UNICORE
Programming Client Plug-ins

Grid Summer School, July28, 2004

Ralf Ratering

Intel

Parallel and Distributed Solutions Division (PDSD)

Overview

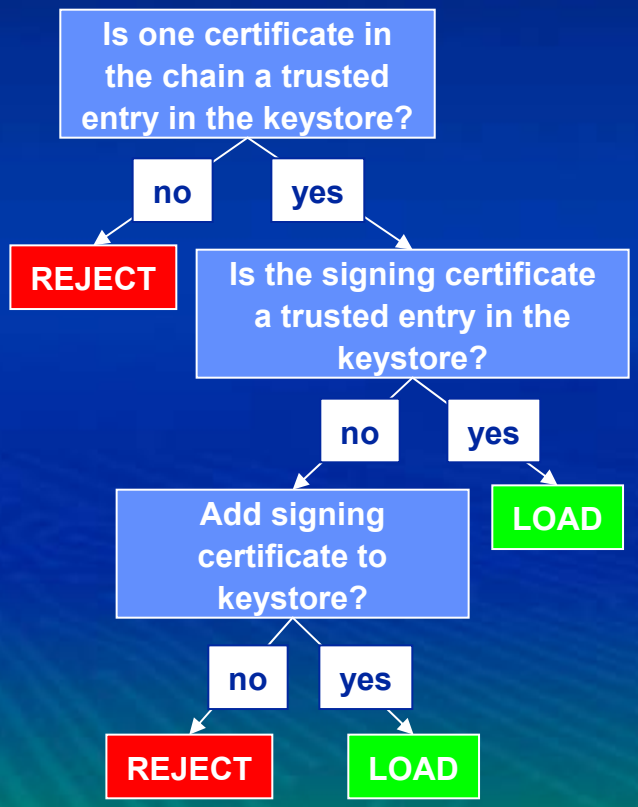
- Client Plug-in Concept
- Existing Plug-ins
- Programming techniques
- Writing your own plug-in

Plug-In Concept

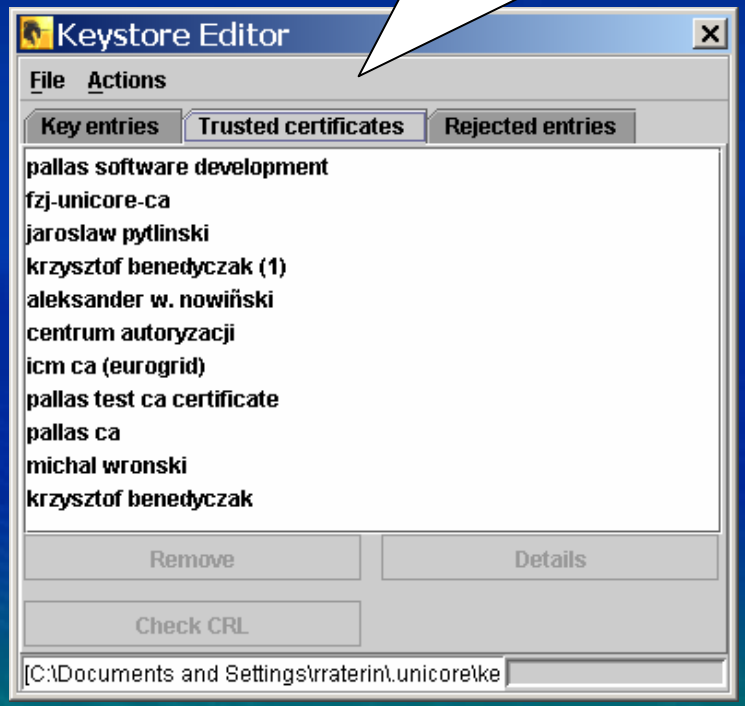
- Add your own functionality to the Client!
  - Heavily used in research projects all over the world
  - More than 20 plug-ins already exist
- No changes to basic Client Software needed
- Plug-Ins are written in Java
- Distribution as signed Jar Archives

Using 3rd Party Plug-ins

- Get plug-in jar file from web-site, email, CD-ROM, etc.
- Store it in client's plug-in directory
- Client will check plug-in signature



Import plug-in certificates from the actions menu in the keystore editor



Task Plug-ins

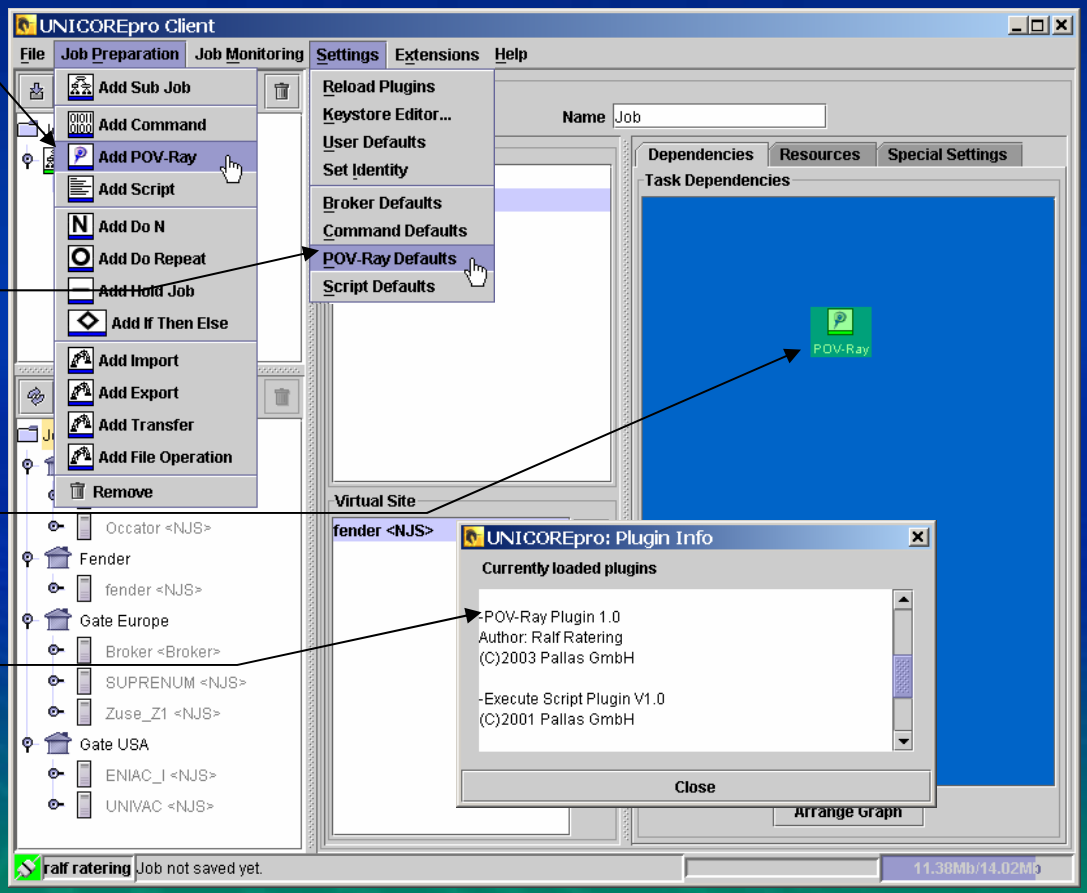
- Add a new type of task to the client GUI
- New task can be integrated into complex jobs
- Application support: CPMD, Fluent, Gaussian, etc.

