



OpenSolaris pNFS Server

Piyush Shivam

piyush.shivam@sun.com

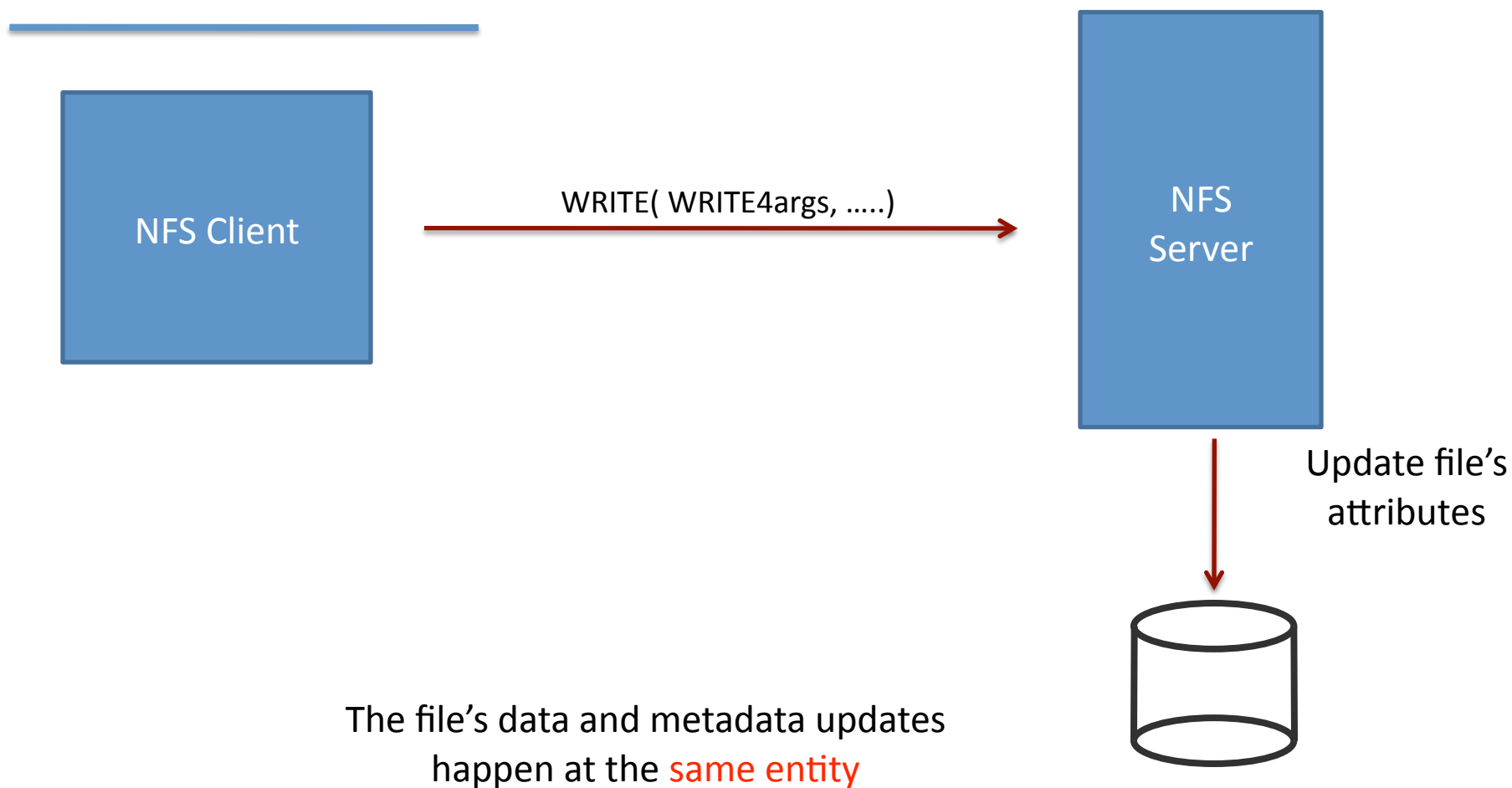
Outline

- Problem of the “state divide” in pNFS
- Updating the file’s mtime in the context of write delegation
- Updating a file’s size

