How to Build a Service Using GT4

Globus Alliance Staff

Rachana Ananthakrishnan, Charles Bacon,
Lisa Childers, Jarek Gawor, Joe Insley,
Argonne National Laboratory and the University of Chicago
Ben Clifford, formerly of the USC/Information Sciences Institute

Important Note

- The security chapters that usually appear in this tutorial (chapters 9 and 10) have been removed from this presentation

- Security will be addressed in a separate presentation

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. Discovery: Find a Resource

    - 6. Building a VO: Register with a Community Index

    - 7. Lifetime Management Part II: Lease-based Model

    - 8. Notification: Resource as Notification Producer

- Overview of Tools for GT4 Service Developers

- Tutorial Summary

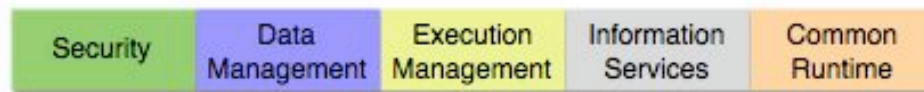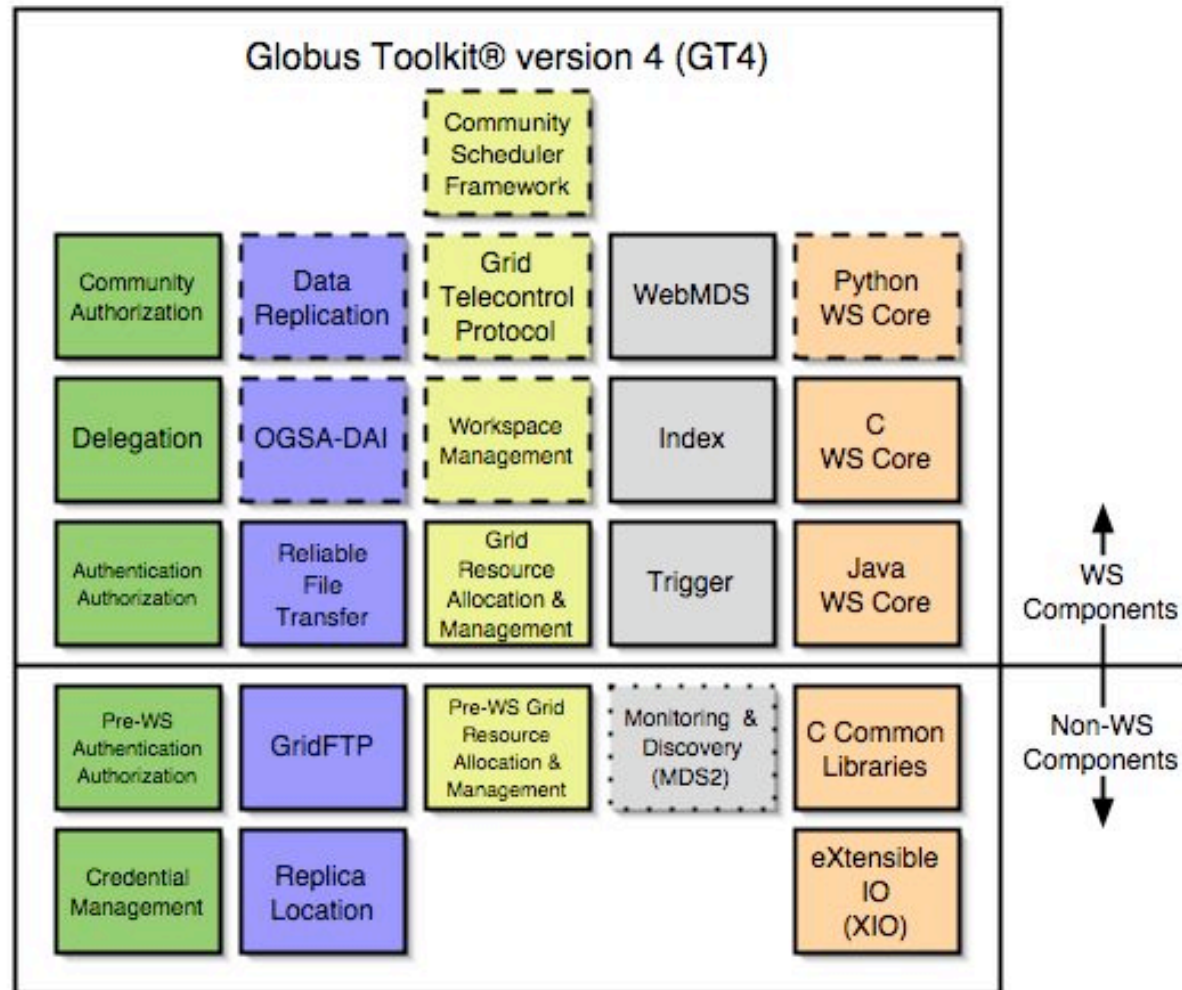- Ideas for Further Work

Very Brief Overview of GT

- The Globus Toolkit (GT) is organized as a collection of loosely-coupled components

- GT components contain:
    - services and clients
    - libraries
    - development tools

- GT components are used to build Grid-based applications and services
    - GT can be viewed as a Grid SDK

- GT components can be categorized across a couple of different dimensions
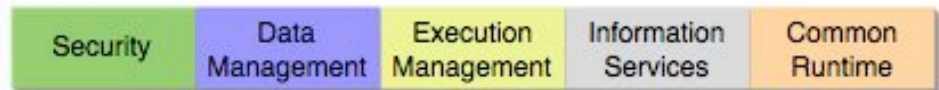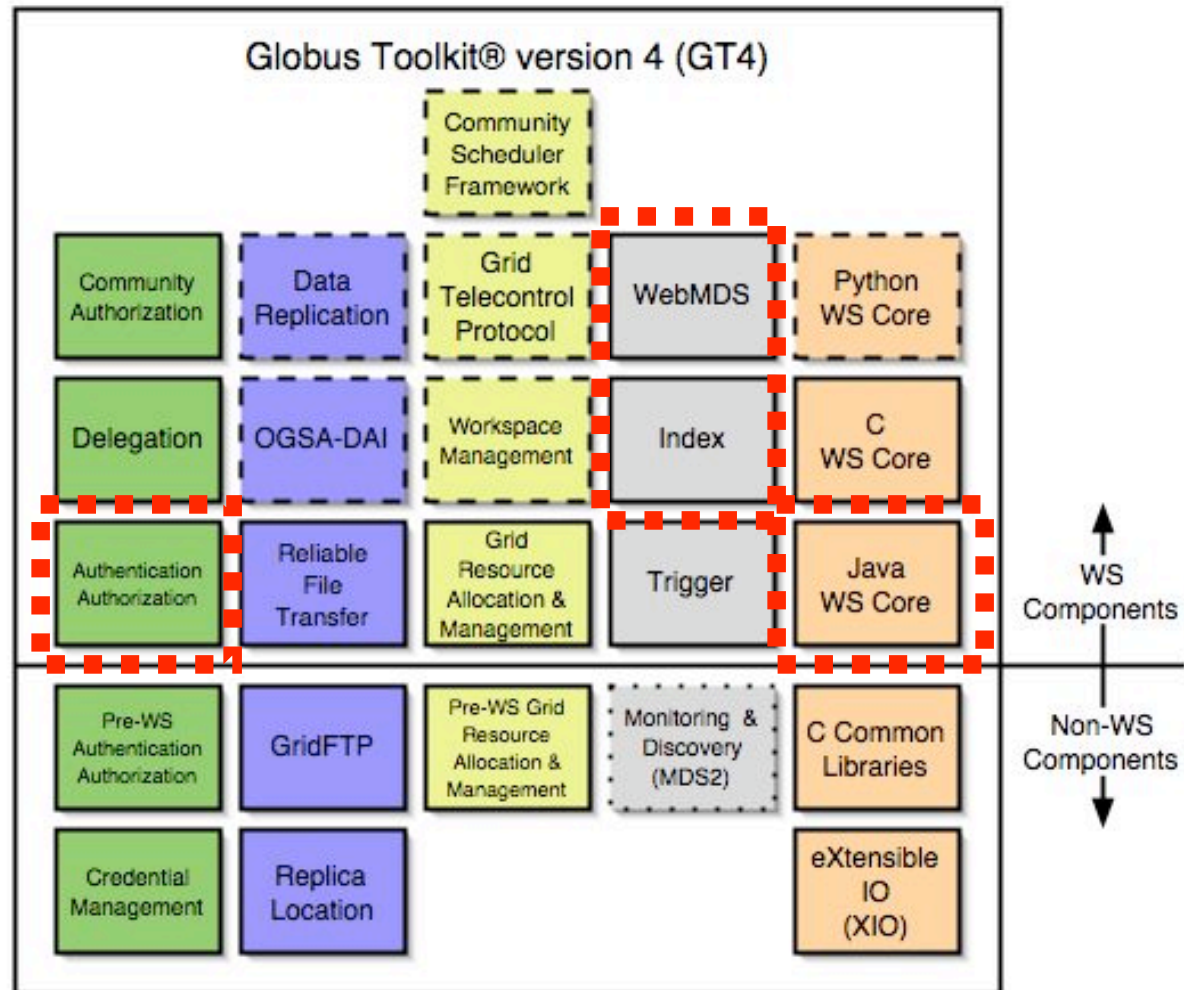   - By broad domain area
    - By protocol support

GT Domain Areas

- Security
  - Apply uniform policy across distinct systems
- Data management
  - Discover, transfer, & access large data
- Execution management
  - Provision, deploy, & manage resources
- Information services
  - Discover & monitor resources
- Common runtime
  - Host existing services and build new services

GT Protocol Support

- Web service protocols
  - WS-I, WSRF, WSN, WS Addressing, WS Security

- Non Web service protocols
  - Standards-based, such as GridFTP
  - Proprietary

Globus Toolkit® version 4 (GT4)

| Security | Data Management | Execution Management | Information Services | Common Runtime |

Legend:
- Core GT Component: public interfaces frozen between incremental releases; best effort support
- Contribution/Tech Preview: public interfaces may change between incremental releases
- Deprecated Component: not supported; will be dropped in a future release

Globus Toolkit® version 4 (GT4)

		Community Scheduler Framework		
Community Authorization	Data Replication	Grid Telecontrol Protocol	WebMDS	Python WS Core
Delegation	OGSA-DAI	Workspace Management	Index	C WS Core
Authentication Authorization	Reliable File Transfer	Grid Resource Allocation & Management	Trigger	Java WS Core
Pre-WS Authentication Authorization	GridFTP	Pre-WS Grid Resource Allocation & Management	Monitoring & Discovery (MDS2)	C Common Libraries
Credential Management	Replica Location			eXtensible IO (XIO)

WS Components

Non-WS Components

Security	Data Management	Execution Management	Information Services	Common Runtime

Core GT Component: public interfaces frozen between incremental releases; best effort support

Contribution/Tech Preview: public interfaces may change between incremental releases

Deprecated Component: not supported; will be dropped in a future release

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. Discovery: Find a Resource
  - 6. Building a VO: Register with a Community Index
  - 7. Lifetime Management Part II: Lease-based Model
  - 8. Notification: Resource as Notification Producer

- Overview of Tools for GT4 Service Developers
- Tutorial Summary
- Ideas for Further Work

Tutorial Structure

- The tutorial is organized as a series of exercises in which increasing functionality is added to a skeletal service implementation

- The exercises demonstrate common Web service interactions using the Globus Toolkit® 4.0

- The demonstration code has trivial functionality in order to maintain focus on the important bits
  - ◆ Teach the mechanics of writing a GT4 service
  - ◆ Highlight fundamental Grid computing interactions and patterns

- Each chapter includes:
  - ◆ A discussion of the concepts behind the exercise
  - ◆ A discussion of implementation details
  - ◆ Hands-on exercises
 - ◆ A summary

Tutorial Materials

- This slideset
- A GT distribution
- Demonstration code
- Exercise notes
- A community index service
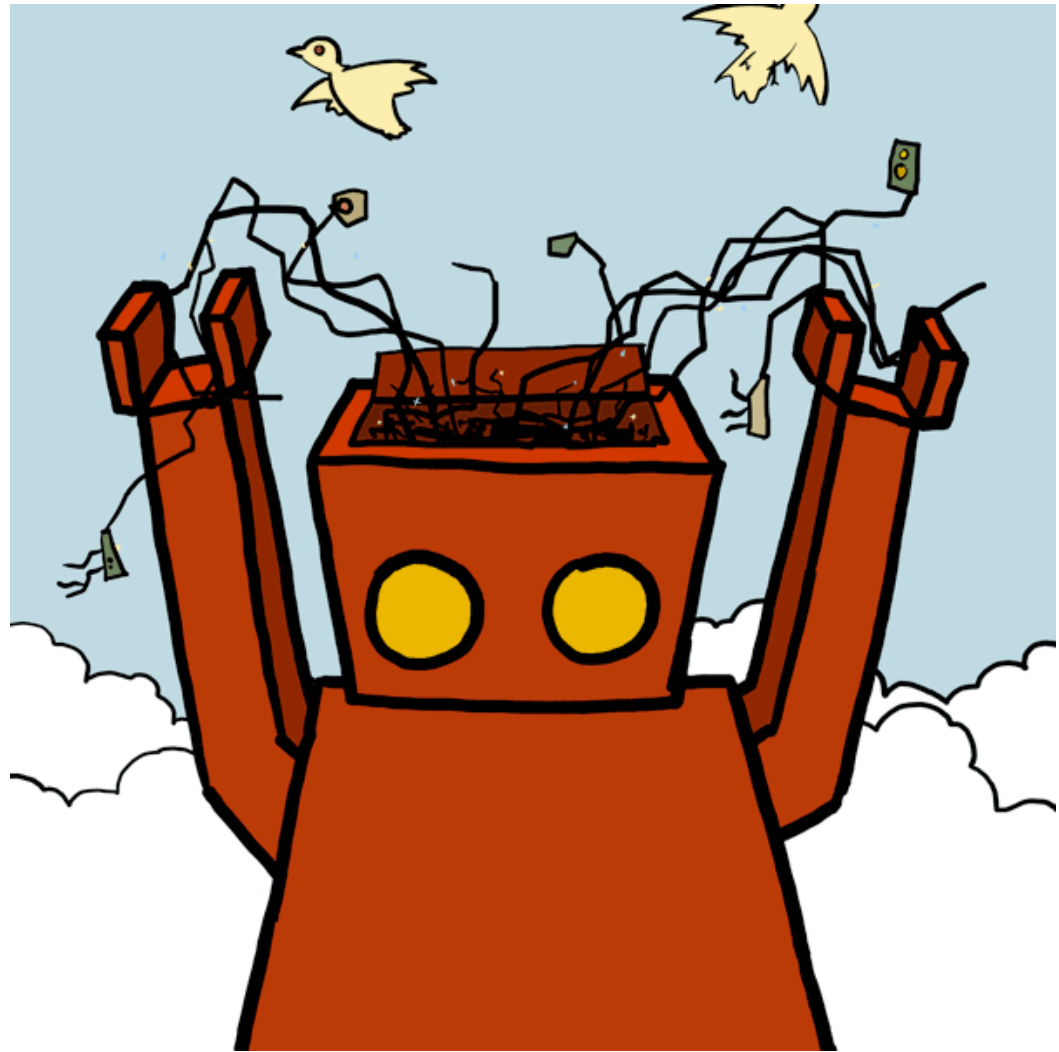- Two visualizers for the community index's data

The oracle

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. Discovery: Find a Resource
    - 6. Building a VO: Register with a Community Index
    - 7. Lifetime Management Part II: Lease-based Model
    - 8. Notification: Resource as Notification Producer

- Overview of Tools for GT4 Service Developers
- Tutorial Summary
- Ideas for Further Work

Exercise 1: Deploy a Service

StickyNote ← → Client

**1. *Deployment:*
Stand up a StickyNote service**
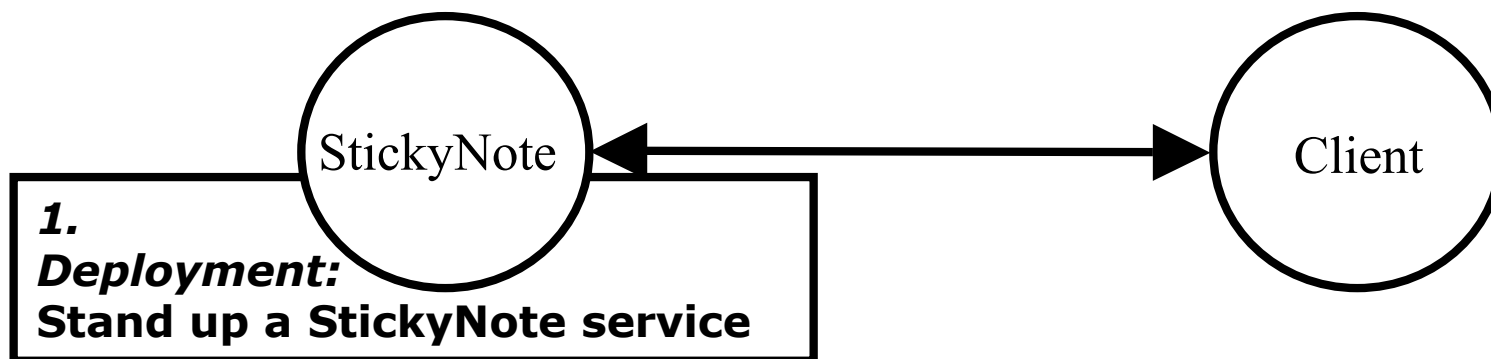
Concepts

The StickyNote Service

- Today we will work with demonstration code that writes and shows a text message

- The demonstration code is a web service called StickyNote

- StickyNote, like all web services, executes inside a hosting environment (also called a container)

- Today we will use the Java container that is included in GT4

The StickyNote Service

write-note client

show-note client

StickyNote service

GT4 Java WS Core

Implementation details

Software Requirements

- jdk
  - The Java software development kit
- jakarta ant
  - Open source tool for building Java code (similar to "make")
- a GT4 binary installation
  - A subset of the Globus Toolkit: the wsrf and ws-mds modules taken from the globus_4_0_branch in CVS
- StickyNote code
  - Demonstration code written specifically for this tutorial

Hands-on Exercises

Student Notes, Chapter 1

Exercise 1 Review

- Grid services execute inside a hosting environment
  - Services are deployed into a container
  - Deployed services start-up when the hosting environment is started
- GT4 includes a runtime environment for hosting Java services
- Service capabilities are invoked by clients
  - show-note, write-note
- Clients do not need to be written in the same language, nor run on the same platform as the services

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. Discovery: Find a Resource
  - 6. Building a VO: Register with a Community Index
  - 7. Lifetime Management Part II: Lease-based Model
  - 8. Notification: Resource as Notification Producer
- Overview of Tools for GT4 Service Developers
- Tutorial Summary
- Ideas for Further Work

Exercise 2: State Management I

Resource

StickyNote

Client

2. *State Management I:* Create Resources
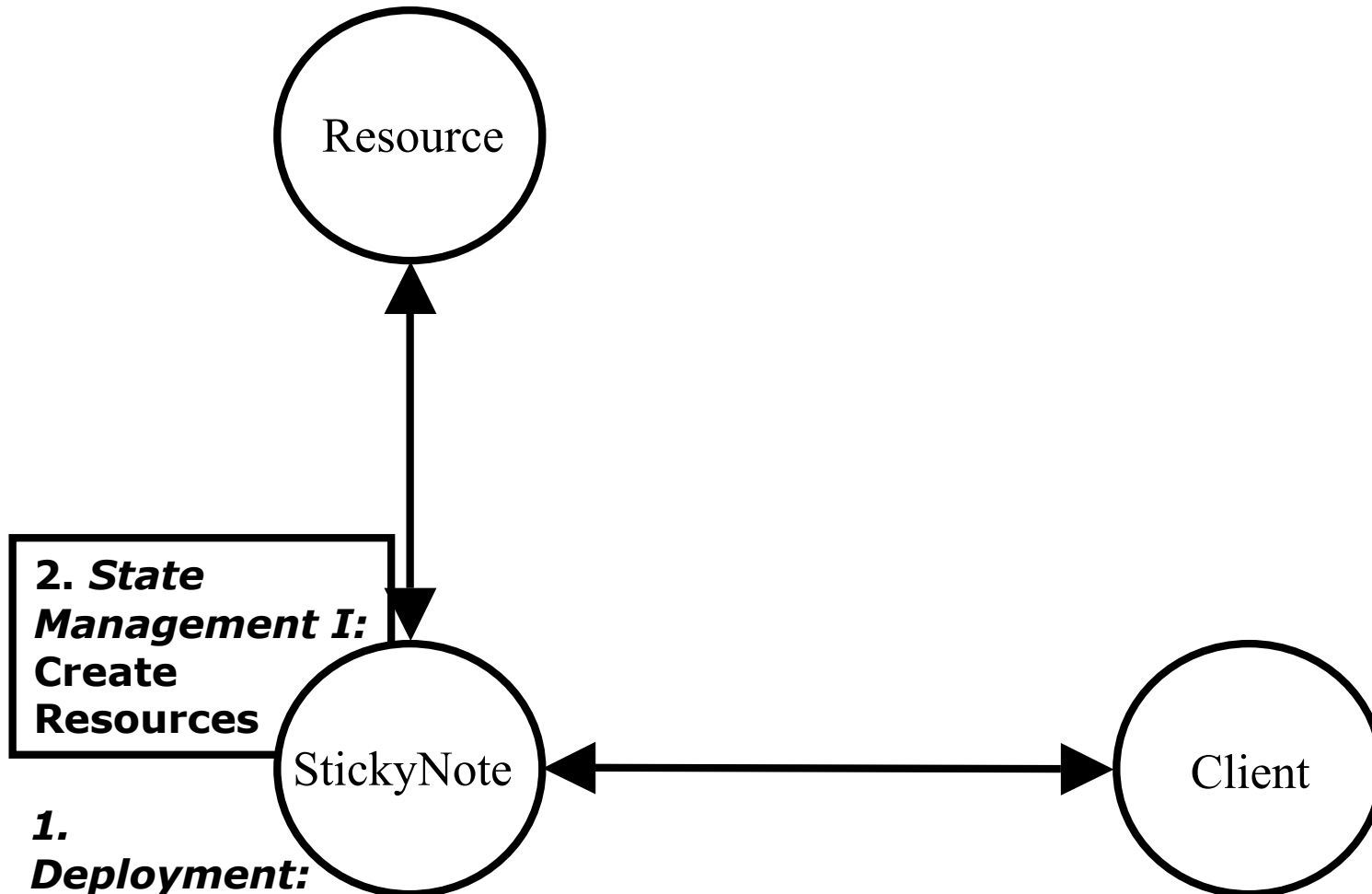
***1. Deployment:* Stand up a StickyNote service**
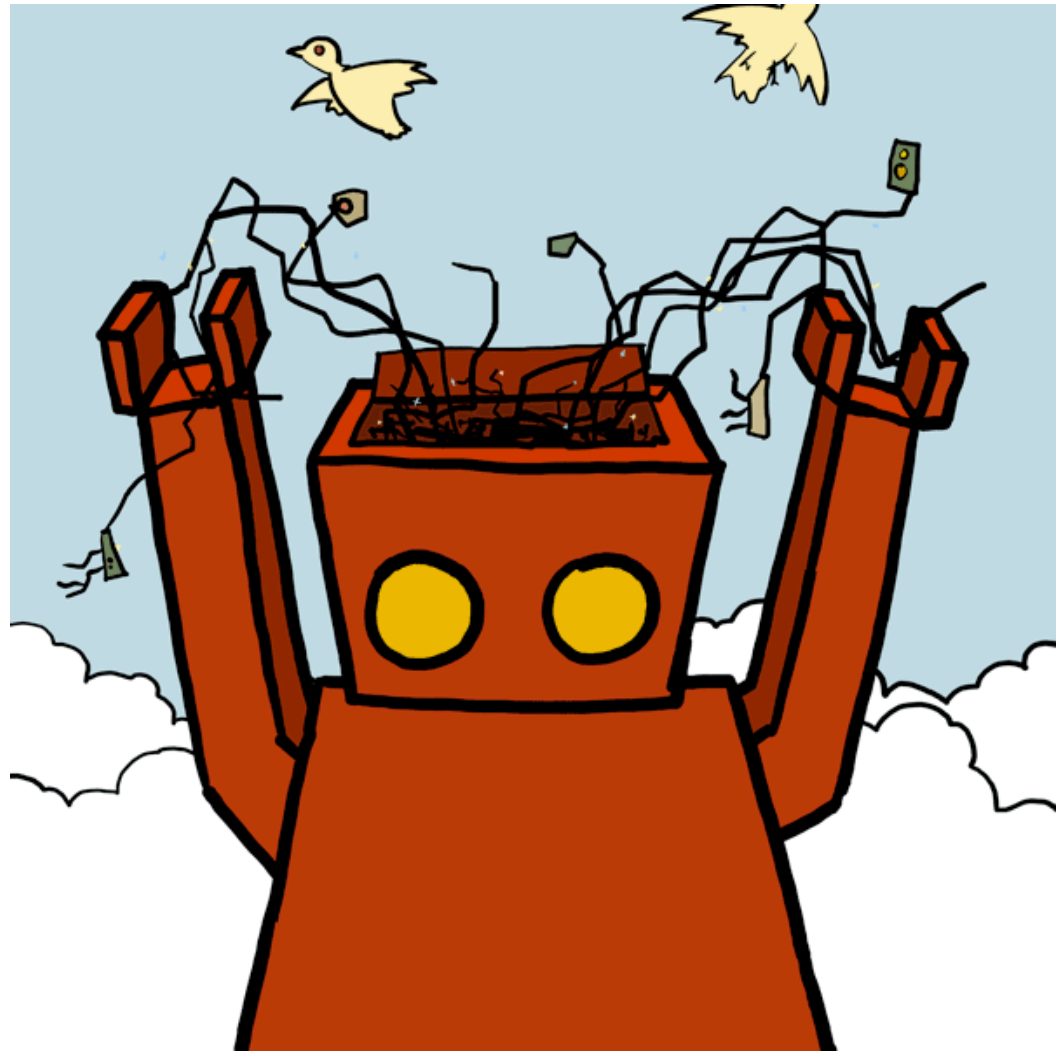
Concepts

Service Basics

Now that we've deployed and interacted with the barebones demonstration service we'll begin to modify it in ways that highlight some key Grid computing concepts

We begin with explanations of some GT4 service basics

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

write-note   show-note

Operations

StickyNote service

GT4 Java WS Core
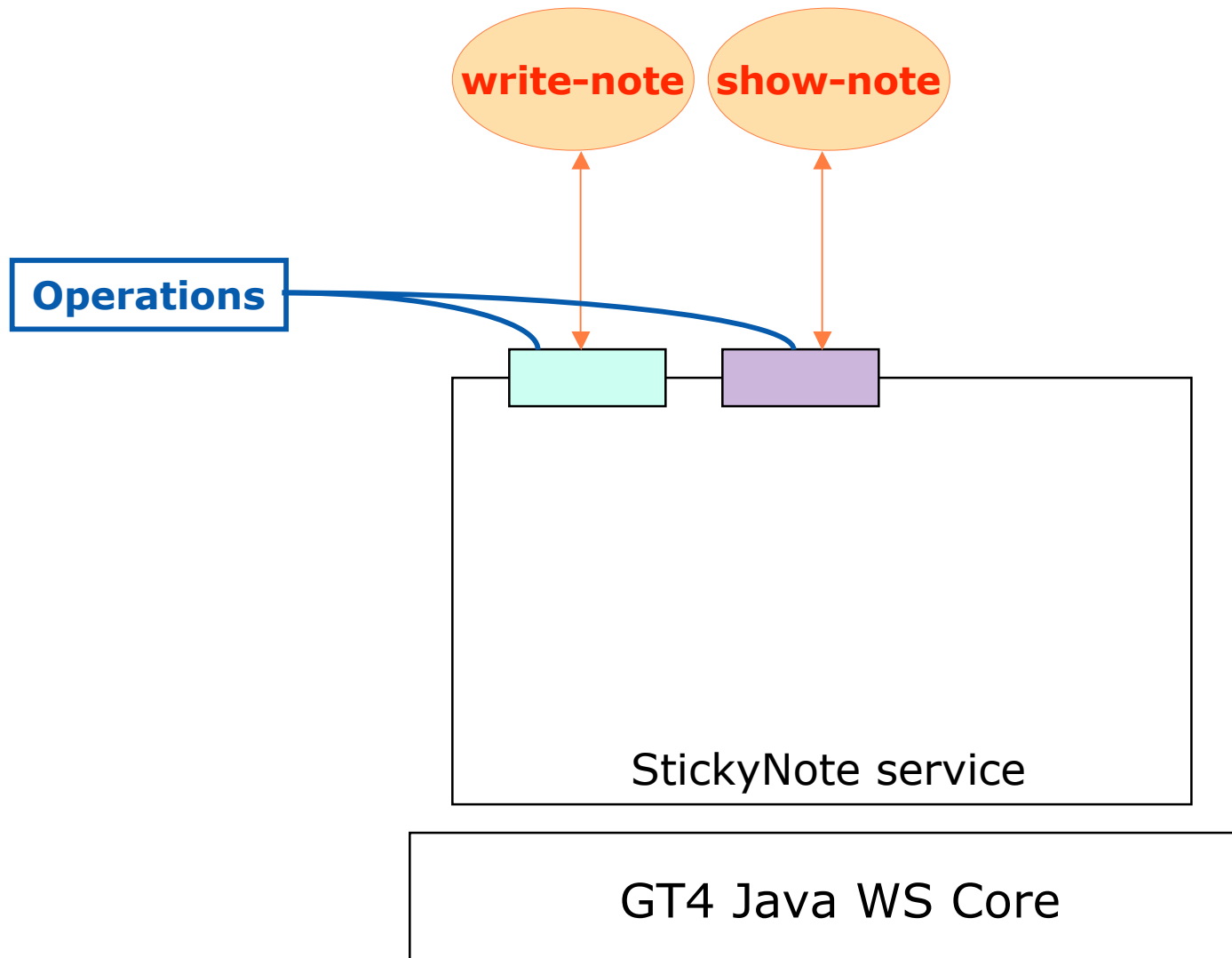
Service Basics: Resources

- In addition to operations, GT4 Web services can provide access to state

  - state is service-related data that persists across client invocations

- The state of our StickyNote service is note content

Service Basics: Resources

- In the GT4 Java service programming model, state is stored in a "Resource" class

- The StickyNote service currently contains only a single Resource

- In this chapter we will learn how to add support for multiple notes

Note that GT4 does not mandate the use of state. Developers can also write and host stateless services using GT4

Service Basics: Resources

Operations

Resource

hello.

