Your File System

The Native AFS Client on Windows
The Road to a Functional Design

Jeffrey Altman, President

Your File System Inc.

14 September 2010

The Team

- Peter Scott
  - Principal Consultant and founding partner at Kernel Drivers, LLC
  - Microsoft MVP
- Jeffrey Altman
  - OpenAFS Gatekeeper and Elder
  - President of Your File System, Inc.

SMB …

- The Windows AFS architecture developed by Transarc leveraged the SMB redirector to pass file system requests to the AFS Cache Manager
- Microsoft Loopback adapter used to permit local NetBIOS name binding of \\AFS
- “Easier to implement” but reliant on Microsoft system components
  - Hard to get bugs fixed in these modules
  - Not very performance focused
    - Generic solution to fit all situations
  - Typical Microsoft interface … minimal documentation

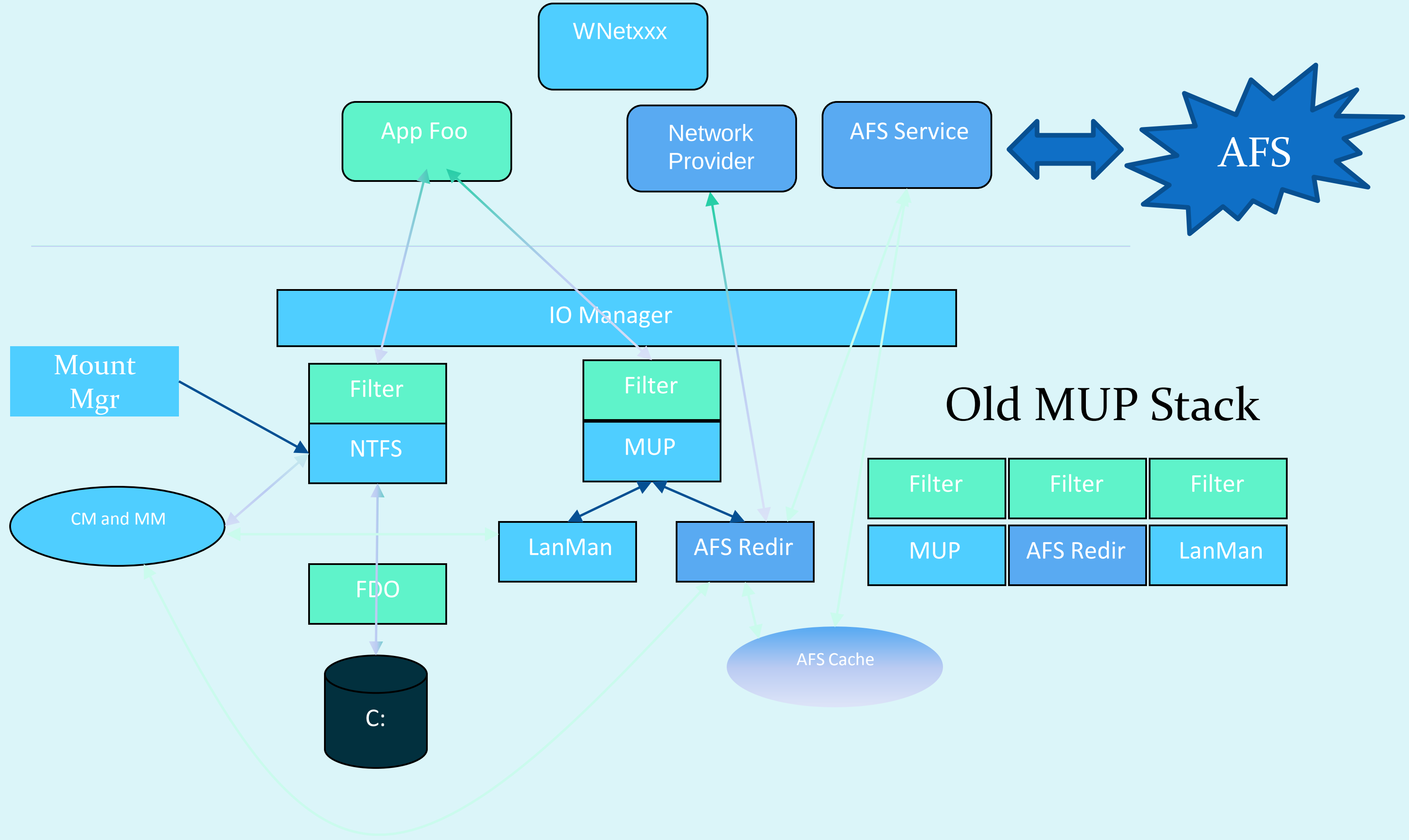
Goals of the Design

- Need to leverage as much functionality within the AFS Service as possible
  - Keep all server communication in service
    - Data retrieval
    - Callback registration and notification
    - Metadata management
- Complete integration into the Microsoft IFS (Installable File System) API
- Stability and performance
- Easy rollback to SMB interface without uninstall

Windows File System Model

- Windows IFS Interface
  - IRP (I/O Request Packet) based
  - ‘Fast IO’ Interface used for more than just I/O
  - Network Provider Interface for Network Redirectors only
- A network file system is not much different from a local file system, in Windows
  - **MUP (Multiple UNC Provider) Registration**
    - Pre-Vista uses different model
  - **IOCTL_REDIR_QUERY_PATH(_EX)**
    - \\afs\your-file-system.com\user\foo.txt
  - **Path Parsing**
    - \Device\MUP\;AFS\Redirector\;C:\AFS\your-file-system.com\user\foo.txt
      - \;C:\AFS\your-file-system.com\user\foo.txt
    - \device\MUP\AFS\your-file-system.com\user\foo.txt
      - \AFS\your-file-system.com\user\foo.txt
  - **Network Provider Library**
    - User mode interface for WNet API

Windows Internals



Windows Internals

- Windows Vista Changes
  - Memory Manger and Cache Manager changes
    - Theoretical limit of 4GB paging I/O requests but have not seen anything larger than 256MB
      - Pre-Vista had a maximum of 64KB
    - ‘Dummy’ pages in Memory Manager – does not effect redirector
  - MUP Changes
  - Tons of new ‘features’ – Bitlocker, built in AV, Indexer, Single Instance Storage, etc.

