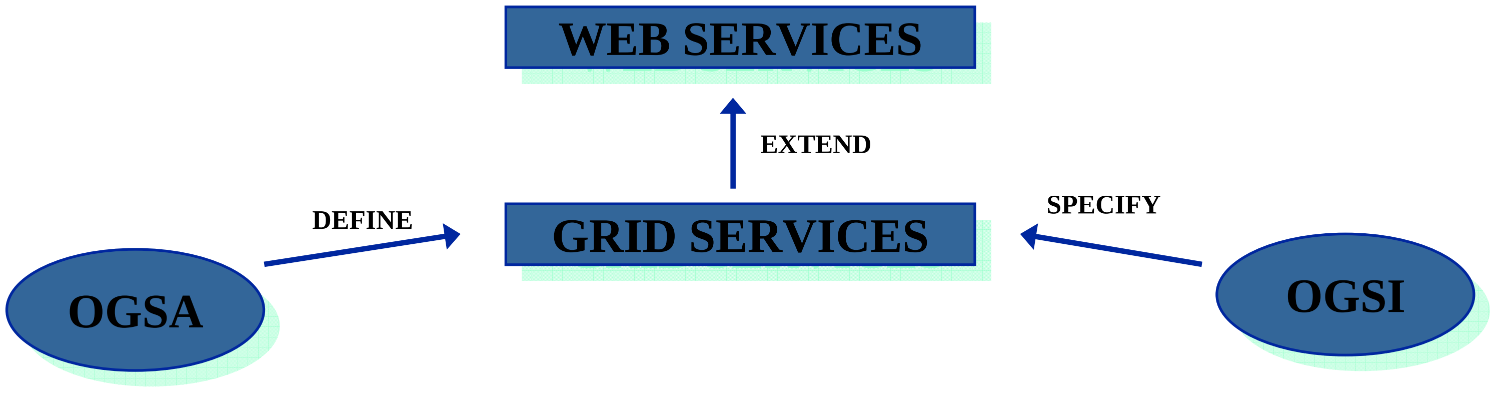
OGSI and GT3 Initial Experiences

DataGrid is a project funded by the European Commission
under contract IST-2000-25182

Contents

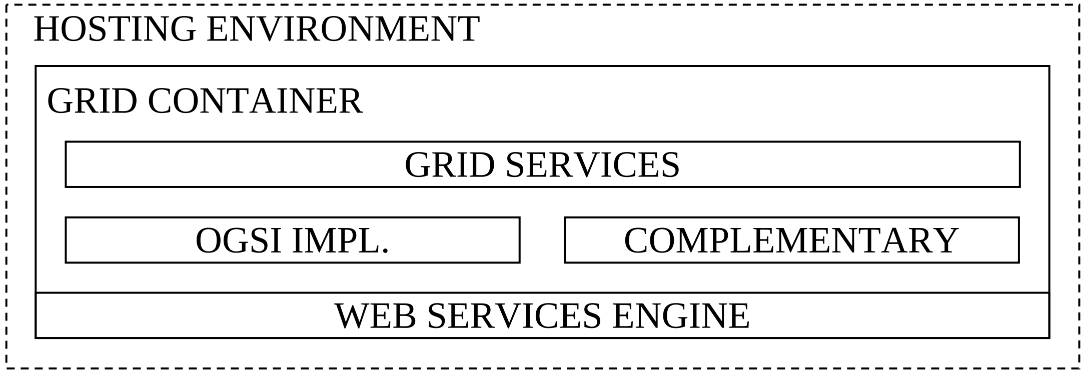
Understanding OGSA / OGSI

- Web Services
  - Provide interoperability for services interaction
  - XML, SOAP, WSDL
- Open Grid Service Architecture
  - Integrates grid technologies with Web Services
  - Defines the key components of the grid
- Open Grid Service Infrastructure
  - Formal and technical specification of the services described in OGSA
  - Defines interfaces for interaction with and between Grid Services



