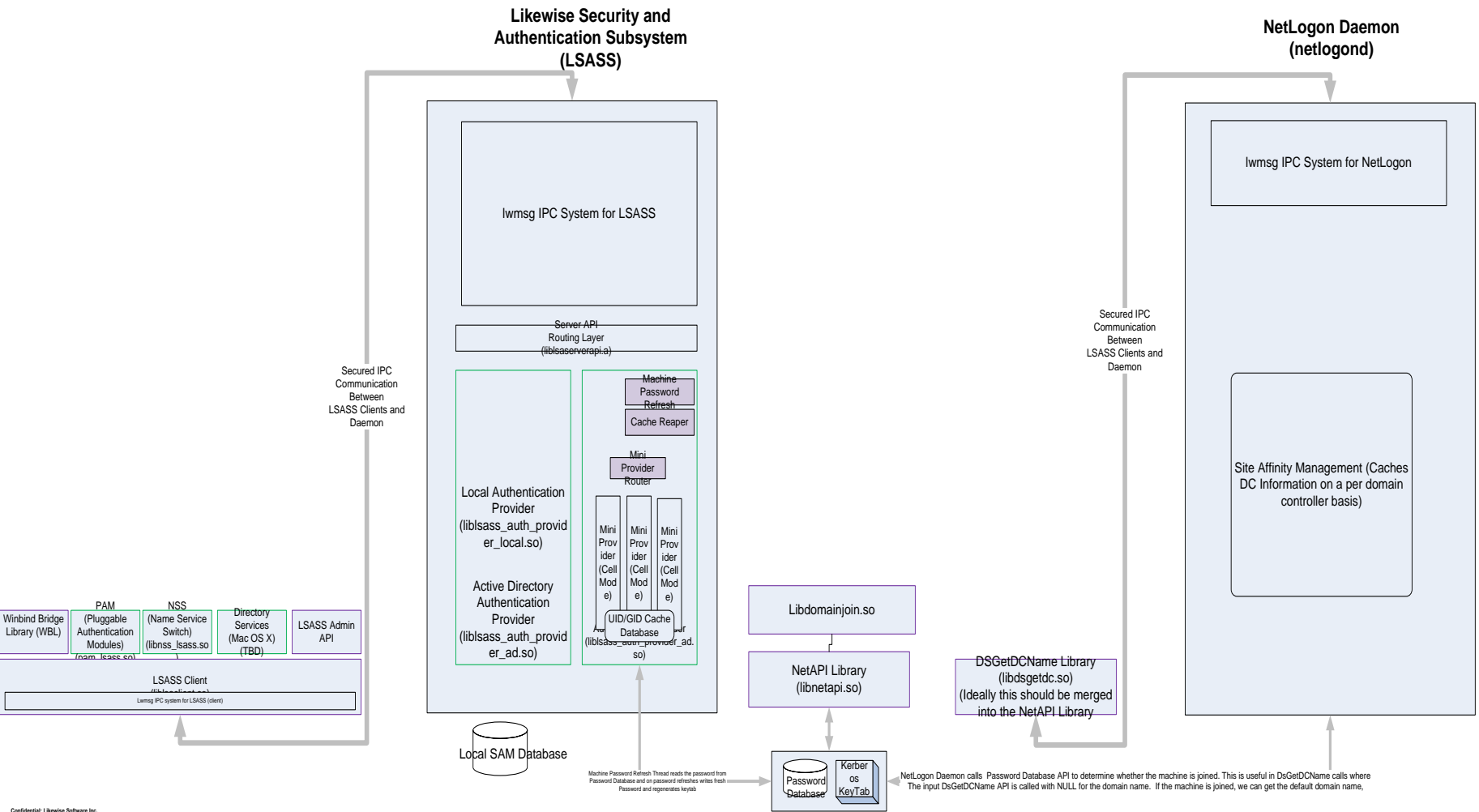


The LWIO Architecture



Krishna Ganugapati, Danilo Almeida,
Jerry Carter, Brian Koropoff,
Sriram Nambakam, and Rafal Szczesniak

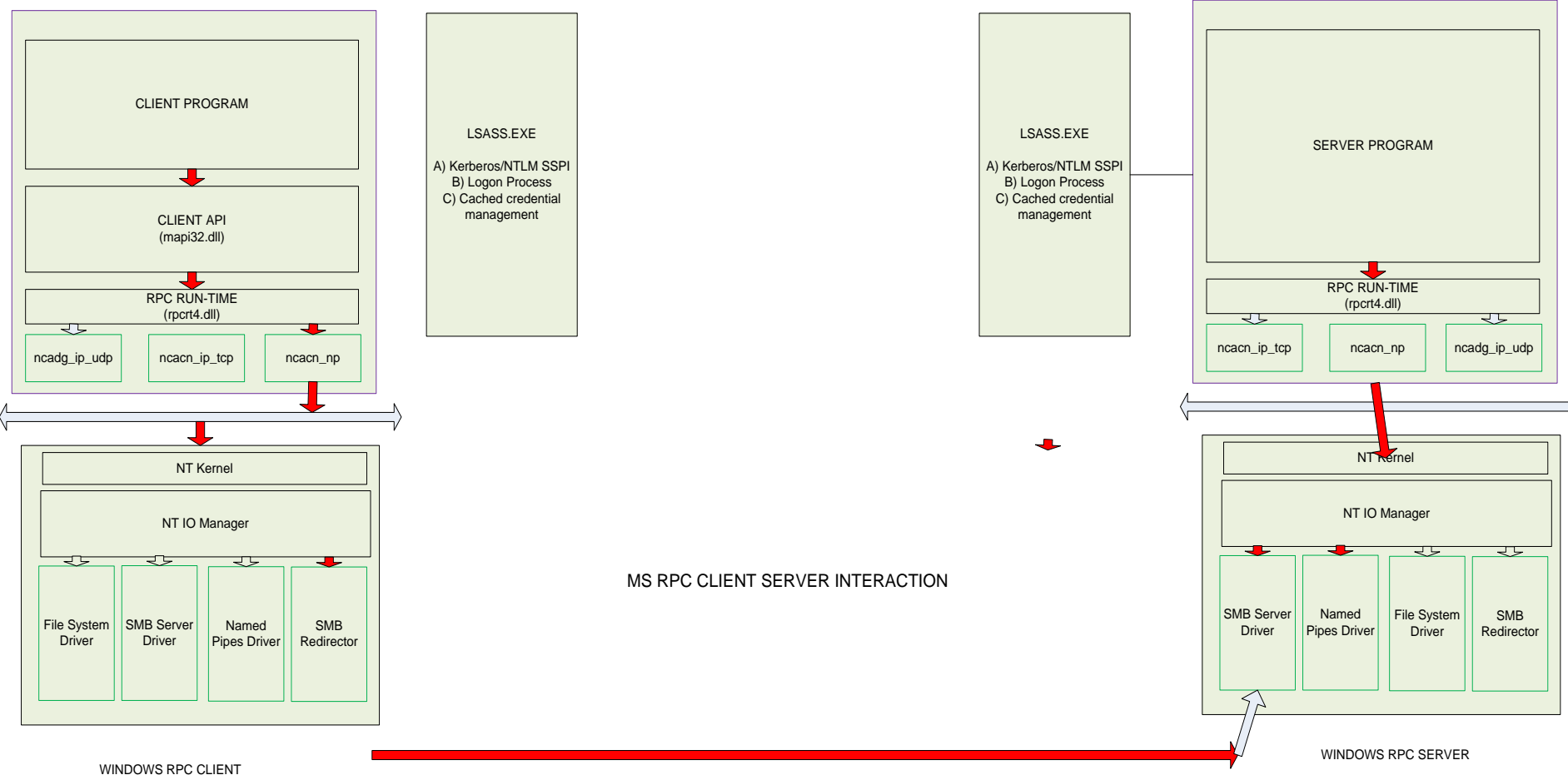
December 2008 What we had: The LWIS Authentication Infrastructure



