Grid Computing

Globus Toolkit 4.0 – Part 1

Paul A. Farrell
Fall 2006

Based on

Ian Foster, Globus Toolkit version 4
Tutorial, 1st Intl. Conf. on e-Science and Grid Computing,
Melbourne, Australia, December 12, 2005

The Globus Toolkit is a Collection of Components

- A set of loosely-coupled components, with:
  - Services and clients
  - Libraries
  - Development tools
- GT components are used to build Grid-based applications and services
  - GT can be viewed as a Grid SDK
- GT components can be categorized across two different dimensions
  - By broad domain area
  - By protocol support

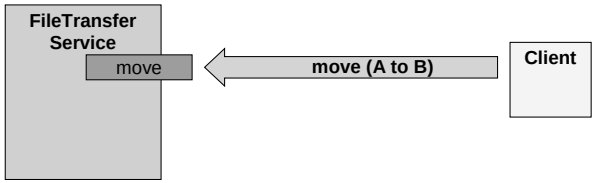
GT Domain Areas

- Core runtime
  - Infrastructure for building new services
- Security
  - Apply uniform policy across distinct systems
- Execution management
  - Provision, deploy, & manage services
- Data management
  - Discover, transfer, & access large data
- Monitoring
  - Discover & monitor dynamic services

GT Protocols

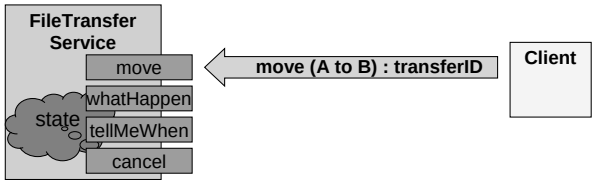
- Web service protocols
  - WSDL, SOAP
  - WS Addressing, WSRF, WSN
  - WS Security, SAML, XACML
  - WS-Interoperability profile
- Non Web service protocols
  - Standards-based, such as GridFTP
  - Custom

“Stateless” vs. “Stateful” Services



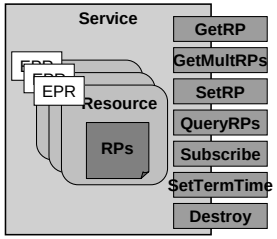
- Without state, how does client:
  - Determine what happened (success/failure)?
  - Find out how many files completed?
  - Receive updates when interesting events arise?
  - Terminate a request?
- Few useful services are truly “stateless”, but WS interfaces alone do not provide built-in support for state

FileTransferService (without WSRF)



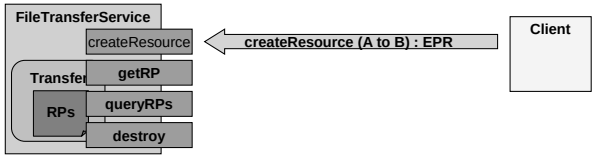
- Developer reinvents wheel for each new service
  - Custom management and identification of state: **transferID**
  - Custom operations to inspect state synchronously (**whatHappen**) and asynchronously (**tellMeWhen**)
  - Custom lifetime operation (**cancel**)

WSRF in a Nutshell



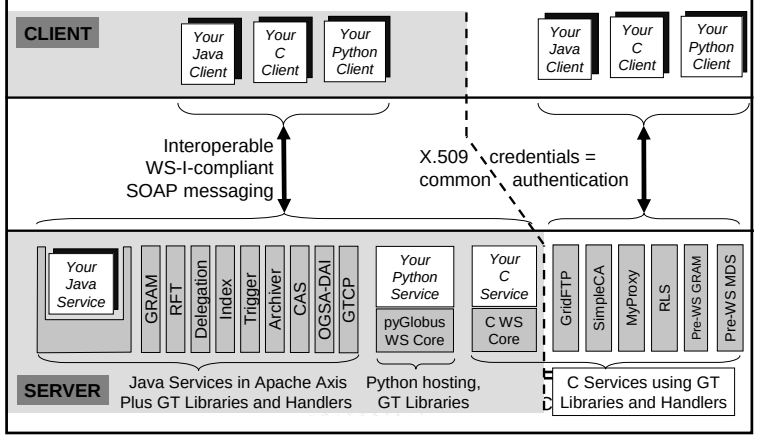
- Service
- State representation
  - Resource
  - Resource Property
- State identification
  - Endpoint Reference
- State Interfaces
  - GetRP, QueryRPs, GetMultipleRPs, SetRP
- Lifetime Interfaces
  - SetTerminationTime
  - ImmediateDestruction
- Notification Interfaces
  - Subscribe
  - Notify
- ServiceGroups

