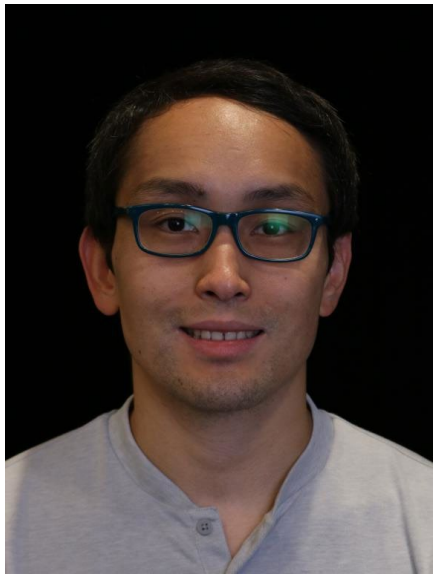

KubeCon
CloudNativeCon
North America 2022
BUILDING FOR THE ROAD AHEAD
DETROIT 2022
TiKV
TiKV

Secure Multi-Tenant GitOps Application & Infrastructure Rollouts At Adobe



Vikram Sethi
Sr. Architect
Adobe

Manabu McCloskey
Sr. Solutions Architect
AWS

Artwork by **Dan Zucco**

Agenda

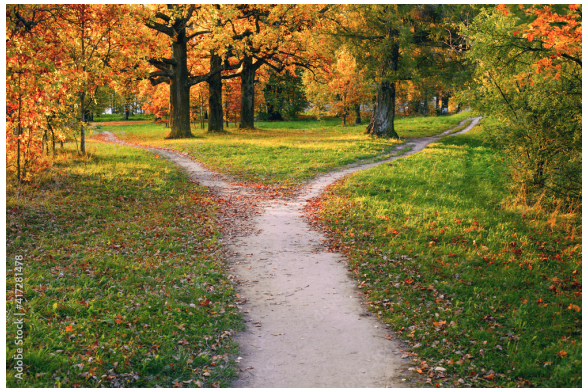
Pain Points, Need, Requirements
Adobe's Services Landscape
Adobe Internal Developer Platform (IDP) - Overview
Deployments using Argo projects
Infrastructure Provisioning using Crossplane + Argo
Multi-tenancy and Security requirements
Developer Experience – Previous vs New
Challenges, Unknowns

Previous State - Pain Points

Orange – Pain Point
Green – Working well
Black - Neutral

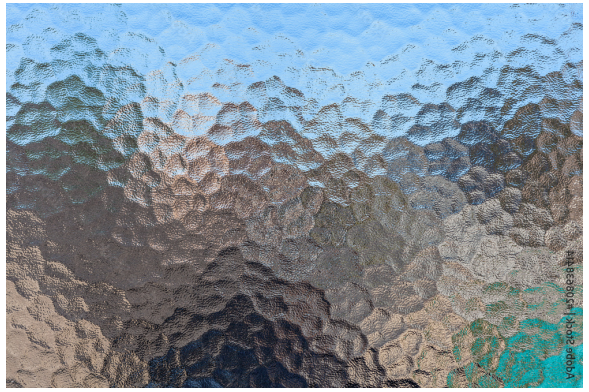
Service Team

- Provision cloud accounts
- Use GitOps for compute deployments
- Define infrastructure resources / templates
- Learn custom workflows / tools
- Use custom solution for Infra provisioning
- Maintain/Manage provisioned Infra resources.
- Track Infra resources separately than compute resources



Platform Team

- Manage GitOps workflows
- Enforce Multi-tenancy and Security
- No visibility into Infra resources provisioned by Service teams
- Harder troubleshooting in real world scenarios



The Need

Adobe’s Internal Survey Results (237 responses)

89%
Need a templatized infrastructure provisioning solution

65%
Want infrastructure provisioning integrated with existing GitOps based workflows