Confidential: Likewise Software Inc.

- Product requirement: Privileged user management
- Scenario - Remote management of local users and groups on a Linux machine
- Technical requirements
 - Create local SAM for Linux
 - Server side DCE/RPC support over named pipes
 - Build named pipe file system
 - Build minimal SMB server for named pipes
 - Retrofit server side named pipes to DCE/RPC
- Goal: Do this by April 2009

MS RPC Client Server Interaction (circa 1989-1993)

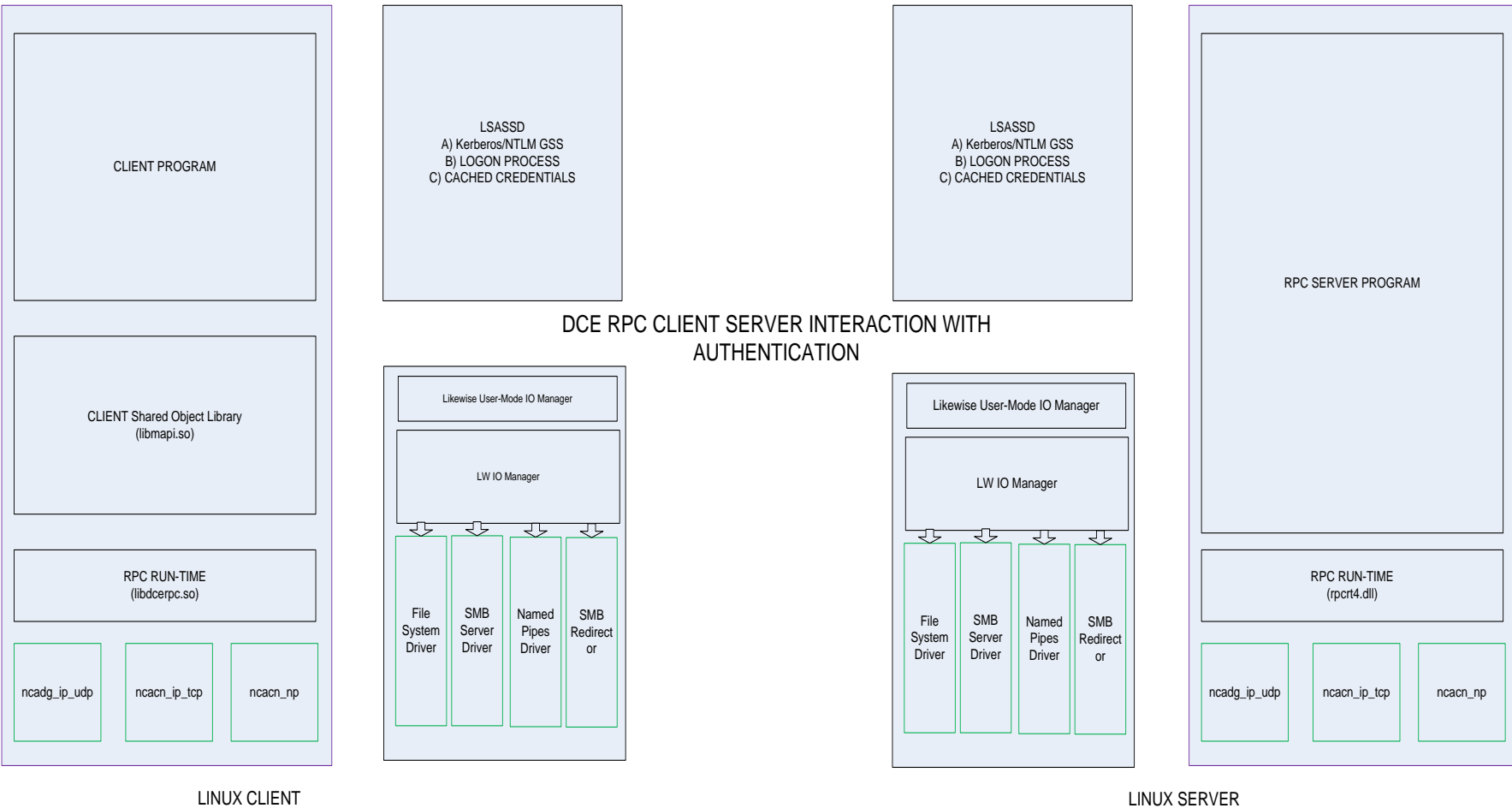


- Kernel subsystem that manages the communication between applications (in user space) and device drivers
- Communication is done primarily between operating system and device drivers through I/O Request packets (IRPs)
- Not really packets; more like a well-defined data structure
- Layered I/O Model: Drivers can pass IRPs to lower level drivers through the I/O Manager
- IRP processing is asynchronous; drivers receive IRPs; can queue them for delayed processing; and when complete, call a caller-registered completion routine

LWIO is

- a user space I/O manager
- pluggable driver model
 - SMB redirector
 - SMB server
 - named pipe file system
 - Posix virtual file system
- a base run-time library - lwbase

Likewise DCE RPC Client Server Interaction with Authentication



- Four month development cycle
- Milestones
 - M1 – Demonstration of Windows client copying files from a Linux server over the new SMB framework
 - M2 – Named pipe client and named pipe server
 - M3 - Echo DCE/RPC server over named pipes
 - M4 - MMC running “Local users and groups” on Windows remotely adding local users and groups on Linux
- This was the plan..

