Web Service & Grid Service

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
- 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)

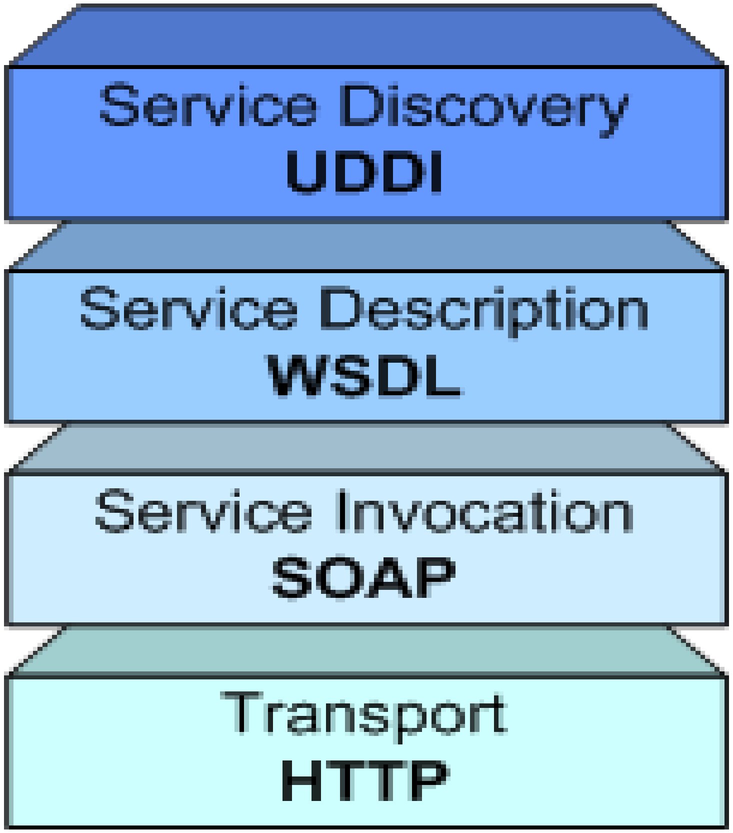
Services in the Web and the Grid

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)

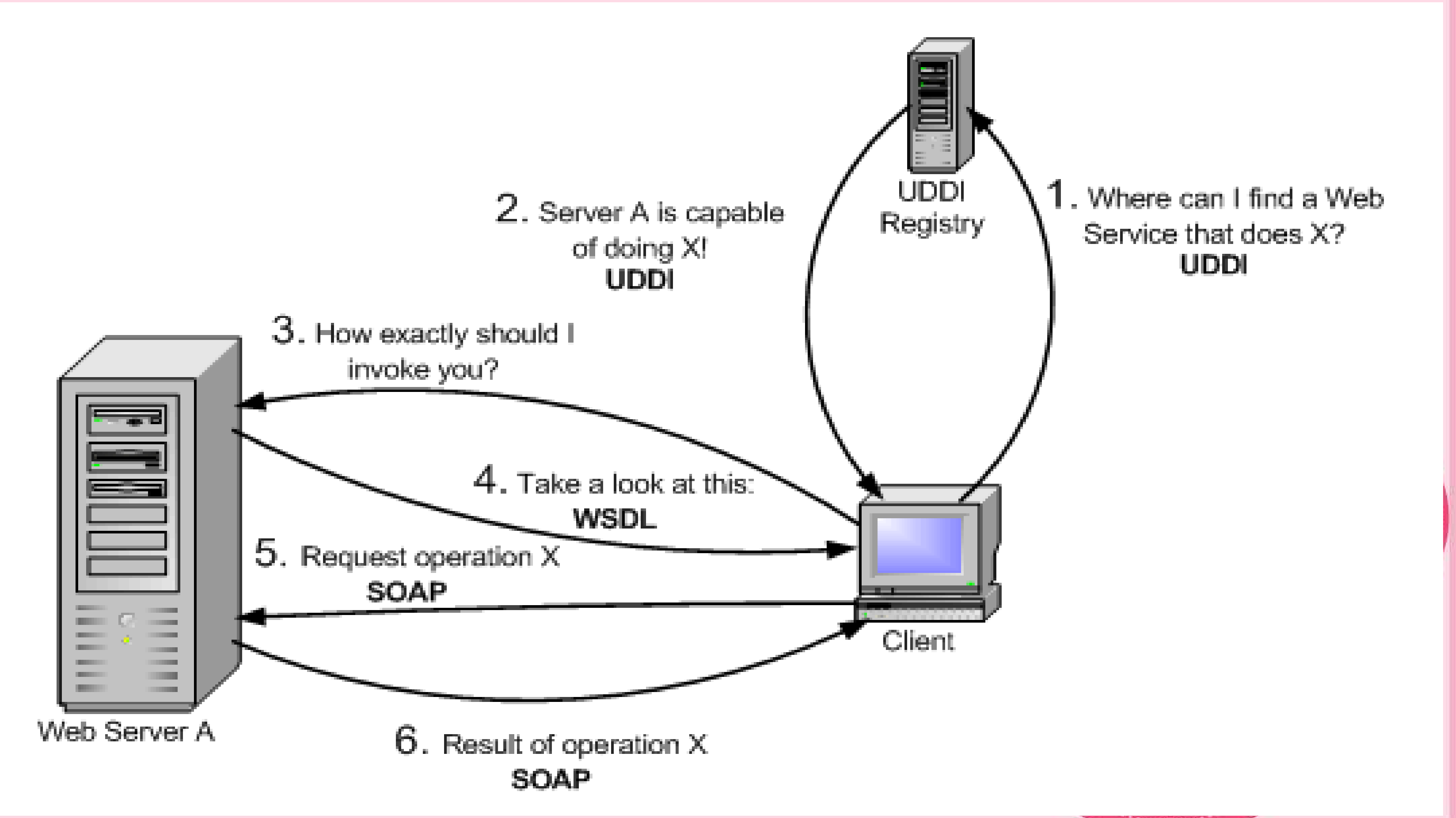
Format of requests(client) and responses (server)
(Simple Object Access Protocol)

