Grid Services Overview

Keith R. Jackson
Distributed Systems Department
Lawrence Berkeley National Lab

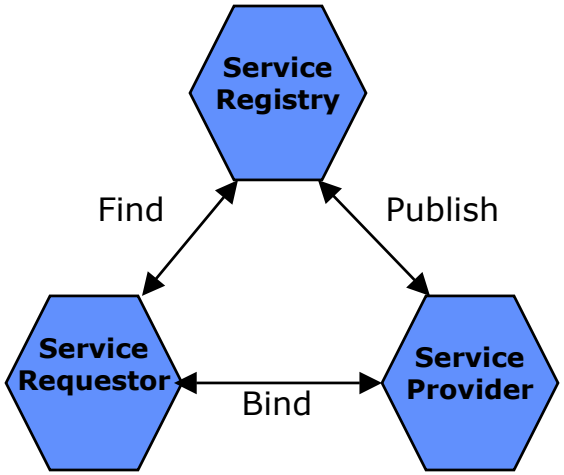
Overview

- Web Service Overview
- WSDL Document Review
- Web Service Evaluation
- Web Services and Grids
- Virtualizing Services
- Open Grid Services Architecture (OGSA) Overview
- WSDL Extensions and Conventions
- Standard OGSA Interfaces
- Grid Portals
- Contacts & Acknowledgements

What is a Web Service?

- Web service is an entity that can be:
  - Described (using WSDL)
  - Published
  - Discovered
  - Invoked by a client
- W3C technology standardization process
- Often associated with specific technologies and implementations
  - Standards: XML, WSDL, SOAP, UDDI
  - Implementations: WebSphere, .NET, others…

Service Oriented Architecture



- Publish
  - WSDL: Web Services Description Language
  - UDDI: Universal Description, Discovery & Integration
- Find
  - WS-Inspection
- Bind
  - SOAP: Simple Object Access Protocol

WSDL Document Structure

- WSDL: Web Services Definition Language
- Document structure:
  - — Service Description
  - — Implementation Details
- Service Description
  - — Elements
    - PortType (~ class)
    - Operations (~ method)
    - Messages, message parts (~ parameters)
    - Types (type definitions)
  - — Used for
    - Generating stubs and skeletons
    - Service discovery

WSDL Document Structure (cont.)

- Implementation Details
  - — Binding
    - Messaging protocol (e.g. SOAP)
    - Message Interpretation (e.g. RPC or literal)
    - Data-encoding model (e.g. SOAP or literal encoding)
    - Transport protocol (e.g. HTTP or FTP)
  - — Port: describes service endpoint
  - — Service Element: groups Port elements together

Web Service Evaluation (+)

- Key to success:
  - — Emphasize protocols rather than APIs
  - — Build on established technologies and protocols
  - — Web-wide rather than enterprise-wide scope
  - — A set of independent technologies
  - — Industry support

Web Service Evaluation (-)

- Developing technology:
  - — Lack of standard language bindings
  - — Others
- Web Services applied to Grids:
  - — WS describe persistent stateless services
    - For Grids we must also support transient instances
      - — Implies a need to uniquely name instances
    - Lifecycle management issues
  - — Need to provide information about a service
    - Need ways to access that information
  - — Implications on how services are managed

WS & Grids

- Service orientation
  - — virtualize resources
  - — unify resources/services/information
- Capitalize on useful WS properties
  - — Standards for service description and discovery
  - — Leverage commercial efforts
- Refactor Globus protocol suite to enable common base and expose key capabilities
- Provide a unifying architecture for computational Grids

Globus Toolkit Refactoring

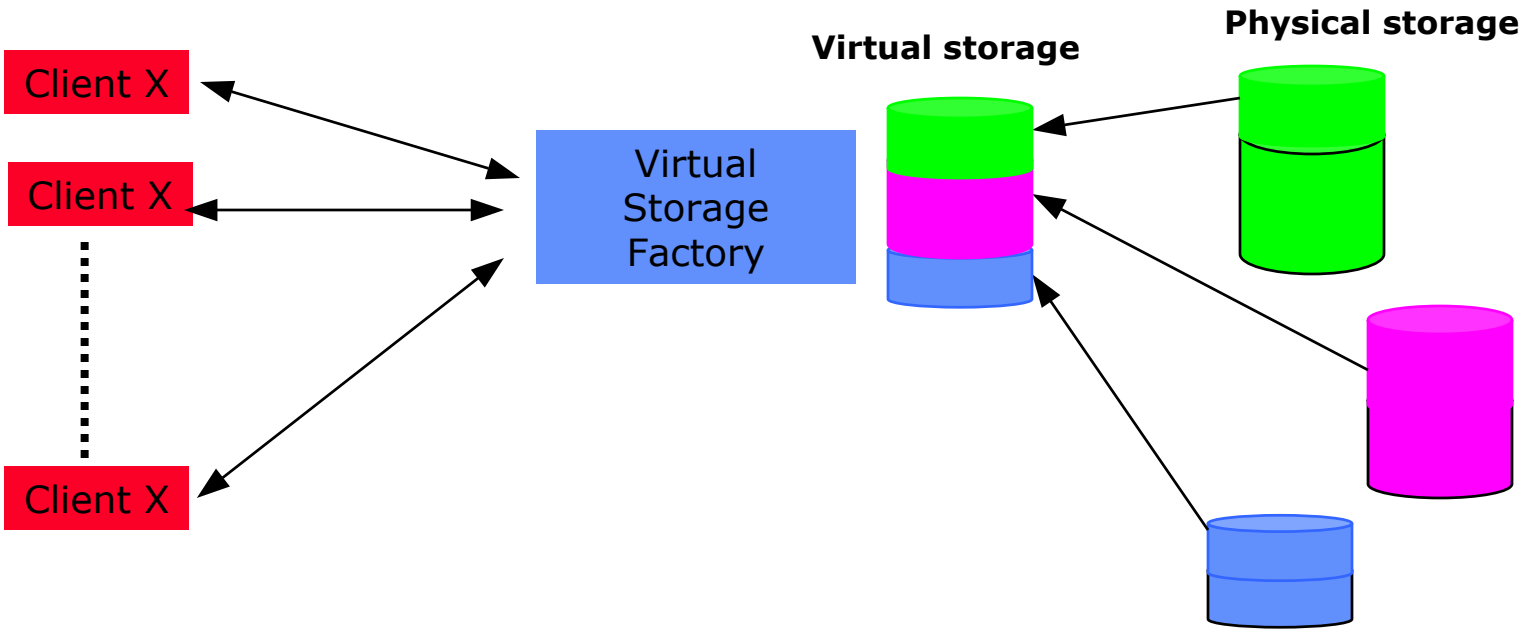
- Grid Security Infrastructure (GSI)
  - — Used in Grid service network protocol bindings
  - — Also: Security Services
- Meta Directory Service 2 (MDS-2)
  - — Native part of each Grid service:
    - Discovery, Notification, Registry, RegistryManagement
- Grid Resource Allocation & Mngt (GRAM)
  - — Job Manager Service
  - — Gatekeeper -> Factory for job mgr instances
- GridFTP
  - — Refactor control channel protocol
- Other services refactored to used Grid Services

Moving Forward with Grid Services

- Benefits of service orientation
  - — Focus on interface
    - Minimal shared understanding between interacting entities
  - — Local/remote transparency
  - — Modularity, Reusability, etc.
- Virtualization
  - — Encapsulation of diverse implementation behind a common interface
  - — Defining interactions with services in terms of QoS constraints and Service Level Agreements (SLA)
  - — Living up to SLAs: Adaptive behaviors

