

OGSA Architecture

Adapted from

Ian Foster

Mathematics and Computer Science Division

Argonne National Laboratory

and

Department of Computer Science

The University of Chicago



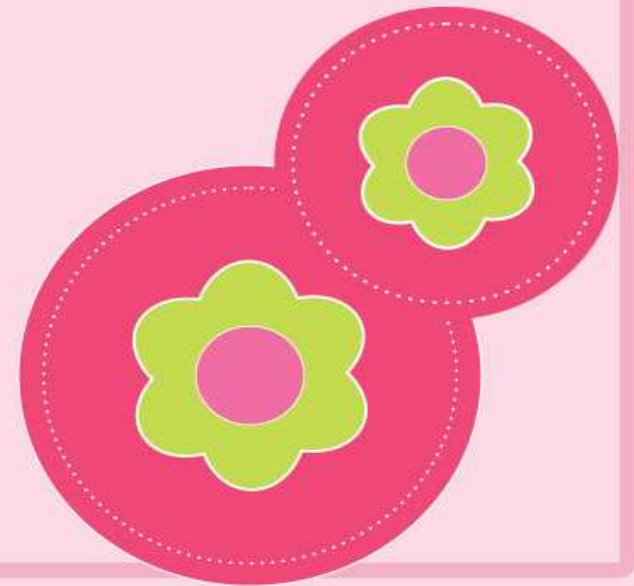
UNIT II GRID SERVICES

Introduction to Open Grid Services
Architecture (OGSA) – Motivation –
Functionality Requirements – Practical &
Detailed view of OGSA/OGSI – Data
intensive grid service models – OGSA
services.



Overview

- Grid Computing
- Web Service
- Grid Service
- OGSA Architecture
- Example Grid Service



Introduction to Grid Architecture

The nature of grid architecture

- A grid architecture identifies fundamental system components, specifies the purpose and function of these components, and indicate how these components interact.



Introduction to Grid Architecture

The Nature of Grid Architecture

- Grid's protocols allow **VO** users and resources to negotiate, establish, manage and exploit sharing relationships.
 - **Interoperability** a fundamental concern
 - The protocols are critical to interoperability
 - **Services** are important
 - We need to consider APIs and SDKs

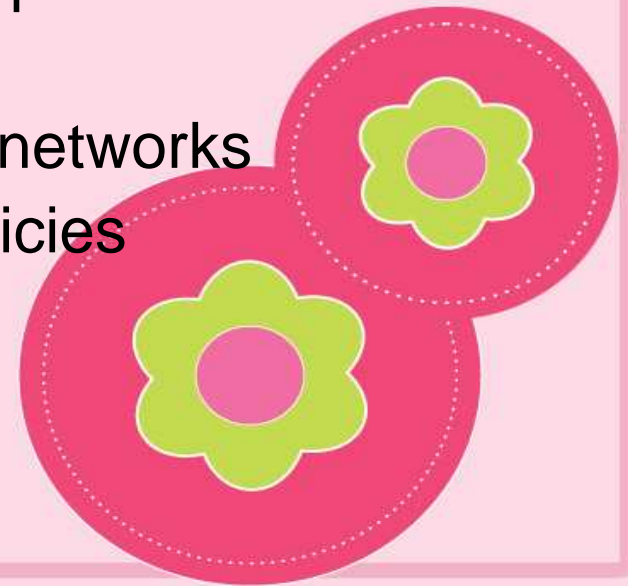
VO: Virtual Organization



Introduction to Grid Architecture

Grid architecture requirements

- The components are
 - numerous
 - owned and managed by different, potentially mutually distrustful organisations and individuals
 - may be potentially faulty
 - have different security requirements and policies
 - heterogeneous
 - connected by heterogeneous, multilevel networks
 - have different resource management policies
 - are likely to be geographically separated



Services in the Web and the Grid

Web services

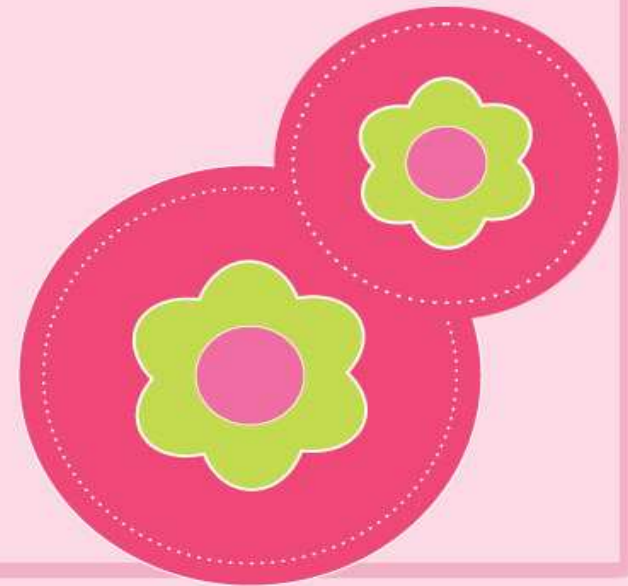
- Define a technique for describing software components to be accessed, methods for accessing these components, and discovery methods that enable the identification of relevant service providers
- A distributed computing technology (like CORBA, RMI...)
- They allow us to create loosely coupled client/server applications.



Services in the Web and the Grid

Web Services:Advantages

- Platform and language independent since they use XML language.
- Most use HTTP for transmitting messages (such as the service request and response)



Services in the Web and the Grid

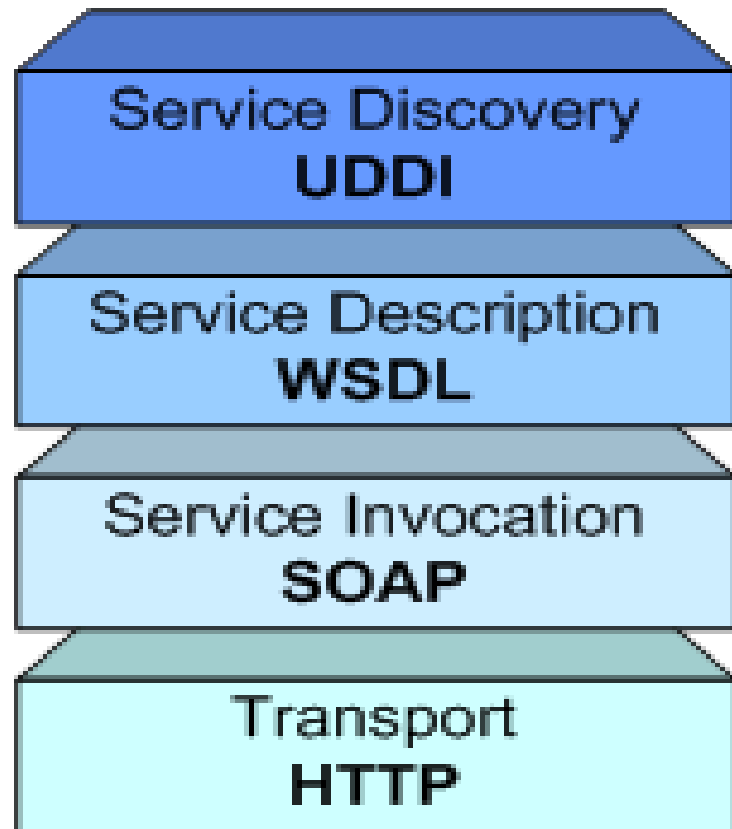
Web Services: Disadvantages

- Overhead : Transmitting data in XML is not as convenient as binary codes.
- Lack of versatility: They allow very basic forms of service invocation (Grid services make up this versatility).
 - Stateless: *They can't remember what you have done from one invocation to another*
 - Non-transient: *They outlive all their clients.*



Services in the Web and the Grid

Web Services Architecture



Find Web services which meet certain requirements
(Universal Description, Discovery and Integration)

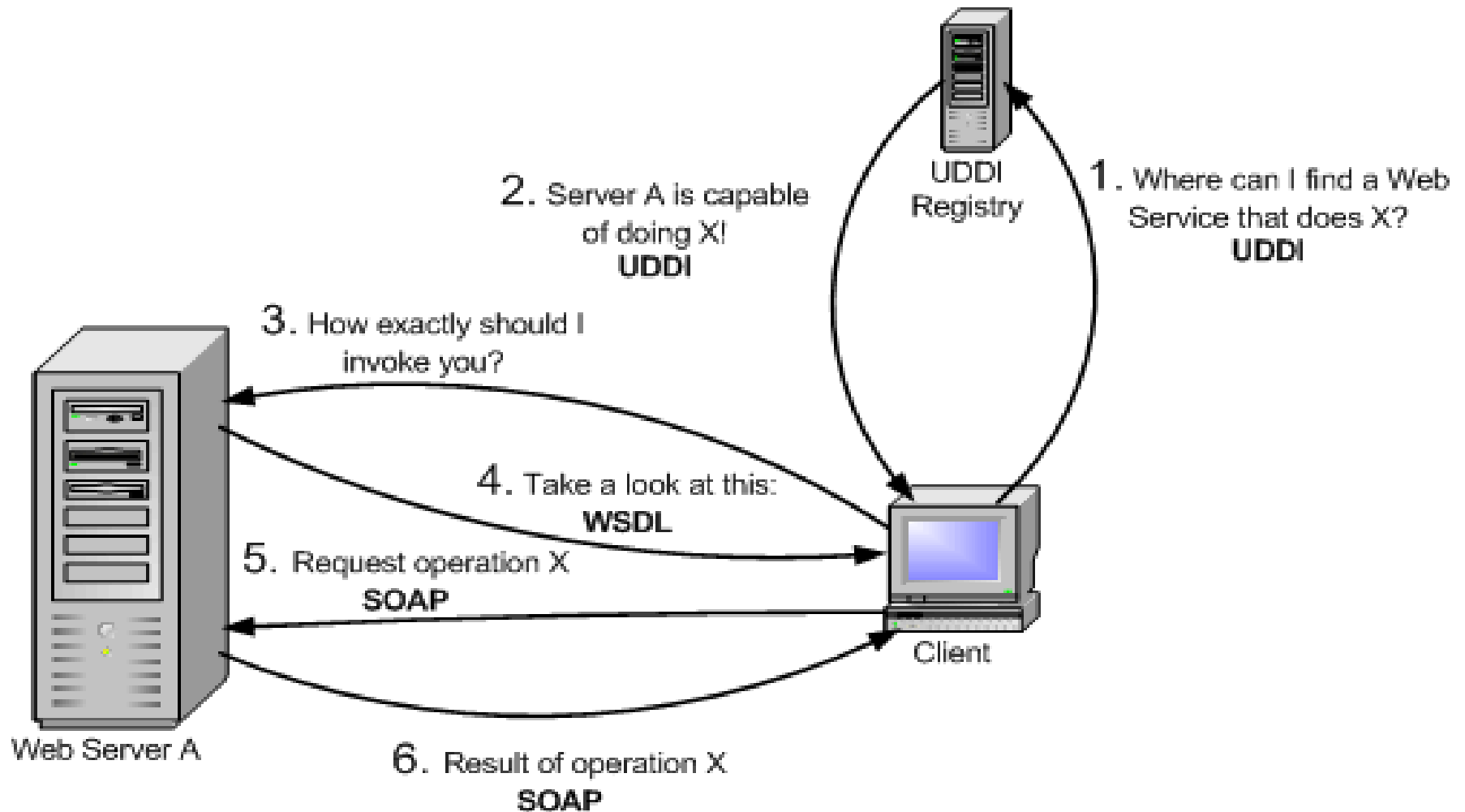
Services describe their own properties and methods
(Web Services Description Language)

Format of requests(client) and responses (server)
(Simple Object Access Protocol)

Message transfer protocol
(Hypertext Transfer Protocol)

Services in the Web and the Grid

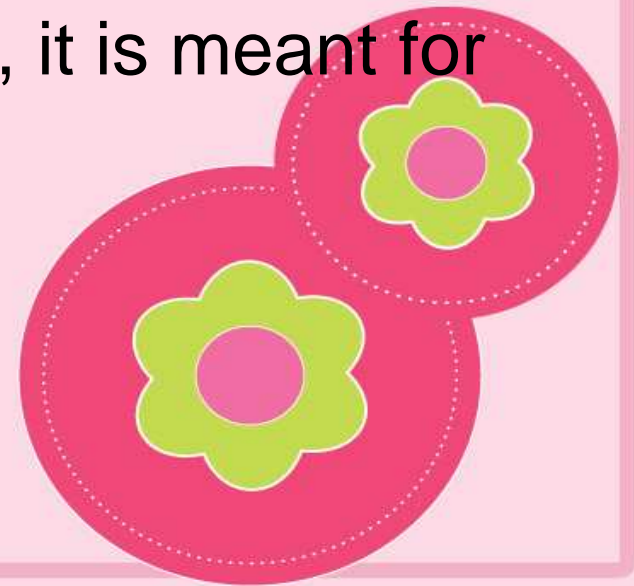
Invoking a Typical Web Service



Services in the Web and the Grid

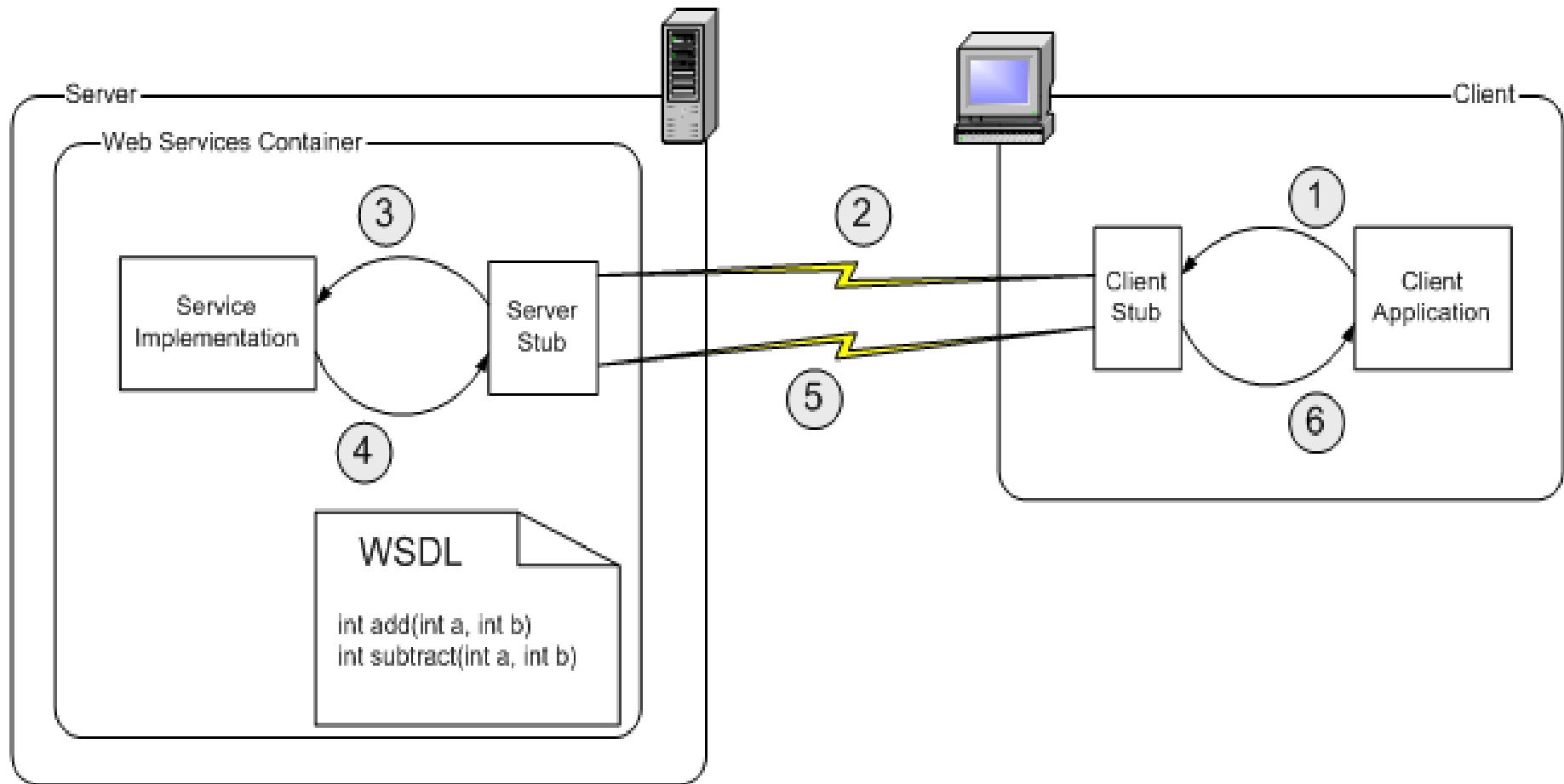
Web Service Addressing

- URI: Uniform Resource Identifiers
- URI and URL are practically the same thing.
 - Example:
`http://webservices.mysite.com/weather/us/WeatherService`
- It can not be used with web browsers, it is meant for softwares.



