# Messaging

Unit - IV

# Introduction

- Communication between services is message-based & should be standardized (same format and transport protocol)
- SOA has message-centric application design
- Business and application logic is embedded into messages
- Messaging is responsible for initiating service-oriented automation
- So, demands flexible and highly extensible messaging framework
- SOAP specification supports all the above needs
- Also extended with sophisticated message structure to support enterprise applications

# Contd...

- SOAP communications framework for SOA creates intelligence-heavy and self-sufficient
- For loosely coupled environment – message independence, robustness and extensibility is needed

# SOAP Envelope, header, and body

- Every SOAP message is enclosed in an envelope
- In each message header is optional, but rarely omitted because of its importance
- message body / payload contains the actual message content in XML format

# Header Blocks

- Message independence is implemented through *header blocks*
- *It stores meta information stored in* envelope's header area.
- It contains all info required for services
- Using this info' it delivers and process the message contents
- This alleviates services storing message-specific logic
- It further reinforces the characteristics of contemporary SOA

- **Web services designed with generic processing functionality driven by various types of meta information**

# Types of Features embed in Header

- processing instructions executed by service intermediaries or the ultimate receiver
- routing or workflow information associated with the message
- security measures
- reliability rules related to the delivery of the message
- context and transaction management information
- correlation information

# Faults and Nodes

- SOAP messages can add exception handling logic by providing an optional fault section present in body area
- It stores error condition information as simple message, when an exception occurs
- **Nodes**
- Web services require communications infrastructure to process and manage exchange of SOAP messages
- Each platform has its own implementation of SOAP communication server and its own software
- programs that services use to transmit and receive SOAP messages are referred to as SOAP nodes

# Message Style

- SOAP specification was originally designed to replace proprietary RPC protocols
- SOA relies on document-style messages
- It enables larger payloads, coarser interface operations, and reduced message transmission between services

- **Attachments**
- To facilitate data other than XML, soap attachments are used
- Different encoding mechanism used to bundle data in its native format with a SOAP message
- Commonly used to transport binary files, such as images

# Contd...

- Irrespective of different platforms, vendor-neutral communications framework is preserved
- It guarantees SOAP message sent by a SOAP node received and processed by another SOAP node
- **SOAP intermediaries**
- Moves messages between service provider and service requestor
- Soap Nodes – forwarding or active
- Forwarding - relaying contents of a message to a subsequent SOAP node
- Alter header block information relating to the forwarding logic

# Contd…

- Active intermediary nodes – does processing beyond forwarding-related functions
- Processing logic  not limit to rules and instructions in the header blocks
- It can alter existing header blocks, insert new ones, and execute a variety of supporting actions

- **Message paths**
- Message path consists of at least one initial sender, one ultimate receiver, and zero or more intermediaries
- Modelling message paths becomes important
- Message path should consider performance, security, context management, and reliable messaging