DOMAIN MODEL FOR SOA

Realizing the Business Benefit of Service-Oriented Architecture

VERSION 1.0

COPYRIGHT

RESTRICTED RIGHTS LEGEND

TRADEMARKS

CONTENTS

TABLE OF FIGURES

INTRODUCTION

The goal of a Service-Oriented Architecture (SOA) is to enable organizations to realize business and technology advantages through a combination of process innovation, effective governance, and a technology strategy that revolves around the definition and re-use of services.

While SOA is a long-term strategy that requires sustained focus on transforming the way IT is delivered, it must respond to immediate business initiatives. In consequence, the benefits of SOA will only be realized if you preserve a balance between the long-term goals and the shorter-term needs of the business. You can maintain that balance by instituting a set of organizational, financial, operational, design and delivery practices from the outset of the SOA initiative.

BEA's Domain Model for SOA encapsulates these practices within six domains, each of which must be considered equally to provide a focused framework for a Service-Oriented Architecture.

The six domains, while distinct, are interrelated and interdependent. Executing on each domain with equal focus is fundamental to the success of an enterprise-wide SOA initiative.

This paper describes the challenges encountered in each domain and the practices that must be in place for a successful SOA implementation.

WHAT IS SOA?

Service-Oriented Architecture is an IT strategy that organizes the discrete functions contained in enterprise applications into interoperable, standards-based services that can be combined and reused quickly to meet business needs.

By organizing enterprise IT around services instead of around applications, SOA provides key benefits:

- Improves productivity, agility and speed for both Business and IT
- Allows IT to deliver services faster and align closer with business
- Allows the business to respond quicker and deliver optimal user experience.

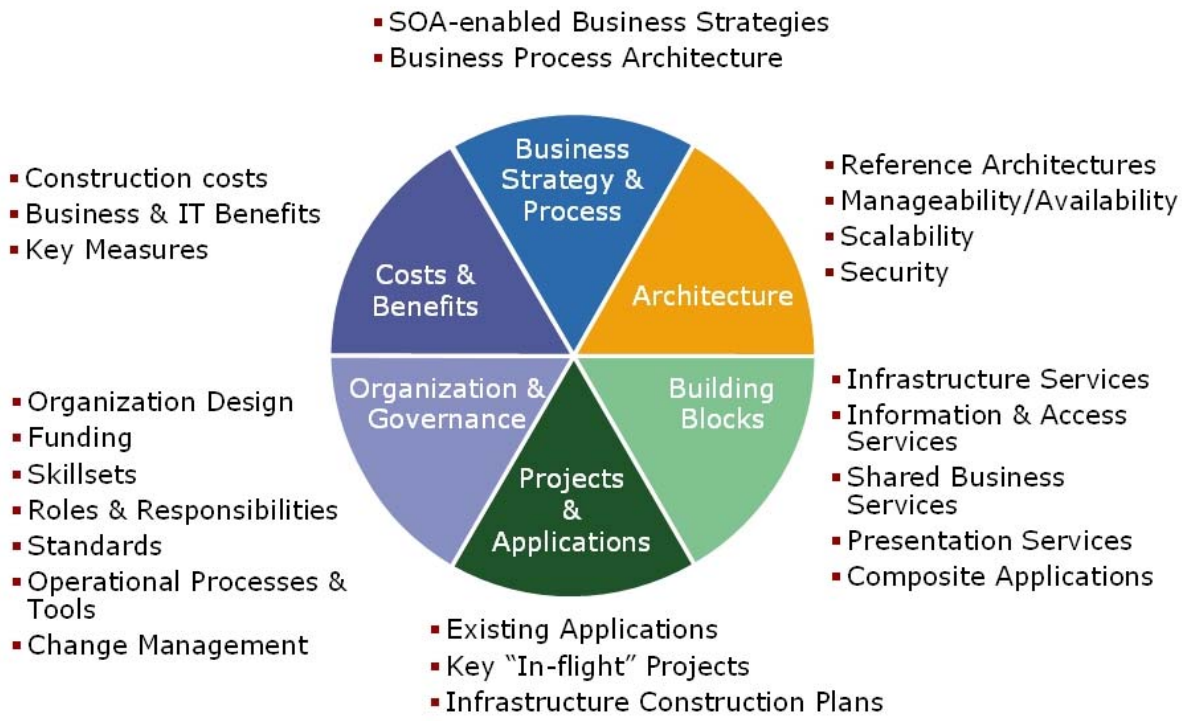

FIGURE 1: THE SIX SOA DOMAINS

THE SIX DOMAINS – RESPONDING TO CHALLENGES

Each of the six domains addresses a challenge to successfully delivering a Service-Oriented Architecture.

1. **Business Strategy and Process:**

Challenge: Providing IT implementations that support the business and its changing needs.

Response: Provide an environment that links the management and measurement of IT with the business strategy and empowers both to work together for continual process improvement.

2. **Architecture:**

Challenge: Nearly all enterprises fund and build IT by projects in lines of business, leaving enterprise-wide processes and integration to be considered as afterthoughts and creating barriers to change.

Response: An IT environment based on standards, distribution, loose coupling and business process representation that is designed to respond to change and integrate functionality at an enterprise level.

3. **Building Blocks:**

Challenge: Lack of consistency and repeatability in IT implementation hinders most enterprises in achieving their goals with respect to IT budgets and agility.

Response: A common, standards-based foundation on which to deliver IT provides a basis to achieve consistency and maximizes the ability to repeat successes through reuse of implementations and core infrastructure.

4. **Projects and Applications:**

Challenge: IT is traditionally developed by projects within lines of business creating situations where excessive capital is spent duplicating functionality and the integrity of enterprise processes is compromised.

Response: Catalog, categorize, and re-factor functionality offered by the systems and applications in the enterprise to standardize the manner in which that functionality is offered while driving out redundancy and promoting consistency in business execution.

5. **Organization and Governance:**

Challenge: The organic growth of enterprises through the creation of individual solutions for new requirements yields IT architectures that are difficult and costly to change.

Response: An organizational structure and mandate to govern and standardize the delivery of IT to assure IT meets business needs and maximizes the use of developed functionality.

6. **Costs and Benefits:**

Challenge: The cost of IT for the benefit received is a constant source of friction between IT organizations and the business they support.

Response: Plan and execute IT implementations to create early and sustainable value that leverages existing investments in IT while accommodating change and growth.

BUSINESS STRATEGY AND PROCESS

Information Technology plays a vital role in delivering the information business needs to make and act on decisions. Yet frequently, IT is not viewed as a strategic partner to the business and is often perceived as slow to deliver the productivity gains that the business requires.

This gap between business need and IT execution often has less to do with the technology leveraged than the way in which IT as a whole is delivered to the business. Often, companies develop a business strategy and then attempt to implement it through a separate IT strategy.

In addition, business and IT strategies usually work on separate timescales. While the business strategy may consider the entire enterprise for the long-term, the IT strategy usually addresses the shorter-term need of an individual line of business. When IT is also funded on a line of business basis, the chasm between the two strategies widens. The result is a siloed IT program that does not support the enterprise as a whole.

The challenge in achieving productivity gains, is to close the gap between business need and IT execution. This requires fundamental changes to the alignment of IT and business strategies.