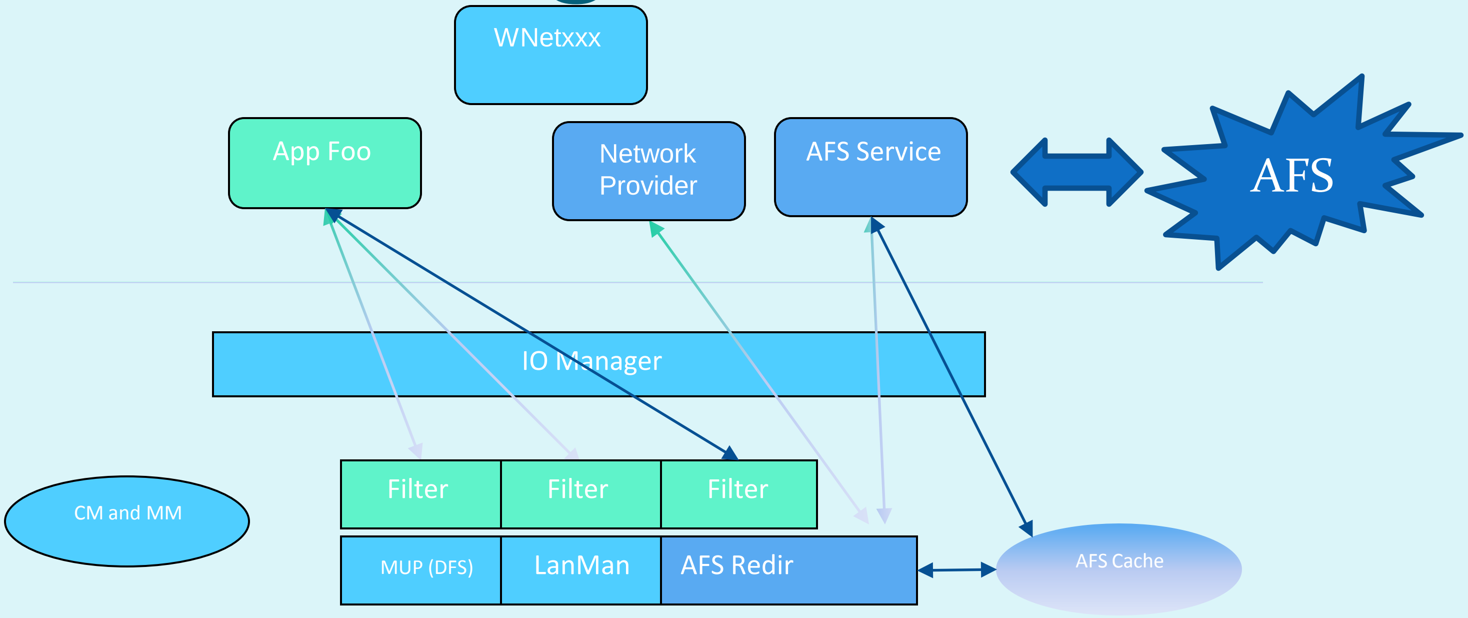
MUP Registration

- MUP – Handles mappings between the UNC name space and the file systems which manage them
- MUP changes in Windows Vista
  - Old model
    - Register with MUP using a named device object
    - Prefix resolution and IRP_MJ_CREATE requests handled by MUP, all others sent to file system
  - New Model
    - Register with MUP using an unnamed device object and a name of the file system control device

Old MUP Design

- Registration with MUP used a named device object
    - Prefix resolution by MUP used the IOCTL_REDIR_QUERY_PATH request
        - Cache entries for 15 minutes unless flushed
- IO Manager would send all requests, post IRP_MJ_CREATE, directly to file system
- Network redirectors would register, separately, as a file system resulting in filter attachment issues

