UNIVERSITY OF Southampton

Topics on Web Services

COMP6017

Dr Nicholas Gibbins – nmg@ecs.soton.ac.uk

2013-2014

UNIVERSITY OF Southampton

Module Aims

- Introduce you to service oriented architectures
- Introduce you to both traditional and RESTful Web Services
- Give you in-depth knowledge of Web Services
- Give you practical hands-on experience of RESTful Web Services

University of Southampton

People

Nick Gibbins

Enrico Costanza

Office hours for nmg: Friday 1100-1200

UNIVERSITY OF Southampton

Module Structure

One double lecture each week

- Friday 0900-1100, 05/2015
- Note: extra slots in timetable in week 4 are incorrect

Assessment

- 75% Exam
- 25% Group Coursework

UNIVERSITY OF Southampton

Coursework

One coursework worth 25% of your final mark

- Groups of three
- Construct a RESTful web service and client using JavaScript and node.js
- Coursework specification published on Friday of week 5
- Deadline on Tuesday of week 11 (10th December)
- Feedback by the end of week 12

UNIVERSITY OF Southampton

Teaching Schedule

Week 1	Overview Web Services Architecture Architecture of the World Wide Web
Week 2	CANCELLED
Week 3	Web Protocols: HTTP REST and Resource Oriented Architectures
Week 4	Web Protocols: SOAP REST in Practice
Week 5	Service Description: WSDL Introduction to JavaScript and node.js

University of Southampton

Teaching Schedule

Week 6	Service Discovery: UDDI Coursework Briefing
Week 7	Addressing and Policy Coursework Support
Week 8	Security Coursework Support
Week 9-11	Coursework Support
Week 12	Review

UNIVERSITY OF Southampton

Service Orientation

UNIVERSITY OF Southampton

Service Orientation

- Component-based software design paradigm
- Organise and use heterogeneous distributed capabilities

- Many existing technologies:
  - Java RMI, CORBA, DCOM, WCF, Web Services, REST

UNIVERSITY OF Southampton

What is a service?

- Services as contractually defined behaviours
- Services as task-performing components
- Services as collections of related capabilities
- Services combine information and behaviour

8 Principles for Service Orientation

- Loose coupling
- Shared formal contracts
- Abstraction
- Composability
- Reusability
- Autonomy
- Statelessness
- Discoverability

Loose Coupling

Coupling is a measure of the degree of dependency between components

- Tight coupling limits flexibility
- Loose coupling promotes ad hoc reuse of components

UNIVERSITY OF Southampton

Shared Formal Contract

Services have descriptions that document:

- programmatic interface
- communication requirements and protocols
- constraints
- usage policies

Abstraction

Service contract describes the external view of a service

- service internals are hidden
- limits formation of dependencies (loosens coupling)

Service internals may change with minimal impact on clients

Composability

Service abstraction allows the encapsulation of other services

- internally, a service may be a client of other services
- services may aggregate several services

Reusability

Separation of concerns encourages reuse of components

- Service contracts describe services to clients

UNIVERSITY OF Southampton

Autonomy

Services exist independently

Services control their underlying logic

- subject to commitments made in service contracts

Statelessness

State consists of data specific to current activity

- State management consumes resources

Statelessness increases scalability and availability

Discoverability

Service discovery is key to SOA

- service contracts describe services to facilitate consumption by potential clients
- clients may search for services by the features of their contracts

UNIVERSITY OF Southampton

Architecture

University of Southampton

What is an architecture?

- Logical architecture
- Process architecture
- Development architecture
- Physical architecture

Logical Architecture

Primarily supports the functional requirements

- i.e. what the system should provide in terms of services to its users.

The system will be decomposed into a set of abstractions, and their high level interactions will be identified

Process Architecture

Takes into account some non-functional requirements, such as performance and availability

Addresses issues of concurrency and distribution, of system integrity, of fault-tolerance

