How I learned to love Sharing Violations

Richard Sharpe

Agenda

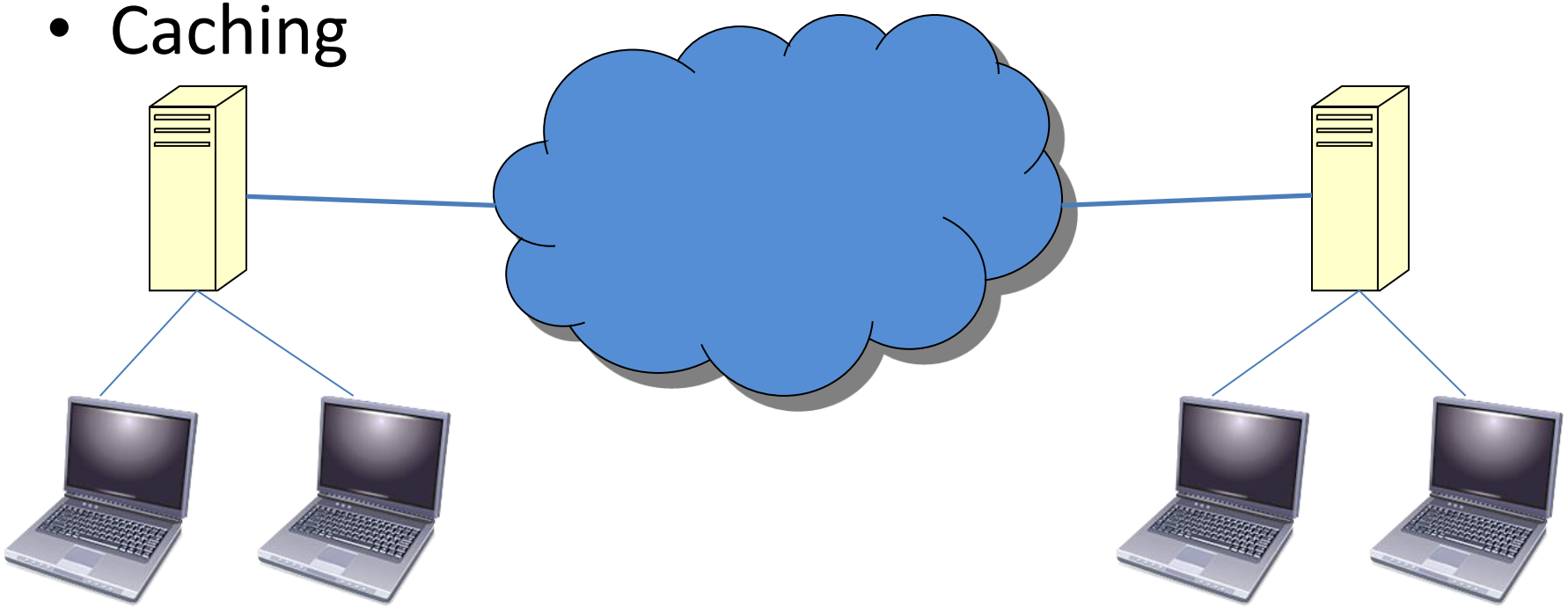
- What is the craziest way to use Samba?
- How Panzura uses Samba
- Some of the problems that have arrisen
- What changes we are making
- What the future holds

What is the craziest way to use Samba?

- I’m serious!
- Bending and twisting NTFS functionality for cloud access

How Panzura uses Samba

- Each Samba instance separate
  - Shared nothing
- Store data & metadata in the cloud
- Caching



Panzura & Samba

- Cloud Controllers (CCs)
  - A NAS
    - FreeBSD, ZFS and Samba
  - NFS & CIFS
  - Stores data & metadata in the cloud (encrypted)
  - Supports many cloud back ends
    - S3, Google, Atmos, Some Openstack, Azure coming up
    - Cloud mirroring
  - Data and metadata sent to the cloud
  - Caches data and metadata
- Samba 3.6.12+
  - Pulled in many post-3.6.6 patches
  - Have to move to 4.x soon

Panzura and Samba, cont

- Some customers have a large number of CCs
- Widely separated geographically
  - Latency sometimes 200mS
- Some interesting domain environments
  - Some have simple forests
  - Some have RODCs
  - Some have resource domains
  - Some have lots of domains

