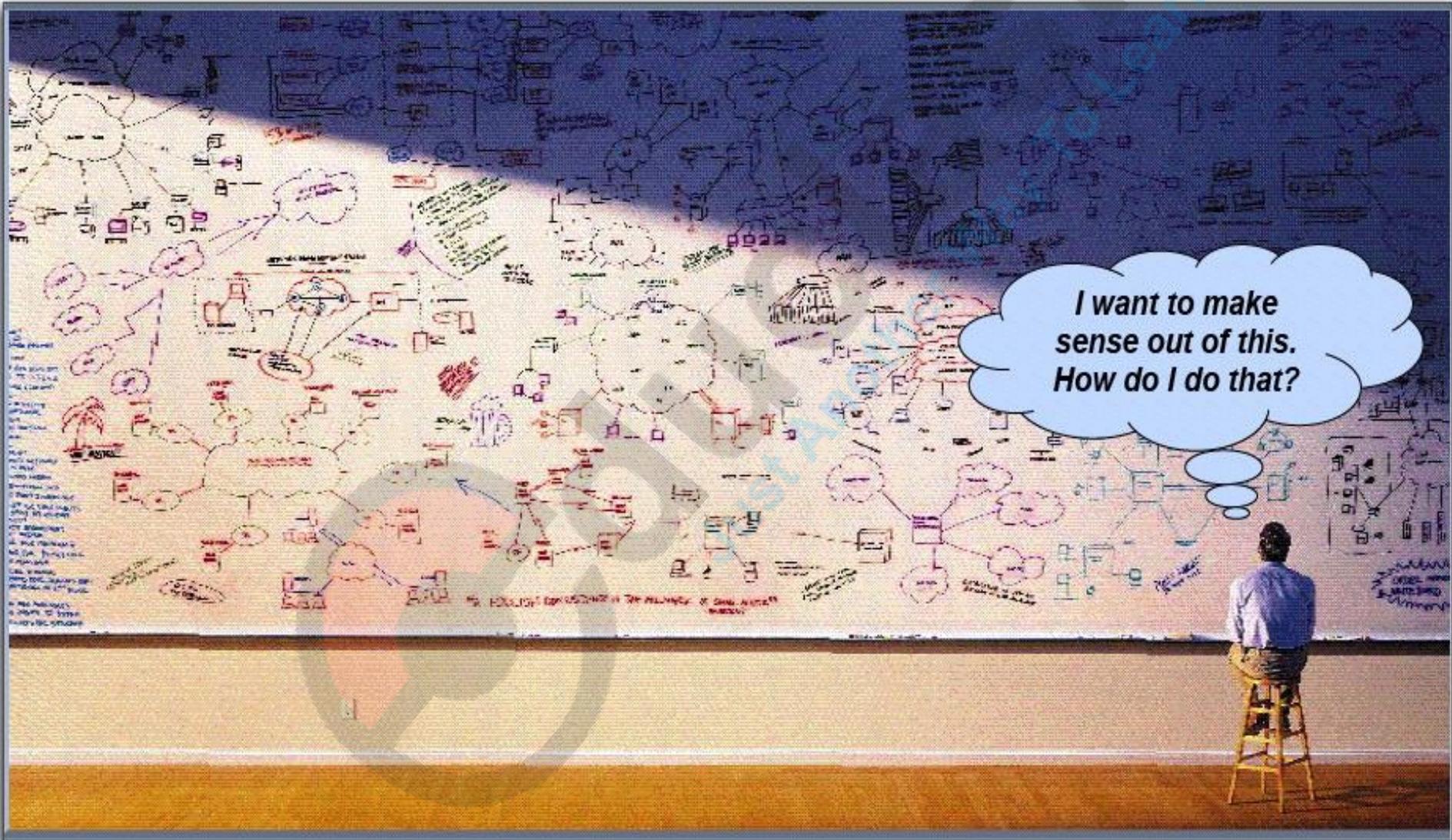


SERVICE-ORIENTED ARCHITECTURE

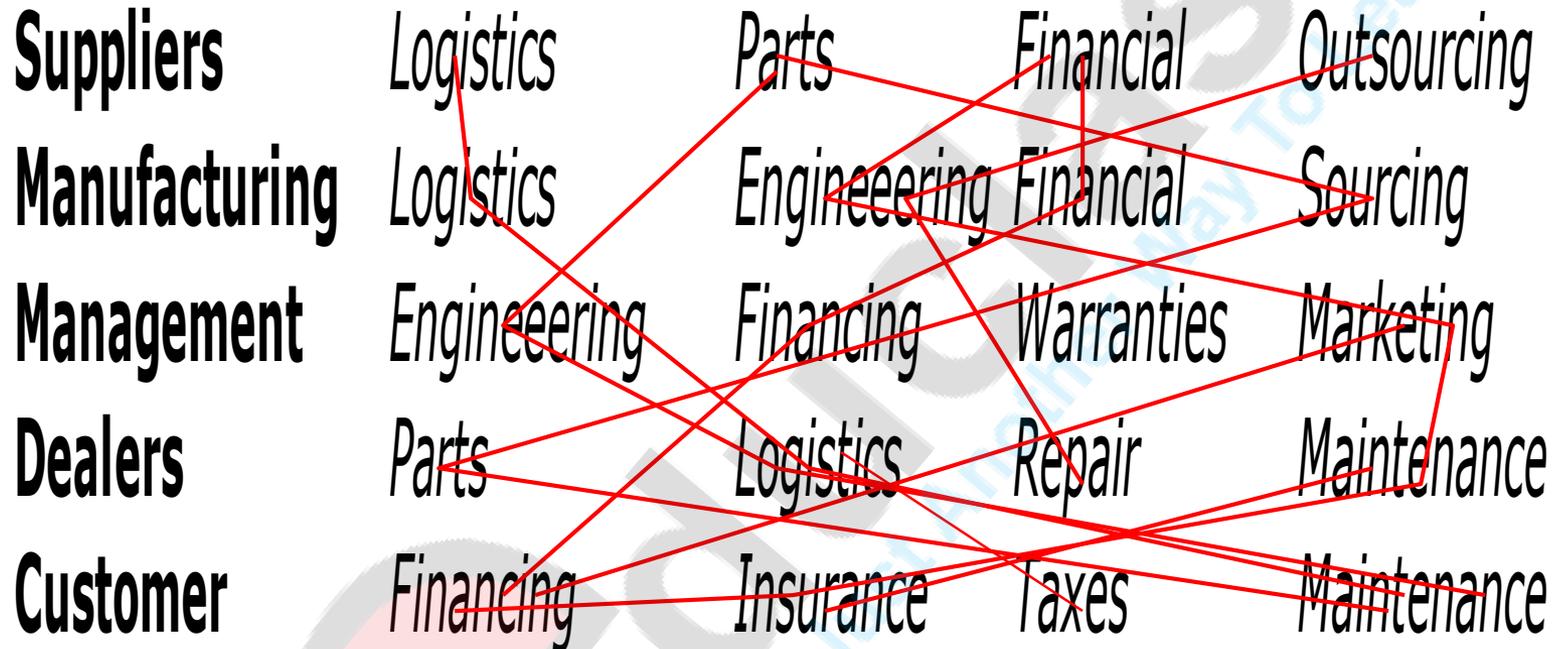
MODULE 8



Problem Addressed by a Service Oriented Architecture



Purpose of Architecture: To Manage Interdependencies



Architecture

- Architecture implies a consistent and coherent design approach. Essential principles include:
- **Consistency:** The same challenges should be addressed in a uniform way.
- **Reliability:** The structures created must be fit to purpose and meet the demands for which they are designed.
- **Extensibility:** A design must provide a framework that can be expanded in ways both foreseen and unforeseen.
- **Scalability:** The implementation must be capable of being scaled to accommodate increasing load by adding hardware to the solution.

What is a Service Oriented Architecture (SOA)?

- A method of design, deployment, and management of both applications and the software infrastructure where:
 - All software is organized into business services that are network accessible and executable.
 - Service interfaces are based on public standards for interoperability.
- A loosely-coupled architecture designed to meet the business needs of the organization.



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 changes business data from one state to another.
- A Service is the only way how data is accessed.
- A service may be as simple as “get me some person data,” or as complex as “process a disbursement

Characteristics of a Service

- **Supports open standards for integration:** Although 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.

Characteristics of a Service

- **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).

Characteristics of a Service

- **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.

