



BITS Pilani
Pilani Campus

Network Programming

K Hari Babu
Department of Computer Science & Information Systems



BITS Pilani
Pilani Campus



Web Services

- Web services evolved from previous technologies that served the same purpose such as RPC, ORPC (DCOM, CORBA and JAVA RMI).
- Web Services were intended to solve three main problems:
 - Interoperability
 - Firewall traversal
 - Complexity

- Earlier distributed systems suffered from interoperability issues because each vendor implemented its own on-wire format for distributed object messaging.
- Examples
 - SUN-RPC is bound to C language
 - Development of DCOM apps strictly bound to Windows Operating system.
 - Development of RMI bound to Java programming language.

Distributed Object models



- DCOM is a Protocol that enables software components to communicate directly over a network in a reliable, secure, and efficient manner. Previously called OLE, DCOM is designed for use across multiple network transports, including Internet Protocols such as HTTP.
- RMI is an RPC mechanism enabling Java programmers to create distributed applications, in which the methods of remote Java objects can be invoked from another JVM, possibly on a different host.

Firewall traversal



- Collaboration across corporations was an issue because distributed systems such as CORBA and DCOM used non-standard ports.
- Web Services use HTTP as a transport protocol and most of the firewalls allow access through port 80 (HTTP), leading to easier and dynamic collaboration.

Complexity



- Web Services is a developer-friendly service system.
- Most of the above-mentioned technologies such as RMI, COM, and CORBA involve a whole learning curve.
- New technologies and languages have to be learnt to implement these services.

Web Service definition



A simple definition:

“a Web Service is an application component accessible over open protocols”.

Web Service Definition



- A software component that can be
 - **Described** using a service-description language, which
 - is in formal XML notation,
 - covers all the details necessary to interact with the service (message formats for operations, transport protocols and location), and
 - hides the implementation details of the service;
 - **Published** to a registry of services;
 - **Discovered** through a standard mechanism;
 - **Invoked** through a declared API, usually through a network; and
 - **Composed** with other services
 - enabling loosely coupled, component-oriented, cross-technology application implementations.

Programming Interfaces and APIs



- API = Application Programming Interface
- Defines a set of functions that may be called by application programs
 - Like a library
 - But application developer may not have access to the code implementing the functions
 - E.g., being an eBay developer means you can use the API, not that you can see how it is implemented
 - And the functions may even be executed on a different computer!
 - E.g., eBay developers are using functions located on eBay servers, not their own computers

Example: eBay's API



Alternative to interacting with the web site

- Functions provided for
 - Listing items
 - Tracking a particular user's auctions
 - Leaving feedback
 - *Everything eBay users can do*
- Allows application developer to provide custom interface to users, with extra features
 - Built on top of eBay functionality

Web Service definition revisited



- A more precise definition:
 - an application component that:
 - Communicates via open protocols (HTTP, SMTP, etc.)
 - Processes XML messages framed using SOAP
 - Describes its messages using XML Schema
 - Provides an endpoint description using WSDL
 - Can be discovered using UDDI

Web Services Components



- **XML** – eXtensible Markup Language – A uniform data representation and exchange mechanism.
- **SOAP** – Simple Object Access Protocol – A standard way for communication.
- **UDDI** – Universal Description, Discovery and Integration specification – A mechanism to register and locate WS based application.
- **WSDL** – Web Services Description Language – A standard meta language to described the services offered.

Example – A simple Web Service



- A buyer (which might be a simple client) is ordering goods from a seller service.
- The buyer finds the seller service by searching the UDDI directory.
- The seller service is a Web Service whose interface is defined using Web Services Description Language (WSDL).
- The buyer is invoking the order method on the seller service using Simple Object Access Protocol (SOAP) and the WSDL definition for the seller service.
- The buyer knows what to expect in the SOAP reply message because this is defined in the WSDL definition for the seller service.

