

## DAIS Data Services Day

Savas Parastatidis

## DISCLAIMER!!!



The specifications were just released. I only had a quick chance to look at them.

Also...

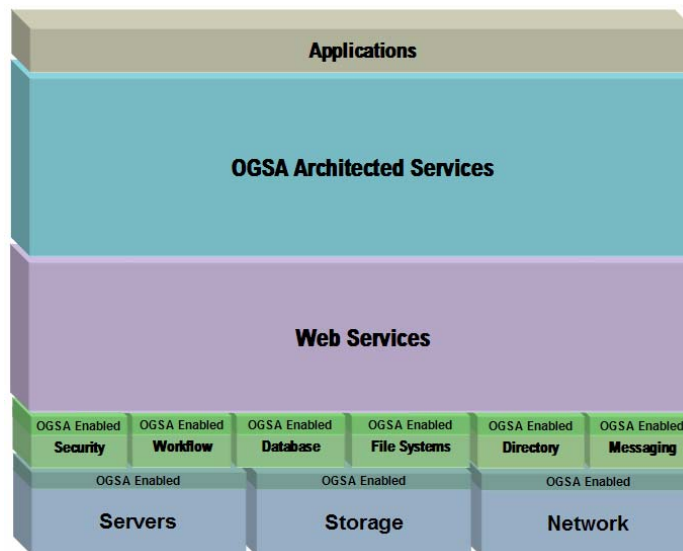
I prepared these slides during the last 3 hours.



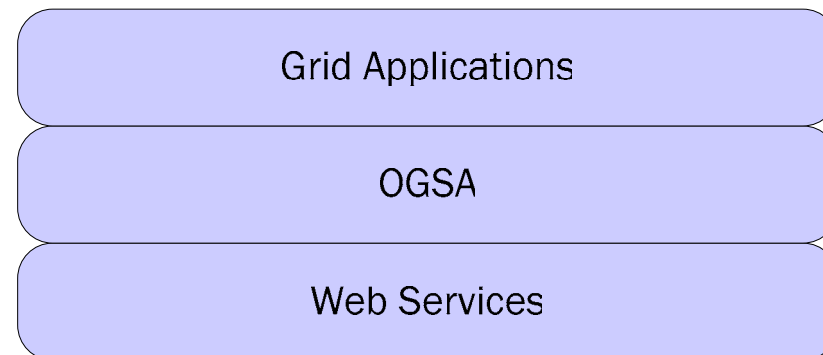
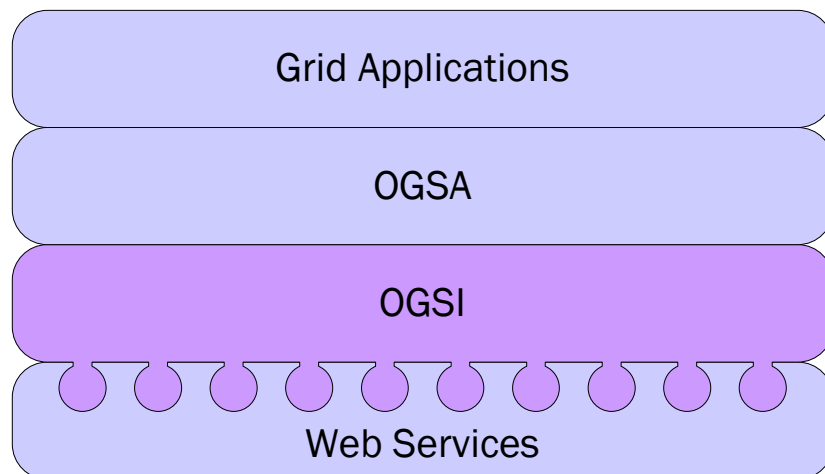
- Initial impressions on WS-Resource
- Web services vs distributed objects
- Grid Applications – Requirements (true for DAIS too)
  - The WS-GAF approach to building Grid applications
- DAIS choices
- Conclusions
- Future

- A clear separation between the terms “service” and “resource”
- Services are not dynamically created; they are deployed
- Services provide operations on multiple resources (one-to-many; no implicit one-to-one association between the two)
- Factorisation on the functionality into separate specs (not all the way though)
- A document-based approach to resource properties
- WSA-friendly specs (no more GWSDL)
- Respect to existing tooling; use it without modifications
- Still issues with the conceptual model (scalability, loose coupling, is there an actual need for it... use cases?)
- Please refer to our August 2003  
<http://www.neresc.ac.uk/ws-gaf>

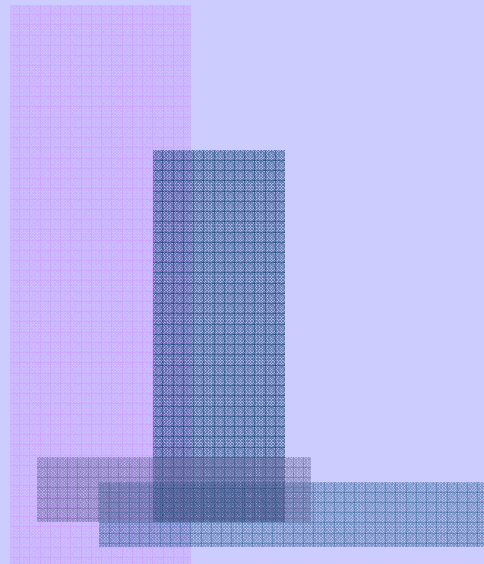
# Initial impressions



From our GGF9 presentation:



## Web services vs distributed objects



- Built around the concepts of service and message
- A service may be defined as *a logical manifestation of some physical resources (like databases, programs, devices, or humans) that an organization exposes to the network* and
- *Services interaction are facilitated by exchanging messages*
- A service adheres to a contract
  - Describes the format of the messages exchanged
  - Defines the message exchange patterns in which a service is prepared to participate

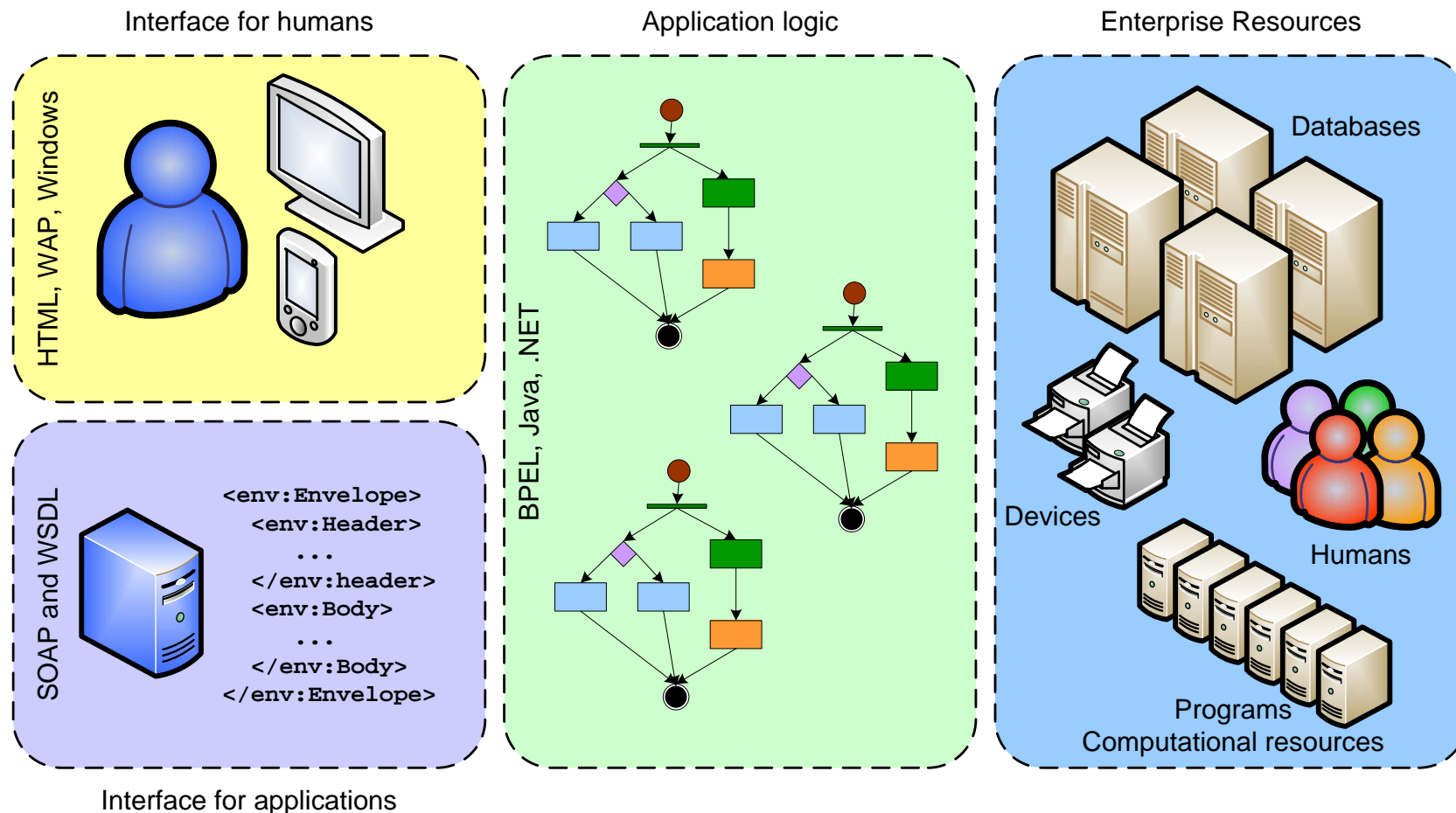
- Don Box's four tenets about Service Orientation
  - Boundaries are explicit
  - Services are autonomous
  - Services share schema and contract, not class
  - Service compatibility is determined based on policy

Source: "A Guide to Developing and Running Connected Systems with Indigo"  
<http://msdn.microsoft.com/Longhorn/understanding/mag/default.aspx?pull=/msdnmag/issues/04/01/Indigo/default.aspx>  
and various talks

