

Web and Grid Services – Impact on DAIS Specification

Savas Parastatidis

Savas@Parastatidis.name
<http://savas.parastatidis.name>

Manchester – 09.12.2003

- The following are just our views on how DAIS could fit well with the Web Services Architecture. They may still be far from those in the OGSi 1.5 proposal.
- We are not involved in any of the OGSi 1.5 related discussions

- Service Oriented Computing
- Web Services
- Web Services vs Distributed Objects
- DAIS and OGSi
- DAIS and WSA
- Grid Resource Metadata Demo

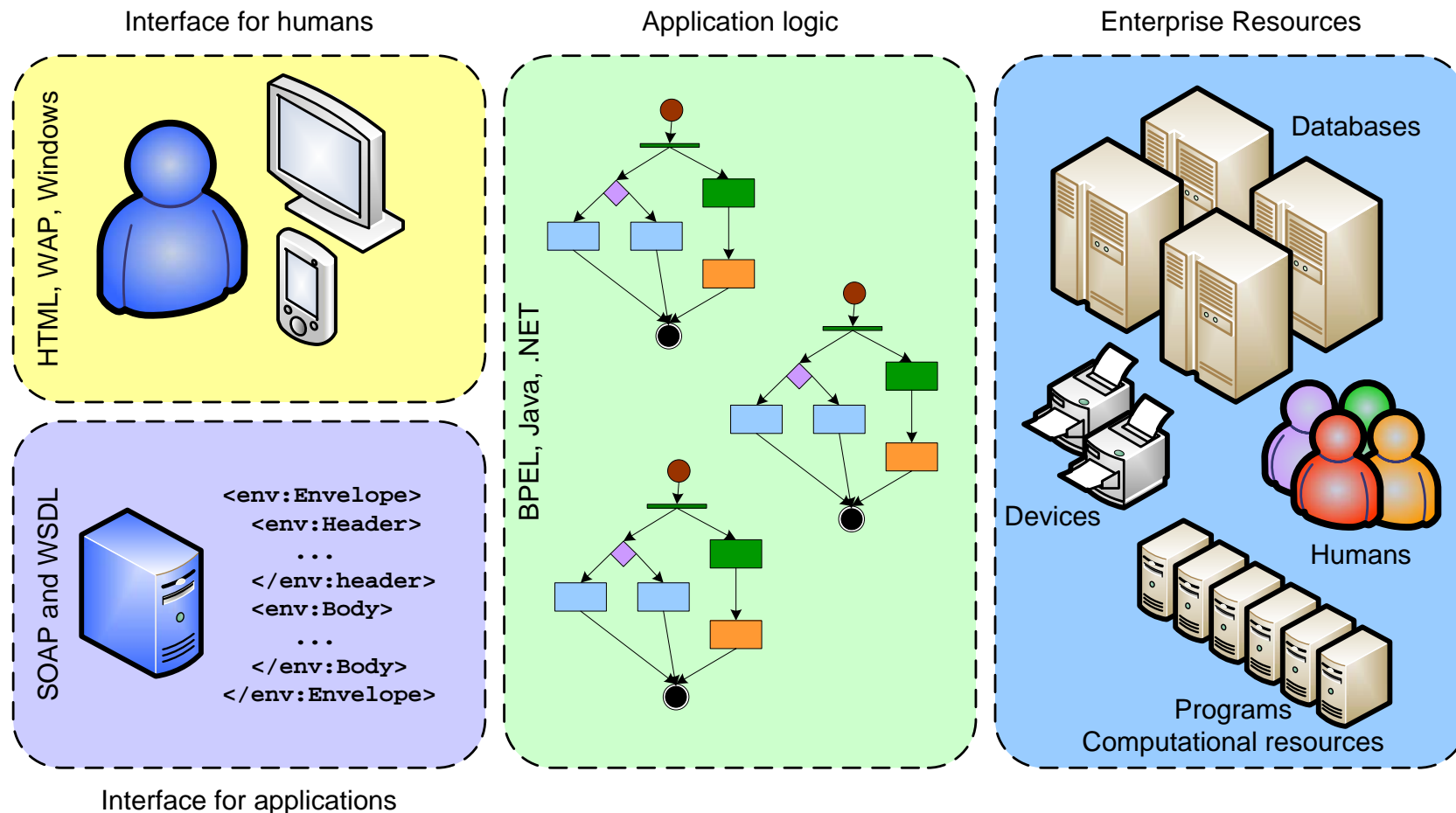
- Built around the concepts of service and message
- A service may be defined as *a logical manifestation of some physical resources (like databases, programs, devices, or humans) that an organization exposes to the network*
- A service is an entity that can send and receive messages
- A service adheres 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

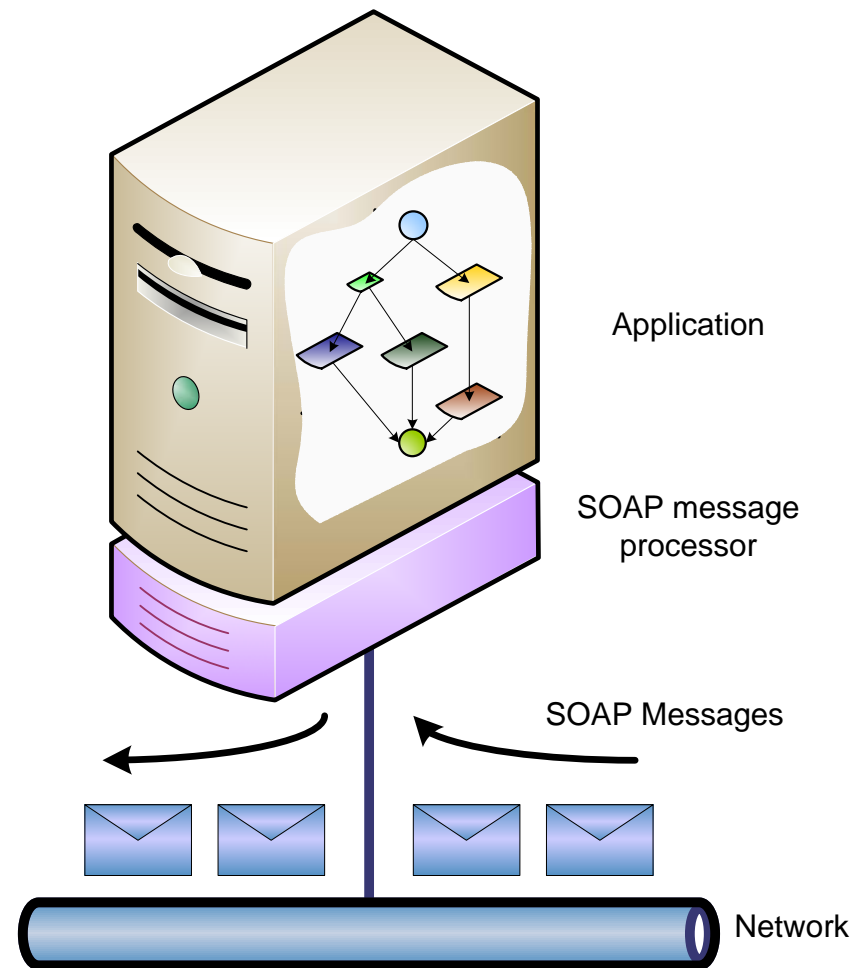
Source: "A Guide to Developing and Running Connected Systems with Indigo"
<http://msdn.microsoft.com/Longhorn/understanding/mag/default.aspx?pull=/msdnmag/issues/04/01/Indigo/default.aspx>
and various talks

The Anatomy of a Web Service

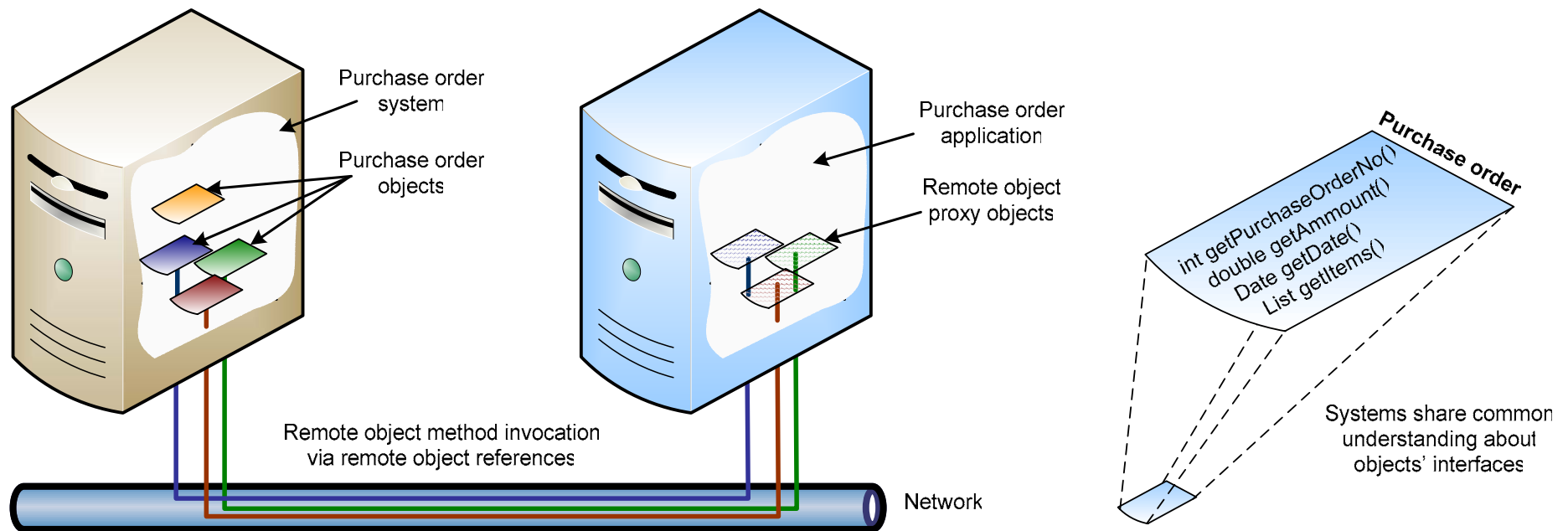
- Large grained, loosely coupled
 - Performance, scalability, maintenance, re-use, etc.



