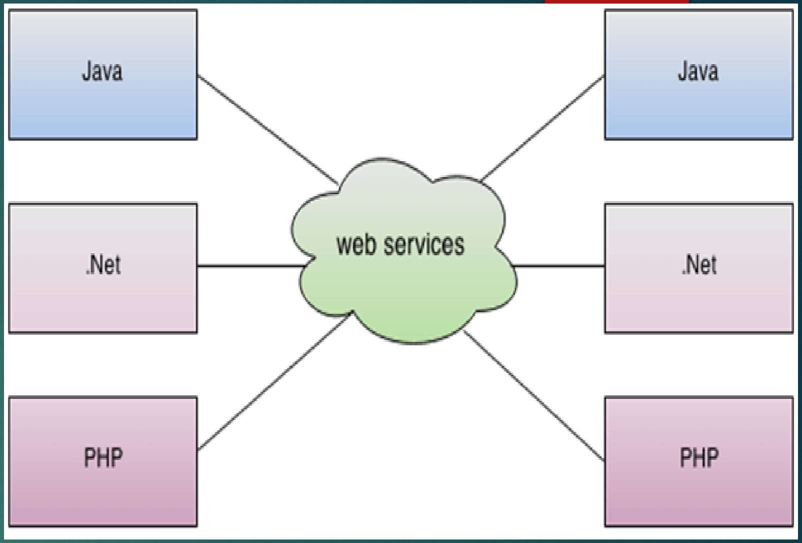
Web services paper-5

USCS505

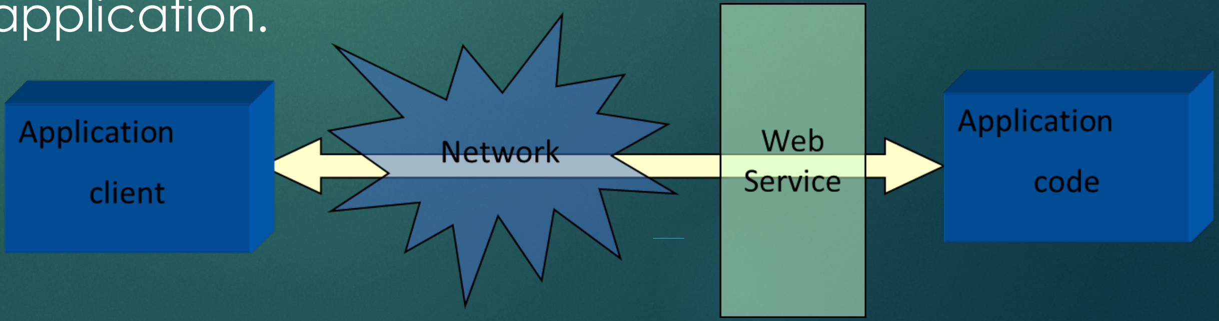
Unit 1 Web services basics

- ❑ **WHAT ARE WEB SERVICES**?
- ❑ TYPES OF WEB SERVICES
- ❑ DISTRIBUTED COMPUTING INFRASTRUCTURE
- ❑ OVERVIEW OF XML
- ❑ SOAP
- ❑ BUILDING WEB SERVICES WITH JAX-WS
- ❑ REGISTERING AND DISCOVERING WEB SERVICES
- ❑ SERVICE ORIENTED ARCHITECTURE
- ❑ WEB SERVICES DEVELOPMENT LIFE CYCLE
- ❑ DEVELOPING AND CONSUMING SIMPLE WEB SERVICES ACROSS PLATFORM

Web Service ?



- It is a client-server application or application component for communication.
- The method of communication between two devices over the network.
- It is a software system for the interoperable machine to machine communication.
- It is a collection of standards or protocols for exchanging information between two devices or application.



Why WS?

- inability to develop highly inter-active applications
- inability to provide complete customizable applications
- inability to integrate applications
- resulted in monolithic architectures, highly fragile, customer-specific, non-reusable integration of applications based on tight coupling principles.
- software-as-a-service
- The new architecture allows for loosely coupled asynchronous interactions on the basis of eXtensible Markup Language (XML) standards with the intention of making access to, and communications between, applications over the Internet easier.
- Web services paradigm allows the software-as-a-service concept to expand to include the delivery of complex business processes and transactions as a service
- use of Web services provides a more flexible solution for ASPs.The core of the application – the business and data components – remains on the ASP'smachines, but is now accessed programmatically via Web services interfaces.