StickyNote service

GT4 Java WS Core
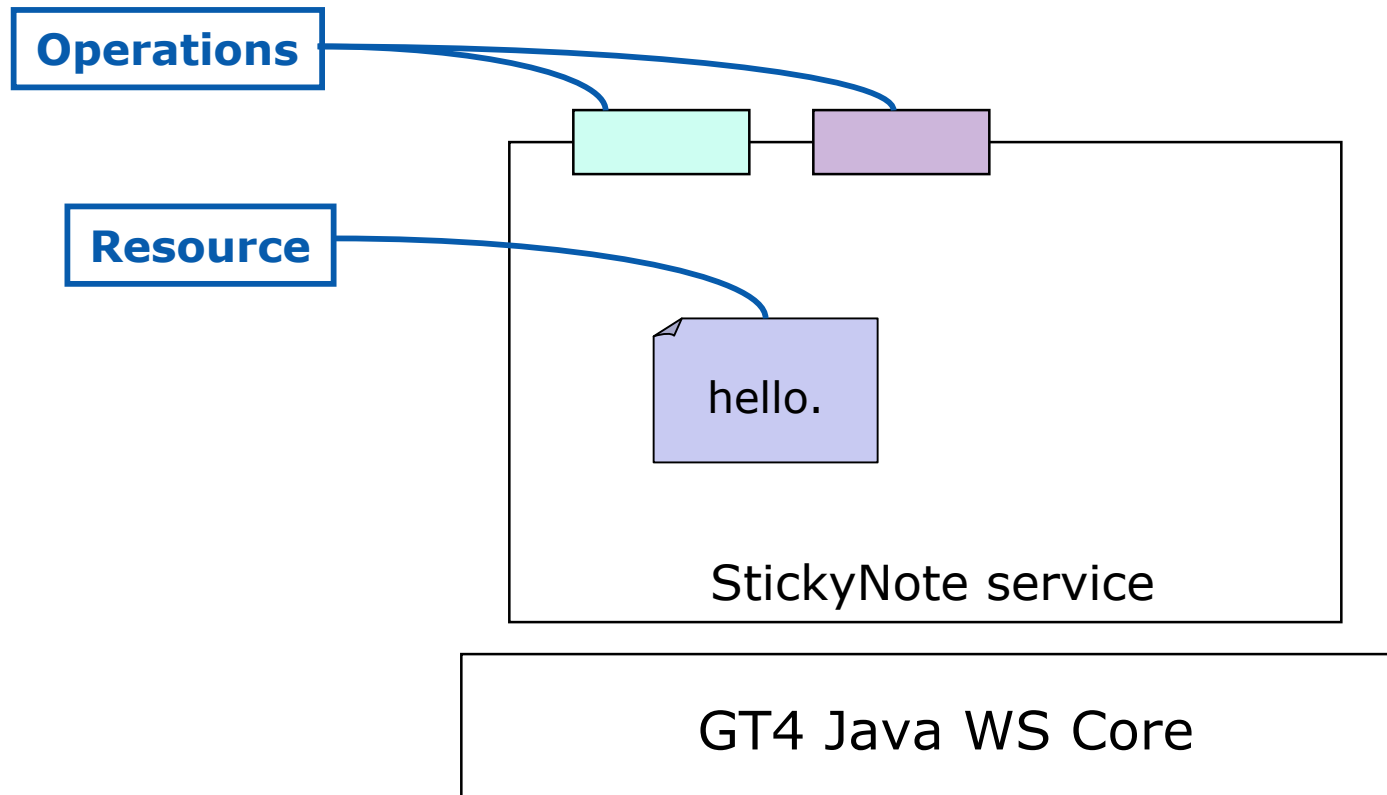
Service Basics: Resource Homes

- **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 …**

  a Resource Home that supports a single Resource

  to …

 a Resource Home that can hold multiple Resources

Service Basics: Resource Homes

Operations

Resources

ResourceHome

hello.

service

GT4 Java WS Core
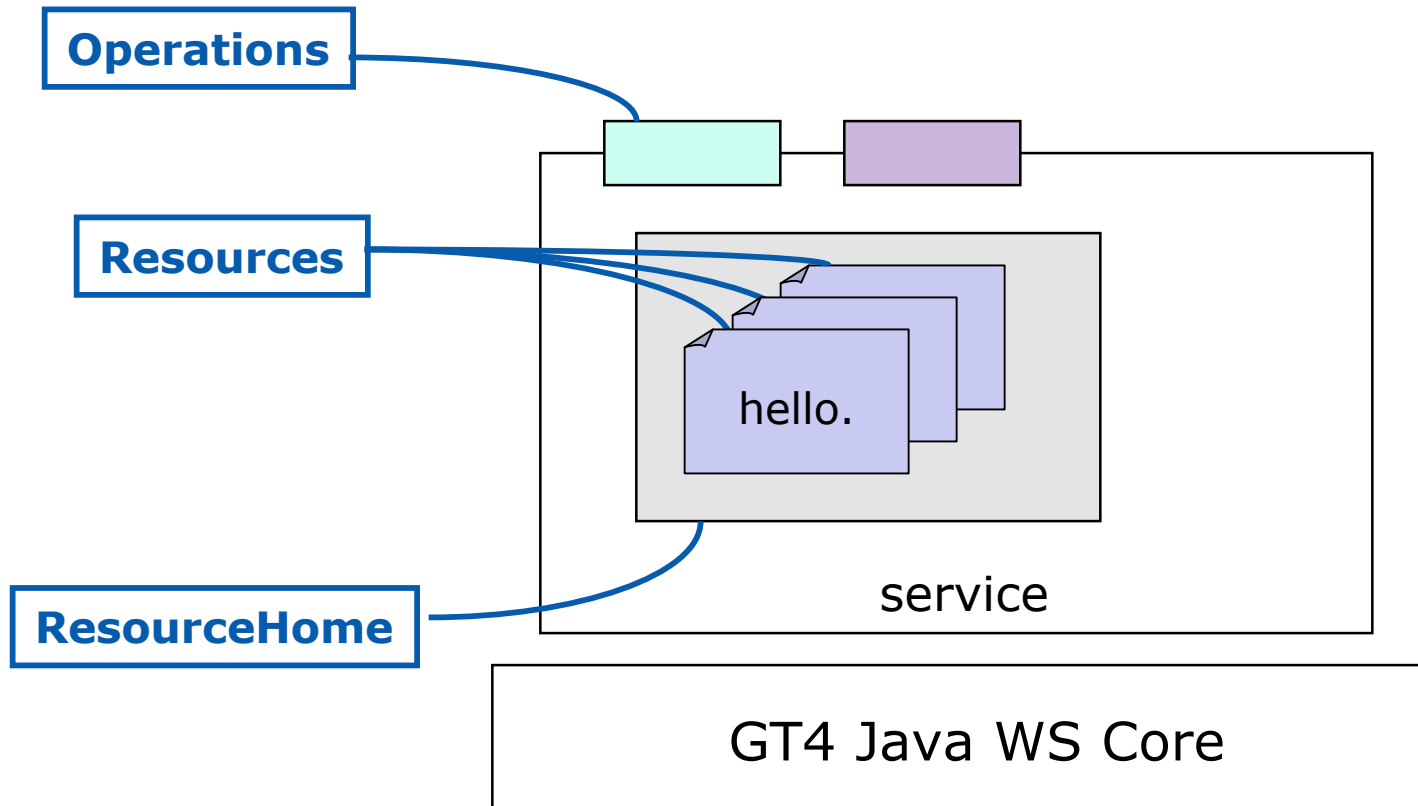
Sidebar: GT4 Resources and the Factory Pattern

- A *factory* is a well-accepted software design pattern

- Factories encapsulate rules for the dynamic creation of entities

Let's look briefly at two example approaches for implementing factories using GT4…

# Sidebar: GT4 Resources and the Factory Pattern

Factory approach #1: Single factory service with a Resource Home supporting multiple resources

Instance Resources

Resource home supporting multiples
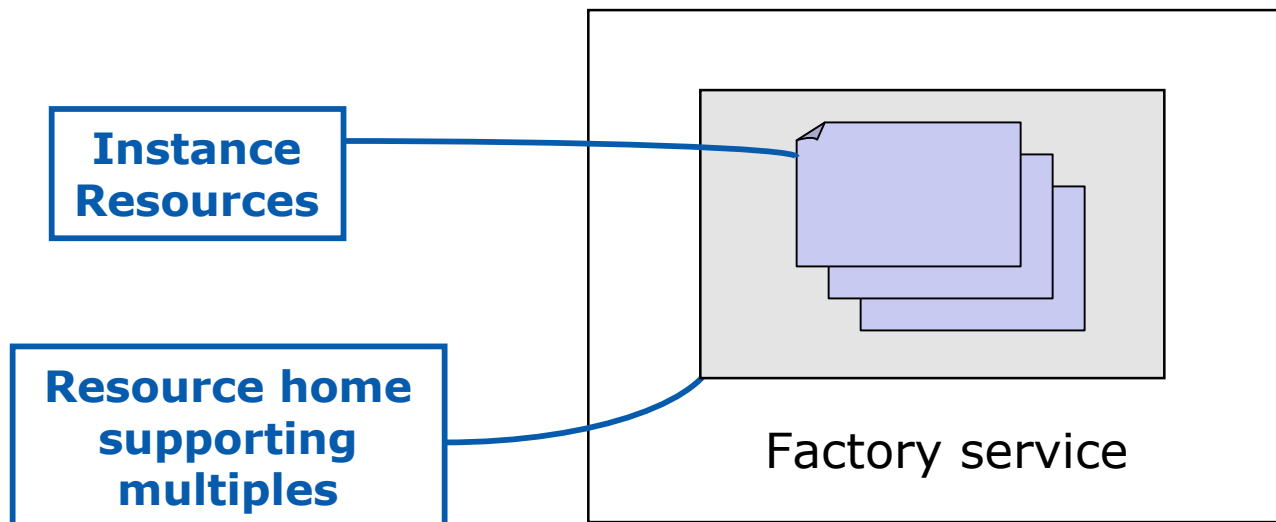
Factory service

Sidebar: GT4 Resources and the Factory Pattern

Factory approach #2: Factory service containing a singleton resource home and a companion instance service

Factory service

Instance service

Singleton resource home

Resource containing state applicable to all instances

Instance Resources
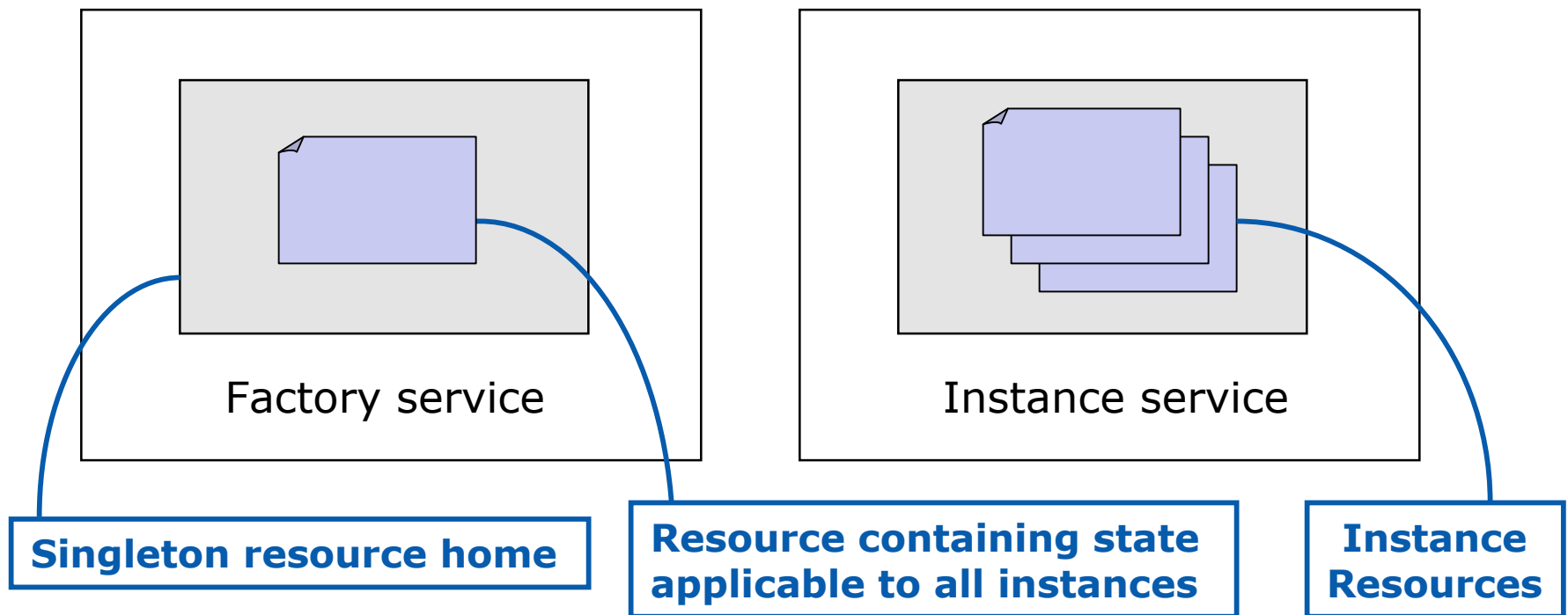
Implementation details

Types of Files

- Interface description files: *.wsdl
  - ◆ Defines the public interface for the service
- Service and client source files: *.java
  - ◆ Contain the implementation of the interface
- Build instructions for ant: build.xml
  - ◆ Like a makefile
- Service deployment instructions: *.wsdd & jndi
  - ◆ Specifies container-level service configuration information
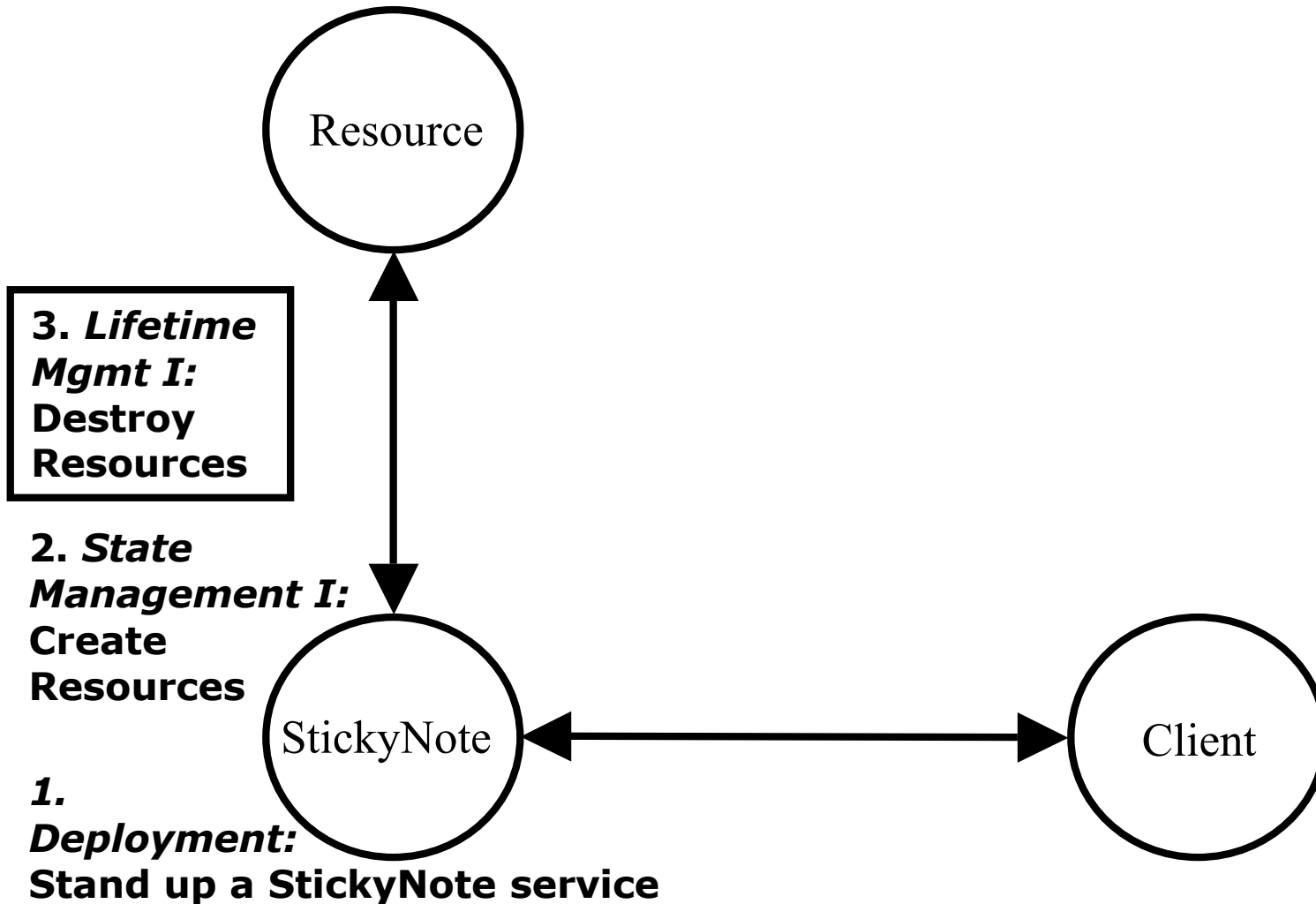
Hands-on Exercises

Student Notes, Chapter 2

Exercise 2 Review

- Services have operations

- Services can have stateful components, called Resources

- The multiple resources contain different state, but are accessed via the same service

- The WSDL contains definitions of types, messages, and operations

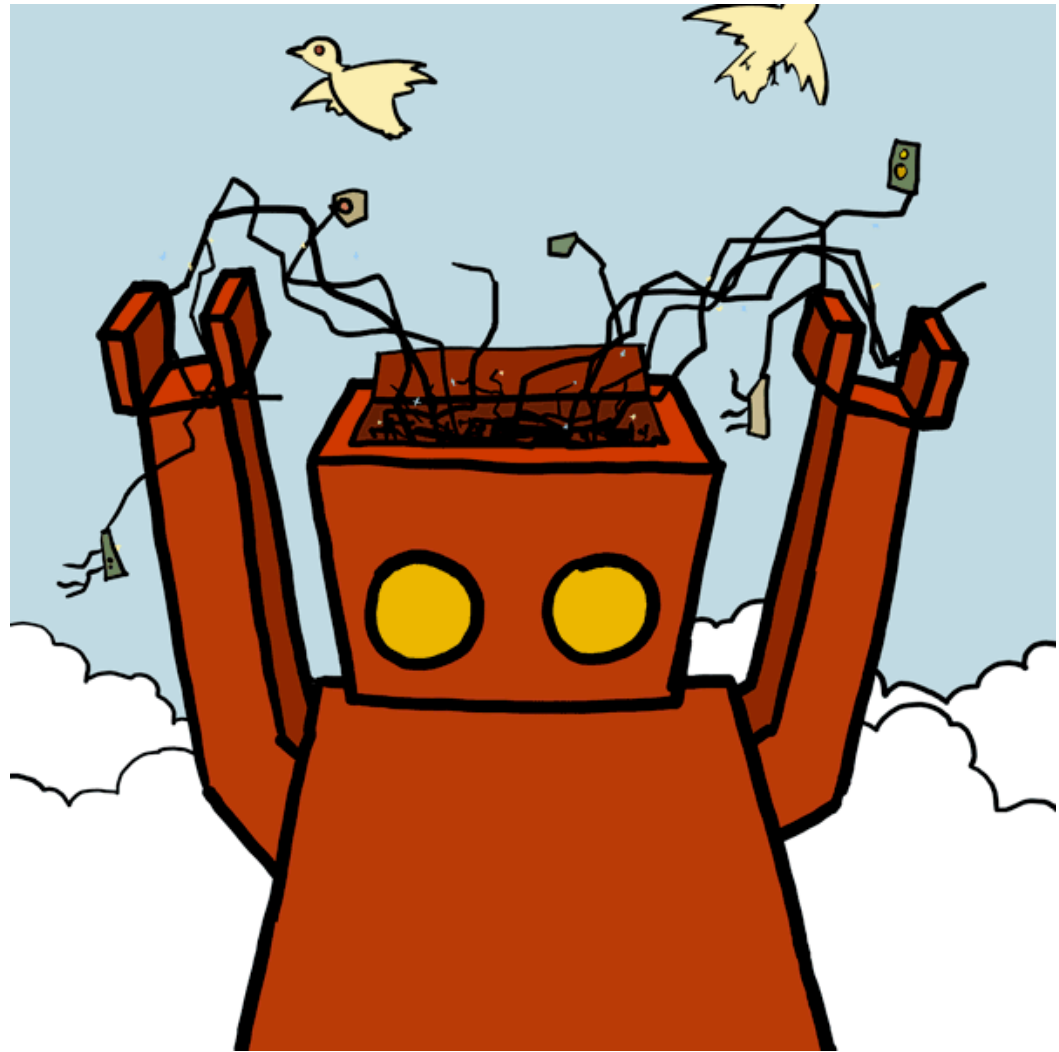- The Java contains implementations of the operations defined in WSDL

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. Discovery: Find a Resource
  - 6. Building a VO: Register with a Community Index
  - 7. Lifetime Management Part II: Lease-based Model
  - 8. Notification: Resource as Notification Producer

- Overview of Tools for GT4 Service Developers

- Tutorial Summary

- Ideas for Further Work

Exercise 3:
Lifetime Management

Resource

3. *Lifetime Mgmt I:* Destroy Resources

2. *State Management I:* Create Resources

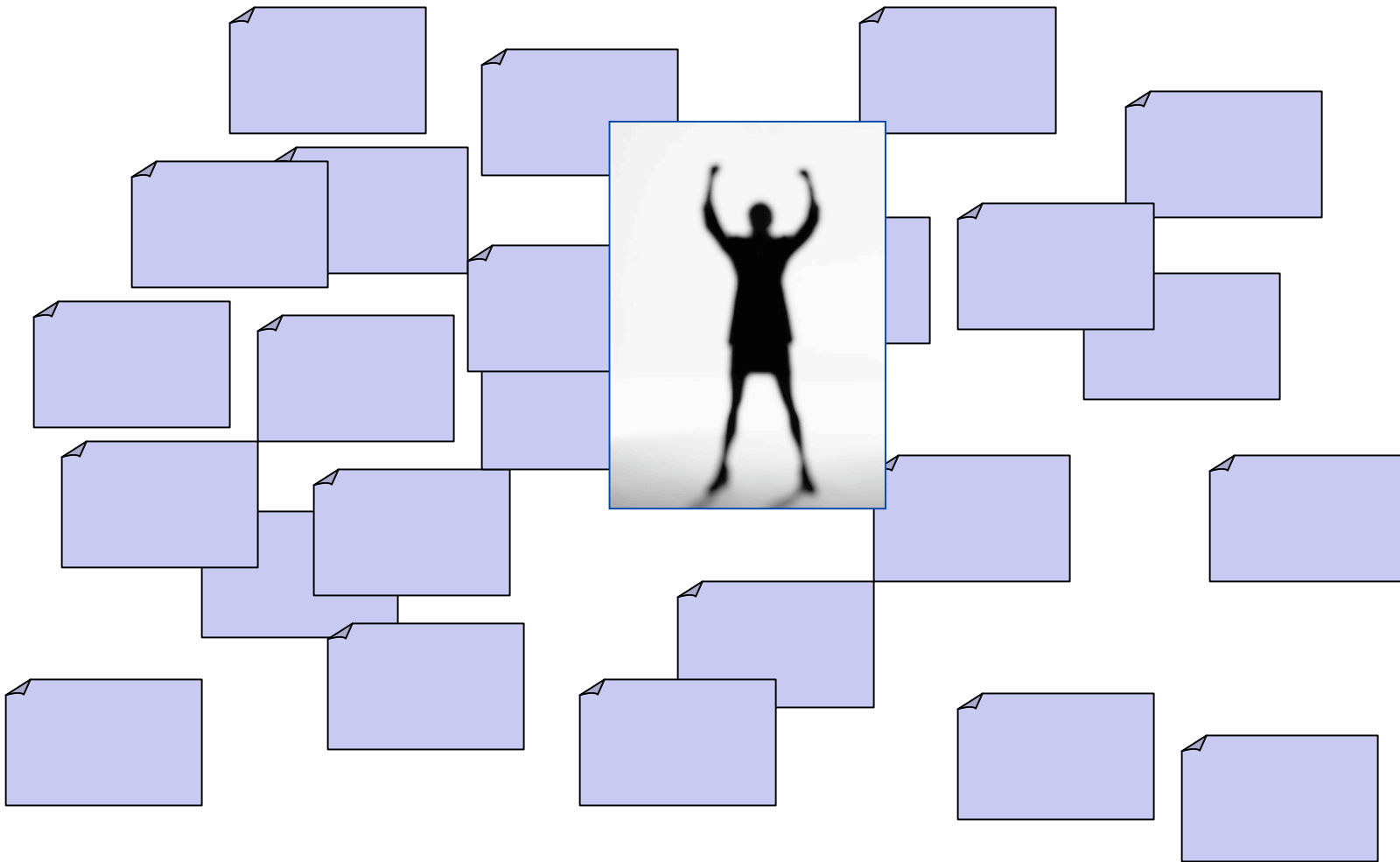**1. *Deployment:* Stand up a StickyNote service**

