UniGrids and GPE
A Client Framework for Interoperability

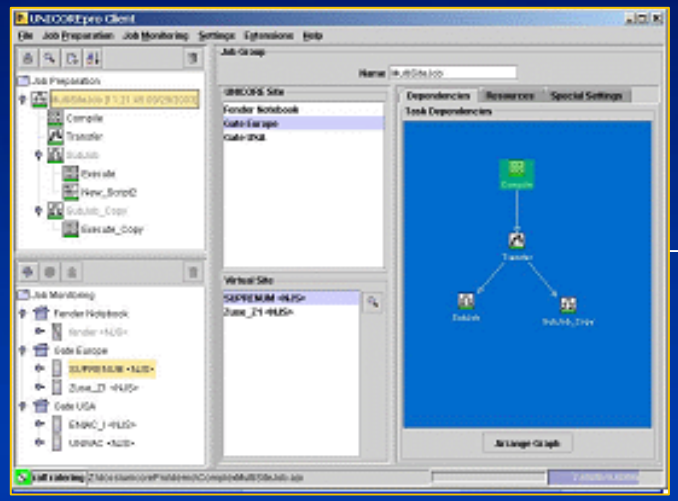
Unicore Summit Sophia Antipolis, October 11-12, 2005

Thomas Kentemich, Intel PDSD

First of all...

- What do we mean by interoperability?

Production UNICORE

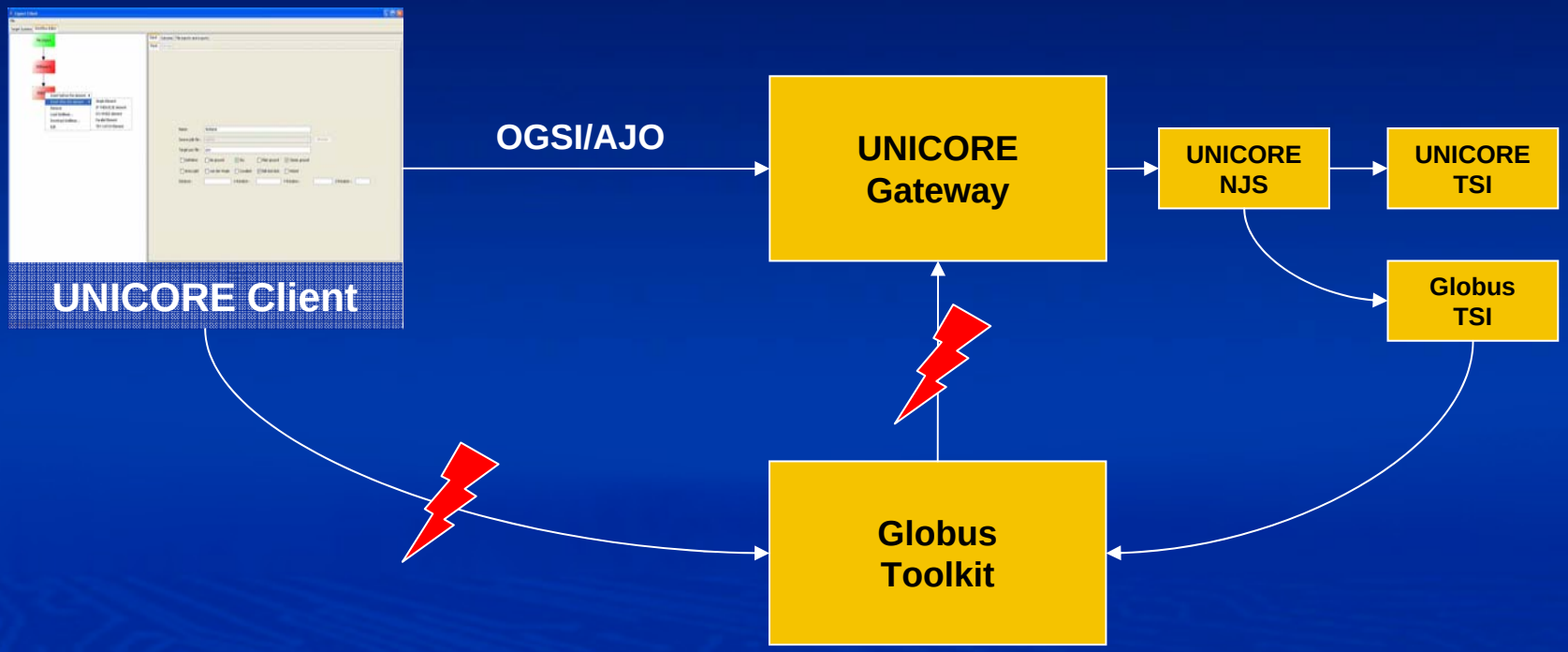


UNICORE Client

UPL/AJO

- Stable, working solution
- No interoperability

GRIP Interoperability

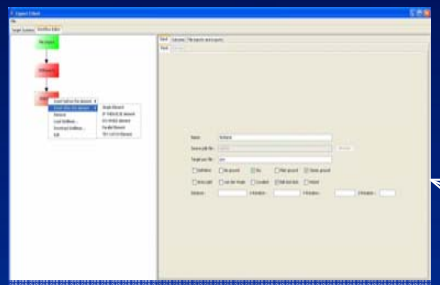


- New web service and Grid standards will allow better ways of interoperability

Need for Standards

- Defining standards like WSRF is not enough
- Need to define semantics of WSRF enabled services
- GPE defines a set of atomic services for job execution and data management

UniGrids Interoperability



GPE Expert Client

SOAP, WSDL, WSRF, WS-Addressing, WS-Security, JSDL and other standards

UNICORE/GS

Globus Toolkit

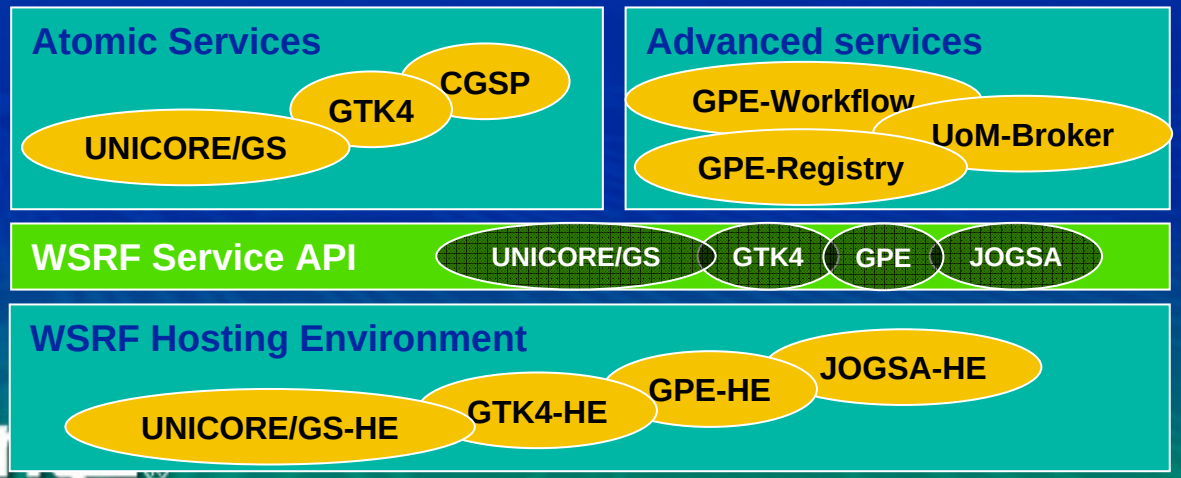
Other OGSA-compliant servers

Atomic Services

- Atomic service interfaces define mandatory functionality for system, file and job management
- Different protocols and description languages are announced via resource properties

