

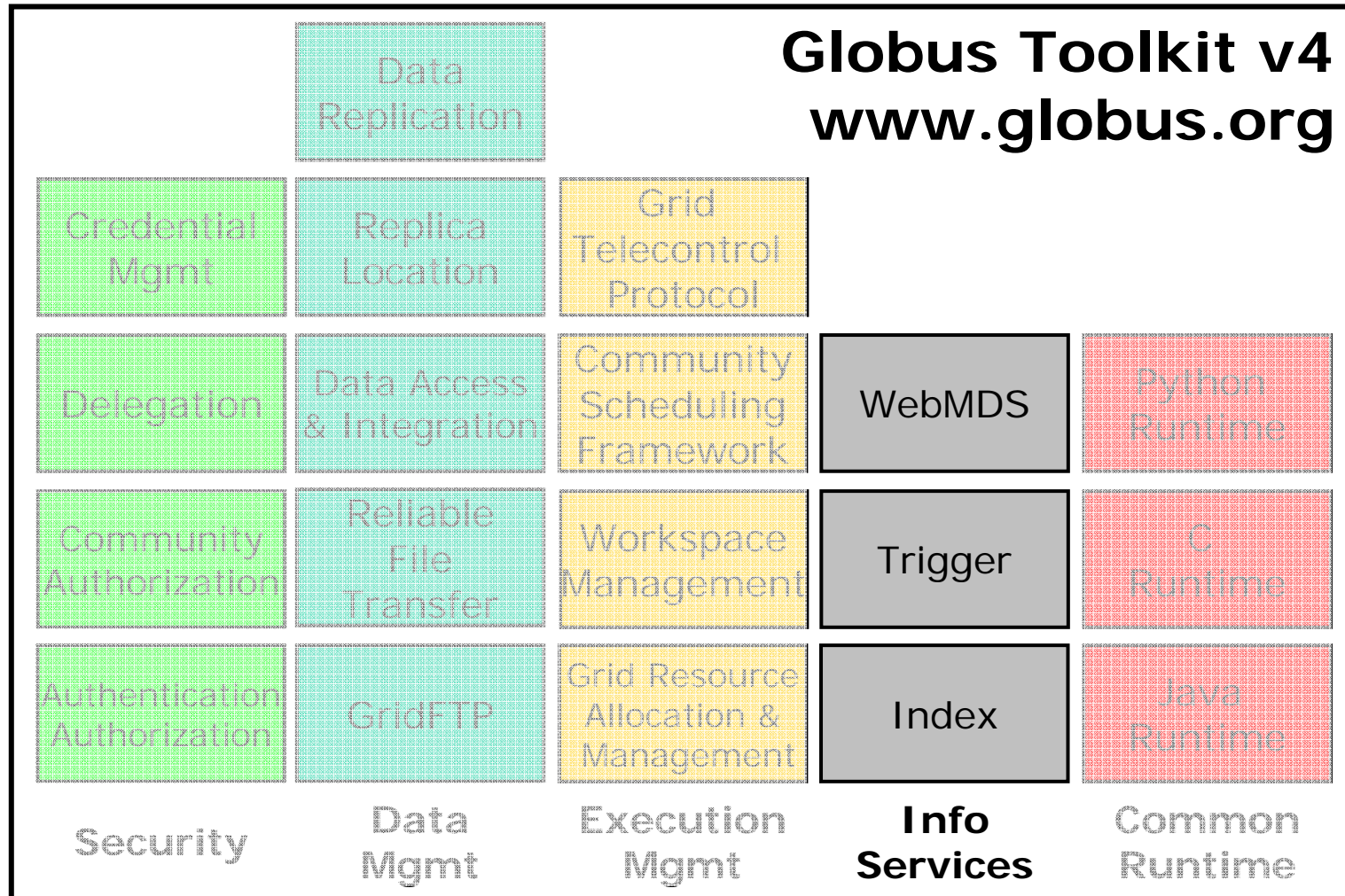


GT4 WSRF Core and MDS4

Gabor Kecskemeti

- MTA SZTAKI, Hungary
- Univ. Westminster, UK
kecskemeti@sztaki.hu

Globus Toolkit: Open Source Grid Infrastructure



Monitoring and Discovery

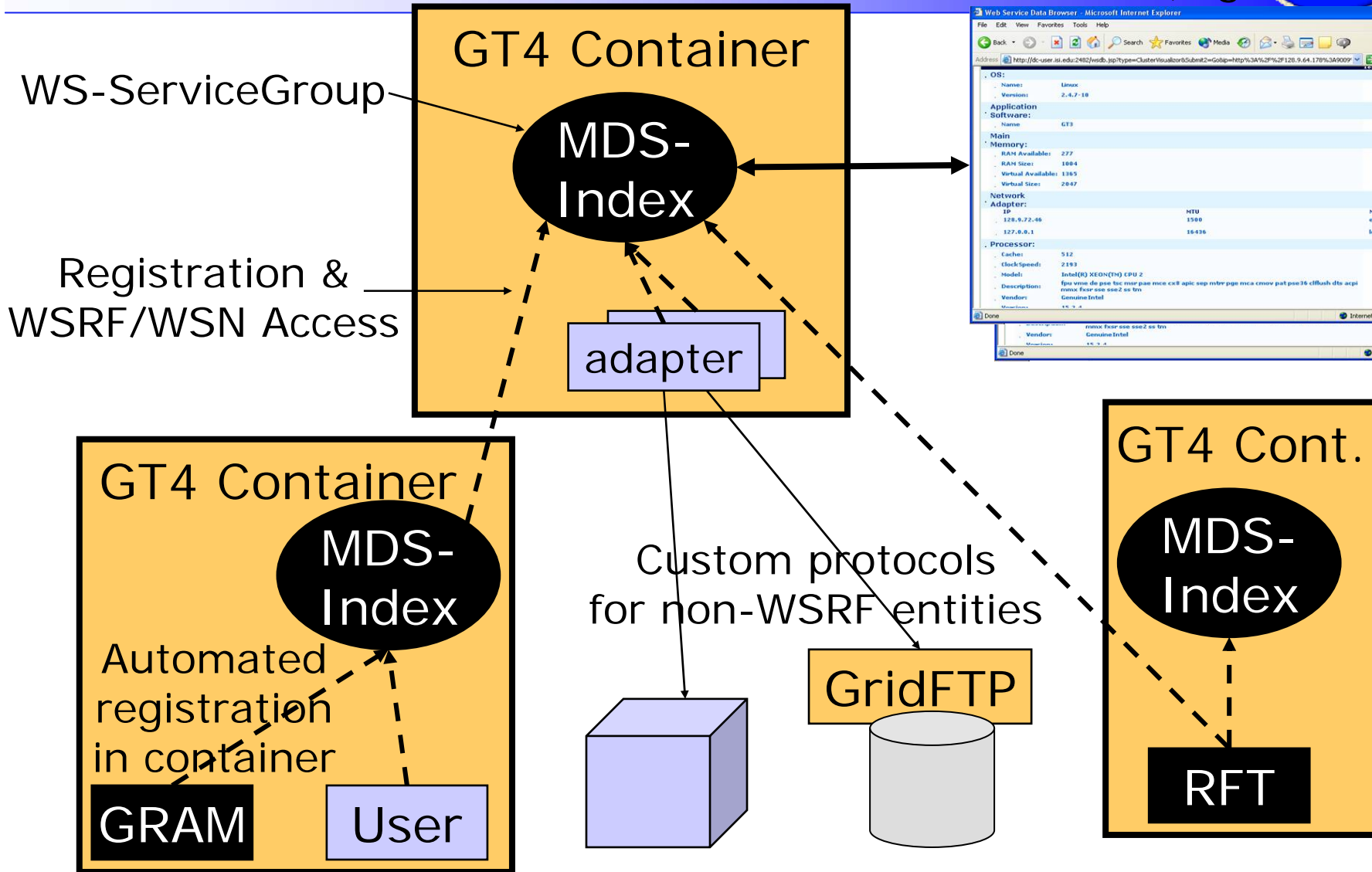
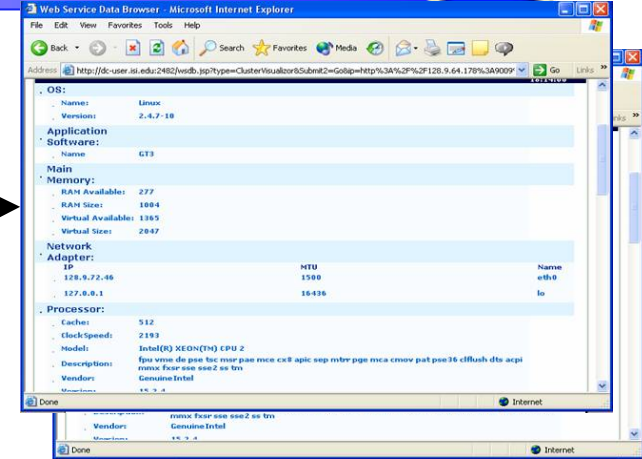


- “Every service should be monitorable and discoverable using common mechanisms”
 - WSRF/WSN provides those mechanisms
- A common aggregator framework for collecting information from services, thus:
 - MDS-Index: Xpath queries, with caching
 - MDS-Trigger: perform action on condition
 - (MDS-Archiver: Xpath on historical data)
- Deep integration with Globus containers & services: every GT4 service is discoverable
 - GRAM, RFT, GridFTP, CAS, ...

