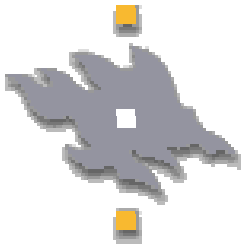


Lecture #3: February 2 2004

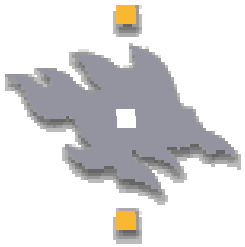
Web Services Overview

Web Services Infrastructures

A Complete View of the Web Services Stack

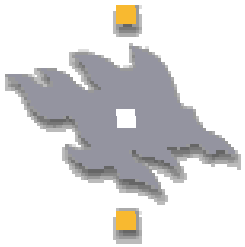
Suresh Chande

Department of Computer Science



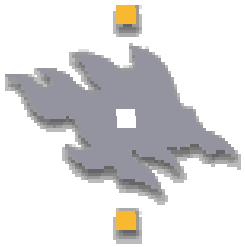
Web Services Architectures

- Standards defined around the *Web Services* technologies provide the required level of convention to maintain the required inter-operability (WS-I) between the Web Services.
- These current standards are not sufficient for the Service oriented enterprise systems to emerge and evolve based on them as they only help in developing inter-operable solutions, but do not provide the necessary infrastructure requirements.
- The Web Services standards are numerous and are also to some extent overlapping due to competing proposals from IT Vendors encouraged by potential IPRs behind the proposals



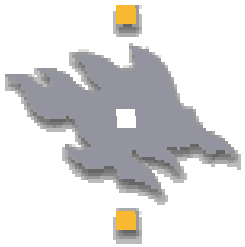
Web Services Architectures

- There is a need for a Web services infrastructure which can gaurantee the basic needs for Web Services oriented eco systems, the basic needs being :
  - Security,
  - Authorization,
  - Debugging capabilities
  - Non-repudiation
  - Real time availability and reliability
  - Service level agreements,
  - Failure recovery,
  - Monitoring and logging
  - Billing & Metering



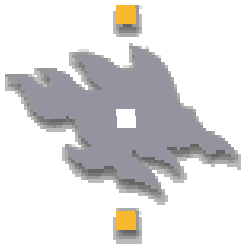
Web Services Infrastructure

- This is a run time environment over which the Service providers, brokers and consumers can inter-operate with trust ensuring that the different players are operating in a predictable and reliable fashion
- Web Services infrastructure could be hosted by corporate networks, network infrastructure providers, service integrators or the service providers.
- The players within the infrastructure comply with a set of rules, standards and build an element of trust within the networked environment



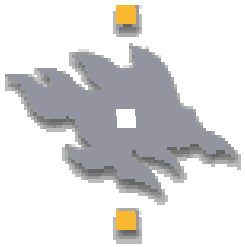
Web Services Infrastructure

- The players could be held account for malicous usage of the network or demand for the rightful services of such an infrastructure
- The basic services required within such an infrastructure are :
  - Security
    - Authentication and Authorization
    - Confidentiality
  - Connectivity
  - Management
  - Billing & Metering
  - Reliability



Security [Contd]

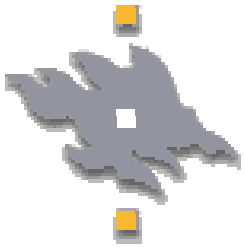
- Authentication
  - The infrastructure must provide a gauranteed identity of originator of the request and response messages
  - The originator is truly who they claim to be
- Authorization
  - Authorization to the rights and usage of a specific services within the infrastructure based policies and access control
- Confidentiality
  - The messages are untampered and are as orginated or as intended to be or as they are expected to be as they are processed over multiple intermediaries and finally at the target receipient.
  - Information exchanged within the network environment is kept confidential and there is other use of the information exchanged between the participants of the Web Services within environment other than for the intended purpose agreed upon



Security

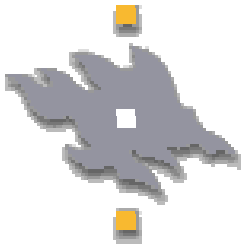
- Signatures
  - Digitally signed certificates are the best possible reliable means of authentication for both senders/receivers and message authentication

Note: Management of the authentication and the digital certificates and their seamless integration within a Web services framework is a key functionality that distinguishes Web services infrastructures



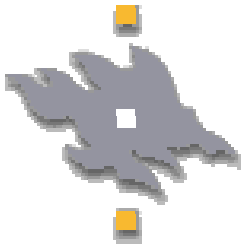
Connectivity & Reliability

- Guaranteed connectivity at all times over the Web Services Infrastructure for business critical application.
- The agreed reliability of the connection between the service participants plays a key role to success or failure.
- The WS infrastructure should be able to provide sufficient class of connectivity levels at all possible loads a business needs to perform efficiently.



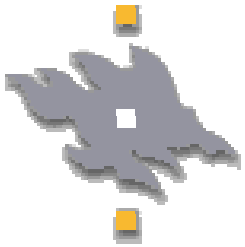
Web Services Management

- Provide simple and easy methods to manage Web Services within the WS Infrastructure:
  - Deploy
  - Monitor
  - Debug and
  - Maintain
- Simplifiy the Web Services by providing added services such as:
  - Logging and Tracing of Web Services utilization
  - Reporting the status of the Web Services
  - Providing default features within the infrastructure such as:
    - Billing
    - Charging



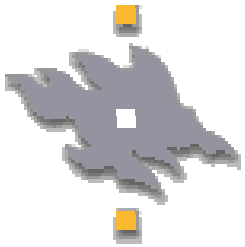
Web Services Reliability

- This infrastructure should ensure that the deployed Web Services are functioning in a reliable manner by ensuring that :
  - Guaranteed Delivery of messages to the intended recipient within the time critical factor.
  - Non-Repudiation, contain sufficient traces of the action so that recipient or the sender can not repudiate over the performed actions within the Web Services Environment.
  - Ensure the availability of functioning services at all times 24X7 as they are based on the Internet technologies, which are meant to be available at all times.
  - Maintain the level of confidentiality within the infrastructure environment.



