# Service Oriented Architecture

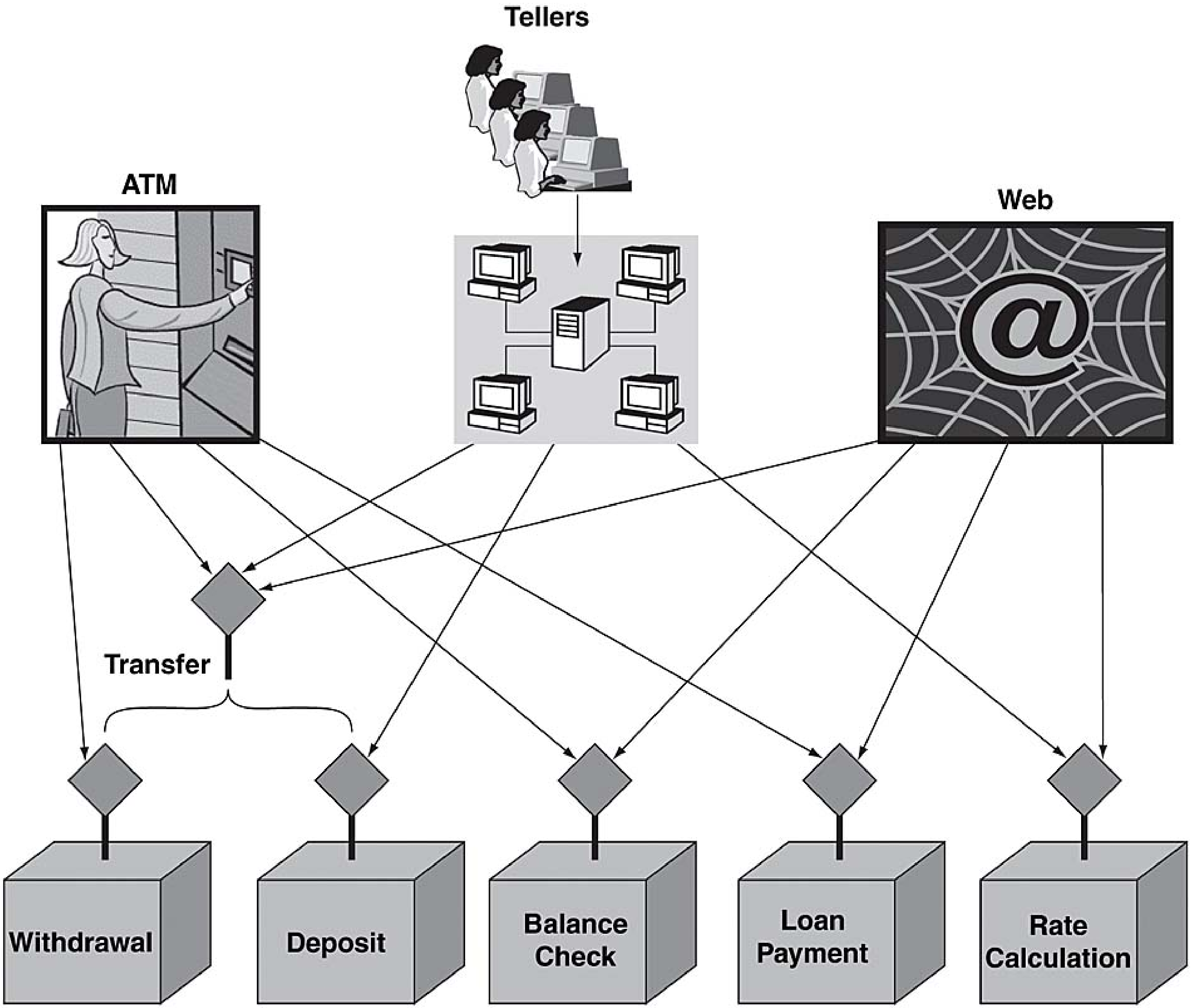
Prof. P. M. Jadav
Associate Professor,
CE Department,
Faculty of Technology,DDU

