Introduction to Web Services & SOA

References:

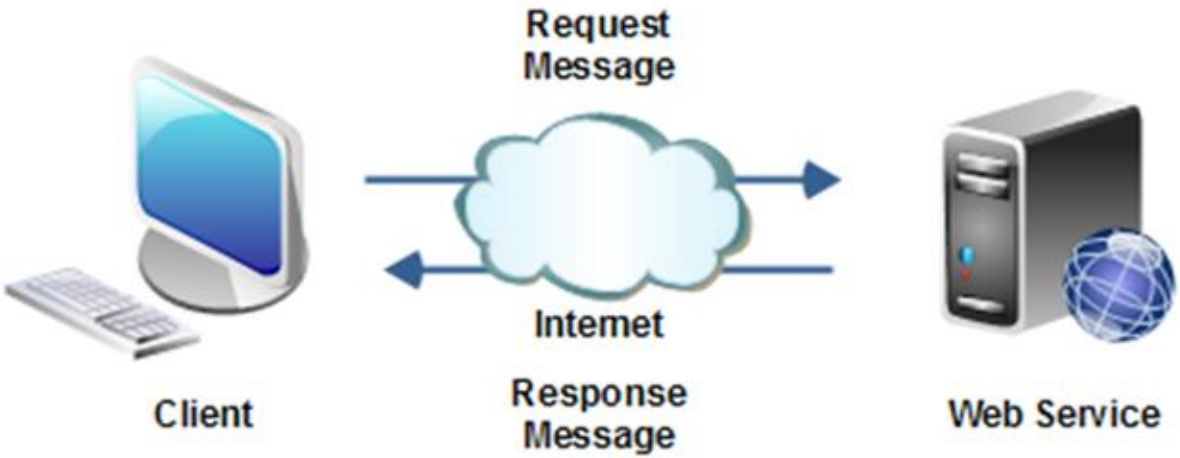
- Web Services, A Technical Introduction, Deitel & Deitel
- Building Scalable and High Performance Java Web Applications, Barish

Web Service Definition

- The term "Web Services" can be confusing. It is, unfortunately, often used in many different ways. Compounding this confusion is term "services" that has a different meaning than the term "Web Services."
- ***Web Services*** refers to the technologies that allow for making connections.
- ***Services*** are what you connect together using Web Services. A service is the endpoint of a connection. Also, a service has some type of underlying computer system that supports the connection offered.
- The combination of services - internal and external to an organization - make up a ***service-oriented architecture***.

What is a Web Service?

- Defined by World Wide Web Consortium (W3C):
  - A software system designed to support interoperable machine-to-machine interaction over a network.
  - It has an interface described in a machine-processable format (specifically WSDL).
  - 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 in conjunction with other Web-related standards.


Request
Message
Internet
Response
Message
Client
Web Service

Examples of Web Services

- A self-contained business task
  - A money withdrawal or funds deposit service for a bank
- A full-fledged business process with multiple tasks
  - Automated purchasing of office supplies with approvals at different levels
- An application
  - A complete life insurance application
- A service-enabled resource
  - Access to a remote database containing patient medical records

Service Oriented Architecture

- A ***service-oriented architecture (SOA)*** is essentially a collection of communicating services. The communication can involve either simple data passing or it could involve two or more services coordinating some activity. Some means of connecting services to each other is needed.
- Service-oriented architectures are not a new thing. The first service-oriented architecture for many people in the past was with the use of Java-RMI, Microsoft DCOM or Object Request Brokers (ORBs) based on the CORBA specification.

