**Value of SOA**

Moving to a new state…

On demand business:-

*An enterprise whose business processes — integrated end-to-end across the company and with key partners, suppliers and customers — can respond with flexibility and speed to   
any customer demand, market opportunity or external threat.*

Today’s Reality: Functional Automation

Moving beyond Today’s Reality

**Historical limitations:**

* Monolithic applications can’t be reused
* Ad hoc integration creates connections that are difficult to change/maintain
* Lack of standards limits ability to deliver meaningful interoperability

**What are the barriers to business flexibility and reuse?**

* Lack of business process standards
* Architectural policy limited
* Point application buys to support redundant LOB needs
* Infrastructure built with no roadmap

**Key terms for SOA**

* A *service* is representative of a repeatable business task. Services are used to encapsulate the functional units of an application by providing an interface that is well defined and implementation independent. Services can be invoked (consumed) by other services or client applications.
* *Service orientation* defines a method of integrating business applications and processes as linked services. we’re talking about a thought process and a business philosophy not about technology.
* **SOA Defined**
* From a **business perspective**, SOA defines a set of business services composed to capture the business design that the enterprise wants to expose internally, as well as its customers and partners.
* From an **architecture perspective**, SOA is an architectural style that supports service orientation. Architecture is an investment in process, technology, and interface standard for the purpose of improving the organization’s capabilities, maximizing business agility, or reducing the cost of IT development and operations
* From an **operational perspective**, SOA includes a set of agreements between service consumers and providers that specify the quality of service, as well as reporting on the key business and IT metrics.

**Achieving Business Agility** :

* Agile means able to move or change direction easily and quickly
* As the business transforms, the IT systems implementing transformations have to be agile enough to change quickly and cost effectively while still performing current business function.
* SOA provides business agility in three ways:
* ***Loosely coupled*** *services* are ones that no longer require the same technological implementation at each end of the connection. A simple mechanism connects applications regardless of the devices & location.
* ***Reuse*** The reuse of software, hardware, processes, code, services, and infrastructure provides some of the most measurable factors for an SOA return on investment (ROI) calculation
* ***Extensibility***is defined as the ability to easily expand internal operations with new functions and to easily access organizations outside the enterprise.

Digital Model of Business with SOA - **Randy Heffner**

* **Business is done via business processes**.
* **Business processes tie to business measurements**.
* **Business process steps are delivered via IT applications.**
* **Business process steps align with business services**.
* **Business services create a digital model of the business**.
* This is powerful for two reasons –
  + As the business changes, that is, as you create new process steps and optimize existing steps by changing who does what, when many of the underlying services are stable and only the method of access needs to change.
  + The digital model provides a structure to collect, examine, and align business and IT metrics. Also provides correlation between IT costs and business process results.

**Characteristics of a Service:**

* Services invoked through defined communication protocols
* Stress on interoperability and location transparency
* **Appear as a self-contained function**
* Use a **well-defined** interface
  + - expose business functions
    - **hide underlying implementation** details
      * Loosely coupled
    - independent of any particular technology
* Services are not dependent on the context or state of other services.
  + - Any dependencies between services are defined in terms of common business process, function and data models

Align Business with IT using SOA - **Randy Heffner**

* Business services push IT to understand business processes.
* Which leads IT to examine business process metrics.
* Which leads IT to understanding the business in business terms.
* Giving IT a new way to prove its value: business results measurement.
* SOA can reduce or eliminate IT frustrations and provide a way to quantifiably measure IT business value.

**SOA Benefits**

* Saves money, time, and people
* Eliminates frustrations with IT
* Justifies IT investments
* Provides business executives with a clear understanding of what IT does and its value
* Eliminates IT’s 6-6 answer (that is, the project will take 6 months and cost 6 figures)
* Provides a business and competitive differentiator

When a change in business process no longer requires a change to application programming logic, you have a successful SOA; your company has attained competitive business agility.