StickyNote

Client

Concepts

Chapter Overview

- If we keep creating resources, we'll eventually overwhelm our system

- In this chapter we will learn how to destroy resources

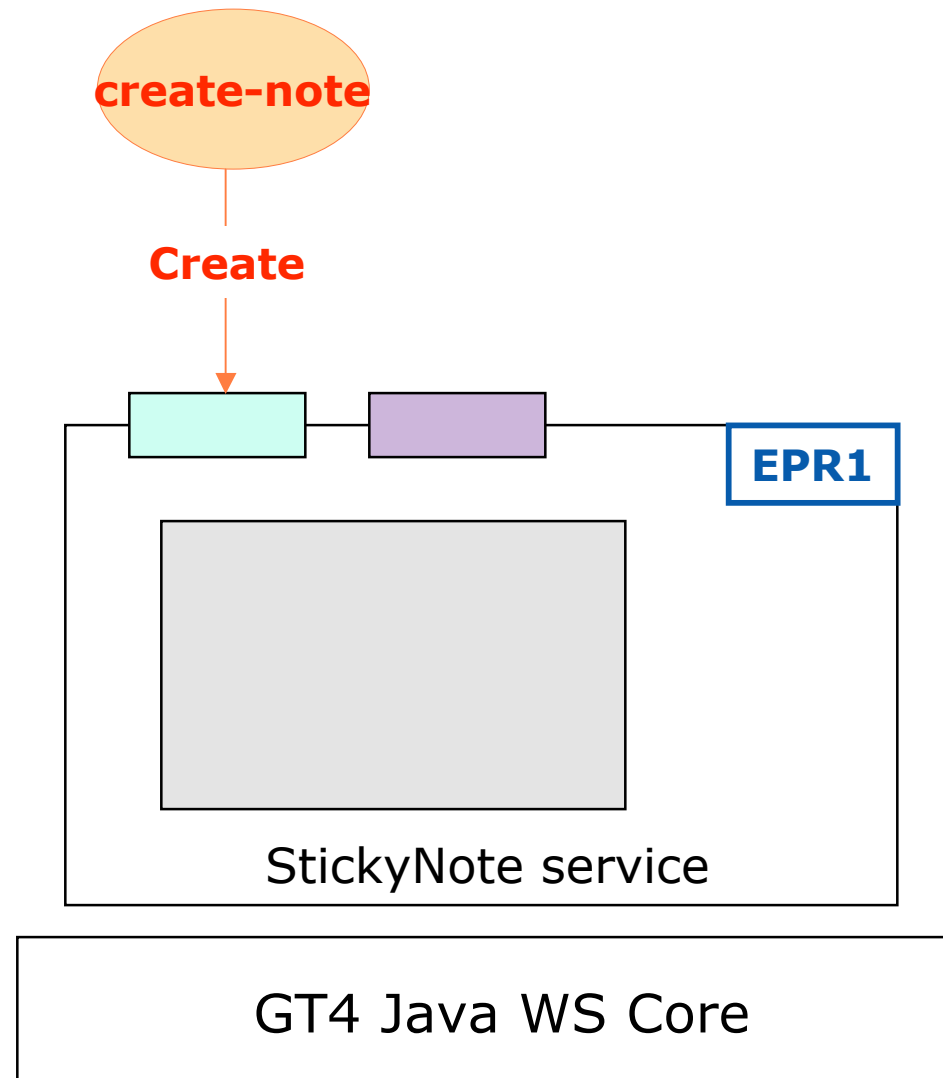- To do so, we must be able to identify an individual resource

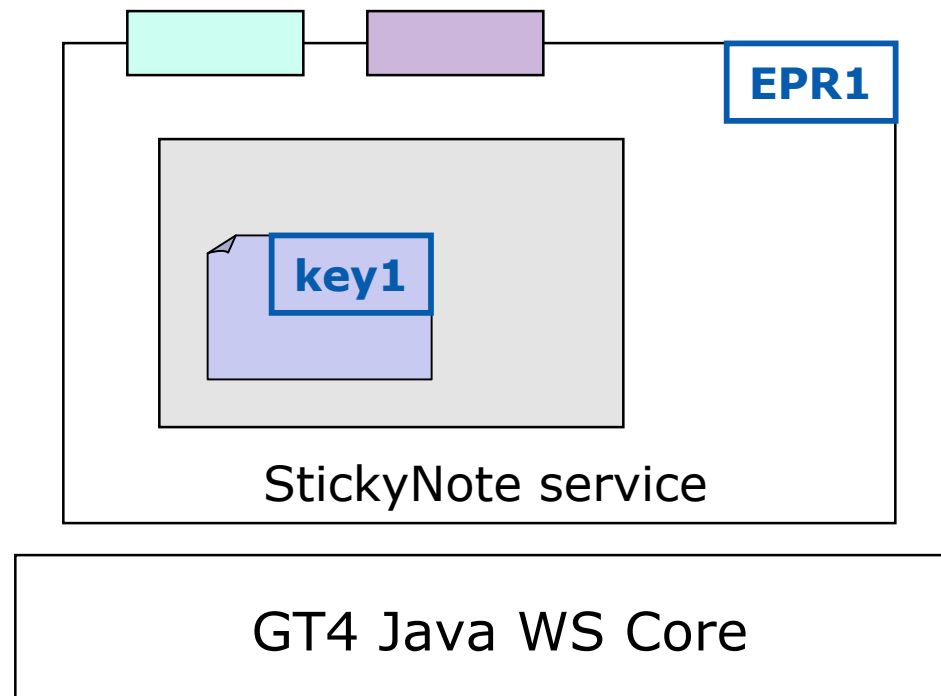# On Lifetime Management

EPRs as Identifiers

- In order to destroy a particular Resource, we must be able to identify it

- In Web services, End Point References (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
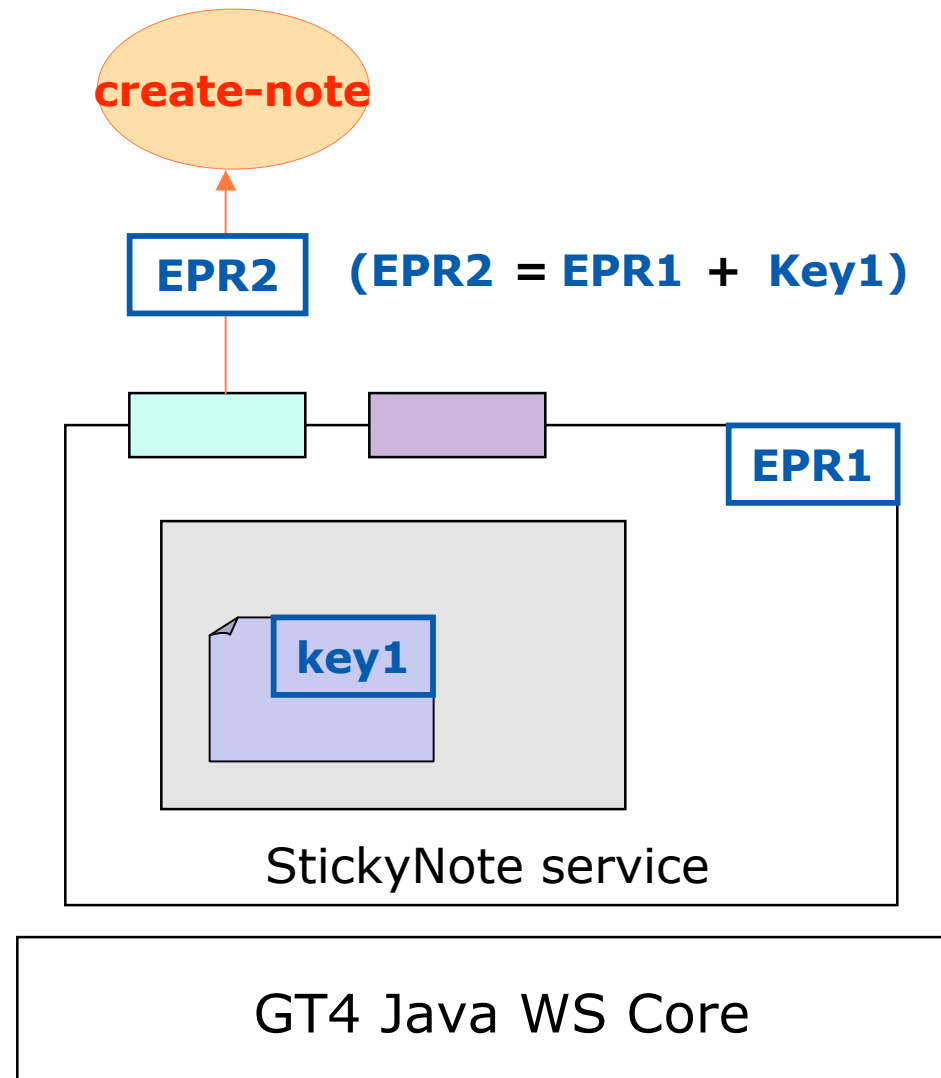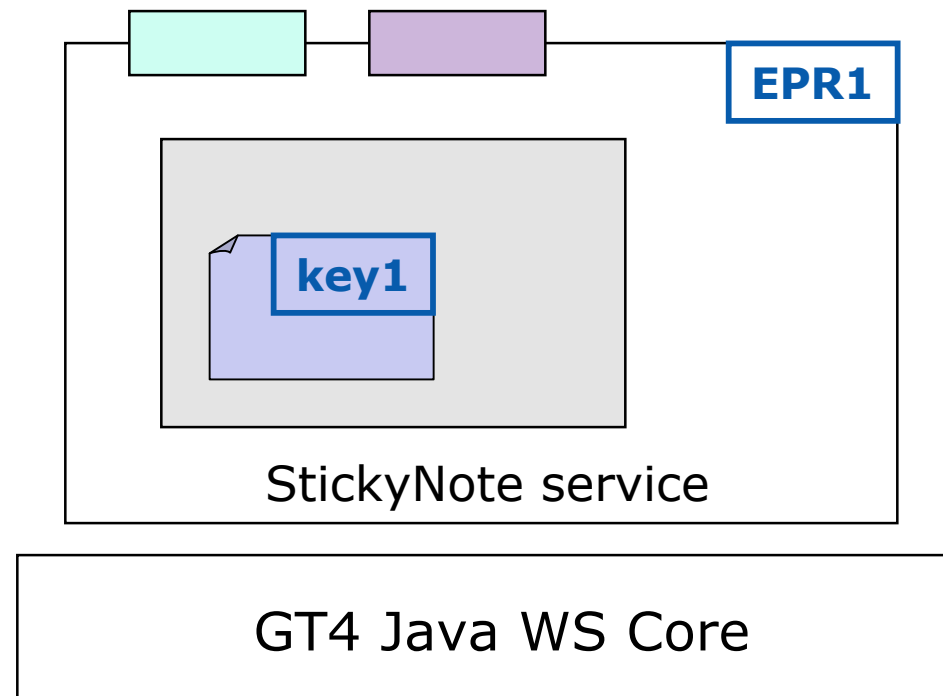
A Runtime Example of EPR Creation

create-note

Create

EPR1

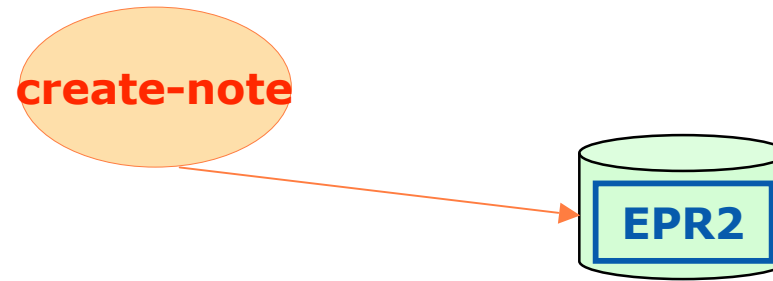StickyNote service

GT4 Java WS Core

A Runtime Example of EPR Creation

create-note

EPR1

key1

StickyNote service

GT4 Java WS Core

A Runtime Example of EPR Creation

create-note

EPR2

$(EPR2 = EPR1 + Key1)$

EPR1

key1

StickyNote service

GT4 Java WS Core

A Runtime Example of EPR Creation

create-note

EPR2

EPR1

key1

StickyNote service

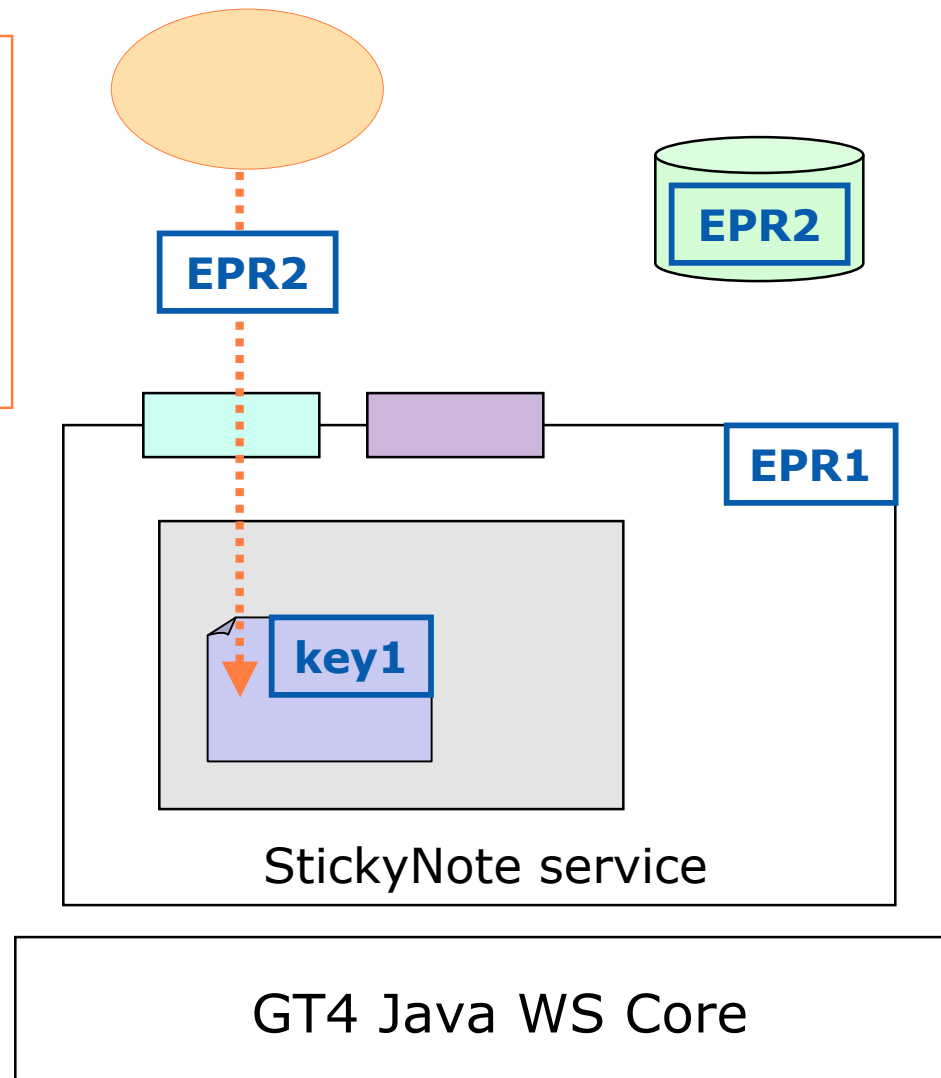GT4 Java WS Core

A Runtime Example of EPR Creation

Subsequent messages that refer to EPR2 will be associated with the new Resource

EPR2

EPR2

EPR2

EPR1

key1

StickyNote service

GT4 Java WS Core

Destroy the Resources you must

- EPRs are useful because they uniquely identify a network-based entity

- Now that we know how to uniquely identify a resource, we can destroy it without risk of destroying the wrong thing
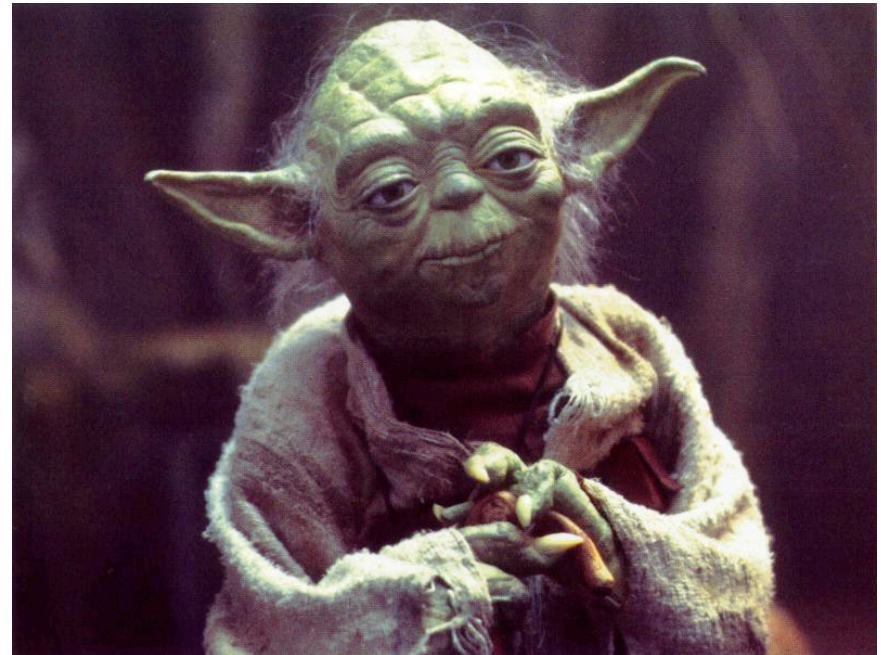
Resource Destruction

- 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

Operation providers are fundamental to the GT4 service programming model

EPR

Operation Providers
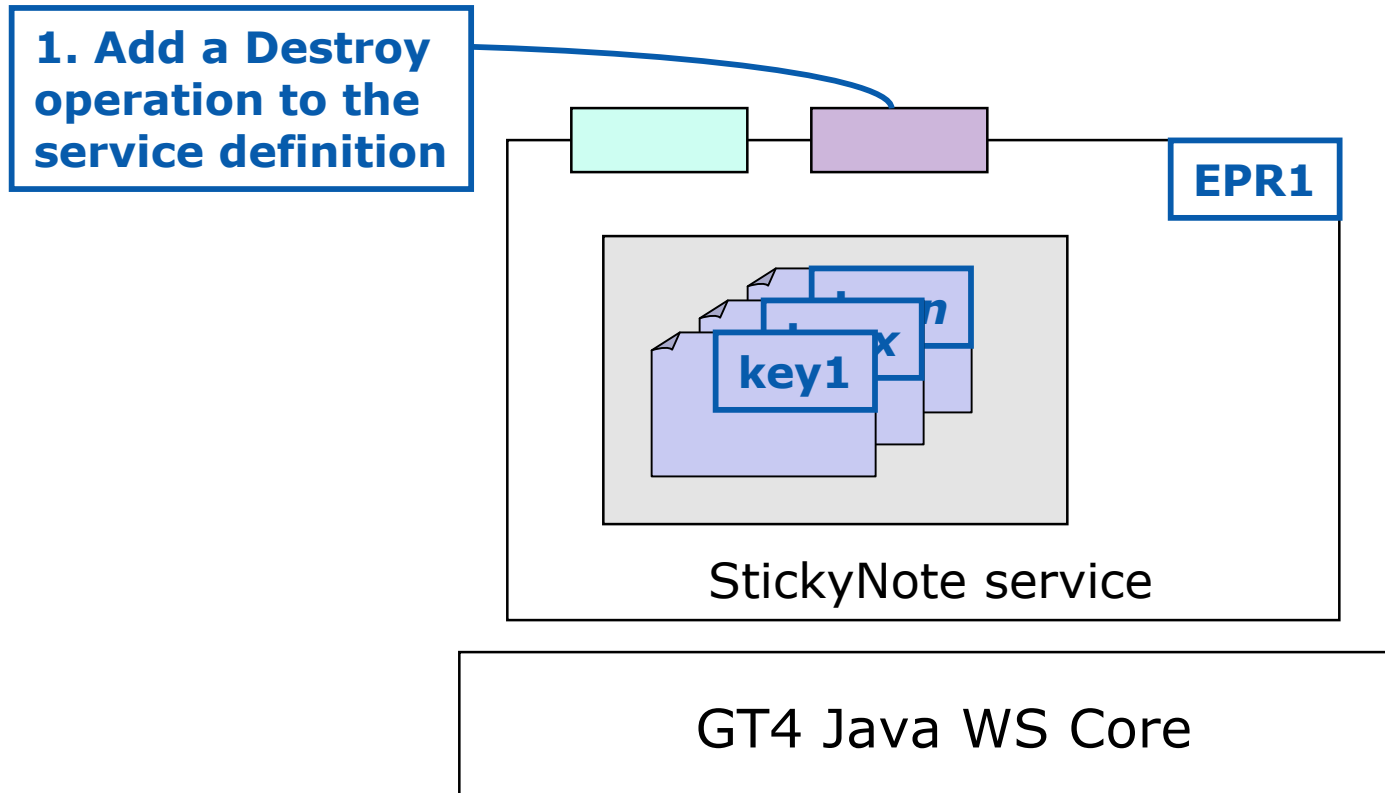
key1

service

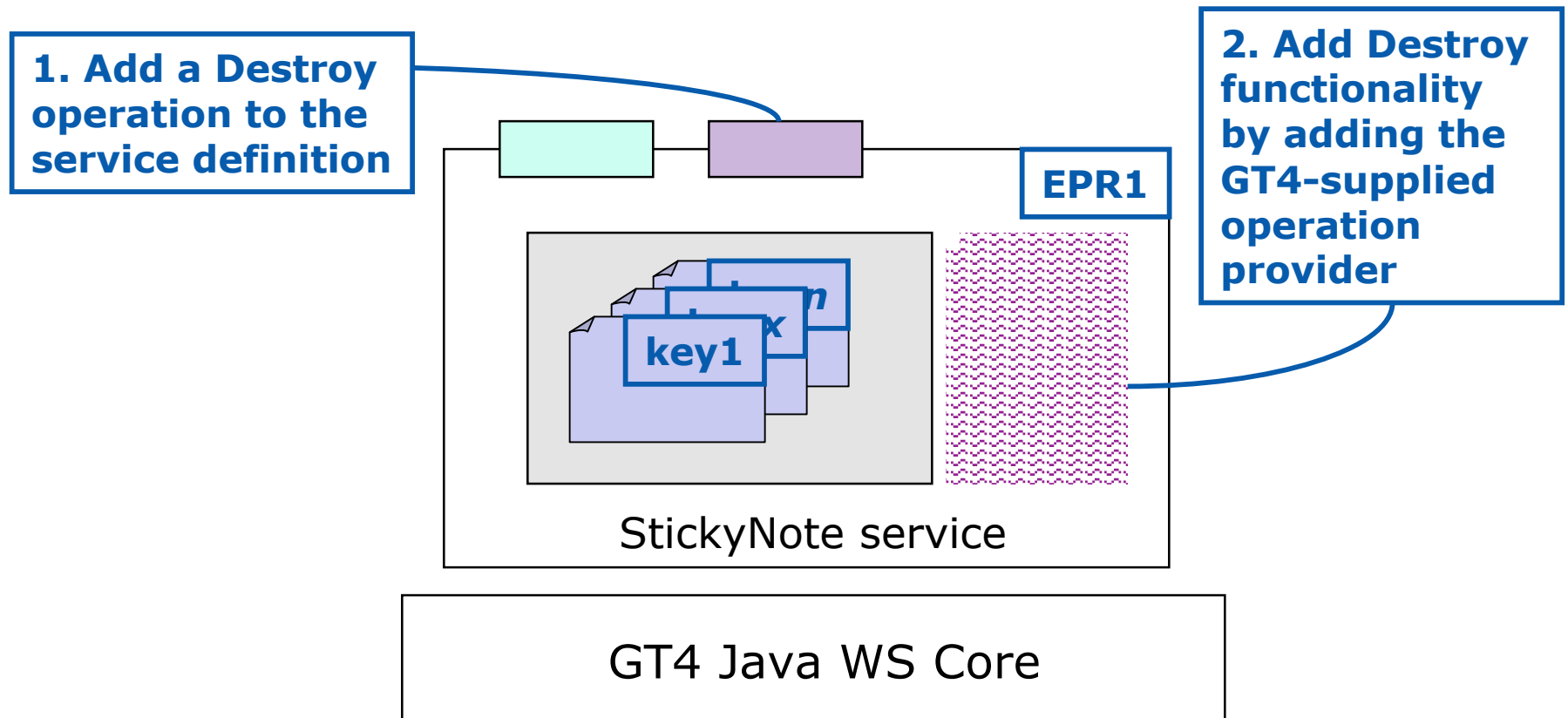GT4 Java WS Core
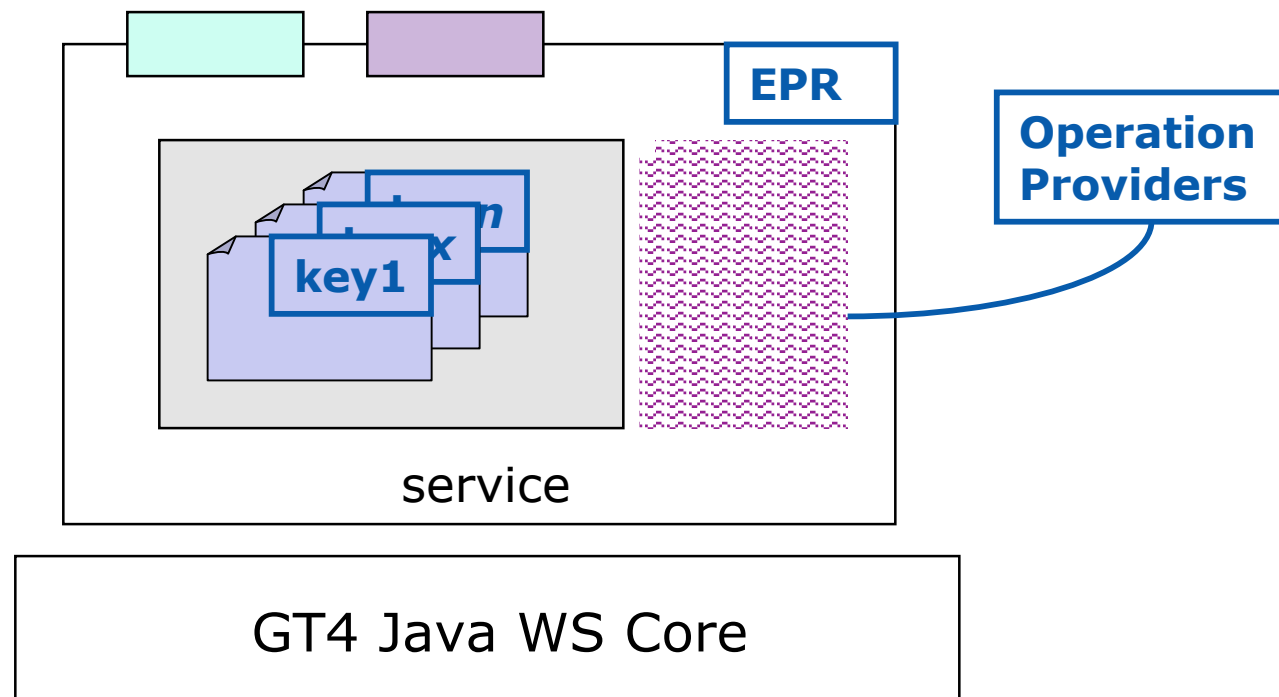
Implementation details

