Lesson 6 – Directory services (Part I)

Service Oriented Architectures

Module 1 – Basic technologies

Unit 4 – UDDI

Ernesto Damiani

Università di Milano

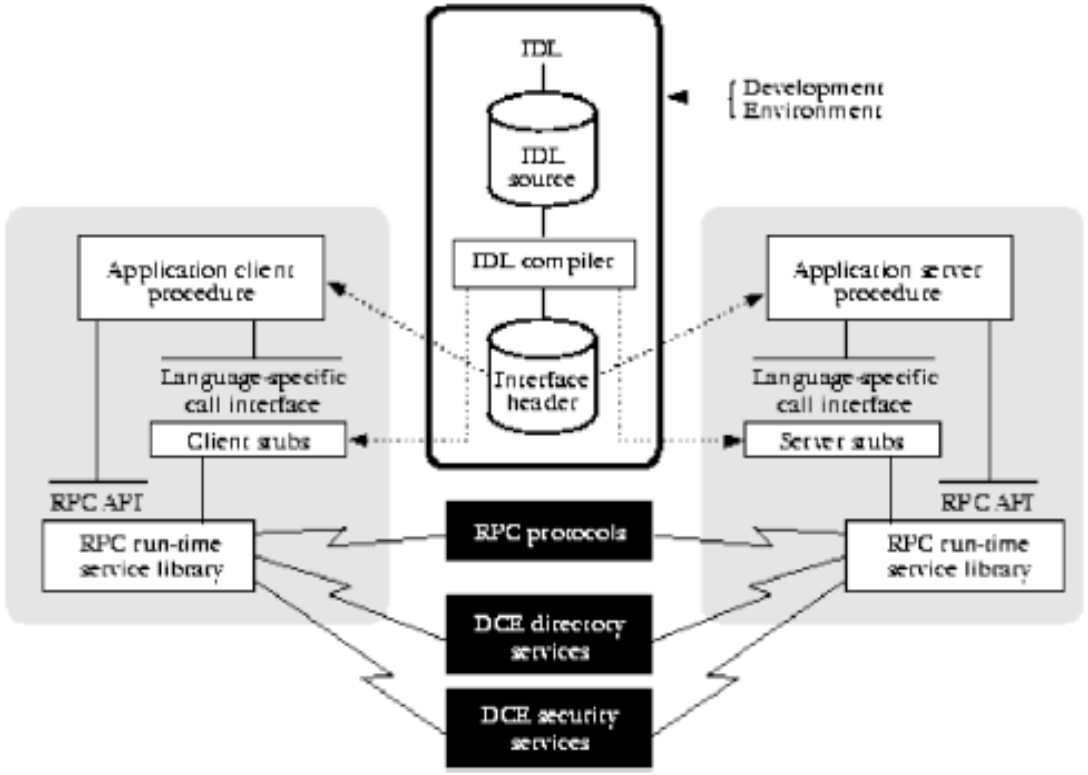
RPC binding (1)

- A service is provided by a server located at a particular IP address and listening to a given port
- Binding is the process of mapping a service name to an address and port that can be used for communication purposes
  - Binding can be done:
    - locally: the client must know the name (address) of the host of the server
    - distributed: there is a separate service (service location, name and directory services, etc.) in charge of mapping names and addresses. These service must be reachable to all participants

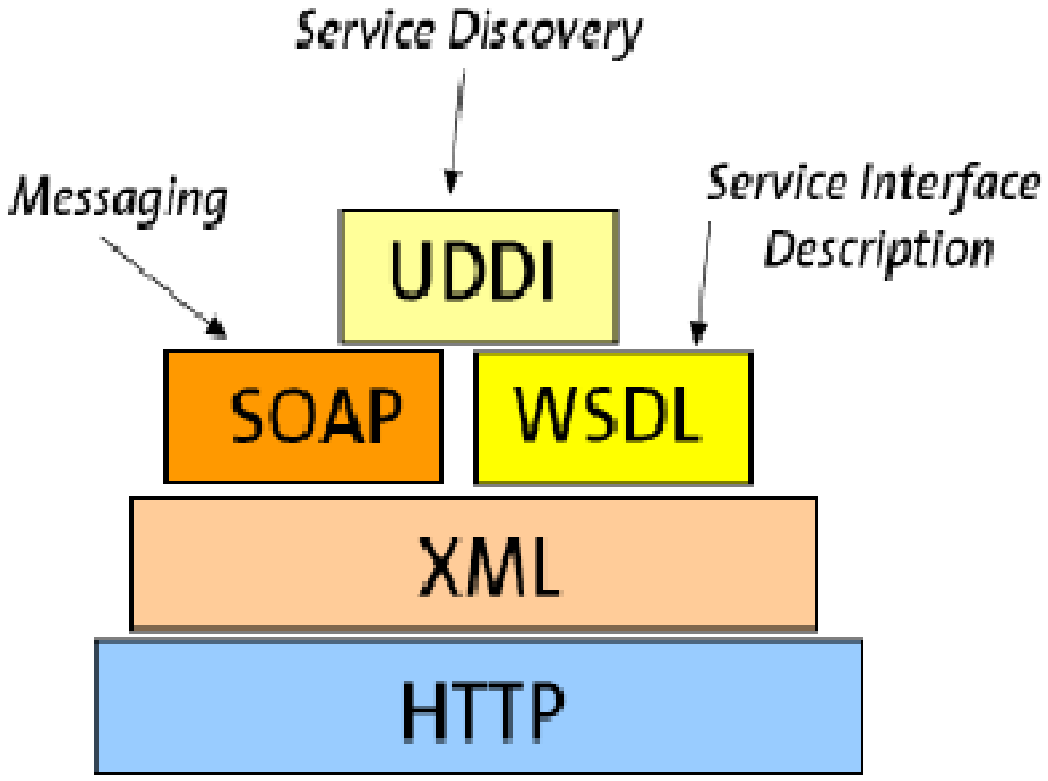
RPC binding (2)

