



IFEAD is an independent research and information exchange organization working on the future state of Enterprise Architecture.

What you all need to know about Services Orientation!!

Structuring the Enterprise around Services

The Differences between Hype, Hope and Reality?

**"Things should be made as simple as possible, but not any simpler." --
Albert Einstein**

By J. Schekkerman¹

The terms Enterprise Architecture (**EA**), Services Oriented Enterprise (**SOE**), Service-Oriented Architecture (**SOA**) and Service Oriented Computing (**SOC**) are being exposed to an ever wider and more influential audience. Unfortunately, as with many "new concepts" there is a common misunderstanding about prior ideas and practices from which they are derived. Accordingly, these terms will be bandied about as buzzwords and/or marketing hyperbole assaults.

This section is a pre-emptory effort to provide both a basic and somewhat advanced understanding of these terms: why they are important to us, where they come from and what this means for business & information technology (IT). As with most innovative concepts that seemingly come into vogue in a sudden and haphazard manner, there is both a history leading up to its short sojourn in the spotlight of popular perception and a predictable fade unless popular uptake makes it a structuring paradigm of the Enterprise. With reference to EA, SOE, SOA and SOC, it is fairly certain that they will have their role in this paradigm shift.

Frequently asked questions that revolve around this topic include: What is SPA? What is SOE? What is SOA? and what is SOC and how are they interrelated? To answer this multi part question, let us first try to gain the proper perspective for being able to see the overall Enterprise Architecture landscape. This is commonly called the Heliview. This view refers to the landscape from a chopper at a few thousands meter height and, that is an apt metaphor, since at that height and from that perspective we see a larger picture. We see the mountain ranges not just the mountain in front of us, or the trail we are hiking up through the trees in the foothills and lower slopes. We see the checkerboard layout of farmlands. We see the green meandering paths of the watershed streams and rivers, and this is appropriate to the level of Enterprise Architecture (EA). In this view an enterprise is seen simply as an organization or a part of an organization.

Enterprise Architecture in the Context of Services Orientation

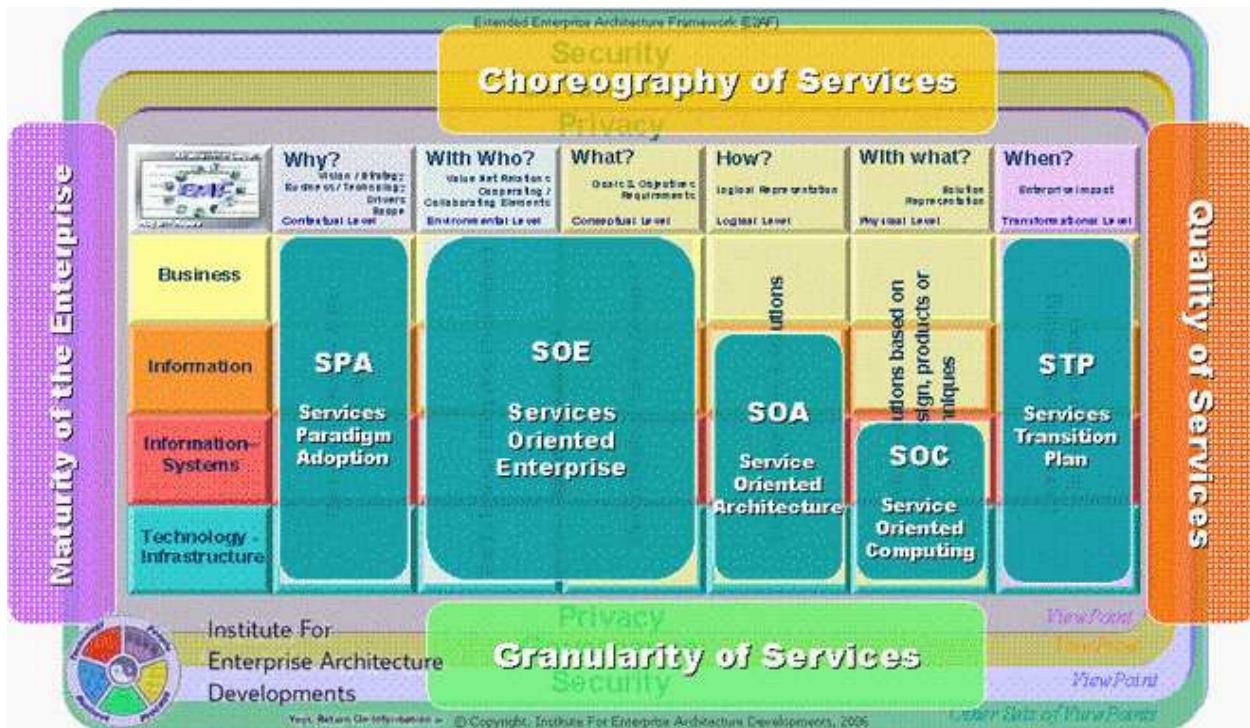
However to truly see the outlines and parameters of what is called EA, and from that perspective begin to understand how a SOE and SOA can enable an EA, we need to get up to

¹ Jaap Schekkerman BSc, is the Founder, President & Thought Leader of the Institute For Enterprise Architecture Developments. Jaap Schekkerman is the Opinion Leader Enterprise Architecture of Verdonck, Klooster & Associates and the Vice-President of the International Association of Enterprise Architects.



IFEAD is an independent research and information exchange organization working on the future state of Enterprise Architecture.

sub orbital or low orbit viewpoint at least, and it could be argued that we need to get all the way out to the viewpoint of our moon to see the entire environment within which these concepts work. The fact is that the word enterprise is by no means restricted to a business, or even an industry, nor does it refer to a particular time in the life of an organization. The enterprise encompasses the entire life cycle of an organization or an organism.



Enterprise Architecture Services Model

After having achieved the desired scope of view, we can apply that perspective, the perspective of the economic or ecological lifecycle, to provide measures to ensure an extended life beyond the restrictions of the individual biological entity model we necessarily bring with us. Enterprises, like cultures, outlive individual members, and those life cycles require us to extend our vision further than our own lives. In many areas in our lives, such as in educational institutions, the familial clan or kinship networks, governments, etc., we take this viewpoint for granted, but in EA, we now need to be explicit and put in place the kinds of mechanisms that can conduct constant review and institute quality assurance as a matter of course and, in effect, institutionalize the cycle of build, use, learn, assess, build (adjust/rebuild), use, learn, etc., ad infinitum. This presupposes, of course, that the "de facto" starting point for this process is where an enterprise or organization happens to be at that the point when it is decided to begin this process of building and deploying an EA. Some comprehensive review is not required to start. In effect, gaining the correct perspective should allow an enterprise to refocus on the immediate with that larger focus still in mind. Such reviews or foundational reports have their place, but, unless an enterprise is at its founding point, the larger picture can be developed while the more functional aspects of



IFEAD is an independent research and information exchange organization working on the future state of Enterprise Architecture.

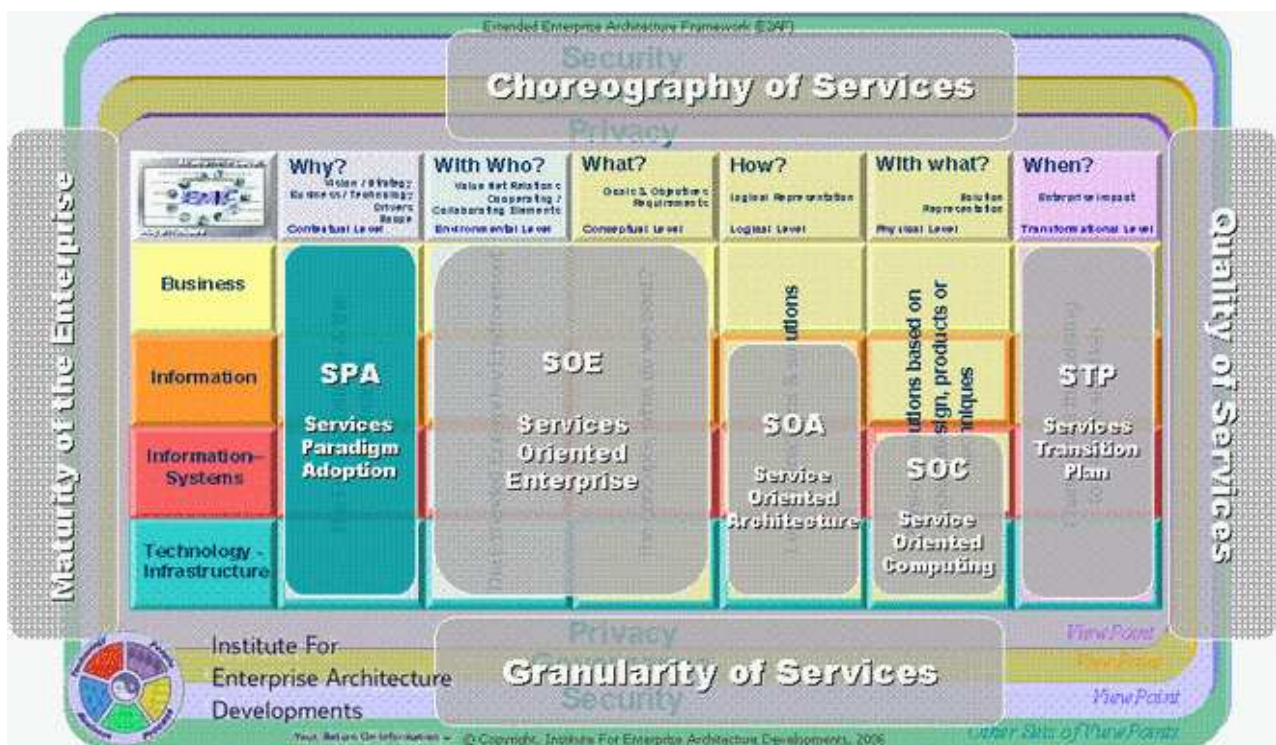
current operations and immediate planning based on immediate findings take precedence in simply getting the process moving.

