“Open Source, OGSA Implementation”

Simplifying Access to Grid Resources

Andrew Grimshaw

University of Virginia,

OGF Architecture Area Director

CCGSC 2008, Flat Rock, NC

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

Outline

- Why Grids have failed to cross the chasm
- What can be done?
- Genesis II
- Summary

Why have Grids failed to cross the chasm?

- Applications programmers are exposed to the complexity of the underlying environment
  - Failures, data/application management, logging, …
  - Only the enthusiasts want to live on the bleeding edge
  - Joe six-pack (biologist | chemist | economist | *) just wants to do their domain research – not do heavy duty hacking
  - “Activation energy” is too high
- Once they have spent the effort to port there is little leverage – applications written for one software stack will not work on other stacks – often they will not even work on different versions of the same software stack
  - They must suffer the pain all over again
- Business needs solutions not technology
  - HTC solutions (e.g., SGE, LSF, Condor) are easy to use – but not really needed cross-site.
  - Data sharing solutions are primitive and do not integrate well with the existing infrastructure

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

What can be done?

Simplify

Standardize

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

Simplify

- Rather than make the programmer and application adapt to the Grid – make the Grid adapt to the user/application.
  - Reduce activation energy
- Remember – the world is filled with legacy code.
- Use paradigms and metaphors with which users are familiar, e.g., directories and files.
- In the limit – the user and application should be unaware of the Grid – the Grid is plumbing.
- This does not mean everyone must use high-level abstractions (End-to-End $_{\text{Saltzer}}$)

Ease-of-use is critical

- Look at the Web – it is ubiquitous because the browser made it easy
  - Hypertext and remote access existed for years before the "web" happened
- Users don't want to know about Web Services, delegation mechanisms, XML, WS containers, etc.
- They don't want to know the Grid exists at all – it should be like plumbing.
- It should install like a game or Turbotax

Standards

Why Standardize?

- What is the value of implementing standards?
- For vendors
  - meet customer demand for interoperability
- For developers
  - leverage the expertise of other developers
  - offer a choice of tools and platforms in order to speed implementations
  - only need to support one integration interface
- For end-users
  - reduce the costs and risks of adopting grid technology
  - get insight into the best practices of the industry at large
  - Use “best-in-breed” implementations and mix and match

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

Where are we now?

- We have a sufficient corpus of specifications and profiles to realize (implement) identified use cases in computing and data.
- Inter-operation has been demonstrated for a number of specs the HPC – Basic Profile and its’ descendants, RNS 1.0, ByteIO, OGSA WSRF-BP

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

Genesis II

Basic idea: By focusing on familiar, traditional abstractions such as files and directories we better serve the target grid user

Genesis II: (http://www.cs.virginia.edu/~vcgr)

“Open Source, OGSA Implementation”

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!

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

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**!