Features of WS

- XML-Based
- Loosely Coupled
- Coarse-Grained
- Ability to be Synchronous or Asynchronous
- Supports Remote Procedure Calls (RPCs)
- Supports Document Exchange

Web services Vs Web-based applications

1. human intervention, high degree of flexibility and adaptability
2. modular, self-aware, and self-describing applications
3. more visible and manageable
4. brokered or auctioned(cost, speed, degree of security)

Check your understanding:

- _______________has highest priority

a. Reusability

b. Interoperability

c. Both

d. None

Amazon provides a web service that provides prices for products sold online via amazon.com. The front end or presentation layer can be in .Net or Java but either programming language would have the ability to communicate with the web service.

a. True

b. False

Types of Web Services

- 2 flavours simple/informational and complex

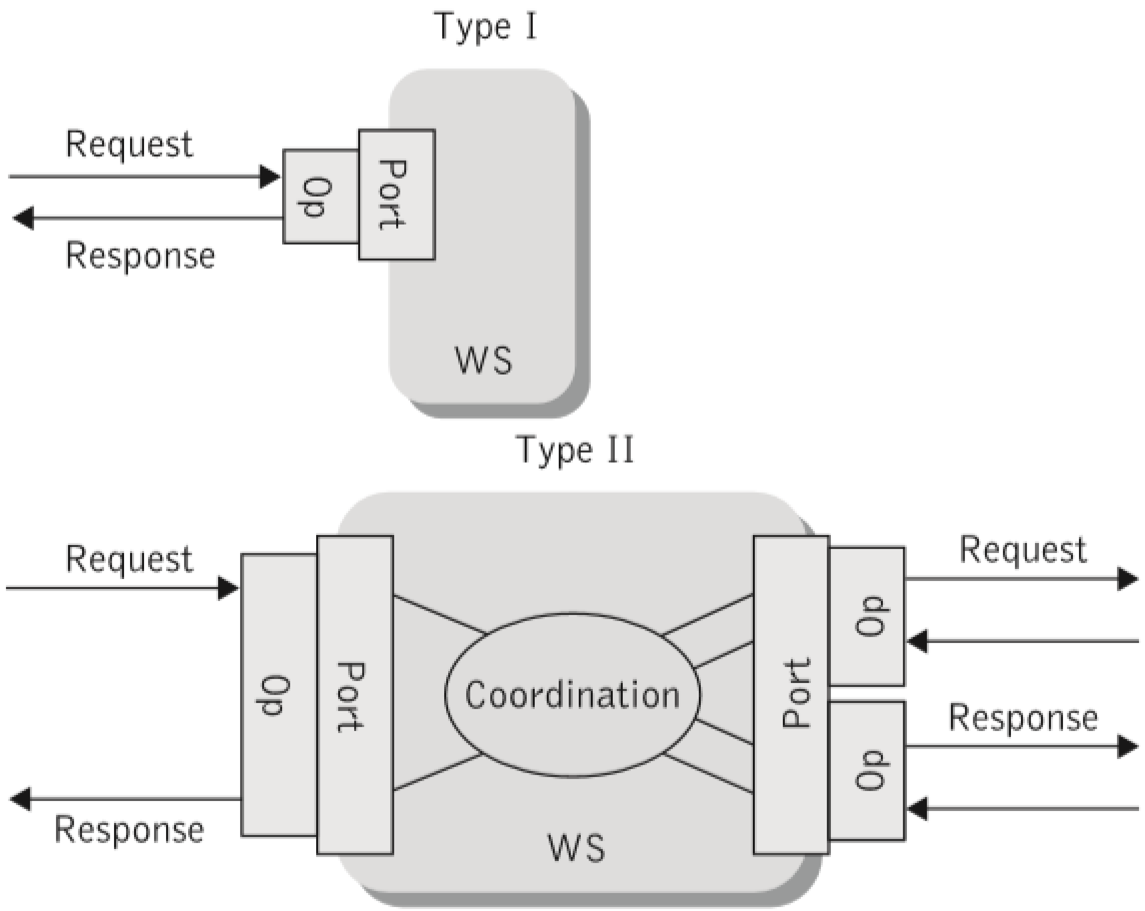
Informational services

- Simple request/response and wait for request
- Programmatic access to content and stateless
- Expose backend applications to other applications
- Access function call by executing a web service through WSDL
- Can be divided in 3 subcategories