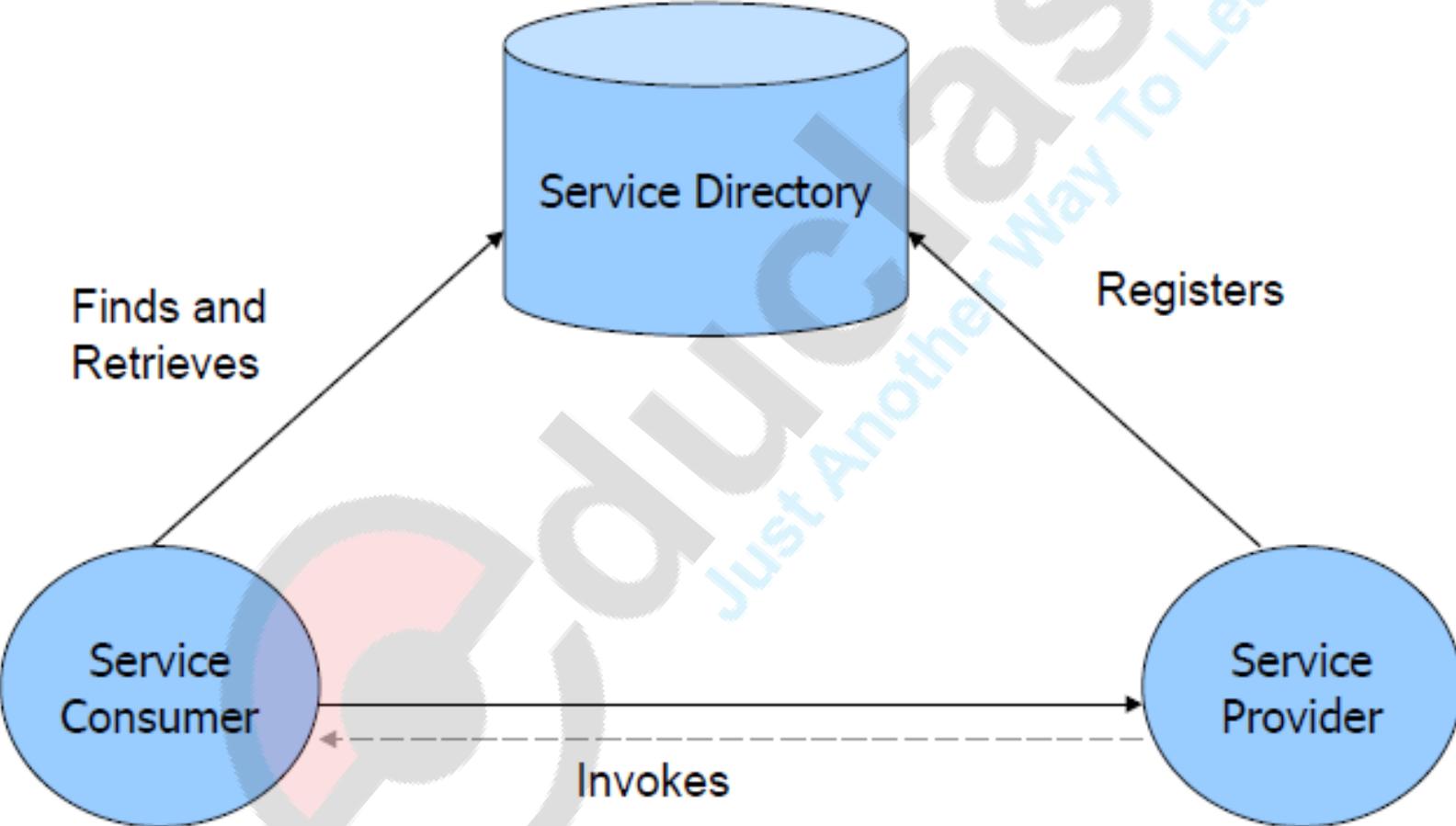
Characteristics of a Service

- **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.

SOA Defined

- “SOA is the architectural style that supports loosely coupled services to enable business flexibility in an interoperable, technology agnostic manner. SOA consists of a composite set of business-aligned services that support a flexible and dynamically re-configurable end-to-end business process realization using interface-based service descriptions.”

