

# The SOA Magazine

## Feature Article



### Principles for Implementing a Service-Oriented Enterprise Architecture

by Tyson Brooks

Published: May 30, 2009 (SOA Magazine Issue XXIX: May/June 2009)

[Download PDF](#)

[Digg This](#) • [De.licio.us](#) • [Slashdot](#) • [Technorati](#) • [StumbleUpon](#) • [Google Bookmark](#)

*Abstract: This article provides four key principles and actions an organization can undertake for implementing a Service-Oriented Enterprise Architecture (SOEA). SOEA adopts the concepts used in enterprise architecture and service-oriented systems design and adapts them to the enterprise level using service-oriented computing (SOC). A SOEA should provide the guidance to enterprise transformation of the organizational, business process management/modeling and reengineering, enterprise resources and information technology (IT) systems and application development in support of the organization. Although technical in nature, the SOEA form of architecture does not focus solely on network (i.e. infrastructure, core services), information (i.e. data, enabling services), and resources (i.e. providers and consumers of services and data) architectural views.*

#### Introduction

One of the most intriguing new information systems methodological approaches in IT to come along is SOA. The emergence of the SOC paradigm and Web service technology, in particular, has aroused enormous interest in SOA [REF-1]. As organizations grow and become more complex, consumers make greater demands of their business functions and models. Organizations have continually enhanced and expanded their IT infrastructures more and more as information has become available in electronic form. SOA activities focus on hypothesizing new ways of working by developing practical implementation scenarios that build on and exploit existing information systems and networking technology. Organizations have recognized that collecting and disseminating data represents the gathering of very valuable "services" and the potentially tremendous opportunities for the organization to collect and reuse this data.

The capacity to recycle these services across the organization requires the integration of technologies including internets/intranets/extranets, e-mail, data warehousing, data mining and workflow/document management systems. Typically, an organization has data and information residing in various tacit as well as explicit sources including documents, databases, human intellect, and day-to-day practical experiences. The primary challenge of a SOEA is to create a service-oriented environment that: (1) provides the reuse of services through SOA capabilities, (2) captures the full breadth of data/information available, (3) offers automated services techniques, and (4) supports advanced information analysis.

With these capabilities, an organization can "create services" by identifying new relationships and trends that affect their business. This form of architecture can then be formatted and indexed into the enterprise architecture to be shared and disseminated across the enterprise. This creates a SOEA which is the resulting capability to foster innovation by applying this knowledge to business requirements such as strategic planning, process improvement, technology reuse and tactical problem solving.

#### The Correlation Amongst SOA and Enterprise Architecture

SOA is a way of reorganizing a portfolio of previously siloed software applications and support infrastructure into an interconnected set of services, each accessible through standard interfaces and messaging protocols [REF-2]. It is an approach to capture, define, and expose a collection of services and service components available throughout an organization. SOA is an architectural discipline that centers on the notion that IT assets are described and exposed as services. In simplest terms, a service is a unit of work done by a service provider to achieve desired results for a service consumer as displayed in Figure 1. Standards such as Simple Object Access Protocol (SOAP), Web Service Description Language (WSDL), Universal Description, Discovery and Integration (UDDI), Extensible Markup Language (XML) and Web Service Security (WS-Security) allow organizations to exchange data in a vendor neutral manner using well defined and widely adopted standards. Service components refer to individual processes, data, applications, and technology that an organization can bundle to deliver a particular service. SOA strives to achieve economies of scale in the development and implementation of business solutions via the reuse of services and service components.

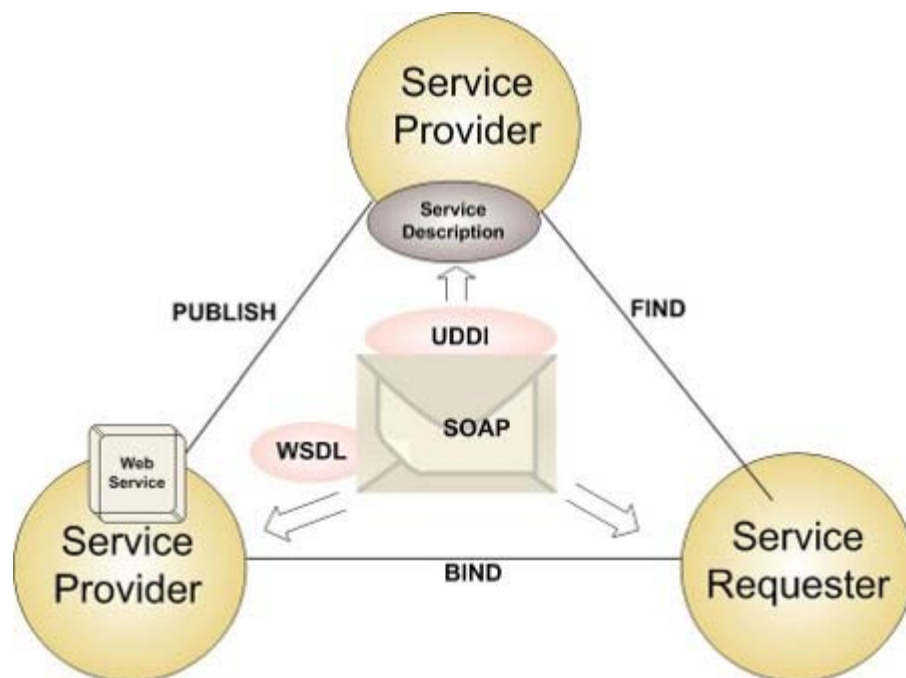


Figure 1: Service-Oriented Architecture

Enterprise Architecture, as displayed in Figure 2, defines how all parts of an enterprise work together to ensure business and technology alignment, realize operating efficiency and effectiveness, identify cost savings / cost avoidance, and provides adaptability for more responsiveness to changing needs of the mission. In short, enterprise architecture provides insight, enables communication among stakeholders and guides complicated change processes [REF-3]. Enterprise Architecture covers the current and desired relationships among operations and management processes and information technology. It describes the baseline or "current architecture" and future or "target architecture", and provides a strategy that will enable the agency to transition from its current state to its target environment.

