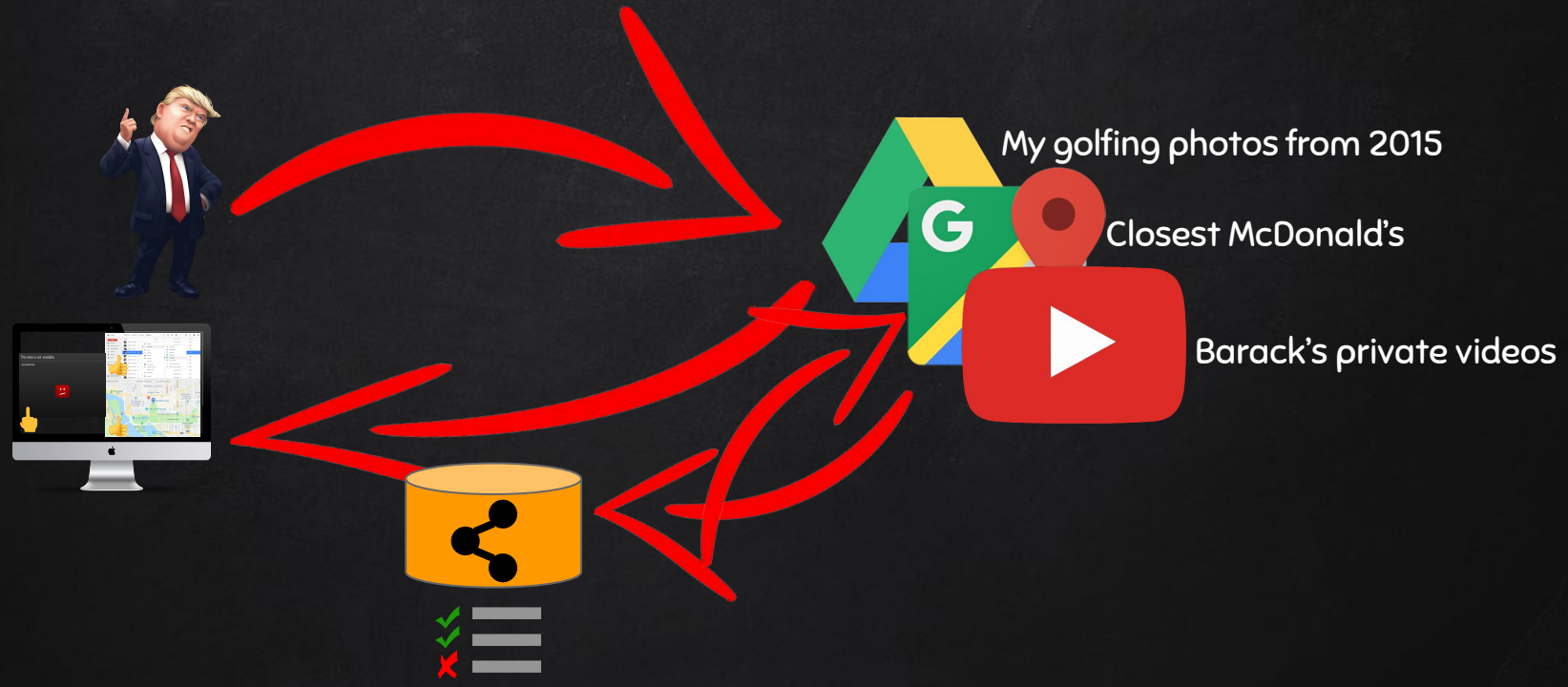
Zanzibar: Google's Consistent, Global Authorization System

Presenter: Amir Ilkhechi

Privacy's core requirement: authorization checks



ACLs (Access Control Lists)

Authorization Engine

Zanzibar Design Goals

Correctness

Flexibility

Scalability

Low Latency

High Availability

Namespaces, relations, usersets, and tuples

Namespace: videos

Object	Relation	Userset
Video A	viewer	User A
Video B	viewer	All users
Video C	commenter	User Z

Namespaces, relations, usersets, and tuples

Namespace: videos

Object	Relation	Userset
Video A	viewer	User A
Video B	viewer	All users
Video C	commenter	User Z

ACL check results:

- Video A, viewer, user A? ✅
- Video A, viewer, user F? ❌
- Video B, viewer, user B? ✅
- Video C, commenter, user Z? ✅

Userset Indirections...

