

Cloud Computing Concepts

CS3132

Dr. Anand Kumar Mishra

NIIT University

Topic – 3

- Service Oriented Architecture –
 - Role: Service Provider, Service Broker, Service Requester
- Quality of Service
- Protocol stack for SOA architecture
- Advantages of SOA
- Applications

Service Oriented Architecture

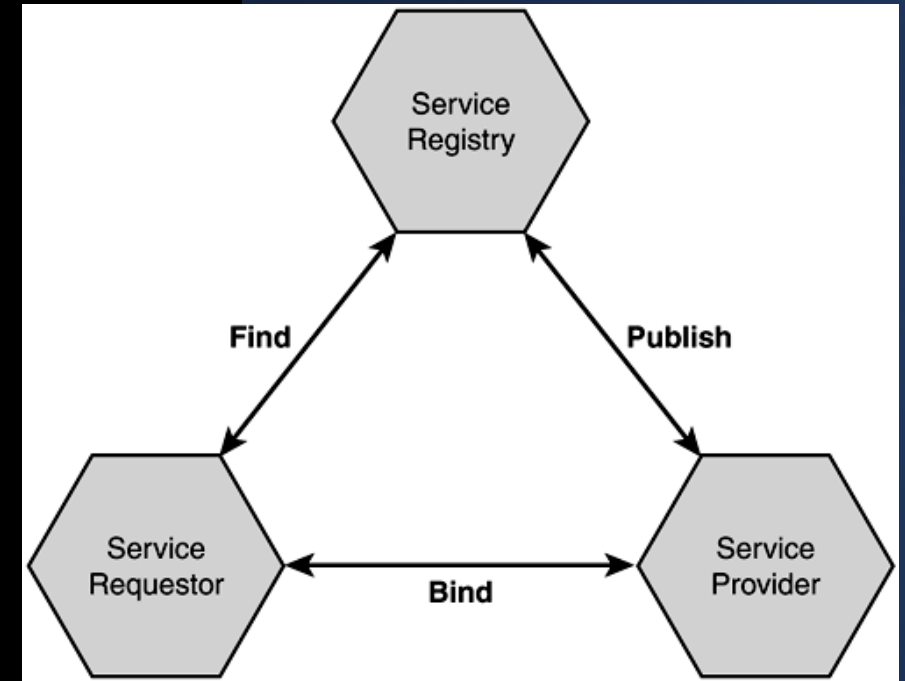
- A loosely-integrated suite of services
 - The service interfaces provide loose coupling,
 - meaning they can be called with little or **no knowledge of how the service is implemented** underneath,
 - **Reducing the dependencies** between applications
- A standard method for **requesting** services from distributed components and **managing** the results

Service Oriented Architecture

- **Clients** requesting services
- **Components** providing the services
- **Protocols** used to **deliver** messages
- **Responses**
 - SOA provides
 - **translation and management layer in an architecture**
 - **removes the barrier for a client obtaining desired services**

Service-Oriented Architecture contains three roles

- Service Requester
- Service Provider
- Service Registry



SOA - Service Provider

- A service provider is responsible for:
 - creating a service description,
 - publishing that service description to one or more service registries, and
 - receiving Web service invocation messages from one or more service requestor
- A service provider can be any company that hosts a Web service made available on some network
 - Think of a service provider as the "server side" of a client-server relationship between the service requestor and the service provider

Service Description

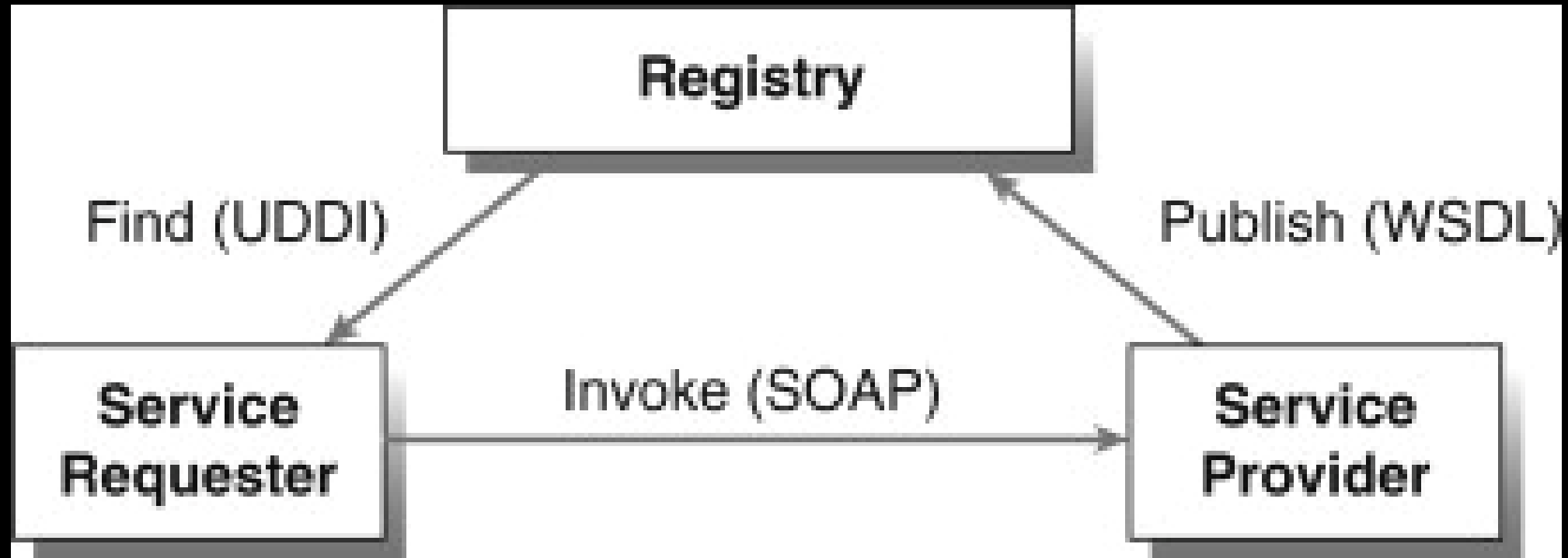
- Service description that is retrieved by the service requestor as a result of the find operation
- Service description tells the service requestor everything it needs to know in order to bind to or invoke the Web service provided by the service provider
- The service description also indicates what information (if any) is returned to the service requestor as a result of the Web service invocation