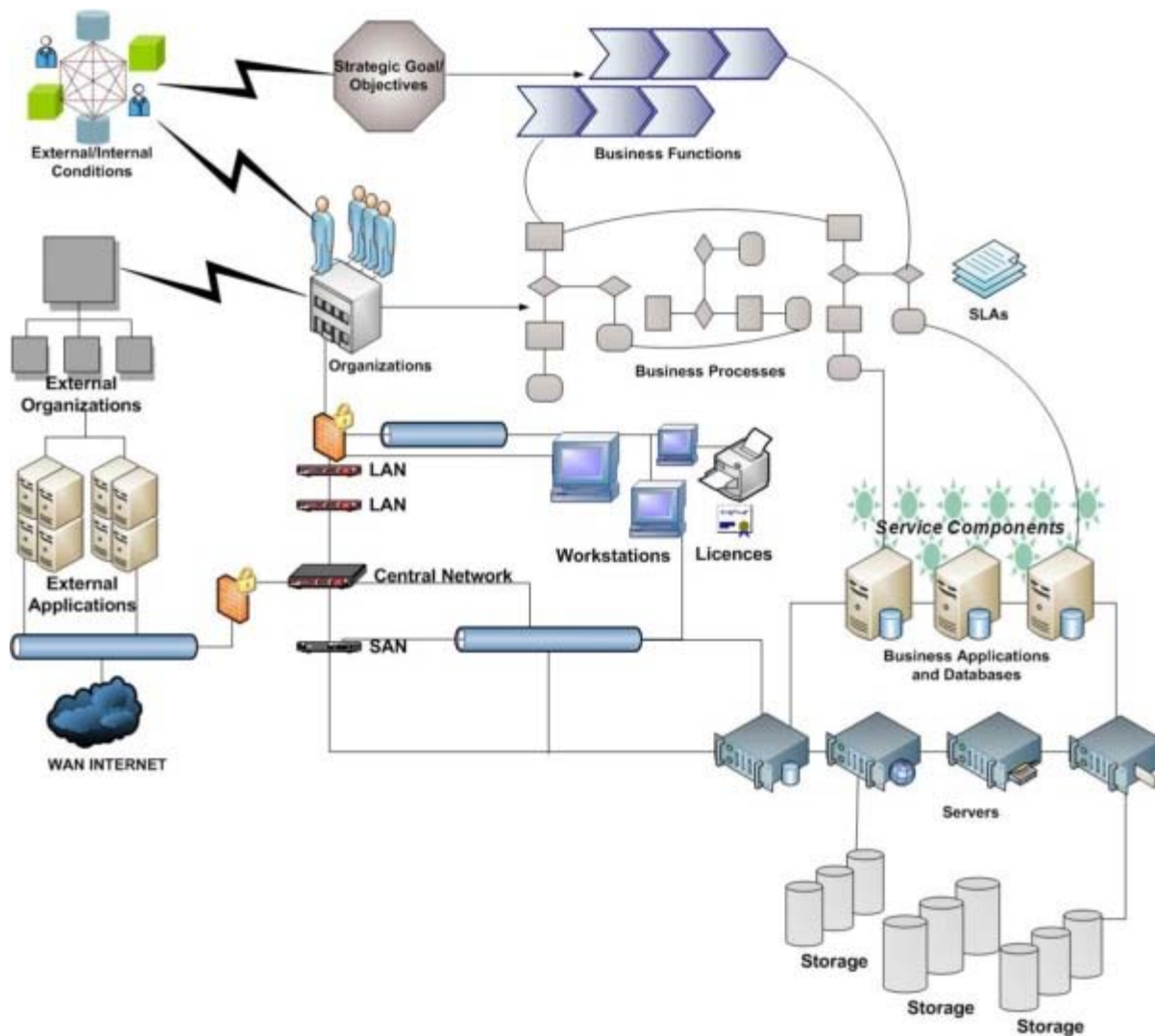


Figure 2: Enterprise Architecture

As a tool to align an organization's business strategy with the processes, data, applications, and technology that support the strategy, enterprise architecture can, and should, apply SOA to capture and expose the services and service components that are available. As such, Chief Information Officers (CIOs) and Chief Architects should ensure that their specific enterprise architecture programs develop high-level strategies which incorporate SOA activities. Enterprise Architecture must support an approach to developing a SOA which identifies, classifies, and manages its services and service components consistent with the direction of the organization's mission. Enterprise Architecture deals with high-level logical concepts of SOA implementation, not with the physical implementation of SOA at the IT level.

Why use SOA? It's simple; SOA reduces the complexity of software and forces organizations to be agile. However, since SOA's are operationally deployed in areas of the IT infrastructure, these implementations can create unexpected complexity. Where traditional IT management processes and tools are not always up to the task of monitoring and managing service-oriented applications, SOA deployments require as much support and investment in infrastructure management as they do in developer kits and testing tools. It is important to approach SOA as part of a broader infrastructure architecture and management program rather than as a standalone application. There is a high correlation between a business' level of satisfaction with SOA and their commitment to managing IT as a set of services.