68%
Ready to use the solution starting in 2022

Platform Requirements

Standardization

Kubernetes Native

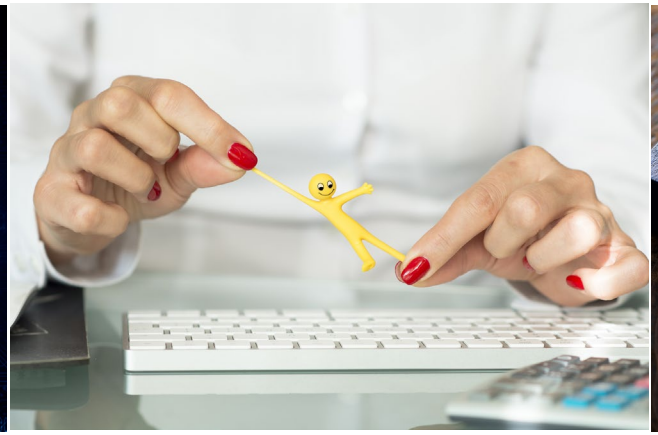
GitOps friendly

Multi-tenant

Secure

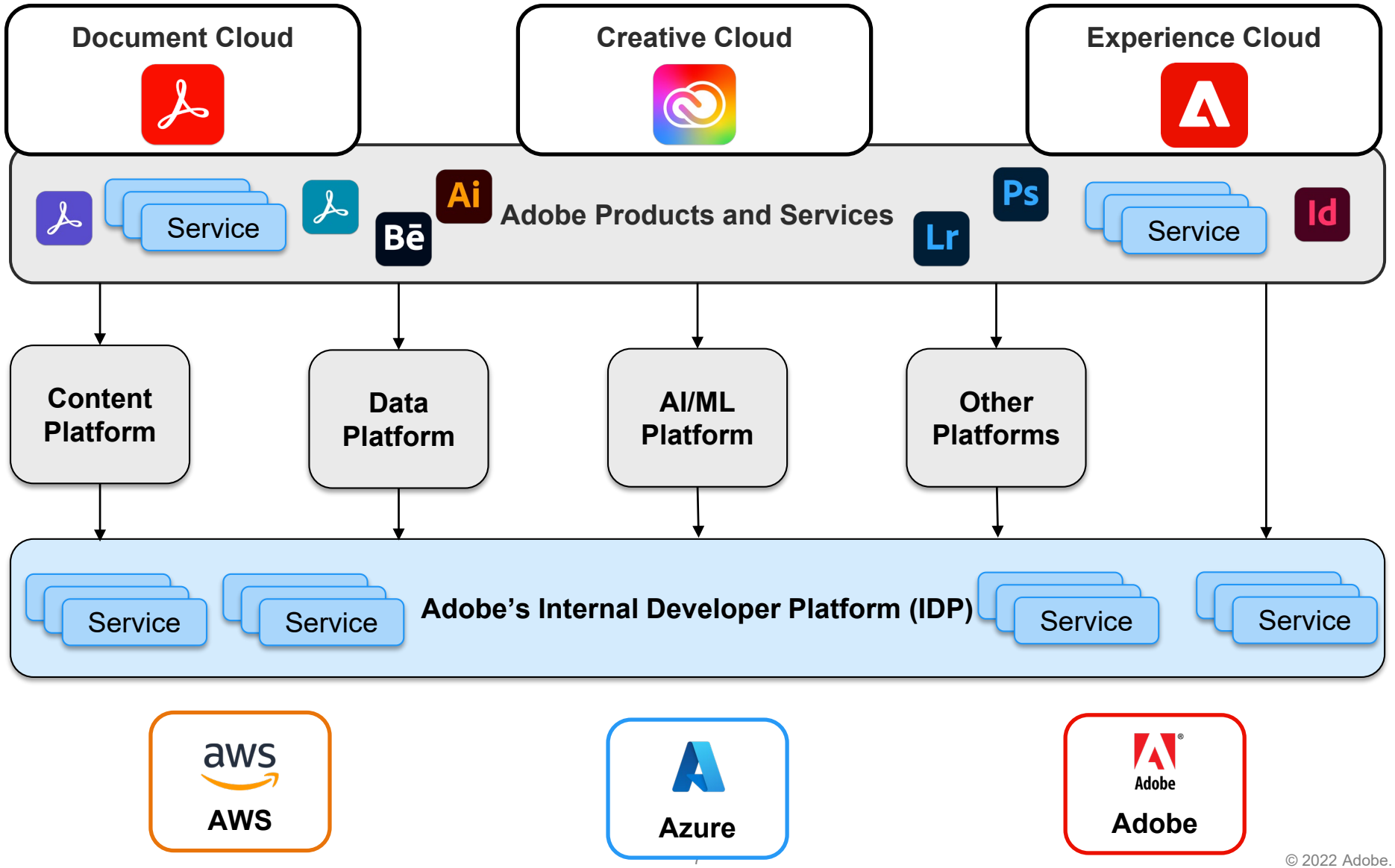
Multi-cloud

