



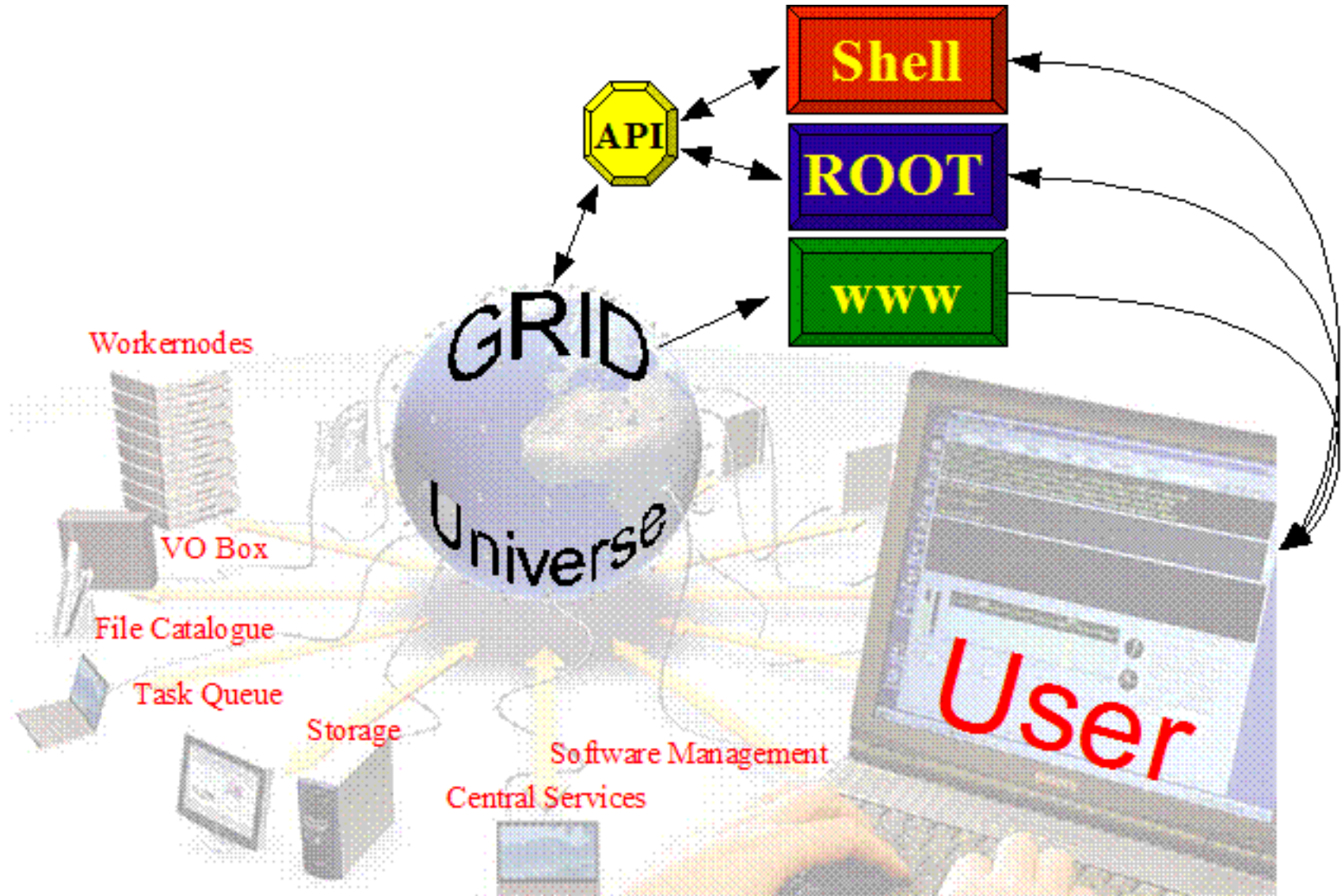
User Interaction with the GRID **API Service & aliensh**