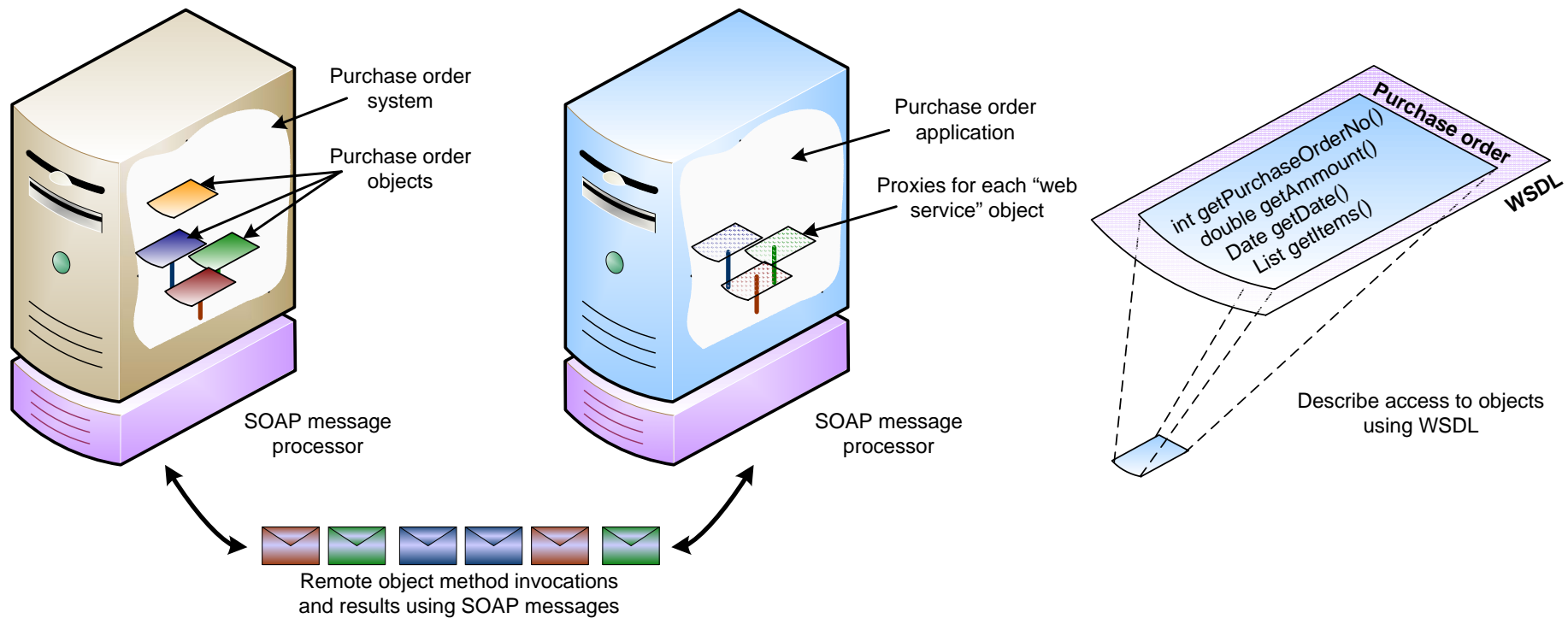
A Web Service

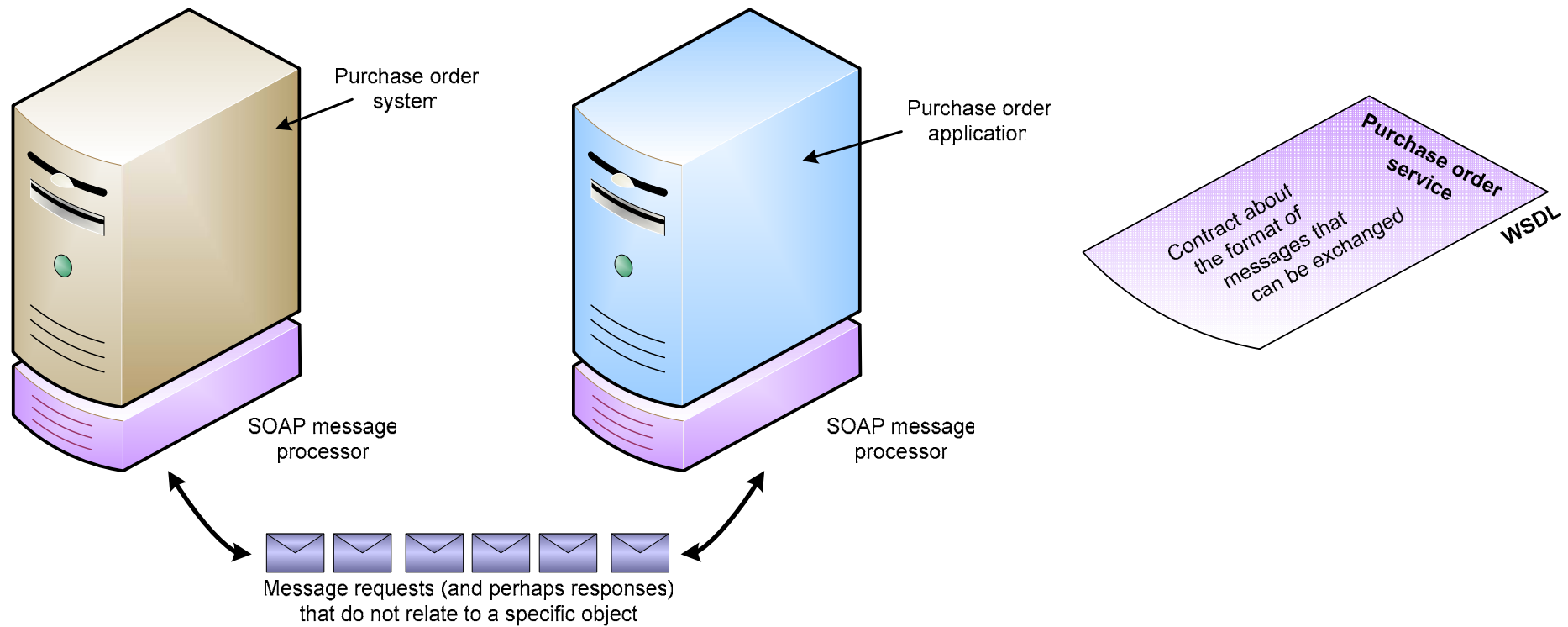


Distributed objects

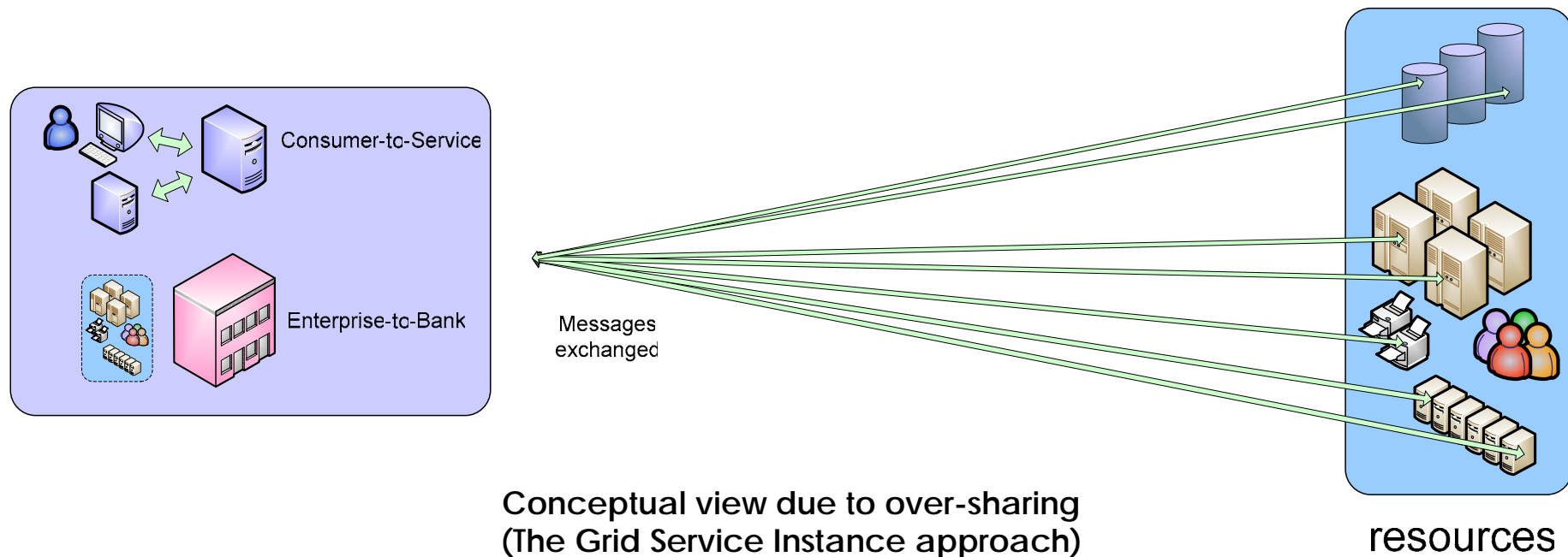


Distributed objects using SOAP and WSDL



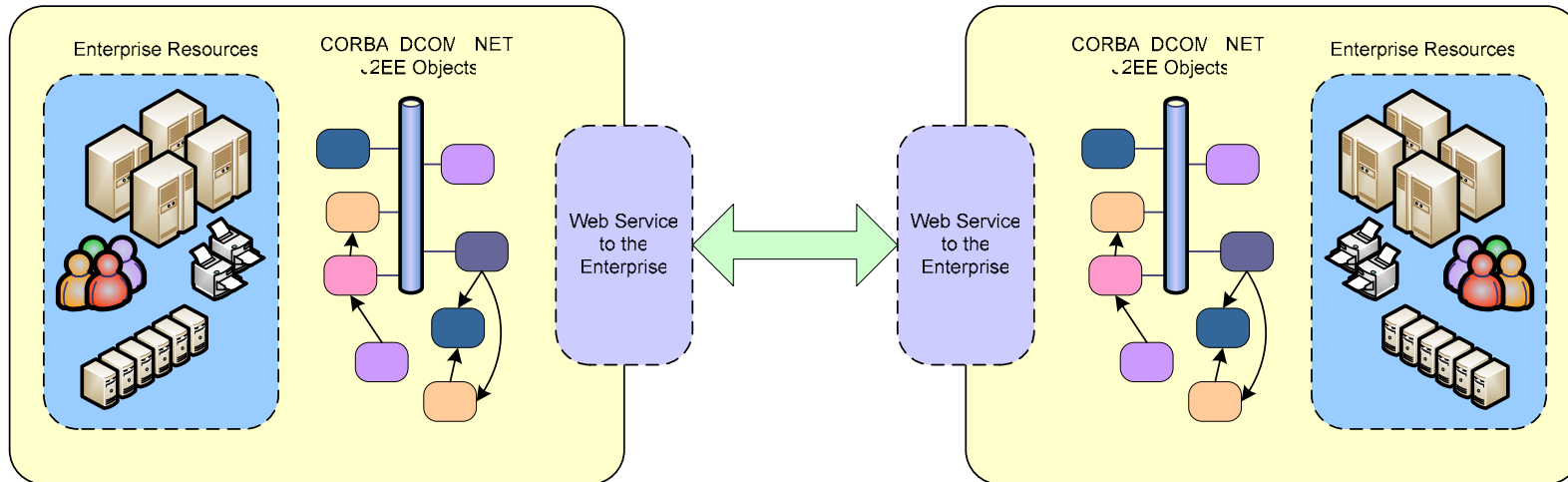


- Tight-coupling
- Easily breakable applications
- Poor scalability

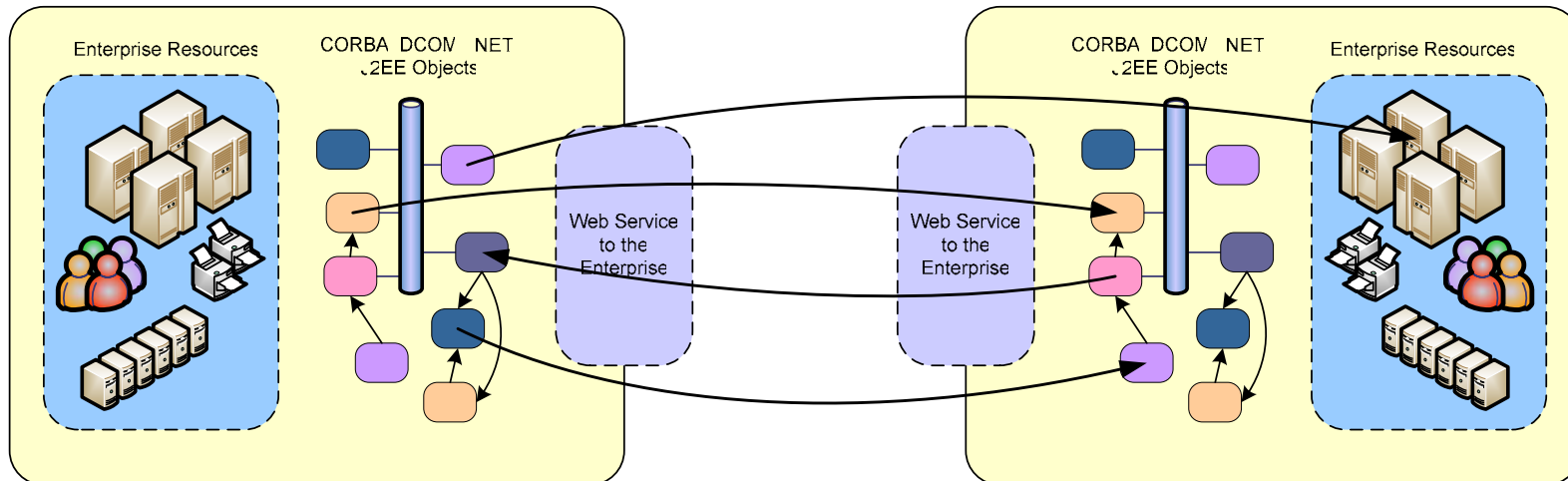


Talking directly to resources

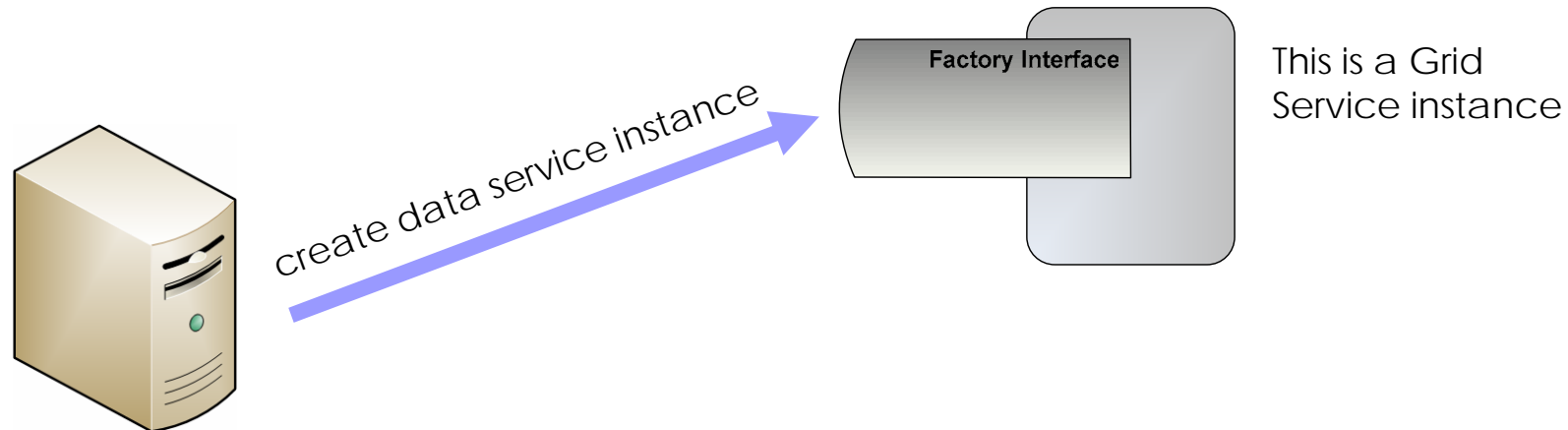
SOA



OGSI

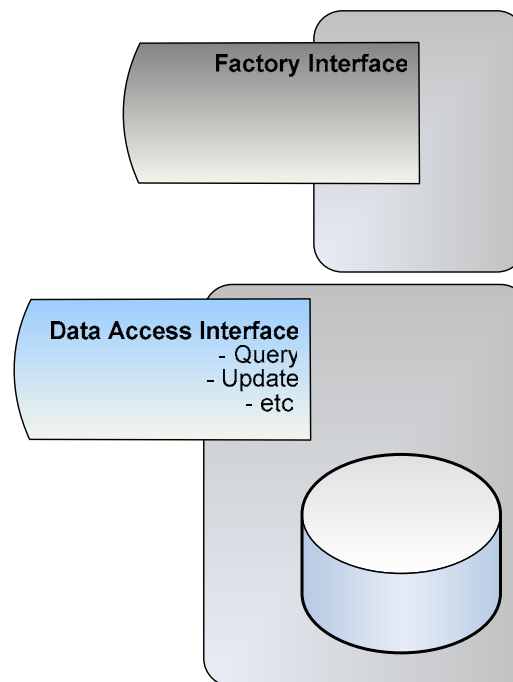


Data Services (the current approach) Factory for service instances



Data Services (the current approach)