Message transfer protocol

(Hypertext Transfer Protocol)

Services in the Web and the Grid

Invoking a Typical Web Service



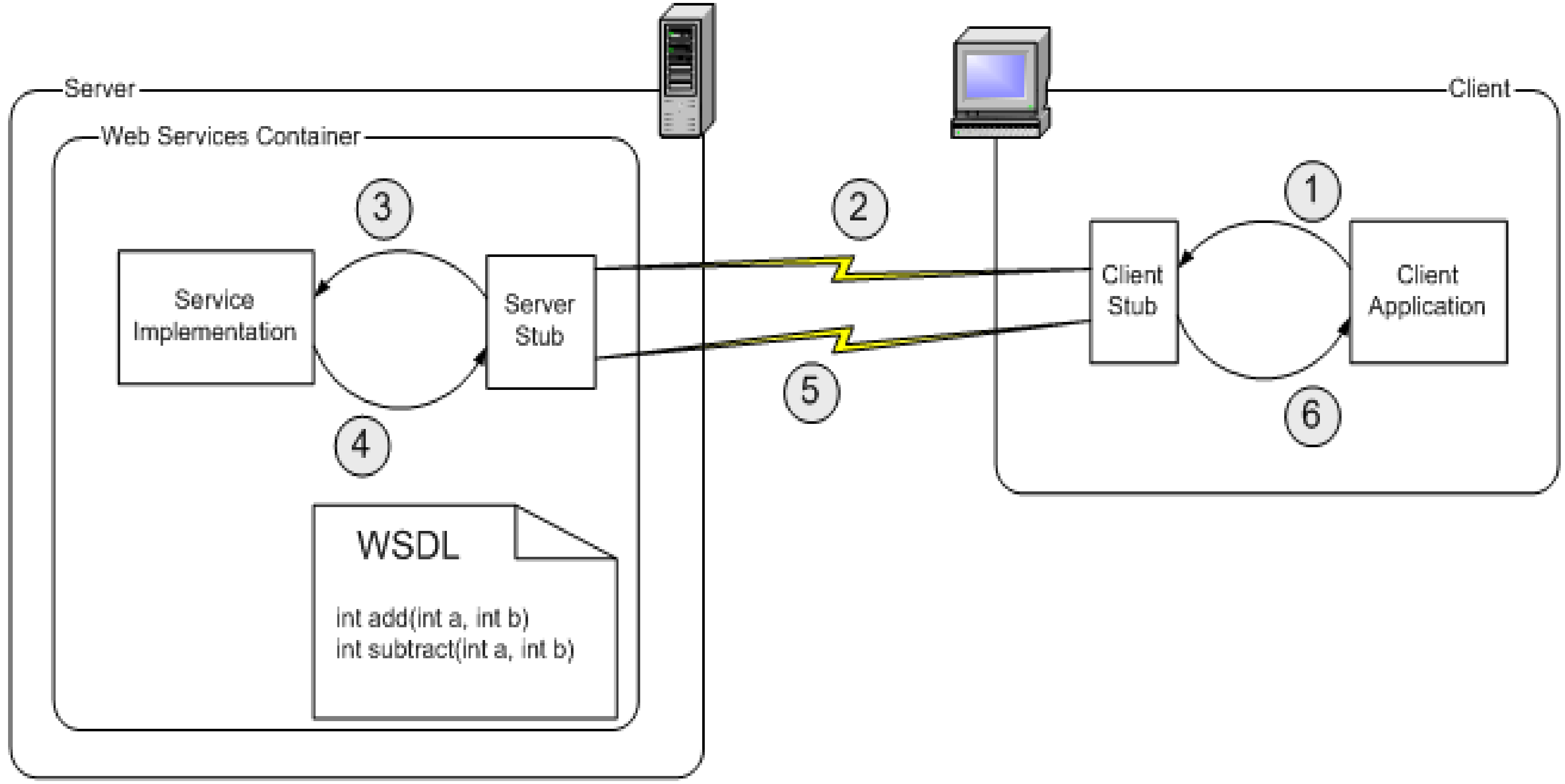
Services in the Web and the Grid

