mountd, exports, and the pseudo-fs in NFSv4

Tom Haynes

Network Appliance

02/23/2004

thomas@netapp.com

Outline

- What motivates a pseudo-file system?
- Relationship between Pseudo-FS and exports
- What was unique about NetApp’s space?
- Where are the bodies buried?
- How does security/authentication shape the Pseudo-FS?
- Feature creep or -actual

NetworkAppliance®

Motivation for Pseudo-FS in NFSv4

- **Firewalls**
  - **One well defined port for piercing security**
  - **Get rid of minion protocols**
  - **This means you mountd!**
- **How do we determine what is exported?**
  - **Single File system:**
    - **Do the Root**
  - **Multiple File systems:**
    - **What is the root?**
- **Need to get our hands on the first File Handle**

NetworkAppliance®

The glue that binds

- Could use automounter maps
  - Lives on the client and not the server
  - Requires more ports to be opened
- Can keep it in memory
  - Construct a consistent virtual file system
    - Filehandles are the same after a reboot
    - All attributes are phony, but realistic
  - Pseudo-FS ties together disjoint real-FS
  - Can go from one PUTROOTFH to every other FH via successive sequences of LOOKUP

NetworkAppliance®

The exports shape the Pseudo-FS

- The job of the pseudo-FS is to get the client to real storage which is exported
- The shape of the pseudo-FS is determined by
  - The ROOTFH
  - The high level export points
- The pseudo-FS is not just an entry for ‘/’
  - We’ll touch on this in later examples

NetworkAppliance®

A short diversion in ONTAP history

- Appliance
  - Strictly a server
  - No mounting of external file systems
- First offerings had 1 volume
  - / was the physical root of the system
  - /etc was for configuration information
- Snapshots are a pseudo-FS
- Can export a descendant

NetworkAppliance®

… multi-volume support

- Majority of installations had single volumes
- Name space was stitched together with a pseudo-FS entry of ‘vol’
- Customers wanted automounts of the form:
  - filer:/
  - filer:/home
  - filer:/etc
- Not:
  - filer:/vol/vol0

NetworkAppliance®

… What does / mean?

- Decided that / would refer to the “root” volume
  - The volume which contains the boot image
  - The volume which contains /etc
    - Configuration
    - Logging
- Need to be able to determine the canonical pathname
  - Symlinks
  - Root volume name

NetworkAppliance®

… Snapshots are a pseudo-FS

- **Snapshots provide online read-only clones of the active file system**
- **".snapshot"**
  - **not in the active file system**
  - **cobble it up in the *readdir()***
- **It ties the active file system to the inactive snapshot file systems**

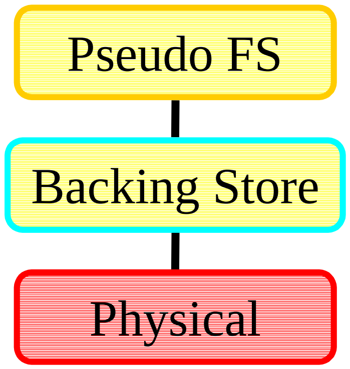
Ramifications on NetApp’s design

- “/vol” is really our ROOTFH
  - ROOTFH for ‘/’
  - VOLFH for ‘vol’
- ‘/’ means something different in v2/v3 and v4
  - Root volume versus Pseudo-FS root
- Inode #64 is the root of a volume, so clients descending into “/vol” have to be wary