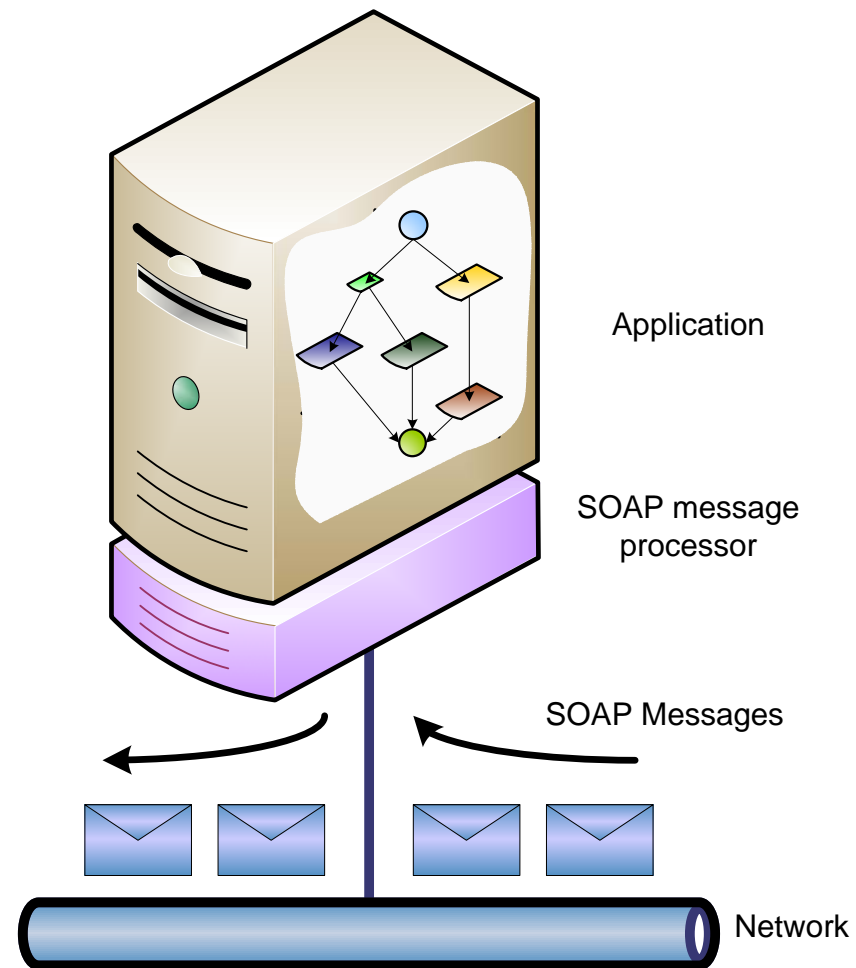
# The Anatomy of a Web Service



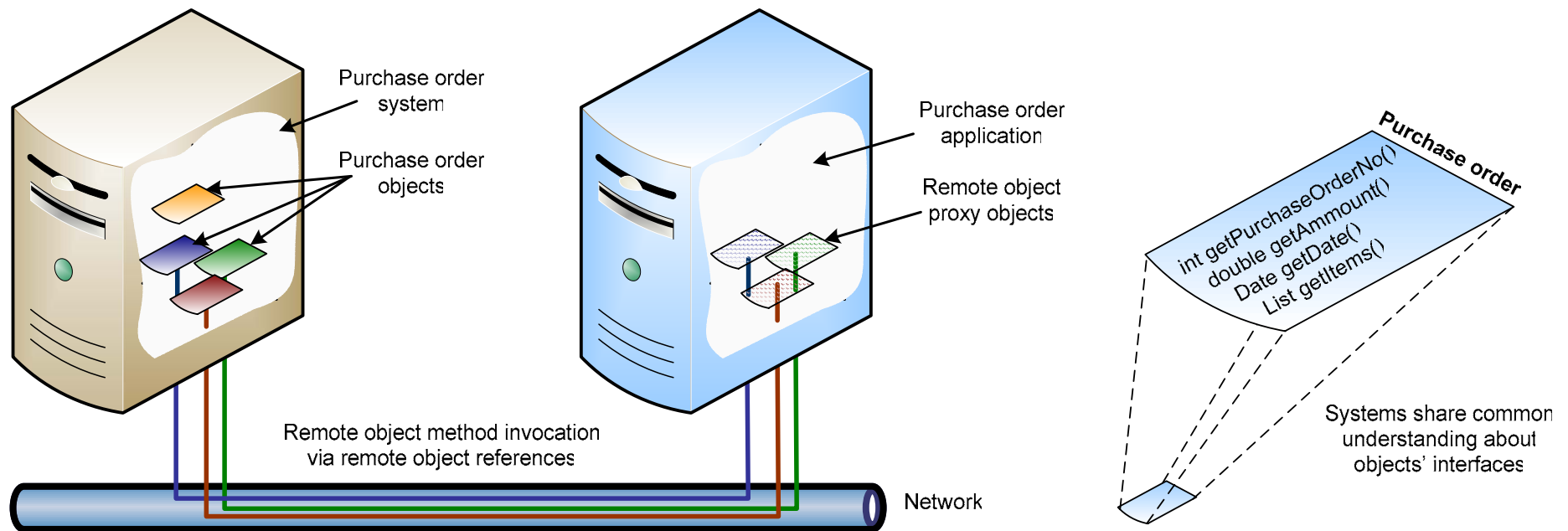
- Large grained, loosely coupled
  - Performance, scalability, maintenance, re-use, etc.



