

This product includes material developed
by the Globus Project (<http://www.globus.org/>)



Enabling Grids for
E-science in Europe

www.eu-egee.org

Introduction to Grid Services and GT3



Grid Services and GT3

- **Outline**
- What is a **Grid Service** ?
- How to build a **Grid Service** ?
- How to access a **Grid Service** ?

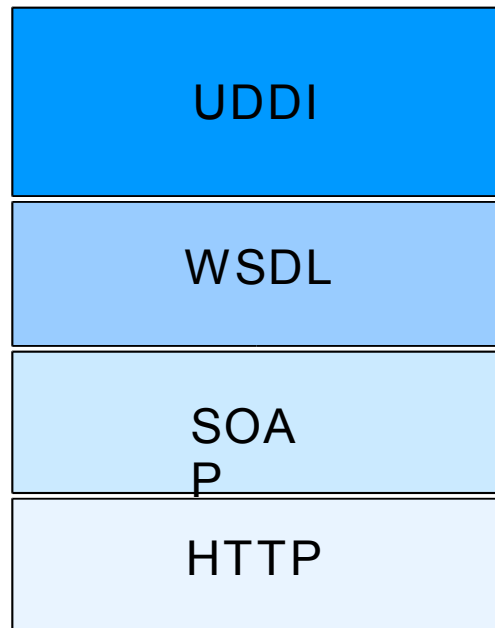
A **Grid Service** is a **Web Service** which conforms to a set of conventions (interfaces and behaviours) that define how a client interacts with a Grid service.

(Foster, Kesselman, Tuecke, et al.)

The basis for **Grid Services** are **Web Services** - a network accessible functions that can be invoked via a well-defined remote interface.

- Web Services enable service-oriented architectures:
 - internet-based applications with loosely-coupled **clients** and **servers**
 - the most common implementation of web services works in a simple *Request-Response* paradigm using the HTTP for transmitting messages

- **Web Services Architecture**
- Web Services are layered on the top of standard transfer protocols for transmitting messages:



Service Discovery

(Universal Description, Discovery and Integration)

Service Description

(Web Services Description Language)

Service Invocation

(Simple Object Basic Protocol)

Transport

(over TCP/IP)

- **Grid Services** - standard Web Services with improved characteristics.
- Grid Services are defined by the Open Grid Service Architecture (**OGSA**).
- Grid Services are specified by the Open Grid Services Infrastructure (**OGSI**).
- Globus Toolkit 3 (**GT3**) is an implementation of everything what is specified in OGSI, and therefore everything what is defined in OGSA.

Open Grid Services Architecture (OGSA)



- The **OGSA** defines a common, standard, and open architecture - a **distributed system framework** based on OGSI - for grid-based applications.
- The **Goal** - standardize services for grid applications by specifying a set of interfaces and conventions for these services.
- A Grid Service Interface corresponds to portType in WSDL.

Open Grid Services Infrastructure (OGSI)

- The **OGSI** defines mechanisms for creating, managing, and exchanging information among entities – **Grid Services**.
- The **OGSI** gives a formal and technical specification of what a Grid Service is and how it works.

Open Grid Services Infrastructure (OGSI)

- **OGSI Improvements :**
 - ◆ Stateful and potentially transient using a factory/instance model
 - ◆ **GSH & GSR**
allow to refer to Grid Service instances
Grid Service Handle – URI
Grid Service Reference – WSDL file
 - ◆ **Lifetime management**
instances can be created and destroyed at any time

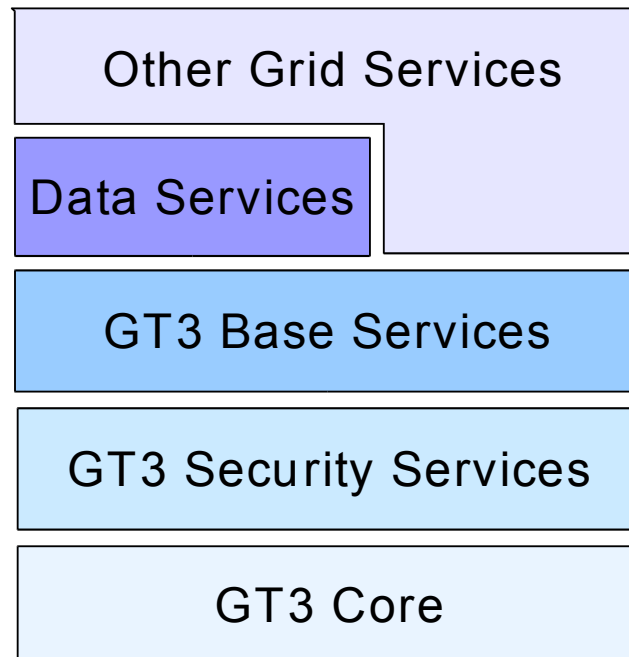
Open Grid Services Infrastructure (OGSI)