FileTransferService (w/ WSRF)



- Developer specifies custom method to createResource and leaves the rest to WSRF standards:
  - State exposed as Resource + Resource Properties and identified by Endpoint Reference (EPR)
  - State inspected by standard interfaces (GetRP, QueryRPs)
  - Lifetime management by standard interfaces (Destroy)

GT4 Components



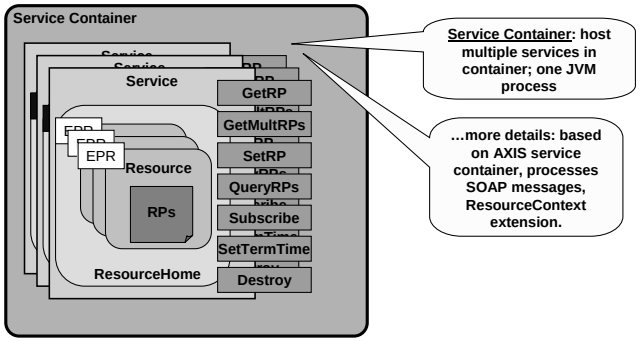
Goals for GT4

- Usability, reliability, scalability, …
  - Web service components have quality equal or superior to pre-WS components
  - Documentation at acceptable quality level
- Consistency with latest standards (WS-*, WSRF, WS-N, etc.) and Apache platform
  - WS-I Basic Profile compliant
  - WS-I Basic Security Profile compliant
- New components, platforms, languages
  - And links to larger Globus ecosystem
- See
  - http://www.globus.org/toolkit/docs/4.0/

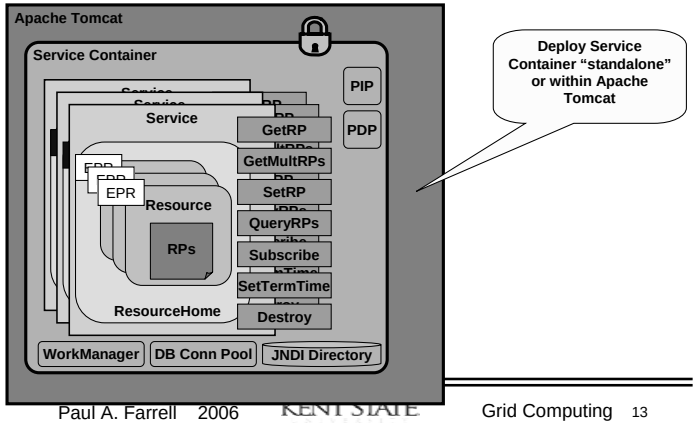
Hosting Environment

- Specific execution environment
- GT3 supports the following four Java hosting environments:
  - Embedded: embedded in any existing J2SE application.
  - Standalone: a lightweight J2SE server that hosts Grid services.
  - J2EE Web Container: an OGSI hosting environment inside a Web server that can be hosted by any Java Servlet-compliant engine, such as the Jakarta Tomcat.
  - J2EE EJB Container: a code generator to allow exposure of stateful J2EE Entity and Session Java Beans as OGSI-compliant Grid services.

GT4 WS Engine



GT4 WS Engine Hosting



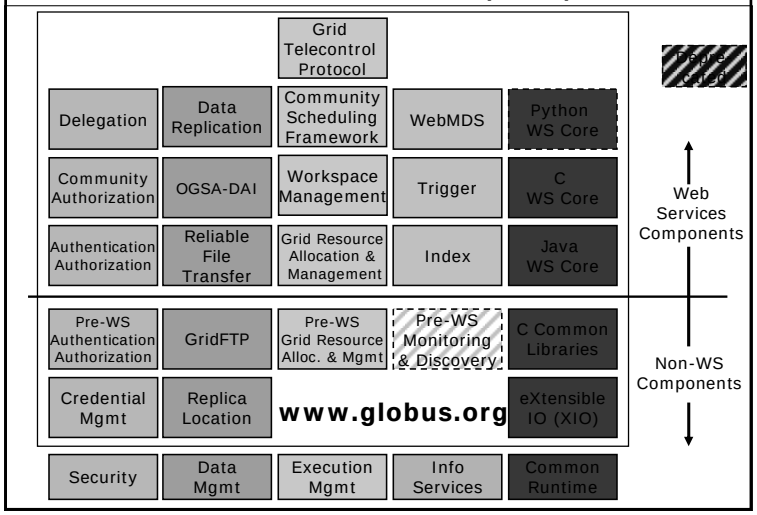
Software Required

- Required to install Globus 4.0.3
  - J2SE 1.4.2+ SDK from Sun, IBM, HP, or BEA (do not use GCJ).
  - Ant 1.6+ (1.6.1+ if using Java 1.5).
  - C compiler. If gcc, avoid version 3.2. 3.2.1 and 2.95.x are okay. gcc 4.1 has a bug
  - GNU tar, GNU sed, GNU Make
  - zlib 1.1.4+
  - Perl 5.005 or newer
  - sudo
  - JDBC compliant database. For instance, PostgreSQL 7.1+
  - gpt-3.2autotools2004 (shipped with the installers, but required if building standalone GPT bundles/packages)