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
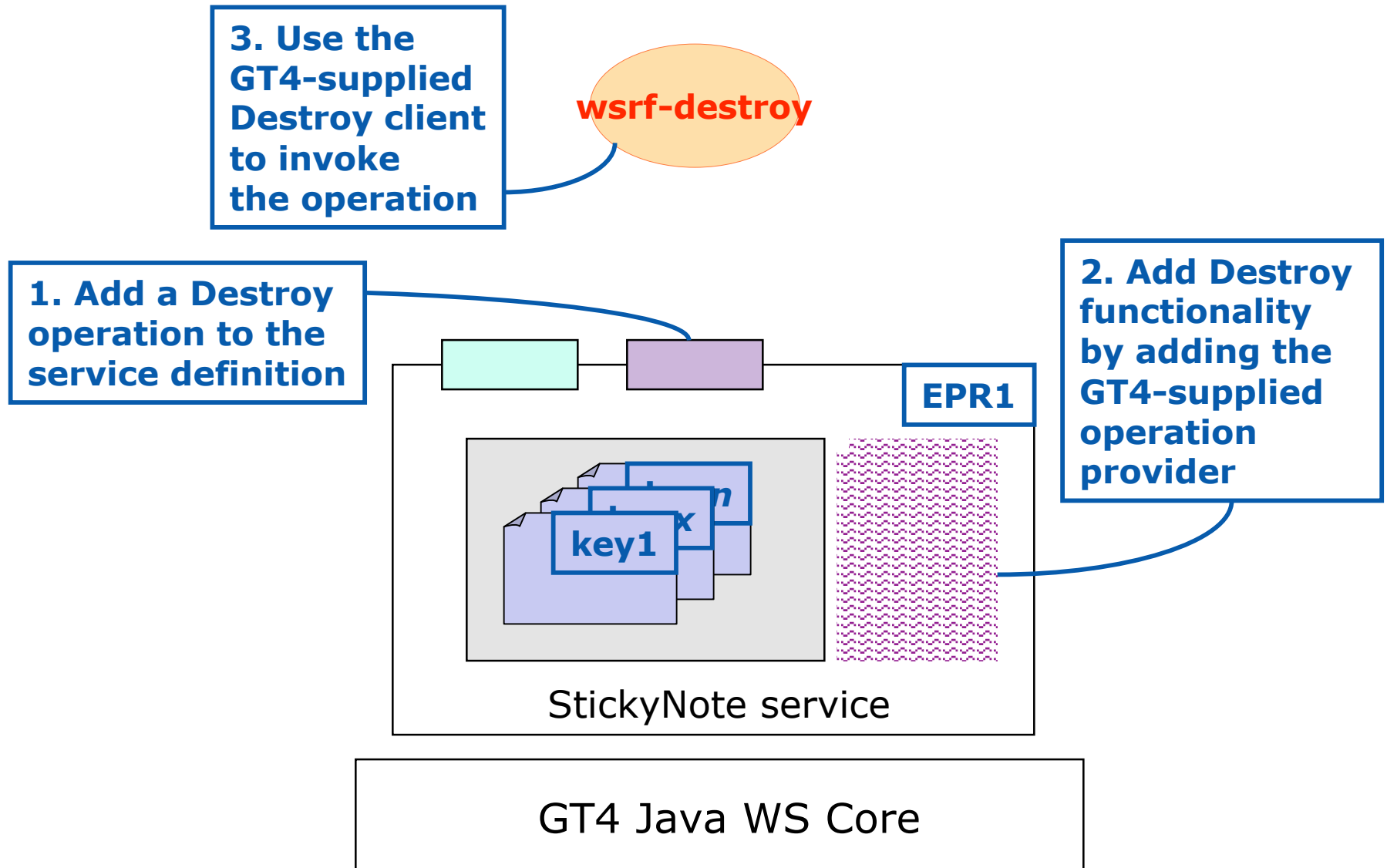
Adding Destroy to StickyNote

1. Add a Destroy operation to the service definition

EPR1

key1

StickyNote service

GT4 Java WS Core

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

StickyNote service

GT4 Java WS Core
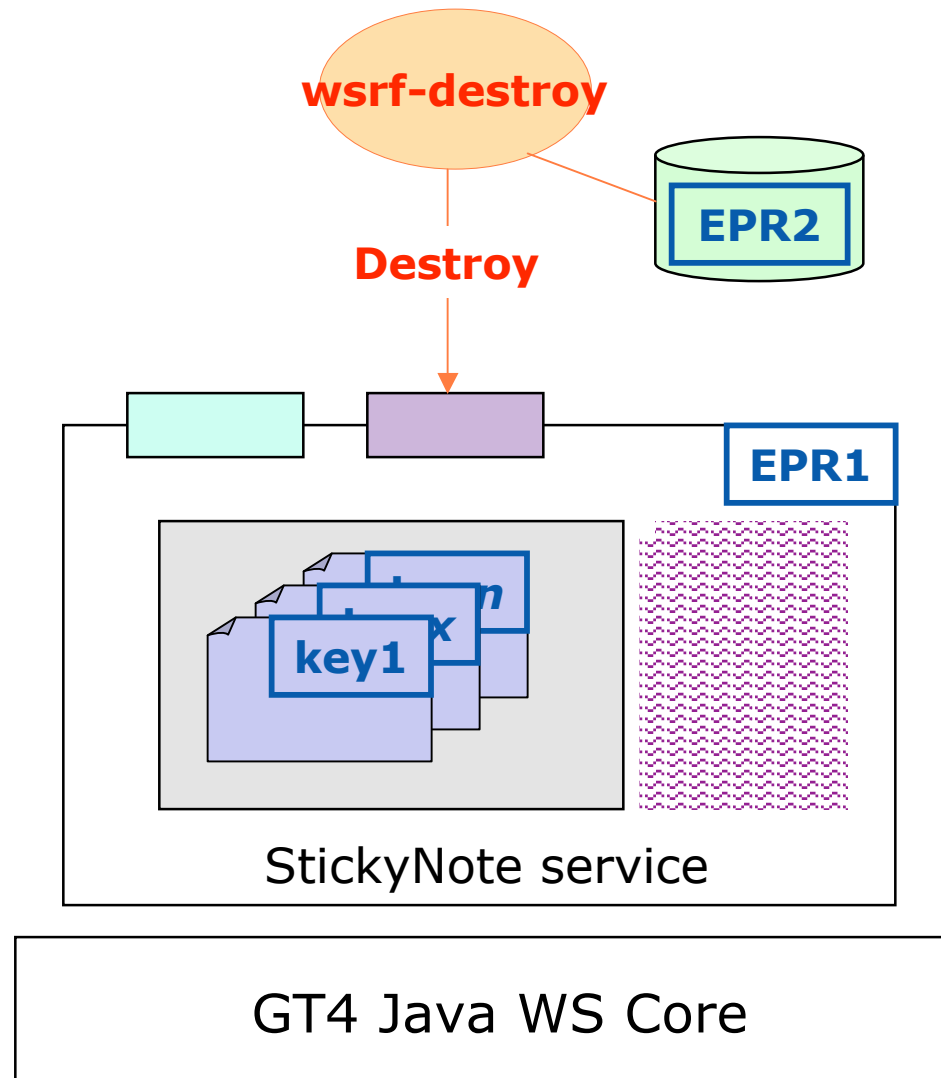
Resource Destruction Runtime
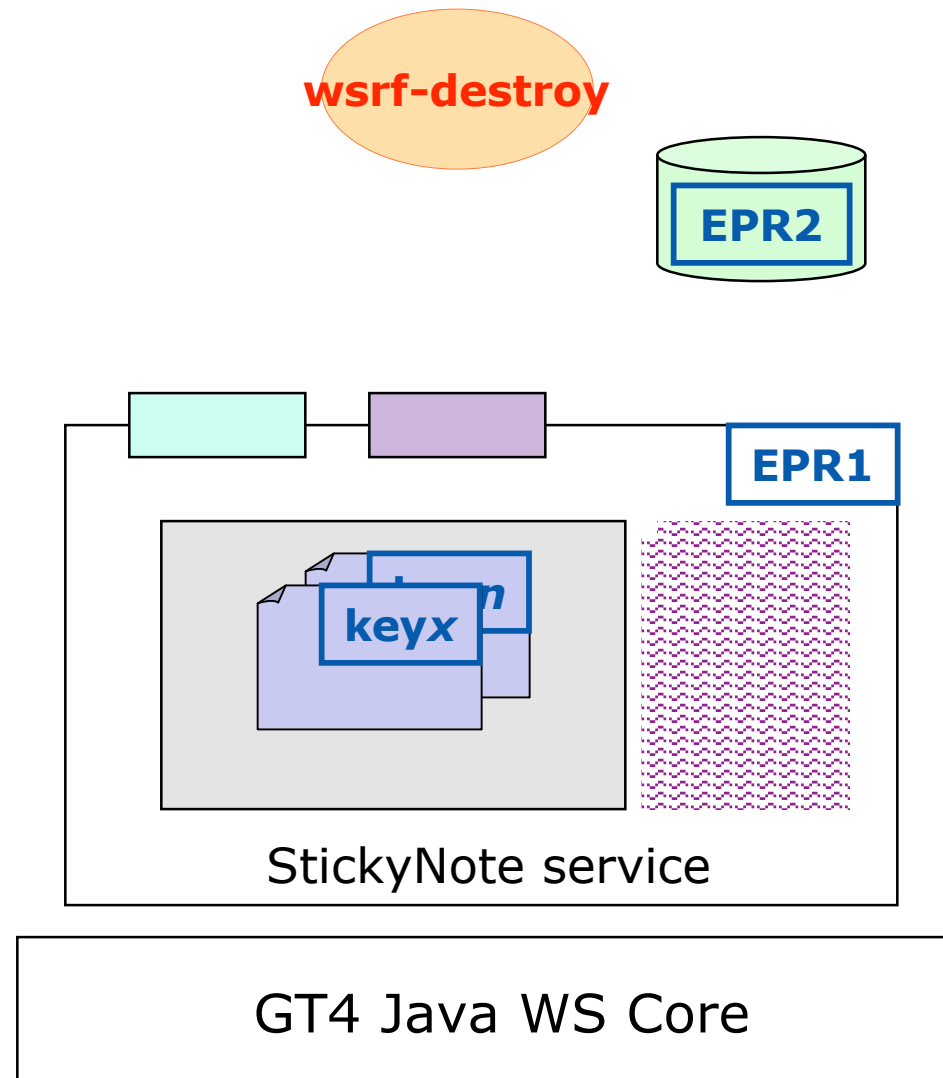
Hands-on Exercises

Student Notes, Chapter 3

Exercise 3 Review

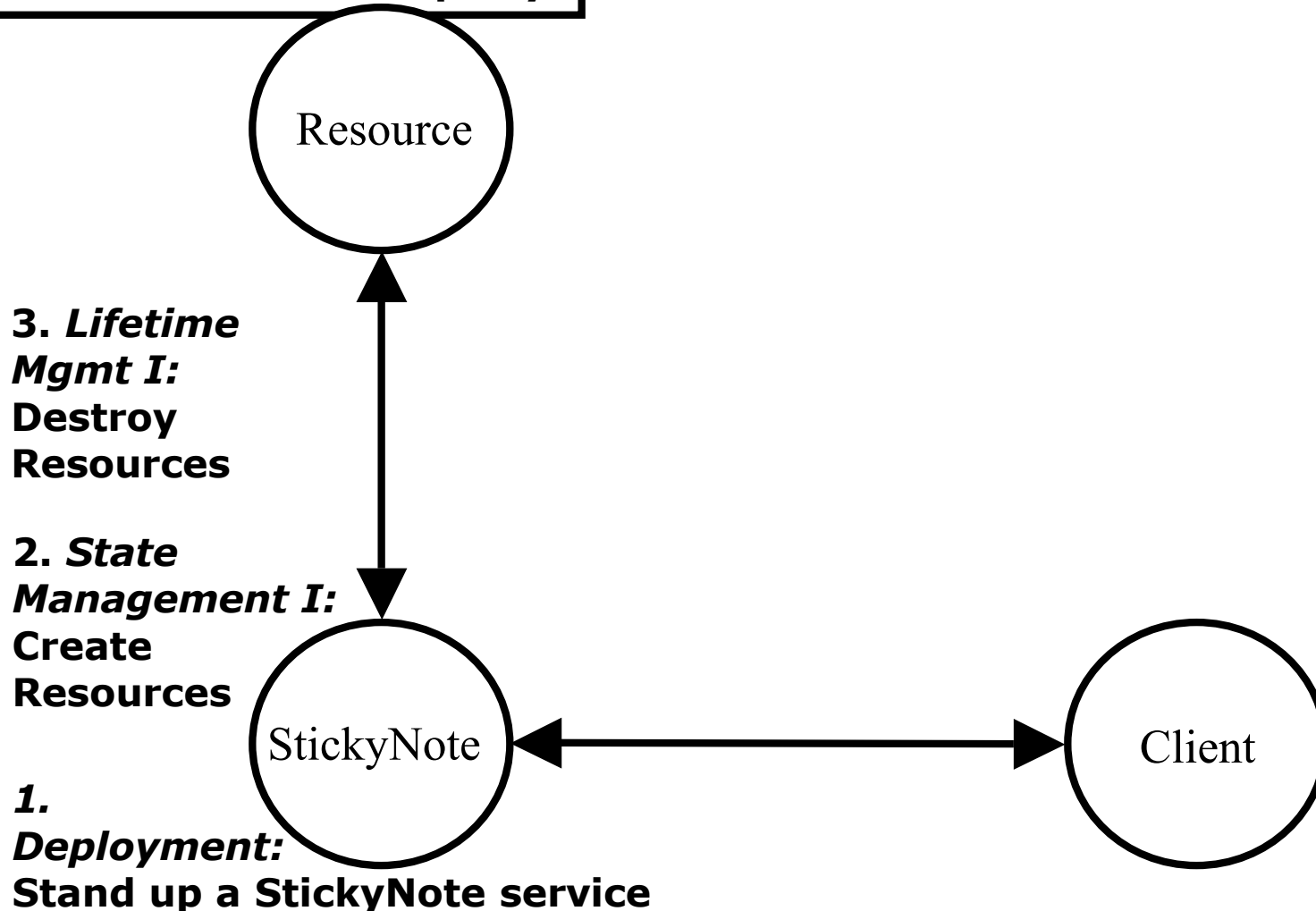- EPRs identify targets (endpoints)
  - In our case, identifying a service+resource pair
- EPRs can be used to identify Resources to destroy
- Service developers use operation providers to add service functionality
- GT4 provides several pre-written operation providers and clients for managing Resources,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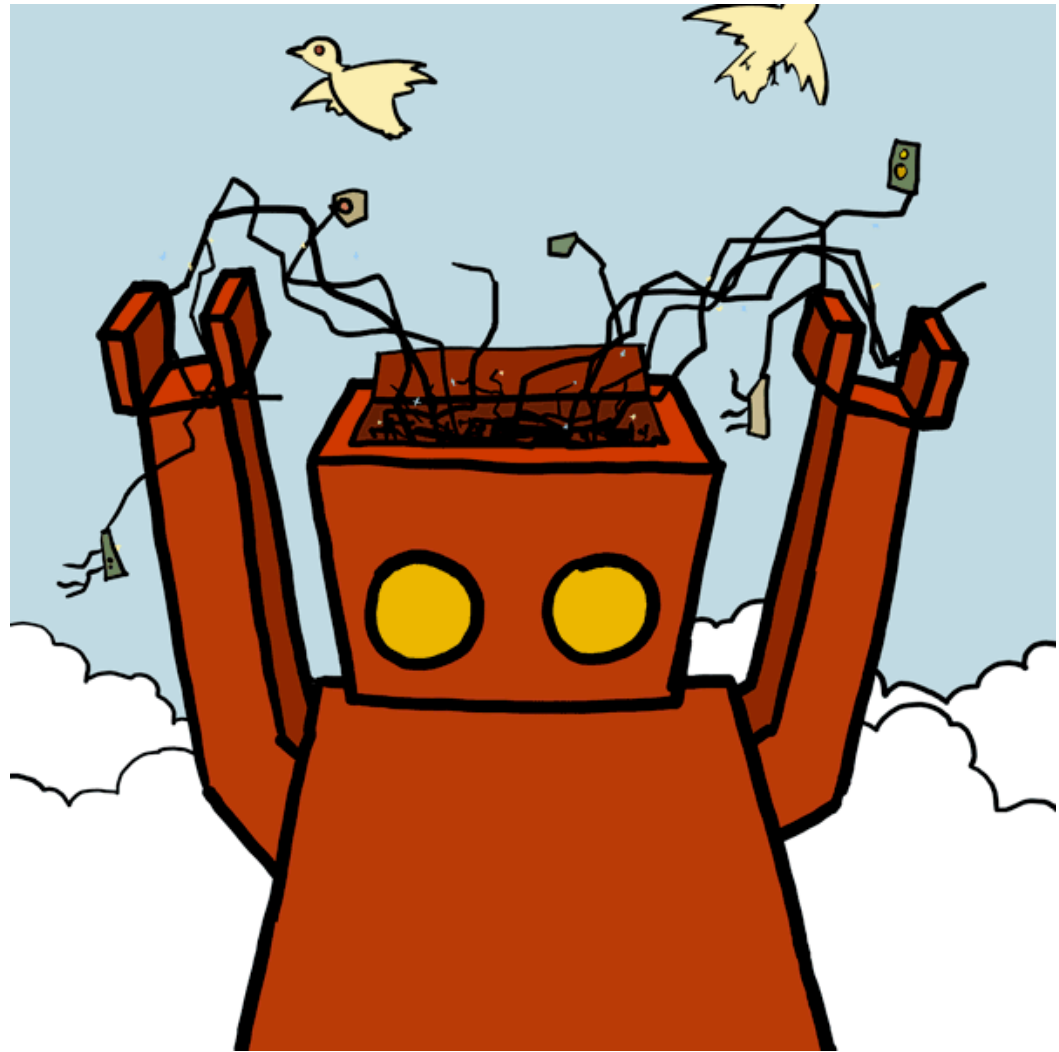
How to Build a Service Using GT4

- 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. Discovery: Find a Resource
    - 6. Building a VO: Register with a Community Index
    - 7. Lifetime Management Part II: Lease-based Model
    - 8. Notification: Resource as Notification Producer
- 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**

Resource

**3. *Lifetime Mgmt I:*
Destroy Resources**

**2. *State Management I:*
Create Resources**

StickyNote

Client

**1. *Deployment:*
Stand up a StickyNote service**

Concepts

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
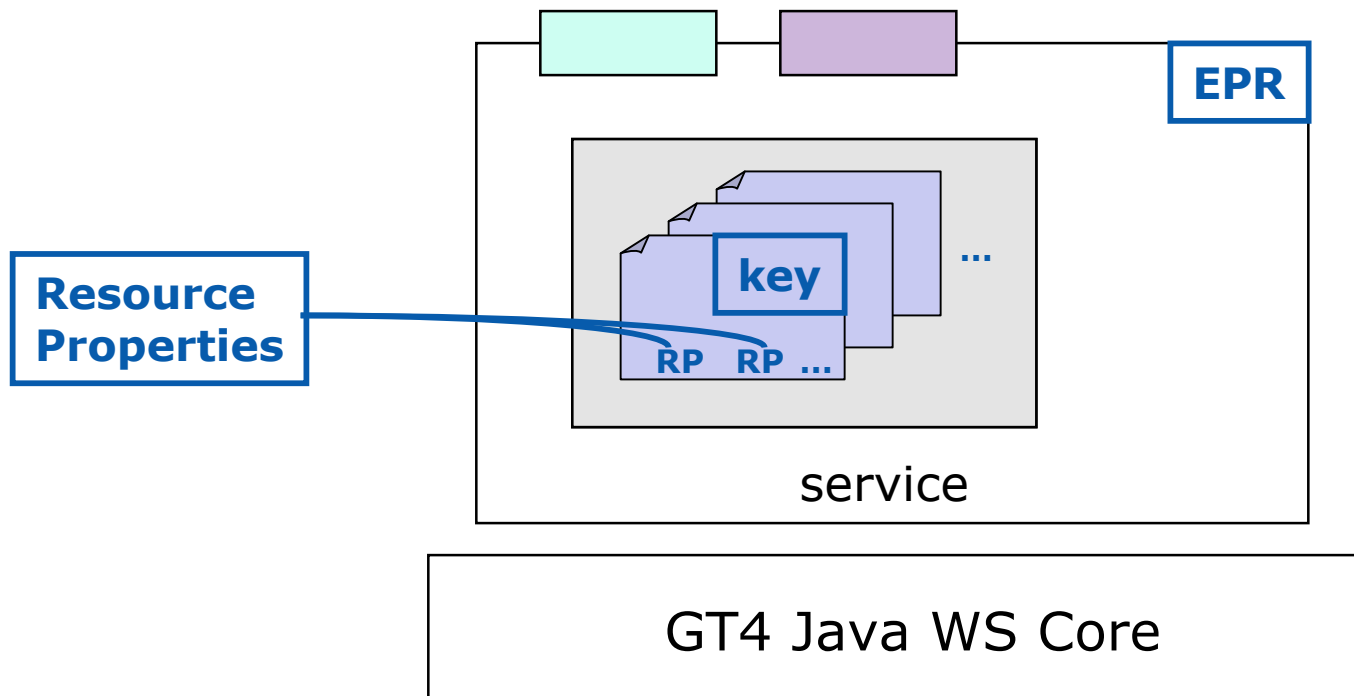
StickyNote Resource Properties

key

NoteContent

EPR

StickyNote service

GT4 Java WS Core
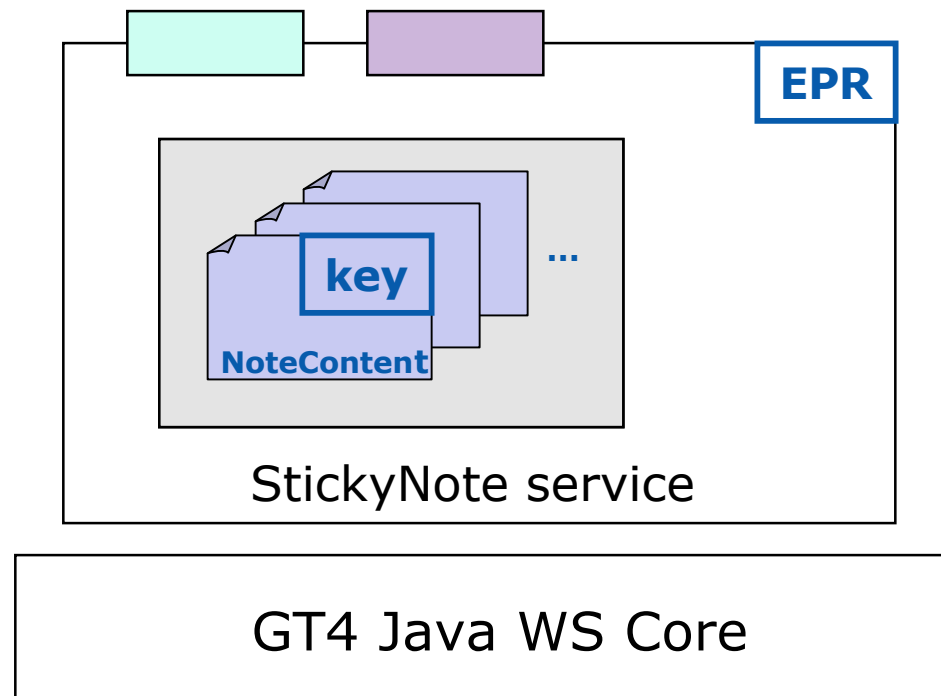
Implementation
details

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 "noteContent" as an RP

- In this exercise, we'll add a "last modified time" resource property to the note
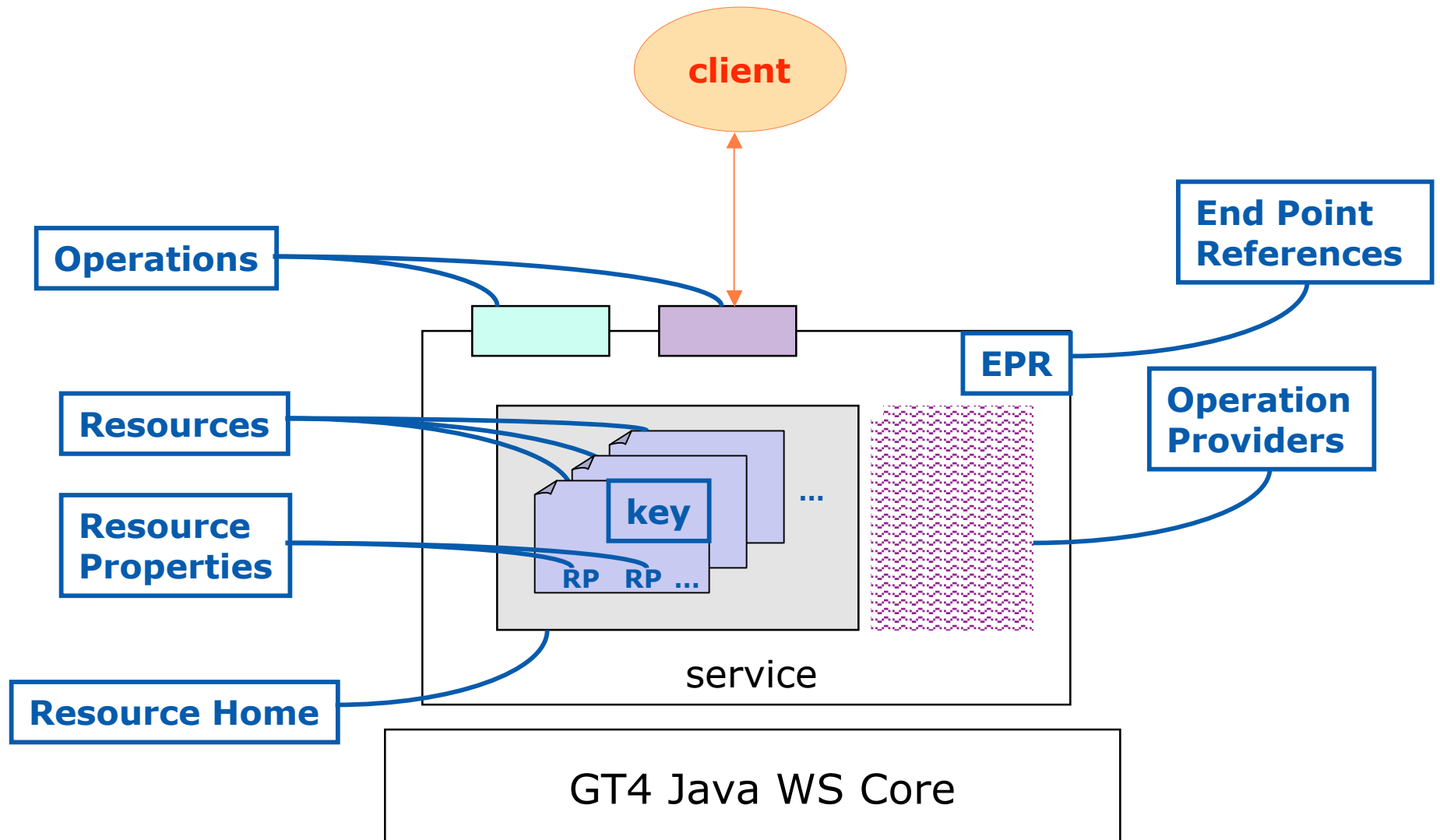
Hands-on Exercises

Student Notes, Chapter 4

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

client

Operations

End Point References

EPR

Resources

Operation Providers

key

...

