

Lesson 17 – Advanced Web Services

Service Oriented Architectures

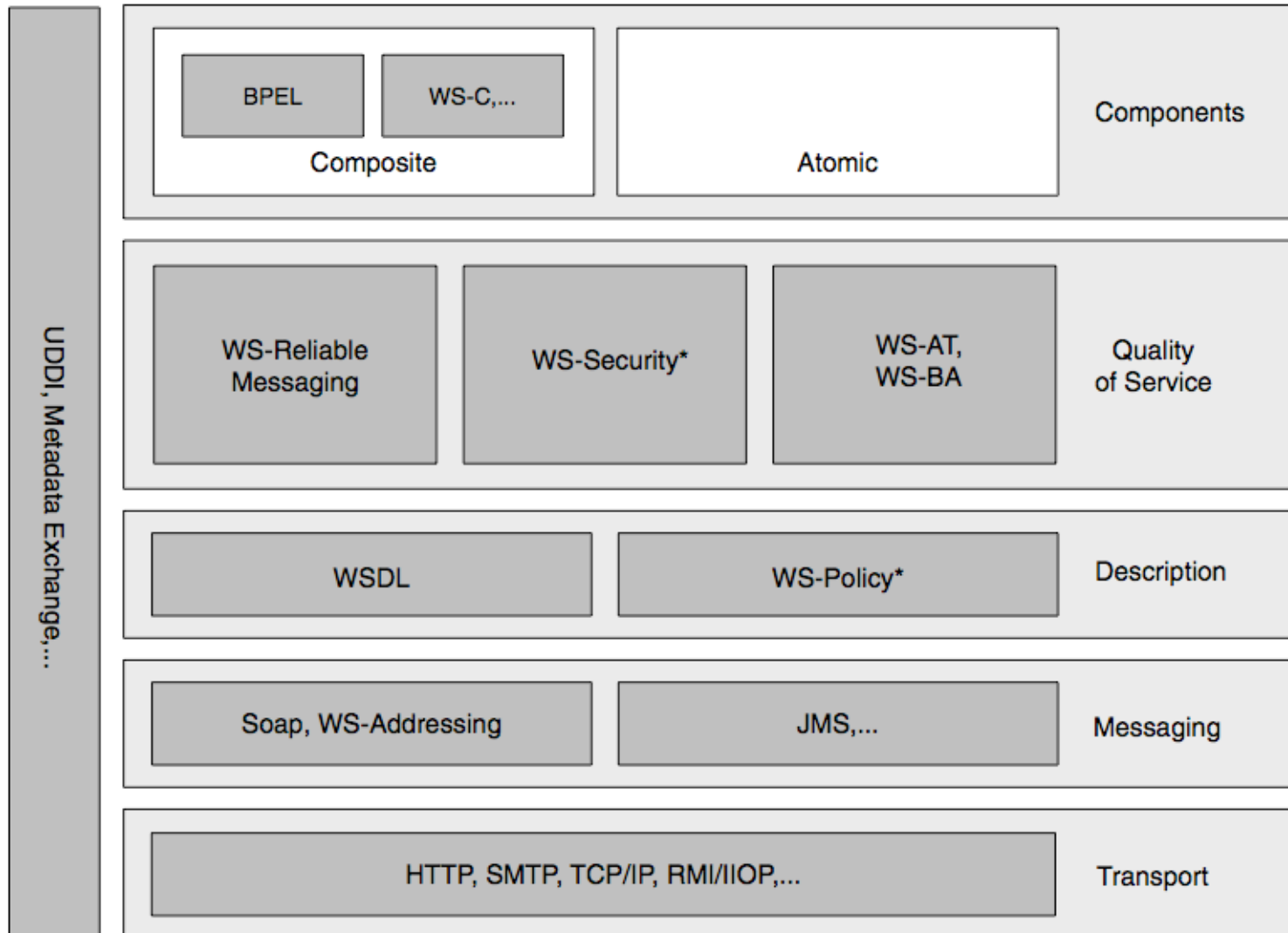
Module 3 - Resource-oriented services

Unit 3 – Advances

Ernesto Damiani

Università di Milano

WS Architecture

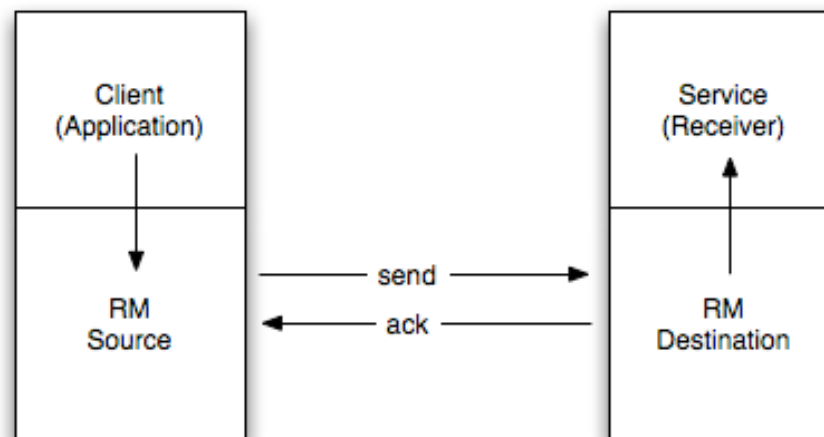


Advanced Topics for WS

- Quality of Service Layer
 - Transactions
 - Security
- Component Layer
 - WS-BPEL (and BPMN) WS-CDL

Other Aspects

- WS-Addressing and WS-Resources
 - standards to express end-points and state (references instead of values)
 - REST concepts
- Reliable Messaging - Add-on for Services/Clients
 - uses layer with message sequencing (acks and resends...)
 - best effort, at least once, at most once, exactly once

