KubeCon | CloudNativeCon
Europe 2025

From Chaos to Control: Migrating Access Control to OpenFGA in a Multi-Tenant World

Jo Guerreiro
Poovamraj Thanganadar Thiagarajan





Jo Guerreiro
Engineering Manager

Staff Engineer turned Engineering Manager for Grafana's Identity and access team.

Creator and Maintainer of yay, AUR helper for Arch Linux

What does your team work on?

The Identity and Access team provides a stable authentication and authorization platform for other teams to build their applications on.



Poovamraj Thanganadar Thiagarajan
Senior Software Engineer

KubeCon CloudNativeCon Europe 2025

Technical Lead for Integrations team in Auth0 FGA.

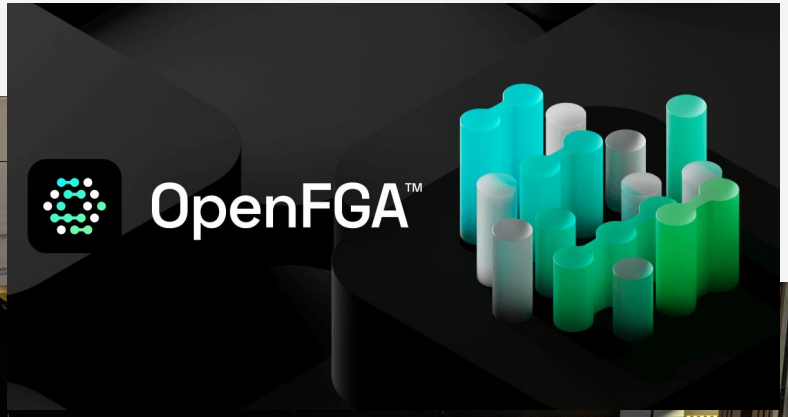
What does your team work on?

We work on integrating OpenFGA with different data sources to make the project easy to use out of the box. We build systems that integrate the access logs to observability/SIEM platforms. We also integrate with different Identity platforms to make it easy to use OpenFGA.

2024 Kubecon EU










Authorization is
unsolved

OWASP Top 10 List

- Broken Access Control
- Cryptographic Failures
- Injection
- Insecure Design
- Security Misconfiguration
- Vulnerable and Outdated Components
- Identification and Authentication Failures
- Software and Data Integrity Failures
- Security Logging and Monitoring Failures
- Server-Side Request Forgery

- Broken Object Level Authorization
- Broken Authentication
- Broken Object Property Level Authorization
- Unrestricted Resource Consumption
- Broken Function Level Authorization
- Unrestricted Access to Sensitive Business Flows
- Server Side Request Forgery
- Security Misconfiguration
- Improper Inventory Management
- Unsafe Consumption of APIs

Why is Authorization a hard problem?



Requirements are granular and diverse



Centralized, Auditable and Orchestratable



No standardization

Why was Data Storage a hard problem?



Requirements are granular and diverse



Centralized, Auditable and orchestratable



No standardization

Google Zanzibar

Requirements are granular and diverse

"Hundreds of client services at Google, including Calendar, Cloud, Drive, Maps, Photos, and YouTube"

Centralized, Auditable and orchestratable

"Useful common infrastructure can be built on top of a unified access control system, in particular, a search index that respects access control"

Standardization

"Zanzibar provides a uniform data model and configuration language for expressing a wide range of access control policies"

- It has maintained 95th-percentile latency of less than 10 milliseconds.
- Availability of greater than 99.999% over 3 years of production use.
- 200 million lookups per second at peak.
- It stores more than two trillion ACLs.

What is OpenFGA?

- OpenFGA is
  - An open-source authorization system
  - Inspired by Google's Zanzibar
  - Designed to provide fine-grained, relationship-based access control.
- Modular Authorization system that can scale for bigger teams
- Community and Governance - CNCF Incubated, Apache 2.0
- Kubernetes Integration - Helm Chart




```
model
  schema 1.1

type user

type team

type folder

type dashboard
```

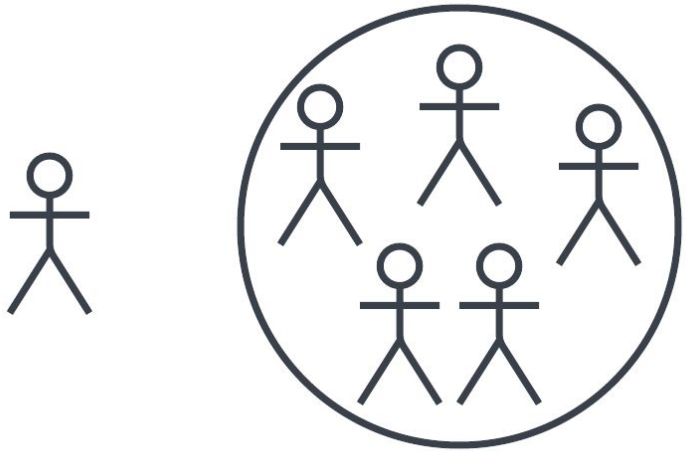
```
model
  schema 1.1

type user

type team

type folder

type dashboard
```



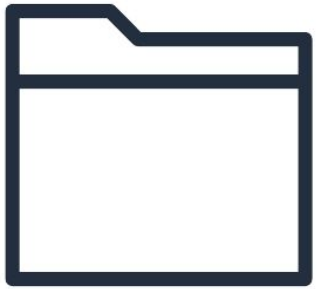
```
model
  schema 1.1

type user

type team

type folder

type dashboard
```





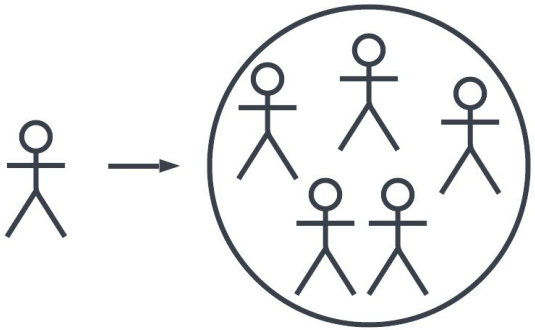
```
model
  schema 1.1

type user

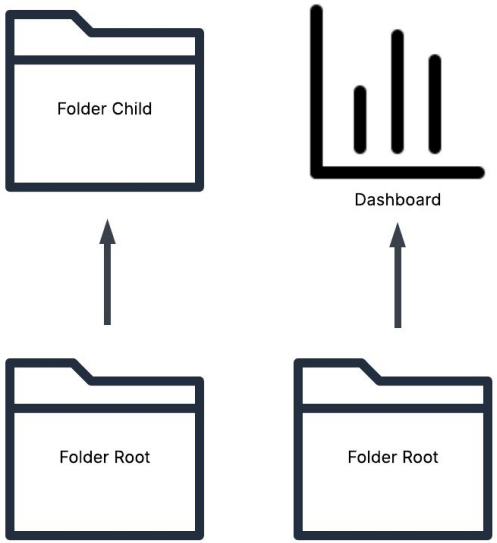
type team
  relations
    define member: [user]

type folder
  relations
    define parent: [folder]

type dashboard
  relations
    define parent: [folder]
```



team:backend#member@user:1



folder:child#parent@folder:root

dashboard:uptime#parent@folder:root

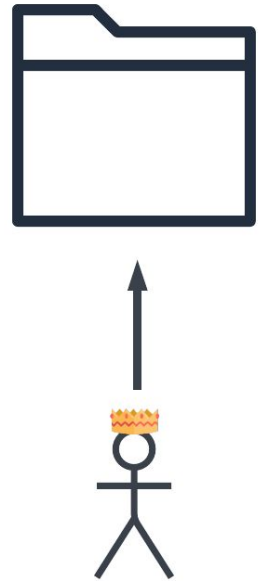
```
model
  schema 1.1

type user

type team
  relations
    define member: [user]

type folder
  relations
    define parent: [folder]
    define admin: [user]

type dashboard
  relations
    define parent: [folder]
```



