the globus toolkit™
www.globustoolkit.org

Introduction to GT3

The Globus Project™

Argonne National Laboratory
USC Information Sciences Institute

http://www.mcs.anl.gov/~childers/GT3IntroJune2003Print.ppt

Introduction to GT3

- **Background**
  - The Grid Problem
  - The Globus Approach
  - OGSA & OGSI
  - Globus Toolkit
- GT3 Architecture and Functionality: The Latest Refinement of the Globus Toolkit
  - Core
  - Base Services
  - User-Defined Services
  - Future Directions
- Installation and Administration
  - Installation
  - Configuration
  - Debugging
  - Support
- Important Things to Remember

A Story of Evolution

- Definition of Grid problem has been stable since original Globus Project proposal in 1995
  - Though we've gotten better at articulating it
- But our approach to its solution has evolved:
  - From APIs and custom protocols…
  - to standard protocols…
  - to Grid services (OGSA).
- Driven by experience implementing and deploying the Globus Toolkit, and building real applications with it

What is a Grid?

- We believe there are three key criteria:
  - Coordinates distributed resources …
  - using standard, open, general-purpose protocols and interfaces …
  - to deliver non-trivial qualities of service.
- What is not a Grid?
  - A cluster, a network attached storage device, a scientific instrument, a network, etc.
  - Each is an important component of a Grid, but by itself does not constitute a Grid

Challenging Technical Requirements

- Dynamic formation and management of virtual organizations
- Discovery & online negotiation of access to services: who, what, why, when, how
- Configuration of applications and systems able to deliver multiple qualities of service
- Autonomic management of distributed infrastructures, services, and applications
- Management of distributed state
- Open, extensible, evolvable infrastructure

The Globus Project™

Making Grid computing a reality (since 1996)

- Close collaboration with real Grid projects in science and industry
- The Globus Toolkit®: Open source software base for building Grid infrastructure and applications
- Development and promotion of standard Grid protocols to enable interoperability and shared infrastructure
- Development and promotion of standard Grid software APIs to enable portability and code sharing
- Global Grid Forum: We co-founded GGF to foster Grid standardization and community

the globus toolkit™
www.globustoolkit.org

From APIs & Custom Protocols, To Standard Protocols

API
Application Programming Interface

- A specification for a set of routines to facilitate application development
  - Refers to definition, not implementation
- Often language-specific (or IDL)
  - Routine name, number, order and type of arguments; mapping to language constructs
  - Behavior or function of routine
- Examples of APIs
  - GSS-API (security), MPI (message passing)

Network Protocol

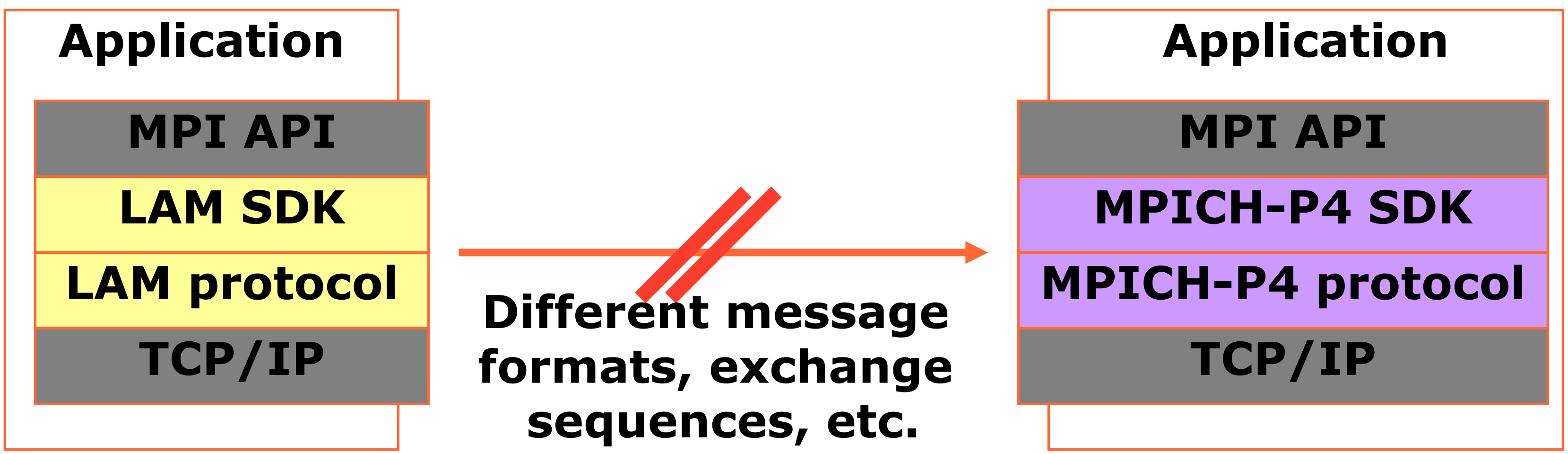
- A formal description of message formats and a set of rules for message exchange
  - Rules may define sequence of message exchanges
  - Protocol may define state-change in endpoint, e.g., file system state change
- Good protocols designed to do one thing
  - Protocols can be layered
- Examples of protocols
  - IP, TCP, TLS (was SSL), HTTP, Kerberos

A Protocol can have Multiple APIs

- TCP/IP APIs include BSD sockets, Winsock, System V streams, …
- The protocol provides **interoperability**: programs using different APIs can exchange information
- I don't need to know remote user's API

An API can have Multiple Protocols

- An API provides **portability**: any correct program compiles & runs on a platform
- Does not provide interoperability: all processes must link against same SDK
  - –E.g., MPICH and LAM versions of MPI



Initial Focus On APIs and Custom Protocols

- Primary concern was allowing Grid applications to be built quickly, in order to demonstrate feasibility
- Good development APIs and SDKs mattered most
- Protocols were a means to an end
  - We borrowed and extended standard protocols to make life easier (e.g. LDAP)
  - We defined custom protocols (e.g. GRAM)

But Focus Shifted To Protocols

- As demand grew, customers worried about:
  - compatibility between versions (i.e. Stop changing the protocols!)
  - independent implementations of some components (i.e. What are the protocols?)
- Ubiquitous adoption demands open, standard protocols
  - Internet and Web as guides
  - Enables innovation/competition on end points
  - Avoid product/vendor lock-in

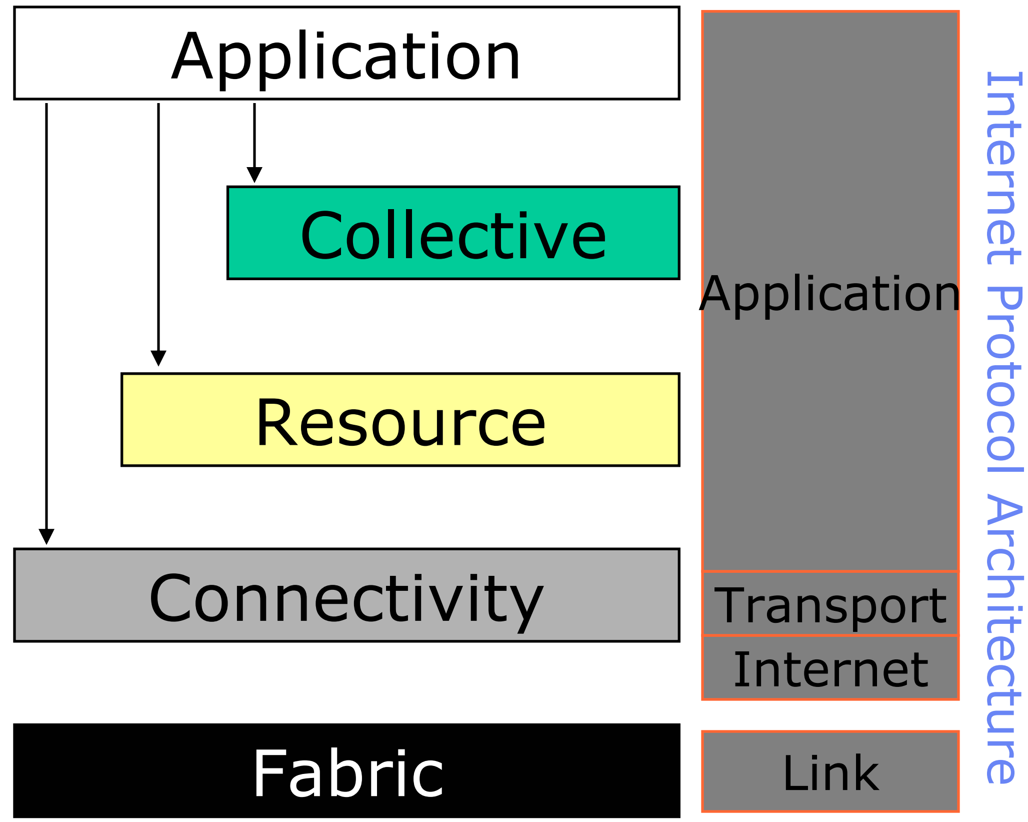
Layered Grid Architecture

"Coordinating multiple resources": ubiquitous infrastructure services, app-specific distributed services

"Sharing single resources": negotiating access, controlling use

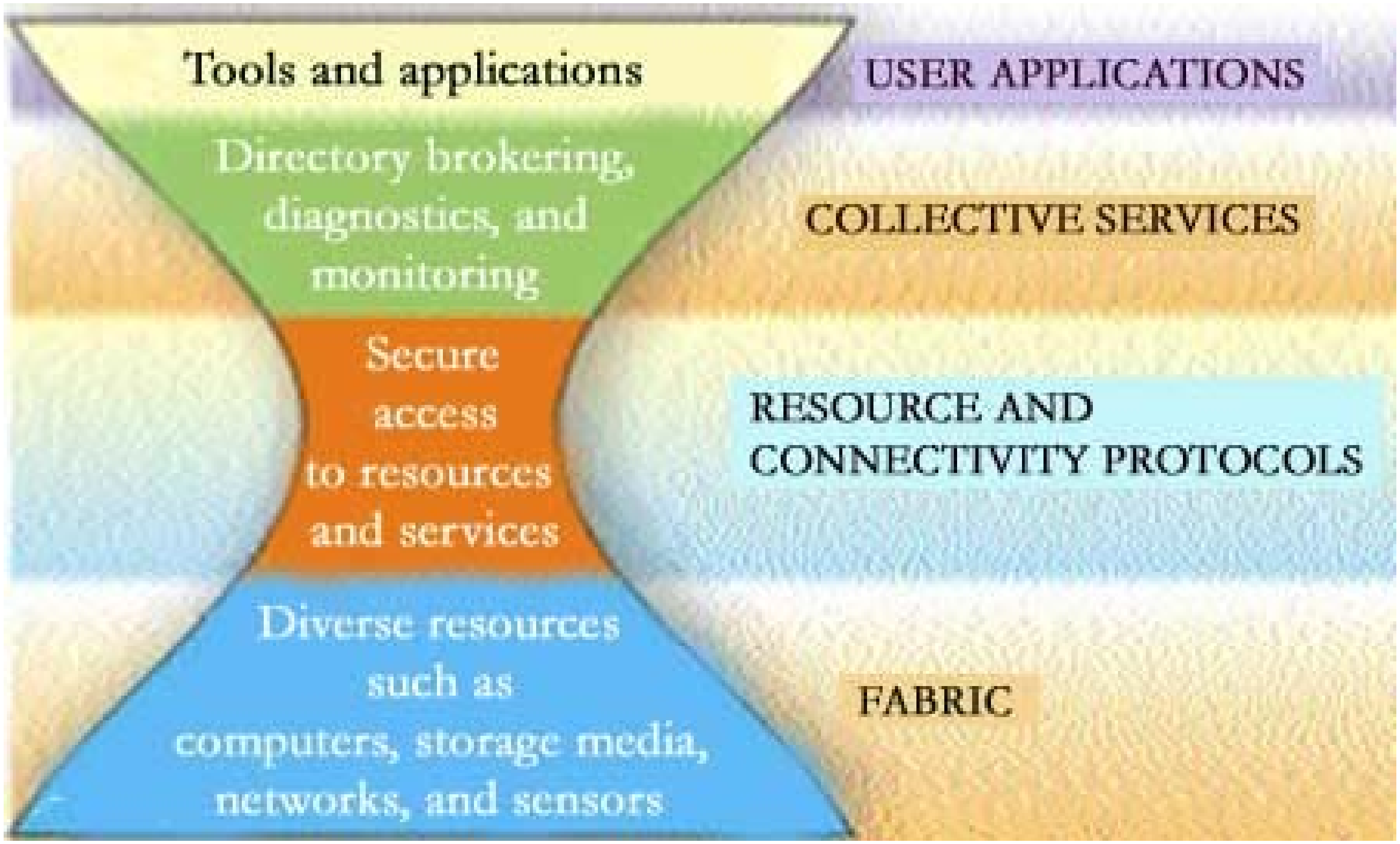
"Talking to things": communication (Internet protocols) & security

"Controlling things locally": Access to, & control of, resources

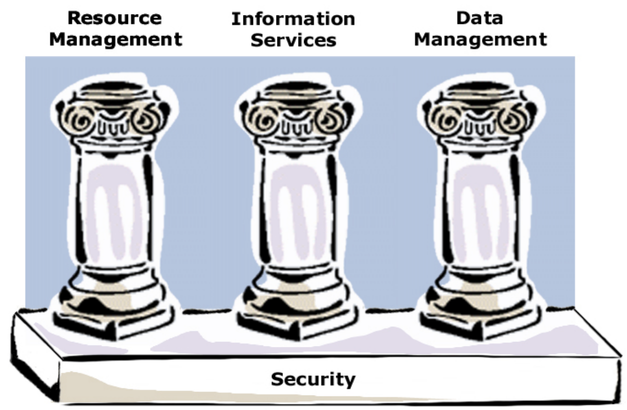


"The Anatomy of the Grid: Enabling Scalable Virtual Organizations", Foster, Kesselman, Tuecke, Intl Journal of High Performance Computing Applications, 15(3), 2001.

Layers of Grid Architecture



GT2 Key Protocols



- The Globus Toolkit v2 (GT2) centers around four key protocols
  - Connectivity layer:
    - *Security*: Grid Security Infrastructure (GSI)
  - Resource layer:
    - *Resource Management*: Grid Resource Allocation Management (GRAM)
    - *Information Services*: Grid Resource Information Protocol (GRIP)
    - *Data Transfer*: Grid File Transfer Protocol (GridFTP)
- Also key collective layer protocols
  - Info Services, Replica Management, etc.

Protocol Standards Efforts

- X.509 Proxy Certificate Profile
  - GGF & IETF
- GridFTP Protocol
  - GGF

From Standard Protocols, To Grid Services

But Along The Way…

- Heterogeneous protocol base was hurting us
- Increasing number of virtual services that needed to be managed
- Web services (WSDL, SOAP) appeared

Web Services

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

Transient Service Instances

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

Standard Interfaces & Behaviors: Four Interrelated Concepts

- Naming and bindings
  - Every service instance has a unique name, from which can discover supported bindings
- Lifecycle
  - Service instances created by factories
  - Destroyed explicitly or via soft state
- Information model
  - Service data associated with Grid service instances, operations for accessing this info
  - Basis for service introspection, monitoring, discovery
- Notification
  - Interfaces for registering existence, and delivering notifications of changes to service data

Grid Evolution: Open Grid Services Architecture

- Refactor Globus protocol suite to enable common base and expose key capabilities
- Service orientation to virtualize resources and unify resources/services/information
- Embrace key Web services technologies for standard IDL, leverage commercial efforts
- Result: standard interfaces & behaviors for distributed system management: the Grid service

OGSA Structure

- A standard substrate: the Grid service
  - OGSI = Open Grid Service Infrastructure
  - Standard interfaces and behaviors that address key distributed system issues
  - Much borrowed from GT abstractions
- … supports standard service specifications
  - Resource mgt, dbms, workflow, security, …
  - Target of current & planned GGF efforts
- … and arbitrary application-specific services based on these & other definitions

OGSI Grid Service Specification

- Defines WSDL conventions and GSDL extensions
  - For describing and structuring services
  - Working with W3C WSDL working group to drive GSDL extensions into WSDL
- Defines fundamental interfaces (using WSDL) and behaviors that define a Grid Service
  - A unifying framework for interoperability & establishment of total system properties

Globus Toolkit (GT)

- A software system addressing key technical problems in the development of Grid-enabled tools, services, and applications
  - Offer a modular set of orthogonal services
  - Middleware for building solutions, not turn-key
  - Enable *incremental* development of Grid-enabled tools and applications
  - Implement and inform Grid standards
  - Available under liberal open source license
  - Large community of developers & users
  - Multiple commercial support providers

Why Open Source is Important

- Leverages large body of code and experience
  - Efforts of a large e-Science community
- Encourages adoption of open standards
  - Reference implementation, community pressure
- Facilitates integration of new platforms
  - Port the implementation
- Allows vendors to focus on value add
  - Platforms, integration, higher-level services, turnkey applications, training, support

OGSA and the Globus Toolkit

- Technically, OGSA enables
  - Refactoring of protocols (GRAM, MDS, GridFTP), *while preserving all GT concepts/features*!
  - Integration with hosting environments: simplifying components, distribution, etc.
  - Greatly expanded standard service set
- Pragmatically, we are proceeding as follows
  - Develop open source OGSA implementation
    - Globus Toolkit 3.0; supports Globus Toolkit 2.0 APIs
  - Partnerships for service development
  - Also expect commercial value-adds

GT2 Evolution To GT3

- What happened to the GT2 key protocols?
  - Security: Adapting X.509 proxy certs to integrate with emerging WS standards
  - GRIP/LDAP: Abstractions integrated into OGSI as serviceData
  - GRAM: ManagedJobFactory and related service definitions
  - GridFTP: Unchanged in 3.0, but will evolve into OGSI-compliant service in 2004
- Also rendering collective services in terms of OGSI: RFT, RLS, etc.

