



Enabling Grids for
E-science in Europe

www.eu-egee.org

This product includes material developed
by the Globus Project (<http://www.globus.org/>).

Excercise 3

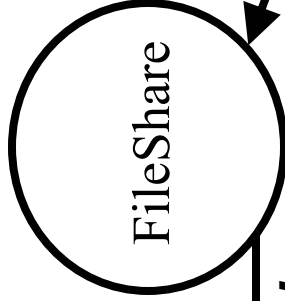


How to Build a Grid Service Using GT3

- Overview of Grid Services and GT3
 - Build a Grid Service
 - Overview
 - 1. Deployment: Stand Up a FileShare Service
 - 2. Naming: Share Files using Identifiers
 - **3. Inspection: Add Service Data**
 - 4. Virtual Organization: Register with a Community Index
 - 5. Lifetime Management: Maintain service registration
 - 6. Discovery: Find a File
 - 7. GT3 Security: Share Files Securely
- Time permitting:**
- 8. Transience: Create and Destroy FileShares
- Publish your Grid Service: The GTR

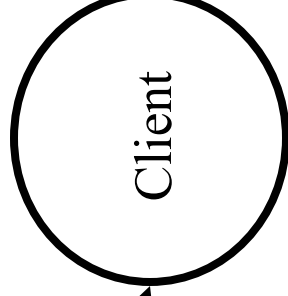
Exercise 3: Inspection

1. Deployment:
Stand up a
FileShare
service on
your laptop



3. Inspection:
Add service
data to your
service

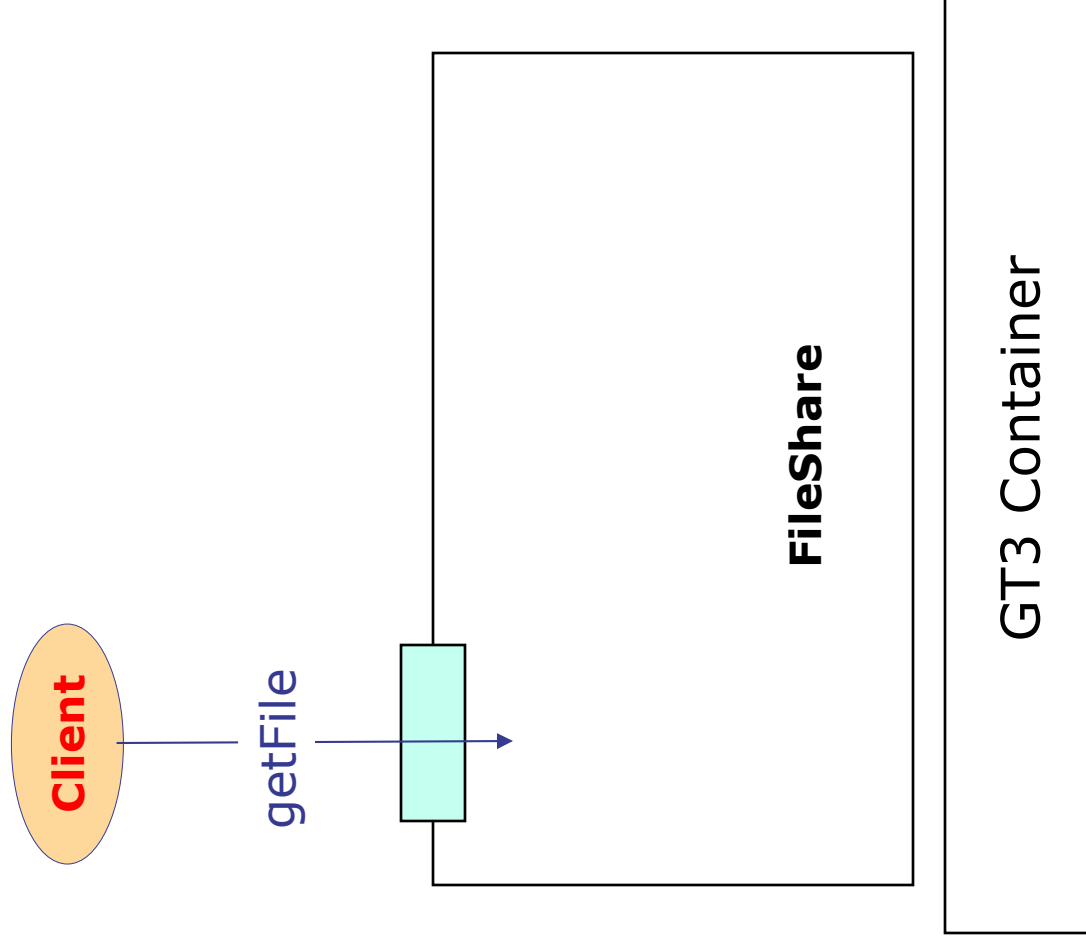
2. Service Naming:
Share files using identifiers