Extensible

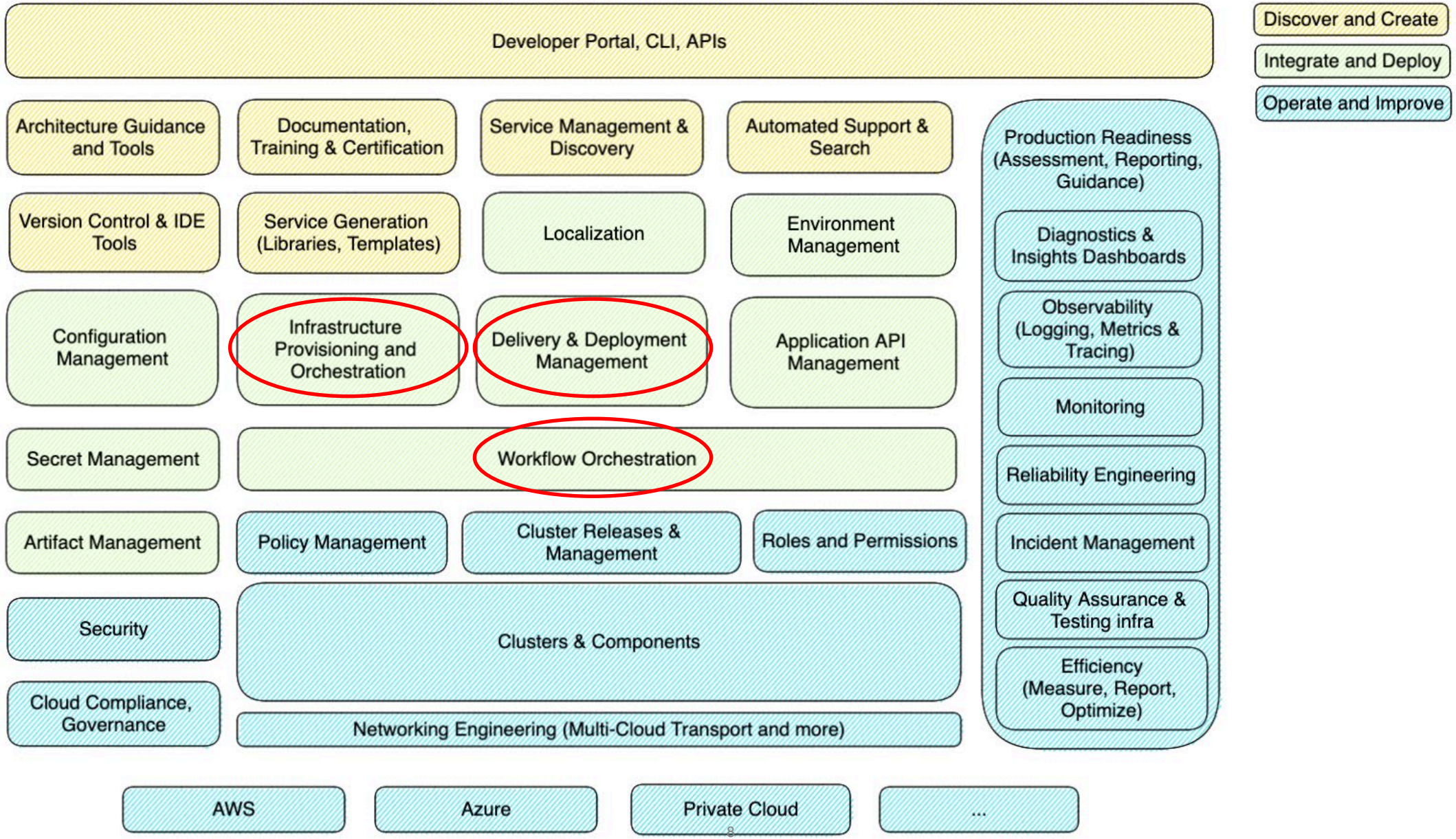


Industry alignment

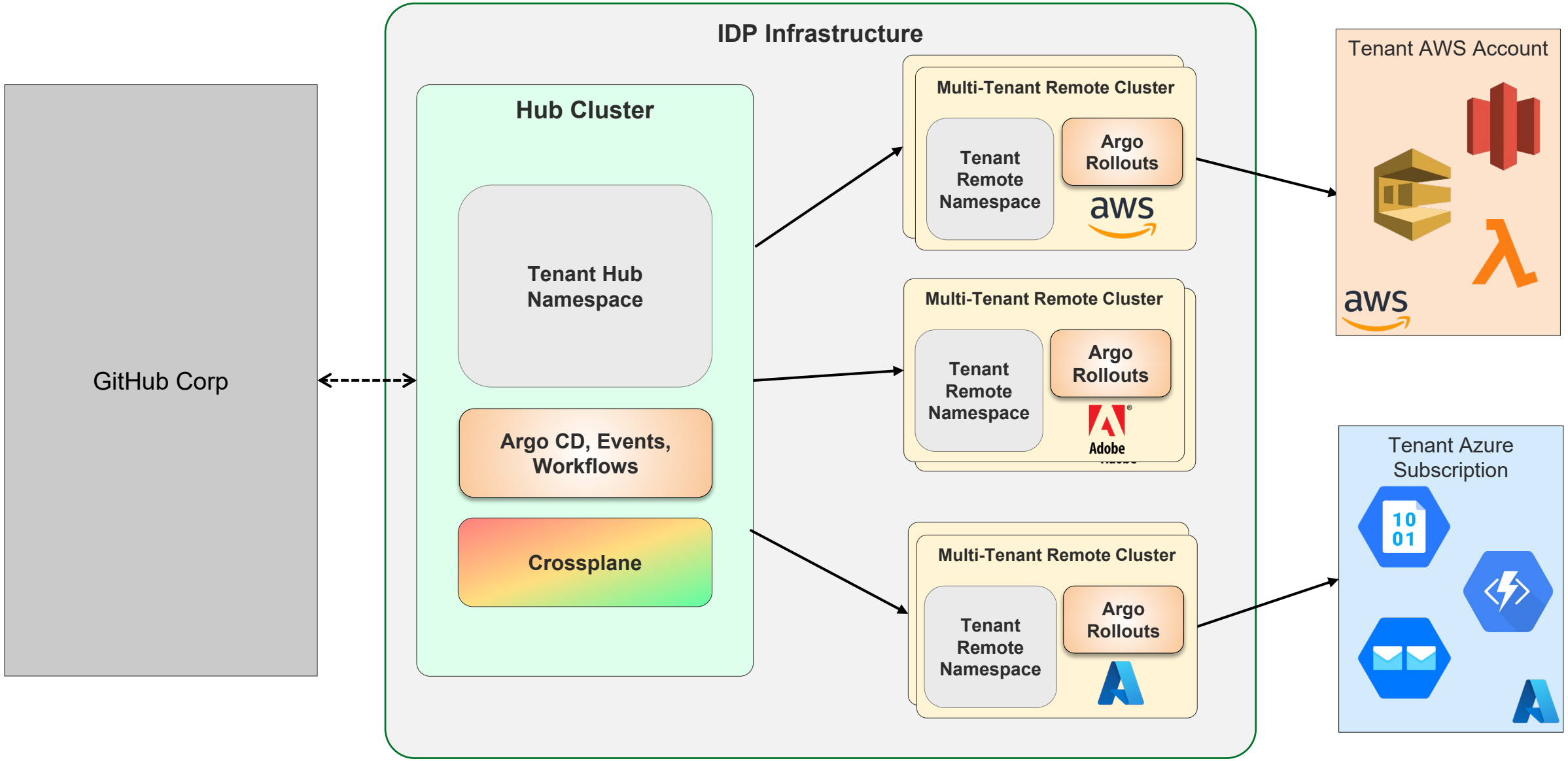
Adobe's Services Landscape



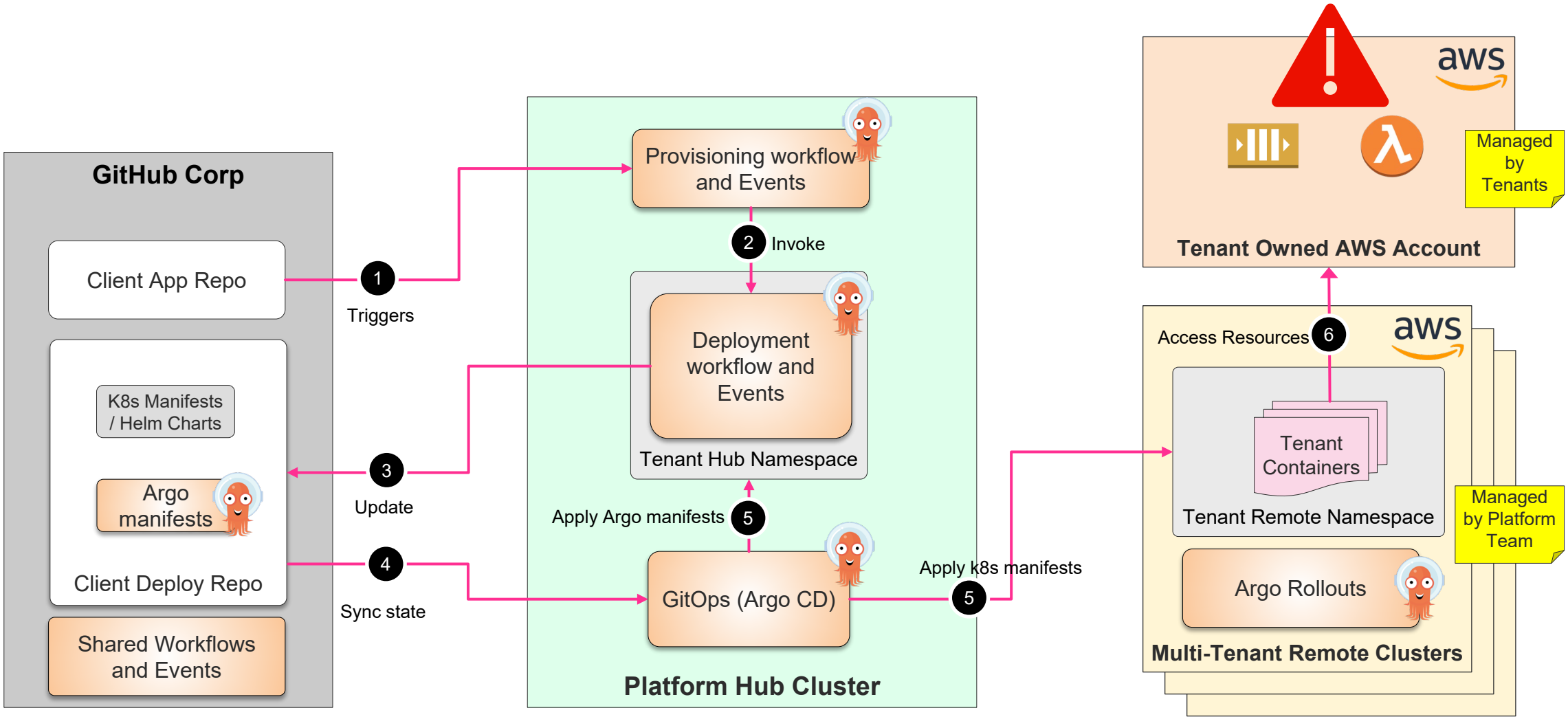
Adobe's Internal Developer Platform (IDP) - Overview