Agile = able to move & change direction quickly and easily

**What will happen if you don’t adopt it?**

An SOA could be the difference between the success and failure of the next:

* Department, intra-company, or inter-company merger
* Acquisition
* Divestiture
* Product or service rollout
* Business partner, customer, or supplier addition
* Geographical expansion
* Competitive onslaught

**When not to implement SOA?**

* When you have a homogeneous IT environment
* When true real-time performance (nanoseconds response times) is absolutely critical
* When flexibility is not needed
* When tight coupling is needed
* If the organization isn’t ready for it

**Elements of SOA**



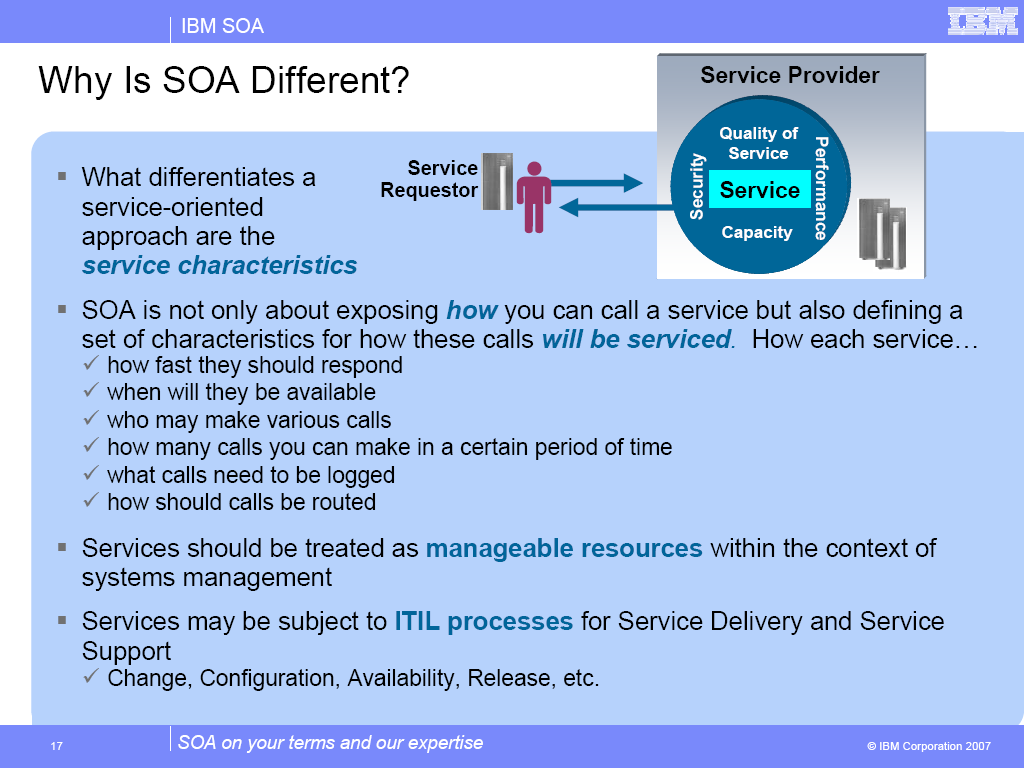
1. A *service* is a unit of work done by a service provider to achieve desired end results for a service consumer.
2. Service provider systems are the providers of *services* accessible by well-defined and published interfaces.
3. Service requester systems are the consumer of *services* accessible by well-defined and published interfaces.
4. *Service directory* is a well-known *directory* of available *services*.
5. Services are created and published by service providers and are made available on a suitable infrastructure for access by the service consumers.
6. Service descriptions are created by the service provider and are published to the service registry for access by the service consumer.
7. Security and management are part of the overall SOA framework.