UNIVERSITY OF Southampton

Development Architecture

Focuses on the actual software module organisation, including libraries

Physical Architecture

Takes into account primarily the non-functional requirements of the system

- availability
- reliability
- performance
- scalability

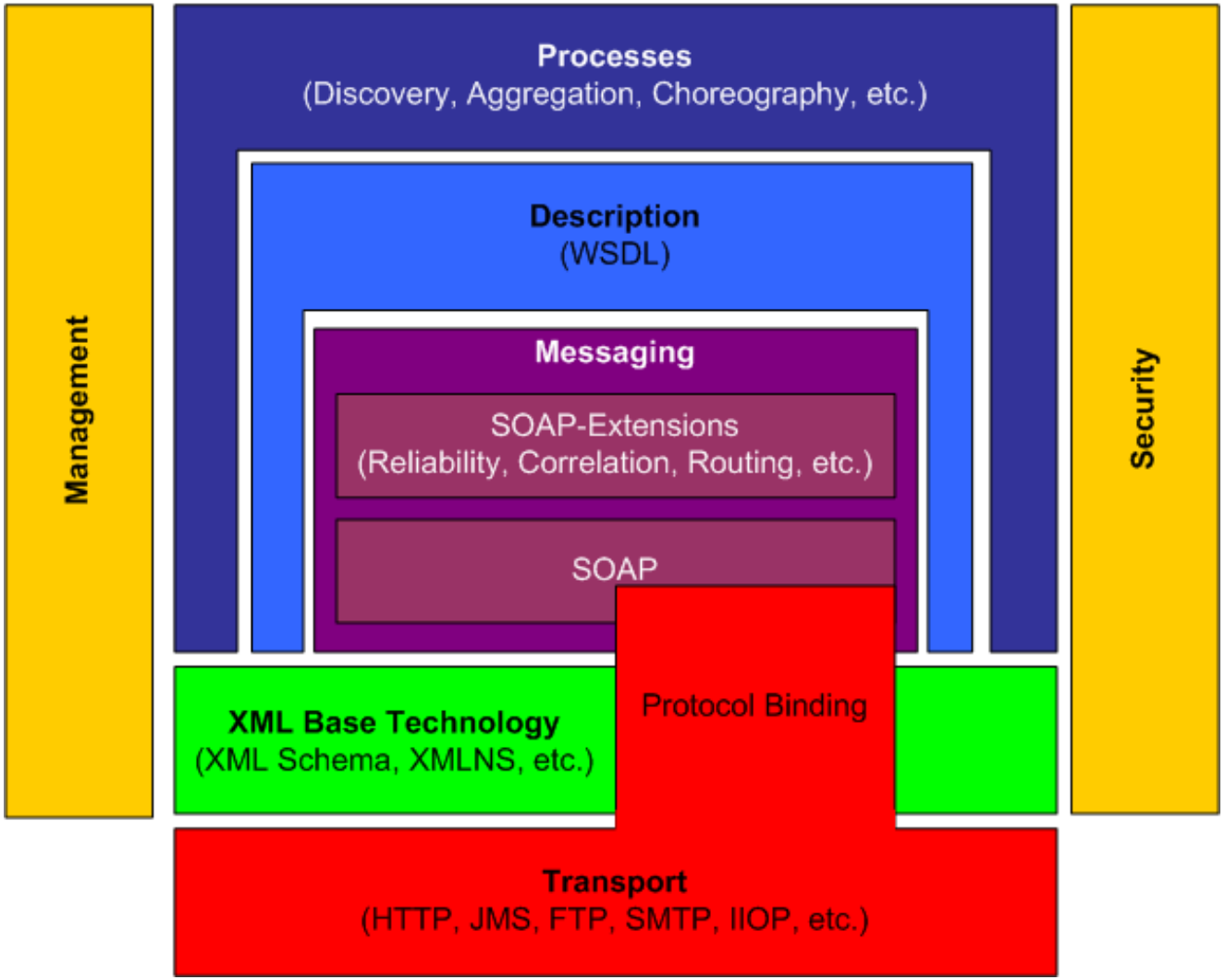
… what about COMP6017?

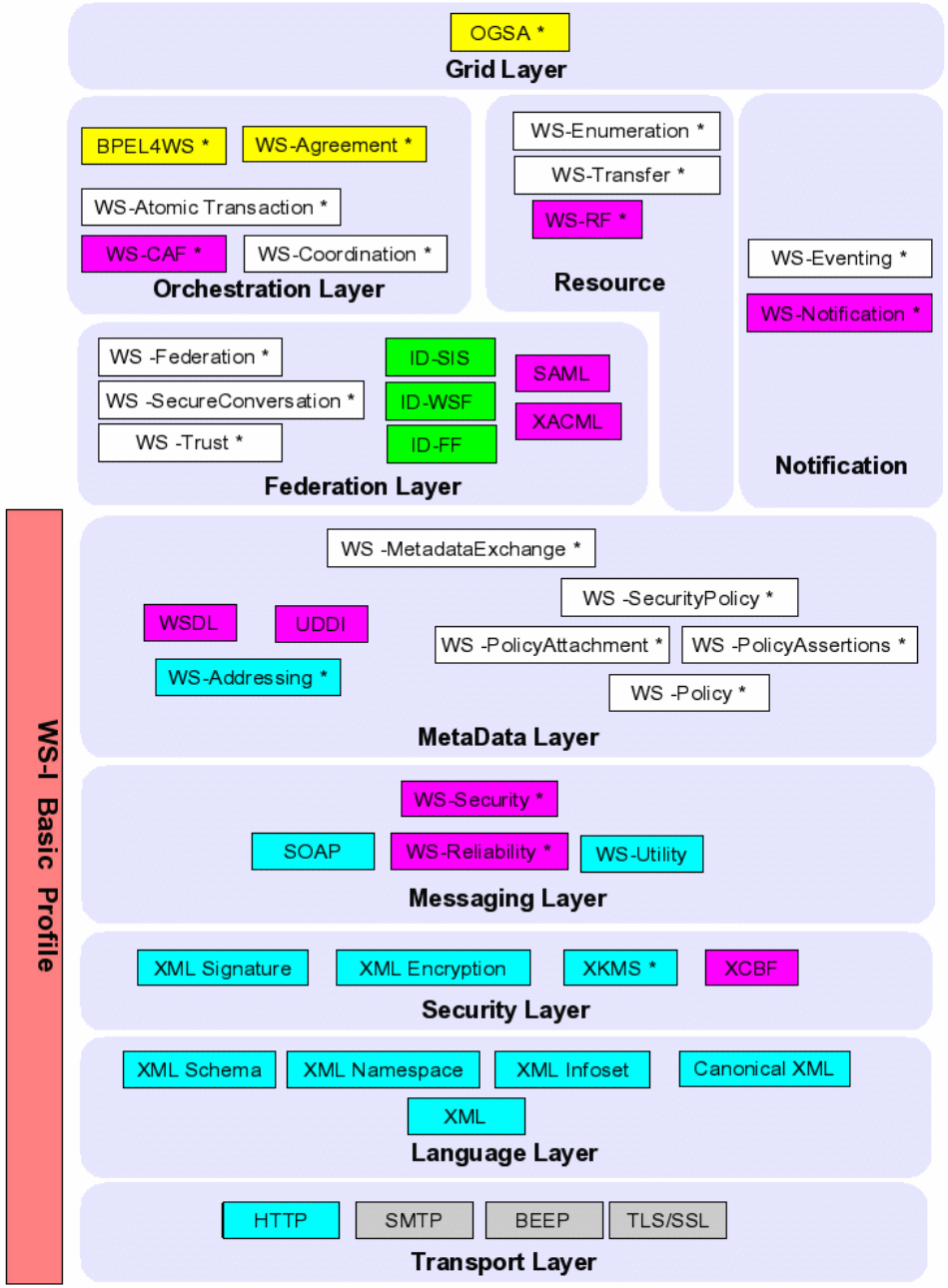
- We are essentially going to cover a logical architecture, identifying core functionality offered by Web Services.
- Some elements of process architecture, and in particular concurrency and distribution, will be addressed too.
- We discuss the development architecture, when investigating REST.