Old MUP Design

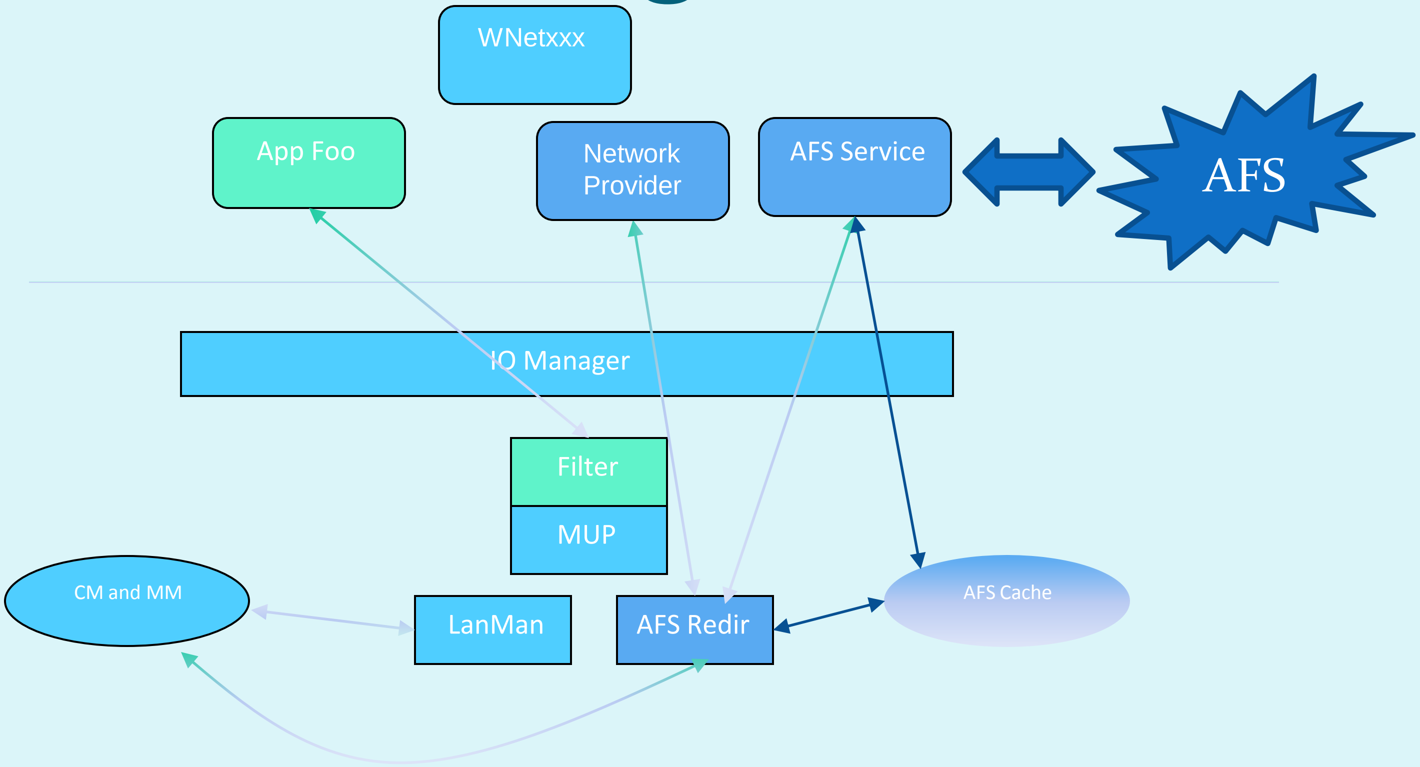


- Only prefix resolution and IRP_MJ_CREATE requests handled through MUP
- All subsequent requests issued to redirector

New MUP Design

- Register with MUP using a device name and an unnamed device object
  - Results in MUP creating a symbolic link from the device name to \Device\MUP
  - Prefix resolution using IOCTL_REDIR_QUERY_PATH_EX
- All requests go through MUP
- Single attachment point for filters

New MUP Design



- All requests go through MUP
- Single point access – Better?

Path Parsing in Windows

- 2 forms can be sent – drive letter or not …
- Drive letter names come into MUP as

