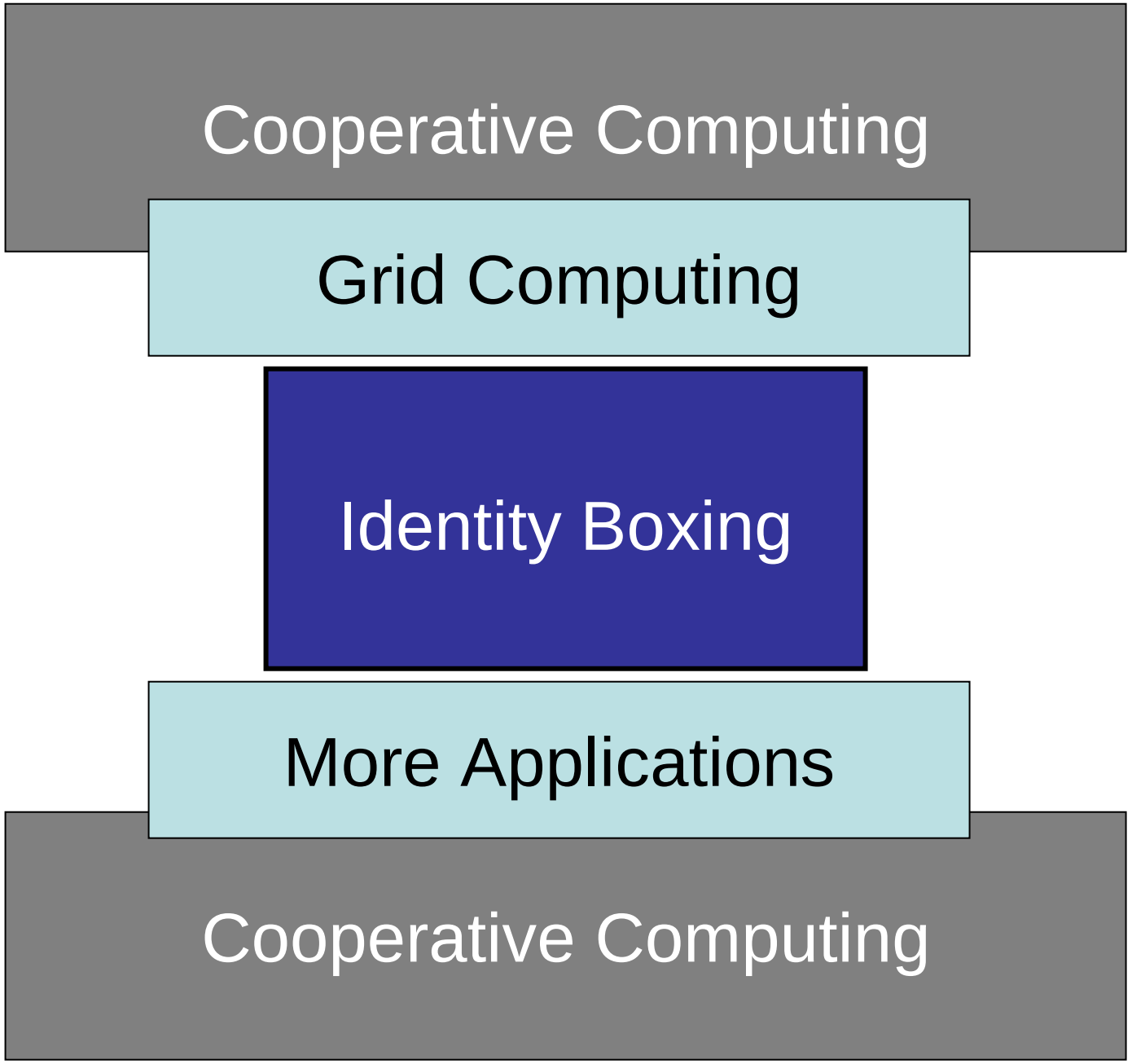
Identity Boxing: A New Technique for Consistent System-Wide Identification

Douglas Thain

http://www.cse.nd.edu/~ccl

UNIVERSITY OF NOTRE DAME


Cooperative Computing
Grid Computing
Identity Boxing
More Applications
Cooperative Computing

Sharing is Hard!

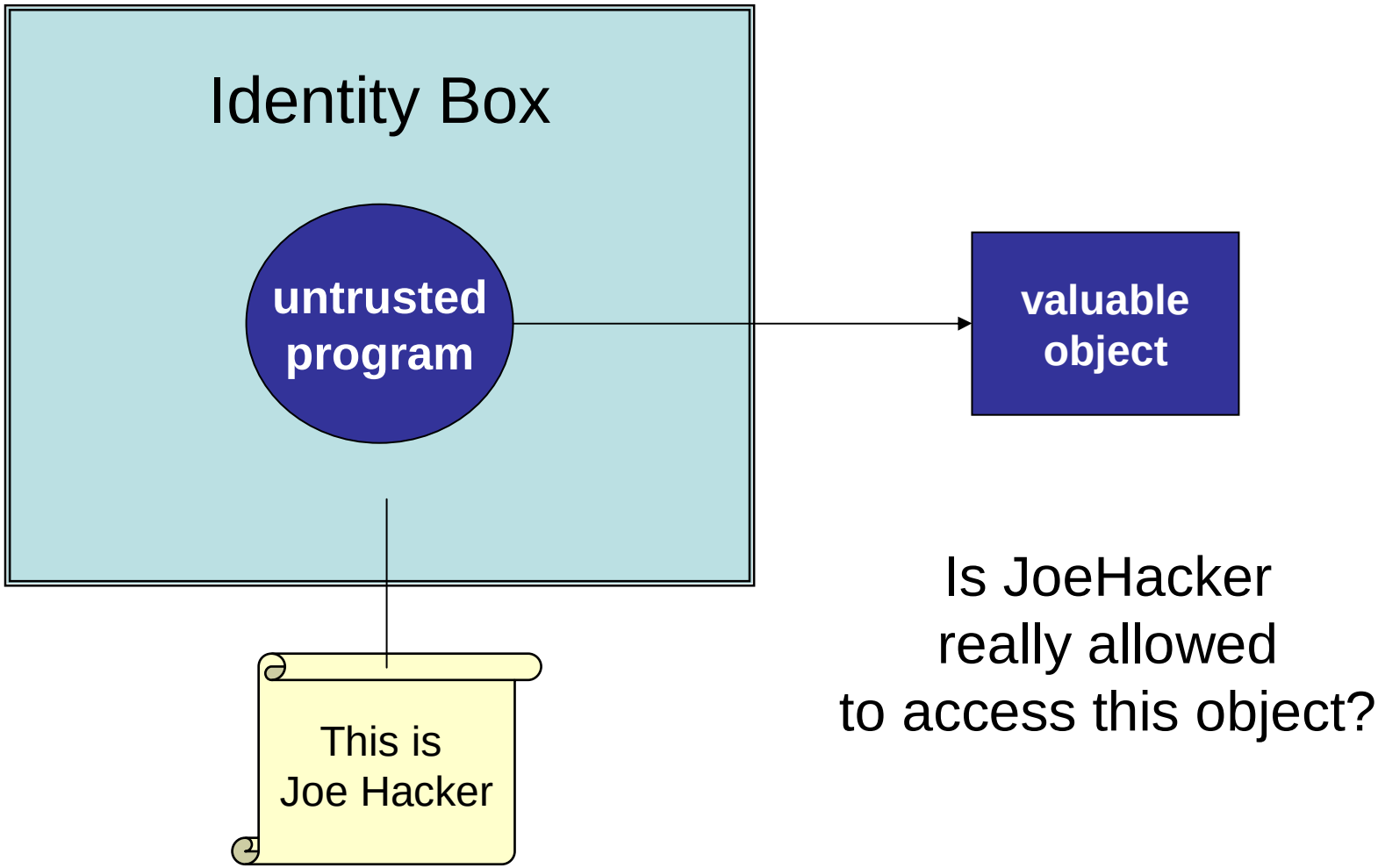
- Despite decades of research in distributed systems and operating systems, sharing computing resources is still technically and socially difficult!