UNIVERSITY OF Southampton

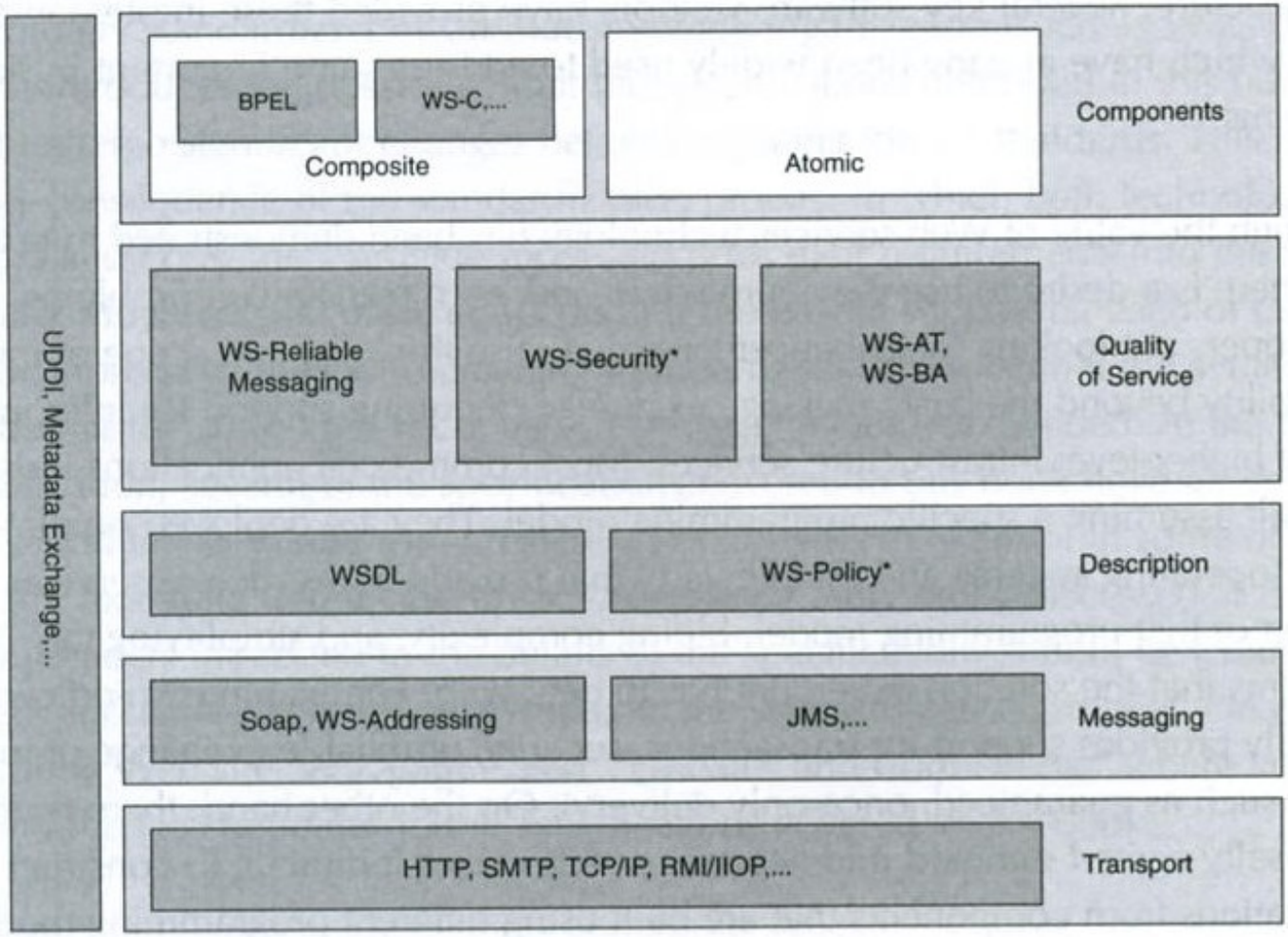
Introduction to Web Services

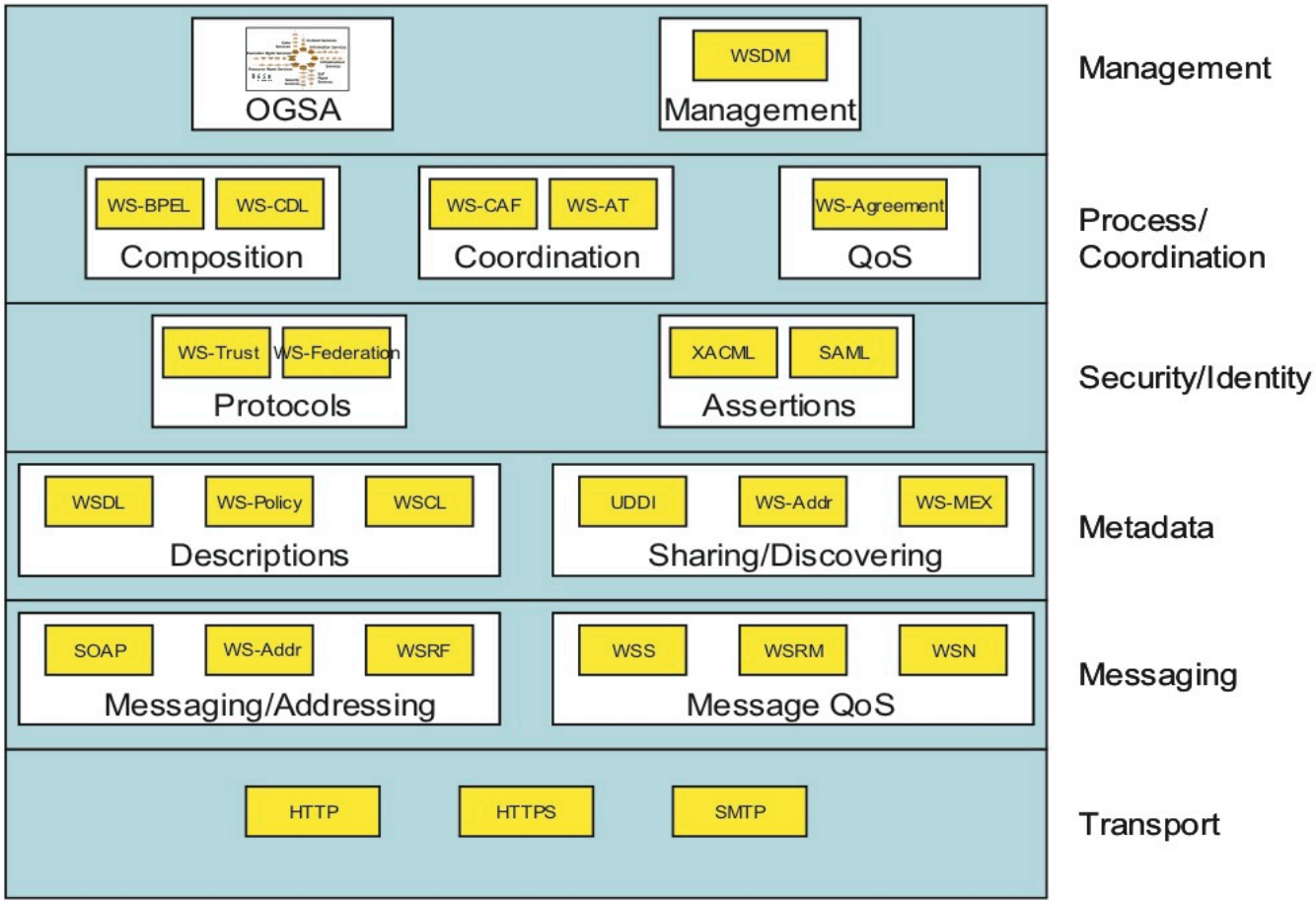
WSA Stack diagram

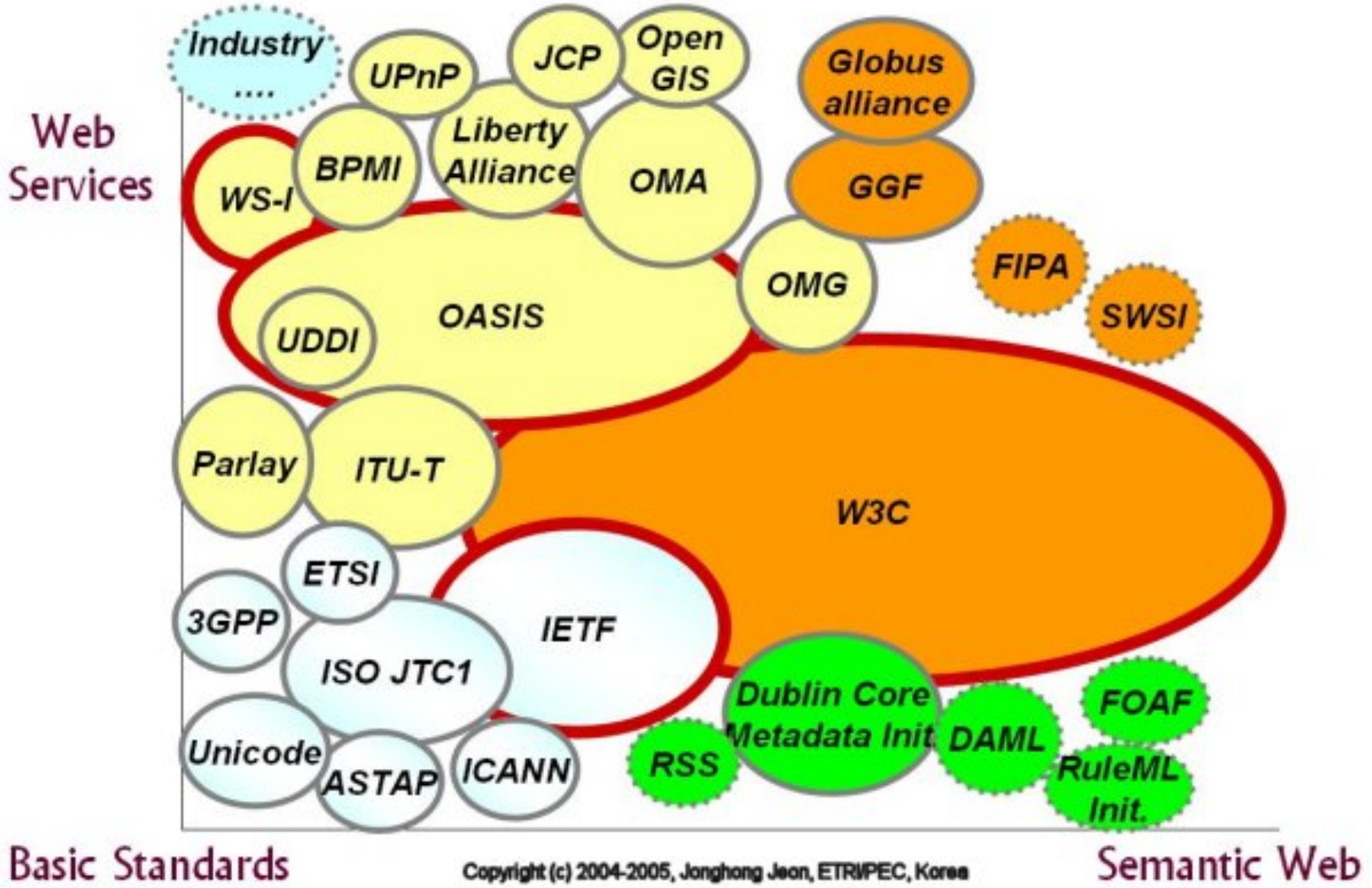



OGSA *
Grid Layer
BPEL4WS *
WS-Agreement *
WS-Atomic Transaction *
WS-CAF *
WS-Coordination *
Orchestration Layer
WS-Enumeration *
WS-Transfer *
WS-RF *
Resource
WS-Eventing *
WS-Notification *
Notification
WS -Federation *
WS -SecureConversation *
WS -Trust *
ID-SIS
ID-WSF
ID-FF
SAML
XACML
Federation Layer
WS -MetadataExchange *
WSDL
UDDI
WS-Addressing *
WS -SecurityPolicy *
WS -PolicyAttachment *
WS -PolicyAssertions *
WS -Policy *
MetaData Layer
WS-Security *
SOAP
WS-Reliability *
WS-Utility
Messaging Layer
XML Signature
XML Encryption
XKMS *
XCBF
Security Layer
XML Schema
XML Namespace
XML Infoset
Canonical XML
XML
Language Layer
HTTP
SMTP
BEEP
TLS/SSL
Transport Layer
WS-I Basic Profile

구 분		프로토콜명 (표준기구명)	표준채택 예상 시기		