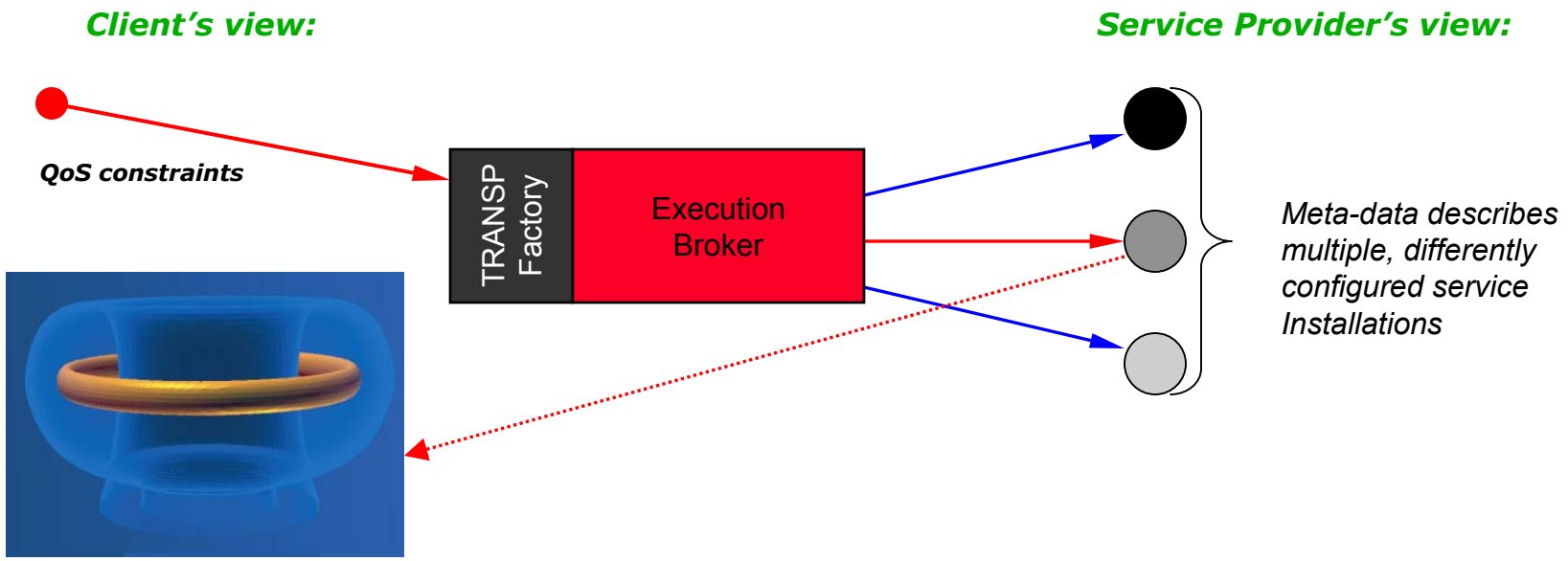
Virtualizing Resources: Example

- Application: Virtual Storage
  - Garbage collecting unused space in an organization
  - Providing it to users as “virtual storage”

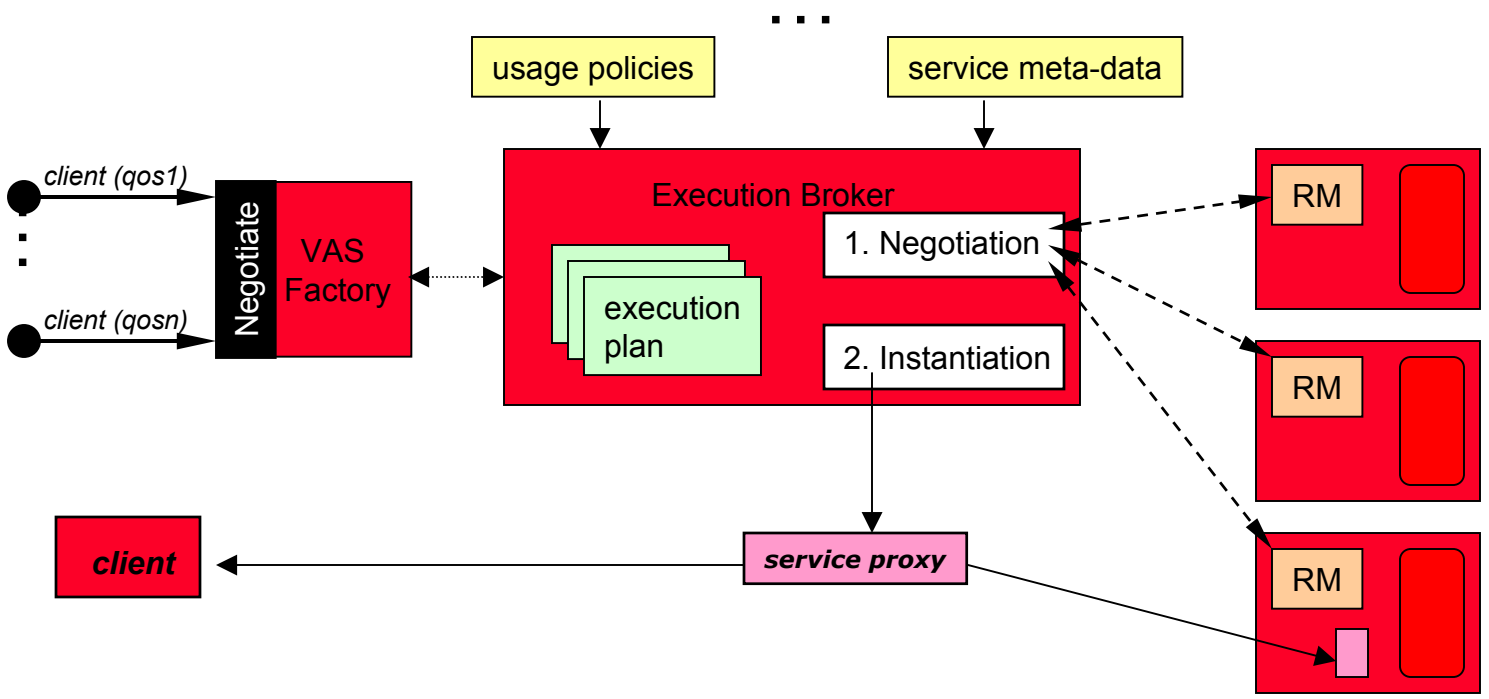


Virtual Application Services (VAS)

- Example: the National Fusion Collaboratory
- Requirements
  - Codes as “network services” (portability reasons)
  - Different interaction modes
    - Real-time constraints (betw. Experimental pulses: ~15mins)
    - Batch jobs where accuracy is important



VAS: Behind the scenes



- Adaptive capabilities
- Capable of adjusting to different models

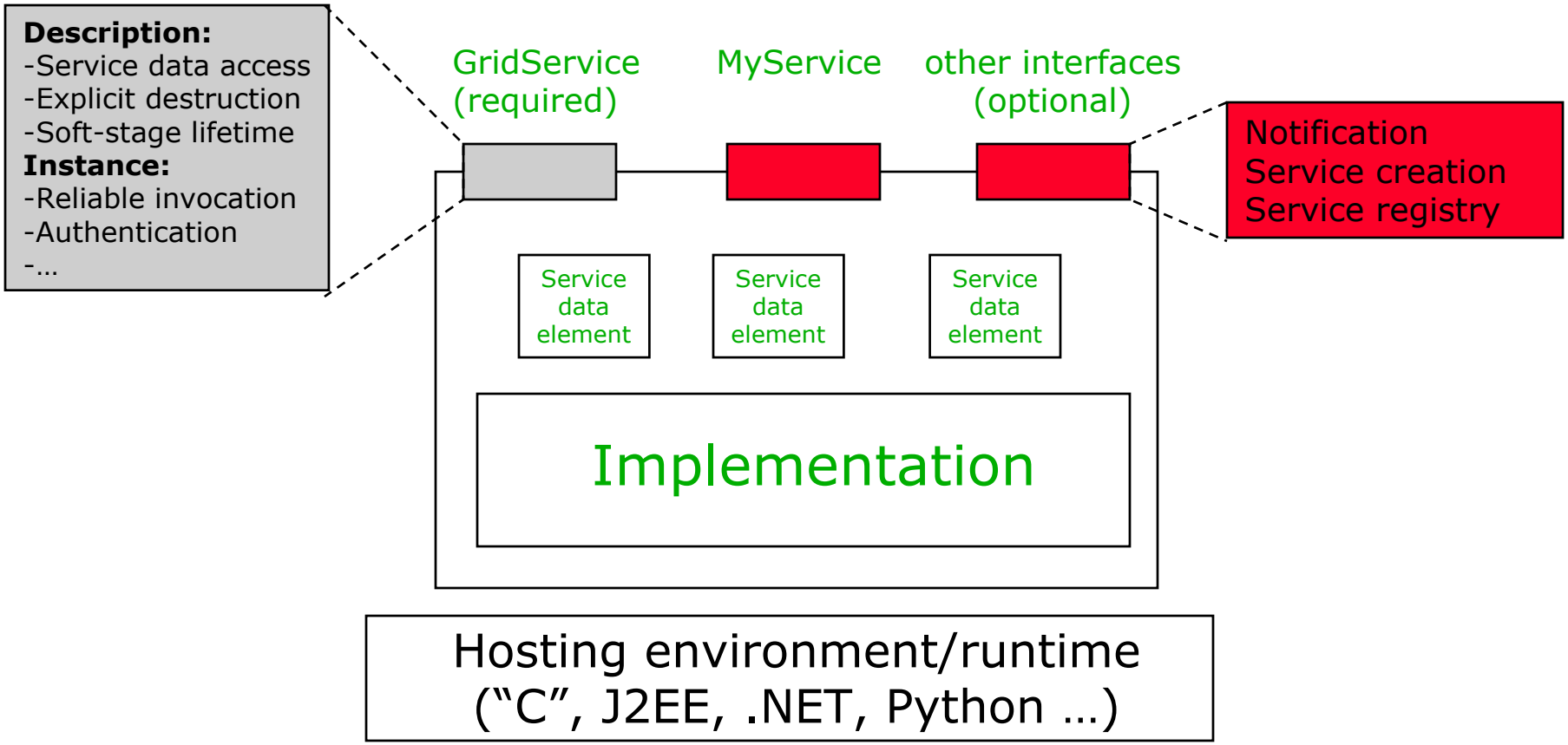
Composing Services

- Resource composition
  - — Complex resource configuration
  - — CPUs, networking, storage…
  - — Redundant configuration to provide for failure
- Application Service Composition
  - — Workflow and orchestration
  - — Constraint-based service discovery
  - — Reliable and Adaptive Workflow execution
  - — Reproducibility
    - Data provenance

Open Grid Services Architecture

- From Web services
  - — Standard interface definition mechanisms
    - Interface and implementation (multiple protocol bindings)
    - local/remote transparency
    - Language interoperability
  - — A homogenous architecture basis
- From Grids
  - — Service semantics
  - — Lifecycle management
  - — Reliability and security models
  - — Discovery
  - — Other services: resource management, authorization, etc.