In implementing our first generation SOA, BEA found that it was indeed the alignment of IT and business strategy through an SOA program that closed the gap between business need and IT execution.

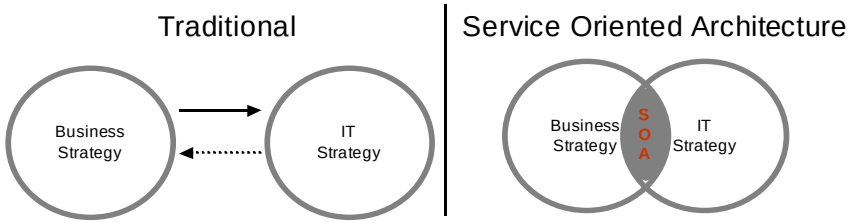

FIGURE 2: TRADITIONAL IT VS. SOA APPROACH

SOA PROGRAM

SOA requires change in the culture of the organization: in the way that people work together, in how they think about architecture and how they deliver functionality to the business. An SOA Program, with strong sponsorship and high-level visibility in IT and the business, is required to see this change implemented across the enterprise. The SOA Program drives change in a number of ways by:

- promoting the sharing and understanding of the whole business strategy, so that decisions are made with an enterprise level view and not just a line of business view
- owning the overall SOA strategy of the enterprise so that appropriate emphasis is given to each of the six inter-related domains of SOA throughout a multi-year roadmap
- defining the architecture to be dynamic, responsive and standards-based

- ensuring cost-effective delivery through identifying and optimizing business processes that can be divided into re-useable shared services, and avoiding duplication of functionality by mining legacy applications for those services
- deciding on the priorities for service development and deployment and choosing the increments and when they will be delivered
- establishing the appropriate organization and governance to ensure that processes, policies and standards are followed
- encouraging change and adoption through publicity and incentive setting
- ensuring that appropriate measurements are put in place to provide constant cost-benefit analysis and provide a continuous feedback loop to interrogate the viability of the program.

BUSINESS PROCESS OPTIMIZATION

SOA makes IT a full partner in enterprise strategy. As a partner, it is possible to view IT as the concrete expression of enterprise business processes rather than a disjointed set of systems that each represents a fragment of a business process. In turn, business processes become fully encapsulated by IT which enables IT to institutionalize the necessary measurement and accountability to the business. IT can develop the responsiveness that the business demands because as an SOA matures, it becomes able to deliver functionality purely by extending business processes rather than by building stand-alone systems and applications.

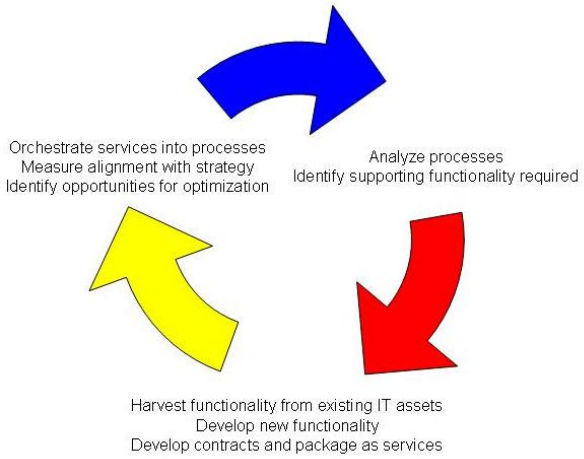

FIGURE 3: PROCESS OPTIMIZATION FEEDBACK LOOP

The study of the existing business processes, systems, and short, medium and long term IT development roadmaps will identify processes to prioritize on a roadmap for the SOA initiative. This analysis is a collaborative, jointly owned effort between the business and IT, that leverages the business strategy to prioritize activities. It starts a continuous feedback loop for alignment with the business, and maximizes the benefit of existing IT investment. *Figure 3* illustrates this feedback loop.

As business processes are inspected, supporting IT functionality is uncovered. This functionality may be found in existing applications, or may require new development. Business-oriented, standards-based functionality will be supplied to the enterprise with contracts that govern usage in the form of services.

Business processes will then be expressed through a set of services that are brought together (*orchestrated*) to comprise the business process. The contracts that govern the services provide the mechanisms to measure overall performance, performance against key business indicators, and compliance with service level agreements. These metrics will expose improvement opportunities completing a feedback loop and aligning IT with the business.

Optimization is not a one-off project but the execution of a multi-year roadmap through the SOA Program. This process will require a number of cycles over time. The resulting standardization of IT

delivery to the enterprise through services is achieved in manageable steps in accordance with business goals, while reaping short, medium, and long-term benefits.

SOA is a fundamental change to the way that IT is delivered. Gains in agility realized from early, ad-hoc use of an SOA strategy in lines of business will encourage close collaboration between the business and IT to realize the benefits of the change throughout the enterprise. As IT becomes better able to express the business in technology terms, it will foster and create the sponsorship required from the business to reach maturity of SOA as an enterprise-wide strategic partnership with IT. Rather than have IT simply translate requirements into a set of disconnected systems and applications to support lines of business, SOA will enable the enterprise to be truly innovative and adapt quickly to changing environments.

ARCHITECTURE

A technology architecture defines how functionality is delivered and deployed for the benefit of a business and its users. Building architectures that can support a rapid pace of change is a key consideration that conventional IT delivery techniques do not adequately address.

For IT architecture to respond as rapidly as the business requires, the role of architecture itself needs to change. Service-Oriented Architecture is the way to provide that change. SOA has defining characteristics that fundamentally differ from the manner in which architecture is defined today in most large enterprises. These characteristics are ideally suited to support faster change and closer alignment between the business and IT in the enterprise.

SERVICE BASED

IT is typically acquired or developed in response to the needs of a particular business segment and deployed solely for the benefit of that segment. Since IT tends to be funded and constructed by projects whose purpose is to address a given set of requirements, functionality tends to be duplicated. Traditional approaches to sharing functionality at a code or component level have failed due to this project-by-project focus of development activities.

A service-based approach to IT changes the way in which functionality is developed and delivered. Functionality is considered, factored, and deployed once for use at all levels of the enterprise yielding the associated benefits of reduced cost, faster delivery, and IT responsiveness to the needs of change. In addition to requiring differences in the way that IT is funded and governed, a service-based approach requires a change in the way that functionality is packaged and deployed. SOA also considers the manner in which functionality is made available as services, and the way in which those services are managed and monitored.

STANDARDS BASED

Another aspect of traditional IT delivery is that each project typically chooses the most expedient method to satisfy its requirements. This leads to a proliferation of technologies that is problematic when deliberating how applications built on those technologies will exchange information. Previous attempts at standards-based component models like CORBA and DCOM suffered because of the technology required to execute them, slow development of supporting standards, or both. More recent developments like XML, Web Services, and UDDI provide the foundation for a standards-based SOA that supports re-use, and for which the technology required to support the standards is readily available and truly platform agnostic.