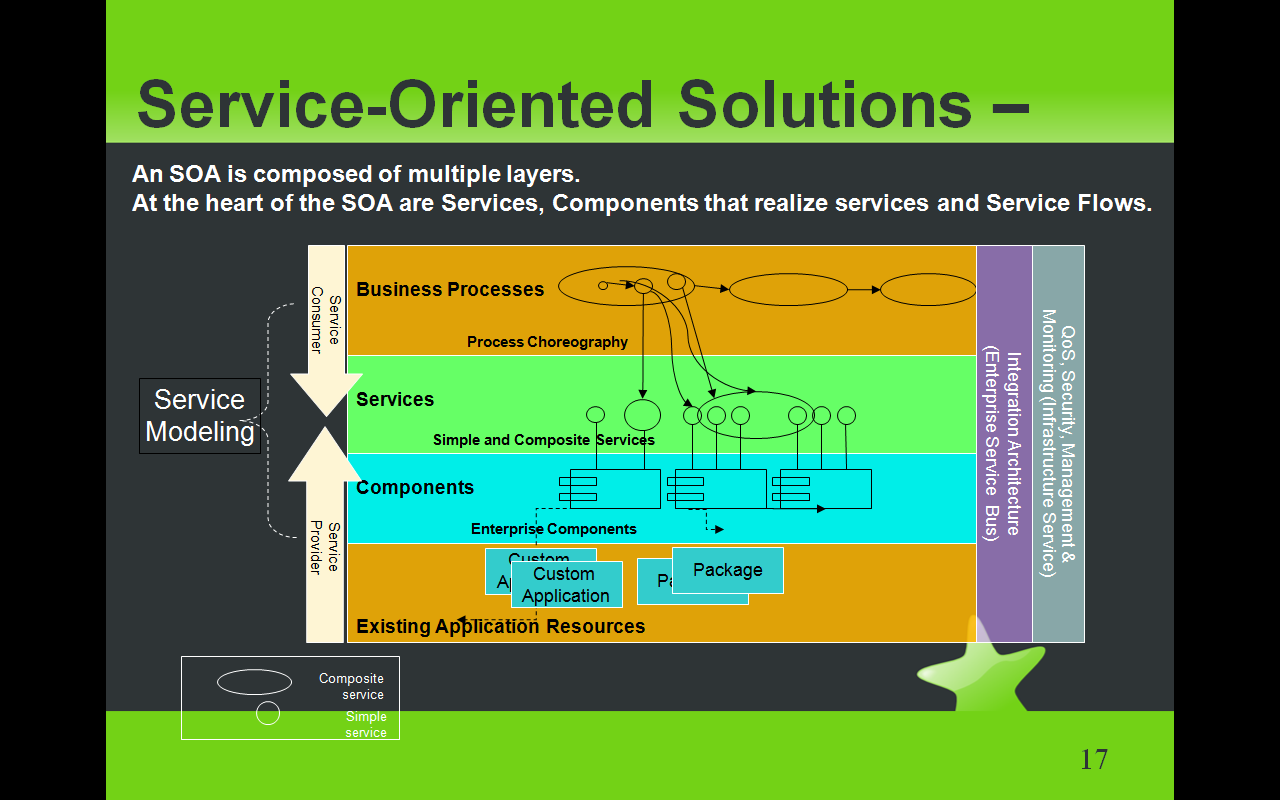
**Key principles of the SOA**

* **Loose coupling** - Services maintain a relationship that minimizes dependencies and only requires that they retain an awareness of each other.
* **Service contract** - Services adhere to a communications agreement, as defined collectively by one or more service descriptions and related documents.
* **Autonomy** - Services have control over the logic they encapsulate.
* **Abstraction** - Beyond what is described in the service contract, services hide logic from the outside world.
* **Reusability** - Logic is divided into services with the intention of promoting reuse.
* **Composability** - Collections of services can be coordinated and assembled to form composite services.
* **Statelessness** - Services minimize retaining information specific to an activity.
* **Discoverability** - Services are designed to be outwardly descriptive so that they can be found and assessed via available discovery mechanisms.

It is important to know that Web services is not *the* SOA, but rather, a technology that follows the SOA style of design. SOA and its underlying principles have existed longer than Web services. Only recently has the Web services SOA gained so much attention. This is due in part to the significant industry support for Web services from organizations like IBM, Microsoft, BEA, Oracle, Hewlett Packard, and others.

* Although stated earlier, it is worth repeating that the proliferation of Web services, or rather a large-scale
* deployment of Web services, does not move an organization closer to an SOA. SOA requires planning
* and introduction of some new processes and technologies over time in the area of services management.
* It also requires an understanding of SOA and what constitutes a service.





The relationships between processes, services, and components are described by looking at each item.

