

Retos Científicos y Tecnológicos en la Combinación de Grid y Semántica

Oscar Corcho
Universidad Politécnica de Madrid
March 3rd 2008



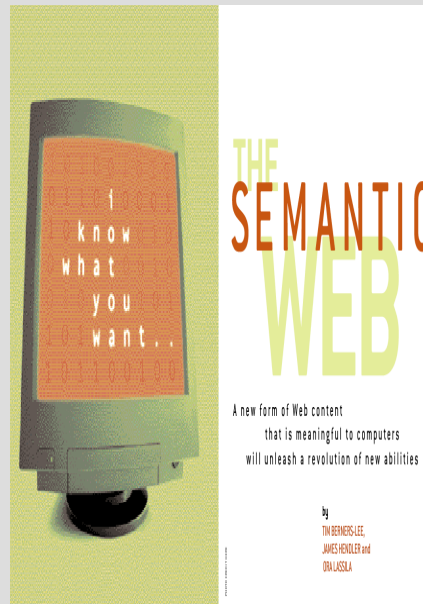
Overview

- **Ontologies and the Semantic Web** *(30 minutes)*
 - Introduction
 - What is the Semantic Web
 - Annotation, Integration, Inference
 - Semantic Web Technologies
 - RDF, RDF Schema and OWL
- **Grid and Semantic Grid: History and Case Studies** *(10 minutes)*
 - What is the Grid?
 - Semantic Grid History and Use Cases
- **Semantic-OGSA (S-OGSA)** *(20 minutes)*
 - S-OGSA Reference Model and Capabilities
 - S-OGSA Mechanisms and Interaction Patterns
 - A Sample Deployment of S-OGSA
- **Challenges** *(5 minutes)*
- **Credits**



What is the Semantic Web

- An extension of the current Web...
 - ... where **information and services** are given **well-defined** and **explicitly represented meaning**, ...
 - ... so that it can be **shared** and used by **humans and machines**, ...
 - ... better enabling them to work in cooperation
- How?
 - Promoting information exchange by **tagging web content** with machine processable descriptions of its meaning.
 - And **technologies** and **infrastructure** to do this

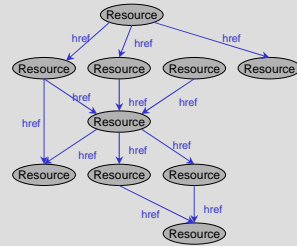


The Semantic Web Vision

- The Web was made possible through established **standards**
 - **TCP/IP** for transporting bits down a wire
 - **HTTP & HTML** for transporting and rendering hyperlinked text
 - **Applications** able to exploit this common infrastructure
 - Result is the WWW as we know it
 - **Generations**
 - **1st generation** web mostly handwritten HTML pages
 - **2nd generation** (current) web often machine generated/active
 - Both intended for direct human processing/interaction
 - In the **next generation** web, **resources** should be more accessible to automated processes
 - To be achieved via **semantic markup**
 - **Metadata** annotations that describe content/function
- } The Syntactic Web
- } The Semantic Web



Where we are Today: the *Syntactic* Web



- A place where computers do the presentation (easy) and people do the linking and interpreting (hard).
- Why not get computers to do more of the hard work?

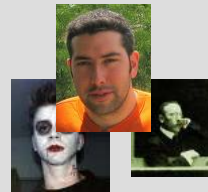


Hard Work using the *Syntactic* Web...

Find images of Oscar Corcho

...David Buján ...

... Josuka Díaz Labrador ...





What's the Problem?



- Typical web page markup consists of:
 - Rendering information (e.g., font size and colour)
 - Hyper-links to related content
- Semantic content is accessible to humans but not (easily) to computers...



Information we can see...

Universidad de Deusto

Historia y Misión
Plan estratégico
Directorio
Campus
Política Lingüística

Eventos... (en la universidad/fuera...?)

SemanaUD, Jornadas, ...

Noticias

Tipos de estudiantes

Nuevos
Actuales
Futuros

...



Need to Add "Semantics"

- External agreement on meaning of annotations
 - E.g., *Dublin Core* for annotation of library/bibliographic information
 - Agree on the meaning of a set of annotation tags
 - Problems with this approach
 - Inflexible
 - Limited number of things can be expressed
- Use **Ontologies** to specify meaning of annotations
 - Ontologies provide a vocabulary of terms
 - New terms can be formed by combining existing ones
 - "Conceptual Lego"
 - Meaning (**semantics**) of such terms is formally specified
 - Can also specify relationships between terms in multiple ontologies



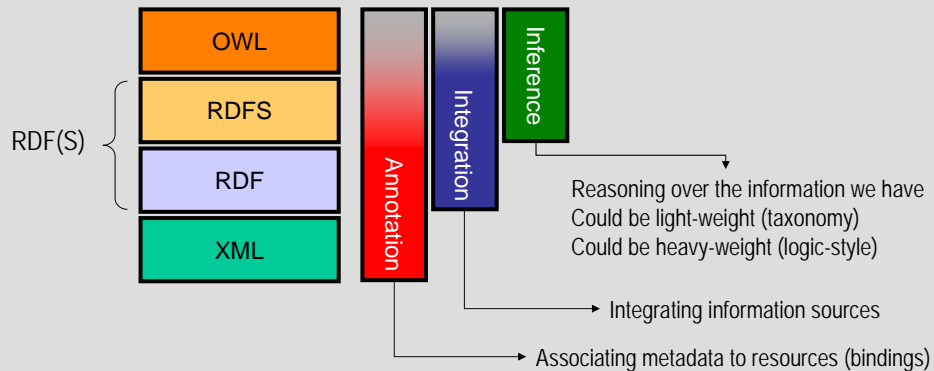
Ontology in Computer Science

- An ontology is an **engineering artifact**:
 - It is constituted by a specific **vocabulary** used to describe a certain reality, plus
 - a set of explicit **assumptions** regarding the intended meaning of the vocabulary.
 - Almost always including concepts and their classification
 - Almost always including properties between concepts
 - ~~Similar to an object-oriented model~~
- Thus, an ontology describes a formal specification of a certain domain:
 - Shared understanding of a domain of interest
 - Formal and machine manipulable model of a domain of interest



Ontology Languages

- Work on Semantic Web has concentrated on the definition of a collection or “stack” of languages.
 - Used to support the representation and use of metadata
 - Basic machinery that we can use to represent the extra semantic information needed for the Semantic Web



RDF

- RDF stands for Resource Description Framework
- It is a W3C Recommendation
 - <http://www.w3.org/RDF>
- RDF is a graphical formalism (+ XML syntax + semantics)
 - for representing metadata
 - for describing the semantics of information in a machine- accessible way
- Provides a simple data model based on triples.

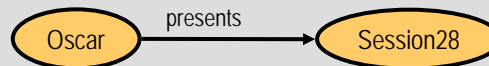


The RDF Data Model

- Statements are <subject, predicate, object> triples:

- <Oscar,presents,Session32>

- Can be represented as a graph:

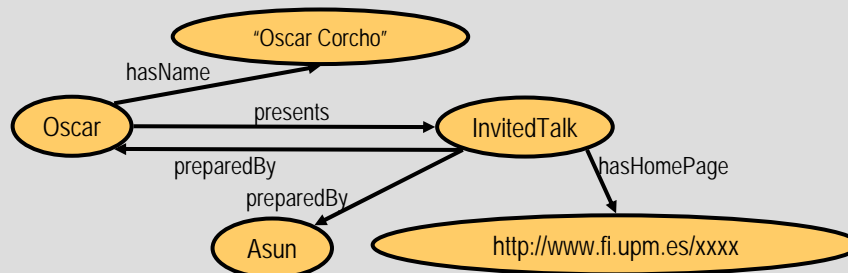


- Statements describe properties of **resources**
- A **resource** is any object that can be pointed to by a URI
 - The generic set of all names/addresses that are short strings that refer to resources
 - a document, a picture, a paragraph on the Web, <http://www.cs.man.ac.uk/~ocorcho/index.html>, a book in the library, a real person, isbn://0141184280
 - *Do not mistake them for Grid resources, though they could be the same, as we will see later in this talk!!*
- Properties themselves are also resources (URIs)



Linking Statements

- The subject of one statement can be the object of another
- Such collections of statements form a directed, labeled graph



- The object of a triple can also be a "literal" (a string)



RDF Syntax

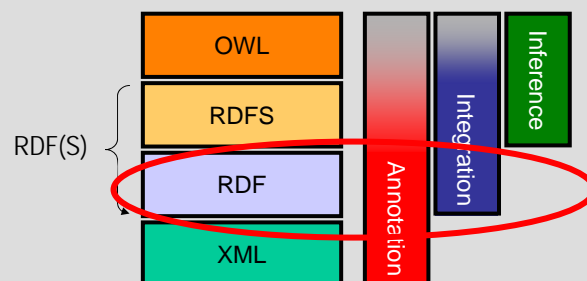
- RDF has an XML syntax that has a specific meaning:
- Every **Description** element describes a resource
- Every attribute or nested element inside a **Description** is a **property** of that **Resource**
- We can refer to resources by URIs

```
<rdf:Description rdf:about="some.uri/person#ocorcho">
  <o:presents rdf:resource="some.uri/session#InvitedTalk"/>
  <o:hasName rdf:datatype="xsd:string">Oscar Corcho</o:hasName>
</rdf:Description>
<rdf:Description rdf:about="some.uri/session#InvitedTalk">
  <o:hasHomePage>http://www.fi.upm.es/xxxx</o:hasHomePage>
  <o:preparedBy rdf:resource="some.uri/person#ocorcho">
  <o:preparedBy rdf:resource="some.uri/person#asun">
</rdf:Description>
```



What does RDF give us?