- Most existing systems for sharing require:
  - Kernel level software.
  - A privileged login.
  - Centralized trust.
  - Loss of control over resources that you own.

Cooperative Computing Credo

- Let’s create tools and systems that make it easy for users to cooperate (or be selfish) as they see fit.
- **Modus operandi:**
  - Make tools that are foolproof enough for casual use by one or two people in the office.
  - If they really are foolproof, then they will also be suitable for deployment in large scale systems such as computational grids.

What is Identity Boxing?



Is JoeHacker
really allowed
to access this object?

What are the Applications?

- Visitor in the Office
  - Mutual Isolation for Security/Privacy
  - Return to a Clean Workspace
- Programs Downloaded from the Web
  - Untrusted Programs are Isolated
  - Associate Creds with Programs (Forensics?)
- Large Scale Account Management
  - No Local Accounts: Just Create on the Fly
- **Grid Computing**

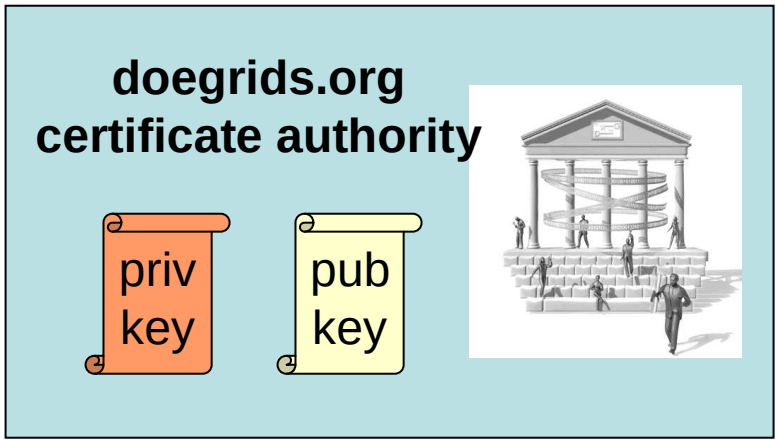
What is the Grid?

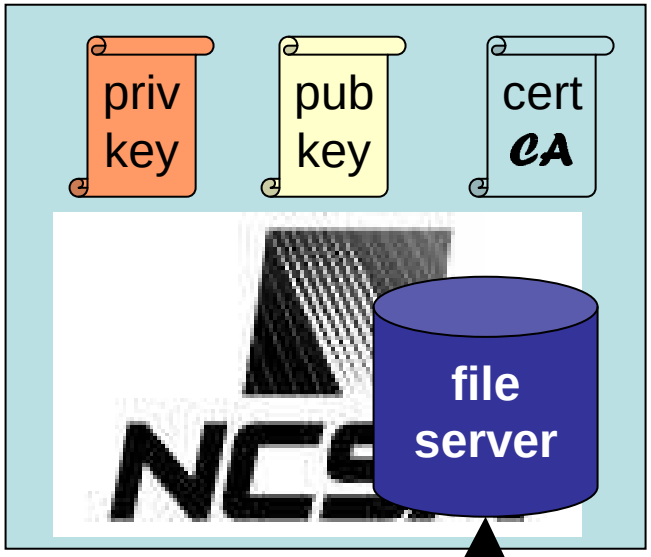
- The Vision:
  - Make large scale computation, storage, and networking as easy as the electric power grid.
- The Reality:
  - Impressive demos by large, skilled teams.
  - Unusable to ordinary scientists and admins.
- The Problems:
  - **Complexity!**
  - Management, Debugging, Scalability

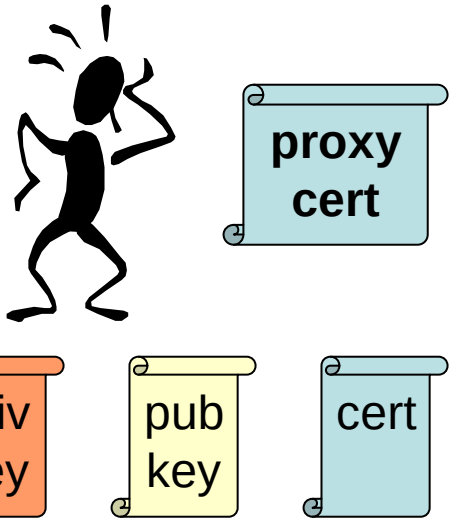
About Grid Credentials

- Generate Public/Private Key Locally
- Generate Certificate Locally
- Send to CA to be Signed
- Login by Exchanging Keys and Certs
- Make Use of Existing Local Unix Account

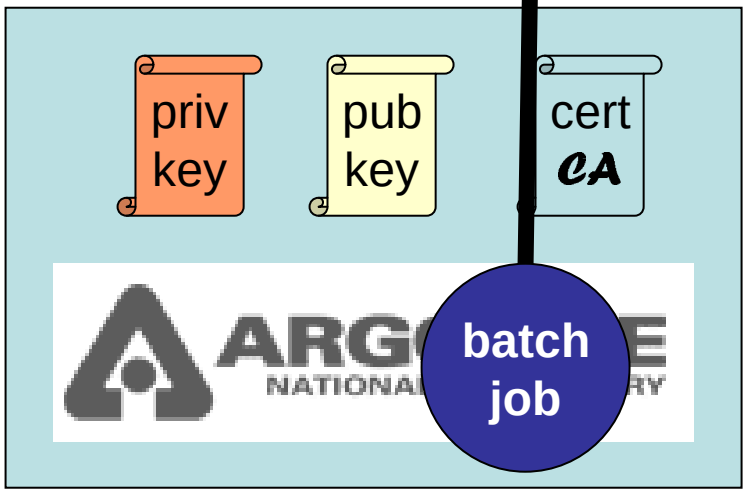
(Something Between Kerberos and PGP)







/O=NotreDame/CN=Douglas Thain



What Does This Get You?

- Single Sign On
  - Single call to grid_proxy_init
  - Same credentials used everywhere.
- Delegation
  - Proxy credentials forwarded from site to site.
  - Remote jobs can authenticate as you.
- Controlled Exposure
  - Proxy certificates expire after a time.
  - Cannot be used for all purposes.

