the globus alliance
www.globus.org

How to Build a Service Using GT4

Globus Alliance Staff

Rachana Ananthakrishnan, Charles Bacon,
Lisa Childers, Jarek Gawor, Joe Insley,
Argonne National Laboratory
Ben Clifford, USC/Information Sciences Institute

Please Ensure that the Following Software is Correctly Installed on Your Laptop Immediately:

- Jakarta Ant 1.5 or 1.6
- Jdk 1.4.2

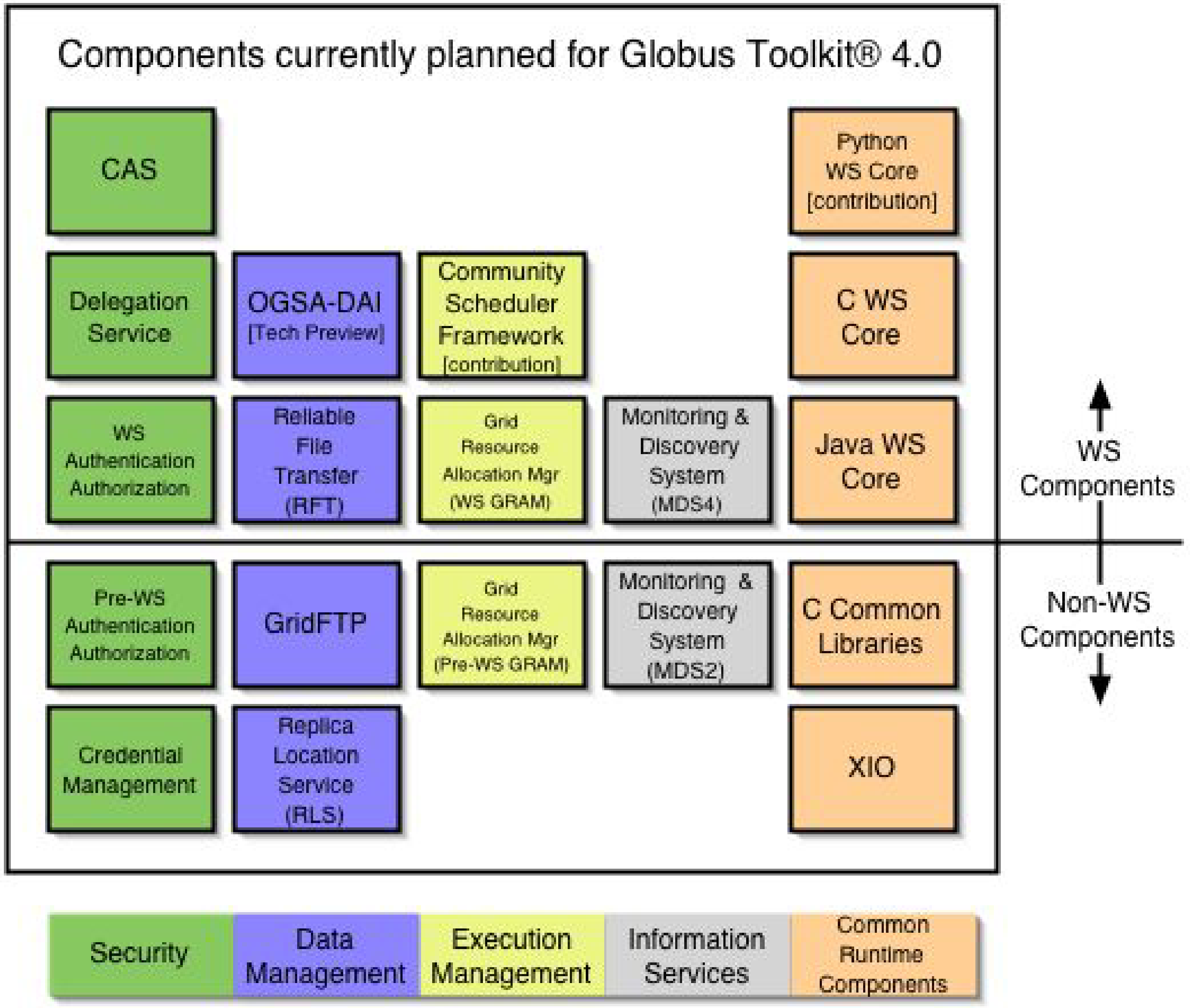
Please download (but do not install) the code bundles located here:

http://www-unix.mcs.anl.gov/~childers/tutorials/BAS/GW2005/index.html

How to Build a Service Using GT4

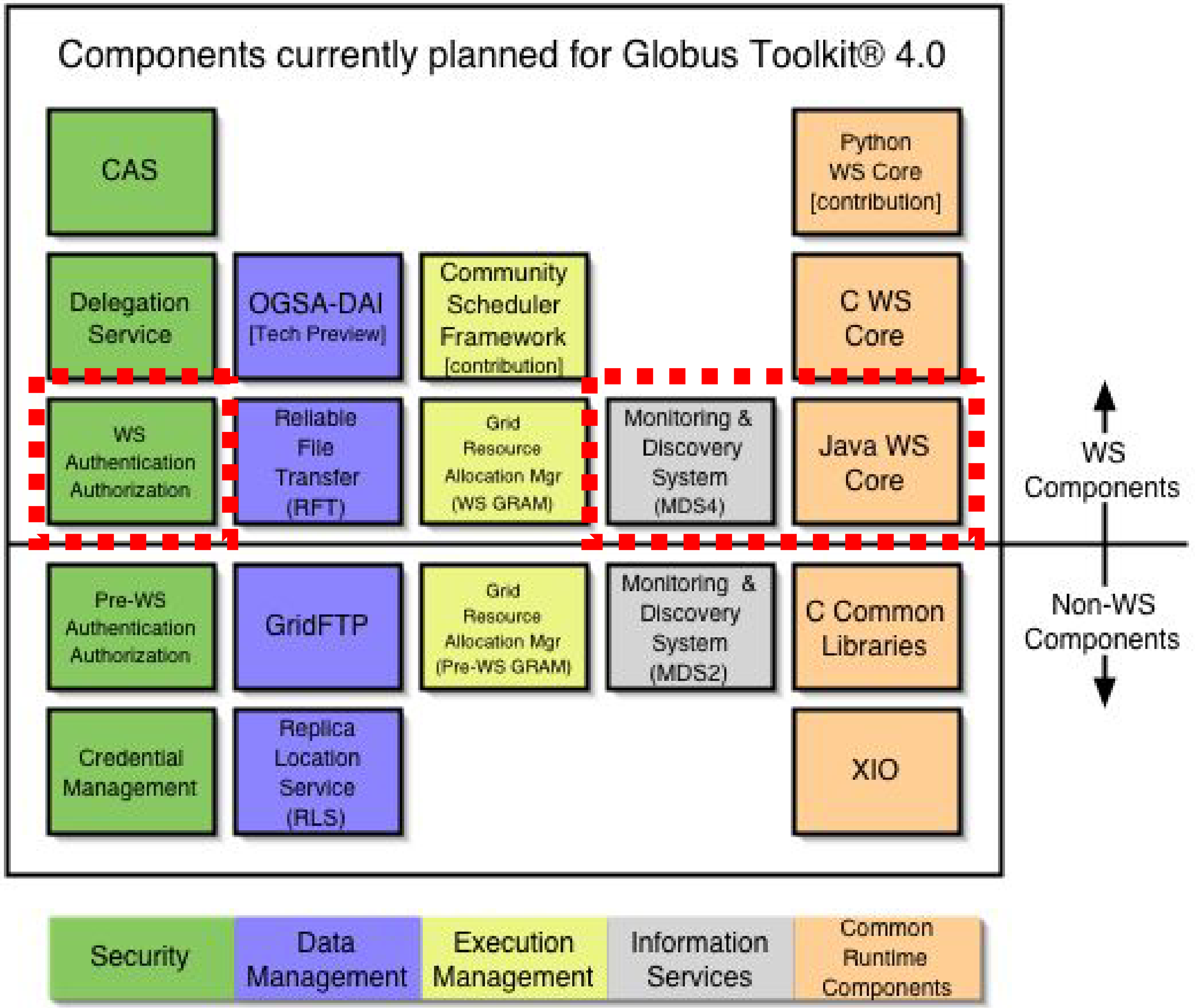
- **Overview of Services and GT4**
- Build a Service
  - 1. Getting Started: Deploy a Service
  - 2. State Management Part I: Create Resources
  - 3. Lifetime Management Part I: Destroy Resources
  - 4. State Management Part II: Add a Resource Property
  - 5. Building a VO: Register with a Community Index
  - 6. Lifetime Management Part II: Lease-based Model
  - 7. Notification: Resource as Notification Producer
  - 8. Discovery: Find a Resource
  - 9. Security Part I: Service-Level
  - 10. Security Part II (optional): Resource-Level
- Overview of Tools for GT4 Service Developers
- Tutorial Summary
- Ideas for Further Work

Components in the Globus Toolkit



Components currently planned for Globus Toolkit® 4.0
CAS
Python WS Core [contribution]
Delegation Service
OGSA-DAI [Tech Preview]
Community Scheduler Framework [contribution]
C WS Core
WS Authentication Authorization
Reliable File Transfer (RFT)
Grid Resource Allocation Mgr (WS GRAM)
Monitoring & Discovery System (MDS4)
Java WS Core
WS Components
Non-WS Components
Pre-WS Authentication Authorization
GridFTP
Grid Resource Allocation Mgr (Pre-WS GRAM)
Monitoring & Discovery System (MDS2)
C Common Libraries
Credential Management
Replica Location Service (RLS)
XIO
Security
Data Management
Execution Management
Information Services
Common Runtime Components

Components used in this Tutorial


Components currently planned for Globus Toolkit® 4.0
CAS
Python WS Core [contribution]
Delegation Service
OGSA-DAI [Tech Preview]
Community Scheduler Framework [contribution]
C WS Core
WS Authentication Authorization
Reliable File Transfer (RFT)
Grid Resource Allocation Mgr (WS GRAM)
Monitoring & Discovery System (MDS4)
Java WS Core
WS Components
Non-WS Components
Pre-WS Authentication Authorization
GridFTP
Grid Resource Allocation Mgr (Pre-WS GRAM)
Monitoring & Discovery System (MDS2)
C Common Libraries
Credential Management
Replica Location Service (RLS)
XIO
Security
Data Management
Execution Management
Information Services
Common Runtime Components

How to Build a Service Using GT4

- Overview of Services and GT4
- **Build a Service**
  - 1. Getting Started: Deploy a Service
  - 2. State Management Part I: Create Resources
  - 3. Lifetime Management Part I: Destroy Resources
  - 4. State Management Part II: Add a Resource Property
  - 5. Building a VO: Register with a Community Index
  - 6. Lifetime Management Part II: Lease-based Model
  - 7. Notification: Resource as Notification Producer
  - 8. Discovery: Find a Resource
  - 9. Security Part I: Service-Level
  - 10. Security Part II (optional): Resource-Level
- Overview of Tools for GT4 Service Developers
- Tutorial Summary
- Ideas for Further Work

Hands-On Tutorial Structure

- The hands-on portion of the tutorial is organized as a series of exercises in which students add increasing functionality to a skeletal service implementation
- The exercises demonstrate fundamental Web service interactions using the Globus Toolkit® 4.0
- Each exercise includes:
  - A discussion of the concepts behind the exercise
  - Implementation details
  - Step-by-step instructions
  - A view of the finished exercise

the globus toolkit®
www.globustoolkit.org

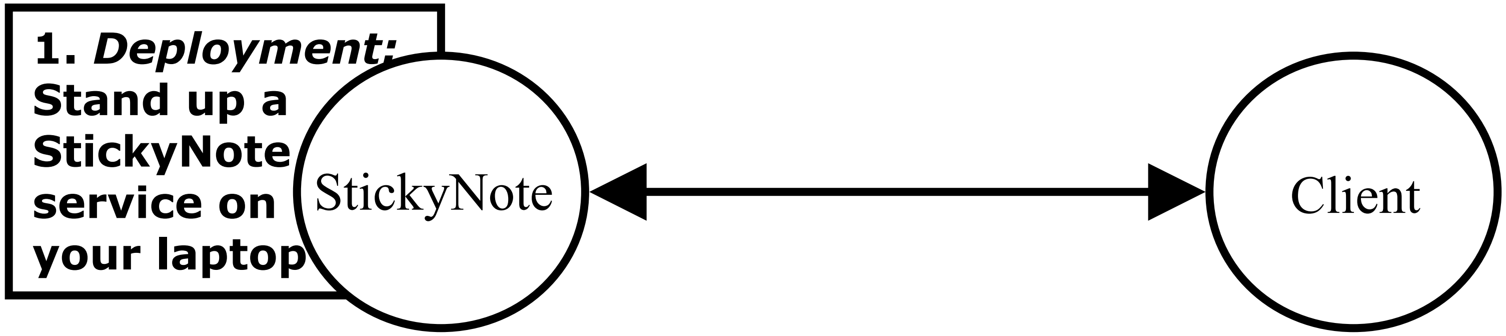
Supporting Tutorial Materials

- Each attendee will use
  - This slideset
  - A code bundle
  - A set of exercise notes
  - X.509 certificates
  - The oracle (Ben)
- The instructors will use
  - An index service containing attendee service entries
  - A visualizer for the index service’s data

How to Build a Service Using GT4

- Overview of Services and GT4
- Build a Service
  - **1. Getting Started: Deploy a Service**
  - 2. State Management Part I: Create Resources
  - 3. Lifetime Management Part I: Destroy Resources
  - 4. State Management Part II: Add a Resource Property
  - 5. Building a VO: Register with a Community Index
  - 6. Lifetime Management Part II: Lease-based Model
  - 7. Notification: Resource as Notification Producer
  - 8. Discovery: Find a Resource
  - 9. Security Part I: Service-Level
  - 10. Security Part II (optional): Resource-Level
- Overview of Tools for GT4 Service Developers
- Tutorial Summary
- Ideas for Further Work

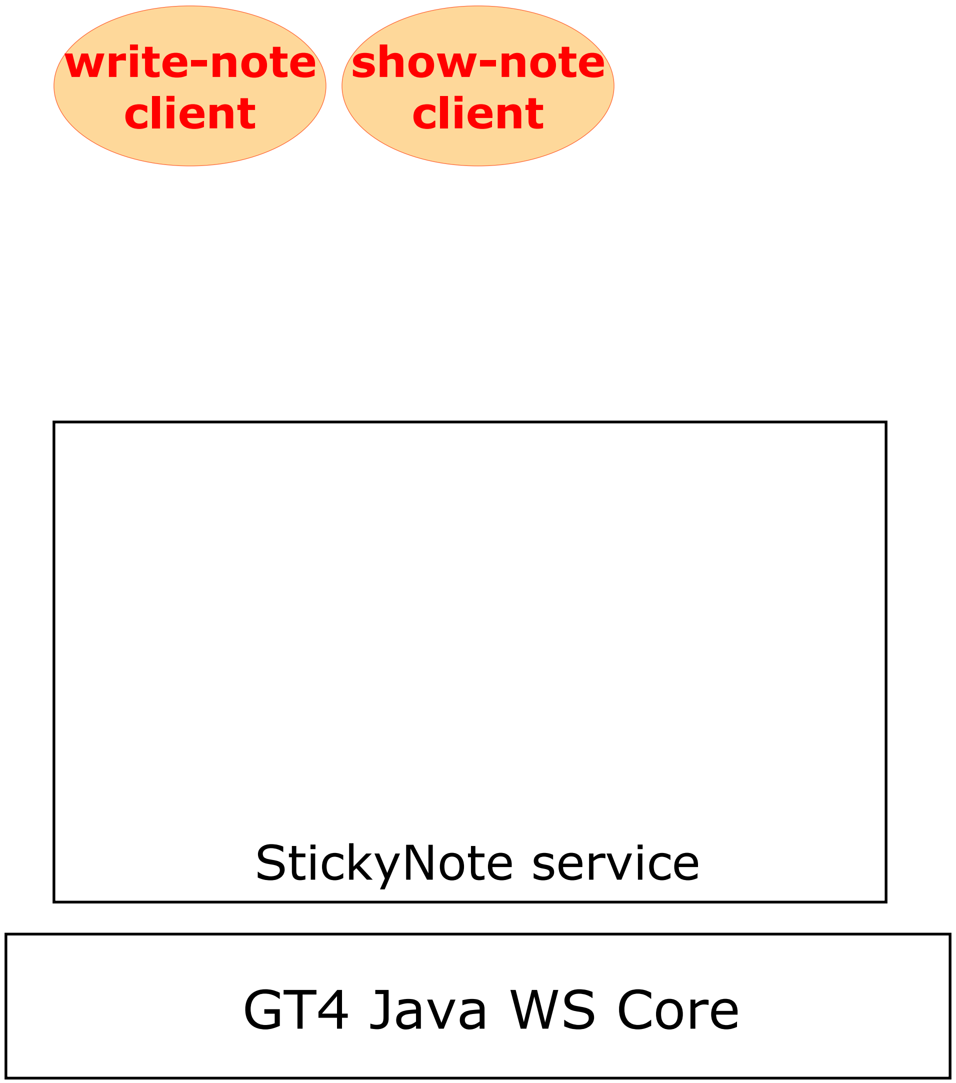
Exercise 1: Deploy a Service


1. *Deployment:* Stand up a StickyNote service on your laptop
StickyNote
Client

The StickyNote Service

- We will begin the tutorial by installing and building demonstration code that writes and shows a text message
- The demonstration code is a web service called StickyNote
- StickyNote executes inside a hosting environment (also called a container)
- During this tutorial we will use GT4's Java hosting environment

The StickyNote Service



Pieces to Install

- GT4 Java WS Core
  - Includes the container
  - globus-start-container – starts container
- Tutorial code
  - Includes the service and client
  - ant clean – similar to "make clean"
  - ant deploy – compiles .java, installs into container

What Attendees Should Do

- Install GT4 and tutorial code
- Start the GT4 container
- Run the show-note client
  - Display the current StickyNote message
- Run the write-note client
  - Write a new message on the StickyNote

What Attendees Should See

bin/globus-start-container

```
Starting SOAP server at:
  http://192.168.123.100:8080/wsrf/services
With the following services:
[1]:http://192.168.123.100:8080/wsrf/servic
  es/InMemoryServiceGroupEntry
[2]:http://192.168.123.100:8080/wsrf/servic
  es/TriggerFactoryService
[… more services listed …]
```

What Attendees Should See

- $GLOBUS_LOCATION/bin/show-note -s <service>
  - hello.
- $GLOBUS_LOCATION/bin/write-note -s <service> cheese
  - Note now reads “cheese”

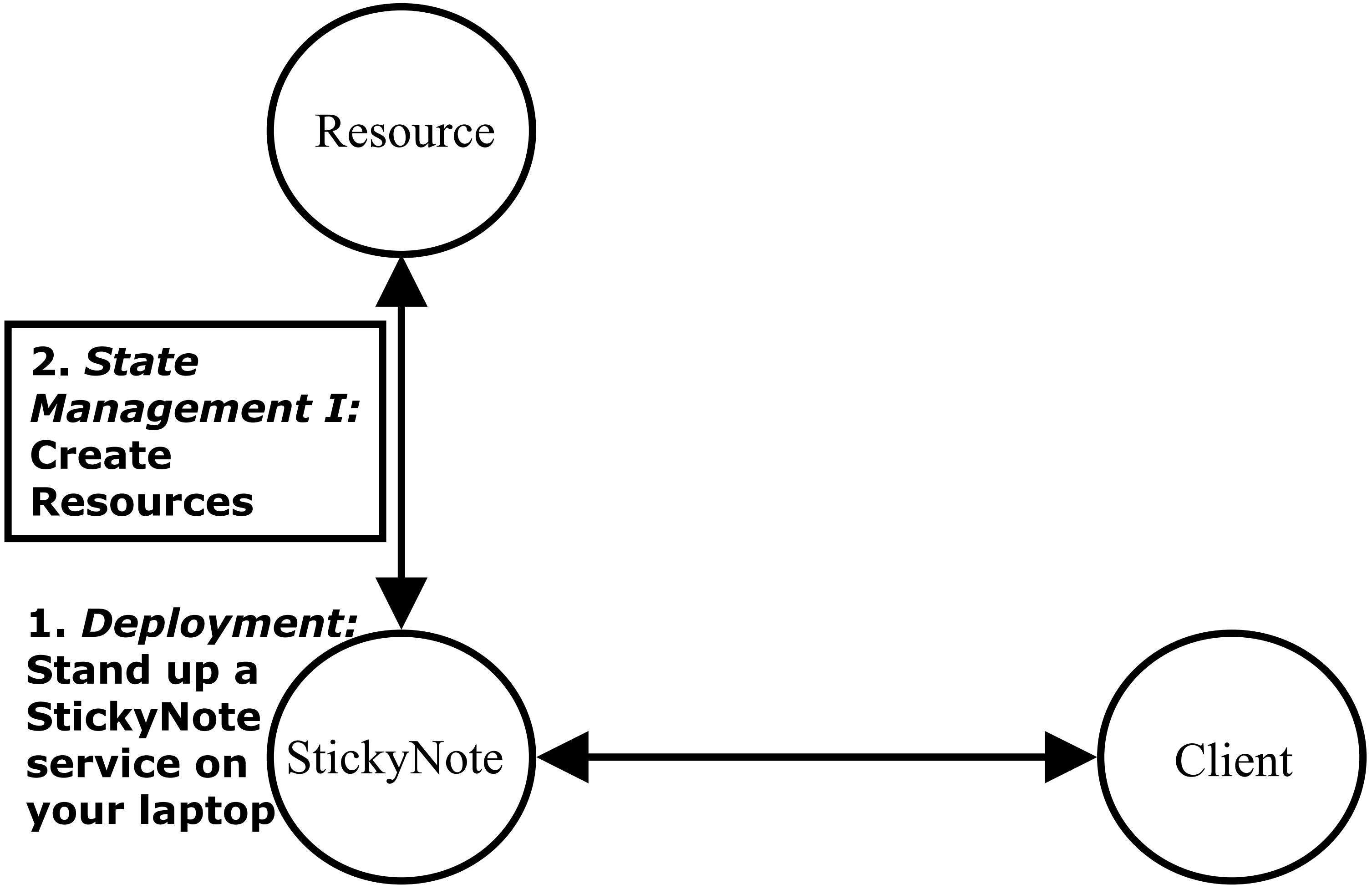
Exercise 1 Review

- Tutorial directory
  - ◆ Run “ant deploy” to install the service
- $GLOBUS_LOCATION
  - ◆ globus-start-container to start the container
  - ◆ show-note to display note contents
  - ◆ write-note to alter note contents

How to Build a Service Using GT4

- Overview of Services and GT4
- Build a Service
  - 1. Getting Started: Deploy a Service
  - **2. State Management Part I: Create Resources**
  - 3. Lifetime Management Part I: Destroy Resources
  - 4. State Management Part II: Add a Resource Property
  - 5. Building a VO: Register with a Community Index
  - 6. Lifetime Management Part II: Lease-based Model
  - 7. Notification: Resource as Notification Producer
  - 8. Discovery: Find a Resource
  - 9. Security Part I: Service-Level
  - 10. Security Part II (optional): Resource-Level
- Overview of Tools for GT4 Service Developers
- Tutorial Summary
- Ideas for Further Work

Exercise 2: State Management I



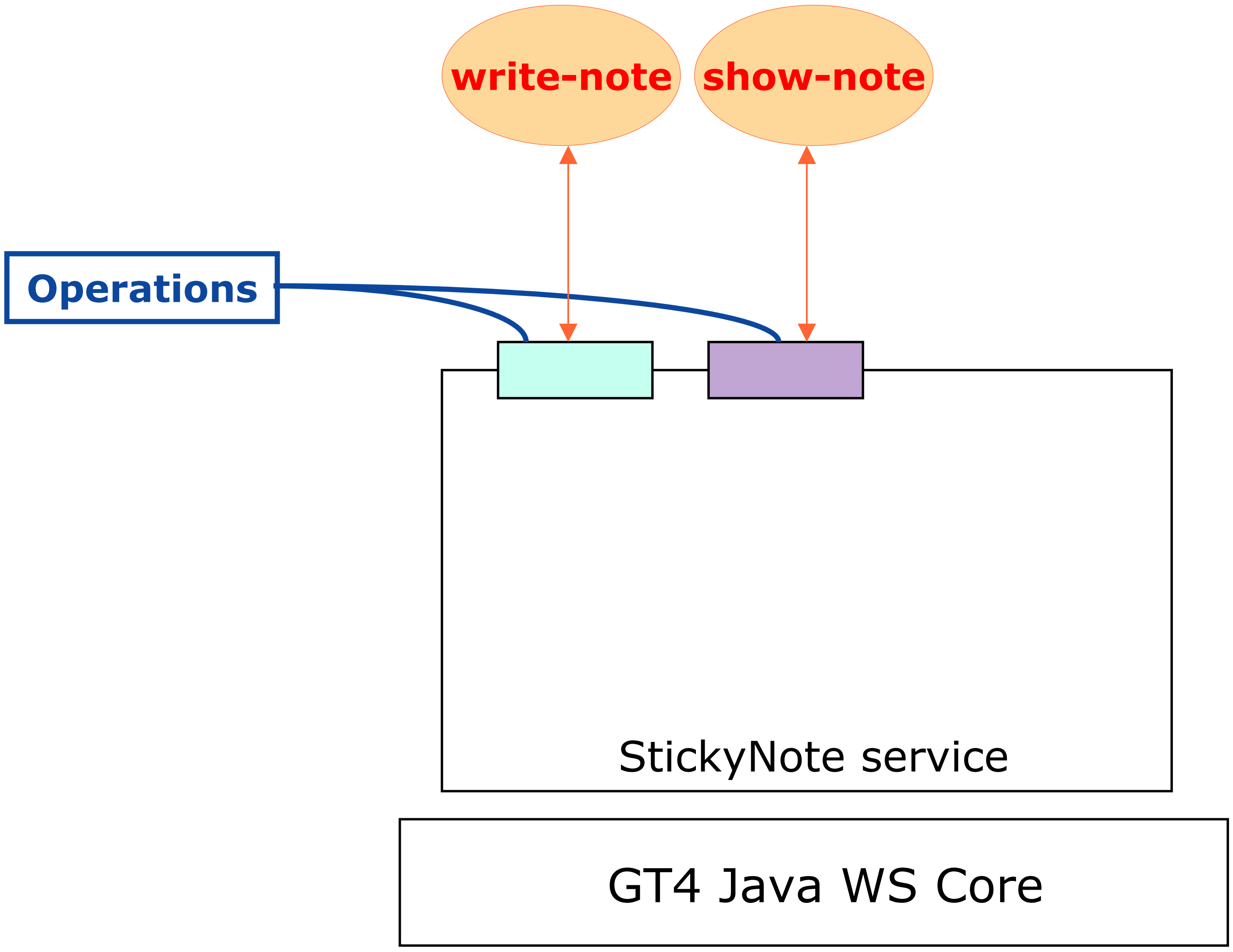
2. *State Management I:* Create Resources

1. *Deployment:* Stand up a StickyNote service on your laptop

Service Basics: Operations

- Services have operations
- Service developers can define as many operations as they wish for their services
- Our StickyNote currently has two operations defined
  - write a note
  - show a note
- In this chapter we will learn how to add a new operation
  - create a note

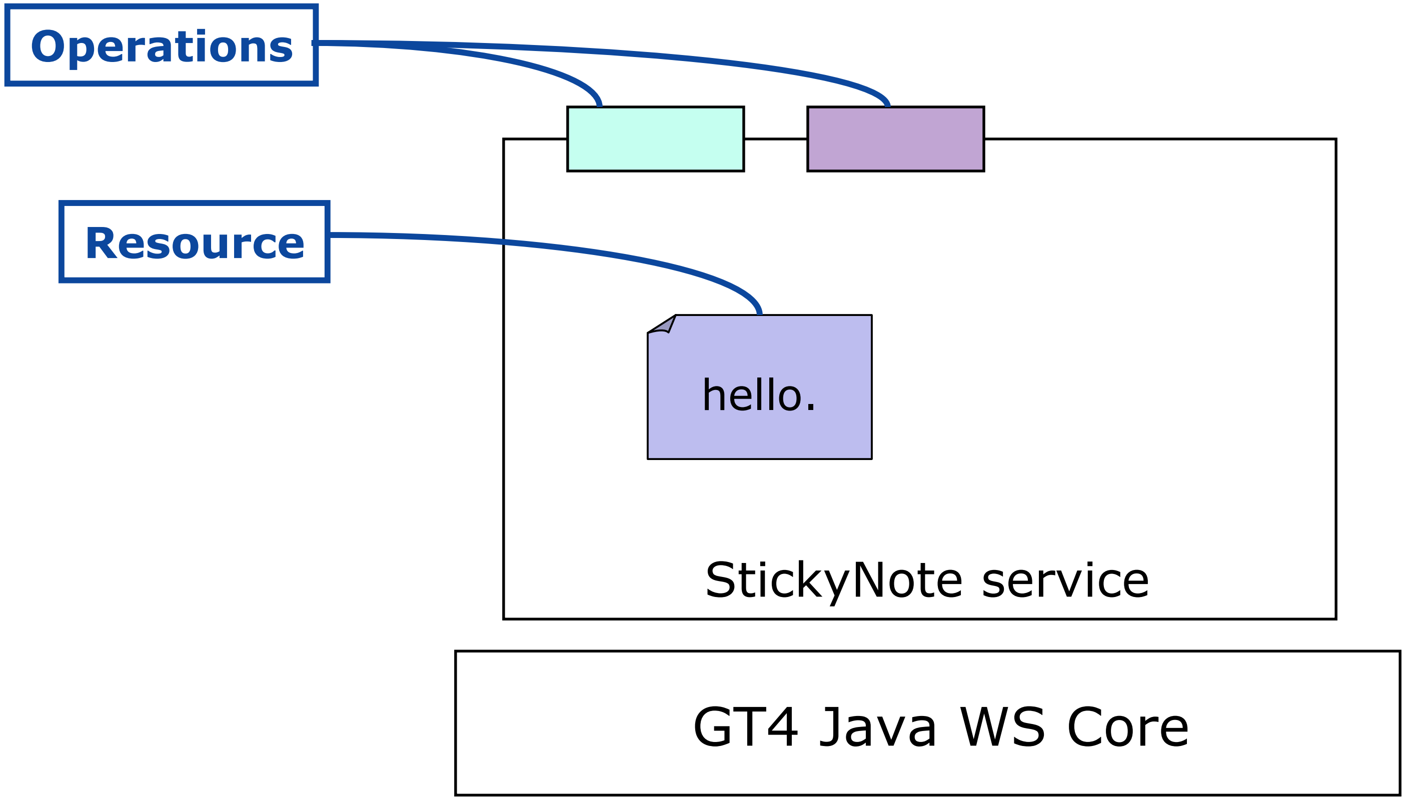
Service Basics: Operations



Service Basics: Resources

- Web services can contain state
  - state is service-related data that persists across client invocations
- The state of the StickyNote is its message
- The state is stored in a Resource
- The StickyNote service currently contains only a single Resource
- In this chapter we will learn how to add support for multiple notes

