

# An Open Grid Services Architecture

Ian Foster

Mathematics and Computer Science Division

Argonne National Laboratory

and

Department of Computer Science

The University of Chicago

<http://www.mcs.anl.gov/~foster>

# Abstract

In both eBusiness and eScience, we often need to integrate services across distributed, heterogeneous, dynamic “virtual organizations” (VOs) formed from the disparate resources within a single enterprise and/or via external resource sharing and service provider relationships. This integration can be technically challenging due to the need to achieve various qualities of service (QoS) when running on top of different native platforms. We present an *Open Grid Services Architecture* that addresses these challenges. Building on concepts and technologies from the Grid and Web services communities, this architecture defines a uniform exposed service semantics (the *Grid service*); defines standard mechanisms for creating and discovering transient Grid service instances; provides location transparency and multiple binding protocols for service instances; and supports mapping services for integration with underlying native platform facilities. The Open Grid Services Architecture defines, in terms of Web Services Description Language (WSDL) interfaces, mechanisms required for creating and composing sophisticated distributed systems, including lifetime management, reliable remote invocation, change management, credential management, and notification. Our presentation extends our earlier work on Grid architecture by describing how Grid mechanisms can implement a service oriented architecture, explaining how Grid functionality can be incorporated into a Web services framework, and illustrating how our architecture can be applied within commercial computing as a basis for distributed system integration—within and across organizational domains.

**DISTRIBUTE THE WEALTH***Distributed computing initiatives.***GLOBUS PROJECT**

Globus is a research and development project focused on enabling the application of grid computing concepts to scientific and engineering computing. The grid is an emerging infrastructure protocol that enables the integrated use of remote high-end computers, databases, scientific instruments, networks, and other resources.

**Computing power on tap**

Jun 21st 2001

From The Economist print edition

**Economist.com**

**In the first of two articles, we look at the most ambitious attempt yet to combine millions of computers seamlessly around the world—to make processing power available on demand anywhere, rather like electrical power.**

# Grid Computing

**I.B.M. Making a Commitment to Next Phase of the Internet**

By STEVE LOHR

**The New York Times**

**I**.B.M. is announcing today a new initiative to support and exploit a technology known as grid computing, which the company and much of the computer research community say is the next evolutionary step in the development of the Internet.

**Globus Grid Computing—the Next Internet**

by John Roy/Steve Milunovich

The Internet was first a network and is now a communications platform. The next evolutionary step could be to a platform for distributed computing. This ability to manage applications and share data over the network is called “grid computing.”

# Overview

- The universal nature of the “Grid problem”
- A review & assessment of Grid technologies, in particular the Globus Toolkit™
- Open Grid Services Architecture as an evolution & integration of Grid technologies and Web services



# Partial Acknowledgements

- Open Grid Services Architecture design
  - Carl Kesselman, Karl Czajkowski @ USC/ISI
  - Steve Tuecke @ANL
  - Jeff Nick, Steve Graham, Jeff Frey @ IBM
- Grid services collaborators at ANL
  - Kate Keahey, Gregor von Laszewski
  - Thomas Sandholm, Jarek Gawor, John Bresnahan
- Globus Toolkit R&D also involves many fine scientists & engineers at ANL, USC/ISI, and elsewhere (see [www.globus.org](http://www.globus.org))
- Strong links with many EU, UK, US Grid projects
- Support from DOE, NASA, NSF, IBM, Microsoft

# The Grid Vision

“Resource sharing & coordinated problem solving in dynamic, multi-institutional virtual organizations”

- On-demand, ubiquitous access to computing, data, and services
- New capabilities constructed dynamically and transparently from distributed services

*“When the network is as fast as the computer's internal links, the machine disintegrates across the net into a set of special purpose appliances”*

(George Gilder)

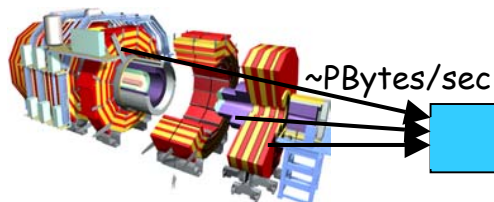


# The Grid Opportunity: eScience and eBusiness

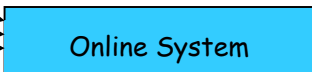
- Physicists worldwide pool resources for peta-op analyses of petabytes of data
- Civil engineers collaborate to design, execute, & analyze shake table experiments
- An insurance company mines data from partner hospitals for fraud detection
- An application service provider offloads excess load to a compute cycle provider
- An enterprise configures internal & external resources to support eBusiness workload



# Grid Communities & Applications: Data Grids for High Energy Physics

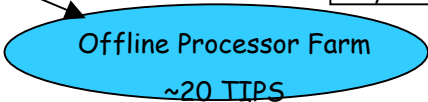


There is a "bunch crossing" every 25 nsecs.  
There are 100 "triggers" per second  
Each triggered event is ~1 MByte in size



~100 MBytes/sec

1 TIPS is approximately 25,000  
SpecInt95 equivalents

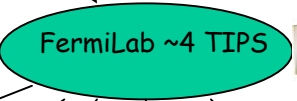


~100 MBytes/sec

**Tier 0**



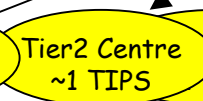
**Tier 1**



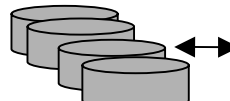
~622 Mbits/sec  
or Air Freight (deprecated)

~622 Mbits/sec

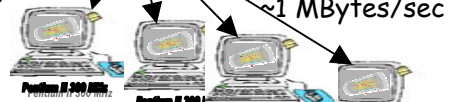
**Tier 2**



~622 Mbits/sec



Physics data cache



Physicist workstations

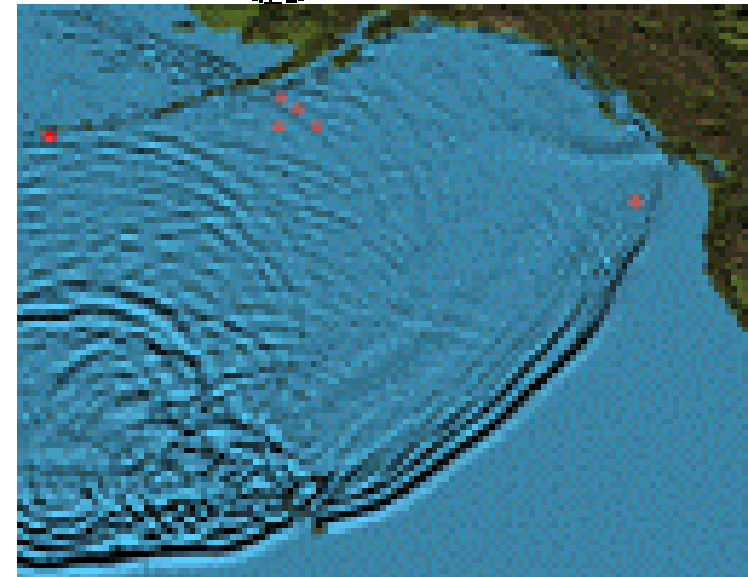
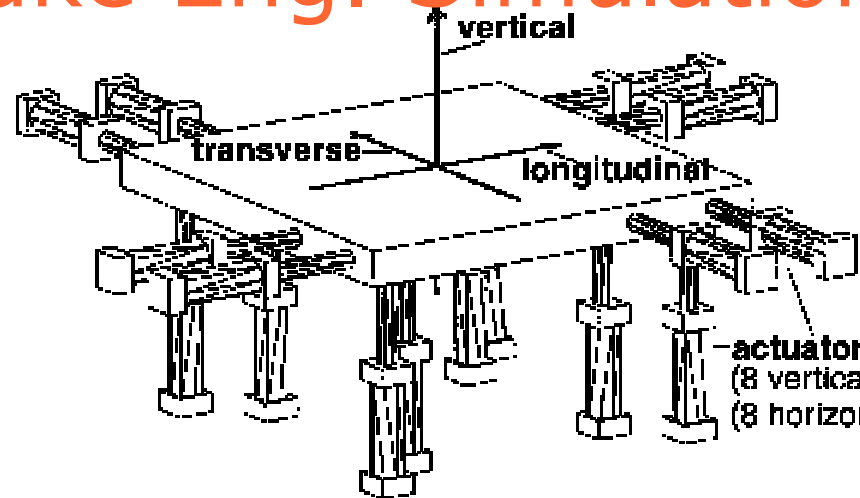
**Tier 4**

~1 MBytes/sec

Physicists work on analysis "channels".  
Each institute will have ~10 physicists working on one or more channels; data for these channels should be cached by the institute server

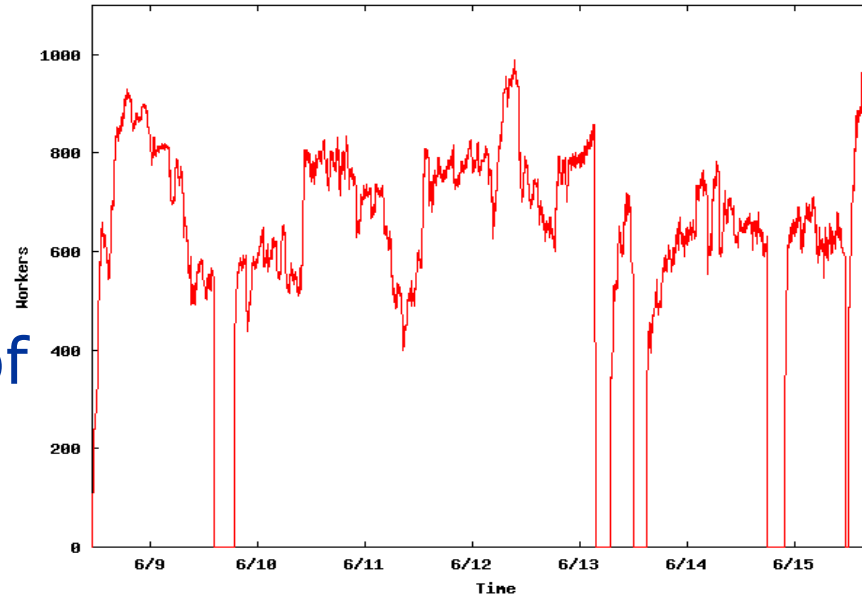
# Grid Communities and Applications: Network for Earthquake Eng. Simulation

- NEESgrid: US national infrastructure to couple earthquake engineers with experimental facilities, databases, computers, & each other
- On-demand access to experiments, data streams, computing, archives, collaboration



# Mathematicians Solve NUG30

- Looking for the solution to the NUG30 quadratic assignment problem
- An informal collaboration of mathematicians and computer scientists
- Condor-G delivered 3.46E8 CPU seconds in 7 days (peak 1009 processors) in U.S. and Italy (8 sites)



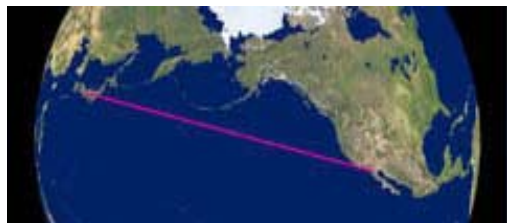
14,5,28,24,1,3,16,15,  
10,9,21,2,4,29,25,22,  
13,26,17,30,6,20,19,  
8,18,7,27,12,11,23



the globus project™  
www.globus.org

# APAN Trans-Pacific Telemicroscopy Collaboration, Osaka-U, UCSD, ISI

(slide courtesy Mark Ellisman@UCSD)



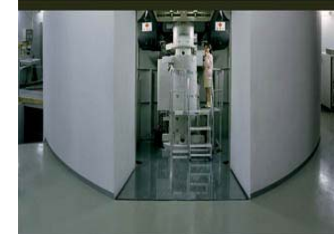
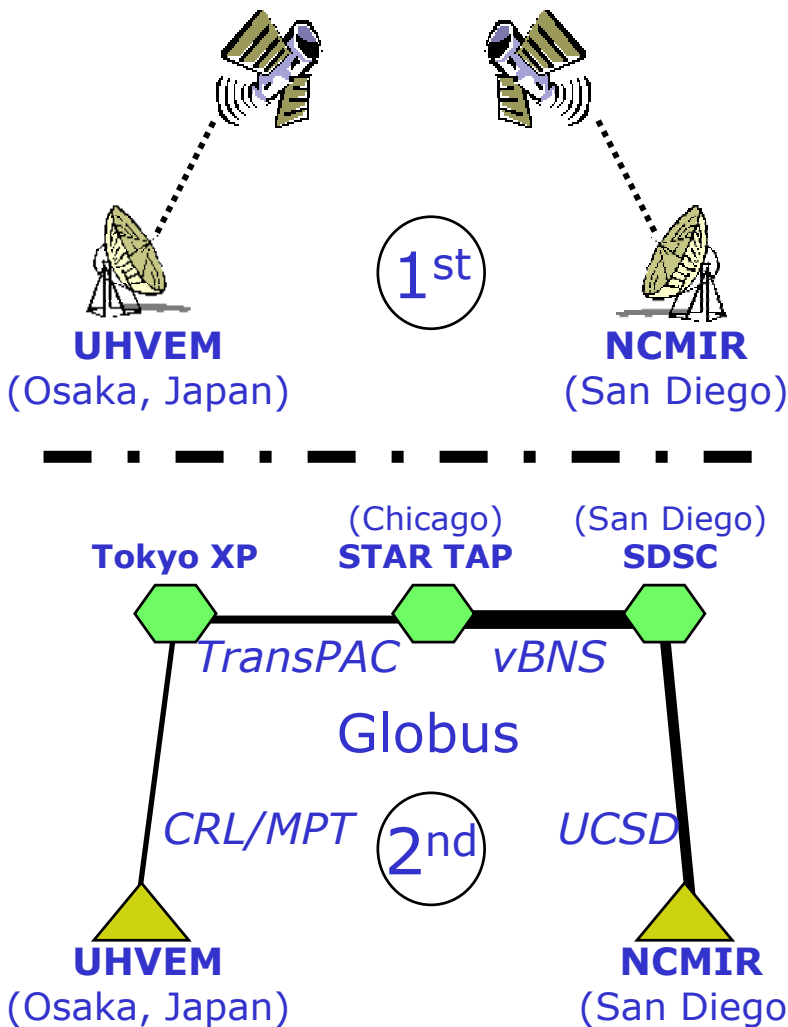
NET NEWS

## Microscopy Across an Ocean

A big push by biologists to use computer networks to operate rare instruments from afar passed a major milestone on 25 June: Scientists took a spin on the world's most powerful electron microscope in Japan—while sitting 6000 kilometers away in California.

