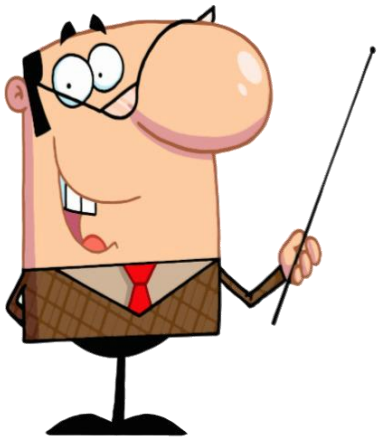
Web Services

10

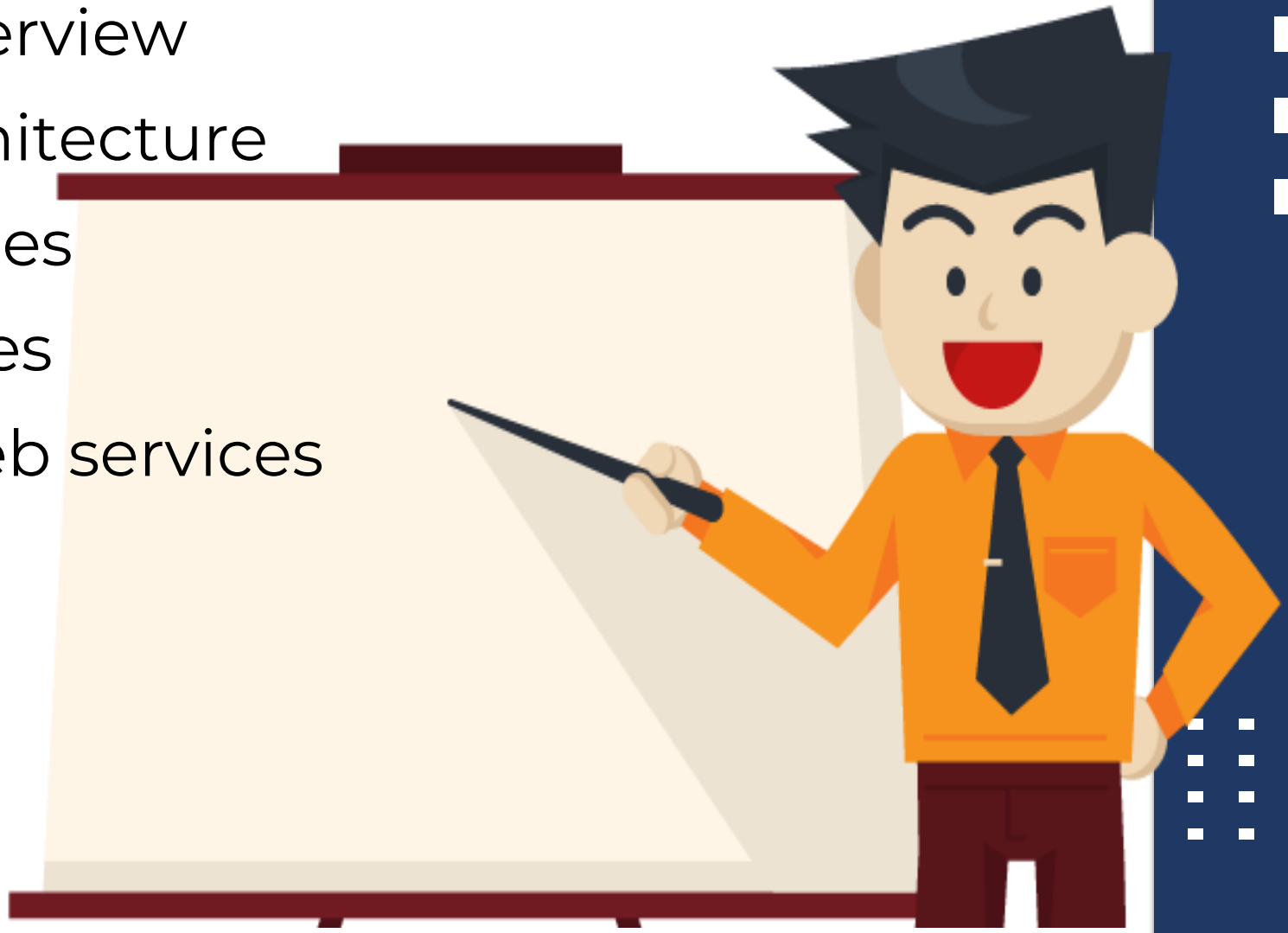
Web Technology



Asst. Prof. Manop Phankokkruad, Ph.D.
Faculty of Information Technology
King Mongkut's Institute of Technology Ladkrabang

