OpenGridForum
OPEN FORUM | OPEN STANDARDS

# Database Access and Integration Services Working Group (DAIS-WG) Status Update

OGF24

Chairs:
Mario Antonioletti
*EPCC*
*Isao Kojima*
*AIST*

# OGF IPR Policies Apply

- "I acknowledge that participation in this meeting is subject to the OGF Intellectual Property Policy."
- Intellectual Property Notices Note Well: All statements related to the activities of the OGF and addressed to the OGF are subject to all provisions of Appendix B of GFD-C.1, which grants to the OGF and its participants certain licenses and rights in such statements. Such statements include verbal statements in OGF meetings, as well as written and electronic communications made at any time or place, which are addressed to:

  the OGF plenary session,
  any OGF working group or portion thereof,
  the OGF Board of Directors, the GFSG, or any member thereof on behalf of the OGF,
  the ADCOM, or any member thereof on behalf of the ADCOM,
  any OGF mailing list, including any group list, or any other list functioning under OGF auspices,
  the OGF Editor or the document authoring and review process
- Statements made outside of a OGF meeting, mailing list or other function, that are clearly not intended to be input to an OGF activity, group or function, are not subject to these provisions.
- Excerpt from Appendix B of GFD-C.1: "Where the OGF knows of rights, or claimed rights, the OGF secretariat shall attempt to obtain from the claimant of such rights, a written assurance that upon approval by the GFSG of the relevant OGF document(s), any party will be able to obtain the right to implement, use and distribute the technology or works when implementing, using or distributing technology based upon the specific specification(s) under openly specified, reasonable, non-discriminatory terms. The working group or research group proposing the use of the technology with respect to which the proprietary rights are claimed may assist the OGF secretariat in this effort. The results of this procedure shall not affect advancement of document, except that the GFSG may defer approval where a delay may facilitate the obtaining of such assurances. The results will, however, be recorded by the OGF Secretariat, and made available. The GFSG may also direct that a summary of the results be included in any GFD published containing the specification."

- OGF Intellectual Property Policies are adapted from the IETF Intellectual Property Policies that support the Internet Standards Process.

# Outline

- Background, History and the Current Status
- WS-DAIR OGSA-DAI Implementation (Alistair)
- RDF/RDF(S)  Realisations status
- Discussions

# Introduction

- **DAIS**=Database Access and Integration Service

- **Aim**=Provide consistent access method to existing, autonomously managed **databases** from **web services.**

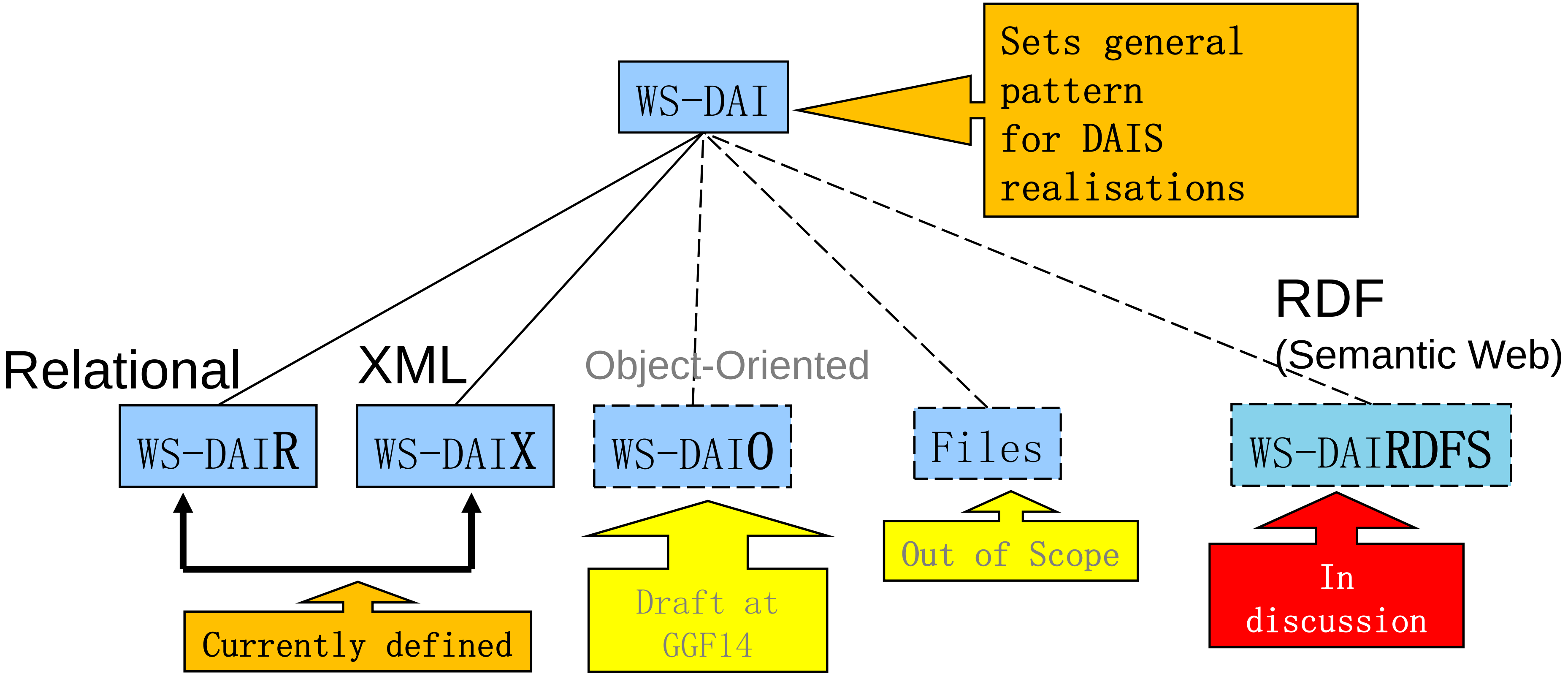
- **Databases**:
  - Relational(Oracle, DB2,MySQL,Postgres,etc)
  - XML(eXist, Xindice,,,)
  - Object-Oriented(OODB)
  - RDF(Sesame,Jena,,,)

- **Web Services**
  - Web/Grid Services(WS-I & WS-RF)

# Specifications overview:
# Define WS-xx for database access

**WS-DAI** Specification Family

WS-DAI = Web Services - Database Access and Integration

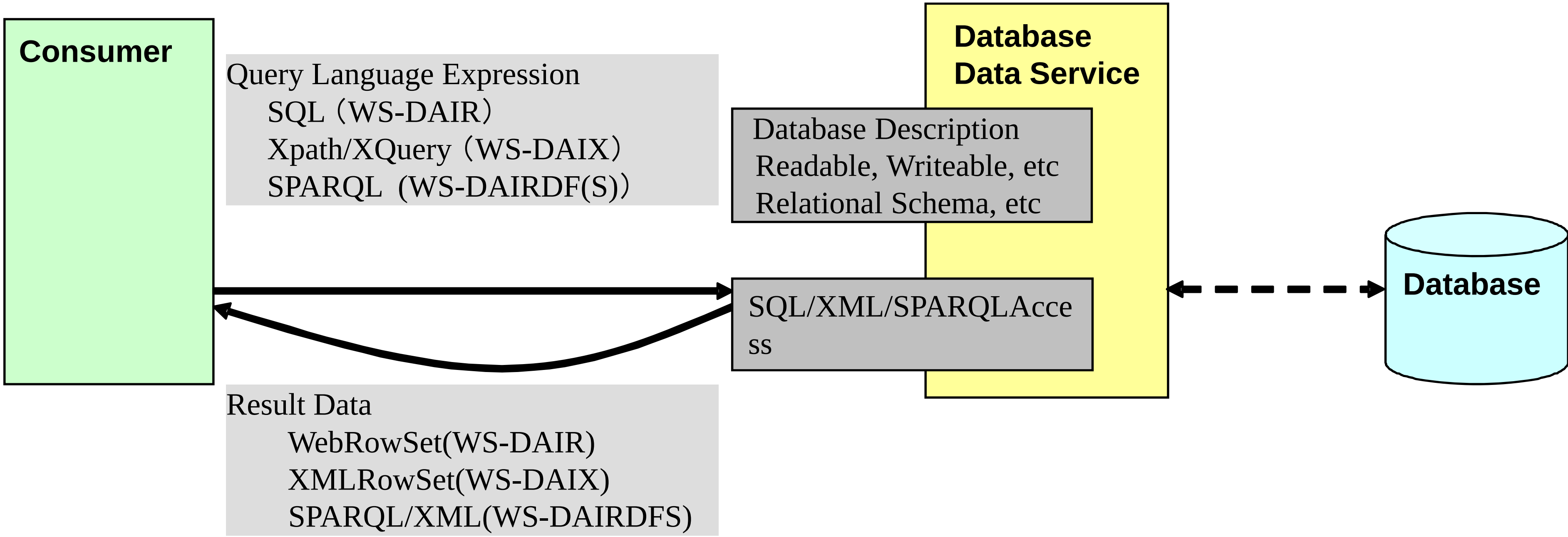


Will be dropped from the charter

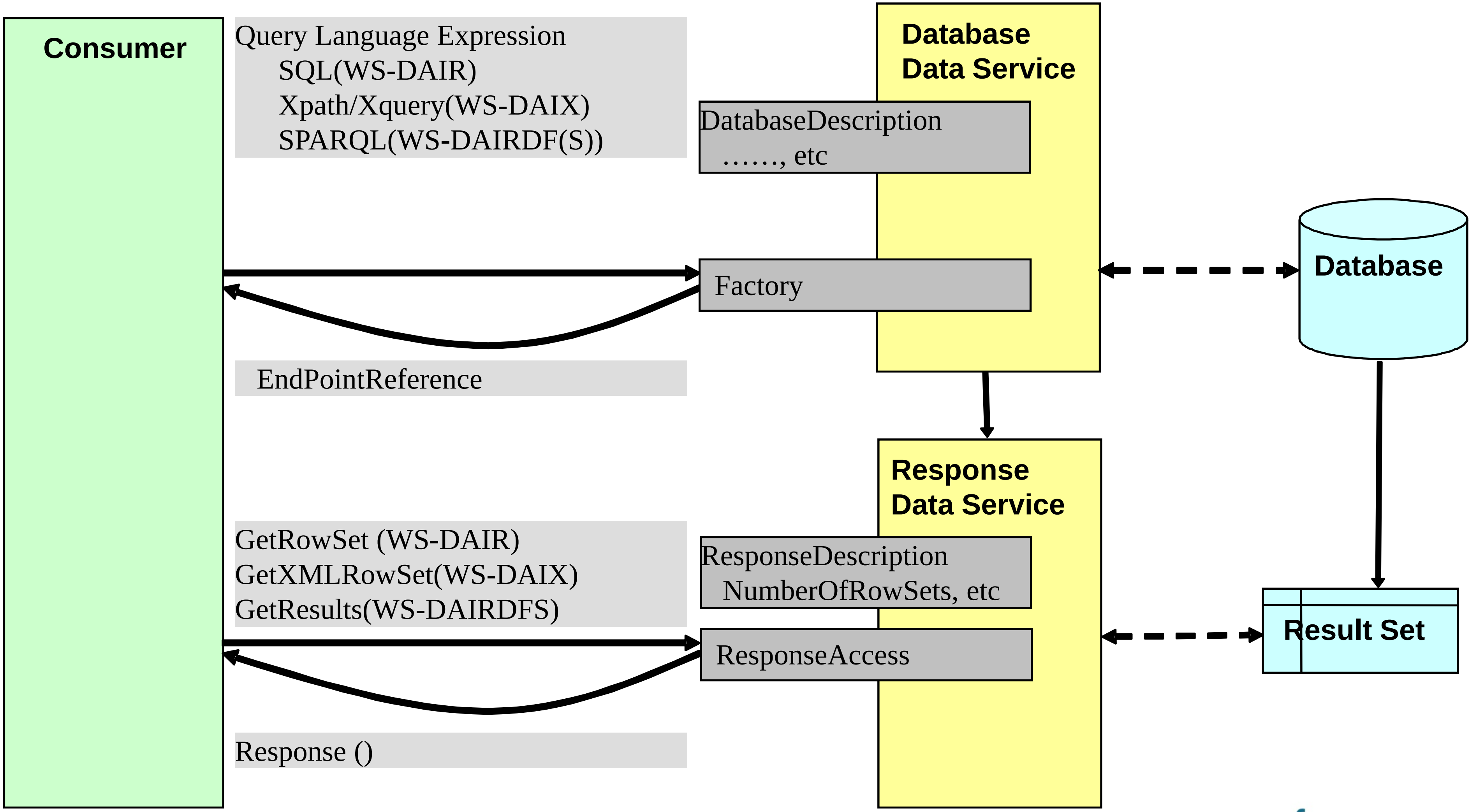
# Specification Overview:

## Provide Direct Access & Indirect Access to Database Resources

## 1) Direct Access



# 2) Indirect Access



Consumer

Query Language Expression
SQL(WS-DAIR)
Xpath/Xquery(WS-DAIX)
SPARQL(WS-DAIRDF(S))

Database
Data Service

DatabaseDescription
……., etc

Factory

EndPointReference

Database

Response
Data Service

ResponseDescription
NumberOfRowSets, etc

GetRowSet (WS-DAIR)
GetXMLRowSet(WS-DAIX)
GetResults(WS-DAIRDFS)

