

White Paper



SOA: The missing link between Enterprise Architecture and Solution Architecture

Jaidip Banerjee and Sohel Aziz

Enterprise Architecture (EA) is increasingly being acknowledged as the way to maximize existing IT investment while planning for future growth. As a discipline, enterprise architecture is tasked to ensure Business-IT alignment through architectural oversight and guidance. It provides enterprises with a 4-dimensional model (Business, Information, Application, and Technical architecture) to analyze and determine the right set of IT-related decisions to enable and support business strategies.

Service-Oriented Architecture (SOA) is essentially about bridging the gap between the business and IT through well defined, business-aligned services, developed by subscribing to established design principles, frameworks, patterns, and methods.

The objectives of EA and SOA are quite similar. However, EA is a framework that covers all dimensions of IT architecture for the enterprise, while SOA provides an architectural strategy that uses the concept of "Service" as the underlining business-IT alignment entity.

This paper appears in the SETLabs Briefings, Vol. 5. No. 2, pp. 69-80 - Banerjee, J. and Aziz, S. (2007), Enterprise Architecture (EA), Service Oriented Architecture (SOA), Solution Architecture (SOLA): The Missing Link?

What is the impact of “Services” on the way we develop our enterprise applications? When and what fuels the need for identification and development of a “Service”? Within the context of enterprise architecture, is there a link between solutions developed by using the SOA strategy? Is it possible to define traceability from the enterprise architecture through to the development of solutions via “Services”?

In this article, we address the dependencies between an EA Framework and a SOA strategy from an artifacts point of view and answer the questions posed above by discussing the impact of introducing a first order entity such as “Service” in EA and Solution Architecture (SOLA). SOLA is the architecture associated with a given business solution. Multiple business solutions collectively define Enterprise Architecture.

Enterprise Architecture – Its Purpose

Enterprise architecture is a guide to an organization’s competitive fitness. It is the dynamic process of managing enterprise IT change through a planned transformation. The transformation plan provides the required explanations, prioritizations, and tradeoffs for the involved deliverables.

The aim is to develop business capabilities over time, which gives the enterprise its competitive edge. This requires investment in people, process and technologies with an expectation of attractive returns.

EA disciplines

The essence of Enterprise Architecture is to create a map of IT assets and business processes and a set of governance processes that drive the alignment between business and IT. There are many different frameworks, to name a few - TOGAF, DODAF, Zachman - for beginning an EA effort. The four architectural disciplines that are commonly accepted as subsets of overall enterprise architecture are highlighted below:

Business Architecture	Defines the business strategy, governance, organization, and key business processes
Applications Architecture	Provides a blueprint of the individual application systems to be deployed, their interactions and relationship with the core business processes of the organization
Information Architecture	Describes the structure of an organization’s logical and physical data assets and data management resources
Technology Architecture	Describes the software infrastructure that supports the deployment of core, mission-critical applications. This type of software is sometimes called the “middleware”

EA without a transformation method

Until recently, most EA efforts focused on developing an enterprise map of IT assets and a set of technical standards that aimed to harmonize IT procurement. This has always been a significant, time consuming and expensive effort, the ROI of which is often opaque to the business.

Standardizing, mapping and controlling IT assets do not make businesses more flexible, capable or profitable. Without a mechanism to deliver the standardization and reuse that it advocates, EA efforts can often fail or become completely technology-centric.

An EA exercise must ensure that there are two major deliverables. First, the target enterprise IT architecture based on the business requirements vision, and second, a governance model that enables the achievement of that vision through a well defined and well orchestrated transformation plan.

SOA and its impact on EA

Advances in integration technology, primarily intelligent and flexible middleware and web services, are providing new ways for designing more agile, more responsive enterprise architecture that can provide the kind of value businesses seek. With new architecture, IT can build new business capabilities faster, cheaper and in a vocabulary the business understands.

These advances are giving a new life to old concepts like Services and Events, which can inspire new EA efforts and revive failing ones.

The first concept is “Service”. This provides business value that was rarely more than a vague promise in the old enterprise architecture. Business people can call for a service in their language and IT can quickly link it with other services to form a process or, if need be, build a new ‘application’.

The second concept is “Event”. An event is something that happens due to changes in enterprise business processes or external influences that cause or influence a shift in enterprise behaviour. If services make enterprise architecture more flexible and responsive, events make them more alert and better prepared to address risks before they become issues.

What is SOA?