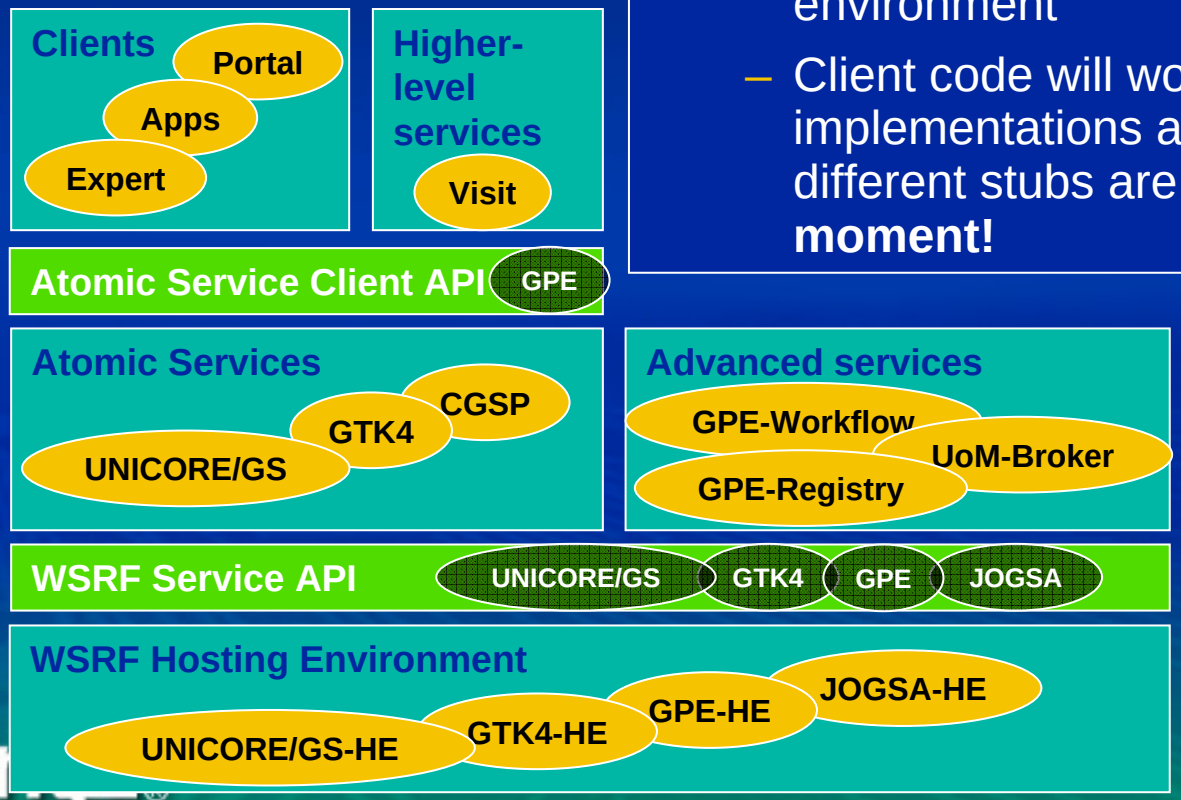
Three levels of interoperability

- Level 1: Interoperability between WSRF services
  - UNICORE/GS passed the official WSRF interop test
  - GPE and JOGSA hosting environments succesfully tested against UNICORE/GS and other endpoints
  - **WSRF specification will be finalized soon!**
    - Currently: UNICORE/GS: WSRF 1.3, GTK: WSRF 1.2 draft 1

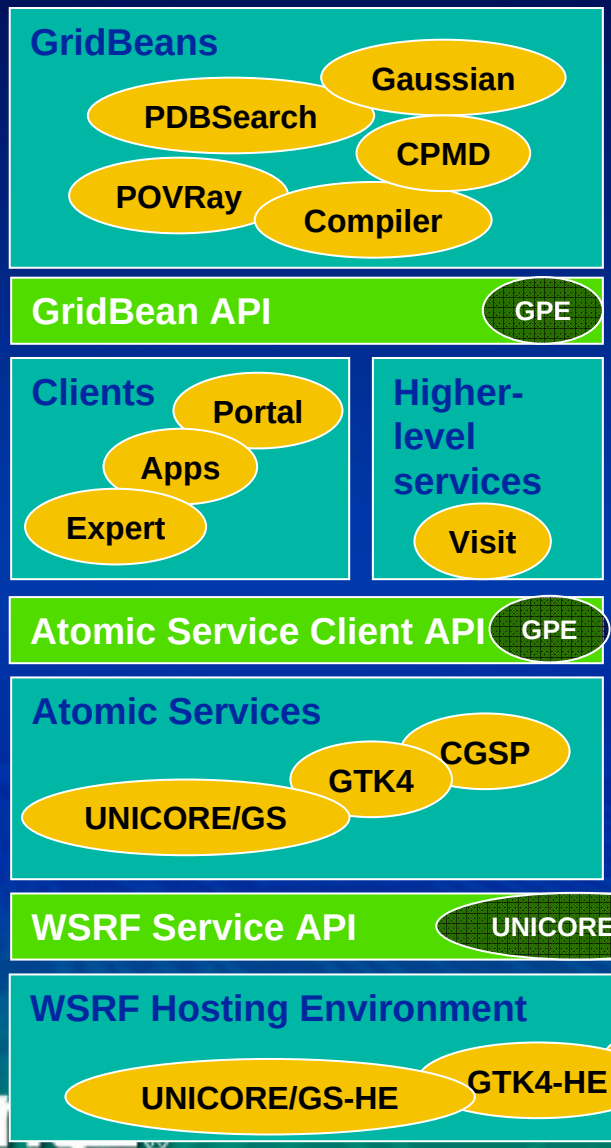


Three levels of interoperability

- Level 2: Interoperability between atomic service implementations
  - Client API hides details about WSRF hosting environment
  - Client code will work with different WSRF implementations and WSRF versions if different stubs are being used **at the moment!**