The transition to a SOEA will not be simple or immediate. Since enterprise architecture provides a strategy to enable an organization to transition from current state to target architecture, SOEA must adopt SOA concepts to the enterprise level to enable changes in both IT development and business process planning and management. Phasing in these changes may require several years. Although SOA applies to both manual business processes (or some automated) as well as IT systems development; its roots are in IT. For years, programmers have reused bits of computer code to save coding time and improve software reliability

and performance. From simple subroutines, to code libraries, to object oriented programming, to Web services, and now to SOA, the concept has been to transform programming from a manual craft to something approaching manufacturing. By breaking complex systems up into component parts, programmers could assemble large applications more quickly and update them more easily. Like a mechanic replacing the engine of an automobile, SOA makes it possible to replace individual components with new or improved versions without having to touch the rest of the system.

Extending the concept of SOEA beyond IT is a large and important step. Traditional organizations used IT to speed up manual tasks. Today's organizations combine human skills and electronic data processing as integrated units—"services"—that did not exist before and do things that neither could do alone. Enterprise architecture is the appropriate tool for implementing SOA across an enterprise, but enterprise architecture is not SOA. Enterprise Architecture is a tool for analyzing organizations and managing change. SOA is a particular business architecture strategy that enterprise architecture can implement. Since an organization's business architecture is widely regarded as the art and science of delivering coherent, dynamic, and complete business designs [REF-4], a sound SOEA should guide both the enterprise architecture and SOA enterprise transformation to support the enterprise.

IT developers tend to see SOA as a purely IT-oriented practice built around specific methodologies like .NET or J2EE. Business managers tend to see SOA, if at all, through the lens of the Capital Planning and Investment Control (CPIC) process, which requires their investment proposals to map to the organizations transition strategy. At best, the two perspectives come together from opposite directions—managers seeing from the top down, developers from the bottom up. At worst, they are inconsistent, with the two communities talking past each other, using the same words with different meanings.

Since it is a recognized IT best practice, SOA has taken root in the IT development community on its own. SOEA supports this trend in the IT arena, and seeks to expand SOA into the business community. Reaping the benefits of SOEA in the non-IT arena will be an ambitious and lengthy task. It will require the recasting of existing business processes as services subject to defined "contracts" between suppliers and users, and governed by service level agreements that define performance quantitatively.

SOEA concepts, however, apply in both arenas. Figure 3 highlights concepts that typify the coming SOEA transition.

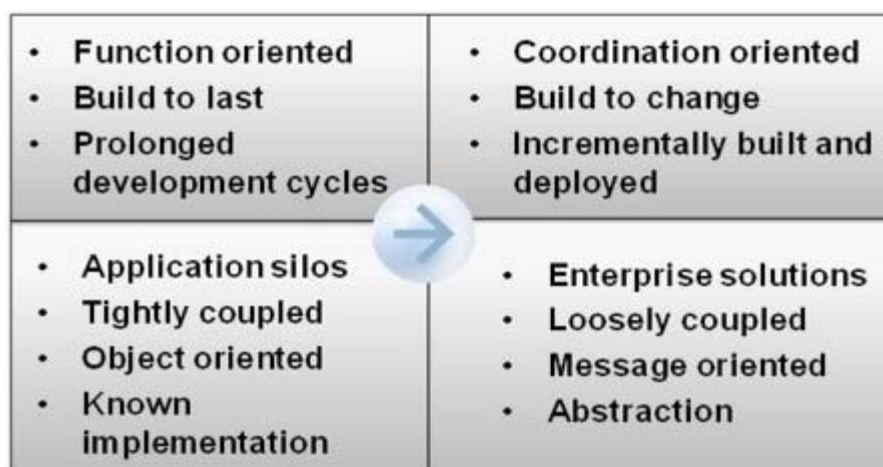


Figure 3: Transitioning to SOEA

Instead of single-use business functions, SOA creates reusable service components. Where traditional business builds monolithic functions, SOA breaks large-scale functions down into components that work as plug-and-play services. It builds for change by adding, improving, and swapping out services independently. It deploys incrementally rather than all-at-once, thereby lowering risks and smoothing the bumps in change management. SOA builds services in loosely coupled, networked clusters rather than tightly clustered hierarchical silos.

This approach clearly applies more self-evidently to software development than to the management of people and organizations. There is clear opportunity to apply SOEA principles to some manual business processes, and especially to hybrid manual/IT processes, but the management implications of this change will be significant and change must be made incrementally.

The remainder of this article develops the structure of integrating an SOEA by:

- Describing four key strategic principles through which an organization can “operationalize” SOEA, and;
- Describing the steps through which an organization can plan to implement SOEA throughout the organization.

These steps support the continued development of SOEA at the IT level and the move toward a broad business implementation of SOEA over the longer term.

### Key SOEA Strategic Principles

Organizations should develop enterprise architectures to support organizational analysis and identification of performance gaps, discover opportunities for collaboration, and avoid duplicative investments within and across the organization. Organizations should also classify their business and IT operations in an integrated and consistent way.

As shown in Table 1, these SOEA principles serve as a bridge between the business and IT “layers” of the enterprise architecture. Organizational services are the foundation to support the reuse of applications, application capabilities, components, and business services. More importantly, well defined services help define SOEA concepts clearly, which help reconcile differing management and IT perspectives considerably, and is thus a good starting point for introducing SOEA within an organization.