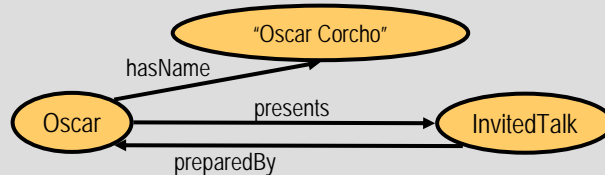
- Single (simple) data model.
- Syntactic consistency between names (URIs).
- A mechanism for **annotating** data and resources.
- Low level **integration** of data.



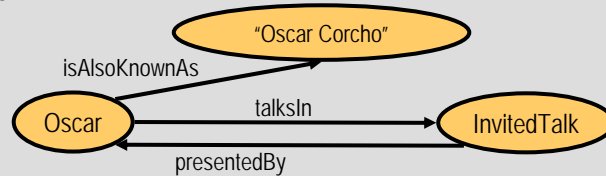


What doesn't RDF give us?

- RDF does not give any special meaning to vocabulary
 - Such as `subClassOf` or `type` (supporting OO-style modelling)
- So, what's the difference between this graph...



- ... and this one?



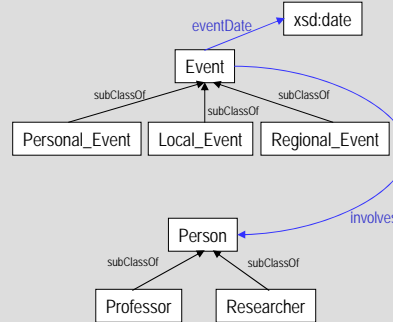
RDFS: RDF Schema

- RDF Schema is another W3C Recommendation
 - <http://www.w3.org/TR/rdf-schema/>
- It extends RDF with a `schema vocabulary` that allows you to define basic vocabulary terms and the relations between those terms
 - `Class`, `type`, `subClassOf`,
 - `Property`, `subPropertyOf`, `range`, `domain`
 - it gives “extra meaning” to particular RDF predicates and resources
 - this “extra meaning”, or `semantics`, specifies how a term should be interpreted
- The combination of RDF and RDF Schema is normally known as RDF(S)