These relationships are illustrated below. Building systems using heterogeneous network-addressable

**software components is what SOA is about.**

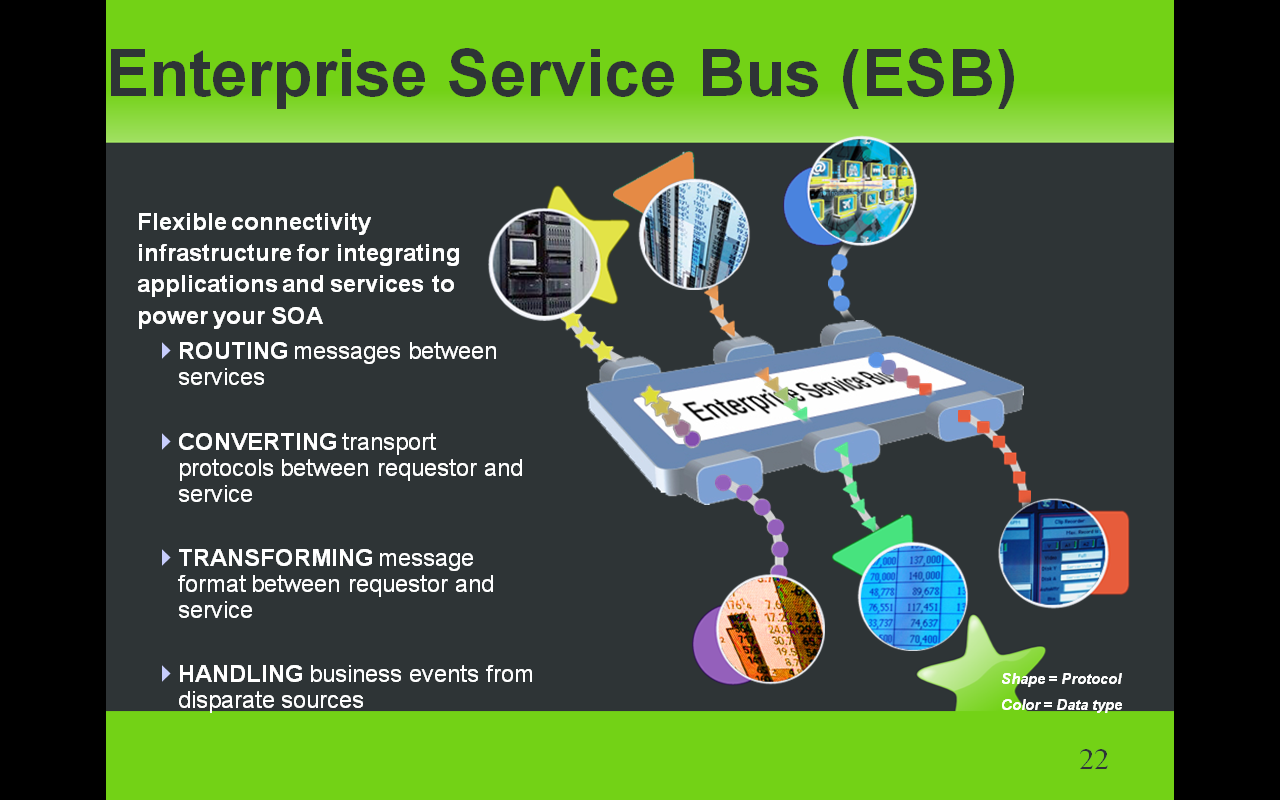
1. Business processes get business value from *things*, such as software assets, for example, applications.
2. Services describe what things do but represent a fundamental shift in thinking from the traditional approach of creating *things*.
3. Components are what *things are*, not what *things do*.
4. Deployment units or software artifacts are how *things* are constructed and deployed.

