NFS INDUSTRY CONFERENCE

NFS Version 4

Spencer Shepler

Sun Microsystems

spencer.shepler@sun.com

NFS INDUSTRY CONFERENCE

Today's Goals

- Describe the NFSv4 protocol
- Identify deployment considerations
- Discuss areas of potential future enhancement

NFS INDUSTRY CONFERENCE

NFSv4 Design Goals

- Improved access and good performance on the Internet
- Strong security with negotiation built into the protocol
- Better cross-platform interoperability
- Designed for protocol extensions

NFS INDUSTRY CONFERENCE

NFSv4 Protocol "Stack"

<table>
<tr><td colspan="2">NFSv4 (RFC3530)</td></tr>
<tr><td colspan="2">KerberosV5 (RFC1510)
SPKM-3
LIPKEY (RFC2847)</td></tr>
<tr><td>RPC (RFC1831)
XDR (RFC1832)</td><td>RPCSEC_GSS (RFC2203)</td></tr>
<tr><td colspan="2">TCP*</td></tr>
</table>

NFS INDUSTRY CONFERENCE

Many To One

PORTMAP	Port 111	→
MOUNT	Dynamic	→
NFSv2/v3	Port 2049	→
LOCK/NLM	Dynamic	→
STATUS	Dynamic	→
ACL*	Dynamic	→

NFSv4	Port 2049 TCP	→

NFS INDUSTRY CONFERENCE

Operation Count

- NFSv2 - 18 ops
- NFSv3 - 22 ops
- NFSv4 - 38 ops
- v2/v3 use “traditional” RPC
- v4 uses COMPOUND procedure to build sequence of operations

NFS INDUSTRY CONFERENCE

- NULL
- ***COMPOUND***

- ACCESS
- ***CLOSE***
- COMMIT
- CREATE
- ***DELEGPURGE***
- ***DELEGRETURN***
- GETATTR
- ***GETFH***
- LINK
- ***LOCK***
- ***LOCKT***
- ***LOCKU***
- LOOKUP
- ***LOOKUPP***
- ***NVERIFY***
- ***OPEN***
- ***OPENATTR***
- ***OPEN_CONFIRM***
- ***OPEN_DOWNGRADE***
- ***PUTFH***
- ***PUTPUBFH***
- ***PUTROOTFH***
- READ
- READDIR
- READLINK
- REMOVE
- RENAME
- ***RESTOREFH***
- ***SAVEFH***
- ***SECINFO***
- SETATTR
- ***SETCLIENTID***
- ***SETCLIENTID_CONFIRM***
- ***VERIFY***
- WRITE
- ***RELEASE_LOCKOWNER***

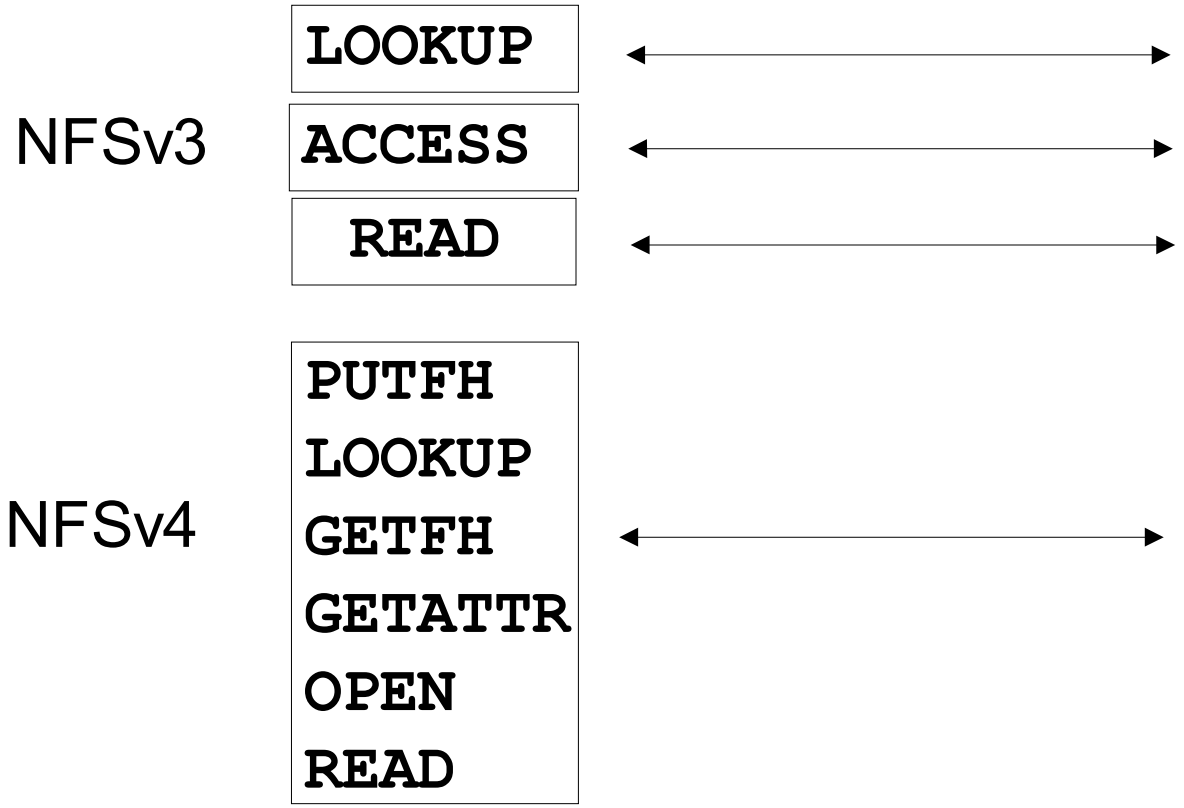
NFS INDUSTRY CONFERENCE

COMPOUND Procedure

- Group related operations in one RPC
- Evaluation stops at first error
- Reduce latency with fewer roundtrips
- Flexibility for various client environments