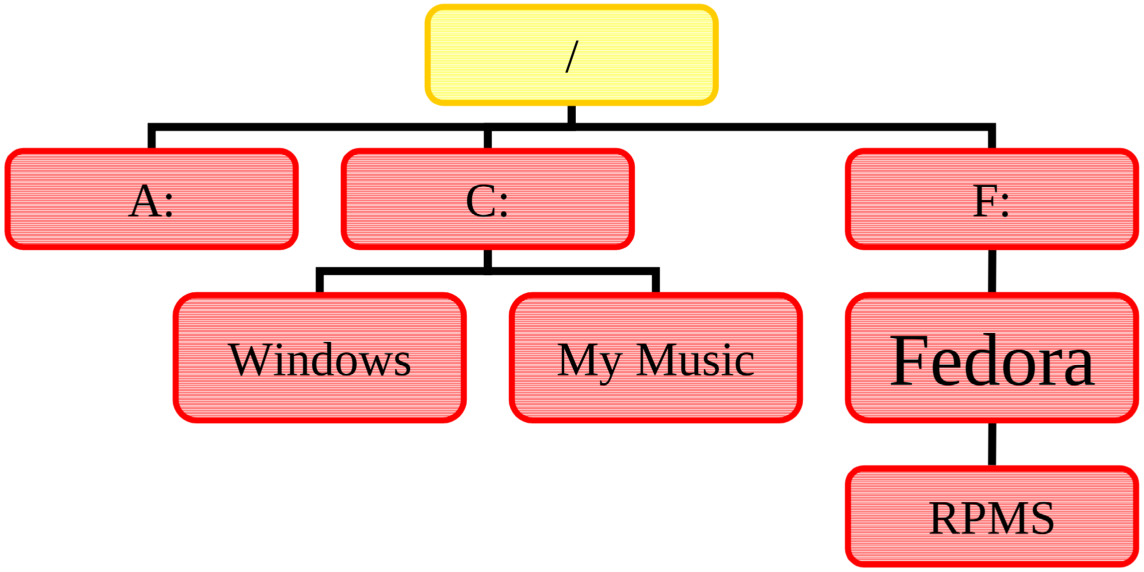
NetworkAppliance®

Node Legend

- Pseudo-FS: Really virtual, no physical storage
- Backing Store: In the pseudo-FS because it is not exported, but a child is exported
- Physical: In the real-FS



Sewing together a MSDOS machine



Describing the MSDOS Example

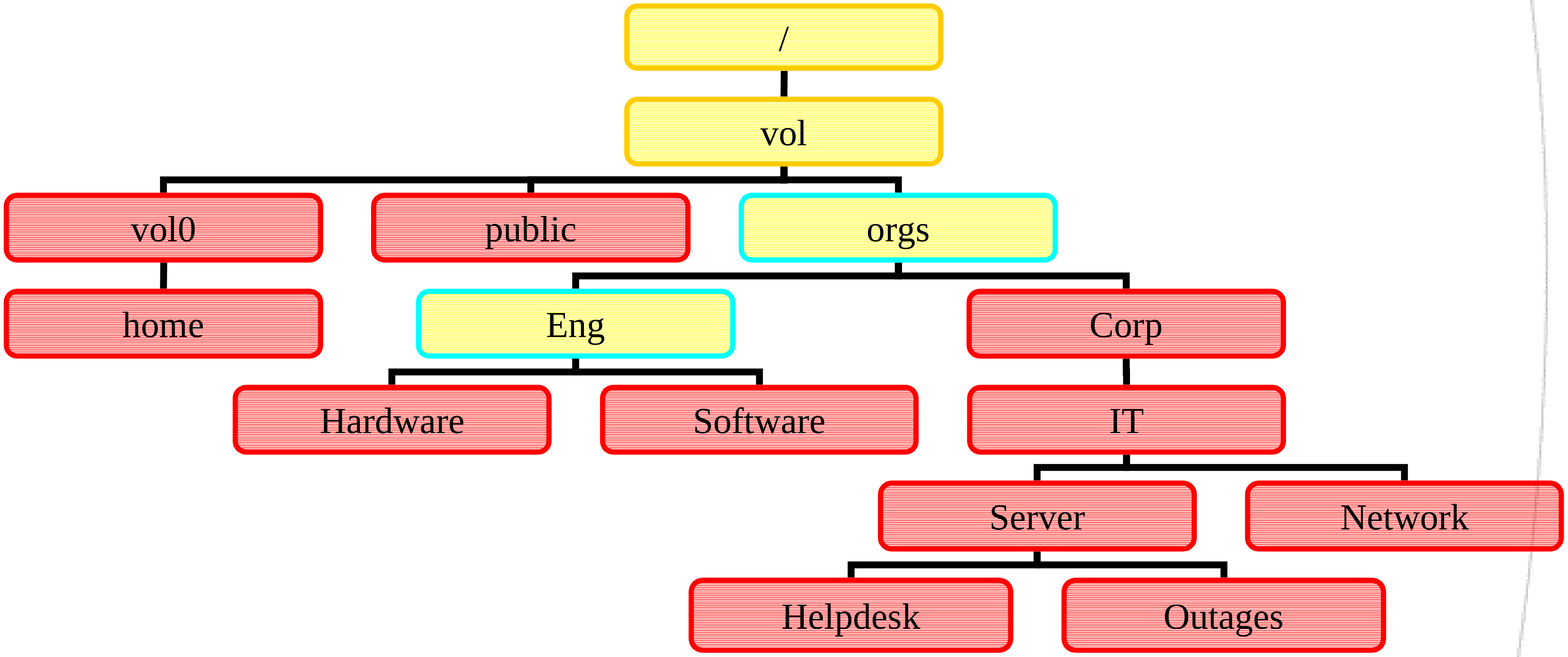
- Each drive is assigned a letter
- Pseudo-FS puts a global root above each drive
- Can mount:
  - Killbill:/F:/Fedora
- Note that Killbill:/F: is evidently a CDROM and we might want to use volatile file handles.