ResponseAccess

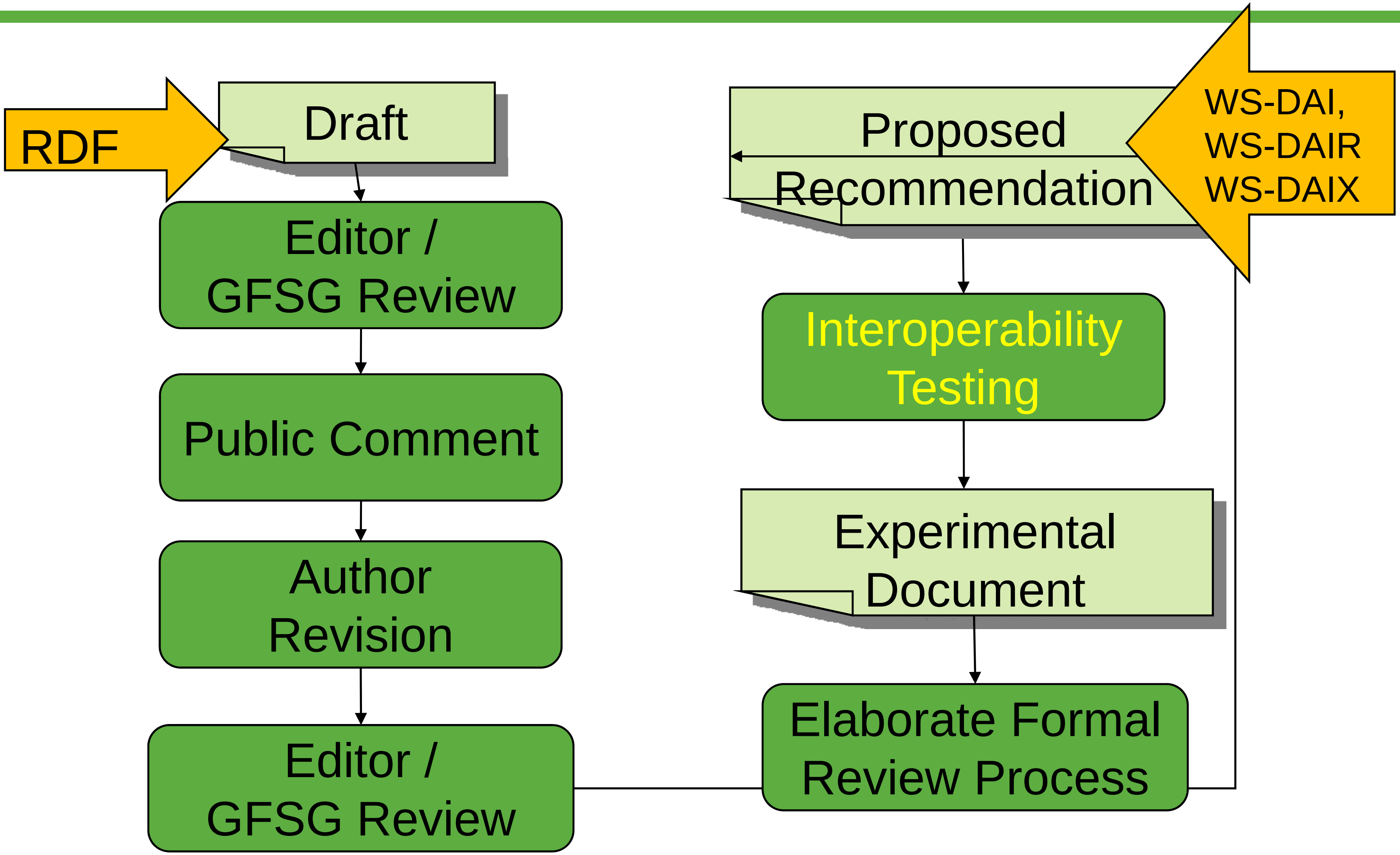
Result Set

Response ()

# History

- DAIS WG has been around for a long time
  - BOF at GGF 4 – February 2002
  - First Meeting at GGF 5 – July 2002
  - …
  - First batch of specs – September 2006
    - GFD.74 – WS-DAI
    - GFD.75 – WS-DAIR
    - GFD.76 – WS-DAIX
    - (GFD.77 Interoperability)
  - Awaiting implementations
    - DAIX with OGSA-DAI/DAIR with AMGA
  - Now RDF and RDF(S) close to completion

# OGF Document Process: Recommendation



# DAIS-WG Documents

- **Recommendation Track**:
  - Web Services Data Access and Integration – The Core (WS-DAI) Specification, Version 1.0
    http://forge.gridforum.org/tracker/index.php?aid=1711
  - Web Services Data Access and Integration – The Relational Realisation (WS-DAIR) Specification, Version 1.0
    http://forge.gridforum.org/tracker/index.php?aid=1712
  - Web Services Data Access and Integration – The XML Realization
    http://forge.gridforum.org/tracker/index.php?aid=1713

- **Informational:**
  - Interoperability Testing for DAIS Working Group Specifications
    - https://forge.gridforum.org/tracker/index.php?aid=1768

# Implementations  Status

- **WS-DAIX(XML)**
  - OGSA-DAI has implemented it(presented at OGF22)
  - Ohio State University – started (no spare cycles)

- **WS-DAIR(Relational)**
  - OGSA-DAI group (will be presented later)
  - GRelC will be implementing it
  - AMGA has implemented it (presented at OGF23)

- **WS-DAIRDF(S)(RDF)** (Specs should be finalized)
  - AIST,UPM will be implementing it.

- **Interoperability testing will be the issue**
  - When?  & How ?

    SC(Supercomputing Conrefence)08?

# Current & Future Activities

- **WS-DAI RDF(S) (will be presented later)**
  - Finish documents
  - Update motivational document with new use cases
  - Align all documents with the glossary of terms
  - Planned to submit at OGF25.
- **Charter Updates**
  - Drop WS-DAIO activity
  - Include WS-DAI RDF(S) activity
- **Collaboration with SAGA folks?**
  - Come up with use cases

    Use case Contributors wanted

# New Topics?

Current WS-DAI Specs = Remote Access Only

- Distributed Query Processing will be the next issue.

How?

Distributed/Parallel Query/Data processing

- Query optimization
- How to coordinate distributed WS-DAIs?
  - Workflow?
  - Google-based Technologies?

    MapReduce Framework, BigTable etc.

Data Integration based on Metadata

- Metadata should be standardized
- Ontology/Semantic Web

Applications

- Geospatial data integration/ Digital Repositories?

Contributors wanted

- Q&A?

# Implementation Report

OGF24

# WS-DAI RDF(S) Specification

## DAIS Working Group

Isao Kojima

*Information Technology Research Institute*
*AIST*
*Japan*

# WS-DAI-RDF(S) Standard Structure

## What is all about?

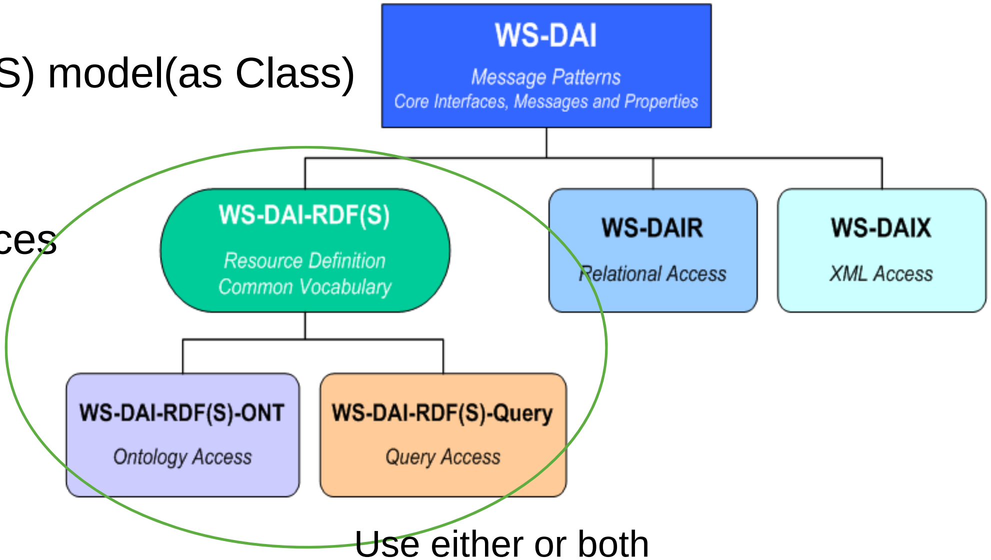
- *Providing an access mechanism to RDF(S) data resources*

New WS-DAI realization for RDF(S) data

which will consist of 2 complementary specifications

1. **RDF(S) Ontology Access**
   Ontological Primitives based on RDF(S) model(as Class)

2. **RDF(S) Querying**
   Query Language(SPARQL) for Instances



Use either or both according to your needs

# RDF (Resource Description Framework)

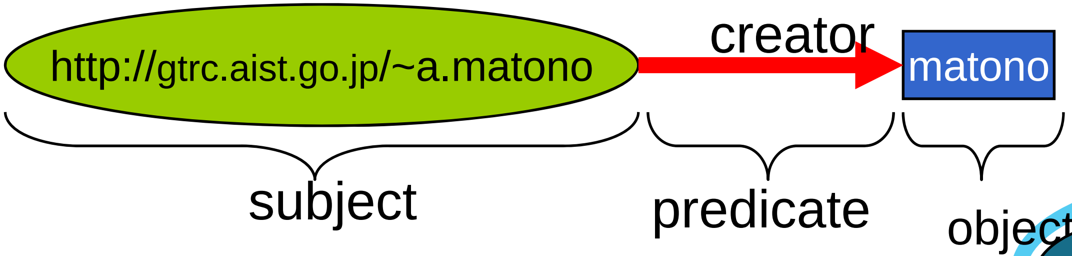
- Common metadata/knowledge description in Semantic Web Applications
- Number of RDF data is increasing

=> Support by the database technology/specs will be important

Swoogle 174,981,639 RDF triples

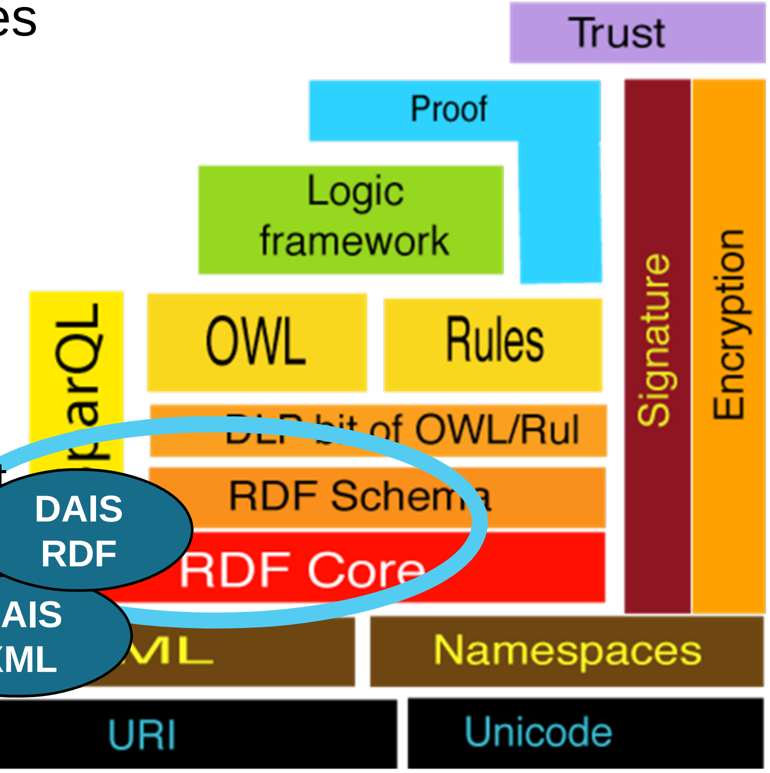
Dbpedia 110,241,463 RDF triples

Statement:
“creator of “http://gtrc.aist.go.jp/~a.matono/ is matono”



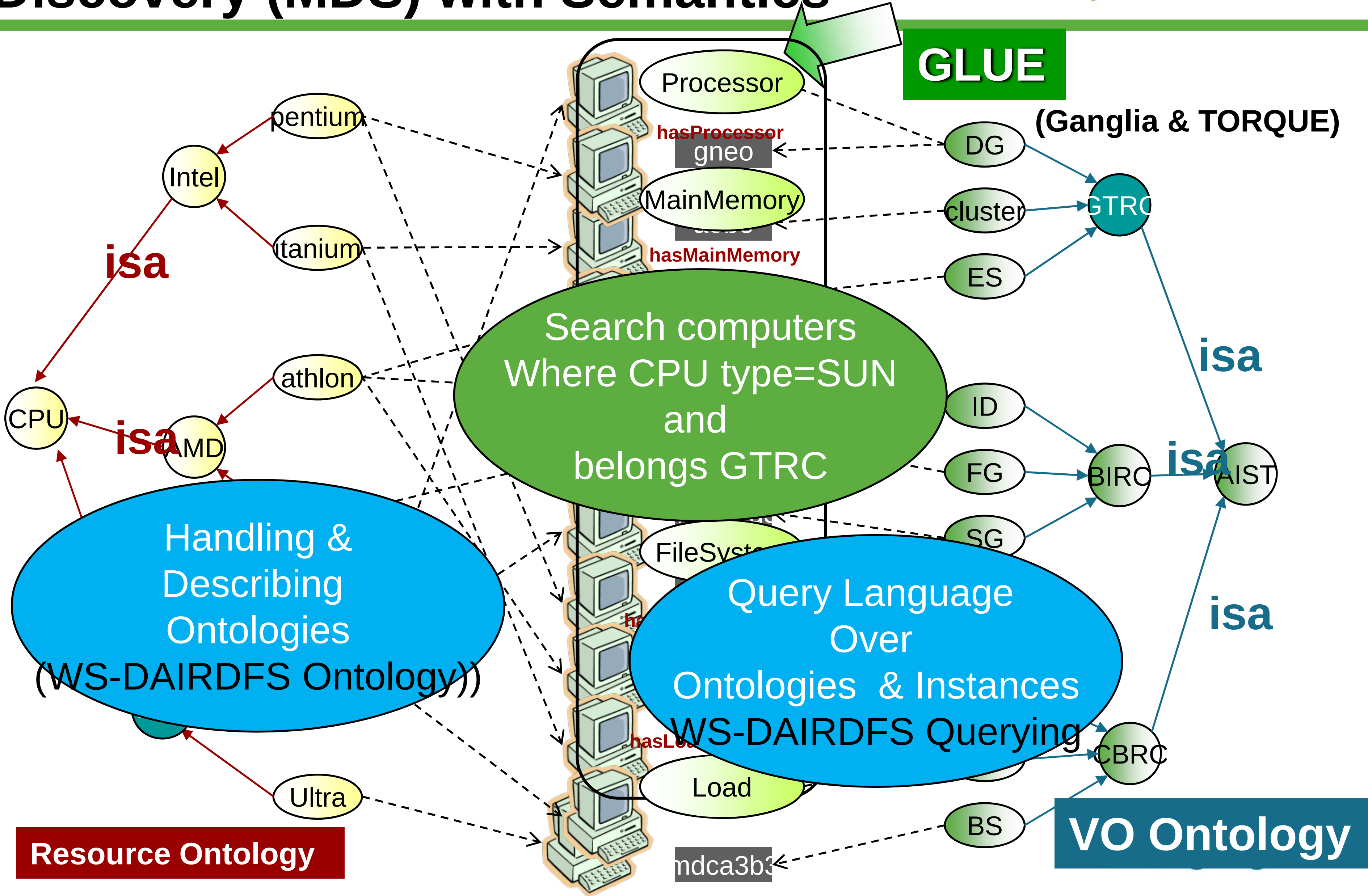
```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
            xmlns:dc="http://dublincore.org/2001/08/14/dces#">
   <rdf:Description about="http://gtrc.aist.go.jp/~a.matono">
        <dc:creator>matono</dc:creator>
   </rdf:Description>
</rdf:RDF>
```

XML-based Format



Layer Cake of the Semantic Web

# Example : Grid Resource Monitoring & Discovery (MDS) with Semantics



GLUE

(Ganglia & TORQUE)

Search computers
Where CPU type=SUN
and
belongs GTRC

Handling &
Describing
Ontologies
(WS-DAIRDFS Ontology))

Query Language
Over
Ontologies & Instances
WS-DAIRDFS Querying

Resource Ontology

VO Ontology

M.Esteban, I Kojima et al.