- With a distributed binder, several general operations are possible:
  - REGISTER (exporting an interface): a server can register service names and the corresponding port
  - WITHDRAW: a server can withdraw a service
  - LookUP (importing an interface): a client can ask the binder for the address and port of a given service
- There must also be a way to locate the binder (predefined location, environment variables, broadcasting to all nodes looking for the binder)

RPC binding (3)



Positioning UDDI (1)



Positioning UDDI (2)

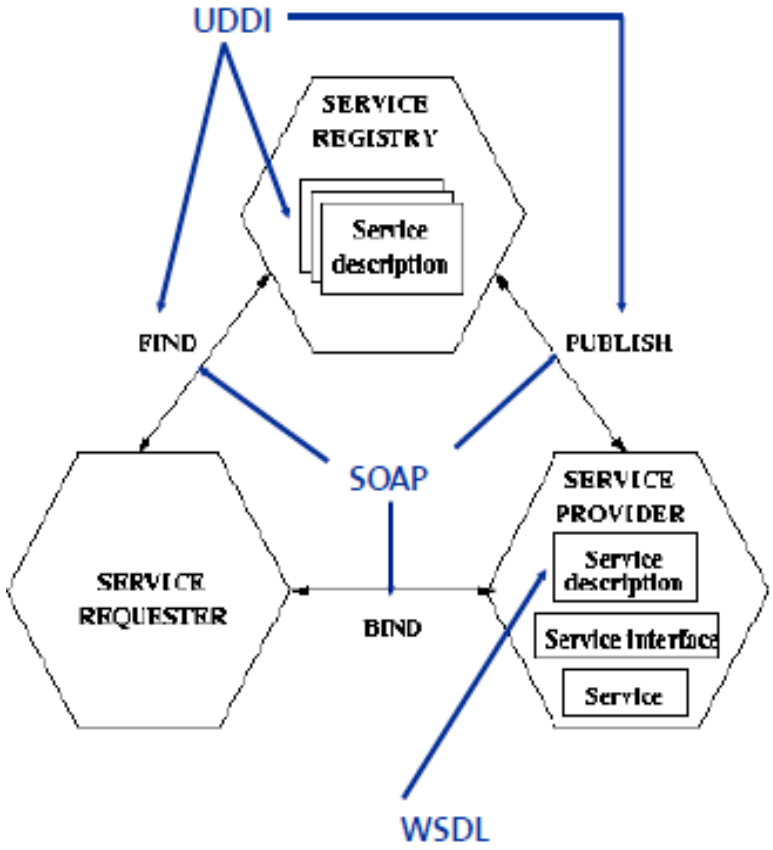
Transport	HTTP, IIOP, SMTP, JMS		
Messaging	XML, **SOAP**		WS-Addressing
Description	XML Schema, **WSDL**		WS-Policy, SSDL
Discovery	**UDDI**		WS-MetadataExchange
Choreography	WSCL	WSCI	WS-Coordination
Business Processes	**WS-BPEL**	BPML	WSCDL
Stateful Resources	**WS-Resource Framework**		
Transactions	WS-CAF	WS-Atomic Transactions WS-Business Activities	
Reliable Messaging	**WS-Reliability**		**WS-ReliableMessaging**
Security	**WS-Security** **SAML, XACML**		WS-Trust, WS-Privacy **WS-SecureConversation**
Event Notification	WS-Notification		WS-Eventing
Management	**WSDM**		**WS-Management**
Data Access	OGSA-DAI		SDO