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

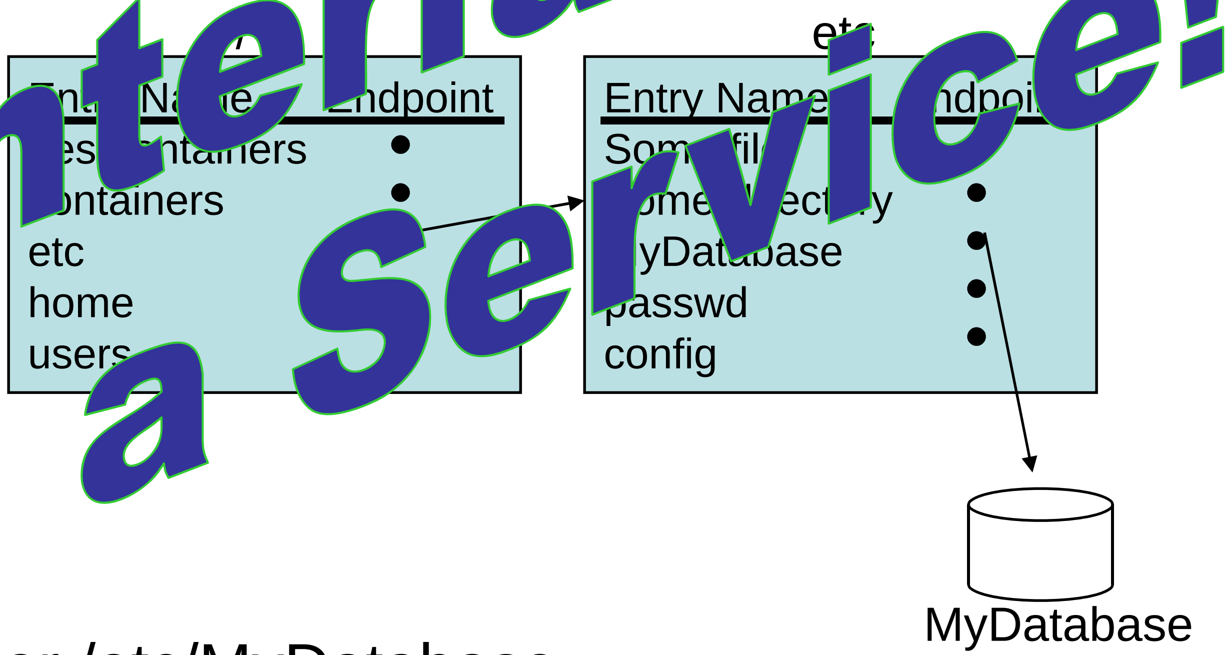
Focus on Four Specifications

There are also a number of security standards – but that is a whole talk unto itself

RNS (Resource Namespace Service)

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

- Add
- Remove
- List



Interface, Not a Service!

Consider /etc/MyDatabase

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

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!

WS-DAIR

Data Access and Integration - Relational

- Relational member of the WS-DAI family
- EPR Execute(SQL query) – returns an EPR

Interface,

Not a Service!

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

BES (Basic Execution Service)

- Service interface for starting and managing remote compute jobs
- JSDL as the job specification language
- Implementation is not specified
  - Queue
  - Fork/exec
  - Virtualize
  - Etc.
  - Emphasis on Basic
- Comes together in the HPC-BP

Now combine these in a familiar way

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

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 and ByteIO provide the foundation for building these abstractions

(Most) everything is a file or directory

- Files and directories can be accessed without knowledge of Grids or Web Services using Grid-aware FUSE and Windows IFS
  - map the Grid into the file system
- BES resources, queues are directories
  - “ls” to list the jobs, “cat” a job to see its state
  - “cp” a JSDL file into the directory -> start the job up
  - A shell script can start jobs by copying
- Genesis II containers are directories
  - “ls” to see the services and porttypes
- IDP are files/directories (WS-Trust STS)
- RDBMS’s are directories and files
  - “cp” a query file to DAIR endpoint, creates result that is also a file/directory/DAIR
  - Means you can “cat” out the results, or load them straight into Excel
- The user can access all of these services without dealing with Web Services!!