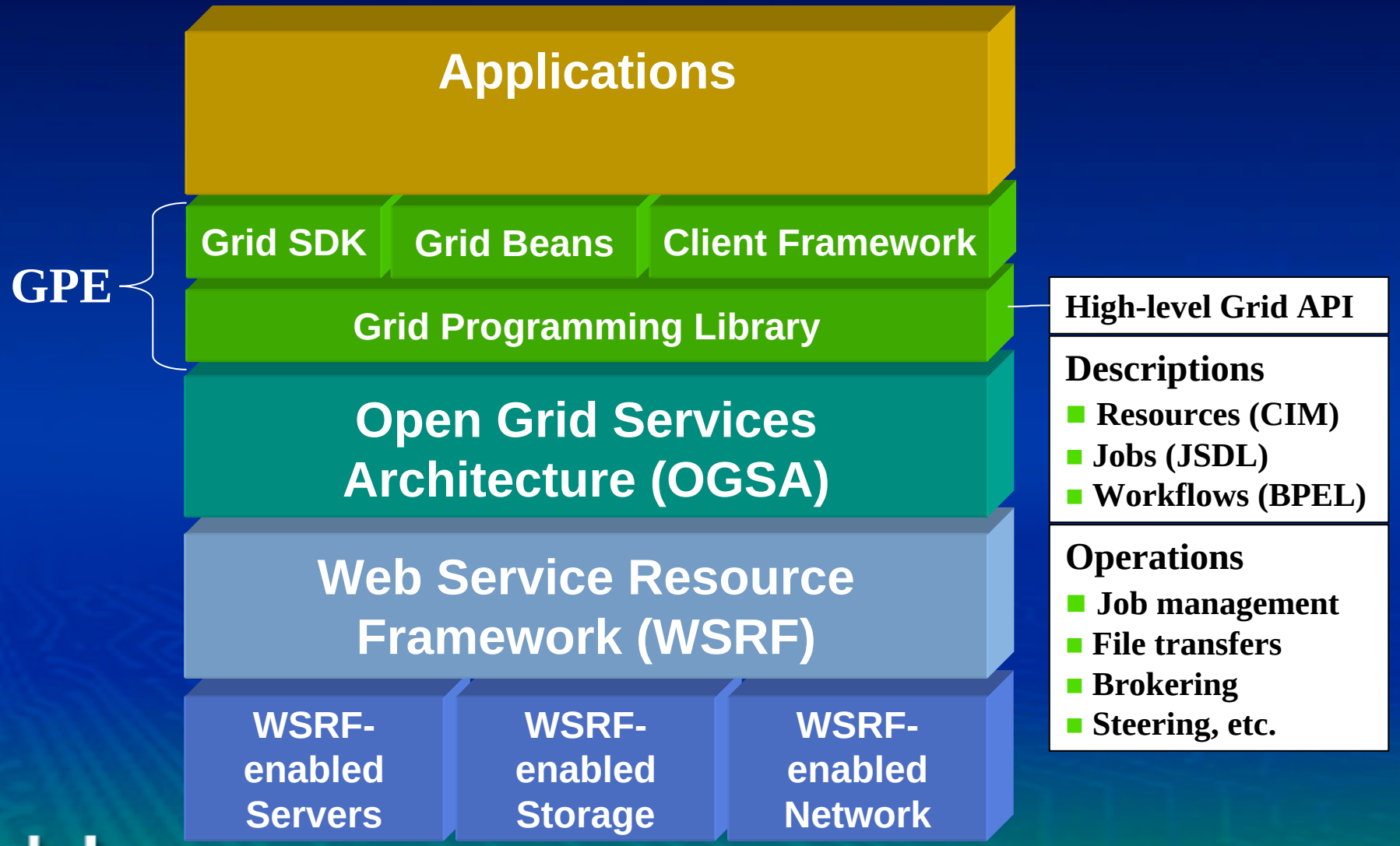
Three levels of interoperability



- Level 3: GridBeans working with different Client implementations
  - Independent of atomic service implementations
  - Independent of specification versions being used
  - GridBean run on different atomic service implementations without modifications
  - GridBeans survive version changes in the underlying layers and are easy to maintain

Details on GPE

Grid Programming Environment (GPE)



Standards

- **JSDL (Job Submission Description Language)**
  - High level job description that can be submitted to all target systems offering a JSDL interface
- **CIM (Common Information Model)**
  - Used to describe resources
  - Usage of CIM management interfaces for Grid administration
- **BPEL (Business Process Execution Language)**
  - Integration of Grid Bean services into larger business process workflows
- **WS* (WS-Addressing, WSRF, WSN, etc.)**
  - Interoperation with other Grid Middleware
- **OGSA (Open Grid Services Architecture)**
  - Share components with other architectures

GPE Components Overview

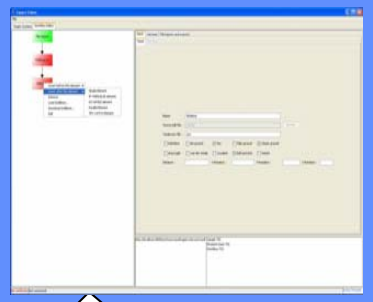
API and library for GridBean development

Grid SDK

Dynamically loaded, portable client plug-ins

GridBeans

Client Framework



- Expert Client
- Application Client
- Portal

WSRF Hosting Environment

- Native GPE
- Globus Toolkit 4.0
- UNICORE/GS
- or other

GridBean Service

Target System Registry

Workflow Execution Service

Target System Factory

Target System Service

Job Management Service

Storage Management Service

File Import Service

File Export Service

Atomic Services

GPE services

Target System

Backend (UnicoreGS)