Add task item

Settings item

Icon

Plugin info



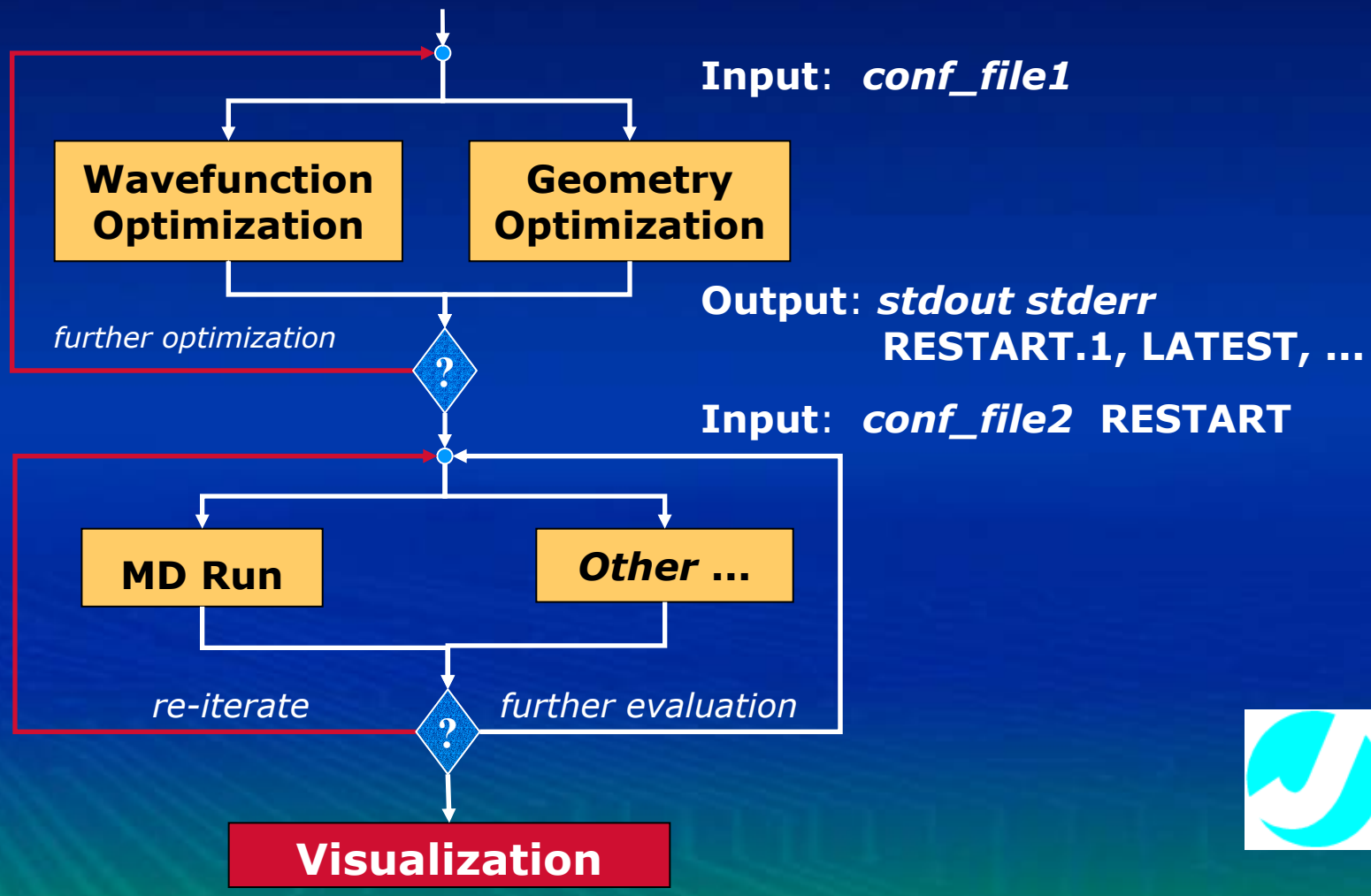
Supporting an application at a site

- Install the application itself
- Add entry to the Incarnation Database (IDB)

```
APPLICATION Boltzmann 1.0
Description „Boltzmann Simulation“
INVOCATION [
      /usr/local/boltzmann/bin/linuxExec.bin
]
END
```

Plug-In Example: CPMD

- **Workflow for Car–Parrinello molecular dynamics code**



Plug-In Example: CPMD

- CPMD plugin constructs UNICORE workflow

UNICORE 4.0 build 8

File Job Preparation Job Monitoring Settings Extensions Help

Job Preparation
- cpmd_job1 [13:31:33 11/13/2002]
  - run_wf_optim
  - if_okay
    - Then
      - run_mdrun
    - Else
      - save_files

Job Monitoring
- DWD
- FZ Juelich 4.0
  - SV1 <NJS>
  - t3e-1200 <NJS>
  - t3e-600 <...
- FZ-Juelich
- LRZ
  - linux <NJS>

Execute CPMD

Name: run_wf_optim

Pseudopotential Library: /usr/local/lib/PP_LIBRARY

Library Source: Root

CPMD Version: CPMD V3.4.1

Visualization: ENERGIES MOVIE

Browse library ... | System library | CPMD Wizard

CPMD Editor | Task Options | File Imports | File Exports

File Edit

Edit the CPMD input file

```
&CPMD
  OPTIMIZE WAVEFUNCTION
&END

&SYSTEM
 SYMMETRY
  1
 CELL
  20.52 1.0   1.0   0.0 0.0 0.0
 CUTOFF
  12.
&END

&ATOMS
SILICON
*SI_SGS KLEINMAN-BYLANDER RAGGIO=1.7
LMAX=P
 64
   .0000   .0000   .0000   1   1
   .0000  5.1300  5.1300   1   1
  5.1300   .0000  5.1300   1   1
```

valentina huber's leibniz-rechenzentrum (lrz) id | C:\unicore\jobs\job2.ajo | 57.44Mb/63.56Mb

Plug-In Example: CPMD

- CPMD wizard assists in setting up the input parameters

CPMD Wizard 2.5.0 [C:\cpmd\cpmd_examples\Si64\si64.xml]

File Help

CPMD Further | System Main | K-points | System further | PIMD | Atoms | DFT | Properties

CPMD Main | Optimization | Molecular Dynamics | Diagonalization | Free Energy | Mixing | Restart/Save

Atom Types

*SI_SGS KLEINMAN-BYLANDER RAGGIO=1.

Add ... | Edit ... | Remove

Change masses of atoms ...

Change charge of atoms ...

Assign special movie type ...

Change number of generator atoms ...

Change bonds ...

Set initial atomic velocities ...

Fix center of mass

Fix atoms ...

Fix coordinates ...

Penalty

Penalty weight factor for

streches:

bends:

torsions:

Add ... | Edit ... | Remove

Add ... | Edit ... | Remove

Shove

Apply | Reset | Default

Atom type

Pseudopotential: SI_SGS

Ionic charge distribution: 1.7

Atom coordinates

Load ... Number of atoms: 64

X-coord.	Y-coord	Z-coord.
0	0	0
0	5.13	5.13
5.13	0	5.13
5.13	5.13	0
2.565	2.565	2.565
2.565	7.695	7.695
7.695	2.565	7.695
7.695	7.695	2.565
0	0	10.26

The coordinates of the atoms (in cartesian coordinates).