NFS INDUSTRY CONFERENCE

RPC vs. COMPOUND



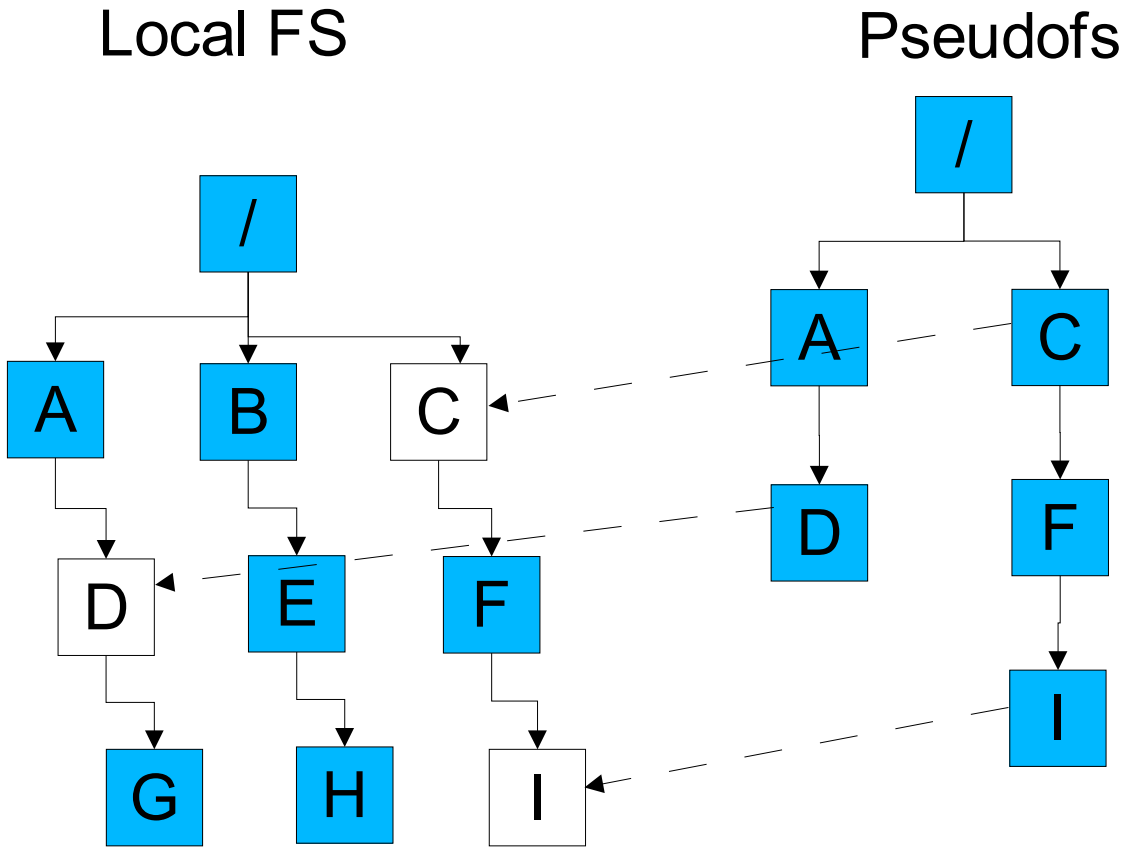
NFS INDUSTRY CONFERENCE

Namespace

- Replaces use of MOUNT protocol
- Server provides access to filesystems from a “root filehandle”
- Server *pseudofs* joins exported subtrees with a read only virtual file system
- Client traverses *pseudofs* with LOOKUP

NFS INDUSTRY CONFERENCE

Building Pseudofs



NFS INDUSTRY CONFERENCE

Client Namespace

- Client starts at root filehandle of server
- Client inspects *fsid* attribute to determine when new filesystem is found
- At each new filesystem, client automatically mounts filesystem into its namespace

NFS INDUSTRY CONFERENCE

Security

- RPCSEC_GSS framework is basis for various security mechanisms
- Provides for Authentication, Integrity, and Privacy
- Kerberos V5 and SPKM/LIPKEY

NFS INDUSTRY CONFERENCE

Security Negotiation

- Policy set at each filesystem
- Access root filehandle with secure channel
- If client mismatches on security mechanism, server returns an NFS4ERR_WRONGSEC error.
- Client will use SECINFO operation to enumerate available mechanisms

NFS INDUSTRY CONFERENCE

Filehandle Types

- Persistent filehandles same as NFSv2/v3
- Volatile filehandles are new
  - Filesystem types like FAT or user-level server implementations may use
  - Filehandle may become invalid and *expire*
  - Different than traditional ESTALE
  - Expiration at server restart
  - Increases implementation burden on client

NFS INDUSTRY CONFERENCE

Volatile Filehandles

- Upon expiration, client attempts recovery
- At initial LOOKUP, client saves path of file
- Pathname used at recovery; client traverses pathname to find new filehandle
- Other expiration events may include RENAME or migration of filesystem

NFS INDUSTRY CONFERENCE

NFSv4 is Stateful

NFS INDUSTRY CONFERENCE

NFSv4's State

- Hierarchy of NFSv4 state begins with association between a single client and server
- This hiearchy is important because the LEASE period and recovery represents all client-server state
- NFSv4 supports many definitions of client

NFS INDUSTRY CONFERENCE

Lease Management

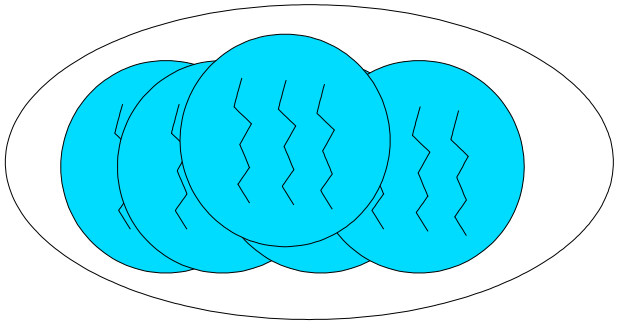
- Lease timeouts used to manage recovery
- Server determines lease period
- Period is for all state generated by client
- Lease renewal occurs at explicit RENEW or by any operation that uses *stateid*
  - CLOSE, DELEGRETURN, LOCK, LOCKU, OPEN, OPEN_CONFIRM, READ, RENEW, SETATTR, WRITE

NFS INDUSTRY CONFERENCE

Lease Management (con't)