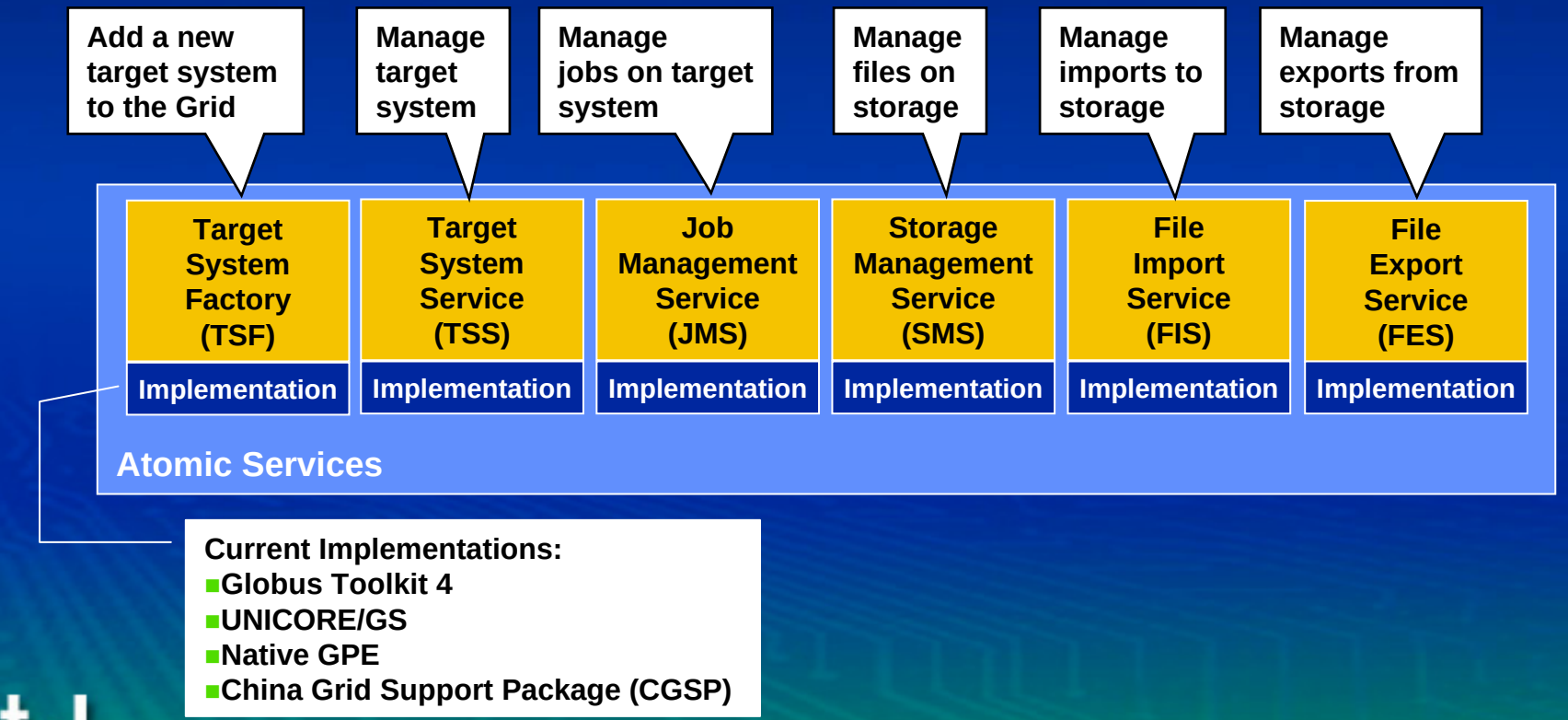
Target System

Target System Interface (TSI)

Lightweight Java component to perform the actual work

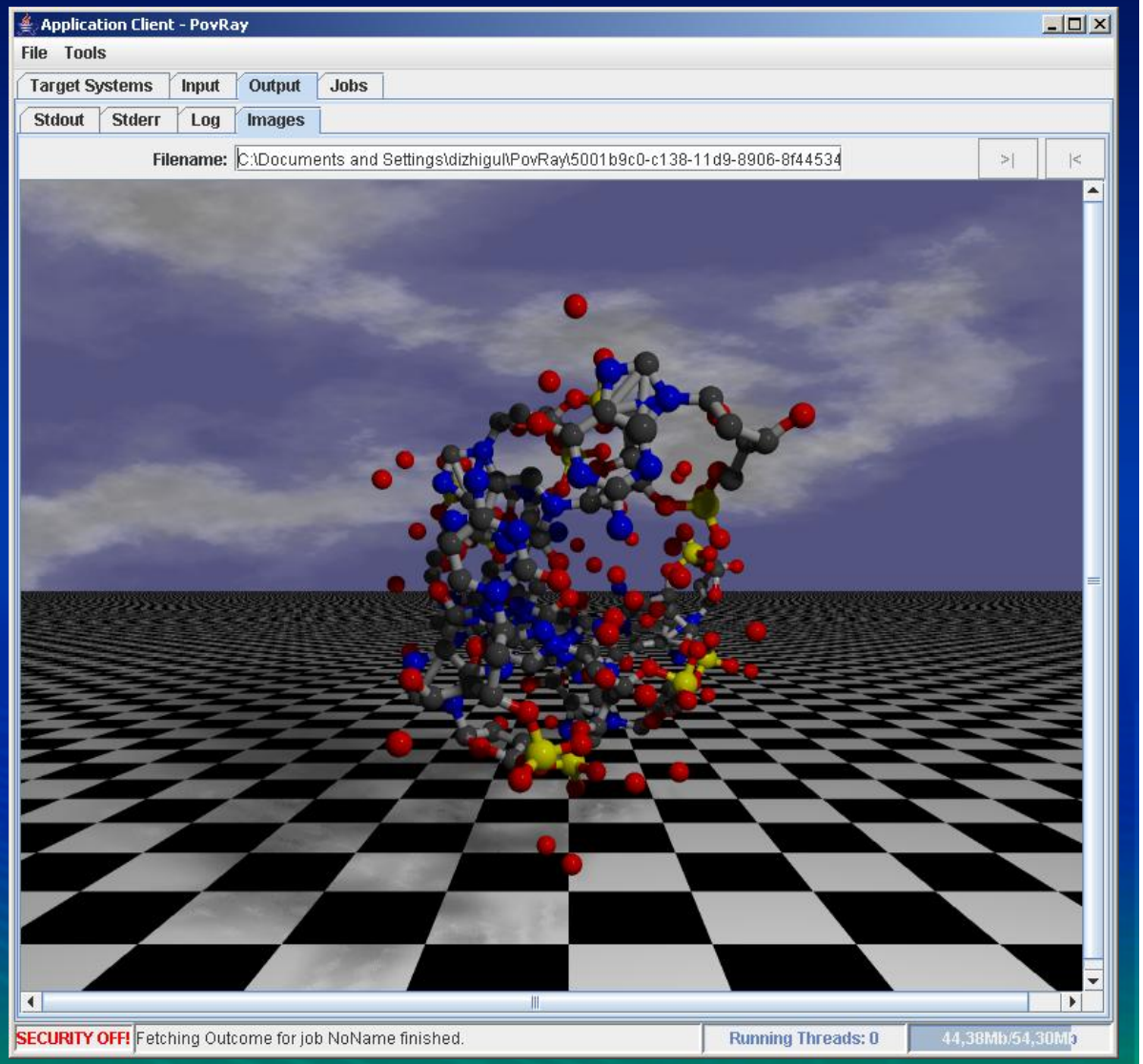
Atomic Services Overview

- Atomic service interfaces define basic set of operations and properties that have to be available on a Grid
- Different implementations of interfaces for different infrastructures



