The SOA Magazine

Feature Article

A Case Study on SOA and Process:
Integrating E-Gov Travel Services with Federal Agency Financial Systems

by Jian "Jeff" Zhong

Published: September 25, 2009 (SOA Magazine Issue XXXII: September 2009)
and November 6, 2009 (SOA Magazine Issue XXXIII: October 2009)
Download PDF

Digg This • De.licio.us • Slashdot • Technorati • StumbleUpon • Google Bookmark

Abstract: In this article I will present a step-by-step service-oriented solution development methodology and then describe how it is being used for the successful implementation of Service-Oriented Architecture (SOA) at a federal agency to integrate its financial systems with E-Gov Travel Services. I will discuss major service components constructed, service-oriented processes followed, business benefits gained, best practices used, and lessons learned. After reading the article, readers should have a good understanding of how SOA is used in real world situations to integrate large scale, highly valued, and mission critical information systems within and across enterprise boundaries.

Introduction

A properly utilized methodology can consistently solve problems effectively and efficiently with great results. One of the best examples is "The Art of War", first drafted and practiced by the ancient military commander and king's advisor Sun Tzu. His methodology, when studied, fine-tuned, and applied in the real world by his followers, produced grand victories for thousands of years [REF-1-3]. The architectural principles and design patterns of Service-Oriented Architecture (SOA) are similar to the war principles and thirty-six stratagems presented in Sun Tzu's masterpiece [REF-3]. Both the "The Art of War" and SOA place deliberate planning, agility, and flexibility as strategic goals in order to respond to fast-changing environments, factors, and changing business objectives [REF-4-7]. In this sense, SOA can be considered "The Art of Services" - the way to use services effectively and efficiently to produce desirable and predictable business results. To be effective, the right services for the right situation must be designed and implemented. To be efficient, services must be identified and constructed on schedule, within budget, and with the promised quality. By extracting and documenting the patterns from successful engagements, leaders in any discipline can be assured of success. This paper will document one such successful engagement.

Methodology is so important, but where should one start? One way is to research industry standard bodies [REF-8-11], major vendors [REF-12-19], and prominent SOA experts [REF-20-23] and reflect on their core principles and best practices. The next step is to embrace, adapt, and fine tune the information to fit your own situation in order to solve your own business problem. In articles published in JavaWorld, I streamlined the Rational Unified Process and summarized its eight steps for J2EE solution development [REF-24-27]. I further elaborated how SOA should be used to migrate an "as-is" to a "to-be" enterprise architecture of an organization [REF-27]. This article will refine my previously published SOA concepts to formulate a simple and practical methodology, based on my past years of SOA practical experience and emerging SOA methodologies from industry experts [REF-4-7, 28, 29]. To develop an executable SOA solution, eight steps (activities) must be performed in one or more iterations. Those steps are:

1. Requirements Analysis
2. Service-Oriented Analysis
3. Solution Architecture

4. Service-Oriented Design
5. Service Component Development
6. Service Component Assembly and Deployment
7. Verification and Validation
8. Operations and Management

I will now present a case study to show how the SOA methodology is used to solve a real world business problem. The Art of War says one must know the big picture before acquiring the important details. So before I begin the steps, I will describe the business background.

The E-Gov Travel Services

The "E-Government Act of 2002" was passed by the United States Senate to improve electronic government services and processes by using Internet technology. Under the President's Management Agenda, the E-Gov Travel Services program [REF-30-31] was one of 24 E-Gov initiatives that were created. As a separate contract line item, E-Gov Travel Services provides integration services to allow the electronic flow of travel data to and from federal agency financial systems and the support of the disbursal of travel-related funds, the synchronization of traveler profiles, and the validation of accounting code information.

Agency Enterprise Business System

The federal agency in this article has an annual budget of over $30 billion (FY2009), with 84% of the total budget supporting over 325,000 extramural scientists and research personnel at more than 3,000 institutions nationwide [REF-32]. Their scientists and physicians travel world-wide to conduct medical research and the agency travels patients for clinical trial studies. The Agency Enterprise Business System (AEBS) is the system of record for all accounting and disbursing transactions processed within the agency. Major functions of AEBS include Accounts Payable, Accounts Receivable, Acquisition and Contracts, General Ledger, Procurement Management, Property Management, Service and Supply Funds Operations, and Travel Management.

Architectural Scope and Constraints

In order for SOA to be productive and gain acceptance, it must solve real world business problems that have critical business impact and have enough importance to secure project funding. Business problems arise as the in-house hosted travel management system is migrated to E-Gov Travel Services. Direct database link has been used to integrate the legacy travel system with backend financials. As the agency migrates to E-Gov Travel Services, direct database access is not practical and it becomes challenging, yet necessary, to allow the flow of financial and travel data between E-Gov Travel Services and AEBS.

The AEBS E-Gov Travel Services project is led and managed directly by the AEBS Director with oversight from the the agency's Chief Financial Officer. The enterprise architecture guidance and infrastructure support is provided by the agency's CIO, Enterprise Architect, and their staff.

This SOA integration approach can be used for E-Gov Travel Services AEBS integration as well as for future system integration projects such as eGrant and other internal and external systems. It is consistent and in compliance with Enterprise Architecture and SOA guidelines and practices [REF-33-37]. The architecture and design, as directed by agency management, is:

- Enterprise scope, reusable across different projects
- Forward compatible to the next Oracle applications (Financials) release
- Leverage existing TIBCO and/or Java EE applications
- Leverage industry best practices and XML data standards

A Step-by-Step SOA Methodology

Requirements Analysis

Federal travel is a set of very sophisticated business processes governed by the Federal Travel Regulation (FTR) [REF-34]. The E-Gov Travel Services involve many business systems including the agency business systems, bank travel card systems, data warehouse and report systems, global distribution systems, travel authorization and voucher system, travel management centers, online booking engines, etc. Readers who are interested in the details can research the E-Gov Travel Services RFP [REF-31]. The RFP presented 11 high-level business use cases common to all agencies:

1. Create and Update Traveler Profile
2. Prepare and Submit Travel Requests
3. Book Travel & Reservations
4. Change Itinerary
5. Prepare and Submit Voucher
6. Scan and Submit Receipt Images
7. Retain Receipt Originals
8. Designate Payment Modalities
9. Receive Itineraries, E-Tickets, Notifications and Reminders
10. Generate Personal Travel Reports
11. Perform Personal Travel Queries

Travel at the agency has many unique business requirements. Overall, medical research and science are really about uniqueness. Each scientist, and associated institute, is very special and often has unique requirements within various levels of subordinate organizations. There are many business requirements; below, I describe only two to show that generic federal E-Gov Travel Services cannot be used for the agency without modification and improvement.

