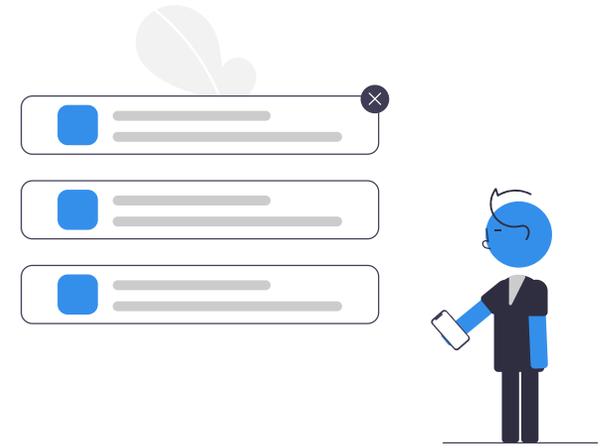


The New Generation of Data Stores

Alexander Reelsen | alr@spinscale.de | @spinscale@mastodon.social

Agenda

- Cloud native shift
- Splitting Storage & Compute
- Impact for cloud providers
- Impact for your business
- Impact for developers





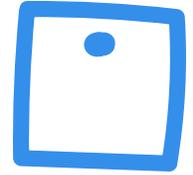
Into the cloud: *Lift + shift*

- Your strategy?
- Your strategy **for databases**?
- Consume more cloud services over time?

Data stores in the cloud

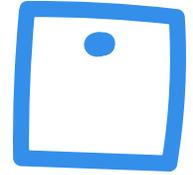
- Hosted by the cloud provider
 - Minor modifications & version lock-in
- Hosted by the database provider
 - Version lock-in
- Hosted by yourself
 - Upgrade complexity
 - Maintenance work





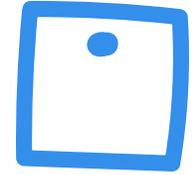
Data store architecture – Single system

- ✓ Single system handling reads & writes
- ✓ No coordination required
- ✓ Easy handling of transactions
- ✗ SPOF
- 💰 Scaling strategy: Scale-up
- 🐘 Example: Postgres



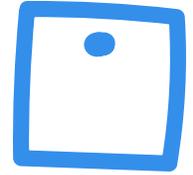
Architecture – Decouple reads and writes

- ✓ Decouple reads from writes
- ✓ Some coordination required
- ✓ Leader/follower architecture
- ✗ SPOF
- ✗ Complexity moves to the client (several endpoints)
- 💰 Scaling strategy: Scale-up for writes, scale-out for reads
- 🐘 Example: Postgres



Architecture – Sharding & Replication

- ✓ Keep copies of data in cluster, read scalability
- ✓ Split data across cluster, write scalability
- ✓ No SPOF (at the price of coordination), zero-downtime upgrades
- ✓ Scaling strategy: Scale-out
- ✗ Complexity, coordination, maintenance
- 💰 More machines to keep data safe, duplicating writes
- 🚦 Example: Elasticsearch



Architecture – Sharding & Replication

- ✗ Reads and writes are not decoupled
- ✗ Performance impact on load shift
- ✓ Heterogenous clusters/Tiering (hot, warm, cold, frozen)

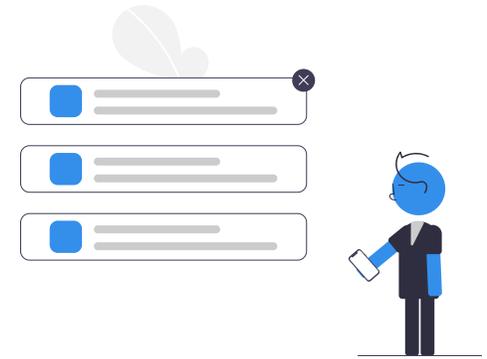


Architecture – Streaming

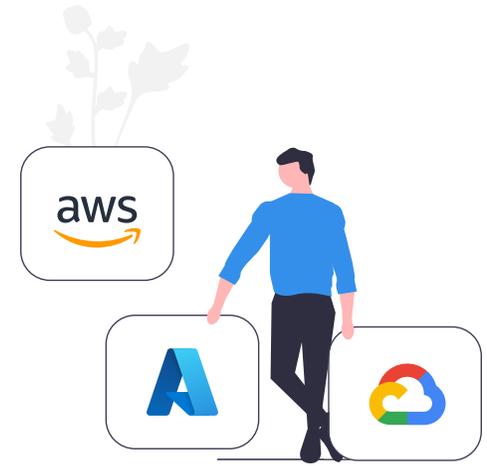
- 💰 Streaming platforms as buffer
- ✅ Decouples writes
- ❌ Increased cycle time from event generation to storage
- ❌ Increases architecture complexity