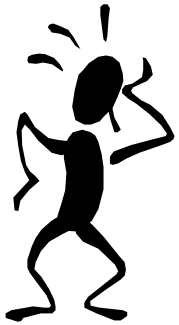
Account Mapping Problem

- Must Map Grid Identity to Unix Account
  - What is my login name here, again?
  - What is Fred's login name here, again?
  - Argh, I don't have a local account here!
    - (Wait until Monday morning to make progress.)
- Some Ugly Solutions Invented
  - Have Poor Sharing Properties
  - Require Large Amounts of Administration

Manual Mapping

Gridmap File
/O=NotreDame/CN=Douglas Thain = thaind
/O=NotreDame/CN=Edward Malloy = monk
/O=UnivNowhere/CN=John Doe = jdoe

grid service admin



proxy cert

/O=NotreDame/CN=Douglas Thain

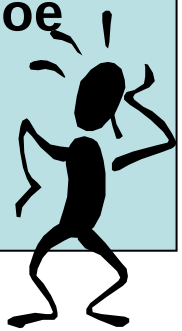
Password File
thaind:546:x:Douglas Thain
monk:309:x:Edward Malloy
jdoe:905:x:John Doe

unix system admin

monk

thaind

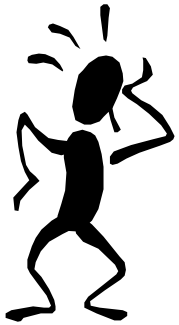
jdoe



Group Mapping

Gridmap File
/O=NotreDame/CN=Douglas Thain = physics
/O=NotreDame/CN=Edward Malloy = chem
/O=UnivNowhere/CN=John Doe = biology

grid service admin

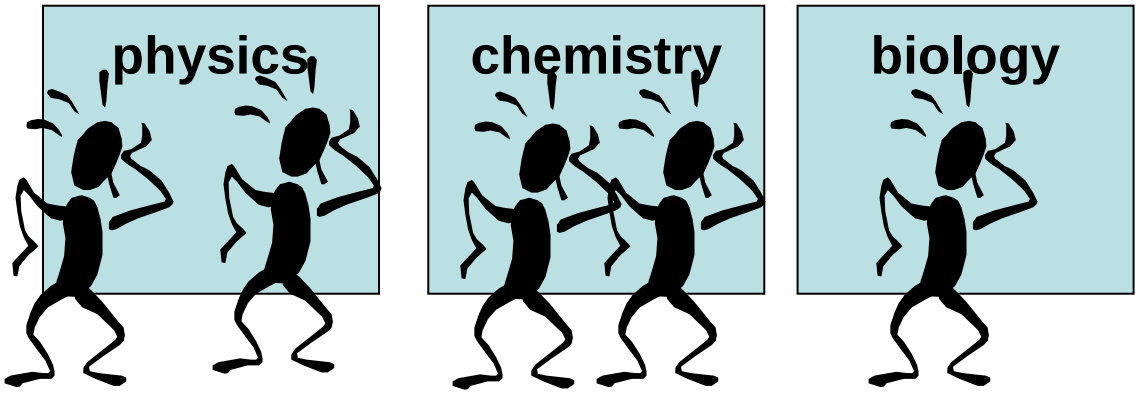


proxy cert

/O=NotreDame/CN=Douglas Thain

Password File
physics:101:x:Physics Group
chem:102:x:Chemistry Group
biology:103:x:Biology Group

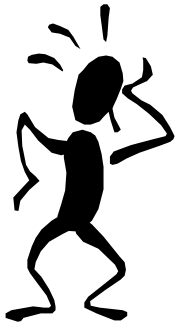
unix system admin



Account Pools

Allowed Grid Users
/O=NotreDame/CN=Douglas Thain
/O=NotreDame/CN=Edward Malloy
/O=UnivNowhere/CN=John Doe

grid service admin



proxy cert

/O=NotreDame/CN=Douglas Thain

Password File
anon1:101:x:Anonymous
anon2:102:x:Anonymous
anon2:103:x:Anonymous

unix system admin

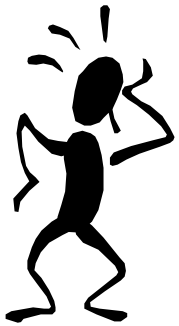
anon1

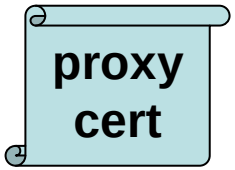
anon2

anon3

Idea:
Forget the Local Account!

Identity Boxing





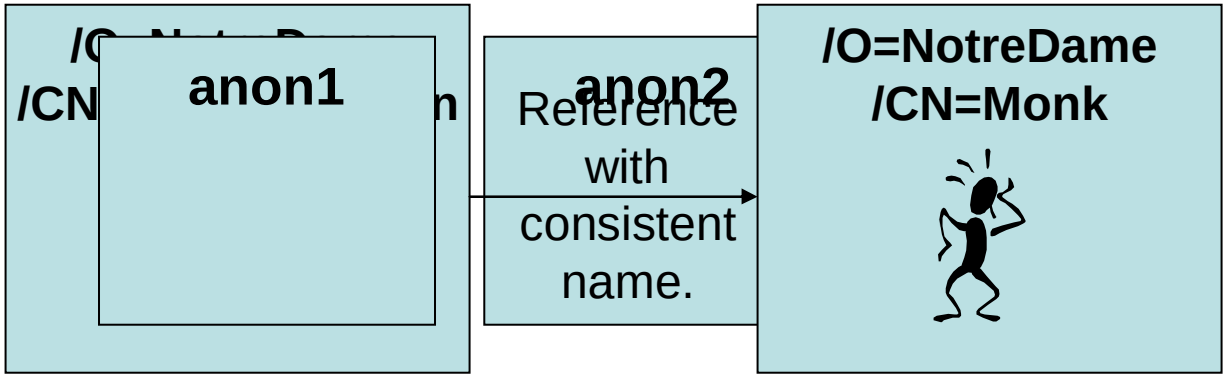
/O=NotreDame/CN=Douglas Thain

Allowed Grid Users
/O=NotreDame/CN=Douglas Thain
/O=NotreDame/CN=Edward Malloy
/O=UnivNowhere/CN=John Doe

