GitOps-Driven Platform Engineering:

Mastering Cluster Fleet Management

Michael Fornaro

Staff DevOps Engineer

Olga Mirensky

Lead Platform Engineer

GDG DevFest Melbourne 2025

$ WHOAMI

Michael Fornaro

Staff DevOps Engineer

Easygo

Olga Mirensky

Lead Platform Engineer

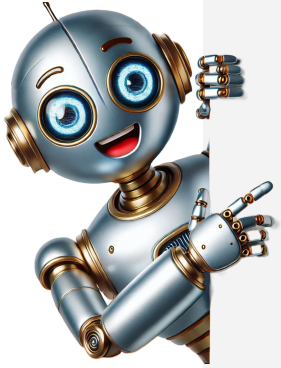
Macquarie Group

Backstory

Stepping back...

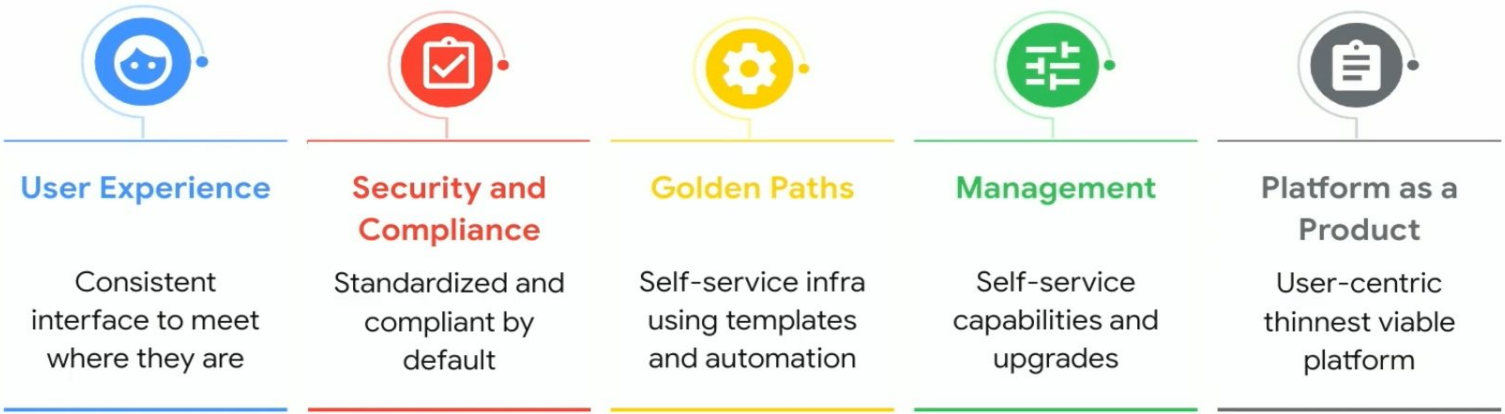
Cloud is complicated!

Teams are swamped.

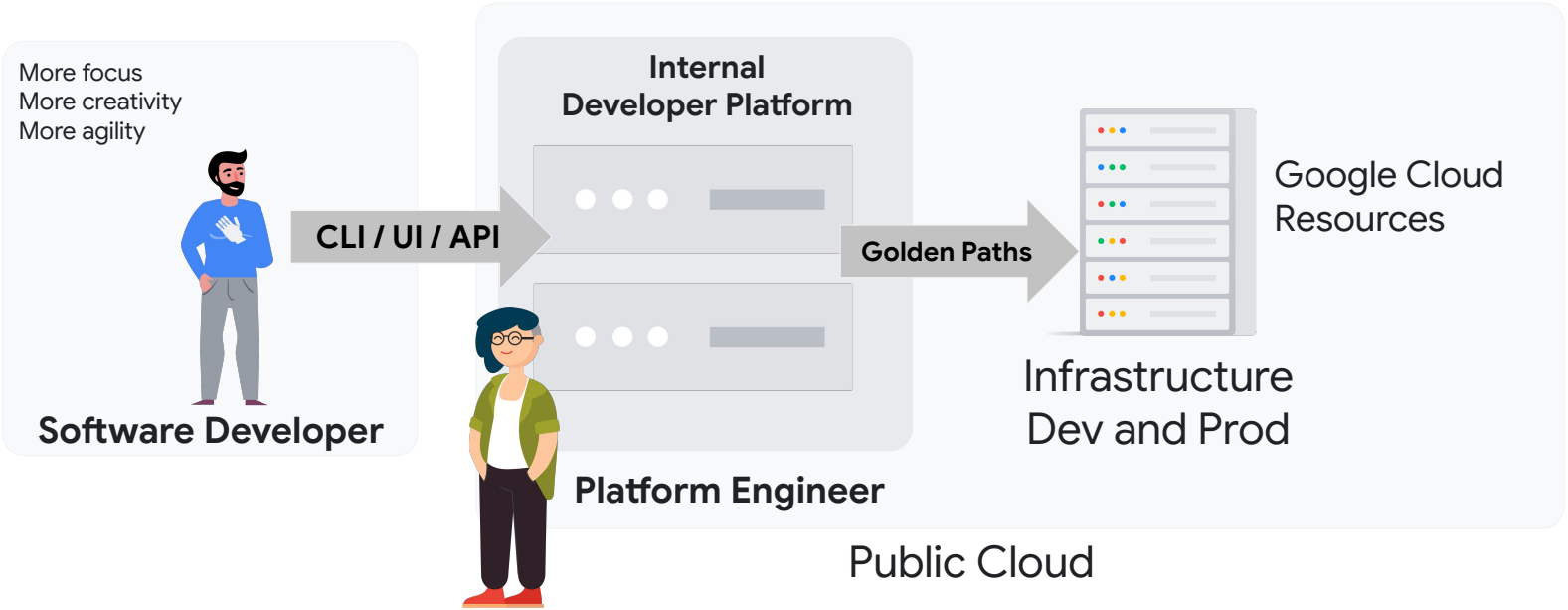


Platforms can help.