    \Device\MUP\;AFS\Redirector\;C:\AFS\your-file-system.com\user

  Which are mapped by MUP into

    \;C:\AFS\your-file-system.com\user

- UNC names come into MUP as

    \device\MUP\AFS\your-file-system.com\user

  Which are mapped by MUP into

    \AFS\your-file-system.com\user\foo.txt

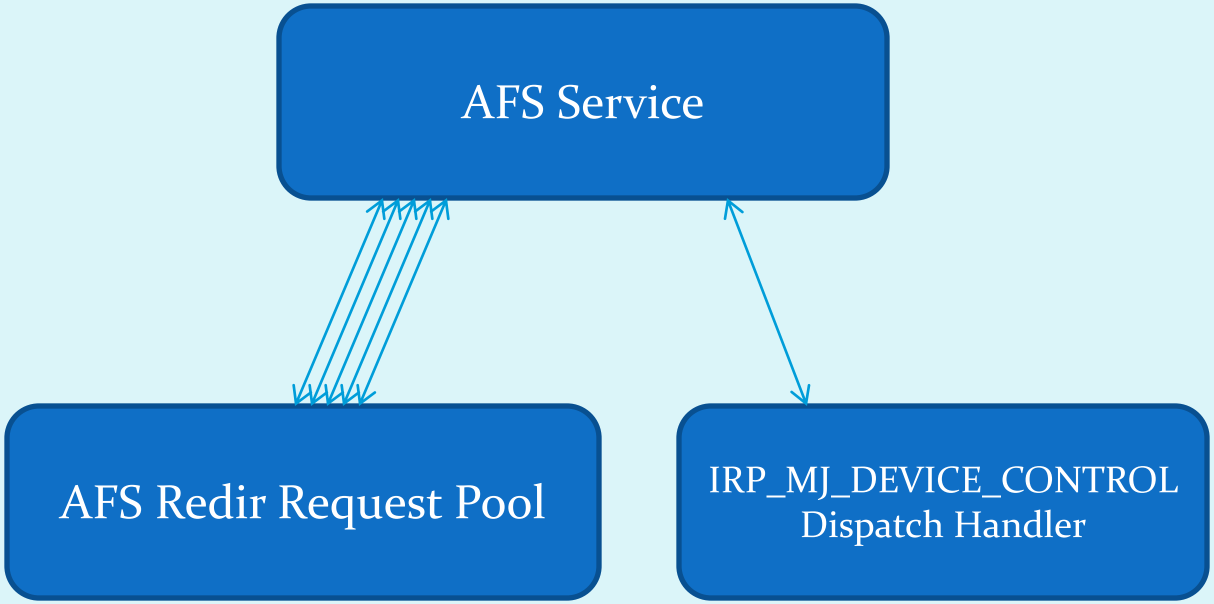
Network Provider Interface

- User mode library with supporting interface in file system
- Used to support WNet API in user mode
- Implements drive letter mapping and share browsing
- Communicates with file system for state and connection information
- Maintains per user information on mappings

AFS Service Communication

- Inverted call model
  - Requests from file system
  - Uses proprietary IOCtl interface
  - Communication through CDO (Control Device Object) symlink
- IOCtl interface
  - Requests to file system
  - Proprietary IOCtl interface for service initiated requests
- Cancellable interface through CDO handle

AFS Service Communication



- All requests issued through CDO symbolic link - \??\AFSRedirector
- Request pool state controlled through open handle

Merging Worlds

- Name space convergence
  - Symbolic Links – Microsoft and AFS
  - Mount Points
  - DFS Links
  - Component substitutions - @SYS
- File data handling
- PIOCtl Interface
- “Special” share name handling
  - PIPE\srvsvc
  - PIPE\wkssvc
- Network Provider Interface