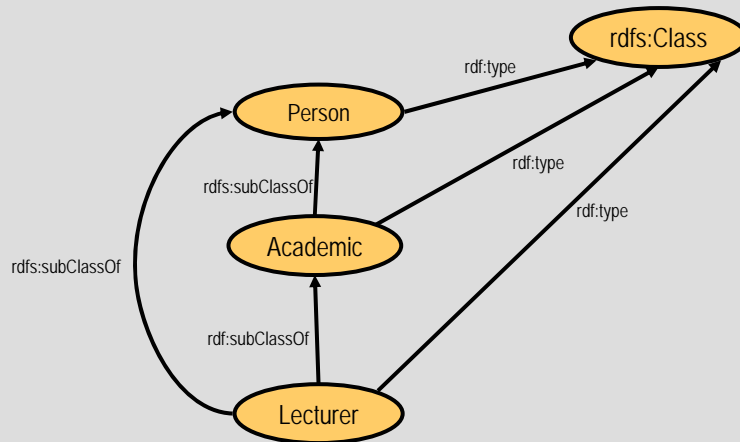


RDFS simple example

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xml:base="http://www.ontogrid.net/StickyNote#"
  xmlns="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <rdf:Class rdf:ID="Event">
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  </rdf:Class>
  <rdf:Class rdf:ID="Local_Event">
    <rdfs:subClassOf rdf:resource="#Event"/>
  </rdf:Class>
  <rdf:Class rdf:ID="Regional_Event">
    <rdfs:subClassOf rdf:resource="#Event"/>
  </rdf:Class>
  <rdf:Class rdf:ID="Personal_Event">
    <rdfs:subClassOf rdf:resource="#Event"/>
  </rdf:Class>
  <rdf:Class rdf:ID="Person">
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  </rdf:Class>
  <rdf:Class rdf:ID="Professor">
    <rdfs:subClassOf rdf:resource="#Person"/>
  </rdf:Class>
  <rdf:Class rdf:ID="Researcher">
    <rdfs:subClassOf rdf:resource="#Person"/>
  </rdf:Class>
  <rdf:Property rdf:ID="involves">
    <rdfs:domain rdf:resource="#Personal_Event"/>
    <rdfs:range rdf:resource="#Person"/>
  </rdf:Property>
  <rdf:Property rdf:ID="eventDate">
    <rdfs:domain rdf:resource="#Event"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#date"/>
  </rdf:Property>
</rdf:RDF>
```

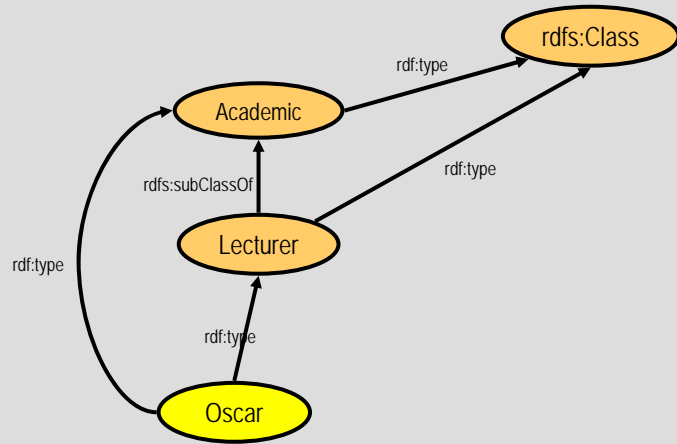


RDF(S) Inference

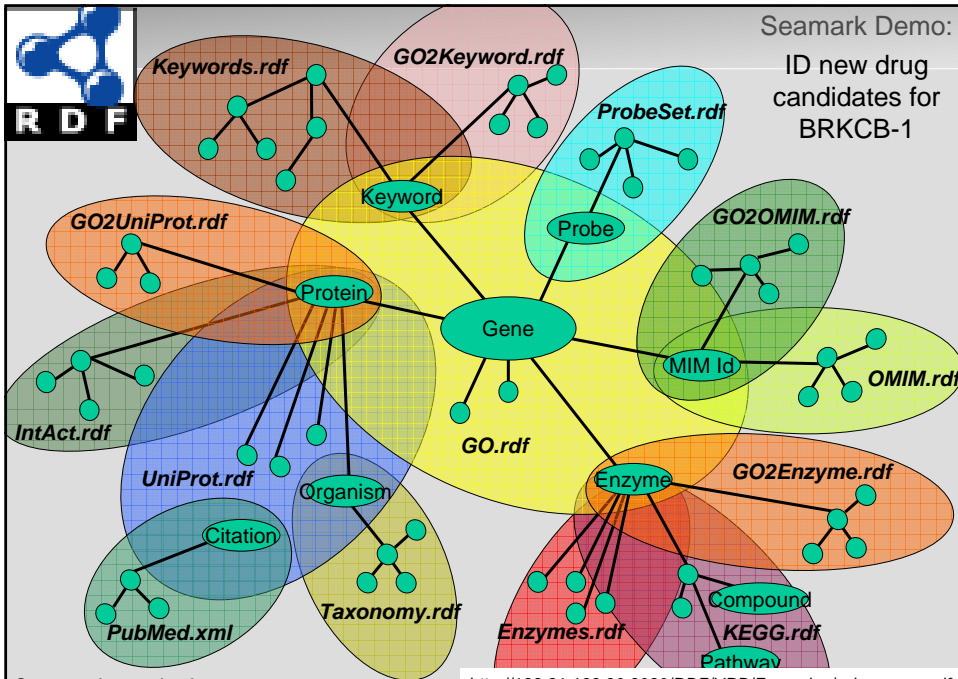




RDF(S) Inference



Seamark Demo:
ID new drug
candidates for
BRKCB-1



Courtesy Joanne Luciano

http://139.91.183.30:9090/RDF/VRP/Examples/schema_go.rdf
<http://139.91.183.30:9090/RDF/VRP/Examples/go.rdf>

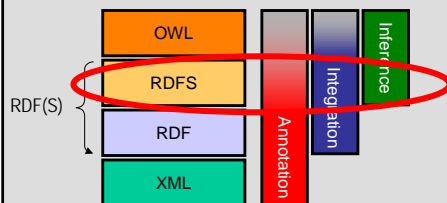
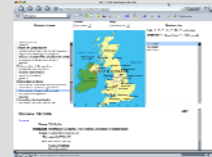


What does RDFS give us?

- Ability to use simple schema/vocabularies to describe our resources
- **Consistent** vocabulary use and **sharing**
- Simple **inference**
- **Query mechanisms: SPARQL, SeRQL, RDQL, ...**
 - `SELECT N FROM {N} rdf:type {sti:Event}`
`USING NAMESPACE sti=<http://www.ontogrid.net/StickyNote#>`

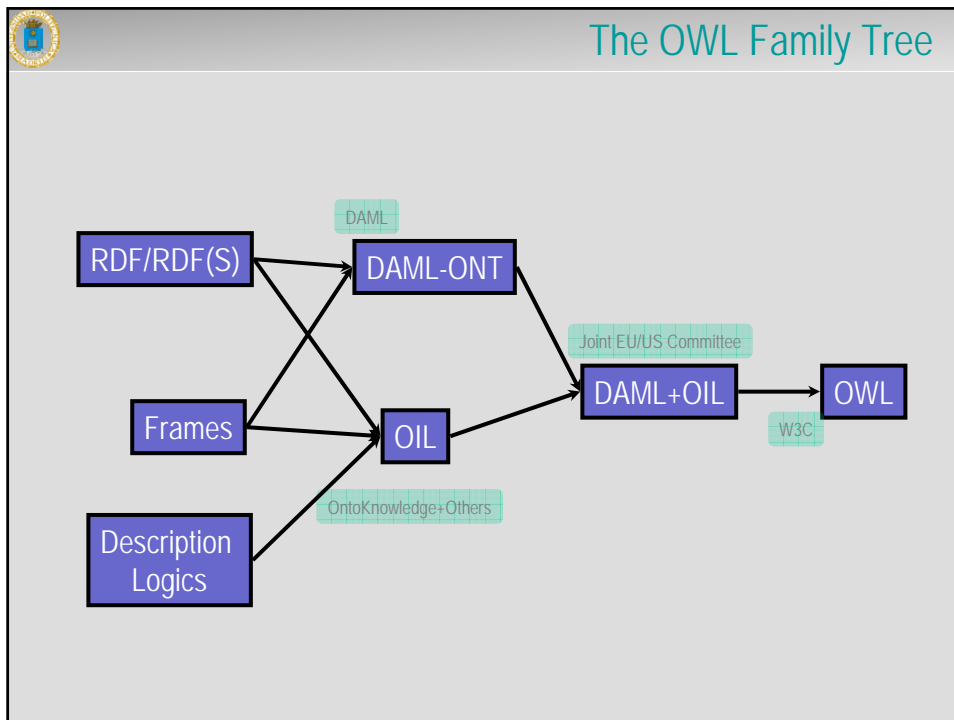
Examples

- CS AktiveSpace
 - Lightweight schema to integrate data from University sites
- myGrid
 - Service descriptions for e-Science



What doesn't RDFS give us?

- RDFS is **too weak** to describe resources in sufficient detail
 - No **localised range and domain** constraints
 - Can't say that the range of `hasEducationalMaterial` is `Slides` when applied to `TheoreticalSession` and `Code` when applied to `HandsonSession`
 - `TheoreticalSession hasEducationalMaterial Slides`
 - `HandsonSession hasEducationalMaterial Code`
 - No **existence/cardinality** constraints
 - Can't say:
 - `Sessions` must have some `EducationalMaterial`
 - `Sessions` have at least one `Presenter`