Differences between hype, Hope and Reality

In the next diagram that is showing the management version of the **Extended Enterprise Architecture Framework (E2AF)** the concepts of SPA, SOE, SOA, SOC and STP are positioned on the framework to help the readers to understand the relative position of these concepts related to the E2AF. Around the E2AF, 4 Critical Success Factors (CSF) are positioned that are critical for the success of implementing Service Orientation in your organization and which makes the difference between Hype, Hope and Reality. So let's have a look at the different concepts of Services Orientation, their relative position in the E2AF and the CSF's to adopt and implement the concepts of Services Orientation (SO).

Services Paradigm Adoption (SPA)

Services orientation presents an ideal vision of a world in which resources are cleanly partitioned and consistently represented in terms of services. Therefore the Services Paradigm has to be adopted by the Enterprise to structure the Enterprise in terms of Services. So the Enterprise aspect areas of Business, Information, Information-Systems and the Technology Infrastructure has to be decomposed in terms of functional Services. Doing that at a consistent and consequent matter will deliver loosely coupled functional services that can be outsourced or insourced or brought together in so called shared services centers.



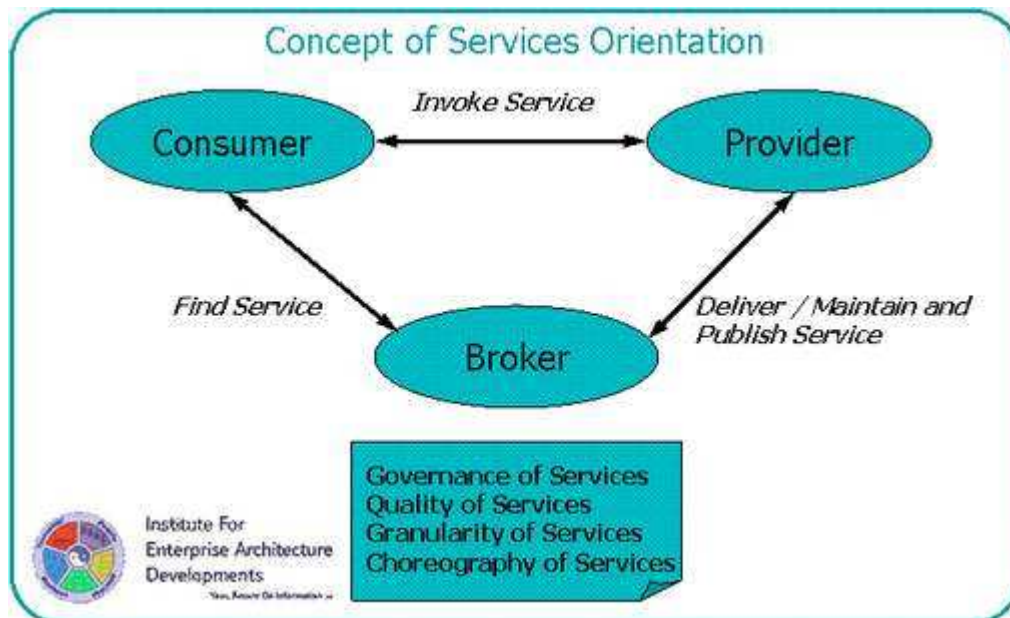


IFEAD is an independent research and information exchange organization working on the future state of Enterprise Architecture.

Adopting the Services Paradigm by your Enterprise and implementing it in an appropriate way, will deliver the Enterprise more flexibility, adaptability and agility than organizations that don't want to adopt the Services Paradigm, however if you can't fulfil or implement the Critical Success Factors, Services Orientation can become a nightmare.

Services Orientation (SO) is an Architectural Style, NOT an Architecture itself.

So, Services Orientation as well as SOA is an architectural style whose goal is to achieve loose coupling among interacting services. A service is a unit of work done by a service provider to achieve desired end results for a service consumer. Both provider and consumer are roles played by organizational units as well as software agents on behalf of their owners.



Service Orientation (SO) is the architectural style that supports loosely coupled services to enable business flexibility in an interoperable, technology-agnostic manner. SO consists of a composite set of business-aligned services that support a flexible and dynamically re-configurable end-to-end business processes realization using interface-based service descriptions. The inherent, salient advantage of today's service descriptions is the decoupling of the SP and SC through open standards and protocols, and the deferment of implementation decisions from the SC to the SP. Moreover, individual or collections of services that enjoy various levels of granularity can be combined and choreographed to produce "new" composite services that not only introduce new levels of reuse but also allow the dynamic reconfiguration of business systems.

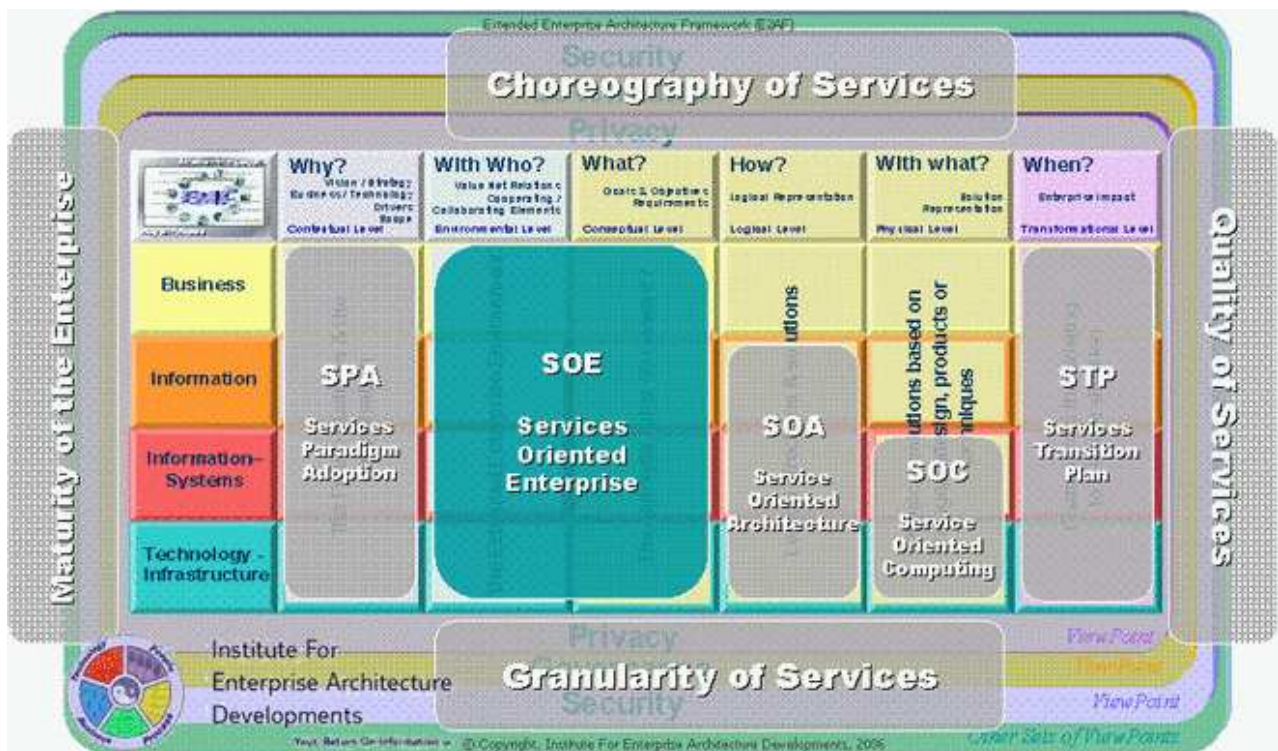
Services Oriented Enterprise (SOE)



IFEAD is an independent research and information exchange organization working on the future state of Enterprise Architecture.