ENTERPRISE FOCUS

The development of IT through projects within lines of business makes the visibility and management of process or information that spans the enterprise extremely difficult. Many organizations responded to this deficiency by forming enterprise architecture groups or committees. These groups are usually focused on technology selection and do not have the authority to enforce other recommendations. In addition to a stronger mandate for governance, these groups need a mechanism to define, deploy, monitor, and manage access to enterprise functionality in a standard way and at the right levels of granularity and visibility for the user communities. Only a properly constructed service-based enterprise architecture with appropriately enforced governance principles can provide the needed deployment platform.

BUSINESS FOCUS

In most enterprises, business users require dozens of applications to perform their daily activities. This again is a by-product of traditional IT delivery, where individual applications are created for different sets of requirements. This leads to wasted effort, increased training overhead, over-reliance on specialist skills, duplicate data entry, and a lack of visibility and control of overall business process. SOA is aimed at providing functionality to the business at the level where business users conceive of the business, making it easier for the user to understand, specify, test, and operate on a daily basis.

REFERENCE ARCHITECTURE

The unique attributes of SOA and its top-down approach to offering functionality make it possible to define a high-level enterprise architecture that is almost universally applicable. This high level, or reference, architecture describes the major components of an SOA and their relationships to one another. The SOA reference architecture is depicted in the following diagram:

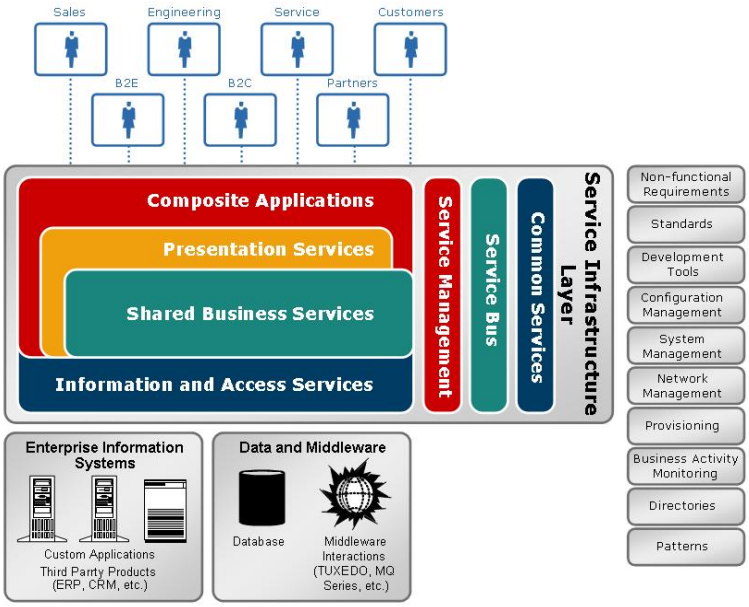

FIGURE 4: SOA REFERENCE ARCHITECTURE

The architecture separates the users of enterprise functionality from the systems and applications that provide that functionality placing the infrastructure for services and service delivery between them. The layers of services and their supporting infrastructure are referred to as the “Services Infrastructure Layer.” This analogy expresses their role in driving change in the way that IT is delivered. Existing applications, data, and middleware form the foundation from which services are drawn.

Supporting and formalizing the existing enterprise activities is a service-oriented infrastructure. Standards-based infrastructure services provide a common basis for the deployment of all other types of service. Infrastructure services include service management to provide location independence, fail over, management, and other enterprise-quality-of-service attributes. A service bus provides common routing and transformation capabilities in much the same way as a traditional message broker or bus in common middleware solutions. Other common services, like logging, auditing, security, and error handling are provided that can be leveraged by enterprise services to standardize delivery of these core capabilities. These shared services are deployed in a shared infrastructure. Shared infrastructure is a new concept in many enterprises, but is key to successfully constructing a service-oriented enterprise.

The information and access services layer represents the functionality of the existing enterprise. These services are created, or harvested from existing enterprise resources, and deployed on the shared infrastructure to ensure enterprise-class quality of service. This layer standardizes access to functionality and information offered in the enterprise. It encapsulates this functionality and information so that consumers need no foreknowledge of the technology or implementation that underlies the delivery of a service. Employing these key concepts provides a common foundation to access enterprise resources on which to build higher-order and higher-value services targeted at job functions and user communities.

The shared business services layer represents the core functionality of the business. It is distinguished from the information and access services layer in that it provides functionality that operates on information, rather than just providing the information itself. In other words, the shared business services layer leverages the services in the information and access services layer to provide common business functions. For example, if an employee is represented by an enterprise entity that is encapsulated as an information service, a shared scheduling business service could leverage that employee information service to obtain information to modify the employee’s schedule.

The presentation services layer represents common presentation components that use shared business information and access services to interact with enterprise resources. This layer provides reusable presentation. For example, an account information portlet might represent a presentation service that is used in a customer self-service portal as well as a customer service portal. The account information portlet might make use of a customer information service to get customer information for display.

The composite applications layer orchestrates other services and functions to deliver high order application functionality for the business. This layer represents business functionality in the way that business users think about and expect to use technology. A customer service portal would be a composite application, as would a new product introduction process. These applications embody business process, and allow that process to be managed and measured to provide tight alignment with business needs and expectations. The true benefits of the “Services Infrastructure Layer” are realized as business users, working seamlessly with their IT colleagues, build genuinely cross-divisional functionality that provides groundbreaking return on investment.

Aside from services layers and the shared infrastructure on which they are deployed, other technology requirements, and disciplines must be addressed to satisfy the needs of the overall architecture. Development disciplines like packaging, deployment, versioning, and change management must be standardized and enforced to provide consistency for the shared service-oriented platform. Deployment platform considerations like reliability and availability must be accounted for in order to provide the expected enterprise-class quality of service.

COSTS AND BENEFITS

Justifying an SOA program is not the same as justifying traditional software projects because SOA delivers its benefits in a range of different ways on an enterprise-wide scale. Through innovations made possible by optimizing business process with shared services, SOA offers value opportunities that are disproportionately high when compared to those expected from traditional software projects. The ability of business to innovate through SOA is a key differentiator in building a strong business case. The business case must also take into account that the up-front cost of establishing an SOA Program yields benefits that accumulate and accelerate substantially over time.

THE BENEFITS OF SOA

On the business side, continual process improvement and business-oriented delivery of functionality through services is made possible by increased collaboration between business and IT stakeholders. Benefits to keep in mind include how IT accountability to the business strategy is improved and how the cost and benefit of functionality can be tracked on a service-by-service basis.

On the IT side, key factors to consider include improvements in delivery capability through incremental deployment, the reuse of services for faster deployment, standardization, benefits in skill set portability, and lower skill set requirements in a standardized environment. An area that warrants special attention is shared infrastructure. A common platform for service deployment yields reliability, availability, scalability, and performance along with common measurement, management, and core functionality.

The greatest impact of an SOA program will typically be in business areas. Given that the average IT budget comprises only 5-11% of the enterprise budget, the gains available from cost containment and higher productivity pale in comparison to the opportunities available from increased business payback. The potential business payback vs. IT payback is shown in an example from a BEA customer.