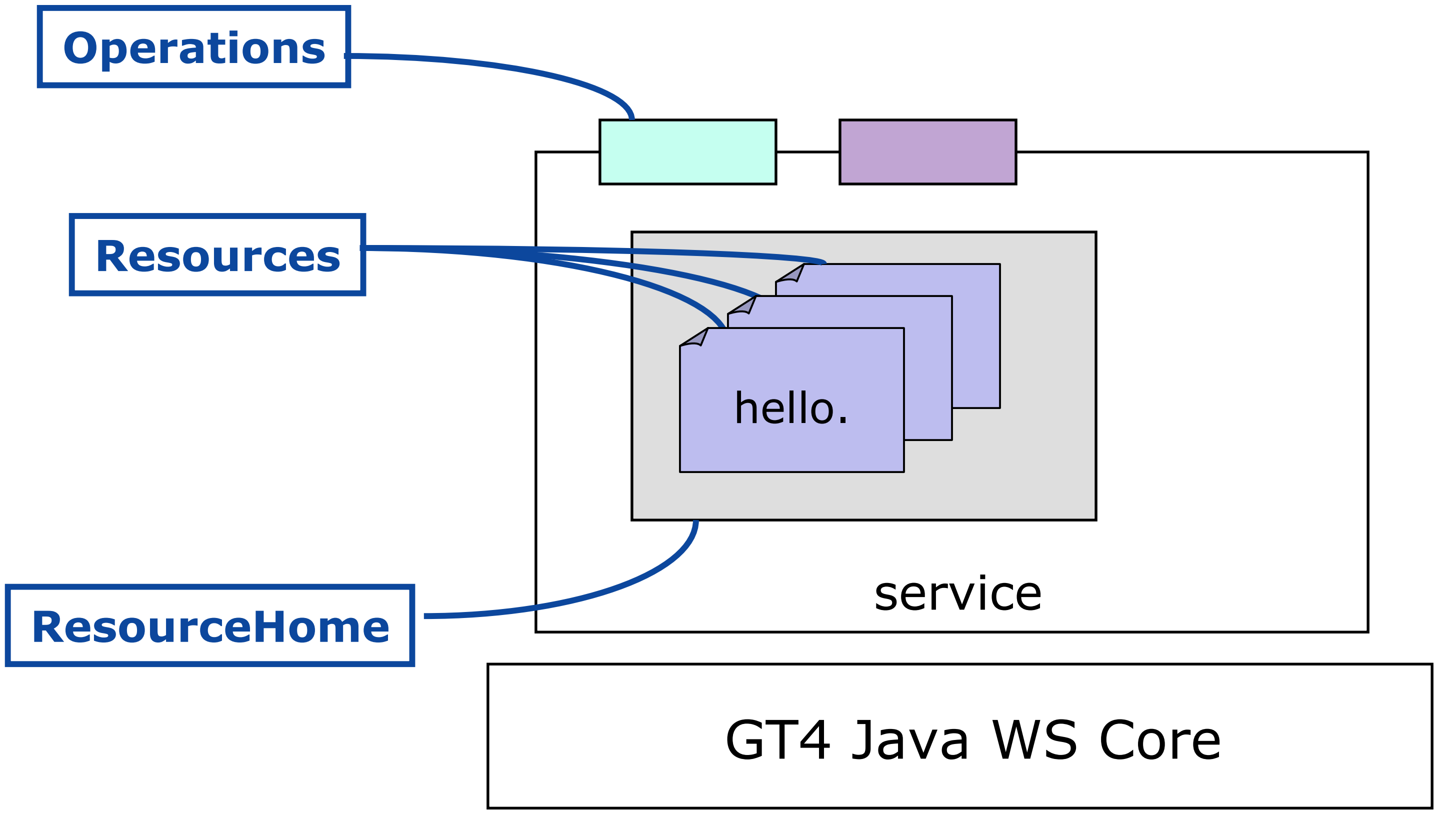
Service Basics: Resources



Service Basics: Resource Homes

- Resources are managed by Resource Homes
  - Resource Homes provide an interface for adding and removing Resources
- In order to add support for multiple notes we must change from using a Resource Home that is limited to one Resource to a Resource Home that can hold multiple Resources

Service Basics: Resource Homes



StickyNote Service Overview

- Interface Description
  - .wsdl files
- Service and Client Implementation
  - .java files
- Build Instructions for Ant
  - build.xml (like a Makefile)
- Deployment Description
  - .wsdd and jndi files

What Attendees Should Do

- Run ant clean first!
- Modify the WSDL to add a create operation
- Implement the create operation in the service
- Write the create-note client
- Edit the JNDI to switch from the SingleNoteHome to the ManyNoteHome

What Attendees Should See

- create-note -s <service>

  new note created...

  EPR written to file: <filename>
- show-note -e <filename>
- write-note -e <filename>

Exercise 2 Review

- The WSDL contains definitions of types, messages, and operations
- The Java contains implementations of the operations defined in WSDL
- The WSDD and JNDI contain deployment information for the container
- The multiple resources contain different state, but are accessed via the same service

How to Build a Service Using GT4

- Overview of Services and GT4
- Build a Service
  - 1. Getting Started: Deploy a Service
  - 2. State Management Part I: Create Resources
  - **3. Lifetime Management Part I: Destroy Resources**
  - 4. State Management Part II: Add a Resource Property
  - 5. Building a VO: Register with a Community Index
  - 6. Lifetime Management Part II: Lease-based Model
  - 7. Notification: Resource as Notification Producer
  - 8. Discovery: Find a Resource
  - 9. Security Part I: Service-Level
  - 10. Security Part II (optional): Resource-Level
- Overview of Tools for GT4 Service Developers
- Tutorial Summary
- Ideas for Further Work

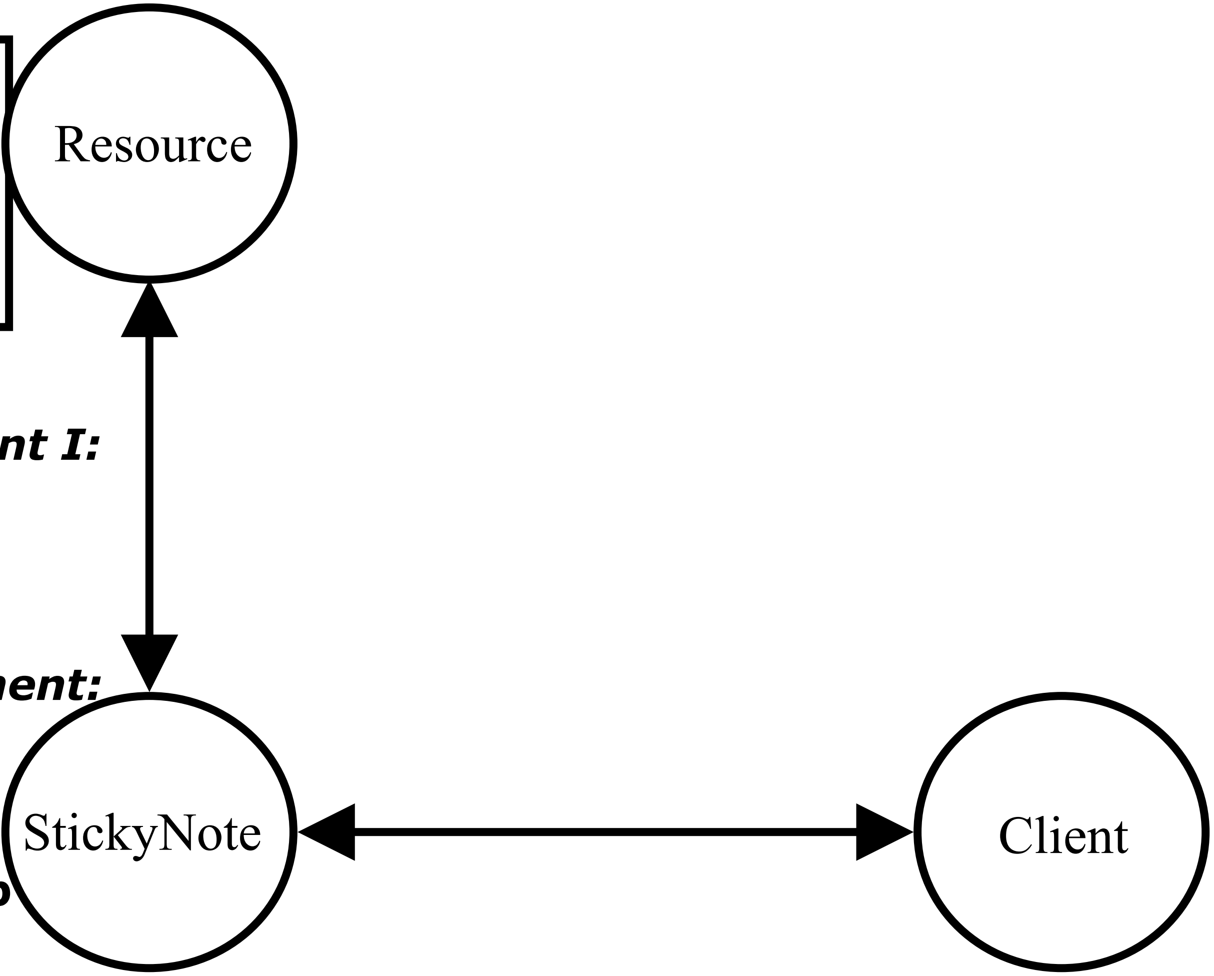
the globus toolkit®
www.globustoolkit.org

Exercise 3: Lifetime Management

3.
Lifetime Mgmt I:
Destroy Resources

2. *State Management I:*
Create Resources

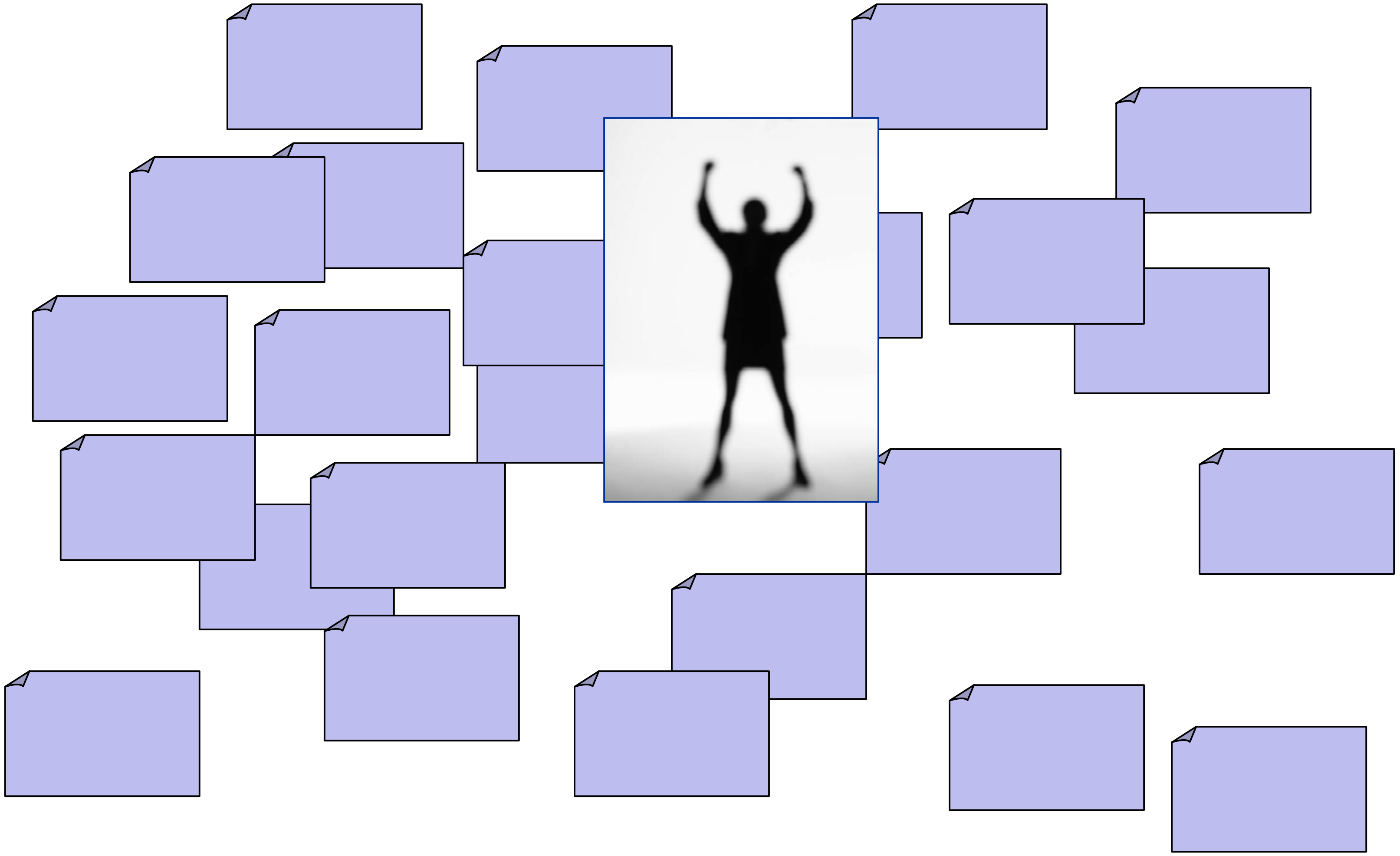
1. *Deployment:*
Stand up a StickyNote service on your laptop



Chapter Overview

- If we keep creating resources, we'll eventually overwhelm our system
- In this chapter we will learn how to destroy resources
- To do so, we must be able to identify an individual resource

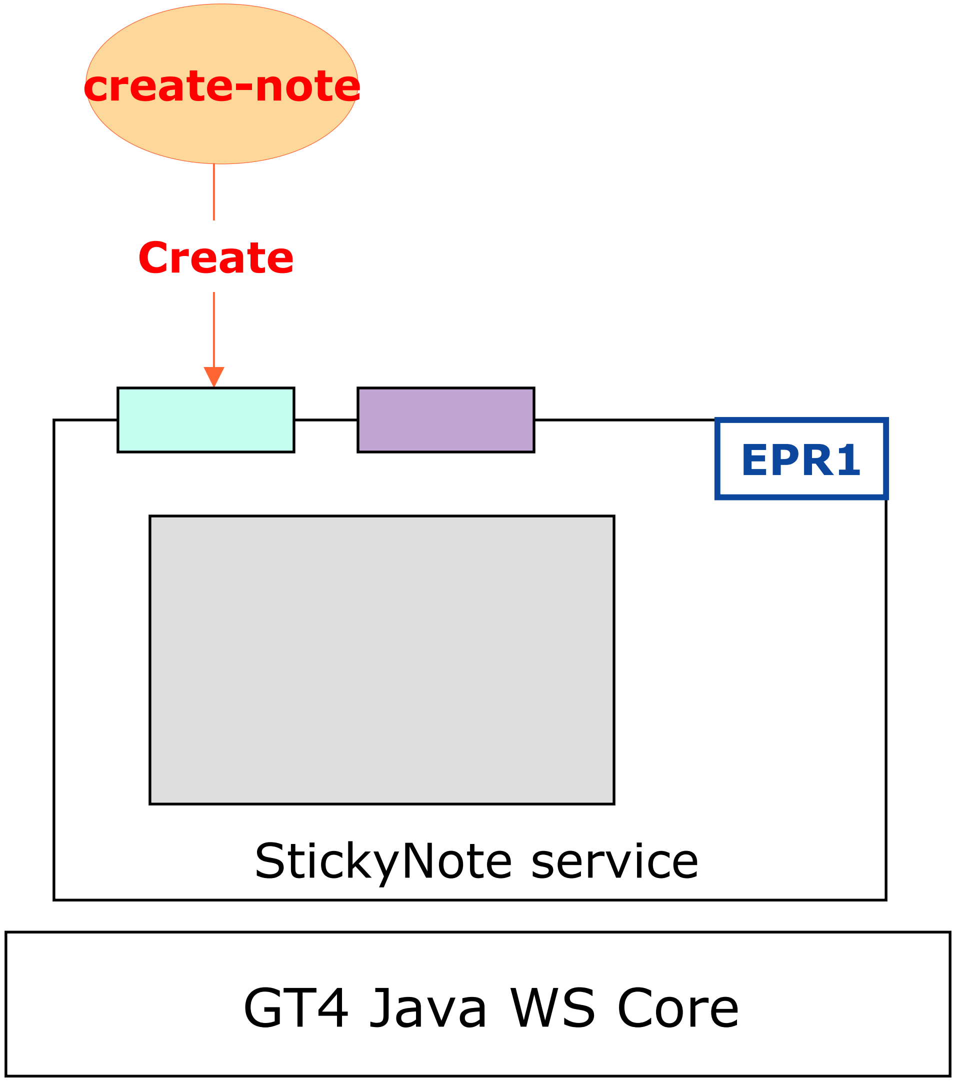
On Lifetime Management



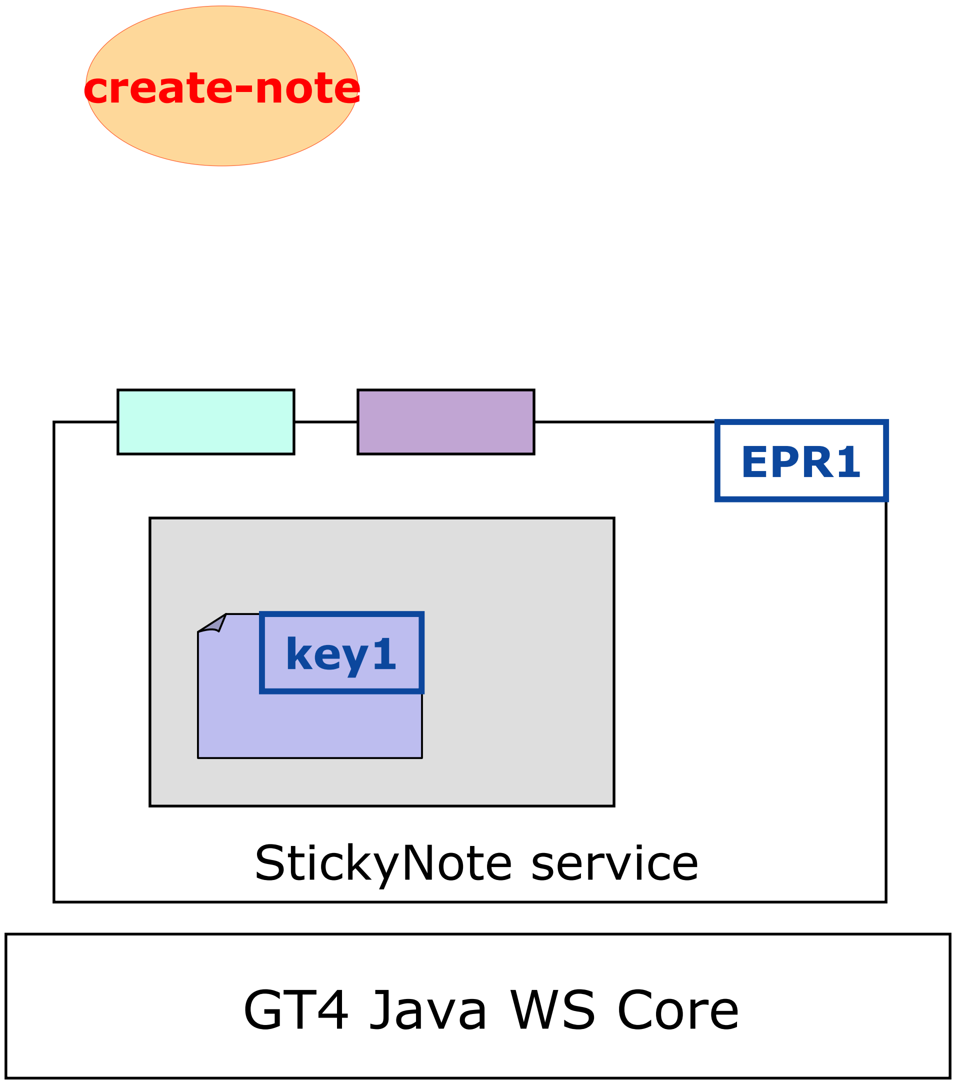
EPRs as Identifiers

- In order to destroy a particular Resource, we must be able to identify it
- In Web services, End Point Identifiers (EPRs) are used to identify a particular endpoint on the network
- The WSRF specs define a way that EPRs can be used to identify a single Resource
- Each Resource in our StickyNote service has an EPR

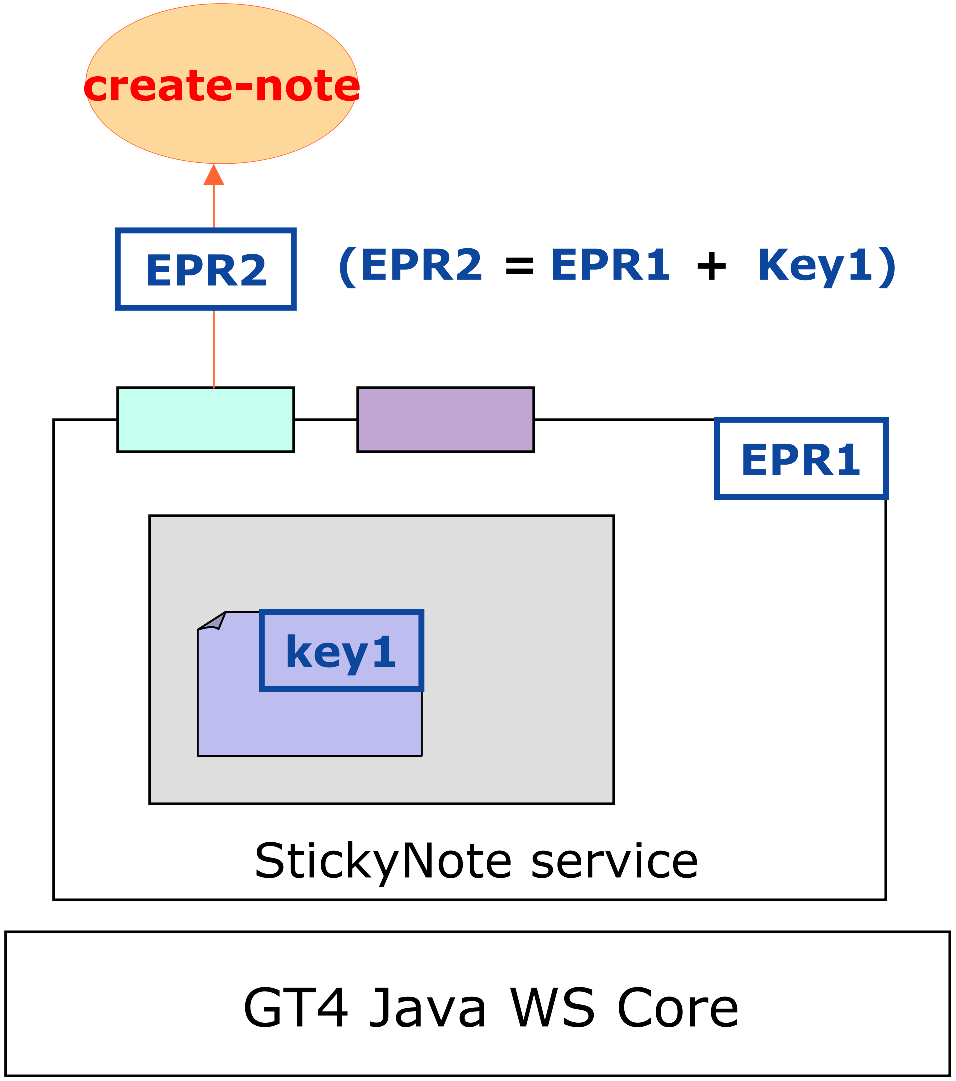
A Runtime Example of EPR Creation



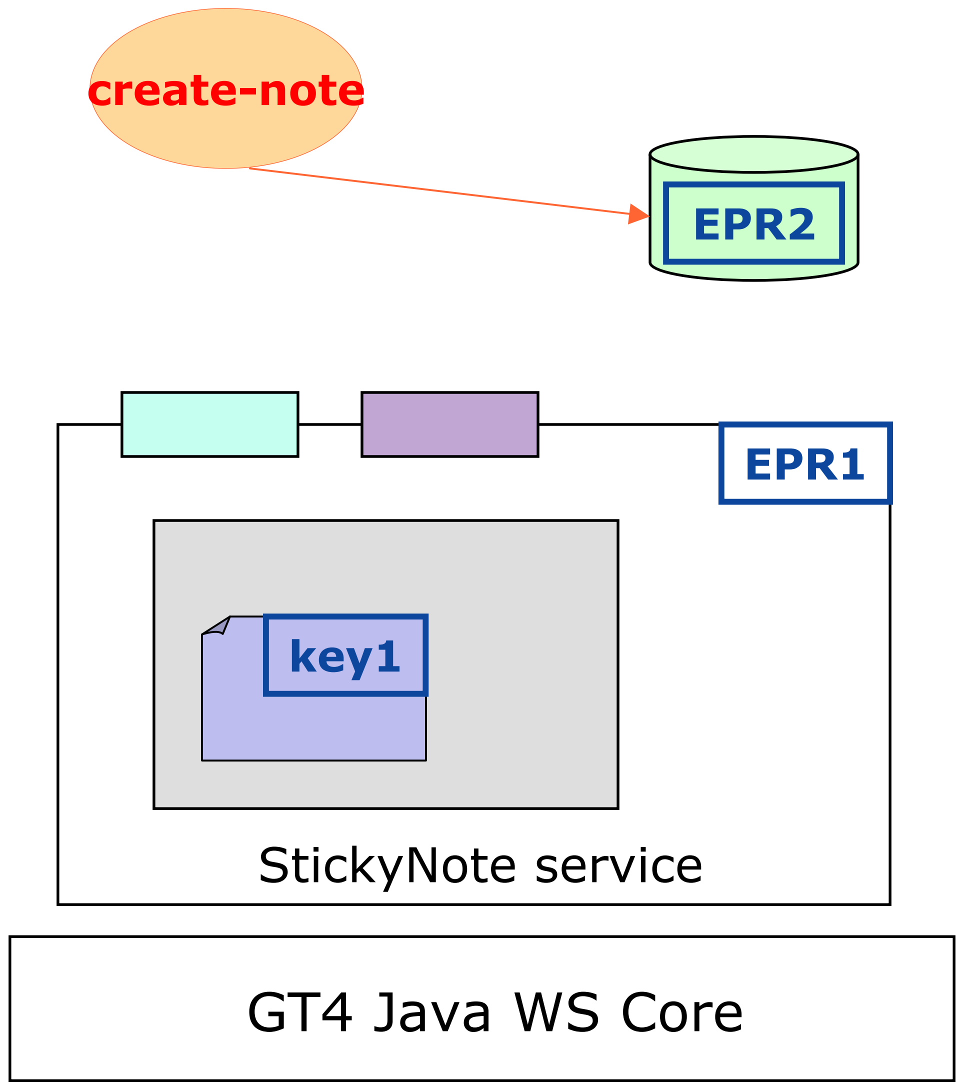
A Runtime Example of EPR Creation



A Runtime Example of EPR Creation



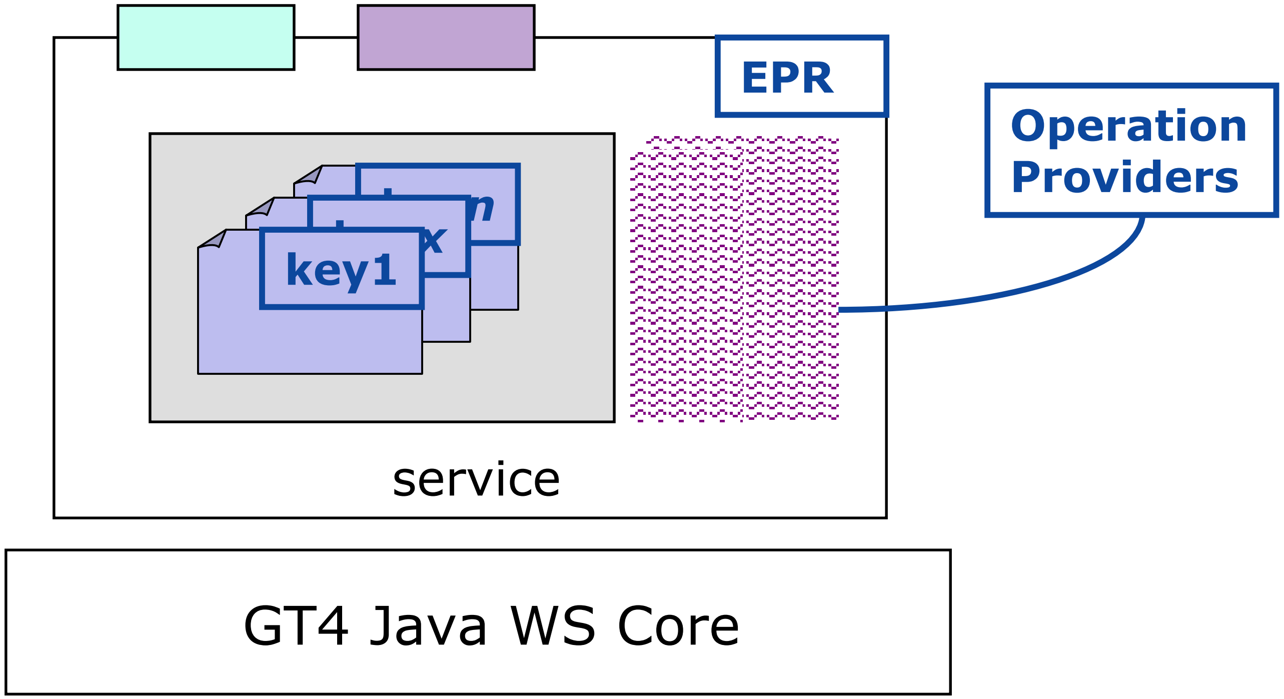
A Runtime Example of EPR Creation



Resource Destruction

- The ability to destroy Resources is built-in to the GT4 hosting environment
- Code to perform destruction is included in the Toolkit as an operation provider
- An operation provider is simply a Java class that contains a discrete set of functionality
- GT4 includes support for adding operation providers to existing services

