



Object Store

- Und was das mit Forschungsdaten zu tun hat

Thomas Eifert



Was ist ObjectStore?

- Was ist es nicht?
 - Kein General Purpose Speicher
 - Kein Ersatz für einen Fileserver (vielleicht für Teile)
 - Kein POSIX-Filesystem
 - Grund-Operationen
 - Rein (speichern)
 - Raus (lesen)
 - (Liste)
 - löschen
 - Zugriff über Objekt-Identifizier
- ➔ Kein Ändern von Dateien (wird ggf. vom Client simuliert)
- ➔ Operation immer auf ganzen Dateien („Objekten“)

Was bringt das?

- Wegfall von sehr viel Synchronisierungsaufwand (Posix: inode-Operationen sind atomar)
- Nur lose Kopplung zwischen „Nodes“ notwendig
- Sehr einfache, damit kostengünstige Skalierbarkeit: Ein Objektspeicher kann viele PB groß werden ohne Zunahme von Komplexität

Verbindung zu FDM

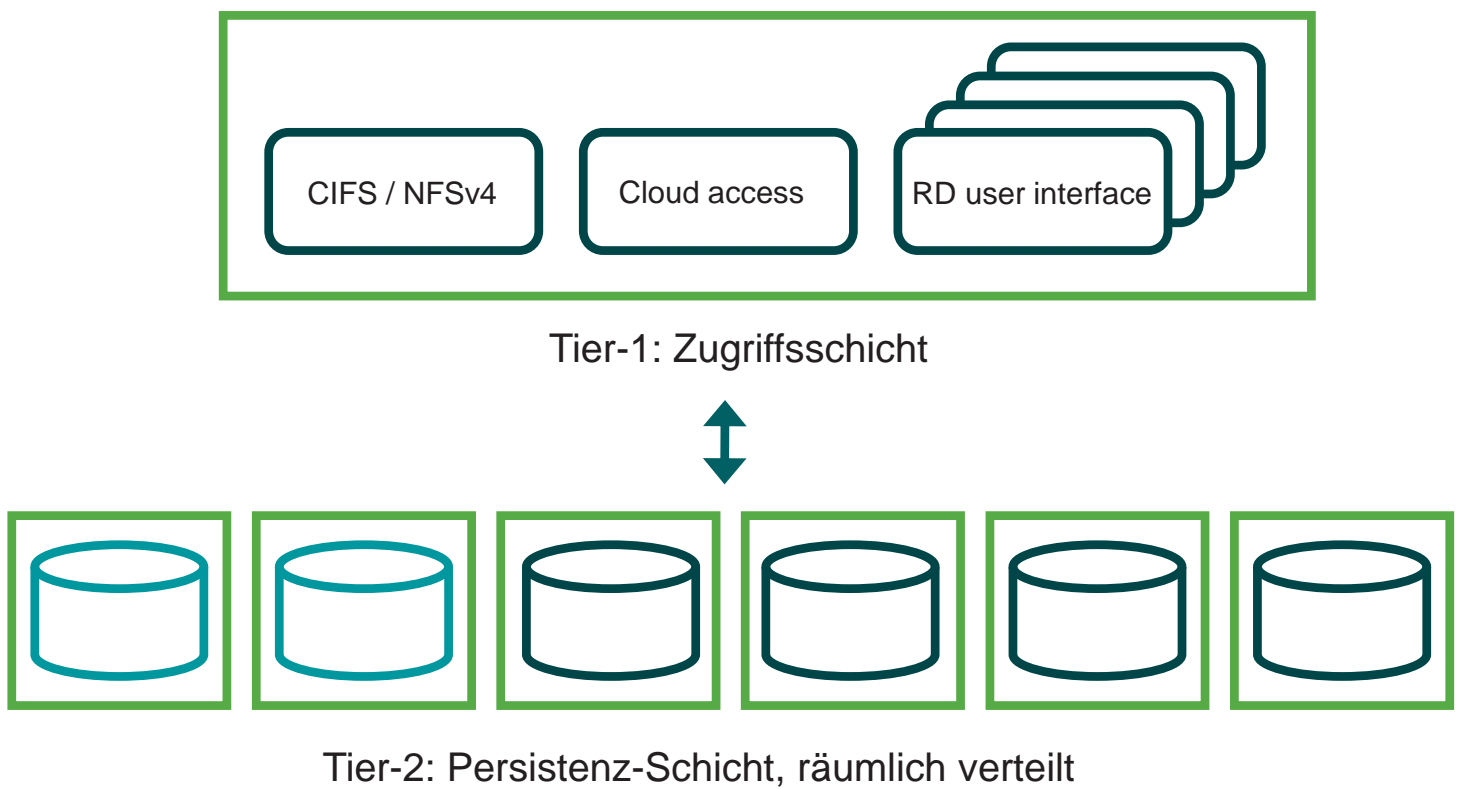
- Primärdaten entstehen meist aus Messungen, bildgebenden Verfahren (Mikroskope, ..), Simulationen, ..
- Nach Abschluss der Messung (..) liegt ein abgeschlossener Datensatz vor
→ ein Daten-Objekt
- FDM-Prozessen soll Speicherort egal sein
→ Zugriff über Objekt-Identifizier, nicht über server:\pfad\dateiname
- Prozess-Logik und Logik von Object Store passen perfekt zusammen