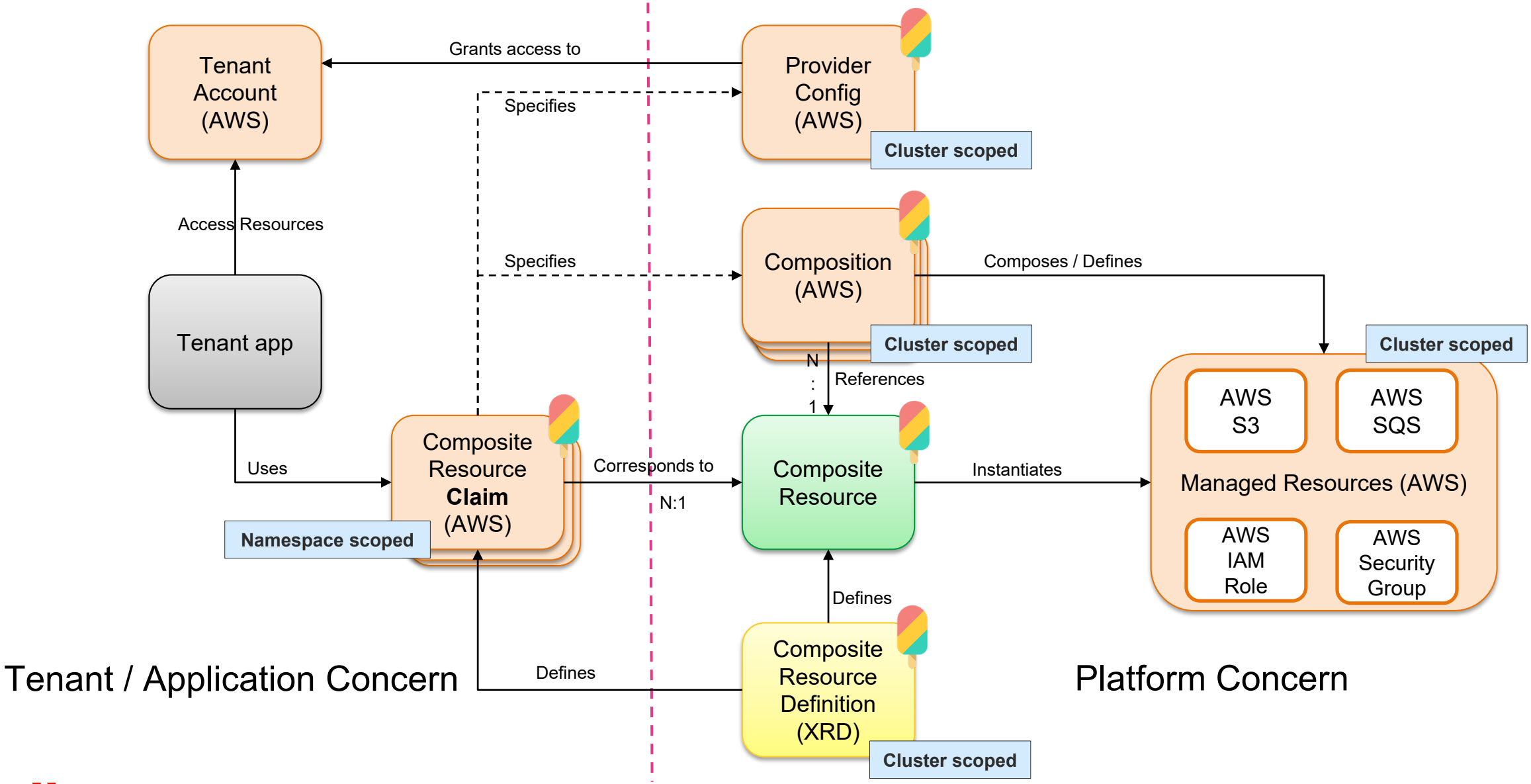
Hub and Spoke Model



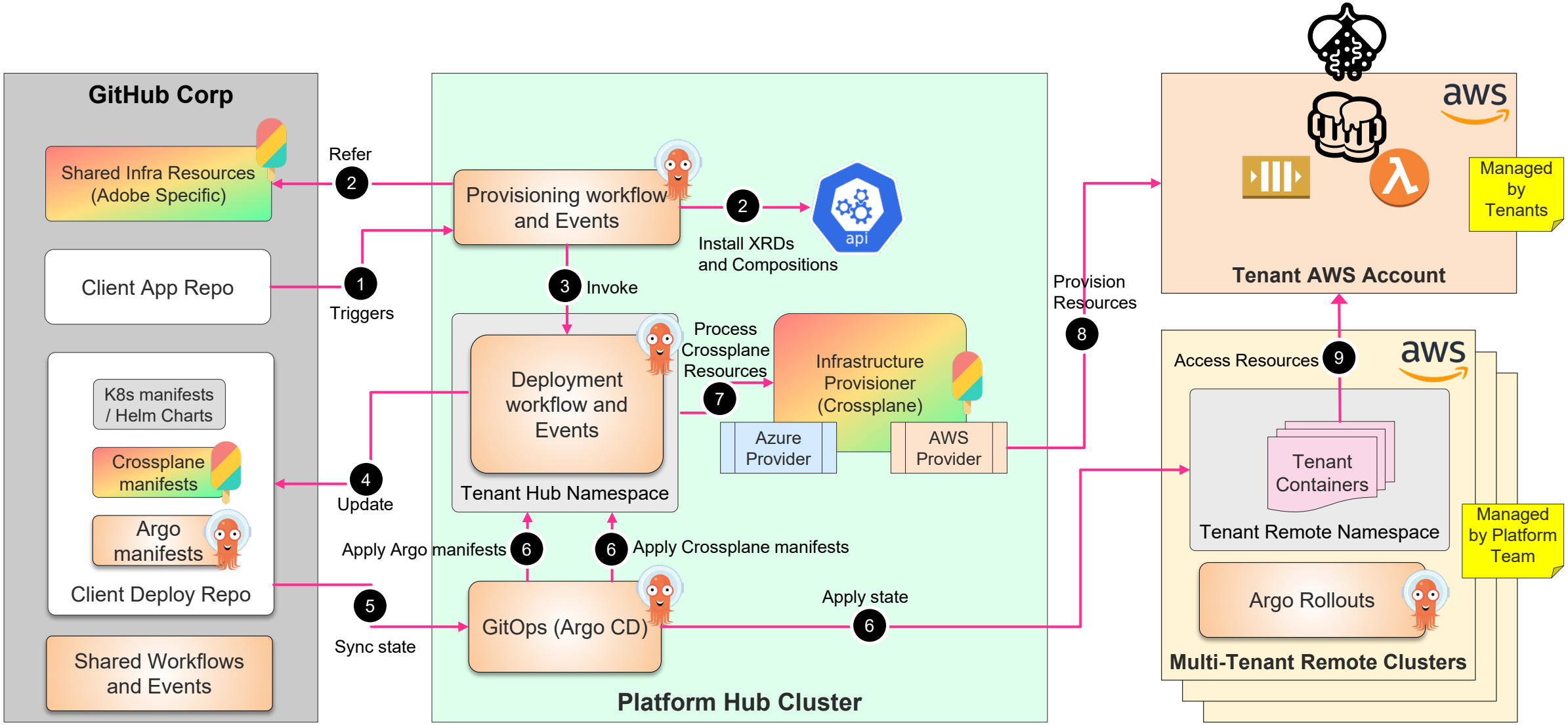
Deployments using Argo projects



Crossplane Concepts



Infra Provisioning using Crossplane + Argo



Multi-tenancy and Security Requirements

Legitimate Access	Namespace Isolation	Block Exploits	High Performance
			
Ability to provision in accounts tenants own	Resource access only at namespace level	Rogue actors should not be able to exploit the system	Existing and new workflows should be performant

Multi-tenancy and Security – Legitimate Access

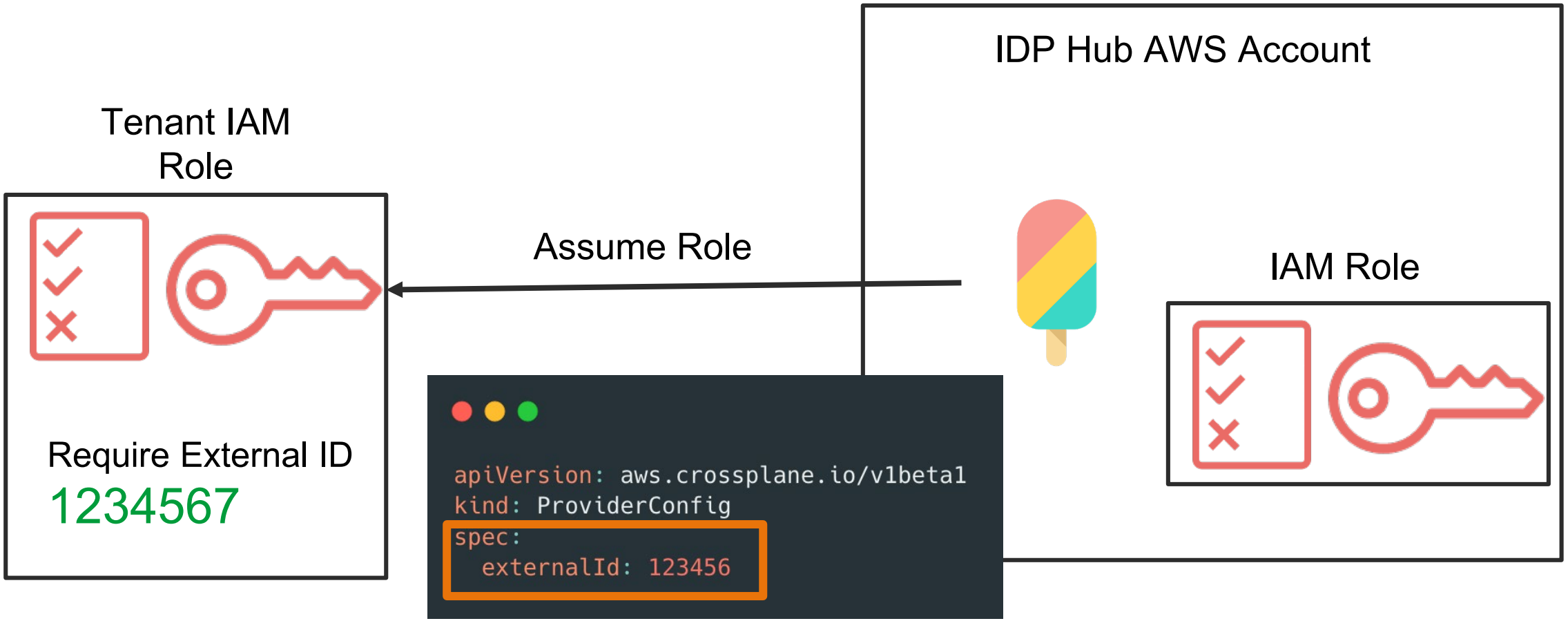
#1 - Tenant should not have access to any other tenant's provisioned resources

```yaml
apiVersion: aws.crossplane.io/v1beta1
kind: ProviderConfig
metadata:
  name: application-provider-config
spec:
  assumeRole:
    roleARN: arn:aws:iam::123456789:role/crossplane
```

≈

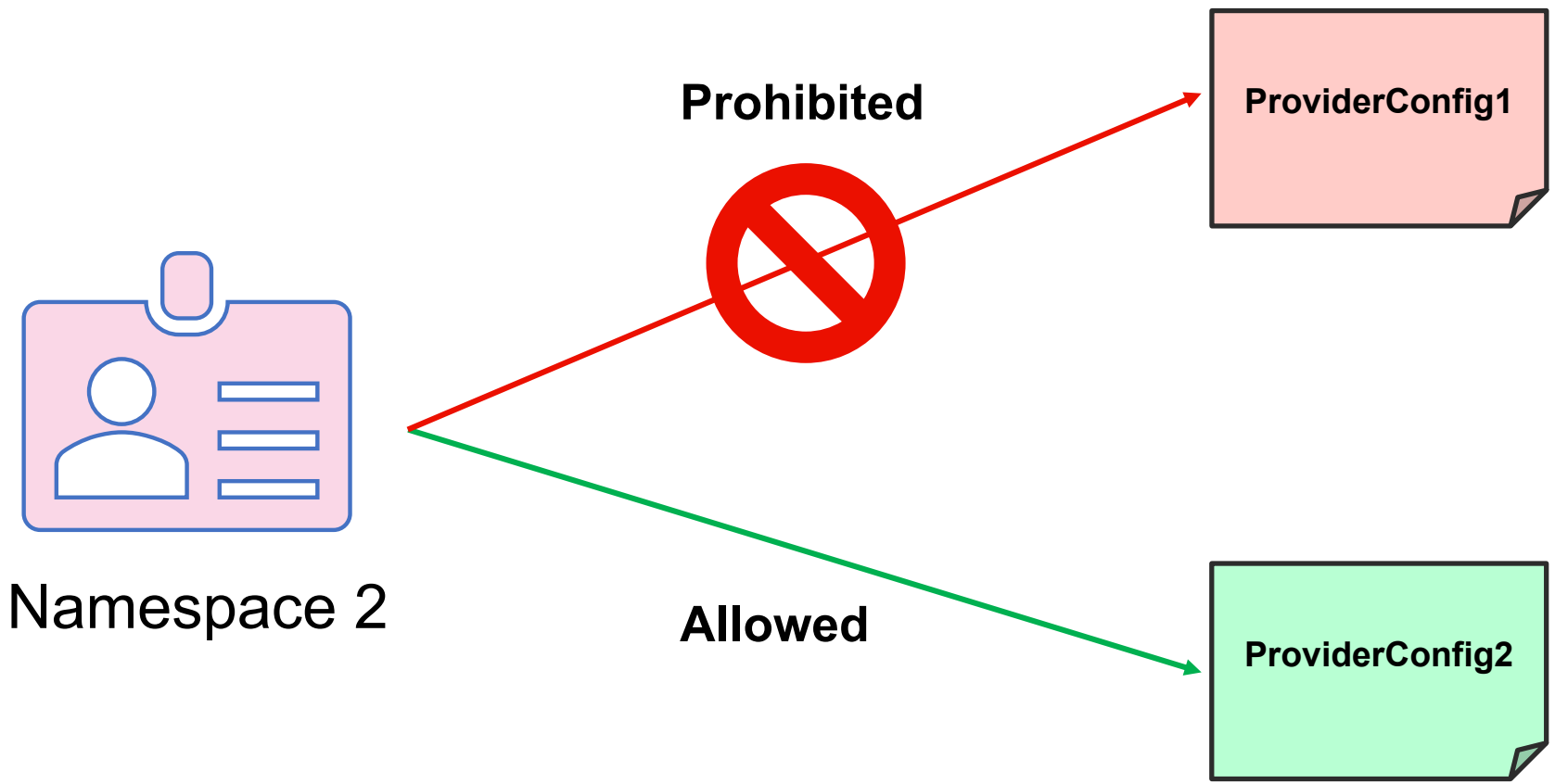
Multi-tenancy and Security – Legitimate Access