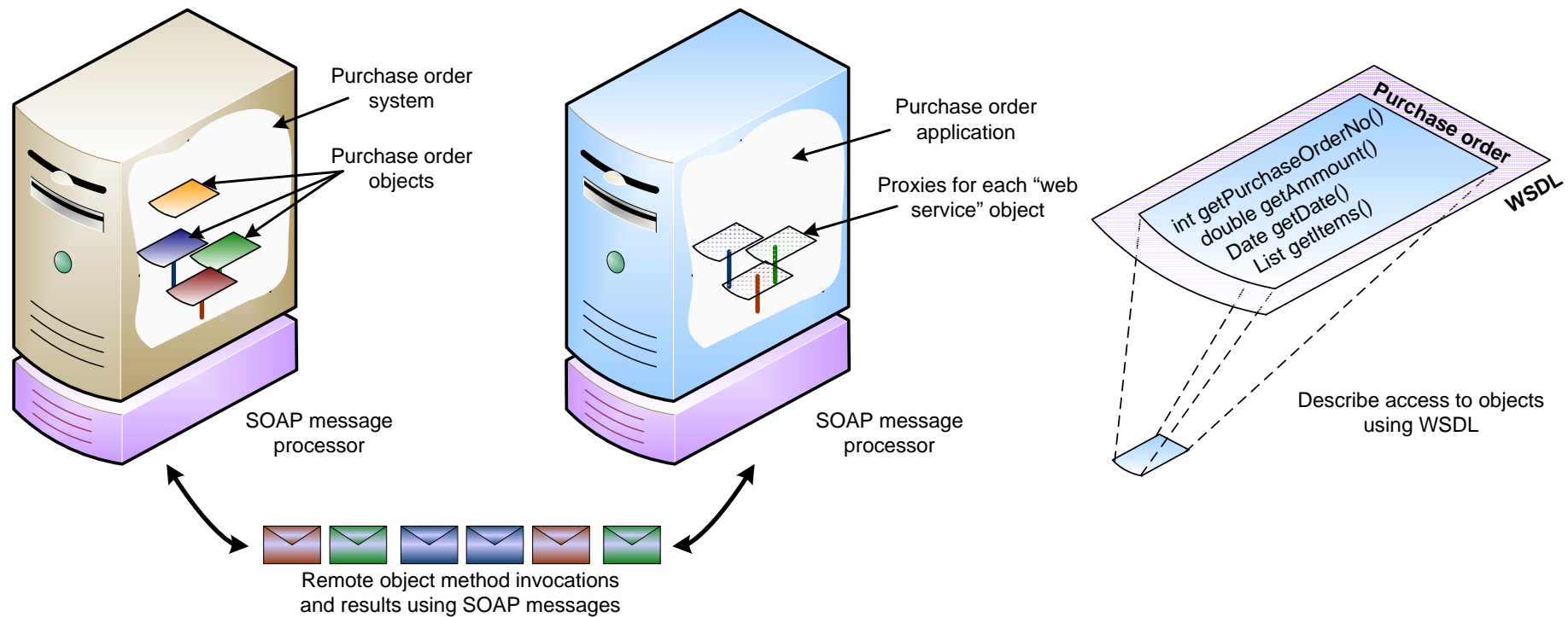
# A Web Service

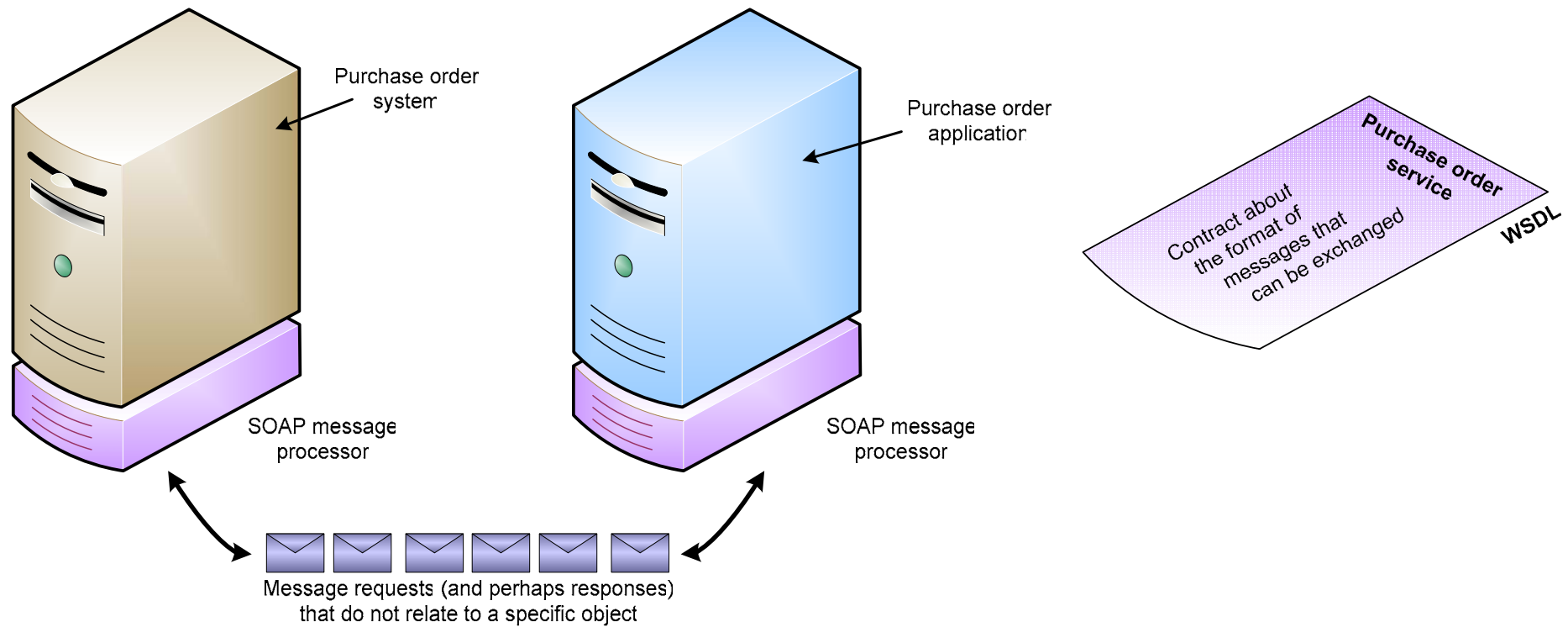


# Distributed objects

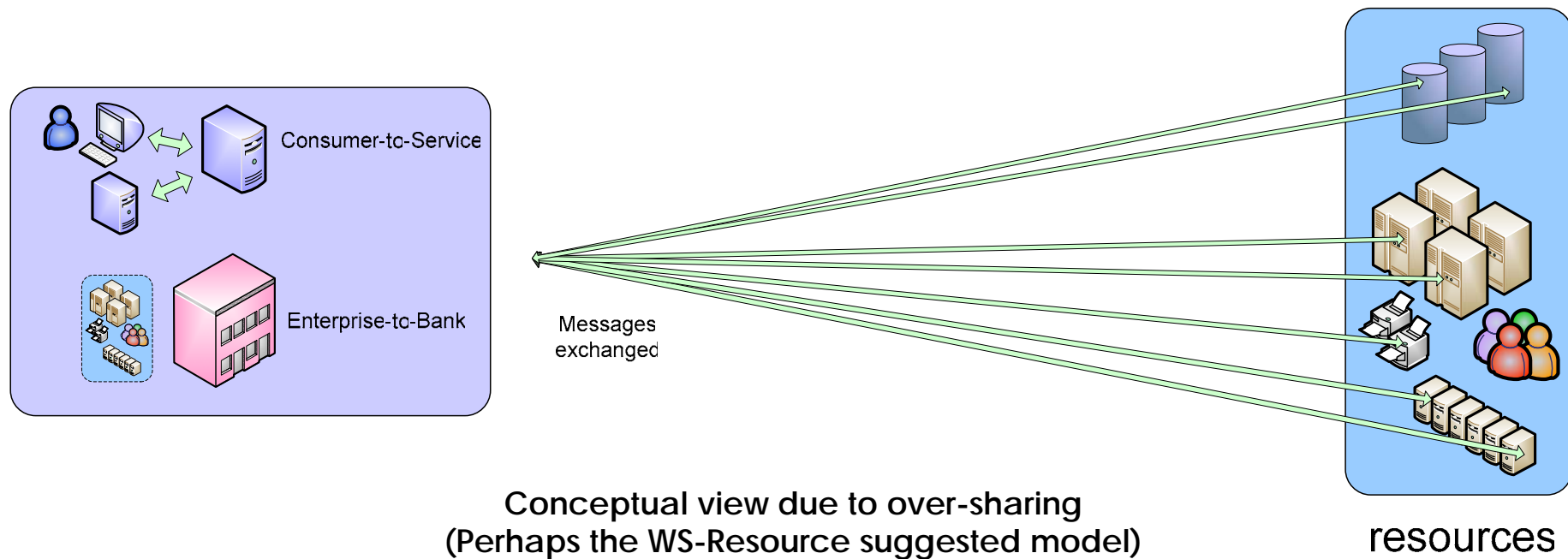


# Distributed objects using SOAP and WSDL



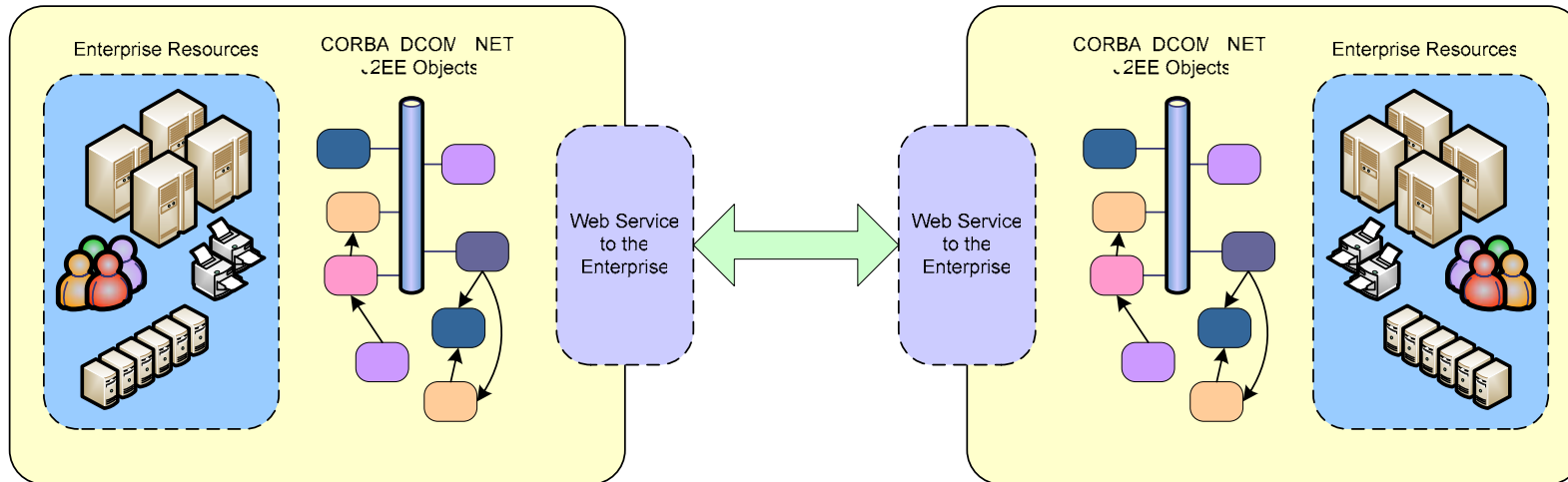


- Tight-coupling
- Easily breakable applications
- Poor scalability

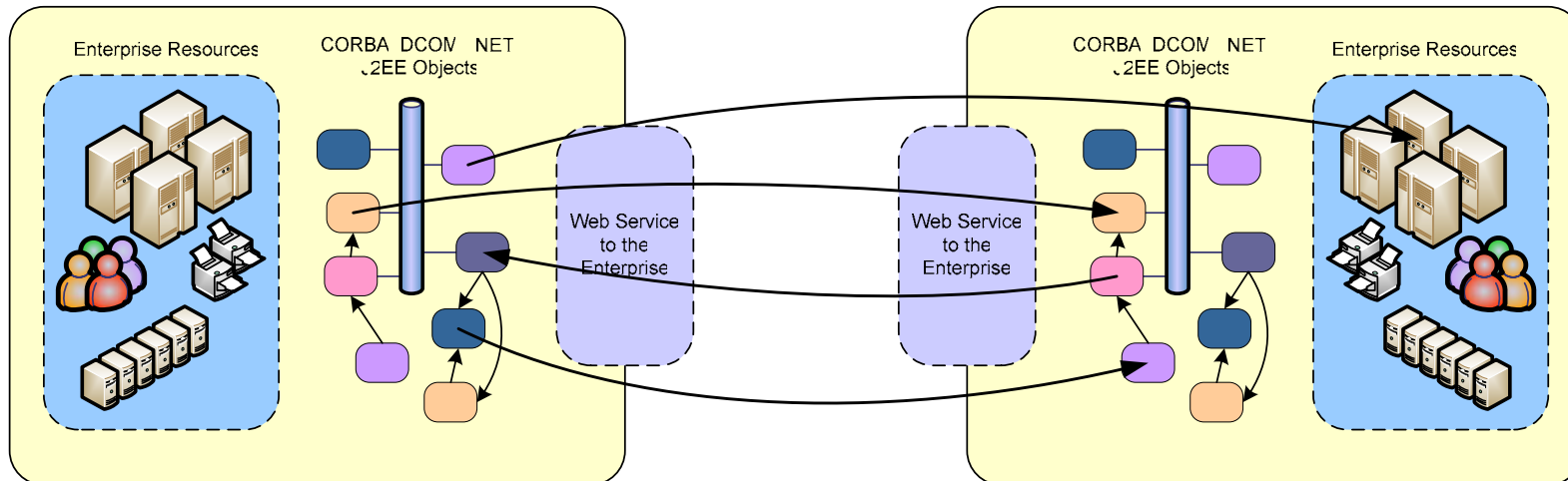


# Talking directly to resources

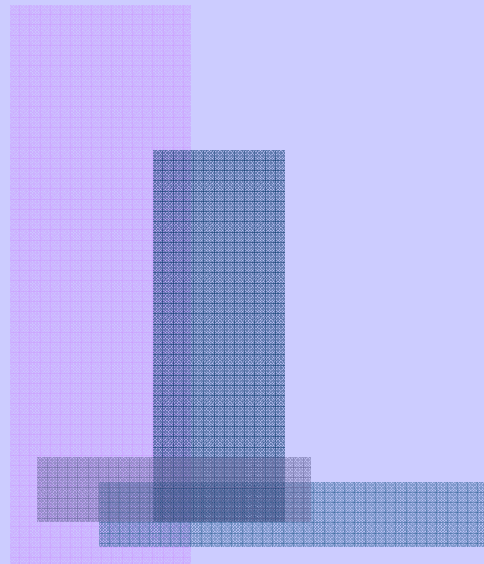
SOA



WS-Resource



## Grid applications



- Two types of state
  - State internal to a service (we can't cross service boundaries; we are not concerned with that)
  - Interaction state

- Stateful interactions
  - Contextualisation



WS-Context, WS-Security, WS-Transactions, WS-Coordination, BPEL (message correlation), etc. etc. etc.

- Resource identification



URN: Uniform **Resource** Names

- Metadata

- Grid Resource Specification (just an XML Schema document)

- Lifetime management of resources

- Just a high-level service interface
- Lifetime information
  - Part of the metadata



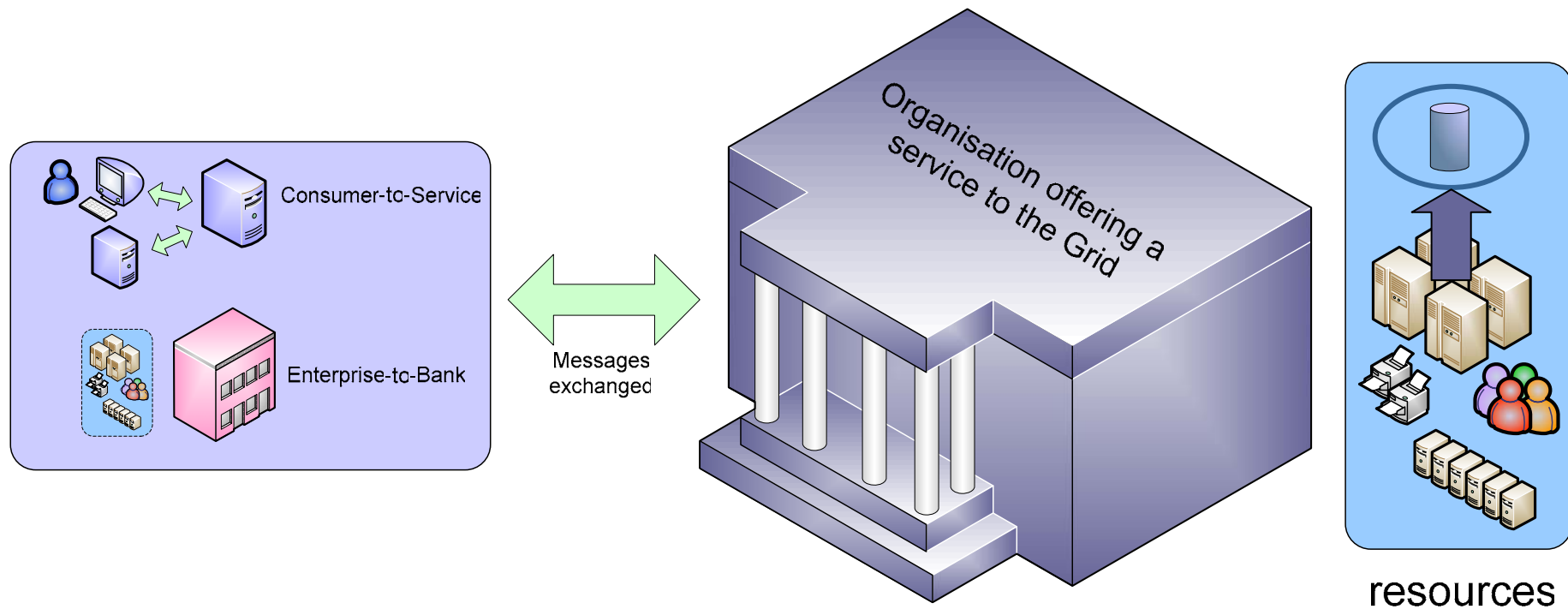
At the OGSA level

- There is a many-to-many relationship between resources and services
- If resources are exposed outside an organisation's boundaries there may be a need for
  - Ontologies
  - Relationships
  - Location information
  - Lifetime information
  - Ownership/access restrictions information
  - Provenance
  - etc.
- Please note that it's not the norm to expose resources outside the boundaries of an organisation
- **Metadata**

- Resources are usually hidden
- There are cases where resources need to be identifiable outside an organisation's boundaries

## Identify

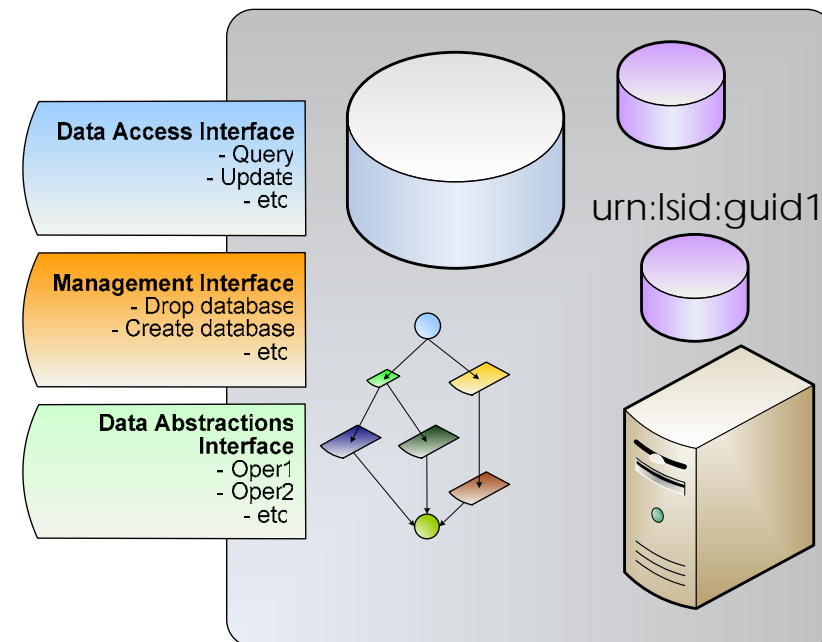
urn:dais:dataset:b4136aa4-2d11-42bd-aa61-8e8aa5223211



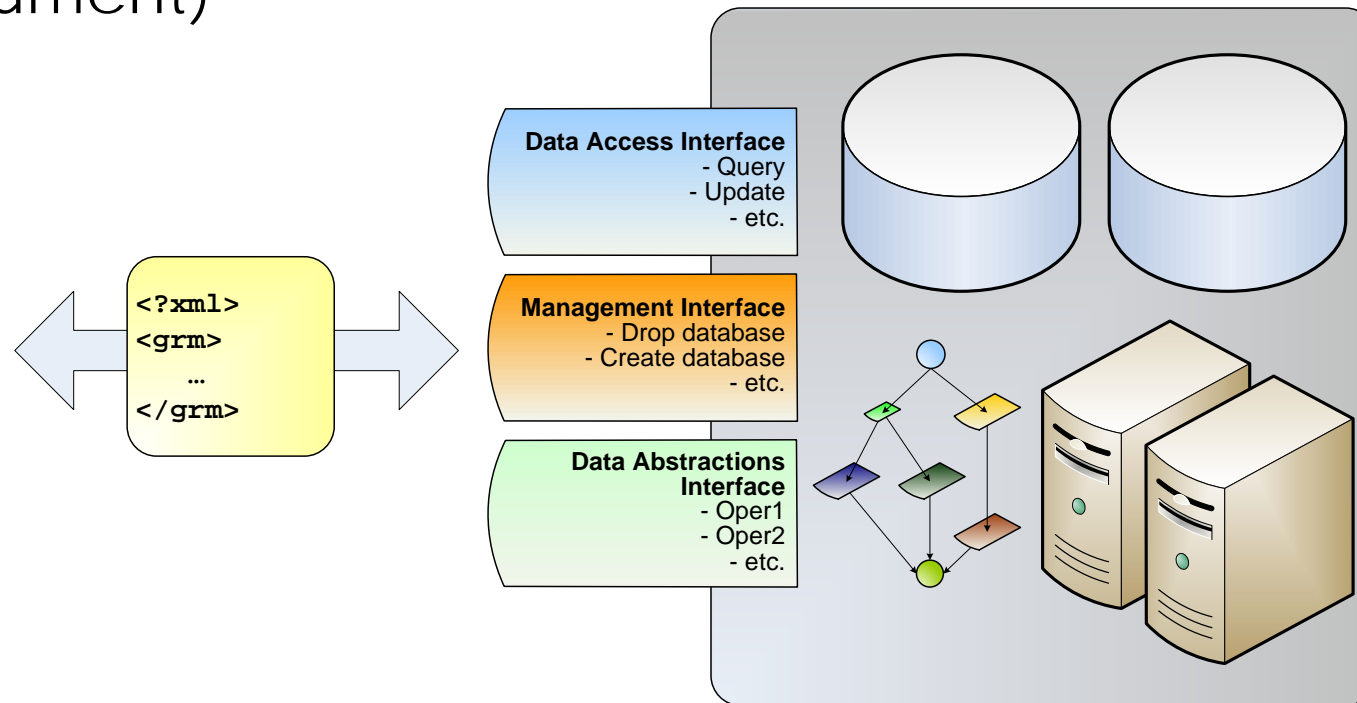
- Grid Resource Identifier (GRI) (like an LSID)
  - Everlasting, unique resource identifier (Uniform Resource Name, URN)
  - Can be stored in a database or printed in a journal
  - Decoupling of identity from interface