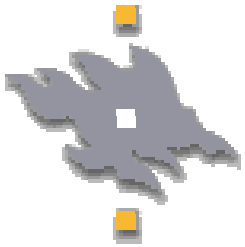
Web Services Architecture - Realization

- New Web Services technologies are being proposed very often as one identifies the lack of features to support in service oriented architectures in bring existing solutions into the Web Services World(Grids for example).
- All these technologies are providing relations to each other at a protocol level and in some cases overlapping with other specifications.
- These technologies require to be mapped to the context of the Web Services infrastructures at architectural level. There are considerable efforts been put into providing these technologies onto existing application platforms/development environment level(J2EE, .Net, etc), but there still lacks a clear Web Services Architecture for internal web services deployment & management



References

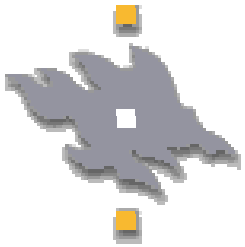
- OMG's Relevant activities related to defining such a middleware architectures have been in collect RFI(request for Information), which basically will drive thier future architectures specifications under a seperate RFP process : http://www.omg.org/cgi-bin/apps/do_doc?bei/03-01-07.pdf



A ~Complete World of Web Services
A Technology Mapping

Suresh Chande

Department of Computer Science

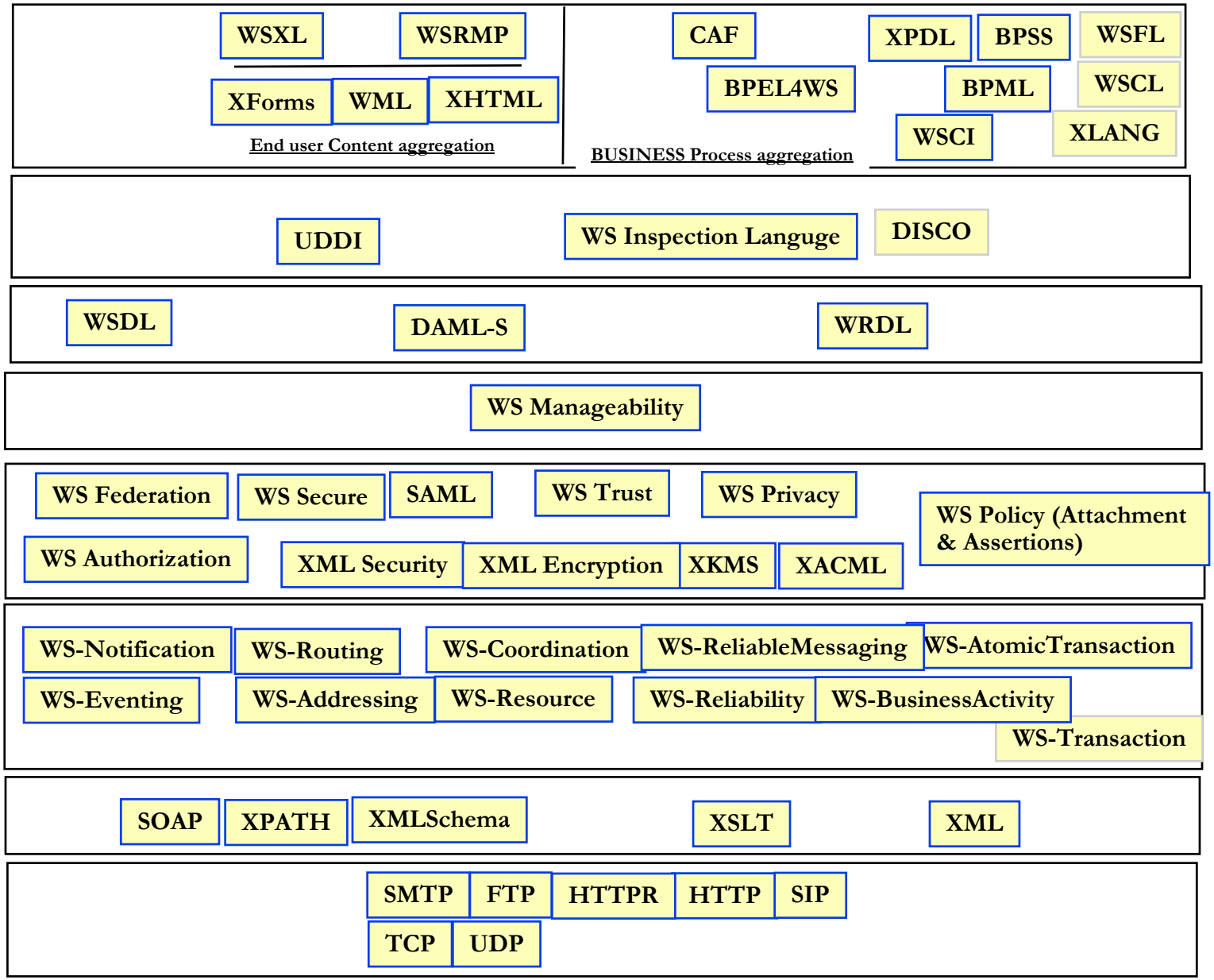


Technology mapping

- Purpose : To understand the complete landscape of Web Services technologies and its development towards enabling services oriented architectures.

