Web Services

Part I

Introduction to Service Oriented Architecture

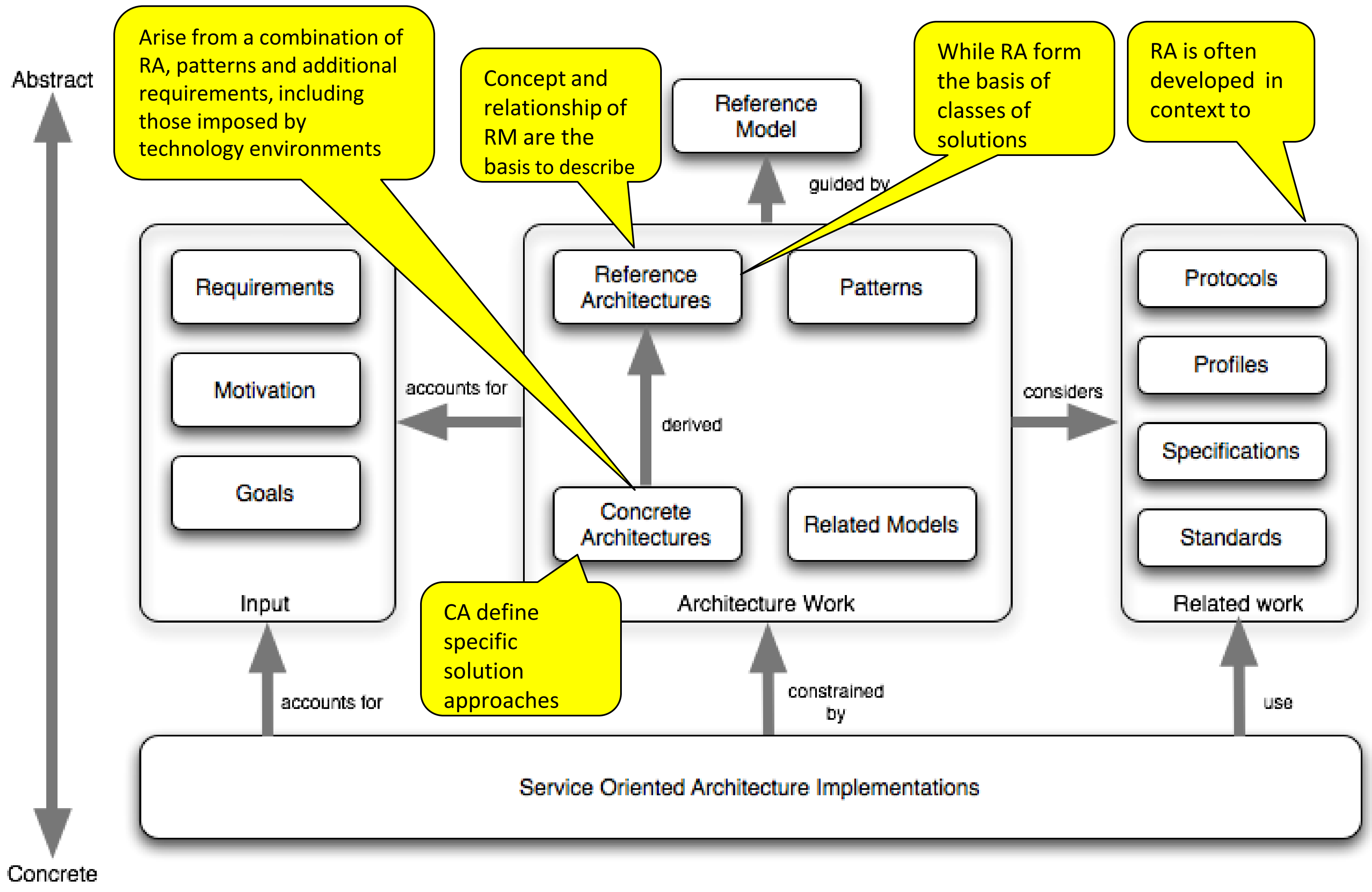
Reference Model (RM) of Service Oriented Architecture (SOA)

- An **abstract framework** for understanding significant relationships among the entities of some environment.
- Consists of a minimal set of unifying concepts, axioms and relationships within a particular problem domain.
- Is independent of specific standards, technologies, implementations, or other concrete details.

What is a Reference Architecture (RA)?

- Recommended patterns.
- Explains and underpins a generic **design template**.
- A **RM**, on the other hand, works at a higher level of abstraction (2-3 level up).

Relationship of RM with other DS Architectures



Abstract
Arise from a combination of RA, patterns and additional requirements, including those imposed by technology environments
Concept and relationship of RM are the basis to describe
Reference Model
guided by
While RA form the basis of classes of solutions
RA is often developed in context to
Requirements
Motivation
Goals
Input
accounts for
Reference Architectures
Patterns
derived
Concrete Architectures
Related Models
Architecture Work
considers
Protocols
Profiles
Specifications
Standards
Related work
CA define specific solution approaches
accounts for
constrained by
use
Service Oriented Architecture Implementations
Concrete

Ref.: Ed. by MacKenzie C.M. *et al.,* Reference Model of Service Oriented Architecture 1.0, August 2006

Residential Housing Example

- A reference model would talk about eating areas, hygiene areas and sleeping areas.
- More than one reference architectures may address the problem of providing housing.
- A reference architecture may exist for a flat, independent houses and for a bungalow.

What is SOA?

- *"Service Oriented Architecture is a paradigm for organizing and utilizing distributed capabilities that <u>may</u> be under control of different ownership domains"*

Ref.: Ed. by MacKenzie C.M. *et al.,* Reference Model of Service Oriented Architecture 1.0, August 2006

What is SOA?

- Entities (people and organizations) create capabilities to solve or support a solution for the problems in their business
- One person's needs being met by capabilities offered by someone else; or, in the world of distributed computing, one computer agent's requirements being met by a computer agent belonging to a different owner

Ref.: Ed. by MacKenzie C.M. *et al.,* Reference Model of Service Oriented Architecture 1.0, August 2006

What is SOA?

- There is not necessarily a one-to-one correlation between needs and capabilities
- The granularity of needs and capabilities vary from fundamental to complex
- Any given need may require the combining of numerous capabilities while any single capability may address more than one need

Ref.: Ed. by MacKenzie C.M. *et al.,* Reference Model of Service Oriented Architecture 1.0, August 2006

What is SOA?

- Not itself a solution to domain problems but rather an organizing and delivery paradigm.
- Key concepts are visibility, interaction and effect.

Ref.: Ed. by MacKenzie C.M. *et al.,* Reference Model of Service Oriented Architecture 1.0, August 2006

Terms

- Visibility refers to the capacity of those with needs and those with capabilities to see each other.
- Interaction is the activity of using capability grounded in a particular execution context.
- Capabilities are used to realize real world effects (return of information or change in state of entities).

SOA in Computer Science Applications

Where is the Origin

SOA was first invented in the 1960s. It was called *Procedural Programming* in those days, and it featured this neat abstraction where:

- A procedure can take requests called "Procedure Calls"
- Given a well-formed request, a procedure will provide a response
- A procedure can use other procedures

Where is the Origin

- In 1976 or thereabouts, **Bruce Jay Nelson** got the bright idea that one should be able to call procedures on other systems

  This was called a remote procedure call, a remarkably simple name.

Why Service Oriented Architecture?

- Large scale Enterprise systems
- Internet scale provisioning of services (in clouds)
- Reducing cost of doing business
- Build scalable, evolvable systems
  - Scalable because minimizes assumptions
- Manage complex systems
- Reusability of business functions

Why is it different?

- SOA reflects the reality of ownership boundaries
  - CORBA, RMI, COM, DCOM, etc. all try to implement *transparent* distributed systems
  - Ownership is the essence in SOA
- SOA is task oriented
  - Services are organized by functions
    - Getting something done
- SOA is inspired by human organizations
  - It worked for us, it should work for machines

Key concepts

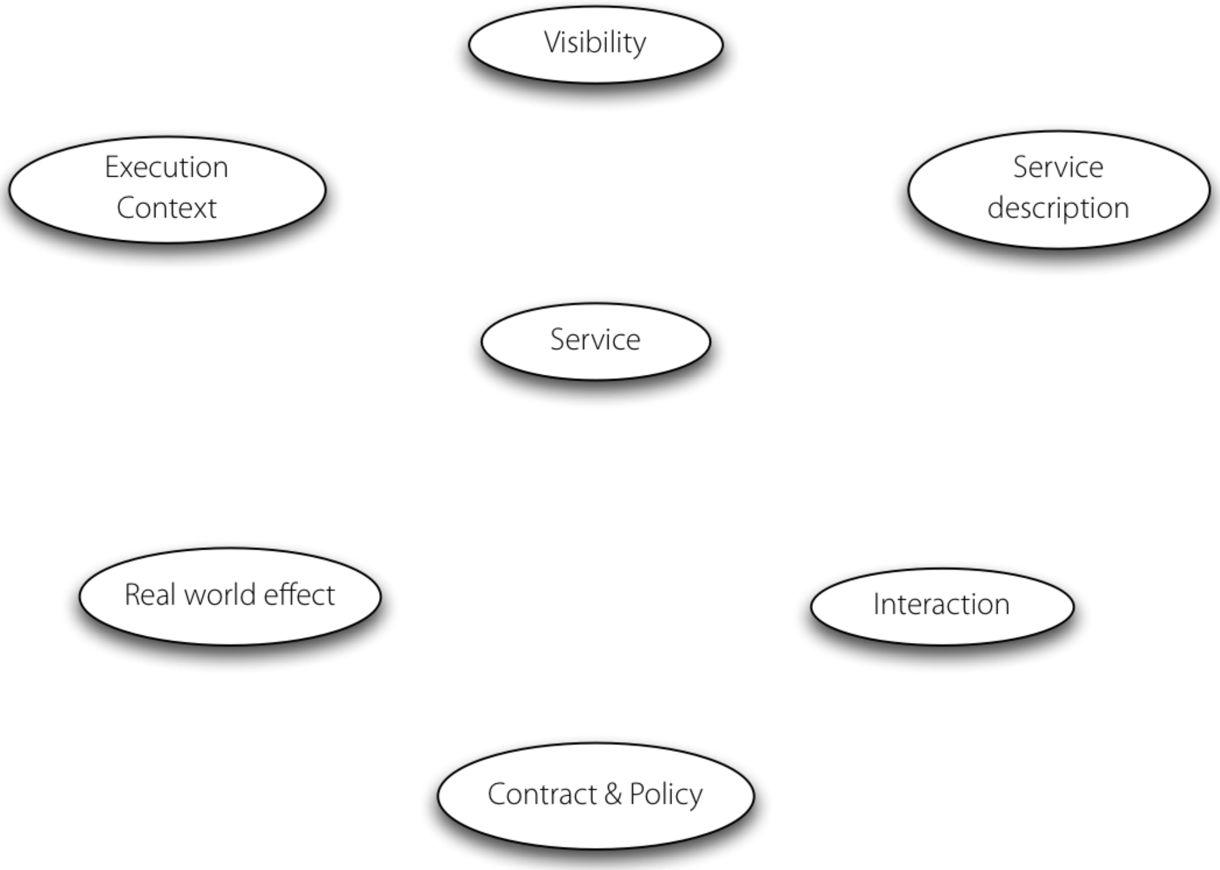


Fig. Courtesy: Reference Model of Service Oriented Architecture 1.0

Service

- A mechanism to enable access to one or more capabilities
  - using a prescribed interface
  - constraints and policies are specified by the service description.

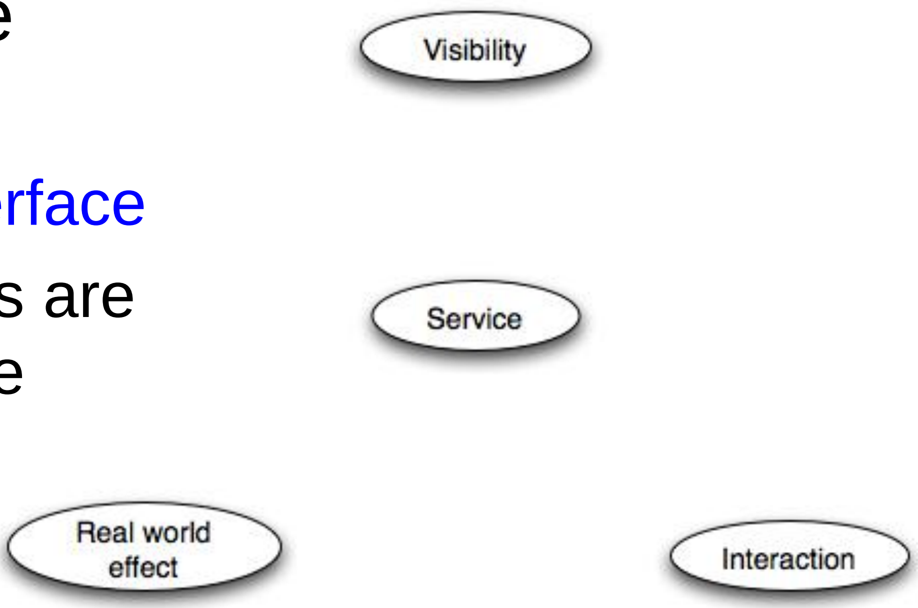


Fig. Courtesy: Reference Model of Service Oriented Architecture 1.0

Service

- Definition: The performance of work (a function) by one for another.

- The notion of **Service** also includes:

  - The capability to perform work for another.
  - The specification of the work offered for another.
  - The offer to perform work for another.

- *In SOA, services are the mechanisms by which needs and capabilities are brought together.*

Visibility

Visibility is the relationship between service participants that is satisfied when they are able to interact with each other

- **Awareness**
  - Service description
  - Discovery mechanism
- **Willingness**
  - Policy & contract
- **Reachability**
  - Communication

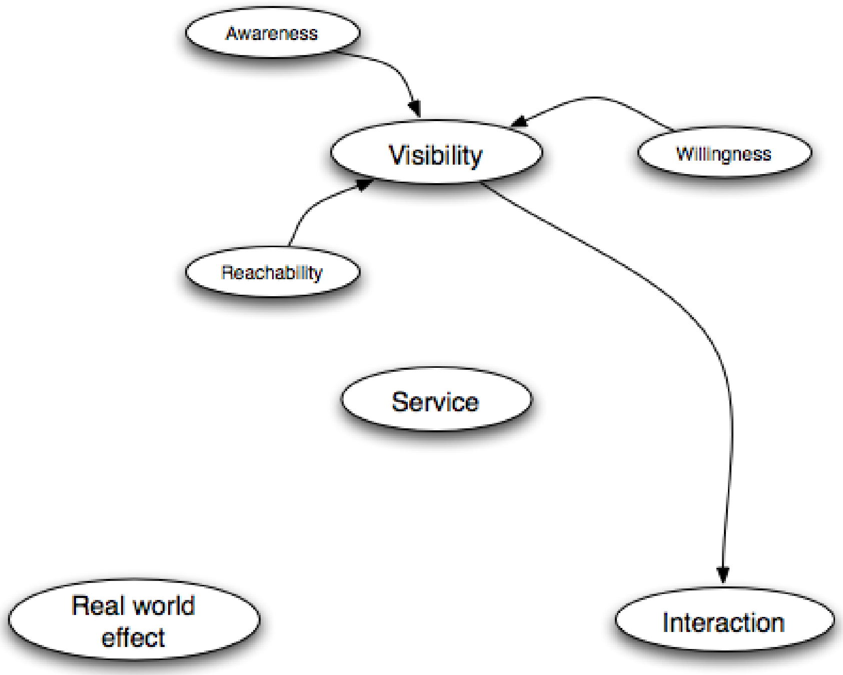


Fig. Courtesy: Reference Model of Service Oriented Architecture 1.0

Interaction

Interacting with a service involves performing actions against the service

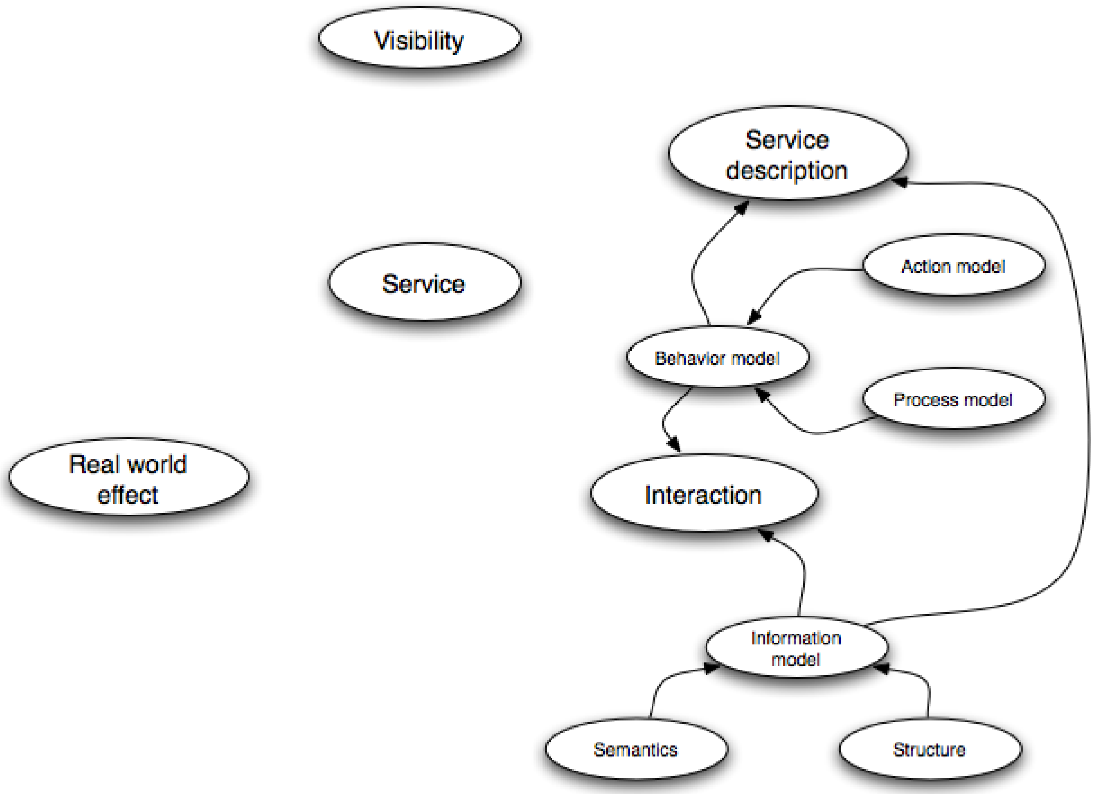


Fig. Courtesy: Reference Model of Service Oriented Architecture 1.0

Policy and Contract

- Policy
  - Constraint representing the intention of a participant in a service
- Contract
  - Constraint representing an agreement between two or more participants