Several definitions of SOA exist from IBM, ZapThink and CBDI-Everware to name a few. Essentially, “Service-Oriented Architecture is an architectural strategy that aims to isolate and separate the consumption of business functionality from the provision of this functionality through a commonly defined and accepted service contract”

Key Concepts

The key concepts of SOA revolve around Service Provider, Service Consumer and Service Registry, as shown in Figure 1.

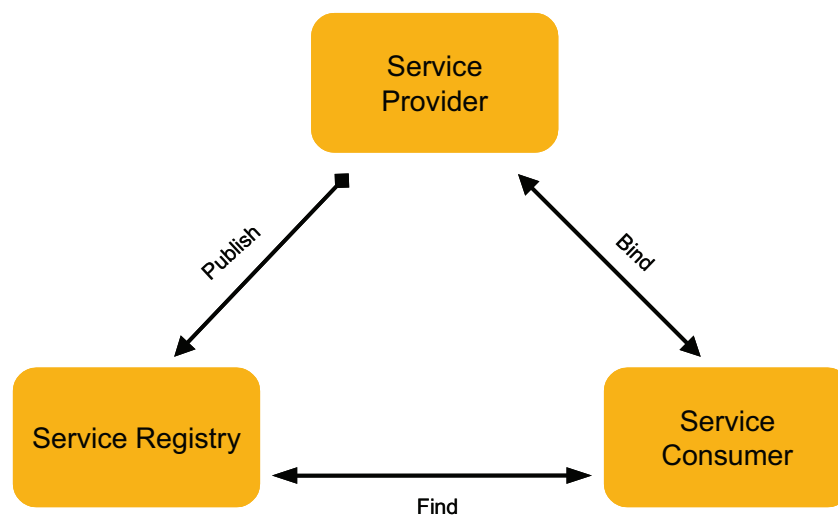


Figure 1: SOA - Key Concepts

Service	Encapsulates business domain specific logic and is exposed through an open standards based interface
Service Contract	Stipulates the terms and conditions under which a service is provided
Service Consumer	Consumes a service or an assembly of services to deliver a particular business process
Service Provider	Provides services based on a pre-defined service contract that guarantees a minimum service level which may include performance, reliability and usage cost
Service Registry	Holds the descriptions and contracts associated with the services available for consumption

Service Orientation

“Service Orientation” is a design philosophy that makes IT resources available on a network in a location independent way. This philosophy allows designers to furnish a layer of abstraction, masking the technical complexities that underlie the services.

This also allows business users to access IT functionality as and when needed, in a flexible, dynamic and cost-effective manner.

Approach to Service Orientation

The two main approaches to introducing service orientation within an enterprise are:

Process-driven (Top-Down) Approach

This approach takes advantage of business processes to identify and categorize services whose implementations can be choreographed to provide scaleable and flexible on-demand solutions. The top-down approach concentrates on an organization's ability to model itself as a provider of "Services" (Business Services). This is realized by:

- Identifying business processes and events
- Gathering business information requirement for the identified processes
- Decomposing the business processes to a level of granularity wherein they are self-contained and independent
- Classifying these self-contained and independent units, also known as "Business Services", for realization by IT

Application-driven (Bottom-Up) Approach

This approach harnesses existing applications and identifies areas of loose coupling and reusability before the development of core services. Assembled services are supported by coherent interfaces to provide scaleable and flexible on-demand solutions.

Steps to introducing SOA to an existing legacy environment following a bottom-up approach generally are:

- Identifying functionality within the legacy systems to be published as services
- Re-factoring and wrapping the functions to be exposed as services
- Layering of services
- Orchestrating the exposed services to achieve the functionality provided by existing applications
- Re-evaluating the existing applications portfolio with the aim of weeding out redundancies

Identification of the business process and its inter-dependencies, business events, business information requirements, application and technology landscapes are some of the key outcomes of EA exercises.

- Business process and its dependencies define how business goals/ objectives are achieved.
- Business events define when a business process must be invoked to achieve the desired objective.
- Business information defines the information needs of the business process and business event.
- Application and Technology landscapes define the bricks and mortars of an enterprise.

Service Portfolio for SOA

As we identify “Services”, there is for rationalization in line with the business processes/ applications these serve have been perceived to serve.

As services are identified, it is also necessary to record their functional and non-functional details in a structured manner and create a portfolio - *Service Portfolio*.

A ‘*service portfolio*’ is a consolidation of all the services identified as required to support the business. It holds information such as service definition, access and usage policies, and non-functional requirements for a service.

Figure 3 illustrates the process of populating a service portfolio in top-down and bottom-up approaches.