A service-oriented enterprise is really connecting business processes in a much more horizontal fashion. It's having an Enterprise infrastructure that provides an enterprise architecture and security foundation to be able to run these services consistently across your enterprise.

While the concept of a service-oriented architecture has been considered a best practice by system architects for the past three decades, it is now being embraced by organizations everywhere as the key to business agility. But SOE and SOA doesn't come in a box, it's not a single technology and it doesn't render all problems solved. While SOE enables and even drives organizational change, it also requires executives, enterprise architects and program managers to think and act differently or simply find themselves in possession of new problems with few or no new benefits.



What is a Service?

A Service is an implementation of a well-defined business functionality that operates independent of the state of any other Service defined within the system. Services have a well-defined set of interfaces and operate through a pre-defined contract between the client of the Service and the Service itself.

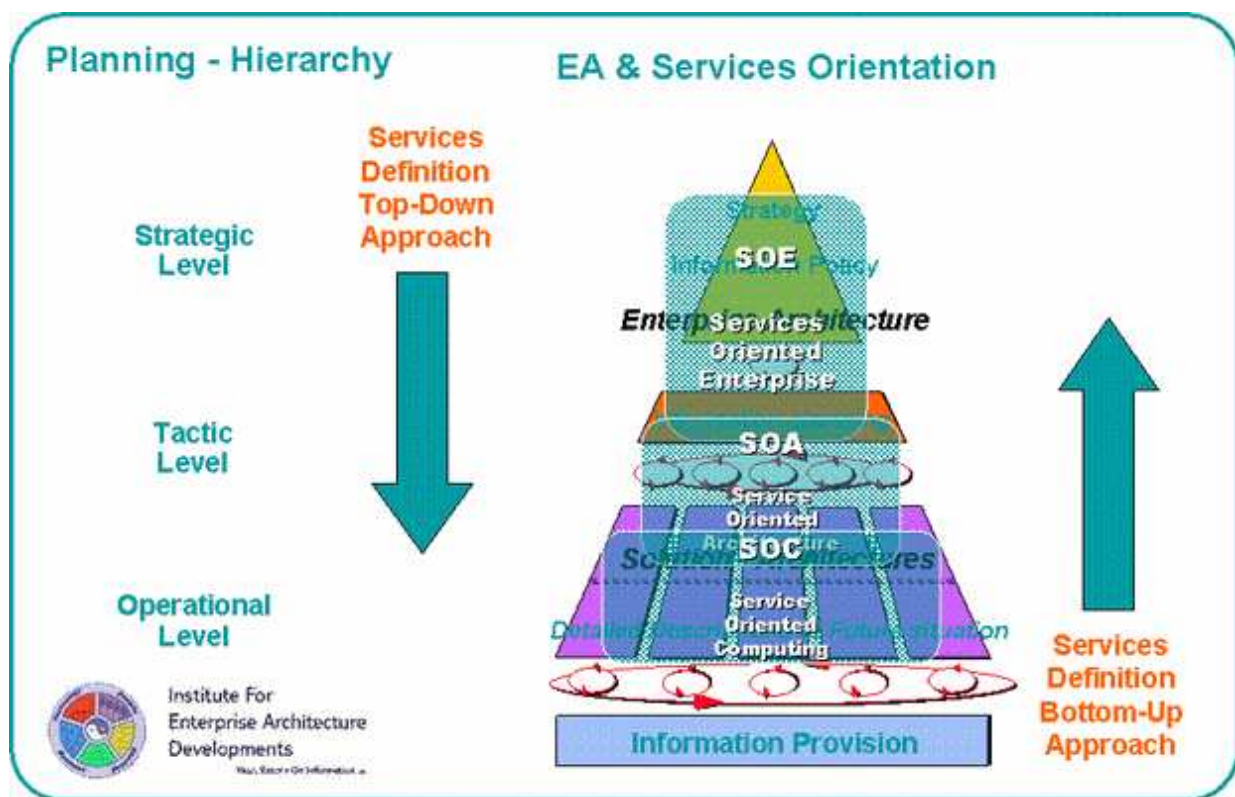
Top-Down versus Bottom-Up Service Definition



IFEAD is an independent research and information exchange organization working on the future state of Enterprise Architecture.

In the ideal situation Services are defined and described Top-Down at Enterprise level in what is called the Service Oriented Enterprise (SOE). From a functional decomposition of well defined Business functionality we can identify the business function 'Financial Services'. This Business function can be decomposed in lower level services like Invoicing, Payments, Banking, etc.

In Service Oriented Architecture, the system operates as a collection of services. Each Service may interact with various other Services to accomplish a certain task. The operation of one Service might be a combination of several low level functions. In that case, these low level functions are NOT considered Services.



Services Orientation; level of Adoption & Ambition

Top-Down - SPA / SOE / SOA

1. On Demand Business Transformation: Broad Business Transformation of existing business models or the deployment of new business models (SOE)
2. Enterprise-wide IT transformation: An enterprise architected implementation enabling integration across business functions throughout an enterprise (SOA/SOE)

Bottom-Up - SOC / SOA

1. Service-oriented integration of business functions Integrating services across multiple applications inside and outside the enterprise for a business objective (SOI/SOA)



IFEAD is an independent research and information exchange organization working on the future state of Enterprise Architecture.

2. Implementing individual Web services Creating services from tasks contained in new or existing applications (SOC/SOA)

Top-down domain decomposition (process modeling and decomposition, variation-oriented analysis, policy and business rules analysis, and domain specific behavior modeling (using grammars and diagrams)) is conducted in parallel with a bottom-up analysis of existing legacy assets that are candidates for componentization (modularization) and service exposure. To catch the business intent behind the project and to align services with this business intent, goal-service modeling is conducted.

Service Oriented Architecture (SOA)

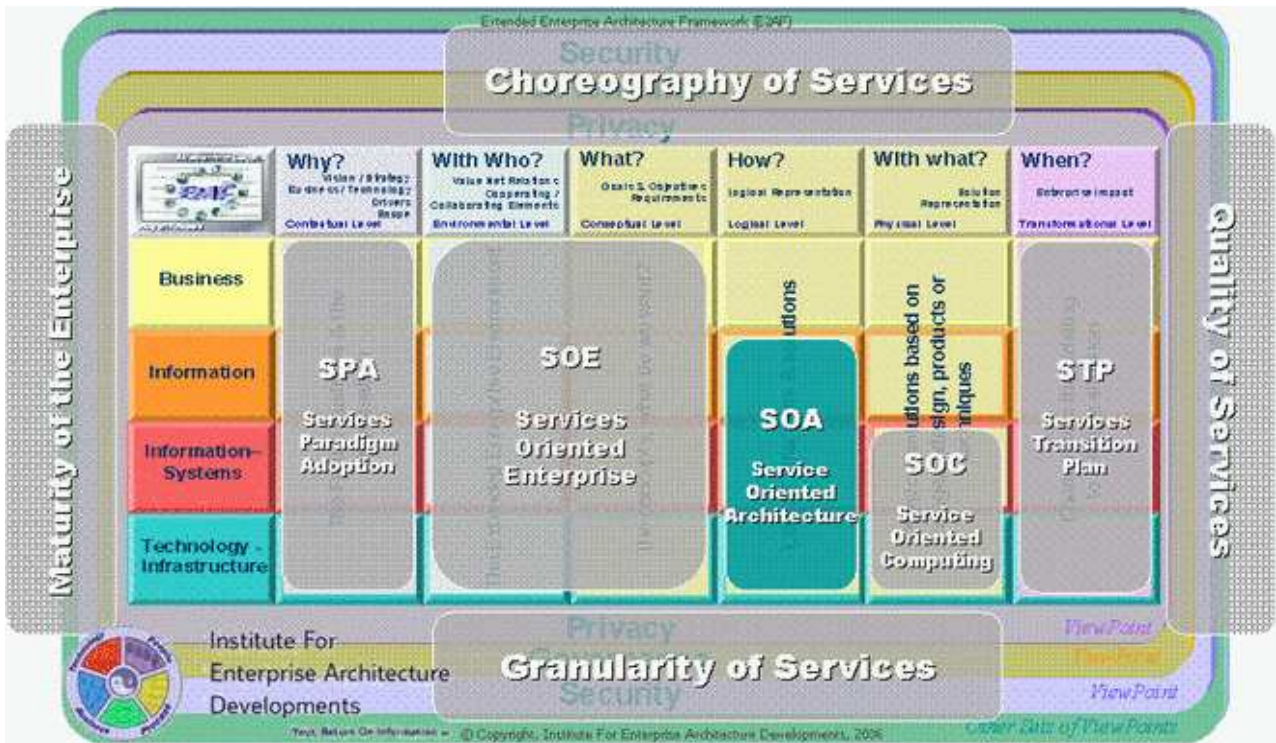
Service-orientation presents an ideal vision of a world in which resources are cleanly partitioned and consistently represented. When applied to IT architecture, service-orientation establishes a universal model in which automation logic and even business logic conform to this vision. This model applies equally to a task, a solution, an enterprise, a community, and beyond.