Grid Services

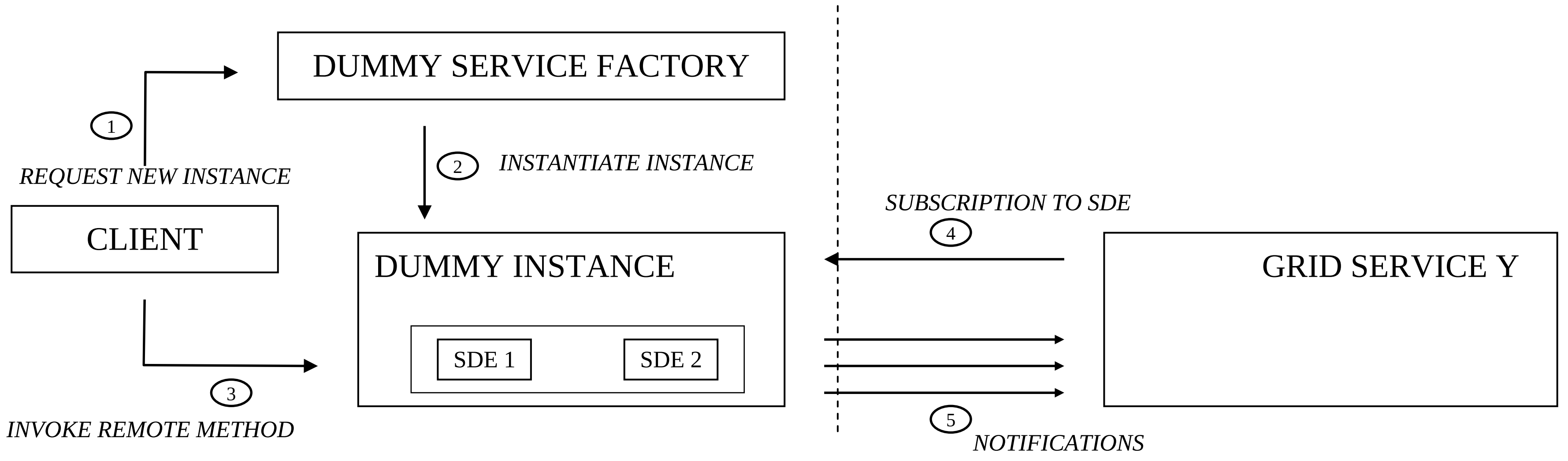
- Core Architecture



- Current options
  - Hosting Environments & OGSI Implementations:
    - J2EE Application Servers -> Globus Toolkit 3
    - Microsoft .NET Platform -> OGSI.NET, MS.NETGrid
  - Others will appear:
    - Any environment with a Web Services engine available is one step away from providing Grid Services
    - Some efforts started: OGSI::Lite (Perl), pyGridWare (Python)

Grid Services

- Web Service Extensions
  - Service Data
    - Each Grid Service has a set of Service Data Elements
  - Interface Inheritance
  - Lifecycle Management
- Services may be persistent or transient
- Dummy Service example to demonstrate:
  - Factories and Instances
  - Notifications



Globus Toolkit 3

- Provides complete implementation of the OGSI specification
- Complementary Services: Security
  - GSI3 – extension of GSI2 providing:
    - Message-Level Security: XML Signature and XML Encryption
    - Transport-Level Security – deprecated
- Higher Level Services
  - Resource Management
    - MMJFS – GT2 GRAM with a Grid Service interface
  - Data Management
    - GT2 GridFTP
    - Reliable File Transfer Service – RFT
    - Replica Location Service – RLS
  - Information Services
    - MDS3 – information aggregation and registry
    - GT2 MDS2 – ldap interface
    - Although both implementations are offered in GT3, they are independent so one knows nothing about the existence of the other

Globus Toolkit 3

- Components to deploy a Grid Service
  - Service Interface – GWSDL (WSDL Extended)
    - Manually written or generated from existing Java code
  - Service Implementation
    - Directly extending a basic Grid Service or using delegation
  - Deployment Descriptor
    - Defined in WSDD (Web Service Deployment Descriptor)
  - Build File
    - Used by the Apache Ant tool for compilation (and deployment)