SOA ARCHITECTURE



Key Standards and Technology of SOA



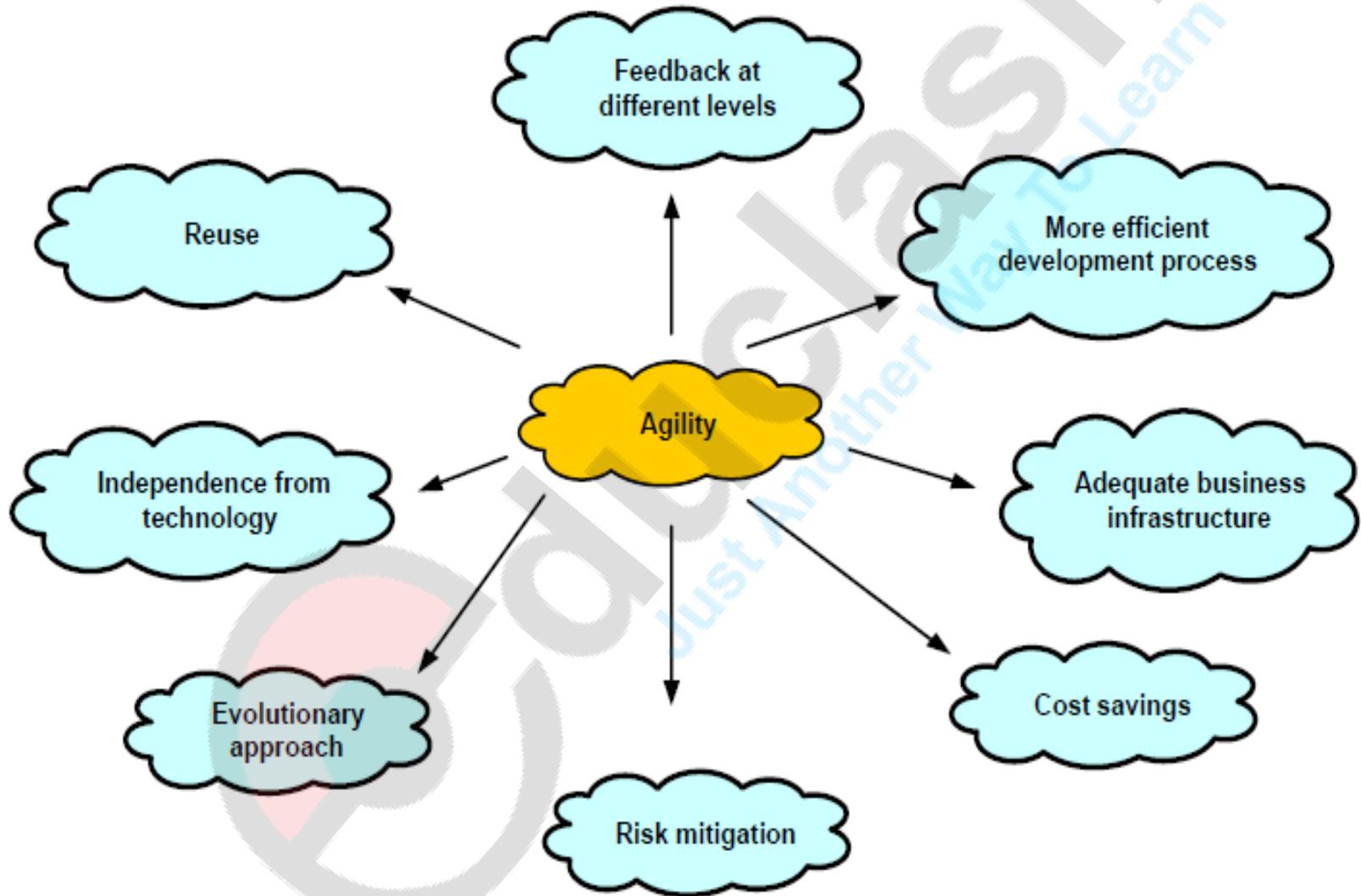
RPC vs Document Orientation

- While accessing Web Services , you follow either Document or RPC messaging style.
- The Document style indicates that the message body contains a XML document.
- RPC indicates that the SOAP message body contains names of the method and its parameters to generate XML structures.