First, only travel planners use E-Gov Travel Services to plan a trip, book a ticket, or submit vouchers to reimburse expenses, while travelers use eVoucher to sign and certify travel documents with a few simple steps. The eVoucher Web application, previously developed specifically for the agency, reduced the huge amount of time, resources and effort that would be required to train (and maintain) more than 77,000 travelers in the use of E-Gov Travel Services. However, with the in-house travel system migration to E-Gov Travel Services, eVoucher no longer had access to the travel data. This problem made SOA attractive, because SOA enabled eVoucher access to E-Gov Travel Services data without changing existing user interfaces.

Second, as part of its mission, the agency travels a large number of patients to conduct scientific and medical research. Patient travel is critical to the agency's mission. Patient traveler profile information, travel authorizations to be approved, and expenses to be dispersed must be concluded as quickly as possible. All interfaces should be real-time or near real-time. When the agency first selected E-Gov Travel Services, most of the E-Gov Travel Services Enterprise Application Integration (EAI) was in batch processes placing a unique challenge on the integration effort. The real world complicated services and systems and the information exchanged among those systems have been simplified in the diagram below.

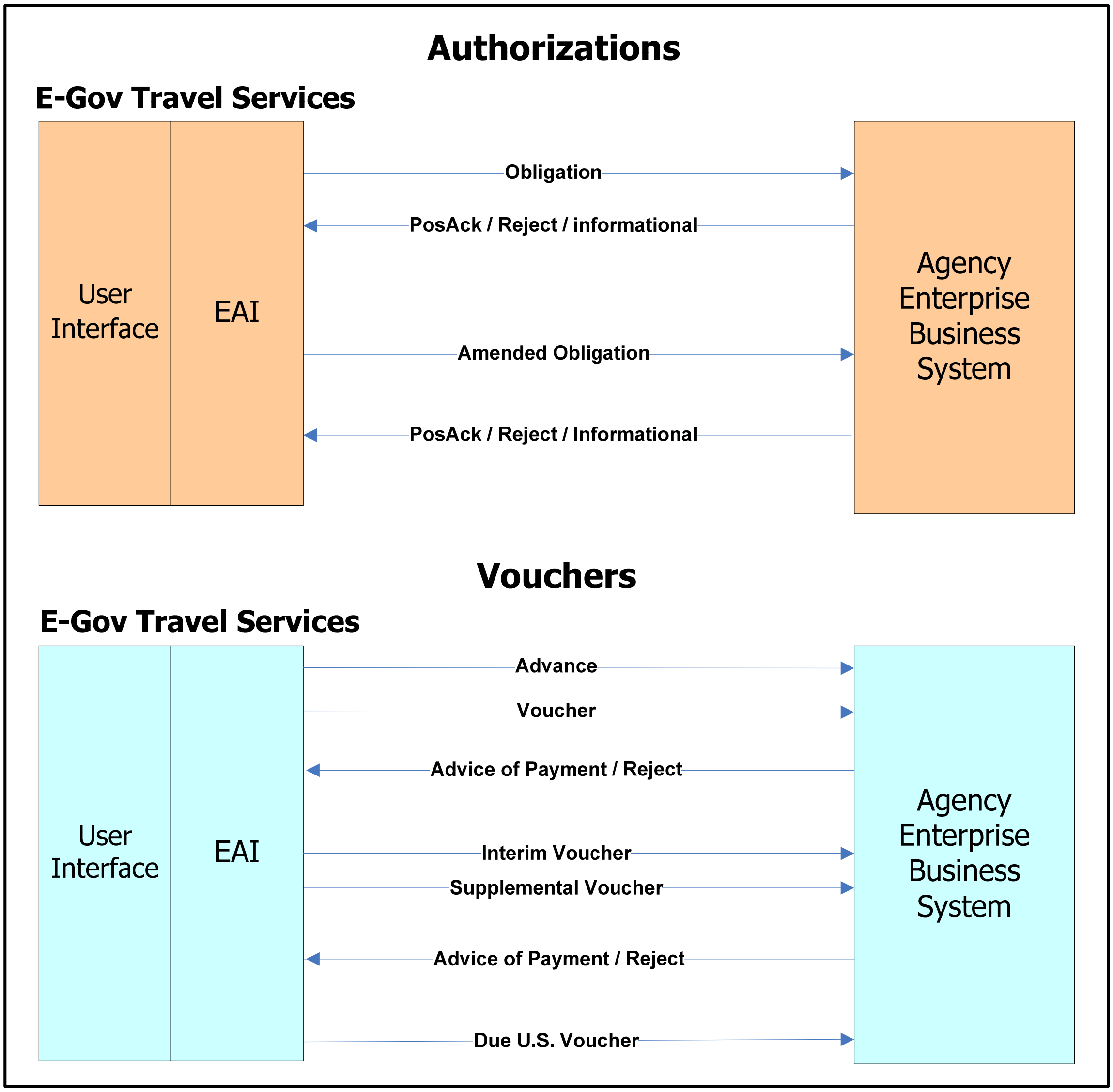

Figure 1: Data Exchange between E-Gov Travel Services and an Agency Business System

In addition to the financial transactions in the diagram above, travel profiles, project accounting, and sponsors' information also must be exchanged. The Requirements Analysis is a separate step from the Service-Oriented Analysis to emphasize that the methodology must be business driven and independent of Web services. In the next step, I will explore when Web services should be used.

Service-Oriented Analysis

The Service-Oriented Analysis is performed by architects to identify services required to support the business requirement in a vendor independent and technology agnostic manner. The top down Web Service Definition Language (WSDL) first design enables the development teams from various companies to begin work independently without being concerned about the technical details and platform used. Since the analysis is technology independent, it is an invaluable asset that can survive the change of technology in the future. Lastly, but not least, this will minimize vendor influence on the architecture development.

The data exchange patterns and formats were analyzed; WSDL and XML schemas were produced as a part of the Integration Agreement. The agreement contains details of the concept of operations and data flow between the E-Gov Travel Services system and the agency's financial system. The following services were identified as common to the integration effort of many agencies:

Web Service (Operations)	Direction	Description
ProjectAccountLookup	E-Gov Travel Services->AEBS	Real-time read-only access to AEBS Projects
SponsorsLookup	E-Gov Travel Services->AEBS	Real-time read-only access to AEBS Customer
FundsCheck	E-Gov Travel Services->AEBS	Real-time read-only access to AEBS Funds Availability
CreatePurchaseOrders	E-GOV TRAVEL SERVICES EAI->AEBS	Transform a travel authorization to many financial obligations and submit to Oracle application workflow
AmendPurchaseOrders	E-GOV TRAVEL SERVICES EAI->AEBS	Transform a travel authorization to many financial obligations and submit the amendment to Oracle applications workflow
CancelPurchaseOrders	E-GOV TRAVEL SERVICES EAI->AEBS	Cancel a trip and all related Transform a travel authorization to many financial obligations and submit the amendment to Oracle applications workflow
CreateVoucher	E-GOV TRAVEL SERVICES EAI->AEBS	An expense voucher (regular, local, supplemental) is transformed to account payables, receivables and general ledger.
PositiveAcknowledgement/Reject	AEBS->E-GOV TRAVEL SERVICES EAI	Positive acknowledgement of financial transactions
AdviceOfPayment	AEBS->E-GOV TRAVEL SERVICES EAI	Payment notification
GetTravelDocumentList	AEBS->E-Gov Travel Services	Required for eVoucher service enable portlet
GetTravelDocumentDetails	AEBS->E-Gov Travel Services	Required for eVoucher service enable portlet
TravelerProfileNew	AEBS->E-Gov Travel Services	Traveler profile synchronization
TravelerProfileUpdate	AEBS->E-Gov Travel Services	Traveler profile synchronization
TravelerProfileTerminate	AEBS->E-Gov Travel Services	Traveler profile synchronization
TravelerProfileReactivate	AEBS->E-Gov Travel Services	Traveler profile synchronization
OrganizationChange	AEBS->E-Gov Travel Services	Traveler profile synchronization

Table 1

The primary purpose for AEBS Web services was to satisfy immediate business requirements. However, the data standard for all services was generic to future projects and could be reused. These services were first produced by architects and then fine tuned by Web service developers with deep domain knowledge. The diagram below is a domain model for the message content in the CreatePOs Web services to illustrate the data models in the XML schema and abstract WSDL. Because AEBS historically mapped a 1:1 relationship between purchase orders and lines, a customized approach to the integration was necessary.

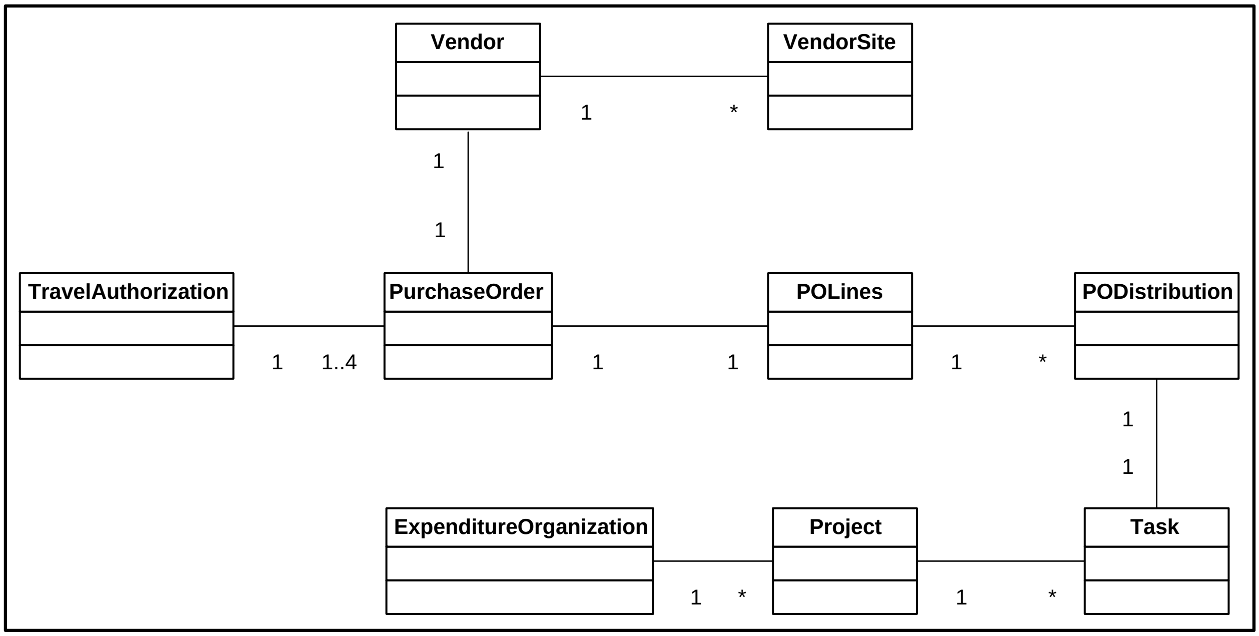

Figure 2: Travel Authorization Class Diagram

Solution Architecture

In the last two sections, I have discussed business problem analysis and the design of an abstract solution. Now I will show how to select vendor technologies and develop an executable architecture with sufficient detail in order to begin our concrete design and construction. Like any technology solution, one must locate commercial tools to facilitate the development effort. The tools for service development, deployment, and operation and management are part of the infrastructure. First, the AEBS eTravel technical team researched Enterprise Architecture (EA) policies and standards. EA has policies with direct impact on:

- Platform Selection
- Certification and Accreditation
- Data Standard
- Federal Information Security Management Act (FISMA) Compliance

In 2004, AEBS conducted a three-month study on the agency's unique requirements and technical challenges necessary to migrate to E-Gov Travel Services. A prototype was also developed using Apache Axis open source software. In 2007, the agency developed and published integration architecture guidelines for all future integration projects, and decided to use TIBCO products for the enterprise-wide services to interface with the AEBS system, while utilizing Oracle products for the business processing logic. The following infrastructure components were available for our project application architecture.

- F5 Load Balancer and SSL Accelerator
- CA SiteMinder Federated Authentication for Web Applications and Services
- Enterprise Directory
- TIBCO BusinessWorks
- TIBCO Enterprise Message Bus

- Oracle 10G Database Real Application Clustering
- Oracle 10G Application Server
- Oracle Plumtree Portal
- Oracle 10G Advanced Queuing (AQ)
- Oracle Applications (Account Payable, Accounts Receivable, General Ledger, HR, Supplier)
- Oracle JDeveloper
- Oracle Enterprise Manager
- Sun Solaris on Sun Fire E25K

After prototyping and many executive briefings and technical discussions, the following diagram illustrates how the above infrastructure components were connected in our high-level application architecture.

