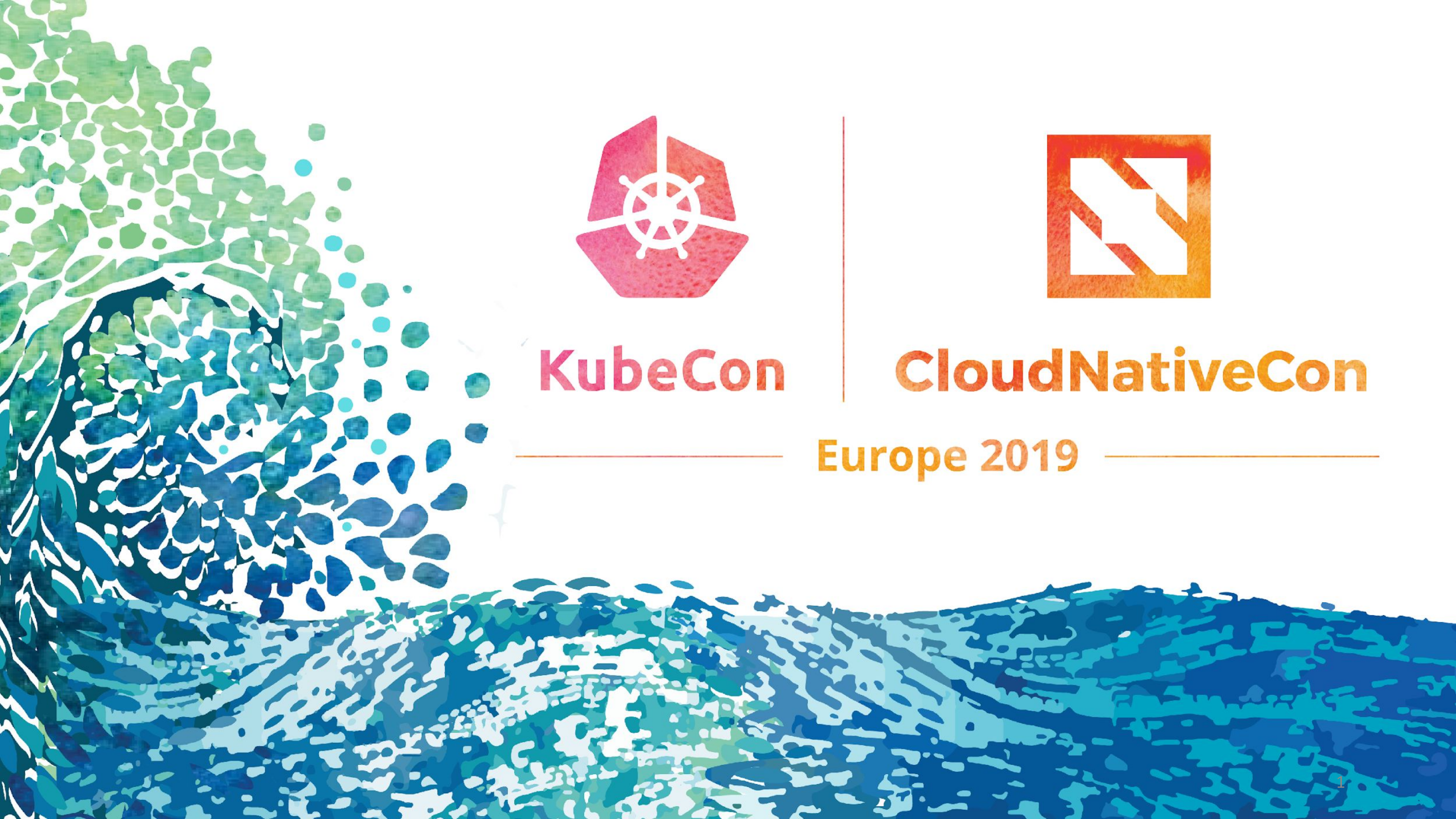


KubeCon | CloudNativeCon

Europe 2019

KubeCon | CloudNativeCon

Europe 2019

Lifecycle Of A kubectl Command:

Harden Kubernetes Setup With Automation

Sanjary Rahman, Booking.com

$ whoami

Sanjary Rahman
Site Reliability Engineer, Booking.com

sanjary.rahman@booking.com
in link.sanjary.dev
@Sanjary

Booking.com

Agenda

Agenda

- Multi-tenant cluster architecture in Booking.com

Agenda

- Multi-tenant cluster architecture in Booking.com
- Deployment workflow on Kubernetes in Booking.com

Agenda

- Multi-tenant cluster architecture in Booking.com
- Deployment workflow on Kubernetes in Booking.com
- Challenges faced managing those clusters

Agenda

- Multi-tenant cluster architecture in Booking.com
- Deployment workflow on Kubernetes in Booking.com
- Challenges faced managing those clusters
- Workspace provisioning automation

Agenda

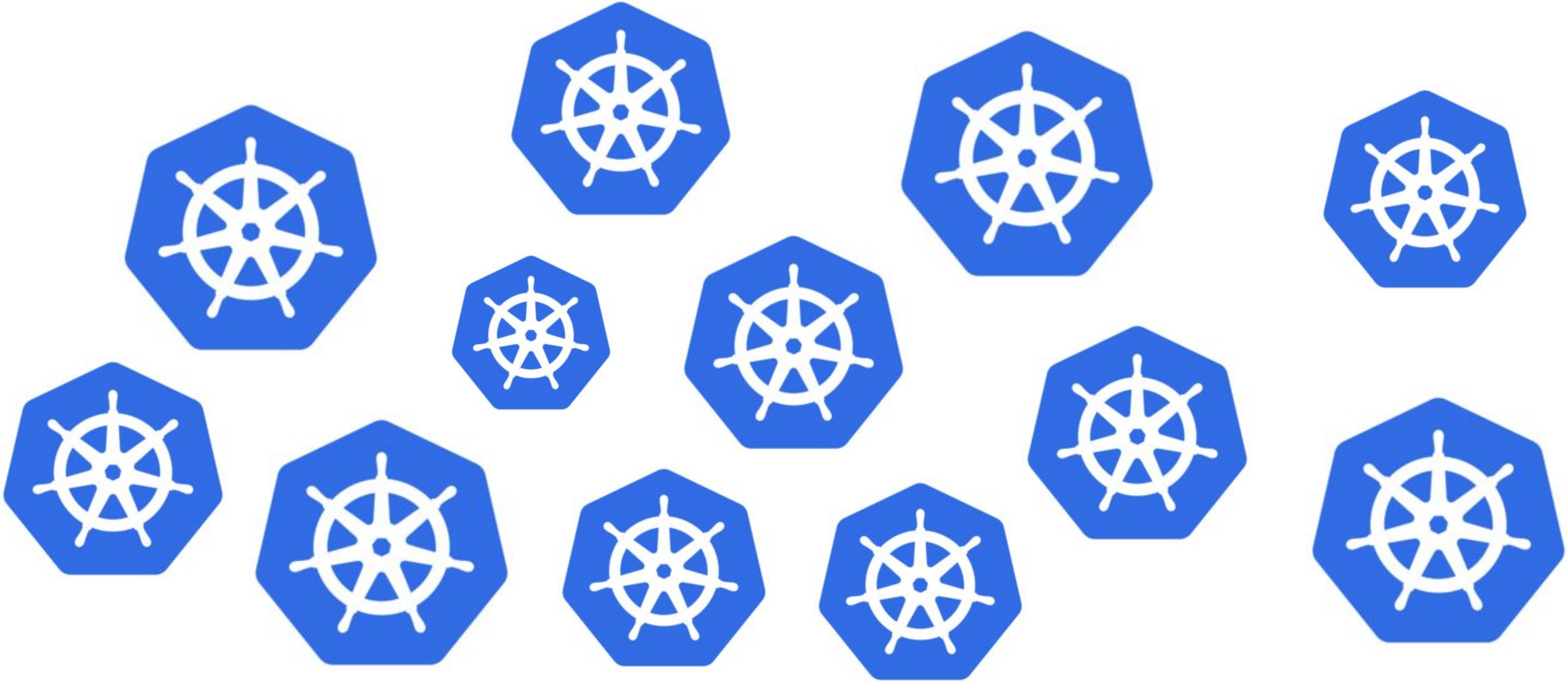
- Multi-tenant cluster architecture in Booking.com
- Deployment workflow on Kubernetes in Booking.com
- Challenges faced managing those clusters
- Workspace provisioning automation
- Lifecycle of a kubectl command

Agenda

- Multi-tenant cluster architecture in Booking.com
- Deployment workflow on Kubernetes in Booking.com
- Challenges faced managing those clusters
- Workspace provisioning automation
- Lifecycle of a kubectl command
- Q/A