GT4 Monitoring & Discovery



(e.g., WebMDS)

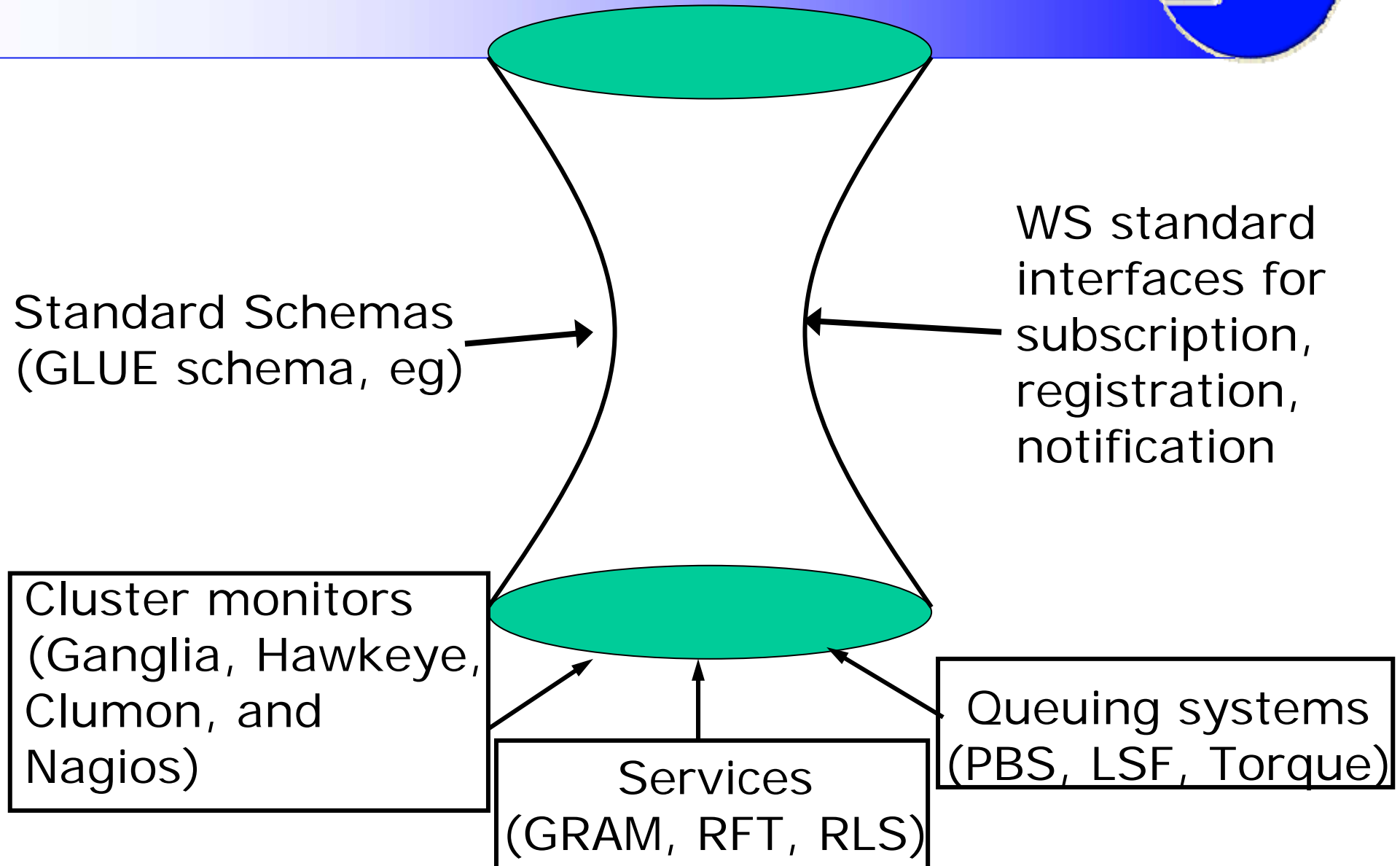


Monitoring and Discovery System (MDS4)



- Grid-level monitoring system
 - Aid user/agent to identify host(s) on which to run an application
 - Warn on errors
- Uses standard interfaces to provide publishing of data, discovery, and data access, including subscription/notification
 - WS-ResourceProperties, WS-BaseNotification, WS-ServiceGroup
- Functions as an hourglass to provide a common interface to lower-level monitoring tools

Information Users :
Schedulers, Portals, Warning Systems, etc



MDS4 Components



- Information providers
 - Monitoring is a part of every WSRF service
 - Non-WS services are also be used
- Higher level services
 - Index Service – a way to aggregate data
 - Trigger Service – a way to be notified of changes
 - Both built on common aggregator framework
- Clients
 - WebMDS
- All of the tools are schema-agnostic, but interoperability needs a well-understood common language

Higher-Level Services



- Index Service
 - Caching registry
- Trigger Service
 - Warn on error conditions
- Archive Service
 - Database store for history (in development)
- All of these have common needs, and are built on a common framework

Common Aggregator Framework



- Basic framework for higher-level functions
 - Subscribe to Information Provider(s)
 - Do some action
 - Present standard interfaces

Aggregator Framework Features



- 1) Common configuration mechanism
 - Specify what data to get, and from where
- 2) Self cleaning
 - Services have lifetimes that must be refreshed
- 3) Soft consistency model
 - Published information is recent, but not guaranteed to be the absolute latest
- 4) Schema Neutral
 - Valid XML document needed only

MDS4 Index Service



- Index Service is both registry and cache
 - Datatype and data provider info, like a registry (UDDI)
 - Last value of data, like a cache
- In memory default approach
 - DB backing store currently being developed to allow for very large indexes
- Can be set up for a site or set of sites, a specific set of project data, or for user-specific data only
- Can be a multi-rooted hierarchy
 - No *global* index

MDS4 Trigger Service



- Subscribe to a set of resource properties
- Evaluate that data against a set of pre-configured conditions (triggers)
- When a condition matches, action occurs
 - Email is sent to pre-defined address
 - Website updated
- Similar functionality in Hawkeye

WebMDS User Interface



- Web-based interface to WSRF resource property information
- User-friendly front-end to Index Service
- Uses standard resource property requests to query resource property data
- XSLT transforms to format and display them
- Customized pages are simply done by using HTML form options and creating your own XSLT transforms
- Sample page:
 - <http://mds.globus.org:8080/webmds/webmds?info=indxinfo&xsl=servicegroupxsl>

Example WebMDS screenshot



ServiceGroup Overview - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address <http://mds.globus.org:8080/webmds/webmds?info=indexinfo&xsl=servicegroupxsl>

Google uk train booking Search Web PageRank 2158 blocked AutoFill Options uk train booking

ServiceGroup Overview

This page provides a brief overview of Web Services and/or WS-Resources that are members of a WS-ServiceGroup.

This WS-ServiceGroup has 4 direct entries, 33 in whole hierarchy.

Resource Type	ID	Information	
Unknown	128.9.72.106	Aggregator entry with no content from https://128.9.72.106:8443/wsrp/services/ReliableFileTransferFactoryService	detail
GRAM	128.9.72.106	0 queues, submitting to 0 cluster(s) of 0 host(s).	detail
ServiceGroup	128.9.72.140	This WS-ServiceGroup has 11 direct entries, 29 including descendants.	detail
ServiceGroup	128.9.72.178	This WS-ServiceGroup has 4 direct entries, 4 including descendants.	detail
RFT	128.9.72.178	0 active transfer resources, transferring 0 files. 40.55 GB transferred in 173769 files since start of database.	detail
GRAM	128.9.72.178	0 queues, submitting to 1 cluster(s) of 10 host(s).	detail
GRAM	128.9.72.178	1 queues, submitting to 1 cluster(s) of 10 host(s).	detail
GRAM	128.9.72.178	2 queues, submitting to 1 cluster(s) of 10 host(s).	detail
ServiceGroup	128.9.72.106	This WS-ServiceGroup has 3 direct entries, 3 including descendants.	detail
GRAM	128.9.72.106	0 queues, submitting to 0 cluster(s) of 0 host(s).	detail
GRAM	128.9.72.106	1 queues, submitting to 0 cluster(s) of 0 host(s).	detail
RFT	128.9.72.106	0 active transfer resources, transferring 0 files. 8.28 GB transferred in 8595 files since start of database.	detail
ServiceGroup	128.9.64.179	This WS-ServiceGroup has 4 direct entries, 4 including descendants.	detail
GRAM	128.9.64.179	1 queues, submitting to 1 cluster(s) of 15 host(s).	detail
GRAM	128.9.64.179	5 queues, submitting to 1 cluster(s) of 15 host(s).	detail
RFT	128.9.64.179	0 active transfer resources, transferring 0 files. 63.16 GB transferred in 108704 files since start of database.	detail
GRAM	128.9.64.179	0 queues, submitting to 1 cluster(s) of 15 host(s).	detail
ServiceGroup	128.9.128.168	This WS-ServiceGroup has 3 direct entries, 3 including descendants.	detail
GRAM	128.9.128.168	0 queues, submitting to 0 cluster(s) of 0 host(s).	detail
RFT	128.9.128.168	0 active transfer resources, transferring 0 files. 10.52 GB transferred in 23489 files since start of database.	detail