A Service Instance for a Data Resource



This is a Grid Service instance

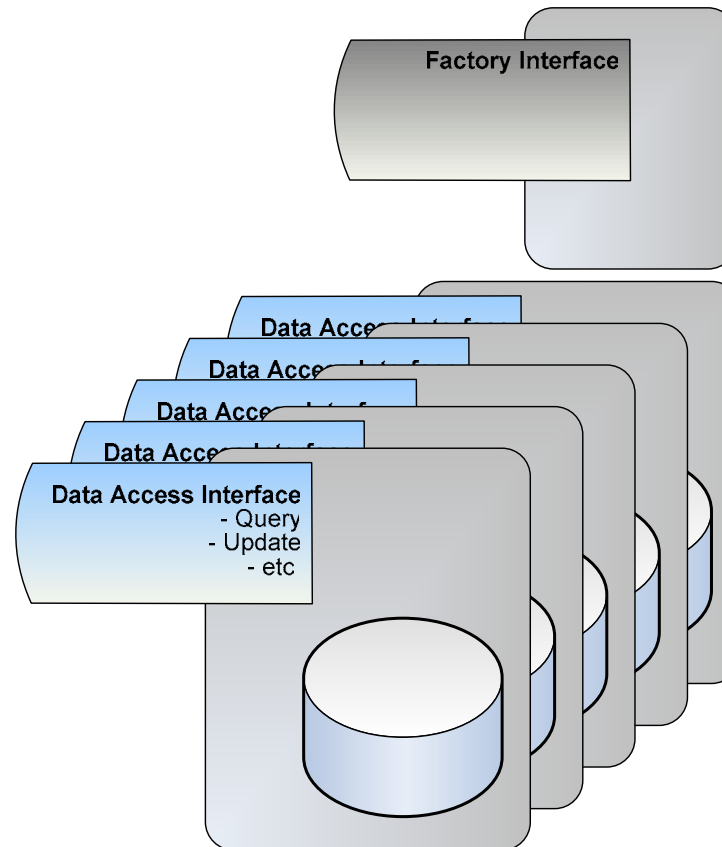
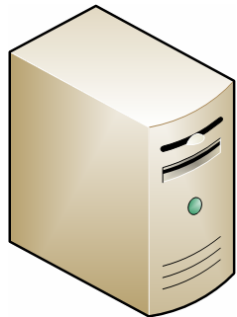
This is another Grid Service instance

The conceptual model encourages a logical 1-1 association with a data resource



Data Services (the current approach)

Multiple Service Instances



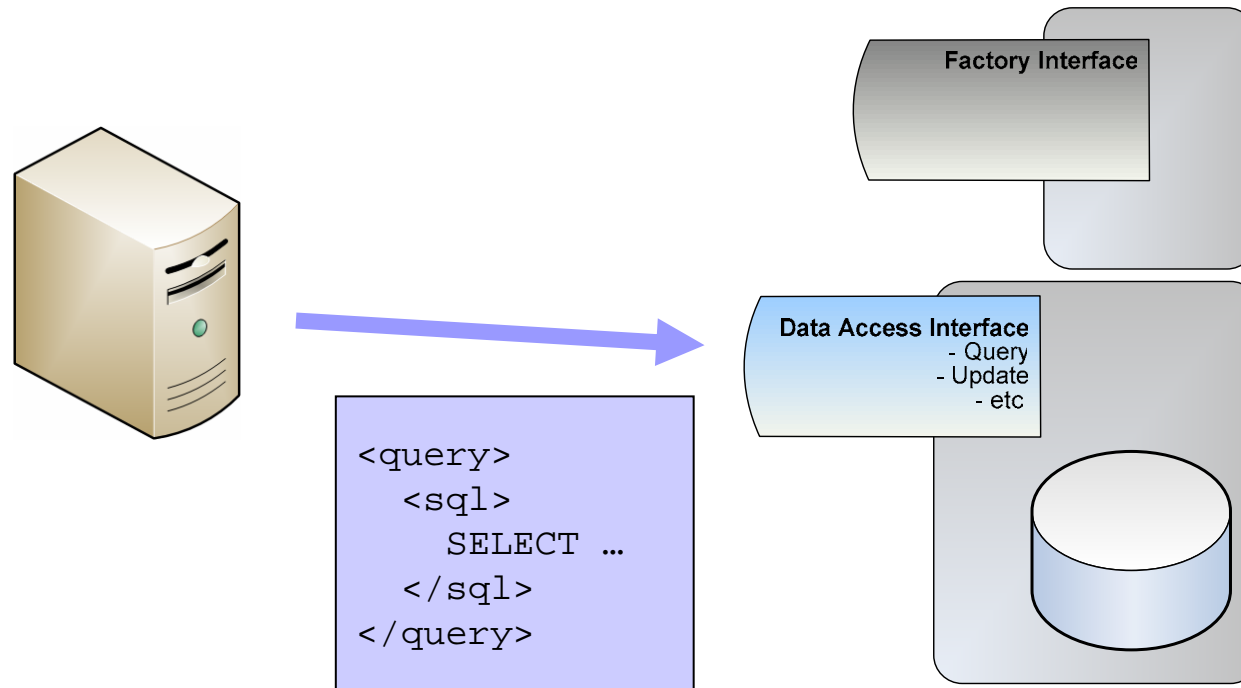
Multiple data resources are represented by multiple service instances

Each service instance also represents a session of interactions with the associated data resource



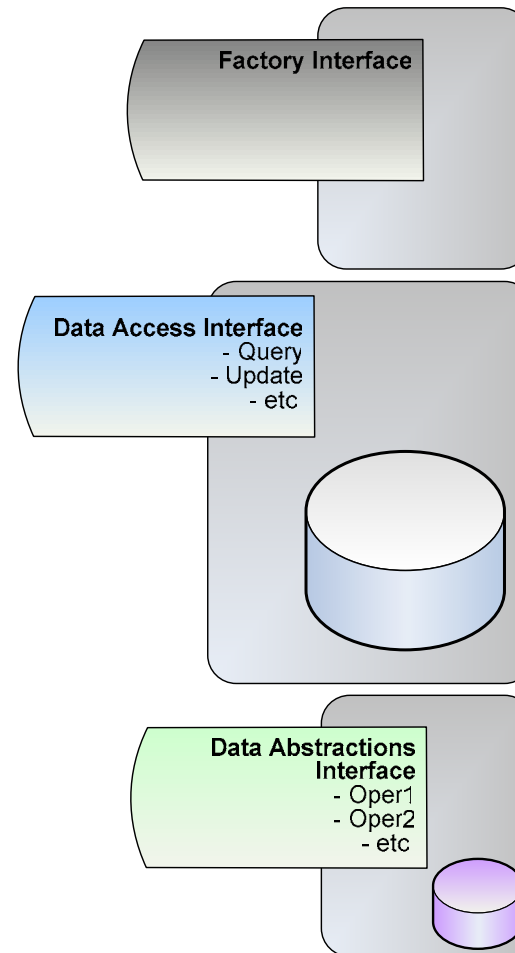
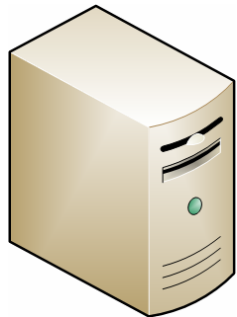
Data Services (the current approach)

A Query



Data Services (the current approach)

A Data Abstraction is Created

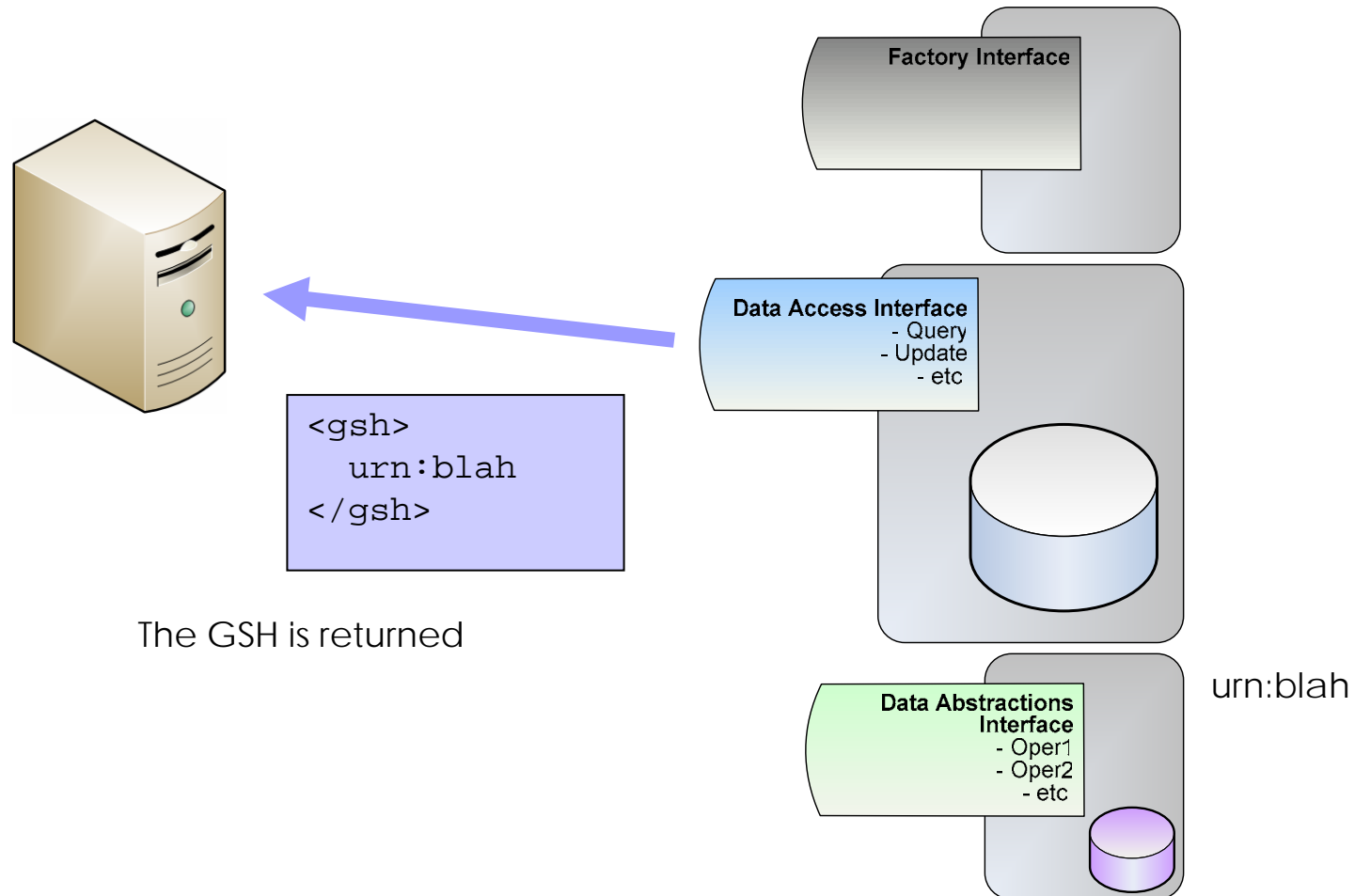


A data abstraction is created and logically associated with the data resource representing the result