Panzura and Samba, cont

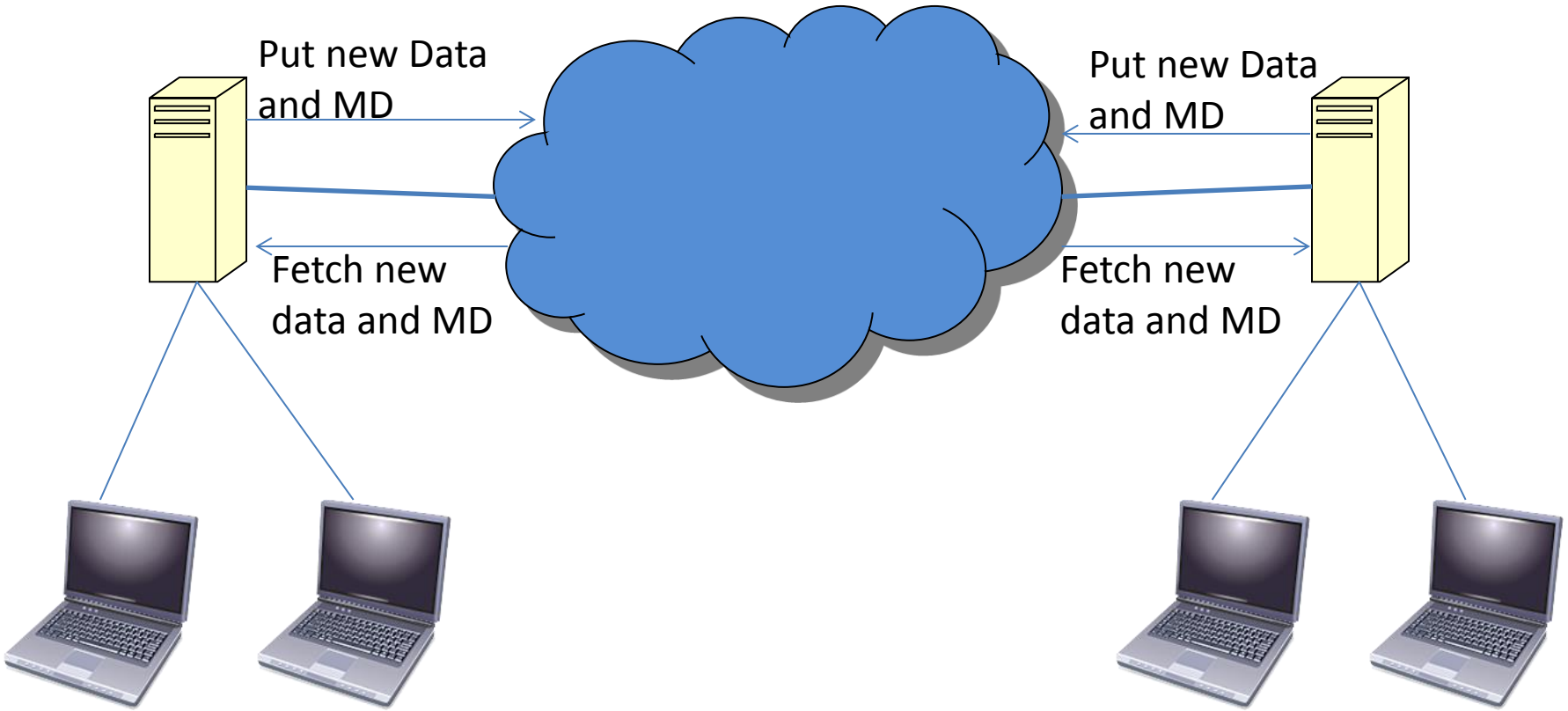
- Found and fixed many interesting problems
  - Joining a domain when local DC is RODC
  - Windbind consuming 100% CPU
  - SMB Signing and Compound Requests
- Developed useful tool for ACLs/SDs
  - smbxcacls
    - Directly reads the XATTR, not need for smbd
  - Would like to extend it to NFSv4 acls

How Panzura uses Samba

- Each Samba instance separate
- Checks with other Samba instances for
  - Share-mode locks
  - Delete-on-close
- Went through an evolution
  - First it was simple
  - Then less simple
  - Then …