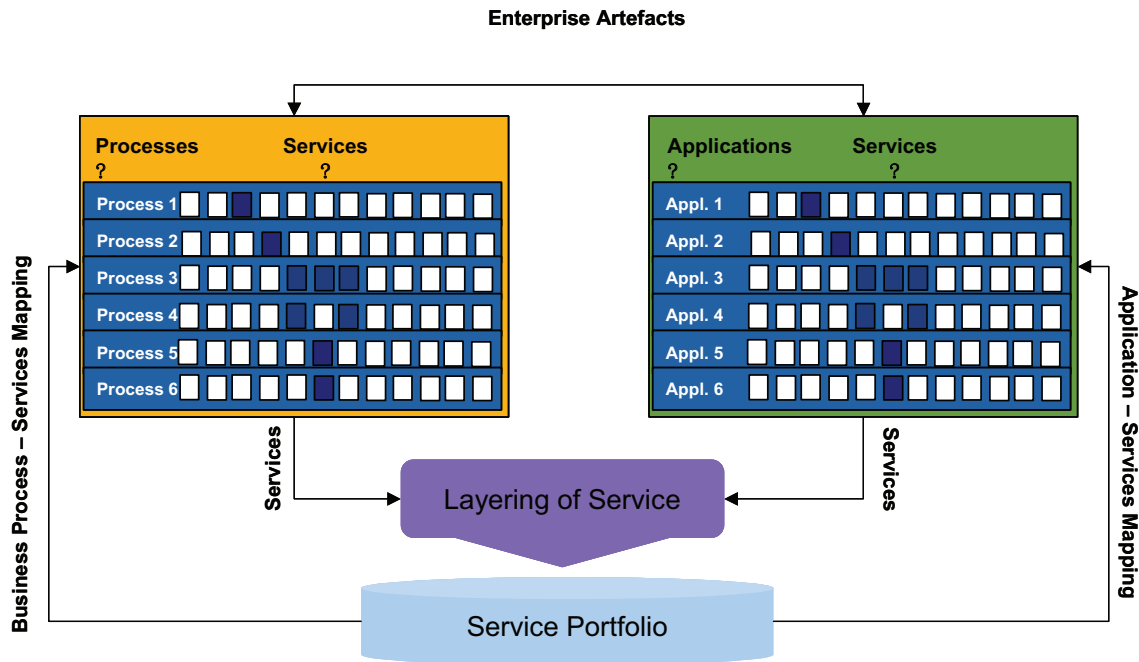


Figure 3: Service Portfolio

The dependencies between enterprise architecture and service-oriented architecture can be addressed by a *portfolio of services that aligns business and IT via a set of planned business services. It provides the mechanism to feed processes, translating business needs into a flexible design capable of easily adopting to business changes.*

A service portfolio is impacted by the business vision and objectives, and how and when the business wants to achieve its objectives. This in turn drives the identification, creation, definition, and specification of services.

A service portfolio plan provides enterprise users with three kinds of views: Conceptual, Logical and Physical as elaborated below:

Conceptual view	Supports the conceptualization of service and governance needs
Logical view	Supports the solutions architecture components for the services conceptualized
Physical view	Identifies the physical implementation components of the services and their characteristics

A central repository is required to ensure that the contextualized information on these artifacts is available to a wider community within the enterprise, in a reliable and searchable way.

This repository is also home to a *Service Portfolio Plan* - a planned list of defined services resulting from the decomposition and subsequent classification of EA artifacts such as business process, events, and information.

