"Open Source, OGSA Implementation"

Genesis II: *Accessible, Standards Based Grid Computing*

Mark Morgan, Andrew Grimshaw

Global Bio Grid Team

University of Virginia

Responsible Parties

- Andrew Grimshaw
- Mark Morgan
- John Karpovich
- Duane Merrill
- Howie Huang
- Krasimira Kapitanova
- Karolina Sarnowska
- Chris Sosa

What’s the Problem

Our target grid users are unable to, or unwilling to learn new programming languages, coding paradigms, or complicated tooling.

Users want the benefit of the grid, but they want it transparently!

Abstract

- Sufficient richness of specification exists in OGF and OGSA to produce interesting and useful grids
- Genesis II is an open source, implementation of these specifications, modeled after the Legion grid system, providing both compute and data grid technology
- **By focusing on familiar, traditional abstractions such as files and directories we better serve the target grid user**

Outline

- Background
- Genesis II
- Summary

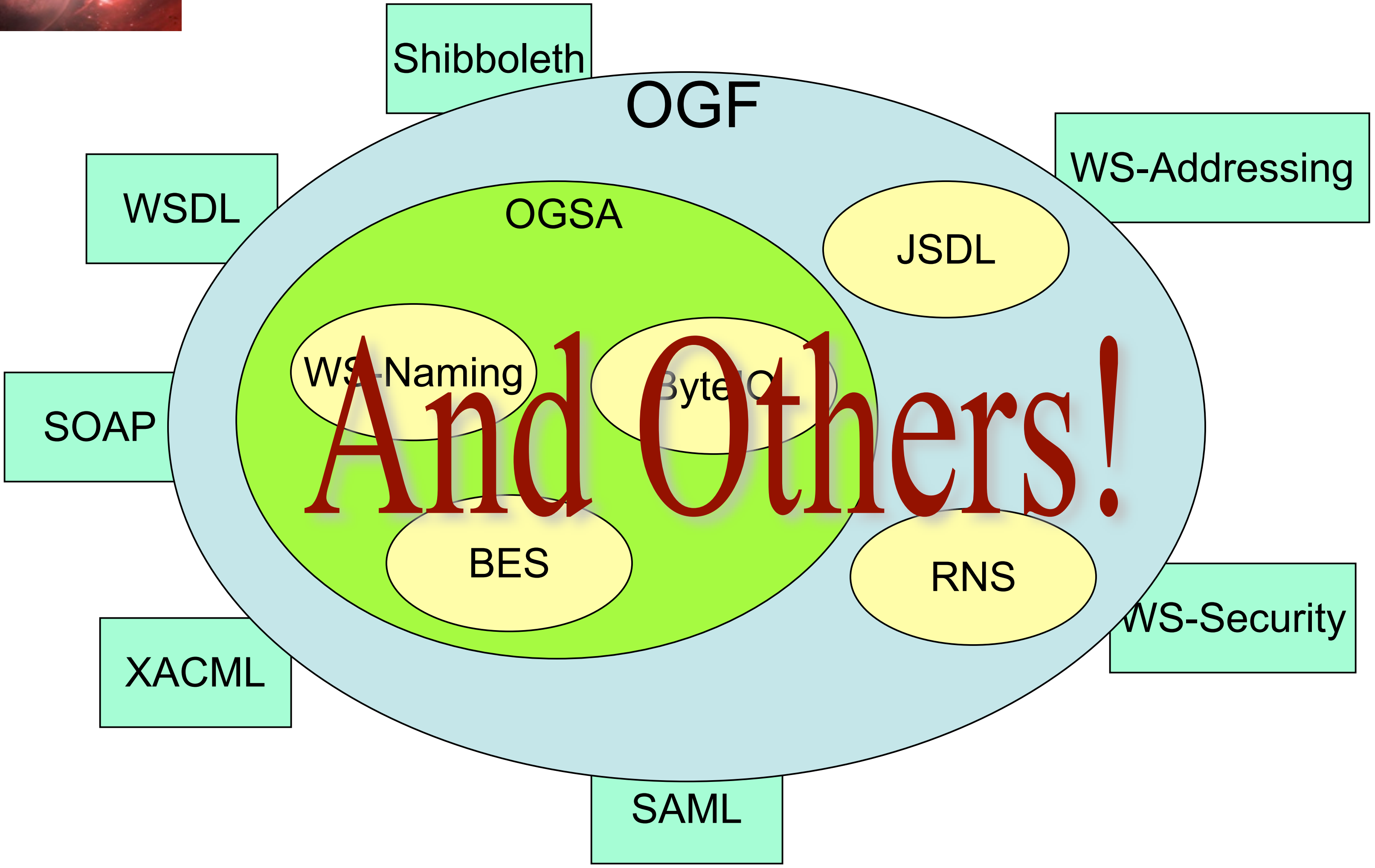
Background

- Specifications cannot exist in vacuum
  - Implementations Vet Specifications
  - Implementation experience shows how spec's interact
- Grids have been around for a while, but adoption remains low – why?
  - Usability
  - Gap between grid designers and grid users

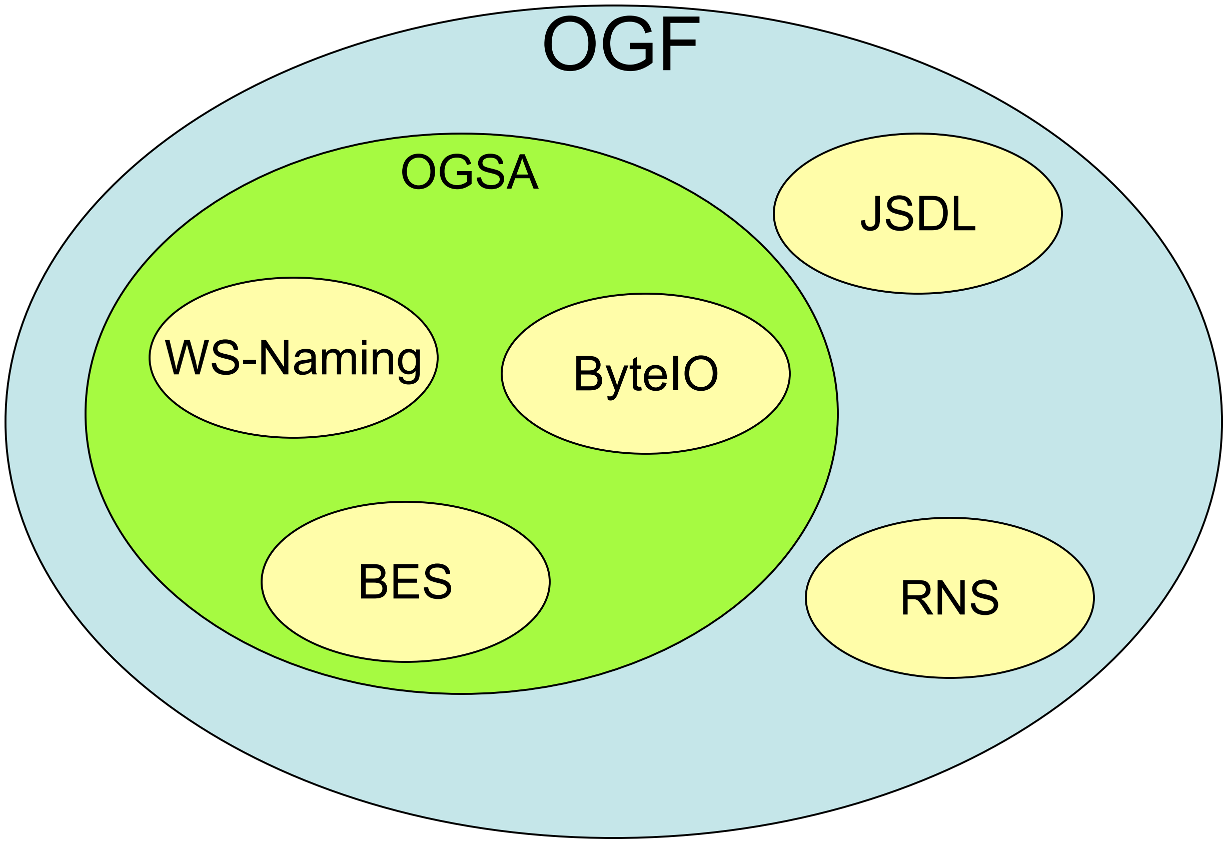
Genesis II Goals

- Provide an open source, reference implementation of the OGSA and OGSA-related specifications
- Use standards and proto-standards available from the OGF and OGSA to
  - Provide a secure, cohesive system in a production system available to users today!
  - Provide feedback into the OGF process on various standards based on implementation experience
  - Design the system from the ground up with the overriding mantra that **users come first**!

