NFSv4 Co-existence with CIFS in a Multi-protocol Environment

John Hayden

EMC Corporation

jhayden@emc.com

NFSv4 and CIFS

- The Forcing Function…
- Commonality and Interoperability
  - Permissions
    - Access Control and Permissions Checking
    - ACLs
  - Locking
    - EMC Model
    - Oplock / Delegations
  - Namespace
    - DFS / FS_LOCATION
    - DFS and Symbolic Links
- Challenges
  - User Mapping
  - Kerberos in Delegated Environments

Why?

- CIFS versus NFS
  - Often falls along the lines of Desktop versus Workstation
  - Desktop versus Infrastructure
- Drawing from our customer base

  (NFSv3 vs. CIFS)
  - > 60% have coexistence
    - NFS & CIFS on the same server
  - > 25% are using the same filesystem

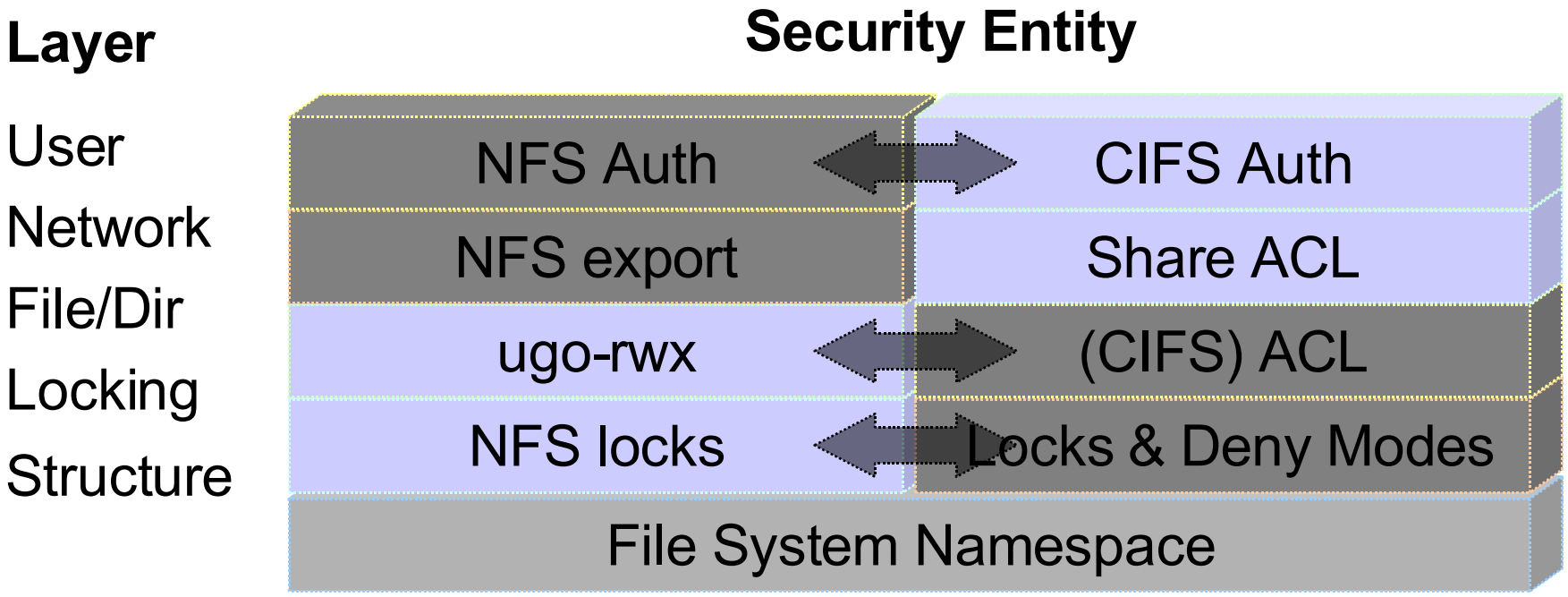
Trends and Use Cases -

- Many customers driving to single sign-on, single directory
- Multi-protocol home directories
- Collaborative environments, including software development
- Manufacturing and design
  - One data set, multiple application platforms
- Infrastructure systems, processing computers are UNIX based, clients are Windows-based