Outline

1. Web services Overview
2. Web Service Architecture
3. SOAP Web services
4. REST Web services
5. Accessing the web services



Web Services

A **web service** is a network accessible interface to application functionality, built using standard Internet technologies.

- Clients of web services do NOT need to know how it is implemented.
- A web service is a set of methods exposed through a web interface.
- Service model assumes services are always available
- Web services facilitate the automated transfer of information from Computer-to-Computer using the XML language.

Service Oriented Architecture(SOA)

- Framework provides a set of fundamental operations via web services
  - May also provide local services using Windows services
- All applications based on that framework share the common services
  - Don’t have to recreate the same functionality for each new application
- Can provide those same services to Partner businesses, suppliers, and customers.

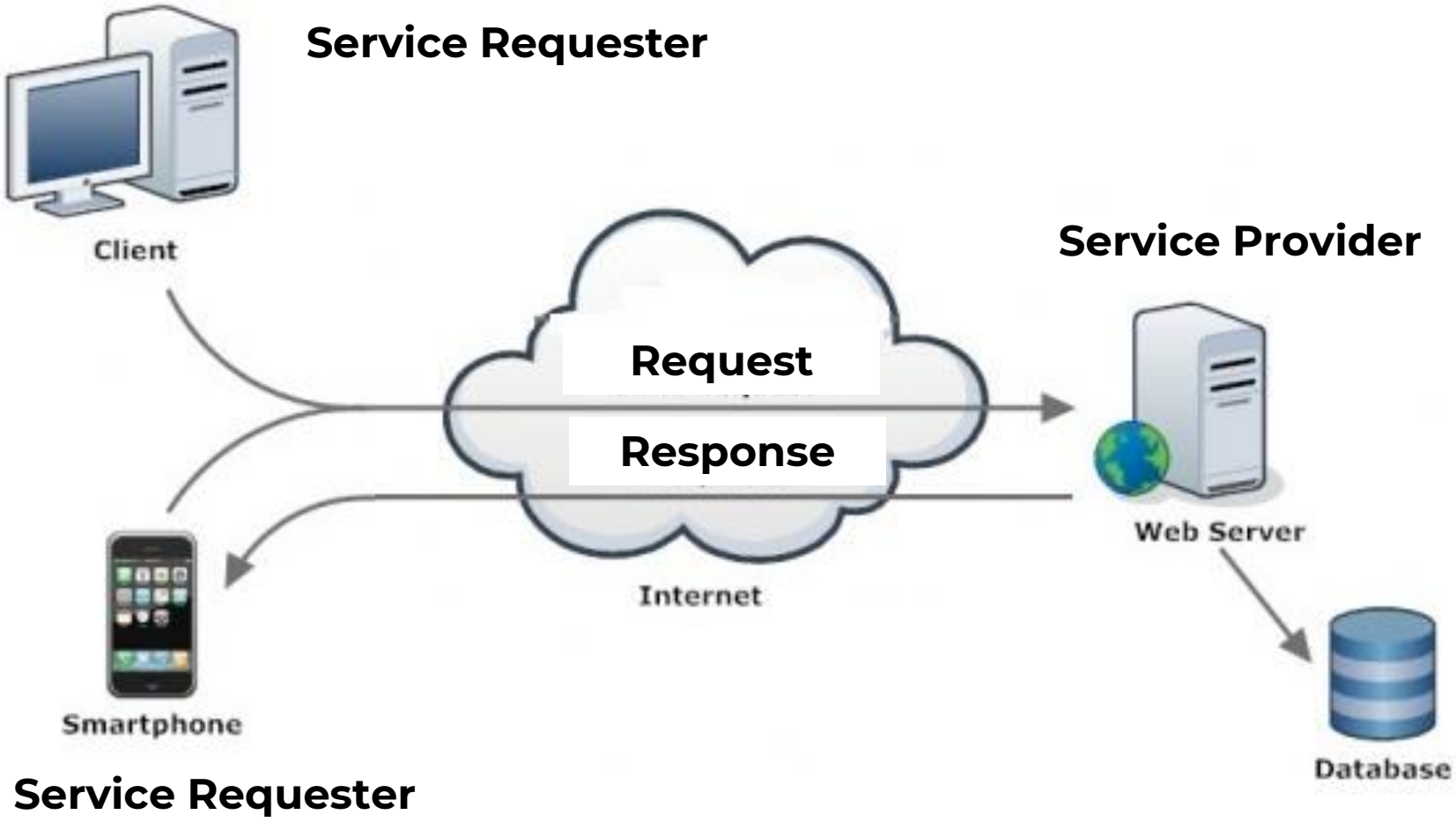
Web Service Architecture

The Web Services architecture is based upon the interactions between three roles:

- **Service provider** implements the service and makes it available on the Internet.
- **Service requestor** utilizes an existing web service by opening a network connection and sending an XML request.
- **Service registry** provides a central place where developers can publish new services or find existing ones. It serves as a centralized trade center for their services.