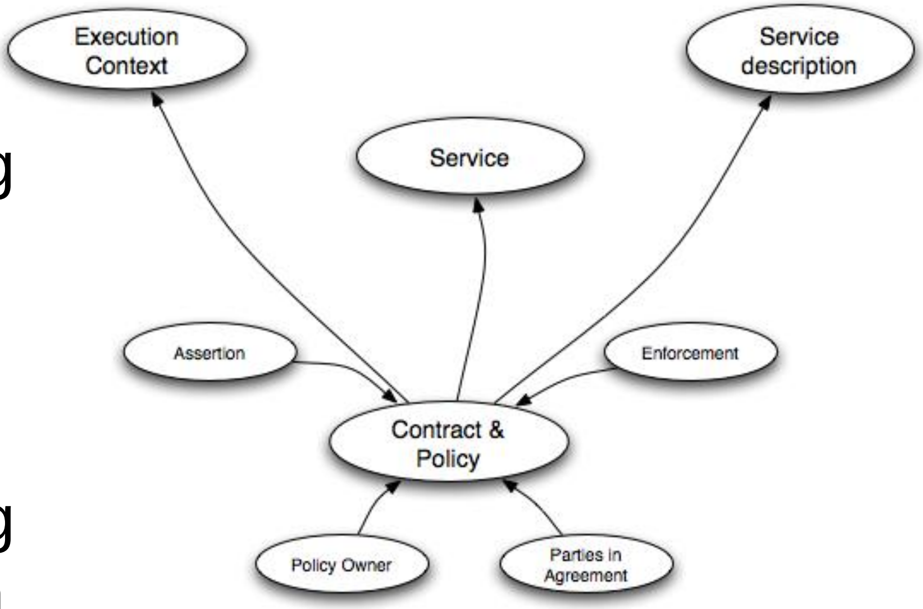


Fig. Courtesy: Reference Model of Service Oriented Architecture 1.0

Description

The **service description** represents the information needed in order to use, manage or provide a service.

- Service Reachability
- Service Functionality
- Service Policies
- Service Interface

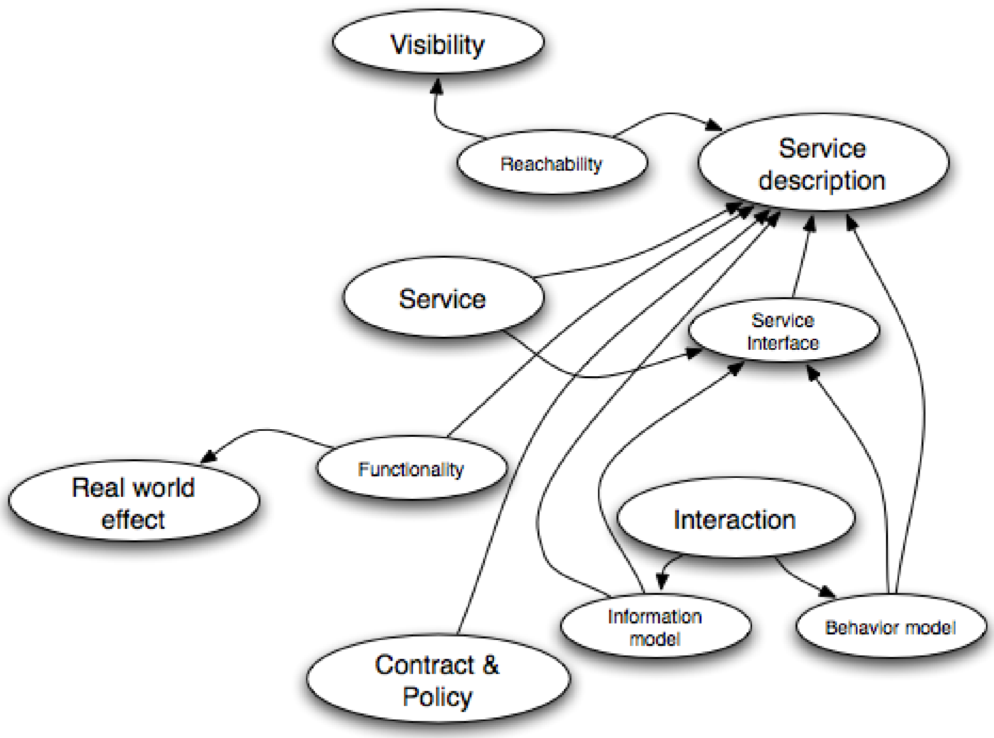


Fig. Courtesy: Reference Model of Service Oriented Architecture 1.0

Execution Context

The **execution context** is the set of infrastructure elements, process entities, policy assertions and agreements that are identified as part of an instantiated service interaction, and thus forms **a path between those with needs and those with capabilities**

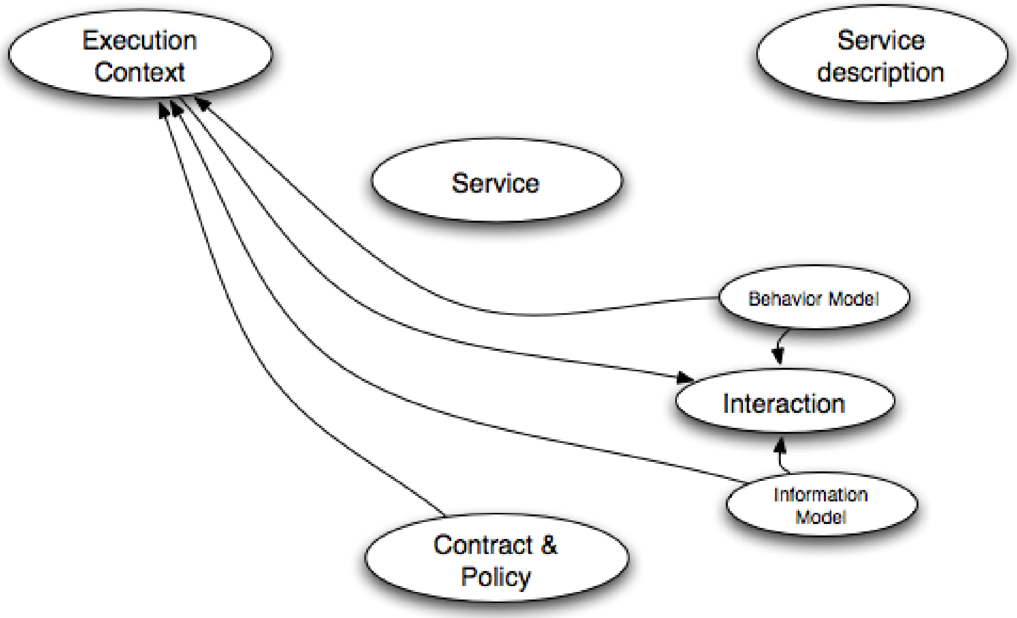


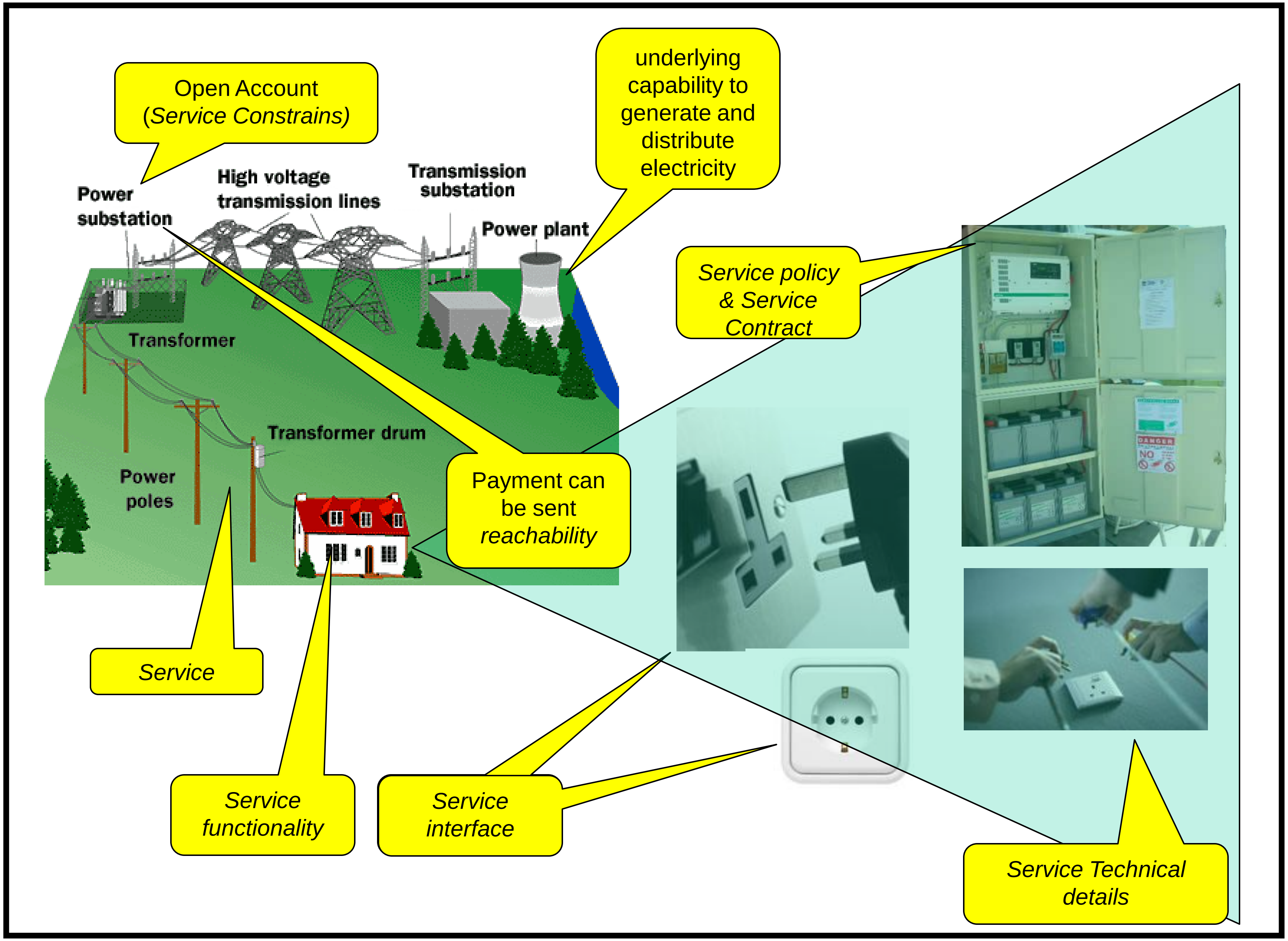
Fig. Courtesy: Reference Model of Service Oriented Architecture 1.0

SOA Differs from OOP

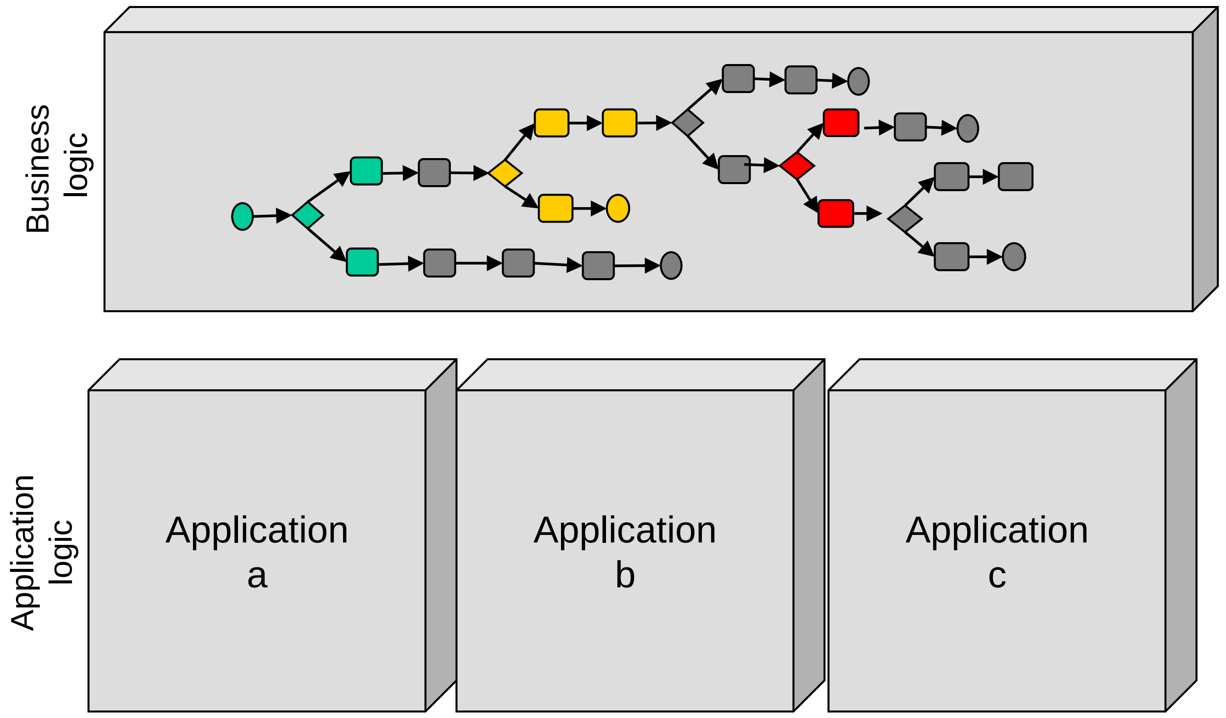
- The **Object Oriented Paradigm** focuses on packaging data with operations. The **SOA Paradigm** focuses on the task or business function. It may or may not be associated with methods and properties.
- To use an object, we first create it. On the other hand we interact with a service where it exists.
- SOA places greater emphasis on clear semantics.
- SOA, like human activity, works by delegation.

SOA Differs from OOP

- SOA takes ownership boundaries more seriously.
- The Four OOP Design Patterns don’t apply here. We don’t talk about inheritance, aggregation, polymorphism and so on…
- Services are simpler things.