Business Payback		IT Payback	
Results tied to critical business issues		Results tied to critical IT issues	
Primary		Secondary	
Revenue	Increased due to product time to market and custom profiling	Back Office Reductions	30-40% cost reduction
Customer Loyalty	75% improvement in retention	Retired Legacy Systems	200+ Applications
Launch Cost per Offer	20% reduction in launch cost	Application Development	On time and on budget
Reporting	Improved customer analytics and campaign metrics	Reuse	On target at 30%

FIGURE 5: SOA JUSTIFICATION

The business, a global banking firm, identified the key drivers of their business strategy, and thus the goals for their SOA Program, and tracked progress during implementation. The IT results were very significant. Over 200 applications were retired eliminating costs to support and maintain duplicate functionality. More remarkable, and more important to the SOA Program sponsors, were the business goals achieved. These included a 75% improvement in customer retention due to improved ability to bring services and capabilities to market through online channels.

Enterprises that have adopted SOA use a number of approaches for the initial and ongoing justification of investment in their SOA Program. Some of the metrics are technically based and quantitatively complex, while others focus upon a simple framework for aligning IT value to business value. If these metrics are identified at the beginning of the SOA planning process, work can be prioritized to realize early value.

The goals and strategy of the business, taken together with the inventory of available functionality and the IT activities required to support the strategy, yield the information necessary to develop a roadmap for SOA implementation that prioritizes for *value*. The execution of this roadmap is the joint responsibility of the business stakeholders and IT. Prioritizing for value provides the early gains necessary to ensure the long-term sustainability of the SOA program.

MANAGING THE COSTS OF SOA

The benefits of an SOA initiative are measured over years rather than months. To lay the foundation for change an initial investment in an SOA Program, with people and technology to support it, is required. As services begin to be used, the return on that investment is manifested through reuse of standard functionality, operational improvement, decommissioning of older applications, and similar IT-related payback factors.

As more services become available for reuse, IT's ability to deliver new capability to capitalize on emerging market opportunities will provide substantial results for the business. Therefore, the initial impact of investing in SOA can be minimized by carefully selecting the right capabilities to spearhead the migration to SOA. These will, over time, change the cost structure of IT investments as illustrated in *Figure 6: SOA vs. Traditional Delivery Cost Structure.*

The diagram depicts BEA's experience in its own SOA transformation as well as the experience of some early SOA adopters. As IT capabilities are delivered to the enterprise via shared services, the incremental cost to deliver these capabilities is offset by the use of shared infrastructure and common, standards-based practices. In conventional delivery techniques, each new set of requirements results in the deployment of new applications, each requiring its own infrastructure, limiting opportunities for reuse, and making integration difficult and IT accountability to the business strategy all but impossible, each factor of which is a significant contributor to the cost of IT.

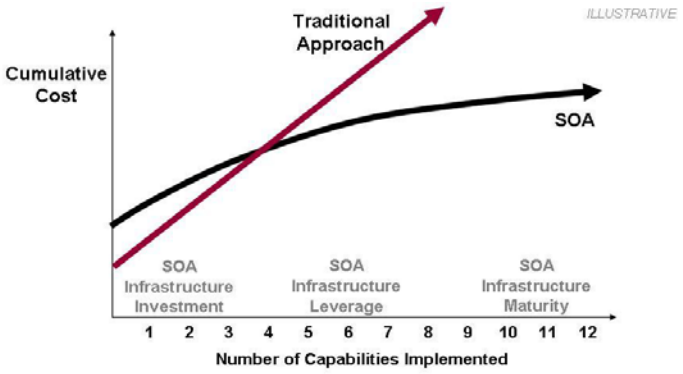

FIGURE 6: SOA VS. TRADITIONAL DELIVERY COST STRUCTURE

As it matures, an SOA inherently develops the ability to measure and contain the cost of IT. Functionality, in the form of services, is standardized for a shared infrastructure that provides the ability to measure service usage and, in consequence, effective monitoring of development and delivery costs. This allows measurement of the extent of re-use and concrete comparison with traditional software delivery, to track cost containment and identify opportunities for improvement. Although the larger proportion of the benefits of SOA are realized on the business side, the ability to concretely measure the ROI of incremental investments in IT is a groundbreaking differentiator.

PROJECTS AND APPLICATIONS

Once an SOA reference architecture and program structure is in place, it is time to consider which services are needed to provide the functionality and return on investment required by the enterprise and when, and in what sequence, they will be deployed. This plan is a Service Roadmap that guides and prioritizes the work of harvesting, or developing, and refining services.

The Service Roadmap begins with identifying the IT *projects* that are already in plan and identifying functionality that already exists in the enterprise's portfolio of applications to determine which existing services in the architecture can be implemented or harvested. Next, the enterprise needs to develop and prioritize the projects that will complete the architecture. The SOA Program will also support the development of many individual projects that deliver business value. These are the projects that gain new customers, gain additional value from existing customers, or allow businesses to create new products and provide new services. Effective planning of those projects to deliver the shared services they provide is a key discipline to ensure success for the SOA Program.

CURRENT PROJECTS AND APPLICATIONS

At the start of most SOA Programs, the enterprise is a heterogeneous environment of technical infrastructure, applications, and in-flight application development projects. The SOA Program must begin with an understanding of the state of existing applications and projects to determine where existing functionality applies to the larger enterprise and can, consequently, be reused. This is critical when functionality is similar or identical among applications. It may also highlight infrastructure components that exist in the enterprise, or that need to be procured or developed to build a fully attributed SOA. Functionality that is wholly specific to the application in which it resides, or the project for which it is being developed, may be safely de-emphasized.

Initiatives must be begin early on in an SOA Program to create an Application Inventory and Project Catalog. The Application Inventory describes current applications. The Project Catalog describes in-flight projects. In both, the relevant features to capture include:

- current application functionality, services and dependencies
- granularity and capability of existing services
- interdependencies of current applications and between them and planned or in-progress projects, and related development and maintenance challenges
- current common service usage
- costs and other metrics relating to application development
- information accessed and delivered by applications
- data models, transformation and translations used in applications
- work flow and process flow involved in applications
- use of services such as single sign-on, logging, error and exception handling, monitoring and notifications
- SLA, QOS and related non-functional business information
- details of current delivery milestones and immediate project timeframes.

Conceivably, there could be more than one catalog or inventory created. For example, there might be catalogs and inventories associated with each line-of-business, division or other grouping. Early in the

SOA Program, essential services will become apparent from the role that their functionality plays in enterprise business process. The outcome of that process analysis provides the basis to understand the higher-level business services to be implemented in the SOA infrastructure. Once the Project Catalog and Application Inventory are created, the initial services for the SOA infrastructure can be selected.

Those decisions are also informed by outputs of the Business Process Optimization activities. Some application services will align well with strategic business services. Other parts of existing applications may provide particularly desirable infrastructure functionality, information and access services, and other infrastructure components. Other existing application services will need modification to apply at an enterprise level.

An initial audit of existing applications and projects will identify functionality present in existing systems that should be represented as services. Details of the services already offered will be published in a Business Service Catalog. For any business service, the Business Service Catalog will capture details of the interface of the service, the functionality provided by the service, and other metadata associated with the service that will be used to define its contract. The documented information should support the shared service development method adopted within the SOA Program and help identify the services to be harnessed in application projects. An SOA Program must carefully govern the structure and ownership of the Business Service Catalog, Service Roadmap, and other common models.