Web Service Architecture

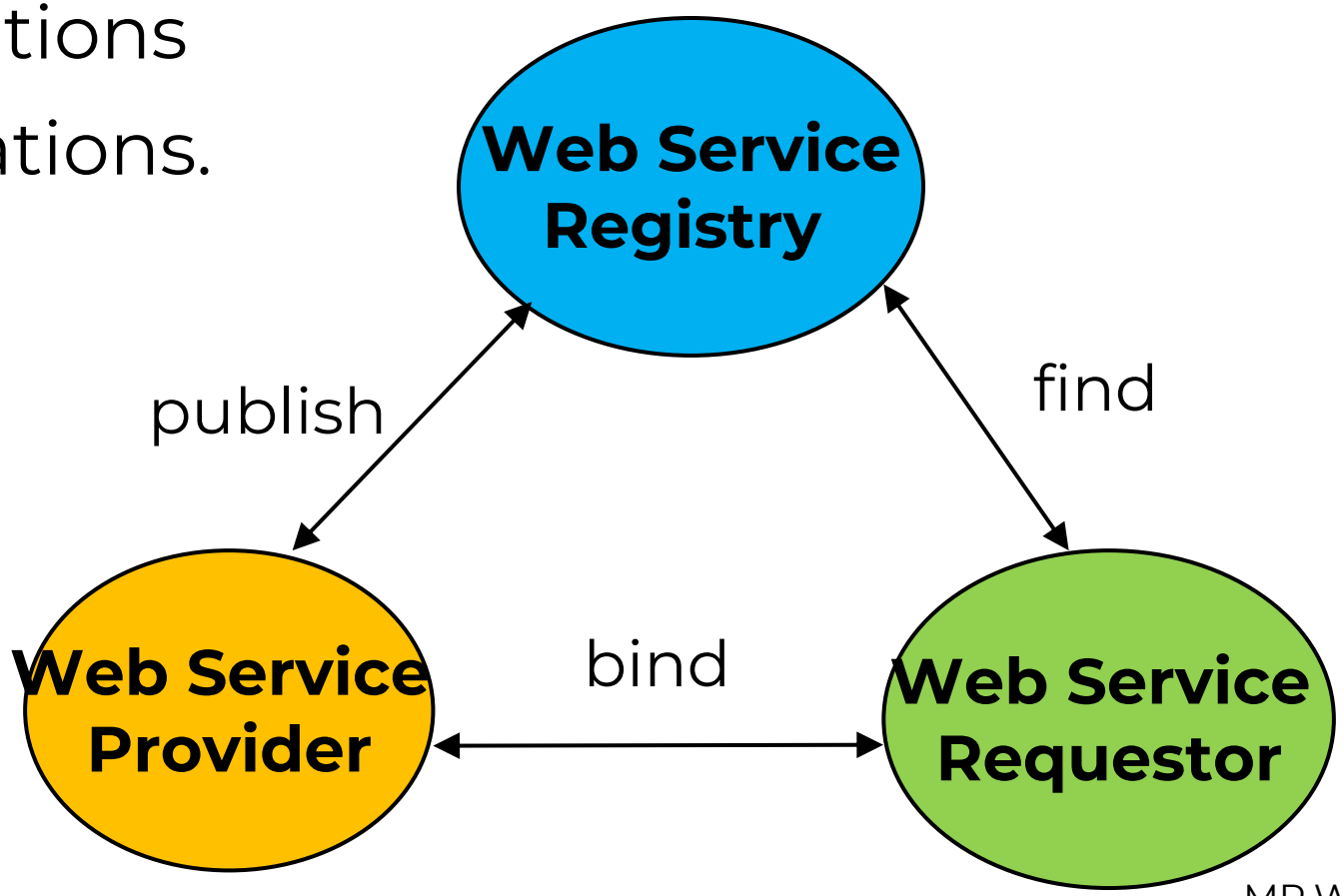
Web Service Architecture



Web Service Architecture

The interactions involve the:

- Publish operations
- Find operations
- Bind operations.



Web Service Architecture

The Web Services architecture.

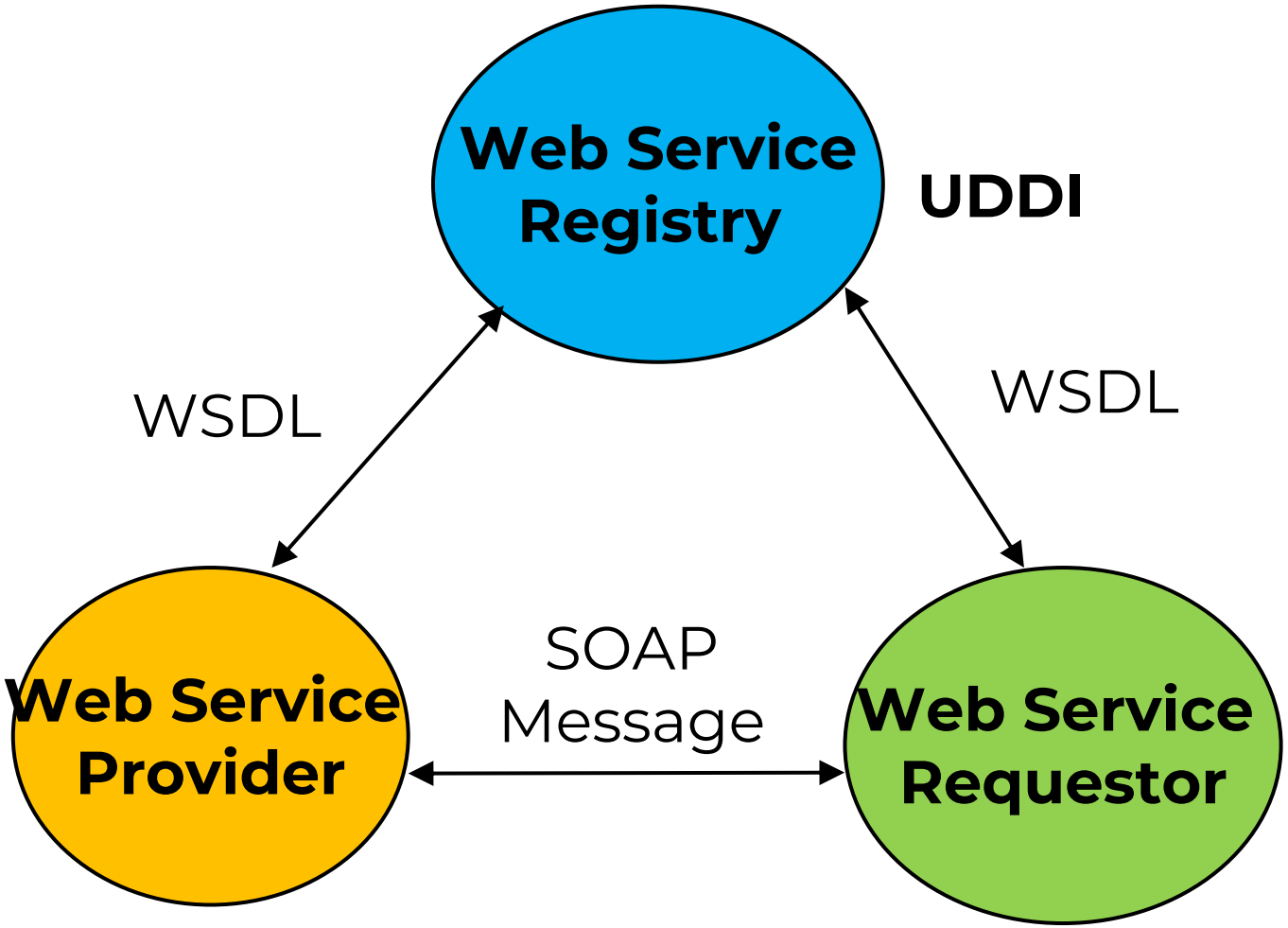
1. Service providers **publish** services by advertising service descriptions in the registry such as UDDI, and WSDL.
2. Service requestors use **find** operation to retrieve service descriptions from the service registry and show operations to see their details to assess their appropriateness.
3. Service requestors **bind** to service providers using binding information found in service descriptions to locate and invoke a service.