#1 Tenant should not have access to any other tenant's provisioned resources



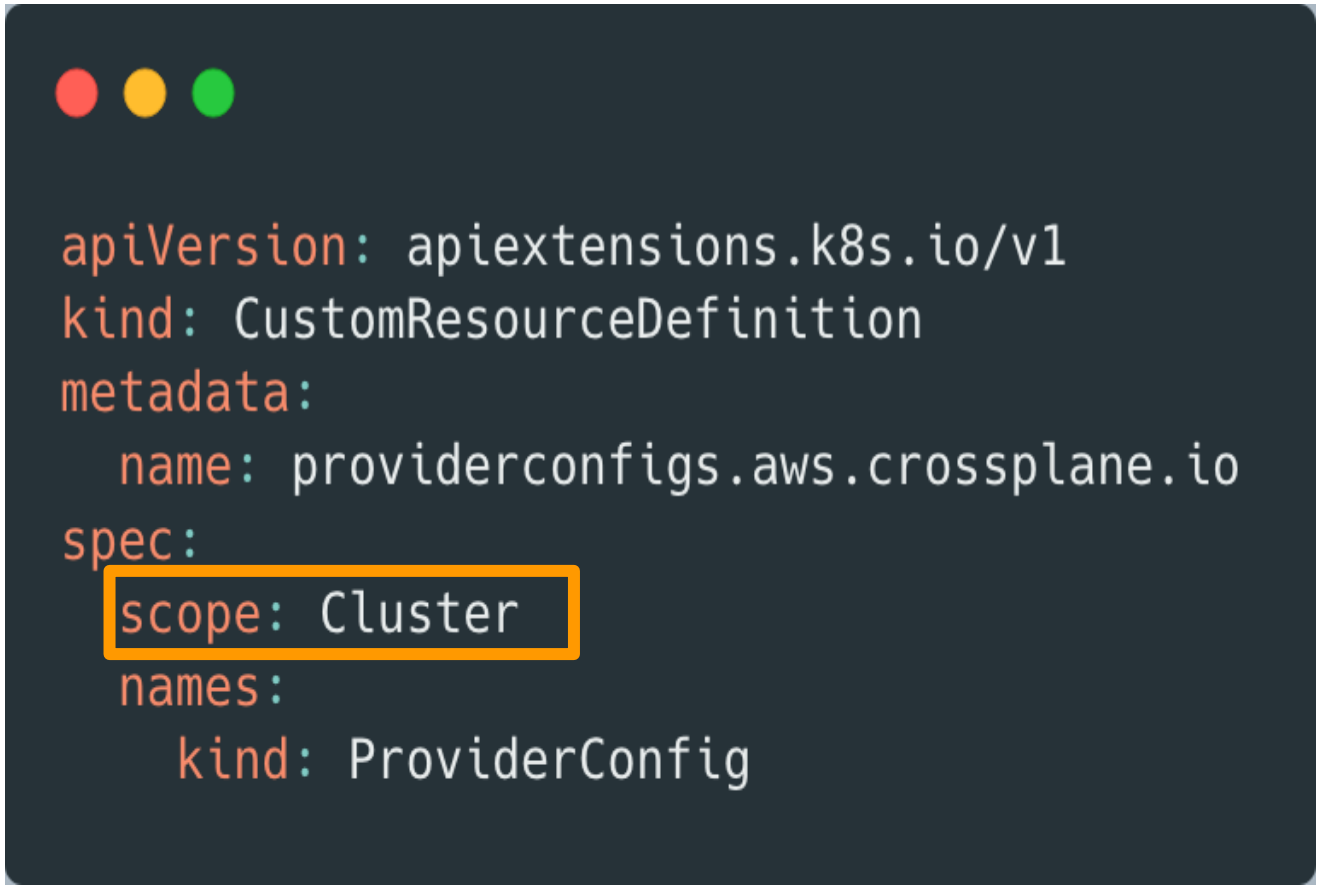
Multi-tenancy and Security – Namespace Isolation

#2 ProviderConfig can only be used in context of a tenant namespace



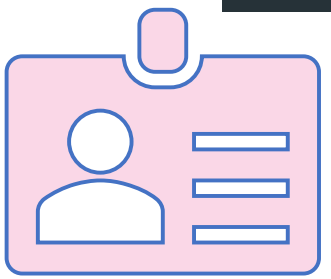
Multi-tenancy and Security – Namespace Isolation

#2 ProviderConfig can only be used in context of a tenant namespace



Multi-tenancy and Security – Namespace Isolation

#2 ProviderConfig can only be used in context of a tenant namespace



Namespace 2

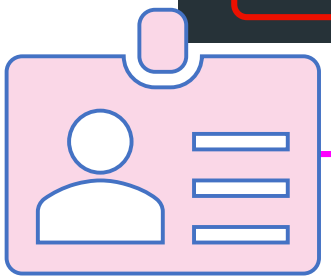
ProviderConfig1

ProviderConfig2

Multi-tenancy and Security – Namespace Isolation

#2 ProviderConfig can only be used in context of a tenant namespace

```yaml
apiVersion: eg.io/v1alpha1
kind: ObjectStorage
spec:
  providerConfig: ProviderConfig1
```



Namespace 2

Rewrite ProviderConfig name to “ProviderConfig2”

ProviderConfig1

ProviderConfig2

```yaml
fromFieldPath: metadata.labels[crossplane.io/claim-namespace]
toFieldPath: spec.providerConfigRef.name
combine:
  string:
    fmt: "%s-ProviderConfig"
```