Data, Metadata to the cloud

- Snapshots sent to the cloud
  - Data first, then metadata



Data, MD to the cloud

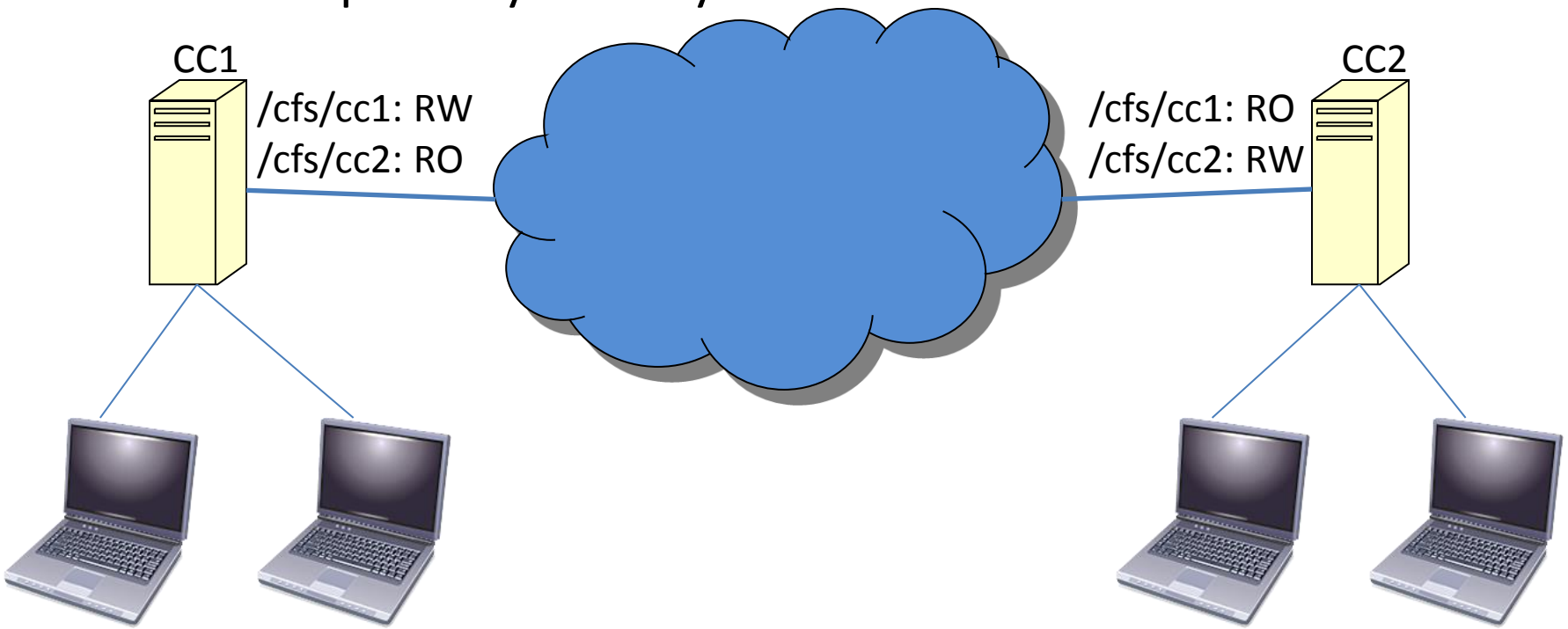
- Data and metadata objects
  - Either after certain amount of data
  - Or max time, like 60 seconds
- Data first (the default)
  - When another CC sees the metadata, can also see the data
- Delay until remote nodes see new/changed files
  - Has interesting consequences

Data, MD to the cloud

- Always fetch new Metadata
- Lazily fetch data
  - Data cached on each CC
  - Can specify pinning rules so file data not evicted