Nonlocal parts of pseudopotential

Method for nonlocal parts of pseudopotential

Gauss-Hermit Kleinman-Bylander

Nonlinear core correction

Only VDB potentials

Binary output New version of atomic code

Ask for external polarization of localized charge

Pseudopotential nonlocality

Maximum quantum number: S P D

Local potential: S P D

al potential: S P D

Atomic basis

min. SLATER SLATER Pseudo atomic orbitals Gaussian basis functions Numerical functions

Number of L-values: 0 Occupation

l-value

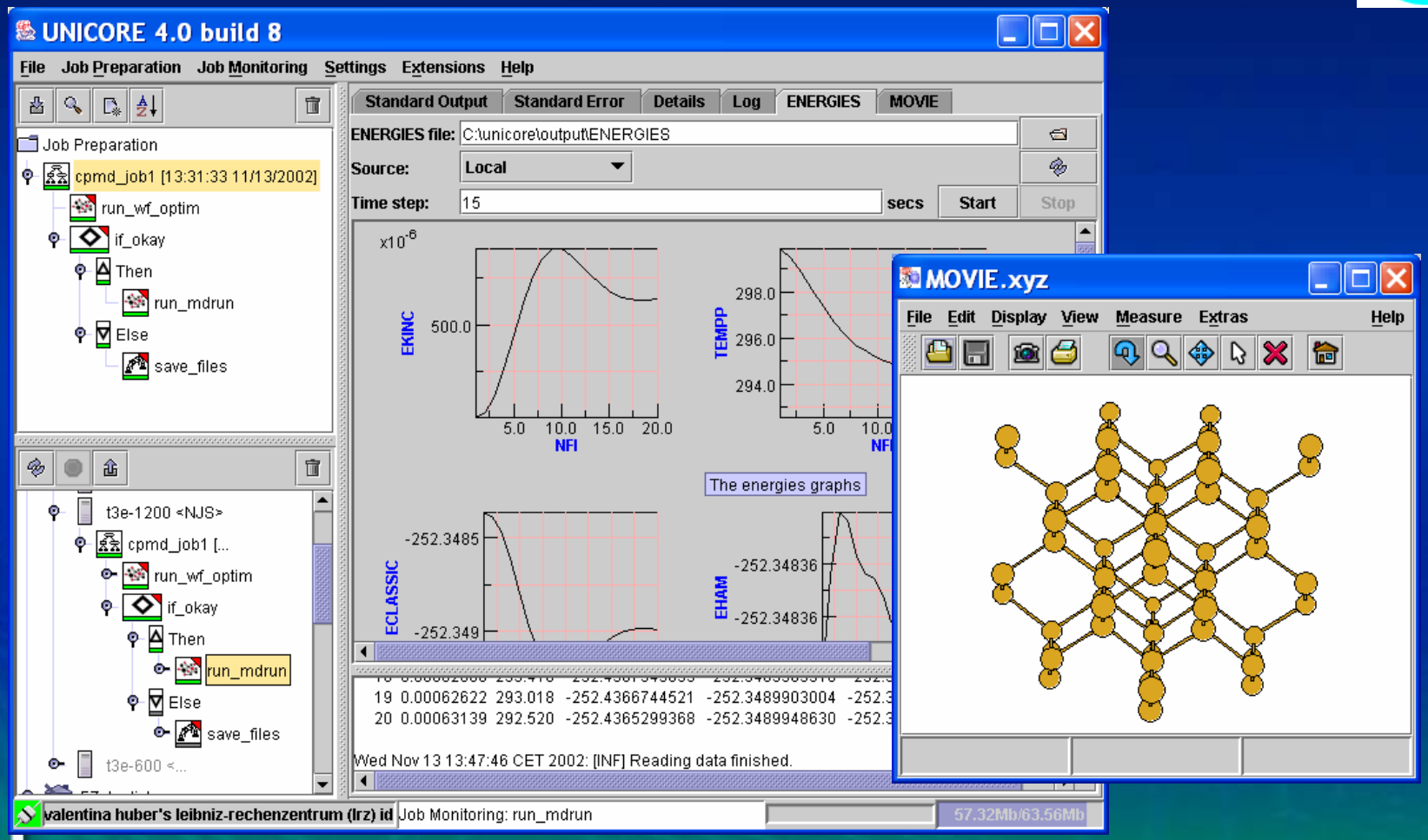
Filename:

Format: 1

OK | Cancel | Default

Plug-In Example: CPMD

- Visualize results



Extension Plug-ins

- Add any other functionality
- Resource Broker, Interactive Access, etc.

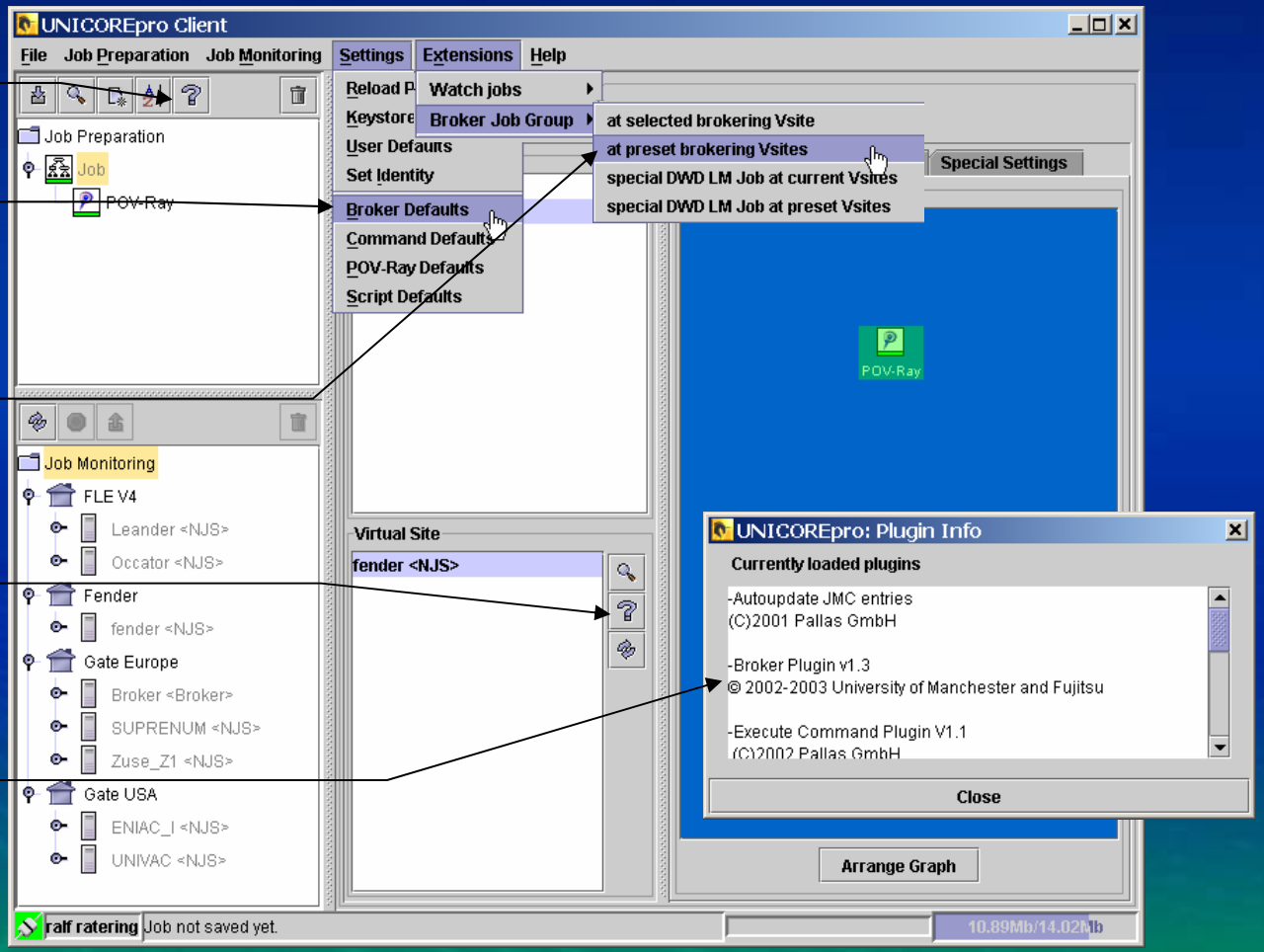
JPA toolbar

Settings item

Extensions menu

Virtual site toolbar

Plugin info



Plug-In Example: Resource Broker

- Specify resource requests in your job
- Submit it to a broker site
- Get back offers from broker