Standards Represented in Genesis II



Standards Represented in Genesis II

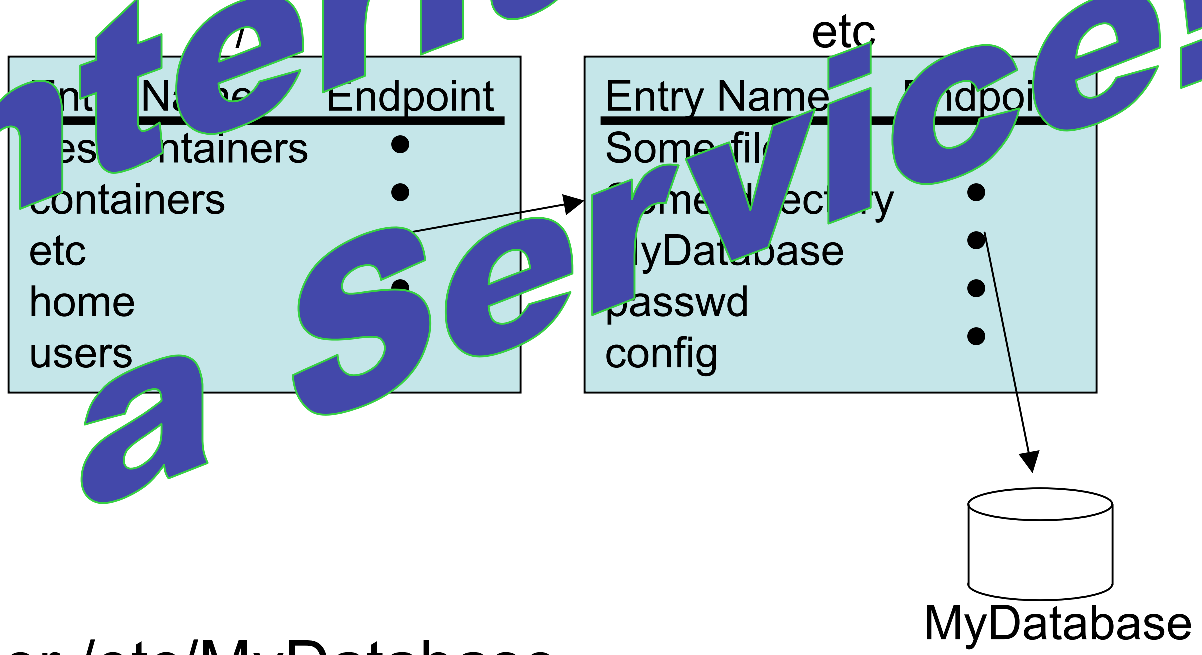


RNS (Resource Naming Service)

- Hierarchically maps human-readable names to web service endpoints (wsa:EndpointReferenceTypes)

- Add
- Remove
- List

Interface, Not a Service!



Consider /etc/MyDatabase

ByteIO

- Posix-*like* data IO
- Treat a resource as if it were a file
- Familiar operations
- Read, write, seek, truncate, append, etc.
- Ideal for mapping into familiar abstractions

Interface,
Not a Service!

BES (Basic Execution Service)

- Service interface for starting and managing remote compute jobs
- Implementation is not specified
  - Queue
  - Fork/exec
  - Virtualize
  - Etc.\
- Emphasis on <u>Basic</u>

WS-Naming

- Two components
  - Endpoint Identifiers
  - Endpoint Resolution

```
<EPR>
    <Address>http://tempuri.org/some-service</Address>
    <ReferenceParameters> xs:any </ReferenceParameters>
    <Metadata>
        <EndpointIdentifier> xs:anyURI </EndpointIdentifier>
        <EndpointIdentifierResolver>
            wsa:EndpointReferenceType
        </EndpointIdentifierResolver> *
    </Metadata>
</EPR>
```

Users First

- A large percentage of a grid's target audience is unable or unwilling to learn new interaction abstractions
- Instead of asking the user to adapt to the grid, we should adapt the grid to the user

User Abstractions

- One of the most ubiquitous user interaction abstractions is the file system
  - Drag-and-drop
  - Double Click
  - Named pipes
  - /proc filesystem
  - Plan 9
- RNS, ByteIO, and WS-Naming provide the foundation for building these abstractions

Thought Games

- What would it mean to browse a grid “directory” structure?
- What if you double-clicked on a ByteIO resource in it?
- What if you double-clicked on a grid resource representing a database query?
- What about dragging a JSDL document onto a BES container? A scheduler?
- What about “browsing” into a BES container?

Outline

- Background
- Genesis II
- Summary

Genesis II

- Standards based, open source, production level grid system
- Designed with the primary goal of putting users first
- Provides data grid and compute grid technology using secure infrastructure

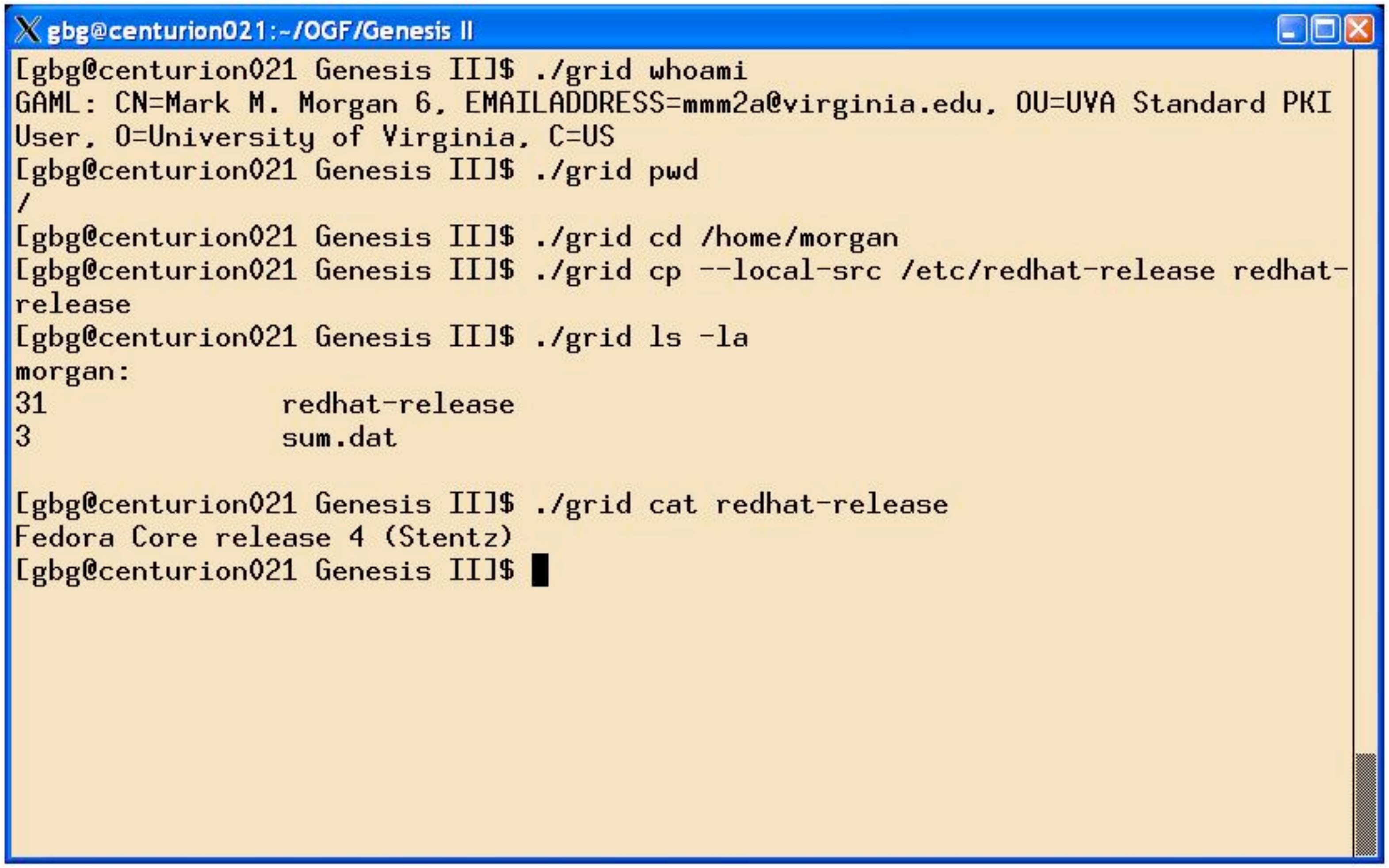
Genesis II Specifications

- Written in Java 1.6.x using Jetty 5, Axis 1.4, Apache Derby, and WSS4J
- Pluggable security infrastructure currently supporting both Username/Password profile, and GAML (Genesis II SAML implementation)
- Currently tested on Windows XP and Linux

Genesis II

- Every service/resource in Genesis II may implement RNS and/or ByteIO (and most do)