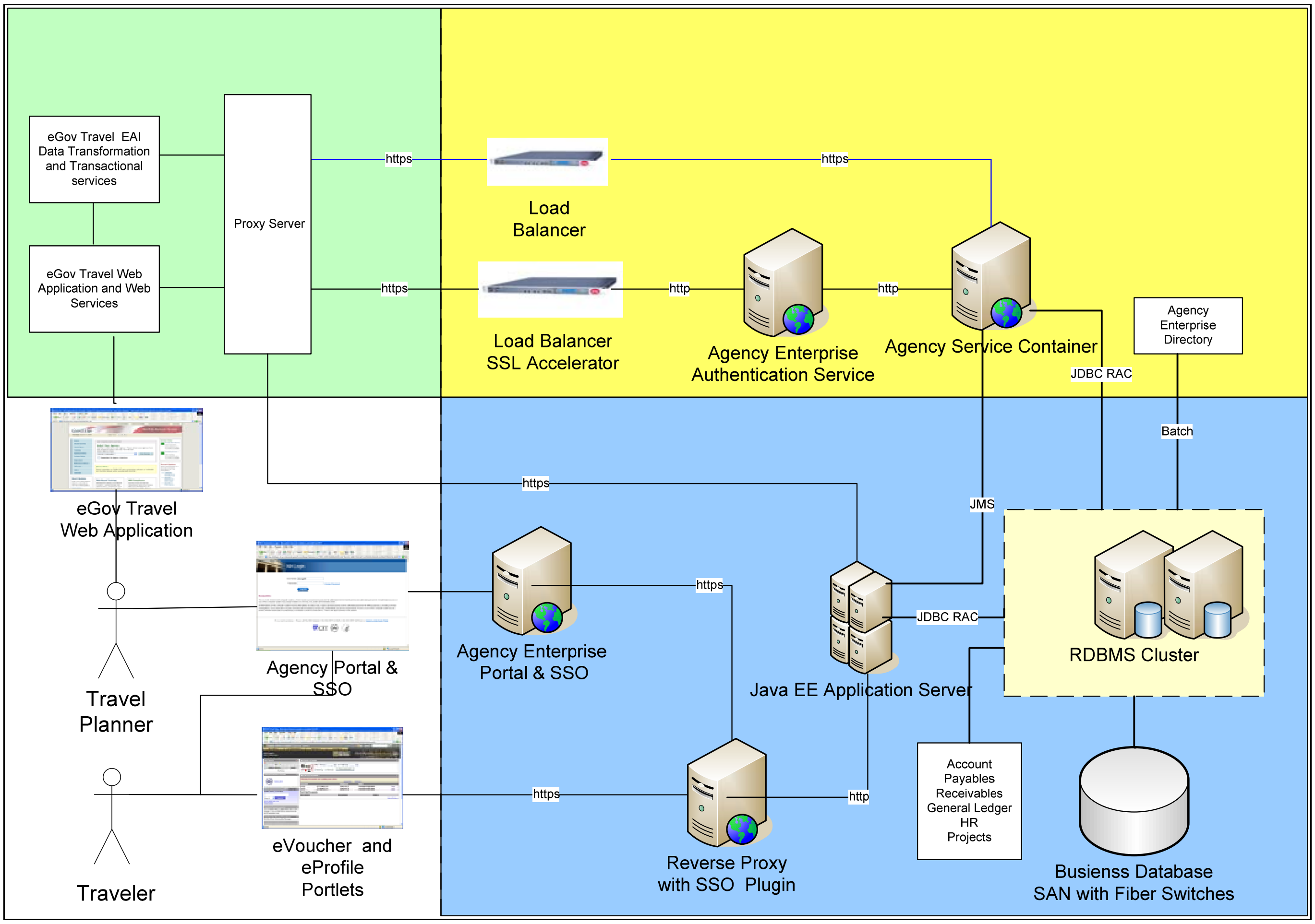
eGov Travel EAI Data Transformation and Transactional services
eGov Travel Web Application and Web Services
Proxy Server
https
https
https
Load Balancer
Load Balancer SSL Accelerator
http
http
Agency Enterprise Authentication Service
Agency Service Container
Agency Enterprise Directory
JDBC RAC
Batch
eGov Travel Web Application
https
JMS
Travel Planner
Agency Portal & SSO
Agency Enterprise Portal & SSO
https
JDBC RAC
RDBMS Cluster
Java EE Application Server
Traveler
eVoucher and eProfile Portlets
https
Reverse Proxy with SSO Plugin
http
Account Payables Receivables General Ledger HR Projects
Busienss Database SAN with Fiber Switches

Figure 3: Integration Architecture

The architecture diagram above follows the standard n-tier application architecture for Web applications and services [REF-25-27].

In the process of analyzing business requirements and architectural constraints, the following major architectural considerations were relevant to our solution design.

Architecture Items	**Decision**
Web Service Transport Layer Security	All external services used FIPS compliant SSL connection.
Web Service Federated Authentication	In addition to transport level security, all external services used Web Service Security (WSSE) message level authentication.
Loosely Coupled Systems	Decoupled Web service end point from E-Gov Travel Services integration project so future projects could use the service directly without modification. Decoupled TIBCO and Oracle so each product suite could be managed and upgraded independently.
Web Service Transaction	All transactional services supported atomic transaction. A transaction had to be committed fully or completely rolled back.
Web Service Reliable Messaging	Message from transactional service was submitted and processed exactly once. No redundant or missing obligations or invoices.
Event Driven SOA	Patient traveler profile was placed on the service bus and transmitted to E-Gov Travel Services immediately after record was created in AEBS.
Service Enabled Portlet	Existing eVoucher Web application user interface remained unchanged and was powered by Web services.
Three Way SSO	The enterprise portal, E-Gov Travel Services and eVoucher all used federated authentication supported by SAML.
Message Driven Bean Durable Subscriber	E-Gov Travel Services specific business logic was implemented in Oracle Message Driven Beans attached to the TIBCO EMS as durable subscribers.
Persistent Publisher	All message publishers had to be persistent.
Component Based Architecture	To facilitate reuse, all E-Gov Travel Services specific code used plug-n-play components.
Service Object Mapping	XML messages were automatically mapped into Java objects by the architectural framework when business logic was performed.
Object Relational Mapping	After business logic was performed, Java objects were automatically mapped into relational data as an atomic transaction by the architectural framework.
Real-time Request Response	Services went directly to the database to perform read-only lookup access.
Real-time Message Requestor	Most of pub/sub transactions were performed asynchronously. There were a few situations performed synchronously.
Generic Value Objects	A generic value object was used for all business objects.
Content Based Message Routing	All reusable services used content-based routing to implement project specific business logic.

Table 2

All relevant technical, security, application, and data architecture specifications were formally documented in the Interconnect Security Agreement (ISA), the Memorandum of Understanding (MOU), and the Data Integration Agreement. These documents governed the relationship between the agency, GSA, and the E-Gov Travel Services provider, including designated managerial and technical staff, in the absence of a common management authority.

From Prototype to Frameworks

Before the design effort, AEBS performed prototype work to help formulate and verify architectural decisions so that there would be no major surprises on the road to implementation. As always, after the prototype, some approaches were validated, and some design ideas were confirmed and fine-tuned. The relevant parts of the components were migrated into an architectural framework to be carried over into the design and construction phase.

Service-Oriented Design

This process, with all the required details in a specific enterprise technology environment, according to Service-Oriented Analysis results and architectural decisions and guidelines, produced a blueprint. It was platform independent and fully utilized vendor capabilities and value added-on features. Using CreatePOs Web service as an example of the scope of the design, a high-level workflow was developed. Major business activities within the workflow are described below.