UNICORE 4.1

File Job Preparation Job Monitoring Settings Extensions Help

Job Group

Name SubJob_Job

Unicore Site

Resources Special Settings

Offers from Broker

Please select a single offer, and click OK

Virtual Site	Unicore Site	Start Time	End Time	Cost
T3E_turing	Manchester Univer...	1:49 PM to 2:19 PM	1:49 PM to 2:19 PM	1.00741E-4 CSAR ...
T3E_turing	Manchester Univer...	1:49 PM to 2:19 PM	1:50 PM to 2:20 PM	1.00741E-4 CSAR ...
O2000_fermat	Manchester Univer...	1:49 PM to 2:49 PM	1:49 PM to 2:49 PM	1.61875E-4 CSAR ...
O2000_fermat	Manchester Univer...	1:49 PM to 2:49 PM	1:49 PM to 2:49 PM	1.61875E-4 CSAR ...
O300_wren	Manchester Univer...	1:49 PM to 1:49 PM	1:49 PM to 1:49 PM	2.06475E-4 CSAR ...
O300_wren	Manchester Univer...	1:49 PM to 1:49 PM	1:49 PM to 1:50 PM	2.06475E-4 CSAR ...

Tickets in detail

The tickets contain the following:

Ticket for Task Script1
Ticket ID: org.unicore.AAIdentifier@2bbe9ee4
Ticket valid until: Tue Jul 08 14:49:23 BST 2003

ResourceSet:
Node (Number of Nodes) = 1.0 Nodes
Processor (Number of PEs per Node) = 1.0 Processors per node
Memory (Total Amount of Memory) = 20.0 Megabytes per node
RunTime (Time per Job) = 15.0 Seconds
org.unicore.resources.QoSCheck: QoSCheck (QoSCheck)

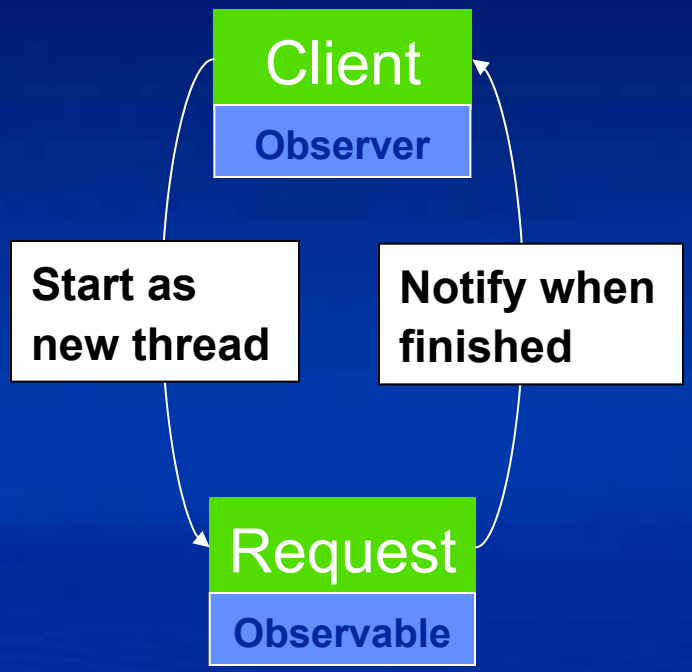
Existing Plug-Ins (incomplete)

- **CPMD (FZ Jülich)**
- **Gaussian (ICM Warsaw)**
- **Amber (ICM Warsaw)**
- **Visualizer (ICM Warsaw)**
- **SQL Database Access (ICM Warsaw)**
- **PDB Search (ICM Warsaw)**
- **Nastran (University of Karlsruhe)**
- **Fluent (University of Karlsruhe)**
- **Star-CD (University of Karlsruhe)**
- **Dyna 3D (T-Systems Germany)**
- **Local Weather Model (DWD)**
- **POV-Ray (Pallas GmbH)**
- **...**

- Resource Broker (University of Manchester)
- Interactive Access (Parallab Norway)
- Billing (T-Systems Germany)
- Application Coupling (IDRIS France)
- Plugin Installer (ICM Warsaw)
- Auto Update (Pallas GmbH)
- ...

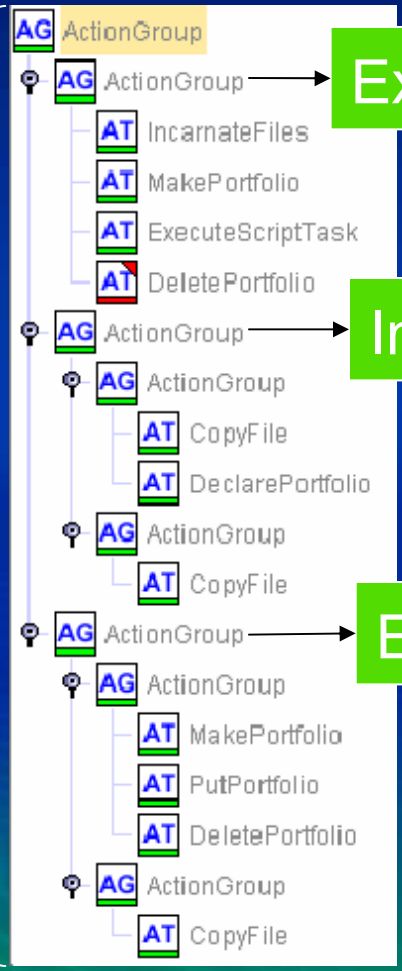
Plug-in Programming: Requests

- GetFilesFromUSpace
- SendFilesToUspace
- GetFilesFromXSpace
- SendFilesToXSpace
- GetByteArrayFromXSpace
- SendByteArrayToXSpace
- GetListings
- GetUsites
- GetVsites
- GetResources
- GetRunningJobs
- GetJobStatus
- GetOutcome
- GetSpooledFiles
- ...