Performance Measurements

- Hardware: 2 * Pentium III 600mhz processors, 256mb RAM
- Dummy Service Performance
  - Setup 1
    - Client calls Dummy Service Factory to create a new Instance
    - Client invokes two methods on the instance created
    - Client destroy the instance
  - Setup 2
    - Same as setup one but in the second step each method is invoked 100 times
  - Up to 1000 clients were used from up to 45 different client nodes
  - Security was tried using instructions from the Globus page
    - GSI Secure Conversation with XML Signature

Performance Measurements

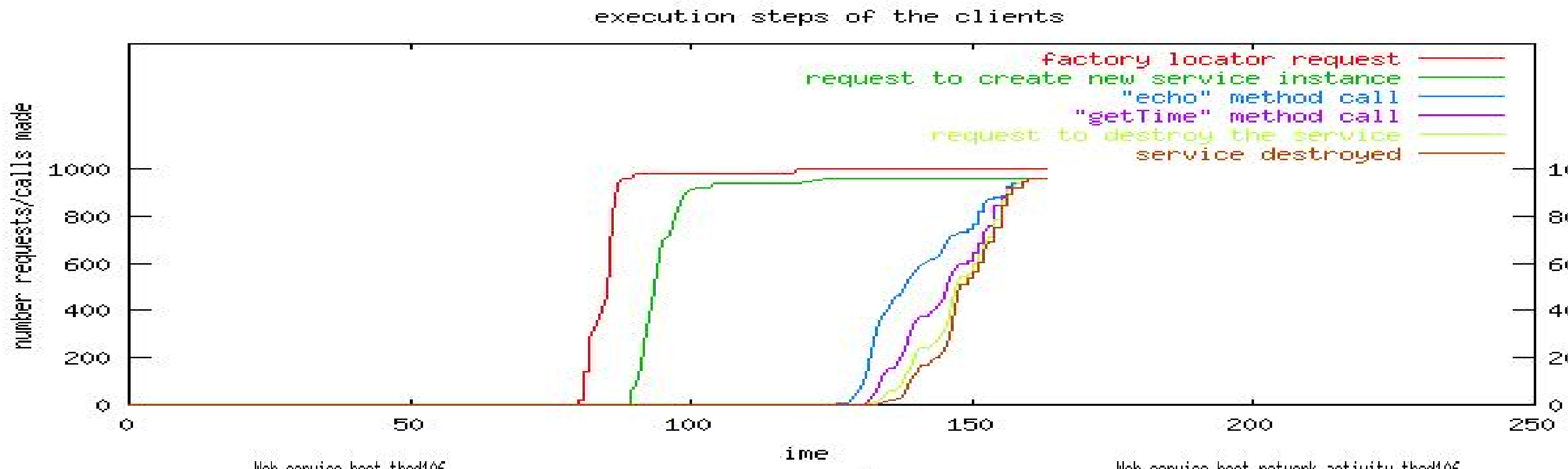
- Dummy Service Performance
  - Preliminary Results

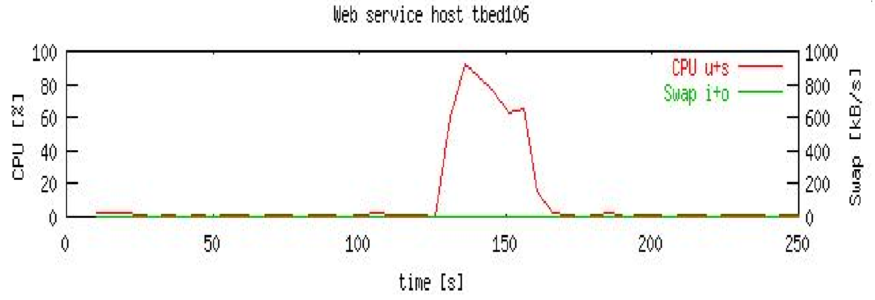
setup	authentication	service container	saturation throughput	average CPU u+s usage, %
1	-	GT3 standalone	41 services/s	89
1	yes	GT3 standalone	1.3 services/s	88
1	-	Tomcat	60 services/s	89
1	yes	Tomcat	1.2 services/s	88
2	-	GT3 standalone	300 method calls/s	96
2	yes	GT3 standalone	10 method calls/s	72
2	-	Tomcat	290 method calls/s	96
2	Yes	Tomcat	13 method calls/s	79