# What are Services?

- Most organizations (whether commercial or government) provide services to customers, clients, citizens, employees, or partners

# Services

- **Bank** (as an Organization) provides services to its customers:
    - Account management (opening and closing accounts).
    - Loans (application processing, inquiries about terms and conditions, accepting payments)
    - Withdrawals, deposits, and transfers
    - Foreign currency exchange


Tellers
ATM
Web
Transfer
Withdrawal
Deposit
Balance Check
Loan Payment
Rate Calculation

# Service-Oriented Architecture

- A SOA is a style of design that guides all aspects of creating and using business services throughout their lifecycle
- An SOA is also a way to define and provision an IT infrastructure to allow different applications to exchange data and participate in business processes, regardless of the OS or programming languages

# Service-Oriented Architecture

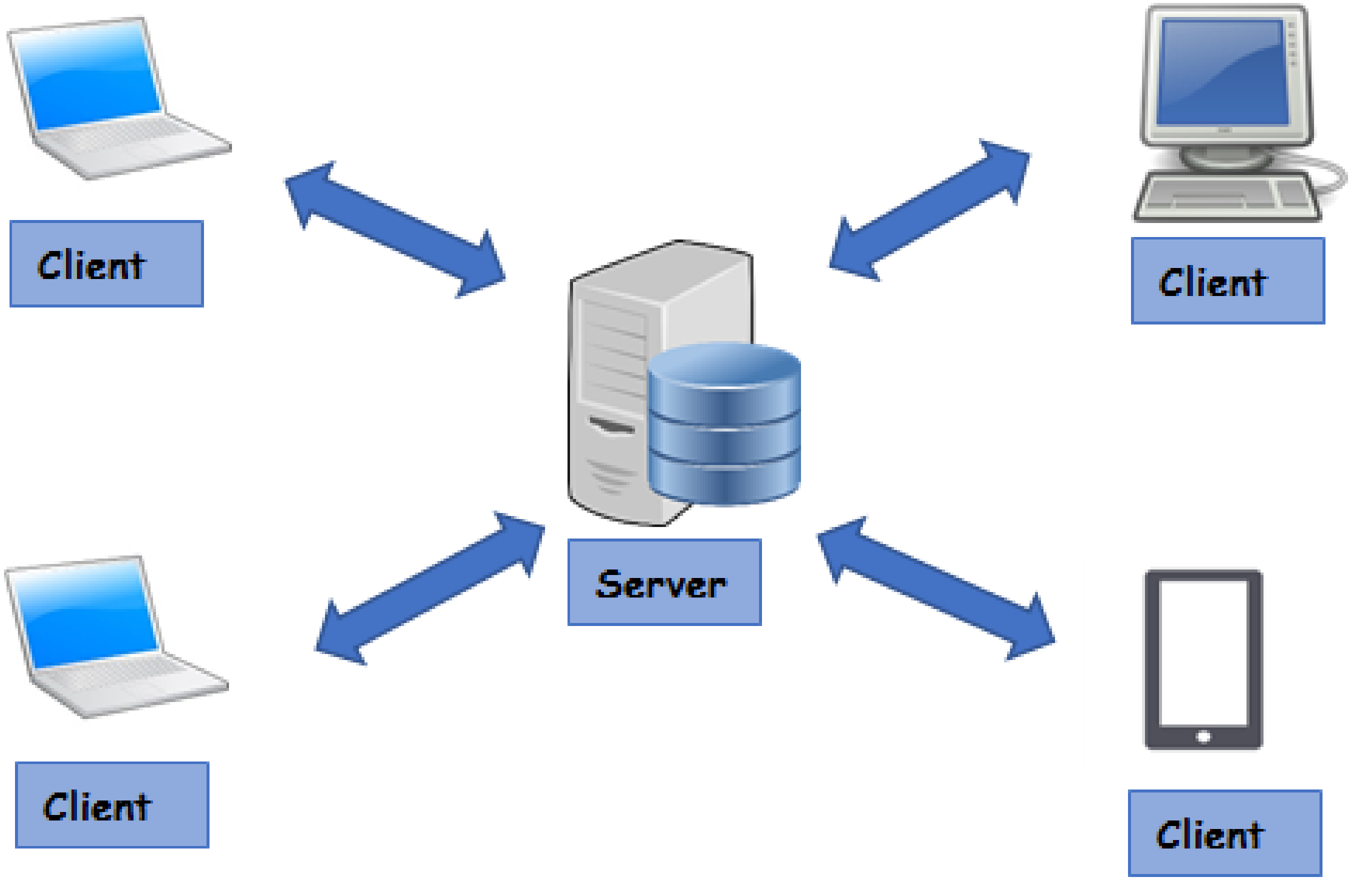
- Its an approach to building IT systems in which business services are the key organizing principle used to align IT systems
- Earlier approaches to building IT systems tended to directly use specific implementation environments such as
  - object orientation
  - procedure orientation and
  - message orientation