NFSv4 and CIFS

- Commonality and Interoperability
  - The Celerra Security Model
  - Permissions
    - Access Control and Permissions Checking
    - ACLs
  - Locking
    - EMC Model
    - Oplock / Delegations
  - Namespace
    - DFS / FS_LOCATION
    - DFS and Symbolic Links

Security Model



Security Model (pre-v4)

File System Access Policy	Access Protocol	NFS		CIFS		Change in one set reflected in the other?
	Permissions Checked	UNIX	ACL	UNIX	ACL	
	NATIVE	✓	X	X	✓	No
	NT	✓	✓	X	✓	
	UNIX	✓	X	✓	✓	
	SECURE	✓	✓	✓	✓	
	MIXED	Whichever permissions were set last				Yes (modifying one set of permissions overwrites the other)
	MIXED_COMPAT					

Security Model for v4

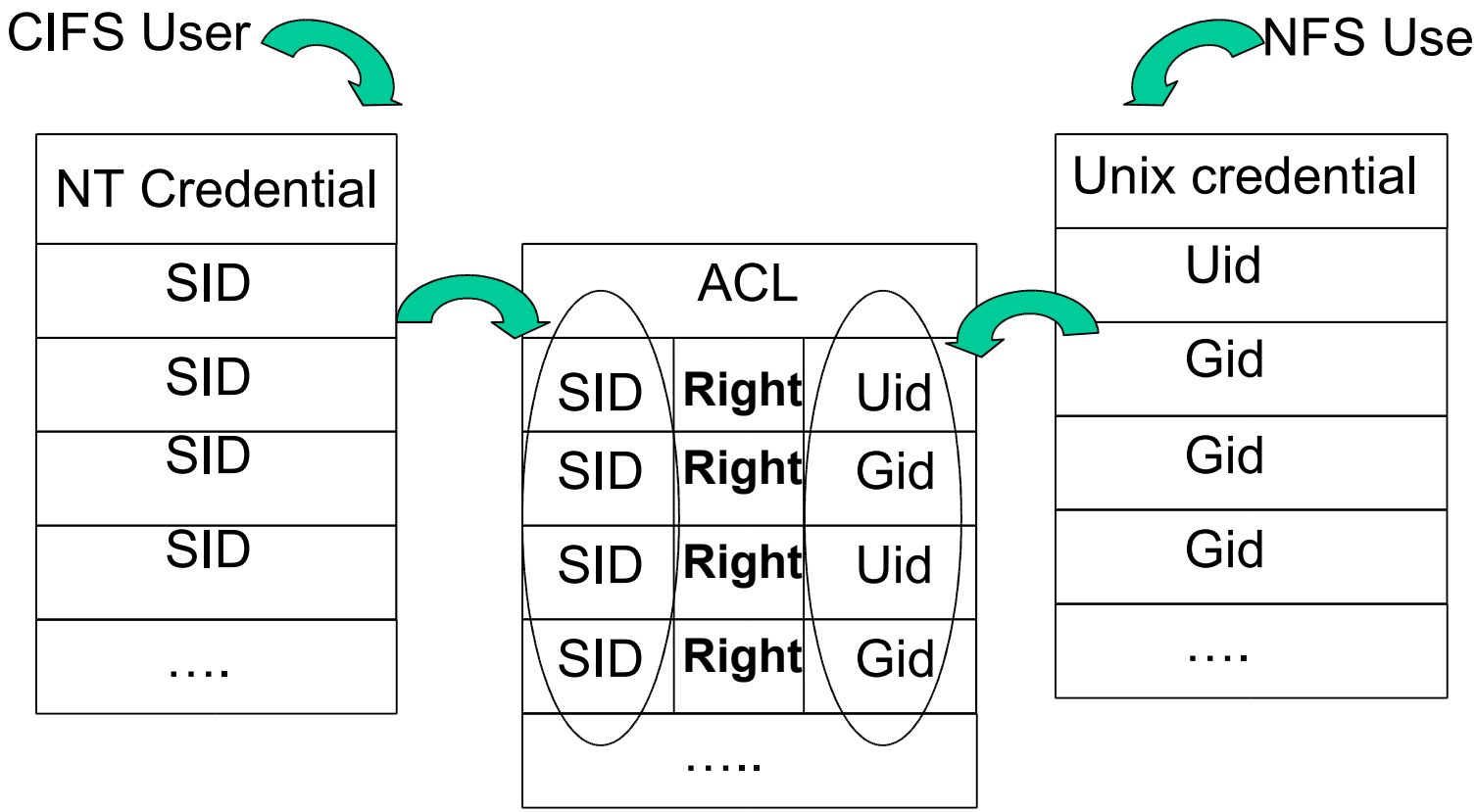
- Both sets of permissions are optional in NFSv4
- Mode-bits are always visible
- ACLs are visible in MIXED mode