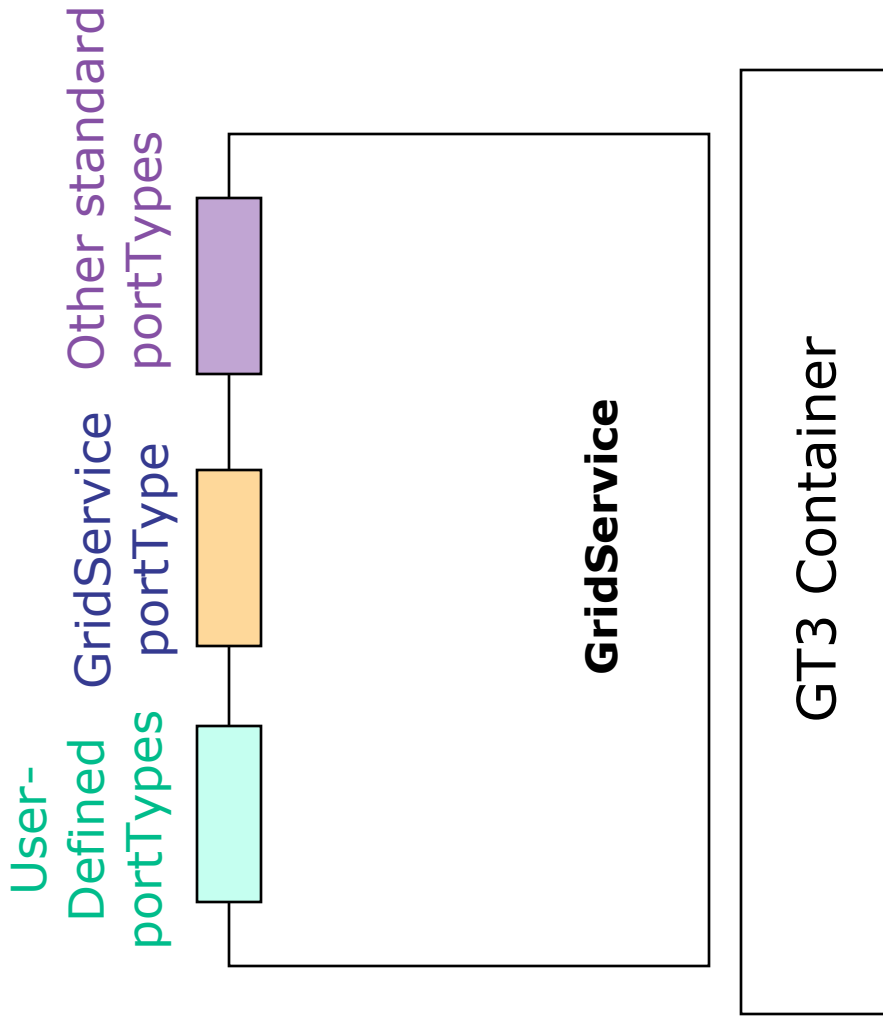
Service Data

- Any Grid Service can expose internal state as Service Data Elements
 - An XML element of arbitrary complexity
- Each service has a set of Service Data Elements