The resource is identified separately from the interface that can provide access to it

A service could be seen as a resource



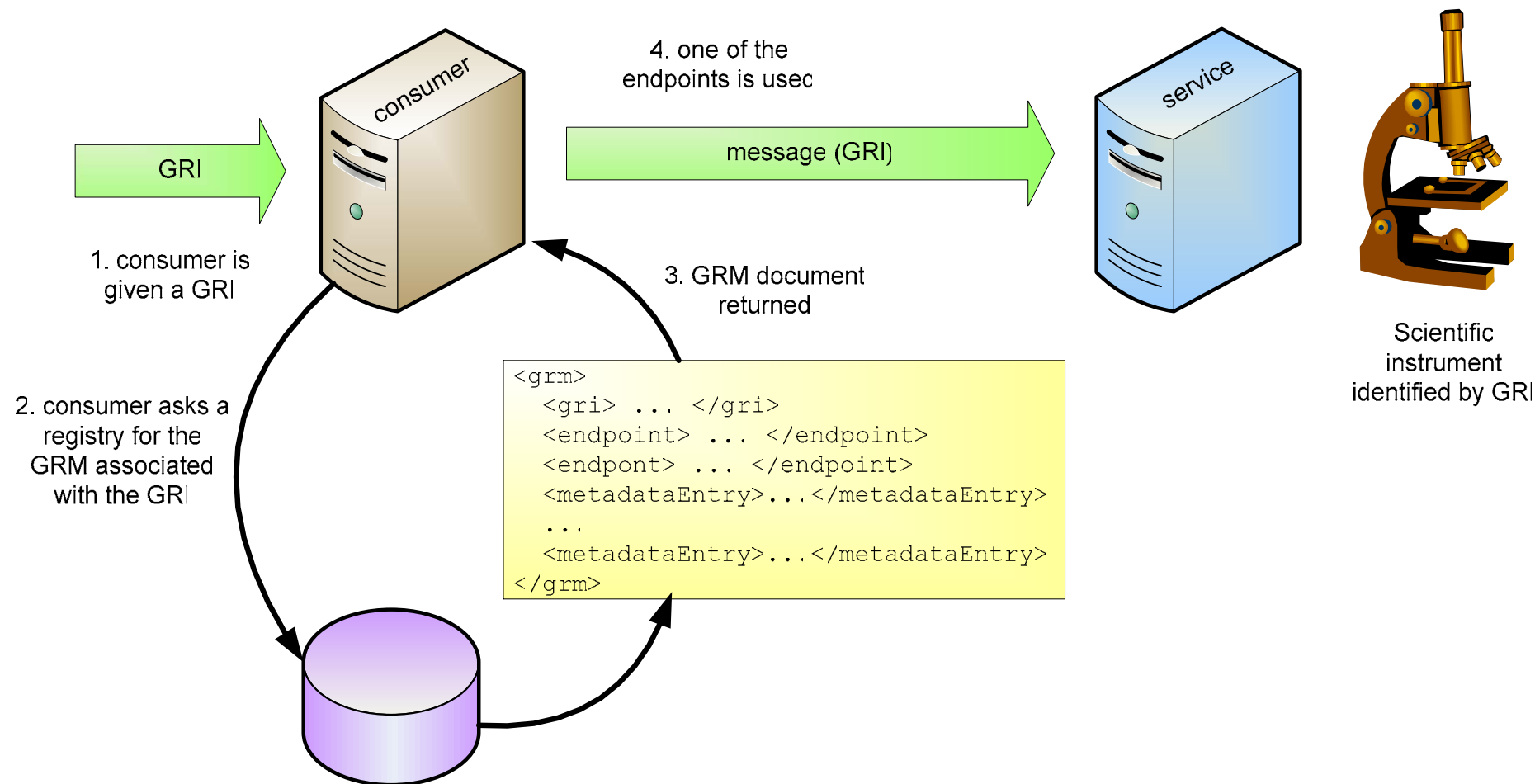
- Functionality equivalent to Service Data Elements
- Everything implemented using existing technologies and tooling
- Not Grid-technology specific (it's just an XML Schema document)



- Infrastructure does not need to be aware of the differences in metadata documents
  - Generic metadata Web services
  - Generic tools for Peer-to-Peer metadata propagation
  - Generic metadata registries
  - Databases
  - etc

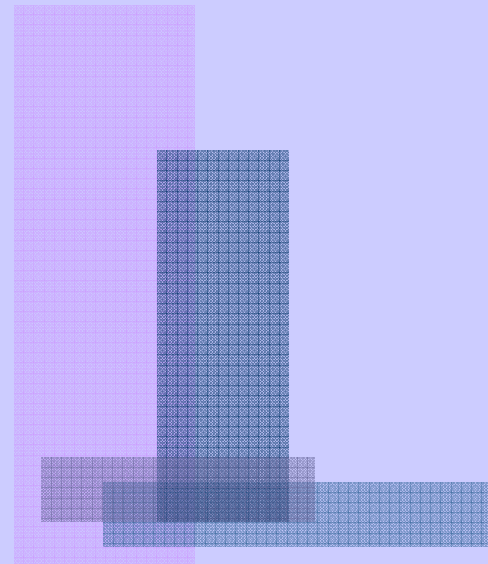
- Separate out and define orthogonal lifetimes
  - Grid Resource lifetime (for identified resources outside the organisation boundaries)
  - Grid Resource Metadata document lifetime
  - Endpoint lifetime
  - Metadata entry lifetime
- Context lifetime (for stateful interactions)

## Example: Using a registry

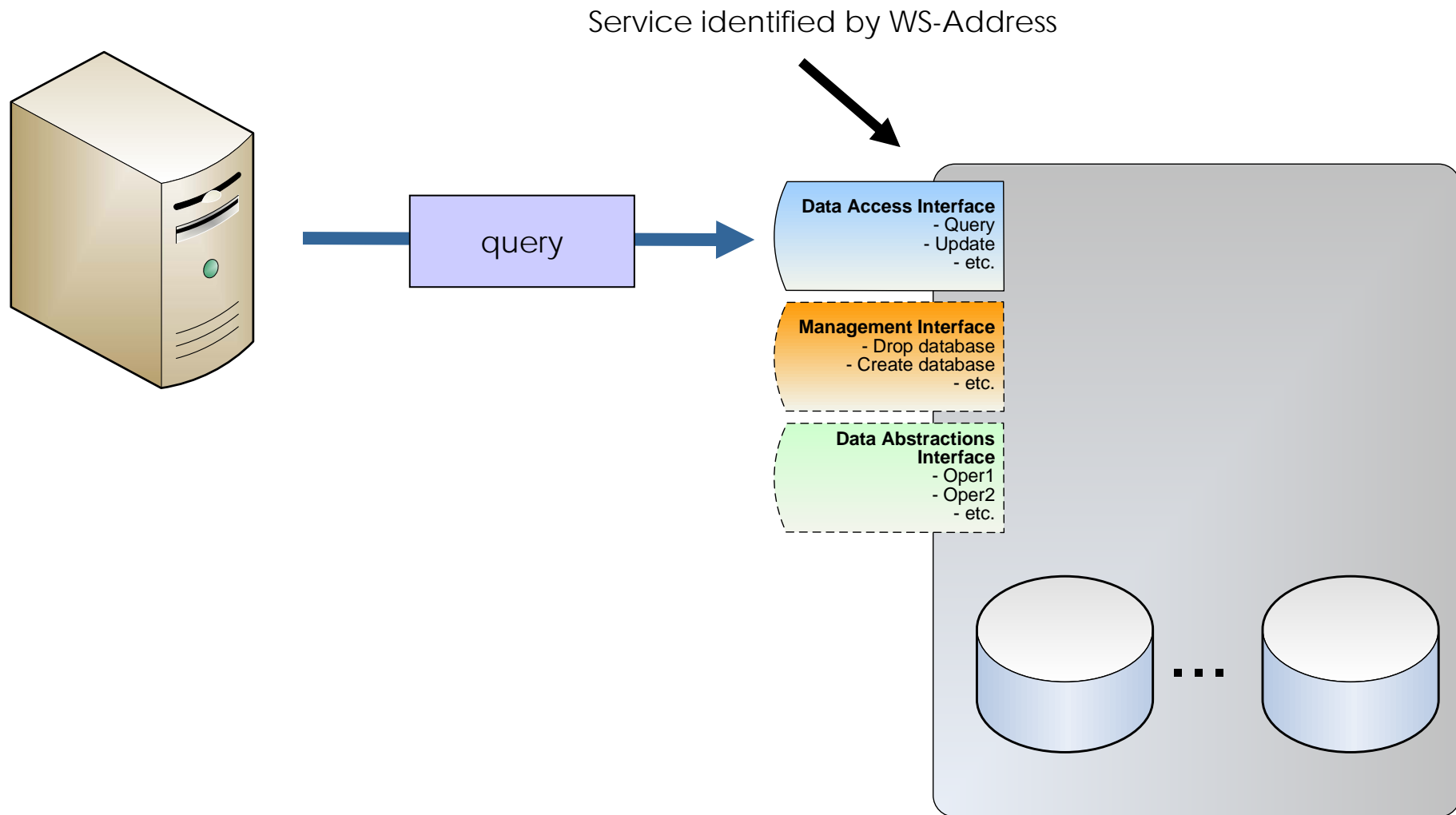


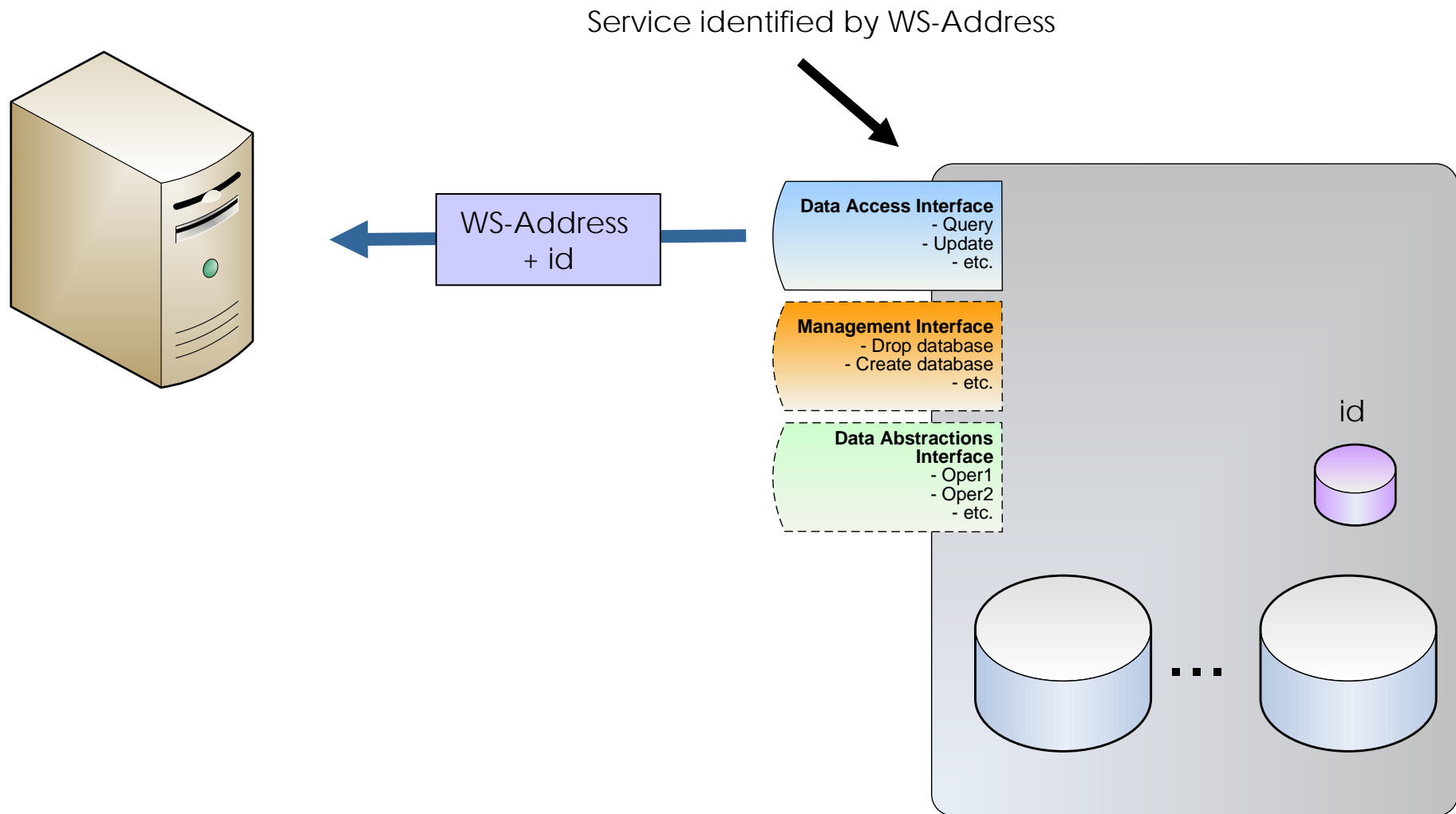
- Simplicity and Minimalism
  - Meets Grid requirements without inventing new infrastructure
  - Uses existing contextualisation and addressing specs
  - Uses URN for resource identification
  - Low entry and maintenance costs for new Grid services
- Distributed technology independence

