

Likewise-CIFS

*Technical Deep Dive into the
Likewise SMB Server*

Likewise Open – Background

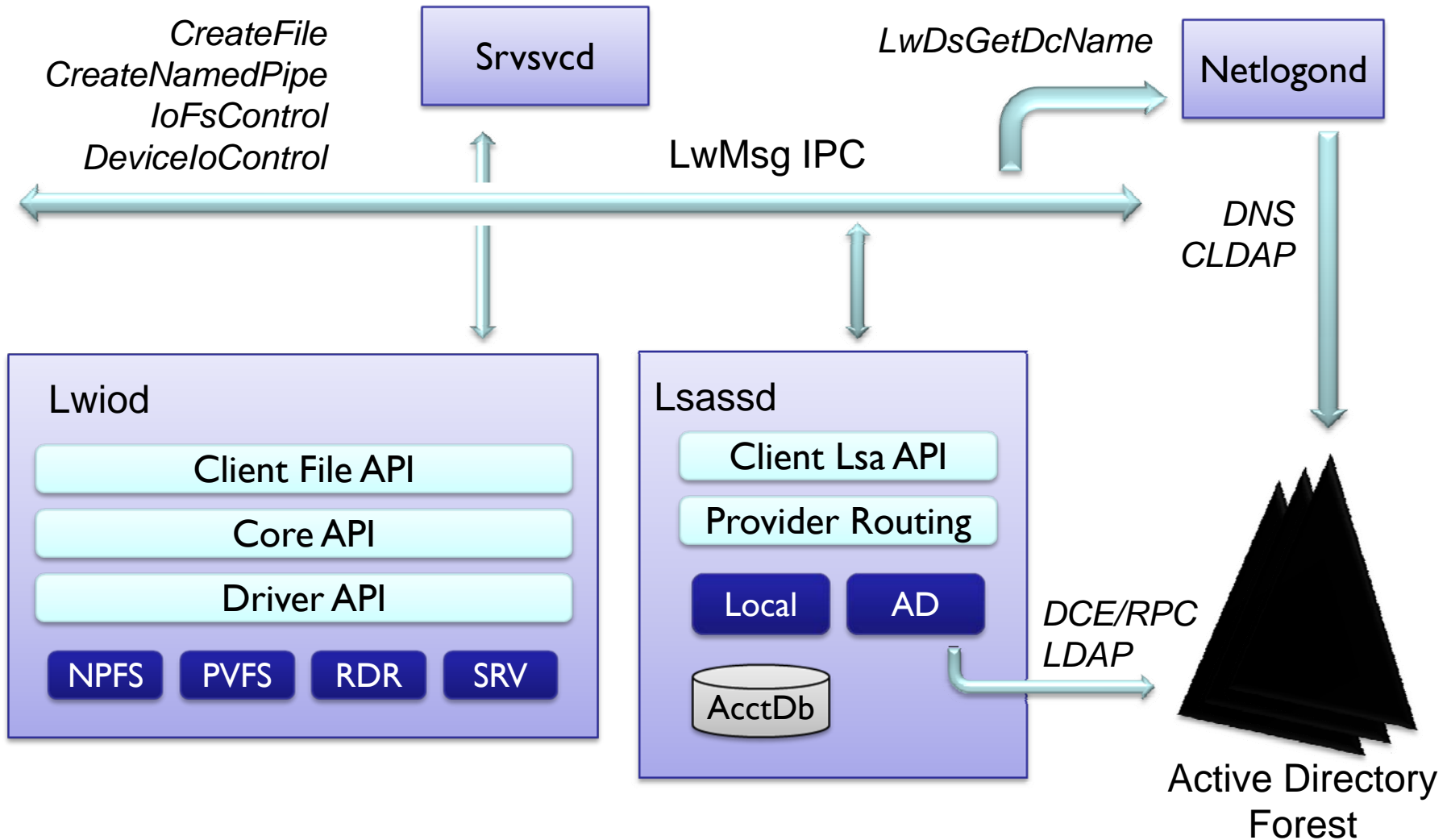
<http://www.likewiseopen.org/>

- Goal – Likewise Open is the umbrella project sponsored by Likewise Software designed to provide an interoperability platform for non-Microsoft clients existing in Microsoft OS dominated networks.
 - Project officially launched Nov. '07
 - L-CIFS development began in Jan. '09
- License – Combination of GPLv2+ and LGPLv2.1+
 - Non-Likewise components (e.g. OpenLDAP and MIT Kerberos) remain under their original license.

Likewise Open Components

- ❑ All are single process, threaded services
- ❑ *lwiod* – Likewise I/O Manager
- ❑ *lsassd* – Likewise Security Authority
- ❑ *srvsvcd* – Server and Workstation RPC Services
- ❑ *netlogond* – Domain Control locator
- ❑ *dcerpcd* – DCE/RPC endpoint-mapper
- ❑ *eventlogd* – Local/Remote logging service

Architectural Overview



Likewise I/O Manager

- ❑ Provides an API inspired by the Windows ZwCreateFile(), et. al. interface
- ❑ Makes use of I/O request packets (IRPs) to communicate with drivers
- ❑ Drivers are loaded at run time by the I/O Mgr core
 - ❑ rdr.sys.so – SMB client file system
 - ❑ npfs.sys.so – Named pipe file system
 - ❑ pvfs.sys.so – POSIX compatible file system
 - ❑ srv.sys.so – SMBv1 & v2 server protocol head

- ❑ IoCreateFile, IoCloseFile
- ❑ IoReadFile, IoWriteFile
- ❑ IoDeviceIoControlFile, IoFsControlFile
- ❑ IoQueryXXXInformation, IoSetXXXInformation
 - ❑ File, Directory, Volume
- ❑ IoLockFile, IoUnlockFile
- ❑ IoQuerySecurityFile, IoSetSecurityFile

I/O Mgr Client API

```
// Client IPC calls defined in <lwio/ntfileapi.h>  
// Internal iomgr calls defined in <ioapi.h>
```

LW_NTSTATUS

LwNtCreateFile(
 LW_OUT PIO_FILE_HANDLE FileHandle,
 LW_IN LW_OUT LW_OPTIONAL PIO_ASYNC_CONTROL_BLOCK AsyncControlBlock,
 LW_OUT PIO_STATUS_BLOCK IoStatusBlock,
 LW_IN PIO_FILE_NAME FileName,
 LW_IN LW_OPTIONAL LW_PVOID SecurityDescriptor,
 LW_IN LW_OPTIONAL LW_PVOID SecurityQualityOfService,
 LW_IN ACCESS_MASK DesiredAccess,
 LW_IN LW_OPTIONAL LONG64 AllocationSize,
 LW_IN FILE_ATTRIBUTES FileAttributes,
 LW_IN FILE_SHARE_FLAGS ShareAccess,
 LW_IN FILE_CREATE_DISPOSITION CreateDisposition,
 LW_IN FILE_CREATE_OPTIONS CreateOptions,
 LW_IN LW_OPTIONAL LW_PVOID EaBuffer,
 LW_IN LW_ULONG EaLength,
 LW_IN LW_OPTIONAL PIO_ECP_LIST EcpList
);