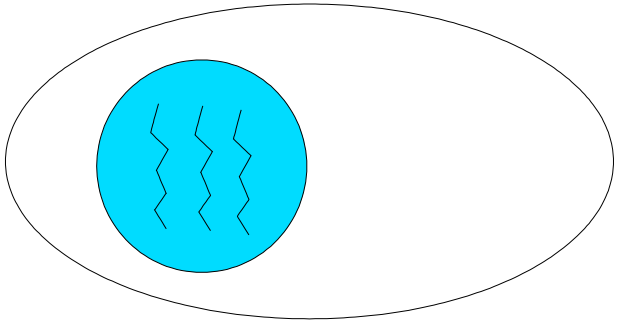
- Lease timeouts adds to complexity
- Client may track implicit lease renewal
- To ease implementation - use RENEW
- Server management of client state
  - Client state may be released at lease timeout
  - Must be released if lease expired and conflict with state (additional overhead)

CLIENTID – may represent these entities at the NFS client



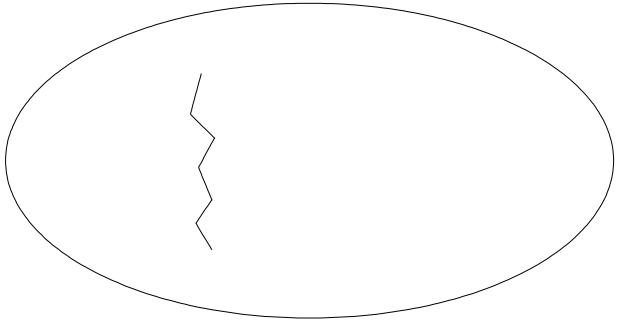
Traditional Unix NFS Client

(M-Processes / N-Threads)
(Y-Users/Principals)



User Level Client

(1-Processes / N-Threads)

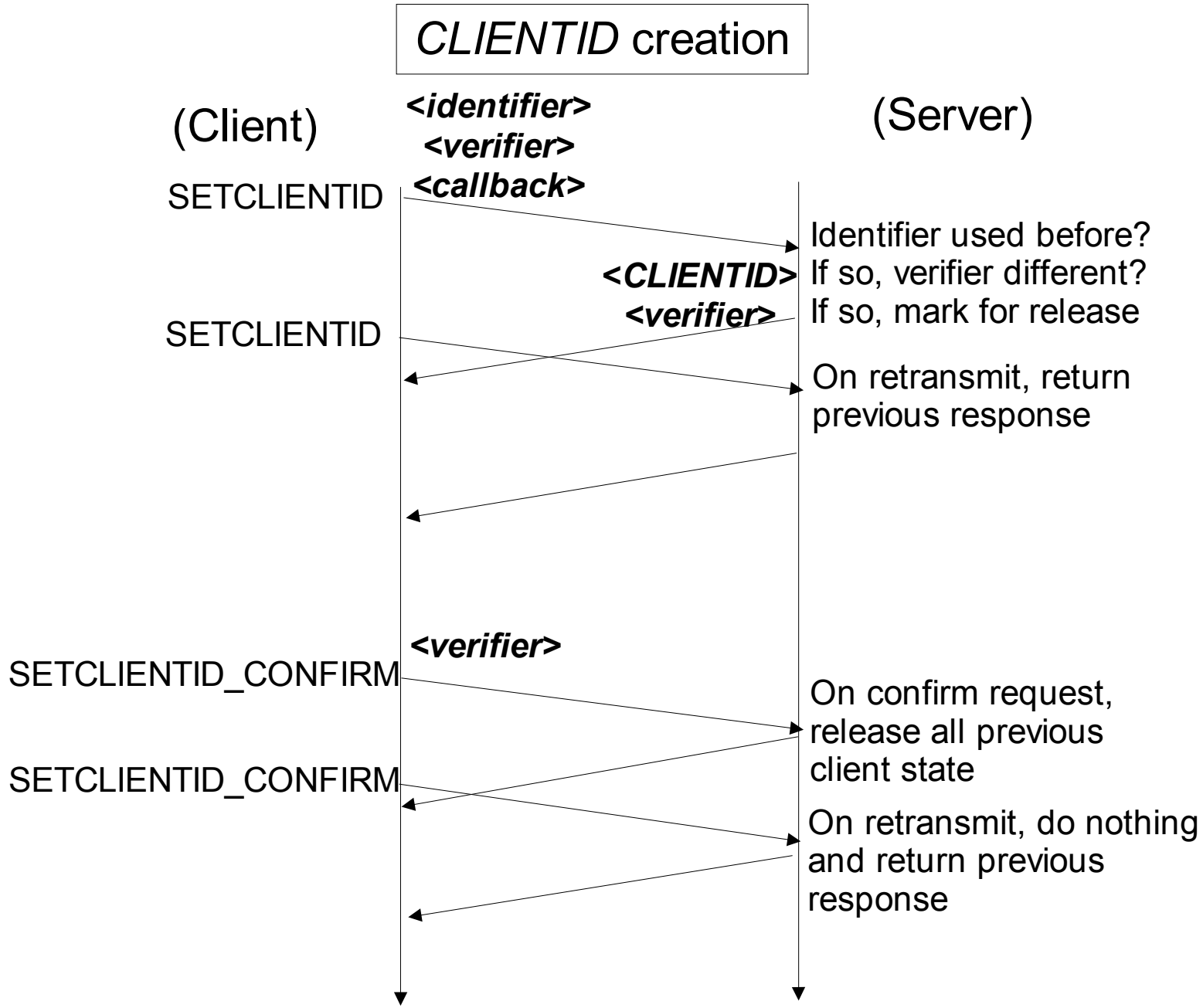


Individual Thread

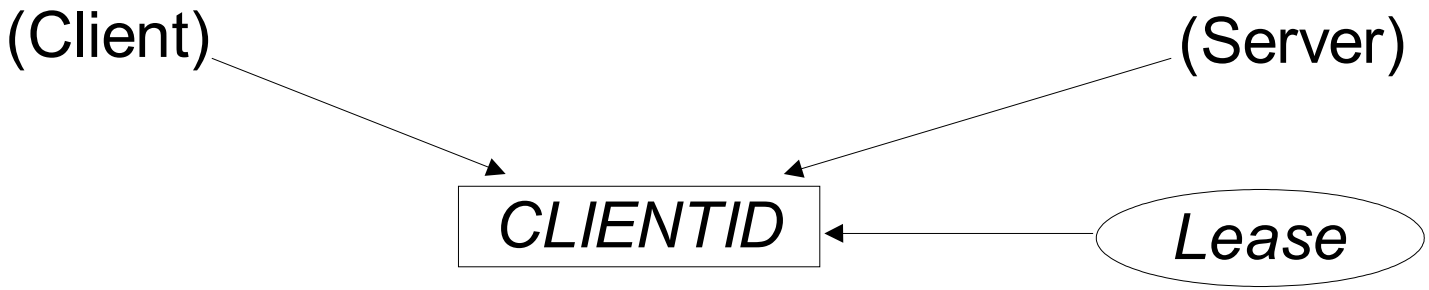
NFS INDUSTRY CONFERENCE

Creation of CLIENTID

- With SETCLIENTID, client chooses an opaque identifier and a verifier
- Uniquely identifies client and client instance
- Server assigns a *CLIENTID*
- Client confirms use with SETCLIENTID_CONFIRM
- Server uses RPC authentication to verify request (saves principal for future reference)