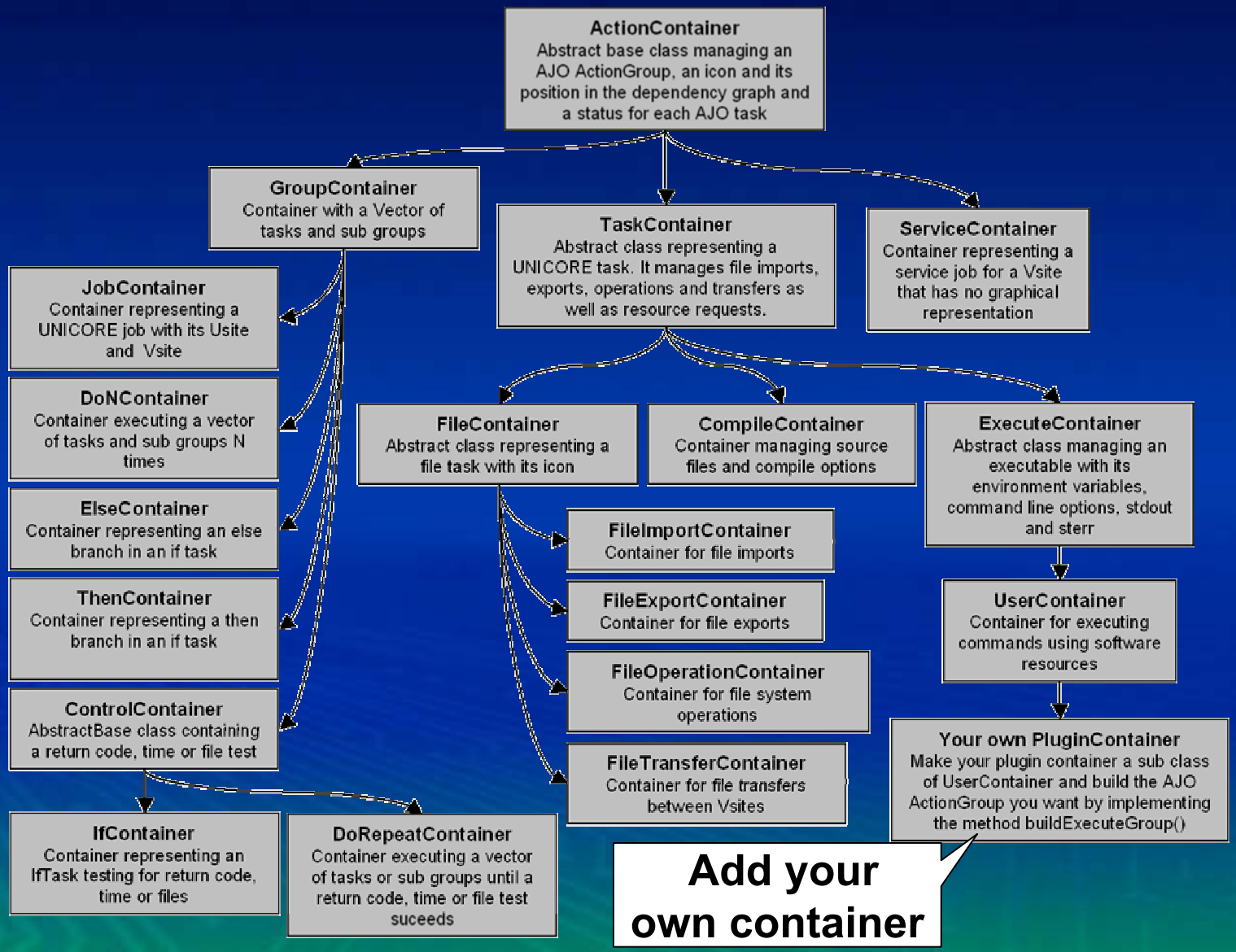


Plug-in Programming: AJOs and Containers

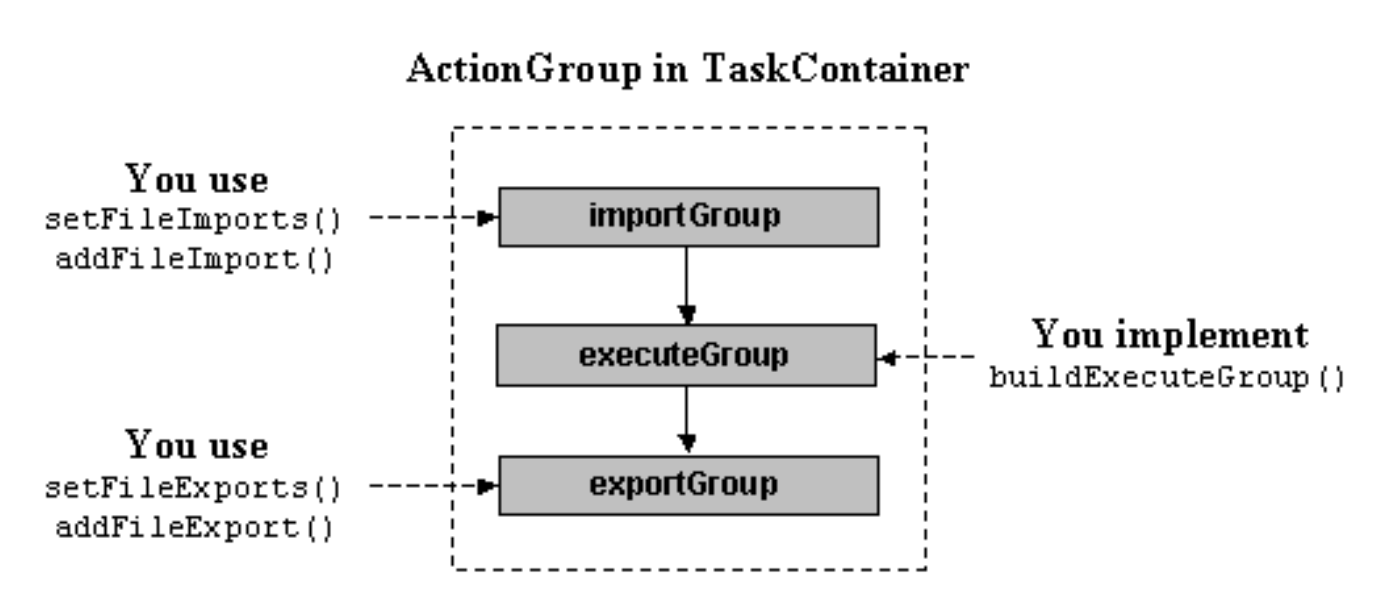
- Client containers encapsulate complex AJOs
- Manage imports, exports and execution
- Hold parameters, keep status, check errors



Client Container Hierarchy



Implementing the Plug-in Container



Write your first plug-in…

Tools you need...

- Java SDK
  - javac, jarsigner, jar
- Ant
- Eclipse
  - Client Source Code
  - AJO Source Code
- Documentation
  - Plug-in Programmer‘s Guide

Add a new application at the virtual site

- Add HelloWorld application to the Incarnation Database (IDB)

```
APPLICATION HelloWorld 1.0
Description „Demo Resource for GridSchool“
INVOCATION [
        /usr/local/unicore/test/helloWorld.sh
]
END
```