1. Pure content services:-weather report, financial info, news,etc
2. Simple trading services:-complicated form of informational services

seamless aggregation of information including backend eg;-logistics

3. Simple syndication services: value added information eg:-ecommerce webs



High-level view of informational and complex services

Complex services

- Can use an atomic service to accomplish a specific business task such as billing or credit checking
- Beyond informational services
- Compose several web services
- Typically involve assembly and invocation of many pre-existing services
- Use of diverse enterprises to complete a multi-step business interaction that requires coordination
- 2 types
- Complex services that compose programmatic web services
- Complex services that compose interactive web services

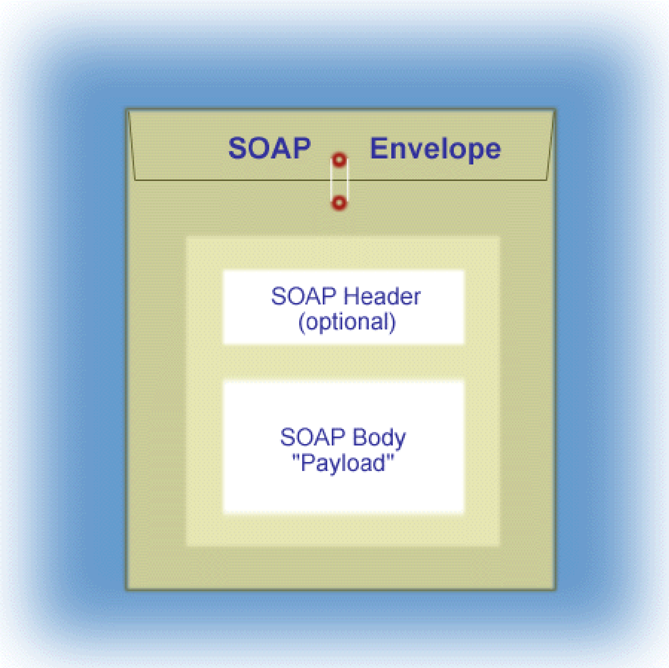
Web Service Components

- SOAP
- WSDL
- UDDI
- An application allows other applications to connect to it over the Web using SOAP
- This Web Service exposes its methods in a WSDL file
- The Web Service is published in a UDDI registry to allow other businesses to find it