Service Oriented Architectures

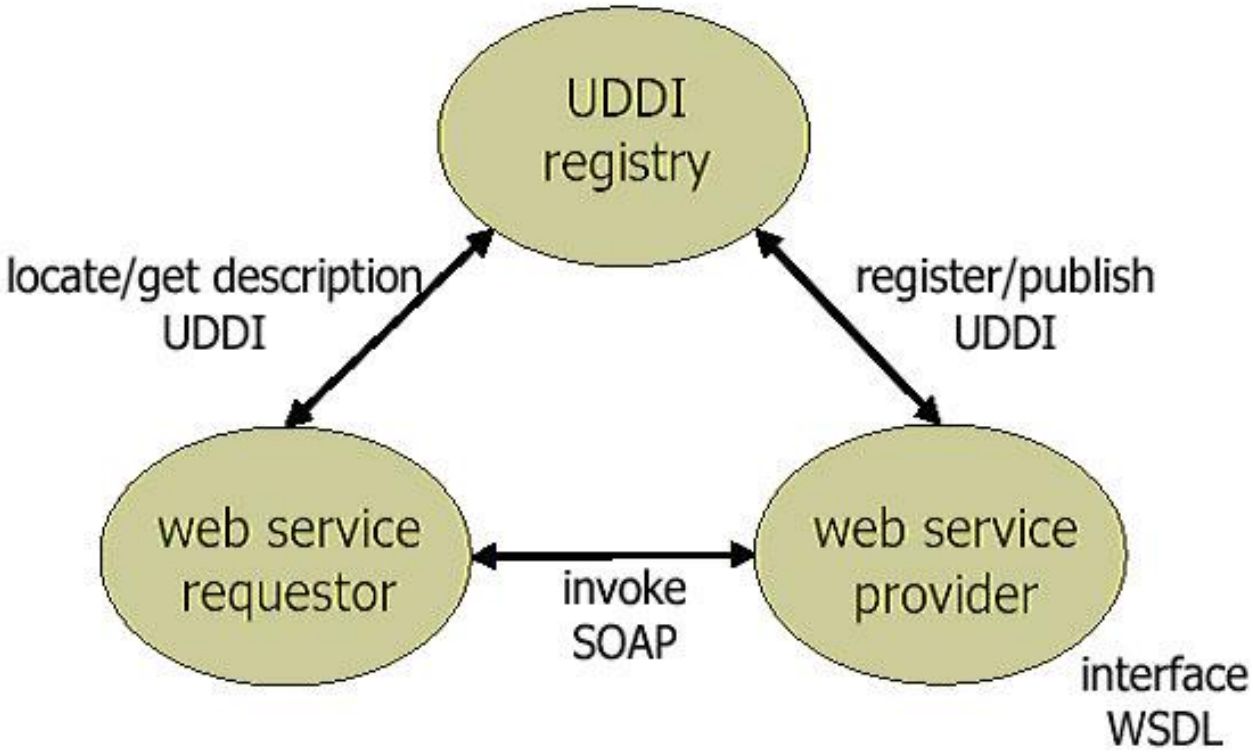
- Build a system by using components that provide services
  - Language independence
  - Platform independence
  - Location independence
  - Static and dynamic service discovery
- Component
  - Medium-grained software entity with a defined interface
    - Interface syntax and semantics – a contract
  - Application domain service (“customer”, “purchase order”, “reservation”) more than a technical infrastructure service (“persistence”, “security”, “messaging”)

Legacy Applications

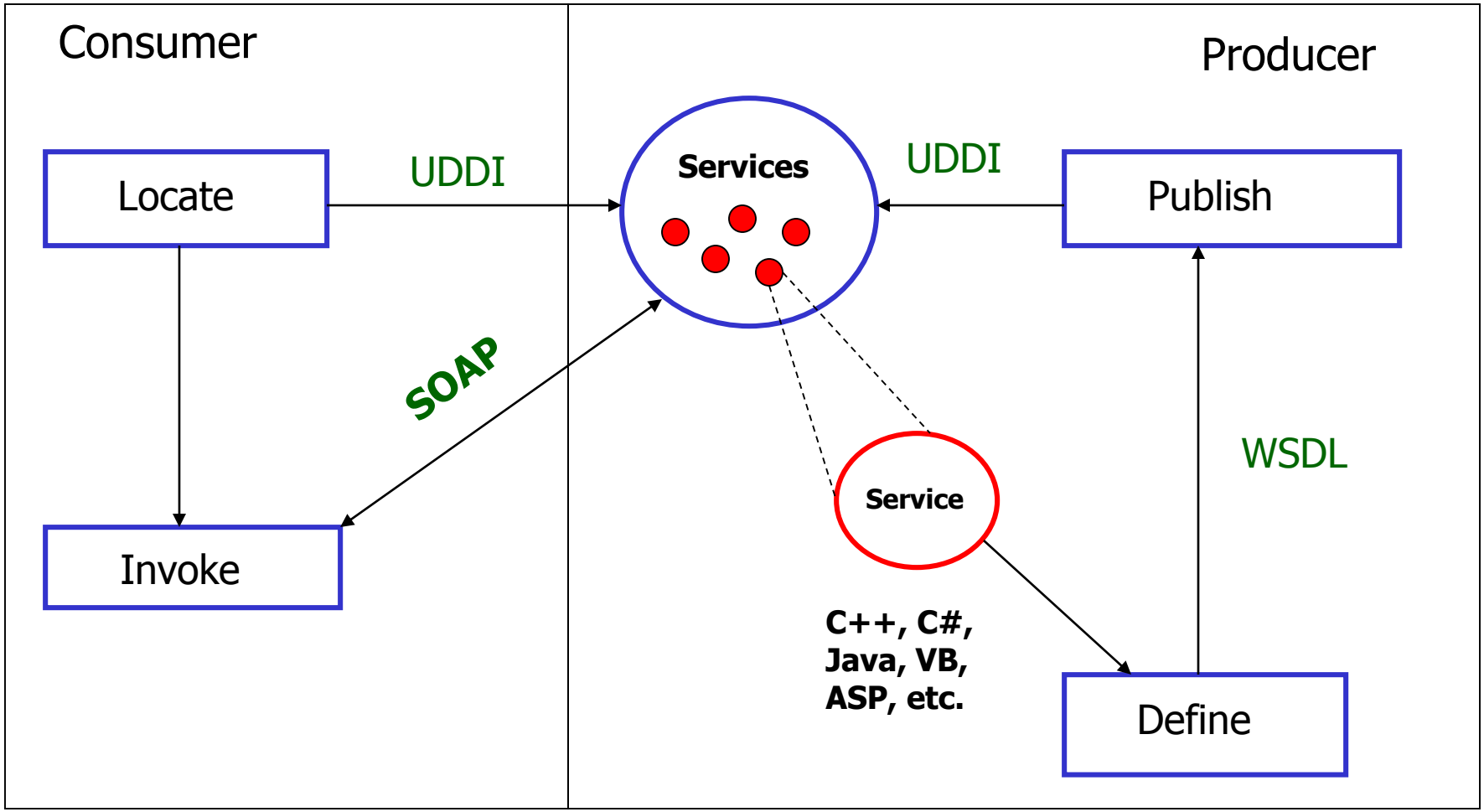
- Legacy applications can be part of an SOA by providing a gateway front-end
  - Translate SOAP invocations and data to/from legacy application's native API and data model
- A more robust "gateway" is an Enterprise Service Bus that can also provide routing, formatting, brokering, etc.
  - A delegate pattern