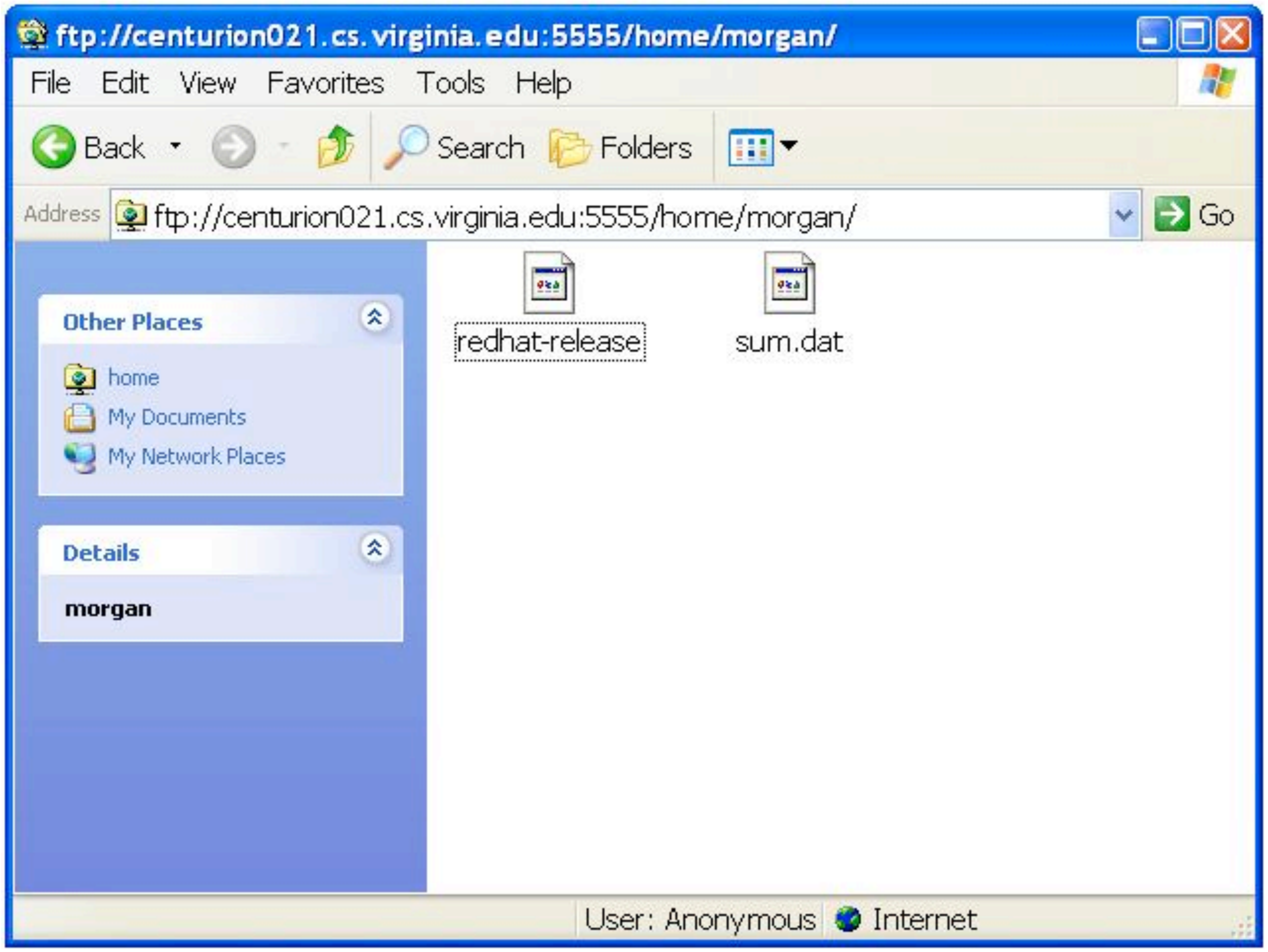
UNIX-like command line interface



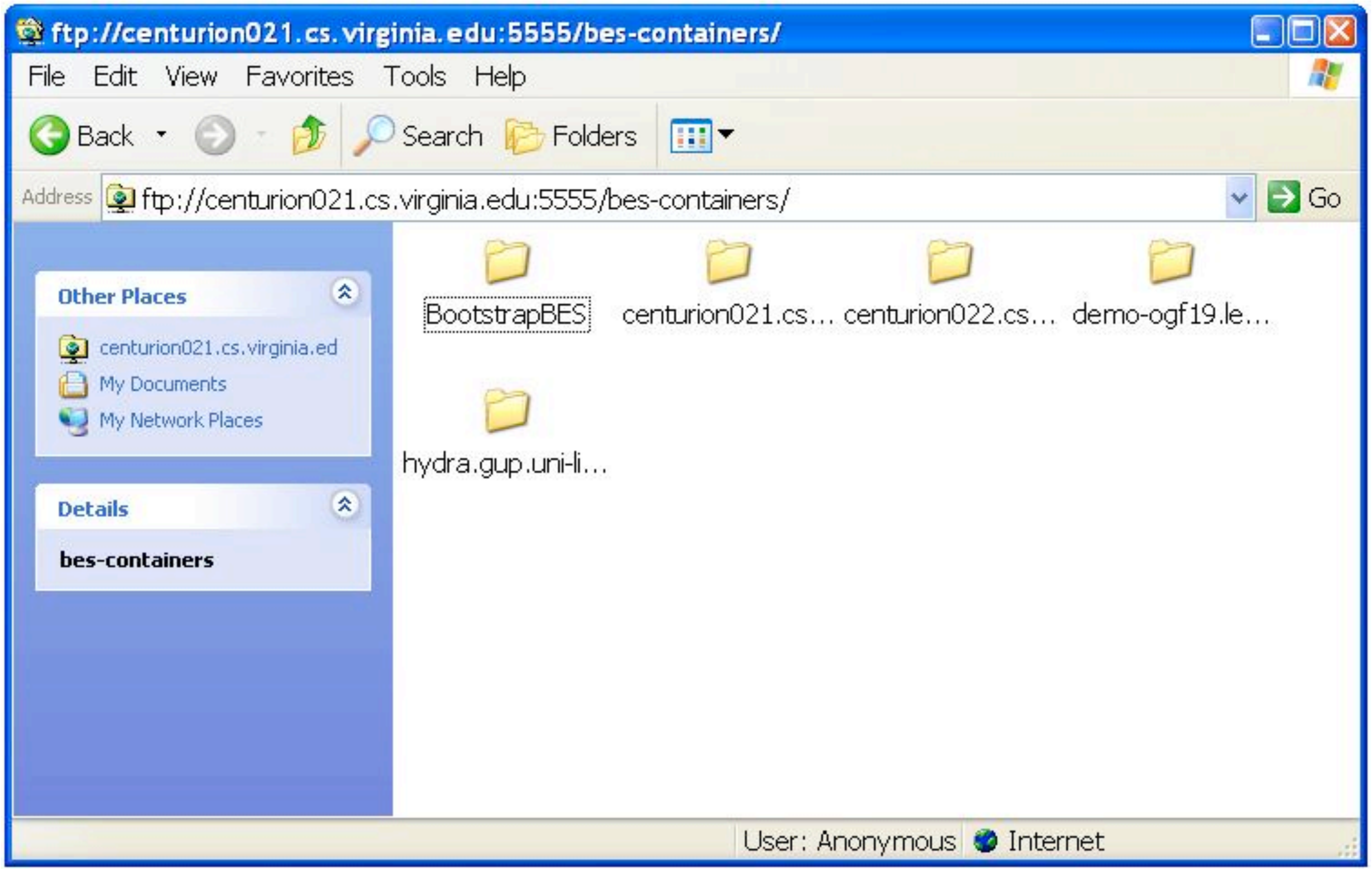
```
[gbg@centurion021 Genesis II]$ ./grid whoami
GAML: CN=Mark M. Morgan 6, EMAILADDRESS=mmm2a@virginia.edu, OU=UVA Standard PKI
User, O=University of Virginia, C=US
[gbg@centurion021 Genesis II]$ ./grid pwd
/
[gbg@centurion021 Genesis II]$ ./grid cd /home/morgan
[gbg@centurion021 Genesis II]$ ./grid cp --local-src /etc/redhat-release redhat-
release
[gbg@centurion021 Genesis II]$ ./grid ls -la
morgan:
31              redhat-release
3               sum.dat

[gbg@centurion021 Genesis II]$ ./grid cat redhat-release
Fedora Core release 4 (Stentz)
[gbg@centurion021 Genesis II]$ █
```

Filesystem Aware Interfaces


ftp://centurion021.cs.virginia.edu:5555/home/morgan/
File Edit View Favorites Tools Help
Back
Search
Folders
Address
ftp://centurion021.cs.virginia.edu:5555/home/morgan/
Go
Other Places
home
My Documents
My Network Places
Details
morgan
redhat-release
sum.dat
User: Anonymous
Internet