IRP_TYPE_CREATE

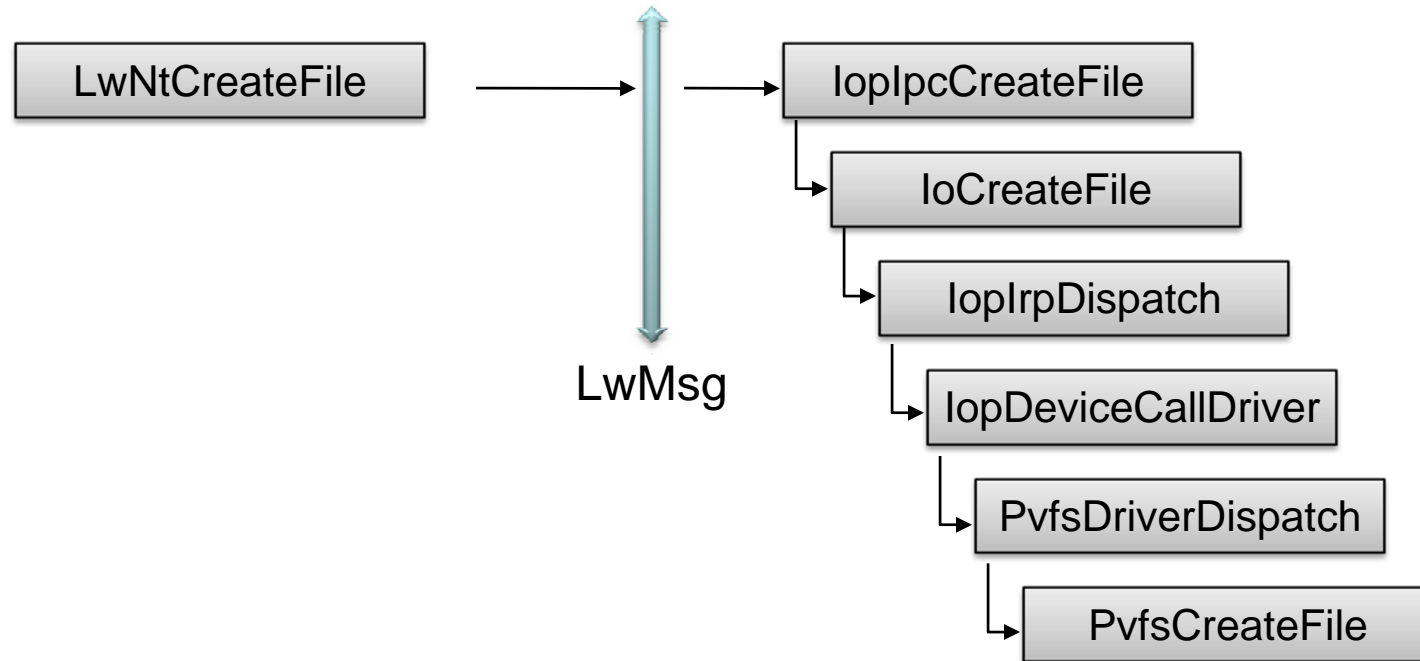
```
typedef struct _IRP {
    IN IRP_TYPE Type;
    OUT IO_STATUS_BLOCK IoStatusBlock;
    IN IO_DRIVER_HANDLE DriverHandle;
    IN IO_DEVICE_HANDLE DeviceHandle;
    IN IO_FILE_HANDLE FileHandle;
    IN union {
        IRP_ARGS_CREATE Create;
        . . .
    } Args;
} IRP, *PIRP;

typedef struct _IRP_ARGS_CREATE {
    IN PIO_CREATE_SECURITY_CONTEXT SecurityContext;
    IN IO_FILE_NAME FileName;
    IN ACCESS_MASK DesiredAccess;
    IN OPTIONAL LONG64 AllocationSize;
    IN FILE_ATTRIBUTES FileAttributes;
    IN FILE_SHARE_FLAGS ShareAccess;
    IN FILE_CREATE_DISPOSITION CreateDisposition;
    IN FILE_CREATE_OPTIONS CreateOptions;
    . . .
} IRP_ARGS_CREATE, *PIRP_ARGS_CREATE;
```

- ❑ All drivers register a supported namespace
 - ❑ For example, “\pvfs” and “\npfs”
- ❑ The LwNtCreateFile() Client API call must include the driver namespace prefix in the filename.
 - ❑ Prefix is stripped by the I/O Mgr before sending the IRP to the correct driver
- ❑ A Win32 compatibility layer can be provided to insulate end-user applications
 - ❑ E.g. CreateFile(“\\server\share\file.txt”)

LwNtCreateFile Example

```
$> test_pvfs --cat /pvfs/etc/hosts
## /etc/hosts
127.0.0.1 localhost
127.0.1.1 sequoia.ad.plainjoe.org sequoia
```