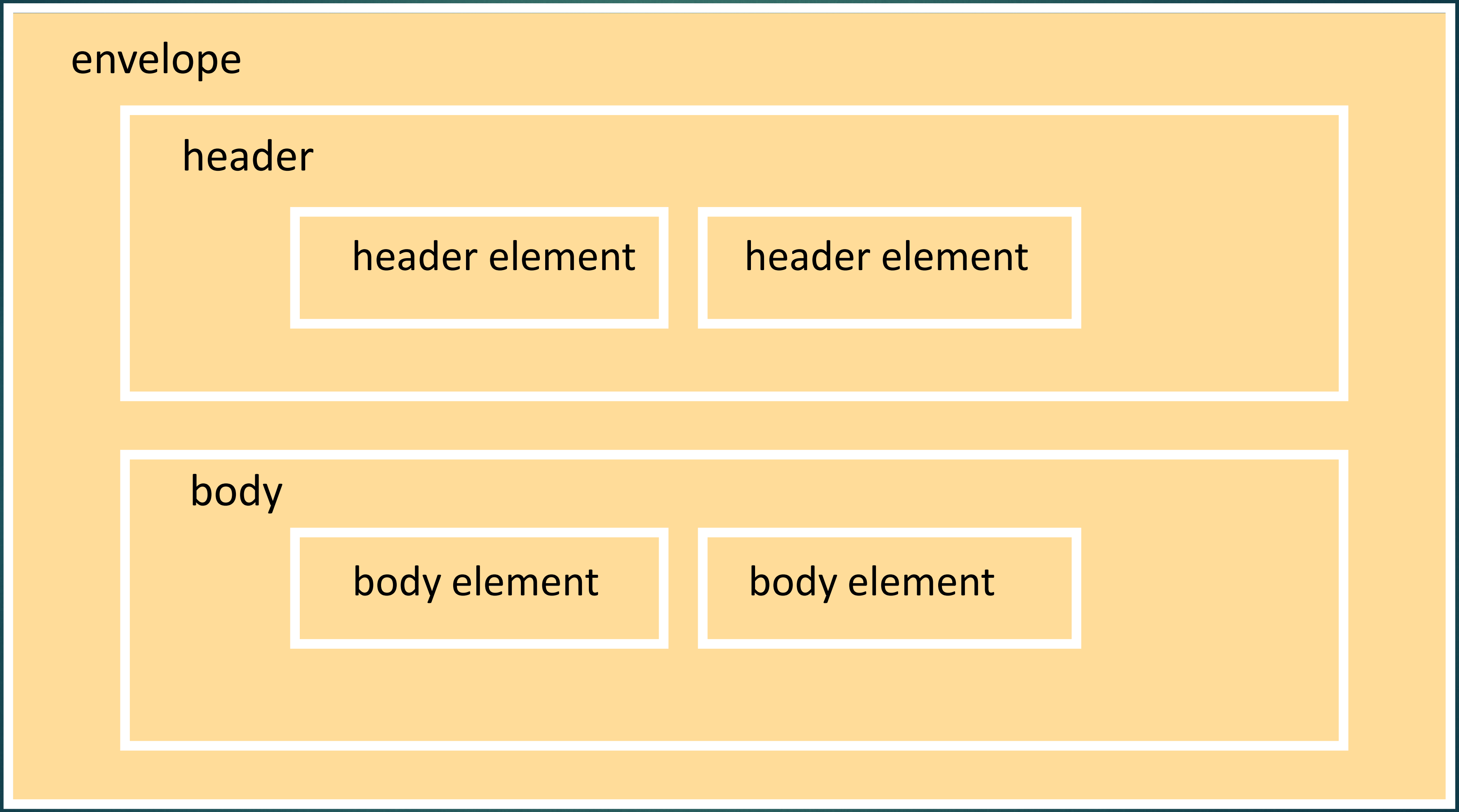
SOAP

- Simple Object Access Protocol
- SOAP is designed to enable both client-server and synchronous interaction over the Web
- Web services communicate using W3C standard SOAP messages.
- SOAP formalizes the use of XML as a way to pass data (therefore can be Objects) from one process to another.
- Originally SOAP was based only on HTTP, but other transport protocols such as TCP are also allowed.

```
<env:Envelope
xmlns:env="http://www.w3.org/2001/12/soap-envelope">
  <env:Body>
    <ns:Order xmlns:ns="urn:it.uu.se:Students">
      <item>Bill</item>
      <item>Bob</item>
      <item>Tony</item>
    </ns:Students>
  </env:Body>
</env:Envelope>
```