  to solve these business problems, resulting in systems that were often tied to the features and functions of a particular execution environment technology such as CICS, IMS, CORBA, J2EE, and COM/DCOM.

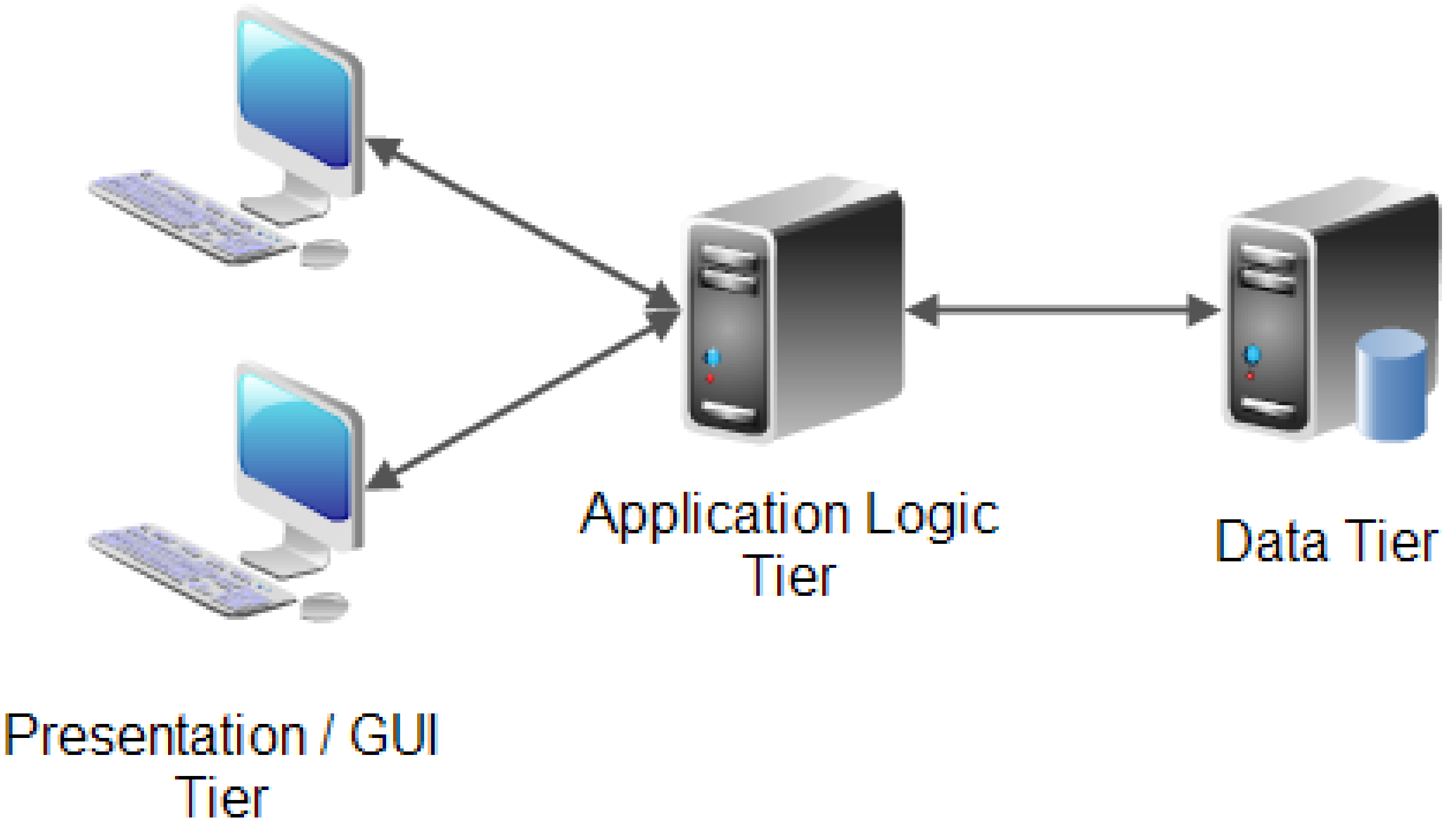
# Evolution of SOA

- Procedural Programming
- Object Oriented Programming
- Centralized Systems
- Client Server Architecture
- N-tier Architecture
- RPC/RMI
- Distributed Architecture
- Cloud Computing