Services in the Web and the Grid

Web Service Application



Services in the Web and the Grid

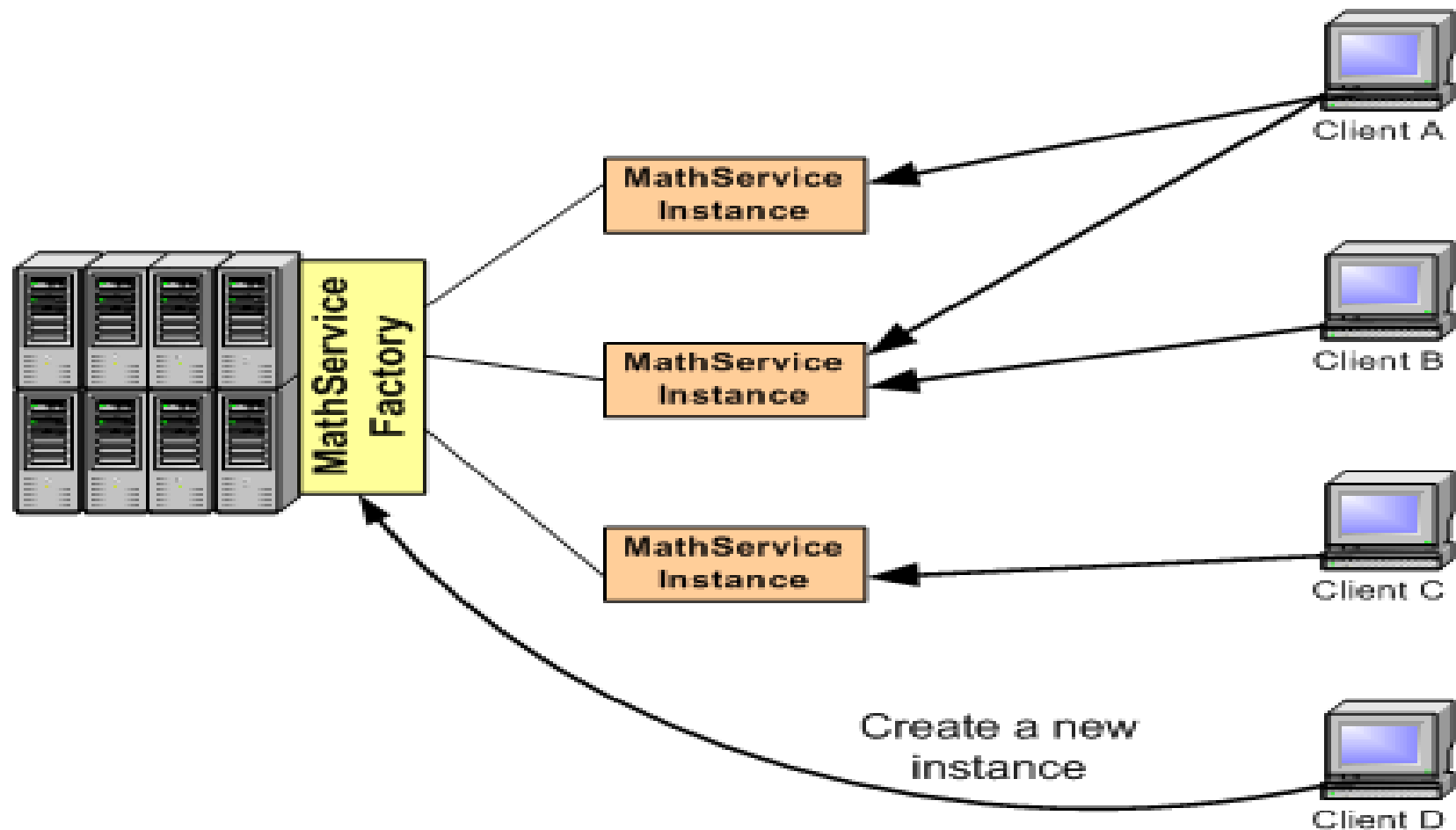
What is a Grid Service?

- It provides a set of well defined interfaces and that follows specific conventions.
- It is a web service with improved characteristics and services.
 - Improvement:
 - Potentially Transient
 - Stateful
 - Delegation
 - Lifecycle management
 - Service Data
 - Notifications
- *Examples* : computational resources, programs, databases...



Services in the Web and the Grid

Factories



Services in the Web and the Grid

GSH & GSR

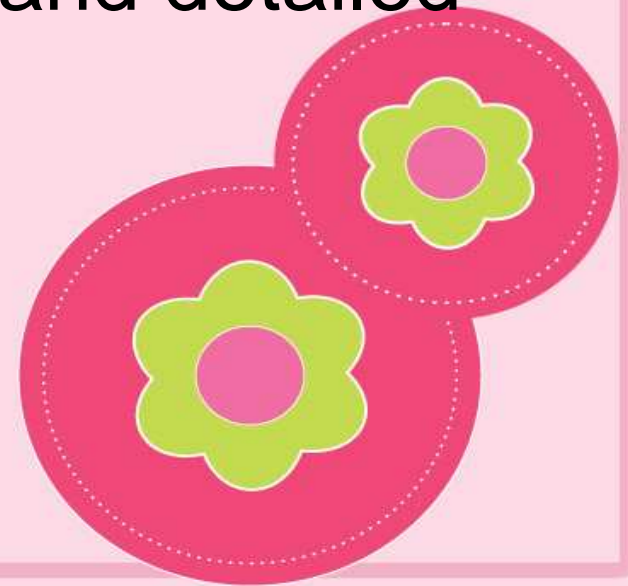
- GSH: Grid Service Handle (URI)
 - Unique
 - Shows the location of the service
- GSR: Grid Service Reference
 - Describes how to communicate with the service
 - As WS use SOAP, our GSR is a WSDL file.



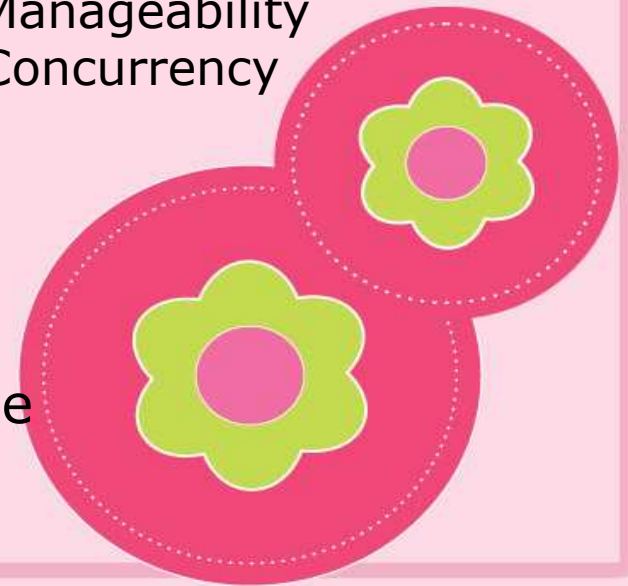
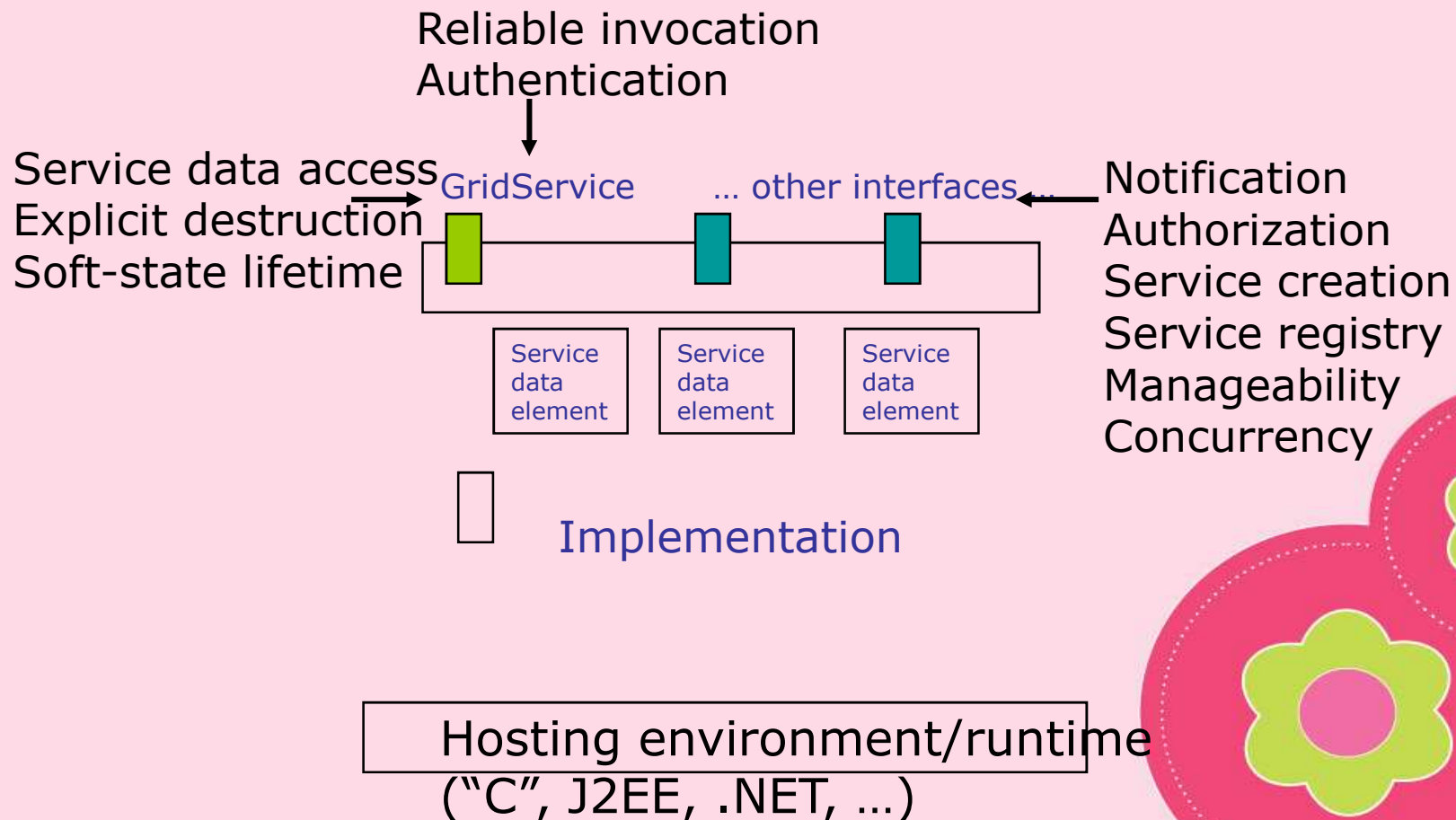
Services in the Web and the Grid

Open Grid Services Architecture (OGSA)

- OGSA defines **what Grid services** are, what they should be **capable** of, what **type** of **technologies** they should be based on.
- OGSA **does not** give a technical and detailed specification. It uses WSDL.