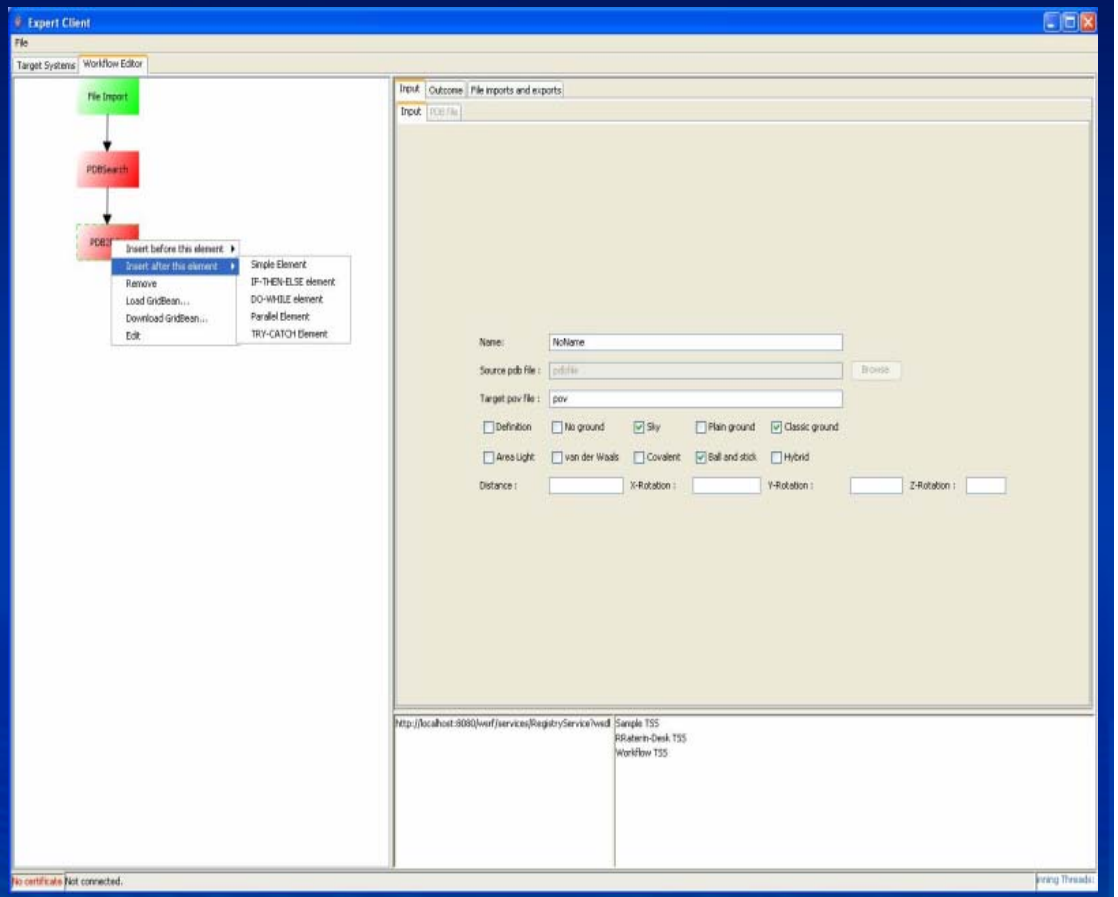
Client Framework: Application Client

- For some users it is sufficient to offer interfaces that are restricted to run and manage a certain application on the Grid.
- For this category of users we implemented a thin **Application Client** with a functionality limited to application specific features.
- Lightweight Java application that can be run on mobile devices



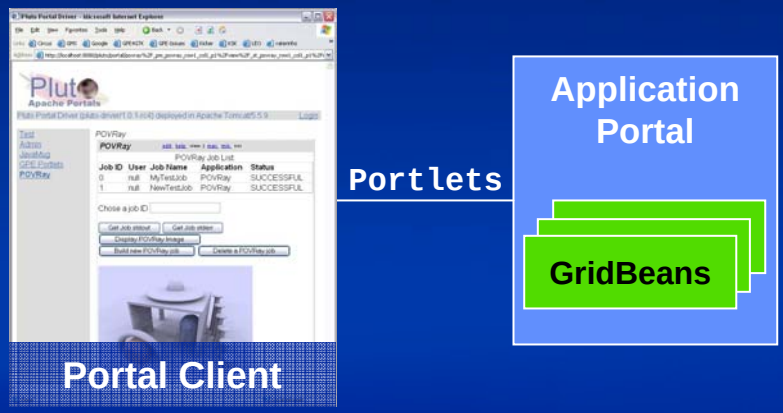
Client Framework: Expert Client

- Expert users want to
  - build their own complex workflows to combine different Grid services to complex applications
  - access information and broker services
  - use different identities on different systems.
- The Expert Client
  - provides a workflow editor to construct Grid specific BPEL workflows
  - manages multiple GridBeans
  - manages multiple certificates

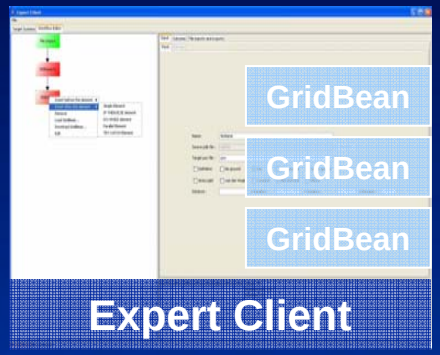


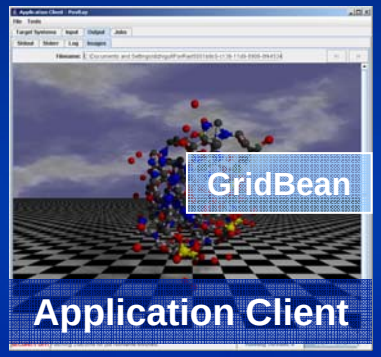
Client Framework: Portal Client

- For the **Grid-unaware user** GPE offers a web portal
  - Provide simple user interface in web browser
  - Hide Grid specific functionality
- GridBeans may provide JSR168 compliant portlets
  - In addition to client plug-ins
- GridBean portlets can be integrated into existing portal solutions
  - UPortal, GridSphere, etc.



Implement portable applications with GridBeans





GridBean SDK

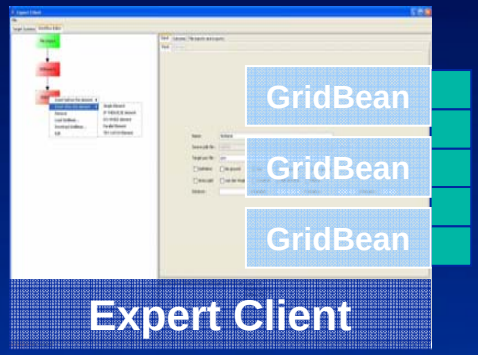
Atomic Service Client API, libraries and tools

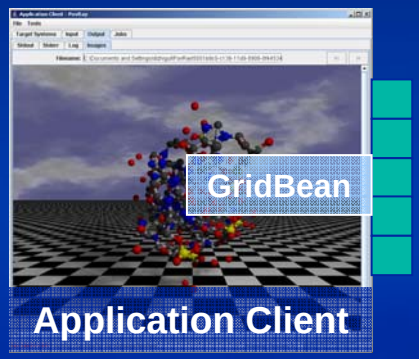
GridBean Service

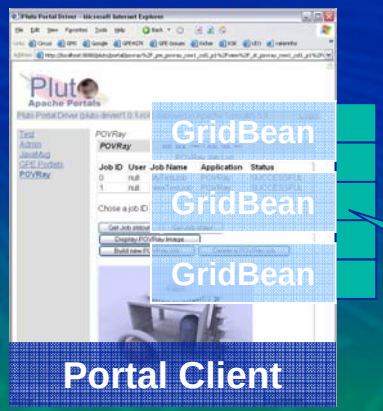
- GridBean
- GridBean
- GridBean

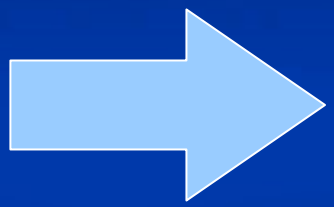
- GridBeans are the interoperable successors of UNICORE Client plug-ins