folder:1#admin@user:1

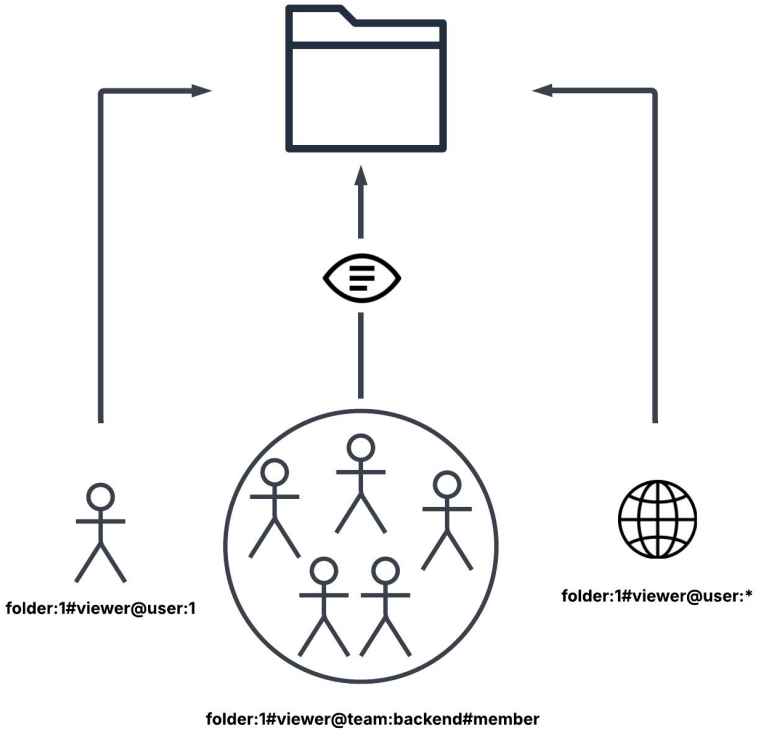
```
model
  schema 1.1

type user

type team
  relations
    define member: [user]

type folder
  relations
    define parent: [folder]
    define admin: [user]
    define viewer: [user, team#member, user:*]

type dashboard
  relations
    define parent: [folder]
    define viewer: [user, team#member, user:*]
```



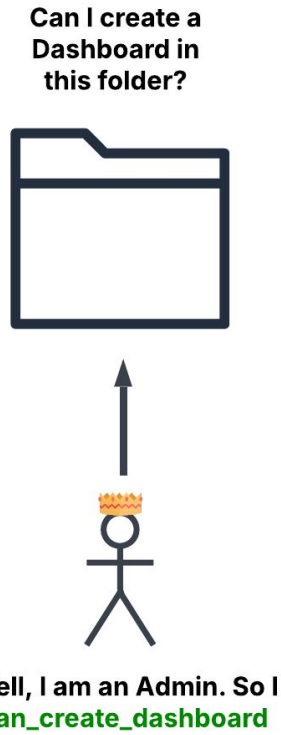
```
model
  schema 1.1

type user

type team
  relations
    define member: [user]

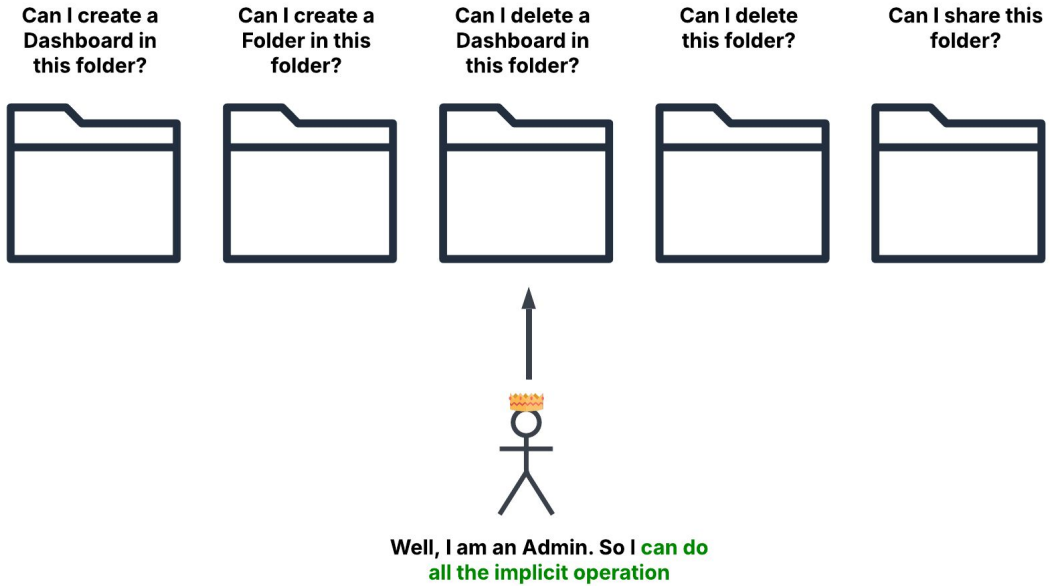
type folder
  relations
    define parent: [folder]
    define admin: [user]
    define viewer: [user, team#member, user:*]
    define can_create_dashboard: admin

type dashboard
  relations
    define parent: [folder]
    define viewer: [user, team#member, user:*]
```



define can_create_dashboard: admin
define can_create_folder: admin
define can_delete_dashboard: admin
define can_delete_folder: admin
define can_share_folder: admin

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....



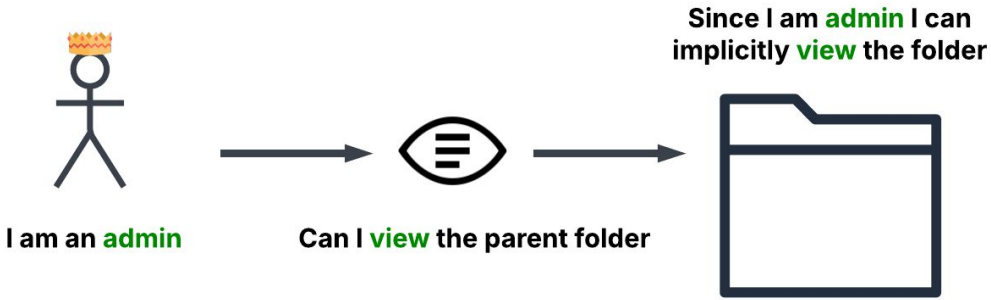
```
model
  schema 1.1

type user

type team
  relations
    define member: [user]

type folder
  relations
    define parent: [folder]
    define admin: [user]
    define viewer: [user, team#member, user:*] or admin (Can also be “and”, “but not”)
    define can_create_dashboard: admin

type dashboard
  relations
    define parent: [folder]
    define viewer: [user, team#member, user:*]
```



