



the globus alliance  
[www.globus.org](http://www.globus.org)

# Transition and Evolution

## Moving to Grid Services



## GT3 is *\*not\** Grid Services

- There is nothing special about GT3
- GT2 was GT1.1.3 with feature enhancements to the existing components and a lot of new functionality (GridFTP and RC)
- GT3 is all of GT2.4 with feature enhancements to the existing components and a lot of new functionality (OGSI compliant components)



## Terminology

- Everyone (including us) has a bad habit of saying GT3 to mean Grid Services.
- To be clear we should refer to OGSI or non-OGSI compliant services.
- One confusing point:
  - ◆ For the first time, there are two ways to do some things in the toolkit.
  - ◆ I.e., non-OGSI job submission .vs. OGSI compliant job submission



## Transition

- We use the Linux versioning convention
  - ◆ Even number releases, I.e. 3.0, are stable releases (bug fixes only)
  - ◆ Odd number releases, I.e., 3.1 are experimental (feature additions, possible interface changes)
- Our support policy has not changed since GT 2.0.
  - ◆ We support the current and one previous stable release



## The Question

- How long will we support GT2.x?
  - ◆ That is the wrong question
- We invested substantial work in packaging so that we could upgrade components separately.
- We can also End-Of-Life (EOL) them independently as well.
- This is what we will do



the globus alliance

[www.globus.org](http://www.globus.org)

## The Right Question... And the Answer

- When will we EOL specific Components
- Depends on the component, but what we do know is this:
  - ◆ All components will be present in 3.2
  - ◆ The earliest we could EOL something is 3.4 (though we have no plans for this yet)
  - ◆ Therefore, support for all existing components AT LEAST until the release of 3.6, probably longer
  - ◆ This means late 2004, probably well into 2005



## Evolution

- How a specific project makes this transition is a huge question depending on a lot of details.
- Largely depends on how much you have already invested in Grid development
- Obviously, EDG has a significant investment, which makes it harder
- However, there are some techniques that should help.



## OGSI and non-OGSI in Toolkit

- For now, the entire non-OGSI suite is present.
- You can install non-OGSI just like you do today and run production, while working on transition issues.
- We have provided transition clients where we think it is needed or feasible
- Let us know if you think there is something missing.





the globus alliance  
www.globus.org

## Where do you put the OGSI Interface?

- Build an OGSI service to interface to non-OGSI services
  - ◆ RFT to GridFTP
- Maintain your client interface and re-implement to send SOAP messages
  - ◆ globus-job-run
  - ◆ globus-url-copy?
- What if I want to use both OGSI and non-OGSI services?
  - ◆ Must use discovery of some kind to choose appropriate client interface



the globus alliance  
[www.globus.org](http://www.globus.org)

# GT3 Key Points

And a glimpse of things to come



## Web Services

- At the heart of Web services is:
  - ◆ WSDL: Language for defining abstract service interfaces
  - ◆ SOAP (and friends): Binding from WSDL to bytes on the wire
- Web services appears to offer a fighting chance at ubiquity (unlike CORBA)
- But Web services does not go far enough to serve a common base for the Grid...



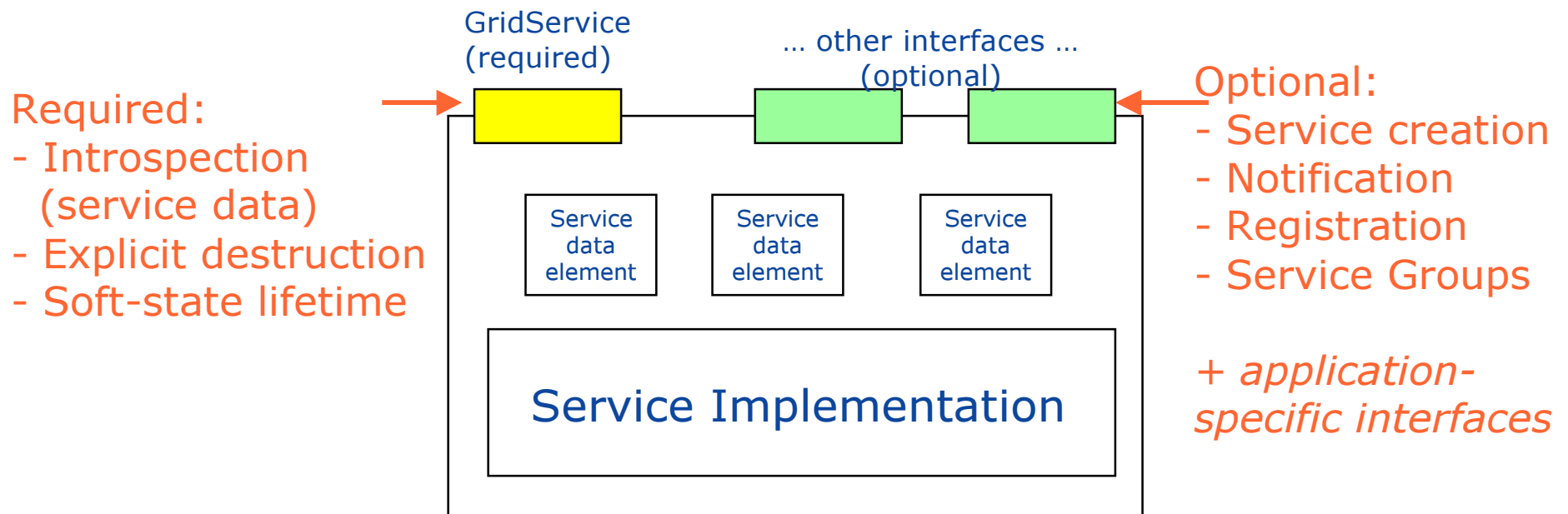
# Transient Service Instances

- “Web services” address discovery & invocation of persistent services
  - ◆ Interface to persistent state of entire enterprise
- In Grids, must also support transient service instances, created/destroyed dynamically
  - ◆ Interfaces to the states of distributed activities
  - ◆ E.g. workflow, video conf., dist. data analysis, subscription
- Significant implications for how services are managed, named, discovered, and used
  - ◆ In fact, much of Grid is concerned with the management of service instances



# GT3 Core: OGSI Specification

The Specification Defines how Entities can Create, Discover and Interact with a Grid Service