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



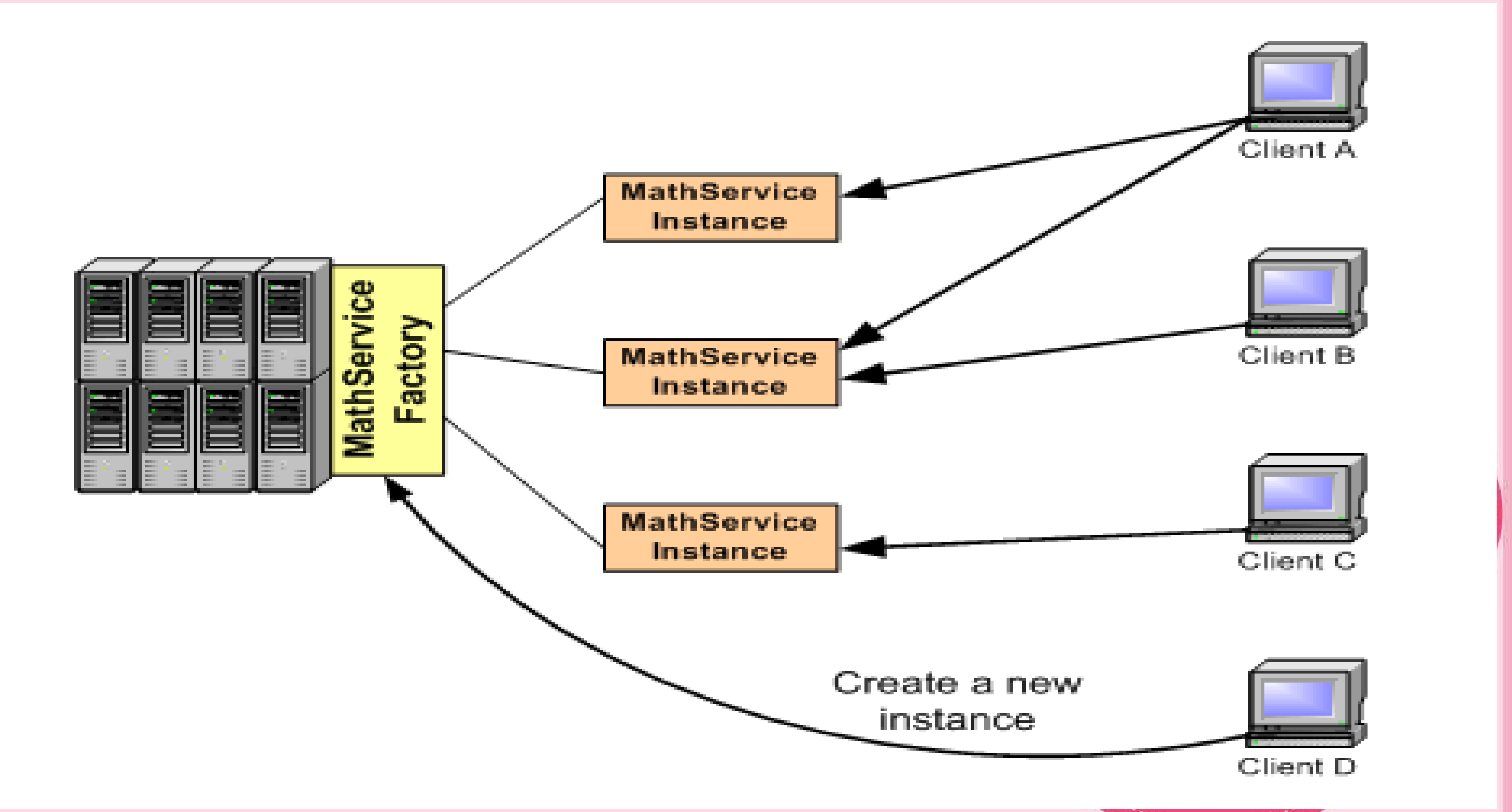
Services in the Web and the Grid

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

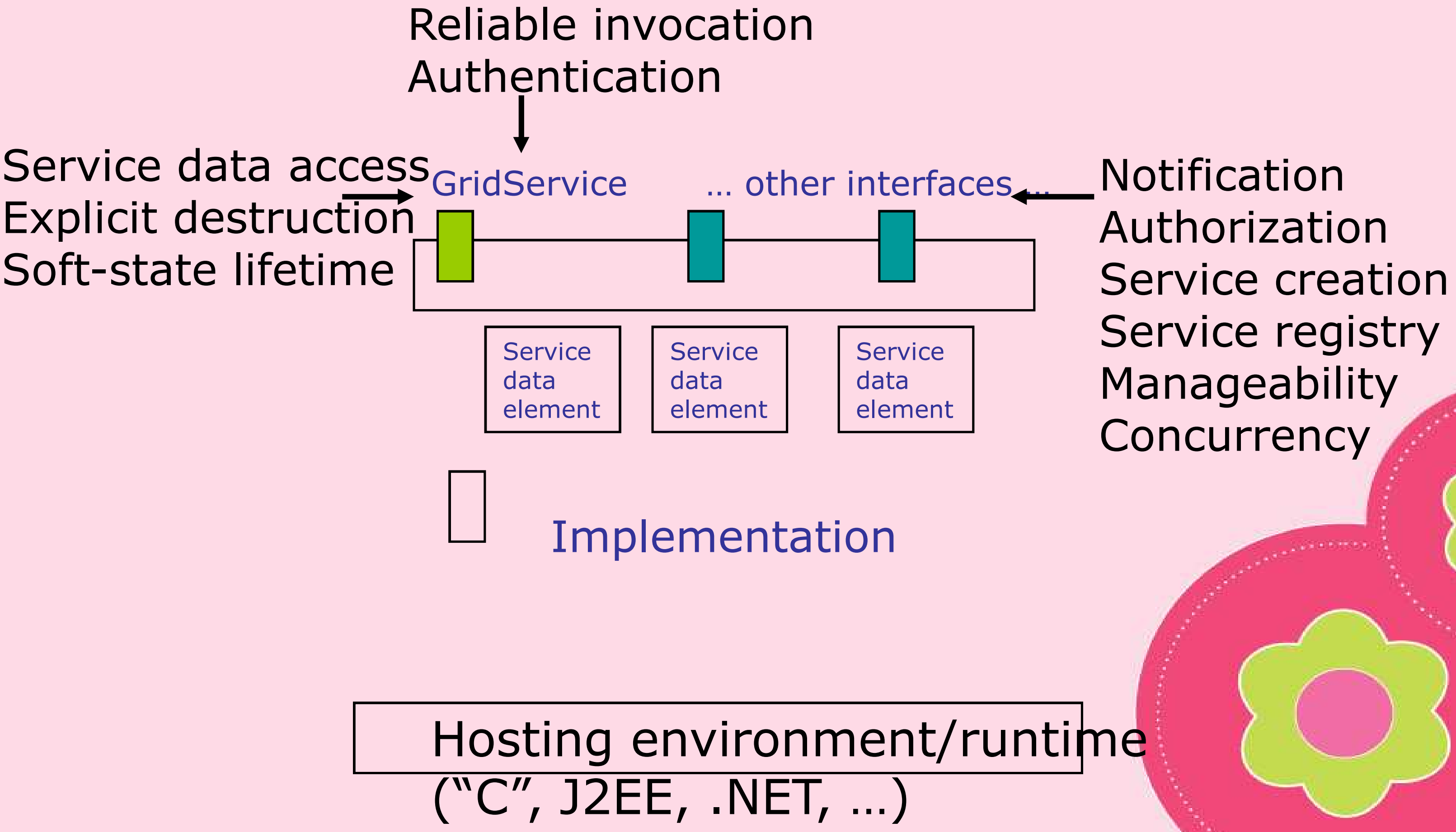
- GSH: Grid Service Handle (URI)
  - Unique
  - Shows the location of the service
- GSR: Grid Service Reference
  - Describes how to communicate with the service
  - As WS use SOAP, our GSR is a WSDL file.

