Service Oriented Architecture (SOA)

Engr. Afzal Ahmed

Service Oriented Software Engineering

- Developments in service-centric S.E. have been rapid
  - for example, standards such as Simple Object Access Protocol (SOAP) and the Web Services Description Language (WSDL).
- Major vendors such as IBM, Microsoft, Sun, and Hewlett-Packard now provide support for services in their development platforms.

What is SOA?

"Service-Oriented Architectures (SOA) is a set of patterns for building distributed systems where one application that comprises a system can find another application that provides needed service and can exchange data with it."

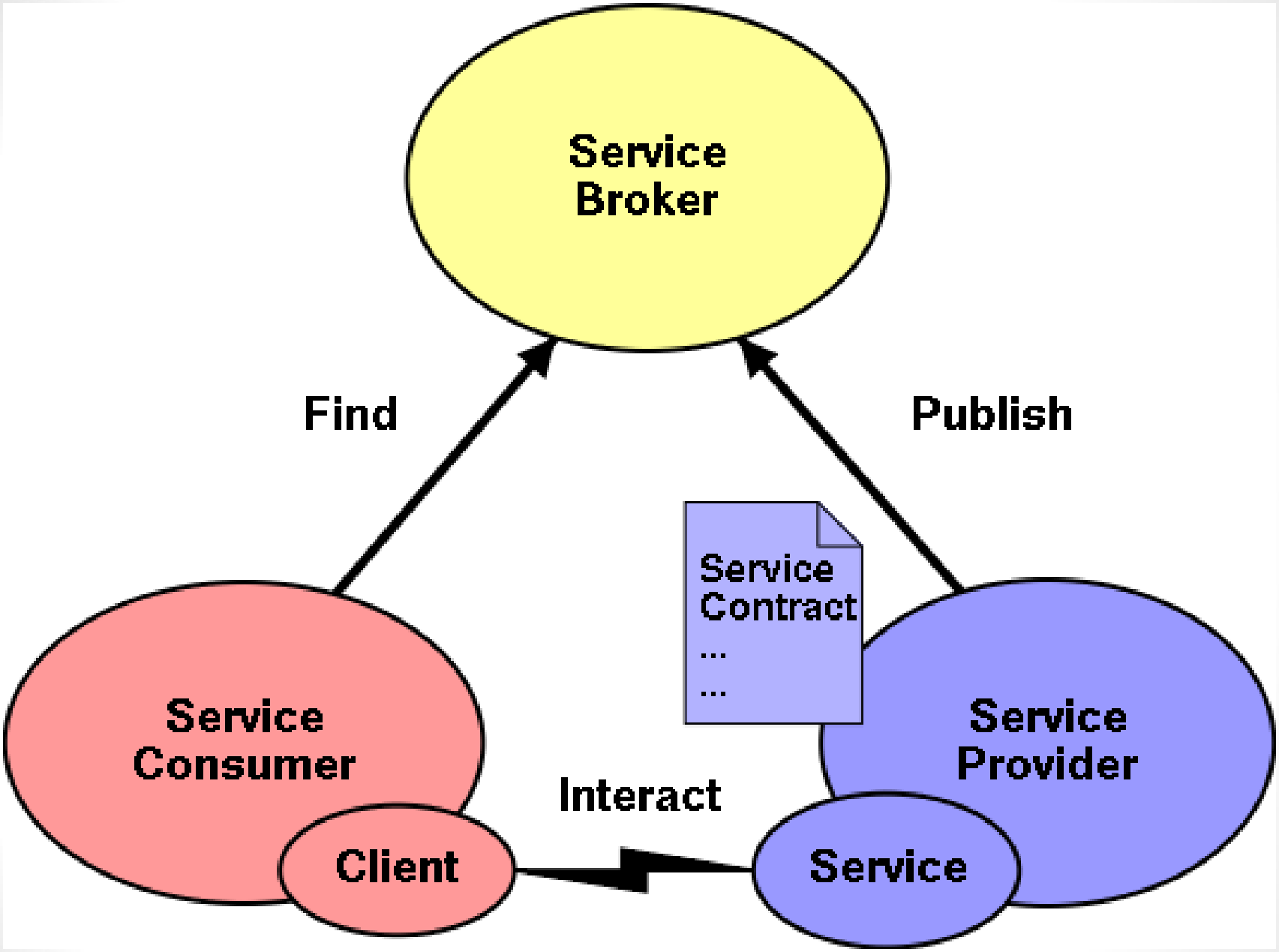
SOA is not a product

Why is SOA Different?