Example: Writes in NFS v4

Application

`write(fd, buf, size)`

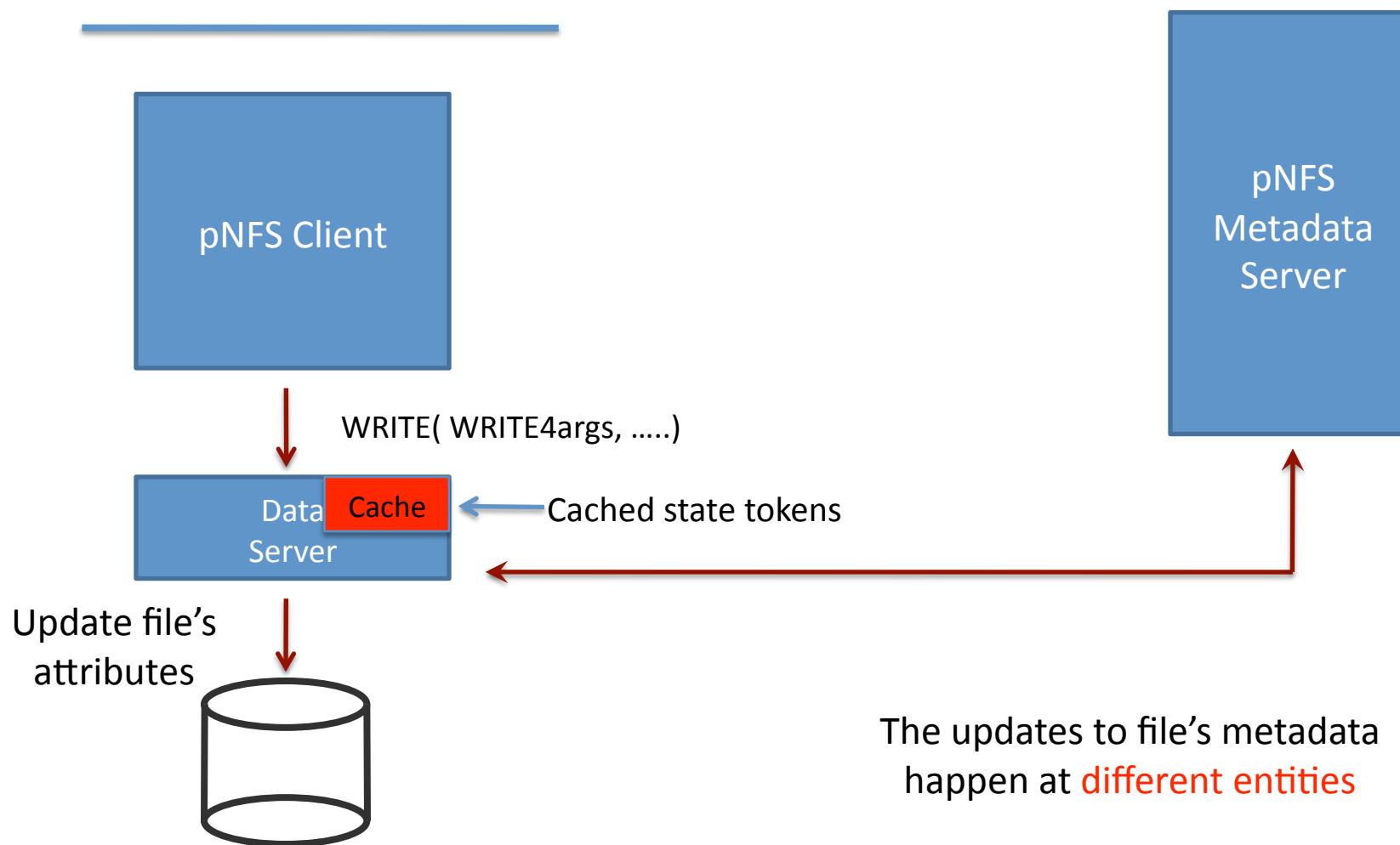


The file's data and metadata