SOA PROGRAM PLANNING

Embarking on an SOA Program is a large undertaking. A mature SOA infrastructure will encompass many enterprise-wide applications. There will be publication, discovery, and reuse of services across business units throughout the enterprise, and services will eventually be extended to partners and customers.

It is important that adopters of SOA take an incremental implementation, rather than a “big bang” approach. BEA’s experience, as well as wider accepted best practice, recognizes that an incremental approach to software development is suited to breaking down the complexity of large undertakings and, in doing so, provides an ideal framework for containing risks associated with software development. As well as addressing the complexity of an SOA infrastructure, an incremental approach allows business benefits to be derived throughout the lifetime of the program, including the early phases. This is key as justification for an SOA is ultimately based on analysis of the costs and benefits it brings to the business.

There will be several underlying motivations in the identification, scheduling, and content determination of projects in the SOA Program. There will be a requirement to build shared services and building blocks. Additionally, work to evolve existing monolithic or point based applications into the SOA must be incorporated into the planning. Some of these motivations extend beyond those normally influencing non-SOA based development carried out within insulated LOB development, highlighting the need for governance.

The experiences of SOA adopters have allowed us to classify projects that produce shared services into groups based on two attributes: complexity and phase. Complexity refers to the complexity of the technical solution required for a project and the type of applications developed in that project. It is useful to recognize three levels of complexity: Simple, Medium, and Complex. Dimensions of complexity include factors such as organizational scope, tactical or strategic nature of the application, level of maturity of policy management present in the application, and sophistication of governance mechanisms applied to the application development. Phase refers to the timeline-based growth of the project. Typically, enterprises embarking on SOA go through three phases of growth. Each phase is characterized by the number of applications, and development, business, operational, and technical issues triggered by the increasing number of applications on SOA.

Figure 7 illustrates how phasing and complexity, in conjunction with the motivations underpinning incremental project delivery and the initial understanding of the starting IT environment, guides the selection and timing of projects in planning an SOA Program. This discipline ensures early business visibility of successes and the creation of essential technical infrastructure early in the SOA lifecycle followed by more complex application delivery as the SOA infrastructure matures.

As the SOA infrastructure grows incrementally with the successful delivery of projects, so too will the Business Service Catalog grow. The extent of reliance on shared services will also increase in new projects appearing over time.

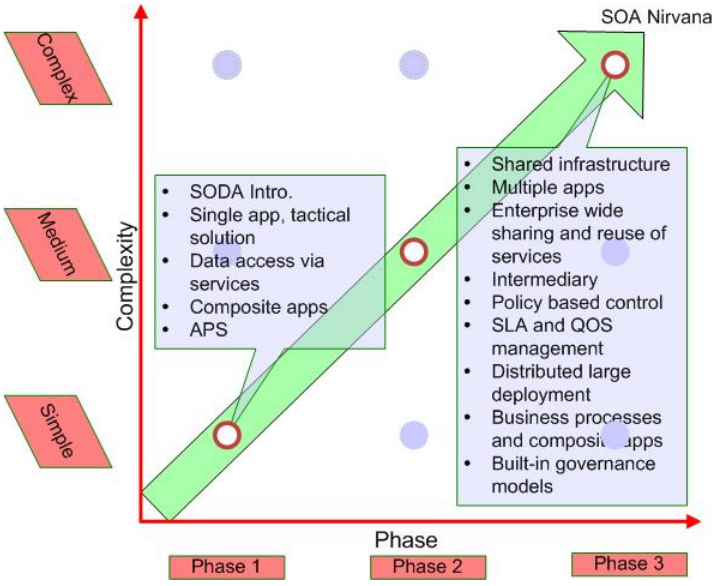

FIGURE 7 INCREMENTAL DEVELOPMENT APPROACH

Figure 8 depicts a way in which services might be harvested from different projects, how some services would be shared between projects, and how the overall target infrastructure is built out using the recommended incremental project approach.

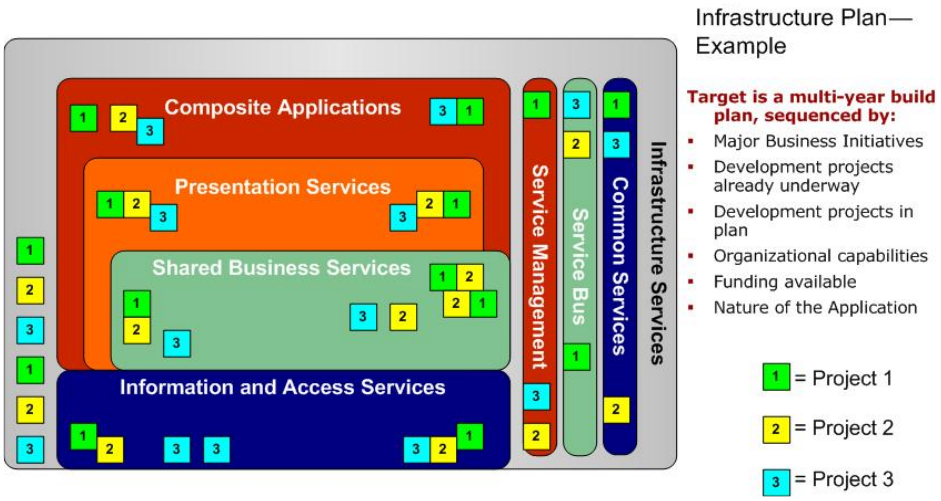

FIGURE 8: INCREMENTAL PROJECT HARVESTING APPROACH

Having identified the building blocks based on the applications, projects, and target architecture, the building blocks are cataloged into the reference architecture layers. Then the building block implementation is planned. The plan is driven by the applications and project release schedules, the target architecture roadmap, and the effort required for implementation. BEA recommends building the target enterprise architecture, and related building blocks in incremental projects. This allows the enterprise to reap the benefits of the investments in building blocks and applications immediately rather than wait for two to three years for a complete SOA infrastructure.

Planning for building block implementation depends on the project needs and the design, development, and test efforts required for each block. BEA's Project Estimation Guide can be used to estimate the effort required to develop each building block, and each application. The estimates for building blocks define phasing and ways to create early value from the investment the infrastructure that supports the architecture. This estimate, along with non-functional and business requirements can then be used to plan the building of the SOA based target architecture.

BUILDING BLOCKS

The reusable elements that are developed from the first application to the final application of a multi-year SOA program and the infrastructure in which to deploy, monitor, and manage them form the *building blocks* of SOA.

Building blocks can be grouped into two categories: software building blocks and organizational building blocks. Software building blocks include code, canonical data models, processes, services, applications, and components. Examples of organizational building blocks are best practices, standards, and tools for development, deployment, and operations, administration, and maintenance (OA&M). Applications are built using a collection of building blocks that make up the enterprise infrastructure. BEA recommends that building blocks be developed incrementally and refined iteratively as with the build-up of the target architectures. Services are the primary elements of the software building blocks. A service can be described as a way of packaging reusable software building blocks to provide functionality to users and to other services. When a service provides functionality to another service, we refer to the receiving service a *consumer* to distinguish it from a human *user*.