Web Services: Technology for SOA

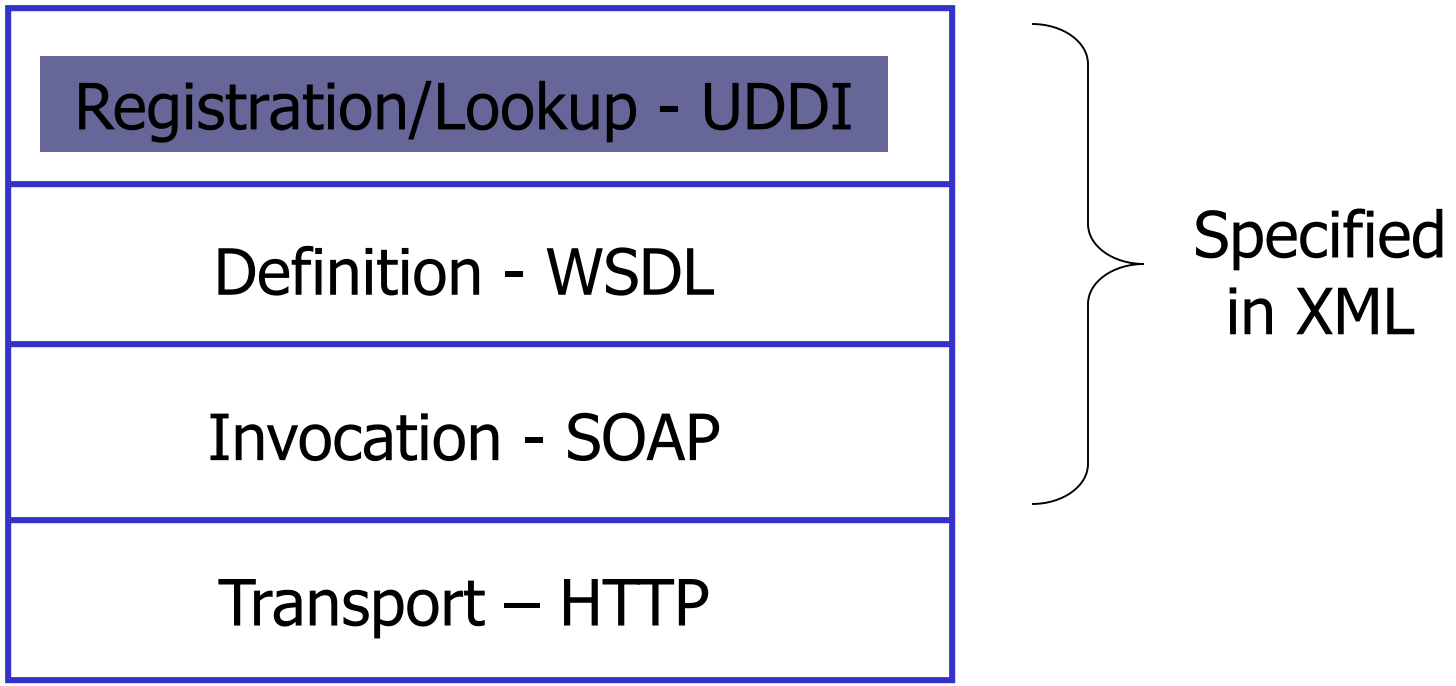
- HTTP for network communications
  - Hypertext Transfer Protocol
- XML for content (data, declarations, messages, results, etc.)
  - eXtensible Markup Language
- SOAP for method invocation
  - Simple Object Access Protocol
- WSDL for service description
  - Web Services Definition Language
- UDDI for registering and discovering service providers
  - Universal Description, Discovery, and Integration