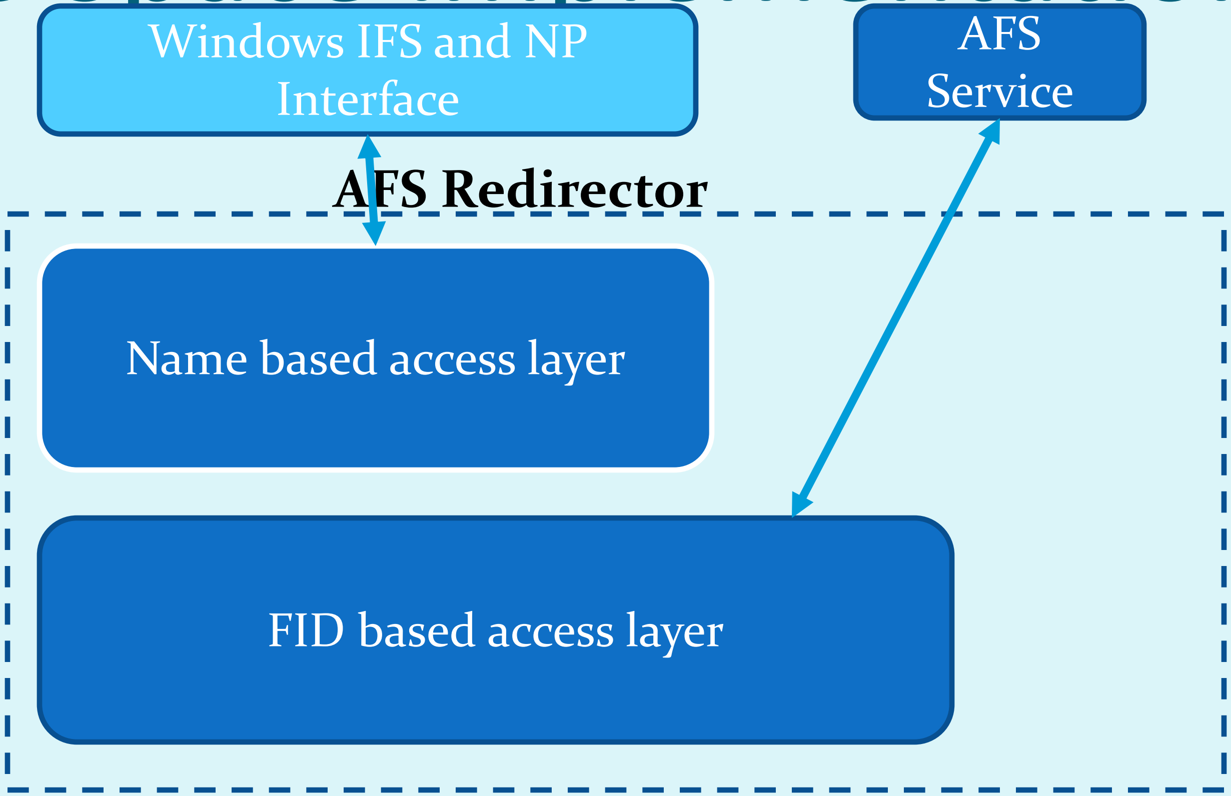
Name Space Convergence

- Cells and Shares
  - Share access mapped into cell names or volume names
    - \\AFS\your-file-system.com
    - \\AFS\your-file-system.com#root.cell
  - Dynamic discovery
- Reparse points and symbolic links
  - Must handle all symbolic links internally, they are not understood by Windows
  - Support the generic reparse point interface through FSCTL_xxx_REPARSE_POINT controls – no support to write this data
- Mount point processing managed internally
- DFS Links are supported through Windows reparse processing

Metadata Handling

- Redirector caching model
  - Cache objects based on FID on a per volume basis
  - Cache directory entries based on hash of name on a per directory basis
  - Support case insensitive, sensitive and short name lookups
