NFS in Userspace: Goals and Challenges

Tai Horgan

EMC Isilon Storage Division

Introduction: OneFS

Clustered NAS File Server

- Single Volume
- Multiple Protocol Access
  - SMB 1/2/2.1
  - NFS 2/3/4
  - HTTP
  - FTP
  - Hadoop
- Hybrid NTFS/Posix semantics

NFS in Userspace

Why userspace?

1. Single Point of Access

2. Unified Lock Domains

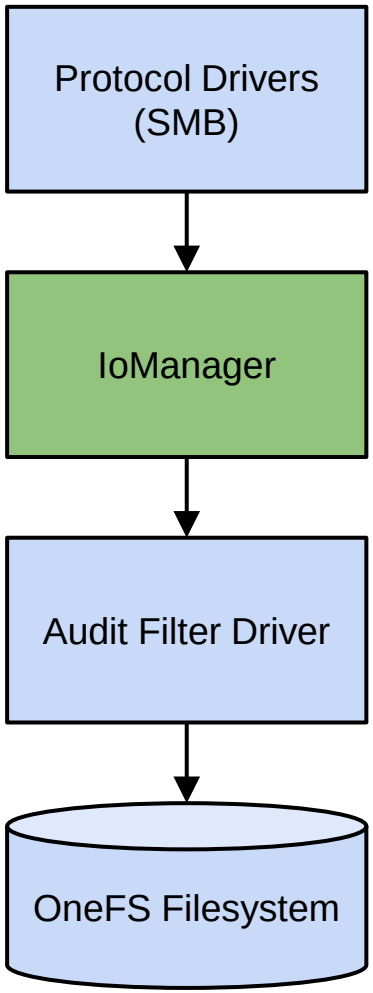
3. Data Protection

4. Lower Maintenance Cost

NFS in Userspace: Why?

Single Point of Entry to Filesystem

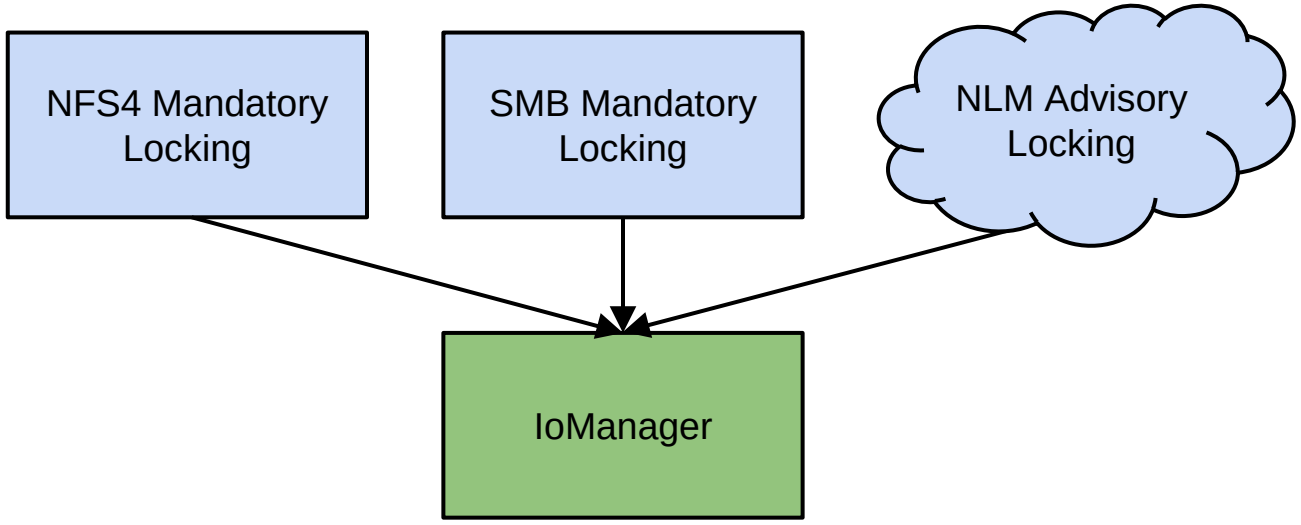
- SMB, NFS, FTP, etc
- Unified access auditing for all protocols
- Prevent unnecessary duplication of features
- Maintainable API into filesystem



NFS in Userspace: Why?