UDDI
registry
locate/get description
UDDI
register/publish
UDDI
web service
requestor
invoke
SOAP
web service
provider
interface
WSDL

Using Web Service Technologies



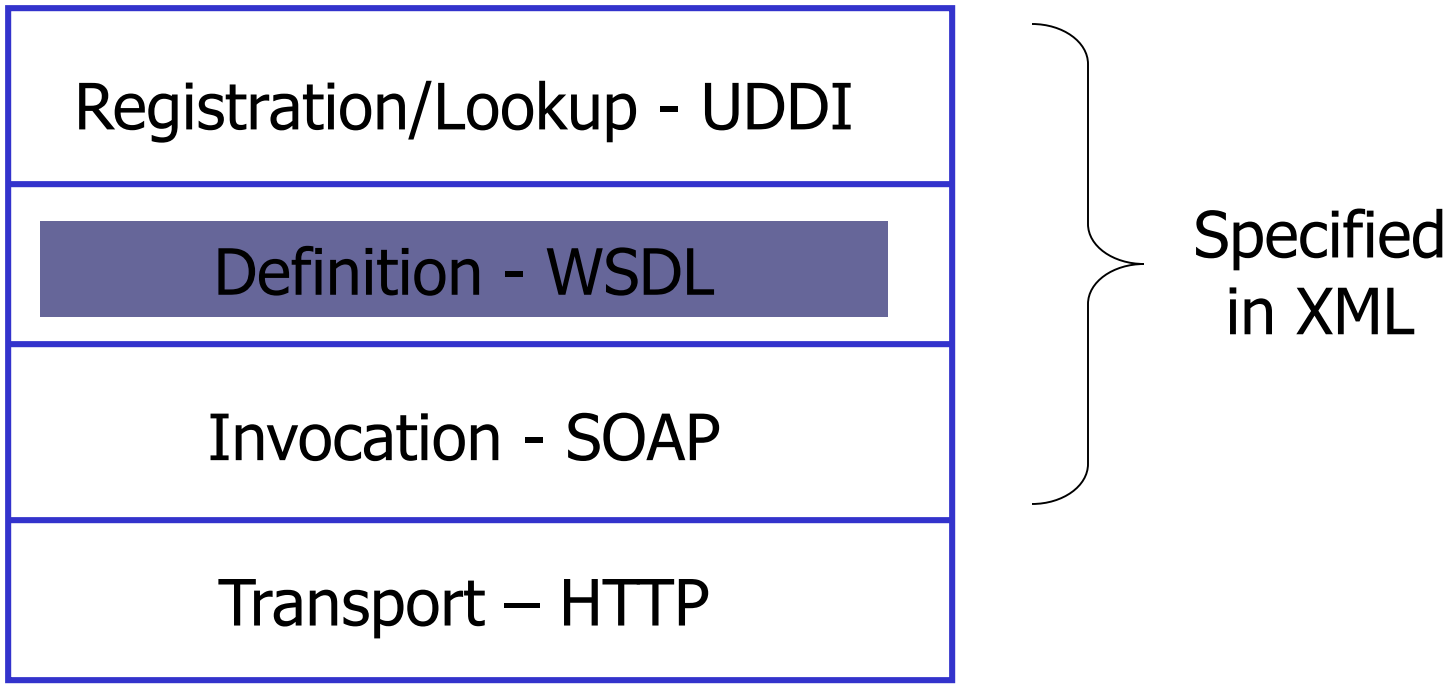
Web Service Technology Stack



UDDI - Universal Descriptor, Discovery and Integration

- Operates like a registry or name server for distributed objects.
- Unlike traditional name servers, UDDI includes metadata about each service – encourages discovery of services.
- Structured like a phone book:
  - white pages – contact information, description
  - yellow pages – classifications about companies
  - green pages – technical data relating to services
- Simplifies process of creating B2B relationships and connecting systems to exchange services and data.