- Part I: Introduction and Overview
 - Overview
 - LWIO Internals
- Part 2: Scenarios and Demos
 - The SMB Redirector
 - The SMB Server
 - The Samr/Lsa RPC Servers
- Wrap-Up and Q&A

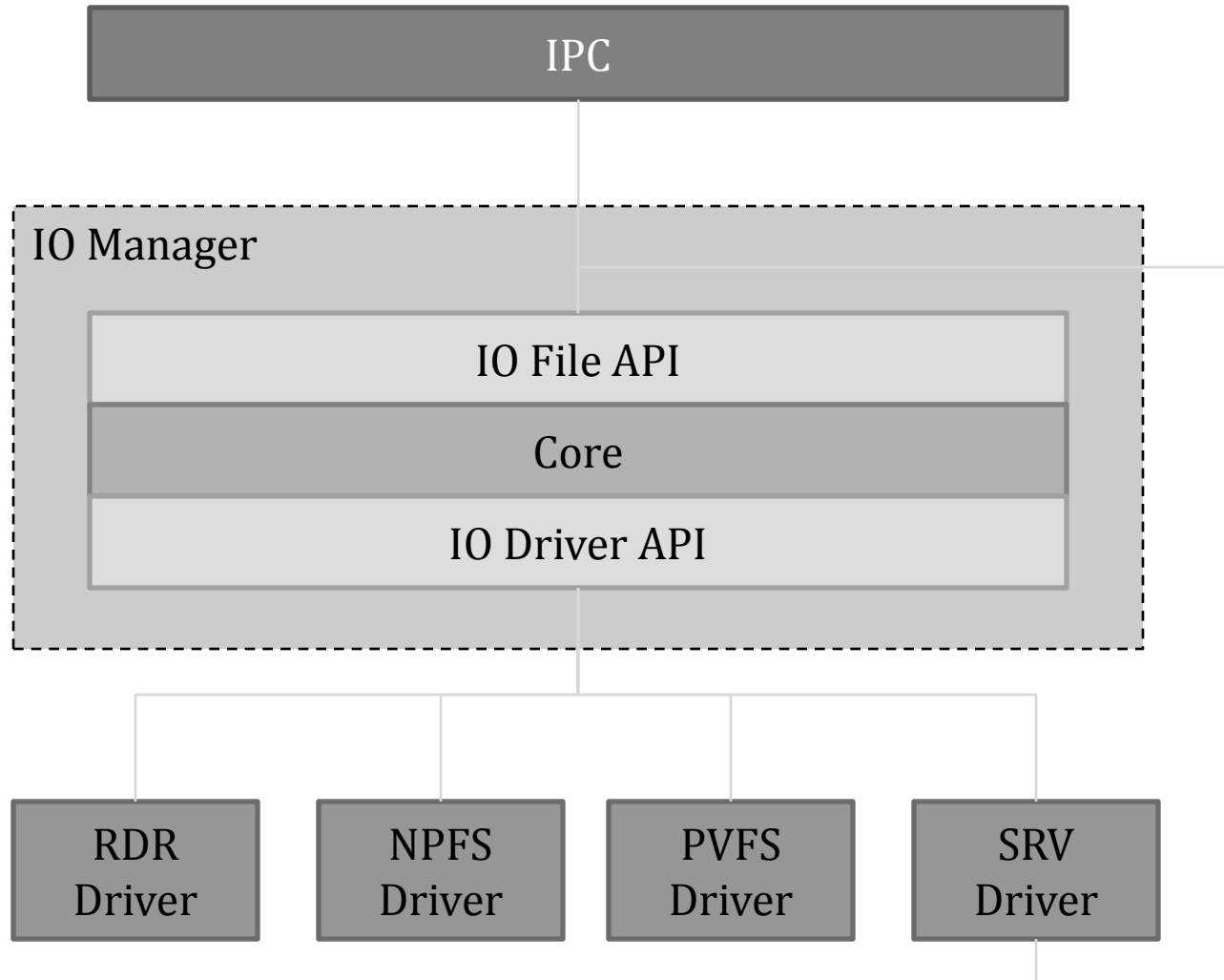
Part I: LWIO Internals

Danilo Almeida

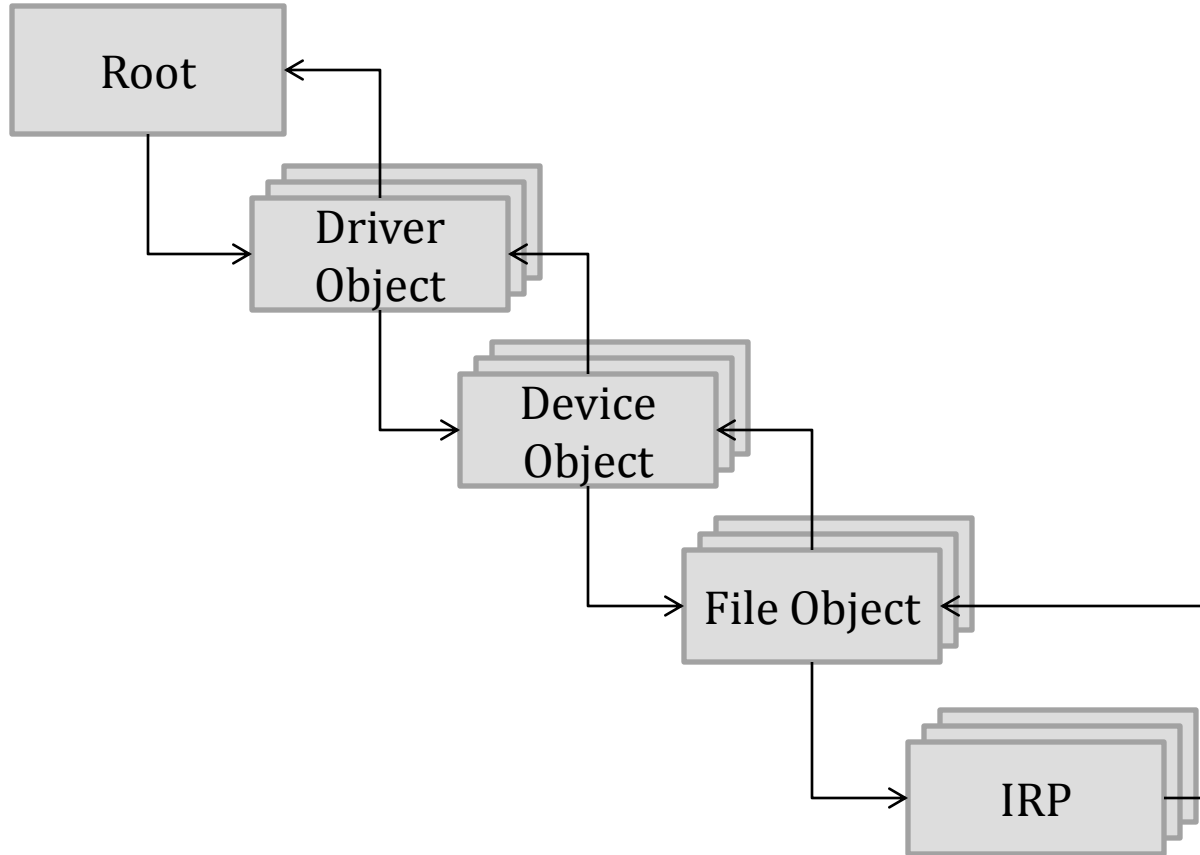


- IO Manager
 - Modeled after Windows IO model, but simpler
 - Everyone is a driver
 - SMB Redirector
 - SMB Server
 - Local File System
 - Named Pipe File System
 - Drivers can call each other via common IO model
- Advantages
 - Small interface
 - Easy to add functionality
 - Based on existing documented IO model (WDK)