Unified Lock Domain

- SMB already in userspace
- Correctly contend with NFS4
- Best effort contention with NLM



NFS in Userspace: Why?

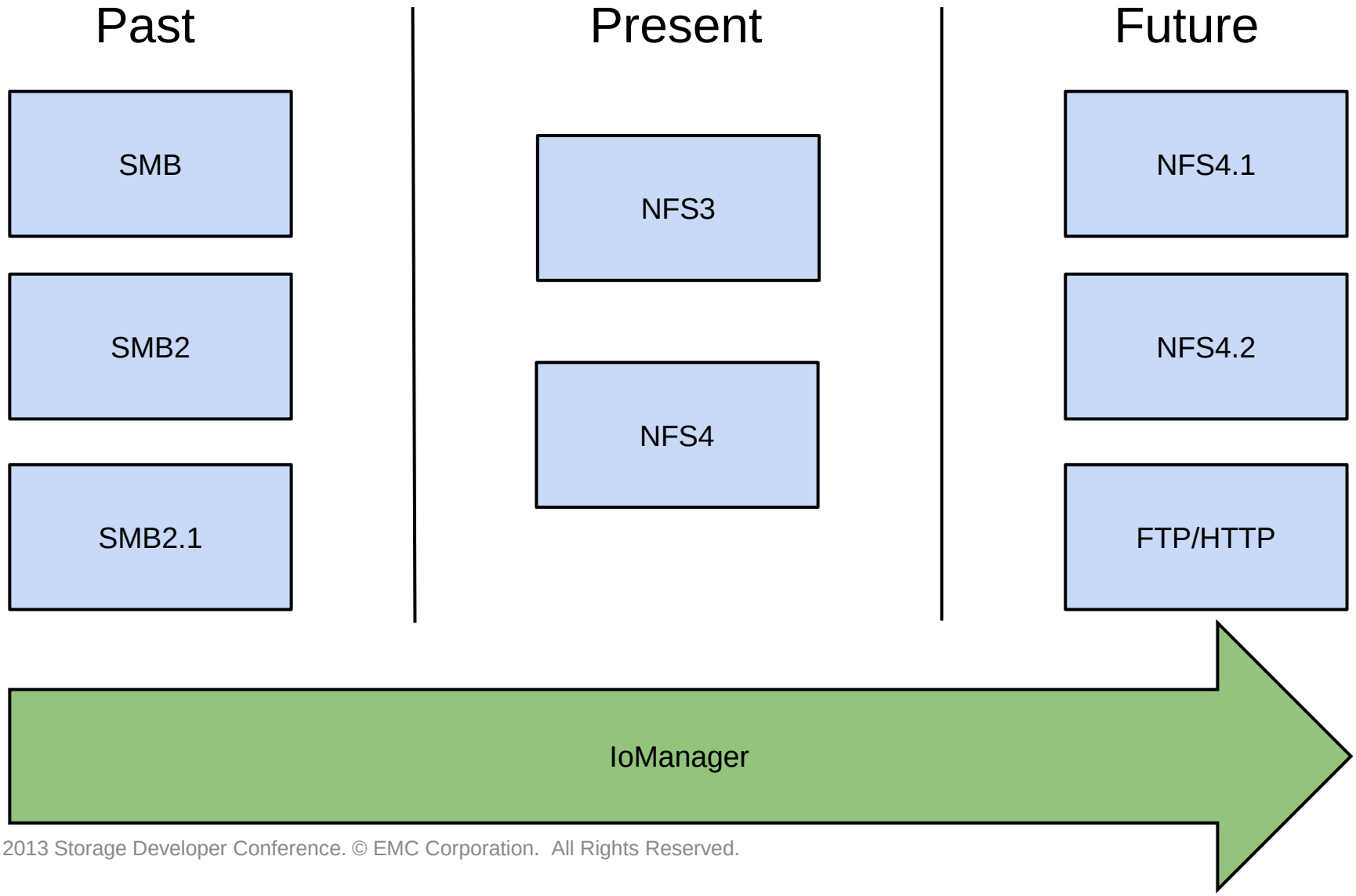
OneFS Clustered Filesystem

- Reboot implies potential loss of data protection
- Reboots are more than panics - userland is easily patched