Kubernetes Clusters in Booking.com

Kubernetes Clusters in Booking.com

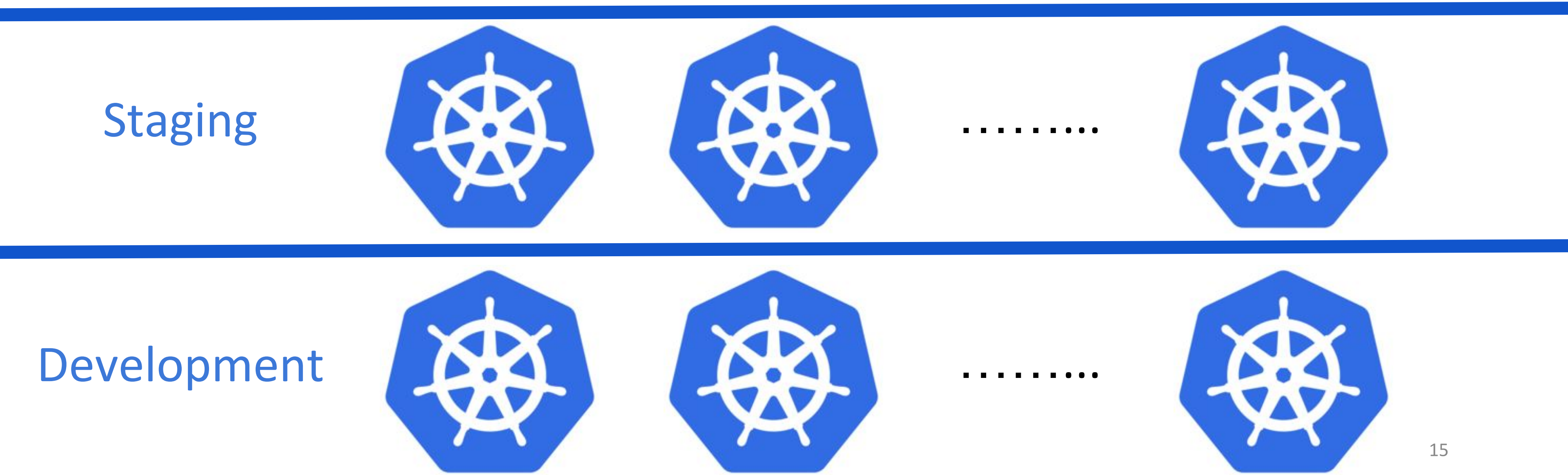


Kubernetes Clusters in Booking.com

Kubernetes Clusters in Booking.com

Development

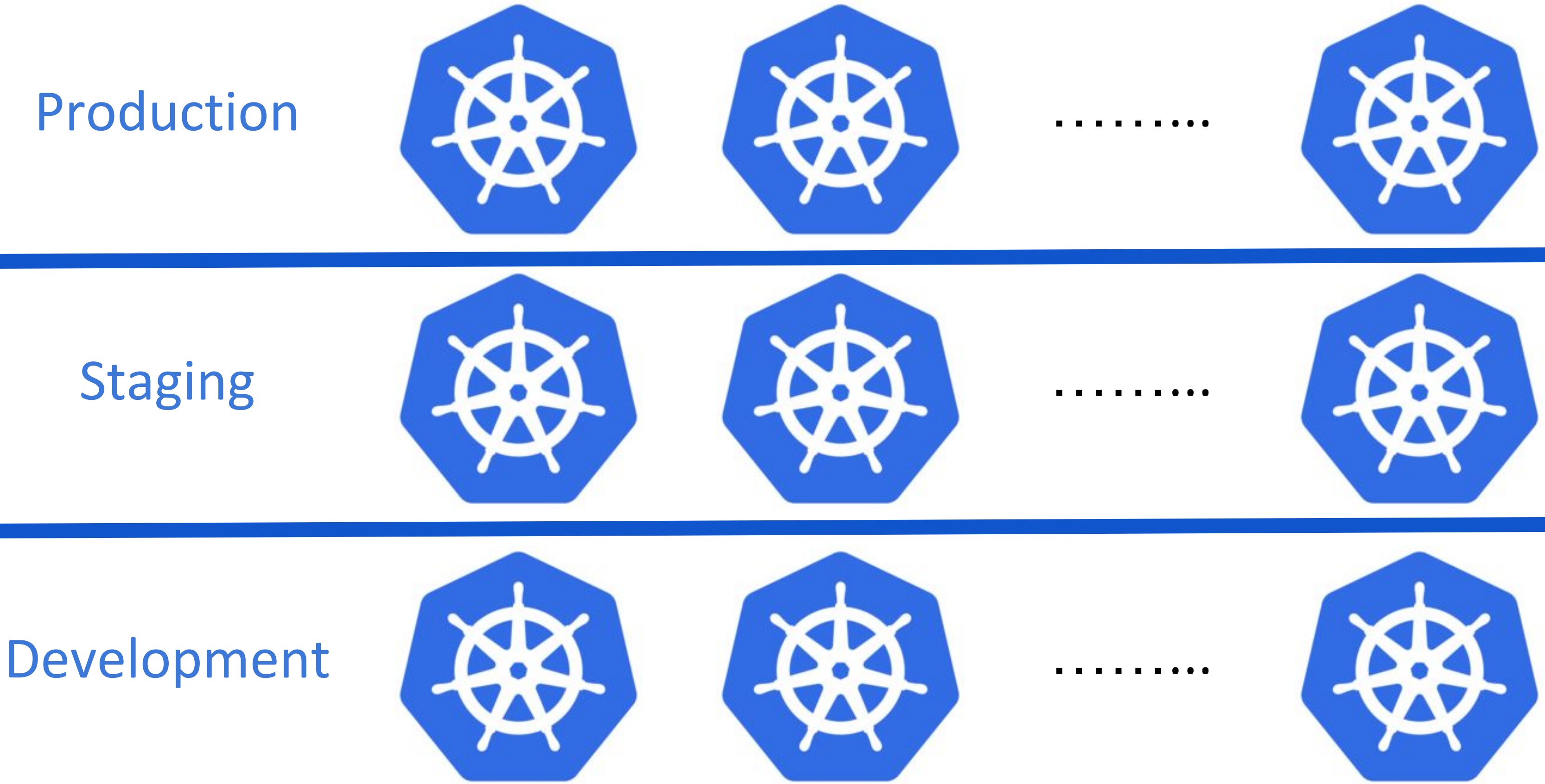
Kubernetes Clusters in Booking.com



Staging ………

Development ………

Kubernetes Clusters in Booking.com



Production

Staging

Development

Deployment Workflow

Deployment Workflow

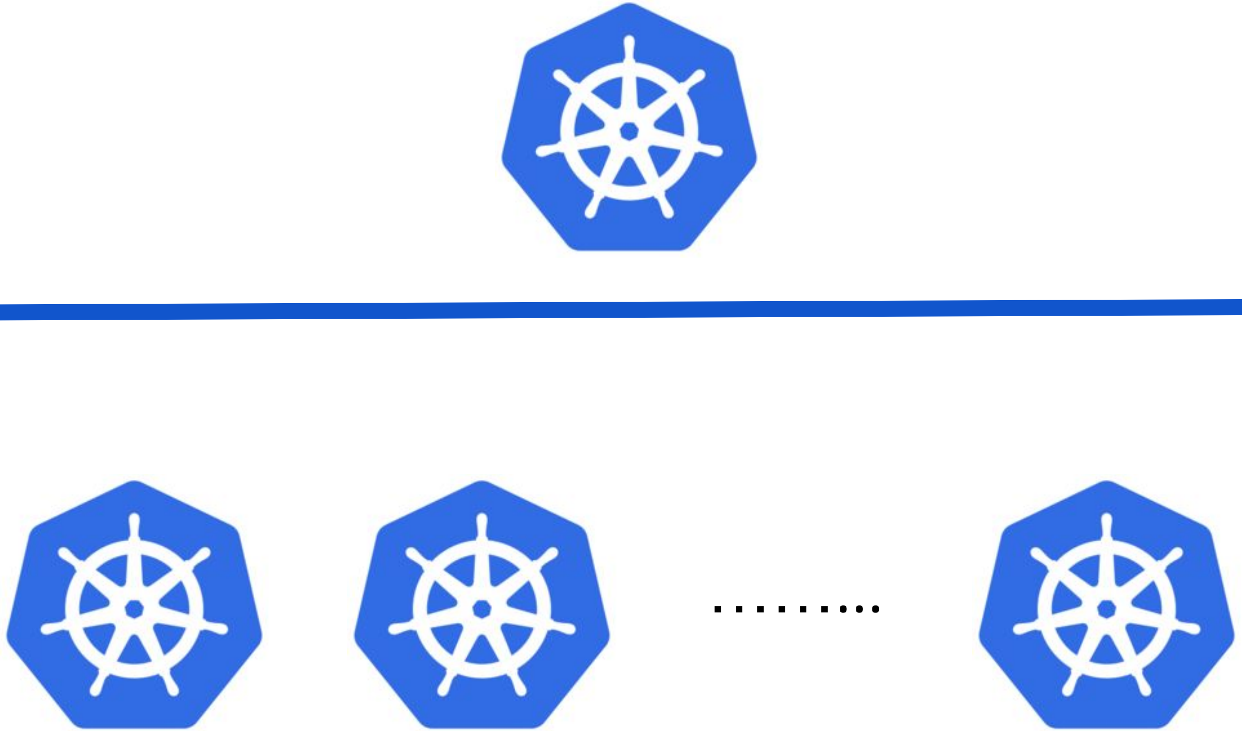
Deployment Workflow

Application

Deployment Workflow

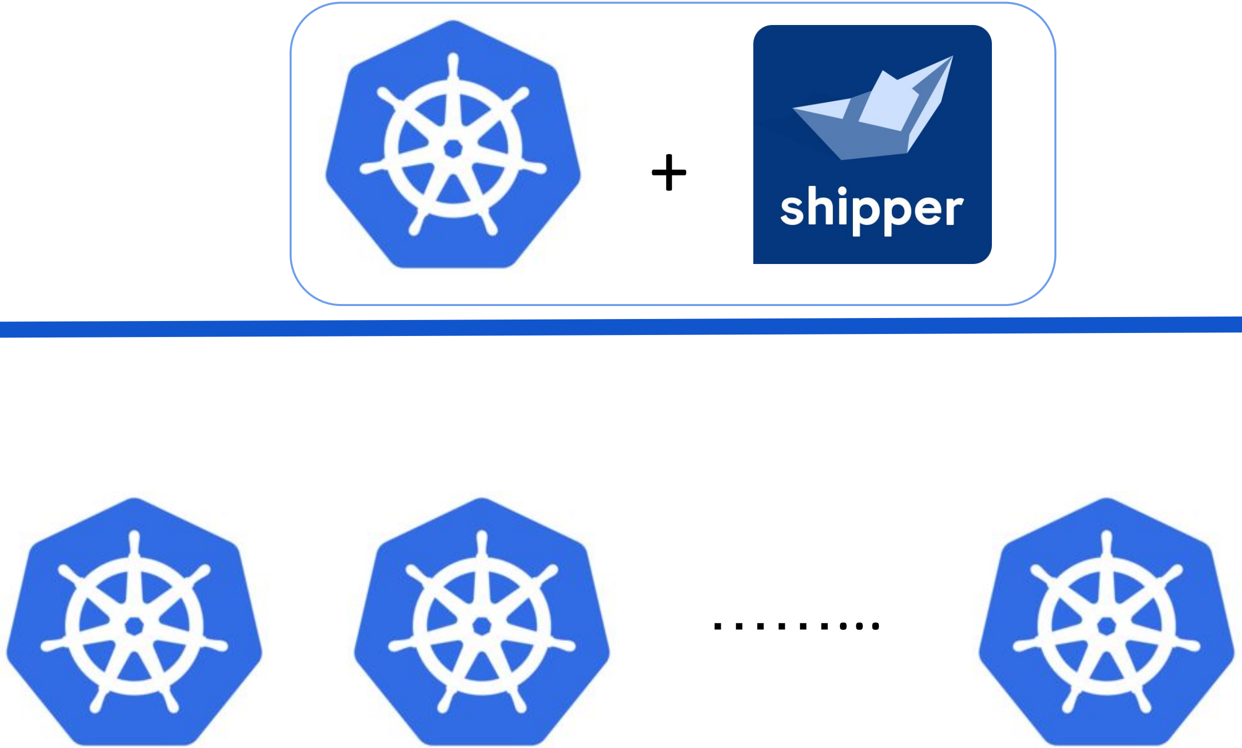
Management

Application



Deployment Workflow

Management

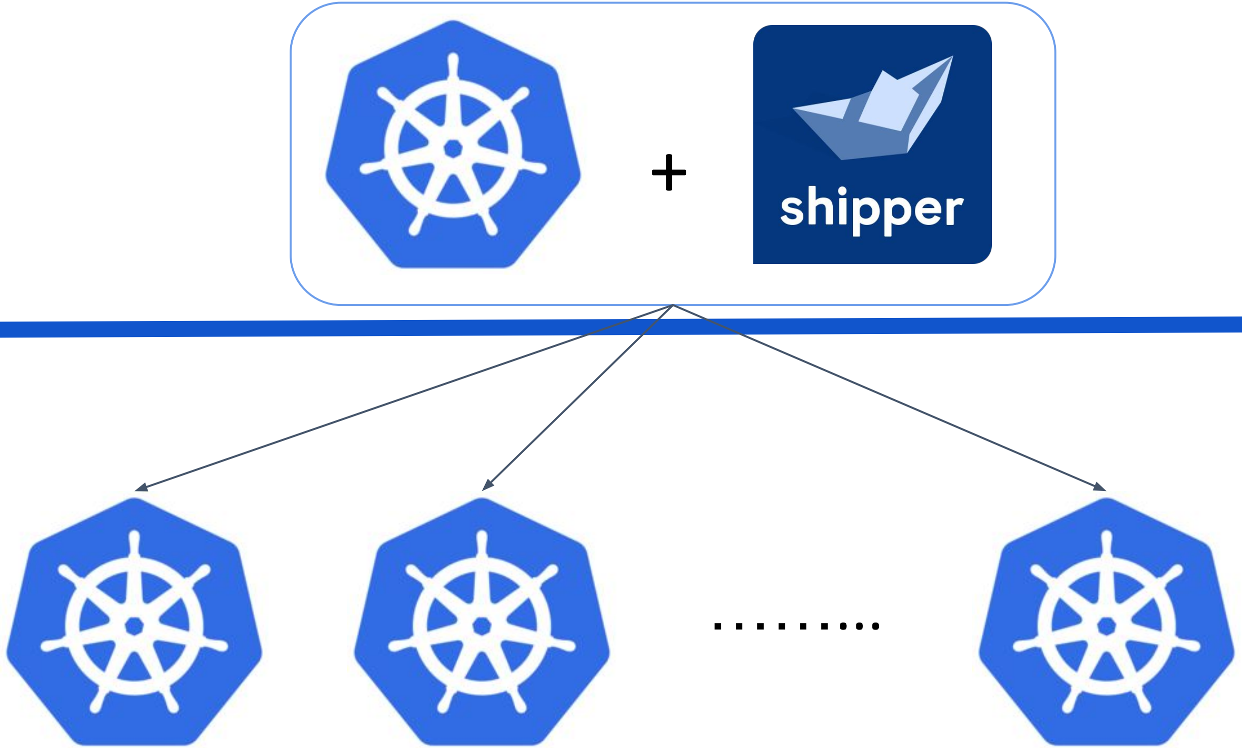


Application

Deployment Workflow

Management

Application



More info on shipper

https://shipper-k8s.io

https://docs.shipper-k8s.io

https://medium.com/booking-com-infrastructure/introducing-shipper-daf9244e3882

Challenges In Multitenant Clusters

Challenges

Challenges

Solution

Challenges

Challenges

- Project management

Solution

- Kubernetes namespaces

Challenges

Challenges

- Project management
- Resource management

Solution

- Kubernetes namespaces
- ResourceQuotas

Challenges

Challenges

- Project management
- Resource management
- Auth management

Solution

- Kubernetes namespaces
- ResourceQuotas
- Rolebindings + Auth Webhook

Challenges

Challenges	Solution
- Project management	- Kubernetes namespaces
- Resource management	- ResourceQuotas
- Auth management	- Rolebindings + Auth Webhook
- Config management	- Configmaps

Challenges

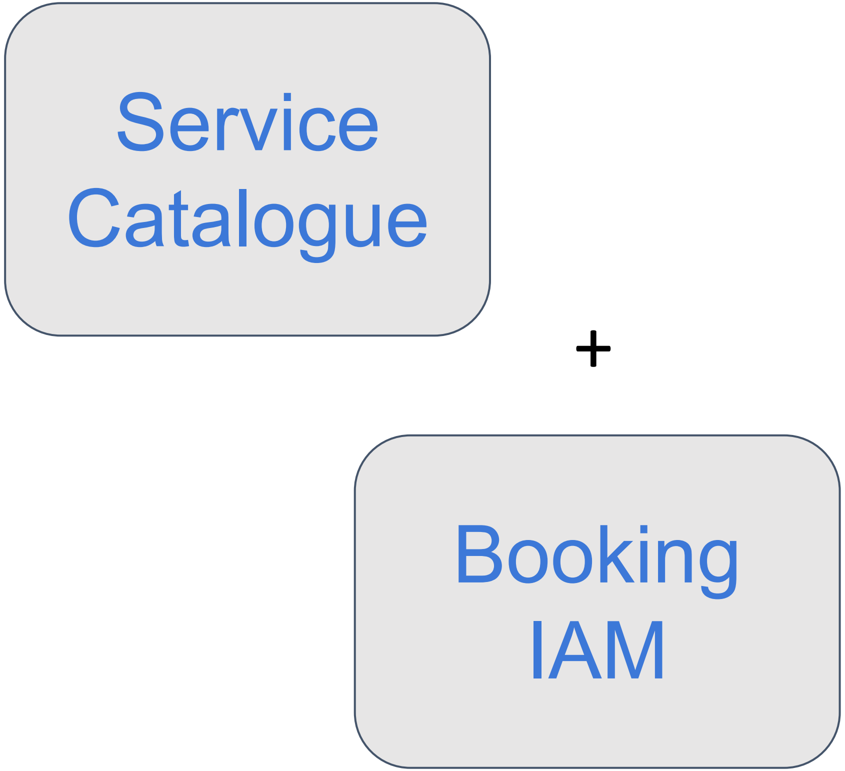
Challenges	Solution
- Project management	- Kubernetes namespaces
- Resource management	- ResourceQuotas
- Auth management	- Rolebindings + Auth Webhook
- Config management	- Configmaps
- Validation and safeguards	- Admission Webhooks + PSP