CLIENTID creation
(Client)
<identifier>
<verifier>
<callback>
(Server)
SETCLIENTID
Identifier used before?
If so, verifier different?
If so, mark for release
<CLIENTID>
<verifier>
SETCLIENTID
On retransmit, return previous response
<verifier>
SETCLIENTID_CONFIRM
On confirm request, release all previous client state
SETCLIENTID_CONFIRM
On retransmit, do nothing and return previous response

What do we have?



NFS INDUSTRY CONFERENCE

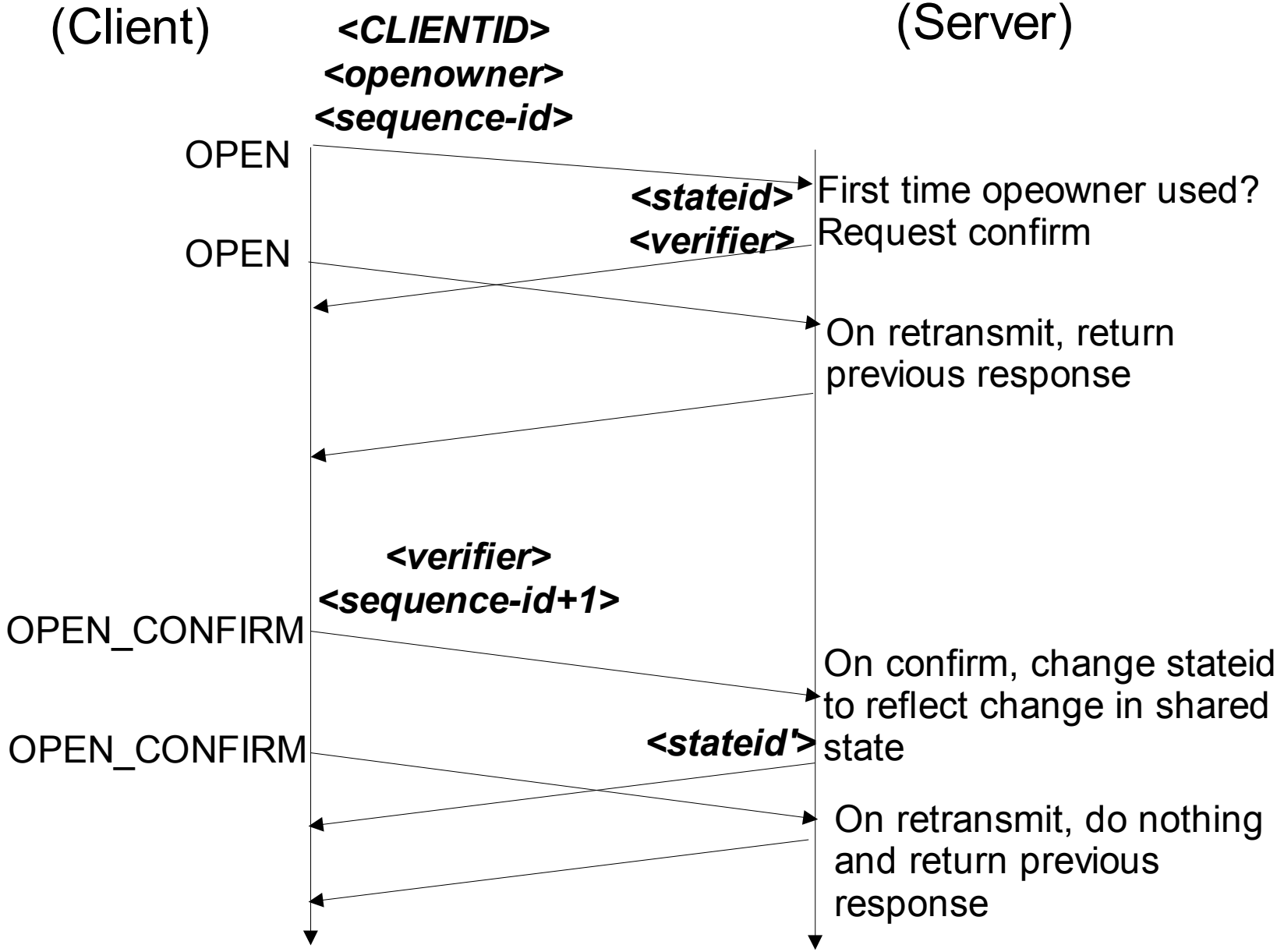
OPEN and its state

- Combines regular file CREATE, LOOKUP, and share reservations
- Requires name of regular file object
- Identifies owner for OPEN and LOCK state
- Server may provide delegation* in response