Absicherung

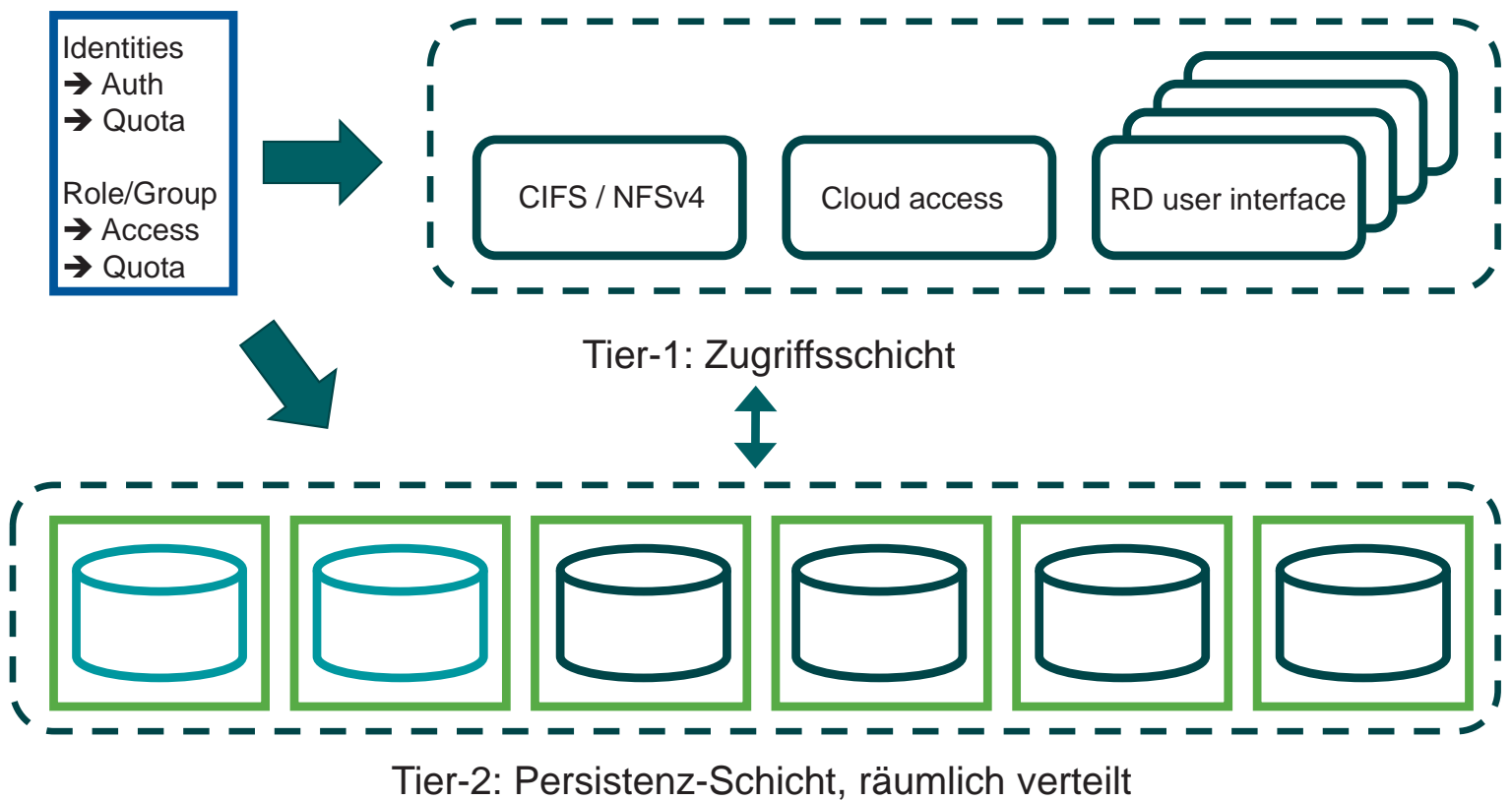
Im FDM-Kontext: Persistenz-Schicht

- Annahme: Daten-Verfügbarkeit nicht mittels Backup/Restore sicherzustellen: es geht um einige PB
- Lösung: Redundanz-Konzept auf Objekt-Ebene: Erasure-Coding
 - Grundidee wie bei RAID: Zu Nutzdaten wird Redundanz erzeugt (RAID5: 1 Platte, RAID6: 2 Platten – Block-Ebene, Latenz-sensitiv)
 - Objekte nicht Latenz-sensitiv → Räumlich verteilbare Redundanz
 - Ziel: Dienst härten auch gegen Totalverlust eines Gebäudes → N+2-EC; typisch zwischen 4+2 und 8+2
 - Verteilung der Knoten auf mehrere verteilte Server-Räume

Logisches Schichtenmodell



Logisches Schichtenmodell + Verzeichnisdienste



**Vielen Dank
für Ihre Aufmerksamkeit**