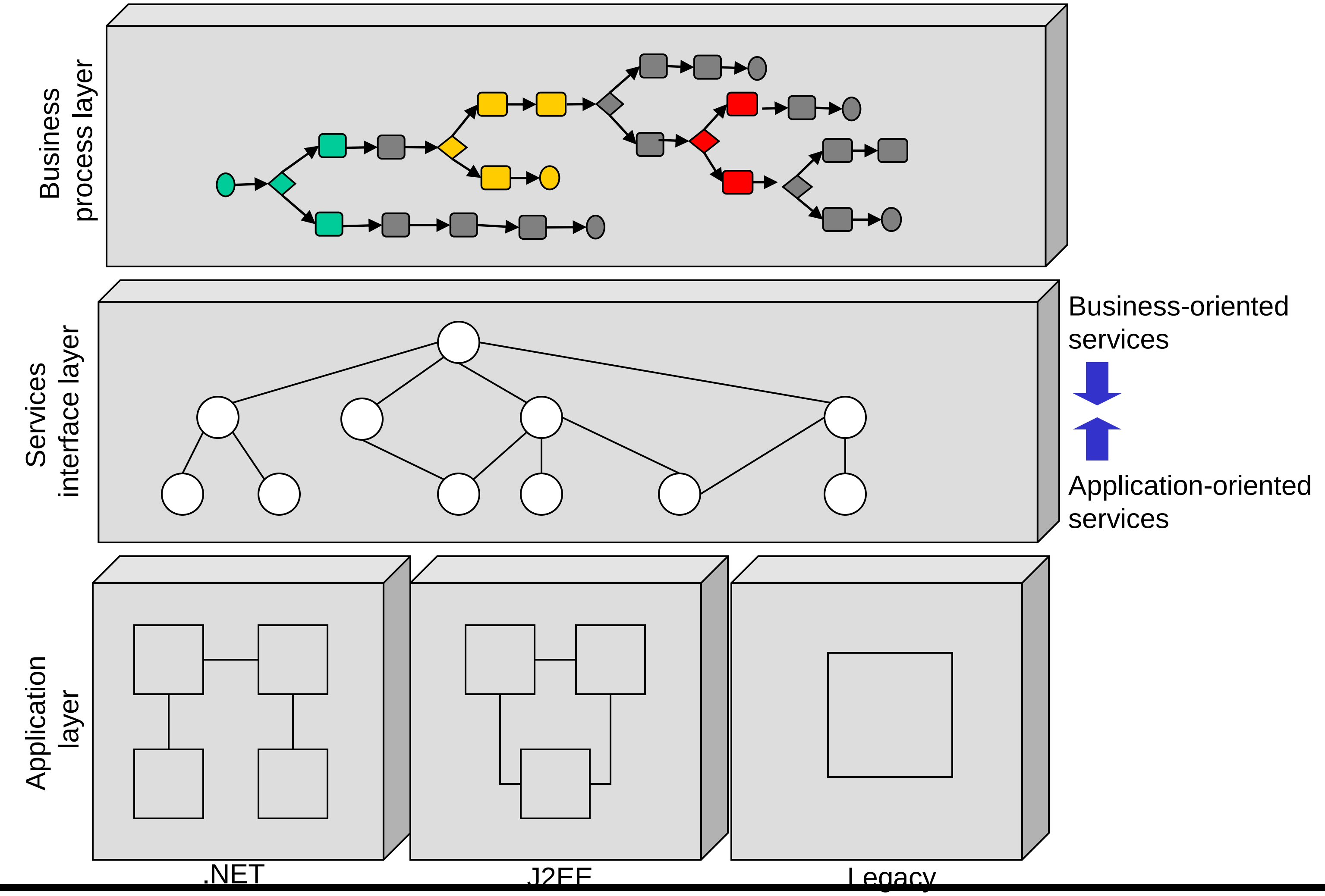

Open Account (*Service Constrains*)
underlying capability to generate and distribute electricity
Power substation
High voltage transmission lines
Transmission substation
Power plant
Transformer
Transformer drum
Power poles
Service policy & Service Contract
Payment can be sent *reachability*
Service
Service functionality
Service interface
Service Technical details
DANGER
NO

Focus on the Business– Process and Services

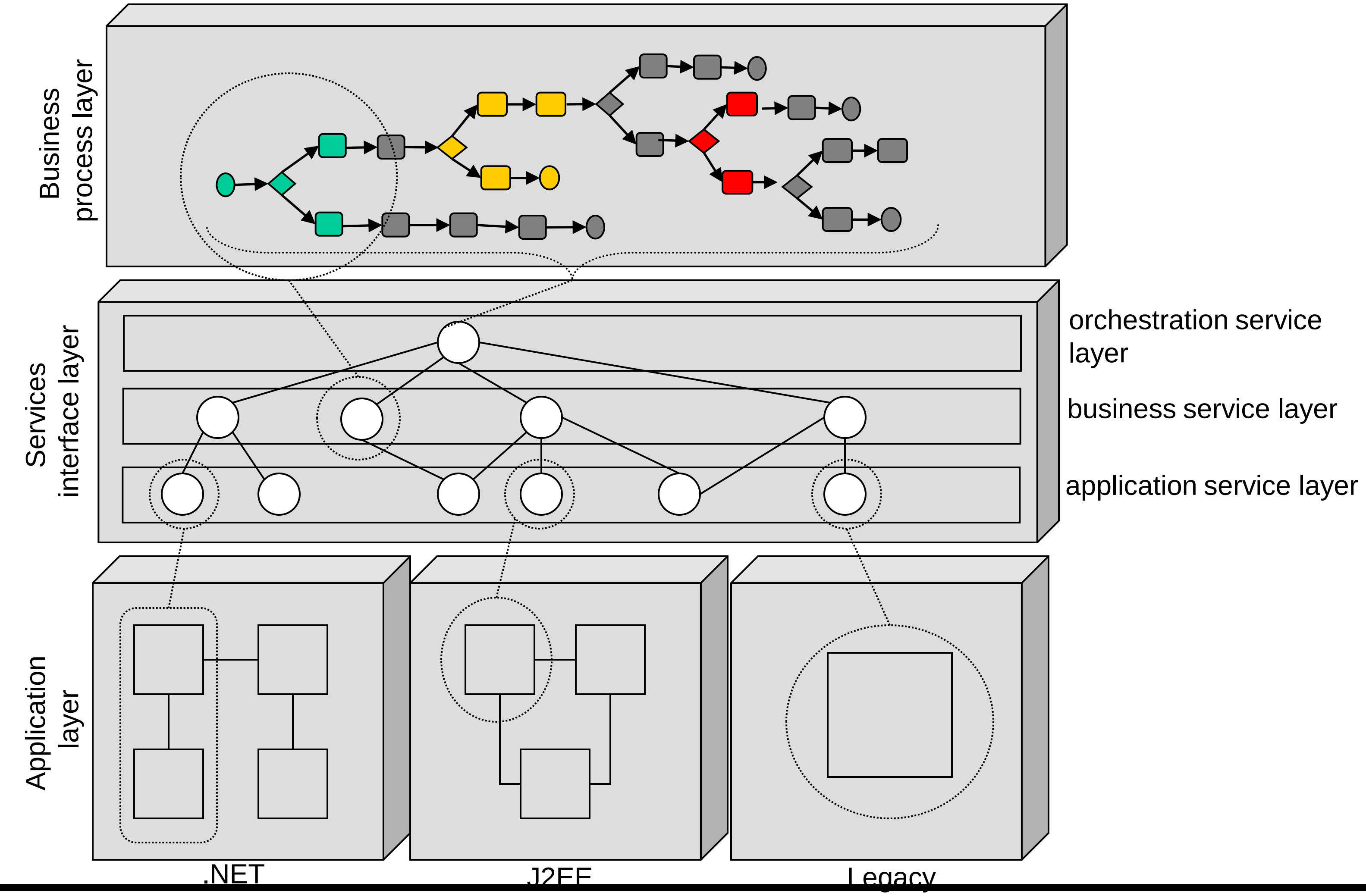


Source: Service-Oriented Architecture, Thomas Erl

Focus on the Business– Process and Services



Focus on the Business– Process and Services



What is a Web Service?
(Some suggested definitions)

- "… a piece of business logic accessible via the Internet using *open standards*…" (Microsoft)
- Encapsulated, loosely coupled, contracted software functions, offered via standard protocols over the web (Dest*i*Corp)
- A set of interfaces, which provide a standard means of interoperation between different software applications, running on a variety of platforms and/or frameworks (W3C)
- is a functionality that can be *engaged* over the Web

Web Services (User's view)

- Software components designed to provide specific operations ("services") accessible using standard Internet technology
- Not tied to any one operating system or programming language
- Should be *self-describing:*
  - publish a public interface to the services
- Should be *discoverable*:
  - Mechanism for publishing what you have created
  - Can be found via a simple ***'find'*** mechanism
- Usually through SOAP (**Simple Object Access Protocol**) messages carrying XML documents, and a HTTP transport protocol or using REST

Web Services (User's view)

- Software components designed to provide specific operations ("services") accessible using standard Internet technology
- Not tied to any one operating system or programming language
- Should be *self-describing:*
  - publish a public interface to the services
- Should be *discoverable*:
  - Mechanism for publishing what you have created
  - Can be found via a simple ***'find'*** mechanism
- Usually through SOAP (**Simple Object Access Protocol**) messages carrying XML documents, and a HTTP transport protocol or using REST

Web Services (User's view)

- Software components designed to provide specific operations ("services") accessible using standard Internet technology
- Not tied to any one operating system or programming language
- Should be *self-describing:*
  - publish a public interface to the services
- Should be *discoverable*:
  - Mechanism for publishing what you have created
  - Can be found via a simple ***'find'*** mechanism
- Usually through SOAP (**Simple Object Access Protocol**) messages carrying XML documents, and a HTTP transport protocol or using REST

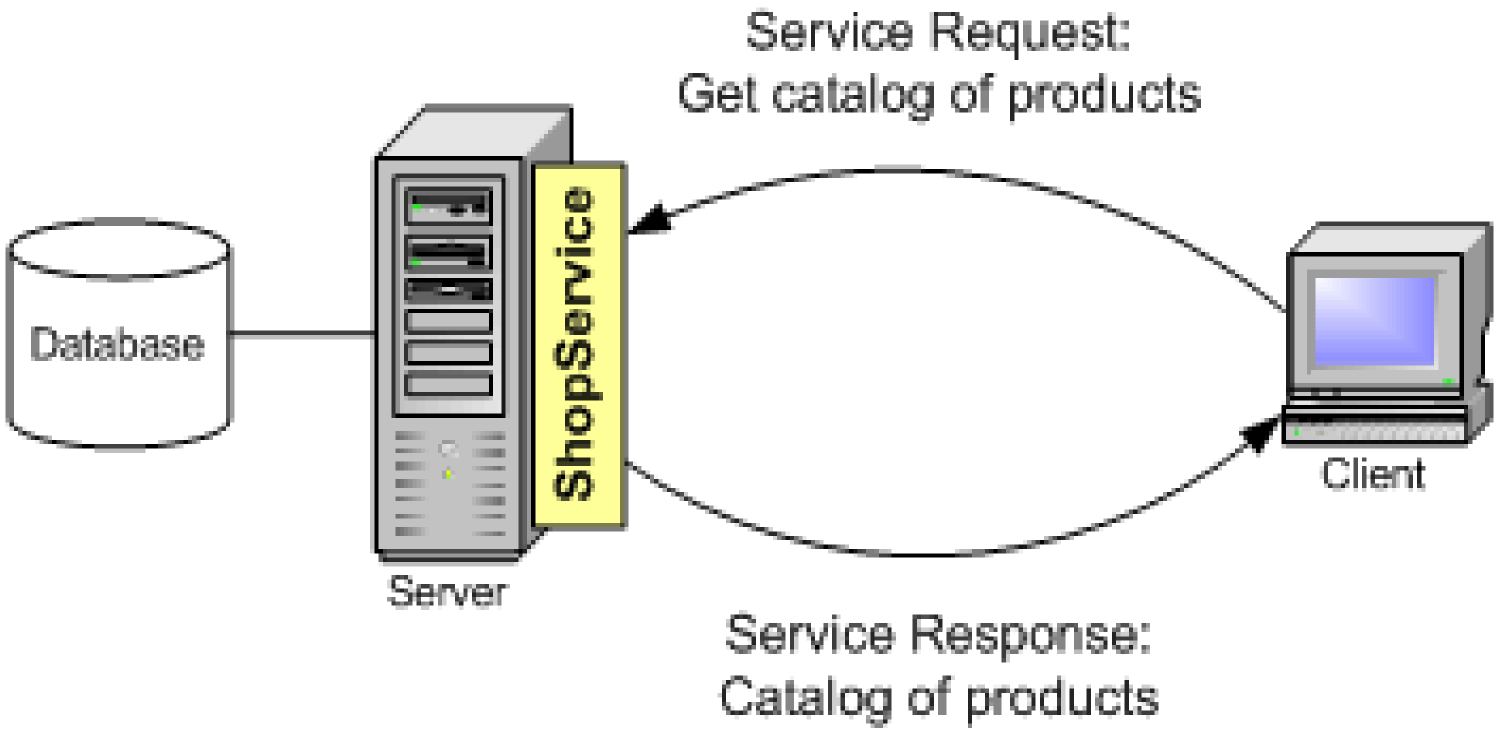
Web Services (User's view)

- Software components designed to provide specific operations ("services") accessible using standard Internet technology
- Not tied to any one operating system or programming language
- Should be *self-describing:*
  - publish a public interface to the services
- Should be *discoverable*:
  - Mechanism for publishing what you have created
  - Can be found via a simple ***'find'*** mechanism
- Usually through SOAP (**Simple Object Access Protocol**) messages carrying XML documents, and a HTTP transport protocol or using REST

Web Services (User's view)

- Software components designed to provide specific operations ("services") accessible using standard Internet technology
- Not tied to any one operating system or programming language
- Should be *self-describing:*
  - publish a public interface to the services
- Should be *discoverable*:
  - Mechanism for publishing what you have created
  - Can be found via a simple ***'find'*** mechanism
- Usually through SOAP (**Simple Object Access Protocol**) messages carrying XML documents, and a HTTP transport protocol or using REST

Basic client-server model



Source: http://www.globus.org

Basic client-server model

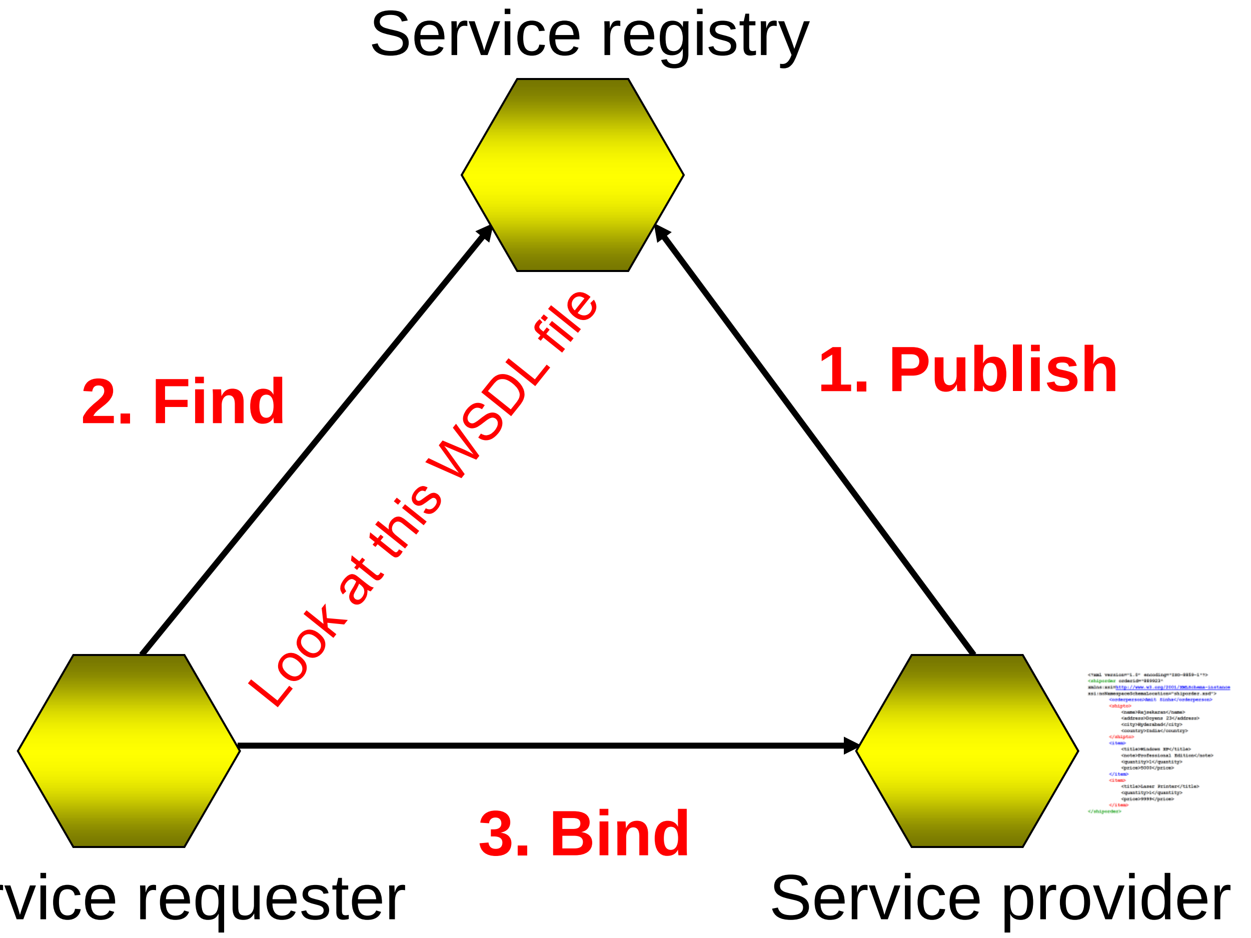
- Client needs to:
  - identify location of the required service
  - know how to communicate with the service to get what it required.
- Uses service registry - a third party.

Service-Oriented Architecture

Steps:

- Services “published” in a Service registry.
- Service requestor asks Service registry to locate the service.
- Service requestor “binds” with service provider to invoke service.

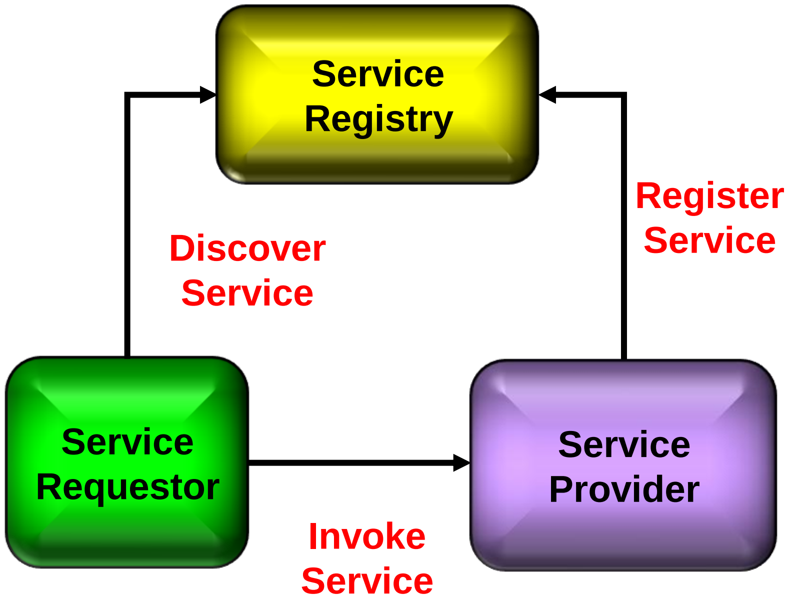
Service-Oriented Architecture in Web Servies



```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<shiporder orderid="889923"
xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
xsi:noNamespaceSchemaLocation="shiporder.xsd">
        <orderperson>Amit Sinha</orderperson>
        <shipto>
            <name>Rajsekaran</name>
            <address>Goyens 23</address>
            <city>Hyderabad</city>
            <country>India</country>
        </shipto>
        <item>
            <title>Windows XP</title>
            <note>Professional Edition</note>
            <quantity>1</quantity>
            <price>5000</price>
        </item>
        <item>
            <title>Laser Printer</title>
            <quantity>1</quantity>
            <price>9999</price>
        </item>
</shiporder>
```

Web Services Roles

- Service Provider:
  - Implements service
  - Makes it available on Internet
- Service Requestor:
  - Consumer of WS
- Service Registry:
  - WS directory



Services Oriented Architecture (SOA)

- A service-oriented architecture is essentially a collection of services.
- These services communicate with each other.
- The communication can involve either simple data passing or it could involve orchestration of two or more services coordinating some activity.
- Some means of connecting services to each other is needed.