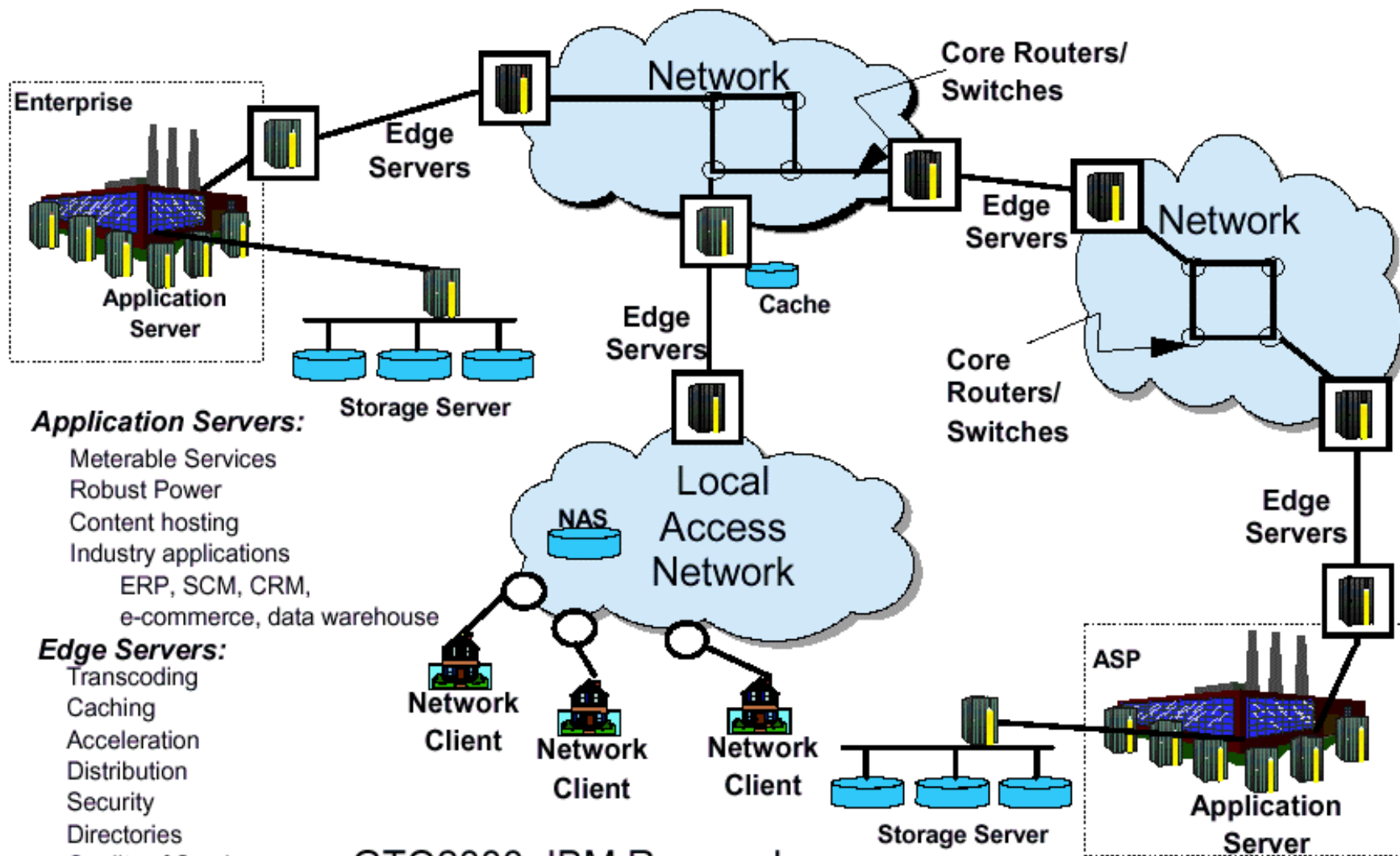
Six years ago, University of California, San Diego, neuroscientist Mark Ellisman thrilled audiences at a conference in Chicago by using the Internet to control an electron microscope in San Diego. Several U.S. agencies jumped in to fund projects for operating microscopes by remote control, and by now at least a dozen groups are doing so in the United States. Ellisman's team has since moved on to the Mount Everest of microscopes: Osaka University's Ultra High Voltage Electron Microscope, a 3,000,000-volt behemoth that can create three-dimensional images from much thicker samples (such as biological cells) than ordinary microscopes can. Ellisman and his U.S. and Japanese colleagues wondered if they could operate this instrument's roomful of controls from across the Pacific Ocean.

They showed they could. Over 5 hours, San Diego scientists imaged nerve cells from a rat and a frog without setting foot in Japan, controlling things like focus and specimen position across a private data line while the images came in across a satellite video link. Ellisman says this lays the groundwork for researchers all over the United States and Japan to borrow each others' specialized microscopes, probably via a high-speed Internet link, "within a year or two."





# What Makes it Hard (A Commercial Perspective)



### Application Servers:

- Meterable Services
- Robust Power
- Content hosting
- Industry applications
- ERP, SCM, CRM,
- e-commerce, data warehouse

### Edge Servers:

- Transcoding
- Caching
- Acceleration
- Distribution
- Security
- Directories
- Quality of Service

GTO2000: IBM Research

## Requirements Include ...

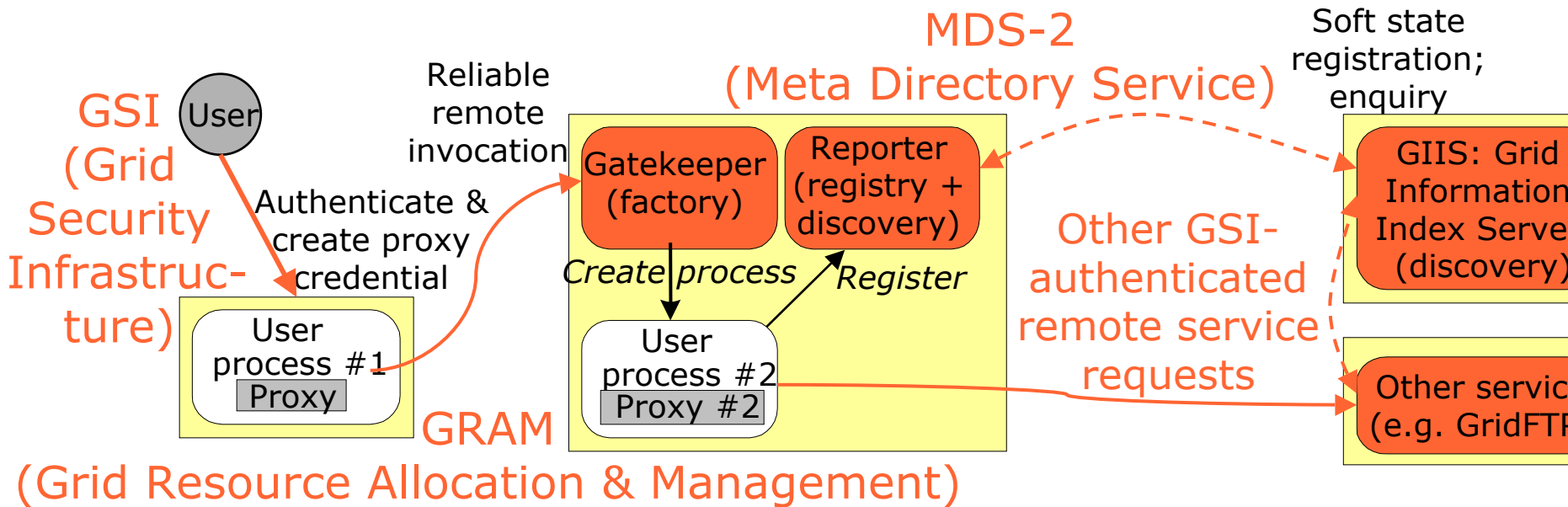
- Dynamic formation and management of virtual organizations
- Online negotiation of access to services: who, what, why, when, how
- Establishment of applications and systems able to deliver multiple qualities of service
- Autonomic management of infrastructure elements
- Open, extensible, evolvable infrastructure

# The Grid World: Current Status

- Dozens of major Grid projects in scientific & technical computing/research & education
- Considerable consensus on key concepts and technologies
  - Open source Globus Toolkit™ a de facto standard for major protocols & services
  - Far from complete or perfect, but out there, evolving rapidly, and large tool/user base
- Industrial interest emerging rapidly
- Opportunity: convergence of eScience and eBusiness requirements & technologies

# The Globus Toolkit in One Slide

- Grid protocols (**GSI**, **GRAM**, ...) enable resource sharing within virtual orgs; toolkit provides reference implementation (■ = Globus Toolkit services)



- Protocols (and APIs) enable other tools and services for membership, discovery, data mgmt, workflow, ...

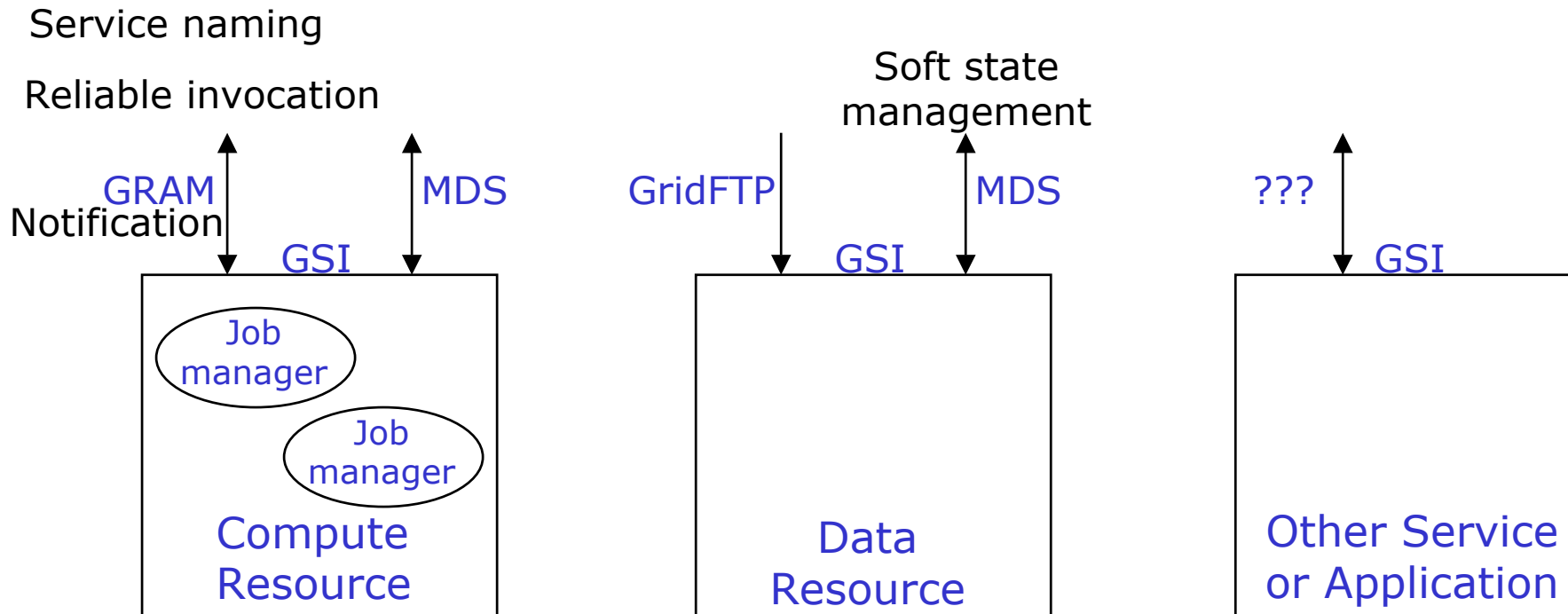
# Globus Toolkit: Evaluation (+)