Multi-tenancy and Security – Namespace Isolation

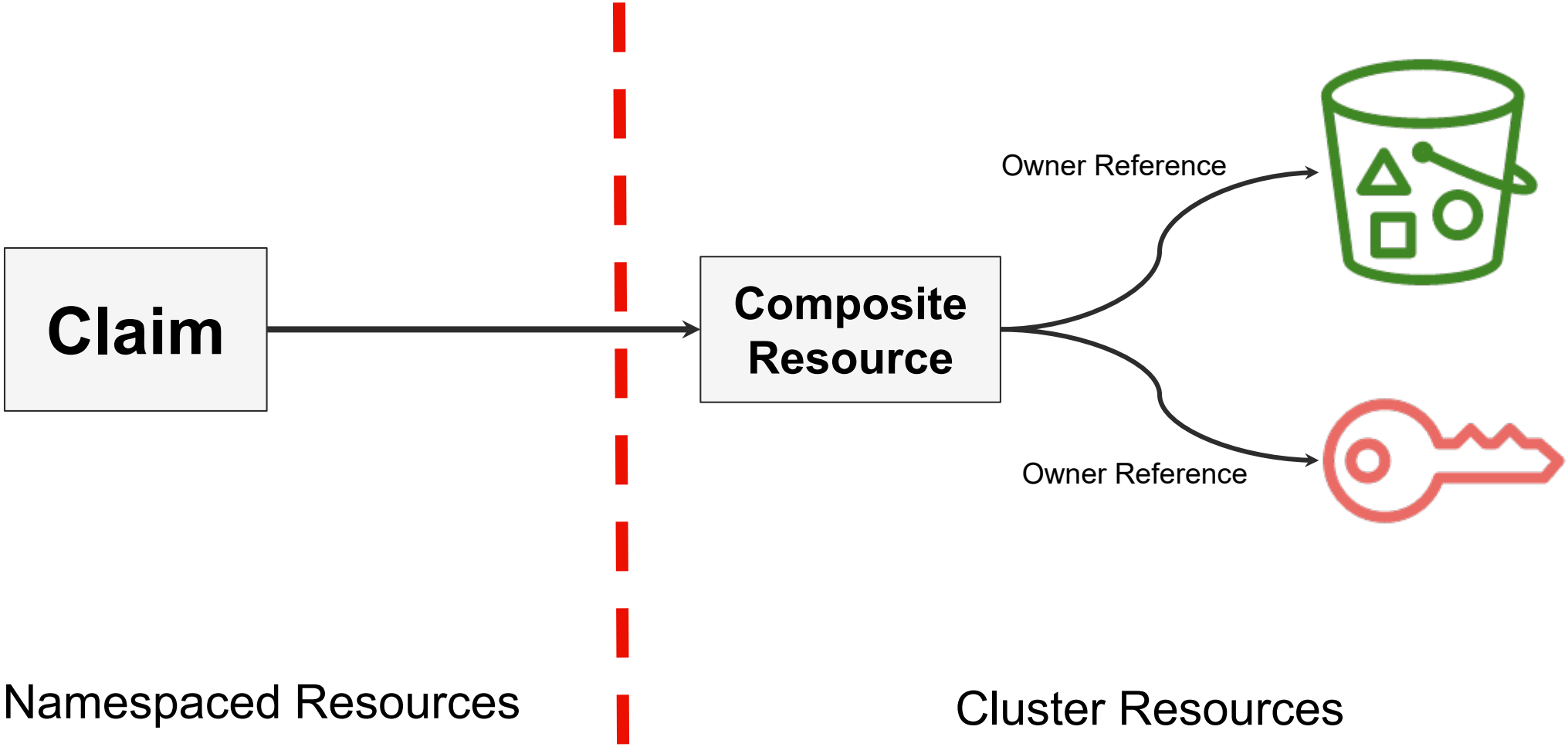
#3 Tenant should be able to use any managed resource from providers, but with namespace scope

```yaml
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  name: buckets.s3.aws.crossplane.io
spec:
  scope: Cluster
```

```yaml
apiVersion: s3.aws.crossplane.io/v1beta1
kind: Bucket
metadata:
  name: test-bucket
spec:
  forProvider:
    acl: private
    locationConstraint: us-east-1
  providerConfigRef:
    name: provider-config-1
```

Multi-tenancy and Security – Namespace Isolation

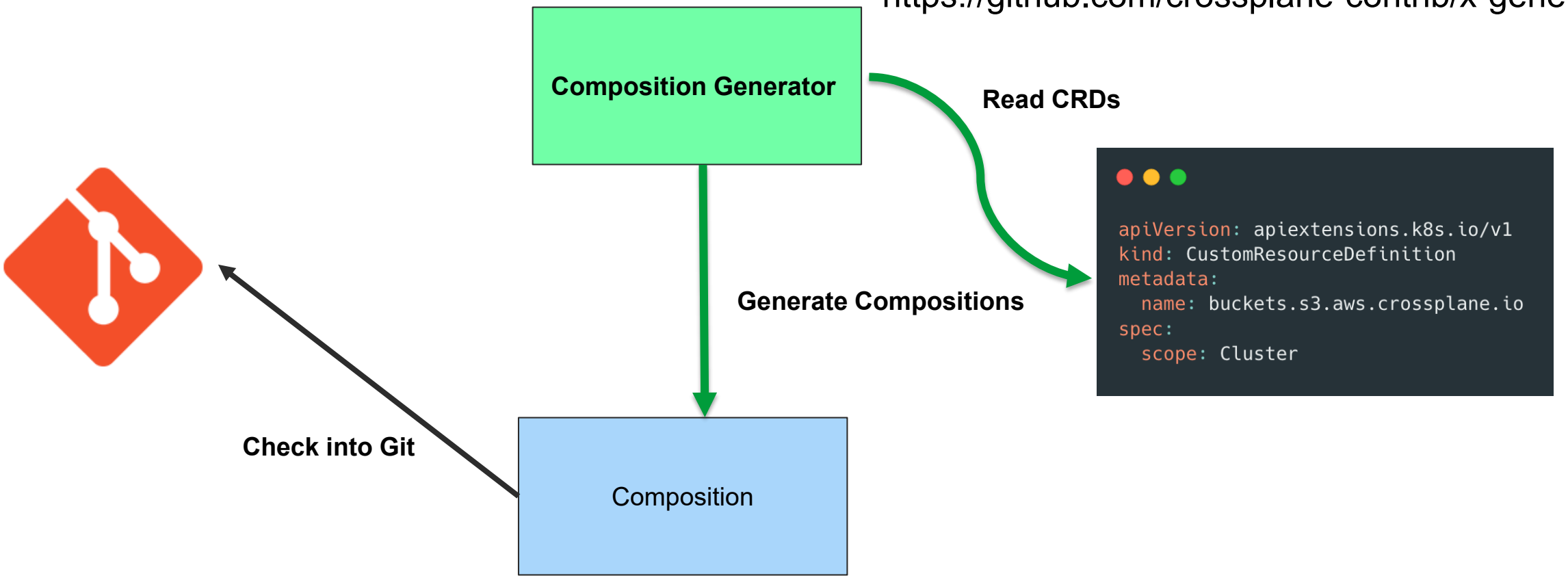
#3 Tenant should be able to use any managed resource from providers, but with namespace scope



Multi-tenancy and Security – Namespace Isolation

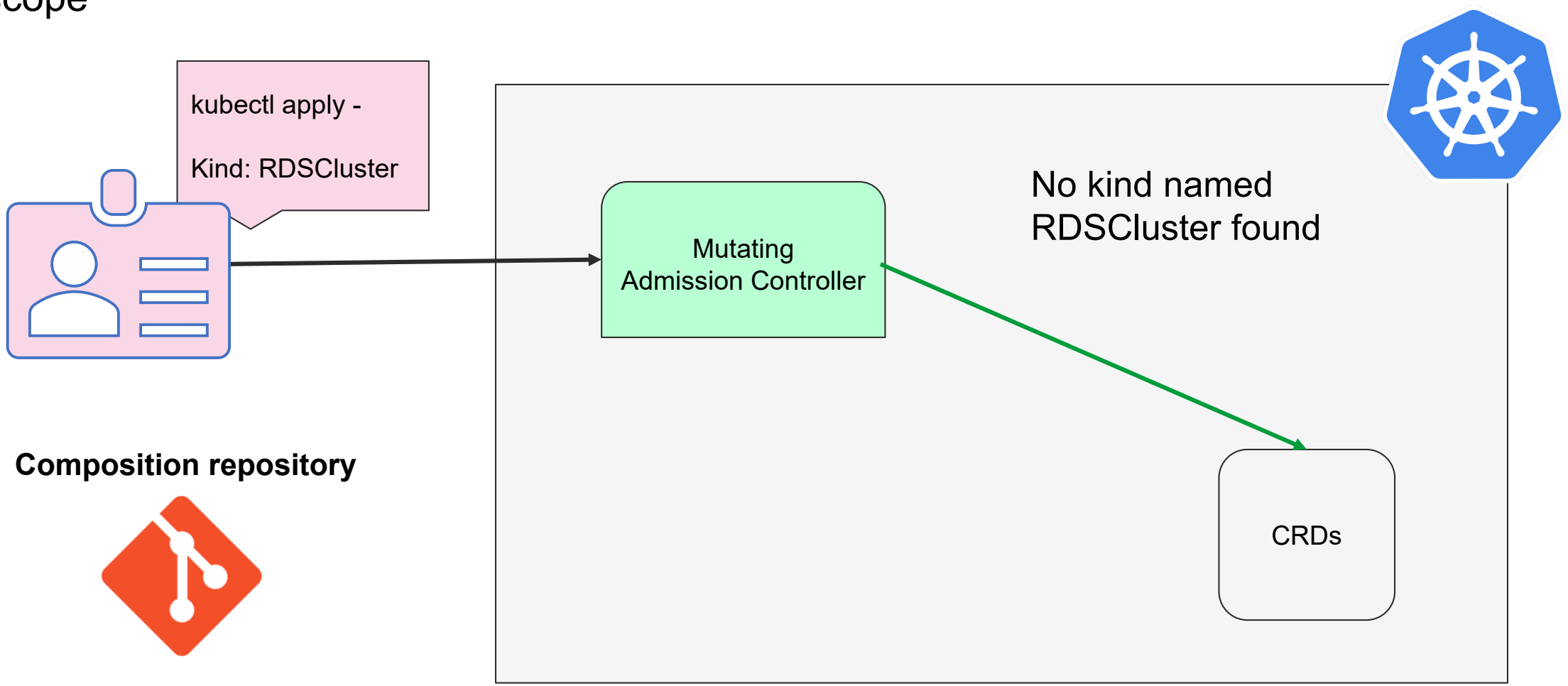
#3 Tenant should be able to use any managed resource from providers, but with namespace scope