 - No **transitive, inverse or symmetrical** properties
 - Can't say that `presents` is the inverse property of `isPresentedBy`

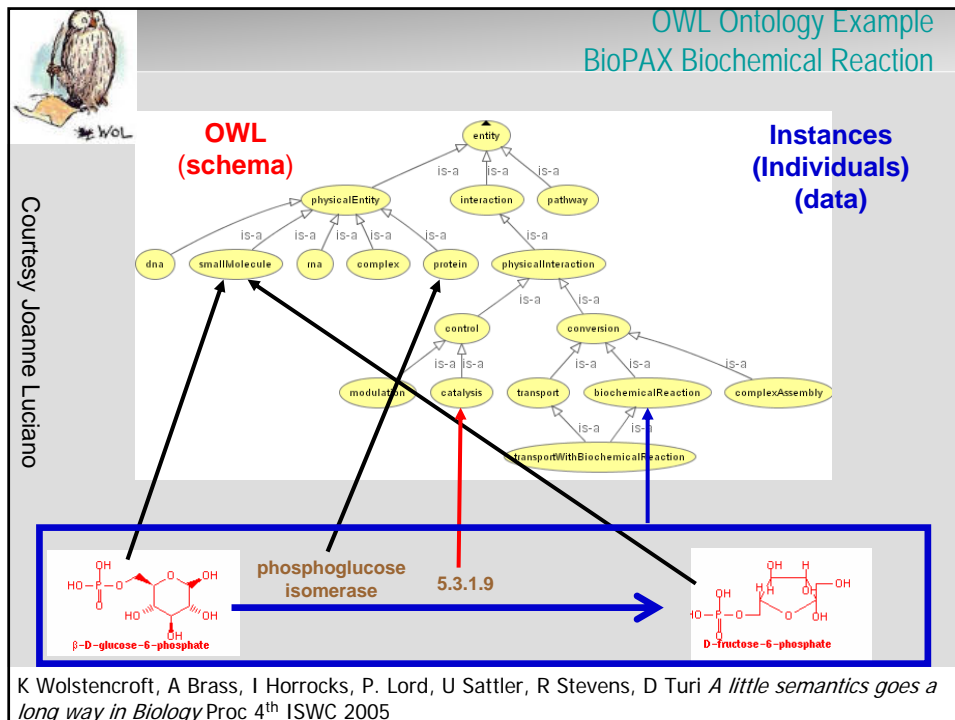


- OWL
- W3C Recommendation (February 2004)
 - A family of Languages
 - OWL Full
 - OWL DL
 - OWL Lite
 - Formal semantics
 - Description Logics (DL/Lite)
 - Relationship with RDF



OWL Basics (on top of RDF and RDFS)

- Set of constructors for **concept expressions**
 - Booleans: and/or/not
 - A Session is a TheoreticalSession or a HandsonSession
 - Slides are not the same as Code
 - Quantification: some/all
 - Sessions must have some *EducationalMaterial*
 - Sessions can only have *Presenters* that have developed *Grid applications* or *Grid middleware*
- Axioms for expressing **constraints**
 - Necessary and Sufficient conditions on classes
 - A Session that *hasEducationalMaterial Code* is a *HandsonSession*.
 - Disjointness
 - *TheoreticalSessions* are disjoint with *HandsonSessions*
 - Property characteristics: transitivity, inverse





OWL Ontology Example. BioPAX ontology

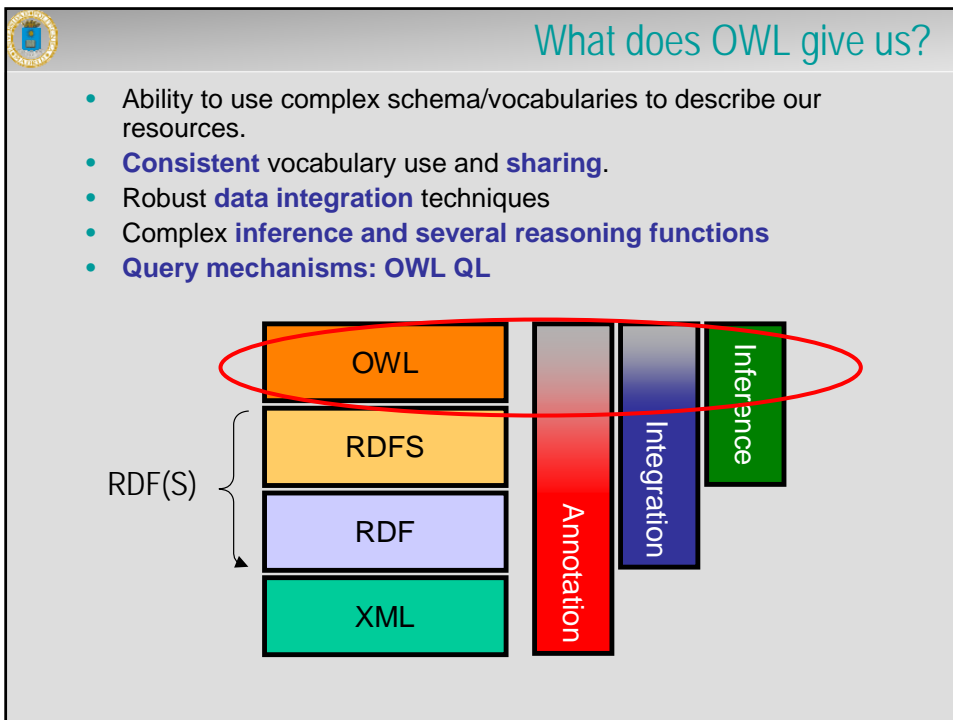
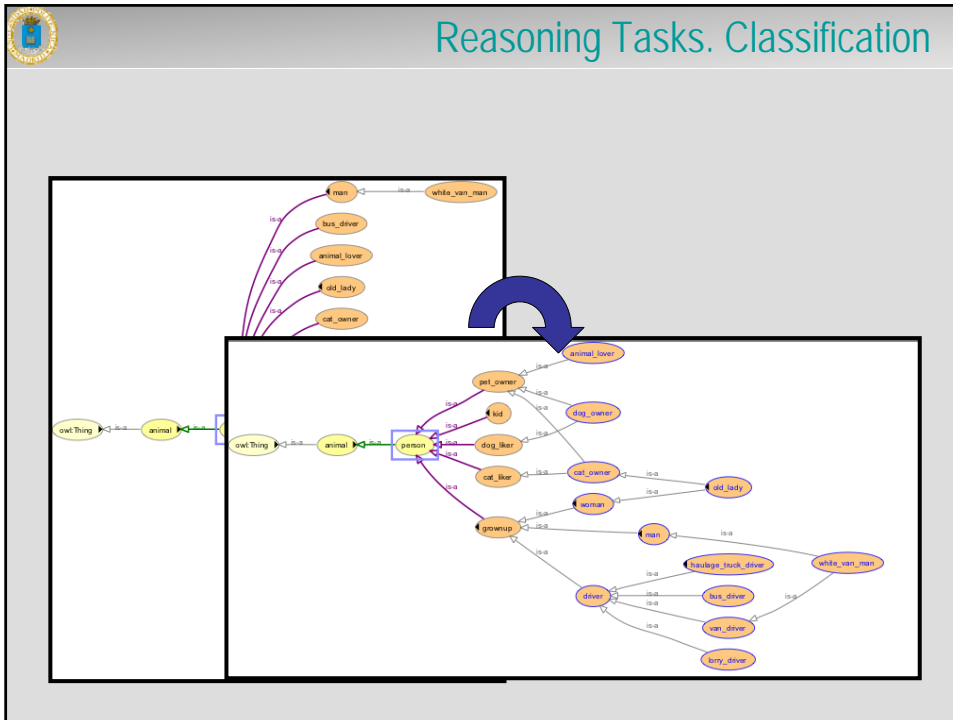
- <http://www.biopax.org/release/biopax-level2.owl>

The screenshot shows the Protege software interface for editing the BioPAX ontology. The left pane displays a class hierarchy with 'physicalEntity' selected. The central pane shows the definition of 'physicalEntity' as a class of entities with a physical structure. The right pane lists various properties associated with the class, such as 'NAME', 'SHORT_NAME', 'ABILITY', 'COMMENT', 'DATA_SOURCE', 'FINDINGS', and 'REF'.



Reasoning Tasks

- OWL DL based on a well understood Description Logic ($\text{SHOIN}(\mathcal{D}_n)$)
 - Formal properties well understood (complexity, decidability)
 - Known reasoning algorithms
 - Implemented systems (highly optimised)
- Because of this, we can reason about OWL ontologies
 - Subsumption reasoning
 - Allows us to infer when one class is a subclass of another
 - Can then build concept hierarchies representing the taxonomy.
 - This is classification of **classes**.
 - Satisfiability reasoning
 - Tells us when a concept is **unsatisfiable**
 - i.e. when it is impossible to have instances of the class.
 - Allows us to check whether our model is **consistent**.
 - Instance Retrieval/Instantiation
 - What are the **instances** of a particular class **C**?
 - What are the classes that **x** is an instance of?





Overview

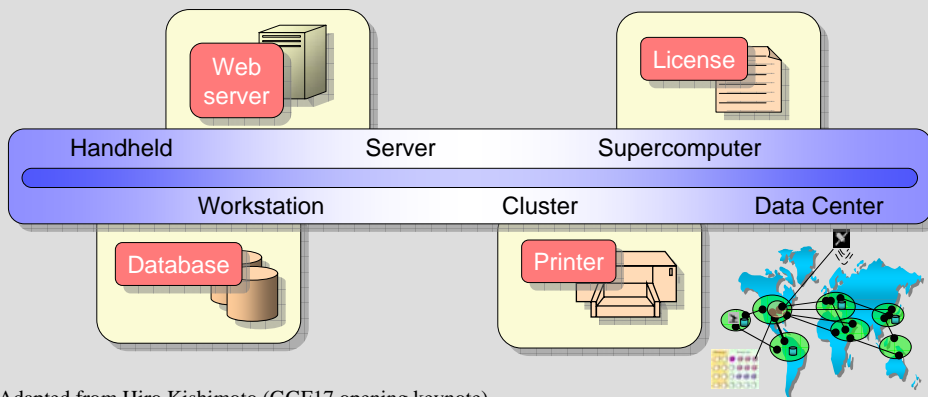
- Ontologies and the Semantic Web (30 minutes)
 - Introduction
 - What is the Semantic Web
 - Annotation, Integration, Inference
 - Semantic Web Technologies
 - RDF, RDF Schema and OWL