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

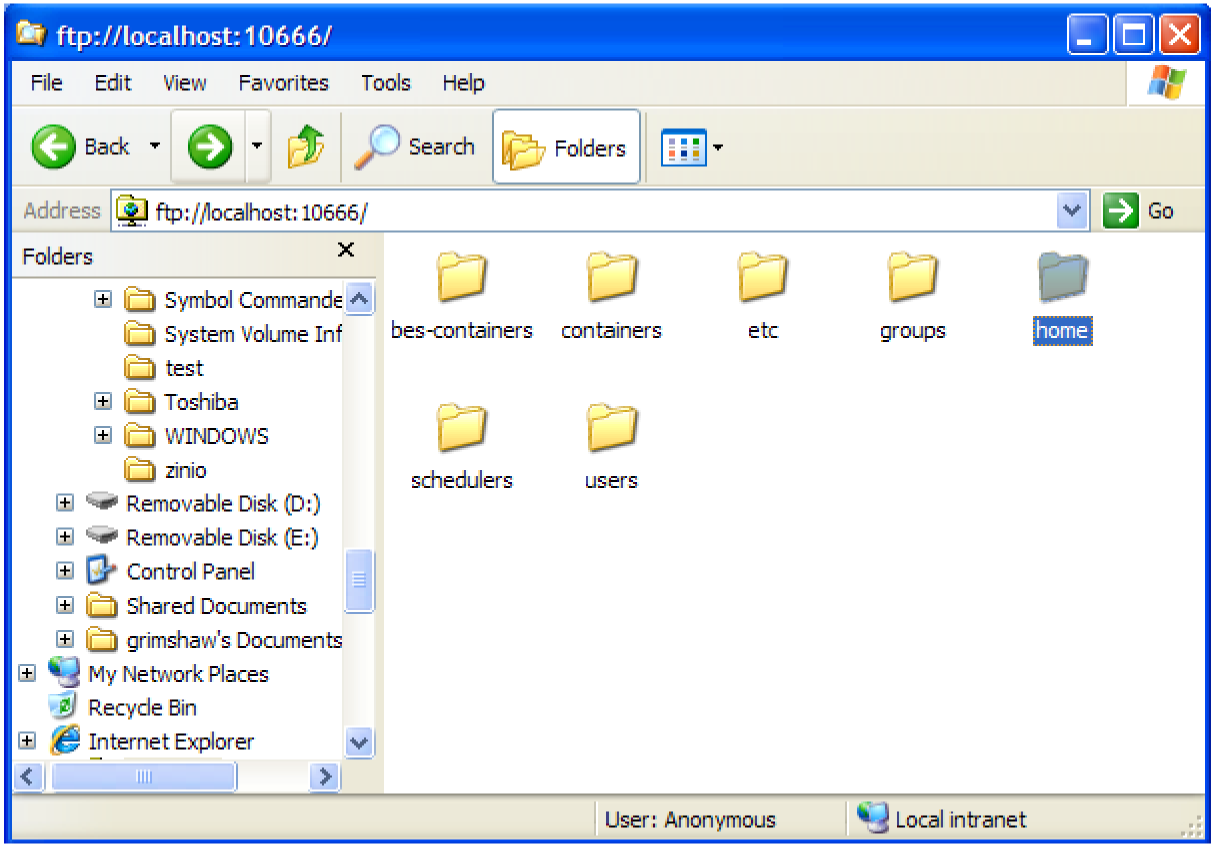
Genesis II - Data

- FTPd for Windows
- Grid-aware FUSE driver for Linux
- IFS for Windows (a.k.a. G-icing)
- ExportDir
- *Replicated ExportDir*

