SOA Reference Architecture

August 15, 2007
Prepared by Robert Woolley, Chief Technologist and Strategic Planner

Introduction

This document is a derivative work of current documentation and presentations by OASIS and the Open Group Service Oriented Architecture (SOA) working groups, and applicable SOA research from IBM and other vendors.

Reference Architecture (RA) and related reference models constitute an abstract framework for understanding significant entities and relationships between them within a service oriented environment, and for the development of consistent standards or specifications supporting that environment.

Reference models within a reference architecture is based on unifying concepts of SOA and may be used to develop specific service oriented architectures or for explaining SOA. A reference model is not directly tied to any standards, technologies, or other concrete implementation details. It does seek to provide a common terminology that can be used across and between different implementations.[i]

SOA brings forward many benefits, including greater agency flexibility and response. This is achieved, at the enterprise level, through the application of a service oriented style to the enterprise architecture fabric of the State. There are many different approaches for achieving this, and an at least equal set of possible obstacles, such as:

- difficulty identifying compatible solutions across agencies;
- agencies that follow different sets of standards;
- agencies that use different types of terminologies, making service discovery much more difficult;
- difficulty mapping customer needs to capabilities;
- different asset reuse policies among agencies; and,
- different tools and governance approaches among agencies.

These issues have no simple, single answer; there are, in fact, many answers. One best practice approach is to use a Reference Architecture for the

development of consistent standards or specifications supporting that environment.

SOA Integration Values
SOA brings some sharply contrasting value patterns with previous integration methodologies, such as point to point integration and traditional Enterprise Application Integration (EAI).[ii]

- **Point to Point Integration**
    - Process flows were hard coded and difficult to change.
    - Development projects were inherently risky.
    - No out of the box management and monitoring was available.
- **Traditional EAI**
    - Proprietary (metadata, process, security, and User Interface (UI)).
    - Intrusive application models.
    - Separate dedicated infrastructure.
    - Expensive to implement and operate.
- **SOA Based Integration**
    - Standards Based (e.g., XML, WSDL, BPEL, etc.).
    - Synchronous and asynchronous interactions.
    - Flow coordination.
    - Advanced exception management.
    - Extensible and composable.

Reduced integration efforts using SOA patterns have had a dramatic impact on the time required for integration efforts. In some cases, projects that took a year or more with previous models have been reduced to less than a month in terms of time to benefit, and at greatly reduced costs.

Reference Architecture
An RA can be defined as a collection of Architectural Building Blocks (ABBs), their interactions, the standards associated with each building block and its interactions, the architectural decision points, and the patterns that define and incorporate the utilization of these building blocks. This SOA-RA focuses on ensuring a service oriented architectural style across relevant architectures. The SOA-RA encourages:

- compatible solutions across groups;
- adherence to common standards;
- common terminologies;
- adoption of a common set of tools;
- adoption of common governance; and,
- adoption of common asset reuse approaches.

The SOA-RA identifies base standards that must be defined, published, and distributed to ensure a consistent, functional enterprise implementation of SOA resources by the agencies. Figure one illustrates the positioning of the SOA-RA to the overall State EA. The base standards categories that are suggested for initial inclusion incorporate definition and identification of:

- Agency Specific Web Services
- Common Services Provisioned to the Enterprise Service Bus
- Other External Business and Government Web Services
- Recommended Tools and Platforms
- SOAP
- UDDI
- WSDL
- WS Addressing
- WS-Remote Messaging
- WS-Security
- WS-Trust
- XML and Schemas
- Other Relevant WS* Standards

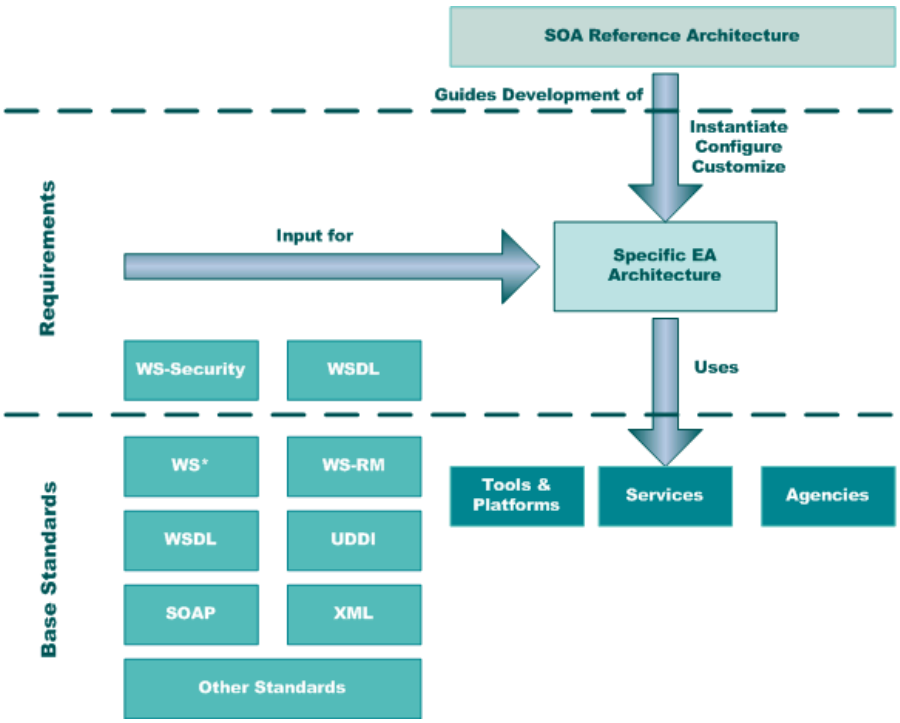

Figure 1. SOA-RA Positioning

Recommendation: These standards must be defined in cooperation with the agencies so that there is a common enterprise approach to the SOA and ESB implementations.

Observation: The RA serves as a pattern for the development of a detailed and tactical SOA for the State.

