

NFSv3 and SMB/SMB2 Interoperability in Likewise Storage Services

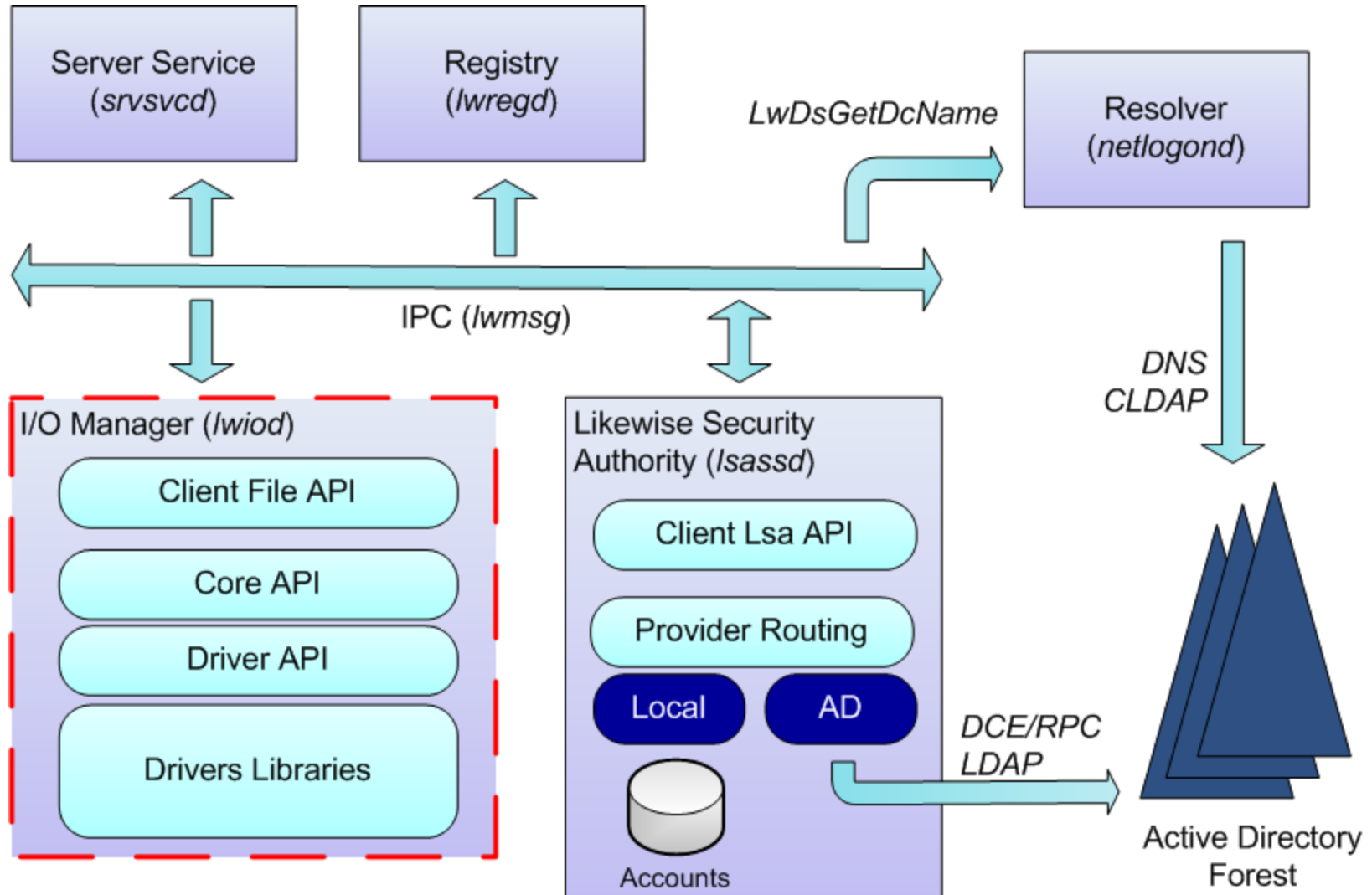
Evgeny Popovich
Likewise Software

- Likewise File Server Architecture
- NFS and CIFS Interoperability Challenges
- NFS Driver as Part of Likewise File Server
- Cross-Protocol Access Check and File Locking
- NFS in User Space Challenges and Solutions
- Plans for Future Development