Resource Properties

RP RP ...

service

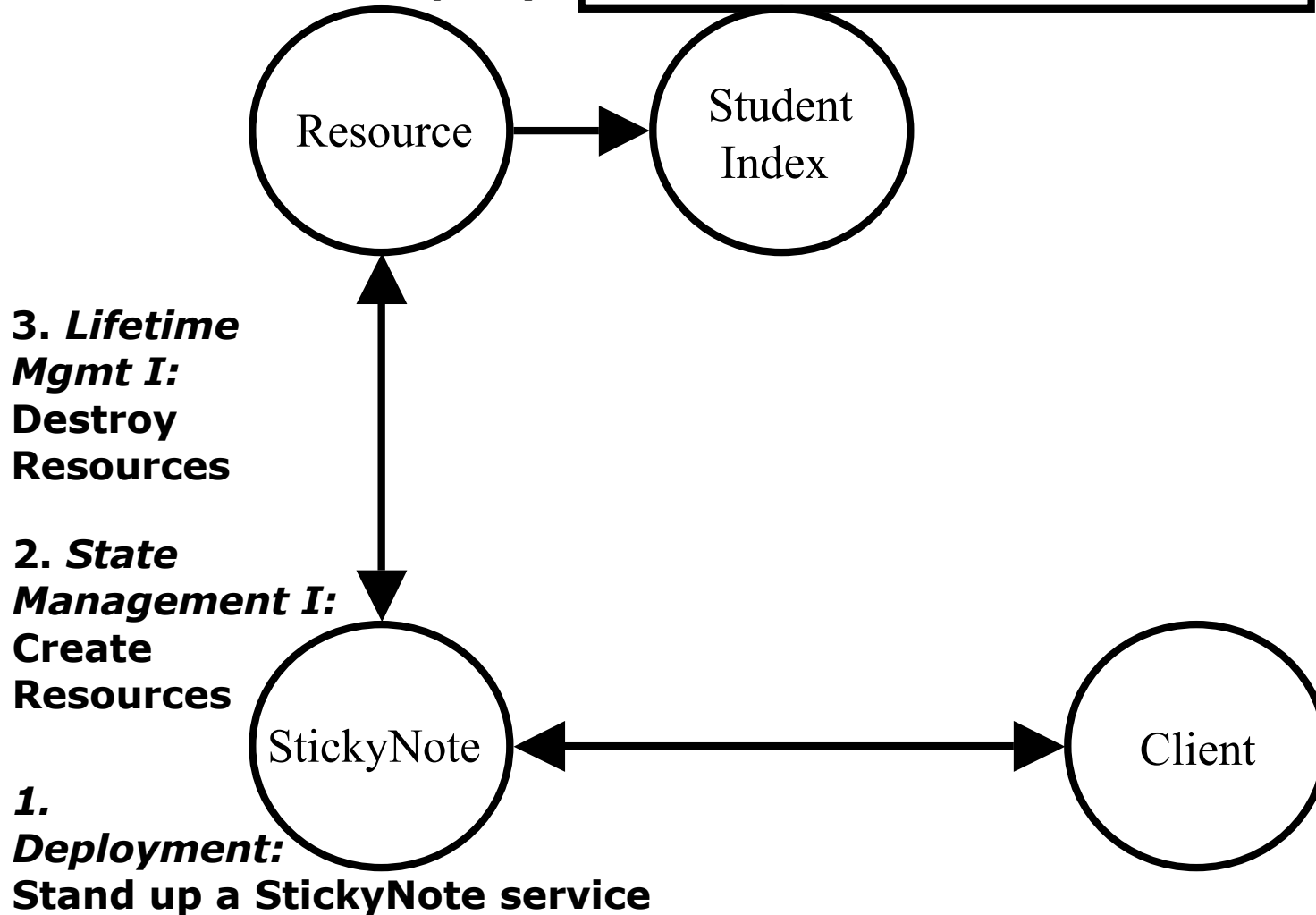**Resource Home**

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. Discovery: Find a Resource**
  - ◆ 6. Building a VO: Register with a Community Index
  - ◆ 7. Lifetime Management Part II: Lease-based Model
  - ◆ 8. Notification: Resource as Notification Producer

- Overview of Tools for GT4 Service Developers
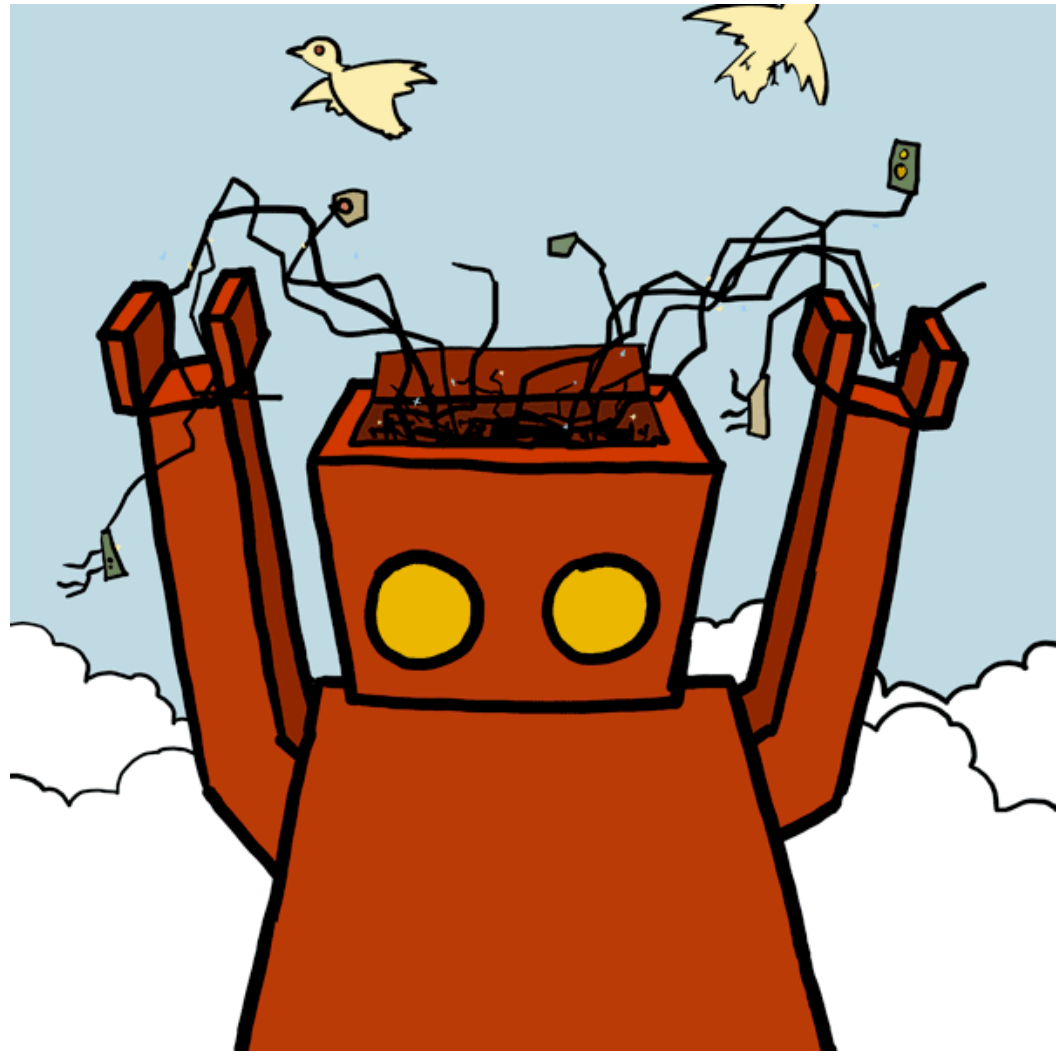- Tutorial Summary
- Ideas for Further Work

Exercise 5: Discovery

4. *State Management II:* Add a Resource Property

5. *Discovery:* Find a Resource

Resource → Student Index

3. *Lifetime Mgmt I:* Destroy Resources

2. *State Management I:* Create Resources

1. *Deployment:* Stand up a StickyNote service
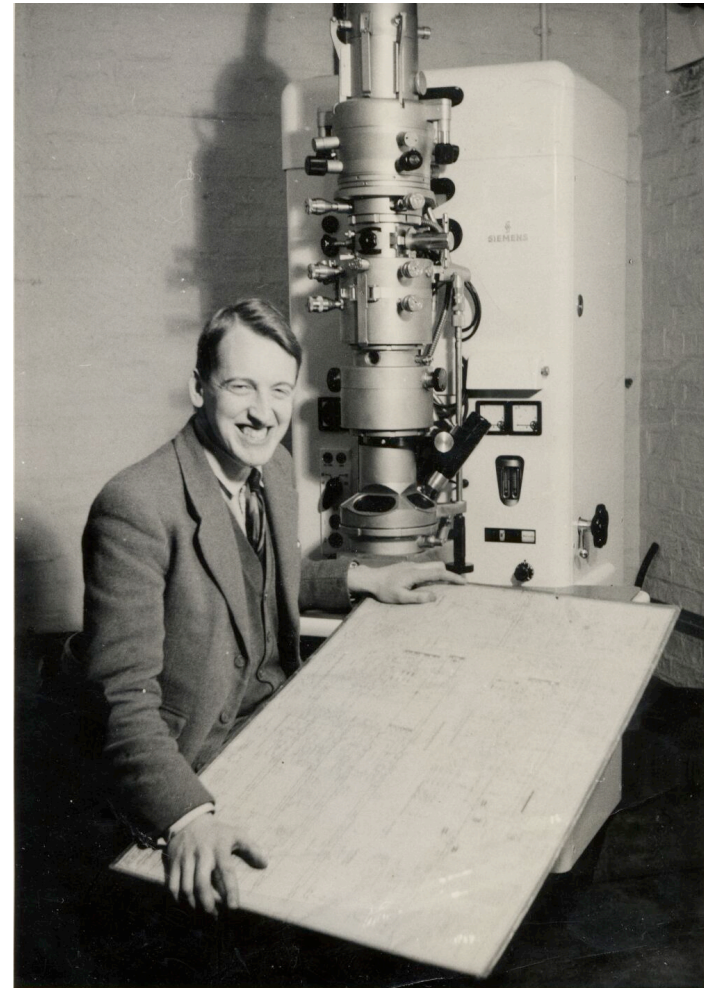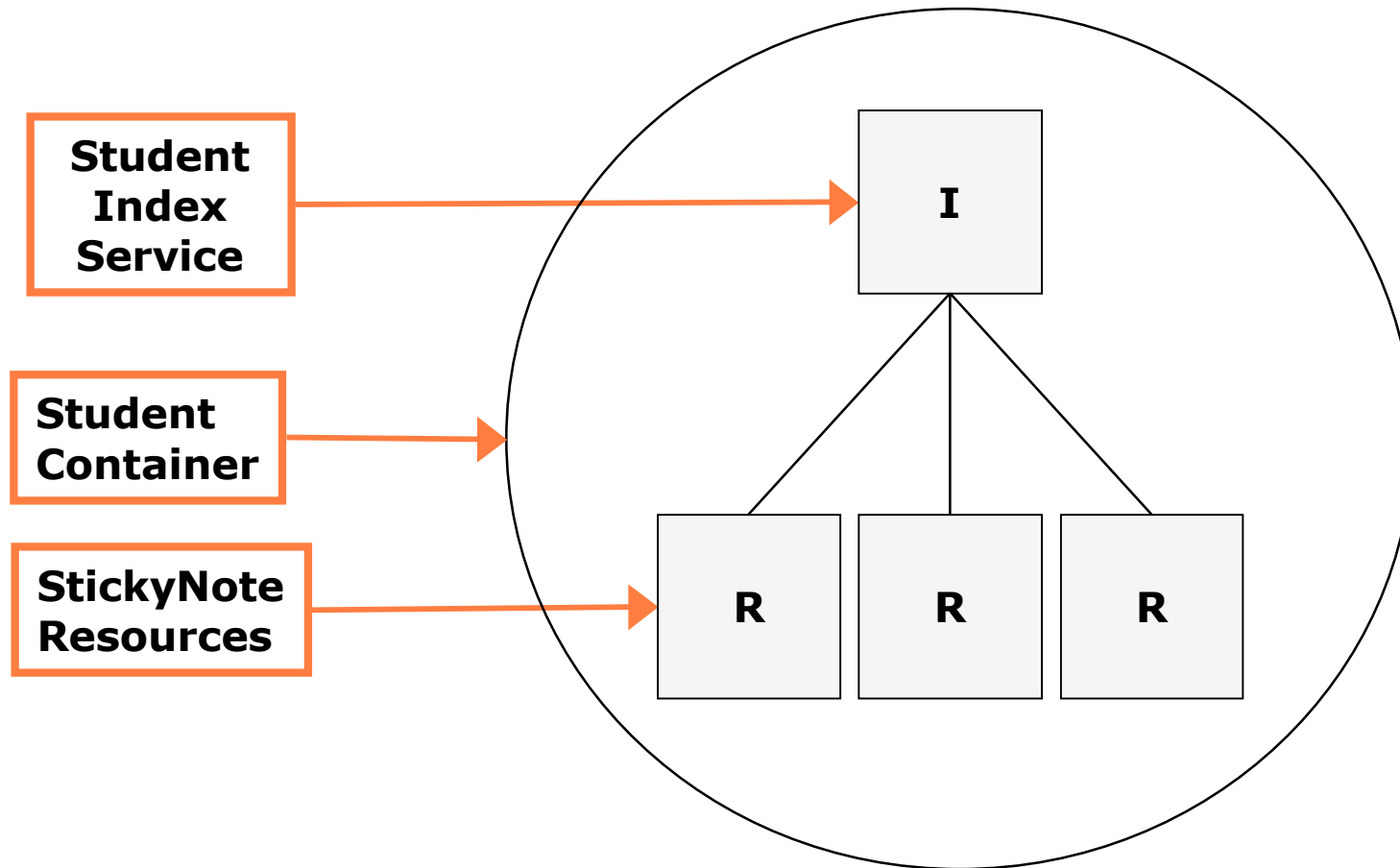
StickyNote ↔ Client

Concepts

On Discovery

- In Grid computing we often need to find a resource that has some property

- In our case, a person may want to find a note containing a particular content, such as "FindMe"

- *Discovery* is the process of identifying resources based on their attributes
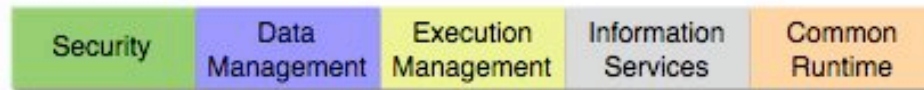
Aggregating Resources

- Thus far we have manually kept track of the existence our individual sticky notes
  - In real grid deployments this is infeasible
- GT4 includes an implementation of an index service that can be used to collect available resources
- In this chapter we will register our sticky note resources with a local Index
- After stickynote resources are registered, we will be able to search the Index for notes matching a particular value, facilitating resource discovery

Registering with a Local Index

Student Index Service

Student Container

StickyNote Resources

Globus Toolkit® version 4 (GT4)

Community Scheduler Framework

Community Authorization	Data Replication	Grid Telecontrol Protocol	WebMDS	Python WS Core
Delegation	OGSA-DAI	Workspace Management	Index	C WS Core
Authentication Authorization	Reliable File Transfer	Grid Resource Allocation & Management	Trigger	Java WS Core

WS Components

| Pre-WS Authentication Authorization | GridFTP | Pre-WS Grid Resource Allocation & Management | Monitoring & Discovery (MDS2) | C Common Libraries |
| Credential Management | Replica Location | | | eXtensible IO (XIO) |

Non-WS Components

| Security | Data Management | Execution Management | Information Services | Common Runtime |

Core GT Component: public interfaces frozen between incremental releases; best effort support

Contribution/Tech Preview: public interfaces may change between incremental releases

Deprecated Component: not supported; will be dropped in a future release

Implementation details

On Searching the Index

- The index service caches data of the Resources that have registered with it

- We can dump the entire content of the Index using a GT4 command-line client called *wsrf-query*

- However, if many Resources have registered, the index service will contain a large amount of data

- It is therefore often desirable to query an Index such that only interesting data is returned

Searching Inside Resource Properties

- To search the data we can use an XPath query against the Resource Property Set of the Index

- 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 Index…

Searching the Entries of a GT4 Index

We can find each note that contains the word 'hello.' by sending the following XPath query to the Index:

//*[local-name()='Entry'][.//*[contains(.,"hello.")]]

A human translation of this syntax:

"Select all the entries that contain the string 'hello.'"

The Work of this Chapter

- In this exercise we'll create a new client called *search-note* for searching the index using XPath

- We'll also override the "add" method in the Resource Home with logic to register the StickyNote Resource in the local index service

- The registration will cause a new entry in the index to be created each time a new StickyNote resource is created

Sidebar: Service Group

- Indexes are a kind of **service group**

- Service Groups are defined as part of the WSRF specifications

- Service groups represent collections of services and resources

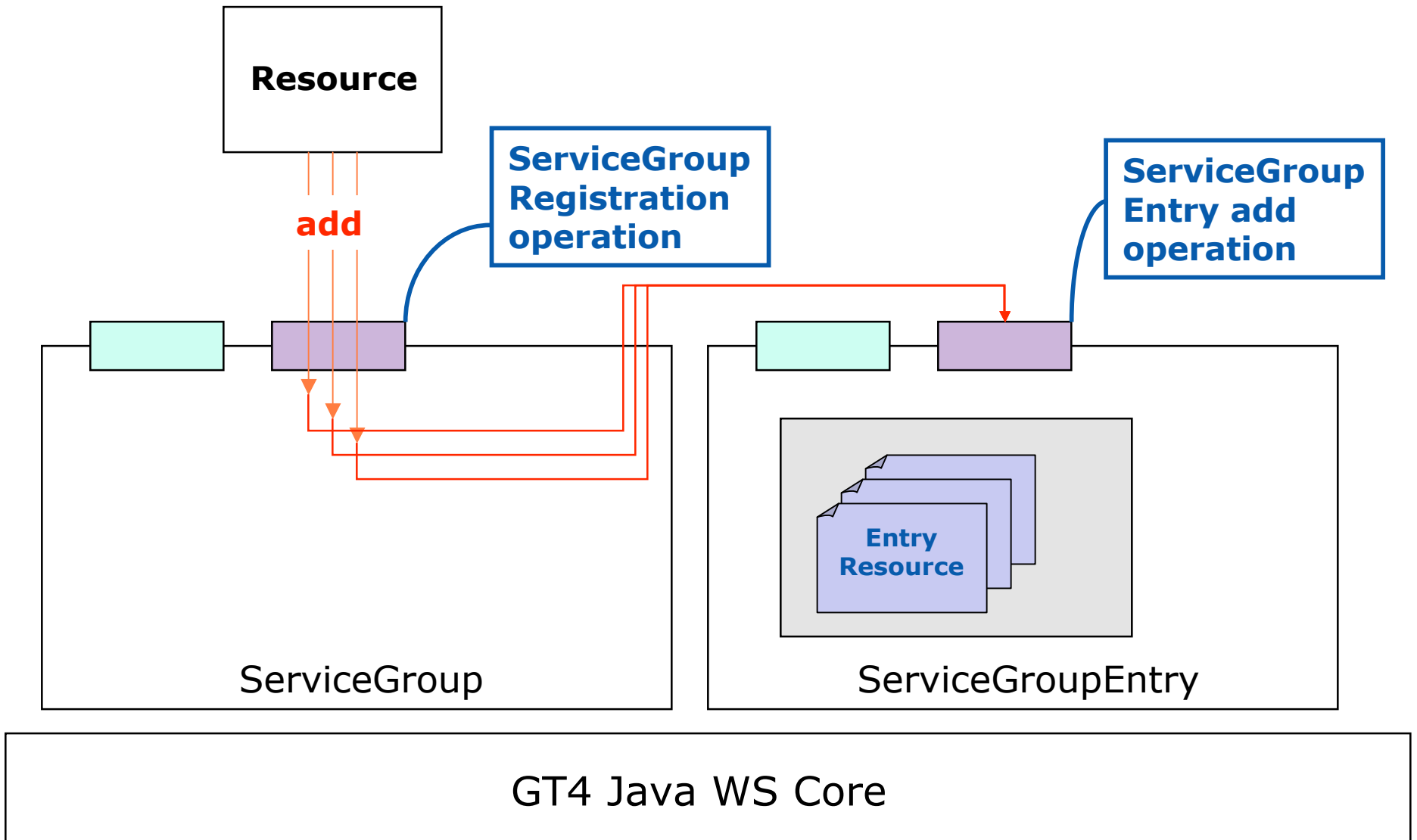- Each service group registration is represented as a ServiceGroupEntry resource

Service Group Runtime

Resource

add

ServiceGroup Registration operation

ServiceGroup

GT4 Java WS Core
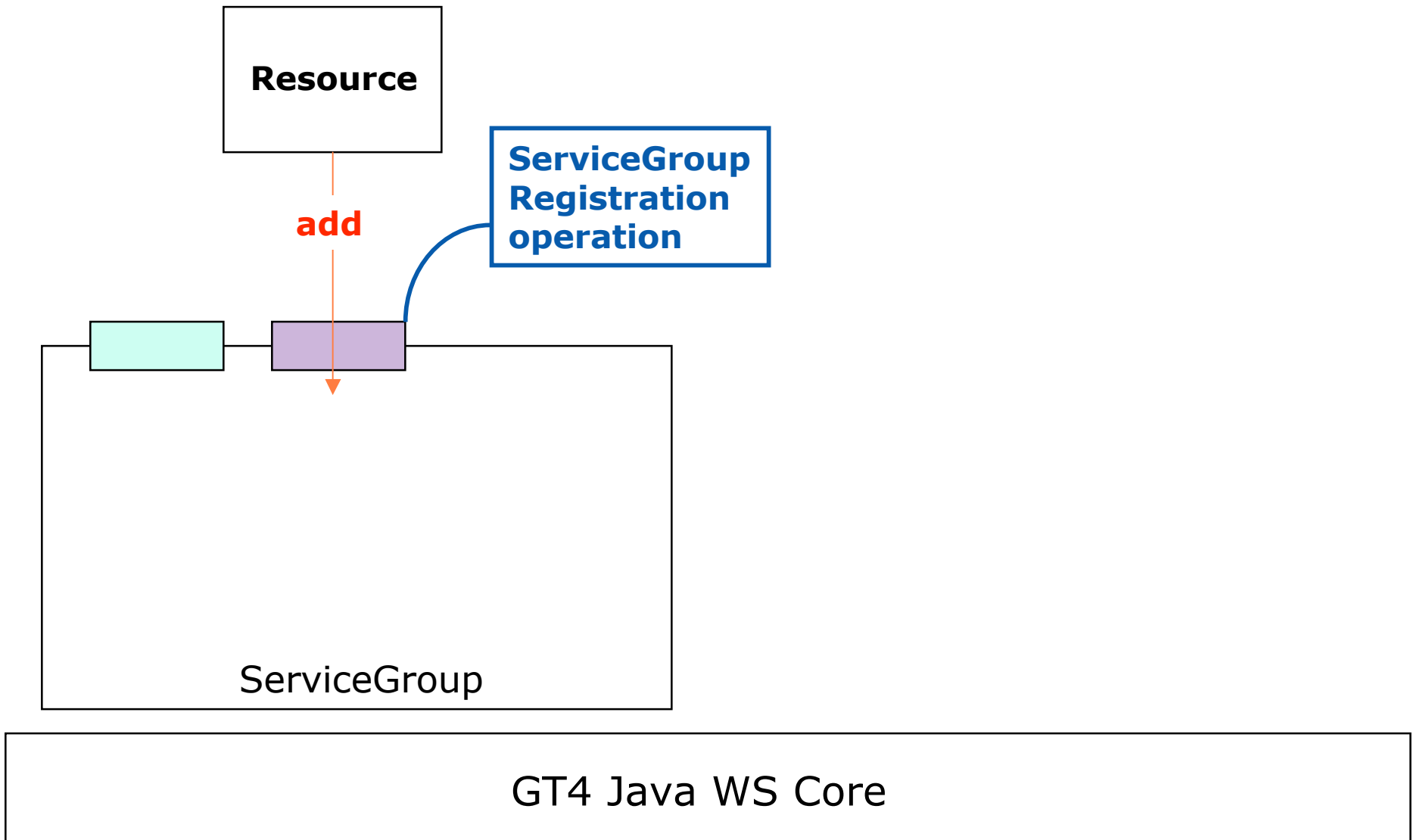
Service Group Runtime
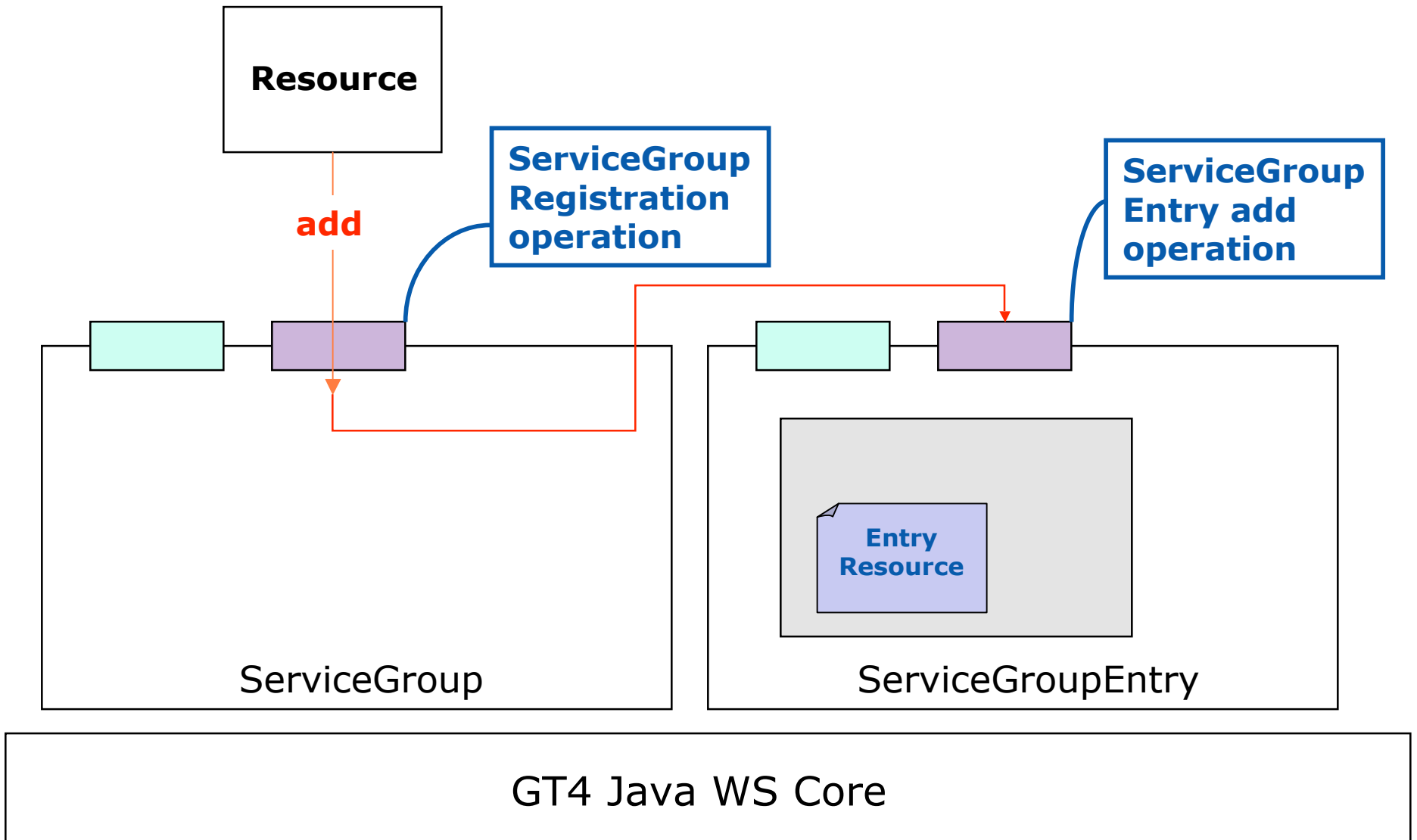
Service Group Runtime

