

## 2.1 Representational State Transfer (REST)

- ❑ APIs are for software components; a way for software to interact with other software.
- ❑ Web Services are a set of rules and technologies that enable two or more components on the web to talk to each other.
- ❑ Not every API is a web service.
- ❑ REST API is a web service.
- ❑ REST API is an API that follows the rules of REST specification.
- ❑ A web service is defined by rules:
  - a) How software components will talk?
  - b) What kind of messages they will send to each other?
  - c) How requests and responses will be handled?

### HTTP and REST

- ❑ HTTP is an application layer protocol for sending and receiving messages over a network.
- ❑ REST is a specification that dictates how distributed systems on the web should communicate.
- ❑ REST is a way to implement and use the HTTP protocol.

### REST and Systems of Systems

- ❑ SOA focuses on loosely coupled software applications running across different administrative domains, based on common protocols and technologies, such as HTTP and XML.
- ❑ SOA is related to early efforts on the architecture style of large scale distributed systems, particularly Representational State Transfer (REST).
- ❑ REST still provides an alternative to the complex standard-driven web services technology.
- ❑ Used in many Web 2.0 services.

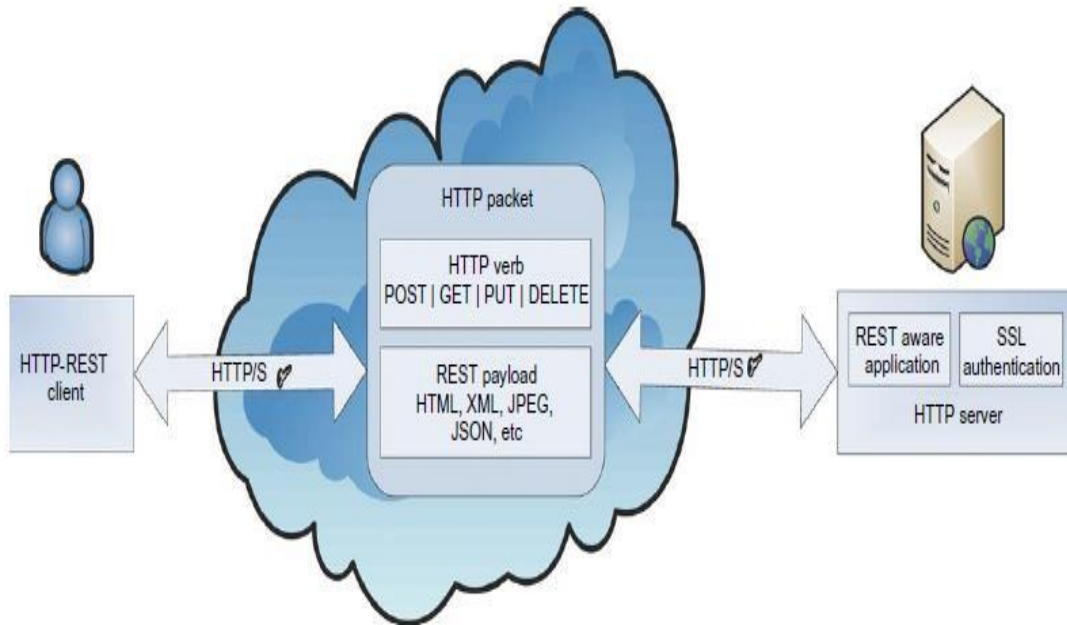
- REST is a software architecture style for distributed systems, particularly distributed hypermedia systems, such as the World Wide Web.

**Applications:**

Google, Amazon, Yahoo, Facebook and Twitter

**Advantage:**

- Simplicity
- Ease of being published and consumed by clients.



**Figure 2.5 A simple REST interaction between user and server in HTTP specification**

**REST Principles**

The REST architectural style is based on four principles:

- Resource Identification through URIs
- Uniform, Constrained Interface
- Self-Descriptive Message

## □ Stateless Interactions

### Resource Identification through URIs

- The RESTful web service exposes a set of resources which identify targets of interaction with its clients.
- The key abstraction of information in REST is a resource.
- Any information that can be named can be a resource, such as a document or image or a temporal service.
- A resource is a conceptual mapping to a set of entities.
- Each particular resource is identified by a unique name, or more precisely, a Uniform Resource Identifier (URI).
- URI is of type URL, providing a global addressing space for resources involved in an interaction between components as well as facilitating service discovery.
- The URIs can be bookmarked or exchanged via hyperlinks.
- URIs provide more readability and the potential for advertisement.

