



# Blob/Object Storage

Alexander Timmermann

Proseminar “Speicher- und Dateisysteme”



# Klassifikationen

## Block-based

- buffered access auf Systemressourcen
- Applikationen können Blöcke von beliebiger Größe und beliebigem Offset schreiben/lesen
- Kernel (o.Ä) kümmert sich um Zugriff auf Hardware

## File-based

- Organisation der Blöcke in Dateien und Ordner mithilfe von Metadaten
- Erfassung weiterer Metadaten (letzter Zugriff, Berechtigungen, Modifikationen...)
- spezifische Implementationen mit speziellen Features

## Object-based

- Speichereinheit: ein Objekt
- flacher Namespace
- Objekte haben einen fixen Identifier, angehängte Metadaten, und variablen Inhalt



# Object Storage

- Kann auf unterschiedlichen Ebenen implementiert werden
  - device level
  - system level
  - interface level
- große Mengen unstrukturierte Daten speicherbar
- Nutzer:
  - Facebook (*Haystack*) → Nutzerfotos
  - Spotify → Songs
  - Dropbox → alle Dateien



# Object Storage

- Object Storage ist ähnlich zu Key-Value-Stores, mit ein paar Unterschieden:
  - Bei Object Stores sind Metadaten ein wichtiger Teil des Objekts
  - Object Stores sind optimiert für große Mengen von Daten
  - Konsistenzkriterien in Object Stores oft geringer (*eventual consistency*)



## Vorteile

- ⊕ Hoher Durchsatz
- ⊕ Hohe Skalierbarkeit von Kapazität
- ⊕ Zugriffs- und Berechtigungseinstellungen auf Objekt-Ebene
- ⊕ Metadaten sind separat und anpassbar

## Nachteile

- ⊖ hohe Komplexität



# Beispiele

- Amazon S3, Azure Object Storage,...
- OpenStack Swift
- Ceph/RADOS
- Lustre
- BlobSeer
- Týr
- dedizierte OSDs (Object Storage Devices)



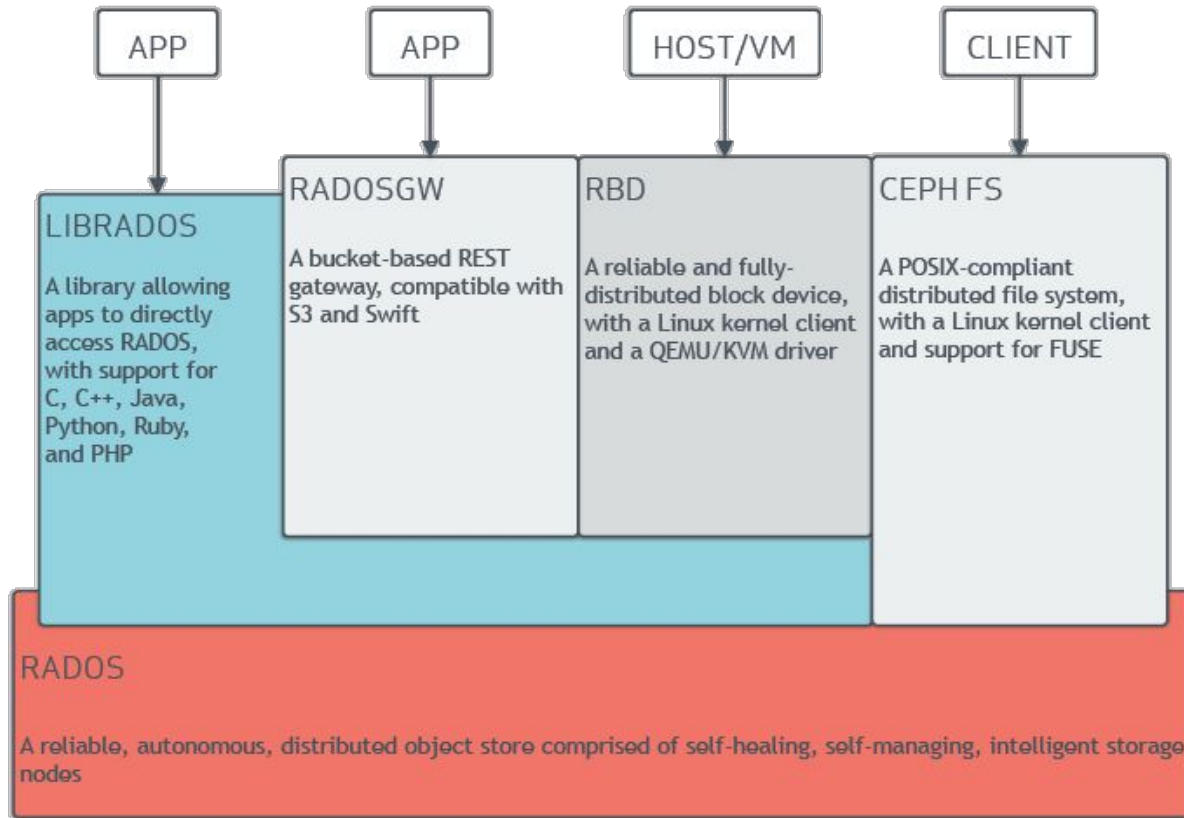
# POSIX...?