NetworkAppliance®

Export table for the filer: maytag

```
/vol/vol0  -sec=krb5,rw=@trusted,root=@trusted
/vol/vol0/home –rw
/vol/public –rw,anon=0
/vol/orgs/Eng/Hardware  -rw=10.56.10.0/24,anon=0
/vol/orgs/Eng/Software  -rw=10.56.11.0/24,anon=0
/vol/orgs/Corp   -sec=sys:krb5:krb5i:krb5p,ro
/vol/orgs/Corp/IT  -sec=krb5:krb5i:krb5p,rw=10.55.9.0/24
/vol/orgs/Corp/IT/Server/Outages  \
   -sec=krb5,krb5i,krb5p,rw=@it,root=@it,sec=sys,ro
/vol/orgs/Corp/IT/Server/Helpdesk  \
   -sec=sys:none:krb5:krb5i:krb5p,rw
/vol/orgs/Corp/IT/Network \
   -sec=krb5p,rw,sec=none:krb5i,anon=1066,ro
```

Sewing together a Filer: Today



Describing the Filer example

- **Not exported:**
  - **/vol/orgs**
  - **/vol/orgs/Eng**
  - **/vol/orgs/IT/Servers**
    - **Red because of inheritence**
    - **/vol/orgs/IT is exported**
    - **Children are accessible**
- **Inheritence issue may be NetApp specific**

Reality Bites

“Can go from one PUTROOTFH to every other FH via successive sequences of LOOKUP”

- Can just a LOOKUP traverse the file system?
- Or do we have to negotiate both the pseudo and real file systems?
- The pseudo-FS needs to be
  - Client specific
  - Security flavor specific

NetworkAppliance®

Exports further shape Pseudo-FS

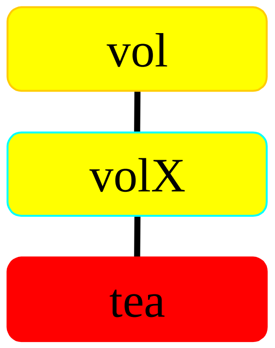
- Security flavor:
  - export is krb5
  - client has krb5i
- Access list – client name spudder
  - /vol/parent –rw=-spudder no access
  - /vol/parent/child –rw=spudder access
- Inheritance
  - /vol/parent -rw=spudder access
  - /vol/parent/child -rw=-spudder access? No!
  - Access should not be granted to spudder if it comes down the path

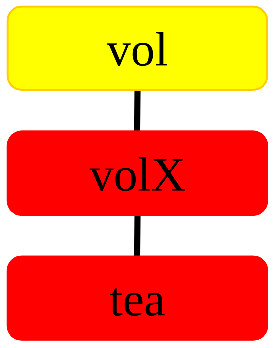
Our first implementation is very static

- Can have pseudo-FS nodes which can be mounted on
  - Recall ONTAP is an appliance, no real prior concept of mounting
  - An export can turn a pseudo-FS to a real-FS node
  - An unexport can turn a real-FS to a pseudo-FS one
- Other than that, pseudo-FS entries do not change

Touching entries – pseudo to real

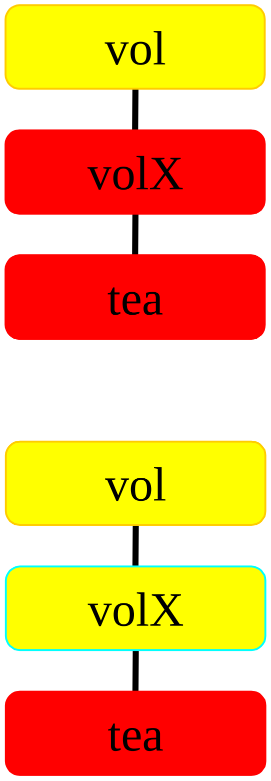
```
>exportfs
/vol/volX/tea -ro
>exportfs -p rw,anon=0
   /vol/volX
>exportfs
/vol/volX/tea –ro
/vol/volX –rw,anon=0
```