Service Interfaces

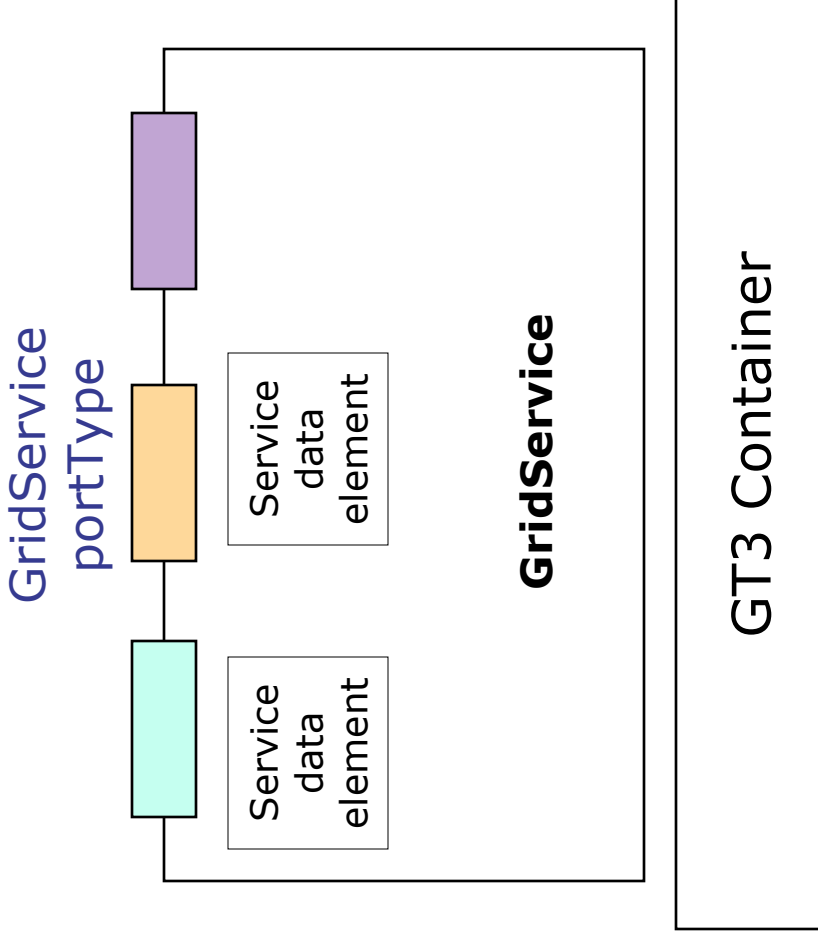


Types of Interfaces

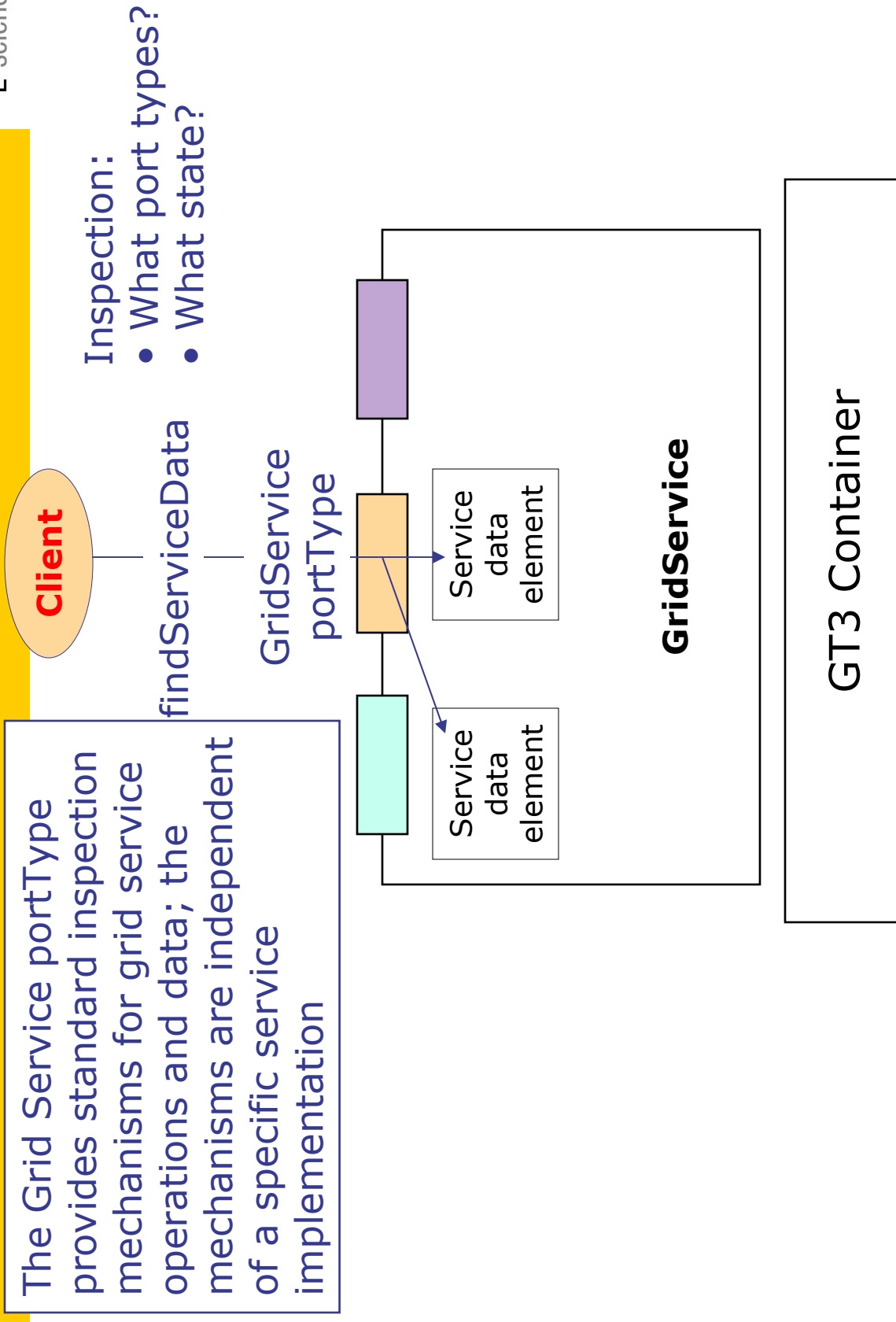