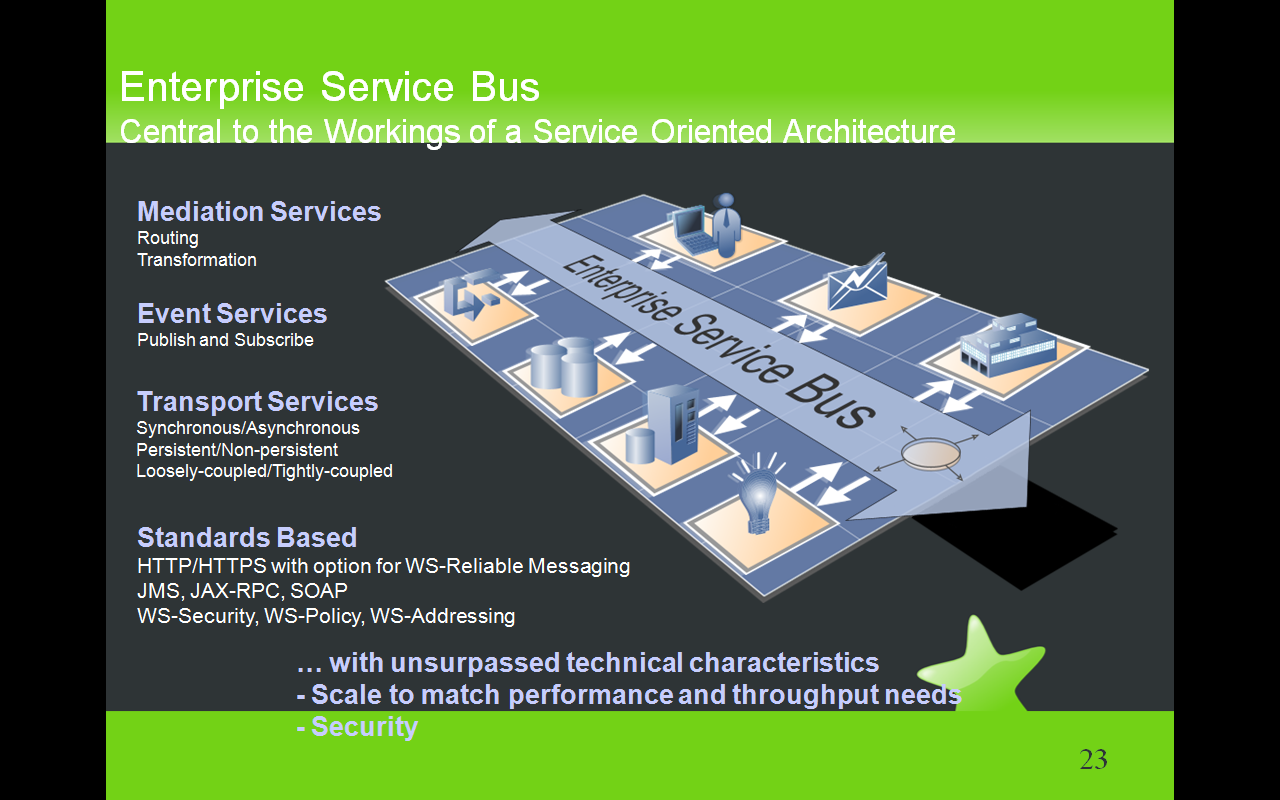
**Slide Objectives:** This slide focuses on the services perspective, showing how a SOA-based solution is realized within the architecture.

**Details:** · The key aspects of this perspective is to show how the key principles of SOA are realized within the Reference Architecture so as to achieve the necessary alignment of IT with business and providing the flexibility within IT to respond to the agile business changes. The key points:

1. The Services layer is the specification of the services that are used by the service consumers and implemented by the service providers. This layer directly supports the principle of “programming to an implementation not to an interface.” As shown in the reference, atomic services can be exposed directly to consumers, or exposed via a business process; and implemented via service components or directly (via a service façade) through the operational systems. The consumer of the service does NOT know how the services are implemented, which facilitates great flexibility in changing implementations without impacting the consumers.
2. The services layer also shows the ability to compose services to achieve increasing levels of behavior.
3. The business process layers shows the ability to separate collaboration/control behavior and logic from the underlying business behavior and logic that is provided by the services themselves. This approach (especially when defined “declaratively”) supports changing of behavior without the need to re-code the underlying logic.
4. The reference architecture directly handles separation of concerns by providing separate “backplanes” that support integration (the ESB), qualities of services, data architecture and governance.