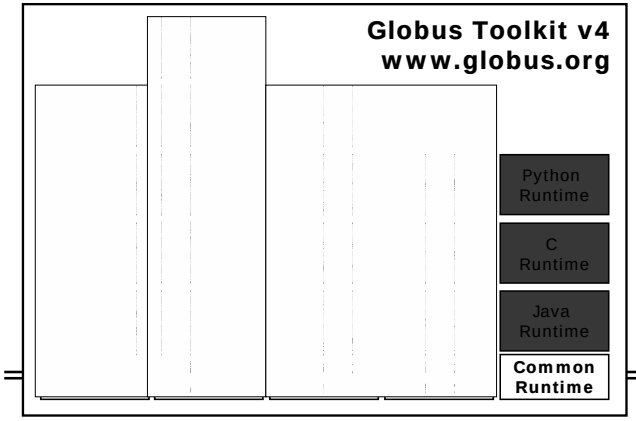
Optional Software

- IODBC (compile requirement for RLS)
- Tomcat (required by WebMDS, optional for other services
- gLite Java VOMS parsing libraries (compile requirement for Workspace Service)

Globus Toolkit version 4 (**GT4**)



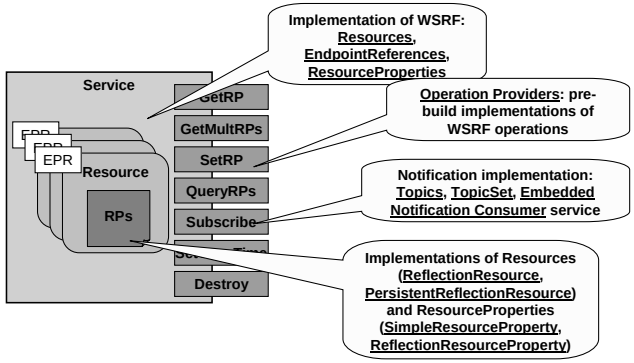
Globus Toolkit: Runtime



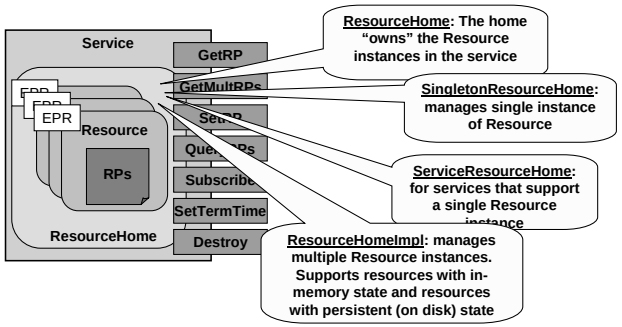
GT4 Web Services Runtime

- Supports both GT (GRAM, RFT, Delegation, etc.) & user-developed services
- Redesign to enhance scalability, modularity, performance, usability
- Leverages existing WS standards
  - WS-I Basic Profile: WSDL, SOAP, etc.
  - WS-Security, WS-Addressing
- Adds support for emerging WS standards
  - WS-Resource Framework, WS-Notification
- Java, Python, & C hosting environments
  - Java is standard Apache