Touching entries – real to pseudo

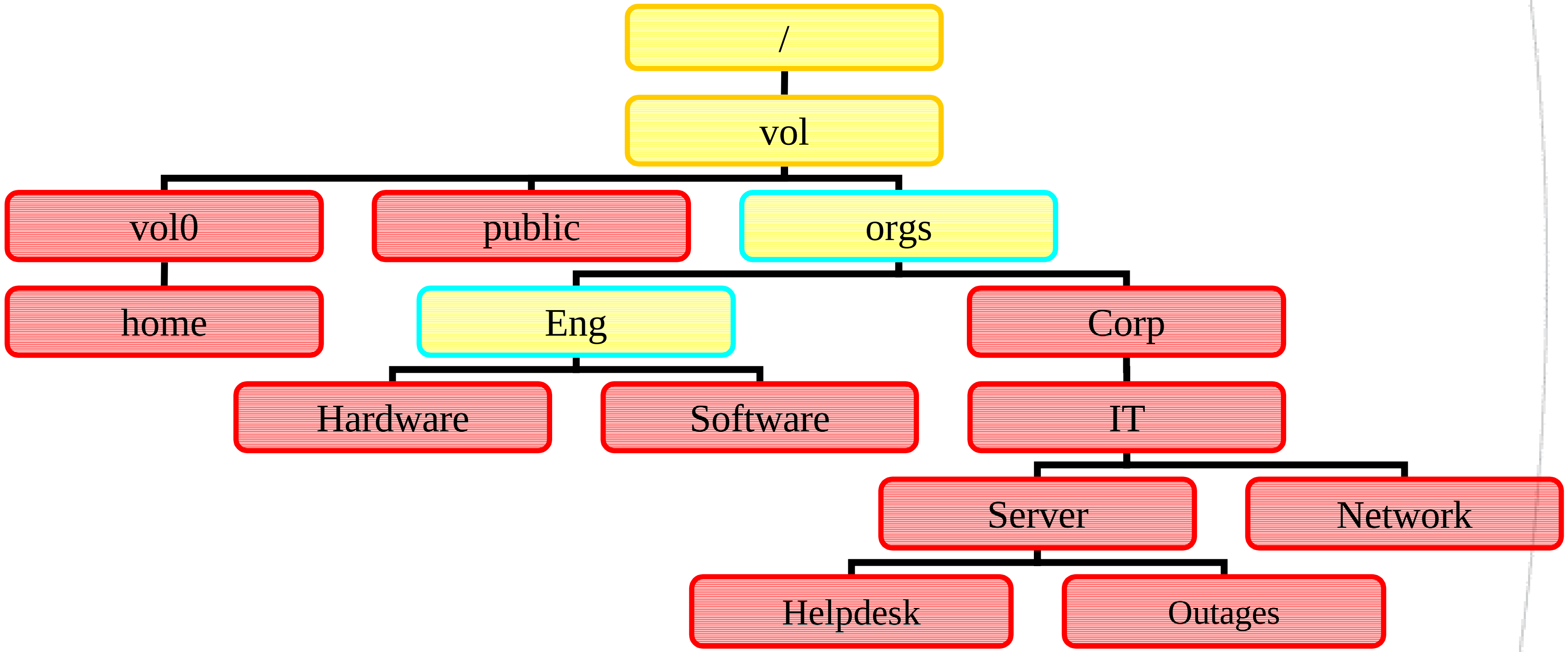
```
>exportfs
/vol/volX/tea –ro
/vol/volX –rw,anon=0
>exportfs -u /vol/volX
>exportfs
/vol/volX/tea -ro
```