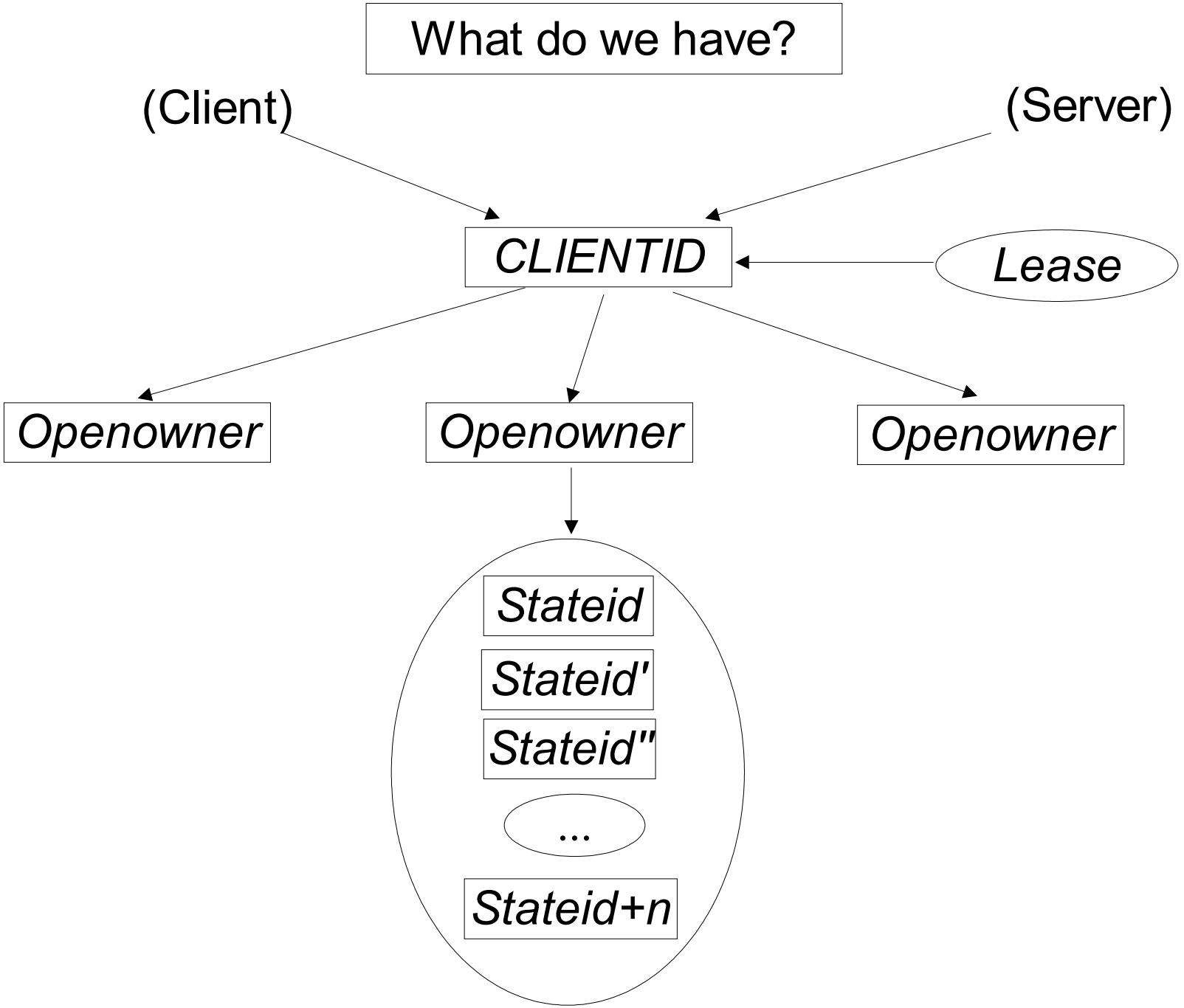
NFS INDUSTRY CONFERENCE

Openowner to openstateid

(Client) (Server)