What's the goal?

- ✓ Decouple reads & writes => independent scalability
- ✓ Write data once
- ✓ Share storage among readers
- ✓ Minimal coordination
- ✓ Replace tiers with tasks
- ✓ Cross region replication
- ✓ **Cost reduction**

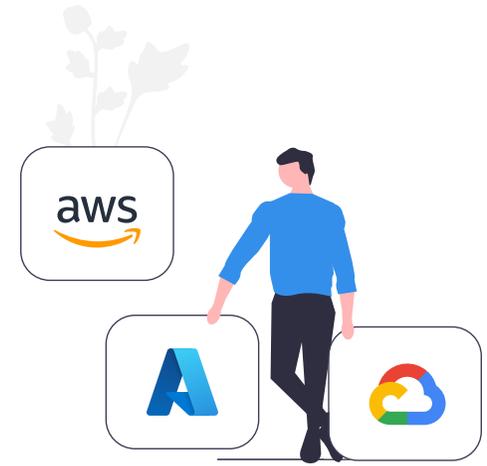


OHAH hyperscalers

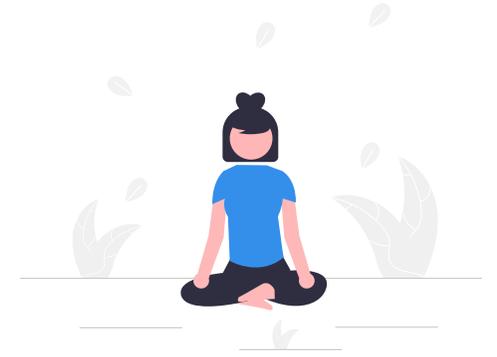


- Major advancements in cloud services
- Object storages improved on durability, semantics & speed, reducing the requirement of replication within your application
- **Significant** price drops (1TB S3 \$23 a month, Cloudflare \$16, Backblaze \$5)

Hyperscalers rule the data world

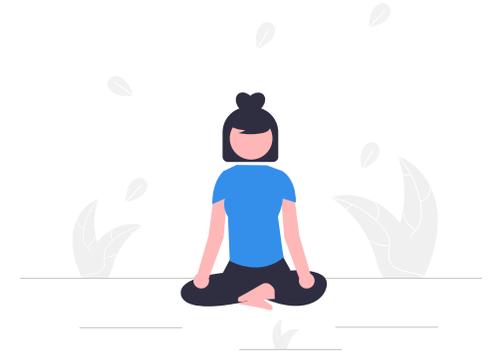


- ... have the biggest incentive to drive down cost of services
- ... have the biggest incentive to scale down to zero
- ... have always been drivers for data store change
- ... are a couple of years ahead



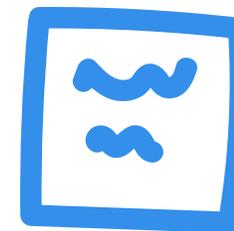
Scale to zero

- Runtime environments: AWS Lambda (2014!), Google Cloud Run
- Databases: AWS Aurora (2018!), GCP AlloyDB, CockroachDB
- Drive-by effect: Pay per use



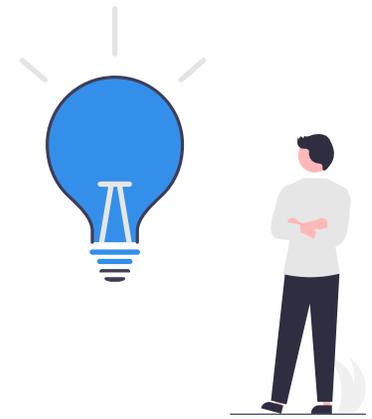
Scale to zero – data store requirements

- Shut down an instance without losing data
- Quick start-up, single digit seconds
- Split storage & compute