			2003	2004	2005
Management	Distributed Management	WSDM(OASIS)	S E EA		
Security	Security	WS-Security(OASIS)	E EA M		
	Security Policy	WS-SecurityPolicy(N/A)	S E EA M		
	Secure Conversation	WS-SecureConversation (N/A)	S E EA M		
	Trusted Message	WS-Trust(N/A)	S E EA M		
Discovery	Discovery / Publication	UDDI(OASIS)	EA M		
	Inspection	WSIL(N/A)	EA M		
Description	Portal	WSRP(OASIS)	S E EA M		
	Transaction	WS-Transactions(N/A)	S E EA M		
		WS-Coordination(N/A)	S E EA M		
	Orchestration	BPEL4WS(OASIS)	S E EA M		
		WS-Choreography(W3C)	S E U		
	Presentation	WSIA	S		
	Policy	WS-Policy(N/A)	S		
	Implementation / Interface	WSDL(W3C)	EA M		
Transport	Routing/ Addressing	WS-Addressing(N/A)	S E EA M		
	Reliable Messaging	WS-ReliableMessaging (OASIS)	S E EA M		
		WS-Reliability(OASIS)	S E U		
	Packaging	SOAP(W3C)	EA M		
		WS-Attachments(IETF)	S E EA M		
		DIME(IETF)	S E EA M		
	Transport	HTTP, TCP, SMTP, etc			


UDDI, Metadata Exchange,...
BPEL
WS-C,...
Composite
Atomic
Components
WS-Reliable Messaging
WS-Security*
WS-AT, WS-BA
Quality of Service
WSDL
WS-Policy*
Description
Soap, WS-Addressing
JMS,...
Messaging
HTTP, SMTP, TCP/IP, RMI/IIOP,...
Transport


OGSA
WSDM
Management
Management
WS-BPEL
WS-CDL
Composition
WS-CAF
WS-AT
Coordination
WS-Agreement
QoS
Process/
Coordination
WS-Trust
WS-Federation
Protocols
XACML
SAML
Assertions
Security/Identity
WSDL
WS-Policy
WSCL
Descriptions
UDDI
WS-Addr
WS-MEX
Sharing/Discovering
Metadata
SOAP
WS-Addr
WSRF
Messaging/Addressing
WSS
WSRM
WSN
Message QoS
Messaging
HTTP
HTTPS
SMTP
Transport



Web Service Definition

- A Web Service is a software system designed to support interoperable machine-to-machine interaction over a network.
- It has an interface described in a machine-processable format (specifically WSDL).
- Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

(Web Service Glossary)