The SOA Solutions Reference Model

A reference model is used by architects as a template for composing architectures. It is based on a small number of unifying concepts, and may be used as a basis for education and explaining standards. A reference model is not directly tied to any standards, technologies, or other concrete implementation details, but does seek to provide common semantics that can be used unambiguously across and between different implementations. The way in which reference models relate to other work is illustrated in figure 2.[iii]

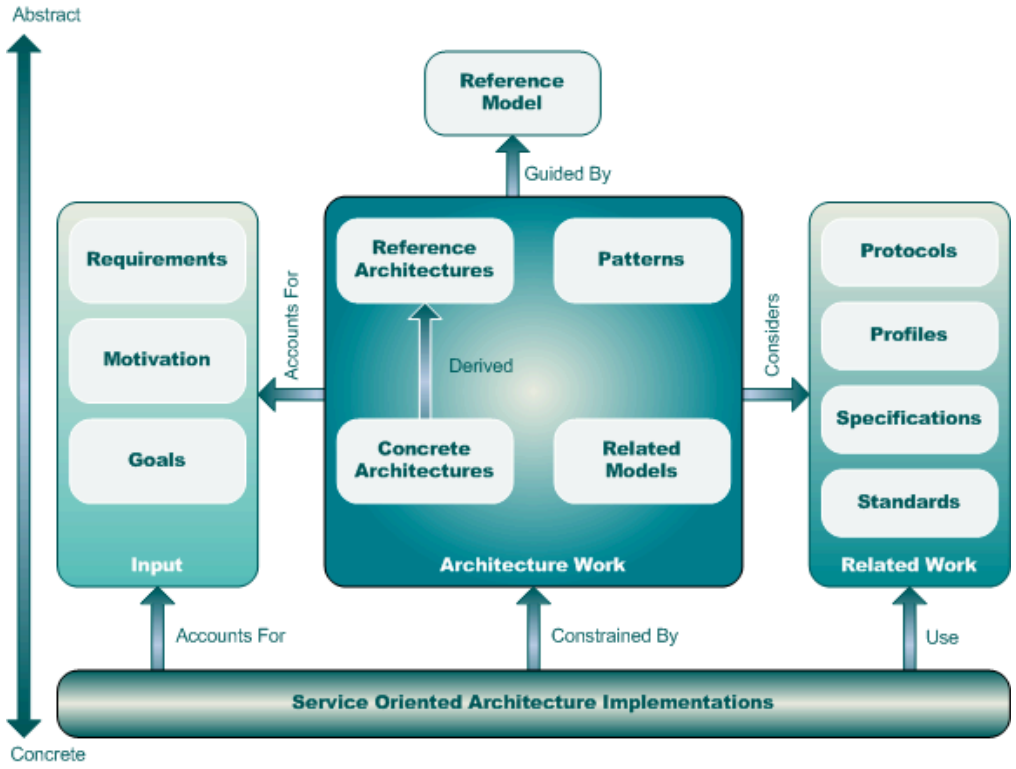

Figure 2. How the Reference Model Relates to Other Work.

The SOA Solutions Reference model illustrated in Figure 3 has been adapted from IBM's work with the Open Group on SOA architecture and reference models.[iv] The SOA solutions reference model consists of:

- A set of layers which contain a set of attributes, responsibilities, and rules associated with that layer.
- Layers tend to have typical interaction mechanisms with one another.
- The set of architectural building blocks that reside in a layer (or constitute that layer) are linked via a set of dependencies, associations, and contracts.

- There are a set of options available at each layer, about which architectural decisions must be made that impact another aspect of the layers.
- An architectural decision concerns itself with the configuration and usage of architectural building blocks.

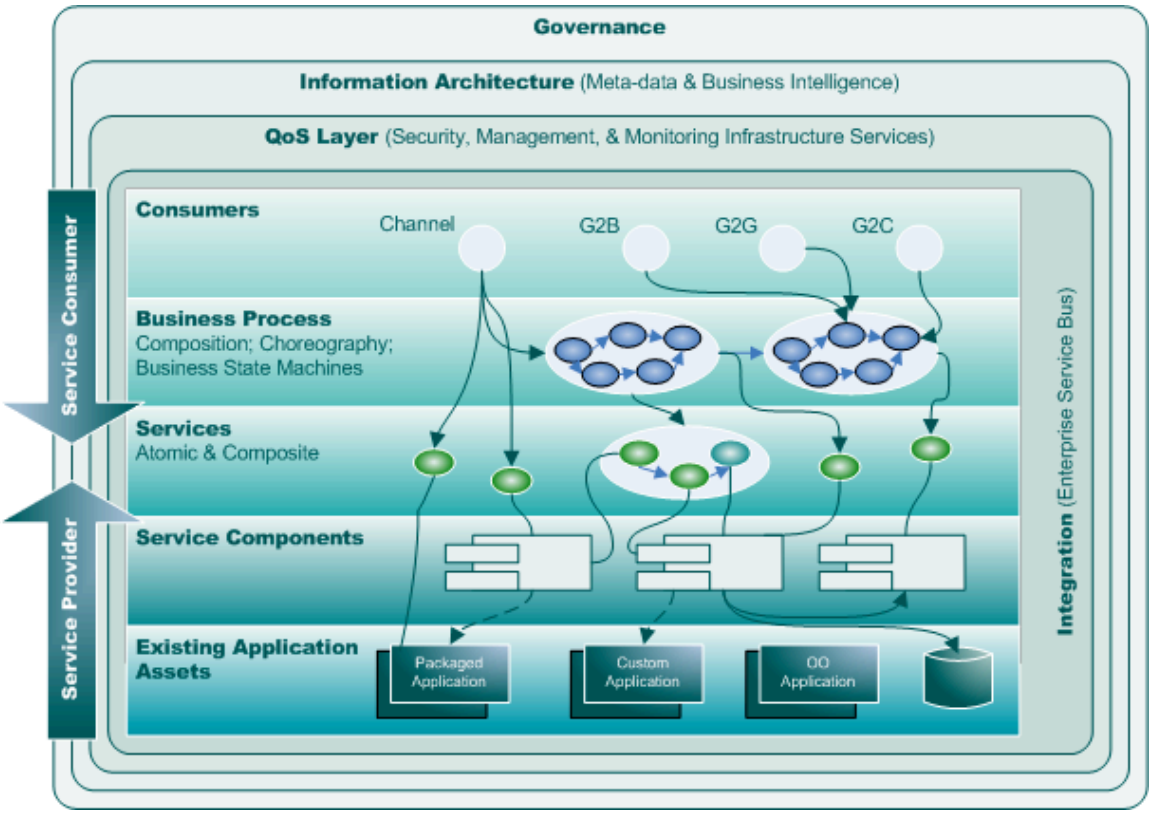

Figure 3: SOA Solutions Reference Model

Layer 1: Operational Systems (Applications and Data)

Layer One seeks to leverage the value of existing IT investments. The operational systems layer illustrated in Figure 4 recognizes the value of existing Information Technology (IT) investment. Related SOA activities include:

- Asset inventory, such as the State's existing information systems database, and other relevant applications' inventories, such as those created by the State IT Assessment project.
- Re-factor existing applications to unlock business value, and identify components, data, and services that could have value to other agencies.
- Identify valuable assets hidden inside an application (e.g., a name search algorithm used in services like the High Level Client Index (HLCI) associated with PACMIS, unique data repositories, etc.).