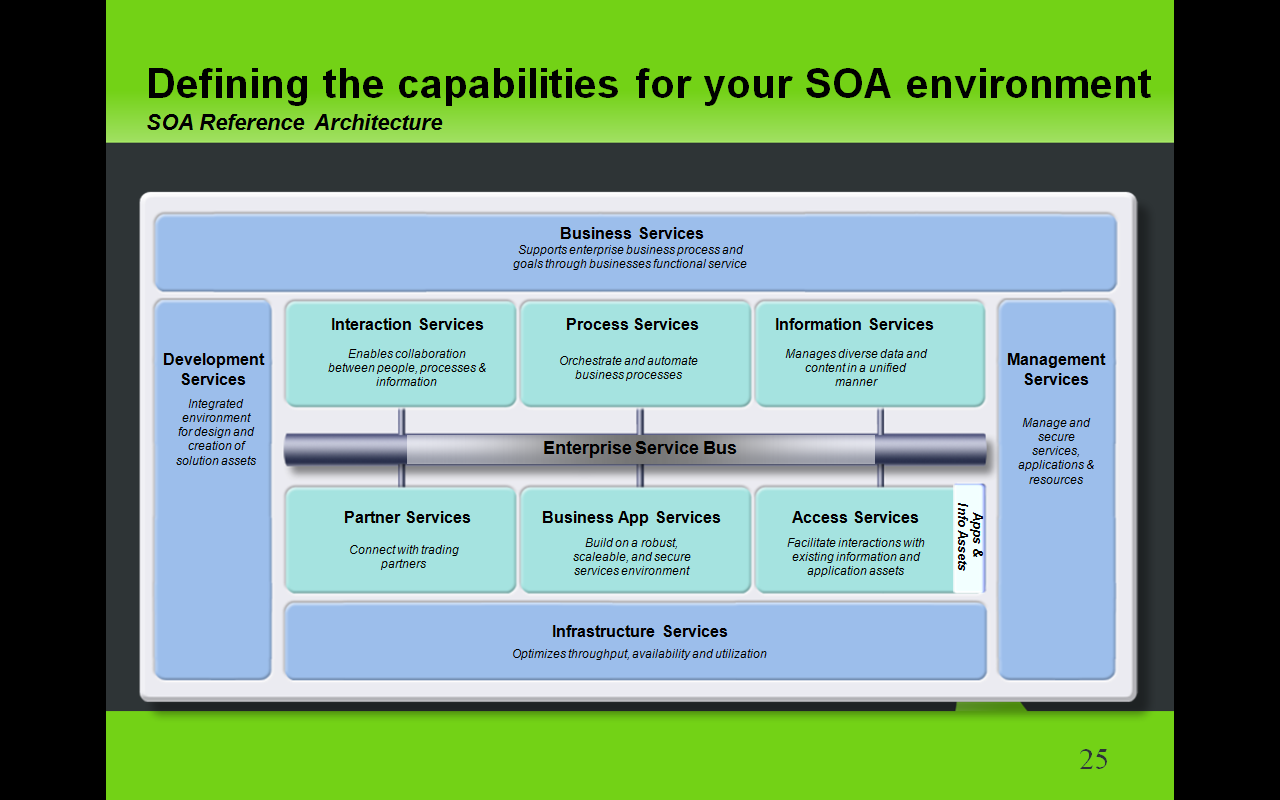
**ESB (enterprise service bus)**





The Enterprise Service Bus and the Service Oriented Architecture are closely linked. The ESB provides the connectivity infrastructure for a Service Oriented Architecture. This places the ESB in context as it’s reason for being is to support an open, standards based, service oriented architecture. As shown on the picture, the SOA provides the ‘users’ of the ESB – these are the orange/pink boxes on the diagram. Each of the interactions with the ESB (the white arrows) ideally makes use of a WSDL based service definition, invoking the required transport services and quality of service. This will frequently make use of a capability like WSIF to allow a standard service based interaction to invoke different implementation types.

Note that the text on the left and at the bottom of the chart describes the characteristics of the ESB, not the Service Oriented Architecture.



**\*\*Main point: The SOA reference architecture is a vendor-neutral way of looking at and planning the set of services that go into building an SOA. It's a visual tool for laying out projects at a service level.**

Let's take a look at how some of these companies are looking at the more technical, architectural ways of laying out their SOA projects. It's called the SOA Reference Architecture

The SOA reference architecture is a way of looking at the set of services that go into building an SOA. This architecture is **not unique to IBM**; these are things that you need to consider when approaching SOA **regardless of what products and services** are used. These capabilities can be implemented on a **build-as-you-go basis** allowing capabilities and project level solutions to be easily added as new requirements are addressed over time. You can see that these services organized **along the same lifecycle** we’ve discussed. On the left in is Development Services which is model and assemble, in the middle are the elements of the deployment run-time environment you use and on the right is management. The backbone of the reference architecture is the enterprise service bus which facilitates communication between services. The reference architecture is a great tool for laying out roadmaps for pursuing SOA. Regardless of what kind of project you’re undertaking, it makes sense to lay it out on a reference architecture to see how the various services you’re designing are going to interact with each other

**Additional detail:**

The SOA Reference Architecture outlines the key capabilities that are required for comprehensive, enterprise wide SOA solutions. These capabilities can be implemented on a build-as-you-go basis allowing capabilities and project level solutions to be easily added as new requirements are addressed over time.