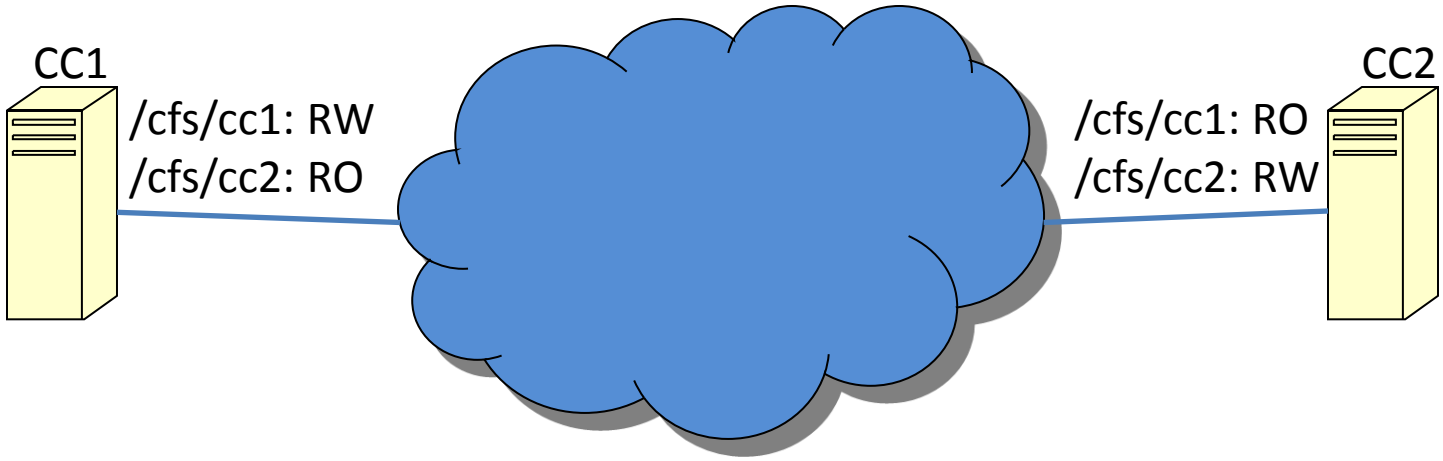
What the file system looks like

- Each CC sees all file systems
  - All but local are RO
  - Delay in seeing remote
    - Snapshot sync delay



Customers wanted more

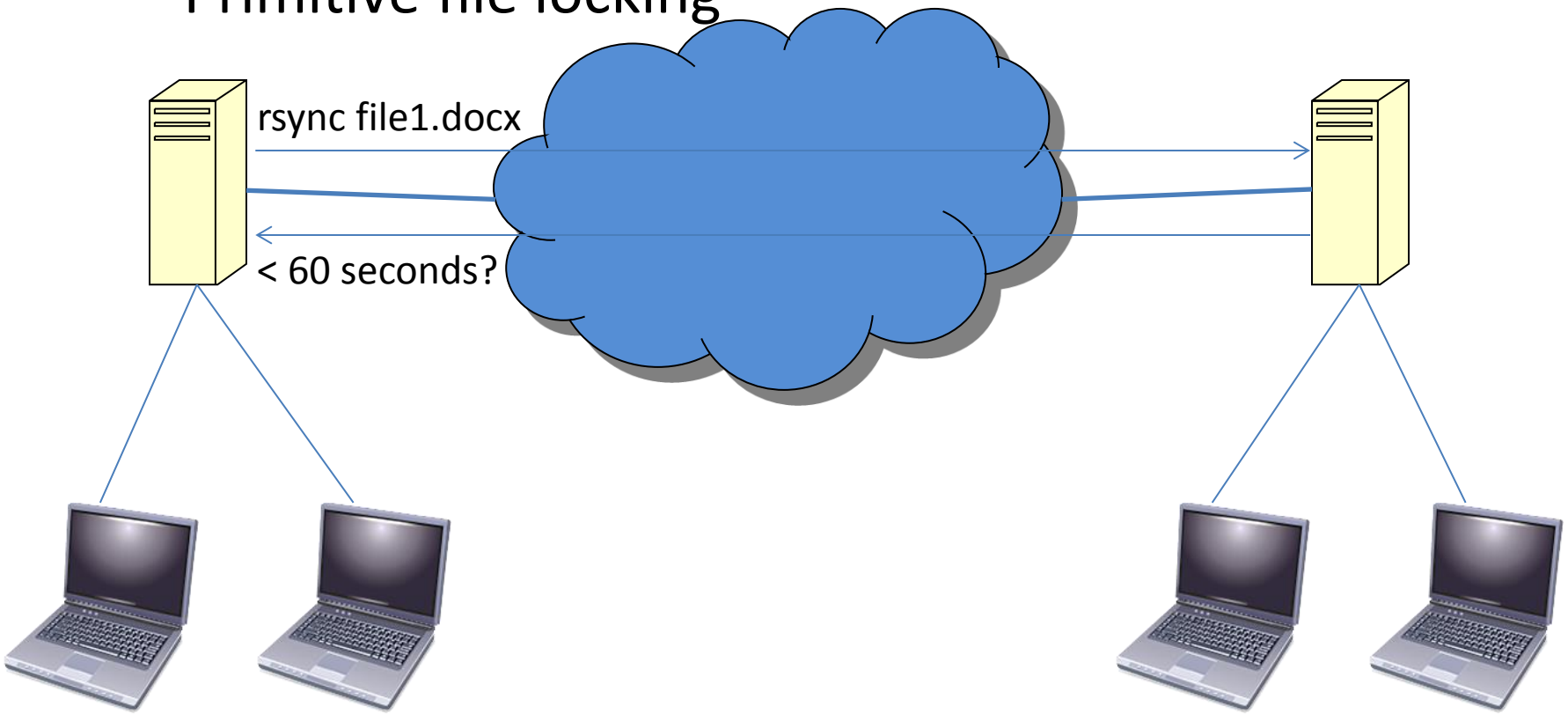
- The initial functionality was OK in some cases
  - Customers wanted tighter coupling between CCs
    - Between file systems
  - Same shares on each CC