- Look at the main purpose of a specific Web Services technology being proposed and to understand the relation of each other and the manner it affects the Web Services architecture

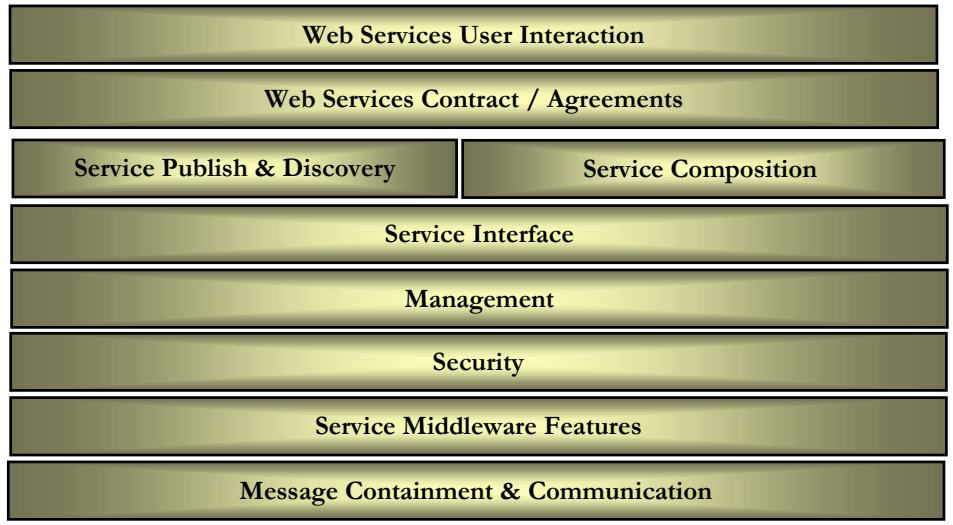
A Complete Web Services Technology Stack

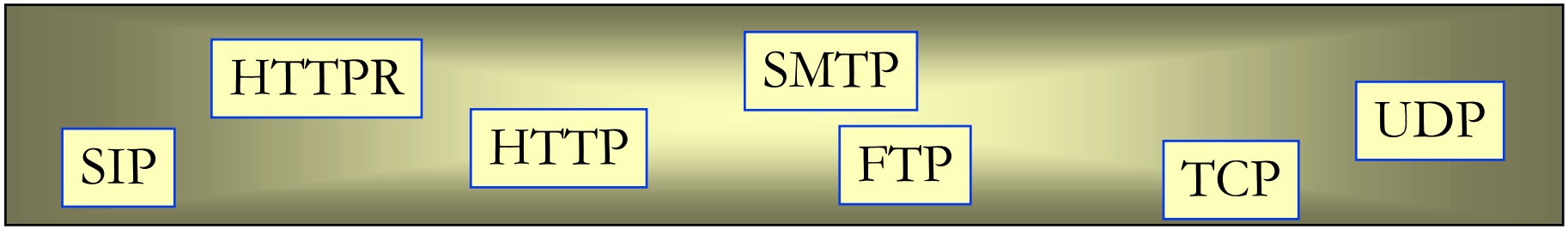


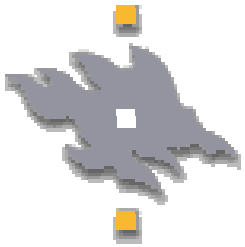
Web Services Architectural Stack Elements

Message Transportation

- Means of transporting XML between Web Services end points / nodes
- Bindings today are only widely applied to HTTP
- There could several different manners of transportation for Web Services protocols in the future.

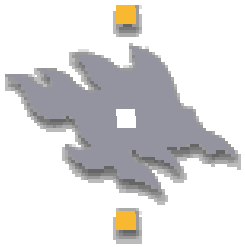






Message Transportation - HTTPR

- Purpose:
  - HTTP provides reliable delivery of HTTP packets as long as failure does not occur. If there is a failure there is not means to recover or trace the cause of error.
- HTTPR solves this by:
  - Providing a protocol that provides reliable delivery of HTTP packets between the server and client
  - Provides rules that make it possible to ensure that all messages are delivered to their destination in their exact form and only once.
- HTTPR:
  - Defines how metadata and application messages are encapsulated
  - Message is delivered to its destination application exactly once or then a failure is notified to the requester/responder correspondingly
  - Emphasises the storage of the messages prior to successful exchange of the messages between client & Servers
  - Further reading : http://www-106.ibm.com/developerworks/webservices/library/ws-phtt/