Thanks **Christopher Haar**!
https://github.com/crossplane-contrib/x-generation



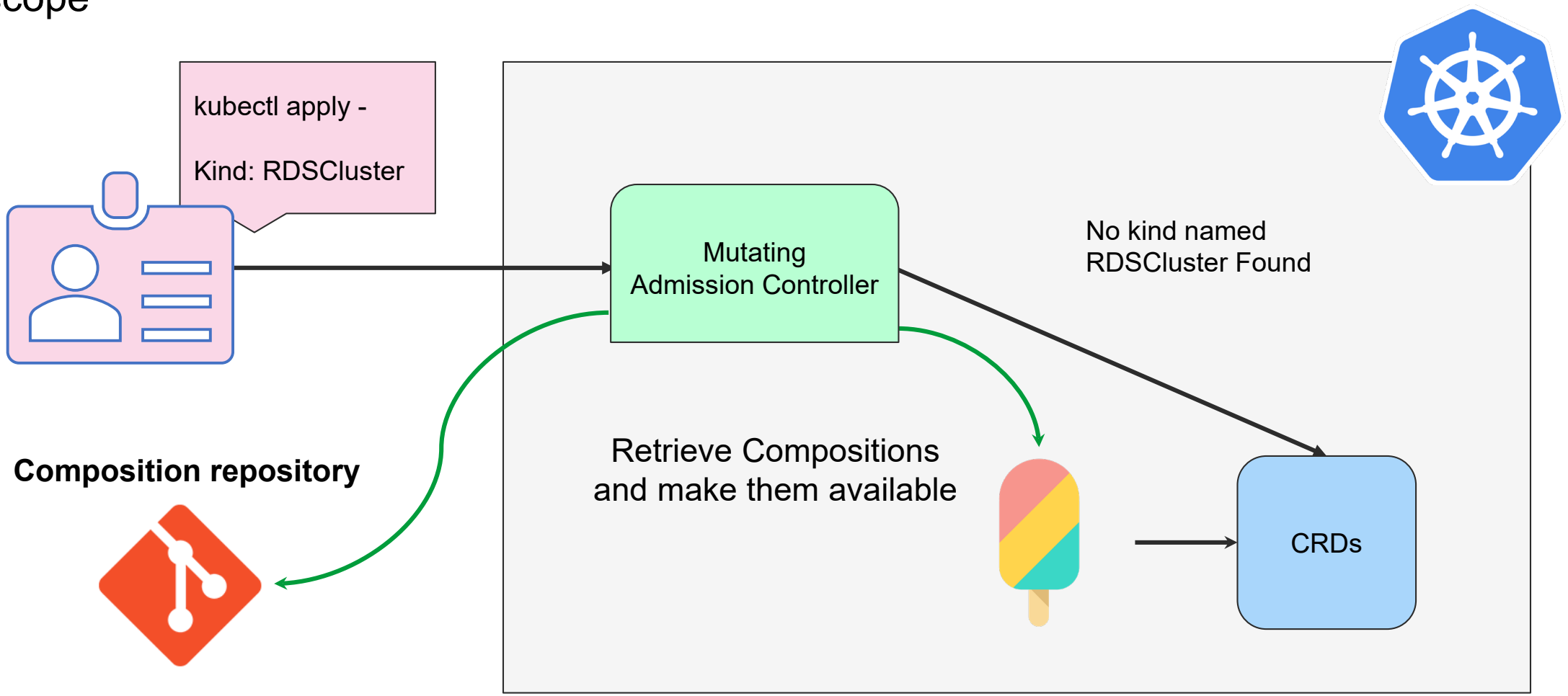
Multi-tenancy and Security – Namespace Isolation

#3 Tenant should be able to use any managed resource from providers, but with namespace scope



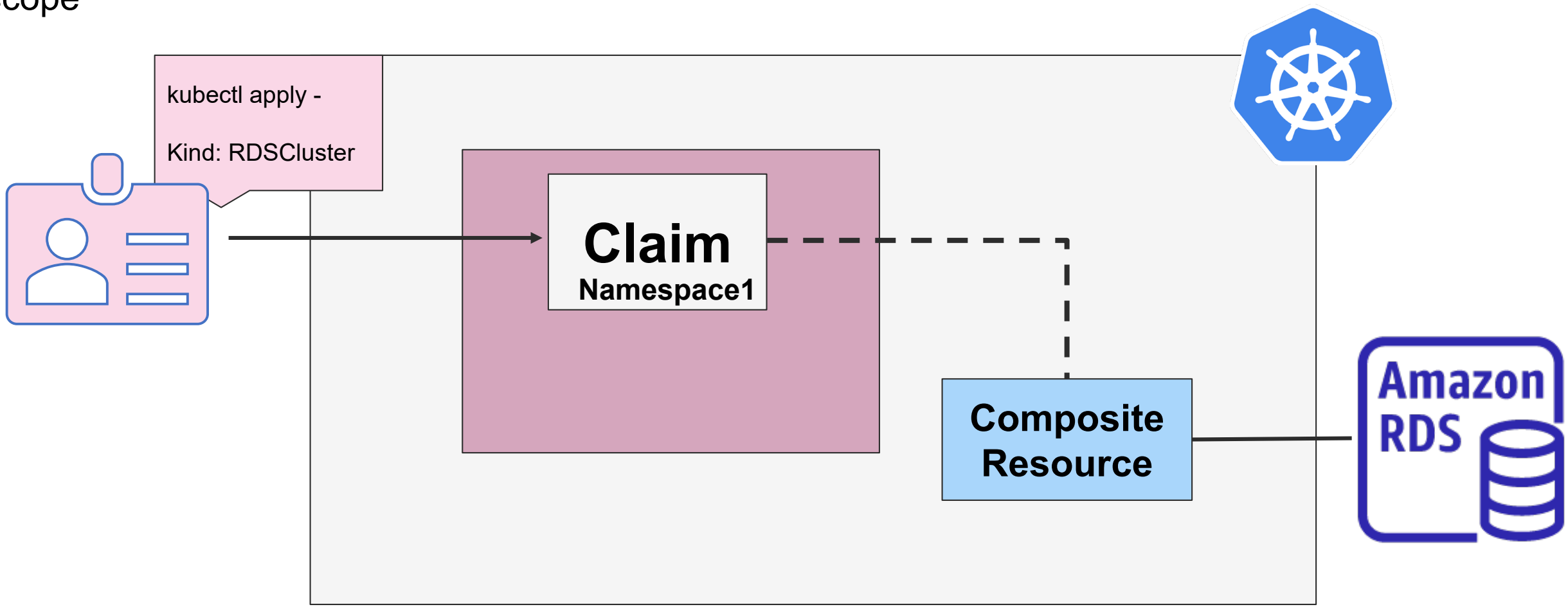
Multi-tenancy and Security – Namespace Isolation

#3 Tenant should be able to use any managed resource from providers, but with namespace scope



Multi-tenancy and Security – Namespace Isolation

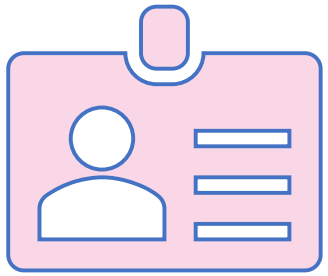
#3 Tenant should be able to use any managed resource from providers, but with namespace scope