//cc1/share1 -> /cfs/cc1 everywhere

Ownership of files

- At Create, transfer ownership
  - If a CC is not the owner
  - Primitive file locking



Ownership of files

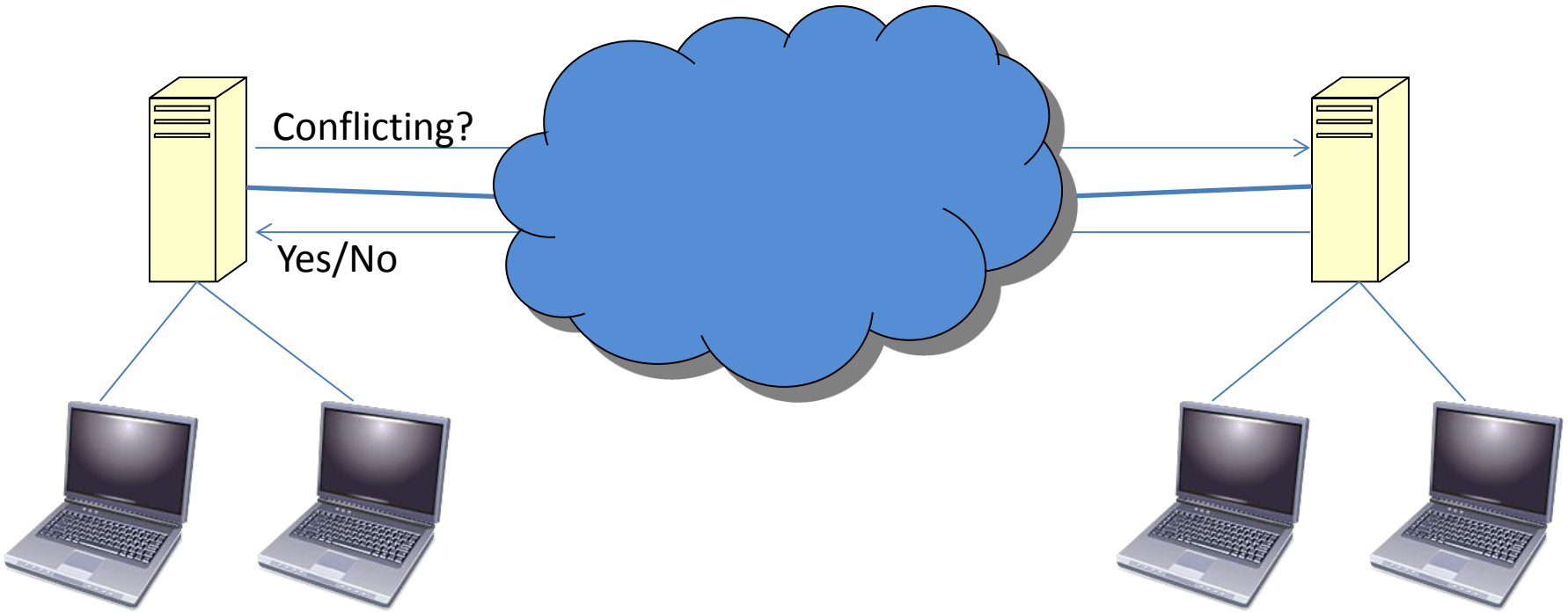
- Transfer of ownership specifies who gets RW access
- All others get SHARING VIOLATION