GPE as interoperability framework









UNICORE/GS

Globus Toolkit 4

China Grid Support Package

Other OGSA-compliant Grid servers

Atomic Service Client API

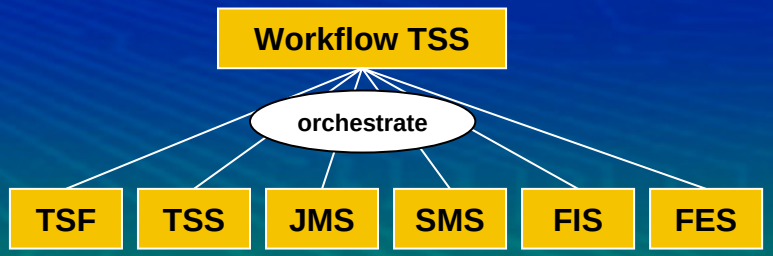
Atomic Services

Additional GPE Services: Registry

- Registry keeps track of static and dynamic information
  - Hardware properties, available software, ...
  - Workload, available disk space, ...
- TSS contacts registry on startup and when properties change using WS-Notification
- Clients and services query informations about target systems from registry
- Implemented as WSRF Service Group

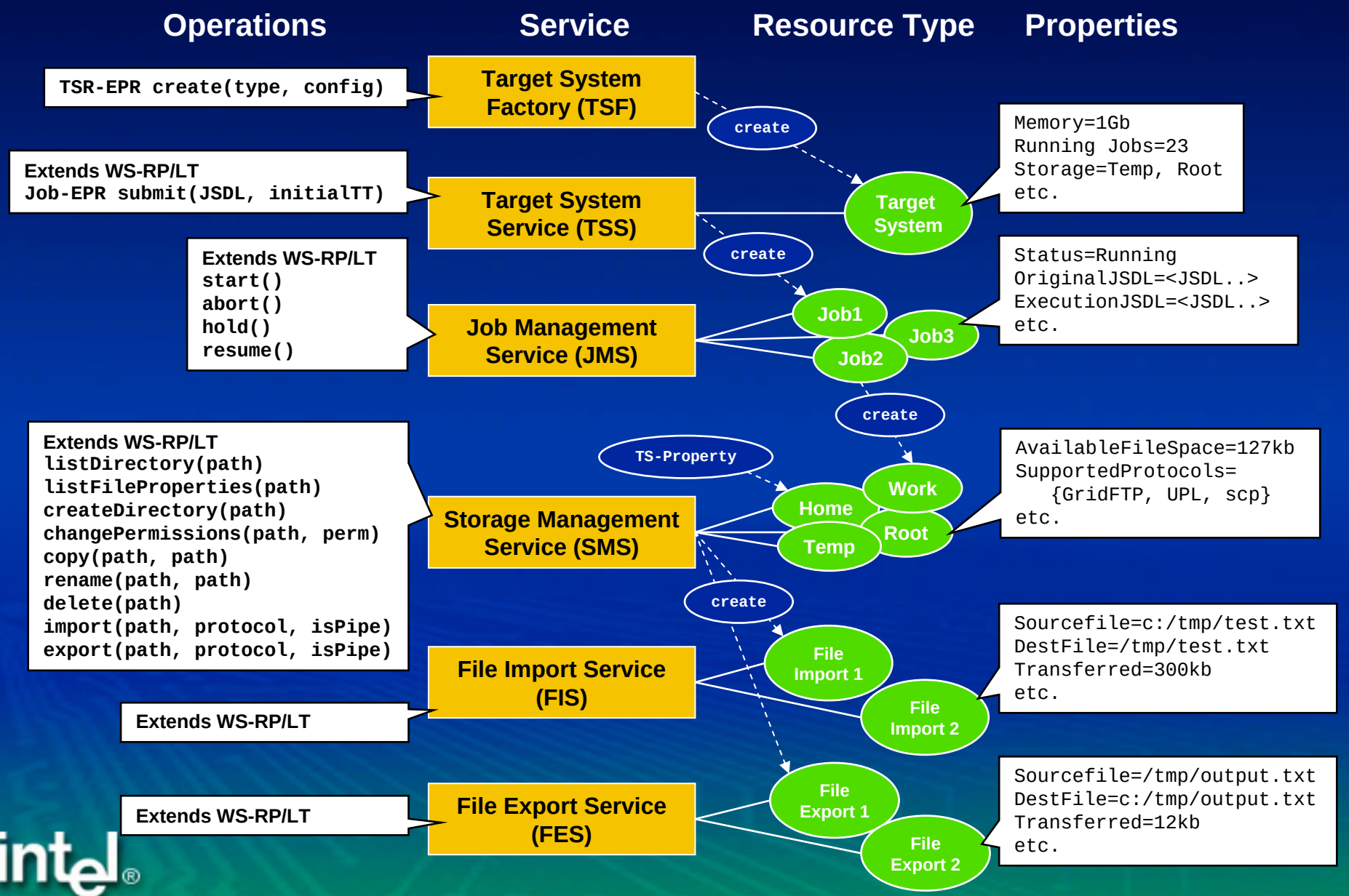
Additional GPE Services: Workflow Execution Service

- Use Grid-specific BPEL subset to orchestrate WSRF services in complex workflows
  - Allows integration into larger business processes
- Implemented as WSRF service itself
  - *Workflow TSS* accepts workflow JSDL descriptions in submit operation
  - BPEL description is extension in JSDL
- Information about workflow (state, input/output files, etc.) is kept in BPEL variables
  - BPEL variables are accessible as resource properties of the Workflow TSS



How does it work in concrete?