SOAP MESSAGES



envelope

header

header element

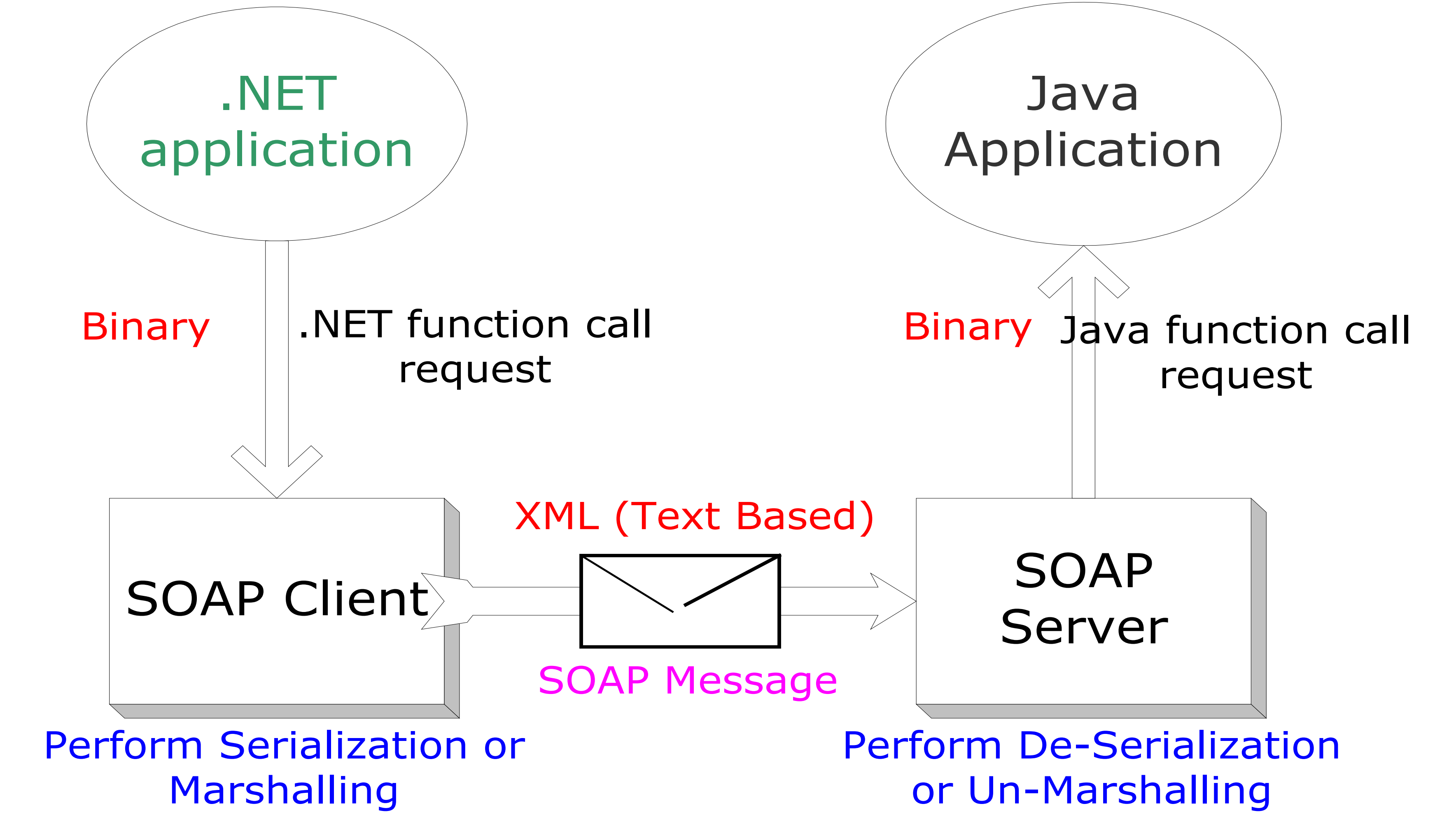
header element

body

body element

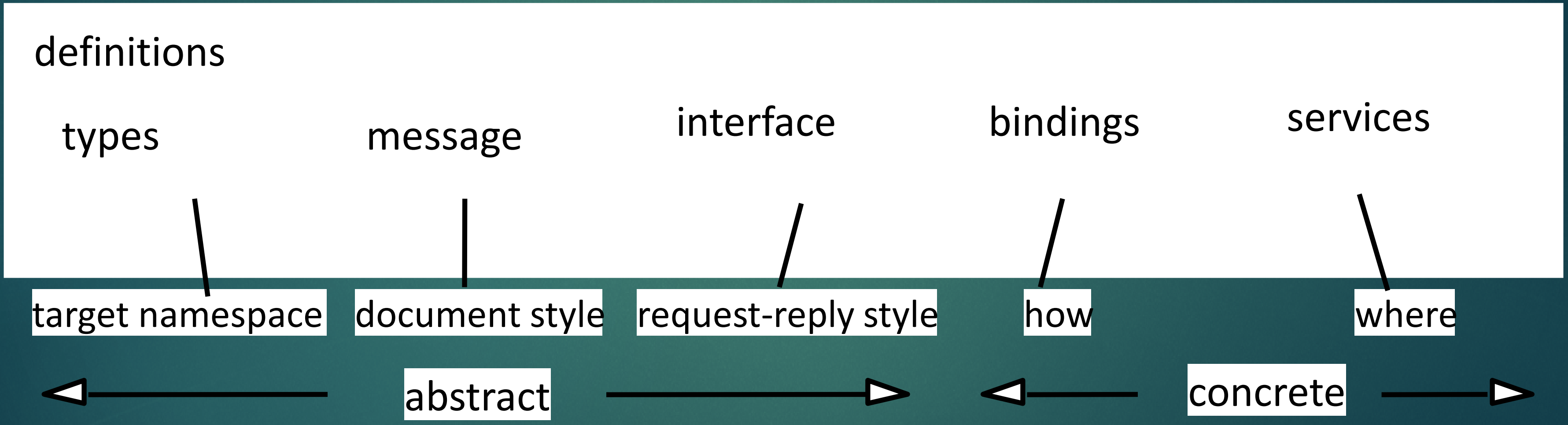
body element

Transport of SOAP Messages


.NET application
Java Application
Binary
.NET function call request
Binary
Java function call request
XML (Text Based)
SOAP Client
SOAP Server
SOAP Message
Perform Serialization or Marshalling
Perform De-Serialization or Un-Marshalling