Grid Service portType

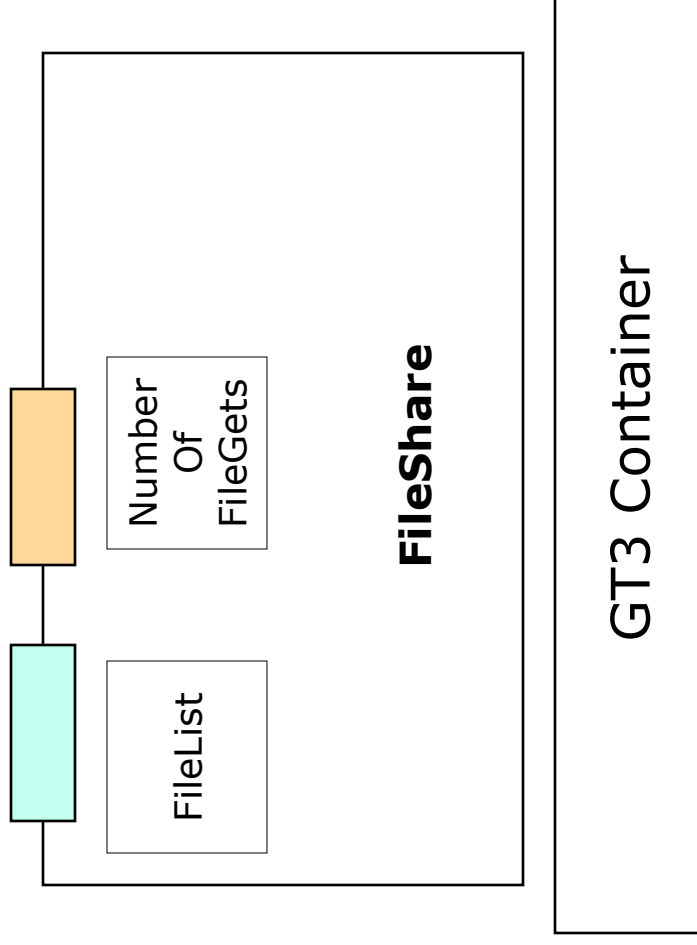
The GridService portType is mandated in OGSI