grid
service
admin

Password File
anon1:101:x:Anonymous
anon2:102:x:Anonymous
anon2:103:x:Anonymous

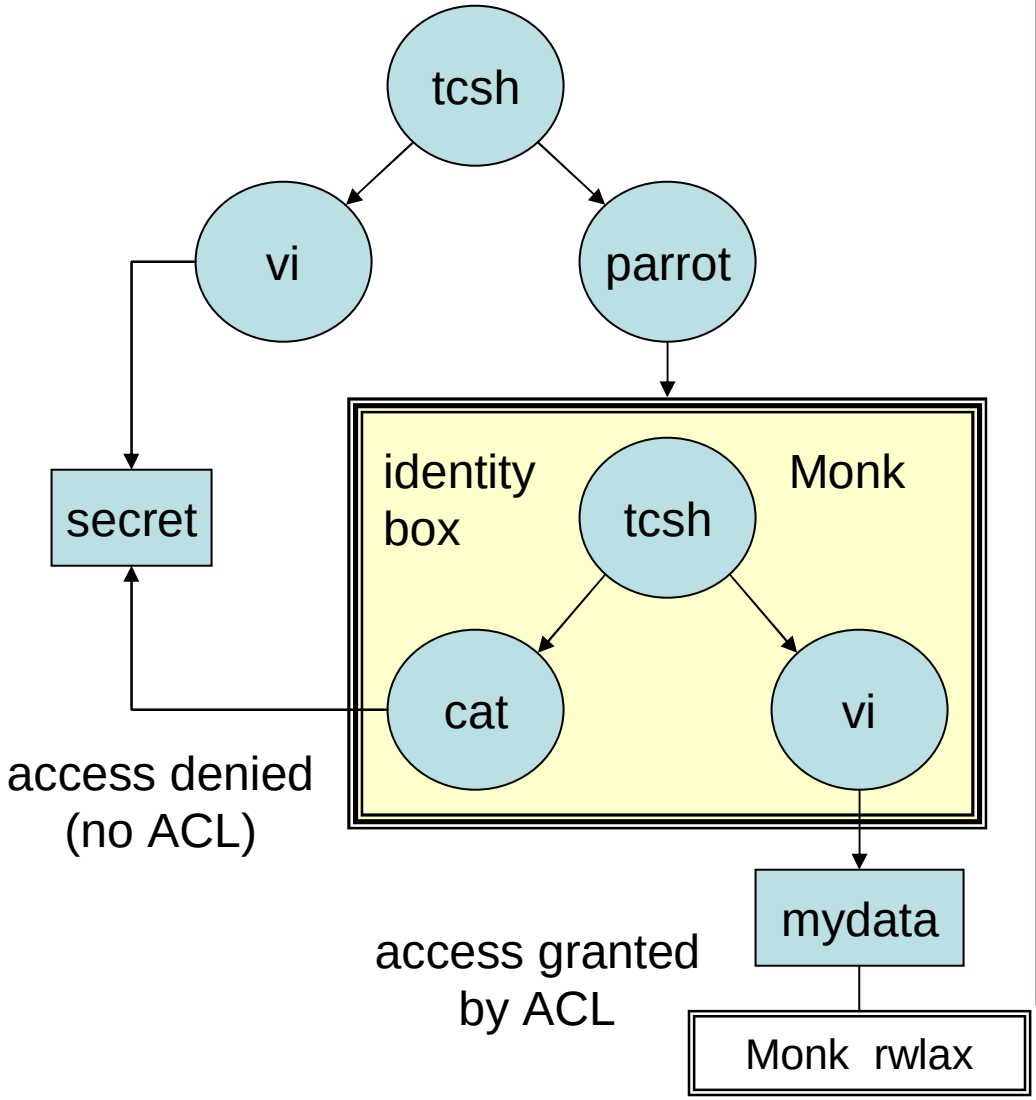
unix
system
admin



Techniques Compared

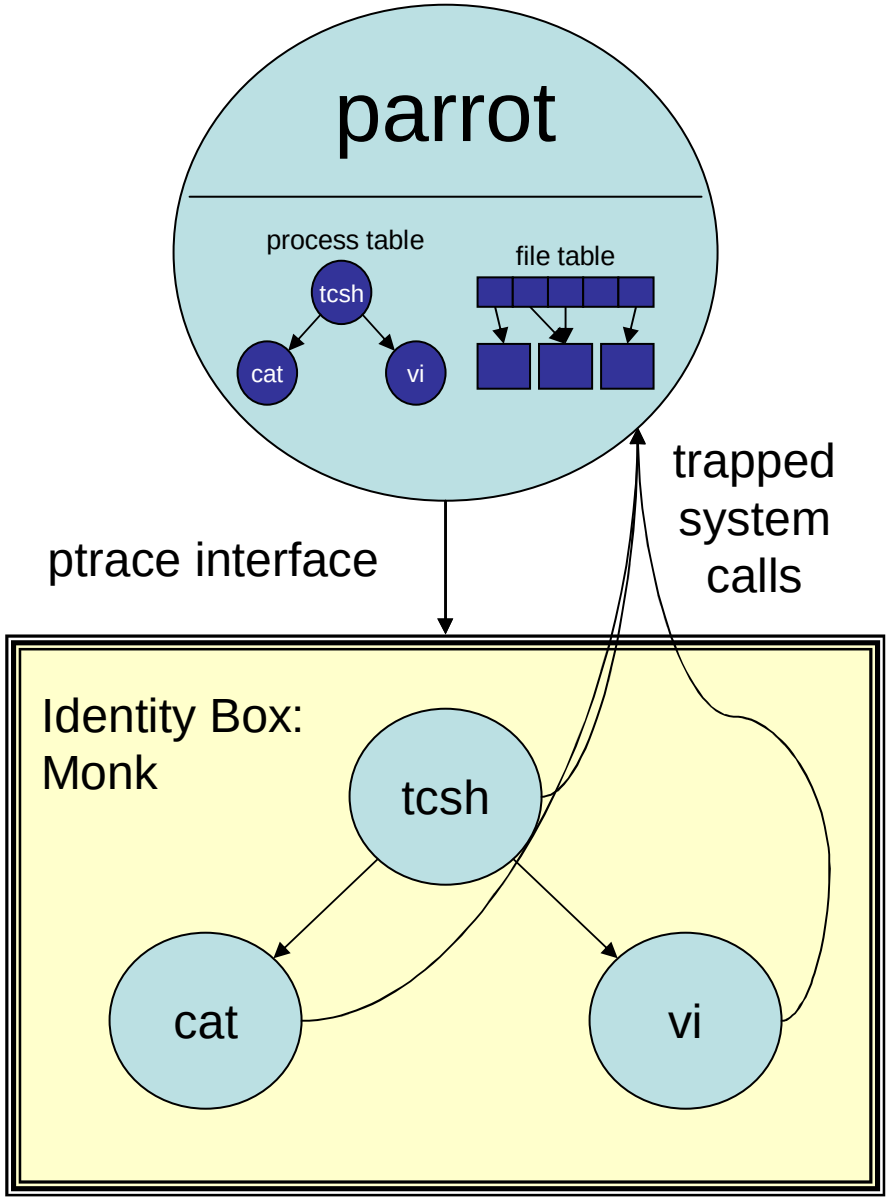
	Privilege Required	**Admin Burden**	**Share Data?**	**Return Later?**
Manual	root	per user	no	yes
Shared	root	per group	fixed	yes
Pool	root	setup	no	no
Identity Boxing	none	none	yes	yes

Identity Boxing in Detail




```
% tcsh
dthain> vi secret
dthain> parrot_identity_box  Monk
Monk> whoami
   Monk
Monk> pwd
/tmp/home.102744/Monk
Monk> cat ~dthain/secret
cat: ~dthain/secret: Permission denied.
Monk> vi mydata
Monk> cat .__acl
     Monk rwlax
Monk> exit
dthain>
```

Parrot



- Like an OS Kernel
  - Tracks procs, files, etc.
  - Adds new capabilities.
  - Enforces owner's policies.
- Delegated Syscalls
  - Trapped via ptrace interface.
  - Action taken by Parrot.
  - Resources chrgd to Parrot.
- Research Platform
  - Distributed file systems.
  - Grid appl. environments.
  - Debugging.
  - Easier than OS coding!