Workspace Provisioning

Workspace Provisioning

Service Catalogue

Workspace Provisioning

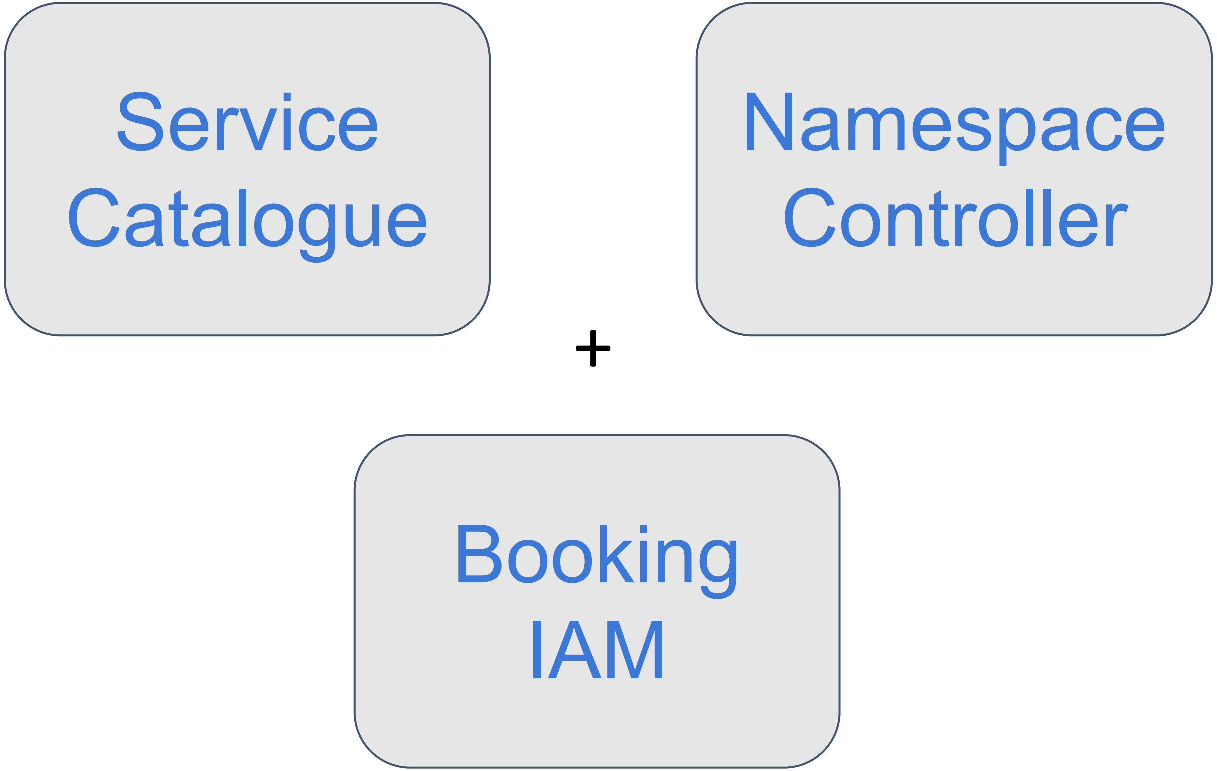


Service Catalogue

+

Booking IAM

Workspace Provisioning



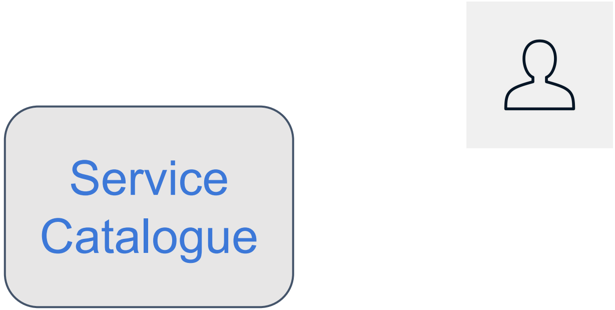
Service Catalogue

Namespace Controller

+

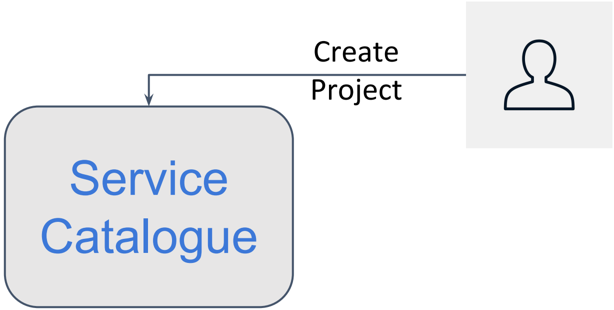
Booking IAM

Workspace Provisioning

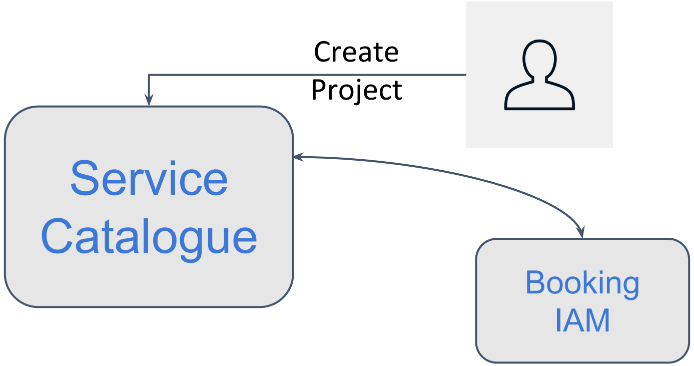


Service Catalogue

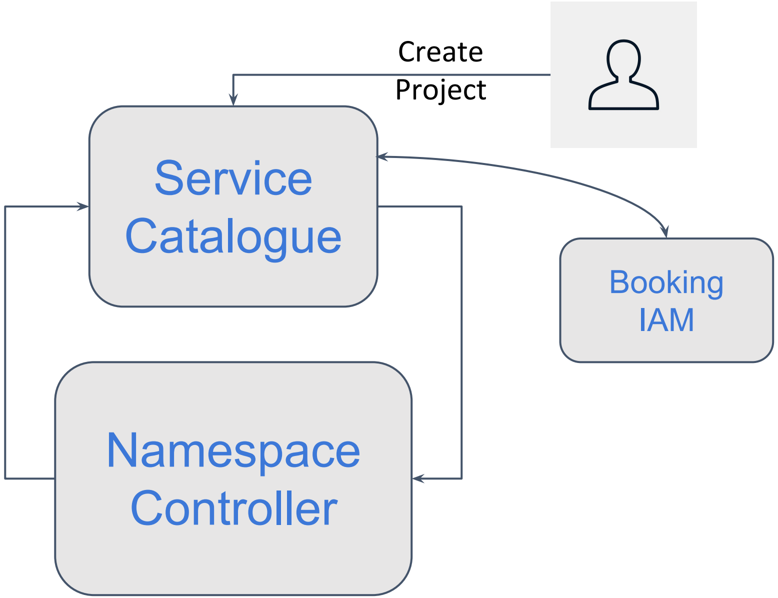
Workspace Provisioning



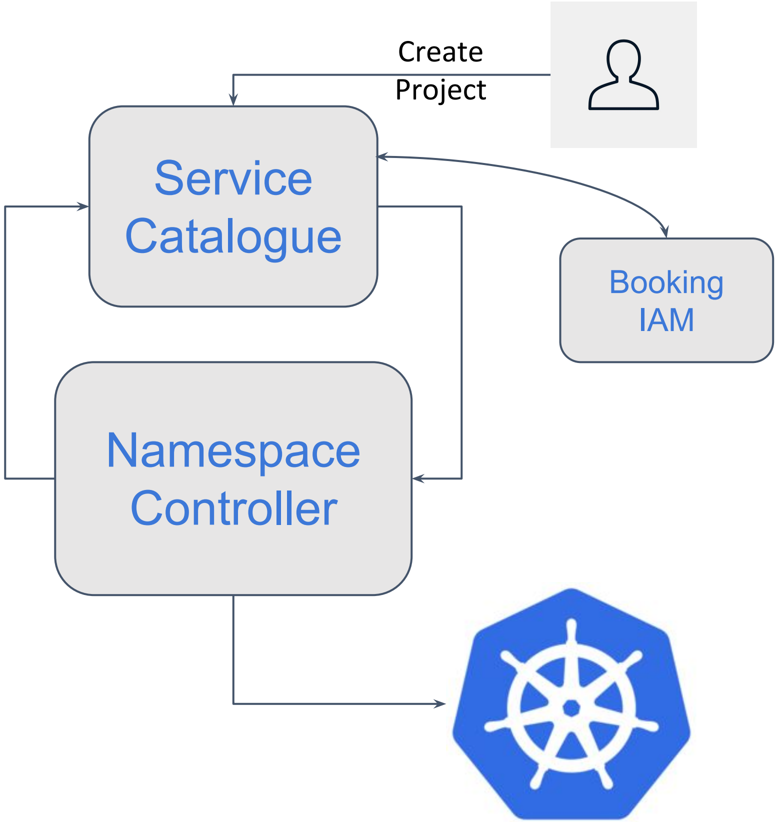
Workspace Provisioning



Workspace Provisioning



Workspace Provisioning



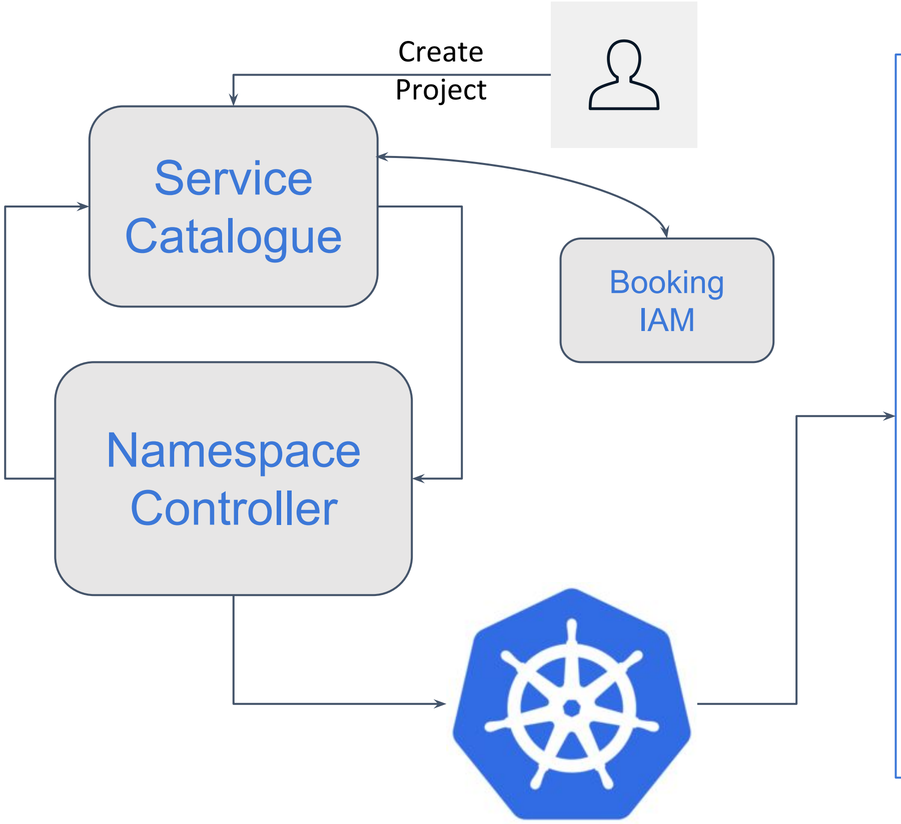
Create Project

Service Catalogue

Booking IAM

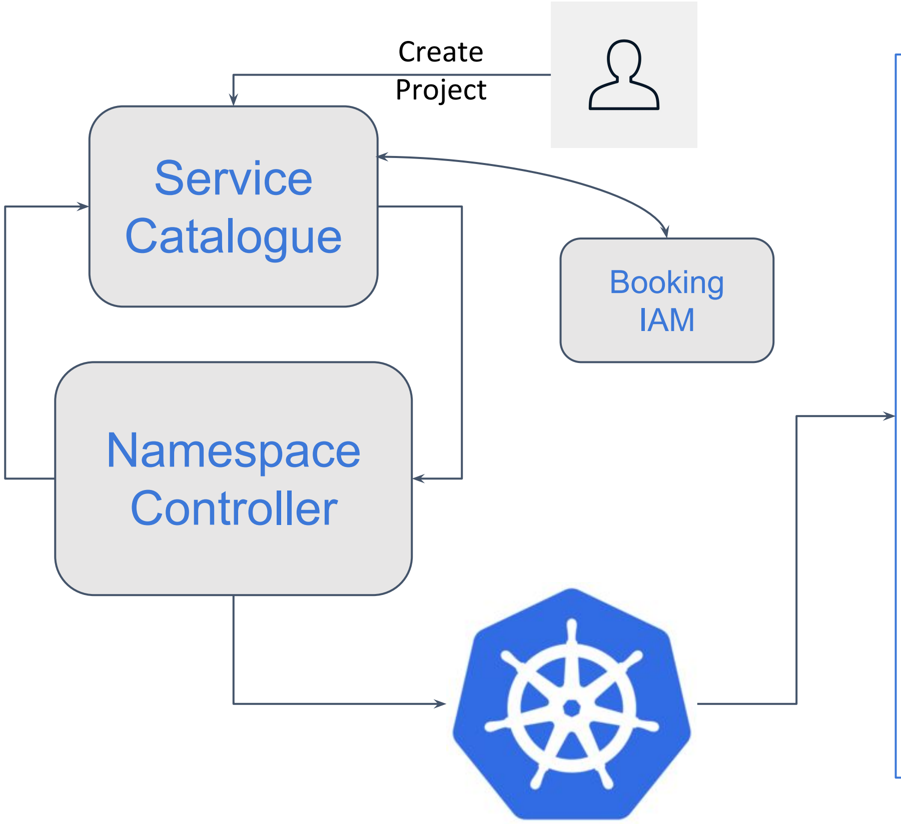
Namespace Controller

Workspace Provisioning



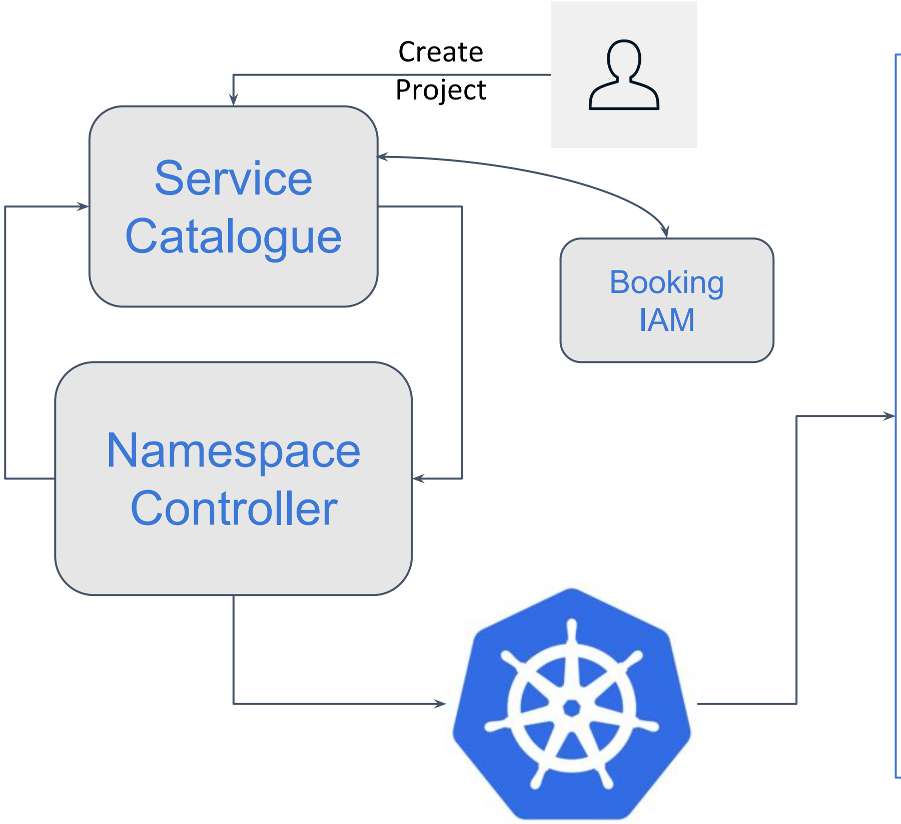
- Creates Namespace

Workspace Provisioning



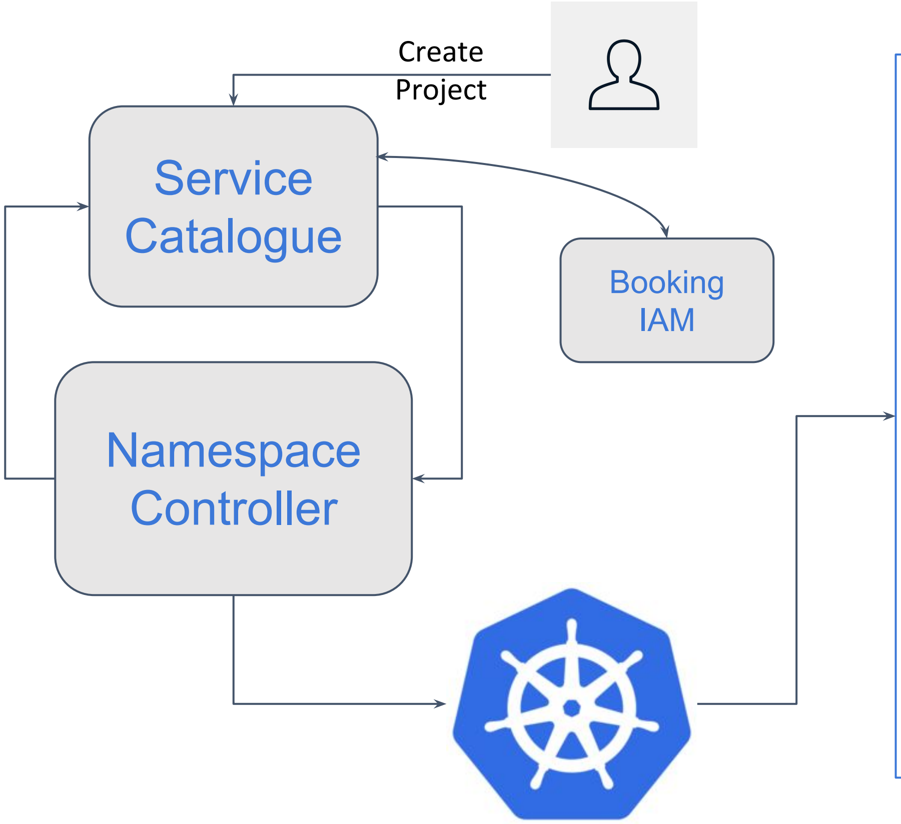
- Creates Namespace
- Creates Rolebinding

Workspace Provisioning



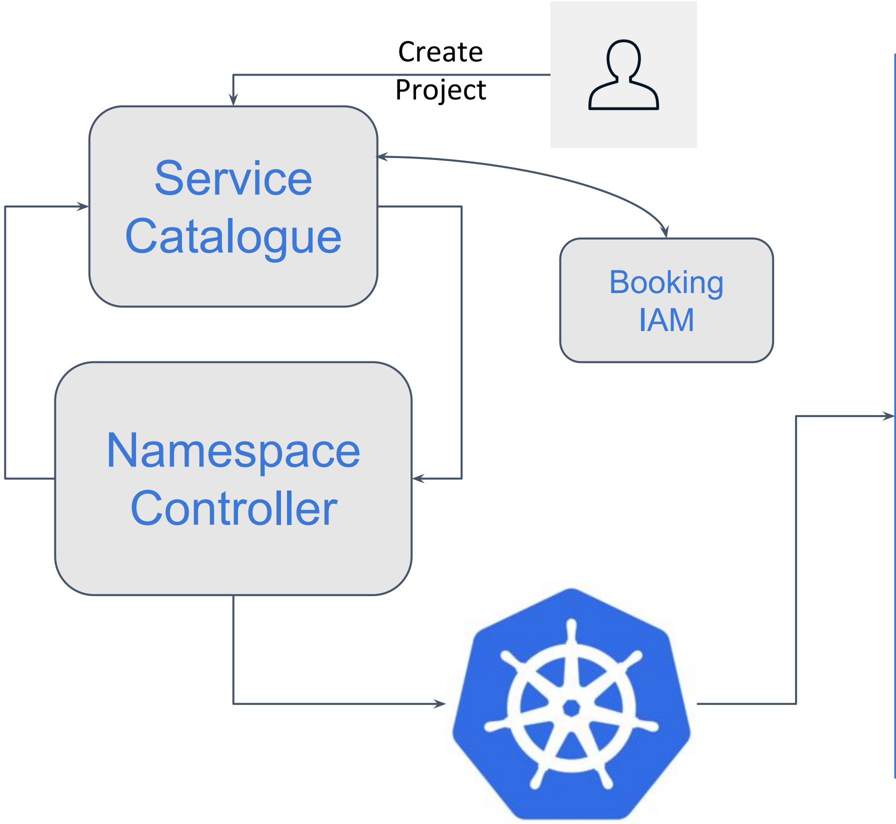
- Creates Namespace
- Creates Rolebinding
- Creates ResourceQuota

