Zanzibar: Google’s Consistent, Global Authorization System

Presenter: Amir Ilkhechi
Privacy’s core requirement: authorization checks

My golfing photos from 2015
Closest McDonald’s
Barack’s private videos
ACLs (Access Control Lists)

- “Video Gangnam Style is public”
- “Folder Istanbul 2015 is private”
- “Group G manages cloud project P”
Authorization Engine

"Can Alice view Video Gangnam Style?"

"Can friends group access Folder Istanbul 2015?"

"Can Bob create a VM under cloud project P?"
Zanzibar Design Goals

- Correctness
- Flexibility
- Scalability
- Low Latency
- High Availability
**Namespaces, relations, usersets, and tuples**

Namespace: videos

<table>
<thead>
<tr>
<th>Object</th>
<th>Relation</th>
<th>Userset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video A</td>
<td>viewer</td>
<td>User A</td>
</tr>
<tr>
<td>Video B</td>
<td>viewer</td>
<td>All users</td>
</tr>
<tr>
<td>Video C</td>
<td>commenter</td>
<td>User Z</td>
</tr>
</tbody>
</table>
Namespaces, relations, user sets, and tuples

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...

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_USERTSET_OBJECT # parent folder
                    relation: "viewer"
                }
            }
        }
    }
}
“New Enemy” Protection 1

Alice

Bob

Charlie
“New Enemy” Protection 2

Alice

Bob

Charlie
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]
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

- Safe
- Recent

Dec '15  Dec '16  Dec '17  Dec '18
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