



# *Clarens Client and Server Applications*



**CHEP 2003 March 24 - 28,**

**Grid Architecture, Infrastructure & Middleware**

***Conrad Steenberg, Eric Aslakson, Julian Bunn, Harvey Newman***

California Institute of Technology



Developed as part of the  
**Particle Physics DataGrid**



# Overview



Web services: lots of hype very few applications

**However**, several web-services applications have been built on the Clarens web service architecture. This talk will describe some of these:

- Proxy escrow
- Web based Java/ Javascript client
- Access to JetMET data via SQL2Root
- Root client access to remote data files
- Access to files managed by the SDSC storage resource broker (SRB)



# Proxy Escrow



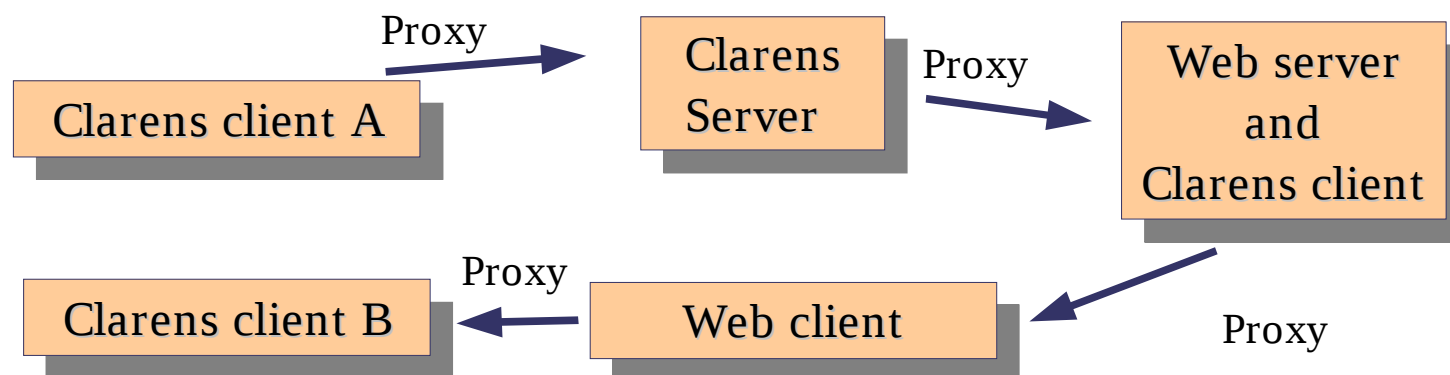
- Similar to MyProxy (<http://www.ncsa.uiuc.edu/Divisions/ACES/MyProxy/>)
- Eases administration of certificates and keys for users: credentials can be accessed using a browser
- Only need certificate distinguished name and password
- Some applications can not access a file system to read cert/ key files (proxy credentials), e.g. Java applet in browser
- Credentials are stored in a database by the Clarens server.  
Communication encryption via SSL limits pilfering by eavesdroppers
- Data is encrypted before being stored, no password stored
- Benefits from the Clarens access control infrastructure



# Proxy Escrow II



- API:  
`proxy.store(proxy, cert, password)`  
`proxy.retrieve(DN, password)`  
`proxy.delete(DN, password)`  
`proxy.delete_admin(DN)`  
`proxy.list()`
- The last two are limited to administrators via ACLs
- Chicken-and-egg problem: Need key/ cert to log into Clarens to retrieve key/ cert!
- Solved by implementing an intermediary for proxy retrieval:





# Java/ Javascript client



- First application to require proxy escrow
- Stand-alone Java client simple to implement – Java crypto API unwieldy though
- Browser-based applet solves software distribution/ installation problem
- BUT restrictions placed on applet:
  - Unable to access filesystem
  - Limited to certain APIs – not allowed to access JCE/ JSSE
  - Can only access server where applet was loaded from
- Use bouncycastle.org crypto API instead
- Use proxy escrow to access cert/ keyfile instead of from filesystem
- Use Javascript to manage web page contents presented to user



# Java/ Javascript client II



Store proxy:

Clarens Proxy upload - Mozilla (Build ID: 2003021212)

File Edit View Go Bookmarks Tools Window Help Debug QA

https://heppc22.

Home Bookmarks Members WebMail Connections BizJournal

## Proxy Upload

Store type:

Proxy or key file:

Certificate file:

Password:

Confirm password:

### Security note:

If you have JavaScript enabled, your browser might show a warning that a script wants to access the passwords in the above form. The script merely checks whether both passwords are filled in and matches.