“Accessing RDF(S) Data Resources in Service-based Grid Environment”

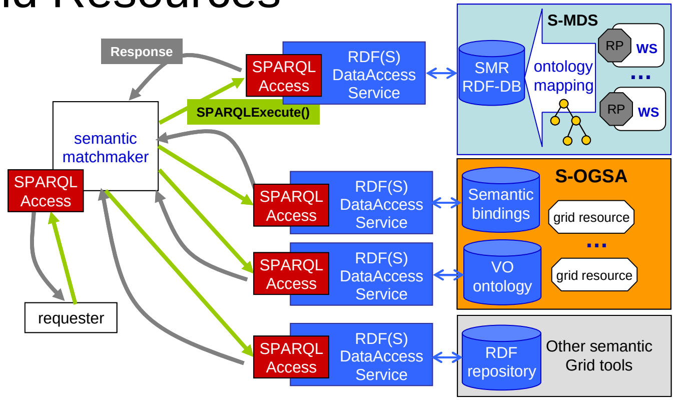
To appear in the OGF Special Issue of

“Concurrency and Computation” Journal

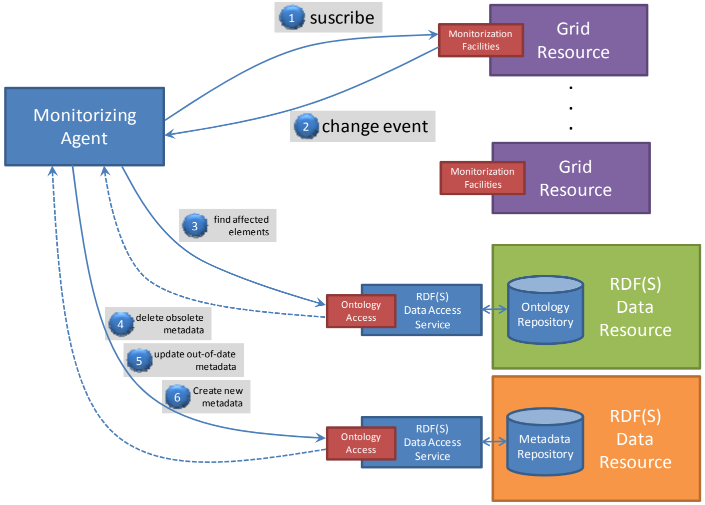
Preliminary Version is available from OGF  website.

# Usecases – Semantic Web & the Grid

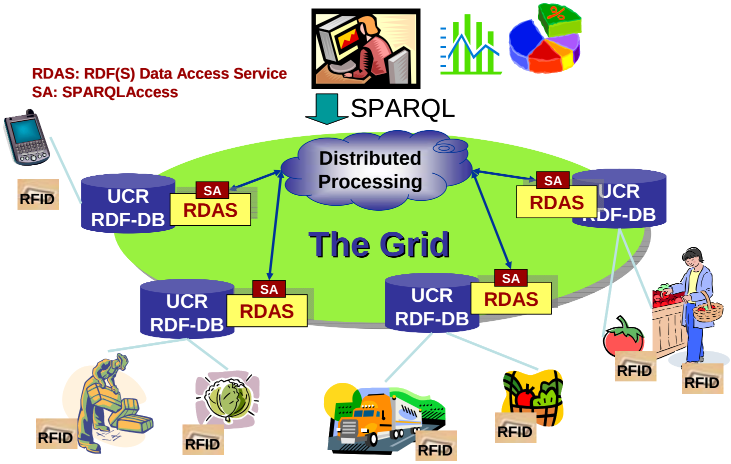
## Query-based Semantic Matchmaking for Grid Resources



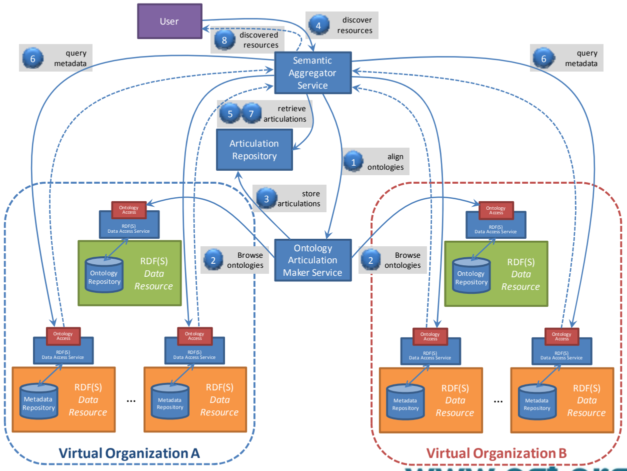
## Grid Resource Monitoring & Annotation



## Distributed SPARQL Processing for Distributed Metadata for RFIDs



## Aggregation of Heterogeneous Grid Resource Metadata



# Uniqueness of the Activity

- **In OGF**
  - Currently only one activity to set the standards for handling RDF data
- **Ontology Access**
  - No corresponding  standards in W3C
- **Querying**
  - Support W3C SPARQL and related standards.
  - No indirect access is supported in W3C standards.
- **No such activity that supports both in a single framework.**

# DAIS for RDF:  History

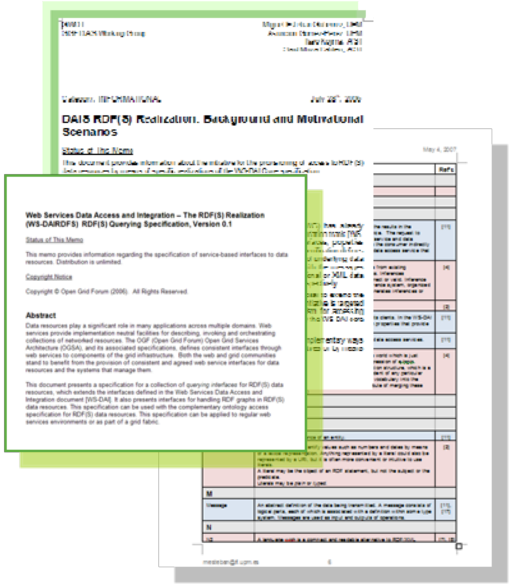
- **2006.02: GGF16 at Athens**
  - DAIS for RDF  BOF: Share the Motivation
- **2006.05: GGF17 at Tokyo**
  - Charter Discussion
  - Focus on RDF and RDF Schema (RDF(S))
  - Scope/Roadmap/Deliverables
- **2006.06: RDF F2F at Edinburgh**
  - Decide to make an informational document
    - Motivational Document : Structure Discussion
- **2006.09: GGF18 at Washington**
  - Motivational Doc Presented
- **2007.02 : OGF19 at  Chapel Hill, NC**
  - 2 initial Specification Documents Presented
- **2007.05: OGF20 at Manchester**
  - Initial version of the “Glossary of the Terms” is presented
  - More Use Cases Presented.
- **2007.09 : OGF21 at Seattle**
  - Documents updated
    - Glossary of Terms
    - Executive summary for Querying spec is presented.
  - Roadmap/Schedule is  revised
- **2008.02:OGF22 at Boston**
  - Agreed Terms Presented
  - Documents updated

# Current Documents

## 4 documents are on the forge

*Please download and have a look*

1. **DAIS RDF(S) Background & Motivational Scenarios**
2. **WS-DAI RDF(S) Querying**
3. **WS-DAI RDF(S) Ontology Access**
4. **Glossary of Terms**



DAIS RDF(S) Realization: Background and Motivational Scenarios

Status of This Memo

May 4, 2007

Refs

M

Message

N

Web Services Data Access and Integration – The RDF(S) Realization (WS-DAIRDFS) RDF(S) Querying Specification, Version 0.1

Status of This Memo

This memo provides information regarding the specification of service-based interfaces to data resources. Distribution is unlimited.

Copyright Notice

Copyright © Open Grid Forum (2006). All Rights Reserved.

Abstract

Data resources play a significant role in many applications across multiple domains. Web services provide implementation neutral facilities for describing, invoking and orchestrating collections of networked resources. The OGF (Open Grid Forum) Open Grid Services Architecture (OGSA), and its associated specifications, defines consistent interfaces through web services to components of the grid infrastructure. Both the web and grid communities stand to benefit from the provision of consistent and agreed web service interfaces for data resources and the systems that manage them.

This document presents a specification for a collection of querying interfaces for RDF(S) data resources, which extends the interfaces defined in the Web Services Data Access and Integration document [WS-DAI]. It also presents interfaces for handling RDF graphs in RDF(S) data resources. This specification can be used with the complementary ontology access specification for RDF(S) data resources. This specification can be applied to regular web services environments or as part of a grid fabric.

*Any comments and feedbacks are welcome*

*5 .* **Executive summary of Querying Specification**

*Brief overview and motivational use cases*

*- To promote this activity to wider audience*

QUERYING RDF DATA IN THE GRID

<WS-DAI RDF(S) Querying Specification>

QUERYING RDF DATA: USE CASES

<WS-DAI RDF(S) Querying Specification>

Semantic Matchmaking of Grid Resources

Distributed RDF Storage for Ubiquitous Objects

The Grid

# Terminology

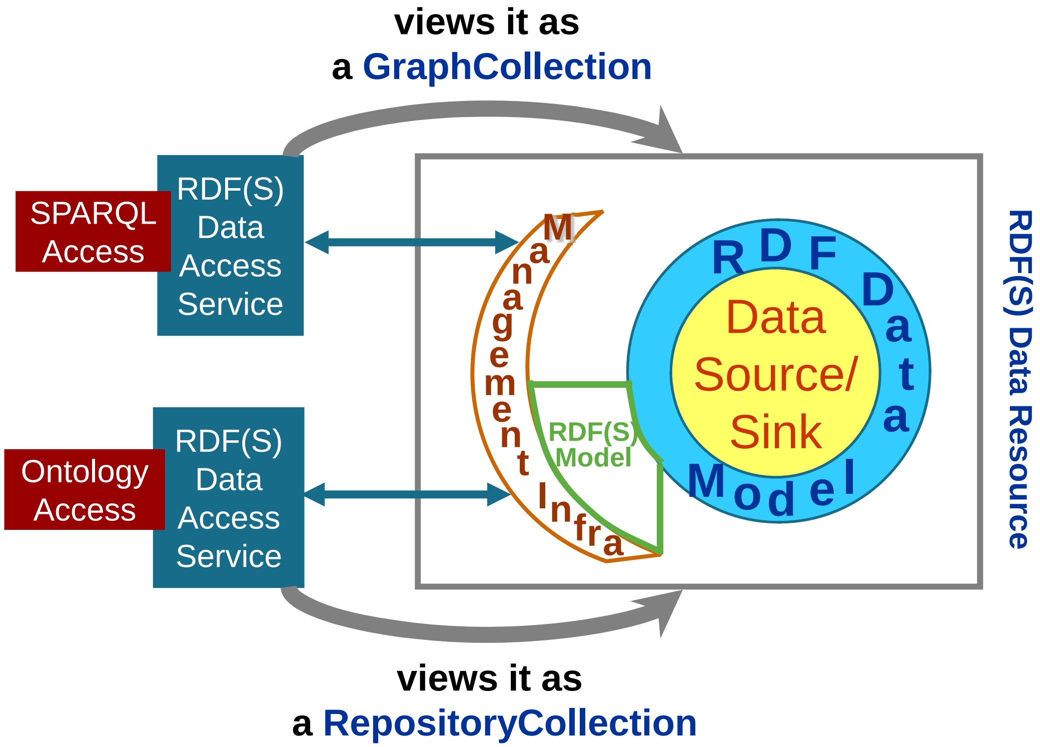
- RDF(S) Data Resource
  - Data source/sink that is based on the RDF data model + management infrastructure which may exhibit RDF(S) model based views

## Relationships of Terms between 2 specs.

*Ontology* *Querying*

- Repository ←→ RDF Graph
  - A set of RDF Triples
- RepositoryCollection ←→ GraphCollection
  - A set of Repositories or RDF Graphs

# Terminology



- The naming duality is due to the way in which each specification views the RDF(S) data resources
- (The location transparent names via an End Point Identifier (EPI) can be used to identify the resource sameness)

# Progress after OGF23

- *Specification Document Updates*
  - *Based on the agreed terms.*
    - *Will be Presented Here*

      *Query Spec is almost stable*

      *Ontology Spec still need to be  concised*

- *Will Update the Charter to include RDF(S) Activity*
  - *Deliverables: Re-scheduled*

# WS-DAI RDF(S) Ontology Access

OGF24 DAIS Working Group

# Introduction

## Goal & Objectives

- Goal:

  “To provide access to RDF(S) data sources in a grid fashion, *without constraining what the user could manually do* (specially when serializing a local RDF/XML file), and *facilitating common tasks*, providing a highly *flexible and adaptable* access mechanism that *hides technicalities of RDF(S)* to the user, whilst *transparently exploits* its *full semantics*”

- Objectives:
  - Full RDF(S) coverage
  - R+W capabilities
  - Granular data access
  - Resource centric API

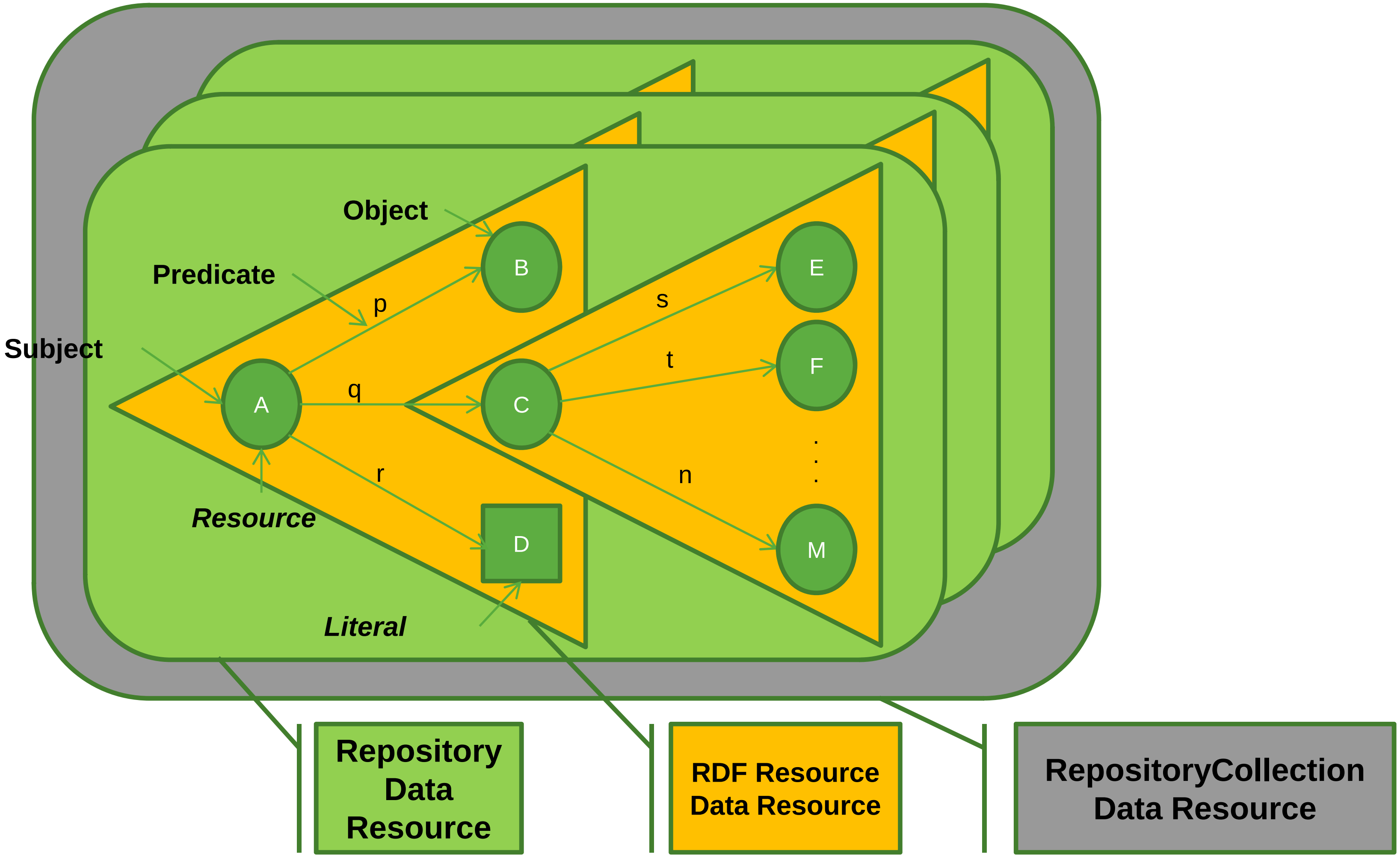
# Outline

- Introduction
- Overview
  - Concepts
  - Data Resources
  - Interfaces
  - Profiles
- Current status
- Outstanding issues
- Future work
- Conclusions