- LWIO Consists of
 - liblwioclient library that IPCs to server daemon
 - lwiod server daemon
- lwiod Includes
 - Daemon scaffolding
 - IPC Interface
 - IO Manager
 - Loadable drivers



IO Manager Objects



- Shared object provides a DriverEntry function
 - Called by IO Manager
 - Calls IoDriverInitialize()
 - Driver-specific context
 - Shutdown callback
 - Dispatch callback
 - Instantiates named Device Objects
- Drivers interface with IO Manager
 - IO Driver APIs
 - IO File APIs

- IoDriver{Initialize,GetName,GetContext}
- IoDevice{Create,Delete,GetContext}
- IoFile{Get,Set}Context
- Cancellation support (not yet implemented)
 - IoIrpSetCancelCallback
- Asynchronous support (not yet implemented)
 - IoIrpComplete

- Io{Create,Close}File()
 - Returns or closes a PFILE_OBJECT
- Io{Read,Write}File()
- IoDeviceIoControlFile()
- IoFsControlFile()
- IoFlushBuffersFile()
- Io{Query,Set}InformationFile()
- IoQueryDirectoryFile()
- Io{Query,Set}VolumeInformationFile()
- Io{Lock,Unlock}File()
- Io{Query,Set}SecurityFile()

- `IoCreateFile(path)` → File Object
 - Resolve path to Device Object
 - Dispatch `IRP_TYPE_CREATE` IRP with remaining path to corresponding Driver Object
 - Can open or create files and directories
 - Driver checks security context against desired access
 - Extra Create Parameters (ECPs)
 - ECP list used in create to pass arbitrary arguments to be processed by IO manager or driver
 - Used for create named pipe arguments

- Io<Operation>File(File Object, arguments...)
 - Check for necessary granted access on File Object (to be implemented)
 - E.g., IoReadFile() requires READ_FILE access
 - Create IRP for <Operation>
 - IRP_TYPE_<OPERATION>
 - Points to File Object
 - Additional operation-specific arguments
 - Call File Object's Device Object's Driver Object's dispatch callback

- Pointer to driver-allocated context
- List of Device Objects
- Shutdown Callback
- Dispatch Callback

- Pointer to Driver Object
- Pointer to driver-allocated context
- List of File Objects
- Name

- Pointer to Device Object
- Pointer to driver-allocated context
- List of pending IRPs
- Granted access mask (to be implemented)

- Type
 - IRP_TYPE_{CREATE,CLOSE,READ,WRITE,etc.}
- Pointer to File Object
- Type-dependent arguments

- Driver supports cancellation by calling `IoIrpSetCancelCallback(Irp, Callback, Context)`
 - Callback is NULL to disable cancellation
- Cancellation triggered on:
 - `IoCancelFile()` – to be implemented
 - Shutdown
- Cancellation causes IO to complete or return `STATUS_CANCELLED`
 - Driver cancel callback is called to tell driver to complete or cancel IO

- IoXxxFile(FileObject, AsyncControlBlock, ...)
- Async Control Block
 - Callback
 - Context
- Driver returns STATUS_PENDING
- Driver calls IoIrpComplete(Irp)
 - Completes IO by calling callback in async control block

- Server-side queues completed IO
- Client-side library function reads next IO completion from queue
 - Trigger callbacks for client-side callers

- **Directories**

- include
- client
- **server**
- ipc
- (others)

- **server**

- include
- iomgr2
- lwiod
- smbwire
- rdr
- srv
- pvfs
- npvfs

- **iomgr2**

- ioapi.c
- ioconfig.c
- iodevice.c
- iodriver.c
- iofile.c
- ioinit.c
- ioipc.c
- ioirp.c
- iomem.c
- ioroot.c
- iosecurity.c

- **pvfs**

- acl.c
- acl_xattr.c
- alloc.c
- attrib.c
- attrib_xattr.c
- ccb.c
- **close.c**
- **create.c**
- create_dir.c
- create_file.c
- **deviceio.c**
- **driver.c**
- errno.c
- fcb.c
- file*Info.c
- **flush.c**
- **fsctrl.c**

- **pvfs (cont)**

- globals.c
- lock.c
- locking.c
- pathcache.c
- **querydir.c**
- **queryfile.c**
- **querysecdesc.c**
- **queryvol.c**
- **read.c**
- **setinfo.c**
- sharemode.c
- string.c
- syswrap.c
- unixpath.c
- util_*.c
- wildcard.c
- **write.c**

- `gdb lwiod`
 - `break loplpcCreateFile`
 - `continue`
- `lwio-tool createfile /pvfs/tmp/test.txt`

Part II: Scenarios and Demonstrations



- Scenario 1: The SMB Redirector
- Scenario 2: The SMB Server and PVFS
- Scenario 3: The Named Pipe RPC Server System
- Summary

Part II: The SMB Redirector

Brian Koropoff



- History
- Architecture
- FUSE Integration
- Remaining Work

- Issues with existing libraries
 - Thread safety
 - Direct access to NT semantics/errors
- Initial scope limited
 - Minimal subset needed for RPC
- Evolution
 - LWIO driver
 - General file operations

- Packet encoding/decoding
 - Uses packed data structures when possible
- I/O
 - Dedicated receiver thread per socket
 - Sending performed synchronously
- Connection lifecycle
 - Lazy initialization, reference counting
 - Asynchronous connection reaper

- Aggressive caching
 - Connections
 - Sessions
 - Trees
- Reaper
 - Notified on zero reference count
 - Dedicated thread closes expired objects
 - Careful locking ensures consistency

- Exercises LWIO client API
- Supports essential operations
 - stat, readdir
 - open, truncate
 - read, write
 - unlink, rename