Service Descriptions and IDL for Web Services

- Interface definitions are needed for clients to communicate with services.
- *Service description* specifies two characteristics – how the message are to be communicated and the URI of service.
- Web Services Description Language (WSDL)



definitions

types

message

interface

bindings

services

target namespace

document style

request-reply style

how

where

abstract

concrete

WSDL

- WSDL has a well-defined XML vocabulary to answer the following questions regarding the web service involved:
- What does the service do?
  - Both in machine and human-readable forms
- What language does the service speak?
  - The format/data structure of the message exchanged
- How does the client talk to the service?
  - HTTP/SMTP/FTP
- Where is the location of the web service?
  - The access point (URL)

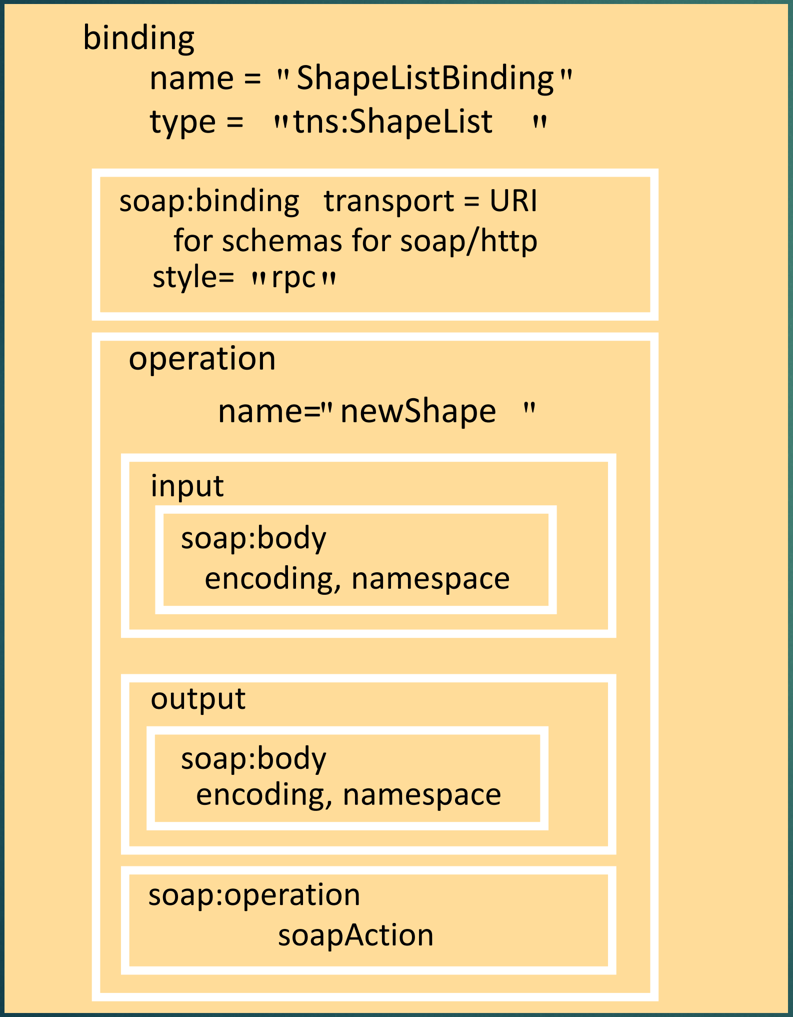
WSDL – INTERFACE

Name	Client	Server	Delivery	Fault message
In-Out	Request	Reply		may replace Reply
In-Only	Request			no fault message
Robust In-Only	Request		guaranteed	may be sent
Out-In	Reply	Request		may replace Reply
Out-Only		Request		no fault message
Robust Out-Only		Request	guaranteed	may send fault

Message exchang

WSDL – CONCRETE PART

- Binding (choice of protocols) and Service (choice of endpoint or sever address):



binding
name = "ShapeListBinding"
type = "tns:ShapeList "