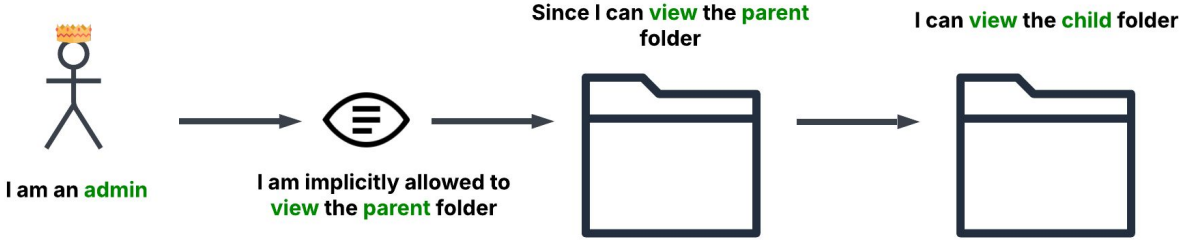
```
model
  schema 1.1

type user

type team
  relations
    define member: [user]

type folder
  relations
    define parent: [folder]
    define admin: [user]
    define viewer: [user, team#member, user:*] or admin or viewer from parent
    define can_create_dashboard: admin or admin from parent

type dashboard
  relations
    define parent: [folder]
    define viewer: [user, team#member, user:*] or viewer from parent
```



```
model
  schema 1.1

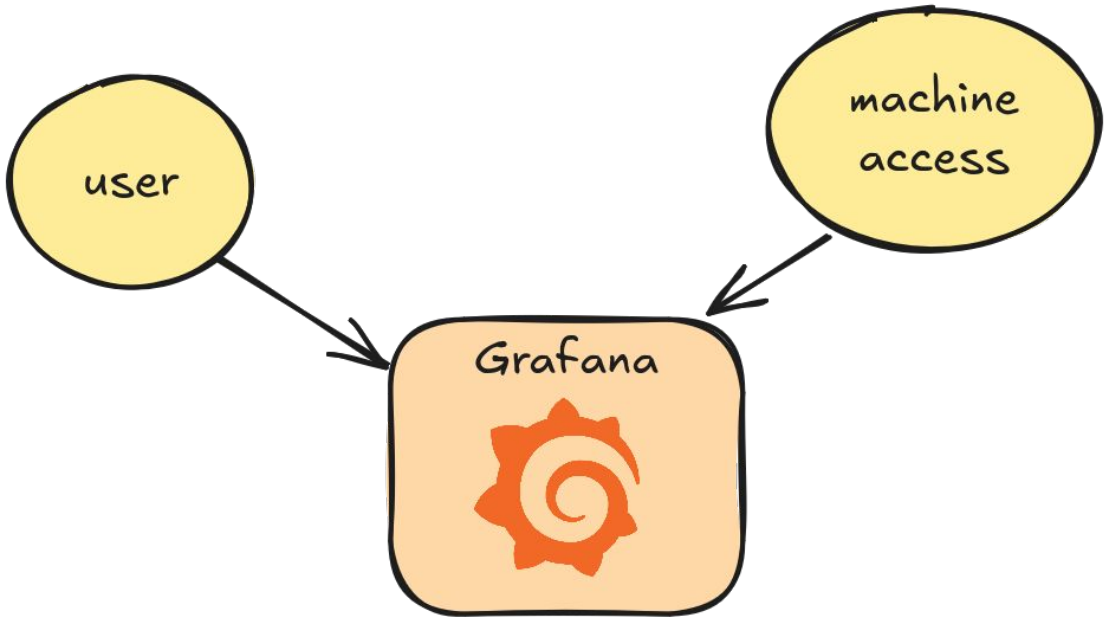
type user

type team
  relations
    define member: [user]

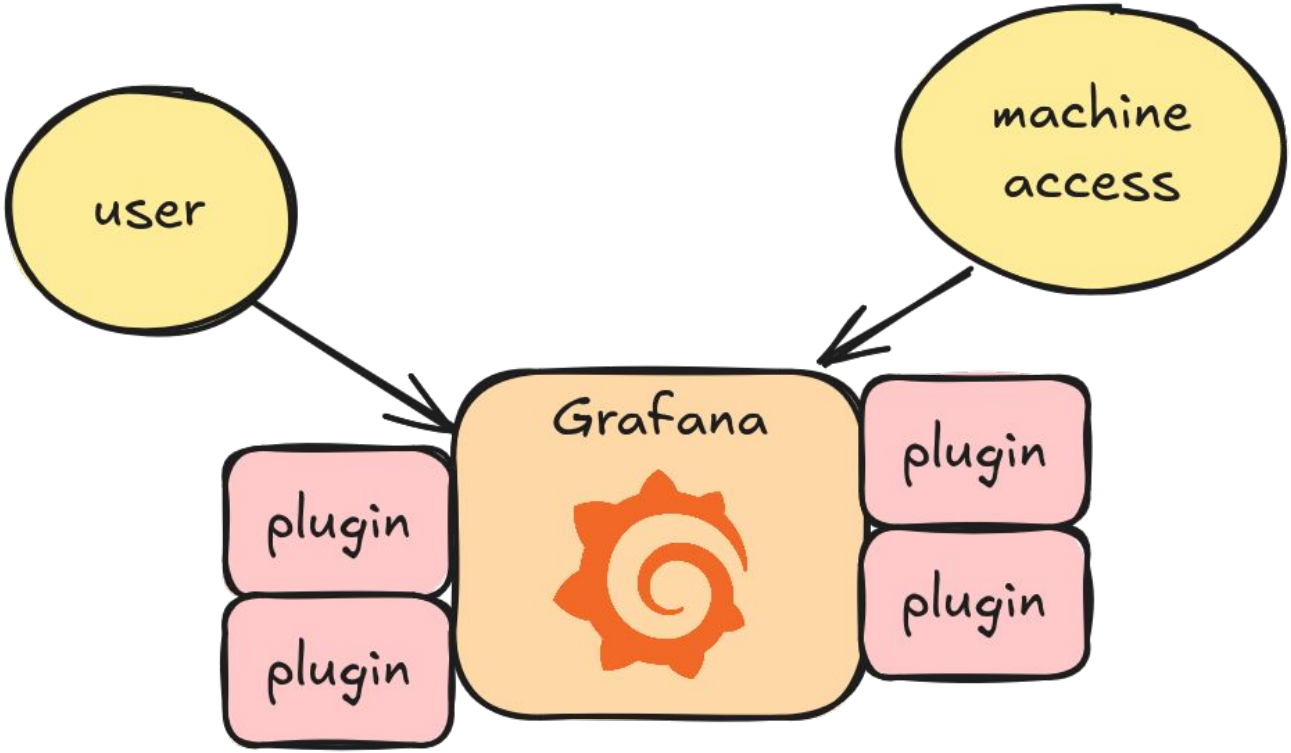
type folder
  relations
    define parent: [folder]
    define admin: [user]
    define viewer: [user, team#member, user:*] or admin or viewer from parent
    define can_create_dashboard: admin or admin from parent

type dashboard
  relations
    define parent: [folder]
    define viewer: [user, team#member, user:*] or viewer from parent
```