Namespace: videos

Object	Relation	Userset
Video A	viewer	(Group G, member)
Video B	viewer	(Group K, member)
Video C	commenter	User Z

Namespace: groups

Object	Relation	Userset
Group G	member	(Group A, member)
Group K	member	User B
Group K	member	User C

deep

wide

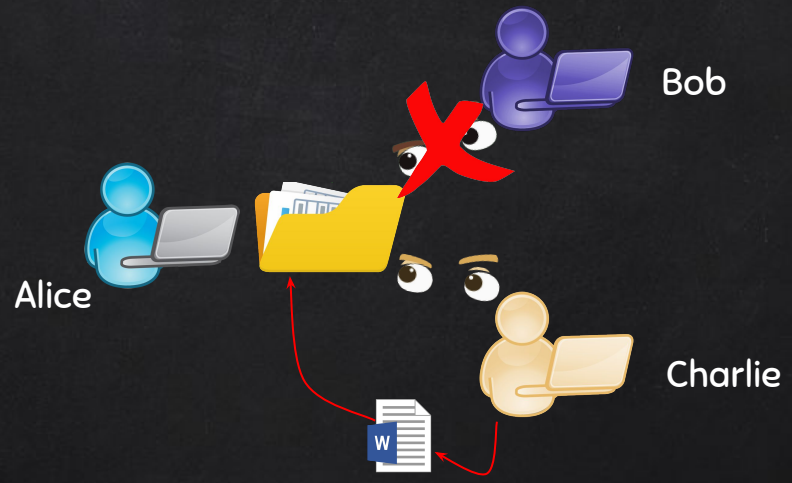
ACL check results:

- Video B, viewer, user B? ✅
- Video B, viewer, user F? ❌

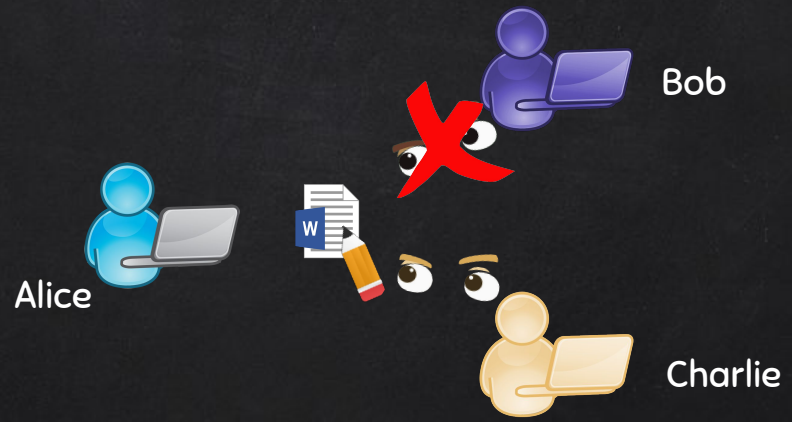
Namespace config

```
name: "doc"

relation { name: "owner" }

relation {
 name: "editor"
 userset_rewrite {
  union {
   child { _this {} }
   child { computed_userset { relation: "owner" } }
} } }

relation {
 name: "viewer"
 userset_rewrite {
  union {
   child { _this {} }
   child { computed_userset { relation: "editor" } }
   child { tuple_to_userset {
     tupleset { relation: "parent" }
     computed_userset {
       object: $TUPLE_USERSET_OBJECT  # parent folder
       relation: "viewer"
   } } }
} } }
```

"New Enemy" Protection 1



"New Enemy" Protection 2



Consistency Protocol

Updates by Alice:

updateACL(doc x, viewer, remove Bob)

Timestamp T_0

Content update by Charlie:

CheckContentUpdate(doc x, writer, Charlie)