The Grid Service



The Grid Service

- A WSDL-defined service that conforms to a set of conventions relating to its interface and behaviors
- Description composed of two parts:
  - Grid service description
    - Describes how a client can interact with service instances: syntax and semantics (portTypes)
    - Can be used by any number of GS instances
  - Grid service instance
    - Embodies state
    - Has one or more unique Grid Service Handles
    - Has one or more Grid Service References

Grid Service Example

```
<portType name="GridServicePortType">
  <operation name="findServiceData">
    <input message="tns:FindServiceDataInputMessage"/>
    <output message="tns:FindServiceDataOutputMessage"/>
    <fault name="QueryNotSupportedFault" message="ogsa-faults:QueryNotSupportedFault"/>
    <fault name="InvalidQueryFault" message="ogsa-faults:InvalidQueryFault"/>
  </operation>
  <operation name="setTerminationTime">
    …
  </operation>
  <operation name="getTerminationTime">
    …
  </operation>
  <operation name="destroy">
    …
  </operation>
</portType>
```

Grid Service Functionality

Database_PortType
Inherits from GridService

```
<portType name="Database_PortType" extends "gsdl:GridService">
      <operation name="databaseQueryOperation">
            <input message="tns:myDatabaseQuery"/>
            <output message="tns:myDatabaseResponse"/>
      </operation>
</portType>
```

OGSA Structure

1) WSDL conventions and extensions for describing and structuring services
   — Useful independent of “Grid” computing
2) Standard WSDL interfaces & behaviors for core service activities
   — Necessary for Grid computing
3) Higher-level services

WSDL Extensions and Conventions

- Defined using WSDL extensibility elements
- WSDL conventions and extensions
  - serviceData: properties of a service that may be queried
  - serviceDataDescription:formal description of serviceData elements
  - portType inheritance: recently added to WSDL
    - Extending portTypes
  - Naming conventions on portType and serviceType
  - Grid Service Reference (can be a WSDL document)
  - Grid Service Handle

Service Data

- Describes
  - Meta-data (info about the service instance)
  - State data (runtime properties)
- Service Data Element (SDE)
  - Name
  - Type (XML type)
  - Extensibility attributes
    - Lifetime declarations
      - goodFrom, goodUntil, availableUntil
    - Application-specific
  - Extensibility elements
    - Service data value
    - Application-specific

Service Data Set

- A set of SDEs
- Each Grid Service must have exactly one service Data Set
- Accessible in two ways:
  - FindServiceData
  - Notification

Naming and Change Management

- The change management problem
  - GS semantics may evolve
    - On the interface level: adding new operations
    - On the implementation level: bug fixes, etc.
  - Users rely on this behavior
- OGSA requirement: all elements of a GS description must be immutable
  - Qualified name (namespace and locally unique name) must refer to only one WSDL specification
  - If a change is needed a new service with a new qualified name must be defined

Handles and References

- Grid Service Handle (GSH)
  - Uniquely identifies a service
  - Has the form of URI
- Grid Service Reference (GSR)
  - Contains all the information a client needs in order to communicate with a service
  - Its form depends on the binding
- GSH must be resolved to GSR in order to use a service
  - Information on how to resolve encoded in the URI
- Separation of name from implementation details facilitates manipulation of a service

Grid Service Handle

- Name in the form of URI
  - The URI scheme defines the protocol for resolving it
- Properties
  - GSH is valid for the lifetime of a GS instance
  - Must not refer to more than one service instance
  - A GS has at least one GSH
  - GSH may resolve to different GSRs pointing to the same service
- Resolver protocols
  - Untrusted (http)
  - Trusted (https)

Grid Service Reference

- Network-wide pointer to a specific GS instance
  - Web service binding mechanism
  - Binding-specific information about the endpoint
  - May include expiration time (treat is as a hint)
- Binding-specific
  - SOAP: WSDL document
  - RMI/IIOP: CORBA-compliant IOR
- May become invalid during the lifetime of an instance (independent lifecycle)
- Many GSRs to a service may exist at the same time
- Use of invalid GSR should result in an exception

Standard Interfaces

- Grid Service: basic behavior
- HandleResolver: mapping from GSH to GSR
- Lifecycle
  - Support transient services
  - Service instances created by factories
  - Destroyed explicitly or via soft state
- Notifications
  - Registering interest and delivering notifications
- Registration
  - Allows clients to register and unregister registry contents

Grid Service Interface

- Must be implemented by all Grid services
- Interface:
  - FindServiceData
    - Input
      - QueryExpressionType: query mechanism used
      - QueryExpression: actual query
    - Output
      - Result of Query
  - SetTerminationTime
    - Request that termination time of this service be changed
    - Input: client timestamp and new termination time
    - Output: service timestamp and current termination time
  - Destroy
    - Explicit destruction request, returns an ack

Handle Resolver

- Resolves GSH into GSR
  - — Optionally, the client can do it by itself
- Interface
  - — FindByHandle
    - Input: GSH & unsatisfactory GSRs
    - Output: GSR
    - Faults: invalidHandle, no valid references, etc.

Lifecycle

- GS instances created by factory or manually
- Destroyed explicitly or via soft state
  - Negotiation of initial lifetime with a factory (service supporting Factory interface)
  - Lifetime can subsequently be extended by sending “keepalive” messages
- Soft state lifetime management avoids
  - Explicit client teardown of complex state
  - Prevents resource “leaks” in hosting environments

GS Creation: Factory

- Creates a new service instance
  - — Reliable once and only once creation
- Interface
  - — CreateService
    - Input:
      - — TerminationTime
      - — ServiceParameters (specific to a service)
    - Output: ServiceTimestamp information & Service Locator
- ServiceLocator can be used to obtain GSH