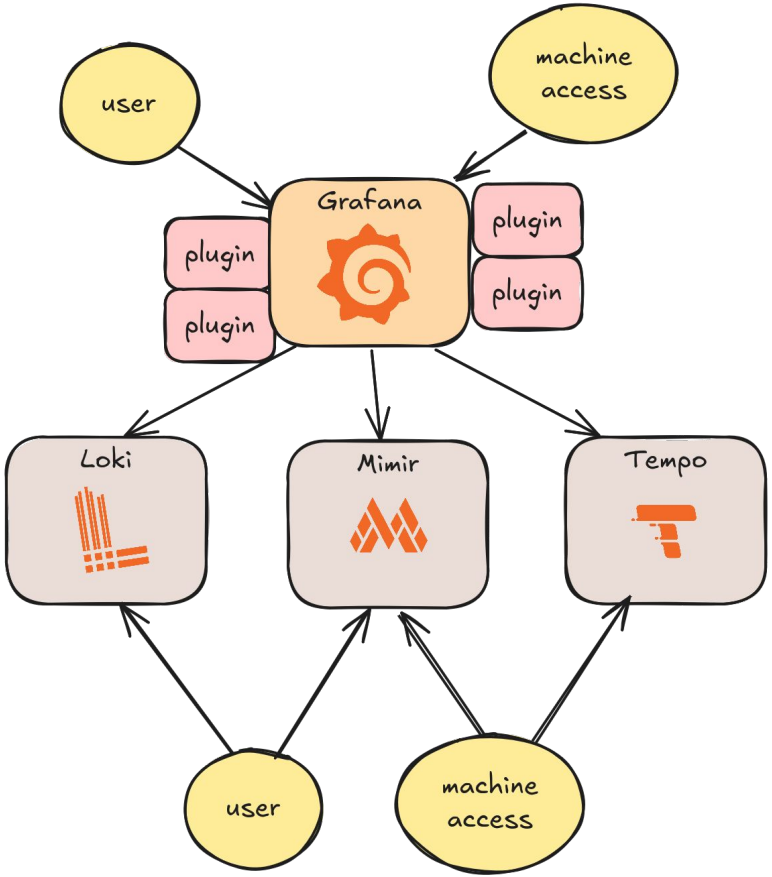
Things were simple in the beginning



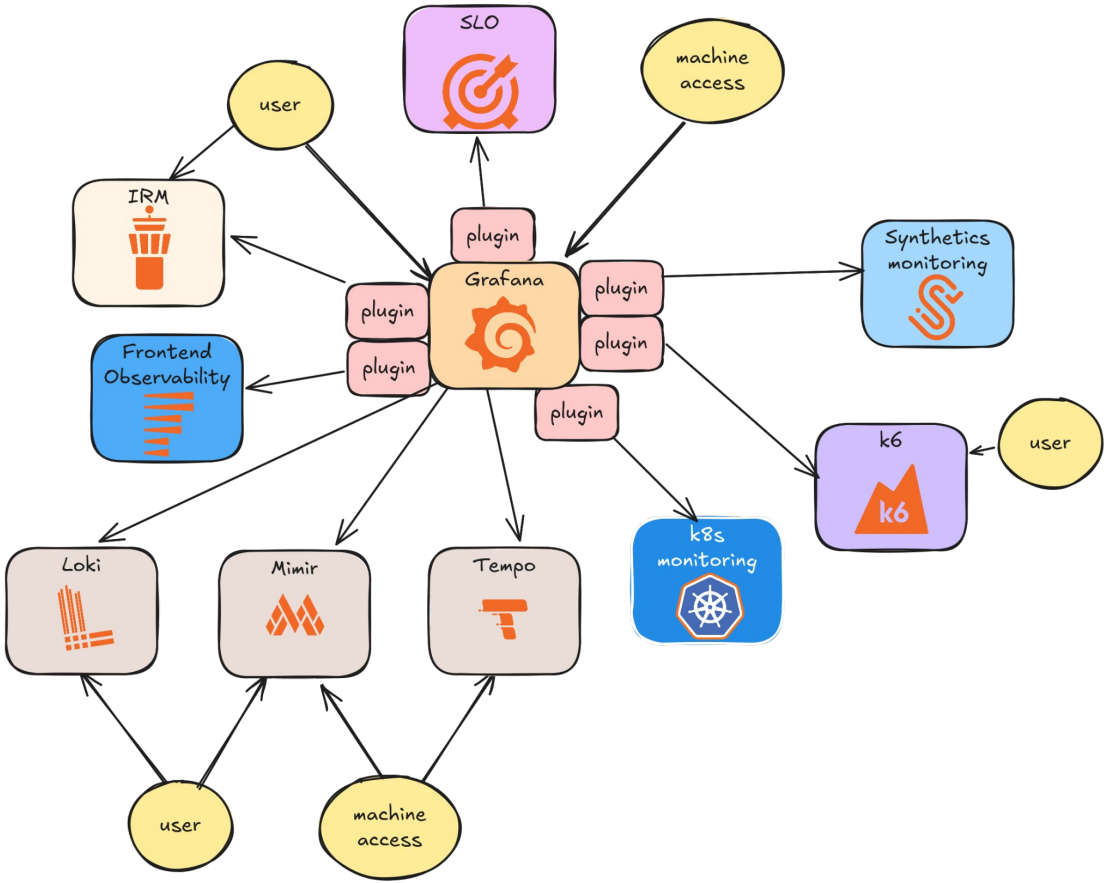
Third party risks



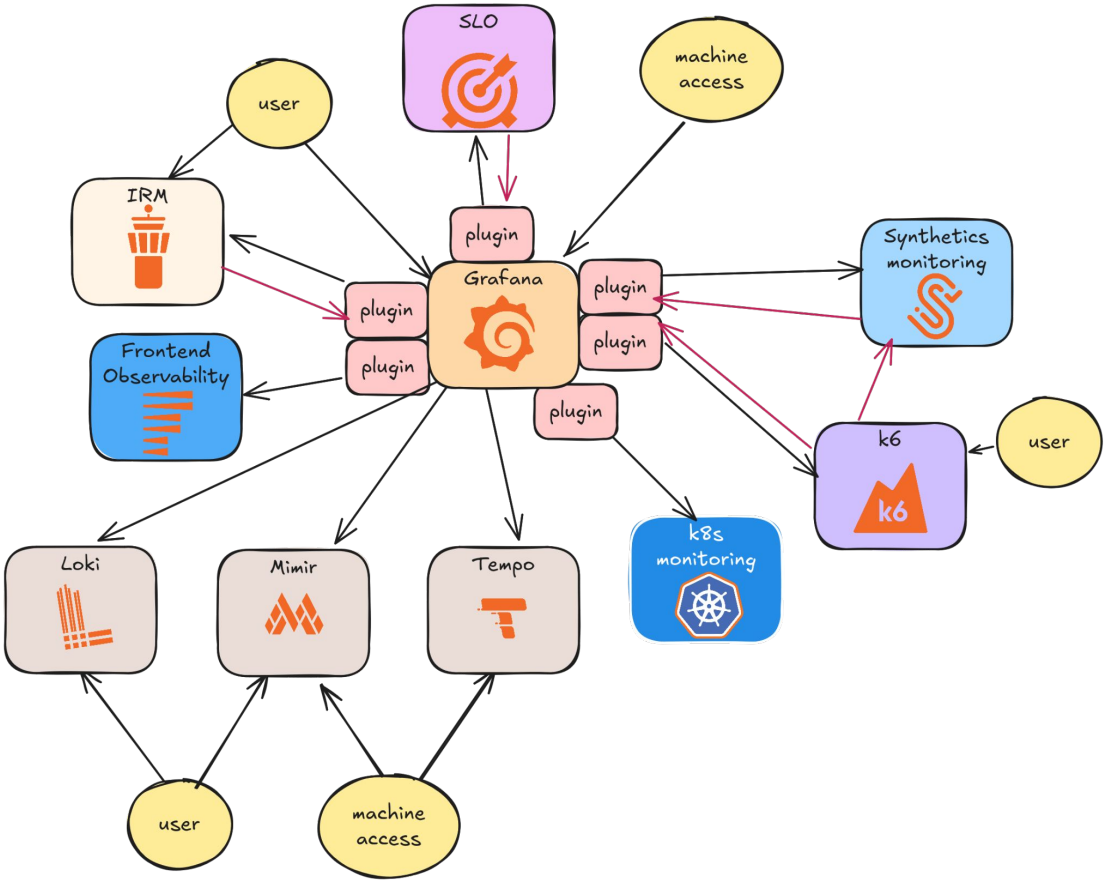
Databases


user
machine access
Grafana
plugin
plugin
plugin
plugin
Loki
Mimir
Tempo
user
machine access