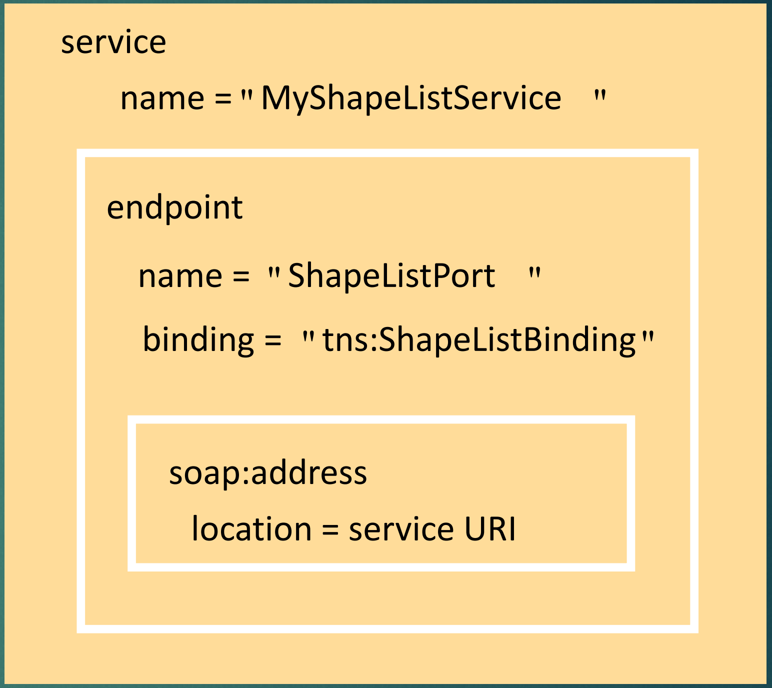
soap:binding transport = URI
for schemas for soap/http
style= "rpc"

operation
name=" newShape "

input
soap:body
encoding, namespace

output
soap:body
encoding, namespace

soap:operation
soapAction



service
name = " MyShapeListService "

endpoint
name = " ShapeListPort "
binding = "tns:ShapeListBinding"

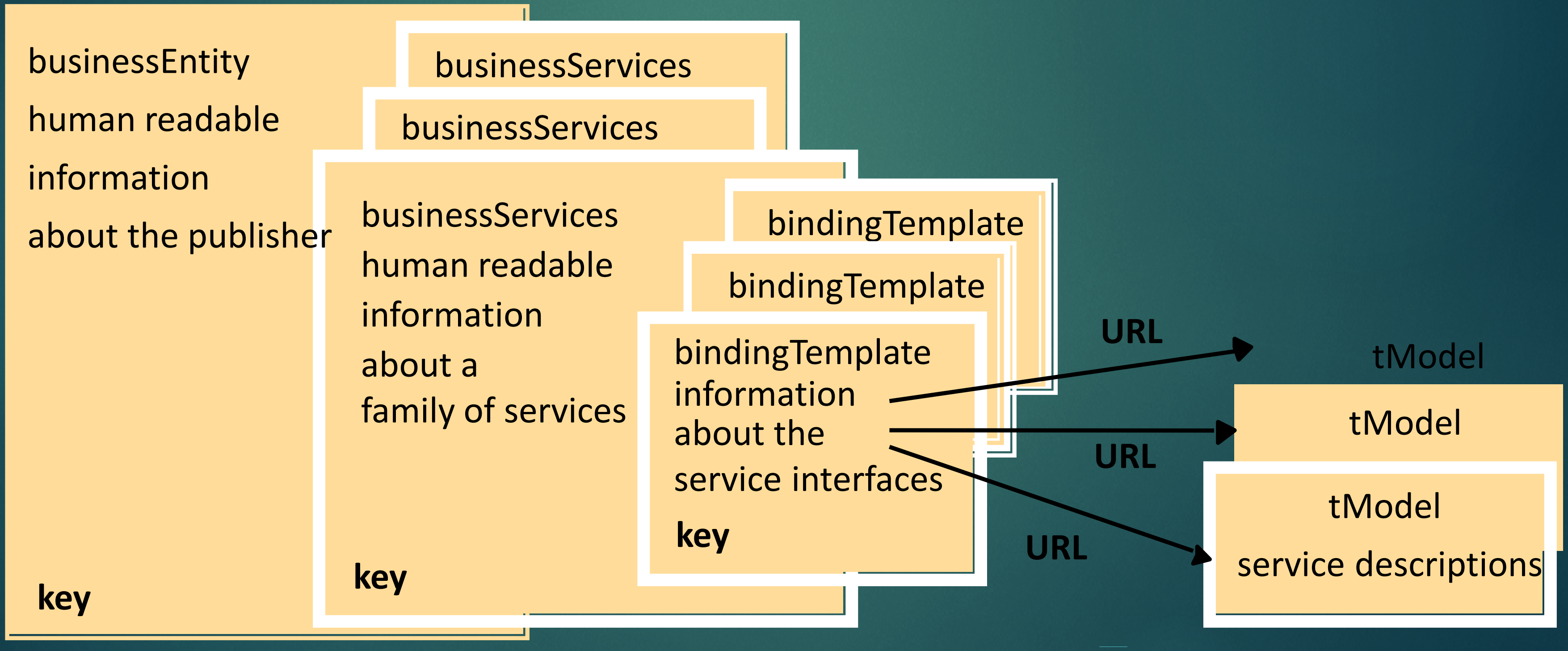
soap:address
location = service URI

the service URI is:

“http://localhost:8080/ShapeList-jaxrpc/ShapeList”

A DIRECTORY SERVICE FOR USE WITH WEB SERVICES

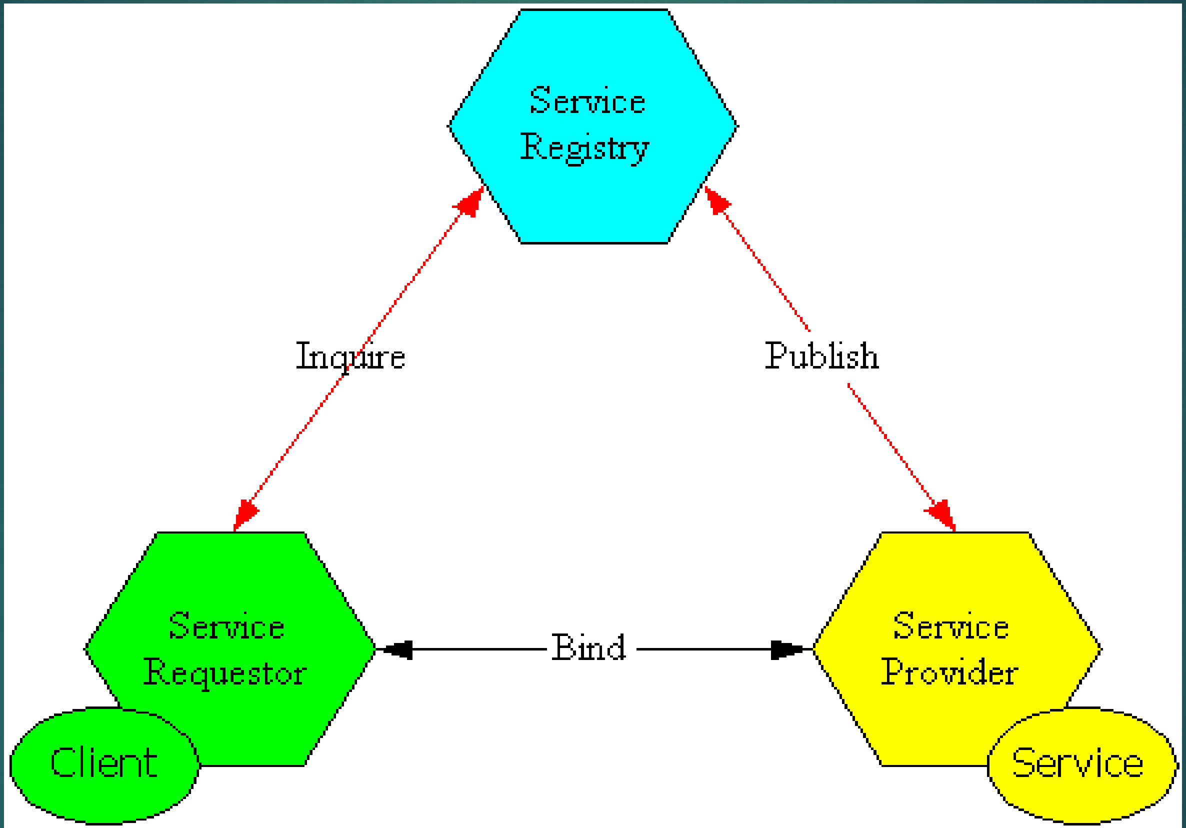
- UDDI – Universal Description, Discovery, and Integration Service
- Data structures allow human-readable information access



A DIRECTORY SERVICE FOR USE WITH WEB SERVICES

- Lookup
  - UDDI provides an API for looking up services based on 2 sets of query operation: *get_xxx, find_xxx*.
  - UDDI provides a notify/subscribe interface
- Publication
  - UDDI provides an interface for publishing and updating information about web services.
- Registries
  - UDDI service is based on replicated data stored in registries

UDDI



UDDI defines a way to publish and discover information about Web services