- Core
 - Threading model
 - Static thread pool
 - Asynchronous call support
 - Oplocks
 - NTLM
- FUSE
 - Permission setting
 - Convert to LWIO driver

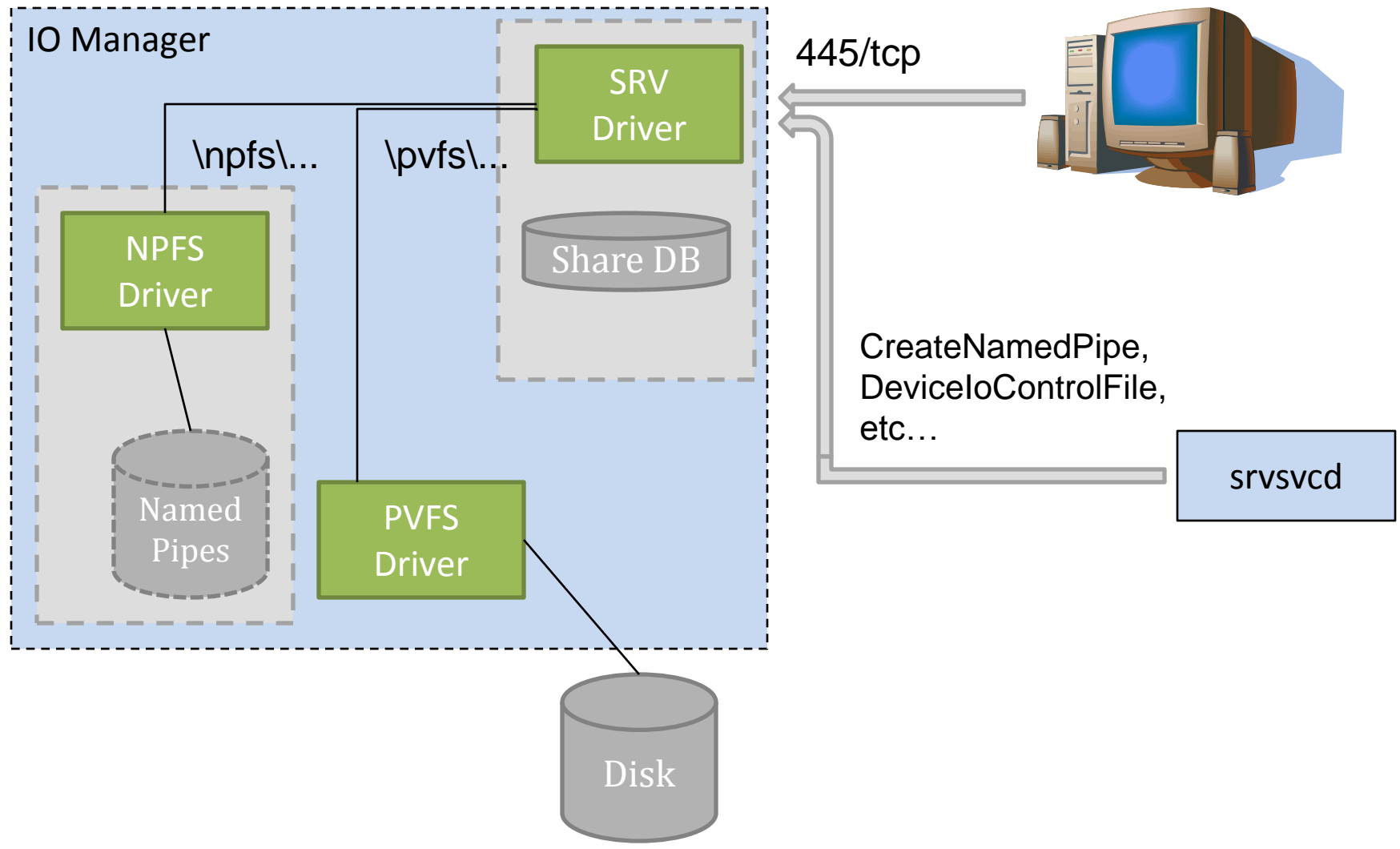
**The SMB Server
and
The Posix Virtual File System**