Andreas-Joachim Peters IT-DM-SMD

09.04.2008



API Service for GRID access





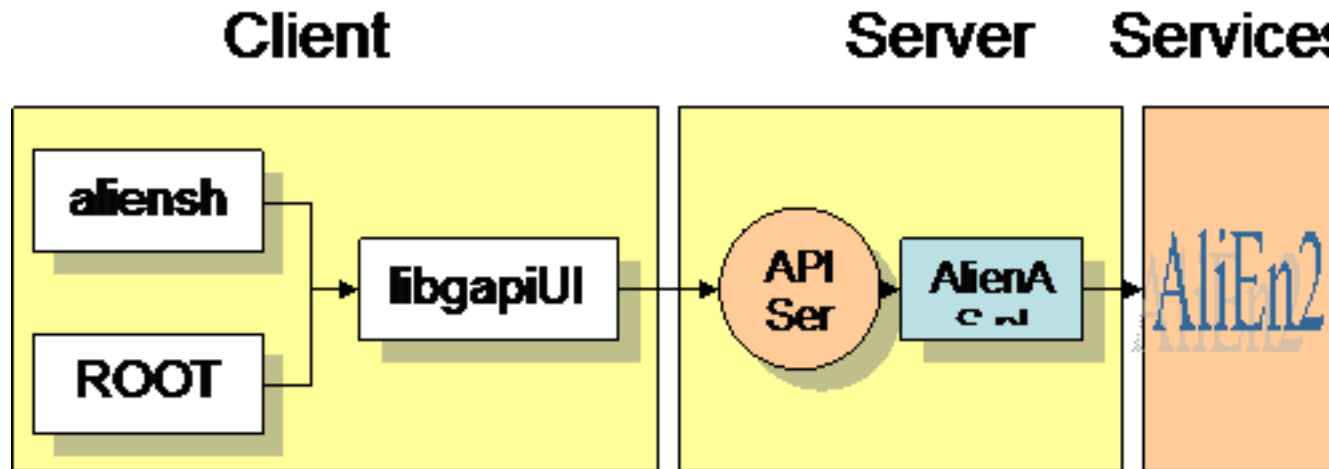
Why API Client/Server Architecture

- Need **unified high performance access to GRID services & databases** (FC & TQ) with GSI authentication
 - Single user entrance point to GRID services
 - Avoid direct client connections to databases
 - CLI (**shell**)
 - need to use session model to avoid authentication for every shell command
 - API library for application access (**ROOT**)



API Client Interfaces

- **aliensh**
 - Interactive shell for FC and TQ commands
- **TAlien**
 - ROOT interface for FC and TQ commands



Both interfaces use libgapiUI client library to communicate with API Service



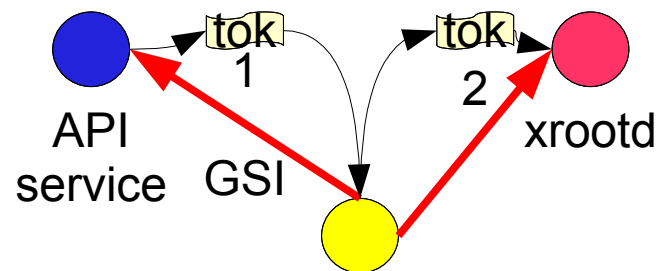
How does it work?

- API client & server is based on **SOAP protocol**
- **Session Model**
 - User authenticates on **GSI/SSL** connection with GRID certificate (port 8002/10001)
 - User obtains **symmetric key over SSL/GSI** connection
 - Concurrent commands use a dynamic **symmetric key for message level encryption** for standard SOAP calls
- **C++ client library** – C++ server with PERL interpreter for AliEn code



How fast is it?

- Some timings to get the taste
 - A **GSI** authentication takes ~**150-200 ms**
 - An API call executed from a shell takes ~**16 ms**
(+ processing on the server ~ 30 ms)
 - A **ROOT** client can open concurrently **9 files/s** via AliEn including the time to open the file via xrootd (30 ms API + 80 ms ROOT)



Client (TAlienFile/aliencp)

- A **single API server** executes upto **30 comands/s**



Client Installation

- Download the AliEn installer from:
- <http://alien.cern.ch/alien-installer>
- using a browser or wget. Set the permissions for execution of the installer:
- `chmod ugo+rx alien-installer`
- Run the installer:
- `./alien-installer`
- **Currently the client comes still precompiled with gcc 3.2.3**
- **If you have a different compiler in your default PATH you will be asked whenever you try to authenticate with 'alien-token-init' if you want to recompile the client**
...