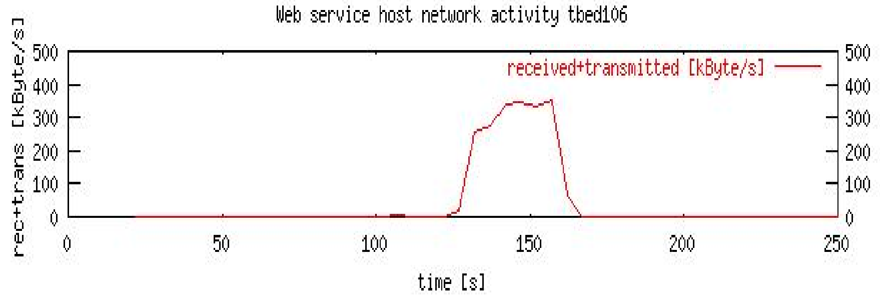
- Conclusions:
  - Security overhead needs further investigation
  - Tomcat can be from a bit slower to 50% faster
  - Some additional tests on more powerful machines should be done

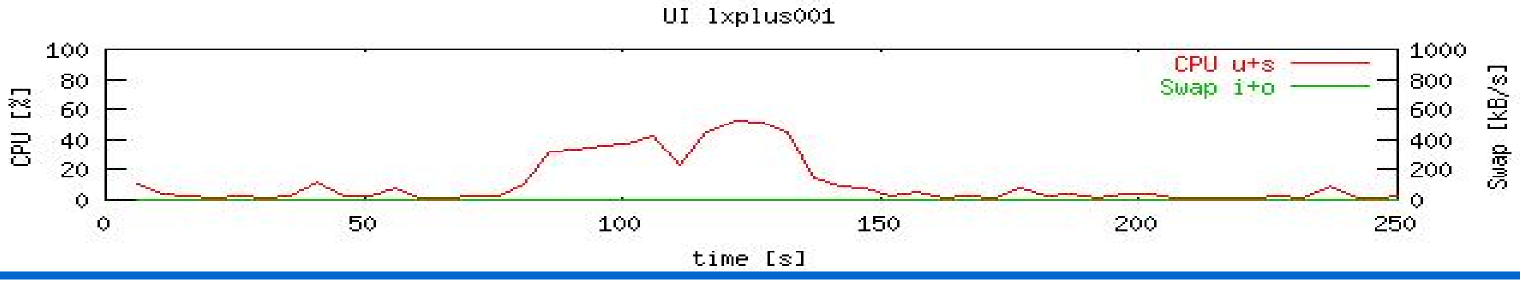
Performance Measurements

◆ Dummy Service Performance









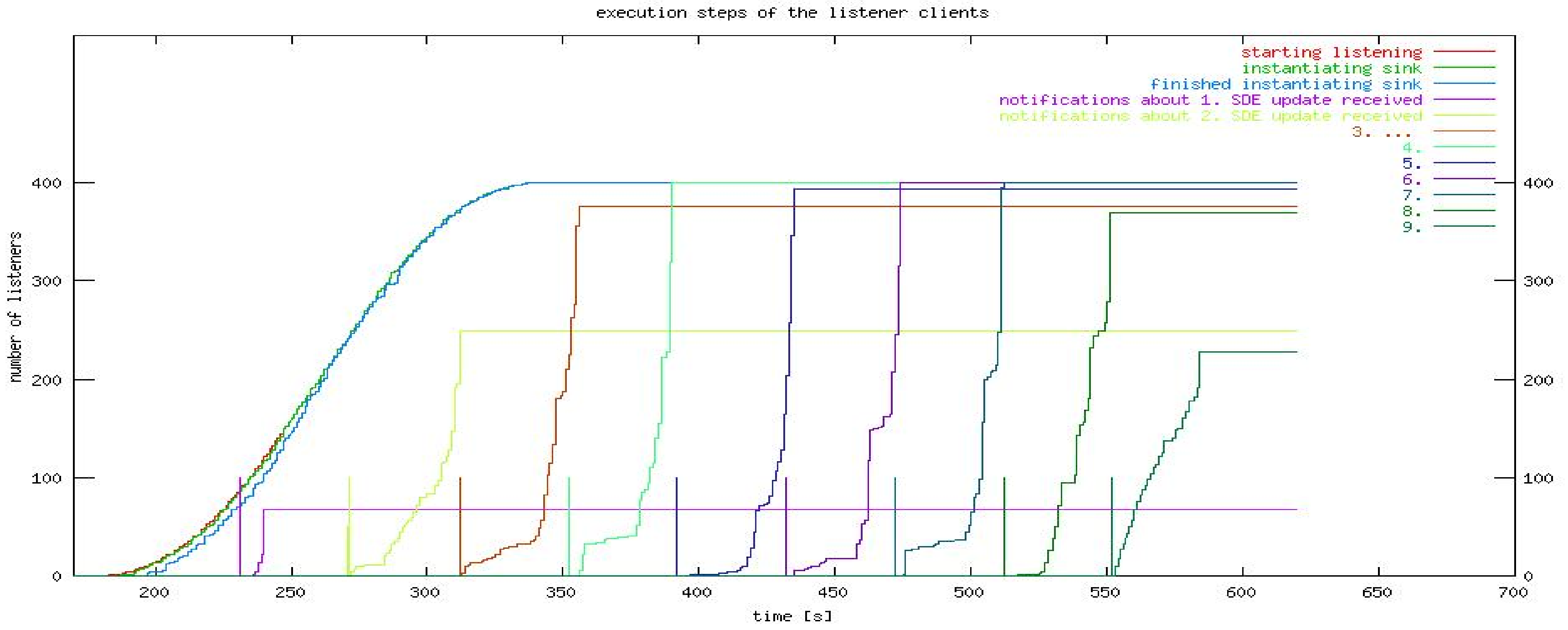
Performance Measurements

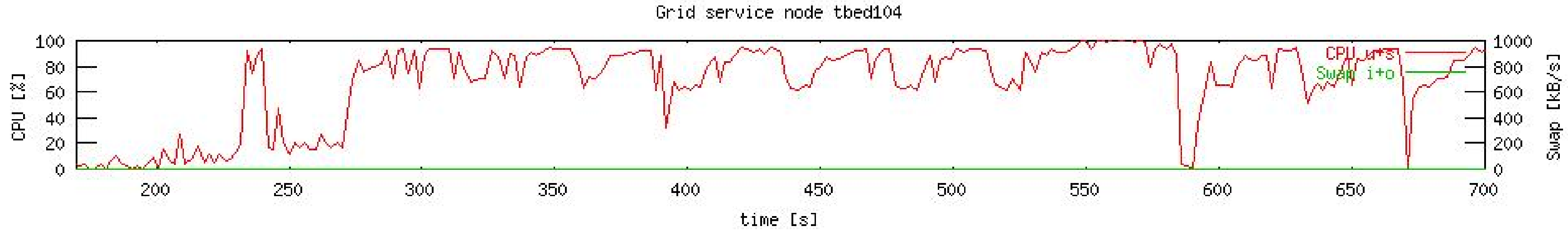
- Index Service Performance
  - Setup 1 (push)
    - Notification mechanism is used
    - Index Service acts as a Notification Source
    - Multiple Notification Sinks subscribe to the 'Host' SDE
  - Setup 2 (pull)
    - Multiple clients querying the Index Service using ogsi-find-service-data
    - Clients run in parallel
  - No security is used in either setup

setup	service container	saturation throughput	average CPU u+s usage, %
1	GT3 standalone	10-15 notifications/s	81 – 87
1	Tomcat	-	
2	GT3 standalone	200 requests/s	88
2	Tomcat	200 requests/s	90

Performance Measurements

- Index Service Performance





Conclusions

- Generally impressed with GT3 and the overall concept
- Major Issues:
  - Performance
  - Incomplete documentation
  - Several bugs found:
    - Core implementation related – due to framework short lifetime
    - From tools deployed with the framework – hard to solve (e.g. Axis)
    - From the outside – easier to solve (e.g. Tomcat)
- System architecture must consider performance issues
- Next Actions:
  - Understand better the performance issues on Security, Factories and Notifications
  - Validate the interoperability claim by trying other toolkits