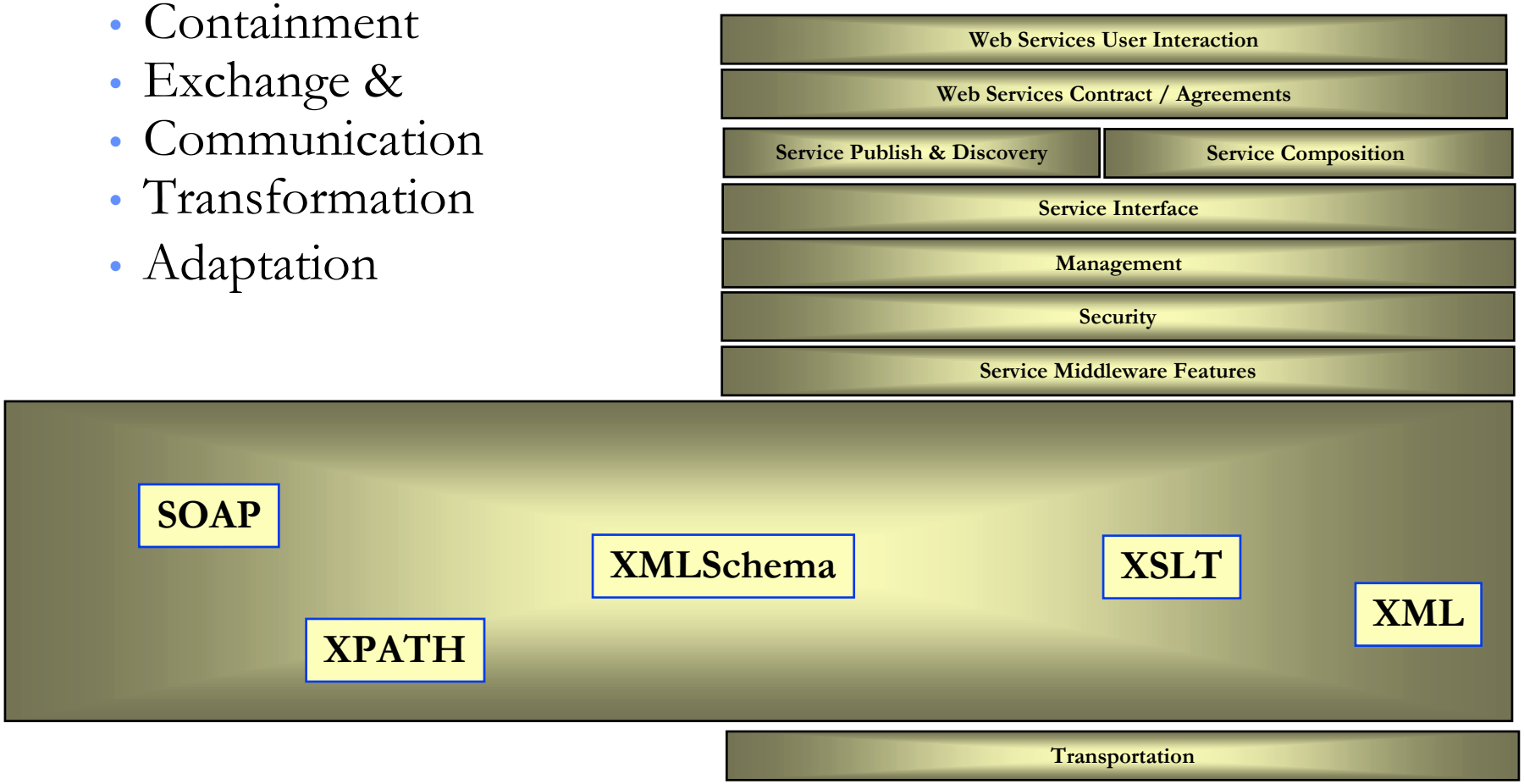
Message Transportation - SIP

- SIP is a Session Intiation Protocol
- SIP is a text-based protocol, similar to HTTP and SMTP, for initiating interactive communication sessions between users.
- SIP can be used in the Web Services world to set up a connection between two Web Services end points.
- SIP works on many different types of transportation protocols such as TCP and UDP.

- **Conclusion:** Web Services provide open standards based services over the Web and SIP could allow them to be presented within integrated communication applications that include Voice and multi-media services.
  Further reading:
  - SIP Specifications: http://www.ietf.org/html.charters/sip-charter.html
  - SIP & SOAP: http://www.sipcenter.com/files/Ubiquity_SIP_and_SOAP.pdf

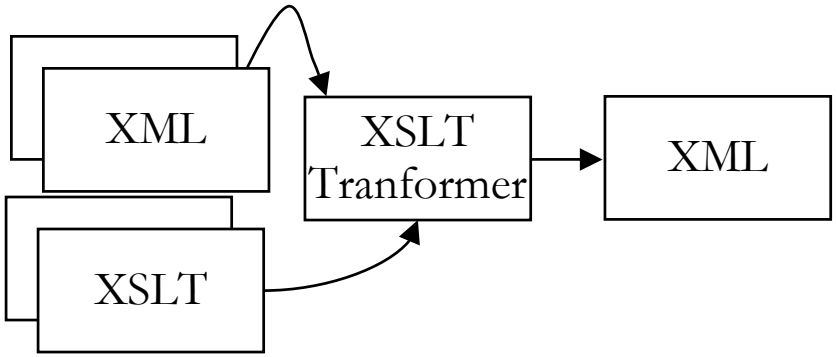
Message Containment & Communication

- An Application Message:
  - Definition
  - Containment
  - Exchange &
  - Communication
  - Transformation
  - Adaptation



Message Containment & Communication - XSLT

- XSLT is a language for transforming XML documents into other XML documents

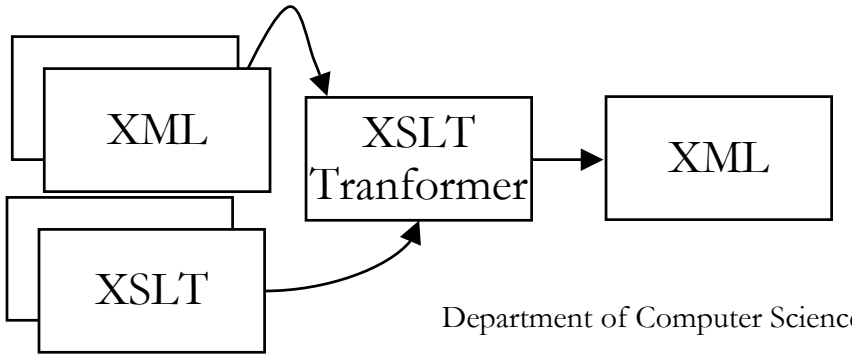


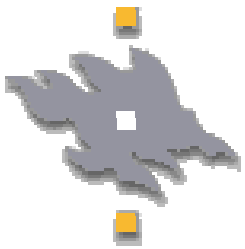
- It is usually used as part of the XSL which defines the formatting styles for XML
- This language specification defines the syntac and semantics of transforming XML documents. This specifications is referred to be the namespace : http://www.w3.org/1999/XSL/Transform
- **Conclusion:** XSLT can be utilised in Web Services for message adaptation and transformation of messages to handle message incompatibilities and to some extent also define Stylesheet for presentation purposes

Message Containment & Communication - XSLT

- XSLT describes rules for transforming a source XML document into a targer XML document
- The rules consists of associated patterns with templates.
  - A pattern is matched against elements in the source XML Document using the XPATH.
  - A template is instantiated to create part of the target XML Documents.
- The elements from the source XML document are used to alter in the transformation process and included as part of the target document

Further Reading:
http://www.w3.org/TR/xslt





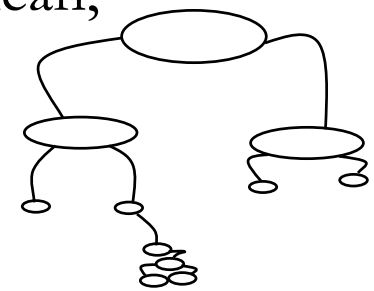
Message Containment & Communication - XMLSchema