Services & Patterns

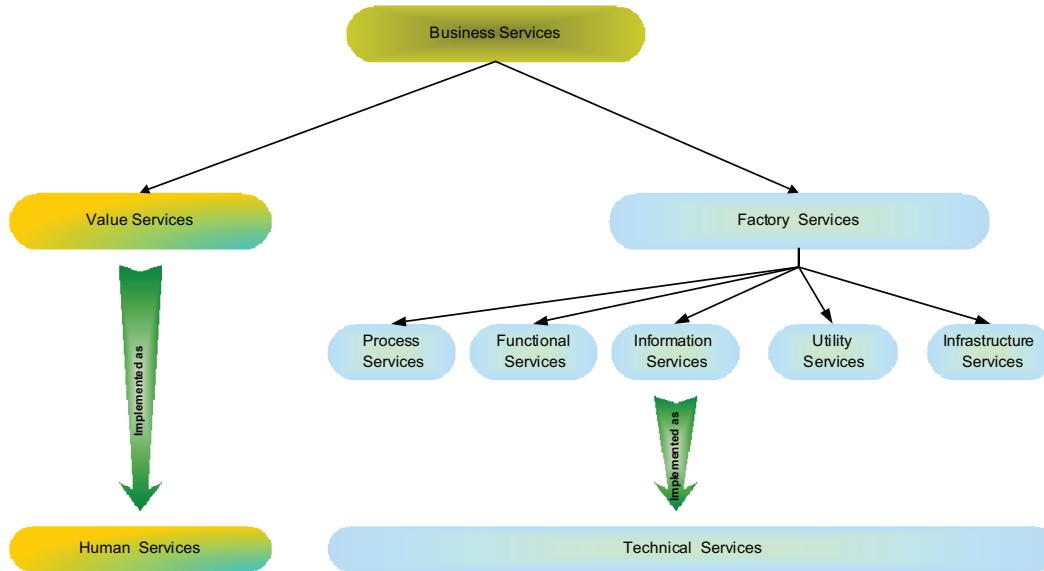


Figure 4: Services & Patterns

In the top-down approach to service orientation, business processes are core to the business. When decomposed to a level where they can be considered self-contained and independent, these processes become *Business Services*.

Once business services are identified, we can decompose and classify them further, based on established patterns to populate the service portfolio with information that is required to support the different views that a service portfolio offers to its users.

Business services can be classified as “Value Services” i.e. services that can only be provided manually, and “Factory Services” i.e. services that are candidates for automation and are implemented by “Technical Services”. Technical services are the physical implementations that support various service patterns. These services implement both functional and non-functional requirements. They include infrastructure services that help in operating the services and provide monitoring and management capabilities.

How does EA enable an SOA strategy?

One of the outcomes of an EA exercise - the definition of an enterprise business process model – provides a basis for service orientation through a top-down approach.

The dependency of a planned list of EA artifacts on the development of SOA is shown in Figure 5.