# Overview

## Concepts

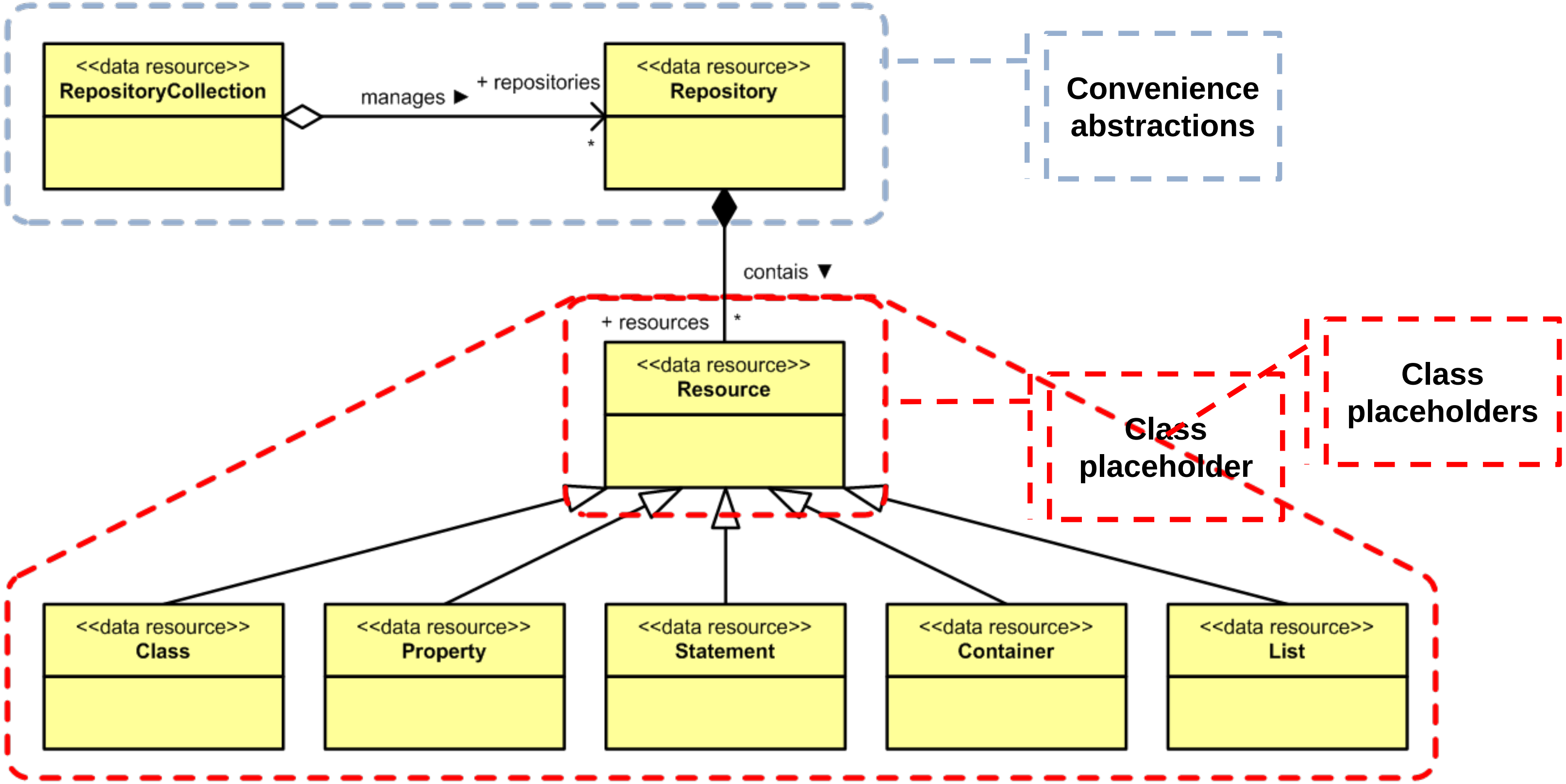
### Merging RDF(S) and WS-DAI Concepts



# Overview

## Data Resources

### Types & Organization



Convenience abstractions

Class placeholder

Class placeholders

# Overview

## Concepts, revisited

### RDF Resource lifecycle

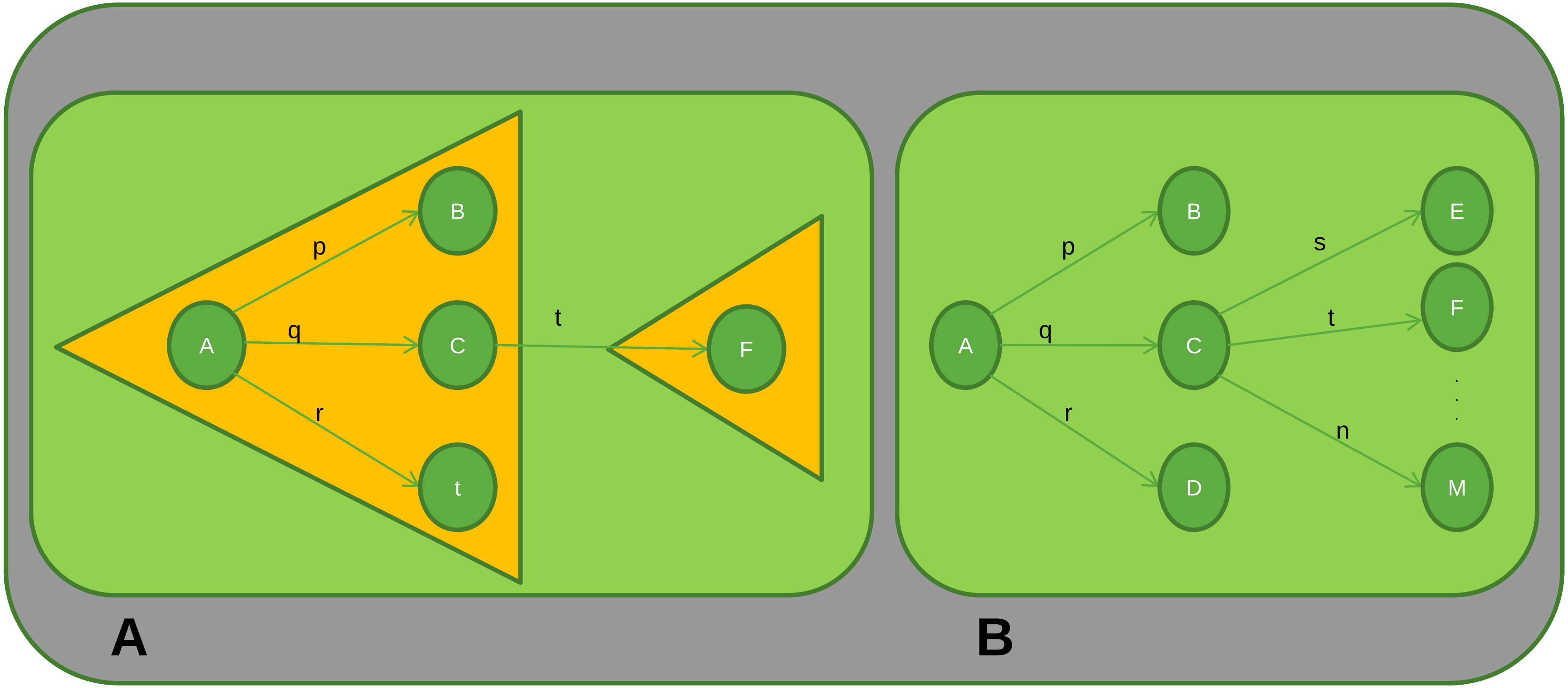
- **Creation:**
  - *Explicit:* a triple is created with the resource as subject.
  - *Implicit:* a triple is created with the resource as predicate or object.
- **(Property value) Attachment:**
  - *Explicit:* adding a new triple which uses an already existing resource as subject.
  - *Implicit:* adding a triple which uses an existing resource as predicate or object, and due to RDF(S) entailment rules, new property values are automatically attached.
- **(Property value) Detachment:**
  - *Explicit:* removing a triple which has the resource as subject.
  - *Implicit:* removing a triple that has the resource as predicate or object, and as a result inferred property values are lost (no longer explicit).
- **Removal:**
  - No triples using the resource exist.