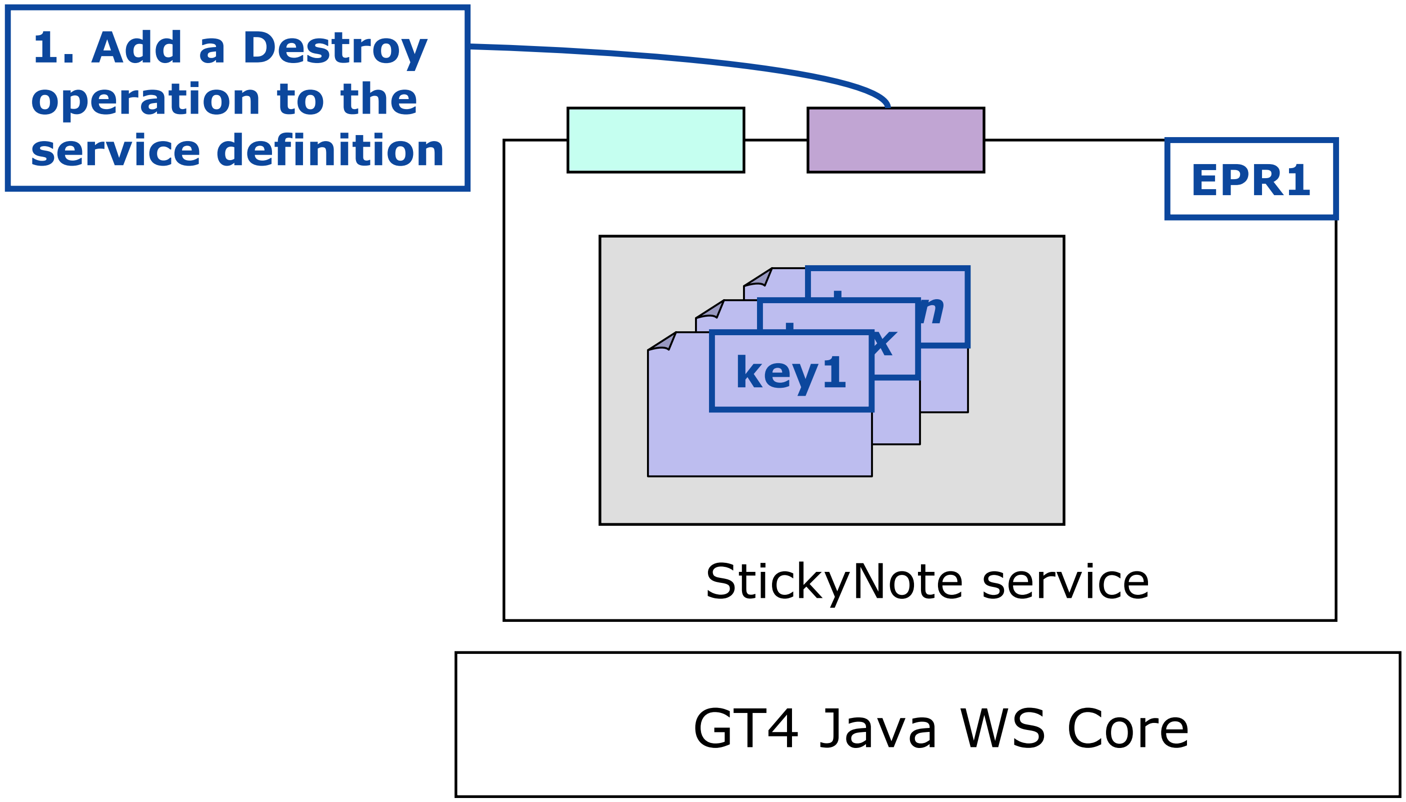
On Operation Providers



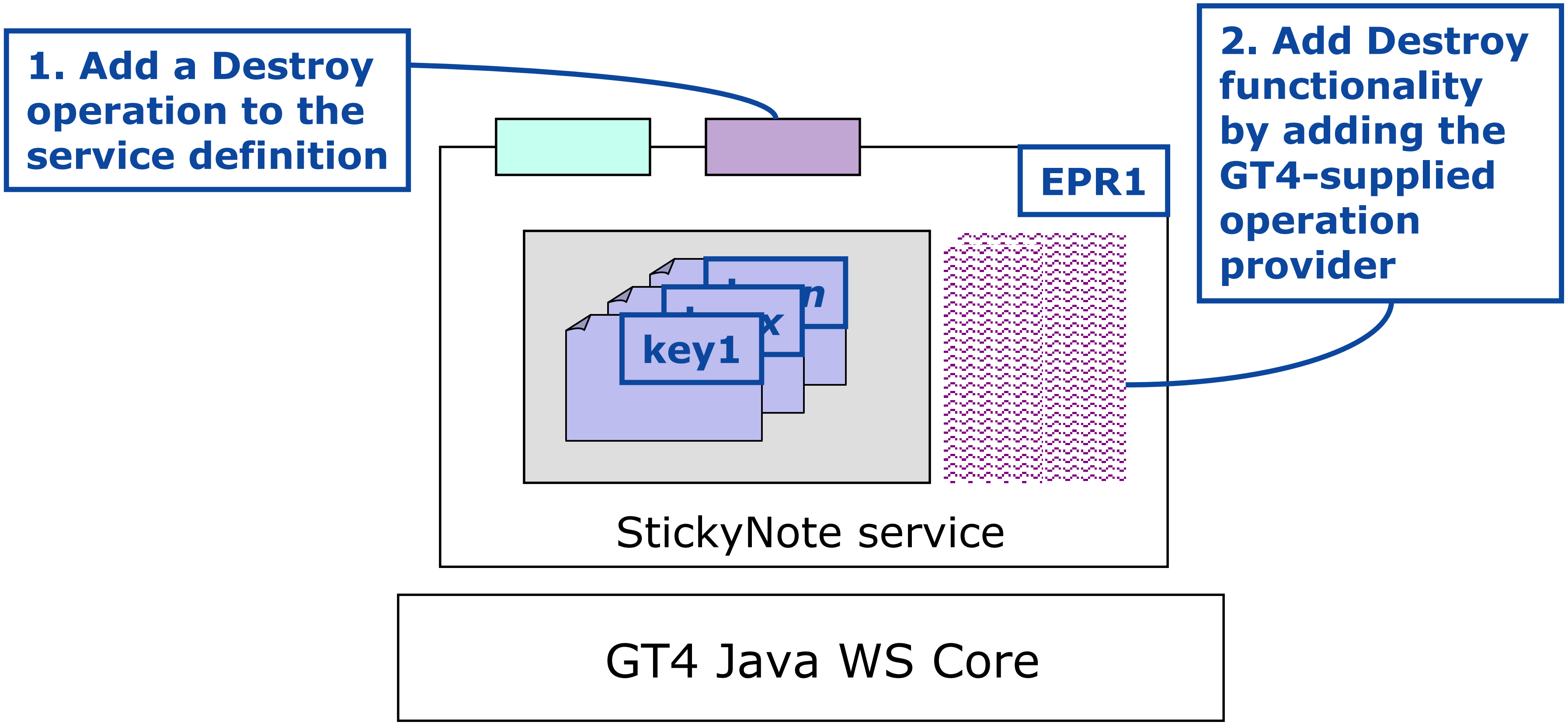
Destroying StickyNote Resources

- We will use the EPR generated by the create operation to identify the resource we want to destroy
- To do this, we need to add a destroy operation defined in WSDL, and provide the implementation of the operation in our service via an operation provider

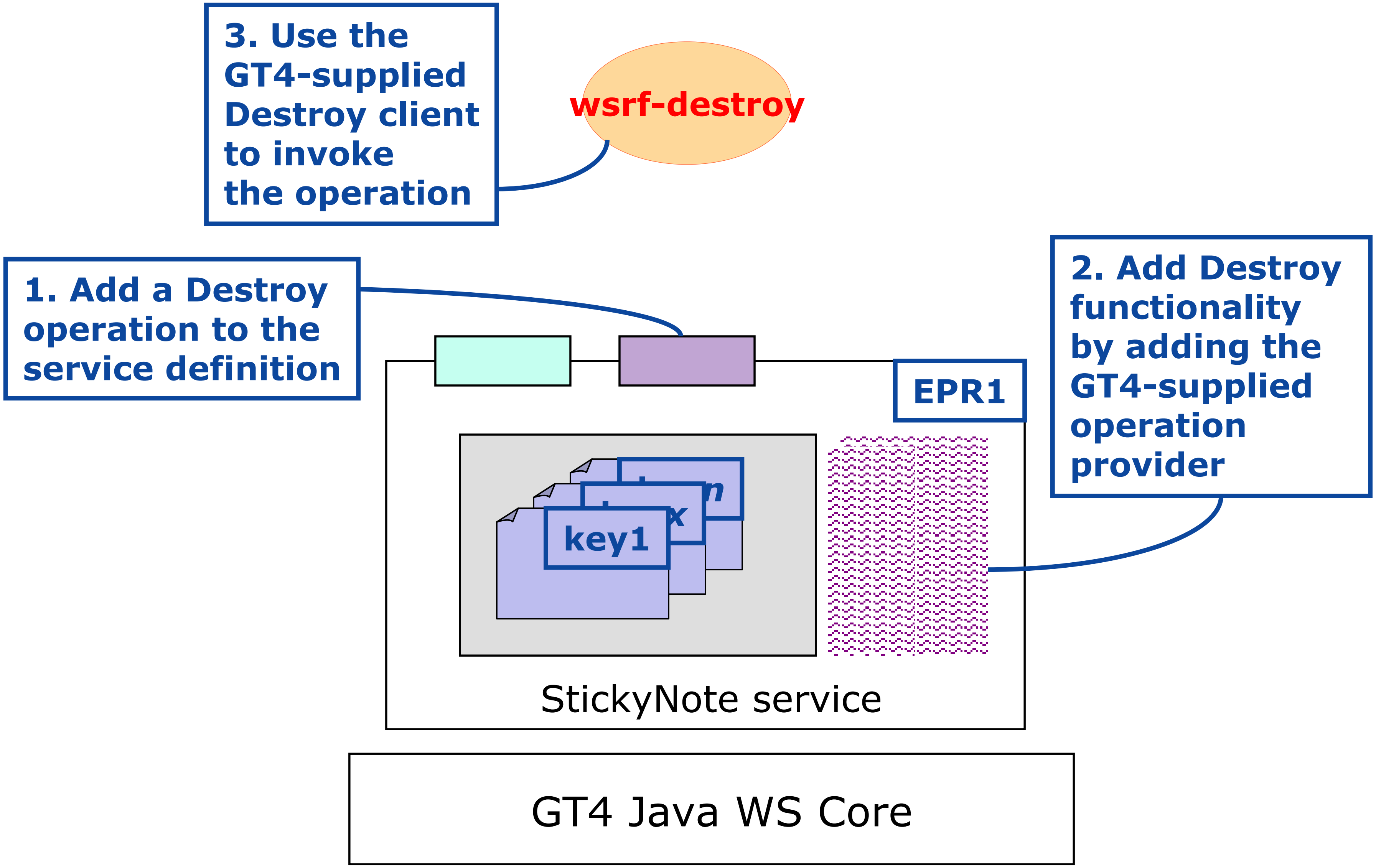
Adding Destroy to StickyNote



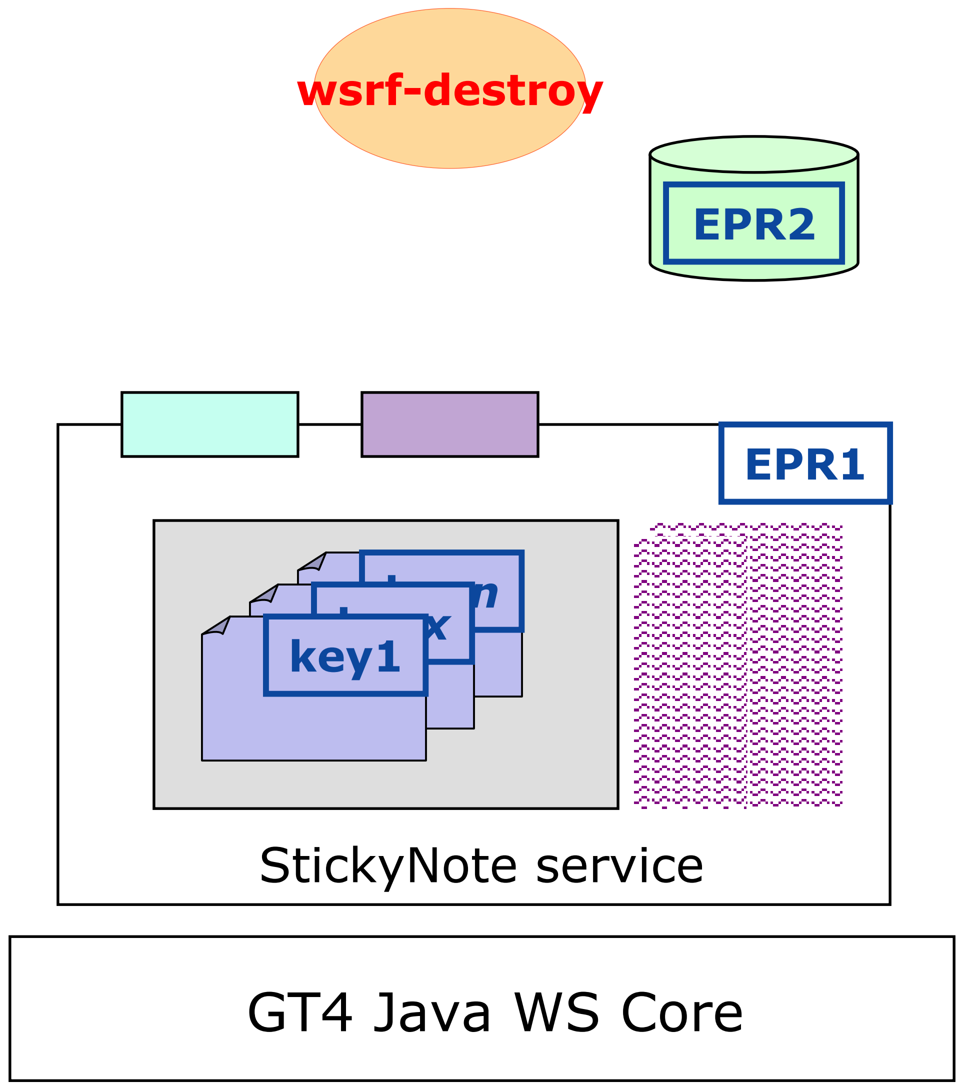
Adding Destroy to StickyNote


1. Add a Destroy operation to the service definition
2. Add Destroy functionality by adding the GT4-supplied operation provider
EPR1
key1
StickyNote service
GT4 Java WS Core

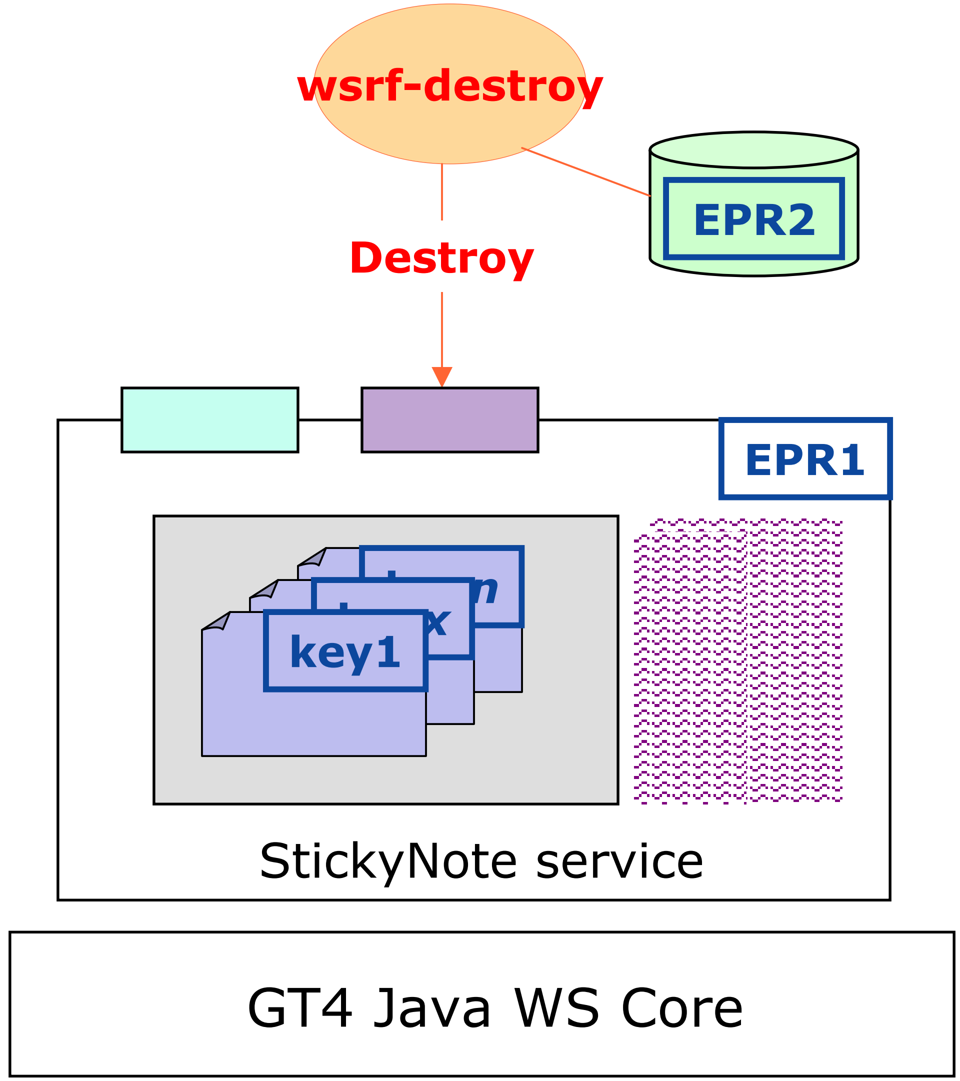
Adding Destroy to StickyNote


3. Use the GT4-supplied Destroy client to invoke the operation
wsrf-destroy
1. Add a Destroy operation to the service definition
2. Add Destroy functionality by adding the GT4-supplied operation provider
EPR1
key1
StickyNote service
GT4 Java WS Core

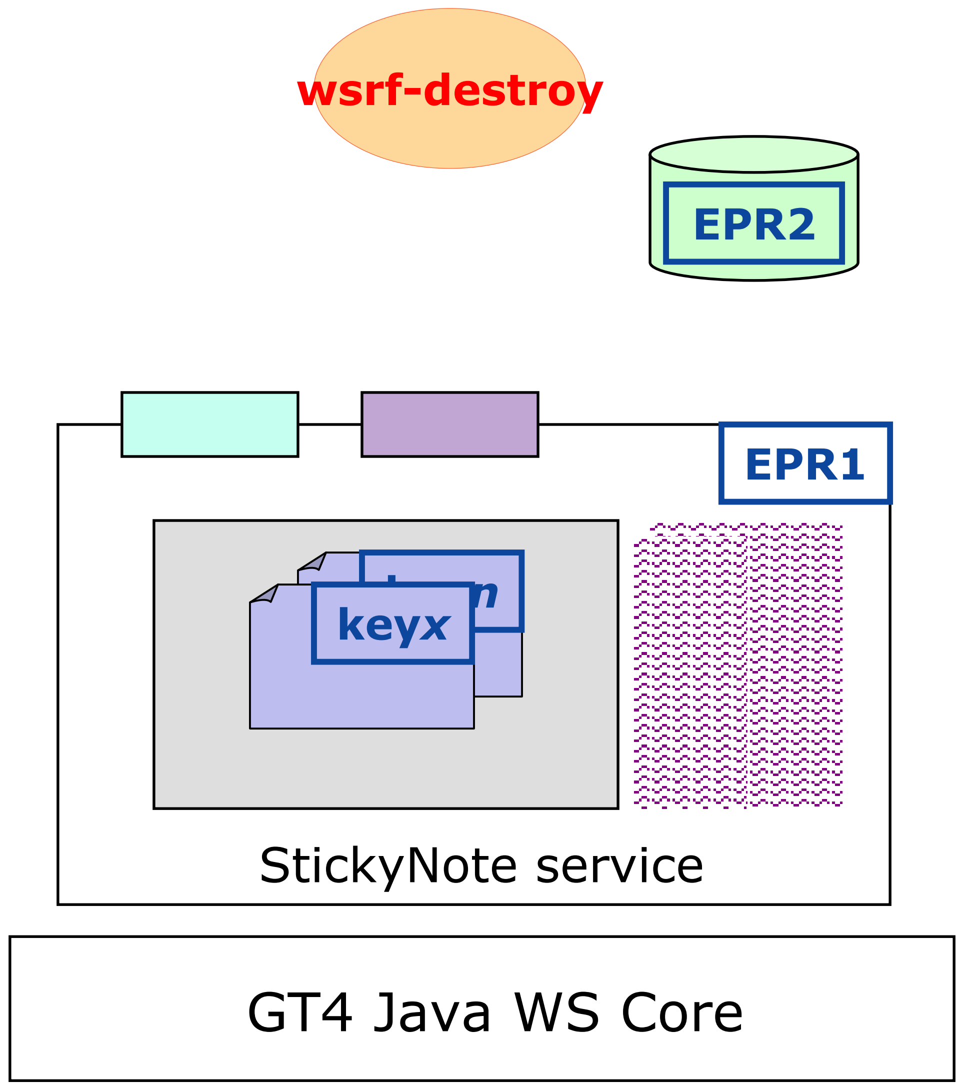
Resource Destruction Runtime



Resource Destruction Runtime



Resource Destruction Runtime



What Attendees Should Do

- Add the destroy operation provider to the WSDD
- Add the destroy interface to the WSDL
- Use the wsrf-destroy command to destroy sticky notes

What Attendees Should See

- wsrf-destroy -e note-1234.epr
  - Resource destroyed

Concepts in Exercise

- EPRs identify targets (endpoints)
  - In our case, identifying a service+resource pair
- EPRs can be used to identify Resources to destroy
- Service developers use operation providers to add service functionality
- GT4 provides several pre-written operation providers and clients for managing Resources, including the destroy capability

Exercise 3 Review

- wsrf-destroy -e <filename>
  - Destroys a resource
- Contents of our EPRs:

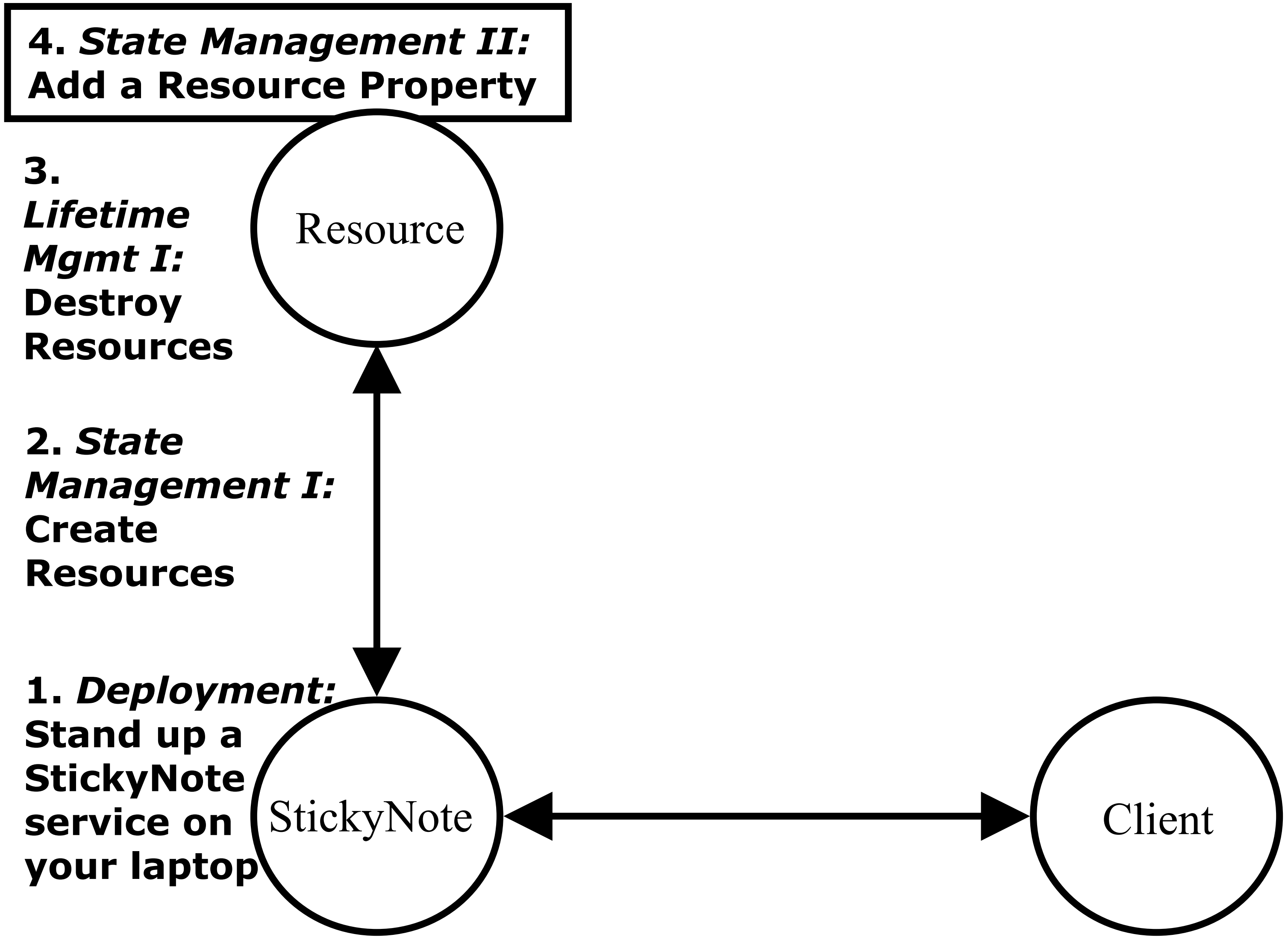
```
<Address>[service]</Address>
<ReferenceProperties><NoteKey>
  [resource key]
</NoteKey></ReferenceProperties>
```

the globus toolkit®
www.globustoolkit.org

How to Build a Service Using GT4

- Overview of Services and GT4
- Build a Service
  - 1. Getting Started: Deploy a Service
  - 2. State Management Part I: Create Resources
  - 3. Lifetime Management Part I: Destroy Resources
  - **4. State Management Part II: Add a Resource Property**
  - 5. Building a VO: Register with a Community Index
  - 6. Lifetime Management Part II: Lease-based Model
  - 7. Notification: Resource as Notification Producer
  - 8. Discovery: Find a Resource
  - 9. Security Part I: Service-Level
  - 10. Security Part II (optional): Resource-Level
- Overview of Tools for GT4 Service Developers
- Tutorial Summary
- Ideas for Further Work