Incarnation Data Base

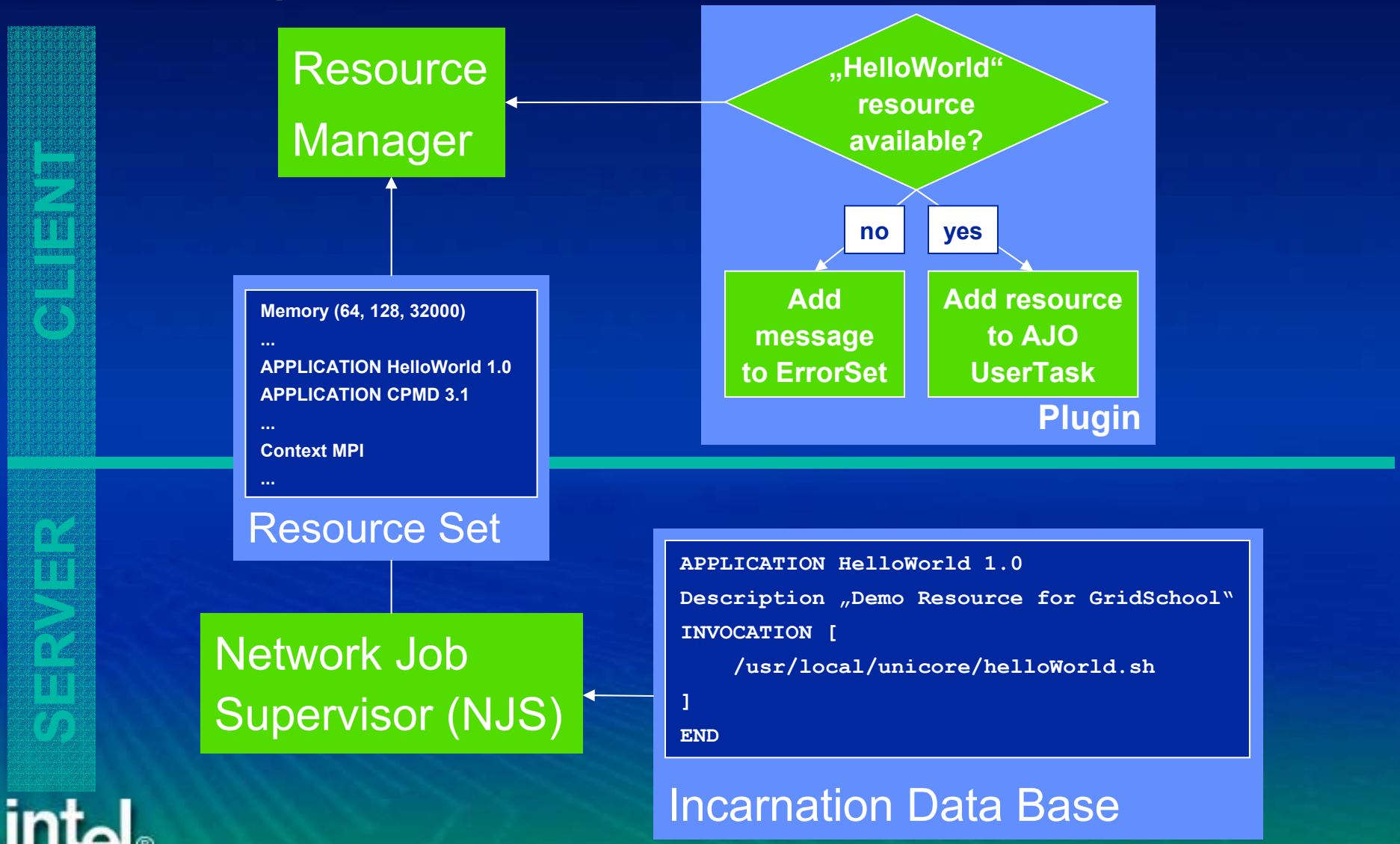
Can be executed with a Command Task!

Disadvantages of Command task

- Input file has to be edited outside Client
- Imports and Exports have to be specified manually
- No integrated GUI for parameters
- Results have to be visualized outside client
- No additional functionality possible
  - sample files
  - application steering

→ **Use a specialized Plug-in Task!**

Running Application Resources from a Plug-in

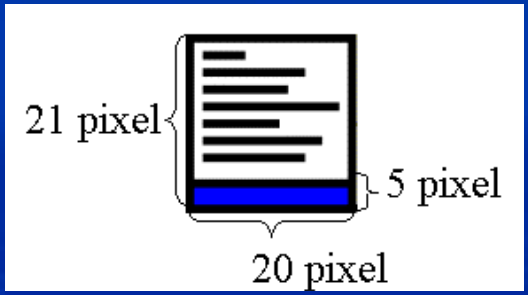


The Hello Plug-in

- Task Plugin
  - Add new type of task „Hello“
- Implemented Classes
  - Main plugin class
  - Plug-in container
  - JPA panel

The Main Plug-in Class

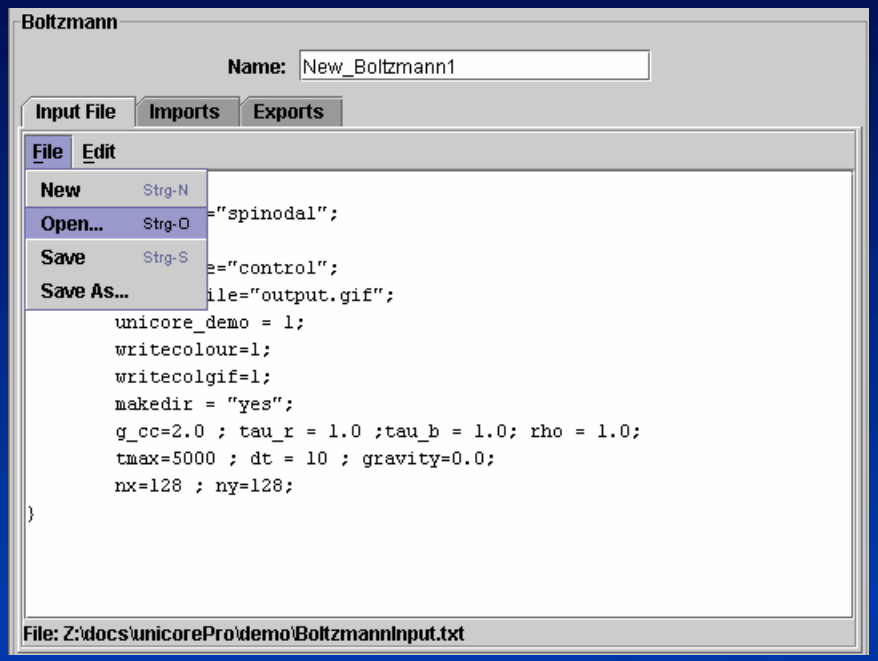
- Start and stop plug-in
- Provide task icon and other resources
- Create containers and panels