- Good technical solutions for key problems, e.g.
  - Authentication and authorization
  - Resource discovery and monitoring
  - Reliable remote service invocation
  - High-performance remote data access
- This & good engineering is enabling progress
  - Good quality reference implementation, multi-language support, interfaces to many systems, large user base, industrial support
  - Growing community code base built on tools

# Globus Toolkit: Evaluation (-)

- Protocol deficiencies, e.g.
  - Heterogeneous basis: HTTP, LDAP, FTP
  - No standard means of invocation, notification, error propagation, authorization, termination, ..
- Significant missing functionality, e.g.
  - Databases, sensors, instruments, workflow, ...
  - Virtualization of end systems (hosting envs.)
- Little work on total system properties, e.g.
  - Dependability, end-to-end QoS, ...
  - Reasoning about system properties

# Globus Toolkit Structure



*Lots of good mechanisms, but (with the exception of GSI) not that easily incorporated into other systems*

# Open Grid Services Architecture

- Service orientation to virtualize resources
- Define fundamental Grid service behaviors
  - Core set required, others optional
  - ⇒ A unifying framework for interoperability & establishment of total system properties
- Integration with Web services and hosting environment technologies
  - ⇒ Leverage tremendous commercial base
  - ⇒ Standard IDL accelerates community code
- Delivery via open source Globus Toolkit 3.0
  - ⇒ Leverage GT experience, code, mindshare

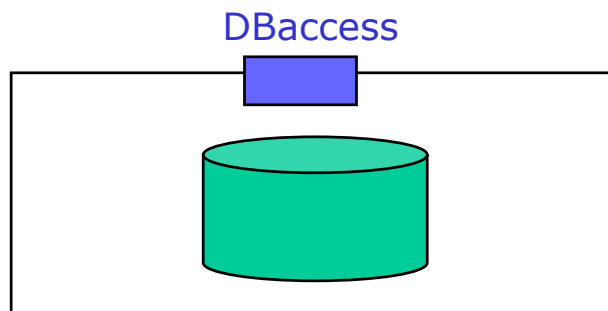
# “Web Services”

- Increasingly popular standards-based framework for accessing network applications
  - W3C standardization; Microsoft, IBM, Sun, others
- **WSDL: Web Services Description Language**
  - Interface Definition Language for Web services
- **SOAP: Simple Object Access Protocol**
  - XML-based RPC protocol; common WSDL target
- **WS-Inspection**
  - Conventions for locating service descriptions
- **UDDI: Universal Desc., Discovery, & Integration**
  - Directory for Web services



# Web Services Example: Database Service

- WSDL definition for “DBaccess” porttype defines operations and bindings, e.g.:
  - Query(QueryLanguage, Query, Result)
  - SOAP protocol



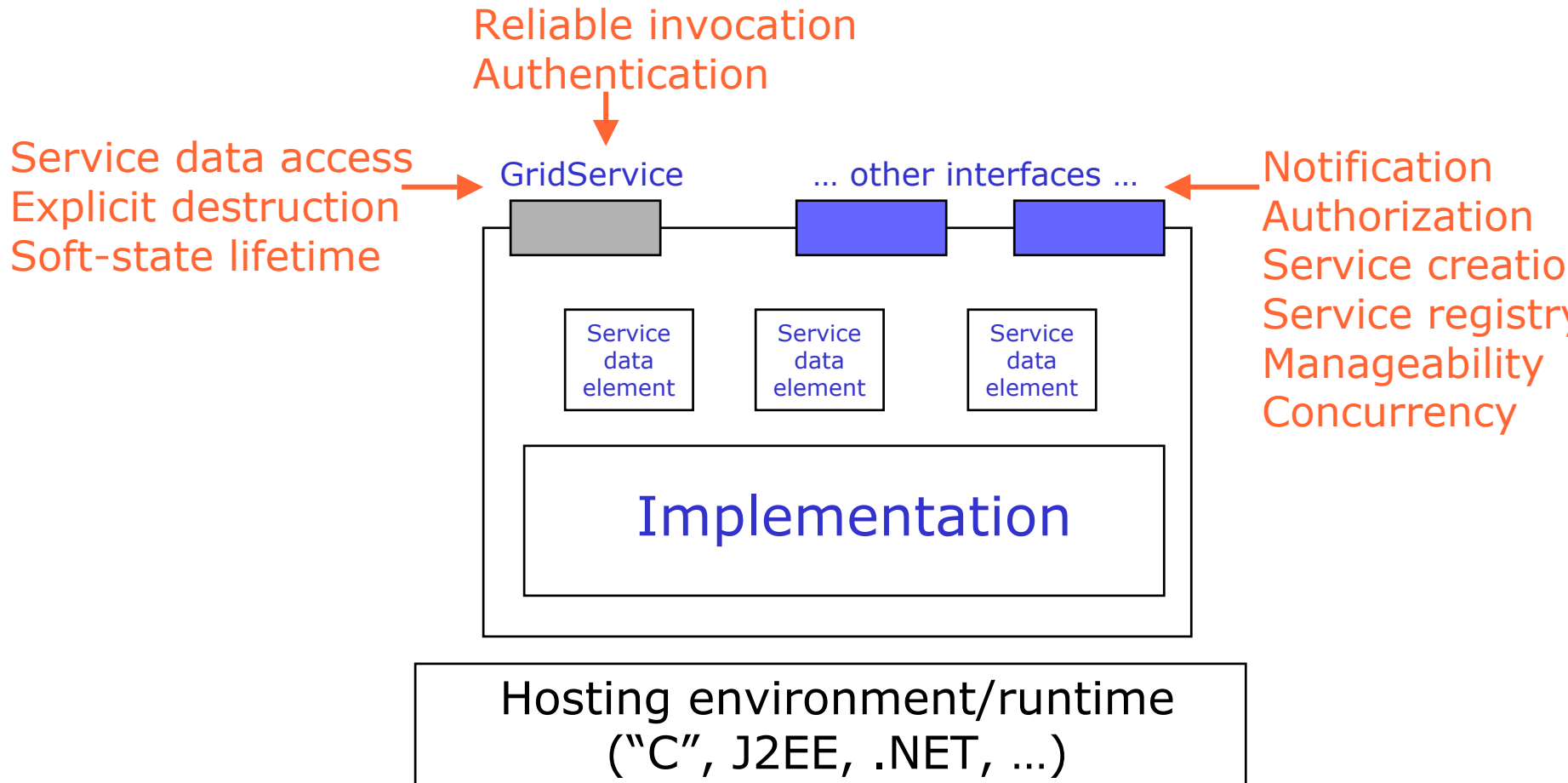
- Client C, Java, Python, etc., APIs can then be generated



# 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
- Significant implications for how services are managed, named, discovered, and used
  - In fact, much of our work is concerned with the management of service instances

# The Grid Service = Interfaces + Service Data



# Open Grid Services Architecture: Fundamental Structure

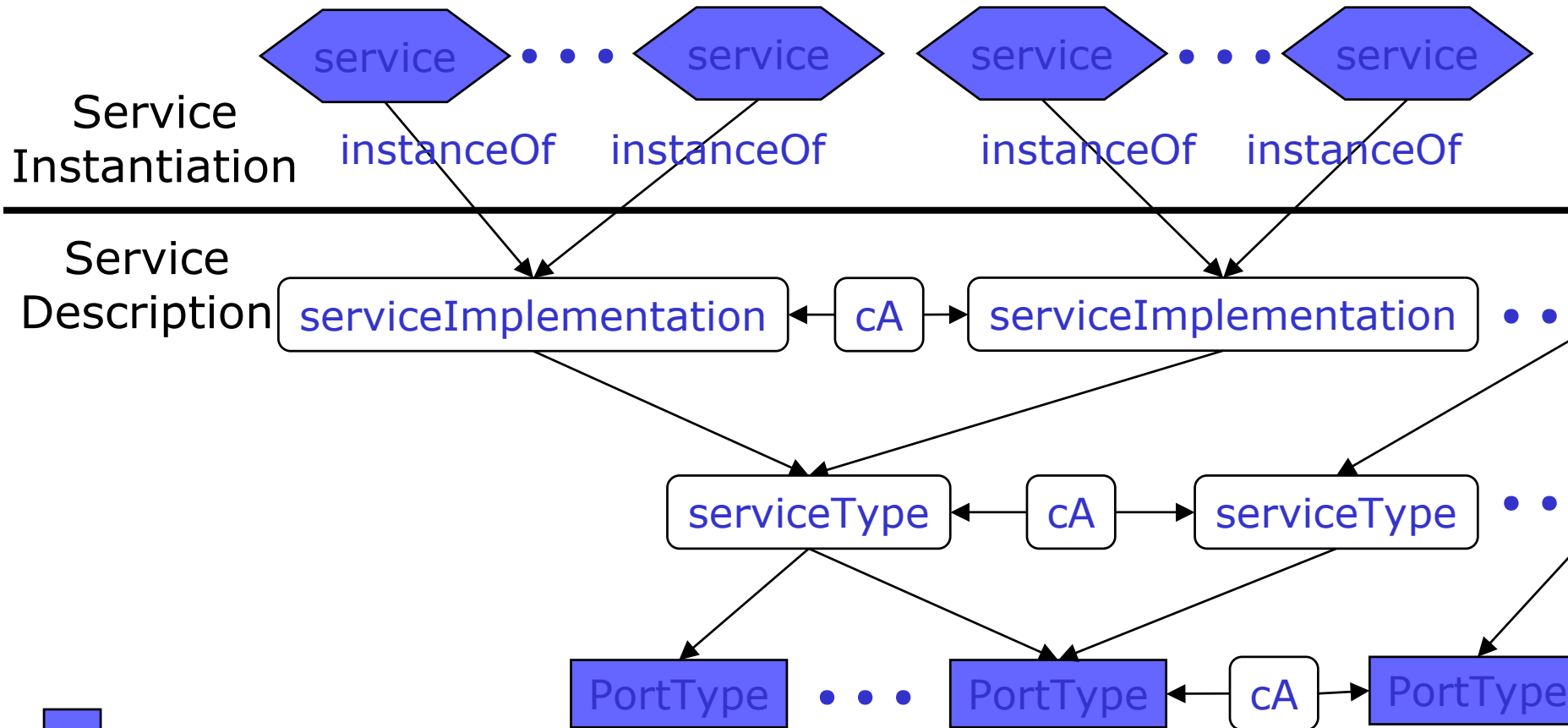
- 1) WSDL conventions and extensions for describing and structuring services
  - Useful independent of “Grid” computing
- 2) Standard WSDL interfaces & behaviors for core service activities
  - portTypes and operations => protocols

# WSDL Conventions & Extensions

- portType (standard WSDL)
  - Define an interface: a set of related operations
- serviceType (extensibility element)
  - List of port types: enables aggregation
- serviceImplementation (extensibility element)
  - Represents actual code
- service (standard WSDL)
  - instanceOf extension: map descr.->instance
- compatibilityAssertion (extensibility element)
  - portType, serviceType, serviceImplementation



# Structure of a Grid Service



 = Standard WSDL

 cA = compatibilityAssertion



# Standard Interfaces & Behaviors: Four Interrelated Concepts

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

# OGSA Interfaces and Operations Defined to Date

- GridService *Required*
  - FindServiceData
  - Destroy
  - SetTerminationTime
- NotificationSource
  - SubscribeToNotificationTopic
  - UnsubscribeToNotificationTopic
- NotificationSink
  - DeliverNotification
- Factory
  - CreateService
- PrimaryKey
  - FindByPrimaryKey
  - DestroyByPrimaryKey
- Registry
  - RegisterService
  - UnregisterService
- HandleMap
  - FindByHandle

*Authentication, reliability are binding properties  
Manageability, concurrency, etc., to be defined*

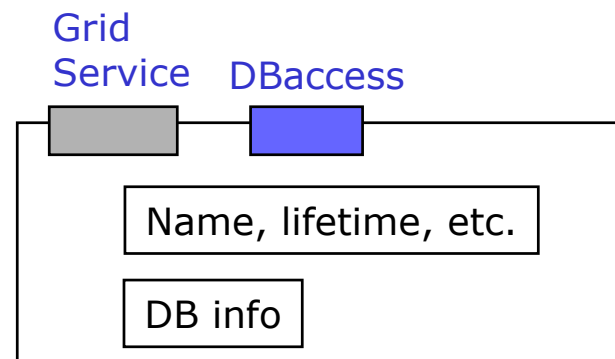
# Service Data

- A Grid service instance maintains a set of service data elements
  - XML fragments encapsulated in standard <name, type, TTL-info> containers
  - Includes basic introspection information, interface-specific data, and application data
- **FindServiceData** operation (GridService interface) queries this information
  - Extensible query language support
- See also notification interfaces
  - Allows notification of service existence and changes in service data



# Grid Service Example: Database Service

- A DBAccess Grid service will support at least two portTypes
  - GridService
  - DBAccess
- Each has service data
  - GridService: basic introspection information, lifetime, ...
  - DBAccess: database type, query languages supported, current load, ..., ...