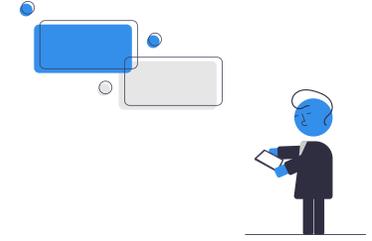
Storage

- Share storage among instances
- Concurrent read access (free!)
- Concurrent write access (locking!)
- Slower than local storage!



Become object storage smart

- Full table scan? aaaaaah
- Optimize for object storage access patterns
- Smarter query plans will not just mean less CPU/IO but lower the bill
- Assign exact costs to queries & inserts



Compute

- Indexing and querying can be split
- No activity allows shutting down
- Distributed System with shared storage (consistency/transactions)
- Merging/Vacuuming can now be another compute instance
- Local storage acts as cache, bigger compute means bigger cache



Increased latency as an acceptable tradeoff

- Transactional writes need to be duplicated or waited for on the object storage (WAL)
- Batch friendlier
- Not suitable for (milli)second latency



Increased latency as an acceptable tradeoff

We ultimately decided that a few hundred milliseconds increase in the median latency was the right decision for our customers to save seconds, tens of seconds, and hundreds of seconds of p95/p99/max latency, respectively.

Richard Artoul about [Husky](#), Datadog Event Storage



Data Stores Splitting Storage & Compute

- [ClickHouse](#) (analytics)
- [CockroachDB](#) (distributed SQL)
- [Firebolt](#) (cloud data warehouse)
- [Neon](#) (postgres extension)
- [Quickwit](#) (log analytics & search)
- [SingleStore](#) (analytics)
- [Yellowbrick](#) (data warehouse)
- [Yugabyte](#) (distributed SQL)



Data Stores Splitting Storage & Compute

- [Amazon RedShift](#)
- [Google BigQuery](#) (2012!)
- [Snowflake](#)



Elasticsearch

separate compute from storage to simplify your cluster topology

Stateless – your new state of find with Elasticsearch



Cloud Native Lucene-Based Search Engines

- [nrtSearch](#) by Yelp
- [KalDB](#) by Slack
 - Kafka, Zookeeper, S3



Retrofitting is tough!

After undertaking a multi-month proof-of-concept and experimental phase

Stateless – your new state of find with Elasticsearch



Why are they doing it?

- Stay competitive in pricing to the hyperscalers
- Provide economically feasible many TB/PB storage engines
- Reduce resource consumption
- Point-in-time recovery
- Fork your data , but cheap
- Differentiator in SaaS offering

Downsides of this trend



Hyperscalers — a threat to Open Source?

- Problem: **Hyperscalers & other companies** offer open source software as SaaS without contributing back (to VC backed projects)
- Relicensing: [Mongo](#), [Elastic](#), [Confluent](#), [Graylog](#), [Grafana](#), [Redis](#)
- BSL: [CockroachDB](#), [Couchbase](#)
- Forks: [Opensearch](#)
- Good read: [Righteous, Expedient, Wrong](#)



Hyperscalers — a threat to Open Source?

- OSS monetization moves to SaaS monetization (investor driven)
- Open Source vs. Open Code
- Future: Less open source, partial open source, closed source control planes?
- More blackboxed systems?
- Different compatibility guarantees - wire protocol, SQL