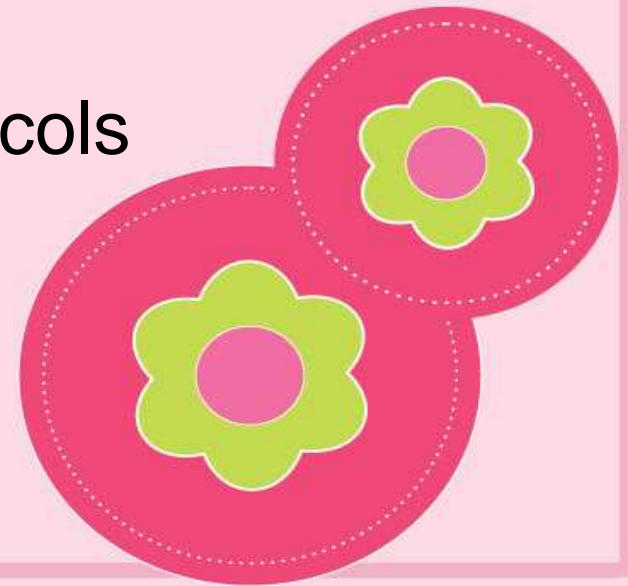


The Grid Service = Interfaces + Service Data



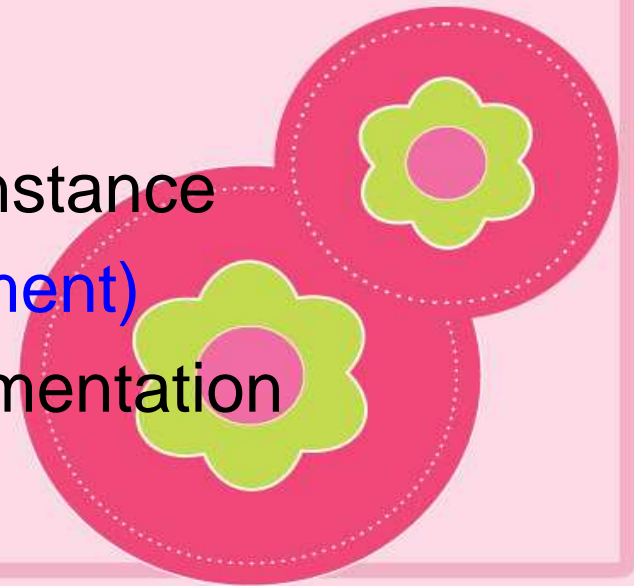
Open Grid Services Architecture: Fundamental Structure

- 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

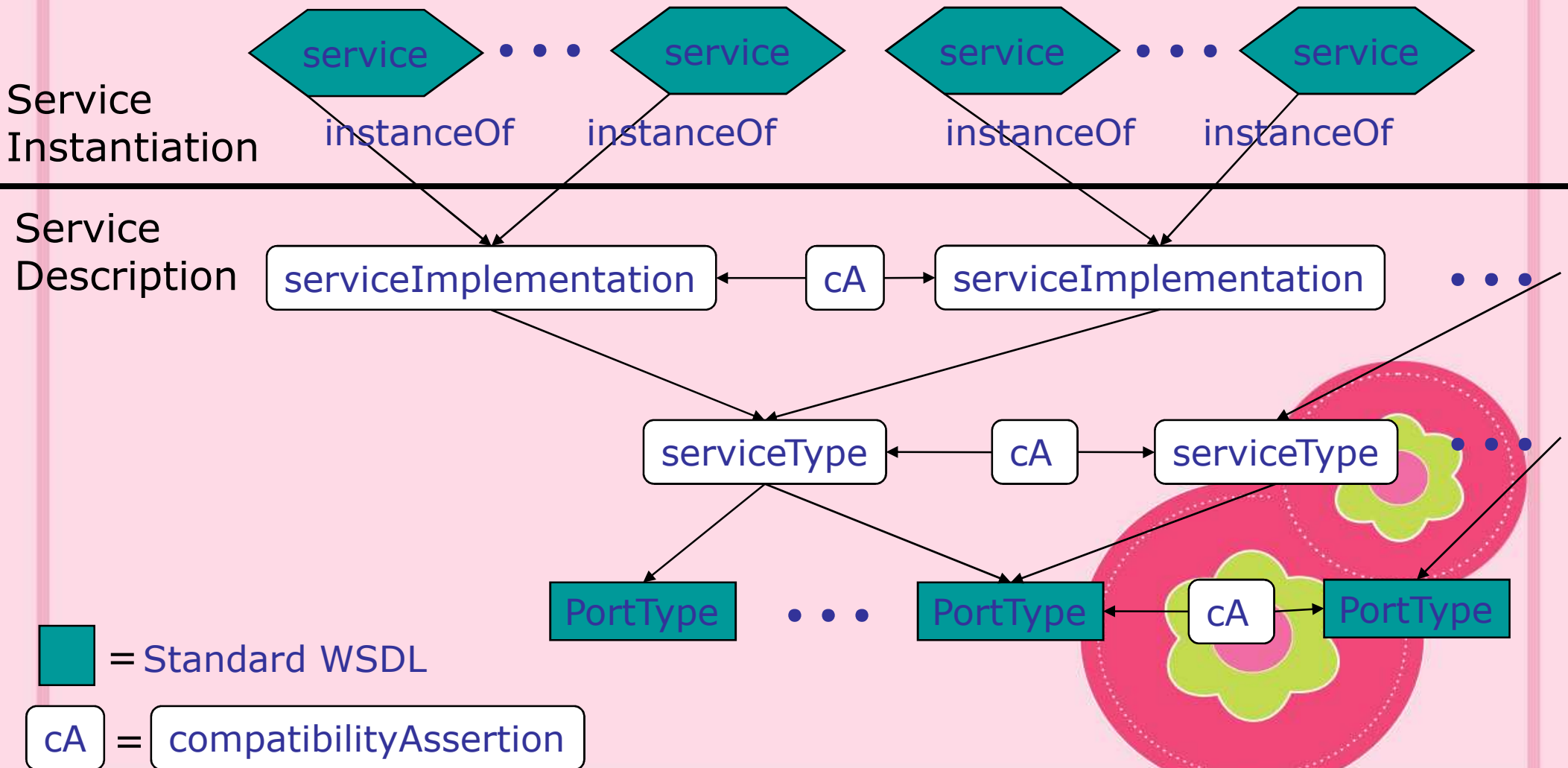


WSDL Conventions & Extensions

- 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 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**

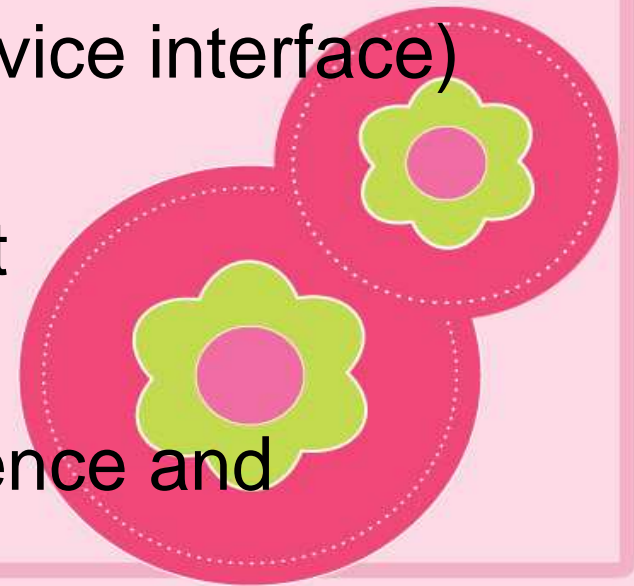
- FindByHandle

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



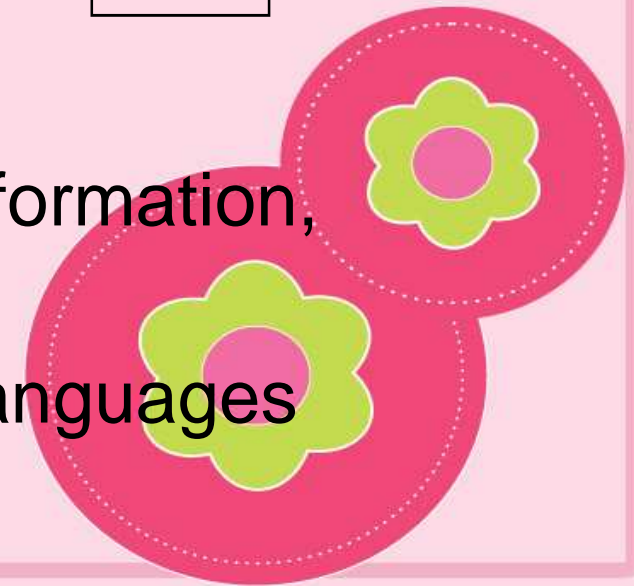
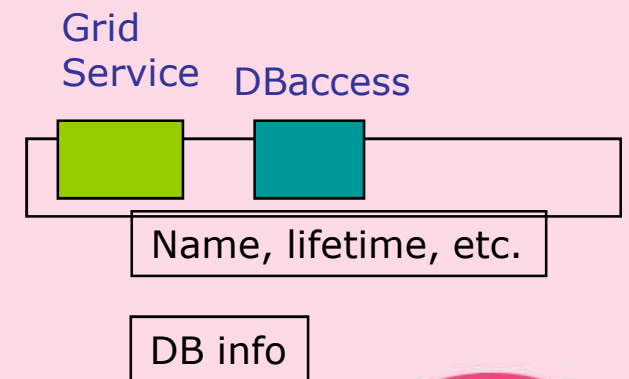
Service Data

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



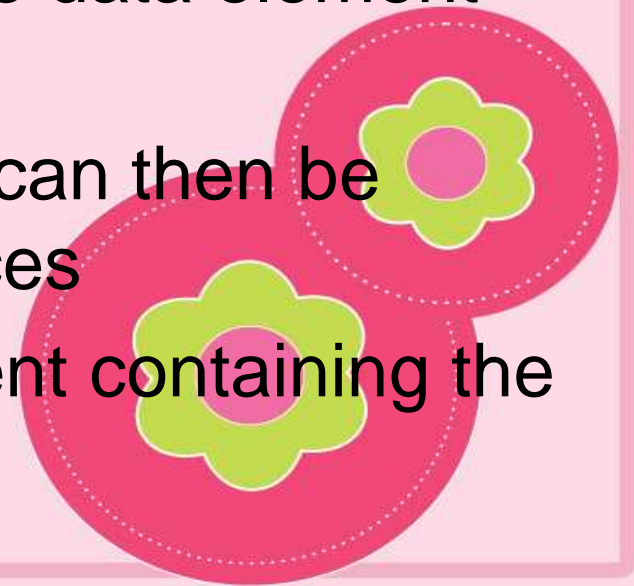
Naming and Bindings

- 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



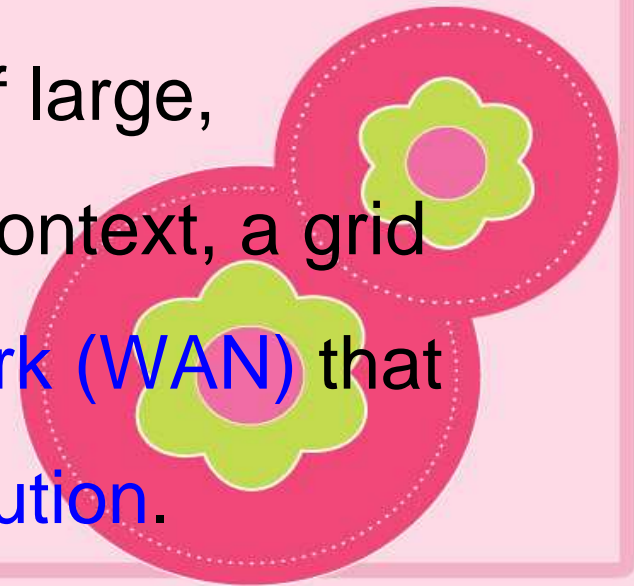
Factory

- **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



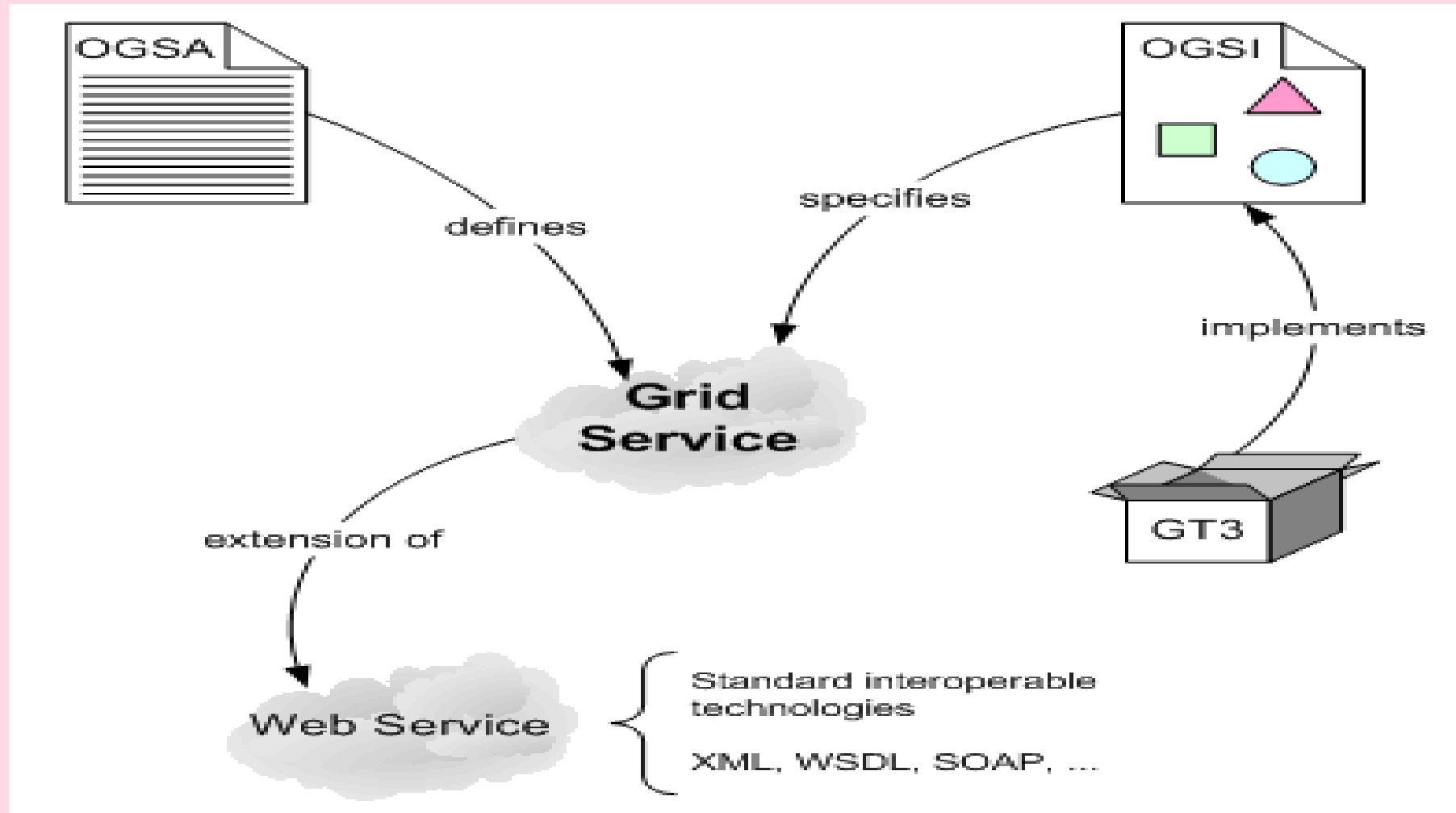
What is the OGSA Standard?

- Open Grid Service Architecture
- OGSA define how different components in grid interact
- Open Grid Services Architecture (OGSA) is a set of standards defining the way in which information is shared among diverse components of large, heterogeneous grid systems. In this context, a grid system is a scalable wide area network (WAN) that supports resource sharing and distribution.