Comparison

Has similarities with RMI and other distributed object technologies (CORBA etc.) **but::**

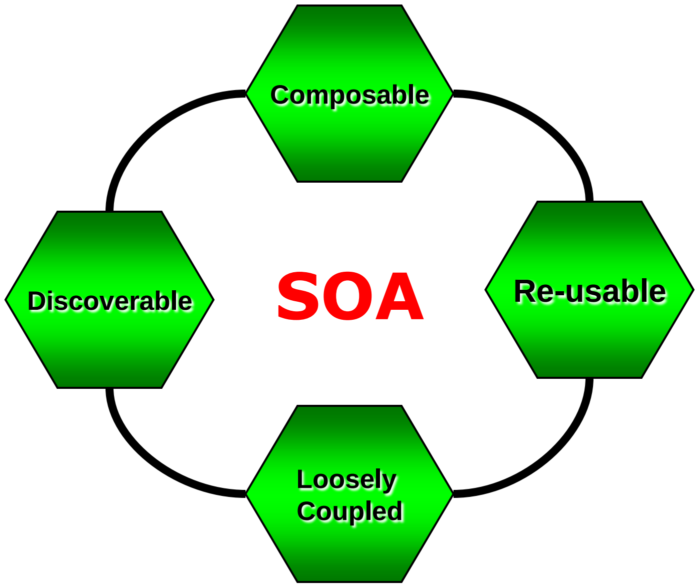
- Web Services are platform independent
  - They use XML (within a SOAP message)/REST
  - Most use HTTP to transmit message.
- RMI (Java Reliable Multicast Protocol) can be used only with Java technology.
- CORBA doesn't work properly on WAN, it is suited for tightly coupled designs and good for object oriented languages and vendor specific ORBs.

XML-based Web Services

- XML provides a flexible basis for storing and retrieving service information on web services.
- Web services use data-centric XML documents to communicate information.

Key Aspects of SOA

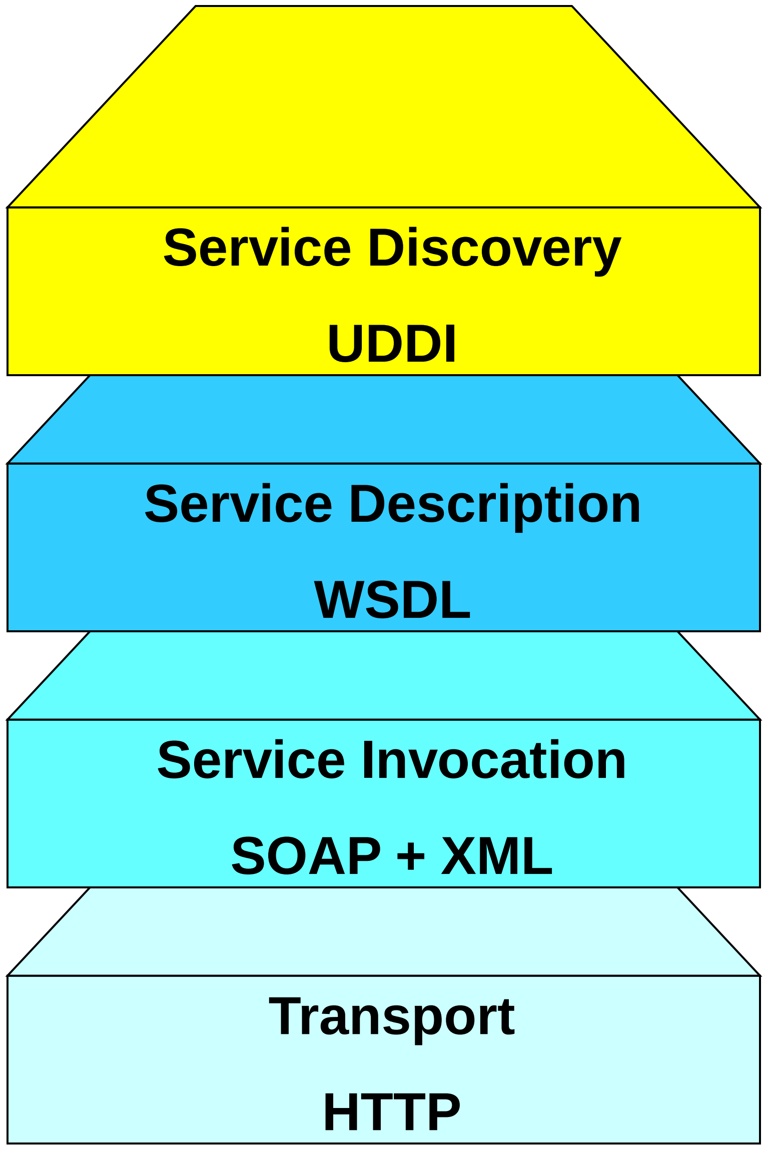
- Loose coupling
  - minimizes dependencies
- Service contract
  - adhere to a communications agreement
- Autonomy
  - control the logic they encapsulate
- Abstraction
  - hide logic from the outside world
- Reusability
  - promoting reuse
- Composability
  - coordinated and assembled to form composite services
- Statelessness
  - minimize retaining information specific to an activity
- Discoverability
  - externally descriptive



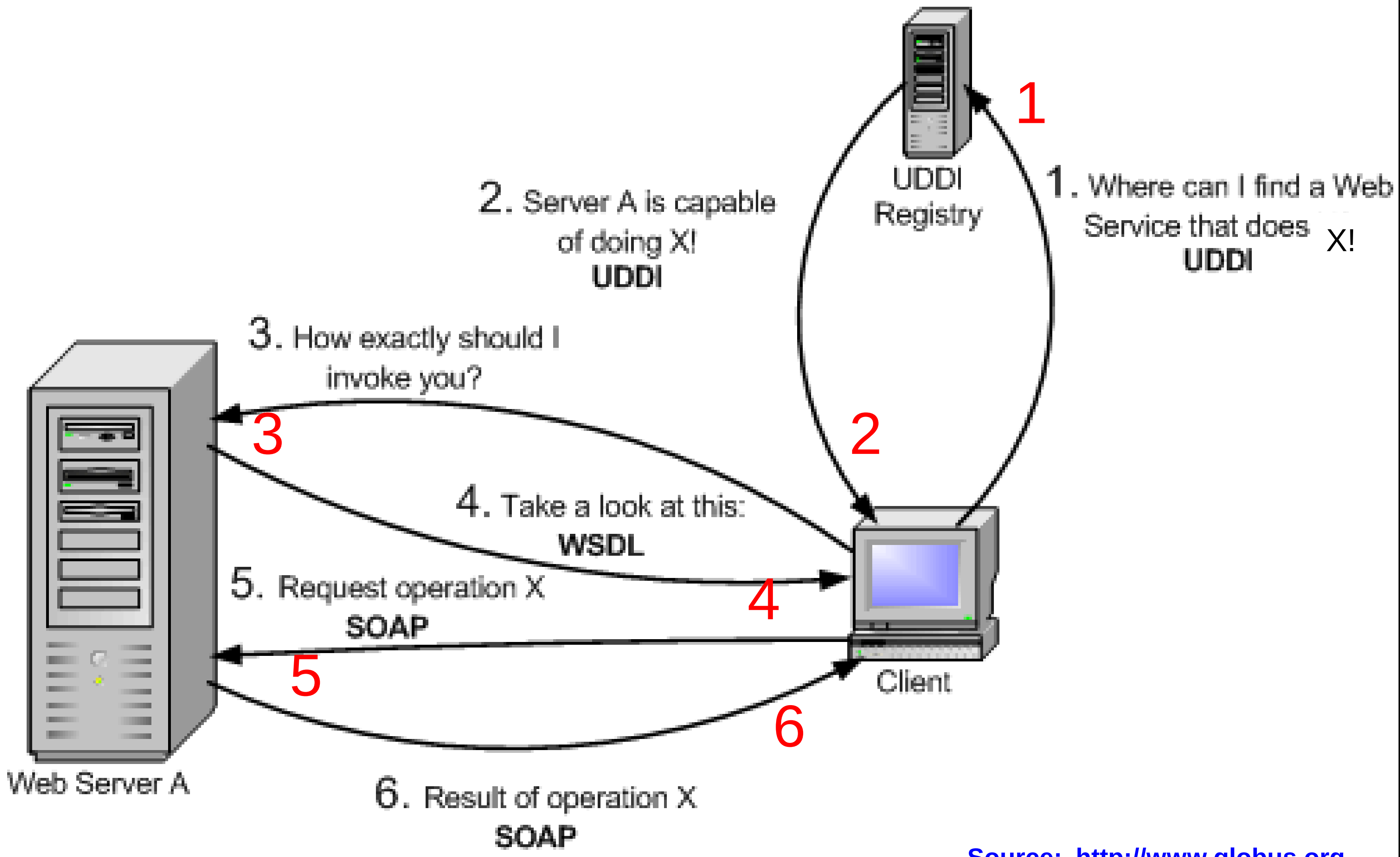
Web Services “Stack”

- HTTP transport
- SOAP message carrying XML documents
- WSDL (Web Services Description Language) used to describe message syntax for invoking a service and its response
- UDDI (Universal Description, Discovery and Integration) used as web service discovery mechanism

Web Services "Stack"



Web Services



Source: http://www.globus.org

Simple Object Access Protocol (SOAP)

- SOAP is a XML base RPC protocol
- A SOAP message is fundamentally a one-way transmission between SOAP nodes
  - from a SOAP sender to a SOAP receiver
- SOAP messages are expected to be combined by applications to implement more complex interaction patterns:
  - request/response
  - multiple, back-and-forth "conversational” exchanges
  - etc.

Simple Object Access Protocol (SOAP)

- SOAP, to put it simply, allows Java objects and COM objects to talk to each other in a distributed, decentralized, Web-based environment.

- More generally, SOAP allows objects (or code) of any kind -- on any platform, in any language -- to cross-communicate. At present, SOAP has been implemented in many languages on several platforms. Suddenly objects everywhere, local and remote, large and small, are able to interoperate.

SOAP - Acronym for

- **Simple:** transporting XML structured messages across internet using HTTP
- **Object:** transportation of COM objects
  - Common Object Model (COM): open software architecture from Microsoft, DEC, allowing interoperation between Object Broker and OLE
- **Access:** services will be easier to deploy when binding them to common protocols (HTTP)
  - web page data pass through most firewalls
- **Protocol:** SOAP is an XML based protocol used to exchange distributed data over HTTP
  - Origins in RPC

What is SOAP?

- SOAP stands for Simple Object Access Protocol
- SOAP is a communication protocol
- SOAP is for communication between applications
- SOAP is a format for sending messages
- SOAP communicates via Internet
- SOAP is platform independent
- SOAP is language independent
- SOAP is based on XML
- SOAP is simple and extensible
- SOAP allows you to get around firewalls
- SOAP is a W3C recommendation

SOAP vs. DCOM/CORBA

- SOAP is a protocol:
  - mandates how a method call transfers over the wire.
- SOAP model follows standard web and distributed apps:
  - Stateless transactions provide client with data needed for task:
    - get data from web server via method call, use data, send updated data back to server in another stateless remote call.
- SOAP uses single function or method invocations
  - does not maintain state to an object between calls.
  - client makes single method calls, per HTTP request.

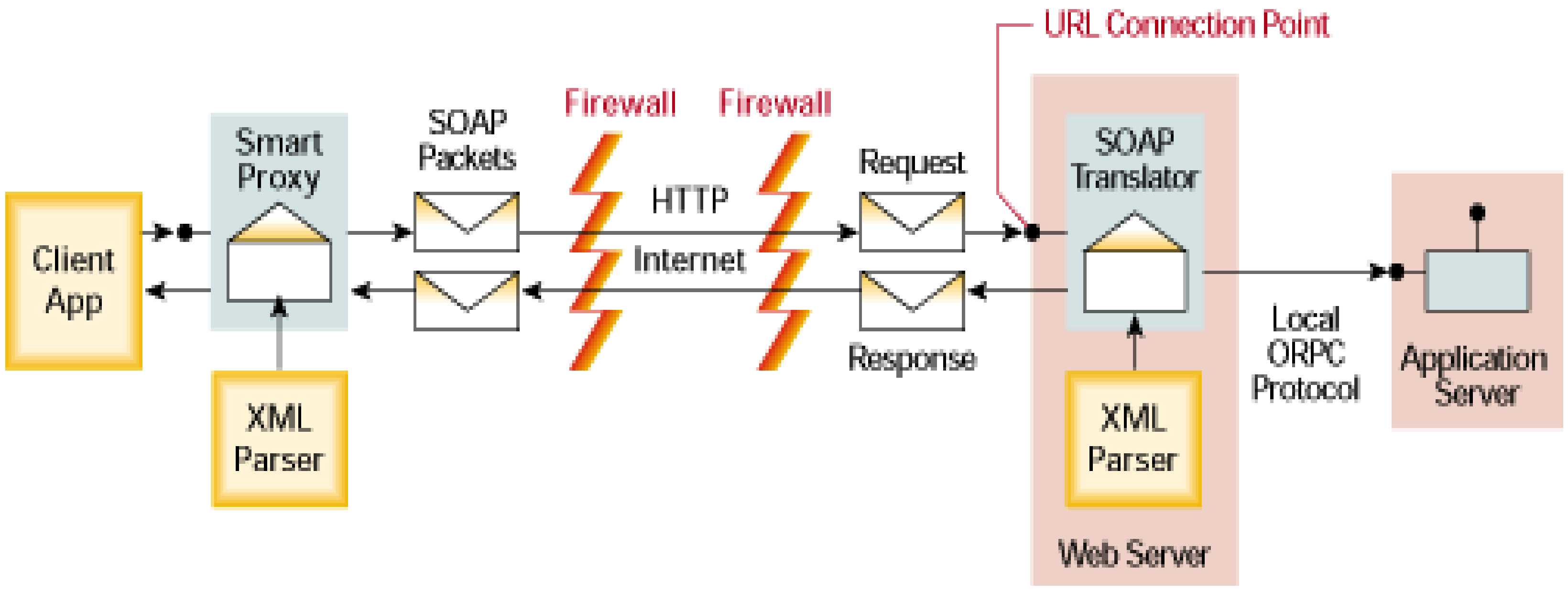
SOAP vs. DCOM/CORBA

- DCOM/CORBA:
  - client creates a *persistent* connection to the object to perform multiple property accesses and method calls.
- Both support stateful remote connections –
  - SOAP typically does not
    - could be implemented with server side state management of an application *(we will see in the next classes how WSRF maintains the state)*

SOAP Syntax Rules

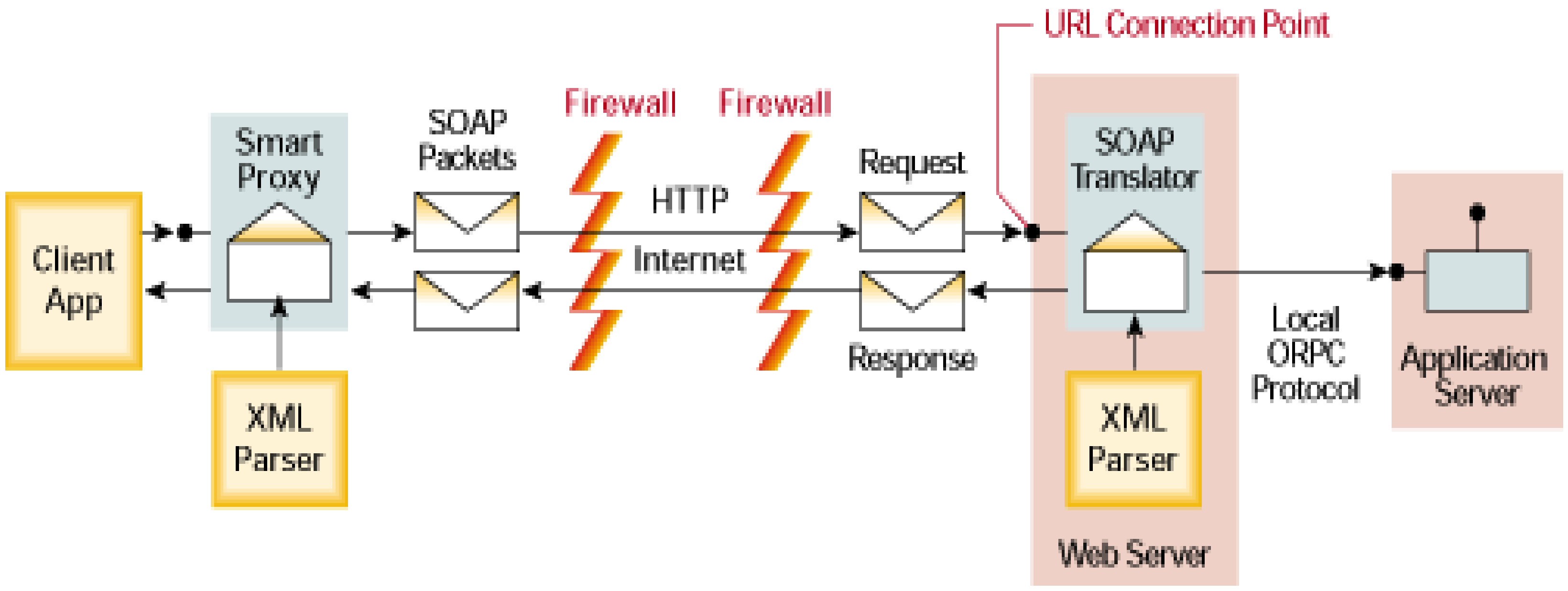
- A SOAP message **MUST** be encoded using XML
- A SOAP message **MUST** use the SOAP Envelope namespace
- A SOAP message **MUST** use the SOAP Encoding namespace
- A SOAP message **must NOT** contain a DTD reference
- A SOAP message **must NOT** contain XML Processing Instruction