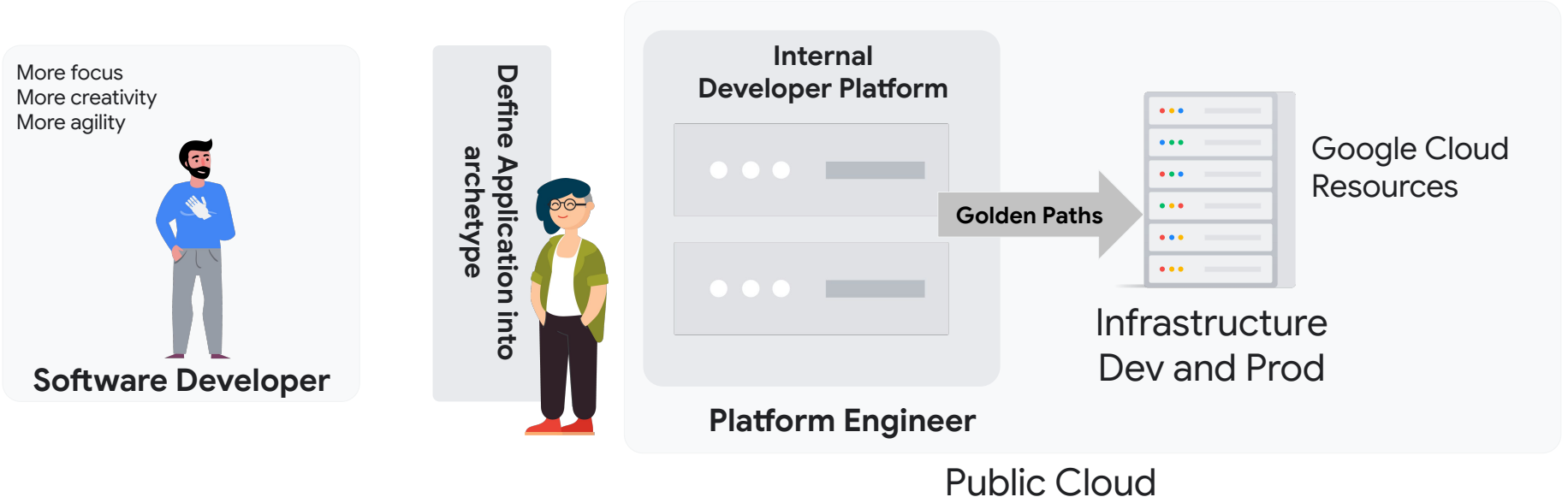
What is Platform Engineering ?!



Platform Engineering



Platform Engineering



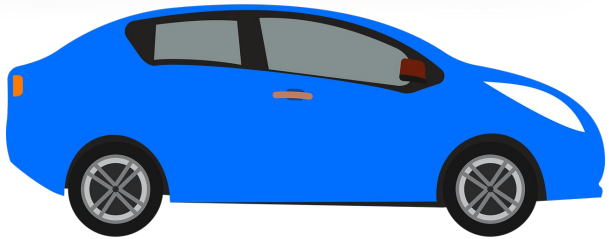
Archetypes Introduction

Let's Go: **Car Shopping**

"I need a new <vehicle>!"

Really:

A. I want to **go fast**
B. I need to carry **lots of stuff**
C. I don't want to **spend** much



Refinement

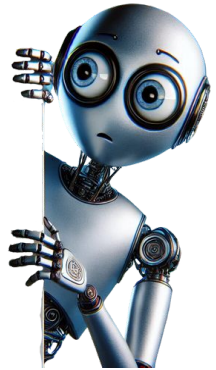
Once you pick a **type**, you can choose a **make** and **model**.

Customization

Then, you choose the **one for you**.

Maybe you make it even **more** for you.

NOT Car Shopping!



Too Much Choice can be Disastrous

What we think we want

vs

How it actually turns out

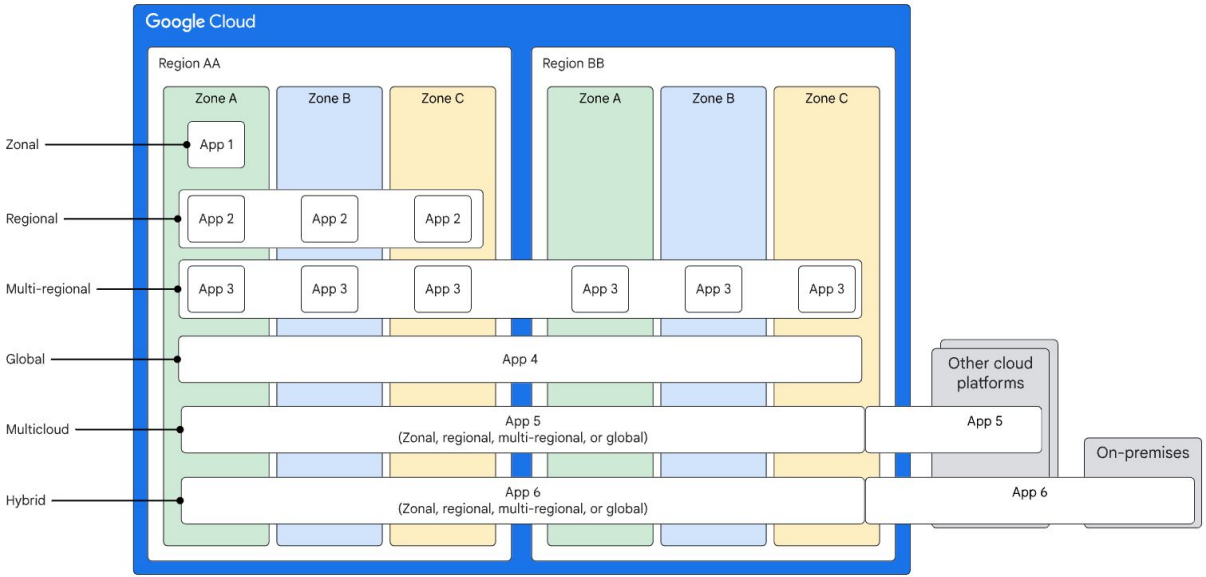
https://simpsons.fandom.com/wiki/The_Homer