Toolsare an essential component of any comprehensive integration architecture. The SOA Architecture includes both **Development Services** which are used to implement custom artifacts that leverage the infrastructure capabilities, and **Business Innovation & Optimization Services** which are used to monitor and manage the runtime implementations at both the IT and business process levels.

At the core of the SOA Reference Architecture is the **Enterprise Service Bus**. This delivers all of the inter-connectivity capabilities required to leverage the services implemented across the entire architecture. Transport services, event services, and mediation services are all provided through the ESB.

The SOA Reference Architecture also contains a set of services that are oriented toward the integration of people, processes, and information:

**Interaction Services** provide the capabilities required to deliver IT functions and data to end users, meeting the end-user's specific usage preferences.

**Process Services** provide the control services required to manage the flow and interactions of multiple services in ways that implement business processes.

**Information Services** provide the capabilities required to federate, replicate, and transform data sources that may be implemented in a variety of ways.

Many of the services in an SOA are provided through existing applications; others are provided in newly implemented components; and others are provided through external connections to third party systems.

Existing enterprise applications and enterprise data are accessible from the ESB through a set of **Access Services** that provide the bridging capabilities between legacy applications, pre-packaged applications, enterprise data stores and the ESB.

The SOA Reference Architecture also contains a set of **Partner Services** that provide the document, protocol, and partner management capabilities required for business processes that involve inter-actions with outside partners and suppliers.

**Business Application Services** provide runtime services required for new application components to be included in the integrated system.

Underlying all these capabilities of the SOA Reference Architecture is a set of **Infrastructure Services** which are used to optimize throughput, availability and performance.

**IT Services Management Services** include capabilities that relate to scale and performance, for example edge services, clustering services, and virtualization capabilities allow efficient use of computing resources based on load patterns.

The SOA Reference Architecture is a complete and comprehensive architecture that covers all the integration needs of an enterprise. Its services are well integrated and are delivered in a modular way, allowing SOA implementations to start at a small project level. As each additional project is addressed, new functions can be easily added, incrementally enhancing the scope of integration across the enterprise.

**Background:**

The IBM SOA Foundation delivers the capabilities you need to adopt SOA through a comprehensive architecture. These capabilities can be implemented on a build-as-you-go basis, and yet, because of the architecture and its service orientation, capabilities and project level solutions can be easily added as new requirements are addressed over time.

The SOA Reference Architecture shows the key capabilities that are required for comprehensive, enterprise wide SOA solutions.

**Development Services** are an essential component of any comprehensive integration architecture. The SOA Architecture includes development tools, used to implement custom artifacts that leverage the infrastructure capabilities, and business performance management tools, used to monitor and manage the runtime implementations at both the IT and business process levels. Development tools allow people to efficiently complete specific tasks and create specific output based on their skills, their expertise, and their role within the enterprise. Business Analysts who analyze business process requirements need modeling tools that allow business processes to be charted and simulated. Software Architects need tool perspectives that allow them to model data, functional flows, system interactions, etc. Integration Specialists require capabilities that allow them to configure specific inter-connections in the integration solution. Programmers need tools that allow them to develop new business logic with little concern for the underlying platform. Yet, while it is important for each person to have a specific set of tool functions based on their role in the enterprise, the tooling environment must provide a framework that promotes joint development, asset management and deep collaboration among all these people. A common repository and functions common across all the developer perspectives (e.g. version control functions, project management functions, etc) are provided in the SOA Reference Architecture through a unified development platform.

The **Business Innovation & Optimization Services** incorporate monitoring capabilities that aggregate operational and process metrics in order to efficiently manage systems and processes. Managing these systems requires a set of capabilities that span the needs of IT operations professionals and business analysts who manage the business operations of the enterprise. These capabilities are delivered through a set of comprehensive services that collect and present both IT and process-level data, allowing business dashboards, administrative dashboards, and other IT level displays to be used to manage system resources and business processes. Through these displays and services, it is possible for LOB and IT personnel to collaborate to determine, for example, what business process paths may not be performing at maximum efficiency, the impact of system problems on specific processes, or the relationship of system performance to business process performance. This collaboration allows IT personnel and assets to be tied more directly to the business success of the enterprise than they traditionally have been.