- Grid and Semantic Grid: History and Case Studies (10 minutes)
 - What is the Grid?
 - Semantic Grid History and Use Cases
- Semantic-OGSA (S-OGSA) (20 minutes)
 - S-OGSA Reference Model and Capabilities
 - S-OGSA Mechanisms and Interaction Patterns
 - A Sample Deployment of S-OGSA
- Challenges (5 minutes)
- Credits



What is a Grid?

A grid is a system consisting of

- Distributed but connected resources and
- Software and/or hardware that provides and manages logically seamless access to those resources to meet desired objectives
- Infrastructure that will enable "coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations"



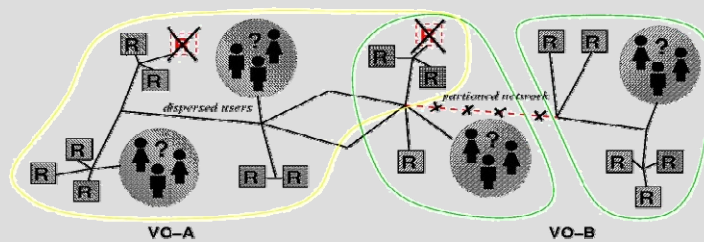
Adapted from Hiro Kishimoto (GGF17 opening keynote)



Virtual Organizations

- Dynamic confederations organized around common goals
 - Diverse membership & capabilities
 - People, compute resources, data resources, etc.
 - Diverse geographic distribution
- Sharing is well-controlled
- Minimum knowledge about physical characteristics of resources
- Construction of higher level capabilities via composition of existing ones similar to SOA

o



From <http://www.globus.org>



Grid & Related Paradigms

Distributed Computing

- Loosely coupled
- Heterogeneous
- Single Administration

Grid Computing

- Large scale
- Cross-organizational
- Geographical distribution
- Distributed Management

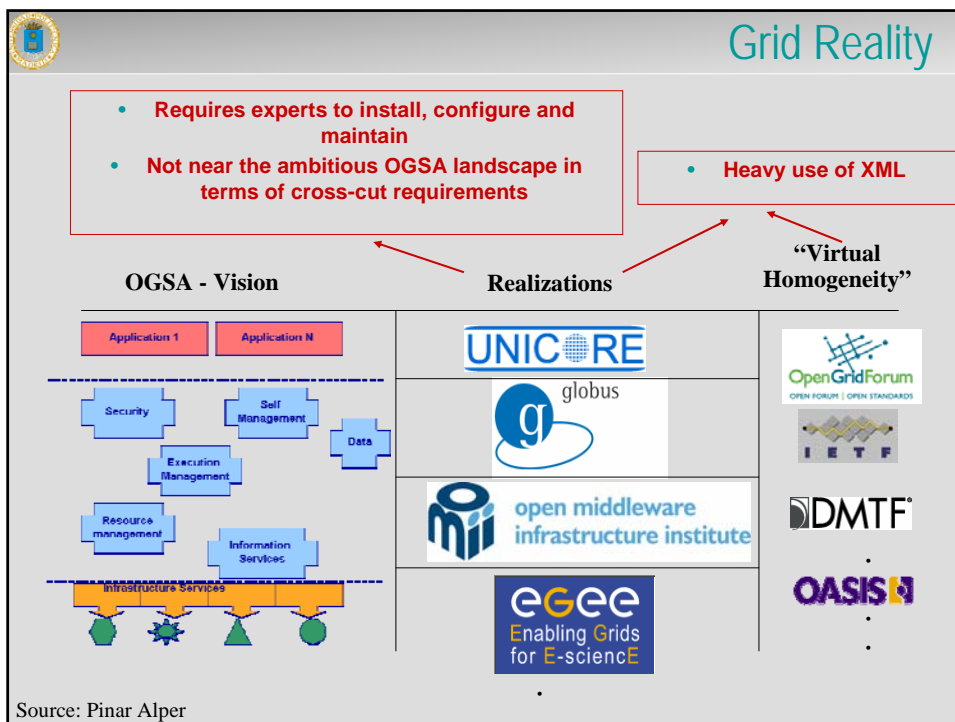
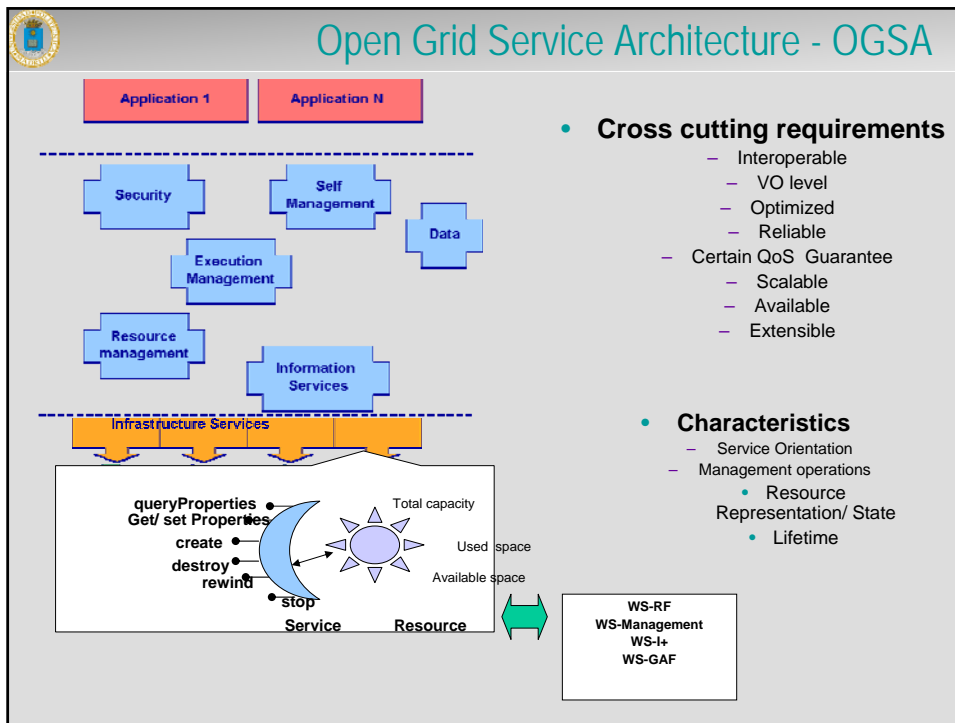
Utility Computing

- Computing "services"
- No knowledge of provider
- Enabled by grid technology

Cluster

- Tightly coupled
- Homogeneous
- Cooperative working

Source: Hiro Kishimoto (GGF17 opening keynote)





The Semantic Grid

"The Semantic Grid is an extension of the current Grid in which information and services are given **well-defined and explicitly represented meaning**, so that it can be **shared and used by humans and machines**, better enabling computers and people to work in cooperation" *D. De Roure, et. al*

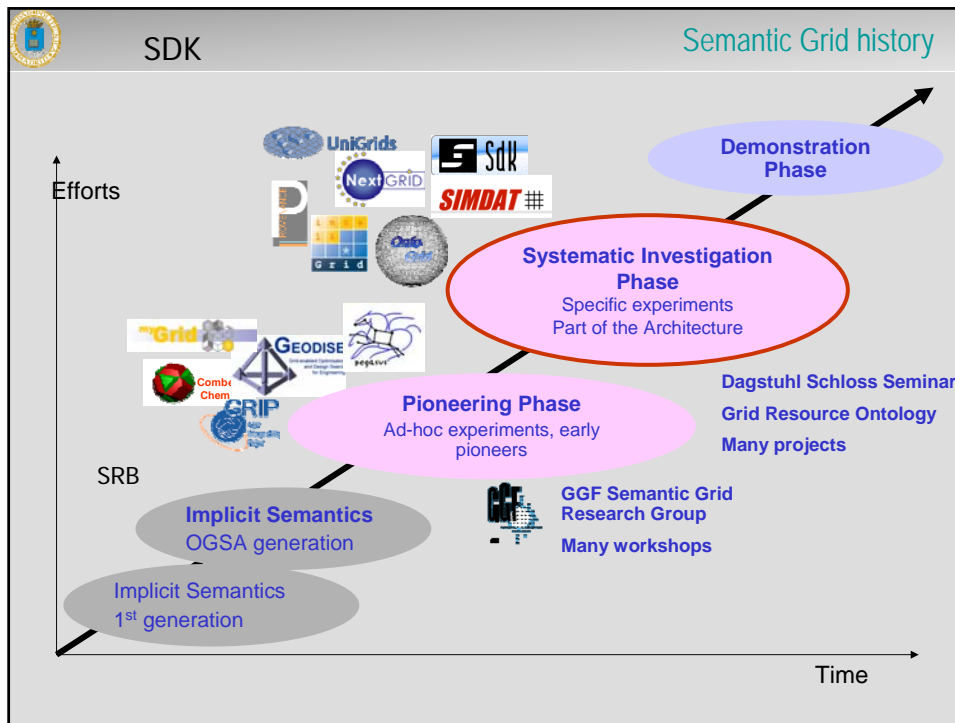
Semantics in and on the Grid

- Web Sites
 - www.semanticgrid.org
- GGF Semantic Grid Research Group (SEM-RG)
 - Mailing List: sem-grd@gridforum.org



Motivation. Metadata Matters

- Particularly for the following activities:
 - Information provision and resource discovery
 - Data integration
 - Provenance
 - Systems Configuration
 - Policy representation and reconciliation
- Using:
 - Open, **flexible** and **extensible** self describing schemas that don't have to be nailed down
 - "Let's describe my data set, or the output format of this tool"
 - Lightweight schemas
 - Decoupled, interoperable systems, which **resist to syntactic changes**