The role of UDDI (1)

• Once it is possible to interact with any service provider using the standard SOAP protocol, it is still necessary to:

  – describe the services (WSDL)
  – discover the services (UDDI)

The role of UDDI (2)



What is UDDI? (1)

• The UDDI specification is probably the one that has evolved the most from all specifications we have seen so far. The latest version is Version 3 (OASIS Standard Feb 2005):

- Version 1 defined the basis for a business service registry
- Version 2 adapted the working of the registry to SOAP and WSDL
- Version 3 redefines the role and purpose of UDDI registries, emphasizes the role of private implementations, and deals with the problem of interaction across private and public UDDI registries

What is UDDI? (2)

• Originally, UDDI was conceived as an “Universal Business Registry” similar to search engines (e.g., Google) which will be used as the main mechanism to find electronic services provided by companies worldwide. This triggered a significant amount of activity around very advanced and complex scenarios (Semantic Web, dynamic binding to partners, runtime/automatic partner selection, etc.)

What is UDDI? (3)

- Nowadays, UDDI is far more pragmatic and recognizes the realities of B2B interactions: it presents itself as the “infrastructure for Web services”, meaning the same role as a name and directory service (i.e., binder in RPC) but applied to Web services and mostly used in constrained environments (internally within a company or among a predefined set of business partners)

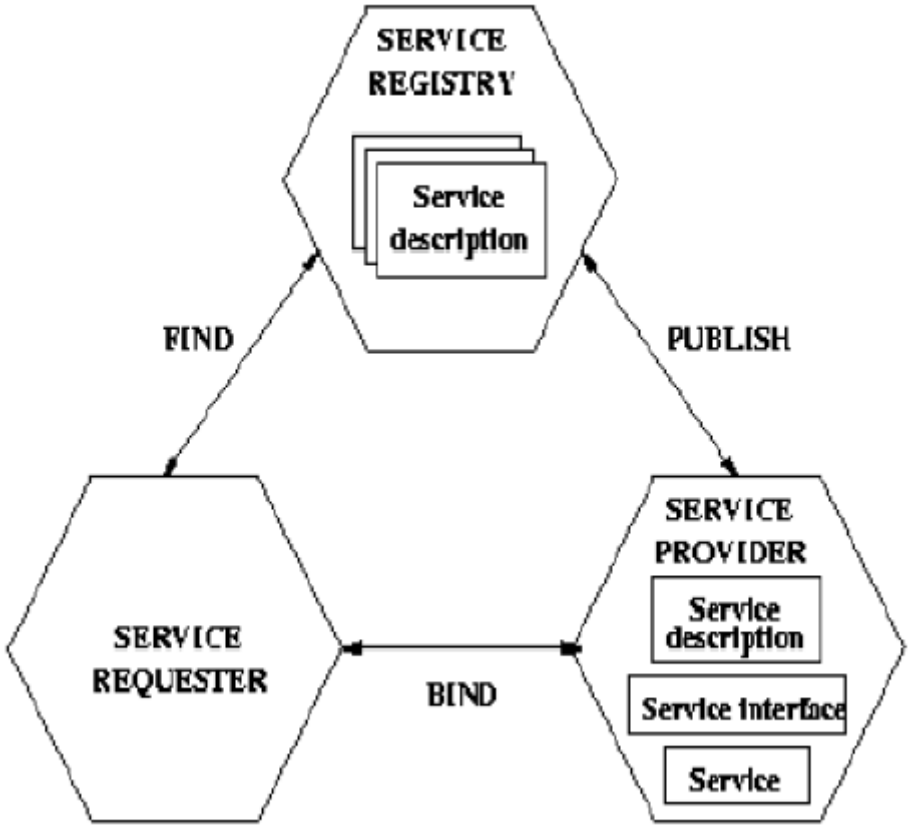
Role of UDDI (1)

- Services offered through the Internet to other companies require much more information than a typical middleware service
- In many middleware and EAI efforts, the same people develop the service and the application using the service