WSDL & UDDI

- **WSDL (Web Services Description Language)** is an XML vocabulary for describing Web services. It allows developers to describe Web Services and their capabilities, in a standard manner. **WSDL** describes message contents, WSDL defines where the service is available and what communications protocol is used to talk to the service.
- **UDDI (Universal Description, Discovery and Integration)** is a directory for storing information about web services , like yellow pages. **UDDI** is a directory of web service interfaces described by WSDL.

Web Service Model

Web service interactions

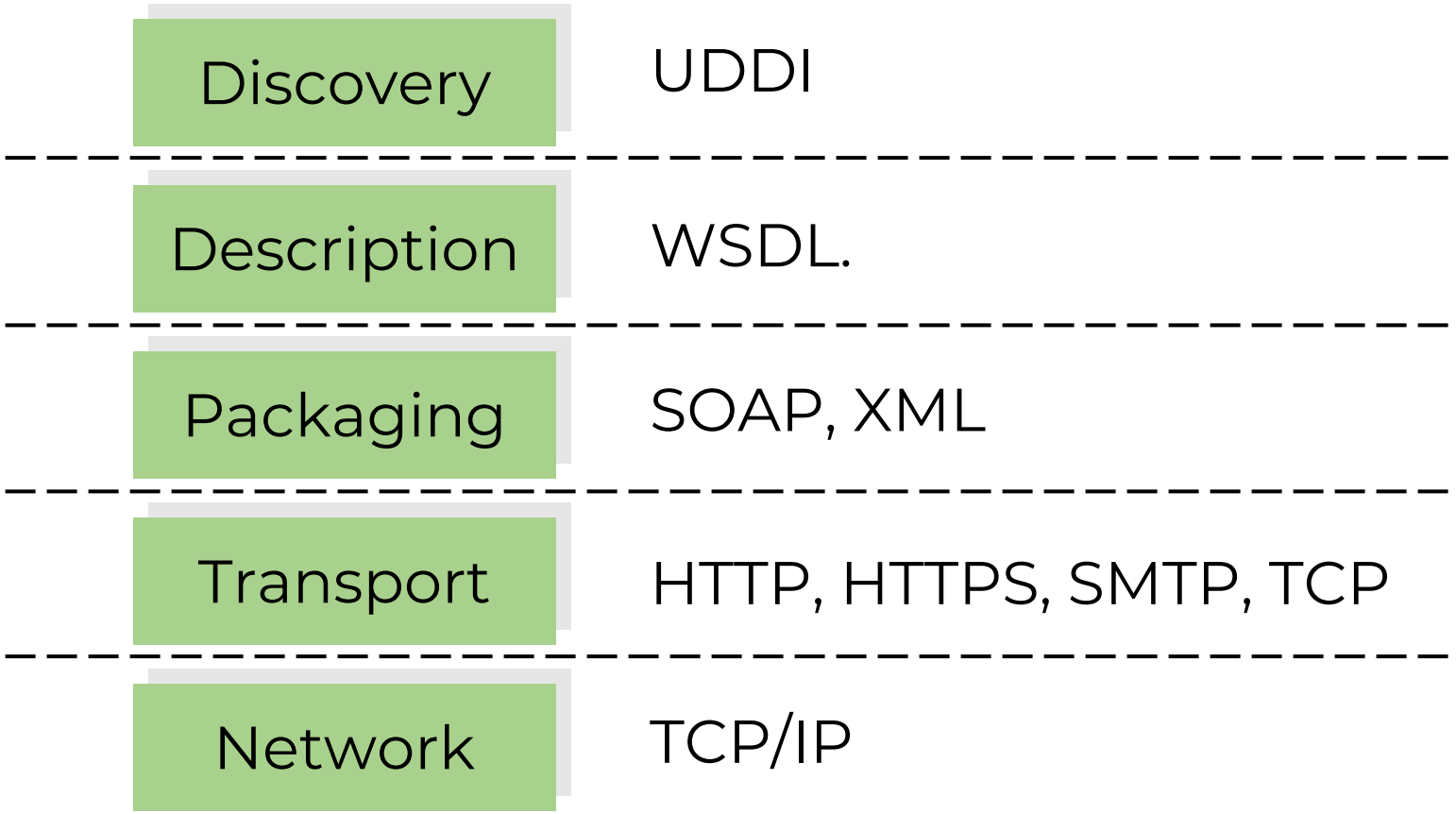


WSDL example

```
<message name="GetStockPriceRequest">
  <part name="stock" type="xs:string"/>
</message>
<message name="GetStockPriceResponse">
    <part name="value" type="xs:string"/>
</message>
```

```
<portType name="StocksRates">
      <operation name="GetStockPrice">
          <input message="GetStockPriceRequest"/>
          <output message="GetStockPriceResponse"/>
      </operation>
</portType>
```

Web Service Technology Stack



Web Service Protocols

Web services are based on four protocols:

1. Web Service Description Language (WSDL) specifies the interface of the Web service.
2. Discovery Protocol (DISCO)
    - Pointer to all web services on a particular web site.
    - Defines a discovery document format and a protocol for retrieving the discovery document, enabling developers to discover services at a known URL.

Web Service Protocols

Web services are based on four protocols(cont.):

3. Extensible Markup Language (XML) defines complex data structures.
4. Universal Description, Discovery, and Integration (UDDI)
    - Central repository of web service descriptions.
    - Define a registry service for Web services.

Web Service Approaches

- There is a division in the web community about the best method for accessing resources over the internet.
- The two leading methods are
  - **SOAP**, a standards-based approach.
  - **REST**, a non standards-based approach.