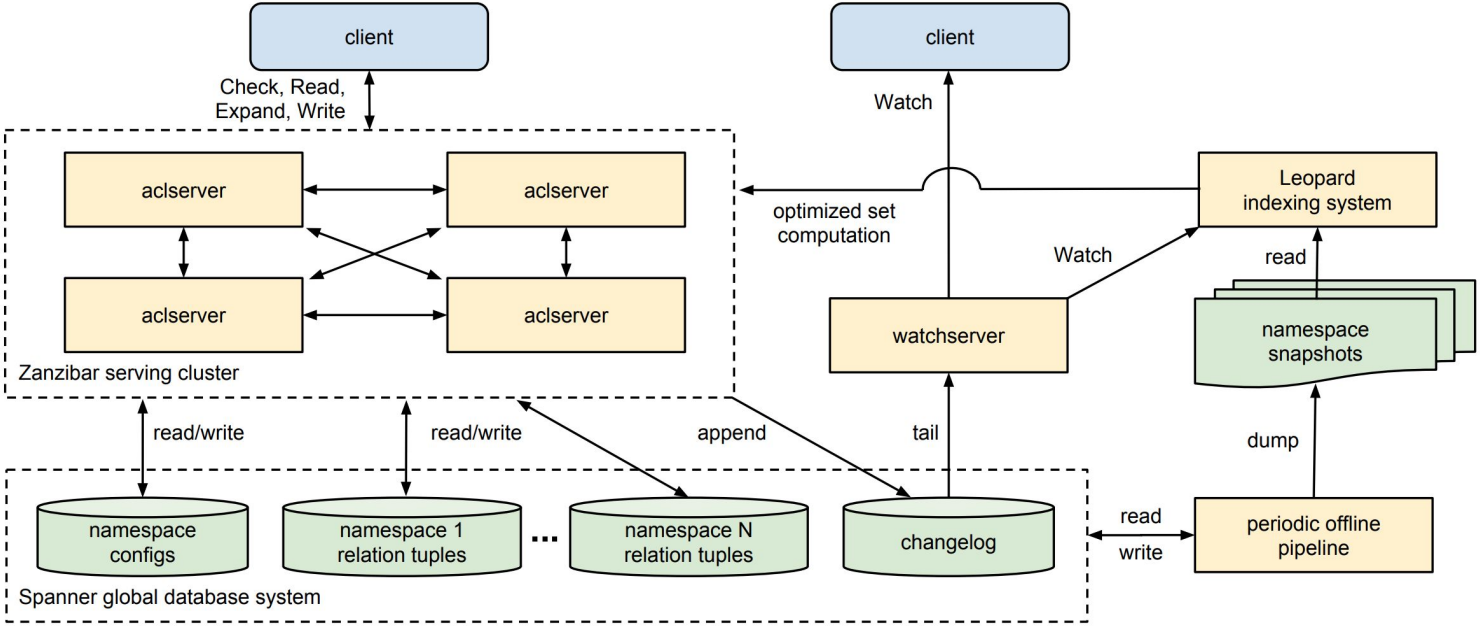
Yes, Timestamp T_1 [T_1 > T_0]

ACL check for Bob:

CheckACL(doc x, viewer, Bob, T_1)

No [at T_2 >T_1> T_0]

Architecture



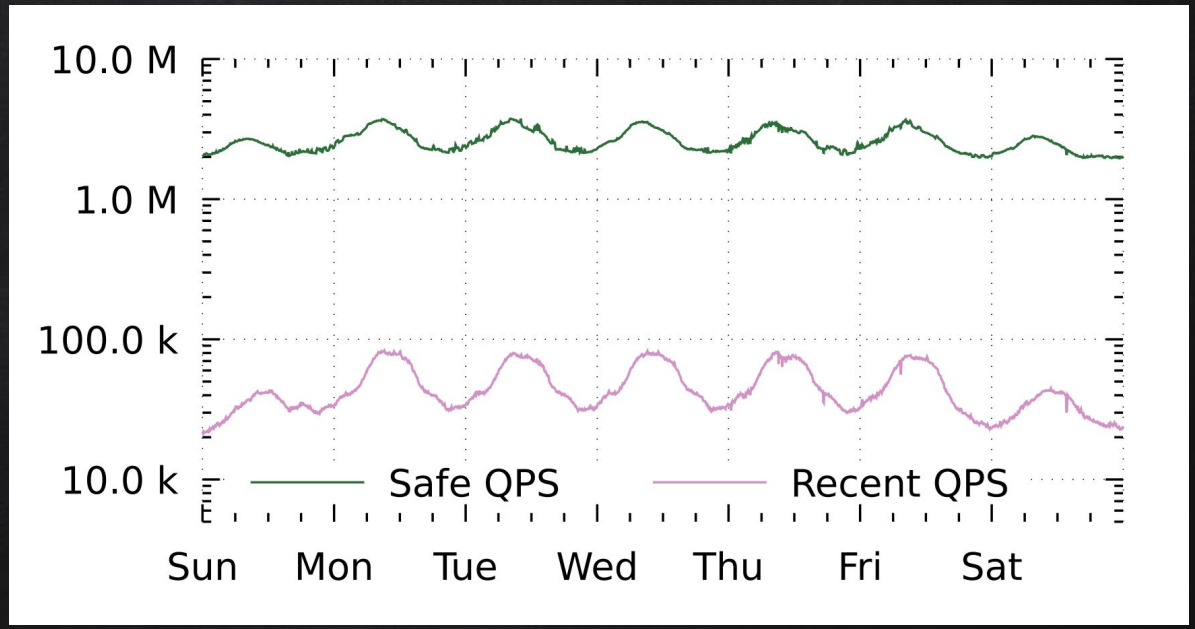
Optimizations

- Timestamps are chosen to reduce latency
- Hot-spot mitigation
- Request hedging
- Isolation
- Optimized processing of nested sets