Jerry Carter and Sriram Nambakam

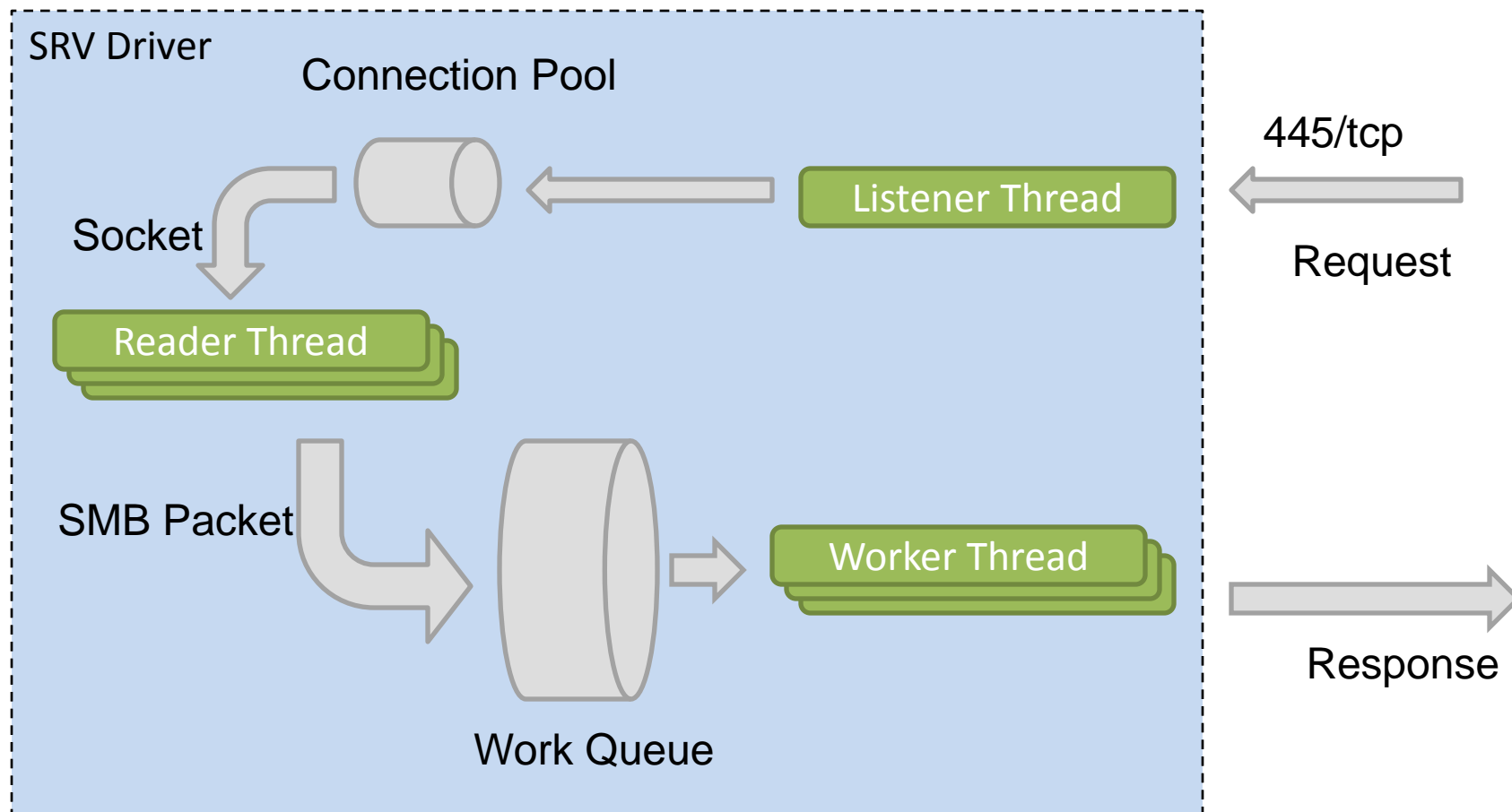


- Composed of three drivers
 - `srv.sys.so` – SMB Protocol head
 - `pvfs.sys.so` – POSIX user-space file system
 - `npfs.sys.so` – Named Pipe file system
- Topics
 - Driver interaction
 - SRV threading model
 - PVFS data structures
 - AccessTokens and authorization

SMB Server - Overview

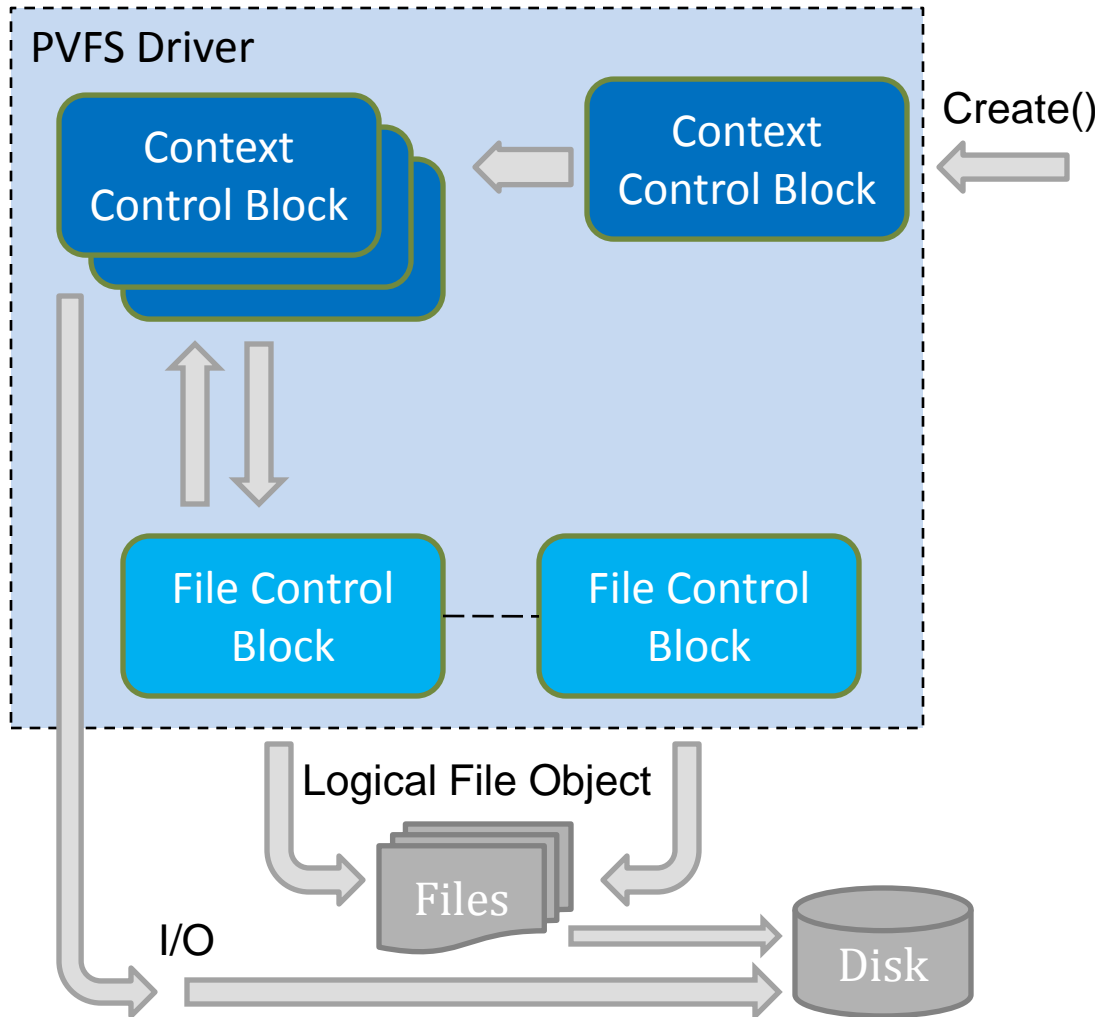


- SRV
 - Listens on 445/tcp
 - Owns the file share database
 - Call into the IoManager to access the FS drivers
- PVFS
 - User space file system
 - Makes use of EAs for storing Attributes and SDs
- NPFS
 - In memory Named Pipe file system