You may disable JavaScript if you are concerned about trusting the script on this page.

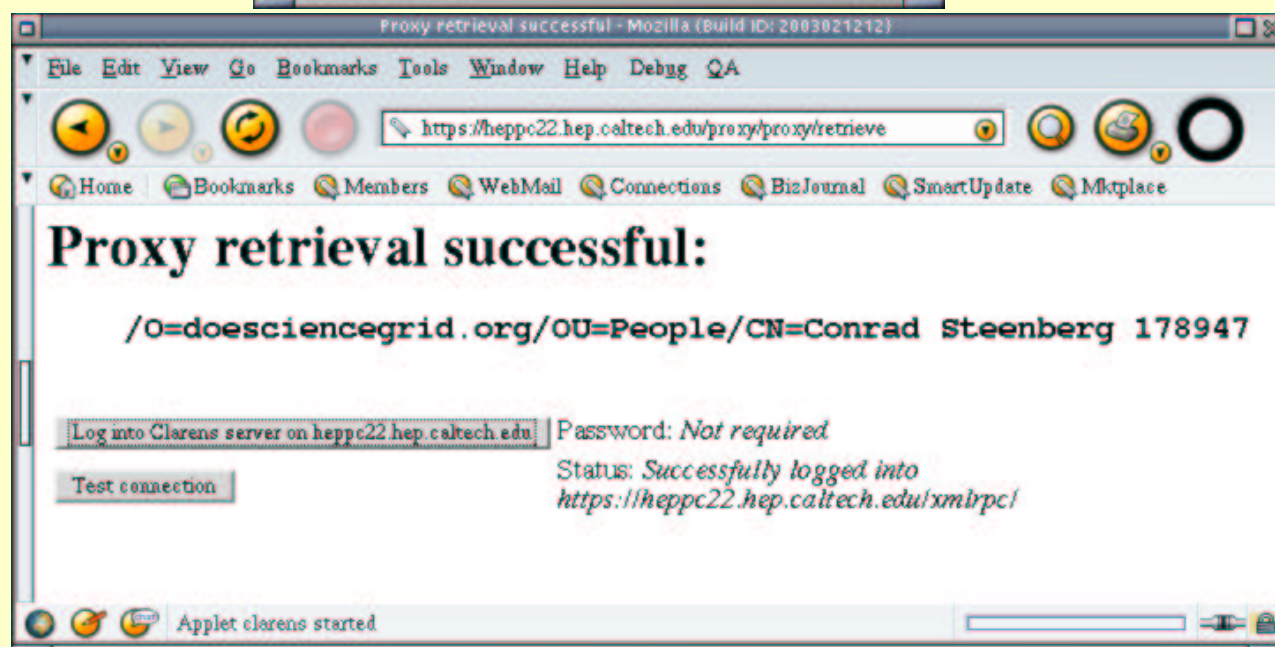
Done



# Java/ Javascript client III



Retrieve proxy:

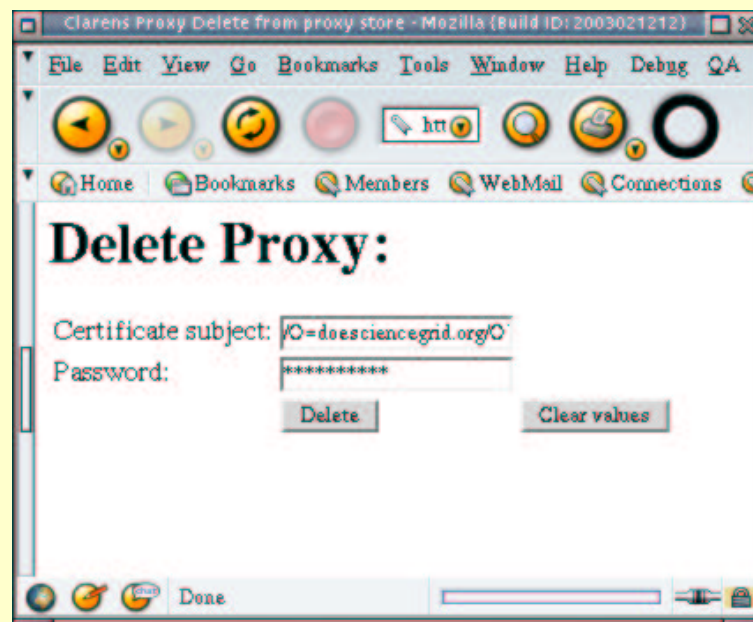




# Java/ Javascript client IV



Delete proxy:





# Access to CMS JETMet data



- Part of the CAIGEE project to store CMS analysis data in SQL databases (Oracle 9i and SQLServer)
- See <http://pcbunn.cithep.caltech.edu/GAE/GAE.htm>
- OO data extracted from Objectivity DB stored in RDBMS
- Clarens exposes API to pass data queries to SQL2Root backend
- Data can be accessed using Root Clarens client or downloaded to client machine
- API:  

```
write_tree(tablename, where_clause, filename)
```
- Method implemented in C++ , accessed via a Python wrapper



# Root client access

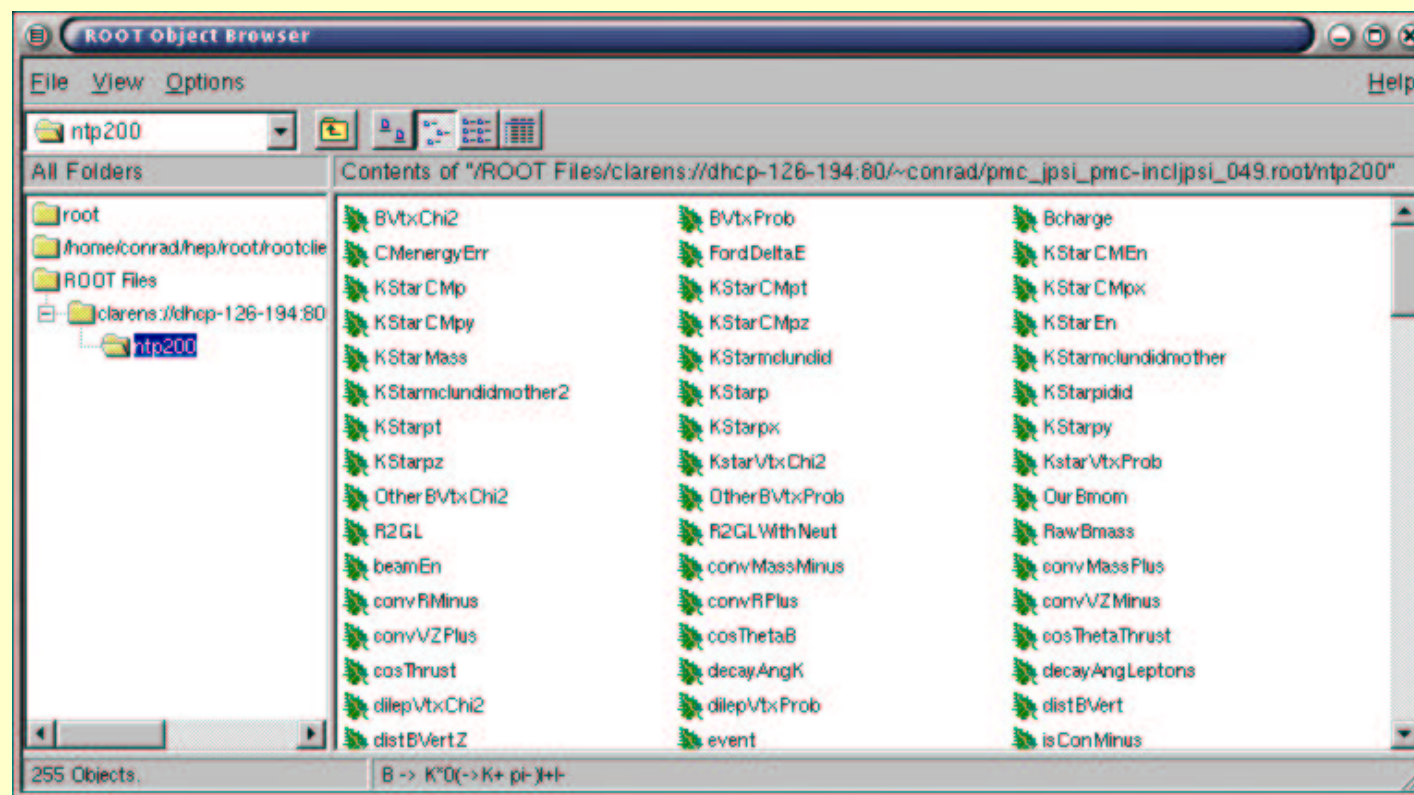


- Root analysis environment widely used in HENP

See: <http://root.cern.ch>

- Client written in C++ as a Root extension file
- Can access any Clarendon methods from command line
- Special support for remote file access via the **TCWebFile** class, accessible as a Root file similar to a local file
- Also **TCSysmDirectory** class for remote file browsing using the Root browser
- Client used for distributed analysis demonstrations at SC2002 and iGrid2002.