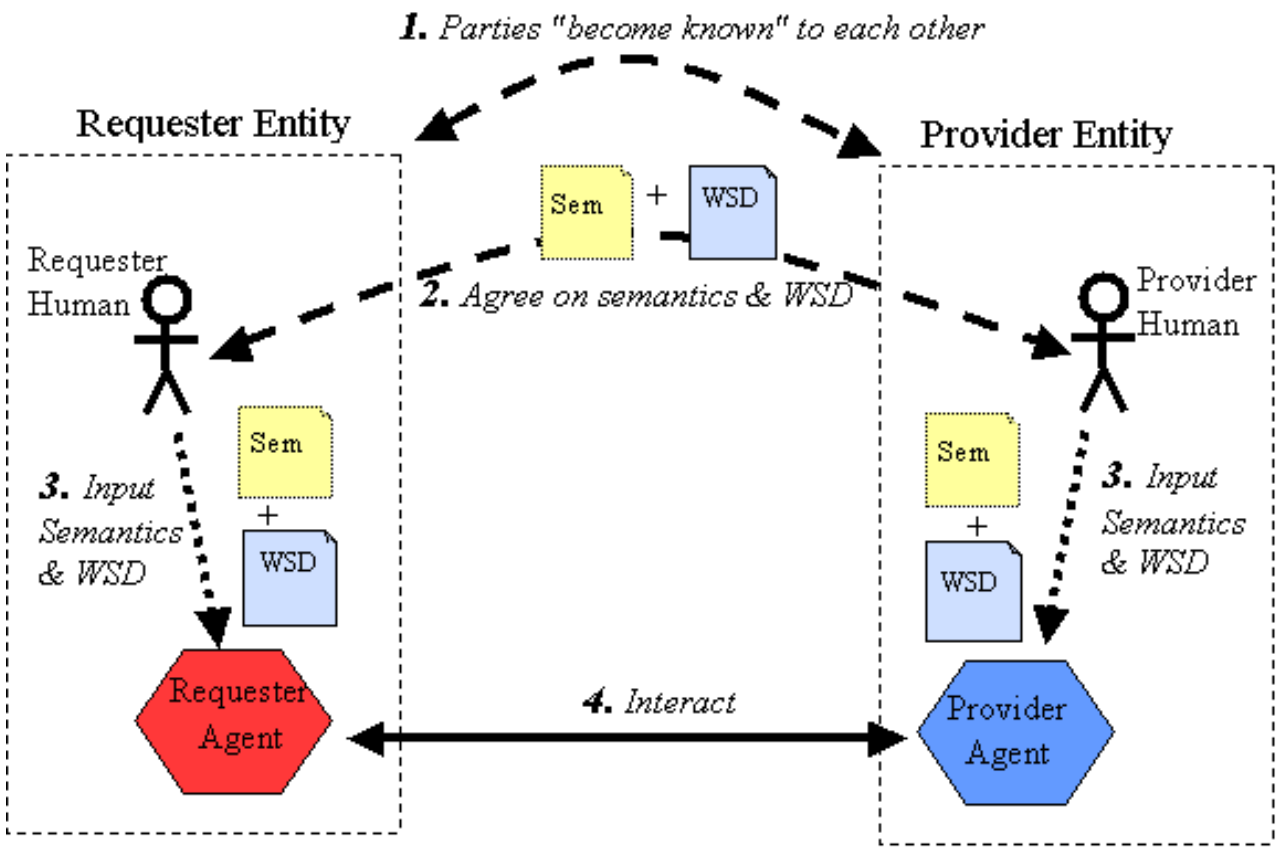
Discussion

- Not universally accepted
- Too minimalist: does not mention policies, choreography, security
- Too specific, refers to standards that are not universally accepted (e.g., wsdl)

Service Definition

- A service is a mechanism to enable access to a set of capabilities, where the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the service description.
- A service is provided by one entity for use by others, but the eventual consumers of the service may not be known to the service provider and may demonstrate uses of the service beyond the scope originally conceived by the provider. (OASIS SOA Reference Model)

Engaging a Web Service



Roles (1): service vs. agent

- A **Web Service** is an abstract notion that must be implemented by a concrete agent.
- The **Agent** is the concrete piece of software that sends and receives messages, while the service is the resource characterized by the abstract set of functionality that is provided

Roles (2): provider vs requester

- The service provides functionality on behalf of its owner (a person or organisation): the **provider entity**.
- A **requester entity** is a person or organization that wishes to make use of a provider entity's Web service.
- A requester entity uses a **requester agent** to exchange messages with the provider entity's **provider agent**.

Roles (3): description

- The mechanics of the message exchange are documented in a Web Service **description**.
- It defines the
  - message formats,
  - datatypes,
  - transport protocols, and
  - transport serialization formats

that should be used between the requester agent and the provider agent.

Roles (4): semantics

- The **semantics** of a Web service is the shared expectation about the behaviour of the service, in particular in response to messages that are sent to it.
- In effect, this is the **contract** between the requester entity and the provider entity regarding the purpose and consequences of the interaction.

Engaging a Web Service

1. The requester and provider entities become known to each other (or at least one becomes known to the other);
2. The requester and provider entities somehow agree on the service description and semantics that will govern the interactions between them;
3. The service description and semantics are exploited by the requester and provider agents;
4. The requester and provider agents exchange messages. (I.e., the exchange of messages with the provider agent represents the concrete manifestation of interacting with the provider entity's Web service.)