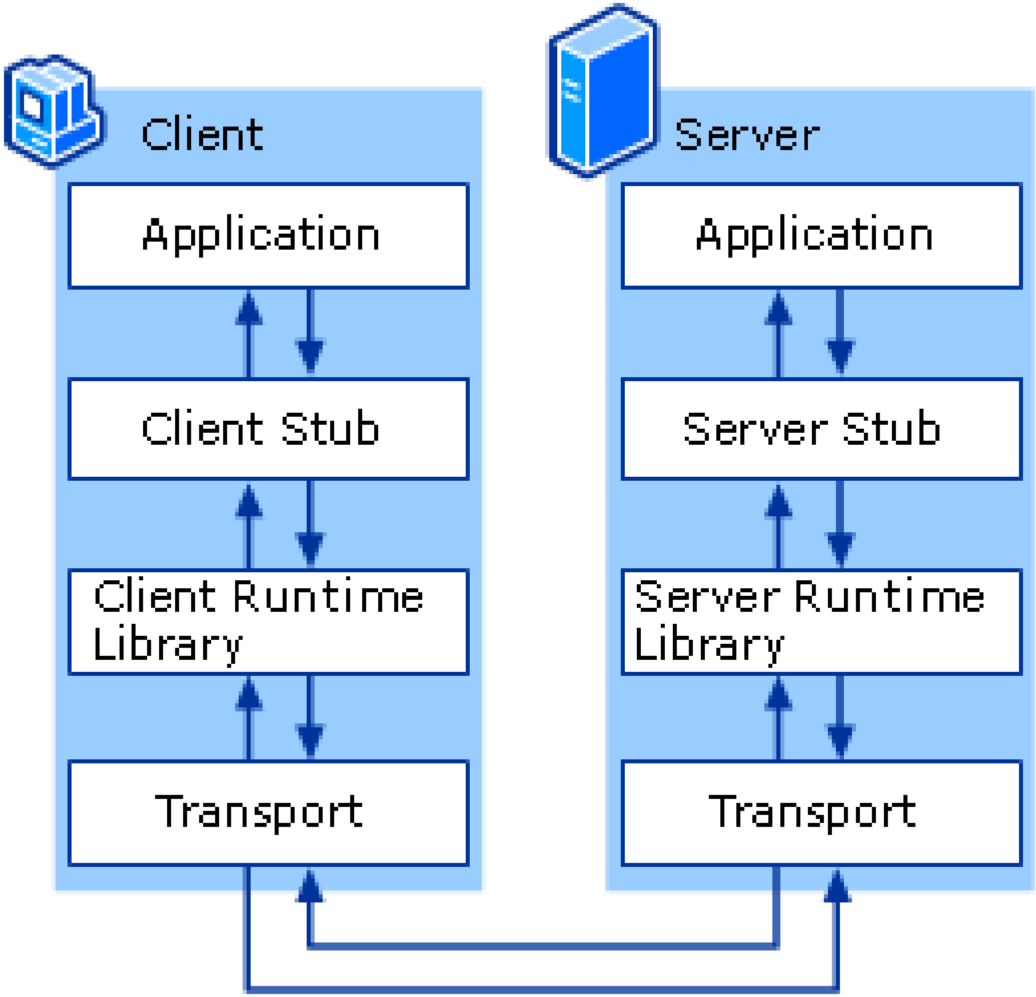
# Client Server Architecture



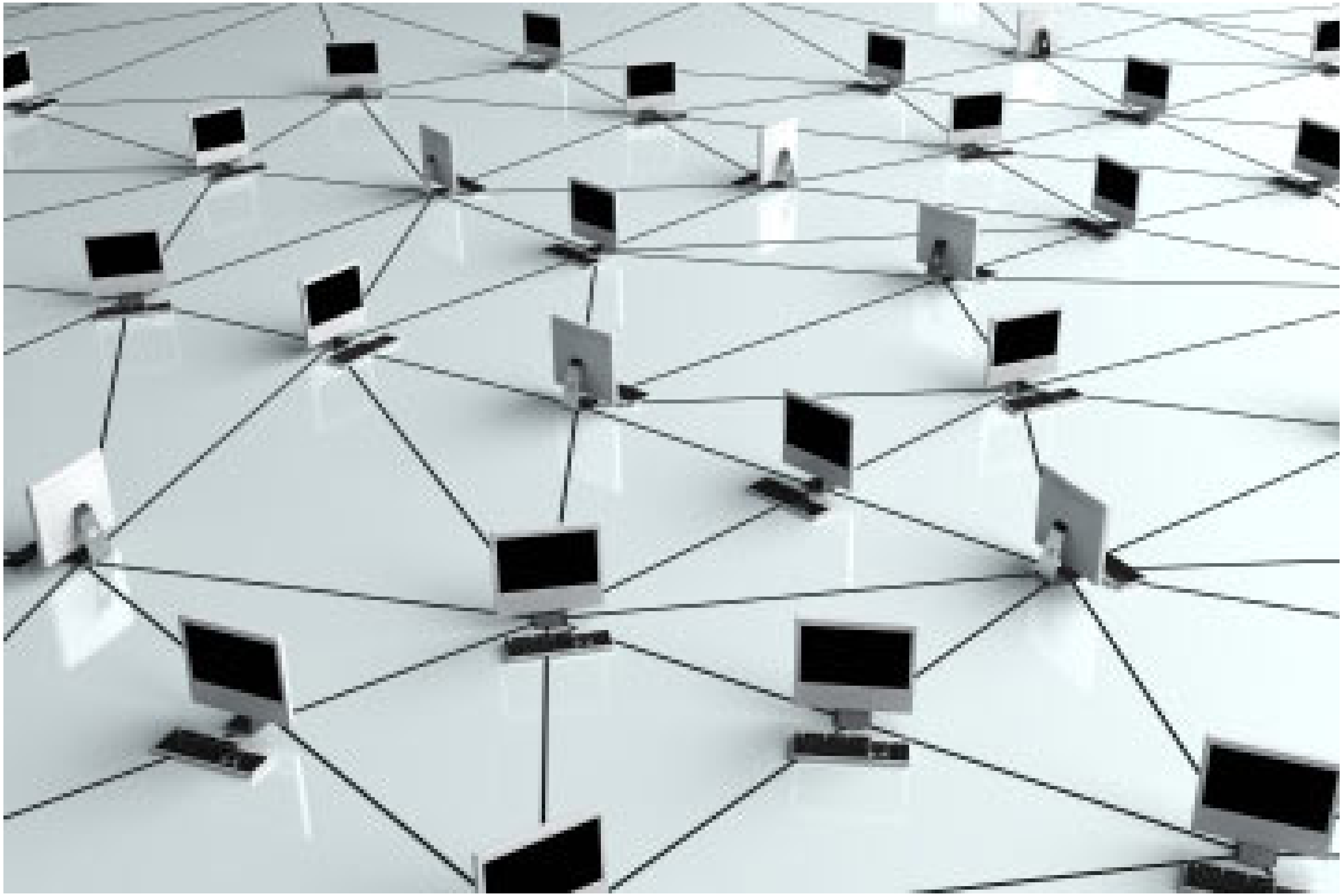
# 3-Tier Architecture



# RPC Communication



# Distributed Systems



# Advantage of Distributed Systems

- Availability
- Fault tolerance and Recovery
- Concurrency
- Scalability
- Transparency
- Load balancing
- Performance
- Heterogeneity
- Openness
- Loose coupling

# Web Service Roles

- **Service Provider**
  - The service provider implements the service and makes it available on the Internet.
- **Service Requestor**
  - This is any consumer of the web service
  - The requestor utilizes an existing web service by opening a network connection and sending an XML request.
- **Service Registry**
  - This is a logically centralized directory of services
  - The registry provides a central place where developers can publish new services or find existing ones