- Examining a remote file inside a Root object browser:

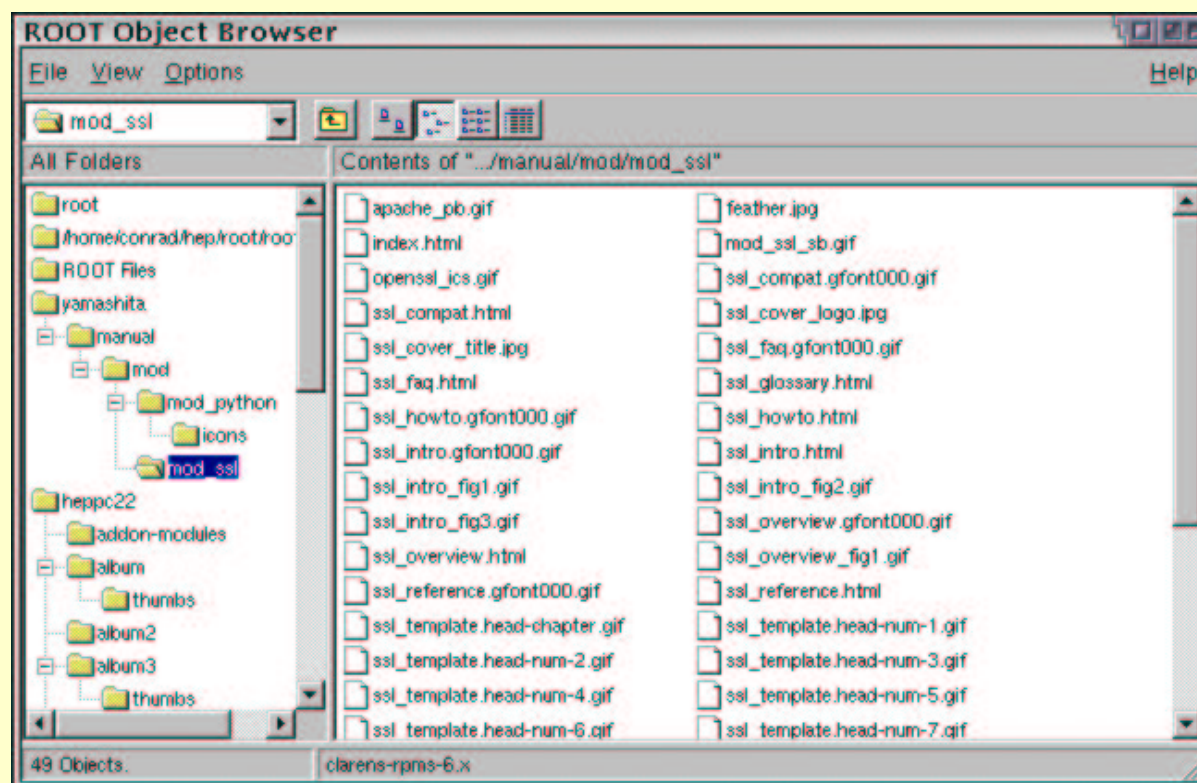




# Root client access III



- Browsing a remote file system on two servers, **yamashita** and **heppc22**:





# SRB file access



- The Storage Resource Broker is a network-enabled file and collection management system with a metadata catalog stored in an RDBMS (<http://www.npaci.edu/DICE/SRB/>)
- A Python binding for the SRB C-client was developed at SDSC, and the API subsequently exposed via Clarens.
- Since SRB uses a stateful network protocol, connections to the SRB server is opened and closed for every access by a Clarens client.
- Work is under way to use a temporary **pClarens** instantiation for persistent connections to the SRB server.

- Api:

```
connect(string srb_host,    string srb_port,  
        string srb_domain, string auth_scheme,  
        string user_name,  string srb_passwd)  
disconnect(string connection_id)  
read(string conn_id, string coll_name, string obj_id,  
     int  oflag,    int  offset,      int  max_size)
```



# Summary



- Client access is now available from a wide variety of client languages and environments:
  - Python
  - C/ C++
  - Root (C++ )
  - Java application
  - Java/ Javascript browser-based client
- GUI and command-line clients allow the user to pick the most convenient interface for the task at hand.
- Proxy escrow service allows access to Grid credentials from any web browser, or in environments where file access is impossible
- SQL2Root and SRB services give access to new data storage media from within the Clarens client environment