"the type of car Americans really want, not the kind we tell them they want"

A deployment archetype is an abstract, provider-independent model that you use as the foundation to build application-specific *deployment architectures* that meet your business and technical requirements

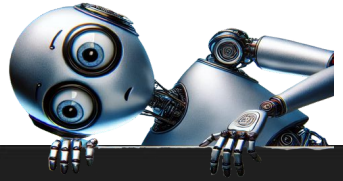
goo.gle/app-archetypes

Deployment Archetypes



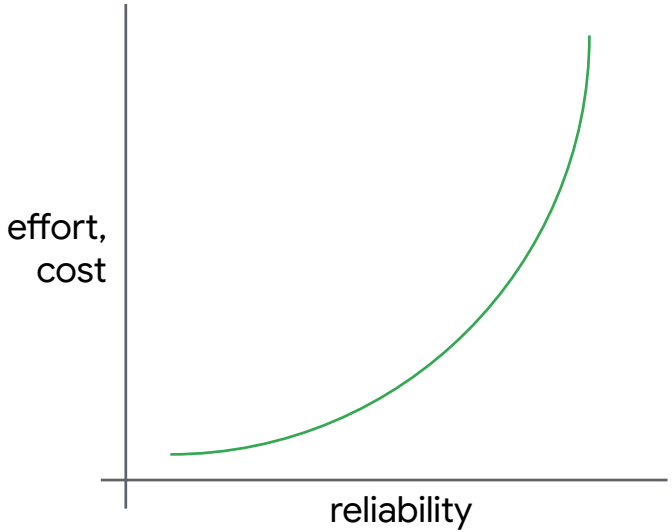
goo.gle/app-archetypes

From Abstract Model to Implementation



achetype.yaml

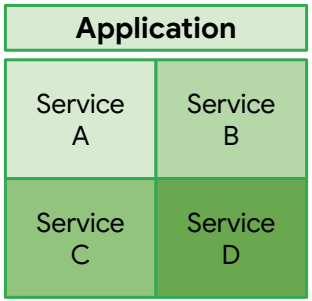
```yaml
---
apiVersion: company/apps/v1alpha1
kind: Application
metadata:
  name: whereami-frontend # Namespace, Service, Deployment, KSA, GSA, and repo names based on this.
spec:
  workload:
    application-group: "whereami" # Multiple Services belong to an application-group
    template: "golang" # Supported values are golang, python, java, javascript
    deployment-strategy: "global" # Strategies supported are multi-zonal, multi-regional, and global
    # Supported regions are us-west2, us-central1 (examples)
    # The "global" strategy ignores this field
    locations: ["us-west2", "us-central1"]
  service:
    ports: ["8080", "443"]
  config: ···
  traffic: ···
  security: ···
  access: ···
  deployment:
    dockerfile: "/path/to/Dockerfile" # Path to the Dockerfile.
    strategy: "canary" # Supported values are Rolling, Blue-Green and Canary, default is Rolling
    frequency: "daily" # Supported values are Hourly, Daily, Weekly, Fortnightly, Monthly, Quarterly, default is Daily
  infra:
    gcp:
      databases: ["sql","spanner"] # Supported values are "sql", "firestore", "spanner", "redis"
  requirements:
    availability: "99.999%"  # Target uptime percentage.
    latency: "200ms"  # Maximum acceptable response time.
    errorRate: "0.5%"  # Target maximum error rate percentage.
    throughput: "5000qps"  # Queries per second.
    durability: "99.99%"  # Data durability percentage.
    recoveryTimeObjective: "30m"  # Maximum time to recover services.
    recoveryPointObjective: "5m"  # Maximum acceptable data loss period.

```


effort,
cost
reliability

How to use Archetypes?

Services can be deployed to a **single archetype**

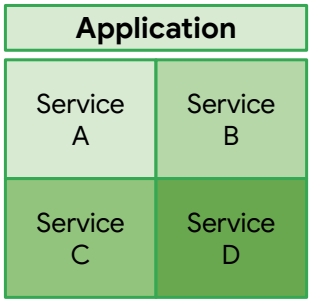
Applications can use services across **multiple archetypes**



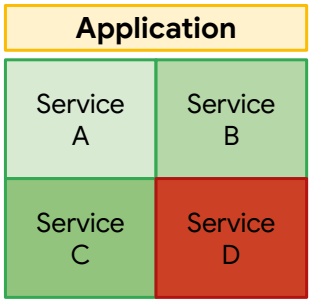
How to use Archetypes?