Development becomes more complex

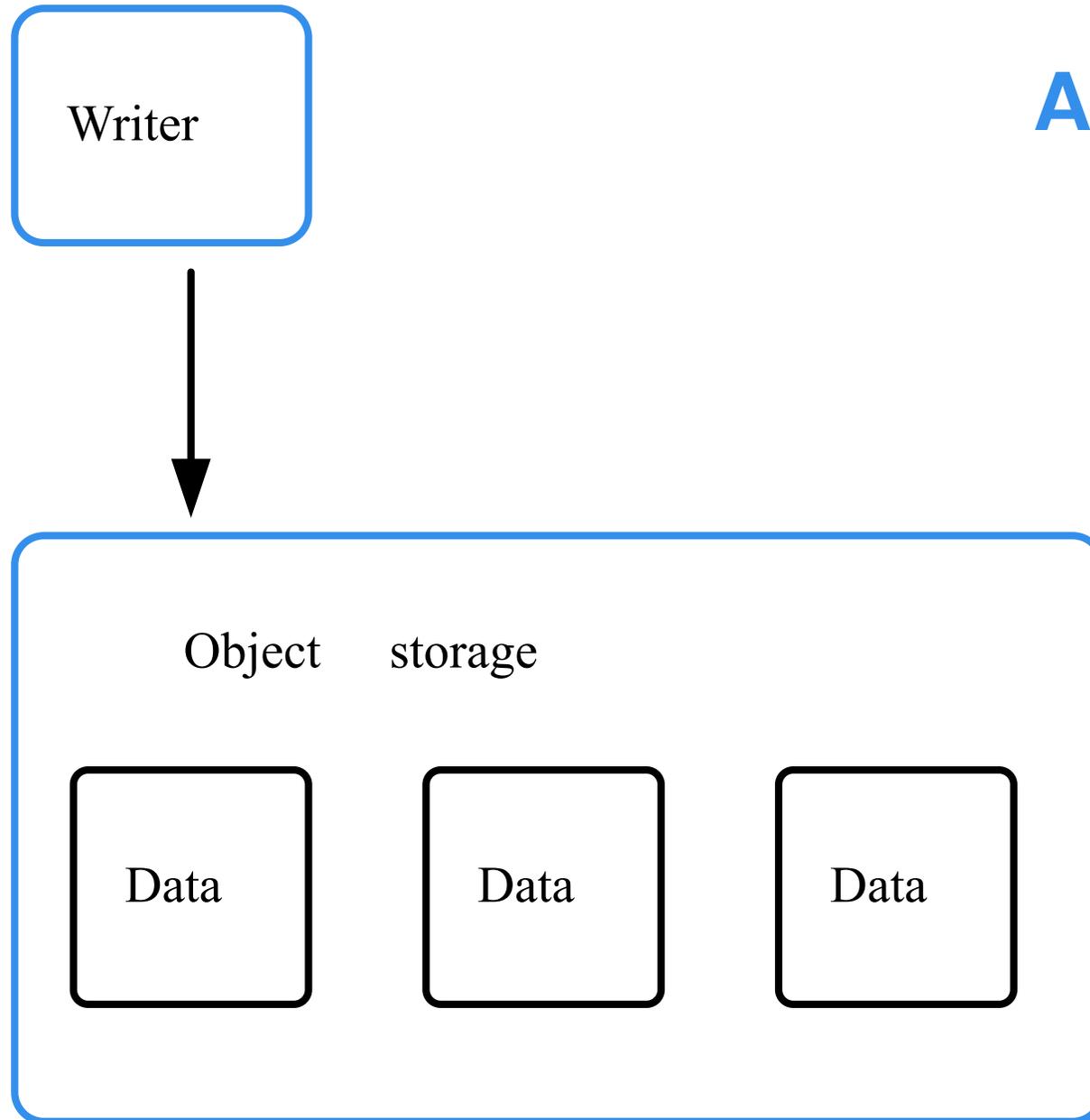
- New generation of data stores will not run on all cloud providers
 - Threat to DigitalOcean, Scaleway, Scalingo, Vultr, OVH
- New generation of data stores will not run locally
 - Testing
 - Fast feedback
 - Offline development
 - Preproduction systems (just kidding!)
 - **Cheap** CI runs