GT Timeline

- GT 1.0: *1998*
  - GRAM, MDS
- GT 2.0: *2001*
  - GridFTP, packaging, reliability
- GT3 Technology Preview: *Apr-Dec 2002*
  - Tracking OGSI definition
- GT3.0 Alpha: *Jan 2003*
  - OGSI Base, GT2 functionality
- GT3.0 Production: *June 2003*
  - Tested, documented, etc.

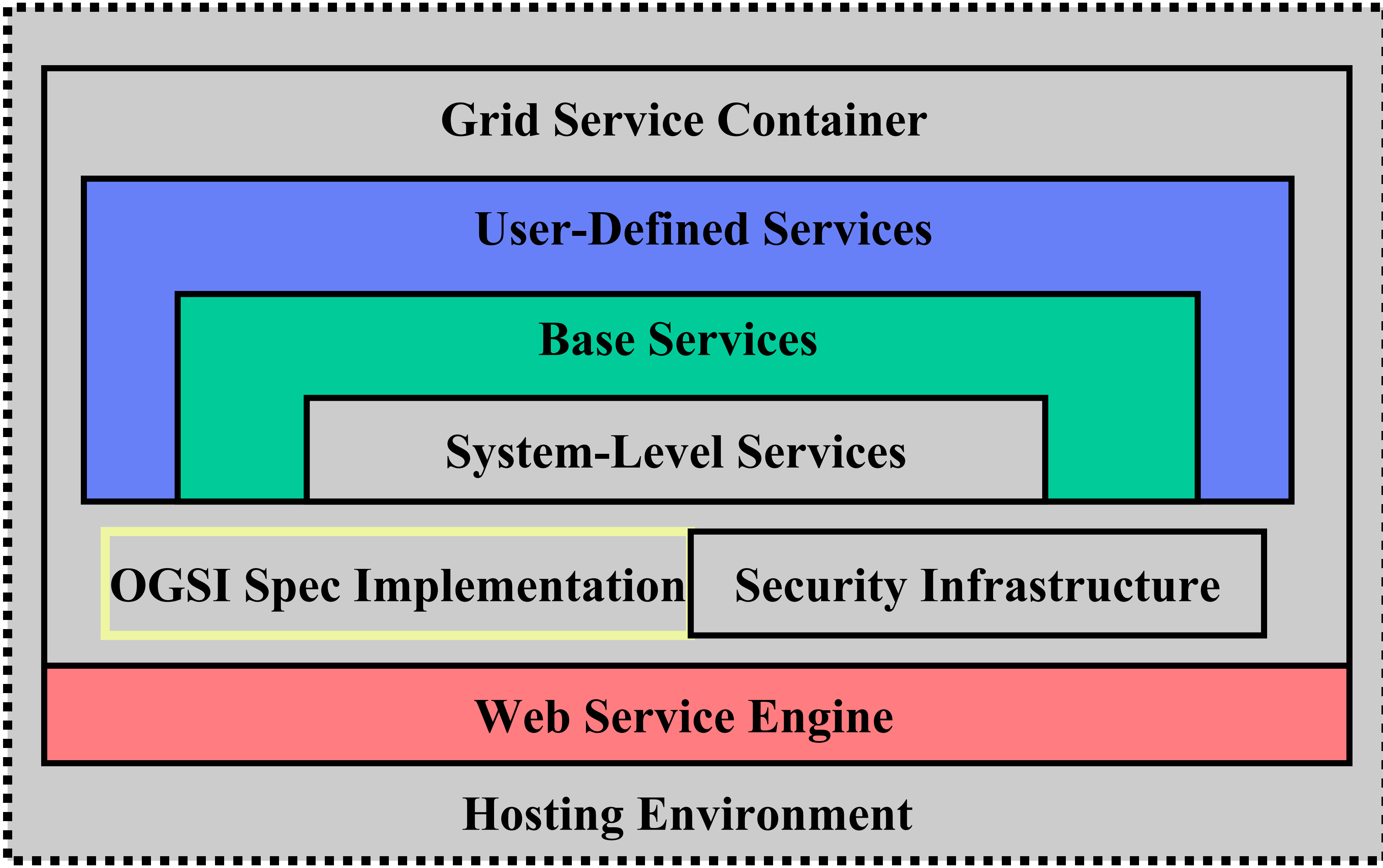
Summary

- The Grid: Coordinates resources that are not subject to centralized control; using standard, open, general-purpose protocols and interfaces; to deliver non-trivial qualities of service.
- Considerable impact within eScience, growing interest & adoption within eBusiness
- Globus Toolkit an open source, defacto standard source of protocol and API definitions—and reference implementations
- GT3 is evolution of the Globus Toolkit path

Introduction to GT3

- Background
  - The Grid Problem
  - The Globus Approach
  - OGSA & OGSI
  - Globus Toolkit
- **GT3 Architecture and Functionality: The Latest Refinement of the Globus Toolkit**
  - Core
  - Base Services
  - User-Defined Services
  - Future Directions
- Installation and Administration
  - Installation
  - Configuration
  - Debugging
  - Support
- Important Things to Remember

GT-OGSA Grid Service Infrastructure

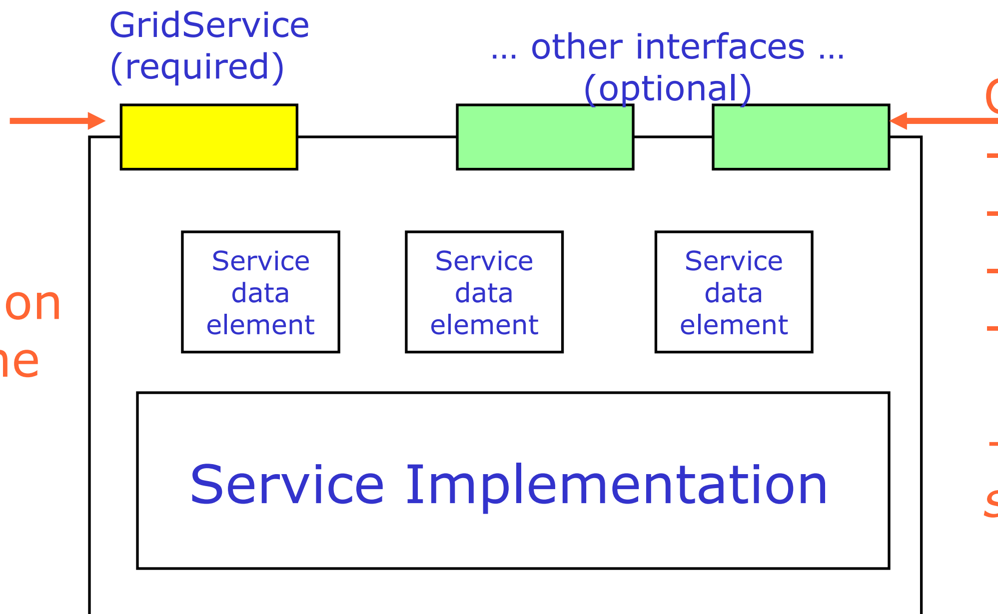


GT3 Core

GT3 Core:
OGSI Specification

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

Required:
- Introspection (service data)
- Explicit destruction
- Soft-state lifetime



Optional:
- Service creation
- Notification
- Registration
- Service Groups

+ application-specific interfaces

Service locator

Includes 0 or more Grid Service Handles (GSHs)

Includes 0 or more Grid Service References (GSRs)

GT3 Core: OGSI Implementation

- GT3 includes a set of primitives that implement the interfaces and behaviors defined in the latest version of the OGSI Specification
- The implementation supports a declarative programming model in which GT3 users can compose OGSI-Compliant grid services by plugging the desired primitives into their implementation

GT3 Core:
OGSI Specification (cont.)

GridService portType

- Defines the fundamental behavior of a Grid Service
  - Introspection
  - Discovery
  - Soft State Lifetime Management
- Mandated by the Spec

GT3 Core:
OGSI Specification (cont.)

Factory portType

- Factories create services
- Factories are typically persistent services
- Factory is an optional OGSI interface

(Grid Services can also be instantiated by other mechanisms)

GT3 Core: OGSI Specification (cont.)

Notification portTypes

- A subscription for notification causes the creation of a NotificationSubscription service
- NotificationSinks are not required to implement the GridService portType
- Notifications can be set on Service Data Elements
- Notification portTypes are optional

GT3 Core: OGSI Specification (cont.)

Service group portTypes

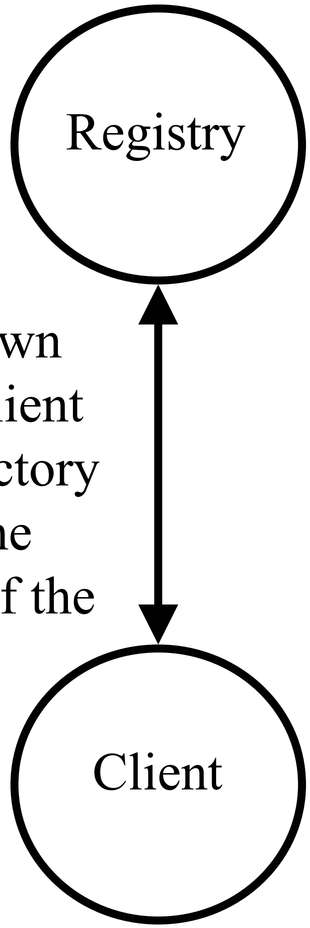
- A ServiceGroup is a grid service that maintains information about a group of other grid services
- The classic registry model can be implemented with the ServiceGroup portTypes
- A grid service can belong to more than one ServiceGroup
- Members of a ServiceGroup can be heterogenous or homogenous
- Each entry in a service group can be represented as its own service
- Service group portTypes are optional OGSI interfaces

GT3 Core: OGSI Specification (cont.)

HandleResolver portType

- Defines a means for resolving a GSH (Grid Service Handle) to a GSR (Grid Service Reference)
  - A GSH points to a Grid Service
    (GT3 uses a hostname-based GSH scheme)
  - A GSR specifies how to communicate with the Grid Service
    (GT3 currently supports SOAP over HTTP, so GSRs are in WSDL format)
- HandleResolver is an optional OGSI interface

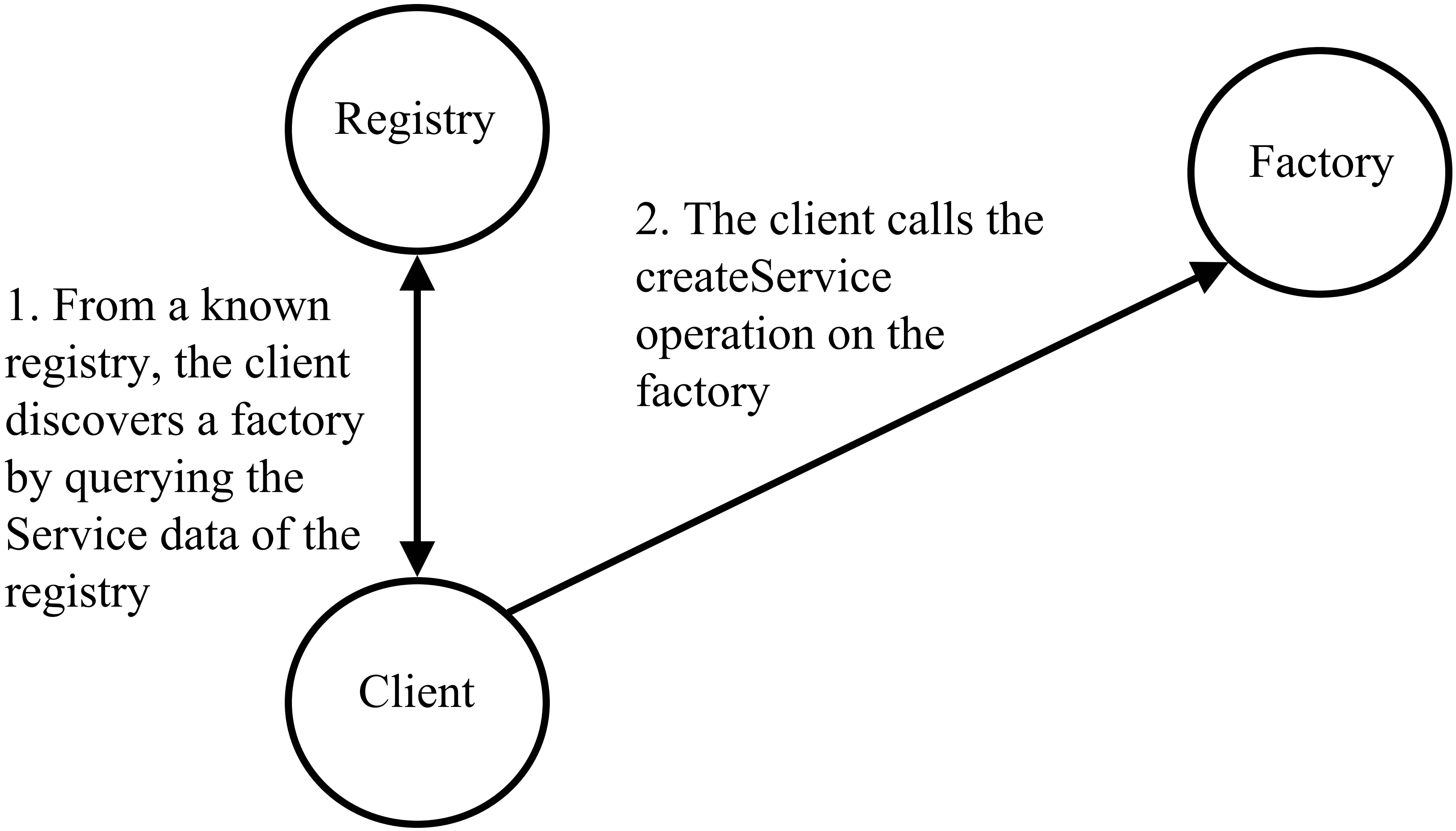
A Service Creation Scenario*



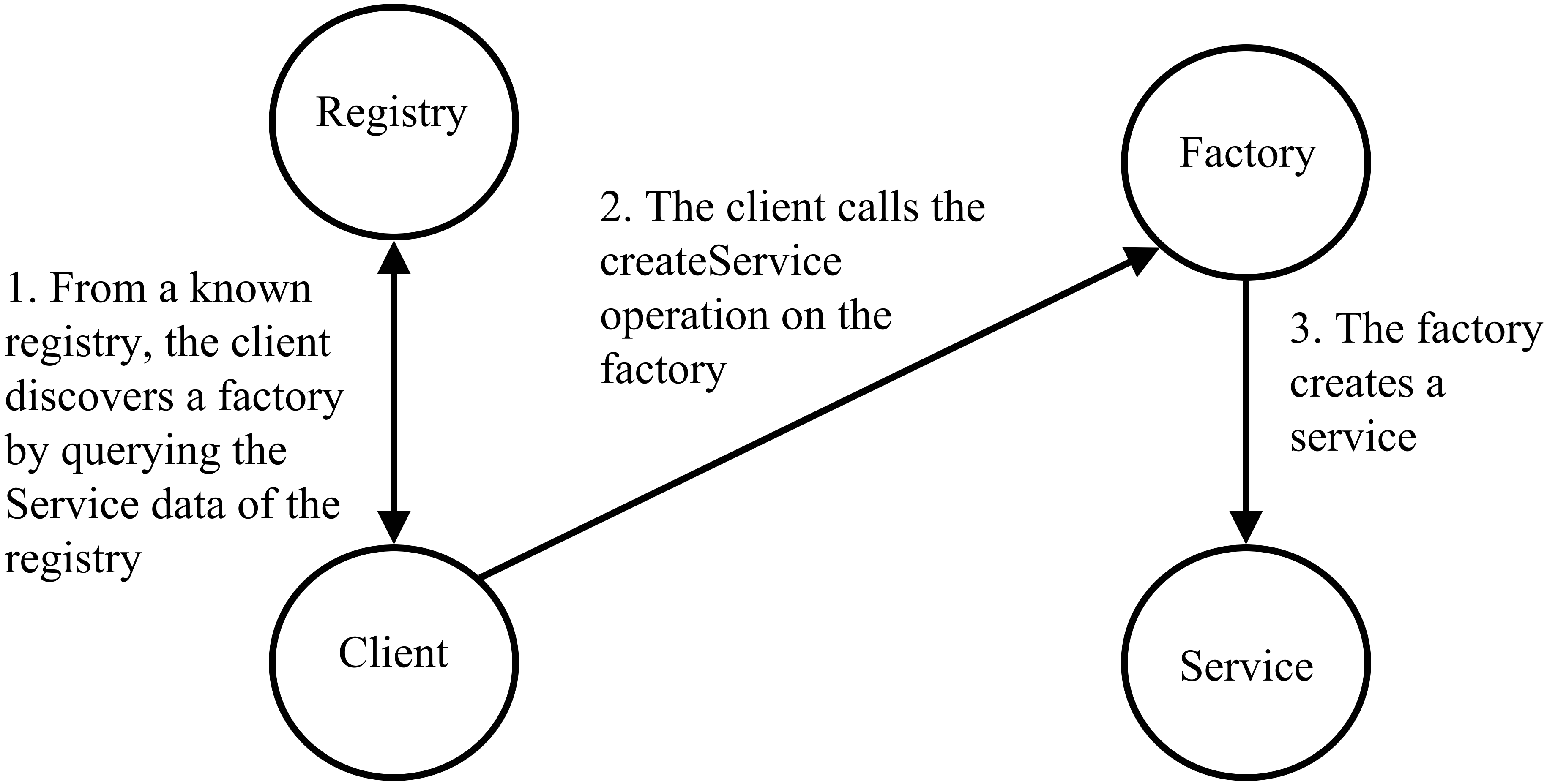
1. From a known registry, the client discovers a factory by querying the Service data of the registry