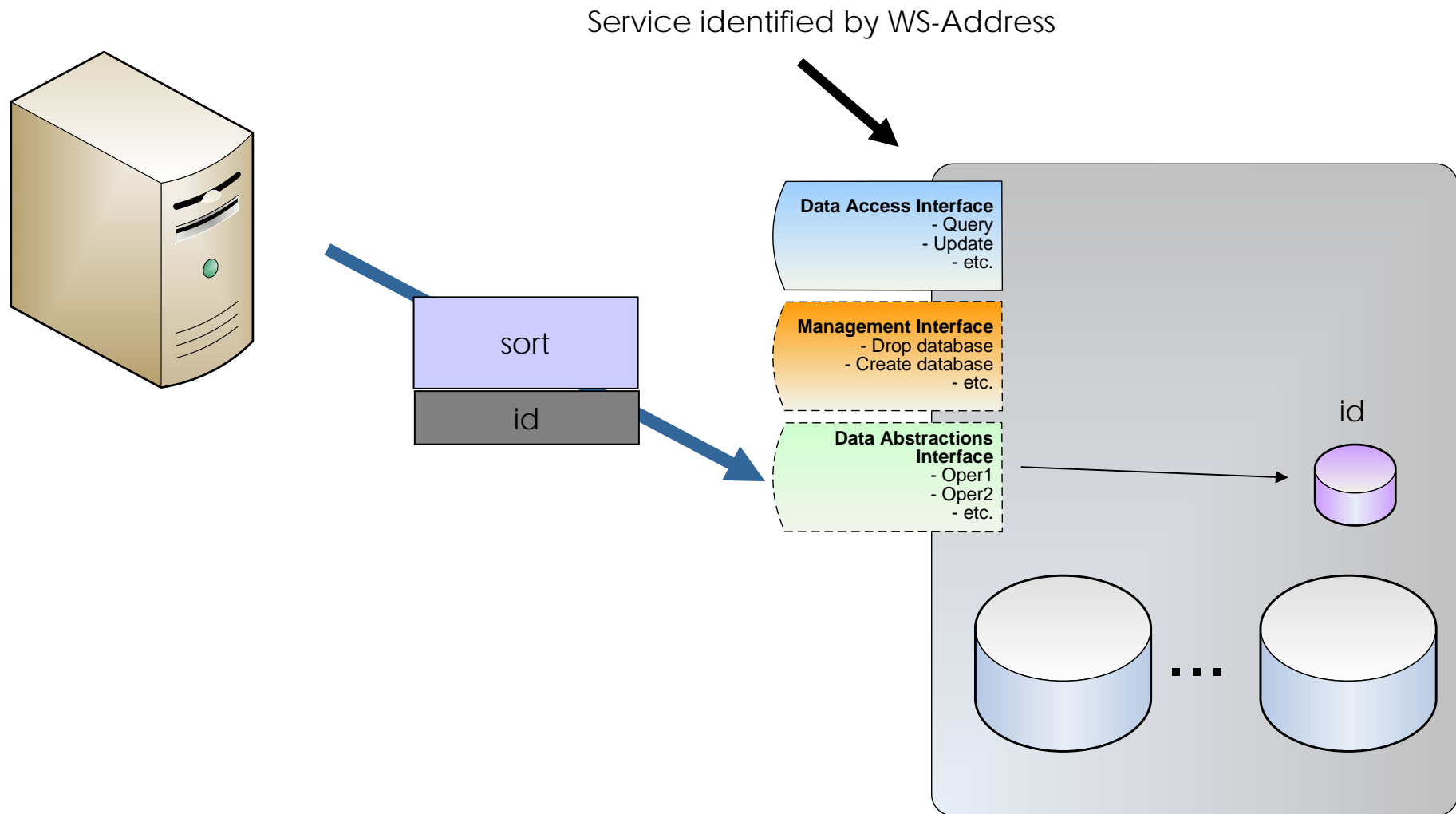
## DAIS

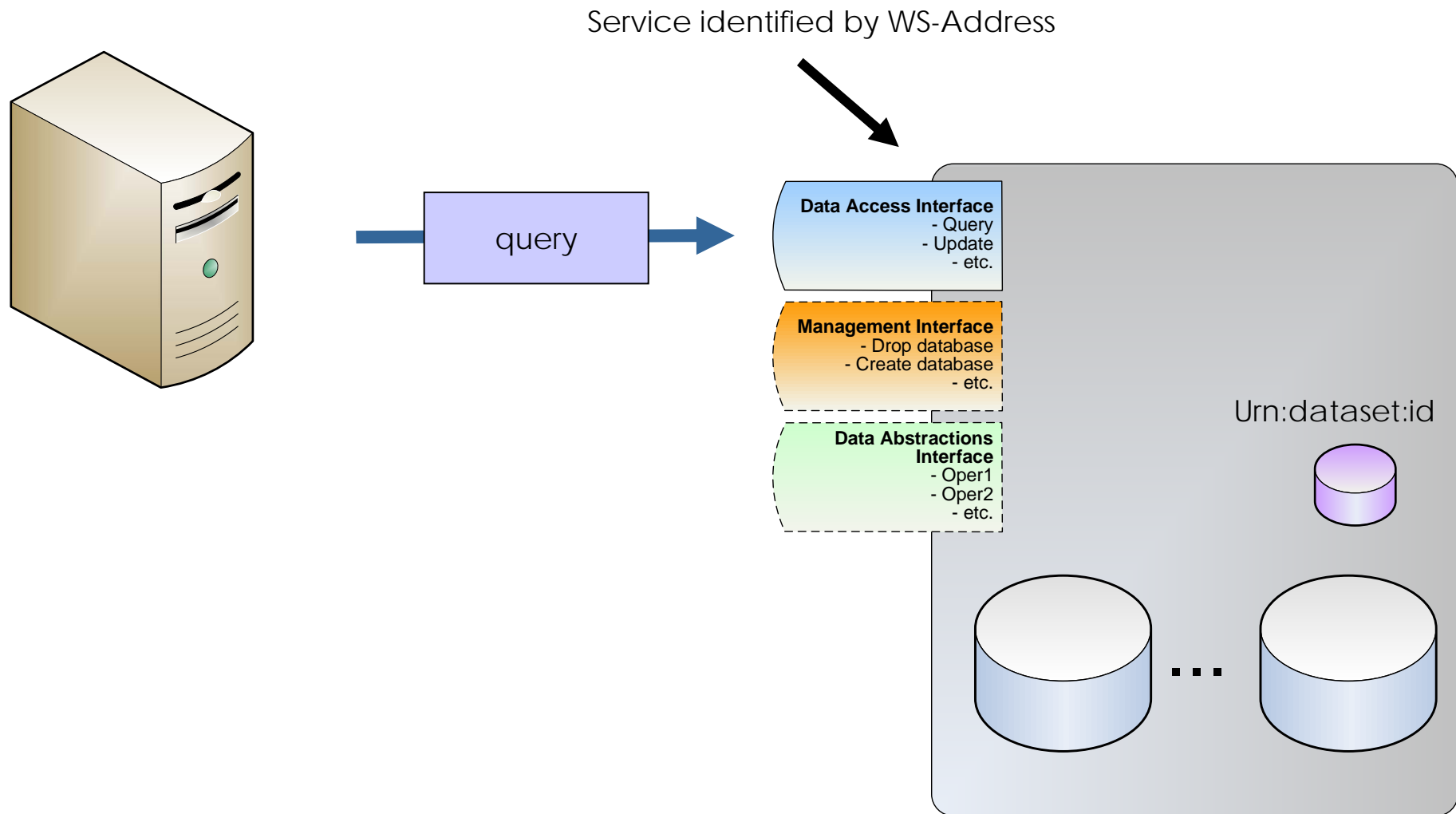


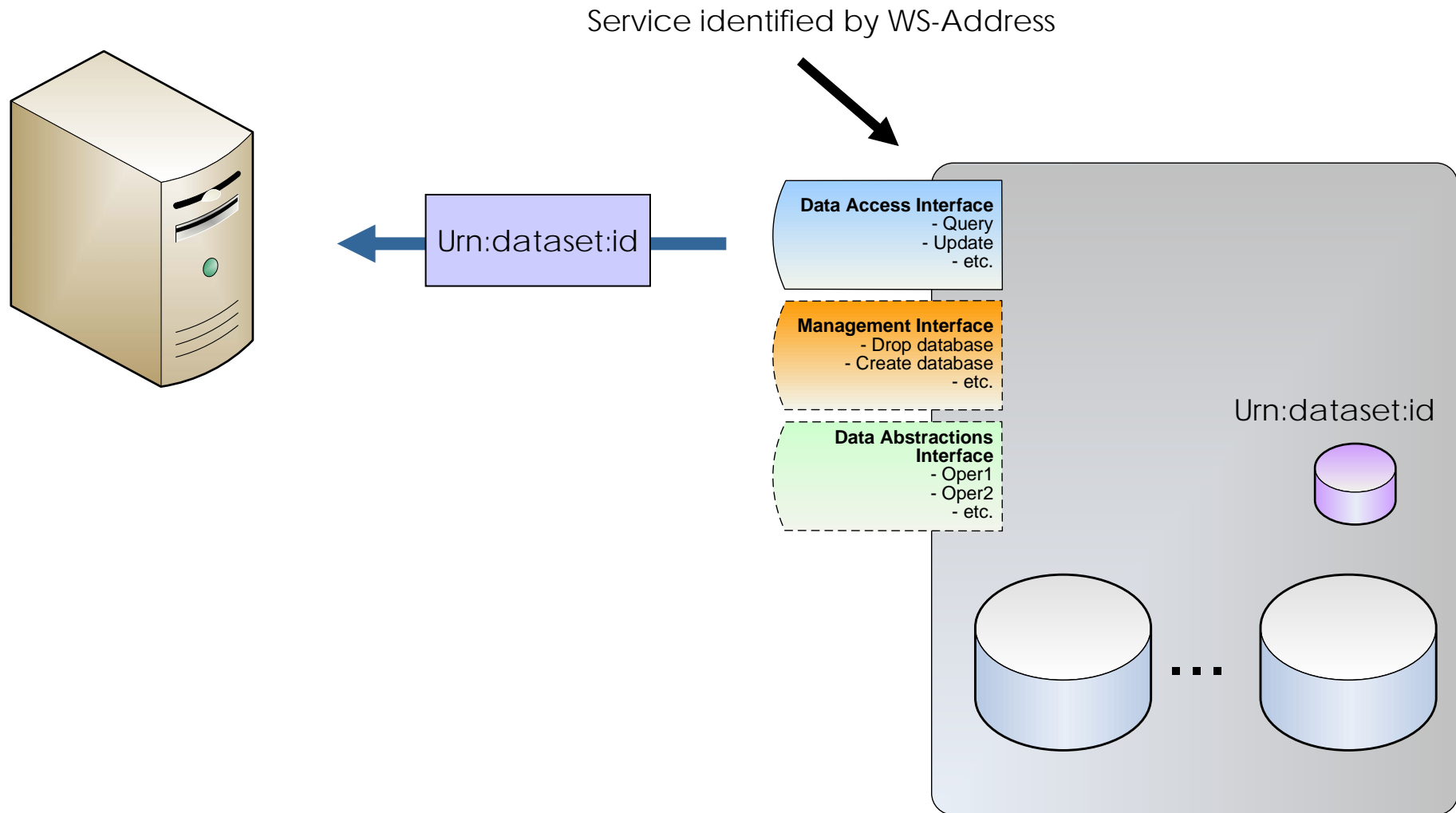
- Sessions
- Datasets

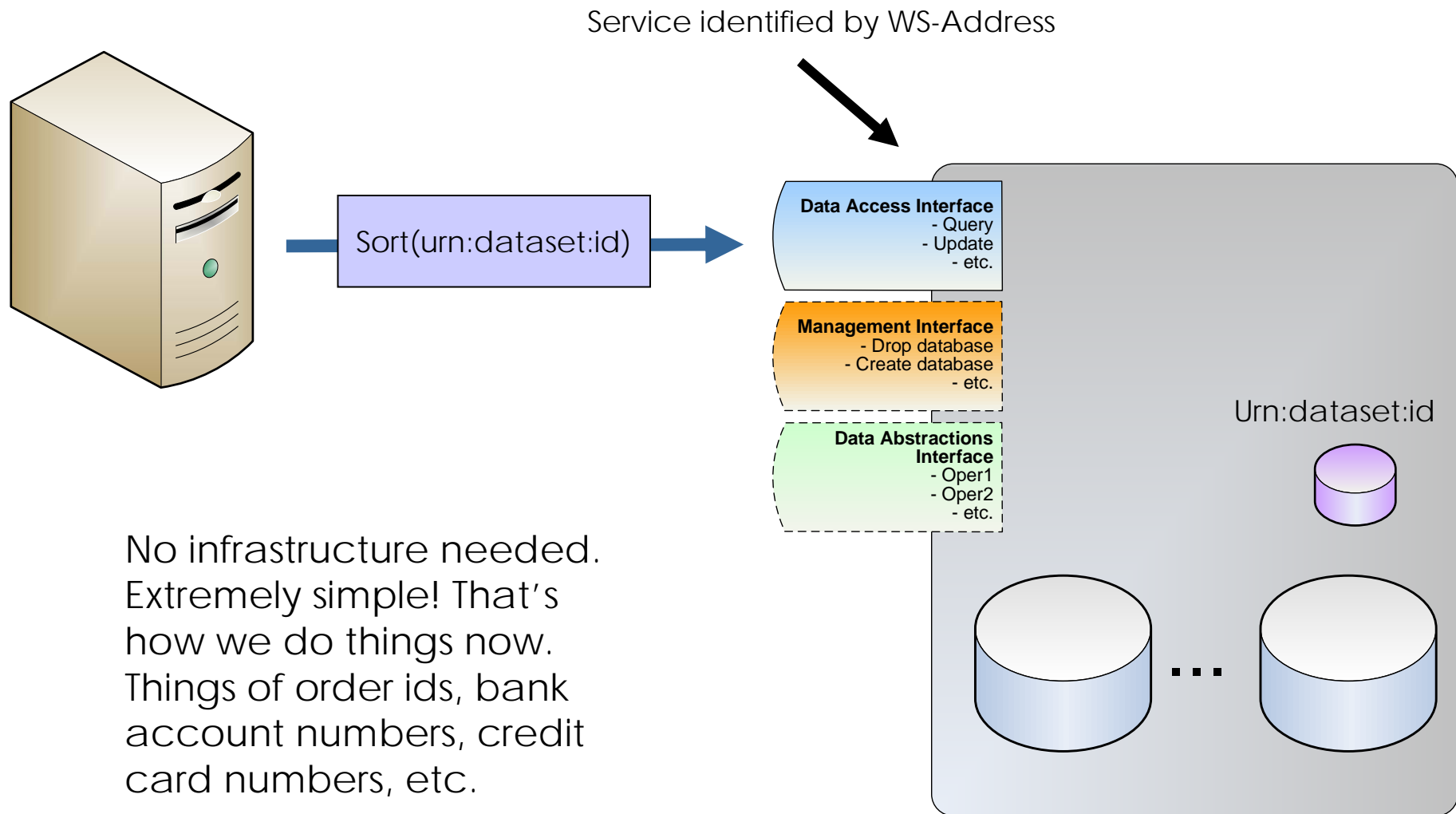