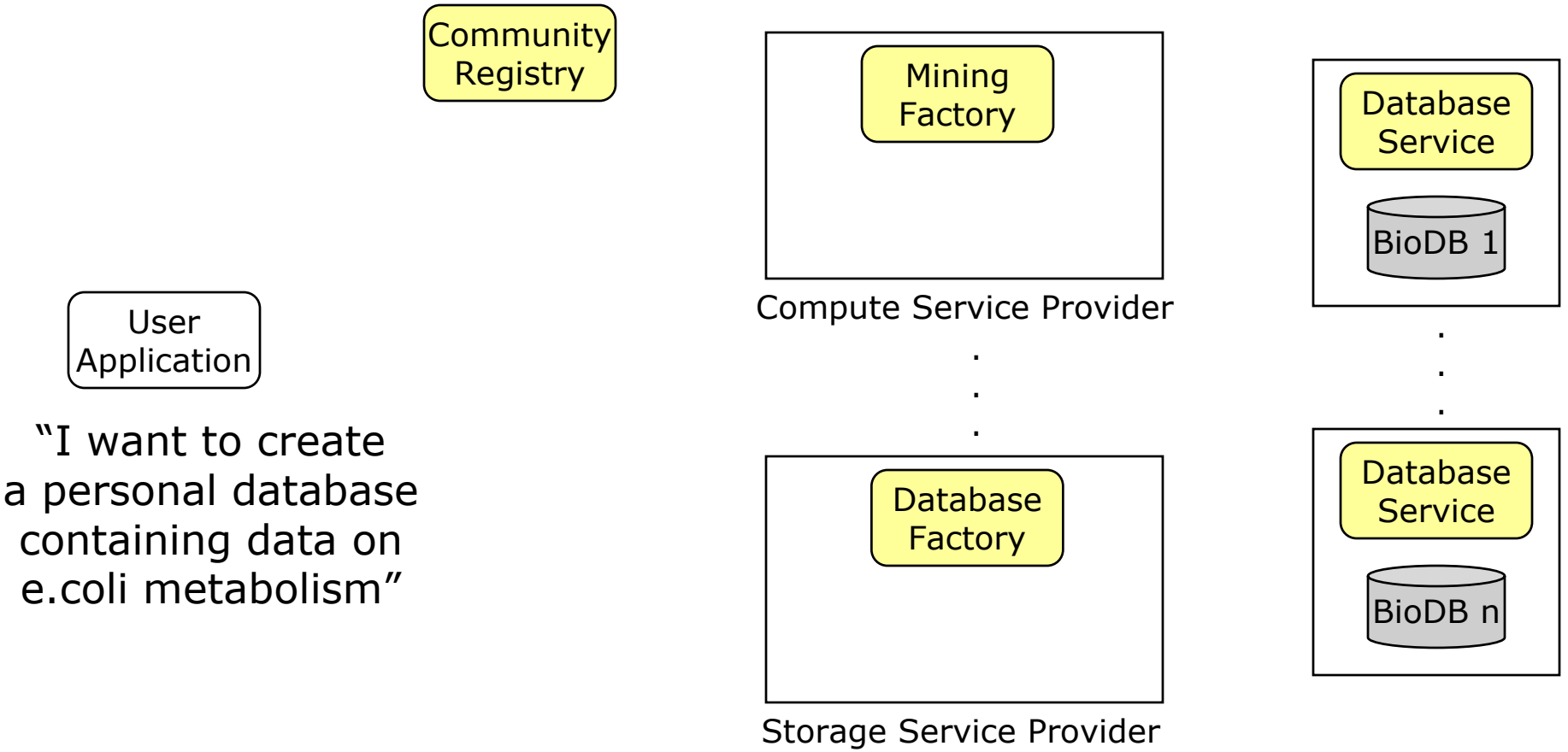
Grid Service Termination

- Explicit destruction
  - — Destroy operation in the Grid Service
- Soft-state destruction
  - — Allowing the termination time to expire
  - — SetTerminationTime operation resets the value of the TerminationTime SDE
  - — Reaffirmation of interest does not guarantee that the service will stay alive

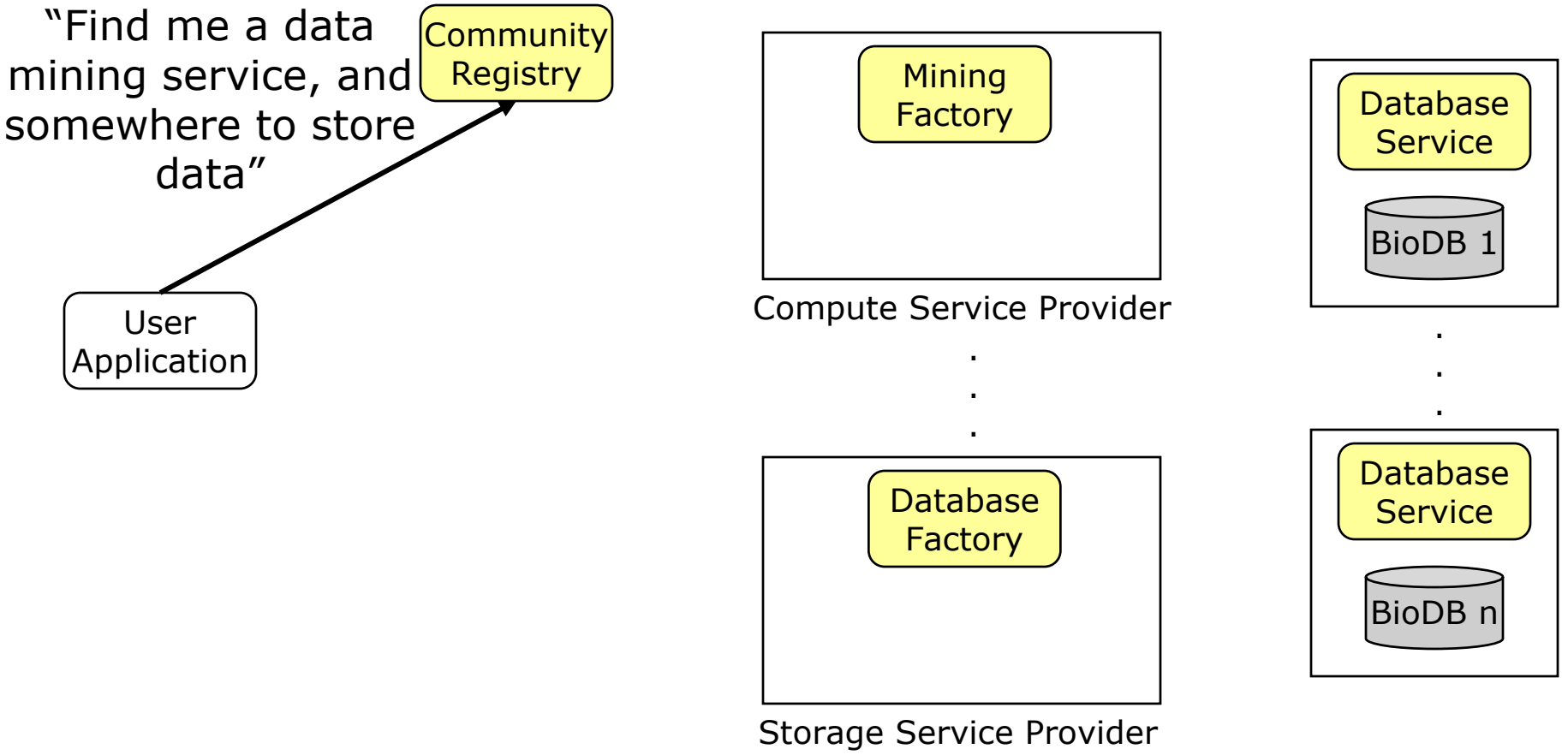
Registry

- The Registry interface may be used to register Grid service instances with a registry
  - A set of Grid services can periodically register their GSHs into a registry service, to allow for discovery of services in that set
- Registrations maintained in a service data element associated with Registry interface
  - Standard discovery mechanisms can then be used to discover registered services
  - Returns a WS-Inspection document containing the GSHs of a set of Grid services