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

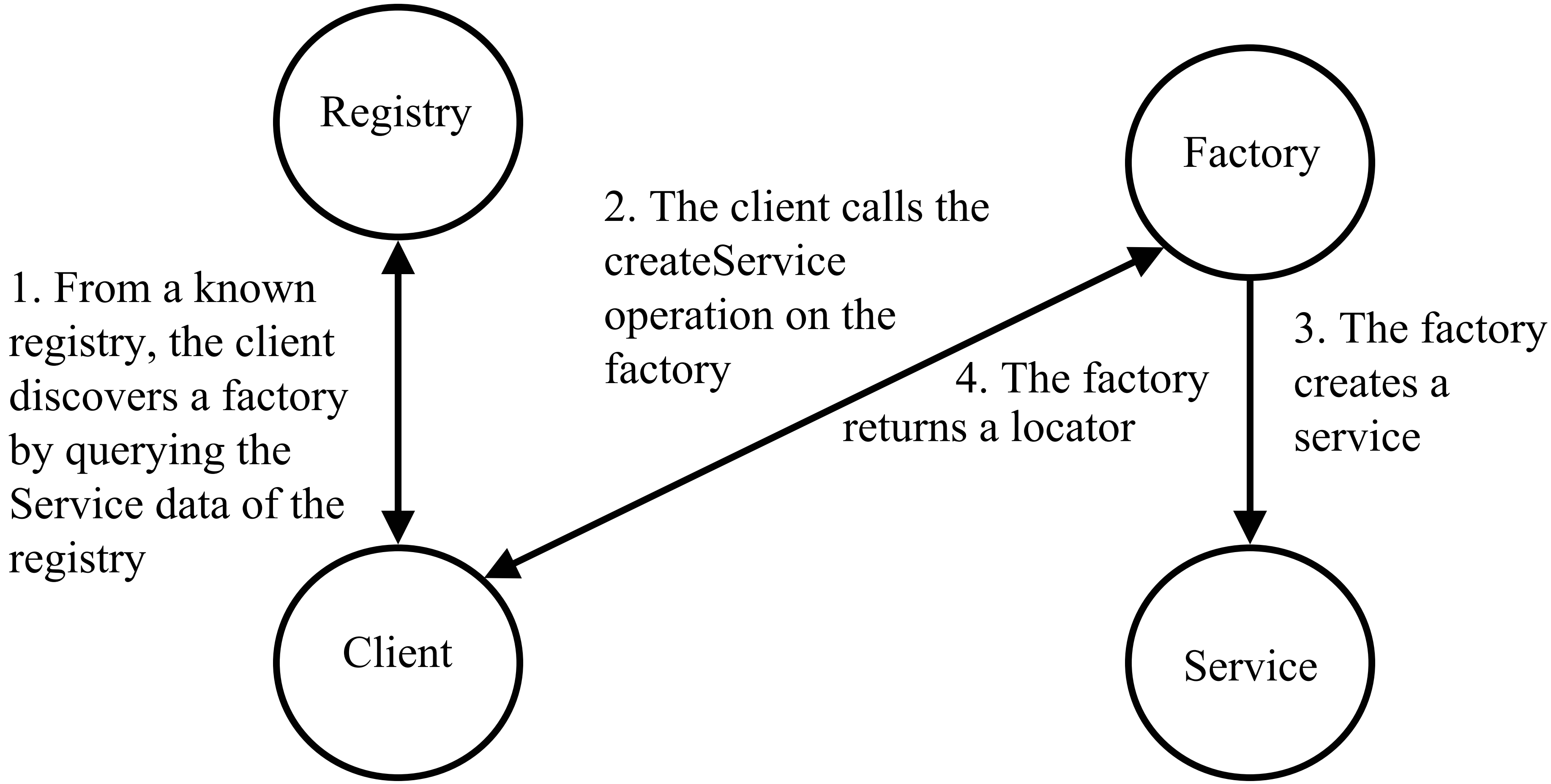
A Service Creation Scenario



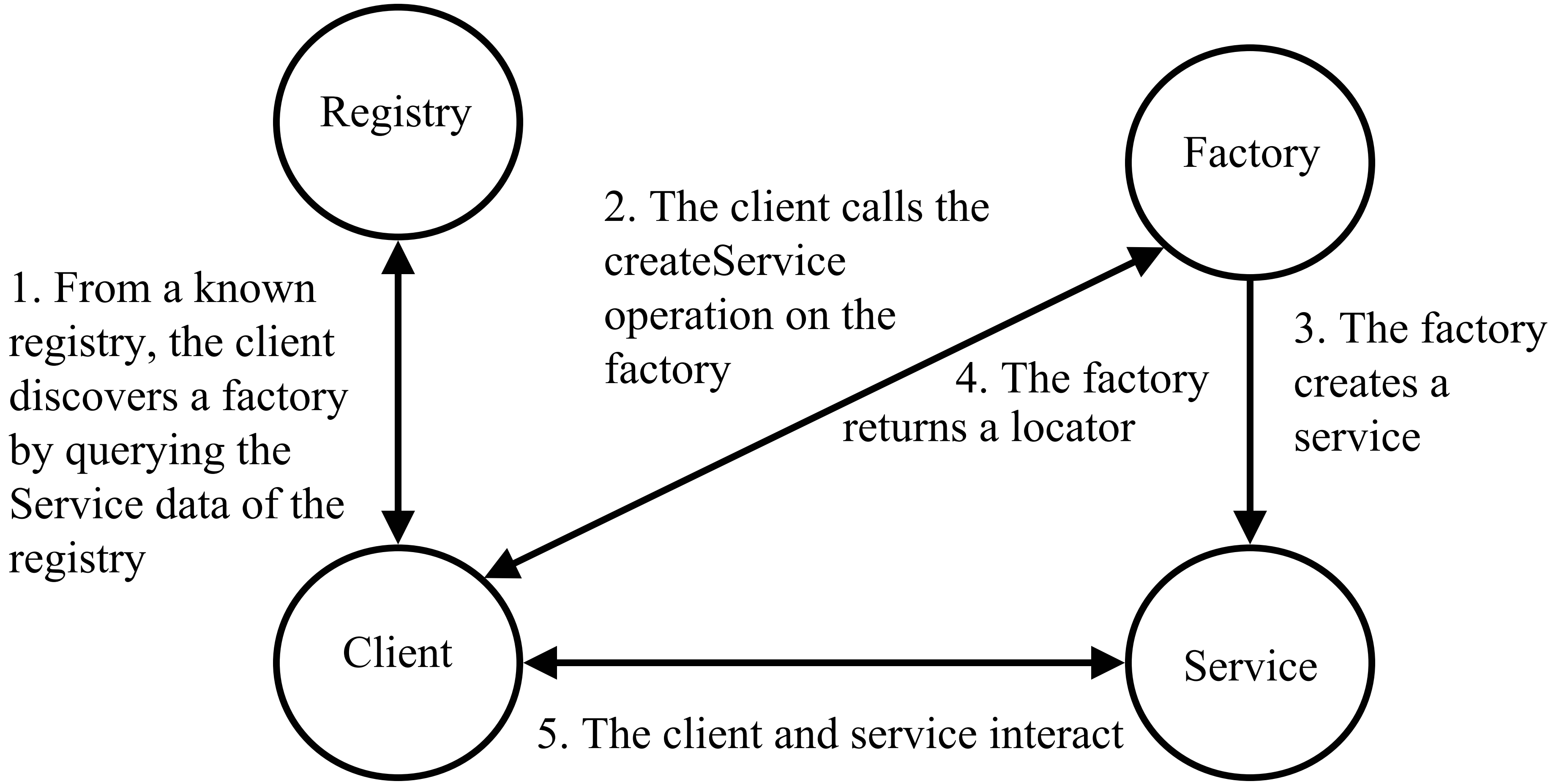
A Service Creation Scenario



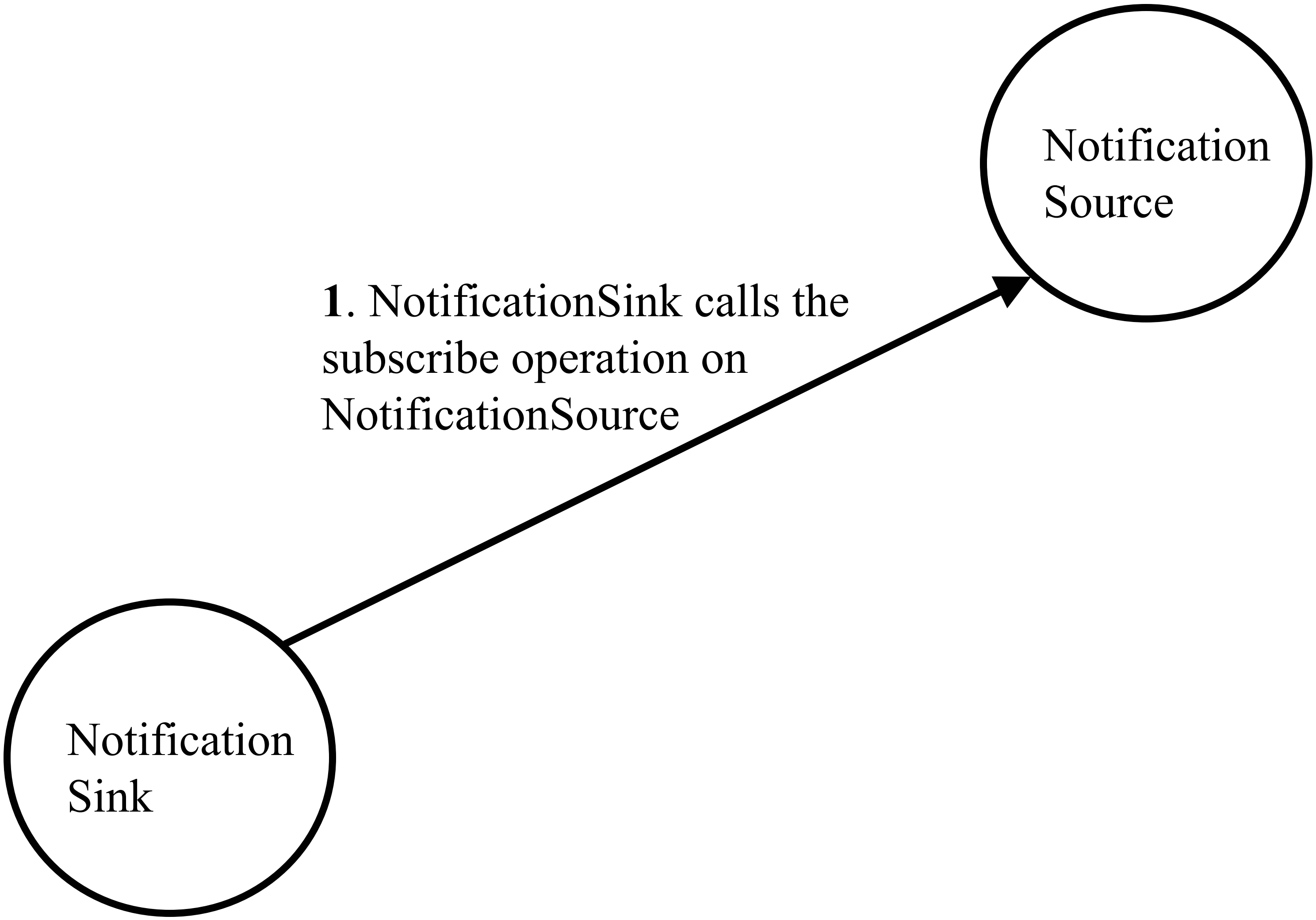
A Service Creation Scenario



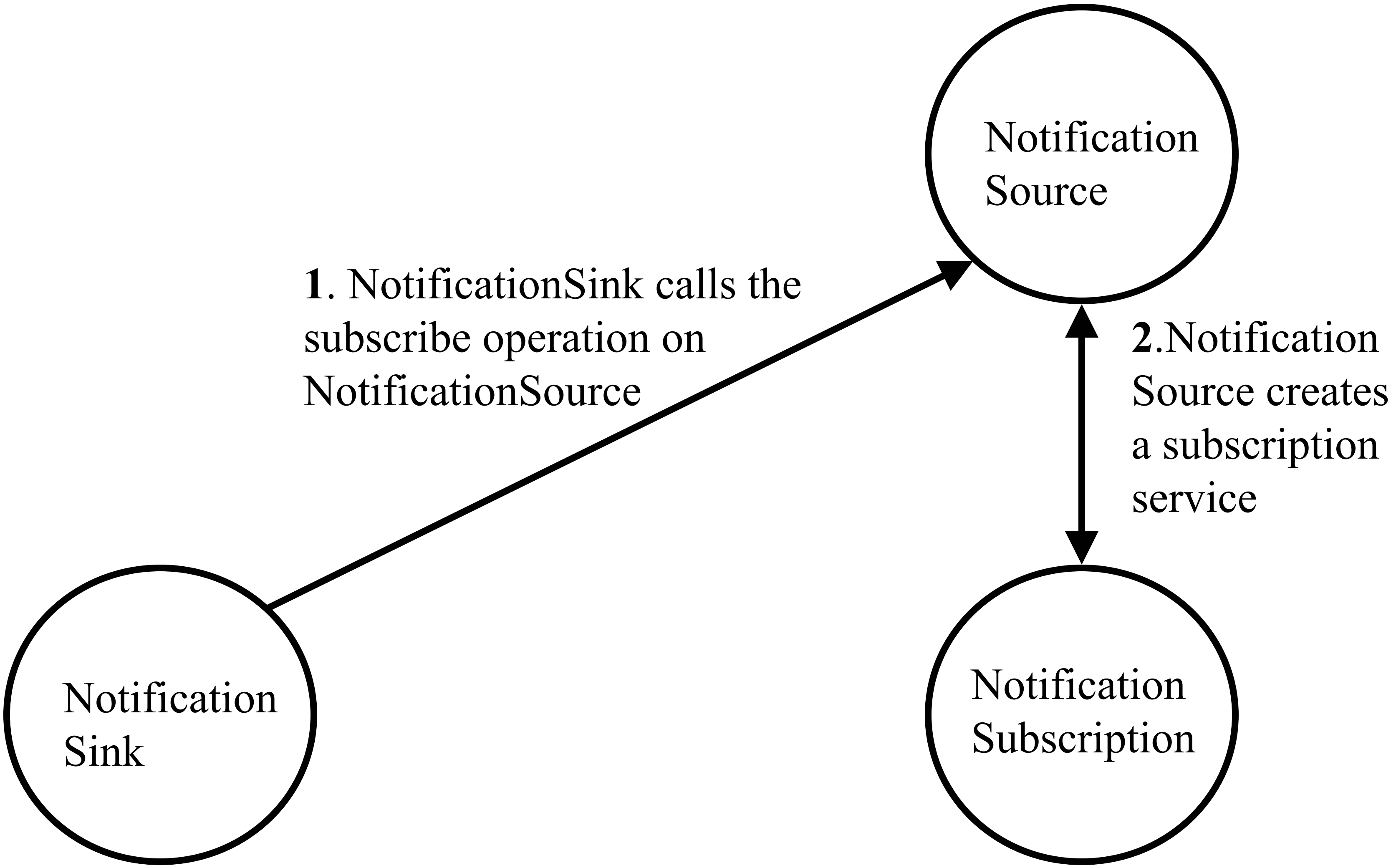
A Service Creation Scenario



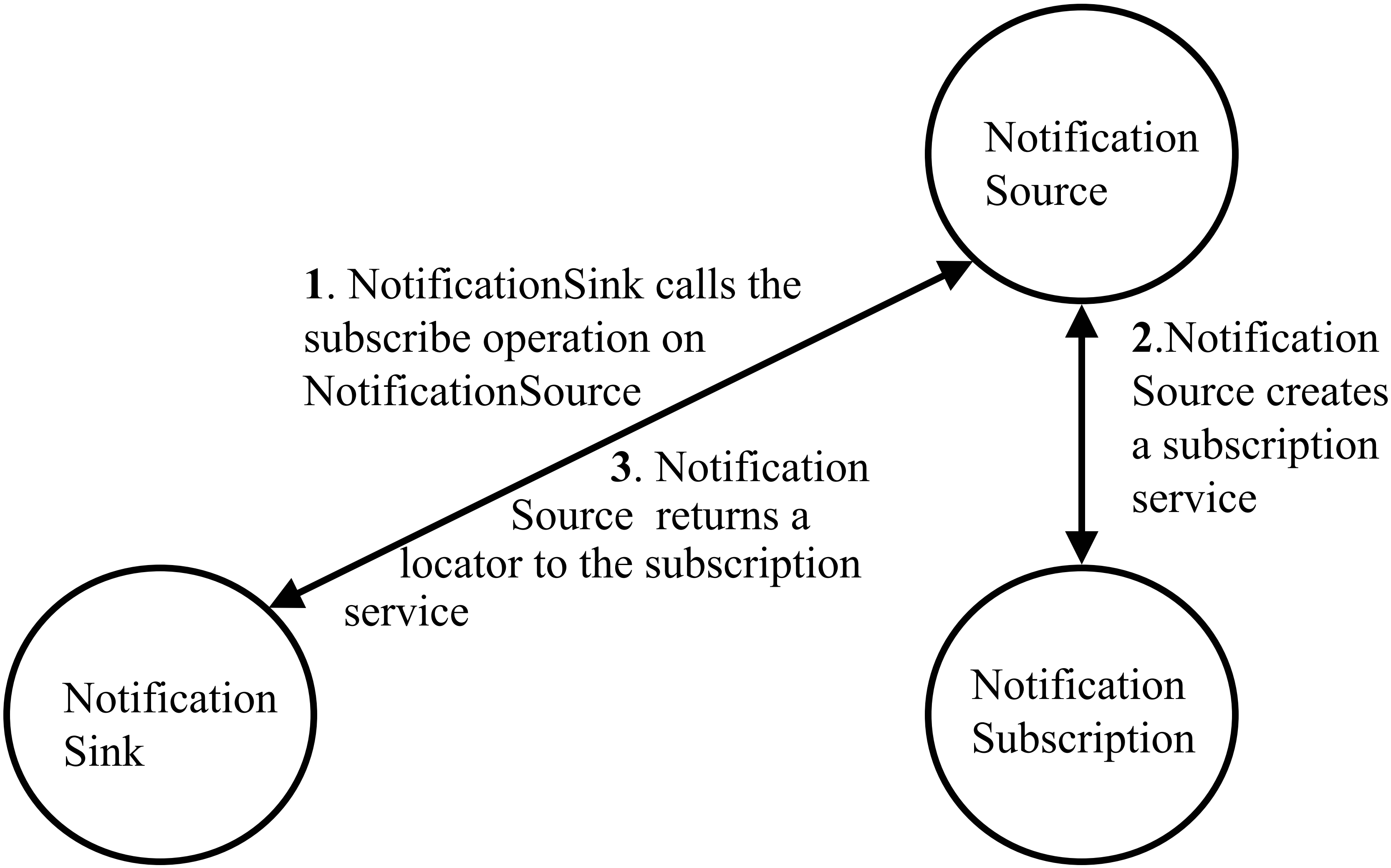
A Notification Scenario



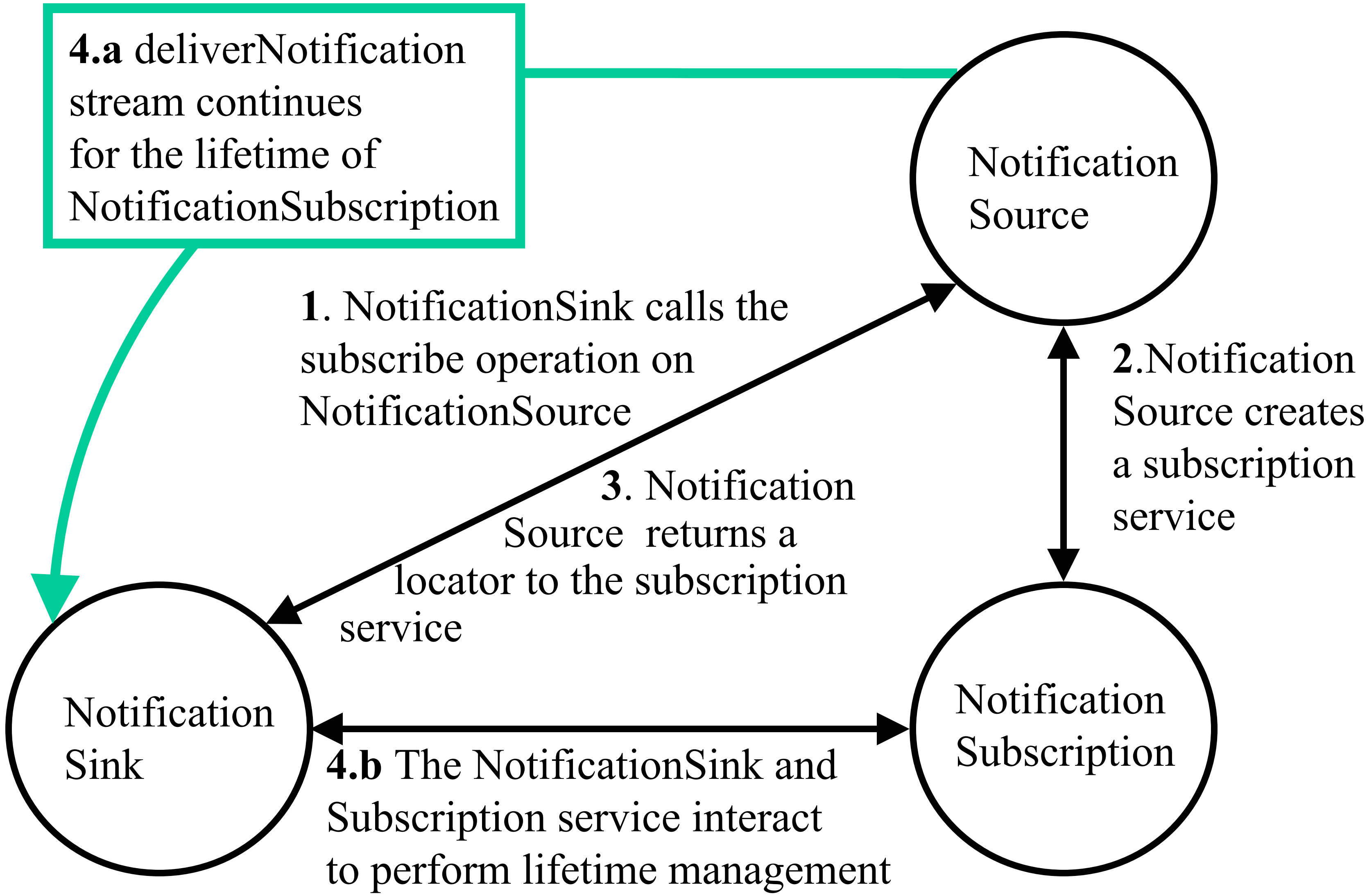
A Notification Scenario



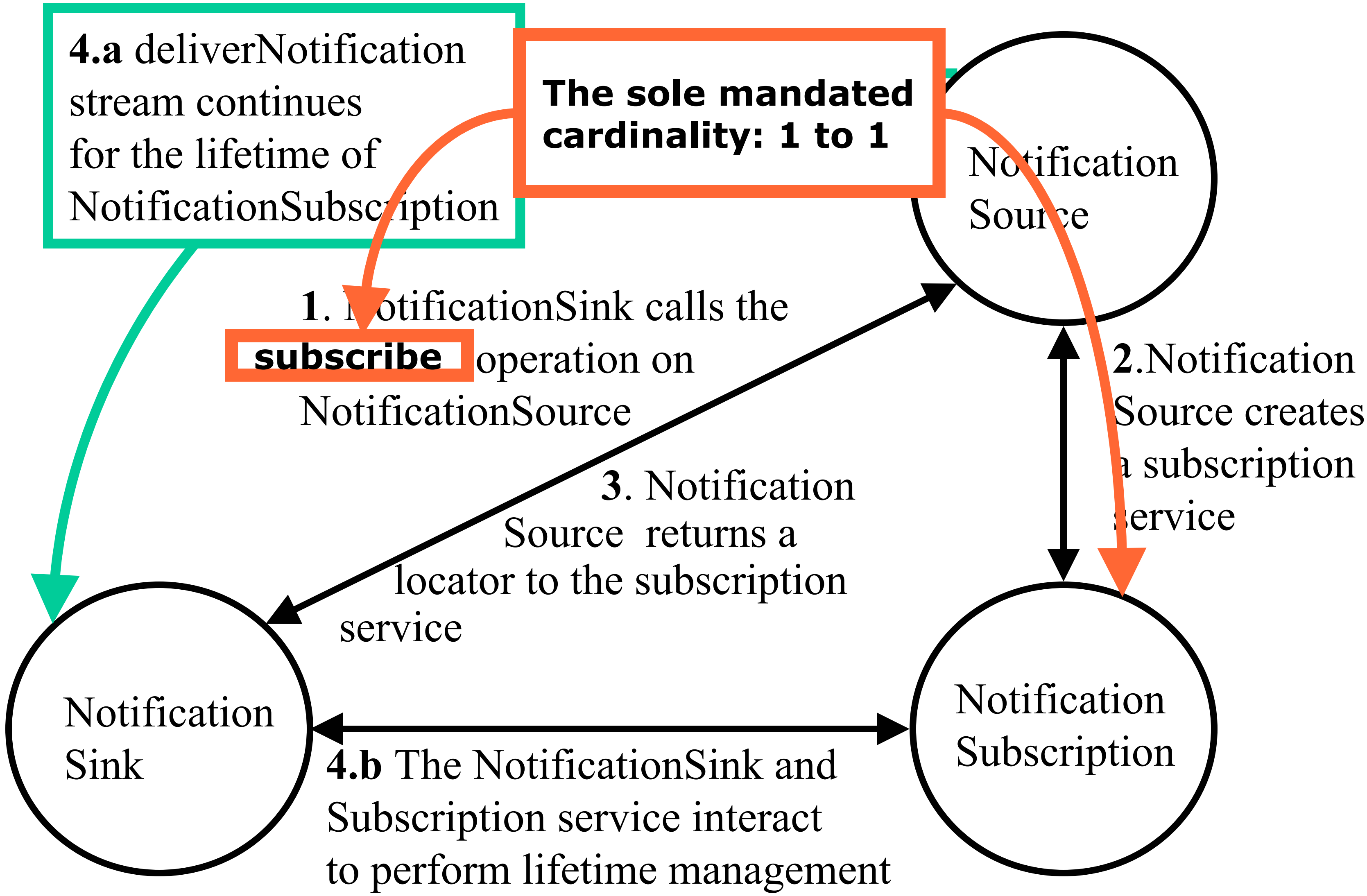
A Notification Scenario