Here are the four principles that can provide assistance and guidance in helping to reconcile differing SOEA viewpoints:

Principle	Description
<i>Services can include both business processes and IT services, and can exist at all levels of the architecture.</i>	Services should include both business processes and IT services, and can exist at all levels of the architecture. This is a deliberate expansion of the conventional understanding of SOA in the IT world and reflects the broader SOA purpose.
<i>Service Components are characterized by how well they meet the definition of a service component, and by their level of complexity.</i>	Service components can be, and often are, hierarchical or networked, aggregating lower level components into complex, high-level services. They are also characterized by how well they meet your organizations definition of a service component, and by their level of complexity.
<i>Services Components must be traceable throughout the organization’s Enterprise Architecture.</i>	All services must map to business processes and applications within the Enterprise Architecture.
<i>Services, service components, and service level agreements (SLAs) should be defined, cataloged, and published.</i>	Services, service components, and SLAs should be defined, cataloged, and published to make their capabilities known to potential customers. It follows that higher-level services and service components should be configured, managed, and governed to provide a stable and reliable service experience for customers.
Table 1. SOEA Principles	

*“Services can include both business processes and IT services, and can exist at all levels of the*

architecture.”

The first principle on which to build a SOEA is that services can include business processes, IT elements, or both. It suggests that services can exist at all levels of the architecture.

A service component is defined as a self-contained business process or service with predetermined functionality that may be exposed through a business or technology interface. This definition makes clear that services can be provided by a combination of business and technology elements, not necessarily by IT alone. An example of a business process that works as a service would be a cashier at a fast food restaurant. The cashier offers well-defined services to all customers and is “reusable” in the sense that it is replicated throughout all the fast food chain’s locations. IT elements could include a timesheet application within a financial management system as an example of a service provided through IT.

IT-based services may often be implemented as Web services, but SOA is not exclusively about Web services; Web services are a specific subset of IT services. Combined business/IT services may use some form of data exchange (i.e. Enterprise Service Bus) as an example of a combined business/IT service. Data that arrives on paper or on computer media is processed partly by hand; data sent electronically is fully automated. These are simply two pathways of a general service to customers using a comprehensive data exchange. These different types of services map to different levels of the enterprise architecture layers (i.e. Strategy, Business, Data, Application, Technology and Security architecture) as displayed in Figure 4.