# Overview

## Data Resources

### Lifecycle example scenario

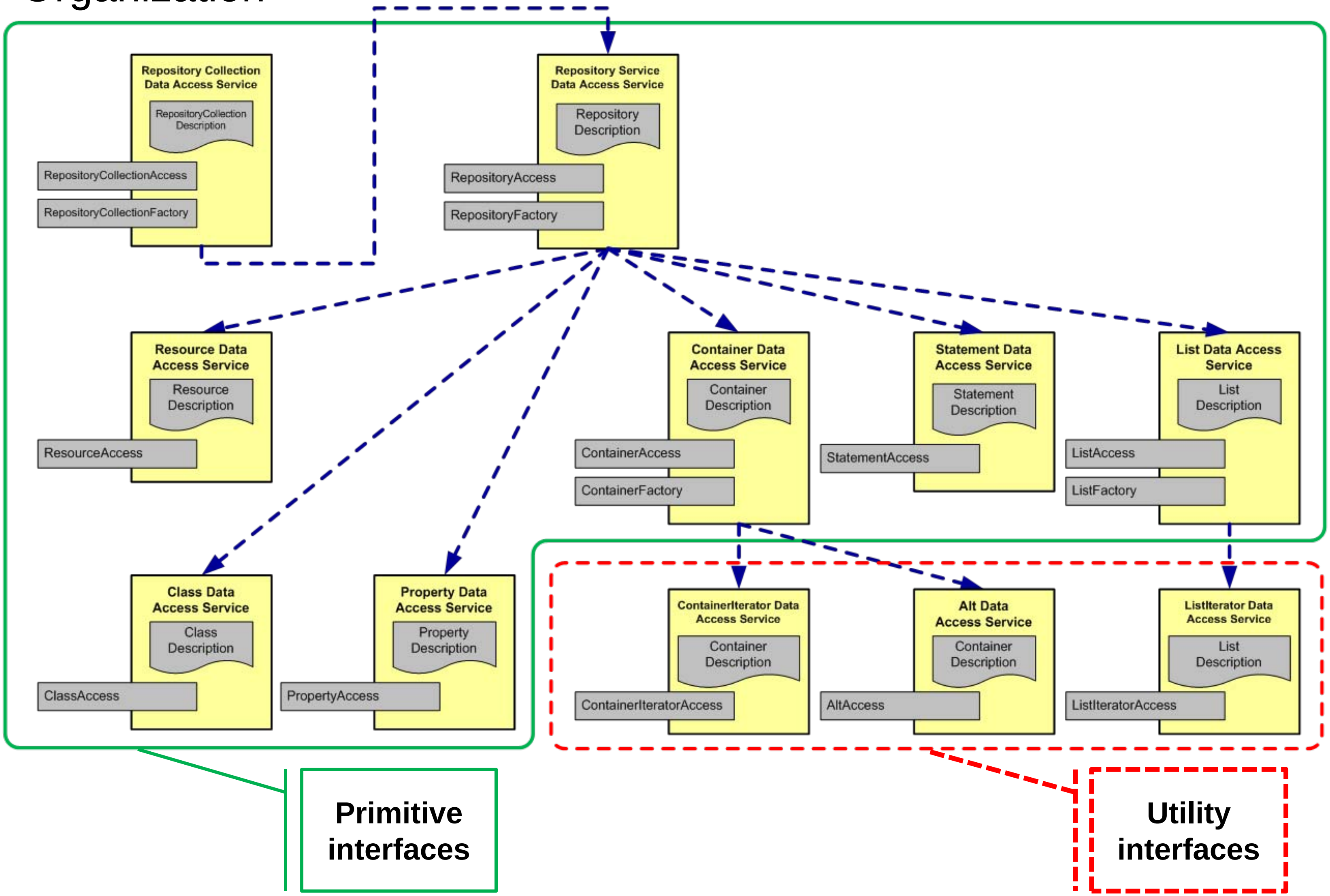
**“A”.remove (“C”, “t”, “F”)**



# Overview

## Interfaces

### Organization



**Primitive interfaces**

**Utility interfaces**

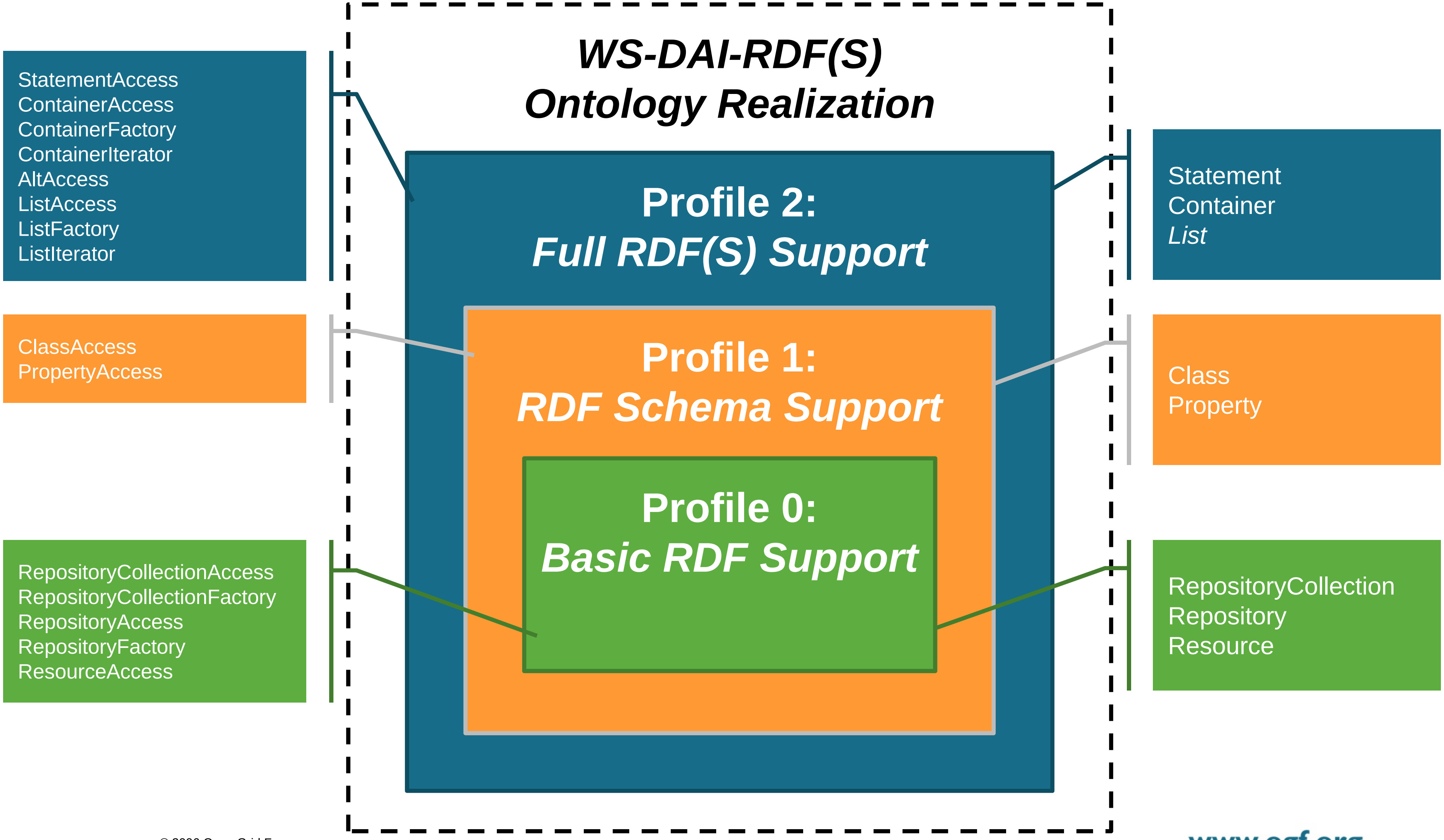
# Overview

## Interfaces

### Summary

| Data Resource | Interface Name | Category | Properties | | | Messages | |
|---|---|---|---|---|---|---|---|
| | | | *Static* | *Confi.* | *Total* | *Direct Access* | *Indirect Access* |
| **Repository Collection** | **RepositoryCollectionDescription** | Description | 2 | 0 | 2 | 0 | 0 |
| | **RepositoryCollectionAccess** | Access | 0 | 0 | 0 | 4 | 0 |
| | **RepositoryCollectionFactory** | Factory | 0 | 0 | 0 | 0 | 1 |
| **Repository** | **RepositoryDescription** | Description | 1 | 0 | 1 | 0 | 0 |
| | **RepositoryAccess** | Access | 0 | 0 | 0 | 6 | 0 |
| | **RepositoryFactory** | Factory | 0 | 0 | 0 | 0 | 6 |
| **Resource** | **ResourceDescription** | Description | 2 | 0 | 2 | 0 | 0 |
| | **ResourceAccess** | Access | 0 | 0 | 0 | 13 | 0 |
| **Class** | **ClassDescription** | Description | 2 | 0 | 2 | 0 | 0 |
| | **ClassAccess** | Access | 0 | 0 | 0 | 8 | 0 |
| **Property** | **PropertyDescription** | Description | 2 | 0 | 2 | 0 | 0 |
| | **PropertyAccess** | Access | 0 | 0 | 0 | 13 | 0 |
| **Statement** | **StatementDescription** | Description | 2 | 0 | 2 | 0 | 0 |
| | **StatementAccess** | Access | 0 | 0 | 0 | 5 | 0 |
| **List** | **ListDescription** | Description | 2 | 0 | 2 | 0 | 0 |
| | **ListAccess** | Access | 0 | 0 | 0 | 9 | 0 |
| | **ListFactory** | Factory | 0 | 0 | 0 | 0 | 1 |
| | **ListIteratorAccess** | Access | 0 | 0 | 0 | 2 | 0 |
| **Container** | **ContainerDescription** | Description | 3 | 0 | 3 | 0 | 0 |
| | **ContainerAccess** | Access | 0 | 0 | 0 | 6 | 0 |
| | **ContainerFactory** | Factory | 0 | 0 | 0 | 0 | 2 |
| | **ContainerIteratorAccess** | Access | 0 | 0 | 0 | 2 | 0 |
| | **AltAccess** | Access | 0 | 0 | 0 | 8 | 0 |
| **TOTAL** | | | 16 | 0 | 16 | 76 | 10 |

# Overview
## Profiles



StatementAccess
ContainerAccess
ContainerFactory
ContainerIterator
AltAccess
ListAccess
ListFactory
ListIterator

ClassAccess
PropertyAccess

RepositoryCollectionAccess
RepositoryCollectionFactory
RepositoryAccess
RepositoryFactory
ResourceAccess

***WS-DAI-RDF(S)***
***Ontology Realization***

**Profile 2:**
***Full RDF(S) Support***

**Profile 1:**
***RDF Schema Support***

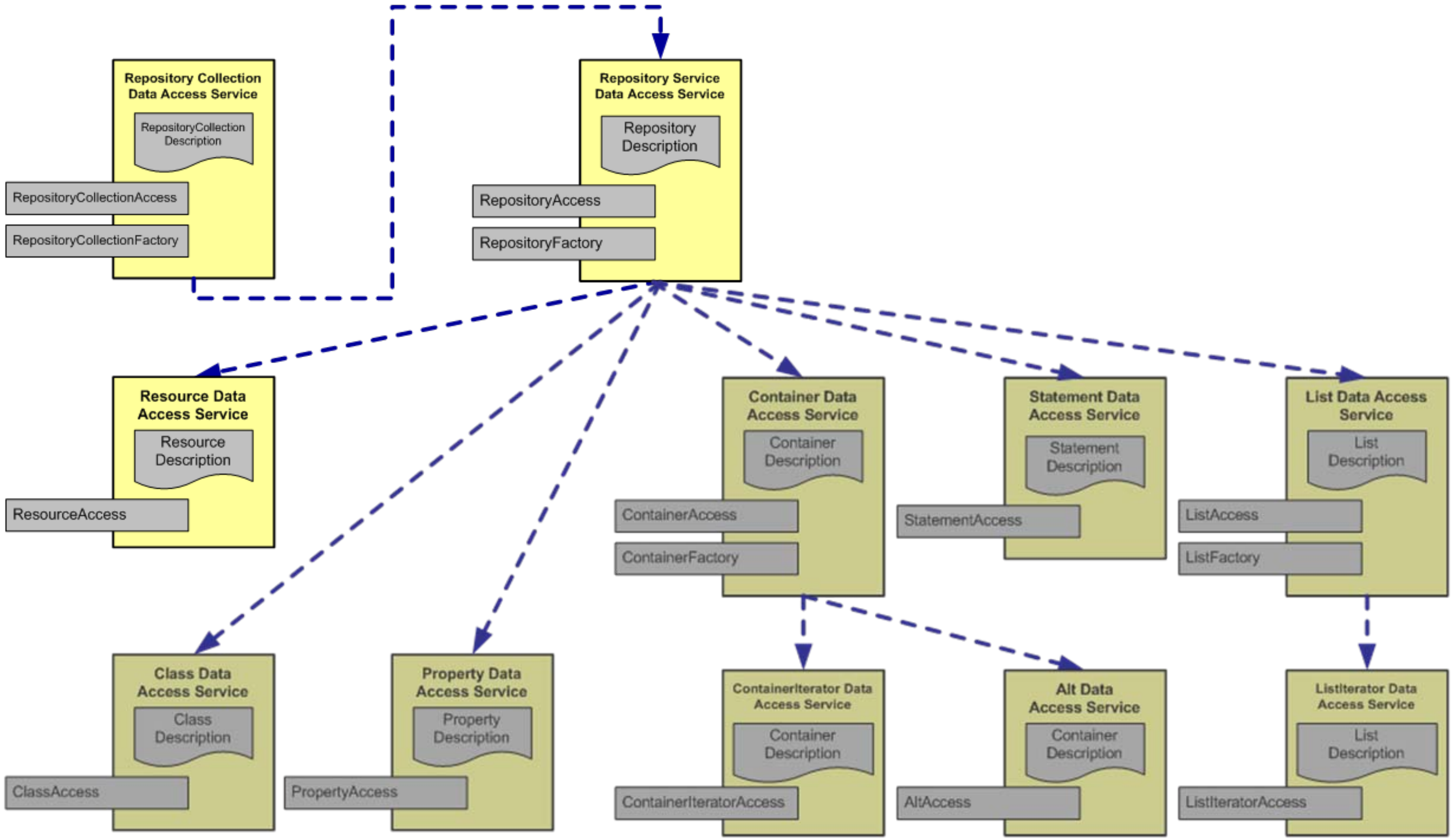
**Profile 0:**
***Basic RDF Support***

Statement
Container
*List*

Class
Property

RepositoryCollection
Repository
Resource

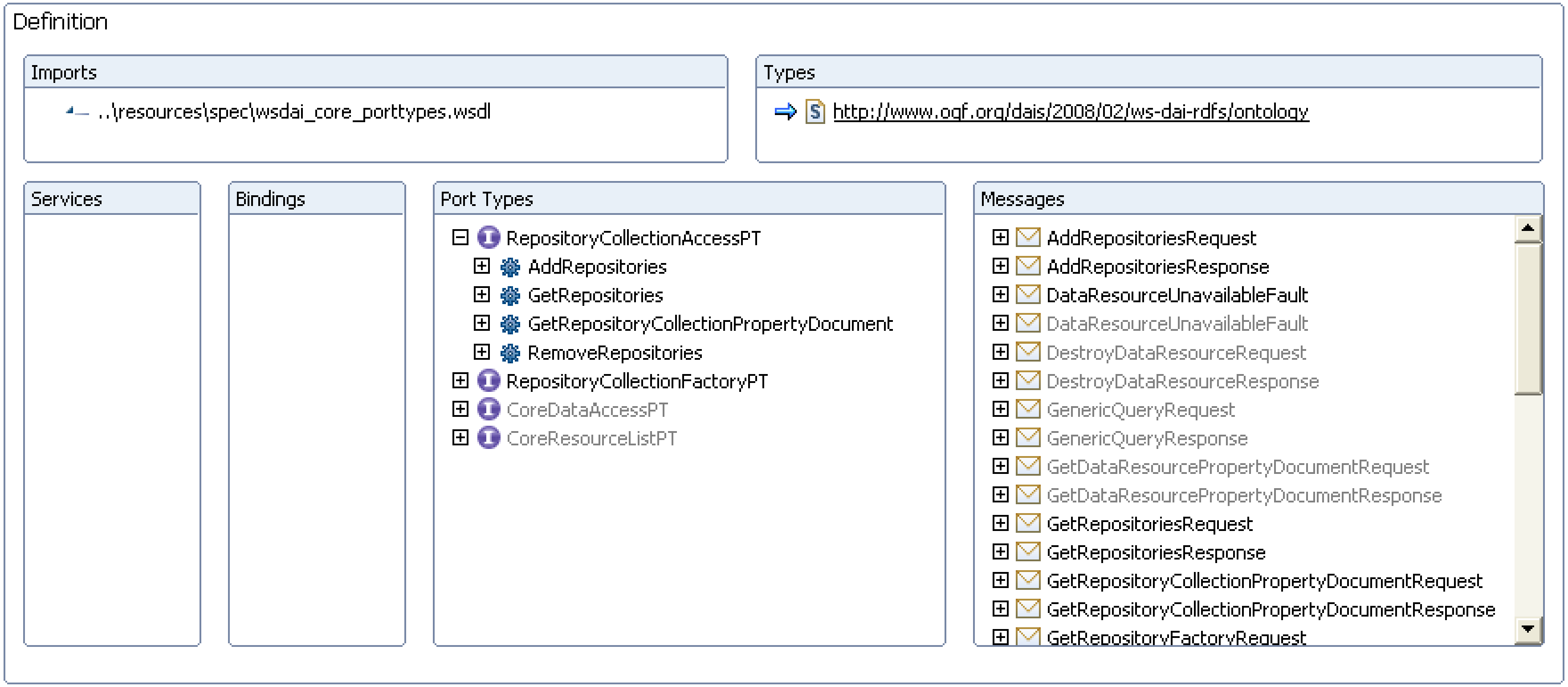
# Current Status
## Profile 0



# Current Status

## Profile 0

### RepositoryCollectionAccess



Definition

Imports

- ..\resources\spec\wsdai_core_porttypes.wsdl

Types

- http://www.ogf.org/dais/2008/02/ws-dai-rdfs/ontology

Services

Bindings

Port Types

- ⊟ RepositoryCollectionAccessPT
  - ⊞ AddRepositories
  - ⊞ GetRepositories
  - ⊞ GetRepositoryCollectionPropertyDocument
  - ⊞ RemoveRepositories
- ⊞ RepositoryCollectionFactoryPT
- ⊞ CoreDataAccessPT
- ⊞ CoreResourceListPT