	UNIX	NT	NATIVE	SECURE	MIXED
CIFS	ACL + MB	ACL	ACL	ACL + MB	ACL
NFSv3	MB	ACL + MB	MB	ACL + MB	ACL
NFSv4	MB	ACL + MB	MB	ACL + MB	ACL

v4 & CIFS - Permissions

- Access Control and Permissions Checking
  - In EMC model, access right checking uses the credential provided by the protocol
    - For CIFS, the credential is built from the information return by the authentication process (based on SID)
    - For NFS, the credential is passed in the NFS request itself (built on the client side at logon time, based on uid/gids)
- In NT/Secure modes, ACLs are checked using uid/gid mapping stored in the ACL or through reconstruction of the Windows credential
- Conversely, in UNIX/Secure modes, the Unix groups can be added to the Windows credential
- NFS credential (in auth_sys) is limited to 16 groups

v4 & CIFS - Permissions



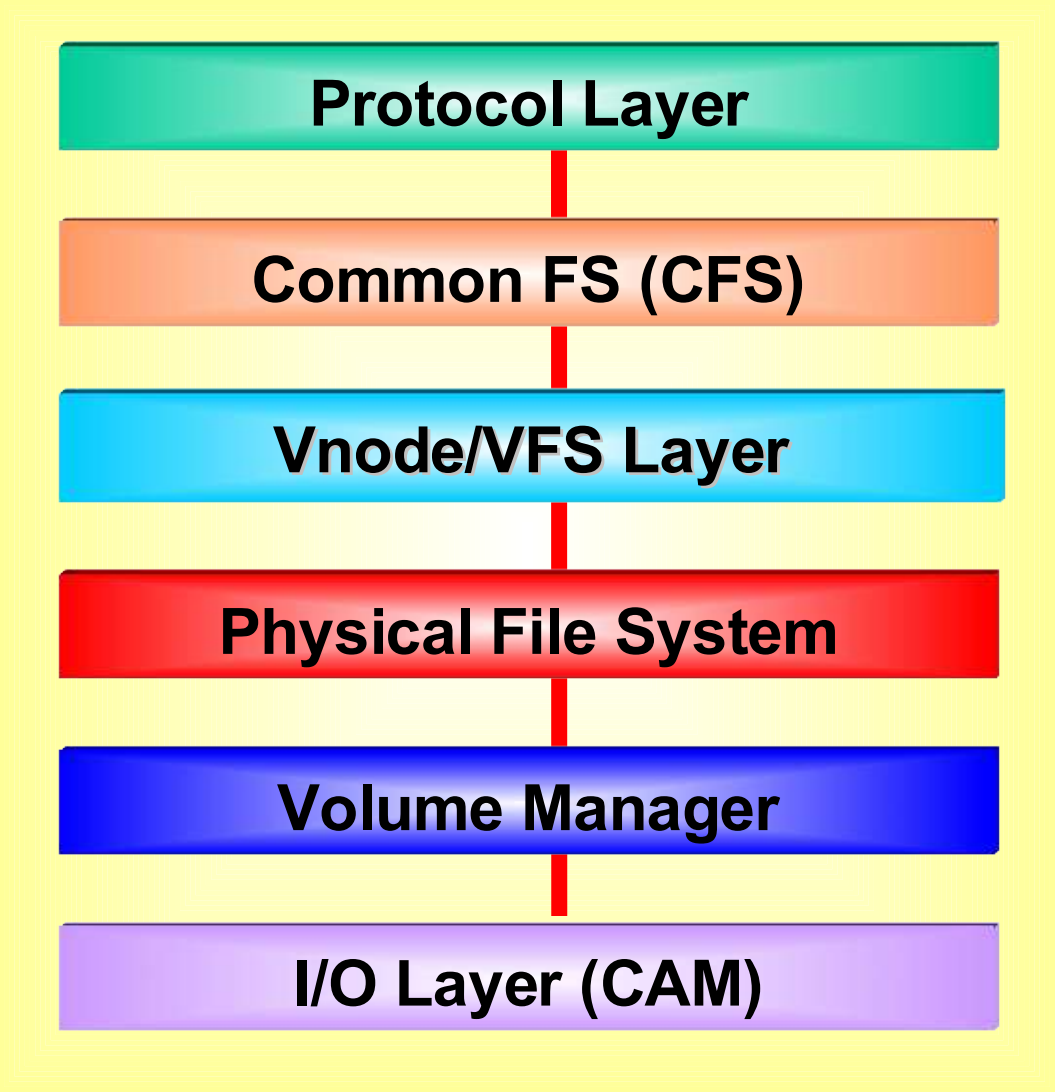
Can generate the NFS Credential based on Unix Groups or Windows Groups (to get around 16 group limitation)