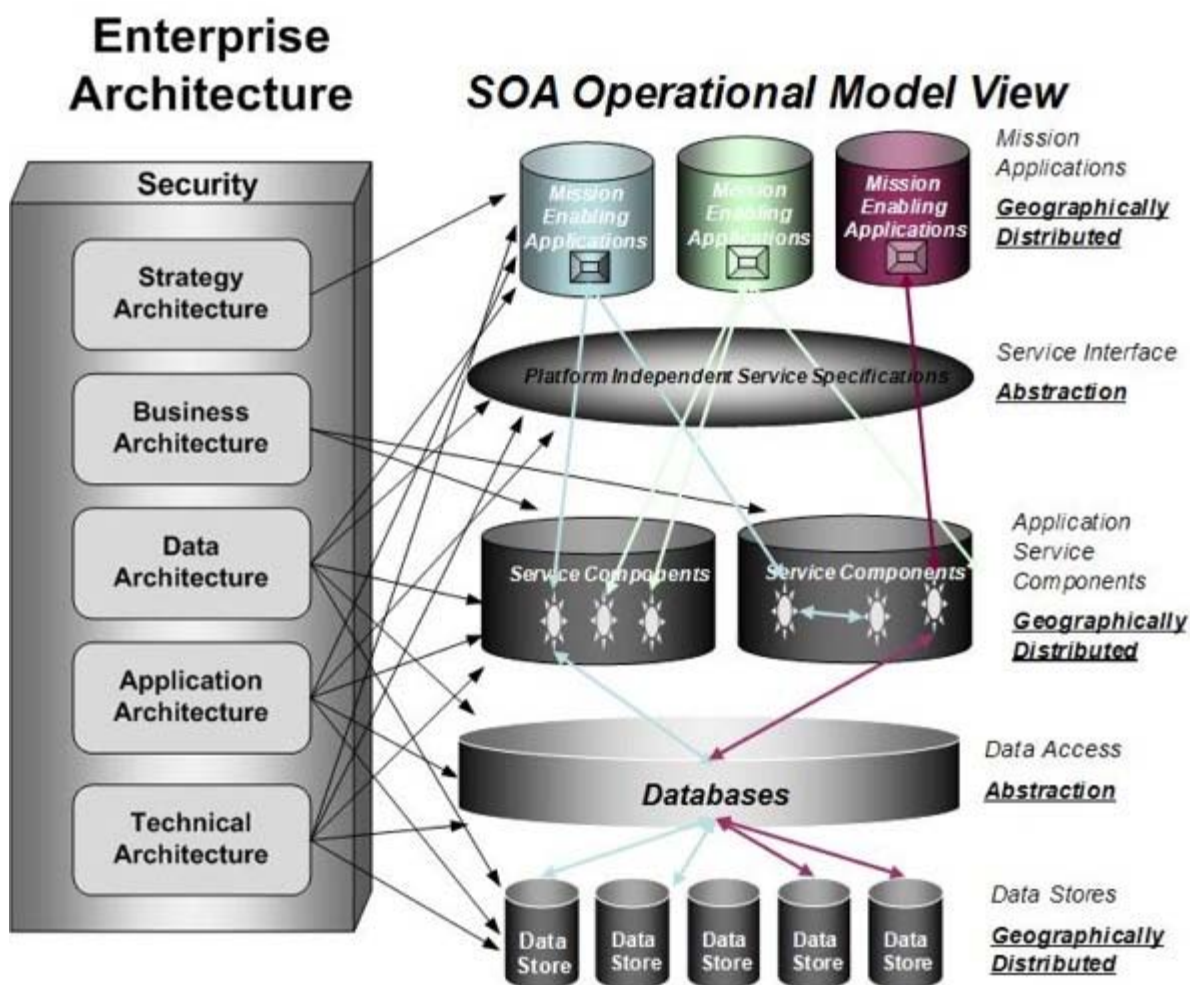


Figure 4: Enterprise Architecture to SOA

The primary purpose of the enterprise architecture layers displayed in Figure 4 are as follows:

- The strategic layer is to describe the goal structure of the organization. The goal structure consists of the mission/vision, supported by organizational strategic goals, supported by strategic objectives and initiatives.
- The business layer provides the context and description of the business functions, business processes,

business model, and initiatives that compose the organization's business domain.

- The data layer provides data integration and data management and is structured to represent the data resources of the organization and to show critical relationships between these vital resources and the applications and business processes that use them.
- The application layer facilitates improved data management and application interoperability which is achieved in part by providing the key information about applications, the nature of their interfaces with other applications, the mappings to the business processes they support, and the degree and quality of support provided.
- The technology layer facilitates improved application and network interoperability, reliability, processing capacity, and adherence to standards and enables organizations to identify the capabilities and capacities of its hardware and software base.
- The security layer represents the baseline and target states of the security architecture in order for the organization to trace and demonstrate compliance with applicable security drivers and standards.

Through the enterprise architecture, business process services should align to the business layer of the architecture, IT services align to the data level (data warehouses and marts, for instance, can provide services), the applications level (through applications and their modules), and the technology level (i.e. commercial off-the shelf [COTS] and government off-the-shelf [GOTS] software products can provide services out of the box; specialized services also provide defined services). These alignments emphasize why the services of SOA are likely to remain predominantly IT-focused in the short-term, but viewing business processes as service components, is a larger umbrella for future work. This larger view should pay off in more flexible and efficient implementation of mission-critical functions, based largely on the benefits of the second SOEA principle.

*"Service components are characterized by how well they meet the definition of a service component, and by their level of complexity."*

The second SOEA principle is that organizations should build complex systems, to the extent possible, from reusable service components. SOA encourages managers to break down established services into simpler components and determine if the components can be repurposed to the task at hand and enterprise architecture identifies all applicable services, systems, applications and technologies used within the organization.

As organizations moves towards SOA, it will design and develop IT or non-IT services as components that comprise one or more services. Components are defined by (1) whether or not they meet the definition of a service component (quoted in section 1 above), and (2) by their level of complexity or granularity—how they relate to each other and network with each other to create complex functionalities.

This second SOEA principle, in effect, distinguishes a "service" from a "component." It will allow organizations to track its progress in converting non-SOA systems to SOA systems. An old-style, tightly coupled application module may provide a service, but because it is not reusable in other systems—not "exposed" through an interface so that other users can get at it and use it—it does not meet the definition of a component. When it does become reusable and accessible, it becomes a service component.

Components can be, and usually are, networked. Lower-level services often aggregate into more complex, high-level services, with components calling each other at multiple levels of aggregation. The following defines three levels of granularity in the component hierarchy:

- *Distributed Component (DC)* - A DC is the lowest level of component granularity. It is a software element that can be called at run-time with a clear interface and a clear separation between interface and implementation. A DC is autonomously deployable and often a COTS or GOTS package. DCs may include data assets, application modules (often COTS or GOTS modules) that provide a component function of a larger IT Business Component (BC) or hardware or software platforms. An accurate baseline inventory of DCs will support redundancy and gap analysis and help identify situations in which DCs can be more widely exposed for reuse.
- *Business Component (BC)* - A BC represents the implementation of an independent business concept, business service, or business process. It consists of all the technology elements (software, hardware, and data) necessary to express, implement, and deploy a business concept as an autonomous,

reusable element of a large information system. A BC is a unifying concept across the development lifecycle and distribution tiers. BCs may include multi-function business processes, or a single function business process that operates with the support of DCs, multi-function data assets such as warehouses or multi-function applications, including multi-module COTS or GOTS packages. BCs, like DCs, are relatively low-level acquisitions. As for DCs, an accurate baseline inventory of BCs supports redundancy analysis; helps identify situations in which BCs can be more widely exposed for reuse.

- **Business Component System (BCS)** - A BCS is a set of cooperating business components assembled to deliver a solution to a business problem. A BCS includes more than one BC, or includes a manual business process in addition to an IT-based component. BCSs may include IT systems that are hierarchies or networks of IT BCs or business processes supported by one or more IT BCs.

Figure 5 below illustrates the relationships between architectural elements and their level of granularity.