Atomic Service Interfaces based on WSRF



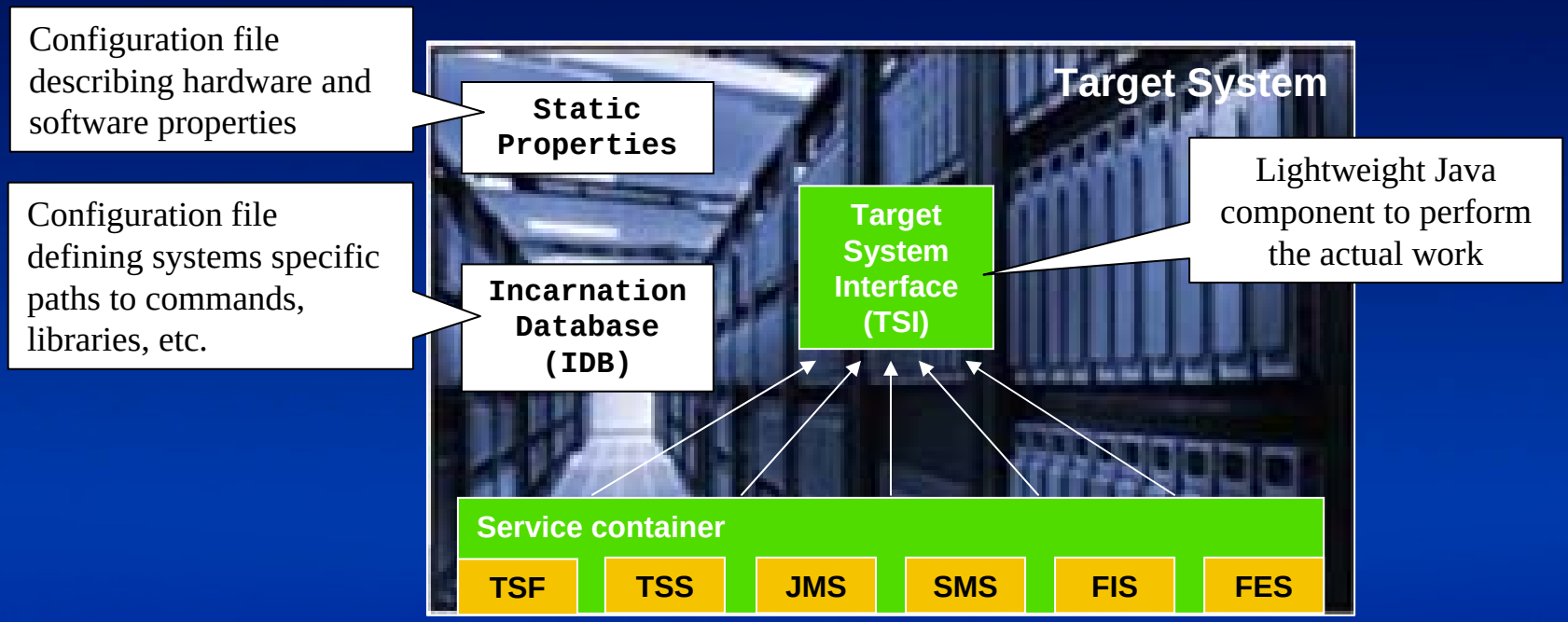
Hiding platform-specific information with Application Resources

- Abstract job concept borrowed from UNICORE
  - No concrete platform specific information in job decription (paths, libraries, etc.)
  - Job will be **incarnated** on target system
  - For security and portability reasons!
- Available applications can be queried from target system resource properties
- Use JSDL Posix extensions to specify required application resources in submitted job

File Transfers

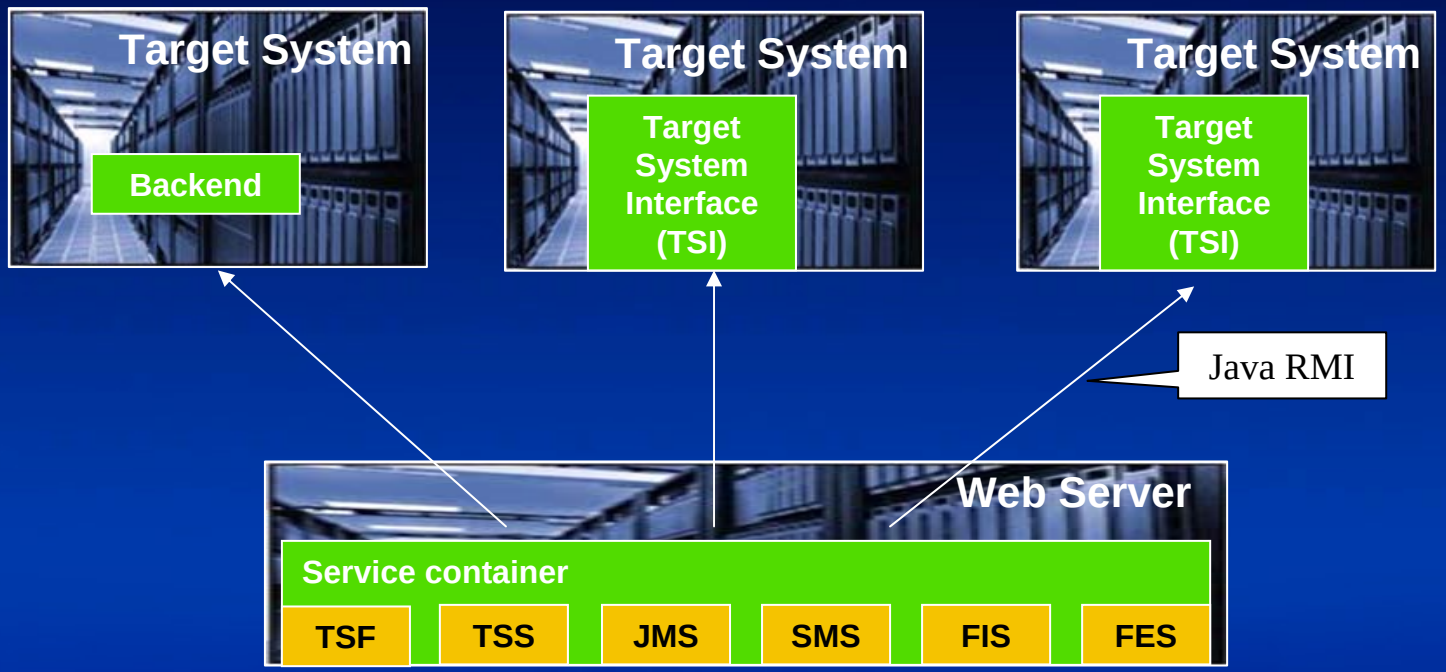
- Atomic Services support different protocols
  - FTP
  - GridFTP
  - plain HTTP(s)
  - SOAP with Attachements (parallel)
  - Baseline file transfer
  - <add your own here...>
- Storage management announces available protocols via its resource properties
  - Client queries available protocols and selects appropriate one
    - GridFTP for large high-performant transfers
    - HTTP, SOAP w/a to work with firewall limitations