- Portable Operating System Interface
- Gruppe von Standards des IEEE für eine API zwischen Userspace und OS
- u.A. für IO-Aufrufe

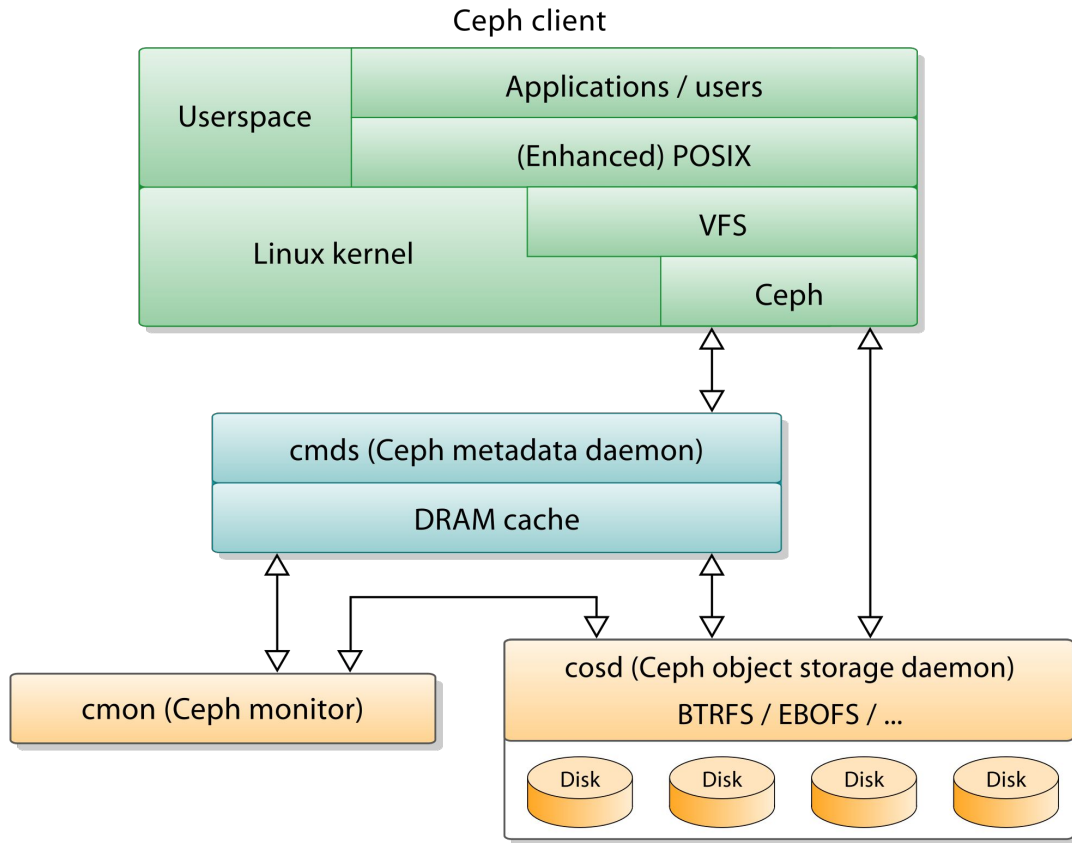


- verteilte Storage-Lösung auf Basis von RADOS (*Reliable Autonomous Distributed Object Storage*)
- bietet drei Arten von Storage-Architekturen an