Allow RO access on non-owner

- Customers wanted more
- First one to ask for RW access gets ownership
- Non-owners can open RO
- Rsync on open for read
  - Large files?

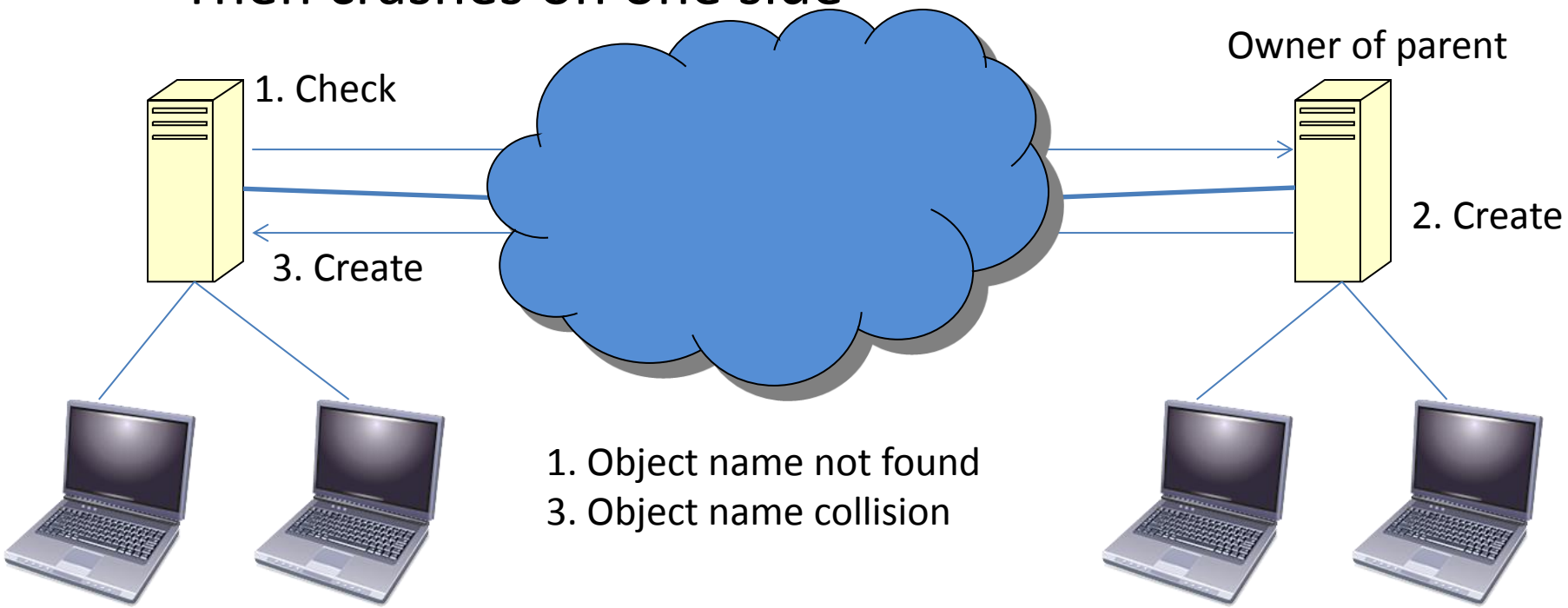
Share-mode lock exchange

- Check remote for sharing mode conflicts
  - On Opens
  - Office Apps need this



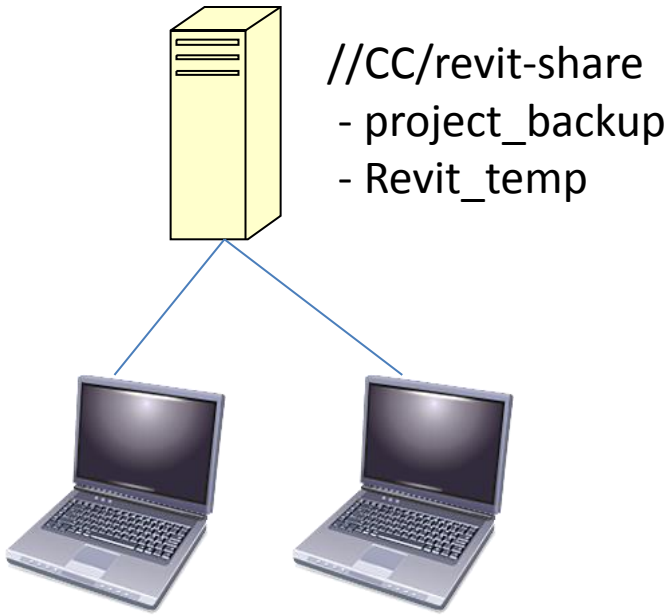