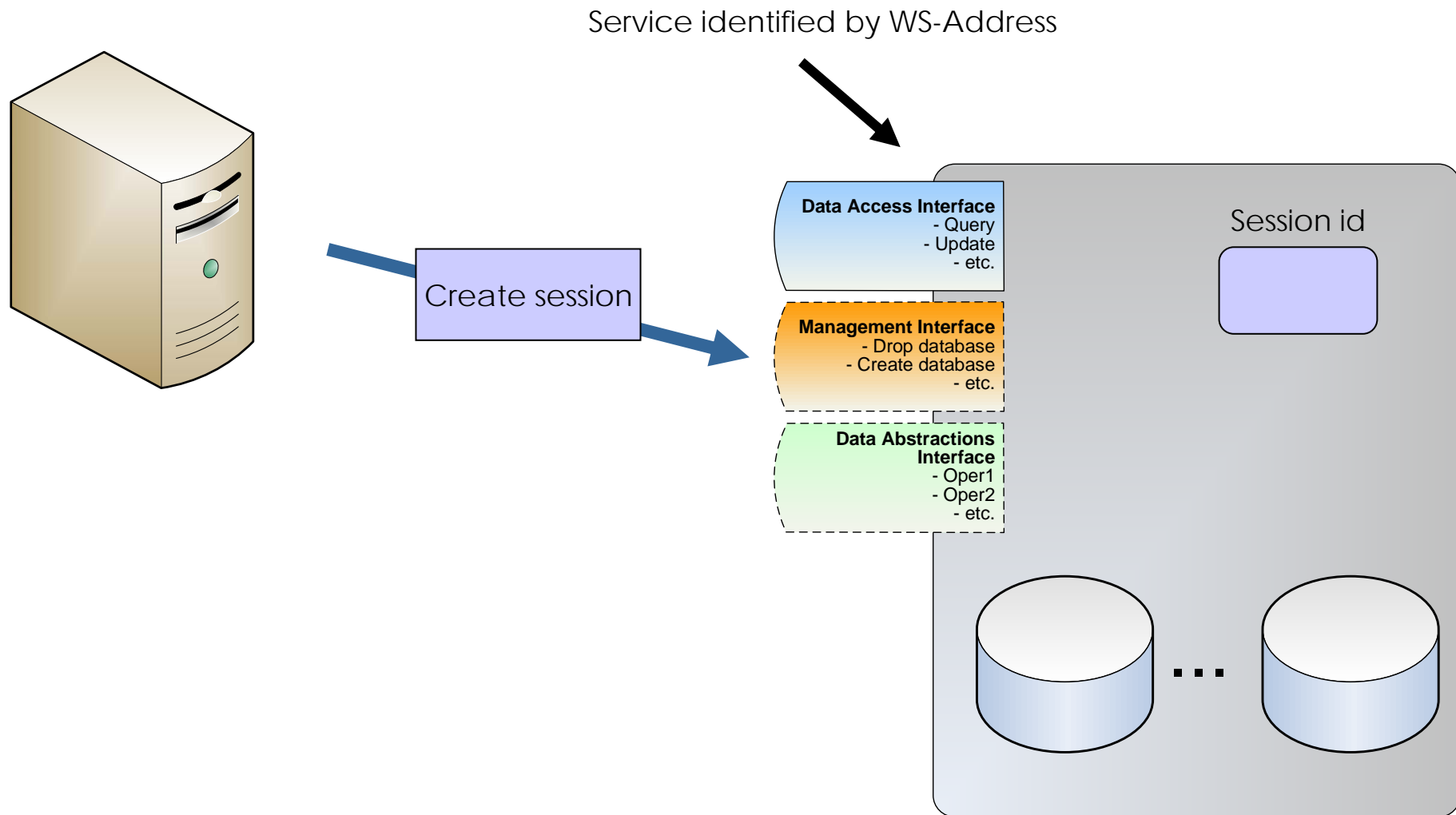


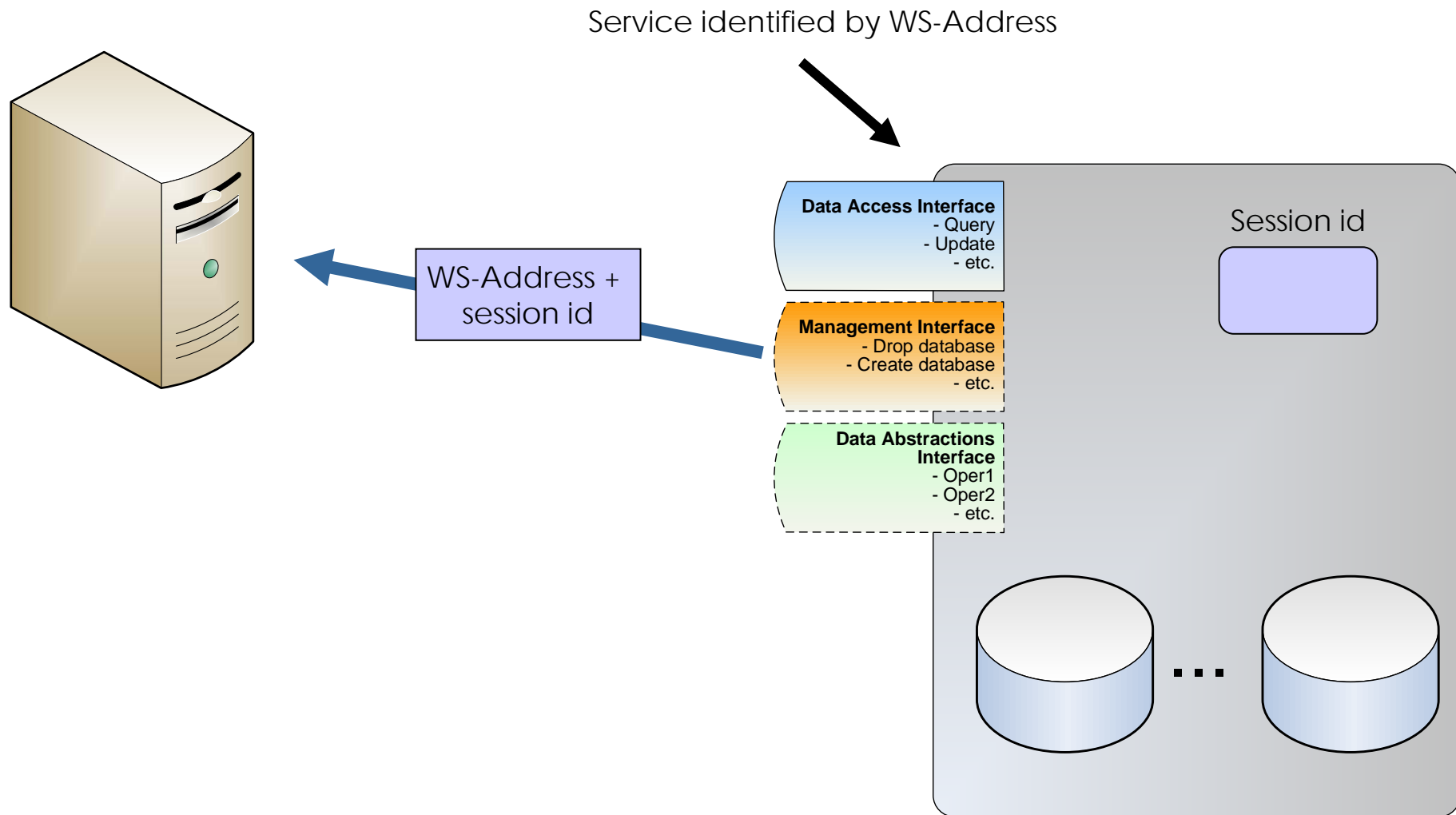


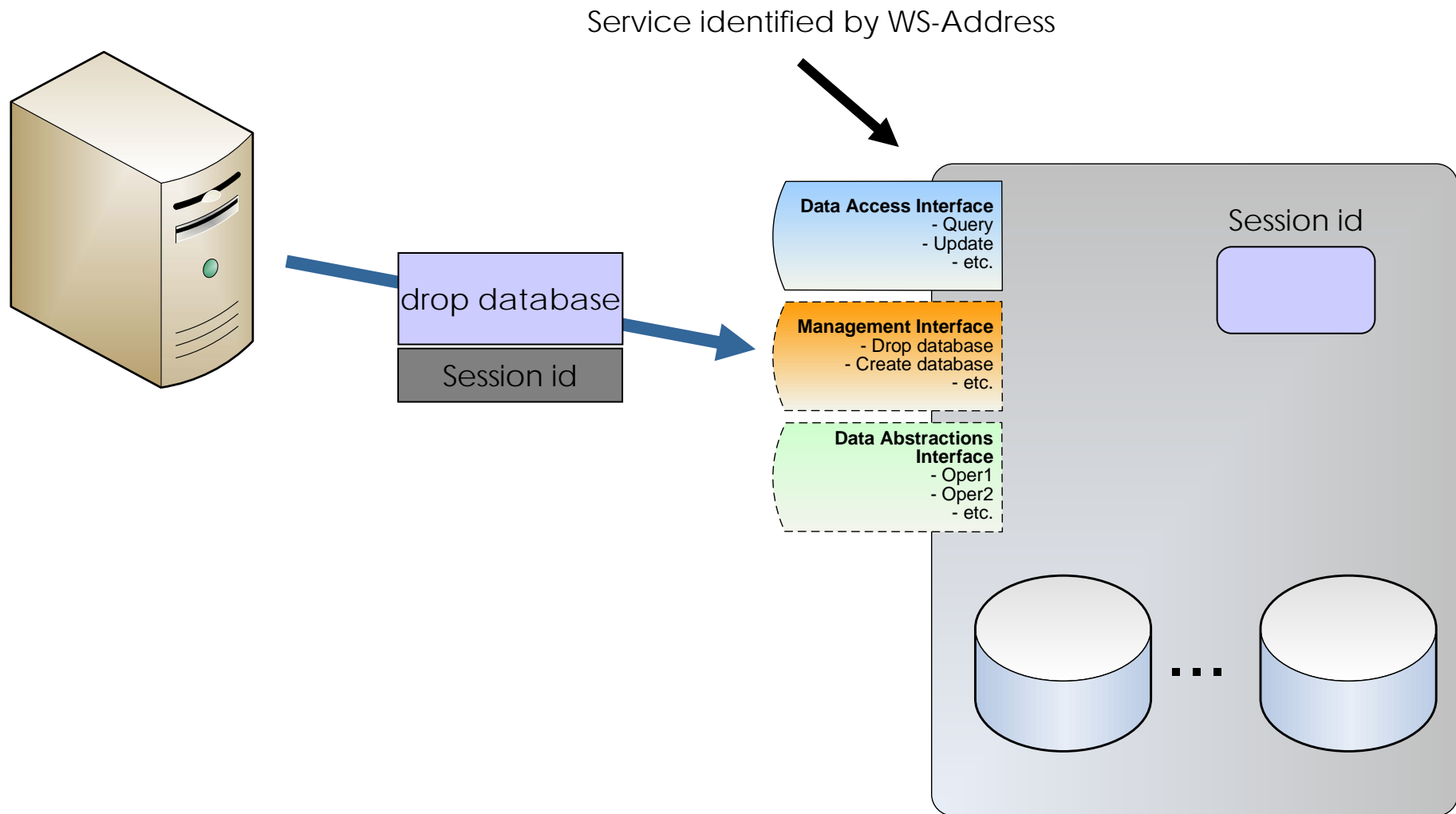




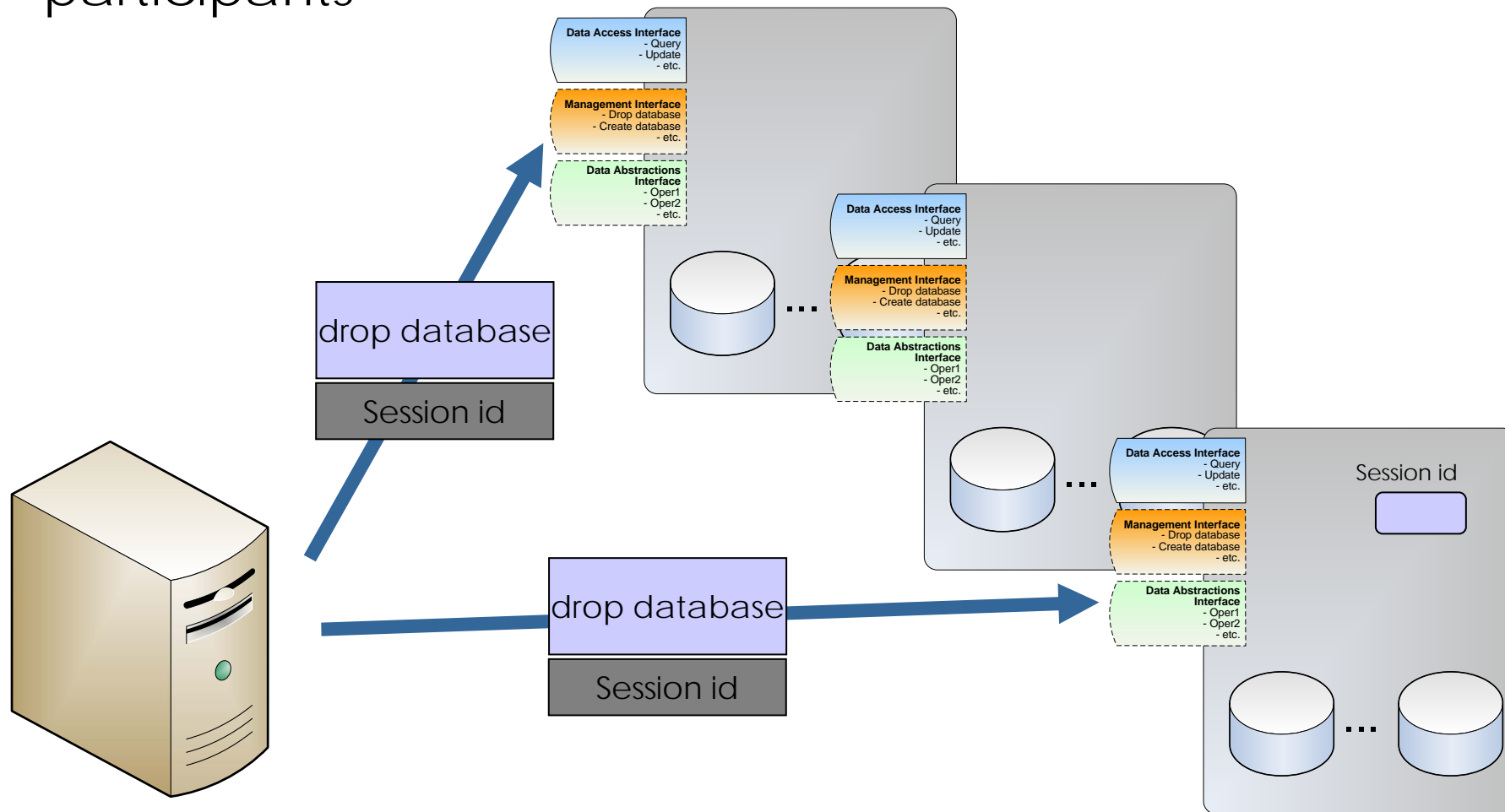