1. After a travel authorization was approved in E-Gov Travel Services, up to four purchase orders were created.
2. E-Gov Travel Enterprise Application Integration (EAI) made a Web service call to the CreatePOs service for PO creation in AEBS.
3. Upon TIBCO BusinessWorks (BW) receipt of the request, appropriate security measures were applied. When it was a valid user, a notification message was immediately sent to the E-Gov Travel EAI component.
4. TIBCO BW then published the request to the Enterprise Message Service (EMS). The subscriber (Oracle AppServer) received and processed the message, which traveled through a set of validations for E-Gov Travel related PO edits.
5. When a request passed all edits and validations, up to four purchase orders were inserted into the following Oracle Open Interface tables: PO_HEADERS_INTERFACE, PO_LINES_INTERFACE, and PO_DISTRIBUTIONS_INTERFACE. Each PO line may have had multiple distribution lines associated with it.
6. When the PO creation was successfully completed in the Open Interface tables, the interface sent a notification message to TIBCO EMS.
7. This triggered the PosAck (positive acknowledgement) Web service request from TIBCO BW to E-Gov Travel EAI, completing the CreatePOs Web service transaction.

After the business activities had been identified, AEBS added enterprise-wide components such as Web Service Security (WSSE),and logging, and exception handling. Starting from abstract WSDL, TIBCO BusinessWorks was used to generate concrete WSDLs for all inbound services. The combined diagram that includes the business activities, security, and infrastructure components is shown below.

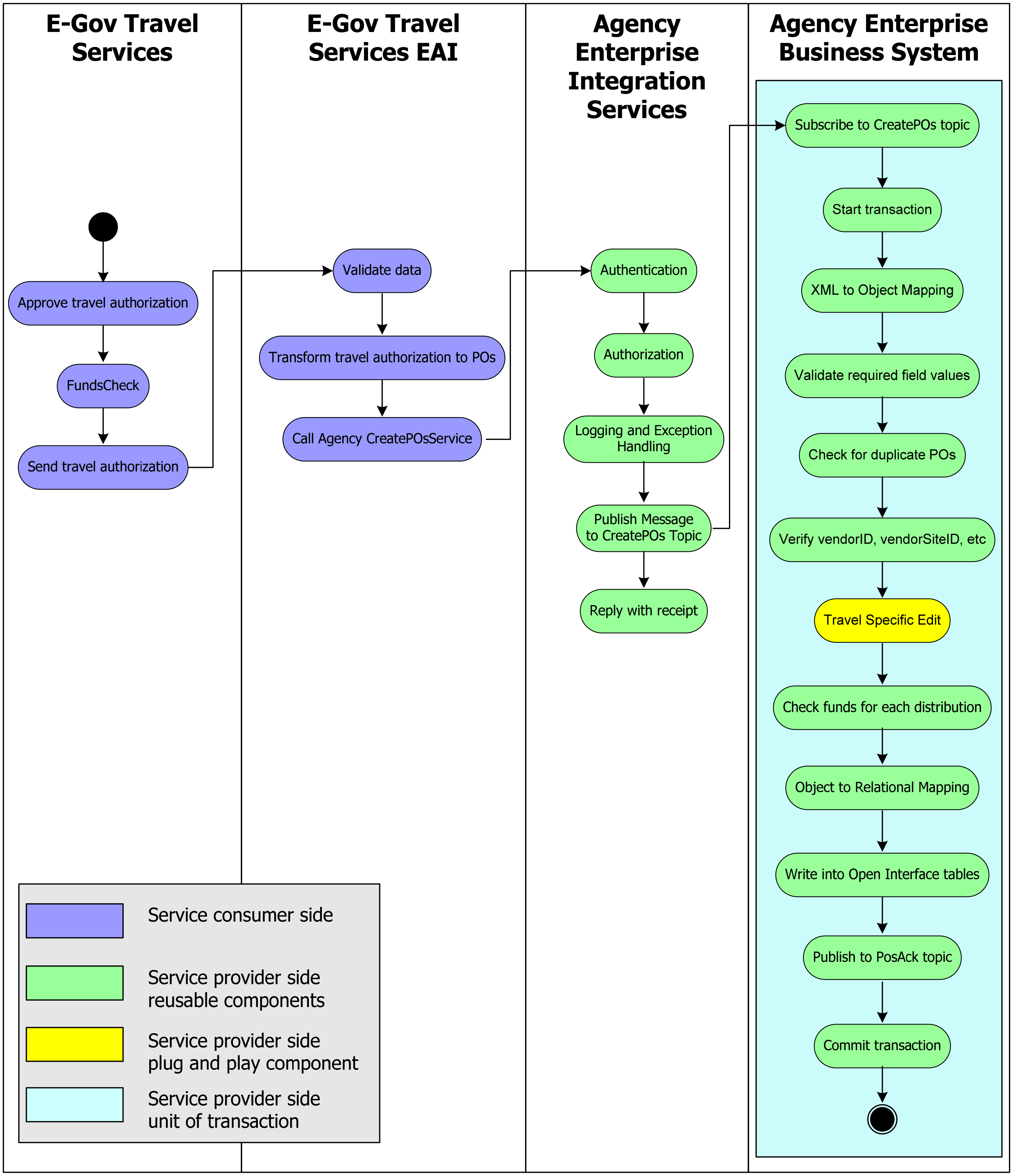
E-Gov Travel Services
E-Gov Travel Services EAI
Agency Enterprise Integration Services
Agency Enterprise Business System
Approve travel authorization
FundsCheck
Send travel authorization
Validate data
Transform travel authorization to POs
Call Agency CreatePOsService
Authentication
Authorization
Logging and Exception Handling
Publish Message to CreatePOs Topic
Reply with receipt
Subscribe to CreatePOs topic
Start transaction
XML to Object Mapping
Validate required field values
Check for duplicate POs
Verify vendorID, vendorSiteID, etc
Travel Specific Edit
Check funds for each distribution
Object to Relational Mapping
Write into Open Interface tables
Publish to PosAck topic
Commit transaction
Service consumer side
Service provider side reusable components
Service provider side plug and play component
Service provider side unit of transaction

Figure 4: CreatePOs Web Service Workflow Diagram

When the design progressed into component level, such as Message-Driven Beans for transactional business logic, I felt that Object-Oriented Design still added significant value to SOA. From this point of view, I agree with Thomas Erl's opinion and proved that Service-Oriented design and Object-Oriented design are not mutually exclusive.

At the end, coarse grained services must be implemented in executable software components. AEBS clearly separated the generic framework components such as service-to-object mapping, object-to-relational mapping, and persistent framework from business logic components; then further separated AEBS generic components from travel specific components so that the framework could be used for any future SOA project without much modification.

Service Component Development

The above four steps concluded the planning stage of the service development. Business problems were identified, abstract solutions were analyzed, technologies were selected, architectural decisions were made, and software components were identified. Our development effort was divided into two major categories:

1. All agency Web service end points, logging and exception handing, and reliable messaging were handled by TIBCO BusinessWorks and the following components were developed with a GUI tool called TIBCO Designer:

   Inbound Lookup and Transactional Services This component provided the end points for all of the agency developed Web services for E-Gov Travel Services-AEBS integration.