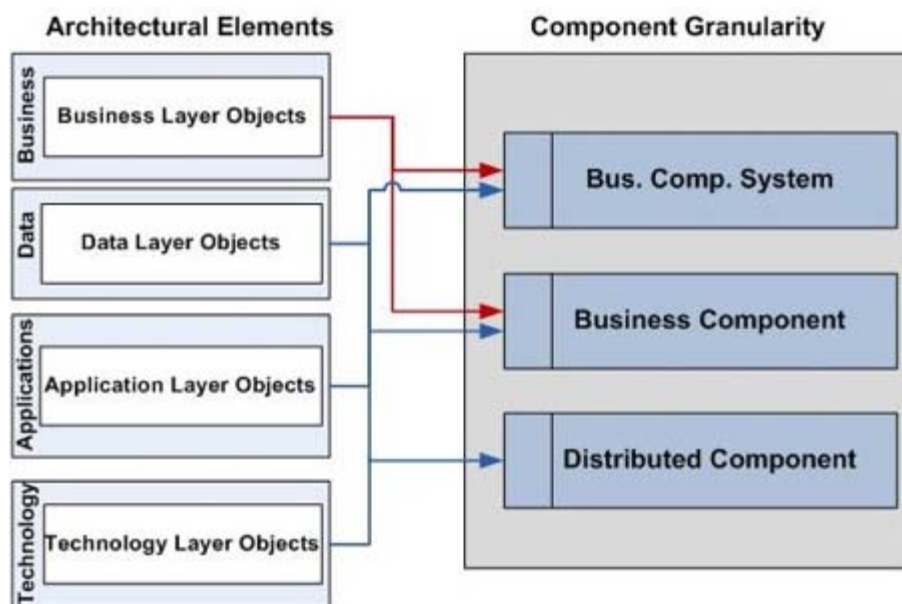


Figure 5: Architecture Elements and Levels of Granularity

*"Services components must be traceable throughout the organization's enterprise architecture."*

The third SOEA principle is that all services must be traceable throughout the organization's Enterprise Architecture. This permits the community of users to search for and find services that they require to conduct their business.

Figure 6 below how the component granularity of an organization's architecture associates with the application's infrastructure. Note that a single service component maps directly to the service components provided by the organization. For high-level services, this may mean mapping to higher levels of the service for mapping to all or most of the services beneath it since many components are involved in transferring data through some form of data exchange. At the other extreme, it may be necessary to decompose existing service categories to better describe narrow, specific service components whose functions are not clearly covered in the service registry.

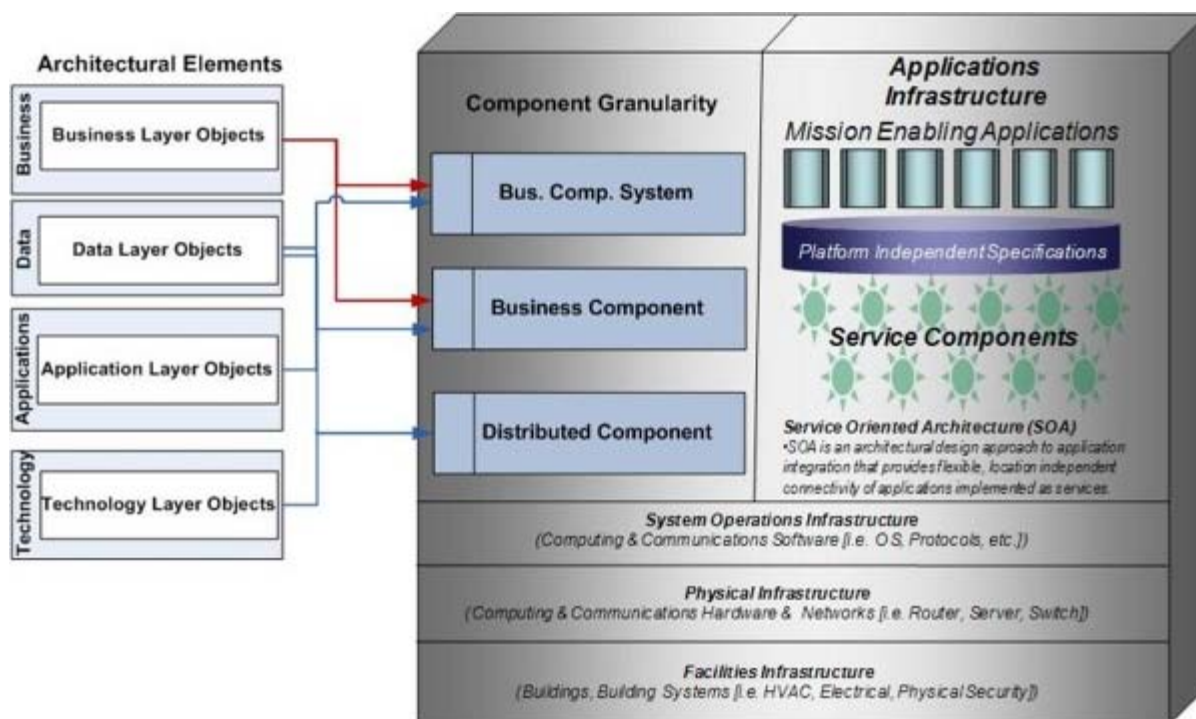


Figure 6: Architecture Elements and Types of Services

The spirit of SOA is that all service components, no matter where they originate, should be available to any authorized user. Enterprise Architecture also suggests that organizations accurately represent the alignment, dependencies, and relationships between enterprise architecture artifacts, the enterprise, and the external environment. By creating this alignment and creating a SOEA, this will help ensure that individuals at all levels of the organization have the right information to help transform the organization to a more efficient and effective agency and support its overall mission, goals, and objectives.