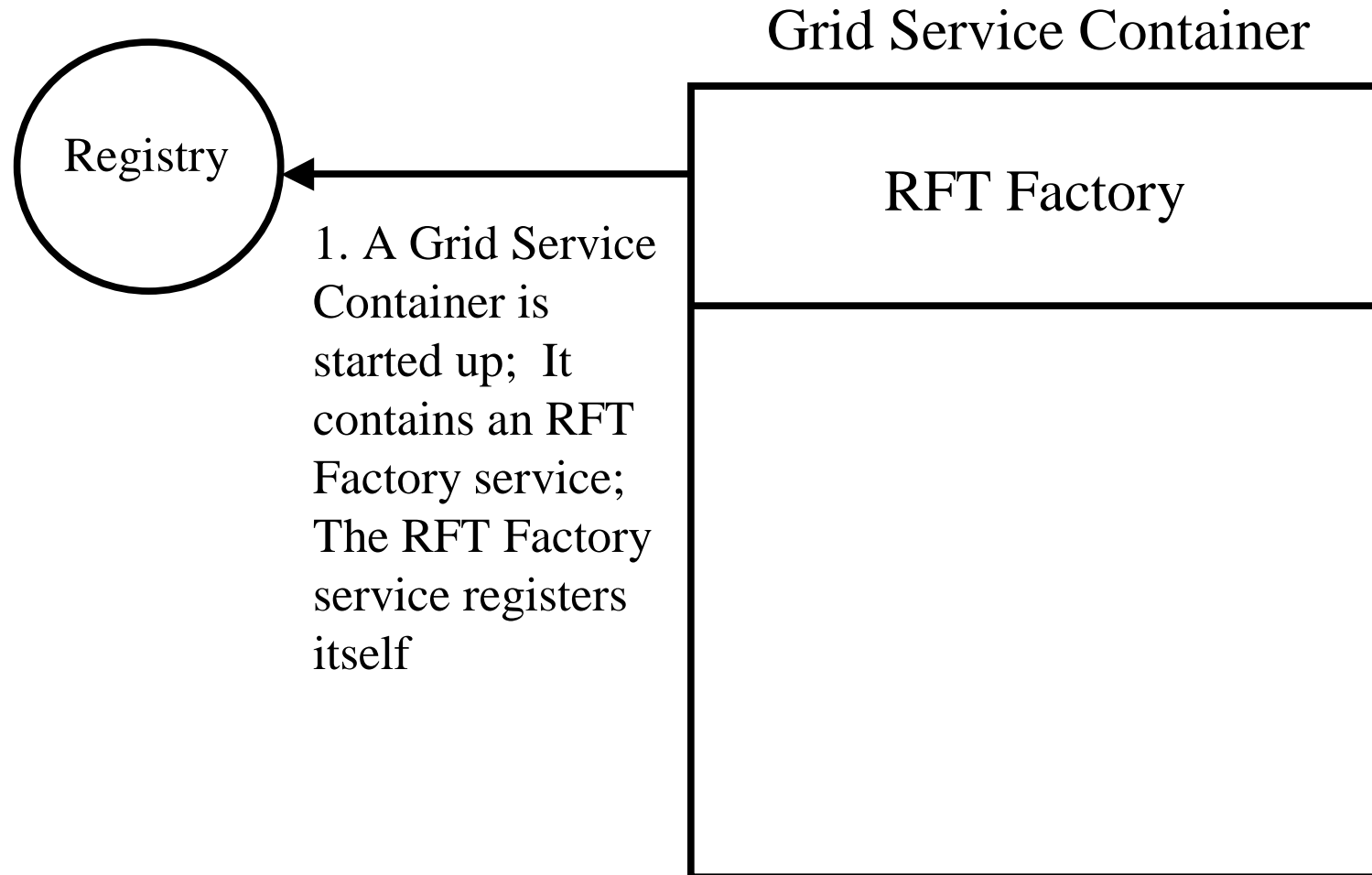
## Service locator

Includes 0 or more Grid Service Handles (GSHs)

Includes 0 or more Grid Service References (GSRs)



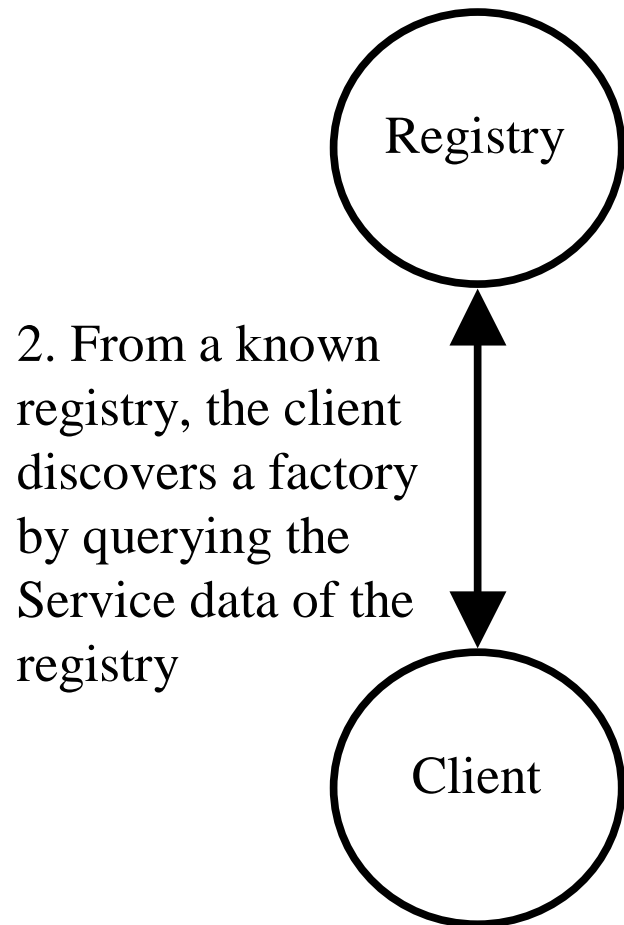
# RFT in Action



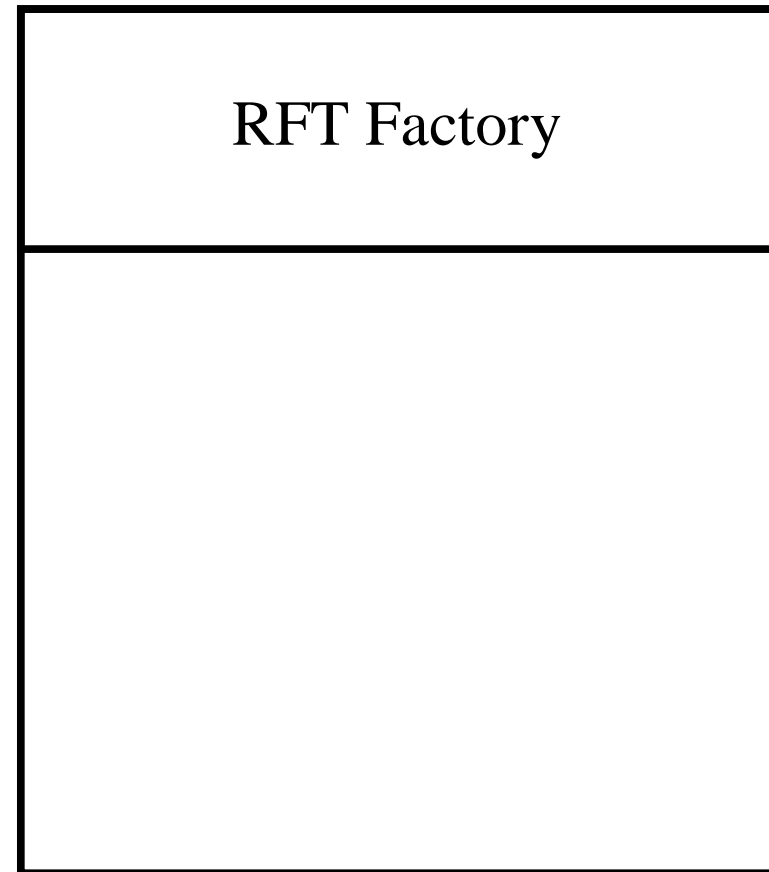
\* The scenarios in this presentation are offered as examples and are not prescriptive