# Naming and Bindings

- Every service instance has a unique and immutable name: Grid Service Handle (GSH)
  - Basically just a URL
- Handle must be converted to a Grid Service Reference (GSR) to use service
  - Includes binding information; may expire
  - Separation of name from implementation facilitates service evolution
- The HandleMap interface allows a client to map from a GSH to a GSR
  - Each service instance has home HandleMap

# 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

# Lifetime Management

- GS instances created by factory or manually; destroyed explicitly or via soft state
  - Negotiation of initial lifetime with a factory (=service supporting Factory interface)
- **GridService** interface supports
  - **Destroy** operation for explicit destruction
  - **SetTerminationTime** operation for keepalive
- Soft state lifetime management avoids
  - Explicit client teardown of complex state
  - Resource “leaks” in hosting environments

# Factory

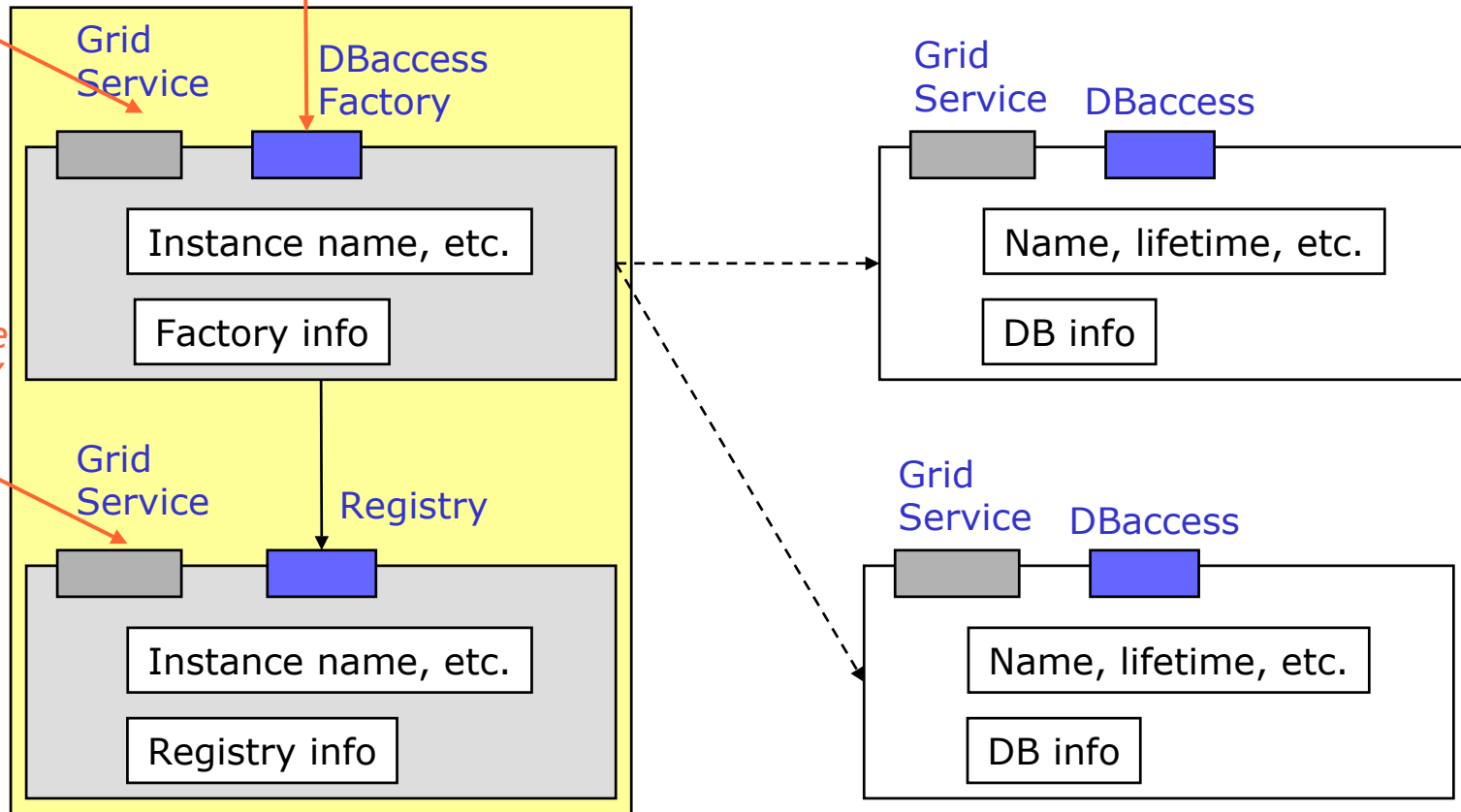
- **Factory** interface's **CreateService** operation creates a new Grid service instance
  - Reliable creation (once-and-only-once)
- **CreateService** operation can be extended to accept service-specific creation parameters
- Returns a **Grid Service Handle (GSH)**
  - A globally unique URL
  - Uniquely identifies the instance for all time
  - Based on name of a home handleMap service

# Transient Database Services

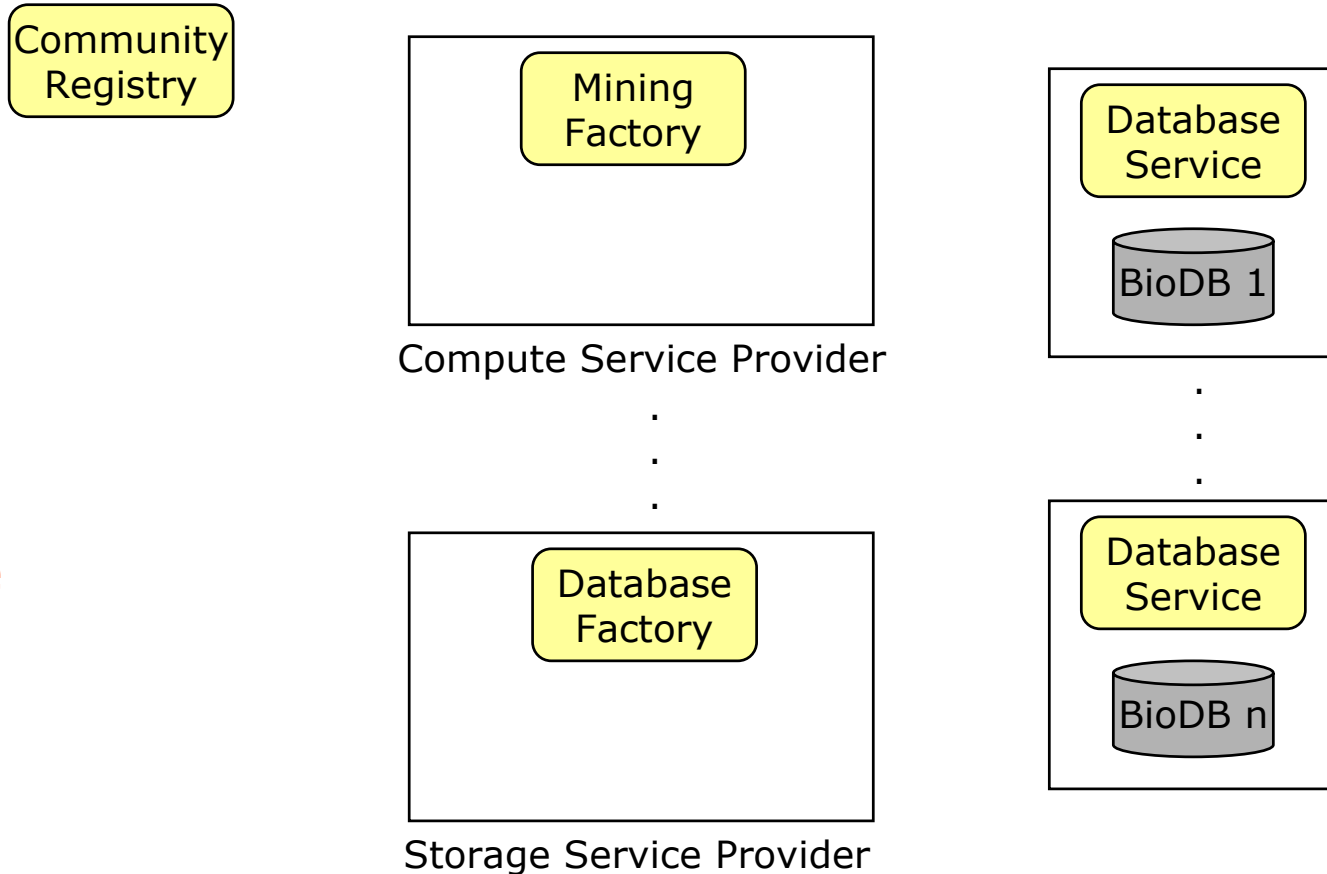
"What services can you create?"

"Create a database service"

"What database services exist?"

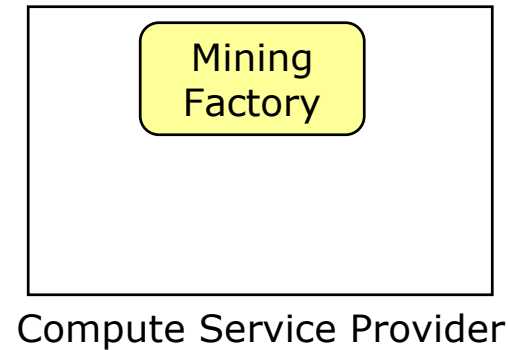
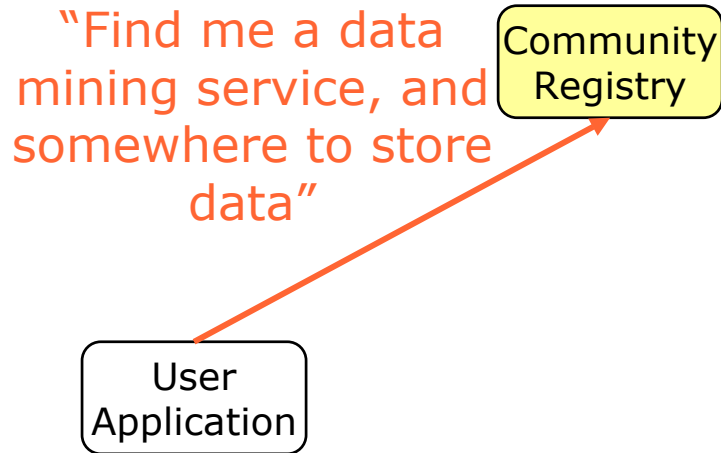


# Example: Data Mining for Bioinformatics

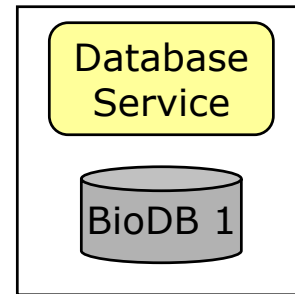
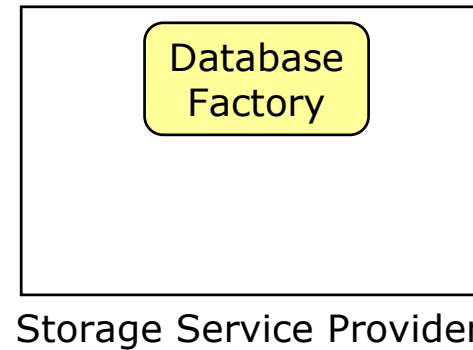


“I want to create  
a personal database  
containing data on  
e.coli metabolism”

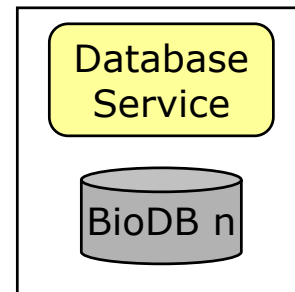
# Example: Data Mining for Bioinformatics