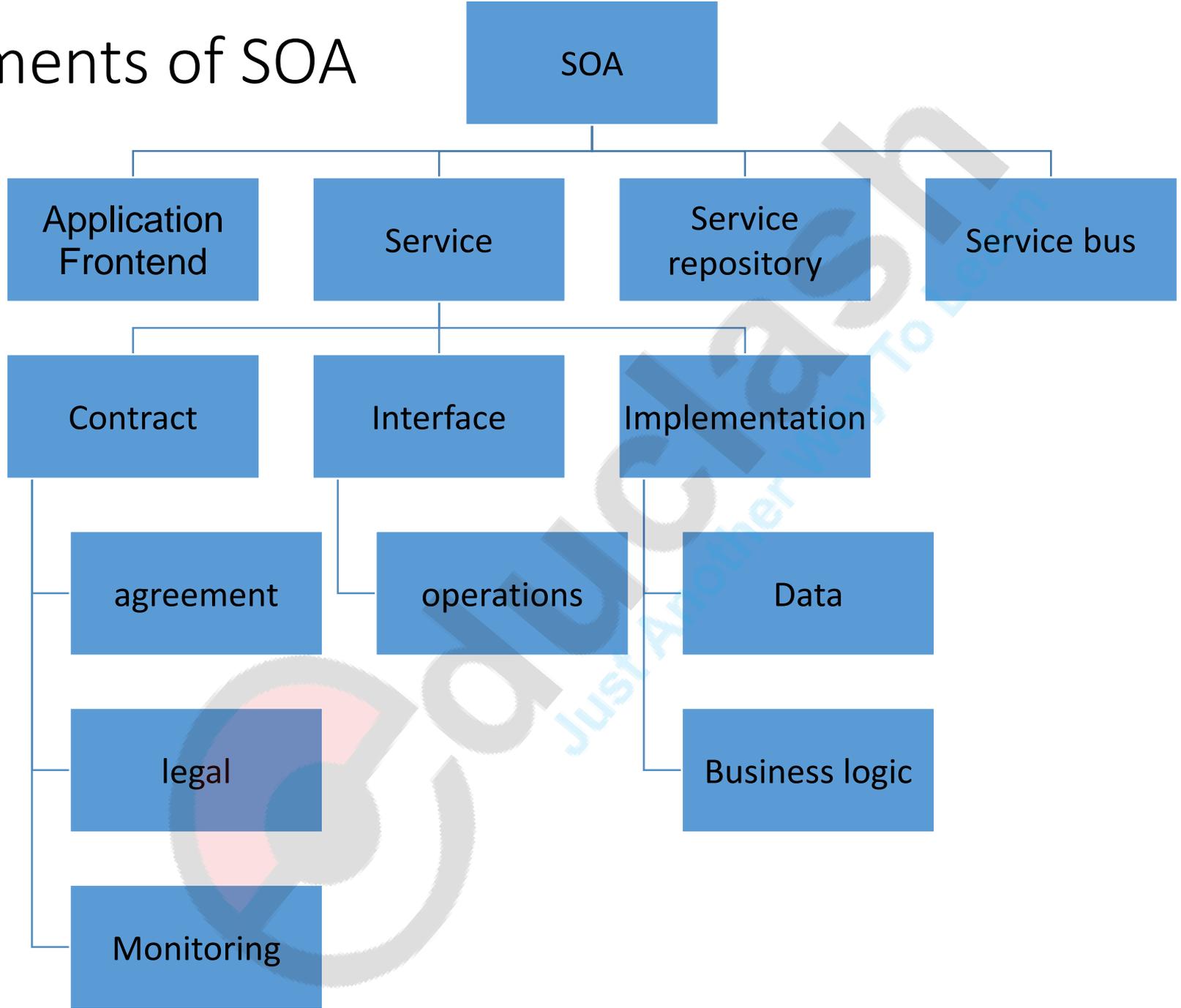


educlash
Just Another Way To Learn

BENEFITS OF SOA



Elements of SOA



Elements of SOA

1. **Application frontends:** are active elements of the SOA, delivering the value of SOA to the end users.
 - They initiate and control all activity of the enterprise system.
 - Web application, application with GUI, or a batch application.
2. **Service:** a software component that encapsulates a high level business concept.
 1. **Contract:** provides a specification of the purpose, functionality, constraints, and usage of services.
 2. **Interface:** functionality of the service exposed by the service to the clients that are connected to the service.
 3. **Implementation:** the service implementation provides the required business logic and appropriate data. It contains one or more of the artifacts: programs, configuration, data and databases.

- **Business logic:** business process represented by the service.
- **Data:** data represented in the service/ used by the service.

3. **Service repository:** it registers the services and their attributes to facilitate the discovery of services; operation, access rights, owner, qualities, etc.
4. **(Enterprise) Service Bus (ESB):** A flexible infrastructure for integrating applications and services by : routing messages, transforming protocols between requestor and service, handling business events and delivering them, providing QoS, mediation and security, and managing the interaction among services.
5. **Open standards:** publicly available implementable standards.

Composing Services

- A *service composition* is an aggregate of services collectively composed to automate a particular task or business process.
- To qualify as a composition, at least two participating services plus one composition initiator need to be present.