Internet

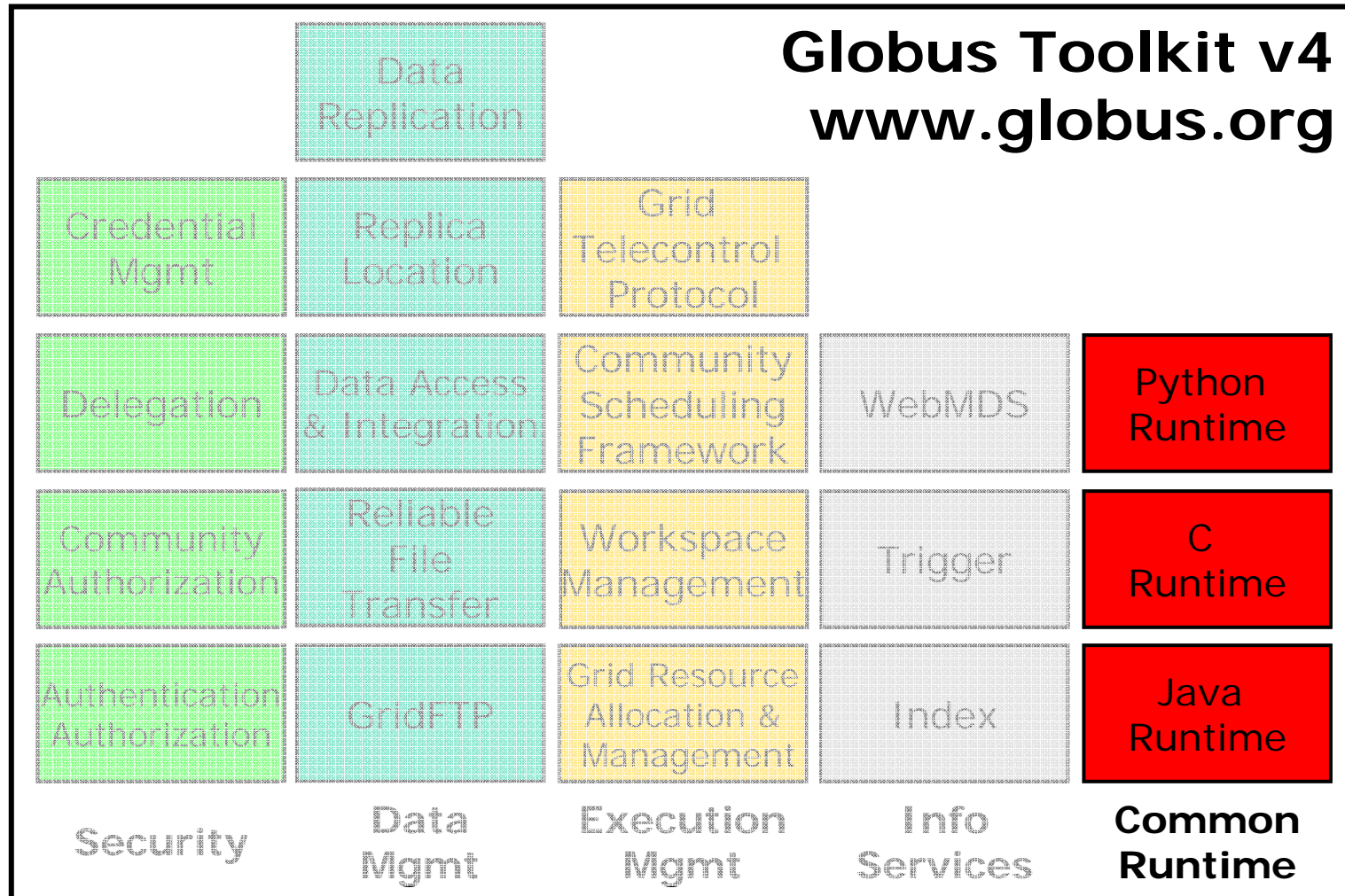
start Eudora - [In] ServiceGroup Over... Windows Messenger shakey.mcs.anl.go... Microsoft PowerPoi... 9:42 AM

Information Providers

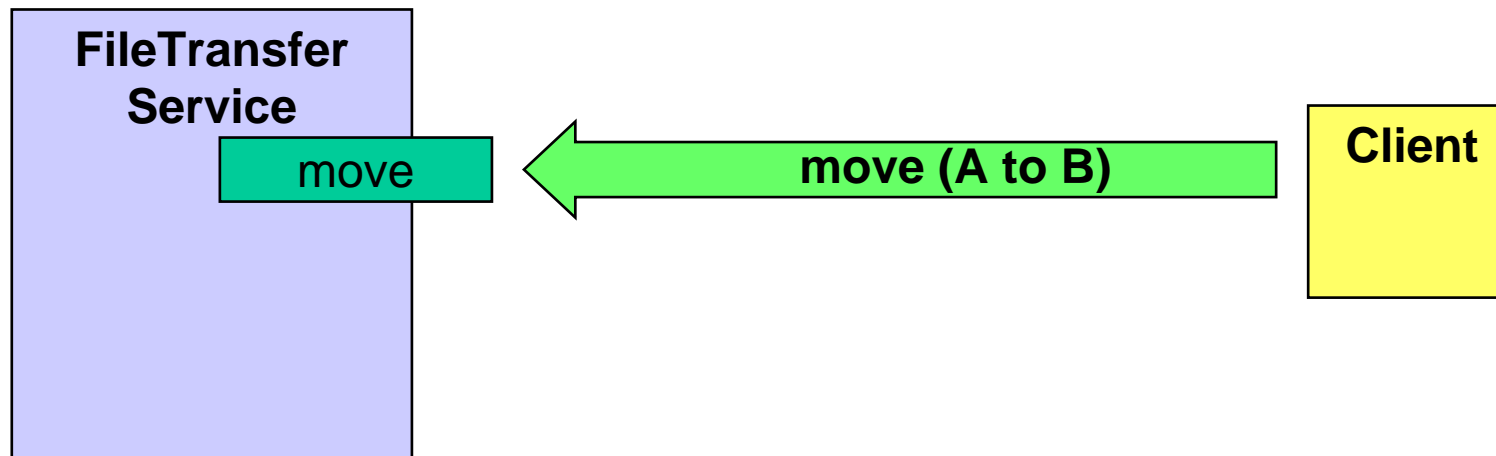


- **GT4 information providers** collect information from some system and make it accessible as WSRF resource properties
- Growing number of information providers
 - Ganglia, CluMon, Nagios
 - SGE, LSF, OpenPBS, PBSPro, Torque
- Many opportunities to build additional ones
 - E.g., network monitoring, storage systems, various sensors, see **GEMLCA Monitoring Tool (GMT)**