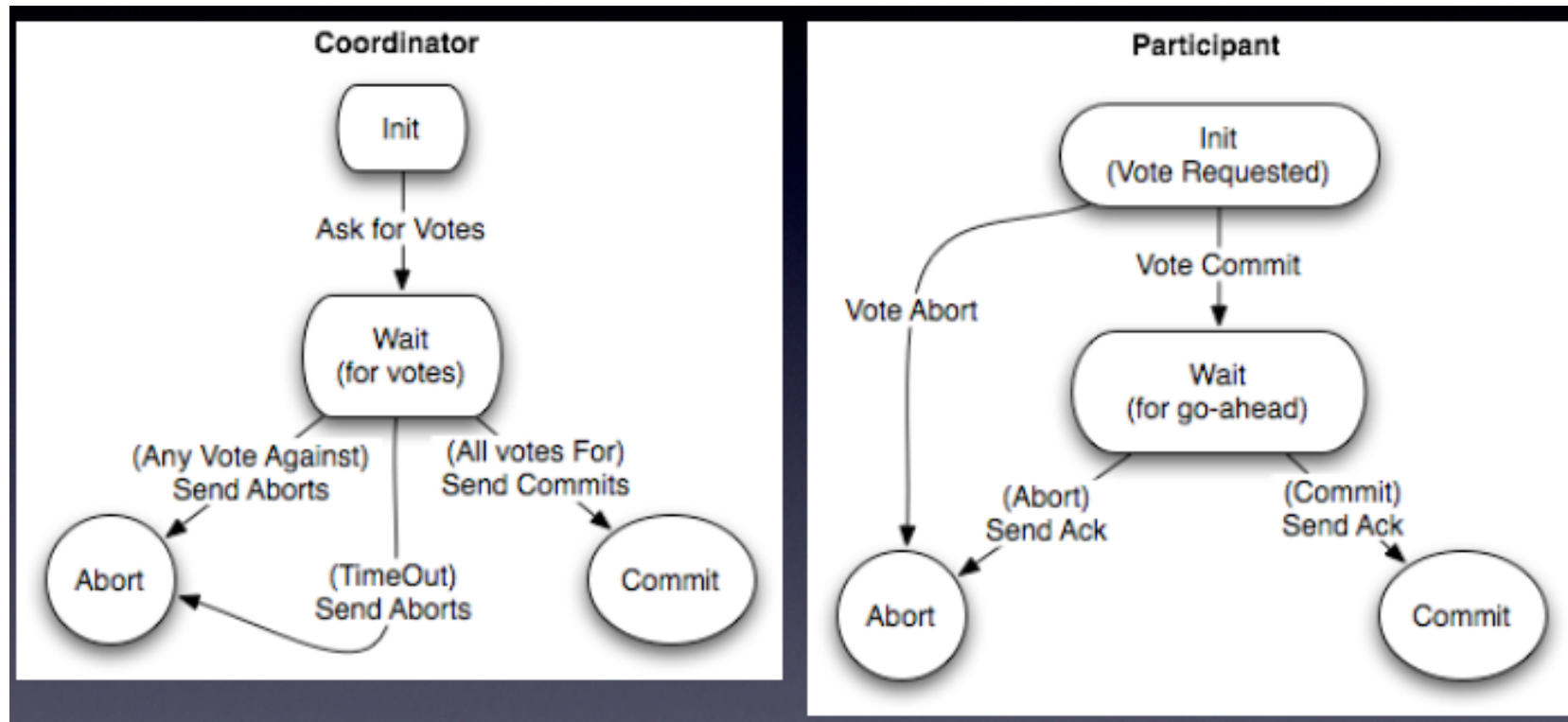


WS Transactions

Traditional transactions

- The problem
 - e.g. in programming: $x = x+1$ and $x = x+y$
in sequence/in parallel
- Databases, Distributed Networking
- ACID
 - Atomic
 - Consistent
 - Isolated
 - Durable