Workspace Provisioning



- Creates Namespace
- Creates Rolebinding
- Creates ResourceQuota
- Creates LimitRanges

Workspace Provisioning



- Creates Namespace
- Creates Rolebinding
- Creates ResourceQuota
- Creates LimitRanges
- Creates Configmaps

Lifecycle of a kubectl command

pod.yaml

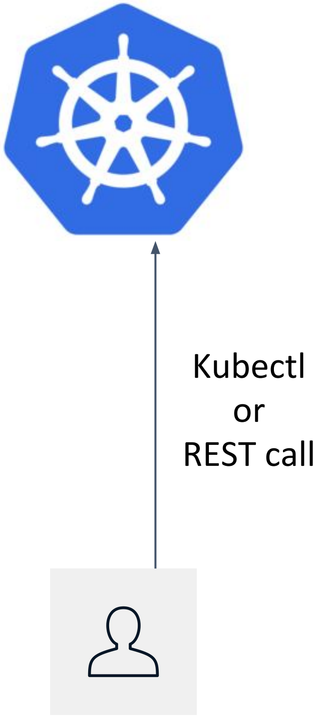
```
apiVersion: v1
kind: Pod
metadata:
  name: busybox
  namespace: bikerental
spec:
  containers:
  - image: busybox
    imagePullPolicy: IfNotPresent
    name: busybox
```

Lifecycle of a kubectl command

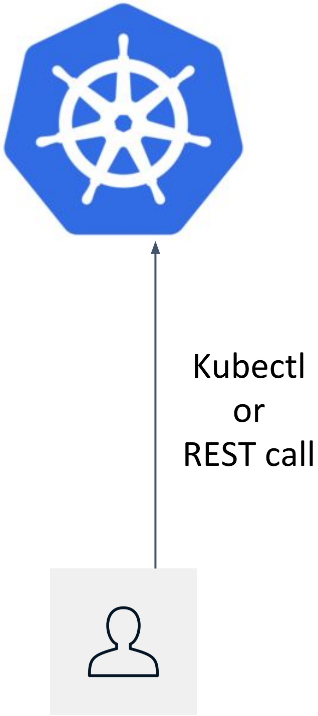
```
kubectl  create  -f  pod.yaml
```

Lifecycle of a kubectl command

Lifecycle of a kubectl command



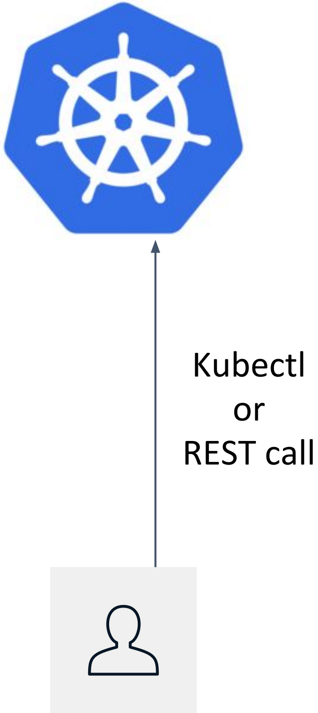
Lifecycle of a kubectl command



exec-credential Plugin

```
apiVersion: v1
clusters:
- cluster:
    server: https://auth.example.com
  name: my-cluster
contexts:
- context:
    cluster: my-cluster
    namespace: bikerental
    user: example
  name: my-context
current-context: my-context
kind: Config
users:
- name: sanjary
  user:
    exec:
      apiVersion: client.authentication.k8s.io/v1alpha1
      command: /usr/local/bin/generate-bearer-token
```

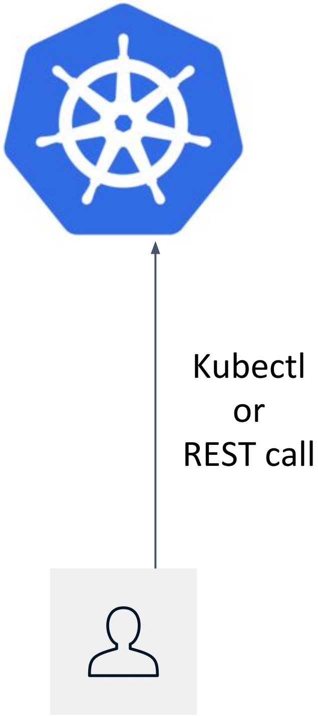
Lifecycle of a kubectl command



exec-credential Plugin

```
apiVersion: v1
clusters:
- cluster:
    server: https://auth.example.com
  name: my-cluster
contexts:
- context:
    cluster: my-cluster
    namespace: bikerental
    user: example
  name: my-context
current-context: my-context
kind: Config
users:
- name: sanjary
  user:
    exec:
      apiVersion: client.authentication.k8s.io/v1alpha1
      command: /usr/local/bin/generate-bearer-token
```

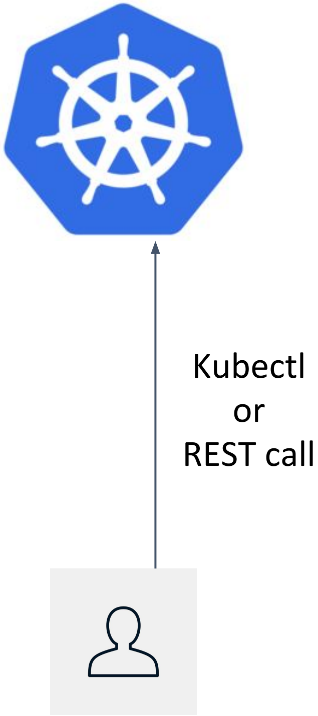
Lifecycle of a kubectl command



exec-credential Plugin

```
apiVersion: v1
clusters:
- cluster:
    server: https://auth.example.com
  name: my-cluster
contexts:
- context:
    cluster: my-cluster
    namespace: bikerental
    user: example
  name: my-context
current-context: my-context
kind: Config
users:
- name: sanjary
  user:
    exec:
      apiVersion: client.authentication.k8s.io/v1alpha1
      command: /usr/local/bin/generate-bearer-token
```

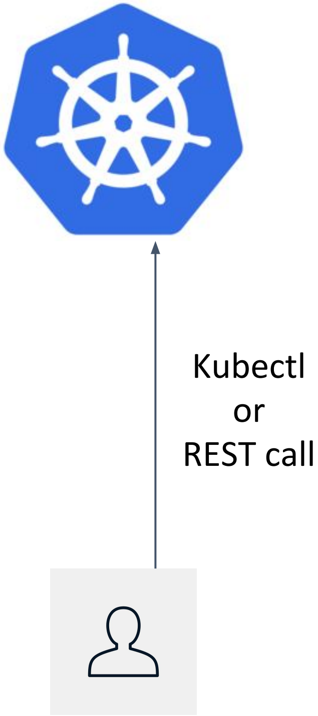
Lifecycle of a kubectl command



```
Example output:
```

```
{
  "apiVersion": "client.authentication.k8s.io/v1beta1",
  "kind": "ExecCredential",
  "status": {
    "token": "my-bearer-token",
    "expirationTimestamp": "2019-23-05T17:30:20-08:00"
  }
}
```

Lifecycle of a kubectl command

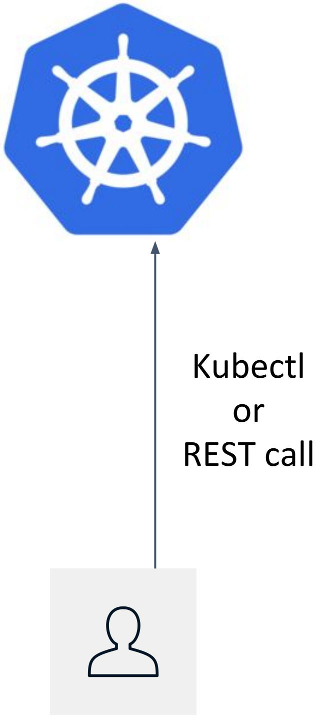


Kubectl
or
REST call

`Example output:`

```
{
  "apiVersion": "client.authentication.k8s.io/v1beta1",
  "kind": "ExecCredential",
  "status": {
    "token": "my-bearer-token",
    "expirationTimestamp": "2019-23-05T17:30:20-08:00"
  }
}
```

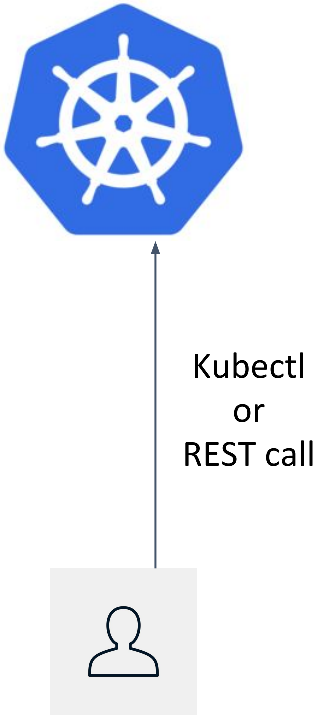
Lifecycle of a kubectl command



Example output:

```
{
  "apiVersion": "client.authentication.k8s.io/v1beta1",
  "kind": "ExecCredential",
  "status": {
    "token": "my-bearer-token",
    "expirationTimestamp": "2019-23-05T17:30:20-08:00"
  }
}
```

Lifecycle of a kubectl command

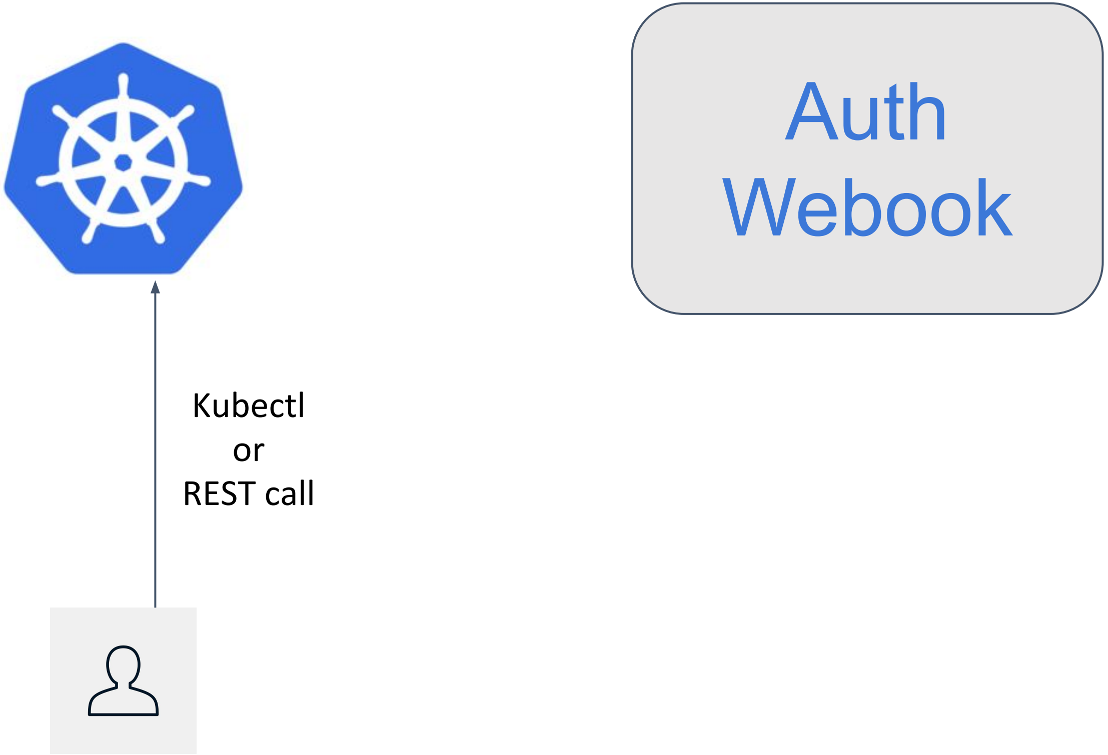


Kubectl
or
REST call

`Example output:`

```
{
  "apiVersion": "client.authentication.k8s.io/v1beta1",
  "kind": "ExecCredential",
  "status": {
    "token": "my-bearer-token",
    "expirationTimestamp": "2019-23-05T17:30:20-08:00"
  }
}
```