SOA – Service Registry

- The service registry is responsible for:
 - Advertising Web service descriptions published to it by service providers and for allowing service requestors to search the collection of service descriptions contained within the service registry
- The service registry role is simple:
 - be a match-maker between service requestor and service provider
 - Once the service registry makes the match, it is no longer needed in the picture;
 - the rest of the interaction is directly between the service requestor and the service provider for the Web service invocation

SOA – Service Requester

- A service requestor is responsible for:
 - finding a service description published to one or more service registries and
 - It is responsible for using service descriptions to bind to or invoke Web services hosted by service providers
- Any consumer of a Web service can be considered a service requestor
 - Think of a service requestor as the "client side" of a client-server relationship between the service requestor and the service provider



SOA Operations



Publish



Find



Bind

SOA – Publish Operation

The publish operation is an act of service registration or service advertisement

It acts as the contract between the service registry and the service provider

When a service provider publishes its Web service description to a service registry, it is advertising the details of that Web service to a community of service requestors

The actual details of the publish API depend on how the service registry is implemented

SOA – Find Operation

The find operation is the contract between a service requestor and a service registry

With the find operation:

- the service requestor states a search criteria, such as:
 - type of service,
 - various other aspects of the service such as -
 - quality of service guarantees

The service registry matches the find criteria against its collection of published Web service descriptions

The result of the find operation is a list of service descriptions that match the find criteria

Find operation always returns all Web services published to the service registry

- Its the service requestor's job to figure out which Web service description matches its need

SOA – Bind Operation

The bind operation embodies the client-server relationship between the service requestor and the service provider

The bind operation can be quite sophisticated and dynamic, such as:

- on-the-fly generation of a client-side proxy based on the service description used to invoke the Web service; or
- it can be a very static model, where a developer hand-codes the way a client application invokes a Web service

SOA – Quality of Service

Policy

- Set of protocols according to which a service provider make and provide the services to consumers

Security

- Set of protocols required for identification and authorization

Transaction

- Complete a business function

Management

- Set of attributes used to manage the services

Basic principles of SOA

- Interoperability - Any client system can run a service, regardless of the underlying platform or programming language
 - For instance, business processes can use services written in both C# and Python. Since there are no direct interactions, changes in one service do not affect other components using the service

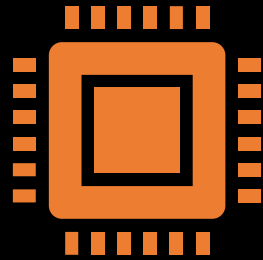
Basic principles of SOA

- Loose coupling - Services in SOA should be loosely coupled, having as little dependency as possible on external resources such as data models or information systems
 - They should also be stateless without retaining any information from past sessions or transactions
 - This way, if you modify a service, it won't significantly impact the client applications and other services using the service

Basic principles of SOA

- Abstraction - Clients or service users in SOA need not know the service's code logic or implementation details
 - To them, services should appear like a black box
 - Clients get the required information about what the service does and how to use it through service contracts and other service description documents.
- Granularity - Services in SOA should have an appropriate size and scope, ideally packing one discrete business function per service
 - Developers can then use multiple services to create a composite service for performing complex operations

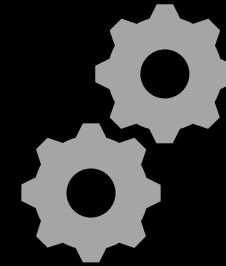
Benefits of service-oriented architecture



Faster time to market

Developers reuse services across different business processes to save time and costs

They can assemble applications much faster with SOA than by writing code and performing integrations from scratch



Efficient maintenance

It's easier to create, update, and debug small services than large code blocks in monolithic applications

Modifying any service in SOA does not impact the overall functionality of the business process

Benefits of service-oriented architecture

- Greater adaptability
 - SOA is more adaptable to advances in technology. You can modernize your applications efficiently and cost effectively
 - For example, healthcare organizations can use the functionality of older electronic health record systems in newer cloud-based applications

Limitations in implementing service-oriented architecture

- Limited scalability - System scalability is significantly impacted when services share many resources and need to coordinate to perform their functionality
- Increasing interdependencies – SOA systems can become more complex over time and develop several interdependencies between services
 - They can be hard to modify or debug if several services are calling each other in a loop
 - Shared resources, such as centralized databases, can also slow down the system

Limitations in implementing service-oriented architecture

- High overhead - A validation of input parameters of services is done whenever services interact this decreases performance as it increases load and response time
- High investment - A huge initial investment is required for SOA