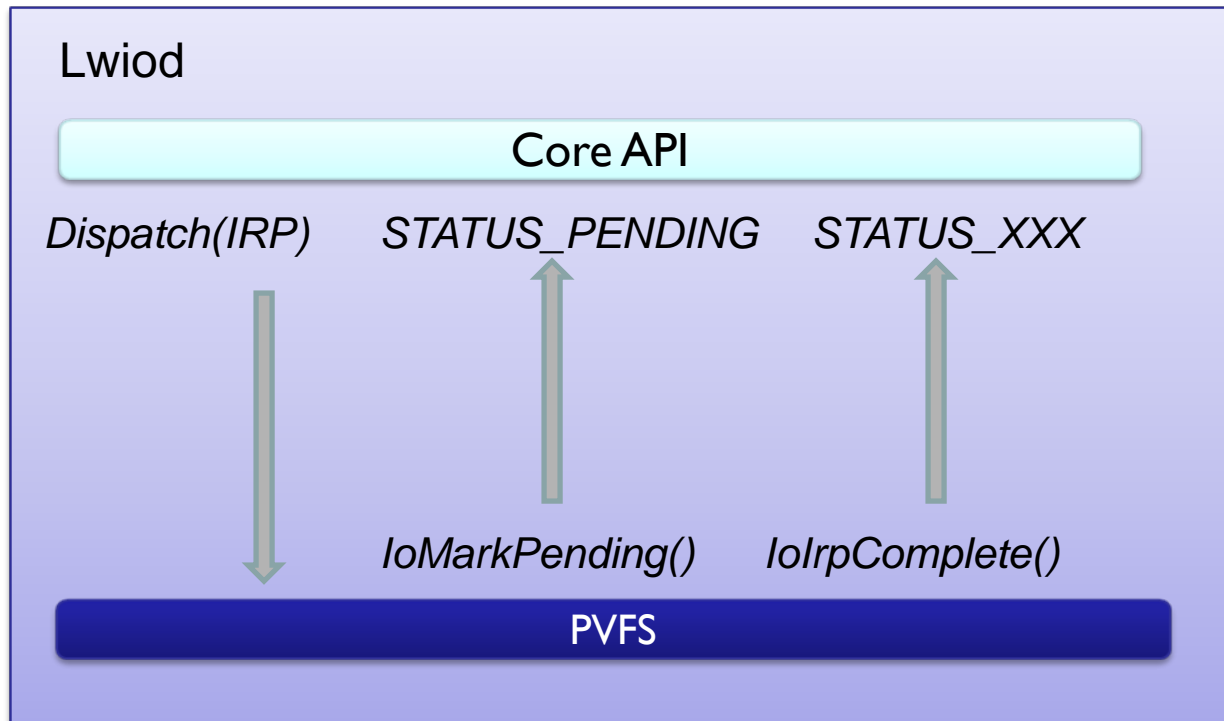


- ❑ Only the internal API support async calls currently
- ❑ loXX() calls accept an async control block

```
typedef struct _IO_ASYNC_CONTROL_BLOCK {  
    IN PIO_ASYNC_COMPLETE_CALLBACK Callback;  
    IN PVOID CallbackContext;  
    OUT PIO_ASYNC_CANCEL_CONTEXT AsyncCancelContext;  
} IO_ASYNC_CONTROL_BLOCK, *PIO_ASYNC_CONTROL_BLOCK;
```

- ❑ Driver can return PENDING to any request
 - ❑ The I/O Mgr simply blocks the caller thread on synchronous requests
- ❑ ACB->AsyncCancelContext
 - ❑ Can be used to cancel the pending request

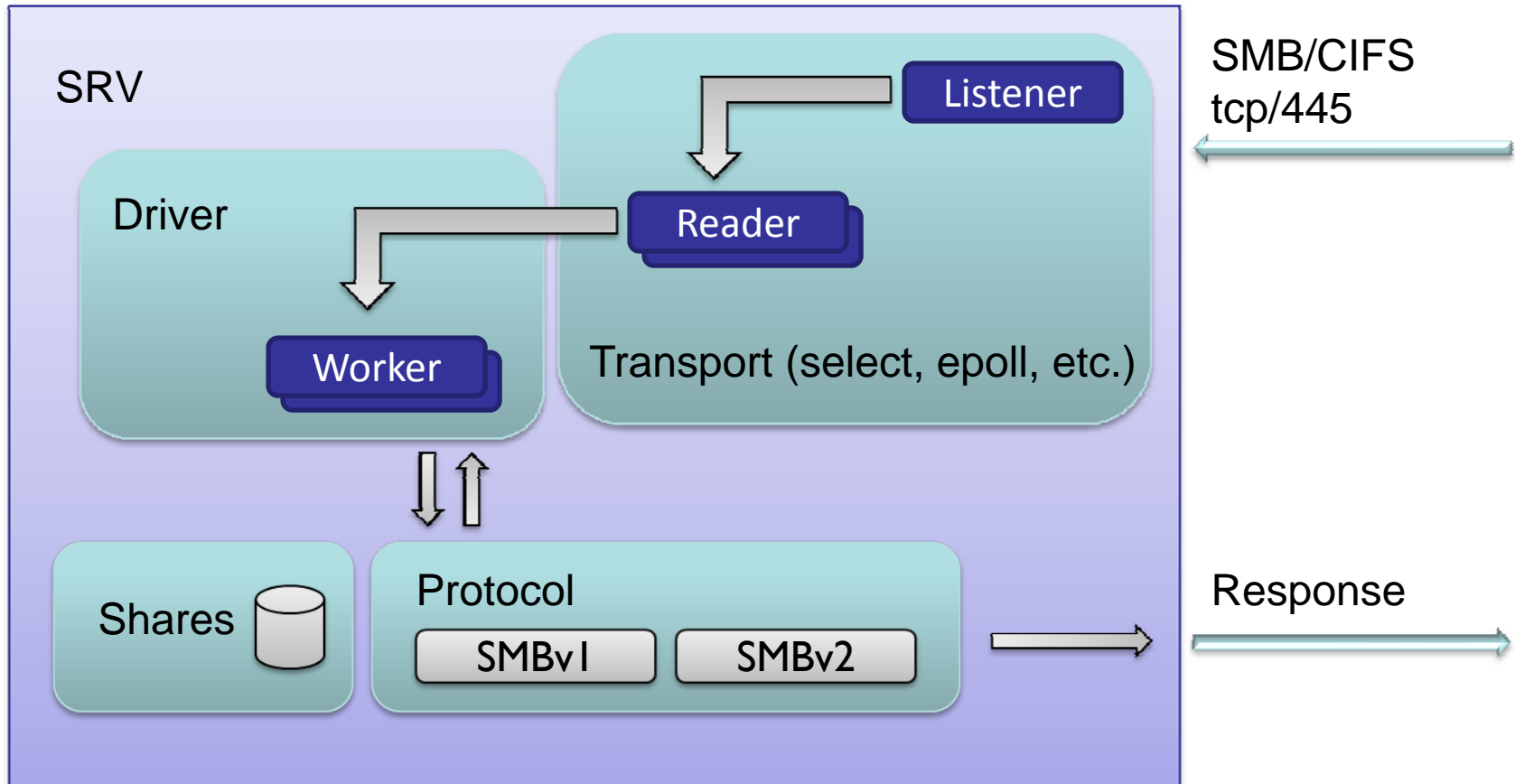
I/O Manager Async Calls (cont)