A service consists of three components: implementation, interface, and contract. Services represent functionality that is required by a set of users or consumers. That functionality is the service *implementation*. In order to access the implementation, users or consumers need an *interface* to it. The usage of the functionality embodied by a service is governed by a *contract*.

Service Implementation: The implementation part of a service is the actual code, application interface or other technology asset containing functionality that is needed in the enterprise. The implementation may be accomplished using any technology. Implementations, especially early ones, often represent functionality that already exists in an enterprise.

Service Interfaces: A Service interface provides a means for the users of a service to access its functionality according to the contract it offers. A given service may offer multiple interfaces to allow consumption of the service through different means. For example, an interface may be offered to provide synchronous access to some functionality for interactive usage while an asynchronous interface is provided for other purposes.

Service Contract: The service contract specifies the purpose, functionality, constraints, and usage of a service. The contract is defined by the business in business terms. For instance, a contract may constrain the functionality offered through a given interface according to the job function of the service consumer, or by whether or not the consumer is an internal or external consumer. The infrastructure required for SOA supports the definition and enforcement of contracts. Tools, such as Web Services management platforms and enterprise service buses, are available to provide this support.

CHARACTERISTICS

The main functional characteristics of a service are its calling paradigm, exchange paradigm, and complexity. The calling paradigm consists of two approaches: synchronous or asynchronous. The

exchange paradigm has two approaches as well: unidirectional, or bi-directional. Complexity of a service is related to its *granularity*.

Granularity refers to the level of abstraction of the service. A *fine-grained* service is one that offers discrete functionality, such as a standards-based method of calling an application API (an access service) or a way to operate on an enterprise data entity like an employee (an information service). Shared business services are also typically fine-grained, providing common business operations such as financial calculations. *Coarse-grained* services provide a mechanism to access complex business functionality, such as on-boarding an employee or processing an order for a product or service. Coarse-grained services are often long running, and involve the coordination of the execution of more fine-grained services. Accordingly, the implementation of these services is more complex.

Non-functional characteristics of a service include factors such as volume requirements, quality of service, and length of service execution. These factors comprise part of the contract for a given service.

The characteristics and functionality of services allow for their categorization into the layers of a service-oriented architecture. While the layers may be deployed in a physically distinct way, the categorization serves to inform decisions about utility and thus priority of development. The earliest building blocks will be infrastructure services; logging, auditing, error handling, and similar functions. These widely implemented IT staples often have shared implementations, at least in code form, and ensuring their common usage provides infrastructure needed by all of the services subsequently developed. Information and access services are also early targets since they have a high degree of utility for the enterprise.

IT disciplines that are required to successfully deliver SOA should also be considered building blocks. These include versioning strategy, service packaging and deployment, testing strategy, and similar disciplines. In some IT organizations, these disciplines will vary among business units, perhaps even among projects within a line of business. In others, some of the disciplines will be standardized across the enterprise, but the standards are seldom enforced. To the extent that they influence the delivery of services to consumers throughout an extended enterprise, strict adherence to standardized IT disciplines is a necessity. The scope of the extent of the creation and enforcement of such standards is a principal role of SOA governance.

COMMON INFRASTRUCTURE

In the course of delivering these early building blocks, infrastructure will be required to support them. Common infrastructure components such as a service registry provide a way for service consumers to find available services, and for service producers to advertise the availability of those services. The ability to find and use functionality that conforms to a needed contract at the time that it is needed is one of the most powerful features of SOA. When the first building blocks are deployed, a service registry can make the consumption of those services easier. As more services are deployed, a registry is required. There are several other infrastructure components that support the development and deployment of services and, accordingly, are in and of themselves building blocks of an SOA such as:

- security infrastructure including identity management and enterprise security frameworks
- dynamic routing, transformation and translation infrastructure, often in the form of an enterprise service bus
- configuration management for managing deployable components and for configuration of hardware and service deployment at runtime
- portal technology, for multi-channel delivery, web content federation and other presentation infrastructure.

Another capability that will become important as an SOA matures, is business activity monitoring to measure performance of the SOA against contracts.

The platform on which services are deployed, and the integration of the platform with the needed infrastructure, also bears consideration. There are a great many technologies and platforms that could be used to provide components of an SOA infrastructure. The basis of SOA in standards facilitates a best-of-breed approach to acquiring the needed infrastructure. However, there are compelling advantages to selecting a platform that incorporates most or all of the capability required for SOA in a manner that provides a single development and deployment environment. This greatly simplifies all aspects of operations and facilitates the standardization of IT disciplines.

These and other necessities for SOA infrastructure will be delivered as the services that require them are delivered. In order to minimize risk and cost, building blocks and infrastructure are best delivered in an incremental fashion.

ORGANIZATION AND GOVERNANCE

To date, attempts to transform the enterprise through the re-use of IT assets have met with limited success. Shortcomings in functionality definition, change management, design, software project management, and the solutions available to the IT organization are commonly cited as the reasons for poor results. However, it is pivotal to acknowledge that in most cases, no matter how much improvement could have been made in any one discipline, the absence of rigorous and sustainable organization and governance principles was the major, if not the deciding, factor in the poor outcome.

Any initiative that aims to transform the delivery of IT to the business can succeed only by adopting a genuinely strategic, enterprise-wide, approach to the problem by establishing a best practice organization and governance structure from the outset. SOA is no exception.

Governance and developing an organization to support governance is the foundation of the Domain Model. A strong SOA governance program will control all of the following areas.

- *Standards Compliance*: SOA will be successful only if a reference architecture and the services within it are built on standards defined and enforced on an enterprise-wide basis.
- *Service Roadmap*: Enterprise-wide control of the definition, development, and deployment of services, as they are produced on an initiative-by-initiative basis, is the only way to ensure that a complete and inter-operable portfolio of services is built up over time.
- *Enterprise Change Management*: Businesses go through constant evolution and change. Making a change to the business usually implies changes to core systems. Traditionally, this has a knock-on effect into many other systems, which can be greatly reduced by ensuring optimal use of the reference architecture and a rigorous change management process.
- *Re-use Enforcement*: IT tends to be more costly than it should because functionality is either duplicated many times over or existing functionality is not shared. Through standards and the roadmap, the governance function will be able to establish enterprise functionality tracking, and policies alongside it, to ensure re-use. In addition, because the enterprise will be fully appraised of available functionality, the enterprise change management process will become much easier.
- *Organization Structure*: SOA requires changes in the way that IT teams are organized in order to ensure adherence to disciplines like standards compliance, re-use, and the service roadmap. SOA Organization structures allow for a number of models (from fully centralized to fully federated) to optimally fit each individual enterprise.

- *Culture Change*: Governance efforts must promote a culture shift towards better working relationships between IT departments and lines of business by directing the sharing of information, requirements, and functionality, and providing incentives for closer working relationships. In addition, governance must drive the organization away from negative tendencies such as "re-inventing the wheel."
- *Skills Development and Best Practice*: The organization and governance function takes a holistic view of the technical and non-technical skills required by the enterprise. Attention is paid to the harvesting and sharing of best practice to ensure a more productive workforce, a more even spread of skills across the team, and better portability of skills from one project to another.
- *Funding Model and Accountability*: The reference architecture and services delivered institutionalize the ability to track and report on the cost and benefits of IT at fine levels of granularity. This ensures accountability and aids clearer investment decisions. This is essential in demonstrating value delivered and building the business case for future services, especially infrastructure services to be delivered at the enterprise level.