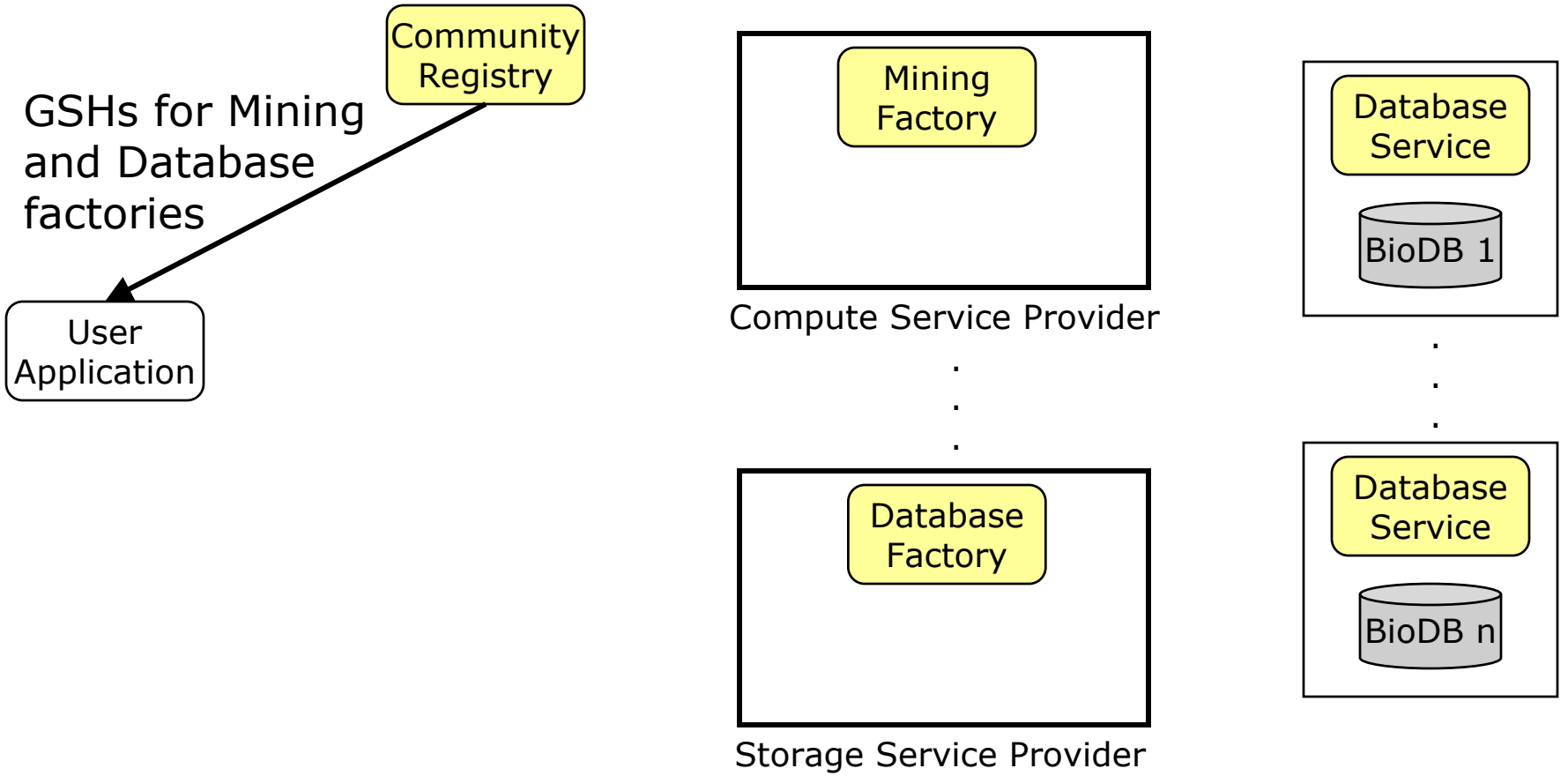
Example: Data Mining for Bioinformatics



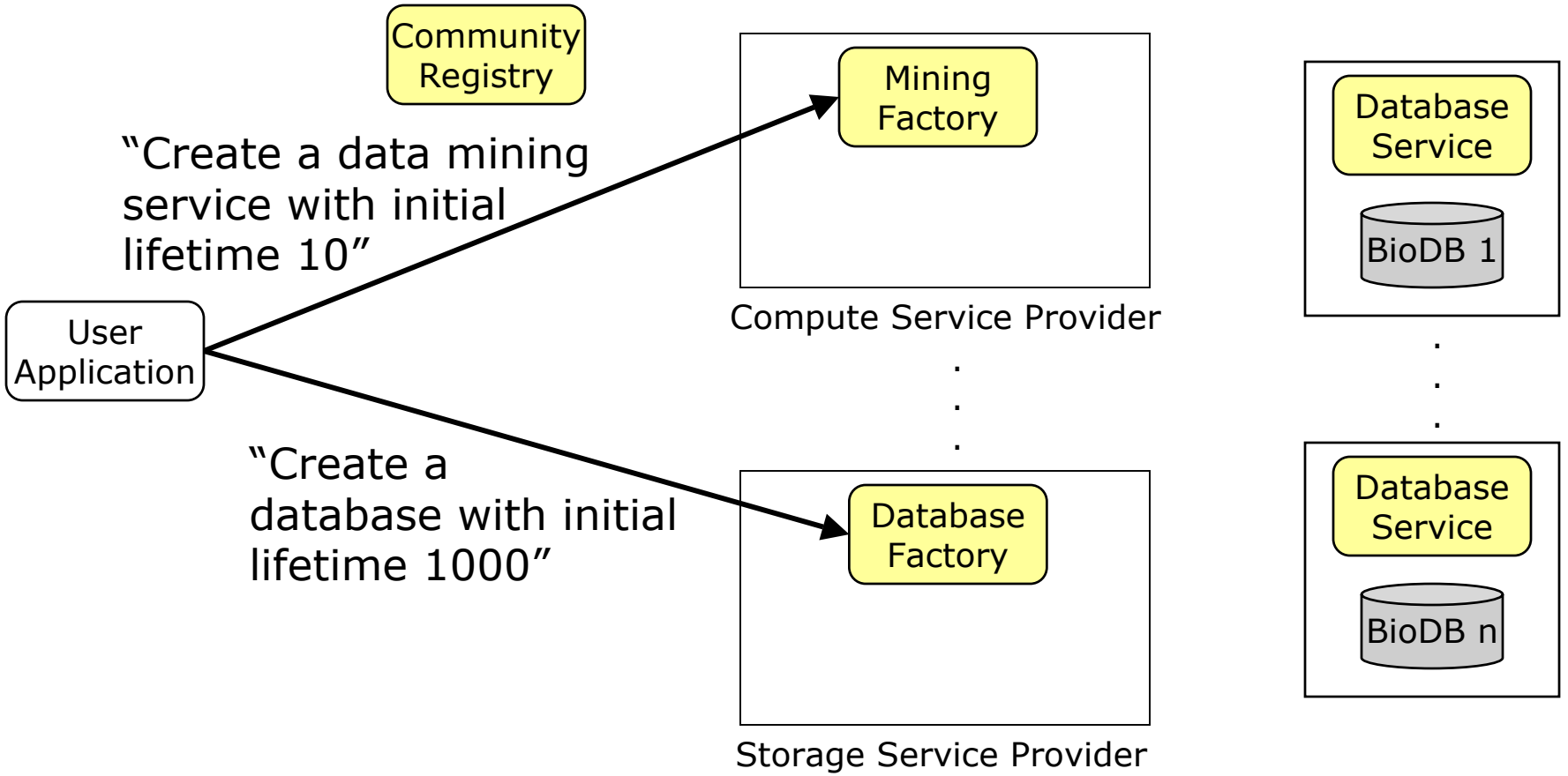
Example:
Data Mining for Bioinformatics



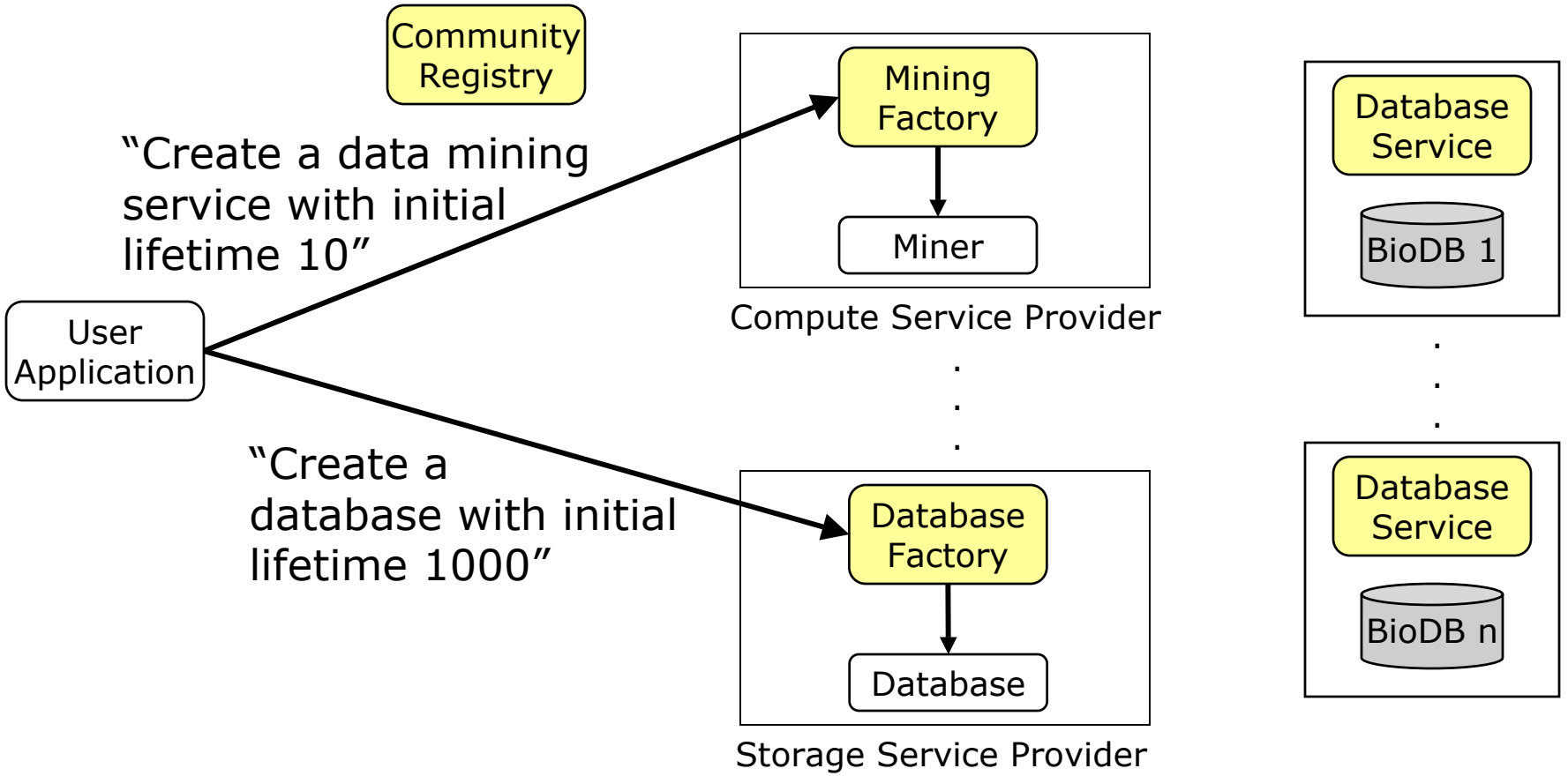
Example:
Data Mining for Bioinformatics



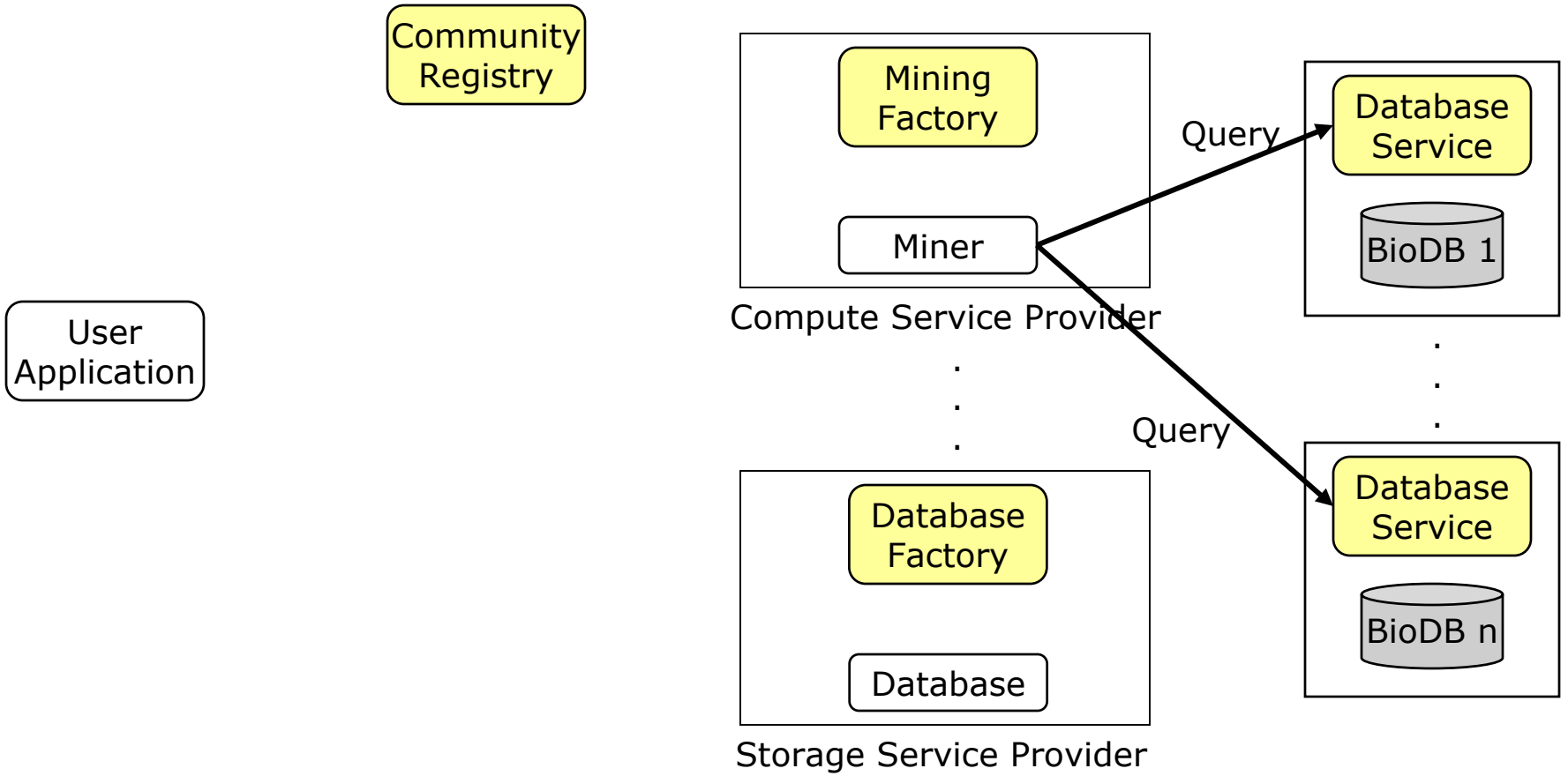
Example:
Data Mining for Bioinformatics



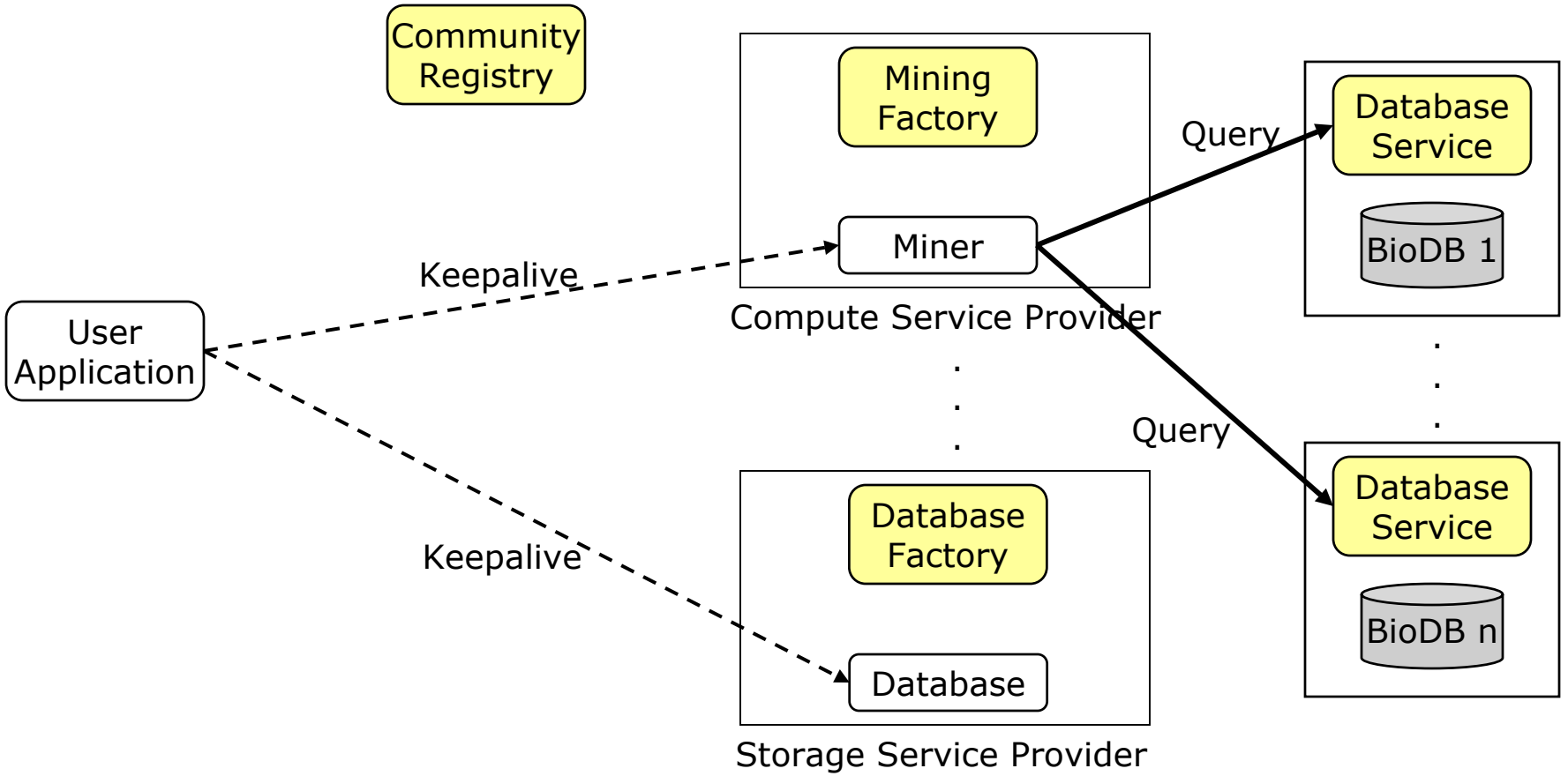
Example:
Data Mining for Bioinformatics



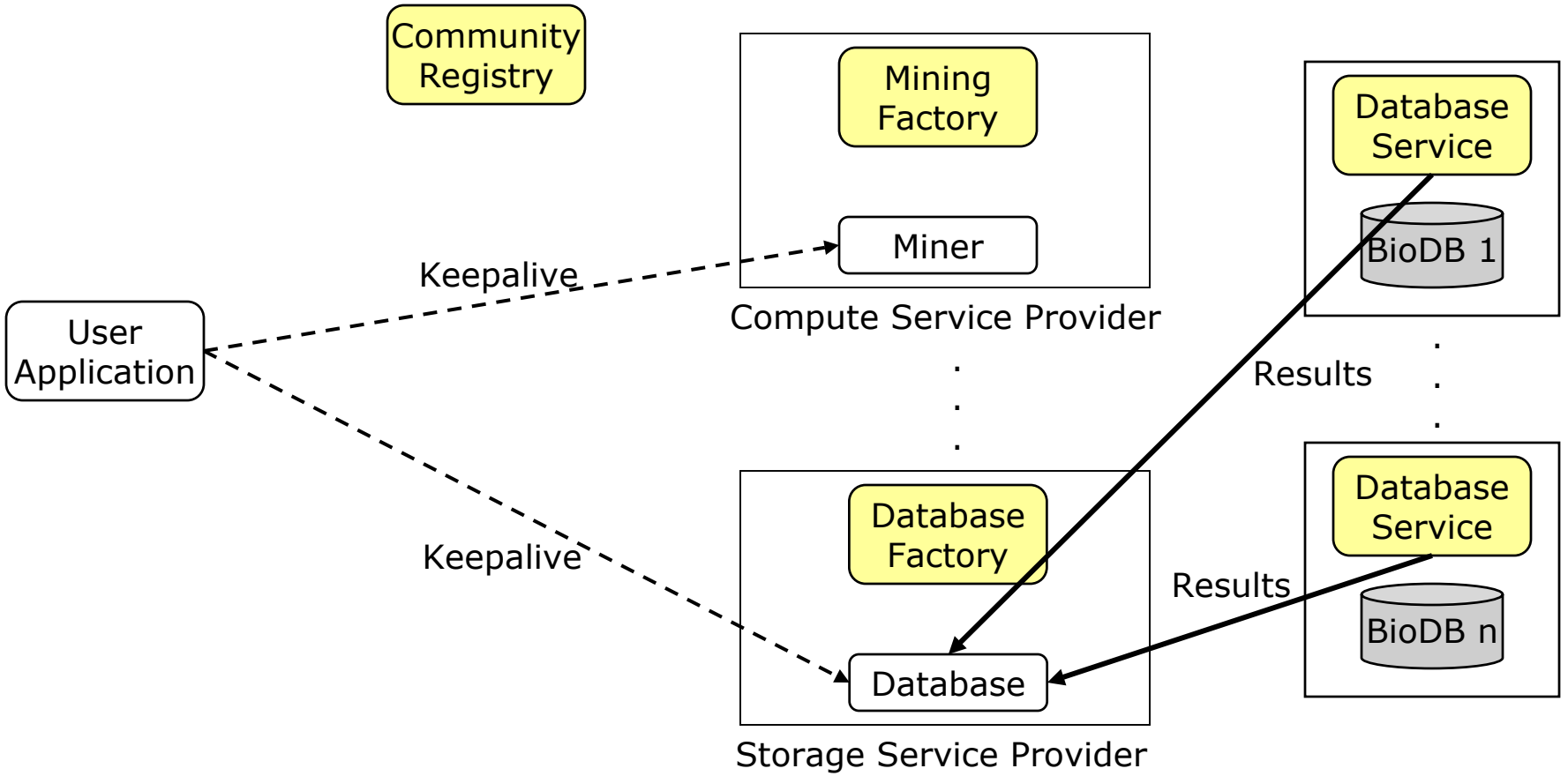
Example:
Data Mining for Bioinformatics



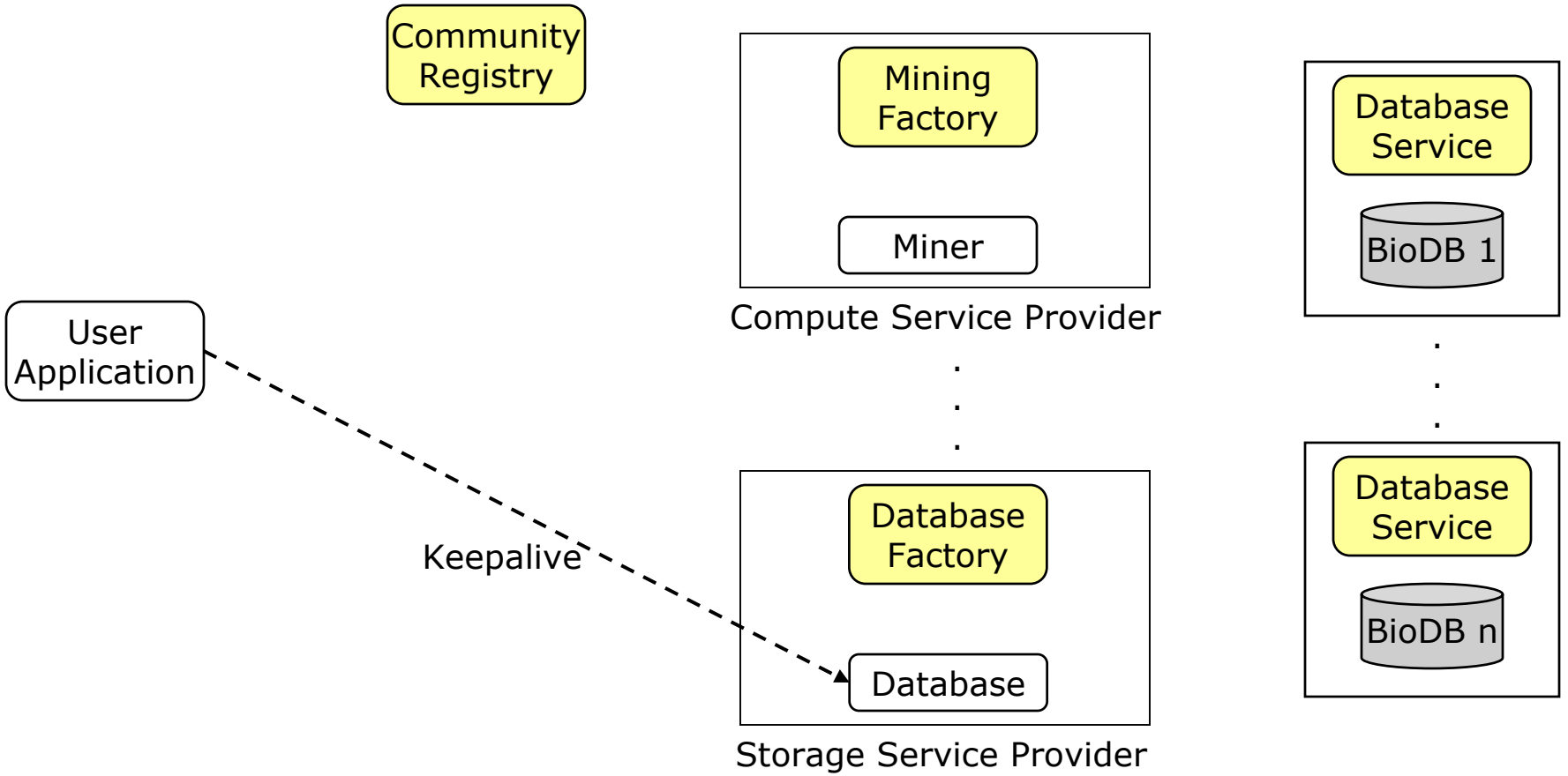
Example:
Data Mining for Bioinformatics



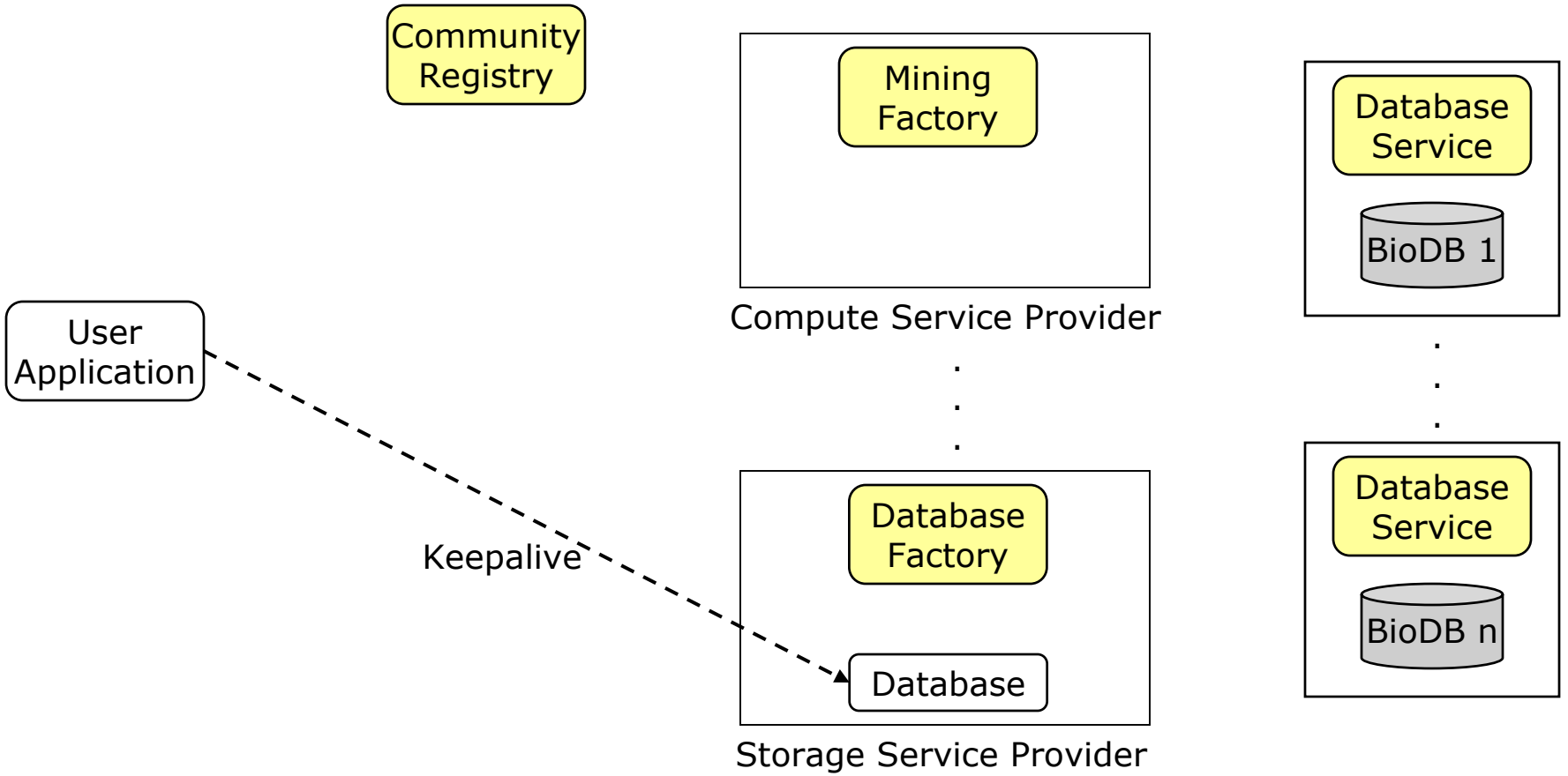
Example:
Data Mining for Bioinformatics



Example:
Data Mining for Bioinformatics



Example:
Data Mining for Bioinformatics



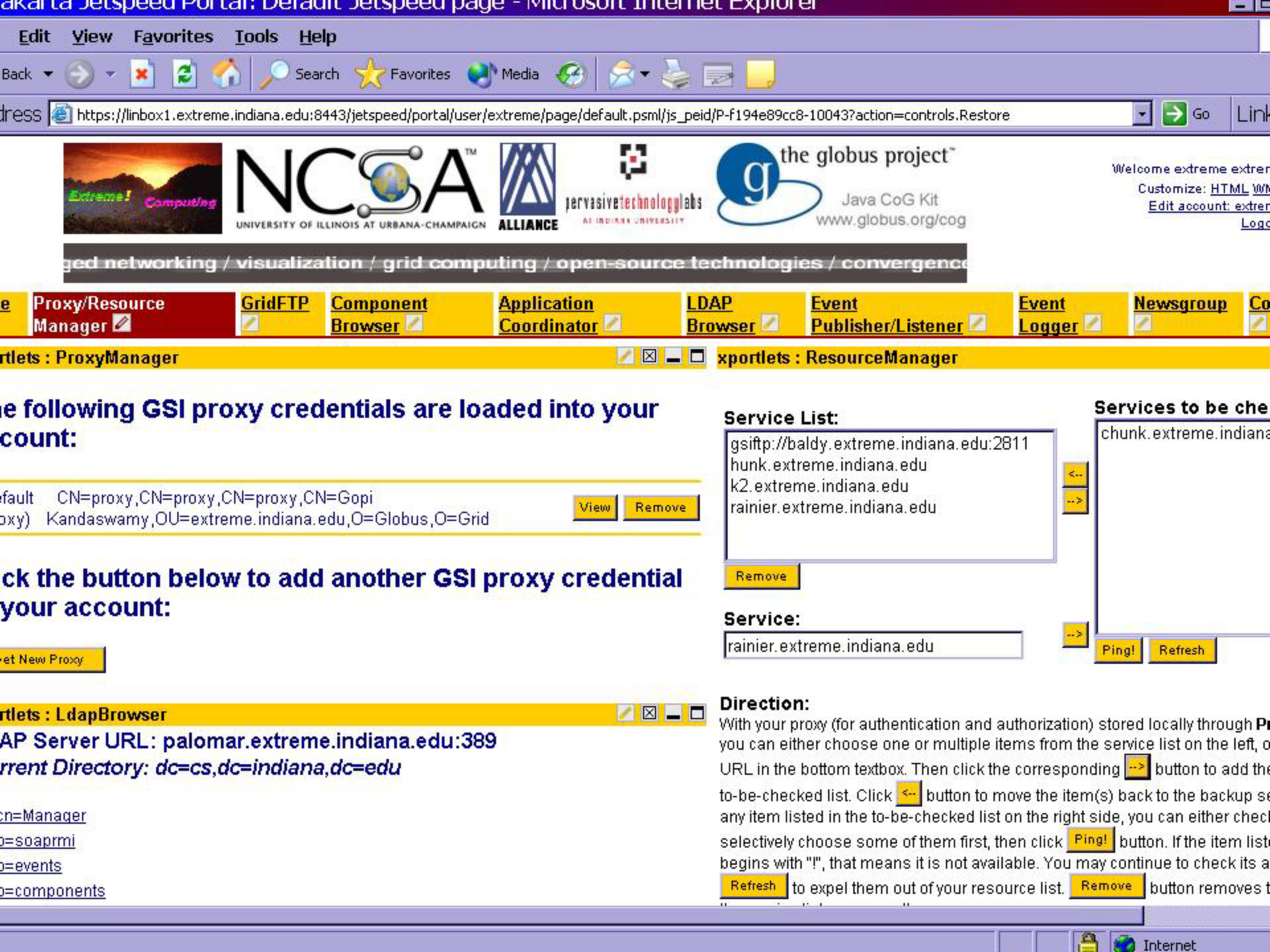
Notification Interfaces

- NotificationSource for client subscription
  - — One or more *notification generators*
    - Generates notification message of a specific type
    - Typed *interest statements*: E.g., Filters, topics, …
    - Supports messaging services, 3rd party filter services, …