 - Open world
 - "This metadata is no longer valid because..."
 - **Data integration** across different data models (e.g. RDF)
 - Like policy or resource models
 - Formalization & Reasoning support



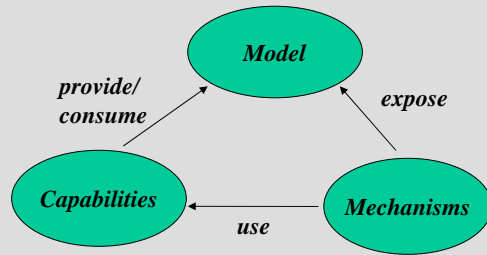
Overview

• Ontologies and the Semantic Web	(30 minutes)
– Introduction	
– What is the Semantic Web	
• Annotation, Integration, Inference	
– Semantic Web Technologies	
• RDF, RDF Schema and OWL	
• Grid and Semantic Grid: History and Case Studies	(10 minutes)
– What is the Grid?	
– Semantic Grid History and Use Cases	
• Semantic-OGSA (S-OGSA)	(20 minutes)
– S-OGSA Reference Model and Capabilities	
– S-OGSA Mechanisms and Interaction Patterns	
– A Sample Deployment of S-OGSA	
• Challenges	(5 minutes)
• Credits	

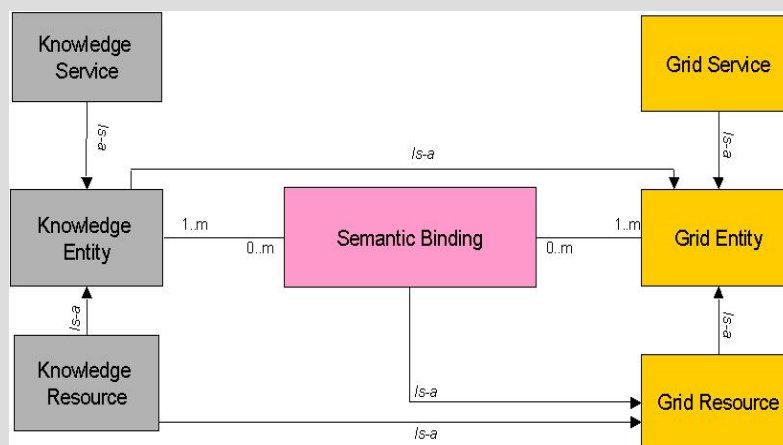


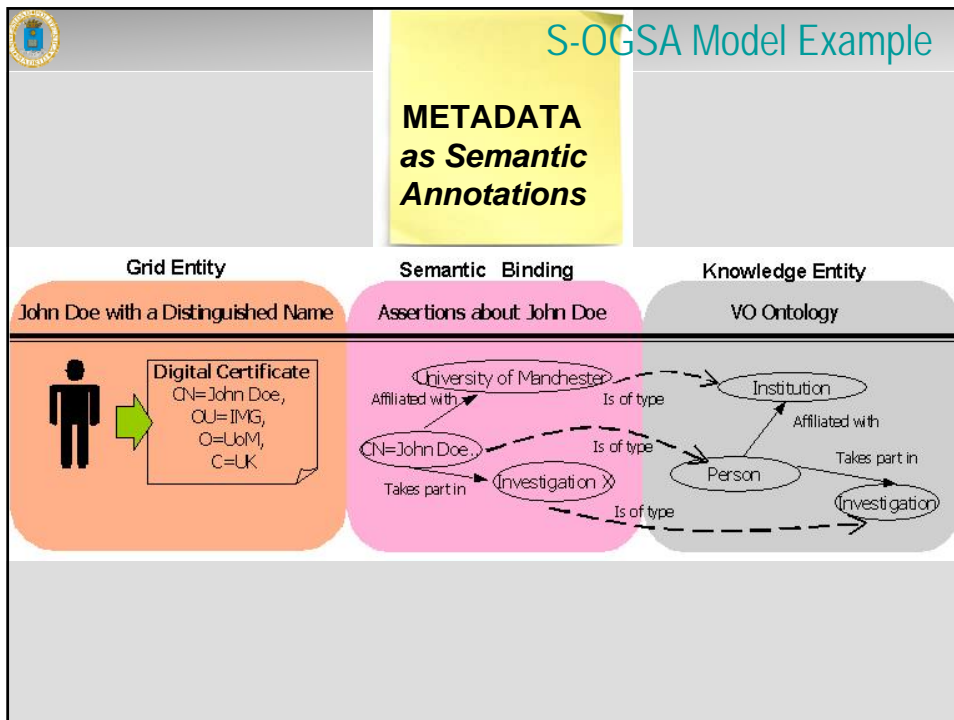
Introduction. Semantic-OGSA

- Semantic-OGSA (S-OGSA) is...
 - A Semantic Grid architecture
 - A low-impact extension of OGSA
 - Mixed ecosystem of Grid and Semantic Grid services
 - Services **ignorant** of semantics
 - Services **aware** of semantics but **unable** to process them
 - Services **aware** of semantics and **able** to process (part of) them
 - Everything is OGSA compliant
 - Defined by
 - Information model
 - New entities
 - Capabilities
 - New functionalities
 - Mechanisms
 - How it is delivered



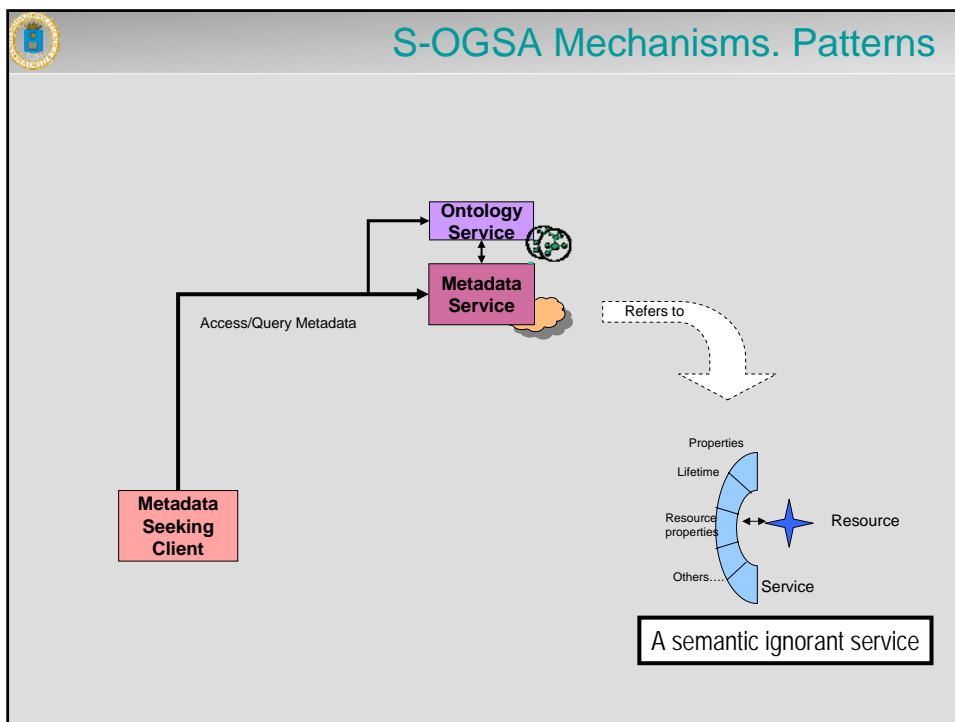
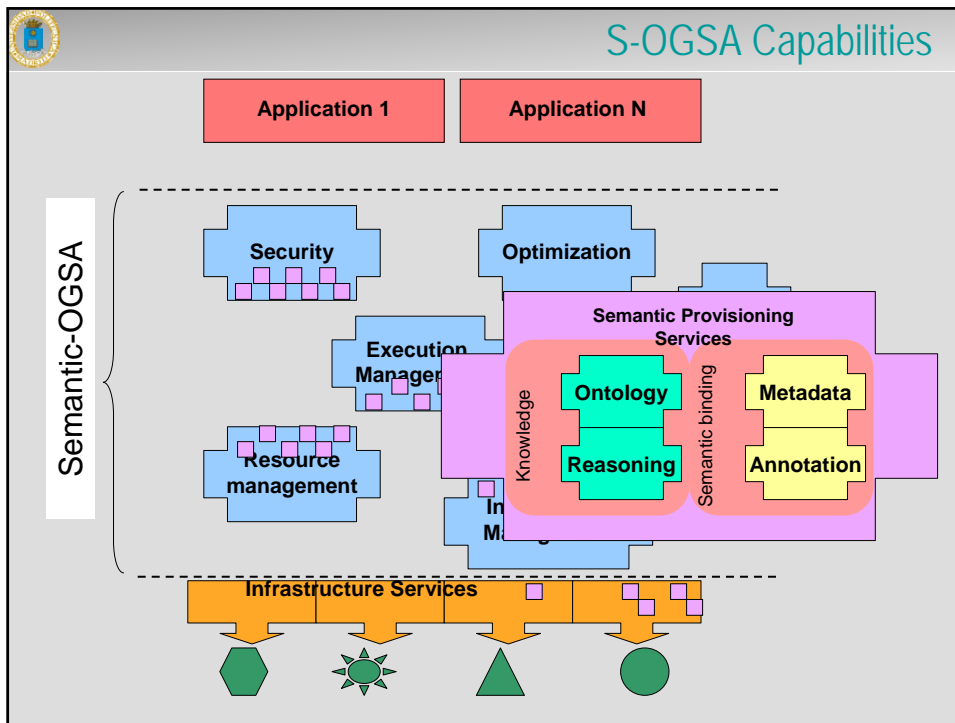
S-OGSA Model

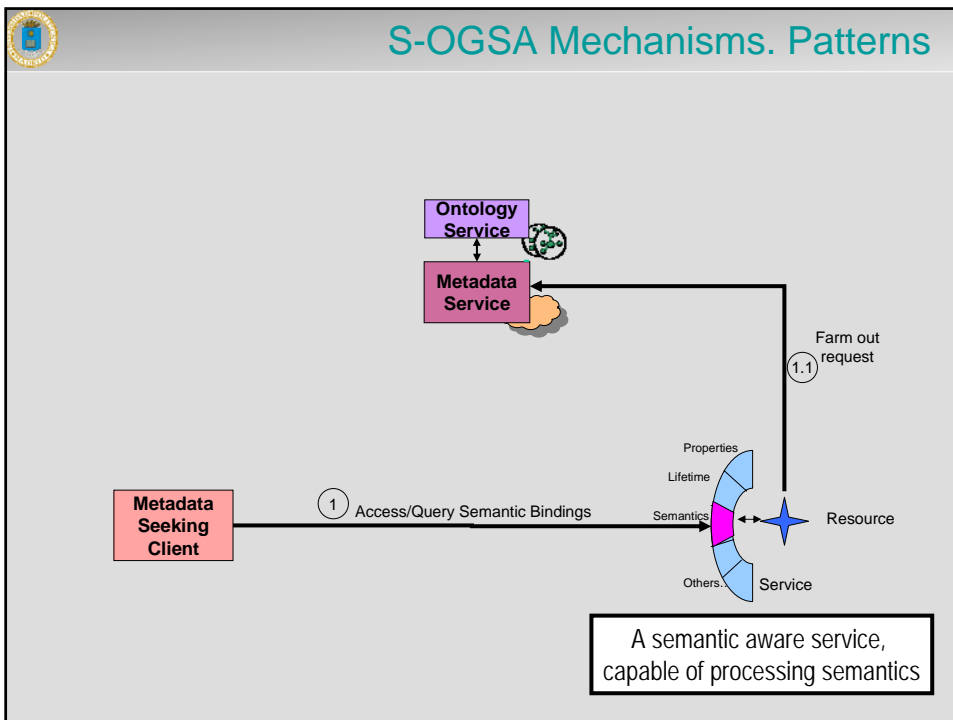
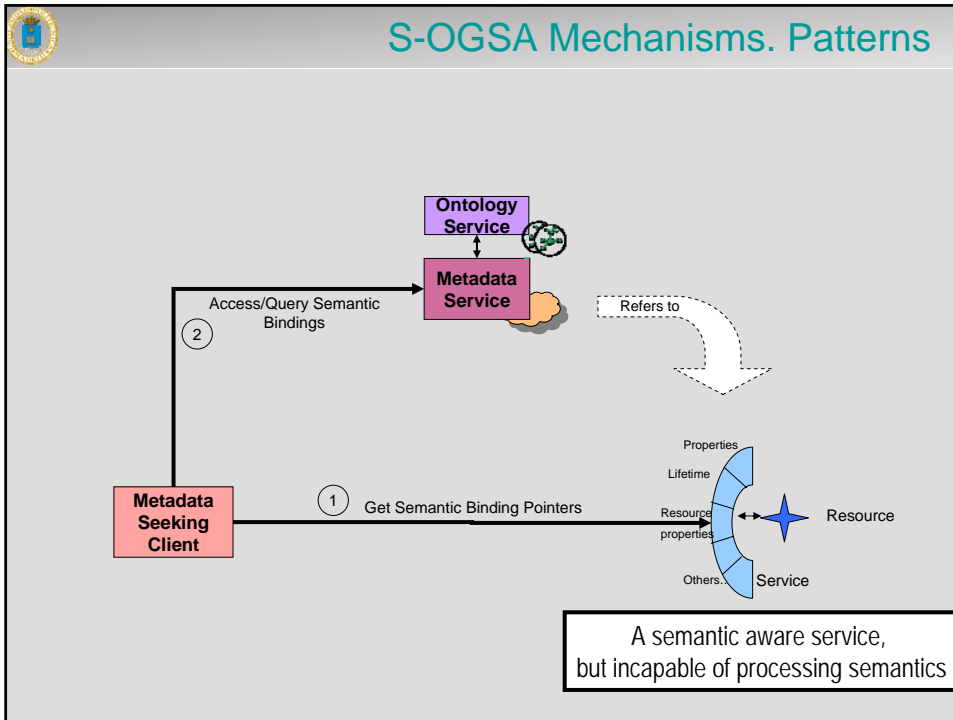


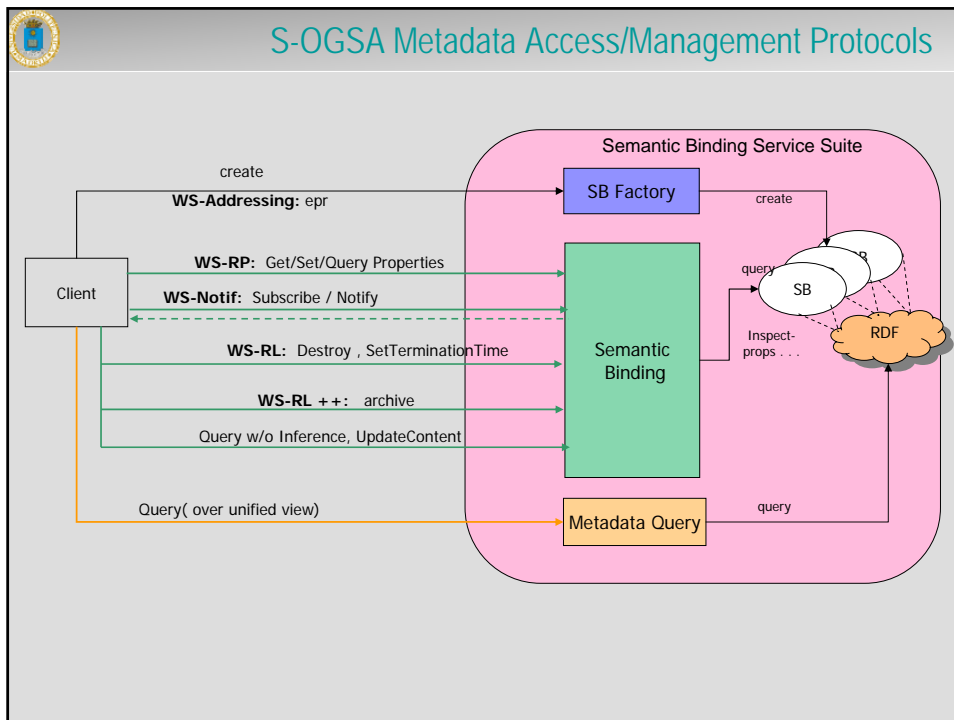


- S-OGSA Model. Grid Entities
- We can attach Semantic Bindings to anything
 - People, meetings, discussions, conference talks
 - Scientific publications, recommendations, quality comments
 - Events, notifications, logs
 - Services and resources
 - Schemas and catalogue entries
 - Models, codes, builds, workflows,
 - Data files and data streams
 - Sensors and sensor data ...

 - To make it more useful, we should agree on
 - Controlled vocabularies / Ontologies
 - Resource description models
 - Grid Resource Ontologies (*work in progress*)
 - Application domain vocabularies



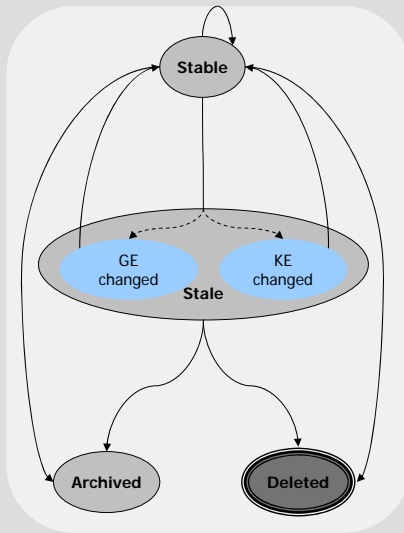




- ### Semantic Binding Service. Lifetime Specification
- What happens if...
 - ...any or all of the Grid entities it refers to disappears?
 - Instrument and planning files for satellites do not disappear
 - Insurance contracts, cars, repair companies, etc., may disappear
 - ...the Knowledge entities disappear or evolve?
 - Ontologies may change
 - ... a SB is no longer available (its content is not useful any more)?