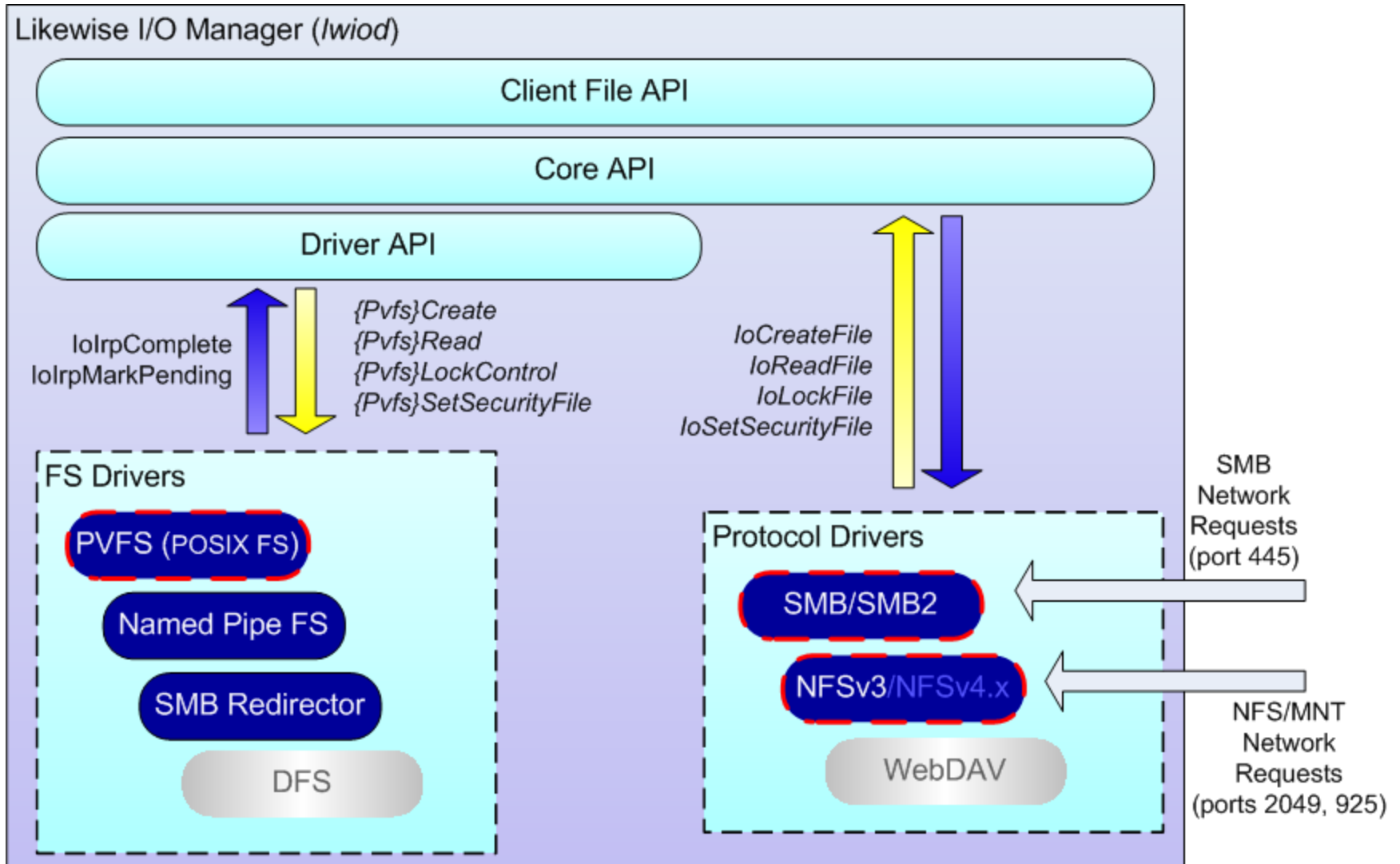
Likewise File Server Introduction

- ❑ Likewise File Server is designed to provide an interoperability storage platform for non-Microsoft clients and servers
- ❑ Includes support for SMB, SMB2, and SMB2.1 (available under GPL and commercial licenses)
- ❑ NFSv3 server is a recent addition to the stack
- ❑ User-space implementation