One key feature of the SOA Reference Architecture is the linkage between the Development and the Business Innovation & OptimizationServices. The ability to deliver runtime data and statistics into the development environment allows analyses to be completed that drive iterative process re-engineering through a continuous business process improvement cycle.

At the core of the SOA Reference Architecture is the **Enterprise Service Bus**. This architectural construct delivers all the inter-connectivity capabilities required to leverage and use services implemented across the entire architecture. Transport services, event services, and mediation services are all provided through the ESB. Transport services provide the fundamental connection layer; event services allow the system to respond to specific stimuli that are part of a business process; and mediation services allow loose-coupling between interacting services in the system. The ESB is a key factor in enabling the service orientation of the SOA Reference Architecture to be leveraged in implementing service oriented solutions and can be implemented today to meet the quality of service requirements of any integration solution.

The SOA Reference Architecture also contains a set of services that are oriented toward the integration of people, processes, and information. These services control the flow of interactions and data among people and automated application services in ways appropriate to the realization of a business process:

- **Interaction Services** provide the capabilities required to deliver IT functions and data to end users, meeting the end-user's specific usage preferences.

- **Process Services** provide the control services required to manage the flow and interactions of multiple services in ways that implement business processes.

- **Information Services** provide the capabilities required to federate, replicate, and transform data sources that may be implemented in a variety of ways.

Automated application services, implementations of business logic in automated systems, are a critical part of any integration architecture or solution. Many of these services are provided through existing applications; others are provided in newly implemented components; and others are provided through external connections to third party systems. Existing enterprise applications and enterprise data are accessible from the ESB through a set of access services. These **Access Services** provide the bridging capabilities between legacy applications, pre-packaged applications, enterprise data stores (including relational, hierarchical and nontraditional, unstructured sources such as XML and Text), etc and the ESB. Using a consistent approach, these access services expose the data and functions of the existing enterprise applications, allowing them to be fully re-used and incorporated into functional flows that represent business processes. Existing enterprise applications and data leverage the Business Application and Data Services of their operating environments such as CICS, IMS, DB2, etc. As these applications and data implementations evolve to become more flexible participants in business processes, enhanced capabilities of their underlying operating environments, for example support of emerging standards, can be fully utilized.

The SOA Reference Architecture also contains a set of **Business Application Services** that provide runtime services required for new application components to be included in the integrated system. These application components provide new business logic required to adapt existing business processes to meet changing competitive and customer demands of the enterprise. Design and implementation of new business logic components for integration enables them to be fully re-useable, allowing them to participate in new and updated business processes over time. The Business Application Services include functions important to the traditional programmer for building maintainable, flexible, and re-useable business logic components.

In many enterprise scenarios, business processes involve inter-actions with outside partners and suppliers. Integrating the systems of the partners and suppliers with those of the enterprise improves efficiency of the overall value chain. **Partner Services** provide the document, protocol, and partner management services required for efficient implementation of business-to-business processes and inter-actions.

Underlying all these capabilities of the SOA Reference Architecture is a set of **Infrastructure Services** which provide security, directory, IT system management, and virtualization functions. The security and directory services include functions involving authentication and authorizations required for implementing, for example, single sign-on capabilities across a distributed and heterogeneous system.

**IT Services Management Services** include functions that relate to scale and performance, for example edge services and clustering services, and the virtualization capabilities allow efficient use of computing resources based on load patterns, etc. The ability to leverage grids and grid computing are also included in infrastructural services.

While many of the Infrastructure and IT Service Management services perform functions tied directly to hardware or system implementations, others provide functions that interact directly with integration services provided in other elements of the architecture through the ESB. These interactions typically involve services related to security, directory, and I/T operational systems management.

The SOA Reference Architecture is a complete and comprehensive architecture that covers all the integration needs of an enterprise. Its services are well integrated and are delivered in a modular way, allowing SOA implementations to start at a small project level. As each additional project is addressed, new functions can be easily added, incrementally enhancing the scope of integration across the enterprise. In addition to supporting SOA strategies and solutions, the architecture itself is designed using principles of service orientation and function isolation.