  - — Soft state subscription to a generator
- NotificationSink for asynchronous delivery of notification messages
- A wide variety of uses are possible
  - — E.g. Dynamic discovery/registry services, monitoring, application error notification, …

Grid Portals

- A Grid Portal
  - An access point and a programming system for using and building grid applications.
- Two basic types
  - Grid Access Portals
    - Provides tools to access standard grid resources and launch and manage jobs
  - Science Portals
    - Provides a domain specific view of a set of applications that run on the grid.
- Accesses Grid Services and uses portlets to provide an interface separate from the service.

Courtesy of Dennis Gannon


akarta Jetspeed Portal: Default Jetspeed page - Microsoft Internet Explorer
Edit View Favorites Tools Help
Back Search Favorites Media
dress https://linbox1.extreme.indiana.edu:8443/jetspeed/portal/user/extreme/page/default.psml/js_peid/P-f194e89cc8-10043?action=controls.Restore
Go
Extreme! Computing
NCSA
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
ALLIANCE
pervasivetechnologylabs
the globus project
Java CoG Kit
www.globus.org/cog
Welcome extreme extren
Customize: HTML WM
Edit account: extren
Logo
ged networking / visualization / grid computing / open-source technologies / convergence
e
Proxy/Resource Manager
GridFTP
Component Browser
Application Coordinator
LDAP Browser
Event Publisher/Listener
Event Logger
Newsgroup
Co
rtlets : ProxyManager
xportlets : ResourceManager
e following GSI proxy credentials are loaded into your count:
efault CN=proxy,CN=proxy,CN=proxy,CN=Gopi
oxy) Kandaswamy,OU=extreme.indiana.edu,O=Globus,O=Grid
View
Remove
ck the button below to add another GSI proxy credential your account:
et New Proxy
Service List:
gsiftp://baldy.extreme.indiana.edu:2811
hunk.extreme.indiana.edu
k2.extreme.indiana.edu
rainier.extreme.indiana.edu
Remove
<--
-->
Services to be che
chunk.extreme.indiana
Service:
rainier.extreme.indiana.edu
-->
Ping!
Refresh
rtlets : LdapBrowser
AP Server URL: palomar.extreme.indiana.edu:389
rrent Directory: dc=cs,dc=indiana,dc=edu
cn=Manager
o=soaprmi
o=events
o=components
Direction:
With your proxy (for authentication and authorization) stored locally through Pr
you can either choose one or multiple items from the service list on the left, o
URL in the bottom textbox. Then click the corresponding --> button to add the
to-be-checked list. Click <-- button to move the item(s) back to the backup se
any item listed in the to-be-checked list on the right side, you can either check
selectively choose some of them first, then click Ping! button. If the item liste
begins with "!", that means it is not available. You may continue to check its a
Refresh to expel them out of your resource list. Remove button removes t
Internet

Contacts / Acknowledgements

- krjackson@lbl.gov

- Slides courtesy of the Globus Project. www.globus.org

- This work was supported by the Mathematical, Information, and Computational Science Division subprogram of the Office of Advanced Scientific Computing Research, U.S. Department of Energy, under Contract DE-AC03-76SF00098 with the University of California.