Apps


SLO
machine access
user
IRM
plugin
Grafana
plugin
plugin
plugin
plugin
plugin
plugin
Synthetics monitoring
Frontend Observability
k6
user
k6
k8s monitoring
Loki
Mimir
Tempo
user
machine access

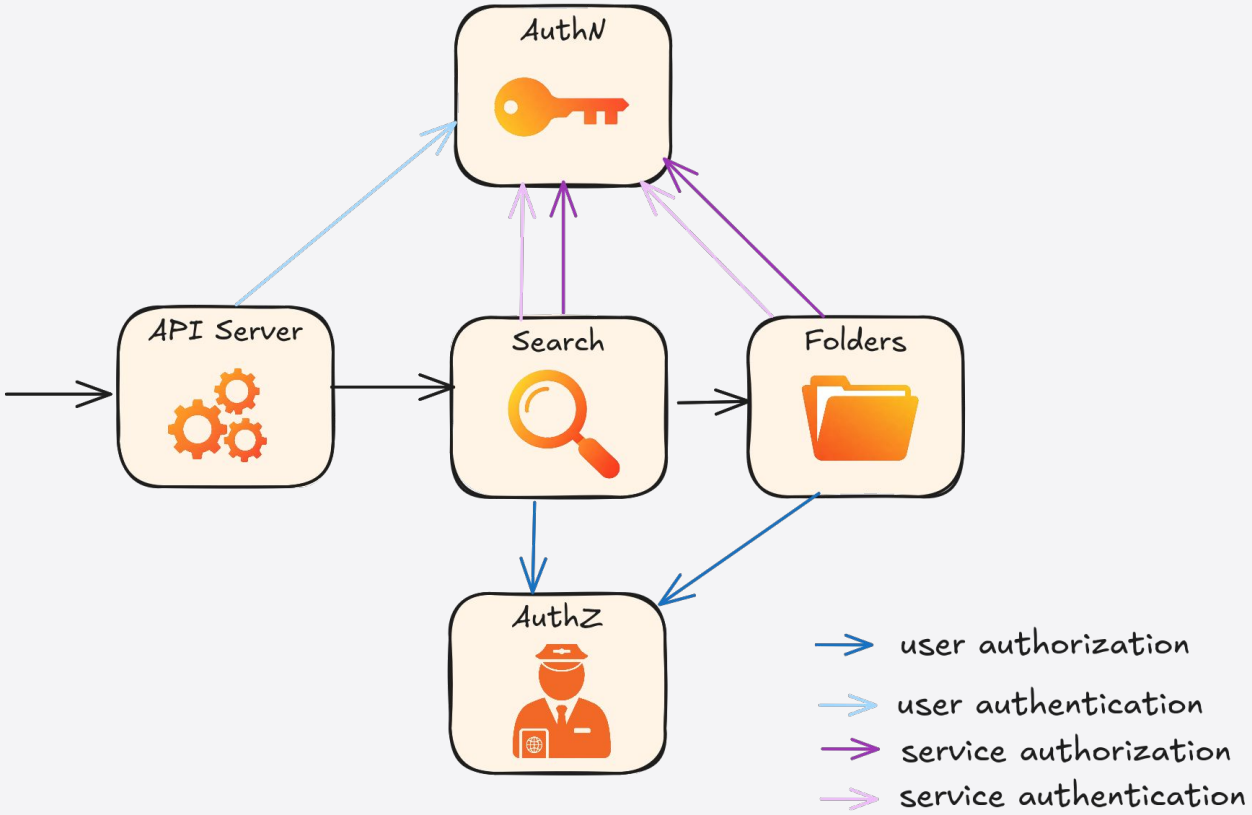
They talk to each other now



The opportunity



Where does openFGA sit in our architecture



Our identities

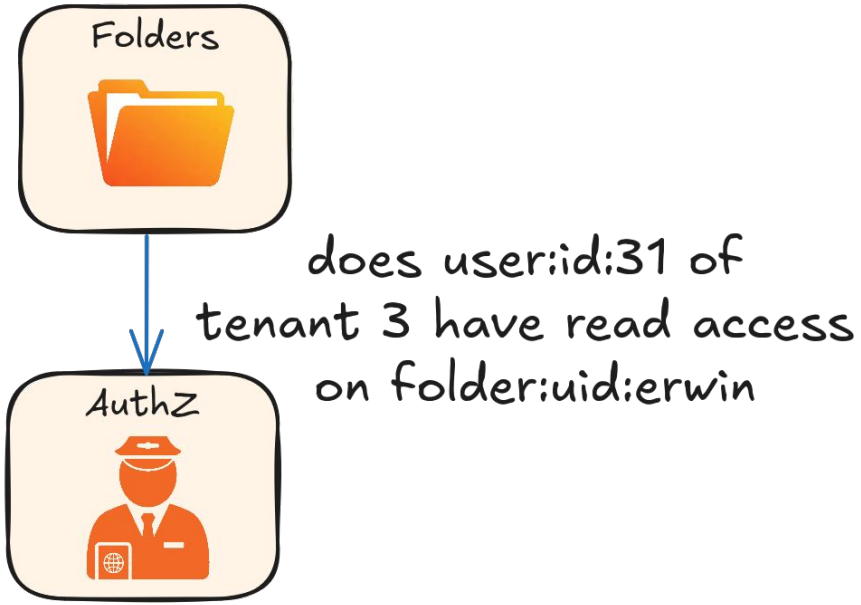


ID Tokens

```
{
  "aud": "stack:1",
  "authenticatedBy": "okta_oauth",
  "exp": 1710261916,
  "iat": 1710261316,
  "iss": "52a33119-545c-496c-8e7d-1f117bde9171",
  "jti": "c45eee0d-6e51-45e2-9368-16d55ffd41e6",
  "org_id": "0",
  "sub": "user:6"
}
```

Access Tokens

```
{
  "aud": "search_service",
  "exp": 1710261916,
  "iat": 1710261316,
  "iss": "52a33119-545c-496c-8e7d-1f117bde9171",
  "jti": "c45eee0d-6e51-45e2-9368-16d55ffd41e6",
  "org_id": "0",
  "sub":
"access-policy:c45eee0d-6e51-45e2-9368-16d55ffd4
1e6"\
  "scopes":
["incidents:write","access-token:sign"],
  "permissions": ["datasources:read",
“datasources:write”],
  "delegatedPermissions": ["datasources:read"],
}
```


Folders
does user:id:31 of
tenant 3 have read access
on folder:uid:erwin
AuthZ



Define 'easy'

Migrating to OpenFGA in 3 easy steps

Schema

Data Migration

Data Retrieval

Our entities



- Library panels
- Dashboards
- API keys
- Datasources
- Service Accounts
- Alert Rules
- Alert Silences
- Alert Notifications
- Alert Receivers
- Support bundles
- External Alert Notification
- External Alert Instance
- LDAP Configs
- OAuth Configs
- SAML Configs
- Settings
- Orgs
- Teams
- Roles
- Users
- User Quotas
- Feature Toggles
- Folders
- Plugin installs
- ...