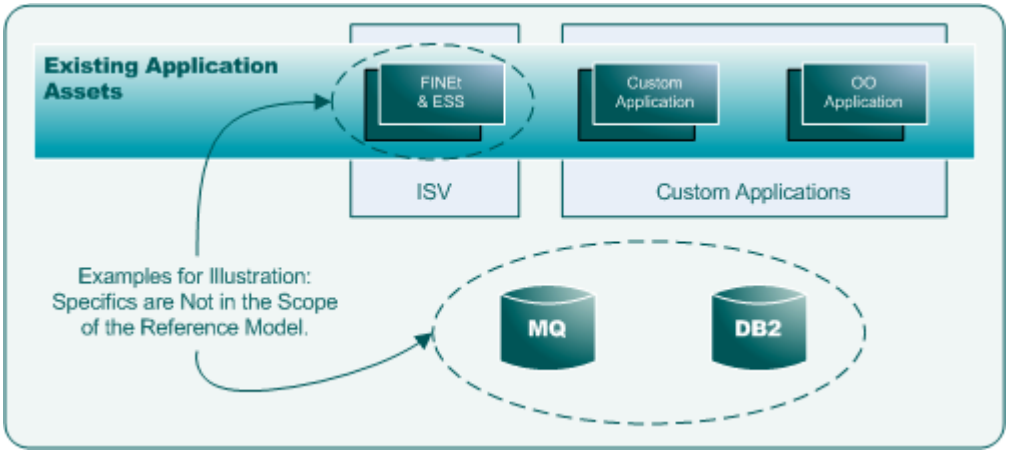

Figure 4. Operational Systems Layer (Applications and Data)

Layer 2: Service Components

The Service Component Layer illustrated in Figure 5 focuses on developing IT flexibility with service facades. This layer enables IT flexibility by strengthening the decoupling in the system. Decoupling is achieved by masking volatile implementation details from consumers. This layer often employs container based technologies like Enterprise Java Beans (EJBs).

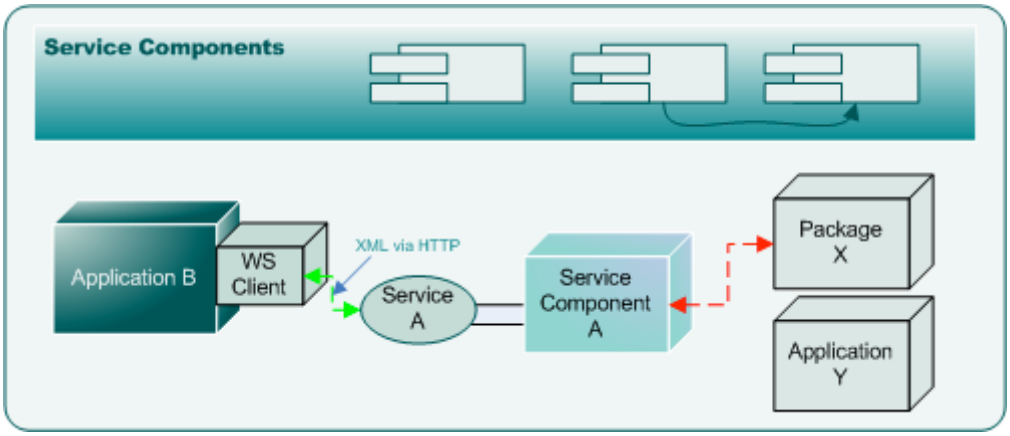

Figure 5. Service Component Layer

Each service component provides an enforcement point for service realization, and offers a facade behind which IT is free to do what is wanted and needed.

Layer 3: Services

The Services Layer in Figure 6 forms the basis for the decoupling of business and IT. This layer captures the functional contract, including Quality of Service (QoS) for each standalone business function or each task in a business process.

It is assumed that within SOA, IT responsibility is to realize and manage service implementations that faithfully conform to the set of services in the service model.

- The service model is a governed asset.
- The services layer is not an operational asset, but is a semantic layer.
- This layer contains all the exposed services in the SOA.
- The services can be “discovered” and invoked, or possibly choreographed into a composition.
- Services are abstract “functions” that are accessible across a network according to specifications captured in the service description.
- Each service is a contract between the consumers and the providers. In a value chain, a service is a contract among all participants.
- The service registry consists of service descriptions, policies, service versions, and SOA management descriptions, as well as attachments that categorize or show service dependencies.

Figure 6. Services Layer (Atomic and Composite Services)

Not all services are created equal. Figure 7 illustrates this concept. Services orientation is a tool, and tools are used for many things. SOA is about business and IT alignment, and represents one use of services, not all uses.

Exposed services are fundamental to SOA, and are subject to governance and management. Each service represents a governed business operation, potentially consumed by several agency business processes and/or partners. Breaking the contract for a service is potentially costly and disruptive to other agency users.