Globus Toolkit: Open Source Grid Infrastructure

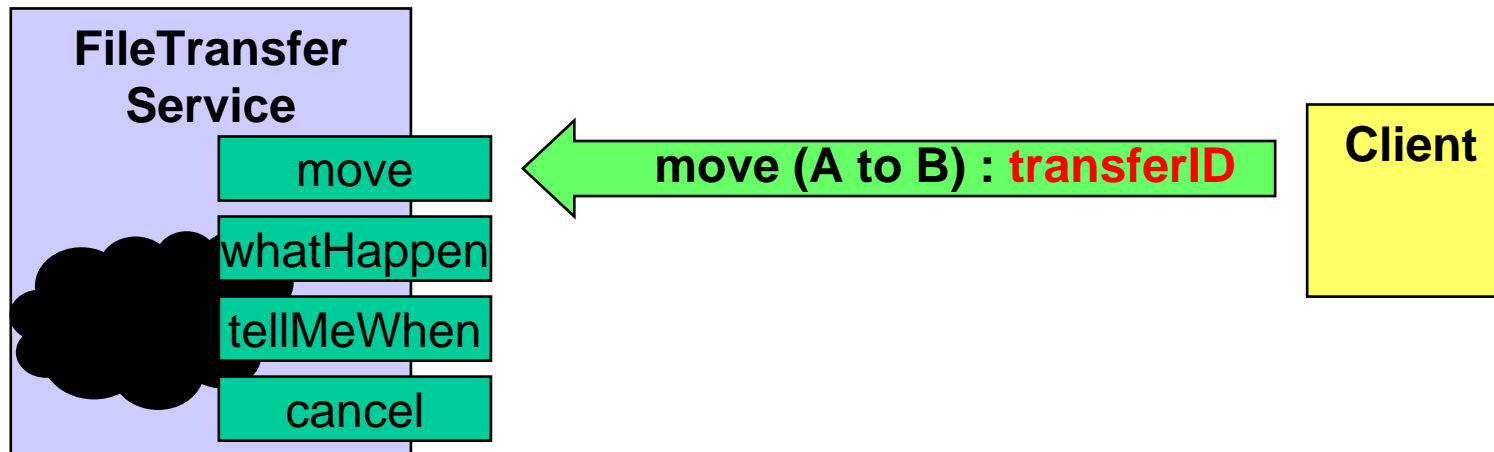


“Stateless” vs. “Stateful” Services



- 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

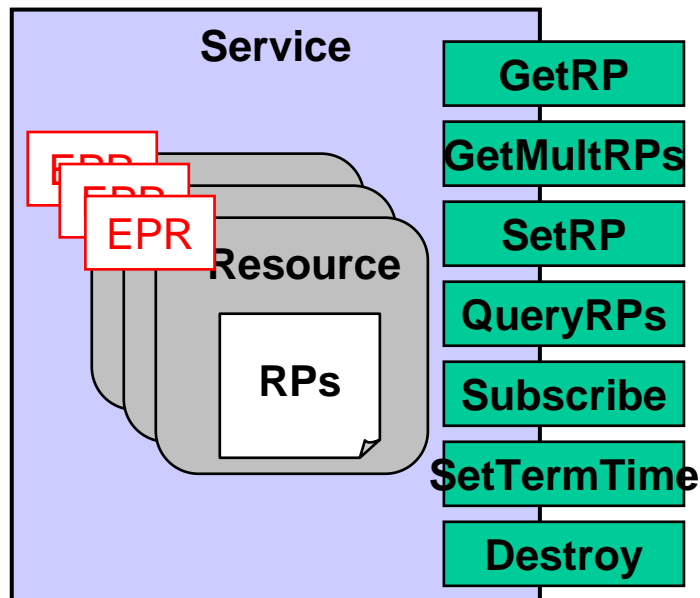
FileTransferService (without WSRF)



- 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**)



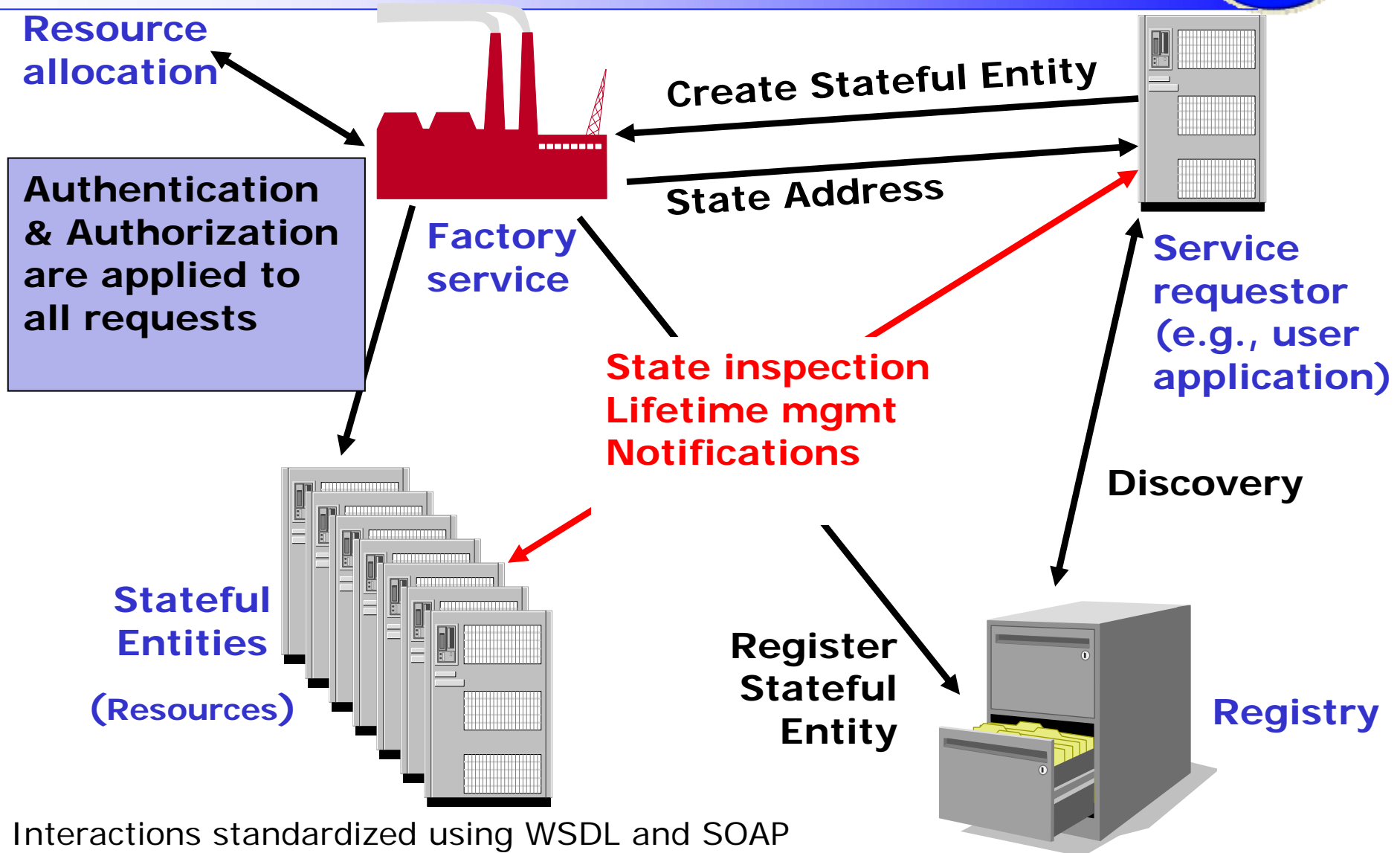
WSRF in a Nutshell



- Service - permanent
- State representation
 - Resource - transient - no creation method in standards
 - Resource Property
- State identification
 - Endpoint Reference
- State Interfaces
 - GetRP, QueryRPs, GetMultipleRPs, SetRP
- Lifetime Interfaces
 - SetTerminationTime, Immediate
- Notification Interfaces
 - Subscribe, Notify
- ServiceGroups - e.g. MDS4 resource document aggregation