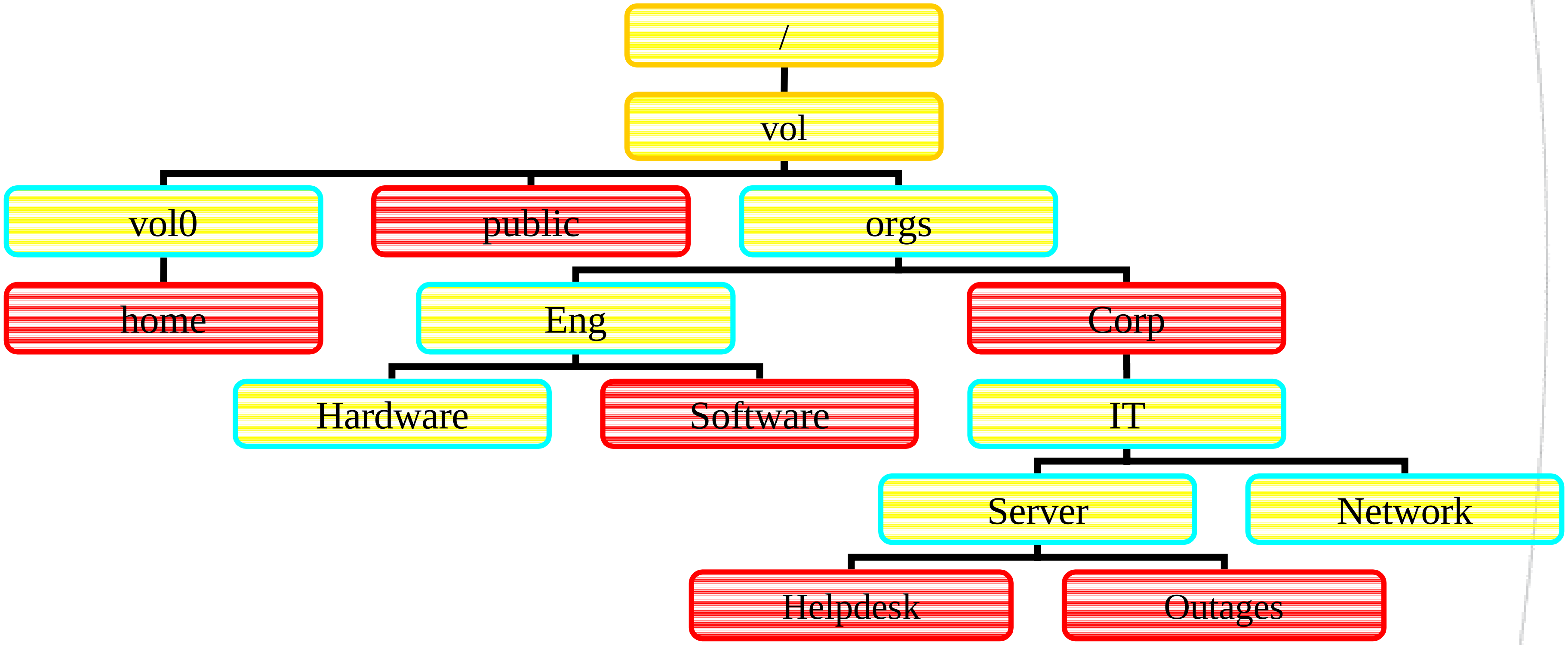
spudder

- spudder.eng.netapp.com
- 10.56.11.13
- Not in the netgroups
  - trusted
  - it

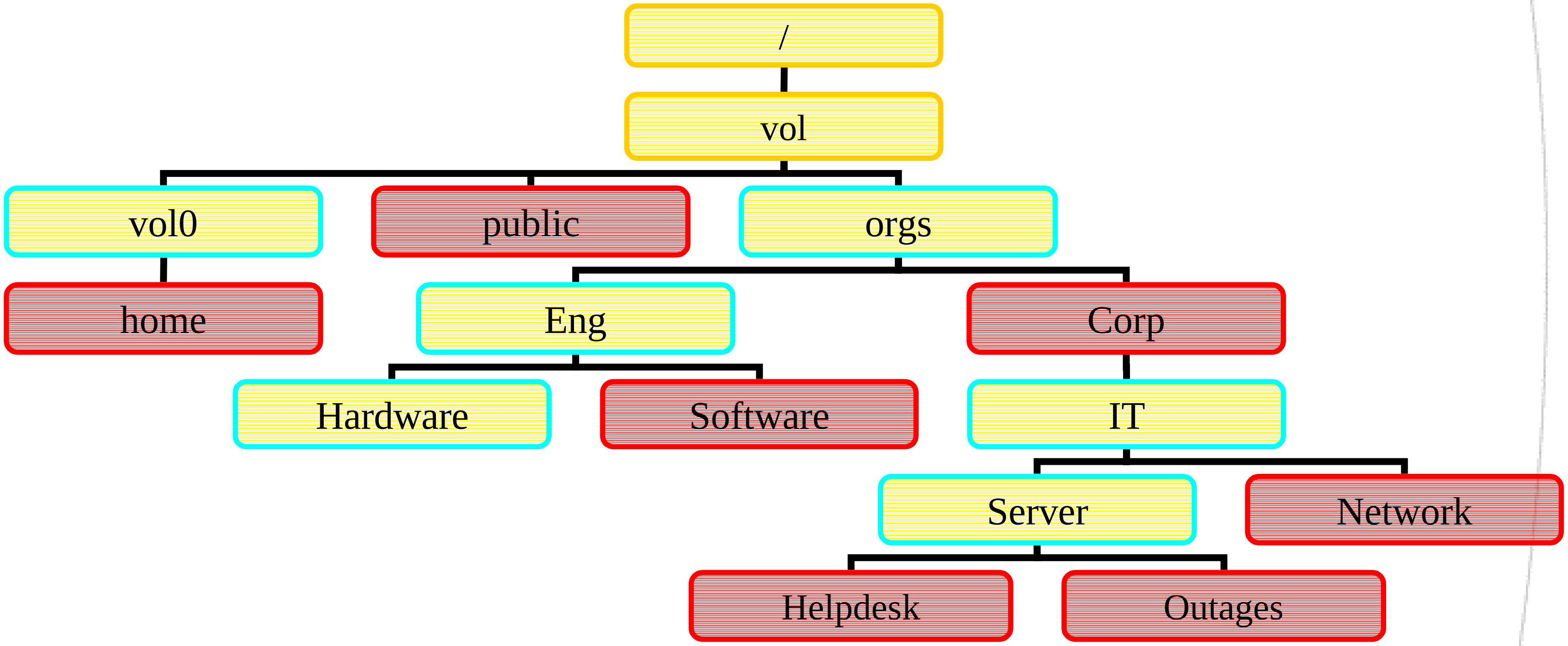
Static Pseudo-FS



Spudder and sec=sys



Spudder and sec=krb5i



Dynamic flipping of real to pseudo FS

- Detection of Inheritence
  - /vol/orgs/IT
- Security flavor mismatch
  - /vol/orgs/IT/Network
- Access list denial
  - /vol/orgs/Eng/Hardware

NetworkAppliance®

-actual directive

- Migrate virtual filer across physical filers
  - Vfiler’s root volume is in the qtree of a physical volume
- Don’t want to force admins to touch every client

/vol/vol2/vf1 –actual=/vol/vfilers/vf1,rw

- Admins don’t want to touch every client
  - /etc/[v]fstab
  - FHs are out there

A path to the physical storage

- Exports are known by their advertised name
- Sometimes the advertised name matches the physical name
- -actual allows us to
  - Avoid an inode lookup
  - Provide aliases to paths
  - Move storage

Learning from our customers

- Customer used –actual out of context
  - /nfs/Engineering –actual=/vol/eng,rw
  - /nfs/CS –actual=/vol/corp/CS,rw
  - /nfs/IT –actual=/vol/corp/IT,rw
- I.e., an attempt at an AFS style namespace
- Can use Pseudo-FS to store global namespace hints

NetworkAppliance®

Migration

- When we migrate, modify the export to reflect

/vol/data/thesis -actual=arena:/vol/jukebox/tdh/thesis

  - Do we provide an access list on who gets to know it was moved?
  - How do we determine which FHs match this export?
- Could use –migrate, but host name clues us into this fact
- Need to be careful about what happens when the path is reused

NetworkAppliance®

Referrals

- Don’t need to worry about the file handle
- Do need to worry about name space collision
- Can use same syntax:

```
/nfs/archer.netapp.com/test -actual=bowman:/vol/test
```

Replication

```
/baseball –actual=/vol/sports/baseball,ro,nosuid,\
   replica=europe.netapp.co.uk:/vol/dnfs/sports/baseball:\
   asia.netapp.com.au:/vol/dnfs/sports/baseball
```

- **Watch out for the complicated use of ‘:’**

Replication: Issues

- **Solely replication or also referrals?**

/tennis –replica=boston:/vol/tennis:dallas:/vol/tennis

- **What happens if we migrate here?**

/rams -actual=stlouis:/go/rams,replica=la:/big/whiners

NetworkAppliance®