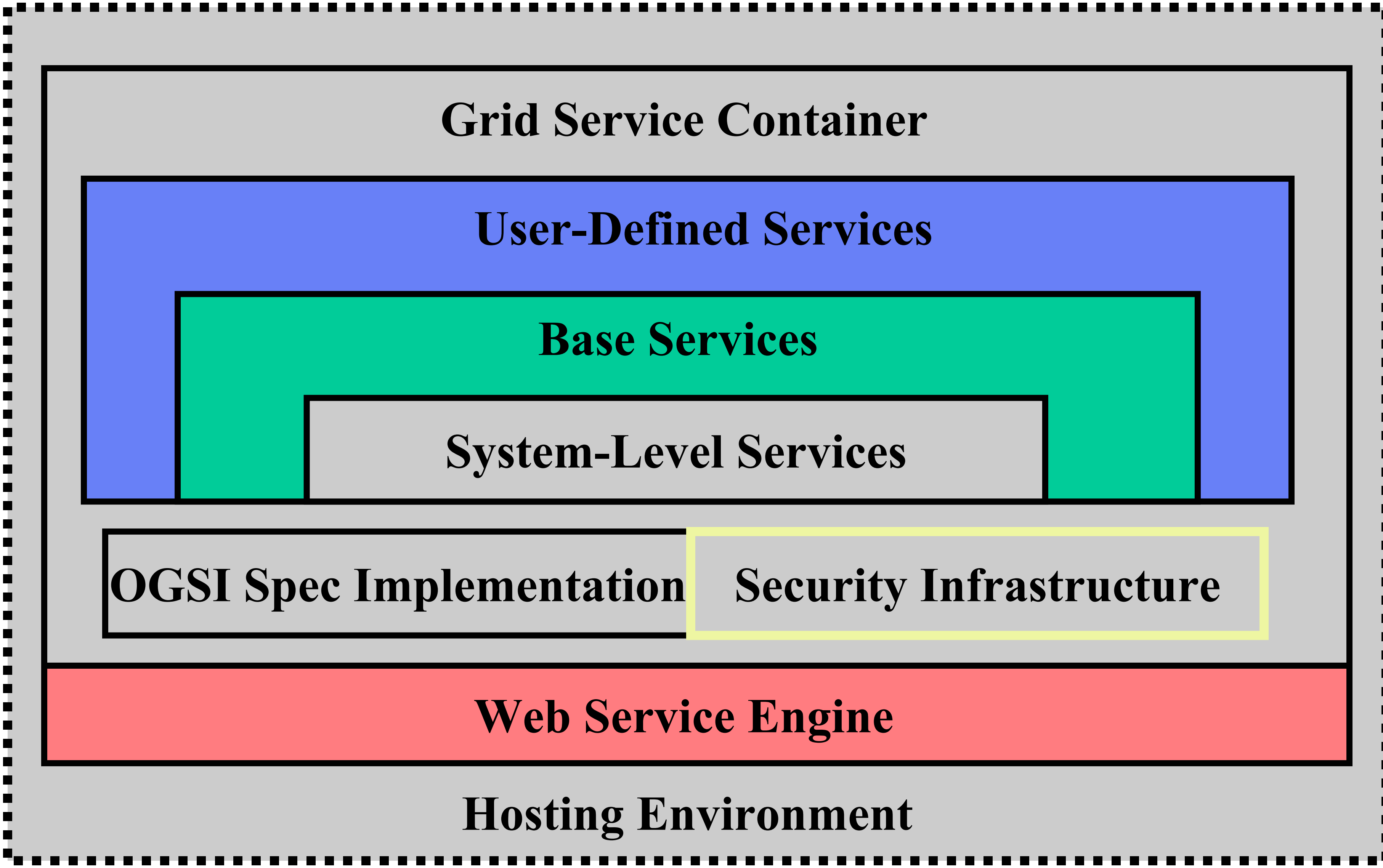
A Notification Scenario



A Notification Scenario



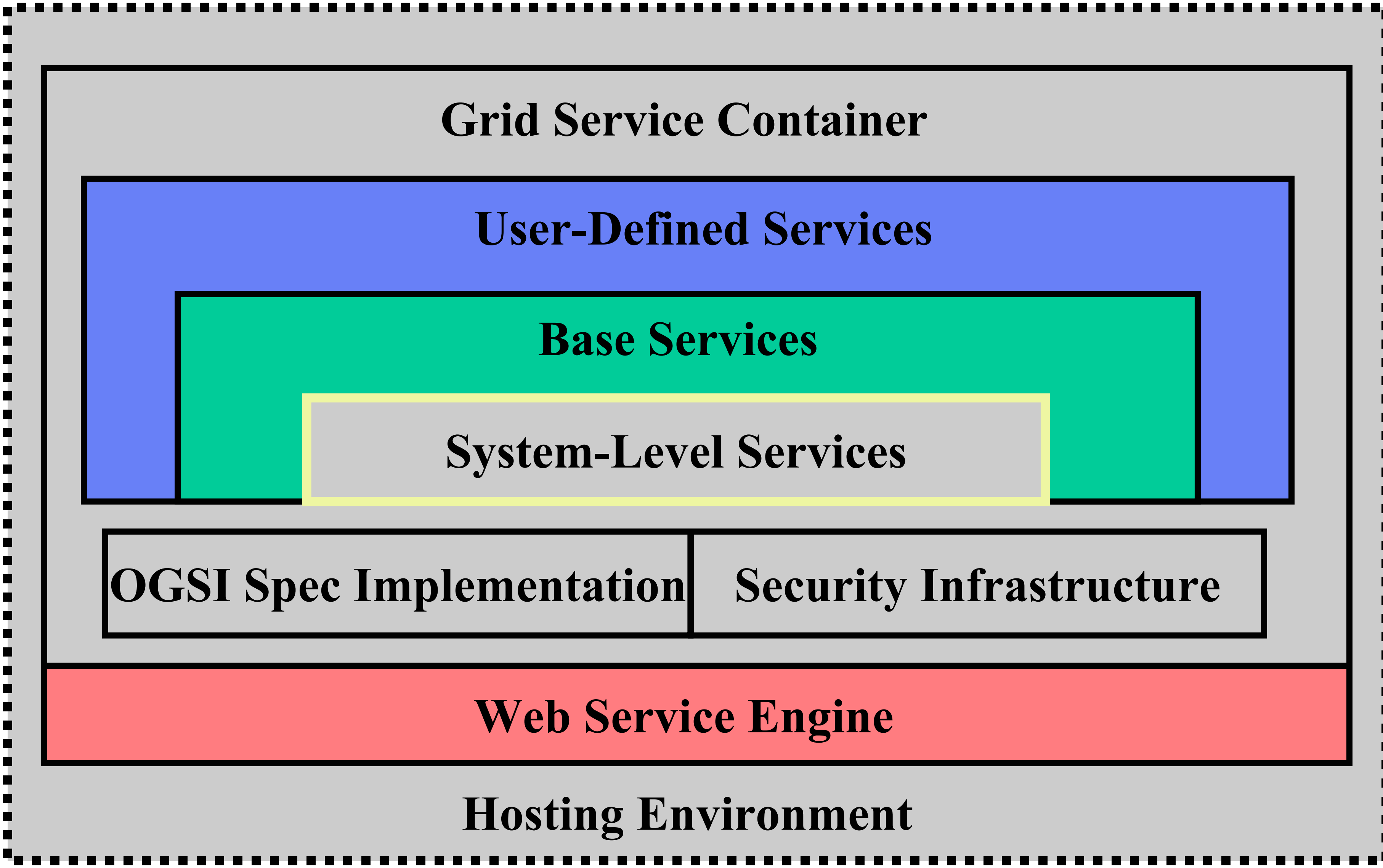
GT-OGSA Grid Service Infrastructure



GT3 Core: Security Infrastructure

- Transport Layer Security/Secure Socket Layer (TLS/SSL)
  - To be deprecated
- SOAP Layer Security
  - Based on WS-Security, XML Encryption, XML Signature
- GT3 uses X.509 identity certificates for authentication
- It also uses X.509 Proxy certificates to support delegation and single sign-on, updated to conform to latest IETF/GGF draft