Modeling State in Web Services



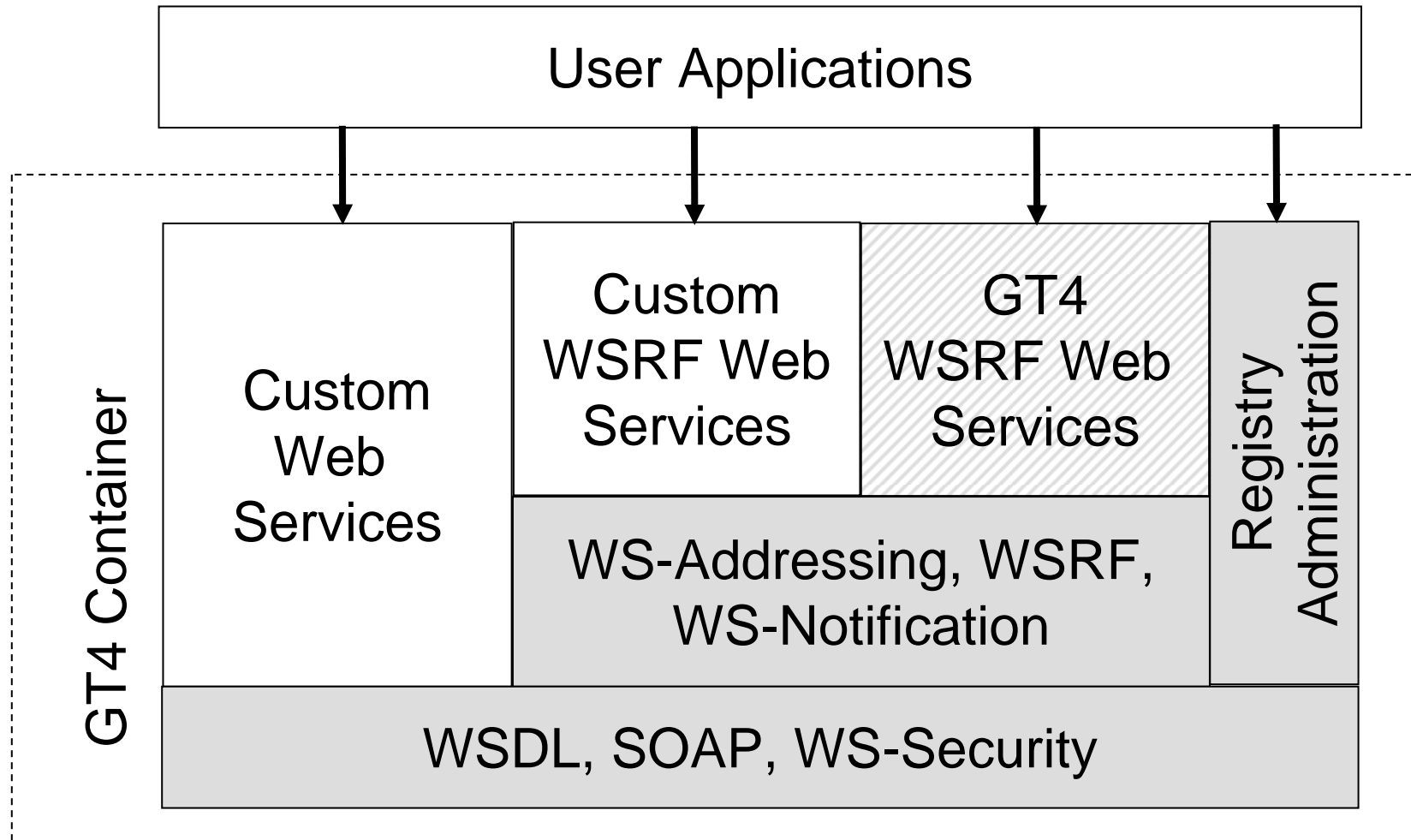
Interactions standardized using WSDL and SOAP

GT4 Web Services Runtime



- How does WSRF meets Grid? GT4 gives a hand with the WSRF Core, the WS-GRAM, RFT and other GT4 services are implemented with its help.
- 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 Web Services Runtime





GT4 WS Core in a Nutshell

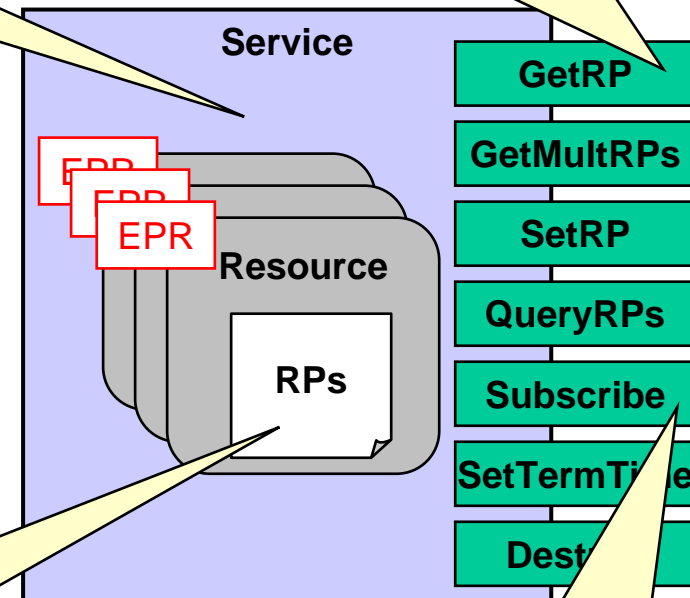
Implementation of WSRF:
Resources,
EndpointReferences,
ResourceProperties

Operation Providers: pre-build
implementations of WSRF
operations

Example WS resource document

- from GEMMLCA:

```
<GLCProcessResourceProperties>
  <UserHomeFolder>
    /home/gkecskem
  </UserHomeFolder>
  <LastJobStateChange>
    335462 DONE
  </LastJobStateChange>
</GLCProcessResourceProperties>
```



Implementations of Resources
(ReflectionResource,
PersistentReflectionResource)
and ResourceProperties
(SimpleResourceProperty,
ReflectionResourceProperty)

Notification implementation:
Topics, TopicSet, Embedded
Notification Consumer service

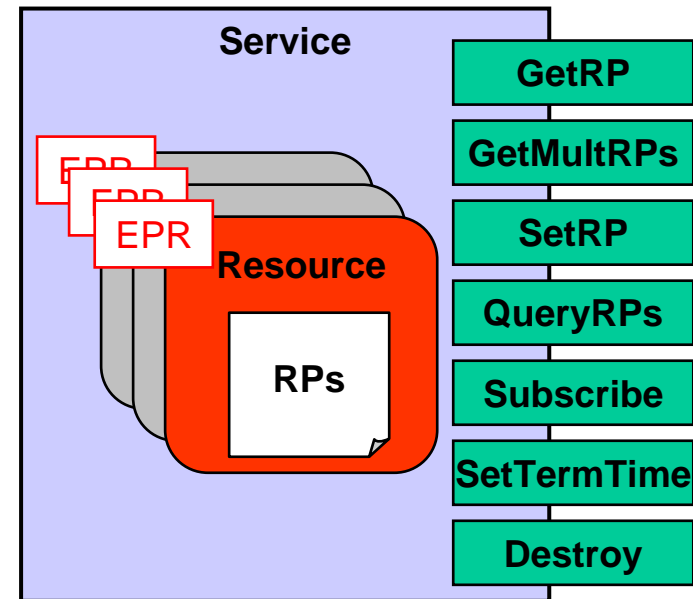
WSDL Resource document definition



```

<definitions>
  <types>
    <xsd:schema>
      ...
      <xsd:element name="GLCProcessResourceProperties">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element ref="tns:UserHomeFolder" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element ref="tns:LastJobStateChange" minOccurs="1"
              maxOccurs="1"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:schema>
  </types>
  ...
  <portType name="GLCProcessPortType"
    ... wsrp:ResourceProperties="tns:GLCProcessResourceProperties">
    ...
  </portType>
  ...
</definitions>

```



Defining WSRF Standard Interfaces with GT4



```
<definitions>
```

```
...
```

```
<portType name="GLCProcessPortType"
```

```
  wsdlpp:extends="
```

```
  wsrpw:GetResourceProperty
```

```
  wsrpw:SetResourceProperties
```

```
  wsntw:NotificationProducer
```

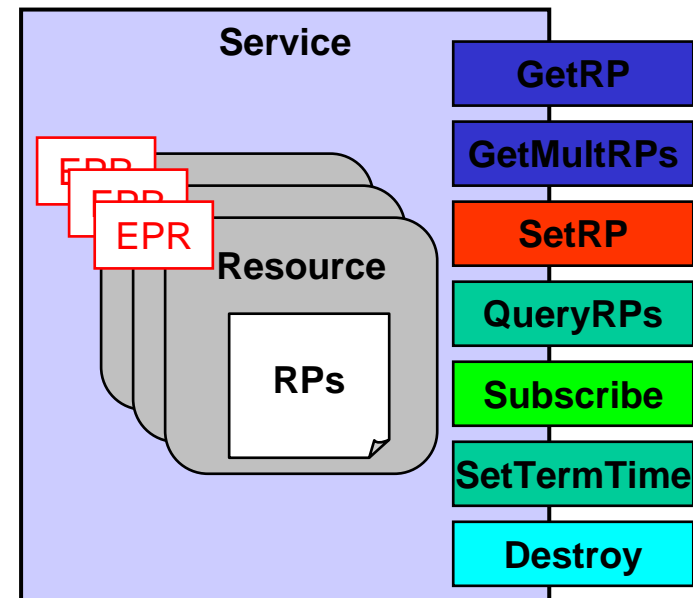
```
  wsrlw:ImmediateResourceTermination">
```

```
...
```

```
</portType>
```

```
...
```

```
</definitions>
```



