Introducing Web Services

- [ACKM04] Ch 5, 6

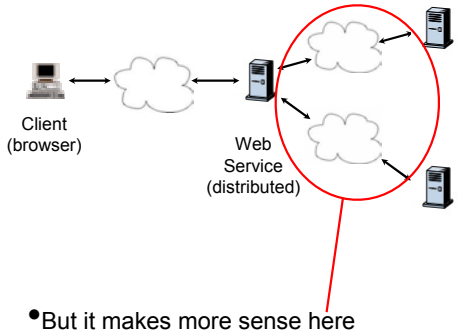
Definition: Web Services

- A standardized way of integrating Web-based applications, using
- XML to tag data
- SOAP to transport data
- Simple Object Access Protocol
- WSDL to describe available services
- Web Services Description Language
- UDDI to list available services
- Universal Description, Discovery and Integration

Not Web Services

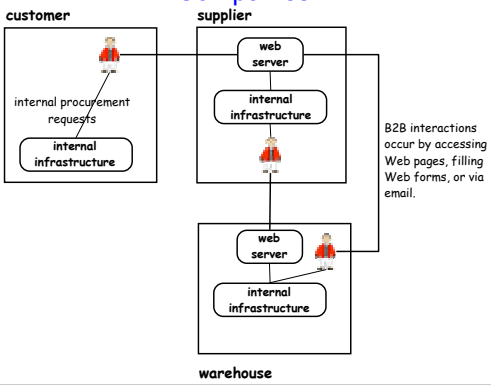
- Generic client (browser)
- RPC or messaging protocol between client and web server is not the focus here

Web Services

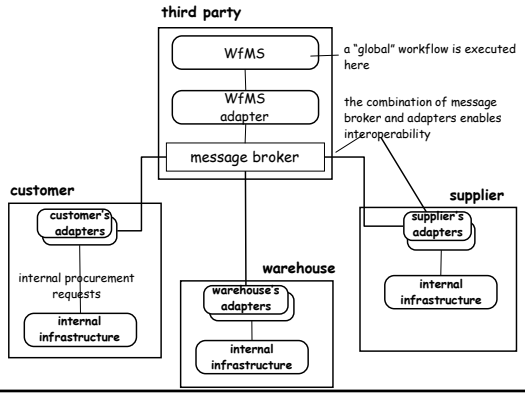


- But it makes more sense here

Manual Integration Between Companies



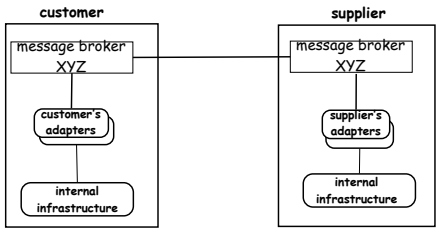
In Principle - Middleware a Service



Why not?

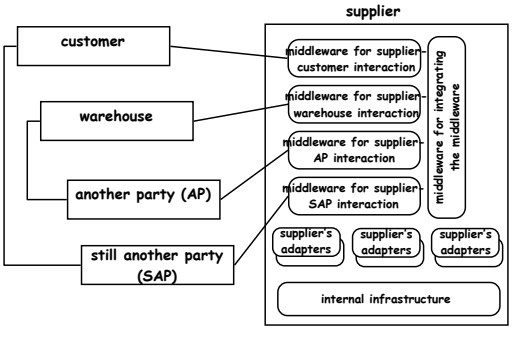
- Autonomy?
- Trust?
- Confidentiality?

How About Point-to-Point?



- Note middleware must interoperate

Middleware Explosion ...



Goals for B2B Integration

- Service-Oriented Architecture
- Redesign of middleware protocols
- Standardization

Service-Oriented

- Functionality always exposed as services
- Loosely coupled
- Invoked by programs as well as users

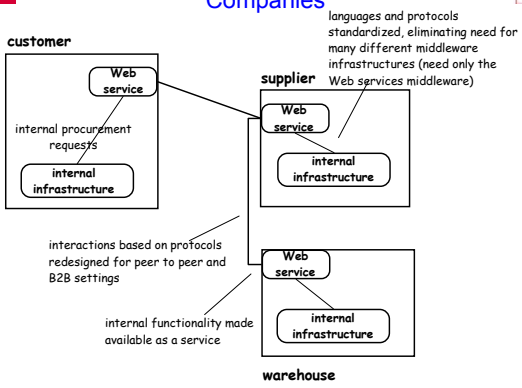
Middleware Redesign

- Peer-to-Peer
- Between trust domains
- Compatibility with Internet

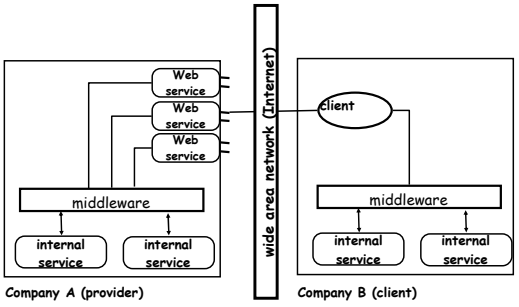
Standardization

- In praise of motherhood ...
- BUT - Web standards seem to have achieved much more traction than previous standards

