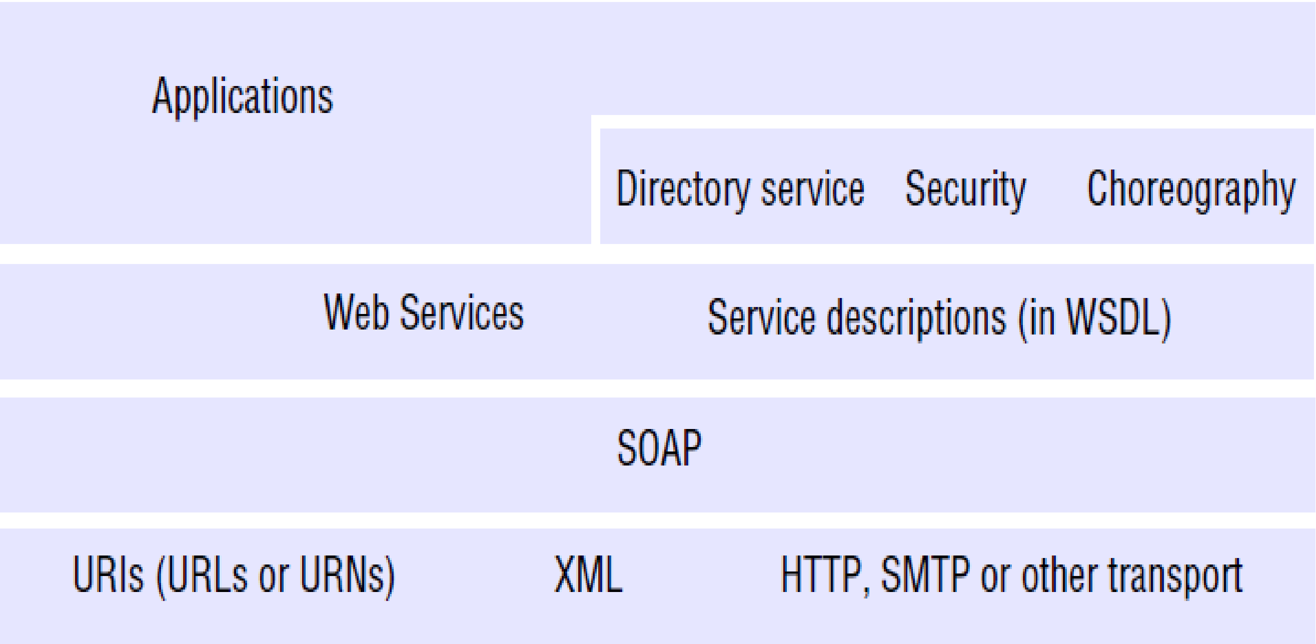
Chapter 9

WEB SERVICES

- A web server provides a basic HTTP service, whereas a web service provides a service based on the operations defined in its interface.
- Web services have arisen from the need to provide an infrastructure to support interworking between different organizations.
- This infrastructure generally uses the widely used HTTP protocol to transport messages between clients and servers over the Internet and is based on the use of URIs to refer to resources.
- XML, a textual format, is used for data representation and marshalling.
- A web service generally provides a *service description*, which includes an interface definition and other information, such as the server's URL.

- Two separate influences led to the emergence of web services.
- One of these was the addition of service interfaces to web servers with a view to allowing the resources on a site to be accessed by client programs other than browsers and using a richer form of interaction.
- The other was the desire to provide something like RPC over the Internet, based on the existing protocols.
- The resulting web services provide interfaces with sets of operations that can be called remotely.
- Like any other form of service, a web service can be the client of another web service, thus allowing a web service to integrate or combine a set of other web services.
- The Uniform Resource Identifier (URI) is a general resource identifier, whose value may be either a URL or a URN. URLs, which include resource location information such as the domain name of the server of a resource being named, are well known to all web users.
- Uniform Resource Names (URNs) are location-independent – they rely on a lookup service to map them onto the URLs of resources

Figure 9.1 Web services infrastructure and components



External data representation and marshalling of messages exchanged between clients and web services is done in XML

The SOAP protocol specifies the rules for using XML to package messages, for example to support a request-reply protocol