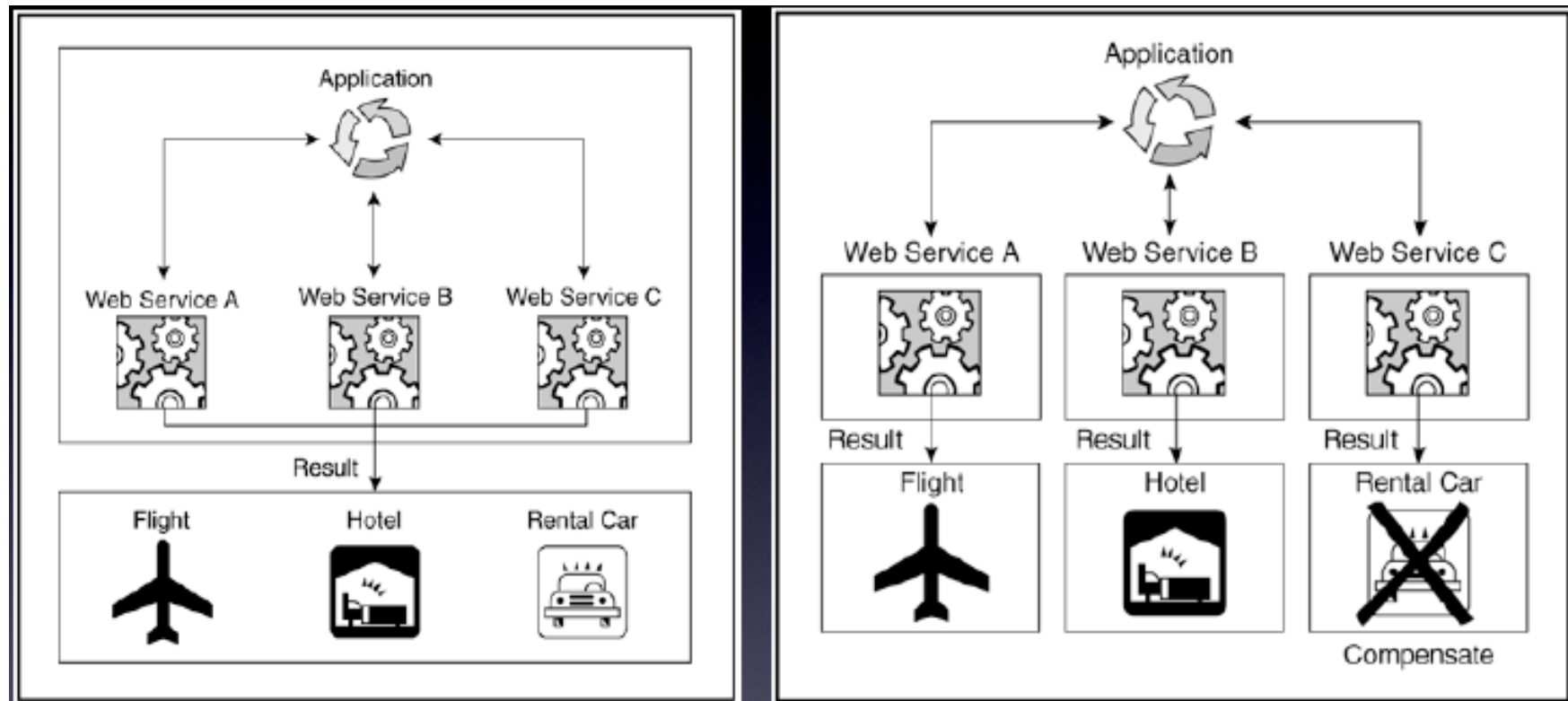
Phase Commit Protocol



Extended Transactions

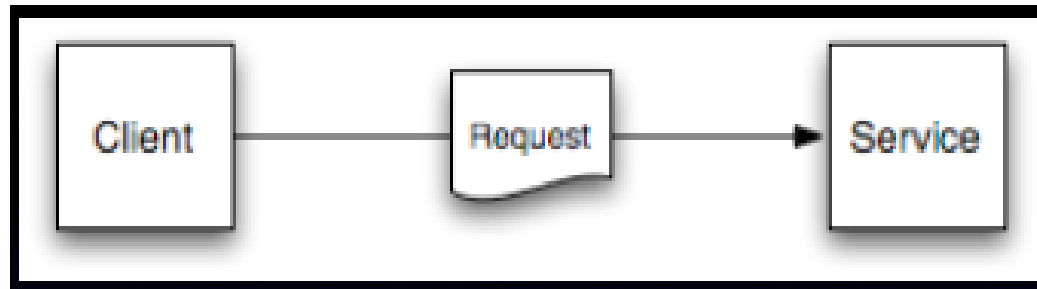
- Need for Extended Transactions in Web Services
 - rationale for Non-ACID requirements
 - long duration, alternate failure handling, selected outcome inclusion, non-blocking across enterprises