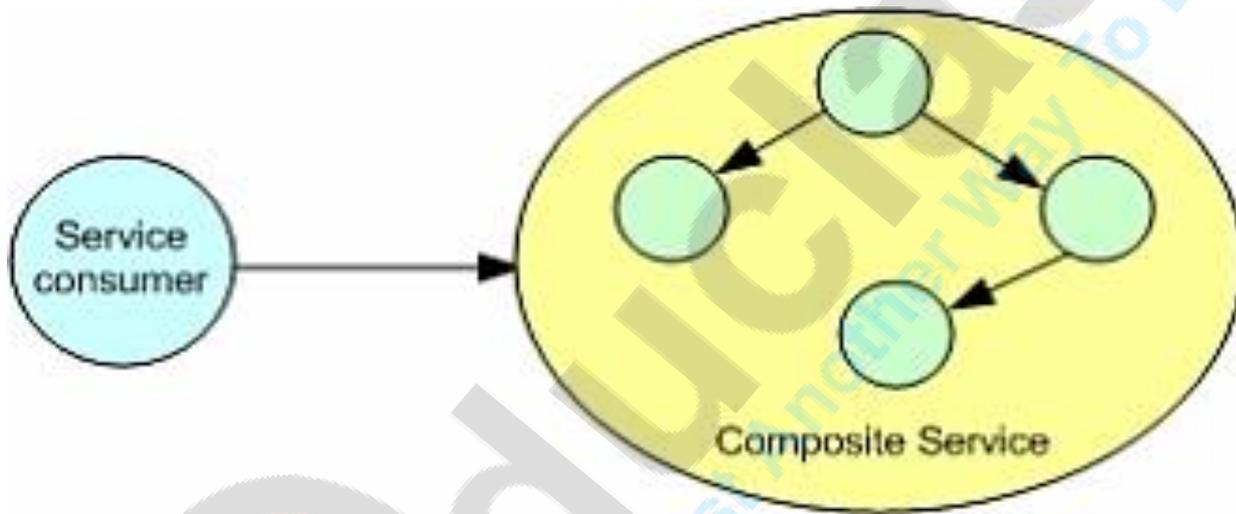
- There are two aspects of composition :
 - Composition Design: concerned with synthesizing a specification of how to coordinate the component services to fulfill the client request.
 - Composition Implementation: concerned with how to actually achieve the coordination among services.
 - Usually achieved by programming languages.



edupicrash
Just Another Way To Learn

Composition Design

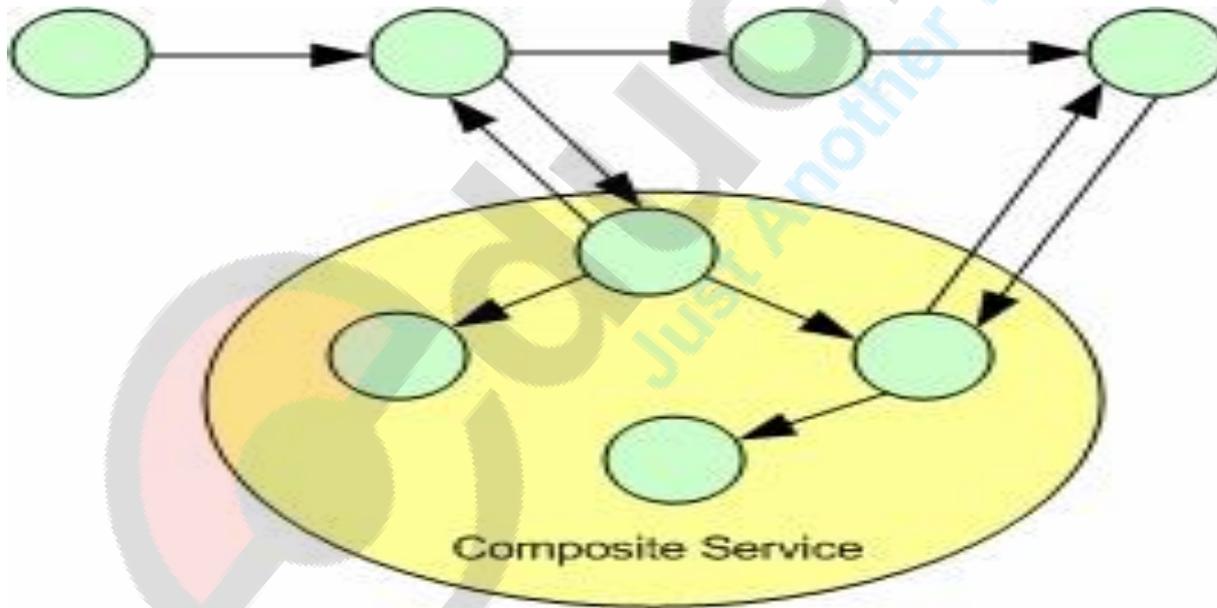
- Composition design is concerned with designing a solution based on a set of existing services.
- Its role is to specify a list of services involved in a composition
- Two ways of composing are:
 - Hierarchical Composition:
 - A consumer invokes this type of composite service, waits until its execution completes and uses the results.
 - Every level of hierarchy is implemented as an independent composite service, coordinating execution of lower level (composite) services.