Will be covered in more detail later in the course

- XML Schema provides a method to define the structure, content and semantics of XML documents.
- An XML document which complies to a Scheme defined by an XML Schema is called an Instance Document.
- XML Schemas are Namespace aware and hence can leverage reuse of schema definitions
- XML schema is defined in terms of:
  - **Simple Types:** A Simple XML elements which do not have subelements/attributes: simple data types, enumeration, lists, restricted formatted values(patterns/regular expression evaluated)
  - **Complex Types:** An XML Element which could contain 1 or more subelements and attributes.
    - This also defines the sequence, Choice of occurrence, number of occurrence (min/max), default values for attributes
  - **Global / Local Element types:** Scope of the element declarations
- XML Schema Namespace : http://www.w3.org/2001/XMLSchema
- **Further reading:** http://www.w3.org/XML/Schema

Message Containment & Communication - XPATH

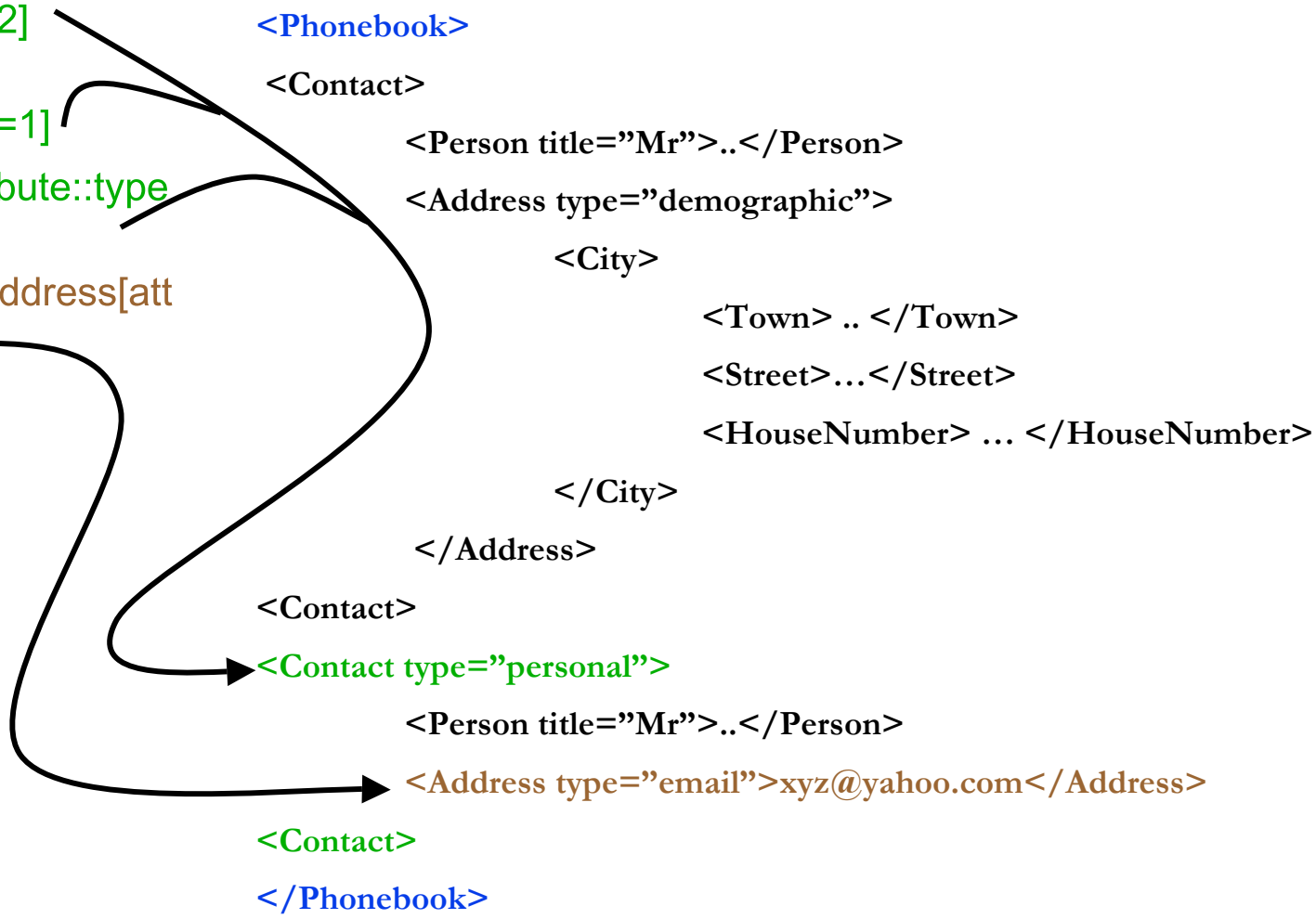
- This is a language for addressing parts of an XML document and matching nodes to patterns
- XPATH represents the XML Document as a tree of node elements, the node being : Elements,Attributes, Text, Process Instructions, Comments, etc..
- XPATH uses an non-XML syntax to address and to specify the expressions referring to nodes/node element, so as to be utilised as value of attributes.
- The Expressions consists of an addresssing mechanism with specific functions over the value/name/structure of the tree elements
- The result of the expression could lead to : node-set, boolean, number, string



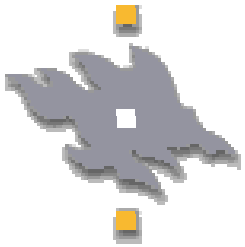
Further Reading : http://www.w3.org/TR/xpath

Message Containment & Communication - XPATH

- child::Contact[position()=2]
- following-sibling::contact[position()=1]
- /Phonebook/Contact[attribute::type="personal"]
- /Phonebook/Contact[2]/Address[attribute::type="email"]