 - (travel booking example)
- Web Services Protocols and Framework Standards
 - WS-Coordination
 - WS-Atomic Transaction
 - WS-Business Activity

Classic and Business Transactions



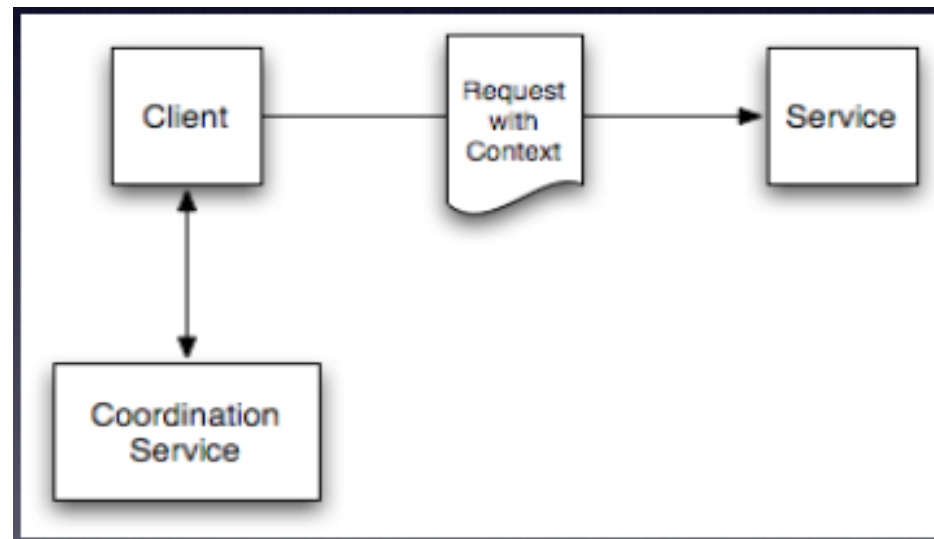
Framework for Transactions

- Need standard way to extend existing services
- Make use of additional services!