An SOA governance program is rolled out in phases. The following table highlights the phases of the program and the activities that are typically part of each phase.

Phase I. Definition	Phase II. Management	Phase III. Support / Control
▪ SOA Guiding Principals ▪ SOA Architecture ▪ Guidelines, Policy & Standards (Design, Development, Operational, Tools, etc.) ▪ SOA Processes & Procedures (Operational, Change Mgt, etc.) ▪ SOA Organization (i.e. Design, Skills, Roles & Responsibilities)	▪ Communication (Internal, External) ▪ SOA Standards and Service Compliance ▪ SOA Architecture Exception Process ▪ SOA Architecture Inspection and Adaptation Process ▪ Change Management Process	▪ Enterprise-Wide Mentoring ▪ Define and Manage Business Services Catalog ▪ Monitor Processes ▪ Funding

FIGURE 9: ORGANIZATION & GOVERNANCE PHASES

GOVERNANCE MODELS

Realizing a shared business services infrastructure relies heavily on using the right SOA governance model to meet both business and IT needs. Predefined, enterprise-wide, standardized, governance models are required. Issues that the governance model will address in IT terms include:

- How reusable shared services will be defined, and by whom
- How services will be built, by whom, and the software engineering approach
- Who uses the services, and in what ways
- How the associated service deployment and operations will run
- Who co-ordinates the four activities above, and the guiding principles to ensure success.

Issues that the governance model will address in business terms include:

- How time-to-market efficiencies and return on investment will be measured and at what level of granularity (composite application or service-by service) in order to maintain accountability for the SOA initiative and also perform ongoing cost-benefit analysis on the program
- How the groups of services orchestrated into business solutions (composite applications) will be managed across a product lifecycle, so that the integrity of the business solution is appropriately maintained throughout, irrespective of the lifecycle of the services that comprise it
- How longer-term domain engineering will be performed to continue to optimize the business processes represented in IT via the services repository.

Actual models will vary greatly from enterprise to enterprise, but adopters should anticipate the need for tiered governance models, since services will be ubiquitous and each will bring its own governance challenges. Tiered governance models will help enterprises address the volume and variety of these issues efficiently.

A typical tiered model, as show below, has three layers with each layer representing a different category of service. The governance model groups the services according to the reusability of the services, and assigns differing degrees of centralized control and direct business control to each layer.

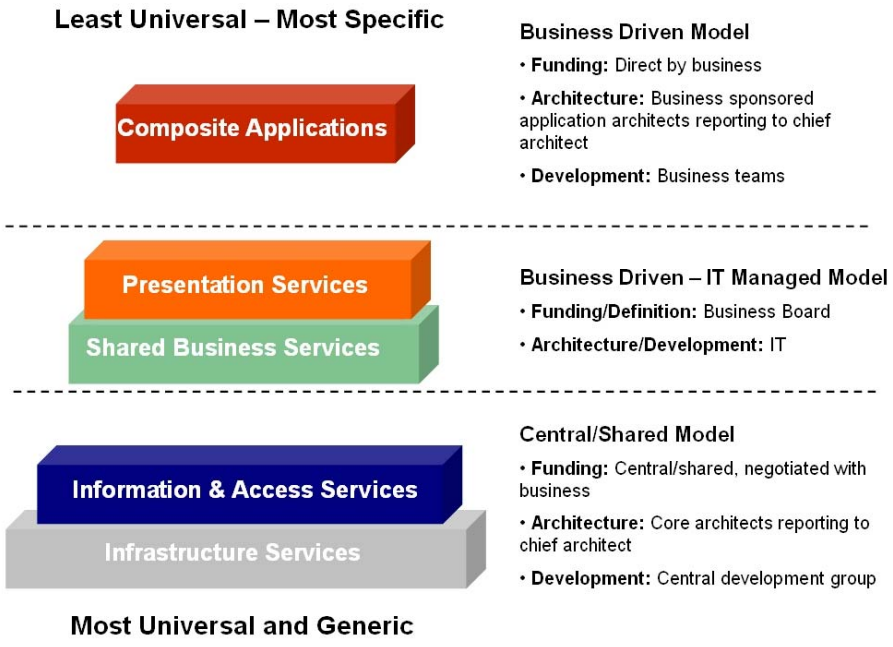

FIGURE 10: EXAMPLE OF TIERED GOVERNANCE MODEL

Governance is not a one-off effort. It requires constant effort to ensure that future development projects will adhere to the architectural principals of the SOA via an SOA compliance discipline.

SOA COMPLIANCE

Organizations must not rely solely on skilled personnel and effective communications to guarantee architectural conformance. SOA compliance is as much about cultural change as it is about

enforcement. In the early stages of an SOA initiative, shared services governance, and its purpose, will be different from that in a mature SOA because in the early days, a more interventionist approach will be required to move the organization towards institutionalized SOA compliance.

Organizations must ensure that architecture standards and best practices are followed, by defining a formal SOA compliance review process. A person from outside the delivery project who, not only understands the SOA architecture guidelines and principals but can also provide a strategic view on shared business services, should perform these reviews.

Failure to enforce SOA compliance leads to:

- SOA services which are poorly aligned to the reference architecture, or do not make appropriate use of standards
- Weak support of business goals because services no longer directly represent the business processes
- Dilution or complete loss of the “Services Inafrastructure Layer” causing the groundbreaking opportunities for accelerated benefits to be lost.

ORGANIZATION

A successful SOA Program requires changes in the way that IT is organized and operates which will have different impact depending on the type of enterprise adopting SOA. Many enterprises are organized into divisions. In many cases each of these divisions has its own IT department or its own dedicated IT resources. This organizational complexity will likely have increased the divergence of divisional IT application portfolios and the way in which the business works with IT.

For SOA benefits to be realized, a greater level of alignment is required between these different organizations. Over time this requires cultural change, skills evolution, re-definition of certain roles and responsibilities, and establishment of a common funding model for services. In addition, it is important that the principles of the SOA governance model are applied equally across the enterprise. This requires a balance of both local control and centralized co-ordination. Centralized co-ordination can be achieved through an enterprise-level SOA architecture group that crosses both organizational and functional divisions.

This SOA architecture group delivers centralized program management, architecture and planning, infrastructure services, and release management. Its exact structure depends upon the organization in which it will sit, and to an extent will be defined by:

- Business alignment and senior management backing
- Organization size and geographical structure
- The tiered SOA Governance Model
- SOA level of maturity within the organization.

The SOA architecture group also needs a structure that can grow as the SOA capabilities within the enterprise expand. A geographically dispersed enterprise that starts adopting an SOA would likely use one of the organizational structures depicted below throughout their SOA maturity cycle. (See *Figure 11: SOA Architecture Group Structures).* Initially, the Centralized structure would be implemented but as other needs arose, the enterprise could migrate to another structure.

The best practice long-term approach, and therefore the end goal of enterprises in implementing SOA, should be to arrive at a tiered governance and hierarchical organization model as this is the only truly scalable and sustainable way to define and deliver an SOA Program to the enterprise.