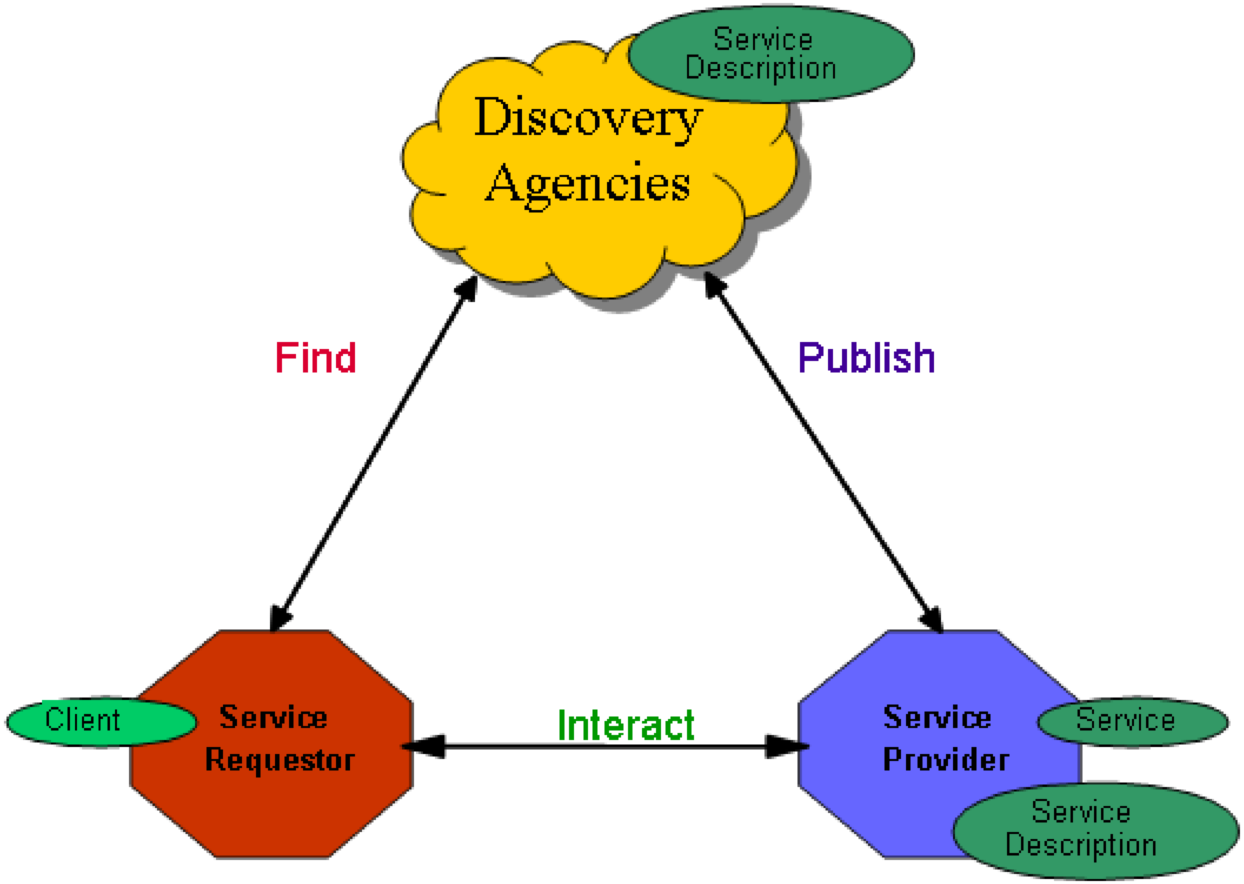
# Web Service Protocol Stack

- **Service Transport**
  - Responsible for transporting messages between applications
  - HTTP,  SMTP, Blocks Extensible Exchange Protocol (BEEP)
- **XML Messaging**
  - Encoding messages in a common XML format so that messages can be understood at either end
  - XML-RPC and SOAP