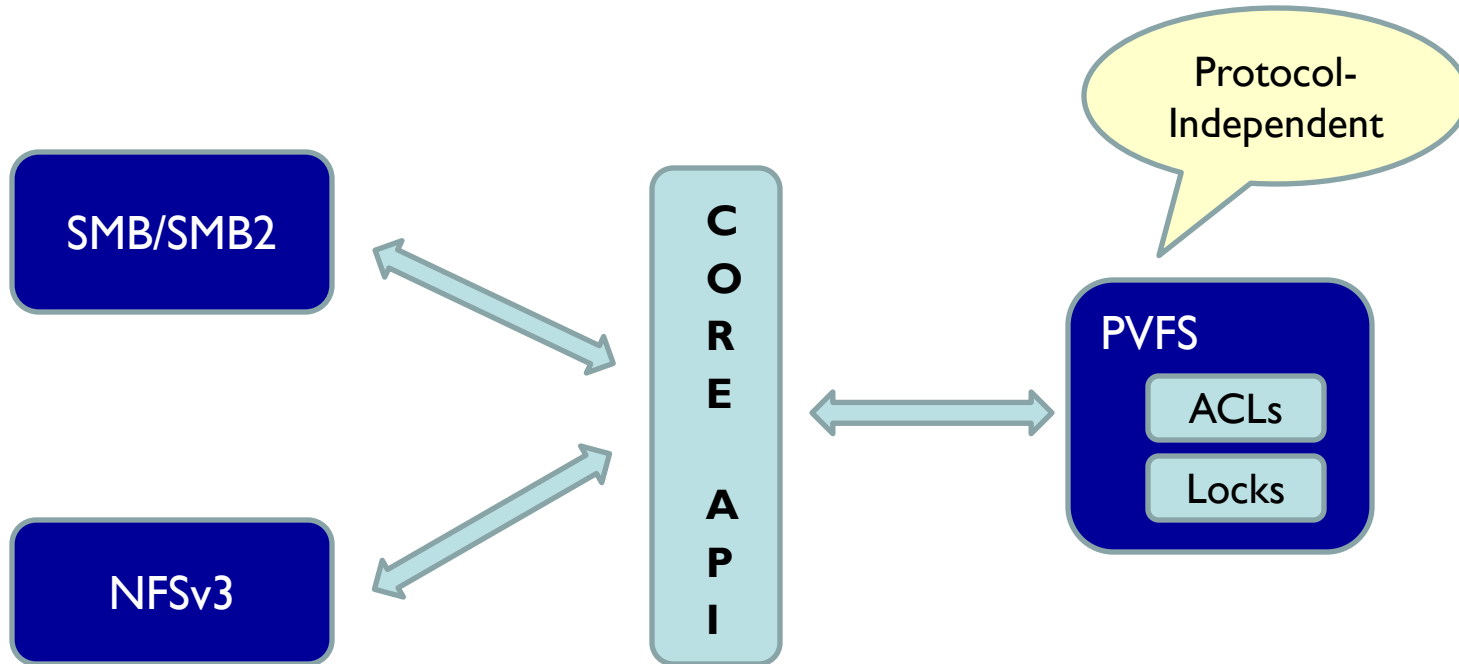
Likewise File Server Architecture



Likewise I/O Manager



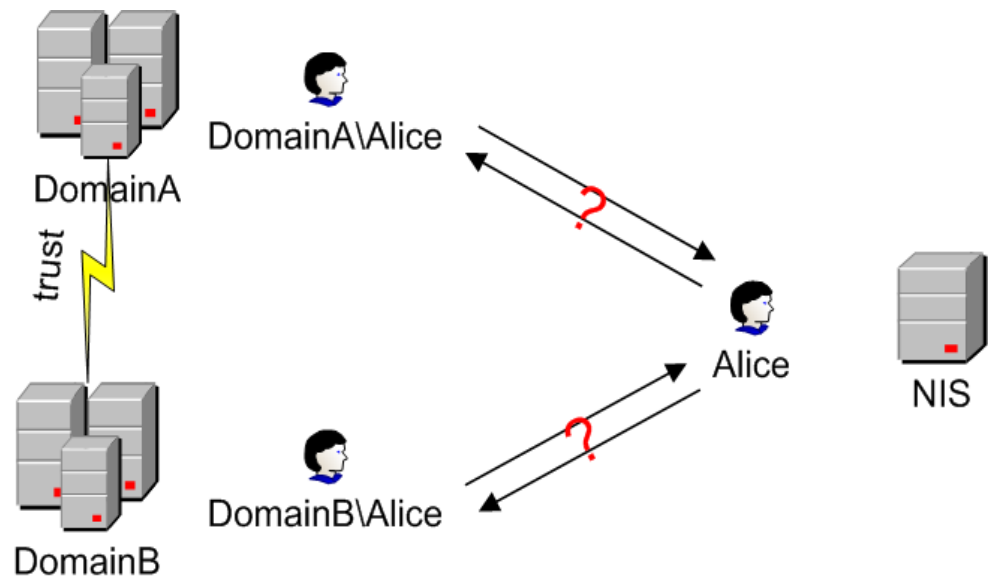
Likewise I/O Manager – IO Flow



- ❑ The File Server has to be able to identify a user regardless of the network file protocol used
- ❑ Consistent behavior while accessing files from different protocols
 - ❑ Provide equal access to the same user when accessing a file from different protocols
 - ❑ Locks acquired from one protocol have to equally affect clients accessing files from all protocols
 - ❑ File names have to be as close to the original names as possible (in cases of unsupported characters in one of the protocols, or for the names that differ only in case)

NFSv3 and SMB Interoperability Challenges

- ❑ Different user repositories
 - ❑ AD/NIS/LDAP Directories
 - ❑ Require users (and sometimes groups) mapping
- ❑ Automatic mapping by Name
 - ❑ Same name in trusted domains