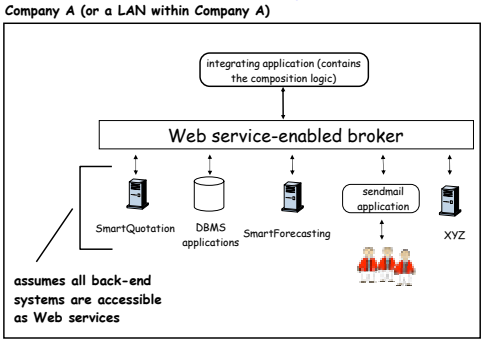
Web Services Integrating Between Companies



A Sophisticated Wrapper - Expose Apps on Web



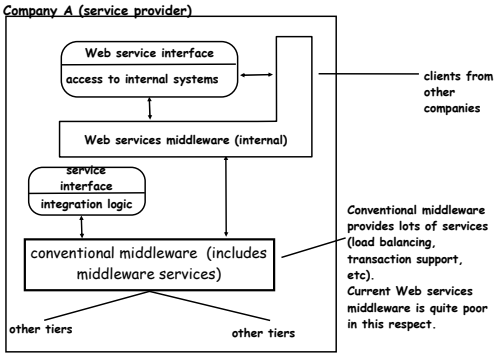
Can Use WS Protocols Internally



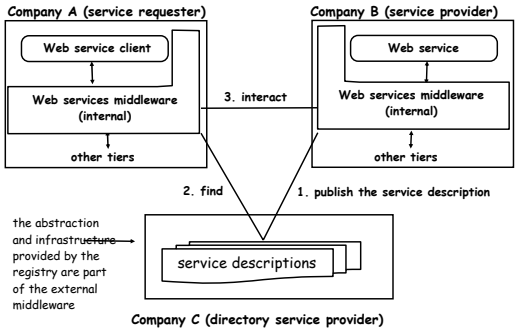
Two Facets of Web Services Architecture

- Internal
- run conventional apps
- expose them as web services
- External
- global services (like DNS)

Internal Architecture



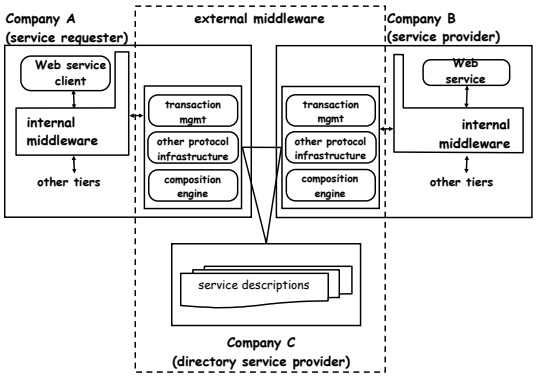
External Architecture



External Architecture

- Is service discovery the only component of Web Services middleware?
- No but external middleware needs to run in peer-to-peer fashion with minimal trust requirements ...

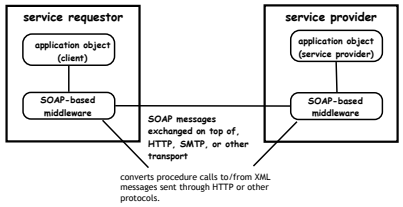
External Architecture With Middleware



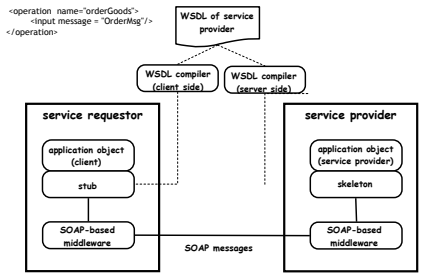
Basic Web Services

- Minimalist definition/requirements:
- communicate (SOAP)
- describe services - IDL (WSDL)
- directory service (UDDI)

Minimalist Infrastructure ...

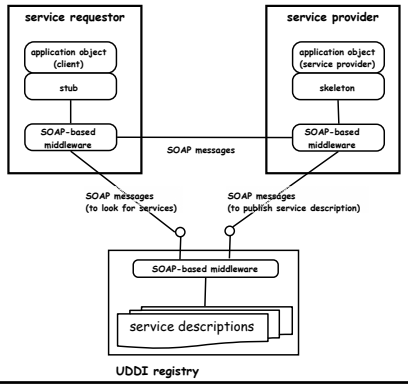


Using WSDL Specification



Note all WSDL "processing" happens at development time.