Services can be deployed to a **single archetype**

Applications can use services across **multiple archetypes**

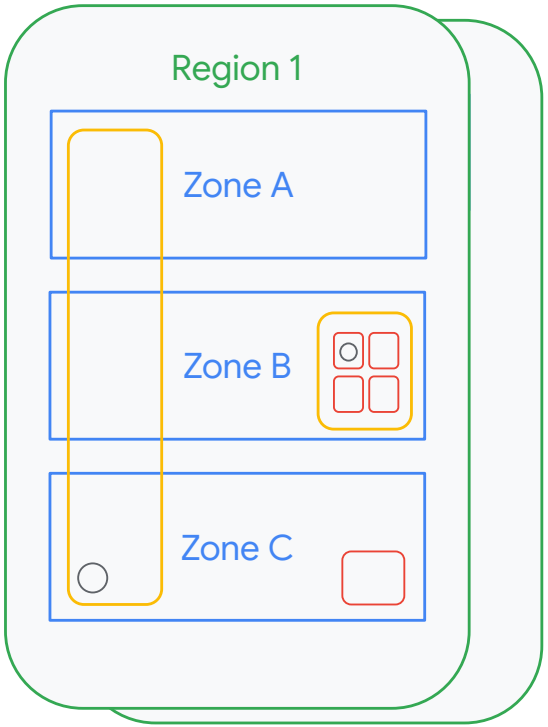


Applications should be designed for **graceful degradation**

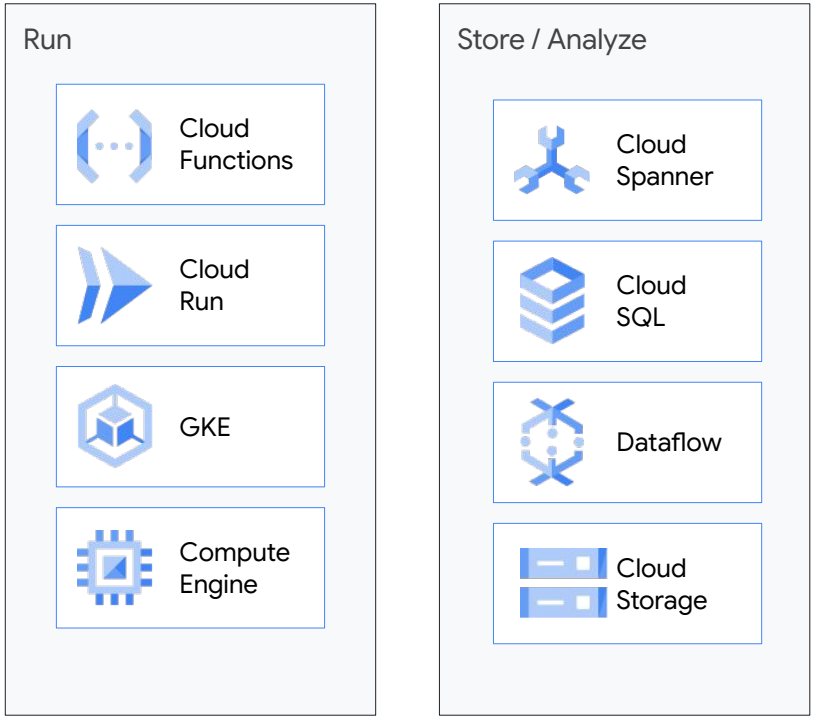


Cloud Failure Domains: Regions, Zones

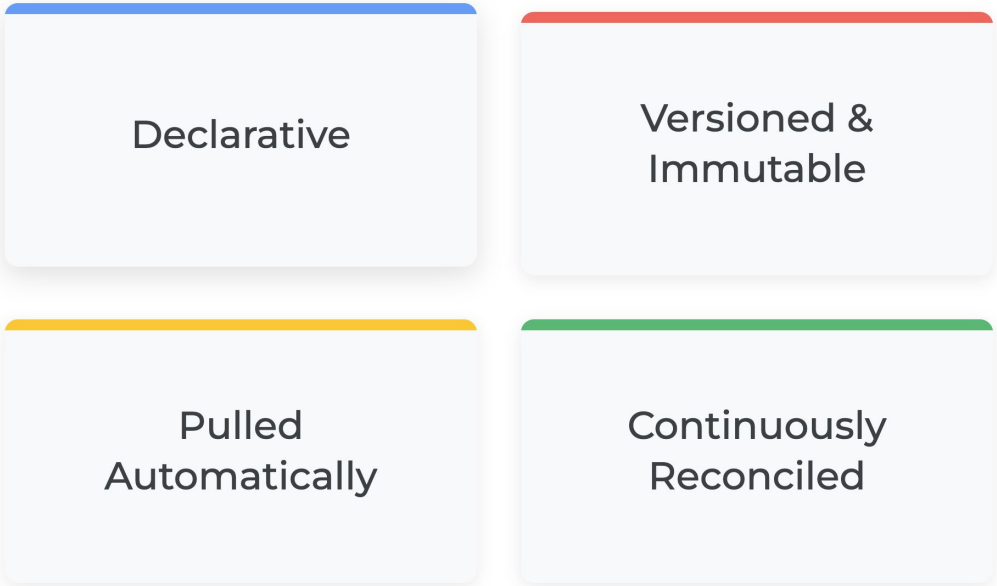