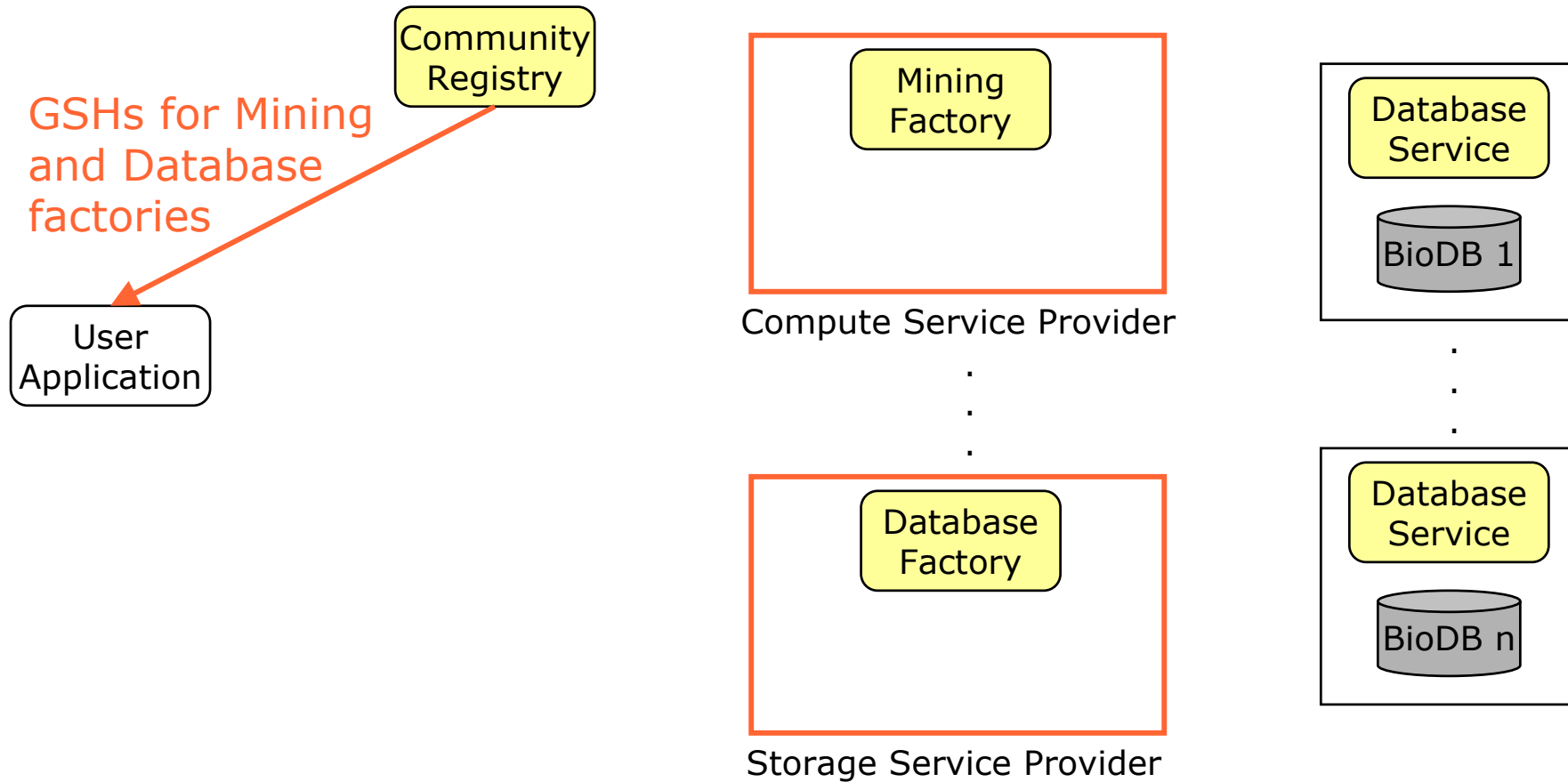
⋮



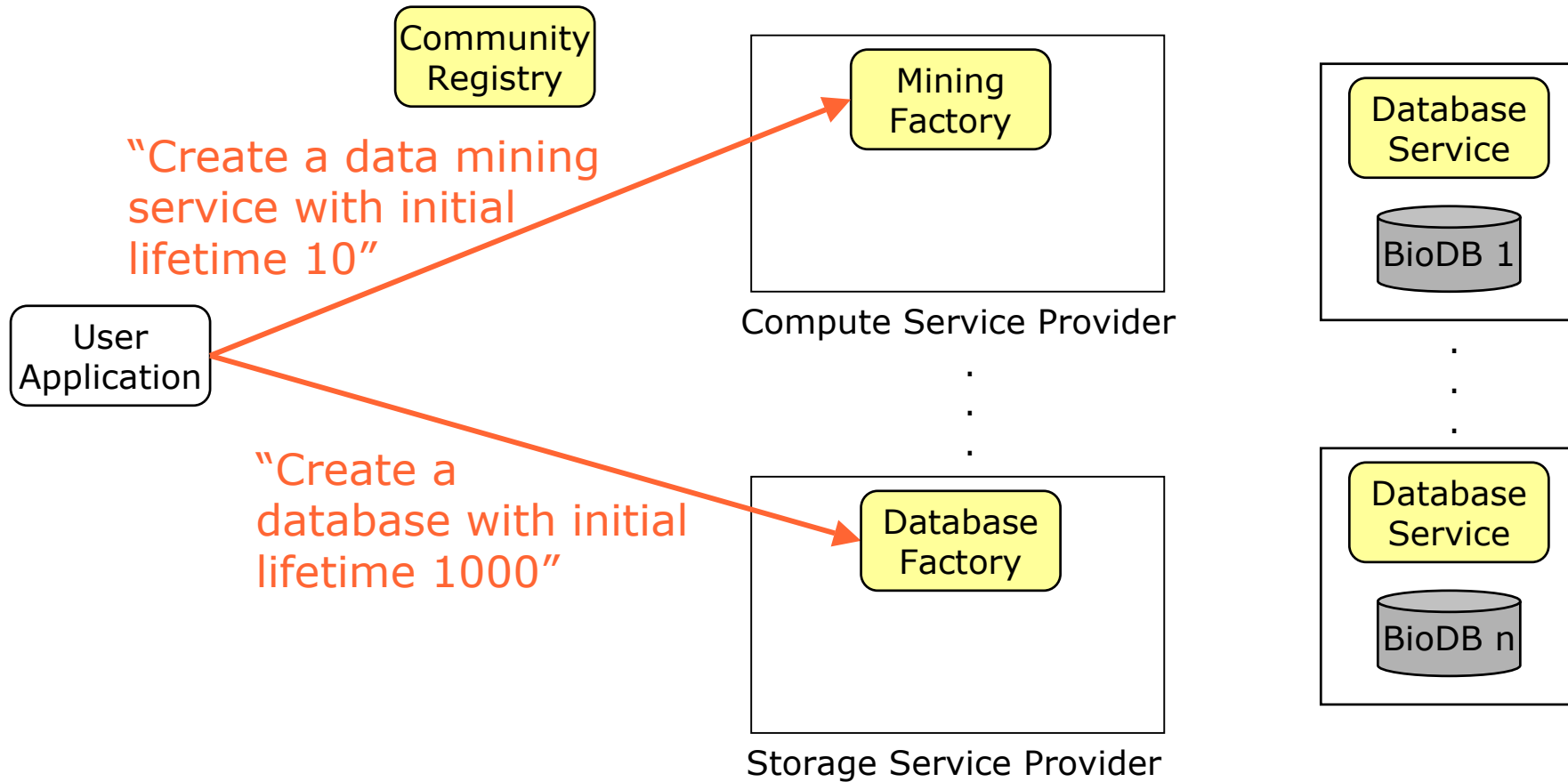
⋮



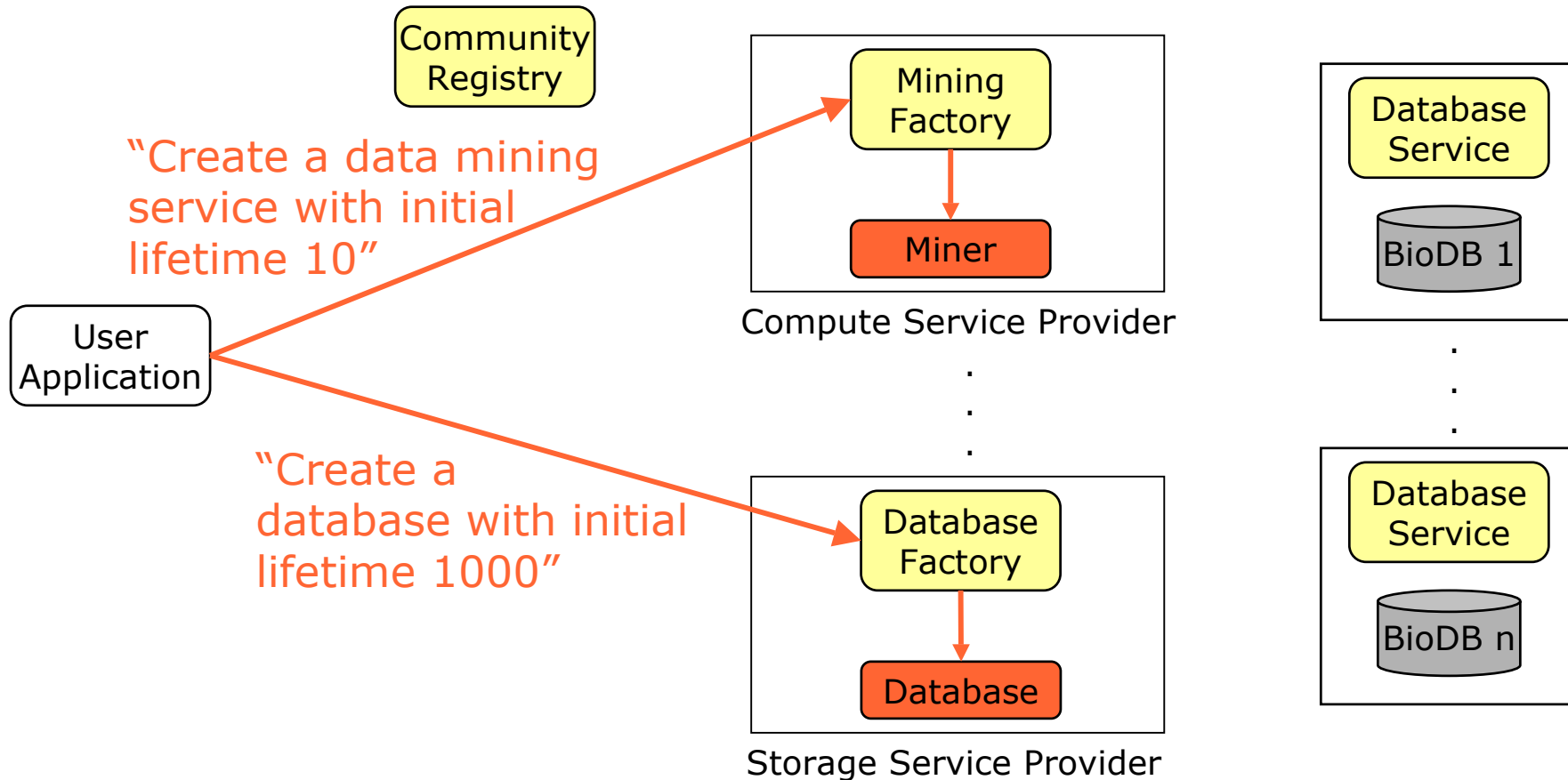
# Example: Data Mining for Bioinformatics



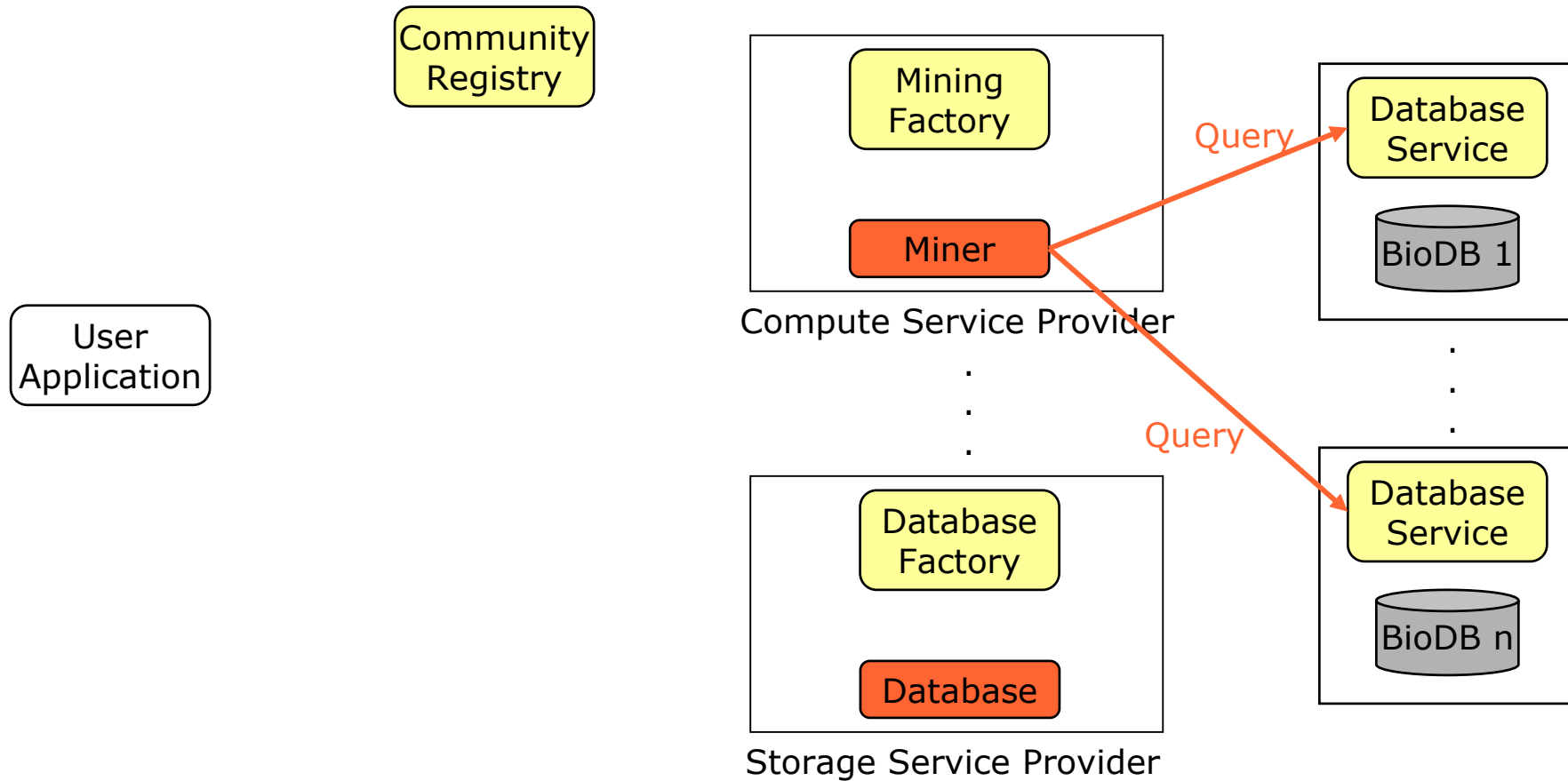
# Example: Data Mining for Bioinformatics