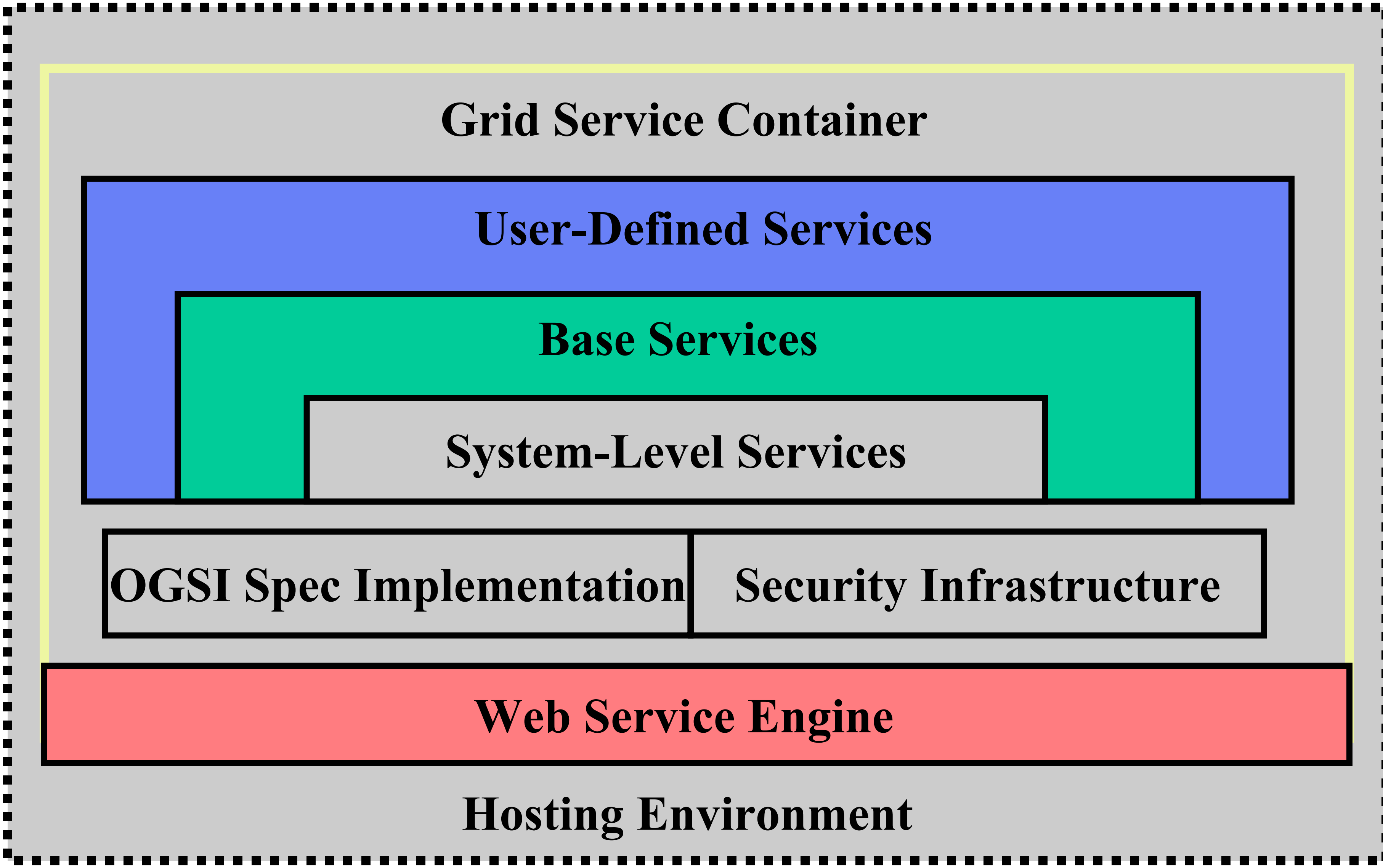
GT-OGSA Grid Service Infrastructure



GT3 Core: System Level Services

- General-purpose services that facilitate the use of Grid Services in production environments
- The 3.0 distribution includes the following System-Level services:
  – An Administration Service
  – A Logging Service
  – A Management Service

GT-OGSA Grid Service Infrastructure



GT3 Core: Grid Service Container

Includes the OGSI Implementation, security infrastructure and system-level services, plus:

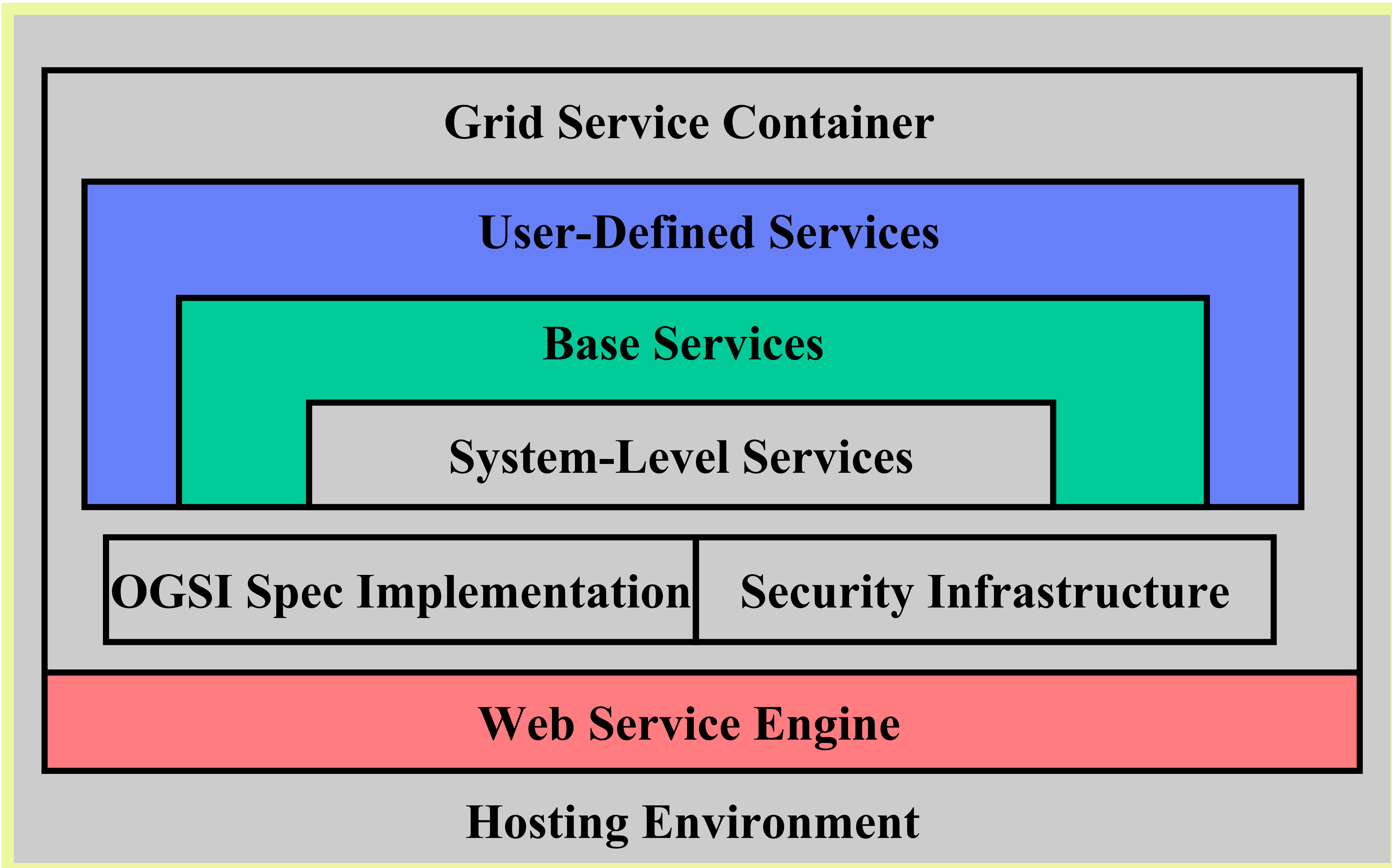
- Service activation, deactivation, construction, destruction, etc.
- Service data element placeholders that allow you to dynamically fetch service data values at query time
- Evaluator framework (supporting ByXPath and ByName notifications and queries)
- Interceptor/callback framework (allows one to intercept certain service lifecycle events)

GT3 Core: Grid Service Container (cont.)

Layers in the Web Services Model

Layer	
Interface Layer	**OGSI Spec is here**
Transport Layer	**Transport/Binding Layer (GT3 supports SOAP over HTTP)**
Implementation Layer	**Container is here**

GT-OGSA Grid Service Infrastructure



GT3 Core: Hosting Environment

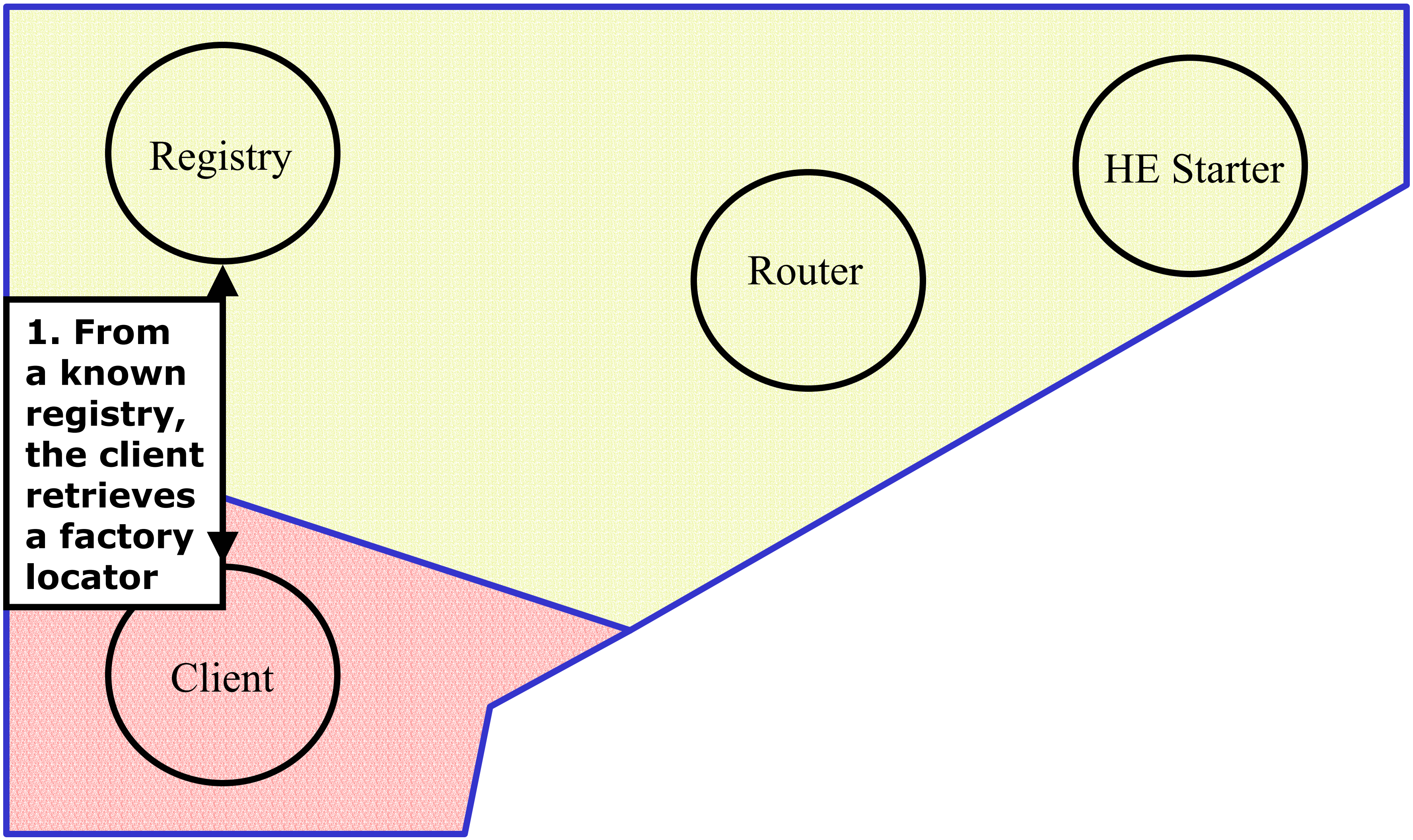
GT3 currently offers support for four Java Hosting Environments:

- Embedded
- Standalone
- Servlet
- EJB

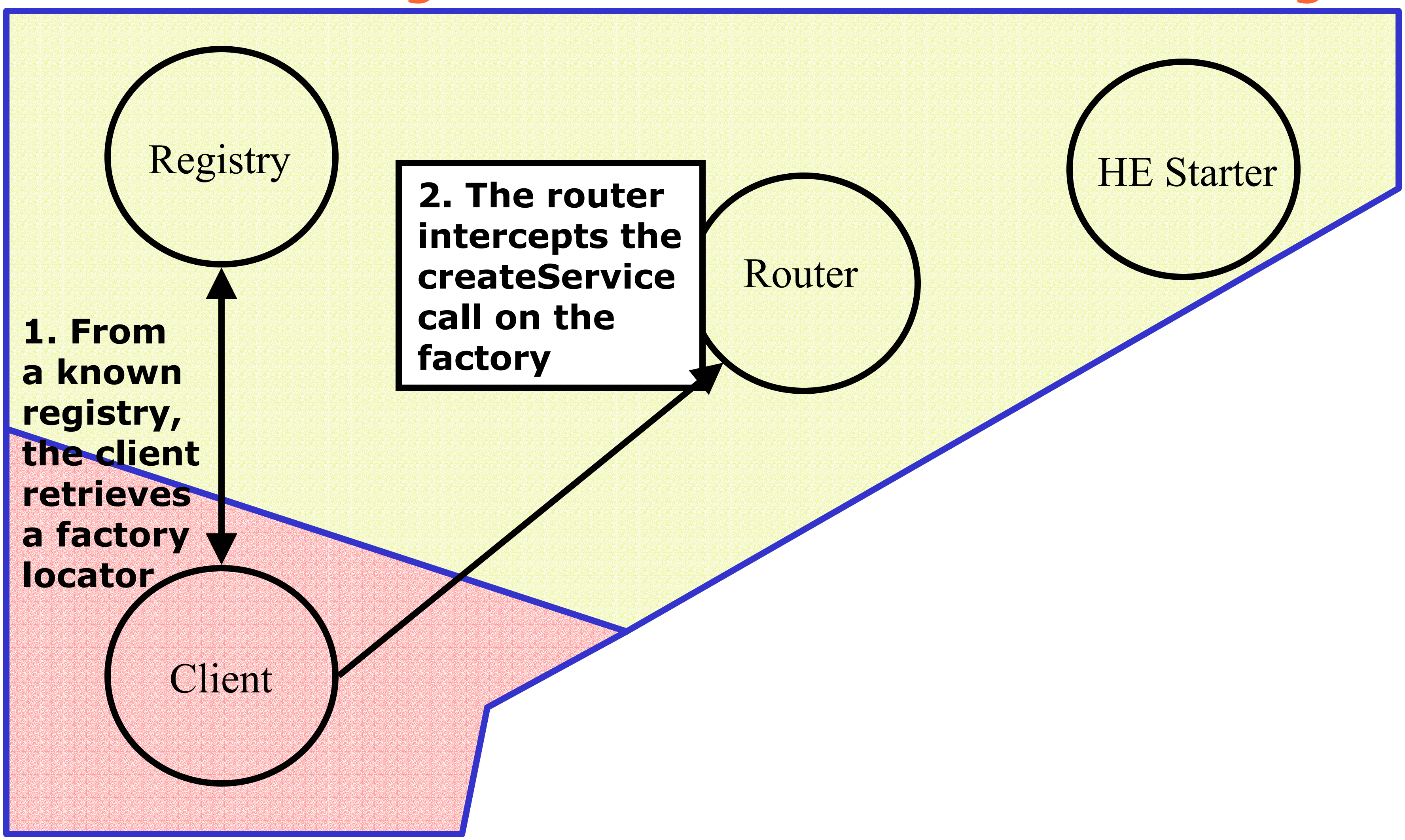
GT3 Core: Virtual Hosting Environment Framework

- Virtual Hosting allows grid services to be distributed across several remote containers
- Useful in implementing solutions for problems common to distributed computing
  - Load balancing
  - User account sandboxing

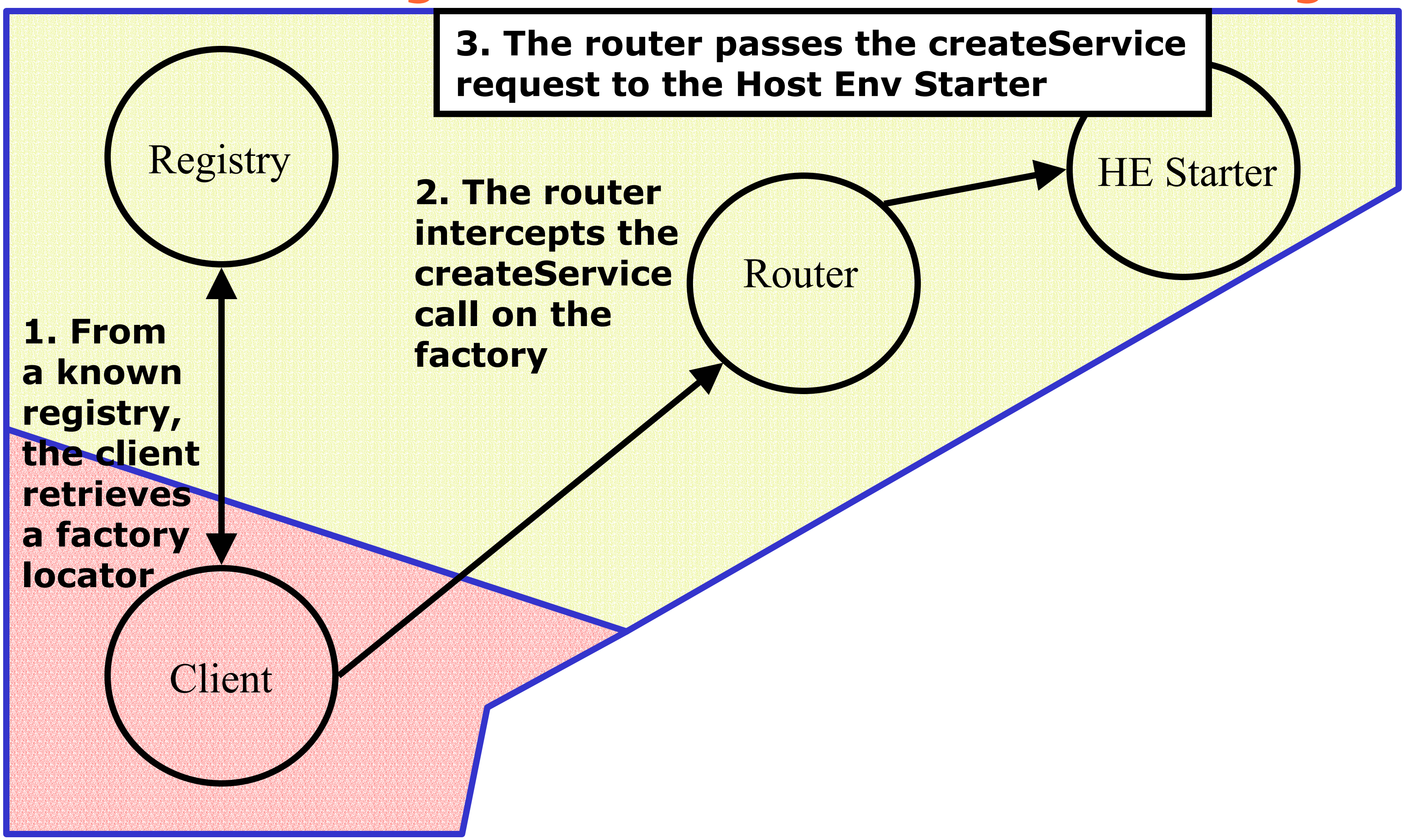
A Service Creation Scenario Illustrating Redirection in Virtual Hosting