 - Damage claims: add witness reports, improve info about location, create new hypothesis...
 - When do/should SBs become invalid? How often should this be checked?
 - What is the status of the content of a SB (e.g., content checked, stable, unchecked, etc.)?
 - Is a SB always active or can it be archived after a period of time?
 - Satellite data that is not used after some time



Semantic Binding Service. WS-SBResourceLifetime



- Lifetime specification based on WS-ResourceLifetime
- Extension with
 - Resource properties (state)
 - Updates
 - Archive
 - Notifications



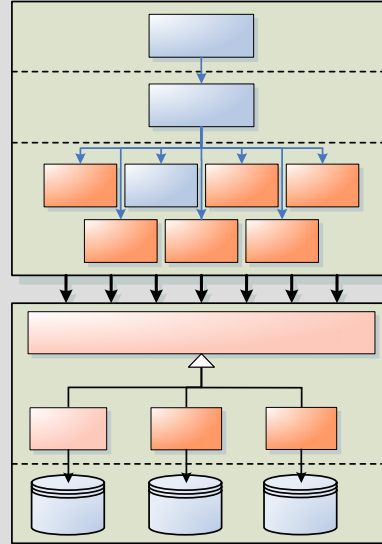
WS-DAIOnt-RDF(S). Specification

- Web Services Data Access and Integration – The Ontology Realization for RDF(S) Access
- Ontologies (and their components) are Grid resources
 - State and lifetime
 - Access control
- Specification of ontology access services
 - Interfaces to access remote RDF(S) ontologies
 - Using model-based access functionalities
 - Using RDF(S) querying languages
 - Based on Web and Grid Service (OASIS, W3C, OGF) standards
 - WS-DAI, WS-RF, WS-Addressing
 - Based on Semantic Web standards (W3C):
 - RDF, RDFS, SPARQL



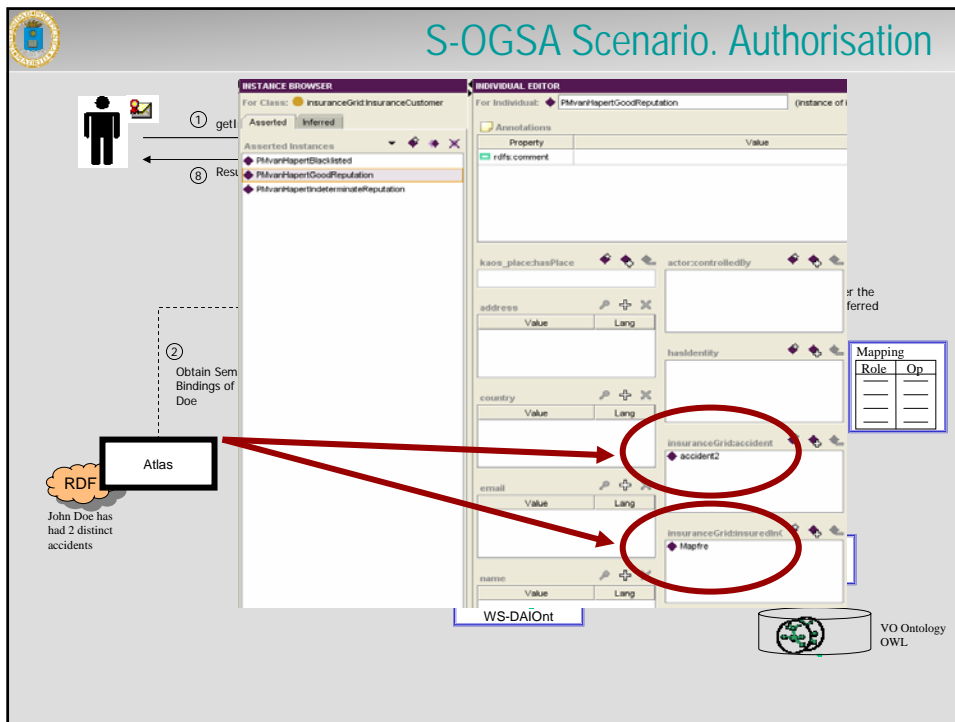
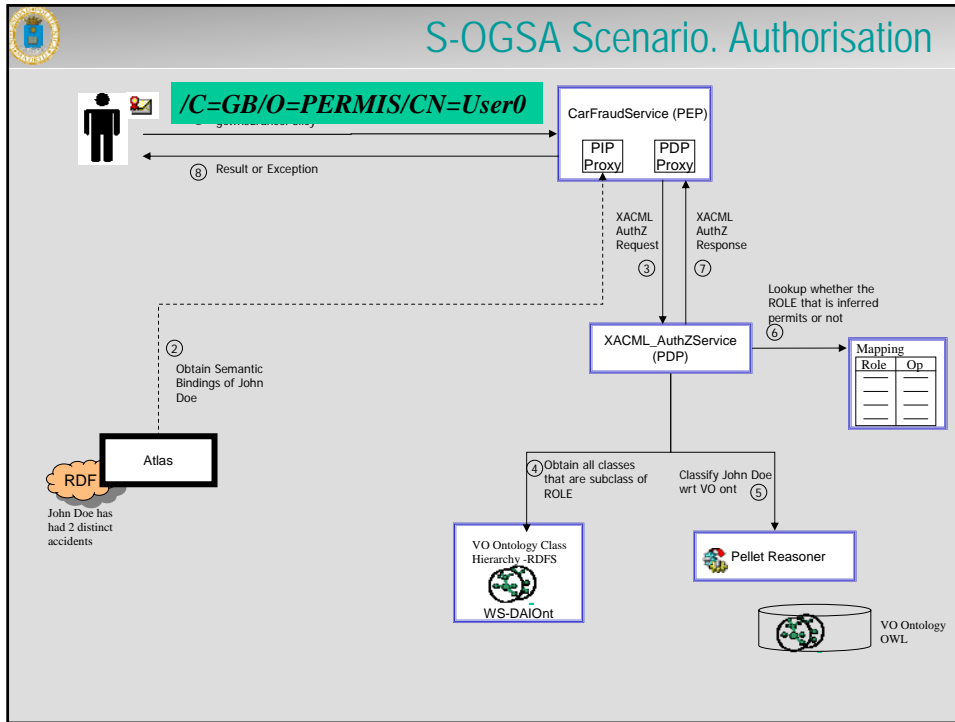
WS-DAIOnt-RDF(S). Implementation

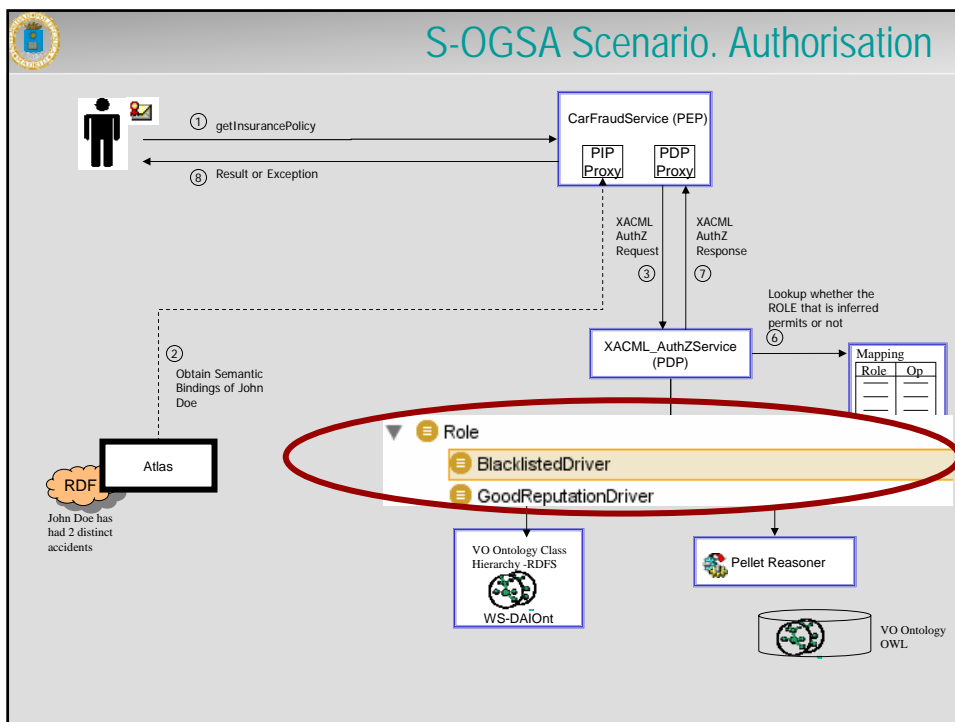
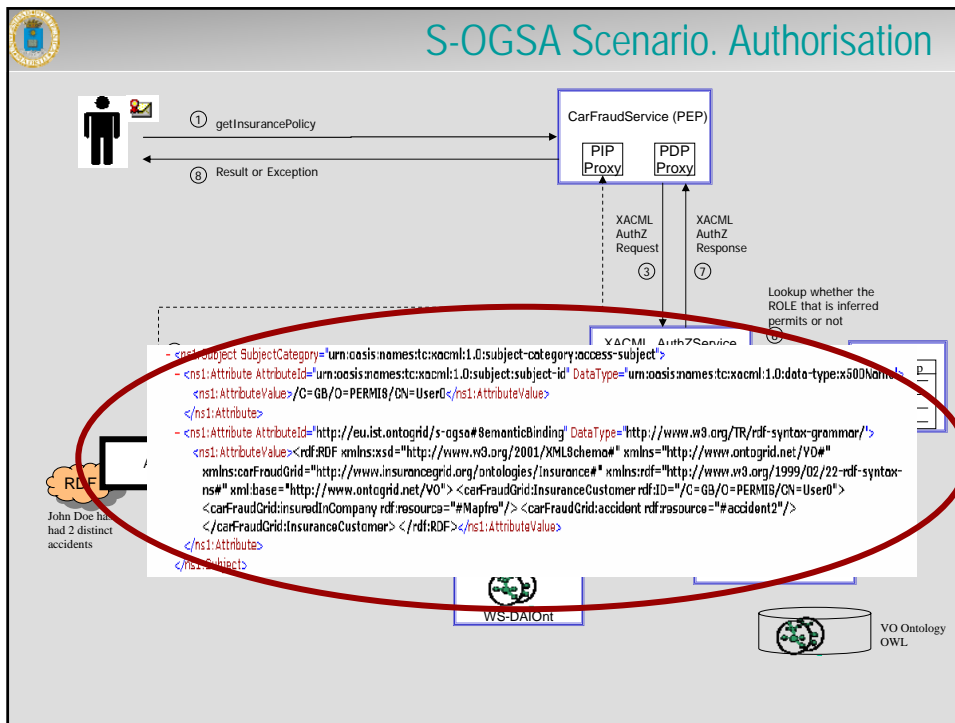
- Two-tier architecture:
 - Web Service tier, different layers according to access granularity
 - Upper layer: management of multiple repositories
 - Intermediate layer: management of a single repository
 - Lower layer: management of knowledge elements of a given repository
 - RDF(S) access tier:
 - Abstracts the interaction with specific RDF(S) storages

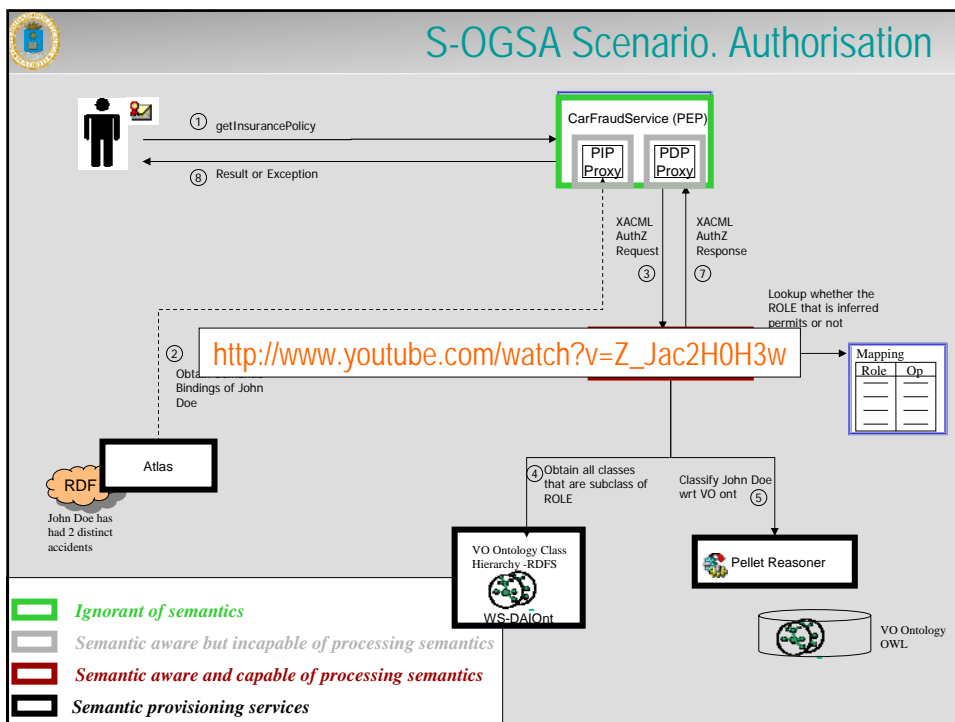
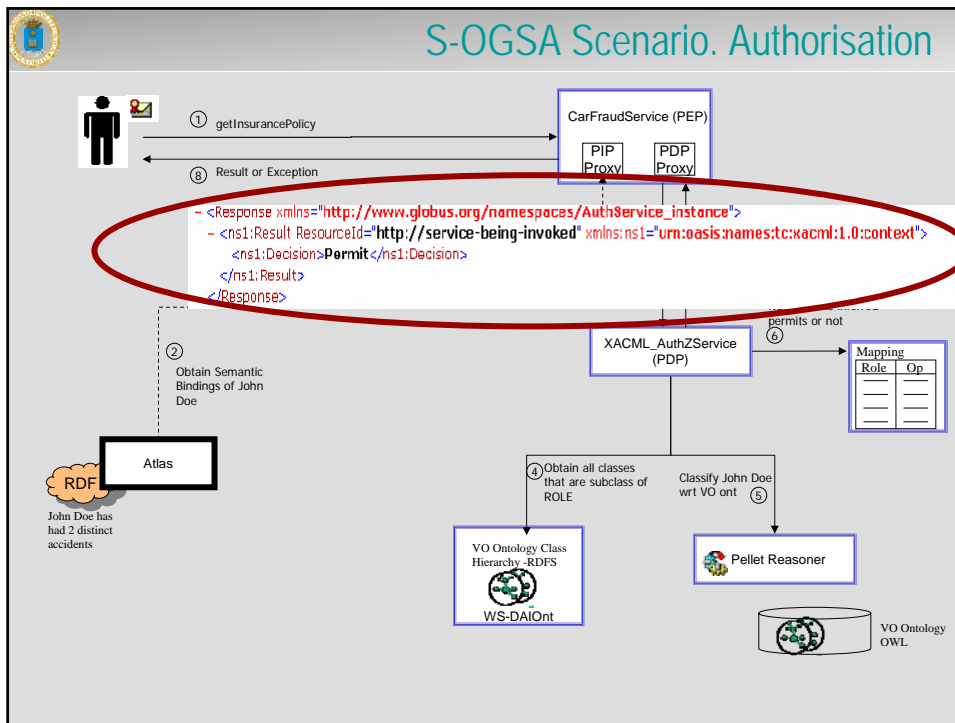


A simple Authorisation Scenario

- A role-based Access Control Scenario in the insurance domain.
- What?
 - Role based **Access Control Policy** is:
 - “Good Reputation Drivers are allowed to ask for an insurance policy. Bad Reputation ones are not.”
- How?
 - VO **ontology** based on
 - KaOS ontologies (Actors, Groups and Actions)
 - **Role definitions**
 - Extend ontology with domain-specific classes and properties
 - Define roles wrt these extensions
 - E.g., a blacklistedDriver is a driver that has had at least 3 accident claims in the past
 - E.g., a goodReputationDriver is a driver that has been insured at least by one trusted company and that has had at most 2 accident claims
 - The **Access Control Function** uses an OWL classifier to obtain roles of a Subject.









Overview

- Ontologies and the Semantic Web *(30 minutes)*
 - Introduction
 - What is the Semantic Web