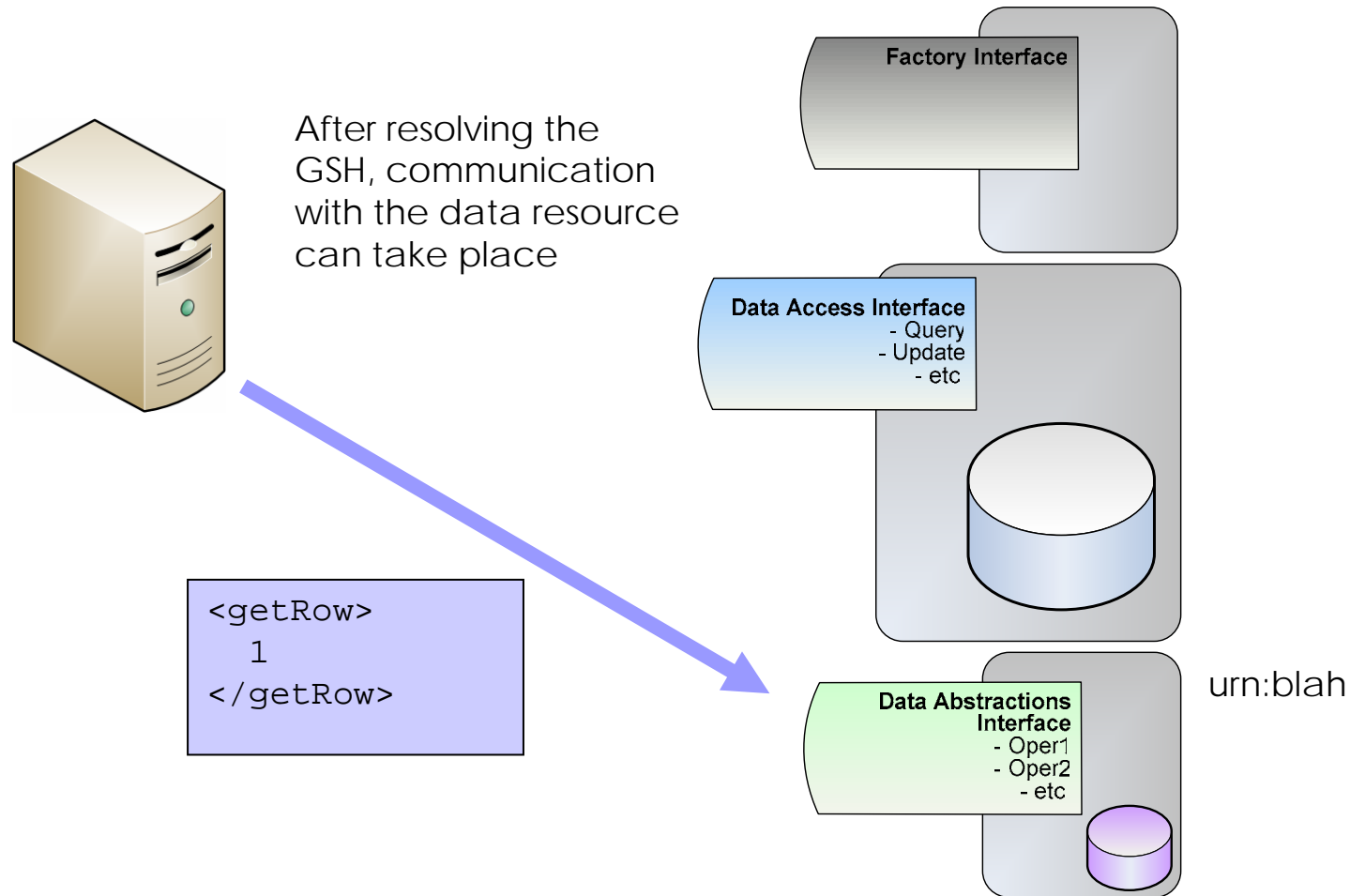


Data Services (the current approach)

GSH to Data Abstraction

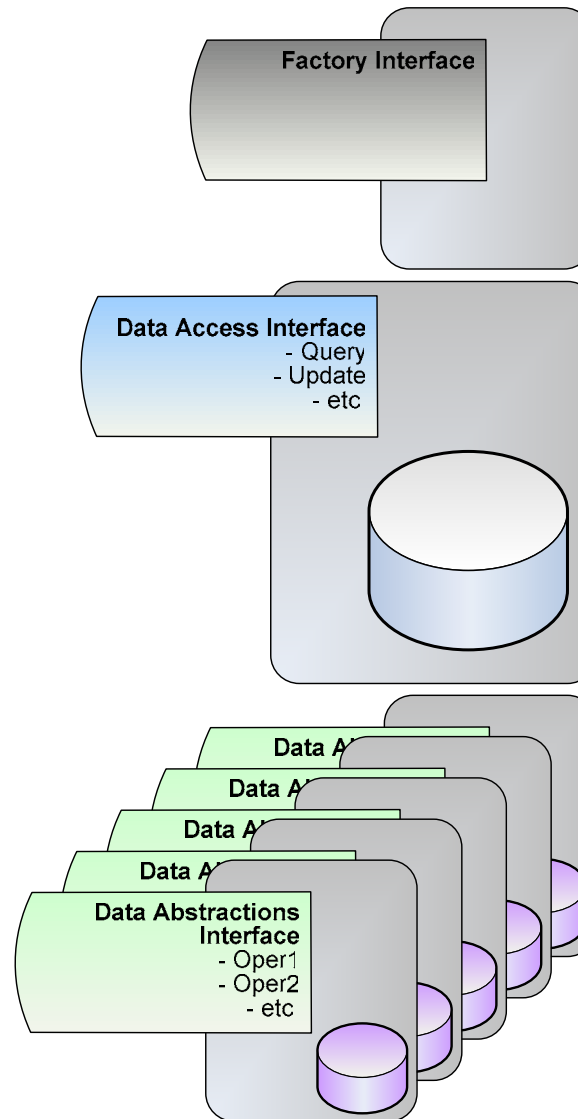
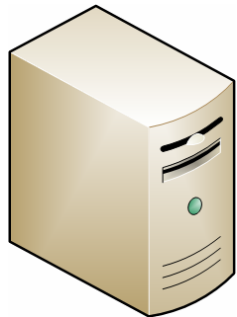


Data Services (the current approach) Communication with Data Abstraction



Data Services (the current approach)

Multiple Data Abstractions

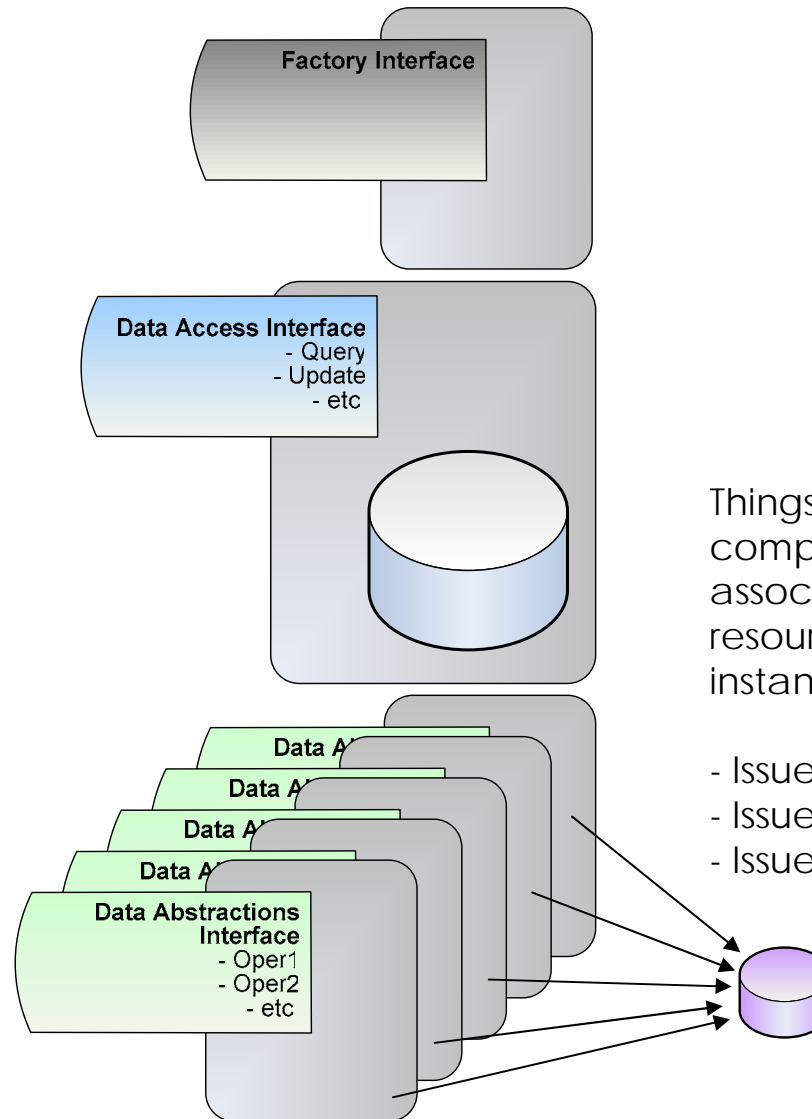
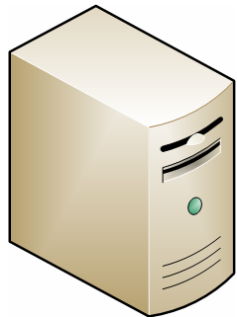


Multiple data
resources mean
multiple service
instances



Data Services (the current approach)

Multiple Data Abstractions



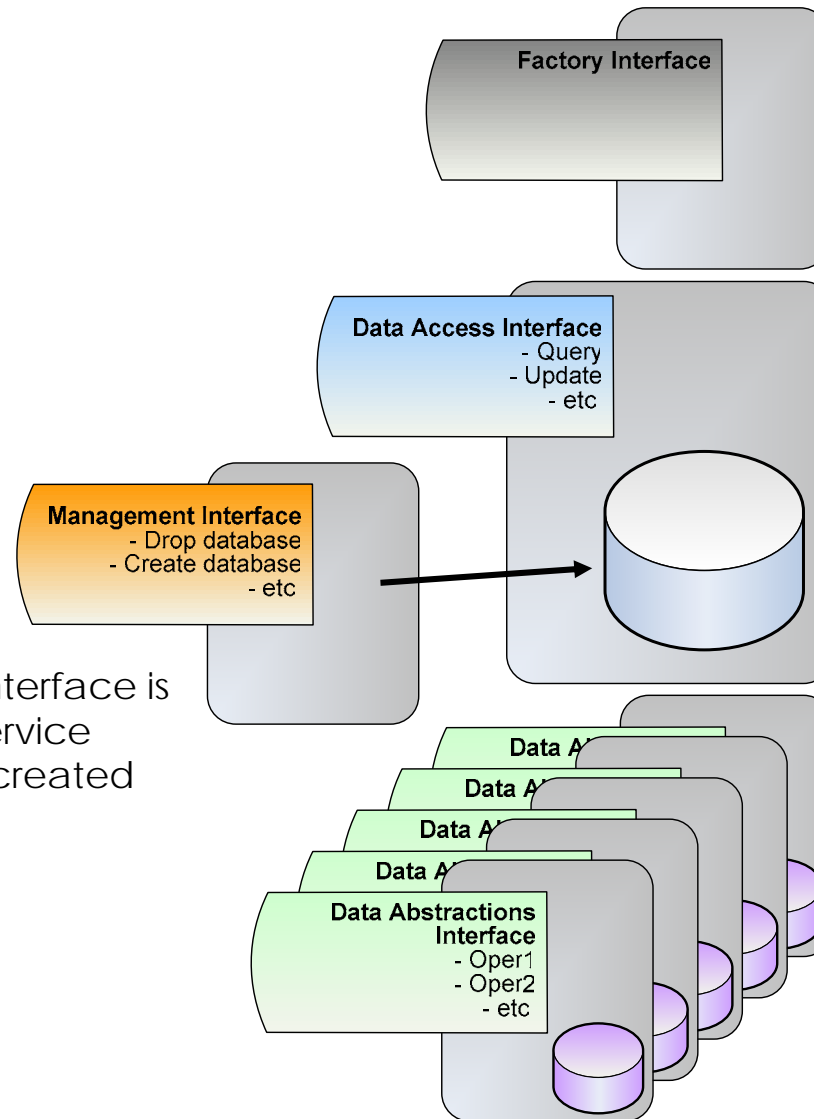
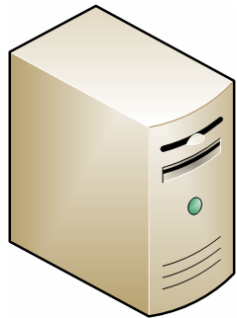
Things may get more complicated when the 1-1 association between a data resource and a service instance is not adhered

- Issues with identity
- Issues with relationships
- Issues with concurrency



Data Services (the current approach)