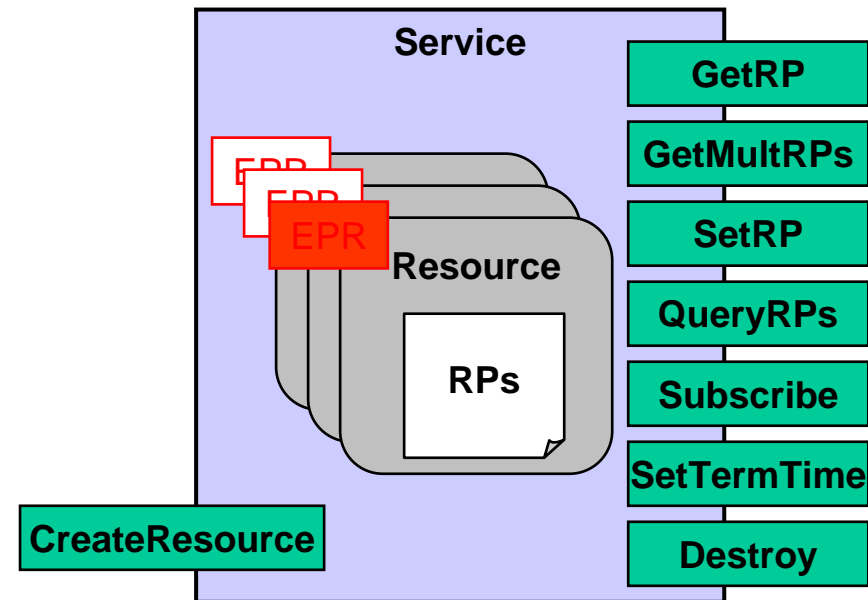
WSDL Resource Creation - Acquiring the EPR



```

<definitions>
<types>
<xsd:schema>
...
<xsd:element name="createResourceResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="wsa:EndpointReference"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
...
</xsd:schema>
</types>
<message name="CreateResourceResponse">
  <part name="response" element="tns:createResourceResponse"/>
</message>
...
</definitions>

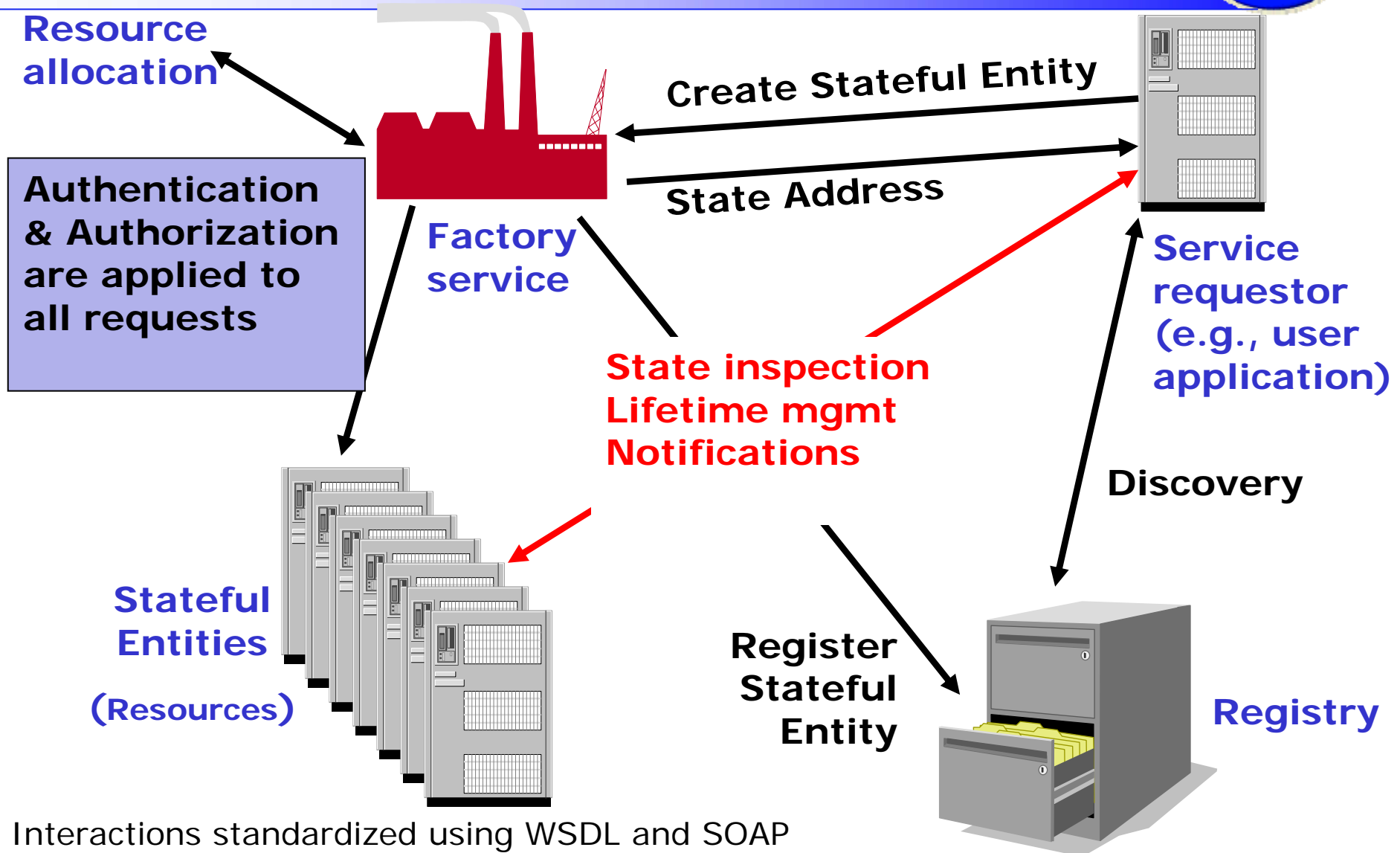
```