The Web Service Model

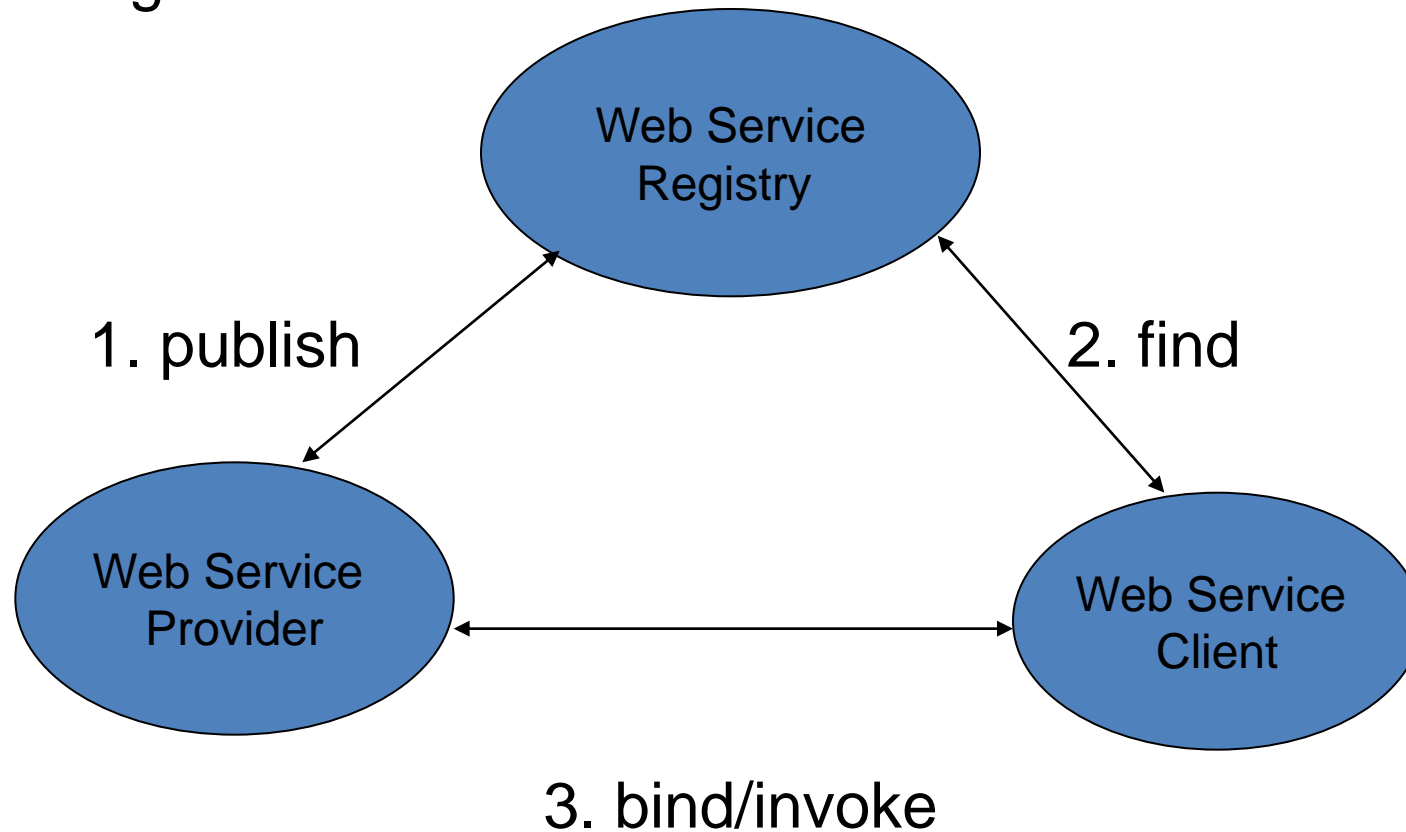


- The Web Services architecture is based upon the interactions between three roles:
 - Service provider
 - Service registry
 - Service requestor
- The interactions involve the:
 - Publish operations
 - Find operation
 - Bind operations.