SOAP Architecture



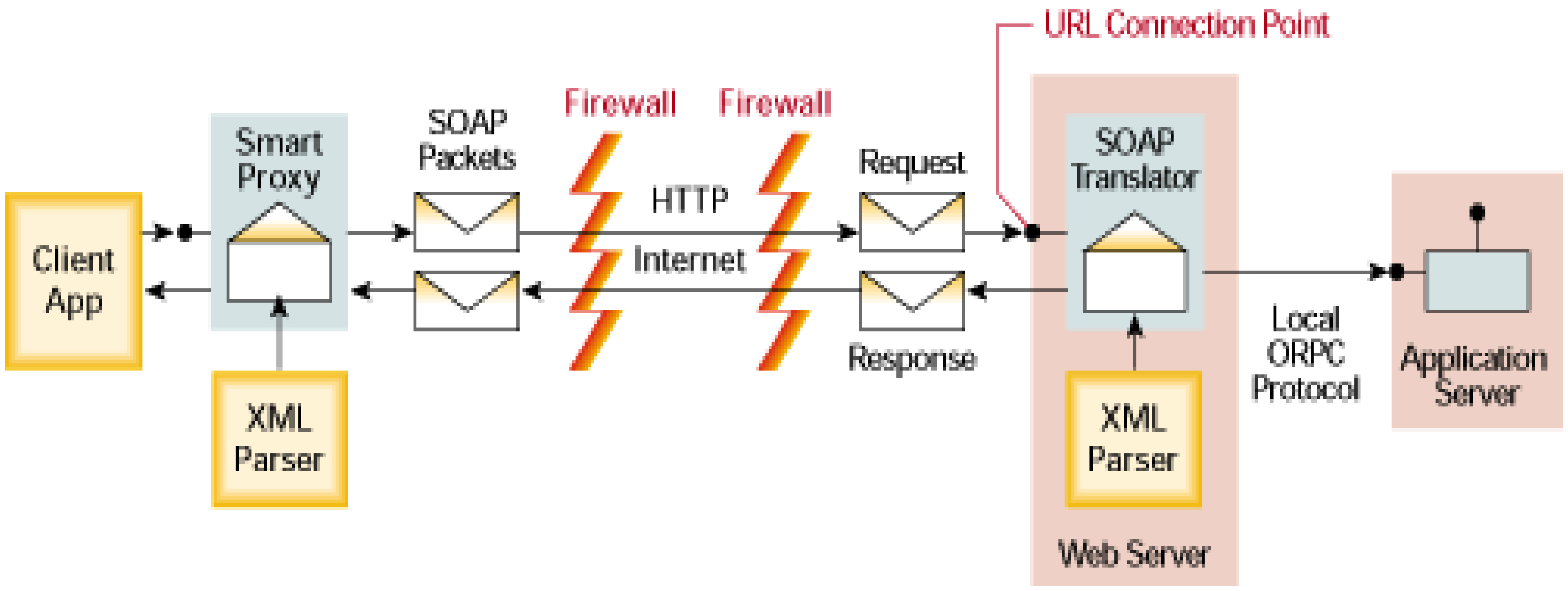
Source: eies.njit.edu/~turoff/ coursenotes/CIS679/sample/soap.doc

SOAP Architecture



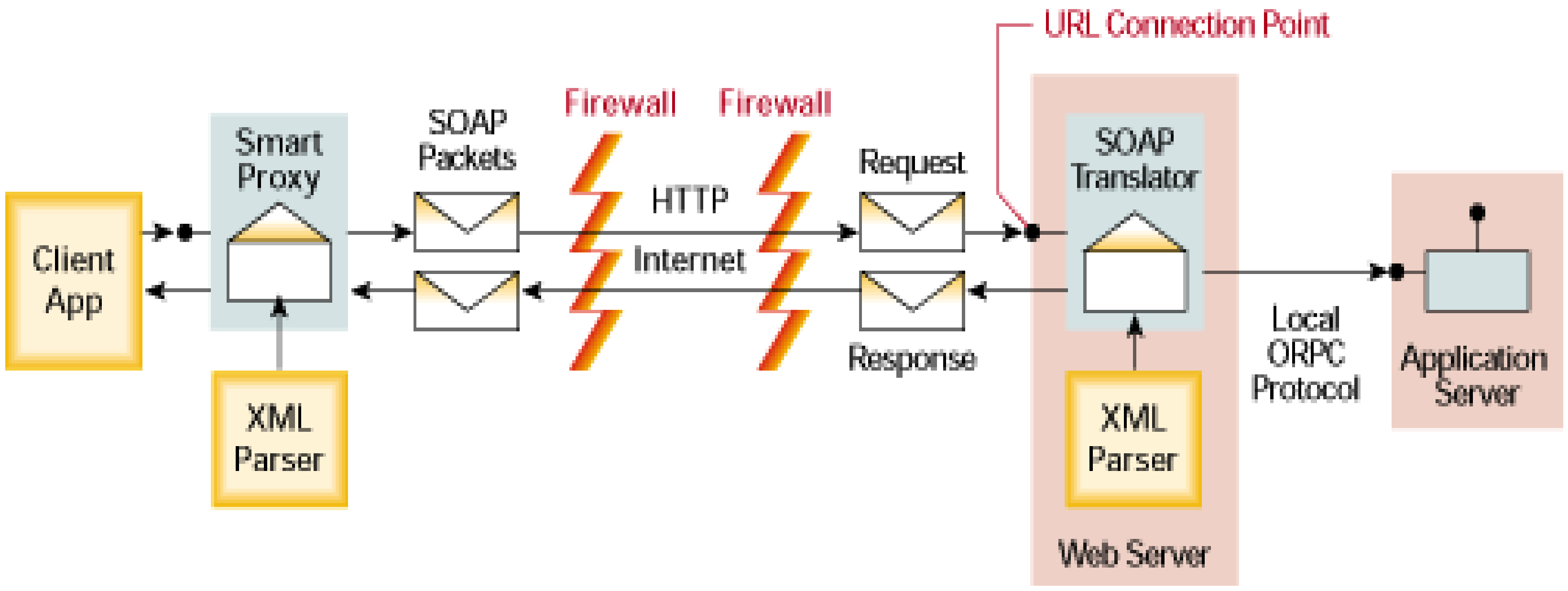
The client application calls a client-side proxy object using its native RPC protocol (such as COM for Microsoft Platform)

SOAP Architecture



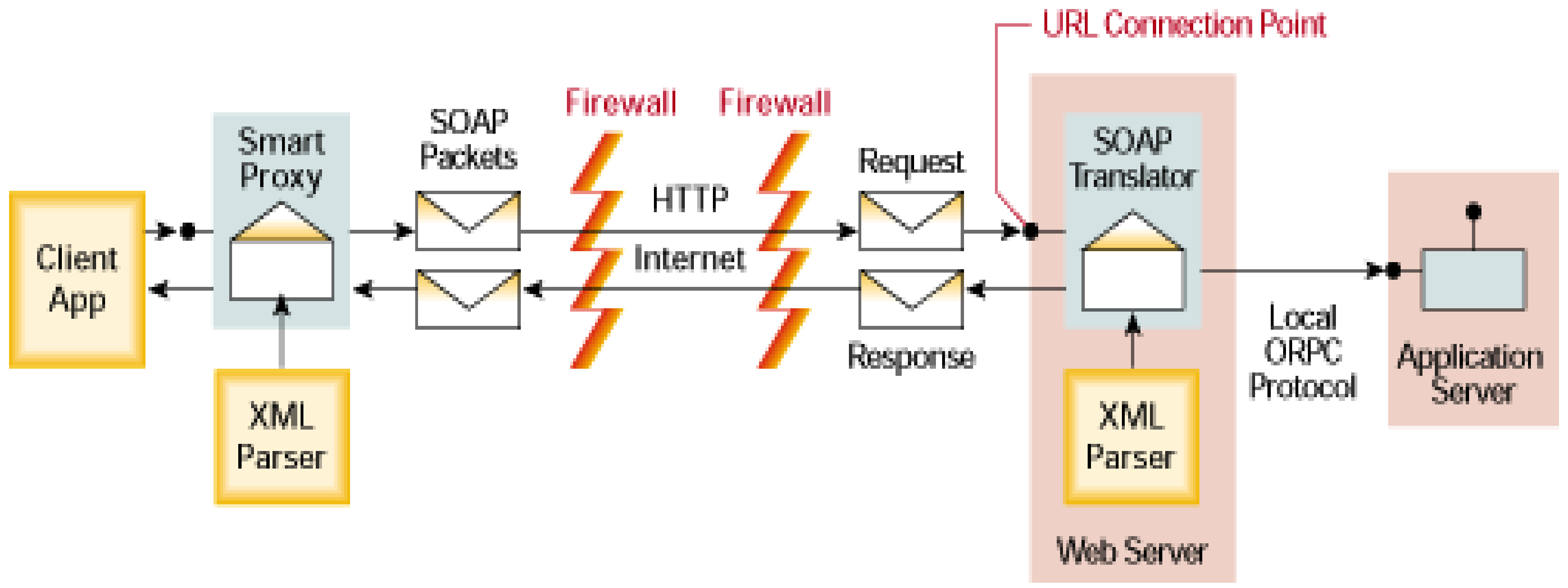
The proxy object uses an XML parser to convert the call into a SOAP packet.

SOAP Architecture



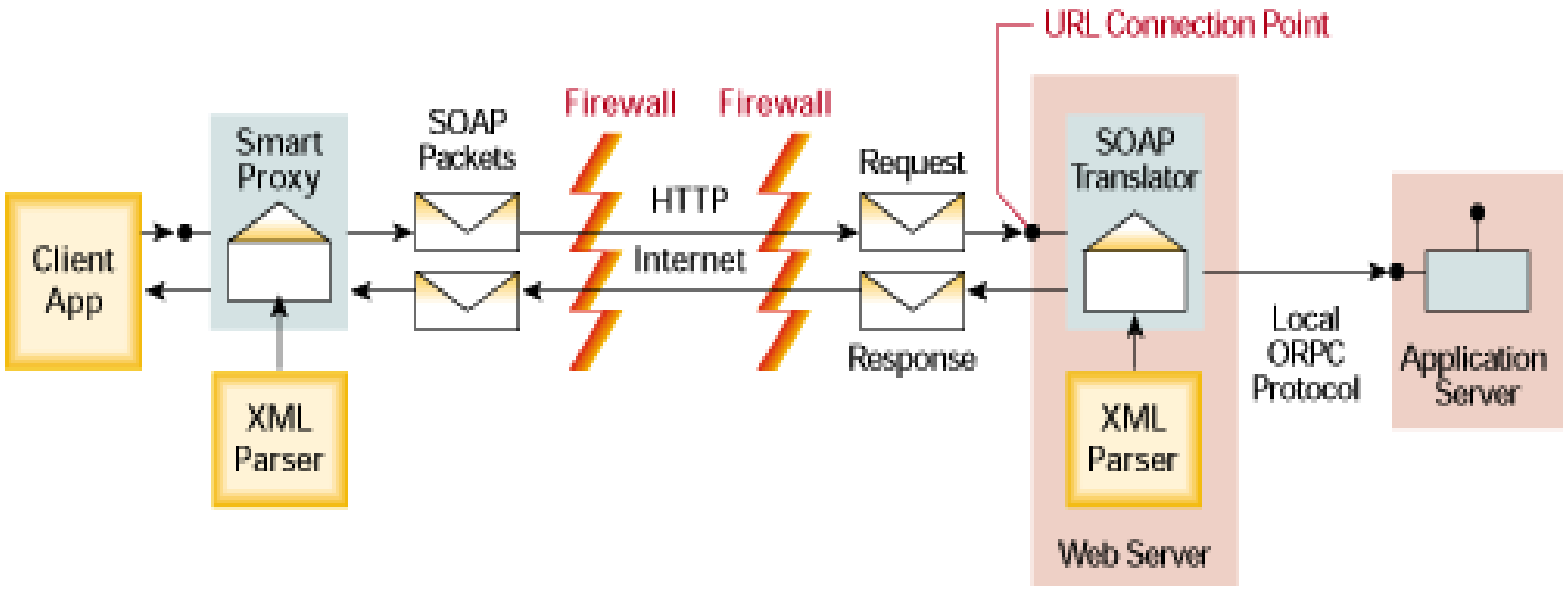
This SOAP packet is then transmitted over the Internet/Intranet to the web server using the HTTP protocol.

SOAP Architecture



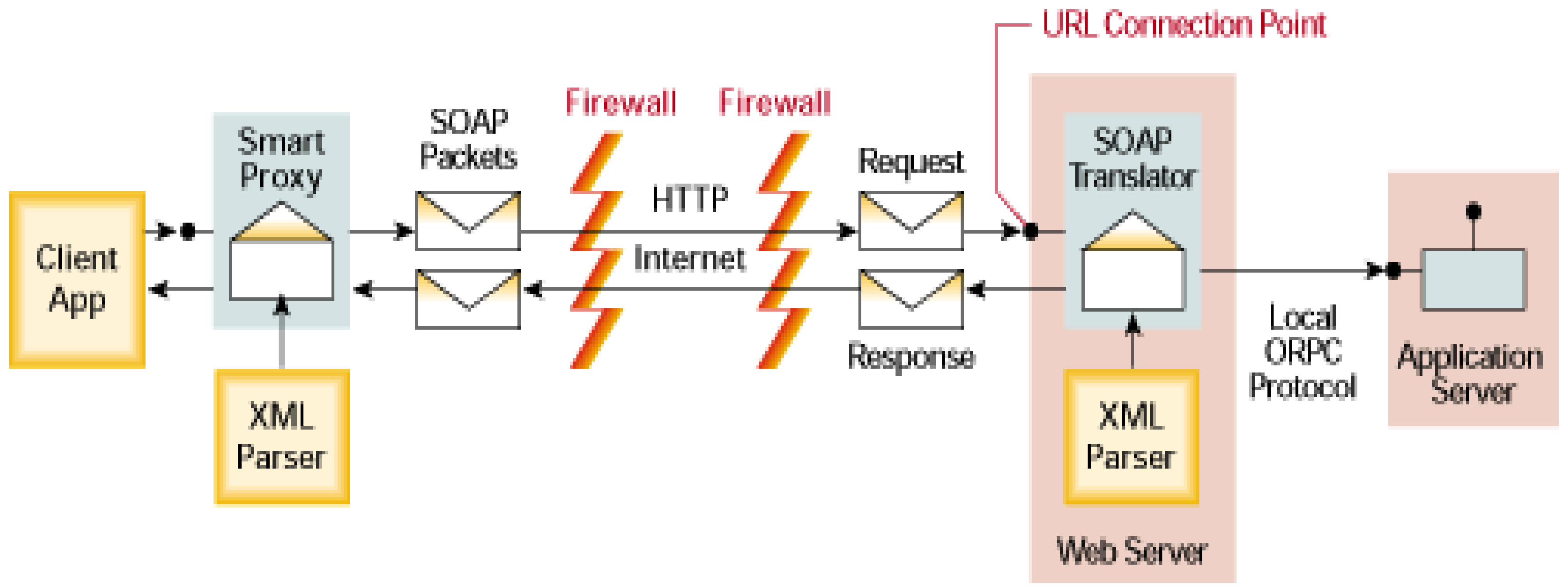
The Web server handles the URL connection point of the remote service, and launches a SOAP translator which may be an ASP page, an ISAPI extension, a CGI program, a Perl script, etc.