WS-Coordination

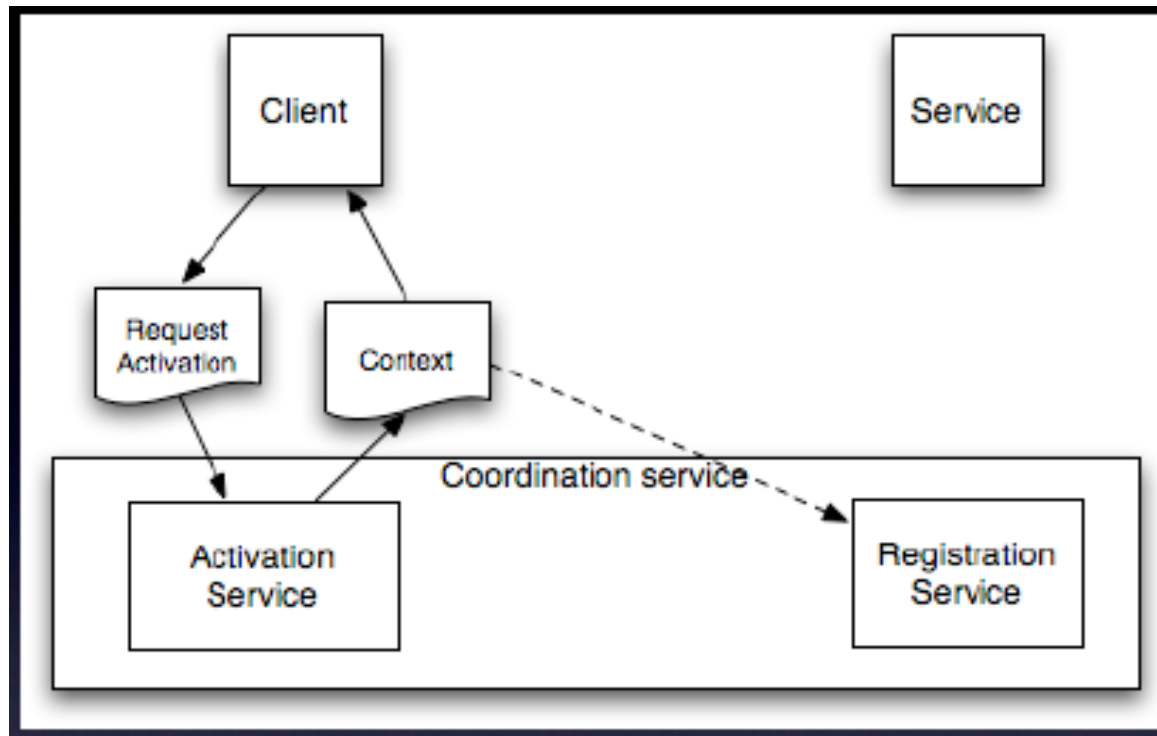
- 'Coordination' rather than 'Transaction' because the framework can be used for more than just transactions
- Pluggable coordination protocols - separate from framework



Coordination Activation 1 (Part I)

- Create (coordinated) activity, supplying
 - choice of coordinator
 - options and info for choices of coordination (completion) protocol
- Context created, recording
 - Unique Identifier for the activity
 - address of registration service with completion protocol options

