GT3 Toolkit Core: Grid Service Container Framework

1. Based on OGSI primitives and protocols
2. Built on top of WS technology
3. Some features: introspection, soft-state management (lease based server side state), notification, discovery, logging, security, management and administration of services, service data
4. Development environment, auto generation of components
5. Runtime environment:
   OGSA: Open Grid Service Architecture (Components)
   OGSI: Open Grid Services Infrastructure (specification) (interfaces)
   GT3: Open source implementation of OGSI
   Opportunity: other implementations of OGSI

6. A grid service is a service compliant with OGSI and which exposes itself through Web Services Description Language (WSDL) interface.
7. GS = WS + OGSI-compliant

8. GT3 Core Architecture:
   Hosting environment
   Grid service container
   WS engine
   OGSI Ref Implementation Security Infrastructure
   System level services: Remote procedure call, etc.
   Base services: Program execution, data management, information services
   User-defined services

9. Container:
   All the services and primitives interact with run-time environment called the “container”
   Container shields the application from specific run-time settings, deployment details (we know about it from J2EE)
   Gars are deployable units. Can be deployed on a container on mills or on cerf or on the trailer one!
   Opportunity: EJB container extension to host grid services?

10. OGSI primitives: standard interfaces for GridService, Factory, Notification (source and Sink), HandleResolver etc.
11. Security infrastructure: Transport-level and message-level security; Core security is Java Authentication and Authorization Based (JAAS). For transport level: httpg: protocol based on GSI is provided.
12. Virtual Hosting Environment Framework: A grid service is typically hosted in the same container as its factory. It is possible to distribute grid services to a number of remote containers: to satisfy a QoS requirement; migration should be transparent to clients. Architecture should simplify firewall traversal. Support provided by routing handlers, redirector, reference and handler rewriters,
redirection exception API; Sample scenario is provided in the paper.

**Opportunity:** Discover application models that can exploit /thrive on this Virtual Hosting concept.

13. Programming Model: two alternatives; Grid Service Base or Operation providers. Implementation vs delegation model; lets go through Figure 2; service programming model

14. Service callback and factory callback

Let's look at some code now: Developer’s Guide

SotoMayer’s Tutorial