Service Oriented Architecture (SOA)
An Introduction
Application Evolution

Monolithic Applications
- Mainframe
- Client / Server

Distributed Applications
- DCE/RPC
- CORBA
- DCOM
- EJB

Service Oriented Applications
- Services
- Messages
- Events

Application Sophistication
- Maintainability
- Cost effectiveness
- Flexibility
- Reusability
- Business Agility

Time
What is SOA?

There are many definitions of what SOA is.....
What is SOA?

- **Gartner.** A **software architecture** that starts with an interface definition and builds the entire application topology as a topology of interfaces, interface implementations and interface calls.

- **IBM.** An **application framework** that takes everyday business applications and breaks them down into individual business functions and processes, called services.

- **Microsoft.** A world-wide **mesh of collaborating services** that are published and available for invocation on a Service Bus.

- **webMethods.** An **approach to software implementation** where systems are composed of reusable services. A service is a software building block that performs a distinct function through a well-defined interface.

- **BearingPoint.** A **software design & implementation approach** ("Architecture") of loosely coupled, coarse grained, reusable artifacts ("Services"), which can be integrated with each other, through a wide variety of platform independent service interfaces.
So? What is SOA?
An architectural framework that enables us to:

- Create reusable services that are highly interoperable through the use of broadly-supported standards.
- Support a new generation of agile composite applications assembled from business, application, and technical services.
- Create external / business partner interfaces to streamline inter-enterprise integration, replacing EDI, VANs, etc.
- Aligns business requirements with IT technology assets
SOA Myths – *What is and What is not*

- SOA is a design and architecture philosophy, not a technology
- SOA is not a product
- SOA is *not* Web Services. But, Web Services is one of the means to achieve a SOA
- SOA is a means, not an end.....

The Myths

- Services and components are the same thing.
- Any use of services is SOA.
- Use of J2EE or .NET automatically results in an SOA.
- SOA requires SOAP or, conversely, use of SOAP results in SOA.
SOA – Roles & Collaborations

- **Service Consumer**
  - Find Service
  - Maintain Service

- **Service Provider**
  - Invoke Service
  - Service Implementation

- **Service Broker**
  - Service Registry
  - Service Contract
  - Services Security
  - Services Monitoring
Anatomy of a Service

Service Interface
- Access layer between the service consumer and service provider.
- Contains:
  - Service Identity
  - Service Input & Output data information
  - Service Purpose & Function Metadata

Service Implementation
- Contains the core functional or business logic of the service.
- The implementation should be totally transparent to the service consumer, with no knowledge necessary about the implementation specifics.
SOA Architectural Characteristics

Loose Coupling
- Increases organizational agility; allows easy assembly and modification of business processes in response to market requirements
- Lowers implementation costs by increasing reusability; services can easily be shared across multiple applications
- Increases IT adaptability; changes — resulting from mergers, acquisitions, package application implementations, etc.

Standards-based Interoperability
- Platform independence allows companies to use the software and hardware of their choice
- Allows companies to engage in a multi-source strategy, reducing threat of vendor lock-in
- Reduces complexity and fragmentation resulting from use of proprietary technologies
- Lowers training requirements; increases available labor pool
SOA Architectural Characteristics – cont.

Service Interfaces
- The service interface is the integration point in a service-oriented architecture
- Service implementation transparency minimizes the impact when infrastructure and implementation changes occur
- Integration is more manageable since complexities are isolated
- Layer of abstraction enables wrapping of existing assets as services that provide business functions so the organization can continue leveraging its investment in its legacy applications

Service Composition
- Ability to create new services by combining existing deployed services
- Leveraging existing components and services reduces the time needed for development and testing of new functionality.
- Rapid development of new business services allows an organization to respond quickly to changes and reduce the time-to-market
SOA Architectural Characteristics – cont.

Modular

- Enables incremental development, deployment, and maintenance; avoids the need to do costly and risky "big bang" software implementations
- Decreases development effort by reducing complexity (through a "divide and conquer" approach)
- Over time, accelerates deployment of new application functionality; process becomes mostly assembly of existing services versus mostly new development

Mediation

- Enables the transformation, routing and securing of messages "mid flight"
- Mediation is carried out by diverse systems such as Enterprise Service Bus, Brokers, Integration hubs, Adaptors and Gateways
Value Proposition - *Business Benefits*

- Agile business and technology environment
- Reduced business, IT, and maintenance costs
- Manageable project size
- Technology autonomy and platform-agnostic systems
- Facilitates automated business process
- Evolutionary approach
Value Proposition - *Technology Benefits*