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

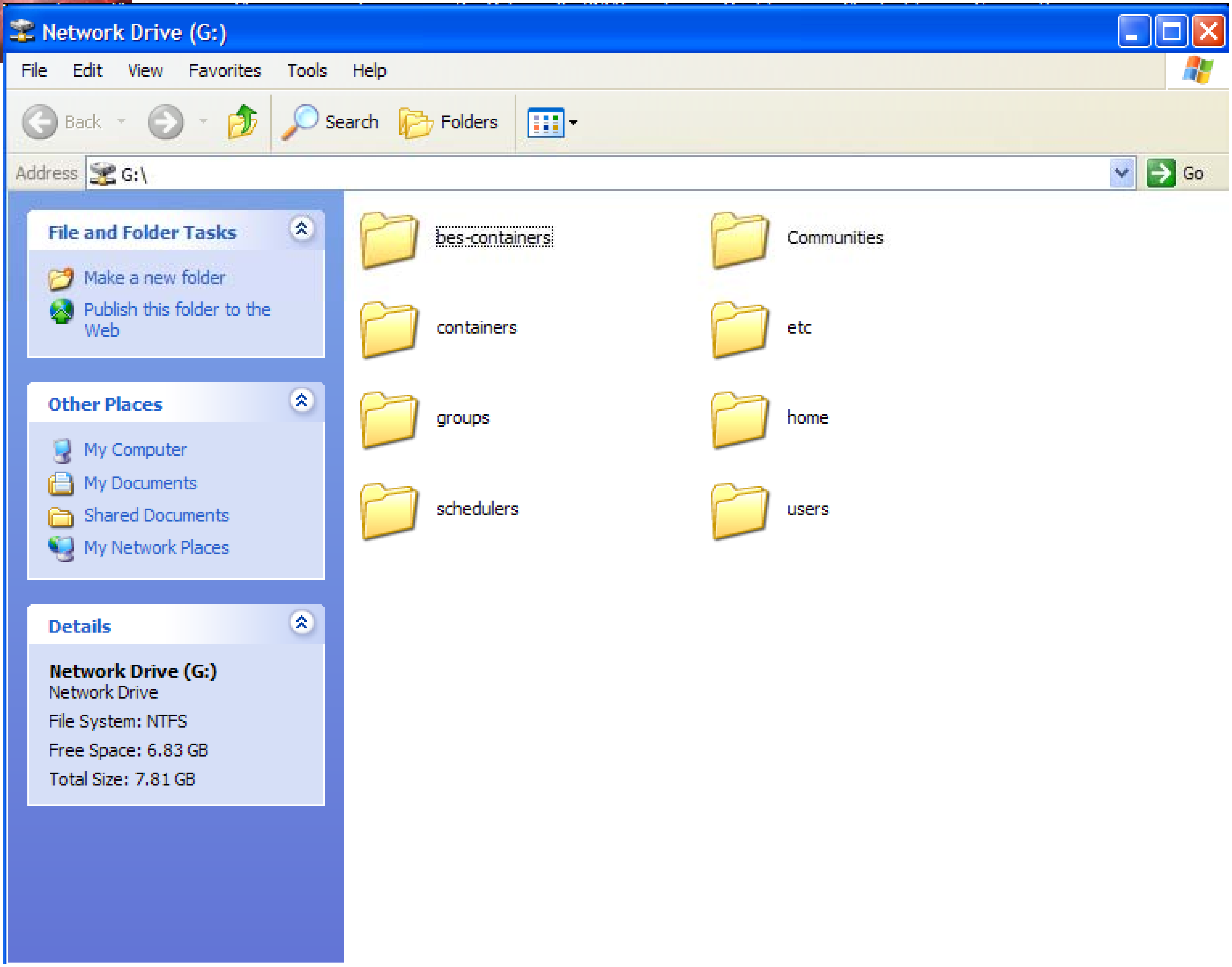
Localhost Fdpd

- Fully secure
  - My X.509
  - SSL
  - No bits in the clear



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

Windows IFS

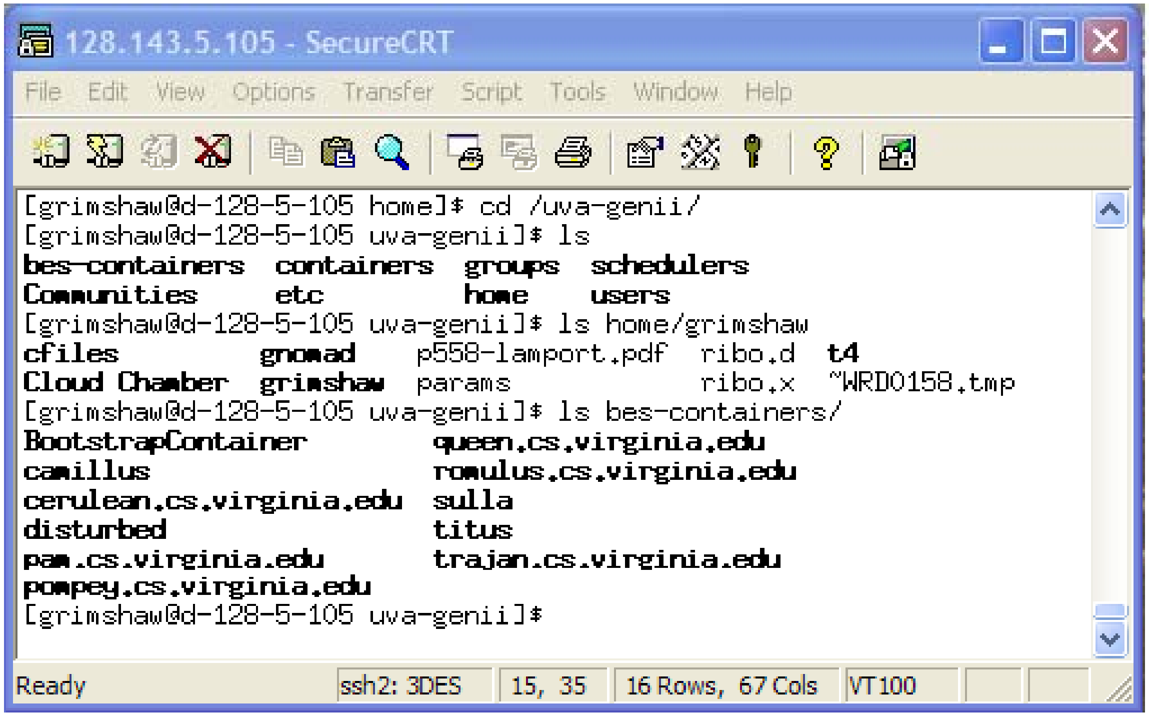


Address: G:\

bes-containers
Communities
containers