Further Reading : http://www.w3.org/TR/xpath



Message Containment & Communication - SOAP

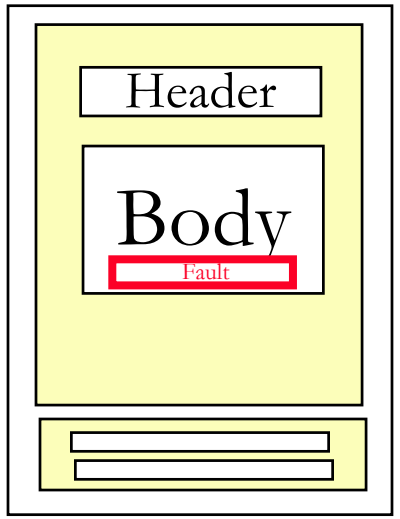
Will be covered in more detail later in the course

- Simple and flexible messaging framework for transferring information specified in the form of an XML infoset between an initial SOAP sender and an ultimate SOAP receiver. SOAP **does not define** application semantics but **defines** a mechanism to express application semantics.
- ”SOAP defines a simple and lightweight mechanism for exchanging structured and typed information between peers over the web in a decentralized, distributed environment using XML.”

Message Containment & Communication - SOAP

- Specifies :
  - Message Parts: Headers, Body, Fault and Attachments
  - Rigid rules for construction of SOAP messages
  - Message processing of SOAP messages : orginatory Intermediaries, ultimate receipient
  - Transportation of SOAP Messages over HTTP, SMTP as means of message exchanges
  - Message Exchange patterns : Request-Response
  - RPC

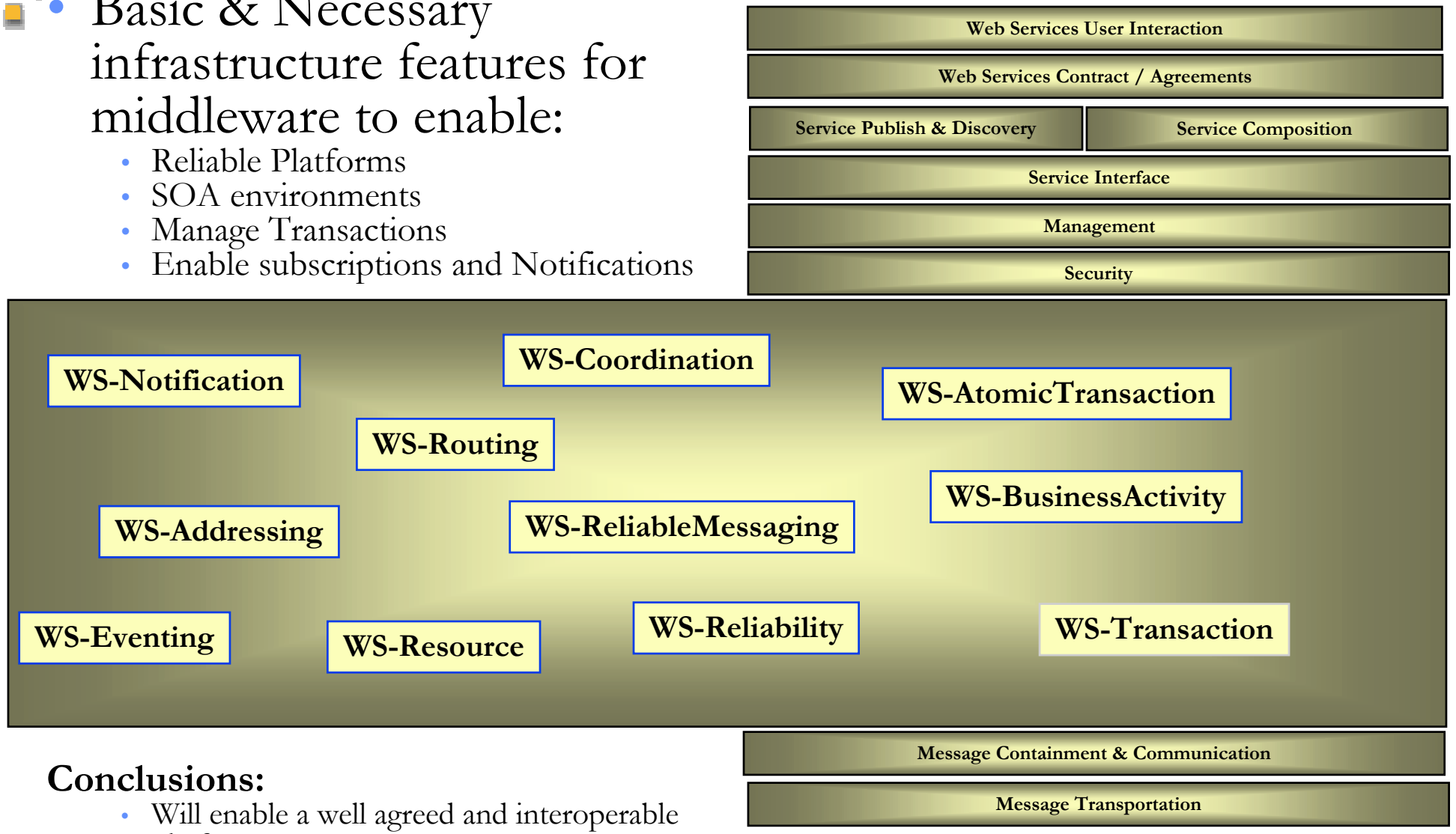
Will be covered in more detail later in the course



- Further reading: http://www.w3.org/2000/xp/Group/

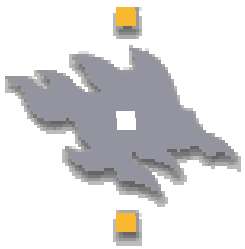
Service Middleware Features

- Basic & Necessary infrastructure features for middleware to enable:
  - Reliable Platforms
  - SOA environments
  - Manage Transactions
  - Enable subscriptions and Notifications



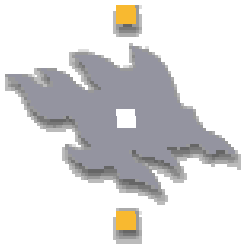
Conclusions:

- Will enable a well agreed and interoperable platform



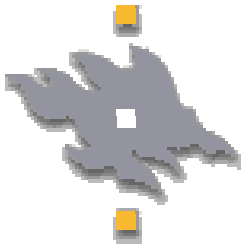
Service Middleware Features – Reliability

- Two or more Web Services nodes or end points can communicate over different transportation protocols relying on the fact that the messages exchanged between them are Guaranteed and done so in predictable manner, even with the assumption that there could be network failures.
- There are two related approaches for ensuring Web Services Reliability:
  - **WS-Reliability:** Driven by OASIS (EbXML): 1.0 - 6th January 2004
  - **WS Reliable Messaging:** Driven by IBM, Bea, Microsoft, Tibco, - 13th March 2003
- These specifications a provide a SOAP based protocols to ensure reliable messaging, gauranteed delivery with right message ordering, no duplicates



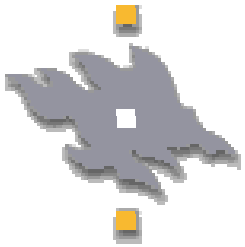
Service Middleware Features – Reliability [Contd..]

- The reliability aspects covered in these specifications span in addressing the following features:
  - Uniquely being able to identify the message, discard duplicates
  - Message order sequencing, Message persistance for recovery
  - Reliability applied to a group of message communications
  - Acknowledgements – on receipt & delivery of messages
  - Failure recovery, using levels of persistence as needed
    - Timeouts & Expiry
    - Re-transmissions
  - Provides HTTP based binding, but the value lies in not restricting the need for HTTP



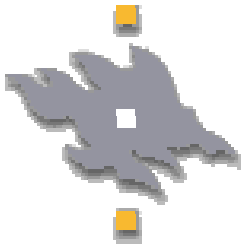
Service Middleware Features – Resource

- Is a set of specifications to access stateful resources of Web Services.
- Web Services can expose resources available over the web and provide corresponding interfaces and define message structures to:
  - Manage and control the resource properties
  - Query, Lease and Expire the validity of the existence of web resources
- This is supported by two specification proposals, namely :
- **Resource properties:**
  - Managing the properties of a Web Resource:
    - Accessing properties of the resource
    - Updating the properties of the resources.
    - Inserting / Deleting resource properties
    - Query resource properties
    - Subscribe for changes of resource properties



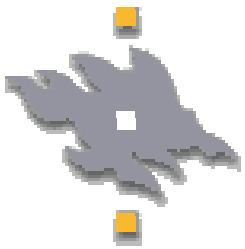
Service Middleware Features – Resource

- Resource Time Span:
  - Managing the expiry and validity of the Resource
    - Any client of a WS-Resource may establish and extend the scheduled termination time of a WS-Resource.
    - WS-Resource can detroy itself without the need for an explicit destroy request message from a client.
    - the WS-Resource can be destroyed at the very instant of time when a request is scheduled or can be scheduled to be destroyed.
    - Lease and renew the termination time of a WS-Resource



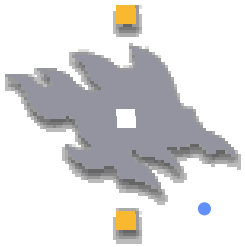
Service Middleware Features – Transaction

- This ensures that all activities in Web Services, meaning a set of message exchanges including the processing of the messages have been succesfully executed or none are allowed to succeed.
- The set of service invocations together are considered to belong to one consistent behaviour and the success of thier execution is gauranted by the success of all other activities
- The effects of such set of executions is commited to the system only once all the activities have confirmed thier execution



Service Middleware Features – Transaction [Contd]

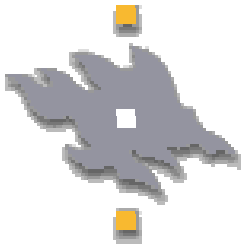
- Transaction in Web Services are considered at two levels:
  - **Atomic Transactions:** is used to coordinate activities having a short duration and executed within limited trust domains. The transaction propogates the properties of "all or nothing". In the case of atomic transaction there is a possibility to hold the resources until the commit as the transactions are short lived
  - **Business level transactions:** This is long lived transactions as the transactions happen across different business boundaries, i.e. in different business partners boundaries. The transactional aspects are handled differently in these environments as the resources can be held until completion of the transaction(due to long live). Instead a loosely coupled isolation of concerns and instead of locking compensational effects are carried our to roll back incase of failures.



Service Middleware Features – Routing

- This is a simple and light weight SOAP header based protocol to define the basically paths of message transportation independent from the transportation protocol.
- The purpose of this being that SOAP can be bound to any lower transportation protocol and maintain the simple and easy routing mechanism without a need to specify the transport level routing
- This specifications addressed the both the forward and return path of the Message paths
  - Forward path : is the route through which the message is transported from the initial sender via zero or more intermediaries and finally to the ultimate receipient
  - Reverse : Is the route of the message transportation from the ultimate receipient via zero or more intermediaries to the initial message originator
  - Message co-relations: the inter-relations between the messages exhanged across the routing.
- Routing is specified using the SOAP header entries: ***action, to, from, via, fwd, rev,id, relatesTo, fault***
- WS Routing Namespace : http://schemas.xmlsoap.org/rp/

Further Reading :
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnglobspec/html/ws-routing.asp

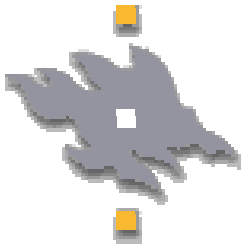


Service Middleware Features – Address

- Addressing in Web Services provides transport-neutral (for ex: HTTP/SMTP) mechanisms to address Web services and messages
- This specification defines XML elements to identify Web service endpoints and to secure end-to-end endpoint identification in messages