SOAP

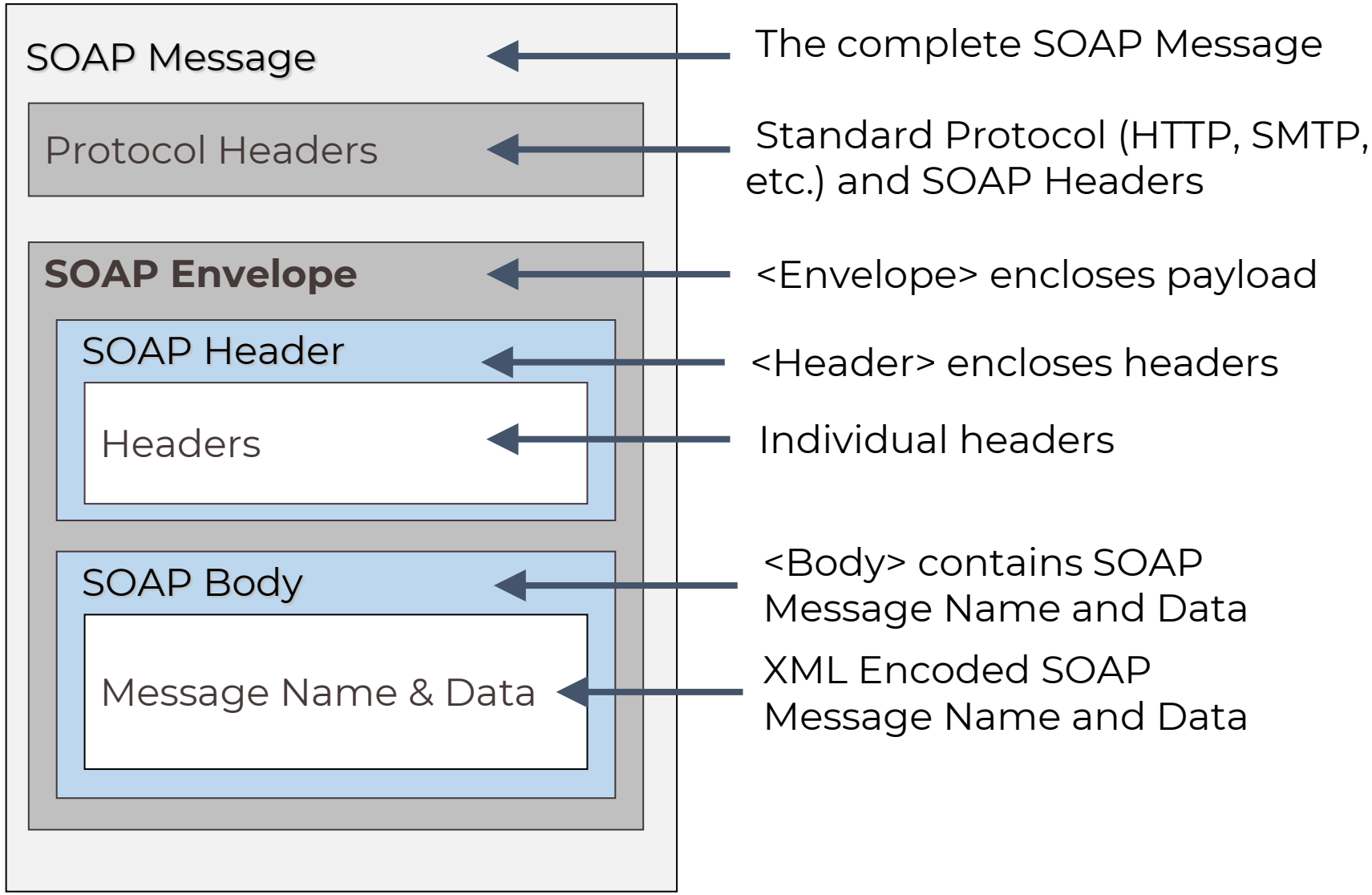
Simple Object Access Protocol (**SOAP**) is a Standards Web Services that expose useful functionality to Web users through a standard Web protocol.

- SOAP encoding is done in XML, using XML Schema and relying heavily on XML namespaces.
- SOAP uses mainly HTTP as a transport protocol. That is, HTTP message contains a SOAP message as its payload section.
- Works with any operating system, any programming language, and any platform.

SOAP

- SOAP is used for passing documents, Electronic Document Interchange (EDI), and Remote Procedure Calls (RPC).
- SOAP is an envelope that a request for resources is packaged in. It includes a header and body.
  - The header can include information about security, recipients, handling, routing, etc.
  - The body includes a standard XML request

What is a SOAP Message?



Simple SOAP Request

POST /StockQuote HTTP/1.1
Host: www.stockquoteserver.com
Content-Type: text/xml
Content-Length: 323
SOAPAction: “www.stockquoteserver.com/GetLastTradePrice”

HTTP Request

```
<?xml version=“1.0” encoding=“utf-8”?>

<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">

  <SOAP-ENV:Body>
    <m:GetLastTradePrice xmlns:m="Some-Namespace-URI">
      <symbol>DIS</symbol>
    </m:GetLastTradePrice>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Simple SOAP Response

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8

Content-Length: nnnn

HTTP Response

```
<?xml version="1.0" encoding="utf-8"?>

<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">

 <SOAP-ENV:Body>
  <m:GetLastTradePriceResponse
        xmlns:m="Some-Namespace-URI">
   <Price>24.5</Price>
  </m:GetLastTradePriceResponse>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

REST

> **REST**(Representational State Transfer) is intended to evoke an image of how a well-designed Web application behaves: a network of web pages, where the user progresses through an application by selecting links, resulting in the next page being transferred to the user and rendered for their use.