etc
groups
home
schedulers
users

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

Grid-aware FUSE

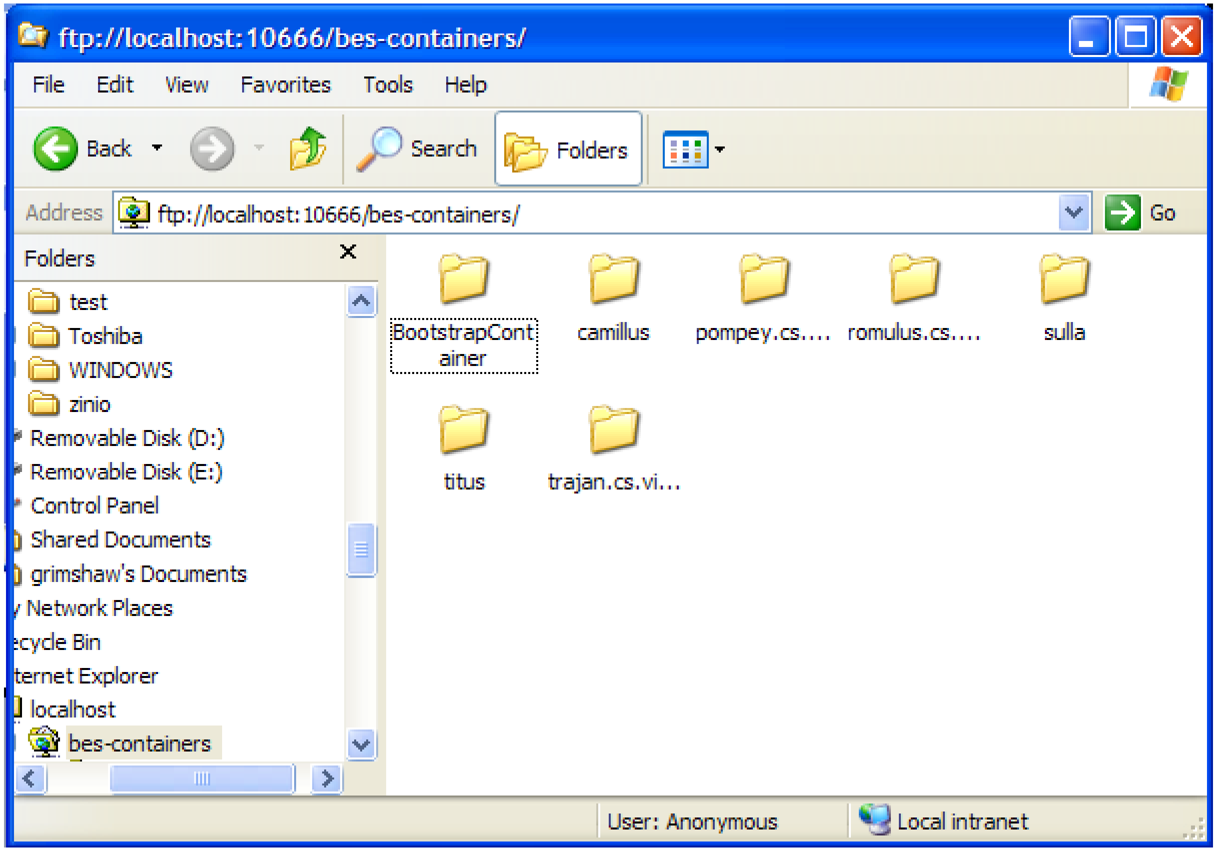


128.143.5.105 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Ready
ssh2: 3DES
15, 35
16 Rows, 67 Cols
VT100

```
[grimshaw@d-128-5-105 home]$ cd /uva-genii/
[grimshaw@d-128-5-105 uva-genii]$ ls