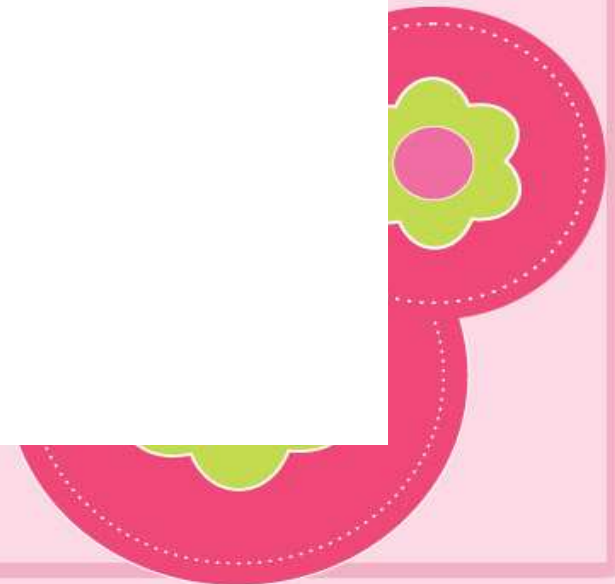
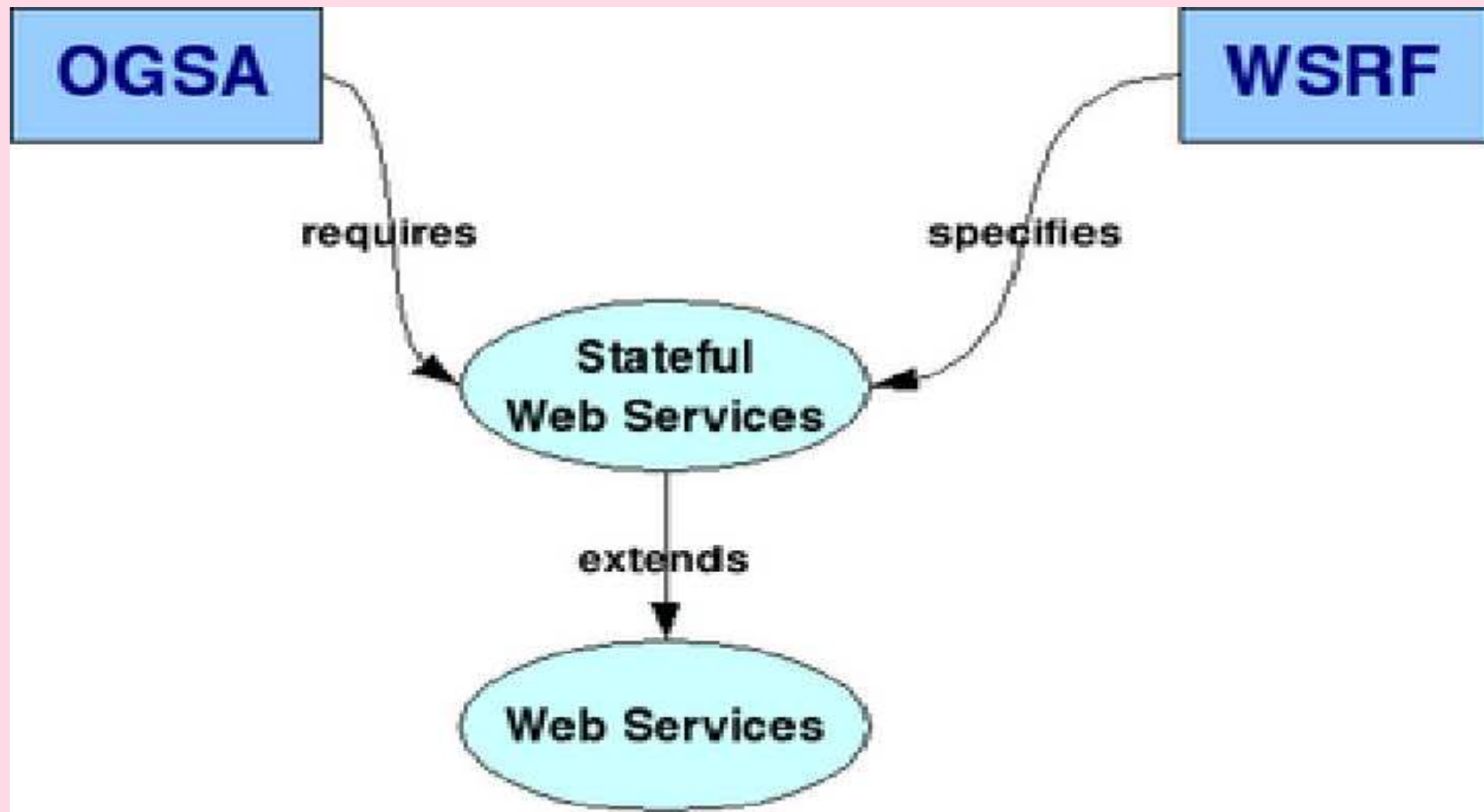
Services in the Web and the Grid

OGSA, OGSi, GT3



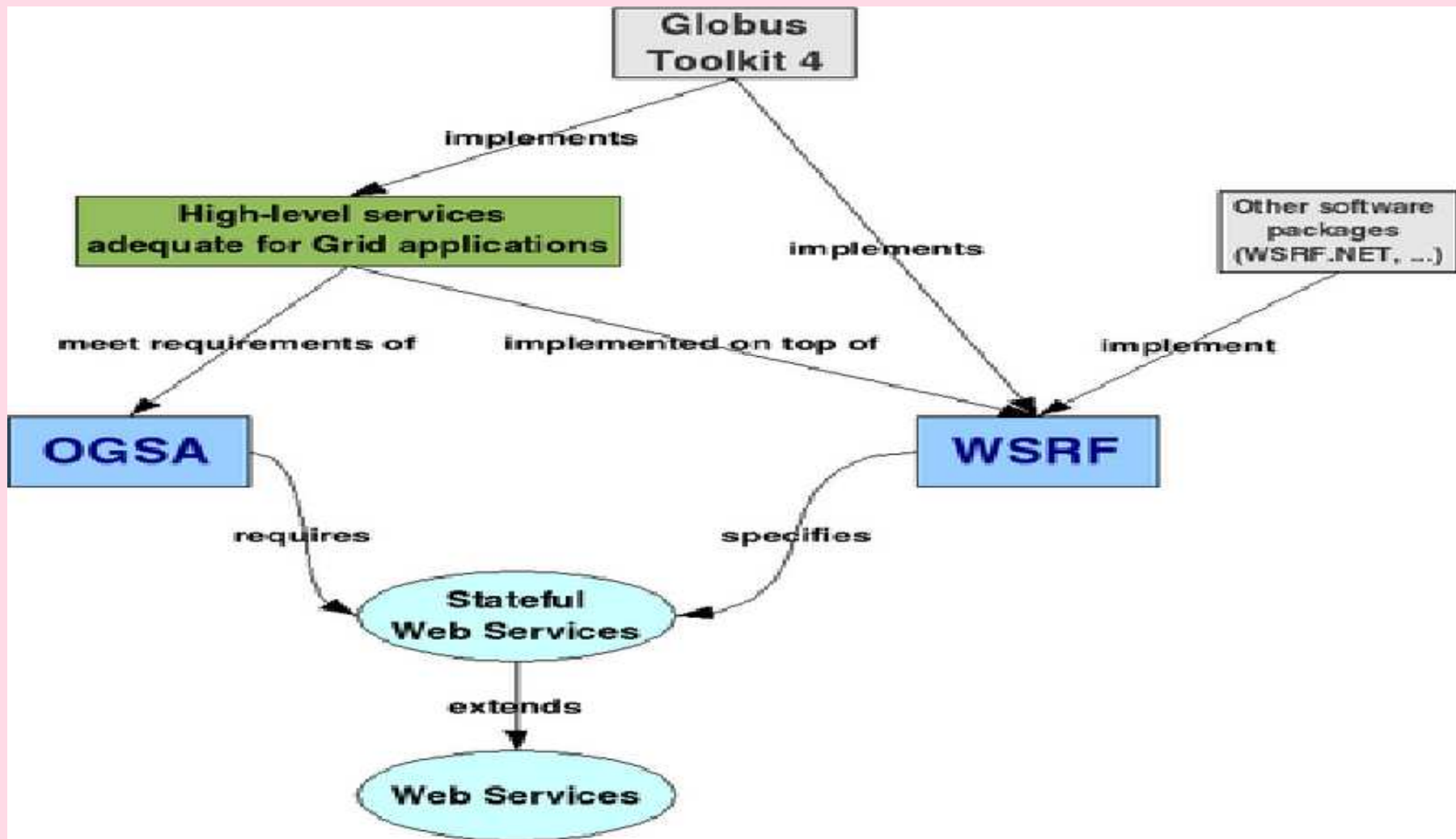
Services in the Web and the Grid

OGSA, WSRF



Web services and the Grid

OGSA, WSRF, GT4



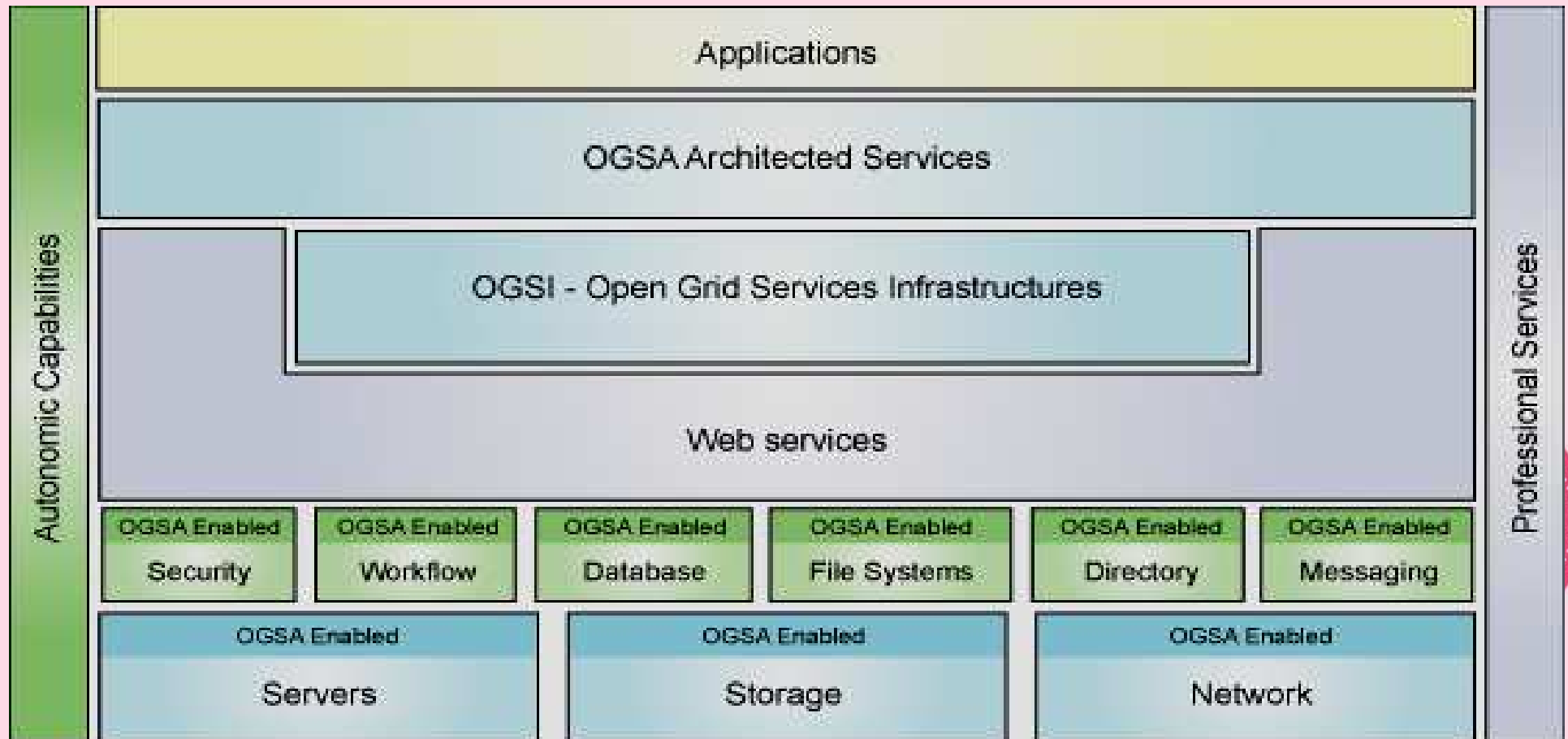
Architecture of OGSA

Comprised of 4 main layers

- Physical and Logical Resources Layer
- Web Service Layer
- OGSA Architected Grid Services Layer
- Grid Applications Layer



OGSA Architecture



OGSA Architecture - Physical and Logical Resources Layer

- **Physical resources** are: servers, storage, network
- **Logical resources** manage physical resources
- Examples of logical resources: database managers, workflow managers

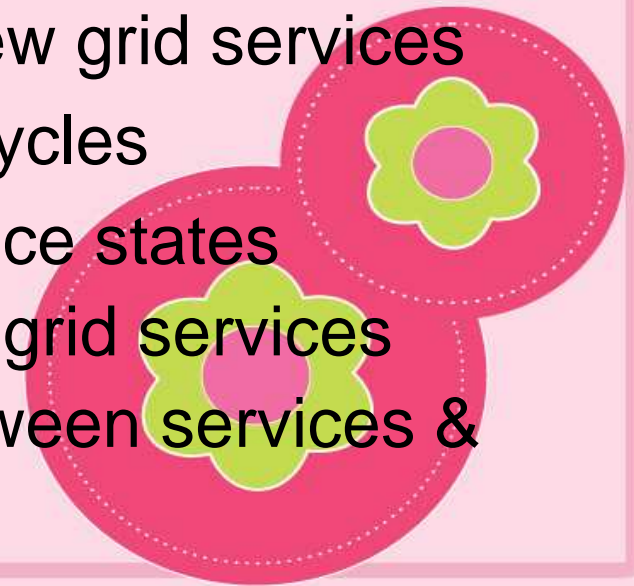


OGSA Architecture - Web Services Layer

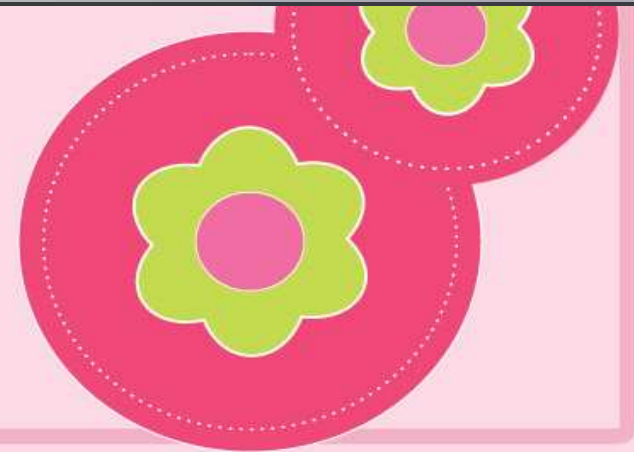
- Web service is software available online that could interact with other software using XML
- Consists of Open Grid Services Infrastructure (OGSI) sub-layer which specifies grid services and provide consistent way to interact with grid services
- Also extends Web Service Capabilities