- REST is a Non-Standards Based Approach.
- REST is an architectural style for accessing distributed media or resources.

REST style

- REST architectures consist of clients and servers.
  - Clients initiate requests to servers who process these requests and return responses based on these requests.
  - These requests and responses are built around the transfer of representations of these resources.
- Each resource is first identified using a URL and a new resource for every service required is created. The data returned by the service must be linked to the other data.

REST style

- An application or architecture considered RESTful or REST-style is characterized by:
  - State and functionality are divided into distributed resources.
- Every resource is uniquely addressable using a uniform and minimal set of commands (typically using HTTP commands of GET, POST, PUT, or DELETE over the Internet)
- The protocol is client/server, stateless, layered, and supports caching

Implementation details

There are several key implementation details with HTTP that you should be aware of :

1. **Resources** – REST uses addressable resources to define the structure of the API. These are the URLs you use to get to pages on the Web.
   - Resources are first-class objects, so “object” is a subtype of “resource”.
   - Resources are retrieved not as character strings but as complete representations
   - A web page is a representation of a resource.

Implementation details

There are several key implementation details with HTTP that you should be aware of (cont.):

2. **Request Verbs** – These describe what you want to do with the resource. A browser typically issues a GET verb to instruct the endpoint it wants to get data, however there are many other verbs available including things like POST, PUT and DELETE.

REST and HTTP

- REST is a post hoc description of the Web
- HTTP 1.1 was designed to conform to REST
- Its methods are defined well enough to get work done
- HTTP is the most RESTful protocol
- It's possible to apply REST concepts to other protocols and systems.

Summary of REST and SOAP

SOAP	REST
▪ Messages are represented in a standardized XML SOAP "envelope" .	▪ Messages are represented in plain XML / JSON.
▪ Can be bound to various protocols including HTTP and SMTP .	▪ HTTP is used for the transfer protocol.
▪ Access to and manipulation of data are application specific.	▪ HTTP verbs are used for access/manipulation commands.
▪ Security is not described by SOAP and is to be provided by the developer.	▪ URIs are used to uniquely identify resources in message.
▪ XML schemas or DTD are used to define the contract between client and service.	▪ HTTP authentication provides security.
	▪ There is no formal method for expressing the interface contract.

Accessing the web services

- REST is a way of accessing the web services. REST as an architecture style does not require processing and is simpler and more flexible than SOAP.
- **REST API** is used to fetch or give some information from a web service.
- The REST based web services can give output in any format like CSV, JSON, XML and RSS. JSON is the most common output format of REST API in a form that's easy to parse within the language for the application.

Accessing the web services

- In RESTful services, URIs are used to access the resources. So, data and functions are called resources in the RESTful glossary.
- To get the book details.

https://openlibrary.org/api/books?bibkeys=ISBN:0452278902&format=json

Response

{"ISBN:0452278902": {"bib_key": "ISBN:0452278902", "preview": "borrow", "thumbnail_url": "https://covers.openlibrary.org/b/id/8231558-S.jpg", "preview_url": "https://archive.org/details/greenmilenovelin00king", "info_url": "https://openlibrary.org/books/OL658734M/The_green_mile"}}

How useful are Web Services?

Web services: Some possibilities

- ❑ Financial information (e.g., stock quotes)
- ❑ Sports information
- ❑ Weather information
- ❑ News
- ❑ Delivery status
- ❑ Tax and shipping calculations
- ❑ Any data that is relevant to the client

More Information

- http://www.w3schools.com/
- http://uddi.microsoft.com/Default.aspx
- http://www.developer.com/services/article.php/2195981
- Web service https://en.wikipedia.org/wiki/Web_service

- Many more on the Web…