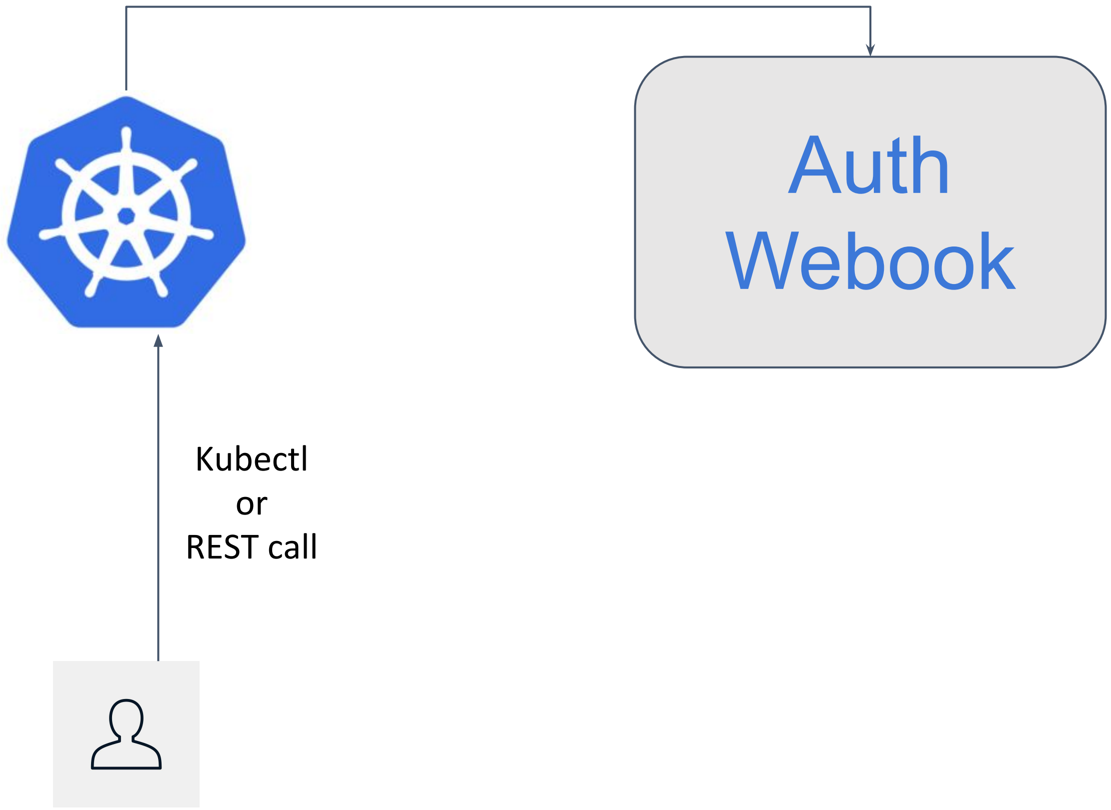
Lifecycle of a kubectl command



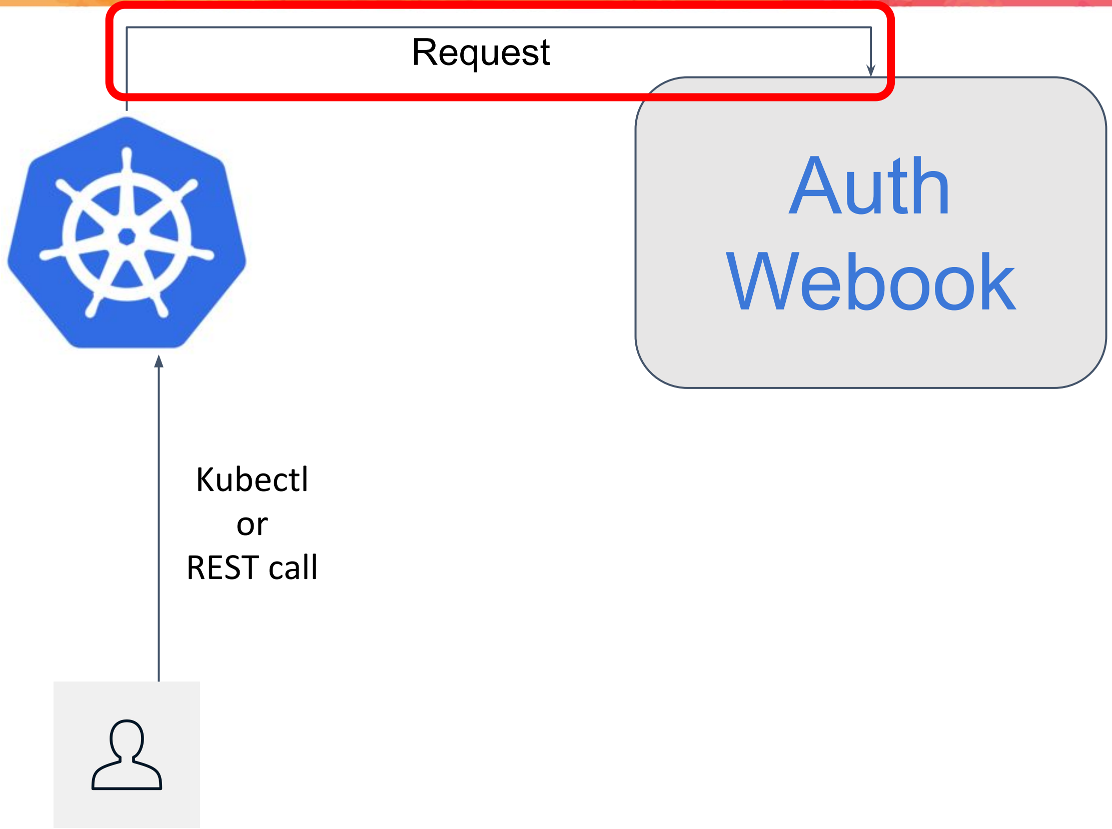
Auth Webook

Kubectl
or
REST call

Lifecycle of a kubectl command



Lifecycle of a kubectl command



Auth Webhook

Example request:

```
{
  "apiVersion": "authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "spec": {
    "token": "my-bearer-token"
  }
}
```

Auth Webhook

Example request:

```
{
  "apiVersion": "authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "spec": {
    "token": "my-bearer-token"
  }
}
```

Auth Webhook

Example request:

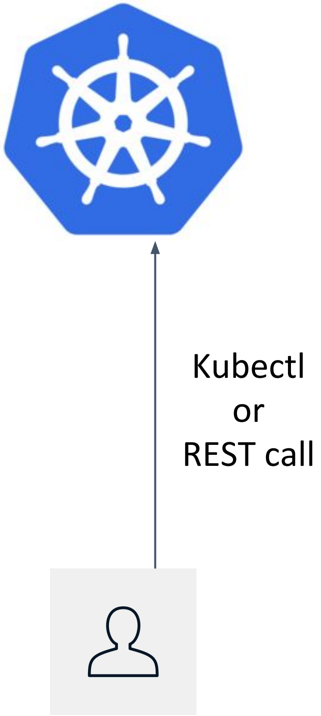
```
{
  "apiVersion": "authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "spec": {
    "token": "my-bearer-token"
  }
}
```

Auth Webhook

Example request:

```
{
  "apiVersion": "authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "spec": {
    "token": "my-bearer-token"
  }
}
```

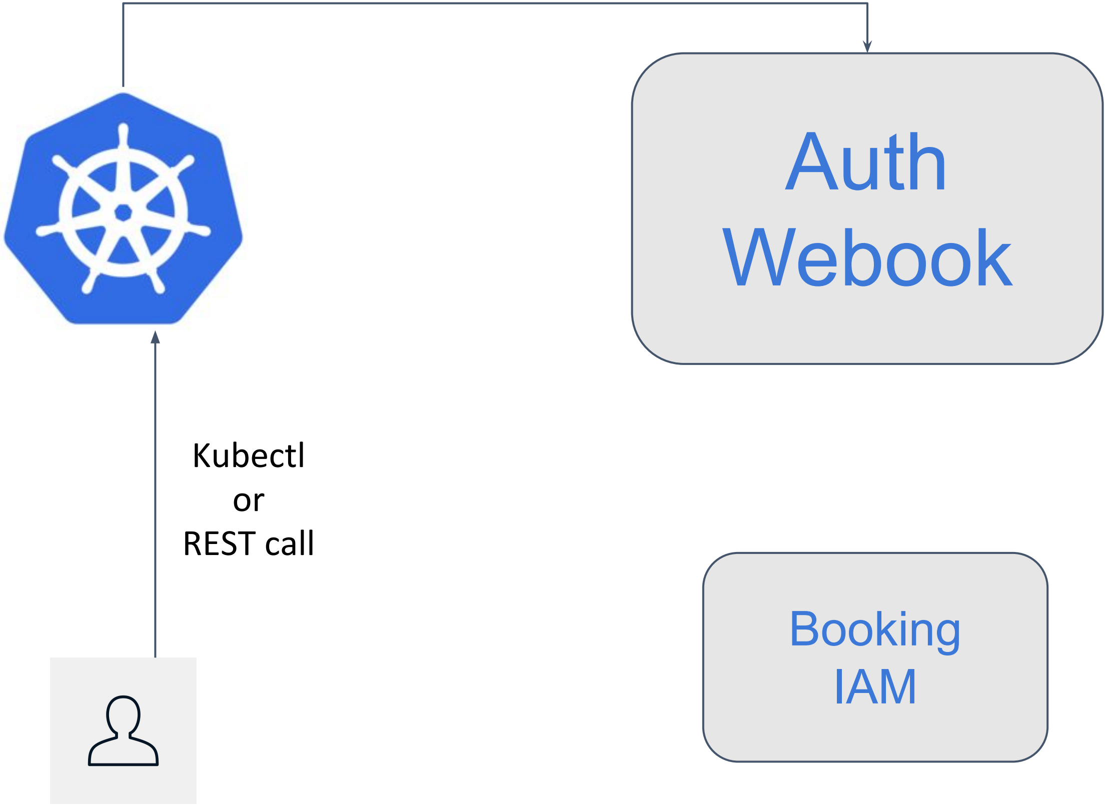
Lifecycle of a kubectl command



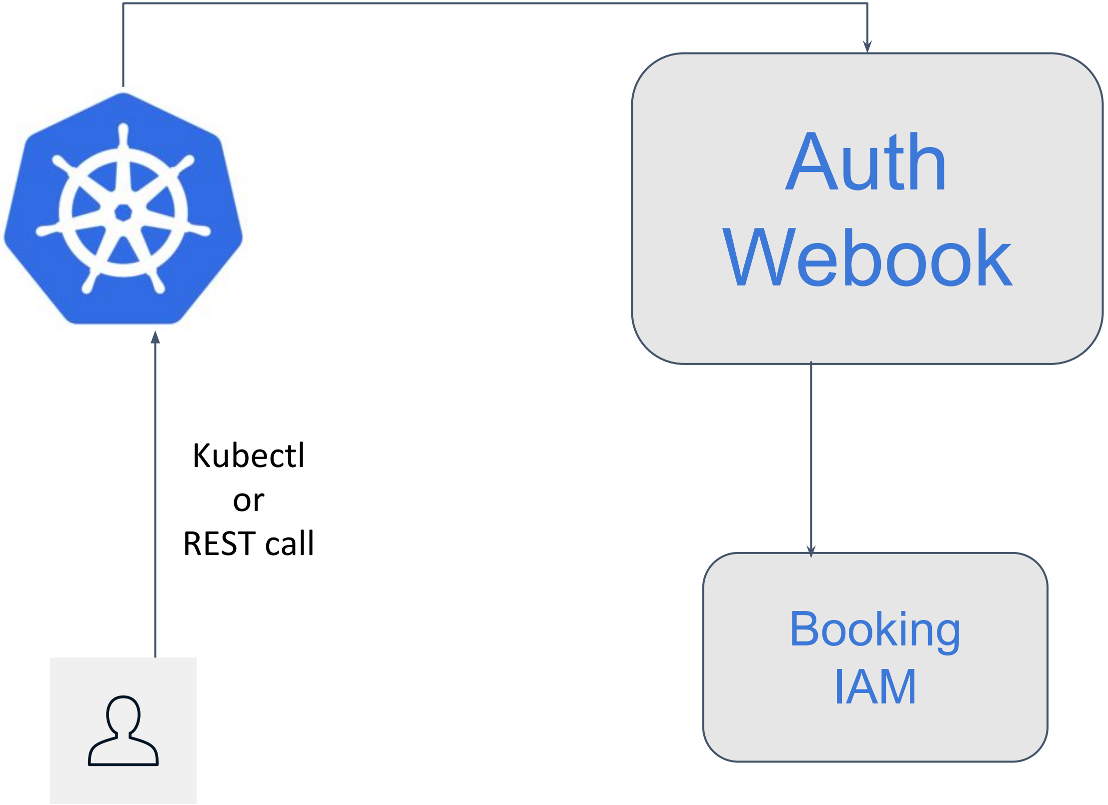
Example output:

```
{
  "apiVersion": "client.authentication.k8s.io/v1beta1",
  "kind": "ExecCredential",
  "status": {
    "token": "my-bearer-token",
    "expirationTimestamp": "2019-23-05T17:30:20-08:00"
  }
}
```