updates happen at the **same entity**

Writes in pNFS

Application

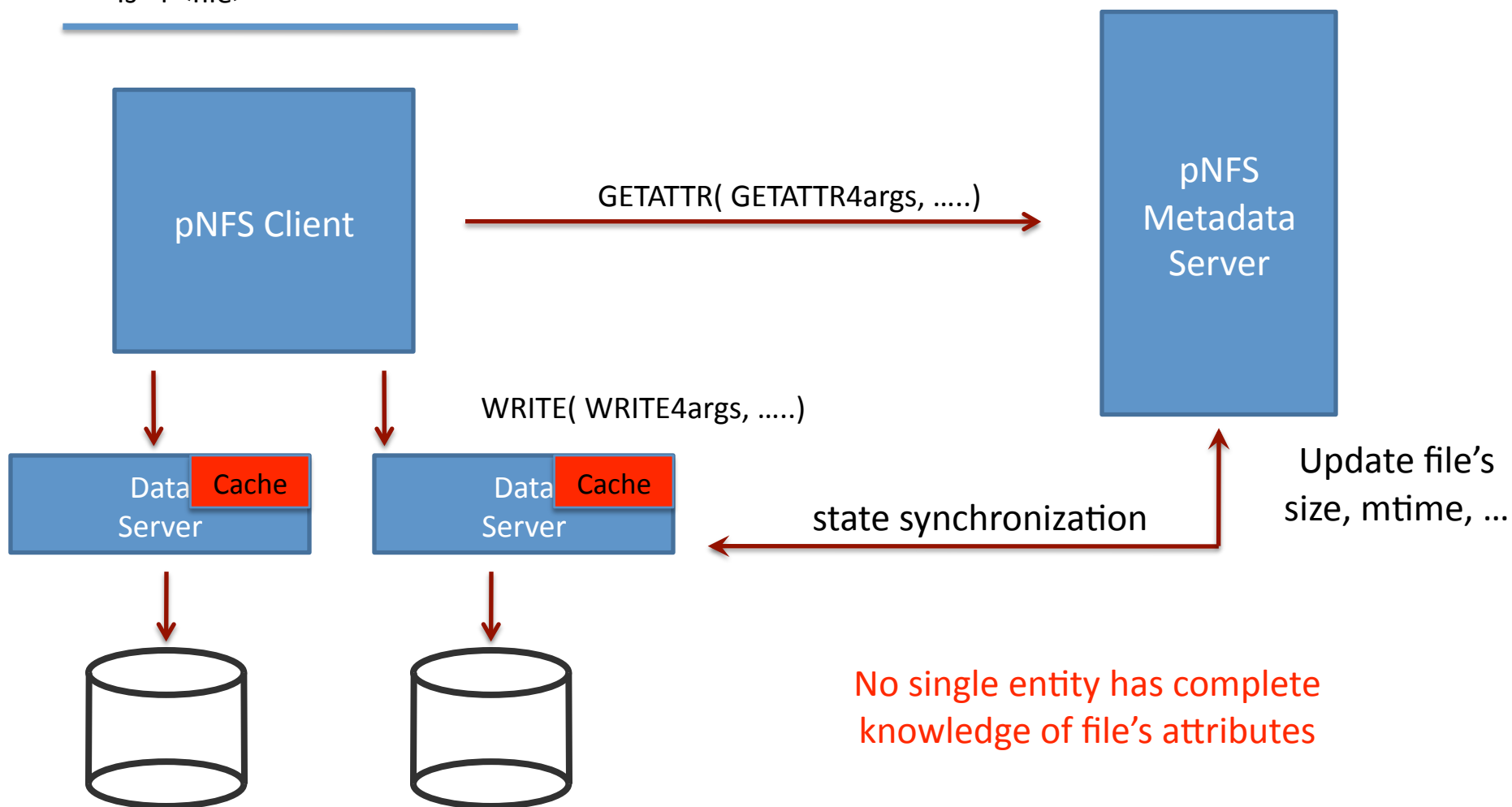
write(fd, buf, size)