Globals



- Library panels
- Dashboards
- API keys
- Datasources
- Service Accounts
- Alert Rules
- Alert Silences
- Alert Notifications
- Alert Receivers

- Support bundles
- External Alert Notification
- External Alert Instance
- LDAP Configs
- OAuth Configs
- SAML Configs
- Settings
- Orgs

- Teams
- Roles
- Users
- User Quotas
- Feature Toggles
- Folders
- Plugin installs
- ...

Tenant scoped

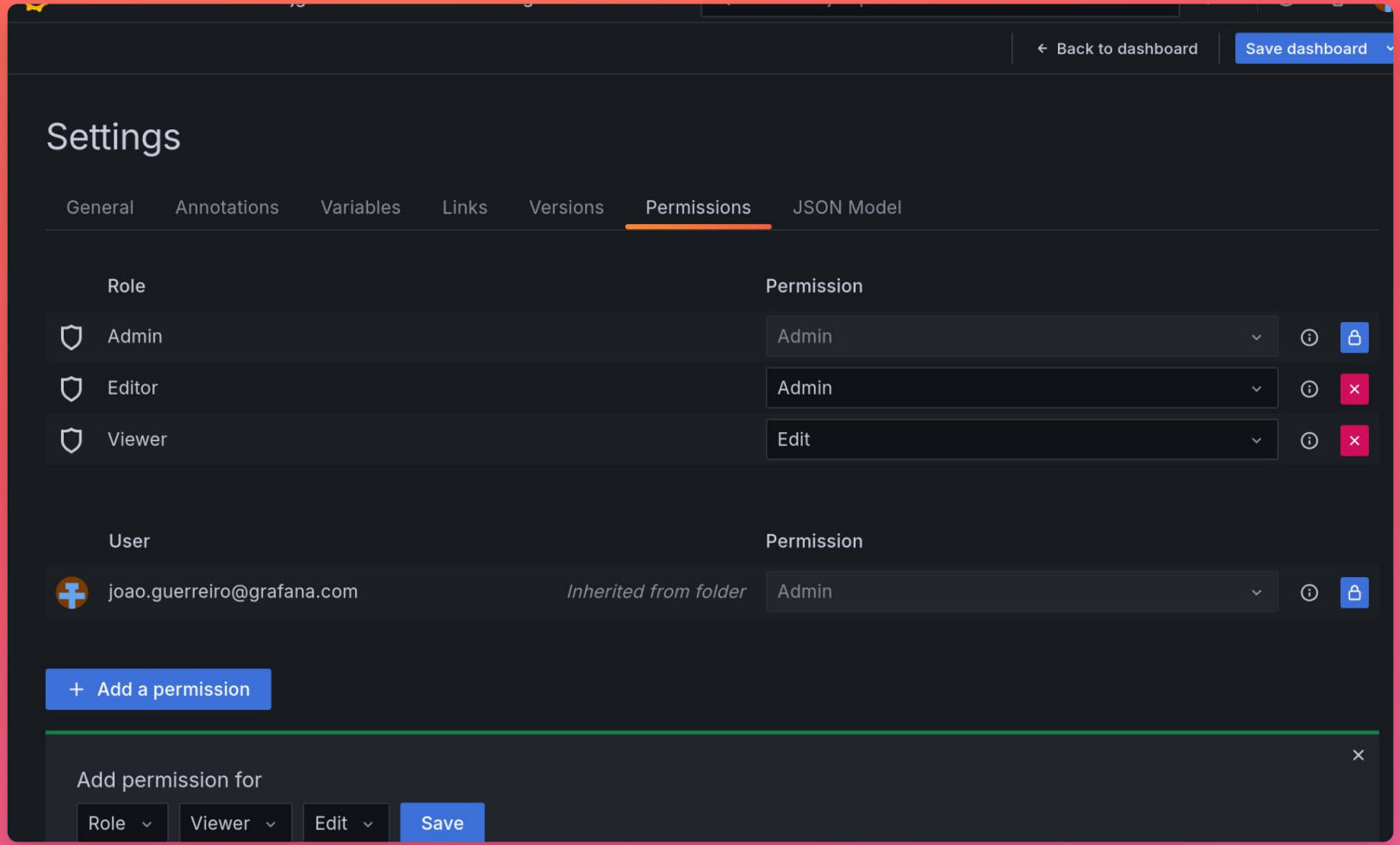


- Library panels
- Dashboards
- API keys
- Datasources
- Service Accounts
- Alert Rules
- Alert Silences
- Alert Notifications
- Alert Receivers
- Support bundles
- External Alert Notification
- External Alert Instance
- LDAP Configs
- OAuth Configs
- SAML Configs
- Settings
- Orgs
- Teams
- Roles
- Users
- User Quotas
- Feature Toggles
- Folders
- Plugin installs
- ...

Folder resources



- Library panels
- Dashboards
- API keys
- Datasources
- Service Accounts
- Alert Rules
- Alert Silences
- Alert Notifications
- Alert Receivers
- Support bundles
- External Alert Notification
- External Alert Instance
- LDAP Configs
- OAuth Configs
- SAML Configs
- Settings
- Orgs
- Teams
- Roles
- Users
- User Quotas
- Feature Toggles
- Folders
- Plugin installs
- ...


Back to dashboard
Save dashboard
Settings
General
Annotations
Variables
Links
Versions
Permissions
JSON Model
Role
Permission
Admin
Admin
Editor
Admin
Viewer
Edit
User
Permission
joao.guerreiro@grafana.com
Inherited from folder
Admin
Add a permission
Add permission for
Role
Viewer
Edit
Save

Caution: Potholes

Number of relations to write

Schema complexity affects performance

Database differences



Help OpenFGA help us

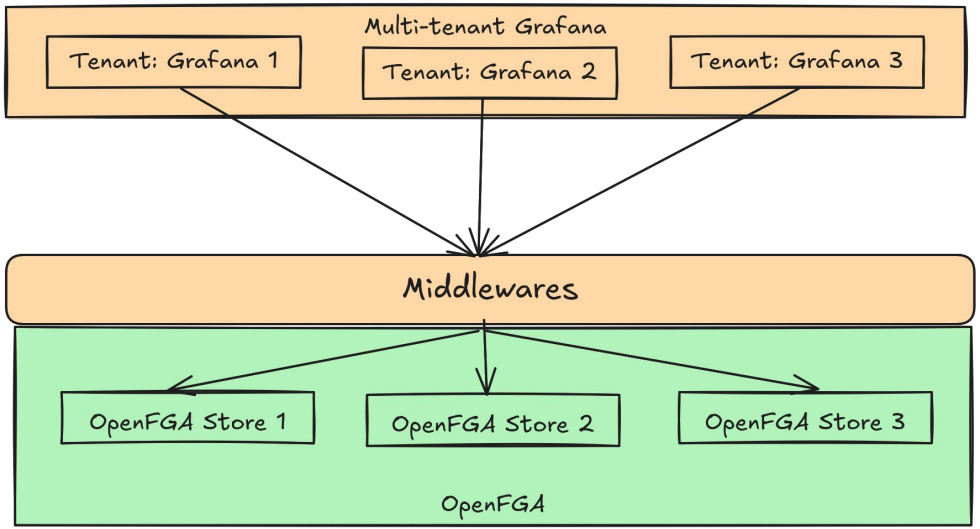
- Schematizing our access control helped us find inconsistencies in our UX
- Revising the need for granularity for better performance
- Reducing the amount of access control verbs
- Reducing the freedom of verb declaration

```
{
  "role": {
    "name": "Probes reader",
    "description": "Read probes in the Synthetic Monitoring app",
    "permissions": [
      { "action": "plugins.app:access", "scope": "plugins:id:grafana-synthetic-monitoring-app" },
      { "action": "grafana-synthetic-monitoring-app:read" },

      { "action": "grafana-synthetic-monitoring-app.probes:read" }
    ]
  },
  "grants": ["Viewer"]
},
```