Problem: Storing Identities

- Unix Only Allows for Integer Identities
- Only Root Can Change File Ownership
- Where to Store Long Names:

  /O=University of Notre Dame/CN=Edward Malloy

```
% ls -la
-rwxr-xr-x    1 dthain   users       86317 Feb 28 18:17 chirp_status
-rw-r--r--    1 dthain   users        2870 Dec 14 21:29 chirp_status.c
-rw-r--r--    1 dthain   users        8972 Feb 28 18:03 chirp_status.o
-rw-r--r--    1 dthain   users       23312 Feb 25 17:42 chirp_tool.c
-rw-r--r--    1 dthain   users       36012 Feb 28 18:03 chirp_tool.o
-rw-r--r--    1 dthain   users      132968 Feb 28 18:17 libchirp.a
```

Solution: Directory ACLs

- Add a New File .__acl to Each Directory
- Looks Like an AFS ACL
- Can't Retrofit Entire File System
  - Consider User to be Unix **nobody**

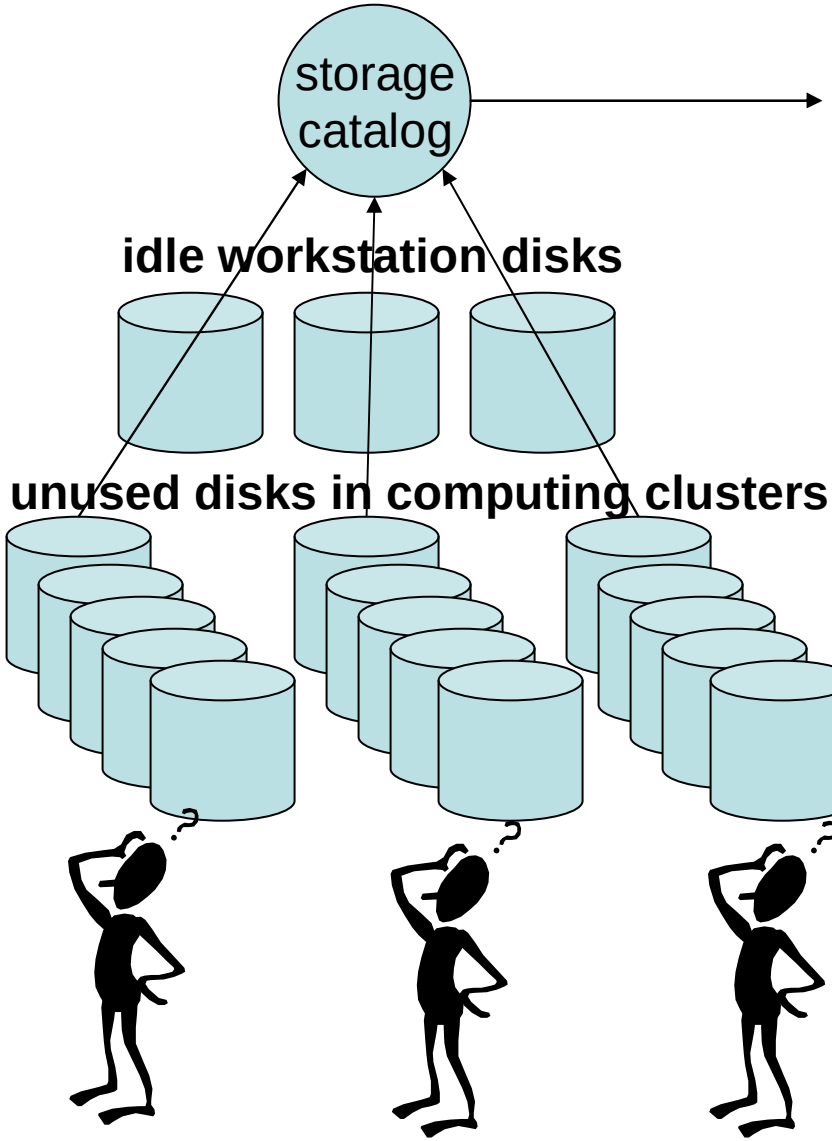
```
% ls -la
-rwxr-xr-x    86317 Feb 28 18:17 chirp_status
-rw-r--r--     2870 Dec 14 21:29 chirp_status.c
-rw-r--r--     8972 Feb 28 18:03 chirp_status.o
-rw-r--r--    23312 Feb 25 17:42 chirp_tool.c
-rw-r--r--    36012 Feb 28 18:03 chirp_tool.o
-rw-r--r--   132968 Feb 28 18:17 libchirp.a
-rw-r--r--       34 Feb 28 05:45 .__acl
```

ACL:
Monk RWLAX
dthain RLX
* RL

Demo Time!

Identity Boxing
in a Distributed System

ND CSE Storage Pool



ccl00.cse.nd.edu storage catalog - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites Media

Address http://ccl00.cse.nd.edu:9097/ Go Links

ccl00.cse.nd.edu storage catalog

text - html - xml - oldclassads - newclassads

5.5 TB available out of 6.6 TB on 102 devices

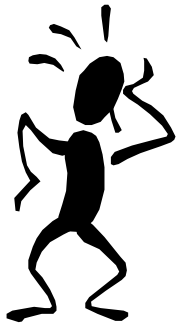
type	name	owner	total	avail	version	
chirp	kamikaze.cse.nd.edu	"curt"	109.2 GB	81.1 GB	2.0.12	detail
chirp	mordoc.cselab.nd.edu	"curt"	34.6 GB	28.0 GB	2.0.12	detail
chirp	ratbert.cselab.nd.edu	"curt"	34.6 GB	28.3 GB	2.0.12	detail
chirp	country.cselab.nd.edu	"curt"	7.8 GB	5.1 GB	2.0.12	detail
chirp	marvin.cse.nd.edu	"curt"	113.1 GB	97.5 GB	2.0.12	detail
chirp	GIPSE-DX-2.cse.nd.edu	"striegel"	31.4 GB	21.3 GB	2.0.12	detail
chirp	grumpy.cse.nd.edu	"curt"	15.7 GB	12.6 GB	2.0.12	detail
chirp	GIPSE-DX-1.cse.nd.edu	"striegel"	31.4 GB	25.0 GB	2.0.12	detail
chirp	ccl04.cse.nd.edu	"dthain"	203.4 GB	190.5 GB	2.0.12	detail
chirp	GIPSE-DP-2.cse.nd.edu	"striegel"	31.6 GB	26.5 GB	2.0.12	detail
chirp	cloud.cse.nd.edu	"curt"	109.2 GB	97.3 GB	2.0.12	detail
chirp	catbert.cselab.nd.edu	"curt"	34.6 GB	28.2 GB	2.0.12	detail
chirp	wombat00.cselab.nd.edu	"dthain"	11.3 GB	2.8 GB	2.0.12	detail