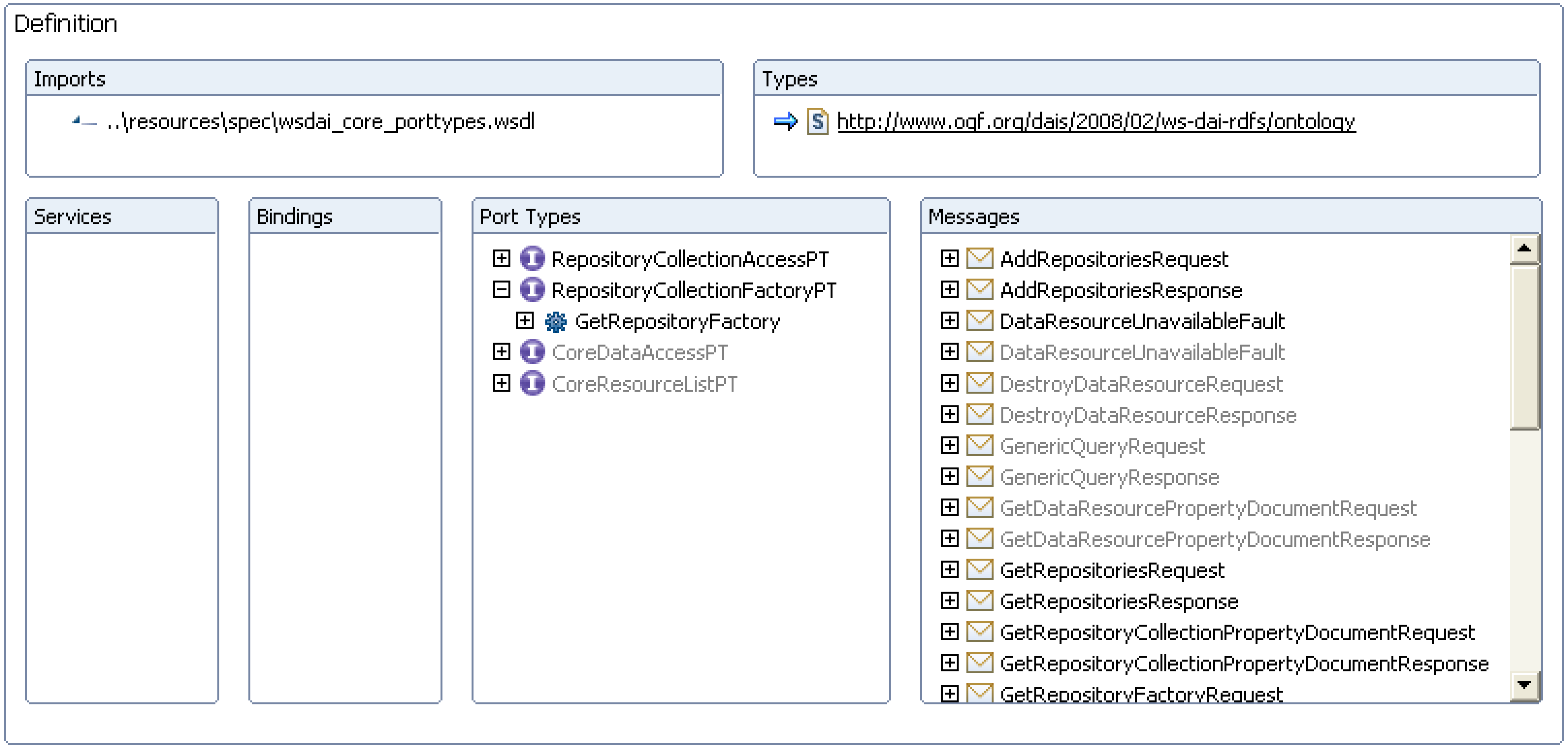
Messages

- ⊞ AddRepositoriesRequest
- ⊞ AddRepositoriesResponse
- ⊞ DataResourceUnavailableFault
- ⊞ DataResourceUnavailableFault
- ⊞ DestroyDataResourceRequest
- ⊞ DestroyDataResourceResponse
- ⊞ GenericQueryRequest
- ⊞ GenericQueryResponse
- ⊞ GetDataResourcePropertyDocumentRequest
- ⊞ GetDataResourcePropertyDocumentResponse
- ⊞ GetRepositoriesRequest
- ⊞ GetRepositoriesResponse
- ⊞ GetRepositoryCollectionPropertyDocumentRequest
- ⊞ GetRepositoryCollectionPropertyDocumentResponse
- ⊞ GetRepositoryFactoryRequest

# Current Status

## Profile 0

### RepositoryCollectionFactory



Definition

Imports

..\resources\spec\wsdai_core_porttypes.wsdl

Types

http://www.ogf.org/dais/2008/02/ws-dai-rdfs/ontology

Services

Bindings

Port Types

- ⊞ RepositoryCollectionAccessPT
- ⊟ RepositoryCollectionFactoryPT
  - ⊞ GetRepositoryFactory
- ⊞ CoreDataAccessPT
- ⊞ CoreResourceListPT

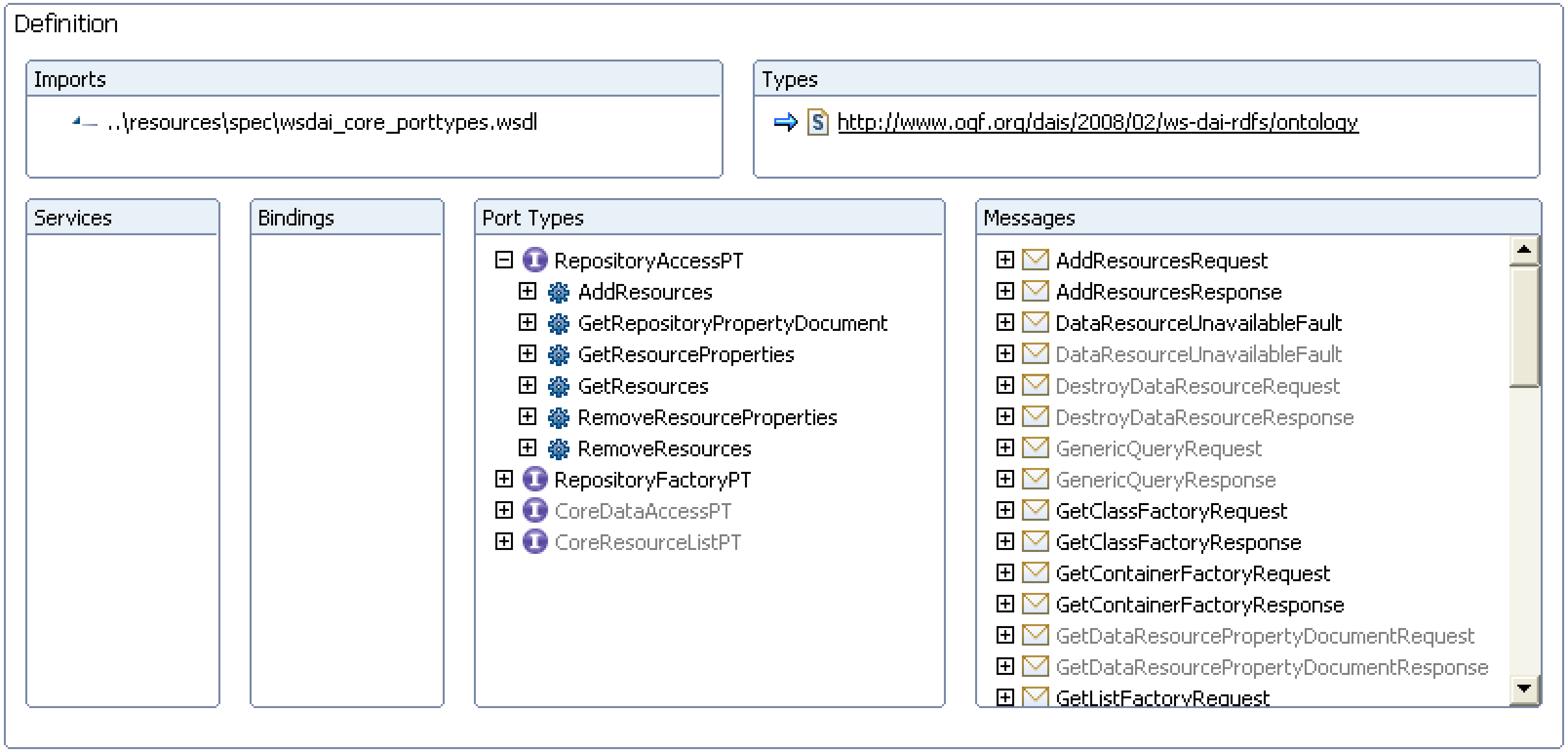
Messages

- ⊞ AddRepositoriesRequest
- ⊞ AddRepositoriesResponse
- ⊞ DataResourceUnavailableFault
- ⊞ DataResourceUnavailableFault
- ⊞ DestroyDataResourceRequest
- ⊞ DestroyDataResourceResponse
- ⊞ GenericQueryRequest
- ⊞ GenericQueryResponse
- ⊞ GetDataResourcePropertyDocumentRequest
- ⊞ GetDataResourcePropertyDocumentResponse
- ⊞ GetRepositoriesRequest
- ⊞ GetRepositoriesResponse
- ⊞ GetRepositoryCollectionPropertyDocumentRequest
- ⊞ GetRepositoryCollectionPropertyDocumentResponse
- ⊞ GetRepositoryFactoryRequest

# Current Status

## Profile 0

### RepositoryAccess



Definition

Imports

- ..\resources\spec\wsdai_core_porttypes.wsdl

Types

- http://www.ogf.org/dais/2008/02/ws-dai-rdfs/ontology

Services

Bindings

Port Types

- RepositoryAccessPT
  - AddResources
  - GetRepositoryPropertyDocument
  - GetResourceProperties
  - GetResources
  - RemoveResourceProperties
  - RemoveResources
- RepositoryFactoryPT
- CoreDataAccessPT
- CoreResourceListPT

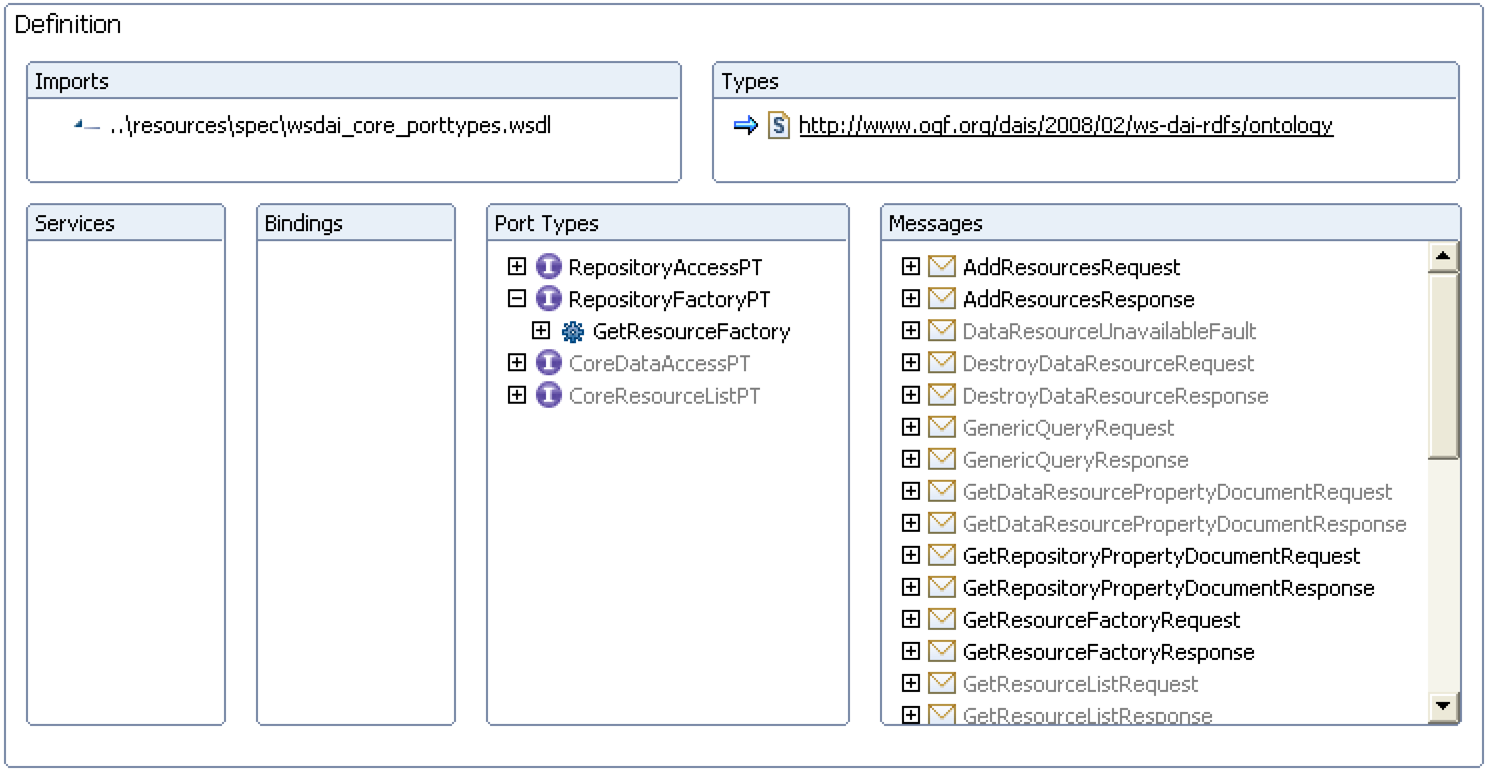
Messages

- AddResourcesRequest
- AddResourcesResponse
- DataResourceUnavailableFault
- DataResourceUnavailableFault
- DestroyDataResourceRequest
- DestroyDataResourceResponse
- GenericQueryRequest
- GenericQueryResponse
- GetClassFactoryRequest
- GetClassFactoryResponse
- GetContainerFactoryRequest
- GetContainerFactoryResponse
- GetDataResourcePropertyDocumentRequest
- GetDataResourcePropertyDocumentResponse
- GetListFactoryRequest

# Current Status

## Profile 0

### RepositoryFactory



Definition

Imports

..\resources\spec\wsdai_core_porttypes.wsdl

Types

http://www.ogf.org/dais/2008/02/ws-dai-rdfs/ontology

Services

Bindings

Port Types

- ⊞ RepositoryAccessPT
- ⊟ RepositoryFactoryPT
  - ⊞ GetResourceFactory
- ⊞ CoreDataAccessPT
- ⊞ CoreResourceListPT

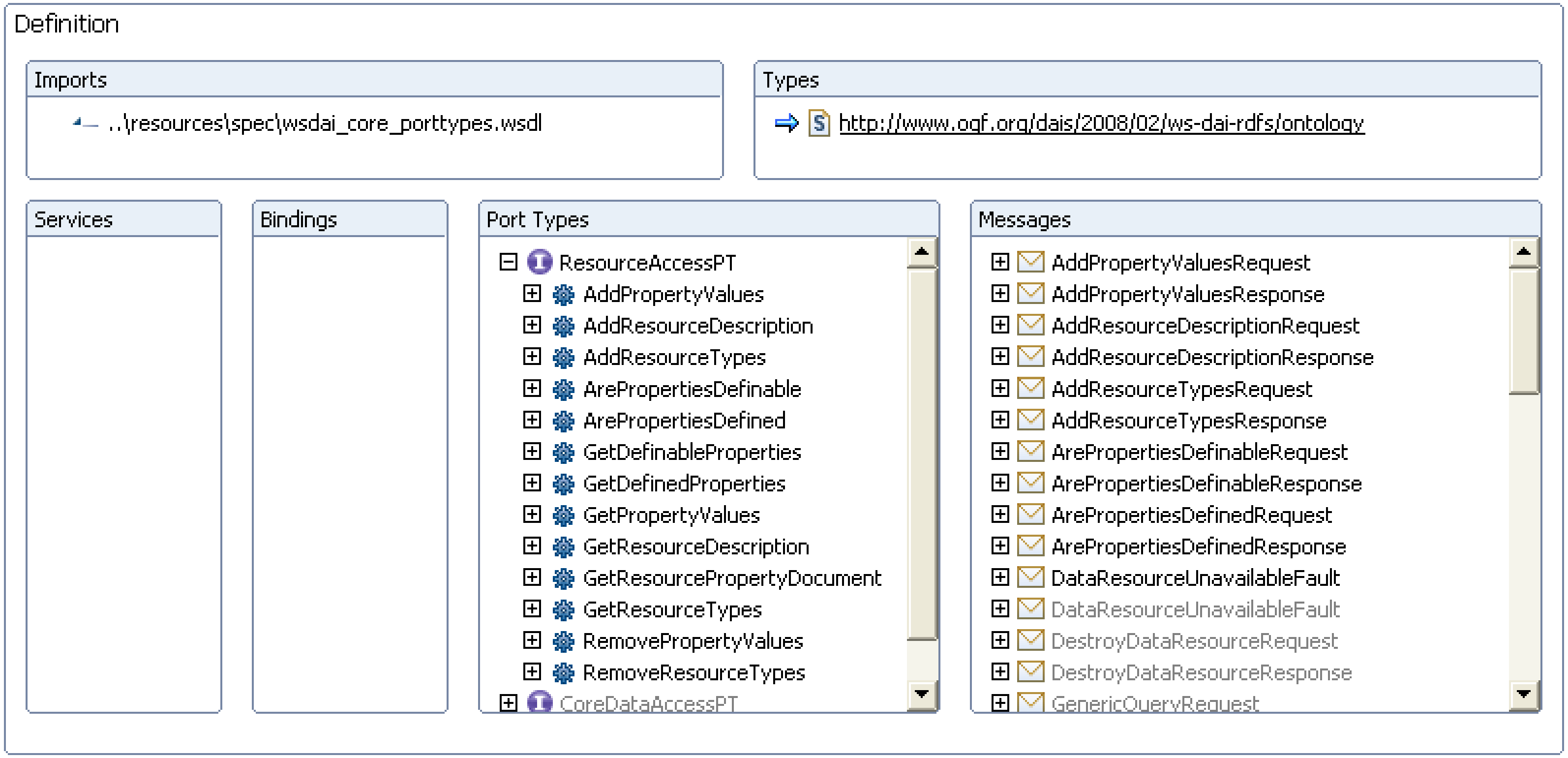
Messages

- ⊞ AddResourcesRequest
- ⊞ AddResourcesResponse
- ⊞ DataResourceUnavailableFault
- ⊞ DestroyDataResourceRequest
- ⊞ DestroyDataResourceResponse
- ⊞ GenericQueryRequest
- ⊞ GenericQueryResponse
- ⊞ GetDataResourcePropertyDocumentRequest
- ⊞ GetDataResourcePropertyDocumentResponse
- ⊞ GetRepositoryPropertyDocumentRequest
- ⊞ GetRepositoryPropertyDocumentResponse
- ⊞ GetResourceFactoryRequest
- ⊞ GetResourceFactoryResponse
- ⊞ GetResourceListRequest
- ⊞ GetResourceListResponse