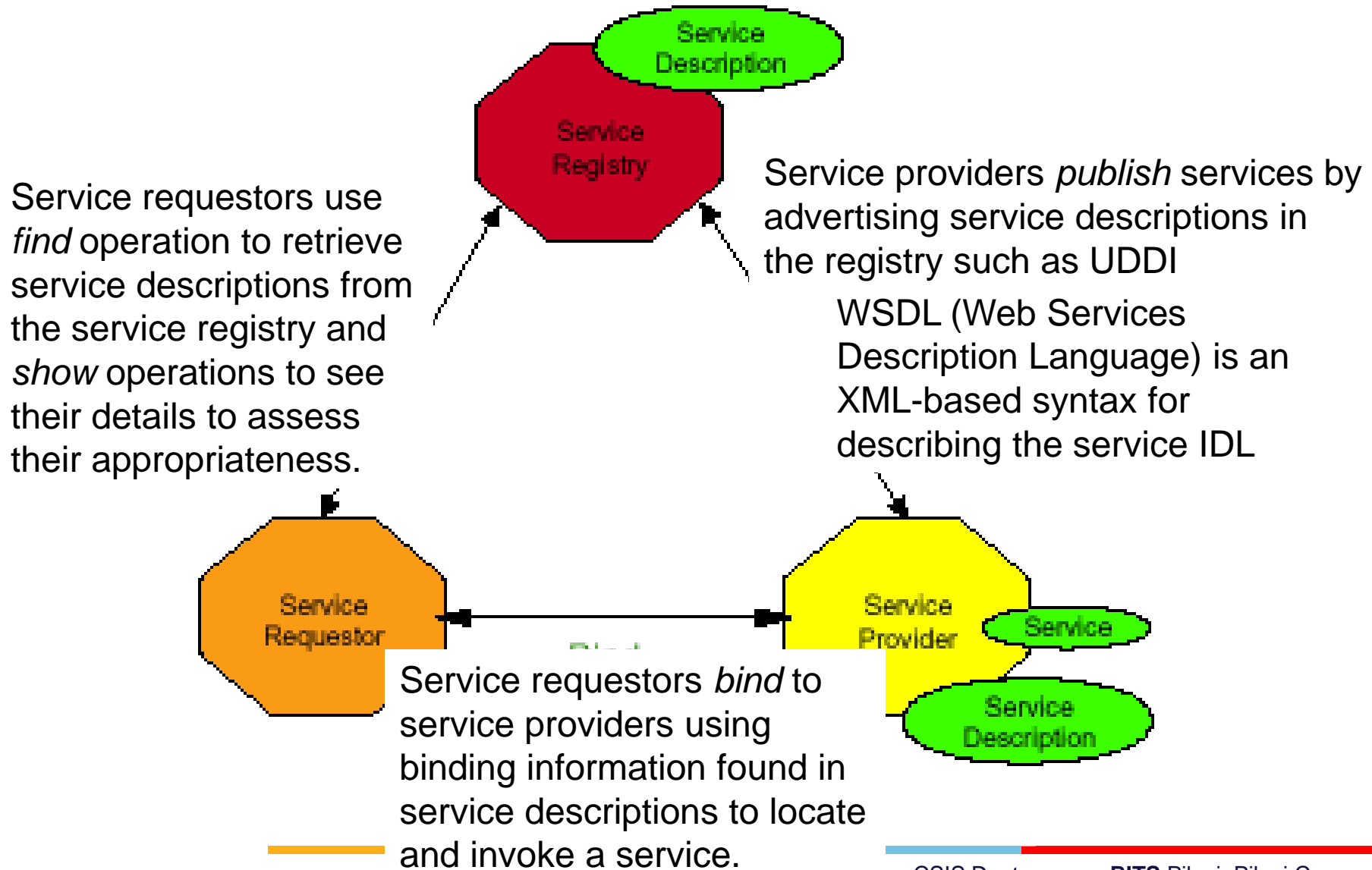
The Web Service Model (cont)



The Web Services model follows the *publish*, *find*, and *bind* paradigm.