Now do it multitenant

- Tenant data is separated by stores.
- Middlewares include
  - Authentication (Calling service identity)
  - Tenant store mapping
  - Customer facing permission mapping to OpenFGA format



We're no strangers to migrations



2014

Orgs and Org Roles

Grafana 1.0

2018

Dashboard/Folder permissions

Grafana 5.0

2021

FGAC

Grafana 8.0

2022

RBAC

Grafana 9.0

What we learned from those

Feature toggles will be toggled

Dual-write always
Behavior changes should be done in both pathways

Upgrade/ Downgrade cycles

Don’t trust one time migrations.
Verify the state often between the two stores

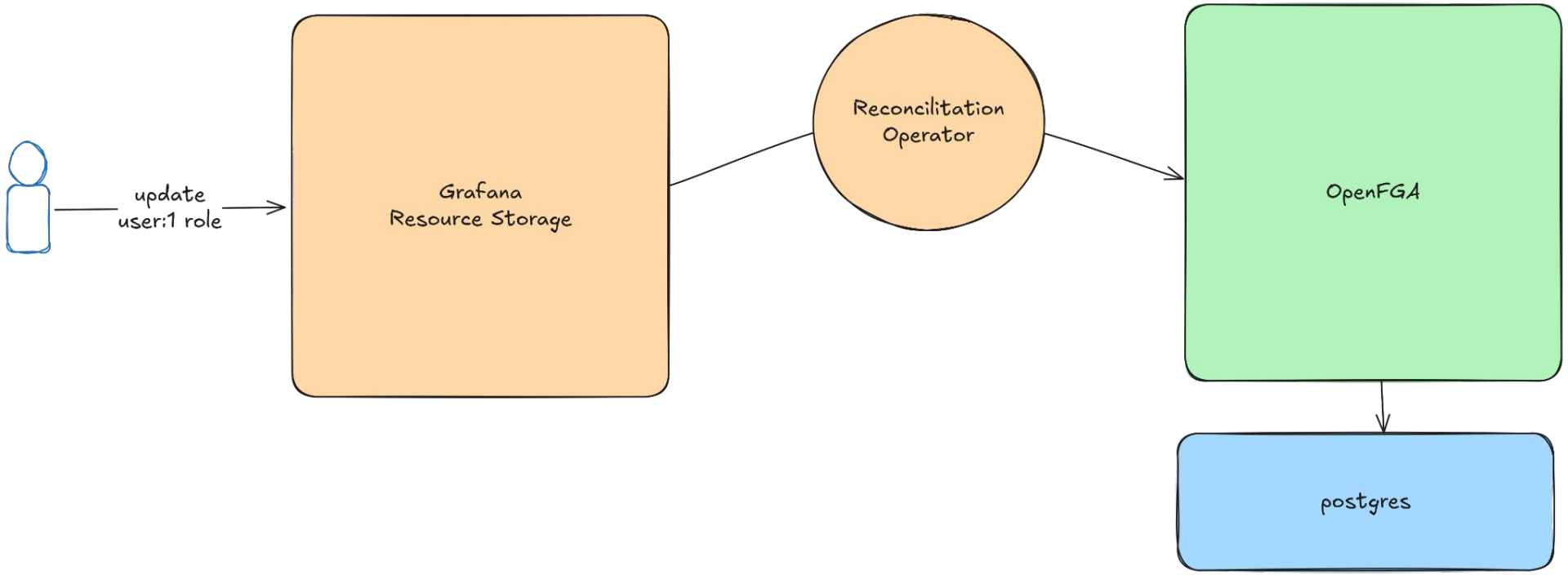
Don’t rush

Migrations take time, ensure legacy works but doesn’t expand



Permission Reconciliation loop

KubeCon CloudNativeCon Europe 2025

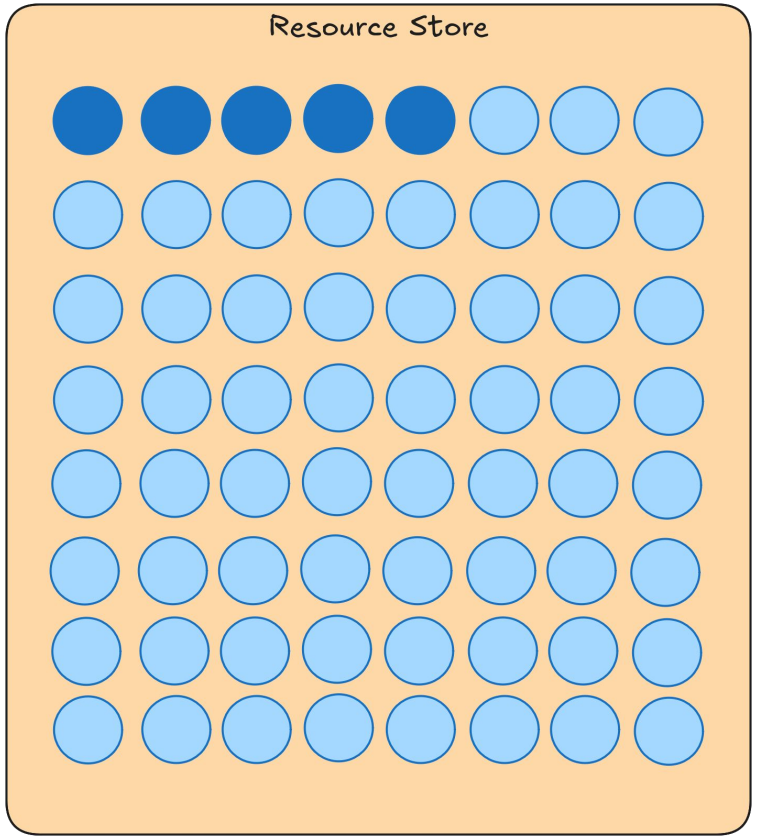


Searching Dashboards



AI generated image (look at the header buttons on the two sides)