Internet

Data Access

chirp client

hostname:laptop.cse.nd.edu



parrot

xterm + tcsh

globus:/O=NotreDame/CN=Monk

storage owner

chirp server

***/* directory ACL:**

hostname:*.cse.nd.edu RWLX

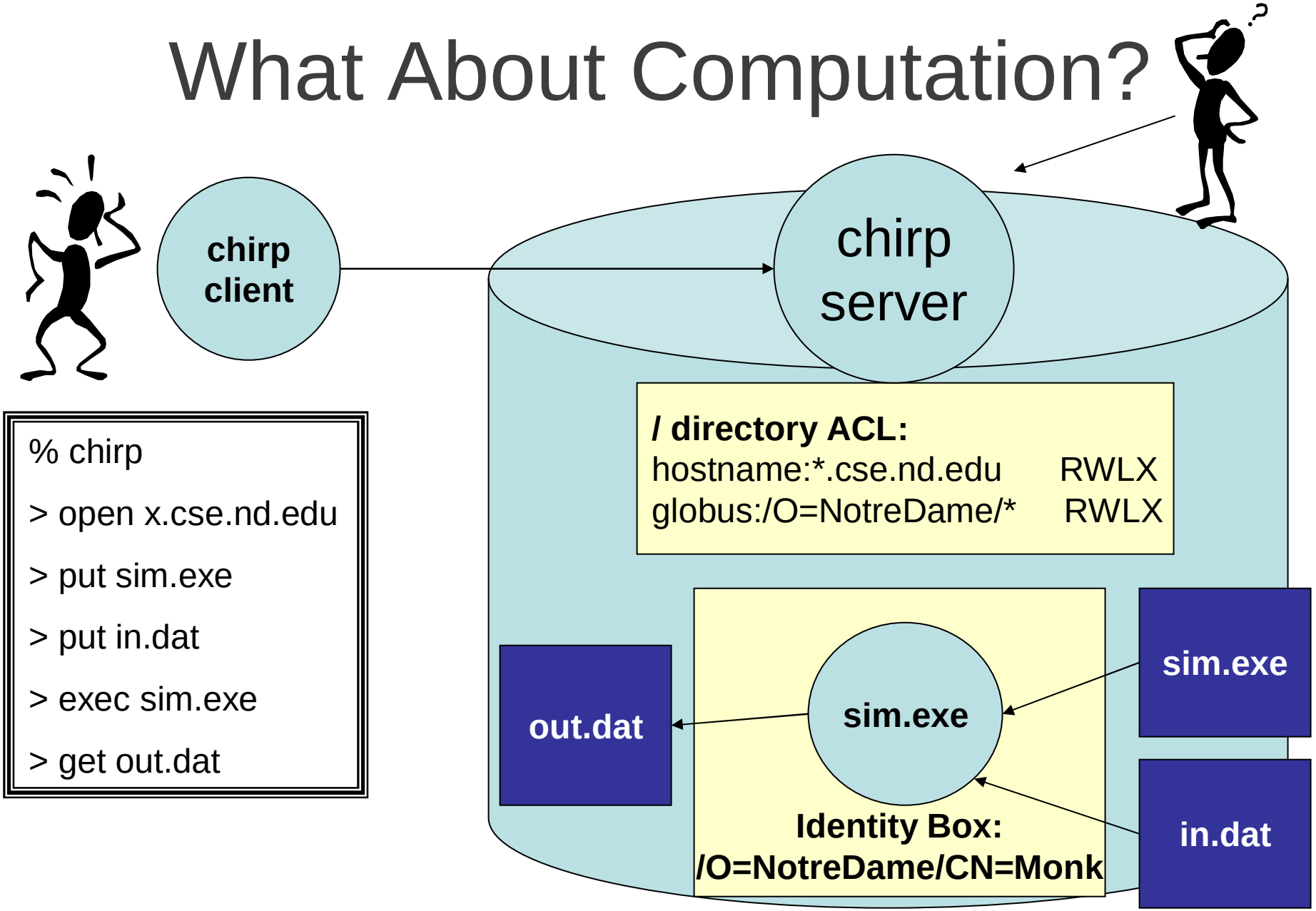
globus:/O=NotreDame/* RWLX

data

data

data

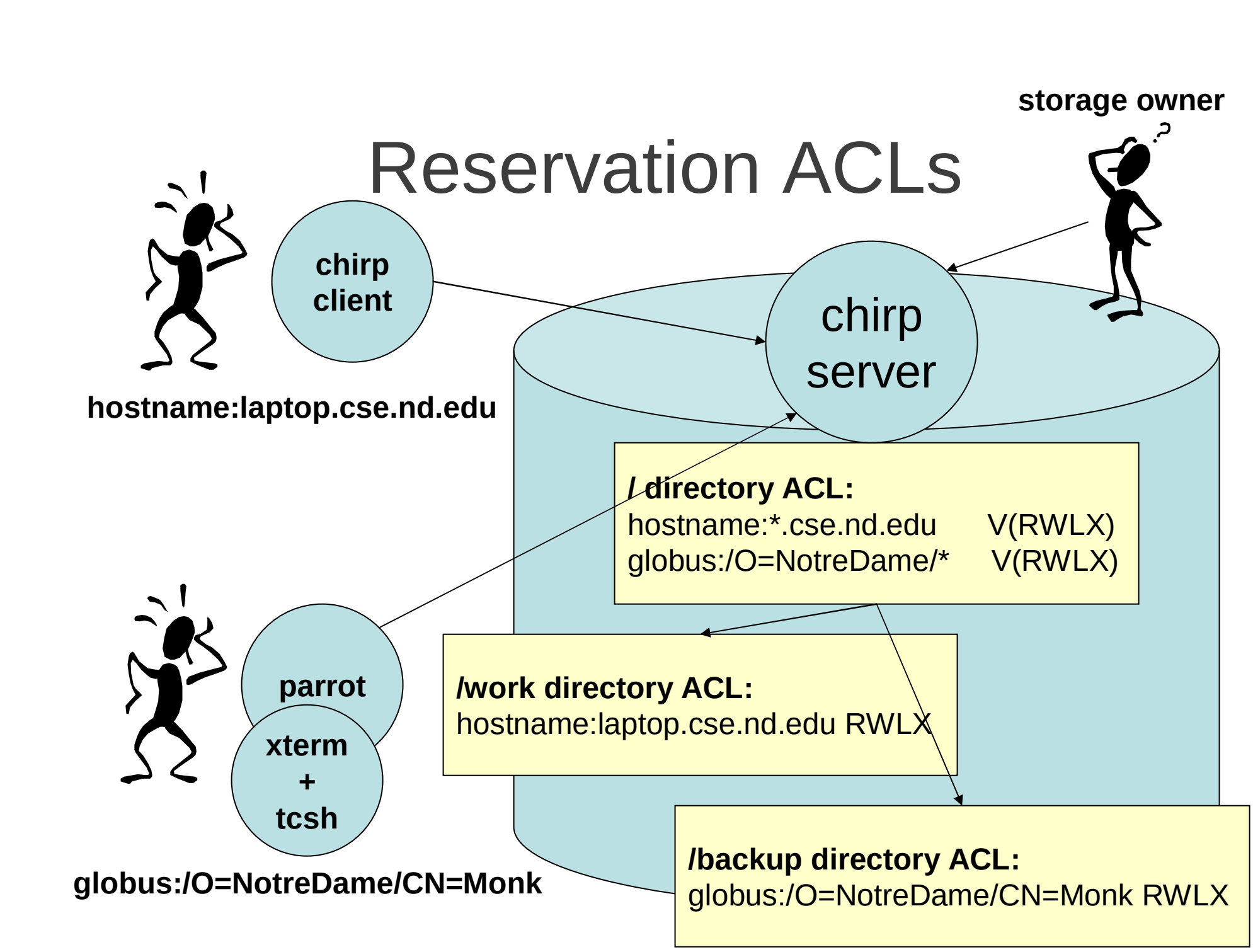
What About Computation?