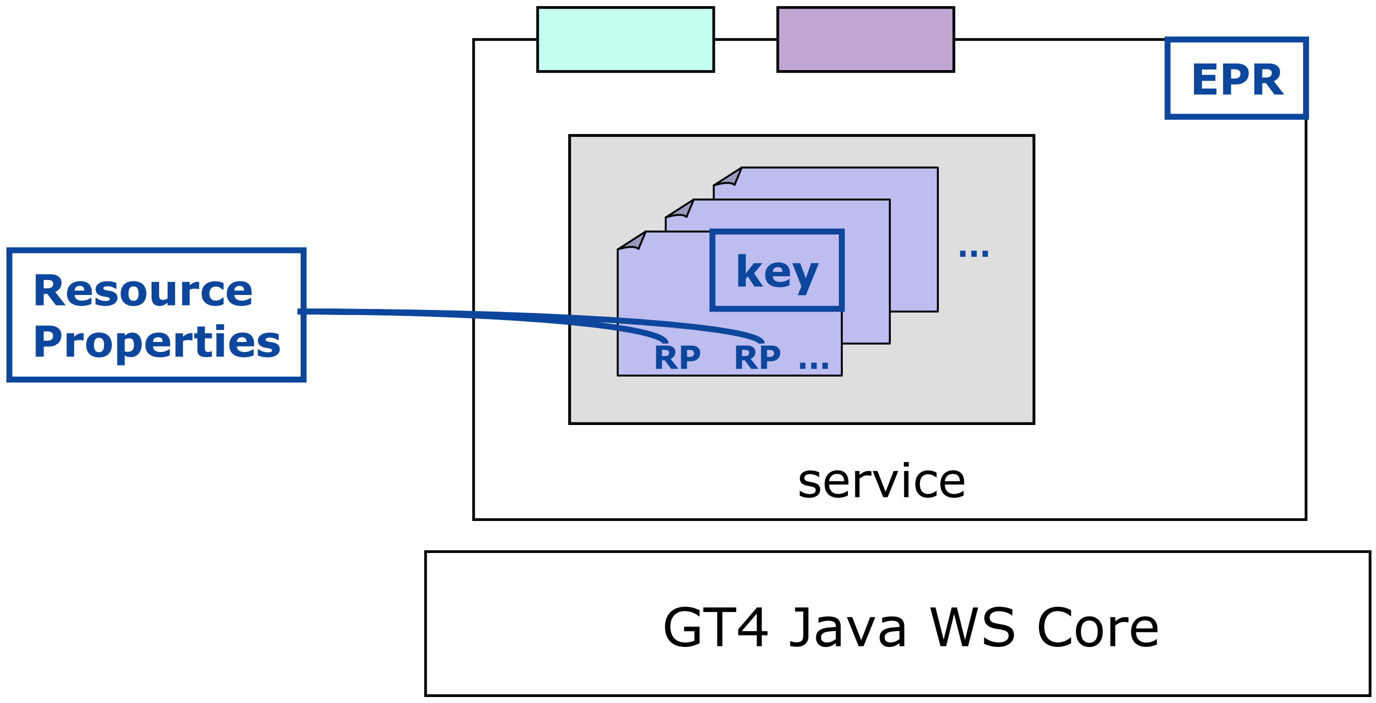
Exercise 4: State Management II


4. ***State Management II:***
Add a Resource Property
3. ***Lifetime Mgmt I:*** **Destroy Resources**
2. ***State Management I:*** **Create Resources**
1. ***Deployment:*** **Stand up a StickyNote service on your laptop**
Resource
StickyNote
Client

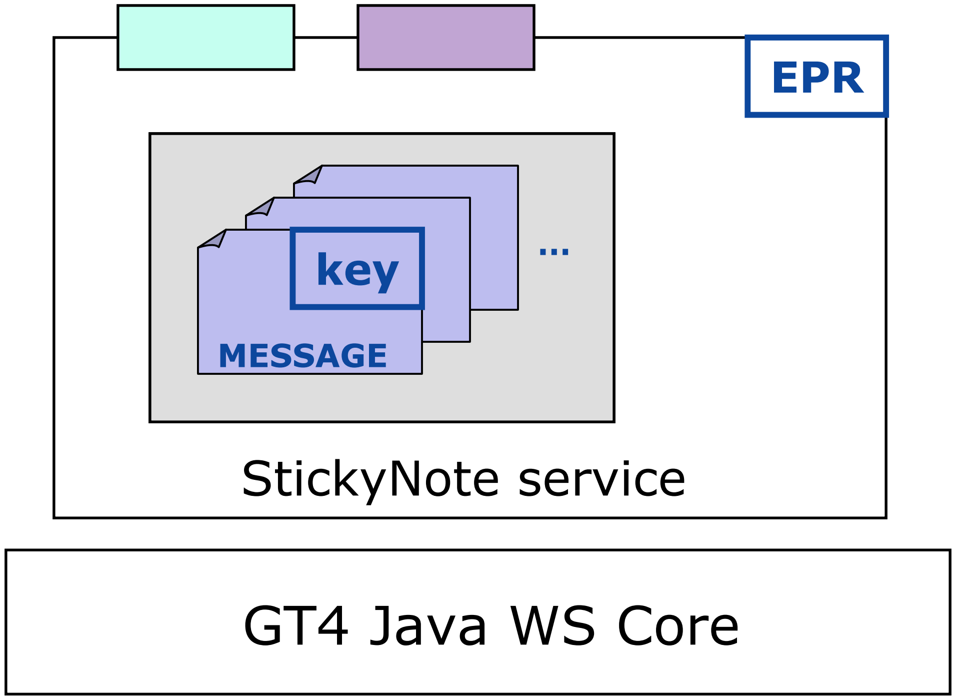
State Management Part II: Resource Properties

- A Resource Property (RP) is an XML element contained within a Resource
- RPs provide a view on the current state of the Resource
  - For us, note contents
  - For data transfer, perhaps # of files successfully transferred, bandwidth available, etc.
- Any Resource can expose internal state as Resource Properties

State Management Part II: Resource Properties



StickyNote Resource Properties



Adding a Resource Property

- Resource Properties are part of the exposed interface of the service/resource pair, so they appear in WSDL
- To add a new RP, we must define it in WSDL, then write code to maintain the value in our implementation
- We already have “message” as an RP
- In this exercise, we’ll add a “last modified time” resource property to the note

RPs in WSDL

```
<element name="message" type="xsd:string"/>
<element name="StickyNoteResourceProperties">
  <complexType><sequence>
  <element ref="tns:message"/>
 </sequence></complexType></element>
<portType name="StickyNotePortType"
  wsrp:ResourceProperties=
     "StickyNoteResourceProperties">
  …
</portType>
```

What Attendees Should Do

- Add new Resource Property to WSDL
- Implement new Resource Property
- Update client to print all Resource Properties

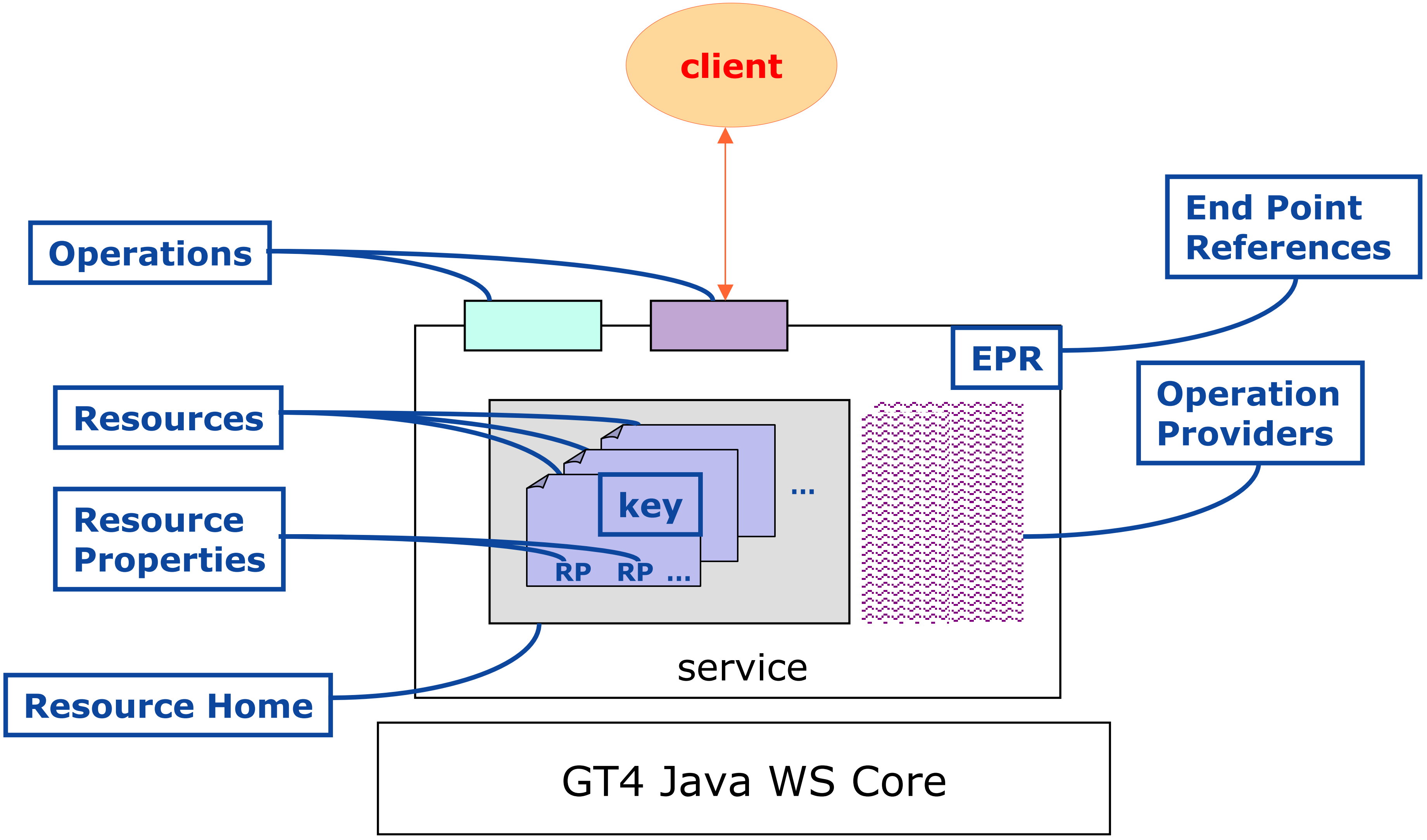
What Attendees Should See

- show-note
  - Will include lastModified information
  - lastModified time updates every time write-note is executed

Exercise 4 Review

- Resource Properties are defined in WSDL
- Resource Properties are XML
- Resources can have as many RPs as they wish
- RPs provide a view on the current state of the Resource
- Any Resource can expose internal state as Resource Properties

What We've Covered So Far

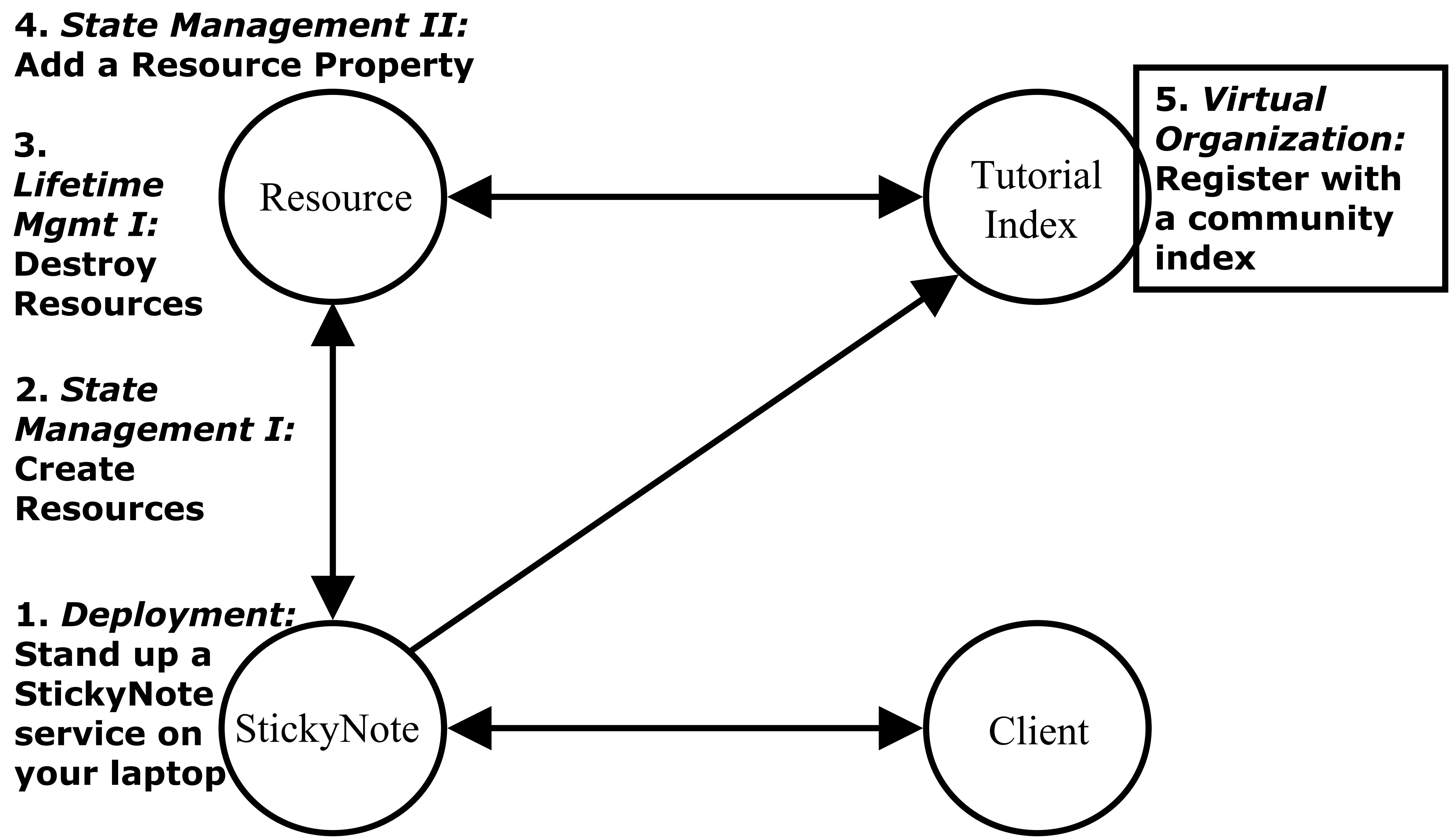


the globus toolkit®
www.globustoolkit.org

How to Build a Service Using GT4

- Overview of Services and GT4
- Build a Service
  - 1. Getting Started: Deploy a Service
  - 2. State Management Part I: Create Resources
  - 3. Lifetime Management Part I: Destroy Resources
  - 4. State Management Part II: Add a Resource Property
  - **5. Building a VO: Register with a Community Index**
  - 6. Lifetime Management Part II: Lease-based Model
  - 7. Notification: Resource as Notification Producer
  - 8. Discovery: Find a Resource
  - 9. Security Part I: Service-Level
  - 10. Security Part II (optional): Resource-Level
- Overview of Tools for GT4 Service Developers
- Tutorial Summary
- Ideas for Further Work

Exercise 5: Virtual Organizations

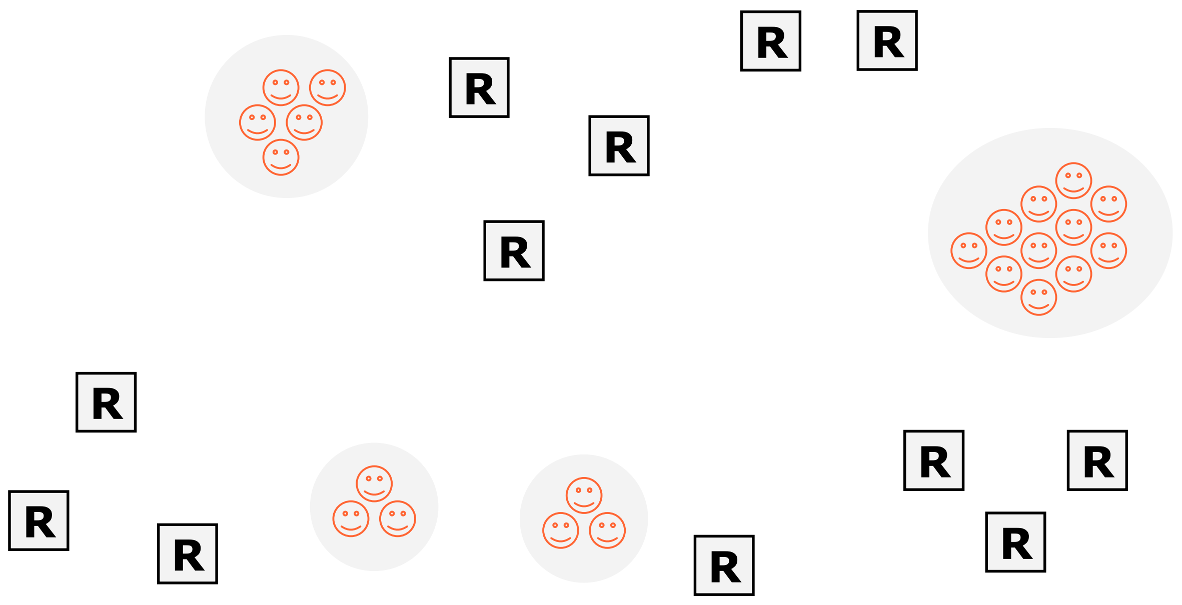

4. *State Management II:* Add a Resource Property
3. *Lifetime Mgmt I:* Destroy Resources
2. *State Management I:* Create Resources
1. *Deployment:* Stand up a StickyNote service on your laptop
5. *Virtual Organization:* Register with a community index
Resource
Tutorial Index
StickyNote
Client

Virtual Organizations

- There is a structure that underlies many Grid problem spaces
- The structure is referred to as a Virtual Organization
- Virtual Organizations drive many of the requirements for Grid technology

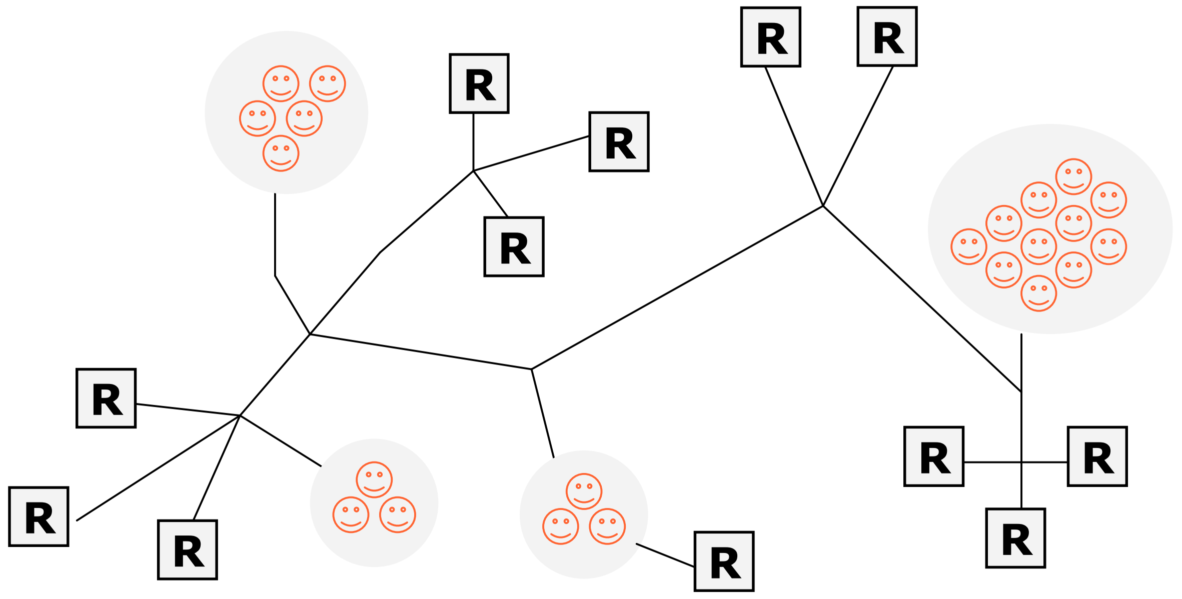
Virtual Organizations

- Distributed resources and people



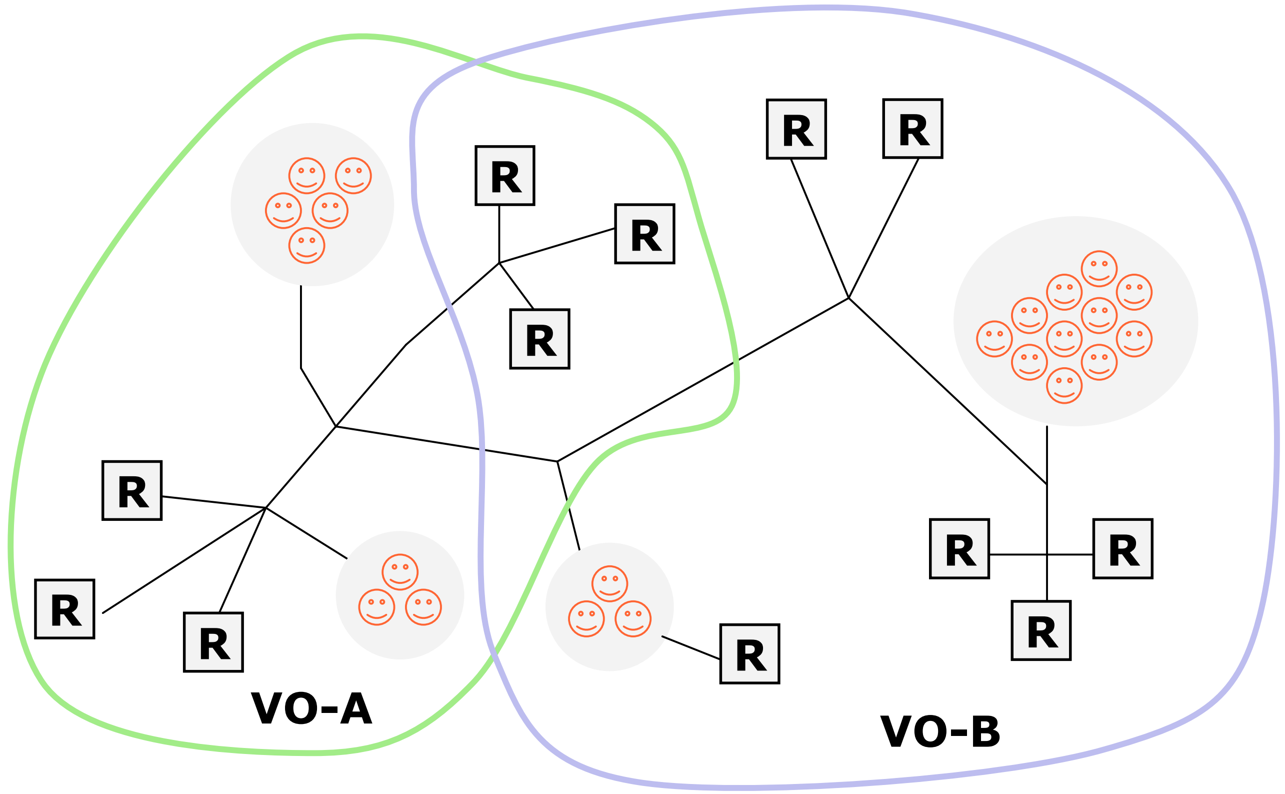
Virtual Organizations

- Distributed resources and people
- Linked by networks, crossing administrative domains



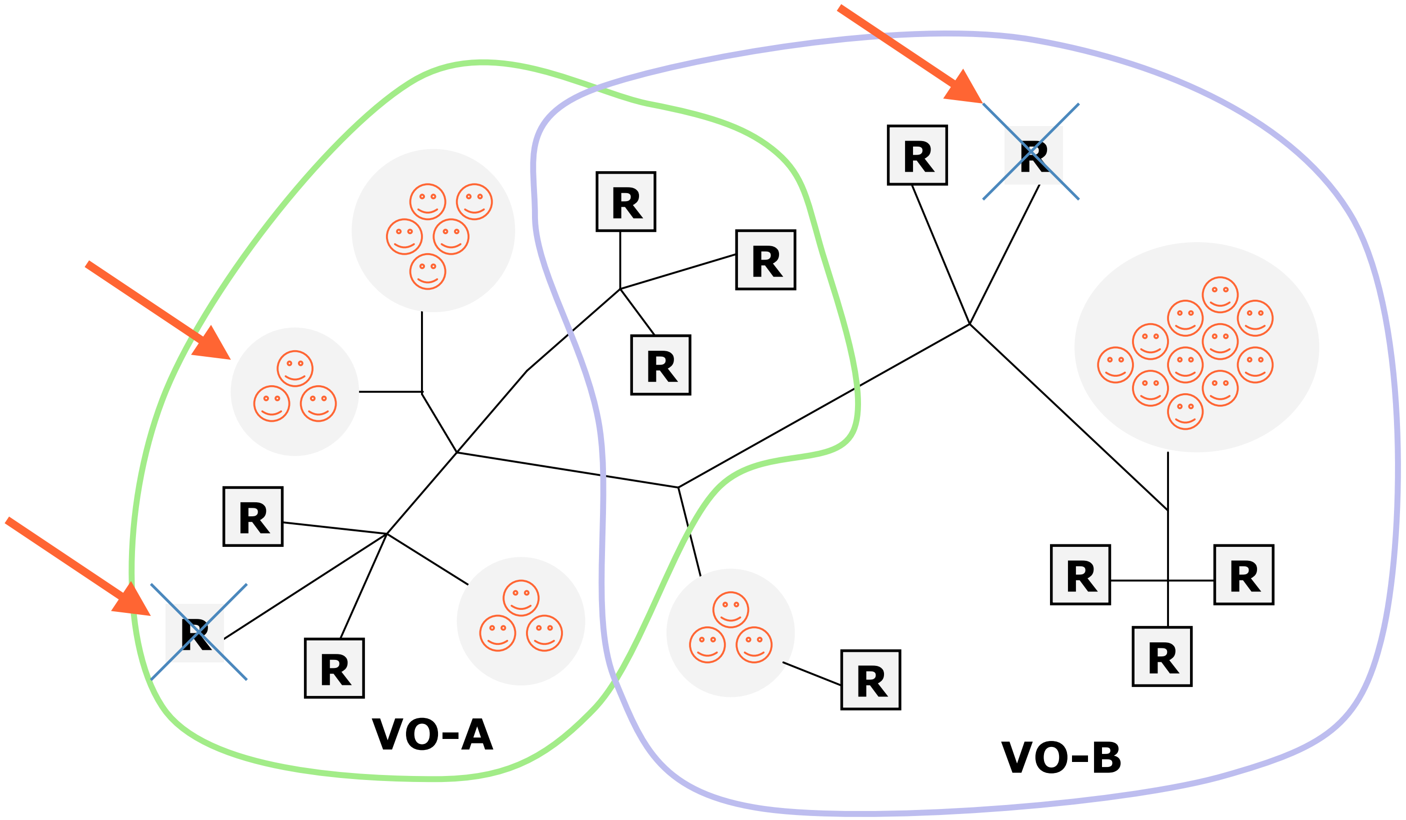
Virtual Organizations

- Distributed resources and people
- Linked by networks, crossing administrative domains
- Sharing resources, common goals



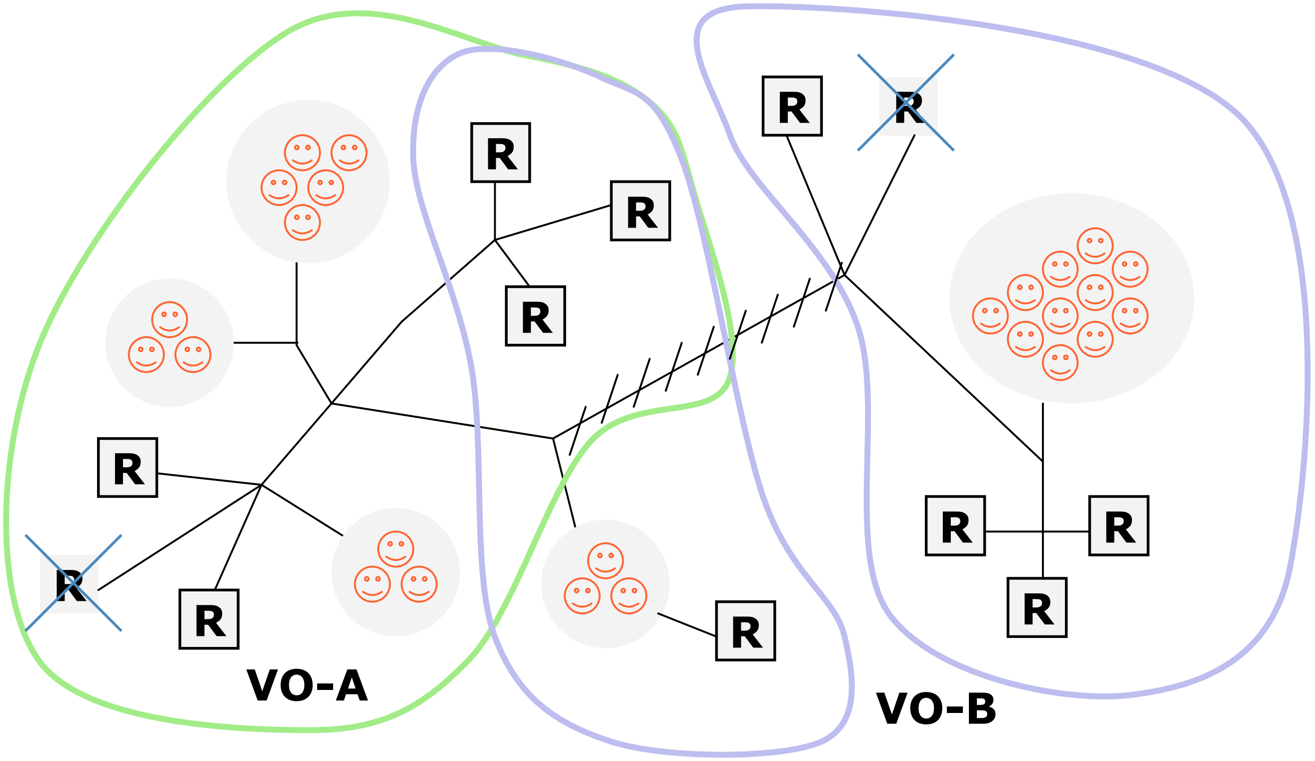
Virtual Organizations

- Distributed resources and people
- Linked by networks, crossing administrative domains
- Sharing resources, common goals
- Dynamic