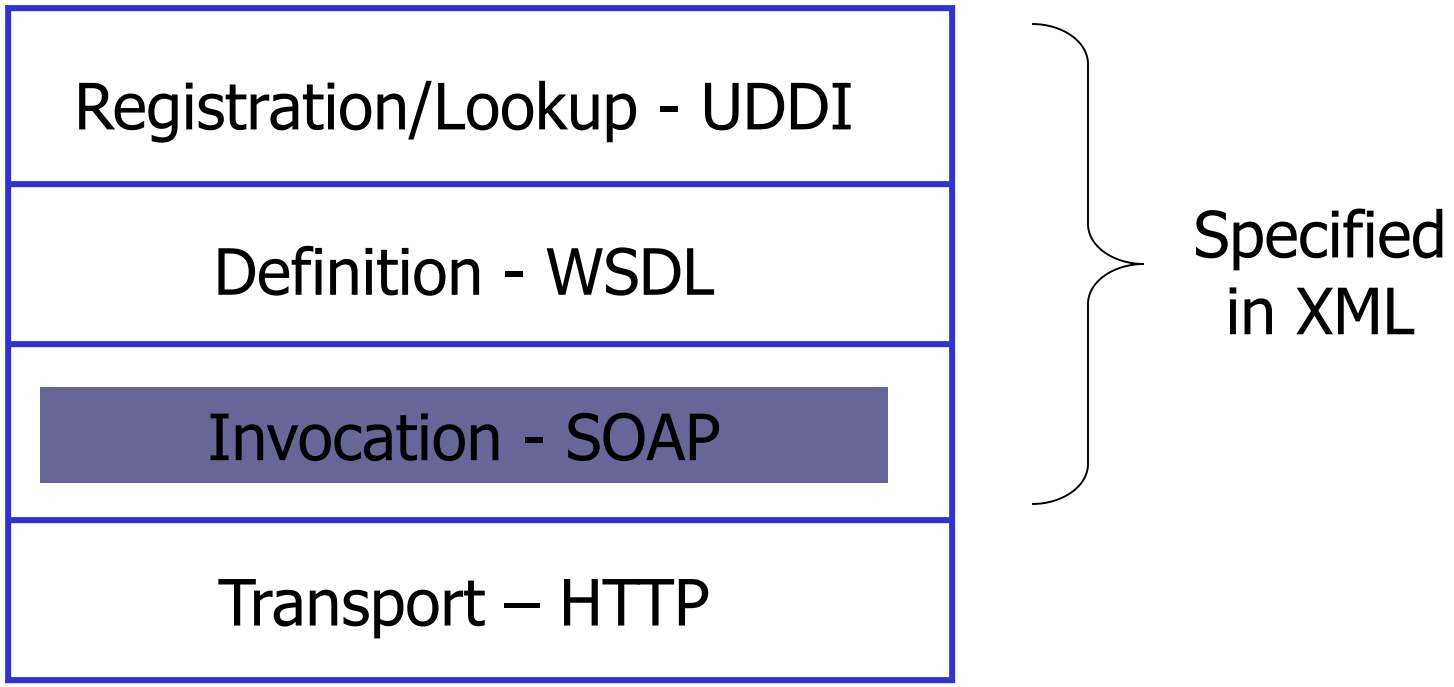
Web Service Technology Stack



WSDL – Web Services Description Language

- Every Web service is accompanied by a WSDL document:
  - XML format
  - defines service capabilities (functionality)
  - location of Web service
  - instructions for use (methods, data types)
- Similar to an IDL in CORBA or Java Interface
- WSDL intended for other applications, not users
- WSDL is usually generated automatically by Web service development tools.