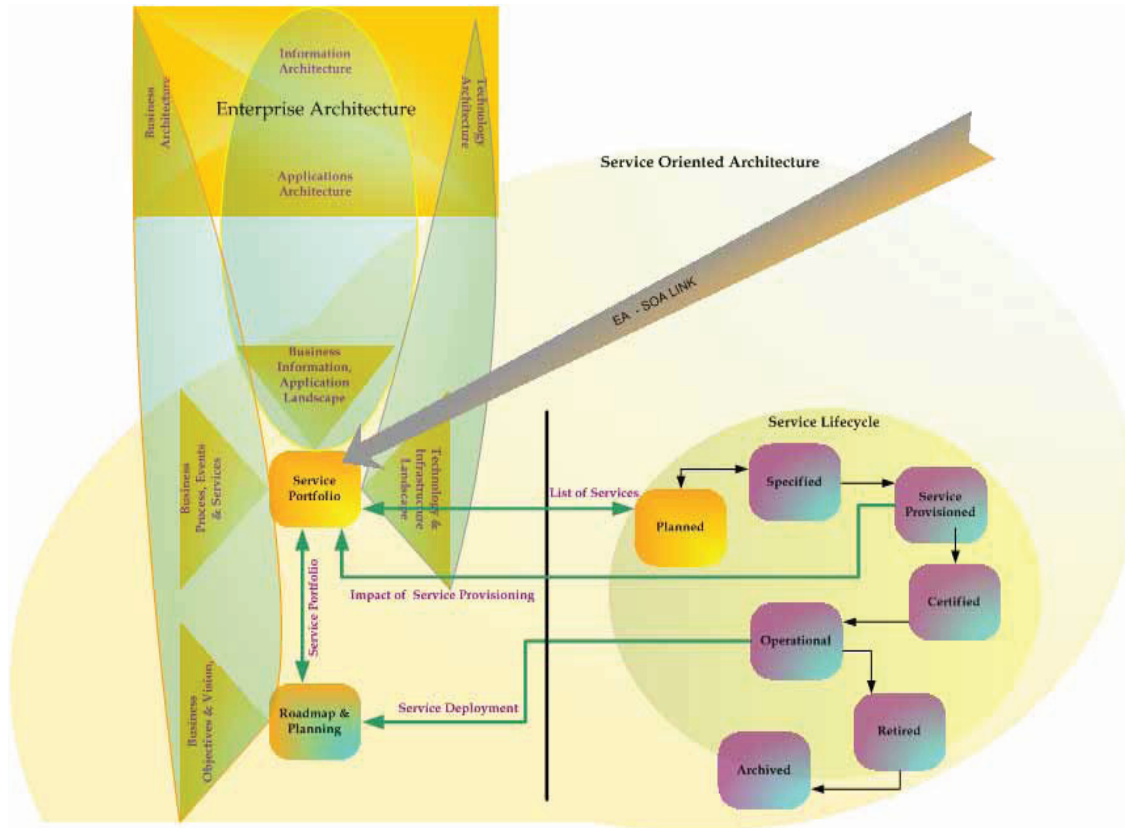


Figure 5: EA artifacts for SOA

Any change within the enterprise has an impact on the constituents of service portfolio, e.g. new business line, introduction of critical customer service SLAs, improved security needs for a business operation, etc. Similarly, provisioned services (services for which detailed specification and system-tested software exist, and are ready to be certified) have to be reflected in the EA artifacts to maintain integrity between the proposed and implemented services.

Solutions Architecture (SOLA)

Solution architecture describes the components and elements required to deliver a *solution*, how they fit together, and the core technologies required. The solution delivers the objectives derived from the business and IT requirements.

Service-Oriented SOLA

Key to comprehensive solution architecture is clarity of the objectives. As well as being precise in the requirement, the definitions have to be located in an environment visible to all stakeholders.

The defined business services must be complete to ensure reusability of services. Designers need to understand the services to manage changes in the service design. Program managers' must be aware of these services to promote reuse and avoid duplication.

As mentioned earlier, identified services, along with their functional and non-functional requirements, have to be in a location that is visible to the enterprise stakeholders. Service Repositories can be such a location.

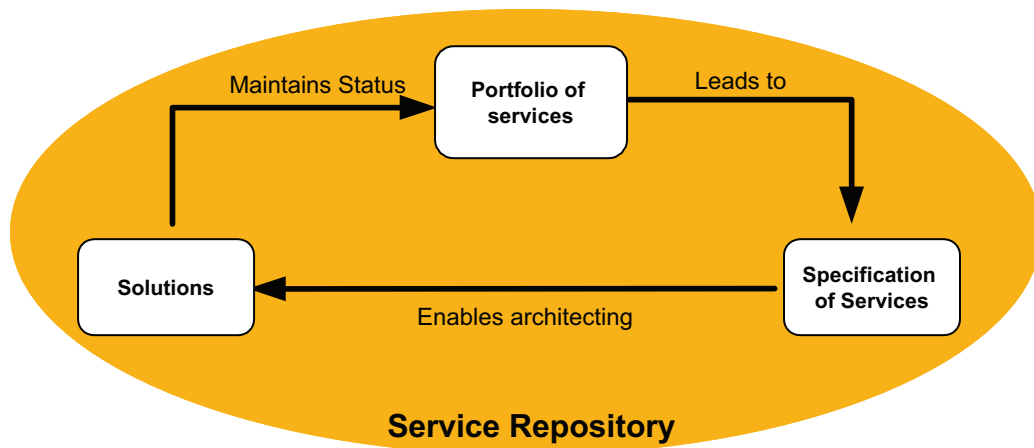


Figure 6: Service-Oriented SOL A

For the services to be developed in line with the concepts of service orientation, solutions have to be crafted, verified, documented and made available to the stakeholders before any development work starts.

Detailed specifications of the services provide an insight into the to-be architecture of the solution. Once the solution is agreed upon, the service portfolio must be updated to reflect current state of the relevant service.

Service Lifecycle

The key to solutions architecture lies in the articulation of its work products, the objectives that the solution is expected to achieve, and dependencies that need to be addressed before and while developing the solution design.