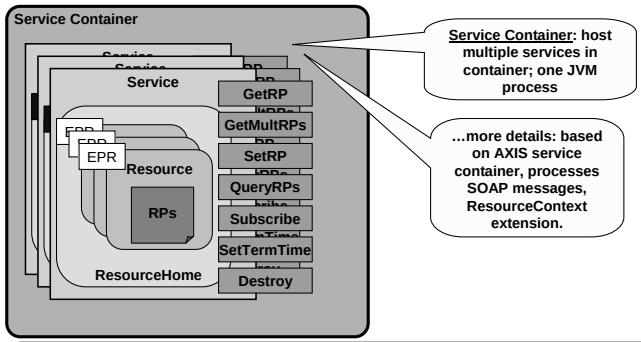
GT4 WS Core in a Nutshell



GT4 WS Core in a Nutshell

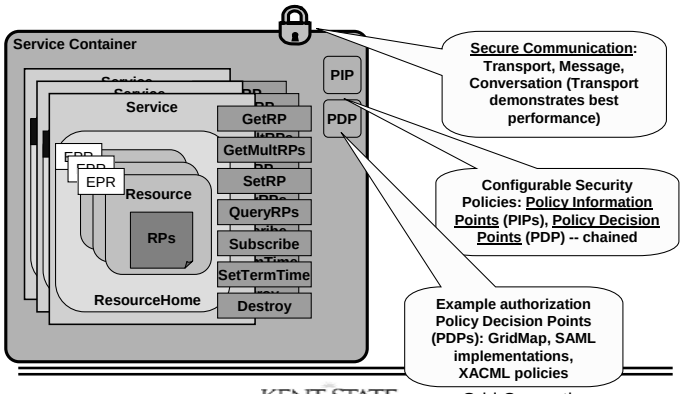


GT4 WS Core in a Nutshell



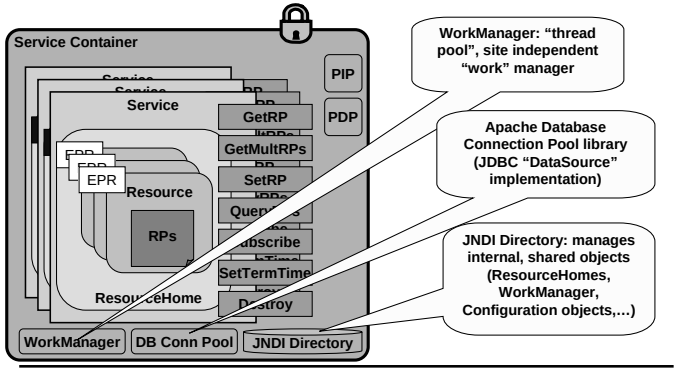
Paul A. Farrell 2006 KENT STATE Grid Computing 21

GT4 WS Core in a Nutshell



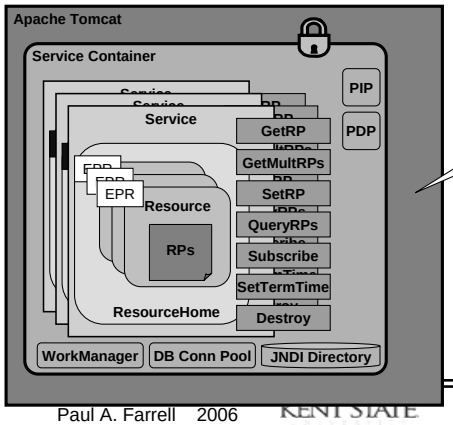
Paul A. Farrell 2006 KENT STATE Grid Computing 22

GT4 WS Core in a Nutshell



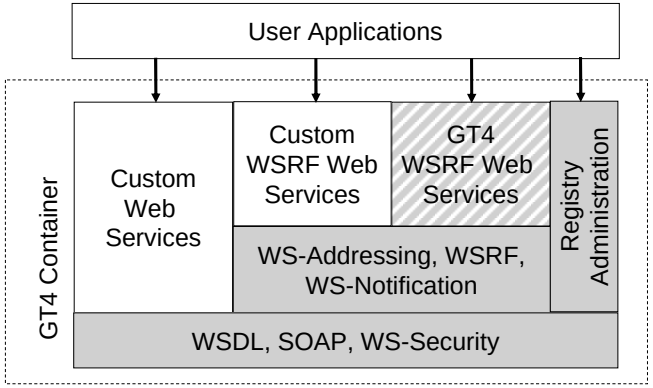
Paul A. Farrell 2006 KENT STATE Grid Computing 23

GT4 WS Core in a Nutshell



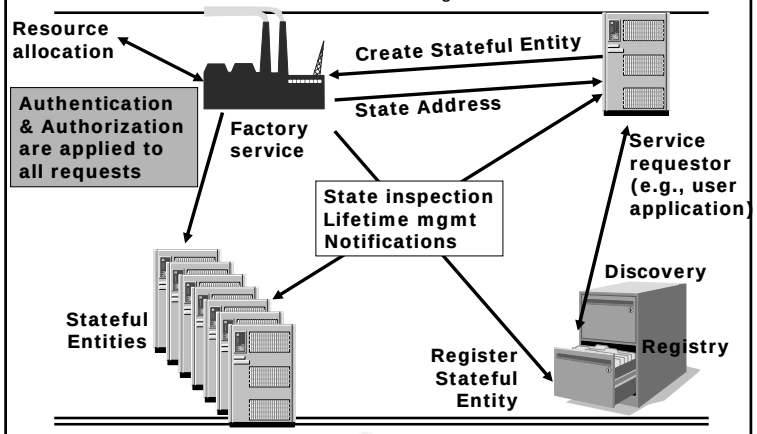
Paul A. Farrell 2006 KENT STATE Grid Computing 24