# Current Status

## Profile 0

### ResourceAccess



Definition

Imports

..\resources\spec\wsdai_core_porttypes.wsdl

Types

http://www.ogf.org/dais/2008/02/ws-dai-rdfs/ontology

Services

Bindings

Port Types

```
⊟ ResourceAccessPT
    ⊞ AddPropertyValues
    ⊞ AddResourceDescription
    ⊞ AddResourceTypes
    ⊞ ArePropertiesDefinable
    ⊞ ArePropertiesDefined
    ⊞ GetDefinableProperties
    ⊞ GetDefinedProperties
    ⊞ GetPropertyValues
    ⊞ GetResourceDescription
    ⊞ GetResourcePropertyDocument
    ⊞ GetResourceTypes
    ⊞ RemovePropertyValues
    ⊞ RemoveResourceTypes
⊞ CoreDataAccessPT
```

Messages

```
⊞ AddPropertyValuesRequest
⊞ AddPropertyValuesResponse
⊞ AddResourceDescriptionRequest
⊞ AddResourceDescriptionResponse
⊞ AddResourceTypesRequest
⊞ AddResourceTypesResponse
⊞ ArePropertiesDefinableRequest
⊞ ArePropertiesDefinableResponse
⊞ ArePropertiesDefinedRequest
⊞ ArePropertiesDefinedResponse
⊞ DataResourceUnavailableFault
⊞ DataResourceUnavailableFault
⊞ DestroyDataResourceRequest
⊞ DestroyDataResourceResponse
⊞ GenericQueryRequest
```

# Current Status

## Profile 0

### Summary

| Data Resource | Interface Name | Category | Properties | | | Messages | |
|---|---|---|---|---|---|---|---|
| | | | *Static* | *Confi.* | *Total* | *Direct Access* | *Indirect Access* |
| **Repository Collection** | **RepositoryCollectionDescription** | Description | 2 | 0 | 2 | 0 | 0 |
| | **RepositoryCollectionAccess** | Access | 0 | 0 | 0 | 4 | 0 |
| | **RepositoryCollectionFactory** | Factory | 0 | 0 | 0 | 0 | 1 |
| **Repository** | **RepositoryDescription** | Description | 1 | 0 | 1 | 0 | 0 |
| | **RepositoryAccess** | Access | 0 | 0 | 0 | 6 | 0 |
| | **RepositoryFactory** | Factory | 0 | 0 | 0 | 0 | 1 |
| **Resource** | **ResourceDescription** | Description | 2 | 0 | 2 | 0 | 0 |
| | **ResourceAccess** | Access | 0 | 0 | 0 | 13 | 0 |
| **TOTAL** | | | 5 | 0 | 5 | 23 | 2 |

**21**

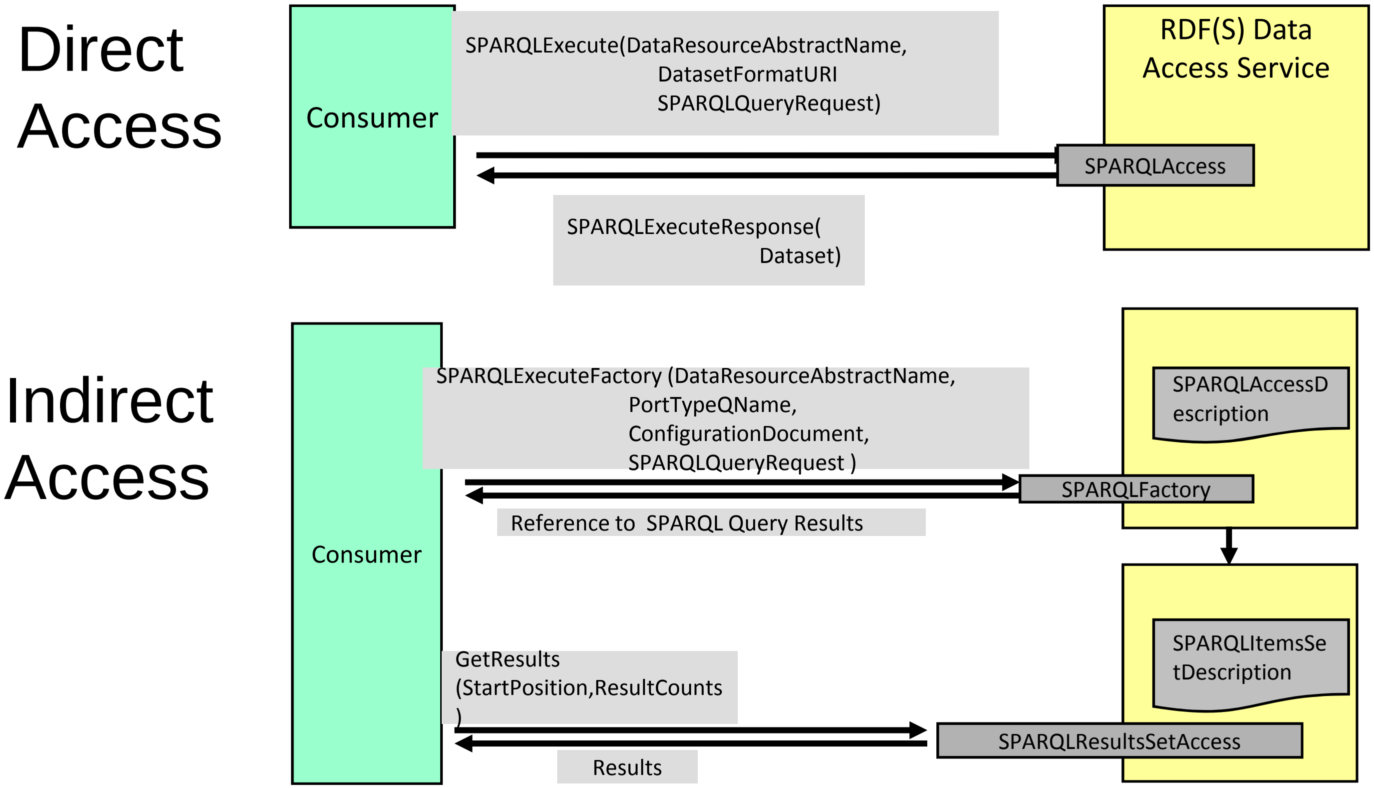
**2**

# WS-DAI RDF(S) Querying Specification Discussions

OGF24 DAIS Working Group

# Querying Specification Overview

- Direct and Indirect Access for RDF data using SPARQL
  - Based on the WS-DAI core model



Direct Access
Consumer
SPARQLExecute(DataResourceAbstractName, DatasetFormatURI SPARQLQueryRequest)
RDF(S) Data Access Service
SPARQLAccess
SPARQLExecuteResponse( Dataset)
Indirect Access
Consumer
SPARQLExecuteFactory (DataResourceAbstractName, PortTypeQName, ConfigurationDocument, SPARQLQueryRequest )
SPARQLAccessDescription
SPARQLFactory
Reference to SPARQL Query Results
SPARQLItemsSetDescription
GetResults (StartPosition,ResultCounts )
SPARQLResultsSetAccess
Results

# Querying Specification Status

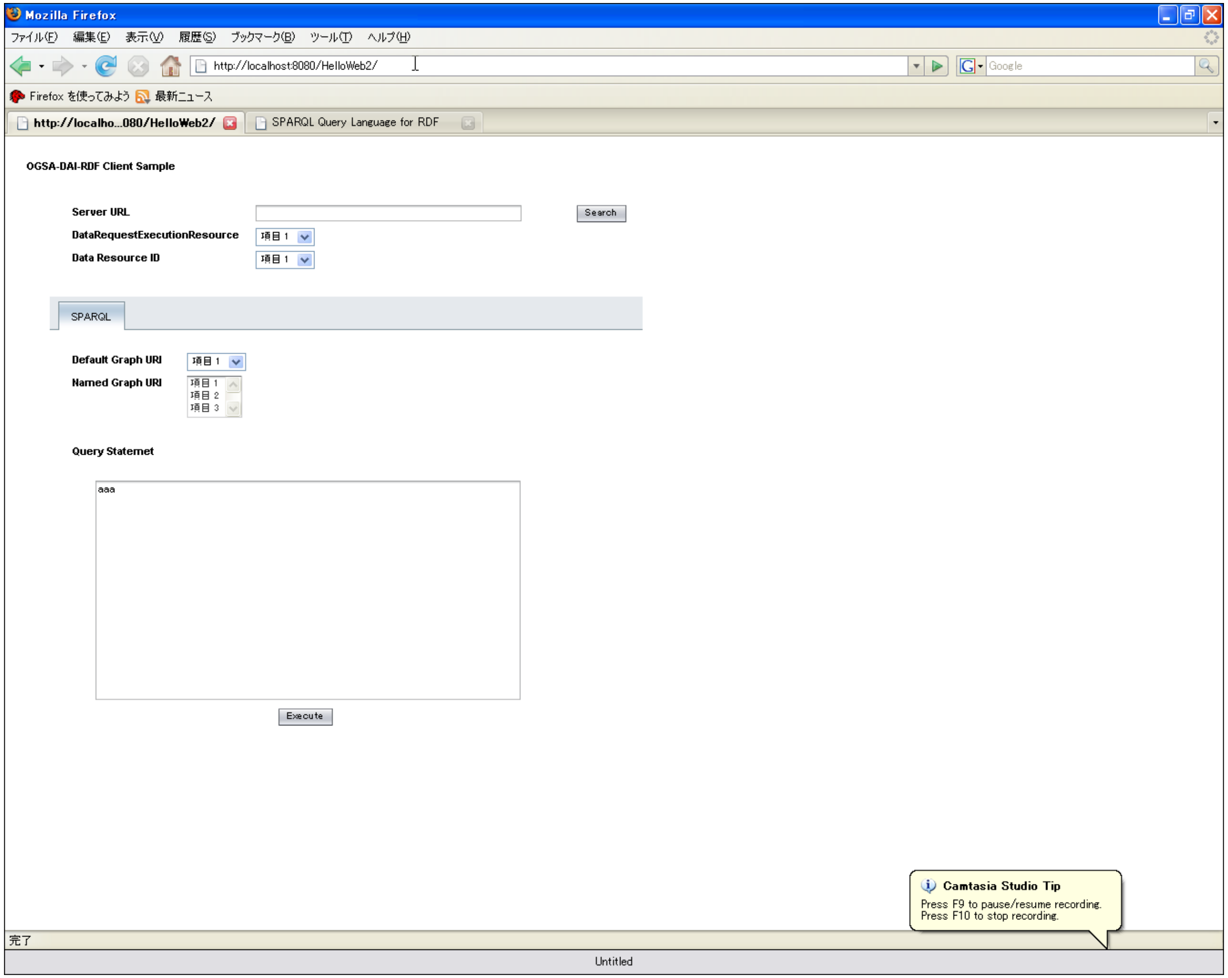
- Slightly modified
  - Based on the agreed terms.
  - Most of the current doc is already stable.
    - No serious impact

  AIST Implementation Started
    - *Based on our OGSA-DAI-RDF*
  - To support core WS-DAI
  - To validate WSDLs in the spec document
-

# Implementation: OGSA-DAI RDF

- A Set of RDF Processing Activities & Utilities
  - **V1.0 (DAI2.2 based) is available**
    - http://dbgrid.org/OGSA-DAI-RDF
  - **V2.0 (DAI3.0 based)**
    - Will be public as Technical Preview @ Grid2008
    - **Currently Used for limited applications**
      - **Monitoring database of S-MDS**
        - Records history of monitoring data with OWL
        - Rule based statistical Processing
      - **Service Repository of AIST Semantic SOA**
        - Service Workflow is represented with RDF and stored.
      - **Federated SPARQL prototype**

# OGSA-DAI-RDF:
## Servlet-based GUI in development


Mozilla Firefox
ファイル(F) 編集(E) 表示(V) 履歴(S) ブックマーク(B) ツール(T) ヘルプ(H)
http://localhost:8080/HelloWeb2/
Google
Firefox を使ってみよう
最新ニュース
http://localho...080/HelloWeb2/
SPARQL Query Language for RDF
完了
Untitled

OGSA-DAI-RDF Client Sample

Server URL [Search]

DataRequestExecutionResource 項目 1

Data Resource ID 項目 1

SPARQL

Default Graph URI 項目 1

Named Graph URI 項目 1<br>項目 2<br>項目 3

Query Statemet

aaa

[Execute]

Camtasia Studio Tip
Press F9 to pause/resume recording.
Press F10 to stop recording.