Service-Oriented Architecture



- XML stands for **EX**tensible **M**arkup **L**anguage.
- XML is a **markup language** much like HTML.
- XML was designed to **describe data**.
- XML tags are not predefined. You must **define your own tags**.
- The prefect choice for enabling cross-platform data communication in Web Services.

XML vs HTML



An HTML example:

```
<html>
<body>
  <h2>John Doe</h2>
  <p>2 Backroads Lane<br>
    New York<br>
    045935435<br>
    john.doe@gmail.com<br>
  </p>
</body>
</html>
```

John Doe

2 Backroads Lane

New York

045935435

John.doe@gmail.com

- **HTML specifies how the document is to be displayed, and not what information is contained in the document.**
- **Hard for machine to extract the embedded information. Relatively easy for human.**

XML vs HTML



```
<?xml version=1.0?>
<contact>
  <name>John Doe</name>
  <address>2 Backroads Lane</address>
  <country>New York</country>
  <phone>045935435</phone>
  <email>john.doe@gmail.com</email>
</contact>
```

In this case:

The information contained is being marked,
but not for displaying.

Readable by both human and machines.

- SOAP originally stood for "Simple Object Access Protocol" .
- Web Services expose useful functionality to Web users through a standard Web protocol called SOAP.
- Soap is an XML vocabulary standard to enable programs on separate computers to interact across any network. SOAP is a simple markup language for describing messages between applications.
- Soap uses mainly HTTP as a transport protocol. That is, HTTP message contains a SOAP message as its payload section.

SOAP Characteristics



- SOAP has three major characteristics:
 - Extensibility – security and WS-routing are among the extensions under development.
 - Neutrality - SOAP can be used over any transport protocol such as HTTP, SMTP or even TCP.
 - Independent - SOAP allows for any programming model .

SOAP Building Blocks



A SOAP message is an ordinary XML document containing the following elements:

- A required Envelope element that identifies the XML document as a SOAP message.
- An optional Header element that contains header information.
- A required Body element that contains call and response information.
- An optional Fault element that provides information about errors that occurred while processing the message.

SOAP Request



POST /InStock HTTP/1.1

Host: www.stock.org

Content-Type: application/soap+xml; charset=utf-8 Content-Length: 150

```
<?xml version="1.0"?>
```

```
<soap:Envelope
```

```
  xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
```

```
  soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
```

```
    <soap:Body xmlns:m="http://www.stock.org/stock">
```

```
      <m:GetStockPrice>
```

```
        <m:StockName>IBM</m:StockName>
```

```
      </m:GetStockPrice>
```

```
    </soap:Body>
```

```
</soap:Envelope>
```

SOAP Response



HTTP/1.1 200 OK

Content-Type: application/soap; charset=utf-8

Content-Length: 126

```
<?xml version="1.0"?>
```

```
<soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
```

```
  <soap:Body xmlns:m="http://www.stock.org/stock">
```

```
    <m:GetStockPriceResponse>
```

```
      <m:Price>34.5</m:Price>
```

```
    </m:GetStockPriceResponse>
```

```
  </soap:Body>
```

```
</soap:Envelope>
```

SOAP Security



- SOAP uses HTTP as a transport protocol and hence can use HTTP security mainly HTTP over SSL.
- But, since SOAP can run over a number of application protocols (such as SMTP) security had to be considered.
- The [WS-Security specification](#) defines a complete encryption system.

- WSDL stands for Web Services Description Language.
- WSDL is an XML vocabulary for describing Web services. It allows developers to describe Web Services and their capabilities, in a standard manner.
- WSDL specifies what a request message must contain and what the response message will look like in unambiguous notation. In other words, it is a contract between the XML Web service and the client who wishes to use this service.
- In addition to describing message contents, WSDL defines where the service is available and what communications protocol is used to talk to the service.

The WSDL Document Structure



- A WSDL document is just a simple XML document.
- It defines a web service using these major elements:
 - **port type** - The operations performed by the web service.
 - **message** - The messages used by the web service.
 - **types** - The data types used by the web service.
 - **binding** - The communication protocols used by the web service.