- **OGSI Improvements :**
 - ◆ **Service Data**
allows to include data - State Information, Service Metadata
 - ◆ **Notification**
changes are notified to all subscribers
 - ◆ **Service Groups**
allows to group different services together

The **Globus Toolkit 3 (GT3)** is a software toolkit developed by the [Globus Alliance](#) that allows to program grid-based applications.

- The GT3 includes the complete implementation of OGSI and a lot of other services, programs, and utilities.



Globus Toolkit® 3 Components

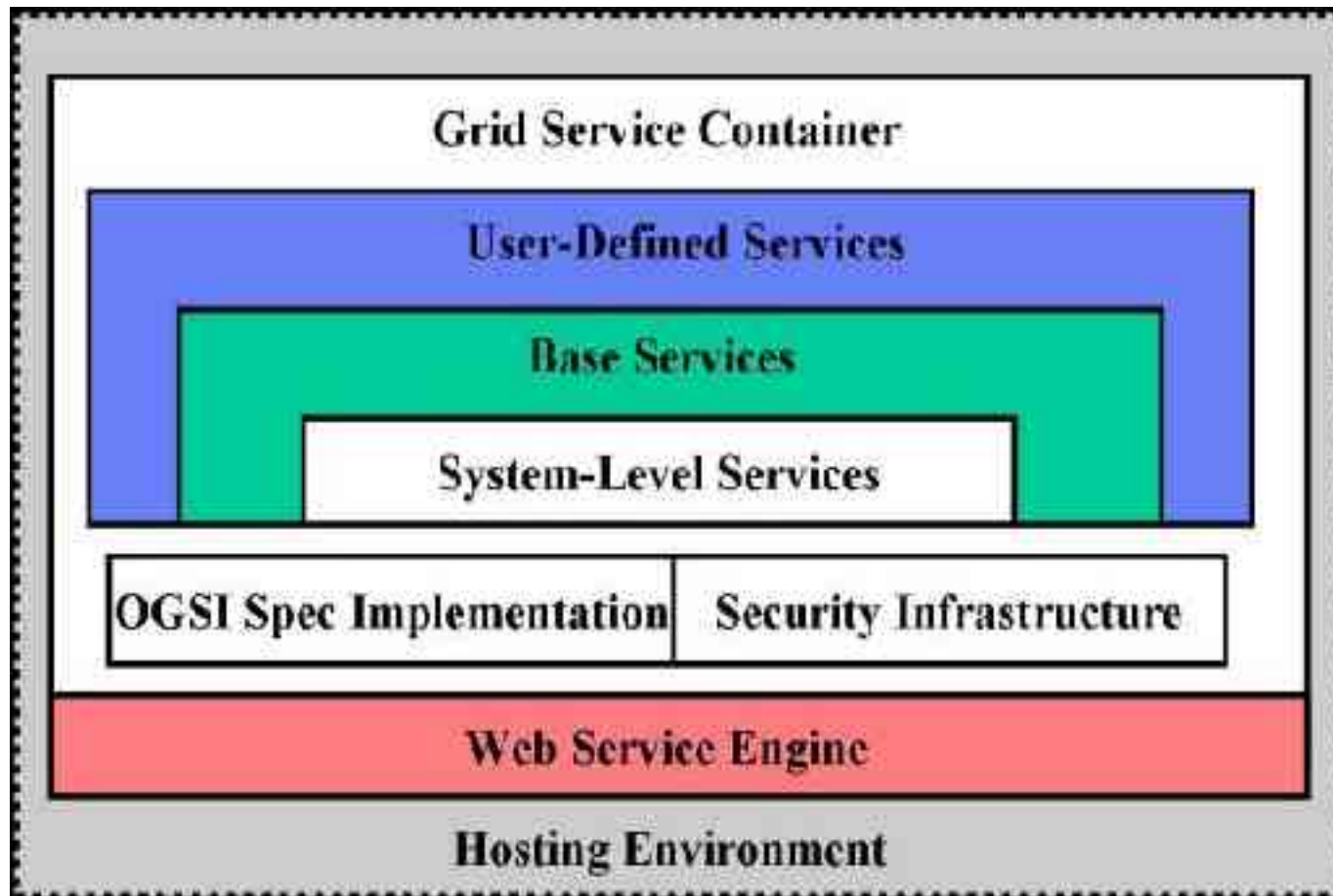


- **GT3 Core**
 - OGSI implementation
- **GT3 Security Services**
 - Several layers of security (SSL, X.509 digital certificates,...)
- **GT3 Base Services**
 - Managed Job Service: allows to control the job
 - Index Service: allows to find the location of the Grid Service
 - Reliable File Transfer Service: allows to transfer data files reliable