Hierarchical Composition

- **Conversational Composition:**

- There are situations when a consumer needs to control execution of the composite service based on the intermediate execution results of service execution, which is supported by conversational composition.
- In this type of composition, the interacting consumer and provider are viewed as peers, exchanging data and control signals.

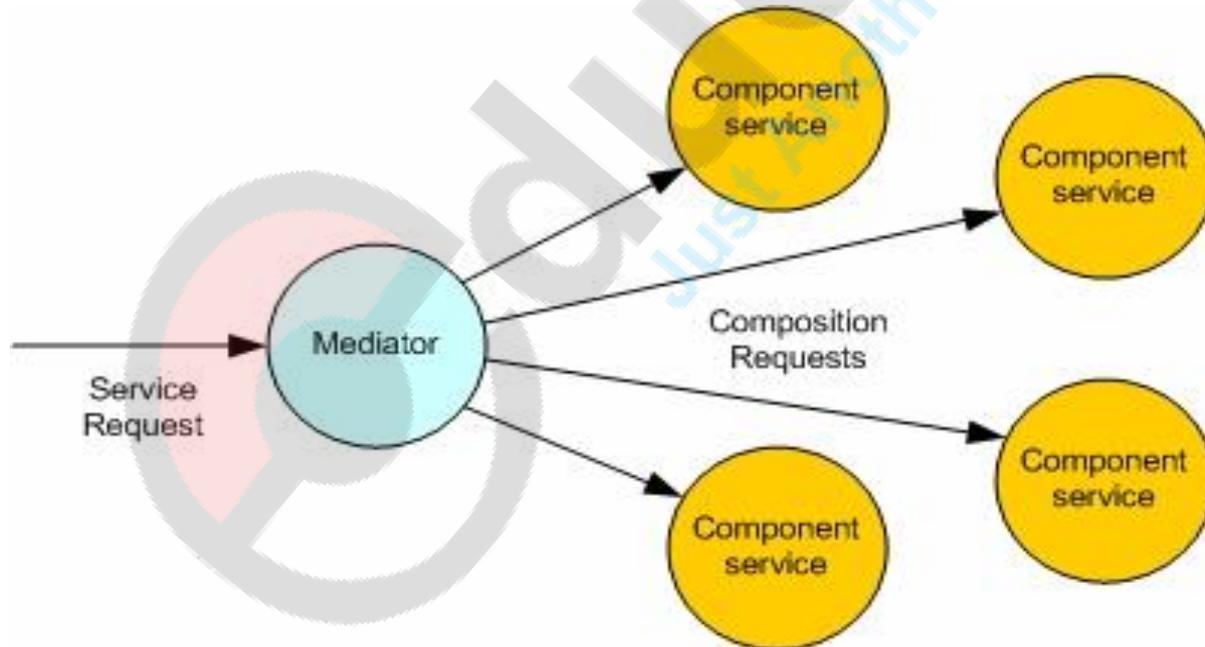


Conversational Composition

Composition Topologies

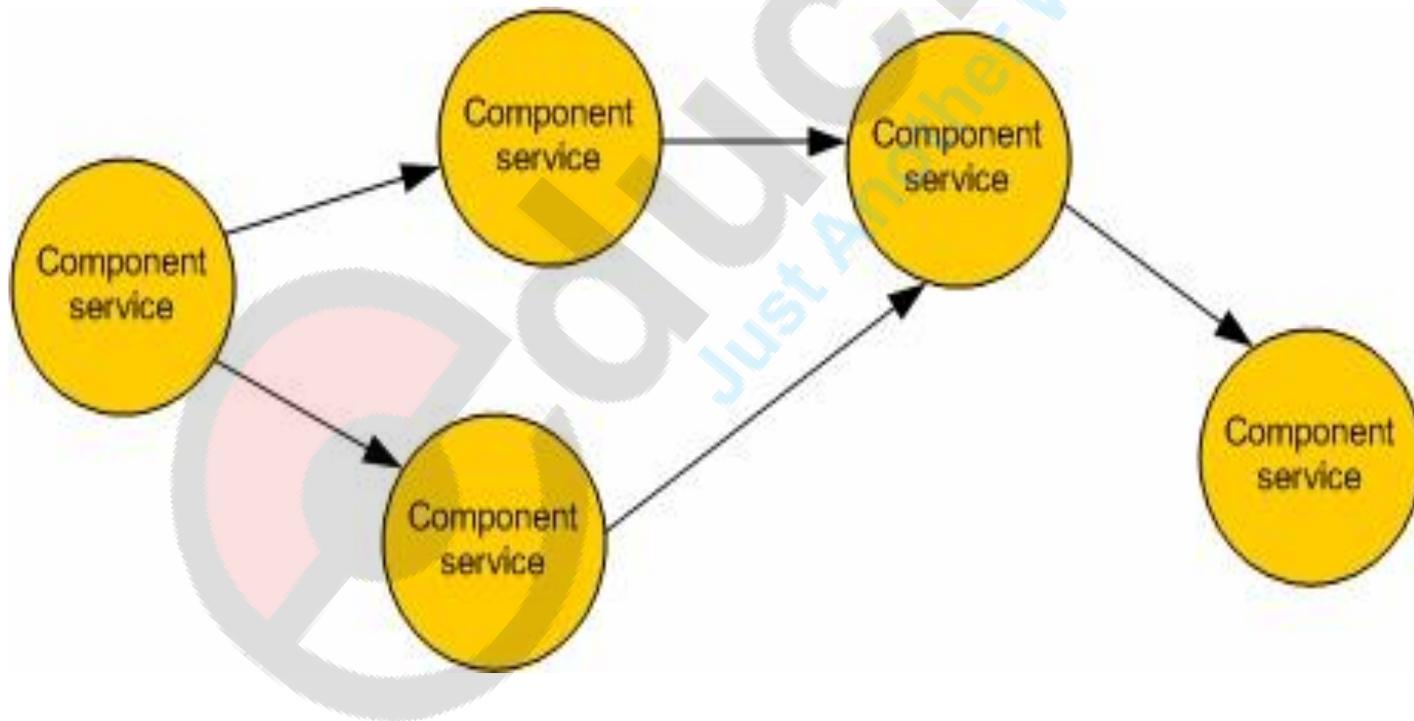
- Mediator-based

- Mediator-based topology assumes a single service, called the **mediator**, which has the specialized role of interacting with the service consumer and controlling the execution of the other services (component services) participating in composition.



- Peer-to-Peer:

- In the case of peer-to-peer topology there is no notion of a mediator service. Every participating service (component service) can execute (partial) composite service



Goals of Composition

- Service composition plays a significant role in SOA implementations by providing the following benefits:
 - **Improved reusability.** The composite service provides a natural way of reusing existing services. It allows service providers to add value through service composition.
 - **Faster time to market.** New solutions can be constructed more quickly. The existence of services at many levels along with a mechanism for combining and managing them allows new services and solutions to be constructed with less time and effort. In particular, operating prototypes can be assembled quickly from existing production quality services.

- **Improved security and auditability.** The composite service represents a single point of access to the set of underlying services. This single point of access provides a simple way of enforcing service invocation contracts, allowing for controlling and metering the access to the component services.
- **Lower duplication.** Redundancy is reduced or eliminated. Instead of replicating the same business functionality multiple times, a single business service that implements required functionality can be reused in multiple compositions.

Challenges of SOA

- Technical Challenges
- Security challenges - loosely coupled environment
- Performance - XML brings robustness not speed
- Optimization
- Organizing the services – registry & repository
- Finding the right services and right interfaces
- Transaction management is complex in interactions between logically separate system

Probable Questions

- What is a service? What is Service-oriented Architecture? Discuss different elements of Service-oriented Architecture.
- Discuss different characteristics of SOA?
- Define Service Composition. Discuss various aspects (composition design and implementation) of service composition.
- What are different goals and challenges of SOA?