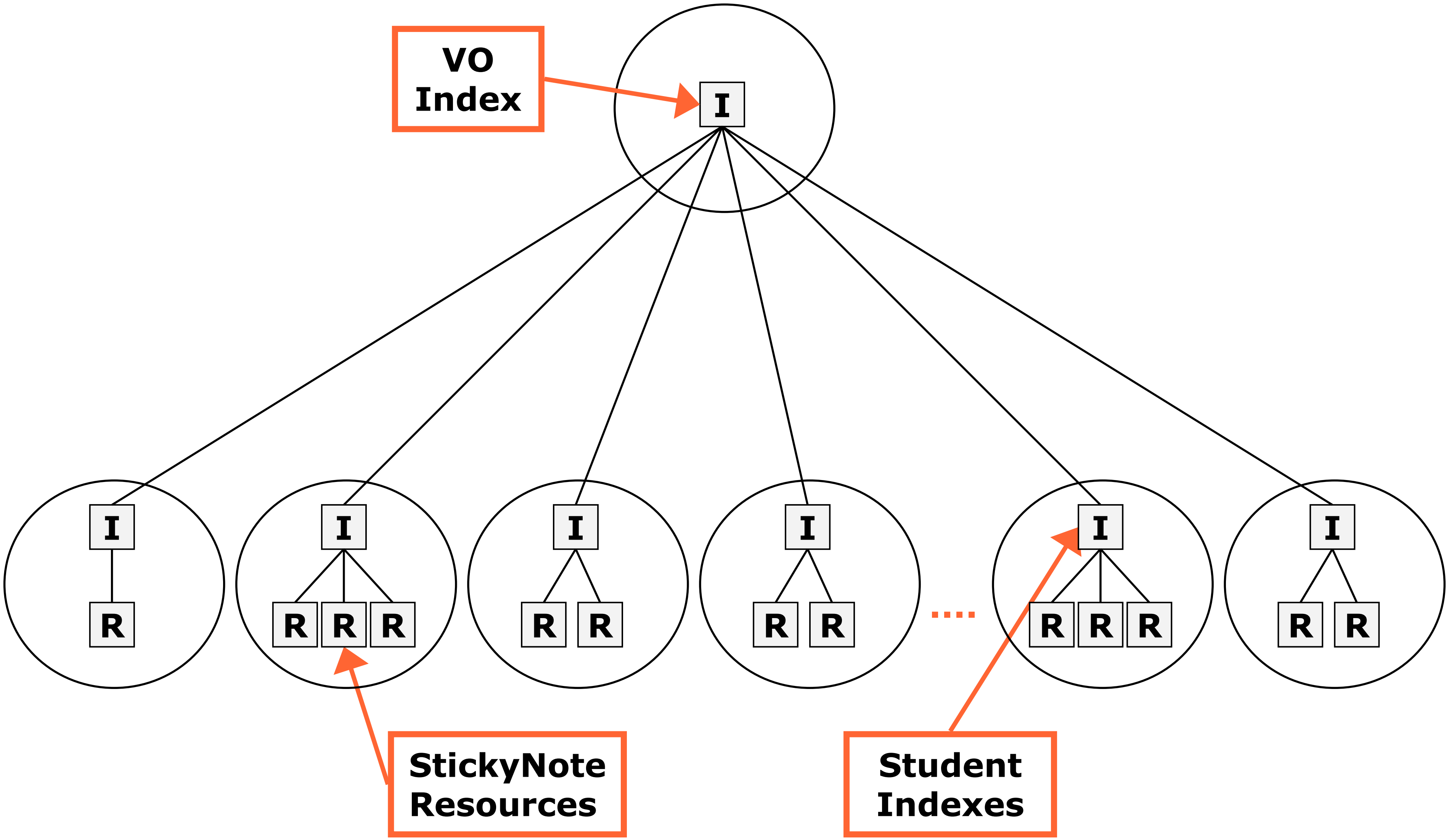


Virtual Organizations

- Distributed resources and people
- Linked by networks, crossing administrative domains
- Sharing resources, common goals
- Dynamic
- Fault tolerant



Tutorial VO

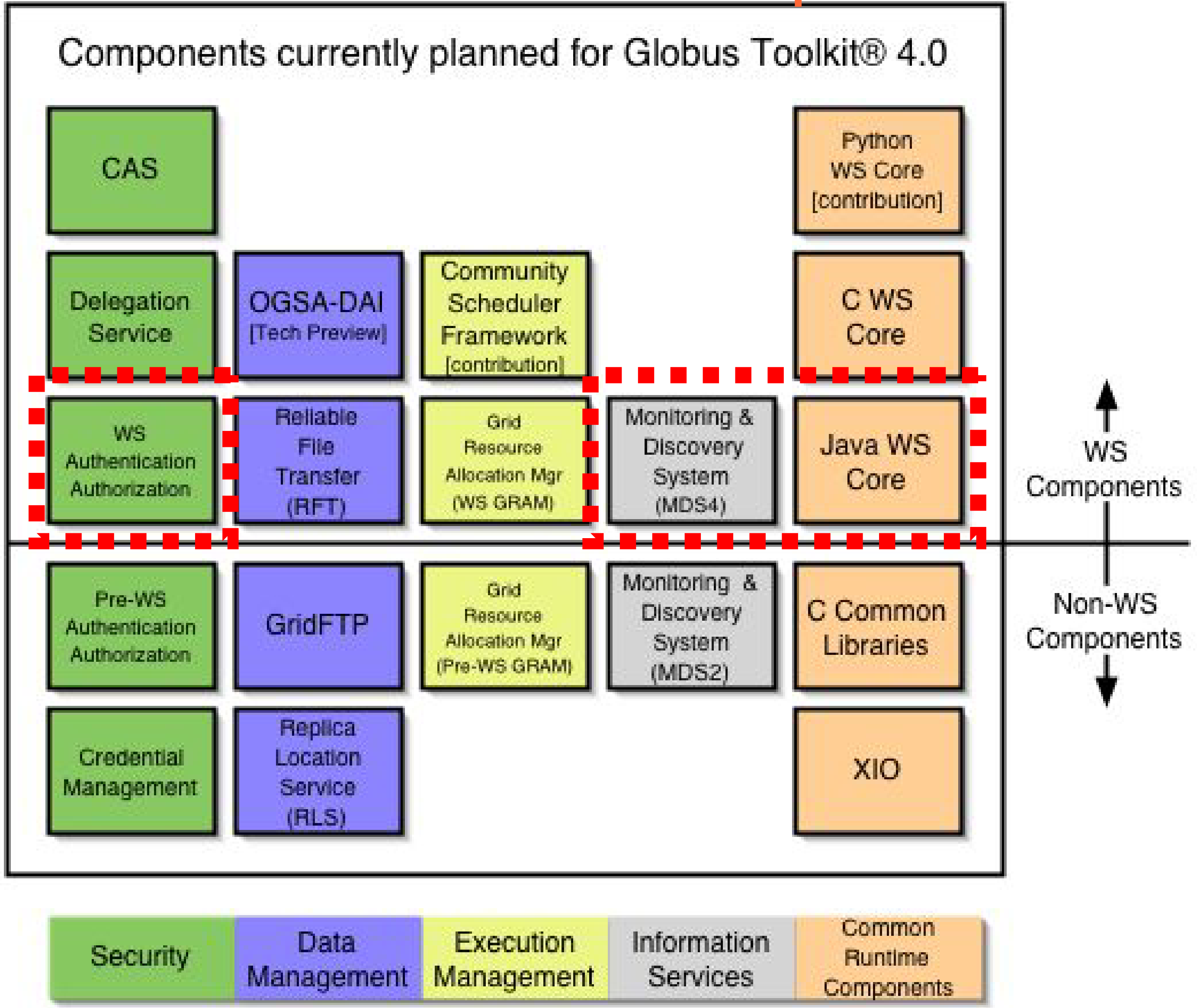

VO Index
I
I
R
I
R
R
R
I
R
R
I
R
R
....
I
R
R
R
I
R
R
StickyNote Resources
Student Indexes

The VO Index

- The community Index will be running on one of the instructor's machines
- The Index represents a collection of Resources
  - A way for each StickyNote Resource to advertise its existence to the VO
  - A copy of each note's Resource Properties will be cached in a single place

the globus toolkit®
www.globustoolkit.org

Sidenote: GT is More than a Service Development Kit!

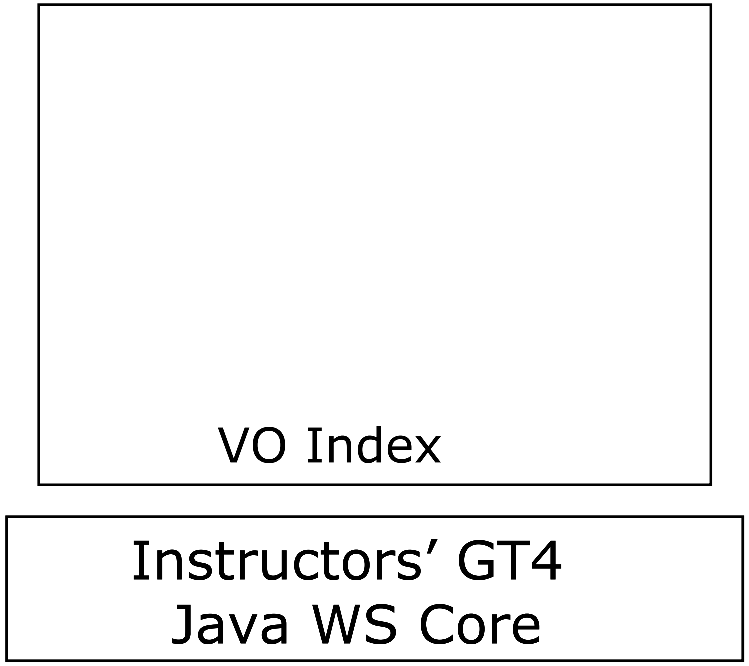


Components currently planned for Globus Toolkit® 4.0
CAS
Python WS Core [contribution]
Delegation Service
OGSA-DAI [Tech Preview]
Community Scheduler Framework [contribution]
C WS Core
WS Authentication Authorization
Reliable File Transfer (RFT)
Grid Resource Allocation Mgr (WS GRAM)
Monitoring & Discovery System (MDS4)
Java WS Core
WS Components
Non-WS Components
Pre-WS Authentication Authorization
GridFTP
Grid Resource Allocation Mgr (Pre-WS GRAM)
Monitoring & Discovery System (MDS2)
C Common Libraries
Credential Management
Replica Location Service (RLS)
XIO
Security
Data Management
Execution Management
Information Services
Common Runtime Components

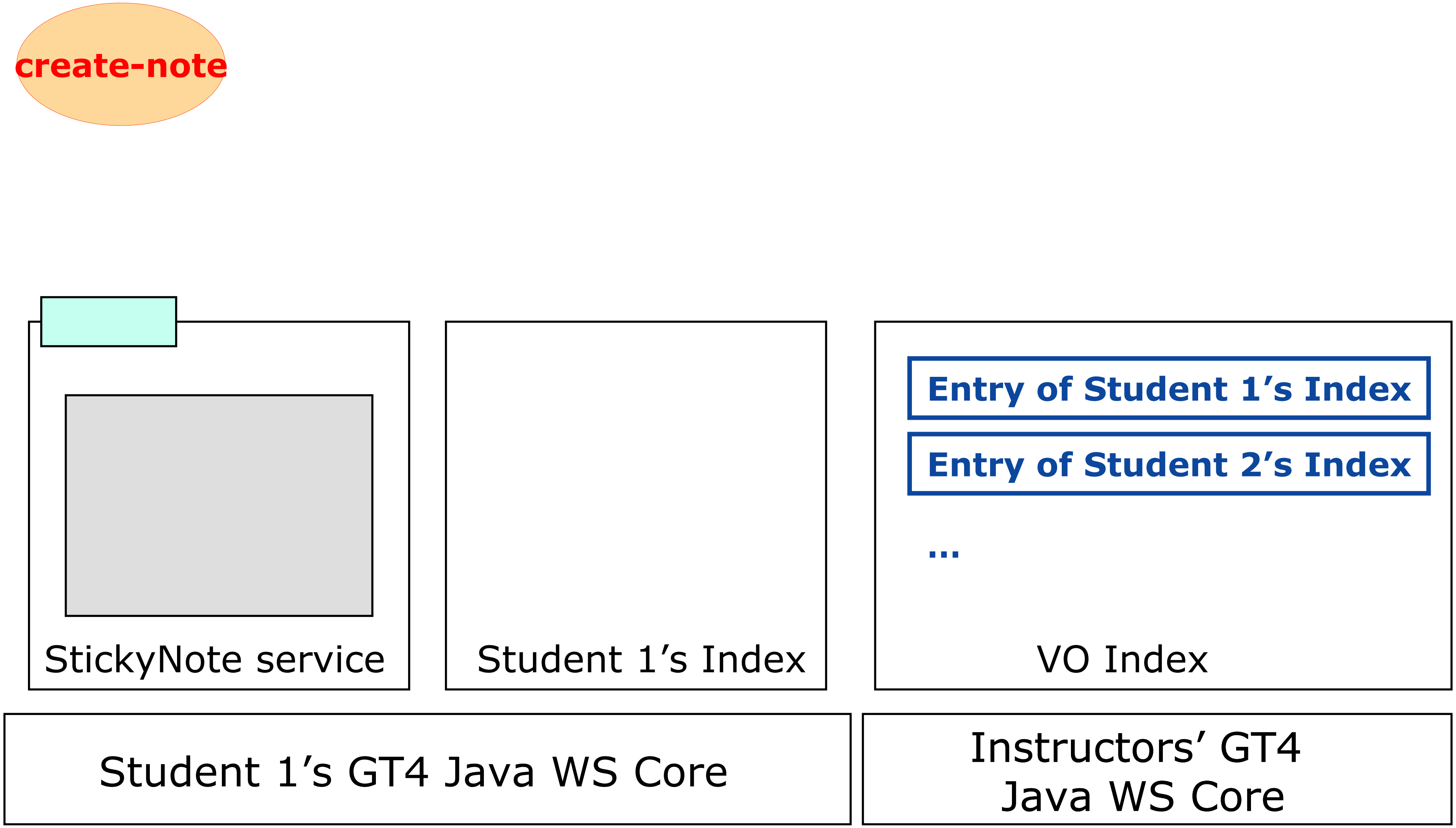
Registering to the Index

- We configure the local Index to register with the VO Index
- Then we will add code to the Resource Home's add operation to register to the local Index
- Because our local index is registered to the VO Index, the information will propagate

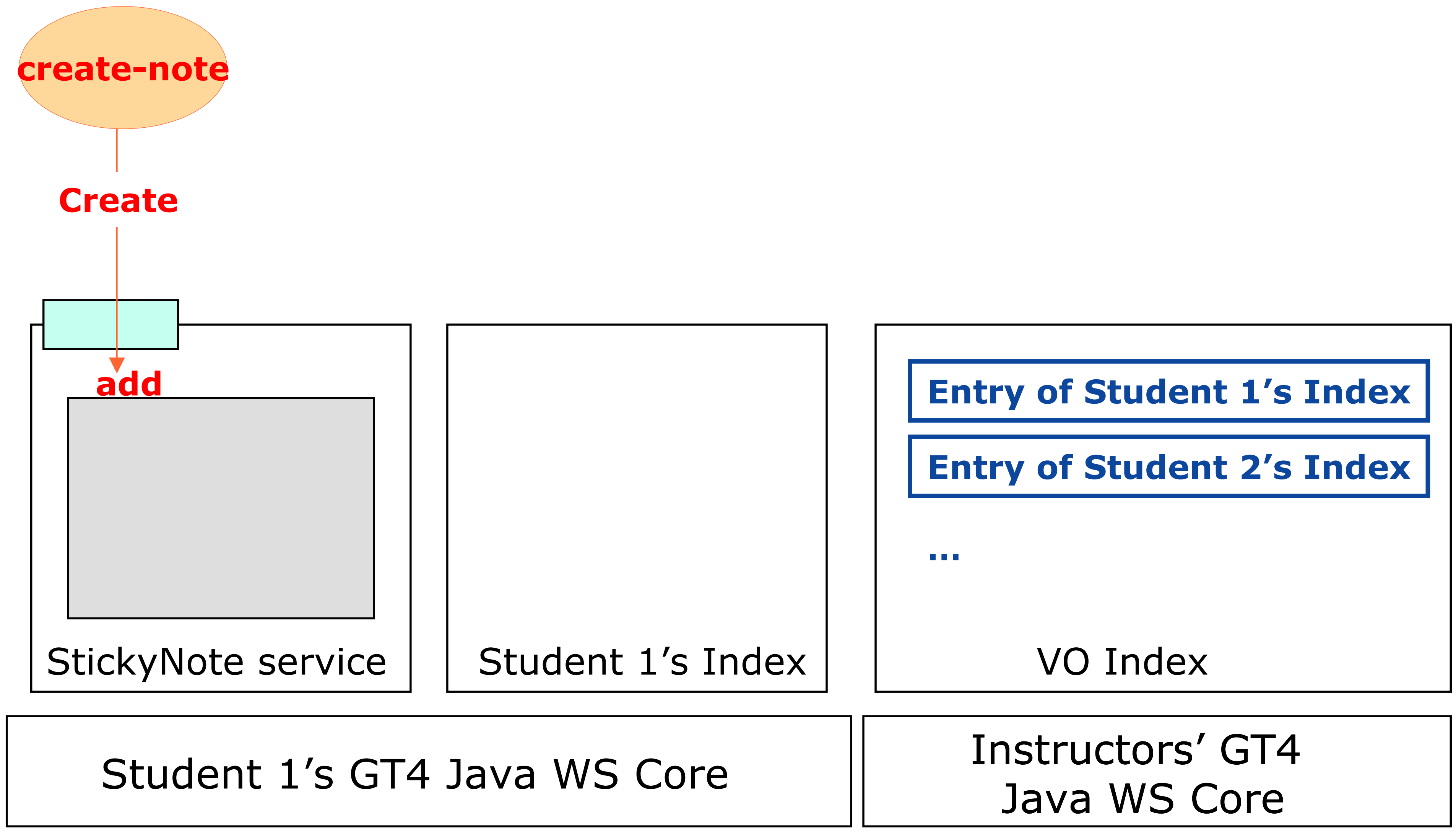
Resource Creation Runtime Revisited



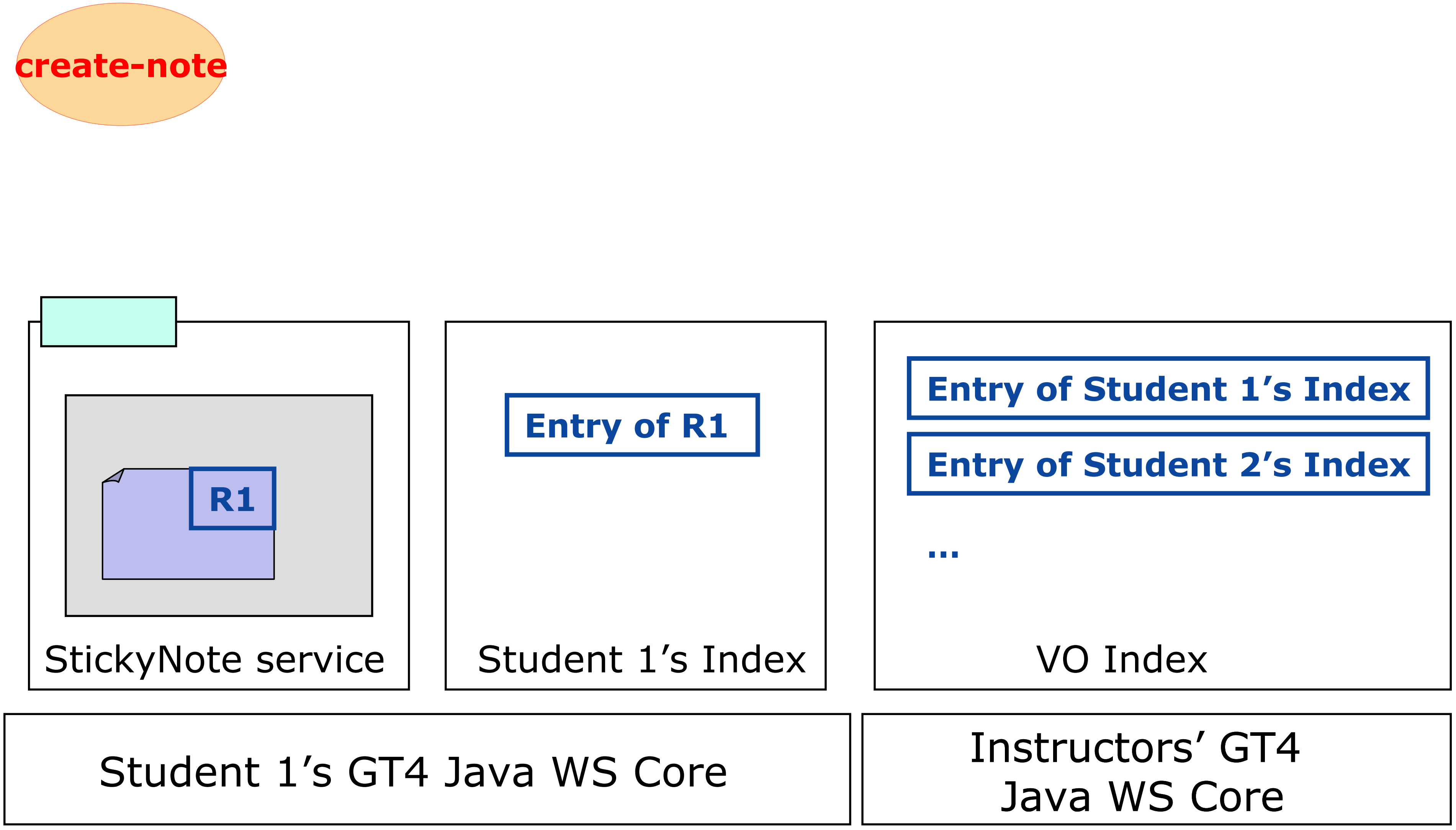
Resource Creation Runtime Revisited


create-note
Entry of Student 1’s Index
Entry of Student 2’s Index
…
StickyNote service
Student 1’s Index
VO Index
Student 1’s GT4 Java WS Core
Instructors’ GT4 Java WS Core

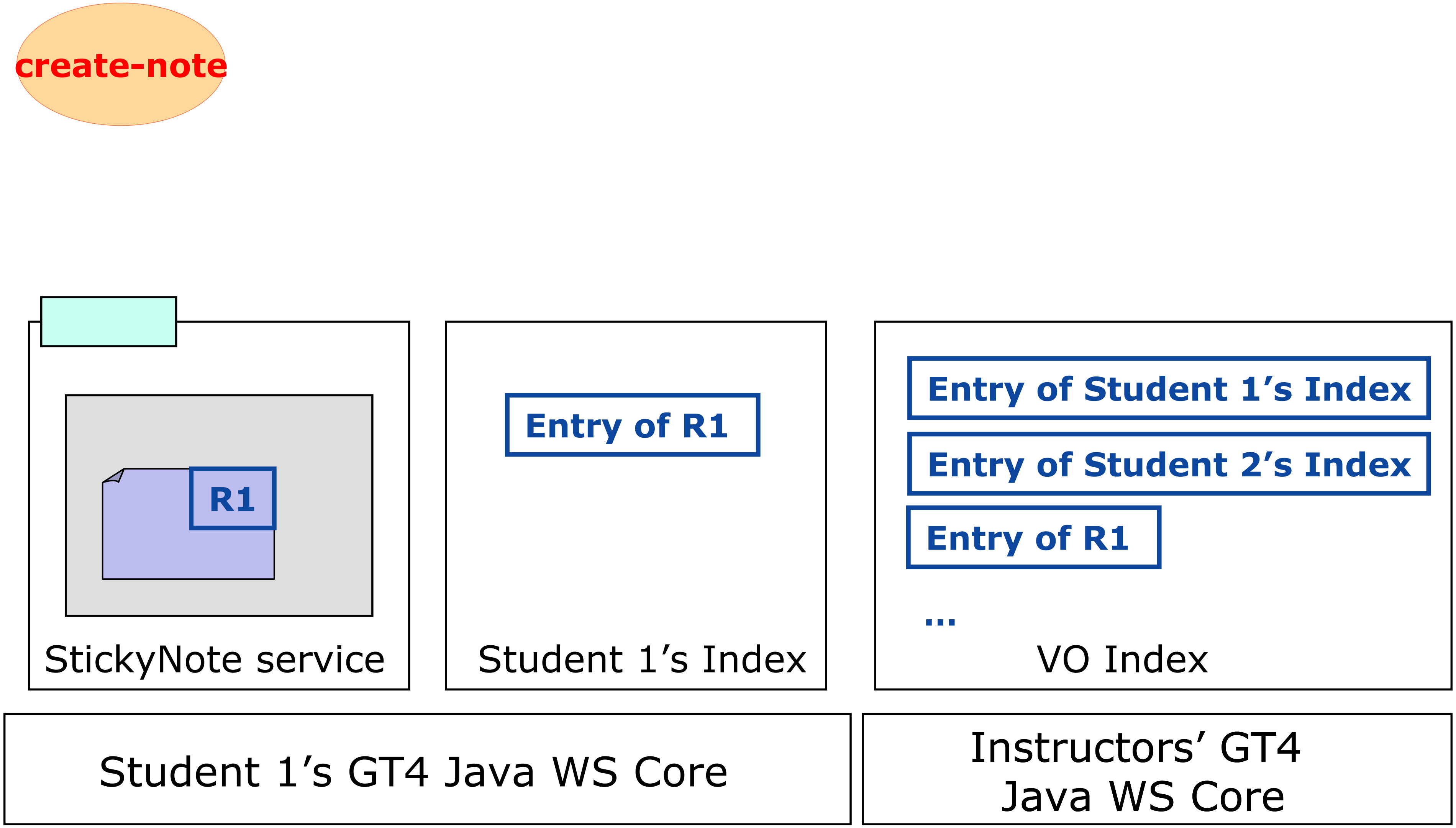
Resource Creation Runtime Revisited


create-note
Create
add
StickyNote service
Student 1's Index
Entry of Student 1's Index
Entry of Student 2's Index
...
VO Index
Student 1's GT4 Java WS Core
Instructors' GT4 Java WS Core

Resource Creation Runtime Revisited


create-note
R1
StickyNote service
Entry of R1
Student 1's Index
Entry of Student 1's Index
Entry of Student 2's Index
...
VO Index
Student 1's GT4 Java WS Core
Instructors' GT4 Java WS Core

Resource Creation Runtime Revisited


create-note
R1
StickyNote service
Entry of R1
Student 1's Index
Entry of Student 1's Index
Entry of Student 2's Index
Entry of R1
...
VO Index
Student 1's GT4 Java WS Core
Instructors' GT4 Java WS Core