Reservation ACLs

- Identity Boxing Encourages Wildcarding
  - /O=NotreDame/* can write to this disk
  - Imagine if everyone did!
- Need Facility for Private Workspaces
  - Reservation == Amplification
  - The V bit creates fresh ACLs in subdirs.
  - Example:
    - root ACL: /O=NotreDame/* V(RWLXA)
    - mkdir(/work): /O=NotreDame/Monk RWLXA

Reservation ACLs



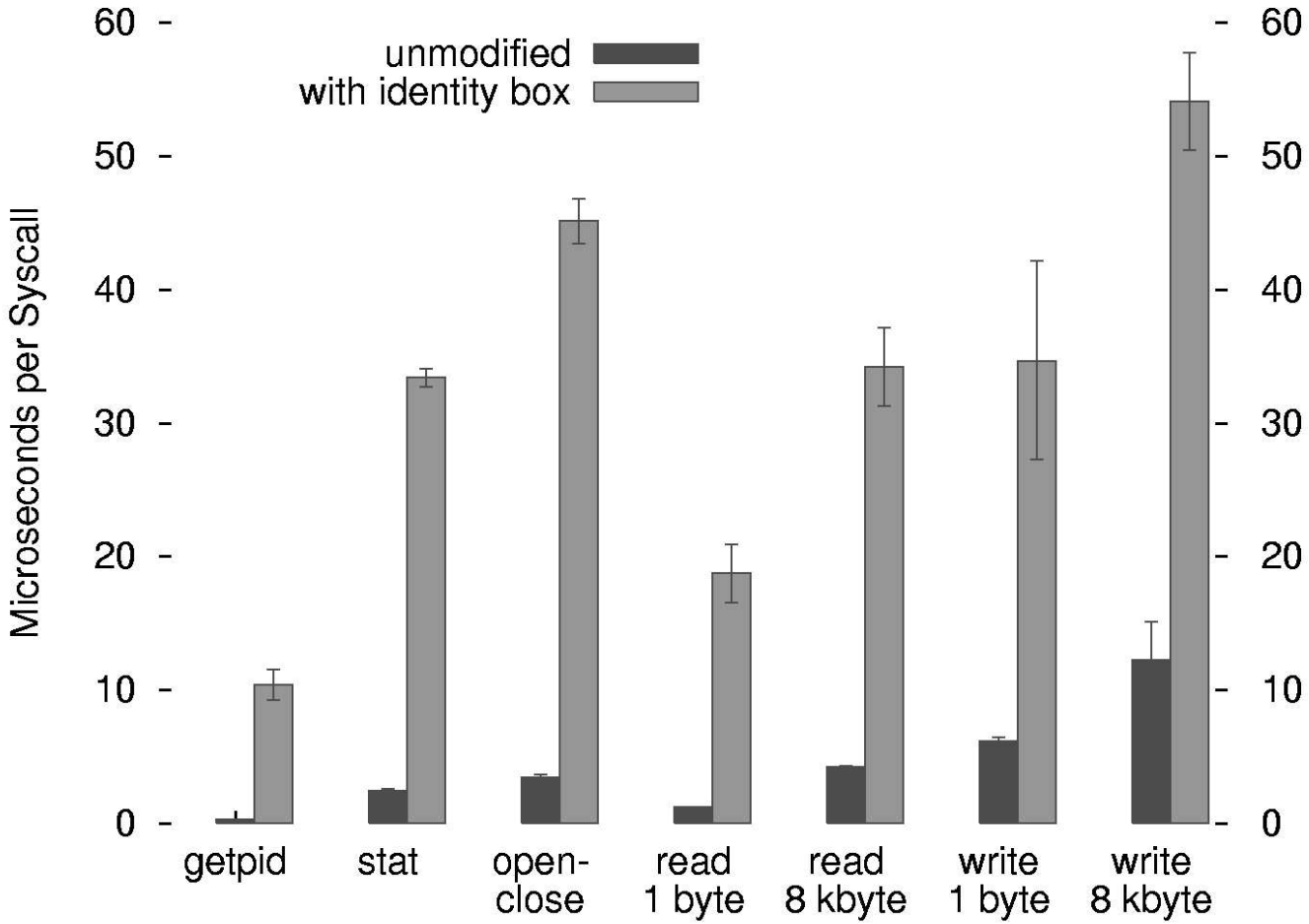
The Big Picture

- Users Employ Consistent Identity
  - Allows owners to deploy interesting policies.
  - Allows users to collaborate in easy ways.
- Example Policies:
  - Anyone at Notre Dame may access this data.
  - Students in my class can execute **only** this executable on the cluster of machines.
  - These project leaders may publish and modify this dataset, but anyone may log in and read it.
  - Researchers in my collaboration may log in and run one of ten appls controlled by our project manager.