Consists of 5 interfaces:

1. **Factory:** provide way for creation of new grid services
2. **Life Cycle:** Manages grid service life cycles
3. **State Management:** Manage grid service states
4. **Service Groups:** collection of indexed grid services
5. **Notification:** Manages notification between services & resources



OGSA Architecture - Web Services Layer (OGSI)



OGSA Architecture – OGSA Architected Services - Layer

Classified into 3 service categories

1. Grid Core Services

2. Grid Program Execution Services

3. Grid Data Services



OGSA Architected Services – Grid Core Services

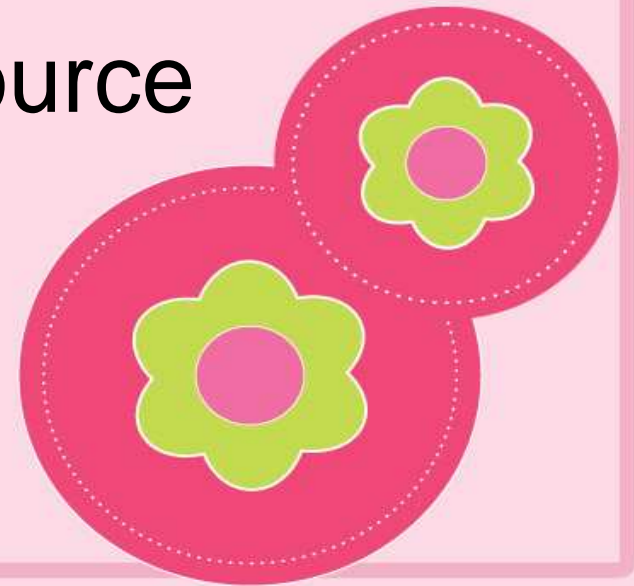
Composed of 4 main types of services:

1. **Service Management**: assist in installation, maintenance, & troubleshooting tasks in grid system
2. **Service Communication**: include functions that allow grid services to communicate
3. **Policy Services**: Provide framework for creation, administration & management of policies for system operation
4. **Security Services**: provide authentication & authorization mechanisms to ensure systems interoperate securely



OGSA Architected Services – Grid Program Execution Services

- Supports unique grid systems in high performance computing, collaboration, parallelism
- Support virtualization of resource processing

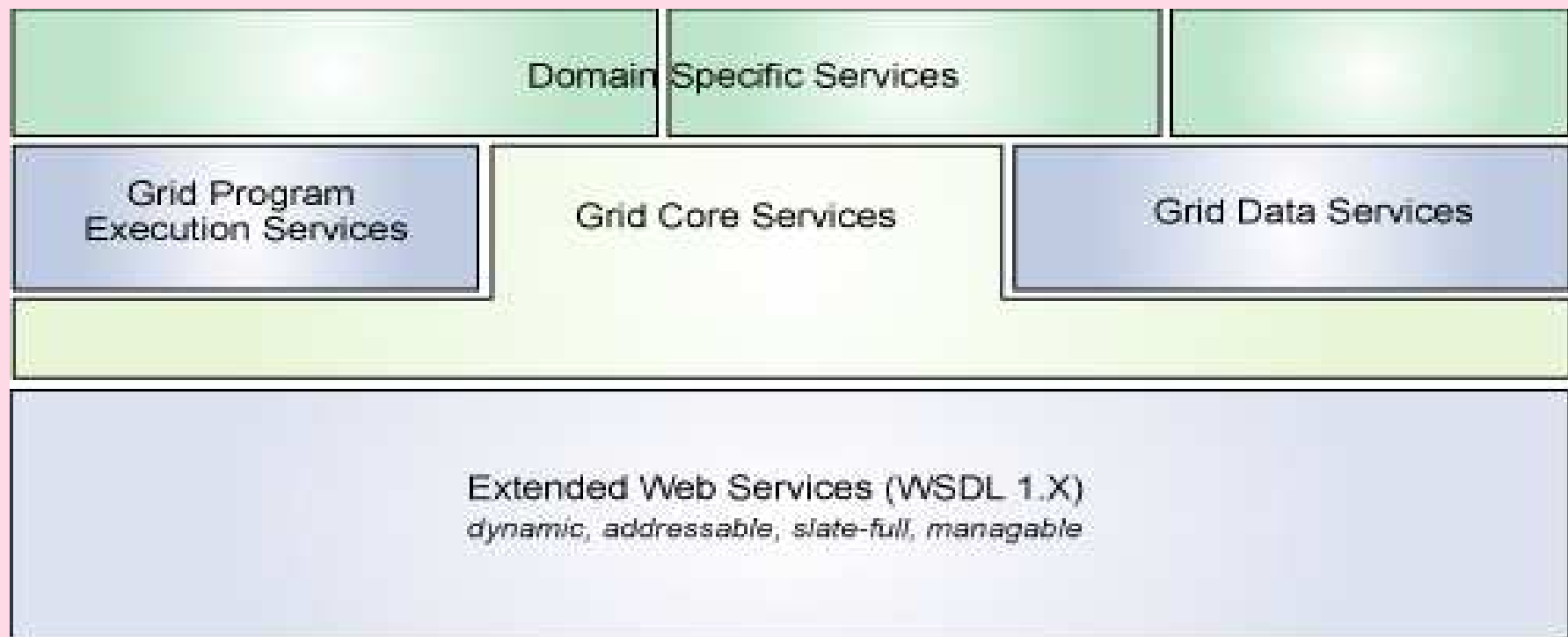


OGSA Architected Services – Grid Data Services

- Support data virtualization
- Provide mechanism for access to distributed resources such as databases, files

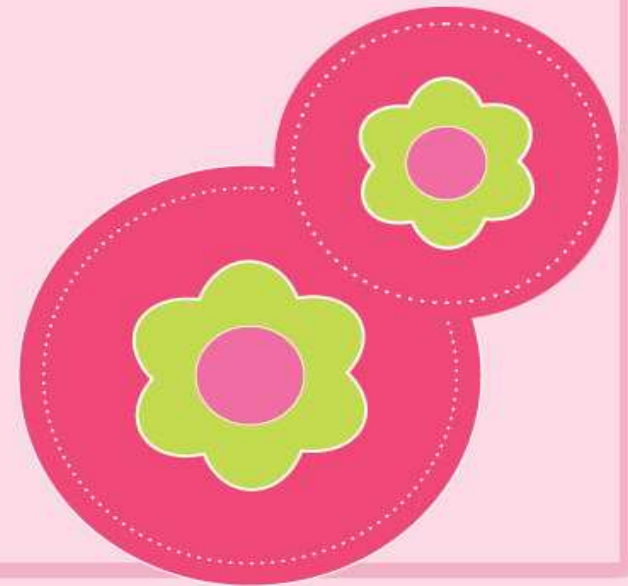


OGSA Architecture – OGSA Architected Services - Layer

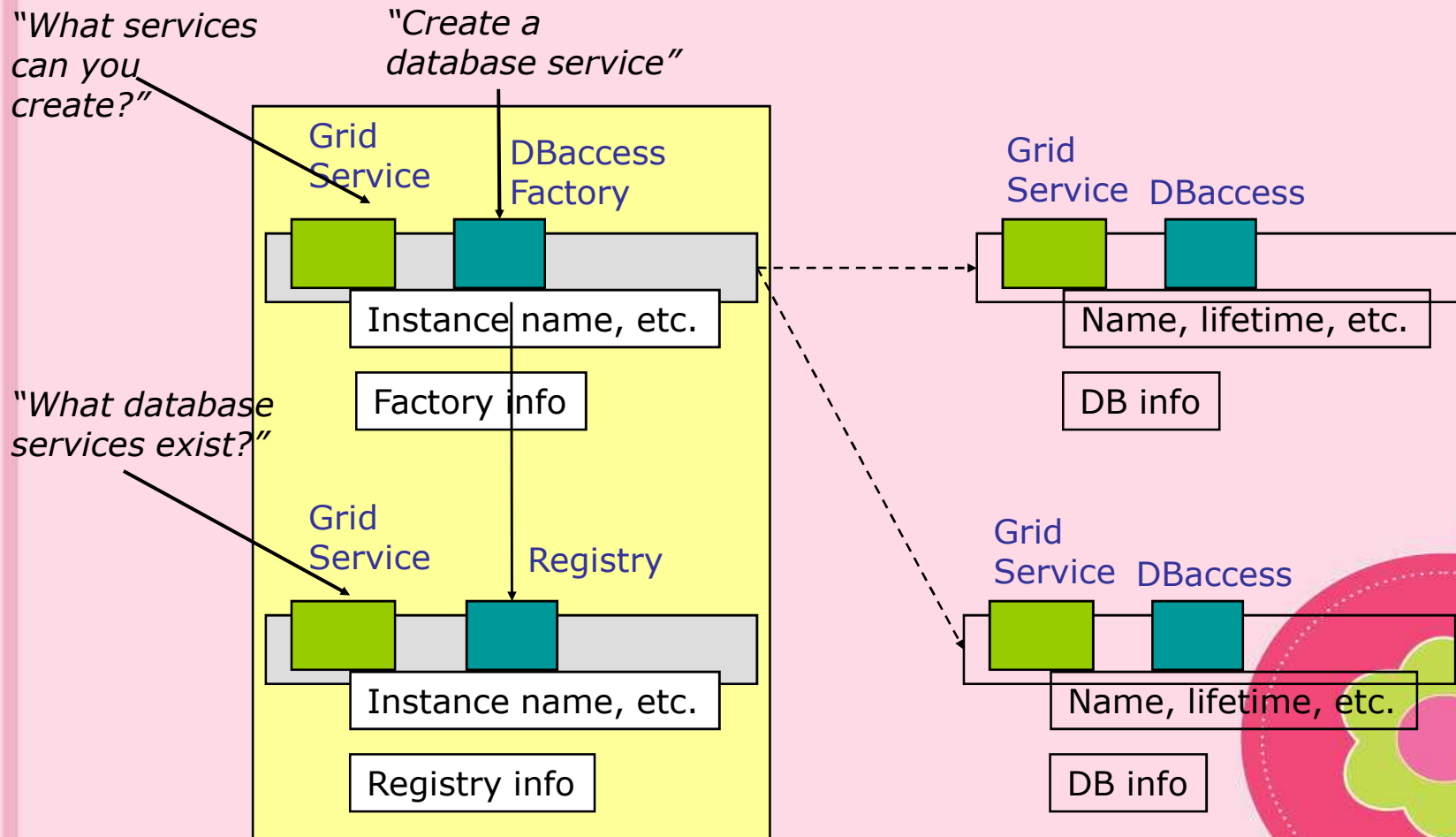


OGSA Architecture – Grid Applications Layer

- This layer comprise of applications that use the grid architected services

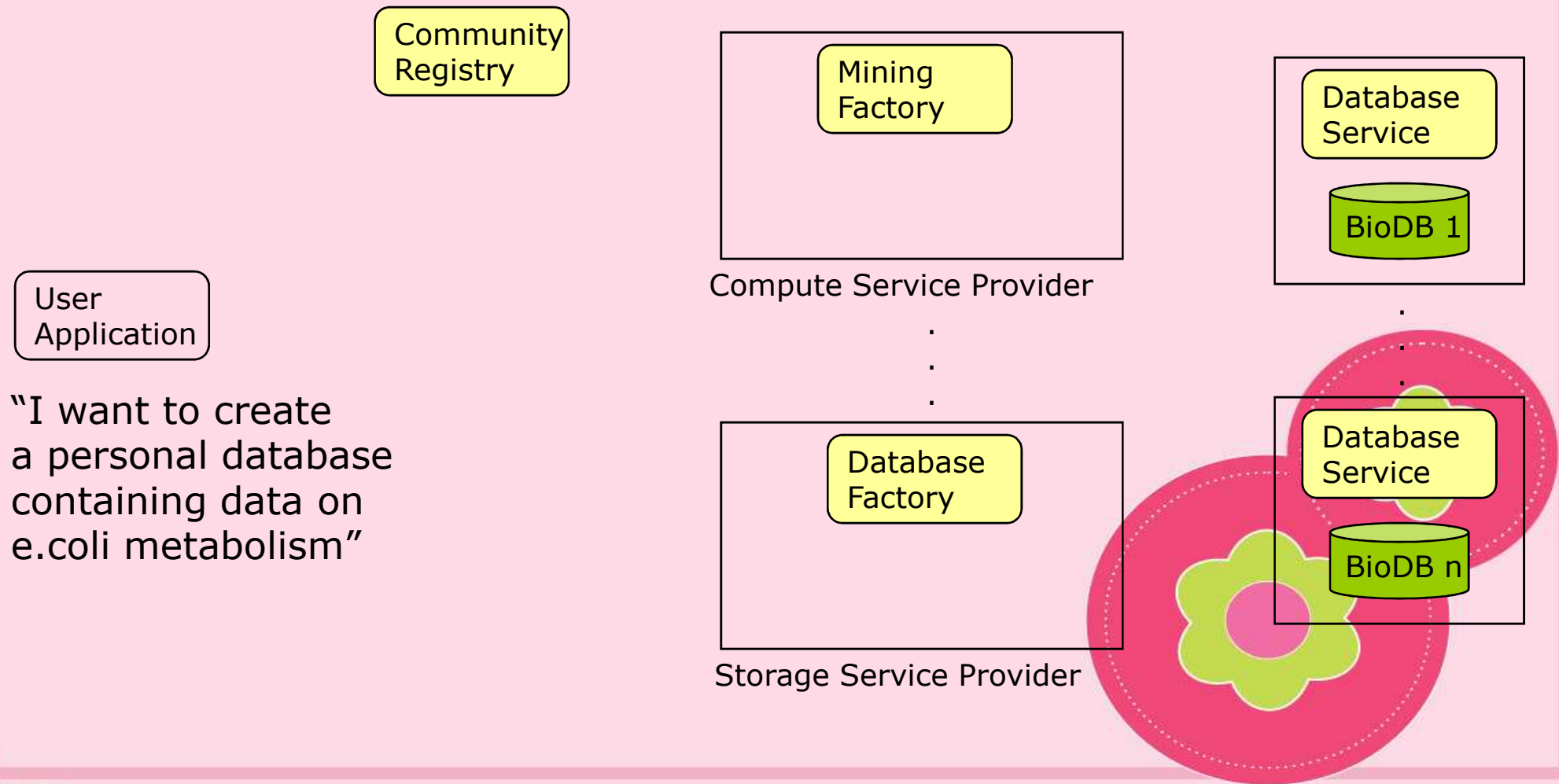


Transient Database Services



Example:

Data Mining for Bioinformatics



Example:

Data Mining for Bioinformatics

“Find me a data mining service, and somewhere to store data”

Community Registry

User Application

Mining Factory

Compute Service Provider

Database Service

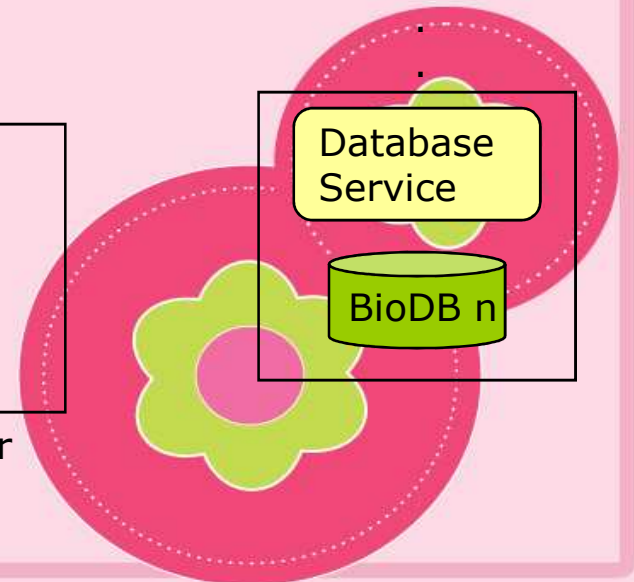
BioDB 1

Database Factory

Storage Service Provider

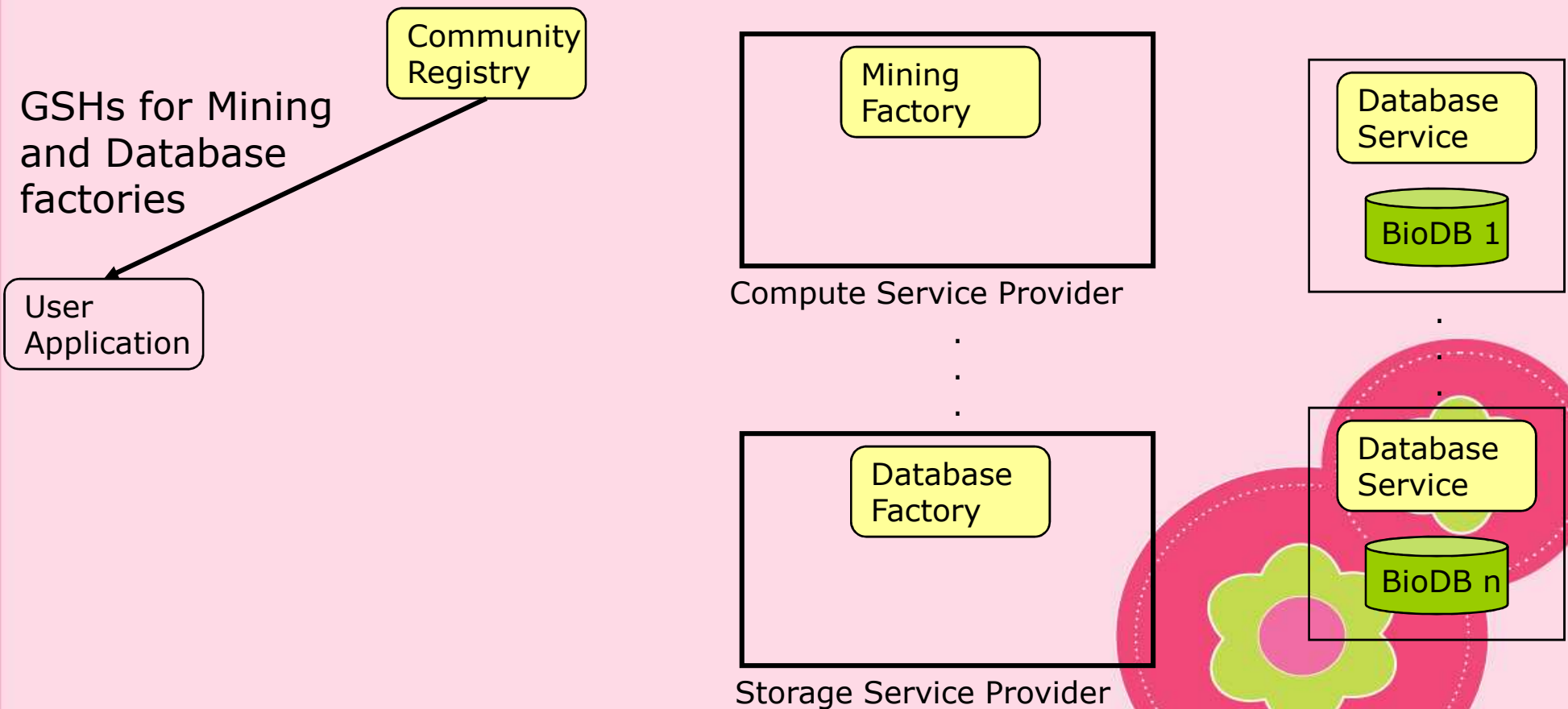
Database Service

BioDB n



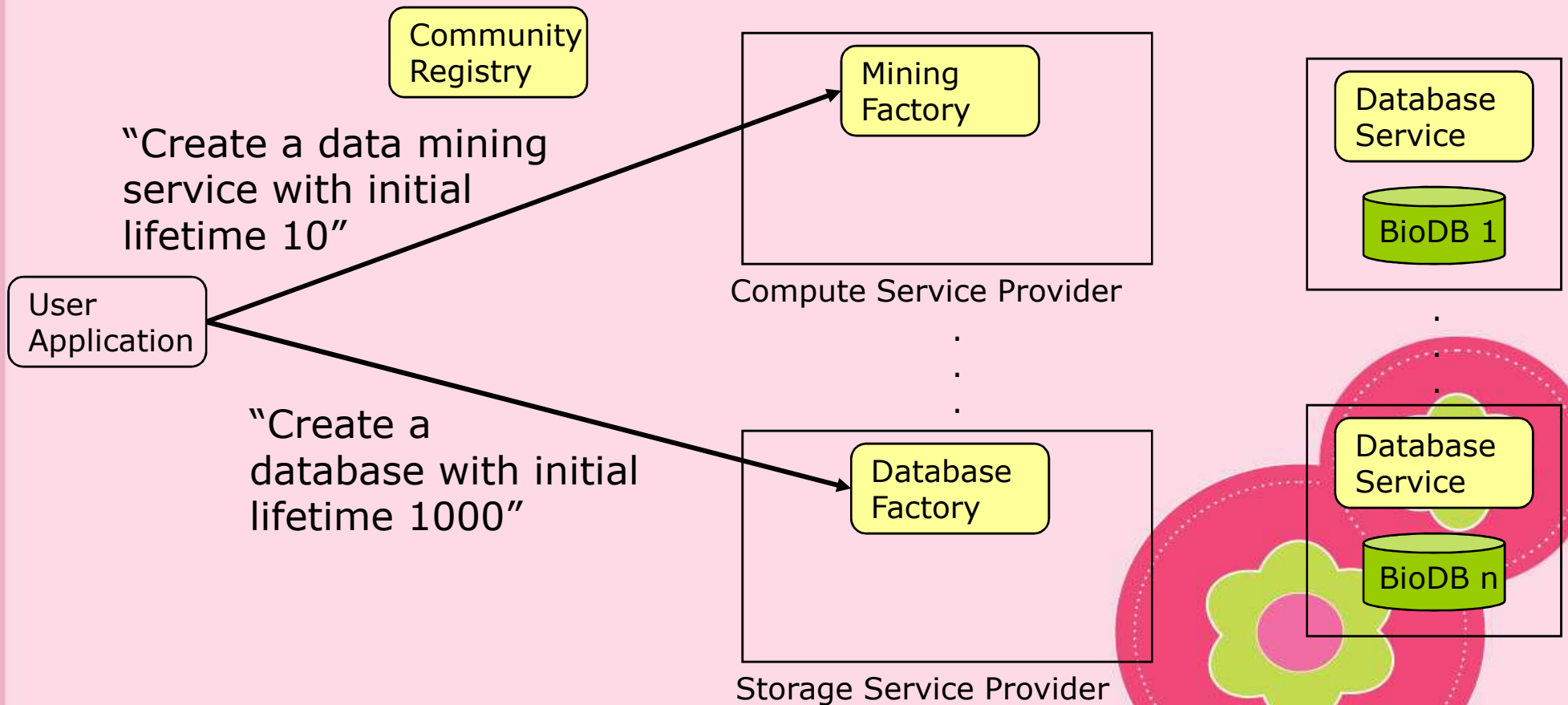
Example:

Data Mining for Bioinformatics



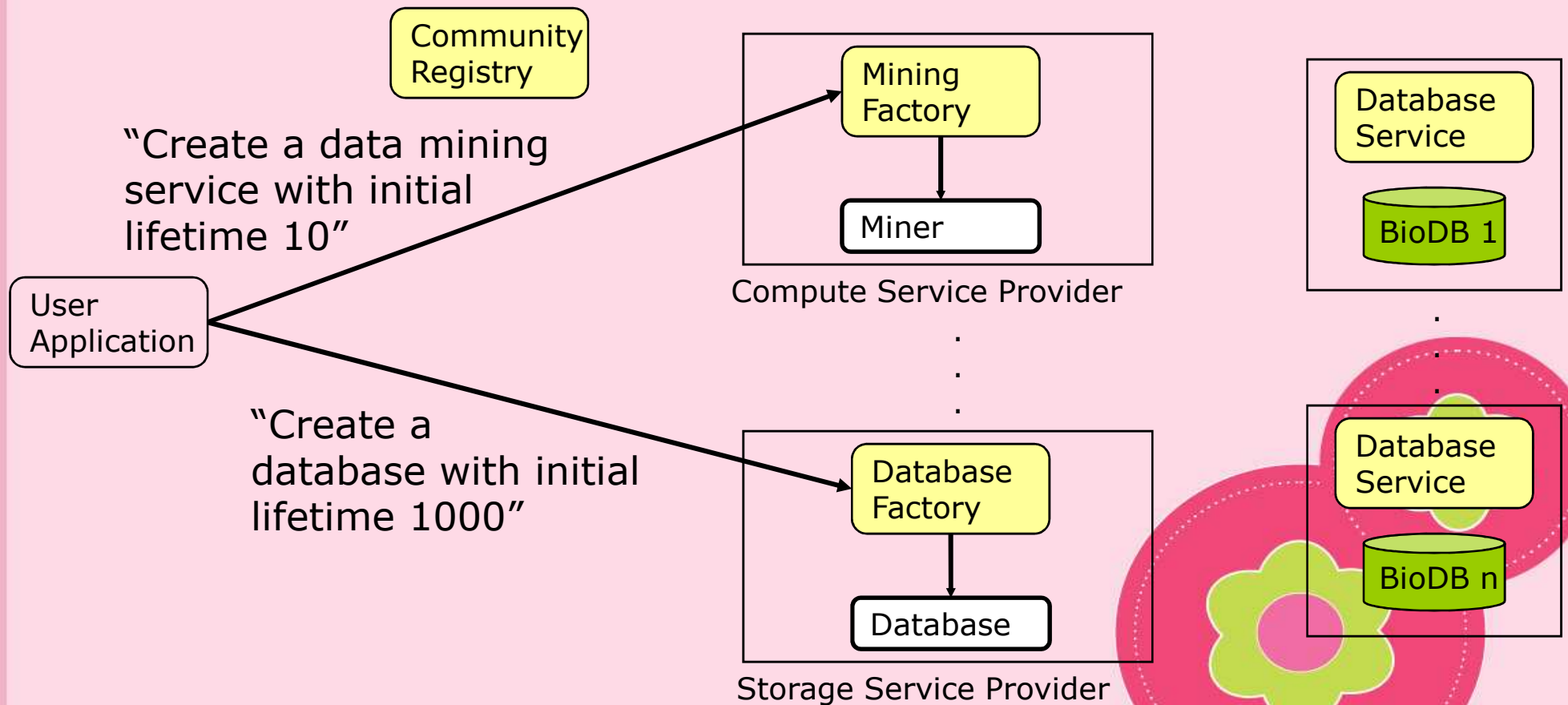
Example:

Data Mining for Bioinformatics

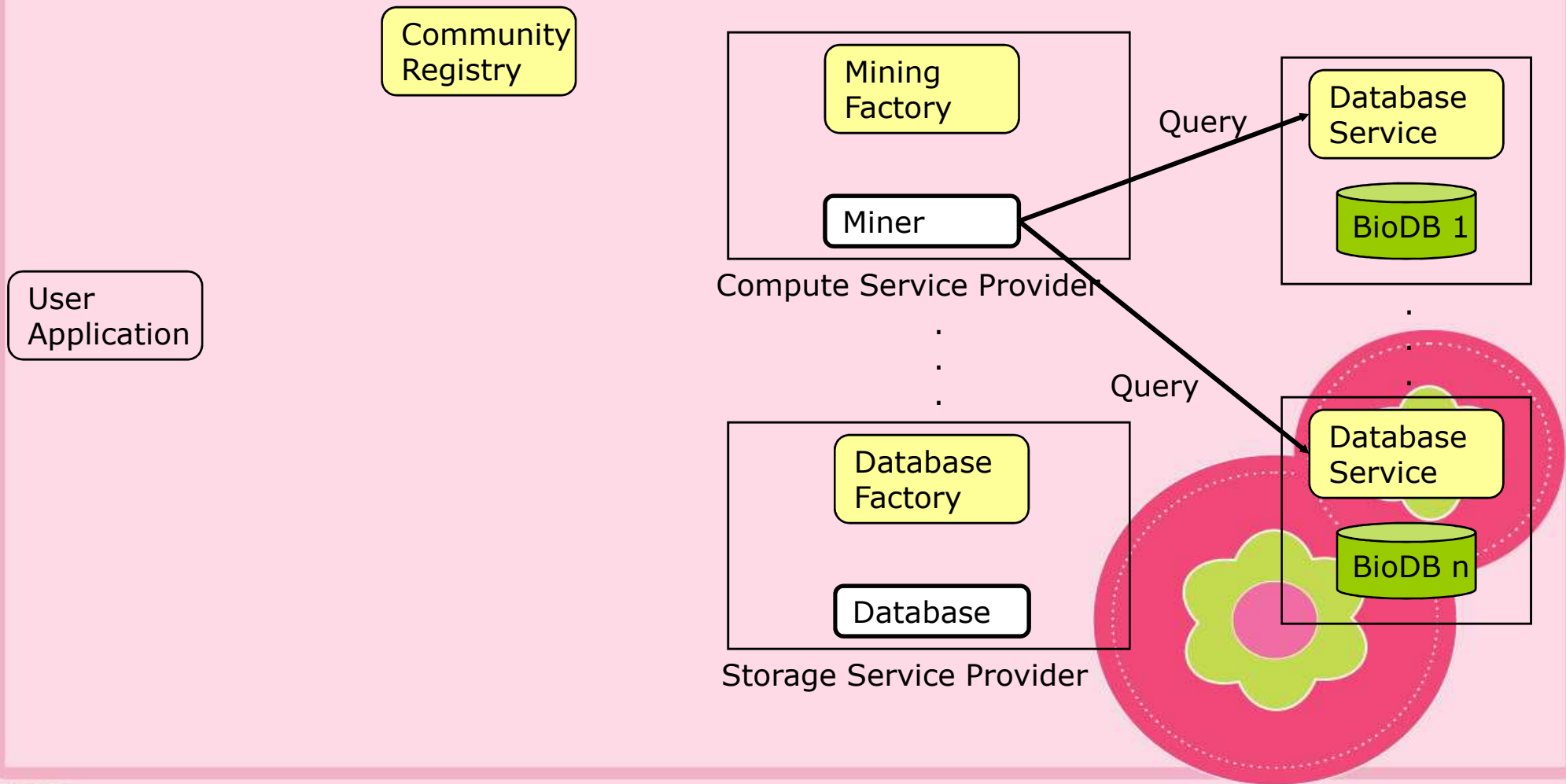


Example:

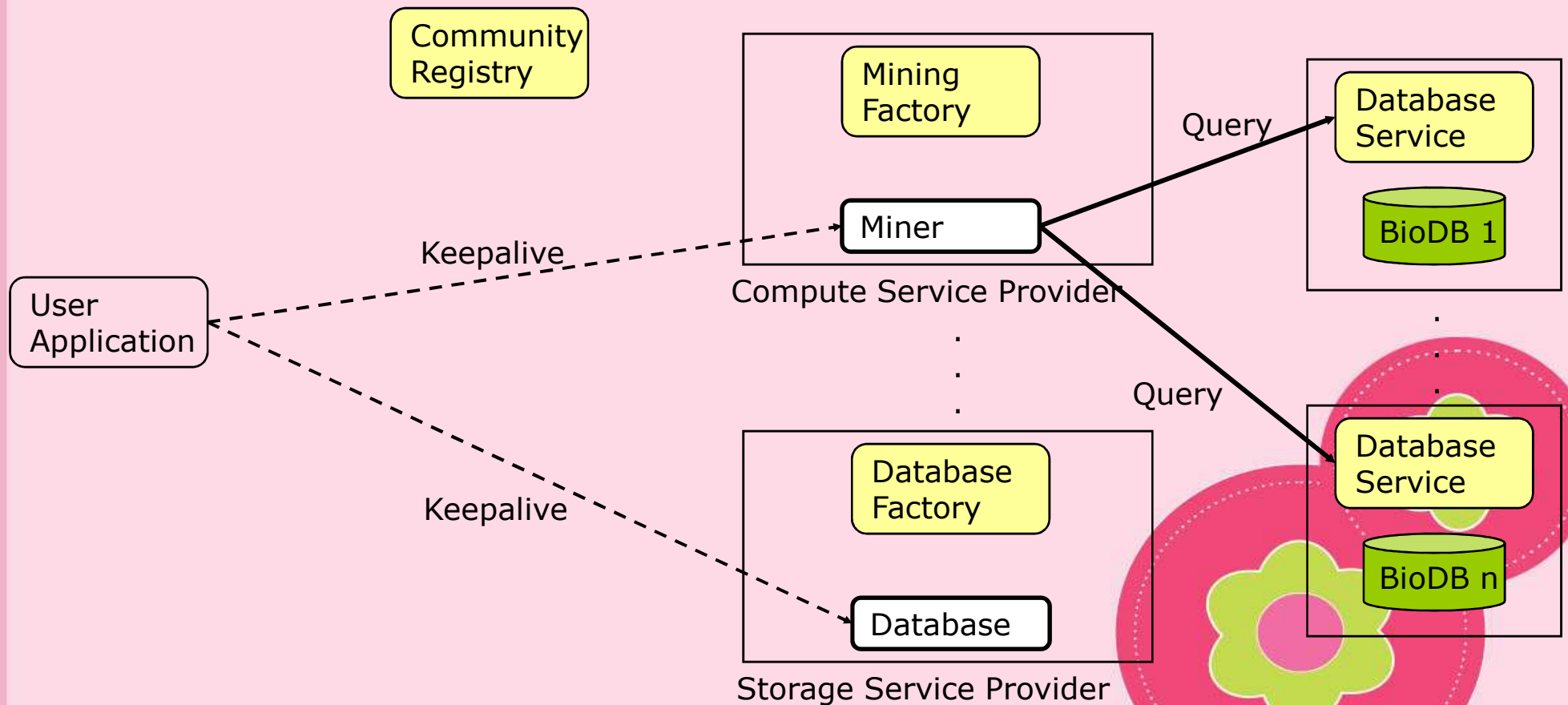
Data Mining for Bioinformatics



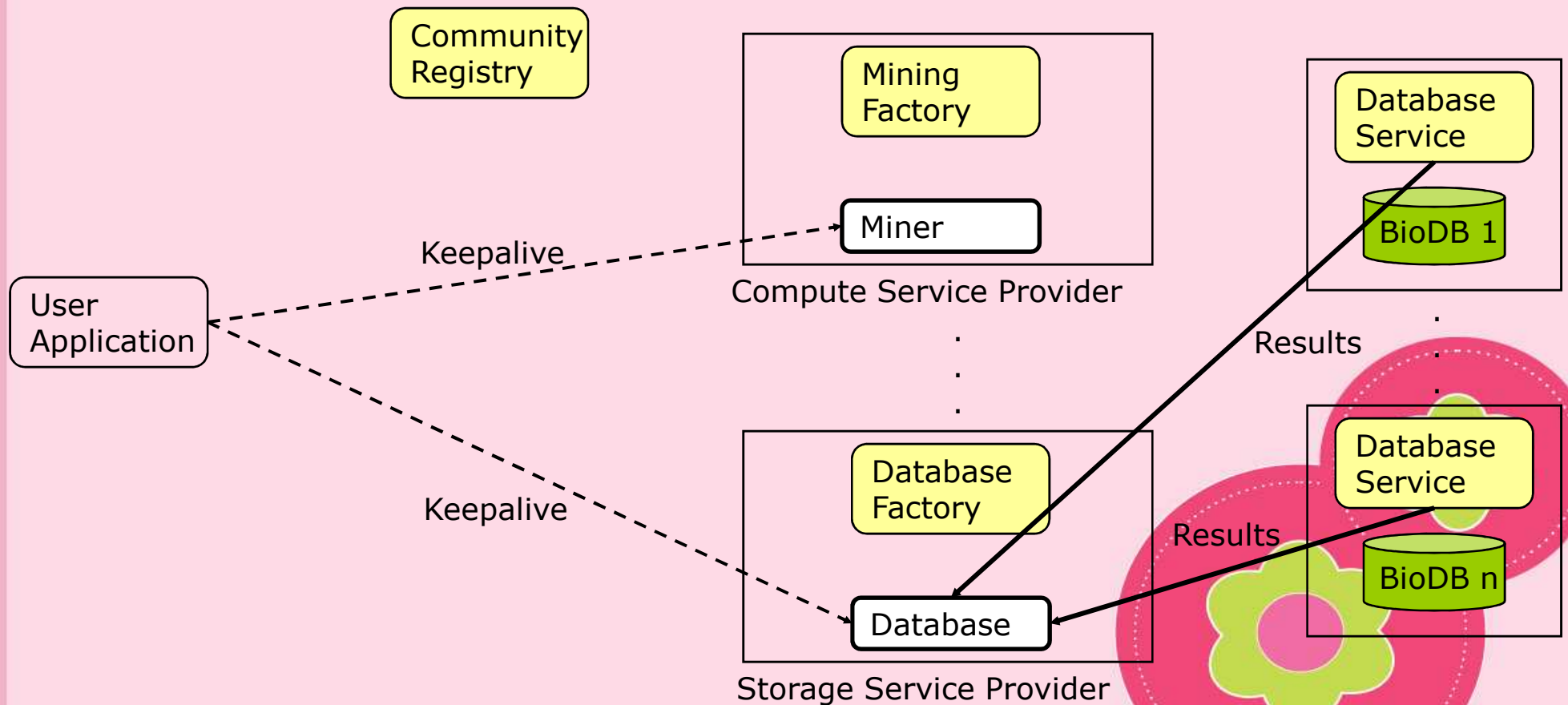
Example: Data Mining for Bioinformatics



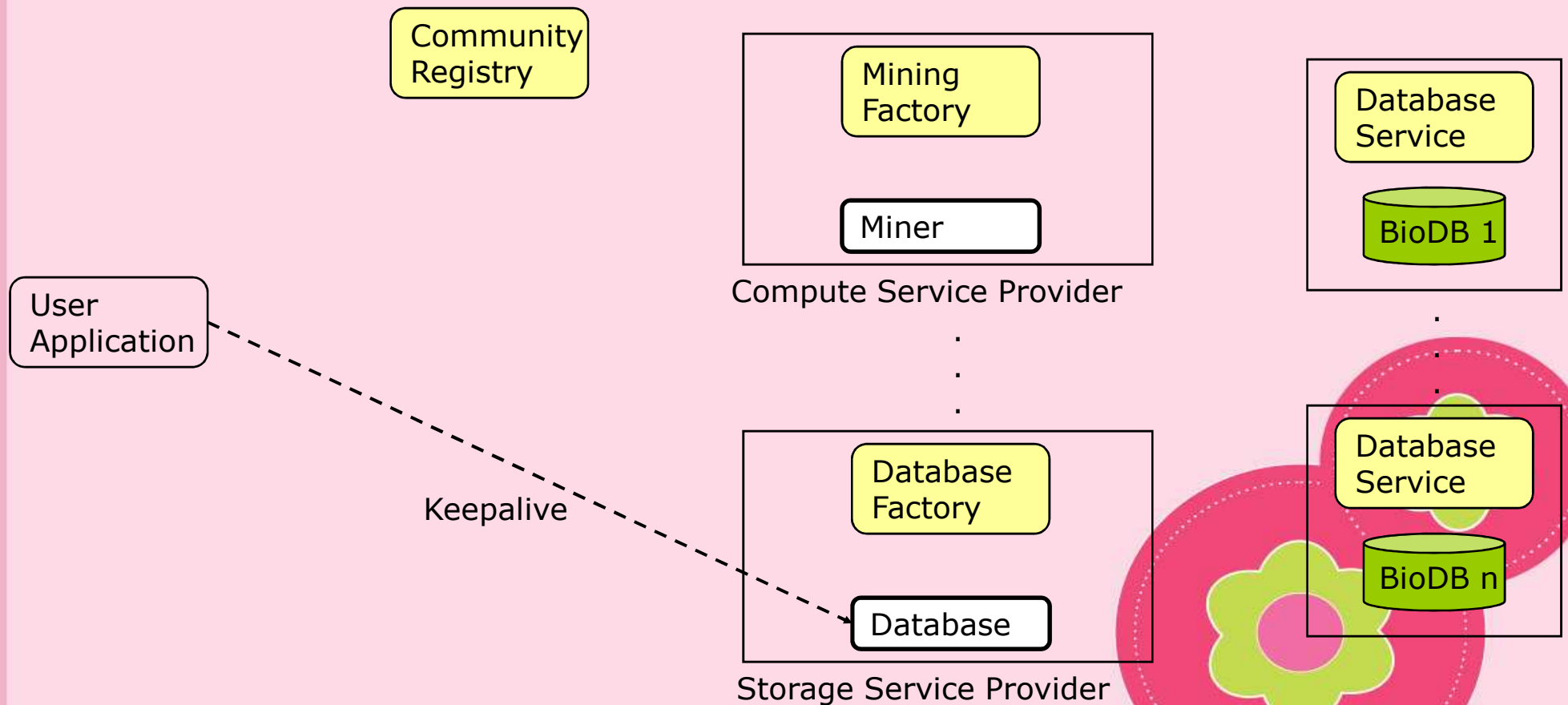
Example: Data Mining for Bioinformatics



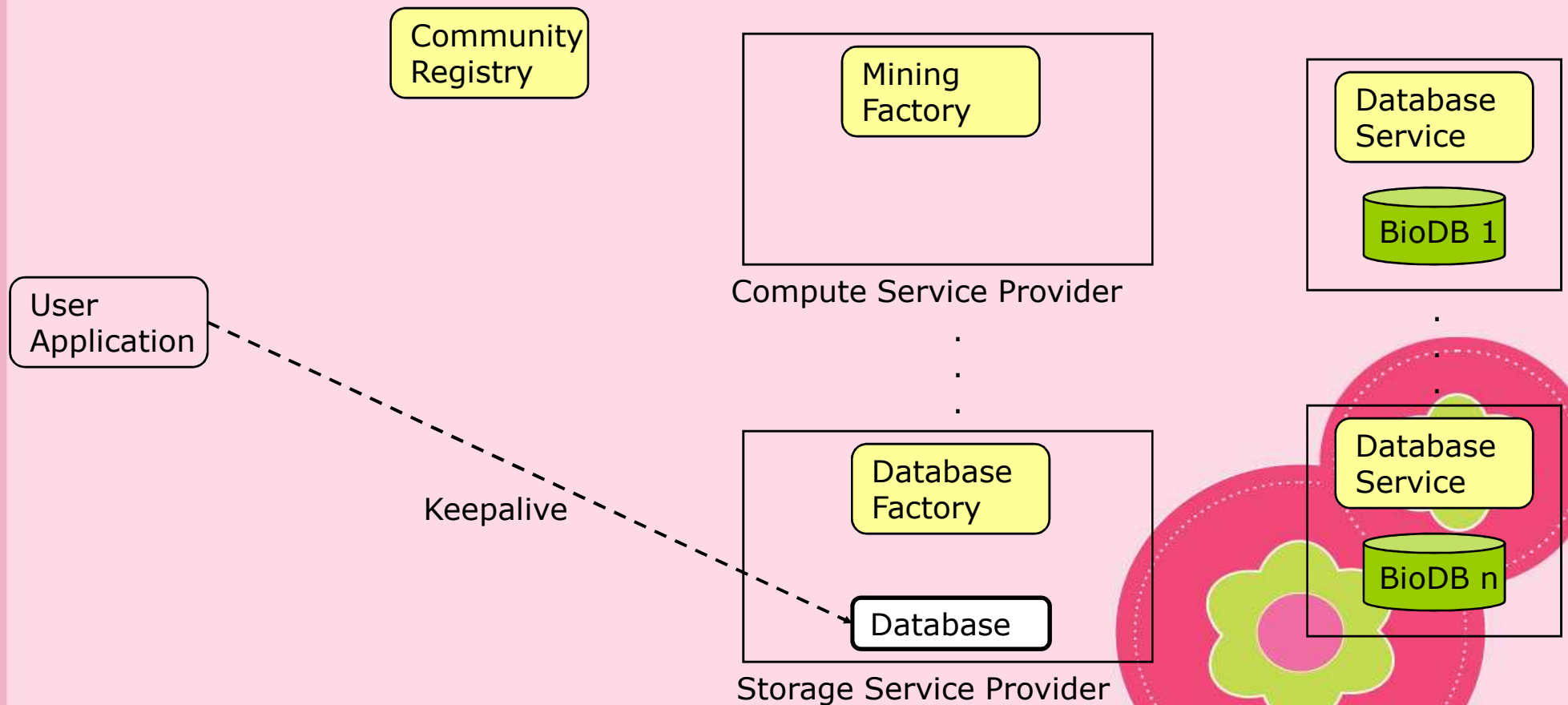
Example: Data Mining for Bioinformatics



Example: Data Mining for Bioinformatics



Example: Data Mining for Bioinformatics



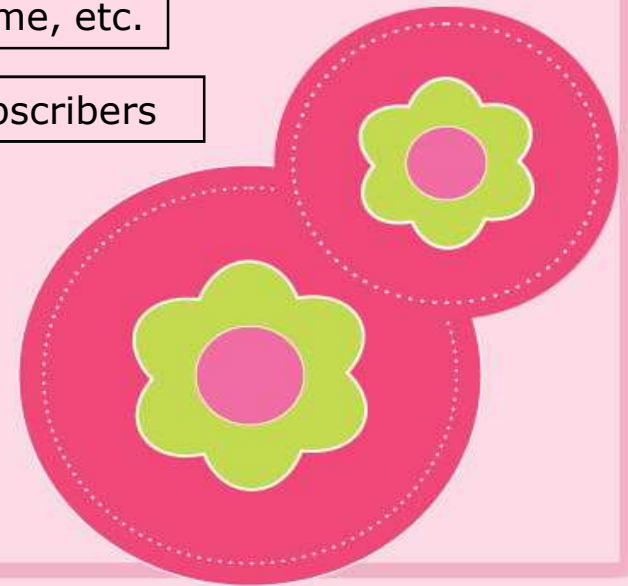
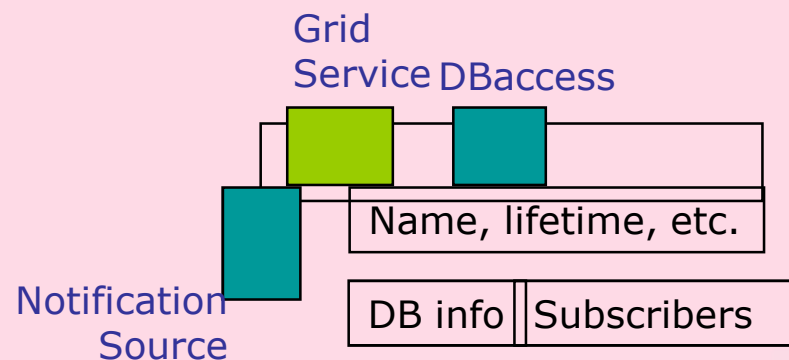
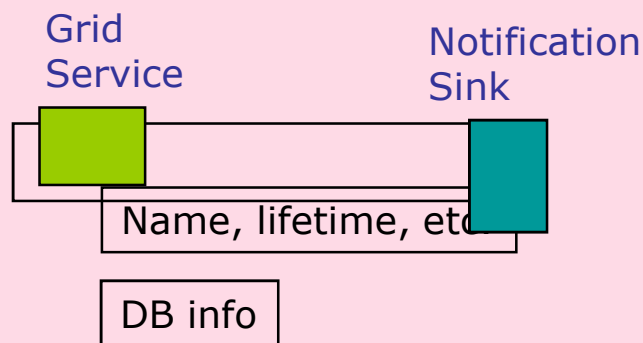
Notification Interfaces

- **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, ...



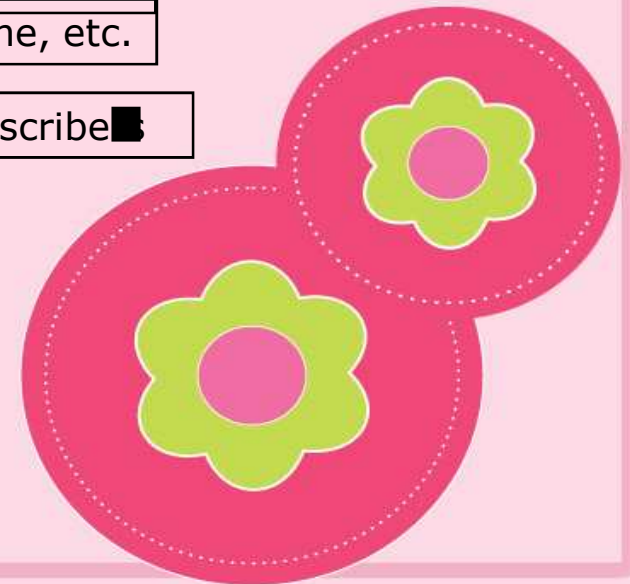
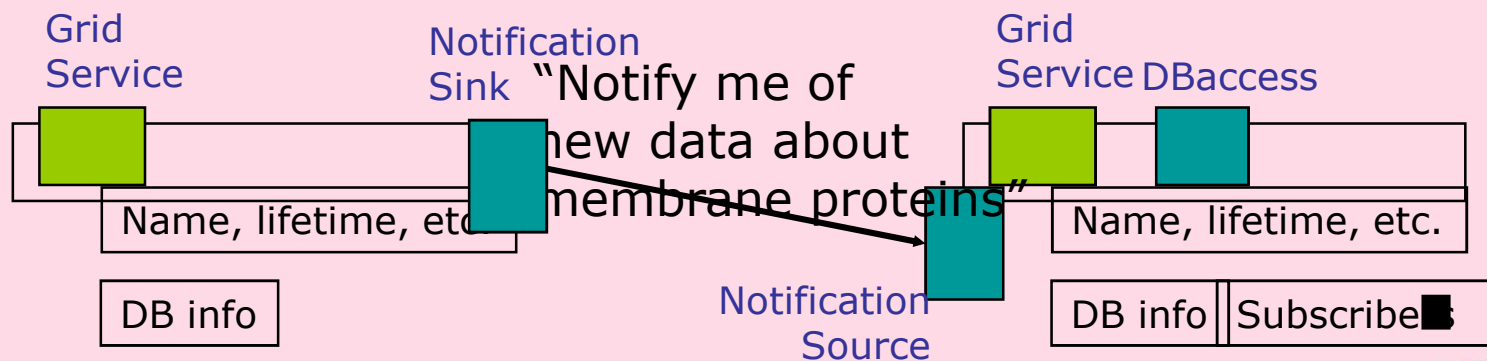
Notification Example

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



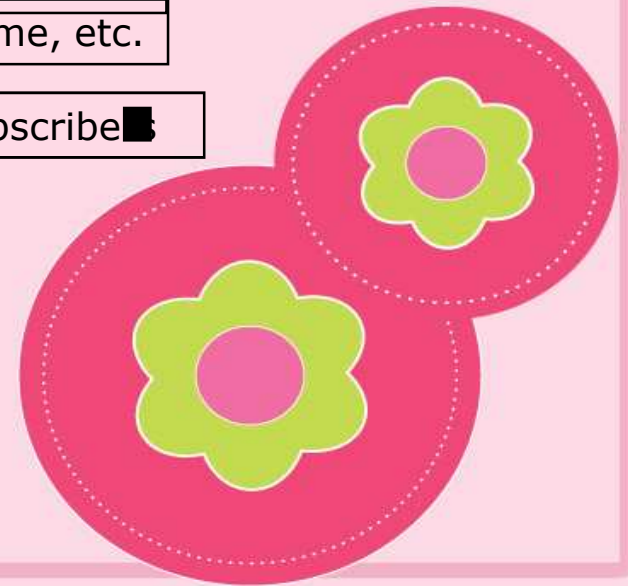
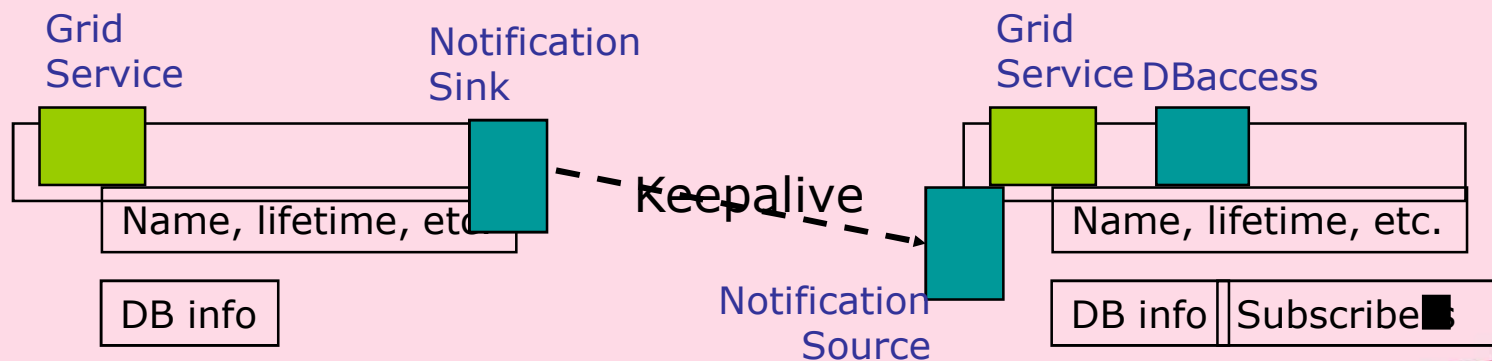
Notification Example

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



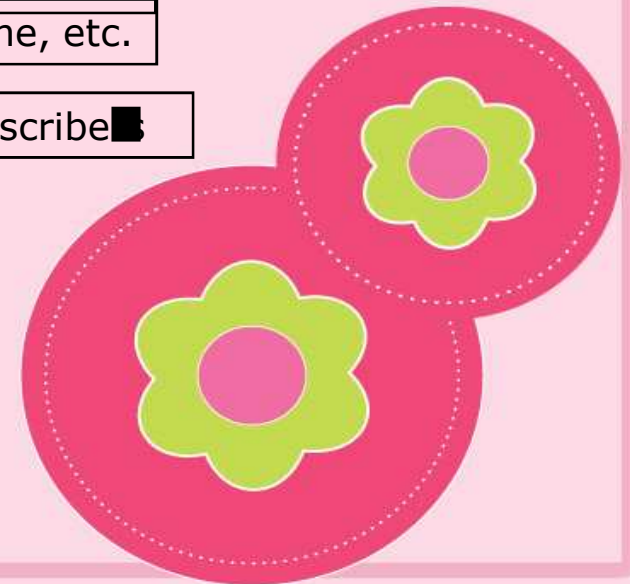