A Management Service

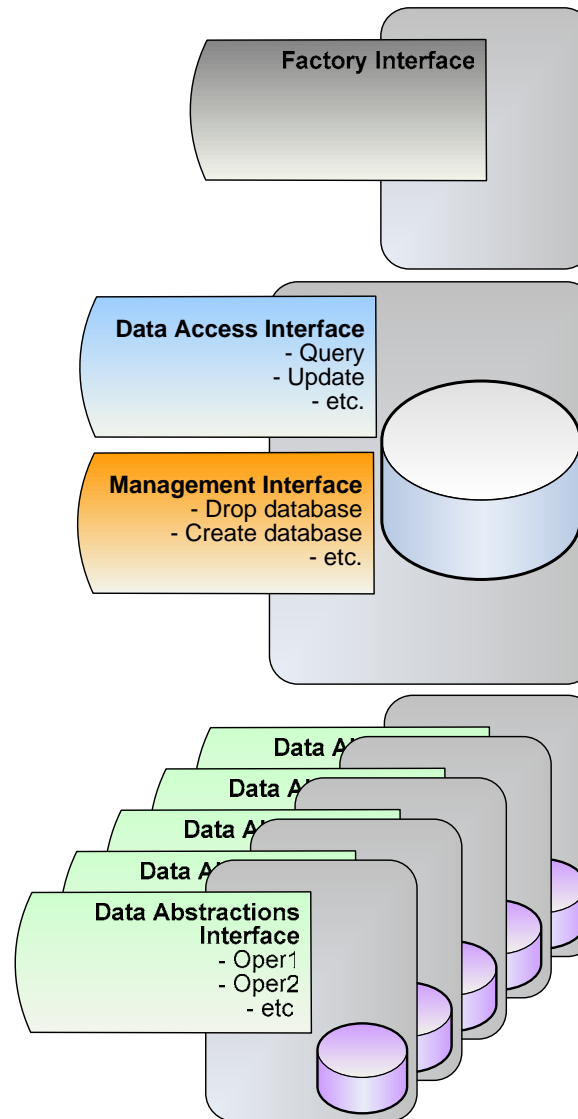
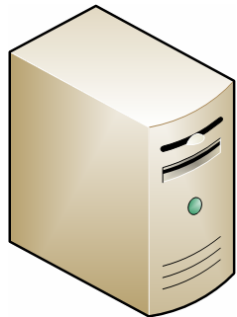


If a management interface is needed, another service instance has to be created



Data Services (the current approach)

A Data Access + Management Service

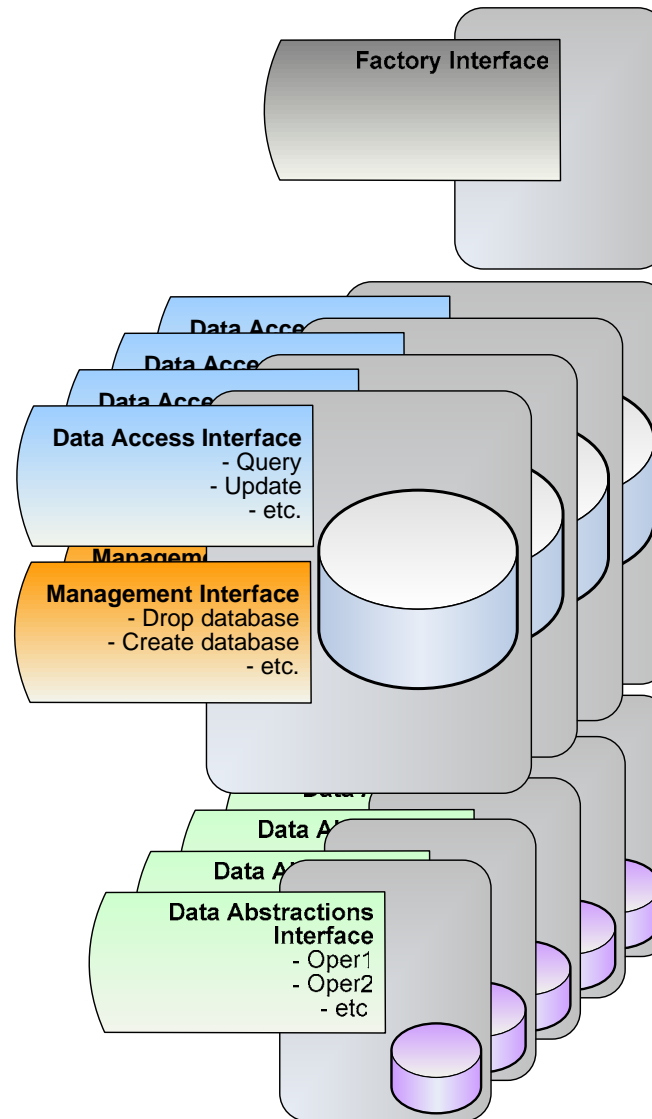
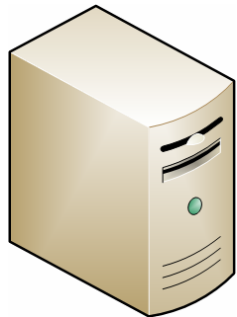


Or, a special service instance could be used to combine the two



Data Services (the current approach)

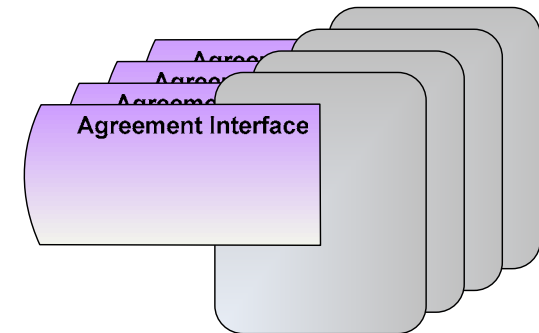
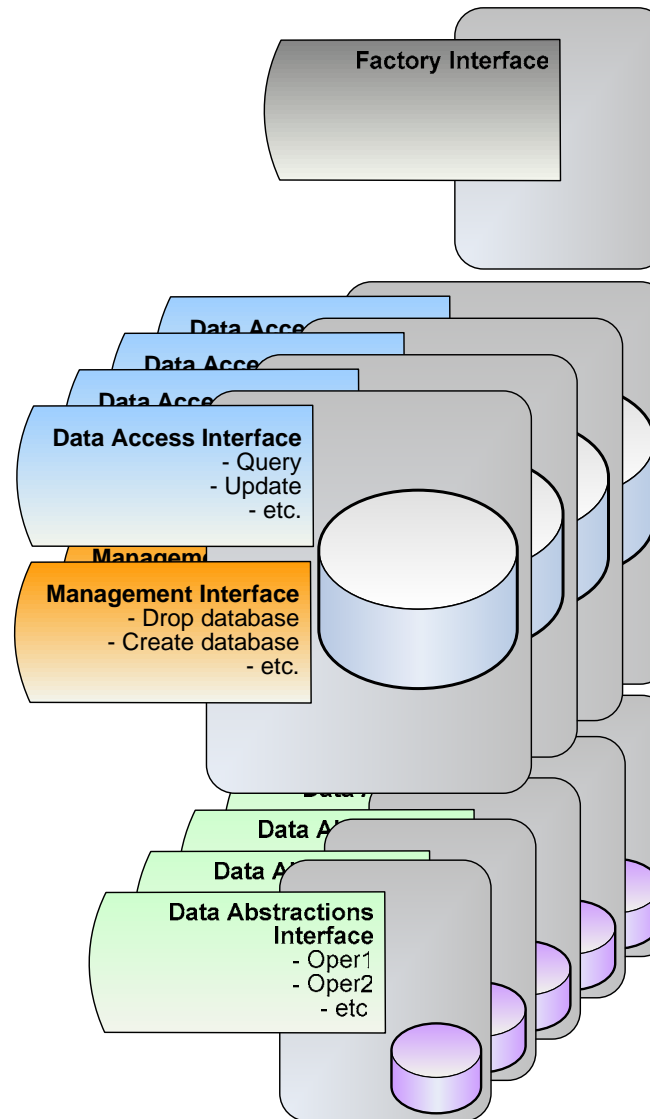
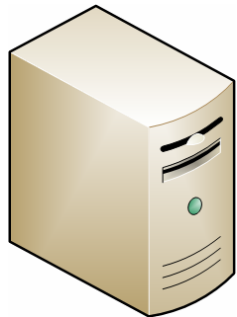
Multiple Data Access + Management Services



If more than one data resources are available by an organisation, it may be necessary to expose them through multiple data access + management interfaces



Data Services (the current approach) Agreements

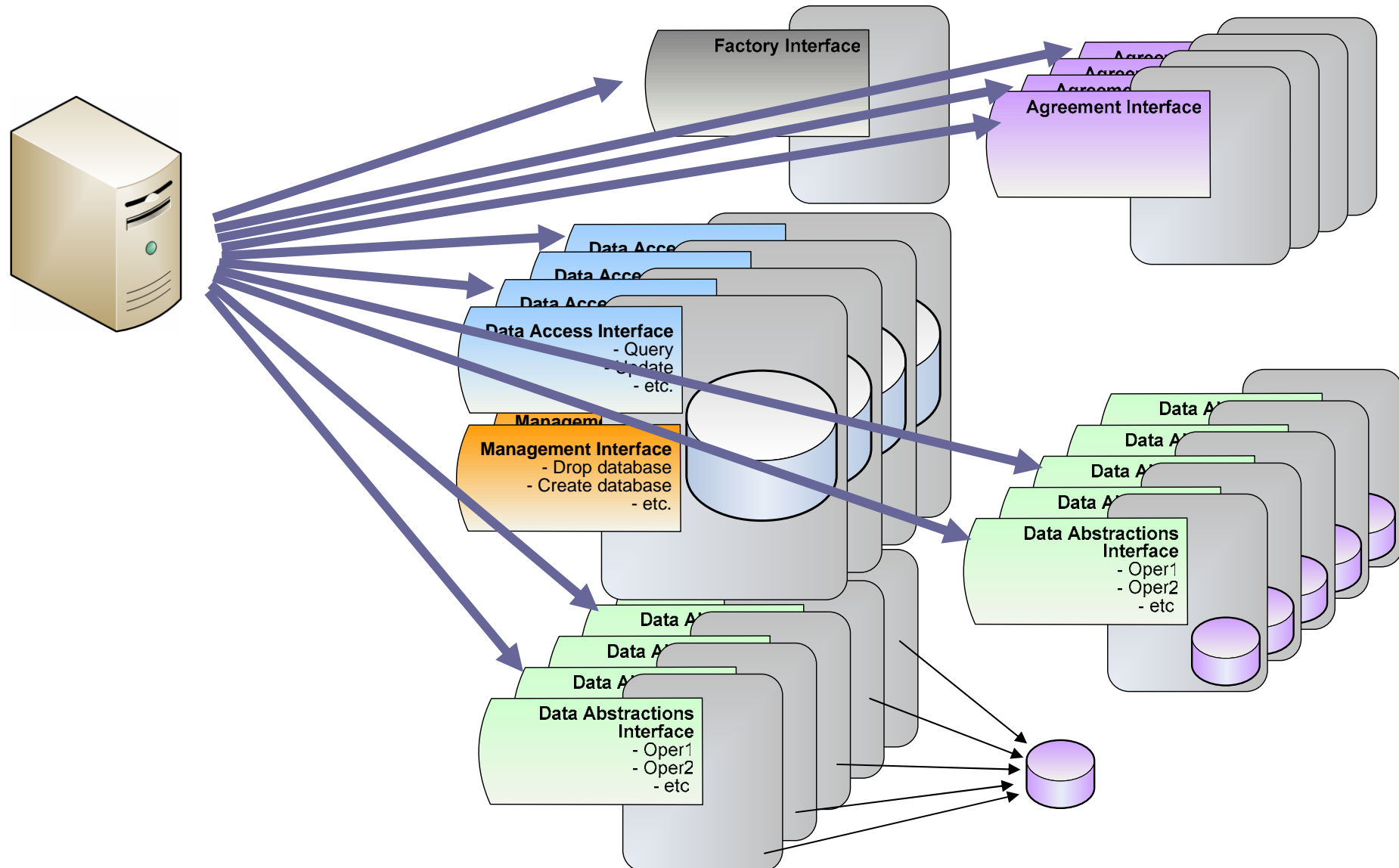


Some agreement service instances are introduced to control the creation of new data service instances

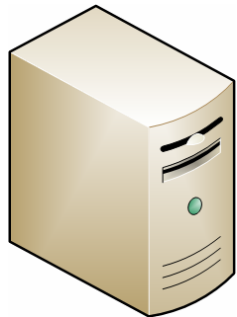


Data Services (the current approach)