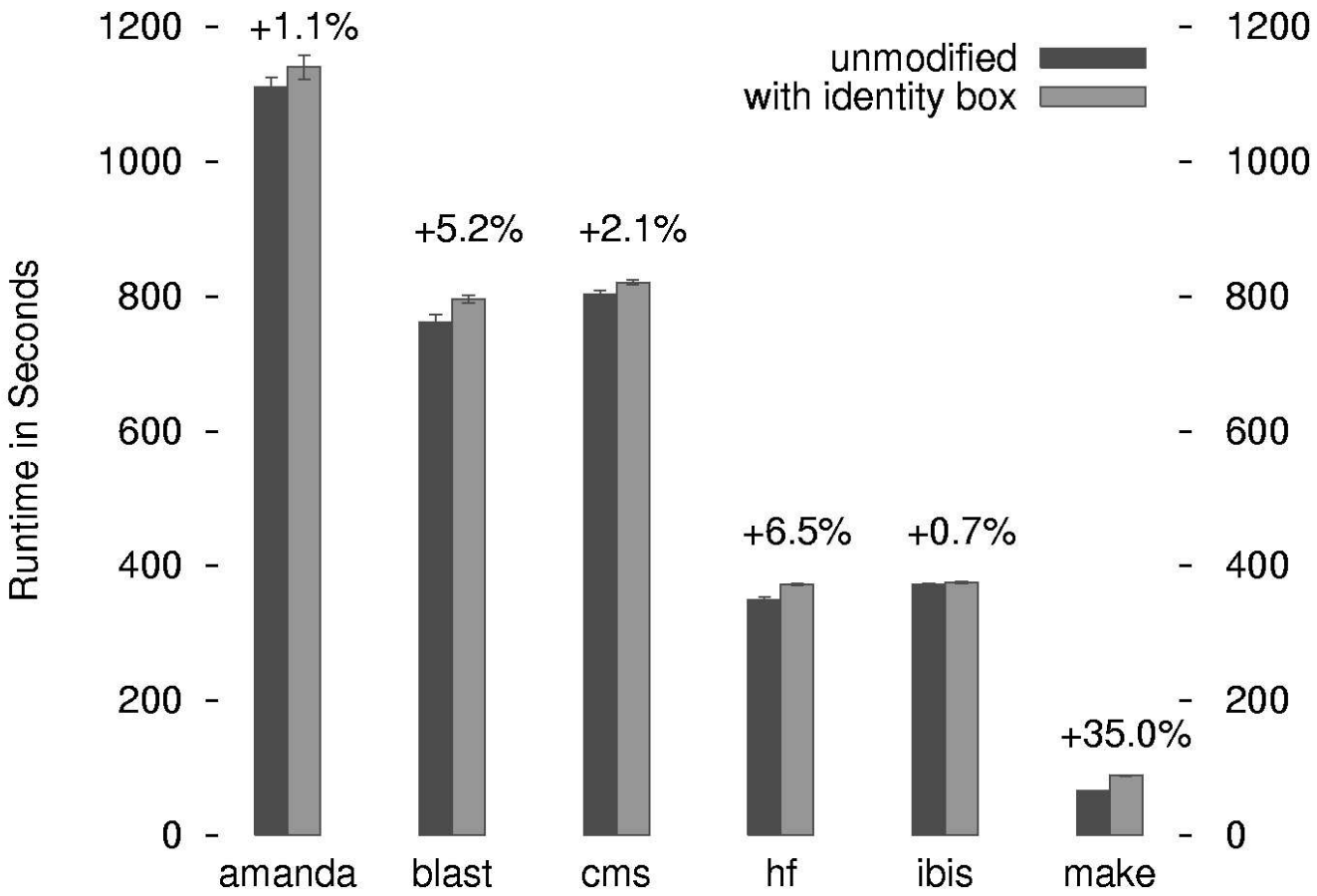
Performance

- Goal: Measure Cost of User-Level Impl
- Microbenchmarks:
  - System calls slowed by order of magnitude.
  - Multiple round trips to service ptrace ops.
- Macrobenchmarks:
  - Scientific apps bound for the grid.
  - AMANDA, BLAST, CMS, HF, IBIS.
  - 0.5 – 6.0 percent slowdown.
  - Make: 35 percent slowdown.

Microbenchmarks



Macrobenchmarks



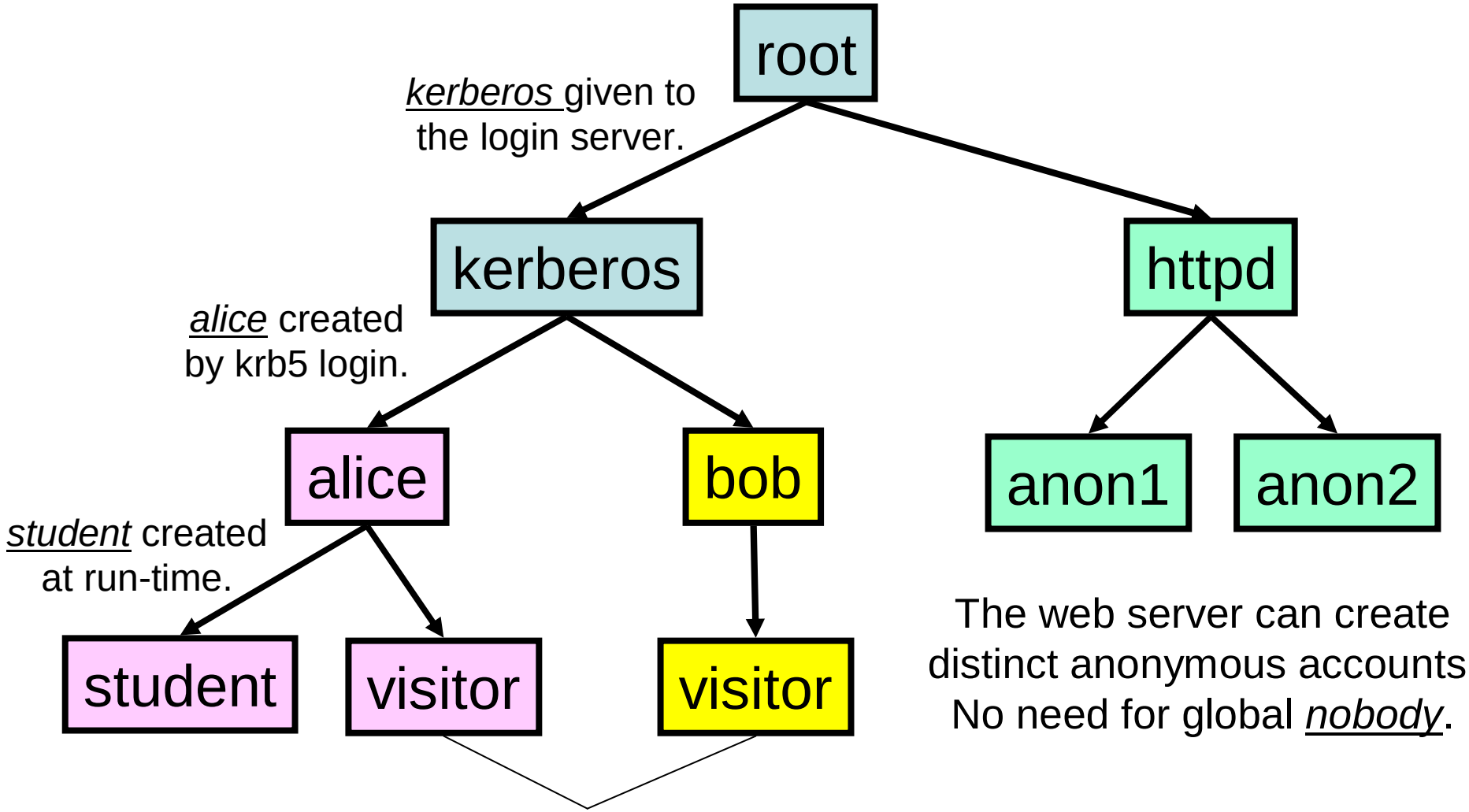
Performance Conclusion:

Identity boxing has acceptable performance for distributed scientific applications...

...especially if it allows us to harness CPUs that would otherwise be unused.

Recapitulation

- Account management is a serious impediment to grid computing.
- Identity boxing allows users to create and destroy protection domains on the fly.
- In a system with identity boxes, there is little or no admin overhead to admitting new users and sharing resources.
- Existing implementation has acceptable performance for scientific applications.


root
kerberos
httpd
alice
bob
anon1
anon2
student
visitor
visitor
kerberos given to the login server.
alice created by krb5 login.
student created at run-time.
The web server can create distinct anonymous accounts. No need for global nobody.
These two users are completely different:
root:kerberos:alice:visitor
root:kerberos:bob:visitor

What Does this Require?

- Operating System Kernel Changes
  - Process and accounting structures.
  - A few new system calls.
- File System Changes
  - Where to store IDs?
  - Who gets charged for space used?
- User-Land Tools
  - How do I manage large user trees?
  - How do I control what sub-users do?

Cooperative Computing Credo

- Let’s create tools and systems that make it easy for users to cooperate (or be selfish) as they see fit.
- **Modus operandi:**
  - Make tools that are foolproof enough for casual use by one or two people in the office.
  - If they really are foolproof, then they will also be suitable for deployment in large scale systems such as computational grids.

More Activities

- Distributed Storage and Data Access
  - GEMS: Grid Enabled Biomolecular Simulation
  - Personal Distributed Filesystems
- Telescopic Debugging
  - Debugging as a distributed query problem.
  - Aspect-Oriented Debugging
- Distributed and Grid Computing
  - Need simulation time for class projects?

For more information:

- Software and Documentation
  - http://www.cctools.org
  - ccl@cse.nd.edu
  - Installed in AFS
- Prof. Douglas Thain
  - http://www.cse.nd.edu/~dthain
  - dthain@cse.nd.edu
  - 356-D Fitzpatrick