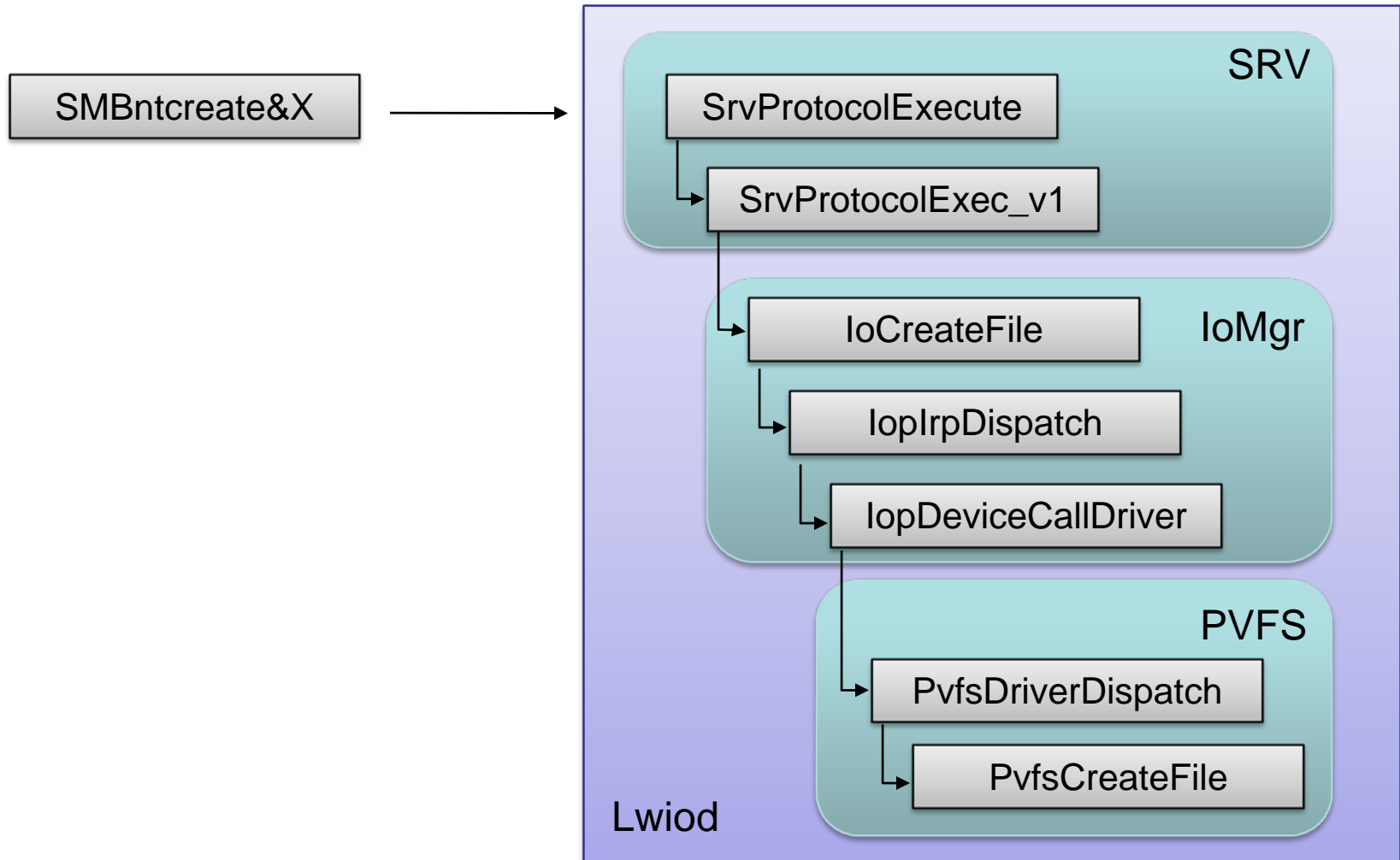
SRV & NPFS Drivers

- ❑ Support for SMBv1 and SMBv2
 - ❑ No NetBIOS support (only tcp/445)
 - ❑ NTLM 0.12 dialect or later
- ❑ Supported Clients
 - ❑ Windows XP/2003 and later
 - ❑ OS X and Linux clients forthcoming
- ❑ User mode security
 - ❑ Domain member and local authentication

SRV.sys.so - Architecture

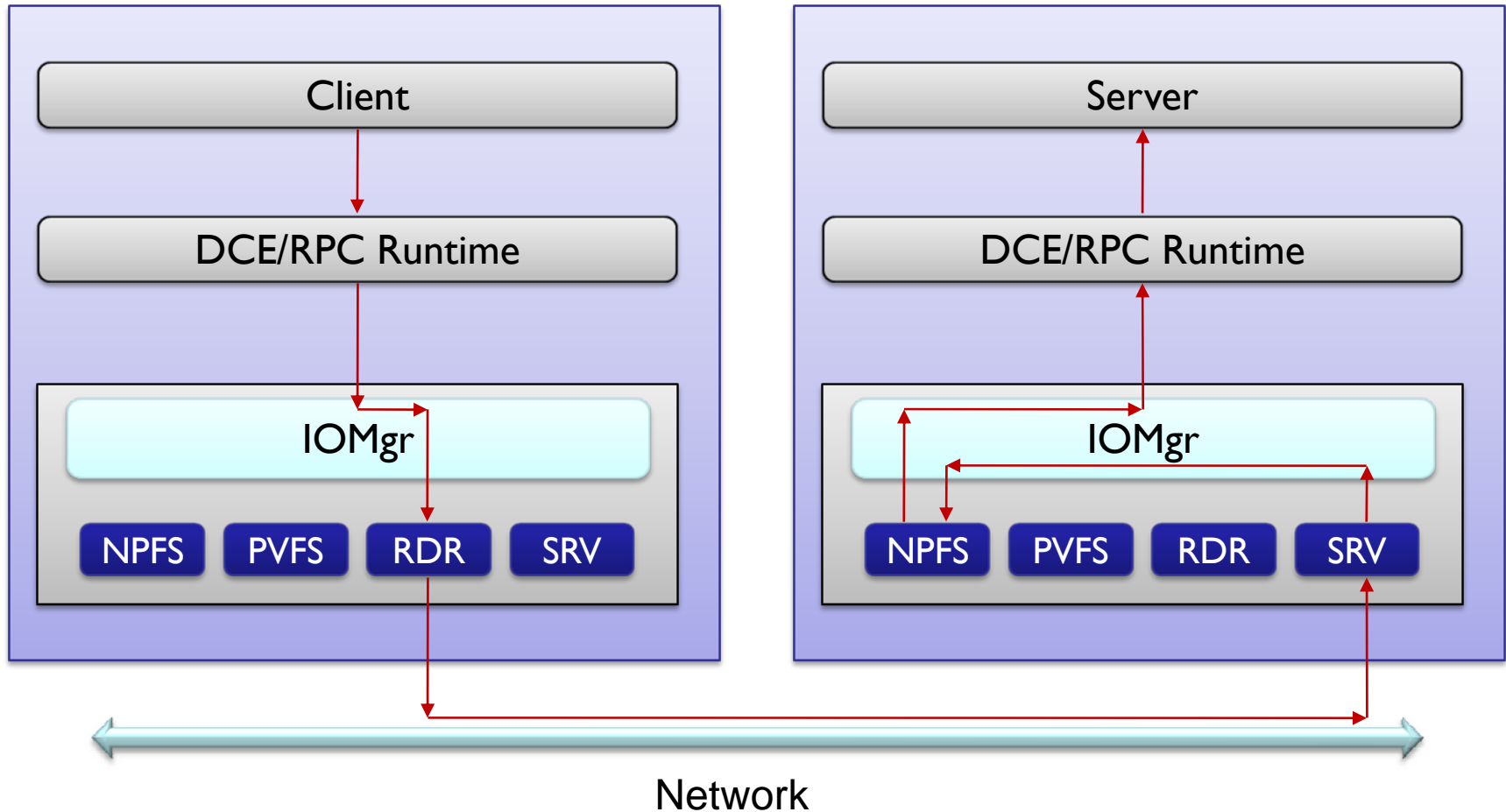


SMBntcreate&X Example



- ❑ NPFS driver implements an in-memory named pipe file system
- ❑ DCE/RPC runtime supports clients and servers using the NPFS driver in `lwiod`
 - ❑ Registers an `ncacn_np` endpoint for server applications using `LwNtCreateNamedPipeFile()`
 - ❑ The client runtime calls `LwNtCreateFile()` to open a pipe on a remote host