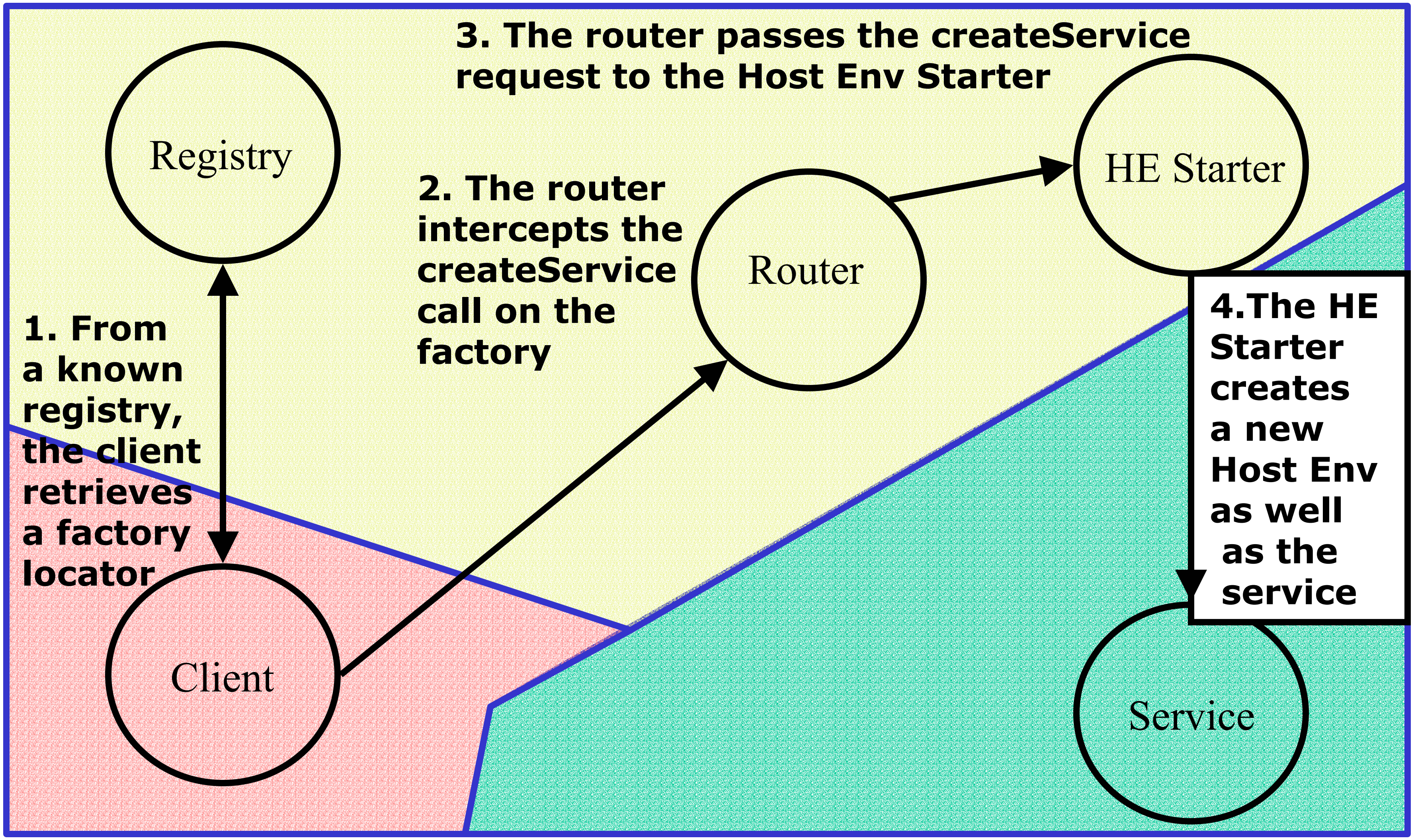
A Service Creation Scenario Illustrating Redirection in Virtual Hosting



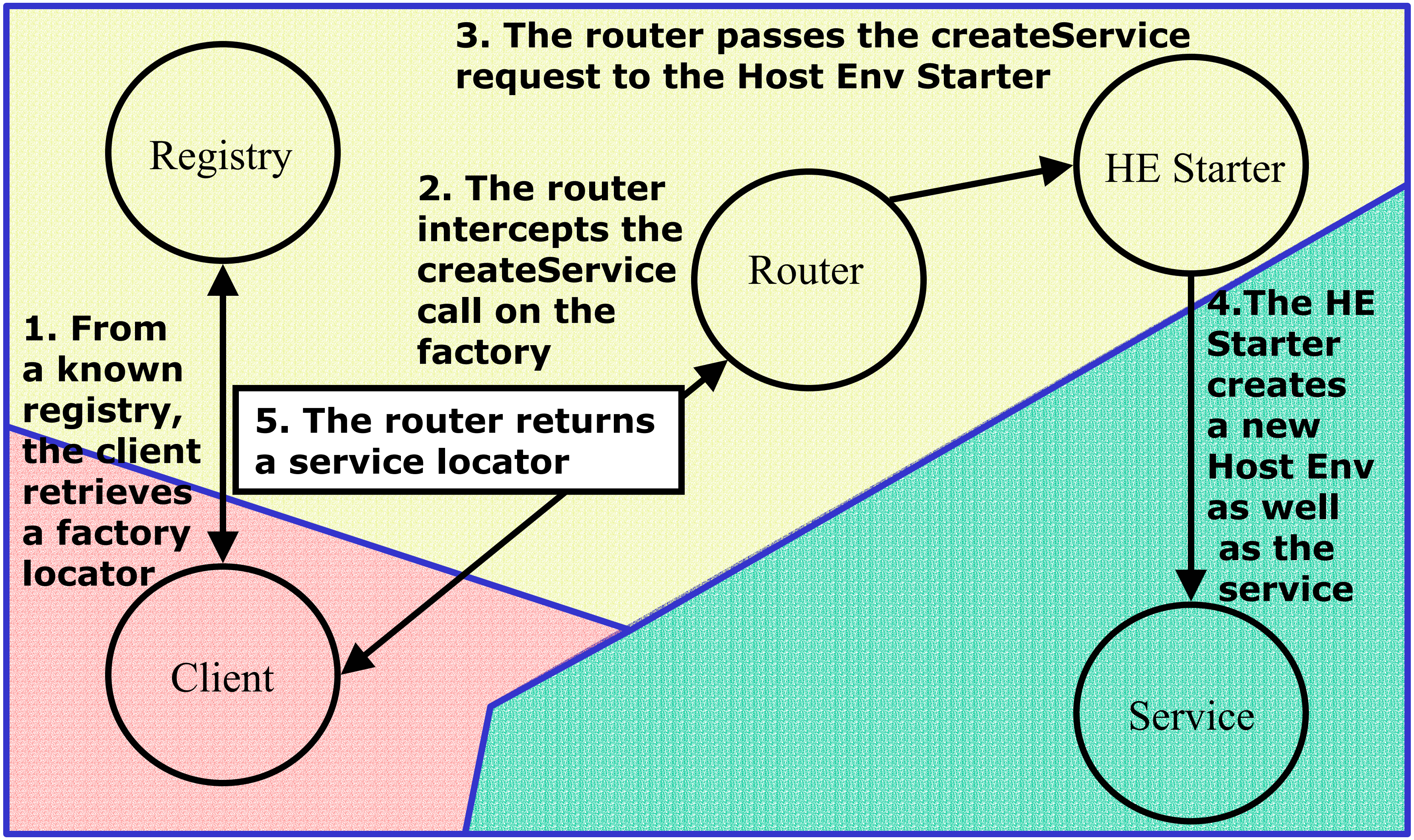
A Service Creation Scenario Illustrating Redirection in Virtual Hosting



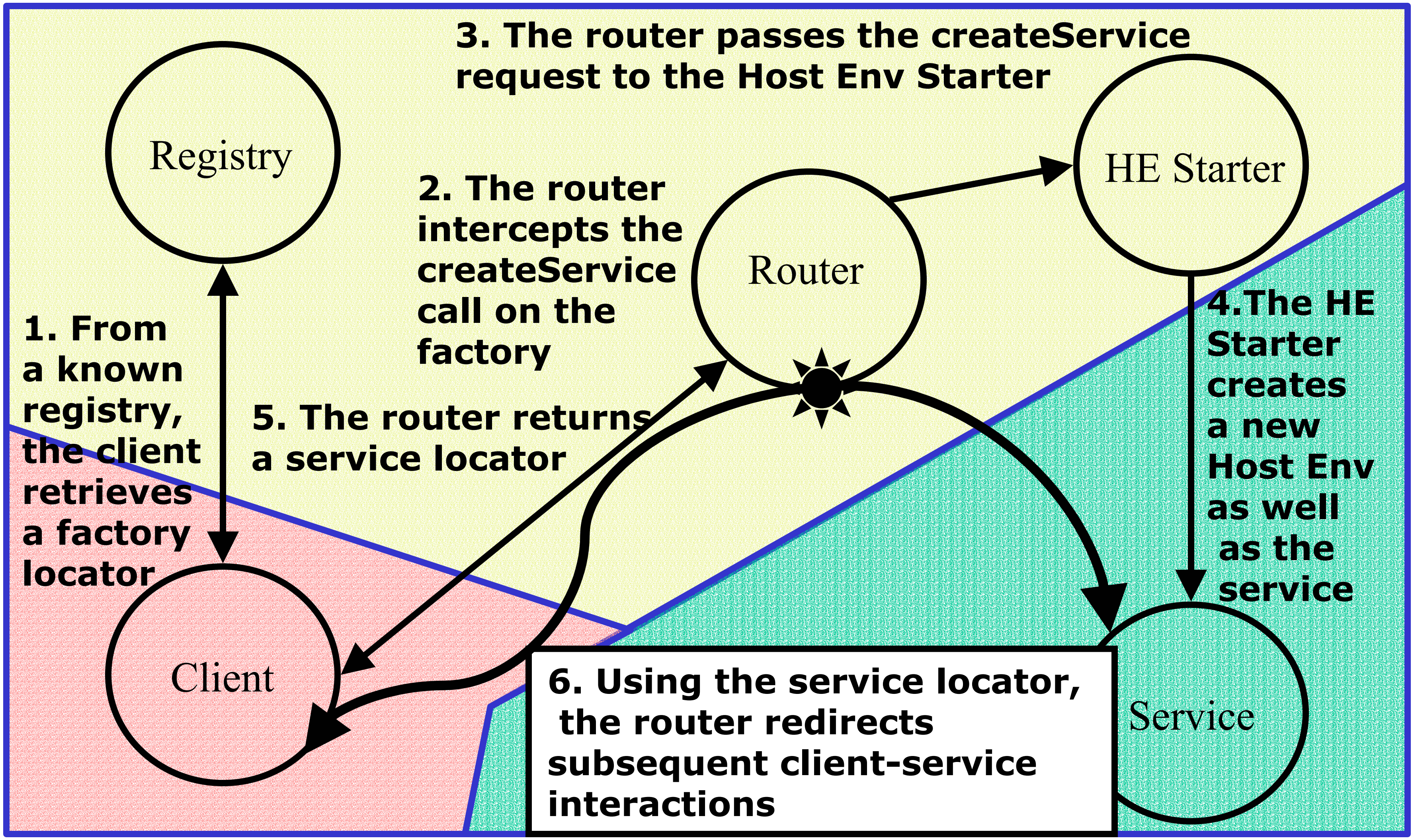
A Service Creation Scenario Illustrating Redirection in Virtual Hosting



A Service Creation Scenario Illustrating Redirection in Virtual Hosting

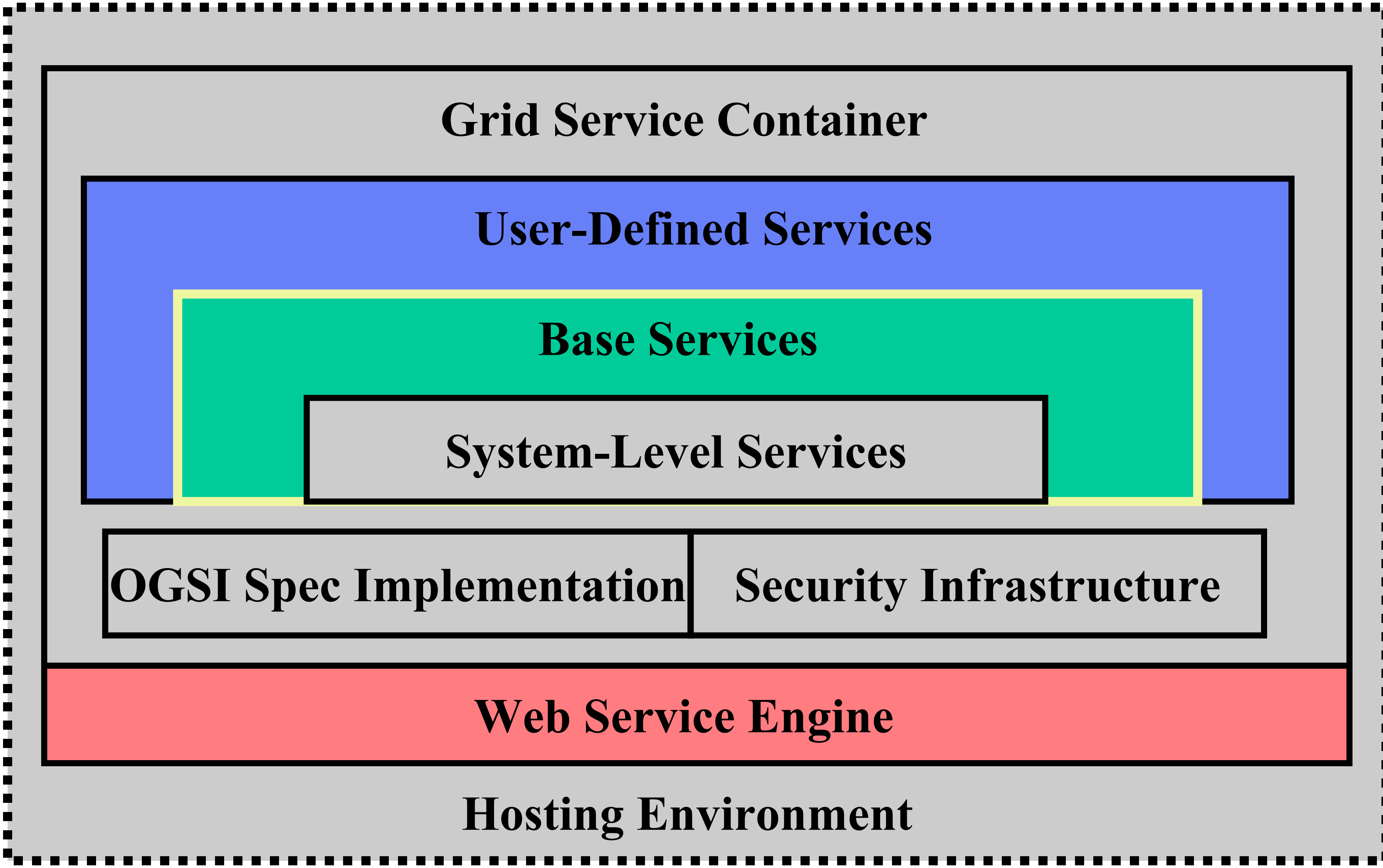


A Service Creation Scenario Illustrating Redirection in Virtual Hosting



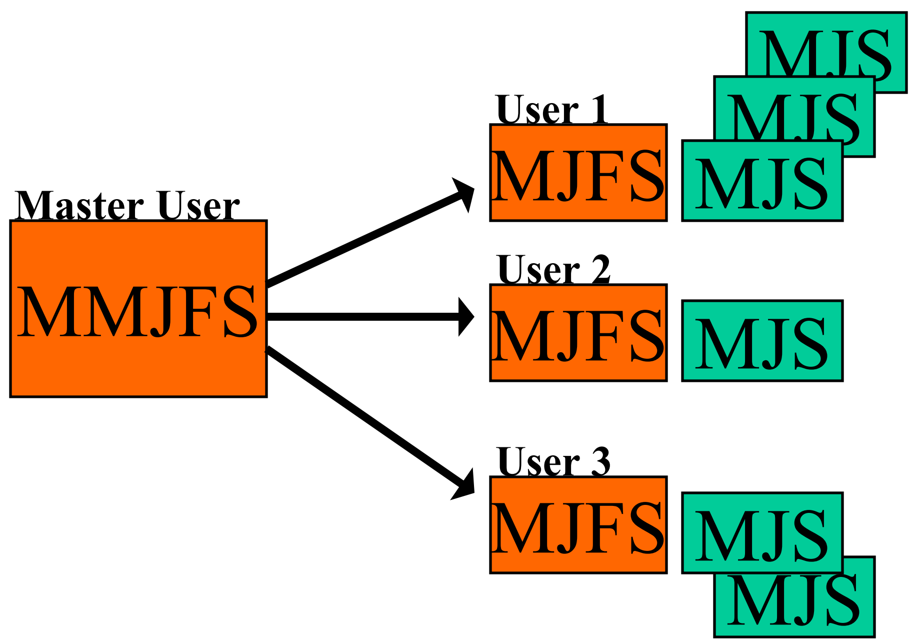
GT3 Base Services

GT-OGSA Grid Service Infrastructure



GT3 Base: Resource Management

- GRAM Architecture rendered in OGSA
- The MMJFS runs as an unprivileged user, with a small highly-constrained setuid executable behind it.

