear.xml

8. Microsoft 09 July 2002. Retrieved 05 August 2002 from http://msdn.microsoft.com/netframework/productinfo/overview.asp

9. PerfectXML. "Web Services: Introduction" (n.d.). Retrieved 31 August 2002 from http://www.perfectxml.com/WebSvc1.asp

10. Vasudevan, Venu. "A Web Services Primer" O'Reilly xml.com 04 April 2001. 01 August 2002 http://www.xml.com/pub/a/2001/04/04/webservices/index.html?page=1

11. Dorobek, Christopher. "Navy refines process for legacy apps." *Federal Computer Week*, 05 August 2002, p.37.

12. Navy and Marine Corps Intranet (NMCI) Legacy Applictions Transition Guide, Version 4.0, 15 August 2002. NMCI Program Management Office – Technical Execution Division, PMW 164-4 Space and Naval Warfare Systems Command

13. Federal Telecommunications Recommendation 1024B-1998, Project Radio Equipment, 27 July 1998, 20 September 2000

http://216.239.53.100/search?q=cache:d0iZJbe0zKoC:www.ncs.gov/n6/content/standard/files/ftr1024b.pdf+%22Type+1+encryption%22&hl=en&ie=UTF-8

14. The Source of Java Technology. 16 September 2002. Retrieved 05 September 2002 from http://java.sun.com/

15. Java Community Press. (n.d). Retrieved 03 August 2002 from http://www.jcp.org/

16. Microsoft. 02 August 2002. Retrieved 25 August 2002 from http://www.microsoft.com/net/basics/

17. Vawter, Chad. "J2EE vs. Microsoft.NET - A comparison of building XML-based web" 15 July 2002. 01 August 2002 http://www.theserverside.com/resources/article.jsp?l=J2EE-vs-DOTNET

18. Samtani, Gunjan, "Web Services and Application Frameworks" Web Services Architect 27 March 2002. 15 May 2002 http://www.webservicesarchitect.com/content/articles/samtani04.asp

19. Berlind, David. "Sun's bundling move: hardball or wiffle-ball?" ZD Net 23 May 2002. 15 September 2002 http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2867661,00.html

20. IBM. 30 July 2002. Retrieved 05 August 2002 from http://www-3.ibm.com/software/webservers/appserv/

21. IBM. 05 August 2002. Retrieved 02 September2002 from http://www-900.ibm.com/websphere/library/bestpractices/ejb_migration_eng.shtml/

22. Hess, Deborah. "BEA Logic Server" ZD Net 20 December 2001. 01 September 2002 http://techrepublictk.cnet.com/enterprise/0-9816770-723-1617147.html

23. BEA.25 July 2002. Retrieved 11 August 2002 from www.bea.com/products/weblogic/server/index.html

24. Hankinson, Whitney. "Servers and Relationships under the .NET Infrastructure" *Web Services Architect*, 19 September. 15 August 2002 http://techrepublictk.cnet.com/enterprise/0-9816770-723-1617147.html

25. Microsoft. 10 August 2002. Retrieved 20 August 2002 from http://www.microsoft.com/biztalk/

26. Kirkpatrick, Donald L., *Managing Change Effectively*, Butterworth-Heinemann, Boston, Massachusetts, 2001.

27. Nicklos, Fred, “Change Management 101- A Primer” Change Management Resources Library 14 February 2002. 25 June 2002 http://home.att.net/~nickols/change.html

28. Bridges, William, *Surviving Corporate Transition*, William Bridges and Associates, Mill Valley, California, 1993.

29. Klein, Stuart M, *Journal of Organizational Change Management*, Emerald, England, 1996.

30. Bridges, William, *Managing Transitions: Making the Most of Change,* William Bridges and Associates, Mill Valley, California, 1992.

31. Logan, Brenda. Interview by author. Norfolk, Virginia, 15 January 2002.

32. IBM. 29 August 2002. Retrieved 15 September2002 from http://www7b.software.ibm.com/wsdd/library/techarticles/0209_searle/searle3.html

33. Aponovich, David. “Five Barriers to Implementing Web Services” IT Management 30 January 2002. 10 August 2002 http://itmanagement.earthweb.com/erp/article/0,,11981_965371,00.html

34. "OASIS Announces Technical Committee for Web Services Security", 23 July 2002. Retrieved 23 September 2002 from http://xml.coverpages.org/ni2002-07-23-a.html.

35. Scannell, E., "Power Trio pushes Web Services Security", 27 June 2002. Retrieved 23 September 2002 from http://www.infoworld.com/articles/hn/xml/02/06/27/020627hnwssecure.xml

36. EDS. 14 January 2002. Retrieved 18 September 2002 from http://www.eds-gov.com/nmcifaqs/nmcifaq.asp?f_cat=&f_keyword=Java&f_sort=F

37. Hanson, Jeffrey J. “.NET versus J2EE Web Services: A Comparison of Approaches”, *Web Services Architect*, 09 January 2002. 28 August 2002.