Clusters, Pods, Apps, Functions, VMs



Products expose their failure domains differently

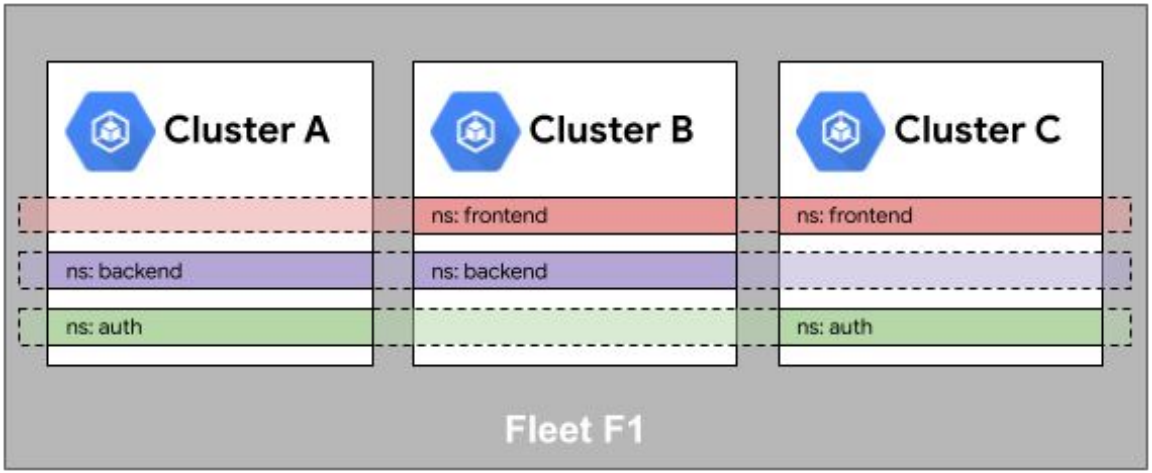


What is GitOps?



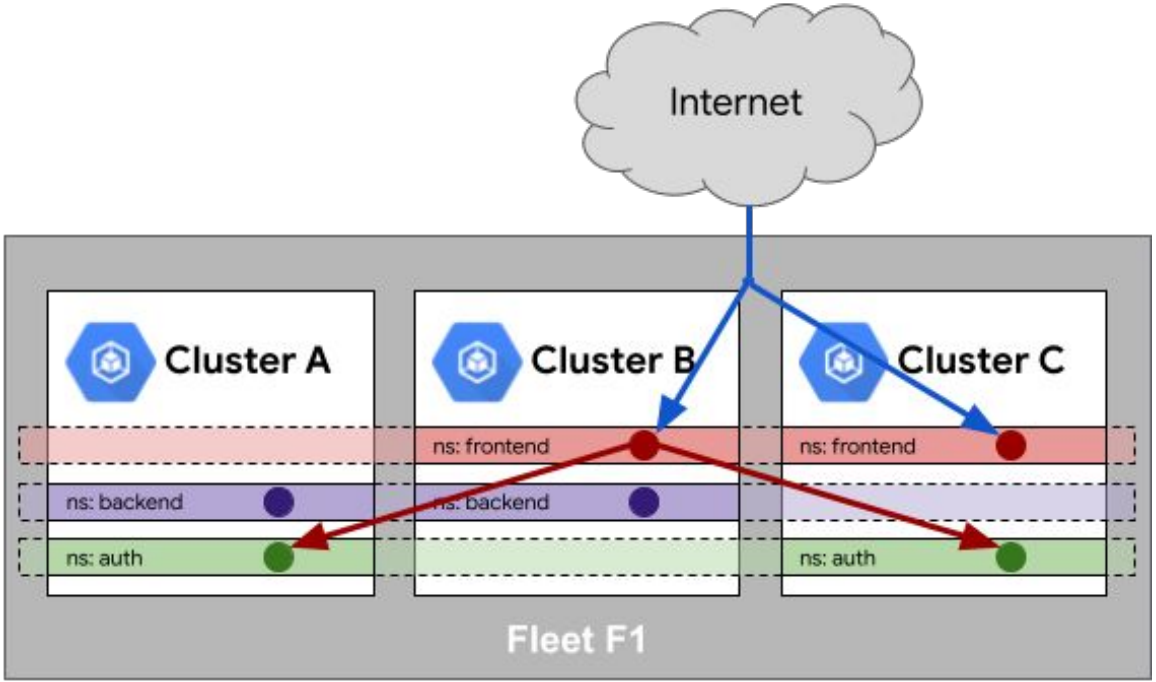
https://opengitops.dev/

What are Fleets of Clusters?



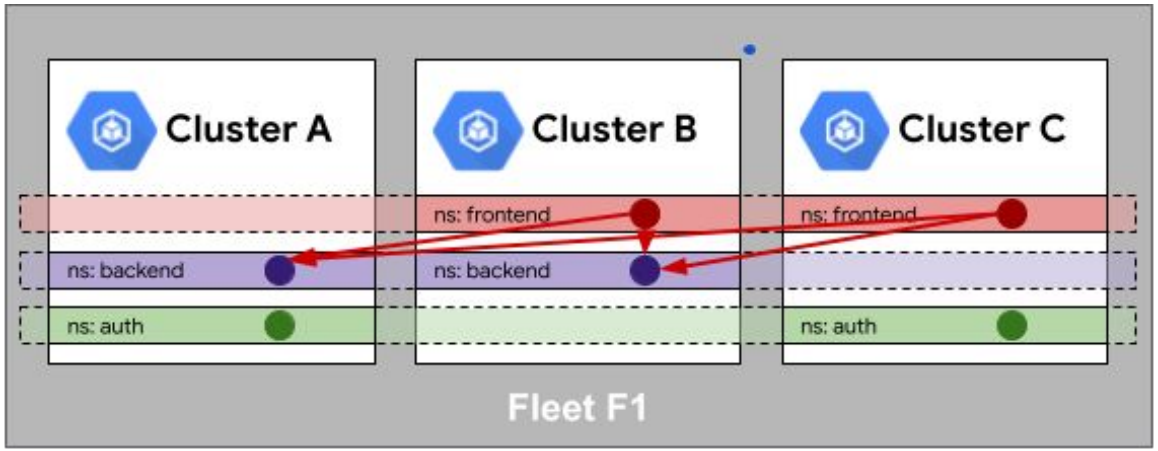
Namespace sameness in a fleet

What are Fleets of Clusters?