Using RNS to name non-filesystem components

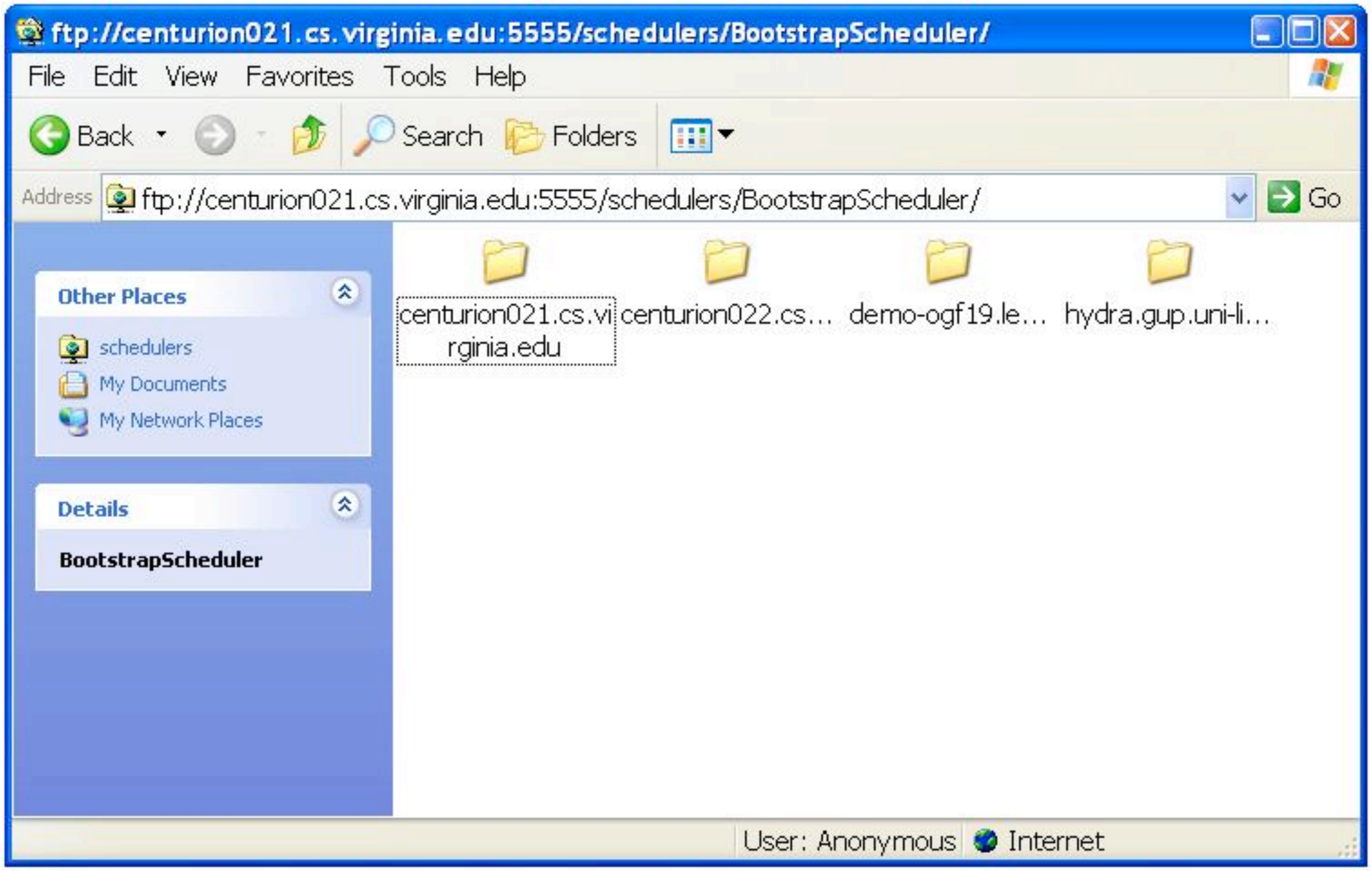


ftp://centurion021.cs.virginia.edu:5555/bes-containers/

BootstrapBES
centurion021.cs...
centurion022.cs...
demo-ogf19.le...
hydra.gup.uni-li...

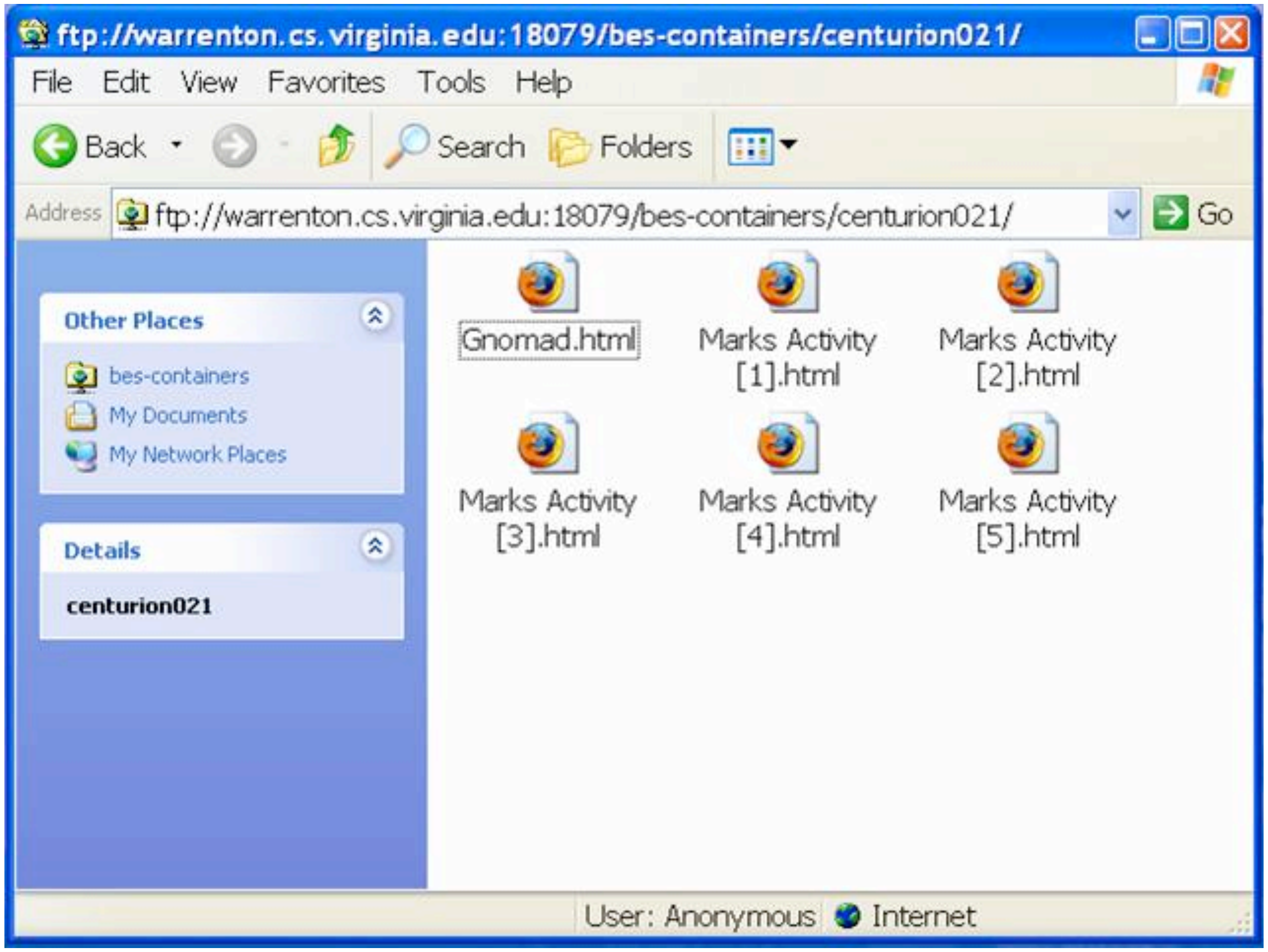
User: Anonymous
Internet

Using RNS to name non-filesystem components



ftp://centurion021.cs.virginia.edu:5555/schedulers/BootstrapScheduler/

Genesis II's BES implements RNS too!