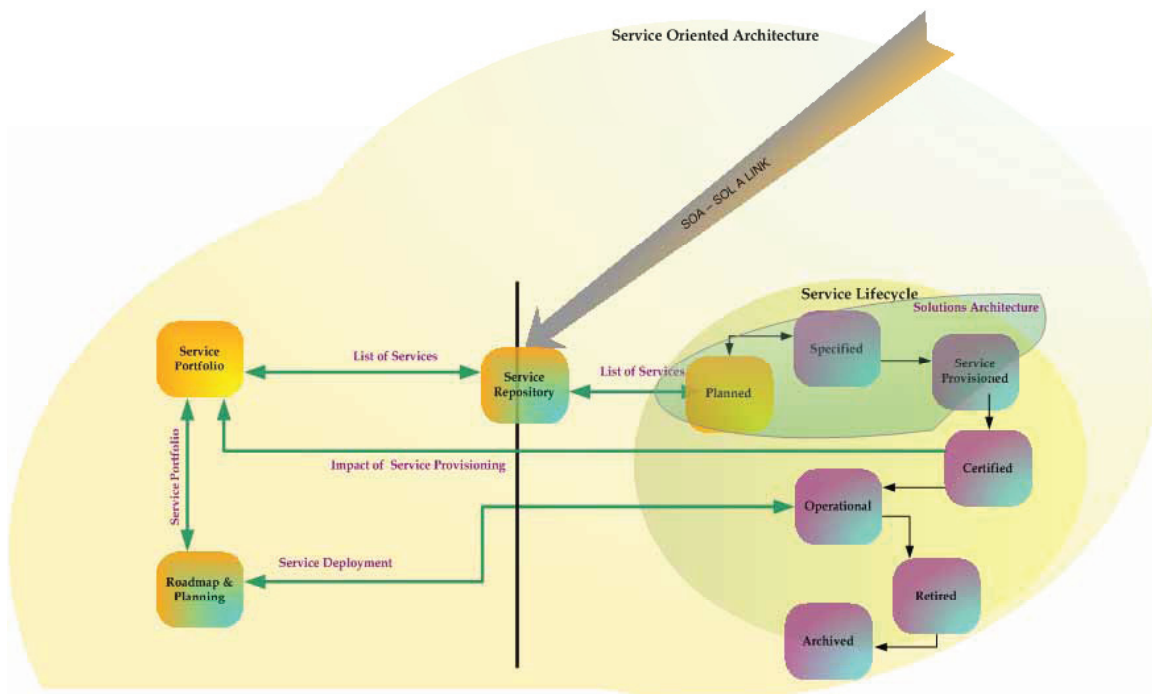


Figure 7: Service Lifecycle

A “service repository” is a natural fit as it provides a centralized view of all service-related information to the service stakeholders. A relationship naturally exists between EA and SOLA via a service repository is shown in Figure 7.

Service Repository

A service repository plays a critical role in providing visibility into services for reuse, governing service production and consumption, and measuring the benefits of SOA. As it provides access to all stakeholders, the accuracy and integrity of information in the repository is maintained by its collaboration capabilities.

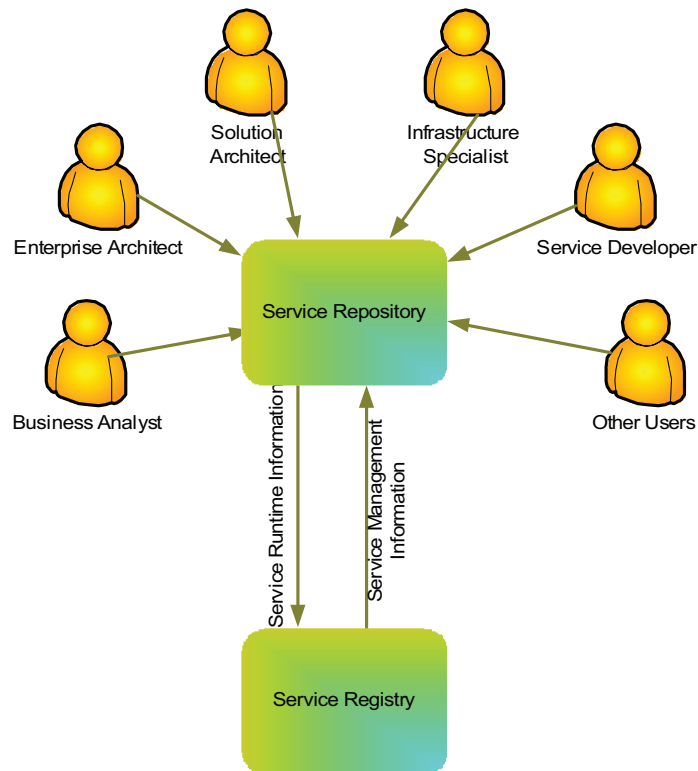


Figure 8: Service Repository

A simplified representation of a service repository is shown in Figure 8. The repository caters to all information pertaining to a service lifecycle.

The table below lists some of the key responsibilities and key collaboration facilities provided by the service repository:

Responsibilities	Maintain a catalogue of all service artifacts
	Provide service change management
	Enforce policies
	Provide dependency management
Collaborations	Provide a comprehensive search facility
	Provide metadata
	Allow service developers access via design tools
	Enable service management information for Infrastructure Specialists
	Provide impact analysis capability to service designers Provide architects with data to analyze service utilization and deployment

Driving SOLA

Solution development is always influenced by the business objectives and is triggered by the service deployment plans recorded in the service portfolio. These deployment plans are always tied to the timelines of the business objectives, and hence, crucial to the development and implementation of the solution.