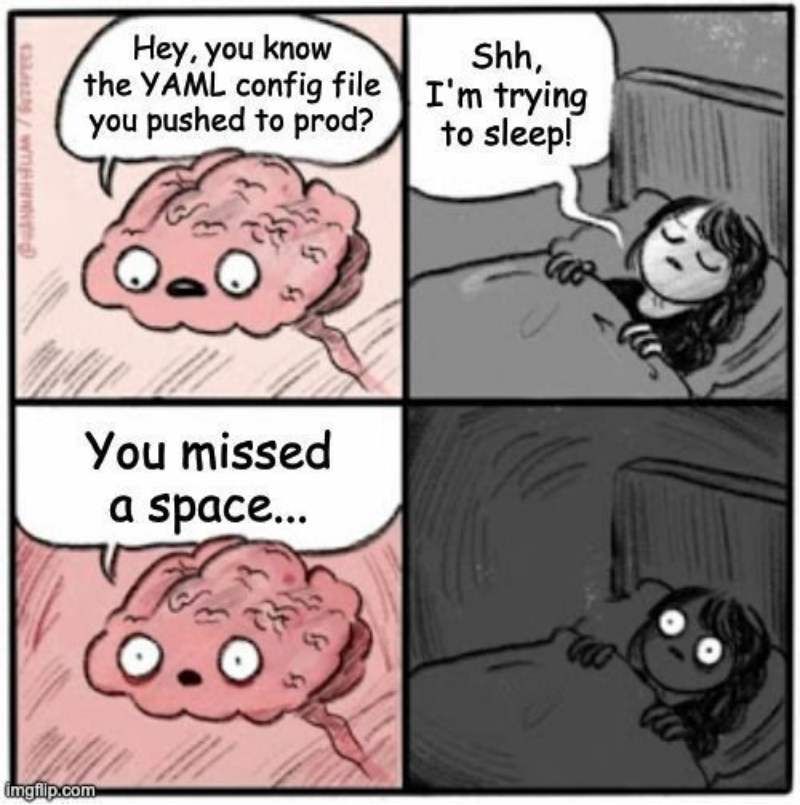
Service sameness in a fleet

What are Fleets of Clusters?