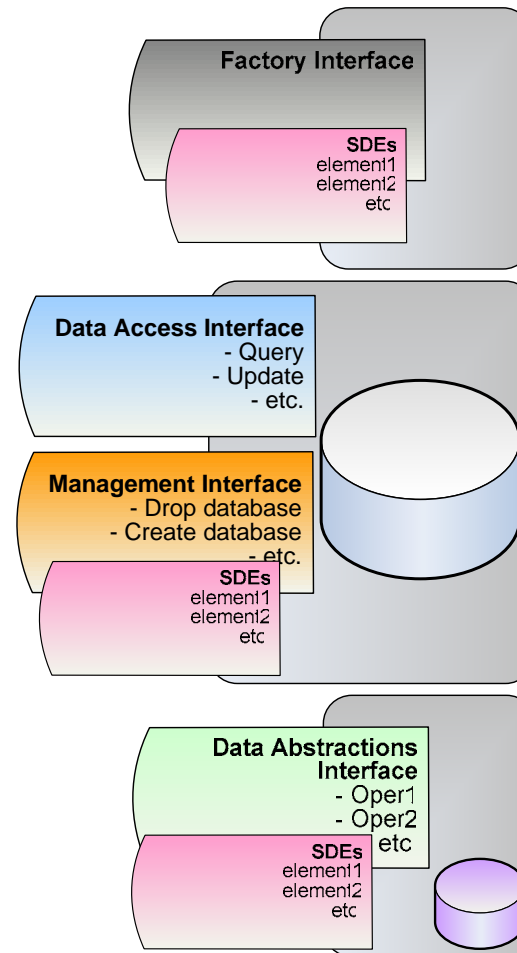
Putting everything together



Data Services (the current approach) Metadata

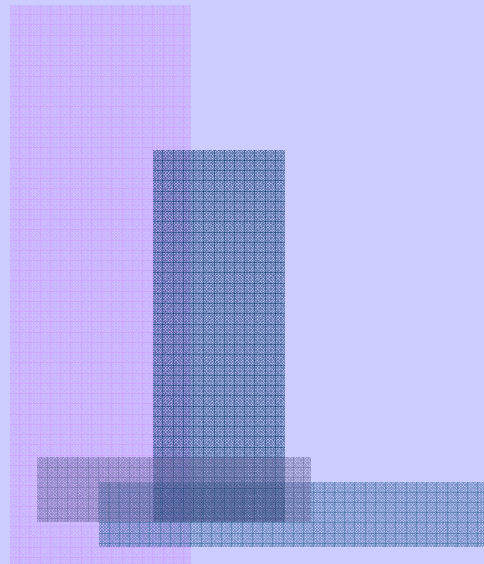


Metadata is exposed
through SDEs



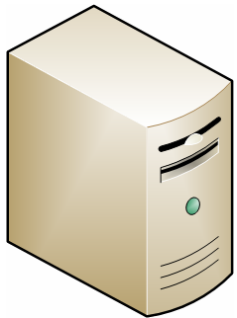
- This approach heavily depends on the OGSI conceptual model
- It is difficult to take advantage of it using existing tooling
- It encourages a distributed-object model
 - This MAY lead to tightly coupled, fragile applications
- An assumption about an 1-1 association between a service instance and a data resource
 - Identity
 - Relationships?
- Service Data Elements
 - Require new tools

A Possible Mapping of DAIS Concepts to WSA

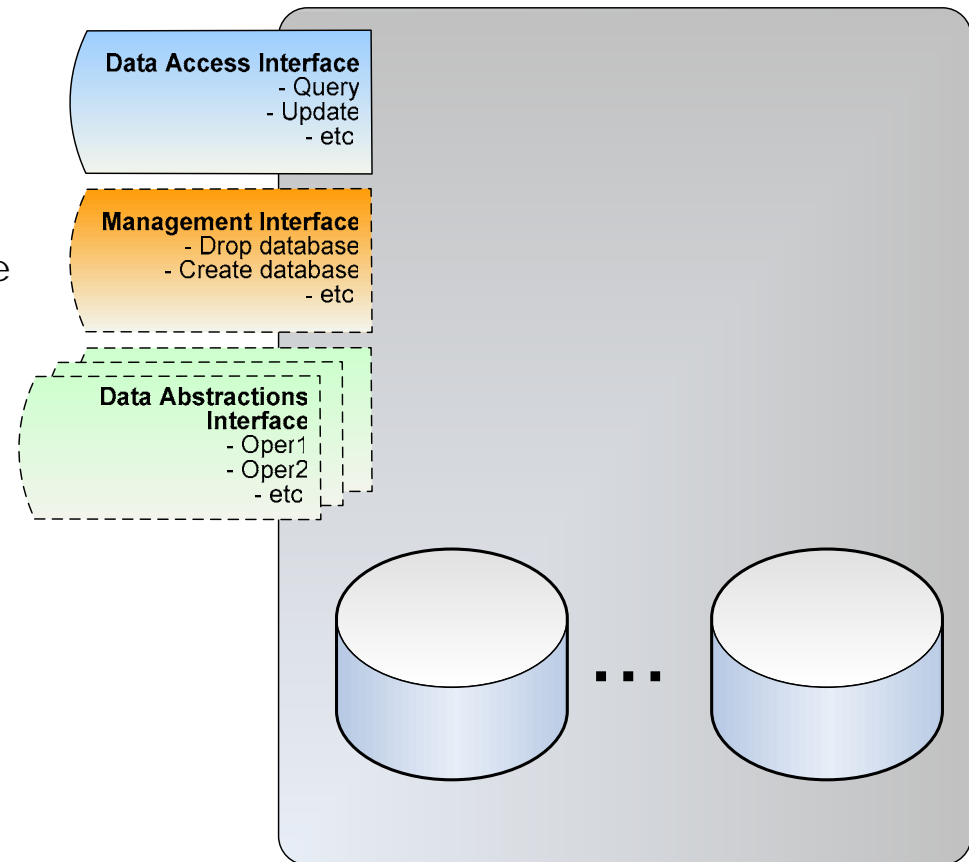


- Maintain same concepts but map them into WSA-friendly technologies
- Sessions
- Exposing data resources
 - Identity
 - Relationships
- Metadata
- Data abstractions
 - Interfaces to access data resources

A Data Service

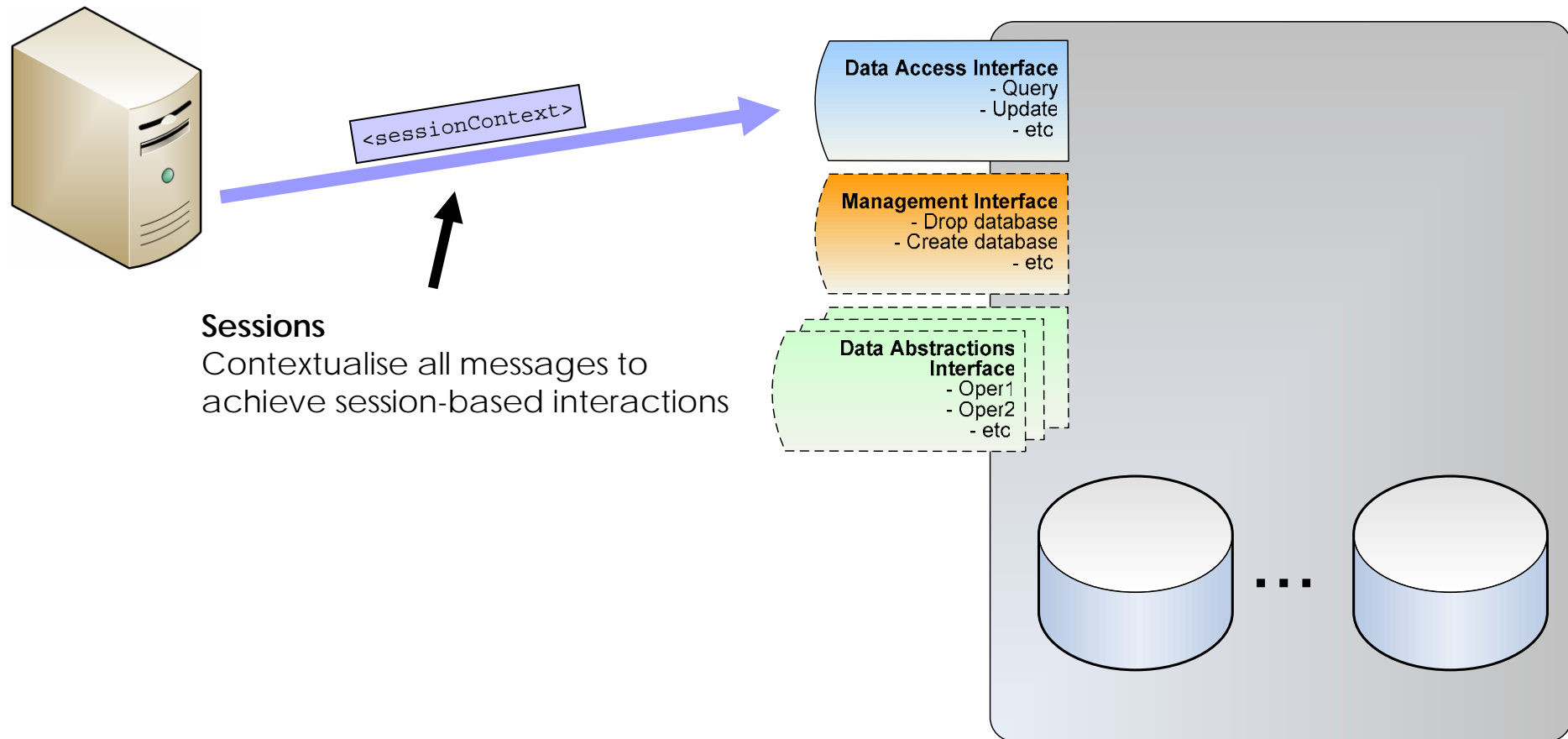


The service may support any number of interfaces that are already defined by DAIS