- Efficient development process
- Reduced number of interfaces and development time
- Platform-agnostic applications
- Design time discovery, introspection and usage of services
- Standards-based architectures
- Several enablers - Portals, BPM, BI
SOA – Benefits Perspective

**CEO Perspective**
- Budget Strategy
  Agile & Reactive IT environments.
- Short Term Planning
  SOA enables step-by-step approaches
- Budget Reduction
- Technology & Vendor Agnostic

**CIO Perspective**
- Independence from technology
- Positive Role of IT Department
  IT – Business bridging
- Cost Reduction
- Increase of Influence
- Manageable Project Size

**IT Architect Perspective**
- Disentanglement
- Loose Coupling
- Code Reuse

**Project Manager Perspective**
- Smaller & Shorter Projects
- Technology Independence
- Parallel Development
- Reduced Project Risk
- Easier Testing & Integration

**Developer Perspective**
- Reduction of Dependency
- Rapid Prototyping
- Better Define Requirements
- Simplified Testing

**Business Department Perspective**
- Independence from Technology
- Shorter Time to Market
- Reduction of Development Costs
SOA Conceptual Model

Portal access, with views customized by user, role and purpose

Integration Tools, Techniques and Common Interface Standards

Interaction Services | Process Services | Business Services | Information Services | Access Services

HR Data | Intermediary Data | Financial Data | Etc
Service Categories

Two broad categories of services

- Business services - reflect real world business concepts, events, process etc
- Technical services - cross domain and horizontal by nature and provide system level behavior such as security, audit and event logging, transformation services etc.
Service Functional Types

Information Services

Business Services

Process Services

Interaction Services

Access Services
Information Services

- Contain the data logic of your business design

- Provide access to the persistent data of your business.

- Can federate multiple data sources – projecting logical views over those multiple sources to render a simpler access pattern for the service composition that needs it.
Example Information Services

Service Consumer

Service Provider

Atomic Service

Wrapped Legacy

Composite Service

Service Interface

Service Implementation

Interface Proxy
Business Services

- Services that implement your core business logic.
- Service components created specifically as services within a business model and that represent the basic building blocks of your business design.
- Services that are not decomposable within the business model, but that can be composed to form higher level services.
- Often these services will be composed in business processes (such as process flows or business state machines). However, these services may also be invoked directly by presentation logic in interaction services.
Example Business Services

Service Consumer

Service Provider

Atomic Service

Wrapped Legacy

Composite Service

Note: Business Service may utilize middle tier
Process Services

- Orchestrate and automate business processes
- Business process flows
- Business state machines
- Long-running business processes that can interact with multiple partners
- Incorporation of people into processes
Example Process Services

Service Consumer

Interface Proxy

Service Interface

Service Implementation

Middle Tier

Mediation

Orchestration

Workflow

Service Provider

Atomic Service

Wrapped Legacy

Composite Service
Interaction Services

- Interaction services are about the presentation logic of the business design.
- Support the interaction between applications and end-users or devices.
- Project a view of the information system tailored to the specific interaction fidelity, frequency of interaction, and presentation composition that best fits the needs of the end user or device.
- May be tailored to role-sensitive contexts adjusting what is seen and the behavior presented to the external world based on who the user is, what role they are performing.
Example Interaction Services

Service Consumer

Service Interface

Service Implementation

Interface Proxy

Service Provider

Atomic Service

Wrapped Legacy

Composite Service
Access Services

- Access services are dedicated to integrating legacy applications and functions into the service-oriented architecture.
- This includes simple wrapping of those functions and rendering them as services.
- May also augment the logic of the existing function to better meet the needs of the business design.
- The access service may invoke multiple legacy functions to achieve the semantic requirements of the service.
Example Access Services

Service Consumer

Interface Proxy

Middle Tier

Mediation
Orchestration
Workflow

Service Provider

Wrapped Legacy

Legacy App

Wrapped Legacy

Service Consumer

Interface Proxy

Service Interface

Service Implementation

Service Consumer

Interface Proxy

Service Interface

Service Implementation

Service Provider

ERP
Criteria for Identifying Services

- Has a well defined business benefit and maps to a business context
- Supports business imperatives and addresses business hot spots (areas in clear need of improvement)
- Reusable by multiple processes, business units, developers & analysts
- Achievable in a reasonable timeframe
- Helps build towards SOA strategy, goals, objectives (agile, composable, discoverable, reduces redundancies,...)