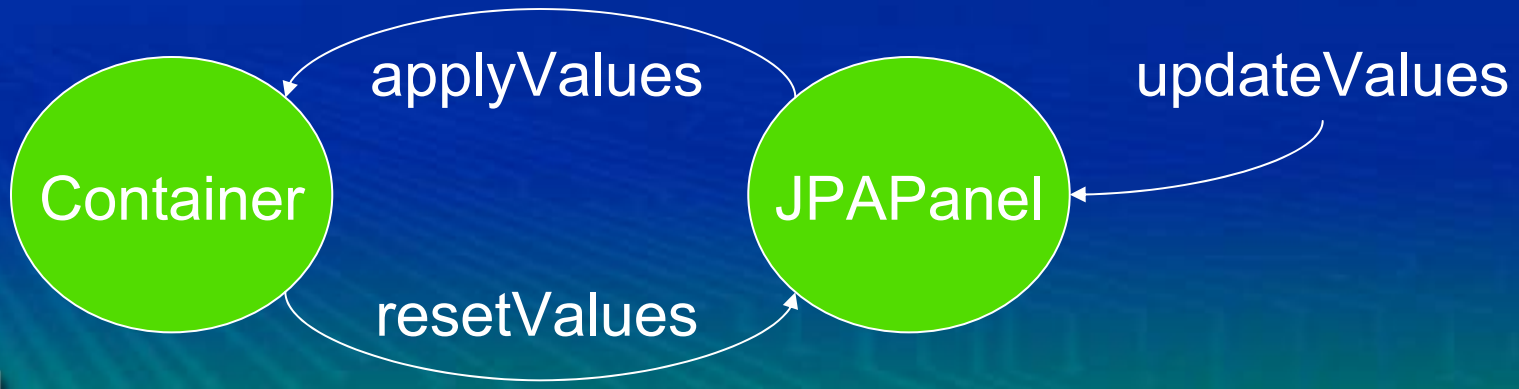


Icon Format

Class JPAPanel

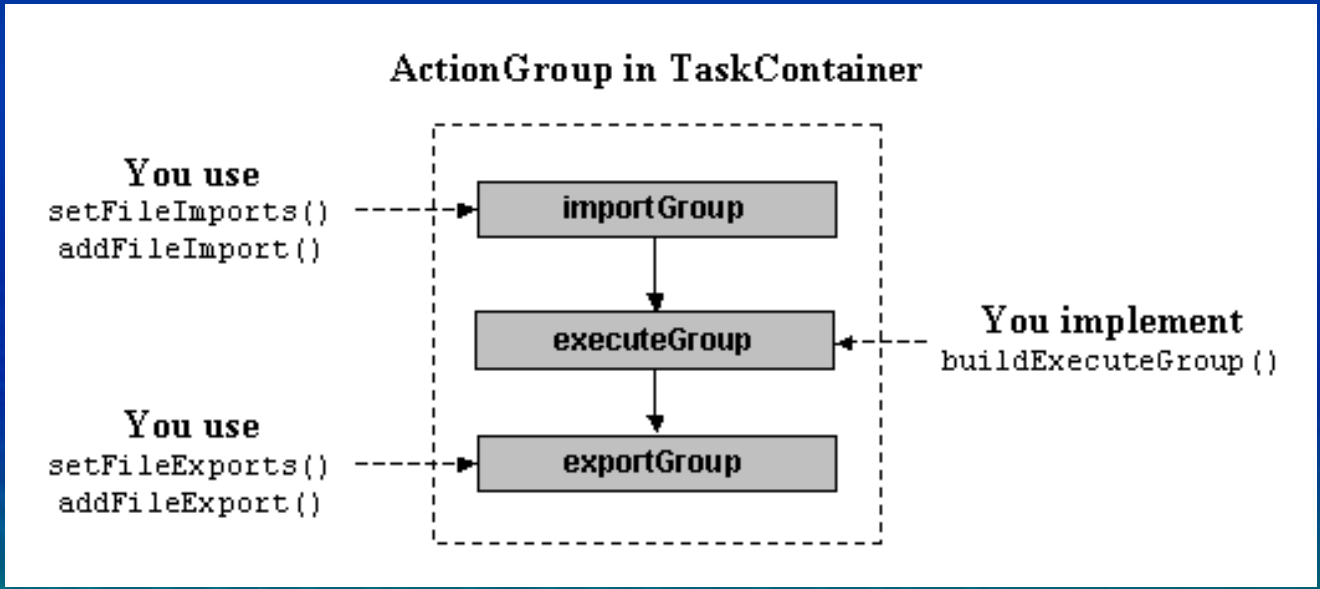
- Set parameters in container
- May use RemoteTextEditor, ImportPanel and ExportPanel
- Implements interface Applyable





Container Class

- Encapsulate AJO
- Inherit from UserContainer to execute application resources
- Implement method buildExecuteGroup()



Exercise 1: Build and use a plug-in

- Running target „all“ in Ant script will do the work for you:
  - Compile the source code
  - Build a jar archive from class files
    - `jar cvf helloPlugin.jar org/gridschool/unicore/plugins/hello`
  - Sign the archive with your user certificate
    - `jarsigner -keystore c:/Documents and Settings/rraterin/.unicore/keystore -storepass abcdefg helloPlugin.jar „gridschool plugin signer"`
- Specify plug-in jar directory in Client UserDefaults
- Run a job containing a Hello task

Exercise 2: Add file imports and exports

- Add an import and an export panel to the JPAPanel

Remove Import

New Import

Browse file systems

File Imports

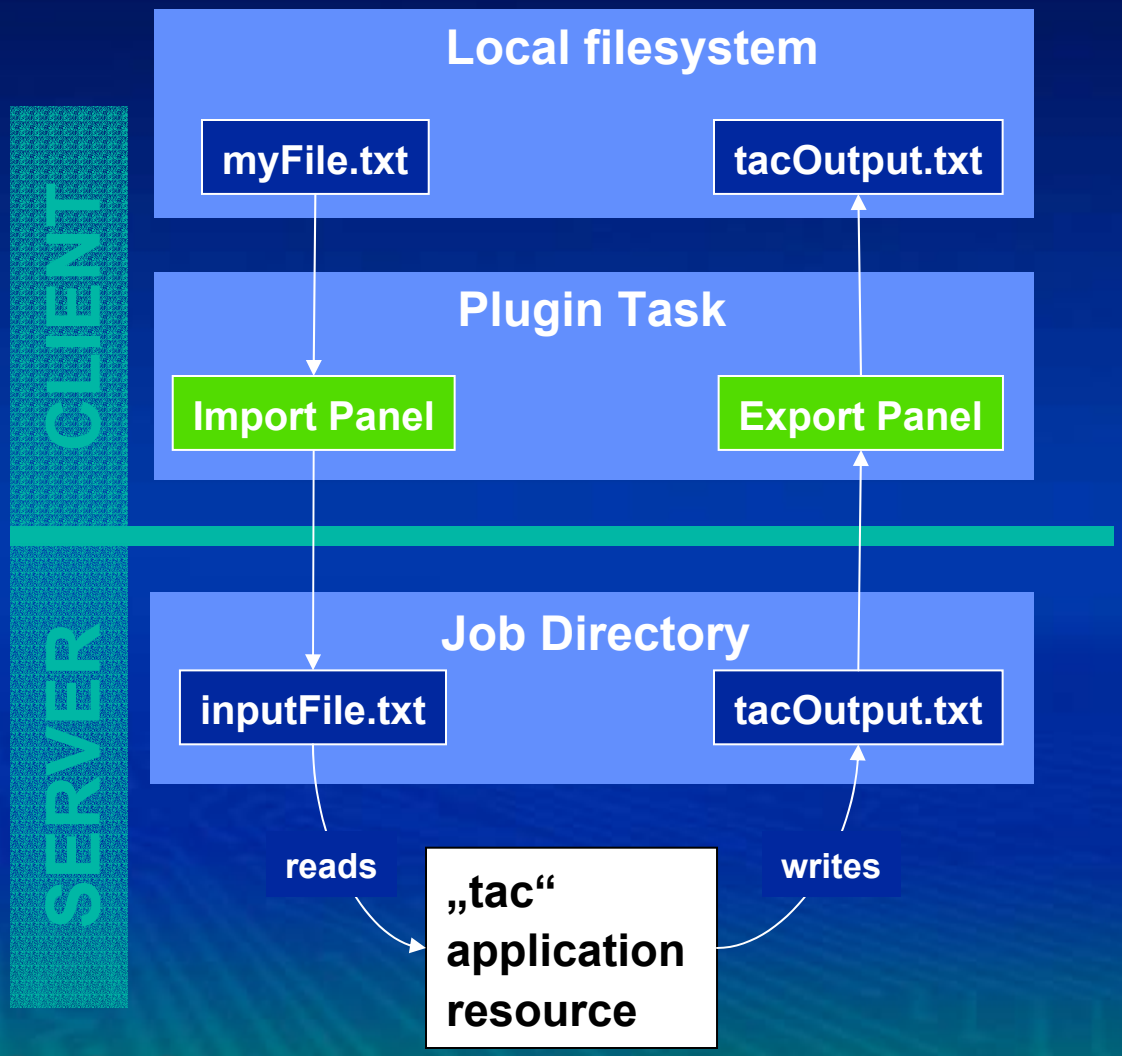
Source	File at Source	File in Job Directory	Overwrite File(s)	Binary
Root	usr/share/info/find.info.gz	find.info.gz	✔	✔
Home	localOutputfile	localOutputfile	✔	✔
Local	C:\tmp\dateLoop.sh	dateLoop.sh	✔	✔

File Exports

File in Job Directory	Destination	File at Destination	Overwrite File(s)	Binary
output.gif	Home	Documents/	✔	✔
outputfile	Local	C:\tmp\outputfile	✔	✔

Use PovRayJPAPanel as reference!