   Outbound Traveler Profile and Transaction Notification Services This component calls E-Gov Travel Services and E-Gov Travel Services EAI Web services to update traveler profiles and send notifications.

2. All business logic, transactional features, and service-enabled portlets were handled by Oracle products, and the following components were developed with Oracle JDeveloper:

   TIBCO Oracle Message Bridge This component enabled real-time patient travel.

   Java EE Message Driven Beans This was the transactional subscriber to encapsulate all business logic related to accounts payable, accounts receivable, general ledger, and purchase orders.

   Java EE eVoucher Service-Oriented Portlet This was a legacy Web application that provided user interface through the the agency portal. The migration maintained the same user interface and called E-Gov Travel Services Web services to retrieve and update data in E-Gov Travel Services.

   Java EE eProfile Service-Oriented Portlet This was a new job aid to allow the agency users to update their personal information such as credit cards.

Service Component Assembly and Deployment

All components were assembled into Java EE EAR [REF-38] or TIBCO EAR files. A Java Enterprise ARchive, or EAR, is a file format used by Java EE for packaging one or more modules into a single archive so that the deployment of the various modules onto an application server occurs simultaneously and coherently. It also contains XML files called deployment descriptors which describe how to deploy the modules. A TIBCO EAR file contains process definitions and WSDLs, XML schemas, configuration files, and resource files. All software components in the E-Gov Travel Services-AEBS integration project were packaged as six EAR files:

1. TIBCO Inbound Services EAR
2. TIBCO Outbound Services EAR
3. Oracle Financials MDBs EAR
4. Oracle eVoucher EAR
5. Oracle eProfile EAR
6. Oracle-TIBCO Message Bridge EAR

TIBCO EARs were preconfigured with global variables and could be reconfigured using a TIBCO Administrator tool by the agency Integration Service Center support. Oracle EARs were preconfigured according to Java EE standards with Oracle Enterprise Manager by the AEBS infrastructure support team. This enabled the same set of code to be deployed into development, test, and production environments, saving significant time and resources.

It is worth mentioning that the development effort was divided into two major phases. Each phase completed the eight cycles: Phase I concentrated on infrastructure and framework development; and Phase II added more business functionality.

Verification and Validation

To obtain the best quality for the project, each component was unit tested thoroughly in a local development environment until the services were assembled and deployed. After service was deployed, and prior to the full integration, a functional expert completed a set of all test scenarios by interjecting XML data files into the workflow. When software deficiencies were identified, issues were addressed and corrected, and the software components were redeployed. After all AEBS generic business logic and E-Gov Travel Services specific business logic was tested and implemented correctly, the services were exposed to E-Gov Travel Services and E-Gov Travel Services EAI teams for consumption, further tests, and final integration. After E-Gov Travel Services and E-Gov Travel Services EAI had finished their development and all the moving parts were connected, functional teams conducted end-to-end independent tests with the predefined test scenarios. After integration tests and all critical bugs were corrected to the satisfaction of AEBS management, end-users conducted User Acceptance Tests.

Operations and Management

Without effective management and monitoring, it would be difficult to diagnose the root cause when a problem occurred in the system. This was especially true for the near real-time financial integration with a long communication and component chain, which involved seven different organizations within the agency and the E-Gov Travel Services provider. The longest communication invoked 21 components to complete a single Web service request. Experience showed that most problems occurred on interfaces between enterprise boundaries and different vendor products.

To facilitate troubleshooting, a close collaboration effort was required and the participating groups often included network, proxy server, security, TIBCO servers, Oracle servers, and the DBA teams. A single command and control team with knowledge and working experience across all platforms and technologies directed the communication and workflow.

Benchmark against Industry SOA and the Agency Enterprise Architecture Principles

Now that the solution development life cycle has been examined, I will show how the steps are related to industry SOA principles [REF-4-7] and the agency Enterprise Architecture principles [REF-33].

Standardized Service Contracts and Service Abstraction

The Service-Oriented Analysis step produced the abstract WSDL and associated XML schemas - a contract between service consumers and providers. For the AEBS integration project, the contract provided sufficient detail so that design and development from both the consumer and the provider could begin to work simultaneously. The abstract WSDL was platform independent so that the architecture could withstand technology evolutions. Most importantly, it provided a benchmark to measure vendor product features.

Service Loose Coupling

The service contract generated from the analysis steps decouples the service interface definition and the underlying implementation technologies. It also decouples the service provider and the consumer design and development effort. In the Solution Architecture step, AEBS adopted a publisher/subscriber design pattern so that publisher or subscriber could change without impacting the system so long as the contract was not changed.

On the subscriber side, a clear delineation of generic service logic from project specific plug and play components made the components and services reusable - yet fulfilled each project's unique requirements.

Service Reusability

As a result of the project, both the E-Gov Travel Services provider and the agency have developed a complete set of Web services to integrate E-Gov Travel Services and AEBS. E-Gov Travel Services is contractor-owned and contractor-operated software as a service (SaaS), while AEBS uses primarily Oracle COTS products with agency customizations. Since the agency was the first to use SOA to integrate with E-Gov Travel Services, the most usability would be achieved in a future project with a similar set of software and services. The lessons learned, processes followed, and components and services developed can all be reused with very little modification. This effort should be ideally coordinated by the E-Gov Travel Program Management Office in GSA.

For E-Gov Travel Services, the traveler profile services and eVoucher services could be used by all agencies. This would provide real-time integration over the currently dominant batch integration.

For future AEBS integration, the purchase order services, invoice services, general ledger services, and project lookup services can be shared and reused. At the agency, this was the largest SOA effort and first across-enterprise integration.

Service Autonomy and Stateless

The WSDL that was defined in the analysis steps has a complete set of data capture for all business requirements and logical processes to complete a single business transaction. It does not have any dependency on context data or session data. A complete set of data in a message can be transmitted from one component to another. It was difficult, if not impossible, to keep session data when messages passed into different layers of services and message systems.

Service Discoverability

Similar to any other integration project, each project has a registry for Web services, a WS-Inspection feature implemented by TIBCO BusinessWorks products.

Service Composability

Since AEBS services are mostly financial transactions, transactional messages are usually composed of entity objects and aggregated into a very high level. For example, a travel authorization is composed of up to four purchase orders, and each purchase order consists of vendor information, project, task, and expenditure organizations. A voucher can be a mixture of accounts payable, accounts receivable and general ledger and be related to purchase orders. To compose a transactional service with transaction services posts some challenges. The following advanced features must be implemented to efficiently compose services from services.