SHARING VIOLATION is good?

- Two clients creating New folder in a share
  - Explorer first check if folder exists
  - Then does Create
  - Then crashes on one side



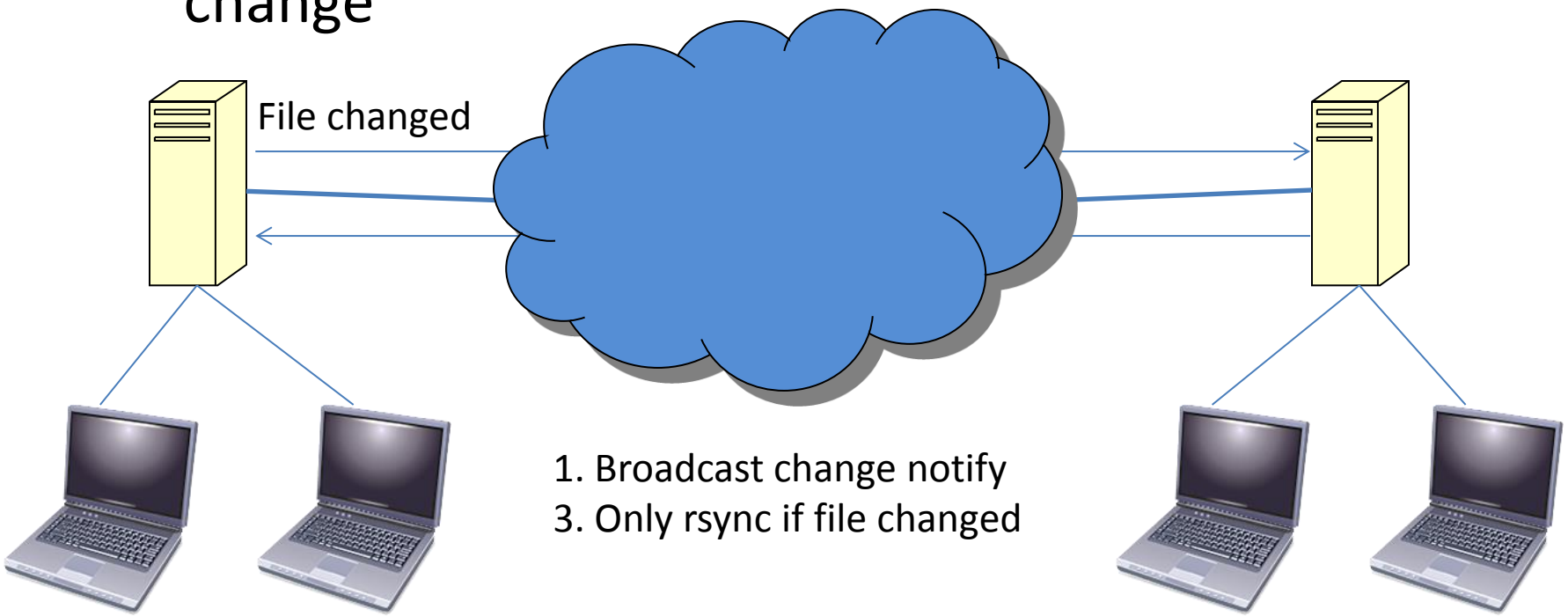
Revit: A file sharing app

- Has a collaborative mode
- Involves much access of shared files
- Heavy use of
  - sharing modes
  - Oplocks
  - Byte range locks