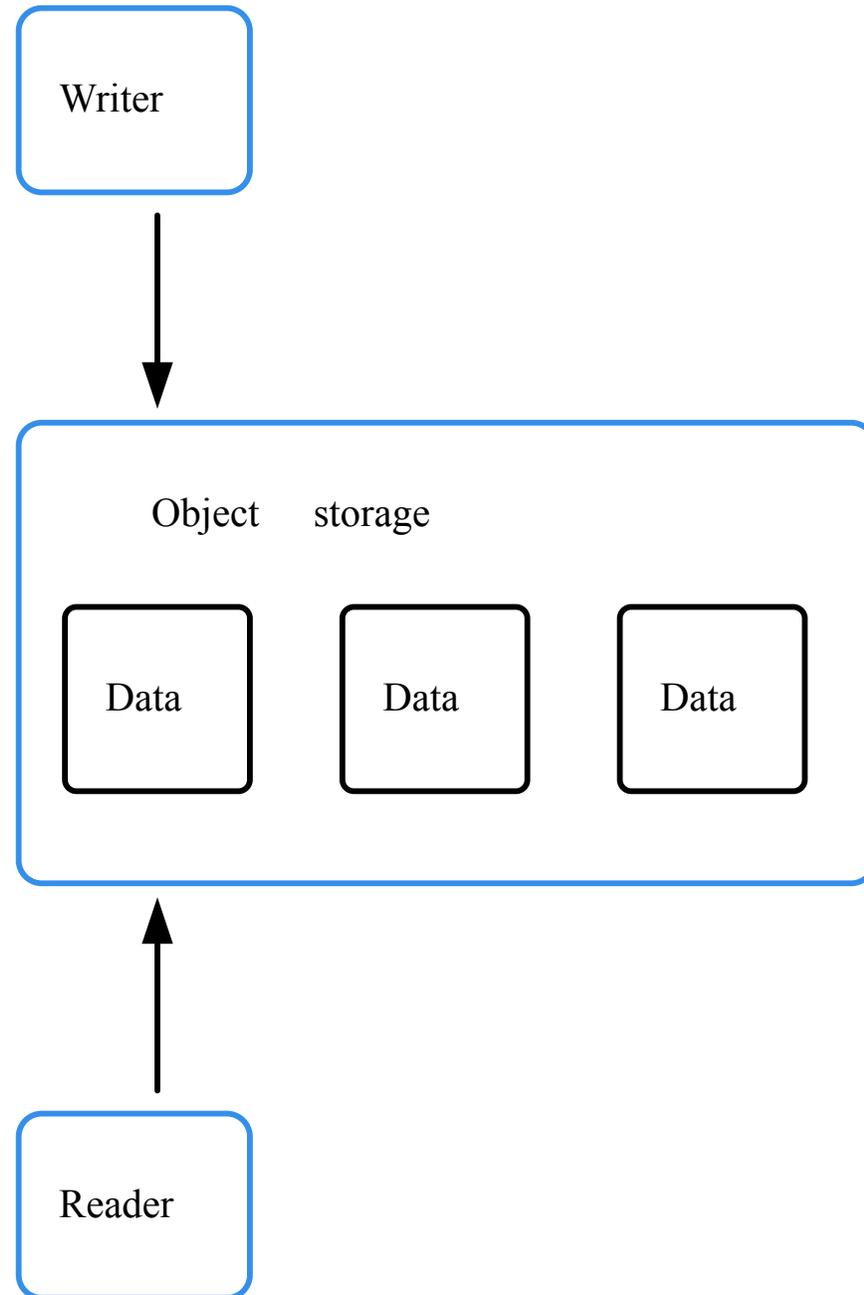
Debugging

- Much more blackbox
- IOPS/bandwidth from remote storage
- Network speed
- Performance stability and predictability