Adding a target system to a Grid



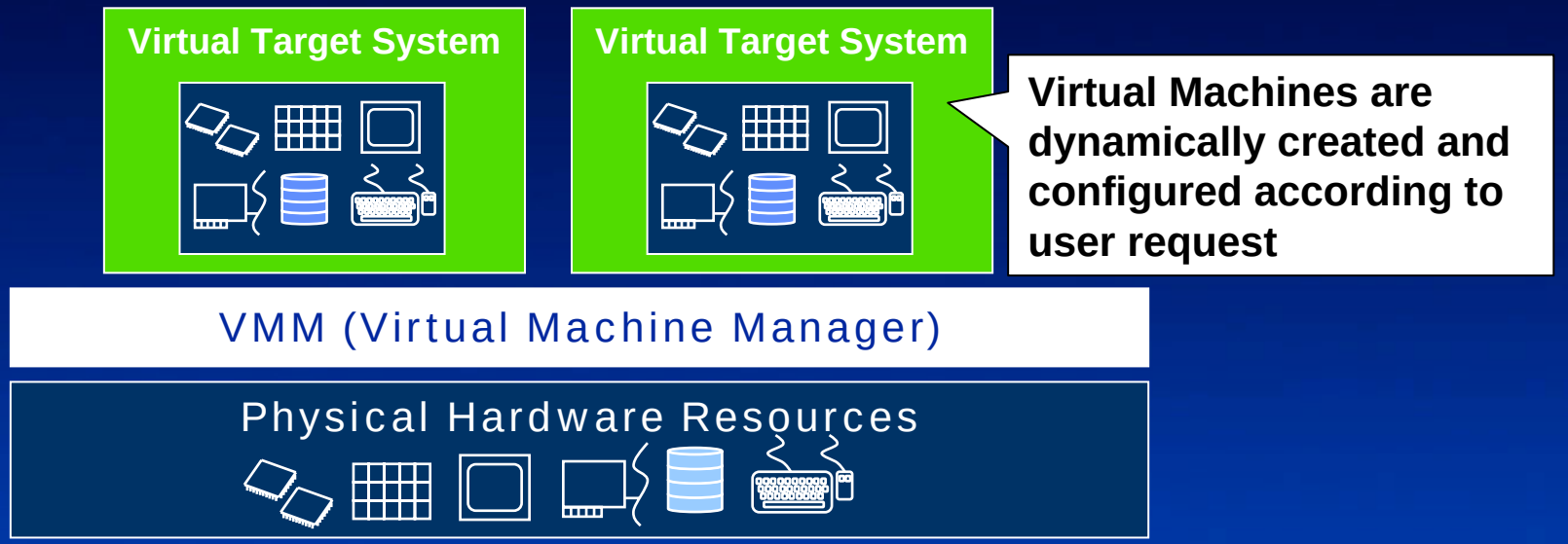
- Start Target System Interface
- Define Static Properties and Incarnation Rules
- Start Service Container

Alternative setup



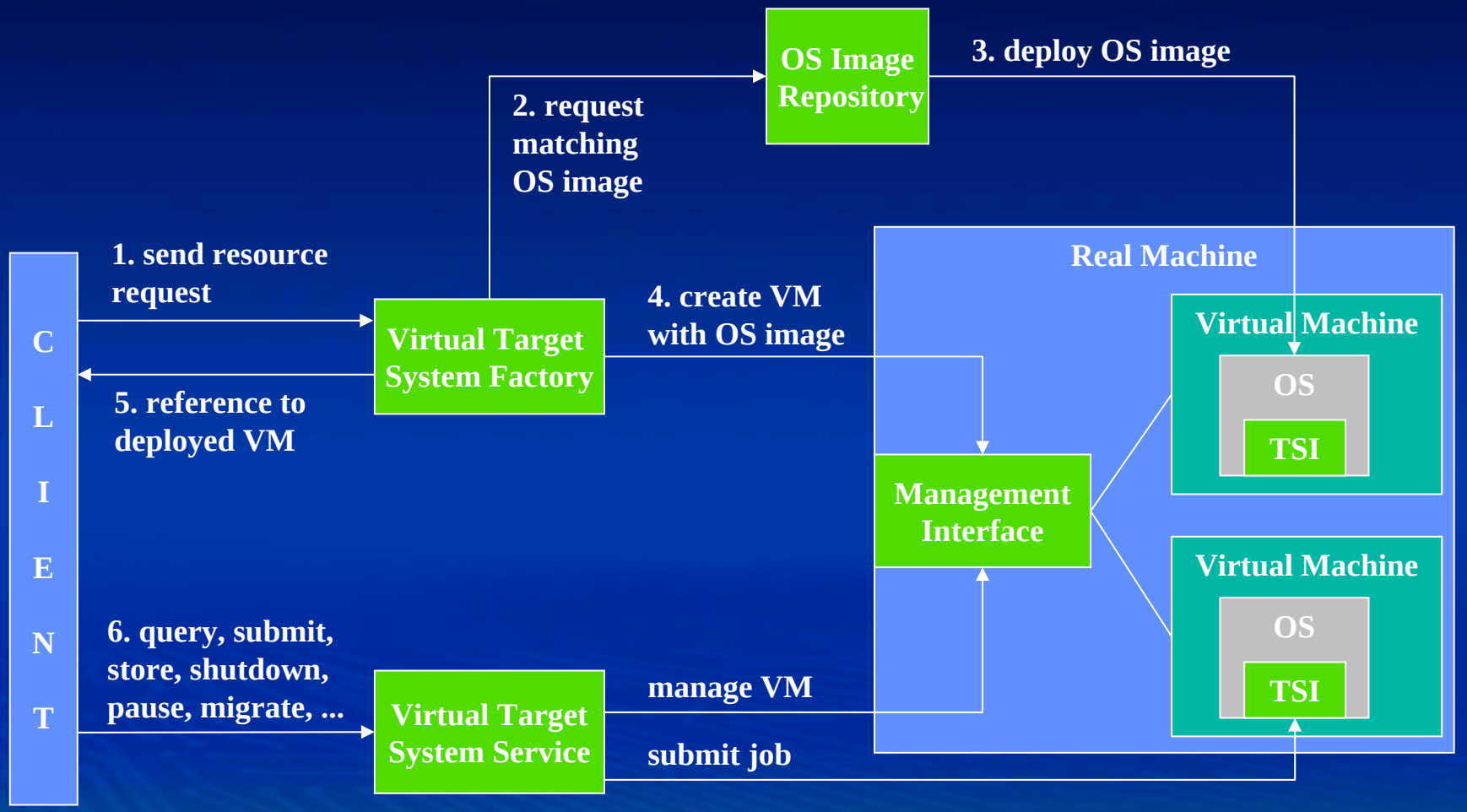
- Start Target System Interface
- Define Static Properties and Incarnation Rules
- Invoke Target System Factory in service container to add new target system

Outlook: Virtualization in Grid Computing



- Security: Protection of sensitive user data
- Reliability
  - Other partitions on the same machine will remain unaffected if one partition crashes
  - Virtual machines can migrate during run-time
- „Configurability“
  - Current model: Static OS and applications
  - With virtualization: Dynamically deployed OS images and applications on user request

Virtualization Architecture in GPE



What is the current state and what are the next steps?

GPE Alpha Release Available for Download

Available at UNICORE SourceForge project

- **http://unicore.sourceforge.net**

Application Client

- Lightweight client to load and run one application at a time

WSRF Hosting Environment

- Based on Axis 1.2 (RC3)
- Deploy services to Tomcat or run standalone server application
- Complete WSRF implementation

Example GridBeans

- Use source code as template for your own implementation

Admin Client

- Graphical administration interface
- Embedded standalone server

Complete Atomic Service Implementation

- File transfers based on SOAP with attachements
- „UNICORE-style“ Java TSI as execution back-end
- Runs on Windows and Linux/Unix

NEW: GPE4GTK!

- New SourceForge project
  - https://sourceforge.net/projects/gpe4gtk/
- Use Globus GASS server for file transfers
  - http/https transfers to work with firewall restrictions
- GridFTP for efficient file transfers
  - Needs opens port range in firewalls
- Includes Expert Client and BPEL Workflow engine
- UnigridsGS port running

Summary

Intel GPE…

...enables applications to run on and across different Grid infrastructures including UNICORE/GS and GTK

...provides a client framework to give users access to the infrastructure

...provides the GridBean concept and a programming API for Grid developers

...will support future virtualization and management concepts

...is available under BSD license

Thank you !