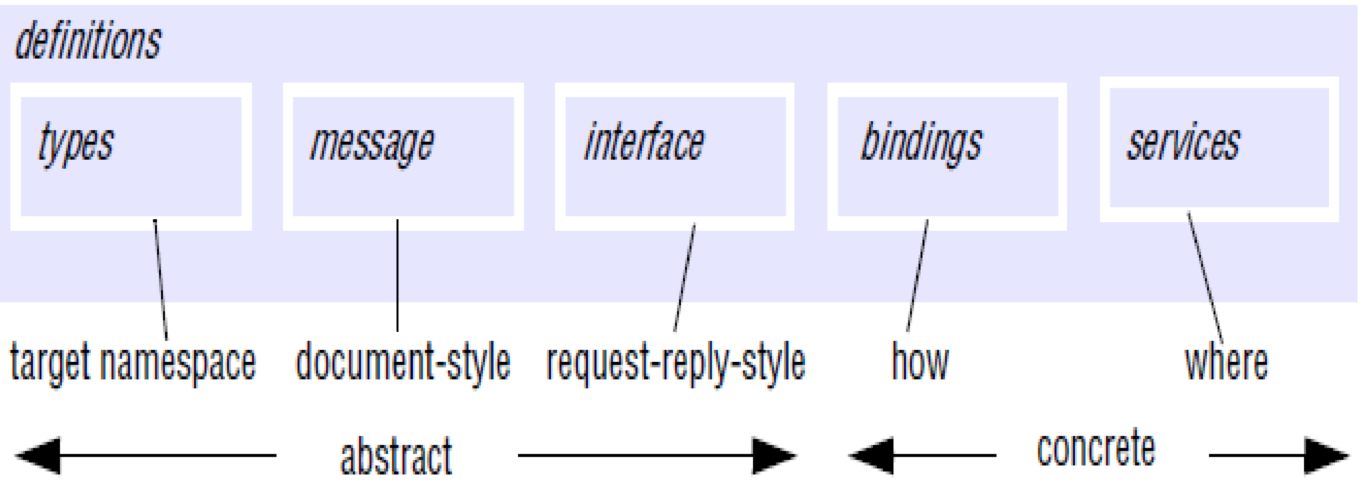
SOAP is used to encapsulate these messages and transmit them over HTTP or another protocol, for example, TCP or SMTP. A web service deploys service descriptions to specify the interface and other aspects of the service for the benefit of potential clients

Comparison of web services with CORBA

- The main difference between web services and CORBA or other similar middleware is the intended usage context. CORBA was designed for use within a single organization or between a small number of collaborating organizations.
- The design of the CORBA Naming Service effectively restricts the sharing of CORBA objects to within a small set of organizations that have federated their naming services.
- In contrast, in the web services model, a service is identified by means of a URL, enabling a client anywhere in the Internet to make a request to a service that may belong to any organization anywhere else.
- The tasks of locating and activating web services are neatly separated. In contrast, a CORBA persistent reference refers to a component of the platform (the implementation repository) that activates the corresponding object on demand on any suitable computer and is also responsible for locating the object once it has been activated.
- The CORBA platform is a large and complex piece of software requiring installation and support whereas HTTP and XML infrastructure is used for web services which is easy to use and well understood.
- CORBA has been designed to be efficient: CORBA CDR is binary, whereas XML is textual.
- The availability of CORBA services for transactions, concurrency control, security and access control, events and persistent objects makes it a desirable choice.

9.3 Service Descriptors & IDL for web services

Figure 9.10 The main elements in a WSDL description



For web services, interface definitions are provided as part of a more general *service description*, which specifies two other additional characteristics – how the messages are to be communicated (for example, by SOAP over HTTP) and the URI of the service. To cater for use in a multi-language environment, service descriptions are written in XML.

WSDL separates the abstract part of a service description from the concrete part, as shown in Figure 9.10. In the web services context, the Web Services Description Language (WSDL) is commonly used for service descriptions. The current version, WSDL 2.0

WSDL

- The set of names defined within the *types* section of a WSDL definition is called its *target namespace*.
- The *message* section of the abstract part contains a description of the set of messages exchanged. For the document style of interaction, these messages will be used directly.
- For the request-reply style of interaction, there are two messages for each operation, which are used to describe the operations in the *interface* section.
- The concrete part specifies how and where the service may be contacted.
- The inherent modularity of a WSDL definition allows its components to be combined in different ways – for example, the same interface may be used with different bindings or locations.