Web Service Technology Stack



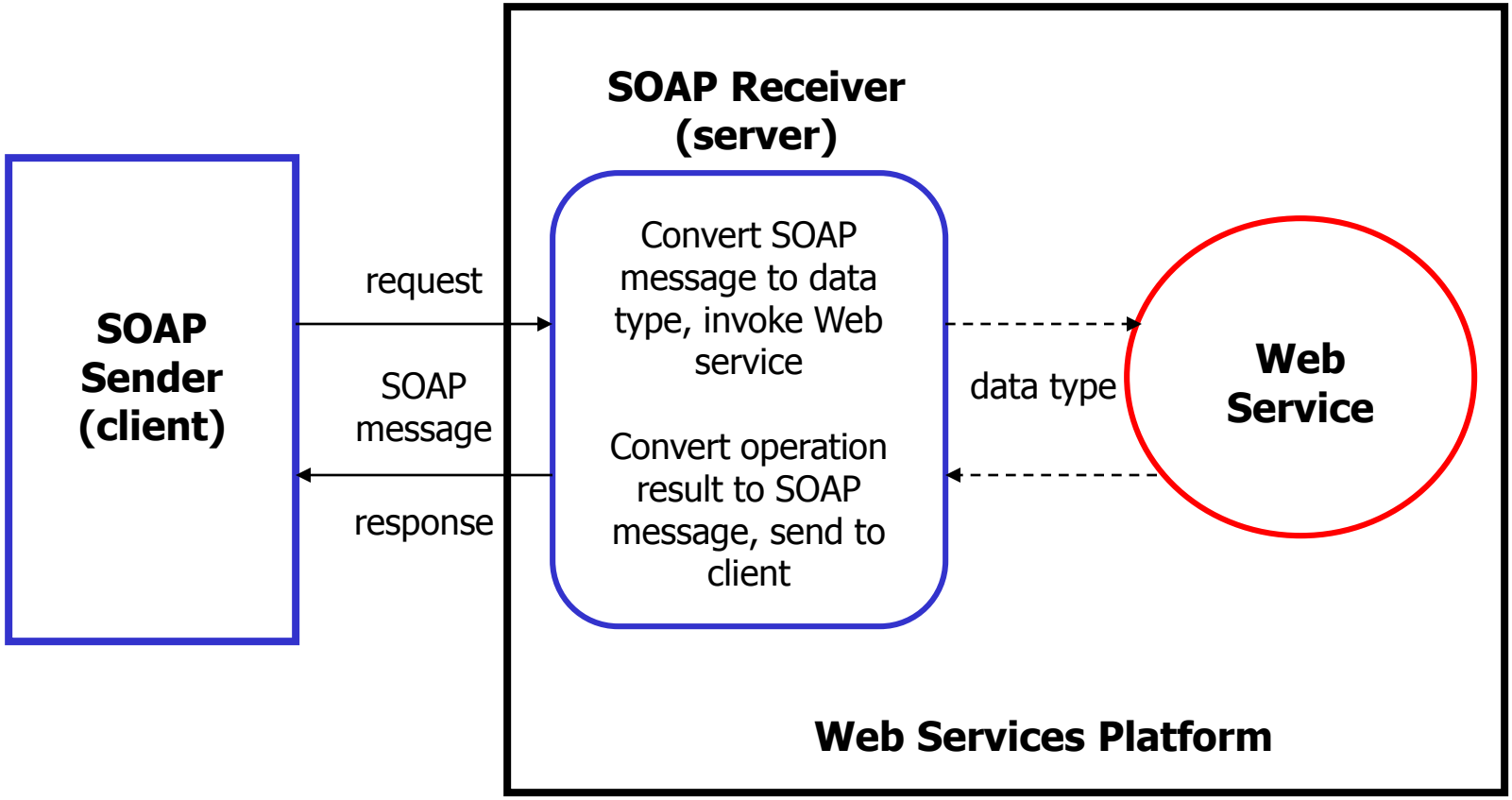
Consuming a Web Service

- The process of invoking a web service is known as ***consuming*** a web service.
  - Option 1 : The consuming program (client) sends a SOAP message to the web service. Both consumer and service use the same WSDL interface that describes the services provided.
  - Option 2 : The consuming program sends a HTTP message