### Uniform, Constrained Interface

- Interaction with RESTful web services is done via the HTTP standard, client/server cacheable protocol.
- Resources are manipulated using a fixed set of four CRUD (create, read, update, delete) verbs or operations:
  - PUT
  - GET
  - POST
  - DELETE
- **PUT** creates a new resource.
- The resource can then be destroyed by using **DELETE**.
- **GET** retrieves the current state of a resource.

- **POST** transfers a new state onto a resource.

### **Self-Descriptive Message**

- A REST message includes enough information to describe how to process the message.
- This enables intermediaries to do more with the message without parsing the message contents.
- In REST, resources are decoupled from their representation so that their content can be accessed in a variety of standard formats

Eg:- HTML, XML, MIME, plain text, PDF, JPEG, JSON, etc.

- REST provides multiple/alternate representations of each resource.
- Metadata about the resource is available and can be used for various purposes.
  - ❖ Cache control
  - ❖ Transmission error detection
  - ❖ Authentication or authorization
  - ❖ Access control.

### **Stateless Interactions**

- The REST interactions are “stateless”
- Message does not depend on the state of the conversation.
- Stateless communications improve visibility, reliability and increases scalability
- Decrease network performance by increasing the repetitive data

### **REST - Advantages**

- RESTful web services can be considered an alternative to SOAP stack or “big web services
- Simplicity
- Lightweight nature
- Integration with HTTP

**REST Architectural Elements**

REST Elements	Elements	Example
<b>Data elements</b>	Resource	The intended conceptual target of a hypertext reference
	Resource identifier	URL
	Representation	HTML document, JPEG image, XML, etc.
	Representation metadata	Media type, last-modified time
	Resource metadata	Source link, alternates, vary
	Control data	If-modified-since, cache-control
<b>Connectors</b>	Client	libwww, libwww-perl
	Server	libwww, Apache API, NSAPI
	Cache	Browser cache, Akamai cache network
	Resolver	Bind (DNS lookup library)
	Tunnel	SSL after HTTP CONNECT
<b>Components</b>	Origin server	Apache httpd, Microsoft IIS
	Gateway	Squid, CGI, Reverse Proxy
	Proxy	CERN Proxy, Netscape Proxy, Gauntlet
	User agent	Netscape Navigator, Lynx, MOMspider

**2.2 Web Services**

- Web service is often referred to a self-contained, self-describing, modular application designed to be used and accessible by other software applications across the web.
- This allows client software to dynamically determine what a service does, the data types that a service uses, how to invoke operations on the service, and the responses that the service may return.
- Once a web service is deployed, other applications and other web services can discover and invoke the deployed service.
- A web service is defined as a software system designed to support interoperable machine-to-machine interaction over a network
- A web service has an interface described in a machine-executable format (specifically Web Services Description Language or WSDL).
- Web services are remotely executed, they do not depend on resources residing on the client system that calls them.

- Other systems interact with the web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization

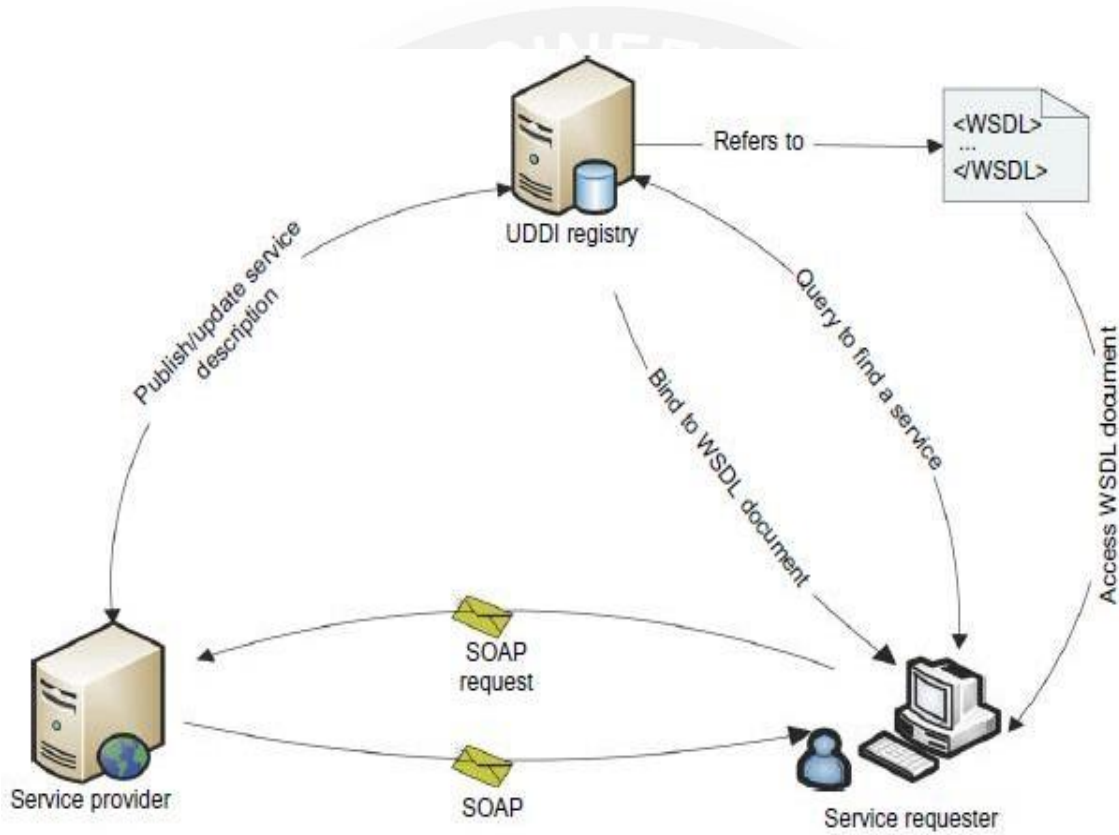


Figure 2.6 Web Services

### Web Services- Technologies

- Simple Object Access Protocol (SOAP)
- Web Services Description Language (WSDL)
- Universal Description, Discovery and Integration (UDDI)

### Simple Object Access Protocol (SOAP)

- SOAP provides a standard packaging structure for transmission of XML documents over various Internet protocols, such as SMTP, HTTP, and FTP.
- A SOAP message consists of a root element called envelope. Envelope contains a header: a container that can be extended by intermediaries.

- Additional application-level elements such as routing information, authentication, transaction management, message parsing instructions and Quality of Service (QoS) configurations are also included.
- Body element that carries the payload of the message.
- The content of the payload will be marshaled by the sender's SOAP engine and unmarshaled at the receiver side, based on the XML schema that describes the structure of the SOAP message.

### **Web Services Description Language (WSDL)**

WSDL describes the interface, a set of operations supported by a web service in a standard format.

- It standardizes the representation of input and output parameters of its operations as well as the service's protocol binding, the way in which the messages will be transferred on the wire.
- Using WSDL enables disparate clients to automatically understand how to interact with a web service.