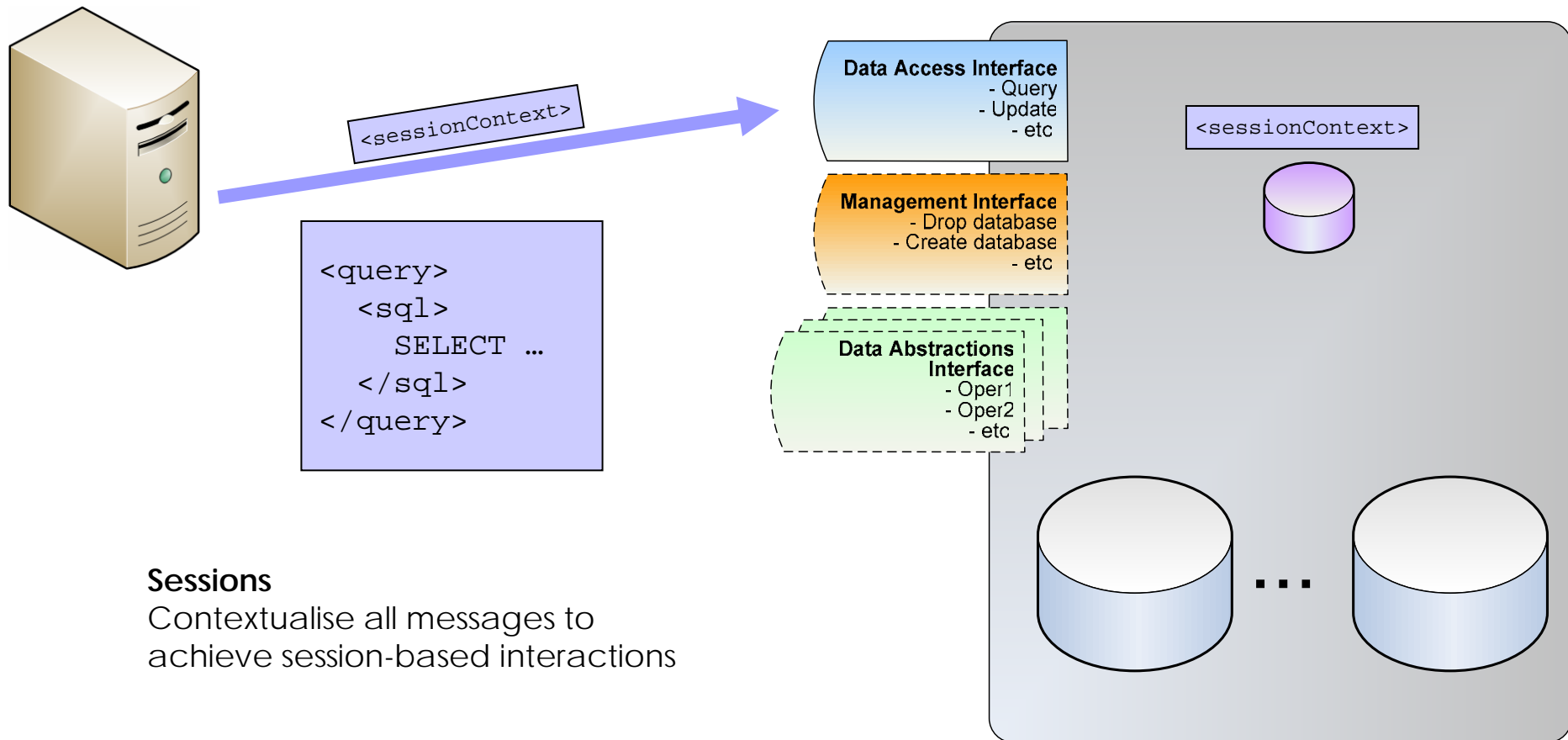


There may be more than one data resources behind a service interface

Session-based interactions



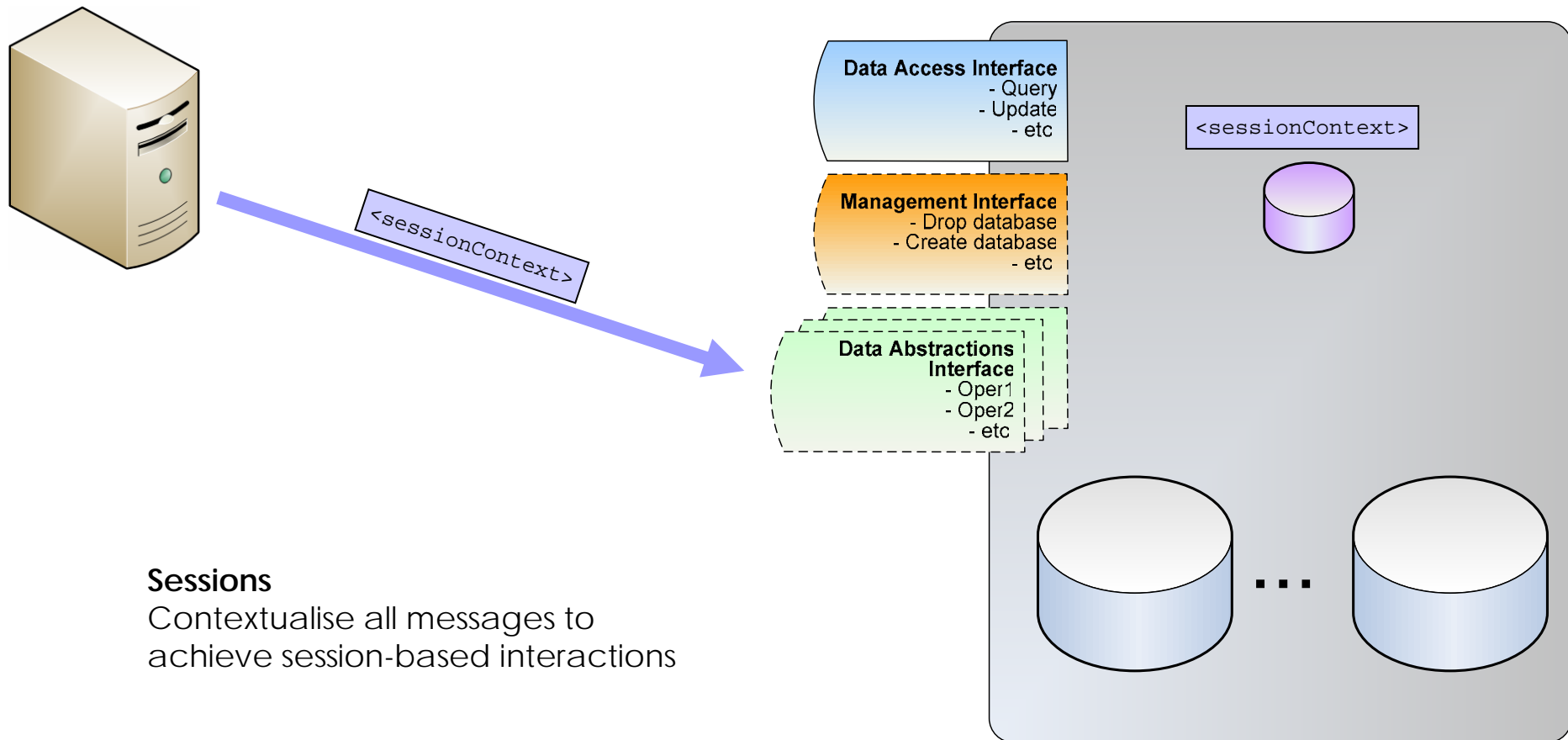
Session-based interactions



Sessions

Contextualise all messages to achieve session-based interactions

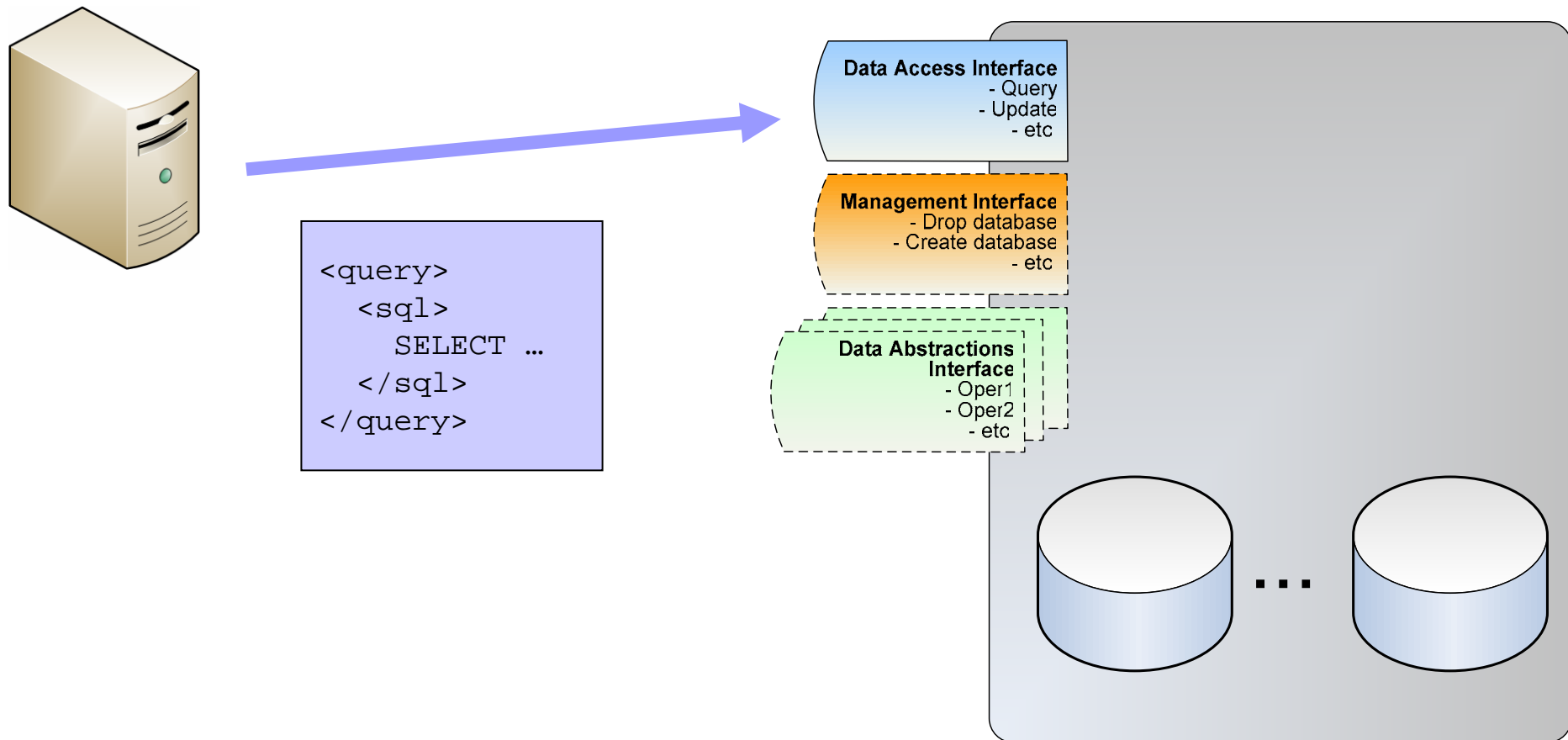
Session-based interactions

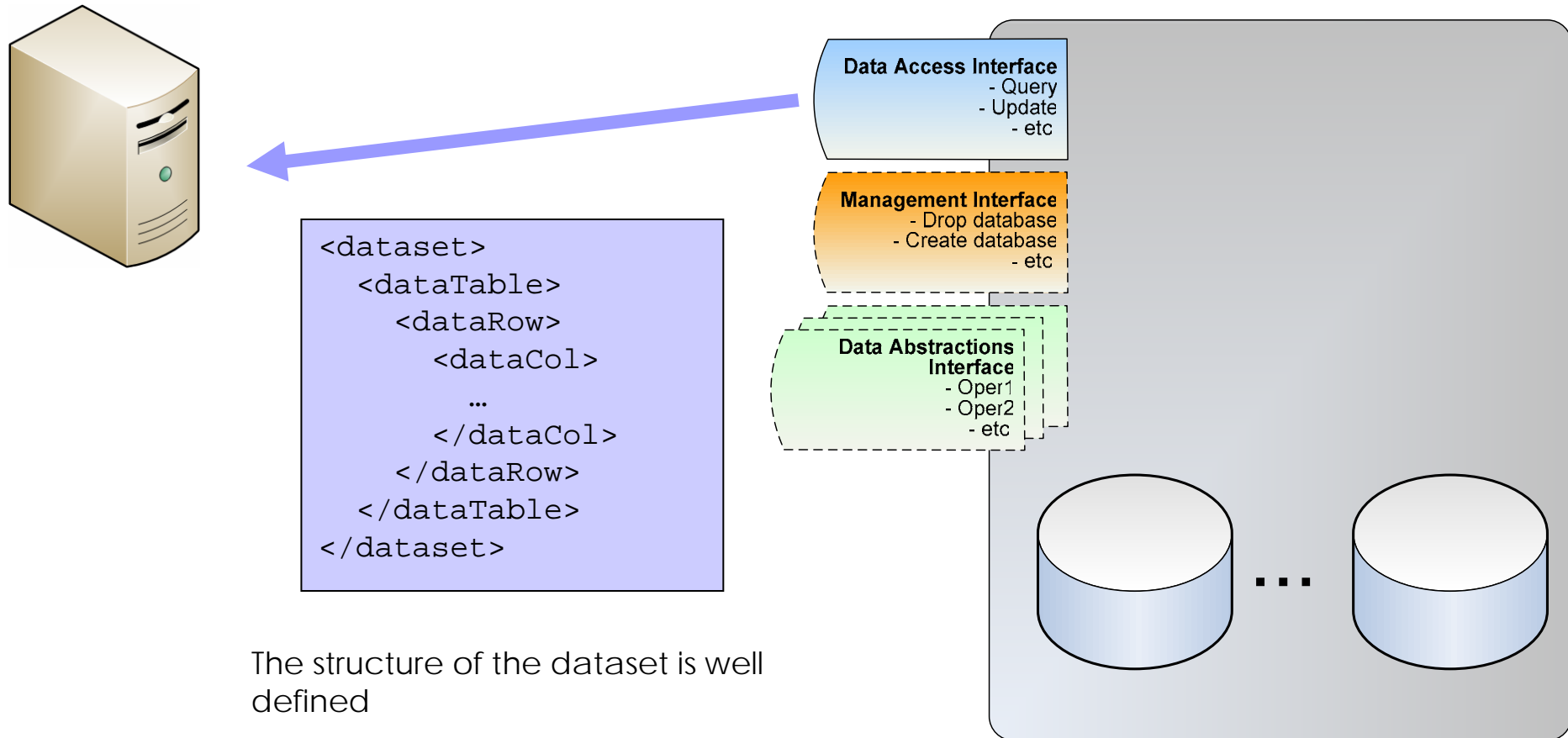


Sessions

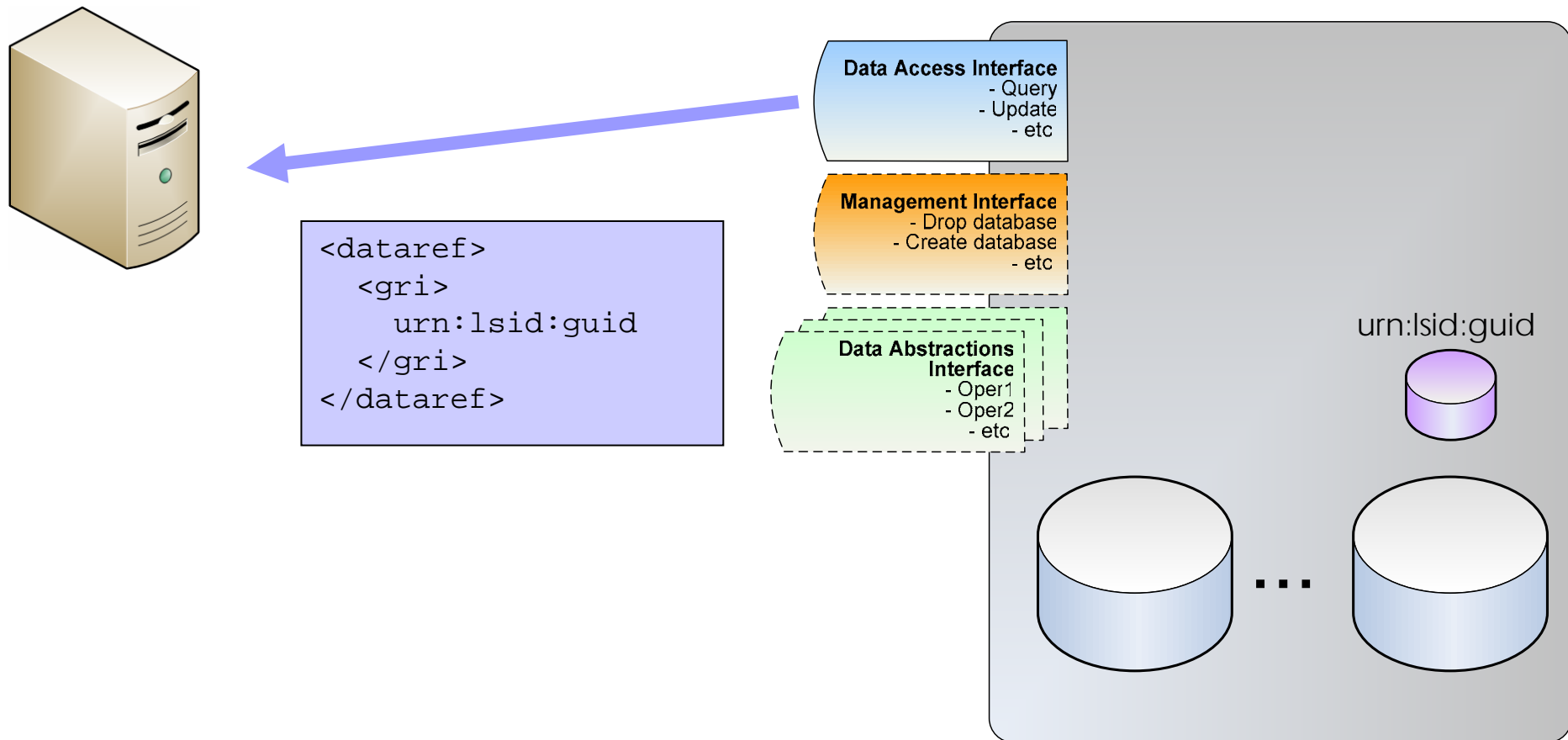
Contextualise all messages to achieve session-based interactions