What Attendees Should Do

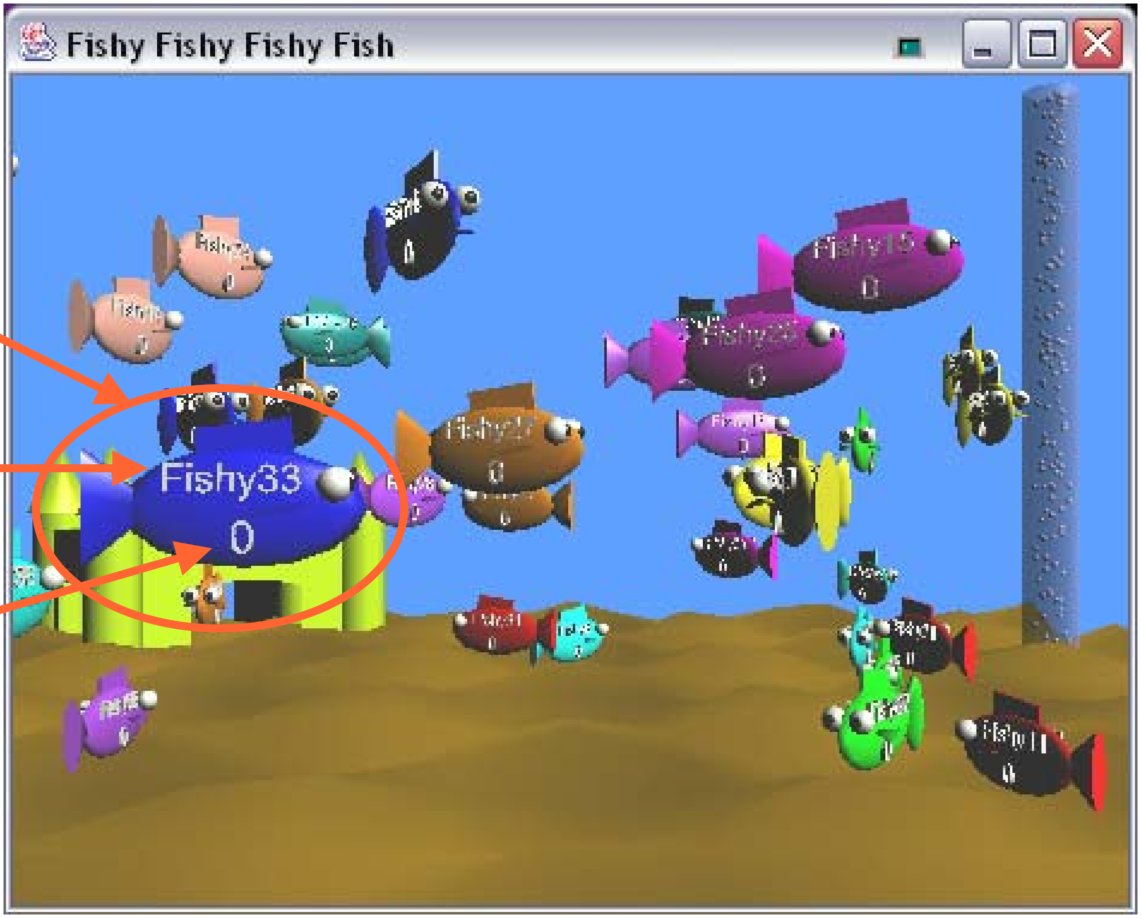
- Register StickyNote Resources with the VO indexing infrastructure
  - Override the add method in the StickyNote’s Resource Home to register new Resources with the local Index
  - Configure the local Index to register with the VO Index
- Browse the VO index to see that your service has registered

What Attendees Should See

A representation of your note inside a representation of the tutorial index

Your IP

note text



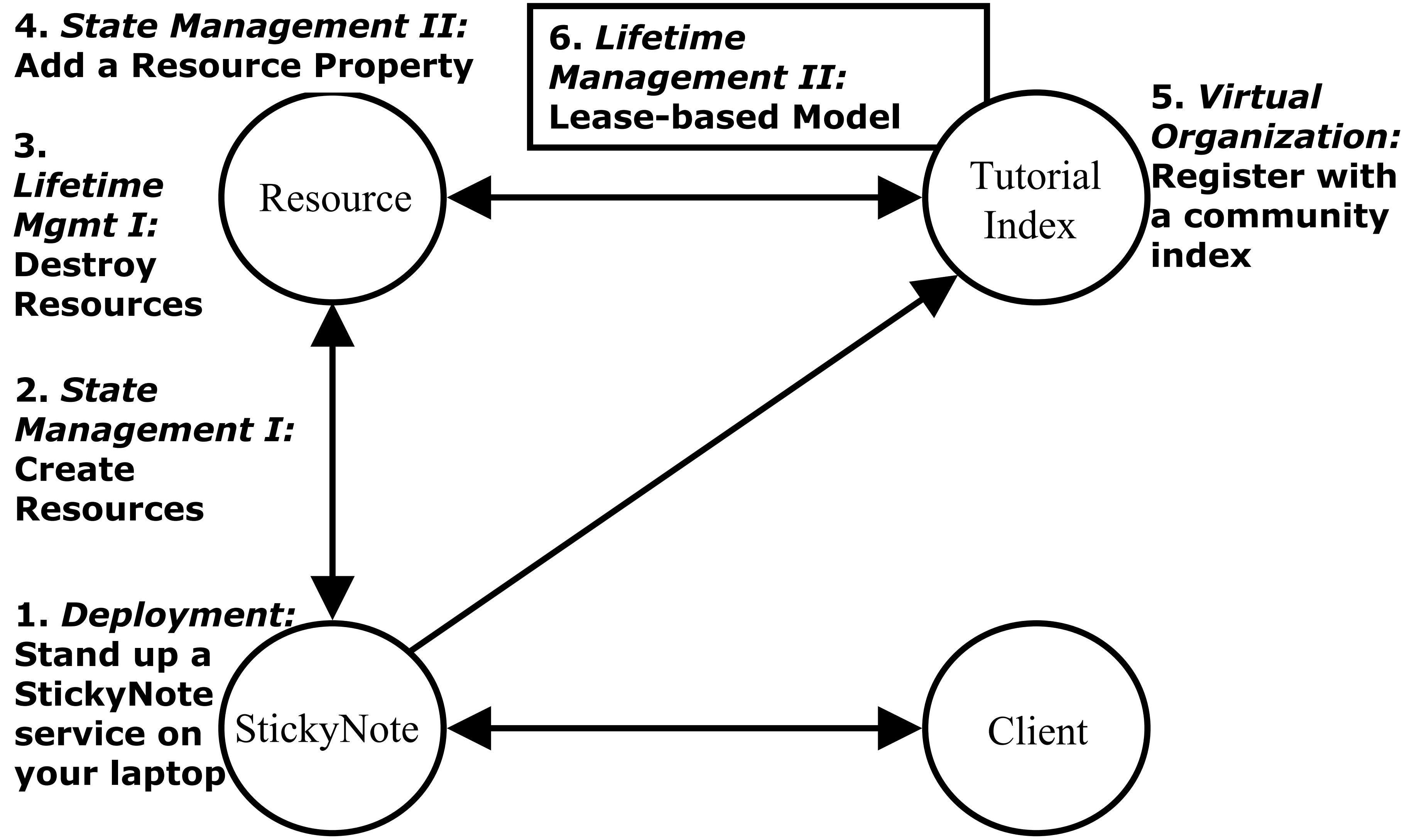
Exercise 5 Review

- The VO Index contains the aggregate contents of student Index services
- You may browse the Index with wsrf-query or WebMDS or the fishtank
- Destroying a resource causes it to be removed from the Index
- Virtual Organizations make the world go round

the globus toolkit®
www.globustoolkit.org

How to Build a Service Using GT4

Exercise 6: Leased-Based Lifetime Mgmt


4. *State Management II:* Add a Resource Property
3. *Lifetime Mgmt I:* Destroy Resources
2. *State Management I:* Create Resources
1. *Deployment:* Stand up a StickyNote service on your laptop
6. *Lifetime Management II:* Lease-based Model
5. *Virtual Organization:* Register with a community index
Resource
Tutorial Index
StickyNote
Client

Lease-Based Lifetime Management

- In Grid computing, we need a mechanism to clean up old/unwanted state
- So far we've cleaned up after ourselves explicitly using the Destroy operation
- Manually destroying our own notes is fine in some contexts, but is not always possible or appropriate
  - If a client dies or the network goes down, you would like a mechanism to clean up unused state automatically
- Under the lease-based model resources must be kept alive by interested parties

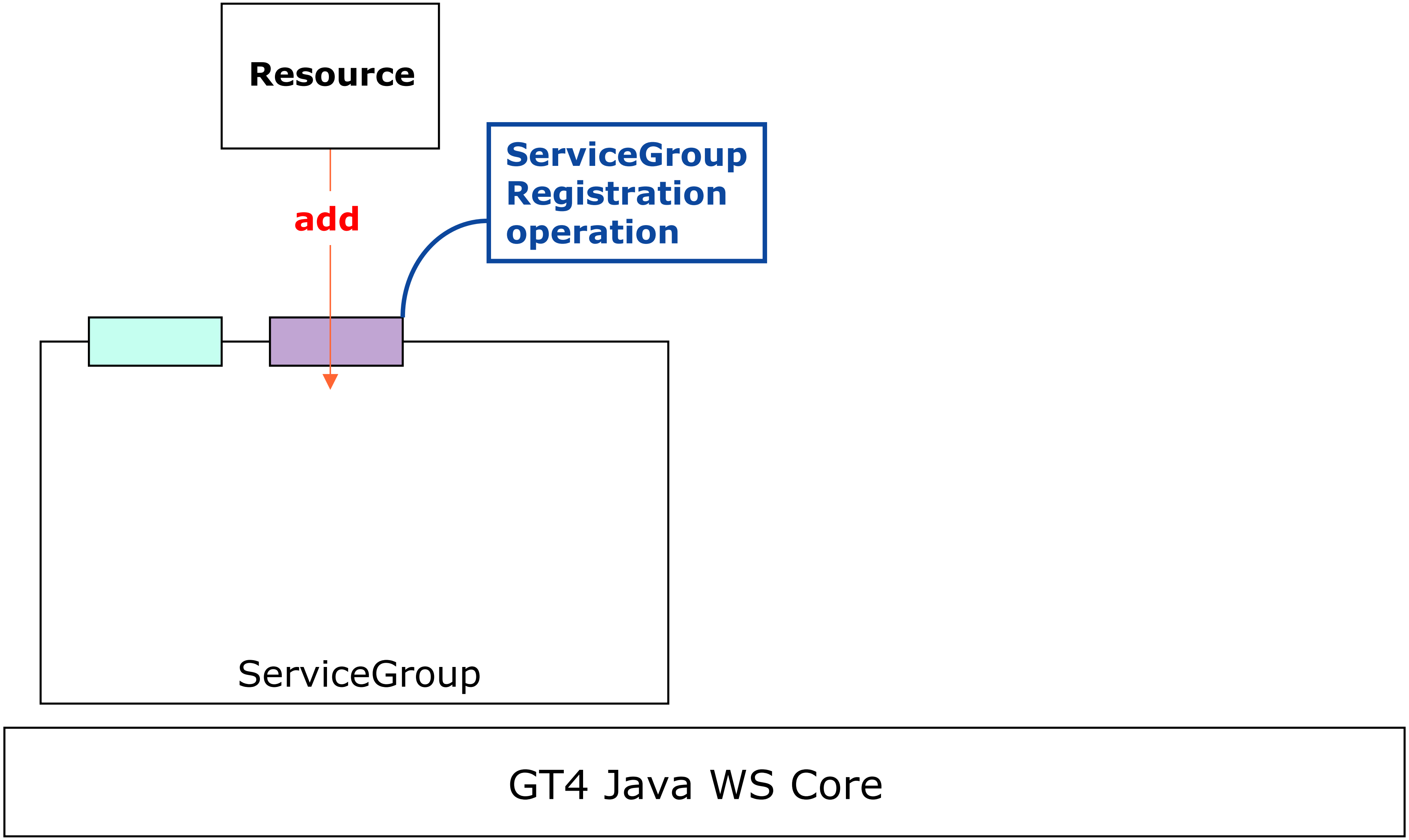
Lease-Based Lifetime Management

- To illustrate this, we will support scheduled destruction of notes
- If the lease expires, the resource will be destroyed
- Interested parties can renew the lease
  - wsrf-set-termination-time
- The termination time is exposed as a Resource Property

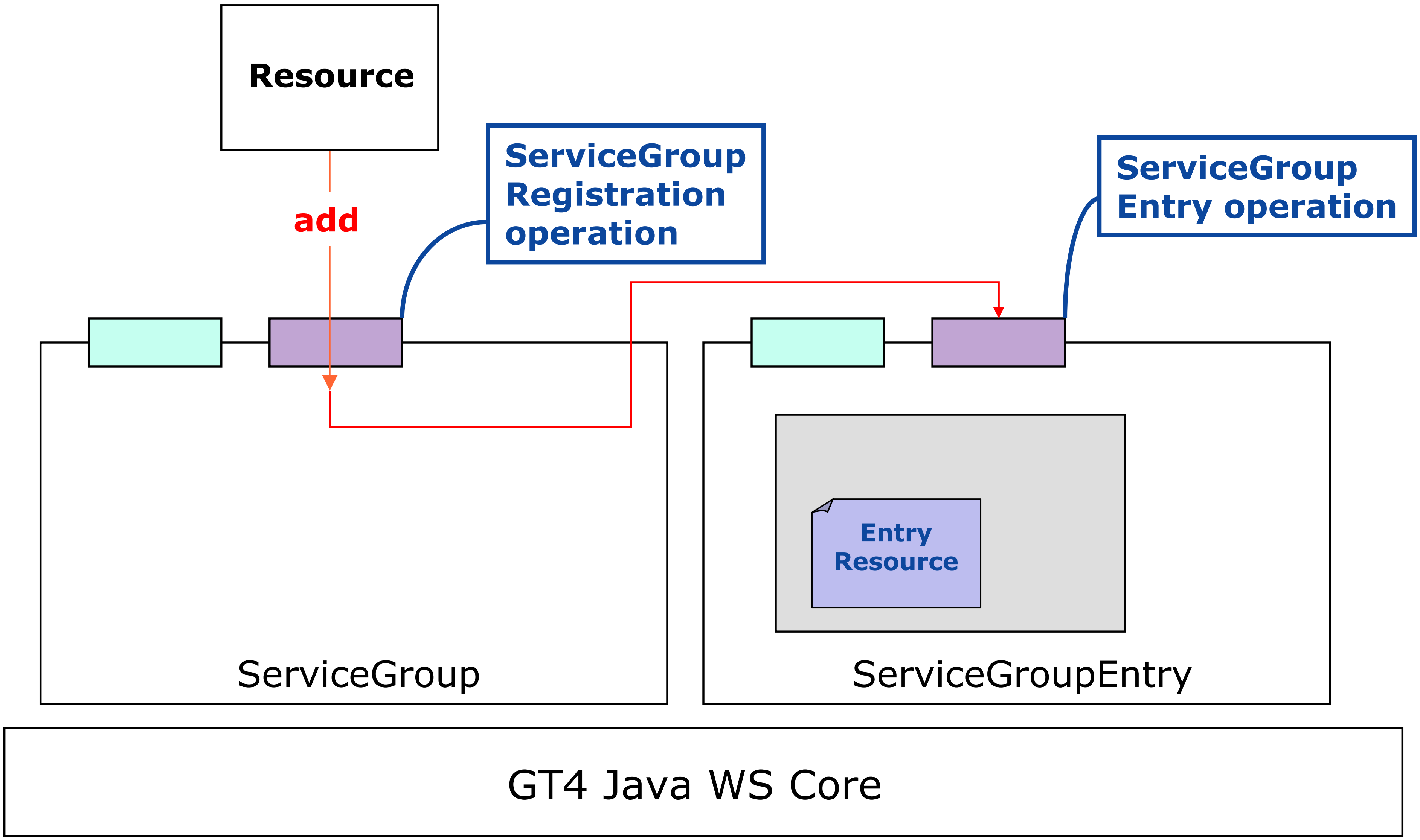
Service Group

- Indexes are a kind of **service group**
- Service groups represent collections of services and resources
- Each registration is represented as a ServiceGroupEntry resource
- Each entry resource has a lifetime, which will be renewed automatically by the StickyNote service

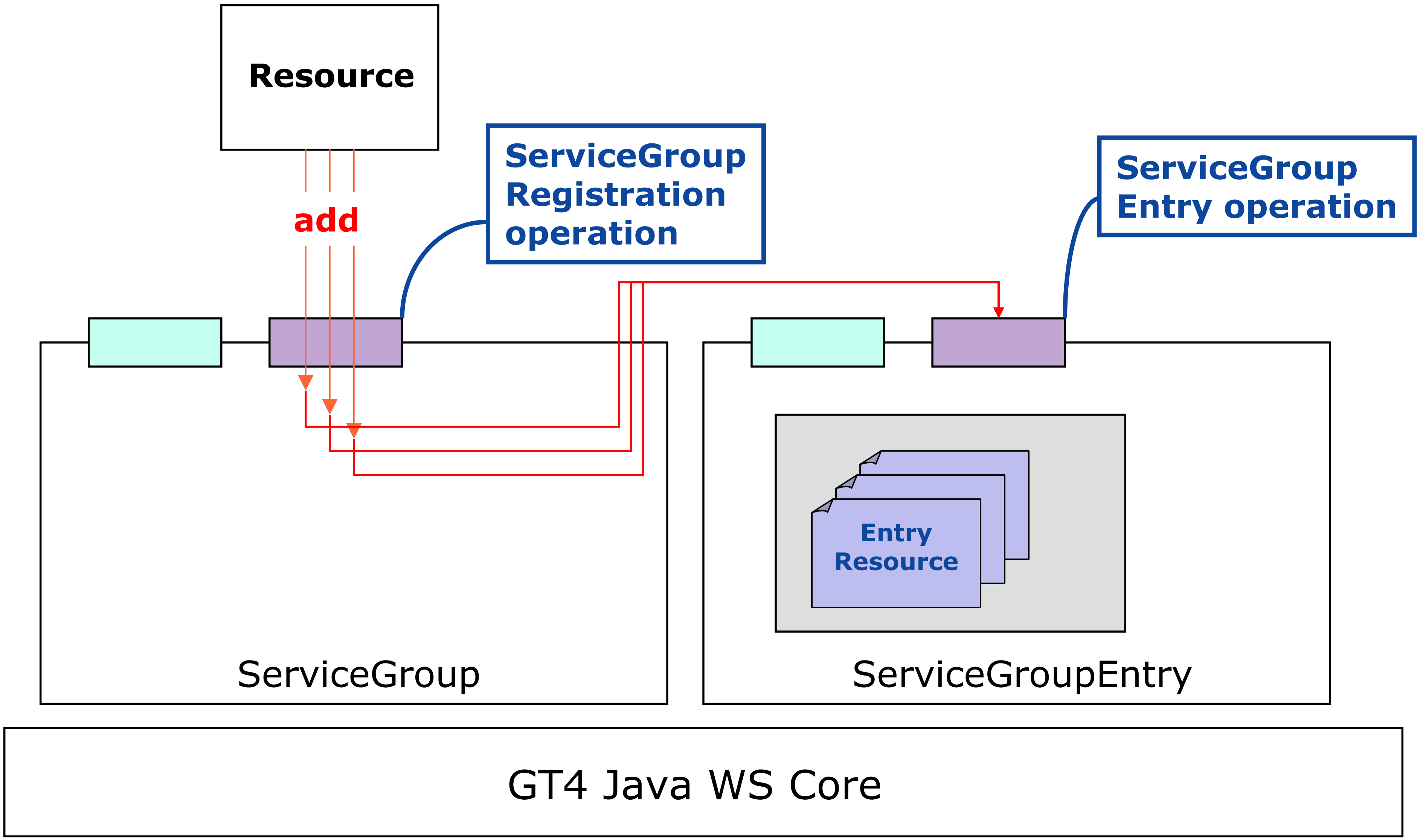
Service Group Structure



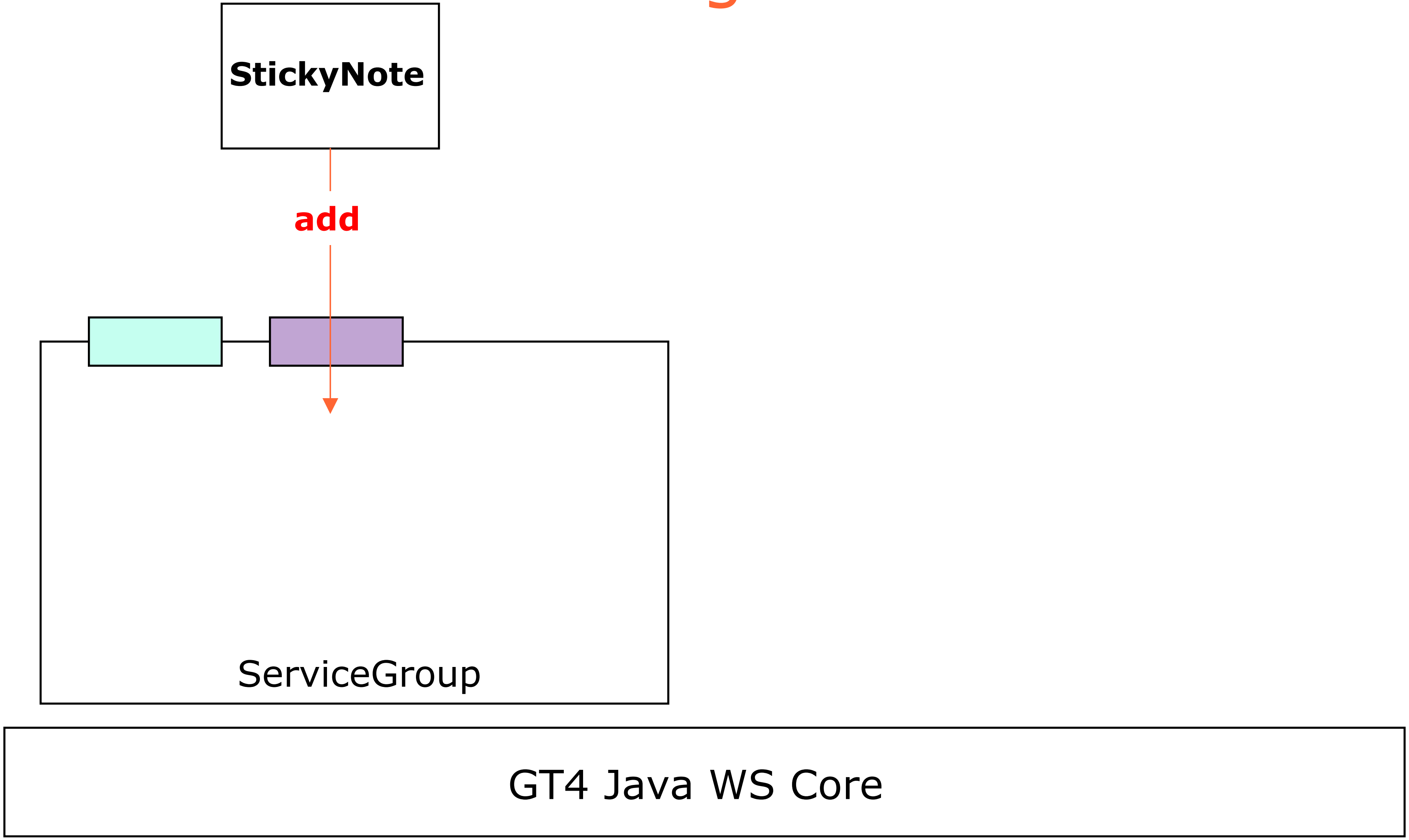
Service Group Structure



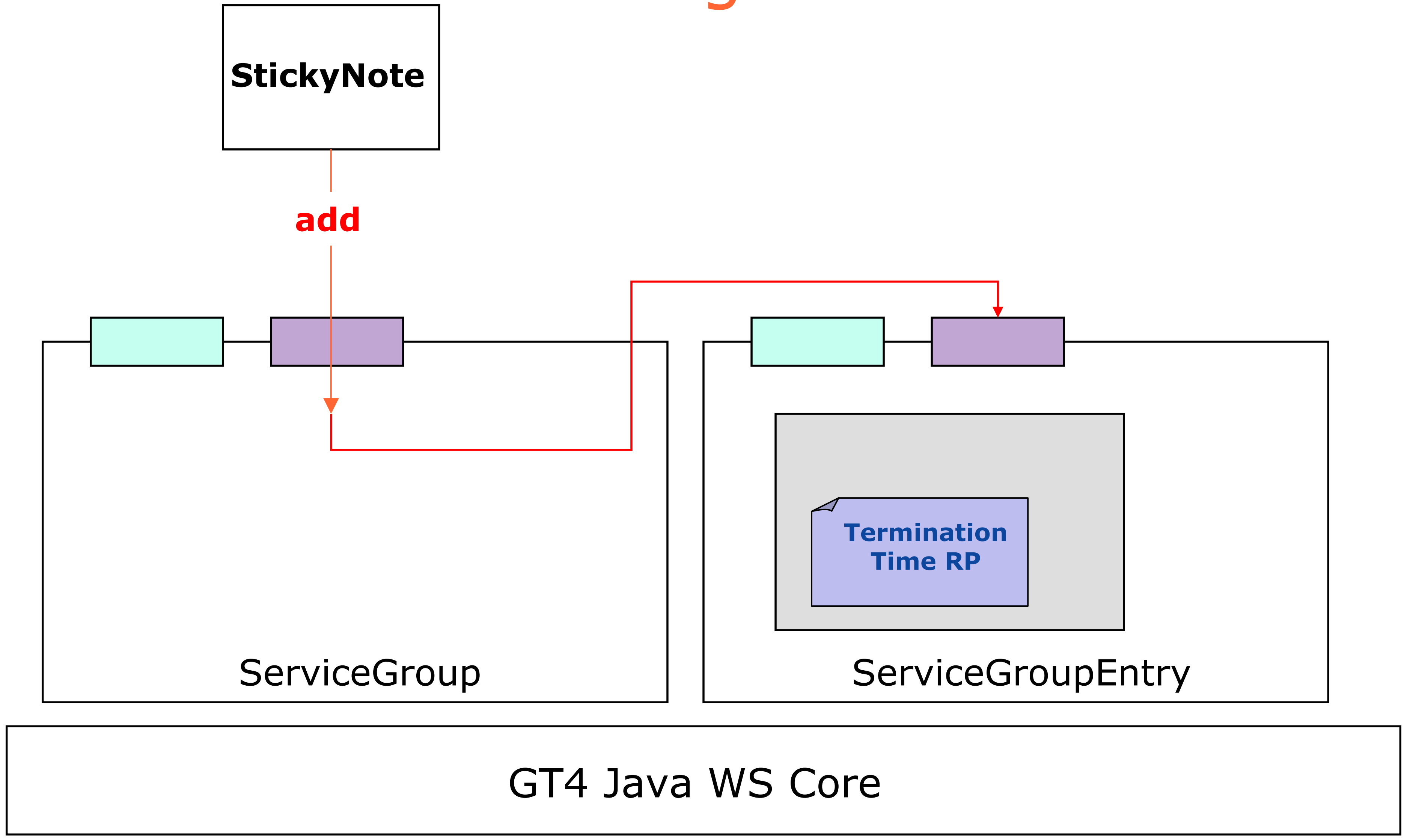
Service Group Structure



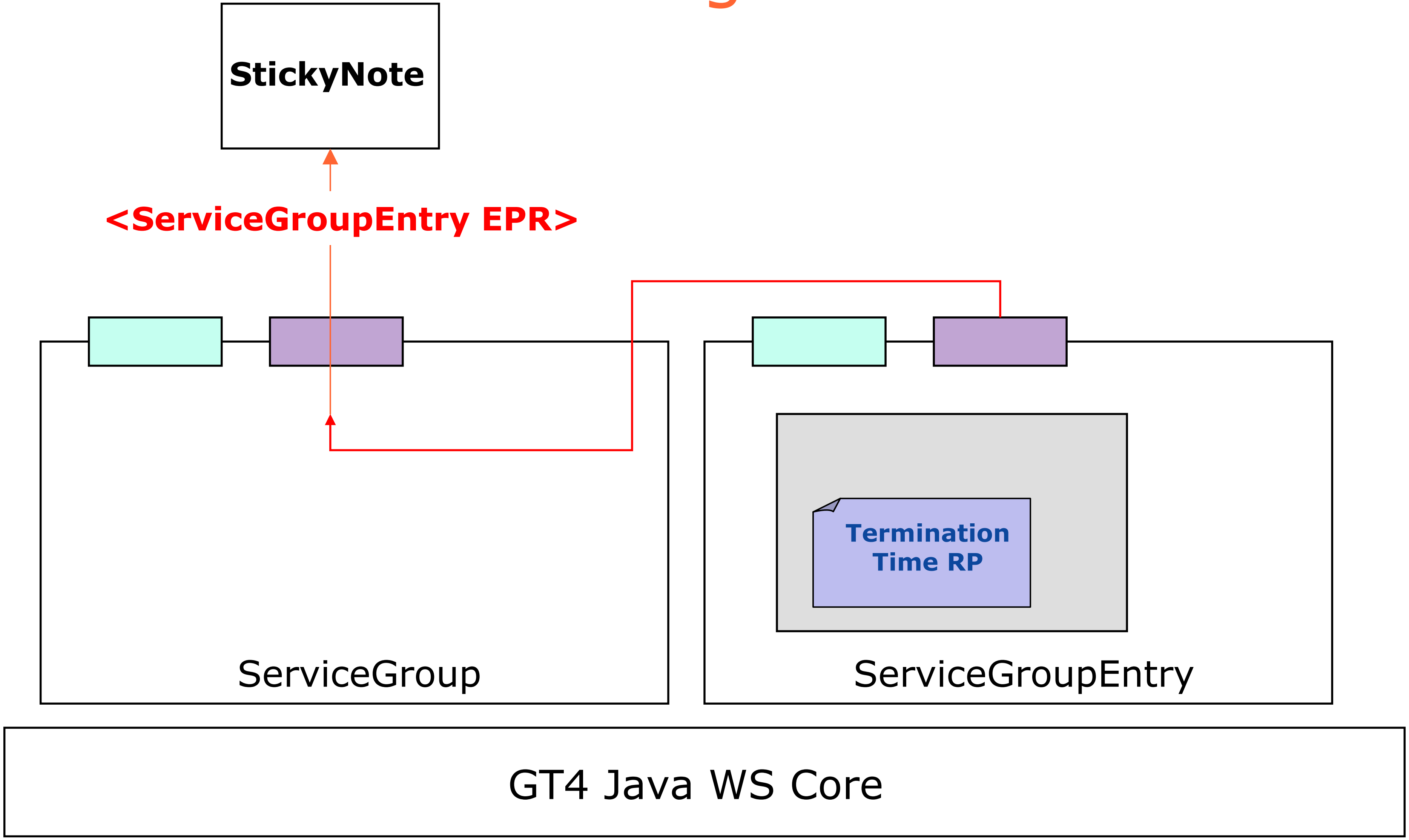
Service Group Lifetime Management Runtime