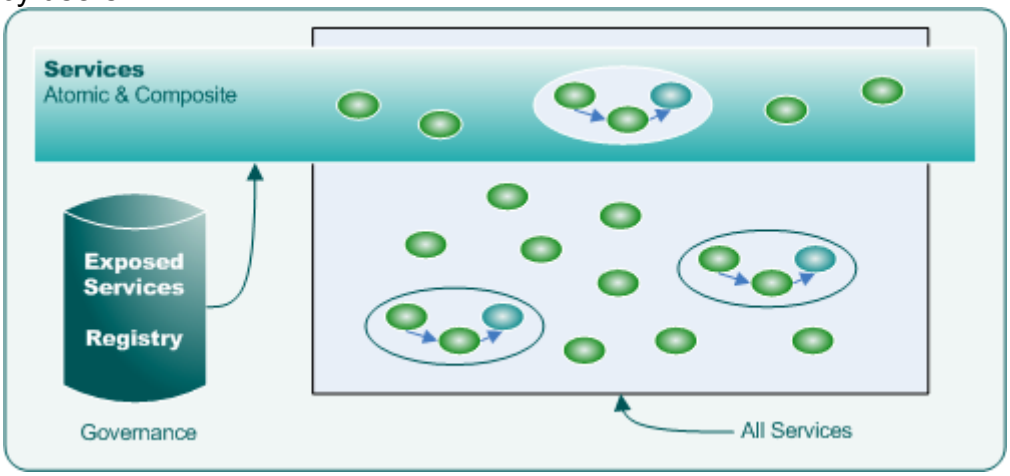

Figure 7. Not all Services are Created Equal

Layer 4: Business Processes (Business Process Alignment of IT)

The Business Process Layer in Figure 8 contains operational IT artifacts that implement business processes as a choreography of services.

- The set of services that are choreographed or composed is restricted to those services that are defined in Layer 3.
- BPEL (Business Process Execution Language) is often used in this layer, but is not a requirement. For example, a Java Bean could be used to choreograph a set of services.
- The choice of technology depends on a set of realization decisions that must be made when establishing a physical reference model for a given SOA. These decisions are typically made based upon requirements and the capabilities of the available alternatives.

Figure 8. Business Process Layer (Composition and Choreography between Business State Machines)

Layer 5: Consumers (Channel Independent Access to Business Processes)

The Consumers Layer in Figure 9 provides channel independent access to business processes. It exists to recognize that the technology chosen to expose business processes and services must permit access from a wide set of potential interaction channels. The State of Utah includes G2G and G2B and G2C as major channel types.

When establishing a physical reference model for a given situation, it is important to populate this layer with the set of channel types that are required in a solution. Each channel type is typically accompanied by limitations and capabilities that will shape the way the physical reference model supports communication with agency business processes and services.

Figure 9. Consumer Layer (Channel Independent Access to Business Processes)

Layers 6 through 9: Cross Cutting Concerns and Capabilities

Several concerns are not restricted to a single layer in the SOA Solutions Reference Model. These concerns are captured in Layers 6 through 9. Cross cutting concerns and capabilities are illustrated in Figure 9.These are not really layers in the same sense as the five layers previously discussed, but treating them as layers provides some ability to focus discussions and decisions. For example:

- What is found where governance intersects services?
- What are the governance concerns specific to services?

There is clear interaction among these layers. For example, it is likely that most data architectures will be subject to governance, as a matter of necessity and common sense.

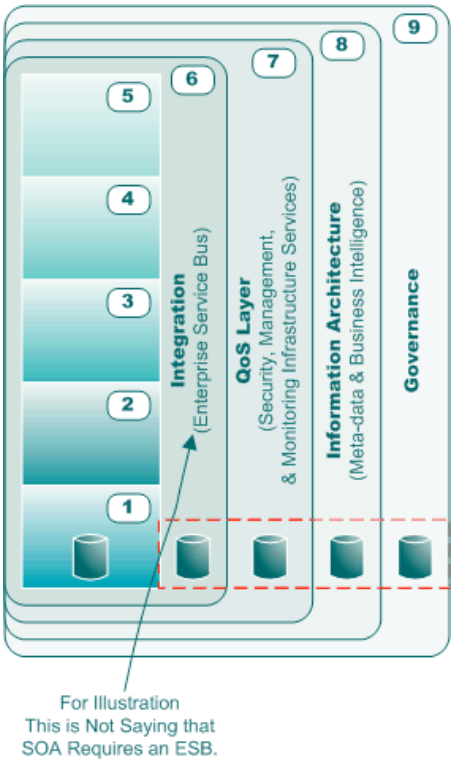

Figure 10. Layers 6-9: Cross Cutting Layers of Concerns and Capabilities

Both the Open Group and OASIS[v] are currently making substantial efforts toward providing a detailed definition of the SOA Reference Architecture in collaboration with major industry providers such as IBM and BEA. RA development is still in its early stages and neither the Open Group nor OASIS have published their detailed reference architectures, and are not expected to do so until sometime in 2008.

SOA Foundation Reference Model

IBM and the Open Group have developed a SOA foundational reference architecture that is illustrated in Figure 11. This forms a useful basis for an RA for the State, especially as the process of applying SOA styles and the implementation of an Enterprise Service Bus (ESB) become high priorities. The reference model establishes the model for SOA from an enterprise perspective, in an architecturally neutral manner.