By adhering to this vision, past technical and philosophical disparities are blanketed by layers of abstraction that introduce a globally accepted standard for representing logic and information. This level of standardization offers an enormous benefit potential for organizations, as many of the traditional challenges faced by ever-changing IT environments can be directly addressed through the application of these standardized layers.

For example, by shaping automation logic through service-orientation, existing investments can be leveraged, business intelligence can be accurately expressed, and inherent automation agility can be achieved. When coupled with the Web services technology platform, SOA offers a significant and real potential for manifesting these benefits.



IFEAD is an independent research and information exchange organization working on the future state of Enterprise Architecture.



The service-orientation ideal has sparked a movement that has positioned SOA as the next phase in the evolution of business automation. In the same manner in which mainframe systems were succeeded by client-server applications, and client-server environments then evolved into distributed solutions based on Web technologies, the contemporary, Web services-driven SOA is succeeding traditional distributed architecture on a global scale.

All major software manufacturers and vendors are promoting support for SOA—some even through direct involvement in the development of open standards. As a result, every major development platform now officially supports the creation of service-oriented solutions.

Be forewarned, though, that SOA makes impositions. A change in mind set is required, as business logic now needs to be viewed within a service-oriented context. Applying this context also requires a change in automation logic, as solutions now need to be built in support of service-orientation. Finally, a technical architecture capable of hosting service-oriented automation logic further introduces new technology and infrastructure requirements.

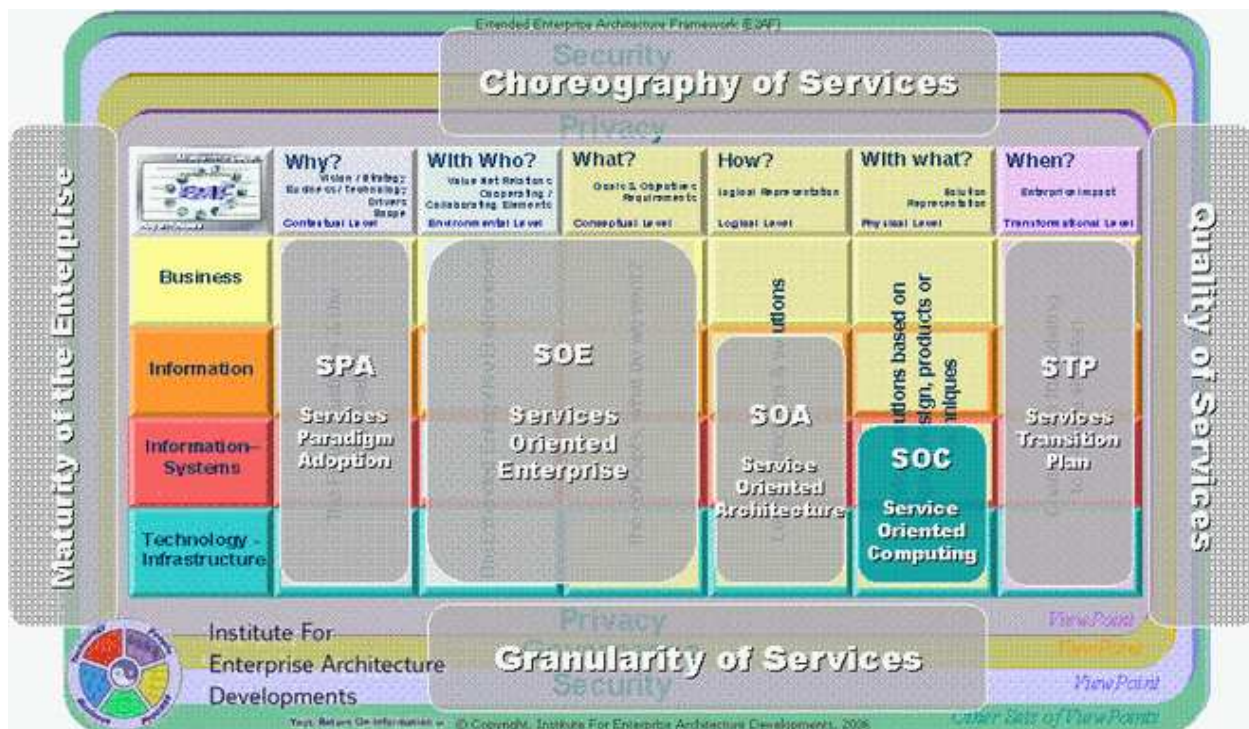
Services Oriented Computing (SOC)

There has been an increase in interest recently within the service oriented community towards "Service Oriented" Computing. Services are often seen as a natural progression from component based software development, and as a means to integrate different component development frameworks. A service in this context may be defined as a behavior that is provided by a component for use by any other component based on a network-addressable



IFEAD is an independent research and information exchange organization working on the future state of Enterprise Architecture.

interface contract (generally identifying some capability provided by the service). A service stresses interoperability and may be dynamically discovered and used. According to the "Open Grid Services Architecture" (OGSA) framework, the service abstraction may be used to specify access to computational resources, storage resources, and networks in a unified way. How the actual service is implemented is hidden from the user through the service interface. Hence, a compute service may be implemented on a single or multi-processor machine -- however, these details may not be directly exposed in the service contract. The granularity of a service can vary -- and a service can be hosted on a single machine, or it may be distributed.



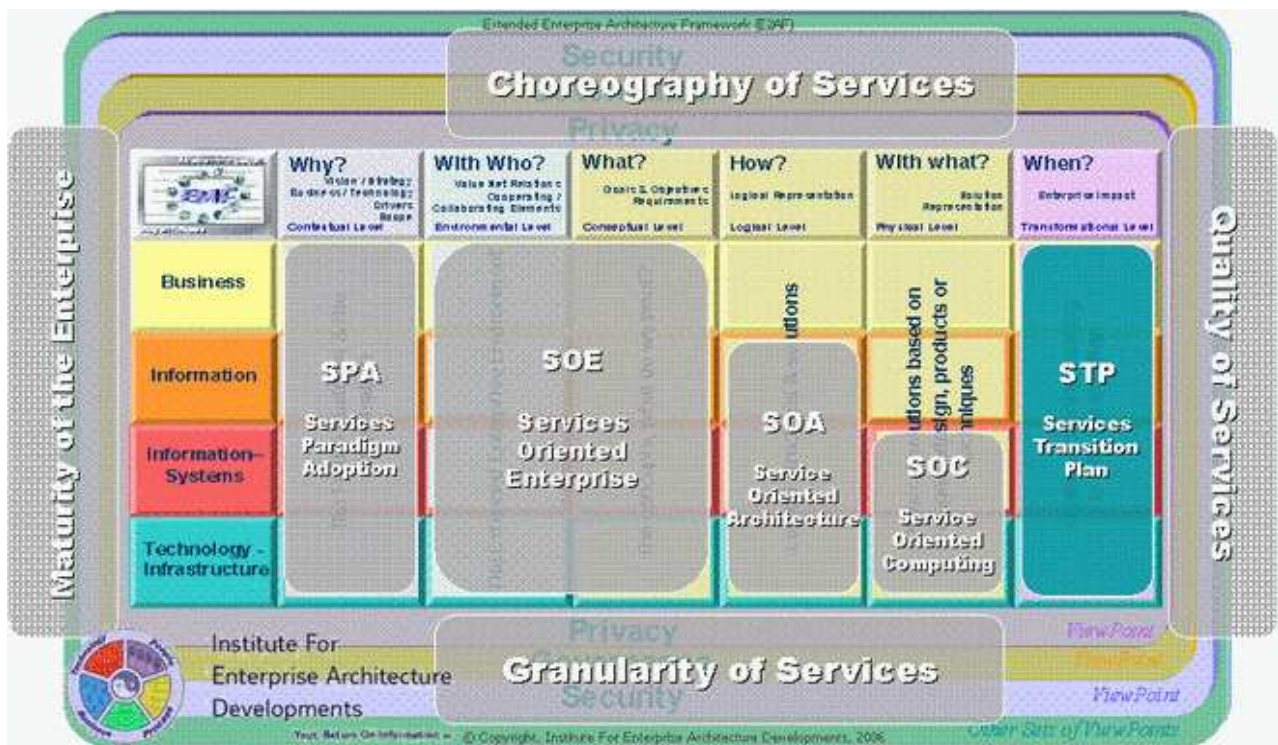
Web Services provide an important instantiation of the Services paradigm, and comprise infrastructure for specifying service properties (in XML -- via the Web Services Description Language (WSDL) for instance), interaction between services (via SOAP), mechanisms for service invocation through a variety of protocols and messaging systems (via the Web Services Invocation Framework), support for a services registry (via UDDI), tunneling through firewalls (via a Web Services Gateway), and scheduling (via the Web Services Choreography Language). A variety of languages and support infrastructure for Web Services has appeared in recent months -- although some of these are still specifications at this stage with no supporting implementation. Web Services play an important role in the Semantic Web vision, aiming to add machine-processable information to the largely human-language content currently on the Web. A list of publicly accessible Web Services (defined in WSDL) can be found at www.xmethods.net. By providing metadata to enable machine processing of information, the Semantic Web provides a useful mechanism to enable automatic interaction between software -- thereby also providing a useful environment for agent systems to interact.