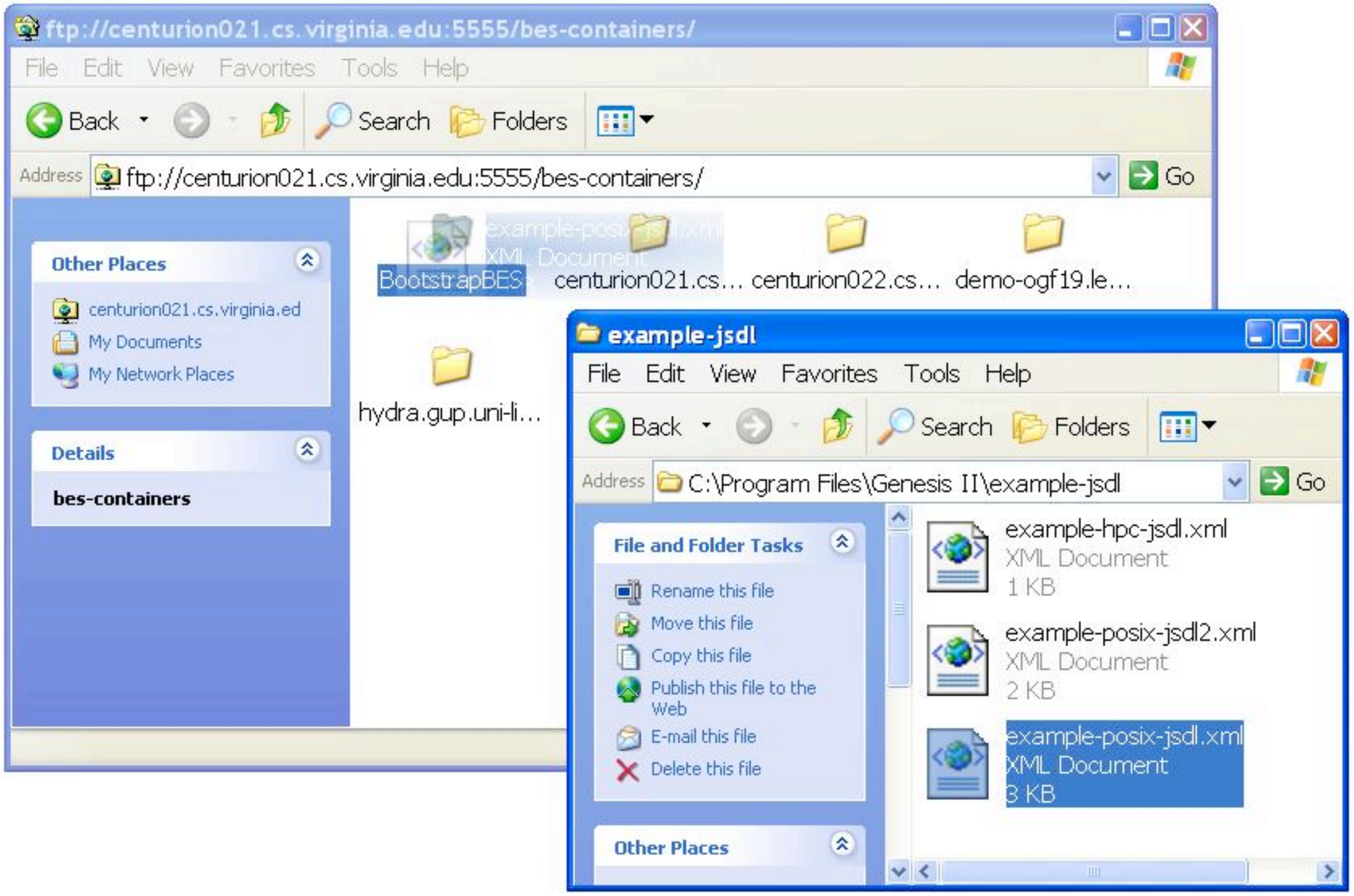


Address: ftp://warrenton.cs.virginia.edu:18079/bes-containers/centurion021/

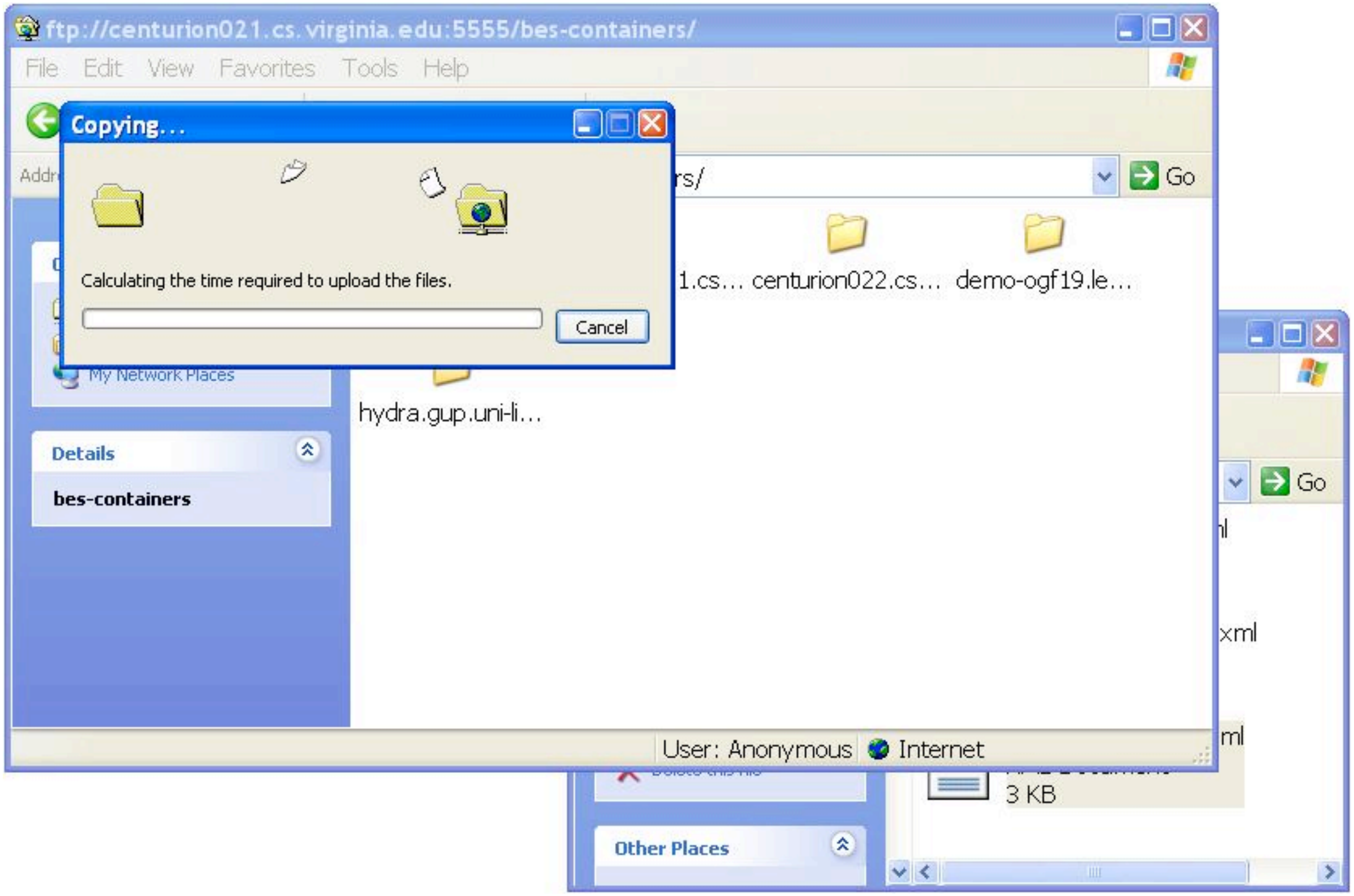
Details: centurion021

Gnomad.html
Marks Activity [1].html
Marks Activity [2].html
Marks Activity [3].html
Marks Activity [4].html
Marks Activity [5].html

Genesis II's BES even implements ByteIO!



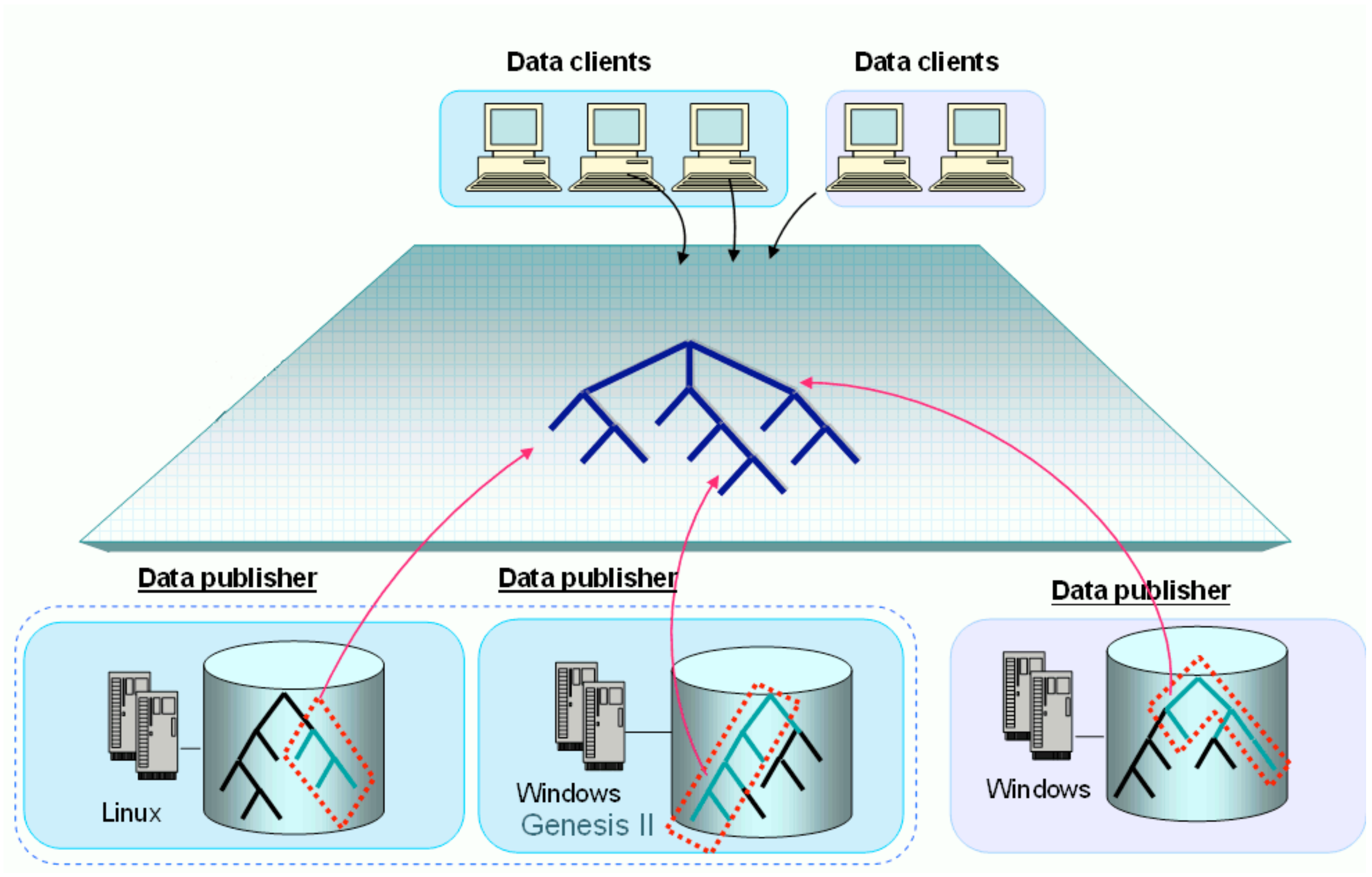
Genesis II's BES even implements ByteIO!


ftp://centurion021.cs.virginia.edu:5555/bes-containers/
File Edit View Favorites Tools Help
Copying...
Cancel
1.cs... centurion022.cs... demo-ogf19.le...
hydra.gup.uni-li...
My Network Places
Details
bes-containers
Go
User: Anonymous
Internet
3 KB
Other Places

Calculating the time required to upload the files.