 - Annotation, Integration, Inference
 - Semantic Web Technologies
 - RDF, RDF Schema and OWL
- Grid and Semantic Grid: History and Case Studies *(10 minutes)*
 - What is the Grid?
 - Semantic Grid History and Use Cases
- Semantic-OGSA (S-OGSA) *(20 minutes)*
 - S-OGSA Reference Model and Capabilities
 - S-OGSA Mechanisms and Interaction Patterns
 - A Sample Deployment of S-OGSA
- **Challenges** *(5 minutes)*
- Credits



Some challenges

- Scientific
 - Distributed reasoning engines for RDF Schema and OWL
 - WS-DAI-OWL specifications
 - QoS analysis of Semantic Grid applications
 - Ontology-based database integration (large scale sensor networks, large databases, etc.)
- Technological
 - Easy-to-deploy Semantic Grid testbed (in Globus, gLite, etc.)
 - Reasoning support implementation
 - More WS-DAI-RDF(S) implementations
 - WS-DAI-OWL implementations
 - Application of security (AuthZ) to knowledge models
 - Virtualisation of ontology access services (database and file system replication, state management, etc.)



Credits

- This tutorial is based on contributions from many authors. I hope to acknowledge all of them...
 - Sean Bechhofer, Carole Goble and David de Roure
 - Section “Ontologies and the Semantic Web”, based on Semantic Grid 101 presented at GGF16 in February 2006
 - The OntoGrid team @ Manchester: Pinar Alper, Ioannis Kotsiopoulos, Paolo Missier, Sean Bechhofer, Carole Goble
 - S-OGSA work
 - Many others whose names appear on the slides
- This tutorial has been funded by the European Commission, under the projects OntoGrid and RSSGRID
- Creative Commons License
- Questions regarding this tutorial should be directed to
 - Oscar Corcho: ocorcho@fi.upm.es



More information

- Publications
 - An overview of S-OGSA: a Reference Semantic Grid Architecture. Corcho O, Alper P, Kotsiopoulos I, Missier P, Bechhofer S, Goble C. *Journal of Web Semantics* 4(2):102-115. June 2006
 - Requirements and Services for Metadata Management. Missier P, Alper P, Corcho O, Dunlop I, Goble C. *IEEE Internet Computing*, special issue on Semantic-based Knowledge Management. 11(5):16-24. September-October 2007
 - Understanding Semantic-Aware Grid Middleware for e-Science. Alper P, Corcho O, Goble CA. *Journal of Computing and Informatics*. January 2008
- Source code
 - <http://www.ontogrid.net/>, For Downloading Distributions
 - Access to CVS
 - Connection type: pserver
 - user: ontogrid
 - password: not needed
 - Host: rpc262.cs.man.ac.uk
 - Port: 2401
 - Repository path: /local/ontogrid/cvsroot
 - module: prototype

Retos Científicos y Tecnológicos en la Combinación de Grid y Semántica

Oscar Corcho
Universidad Politécnica de Madrid
March 3rd 2008