## RFT in Action



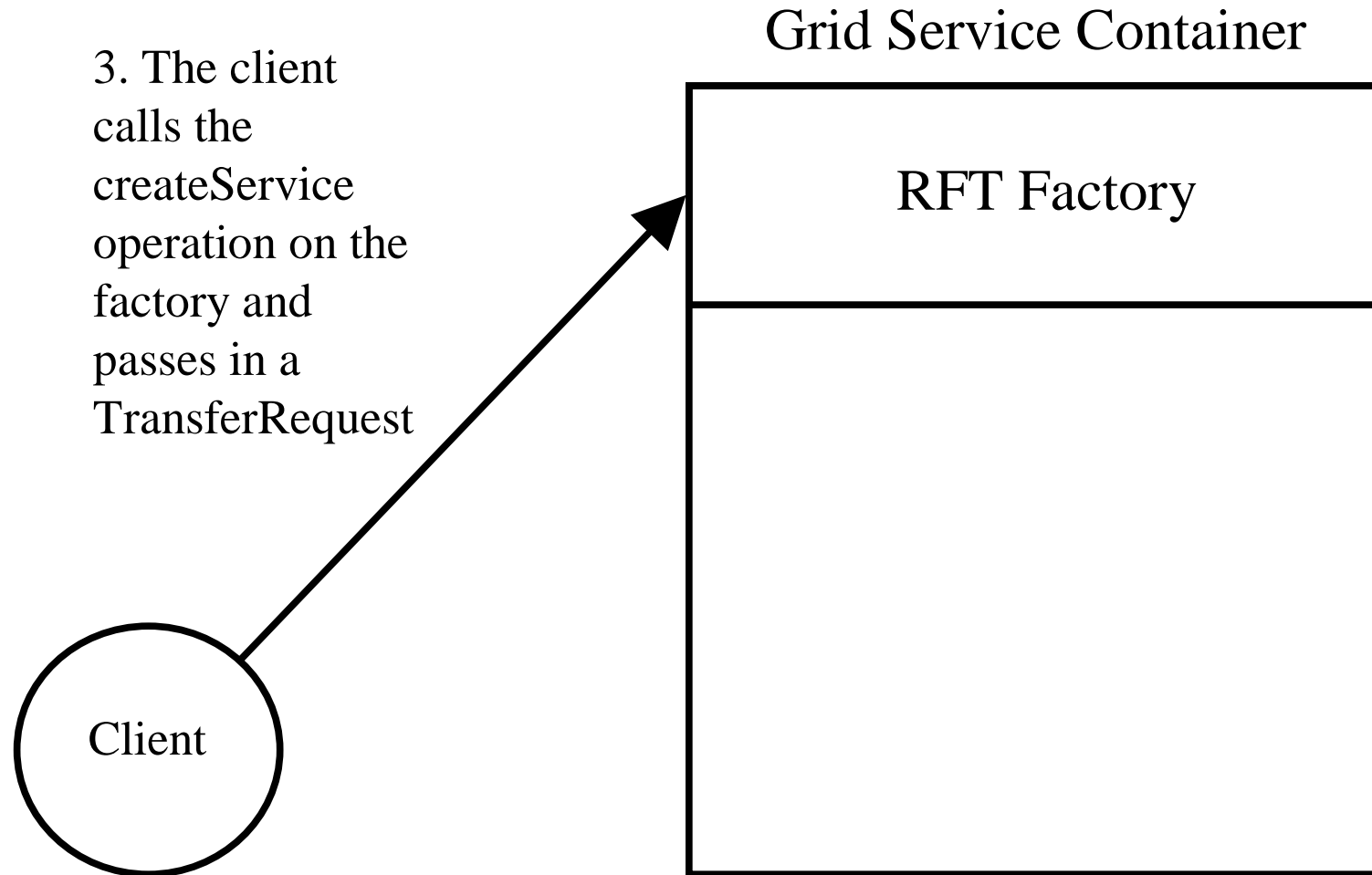
### Grid Service Container



\* The scenarios in this presentation are offered as examples and are not prescriptive



## RFT in Action

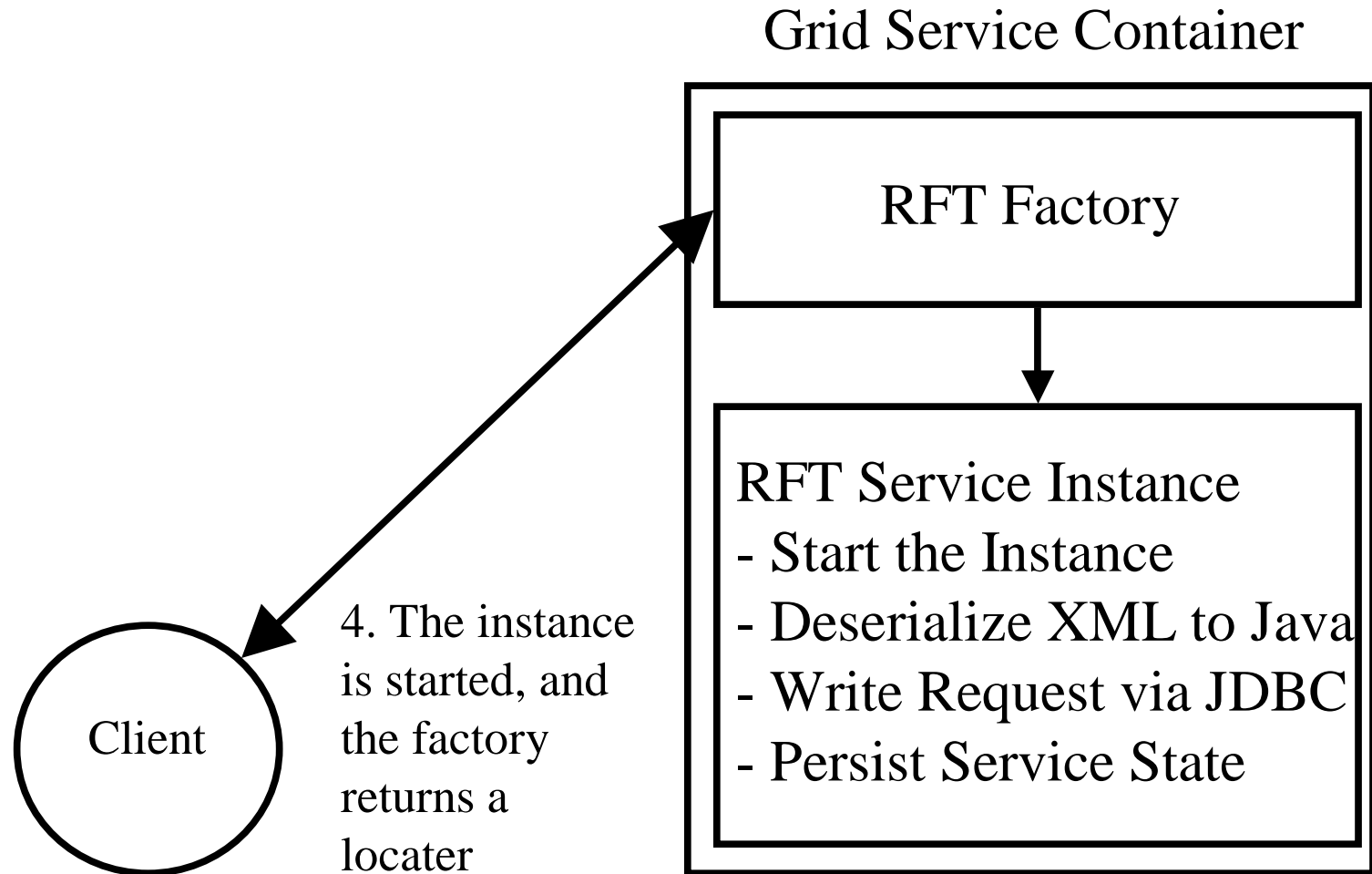


\* The scenarios in this presentation are offered as examples and are not prescriptive