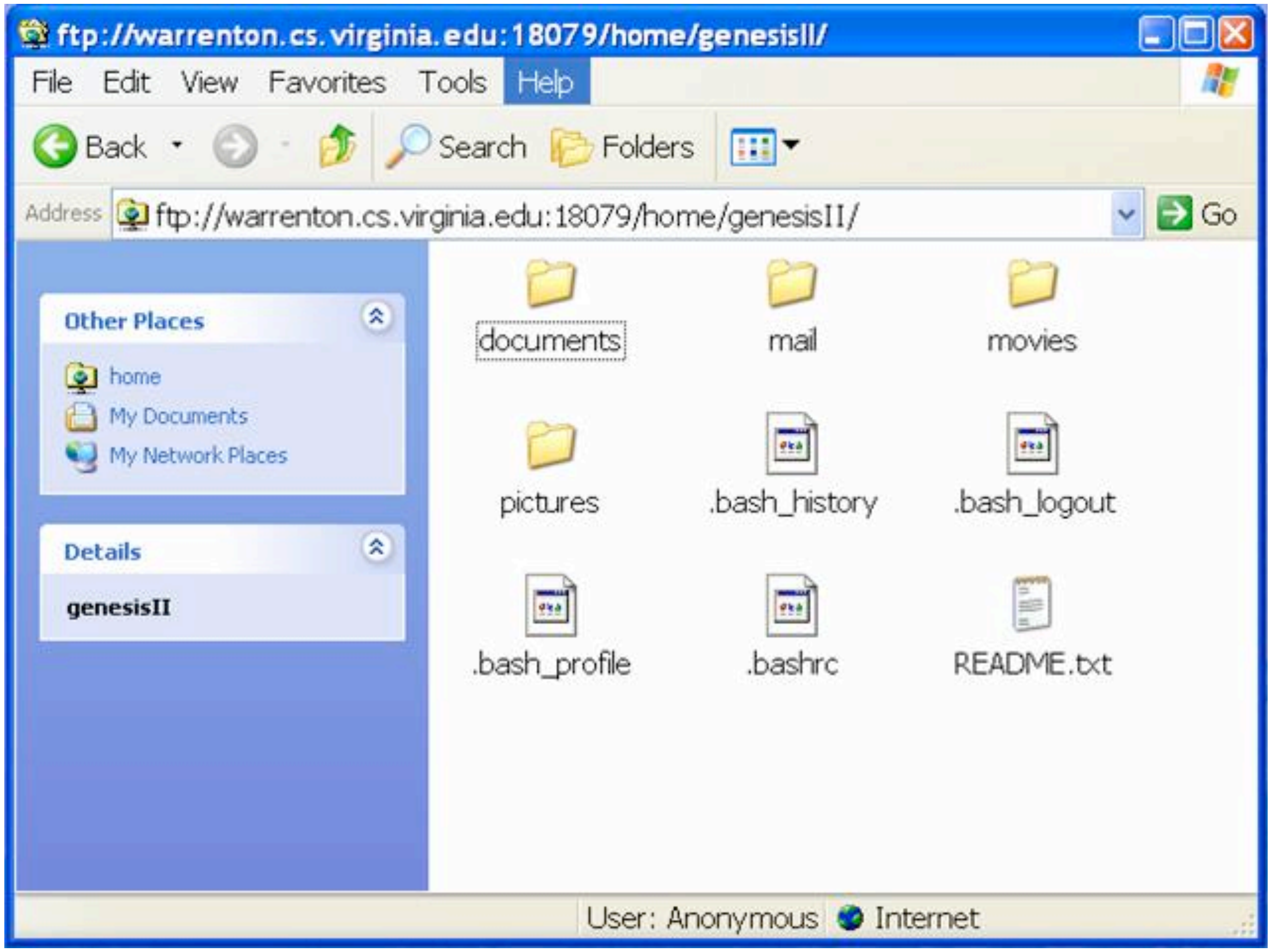
Genesis II (http://www.cs.virginia.edu/~vcgr)
"Open Source, OGSA Implementation"

Export Directory



Genesis II (http://www.cs.virginia.edu/~vcgr)
"Open Source, OGSA Implementation"

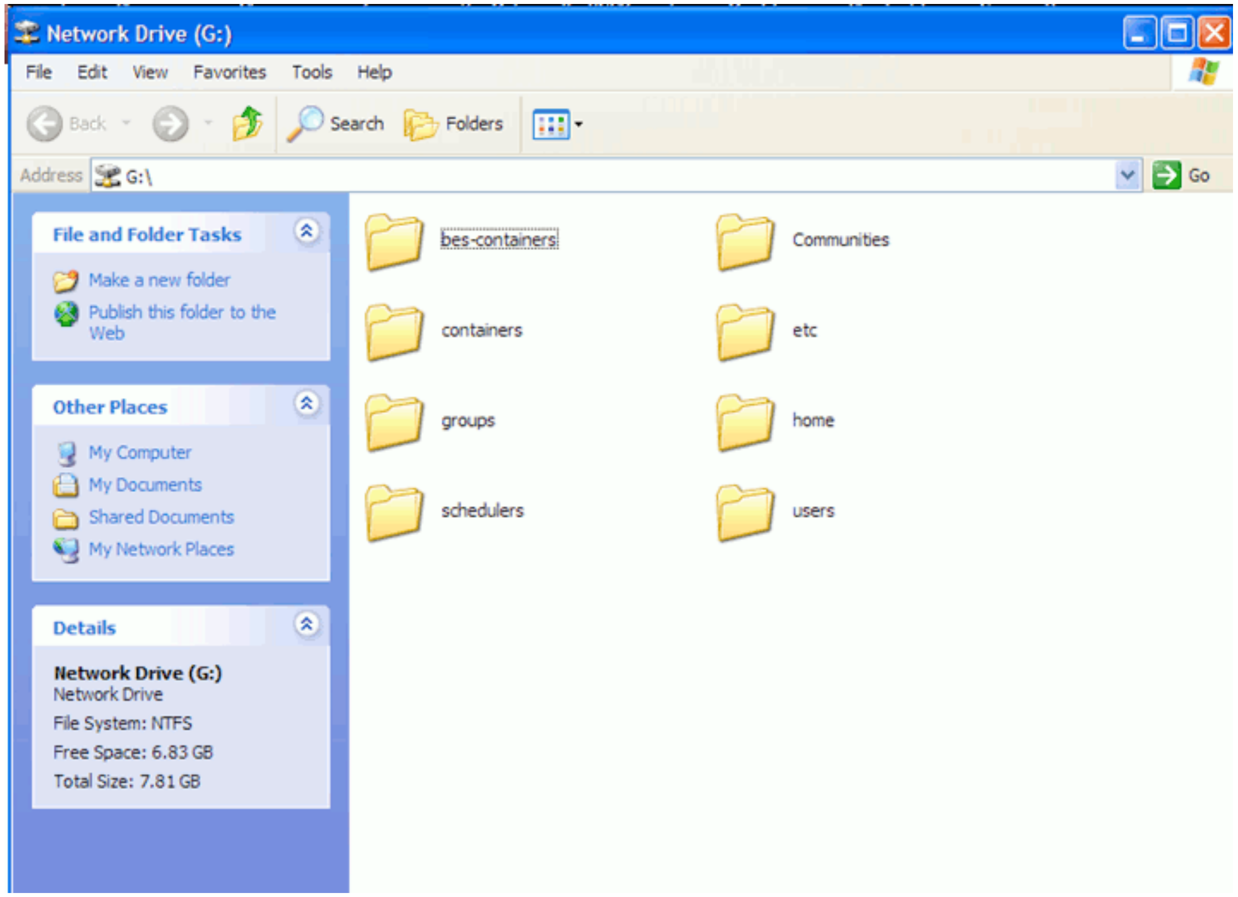
Export Directory



WS-Naming and Genesis II

- Extensive use of Endpoint Identifiers
- Simple Resolvers
- Future Work
  - Security Modifications
  - Fault Tolerance
  - Performance
  - QoX
  - Etc.