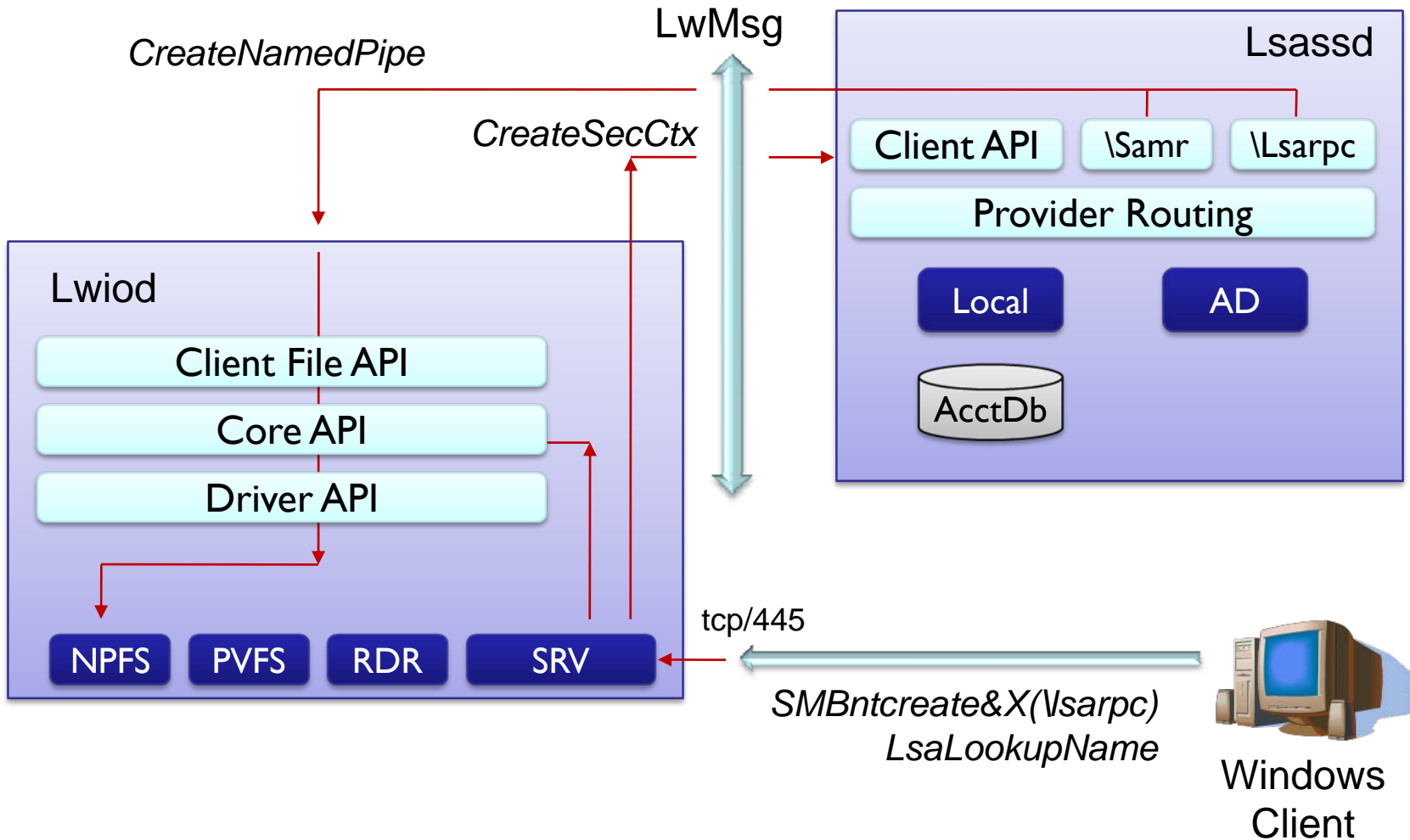
DCE/RPC Clients & Servers



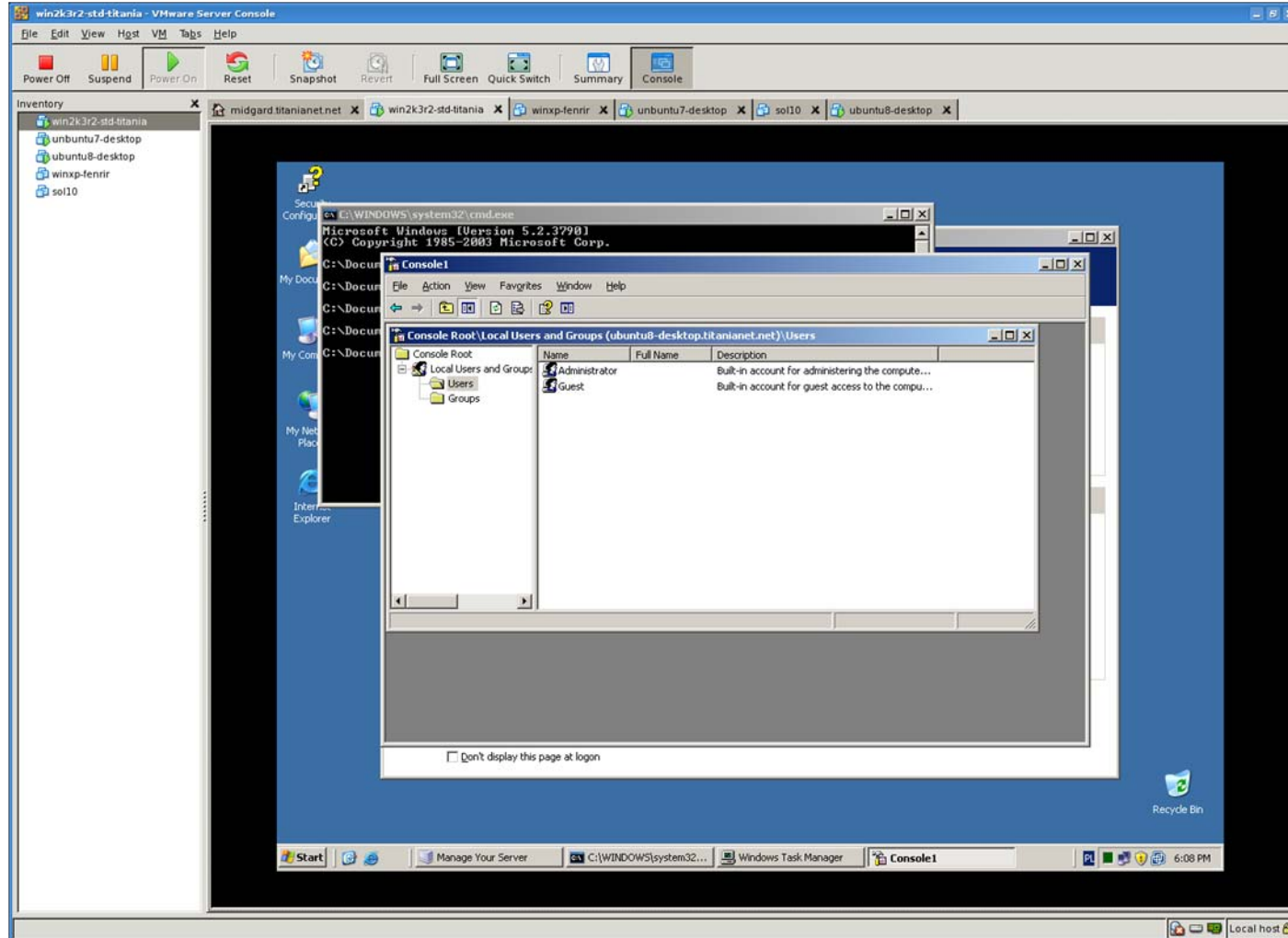
Likewise Security Authority

- ❑ User & Group Provider Routing
 - ❑ Local – Standalone account database
 - ❑ Privileged user management
 - ❑ Group nesting
 - ❑ MACHINE and BUILTIN domains
 - ❑ Active Directory – Member server functionality
 - ❑ Trust scenarios, Authentication, etc...
- ❑ Supplies session security contexts for Lwiod