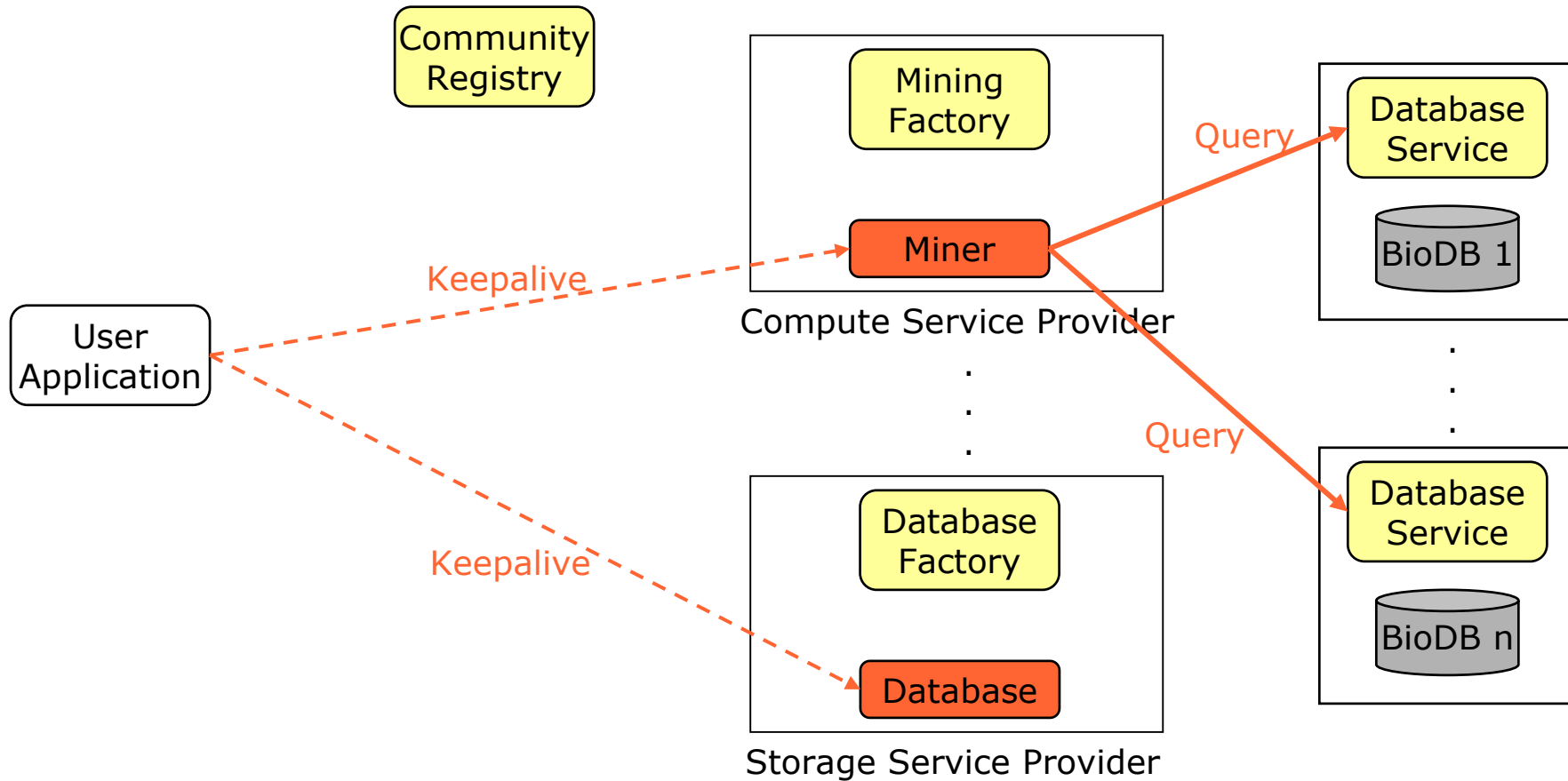
# Example: Data Mining for Bioinformatics



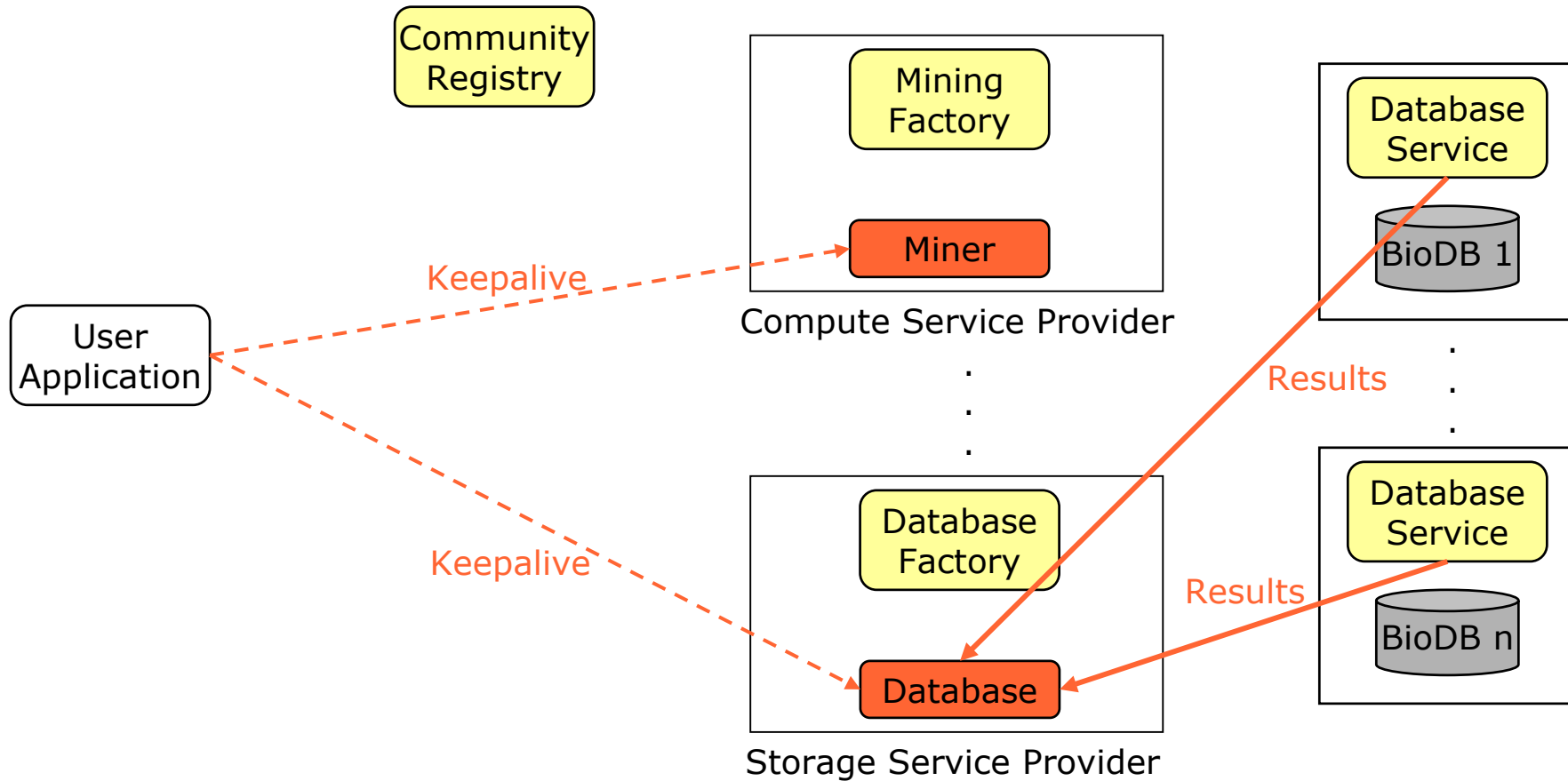
# Example: Data Mining for Bioinformatics



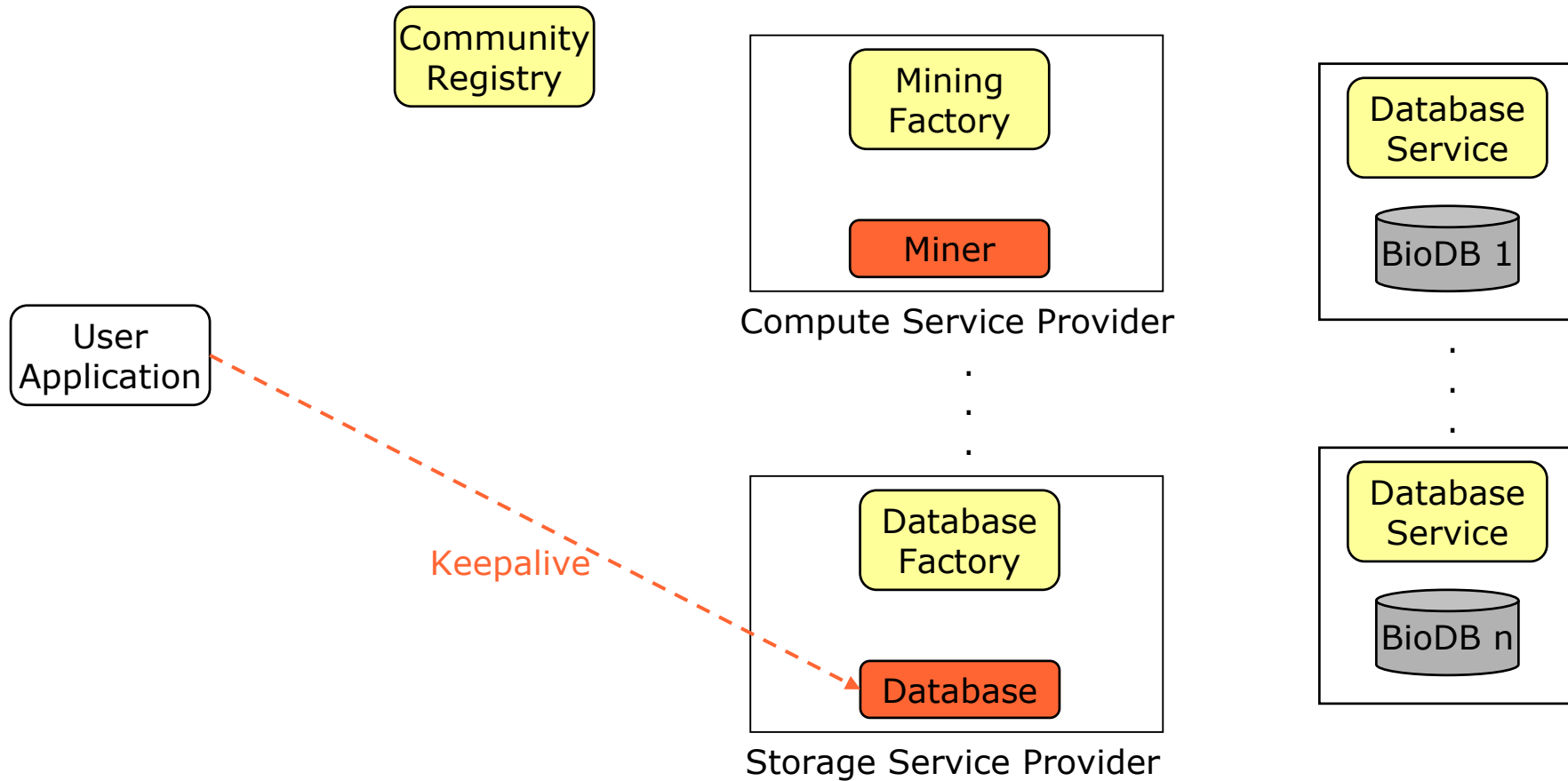
# Example: Data Mining for Bioinformatics