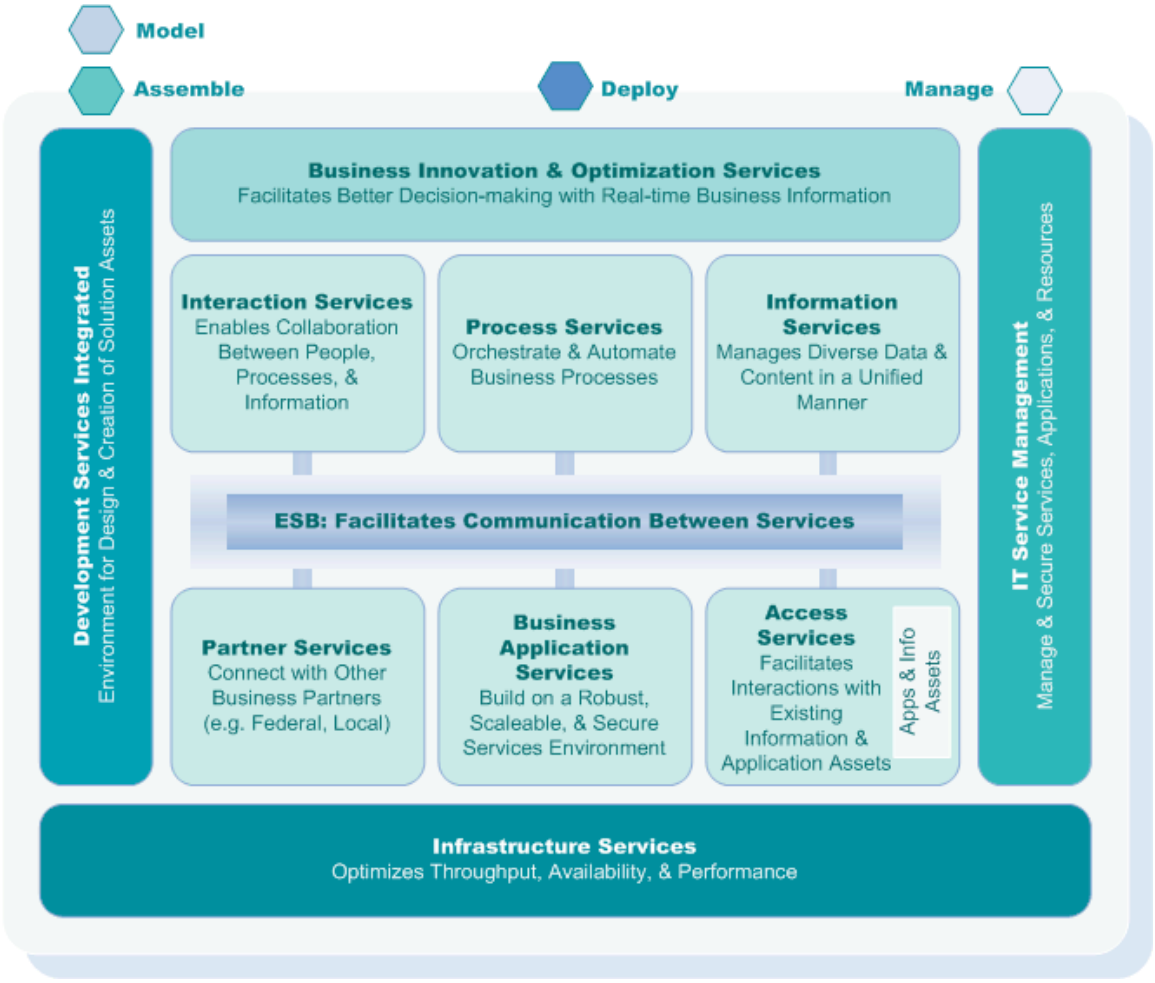

Figure 11. SOA Foundation Reference Model

The model includes the major service types and the relationships between components. The role of the ESB is clearly established to facilitate communication between services. The ESB plays a key role in the SOA-RA. The core principles of the ESB include:

- ESB Inter-connects Requestor and Provider
    - Interactions are Decoupled
    - Supports Key SOA Principle—Separation of Concerns

- ESB Provides Service Virtualization of:
    - Identity via Routing
    - Interaction via Conversion
    - Interface via Transformation
- ESB Enables Aspect Oriented Connectivity
    - Security
    - Management
    - Logging
    - Auditing
- Implemented by a Mediation Framework
    - Offering Pre-built Mediation Pattern (Primitives) Support
    - Enabling Mediation Pattern Composition

Figure 12 represents an ESB centric view of the logical model of the ESB. Characteristics of this logical model include:

- Outside ESB
    - Business Logic (Business Services)
    - ESB Contains Integration Logic or Connectivity Logic
    - Criteria: Semantics Versus Syntax; Aspects
- Loosely Coupled to ESB
    - Security and Management
    - Policy Decision Point Outside the ESB
    - ESB can be a Policy Enforcement Point
- Tightly Coupled to ESB
    - Service Registry
    - Registry is a Policy Decision Point for ESB
    - ESB is a Policy Enforcement Point for Registry
    - Registry has a Broader Scope in SOA
- Tooling Required for ESB
    - Development
    - Administration
    - Configures ESB via Service Registry

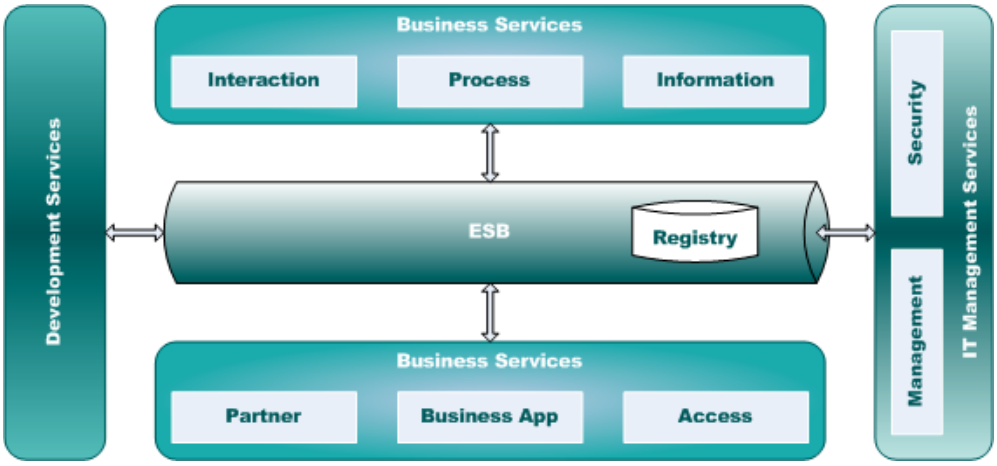

Figure 12. ESB Centric View of the Logical Model

Figure 13 illustrates the role of the ESB using direct, brokered, and federated topology patterns.[vi] The State, over time, is likely to utilize all three of these patterns in the deployment of its ESB. The initial planned pilot SOA implementation uses both the direct and brokered ESB patterns.