IFEAD is an independent research and information exchange organization working on the future state of Enterprise Architecture.

Services Transition Plan (STP)

Your SOE, SOA and SOC is very much a work in progress, and your transition to Services Orientation will take place over a long period, in many phases. Consequently, transition management is one of the most critical concerns in the long ramp-up to Service Orientation everywhere.



Despite the magnitude of a migration to a services-oriented platform, the continuing uncertainty of critical WS-* standards, and the often thundering impact of large-scale SOA deployments, now is the time to start considering the move. The key to a successful transition is to find a spot of calm amidst the storm of activity surrounding SOA, and develop an intuitive plan that will guide your organization through a path of technical obstacles, organizational resistance, and ever-shifting industry trends.

Invest in an Impact Analysis Before Developing the Transition Plan

In order to assess the feasibility of a transition to Services Orientation, you first need to estimate the real-world impact such a migration will have. Therefore, you should consider holding off any sort of planning until an initial impact analysis is complete. Using the impact analysis results as your primary guide, and factoring in budget constraints, related project requirements, and other external drivers (such as strategic business goals), you should be able to determine the scope of your planned transition. It is not uncommon for an SOA transition plan to apply only to a subset of an organization's technical environment. For instance, there may be several legacy areas of an enterprise that do not warrant any intrusion



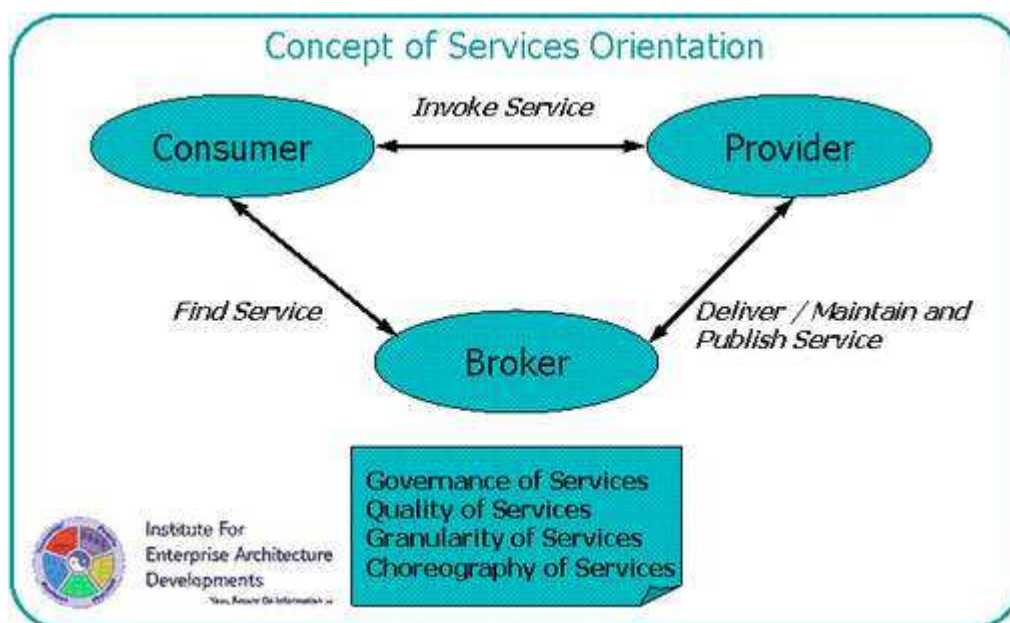
IFEAD is an independent research and information exchange organization working on the future state of Enterprise Architecture.

by service encapsulation. Perhaps your goal is to build a dedicated hosting environment intended only to support new service-oriented applications. More often than not, however, integration requirements drive SOA transitions, in which case the scope of your project could easily see the introduction of SOA affect a majority of your IT enterprise.

Service-oriented principles themselves are not complex, however, the application of these principles can result in relatively complex automation solutions. This is especially the case when services from different solutions are shared and composed to support new or modified business processes. If you're going to live in a service-oriented world, your project team will need to change the way it thinks about fundamental aspects of common architecture, such as componentization, integration, and process automation.

Governance of Services

Adopting and implementing a services oriented architectural style is one element, managing these services later on is another one. Without a well-defined plan to manage these services the continuity of the Enterprise is there.



Management of Services

Enterprise need to manage the delivery of business services (such as services direct to the citizen or to improve staff productivity), and delivery of technical services (such as support of IT infrastructure). Enterprises have to achieve common understanding between the customer



IFEAD is an independent research and information exchange organization working on the future state of Enterprise Architecture.

and provider through managing service level expectations and service level delivery, and delivering and supporting desired results. Business services may be delivered direct to the customer or - in the case of e-government - to the general public on behalf of the customer. Service management also looks at the dependence that businesses and organisations have on IS/IT services to acquire and process the elements which make up many of their services. Service quality monitoring demonstrates ongoing value for money and service improvement. You will also need to make arrangements for the management of infrastructure, which may be carried out on your behalf by service providers. You must have processes in place for business continuity, to ensure that the business can continue to deliver its objectives in the event of things going wrong. In addition, there must be support for the end-users in the form of training, help desk facilities and everything they need to make good use of the services.

The interaction between customer (that is, the business, its partners and end-users such as the citizen) and provider in managing services is managed by the 'informed partner' role, providing the interface to achieve the following goals:

- gain a common understanding between customer and service provider(s) of service expectations and possible achievement
- use service quality monitors as a basis for demonstrating on-going value for money and service improvements
- manage ongoing change and the effect on relationships with partners and providers
- assure consistency in the use of IT and conformance with standards and procedures, making the user community aware of how to exploit the facilities to best effect
- preserve suitable flexibility in service arrangements, including contracts in order to proactively deal with unexpected changes and demands
- establish suitable base-lines on which to track performance relating to service delivery and service improvement
- understand and influence the factors which preserve and enhance relationships to achieve maximum business benefit
- ensure that the benefits approach appraises the full investment in business change and not just individual components such as IT
- ensure that Business Continuity plans are kept up-to-date to reflect changes and new service provision.

Requirements for Services Management

Services Management is the ability to manage both the deployed services and the elements of the services architecture as discrete resources regardless of their implementation. While services address many of the traditional problems of integrating disparate business processes and applications, deploying service-oriented applications introduces new complexities that must be managed. Some key requirements for Services Management are outlined briefly below:

- Discover the existence of all key elements of the Services architecture, including new service interfaces, and identifying their relationships to other services, the IT infrastructure, Business processes—whether or not these are defined using BPEL



IFEAD is an independent research and information exchange organization working on the future state of Enterprise Architecture.

- Monitor the services layer against Service Level Agreements. Performance and availability data collected can also be used to help define an SLA, as well as provide problem identification
- Respect and enforce management policies for the Services architectural elements
- Control the services through configuration and operations
- The Services architectural elements should surface notifications to the management application when states change or problems occur
- Service security
- Managing lifecycle of services in production, facilitating the graceful introduction and deprecation of services including versioning capabilities
- Root cause analysis and correcting problems based on errors at the services layer
- Transforming and/or routing messages based on message content and context, as well as policy (usually in support of SLA or version management)
- Tracking the business impact of services on the business processes that the Web Services support
- Managing the IT infrastructure that supports the services, and associating problems in the infrastructure with manifestation of the problems at the service layer

Testing of Services

Testing the security, compliance, and reliability of (web) services is paramount to successful Service Oriented implementations, particularly those that are externally facing and business critical. Enterprises need to be able to certify and approve services for business and IT standards and deployment readiness. Services need to scale for availability, reliability, integrity, reusability and overall quality. These requirements create the need for a new Service Oriented-specific testing paradigm and test tools that can test the complex layers of a (web) services transaction and guide the control and quality of an enterprise's SOE/SOA.

Services impose specialized testing challenges for Service Oriented implementations. Testing tools must be able to inject realistic usage scenarios onto the infrastructure and validate interoperability as well as transaction completeness for transactions spanning multiple services and involving multiple messaging protocols. Testing must address verification of a service's functionality as well as its performance, scalability and security. With the highly interactive and interdependent nature of SOA based applications even small changes can cause major disruptions. Reuse, service access and service availability are fundamental to achieving a robust service oriented architecture, promoting the additional need for automated regression testing as an integral element of the test process.