Policies around non-functional requirements and quality criteria are usually detailed against planned services and form the basis for the solutions architecture.

Figure 9 shows the flow of development and implementation of services.

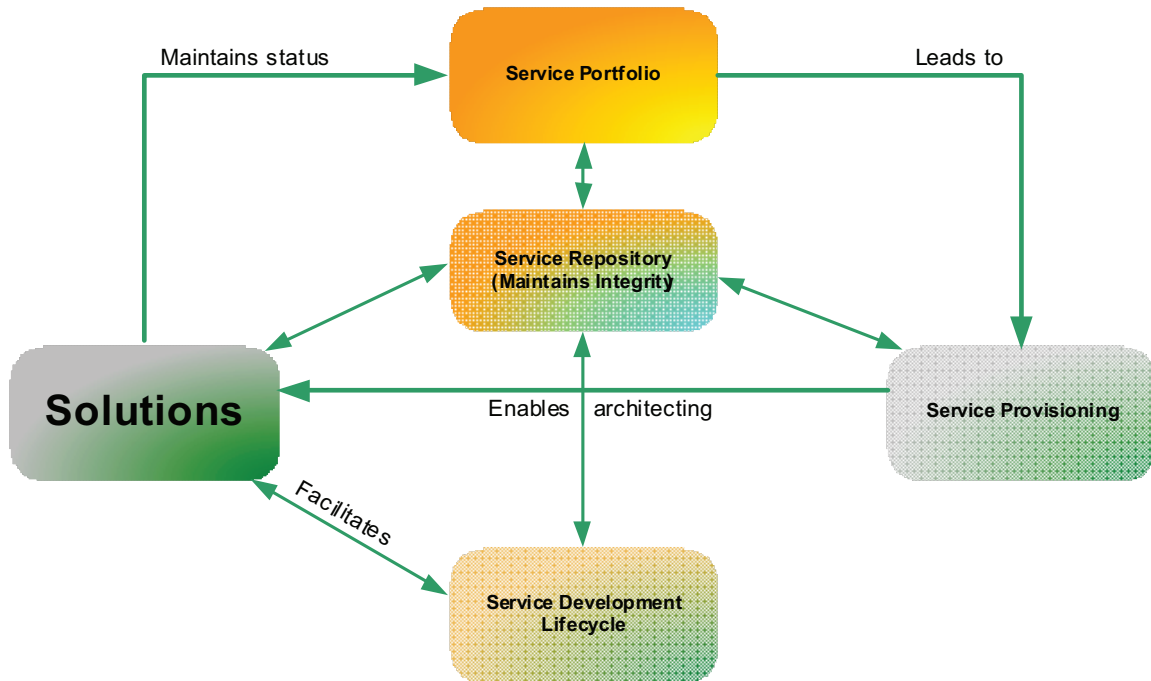


Figure 9: Delivering SOLA in a Service-Oriented fashion

Information recorded in the service repository enables the design and development of solutions, and also reflects changes in the information characteristics, i.e.; when a solution was defined and implemented.

A rich specification of services, completed as part of service planning is critical to solutions design and development. This provides detailed information on the core functional and non-functional capabilities of the services being developed in line with planned services in the service portfolio.

EA to SOLA - The Missing Link

Following the trail of services, from its abstraction within EA through its definition in SOA, and finally in the solution design and deployment, we have unearthed the constituents a common thread that is the missing link.