### **Universal Description, Discovery, and Integration (UDDI)**

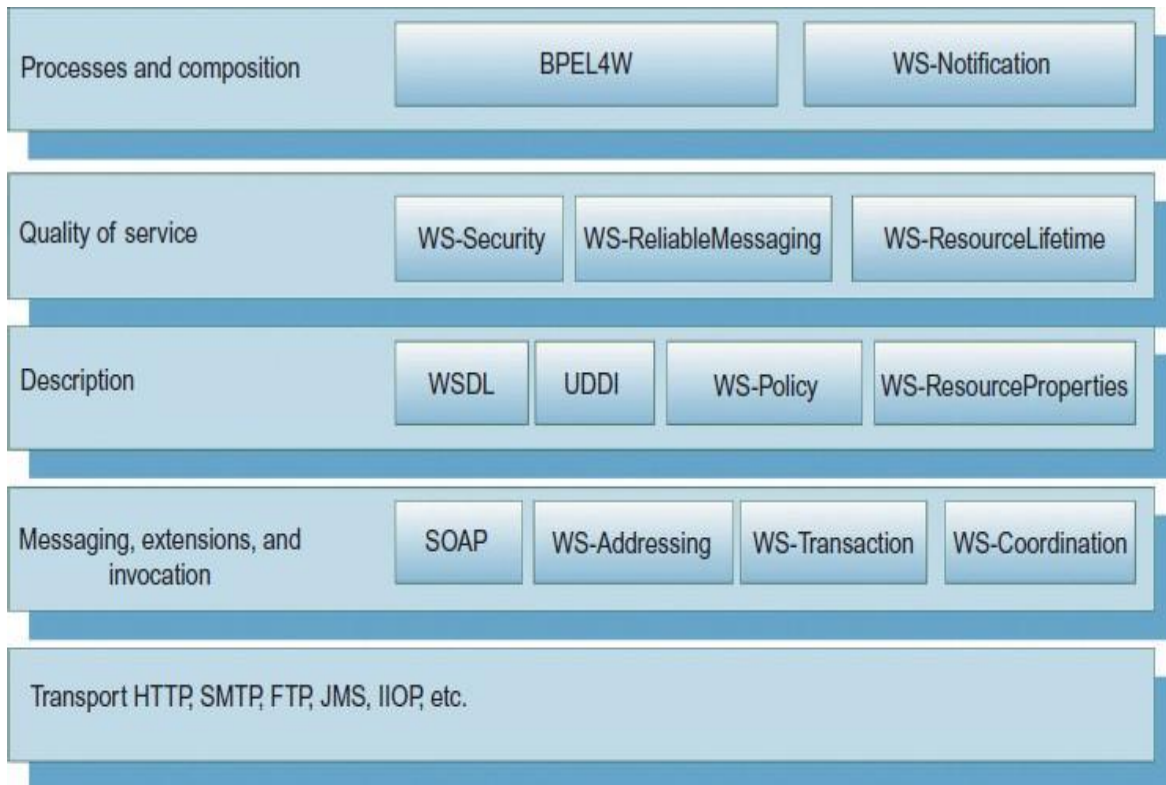
- UDDI provides a global registry for advertising and discovery of web services.
- Performed by searching for names, identifiers, categories or the specification implemented by the web service

SOAP is an extension, and an evolved version of XML-RPC.

A simple and effective remote procedure call protocol which uses XML for encoding its calls and HTTP as a transport mechanism.

- A procedure executed on the server and the value it returns was formatted in XML.
- SOAP mainly describes the protocols between interacting parties
- Data format of exchanging messages is left to XML schema to be handled.





**Figure 2.7 WS-I Protocol Stack**

- **Business Process Execution Language for Web Services (BPEL4WS)**, a standard executable language for specifying interactions between web services.
  - Web services can be composed together to make more complex web services and workflows.
  - BPEL4WS is an XML-based language, built on top of web service specifications, which is used to define and manage long-lived service orchestrations or processes.
  - In BPEL, a business process is a large-grained stateful service, which executes steps to complete a business goal.
  - That goal can be the completion of a business transaction, or fulfillment of the job of a service.
- **Web Service WS-Notification** enables web services to use the publish and subscribe messaging pattern.
- **Web Services Security (WS-Security)** are set of protocols that ensure security for SOAP-based messages by implementing the principles of confidentiality, integrity and authentication.