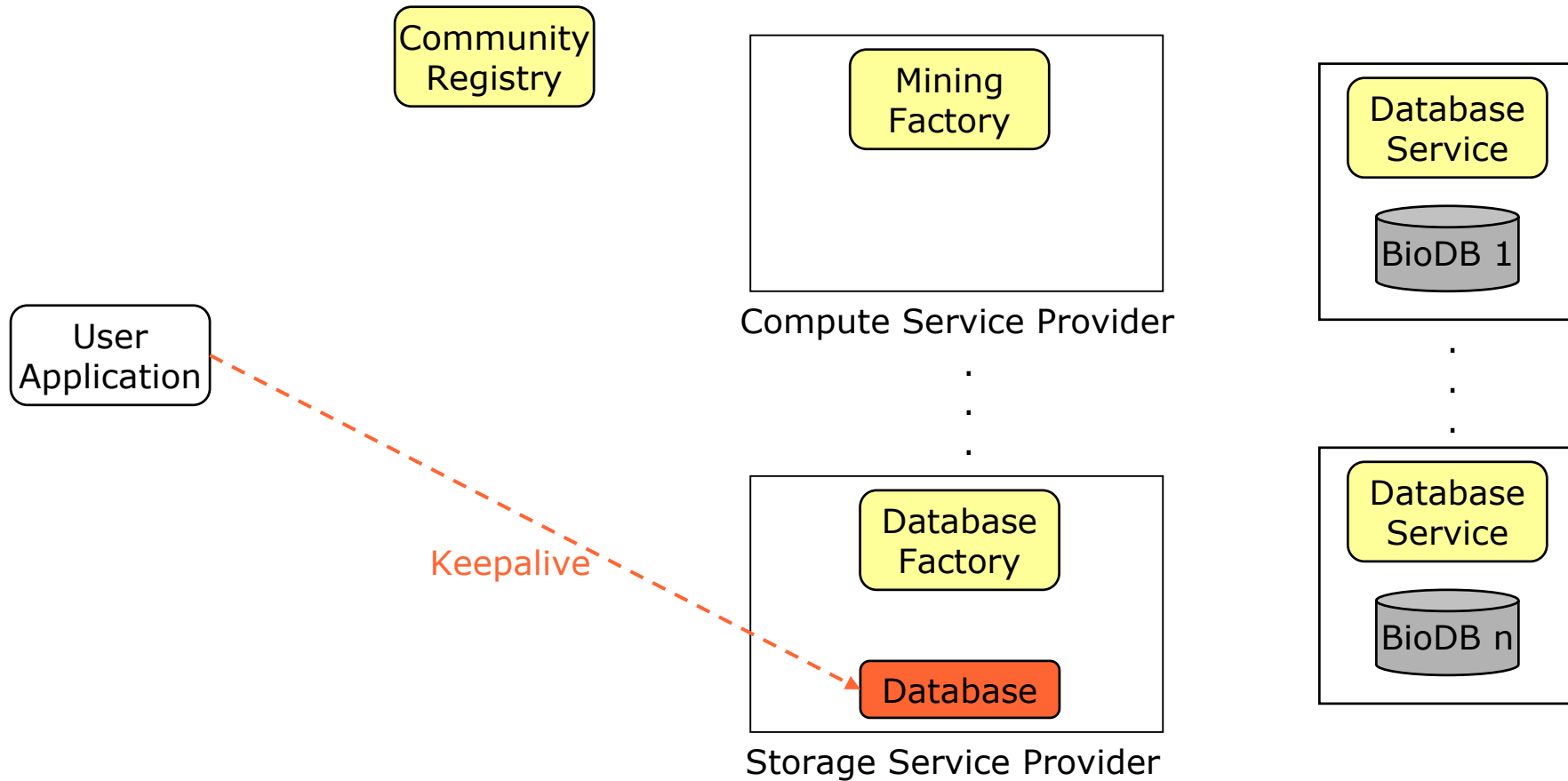
# Example: Data Mining for Bioinformatics



# Example: Data Mining for Bioinformatics



# Example: Data Mining for Bioinformatics

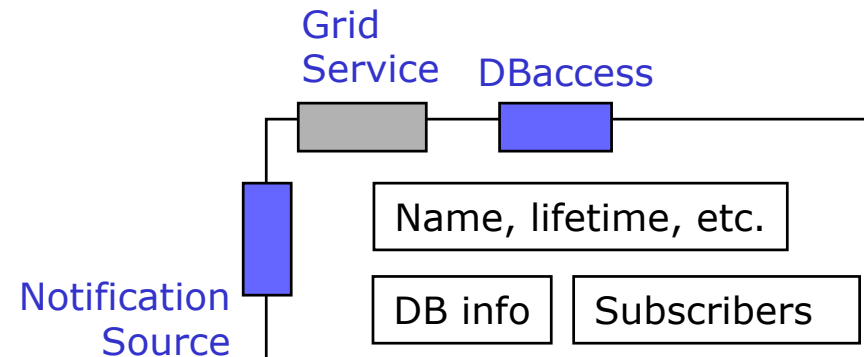
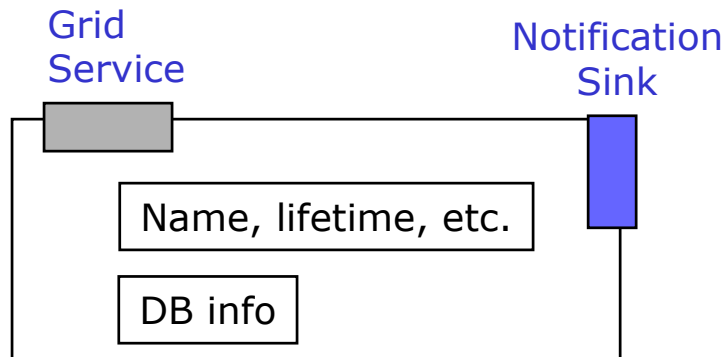


# Notification Interfaces

- **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, 3<sup>rd</sup> 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, ...