Other Filesystem Aware Interfaces

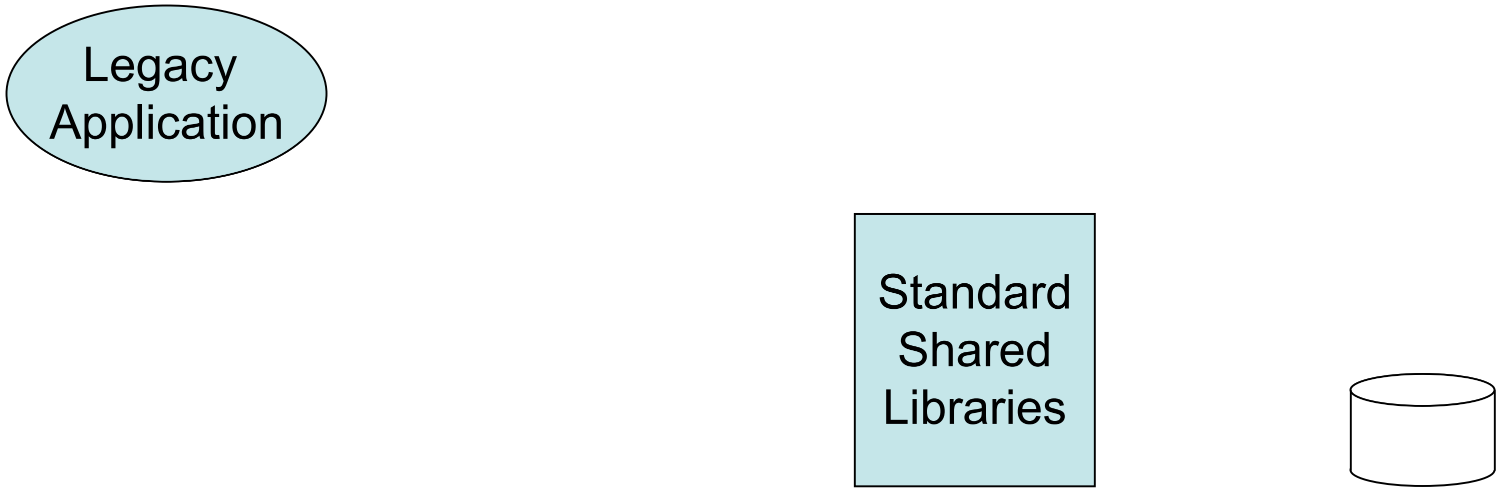


Network Drive (G:)
Network Drive
File System: NTFS
Free Space: 6.83 GB
Total Size: 7.81 GB

bes-containers
Communities
containers
etc
groups
home
schedulers
users

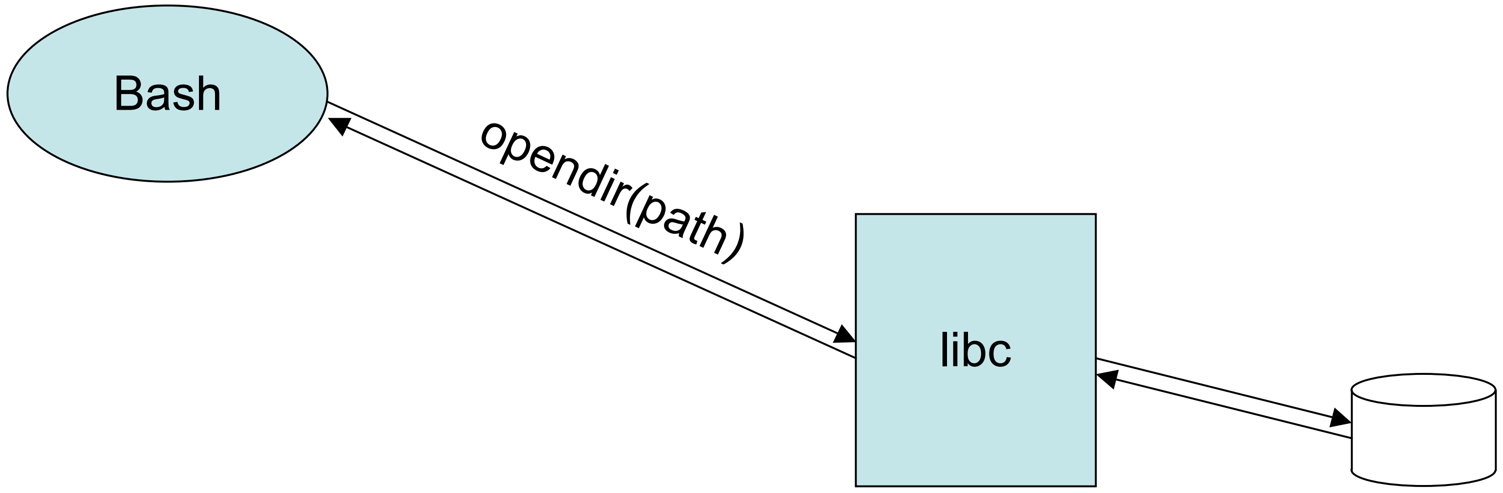
OGRSH (Open Grid Shell)

- Classic software interposition library or shim



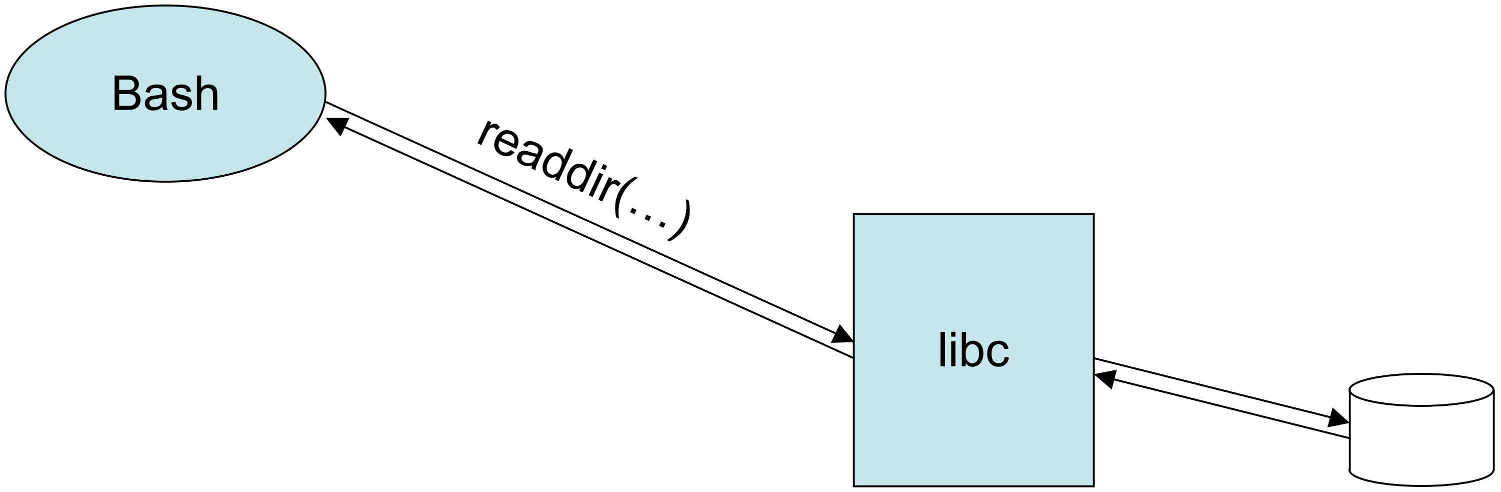
OGRSH (Open Grid Shell)

- Classic software interposition library or shim



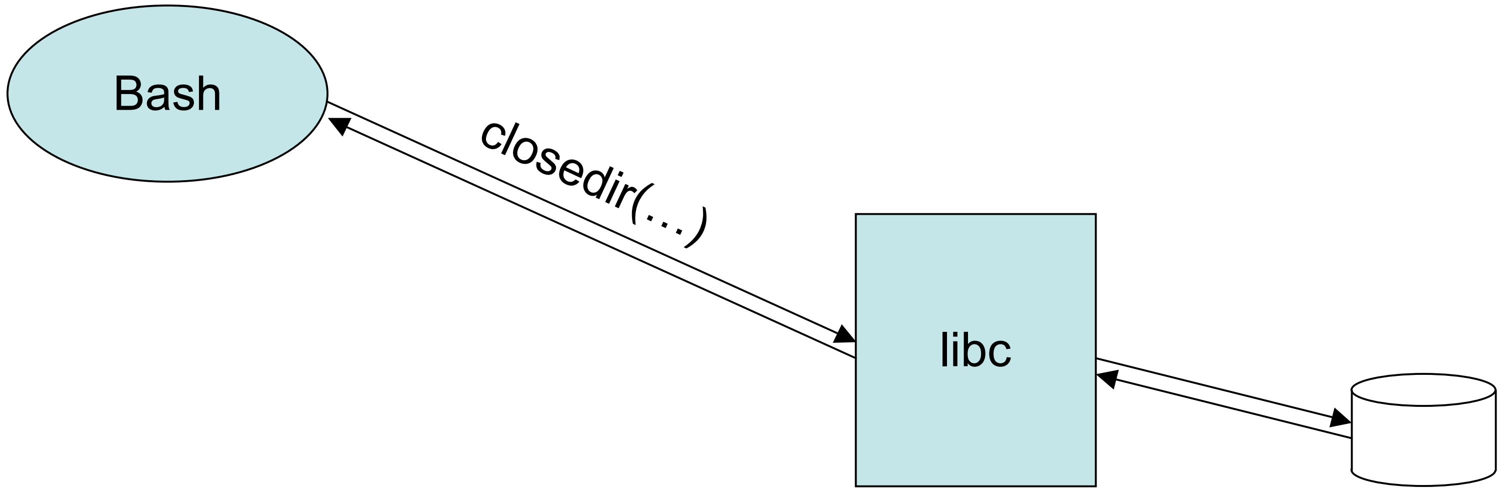
OGRSH (<u>O</u>pen <u>Gr</u>id <u>Sh</u>ell)