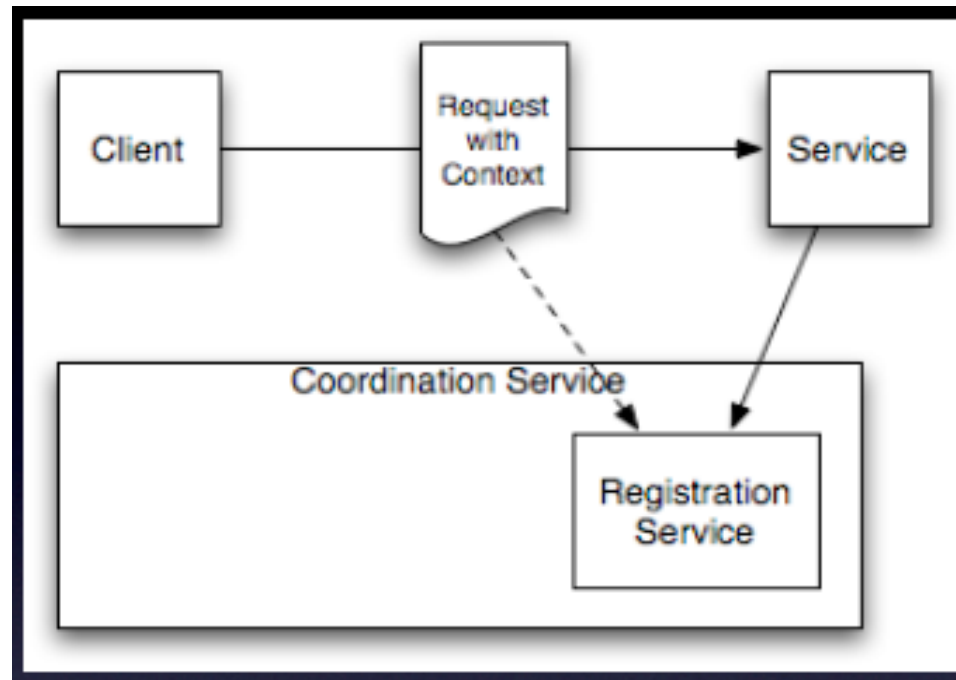
Coordination Activation 1 (Part II)



Coordination Activation 2 (Part I)

- Context is sent with web service request
- Service registers (to be involved in completion protocol) and can choose protocol to be used
- Depending on protocol, client or service can identify who is coordinator (endpoint responsible for running completion protocol)
- At completion, messages sent between coordinator and registered participants

Coordination Activation 2 (Part II)



Completion Protocols

- Examples
 - WS-Atomic Transaction
 - WS-Business Activity
- Exercises:
 1. Go through sequence of steps to understand the examples and use of coordination services and protocols
 2. Consider other examples (e.g. car parts case study)

Security and WS

- Fundamental to development of real web services (controlling access, simple authentication for range of distributed services, confidentially passing on security information, non-repudiation,..)
- Need for WS specific security support
 - Problems with SSL/TLS and IPSec alone
 - Generic protocols to leverage existing security concepts (Certificates, Kerberos tickets, CAs,...)
 - Example illustrating hiding, revealing, authenticating with multiple parties and federated security (policies)

WS-Security (Part I)

- Claims -> Security Tokens
 - (e.g. Kerberos Ticket, X509 certificate)
- Presenting Tokens in messages and service requests. Leveraging standards for XML:
 - Encryption and Signing using XML Encryption and XML Signature standards
 - SAML - security assertion markup language
 - Ways to include in SOAP (keys and encrypt info in headers, possibly partly encrypted/signed body) for intermediary access

WS-Security (Part II)

- Generating and Distributing Tokens (Security Token Supplier as a Service)
- Describing Policies (e.g. to put in WSDL)
- Verification activities (policy checking/managing)

Advanced Security (in Development) (Part I)

- *WS-Trust*
 - specifies models for Security Token Supplier Service
- *WS-SecurityPolicy*
 - how to express security policy for WS-Policy framework
- *WS-SecureConversation*
 - avoiding PKI for all messages
 - analogous to SSL/TLS as a layer between HTTP and TCP
 - exchange session keys (using PKI only at end points to set up)

Advanced Security (in Development) (Part II)

- **WS-Privacy**
 - expressing policies to be processed by clients and services as part of WS-Policy framework
 - constraining how data may be used
- **WS-Federation**
 - multiple enterprises sharing trust of authenticated identity
 - allows single sign-on
- **WS-Authorization**
 - add on to WS-Trust - standards for use of authorization tokens

Composing Web Services

- Problem:
 - how should we describe the combining (composing) of web services to create new services?
- Requirements:
 - descriptions should be relatively easy (test of SOA as good architectural approach) and completely platform independent
 - need a notation/language which is formal enough (executable!)
 - need to take account of both abstract and concrete services
 - need to orchestrate or choreograph the interactions (process description and collaborations)
 - should be linked to theory of processes e.g. Pi Calculus to understand completeness/properties/analysis