Writes in pNFS

Application

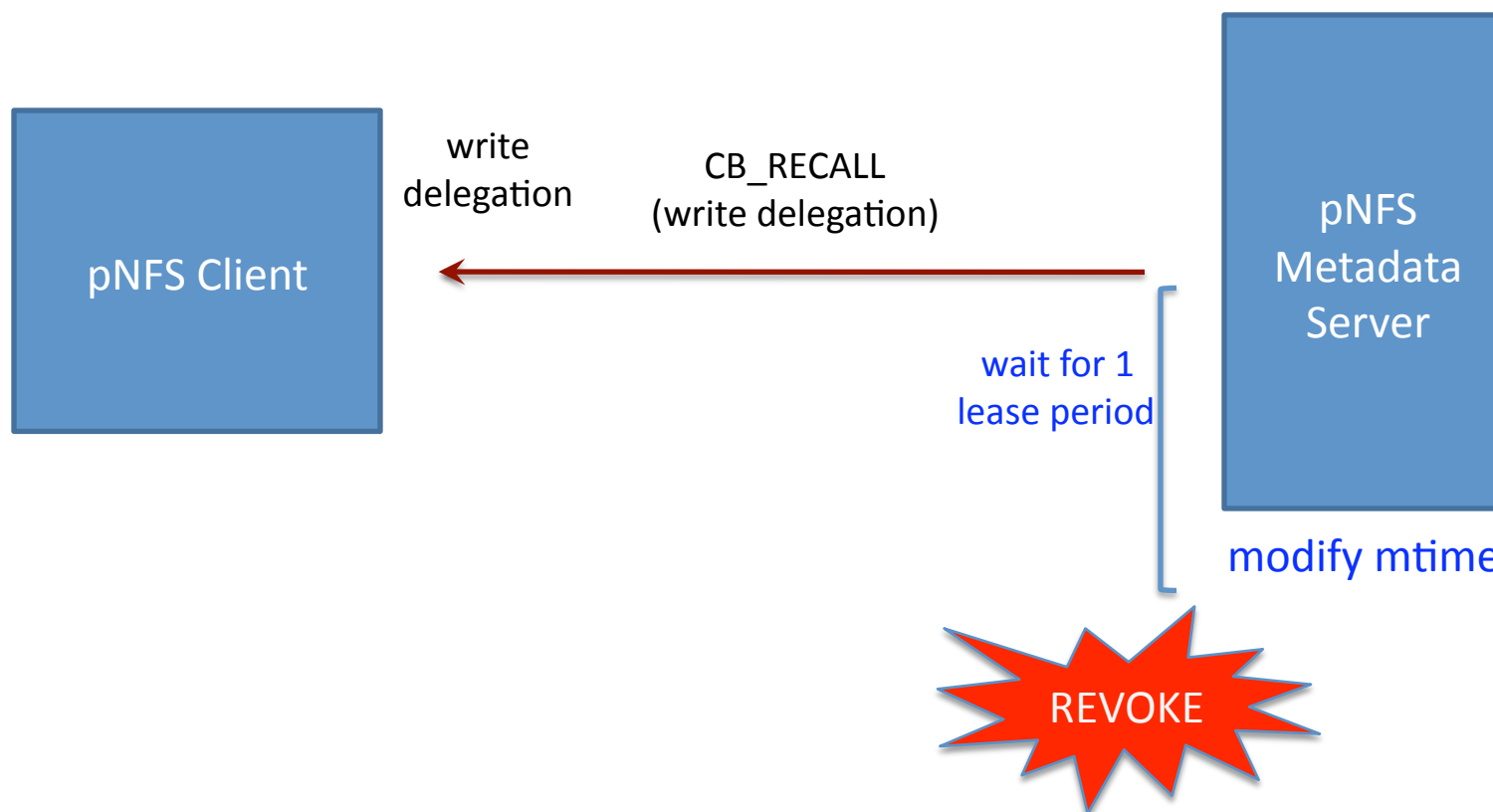
```
write(fd, buf, size)
ls -l <file>
```