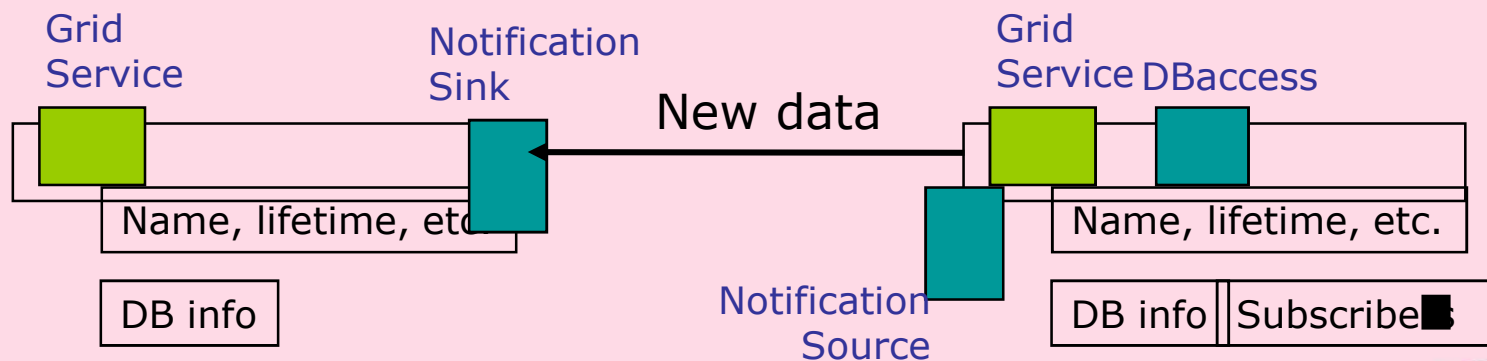
Notification Example

- 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, ...



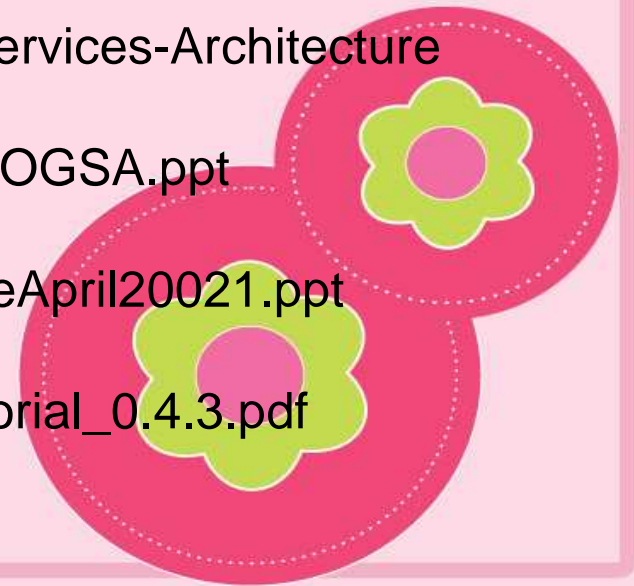
Conclusion

- Grid-Computing allows networked resources to be combined and used
- Grid-Computing offers great benefit to an organization
- OGSA are comprehensive standards which governs grid-computing



References

1. Kai Hwang, Geoffery C. Fox and Jack J. Dongarra, "Distributed and Cloud Computing: Clusters, Grids, Clouds and the Future of Internet",
2. <https://www.dcc.fc.up.pt/~ines/aulas/1213/CG/OGSA.ppt>
3. <http://www.computerworld.com/article/2552339/networking/open-grid-services-architecture.html>
4. <http://searchsoa.techtarget.com/definition/Open-Grid-Services-Architecture>
5. www.cs.umsl.edu/~sanjiv/classes/cs6740/presentation/OGSA.ppt
6. www.nesc.ac.uk/news/.../OpenGridServicesArchitectureApril20021.ppt
7. www.cse.buffalo.edu/~bina/cse486/spring2011/progtutorial_0.4.3.pdf



Thank You

Questions and Comments?