  - Asynchronous pruning of trees when not in use
- Path name parsing in Windows
  - Path analyzed component by component, walking a specific branch for achieve the target object
  - Maintains a list of components used to access current target
  - Need to support relative symbolic links within a pathname

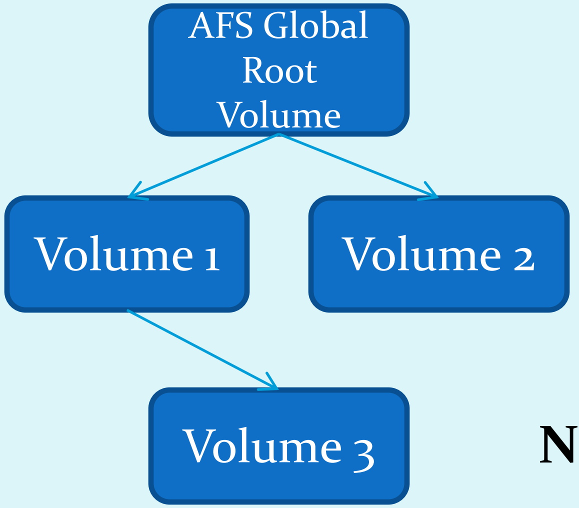
Name Space Implementation



- FID based access is ‘almost’ lockless – Only volume based lock required
- Name based access is complex due to symlink, mount point, DFS link and other abstractions not recognized by Windows

Name Space Implementation

Volume Btree (Cell, Volume)



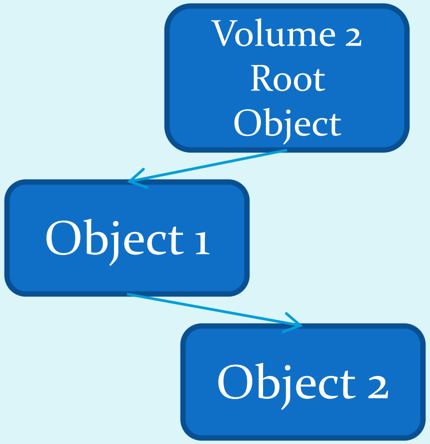
AFS Global Root Volume

Volume 1

Volume 2

Volume 3

Object Btree (Vnode, Unique)

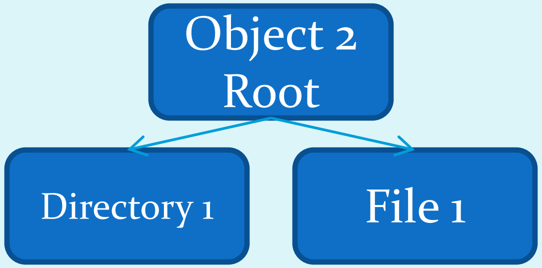


Volume 2 Root Object

Object 1

Object 2

Name BTree (Component CRC)