THIS PAGE INTENTIONALLY LEFT BLANK

BIBLIOGRAPHY

1. Aponovich, David. “Five Barriers to Implementing Web Services” IT Management 30 January 2002. 10 August 2002 http://itmanagement.earthweb.com/erp/article/0,,11981_965371,00.html

2. Berlind, David. "Sun's bundling move: hardball or wiffle-ball?", ZD Net 23 May 2002. 15 September 2002 http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2867661,00.html

3. Bridges, William, *Managing Transitions: Making the Most of Change*, William Bridges and Associates, Mill Valley, California, 1992.

4. Bridges, William, *Surviving Corporate Transition*, William Bridges and Associates, Mill Valley, California, 1993.

5. Buyens, Marc. “Enterprise Application Integration”, September 2000. March 2002 http://eai.ittoolbox.com/documents/document.asp?i=571

6. Cimetiere, Jean-Christophe. “Web Services and EAI – Partners or Rivals”, Trendmarkers June 2002. July 10, 2002 http://www.techmetrix.com/trendmarkers/publi.php?C=22O6O&currentLanguage=E

7. Dorobek, Christopher. “Navy refines process for legacy apps”, *Federal Computer Week*, 05 August 2002, p.37.

8. Fingar, Peter, *Death of "e" and the Birth of the Real New Economy*, Meghan-Kiffer Press, July 2001.

9. Hankinson, Whitney. "Servers and Relationships Under the .NET Infrastructure", Web Services Architect 19 September 2000. 15 August 2002 http://www.webservicesarchitect.com/content/articles/hankinson01.asp

10. Hanson, Jeffrey J. “.NET versus J2EE Web Services: A Comparison of Approaches”, Web Services Architect, 09 January 2002. 28 August 2002 http://www.webservicesarchitect.com/content/articles/hanson01.asp

11. Hess, Deborah. "BEA Logic Server", ZD Net 20 December 2001. 01 September 2002 http://techrepublictk.cnet.com/enterprise/0-9816770-723-1617147.html

12. Kirkpatrick, Donald L., *Managing Change Effectively*, Butterworth-Heinemann, Boston, Massachusetts, 2001.

13. Klein, Stuart M, *Journal of Organizational Change Management*, Emerald, London, England, 1996.

14. Linthicum, David S., *Enterprise Application Integration*, Addison-Wesley, Boston, Massachusetts, 2000.

15. Logan, Brenda. Interview by author. Norfolk, Virginia, 15 January 2002.

16. Nicklos, Fred, "Change Management 101- A Primer" Change Management Resources Library 14 February 2002. 25 June 2002 http://home.att.net/~nickols/change.htm

17. Ruh, William A., *Enterprise Application Integration*, John Wiley & Sons, Inc., New York, New York. 2001.

18. Samtani, Gunjan, "Web Services and Application Frameworks" Web Services Architect 27 March 2002. 15 May 2002 http://www.webservicesarchitect.com/content/articles/samtani04.asp

19. Scannell, E., "Power Trio pushes Web Services Security", 27 June 2002. 23 September 2002 http://www.infoworld.com/articles/hn/xml/02/06/27/020627hnwssecure.xml

20. Slater, Derek. "Middleware Demystified" *CIO Magazine* 15 May 2000. 01 June 2002 http://www.cio.com/archive/051500_middle.html

21. Vasudevan, Venu. "A Web Services Primer", O'Reilly XML.com 04 April 2001. 01 August 2002 http://www.xml.com/pub/a/2001/04/04/webservices/index.html?page=1

22. Vawter, Chad. "J2EE vs. Microsoft.NET - A comparison of building XML-based web services", 15 July 2002. 01 August 2002 http://www.theserverside.com/resources/article.jsp?l=J2EE-vs-DOTNET

23. Vizard, Michael. "EAI is dead; long live Web services", *InfoWorld* 29 March 2002. 05 June 2002 http://www.infoworld.com/articles/hn/xml/02/03/29/020329hncapeclear.xml

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. LCDR Glenn R. Cook
   Naval Postgraduate School
   Monterey, California

4. Rex A. Buddenberg
   Naval Postgraduate School
   Monterey, California

5. Ron Steele
   Fitting Out and Supply Support Assistance Center (FOSSAC)
   Norfolk, Virginia

6. Marine Corps Representative
   Naval Postgraduate School
   Monterey, California

7. Director, Training and Education, MCCDC, Code C46
   Quantico, Virginia

8. Director, Marine Corps Research Center, MCCDC, Code C40RC
   Quantico, Virginia

9. Marine Corps Tactical Systems Support Activity (Attn: Operations Officer)
   Camp Pendleton, California

10. Professor Dan C. Boger
    Naval Postgraduate School
    Monterey, California