Lifecycle of a kubectl command



Lifecycle of a kubectl command

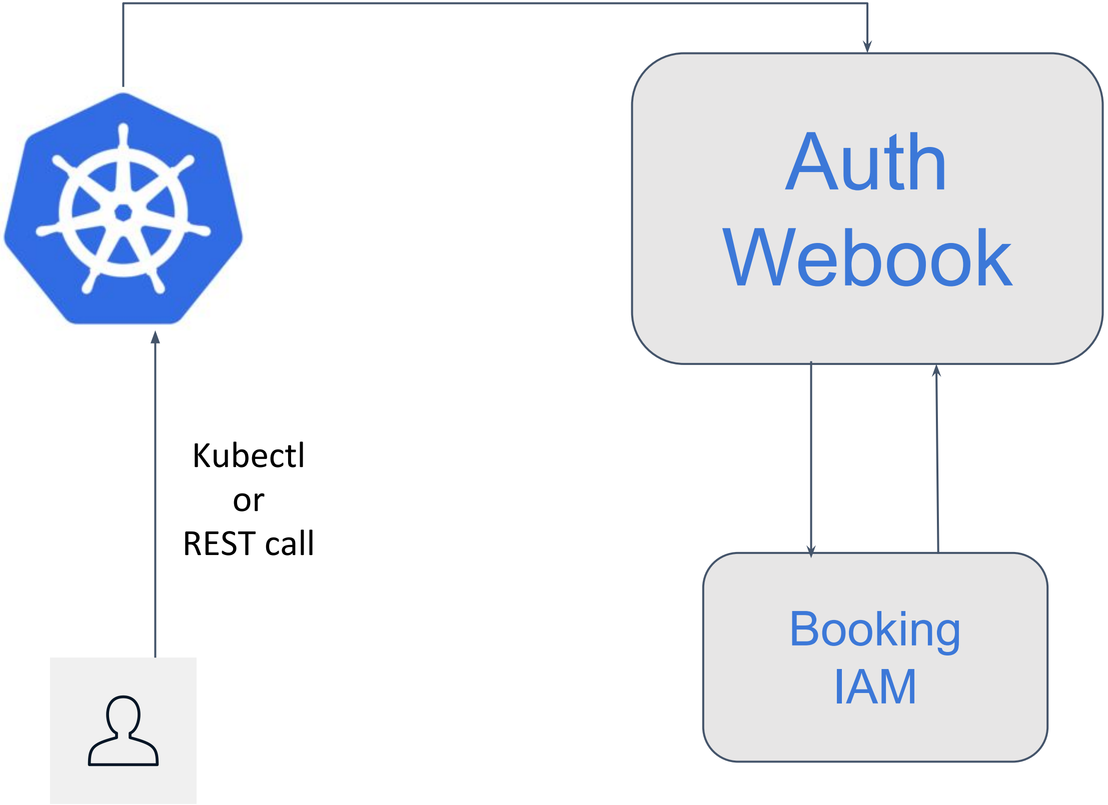


Kubectl
or
REST call

Auth
Webook

Booking
IAM

Lifecycle of a kubectl command

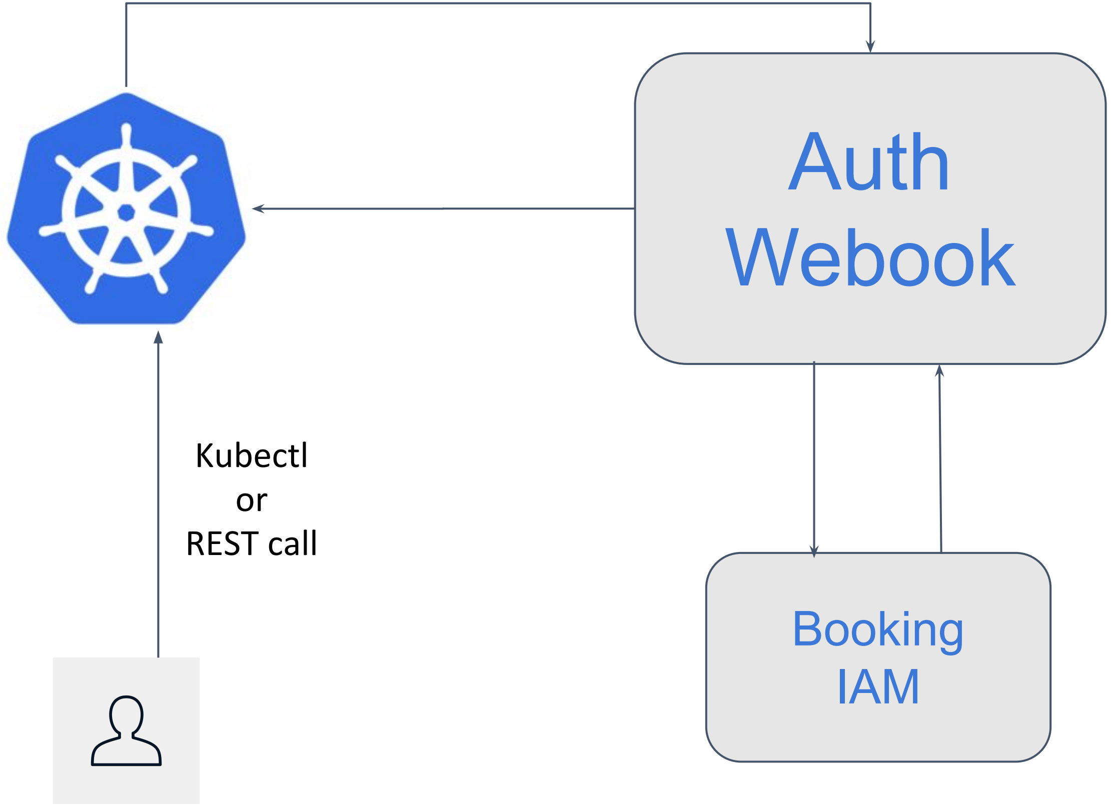


Kubectl
or
REST call

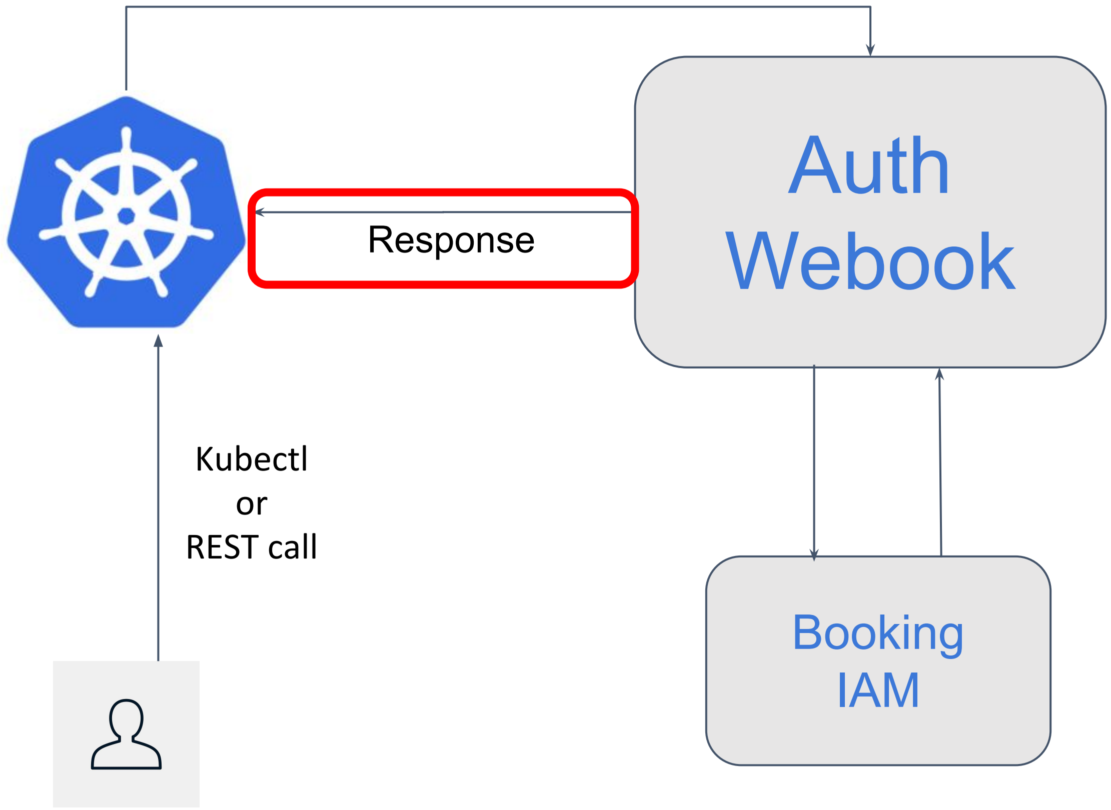
Auth
Webook

Booking
IAM

Lifecycle of a kubectl command



Lifecycle of a kubectl command



Auth Webhook

Example request:

```
{
  "apiVersion":
"authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "spec": {
    "token": "my-bearer-token"
  }
}
```

Example response:

```
Allow:
{
  "apiVersion": "authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "status": {
    "authenticated": true,
    "user": {
      "username": "sanjary",
      "uid": "1234",
      "groups": [ "bikerental:admin" ]
    }
  }
}

Deny:
{
  "apiVersion": "authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "status": {
    "authenticated": false
  }
}
```

Auth Webhook

Example request:

```
{
  "apiVersion":
"authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "spec": {
    "token": "my-bearer-token"
  }
}
```

Example response:

```
Allow:
{
  "apiVersion": "authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "status": {
    "authenticated": true,
    "user": {
      "username": "sanjary",
      "uid": "1234",
      "groups": [ "bikerental:admin" ]
    }
  }
}

Deny:
{
  "apiVersion": "authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "status": {
    "authenticated": false
  }
}
```

Auth Webhook

Example request:

```
{
  "apiVersion":
"authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "spec": {
    "token": "my-bearer-token"
  }
}
```

Example response:

```
Allow:
{
  "apiVersion": "authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "status": {
    "authenticated": true,
    "user": {
      "username": "sanjary",
      "uid": "1234",
      "groups": [ "bikerental:admin" ]
    }
  }
}

Deny:
{
  "apiVersion": "authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "status": {
    "authenticated": false
  }
}
```

Auth Webhook

Example request:

```
{
  "apiVersion":
"authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "spec": {
    "token": "my-bearer-token"
  }
}
```

Example response:

```
Allow:
{
  "apiVersion": "authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "status": {
    "authenticated": true,
    "user": {
      "username": "sanjary",
      "uid": "1234",
      "groups": [ "bikerental:admin" ]
    }
  }
}

Deny:
{
  "apiVersion": "authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "status": {
    "authenticated": false
  }
}
```

Auth Webhook

Example request:

```
{
  "apiVersion":
"authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "spec": {
    "token": "my-bearer-token"
  }
}
```

Example response:

```
Allow:
{
  "apiVersion": "authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "status": {
    "authenticated": true,
    "user": {
      "username": "sanjary",
      "uid": "1234",
      "groups": [ "bikerental:admin" ]
    }
  }
}

Deny:
{
  "apiVersion": "authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "status": {
    "authenticated": false
  }
}
```

Auth Webhook

Example request:

```
{
  "apiVersion":
"authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "spec": {
    "token": "my-bearer-token"
  }
}
```

Example response:

Allow:

```
{
  "apiVersion": "authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "status": {
    "authenticated": true,
    "user": {
      "username": "sanjary",
      "uid": "1234",
      "groups": [ "bikerental:admin" ]
    }
  }
}
```

Deny:

```
{
  "apiVersion": "authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "status": {
    "authenticated": false
  }
}
```

Auth Webhook

Example request:

```
{
  "apiVersion":
"authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "spec": {
    "token": "my-bearer-token"
  }
}
```

Example response:

```
Allow:
{
  "apiVersion": "authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "status": {
    "authenticated": true,
    "user": {
      "username": "sanjary",
      "uid": "1234",
      "groups": [ "bikerental:admin" ]
    }
  }
}
```

```
Deny:
{
  "apiVersion": "authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "status": {
    "authenticated": false
  }
}
```

Auth Webhook

Example request:

```
{
  "apiVersion":
"authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "spec": {
    "token": "my-bearer-token"
  }
}
```

Example response:

```
Allow:
{
  "apiVersion": "authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "status": {
    "authenticated": true,
    "user": {
      "username": "sanjary",
      "uid": "1234",
      "groups": [ "bikerental:admin" ]
    }
  }
}
```

```
Deny:
{
  "apiVersion": "authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "status": {
    "authenticated": false
  }
}
```

Auth Webhook

Example request:

```
{
  "apiVersion":
"authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "spec": {
    "token": "my-bearer-token"
  }
}
```

Example response:

```
Allow:
{
  "apiVersion": "authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "status": {
    "authenticated": true,
    "user": {
      "username": "sanjary",
      "uid": "1234",
      "groups": [ "bikerental:admin" ]
    }
  }
}

Deny:
{
  "apiVersion": "authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "status": {
    "authenticated": false
  }
}
```

Auth Webhook

Example request:

```
{
  "apiVersion":
"authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "spec": {
    "token": "my-bearer-token"
  }
}
```

Example response:

```
Allow:
{
  "apiVersion": "authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "status": {
    "authenticated": true,
    "user": {
      "username": "sanjary",
      "uid": "1234",
      "groups": [ "bikerental:admin" ]
    }
  }
}

Deny:
{
  "apiVersion": "authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "status": {
    "authenticated": false
  }
}
```

Auth Webhook

Example request:

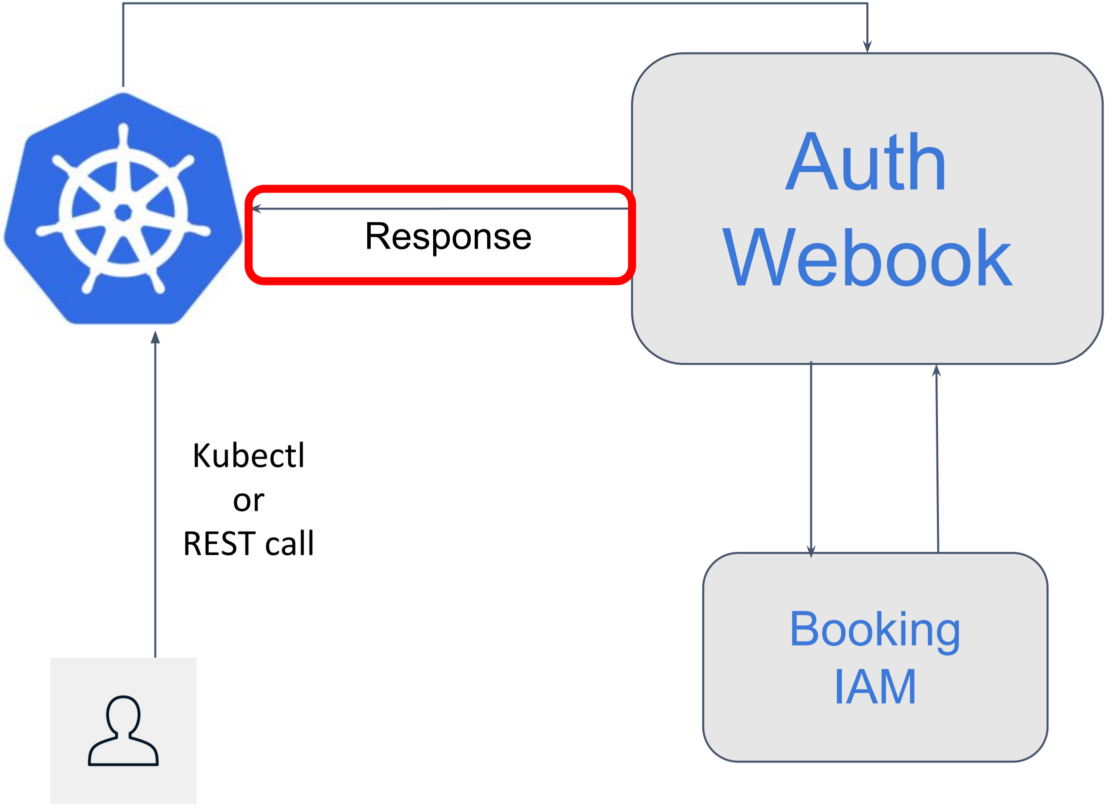
```
{
  "apiVersion":
"authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "spec": {
    "token": "my-bearer-token"
  }
}
```

Example response:

```
Allow:
{
  "apiVersion": "authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "status": {
    "authenticated": true,
    "user": {
      "username": "sanjary",
      "uid": "1234",
      "groups": [ "bikerental:admin" ]
    }
  }
}

Deny:
{
  "apiVersion": "authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "status": {
    "authenticated": false
  }
}
```

Lifecycle of a kubectl command



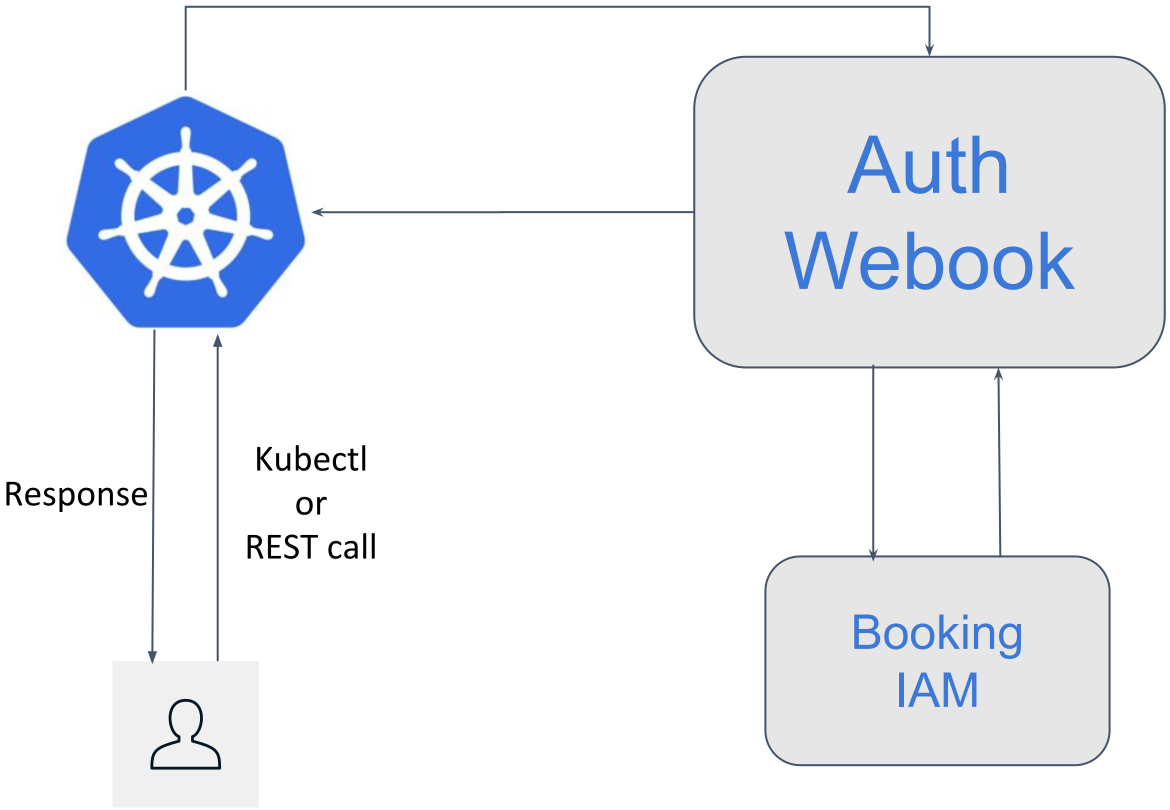
Response

Auth Webook

Kubectl or REST call

Booking IAM

Lifecycle of a kubectl command



Auth Webhook

Configure
kube-apiserver with the
webhook flag

Auth Webhook

Configure kube-apiserver with the webhook flag

```
--authentication-token-webhook-config=<SOME_FILENAME>
```

```
--authentication-token-webhook-cache-ttl=2m (default)
```

Auth Webhook

Configure kube-apiserver with the webhook flag

```
--authentication-token-webhook-config=<SOME_FILENAME>
```

```
--authentication-token-webhook-cache-ttl=2m (default)
```

```
apiVersion: v1
kind: Config
clusters:
  - name: name-of-remote-auth-service
    cluster:
      certificate-authority: /path/to/ca.pem
      # Webhook URL (must be https)
      server: https://auth.example.com/auth
users:
  - name: name-of-api-server
    user:
      client-certificate: /path/to/cert.pem
      client-key: /path/to/key.pem
contexts:
- context:
    cluster: name-of-remote-auth-service
    user: name-of-api-server
  name: auth-webhook
current-context: auth-webhook
```

Auth Webhook

Configure kube-apiserver with the webhook flag

`--authentication-token-webhook-config=<SOME_FILENAME>`

`--authentication-token-webhook-cache-ttl=2m (default)`

```
apiVersion: v1
kind: Config
clusters:
  - name: name-of-remote-auth-service
    cluster:
      certificate-authority: /path/to/ca.pem
      # Webhook URL (must be https)
      server: https://auth.example.com/auth
users:
  - name: name-of-api-server
    user:
      client-certificate: /path/to/cert.pem
      client-key: /path/to/key.pem
contexts:
- context:
    cluster: name-of-remote-auth-service
    user: name-of-api-server
  name: auth-webhook
current-context: auth-webhook
```