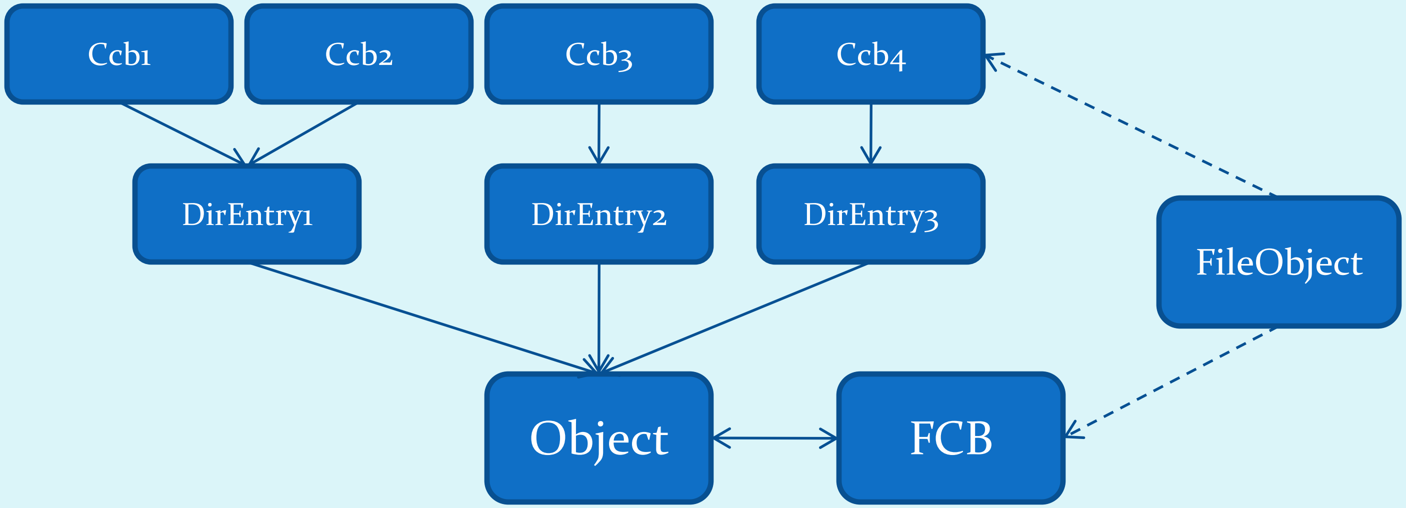


Object 2 Root

Directory 1

File 1

Name Space Implementation



- Handle AFS Symbolic Links, Mount Points, etc.
- DirEntry nodes are tracked per directory, contain name based information
- Object nodes are tracked by FID per volume
- FCB (File Control Block) nodes are used within the Windows IFS interface, tracked under the FileObject->FsContext pointer, one per Object node
- CCB (Context Control Block) nodes are used within the Windows IFS interface, tracked under the FileObject>FsContext2 pointer, one per open instance of a file