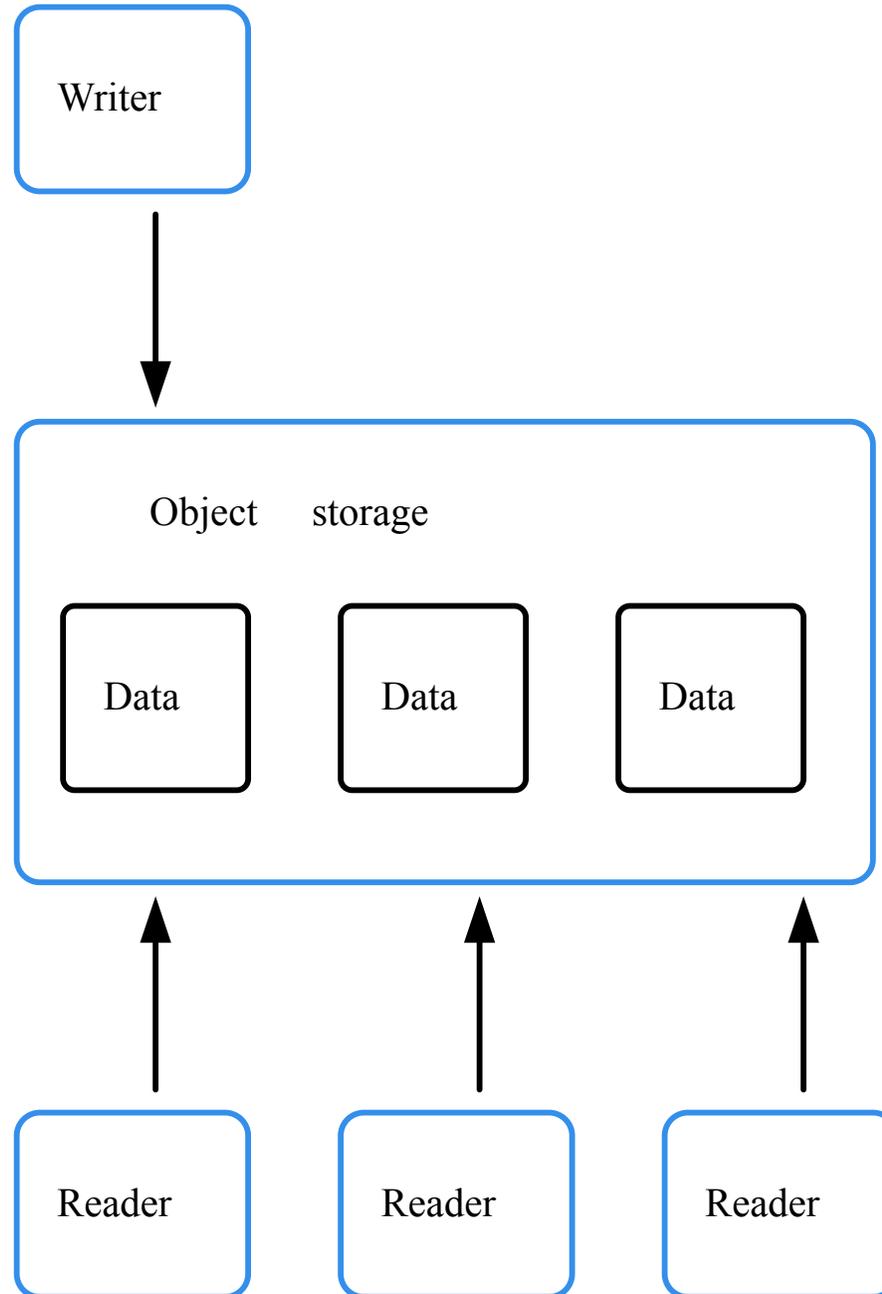
Architecture

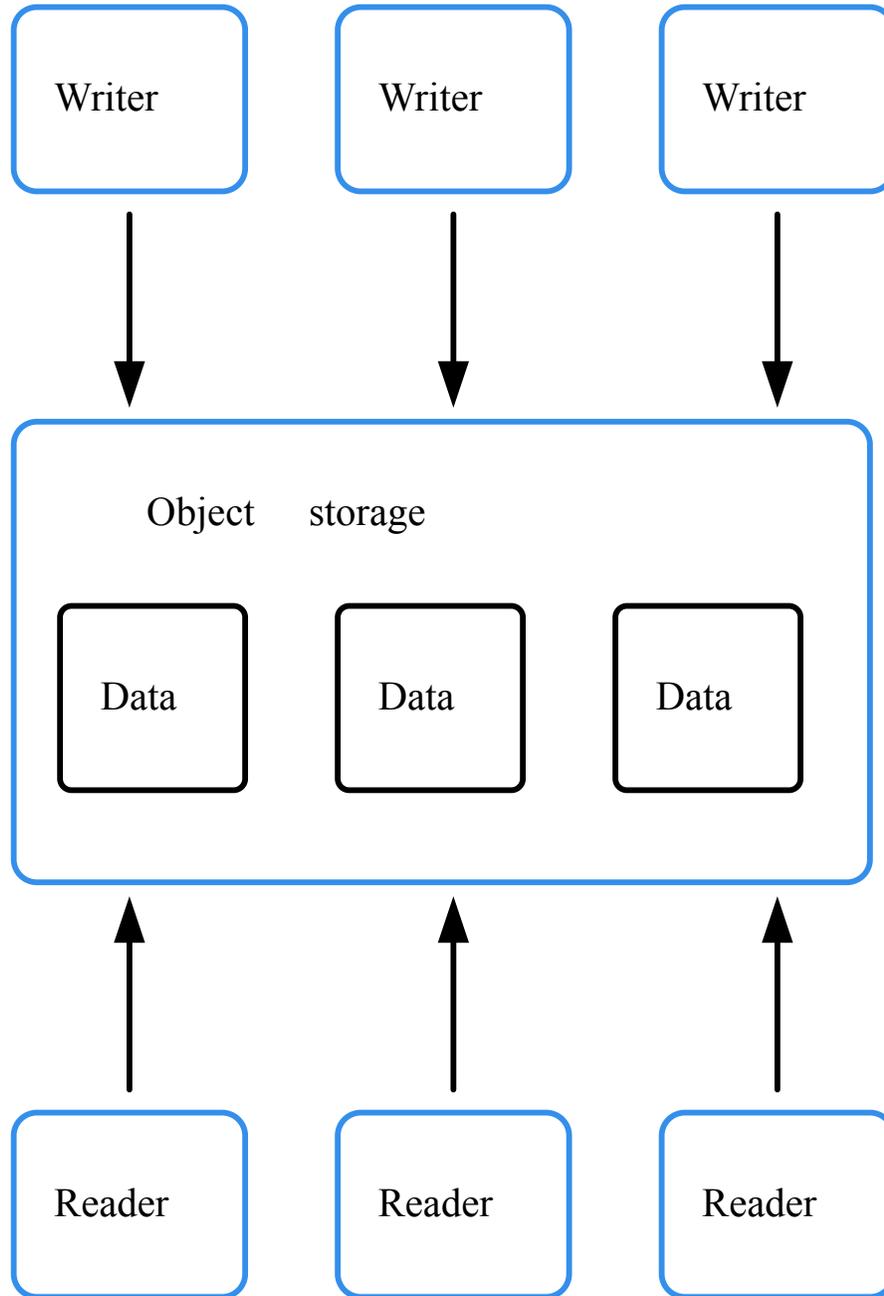


Architecture

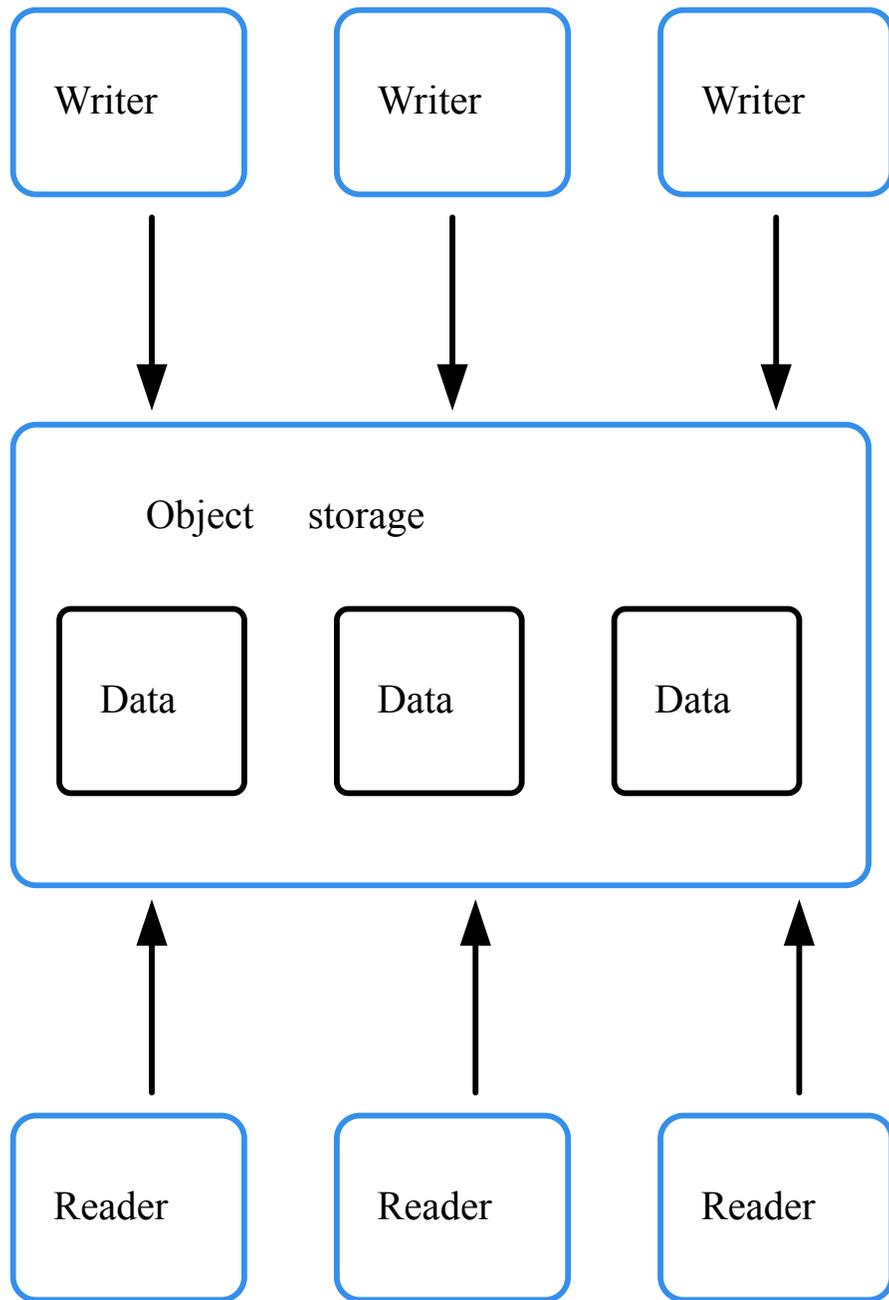


Architecture





Architecture



Architecture





Coordination – Append only

- When is a write `done` ?
- Multi file moves are not atomic
- Notify readers of new data



Coordination – Updates & Deletes

- No `in place` updates
- Append + Mark as deleted
- Notify readers of new data
- Local storage cache invalidation



Coordination - Cleanup/Vacuum

- Remove unused data
- Remove data marked as deleted
- Compact to bigger segments supporting better compression