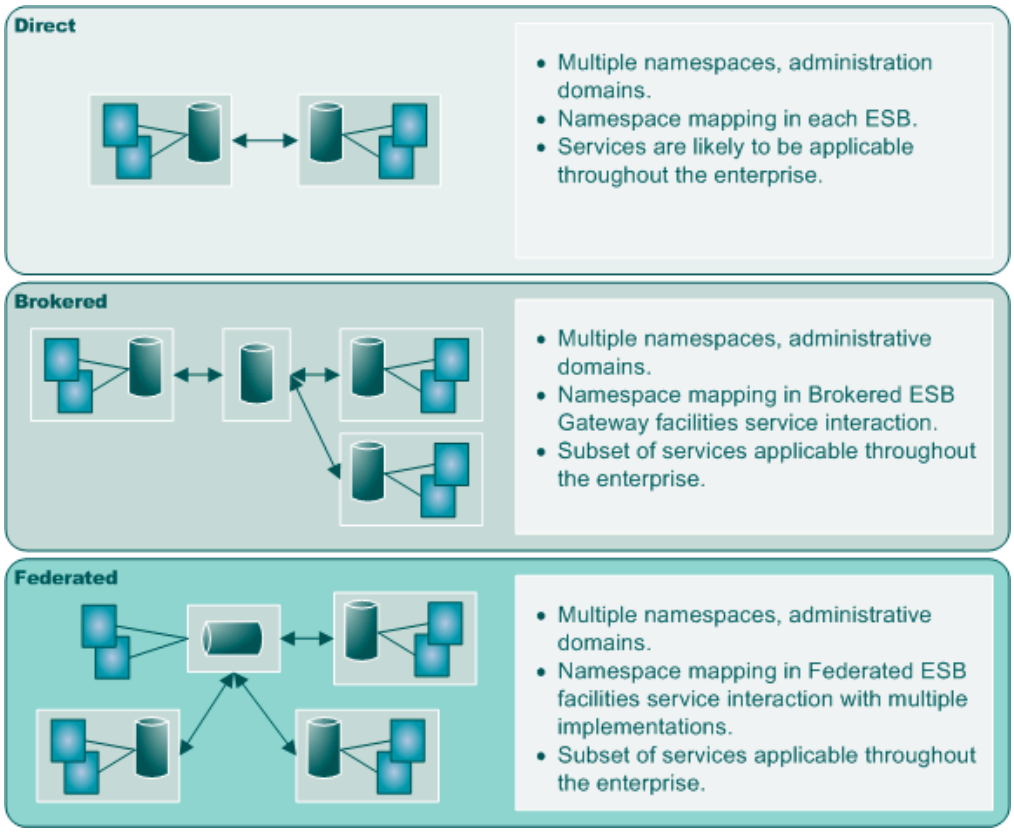

Figure 13. ESB Roles—ESB Integration Topology Patterns

The applicability to the State is illustrated in Figure 13, which addresses mapping the SOA Reference Models to topology. Topology in this context refers to the physical and logical relationship of service components that are connected, located, and accessed over a physical network.

For purposes of simplicity, examples of some of the components which are part of the topology of the State have been illustrated (e.g., CICS, Sun ONE, WAS, etc.). The actual topology is much more diverse, but the mapping concept remains relevant, as does the connectivity to all of the topology components using network services.

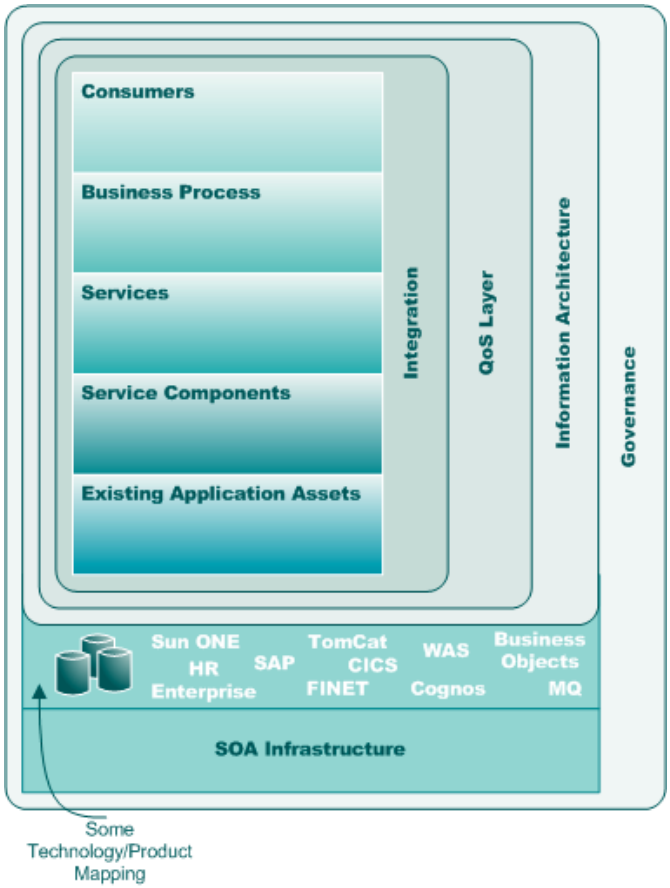

Figure 14. Mapping the SOA Solutions Reference Model to Topology

The SOA Solutions Reference Model layers are overlaid on the topology components. Figure 15 illustrates an overlay of the SOA Foundation Reference Model on the same topology in combination with the SOA Solutions Reference Model layers. Application of the reference models illustrate how the SOA architecture style can articulate and connect what might otherwise be viewed as diverse and abstract ways of thinking about the SOA infrastructure and services.

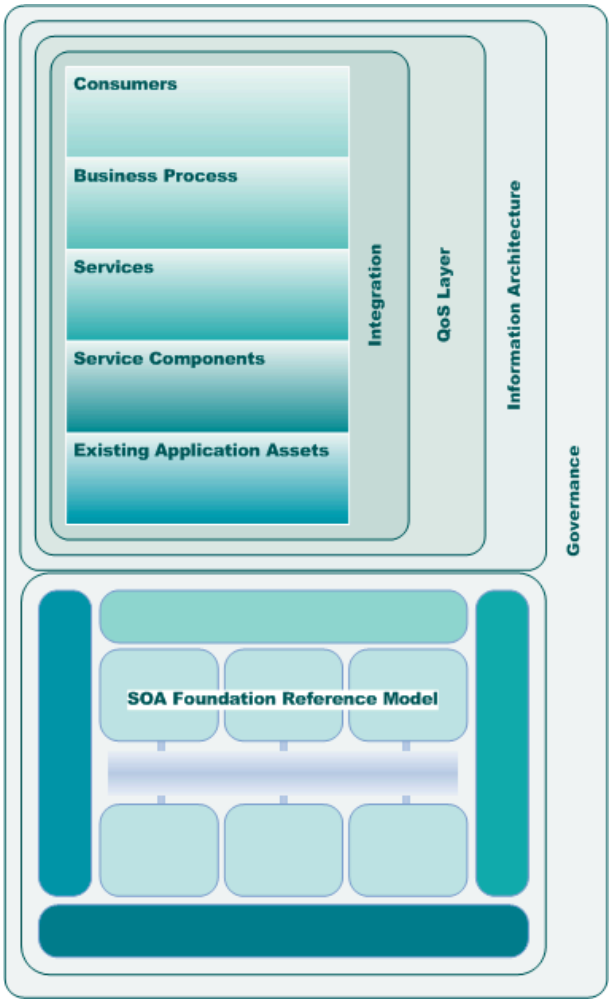

Figure 15. Mapping the SOA Reference Model to Topology and Solutions

Applying the Models to Application Development

An actual application of the reference models using a specific application as a vehicle is illustrated in Figure 16. This application is the Utah GovPay payment processing application used for payment processing on State agency Web sites.