IoManager:

- Virtualized I/O Framework
- Windows-style, open-based file semantics
- Developed by Likewise Software, now a part of EMC/Isilon

IoManager



NFS in Userspace

Challenges:

- Performance
- Lock Semantics
- IoManager Idiosyncrasies
- NFS Filehandle Compatibility

Performance

Usermode is Slow!

Three main mitigation strategies:

- Aggressive Caching
- Custom RPC with Zero-Copy
- Microthreading

Performance: Caching

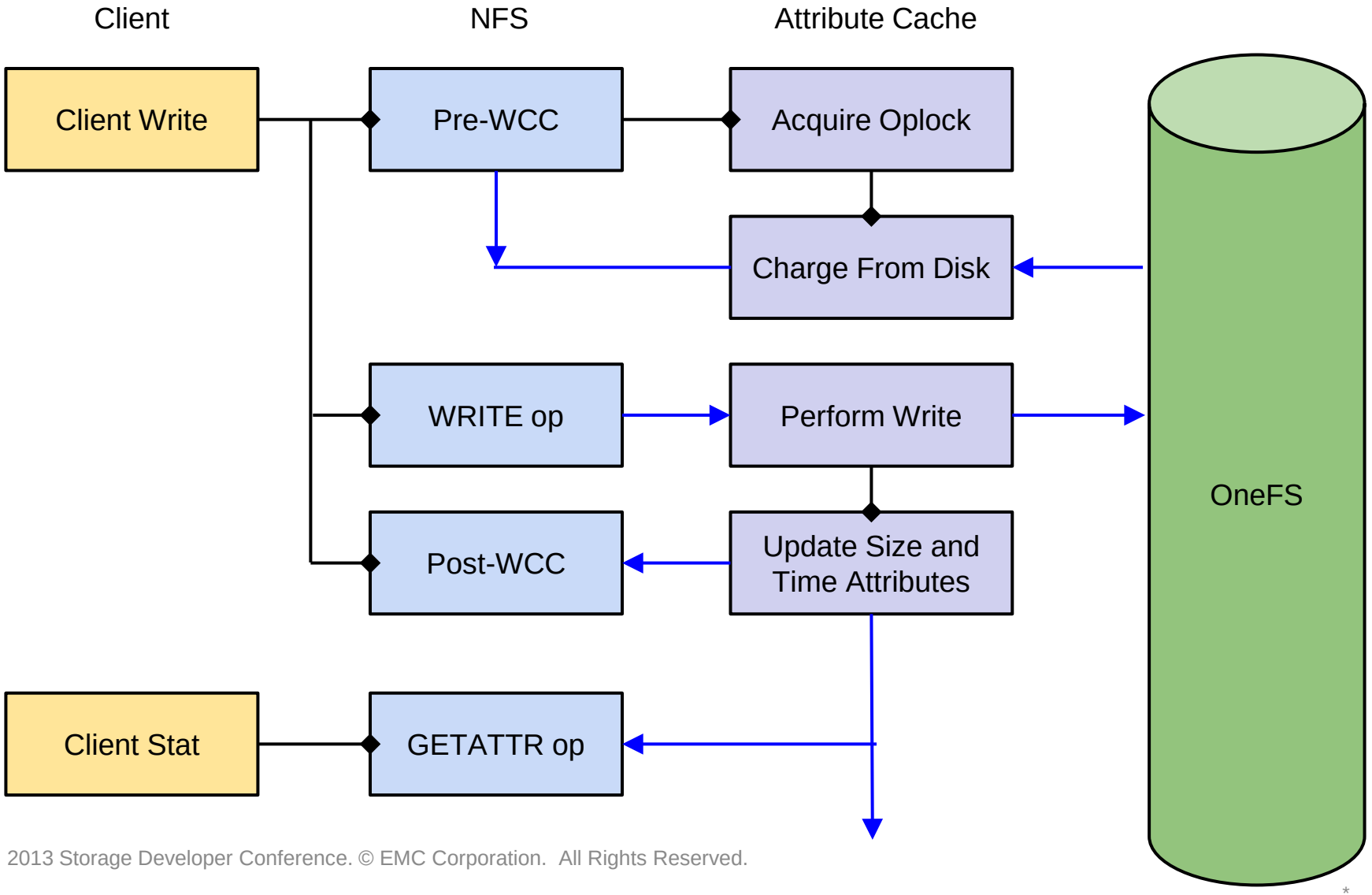
Client Cache invalidation:

- NFSv3: WCC data
- NFSv4: Delegations
- SMB: Oplocks/Leases

Multiprotocol Stack:

- Simulate WCC with delegation/oplock on behalf of client

Performance: Caching



Performance: RPC

Custom User-Mode RPC code

- Nonblocking
- Zero-Copy aware
  - Custom .x file with read and write buffers demarcated
  - Custom fields are replaced with socket pair in place of buffer
  - Fd pairs are passed in/out of kernel

Performance: Microthreading

IoManager lightweight thread model

- IoManager Fiber API
- 1-1 Threads to Cores ratio
- Nonblocking behind the scenes - “blocking” calls are fiber context switches and callbacks

Lock Semantics

IoManager

- Mandatory share mode locks
- SMB-style control path: ops begin with OPEN/CREATE

NFS

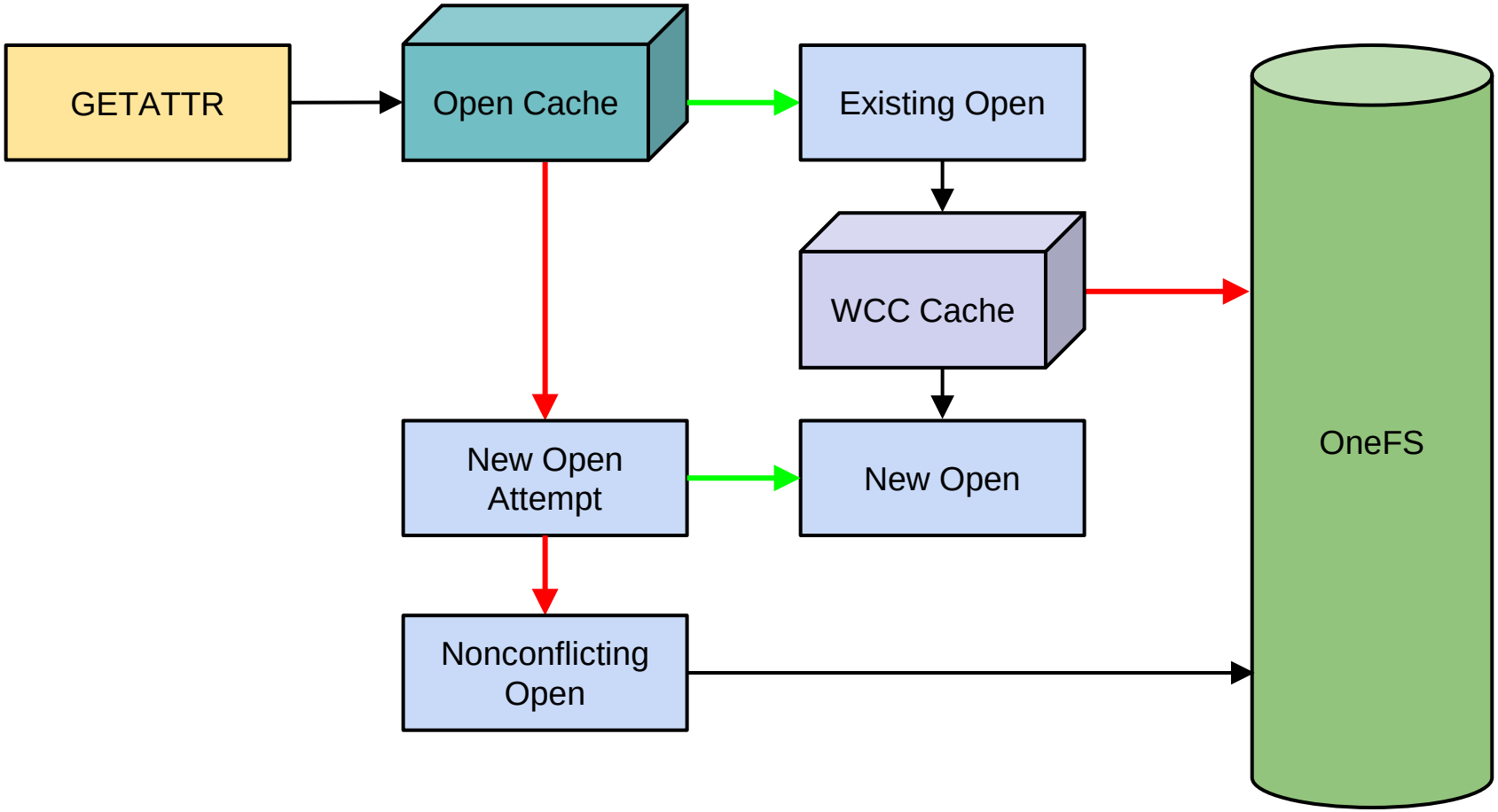
- Advisory locking
- Majority of ops are simple stats, which cannot fail for access violations