Not standardised method

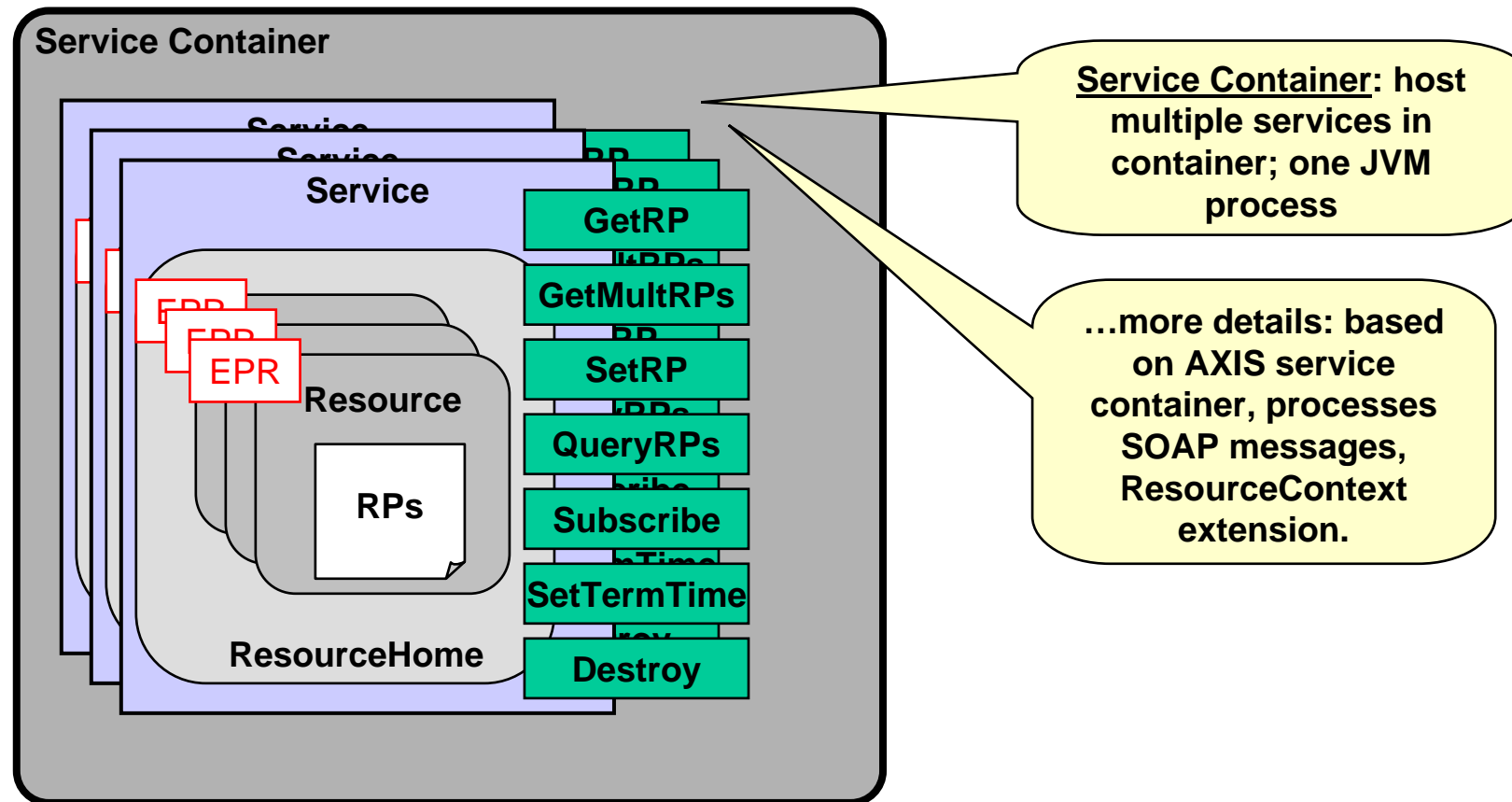


Modeling State in Web Services



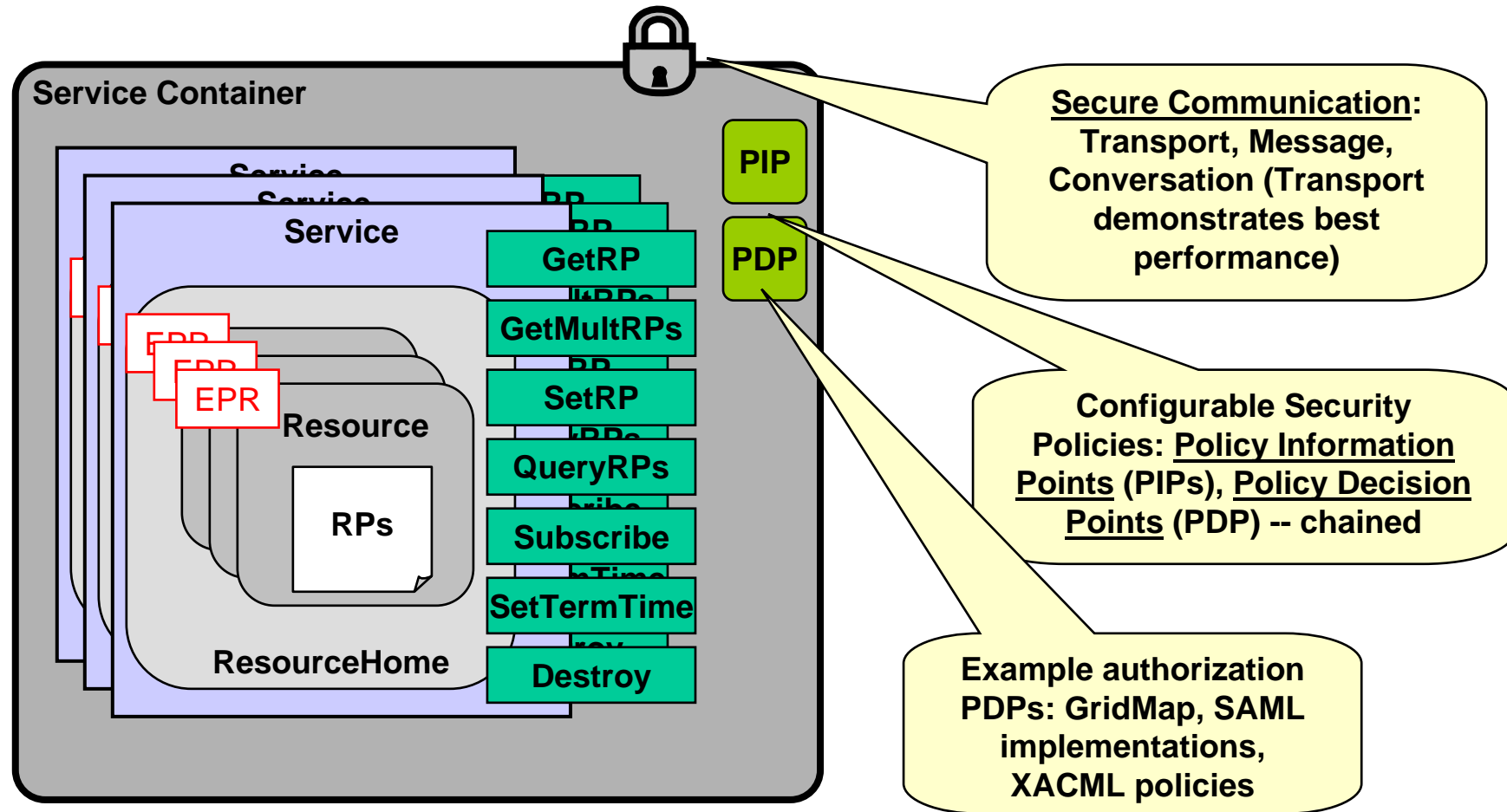


GT4 WS Core in a Nutshell



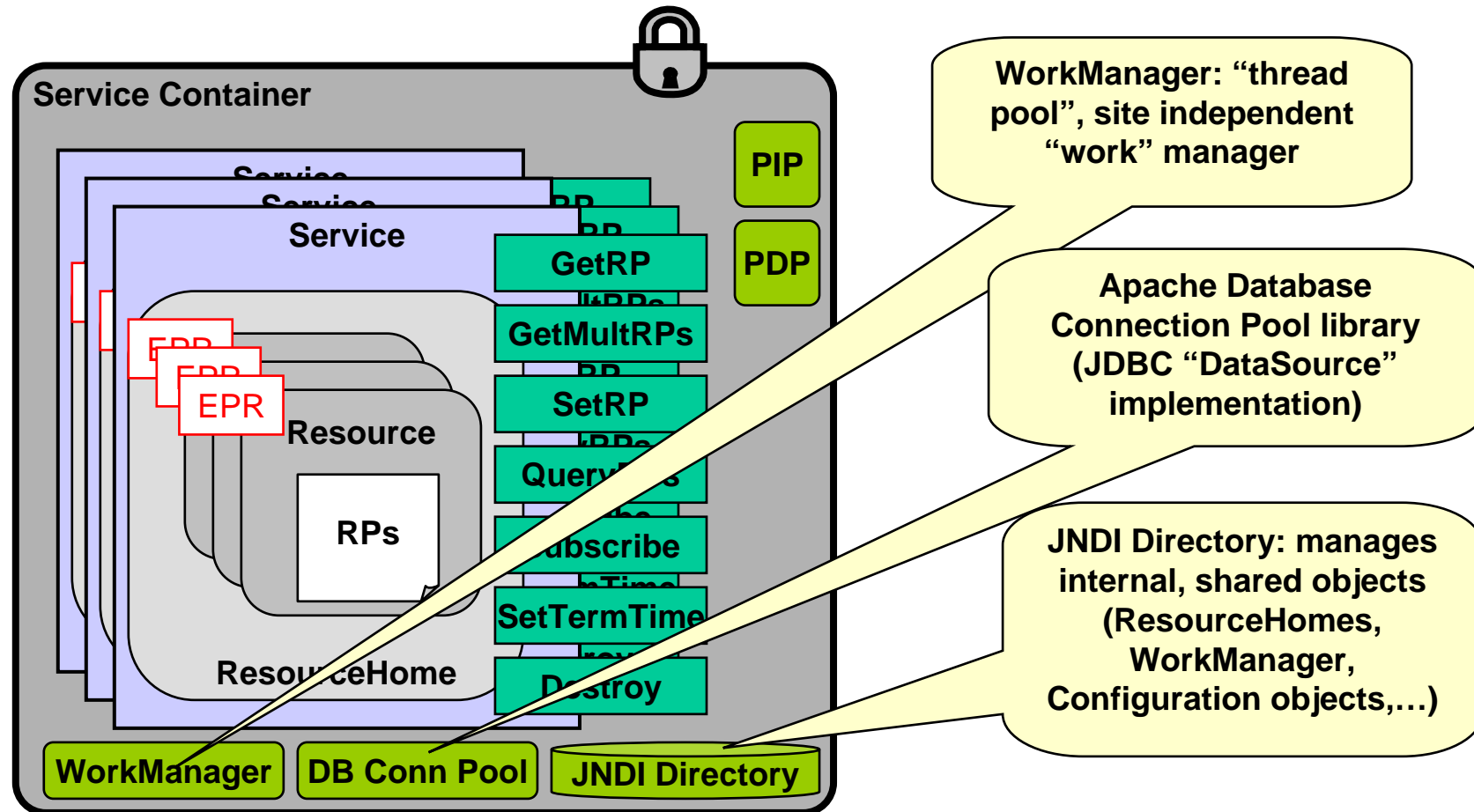


GT4 WS Core in a Nutshell



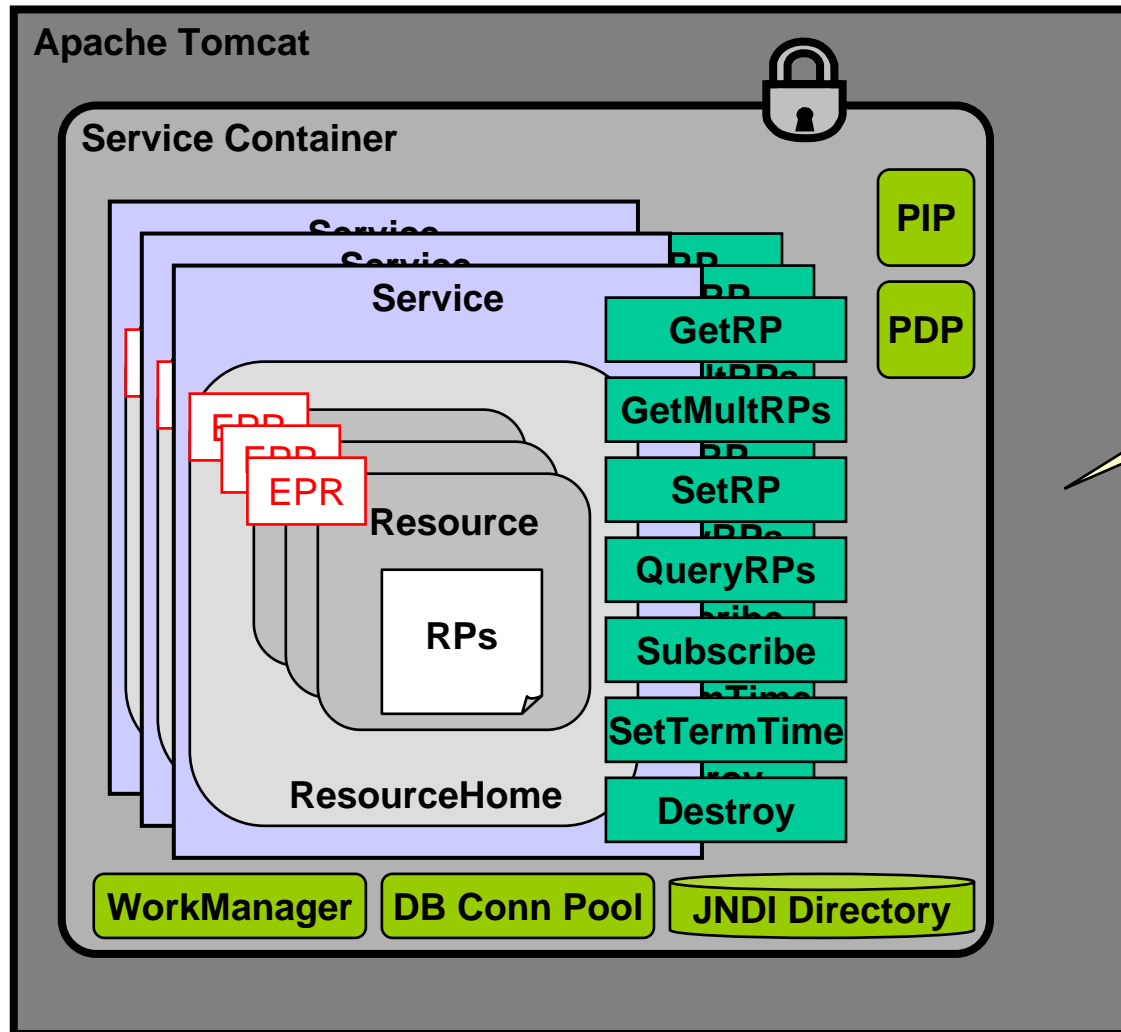


GT4 WS Core in a Nutshell





GT4 WS Core in a Nutshell



Deploy Service Container "standalone" or within Apache Tomcat

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**



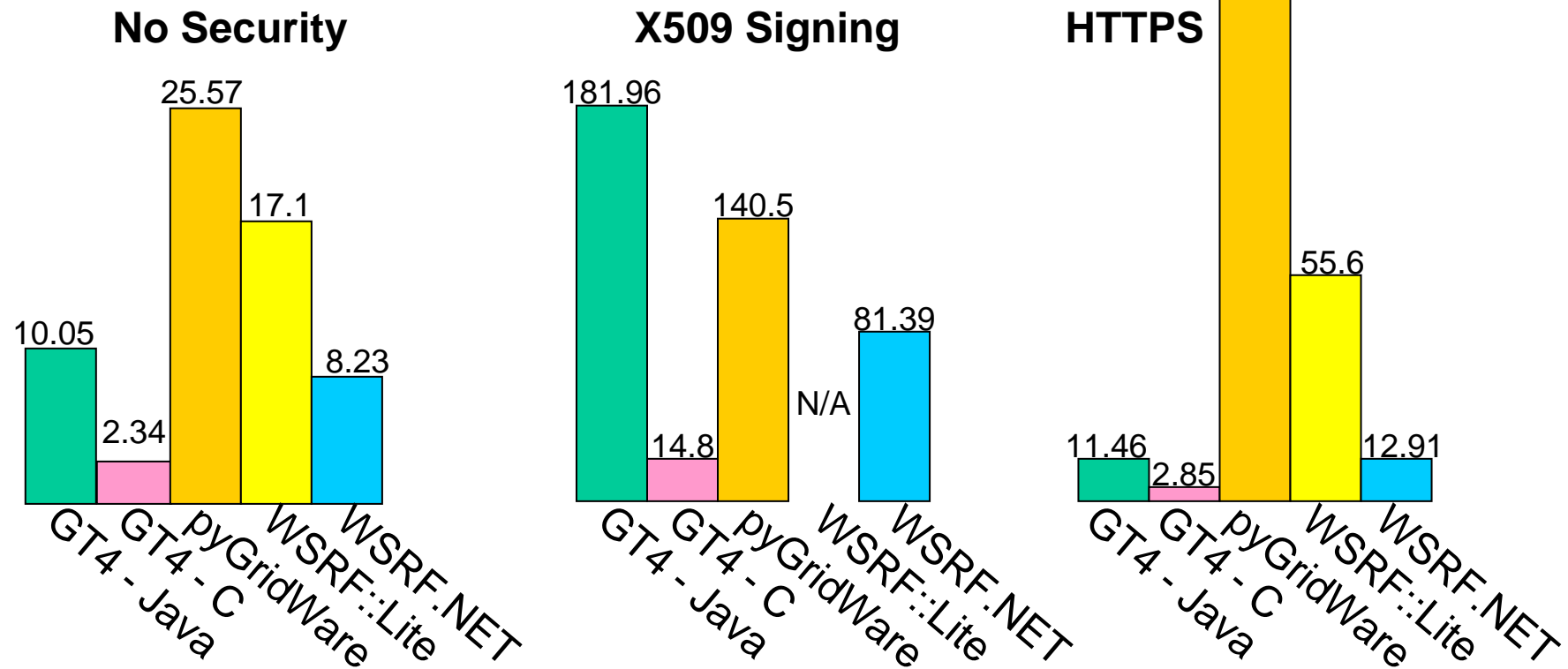
WSRF/WSNs Compared (HPDC 2005)

	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)



Java examples: Acquiring the endpoint reference



```
EndpointReferenceType localendp = new EndpointReferenceType();
localendp.setAddress(new Address("https://grid-compute-
ws.cpc.wmin.ac.uk:8443/wsrf/services/myWSRFService"));
MyWSRFServicePortType serviceContact =
    ServiceLocator.getMyWSRFServicePortTypePort(localendp);
SetSecProps((Stub) serviceContact, Identity, cred);
CreateResourceResponse createResourceResponse =
    serviceContact.createResource(new CreateResource());
localendp=createResourceResponse.getEndpointReference();
// Now localendp holds the resource's EPR, lets use it!
MyWSRFServicePortType resourceContact =
    ServiceLocator.getMyWSRFServicePortTypePort(localendp);
SetSecProps((Stub) resourceContact, Identity, cred);