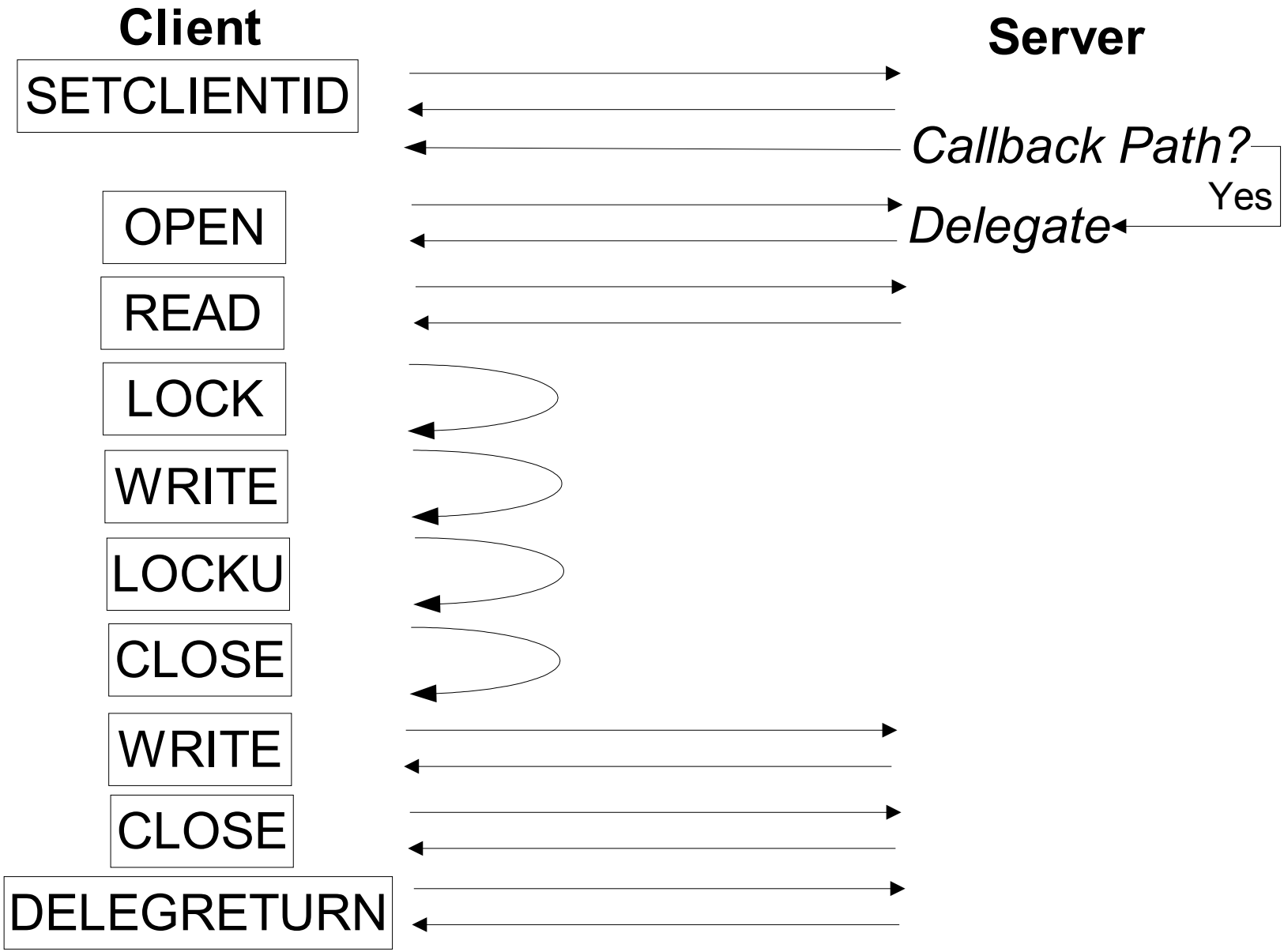
NFS INDUSTRY CONFERENCE



NFS INDUSTRY CONFERENCE

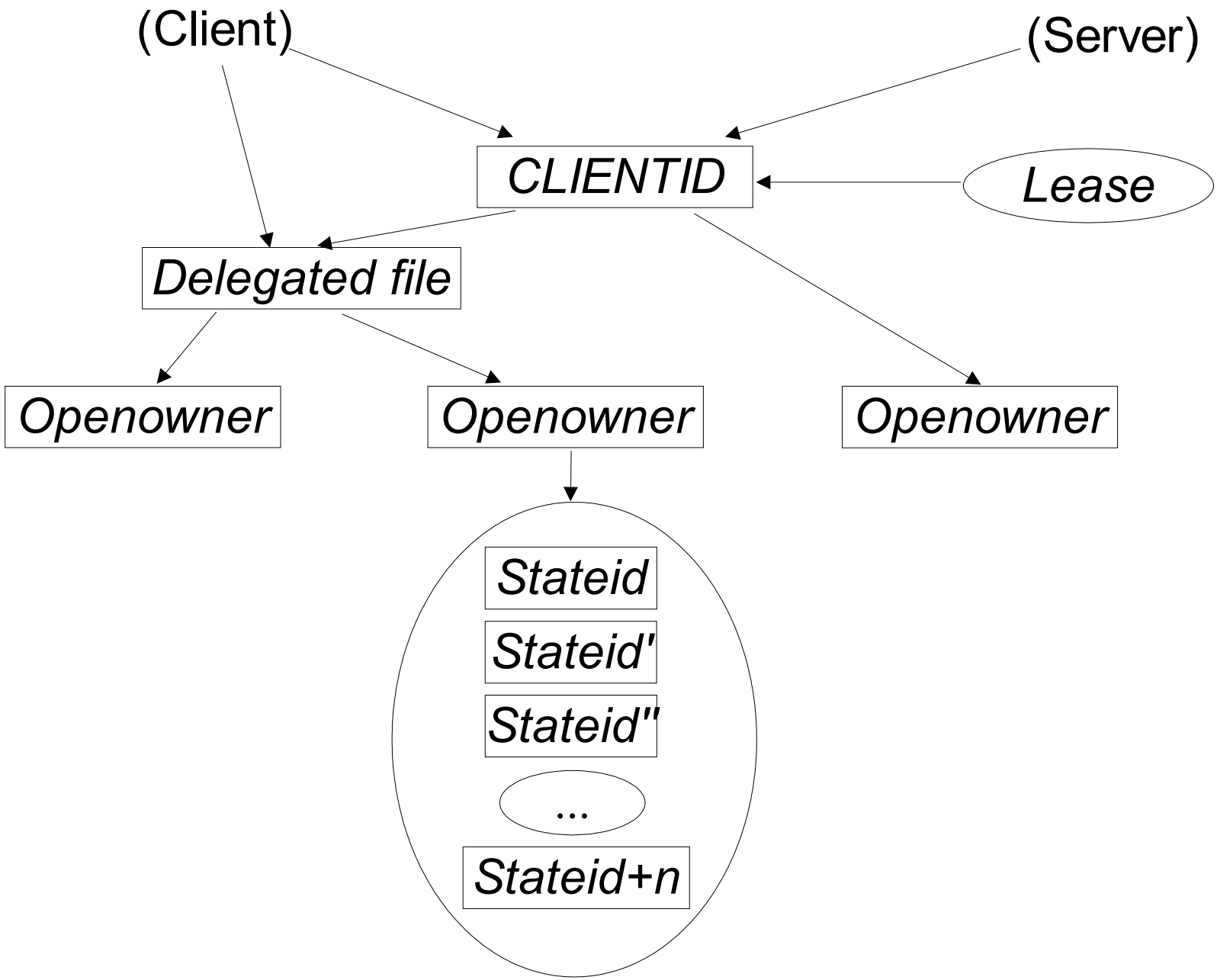
Delegation

- Intended for minimal sharing environments
- Server decides when to provide delegation
- Not required for correct protocol operation
- Callback path to client must exist
- If delegation provided, client does not need to contact server for further OPEN, LOCK, READ, WRITE, CLOSE


Client
Server
SETCLIENTID
Callback Path?
Yes
Delegate
OPEN
READ
LOCK
WRITE
LOCKU
CLOSE
WRITE
CLOSE
DELEGRETURN

NFS INDUSTRY CONFERENCE

What do we have?