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



Reliable invocation
Authentication

Service data access
Explicit destruction
Soft-state lifetime

Notification
Authorization
Service creation
Service registry
Manageability
Concurrency

Implementation

Hosting environment/runtime
("C", J2EE, .NET, …)

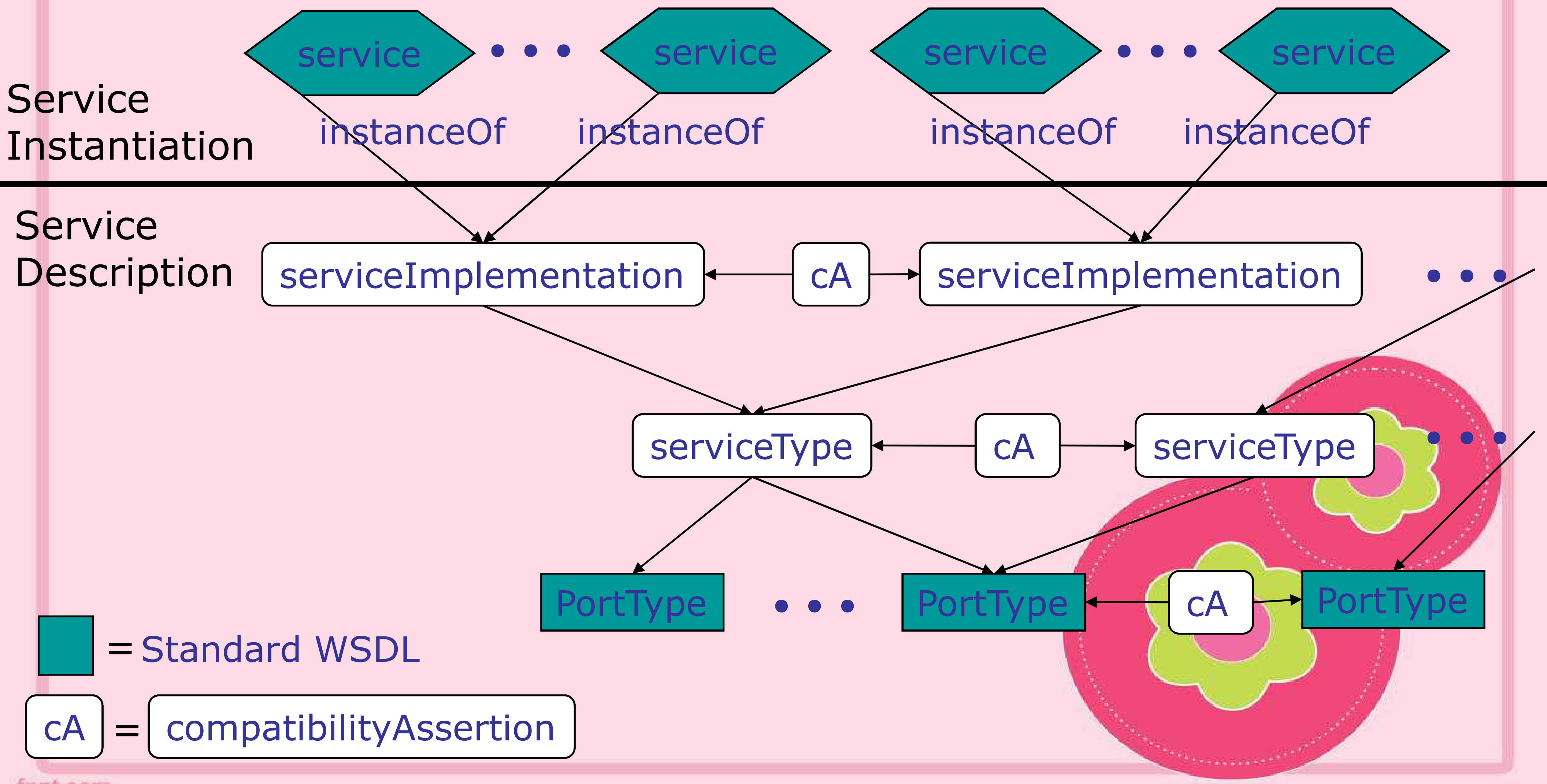
Open Grid Services Architecture: Fundamental Structure

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

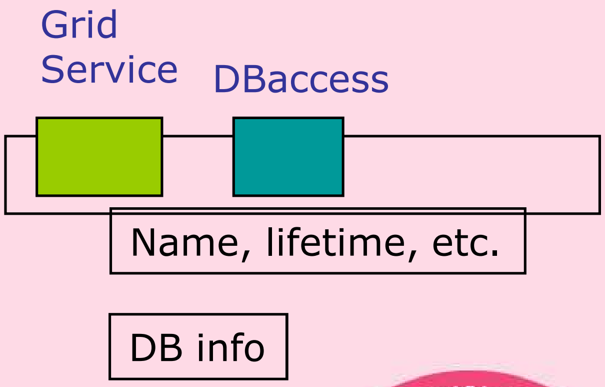
Authentication, reliability are binding properties
Manageability, concurrency, etc., to be defined

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



Grid Service

DBaccess

Name, lifetime, etc.