Auth Webhook

Configure kube-apiserver with the webhook flag

```
--authentication-token-webhook-config=<SOME_FILENAME>
```

```
--authentication-token-webhook-cache-ttl=2m (default)
```

```
apiVersion: v1
kind: Config
clusters:
  - name: name-of-remote-auth-service
    cluster:
      certificate-authority: /path/to/ca.pem
      # Webhook URL (must be https)
      server: https://auth.example.com/auth
users:
  - name: name-of-api-server
    user:
      client-certificate: /path/to/cert.pem
      client-key: /path/to/key.pem
contexts:
- context:
    cluster: name-of-remote-auth-service
    user: name-of-api-server
  name: auth-webhook
current-context: auth-webhook
```

Auth Webhook

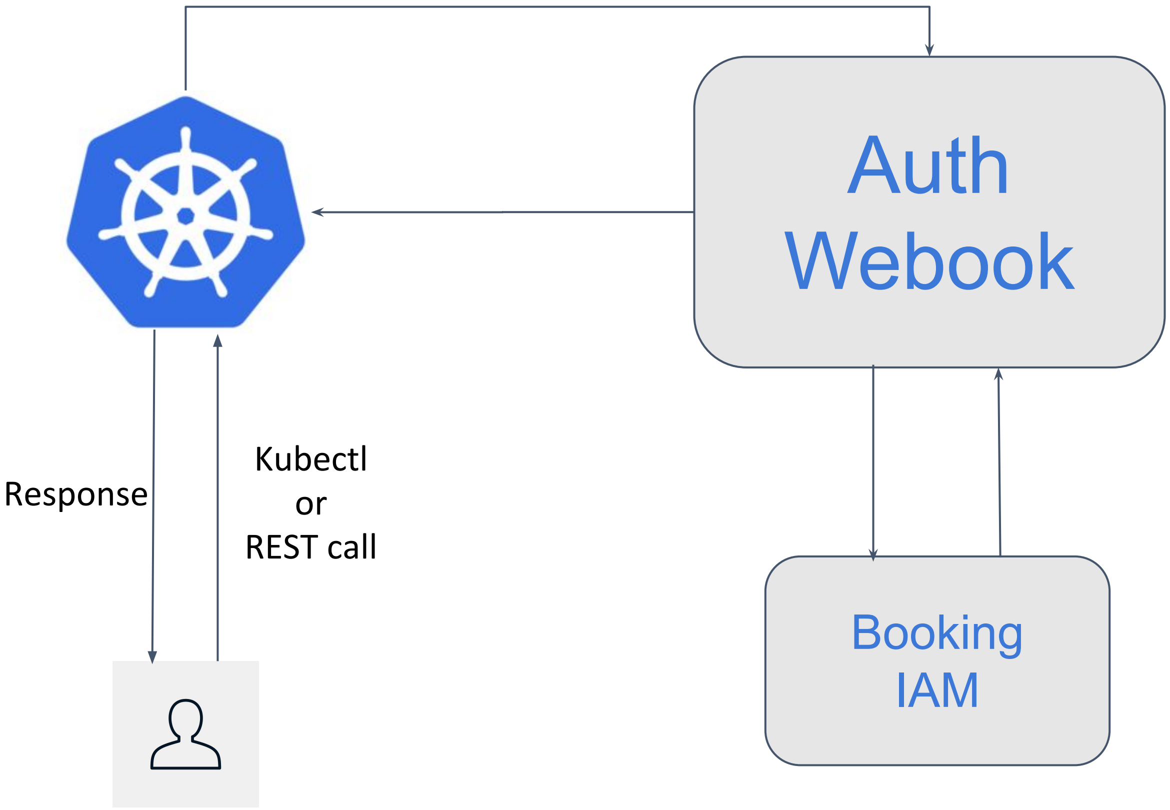
Configure kube-apiserver with the webhook flag

```
--authentication-token-webhook-config=<SOME_FILENAME>
```

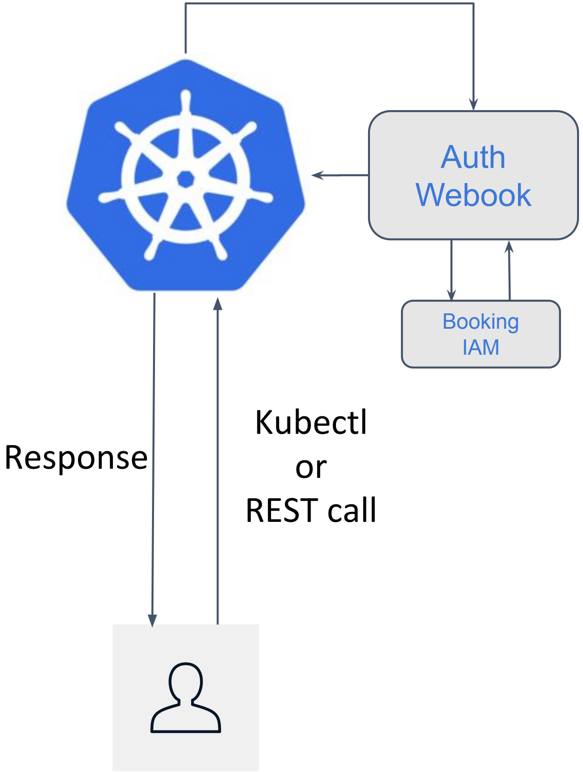
```
--authentication-token-webhook-cache-ttl=2m (default)
```

```
apiVersion: v1
kind: Config
clusters:
  - name: name-of-remote-auth-service
    cluster:
      certificate-authority: /path/to/ca.pem
      # Webhook URL (must be https)
      server: https://auth.example.com/auth
users:
  - name: name-of-api-server
    user:
      client-certificate: /path/to/cert.pem
      client-key: /path/to/key.pem
contexts:
- context:
    cluster: name-of-remote-auth-service
    user: name-of-api-server
  name: auth-webhook
current-context: auth-webhook
```

Lifecycle of a kubectl command



Lifecycle of a kubectl command



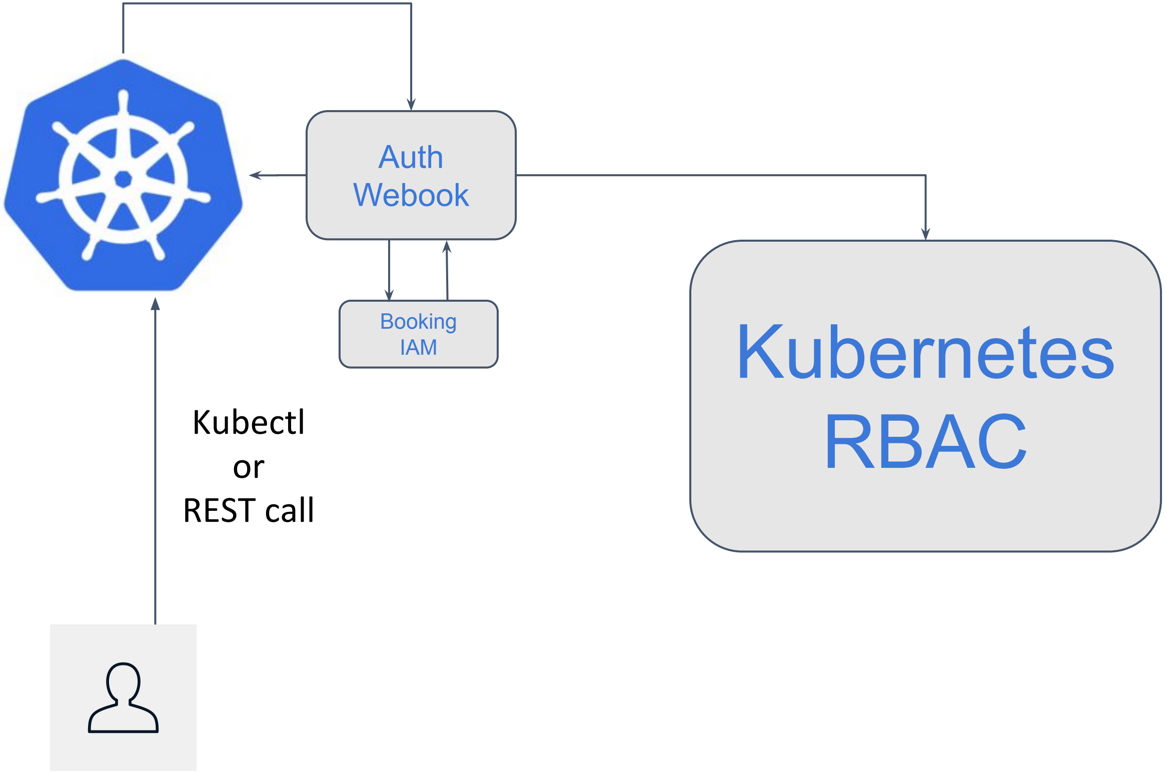
Auth Webook

Booking IAM

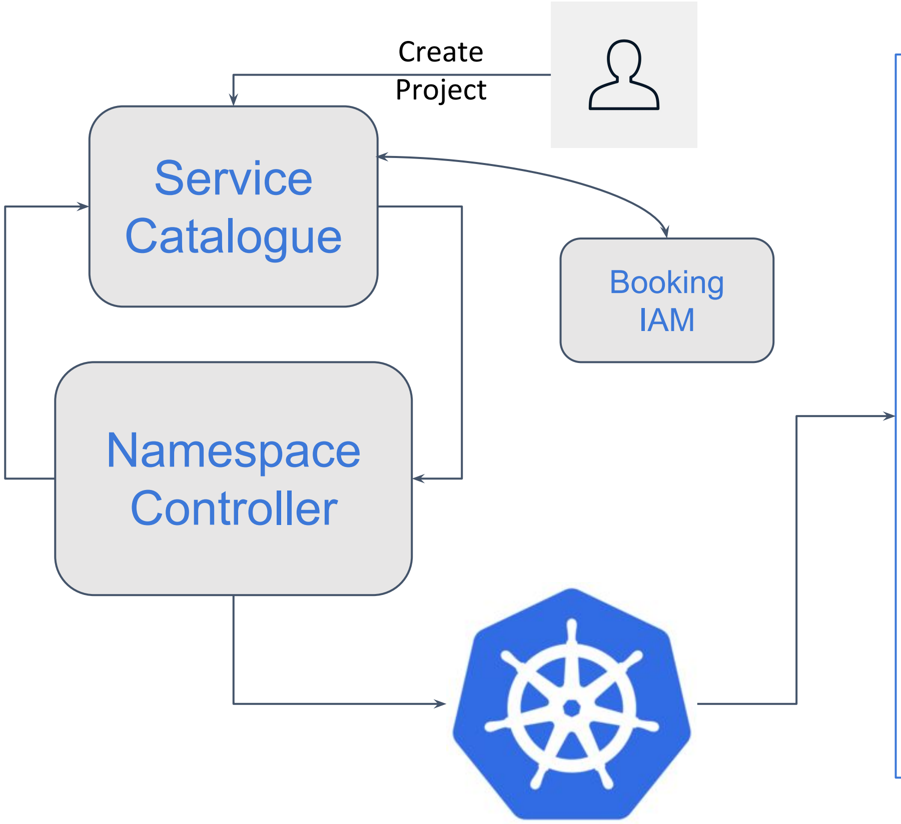
Response

Kubectl or REST call

Lifecycle of a kubectl command

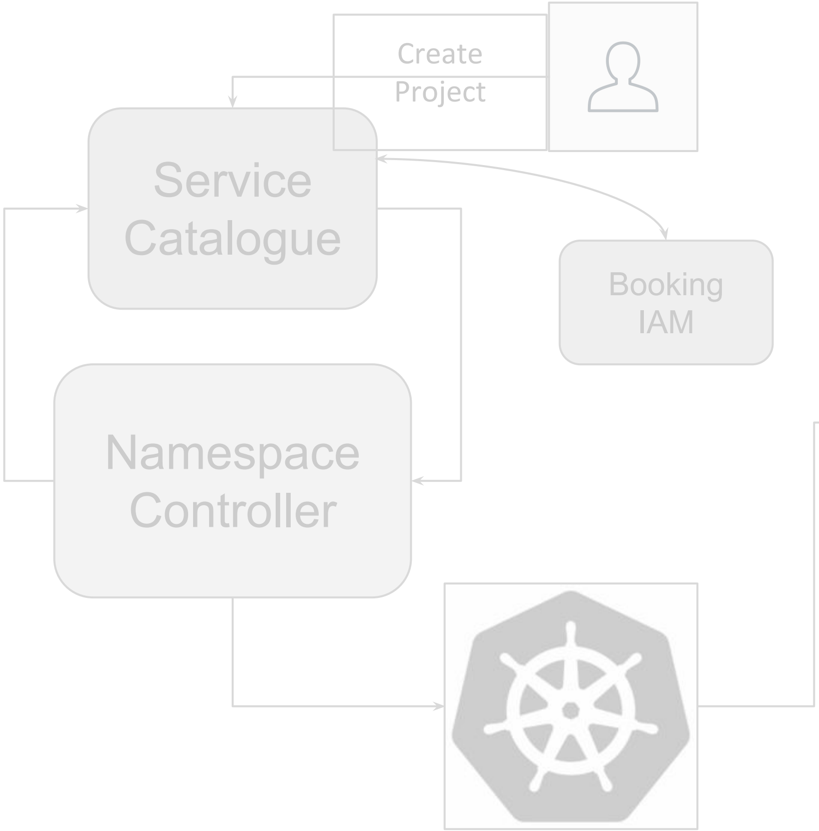


Workspace Provisioning (Recap)



- Creates Namespace
- Creates Rolebinding
- Creates ResourceQuota
- Creates LimitRanges
- Creates Configmaps