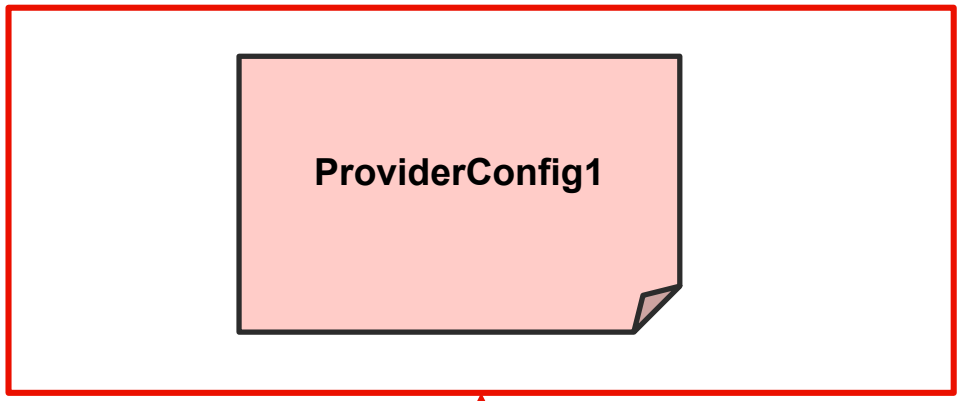
Multi-tenancy and Security – Block Exploits

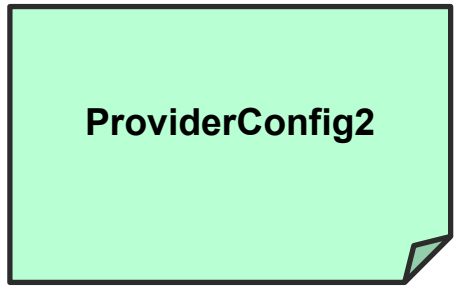
#4 A rogue tenant should not be able to exploit another tenant's ProviderConfig

```yaml
apiVersion: s3.aws.crossplane.io/v1beta1
kind: Bucket
metadata:
  name: test-bucket
spec:
  forProvider:
    acl: private
    locationConstraint: us-east-1
  providerConfigRef:
    name: provider-config-1
```



Namespace 2



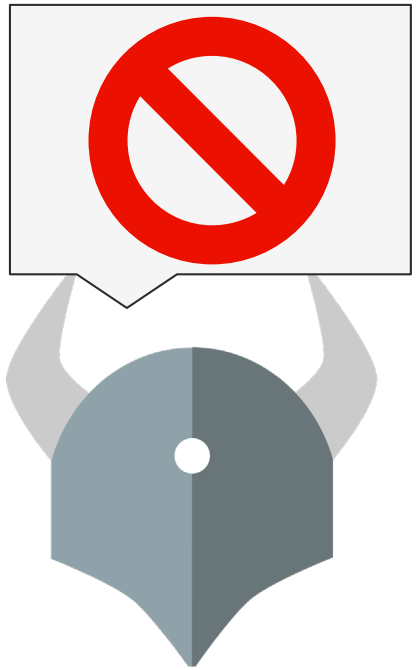


Multi-tenancy and Security – Block Exploits

#4 A rogue tenant should not be able to exploit another tenant's ProviderConfig

```yaml
apiVersion: s3.aws.crossplane.io/v1beta1
kind: Bucket
metadata:
  name: test-bucket
spec:
  forProvider:
    acl: private
    locationConstraint: us-east-1
  providerConfigRef:
    name: provider-config-1
```

Namespace 2



Open Policy Agent

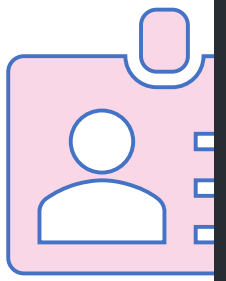
```
violation[{"msg": msg}] {
  not input.review.object.metadata.labels["crossplane.io/composite"]
  msg := "crossplane composition not used"
}

violation[{"msg": msg}] {
  not input.review.object.metadata.ownerReferences
  msg := "ownerReferences field not present"
}

violation[{"msg": msg}] {
  ownerRef = input.review.object.metadata.ownerReferences[0]
  not re_match("^crossplane.adobe.io", ownerRef.apiVersion)
  msg := "owner reference is not valid"
}
```

ProviderConfig1

ProviderConfig2

Multi-tenancy and Security – Block Exploits

#4 A rogue tenant should not be able to exploit another tenant's ProviderConfig



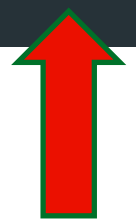
Namesp

```
violation[{"msg": msg}] {
  not input.review.object.metadata.labels["crossplane.io/composite"]
  msg := "crossplane composition not used"
}

violation[{"msg": msg}] {
  not input.review.object.metadata.ownerReferences
  msg := "ownerReferences field not present"
}

violation[{"msg": msg}] {
  ownerRef = input.review.object.metadata.ownerReferences[0]
  not re_match("^crossplane.adobe.io", ownerRef.apiVersion)
  msg := "owner reference is not valid"
}
```

Multi-tenancy and Security – High Performance

#5 Tenant workflows should not slow down with large number of CRDs

```
$ kubectl get crd | wc -l
Waited for 1.033772408s due to client-side throttling, not priority and fairness,
request: GET:https://api.ipa.org
Waited for 5.033772408s due to client-side throttling, not priority and fairness,
request: GET:https://api.ipa.org
     993
```



Multi-tenancy and Security – High Performance

#5 Tenant workflows should not slow down with large number of CRDs

Client: **0.24.0+**

Server: **1.25.0+**

Nic Cope's blog post:

https://blog.upbound.io/scaling-kubernetes-to-thousands-of-crds/

Previous v/s New

Orange – Pain Point
Green – Working well
Black - Neutral

Service Team

- Provision cloud accounts
- Use GitOps for compute deployments
- Define infrastructure resources / templates
- Learn custom workflows / tools
- Use custom solution for Infra provisioning
- Maintain/Manage provisioned non-compute resources.
- Track non-compute resources separately than compute resources

Platform Team

- Manage GitOps workflows
- Enforce Multi-tenancy and Security
- No visibility into non-compute resources
- Harder troubleshooting in real world scenarios

Service Team

- Provision cloud accounts
- Use GitOps for compute deployments
- Provision Infra resources via GitOps workflow
- Maintain/Manage provisioned non-compute resources.
- Specify and Track provisioned non-compute resources in Kubernetes-native way
- Consistent way to create/replicate environments

Platform Team

- Manage GitOps workflows
- Enforce Multi-tenancy and Security
- Define “blessed” infrastructure resources / templates
- Maintain “standardized” Infra provisioning workflow based on GitOps
- Improved Auditability, Observability
- Improved service architecture understanding
- Improved MTTR

Challenges, Unknowns

Hub cluster, K8s – Scale, Performance

Azure Support

Technology Maturity

Tooling Inertia

Ease of Migration

Community Support

THANK YOU!

Please scan the QR Code above to
leave feedback on this session


Adobe
Artwork by Dan Zucco

Crossplane related sessions

Topic	Presenter	Date/Time
Cloud Governance With Infrastructure As Code (IaC) With Kyverno And Crossplane	Dolis Sharma, Nirmata	Wednesday, October 26 4:30pm - 5:05pm
Like Peas And Carrots: Argo CD And Crossplane For Infrastructure Management	Jesse Suen, Akuity Viktor Farcic, Upbound	Wednesday, October 26 5:25pm - 6:00pm
Crossplane Intro And Deep Dive - The Cloud Native Control Plane Framework	Jared Watts, Matthias Luebken & Nic Cope, Upbound Bob Haddleton, Nokia	Friday, October 28 11:00am - 11:35am

Argo related sessions

Topic	Presenter	Date/Time
How To Build Production Grade DevOps Platform Using Argoproj	Alexander Matyushentsev, Akuity Leonardo Luz Almeida, Intuit	Wednesday, October 26 11:00am - 11:35am
Multi-Tenancy For Argo Workflows And Argo CD At Adobe	Srinivas Malladi, Adobe	Thursday, October 27 11:55am - 12:30pm
How the Argo Project Transitioned From Security Aware To Security First	Henrik Blixt & Michael Crenshaw, Intuit	Thursday, October 27 5:25pm - 6:00pm
How Adobe Planned For Scale With Argo CD, Cluster API, And vCluster	Joseph Sandoval, Adobe Dan Garfield, Codefresh	Friday, October 28 11:00am - 11:35am
How Salesforce Is Moving From Spinnaker To Argo Workflows For Provisioning Add-Ons	Mayank Kumar & Andy Chen, Salesforce	Friday, October 28 4:55pm - 5:30pm


Adobe

Multi-tenancy and Security requirements

1. Tenants should be able to provision cloud resources in their own cloud accounts
2. Tenants should not have access to anyone else's provisioned resources
3. ProviderConfig can only be used in context of a tenant namespace
4. A rogue tenant should not be able to exploit another tenant's ProviderConfig
5. Tenant workflows should not slow down with large number of CRDs
6. Tenant should be able to use any managed resource from providers, but with namespace scope
7. Secrets for created infra can be pushed/pulled dynamically to external secret management system (e.g., Vault)