Look at Three Languages

- Business Process Execution Language WS-BPEL
(Formerly BPEL4WS)
- Business Process Modelling Notation BPMN
- Choreography Description Language WS-CDL

Orchestration (1)

- Implementing a simple service (e.g. Java to match existing WSDL with SOAP conversion wrapper for IO)
- In practice, the other way round! (Take some Java, use auto tools to expose it as a service, generating wrapper s/w and also WSDL file)
- What about a more complex service using other services...? (DIAGRAM)

Orchestration (2)

- A description language could be used to describe how the service works (is executed) however it is to be implemented. WSDL only says how to interact with a service. Using the new language we should be able to write an implementation of a new service and also describe how to make use of existing services in a combination. A standardised new language would allow for complete platform independence and possibilities for direct execution (i.e. no need for translation).

BPEL design

- Describing processes
 - flow (sequence, branching, parallel)
 - message exchanges
 - nested/recursive structure (processes within processes)
- Abstract or Executable
 - same notations (abstract can include hidden/unspecified detail)
 - executable means description can be implemented
 - locally (in some other PL), OR
 - with a BPEL engine (i.e. notation is a portable executable language)
- Programming in the Large
 - piecing together existing services, rather than locally implementing atomic services in a conventional PL.

BPEL is closely linked with WSDL

Recall WSDL docs - collection of definitions of

- Abstract Interface
 - types
 - message types
 - portTypes (with op names and message types)
- Deployment Information
 - bindings (to concrete transport protocols..)
 - ports (endpoint addresses for ops)
 - service (collection of ports)
 - Note separation of concerns (interface and deployment)
- BPEL also makes use of WS-Addressing, XPath, XML Schema

BPEL elements (1)

- Partner Links and Partner Link Types (abstraction of connections between 2 processes)
- Variables
 - message types
 - XML Schema types
 - XML elements
- Fault handlers

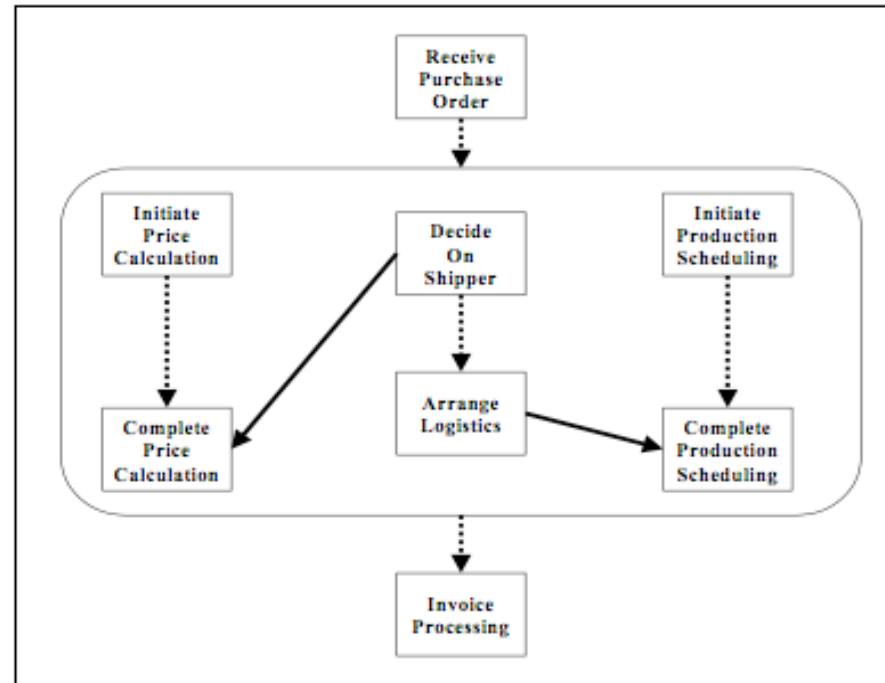
BPEL elements (2)

- Process (activity) description
 - <sequence> <if> <invoke> <receive> <reply> <assign>
<flow> <pick> <wait>
iteration (<while> <for> <repeatUntil> <forEach>)
<scope>
- BPEL only refers to abstract interface parts of WSDL for services
- Deployment of BPEL (e.g. endpoint addresses for services) kept separate
 - not part of the language

Advanced BPEL aspects

- Correlation Sets
- Fault Handling (including compensation handlers)
- Event Handlers
- Abstract and Executable details

An example from the standard (BPEL 1.1)



A picture of some business processing to be described in BPEL.