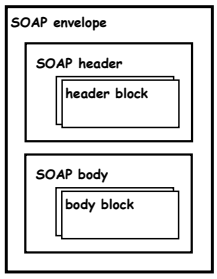
Using UDDI Registry



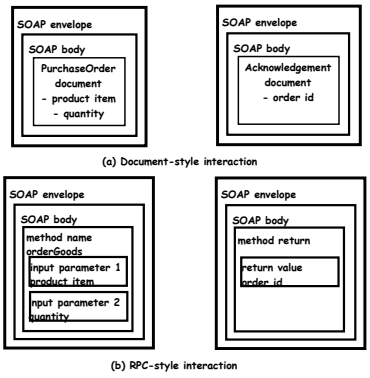
Simple Object Access Protocol

- Specifies:
- message format for one-way comms
- specification for SOAP RPC
- rules for processing SOAP messages
- rules for transport - HTTP and SMTP

A SOAP Message



Document vs RPC



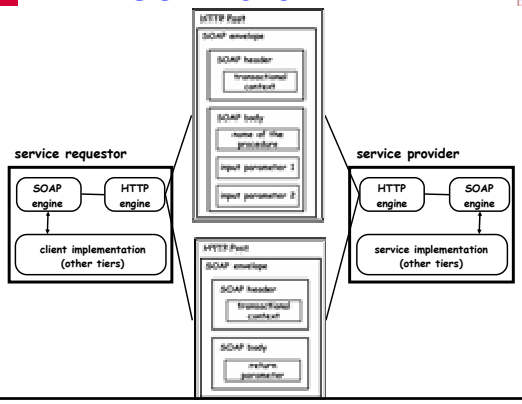
Intermediate Processing - Roles

- Header blocks only ...
- none: no node processes this block
- next: every node processes the block
- ultimateReceiver: only last node in path

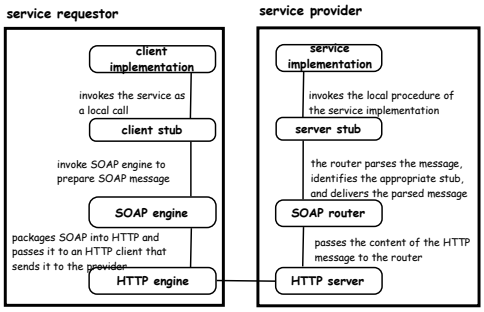
Intermediate Processing - Roles

- Note “next” role in header

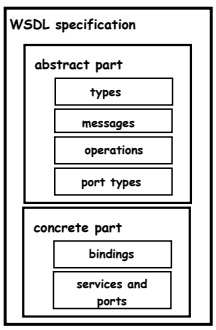
SOAP over HTTP



Simple Implementation



WSDL Specifications

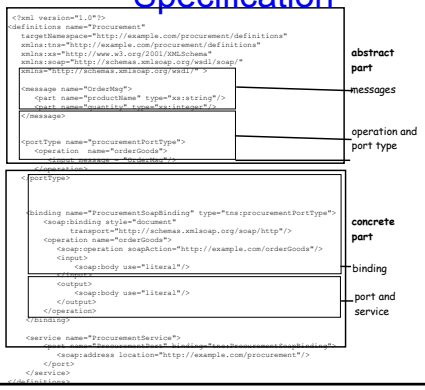


- Abstract part
- signatures, operations
- like traditional IDL
- Concrete part
- bindings / services / ports

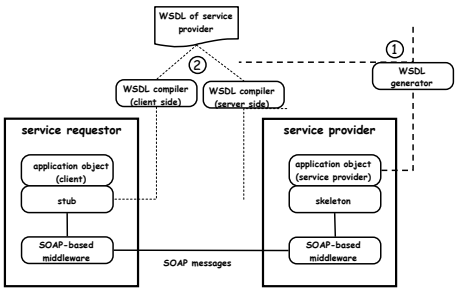
Concrete Part

- Interface Bindings
- message encoding & protocol bindings
- Ports
- interface binding + network address
- Services
- logical groupings of ports

A WSDL Service Specification



Automatic Generation of WSDL



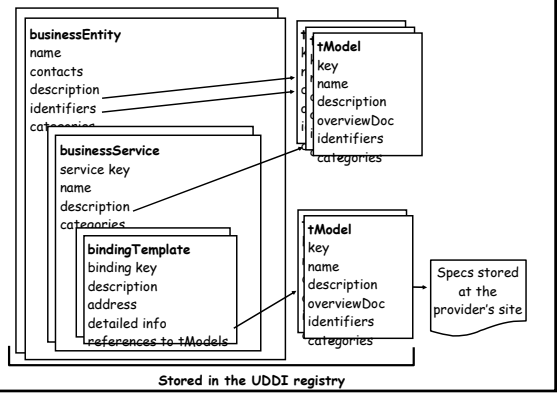
Universal Description Discovery and Integration