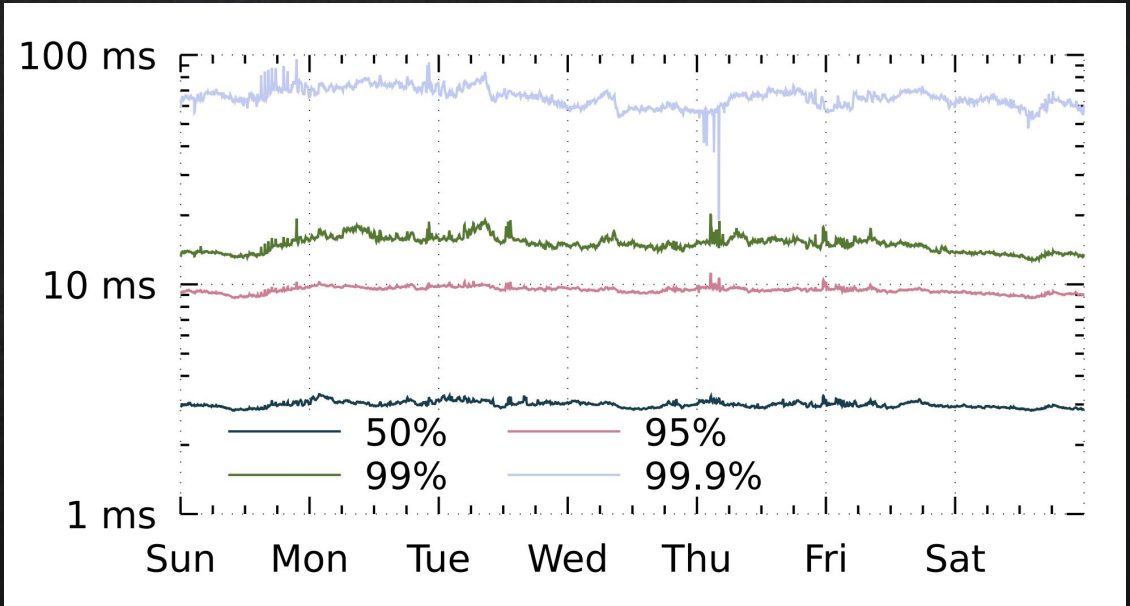
Zanzibar Deployment

- In production use for 5+ years
- Has 1500+ namespaces
- 2+ trillion relation tuples
- 10+ million queries per second
- 10+ thousand servers

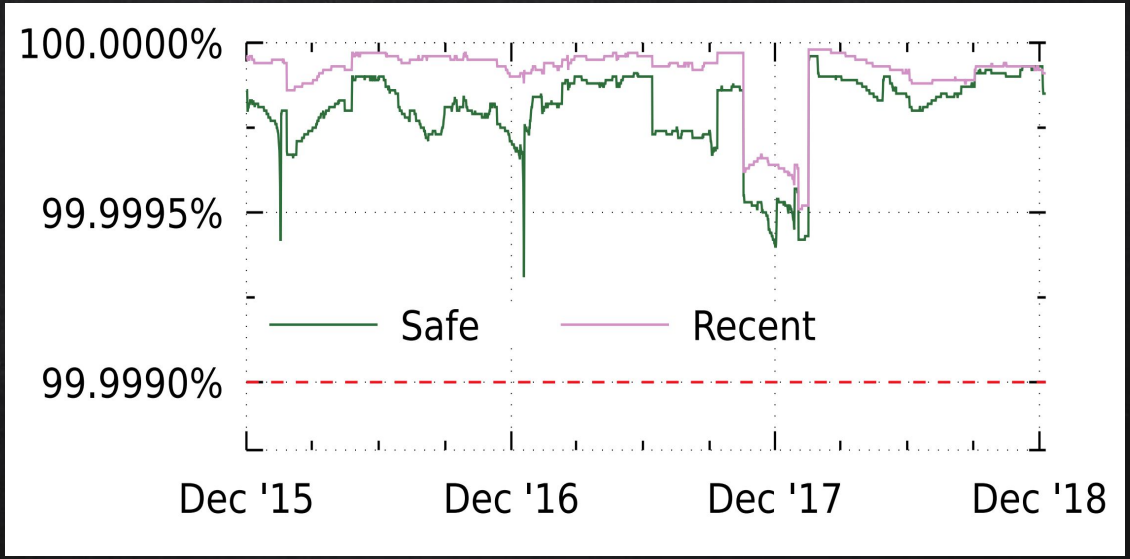
Rate of Check Queries



Check Safe Latency



Availability



Summary

- Robust authorization checks → essential to preserving privacy
- Zanzibar offers a unified authorization system:
  - Causal ordering of actions
  - Rich spectrum of access control policies
  - Low latency and high availability
  - Scalable
  - Flexible