DB info

- 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

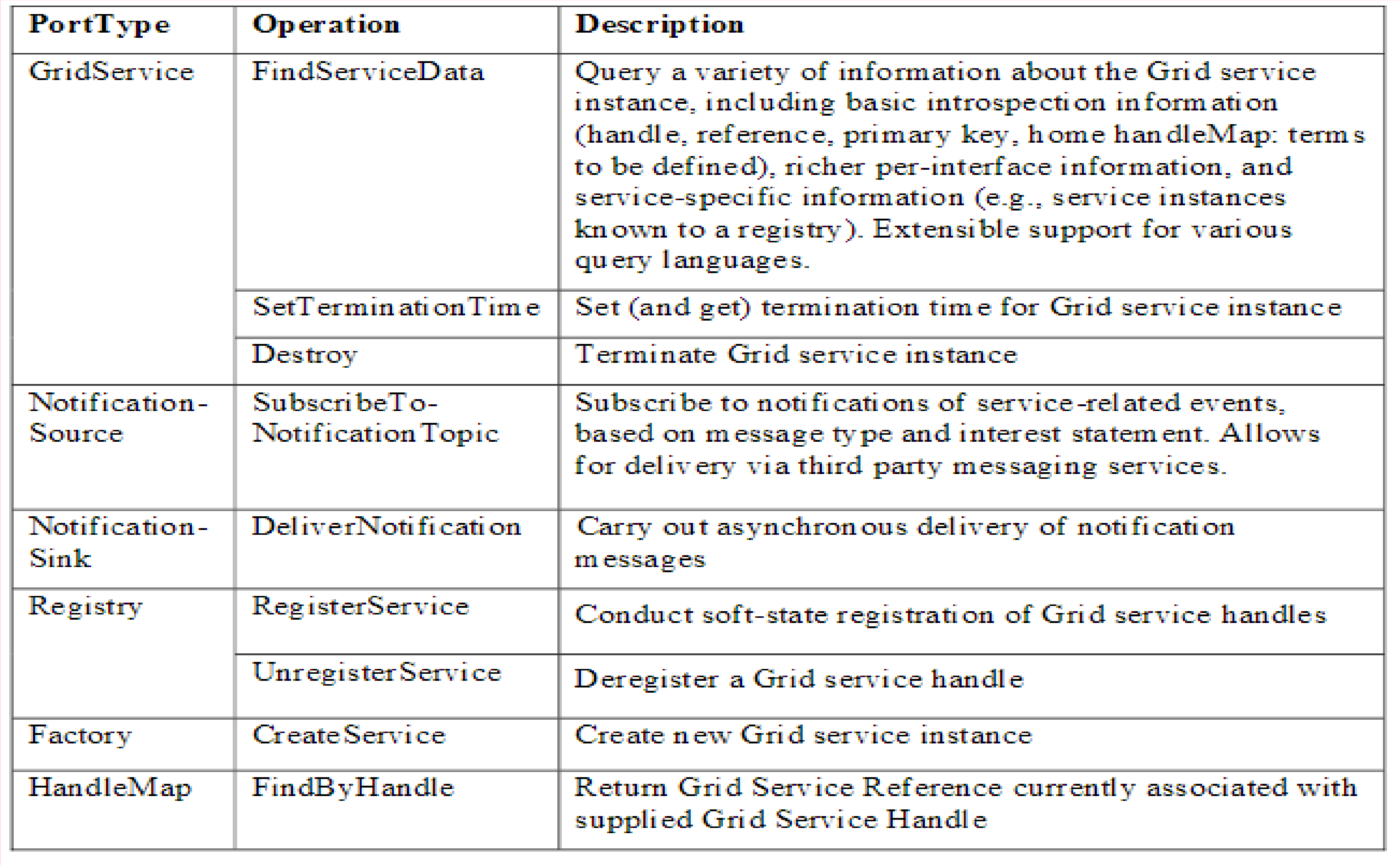
Lifetime Management

- 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

OGSA Interfaces

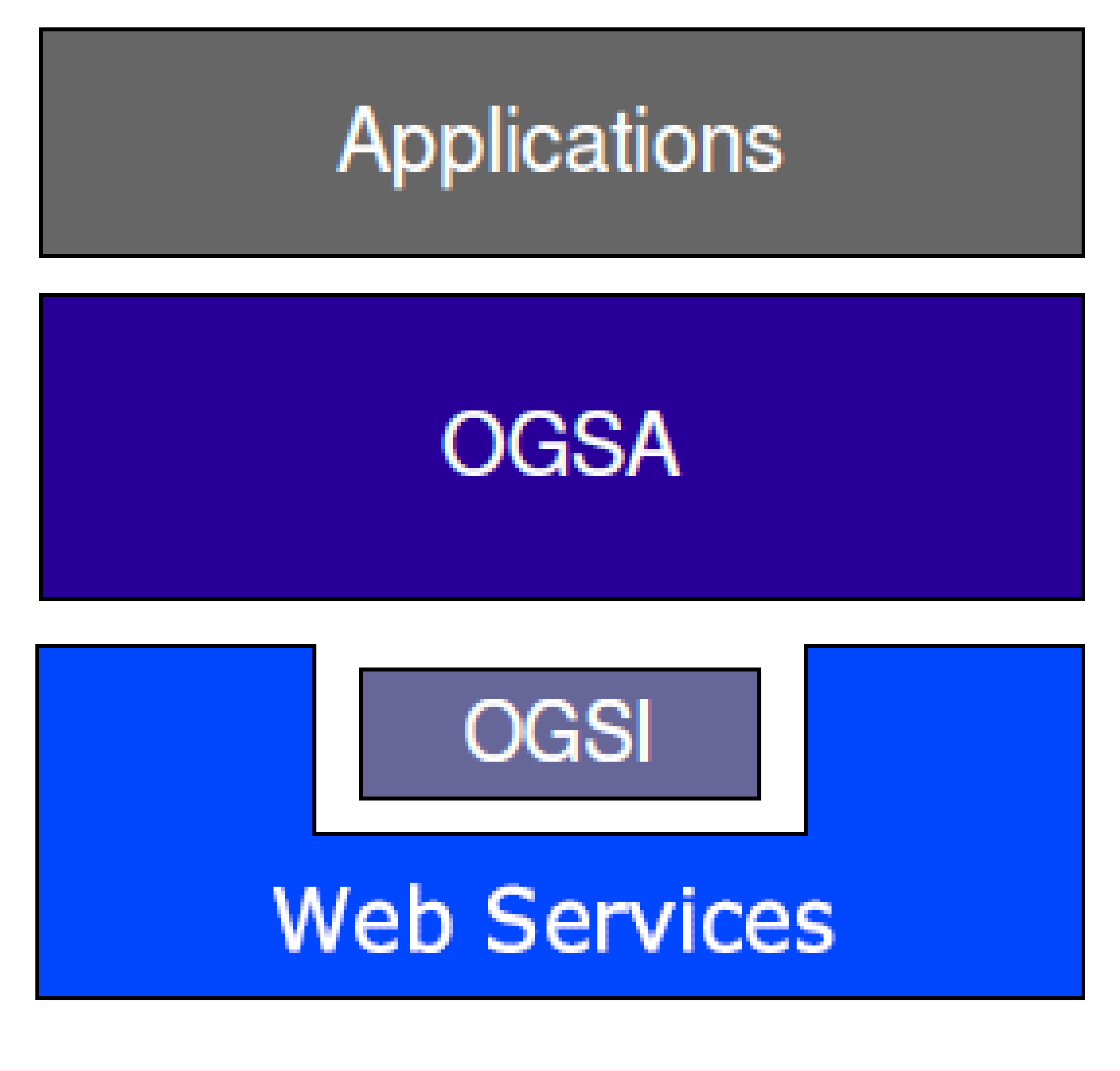


PortType	Operation	Description
GridService	FindServiceData	Query a variety of information about the Grid service instance, including basic introspection information (handle, reference, primary key, home handleMap: terms to be defined), richer per-interface information, and service-specific information (e.g., service instances known to a registry). Extensible support for various query languages.
	SetTerminationTime	Set (and get) termination time for Grid service instance
	Destroy	Terminate Grid service instance
Notification-Source	SubscribeTo-NotificationTopic	Subscribe to notifications of service-related events, based on message type and interest statement. Allows for delivery via third party messaging services.
Notification-Sink	DeliverNotification	Carry out asynchronous delivery of notification messages
Registry	RegisterService	Conduct soft-state registration of Grid service handles
	UnregisterService	Deregister a Grid service handle
Factory	CreateService	Create new Grid service instance
HandleMap	FindByHandle	Return Grid Service Reference currently associated with supplied Grid Service Handle