Traditional testing methods include some form of simple Unit Testing, often performed by developers. These tests are designed with a knowledge of the software internals, and are nearly always aimed at testing a very small and specific part of the product. These kinds of tests are well-suited to simple Web services which have little or no interaction with other code components.

Functional Verification is a testing process in which designers, who have limited knowledge of the product source code, identify the core functionality of a product or service. Tests are designed to prove this core function conforms to the specification. For example, does my online auction display the correct bid entered? Does my insurance broker system find the cheapest quote? If these tests fail, a fundamental problem with the product has usually been



IFEAD is an independent research and information exchange organization working on the future state of Enterprise Architecture.

detected (and usually a problem that is straightforward to fix). Again these tests are suited to simple Web services, allowing you to check whether a service performs its individual function correctly.

System Test usually occurs after the functional verification stage is complete, which is after the core function has been verified. It is intended to find problems with the entire system as a whole -- to see how Web services behave as part of a system and how they interact with each other. Since the system test phase occurs near the end of a development life cycle, there is often a lack of time allocated for its completion. Due to tight release schedules and slipping of development milestones, the stages of system test are often overlooked, and the unique bugs that each uncovers too often go undetected. Even if such bugs are found, it is often too late to determine their cause and attempt to fix them. It is therefore imperative that system test applications are designed to be as efficient as possible in finding code defects. System test usually comprises of three areas. These are :

- **Performance:** It involves the process of determining the relevant product statistics. For example: How many messages per second? How many simultaneous users of a service are acceptable?
- **Scenario:** It is the process of recreating an exact configuration that a customer requires. Any problems found in the scenario can therefore be detected before the customer uses the product.
- **Stress (or workload balancing):** It is different from the other two areas in that it is designed to strain the software by applying a large workload effort. If carried out effectively, by maintaining a highly strenuous usage of the product (but not beyond the limits determined by the performance statistics), stress testing often uncovers many obscure bugs that any of the other techniques mentioned above will not find (it is also often the case that they will be the most difficult to fix).

Arguably the most efficient of the three system test components, in terms of detecting code defects, is the area of stress testing. However, too often the process is confused with other elements of system or functional testing, and the methods involved in the process are not approached or implemented correctly.

Security & Reliability of Services

Companies and governments are increasingly dependent on open source infrastructures and related services. These services are delivered through a network of professional hobbyists and commercial software and service providers. The growing dependency of users on open source services increases the importance to assure the reliability and continuity of these services. However, open source communities have characteristics that lack or even conflict with the mechanisms that are described in theory to assure reliability and continuity.

In open, heterogeneous information systems, information assurance, security, trust and privacy are issues of central concern. In an architecture promoting dynamic service discovery and semantic interoperability, these issues become more critical and more technically challenging. The richer communications model supported on the semantic web also provides



IFEAD is an independent research and information exchange organization working on the future state of Enterprise Architecture.

opportunities for new kinds of solutions to these problems. We see requirements arising from three kinds of policies.

- **Security policies** control the access to services or service capabilities.
- **Privacy policies** control the use of information provided by service clients.
- **Trust-based policies** for security or privacy may be provided when control of service capabilities can be based on reasoning about factors such as reputation or verifiable semantic relationships between agents, rather than explicit authority relationships.

Functional Requirements

- **Semantically described security policies.** Explicit representations of constraints and rules governing agent or system behaviors. Policies can define permissions, obligations, norms and preferences for an agent's actions and interactions with other agents and programs. Explicit policies, especially those expressed in high level declarative languages, can be adjusted or disseminated by communications, used as the basis for electronic contracts and addressed in negotiations for service agreements and commitments.
- **Semantically described privacy policies.** Individuals accessing a service online, are concerned about the subsequent use of data they disclose to the service, in addition to considering whether the service can provide the product and quality they demand. Service users must understand the privacy policy of the services they are accessing, and trust that the policy will be enforced. Privacy issues do not concern individual consumers only. These concerns also permeate organizational interactions, such as medical supply chains, where an increasing amount of regulatory legislation governs the conditions of patient data and other relevant disclosures. Programming this complicated web of policies into software systems is an increasingly expensive and time-consuming task that could benefit from increased automation.
- **Honoring individual client requirements.** Service users may publish or provide to services their own security or privacy requirements and preferences. They may require or negotiate with service providers to adapt to them. Services will need to state whether they can dynamically accept and manage such individual policy requirements.

Service Oriented Architectural requirements

Protocols for publication of service security policies and authentication requirements. Service providers will have to semantically describe and publish their security, privacy and trust practices and policies in conjunction with their service models.

- These descriptions will be available as additional criteria that can be used by service seekers (or their proxies) to choose among or rank the services available.
- **Semantic policy evaluation mechanisms.** Inference or proof checking mechanisms may be used to determine policy applicability. For broadly described policies, inferences may be used to evaluate group membership questions and deconflict among overlapping policy requirements. Proof checking may be used when access authority or permission proofs are provided by potential clients. These policy evaluation mechanisms will require protocols to enlist architectural assistance to enforce the



IFEAD is an independent research and information exchange organization working on the future state of Enterprise Architecture.

policies and to collect data needed to recognize existing or potential security and privacy violations. They will need the means to adapt to dynamically communicated policy revisions, and stated exceptions to policies.

- **Semantically controlled policy enforcement.** Flexible enforcement mechanisms for semantically described policies must interpret more flexible descriptions of enforcement criteria and efficiently determine case-by-case policy applicability using their local means of enforcing those policies at the underlying services architecture level. Enforcement mechanisms will also need means to adapt to dynamically changed policies.
- **Trust-based authentication and authorization.** Security and privacy policies can be based on verifiable relationships among agents. To base policy enforcement decisions on this kind of information requires additional reasoning about these trust relationships, in part based on evidence that an authentication authority can provide. For example, proof of key attributes, signed statements from a trusted source delegating a permission, or an agreement to undertake an obligation in return for access could be considered. As in human societies, authorization can be shared based on these verifiable relationships, rather than explicit delegation or group membership.

Communication and logging of security evaluation results is needed in order to support remediation services and enable the functioning of reputation services.

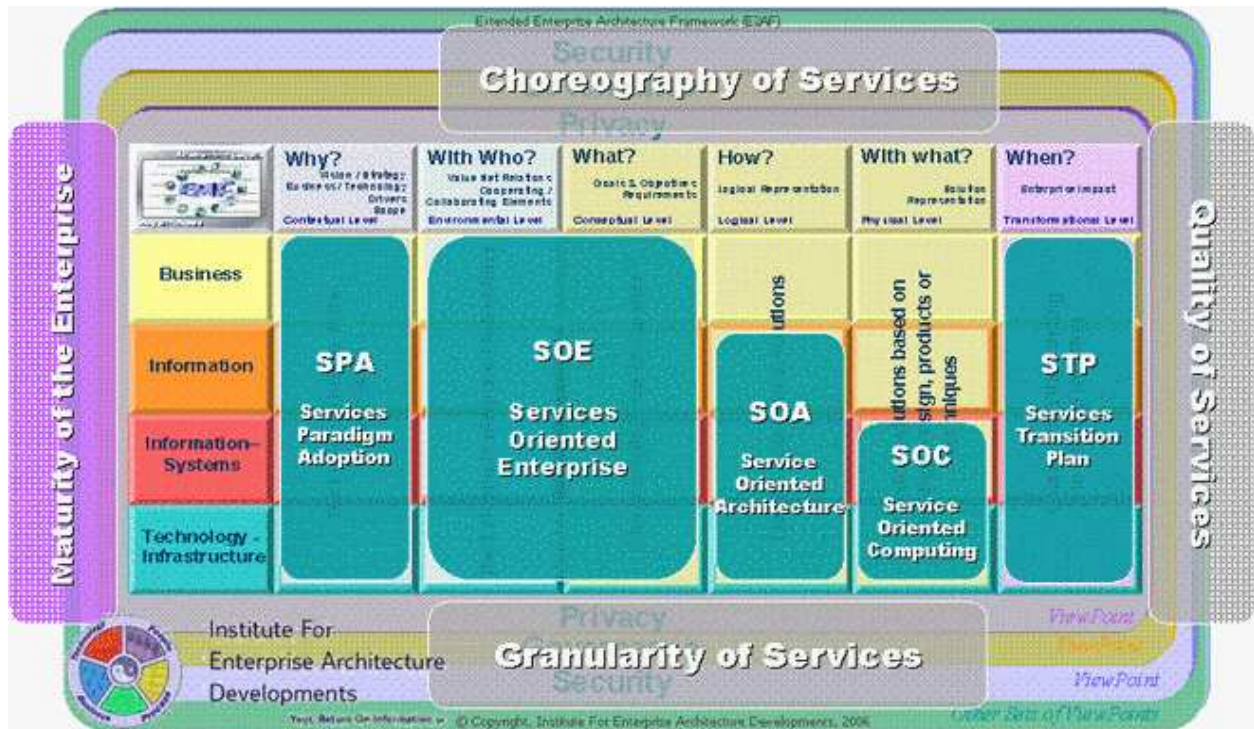
4 Critical Success Factors (CSF's) In Services Orientation, Adaptation & Implementation

Services Oriented Maturity

Companies are at different levels of maturity in the adoption and incorporation of Services-Oriented (SO). Some are just beginning to explore the world of SO using its technology instantiation: Web services. They are wrapping legacy functionality and exposing it for invocation for third-parties, clients, and business partners. This gets them into the game: they ramp up the development team, start the process of changing the corporate culture to better support SO, and take the first steps in the exploration of new technologies and the business capabilities that may be impacted.