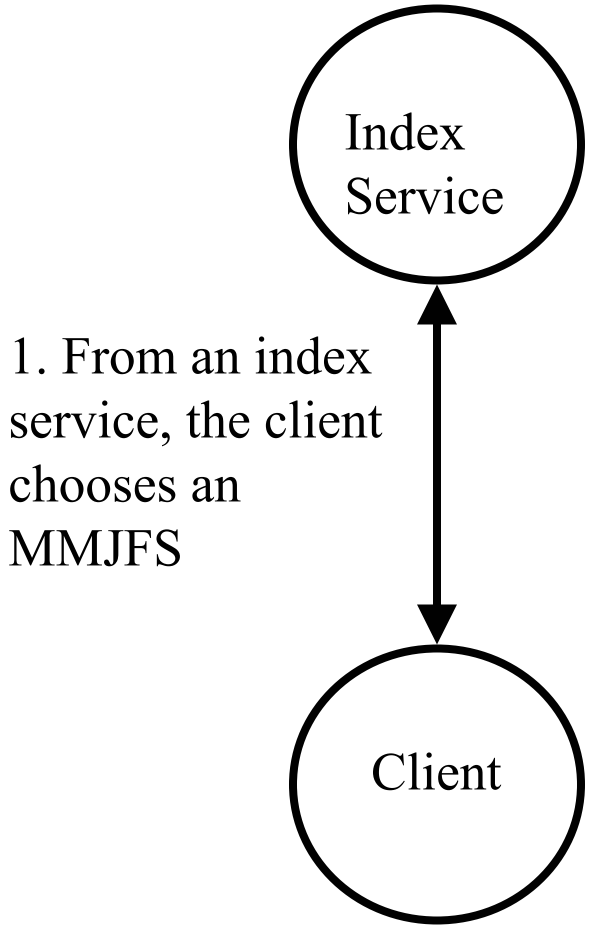


MMJFS: Master Managed Job Factory Service

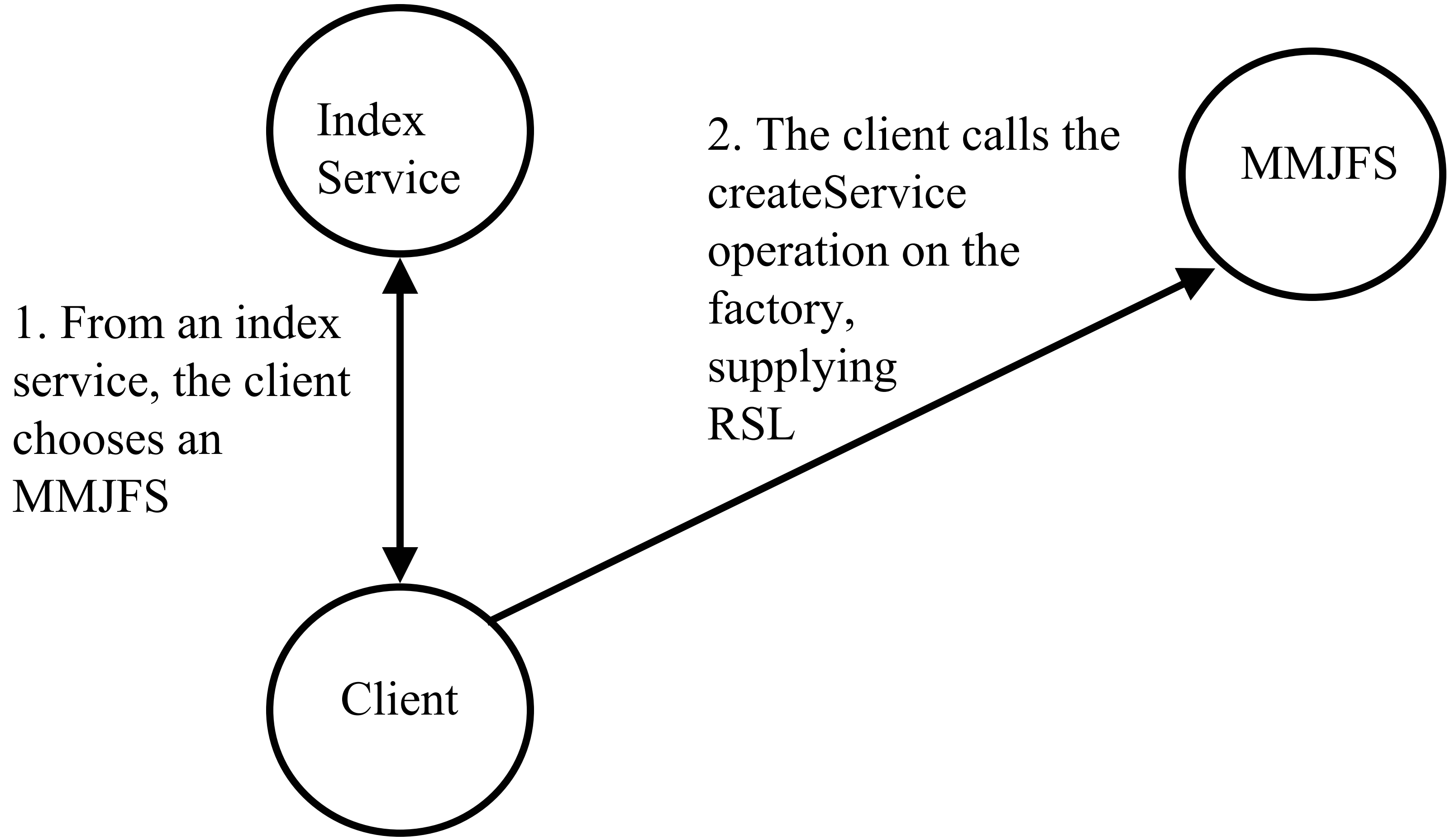
MJFS: Managed Job Factory Service

MJS: Managed Job Service

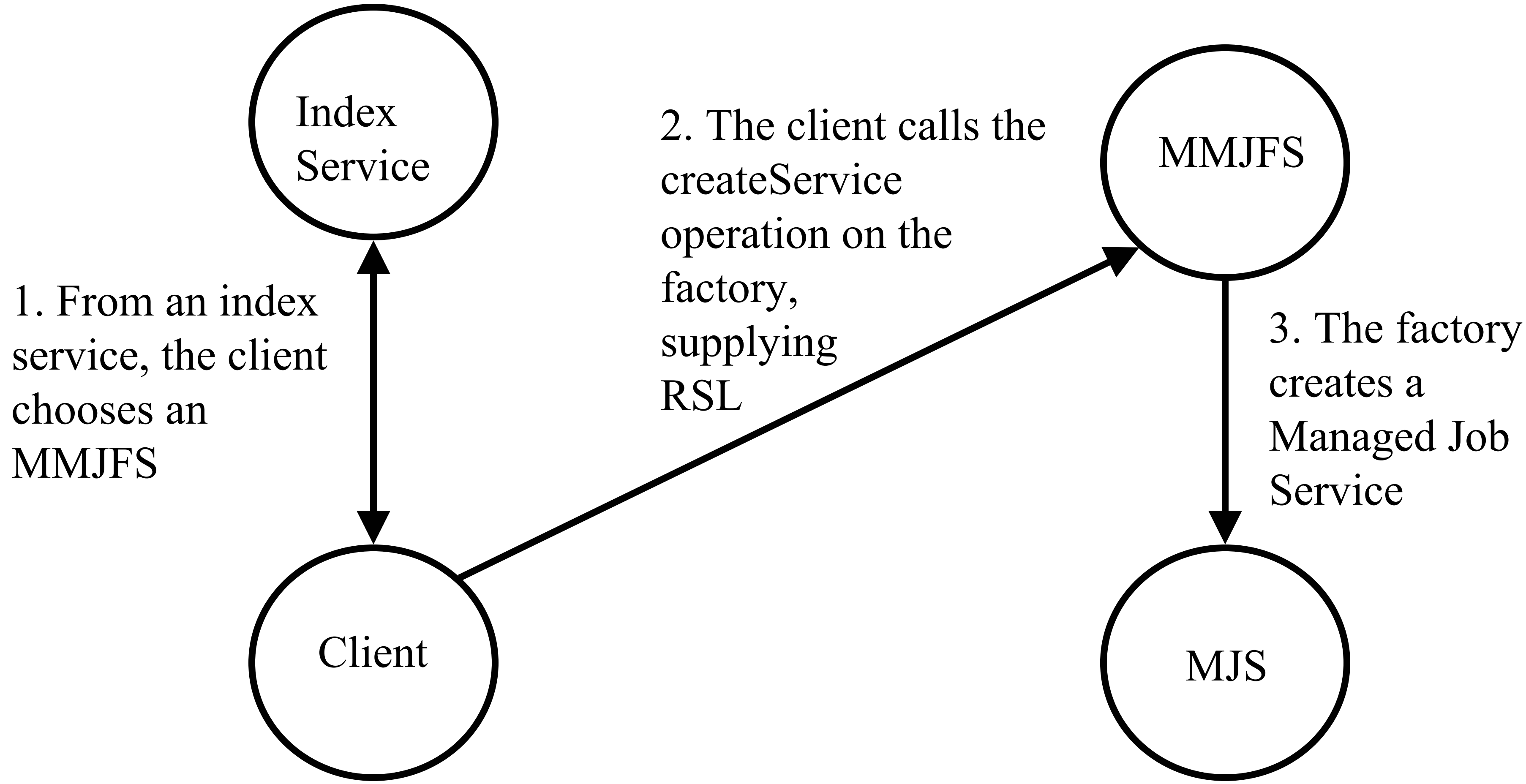
GRAM Job Submission Scenario



GRAM Job Submission Scenario

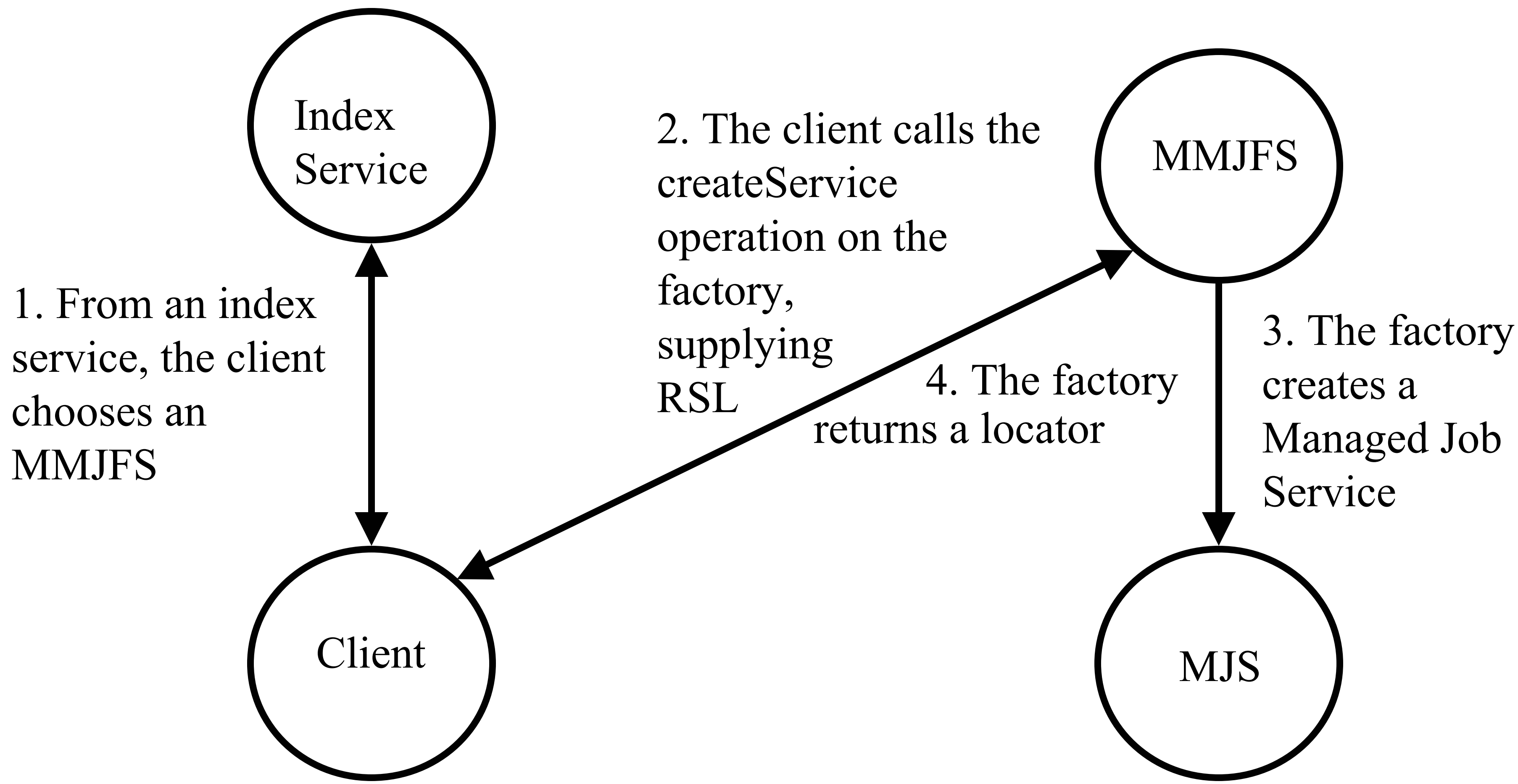


GRAM Job Submission Scenario

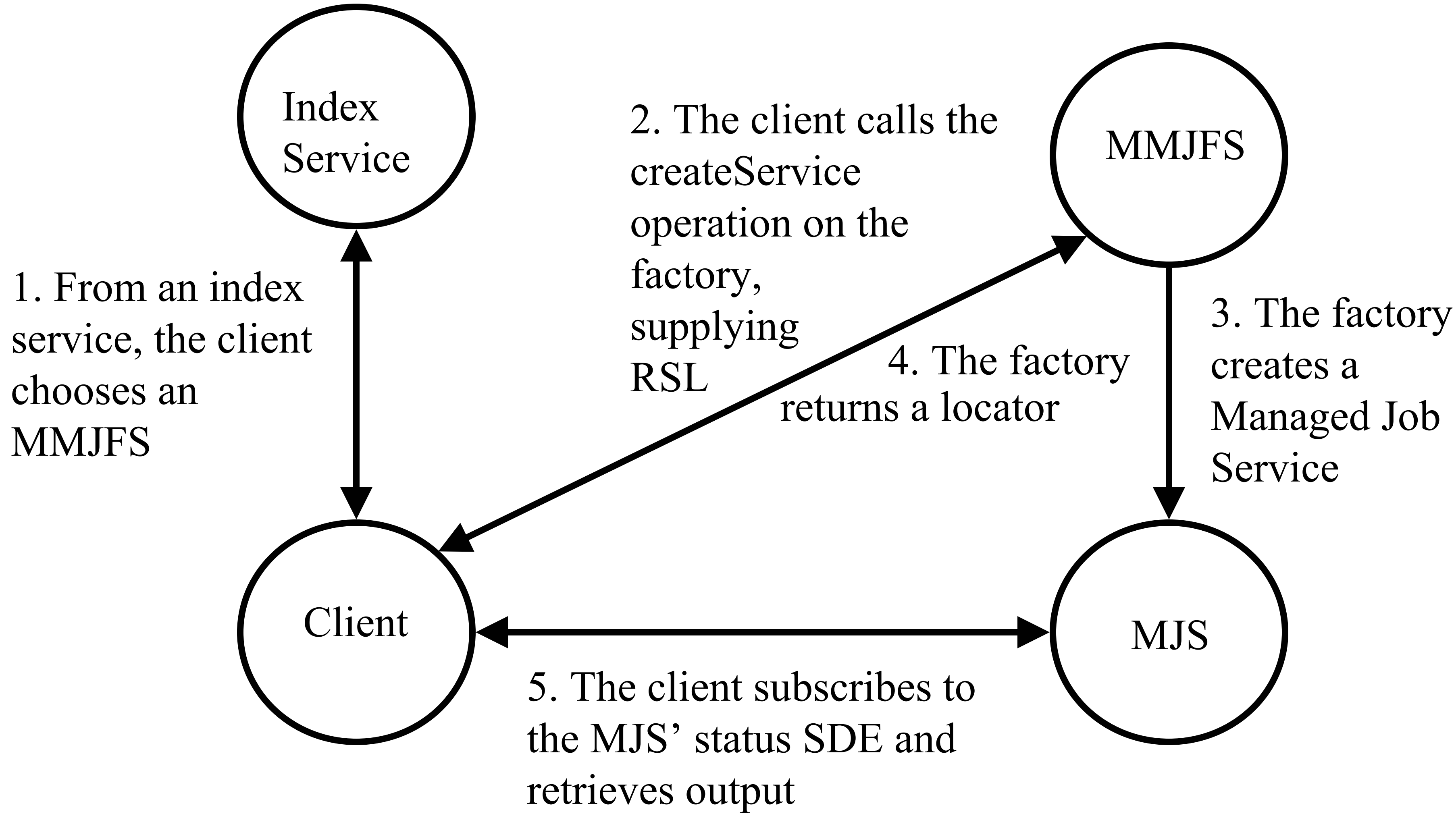


the globus toolkit®
www.globustoolkit.org

GRAM Job Submission Scenario



GRAM Job Submission Scenario

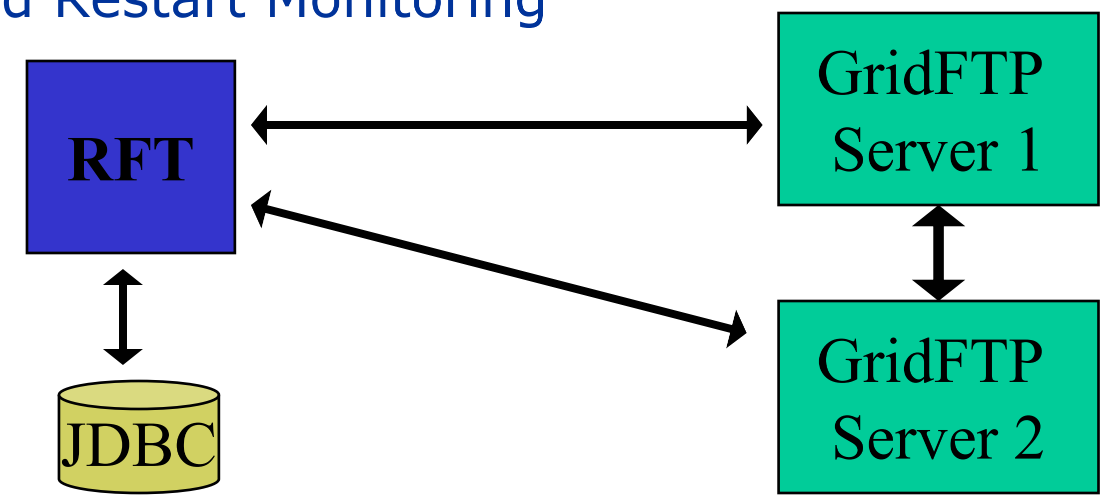


GT3 Base: Information Services

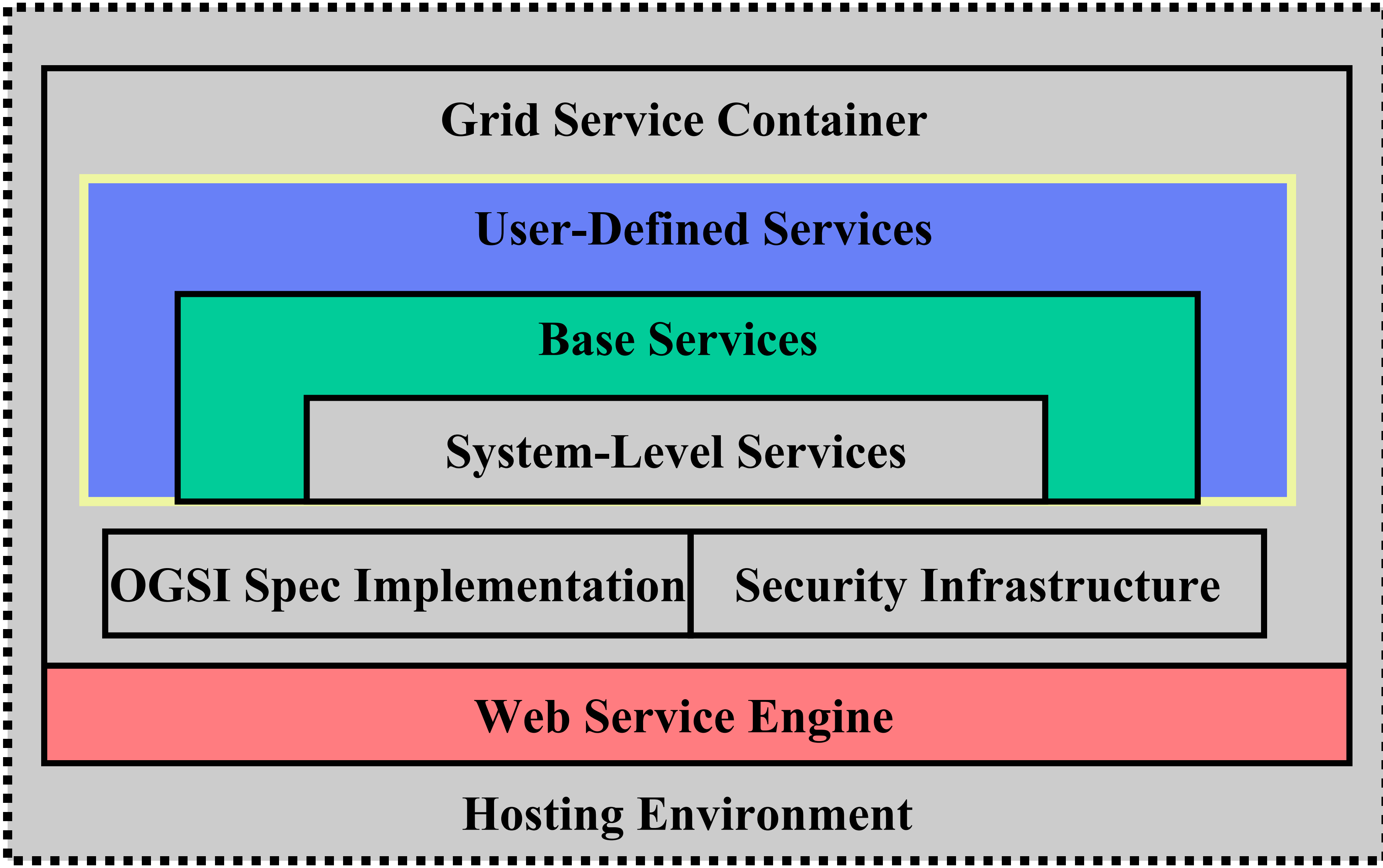
- Index Service as Caching Aggregator
  - Caches service data from other grid services
- Index Service as Provider Framework
  - Serves as a host for service data providers that live outside of a grid service to publish data

GT3 Base: Reliable File Transfer

- Reliably performs a third party transfer between two GridFTP servers
- OGSI-compliant service exposing GridFTP control channel functionality
- Recoverable Grid Service
  - Automatically restarts interrupted transfers from the last checkpoint
- Progress and Restart Monitoring



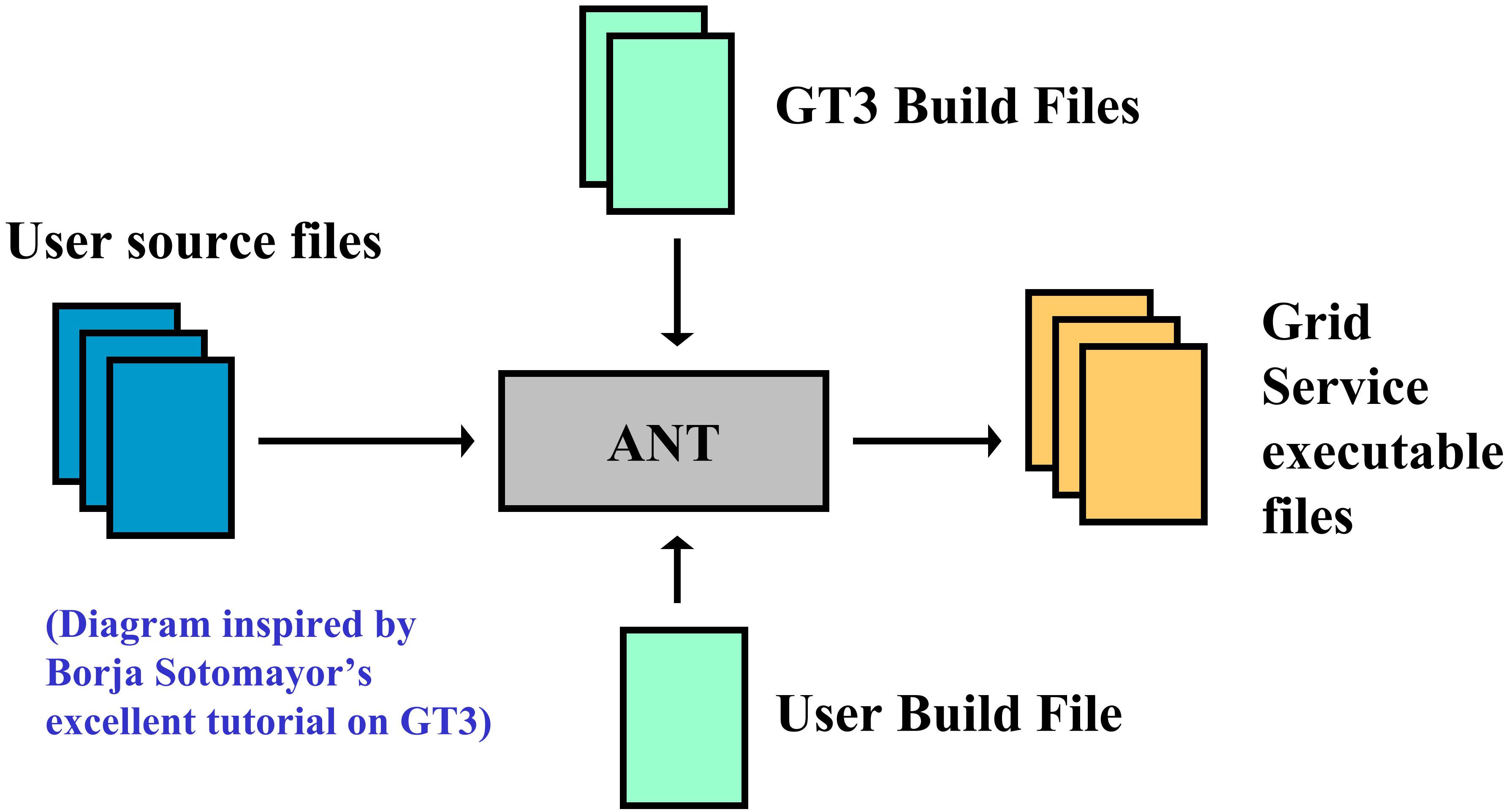
GT-OGSA Grid Service Infrastructure



GT3 User-Defined Services

- GT3 can be viewed as a Grid Service Development Kit that includes:
  - Primitives designed to ease the task of building OGSI-Compliant Services
  - Primitives for provisioning security
  - Base services that provide an infrastructure with which to build higher-level services

GT3 User-Defined Services (cont.)



(Diagram inspired by Borja Sotomayor's excellent tutorial on GT3)

Future Directions of GT

- Standardization of container model
- Development of lightweight container/api
- Adding rich support for queries
- Further refinements of Base Service designs
- Pushing on standardizing at a higher level than OGSI

Introduction to GT3

- Background
  - The Grid Problem
  - The Globus Approach
  - OGSA & OGSI
  - Globus Toolkit
- GT3 Architecture and Functionality: The Latest Refinement of the Globus Toolkit
  - Core
  - Base Services
  - User-Defined Services
  - Future Directions
- **Installation and Administration**
  - Installation
  - Configuration
  - Debugging
  - Support
- Important Things to Remember

Overview

- Installing GT3
- Overview of installed services
- Running clients and services
- Configuring GT3
- Debugging
- Support

Overview of Installing GT3

- Prerequisites
  - JDK, ant
- Optional tools
  - Other hosting environments
- Installation
  - GPT installer or Ant-only
- Required post-installation setup
  - Acquiring certificates
  - Setting permissions

Installation Pre-requisites

- JDK 1.3.1+
  - Xindice + Sun JDK1.3.1 + Linux has errors for some versions of Linux
  - Pre-1.4.1 JDKs require JAAS as a separate download
- Ant 1.5+
  - Required for source builds, recommended for binary
- YACC
  - For cbindings and findServiceData from source
- JDBC compliant database
  - Only required for RFT, RLS

Optional Tools

- Alternate hosting environments
  - Jakarta Tomcat
  - JBOSS
  - Websphere
  - and more …
- Microsoft .NET Framework
- Junit for testing

Installing GT 3.0 (Unix)

- Make sure pre-requisites are available
  - Set JAVA_HOME
  - Add $ANT_HOME/bin to your PATH
- Download the GPT source bundle, or the appropriate binary bundle
- ./install-gt3 /path/to/install
- ./install-gt3-mmjfs /path/to/install
  - After you have certificates

Installing GT 3.0 (Windows)

- Make sure pre-requisites are available
  - Likely to use Cygwin to get the tools you need
- Unset CLASSPATH to avoid conflicting jars
- Install GT3 core by running “ant dist” and “ant setup” in ogsa/impl/java
- Install higher-level services using “ant deployGar”

Installing GT 3.0 (Binaries)

- GPT Binary bundles available for different UNIX platforms
- Core and Higher Level Services binaries available for Windows

Post-installation setup

- GSI uses X.509, so need to get certificates
  - Run setup-gsi as root
  - grid-cert-request for user and host
  - Can re-use GT2 certificates if you have them already
- Run setperms.sh (after install-gt3-mmjfs)
  - This is to make globus-grim setuid to the account which owns the hostcert, and to make the UHE launcher setuid so it can create jobs on behalf of users

Overview of Installed Services

- What just installed, and how?
- Bundles on Unix and Windows:
  - GT3 core + higher-level services
- Bundles on Unix only:
  - GRAM bundle + GT2 dependencies
  - Cbindings bundle + client
  - Replica Location Service (RLS)
  - GT2 components

Other “Services” Bundled with GT3

- GridFTP
  - Used by RFT
- Replica Location Service (RLS)
  - Distributed registry service that records the locations of data copies and allows discovery of replicas
  - Designed and implemented in a collaboration between the Globus and DataGrid projects

The interfaces for these services are not yet OGSI-Compliant

Where did they install?

- /etc/grid-security