Identity sameness inside a fleet

Mastering Cluster Fleet Management



Demo

Demo video

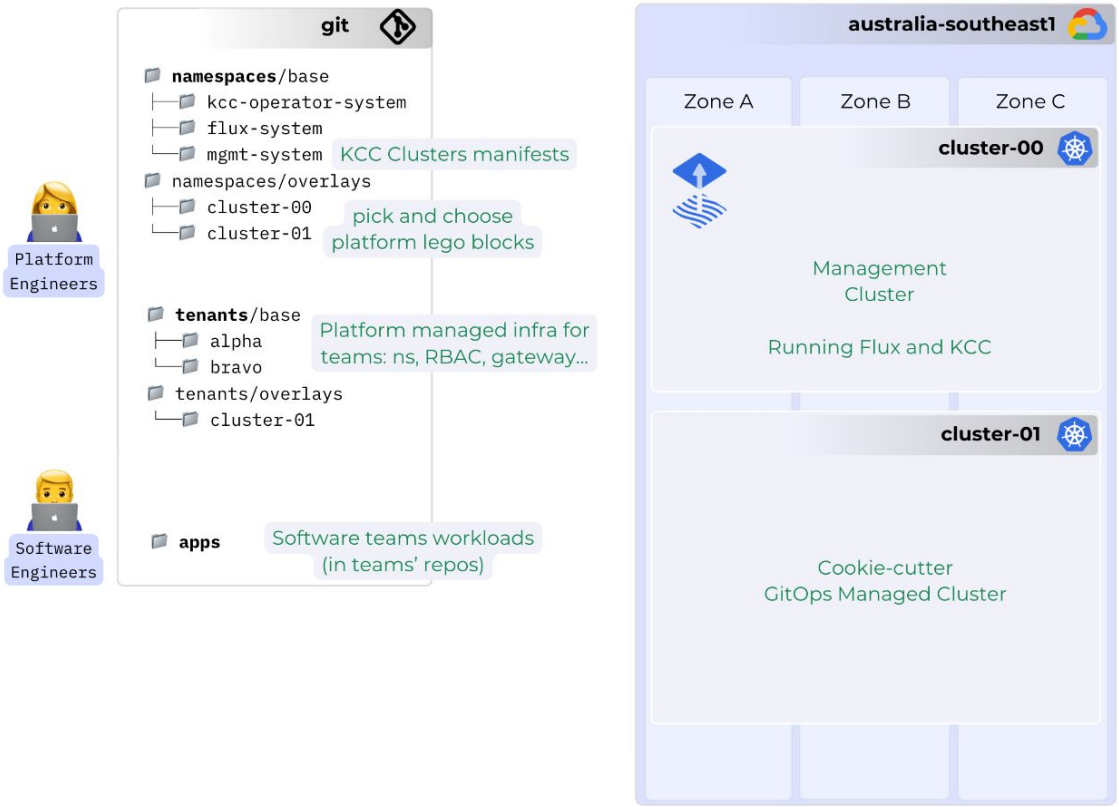
Unpacking the Demo - Principles

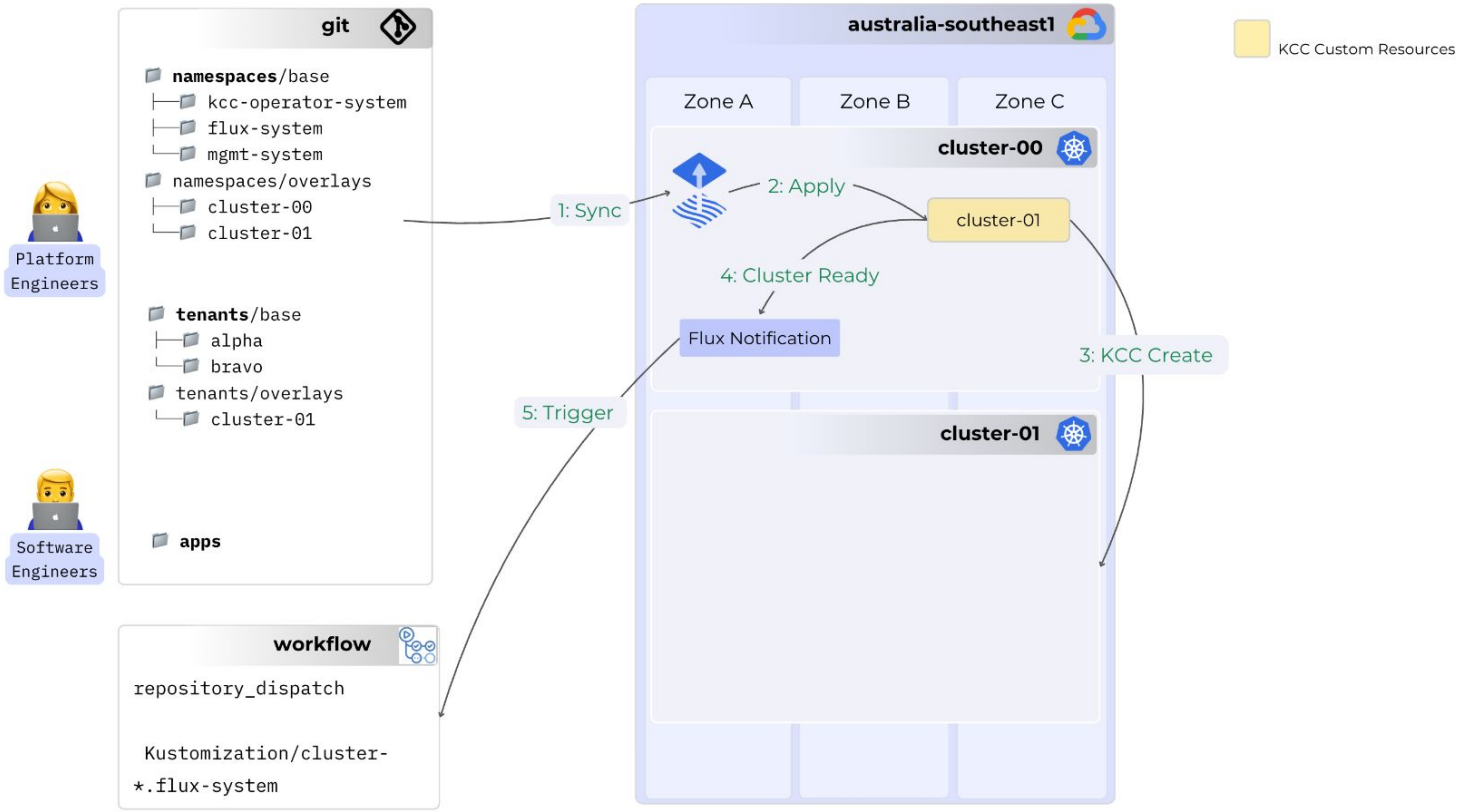
Platform Engineer

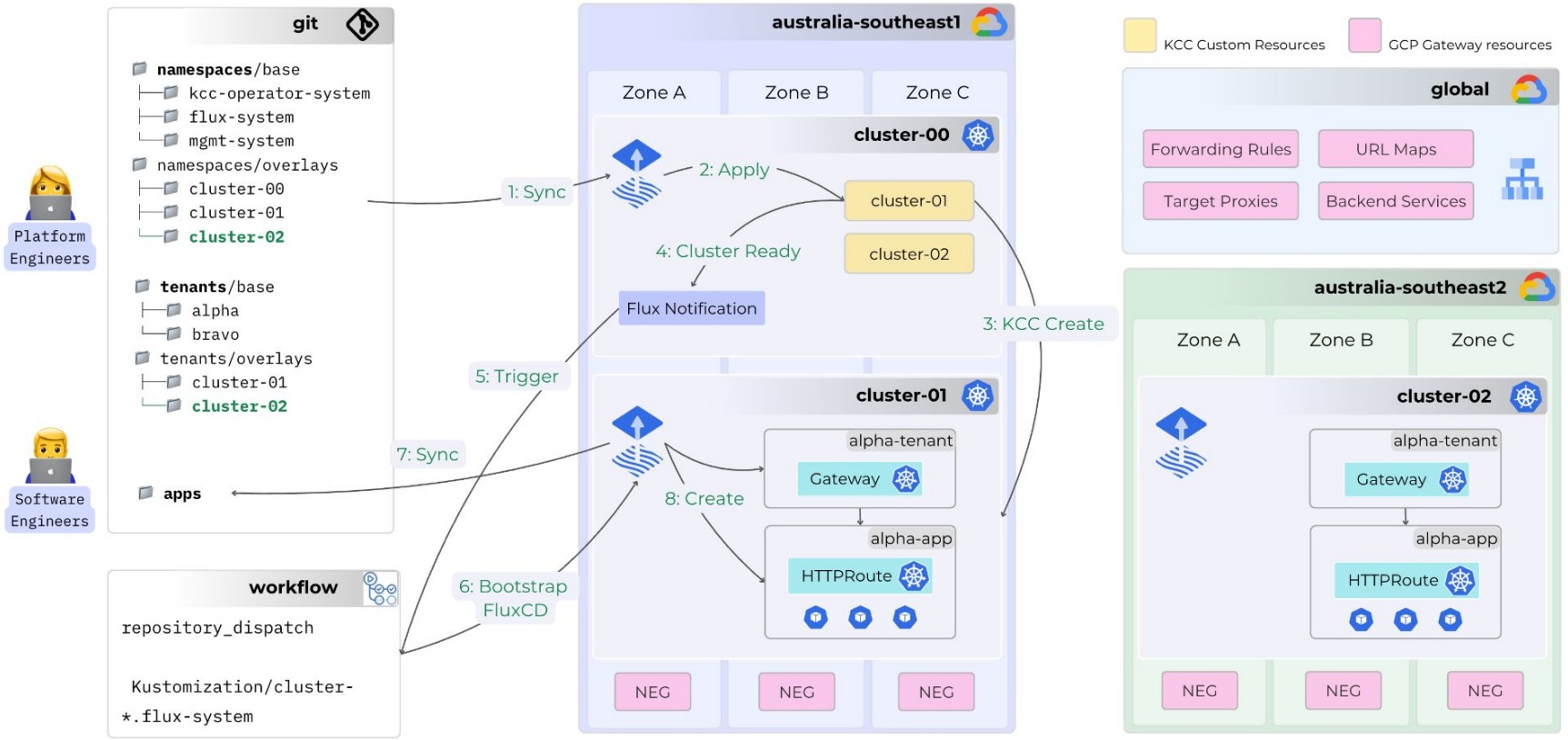
- Cookie-cutter reproducible clusters
- Clusters on-demand: Testing Environments, Disaster Recovery.
- Golden paths and guardrails
- Scalable

Application Developer

- Infrastructure management decoupled from Application
- App placement based on archetype
- Scale seamlessly to 5 or 500 clusters


git
namespaces/base
kcc-operator-system
flux-system
mgmt-system
KCC Clusters manifests
namespaces/overlays
cluster-00
cluster-01
pick and choose
platform lego blocks
Platform
Engineers
tenants/base
alpha
bravo
Platform managed infra for
teams: ns, RBAC, gateway...
tenants/overlays
cluster-01
apps
Software teams workloads
(in teams' repos)
Software
Engineers
australia-southeast1
Zone A
Zone B
Zone C
cluster-00
Management
Cluster
Running Flux and KCC
cluster-01
Cookie-cutter
GitOps Managed Cluster


git
namespaces/base
kcc-operator-system
flux-system
mgmt-system
namespaces/overlays
cluster-00
cluster-01
tenants/base
alpha
bravo
tenants/overlays
cluster-01
apps
Platform Engineers
Software Engineers
1: Sync
australia-southeast1
Zone A
Zone B
Zone C
cluster-00
2: Apply
cluster-01
4: Cluster Ready
Flux Notification
3: KCC Create
5: Trigger
cluster-01