SOAP Architecture



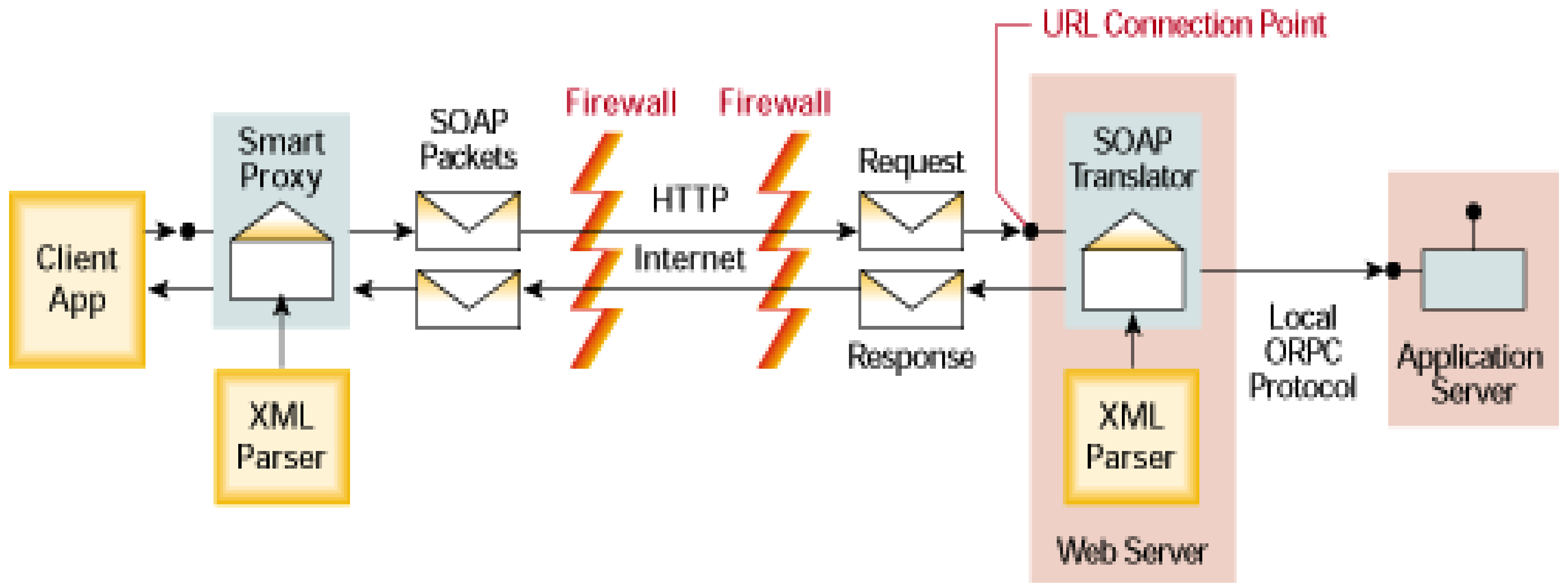
This translator uses a local XML parser to parse out the object name, method name and parameter values from the SOAP package.

SOAP Architecture



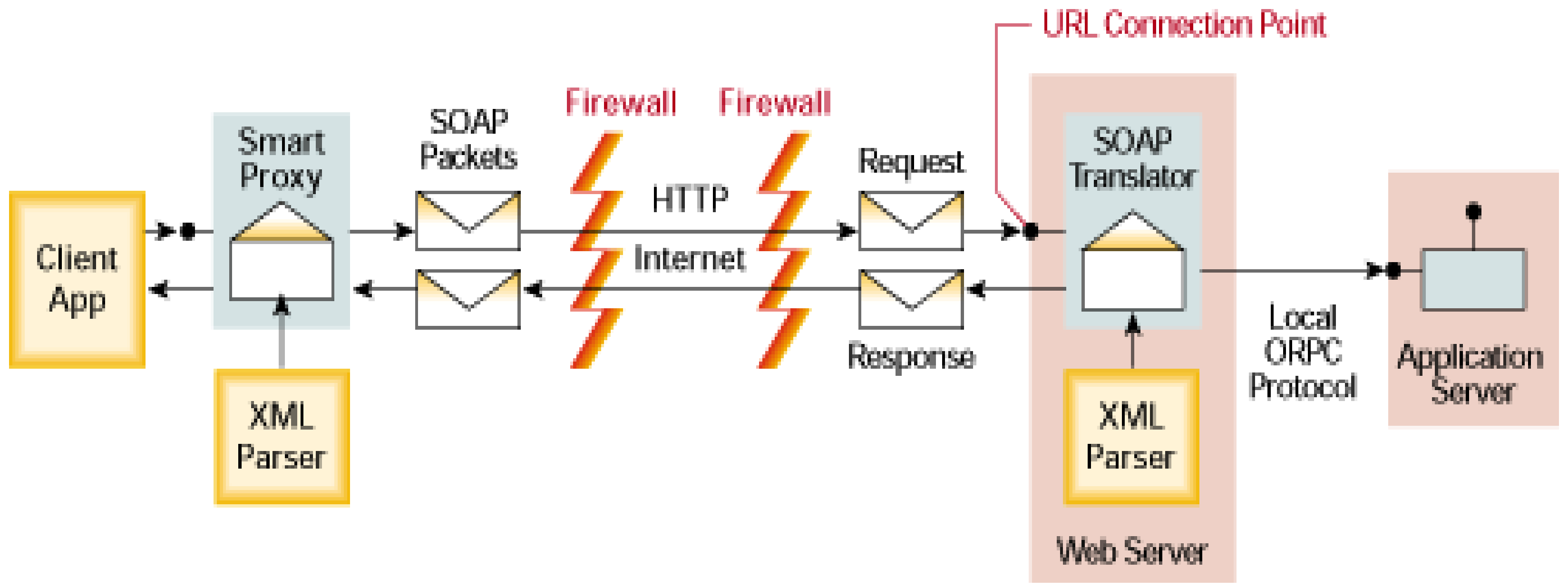
It uses these values to call the particular method of the server object by the local ORPC protocol, and packages the results into a response SOAP packet.

SOAP Architecture



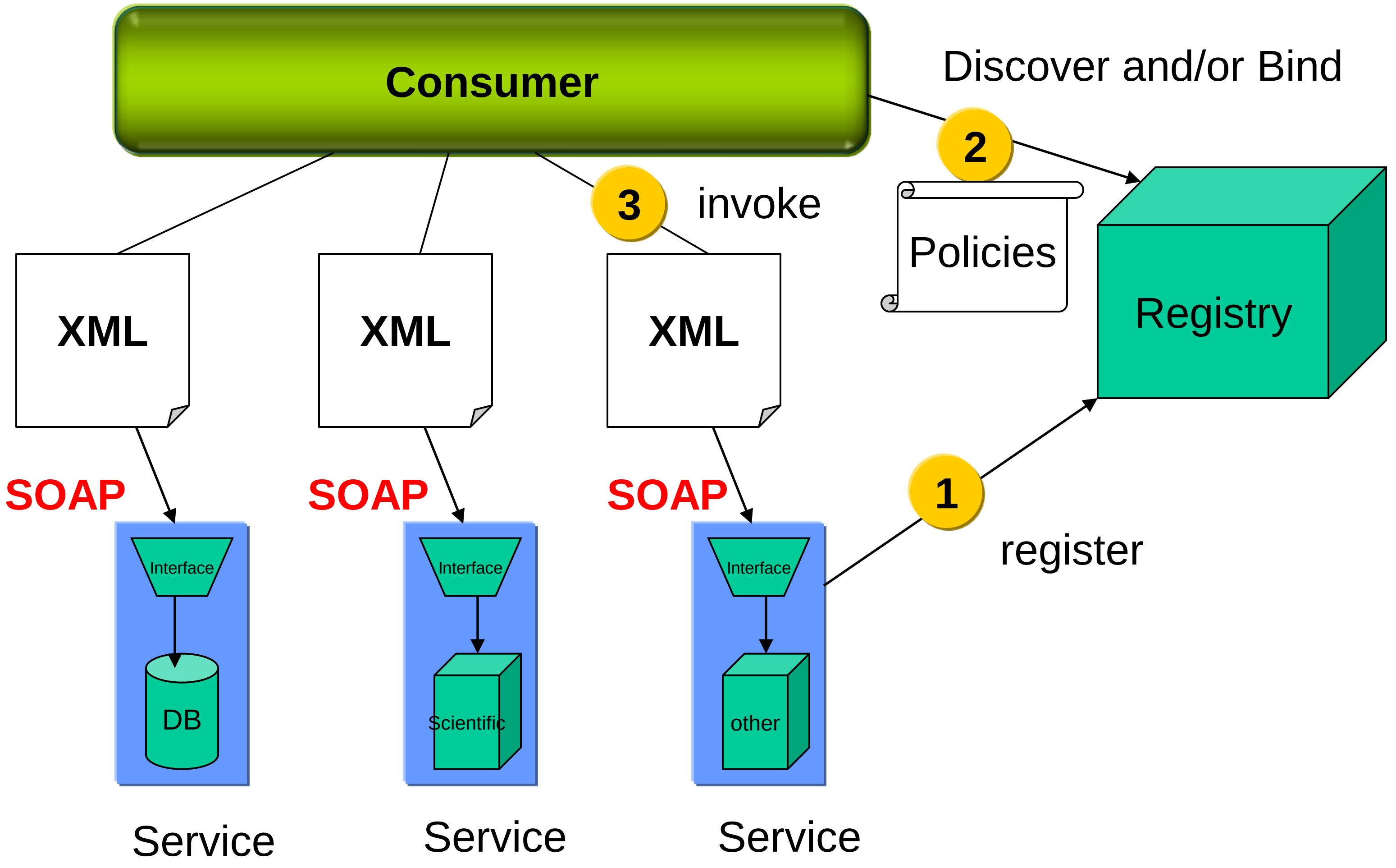
This translator uses a local XML parser to parse out the object name, method name and parameter values from the SOAP package.

SOAP Architecture



This response is unpackaged by the proxy and presented to the client.

Service Running Outside the Consumer Boundaries



SOAP Client and Server

- A SOAP client is a program that creates an XML document containing the information needed to invoke a method remotely in a distributed system. SOAP clients need not be traditional. A SOAP client could also be a Web server or a server-based application.

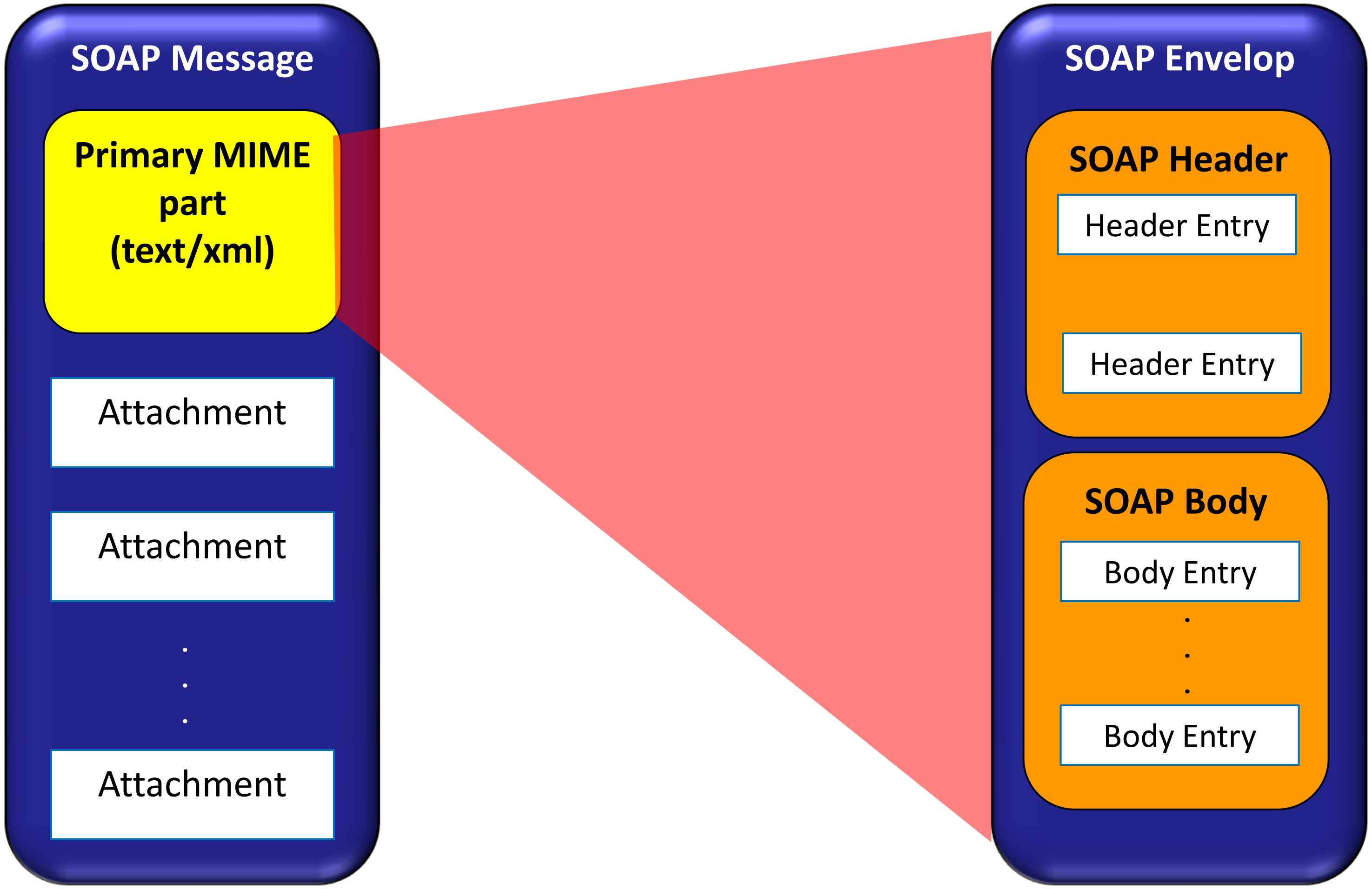
- Messages and requests from SOAP clients are typically sent over HTTP. As a result, SOAP documents are able to traverse almost any firewall, enabling the exchange of information across divergent platforms.

SOAP Client and Server

- A SOAP server is simply special code that listens for SOAP messages and acts as a distributor and interpreter of SOAP documents.

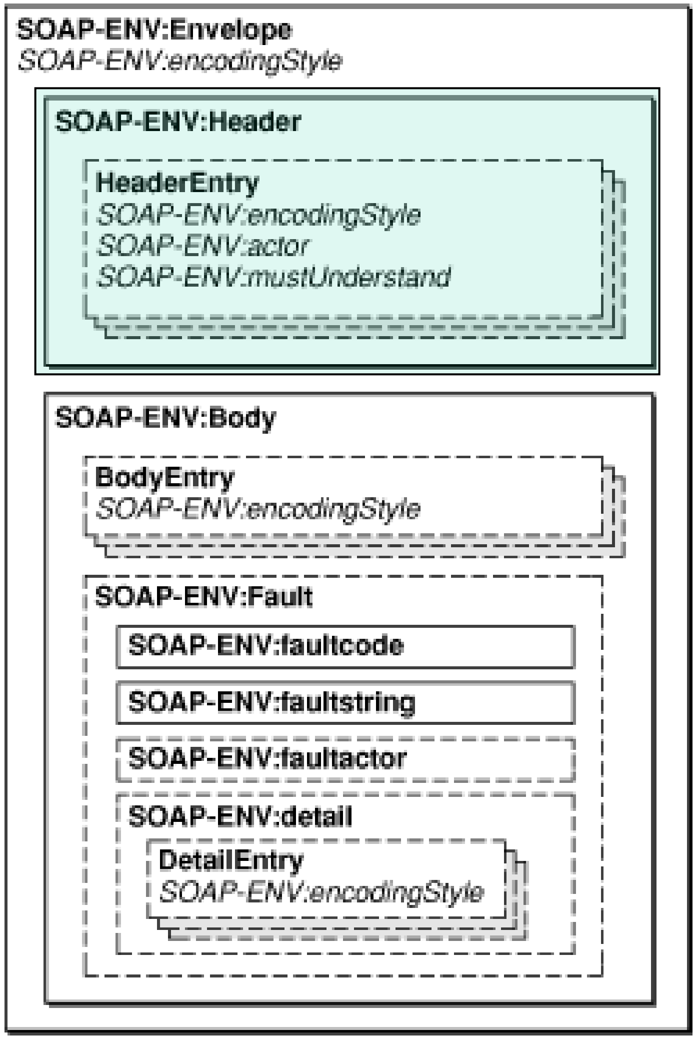
- SOAP servers ensure that documents received over a HTTP connection are converted to a language that the object at the other end understands. Because all communications are made in the form of XML, objects in one language (say, Java) may communicate through SOAP with objects in any other language (C++, for example). It's the job of the SOAP server to make sure the end points understand.

SOAP Message Format



SOAP Components and Elements

- Components:
  - Formatting conventions
  - Transport/protocol binding
  - Encoding rules
  - RPC mechanism
- Elements:
  - **Envelope**: Mandatory
  - **Header** [Optional] –
    - use: Authentication, how to process Transaction data
  - **Body**: Mandatory
    - use: Method call and its parameters



Skeleton SOAP Message

```
<?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:Header>
  ...
  ...
</soap:Header>
<soap:Body>
  ...
  ...
  <soap:Fault>
    ...
    ...
  </soap:Fault>
</soap:Body>
</soap:Envelope>
```

SOAP Envelope

- Root element of a SOAP message.
- Defines the XML document as a SOAP message
- The xmlns:soap Namespace:
  - message must have Envelope element associated with W3C namespace:
    - http://www.w3.org/2001/12/soap-envelope

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

SOAP Header(optional)

- Elements in SOAP Header define how a recipient should process the SOAP message
- Header elements must have attributes:
  - attributes in namespace ("http://www.w3.org/2001/12/soap-envelope")
- *actor*: used to address Header element to endpoint

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org
   /2001/12/soap-envelope"
soap:encodingStyle="http://ww
   w.w3.org/2001/12/soap-
   encoding">
<soap:Header>
<m:Trans
xmlns:m="http://www.w3schools
   .com/transaction/"
soap:actor="http://www.w3scho
   ols.com/appml/">
234
</m:Trans>
</soap:Header>
```

SOAP Header(optional)

- Headers, for example, may be used to provide digital signatures for a request contained in the body. In this circumstance, an authentication or authorization server could process the header entry – independent of the body -- stripping out information to validate the signature.
- Once validated, the rest of the envelope would be passed on to the SOAP server, which would process the body of the message.

SOAP Header(optional)

- The example contains a header with a "Trans" element, a "mustUnderstand" attribute with a value of 1, and a value of 234.
- SOAP defines three attributes in the default namespace
- These attributes are: mustUnderstand, actor, and encodingStyle.