- WS-AtomicTransaction - for short duration transactions
- WS-BusinessActivity - for long duration transactions
- WS-Coordination - to coordinate composing transactional services
- WS-ReliableMessaging - to guarantee a transaction is submitted exactly once in the event of system failure or application exceptions.
- Security infrastructure - to propagate security context from upstream to downstream service components

Unless the above features [REF-39-43] are supported by the technical infrastructure, the transactional services can only be composed of JMS, JDBC, and look-up read-only Web services.

Project Staffing

In a competitive sourcing environment, resource allocation and cost estimating are very important for bidding, winning and successfully implementing any SOA project. Many people contributed and played essential roles in the E-Gov Travel Services-AEBS integration project. To fit into this technical discussion, only the technical team is mentioned here:

People:

- E-Gov Travel Services Integration SME (Phase I and II)
- SOA Architect (Phase I and II)
- Web Service Developer (Phase I and II)
- Web Service Developer (Phase II)
- Oracle Technical Expert (Phase I and II)

Domain Knowledge:

- E-Gov Travel Services, Federal Travel and Agency Travel Policies and Regulations
- Oracle Applications Workflow

Technical Skills:

- TIBCO Production Suite
- Oracle Production Suite
- https protocol, forward proxy server, reverse proxy server, and load balancer
- SSO, SAML
- PKI
- FIPS Compliant Encryption Modules
- XML Schema and WSDL Definitions
- Java EE: JMS, JCA, JDBC, AJAX, and Oracle RAC
- Application Server Administration
- Deployment
- Architecture Principles
- Design Patterns

Lessons Learned

SOA must be business driven and incrementally deliver business results. I have seen organizations that are heavily invested in SOA infrastructure and ambitious big-bang service design, deliver tons of architecture documentation yet fail to reach consensus or deliver concrete services and business value.

SOA is more than just Web services. I have seen a project develop Web services with the calling clients all batch processes. In another situation, a DBA developed a Web service to wrap a database table and in the same time frame also developed a Web service client to move the data into another table in the same domain. These were costly technical endeavors that resulted in little or no value-added benefit.

Do not rush into developing Web services without architecture and systems planning: you may discover, after a project is completed, that the services - which may be redundant - cannot be reused without significant modification. I have seen a case where the first project was copied into subsequent projects to claim reusability, which resulted in multiple versions of code and redundant Web services. The use of copy-and-paste is not SOA. Silo approaches may create redundant Web services; therefore, the architect must capture a holistic view of the enterprise.

Business Benefits Gained

Seamless near-real-time integration ensures that systems are always in sync with each other, reducing potential errors, and time and cost to correct those errors. Real-time integration also eliminated redundant data entry of tens of thousands of travelers' profiles at the agency, improving service performance over the legacy system.

It is a good return on investment (ROI) to utilize the enterprise-wide licenses, infrastructure, and future-proof SOA architecture and design to solve real business issues and provide great service to the user communities. The more projects that utilize the SOA infrastructure and architecture, the more the agency will reap a return on SOA investment and the benefits it offers.

Future Considerations

Industry Trend

In order to make a business case, services must be generic and reusable to accommodate a large enough customer base to sustain a profitable business. However, history in the last decade showed that when service is generic enough and the market is large, some of the giant gorillas will get you even when you have the best technology. Netscape, Sun Microsystems, WebMethods, and BEA are good examples that the best technology may not necessarily win the business war or even survive. Vendor consolidation will continue and after the recession is over, you may find only a very few dominant players left. In this sense, the Art of Services is the Art of War. It is critical for a customer to assess the long-term viability of a software vendor, foresee the changes, and upgrade when necessary.

Oracle Applications Upgrade

A frequently asked question about the AEBS SOA project is what it will look like when the agency upgrades to the newer Oracle Applications Release coupled with the Integrated SOA Gateway [REF-44-47]. The gateway offers the ability to expose AEBS functionality as SOAP-based Web services. While this is not something that can be answered in a short article, you will find that my solution is loosely coupled in many ways so that parts can be replaced without affecting the whole. The architecture has foreseen, and taken into consideration, future change and upgrades. The newer release will not diminish the unique business requirements and specific custom processing logic. However, it could create an opportunity to integrate the best features offered by Oracle into the current SOA architecture creating the most effective solution for supporting agency requirements.

WSSE

One of the most challenging aspects in implementation of SOA in federal government agencies is Web Service Security (WSSE). Without a vigorous security review and enterprise security capability, the components and systems cannot be connected and integrated. AEBS has implemented WSSE with the support of agency security and IT personnel. AEBS also followed NIST Web Service Security Guideline (Pub 805), researched WSSE X509 and SAML authentication, Message Integrity and Confidentiality, and provided a technical proposal to agency management and to GSA. Government-wide WSSE implementation is useful to an individual agency as well as being very beneficial to all E-Gov initiatives, and therefore can reach much larger scale service reusability and realize bigger cost savings. The E-Authentication initiative, now called Federal Identity Management, could be expanded to cover Web applications as well as Web services [REF-48].

Enterprise SOA

There is still a long road from reusable to reusability where SOA is not implemented at the enterprise level. First, most projects are measured by fulfillment of business requirements. It is difficult to tell whether services can satisfy future business requirements and needs. In addition, other projects may not have the motivation or incentive to reuse existing Web services given that projects are often funded and managed by different organizations and sometimes supported by competing contractors. Therefore, to achieve the most business benefit, it is crucial for an organization to take a holistic approach to plan, execute, and manage SOA at the enterprise level. The affirmative experience from this effort will have a very positive impact on AEBS program's vision to become a world class service organization.

Conclusion

In summary, I have presented a step-by-step methodology to plan, design, and develop an effective technical solution for fulfilling a specific set of business requirements. The first four steps are focused on architectural planning and design, while the last four are focused on engineering and implementation. The SOA architecture, technical solution, and its implementation are driven by the business requirements and the needs of the organization, within the constraints of resources, costs, and the timeline. It must have sufficient detail and steps to provide the foundation for successful design, development, and deployment. It is my strong belief that with proper planning and the implementation of SOA, an organization may realize the long-term benefits of agility, reusability, and cost savings.

Acknowledgements

The author would like to thank Kate Burton for editing and proofreading this article and I'd like to thank the following leading industry experts and government executives for reviewing this article and providing valuable input:

- Jerry Zhou, CEO of Futrend Technology Inc., E-Gov Travel Integration Subject Matter Expert
- Venkatapathi Puvvada, CTO, Vice President and Managing Partner of Unisys, Former Chair of the Industry Advisory Council, Federal 100 Winner of 2003, 2005 and 2008
- John Butler, Chief Architect of Everware, Inc and Co-Chair of OMG Government Domain Task Force
- Miko Matsumura, Vice President, Product Marketing and Chief Strategist at Software AG, OASIS SOA Blueprints Adoption Committee Chair, author of "SOA Adoption for Dummies"
- Michael Poulin, Proof of Solution Manager at PADA, OASIS Reference Architecture Committee Member, and author of "Ladder to SOE"
- Dr. Jack Jones, NIH Chief Information Officer and Director of CIT
- Helen M. Schmitz, NIH Acting Chief Architect and Director of Enterprise Architecture
- Andrew Merriman, GovTrip eTravel Technical Director

The author would also like to thank our management and project team members for their outstanding work, excellent support and assistance, and for providing a pleasant working environment.