> The Utah GovPay WPS web service provides a secure method to pass customer transaction information between the Agency's web application and Utah GovPay.
>
> The WPS is the backend link into the Utah GovPay system and was designed to prevent Web users from fraudulently altering their own transaction data. WPS has two main functions: registering transactions and querying transactions. In the registration process, the agency's Web application sends the transaction data to WPS and WPS returns a registration ID. The agency's Web application then forwards the user to Utah GovPay with the registration ID.
>
> After a completed payment transaction, the agency's Web application can use the registration ID to query WPS to find out if the transaction was approved.[vii]

The example illustrates the use of the components of the application and the use of service components that ultimately tie back to existing legacy operational systems and data.

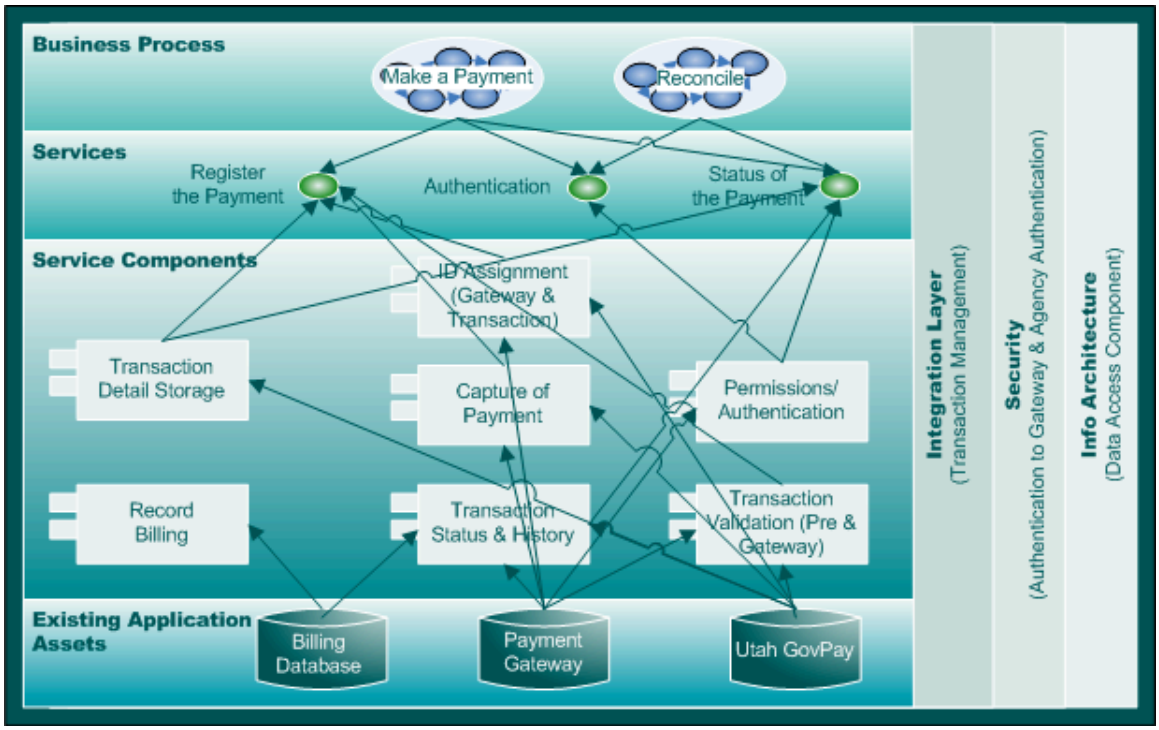

Figure 16. Utah GovPay Web Service Example

Relationship Between SOA and Enterprise Architecture (EA)

Figure 17 illustrates the common issues associated with IT and agency business services and some of the potential benefits of using EA and SOA principles to drive and improve agency business and IT alignment.

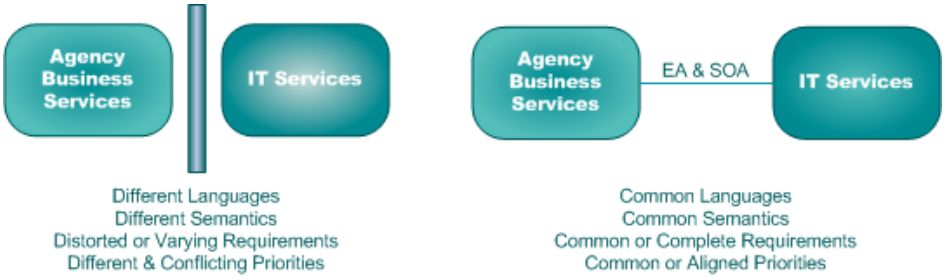

Figure 17. Agency and IT Business Alignment through EA and SOA

Meeting business needs presumes a need for alignment and the removal of gaps to the extent practicable. SOA is a style of EA that has great promise for improving alignment with agency business needs, and is based upon EA principles and patterns. SOA organization and governance patterns and models are not meant to be applied blindly or wholesale. Patterns and models are:

- intended to be inspirational and empowering rather then prescriptive;
- synthesized from many diverse experiences, and each situation is different;
- used by organizations differently, and patterns are used based upon context; and,
- combined and molded to form new patterns and models.

EA and SOA both require effective governance to be successful. An EA approach to SOA identifies possible areas for reuse. A solutions approach to SOA identifies what services are needed for the SOA solution. Both approaches add substantial value to an effective SOA implementation. The EA models incorporated in this paper provide a useful basis for defining a tactically oriented SOA architecture for the State.

REFERENCES

Arsanjani, Ali, and Jorge Diaz, *SOA Reference Architecture: Concepts and Usage*, IBM: SOA Center of Excellence, July 25, 2007.

Bennett, Stephen, *Best Practices and Patterns from the field for SOA Governance*, Open Group Architecture Practitioners Conference, July 23-25, 2007, Austin Texas.