- Loose coupling
  - No interdependencies between systems
- Reuses existing technology infrastructure
- Everything is based on XML and other industry standards
  - Platform and language independent
  - Easy, inexpensive
- Industry momentum:
  - Every player is on board: IBM, Microsoft, SAP, Oracle …
  - Proven interoperability

SOA

- Evolves from the component-based computing by splitting the developers into three independent but collaborative entities:
  - the application builders (also called service requestors)
  - the service brokers (or publishers)
  - The service developers (or providers).


Service
Broker
Find
Publish
Service
Contract
...
...
Service
Consumer
Service
Provider
Interact
Client
Service

Service Oriented Architecture (SOA)

- SOA is a system architecture in which a collection of loosely coupled services (components) communicate with each other using standard interfaces and message exchanging protocols.
- These services are autonomous and are platform independent.
- They can reside on different computers and use each others' services to achieve the desired goals.
- A new service can be composed at runtime based on existing services.
- Remote services can be searched and discovered through service brokers that publish services for public accesses.

SOA Variants

- Producer-centric:
  - Producers publish services; consumers search and discover their needed services, and use them in their application development.
- Consumer -centric:
  - Consumers publish their needs, and let providers supply the needed services and/or workflows, collaboration templates, or even application templates.
- Broker -centric:
  - Brokers publish test scripts or specification, and let producers supply the services/workflows, and let consumers discover the services and use the test scripts for testing.

SOA

- Characteristics
  - Reusability
  - Dynamic and adaptive nature
  - Standard-based interoperability

Common Misperceptions

- SOA is just Web services.
- SOA is just a marketing term.
- SOA is just distributed computing.

SOA is a magic global solution to general interoperability.

SOA Example: Web Services

- Web Services implement a Web-based SOA and a set of enabling standardized protocols such as XML, WSDL, UDDI, and SOAP.
- Web services are operations that users access via the Internet through a well-defined interface independent of where the service is executed.
- Service-centric systems integrate Web services into applications that discover, compose, invoke, and monitor these services.

SOA Example: Web Services

- App Store (Apple, Android, Microsoft)
  - Search
  - Shopping (Google wallet)
  - Google Docs

SOA Example: Steam (Gaming Platform)

- Steam
  - Publish Games (AAA + Indie)
  - Search Games
  - Buy Games

IBM SOA Foundation Architecture

- SOA development lifecycle have been developed, and they are different from traditional software development lifecycles.

- Two development activities
  - modeling and assembly
- Two operation activities
  - deployment and management
- The four phases are performed iteratively.
- The entire process will be controlled and orchestrated by the governance policies.