Hands-on Exercises

Student Notes, Chapter 5

ISSGC'05: How to Build a Service Using GT4

Exercise 5 Review

- GT4 includes an implementation of an Index service which is deployed by default for installations including the WS MDS component

- The default index is often used to aggregate local Resources of interest

- Registrants in an index are represented as service group entry resources

- The contents of an Index can be inspected using the GT command-line tool wsrf-query, as well as more complicated XPath queries

- The Index provides a structure that enables one to discover services based on Resource Properties

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. Discovery: Find a Resource
- ◆ **6. Building a VO: Register with a Community Index**
- ◆ 7. Lifetime Management Part II: Lease-based Model
- ◆ 8. Notification: Resource as Notification Producer

● Overview of Tools for GT4 Service Developers

● Tutorial Summary

● Ideas for Further Work

Exercise 6: Virtual Organizations

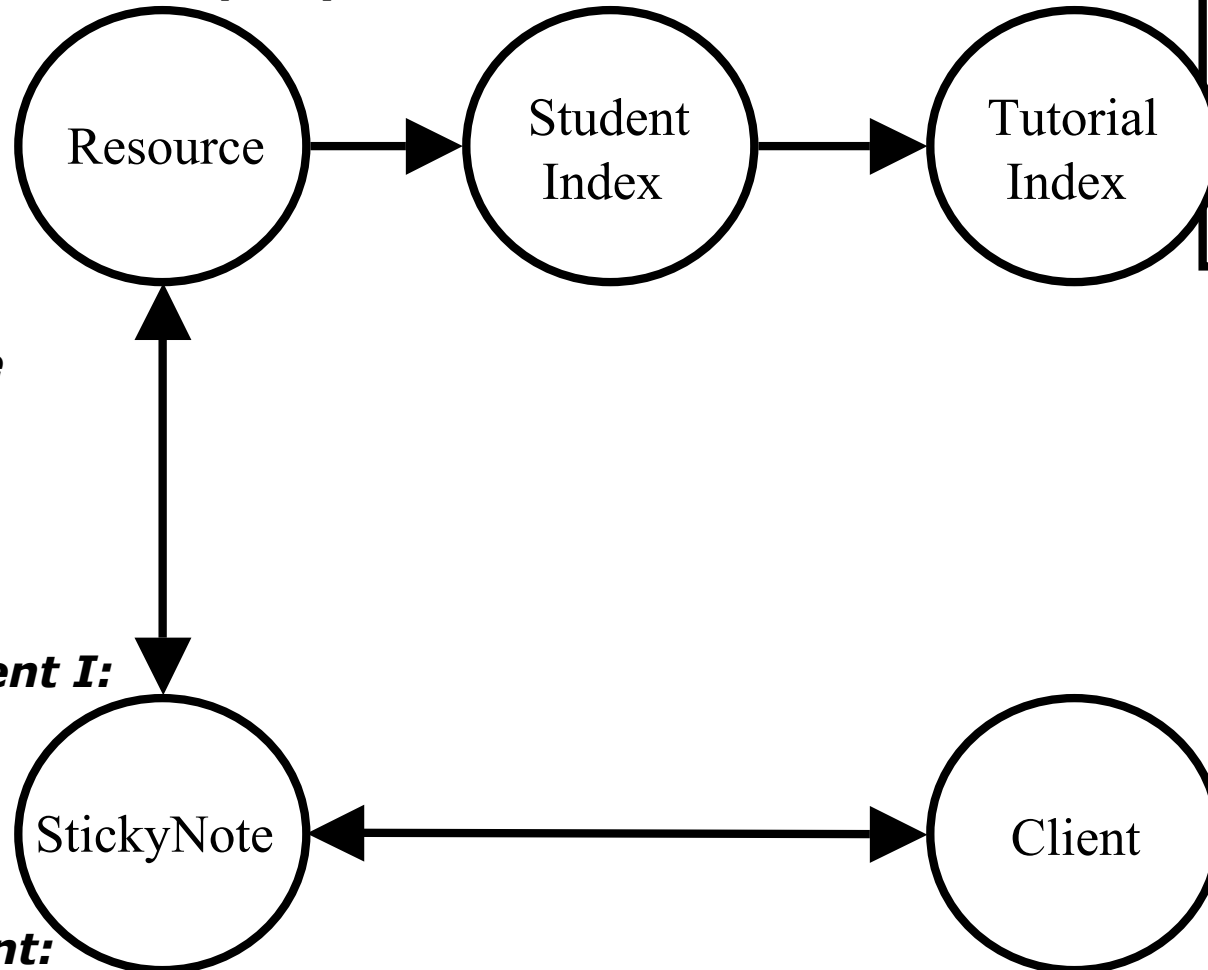**4. *State Management II:* Add a Resource Property**

5. *Discovery:* Find a Resource

6. *Virtual Organization:* Register with a community index
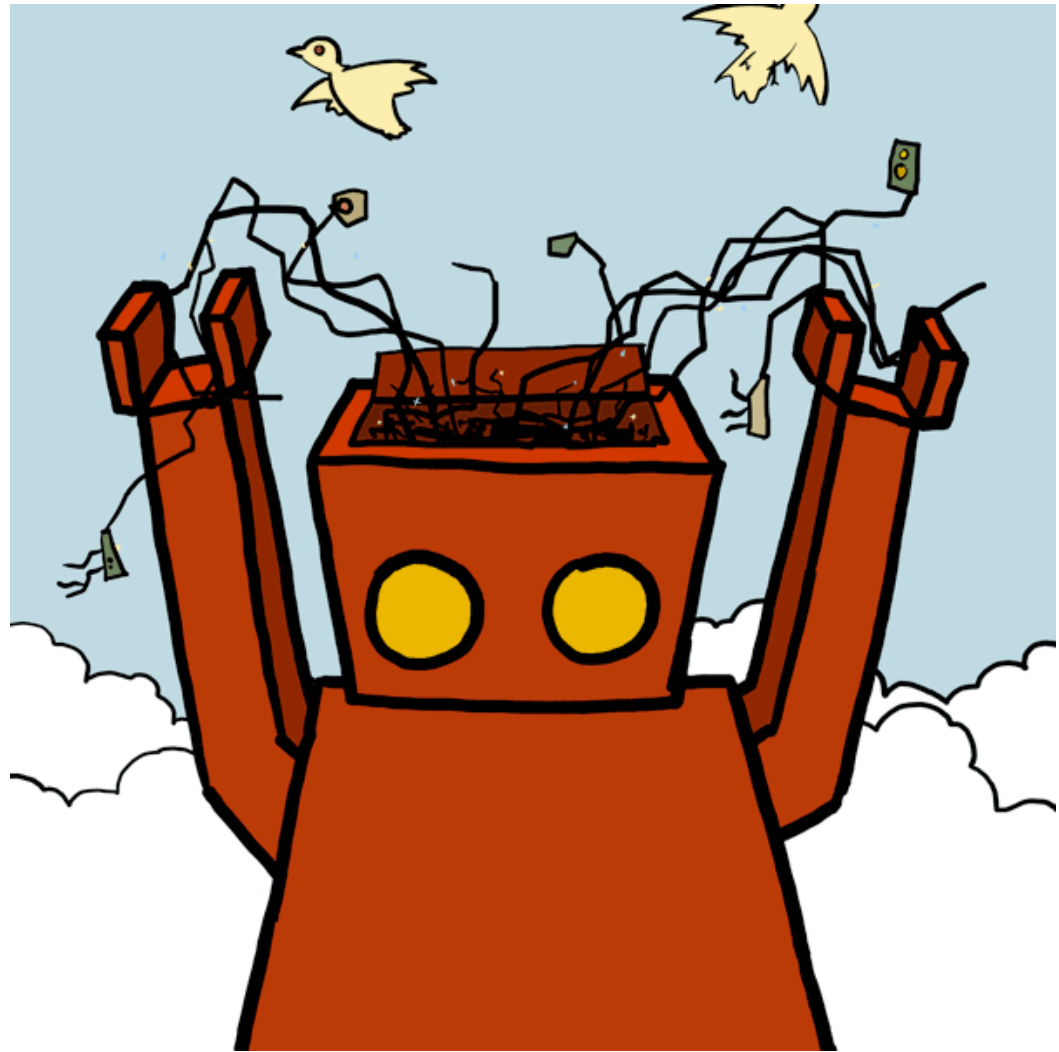
Resource → Student Index → Tutorial Index

3. *Lifetime Mgmt I:* Destroy Resources

2. *State Management I:* Create Resources

1. *Deployment:* Stand up a StickyNote service

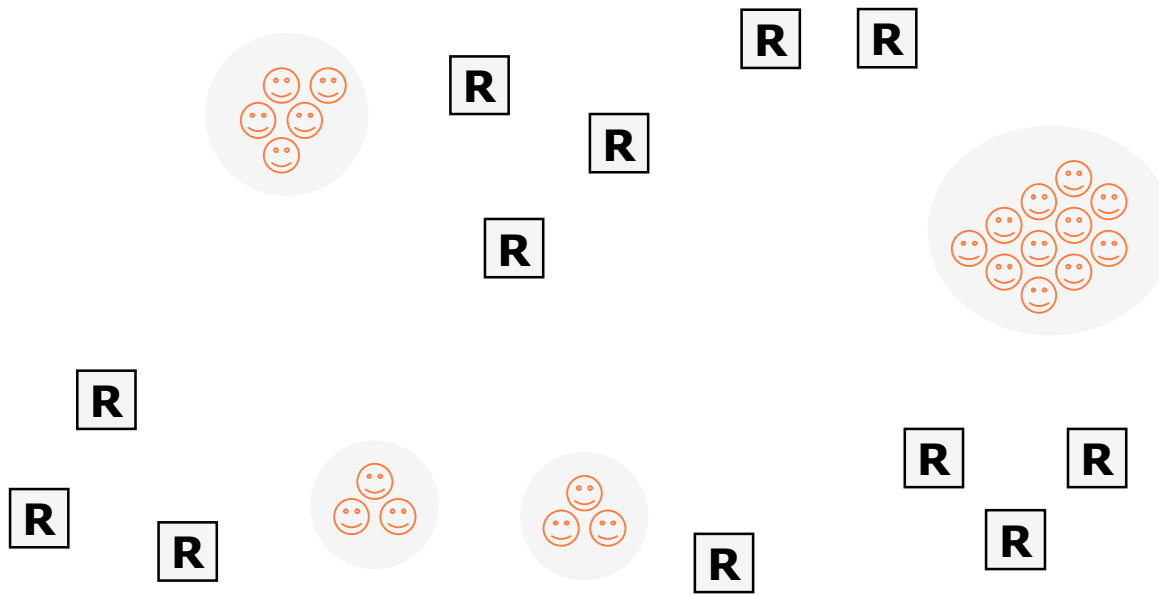StickyNote ↔ Client

Concepts

Virtual Organizations

- There is a structure that underlies many Grid problem spaces

- The structure is referred to as a Virtual Organization

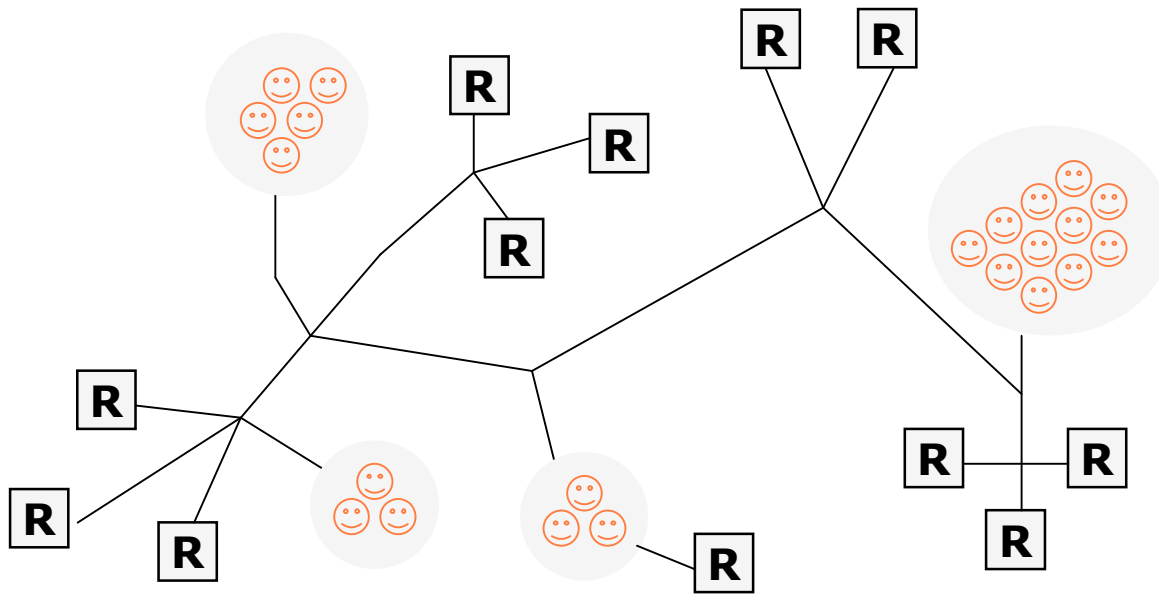- Virtual Organizations drive many of the requirements for Grid technology

Virtual Organizations
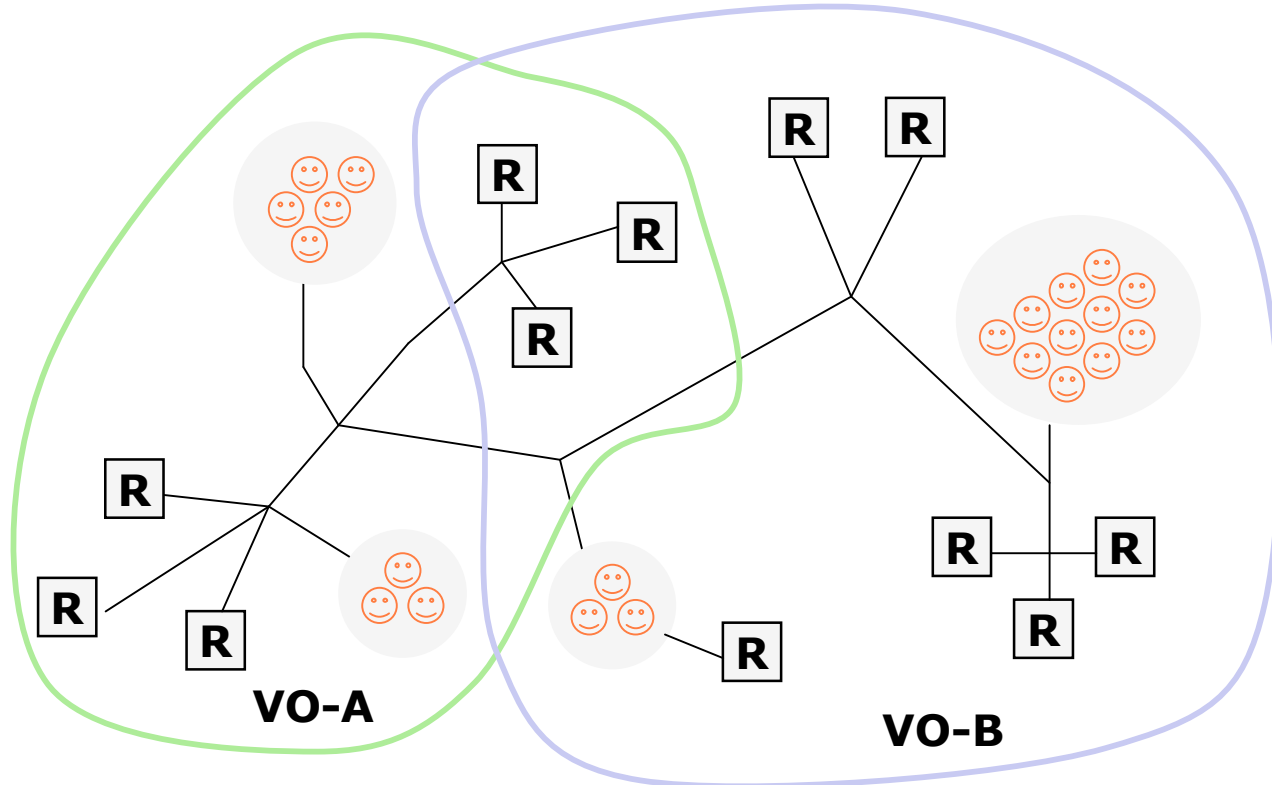
- Distributed resources and people

Virtual Organizations

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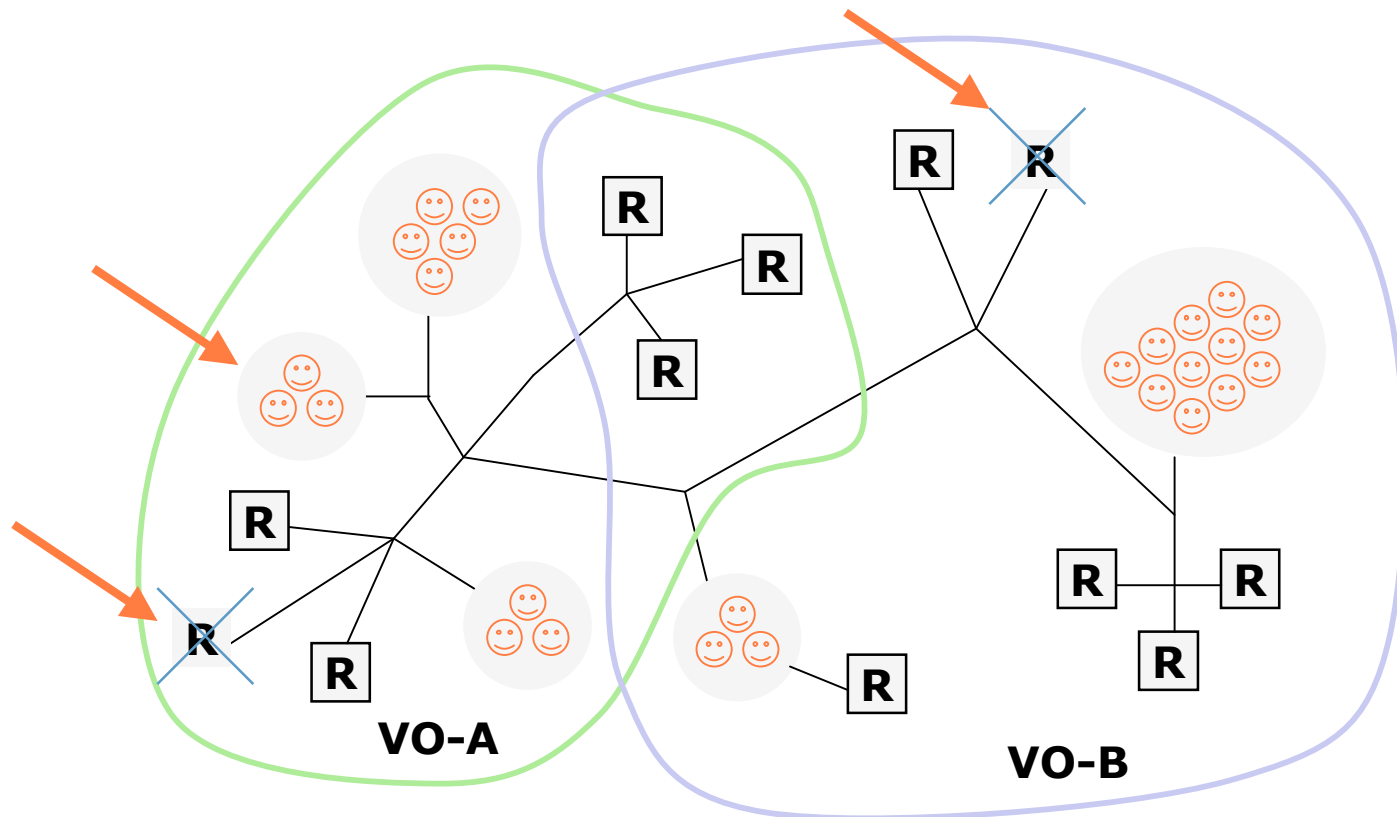- Linked by networks, crossing administrative domains

Virtual Organizations

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

VO-A

VO-B

Virtual Organizations

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

VO-A

VO-B

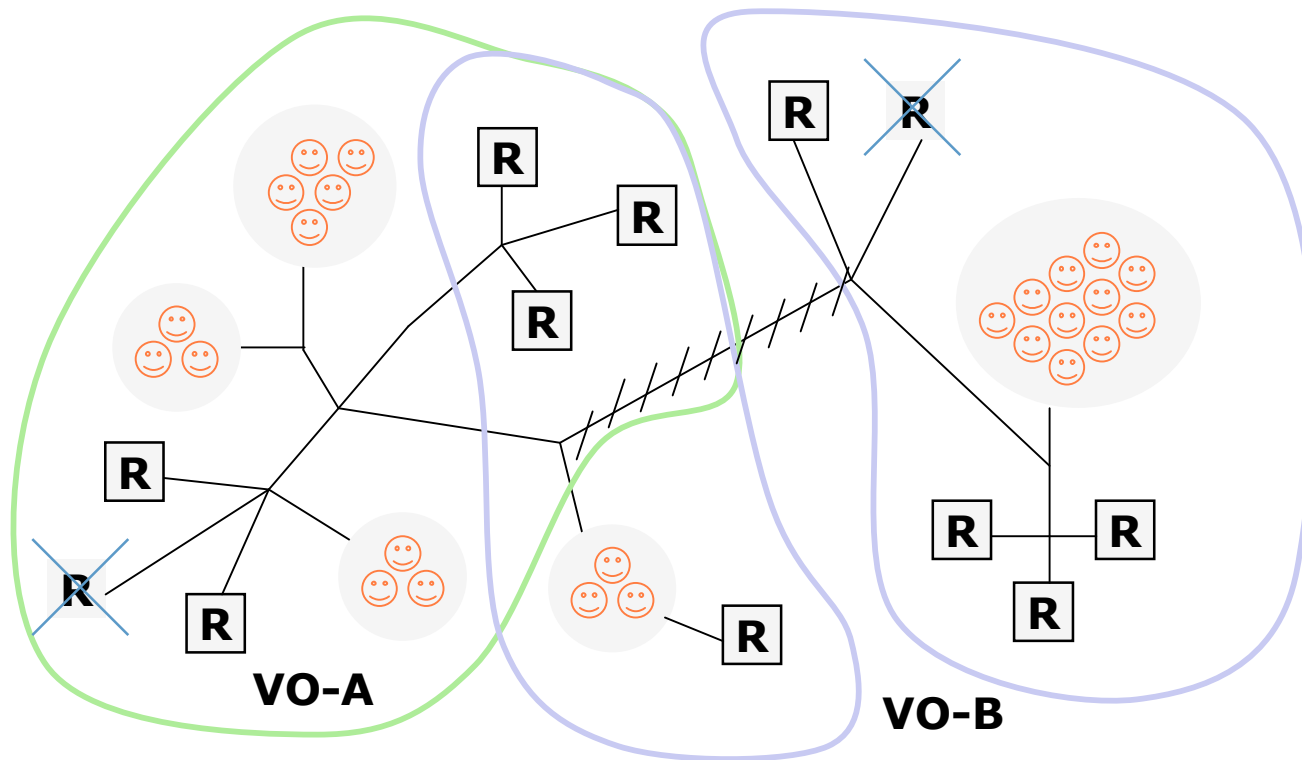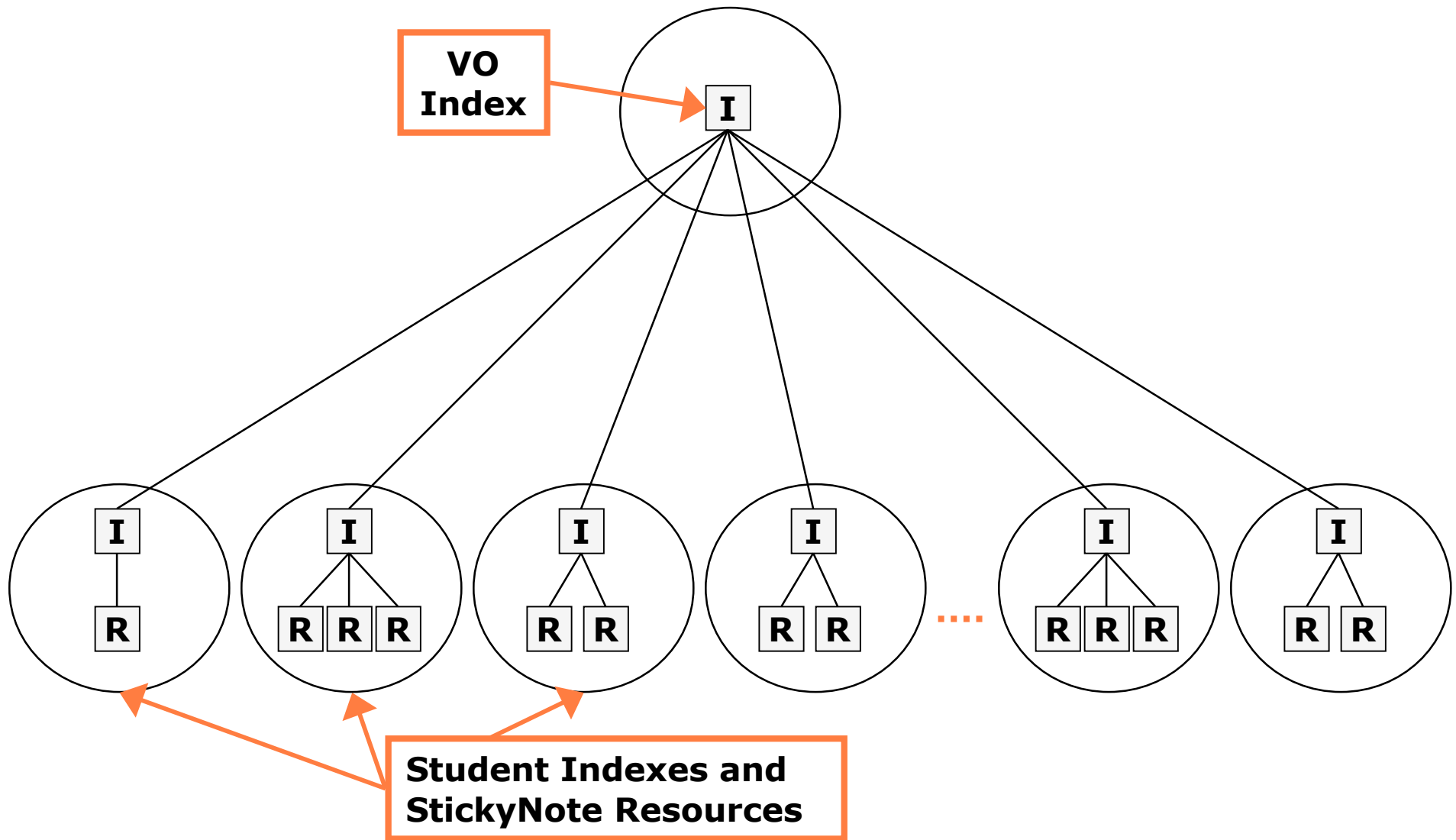# Virtual Organizations