GT4 Web Services Runtime



Modeling State in Web Services

Interactions standardized using WSDL and SOAP



WSRF & WS-Notification

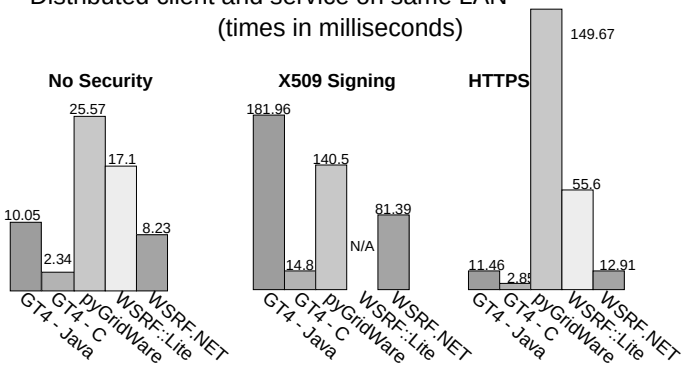
- **Naming and bindings** (basis for virtualization)
  - Every resource can be uniquely referenced, and has one or more associated services for interacting with it
- **Lifecycle** (basis for fault resilient state mgmt)
  - Resources created by services following factory pattern
  - Resources destroyed immediately or scheduled
- **Information model** (basis for monitoring, discovery)
  - Resource properties associated with resources
  - Operations for querying and setting this info
  - Asynchronous notification of changes to properties
- Service groups (basis for registries, collective svcs)
  - Group membership rules & membership management
- Base Fault type

WSRF/WSNs Compared (HPDC 2005)

	GT4-Java	GT4-C	pyGridWare	WSRF::Lite	WSRF.NET
Languages supported	Java	C	Python	Perl	C#/C++/VBasic, etc.
WS-Security password profile	Yes	No	In progress	In progress	Yes
WS-Security X.509 profile	Yes	In progress	Yes	In progress	Yes
WS-SecureConversation	Yes	No	Yes	No	Yes
TLS/SSL	Yes	Yes	Yes	Yes	Yes
Authorization	Multiple	Multiple	Callout	None	
Persistence of WS-Resources	Yes	Not default	Yes	Yes	Yes
Memory Footprint	JVM + 10M	22 KB	12 MB	12 MB	Depends
Memory size per WS-Resource	Depends on resource state	70B	Depends on resource state	0 (file/DB) or 10B (process)	Depends on resource state
Unmodified hosting environment	Yes	No	Yes	Yes (Apache)	Yes
Compliance with WS-I Basic Profile	Yes	Yes	Yes	In progress	Yes
Compliance with WS-I Basic Security Profile	Yes	Yes	Yes	No	Yes
Logging	Log4J	Yes	Yes	Yes	WSE diagnostics
WS-ResourceLifetime	Yes	Yes	Yes	Yes	Yes
WS-ResourceProperties	Yes	Yes	Yes	Yes	Yes
WS-ServiceGroup	Yes	Yes	Yes	Yes	Yes
WS-BaseFaults	Yes	Yes	Yes	Yes	Yes
WS-BaseNotification	Yes	Consumer	Yes	No	Yes
WS-BrokeredNotification	Partial	No	No	No	Yes
WS-Topics	Partial	Partial	Partial	No	Partial

GetRP Test

Distributed client and service on same LAN
(times in milliseconds)



Paul A. Farrell 2006 KENT STATE Grid Computing 29

GT4 WS Core Performance

(1) Message-level security (times in milliseconds)

	GT4 Java	GT4 C	GT4 Python	WSRF.NET
GetRP	181.96	14.77	140.50	81.39
SetRP	182.04	14.99	142.21	82.48
CreateR	188.46	14.98	132.26	96.22
DestroyR	182.03	15.76	136.12	86.89
Notify	219.51	N/A	244.93	101.57

(2) Transport-level security (times in milliseconds)

	GT4 Java	GT4 C	GT4 Python	WSRF.NET
getRP	11.46	2.85	149.67	12.91
setRP	11.47	2.86	150.79	12.3
createR	18.00	2.82	132.60	20.84
destroyR	14.92	2.71	149.21	16.05
Notify	29.26	9.67	169.07	45.0

"WSRF/WSNs Compared," HPDC 2005.