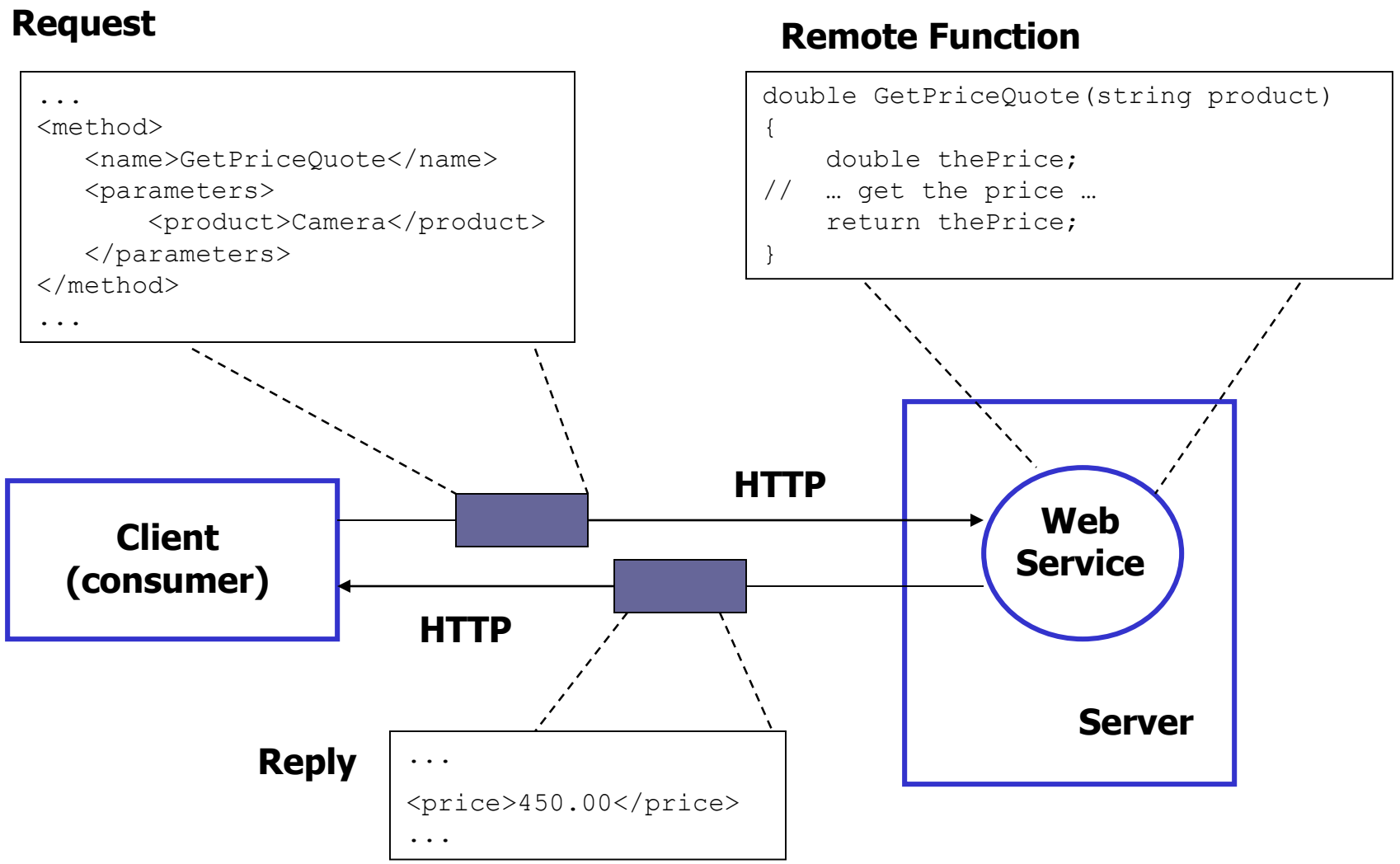
SOAP

- Simple Object Access Protocol
- Describes how to invoke a Web service and process its response.
- Similar to distributed object technologies (CORBA, EJB)
- SOAP messages are received and interpreted by SOAP servers which trigger Web services to perform tasks.
- Consists of a set of standardized XML schemas that define format for transmitting XML messages over a network.
- Layered over HTTP – provides access across firewalls

Typical Web Service Invocation Using SOAP

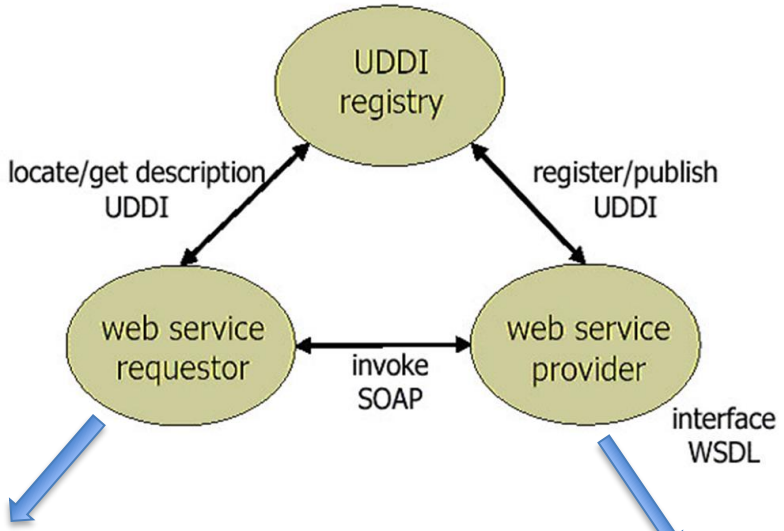


Price Quote Web Service

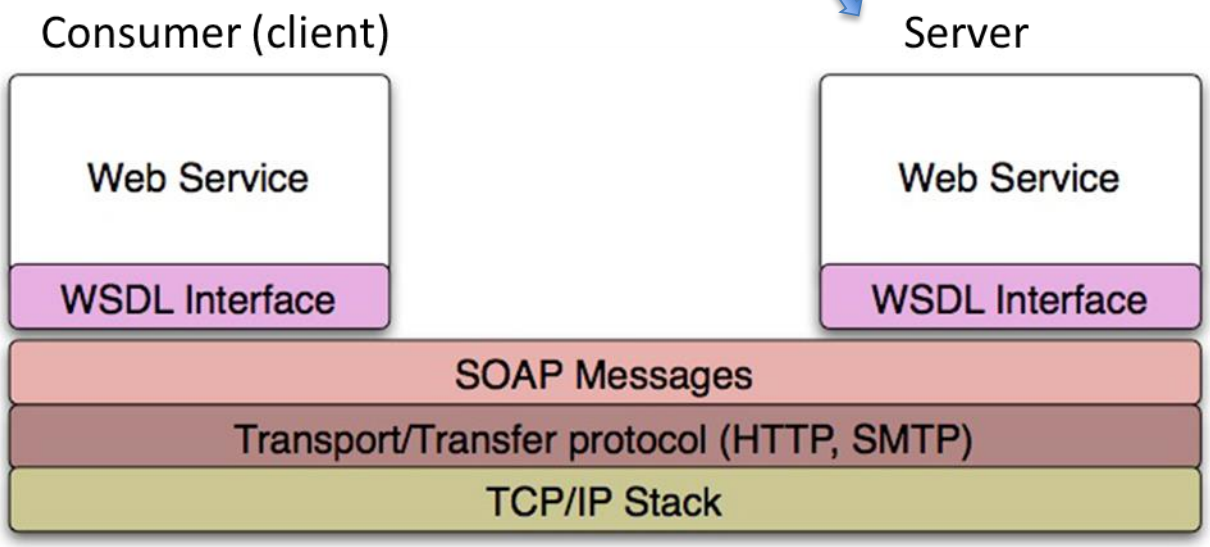


SOAP based Web Service

Web Services ***publish*** their location and services (WSDL Interfaces) in a registry (UDDI).
Clients ***consume*** services using the same WSDL Interface via a SOAP message.
Web Services can act as both publishers and consumers.