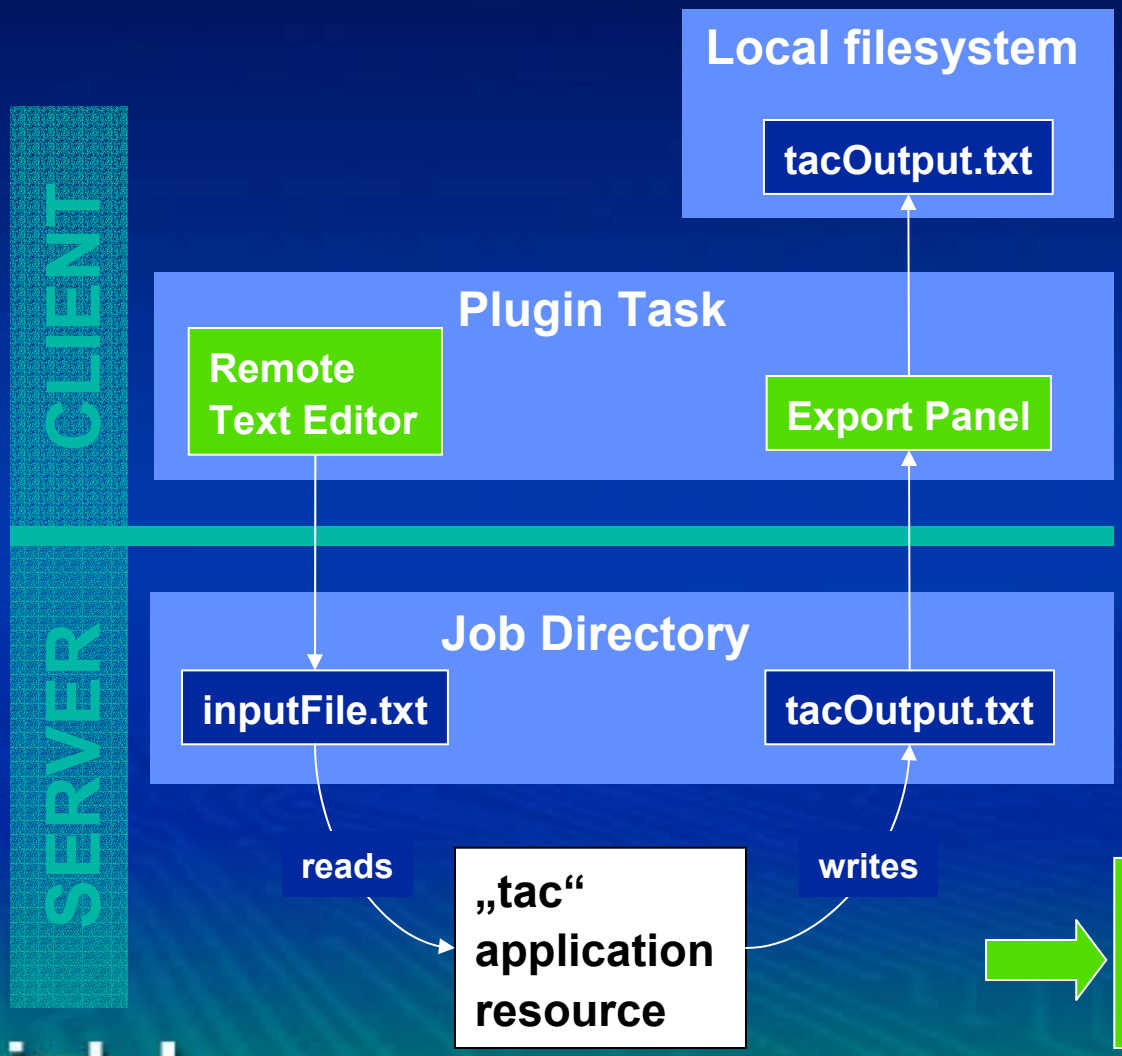
Exercise 3: Run the „tac“ application



- Execute “tac” instead of “HelloWorld” resource
- Specify input file and output file manually in import and export panel

```
APPLICATION tac
Description „tac“
INVOCATION [
  tac inputFile.txt
    > tacOutput.txt
]
END
```

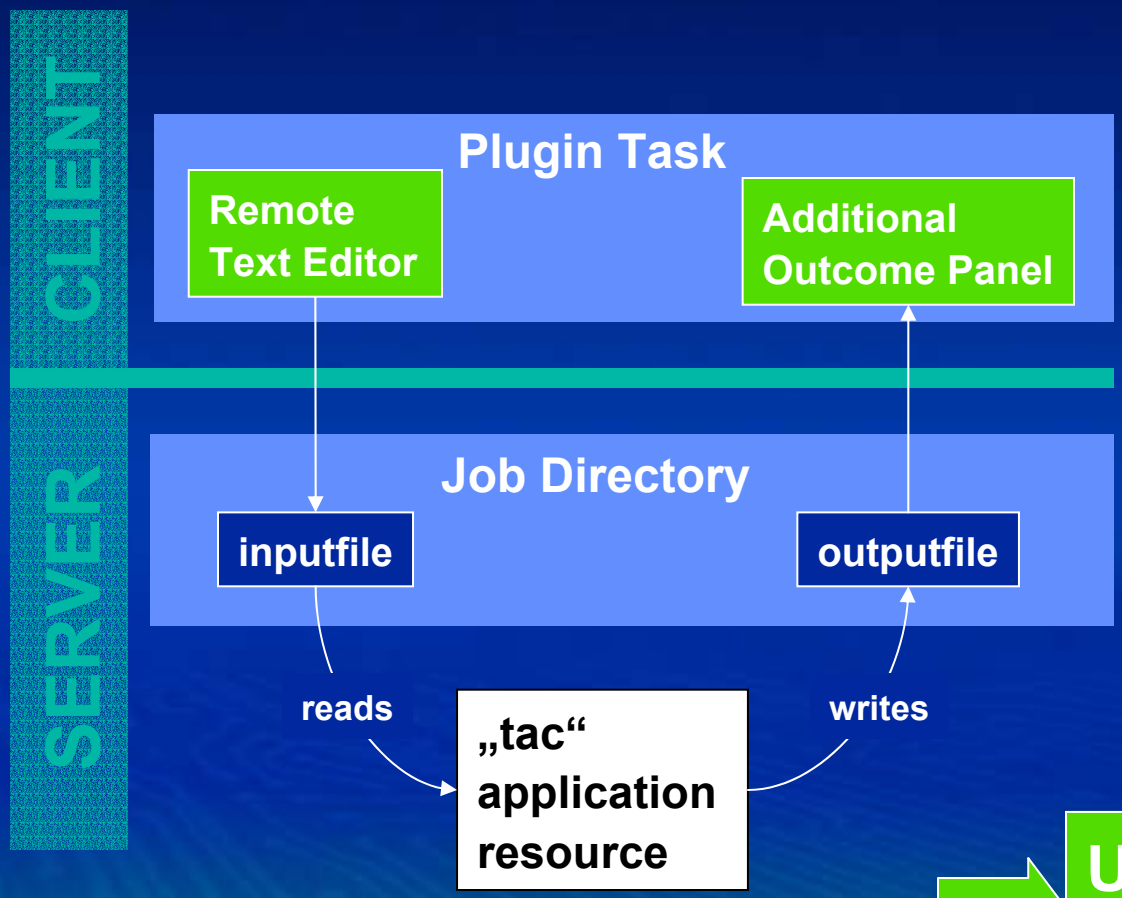
Exercise 4: Use Remote Text Editor to specify input file for „tac“



- **Remote Text Editor allows to edit files from remote and local file systems**

Use PovRayJPAPanel as reference!

Exercise 5: Display „Reverse" output in additional plug-in outcome panel



- Implement interface IPanelProvider in container
- Display output file in text area

Use PovRayContainer as reference!