- How to remove data, that is `in use`? No notion if inodes



Coordinator

- Gets notified when data is ready to be queried
- Notices about new data to query
- Triggers Merging/Vacuuming
- Needs to persist its state in case of crash?
- Coordinator might be embedded into another role

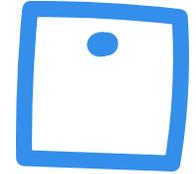


Summary

- Cloud only data store trend will continue
- System will be able to store more data, PBs as commodity
- Resource consumption for CI & development is an unsolved problem
- Offline development will cease to exist (cloud IDEs on the rise)

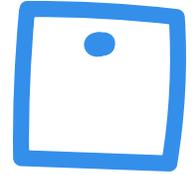
Thanks for listening!

Alexander Reelsen | alr@spinscale.de | @spinscale@mastodon.social



Where can I learn more?

- [CMU Database Talks](#)
- [Book: Understanding Distributed Systems](#)
- [Book: Designing Data Intensive Applications](#)
- [Book: Database Internals](#)
- [The Firebolt Cloud Data Warehouse Whitepaper](#)
- [Article: Separating compute and storage](#), from **2019**
- [BigQuery under the hood](#), from 2016



Where can I learn more?

- [This talk as a blog post](#)
- [spinscale.de](#) - my blog

Thanks for listening!

Alexander Reelsen | alr@spinscale.de | @spinscale@mastodon.social