Users Tokens and RPC Servers



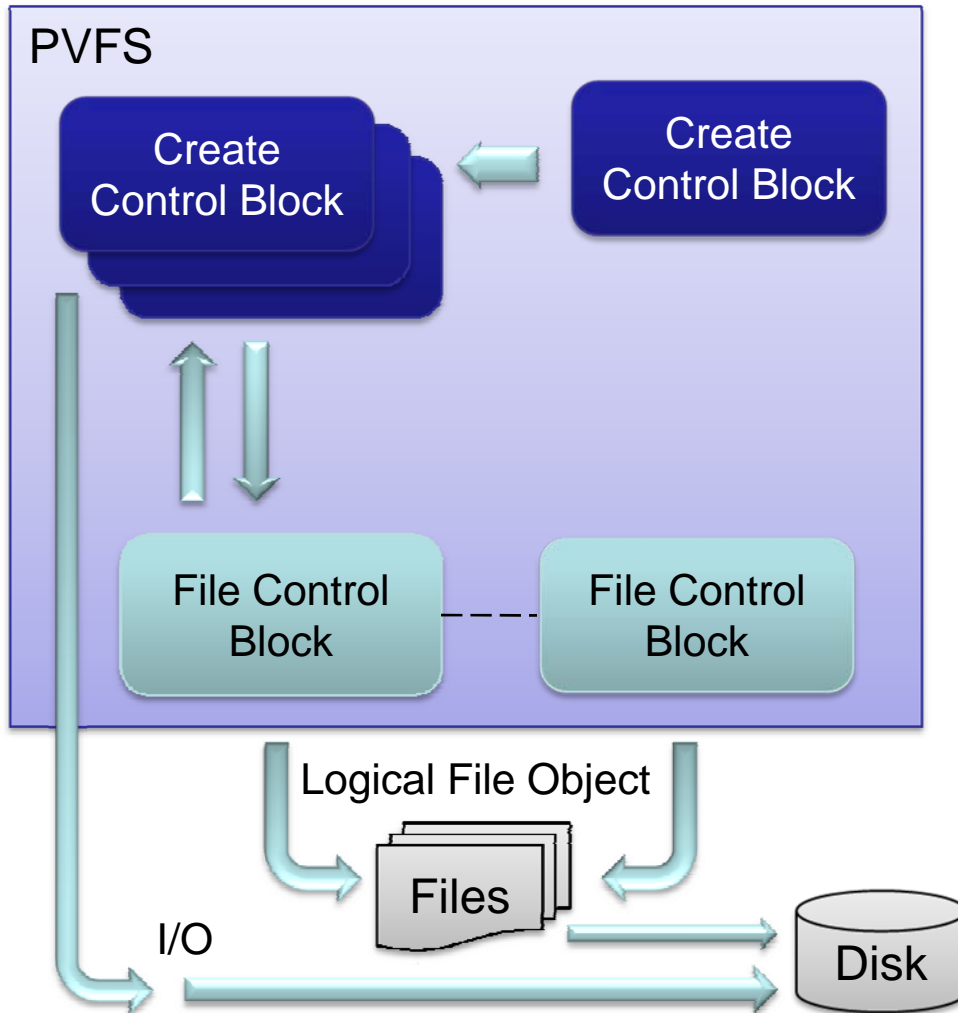
Local Users & Groups Demo



PVFS Driver

- ❑ Integration with POSIX file systems
 - ❑ Uses EAs for storing security descriptors, Attributes, etc..
 - ❑ Implements security and locking checks in process
- ❑ Provides a worker thread pool

PVFS – Data Structures



- ❑ FCB – File Object
 - ❑ Olocks
- ❑ CCB – Open Handle
 - ❑ Pathname
 - ❑ Dev/Inode
 - ❑ BRL
 - ❑ Sharemode
 - ❑ File Descriptor