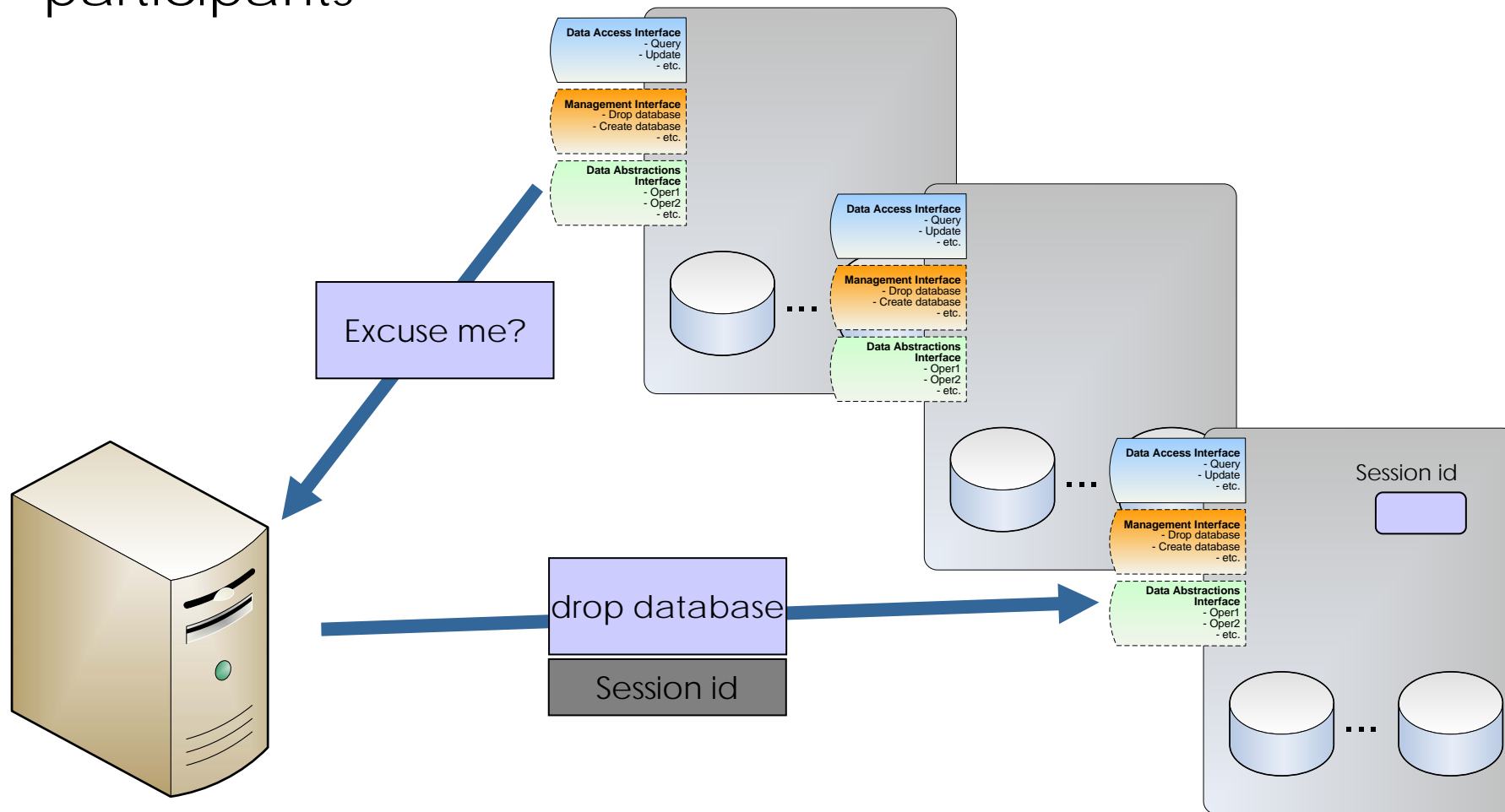




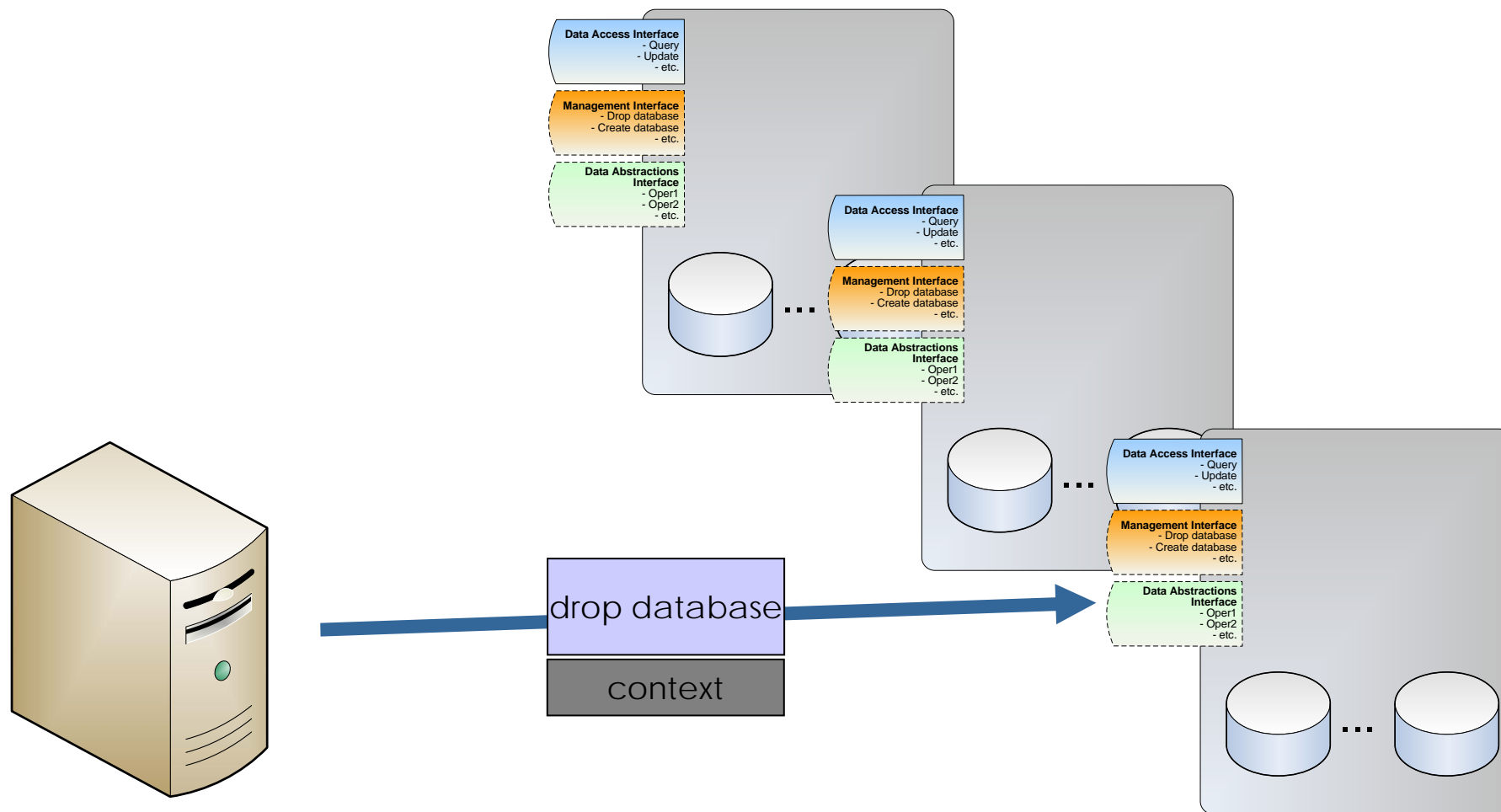
- Problem with a stateful interaction that involves multiple participants