# WS-DAI Implementation

- **RDF(S) specific interfaces are implemented**
  - WS-DAIRDF(S) Querying  with AXIS2
  - Completed except the factory pattern

- **Planned to demonstrate at SC08**
  - WS-DAI core
  - WS-DAIRDF(S)  Ontology level 0

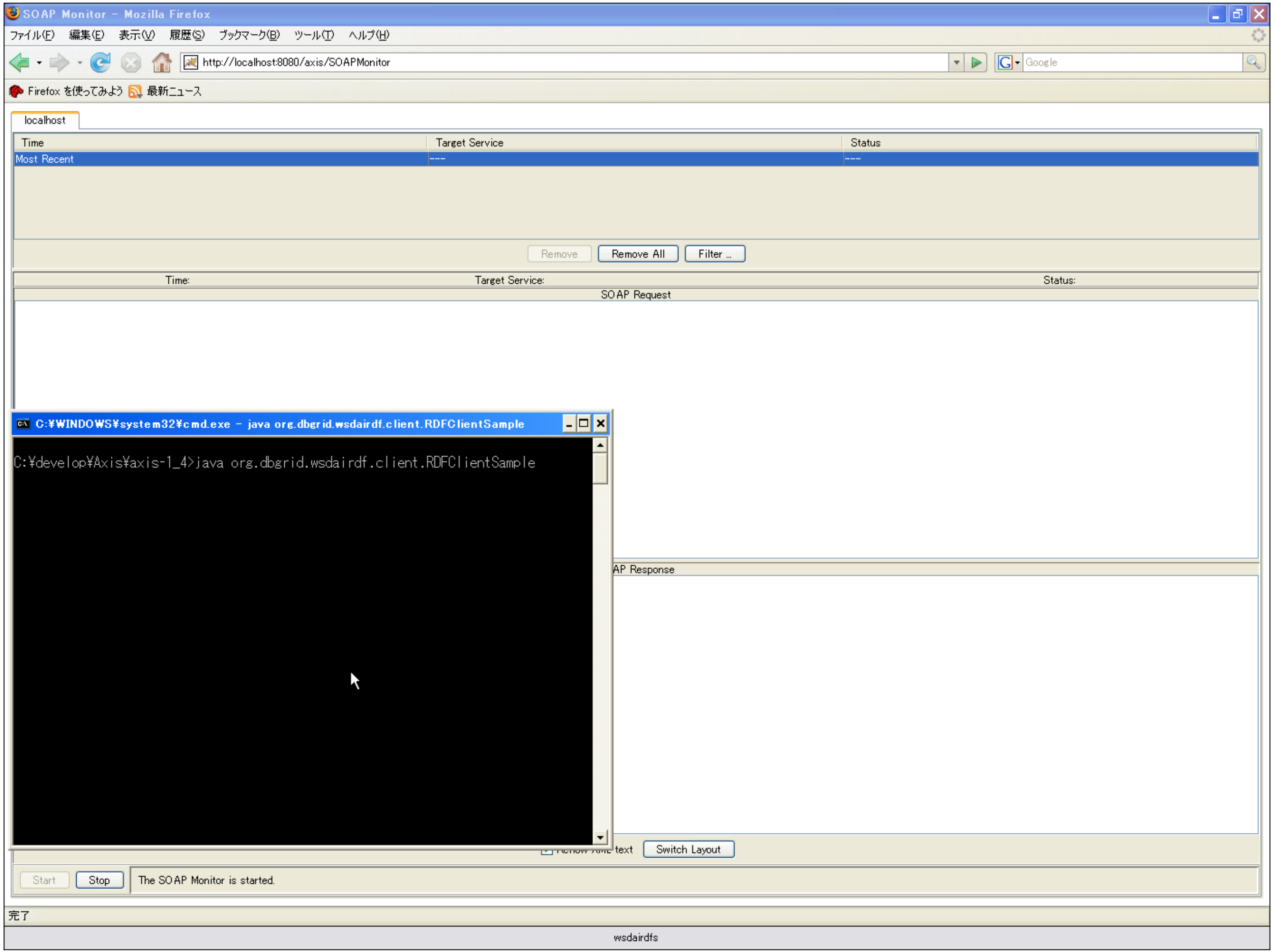
- **Currently No WS-DAI core support**
  - Our implementation is also based on OGSA-DAI3.0
  - Q: our implementation can be  independent one with OGSA-DAI team?

    (Yes) Will create yet another WS-DAI core implementation

    Or :

    (No) Reuse OGSA-DAI based WS-DAI(X) codes

# WS-DAIRDF(S) SparqlQueryStatement Example


SOAP Monitor - Mozilla Firefox
ファイル(F) 編集(E) 表示(V) 履歴(S) ブックマーク(B) ツール(T) ヘルプ(H)
http://localhost:8080/axis/SOAPMonitor
Google
Firefox を使ってみよう
最新ニュース
localhost
Time
Target Service
Status
Most Recent
---
---
Remove
Remove All
Filter ...
Time:
Target Service:
Status:
SOAP Request
AP Response
text
Switch Layout
Start
Stop
The SOAP Monitor is started.
完了
wsdairdfs
C:¥WINDOWS¥system32¥cmd.exe - java org.dbgrid.wsdairdf.client.RDFClientSample

```
C:¥develop¥Axis¥axis-1_4>java org.dbgrid.wsdairdf.client.RDFClientSample
```

# Discussion Issues on Querying Spec

# Querying Spec is almost completed:

- New issue raised in terminology discussion
  - Need to support of *Reasoning/Inference* functions
    - Discussions still within AIST

      Currently we have no good/agreed way to support it
  - Spec without reasoning

    ➔ Already stable & completed
  - Spec with reasoning

    ➔ Still need to discuss how to support it.

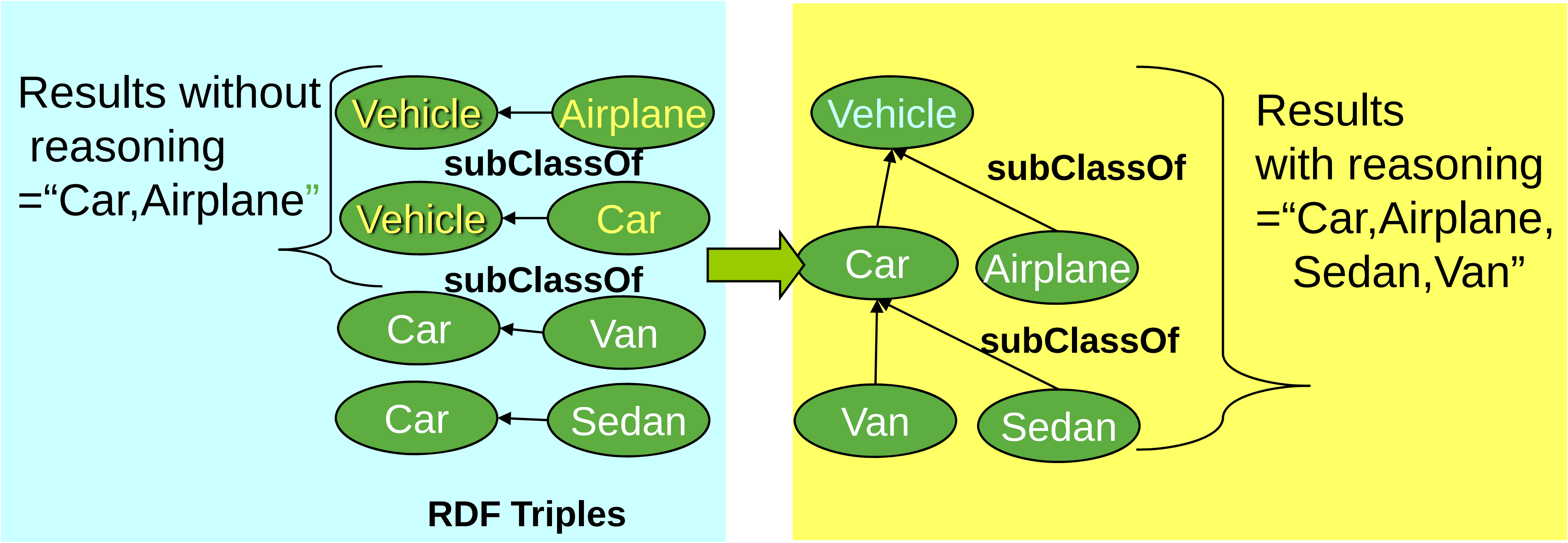
**Current approach:**

**Submit current spec as Profile0?**

**Put the reasoning function into Profile1**

# Reasoning/Inference: introductory example

## Query: Get the subclasses of “Vehicle”



Results without reasoning =“Car,Airplane”

Results with reasoning =“Car,Airplane, Sedan,Van”

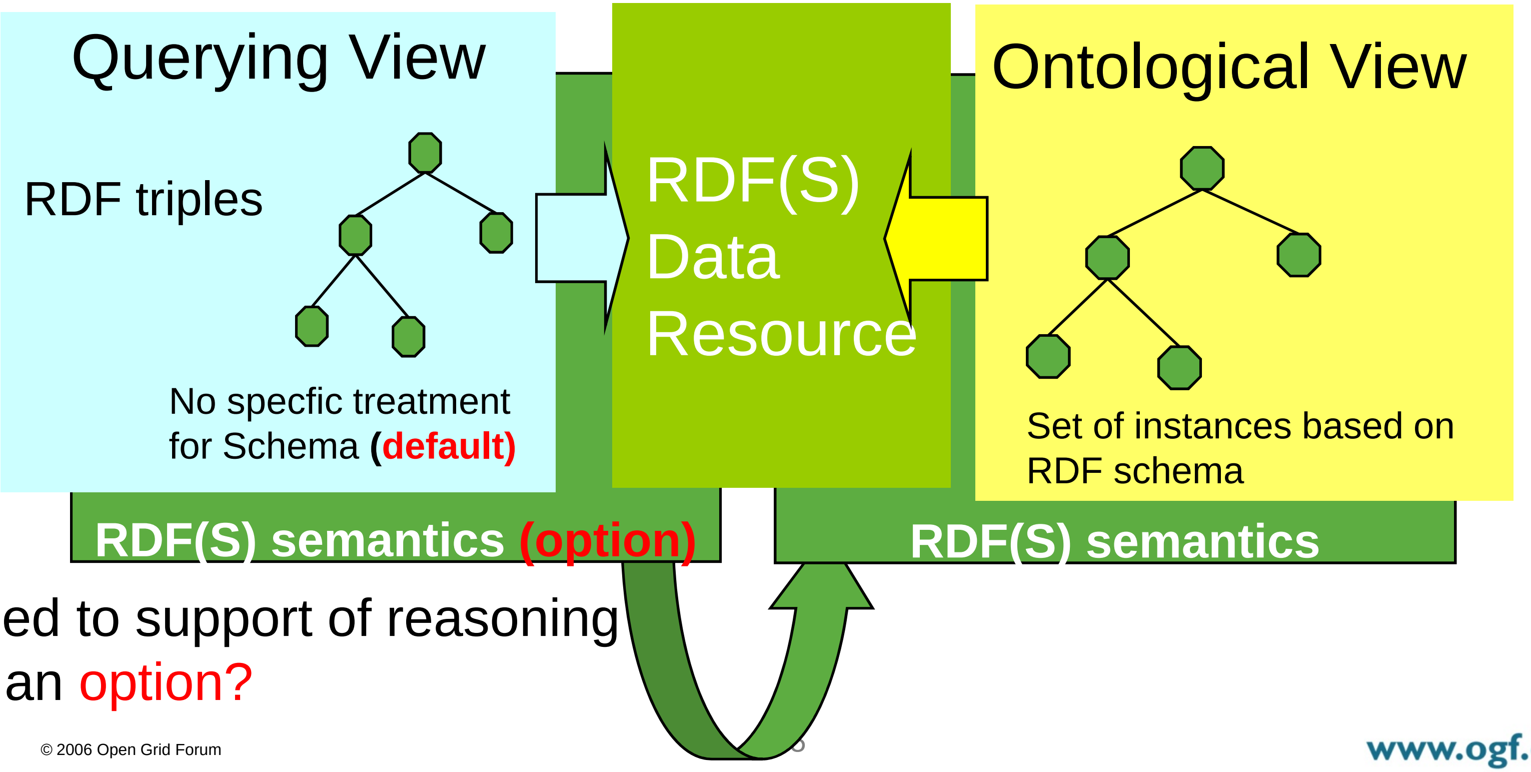
Reasoning is done by using some specific semantic model (**RDFS**, OWL-lite/DL/full,etc)

# Our RDF(S) Data Resource

- Querying = SPARQL is RDF triple matching/constructing language which focus on instances. (without semantics as default)
- Ontology= based on RDF(S) semantics .

These 2 specs share the same definition of RDF(S) data resource

Difficult to use both in one application?



Querying View

RDF triples

No specfic treatment for Schema **(default)**

**RDF(S) semantics (option)**

RDF(S) Data Resource

Ontological View

Set of instances based on RDF schema

**RDF(S) semantics**

Need to support of reasoning as an option?

# Current Status

- W3C SPARQL and Related Standards.
  - *No support of reasoning*
    - SPARQL = Triple matching/constructing language which focus on RDF triples
    - Reasoning is out of focus of the querying standard.
- Existing Software Products
  - *Most of them supports reasoning*
    - In various ways

# Issue

- Need of Reasoning?
  - **very** useful
    - One querying usecase assumes the reasoning
    - Our OGSA-DAI-RDF already supports reasoning…
  - Originally out of focus of SPARQL standards
    - Might collide with future W3C activity.
    - We left the update function for the same reason.…
- How?

# Supporting Reasoning Functions with SPARQL Query Processing

1. Create inferred graph explicitly
   - *Ex. Jena (,  current version of our OGSA-DAI-RDF)*
     - Create a yet another graph as a result of the reasoning
     - Issue a query to the created graph.

2. Reasoning flag/switch when querying
   - *Ex. Allegrograph*
     - Query processing engine supports reasoning option.
     - If the switch=true, the query processing is done with reasoning

3. Configure the resource whether it needs the reasoning or not
   - *Ex. Sesame2*
     - Specify whether the resource uses reasoning or not when creating/configuring  a resource.
     - The query to the resource is always done with reasoning when the resource is configured to do so.

# Current Possible Solutions

1. Define separate SPARQL interface
    - ex. SPARQLQuerywithRDF(S)reasoning
    1. No need to modify the current spec.
    2. Users should know different querying interface specs.
    3. Possible extension to have future SPARQLQuerywithOWLreasoning….?

2. Extend the current SPARQL interface
    - Add a parameter for the reasoning switch(true/false)
    1. Need to extend the current spec with adding parameters
    2. Will be difficult to have future extensions for other reasoning functions
        - It is very difficult for defining the parameter which supports various reasoning

3. Define the resource with configurable parameter
    - ex. Configurable description of the resource to support RDF(S) reasoning
    1. No need to modify the current spec.
    2. Seems to be an elegant solution
    3. Fixed type of reasoning is always done for the resource

    = User could not specify whether he needs to perform the reasoning or not.

**User of the current system wants to specify whether the query uses reasoning or not.**

**(Sesame has the switch to include the inferred result……)**

# Issue

- Need of Reasoning?
  - **very** useful
    - One querying usecase assumes the reasoning
    - Our OGSA-DAI-RDF already supports reasoning
  - Originally out of focus of SPARQL standards
    - We left the update function for the same reason….
    - Might cause future batting with W3C activity.

- How? – No good way at now.

1. Additional new query interface
2. Add parameters
3. Add Configurable Description

Etc.

- Put this function to Profile1?
- Need to have Usecases?
  - Which combines querying and ontology

# Discussions

## DAIS Working Group