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

VO-A

VO-B

Tutorial VO

the globus toolkit®
www.globustoolkit.org

VO Index

I

Student Indexes and StickyNote Resources

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
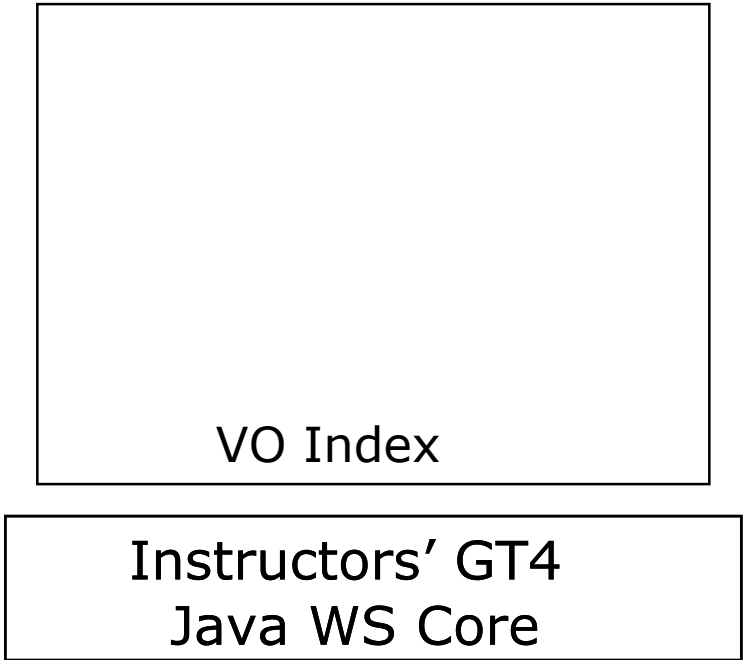 - A copy of each note's Resource Properties will be cached in a single place

Implementation details

Registering to the Index

- We configure the local Index to register with the VO Index

- Because our local index is registered to the VO Index, the local information will propagate
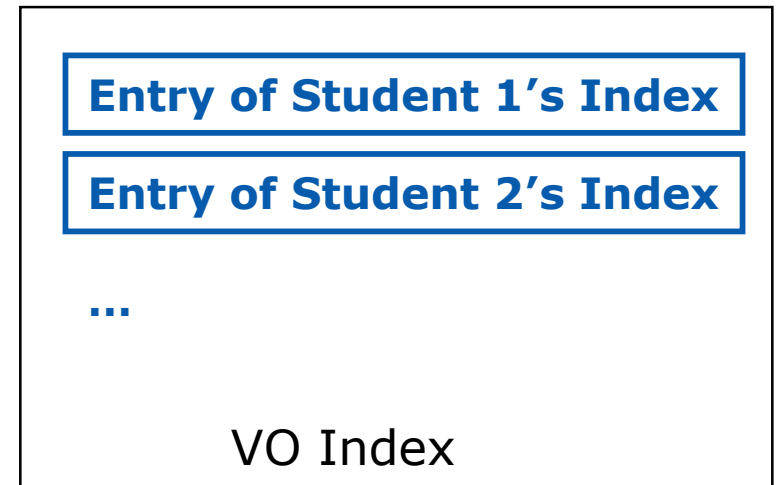
Resource Creation Runtime Revisited

VO Index

Instructors' GT4
Java WS Core

Resource Creation Runtime Revisited

create-note

StickyNote service	Student 1's Index	VO Index
		Entry of Student 1's Index
		Entry of Student 2's 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

VO Index

Entry of Student 1's Index

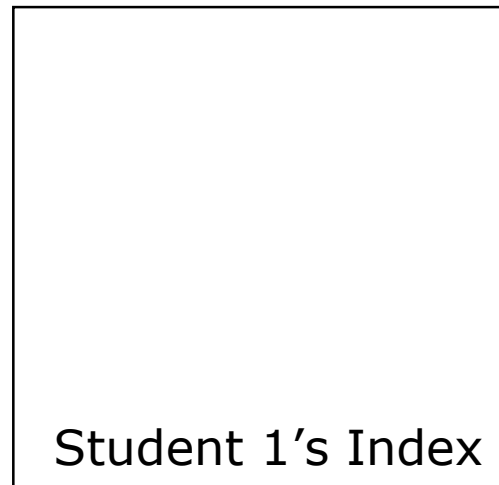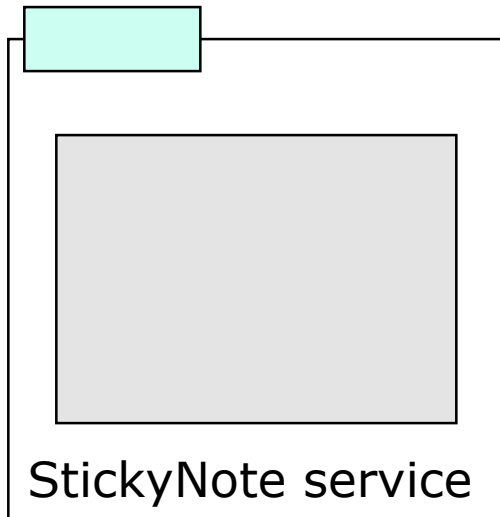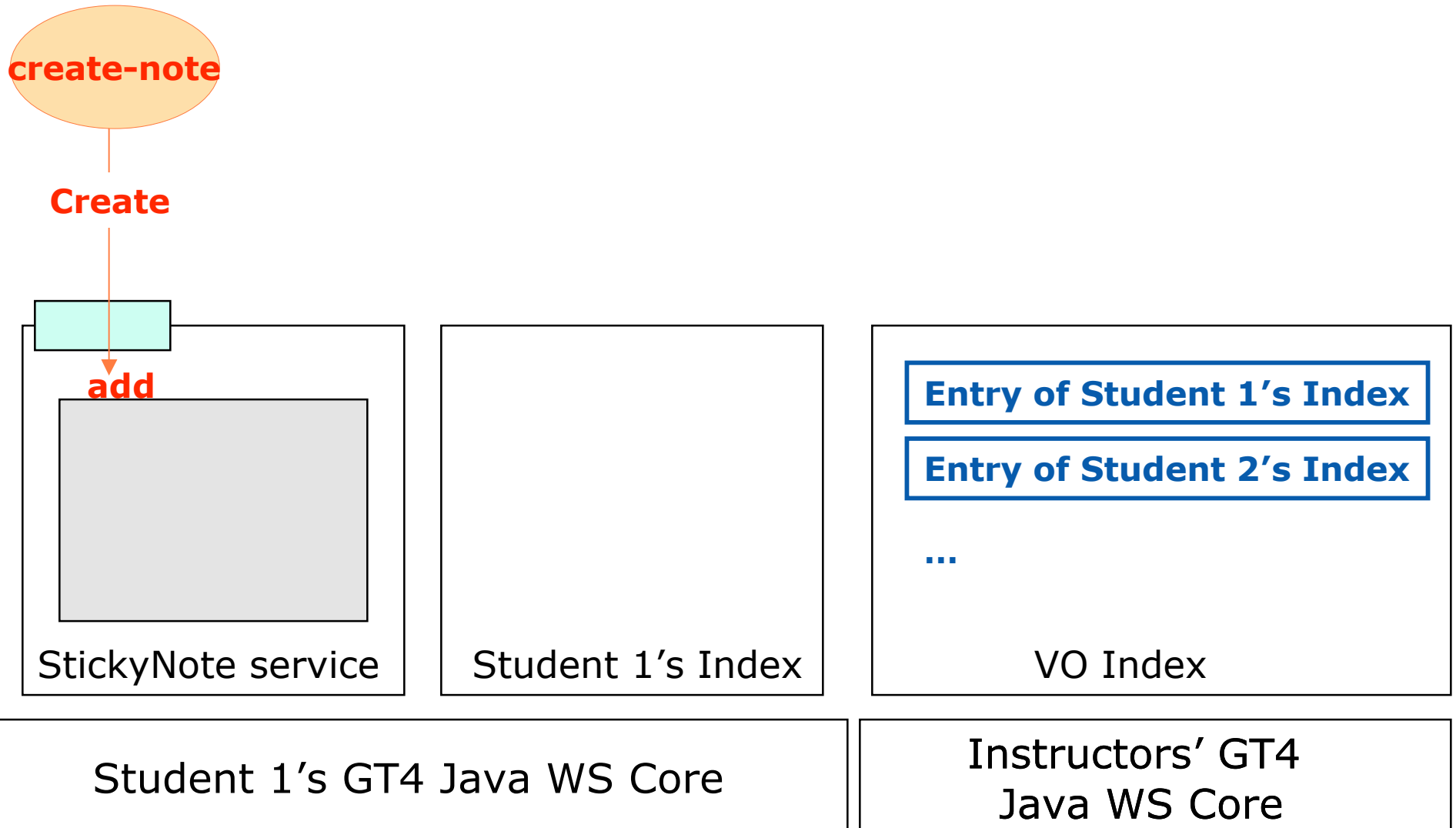**Entry of Student 2's Index**

...

Student 1's GT4 Java WS Core

Instructors' GT4
Java WS Core

Resource Creation Runtime Revisited

create-note

StickyNote service	Student 1's Index	VO Index
R1	Entry of R1	Entry of Student 1's Index Entry of Student 2's Index ...

Student 1's GT4 Java WS Core	Instructors' GT4 Java WS Core
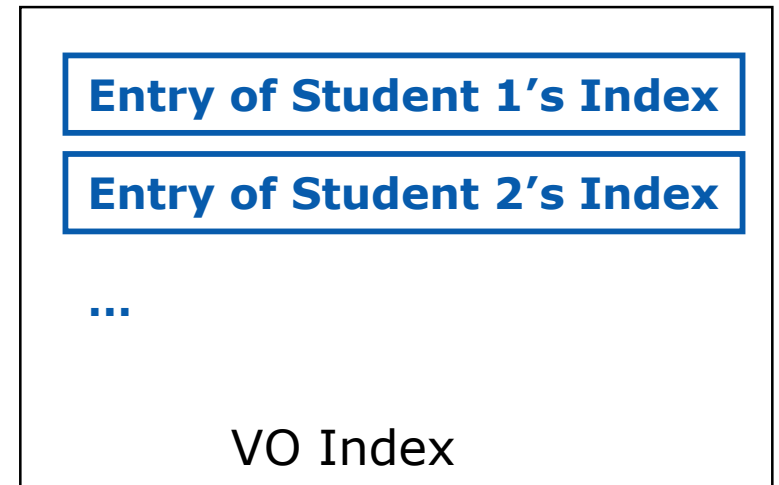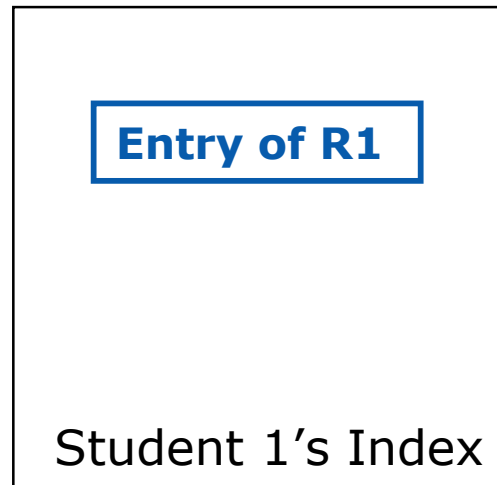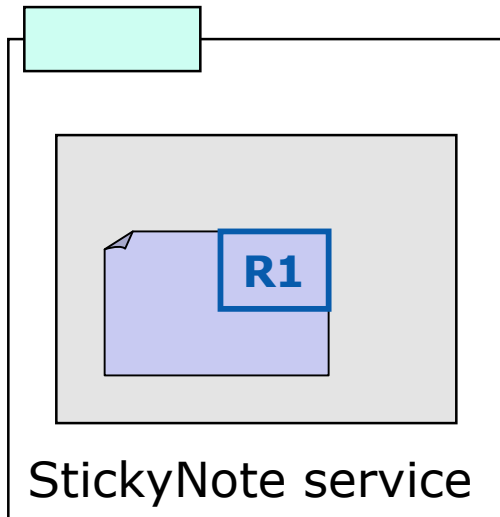
Resource Creation Runtime Revisited

create-note

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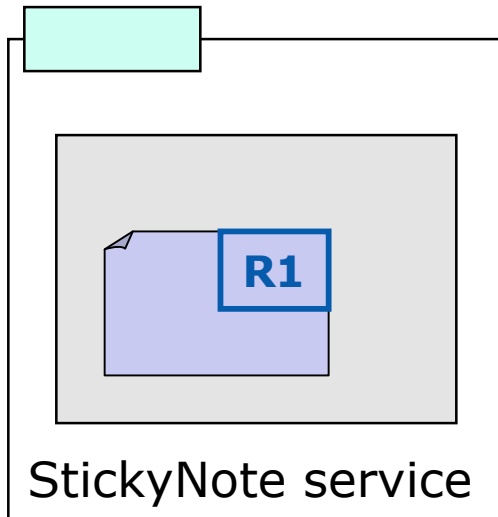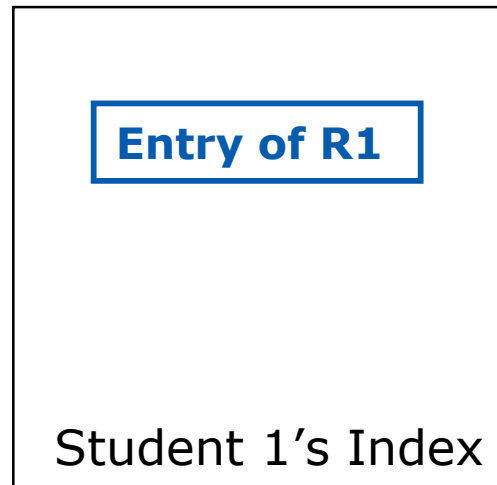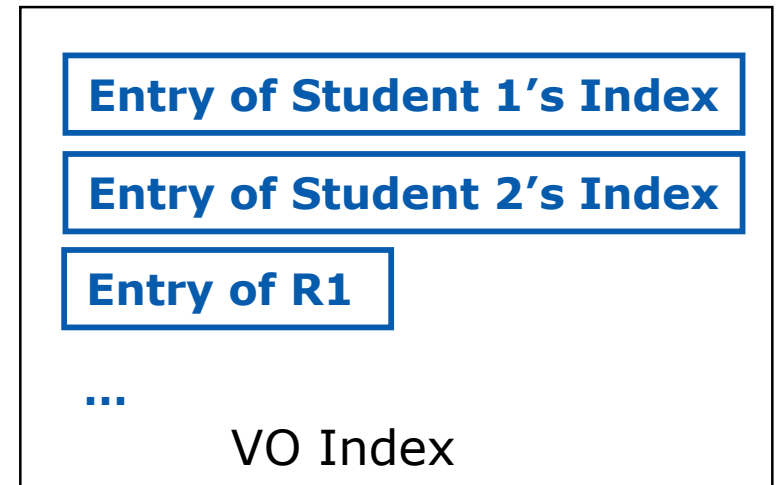
Student 1's GT4 Java WS Core	Instructors' GT4 Java WS Core

Hands-on Exercises

Student Notes, Chapter 6

Exercise 6 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

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. Discovery: Find a Resource
  - 6. Building a VO: Register with a Community Index
  - **7. Lifetime Management Part II: Lease-based Model**
  - 8. Notification: Resource as Notification Producer

- Overview of Tools for GT4 Service Developers
- Tutorial Summary
- Ideas for Further Work

Exercise 7: Leased-Based Lifetime Mgmt

4. *State Management II:* Add a Resource Property

5. *Discovery:* Find a Resource

6. *Virtual Organization:* Register with a community index

7. *Lifetime Mgmt II:* Lease-Based Model

Resource

Student Index

Tutorial Index

3. *Lifetime Mgmt I:* Destroy Resources

2. *State Management I:* Create Resources

1. *Deployment:* Stand up a StickyNote service

StickyNote

Client

the globus toolkit®
www.globustoolkit.org
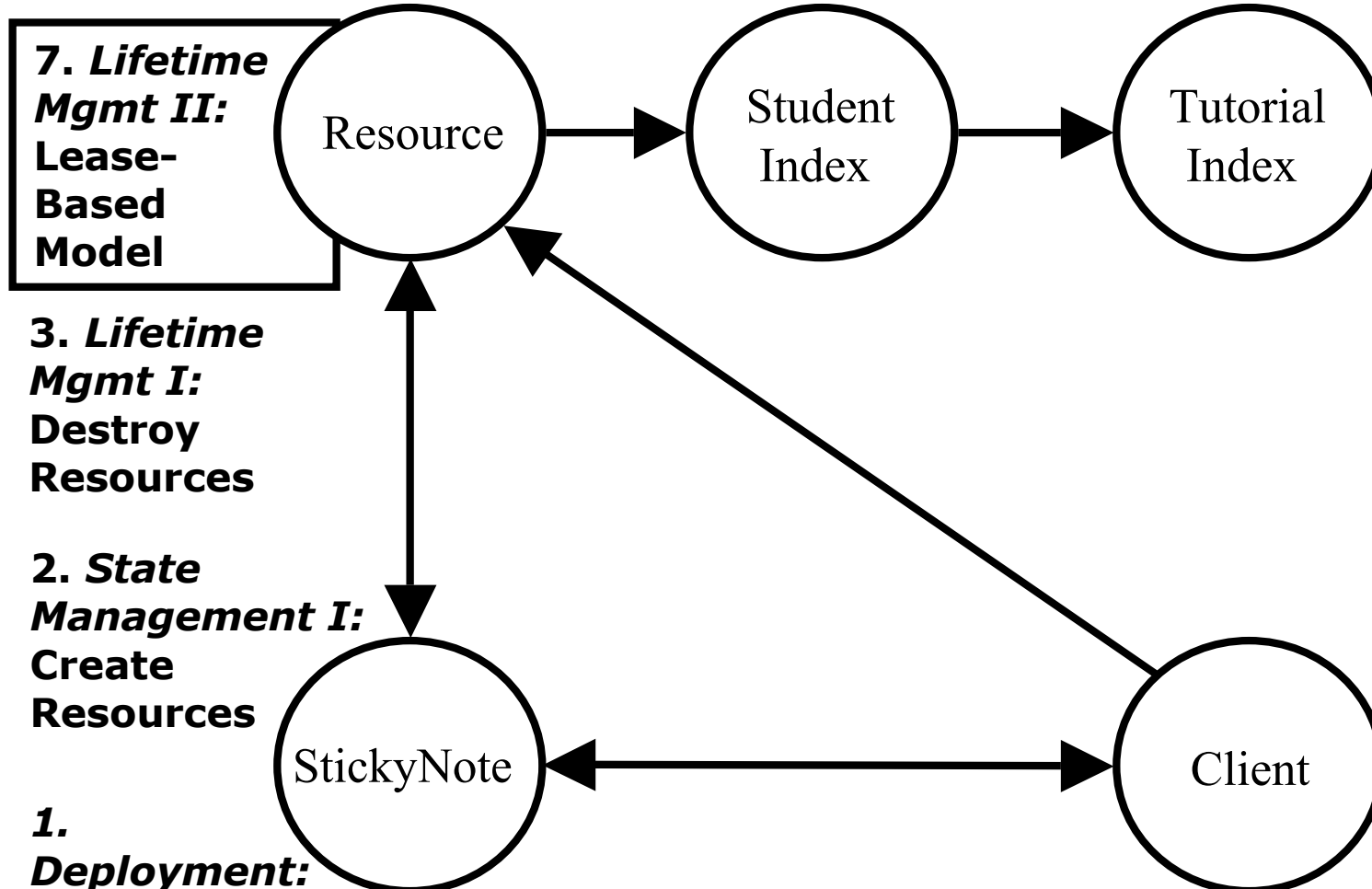
Concepts
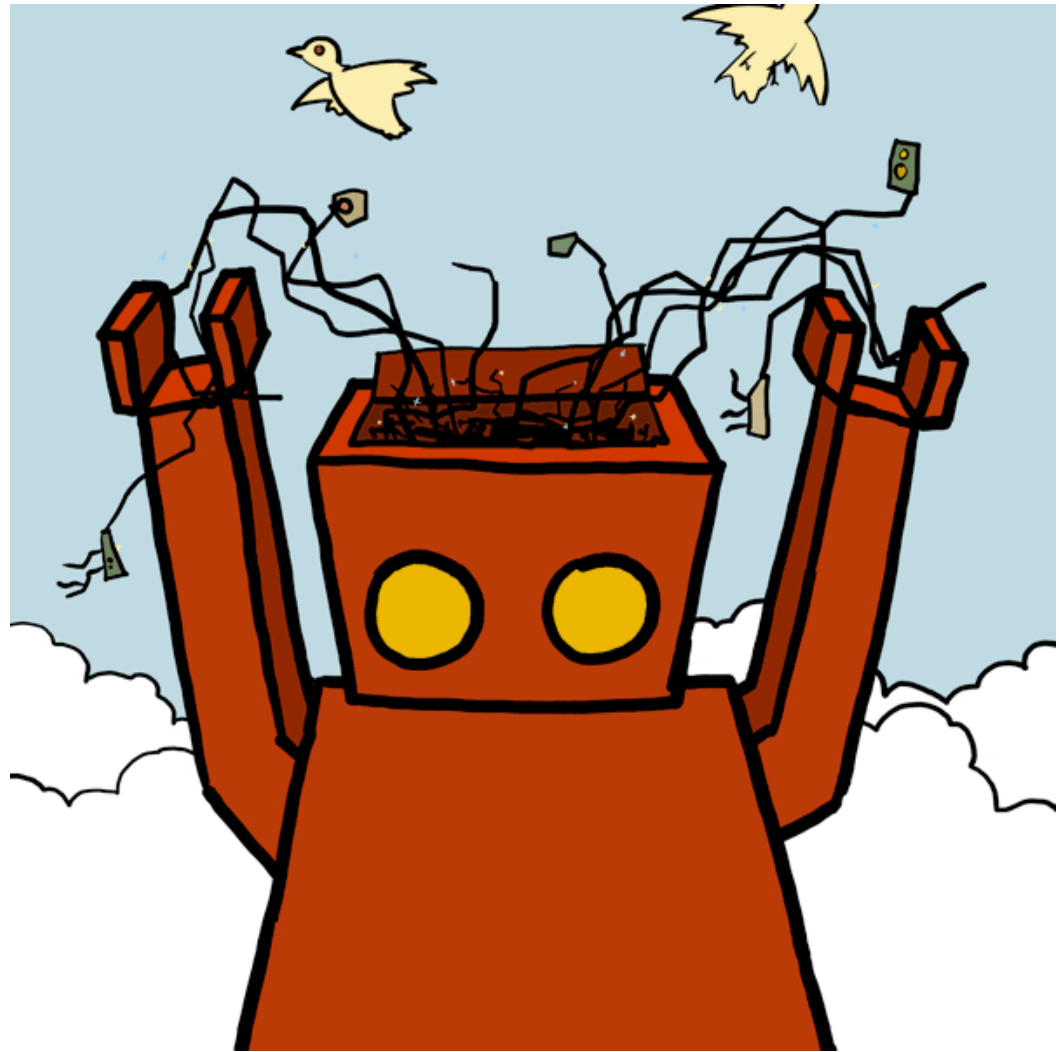
Lease-Based Lifetime Management

- 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 are removed unless kept alive by interested parties

Lease-Based Lifetime Management

- To illustrate lease-based lifetime management, we will experiment with scheduled destruction of sticky notes

- Under scheduled destruction, when the lifetime lease expires, the resource will be destroyed

- Interested parties must renew the lease

Implementation details

Lease-Based
Lifetime Management Runtime

create-note

Create

StickyNote service

GT4 Java WS Core
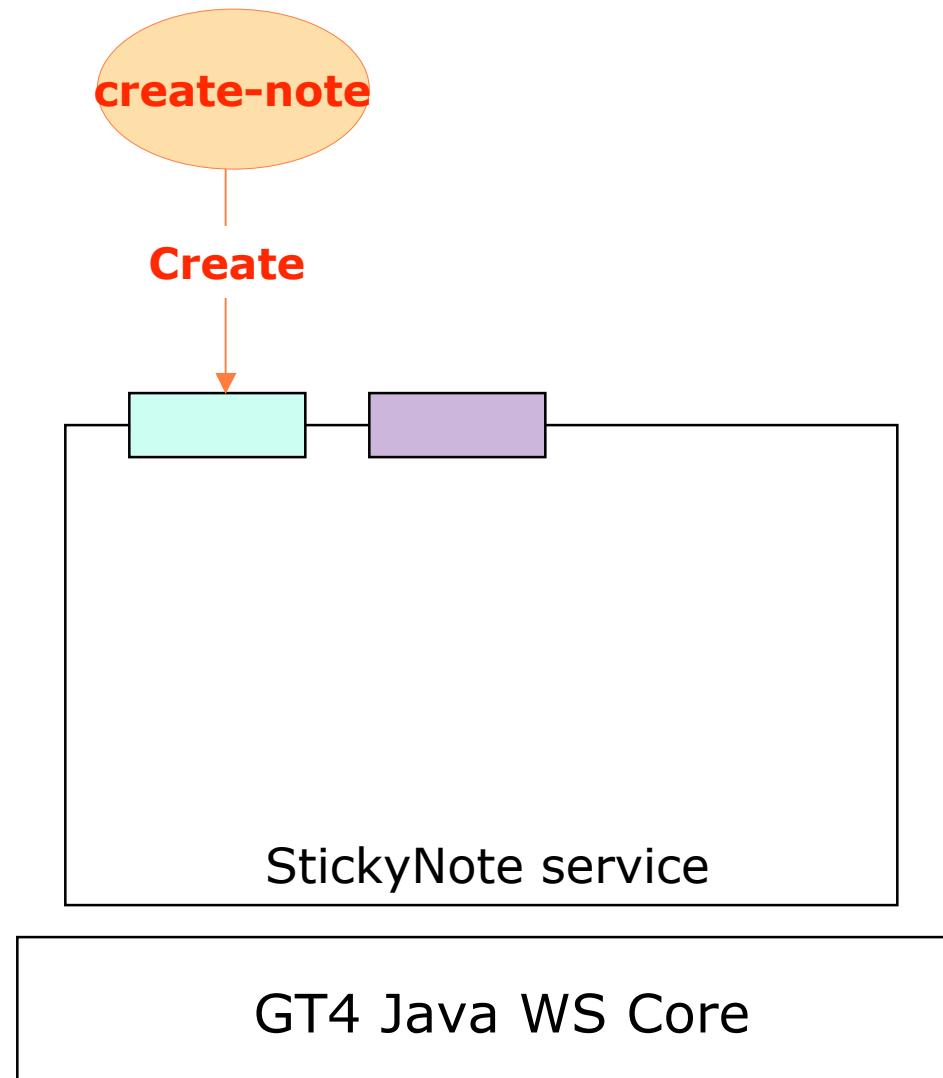
Lease-Based
Lifetime Management Runtime