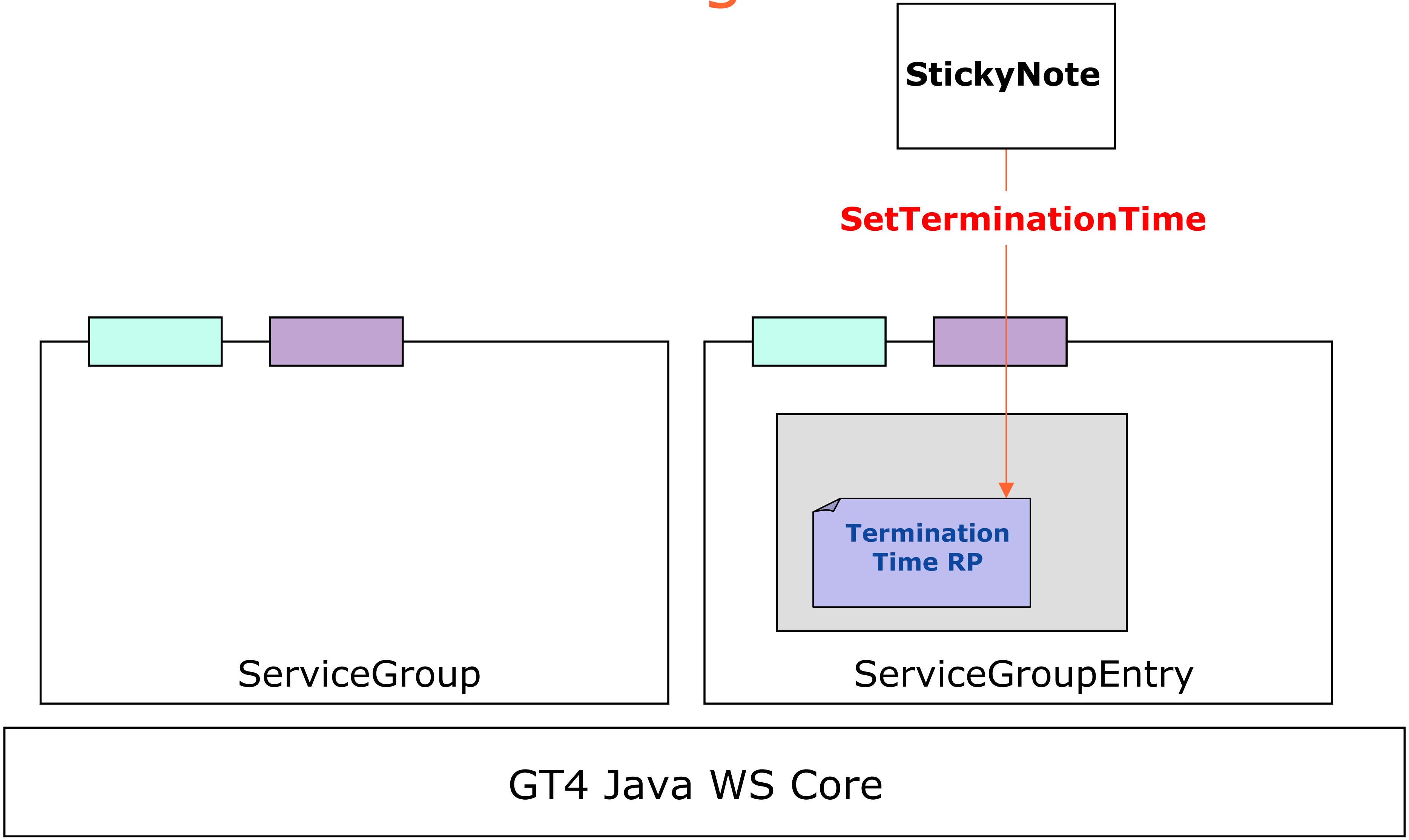
Service Group Lifetime Management Runtime



Service Group Lifetime Management Runtime

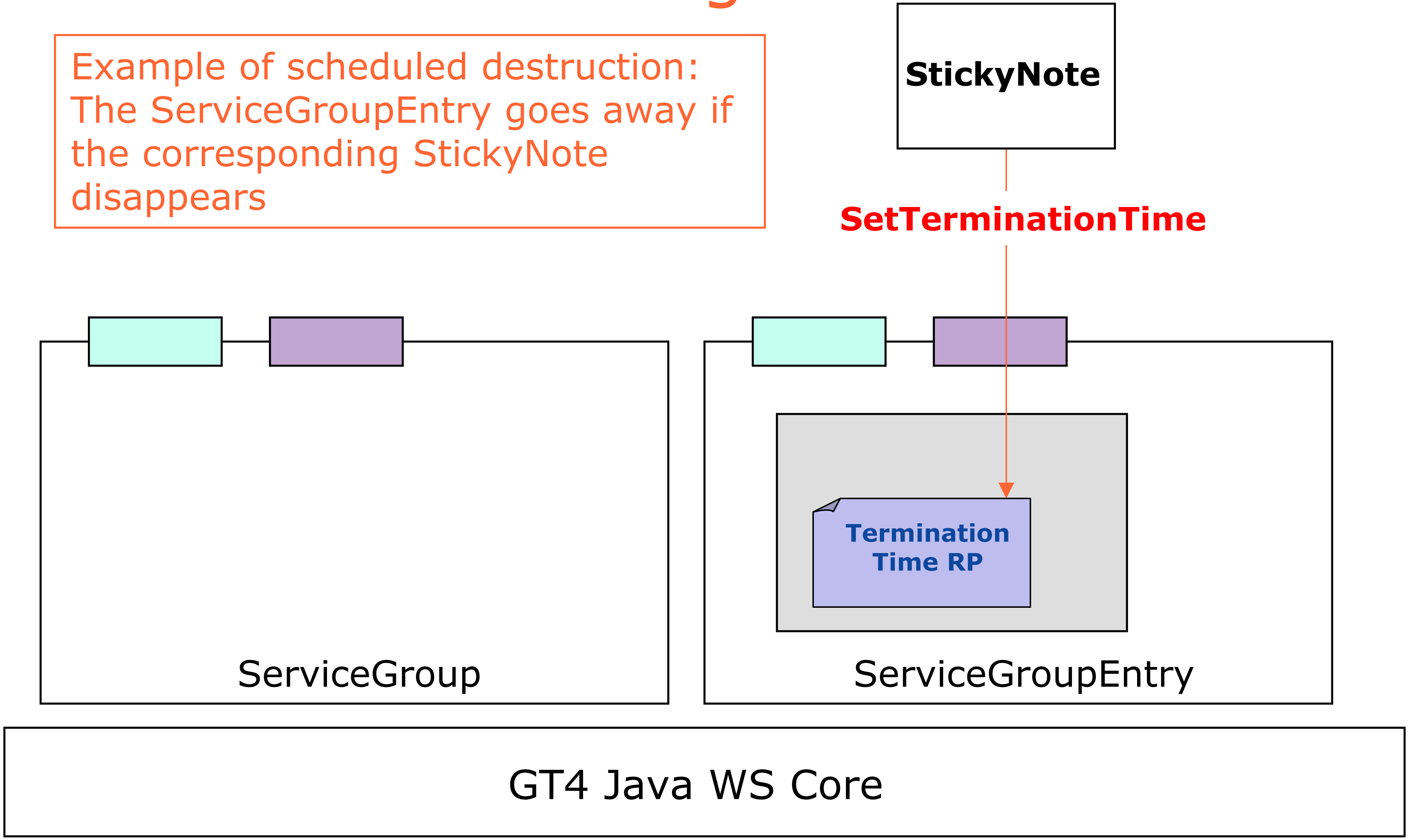


Service Group Lifetime Management Runtime

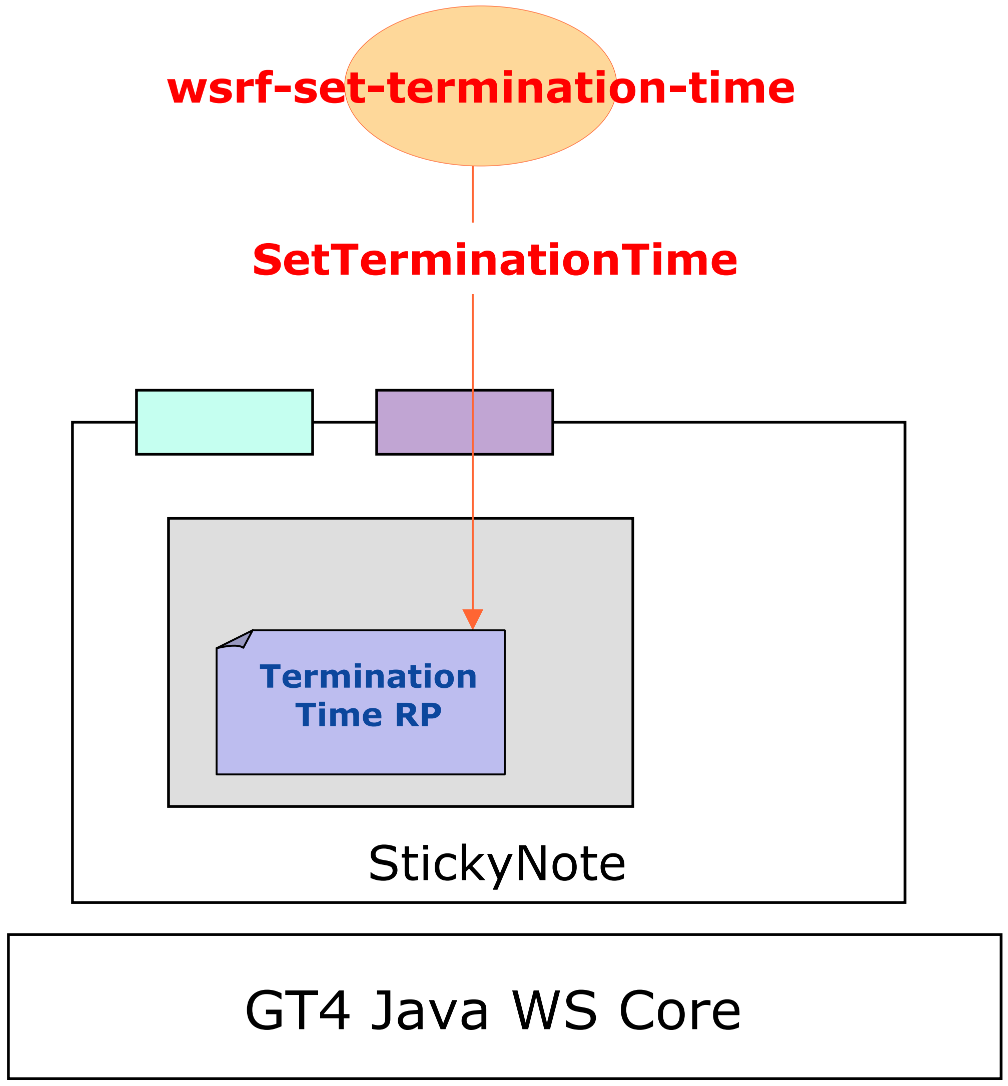


Service Group Lifetime Management Runtime

Example of scheduled destruction: The ServiceGroupEntry goes away if the corresponding StickyNote disappears



StickyNote Lifetime Management Runtime



What Attendees Should Do

- Add support for lease-based lifetime management to the StickyNote service
- Start the container, create a note
- Run wsrf-set-termination-time on that note
- Watch the note be destroyed after the scheduled termination time

What Attendees Should See

```
$ wsrf-set-termination-time -e note-
   974145926.epr 30
(wait…)
$ show-note -e note-974145926.epr
Error:
   org.oasis.wsrf.properties.ResourceUnknow
   nFaultType
```

Exercise 6 Review

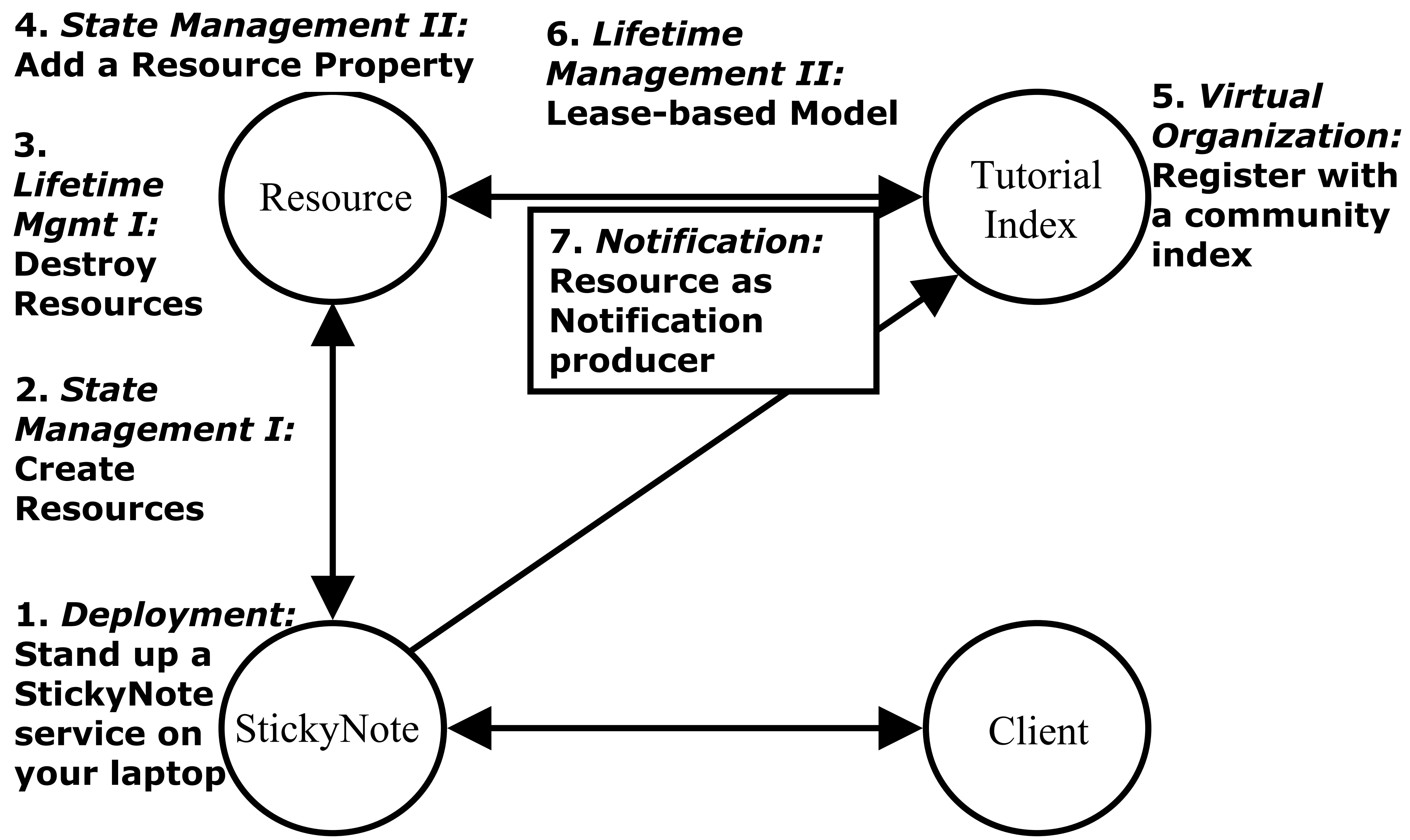
- Resources can have a termination time
- Lifetime can be changed

- Service Groups contain resources
- Registrations are managed using standard lifetime mechanisms

How to Build a Service Using GT4

- Overview of Services and GT4
- Build a Service
  - 1. Getting Started: Deploy a Service
  - 2. State Management Part I: Create Resources
  - 3. Lifetime Management Part I: Destroy Resources
  - 4. State Management Part II: Add a Resource Property
  - 5. Building a VO: Register with a Community Index
  - 6. Lifetime Management Part II: Lease-based Model
  - **7. Notification: Resource as Notification Producer**
  - 8. Discovery: Find a Resource
  - 9. Security Part I: Service-Level
  - 10. Security Part II (optional): Resource-Level
- Overview of Tools for GT4 Service Developers
- Tutorial Summary
- Ideas for Further Work

Exercise 7: Notification


4. *State Management II:* Add a Resource Property
6. *Lifetime Management II:* Lease-based Model
5. *Virtual Organization:* Register with a community index
3. *Lifetime Mgmt I:* Destroy Resources
Resource
Tutorial Index
7. *Notification:* Resource as Notification producer
2. *State Management I:* Create Resources
1. *Deployment:* Stand up a StickyNote service on your laptop
StickyNote
Client

Notifications

- Right now make a one-off query for RP values
  - Show-note command
- We can use notifications to receive new RP values when the RP changes
- Notifications are from one hosting environment to another

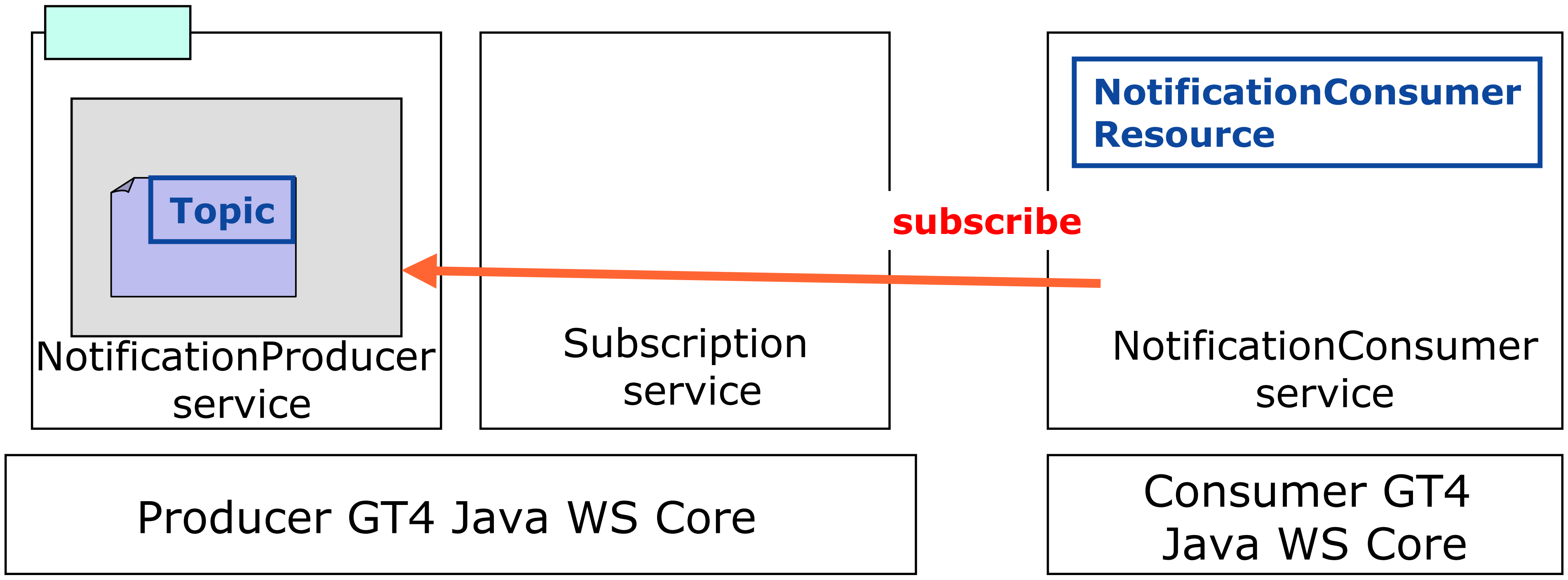
Notification vocabulary

- We can **subscribe** to receive **notifications**
- Subscriptions are for a particular **topic**
- Notifications go from a **Producer** to a **Consumer**
- Each topic has a name
- Topics can be:
  - Changes in RP values
  - Resource destruction
  - Service group changes
  - Other interesting things…

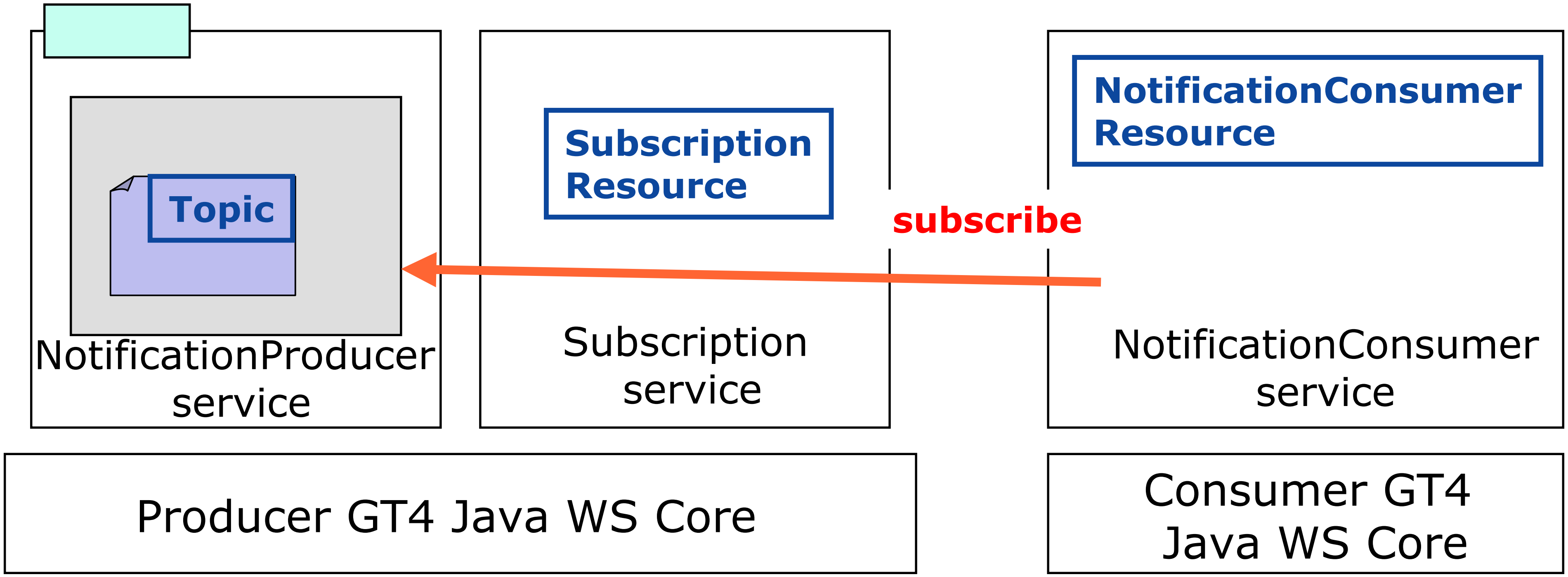
Subscriptions

- Each subscription is represented as a new kind of resource
- We can manage the subscription through this resource
  - Eg. cancel subscription by destroying resource

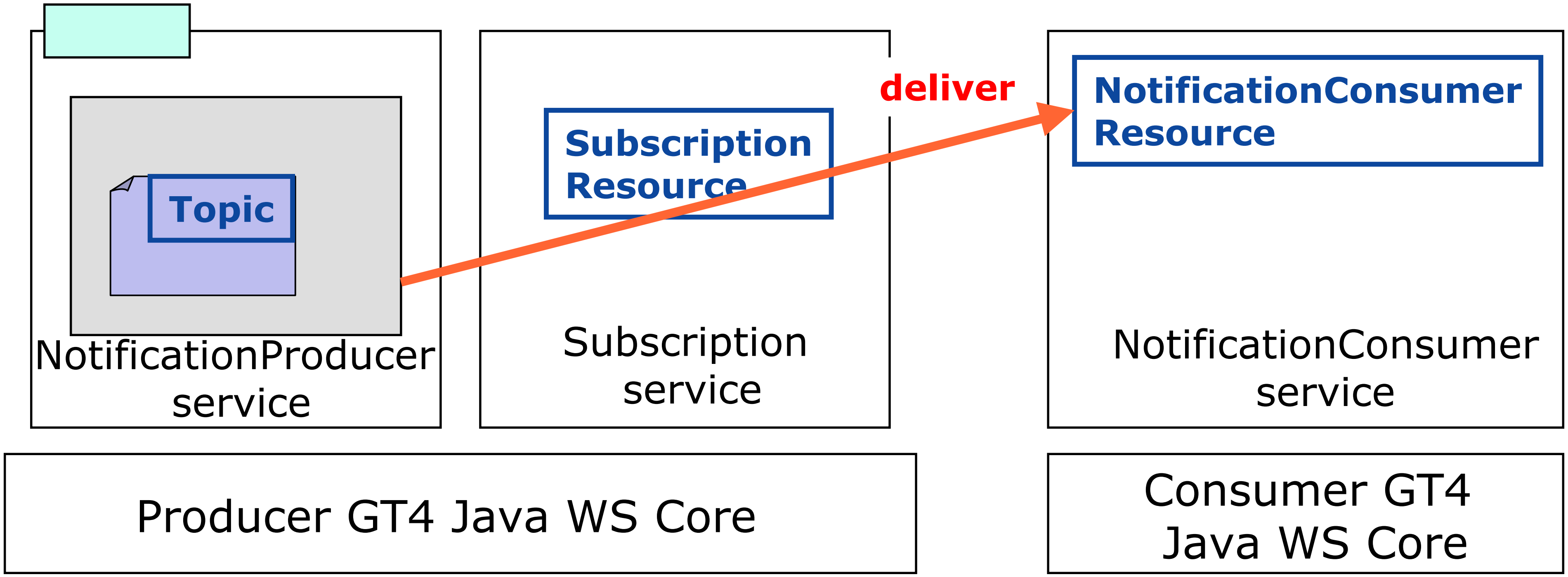
Subscription and Notification


Topic
NotificationProducer service
Subscription service
subscribe
NotificationConsumer Resource
NotificationConsumer service
Producer GT4 Java WS Core
Consumer GT4 Java WS Core

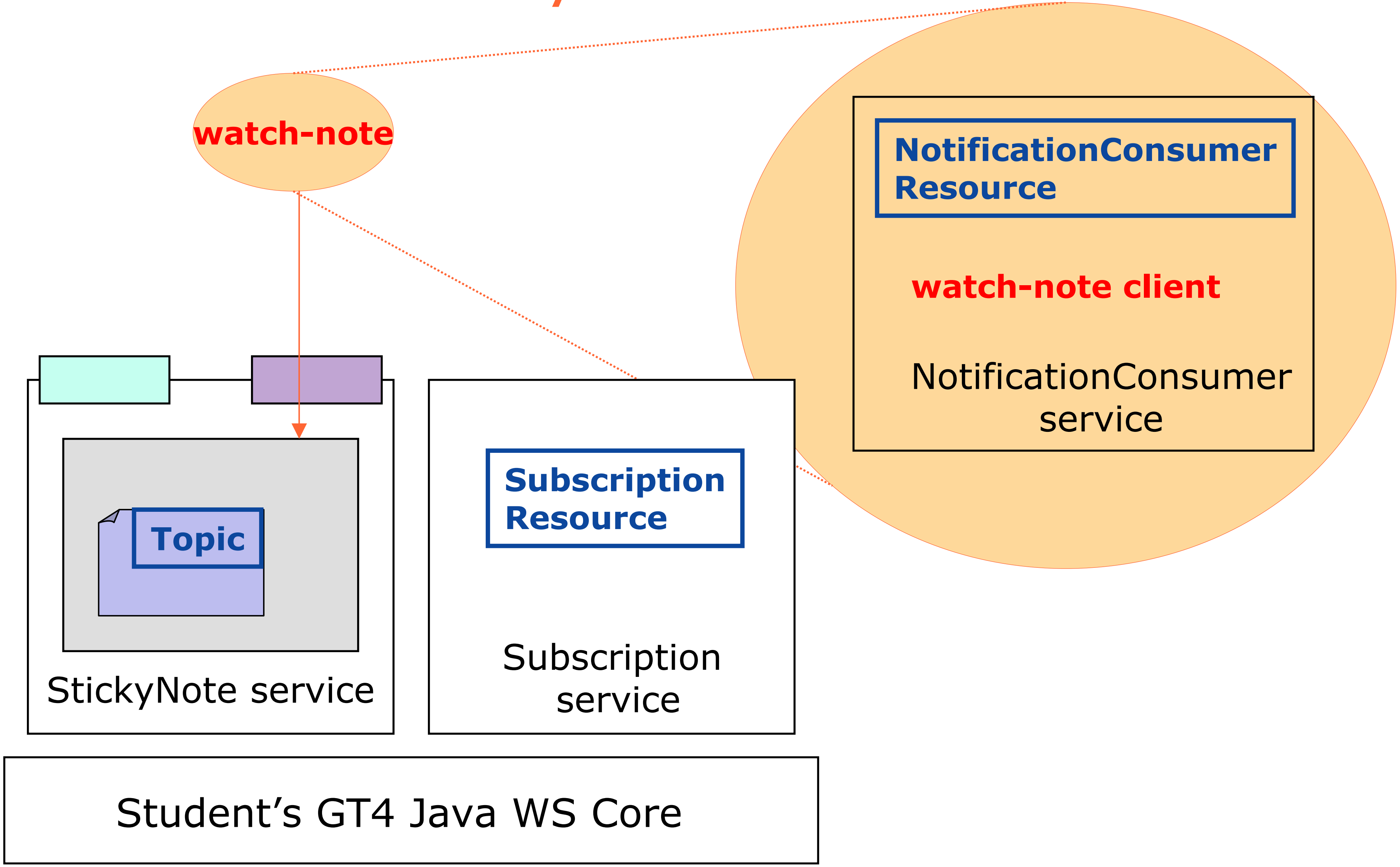
Subscription and Notification


Topic
NotificationProducer service
Subscription Resource
Subscription service
subscribe
NotificationConsumer Resource
NotificationConsumer service
Producer GT4 Java WS Core
Consumer GT4 Java WS Core