bes-containers  containers  groups  schedulers
Communities     etc         home    users
[grimshaw@d-128-5-105 uva-genii]$ ls home/grimshaw
cfiles         gnomad    p558-lamport.pdf  ribo.d  t4
Cloud Chamber  grimshaw  params            ribo.x  ~WRD0158.tmp
[grimshaw@d-128-5-105 uva-genii]$ ls bes-containers/
BootstrapContainer        queen.cs.virginia.edu
camillus                  romulus.cs.virginia.edu
cerulean.cs.virginia.edu  sulla
disturbed                 titus
pam.cs.virginia.edu       trajan.cs.virginia.edu
pompey.cs.virginia.edu
[grimshaw@d-128-5-105 uva-genii]$
```

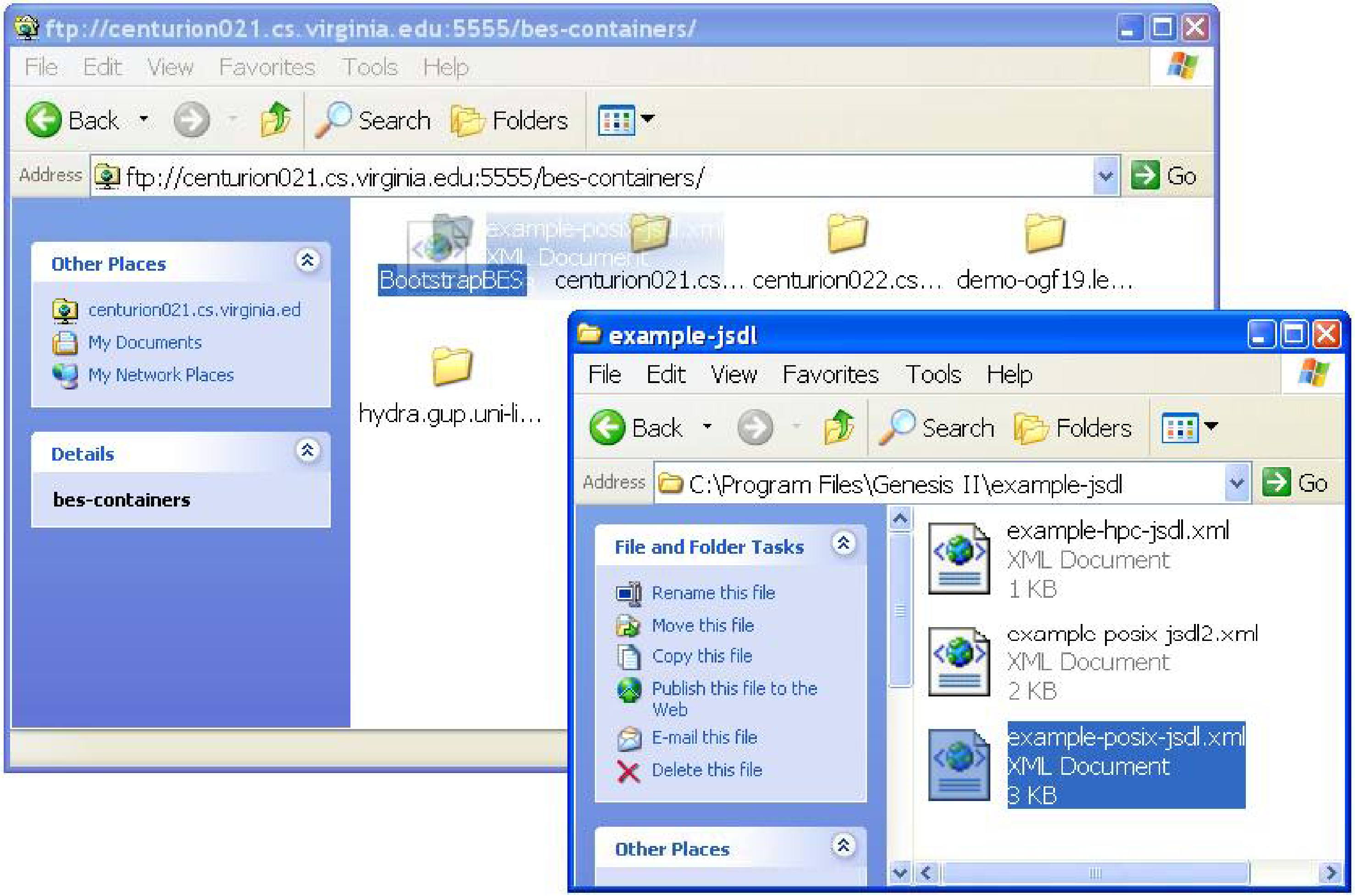
Using RNS to name non-file-system components