- Classic software interposition library or shim



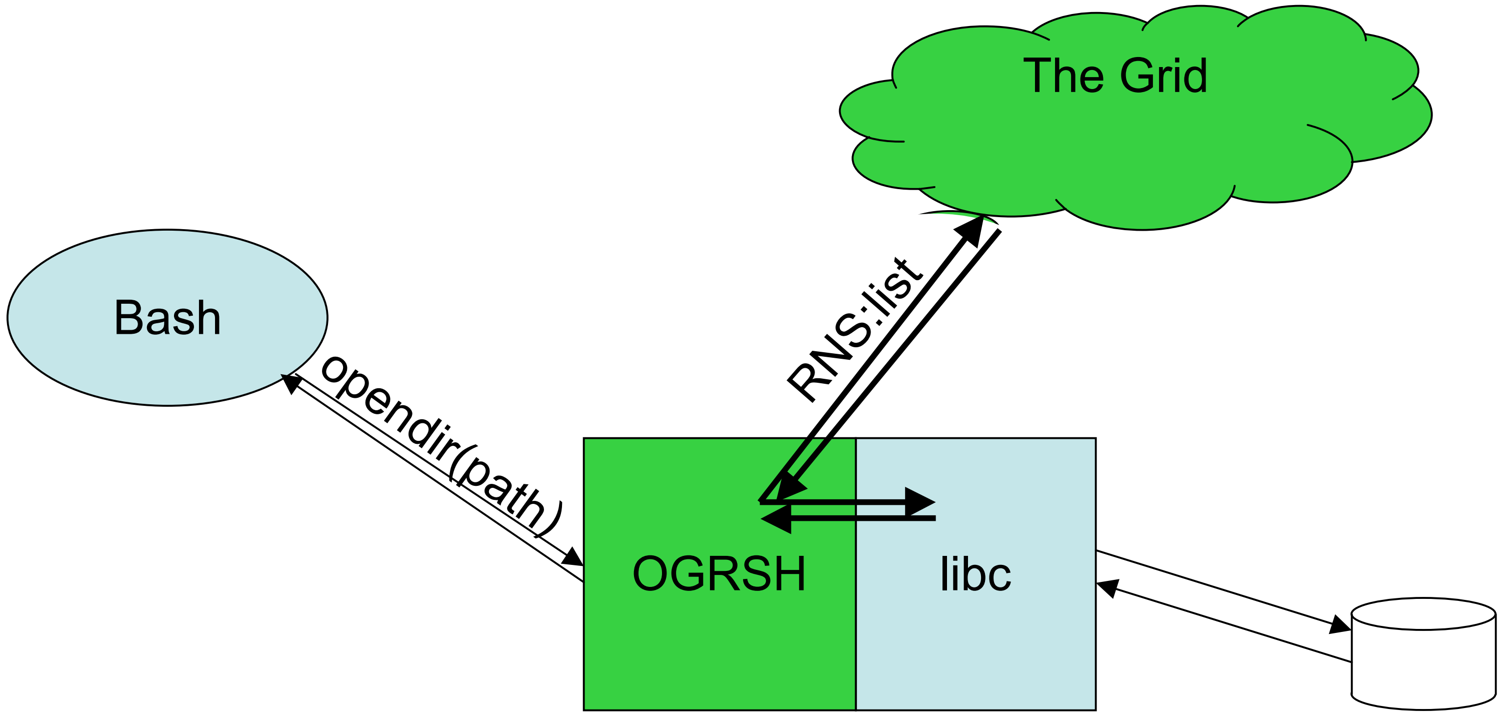
OGRSH (Open Grid Shell)

- Classic software interposition library or shim



OGRSH (Open Grid Shell)

- Classic software interposition library or shim



OGRSH

Insert Picture Here

Outline

- Background
- Genesis II
- Summary

Summary

- Potential Grid users want the benefits of the grid without the pain.
- Grid uptake therefore is closely tied to usability
- As some systems have demonstrated in the past, users are better at learning new semantics then new syntax.
- Genesis II leverages this by providing the familiar syntax or abstractions of file systems to perform “everyday” grid activities.

Genesis II
Take-away messages

- Open Source implementation of the OGF and OGSA standards available
- **Sufficient body of standards exist with which to build interesting, useful grid systems**
- Very active project!
- Information and Download Page
  - http://vcgr.cs.virginia.edu/genesisII
- Forum
  - http://www.cs.virginia.edu/forums/viewforum.php?f=26

Questions?

http://vcgr.cs.virginia.edu/genesisII

Security Design Goals

- **Pluggable security modules (at run time)**
- Secure communication: Authentication, Confidentiality, Integrity, Authorization
- Standards-based:
  - Follow OGSA AuthN & AuthZ models as they develop
  - Leverage standard protocols, specifically Username/Password, X.509, and SAML WS-Security token profiles

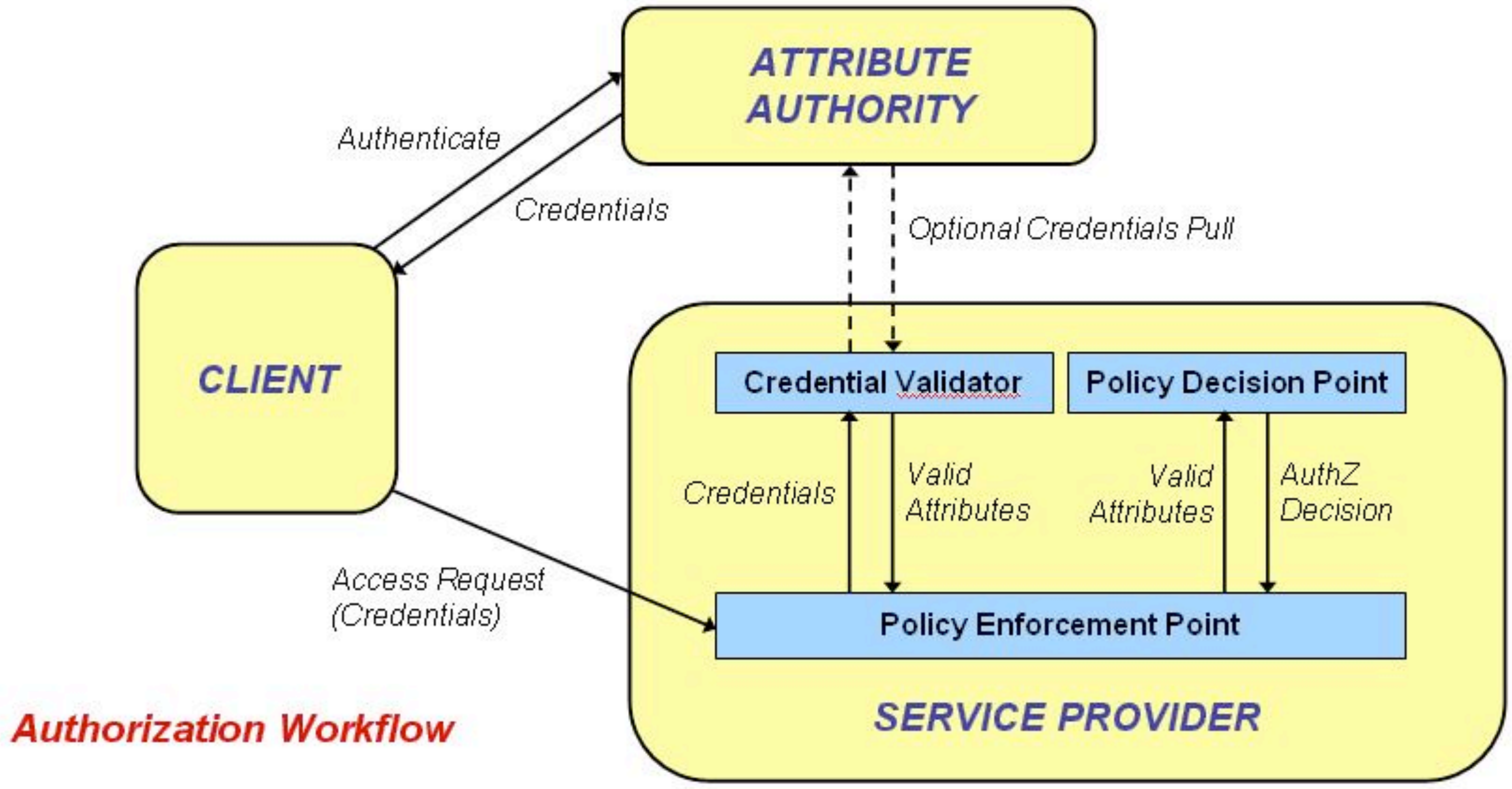
Authorization

- Subject identity (LDAP, NIS, X.509, etc.) is federated via Attribute Authorities (AA)
- Credentials are signed, “holder-of-key” SAML attribute assertions (e.g., identity, roles, capabilities, etc.)
- Credentials may be delegated, but have limited lifetimes
- Attribute Authorities implement the SAML Authentication Request profile (and optionally the Assertion Query and Request profile for a pull-style model enabling Shibboleth-like anonymity)

Authentication, Integrity, and Confidentiality

- Asymmetric public-key cryptography
- Identity via trust hierarchies: X.509 digital certificates
- Resource identity is correlated to digital certificates by embedding WS-Naming EPIs as a X.509 non-critical extension
- Digital certificate distribution via EPRs: certificate-chains embedded as non-critical EPR metadata
- Server-side authentication (as opposed to mutual authentication which restricts privacy, anonymity, delegation)
- For message integrity, the caller can generate a keypair

Authentication, Integrity, and Confidentiality


ATTRIBUTE AUTHORITY
Authenticate
Credentials
Optional Credentials Pull
CLIENT
Credential Validator
Policy Decision Point
Credentials
Valid Attributes
Valid Attributes
AuthZ Decision
Access Request (Credentials)
Policy Enforcement Point
SERVICE PROVIDER
Authorization Workflow