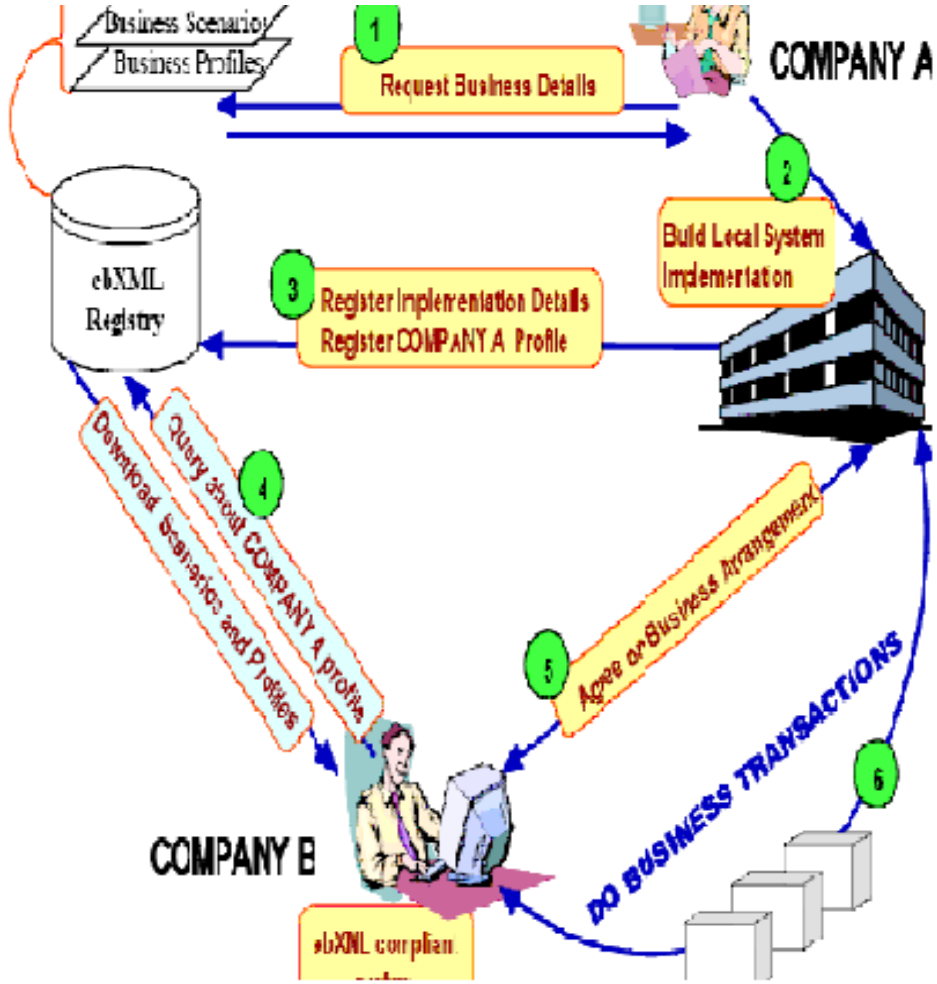
Role of UDDI (2)

- This is obviously no longer the case and, therefore, using a service requires much more information than is typically available for internal company services
  - This documentation has three aspects to it:
    - basic information
    - categorization
    - technical data

Role of UDDI (3)



ebXML architecture



UDDI data (1)

• An entry in an UDDI registry is an XML document composed of different elements (labeled as such in XML), the most important ones being:

- businessService: a list of all the Web services offered by the business entity
- businessEntity: a description of the organization that provides the service
- bindingTemplate: the technical aspects of the service being offered
- tModel ("technical model"): a generic element that can be used to store additional information about the service, typically additional technical information on how to use the service, conditions for use, guarantees, etc.

UDDI data (2)

- Together, these elements are used to provide:
  - white pages information: data about the service provider (name, address, contact person, etc.)
  - yellow pages information: what type of services are offered and a list of the different services offered
  - green pages information: technical information on how to use each one of the services offered

businessEntity (1)

- The generic white and yellow pages information about a service provider is stored in the businessEntity, which contains the following data:
  - discoveryURLs: a list of URLs that point to alternate, file based service discovery mechanisms
  - name (textual information)
  - business description (textual information)
  - contacts (textual information)
  - businessServices: a list of services provided by the businessEntity
  - identifierBag: a list of external identifiers
  - categoryBag: a list of business categories (e.g., industry, product category, geographic region)
    - The each businessEntity has a businessKey

businessEntity (2)

• The businessEntity does not need to be the company. It is meant to represent any entity that provides services: it can be a department, a group of people, a server, a set of servers, etc.

businessService (1)

- The services provided by a businessEntity are described in business terms using businessService elements
- A businessEntity can have several businessServices but a businessService belongs to one businessEntity
- The businessService can actually be provided by a different businessEntity than the one where the element is found. This is called projection and allows to include services provided by other organizations as part of the own services

businessService (2)

- The businessService contains:
  - a serviceKey that uniquely identifies the service and the businessEntity(not necessarily the same as where the businessService is found)
  - name (as before)
  - description (as before)
  - categoryBag (as before)
  - bindingTemplates: a list to all the bindingTemplates for the service with the technical information on how to access and use the service