- BES resources are also RNS directories
- We can schedule a job on a resource simply by “dropping” it into the directory



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

Genesis II's BES even implements ByteIO!

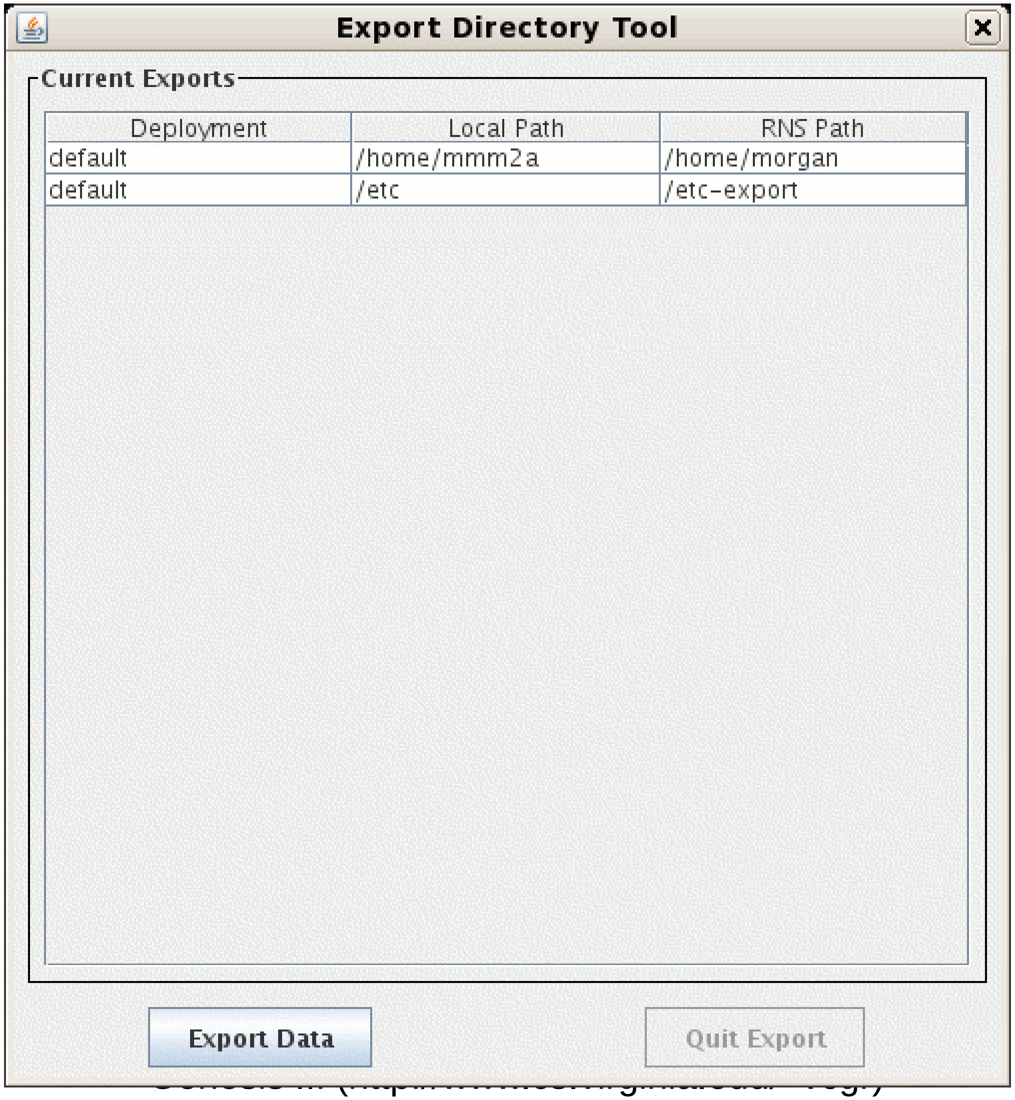


Genesis II: (http://www.cs.virginia.edu/~vcgr)
"Open Source, OGSA Implementation"

Export Directory

- Map a Unix or Windows file system into the Genesis II RNS name space so that others can securely Create/Read/Write/Destroy

Export Directory Tool

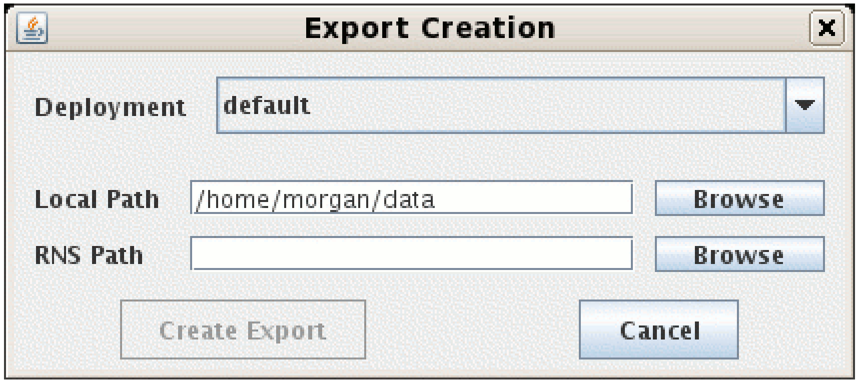


Deployment	Local Path	RNS Path
default	/home/mmm2a	/home/morgan
default	/etc	/etc-export

Genesis II (http://www.cs.virginia.edu/~vcgr)

“Open Source, OGSA Implementation”

Create an Export Dir
Select local path



Export Creation

Deployment: default

Local Path: /home/morgan/data

RNS Path:

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

Compute

- BES containers are basic building block
- Our BES containers are also directories into which JSDL files can be copied
- We have a simple FIFO queue that is also a directory. The queue is configured by linking (ln) BES containers

Running a job

- Just copy a file into a queue or BES
- cp gnomad.xml windowsq
- Can also send batches of JSDL files
- We also have a tool that automatically generates JSDL

Genesis II Summary

- Potential Grid users want the benefits of the grid without the pain.
- Grid uptake is closely tied to usability
- 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: (http://www.cs.virginia.edu/~vcgr)
“Open Source, OGSA Implementation”

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

Conclusion – bad news

- Grids have not “crossed the chasm”
- As a community we failed to manage complexity and provide systems and architectures usable by non-experts
- We failed to provide interoperable stacks
- Most users need higher level, simpler abstractions
  - To build these requires stable lower-level abstractions

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

Conclusion – good news

- Usable, interoperable low-level standards that can be used to solve real problems have been developed and implemented.
- It is possible to bridge the semantic gap between low-level services and abstractions users know how to use
- ISVs are starting to get involved

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