*“Services, service components, and service level agreements (SLAs) should be defined, cataloged, and published.”*

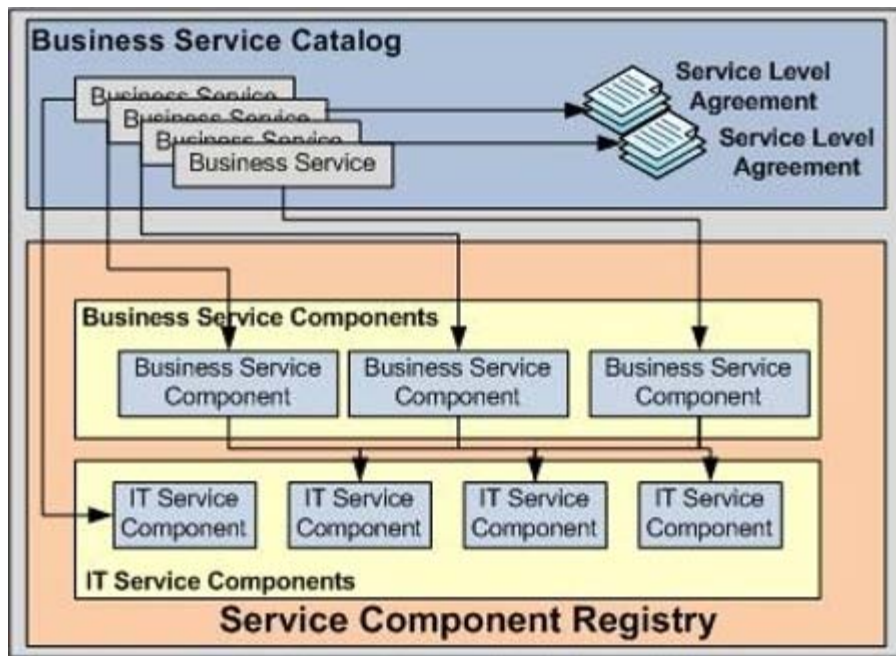
The fourth and last principle of a SOEA strategy is to identify, catalog, and publish all services used within the organization.

Since the essence of SOA is reuse and repurpose of service components, organizations must communicate what each service is, where to find it, and how to use it. This is especially important for higher-level services, where users and providers may enter into formal Service Level Agreements (SLA); however, not all services will require such agreements. Implementation of an organization’s SOEA will combine the following:

- SLAs that define the relationship between the provider and the consumer of a service, especially for high-level services.
- A business service catalog used to publish and discover services. This catalog will contain business information relating to the service, but will drill down to a service component registry.
- A service component catalog consisting of many service components representing business processes and technology service components that provide varying levels and types of services.
- The service components that provide these services.

Figure 7 illustrates this concept. The business service catalog provides “one-stop-shopping” for managers and business planners to publish or discover business services. The catalog provides the internal operating list of organizational-specific functions; it is the de facto “Service Component Registry” which is the catalog of well-described and accessible service components.

Figure 7: Business Service Catalog and Service Component Catalog



Maintaining these catalogs will require active management, especially for high-level services. To provide a stable and reliable service experience for customers, services and their components must be configured, managed, and governed appropriately. The Service Component Catalog cannot exist without a structured governance or change management process.

### Actions To Implement SOEA

SOEA provides a conceptual approach that interprets the terminology and structure of services as a framework for SOA implementation within an enterprise architecture. This lays the basic framework for implementing SOEA as a global management approach for an organization's operations.

Some organizations may be well along in implementing SOEA at the IT level. More organizations have developed some methodology for documenting SOA within their enterprise architecture repositories. This SOEA methodology not only provides four principles outlined above but should include the following actions to provide the basis for developing the SOEA. Organizational systems already in place for managing SOA at the IT levels can serve as the nucleus of longer-term SOEA at the business and architectural levels.

### *Developing a SOEA Governance Structure*

IT governance appears to develop best as a bottom-up response to need, rather than as a top-down imposition of abstract requirements. Since SOA is already taking root at the implementation level of IT, the governance of SOA may be able to grow effectively out of existing workgroups and management processes. Project and portfolio managers must communicate to senior managers and executives to develop an integrated list of the various services offered or planned. This list is the beginning of a Business Services Catalog. They must also develop the technical details needed for the Service Component Registry. Part of the intent of the Project/Portfolio Managers' effort is to begin the process of sharing components. The effort is also seeking to align services and service components; looking for redundancies and opportunities for sharing expenses, such as the procurement of licenses, and sharing of components between systems. These working-level efforts represent a practical start towards building SOA government procedures.

The SOEA governance structure should be the nucleus of the SOA management. Executives and managers need to define the service functionality and supporting business rules of the SOA. These rules, in turn, require the alignment of business service components within the enterprise architecture. Executives and managers can develop the initial specifications of the Business Services Catalog as an extension of this coordination. They would define the content and attributes necessary to identify and select business services, as well as a structure (or multiple structures) for specifying SLAs. Also, by using a SOA modeling methodology to develop interfaces between other major systems, this could assist in tracking the cost for IT investments. Convening a group of representatives from development contracts on one side and CPIC staff

on the other, to create initial specifications for the Service Component Catalog and its instantiation in one or more existing or new tools could deem beneficial.

Once these efforts have reached an initial level of confidence, executives and management should examine what, if any, higher-level governance might be needed at the cross-enterprise level, such as through a SOA Quality Council or a SOA subcommittee.

### *Enterprise Architecture Modeling*

Organizations must establish a SOEA approach that implements their enterprise architecture modeling activities. Multiple types of existing objects at all levels of the architecture can now serve as building blocks for SOA components. Business processes, databases and marts, applications and application modules, and various technology objects can be defined as services and recognized as components if they meet the definition.