```
<?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:Header>
       <m:Trans
     xmlns:m="http://www.w3schools.com/transaction/"
       soap:mustUnderstand="1">234
       </m:Trans>
     </soap:Header>
     ...
     ...
     </soap:Envelope>
```

SOAP Header(optional)

- If we add mustUnderstand="1" to a child element of the Header element it indicates that the receiver processing the Header must recognize the element.
- If the receiver does not recognize the element it will fail when processing the Header.

```
<?xml version="1.0"?>
     <soap:Envelope
     xmlns:soap="http://www.w3.org/20
     01/12/soap-envelope"
     soap:encodingStyle="http://www.w
     3.org/2001/12/soap-encoding">

     <soap:Header>
       <m:Trans
     xmlns:m="http://www.w3schools.c
     om/transaction/"
       soap:mustUnderstand="1">234
       </m:Trans>
     </soap:Header>
     ...
     ...
     </soap:Envelope>
```

SOAP Header(optional)

- **The actor Attribute**
- A SOAP message may travel from a sender to a receiver by passing different endpoints along the message path.
- However, not all parts of a SOAP message may be intended for the ultimate endpoint, instead, it may be intended for one or more of the endpoints on the message path.

```
<?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:Header>
      <m:Trans
    xmlns:m="http://www.w3schools.com/transaction/"
      soap:actor="http://www.w3schools.com/appml/">234
      </m:Trans>
    </soap:Header>
    ...
    ...
    </soap:Envelope>
```

SOAP Body

```
REQUEST:
<?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>
   <m:GetPrice
    xmlns:m="http://www.fruitmrkt.com/
    prices">
      <m:Item>Apples</m:Item>
   </m:GetPrice>
</soap:Body>
</soap:Envelope>
```

```
RESPONSE:
<?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>
   <m:GetPrice
    xmlns:m="http://www.fruitmrkt.com/
    prices">
      <m:Item>40</m:Item>
   </m:GetPrice>
</soap:Body>
</soap:Envelope>
```

- Body is XML data
- User namespace defined
- Fault: defined by SOAP standard for errors

SOAP HTTP Binding

application
/soap+xml

- A SOAP method is an HTTP request/response that complies with the SOAP encoding rules.
- HTTP Binding
- HTTP + XML = SOAP:
  - Content-type
- Protocol:
  - Client sends request
  - Server responds OK

```
POST /item HTTP/1.1
Host: 189.123.345.239
Content-Type: MIMEType;
    charset=character-encoding
    [optional]
Content-Length: 250


200 OK
Content-Type: text/plain
Content-Length: 250
```

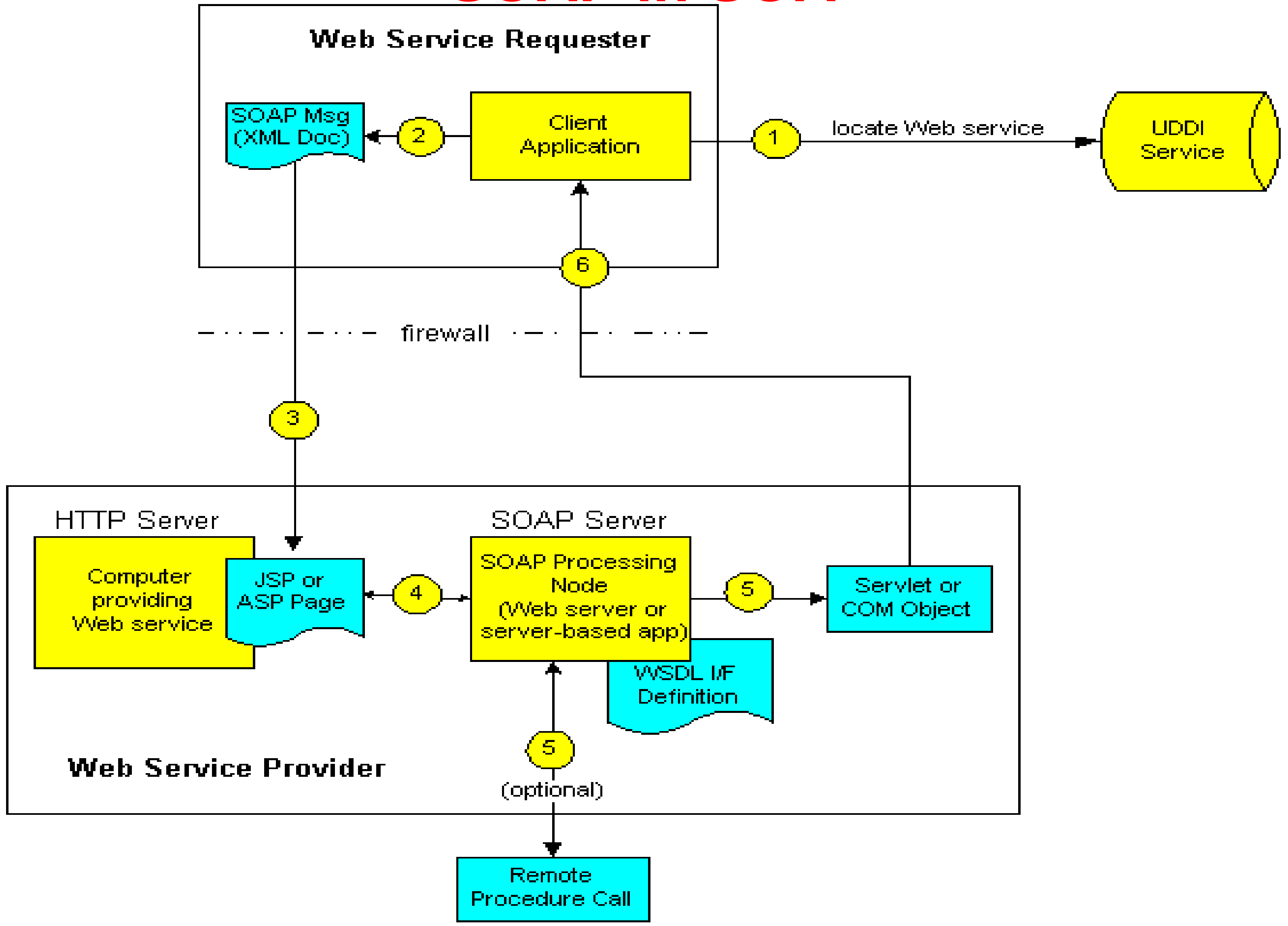
MIME (**M**ultipurpose **I**nternet **M**ail **E**xtensions) is an Internet standard for describes message content types.
MIME messages can contain text, images, audio, video, and other application-specific data. (extends format of e-mail)

SOAP Request/Response Example

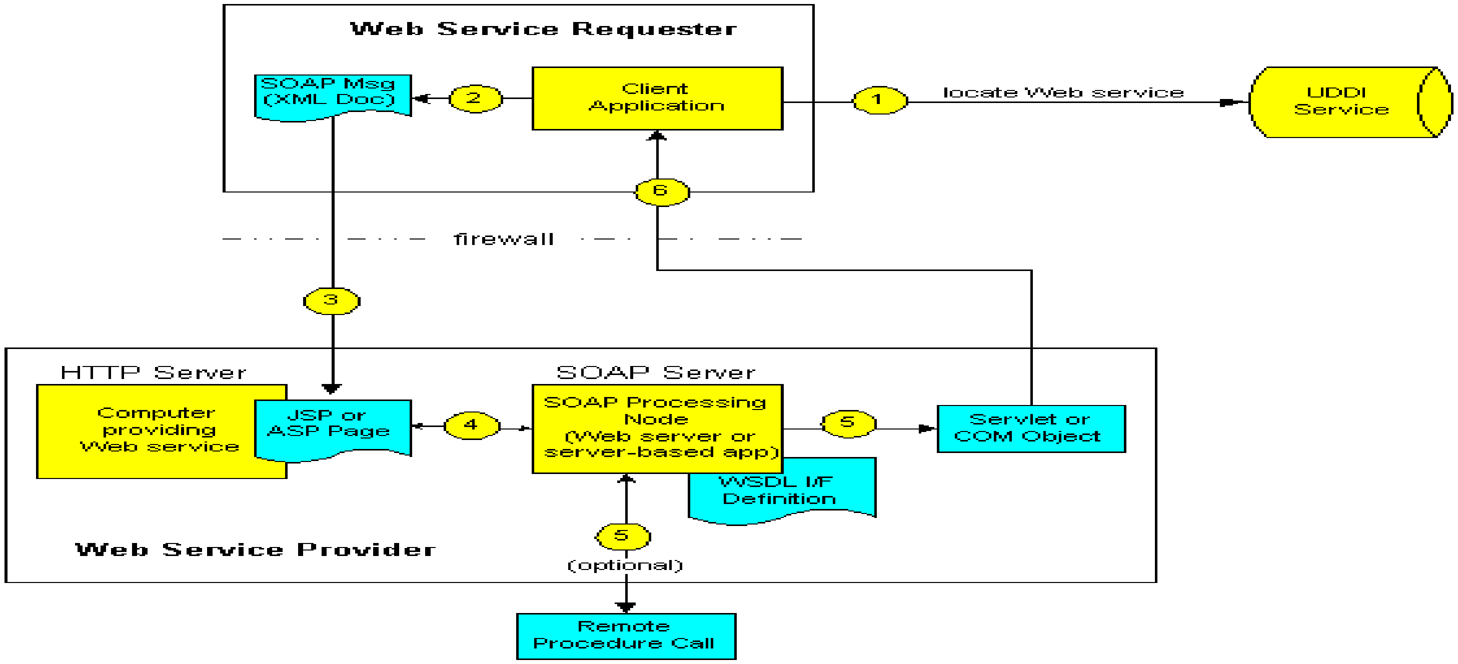
```
POST /InStock HTTP/1.1
Host: www.stock.org
Content-Type: application/soap+xml;
    charset=utf-8
Content-Length: nnn
SOAPAction:"http://www.stock.org/stock"
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/s
    oap-envelope"
soap:encodingStyle="http://www.w3.org/2
    001/12/soap-encoding">

  <soap:Body
    xmlns:m="http://www.stock.org/stock"
    >
    <m:GetStockPrice>
      <m:StockName>IBM</m:StockName>
    </m:GetStockPrice>
  </soap:Body>

</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml;
    charset=utf-8
Content-Length: nnn

<?xml version="1.0"?>

<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/s
    oap-envelope"
soap:encodingStyle="http://www.w3.org/2
    001/12/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 in SOA

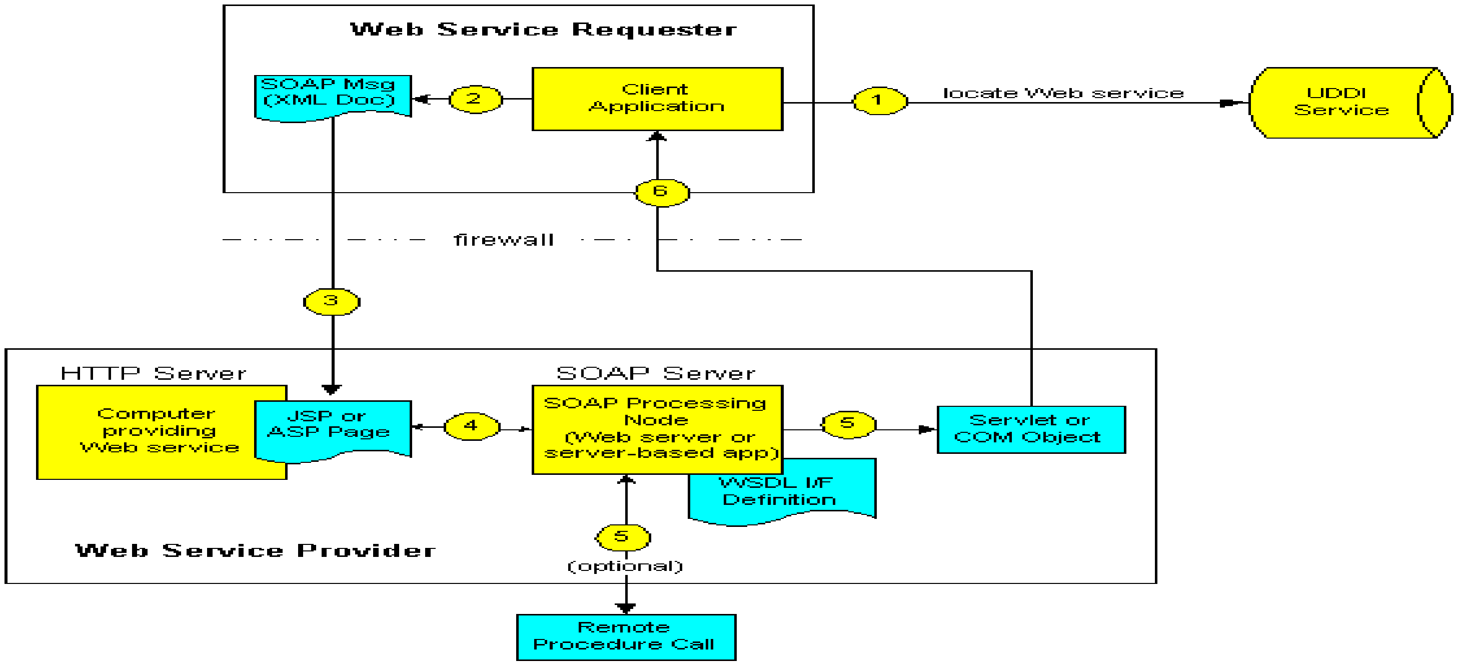

Web Service Requester
SOAP Msg (XML Doc)
2
Client Application
1
locate Web service
UDDI Service
6
firewall
3
HTTP Server
SOAP Server
Computer providing Web service
JSP or ASP Page
4
SOAP Processing Node (Web server or server-based app)
5
Servlet or COM Object
WSDL I/F Definition
5
(optional)
Web Service Provider
Remote Procedure Call

SOAP in SOA



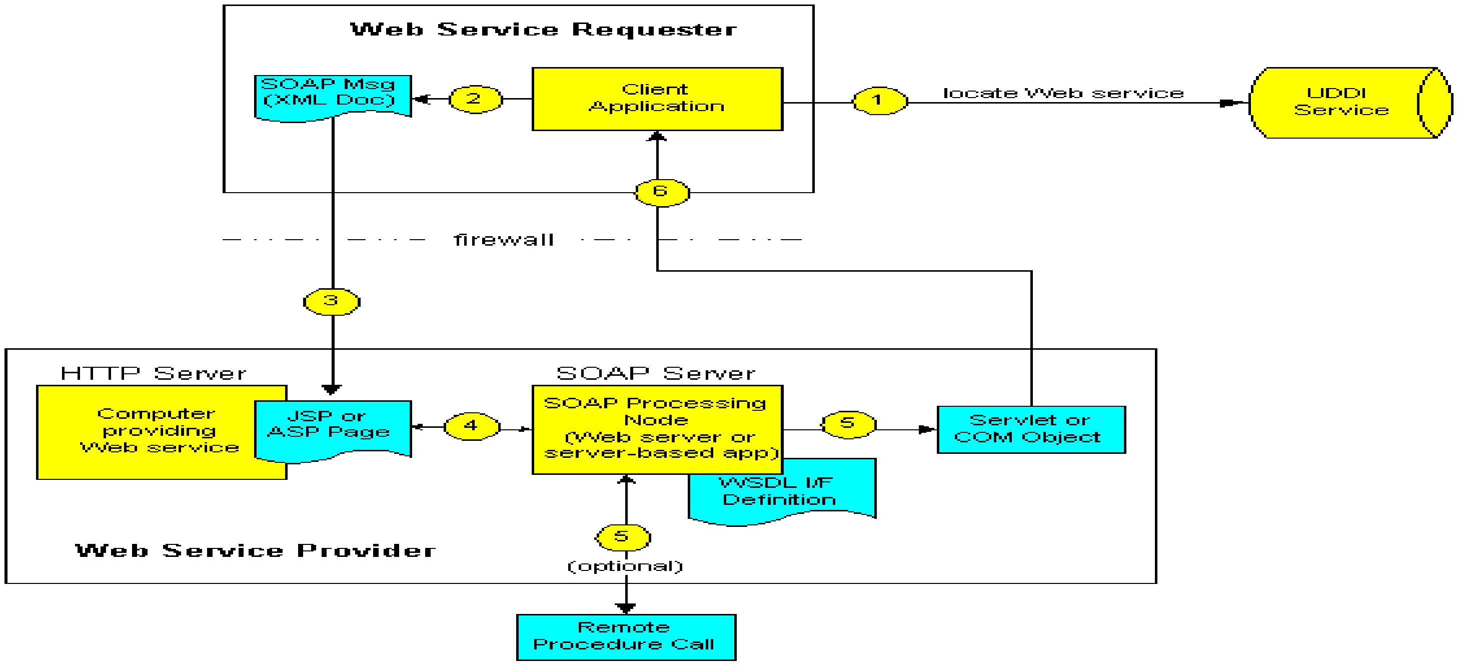
1. A SOAP client uses the UDDI registry to locate a Web service. Rather than manipulate WSDL directly, in most cases a SOAP application will be hardwired to use a particular type of port and style of binding, and it will dynamically configure the address of the service to be invoked to match the ones discovered through UDDI.