Lease-Based Lifetime Management Runtime

wsrf-set-termination-time

SetTerminationTime

Termination Time RP

StickyNote service

GT4 Java WS Core
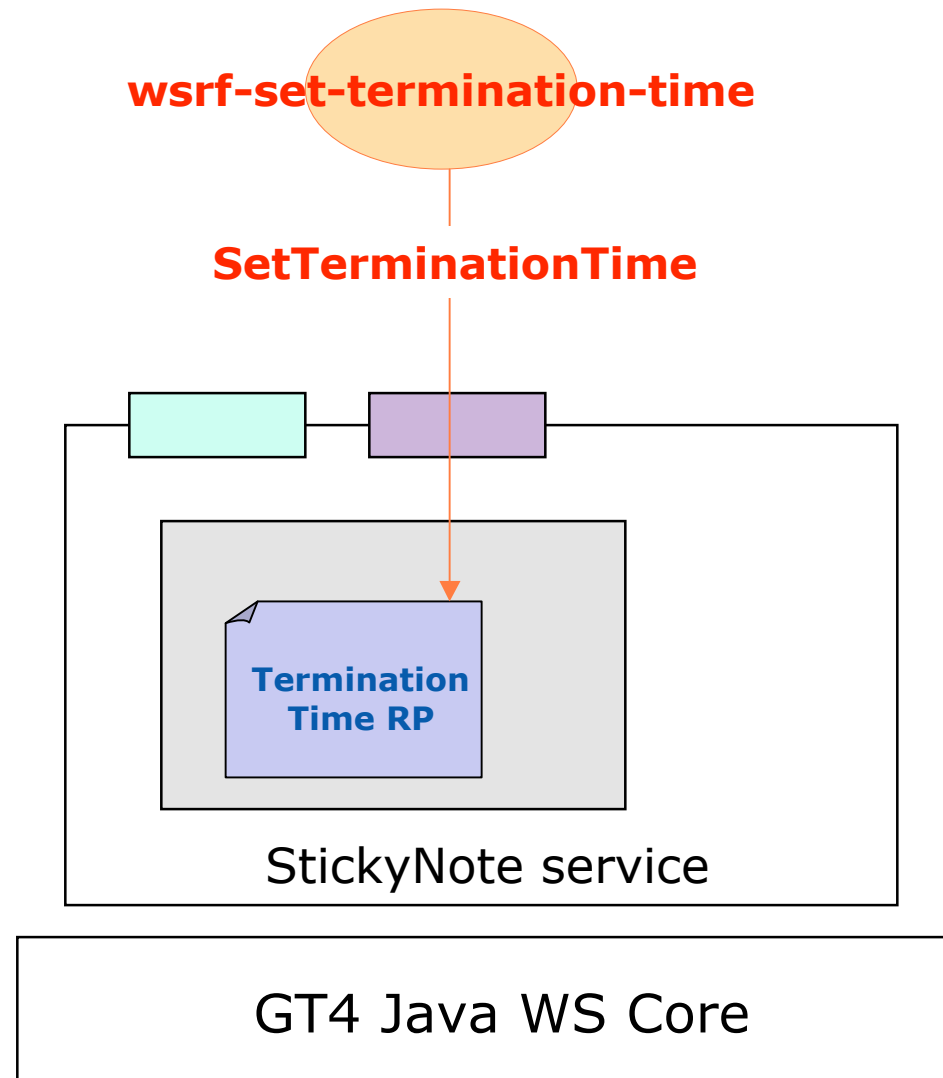
Lease-Based
Lifetime Management Runtime

Example of scheduled destruction:
The StickyNote resource goes away
if its lifetime lease is allowed to
expire

StickyNote service

GT4 Java WS Core
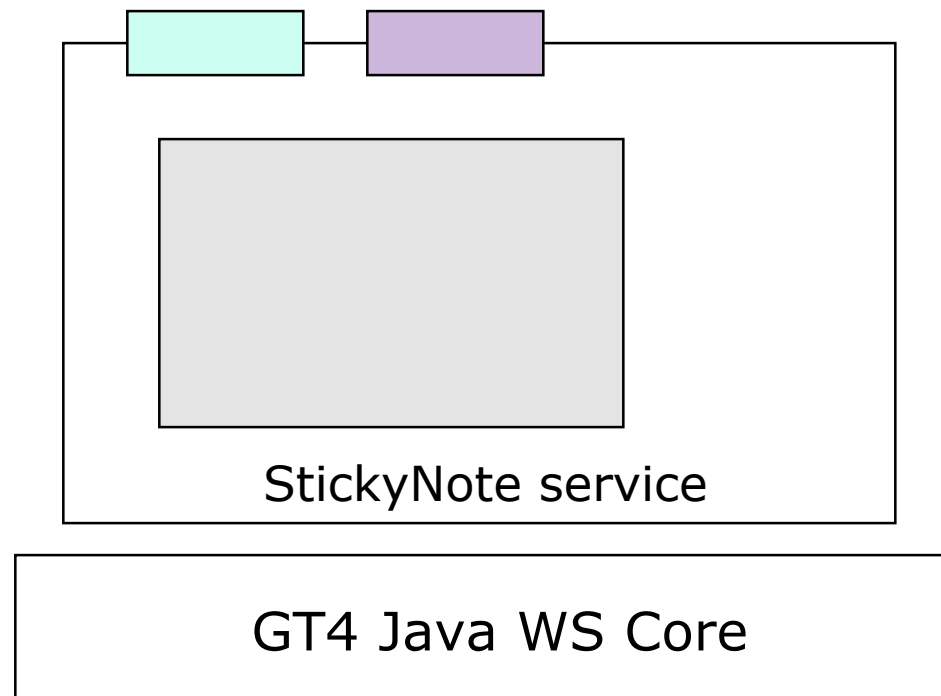
Adding Support for Scheduled Destruction

- Add a SetTerminationTime operation to the WSDL

- Add the ResourceLifetime RP

- Add the SetTerminationTime operation provider

Hands-on Exercises

Student Notes, Chapter 7

ISSGC'05: How to Build a Service Using GT4

Exercise 7 Review

- Resources can have a termination time

- The lifetime of a Resource is controlled by its termination time

- Under the lease-based model resources must be kept alive by interested parties

  - When the lifetimes are allowed to expire, the resources will be garbage collected by the container

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. Discovery: Find a Resource
  - 6. Building a VO: Register with a Community Index
  - 7. Lifetime Management Part II: Lease-based Model
  - **8. Notification: Resource as Notification Producer**

- Overview of Tools for GT4 Service Developers
- Tutorial Summary
- Ideas for Further Work

Exercise 8: Notification

4. *State Management II:* **Add a Resource Property**

5. *Discovery:* **Find a Resource**

6. *Virtual Organization:* **Register with a community index**

7. *Lifetime Mgmt II:* **Lease-Based Model**

Resource

Student Index

Tutorial Index

3. *Lifetime Mgmt I:* **Destroy Resources**

8. *Notification:* **Resource as Notification producer**

2. *State Management I:* **Create Resources**

StickyNote

Client

1. *Deployment:* **Stand up a StickyNote service**
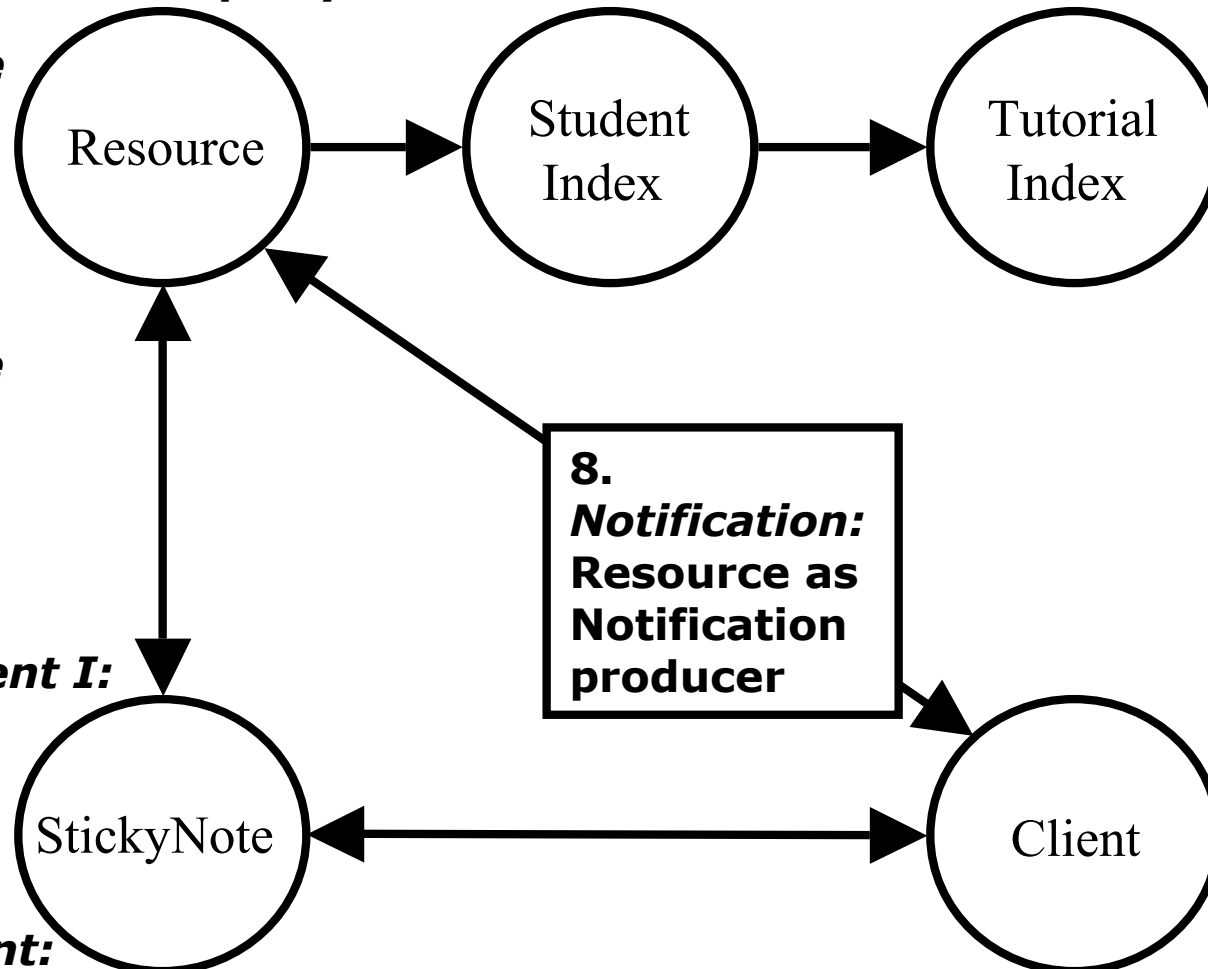
Concepts
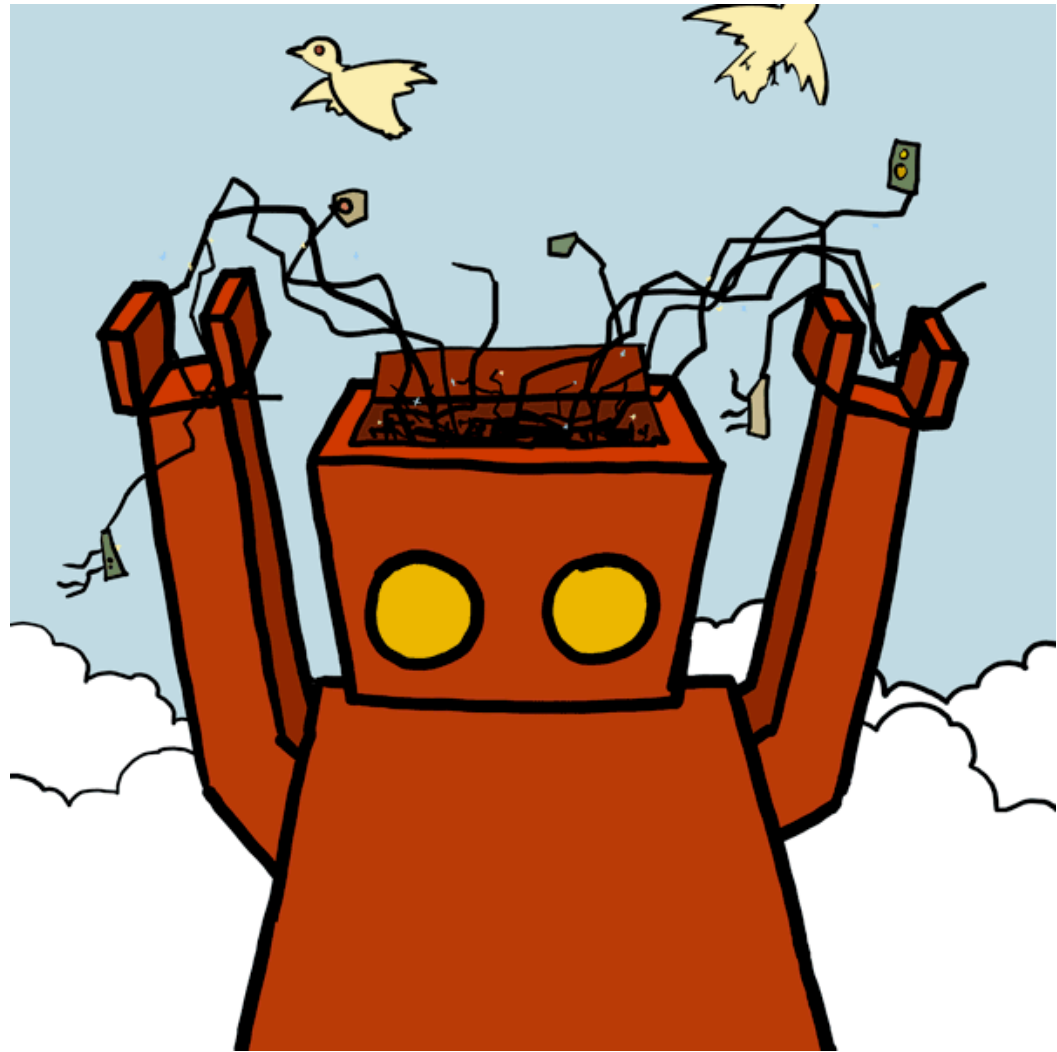
Notifications

- Right now we have the ability to make a one-off query to retrieve RP values
  - Show-note command
  - "pull mode"

- GT4 also supports a "push mode" model in which notifications are received when the RP value 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 subscription resource

Implementation details

Subscription and Notification

the globus toolkit®
www.globustoolkit.org

subscribe

Topic

NotificationConsumer Resource

NotificationProducer service

Subscription service

NotificationConsumer service

Producer GT4 Java WS Core

Consumer GT4 Java WS Core

Subscription and Notification

Topic

Subscription
Resource

NotificationConsumer
Resource

NotificationProducer
service

Subscription
service

NotificationConsumer
service

Producer GT4 Java WS Core

Consumer GT4
Java WS Core

Subscription and Notification

subscribe

Topic

Subscription Resource

NotificationConsumer Resource

NotificationProducer service

Subscription service

NotificationConsumer service

Producer GT4 Java WS Core

Consumer GT4 Java WS Core

Subscription and Notification

deliver

Topic	Subscription Resource	NotificationConsumer Resource
NotificationProducer service	Subscription service	NotificationConsumer service
Producer GT4 Java WS Core		Consumer GT4 Java WS Core
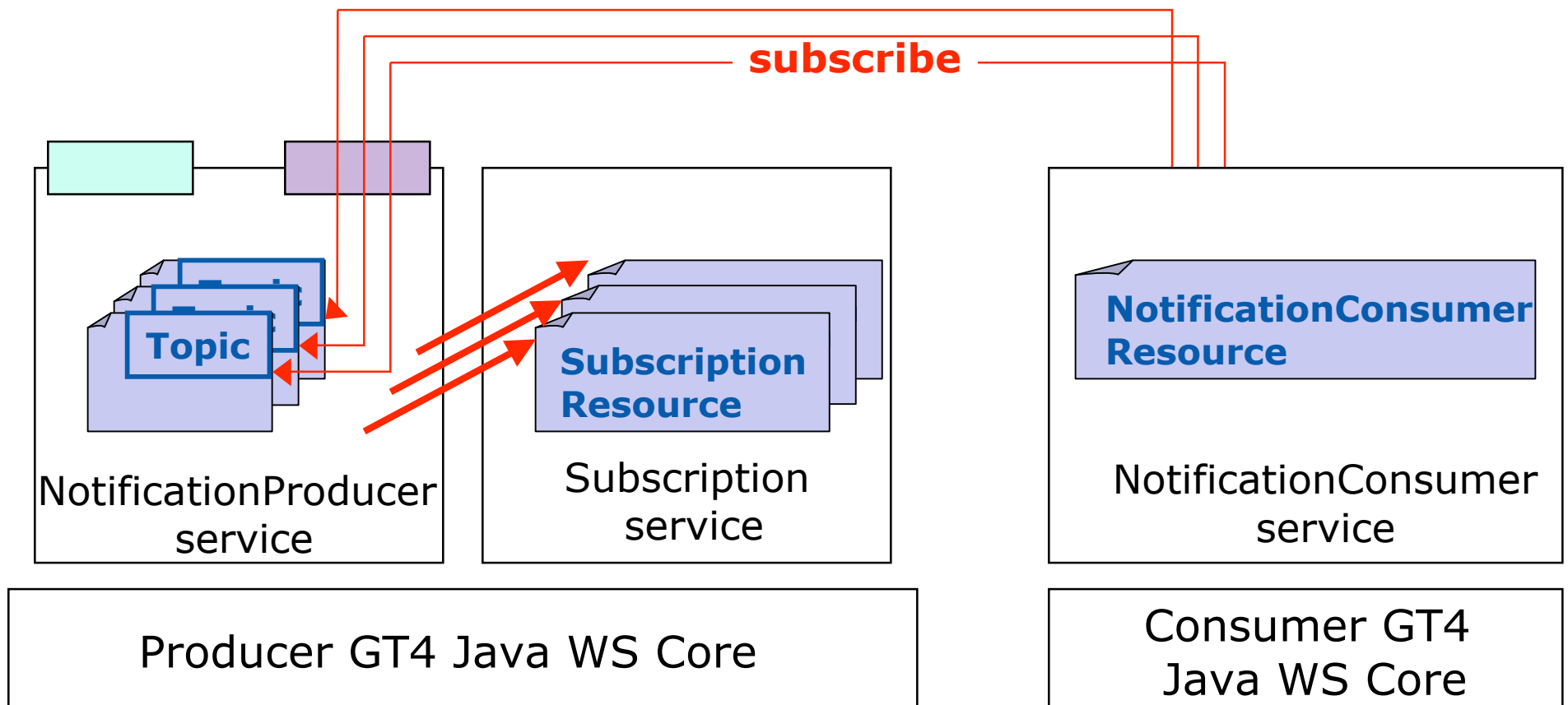
Sticky note notification
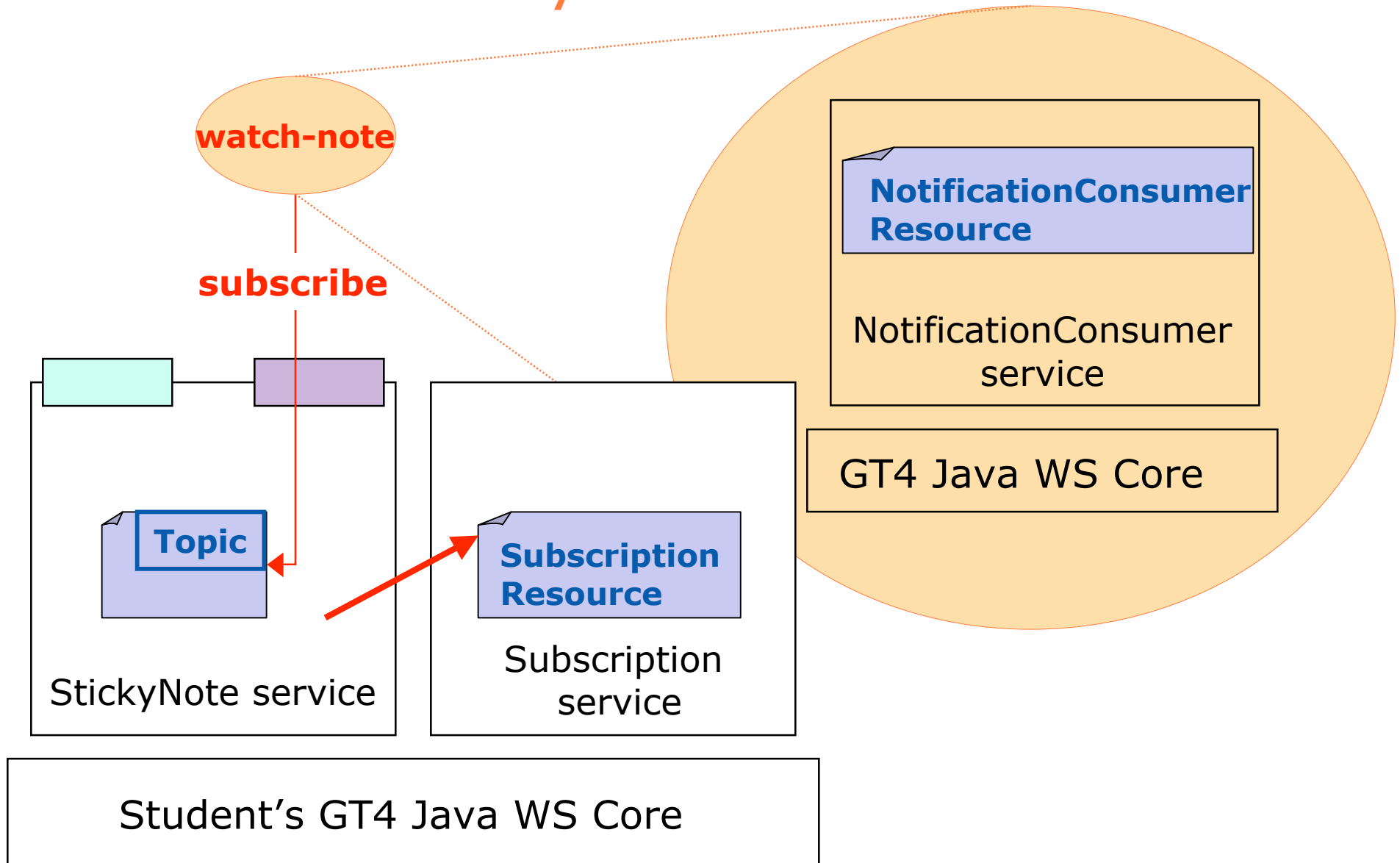
Hands-on Exercises

Student Notes, Chapter 8

Exercise 8 Review

- Notifications can deliver updates of RP changes

- The client must run 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. Discovery: Find a Resource

  - 6. Building a VO: Register with a Community Index

  - 7. Lifetime Management Part II: Lease-based Model

  - 8. Notification: Resource as Notification Producer

- **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>
```

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>
```
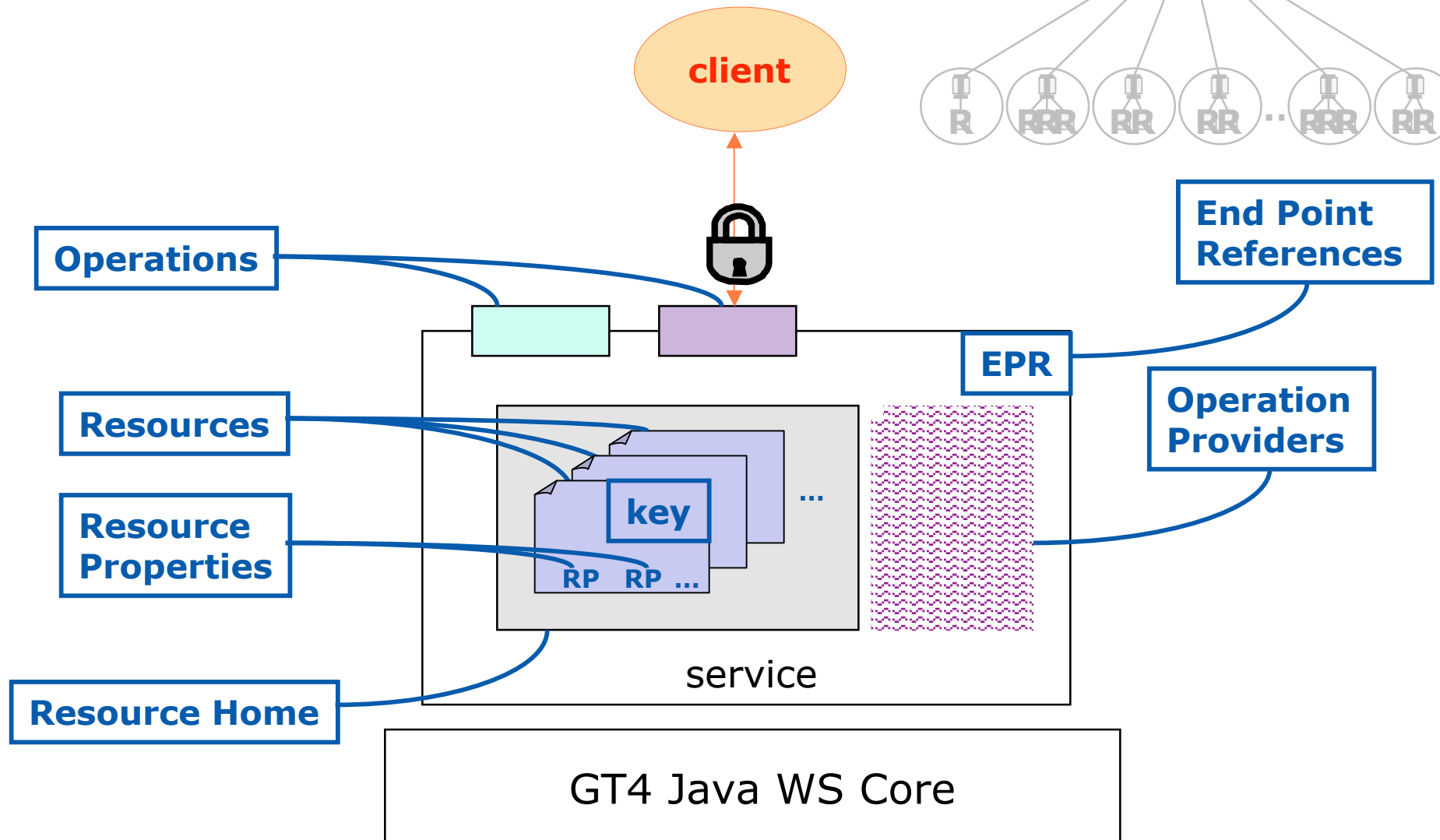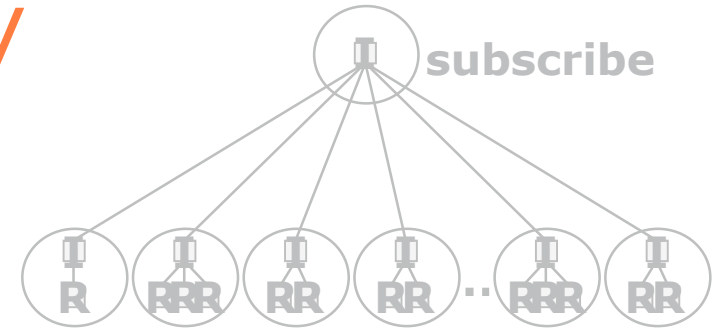
How to Build a Service Using GT4

- 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. Discovery: Find a Resource
  - 6. Building a VO: Register with a Community Index
  - 7. Lifetime Management Part II: Lease-based Model
  - 8. Notification: Resource as Notification Producer

- Overview of Tools for GT4 Service Developers
- **Tutorial Summary**
- Ideas for Further Work

Summary

Operations

Resources

Resource Properties

Resource Home

client

setTerminationTime

subscribe

End Point References

EPR

Operation Providers

key

RP   RP   ...

...

service

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. Discovery: Find a Resource
  - 6. Building a VO: Register with a Community Index
  - 7. Lifetime Management Part II: Lease-based Model
  - 8. Notification: Resource as Notification Producer

- 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

  - Join the **gt4-friends@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

  - Send mail to **info@globus.org**

Further Reading

Further details regarding this tutorial:
http://www.globus.org/toolkit/tutorials/BAS/

Globus Toolkit web:
http://www.globus.org/toolkit/

Globus Alliance web:
http://www.globus.org/

The Globus Documentation Project:
http://gdp.globus.org

WS-Resource Framework:
http://www.globus.org/wsrf/
On WSDL:
http://www.w3.org/TR/wsdl

Credits

the globus toolkit®
www.globustoolkit.org

- If you reuse "Build a Service" material please give credit to the Globus Alliance and the authors:
  - Rachana Ananthakrishnan, Charles Bacon, Lisa Childers, Ben Clifford, Jarek Gawor, Joe Insley

- All tutorial code is open source and covered by the Globus Toolkit Public License
  - http://www.globus.org/toolkit/download/license.html

- Robot pix created by Sam Brown
  - http://www.explodingdog.com

- This or previous iterations of the tutorial have been funded in part by the US Department of Energy, the National Science Foundation and IBM Corp

Questions regarding this tutorial should be directed to
Lisa Childers, childers@mcs.anl.gov