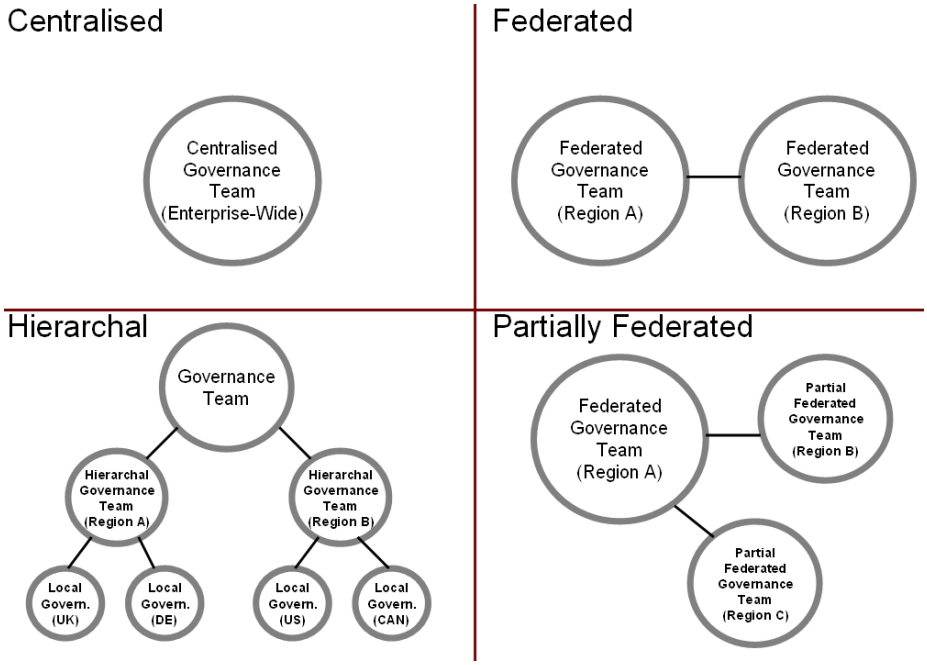

FIGURE 11: SOA ARCHITECTURE GROUP STRUCTURES

This group structures in the diagram are further explained in the following table:

Centralized	Hierarchical
Early in the adoption cycle of SOA a centralized structure is the preferred structure to manage the SOA Program in a coherent and consistent manner.	As an enterprise's SOA capabilities and needs mature, more scalable structures are required to continue managing the SOA Program. Some decisions will be made centrally and others will be delegated to more local groups within the hierarchy.
Federated	**Partially Federated**
When additional remote regions have a consistent need for SOA governance a Federated structure is applicable. These federated service organizations are exact replicas of the centralized structure in terms of SOA compliance and support but overall SOA Governance still remains with a centralized body. Due to the increase in communications and co-ordination lines there are limits to the scalability of this structure and it tends to not scale well above 3 regions. If more regions are required then the enterprise should consider moving to a hierarchical structure.	When additional remote regions have a consistent need for SOA compliance and support but have resource constraints a Partially Federated structure is applicable.

ADDITIONAL RESOURCES

For additional information on BEA's SOA Services and Solutions please visit the BEA SOA Resource Center at:http://bea.com/framework.jsp?CNT=index.htm&FP=/content/solutions/soa/

The SOA Resource Center will provide you with numerous SOA related, documents, presentations and reference material including the BEA SOA Self-Assessment Tool. The tool was developed based on the BEA SOA Domain Model and provides an ideal way for you to begin the process of assessing your maturity in each of the six critical SOA domains. At the completion of your assessment a 9 page customized report will be sent to you with some pragmatic suggestions for improving your SOA maturity and readiness.

ABOUT BEA

BEA Systems, Inc. (NASDAQ: BEAS) is the world's leading application infrastructure software company, providing the enterprise software foundation for more than 13,500 customers around the world, including the majority of the Fortune Global 500. BEA and its WebLogic® brand are among the most trusted names in business. Headquartered in San Jose, California, BEA has 81 offices in 34 countries and is on the Web at www.bea.com.

CONTRIBUTORS

John Allen, Senior Principal Business Consultant

John is software professional who has spent 18 years in the field working as an enterprise architect, team lead, and software engineer. During his 7 years at BEA, John's architecture expertise has helped many customers realize their vision for more robust, agile enterprises. For the last 18 months, he has been a key player in helping BEA to define and execute its SOA strategy.

Ian Barnes, Chief Advocate of Technology in EMEA

Ian's 19 years of industry experience have prepared him for a role in educating business leaders on the benefits of using technology and architectural approaches, specifically SOA. Acting as an advisor to BEA's customers, Ian works to define logical process flow solutions at a board-room level. In addition, Ian evangelises BEA's infrastructure technologies at a horizontal technology plane to clients' architects and development engineers. For large complex SOA implementations, Ian provides expert guidance to both BEA and client lead architects.

Stephen Bennett, Senior Principal Business Consultant

Steve has 20 years of industry experience covering software development, agile best practices, and architecture. He has provided architecture consulting and training to many of our enterprise customers. As part of BEA Worldwide Consulting, Steve is currently focusing on Service-Oriented Architecture. Before becoming a consultant, Steve was involved in enterprise-wide mission critical systems within the investment banking industry for over 12 years.

David Groves, Director Worldwide Consulting

David has 11 years of industry experience covering IT strategy, enterprise solutions consulting, program management, and delivery methodology. He has provided consulting and service delivery management to many of our enterprise customers in the Finance, Telco, and Government sectors. As part of BEA Worldwide Consulting, David is focused on developing BEA's service offerings and capabilities, with a particular interest in SOA and IT Strategy. Before joining IT, David worked as an analyst and strategist within the electricity industry.

Fabrice Lebegue, Practice Director, Americas SOA Practice

Fabrice has 11 years of experience in software engineering, technology, and IT consulting having served in delivery and management roles in software engineering, and consulting services. He has been involved in major integration, transformation, IT productivity improvement strategies and software engineering initiatives across a variety of industries. Fabrice is an internal subject matter expert in SOA, and reuse strategies. Prior to working for BEA, Fabrice Lebegue worked for Rational Software as a software engineering advisor to Rational Software customers.

Rag Ramanathan, Consulting Technical Manager

Rag has 13 years of industry experience as a software engineer in R&D and professional services consulting. Rag has helped many BEA customers successfully architect and implement enterprise applications. His experience includes software development, architecture, and integration of enterprise solutions. As an Enterprise Architect for BEA Worldwide Consulting, Rag is focused on developing SOA best practices for BEA partners and customers, and delivering assessments for SOA readiness.

Dale Slaughenhaupt, VP of IT & Deputy CIO

Dale oversees the development of all eBusiness solutions within BEA's IT department including enterprise architecture, enterprise infrastructure services development, configuration and release management, and the MyBEA program office. Dale has over 11 years of experience in technical program management and application/software development. Dale has helped numerous BEA customers by sharing the best practices that he and his team have picked up in developing BEA's internal SOA over the last 4 years.

Special thanks to Phil Aston, Jeff Block, Daniel Healy, David Miller, Yogish Pai and Nathan Romney for their assistance on this project.