 - ❑ Different names (john in NIS vs. jsmith in AD)



Users Mapping Problems

- ❑ Manual mapping – may not be feasible in large organizations
- ❑ Group membership
 - ❑ Groups are almost never the same in different repositories – is there a Domain Users group in every NIS repository?
 - ❑ Which repository group membership should be used?

NFSv3 and SMB Interoperability Challenges (cont.)

- ❑ Access Control models are very different
- ❑ Windows ACLs may have a large number of ACEs for various users and/or groups.
- ❑ Windows ACLs support inheritance
- ❑ UNIX mode bits is a simple model, no inheritance.
- ❑ Access check algorithms are very different:
 - ❑ Windows ACLs: scan all ACEs (in a specific order) until the access is granted or explicitly denied.
 - ❑ UNIX mode bits: check **ONLY** the matching three access bits (“owner”, “group” or “other”).

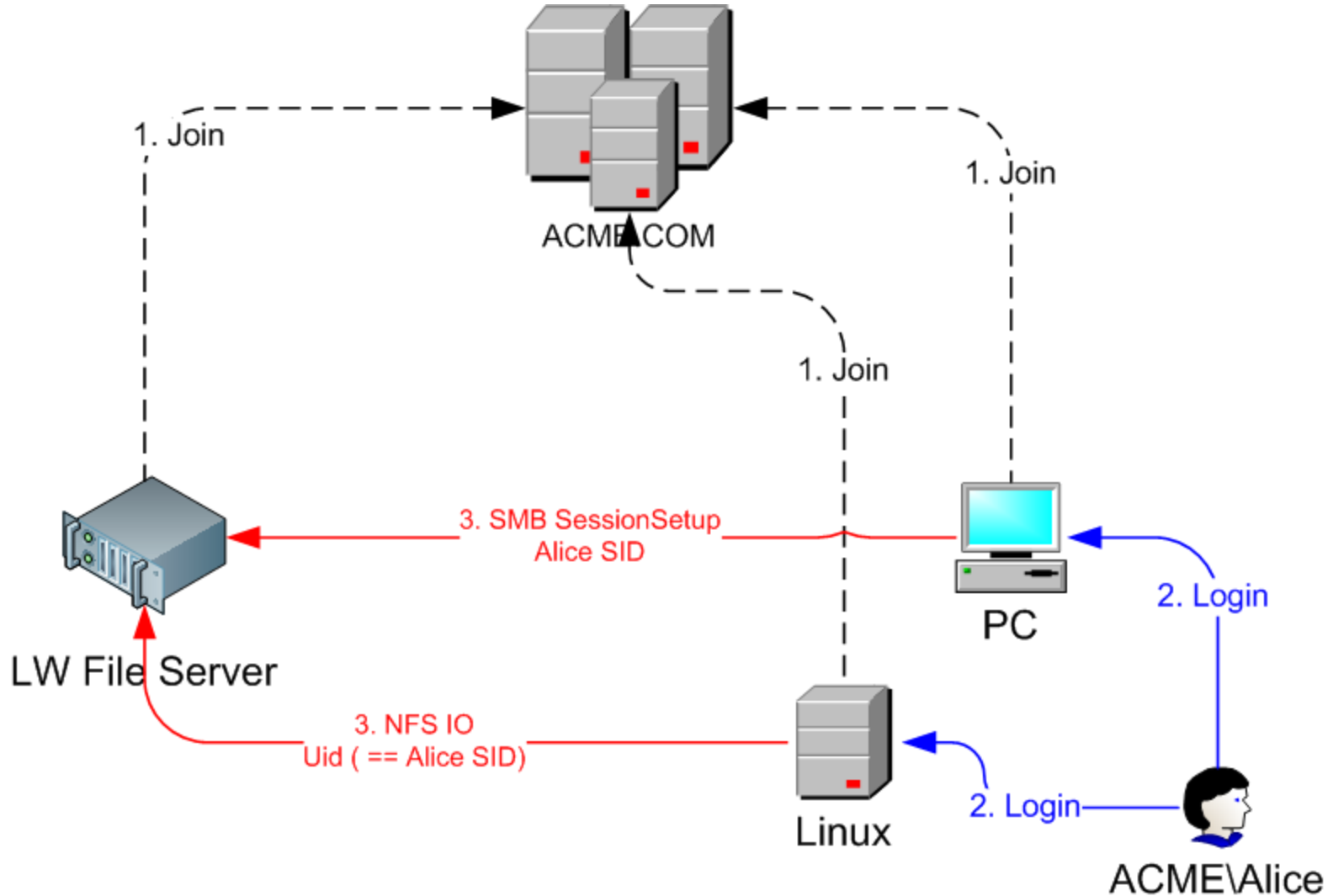
NFSv3 and SMB Interoperability Challenges (cont.)