- SOAP is the communication protocol that is generally used by web services and their clients.
- It can be used to transmit request messages and their replies between client and server, either by the asynchronous exchange of documents or by a form of request reply protocol based on a pair of asynchronous message exchanges.
- In both cases, the request or reply message is enclosed in an XML-formatted document called an envelope.
- The SOAP envelope is generally transmitted over the synchronous HTTP protocol, although other transports can be used.
- XML and SOAP processors are available for all of the widely used programming languages and operating systems. This enables web services and their clients to be deployed almost anywhere.
- This form of interworking is enabled by the facts that web services are not tied to any particular programming language and do not support the distributed object model

- In conventional middleware services, interface definitions provide clients with the details of services. However, in the case of web services, service descriptions are used.

- A service description specifies the communication protocol to be used (for example, SOAP) and the URI of the service, as well as describing its interface. The interface may be described either as a set of operations or as a set of messages to be exchanged between client and server.

9.4 A directory service for use with web services

- Any organization that plans to base its applications on web services will find it more convenient to use a directory service to make these services available to clients.
- This is the purpose of the Universal Description, Discovery and Integration service (UDDI) which provides both a name service and a directory service.
- That is, WSDL service descriptions may be looked up by name (a white pages service) or by attribute (a yellow pages service).
- They may also be accessed directly via their URLs, which is convenient for developers who are designing client programs that use the service.
- UDDI provides an API for looking up services based on two sets of query operations: *get_xxx* set and *find_xxx* set of operations. Examples - *get_BusinessDetail*, *get_ServiceDetail, find_service*, *find_binding, etc.*

9.5 XML SECURITY

- XML security was designed to provide the necessary protection for the contents of a document exchanged by members of a group of people, who have different tasks to perform on that document.
- Different parts of the document will be available to different people, some with the ability to add to or alter the content and others only to read it.
- To enable complete flexibility in its future use, the security properties are defined within the document itself.
- This is achieved by means of XML, which is a self-describing format.
- XML elements are used to specify document parts that are encrypted or signed as well as details of the algorithms used and information to help with finding keys.

- Web services have been used for a variety of purposes in distributed systems. For example, web services provide a natural implementation of the concept of service oriented architecture, in which their loose coupling enables interoperability in Internetscale applications – including business-to-business (B2B) applications.

- Their inherent loose coupling also supports the emergence of a mashup approach to web service construction.

- Web services also underpin the Grid, supporting collaborations between scientists or engineers in organizations in different parts of the world. Their work is very often based on the use of raw data collected by instruments at different sites and then processed locally.

- The Globus toolkit is an implementation of the architecture that has been used in a variety of data-intensive and computationally intensive applications.

- Finally, web services are heavily used in cloud computing. For example Amazon’s AWS is based entirely on web service standards coupled with the REST philosophy of service construction.

9.6 Coordination of web services

- The SOAP infrastructure supports single request-response interactions between clients and web services.

- The W3C uses the term *choreography* to refer to a language based on WSDL for defining coordination. For example, the language might specify constraints on the order and the conditions in which messages are exchanged by participants. A choreography is intended to provide a global description of a set of interactions, showing the behaviour of each member of a set of participants, with a view to enhancing interoperability.

- Choreography is intended to support interactions between web services which are generally managed by different companies and organizations. A collaboration involving multiple web services and clients should be described in terms of the sets of observable interactions between pairs of them.

The W3C working draft document [www.w3.org XV] suggests that a choreography language should include the following features

- hierarchical and recursive composition of choreographies;
- the ability to add new instances of an existing service and new services;
- concurrent paths, alternative paths and the ability to repeat a section of a choreography;
- variable timeouts – for example, different periods for holding reservations;
- exceptions, for example, to deal with messages arriving out of sequence and user actions such as cancellations;
- asynchronous interactions (callbacks);
- reference passing, for example, to allow a car hire company to consult a bank for a credit check on behalf of a user;
- marking of the boundaries of the separate transactions that take place, for example, to allow for recovery;
- the ability to include human-readable documentation

Applications of web services

- **Service-oriented architecture-** *Service-oriented architecture* (SOA) is a set of design principles whereby distributed systems are developed using sets of loosely coupled services that can be dynamically discovered and then communicate with each other or are coordinated through choreography to provide enhanced services.
- **The Grid-** The name ‘Grid’ is used to refer to middleware that is designed to enable the sharing of resources such as files, computers, software, data and sensors on a very large scale.
- **Cloud computing** - Cloud computing promotes a view of everything as a service, from physical or virtual infrastructure through to software, often paid for on a per-usage basis rather than purchased