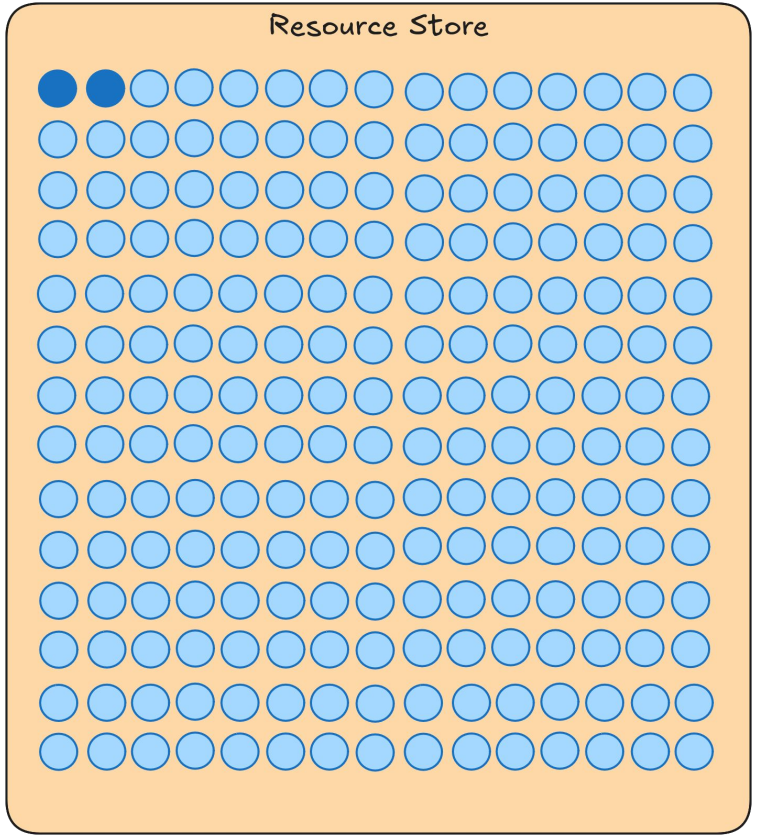
Search results return is low



KubeCon CloudNativeCon Europe 2025

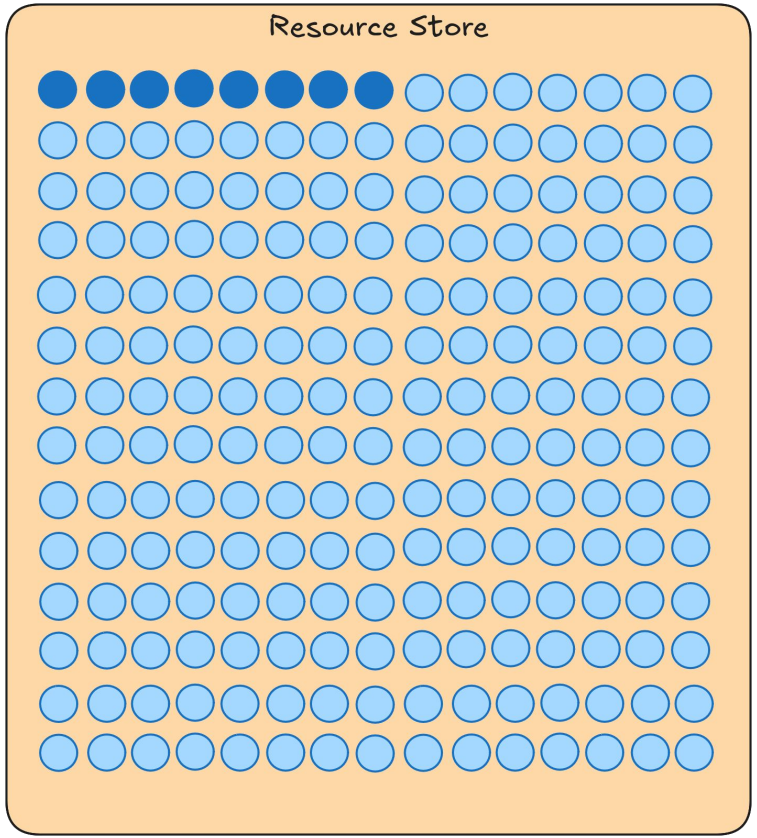
Search then Check will be the easiest here

High Volume, Low Access



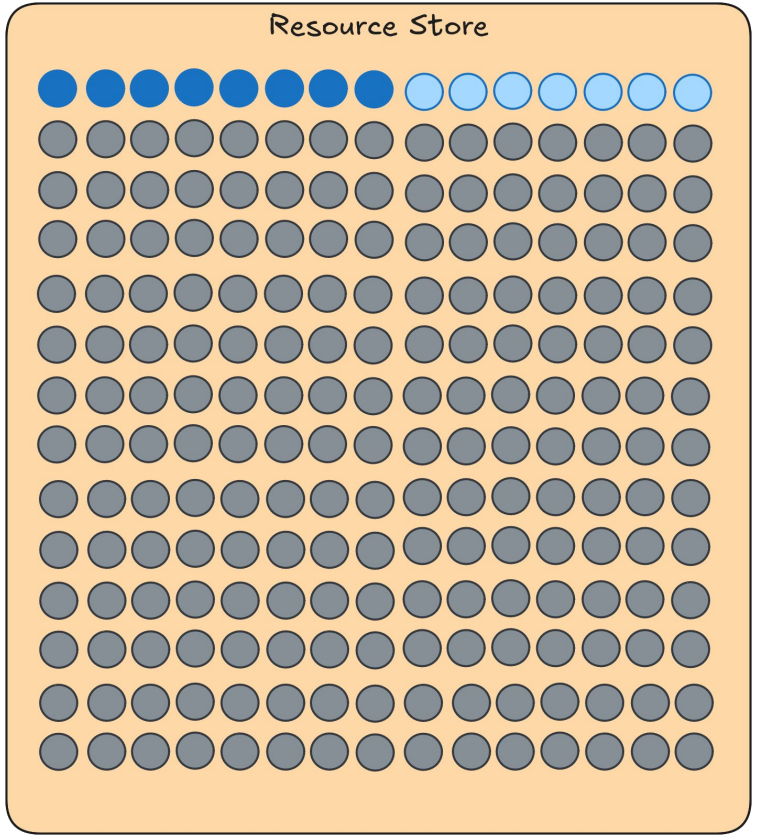
User can access is ~1000. Filter by IDs

High Volume, Low Access+



Local index, then check

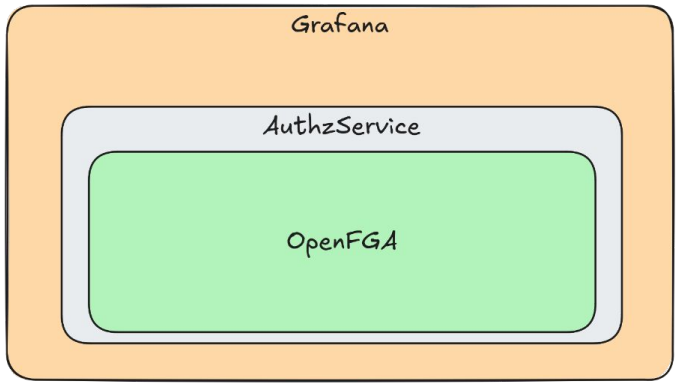
Practical approach: Filter more

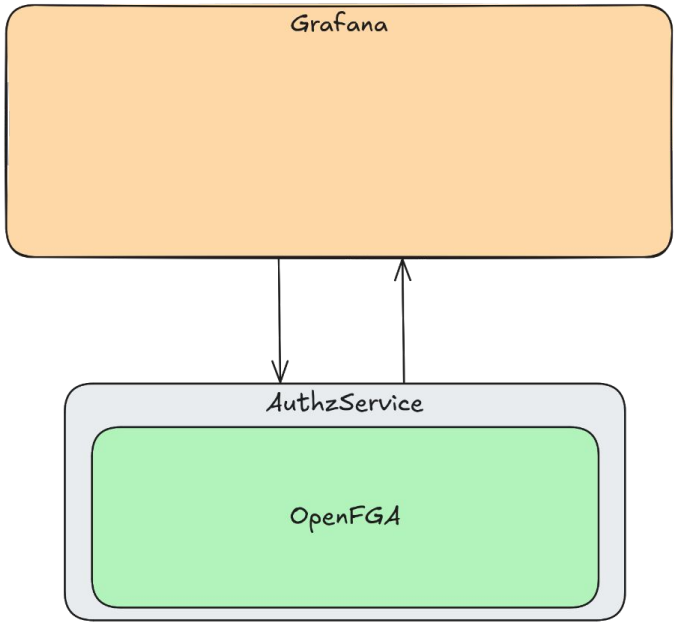


Search then Check will work here

Packaging it all up

- We support separate deployment modes using grafana/dskit
- Grafana with OpenFGA will be available for use in OSS soon improving as-code flows for authorization.





Thanks for listening!

Grafana OpenFGA implementation

OpenFGA Documentation

Have more questions?

Join us at
#openfga in CNCF slack

Jguer

poovamraj