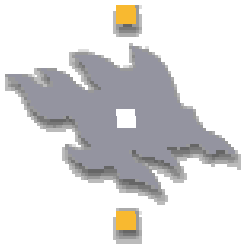
- This specification is defined based on SOAP Header entries containing:
  - ***Message ID, RelatesTo, To, Action, From, ReplyTo, FaultTo, Recipient, EndPointReference(Address, ReferenceProperties, PortType, ServiceName, Policy?)***

- **WS Addressing Namespace:** "http://schemas.xmlsoap.org/ws/2003/03/addressing
- **Specified by:** IBM, BEA, Microsoft - March 2003
- **Further reading:** http://www-106.ibm.com/developerworks/webservices/library/ws-add/



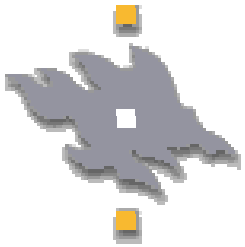
Service Middleware Features – Events

- This specification defines a protocol that allows Web services to subscribe to or accept subscriptions for event notification messages.
- **Provides an ability:**
  - For one Web Services to Subscribe to Events generated by another Web Services
  - Allow the subscriber to relate the events being notified to the subscription
  - Leasing and Expiry of Subscription
- WS Addresssing is utilised for addressing Web Services as part of the SOAP header entries
- The protocol uses the SOAP body to propogate the subscription requests and event responses.
- The elements used in the protocol are : ***Subscribe, SubscribeResponse, Id, NotifyTo, EndTo, Expires, Filter, Renew, RenewResponse, UnSubscribe, SubscriptionEnd, Code, Reason***
- **Further reading:** http://ftpna2.bea.com/pub/downloads/WS-Eventing.pdf



Service Middleware Features – Notification

- A protocols based on SOAP and WSI for Services to notify about events or messages to other services as an act of response to a subscription
- This is based on a topic – oriented publish / subscribe Messaging exchange pattern, which contains standard means for service providers, notification brokers to publish messsages including the operational requirements for Publishers and subscribers based on XML model of topics
- The features considered by this Specifications are:
  - Independence from the message transportation protocol
  - Ability to include the MOM as the implementation platform
  - Notification Service provider/broker
  - Message Topic transformation and aggregation
  - Metadata about subscribable Message topics

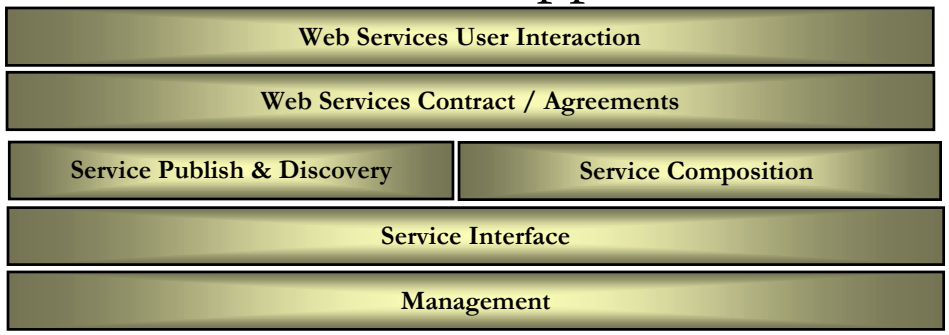


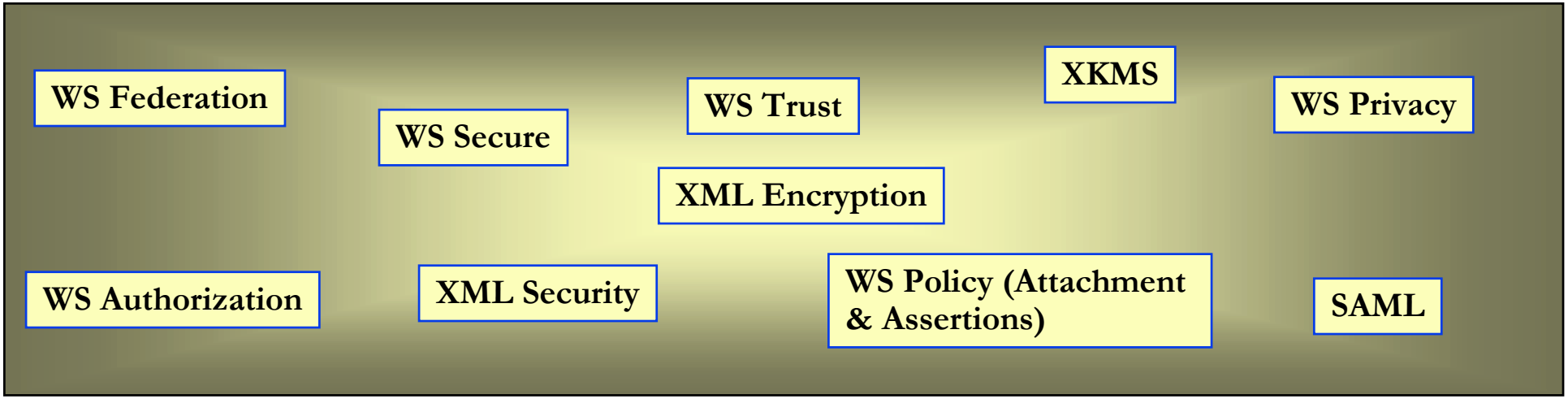
Service Middleware Features – Notification

- Notification provides roles for multiple players: Publisher, Subscriber, NotificationBroker, SubscriptionManager,
- WS-Notification Namespace :
  http://www.ibm.com/xmlns/stdwip/web-services/WS-Notification
- This is a specification proposed by: IBM, Sonic Software, Tibco, HP. Akamai, SAP, Globus, Argonne National laboratory **(20 January 2004)**
- **Further Reading:** http://www-106.ibm.com/developerworks/library/ws-resource/ws-notification.pdf

Security

- In order for Web Services to be utilised in Business Critical Applications It is very important for security to be considered as a key aspects of Web Services





- Web Services security is being addressed under the above several activities