Representation of the organization's service components are expanded to include the concept of service component granularity. Since, multiple objects can be networked together to produce higher-level service components (BCs, BCSs, and DCs), "candidate" service components, those services that do not yet meet all the requirements of SOA, can be represented by mapping the candidate to the service performed, but not to a component level. All services link to the service definitions, with provisions made to decompose the Service Component level into detailed service components as necessary. Objects can link to multiple service components within the organization.

The SOEA must also define the objects available in each layer of the architecture (as display in Figure 4), the relationships between them, and the properties of these objects that are of value in supporting the use of the enterprise architecture as a decision-making tool. While the enterprise architecture repository defines "what" is in the architecture, the SOEA should provide the guidance on "how" the services support the business and align throughout an organization's architecture.

### *Create a Generalized Methodology for the Implementation of SOA at the Business Level*

Applying SOA principles to existing projects in the pipeline offers many benefits (such as cost savings in development and licensing), but it does not capture the higher ideas of SOA-enabling manual business areas or re-engineering of existing business process systems on a large scale. It risks the creation of incomplete service components and may inadvertently preserve redundant or obsolete business processes. Organizations will need to broaden its component aggregation of its business processes by combining the practical benefits of this bottom-up approach with a more generalized, top-down methodology.

A particularly top-down aggressive approach would be to conduct Business Process Re-engineering (BPR) across the entire organization to identify a series of core essential activities that need support from service components. This would produce the required high-level integration of services, but it would be prolonged and expensive.

A middle ground is the application of Value Chain Analysis [REF-5]. This approach focuses on required business results. Instead of mapping individual activities and processes, it defines "value chains"—sequences of business process threads that yield intermediate identifiable customer benefits or results that expand over time to a desired end result. As more value chains are identified, the objective is to "grow" service components by integrating and consolidating comparable functions across value chains as more and more value chains in the enterprise make use of them.

Although Value Chain Analysis works well in the commercial sector, its applicability in the government sector will need to be examined differently. The commercial sector often breaks down into divisions that share many like processes. Government agencies, on the other hand, often have numerous; heterogeneous processes within their various programs whose value chains look very different from one another. Nevertheless, many similar value chains clearly do exist within organizations such as regulatory creation and management, permitting, licensing, enforcement, public information and outreach, records management, and applied research are examples.

### *Protect Service Components through Defined Security Policies and Procedures*

Service component protection must begin with the creation of data and information, with particular focus on

defining and documenting protection levels and access control decisions for the services in use. Protection must be assured throughout the life cycle of the data: creation, modification, storage, transport, and destruction. Organizations can no longer rely simply on transport mechanisms/link encryption to provide end-to-end protection. Since organizations are part of the myriad of interconnected networks, services routinely flows in and out of a network through numerous access points. This separation of information from systems requires that the information must receive adequate protection, regardless of physical or logical location.

A critical factor in ensuring adequate protection for all SOA and enterprise architecture services is the responsive updating and application of policies and guidance to address the latest changes in technologies while defending against the latest new and developing threats. Equally important is the necessity to ensure that the policies and guidance provide sufficient flexibility to allow their adaptation to the diverse missions across the organization. In addition, achieving the goal of trusted data everywhere within the enterprise requires partnerships and combined efforts with other components of the security community (i.e., Intelligence, Counterintelligence, Operations, Physical/Personnel Security, and Critical Infrastructure Protection) to provide an integrated systems security posture.

## Conclusion

In conclusion, organizations are developing integrated approaches to SOEA implementation and need to establish the practice in many areas of the IT development community. It is clear that the full implications of SOEA are unknown to the IT and management communities. Communication, outreach, training, and preparation for SOA in the IT and management community is therefore a high priority.

Implementation of this SOEA is likely to be, and probably should be, incremental. More progress needs to be made at the development level. Organizations need to develop the implementation, governance and configuration management aspects of an SOEA methodology. Their efforts must address governance, security considerations, configuration management, and submission of data and data components. This SOEA implementation approach must also identify the actions and elements needed to implement the associated infrastructure to manage these efforts and make SOEA a practical reality.

## References

- [REF-1] Krafzig, D., Banke, K., and Slama, D. *Enterprise SOA: Service-Oriented Architecture Best Practices*. Prentice Hall, 2005.
- [REF-2] Papazoglou, M. Service-Oriented Computing: Concepts, Characteristics and Directions, in: *Proceedings of 4th International Conference on Web Information Systems Engineering (WISE 2003)*, Rome, Italy, 3-12 (2003).
- [REF-3] Jonkers, H., Lankhorst, M., van Buuren, R., Hoppenbrouwers, S., Bonsangue, M., and van der Torre, L. (2004). Concepts for Modelling Enterprise Architectures. *International Journal of Cooperative Information Systems*, 13(3), 257–287.
- [REF-4] Nayak, N., Linehan, M., Nigam, A., Marston, D., Jeng, J., Wu F., Boullery, D., White, L., Nandi, P., Sanz, J. Core Business Architecture for a Service- Oriented Enterprise. *IBM Systems Journal*, 46(4):723-742, 2007.
- [REF-5] Porter, M. E. (1985). *Competitive Advantage: Creating and Sustaining Superior Performance*. Free Press, New York.

THE PRENTICE HALL SERVICE-ORIENTED COMPUTING SERIES FROM THOMAS ERL