KCC Custom Resources
workflow
repository_dispatch
Kustomization/cluster-*.flux-system


git
namespaces/base
kcc-operator-system
flux-system
mgmt-system
namespaces/overlays
cluster-00
cluster-01
cluster-02
tenants/base
alpha
bravo
tenants/overlays
cluster-01
cluster-02
apps
Platform Engineers
Software Engineers
workflow
repository_dispatch
Kustomization/cluster-*.flux-system
1: Sync
2: Apply
3: KCC Create
4: Cluster Ready
5: Trigger
6: Bootstrap FluxCD
7: Sync
8: Create
australia-southeast1
Zone A
Zone B
Zone C
cluster-00
cluster-01
cluster-02
Flux Notification
cluster-01
alpha-tenant
Gateway
alpha-app
HTTPRoute
NEG
NEG
NEG
KCC Custom Resources
GCP Gateway resources
global
Forwarding Rules
URL Maps
Target Proxies
Backend Services
australia-southeast2
Zone A
Zone B
Zone C
cluster-02
alpha-tenant
Gateway
alpha-app
HTTPRoute
NEG
NEG
NEG

Takeaways

Unlock the power of archetypes

Unlock the power GitOps

Scale kubernetes configuration

Conclusion & Links

- https://fluxcd.io/flux/
- https://control-plane.io/posts/d2-reference-architecture-guide/
- https://rawkode.academy/read/fluxcd-the-inevitable-choice/
- https://slsa.dev/spec/v1.2-rc1/use-cases
- https://cloud.google.com/architecture/deployment-archetypes
- https://opengitops.dev/
- https://platformengineering.org/blog/what-is-platform-engineering

Let’s Connect!

linkedin.com/in/olgamirensky/

linkedin.com/in/michael-fornaro/

Easygo Careers