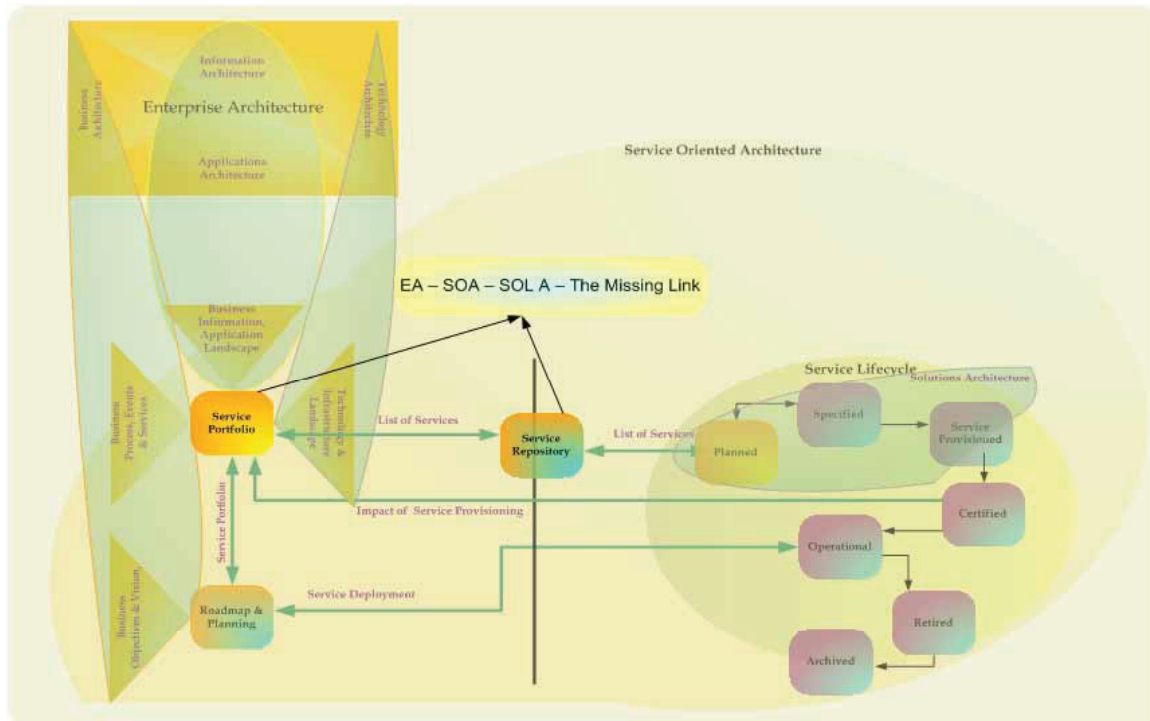


Figure 10: EA - SOL A - The Missing Link

Figure 10 highlights this missing link – the artifacts *Service Portfolio* and *Service Repository*. These artifacts drive the transition from envisioning services as a means to achieve business objectives to managing its lifecycle after implementation. They provide the quality of service data to enable continuous improvement. Between them, they also enable SOA governance and make visible all changes, both current and future.

SOA without EA

Organizations without an EA function, more often than not, have some of the elements of EA artifacts, i.e. business processes, business event, etc. These artifacts in most cases are not recorded; they often are “word of mouth” artifacts. Furthermore, there is no clear planning process for IT architecture.

For an organization to build business solutions with a strong focus on reusability and an agile infrastructure that can change with the changing business requirements, organizations require an enterprise perspective on needs and their implications. This cannot be achieved without an effective EA program.

Without an EA program, there is a clear lack of IT appreciation of business changes and its impact on IT. This lack of visibility results in management’s reliance on inaccurate or incomplete information about the state of the enterprise to arrive at key management decisions. SOA adoption in this case is often “technology” focused and restricted to individual projects. A critical success factor for SOA is its adoption across projects - focusing not only on technical concepts but also on functional aspects, within the boundaries of the reference architecture and managing its introduction as a programme. Without the right focus, SOA adoption can easily lead to an application portfolio in which all applications/services depend on each other in enterprise-level spaghetti architecture.

Conclusion

Service Portfolio and *Service Repository* provide the link between how the business perceives solutions to achieve its objectives and how IT interprets the perception and its implications. They help create a common view of how an enterprise achieves and maintains agility.

The need for a service portfolio is undoubtedly critical. It has to be acknowledged that undertaking such a task of building a service portfolio is anything but trivial, both in terms of complexity and cost. Enterprises must embark on building their service portfolio in an incremental manner and not in a big bang fashion.

Key to the success of any project is in the visibility and communication of its intention and consequences. Service repository is the enabling component that aids stakeholders interested in service-oriented information – definition, design, implementation, testing, use, and retirement – obtain it in a manner appropriate for their objectives.

References

1. ZapThink, LLC : <http://www.zapthink.com/>
2. IBM Developerworks : <http://www-128.ibm.com/developerworks/>
3. The Open Group : <http://www.opengroup.org/>
4. SOA Enterprise Patterns: <http://orchestrationpatterns.com/>
5. CBDI Forum: <http://www.cbdiforum.com/>

About the Author

Jaidip Banerjee is a Senior Architect with Infosys. He has extensive experience in enterprise architecture definition and implementation.

Sohel Aziz is an Associate Vice President and Principal Architect with Infosys. He has extensive experiences in IT strategy, and enterprise architecture definition and implementation.



For more information, contact askus@infosys.com

About Infosys

Many of the world's most successful organizations rely on Infosys to deliver measurable business value. Infosys provides business consulting, technology, engineering and outsourcing services to help clients in over 30 countries build tomorrow's enterprise.

For more information about Infosys (NASDAQ:INFY), visit www.infosys.com.