Bringing SOA Value Patterns to Life, Oracle White Paper, June 2006

Dico, Awel, *Addressing Enterprise Business Needs through SOA and TOGAF*, Open Group Architecture Practitioners Conference, July 23-25, 2007, Austin Texas.

Gejnevall, Matt, *How EA and SOA Connect*, Open Group Architecture Practitioners Conference, July 23-25, 2007, Austin Texas.

High, Rob, Jr. et al, *IBM's SOA Foundation: An Architectural Introduction and Overview.* Version 1.0, November, 2005. http://download.boulder.ibm.com/ibmdl/pub/software/dw/webservices/ws-soa-whitepaper.pdf

Keen, Martin et al*, Patterns: SOA Foundation -Business Process Management Scenario,* Redbook*,* http://www.redbooks.ibm.com/redbooks/pdfs/sg247234.pdf

----------*, Patterns: SOA with an Enterprise Service Bus,* Redbook*,* http://www.redbooks.ibm.com/redbooks/pdfs/sg246494.pdhttp://www-128.ibm.com/developerworks/library/ws-model7

Reference Model for Service Oriented Architecture 1.0, Committee Specification 1, 2 August 2006, Copyright © OASIS Open 2005-2006. All Rights Reserved.

Sides, Jim, *Applying Patterns, Platform Intelligence and Products to ESB Deployments*, IBM, Open Group Architecture Practitioners Conference, July 23-25, 2007, Austin Texas.

Sudarsan, Sridhar*, Adopting SOA - Aligning the business and IT*, IBM, Open Group Architecture Practitioners Conference, July 23-25, 2007, Austin Texas.

TOGAF 2006 Edition (Incorporating 8.1.1), The Open Group, August 2006.

Utah GovPay: The Official Payment Solution for Utah Government (Terchnical Manual), Utah Interactive: Salt Lake City, 2006.

Definitions

.NET
A comprehensive software development platform from Microsoft that was introduced in 2000 as the company's next generation programming environment.

Adapter API
An adapter used as a Web service with an application's API.

API
Application Programming Interface is a language and message format used by an application program to communicate with the operating system or some other control program such as a database management system (DBMS) or communications protocol.

BPEL
Business Process Execution Language is an XML-based language developed by IBM, BEA Systems, Microsoft, and others for defining Web services business processes.

DTD
Document Type Definition is a language that describes the contents of an SGML or XML document.

EDA
Enterprise Data Access provides a uniform way to access data throughout the enterprise. It implies the ability to treat multiple, distributed databases as a single logical entity.

EJB
Enterprise Java Bean is a software component in Sun's J2EE platform, which provides a pure Java environment for developing and running distributed applications.

ESB
Enterprise Services Bus is a message and/or integration broker that supports Web services.

J2EE
Java 2 Platform, Enterprise Edition is a platform from Sun for building distributed enterprise applications.

JAAS
Java Authentication and Authorization Service is an API that enables Java applications to access authentication and access control services without being tied to those services.

Java
An object-oriented programming language that is platform independent.

JDBC
Java DataBase Connectivity is a programming interface that lets Java applications access a database via the SQL language.

Namespace
A collection of names for a particular purpose. Typically, each name is unique. For example, tables in a relational database must all have unique names. The Internet uses a hierarchical namespace that partitions names into top level domains (TLDs) such as .com, .net, and .gov. The Internet namespace includes all the domain names (utah.gov, google.com, Comcast.net, etc.) that fall into these top level domain categories.

ODBC
Open DataBase Connectivity is a database programming interface from Microsoft that provides a common language for Windows applications to access databases on a network.

OO
Refers to object oriented programming language environments.

REST
Representational State Transfer Web services are resource oriented Web services. Resource-oriented services focus on distinct data objects upon which a handful of basic, standard operations can be performed.

RPC
Remote Procedure Call is a programming interface that allows one program to use the services of another program in a remote machine.

SNMP
Simple Network Management Protocol is a widely used network monitoring and control protocol.

SOA
Service-Oriented Architecture was formerly called a "distributed objects" architecture; the SOA term was coined as Web services were evolving.

SOAP
Simple Object Access Protocol is a message-based protocol based on XML for accessing services on the Web.

UDDI

Universal Description Discovery Integration is designed to enable software to automatically discover and integrate with services on the Web.

Session
The active connection between a user and a computer, or between two computers, and from an application use perspective: the period between starting up and quitting the application.

State
In object-oriented programming, the state of an object is the combination of the original values in the object plus any modifications made to them. From a Web perspective, state refers to the current or last-known status, or condition, of a process, transaction, or setting. "Maintaining state" or "managing state" means keeping track of the process.

Web Services
Web-based applications that dynamically interact with other Web applications using open standards that include XML, UDDI, and SOAP.

WSDL
Web Services Description Language is an XML-based language for defining Web services.

WS-REL
Web Services-Reliability defines an open, interoperable wire protocol for reliable messaging based on the SOAP protocol.

WSRM
Web Services Reliable Messaging specifies a generic and open model for ensuring reliable message delivery for Web services.

WSRP
Web Services for Remote Portlets are dynamic plug-ins for portal pages. WSRP defines how to plug remote Web services into the pages of online portals and other user-facing applications.

XML
EXtensible Markup Language is an open standard for describing data from the W3C.

Endnotes:

[i] *Reference Model for Service Oriented Architecture 1.0, Committee Specification 1, 2 August 2006,* Copyright © OASIS Open 2005-2006. All Rights Reserved.

[ii] *Bringing SOA Value Patterns to Life*, Oracle White Paper, June 2006
[iii] *Ibid.*
[iv] Arsanjani, Ali, and Jorge Diaz, *SOA Reference Architecture: Concepts and Usage*, IBM: SOA Center of Excellence, July 25, 2007.
[v] *OASIS SOA Reference Model TC* at http://www.oasis open.org/committees/tc_home.php?wg_abbrev=soa-rm

[vi] Sides, Jim, *Applying Patterns, Platform Intelligence and Products to ESB Deployments*, IBM, Open Group Architecture Practitioners Conference, July 23-25, 2007, Austin Texas.

[vii] *Utah GovPay: The Official Payment Solution for Utah Government (Terchnical Manual),* Utah Interactive: Salt Lake City, 2006.