References

[REF-1] http://en.wikipedia.org/wiki/The_Art_of_War

[REF-2] http://en.wikipedia.org/wiki/Sun_Tzu

[REF-3] http://en.wikipedia.org/wiki/Thirty-Six_Strategies

[REF-4] http://www.soamethodology.com/

[REF-5] An SOA Vendor Evaluation Methodology; http://www.soamag.com/I20/0708-3.asp

[REF-6] SOA and EDA: Using Events to Bridge Decoupled Service Boundaries; http://www.soamag.com/I4/0207-2.asp

[REF-7] Service-Oriented architecture (SOA) was first described by Gartner in 1996; http://www.gartner.com/DisplayDocument?doc_cd=114358

[REF-8] http://www.opengroup.org/projects/soa/

[REF-9] http://soa.omg.org/

[REF-10] http://www.oasis-open.org/committees/tc_cat.php?cat=soa

[REF-11] Navigating the SOA Open Standards Landscape Around Architecture; A paper edited by: Heather Kreger, IBM Jeff Estefan, NASA/Jet Propulsion Laboratory July 2009; https://www.opengroup.org/projects/soa/uploads/40/20044/W096.pdf

[REF-12] http://www-01.ibm.com/software/solutions/soa/

[REF-13] http://www.ibm.com/developerworks/webservices/library/ws-soaintro2/

[REF-14] http://www.microsoft.com/soa/

[REF-15] http://www.oracle.com/technologies/soa/index.html

[REF-16] http://www.tibco.com/software/soa/default.jsp

[REF-17] http://www.softwareag.com/corporate/products/wm/default.asp

[REF-18] http://www.jboss.com/resources/soa/

[REF-19] http://www.sap.com/platform/soa/index.epx

[REF-20] http://www.linkedin.com/pub/anne-thomas-manes/0/1/71

[REF-21] http://blogs.oracle.com/davidchappell/

[REF-22] http://booksbysandy.com/

[REF-23] http://www.thomaserl.com/

[REF-24] From Stove-piped Projects to Unified Enterprise Architecture: Strategic considerations for E-Authentication service development; http://www.javaworld.com/javaworld/jw-03-2003/jw-0321-sso.html

[REF-25] U.S. Department of Energy Signs on to J2EE: Create a secure single sign-on Web service for multiple n-tier Web applications. JavaWorld May 2002; http://www.javaworld.com/javaworld/jw-05-2002/jw-0524-signon.html

[REF-26] A Java Case Study: The power of J2EE How the Energy Information Administration secured a complete J2EE solution in less than five months. JavaWorld January 2002; http://www.javaworld.com/javaworld/jw-01-2002/jw-0118-j2ee.html

[REF-27] Step into the J2EE Architecture and Process: Develop complete J2EE solutions with an eight-step cycle. JavaWorld September 2001; http://www.javaworld.com/javaworld/jw-09-2001/jw-0928-rup.html

[REF-28] Working with SOA and RUP by Solmaz Boroumand; http://www.soamag.com/I16/0308-1.asp

[REF-29] Event-Driven SOA; http://elementallinks.typepad.com/bmichelson/2006/02/eventdriven_arc.html

[REF-30] http://www.gsa.gov/etravel

[REF-31] GSA Solicitation FBGT-CD-030001-N available at https://www.fbo.gov/notices/993abc7c2c6f22ef8bfeb6f94c8861de

[REF-32] http://www.nih.gov/

[REF-33] http://enterprisearchitecture.nih.gov/ArchLib/AT/IA/Integration/IntegrationPrinciples.htm

[REF-34] http://www.gsa.gov/ftr

[REF-35] http://acquisition.gov/far/

[REF-36] http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf

[REF-37] http://enterprisearchitecture.nih.gov

[REF-38] http://java.sun.com/javaee/

[REF-39] http://www.ibm.com/developerworks/library/specification/ws-rm/

[REF-40] https://www.ibm.com/developerworks/webservices/library/ws-wsilover/

[REF-41] http://www.ibm.com/developerworks/library/specification/ws-rm/

[REF-42] http://www.ibm.com/developerworks/library/specification/ws-tx/

[REF-43] http://www.ibm.com/developerworks/library/ar-archtemp/

[REF-44] http://download.oracle.com/docs/cd/B34956_01/current/html/docset.html

[REF-45] http://download.oracle.com/technology/tech/soa/soa_best_practices_1013x_drop3.pdf

[REF-46] http://convergence.satpathy.org/

[REF-47] http://www.oracle.com/technology/products/applications/events/oow-2008/S298465_VeshaalSingh-SOA-EBS.pdf

[REF-48] http://www.idmanagement.gov/

Disclaimer

The SOA methodology discussed in this article is based on my knowledge or direct work experience on E-Gov Travel Service integration projects from federal agencies. The views presented in this article are all personal; the author does not represent any of the government agencies. Some technical details were simplified, generalized, omitted, or sanitized.

Biography: Jian "Jeff" Zhong

Mr. Zhong is Chief SOA Architect at Futrend Technology, Inc (http://www.futrend.com). In this capacity, he provides thought leadership, architecture guidance and hands-on implementation support for Futrend's federal clients in the areas of E-Government and mission critical systems' integration. He has been working in various architect roles for more than a decade and has extensive experience in software architecture, application architecture, enterprise architecture, security architecture, and Service-Oriented Architecture. In the last ten years, he has been providing professional services to clients in the public sector such as National Institute of Standards and Technology (NIST) of the U.S. Department of Commerce, the National Institutes of Health (NIH) at the U.S. Department of Health and Human Services, the Energy Information Administration (EIA) of the U.S. Department of Energy and the General Services Administration (GSA).

He is a Sun Certified Enterprise Architect (SCEA), and is certified in Architect, XML and ITIL from industry leaders such as Sun Microsystems, Oracle and IBM. Zhong earned his Masters of Science degree from University of Maryland College Park. He has had numerous articles published on Java EE, enterprise architecture, enterprise Single Sign-On, and XML related technologies.

Futrend Technology, Inc is a fast growing system integrator/consulting company specializing in major mission-critical federal business systems integration. Futrend has a proven track record in systems integration with major financial systems such as Oracle Financials, PeopleSoft Financials, Federal Financial System (FFS), and Momentum for federal agencies such as CMS, HHS, HUD, NIH, and Smithsonian Institution. For more information about Futrend's SOA practice, please visit http://www.futrend.com/soa.