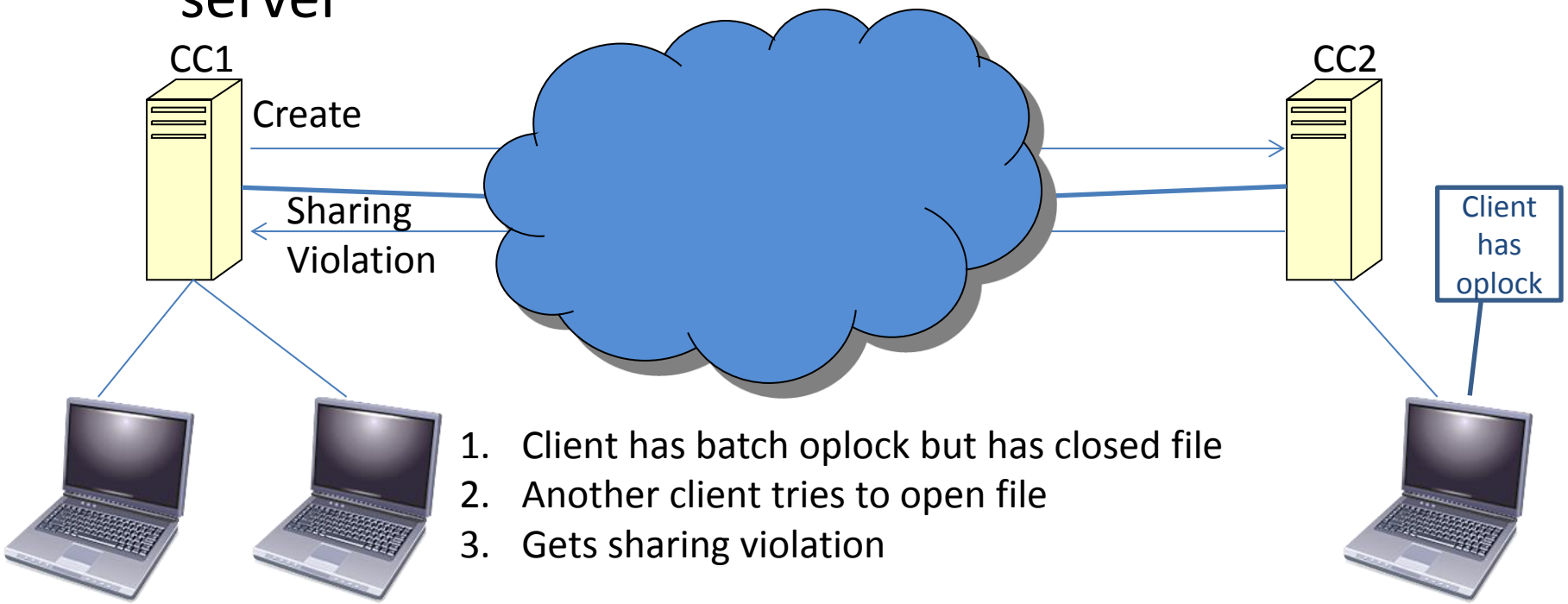
Distributed Change Notify

- When doing Open for Read
  - Rsyncing all the time is expensive
  - Use distribute change notify to tell others about change



Distributed OpLock Break

- Oplocks can cause failure to open when it should not
  - Consequence of not having a single metadata server



1. Client has batch oplock but has closed file
2. Another client tries to open file
3. Gets sharing violation

Where to from here

- Centralized vs distributed metadata
- How to make progress

Avoid Races

- Provide a Rename2 Win32 call and SMB request
  - Provide atomic name swap of two files
  - Office and other apps can use it for Save
  - Linux seems to provide such an API
- Don’t use racy Create approaches
  - Do not check if the name exists before creating
  - Just try Create with appropriate disposition

Extend Create

- Allow Apps to signal consistency requirements
  - Eventual consistency
  - Sequential consistency
  - Strong consistency (read after write)
  - Period after which files must be consistent

Extend UNIX Open?

- Sharing modes are very useful
  - Allow apps to signal what sort of sharing they can tolerate
  - Sharing violation suggests come back later