SOAP in SOA



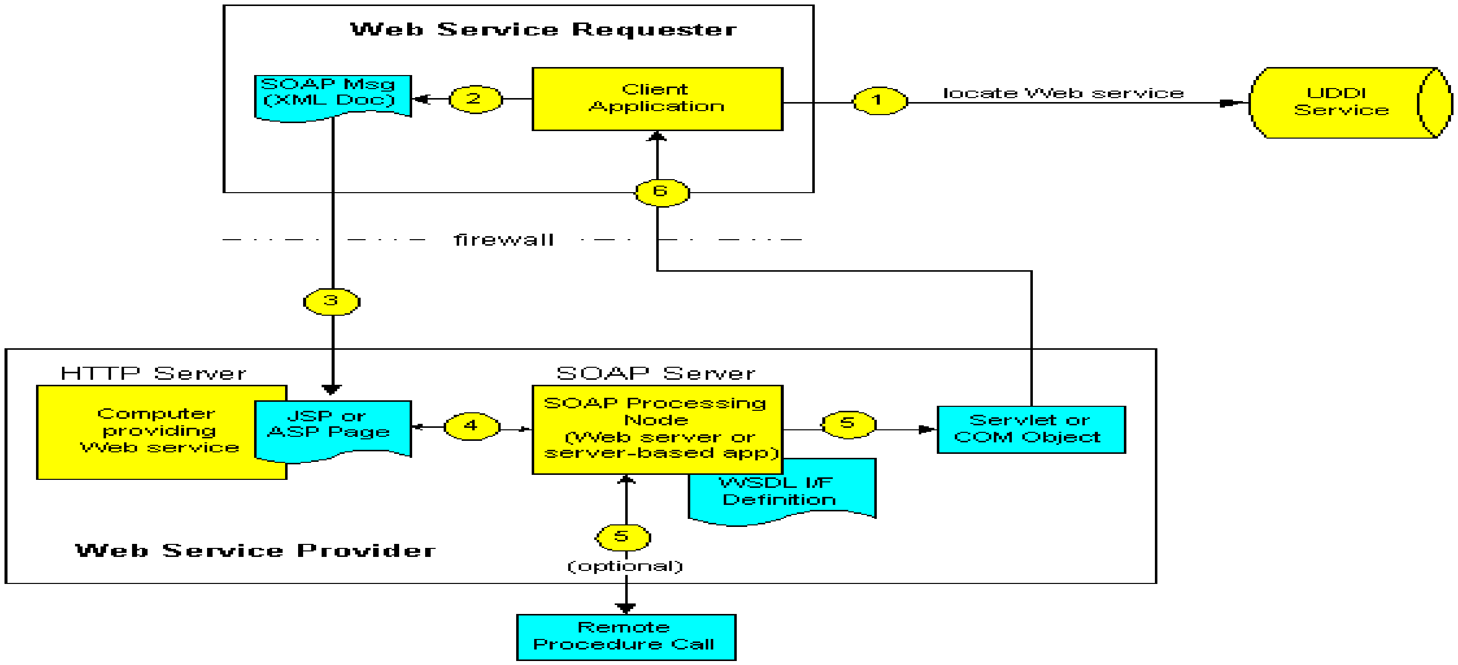
2. The client application builds a SOAP message, which is an XML document capable of performing the desired request/response operation.

SOAP in SOA



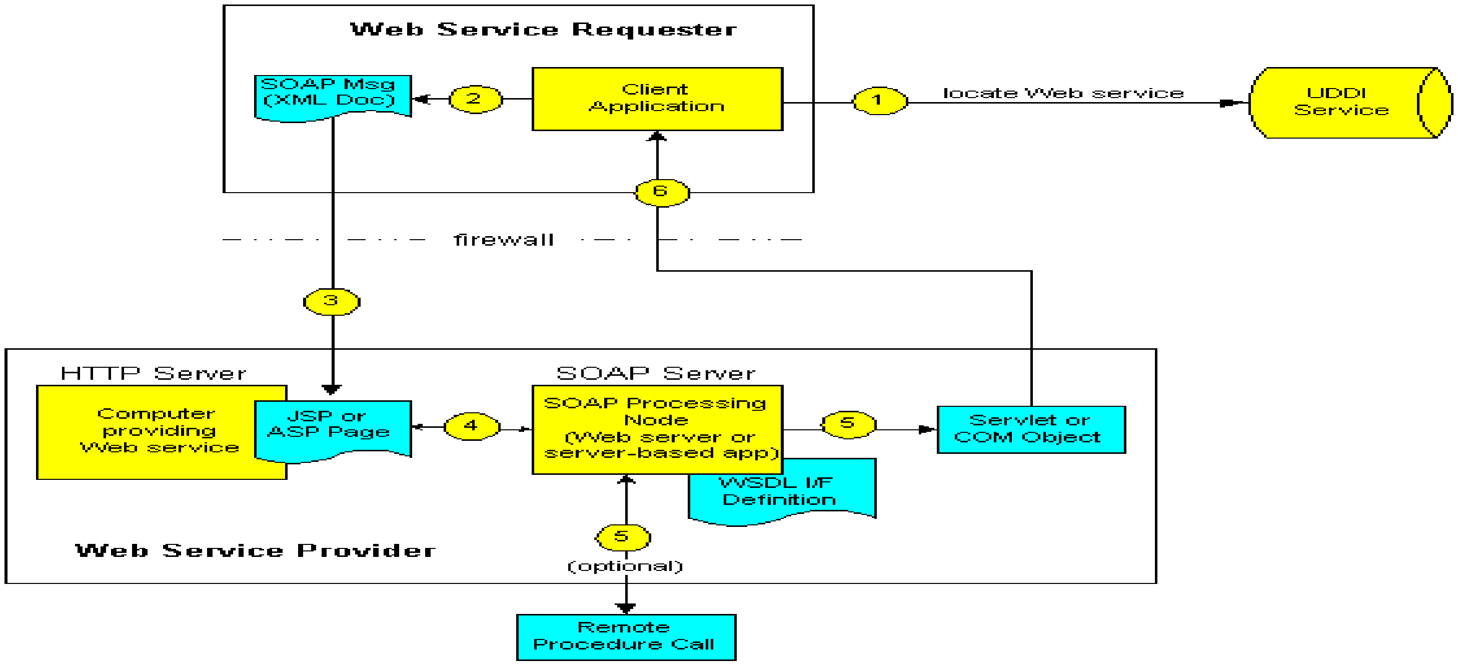
3. The client sends the SOAP message to a JSP or ASP page on a Web server listening for SOAP requests.

SOAP in SOA



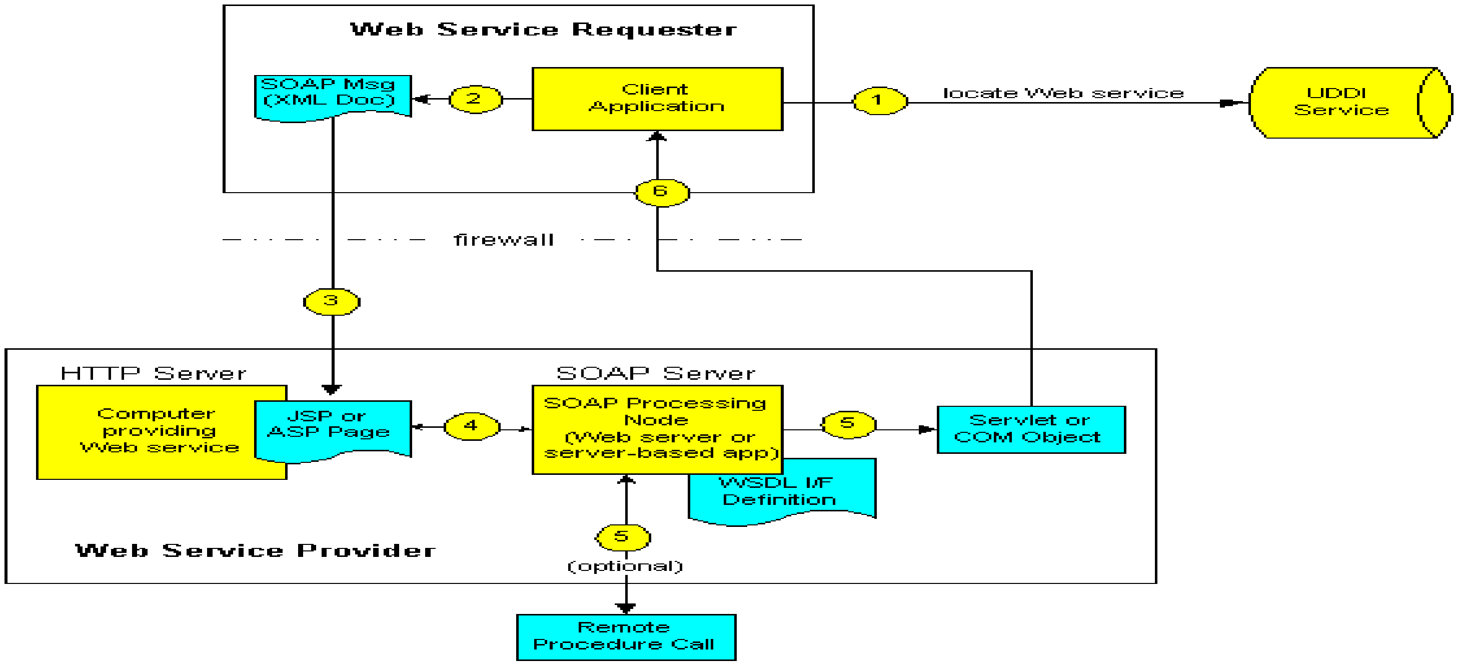
4. The SOAP server parses the SOAP package and invokes the appropriate method of the object in its domain, passing in the parameters included in the SOAP document. Optionally, intermediate processing nodes may have performed special functions as indicated by SOAP headers prior to receipt of the message by the SOAP server.

SOAP in SOA



5. The request object performs the indicated function and returns data to the SOAP server, which packages the response in a SOAP envelope. The server wraps the SOAP envelope in a response object, such as a servlet or a COM object, which is sent back to the requesting machine.

SOAP in SOA



6. The client receives the object, strips off the SOAP envelope and sends the response document to the program originally requesting it, completing the request/response cycle.

What is a Web Server (Definition)

- A computer, including software package, that provides a specific kind of service to client software running on other computers. More specifically, a server is a computer that manages and shares web based applications accessible anytime from any computer connected to the Internet.
- A term often used to describe a computer that hosts a Web site. In fact the term refers to software running on that computer allowing Web pages to be requested and then sent to a user's Web browser.
- Software that sends web site pages back to browsers. Also referred to as an HTTP daemon (HTTP being the protocol used for web pages).

What is a Web Server (Definition)

- Although web server is referred as a machine, it is actually a process running on a machine that serves HTTP content to web browsers on client machines. Along the same lines, there are also FTP servers, mail servers, and so on, each of which handles a specific type of traffic.
- The term web server can mean one of two things:
  - ➢ a computer responsible for serving web pages, mostly HTML documents, via the HTTP protocol to clients, mostly web browsers;
  - ➢ a software program that is working as a daemon serving web documents.

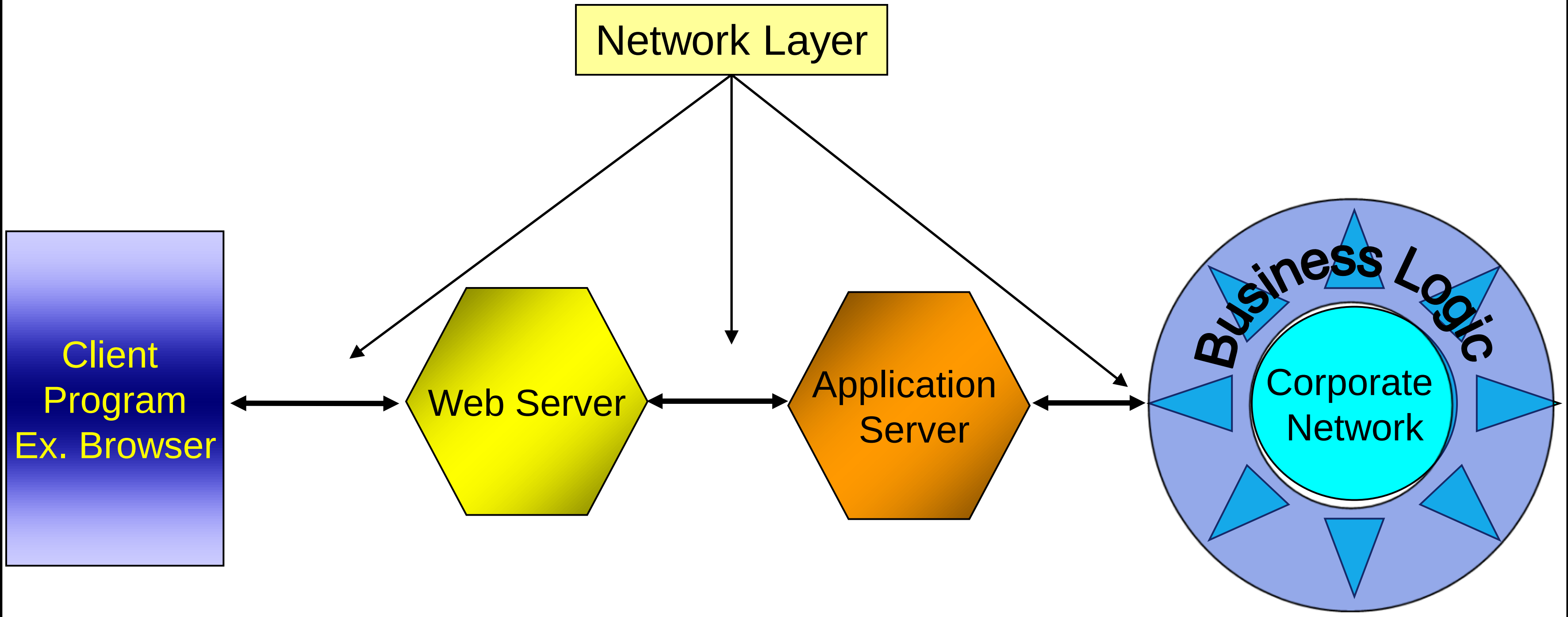
What is a Application Server (Definition)

- A "middle-tier" software and hardware combination that lies between the Web server and the corporate network and systems.
- A specialized network server whose job is to provide access to a client/server application and, sometimes, the data that belongs to that application as well.
- A software platform that provides the services and infrastructure required to develop and deploy middle-tier applications. Middle-tier applications perform the business logic necessary to provide web clients with access to enterprise information systems. In a multi-tier architecture, an application server sits beside a web server or between a web server and enterprise information systems.

What is a Application Server (Definition)

- An application server is a server computer in a computer network dedicated to running certain software applications. The term also refers to the software installed on such a computer to facilitate the serving (running) of other applications

Structure of Servers (Web & Application)



Comparison

- Web server handles the HTTP protocol. When receives an HTTP request, it responds with an HTTP response. To process a request, a Web server may respond with a static HTML page or image, send a redirect, or delegate the dynamic response generation to some other program such as CGI (Common Gateway Interface) scripts, JSPs (JavaServer Pages), servlets, ASPs (Active Server Pages), server-side JavaScripts, or some other server-side technology. Whatever their purpose, such server-side programs generate a response, most often in HTML, for viewing in a Web browser.

Comparison

- The application server exposes business logic to client applications through various protocols, possibly including HTTP. While a Web server mainly deals with sending HTML for display in a Web browser, an application server provides access to business logic for use by client application programs. The application program can use this logic just as it would call a method on an object (or a function in the procedural world).

Comparison

- Recently, XML Web services have blurred the line between application servers and Web servers. By passing an XML payload to a Web server, the Web server can now process the data and respond much as application servers have in the past, so many times web server is referred to as application server also.

Web Computing Vs. Servlet Programming

- Enterprises are using XML for the integration of data, both internally for sharing legacy data among departments, and externally for sharing data with other enterprises.
- Data integration that XML offers, it has become the underpinning for Web-related computing.

Web Computing Vs. Servlet Programming

- In a Web service, a server application is deployed on a server-side container. The container can be a servlet container such as Tomcat or a Java™ 2 Platform, Enterprise Edition (J2EE™) container that is based on Enterprise Java- Beans™ (EJB™) technology.

Web Computing Vs. Servlet Programming

- Web service can make itself available to potential clients by describing itself in a Web Services.
- Description Language (WSDL) document. A WSDL description is an XML document that gives all the pertinent information about a Web service, including its name, the operations that can be called on it, the parameters for those operations, and the location of where to send requests. A consumer (Web client) can use the WSDL document to discover what the service offers and how to access it.

Web Computing Vs. Servlet Programming

- Perhaps the most important requirement for a Web service is that it be interoperable across clients and servers.
- With web service, a client written in a language other than the Java programming language can access a Web service developed and deployed on the Java platform. Conversely, a client written in the Java programming language can communicate with a service that was developed and deployed using some other platform.

Web Computing Vs. Servlet Programming

- The most obvious difference between Servlet and Web Service is we access servlet via HTTP while access Web Service via SOAP
- We can not directly invoke a servlet, we can only open URL connection and put some parameter to the servlet if the caller is out of application. And we can not restrict what parameters the caller can put. The caller does not know what parameters our servlet can receive either.
- WSDL file of our web service can give the caller enough information to invoke our web service.

Web Computing Vs. Servlet Programming

- WS and Servlet are very different. Servlet is more like Server side VB Script or Javascript. Servlet is invoked on the server side, but the web services are invoked directly from the client side. Servlet can use web service when it needs to invoke some function on the other server etc.

Web Services
Part II