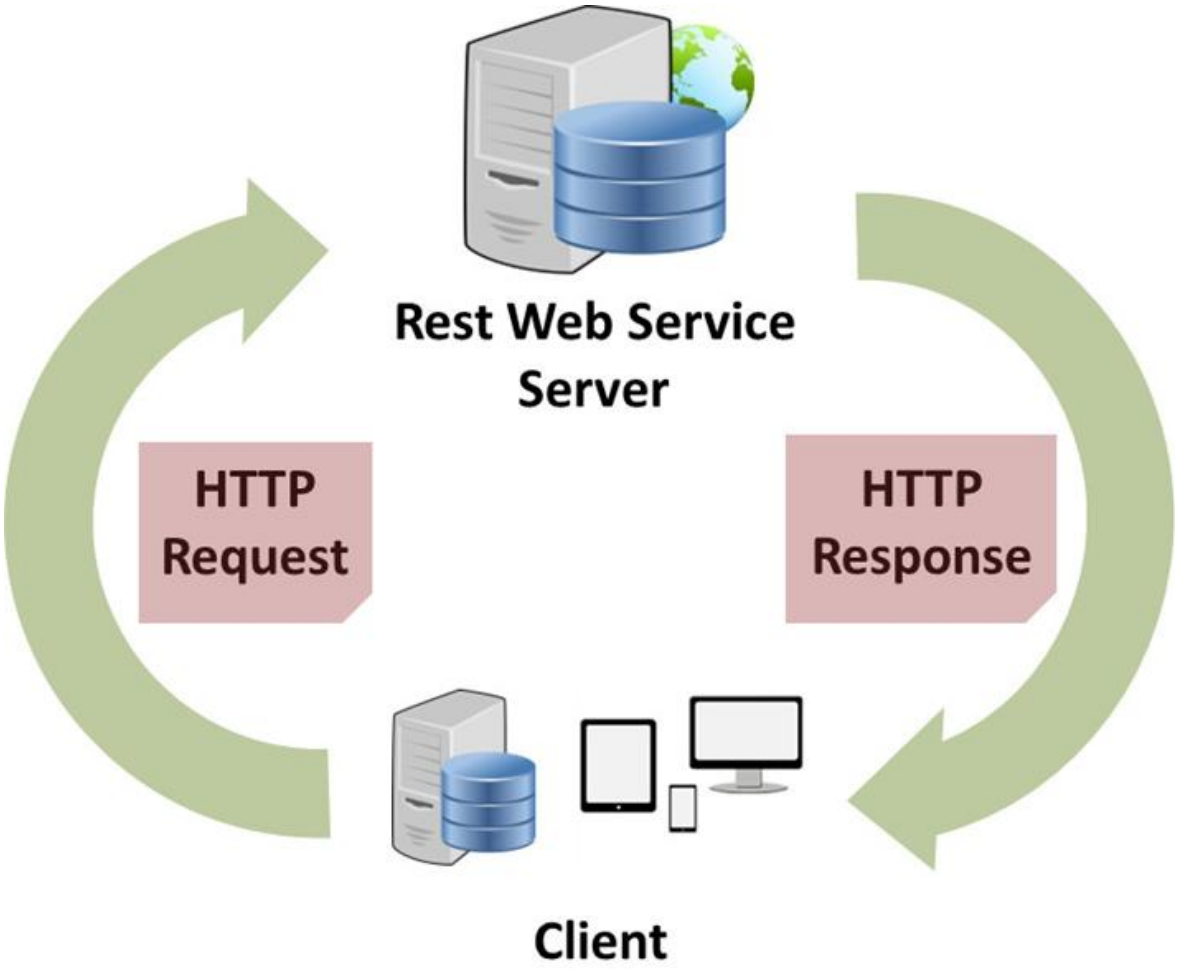


Consumer needs to create a ***remote proxy*** which implements the WSDL Interface ***and*** handles all the responsibilities for sending and receiving SOAP messages.



Consuming a RESTful Web Service

- In a ***Representation State Transfer*** **(REST)** style architecture requests and responses are built around the transfer of representations of ***resources***.
- REST recognizes everything as a resource and each resource implements a standard uniform interface (typically HTTP interface).
- Resources have names and addresses (URLs)
- Each resource has one or more representation (like JSON or XML) and resource representations move across the network usually over HTTP.


Rest Web Service
Server
HTTP
Request
HTTP
Response
Client

Consuming a REST(ful) Web Service

- Create a resource for every service & uniquely identify each resource with a logical URL
- All interactions between a client and a web service are done with simple operations. Most web interactions are done using HTTP and just four operations:
  - retrieve information (HTTP GET)
  - create information (HTTP PUT)
  - update information (HTTP POST)
  - delete information (HTTP DELETE)