IFEAD is an independent research and information exchange organization working on the future state of Enterprise Architecture.



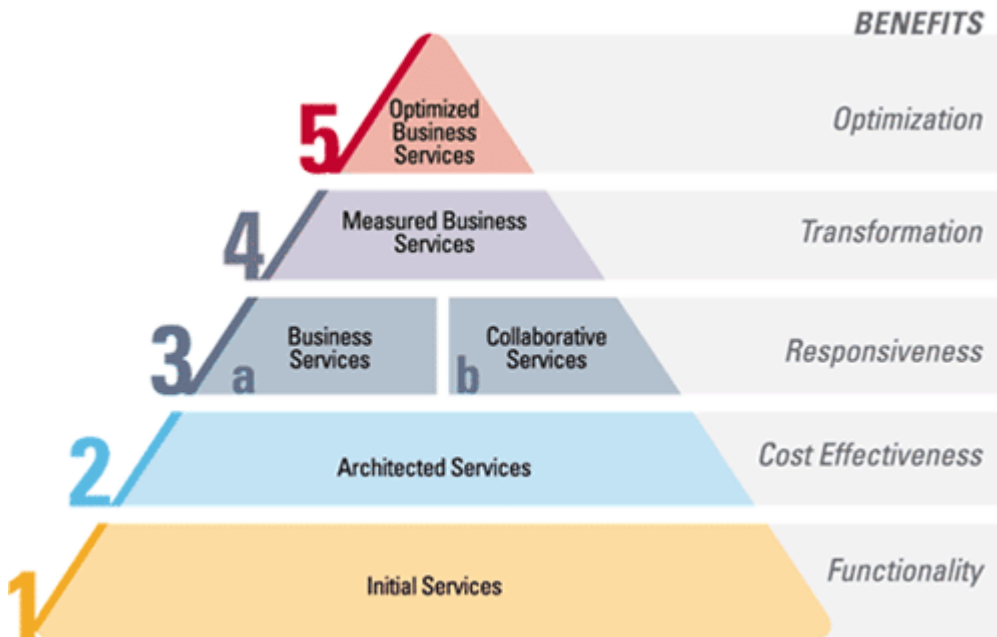
Another level of SO adoption is when the initial testing of Web services has been successfully overcome and now the organization is beginning to integrate systems and applications using services. As proprietary protocols, glue code, and point-to-point connections give way to more open, standards-based protocols and interaction based on service descriptions that each system externalizes, we step into the realm of Service-Oriented Integration (SOI). In this world, the enterprise service bus reigns supreme: a mechanism for mediation, routing, and transformation of service invocations irrespective of the target service provider. It helps overcome many of the shortcomings associated with point-to-point connections.

Service Oriented Maturity Model

This SOA Maturity Model provides a framework for discussion between IT and business users about the applicability and benefits of SOA in an organization across five levels of adoption maturity.



IFEAD is an independent research and information exchange organization working on the future state of Enterprise Architecture.

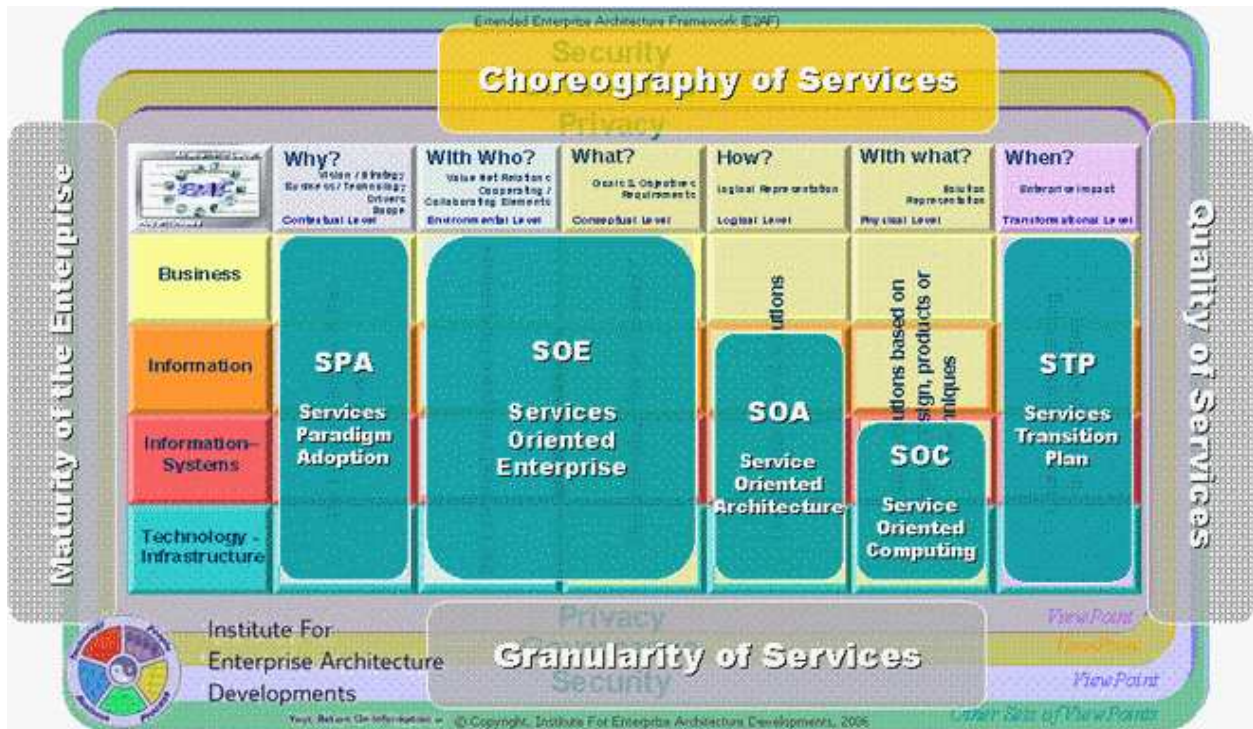


Choreography of Services

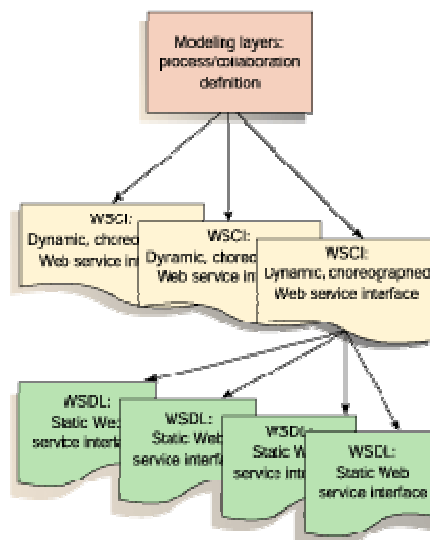
With the growing popularity of Service Oriented Architecture (SOA) and Web Services, these diverse assets can be made available as individual enterprise services. How do we build and expose them as such and how do we leverage them in building new service-based applications or business processes? This is the problem Business Process Choreography tries to solve.



IFEAD is an independent research and information exchange organization working on the future state of Enterprise Architecture.



Web Services Choreography Interface (WSCI) is an XML-based interface description language that describes the flow of messages exchanged by a Web Service participating in choreographed interactions with other services. WSCI describes the dynamic interface of the Web Service participating in a given message exchange by means of reusing the operations defined for a static interface. WSCI describes the observable behavior of a Web Service. This is expressed in terms of temporal and logical dependencies among the exchanged messages, featuring sequencing rules, correlation, exception handling, and transactions. WSCI also describes the collective message exchange among interacting Web Services, thus providing a global, message-oriented view of the interactions.