UNIVERSITY OF Southampton

Web Service Models

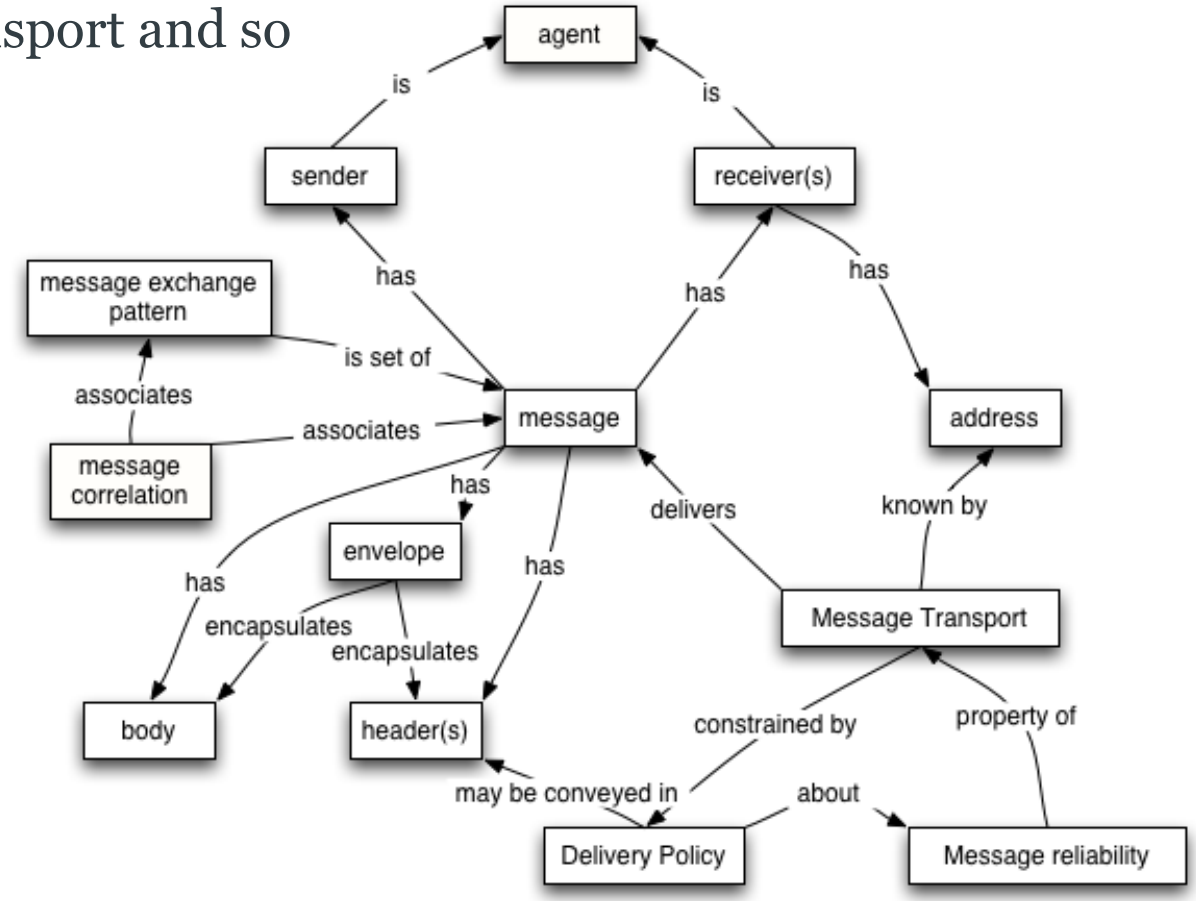
UNIVERSITY OF Southampton

Web Service Models

- Message Oriented Model
- Service Oriented Model
- Resource Oriented Model
- Policy Model

Message Oriented Model

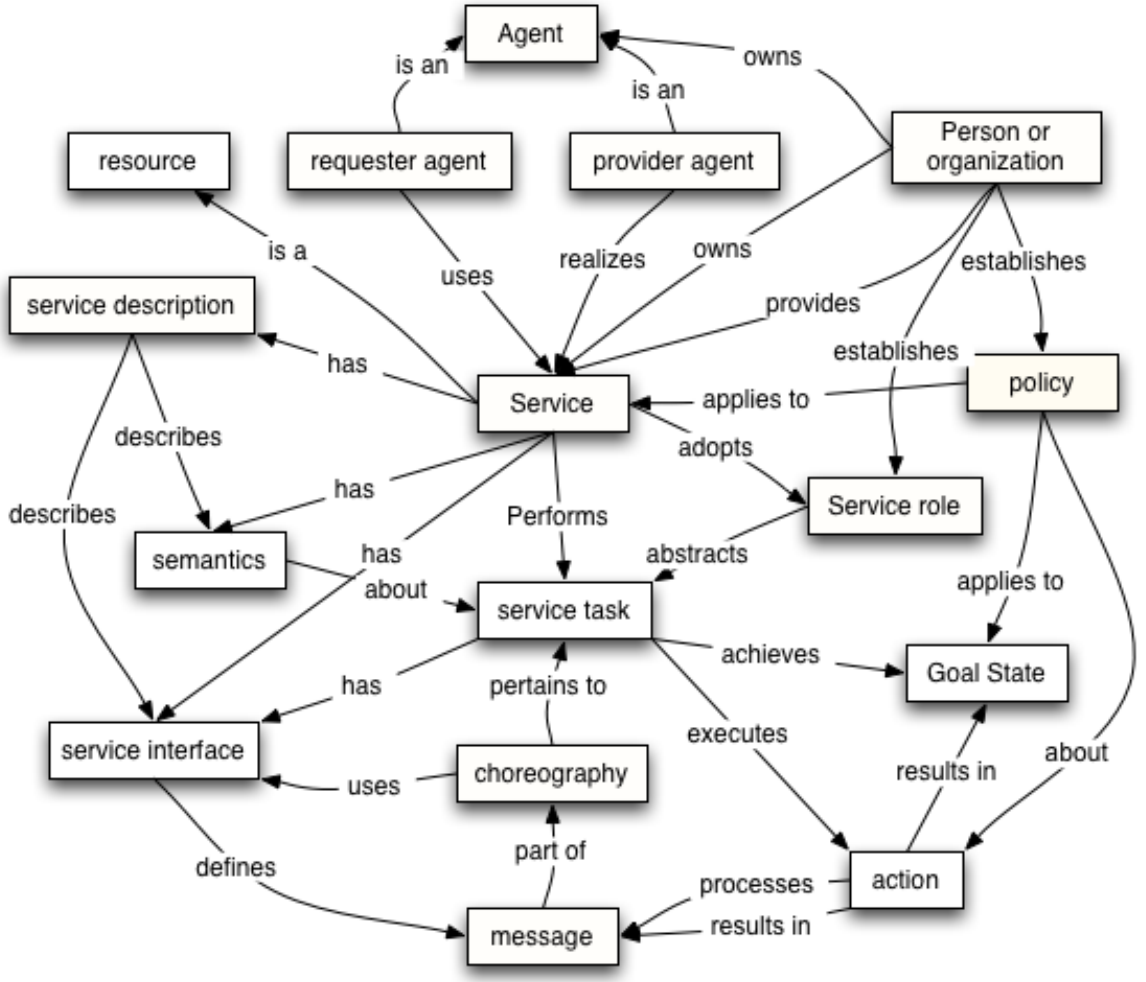
Focuses on messages, message structure, message transport and so on