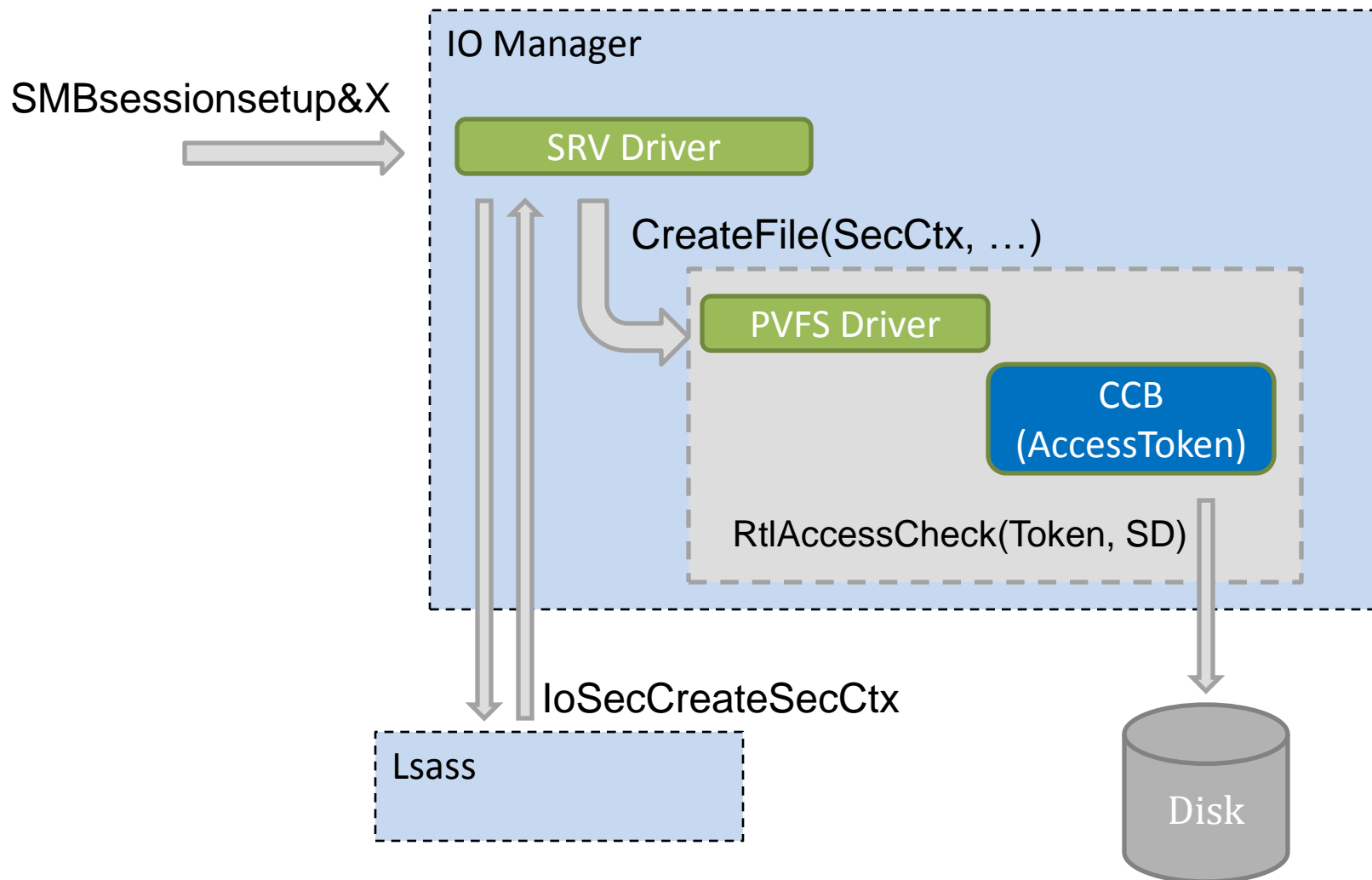
- Listener thread (1)
 - Handles new connections on 445/tcp
 - Hands the new socket off to an available Reader
- Reader Thread Pool (N == number of cores)
 - Manages a connection for its entire lifetime
 - Add new SMB packets to the Work Queue
- Worker Thread Pool ($2 * N$)
 - Grabs next available work item, processes it, and responds to the client

- Utilizes packed structures for marshalling SMB requests
 - Pointers to parameters and data sections
- Buffer pool is used for packets
 - Work queue, responses, etc...



- FCB – File Object
- CCB - Open File
 - Filename
 - File descriptor
 - Dev/Inode
 - Share modes
 - BRL tables
 - FindFirst data

- File Control Block represents the on disk
 - FCB is removed when last open handle is closed
- Context Control Block is open file handle
 - Stored in the `IO_FILE_HANDLE`
 - Maintains granted share modes and BRLs
 - LwIo 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



- SRV authenticates a user session
- User principal name is used to build and IoCreateSecurityContext handle
 - Handle is stored in the session table and passed to all IoCreateFile() calls
- PVFS uses the IoCreateSecurityContext to request the user's Access Token which is then stored as a reference in the CCB
- AccessToken used to call RtlAccessCheck()

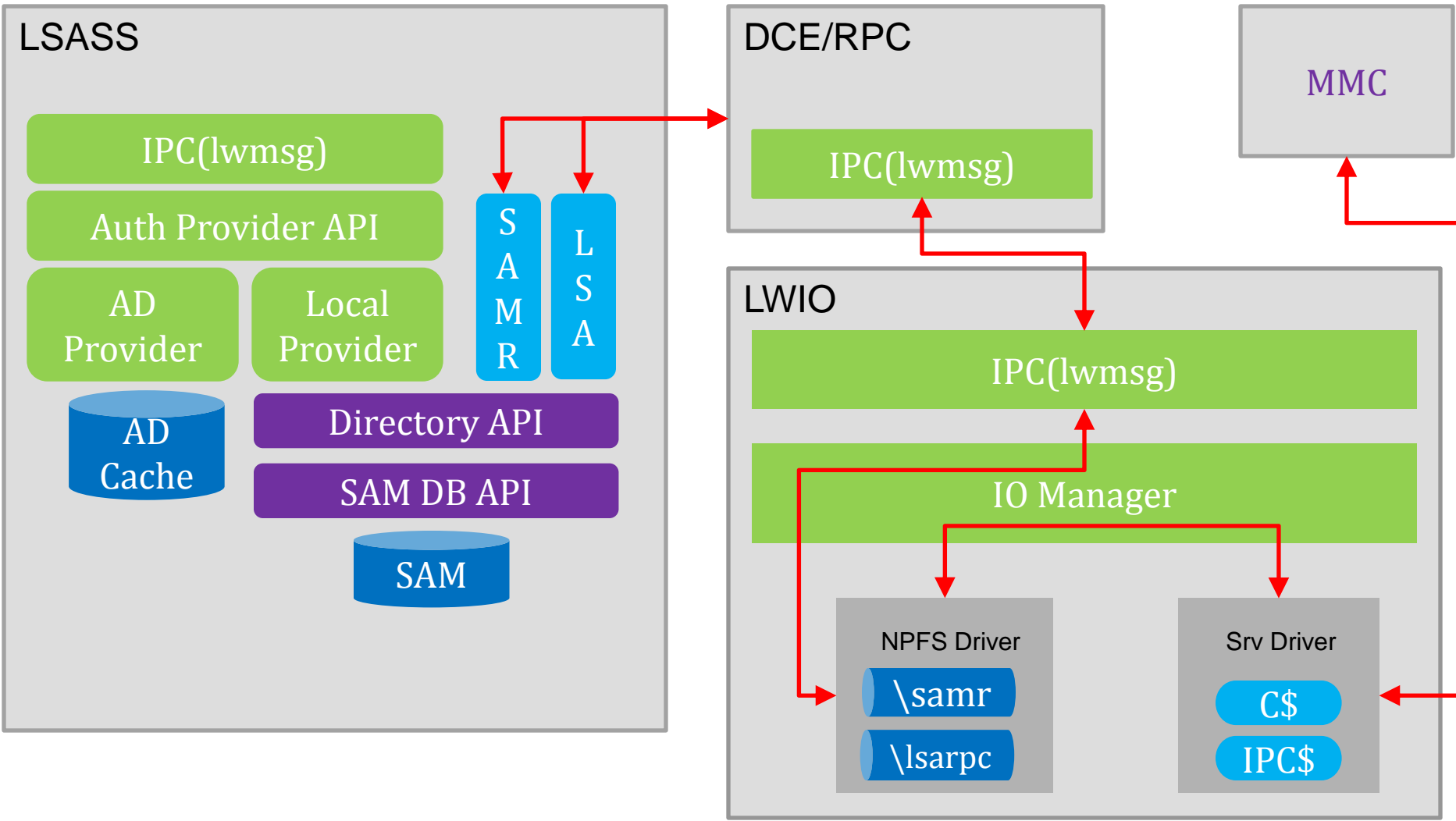
- PVFS focuses on correct semantics
 - Performs access checks in user space using the Security Descriptor and Access Token
- Security Descriptors are in file system extended attributes using Self-Relative form
 - No loss of fidelity; no current integration with POSIX ACLs
- Win32 byte range lock manager
 - Integration with POSIX locks is a future feature