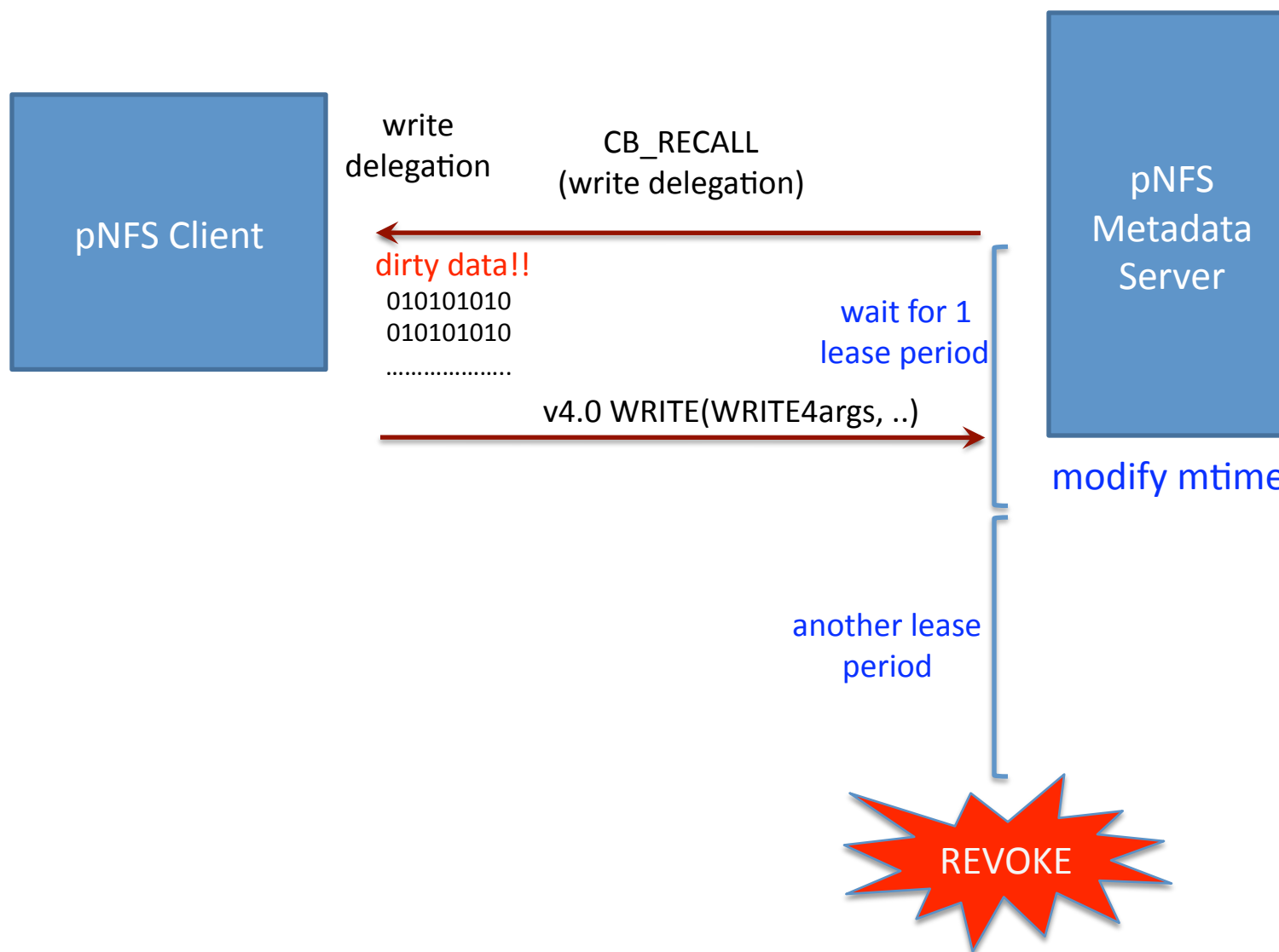
Updating a file's mtime

- **mtime attribute**
 - > Updated when the file's data is modified
 - > Context: write delegations