- ❑ Files naming rules are different
 - ❑ SMB does not allow the following characters: \ / :
* ? ” < >
 - ❑ NFS servers usually do not allow only ‘/’
- ❑ SMB file names are case-insensitive, while NFS is case-sensitive

Likewise Approach

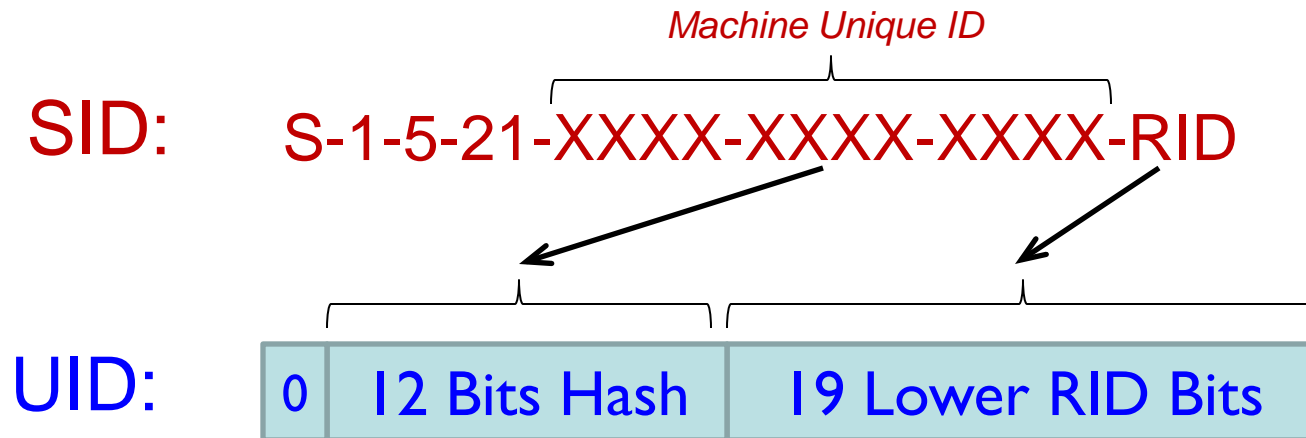
- ❑ Use a single repository by joining Likewise File Server AND *nix / MAC clients to AD Domain (using Likewise Open or Likewise Storage Services)
- ❑ Windows SIDs are translated to 32-bit ids (uids/gids) in a deterministic way. The same SID will be mapped to the same id on all Servers/Clients (explained further)
- ❑ Users logging in to *nix machines with AD credentials will be assigned these 32-bit ids (as uid/gids)
- ❑ The NFS Server (lwiiod) treats 32-bit ids as SID aliases, and works internally with SIDs instead of UIDs – similar to the way SMB driver works

Likewise Approach



SID to UID/GID Translation

1. Use AD that supports RFC2307 (uses UNIX attributes, like uid/gids, stored in AD)
2. Use Lsass SID hashing algorithm (pluggable mechanism)