N.B: BPEL 2.0 is now referred to as WS-BPEL 2.0 (finalized 31/1/2007)

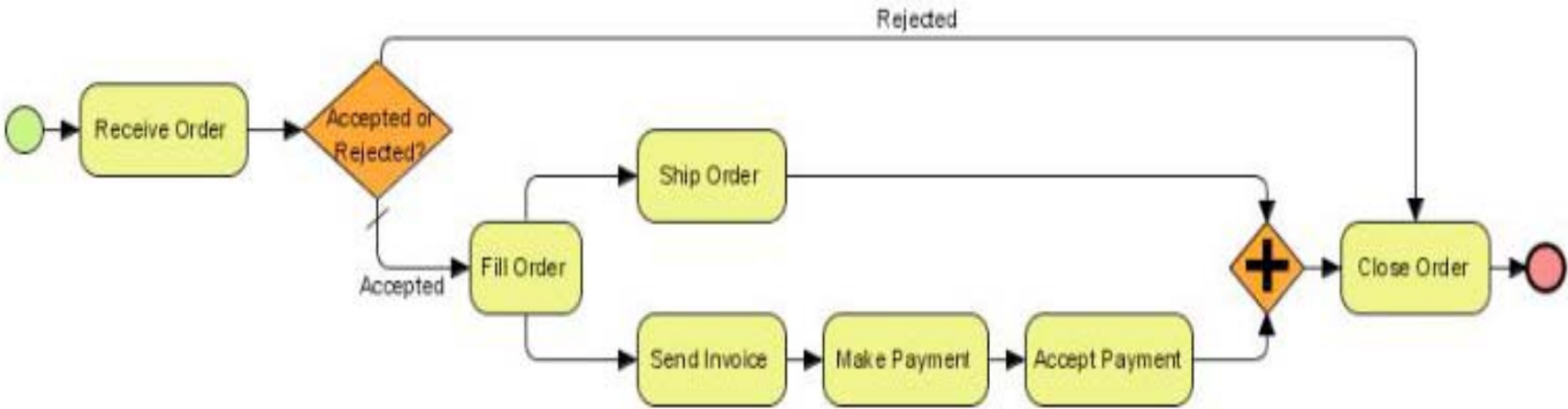
BPEL Examples

- See the Standard (s)
 - WS-BPEL 2.0 (31/1/2007)
 - http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel
- and a tutorial
 - http://www.eclipse.org/tptp/platform/documents/design/choreography_html/tutorials/wsbpel_tut.html

BPMN (Business Process Modelling Notation)

- Draft produced by BPMN-I in 2004 in an attempt to standardise from the multiple process notations being used at the time and designed to link to BPEL 1.1 standard (BPEL4WS at the time)
- Wide industry support
- OMG Standard BPMN 1.0 published Feb. 2006

BPMN example



BPMN Graphical Elements

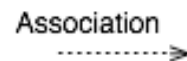
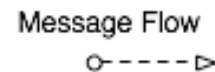
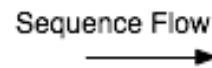
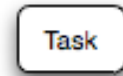
- DEMOs
 - Wikipedia entry for a good introduction
 - <http://en.wikipedia.org/wiki/BPMN>
 - Links to demos and tools (on module page)
 - BP-VA (Flash demo 2006)

BPMN Elements

Events



Activities



Gateways



Orchestration and Choreography

- Orchestration is about describing and executing a single view point model
- Choreography is about describing and guiding a global model
- You can derive the single view point model from the global model by projecting based on participant

Too many languages?

- Still some issues on relationship between BPEL and BPMN
- CDL is NOT a competitor. There are good reasons for needing a global (choreography) view with no centralised control as well as the centralised control glue (orchestration)
- Especially for governance issues