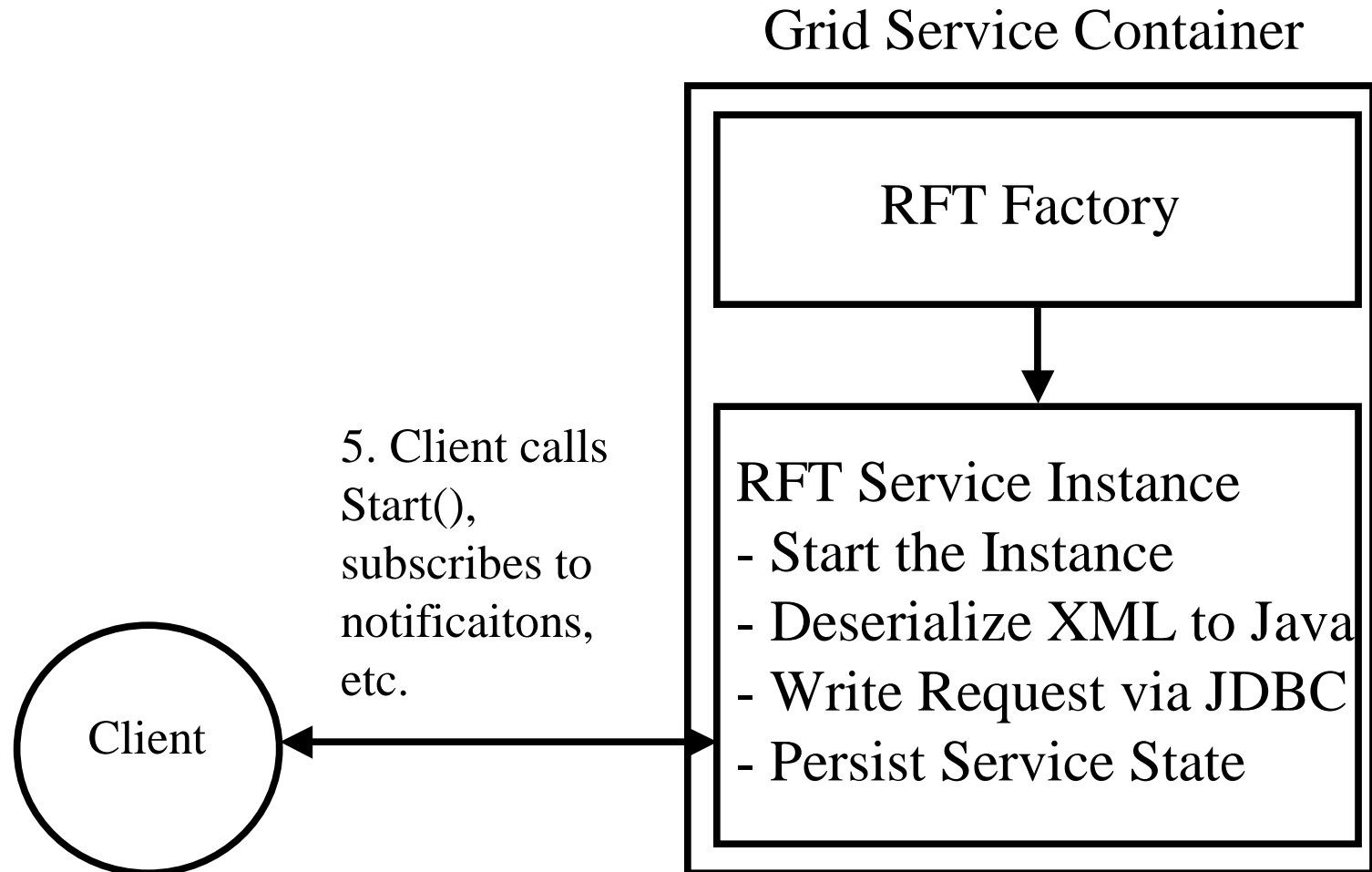
## RFT in Action



\* The scenarios in this presentation are offered as examples and are not prescriptive



# RFT in Action

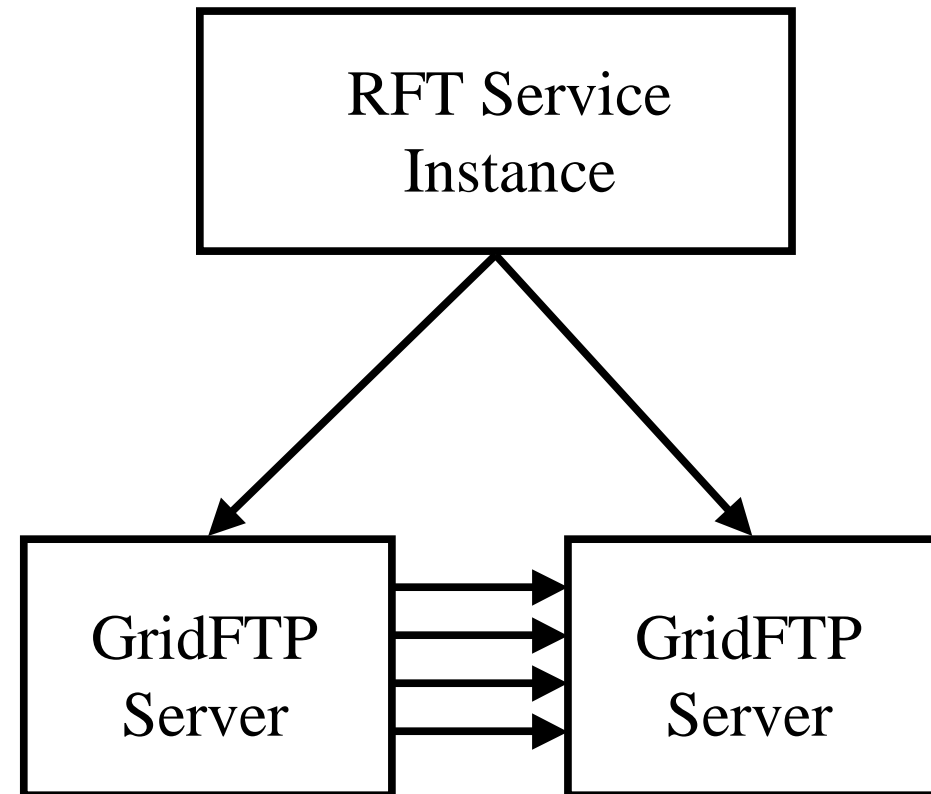


\* The scenarios in this presentation are offered as examples and are not prescriptive