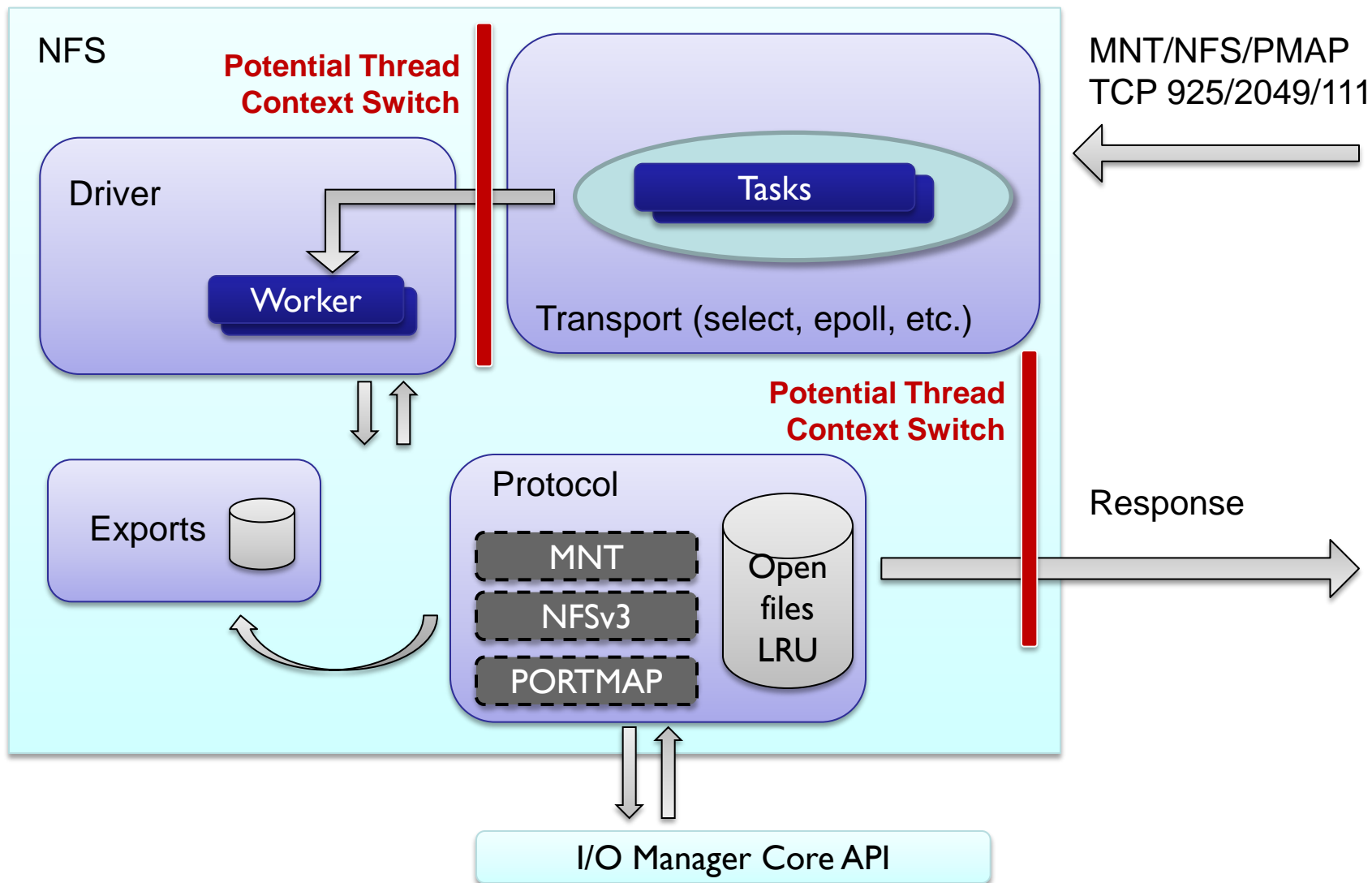


1. Windows will never reuse the RID for an account once it has been deleted

Lsass SID Hashing Limitations

- ❑ At most 2^{19} (~500,000) generated uids or gids are guaranteed to be unique in this space
 - ❑ Number of RID bits may be extended at the expense of domain SID hash collision rate (completely safe in single domain forests)
- ❑ It is possible that the Domain SID hash for the three sub-authorities will collide between trusted domains