<https://github.com/ceph/ceph/blob/master/doc/images/stack.png>



[https://commons.wikimedia.org/wiki/File:Ceph\\_components.svg](https://commons.wikimedia.org/wiki/File:Ceph_components.svg)



- ähnliche Architektur:
  - Metadata Server (MDS)
  - Object Storage Server (OSS)
- MDS in keinen I/O-Operationen involviert ⇒ weniger Bottlenecks

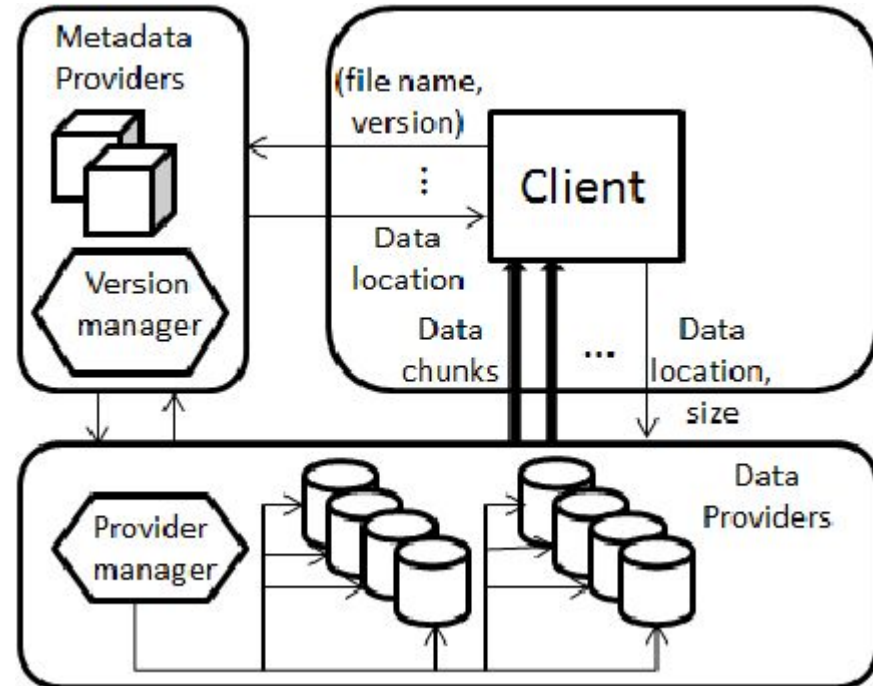


## Blob...?

- “Backronym” für Binary Large Object
- hier: große Kollektion von Daten

# BlobSeer

- akademische Implementation
- handled Blobs bis zum TB-Level
- atomare Snapshots





- Blobs in Containern organisiert
- verschiedene Typen von Blobs
  - Block Blob
  - Append Blob
  - Page Blob
- Closed Source



# Týr

- Blob Storage System mit Transaktions-Support
- höhere Konsistenz-Gewährleistung
- noch sehr experimentell



# Quellen

- Bogdan Nicolae, Gabriel Antoniu, Luc Bougé, Diana Moise, Alexandra CarpenAmarie. BlobSeer: Next Generation Data Management for Large Scale Infrastructures. *Journal of Parallel and Distributed Computing*, Elsevier, 2011, 71 (2), pp.168-184.
- Pierre Matri, Alexandru Costan, Gabriel Antoniu, Jesús Montes, María S. Pérez. Tyr: Blob Storage Meets Built-In Transactions. *IEEE ACM SC16 - The International Conference for High Performance Computing, Networking, Storage and Analysis 2016*, Nov 2016, Salt Lake City
- Chandrasekaran, Arun, Dayley, Alan (11 February 2014). "Critical Capabilities for Object Storage". Gartner Research.
- Factor, Michael; Meth, K.; Naor, D.; Rodeh, O.; Satran, J. "Object Storage: The Future Building Block for Storage Systems" (PDF). IBM Haifa Research Labs.
- P. Matri, Y. Alforov, Á. Brandon, M. Kuhn, P. Carns and T. Ludwig, "Could Blobs Fuel Storage-Based Convergence Between HPC and Big Data?," 2017 IEEE International Conference on Cluster Computing (CLUSTER), Honolulu, HI, 2017, pp. 81-86.