Client Recompilation

- Execute '**alien-token-init**' to trigger **recompilation**
- Run the recompilation script **by hand** doing:

`/opt/alien/api/src/recompile.gapi`

- Pitfalls for recompilation
 - If you overinstall an existing AliEn installation with a new version **remove always old source tarballs** from /opt/alien/api/src (e.g. do `rm -rf /opt/alien/api/src` before you install the new version)
 - If you have a very **old version of autotools** in your PATH the recompilation might fail
 - If you compile ROOT, make sure that you **don't mix other C++ libraries** which have been compiled with a different compiler



Client Configuration

to use aliensh

- **Don't start to modify environment variables**
 - aliensh needs **no special environment variable** to be used (if yes, something is usually wrong in your installation)

```
[.....] /home/apeters > /opt/alien/api/bin/alien-token-init aliprod  
-----  
Setting central config:  
=====
```

```
export alien_API_SERVER_LIST="pcapiserv01.cern.ch:10000|pcapiserv02.cern.ch:10000|  
pcapiserv04.cern.ch:10000"  
export alien_API_PORT=10000  
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$GSHELL_ROOT/lib  
export TERMINFO=/usr/share/terminfo  
=====
```

```
=> Trying to connect to Server [0] http://pcapiserv04.cern.ch:10000 as User aliprod  
Your identity: aliprod  
Creating token ..... Done  
Your token is valid until: Thu Apr 10 00:12:44 2008
```



Client Configuration

to use aliensh

- The API service endpoints are **configured centrally**
 - currently **3 API servers** are available **for interactive users** + **3 API servers for GRID jobs**
- Typical pitfalls
 - Authentication fails if you are **not a registered user**
 - If you see many GSS errors (client side) a very common problem is a **clock skew** between your machine and the server
 - Try to sync your local clock (e.g. ntpdate/xntpd)
 - Many errors **can be** easily **understood on the server** but not on the client
 - Try to contact an admin to check the server log



Client Configuration

to use ROOT

- When you execute 'alien-token-init' a **bash shell script** is created which **defines needed environment variables** to use libgapiUI inside an application
- To use **TAlien in ROOT** do:
 - bash> **source /tmp/gclient_env_\$UID**
 - bash> **root -x**
 - root> **TGrid::Connect("alien://");**
 - Here we authenticate inside the application to the API server – the token is kept in memory and is different from the token used by aliensh!



Client Documentation

- You can find **documentation** about the **GRID API** client under:
- <http://alien.cern.ch/twiki/bin/view/AliEn/GAPI>
[http://project-arda-dev.web.cern.ch/project-arda-dev/alice/apiservice/guide/ALICE Analysis User Guide V1.0.htm](http://project-arda-dev.web.cern.ch/project-arda-dev/alice/apiservice/guide/ALICE%20Analysis%20User%20Guide%20V1.0.htm)



Recent Changes

- In AliEn v2.14 the **virtual memory consumption** has been drastically **reduced**
 - libgapiUI **used a preallocated resultmap** with a fixed number of rows and columns each command could return (e.g. `ls -la` returns 9 columns and one row per listed file)
 - this has been changed to a **minimal resultmap** which is dynamically reallocated if needed
 - on a 64 bit machine ~**450 MB** of **virtual memory** have been **saved**



Future Changes

in libgapiUI

- A small **change in the protocol** should be done to **allow** to use **DNS alias names** for the first server contact
 - Advantages
 - Several servers can be **load balanced via a DNS alias**
 - Currently servers are load balanced by the client random selection algorithm
 - **Configuration** can be **hidden** inside a DNS alias
 - No need anymore to download central configuration to each client – just use 'apiservice.cern.ch'



Future Changes

in the ROOT TAlien

- A **TAlienPackage** class exists (L.Jancurova)
 - Allows to setup any AliEn software package on your local computer and use the same package like in a GRID job
- An enhanced version of **TAlienJDL** exists which can read an AliEn JDL and will be able to execute your JDL with a local sandbox on your desktop
- Next ROOT release ?



Thanks for your attention!

Question or comments?