WSDL Document



```
<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>
```

- UDDI stands for Universal Description, Discovery and Integration.
- UDDI is a directory for storing information about web services , like yellow pages.
- UDDI is a directory of web service interfaces described by WSDL.



BITS Pilani
Pilani Campus



REST

Representational State Transfer (REST)



- SOAP defines a standard communication protocol (set of rules) specification for XML-based message exchange.
- SOAP uses different transport protocols, such as HTTP and SMTP.
 - The standard protocol HTTP makes it easier for SOAP model to tunnel across firewalls and proxies without any modifications to the SOAP protocol.
 - SOAP can sometimes be slower than middleware technologies like CORBA due to its verbose XML format.

Representational State Transfer (REST)



- REST describes a set of architectural principles by which data can be transmitted over a standardized interface (such as HTTP).
- REST does not contain an additional messaging layer and focuses on design rules for creating stateless services.
- A client can access the resource using the unique URI and a representation of the resource (metadata) is returned.
 - With each new resource representation, the client is said to transfer state.
- While accessing RESTful resources with HTTP protocol, the URL of the resource serves as the resource identifier and GET, PUT, DELETE, POST and HEAD are the standard HTTP operations to be performed on that resource.

Constraints To be Satisfied



- Following have to be satisfied
- Uniform Interface
- Stateless
- Cacheable
- Client-Server
- Layered System

- Resource-Based
 - Individual resources are identified in requests using URIs as resource identifiers
 - The resources themselves are conceptually separate from the representations that are returned to the client.
 - JSON record
- Manipulation of Resources Through Representations
 - When a client holds a representation of a resource, including any metadata attached, it has enough information to modify or delete the resource on the server, provided it has permission to do so.
- Self-descriptive Messages
 - Each message tells how to process: compression schemes, parsers etc.

- Hypermedia as the Engine of Application State (HATEOAS)
 - Clients deliver state via body contents, query-string parameters, request headers and the requested URI (the resource name).
 - Services deliver state to clients via body content, response codes, and response headers. This is technically referred-to as hypermedia (or hyperlinks within hypertext).

- The necessary state to handle the request is contained within the request itself, whether as part of the URI, query-string parameters, body, or headers.
- The URI uniquely identifies the resource and the body contains the state (or state change) of that resource.
- After the server does its processing, the appropriate state, or the piece(s) of state that matter, are communicated back to the client via headers, status and response body.

- As on the World Wide Web, clients can cache responses. Responses must therefore, implicitly or explicitly, define themselves as cacheable, or not, to prevent clients reusing stale or inappropriate data in response to further requests.

- The uniform interface separates clients from servers.
- Clients are not concerned with data storage, which remains internal to each server, so that the portability of client code is improved.
- Servers are not concerned with the user interface or user state, so that servers can be simpler and more scalable.
- Servers and clients may also be replaced and developed independently, as long as the interface is not altered.

Layered System



- A client cannot ordinarily tell whether it is connected directly to the end server, or to an intermediary along the way.
- Intermediary servers may improve system scalability by enabling load-balancing and by providing shared caches.
- Layers may also enforce security policies.

REST API- examples



- [The Twitter REST API](#)
- [Facebook REST API \(deprecating\)](#)
- [Google Translate REST API](#)
- [Flickr REST API](#)
- [Dropbox REST API](#)
- [Ebay Developer REST API \(product centres\)](#)
- [BING Maps REST API \(services\)](#)
- [BING Traffic Incidents API](#)
- [Magento REST API](#)

REST - example



- **Example Dropbox API for file operations**

File operations

`/fileops/copy`

`/fileops/create_folder`

`/fileops/delete`

`/fileops/move`

/fileops/copy	
DESCRIPTION	Copies a file or folder to a new location.
URL STRUCTURE	<code>https://api.dropbox.com/1/fileops/copy</code>
VERSIONS	0, 1
METHOD	POST

Acknowledgements



Q&A





BITS Pilani
Pilani Campus



Thank You