NFS Driver Architecture



Cross-Protocol Access Checks

- ❑ Internally PVFS implements the semantics of mode bits using Windows DACLs.
- ❑ Setting mode bits from NFS (chmod) will usually result in a DACL with three ALLOW ACEs (Sometimes DENY ACEs will be added)
- ❑ When file attributes need to be returned to NFS clients, the DACL needs to be mapped to mode bits

Setting Mode Bits from NFS

- ❑ A completely new DACL will be created for a file – the old DACL (including inherited ACEs) will not be preserved
- ❑ In addition to Allow ACEs, Deny ACEs may be added:
 - ❑ 0457 (r-- r-x rwx) =>
 - ❑ Owner Allow FILE_GENERIC_READ
 - ❑ Owner Deny FILE_ALL_ACCESS
 - ❑ Group Allow FILE_GENERIC_READ | FILE_EXECUTE
 - ❑ Group Deny FILE_ALL_ACCESS
 - ❑ Everyone Allow FILE_ALL_ACCESS

- ❑ Happens only when attributes are requested – does not affect access checks
- ❑ More restrictive vs. more permissive mode bits
- ❑ Some clients may rely on file attributes to make access control decisions – the “restrictive” approach will make such clients to deny access, when they actually have access

- ❑ The resulting mode bits may be more permissive than the original DACL:
 - ❑ For Allow ACE, if **any** of the FILE_GENERIC_READ bits is set, we will set the “read” mode bit
 - ❑ For Deny ACE, if **all** of the FILE_GENERIC_READ bits are set, we will reset the “read” mode bit

DACL to UNIX Permissions Mapping

	Modify "user" mode bits	Modify "group" mode bits	Modify "other" mode bits
ACE Type			
USER == FO, Allow	√		
USER == FO, Deny	√		
USER != FO, Allow, USER belongs to FG		√	
USER != FO, Allow, USER does not belong to FG			√
USER != FO, Deny, USER belongs to FG			
USER != FO, Deny, USER does not belong to FG			
GROUP == FG, Allow, FO belongs to FG	√	√	
GROUP == FG, Allow, FO does not belong to FG		√	
GROUP == FG, Deny, FO belongs to FG	√	√	
GROUP == FG, Deny, FO does not belong to FG		√	
GROUP != FG, Allow, FO belongs to GROUP	√	√	√
GROUP != FG, Allow, FO does not belong to GROUP		√	√
GROUP != FG, Deny, FO belongs to GROUP	√		
GROUP != FG, Deny, FO does not belong to GROUP			
Everyone, Allow or Deny	√	√	√

Cross-Protocol File Locking

- ❑ The fact that both SMB and NFS drivers access PVFS driver using the same API allows us to support cross-protocol locking without significant effort
- ❑ BRLs acquired from CIFS will affect NFS clients
- ❑ NFS clients will trigger Oplock breaks for oplocks acquired by CIFS clients
- ❑ Share modes will affect NFS clients

Files Names Issues

- ❑ NFS filenames containing Windows invalid characters:
 - ❑ Return the names as is to Windows clients. The files are visible, but not accessible.
- ❑ NFS-created files which differ only in case
 - ❑ Display and redirect all requests only to the “first” file (decided by the creation time).