```

Legend: Important, Third party (Axis, GT4), Generated from WSDL

Java Examples: Setting a Resource Property



```
WSResourcePropertiesServiceAddressingLocator locator = new
    WSResourcePropertiesServiceAddressingLocator();
SetResourceProperties_PortType rpPort =
    locator.getSetResourcePropertiesPort(endpoint);
SetSecProps((Stub) rpPort, Identity, cred);
UpdateType update = new UpdateType();
MessageElement msg = new
    MessageElement(GLCProcessNamespaces.RP_LASTJOBSTATECHANGED, "-1
    GEMLCA_PRESTATE");
update.set_any(new MessageElement[] { msg });
SetResourceProperties_Element request = new SetResourceProperties_Element();
request.setUpdate(update);
rpPort.setResourceProperties(request);
```

Legend: **Important**, **Third party** (Axis, GT4), **Generated from WSDL**



Example Security setup

```
public static void SetSecProps(Stub OnMe, String Identity,
    GSSCredential cred) {
    OnMe._setProperty(Constants.GSI_TRANSPORT,
    Constants.SIGNATURE);
    OnMe._setProperty(GSIConstants.GSI_MODE,
    GSIConstants.GSI_MODE_FULL_DELEG);
    OnMe._setProperty(Constants.AUTHORIZATION, new
    IdentityAuthorization(this.Identity == null?defaultIdentity:Identity));

    if (cred != null)
    OnMe._setProperty(GSIConstants.GSI_CREDENTIALS, cred);
}
```

Legend: Important, Third party (Axis, GT4), Generated from WSDL