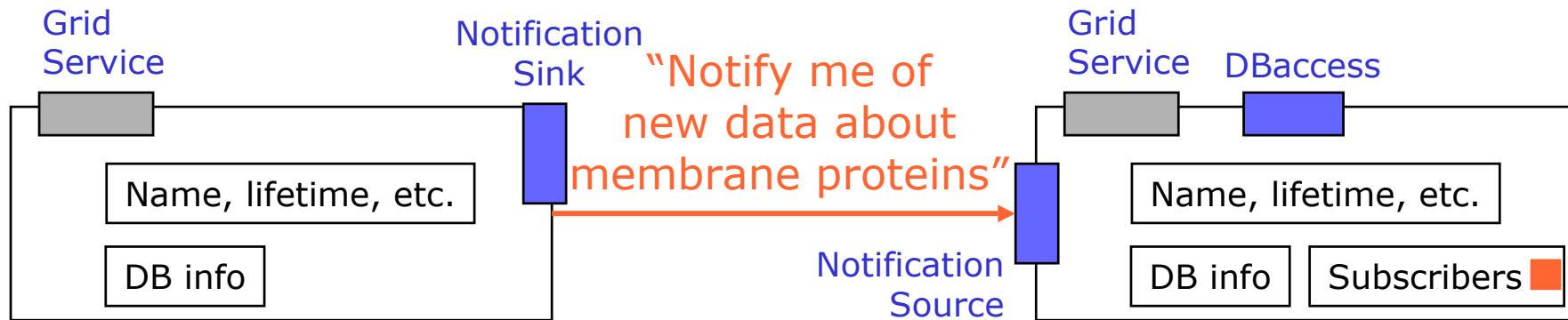
# Notification Example

- Notifications can be associated with any (authorized) service data elements



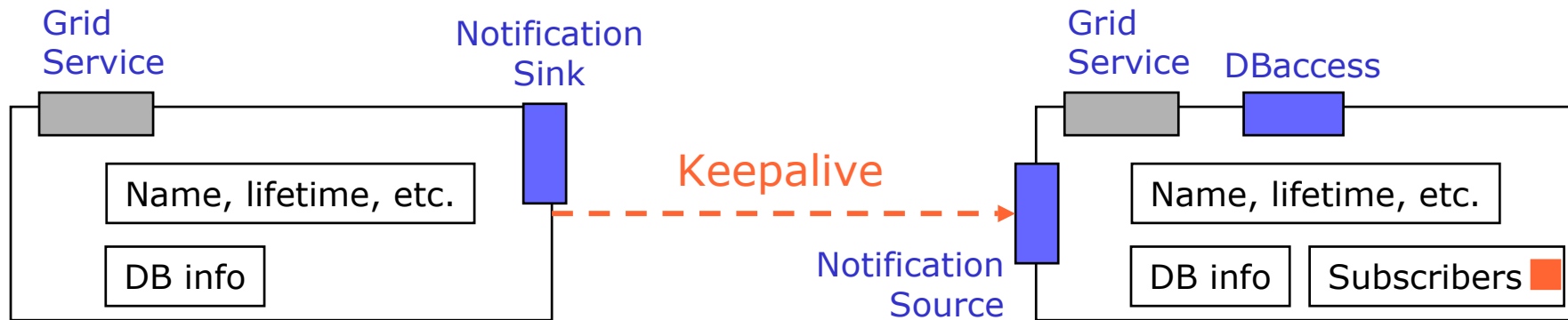
# Notification Example

- Notifications can be associated with any (authorized) service data elements



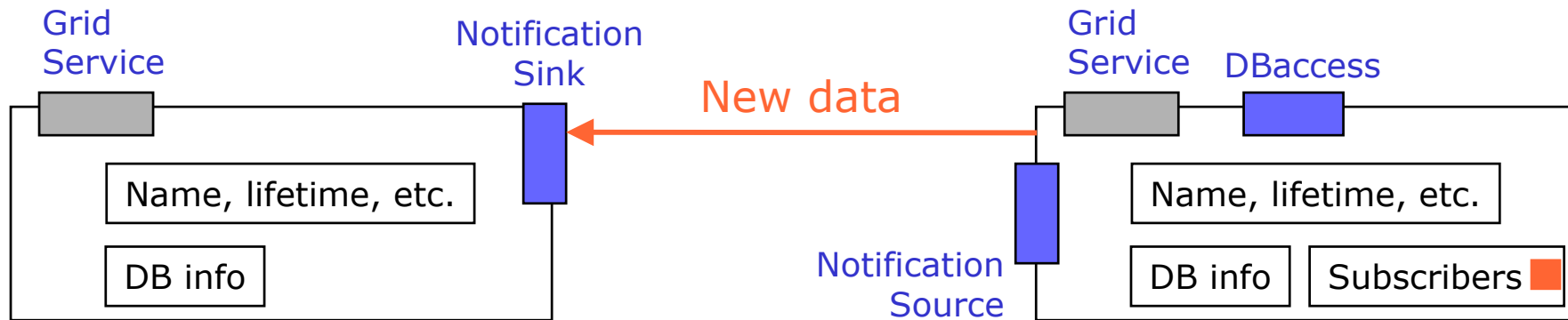
# Notification Example

- Notifications can be associated with any (authorized) service data elements



# Notification Example

- Notifications can be associated with any (authorized) service data elements



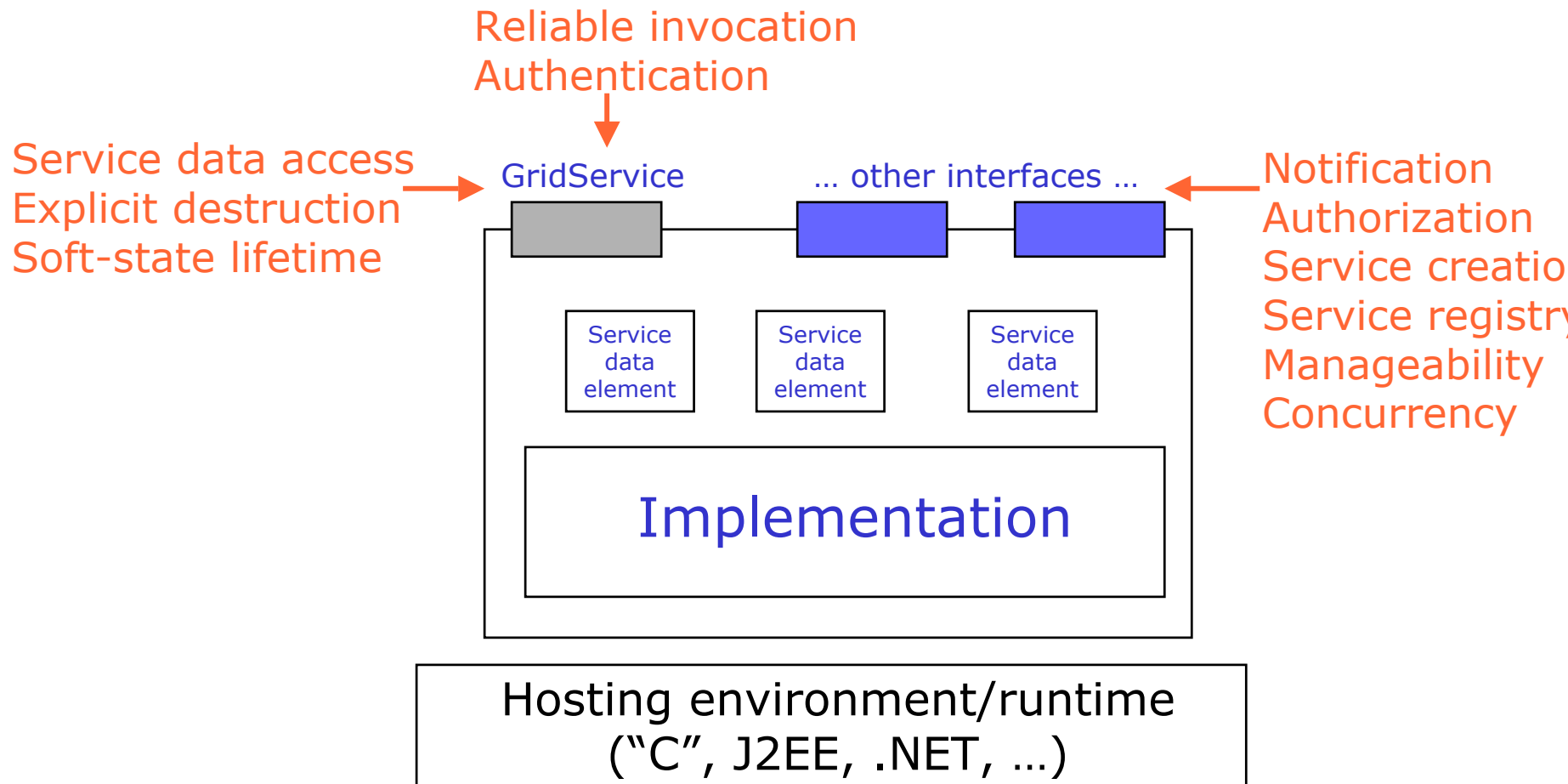


# Open Grid Services Architecture: Summary

- Service orientation to virtualize resources
  - Everything is a service
- From Web services
  - Standard interface definition mechanisms: multiple protocol bindings, local/remote transparency
- From Grids
  - Service semantics, reliability and security models
  - Lifecycle management, discovery, other services
- Multiple “hosting environments”
  - **C**, J2EE, .NET, ...



# Recap: The Grid Service



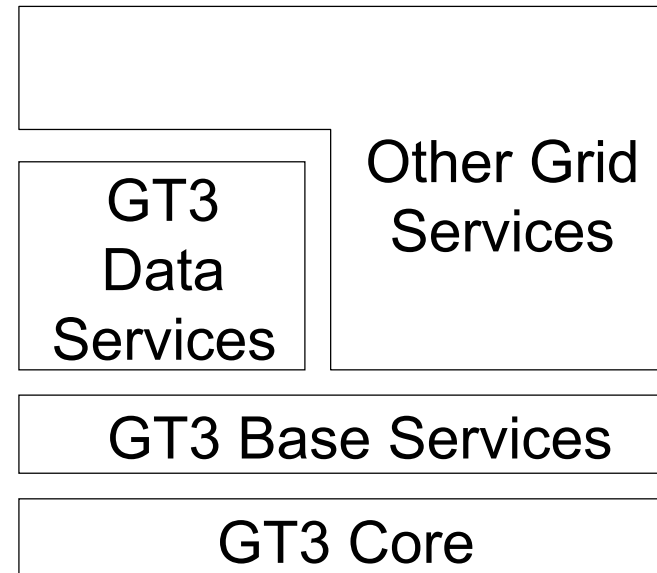
# OGSA and the Globus Toolkit

- Technically, OGSA enables
  - Refactoring of protocols (GRAM, MDS-2, etc.)—*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



# GT3: An Open Source OGSA-Compliant Globus Toolkit

- **GT3 Core**
  - Implements Grid service interfaces & behaviors
  - Reference impln of evolving standard
  - Java first, C soon, C#?
- **GT3 Base Services**
  - Evolution of current Globus Toolkit capabilities
  - Backward compatible
- **Many other Grid services**





# Hmm, Isn't This Just Another Object Model?

- Well, yes, in a sense
  - Strong encapsulation
  - We (can) profit greatly from experiences of previous object-based systems
- But
  - Focus on encapsulation not inheritance
  - Does not require OO implementations
  - Value lies in specific behaviors: lifetime, notification, authorization, ..., ...
  - Document-centric not type-centric

# Grids and OGSA: Research Challenges

- Grids pose profound problems, e.g.
  - Management of virtual organizations
  - Delivery of multiple qualities of service
  - Autonomic management of infrastructure
  - Software and system evolution
- OGSA provides foundation for tackling these problems in a rigorous fashion?
  - Structured establishment/maintenance of global properties
  - Reasoning about total system properties

## Why We Will Succeed

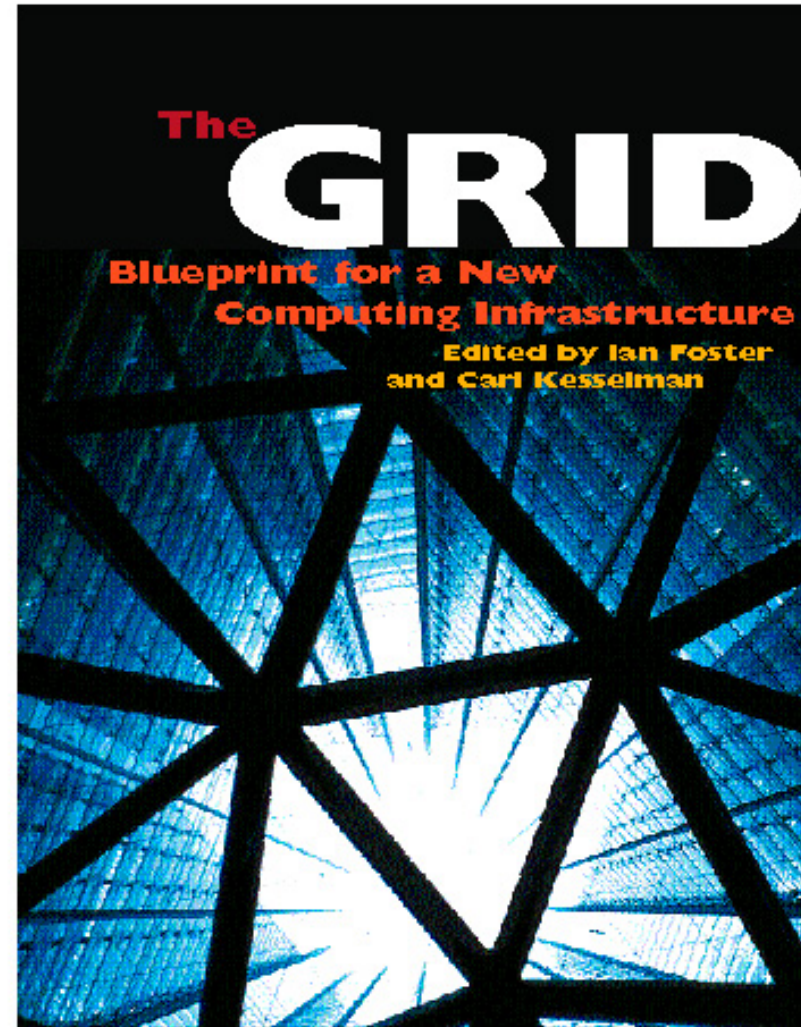
- Strong and compelling vision with broad support across disciplines and borders
- Strong base of technology, experience, success stories, projects
- Genuinely open with respect to intellectual and technology contributions
- Open Grid community process and organization is gaining commercial support
- Strong partnerships, in particular with UK eScience program

# Summary

- The Grid problem: Resource sharing & coordinated problem solving in dynamic, multi-institutional virtual organizations
- Globus Toolkit a source of protocol and API definitions—and reference implementations
  - And many projects applying Grid concepts (& Globus technologies) to important problems
- Open Grid Services Architecture represents (we hope!) next step in evolution
- An enabling framework for investigations of Internet-scale computing systems

# For More Information

- The Globus Project™
  - [www.globus.org](http://www.globus.org)
- Grid architecture
  - [www.globus.org/research/papers/anatomy.pdf](http://www.globus.org/research/papers/anatomy.pdf)
- Open Grid Services Architecture
  - [www.globus.org/ogsa](http://www.globus.org/ogsa)
- Global Grid Forum
  - [www.gridforum.org](http://www.gridforum.org)



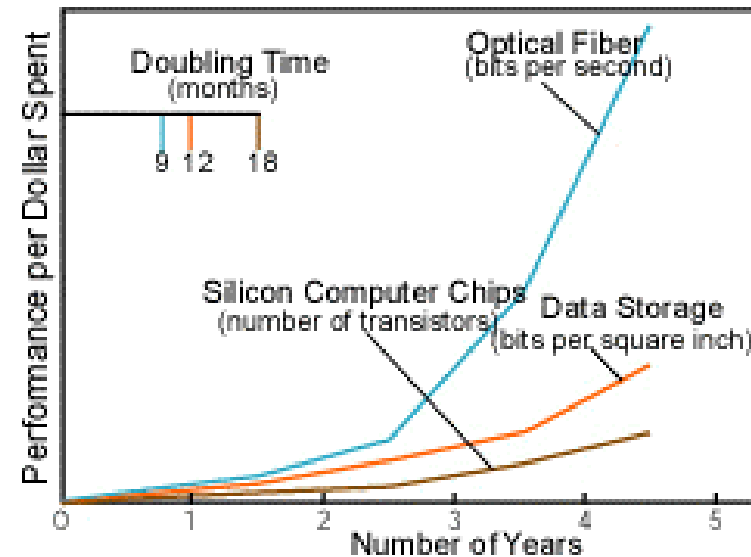
# Extra Slides

## Grids: Why Now?

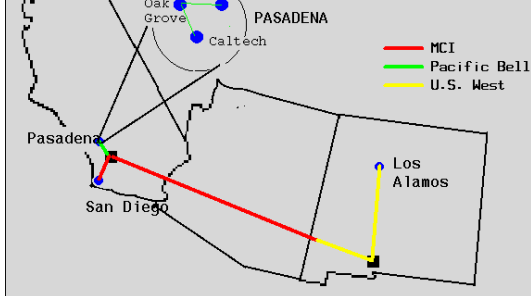
- Moore's law  $\Rightarrow$  highly functional end-systems
- Ubiquitous Internet  $\Rightarrow$  universal connectivity
- Ubiquitous sensors, cheap storage  $\Rightarrow$  DATA!
- Network exponentials produce dramatic changes in geometry and geography
- Web created familiarity with network utilities
- Middleware is no longer vaporware
- Early demonstrators convey benefits
- New business models facilitate outsourcing

# Network Exponentials

- Network vs. computer performance
  - Computer speed doubles every 18 months
  - Network speed doubles every 9 months
  - Difference = order of magnitude per 5 years
- 1986 to 2000
  - Computers: x 500
  - Networks: x 340,000
- 2001 to 2010
  - Computers: x 60
  - Networks: x 4000



# A Brief History



- Early 90s

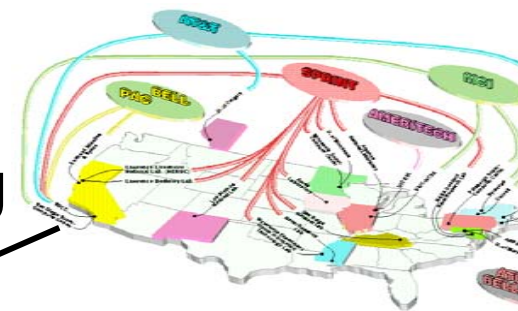
- Gigabit testbeds, metacomputing

- Mid to late 90s

- Early experiments (e.g., I-WAY), academic software projects (e.g., Globus, Legion), application experiments

- 2002

- Major application communities have emerged
- Major infrastructure deployments
- Significant technology base
- Growing industrial interest
- Global Grid Forum: ~500 people, 20+ countries



# Evolution

- This is not happening all at once
  - We have an early prototype of Core (alpha release May)
  - Next we will work on Base, others
  - Full release by end of 2002 (estimate)
  - Partnerships in place for other services
- Backward compatibility
  - API level seems straightforward
  - Protocol level: gateways?
  - We need input on best strategies