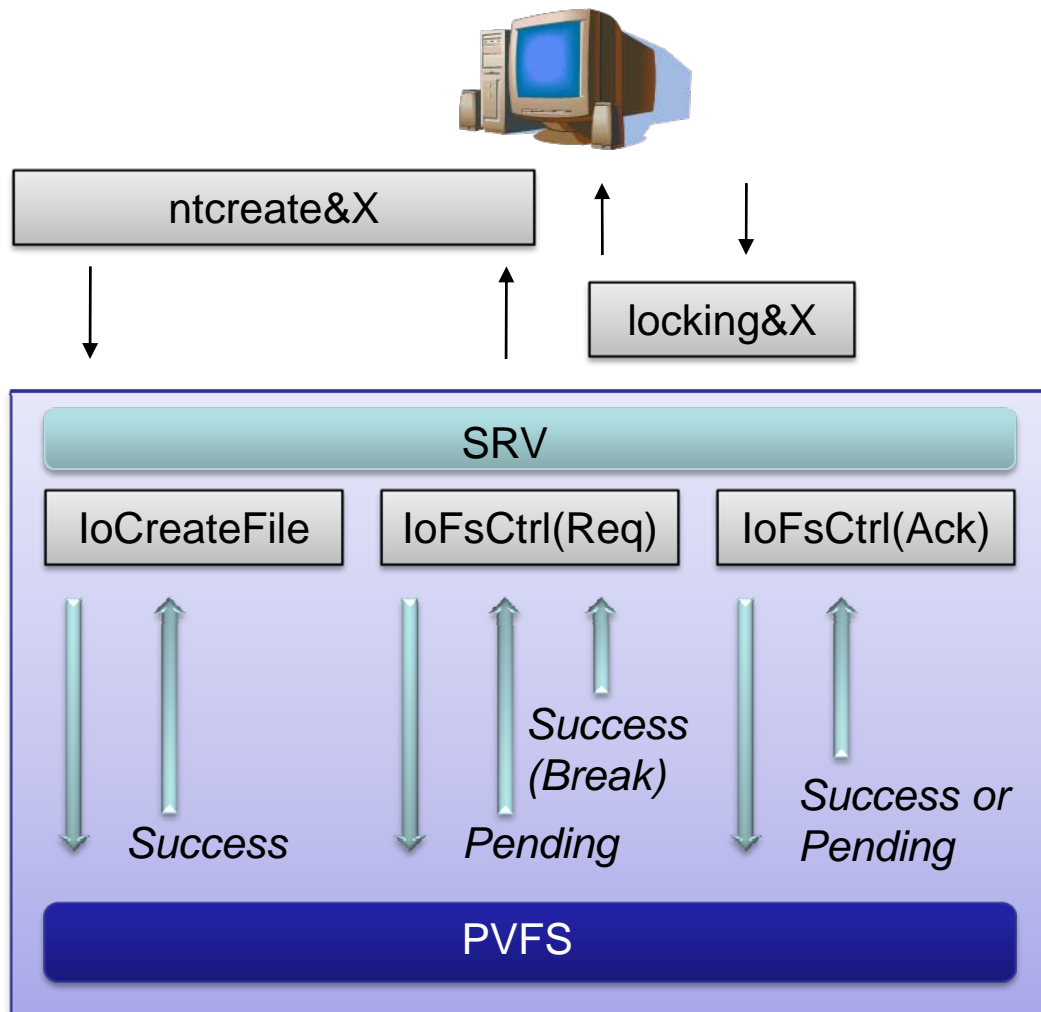
PVFS – Data Structures (cont)

- ❑ File Control Block represents the file on disk
 - ❑ FCB is removed when last open handle is closed
- ❑ Create Control Block is open file handle
 - ❑ Stored in the `IO_FILE_HANDLE`
 - ❑ Lwlo API is handle based (i.e. All files and directories are processed first through `CreateFile`)
- ❑ CCB refers to its FCB; FCB owns a list of its CCBs

- ❑ Share modes and byte range locking information is stored with the open handle in the CCB
- ❑ A share mode or BRL check checks all associated CCBs until a conflict is detected or success
 - ❑ `PvfsEnforceShareMode()`, `PvfsCanLock()`, `PvfsAddLock()`
- ❑ Pending locks are stored on the FCB
 - ❑ Backlink to the requesting CCB
 - ❑ Processed on any change to the lock table

Oplocks

- ❑ Legacy oplocks
- ❑ Requested using FsloCtrl on CCB
- ❑ Oplock list stored on the FCB
- ❑ Deferred ops stored in a queue on the FCB