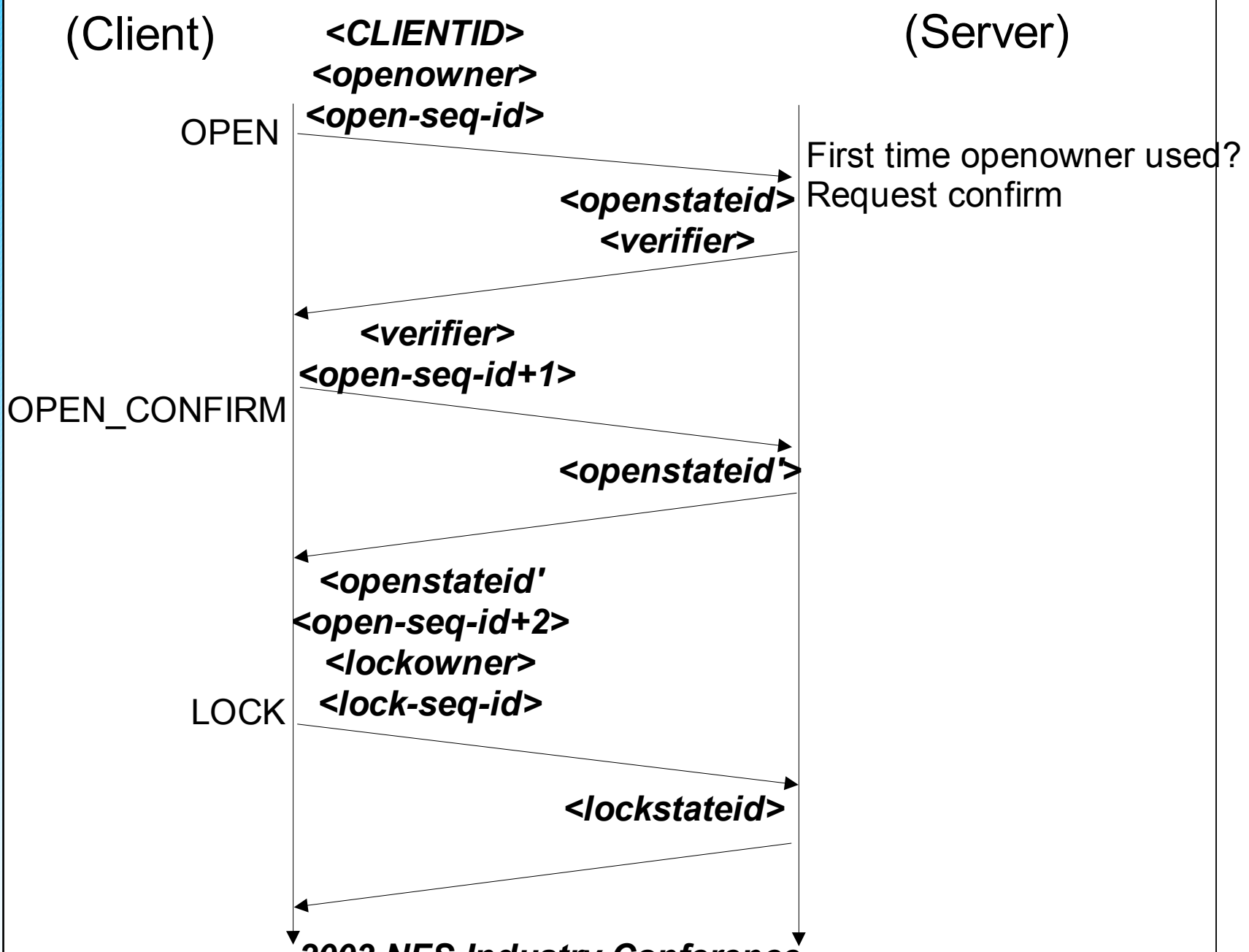


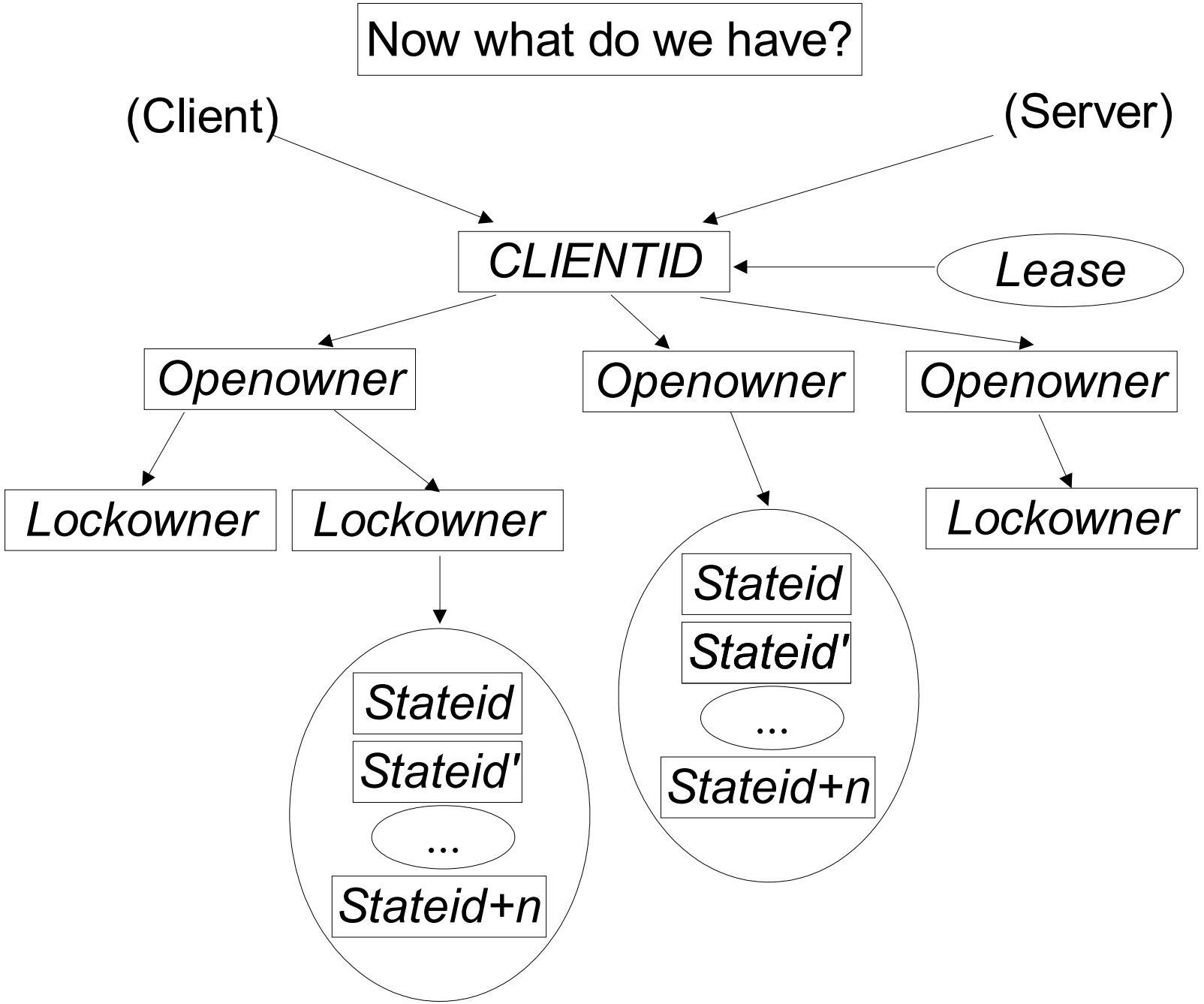
NFS INDUSTRY CONFERENCE

File Locking

- Better implementation a result of integration
- Byte-range locking - non-blocking, mandatory
- Callbacks not used*
- *Stateid* represents transitions of locking state
- *Sequence-id* preserves request ordering
- Locks recovered at lease expiration*
- Client does lock recovery at server restart