- The Web Services Directory Service
- Service Registry
- for browsing by developers
- for dynamic binding
- Business registry
- original goal: global registry, every business and exported service registered there
- now: support interaction between private and public UDDI registries

Information in UDDI

- Business entity
- an organization that provides Web services
- Business Service
- group of Web Services that cooperate in performing some business process
- bindingTemplate
- technical information needed to use the service
- tModel
- general container for a specification

Information in UDDI



tModel Description

overviewDoc (refer to WSDL specs and to API specs)

classification information (specifies that this tModel is about XML, WSDL, and SOAP specs)

```
<tModel tModelKey="uddi:uddi.org:v3_publication">
  <name>uddi-org:publication_v3</name>
  <description>UDDI Publication API V3.0</description>
  <overviewDoc>
      <overviewURL useType="wsdlInterface">
http://uddi.org/wsdl/uddi_api_v3_binding.wsdl#UDDI_Publication_SoapBinding
      </overviewURL>
  </overviewDoc>
  <overviewDoc>
      <overviewURL useType="text">
          http://uddi.org/pubs/uddi_v3.htm#PubV3
      </overviewURL>
  </overviewDoc>

  <categoryBag>
      <keyedReference keyName="uddi-org:types:wsdl"
          keyValue="wsdlSpec"
          tModelKey="uddi:uddi.org:categorization:types"/>
      <keyedReference keyName="uddi-org:types:soap"
          keyValue="soapSpec"
          tModelKey="uddi:uddi.org:categorization:types"/>
      <keyedReference keyName="uddi-org:types:xml"
          keyValue="xmlSpec"
          tModelKey="uddi:uddi.org:categorization:types"/>
      <keyedReference keyName="uddi-org:types:specification"
          keyValue="specification"
          tModelKey="uddi:uddi.org:categorization:types"/>
  </categoryBag>
</tModel>
```

- Note overviewDoc has URL - refers to specification stored outside UDDI server
- Multiple overviewDocs are allowed

Sharing of tModels

- Each tModel is assigned a GUID when it is registered
- The tModels may be shared by multiple bindingTemplates
- Software can use UDDI to search for all instances of a given GUID

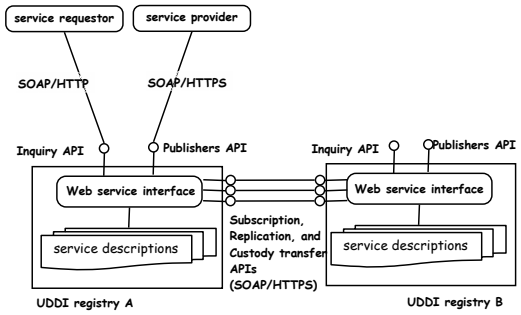
UDDI Registry API

- Inquiry
- search
- Publisher
- register and unregister operations

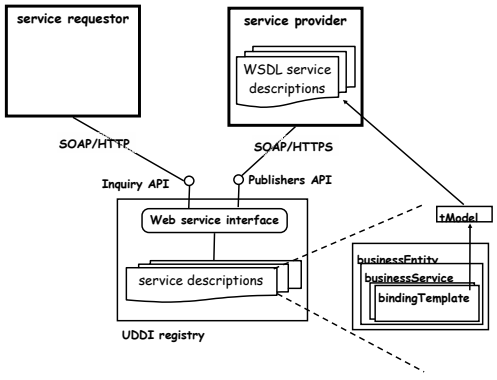
UDDI Registry API - II

- Security
- Custody and Ownership Transfer
- create/destroy/move objects
- Subscription
- monitor for changes in UDDI data
- Replication
- readonly except at “owner” site

Architecture of Distributed UDDI



WSDL in UDDI



Private UDDI Registries

- Public Universal Business Registry not required for Web apps that are intra-enterprise or between trusted partners
- Three categories
- public
- private
- shared
- Must support replication between registries where appropriate

“Advances” in SOAP

- BLOBs
- now Web Services description yields interaction sequence but not data formats, which are agreed to by communicating applications
- Document exchange
- constructing WSDL requires semantic understanding of the documents

Dynamic Binding

- Fully general
- e.g. CORBA DII
- now we have run-time types, introspection
- Restricted
- e.g. all services using a given tModel
- but enterprises usually won’t accept a program effectively signing a contract ...

Grid Computing

- Sharing supercomputer cycles and huge volumes of data in scientific computing
- Open Grid Services Architecture effort
- IBM, Microsoft, Sun, Oracle, ...
- Extensions to Web Services standard