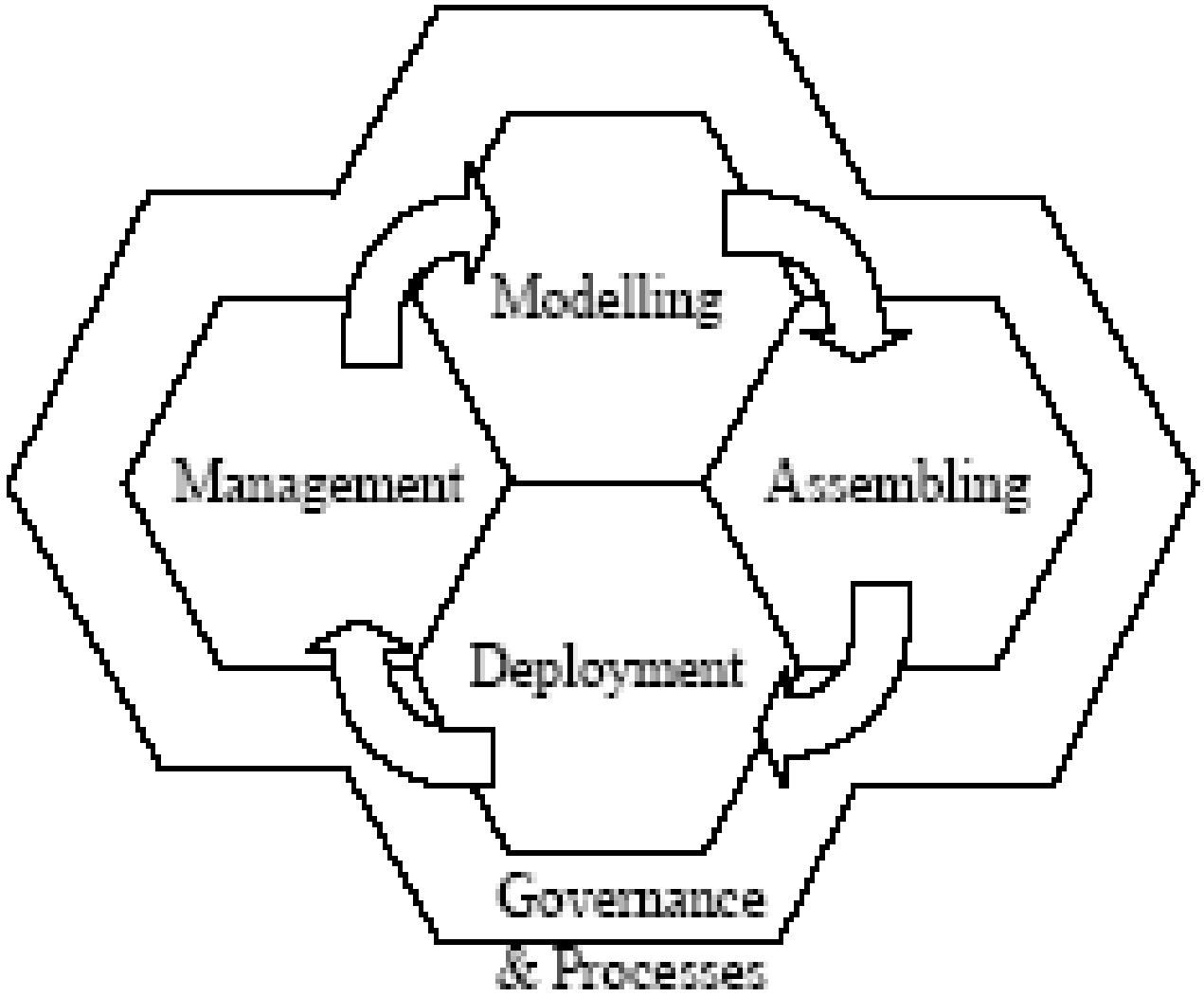


Figure 2 IBM SOA Foundation Architecture

IBM SOA Foundation Architecture

- One distinction from traditional lifecycle models is that the SOA lifecycle includes an explicit process of discovering reusable assets such as services and workflows.
- IBM SOA Foundation model has verification and validation activities, but they are not explicitly shown in the diagram.
- Traditional lifecycle models explicitly show these activities such as simulation, testing, model checking, and completeness and consistency checking.

Major Features of SOA

- **Reusability-Oriented and Cumulative**
  - SOA emphasizes on reusability, once an item is published, it can be reused by others.
  - As SOA items are identified and developed, they can be stored for reuse.