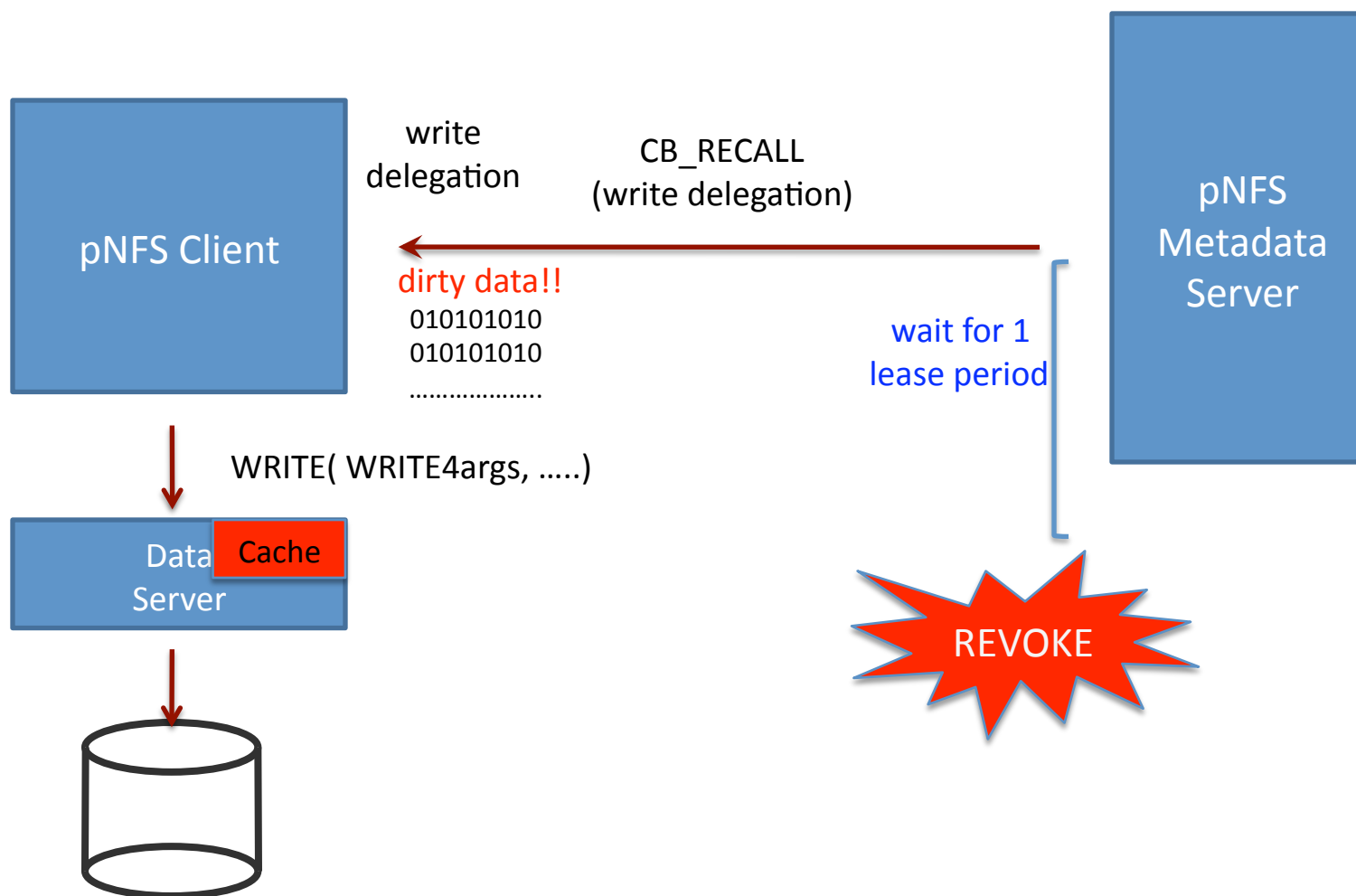
Use case: Delegation Recall



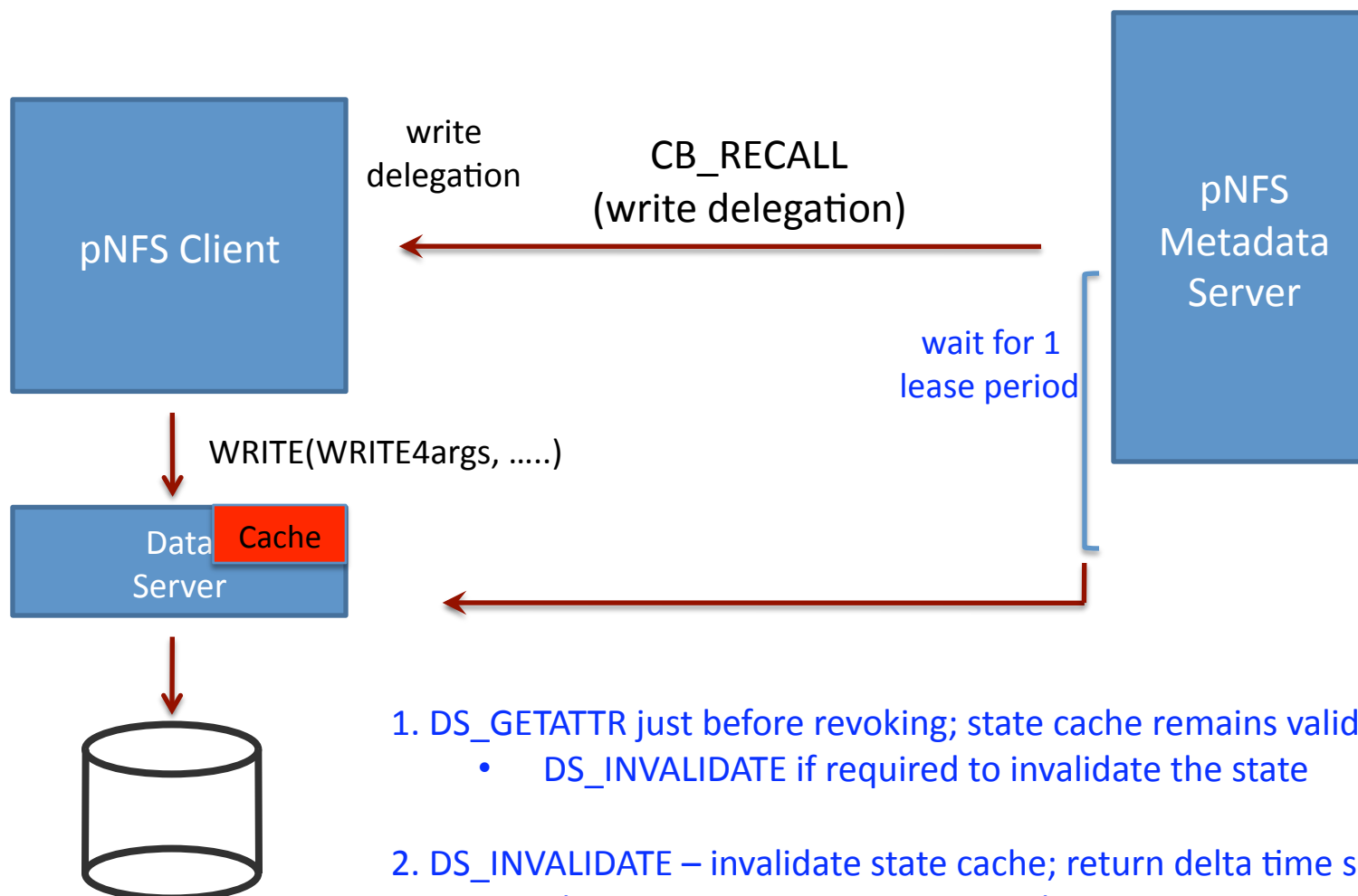
Use case: Delegation Recall



Use case: Delegation Recall

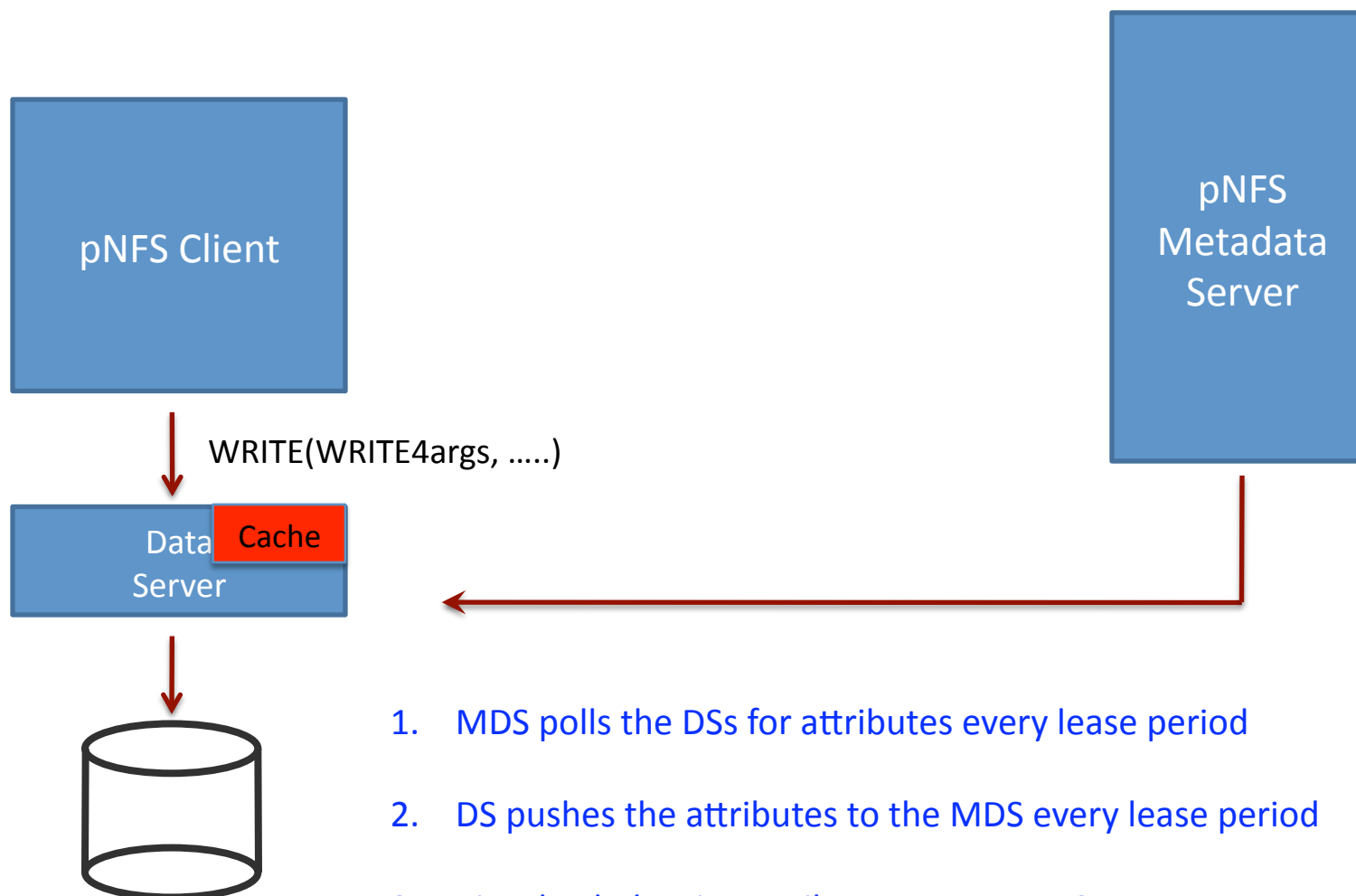


Strategies for mtime Update.



1. `DS_GETATTR` just before revoking; state cache remains valid
 - `DS_INVALIDATE` if required to invalidate the state
2. `DS_INVALIDATE` – invalidate state cache; return delta time since last write
 - The server may or may not revoke

Strategies for Size Update



Summary

- pNFS divides the file's metadata between DSs and MDS
- Introduces new challenges for state management
 - > Impact on performance
 - > Impact on correctness

Questions?