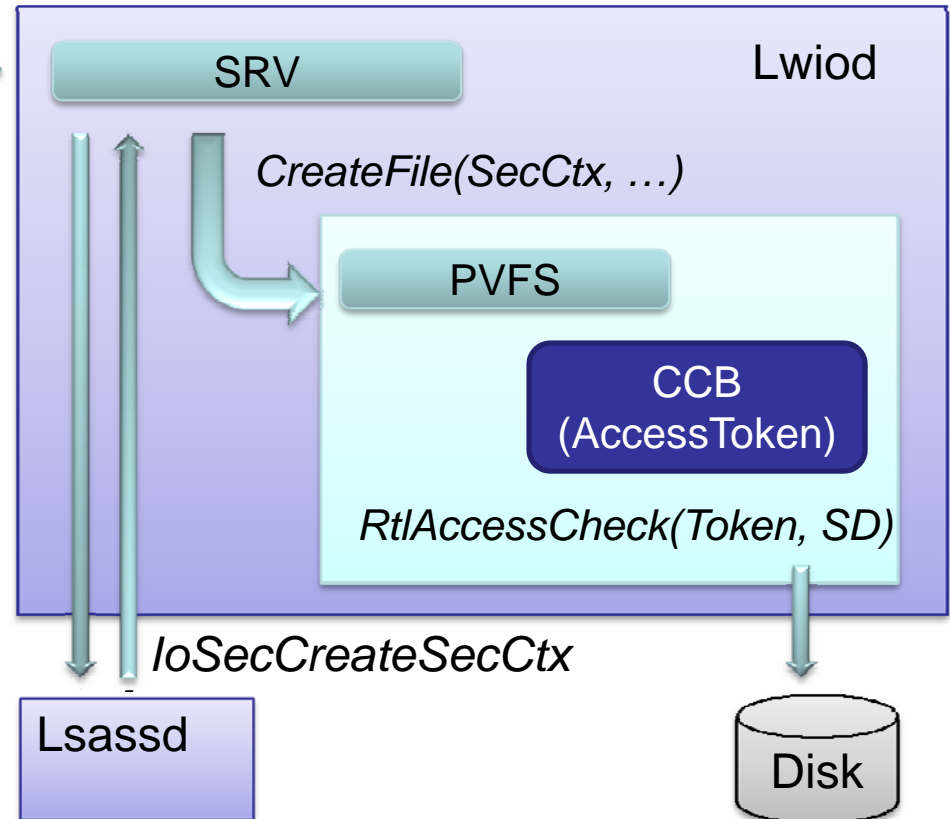
CREATE_SECURITY_CONTEXT



SessionSetup&X



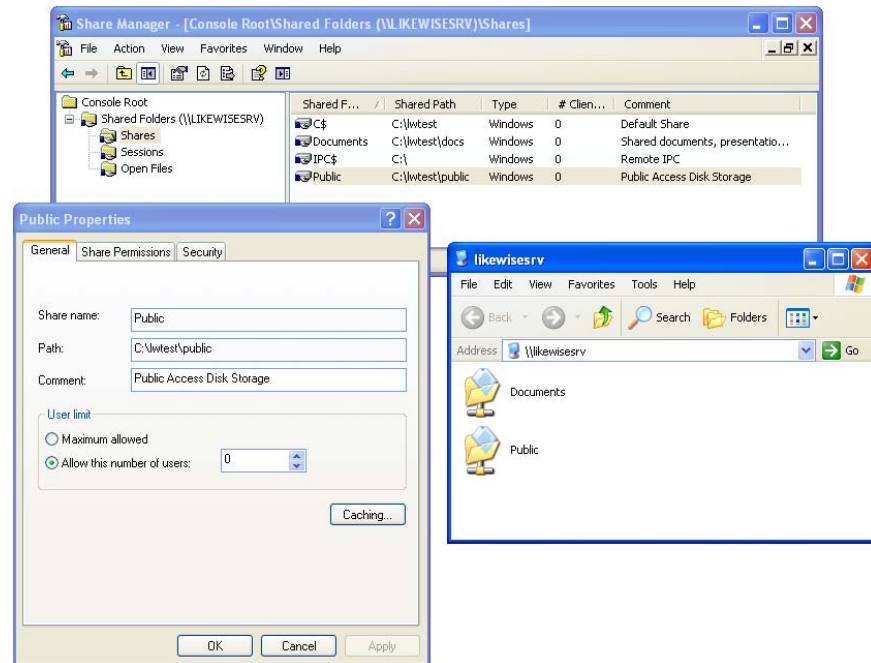
- ❑ Obtained from Lsassd during SessionSetup processing
- ❑ Passed to IoCreateFile()
- ❑ Contains user's Access Token



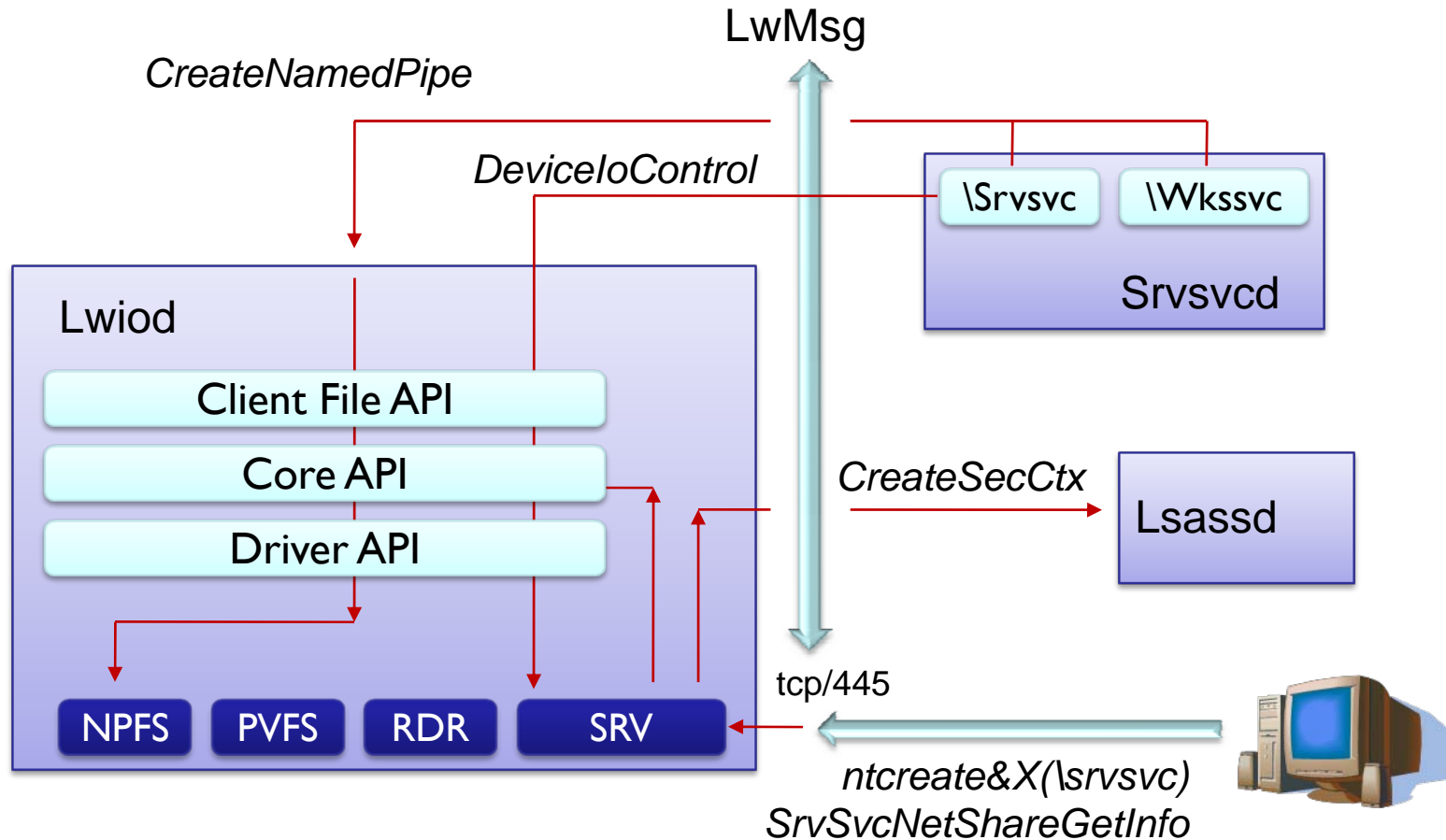
Server & Workstation Service

Server Service (srvsvcd)

- ❑ Implements the Srvsvc & Wkssvc RPC interfaces
- ❑ Retrieves information about file shares from `Lwiod/SRV LwNtDeviceIoControlFile()`



Server Service (cont)



Building Likewise CIFS

- ❑ Simple build system for Linux & FreeBSD
- ❑ Step 1: Download the source code
 - ❑ `$ git clone git://git.likewiseopen.org/likewise-open`
- ❑ Step 2: Build the likewise-open components
 - ❑ `$ build/mkcomp [--noincremental] [--debug] all`
 - ❑ Installs all pieces to “staging/install-root/”
- ❑ Step 3: Generate RPMs/DEBs (Linux only)
 - ❑ `$ build/mkpkg [--debug] all`
 - ❑ Creates packages in “staging/packages/”

Questions?

gcarter@likewise.com

<http://www.likewiseopen.org/>

<git://git.likewiseopen.org/likewise-open>