Globus Toolkit® 3 Components

- **GT3 Data Services**
 - includes Replica Management
- **Other Grid Services**
 - non-GT3 services running on top of the GT3 architecture

- **GT3 Architecture**



Writing and Deploying Grid Services

1. Define the service's interface (in WSDL)
2. Implement the service (in Java)
3. Deploy the service
 - **Configure** the service with deployment descriptor (in WSDD)
 - **Build** the service - compile everything and generate GAR file with Ant
 - **Deploy** the service - deploy the GAR-package into a Grid service hosting environment

Writing and Deploying Grid Services

- **1. Defining the Interface**
- **Description of the service in WSDL**
 - Writing the WSDL directly
 - Generating the WSDL from a Java interface
- **Interface** - extension of the GridService portType

- **2. Implementing the Service**
- **Implementation of a Grid Service**
 - Java class – extension of the skeleton class GridServiceImpl which implements the GridService interface
 - Delegation model – incoming calls are delegated to operation providers

- **3a. Deployment Descriptor**
- Defines how to present the Grid Service to the outer world
 - WSDD format
 - Service name
 - Parameters:
names of interface and implementation files, location,
parameters describing the characteristics

Writing and Deploying Grid Services

- **3b. Building the Service**
- **Creating the Grid Archive (GAR) file** – all files and information need to make the service available
 - **Ant tool:**
 - Create the stubs classes from the WSDL
 - Compile the stubs classes
 - Compile the service implementation
 - Organize all the files into a very specific directory structure

- **3c. Deployment of the Service into a grid service container**
- **Deployment is done with the Ant tool**
 - Unpack the GAR file and copy the files into key locations in the GT3 directory tree
 - Read the deployment descriptor and configure the web server to make the new Grid Service visible

- **Access to a Grid Service**
 - In OGSI: WSDL is used to describe multiple protocol bindings, encoding styles, messaging styles, etc. for a given Web service.
 - Web Service Invocation Framework (WSIF), Java API for XML RPC, ...

- **Access to a Grid Service**
 - Factory/Instance model
 - **Grid Service Handle (GSH)** – a valid URI, a permanent network pointer to a particular Grid service instance
 - **Grid Service Reference (GSR)** – network pointer to a specific Grid service instance - must be **resolved** from GSH (HandleResolver)

- **Service Data** is a structured collection of information associated to a Grid Service.
- Generally, Service Data falls into two categories:
 - ◆ **State Information** (operation results, runtime information, etc)
 - ◆ **Service Metadata** (system data, supported interfaces, cost of the service, etc.)
- Service Data is an essential part of service discovery, and closely related to the notifications.

- Any Grid Service can have attached any number of **Service Data Elements (SDE)**.
- Each SDE must have a unique name and can contain structured data.
- The structure of the datatype is defined in the WSDL file.

- **SDE Access**
- **Fundamental datatypes** - directly by accessing the value of the SDE
- **Complex datatypes** - through an SDE-Java Bean generated from the WSDL (XML) description. It allows to access the individual fields of the SDE.

- **The most interesting GridService SDEs**
- **GridServiceHandle** – GSHs
- **FactoryLocator** – locator for the factory which created this Grid Service
- **TerminationTime** – info about termination time of the Grid service
- **ServiceDataNames** – all the SDEs in the Grid Service
- **Interfaces** – names of all interfaces implemented by this Grid Service

- **Notification** in GT3 – a core feature closely related to service data; it allows clients to be notified of changes occurring in a Grid Service.
- A **Pull** approach – the observable only informs the observers that a change has occurred; to get more information another call to the observable must be done
 - Each observer needs to get different information
 - Observer can ignore the notification
- A **Push** approach - allows data to travel with the notification
 - Each observer needs to get the same information

Lifetime Management

- In the factory/instance model – Grid Services are potentially transient.
- Grid Service instances are created with a specified lifetime (min and max).
- The initial lifetime can be extended by explicit request of the client.