Inspection



FileShare Service Data



WSDL

- Web Service Description Language
- XML-based language for:
 - Abstractly describing message exchanges between clients and services
 - Types defined using XML Schema
 - Message comprising one or more parts of XML Schema types/elements
 - Operation = input/output or input only messages
 - Interface = named group of operations
 - Binding the interfaces to concrete protocols
 - E.g. Soap/http
- It says nothing about what messages are sent

GWSDL

- OGSI requires interface extension/composition
- We worked within W3C WSDL working group to define standard interface extension in WSDL 1.2 that meets OGSI requirements
- But could not wait for WSDL 1.2
- So defined gwSDL:portType that extends WSDL 1.1 portType with:
 - WSDL 1.2 portType extension
 - WSDL 1.2 open content model
- Define GWSDL → WSDL 1.1 & 1.2 mappings

GWSDL (Cont.)

- All operations on our service are defined in our GWSDL
- We provide a definition of our custom portTypes
- We obtain standard Grid Service operations by extending the GridService portType
 - `<grid:portType name="FileSharePortType" extends="ogsi:GridService">`

SDEs in GWSDL

- In addition to operations, SDEs are defined in GWSDL

```
<gwsdl:serviceData name="SDEName"
```

```
  type="xsd:int"
```

```
  maxOccurs="m"
```

```
  minOccurs="n"
```

```
  mutability="mutable">
```

```
</gwsdl:serviceData>
```

Namespaces

- All our operations are defined inside of XML namespaces
- <definitions
 xmlns:tns="http://ogsa.globus.org/
 samples/2003/09/fileshare">
- Our SDEs are also defined inside of namespaces.
We will need to know those namespaces when we query
- For example:
 <http://ogsa.globus.org/samples/2003/09/fileshare:FileList>

XML Types

- Several types are predefined
 - `xsd:int`
- Also can specify more complex datatypes
 - `<xsd:complexType name="foo">`
- Creating new types is out of scope for this tutorial. The types you need for your SDEs have been defined in the GWSDL

What Attendees should Do

- Uncomment serviceData from fileshare_port_type.gwsdl
- Uncomment SDE update code in FileShareImpl.java
- Verify your work by using handy client: ogsi-find-servicedata-by-name
 - Use the pre-built client to inspect service data by name
 - Observe how service data changes over time

What Attendees Should See

- Output of ogsi-find-service-data-by-name

```
[...]  
<ns3:NumberOfFileGets  
  [...]  
  xsi:type="xsd:int">  
  0  
</ns3:NumberOfFileGets>  
[...]
```

Exercise 3 Review

- SDEs are defined in GWSDL
- You can extend GWSDL to get pre-defined operations, like `findServiceData`
- SDEs use XML types
- SDEs are namespace qualified
- The `GridService portType` is mandated by OGSF

What We've Covered So Far