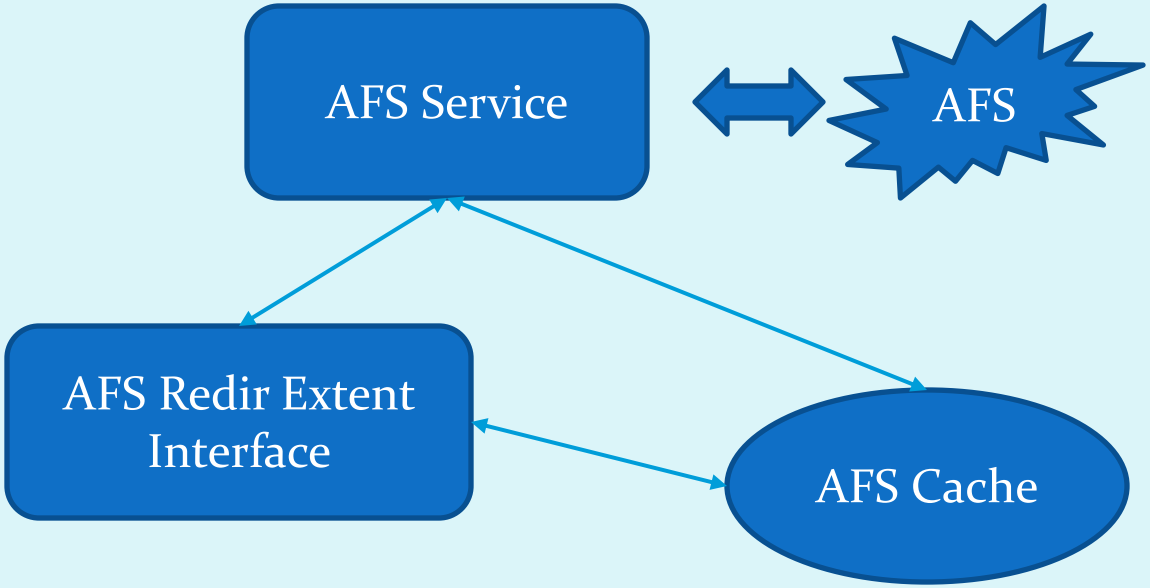
File Data Handling

- Windows caching model
  - Re-entrant model – Need to be careful of locking hierarchy
  - Side band locking interface for memory and cache manager components – Fast IO interface
  - Need to observe IRQ (Interrupt Request) levels while processing requests to underlying AFS Cache
- File Extent Interface
  - Extents describe the location of file data within the AFS Cache
  - Managed by the AFS Service and provided to the redirector upon request

File Data Handling

- AFS Caching
  - AFS Service populates AFS Cache with requested data and flushes dirty data back to server
  - AFS Redirector talks directly to the underlying AFS Cache through extents retrieved from the AFS Service
  - Interesting edge cases arise when performing large file copies using small AFS Cache sizes
    - Windows ‘optimizations’ in flushing
  - Leverage Windows Read-Ahead and Write Behind features

File Data Handling



- Allows for better performance by allowing redirector direct access to cache file
- AFS Service still manages cache layout and population

PIOCTL Interface

- The interface has not changed from the AFS perspective
- Implemented within the redirector as ‘special’ file open requests within active directory
- File information and data management handled within the AFS Service

Special Share Name Handling

- \PIPE\IPC$
  - Used for remote processing – currently not supported within the AFS Redirector
- \PIPE\srvsvc
  - Used for server and share information processing through the Net API
    - Supported through AFS Service
    - Leverages Microsoft RPC engine for translation
- \PIPE\wkssvc
  - Used for workstation information processing through the Net API
    - Supported through AFS Service
    - Leverages Microsoft RPC engine for translation

Invalidation Processing

- Callback processing and issues in Windows
  - Callbacks can be made as a result of requests issued from the file system. Need to ensure these re-entrant calls do not lead to dead locks
    - ‘Almost’ lockless model in the callback routine through FID access layer
  - Server initiated callbacks have interesting effects, particularly in the directory change notification interface
    - Callbacks are FID based while notification is name based

Windows Change Notification

- Windows model for directory change notification
  - Objects added, modified or deleted initiate completion of a notification request
- Windows support API is named based … not in AFS
- Implement layer on top of Windows support API to map names to/from FIDs
  - Some edge cases that are not correctly handled, particularly in callback invalidation

AFS Redirector Trace System

- Command line configurable – Level, subsystem, buffer size, etc.
- Persisted configuration for system startup tracing
- In memory buffer so recoverable in crash dump
- Retrieve buffer through command line as well as dump to debugger

Yet to be Done …

- Alternate Data Streams (requires new RPCs)
- Extended Attributes (requires new RPCs)
- User and process quotas
- Enhanced extent processing interface
- Windows Management Instrumentation
- Dynamically loadable functional driver
  - eliminates reboot for updates to file system
- Microsoft Management Console replacement for AFS Control Panel

Contact Info

- Jeffrey Altman
- President
- Your File System Inc.
- jaltman@your-file-system.com
- +1 212 769-9018

YourFileSystem