Part II: The Named Pipe DCE/RPC Subsystem

Rafal Szczesniak



LSASS :: Local Provider and RPC Server Interfaces



“Houston, Tranquility Base. The Eagle had landed 😊”

From: Rafal Szczesniak
Sent: Tuesday, April 21, 2009 9:27 AM
To: lwio-core
Subject: Houston, Tranquility Base here. The Eagle has landed :-)
Attachments: mmc-users3.png; signature.asc

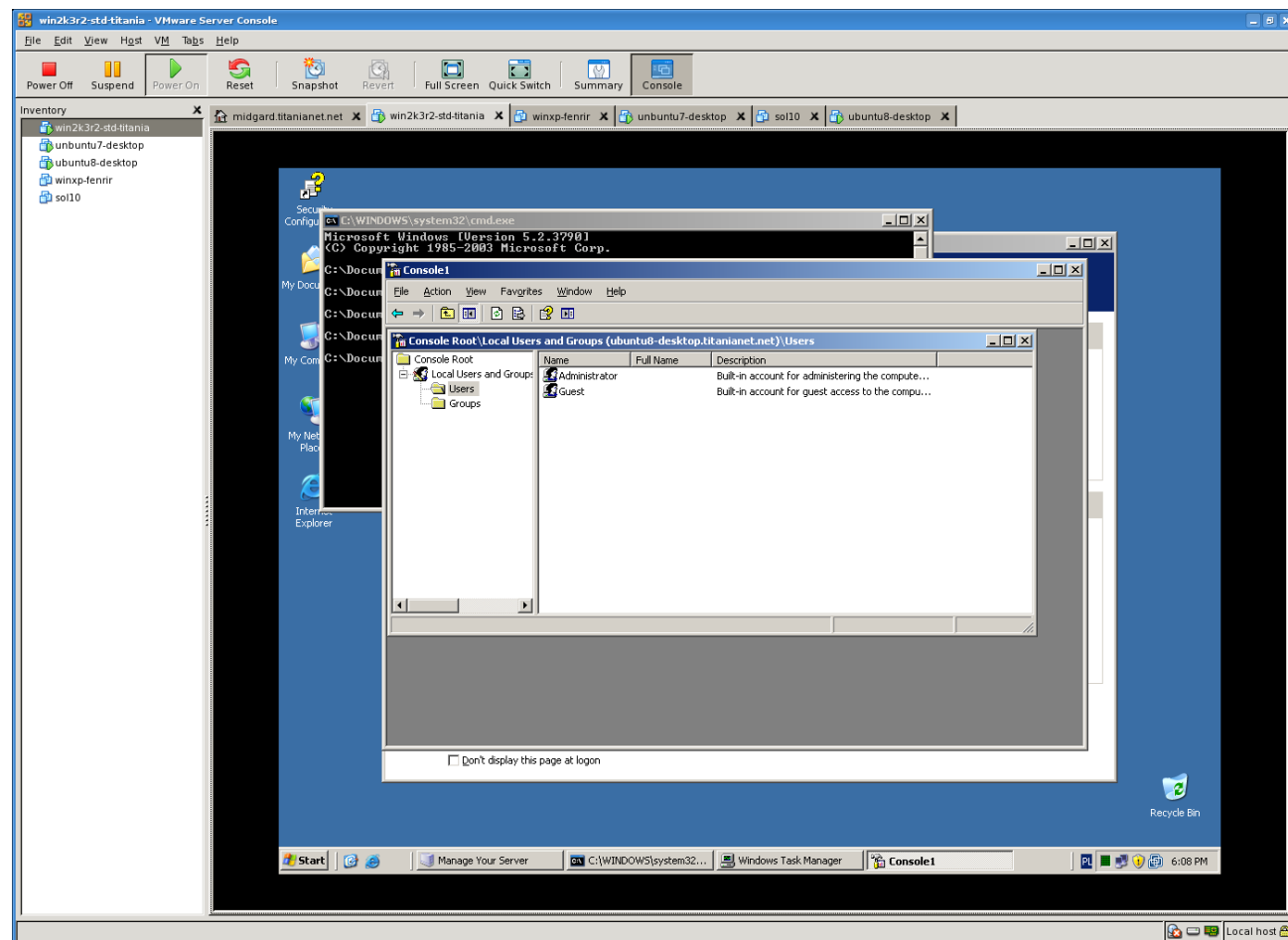
I'll check this stuff in once I get rid of one hack from srvsvcd...

We probably can't do anything else but enumerate users at the moment (haven't tested yet), but it's a nice change when compared where we were last week.

cheers,

--

Rafal Szczesniak
Samba Team member
<http://www.samba.org>
Likewise Software
<http://www.likewise.com>



- Iwio – A new architecture with some interesting results
- We have a long way to go..
- We're by no means complete or full-featured .. yet 😊
- The SMB Server has proceeded far more quickly than we expected – a happy side-effect!
- We plan on continued development of the server
- The Licensing is
 - GPL for all daemon code
 - LGPL for all client libraries

Thank you

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

Get the code:

```
git clone git://git.likewiseopen.org/likewise-open.git
```

krishnag@likewise.com

dalmeida@likewise.com

bkoropoff@likewise.com

gcarter@likewise.com

snambakam@likewise.com

rafal@likewise.com