A query

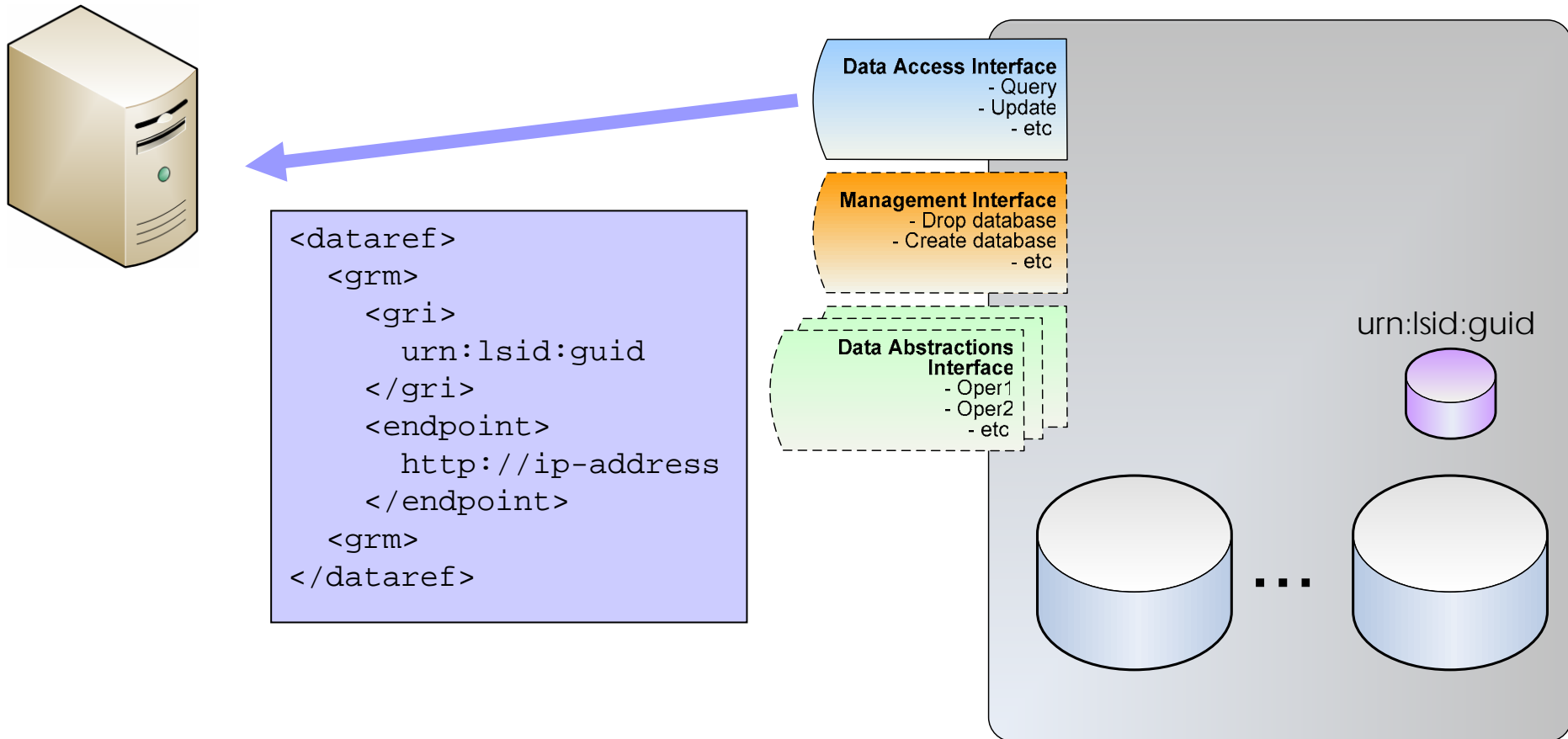




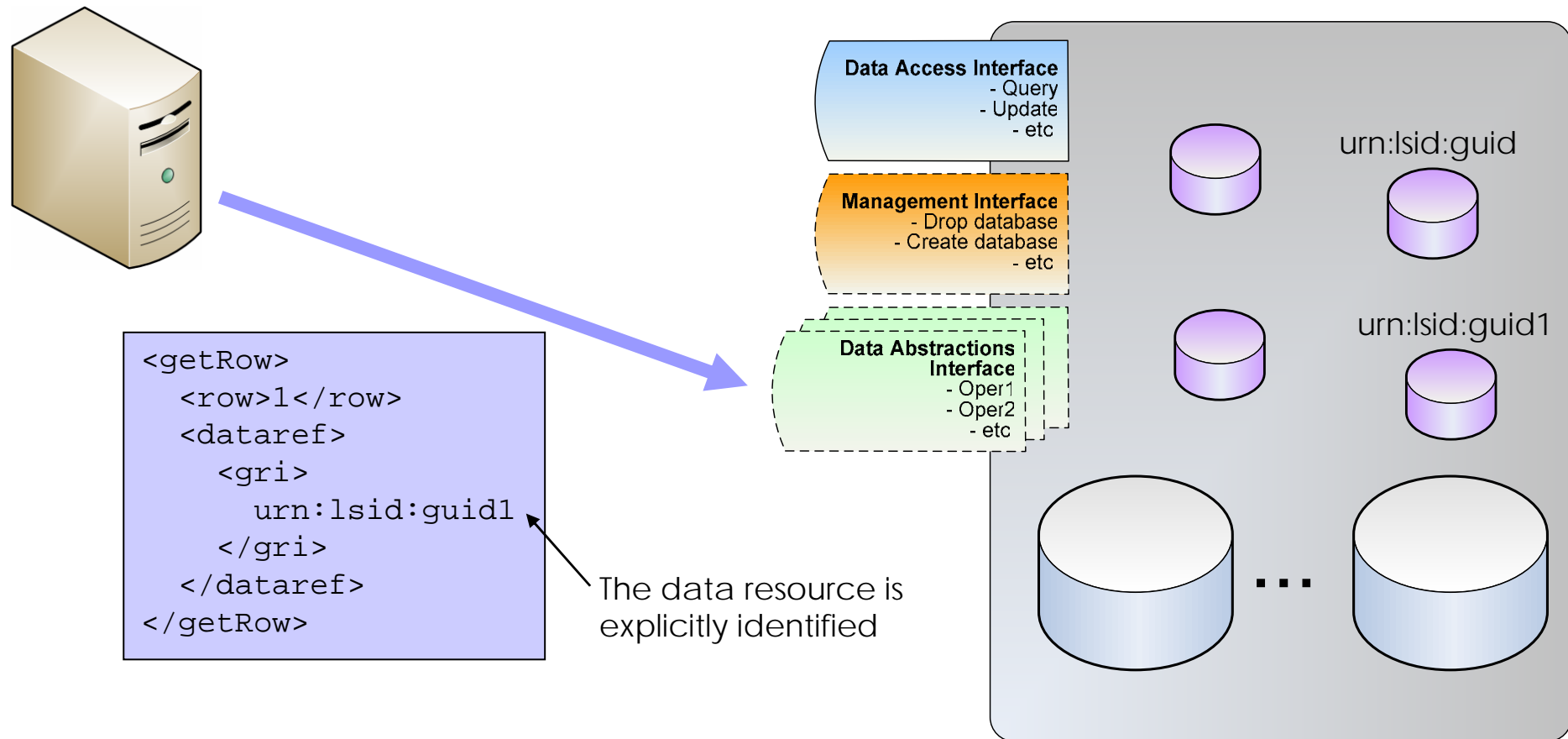
A reference to the result is returned



Metadata document returned



Explicit contextualisation



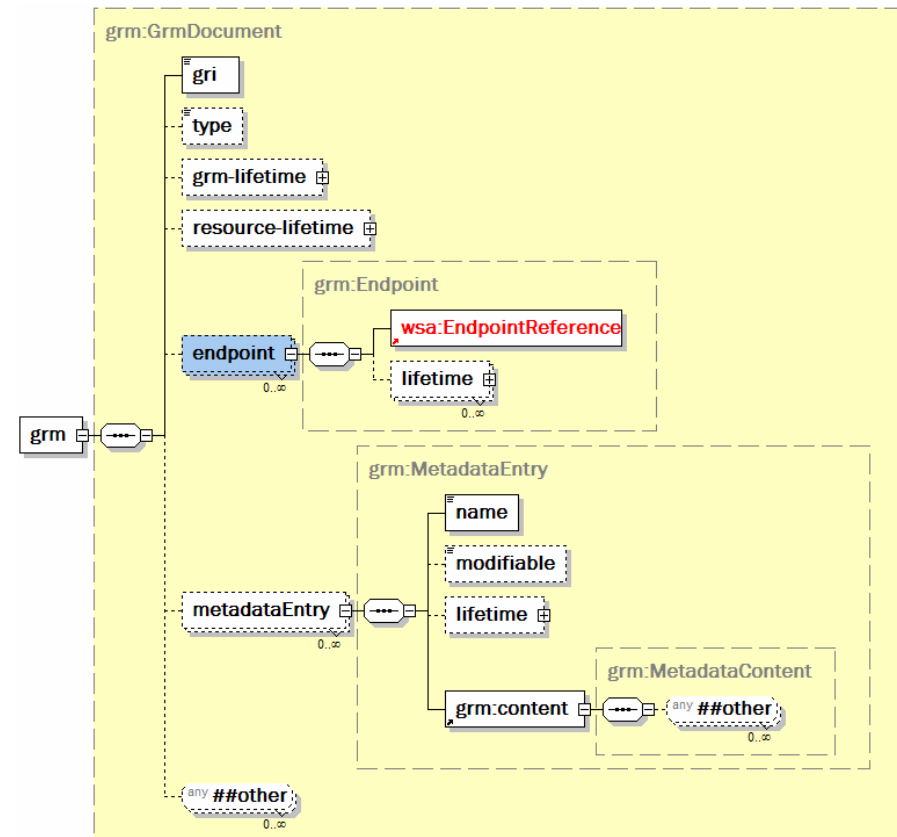
- Metadata about services and data resources can be exposed using documents
- The types of the metadata information already defined by the DAIS spec do not have to change
 - Reuse existing work
- Existing XML Schema and WSDL tooling can be used

- All the DAIS concepts can map to WSA through the use of existing technologies (e.g., XML Schema, WSDL, WS-Context or WS-Addressing)
- Define document formats, not only interfaces
 - DataSet
 - Query documents
 - Result documents
 - Workflow documents Discuss issues of identity
 - e.g., The way exposed DataSets are identified
- Discuss issues of stateful interactions
 - e.g., The use of WS-Context or WS-Addressing as the means for contextualisation

- Discuss interfaces for services (message exchange patterns)
 - e.g., if I receive an "ExecuteQuery" message, I will reply with a "QueryResult" message
- Don't assume 1-1 relationship between a data resource and a service
 - e.g., Use contextualisation to identify resources
- Be independent of specialised tooling
 - e.g., avoid problems introduced by SDEs
 - Return metadata through documents rather than SDEs
- Assume that all additional functionality is orthogonal
 - e.g., Transactions, Security, Agreement
- Exposing resources outside a service's boundaries should be the exception
- Again... don't focus only on interfaces

- The existing work on the DAIS specification can be easily adopted to fit with both the Web Services Architecture and OGSA
- Modularisation is key
- One-to-one association of internal data resources to service instances may lead to fragile distributed applications
- Related work underway in WS-GAF
 - Grid Resource Metadata document and Grid Resource Identifier
 - Metadata service interface
 - Demo (services and browser)

- Grid Resource Metadata Document
- A Browser
- Functionality equivalent to SDEs
- Everything implemented using existing technologies and tooling
- Few Web Services (in Java and .NET)



DEMO

