**SOA Characteristics**

* Loosely coupled: minimizes dependencies between services.
* Contractual: adhere to agreement on service descriptions.
* Autonomous: control the business logic they encapsulate.
* Abstract: hide the business logic from the service consumers.
* Reusable: divide business logic into reusable services.
* Composable: facilitate the assembly of composite services.
* Stateless: minimize retained information specific to an activity.
* Discoverable: self-described so that they can be found and assessed.

**Potential Benefits**

* Based on open standards.
* Supports vendor diversity.
* Fosters intrinsic interoperability.
* Promotes discovery.
* Promotes federation.
* Fosters inherent reusability.
* Emphasizes extensibility.
* Promotes organizational agility.
* Supports incremental implementation.

Technical architecture that adheres to and supports the principles of service orientation.

**What is Service-Oriented Architecture?**

* Service-Oriented Architecture (SOA) is an architectural style. Applications built using an SOA style deliver functionality as services that can be used or reused when building applications or integrating within the enterprise or trading partners.
* Uses open standards to integrate software assets as services
* Standardizes interactions of services
* Services become building blocks that form business flows
* Services can be reused by other applications

**What is a Service?**

* A service is a reusable component that can be used as a building block to form larger, more complex business-application functionality.
* A service may be as simple as “get me some person data,” or as complex as “process a disbursement.”
* A service provides a discrete business function that operates on data. Its job is to ensure that the business functionality is applied consistently, returns predictable results, and operates within the quality of service required.
* How the service is implemented, and how a user of the service accesses it, are limited only by the SOA infrastructure choices of the enterprise.
* From a theory point of view, it really doesn’t matter how a service is implemented.

**Characteristics of a Service**

* Supports open standards for integration: Although proprietary integration mechanisms may be offered by the SOA infrastructure, SOA’s should be based on open standards. Open standards ensure the broadest integration compatibility opportunities.
* Loose coupling: The consumer of the service is required to provide only the stated data on the interface definition, and to expect only the specified results on the interface definition. The service is capable of handling all processing (including exception processing).
* Stateless: The service does not maintain state between invocations. It takes the parameters provided, performs the defined function, and returns the expected result. If a transaction is involved, the transaction is committed and the data is saved to the database.
* Location agnostic: Users of the service do not need to worry about the implementation details for accessing the service. The SOA infrastructure will provide standardized access mechanisms with service-level agreements.