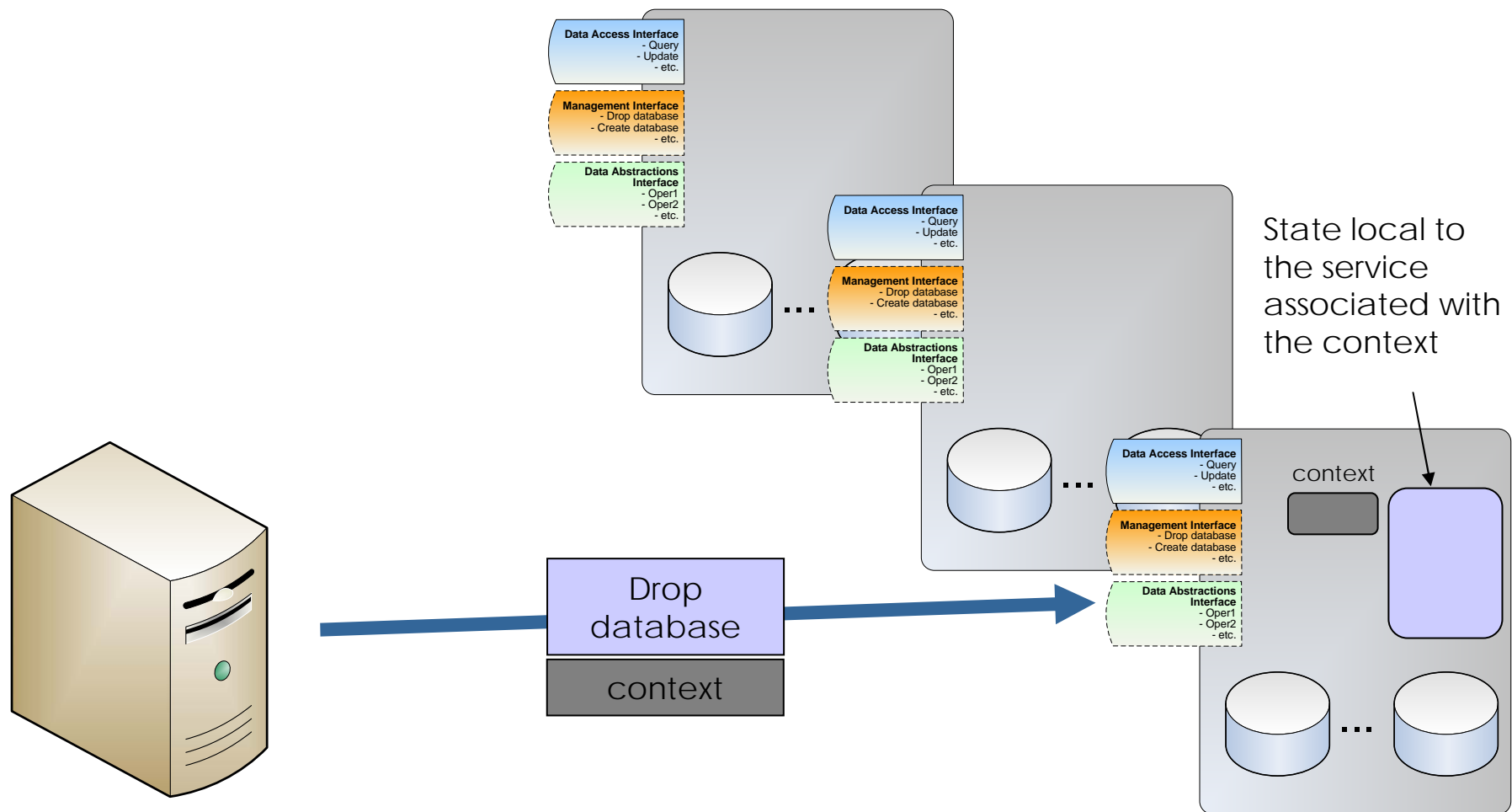
- Problem with a stateful interaction that involves multiple participants



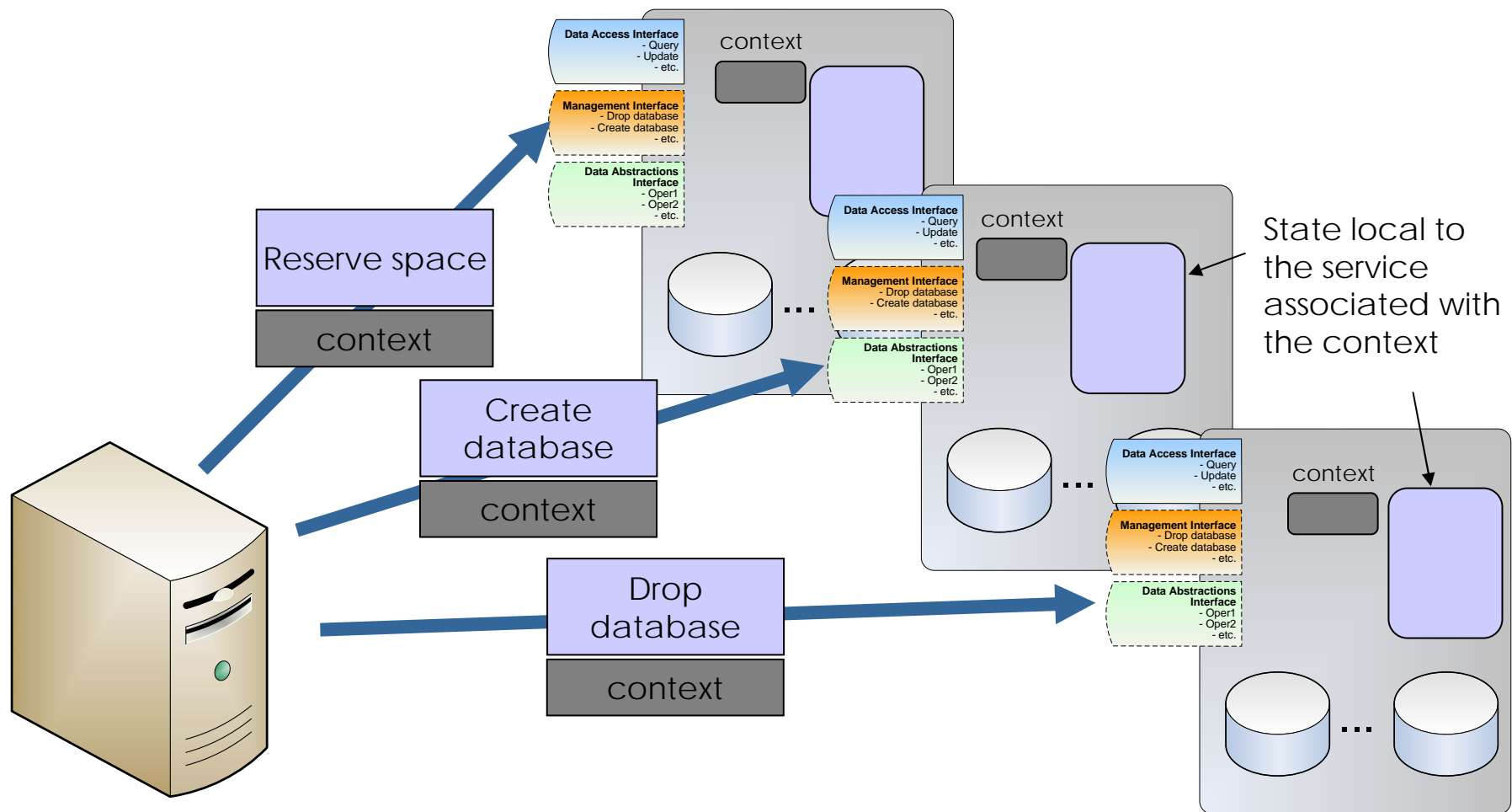
- Multiple participants



- Multiple participants

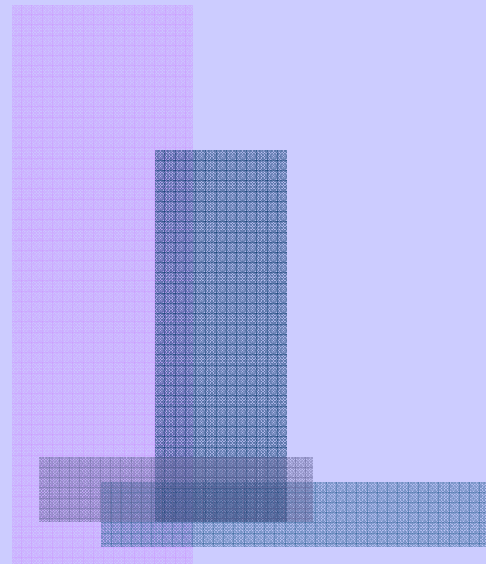


- Multiple participants



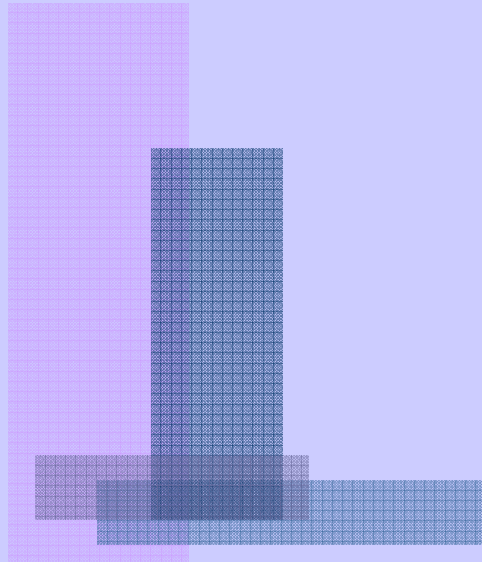
- Notifications from DAI services
  - If resources are not modelled directly using WS-Resource topics could still be used
- Data access Interfaces
  - May have to change depending on what approach you adopt
  - My proposal is to be flexible and allow implicit and explicit contextualisation (WS-Resource vs argument passing)
- Data management interfaces
  - Same as above
- Metadata about resources
  - Metadata document and associated interface
- "Perform" documents
  - Call it "workflow" or "grouped actions" or "savas"
  - They are just XML documents

## Conclusions



- We believe that WS-GAF meets the same requirements as OGSI v1.0 by using today's WS specifications and practices
- We believe that WS-GAF has a range of benefits
- What's next?

## Searching for "White Dwarfs"



- Aims
  - Define the characteristics of a “typical” Grid application
  - Demonstrate the applicability of the WS-GAF approach in building Grid applications
  - Learn from the challenges of constructing a truly global, distributed, scalable, loosely-coupled application
- Working on two “typical”, global-scale Grid applications with international partners
  - Built on the WS-GAF concepts
  - Investigate the need for WS-Resource
  - Document the experiences and report to the community
- We encourage everyone’s involvement

- Search for “white dwarfs” in our galaxy
- Utilise Jim Gray’s SkyServer
- Utilise computational resources
- Security from the beginning
- Visualisation
- Working with many people from the US, UK, and even one in Australia :-)

- Paul Watson (Paul.Watson@newcastle.ac.uk)
- Savas Parastatidis (Savas.Parastatidis@newcastle.ac.uk)
- Jim Webber (Jim.Webber@newcastle.ac.uk)

## Web Services Grid Application Framework (WS-GAF)

<http://www.neresc.ac.uk/ws-gaf>

[ws-gaf@newcastle.ac.uk](mailto:ws-gaf@newcastle.ac.uk)

Join by sending a message to [mailbase@newcastle.ac.uk](mailto:mailbase@newcastle.ac.uk) including the following line in the body

`join ws-gaf YourFirstName YourLastName`

# Thanks



- DTI
- JISC
- UK e-Science Core programme



- Is there a difference in the focus between WS-Resource and WS-GAF?
  - Perhaps, single view of a system that can be managed vs global-scale, services-based, loosely-coupled applications