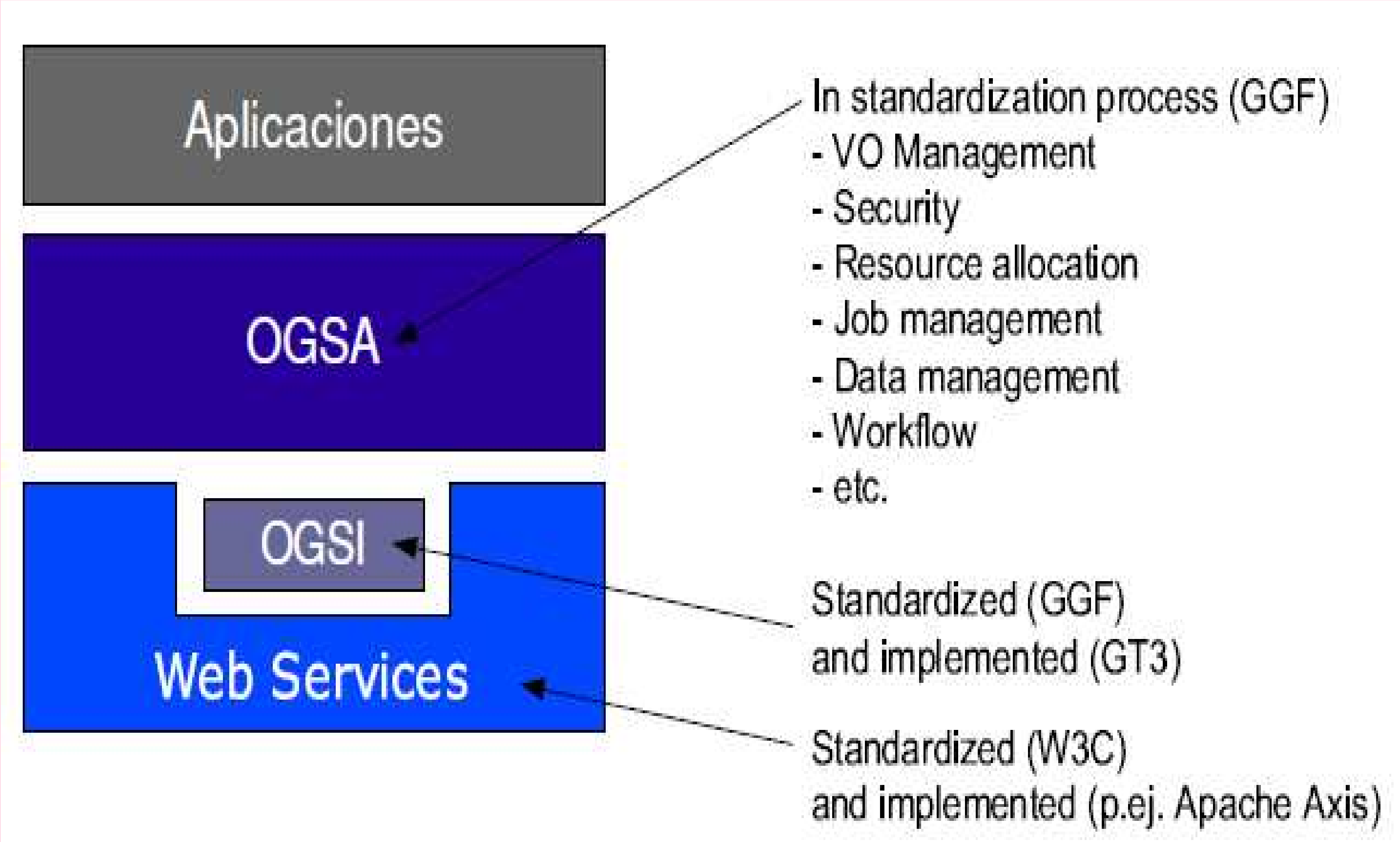
Example Services

- Storage service
  - provide ops for storing and retrieving data, reserving space.
- Data transfer service
  - provide ops for requesting the transfer of data from one storage device to another.
- Troubleshooting service
  - monitor the status of various other services.
- Have common behaviors as well
  - Monitor status
  - Query and define access policy

Grid Service



OGSA, OGSI and GT3



In standardization process (GGF)
- VO Management
- Security
- Resource allocation
- Job management
- Data management
- Workflow
- etc.

Standardized (GGF)
and implemented (GT3)

Standardized (W3C)
and implemented (p.ej. Apache Axis)

Thank You

Questions and Comments?