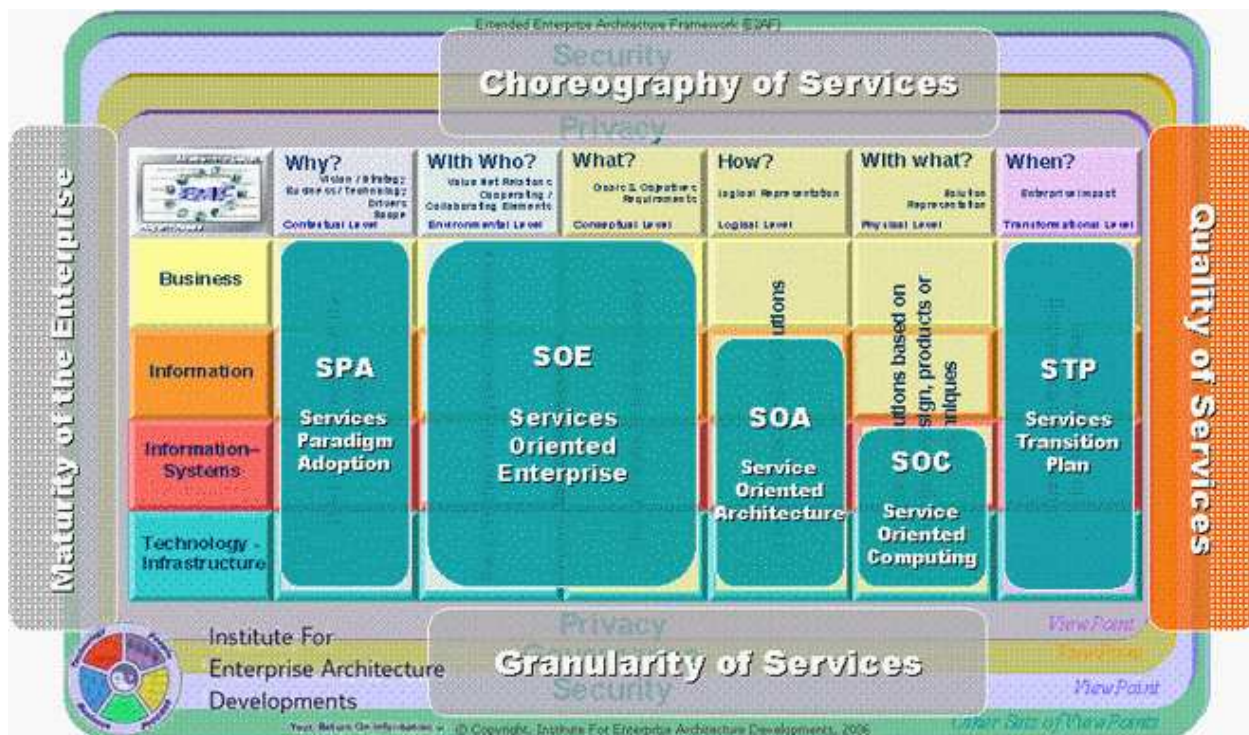




IFEAD is an independent research and information exchange organization working on the future state of Enterprise Architecture.

Quality of Services

One important missing requirement often seen in the context of Services Orientation is the management of Quality of Service (QoS). Organizations operating in modern markets, such as e-commerce activities and distributed Web services interactions, require QoS management. Appropriate control of quality leads to the creation of quality products and services; these, in turn, fulfill customer expectations and achieve customer satisfaction.



While QoS has been a major concern in the areas of networking (Cruz 1995; Georgiadis, Guerin et al. 1996), real-time applications (Clark, Shenker et al. 1992) and middleware (Zinky, Bakken et al. 1997; Frolund and Koistinen 1998; Hiltunen, Schlichting et al. 2000), few research groups have concentrated their efforts on enhancing service oriented systems to support Quality of Service management. Most of the research carried out to extend the functionality of service oriented systems, QoS has only been done in the time dimension, which is only one of the dimensions under the QoS umbrella. Furthermore, the solutions and technologies presented are still preliminary and limited (Eder, Panagos et al. 1999). The industry has a major interest on the QoS of service orientation and service oriented systems.

For organizations, being able to characterize Services, based on QoS, has four distinct advantages.

1. **QoS-based design.** It allows organizations to translate their vision into their business processes more efficiently, since services can be designed according to QoS metrics.



IFEAD is an independent research and information exchange organization working on the future state of Enterprise Architecture.

For e-commerce processes it is important to know the QoS an application will exhibit before making the service available to its customers.

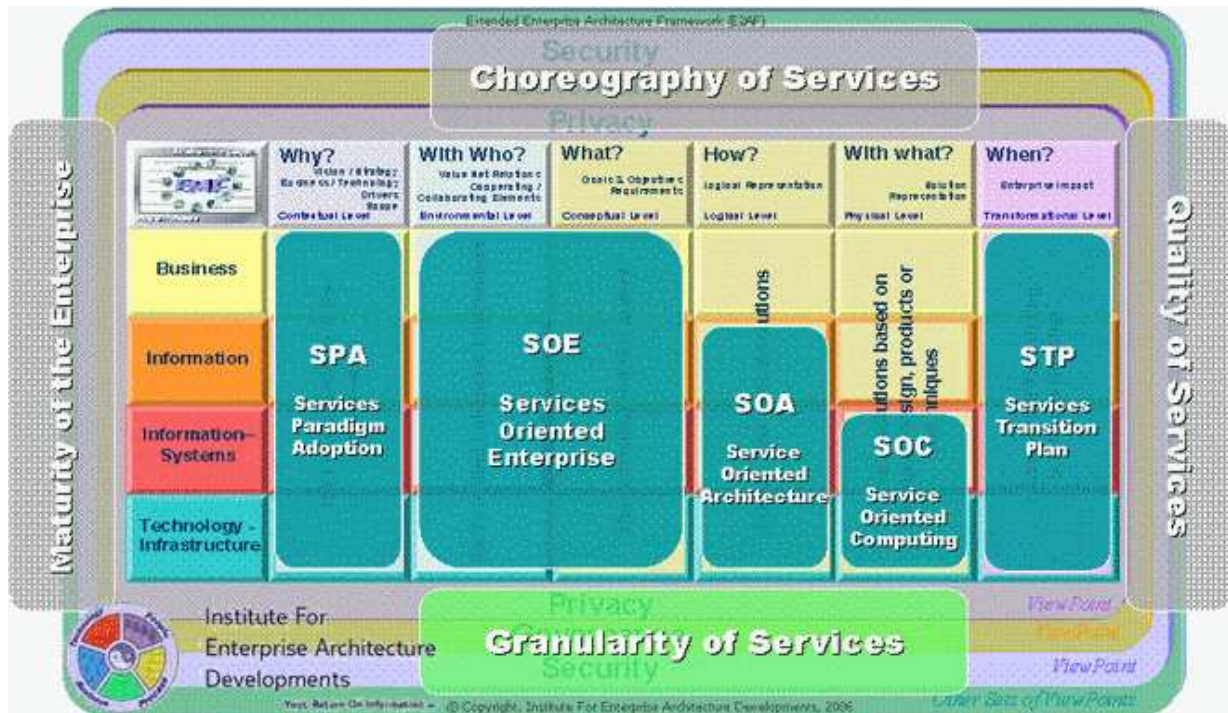
2. **QoS-based selection and execution.** It allows for the selection and execution of services based on their QoS, to better fulfill customer expectations. As service oriented systems carry out more complex and mission-critical applications, QoS analysis serves to ensure that each application meets user requirements.
3. **QoS monitoring.** It makes possible the monitoring of services based on QoS. Services must be rigorously and constantly monitored throughout their life cycles to assure compliance both with initial QoS requirements and targeted objectives. QoS monitoring allows adaptation strategies to be triggered when undesired metrics are identified or when threshold values are reached.
4. **QoS-based adaptation.** It allows for the evaluation of alternative strategies when service orientation adaptation becomes necessary. In order to complete a service according to initial QoS requirements, it is necessary to expect to adapt, replan, and reschedule a service in response to unexpected progress, delays, or technical conditions. When adaptation is necessary, a set of potential alternatives is generated, with the objective of changing a service as its QoS continues to meet initial requirements. For each alternative, prior to actually carrying out the adaptation in a running services environment, it is necessary to estimate its impact on the services QoS.

Granularity of Services

Stating that Services are the central part of both SOA and SOE will probably not offend anybody. But methods to identify Services are only slowly emerging. When talking about SOA the Services are often seen as a task that will require only a small effort – so small that we almost don't bother talking about it. The level of detail, in which this task is described, is usually focused on whether to approach this Top-Down or Bottom-up. In practice you will probably use a mix of both, but should have a strong emphasis on the Top-Down – this is necessary in order to control the coherence between the Services.



IFEAD is an independent research and information exchange organization working on the future state of Enterprise Architecture.



But how do we actually identify our Services? This is now doubt a discipline normally mastered on the abstraction level of EA. In SOE we need a method to identify Services in a way that supports the paradigm of SOE and utilizes the experience of EA. The important notion here is that we don't start from scratch!

If you are experienced working with SOE, SOAA and EA you should however not expect any revolution – it is in fact a very simple method. But this is just where I see the strength of the method. The simple message is: when identifying your Services don't start developing your solution! Keep on track, and identify the Services one level of abstraction at a time.