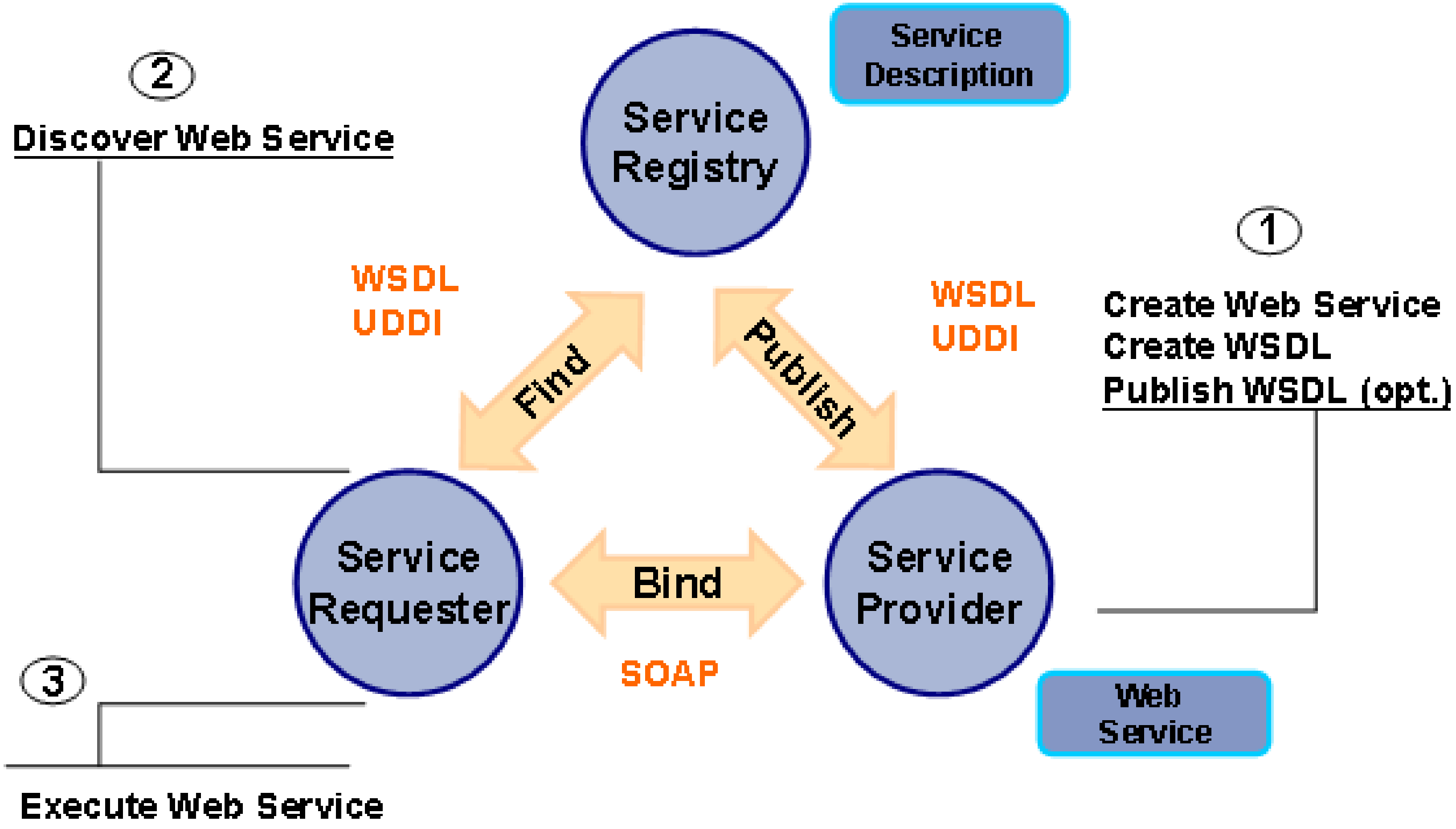
# Web Service Protocol Stack

- **Service Description**
  - describes the public interface to a specific web service
  - Web Service Description Language (WSDL)
- **Service Discovery**
  - centralizes services into a common registry and providing easy publish/find functionality
  - Universal Description, Discovery, and Integration (UDDI)

# Service Oriented Architecture



# Service Oriented Architecture



# Advantages of SOA

- Interoperability
- Usability
- Reusability
- Standardized Protocol
- Low cost communication
- Enterprise Application Integration

# Web Services Examples

- Public/Private Web services
- http://www.webservicex.net/WS/wscatlist.aspx
- Weather service
- Currency converter service
- Stock Market Quotes
- Login (google/facebook/linkedin)
- Payment (debit card/credit card/netbanking)
  - Support from different banks
- Advertizing (e.g. Amazon)

# Web Services Components

- XML-RPC
- SOAP
- WSDL
- UDDI
- XML
- JSON

# XML-RPC

- protocol for exchanging information
- uses XML messages to perform RPCs
- Requests are encoded in XML and sent via HTTP POST
- XML responses are embedded in the body of the HTTP response
- platform-independent
- allows diverse applications to communicate
  - e.g. A Java client can speak XML-RPC to a Perl server
- Easy to get started with web services

# SOAP

- communication protocol
- format for sending messages
- designed to communicate via Internet
- platform independent
- language independent
- simple and extensible
- allows you to get around firewalls
- SOAP version 1.2 is W3C standard

# WSDL

- stands for Web Services Description Language
- was developed jointly by Microsoft and IBM
- An XML based protocol for information exchange in decentralized and distributed environments
- WSDL definition describes how to access a web service and what operations it will perform
- is a language for describing how to interface with XML-based services
- is an integral part of UDDI, an XML-based worldwide business registry

# Universal Description, Discovery, and Integration UDDI

- is an XML-based standard for
  - describing, publishing, and finding web services
- is a specification for a distributed registry of WS
- Is platform independent, open framework
- can communicate via SOAP, CORBA, and Java RMI Protocol
- uses WSDL to describe interfaces to web services
- is an open industry initiative enabling businesses to discover each other and define how they interact over the Internet

# References

- Books:
  - “Service Oriented Architecture: Concepts, Technology and Design” by Thomas Erl, 1st Edition, Pearson publication
  - “Understanding SOA with Web Services” by Greg Lomow and Eric Newcomer, 1st Edition, Pearson Publication

# References

- Web:
  - https://www.tutorialspoint.com/webservices/