v4 & CIFS – ACLs

- We sort the ACE in the ACL in the order expected by Windows Explorer
- The ACE of the ACL set by CIFS or NFSV4 are ordered in the following order:
  - The DENY inherited ACE
  - The ALLOW inherited ACE
  - The DENY non inherited ACE
  - The ALLOW non inherited ACE.
- Reordering the ACEs resolves Windows Explorer complaints and implicit reordering of ACEs.

v4 & CIFS - Locking

Both protocols now support mandatory locking and range locks



v4 & CIFS - Locking

CIFS Oplocks vs. NFS Delegations

- CIFS Oplock Types:
  - Level I (exclusive cached read/write)
  - Level II (shared read)
  - Batch lock (multiple accesses of open/close with Level I)
- Level I Oplocks attempt to negotiate to Level II Oplocks on a CFS read of the exclusively locked object.
  - A write operation results in an level II Oplock break.
- Requested by the client, in-band CIFS call

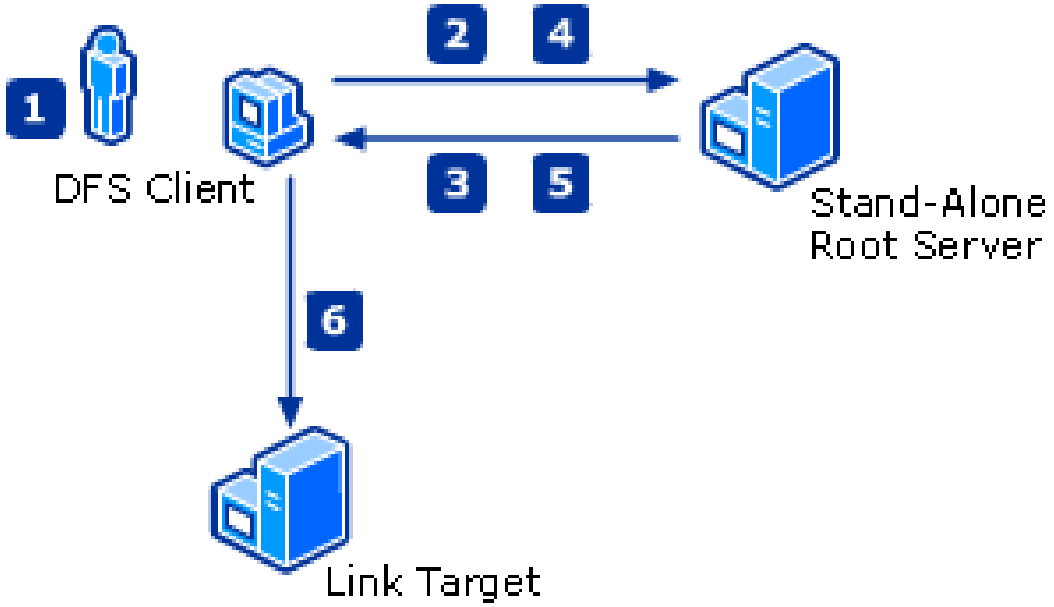
v4 & CIFS - Locking

CIFS Oplocks vs. NFS Delegations

- NFS Delegations
  - Server granted, at the discretion of the server
  - Supports the concepts of renewal and Delegation upgrade
  - Out of protocol revocation method (callback to the client)
- Like a batch oplock (Level II), can persist beyond the close of the file.
- Any access at the CFS layer for a delegated resource will result in a delegation recall.

v4 & CIFS - Namespace

- FS_LOCATION can be used to provide DFS like semantics
- DFS, supported on > Windows 2000, provides the ability to move resources:

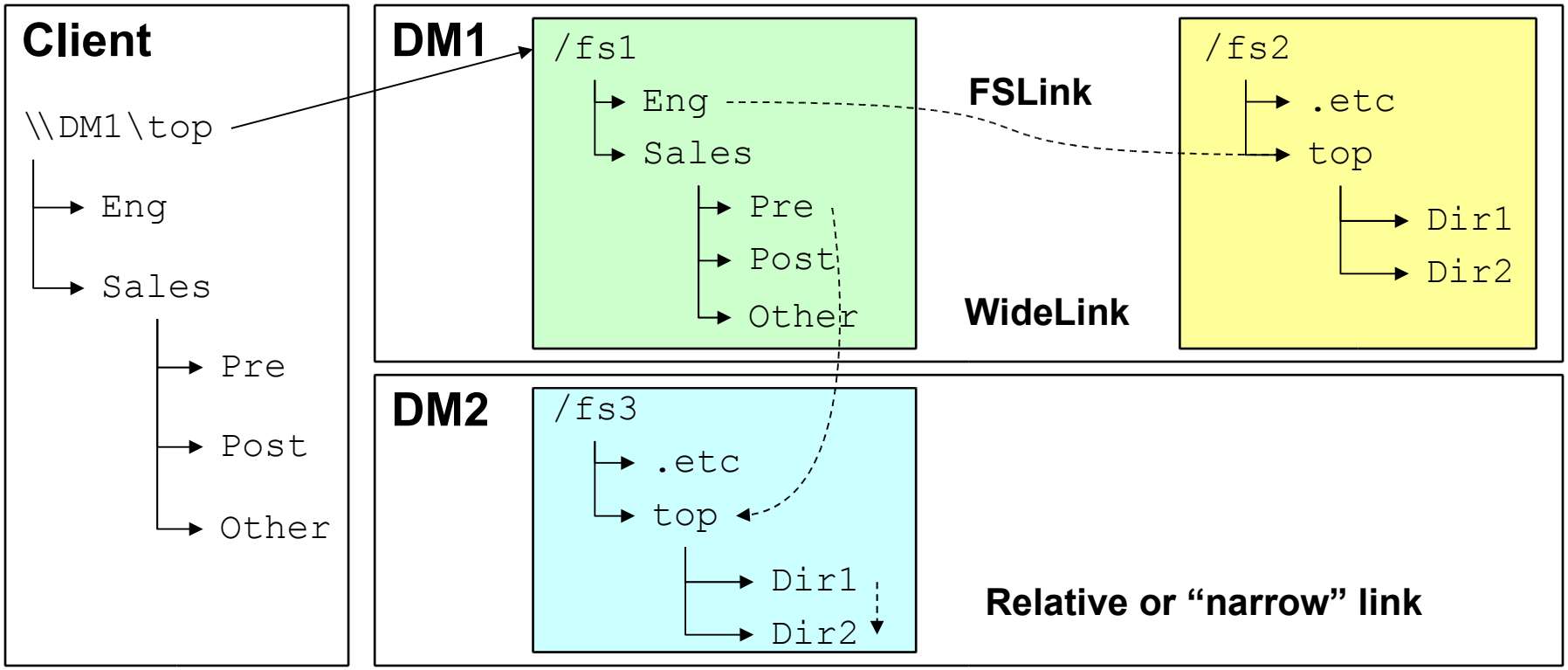


- Limited in its ability to move pinned resources.

v4 & CIFS - Namespace

- FS_LOCATION will allow the same ‘moved’ semantics, but without explicitly requiring a mapping party.
- The mappings could be stored in a LDAP database
- One large advantage is that the resource could be moved, with open handles, the server returns a MOVED error, which the client then does an FS_LOCATION and finds the moved object.

DFS and Links



- Manually setting the DFS referral link -

Challenges

- There are still items to be resolved
- ACL incompatibility
  - Each type of client interprets and sets ACLs in its own way
  - No compatibility among NFS v4 clients
  - Even less compatibility with Windows model
- Syncing the Mode Bits and the ACL is challenging
- Kerberos in delegated Environments
  - Linux requires that each client system has an account with the KDC (CITI plans to change that in the future)
- Maintaining backwards compatibility with NFSv3

Questions?

- Q & A