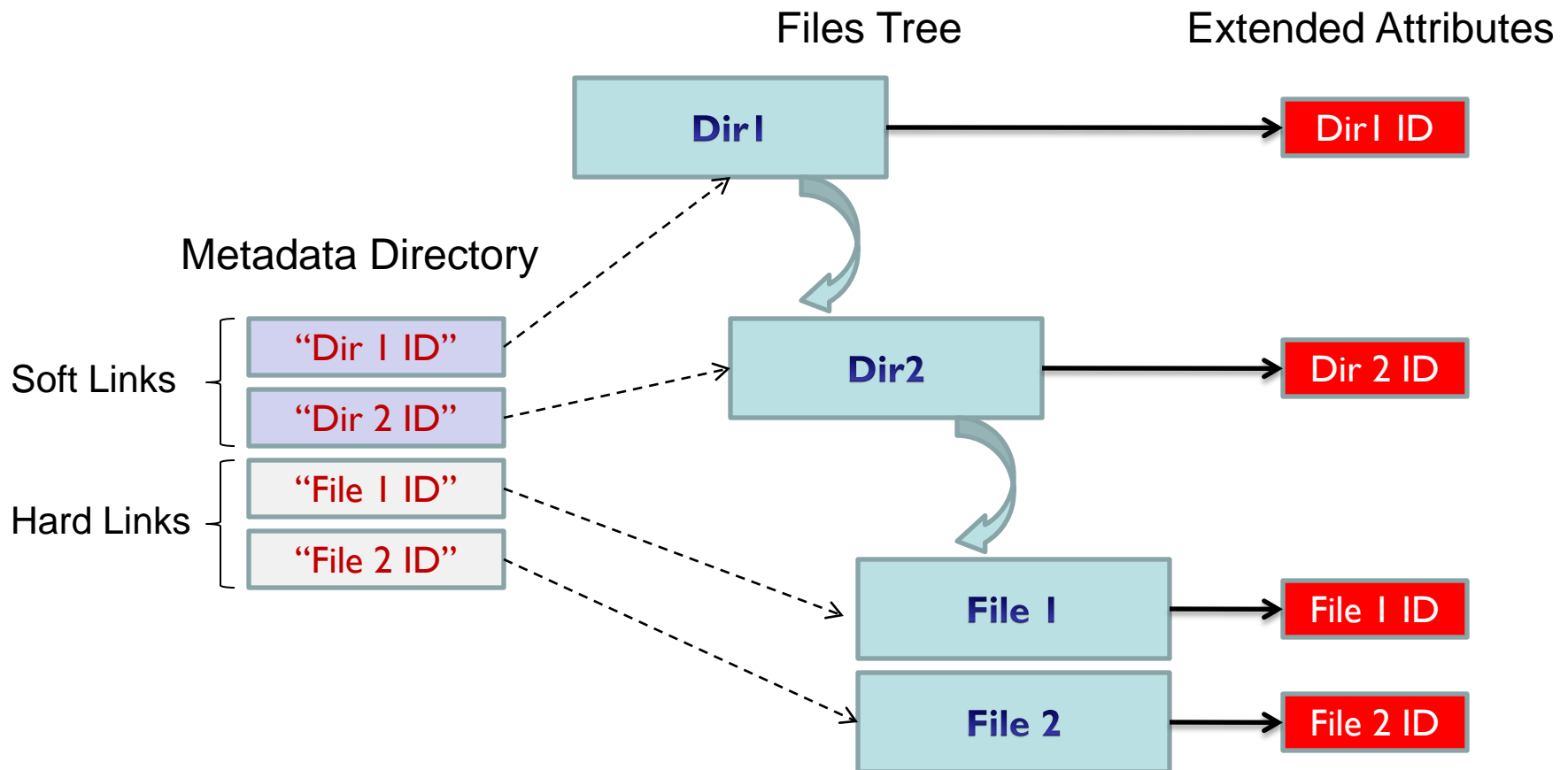


- ❑ 16 groups is a limitation of ONC RPC AUTH_SYS authentication flavor (RFC 1831)
- ❑ NFSv3 servers (using AUTH_SYS) can receive maximum 16 groups in RPC header, even if the user belongs to more groups (as defined in a repository)
- ❑ LW File Server does not use gids array from the RPC header – instead it fetches the groups membership from AD, overcoming the limitation.

- ❑ NFS filehandles have to be unique, persistent across reboots and path-independent
 - ❑ A file path cannot be encoded in a filehandle
- ❑ POSIX allows opening a file only by filenames, there is no way to open a file by file ID.
- ❑ A POSIX-FS-based mapping from ids to files allows opening files by IDs.

NFS in User Space – Open-by-ID

- Implemented as part of PVFS driver



- ❑ For a directory a symlink stores not a full directory path, but a parent directory id and the directory name
- ❑ The metadata directory can be split to multiple directories, to avoid a single huge metadata directory

NFS in User Space (cont.)

- ❑ Weak cache consistency
 - ❑ Most of the NFS operations support Pre- and Post- metadata attributes
 - ❑ Getting Pre-md, executing an operation, and getting Post-md must be an atomic operation
 - ❑ Requires file locking in user-space FS driver on almost every RPC call
- ❑ It is easy to do export access checks on every RPC call (hostnames/netgroups resolution)

Plans for Future Development

- Kerberized NFSv3
 - The Kerberos environment is easy to setup with Likewise solutions (on non-Windows clients and servers)
 - Adding RPCSEC_GSS with Kerberos makes NFSv3 a secure protocol
- NFSv4.x

Questions?

Evgeny Popovich
SDE
Likewise Software

epopovich@likewise.com
<http://www.likewise.com>