## RFT in Action

- Service is OGSI compliant
- Uses existing GridFTP (non-OGSI) protocols and tools to execute 3<sup>rd</sup> Party Transfer for the user
- Provides extensive state transition notification



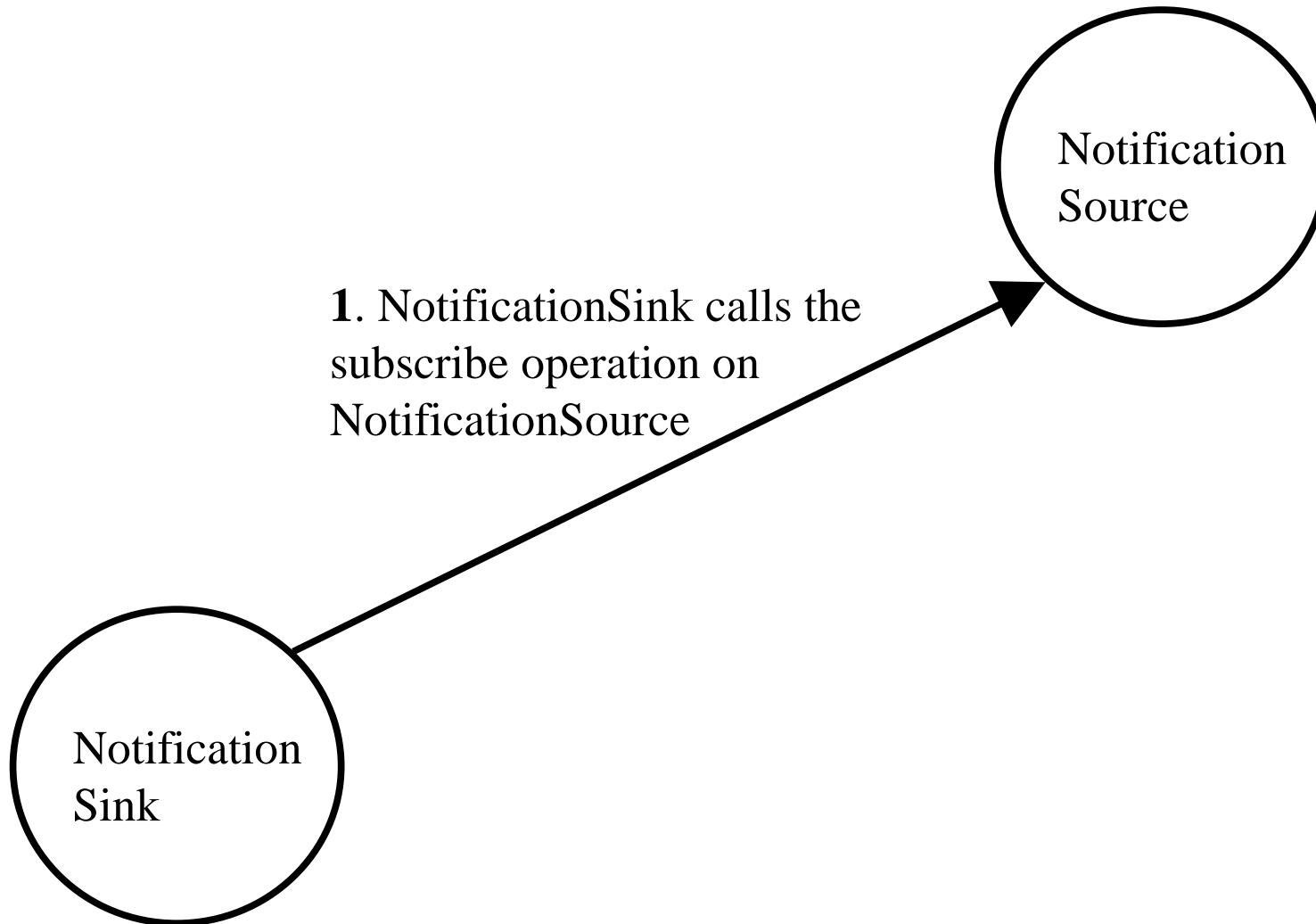
\* The scenarios in this presentation are offered as examples and are not prescriptive