- **Web Services Reliable Messaging (WS-Reliable Messaging)** describes a protocol that allows **messages** to be delivered reliably between distributed applications in the presence of software component, system, or network failures
- **WS-ResourceLifetime** specification standardizes the means by which a WS-Resource can be destroyed.
- **WS-Policy** is a specification that allows web services to use XML to advertise their **policies** (on security, quality of service, etc.) and for web service consumers to specify their **policy** requirements.
- **WS-ResourceProperties** defines a standard set of message exchanges that allow a requestor to query or update the property values of the WS-Resource.
- **WS-Addressing** is a specification of transport-neutral mechanism that allows web services to communicate addressing information.
- **WS-Transaction WS-Transaction** is a specification developed that indicates how transactions will be handled and controlled in Web Services.
- The transaction specification is divided into two parts - short atomic transactions (AT) and long business activity (BA).
- **Web Services Coordination (WS-Coordination)** describes an extensible framework for providing protocols that coordinate the actions of distributed applications.
- **The Java Message Service (JMS) API** is a messaging standard that allows application components based on the Java Platform Enterprise Edition (Java EE) to create, send, receive, and read messages.
- **Internet Inter-ORB Protocol (IIOP)** is an object-oriented protocol
- Used to facilitate network interaction between distributed programs written in different programming languages.
- IIOP is used to enhance Internet and intranet communication for applications and services.

**Table 5.3** Sample SOAP Request-Response for Creating an S3 Bucket

SOAP Request	SOAP Response
<pre> &lt;soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap- envelope" soap:encodingStyle= "http://www.w3.org/2001/12/soap-encoding"&gt; &lt;soap:Body&gt; &lt;CreateBucket xmlns="http://doc.s3.amazonaws .com/2010-03-15"&gt; &lt;Bucket&gt;SampleBucket&lt;/Bucket&gt; &lt;AWSAccessKeyId&gt; 1B9FVRAYCP1VJEXAMPLE= &lt;/AWSAccessKeyId&gt; &lt;Timestamp&gt;2010-03-15T14:40:00.165Z &lt;/Timestamp&gt; &lt;Signature&gt;luyz3d3P0aTou39dzbqaEXAMPLE =&lt;/Signature&gt; &lt;/CreateBucket&gt; &lt;/soap:Body&gt; &lt;/soap:Envelope&gt;                     </pre>	<pre> &lt;soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap- envelope" soap:encodingStyle= "http://www.w3.org/2001/12/soap-encoding"&gt; &lt;soap:Body&gt; &lt;CreateBucket xmlns="http://doc.s3.amazonaws .com/2010-03-15"&gt; &lt;Bucket&gt;SampleBucket&lt;/Bucket&gt; &lt;AWSAccessKeyId&gt;1B9FVRAYCP1VJEXAMPLE= &lt;/AWSAccessKeyId&gt; &lt;Timestamp&gt;2010-03-15T14:40:00.165Z &lt;/Timestamp&gt; &lt;Signature&gt;luyz3d3P0aTou39dzbqaEXAMPLE =&lt;/Signature&gt; &lt;/CreateBucket&gt; &lt;/soap:Body&gt; &lt;/soap:Envelope&gt;                     </pre>

- A SOAP message consists of an envelope used by the applications to enclose information that need to be sent.
- An envelope contains a header and a body block.
- The EncodingStyle element refers to the URI address of an XML schema for encoding elements of the message.
- Each element of a SOAP message may have a different encoding, but unless specified, the encoding of the whole message is as defined in the XML schema of the root element.
- The header is an optional part of a SOAP message that may contain auxiliary information.
- The body of a SOAP request-response message contains the main information of the conversation, formatted in one or more XML blocks.
- In example, the client is calling CreateBucket of the Amazon S3 web service interface.
- In case of an error in service invocation, a SOAP message including a Fault element in the body.

**Table 5.4** The 10 Areas Covered by the Core WS-\* Specifications

WS-* Specification Area	Examples
1. Core Service Model	XML, WSDL, SOAP
2. Service Internet	WS-Addressing, WS-MessageDelivery, Reliable WSRM, Efficient MOTM
3. Notification	WS-Notification, WS-Eventing (Publish-Subscribe)
4. Workflow and Transactions	BPEL, WS-Choreography, WS-Coordination
5. Security	WS-Security, WS-Trust, WS-Federation, SAML, WS-SecureConversation
6. Service Discovery	UDDI, WS-Discovery
7. System Metadata and State	WSRF, WS-MetadataExchange, WS-Context
8. Management	WSDM, WS-Management, WS-Transfer
9. Policy and Agreements	WS-Policy, WS-Agreement
10. Portals and User Interfaces	WSRP (Remote Portlets)