Subscription and Notification


Topic
NotificationProducer service
Subscription Resource
Subscription service
deliver
NotificationConsumer Resource
NotificationConsumer service
Producer GT4 Java WS Core
Consumer GT4 Java WS Core

Sticky note notification



What Attendees Should Do

- Make StickyNote service a NotificationProducer
- Make message RP into a topic
- Implement the watch-note client
- Start the container, create a note
- Run watch-note on that note
- Change the contents of the note using write-note, and see the notifications being delivered to watch-note

What Attendees Should See

- watch-note

```
Waiting for notification. Ctrl-C to end.
Received notification with value: coconut
```

Exercise 7 Review

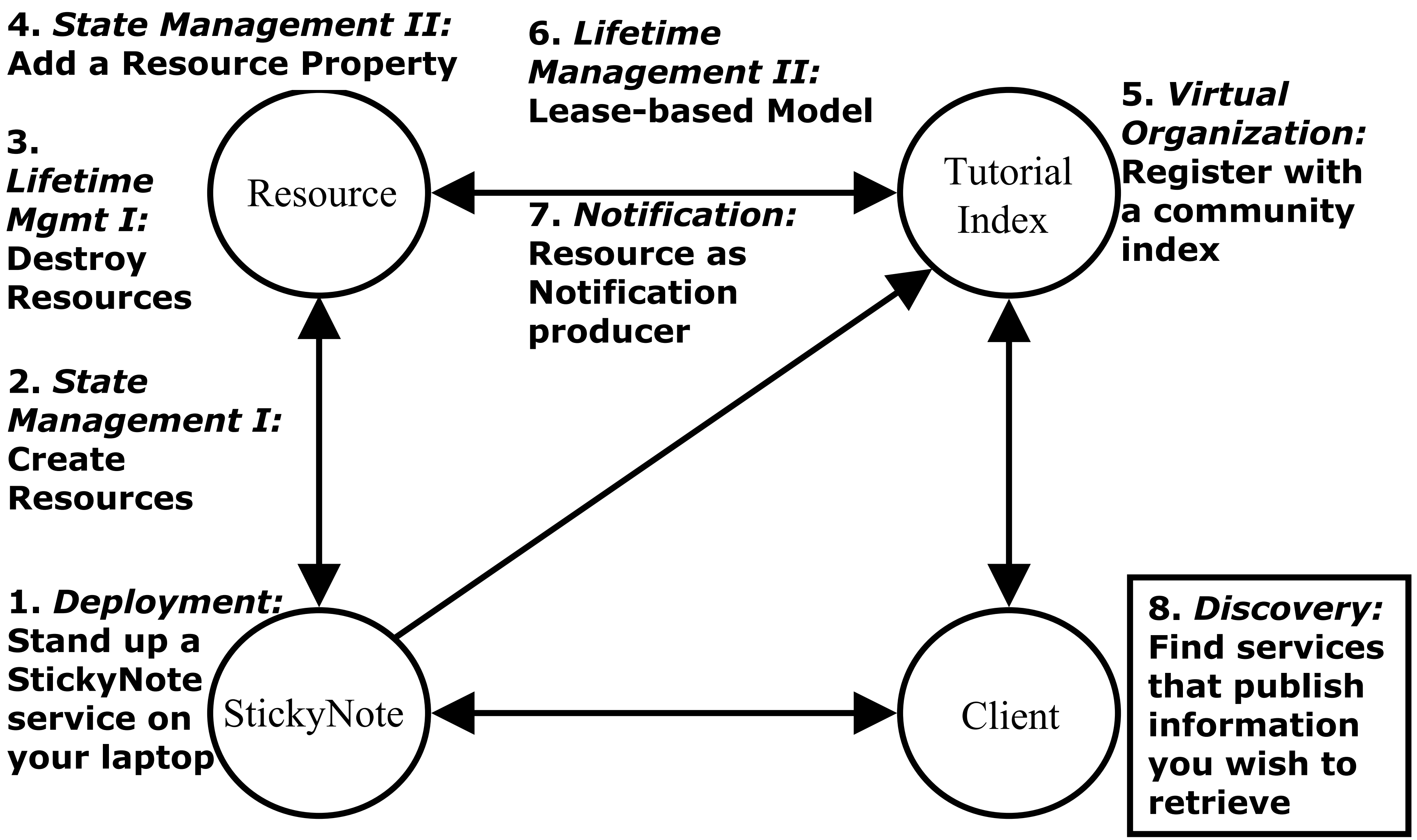
- Notifications can deliver updates of RP changes
- The service needs to support notification
- The client starts another container to receive the notifications
- Notification delivery and order is not guaranteed

How to Build a Service Using GT4

- Overview of Services and GT4
- Build a Service
  - 1. Getting Started: Deploy a Service
  - 2. State Management Part I: Create Resources
  - 3. Lifetime Management Part I: Destroy Resources
  - 4. State Management Part II: Add a Resource Property
  - 5. Building a VO: Register with a Community Index
  - 6. Lifetime Management Part II: Lease-based Model
  - 7. Notification: Resource as Notification Producer
  - **8. Discovery: Find a Resource**
  - 9. Security Part I: Service-Level
  - 10. Security Part II (optional): Resource-Level
- Overview of Tools for GT4 Service Developers
- Tutorial Summary
- Ideas for Further Work

the globus toolkit®
www.globustoolkit.org

Exercise 8: Discovery


4. *State Management II:* Add a Resource Property
6. *Lifetime Management II:* Lease-based Model
3. *Lifetime Mgmt I:* Destroy Resources
Resource
Tutorial Index
5. *Virtual Organization:* Register with a community index
7. *Notification:* Resource as Notification producer
2. *State Management I:* Create Resources
1. *Deployment:* Stand up a StickyNote service on your laptop
StickyNote
Client
8. *Discovery:* Find services that publish information you wish to retrieve

What is Discovery?

- We want to find a resource that has some property
- In our case, a person wants to find a note that has particular content, like “hello.”
- The VO Index knows about of all the notes and their messages
- We can search the VO Index for the notes that contain the word “hello.”
  - As opposed to using EPRs
- The identification of the resource(s) meeting our criteria is called Discovery

Implementation Details

- The VO Index publishes information on every StickyNote Resource that it knows about
- We can dump the entire content of that data using wsrf-query
- However, our Index will contain a large amount of data; it will be necessary to search inside of the output to find the data we need

Searching Inside Resource Properties

- To search the data we can use an XPath query against the Resource Property Set
- An XPath overview in two bullets:
  - XPath is a convenient query language for searching XML documents
  - XPath queries are formed by identifying a route to the desired data

We shall provide you with an XPath query to search the RP of the Tutorial Index…

Searching the Entries of the Tutorial Index

- We can find each note that contains the word ‘hello.’ by sending the following XPath query to the Tutorial Index:

//*[local-name()=‘Entry’][.//*[contains(.,”hello.”)]]

A human translation of this syntax:

“Select all the entries that contain the string ‘hello.’”

What Attendees Should Do

- Create a note containing your name
- Use the search-note client to perform an XPath query against the tutorial Index to retrieve the note containing their own name
- Use search-note to find a note containing your neighbor’s name

What Attendees Should See

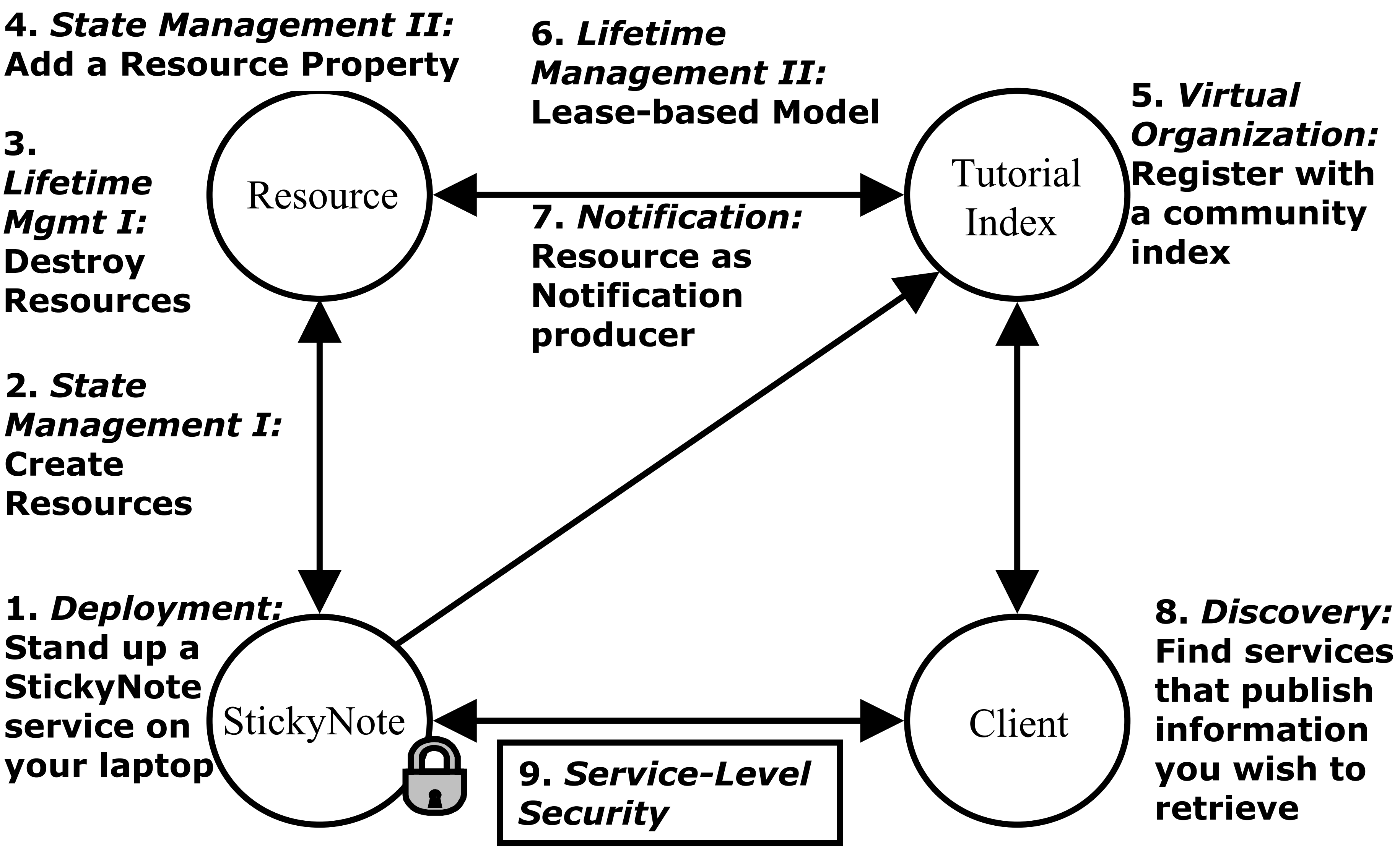
- $ search-note -s http://localhost:8080/wsrf/services/DefaultIndexService 'ell'
- Note 0: <ns1:message xmlns:ns1="http://tutorial.globus.org/stickynote">hello.</ns1:message>
- Note 1: <ns1:message xmlns:ns1="http://tutorial.globus.org/stickynote">hello.</ns1:message>

Exercise 8 Review

- The Index provides a way to discover services based on Resource Properties
- Because RPs are XML, GT4 allows XPath queries for searching and retrieving them

How to Build a Service Using GT4

Exercise 9: Service-Level Security



4. *State Management II:* Add a Resource Property
6. *Lifetime Management II:* Lease-based Model
3. *Lifetime Mgmt I:* Destroy Resources
Resource
Tutorial Index
5. *Virtual Organization:* Register with a community index
7. *Notification:* Resource as Notification producer
2. *State Management I:* Create Resources
1. *Deployment:* Stand up a StickyNote service on your laptop
StickyNote
Client
8. *Discovery:* Find services that publish information you wish to retrieve
9. *Service-Level Security*

GT4 Security Details

- Built on top of PKI
  - ◆ Each entity has two keys: public and private
  - ◆ Data encrypted with one key can only be decrypted with other
  - ◆ The private key is known only to the entity
- The public key is given to the world encapsulated in a X.509 certificate

Certificates

- A X.509 certificate binds a public key to a name
- It includes a name and a public key bundled together and signed by a trusted party (Certificate Authority)
- An example of a Distinguished Name (DN):

  "/O=Tutorial/OU=User/CN=Charles Bacon"

Certificate Authorities

- A Certificate Authority (CA) signs certificate requests
- To verify a certificate signature, you must have a copy of the CA certificate
- In GT land
  - By default, stored in /etc/grid-security/certificates
  - For our tutorial, stored in $GLOBUS_LOCATION/share/certificates

Proxy Certificates

- Proxy certificates contain a new key pair, and are signed by the original certificate
  - Typically has shorter lifetime
  - By default stored in /tmp/x509up_u$UID
  - Protected by file system permissions
- Create a proxy using grid-proxy-init
  - Full GT4 install includes C command line client

Globus Certificate Service

- Globus Certificate Service
  - Web interface to get certificates
  - A certificate authority, but not quite
  - Low-quality certificates
- http://gcs.globus.org:8080/gcs/index.html
- Need CA certificate, already stored in $GLOBUS_LOCATION/share/certificates for the tutorial.

Certificates for tutorial

- Generate certificate to be signed:
  - $GLOBUS_LOCATION/etc/grid-cert-request –caEmail false
- Generates key (userkey.pem) and request (usercert_request.pem)
- Upload/Paste request in http://gcs.globus.org:8080/gcs/usercert.html
- Store returned certificate as usercert.pem in same directory as userkey.pem
- To generate proxy
  - $GLOBUS_LOCATION/bin/visual-grid-proxy-init

Authentication & Authorization

- Authentication
  - Establishing identity of an entity is what it is claimed to be
- Authorization
  - Ensuring an entity is allowed to access resource

GT4 Authentication Support

- Three types of authentication
  - Transport Level (HTTPS)
  - Message Level
    - Secure message
    - Secure conversation
- Signature and encryption
- Anonymous Clients

Server-side Security

- Security descriptors used to set security parameters
- Three levels of granularity:
  - Container level settings
    - Container security descriptor in WSDD file
  - Service level settings
    - Service security descriptor in service deployment in WSDD
  - Resource level settings
    - Programmatically set in resource object
- Resource-level setting has highest precedence

Server-side Security: Authentication

- Per-method authentication and run-as settings
- auth-method
  - none: no authentication
  - SecureMessage: GSI Secure Message
  - SecureConversation: GSI Secure Conversation
  - SecureTransport: GSI Transport
- run-as
  - caller: Execute method with caller's credential
  - system: Execute method with container credential
  - service: Execute method with service credential
  - resource: Execute method with resource credential

Server-side Security: Authorization

- Service-wide/resource-wide authorization settings
- A few built-in types:
  - None: no authorization of the client will be performed
  - Self: a client will be authorized if it has the same identity as the service/resource/container
  - Identity: a client will be authorized if it matches the specified by the service/resource/container
  - GridMap: a client will be authorized if its identity is listed in a gridmap file.
- Custom methods can be plugged in
  - For example, perform per-method checks
- Chainable

Client-side Security

- Security parameters set programmatically on the Stub (or Call object)
  - As individual properties
  - Using client security descriptor

- For example, the create-note client will set the protection mode of Secure Transport to encryption.

```
((Stub)portType)._setProperty(
                    GSIConstants.GSI_TRANSPORT,
                    Constants.ENCRYPTION);
```

Client-side Security: Authorization

- A few built-in types:
  - None, Self, Identity (same as on server)
  - Host: service/resource will be authorized if the host returns an identity containing the hostname

Client authorization is independent and separate from server authorization!

What Attendees Should Do

- Uncomment the “securityDescriptor” parameter in WSDD
- Edit grid-mapfile
  - Add identity of another person
  - Have them create/edit note on your service
- Create a proxy
- Create a note and write to it

What Attendees Should See

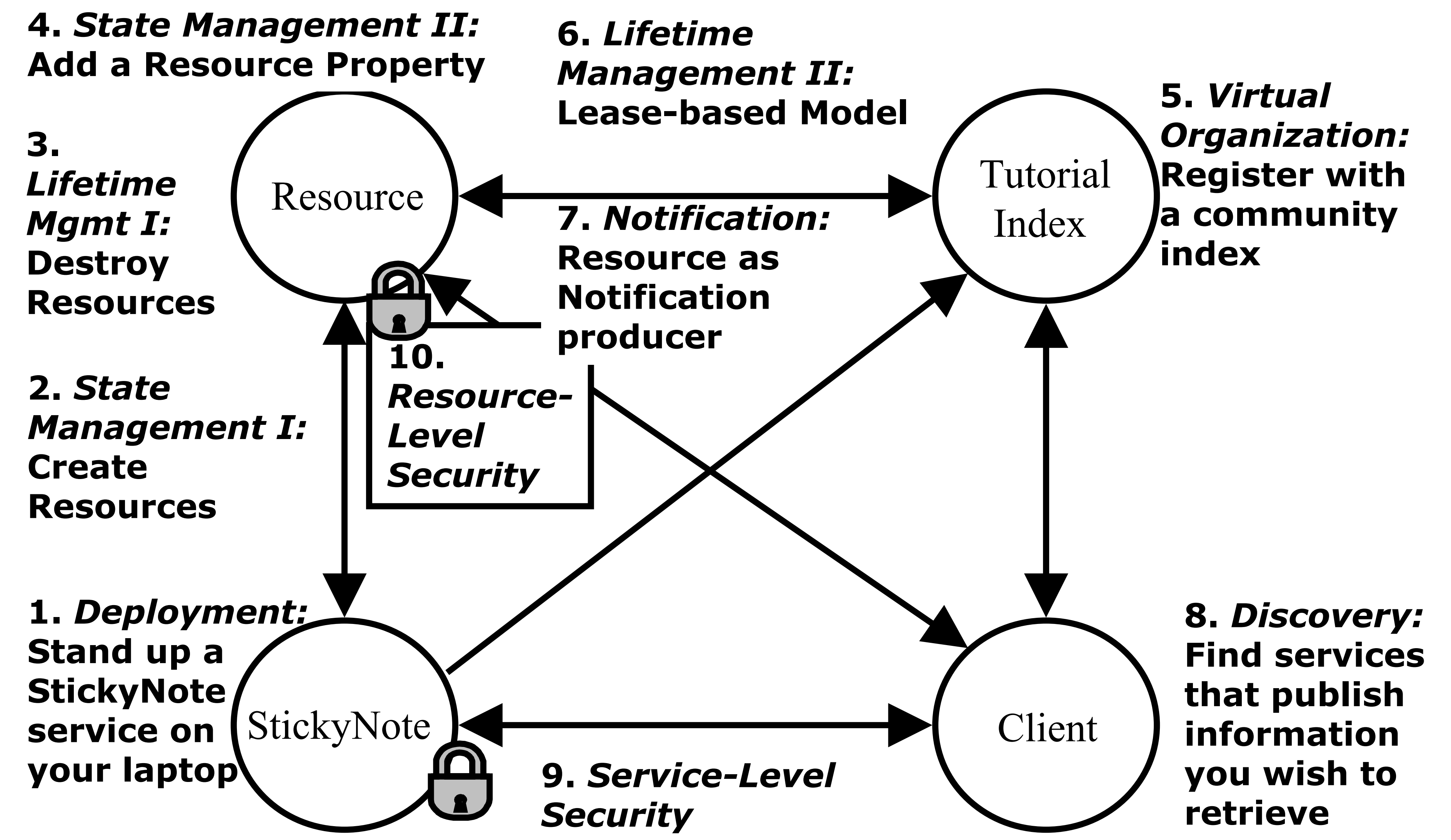
- Without proxy, operations should fail
- With valid credentials,
  - If you are not in gridmap, you cannot create or write notes
  - Read note (show-note) is not secure and should be allowed even if not in gridmap

Exercise 9 Review

- Container and Service security is configured declaratively using security descriptor file
- Resource and Client security is configured programmatically by setting properties in the code
- Service-side authentication may be specified on a per-operation basis
- Custom service-side authorization may be plugged in and chained with other schemes

How to Build a Service Using GT4

Exercise 10: Resource-Level Security


4. *State Management II:* Add a Resource Property
6. *Lifetime Management II:* Lease-based Model
3. *Lifetime Mgmt I:* Destroy Resources
Resource
Tutorial Index
5. *Virtual Organization:* Register with a community index
7. *Notification:* Resource as Notification producer
10. *Resource-Level Security*
2. *State Management I:* Create Resources
1. *Deployment:* Stand up a StickyNote service on your laptop
StickyNote
Client
8. *Discovery:* Find services that publish information you wish to retrieve
9. *Service-Level Security*

Resource-Level Security

- This is an optional exercise
- There is no lecture for this chapter
- The lesson is contained in the student notes

How to Build a Service Using GT4

- Overview of Services and GT4
- Build a Service
  - 1. Getting Started: Deploy a Service
  - 2. State Management Part I: Create Resources
  - 3. Lifetime Management Part I: Destroy Resources
  - 4. State Management Part II: Add a Resource Property
  - 5. Building a VO: Register with a Community Index
  - 6. Lifetime Management Part II: Lease-based Model
  - 7. Notification: Resource as Notification Producer
  - 8. Discovery: Find a Resource
  - 9. Security Part I: Service-Level
  - 10. Security Part II (optional): Resource-Level
- **Overview of Tools for GT4 Service Developers**
- Tutorial Summary
- Ideas for Further Work

Tools

- WSDLPP
- Binding Generator
- Stub Generator (Axis)
- Launcher Generator

Tools: WSDLPP

- WSDL Preprocessor
- Custom attribute specifies PortType composition
- Avoids copy-and-paste errors
- Handles Resource Properties merging

```
<ant antfile="share/globus_wsrf_tools/build-stubs.xml" target="flatten">
        <property name="source.flatten.dir"
                location="core/samples/counter"/>
        <property name="target.flatten.dir"
                location="core/samples/counter"/>
        <property name="wsdl.source"
                   value="counter_port_type.wsdl"/>
        <property name="wsdl.target"
                   value="counter_flattened.wsdl"/>
        <property name="wsdl.porttype"
                   value="CounterPortType"/>
</ant>
```

Tools: WSDLPP

Before:

```
<portType name="CounterPortType"
      wsdlpp:extends="wsrpw:GetResourceProperty …”>
  <operation name=“add”>
    …
  </operation>
</portType>
```

(In the original, `wsdlpp:extends="wsrpw:GetResourceProperty …”` is set in bold.)

After:

```
<portType name="CounterPortType“>
  <operation name=“add”>
    …
  </operation>
 <operation name=“GetResourceProperty”>
    …
  </operation>

</portType>
```

Tools: Binding Generator

- Generates WSDL Binding and Service parts from WSDL PortType
- SOAP binding with Document/literal
- Handles WS-Addressing action attributes

```
<ant antfile="share/globus_wsrf_tools/build-stubs.xml"
      target="generateBinding">
         <property name="source.binding.dir"
                        value="core/samples/counter"/>
         <property name="target.binding.dir"
                        value="core/samples/counter"/>
         <property name="porttype.wsdl"
                        value="counter_flattened.wsdl"/>
         <property name="binding.root"
                        value="counter"/>
</ant>
```

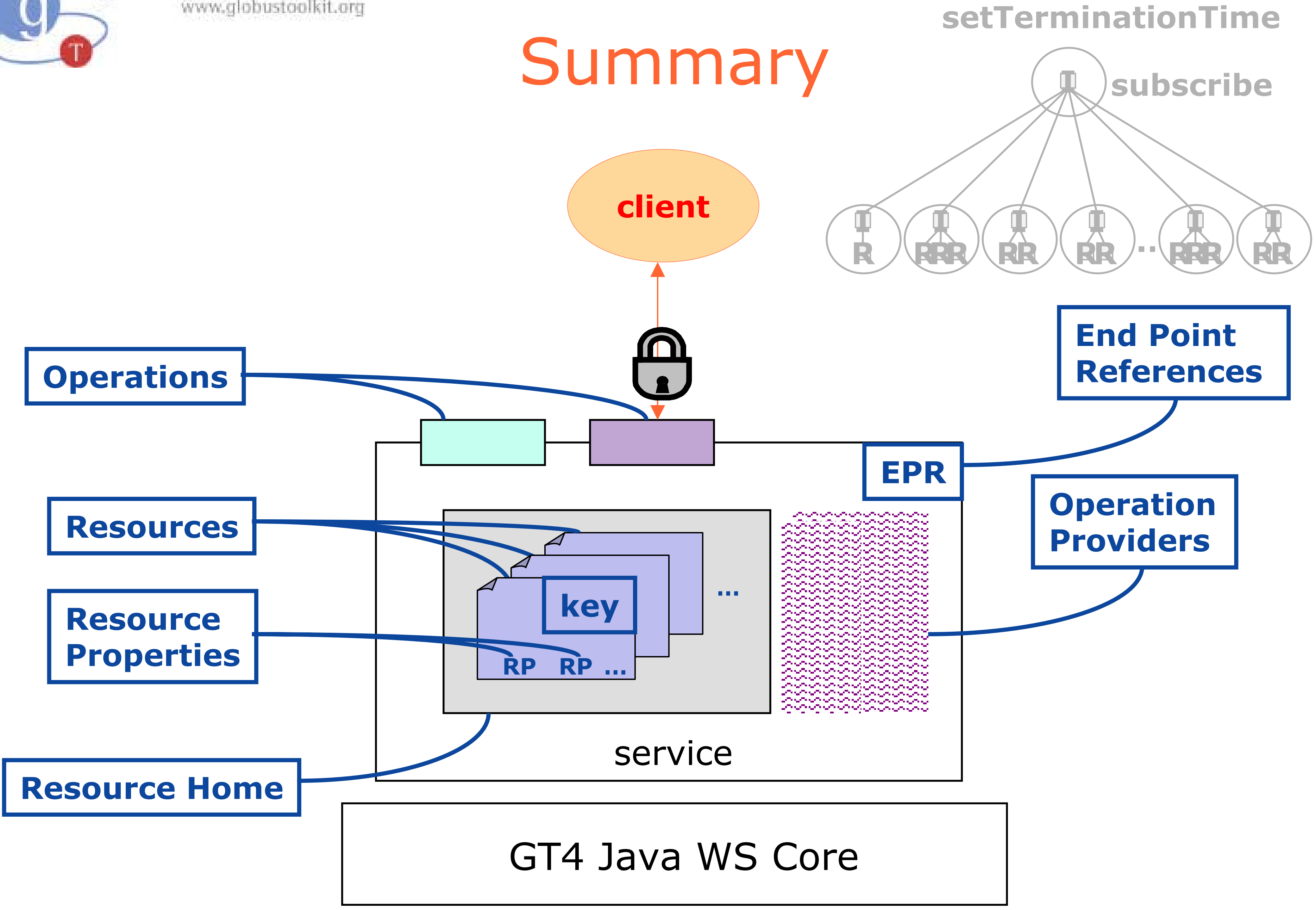
Tools: Launcher Generator

- Generates Windows batch files and Unix/Linux shell scripts for Java clients
- Classpath, standard environment variables are automatically handled

```
<ant antfile="share/globus_wsrf_common/build-launcher.xml"
      target="generateLauncher">
          <property name="launcher-name"
                        value="grid-proxy-init"/>
          <property name="class.name“
                        value="org.globus.tools.ProxyInit"/>
</ant>
```

How to Build a Service Using GT4

Summary


setTerminationTime
subscribe
R
RRR
RR
RR
RRR
RR
client
End Point References
Operations
EPR
Resources
Operation Providers
key
…
Resource Properties
RP
RP
…
service
Resource Home
GT4 Java WS Core

How to Build a Service Using GT4

- Overview of Services and GT4
- Build a Service
  - 1. Getting Started: Deploy a Service
  - 2. State Management Part I: Create Resources
  - 3. Lifetime Management Part I: Destroy Resources
  - 4. State Management Part II: Add a Resource Property
  - 5. Building a VO: Register with a Community Index
  - 6. Lifetime Management Part II: Lease-based Model
  - 7. Notification: Resource as Notification Producer
  - 8. Discovery: Find a Resource
  - 9. Security Part I: Service-Level
  - 10. Security Part II (optional): Resource-Level
- Overview of Tools for GT4 Service Developers
- Tutorial Summary
- **Ideas for Further Work**

Join the GT4 Community

- Download and use the software, and provide feedback
  - Beta release available at the end of February
  - Join **gt4friends@globus.org** mail list
- Review, critique, add to documentation
  - Globus Doc Project: **http://gdp.globus.org**
- Tell us about your GT4-related tool, service, or application

Further Reading

GT4 draft documentation page:

http://www-unix.globus.org/toolkit/docs/development/4.0-drafts/toc_all.html

Globus Alliance publications page:

http://www.globus.org/research/papers.html

WS-Resource Framework:

http://www.globus.org/wsrf/

On WSDL:

http://www.w3.org/TR/wsdl

The GT4 Fact Sheet:

http://www-unix.globus.org/toolkit/docs/development/4.0-drafts/GT4Facts/index.html

The Globus Documentation Project:

http://gdp.globus.org

STOP