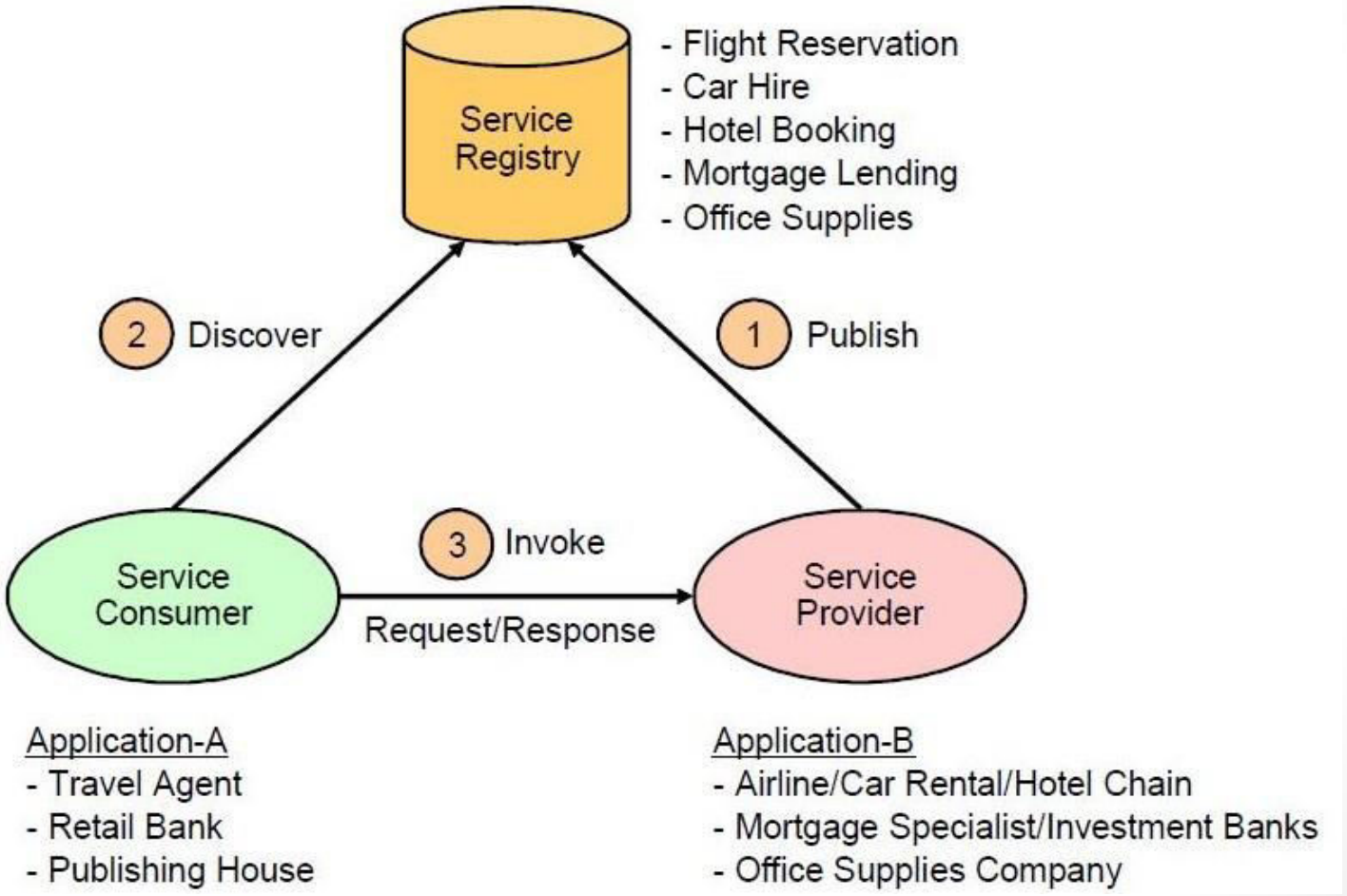
Major Features of SOA

- **Framework-Oriented Analysis**
  - As reusable assets will be accumulated, many assets will be available, and these assets will be organized in some manner, such as according to the classification tree in an ontology system or other manners.
  - Requirement analysis is no longer confined to analyzing the requirements given, but also examining the reusable assets available in the given framework to see if these assets can be reused. This process is called framework-oriented analysis.

Major Features of SOA

- **Domain-Specific**
  - An important feature of SORE is that it will use domain-specific items including ontology, services, workflows, collaboration templates, application templates, user interfaces, and policies, and thus SORE is inherently domain specific.
  - For example, a banking application may use reusable banking-related ontology, services, workflows and application templates, but it probably will not find biomedical ontology, services and workflows helpful.

Basic Components of SOA



De-mystifying SOA

- **SOA is…**

☑ A design philosophy and Architecture

☑ A means

☑ A solution

☑ Achieved through Web Services and related technologies

- **SOA is NOT…**

☒ A technology or a methodology

☒ Not an end

☒ Not a product

☒ Only Web Services

According to the SOA

A Web service is:

- An interface that describes a collection of network accessible operations
- Described using a service description language
- Published by making this service description available to users
- Found by sending queries to a registry matching service descriptions
- Composed with other services to create new services

Trade off SOA

- Save Development Time
- Can cost more
- Need to prioritize.

Summary

- Developments in service-centric computing have been rapid.
- S.E is changing with time
- Traditional S.E. will always be there.