  - certificates/ subdirectory of trusted CAs
  - grid-mapfile
  - grim-port-type.xml
  - hostcert.pem, hostkey.pem
  - grid-security.conf
- $GLOBUS_LOCATION
  - Everything else

Location of GARs

- Before the GARs are deployed, a copy is stored in gars/
- Contains the client and server Webservices Deployment Descriptor (WSDD), as well as the jar files
- To change the main server-config.wsdd, can edit the service’s .wsdd file and re-deploy

GPT Wrappers

- The GARs apply to both Windows and Unix
- GPT wraps the GAR with metadata, including dependency information and version number
- Allows for easier upgrades, and for other software to indicate dependencies
- http://www.gridpackagingtools.com/

GT3 Services

- Core
  - OGSI hosting environment
- MMJFS
  - The single point for submitting jobs
- MJS
  - Instances created per submitted job
- GRIM
  - Security tool for creating hostcert proxies

GT3 services (cont.)

- Index
  - A point to query for information
- Aggregator
  - SDE aggregation tool. Used by index
- mds_db
  - Used by index to track data
- Providers
  - Used to obtain host information

GT3 services (cont.)

- RIPS
  - Queue information for jobs
- Filestreaming
  - Used to move stdin/stdout for jobs
- RFT
  - Reliable File Transfer
- JMS
  - JMS notification source for J2SE/J2EE
- Servicegroup
  - OGSI service groups

Dependencies

- Some backend tools see re-use from previous releases for use in resource management
- For instance, jobmanager scripts for interfacing to local scheduling systems are the same
- Besides GT2, Java CoG supplies integrated security

Overview of Running Clients and Services

- Set your environment
- Create a proxy (single sign-on)
- Available clients
  - GRAM client
  - Index clients
  - RFT client

Environment Setup

- Set your environment:
  - export GLOBUS_LOCATION
  - source etc/globus-user-env.sh
- grid-proxy-init
- This environment setup is assumed for all later slides

Starting the container

- Services run in the container
- bin/globus-start-container –p <port>
- The container will print a list of Grid Service Handles (GSH) that can be used by clients
  - http://127.0.0.1:8080/ogsa/services/base/gram/MasterForkManagedJobFactoryService for instance
- User's Guide has more details

Running the GRAM Client

- To submit a job:
  - bin/managed-job-globusrun –factory host{:port/service} -file etc/test.xml
- Prerequisites:
  - Authorized to use the service by grid-mapfile and grim-port-type.xml
  - Have a proxy
  - Setuid GRIM and User Hosting Environment (UHE) launcher

Running the GRAM Client (Cont.)

- etc/test.xml contains the new format for the Resource Specification Language (RSL)
- By default, outputs to /tmp/stdout /tmp/stderr
  - Probably good to customize it for yourself to avoid permissions errors
- See Resource Management links under http://www-unix.globus.org/developer

Reading From the Index Service

- bin/globus-service-browser
- bin/ogsi-find-service-data
  - requires C bindings
- Prerequisites:
  - Have to add ServiceDataProviderExecutionPortType and ServiceDataAggregatorPortType to client-gui-config.xml
- See Information Services for more details

Using RFT

- First, start a GridFTP server
- Setup a postgres database for RFT to use
- Enter DB values into server-config.wsdd
- java org.globus.ogsa.gui.RFTClient <RFT factory> <path to transfers>
- Store transfers in a file, one URL per line
- See Data Management for more details

Using RLS

- globus-rls-server
  - New version uses PostgreSQL and psqlODBC
- bin/globus-rls-admin -p rls://serverhost
  - ping test of server
- http://www.globus.org/rls/

Configuring GT3

- Add backend schedulers to MMJFS
  - PBS, LSF, Condor, …
  - These will be GPT setup packages. Run gpt-install and gpt-postinstall
- Configure a hierarchy of index services
- Adding new services
  - ant deployGar –Dgar.name=/path/to/gar
- Check GTR for new services, and consider publishing your own (gtr.globus.org)

Debugging

- Always go to the most basic levels first!
- If “connection refused” try telnet
- If “DB connection refused” try your native DB client first
- If trouble with your proxy, try
  - grid-proxy-init –verify –debug
- If trouble with “policy” or GRIM, try
  - bin/globus-grim –out /tmp/grim_test

Debugging (Cont.)

- Logfiles
  - Make sure to redirect container logs to a file
  - ~user/.globus/uhe-<host>/log contains the logs for the UHE running on <host>
  - GridFTP servers use syslog for keeping logs
- Increase debugging level in ogsilogging.properties file

Support

- See
  - http://www.globus.org/toolkit/support.html
- It gives an overview of documentation, mailing lists, and bugzilla

Introduction to GT3

- Background
  - The Grid Problem
  - The Globus Approach
  - OGSA & OGSI
  - Globus Toolkit
- GT3 Architecture and Functionality: The Latest Refinement of the Globus Toolkit
  - Core
  - Base Services
  - User-Defined Services
  - Future Directions
- Using GT3
  - Installation
  - Configuration
  - Debugging
  - Support
- **Important Things to Remember**