Service Oriented Model

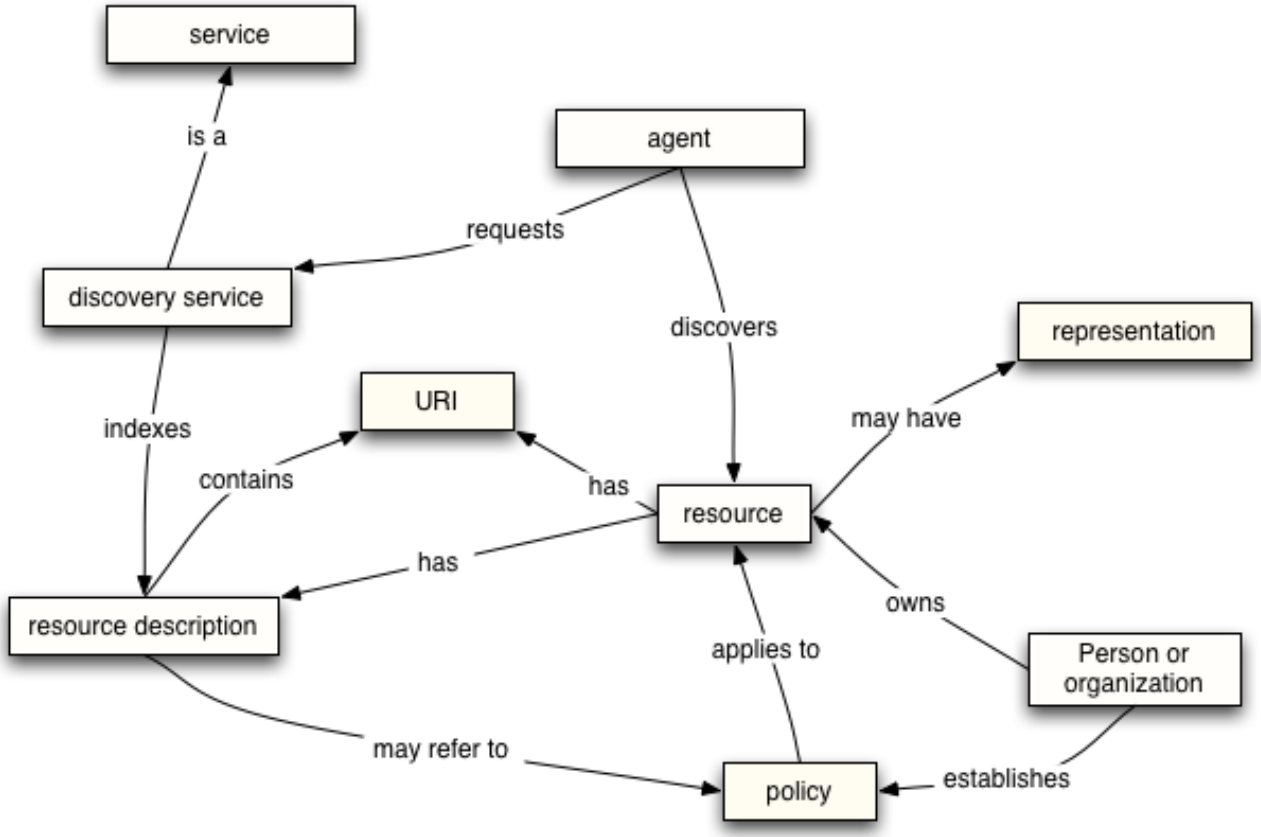
Focuses on aspects of service, action and so on.

- In any distributed system, services cannot be adequately realized without some means of messaging, the converse is not the case: messages do not need to relate to services.



Resource Oriented Model

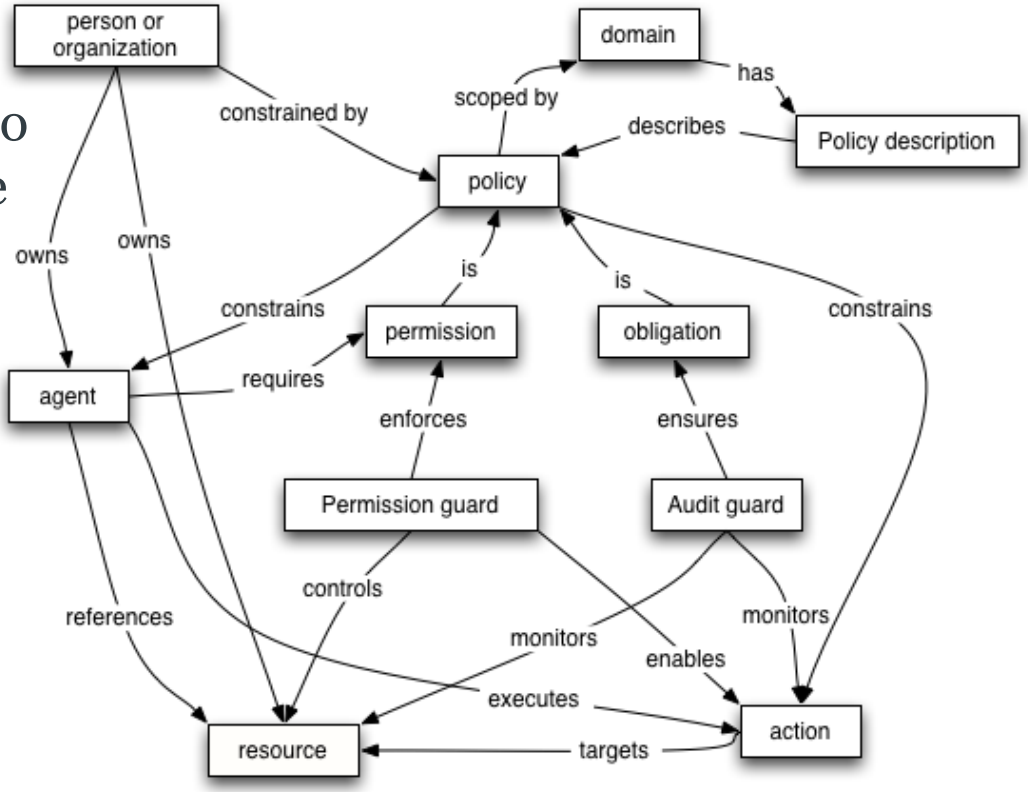
Focuses on resources that exist and have owners.



Policy Model

Focuses on constraints on the behaviour of agents and services.

- Generalize to resources
- Policies can apply equally to documents (such as service descriptions) as well as active computational resources.



Conclusion

- No agreed definition of what a Web Service is, but a reasonable intuition
- Not a single vision for Web Services (multiple standardisation committees, etc)
- Multiple and competing WS stacks
- Architecture does not discuss resources with state