Openowner / Lockowner association




Now what do we have?
(Client)
(Server)
CLIENTID
Lease
Openowner
Openowner
Openowner
Lockowner
Lockowner
Lockowner
Stateid
Stateid'
...
Stateid+n
Stateid
Stateid'
...
Stateid+n

NFS INDUSTRY CONFERENCE

Attributes

- Mandatory, Recommended, and Named
- 52 mandatory/recommended attrs
- Extends beyond traditional Unix attrs
- Protocol encoding adds to overhead
- Encoding allows for extensibility
- Mandatory: type, expiration type, change, size, link?, symlink?, fsid, lease duration

NFS INDUSTRY CONFERENCE

Named or Extended Attrs

- Support is optional
- OPENATTR operation returns directory filehandle of named attributes
- Named attrs may be recursive
- Intended for application use
- Semantics are opaque to client and server

NFS INDUSTRY CONFERENCE

Access Control Lists

- Definition finally in the protocol
- Solves today's problem of 4+ ACL protocols
- ACLs may be manipulated by client
- Windows/NT ACL compatible
- Combined with RPC security provides for strong security mechanism
- POSIX stops at first denial; NT does full evaluation searching for allowance

NFS INDUSTRY CONFERENCE

Owner / Group Attributes

- String based identifiers
- Take place of numeric uid / gid
- Identier looks like: user@domain
- External mapping outside of protocol is not defined

NFS INDUSTRY CONFERENCE

Filesystem Replication

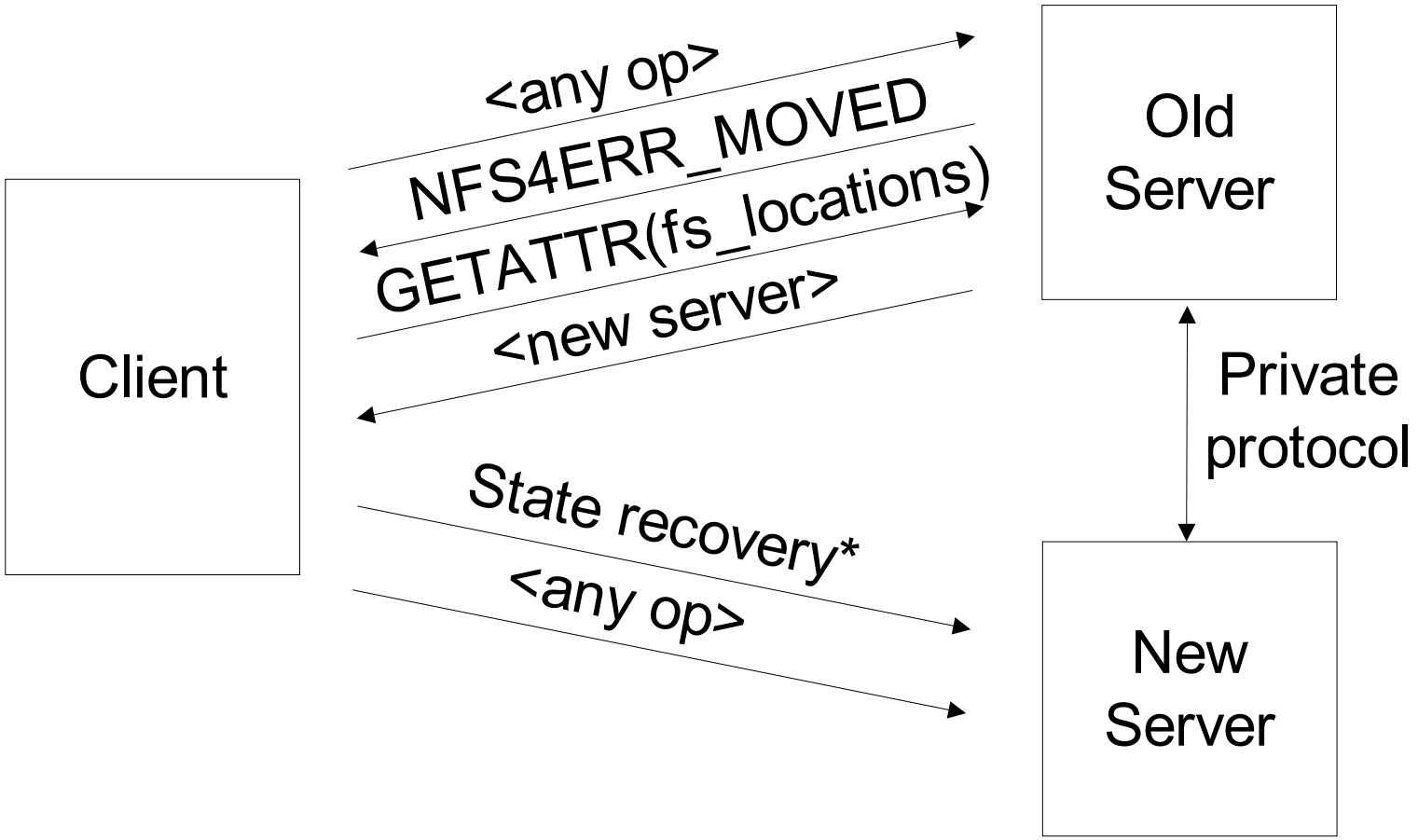
- Intended for read-only filesystems
- Increases availability
- Client's policy directs when to switch to another replica
- *fs_locations* attribute enumerates replicas
- Client needs to reconstruct state at new server
- Server to server replication undefined

NFS INDUSTRY CONFERENCE

Filesystem Migration

- Enables load balancing or server reoganization
- Server to server transfer is undefined
- Client receives NFS4ERR_MOVED at migration event
- *fs_locations* attribute provides new location

Migration Event



NFS INDUSTRY CONFERENCE

Minor Versioning

- Difficult to revise previous versions (v2->v3)
- Realize that protocol is not perfect
- Protocol must be allowed to evolve
- Changes allowed in minor version
  - Operations (new argument types) added
  - Add new Attributes

NFS INDUSTRY CONFERENCE

Minor Version Examples

- SECINFO fixes
- Delegations for Directories
- Support for RDMA/RDDP

NFS INDUSTRY CONFERENCE

NFS Version 4

Spencer Shepler

Sun Microsystems

spencer.shepler@sun.com