the globus alliance

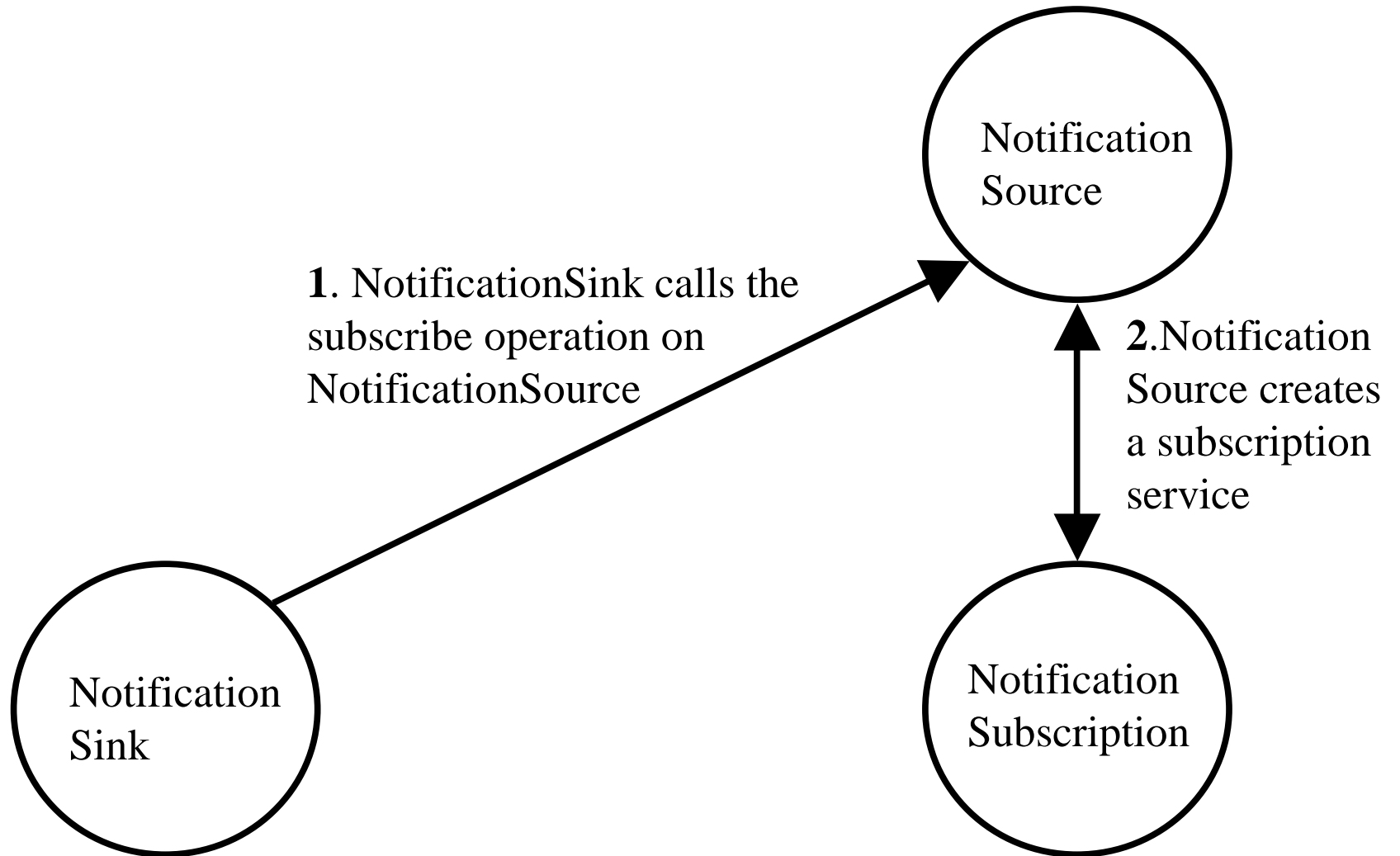
www.globus.org

# A Notification Scenario



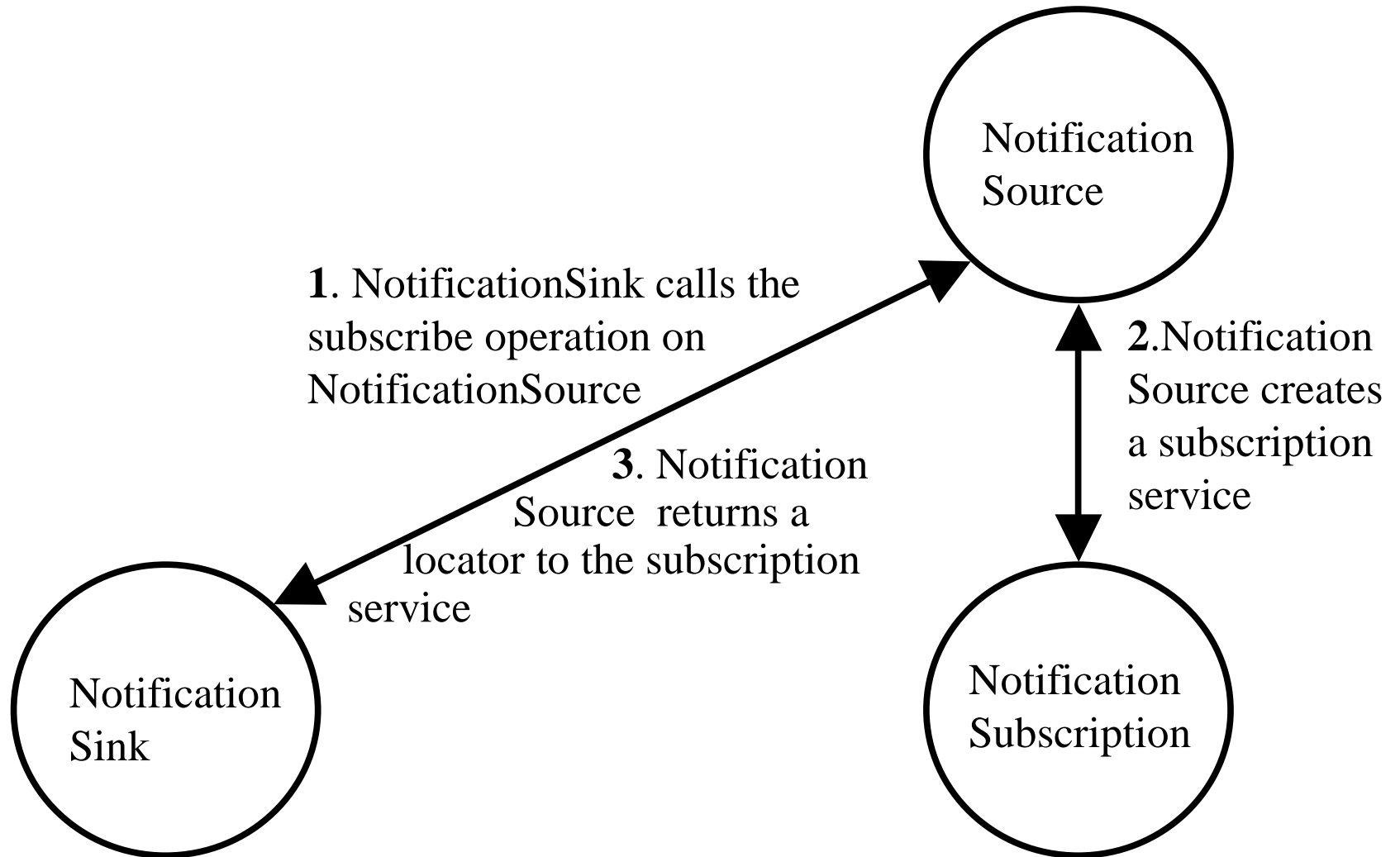


# A Notification Scenario



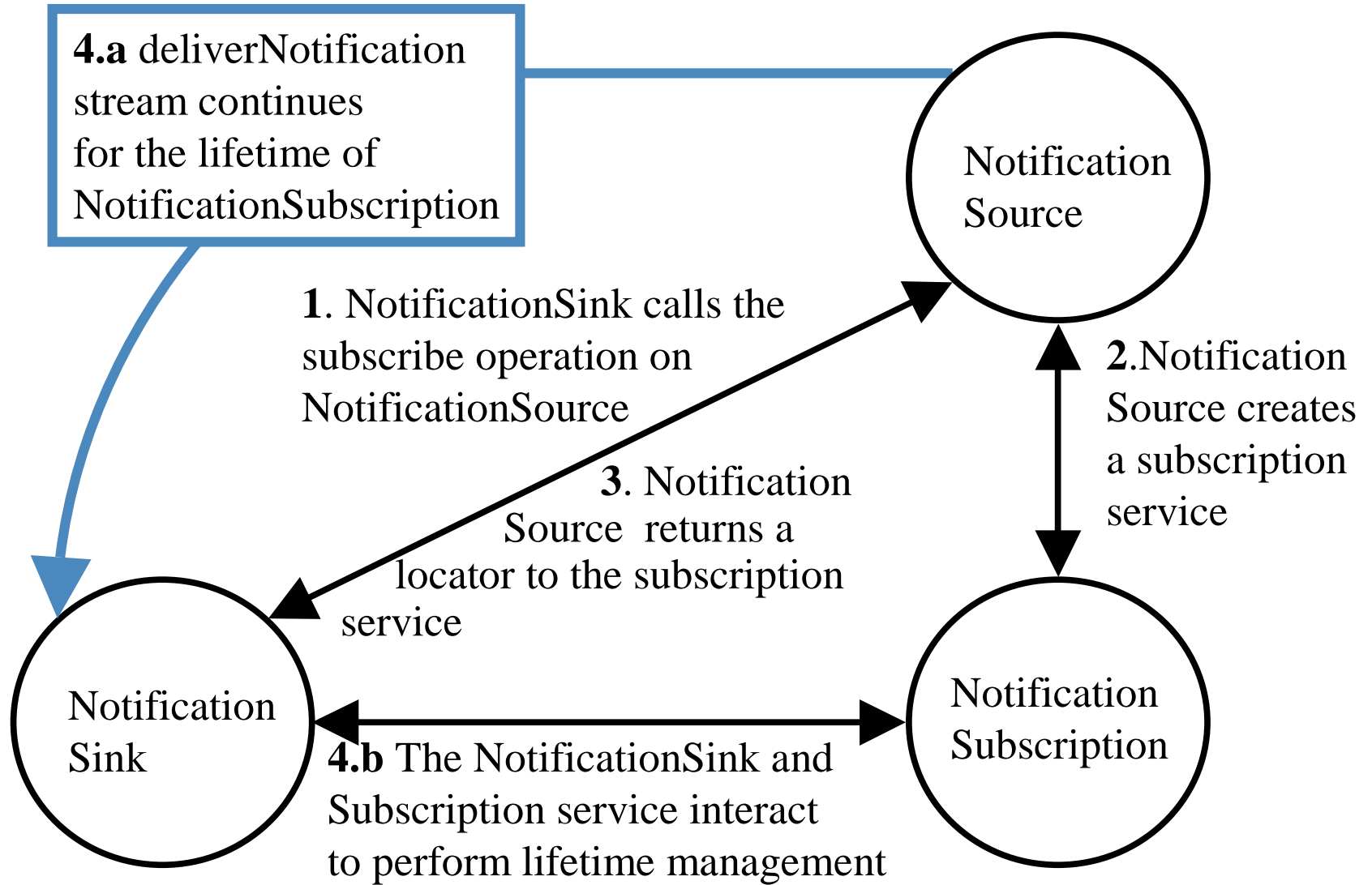


# A Notification Scenario



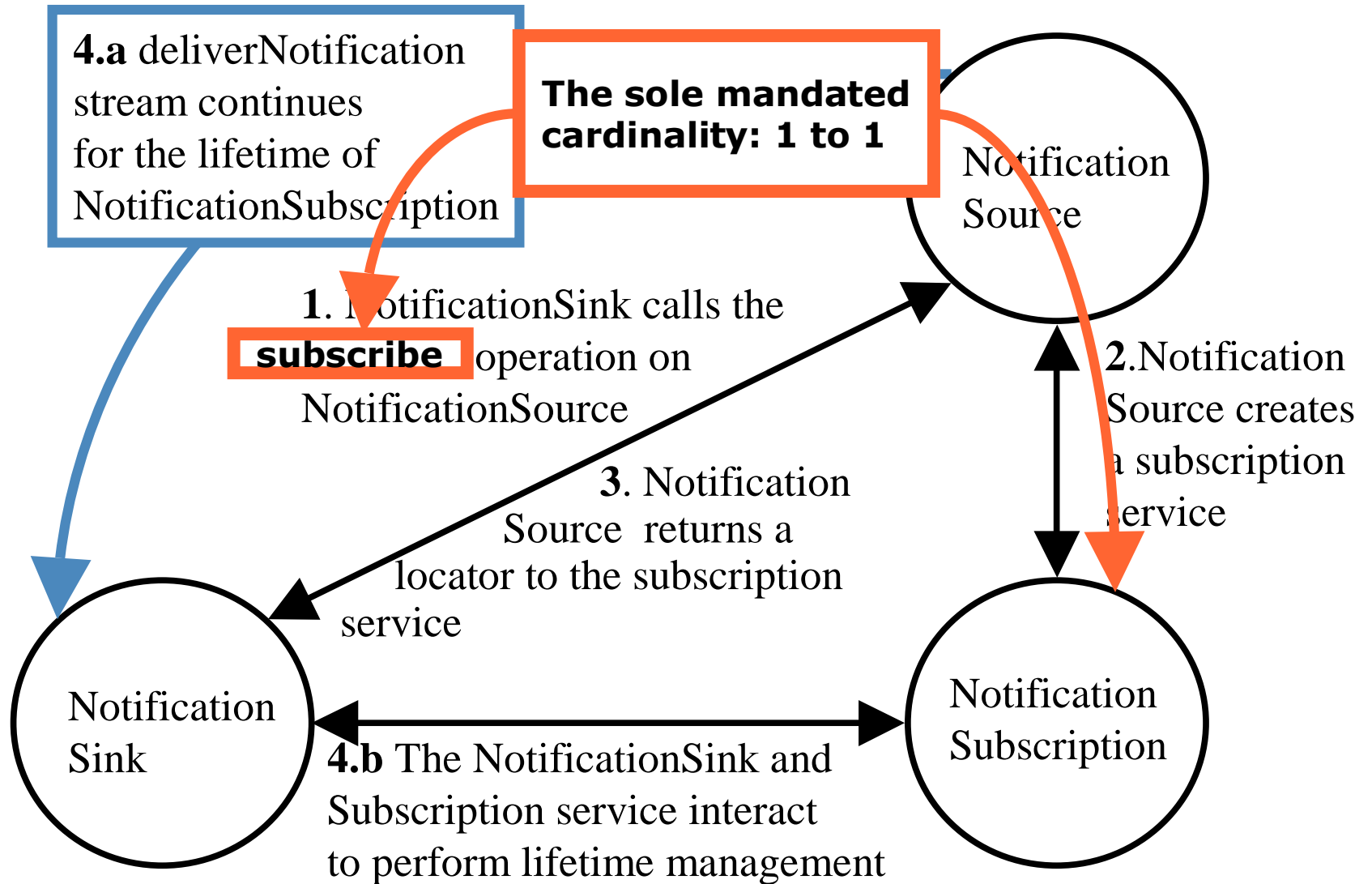


# A Notification Scenario





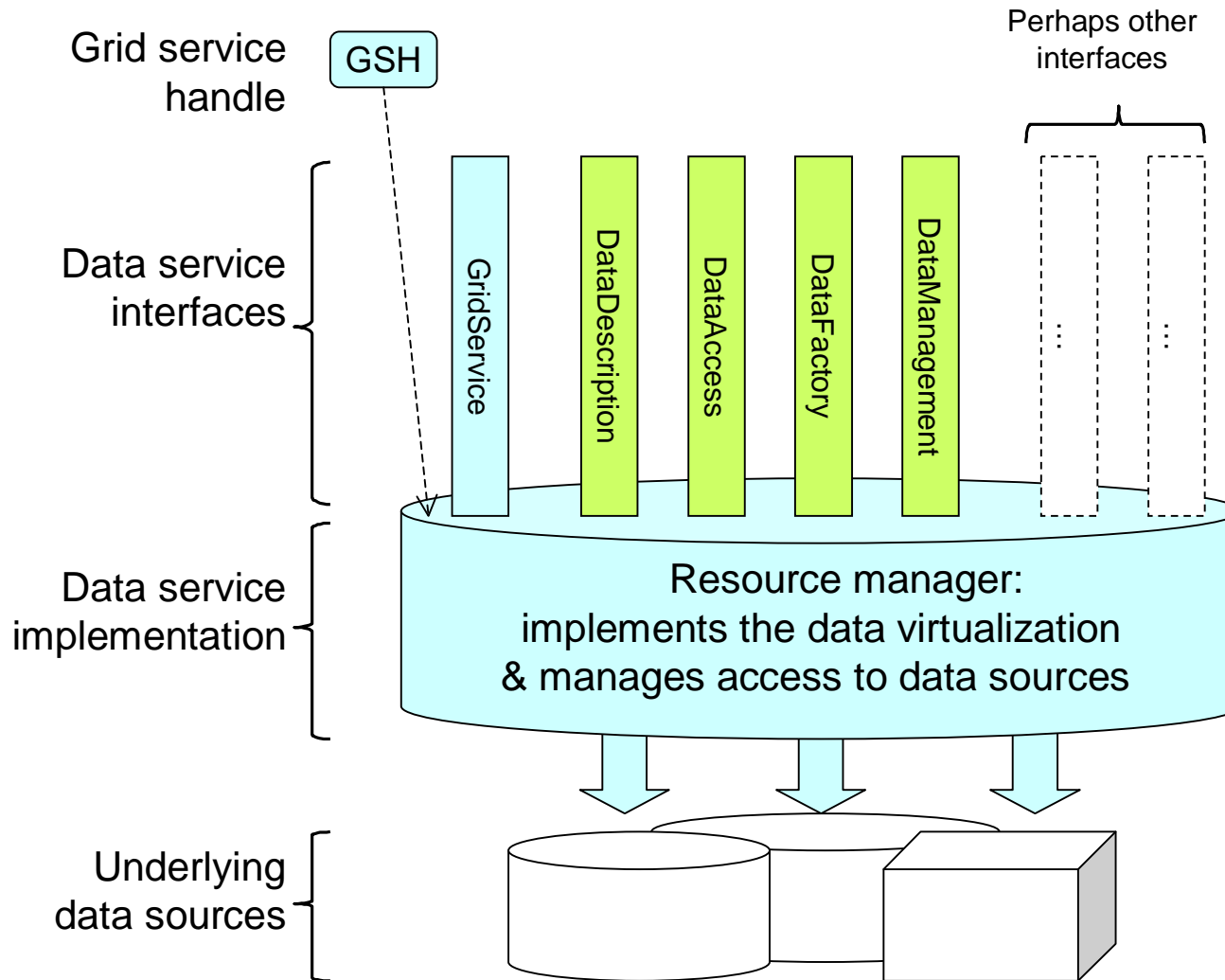
# A Notification Scenario





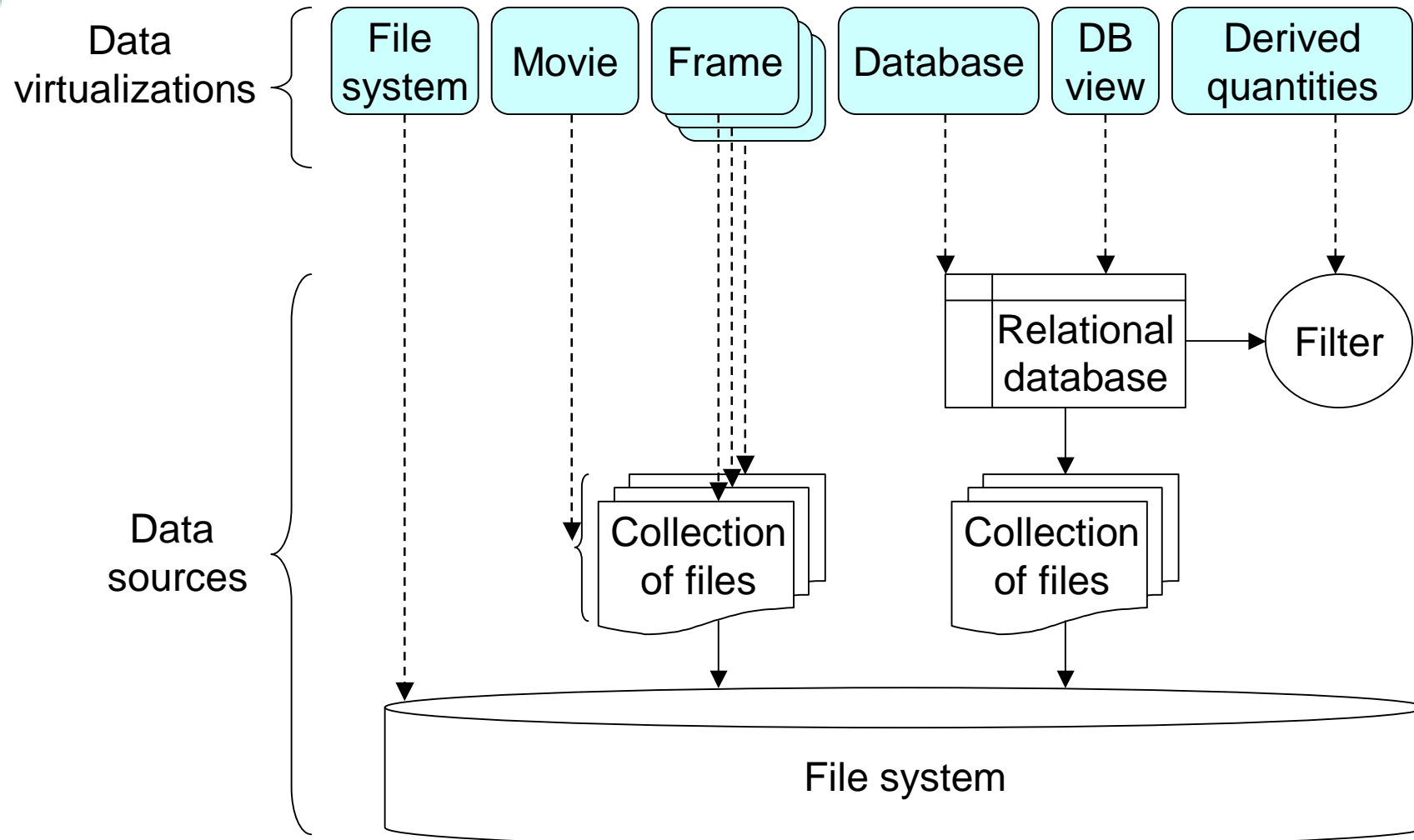


# Data Service Overview





# Multiple Virtualizations Example





## (OGSI-) WS-Agreement

- Recall key criteria of a Grid:
  - ◆ *Coordinates resources that are not subject to centralized control ...*
  - ◆ using standard, open, general-purpose protocols and interfaces ...
  - ◆ *to deliver non-trivial qualities of service.*
- Implies need to express and negotiate agreements that govern the delivery of services to clients
  - ◆ Agreement = what will be done, QoS, billing, compliance monitoring
- All interesting Web/Grid services interactions will be governed by agreements!



# WS-Agreement Contents

- Standard agreement *language*
  - ◆ A composition of a set of terms that govern a service's behavior with respect to clients
  - ◆ Agreement language uses WS-Policy (currently)
  - ◆ Standard attributes for terms that express current state of negotiation
  - ◆ Other groups define specific terms
- Standard agreement negotiation *protocol*
  - ◆ Establish, monitor, re-negotiate agreement
  - ◆ Expressed using OGSII GWSDL interfaces
  - ◆ Each agreement represented by a service



# Agreement Overview

**Steps (Operations):**

- [1] Create Agreement
- [2] Create Data Service
- [3] Access Data Service
- [4] Monitor Agreement