Lock Semantics

Solution: More caching

- Cache opens for recent files
- Cached opens contend on share mode locks
- Non-conflicting fallback for operations that cannot return access errors (GETATTR)

Lock Semantics



IoManager Special Cases

NFS REMOVE and delete-on-close

- Immediate stat should fail
- REMOVE of link should delete link, not target

Non-regular file types

Posix filenames

Filehandle Compatibility

NFS Model

- Filehandle - unique file ID
  - OneFS: Inode # / Snapshot ID / Export ID / FSID

IoManager

- Path based API
  - Full path
  - Parent open + relative path

Filehandle Compatibility

Hybrid model - Multiple combinations

- Filehandle only
  - GETATTR, READ, WRITE
- Filehandle + relative path
  - LOOKUP, NFSv4
- Full path
  - SMB, MOUNT
- Path + relative path
  - SMB

NFSv3 forced to suffer an extra OPEN for parent directory - offset by caching

NFS4-Specific Challenges

Generally much easier than v3

- Open semantics for READ and WRITE
- Mandatory locking
- NFS4 CurFH state provides natural parent open for relative paths

NFS4-Specific Challenges

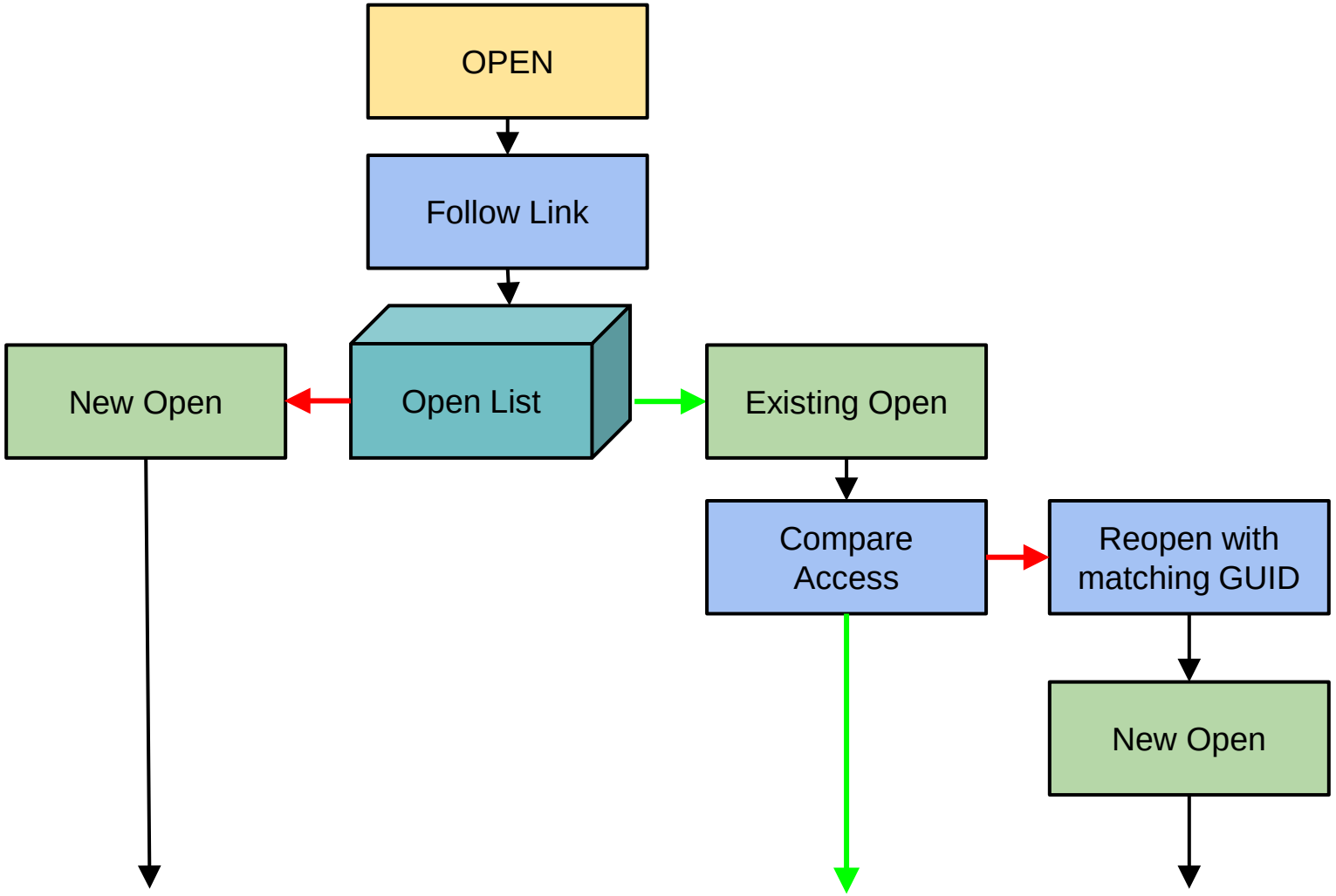
Atomicity of open upgrade and downgrade

- v4 open semantics require unioning of access on multiple opens of a single file
- Dropping and re-acquiring is racey
- Determining if we even have an open is difficult, due to hard links

IoManager

- No upgrade or downgrade, but provides a GUID key to atomically replace an open

NFS4-Specific Challenges



NFS4-Specific Challenges

Cluster Node Failover

- OneFS supports arbitrarily moving IP addresses from one cluster node to another
- IP Migration should be transparent to the client - service must not be affected
- Similar to NFSv4 Migration, though without the use of fs_location attribute

NFS4-Specific Challenges

Migration of NFSv4 State

- Cannot halt cluster locking to allow lock reclaim
- Client state must be transferred with address

Synchronize servers on per-IP basis

- Immigrating and emigrating servers must synchronize state before migration occurs
- No such scheme exists in OneFS… yet

NFS in Userspace

Goals:

- Performance parity
- Flesh out NFSv4 implementation (current server lacks delegation support)

Where we stand now:

- Performance tuning
- Ironing out edge cases

Plenty of work to do!

QA