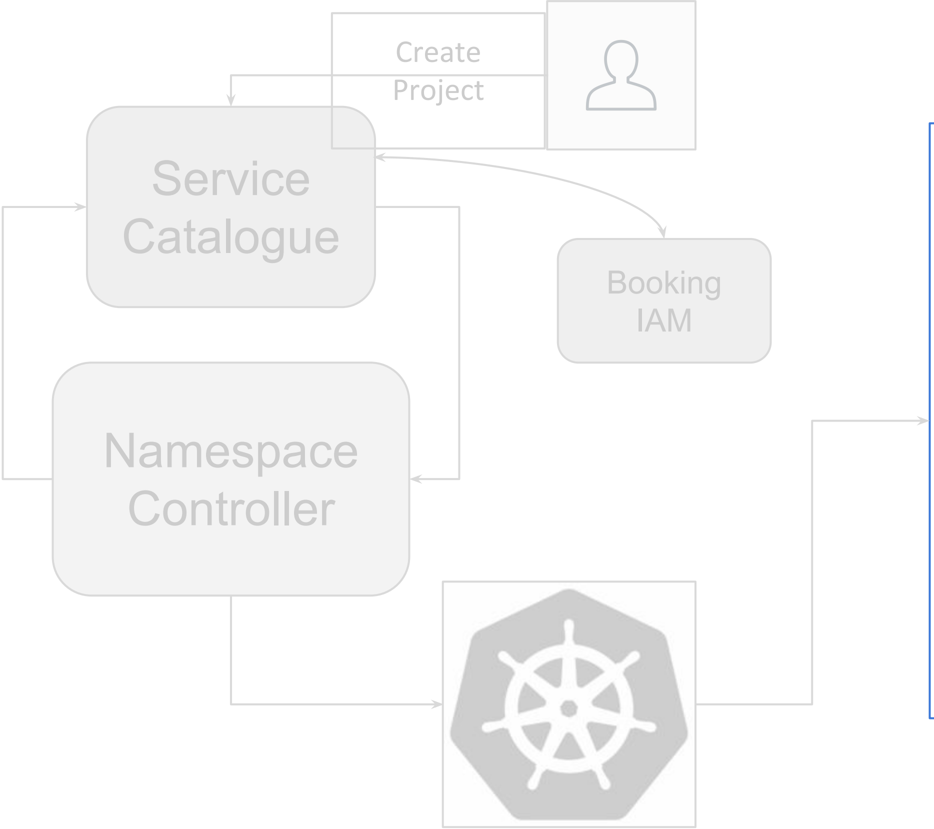
Auth Webhook + Namespace Controller



Rolebinding:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: admin
  namespace: bikerental
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: admin
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: bikerental:admin
```

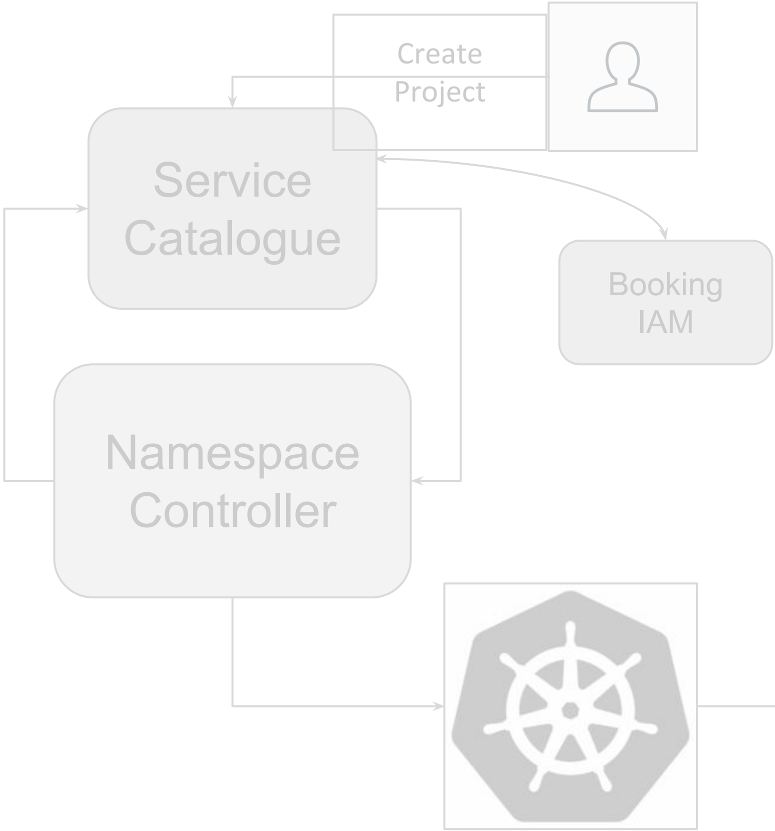
Auth Webhook + Namespace Controller



Rolebinding:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: admin
  namespace: bikerental
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: admin
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: bikerental:admin
```

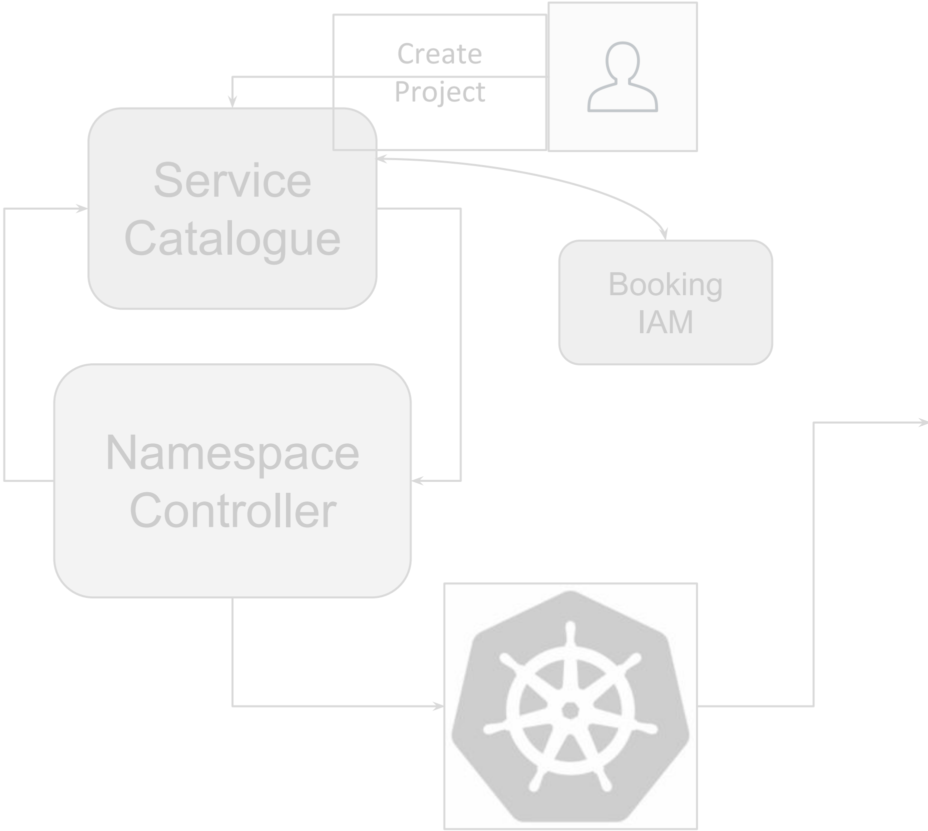
Auth Webhook + Namespace Controller



Rolebinding:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: admin
  namespace: bikerental
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: admin
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: bikerental:admin
```

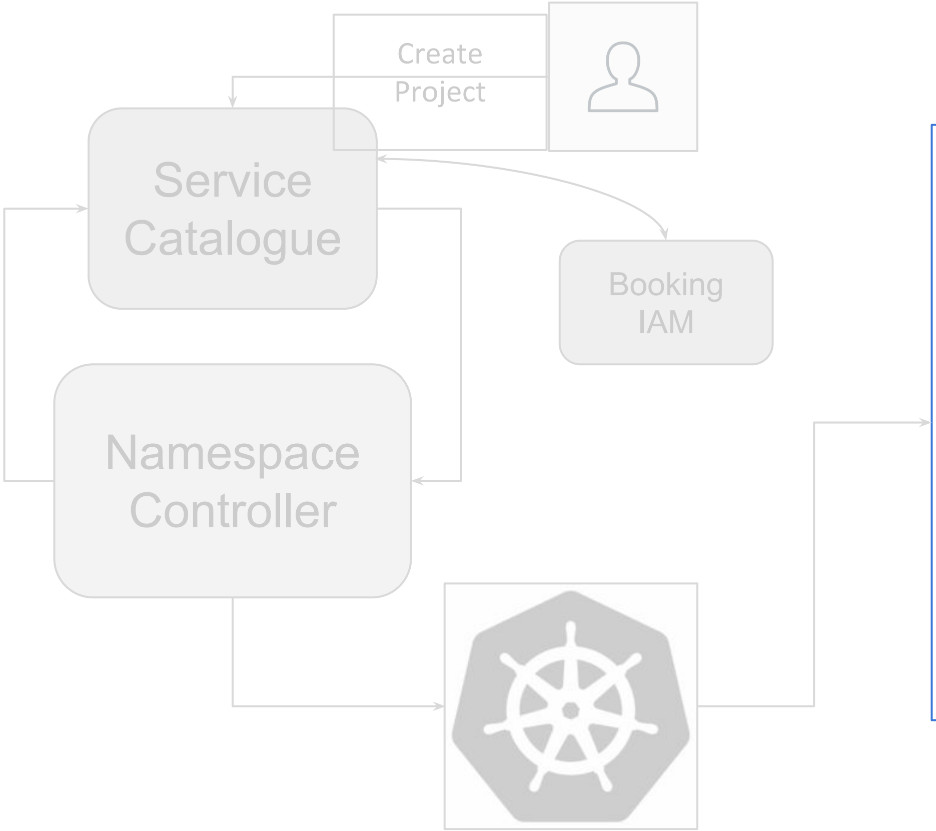
Auth Webhook + Namespace Controller



Rolebinding:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: admin
  namespace: bikerental
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: admin
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: bikerental:admin
```

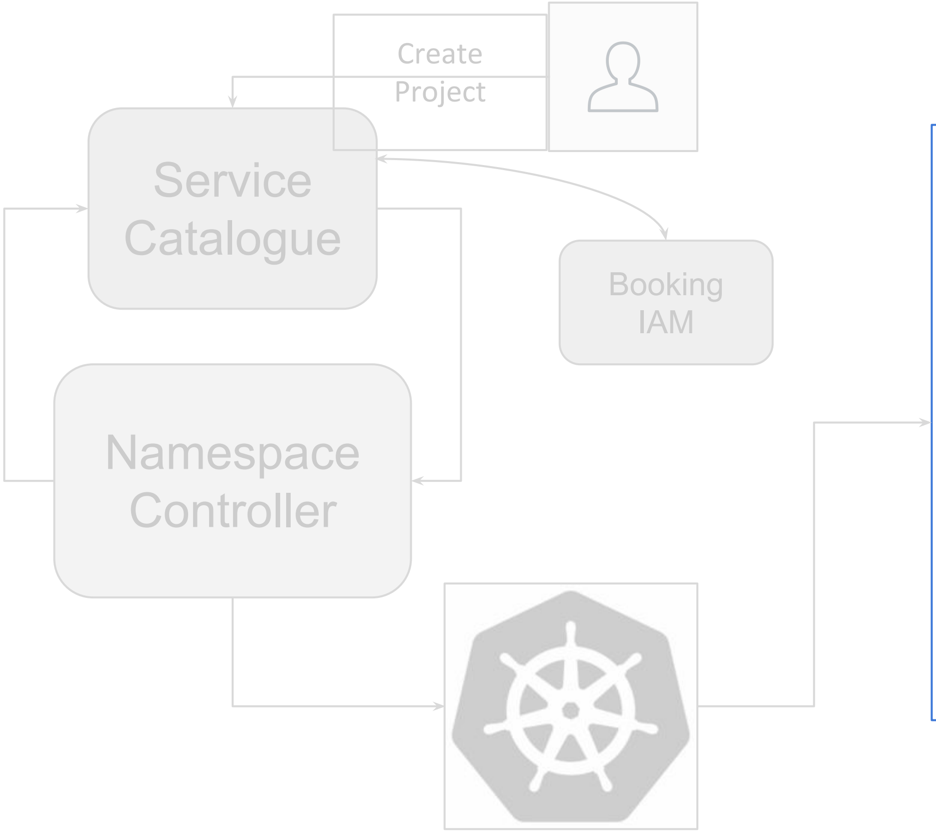
Auth Webhook + Namespace Controller



Rolebinding:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: admin
  namespace: bikerental
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: admin
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: bikerental:admin
```

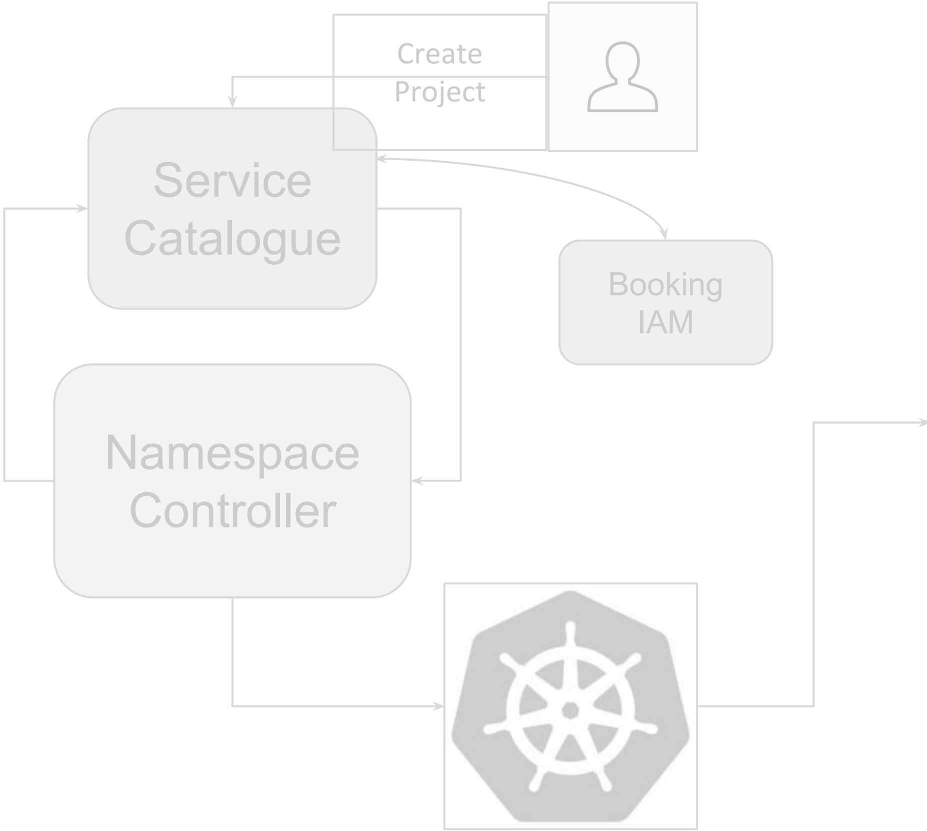
Auth Webhook + Namespace Controller



Rolebinding:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: admin
  namespace: bikerental
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: admin
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: bikerental:admin
```

Auth Webhook + Namespace Controller



Rolebinding:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: admin
  namespace: bikerental
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: admin
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: bikerental:admin
```

Auth Webhook (Recap)

Example request:

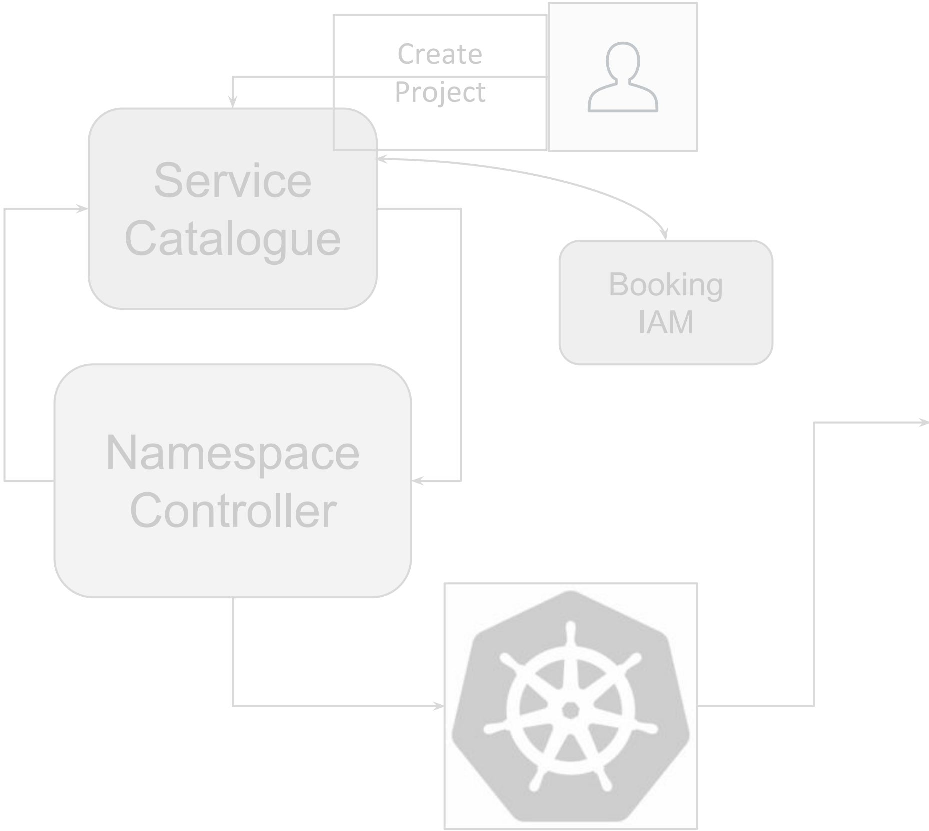
```
{
  "apiVersion":
"authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "spec": {
    "token": "my-bearer-token"
  }
}
```

Example response:

```
Allow:
{
  "apiVersion": "authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "status": {
    "authenticated": true,
    "user": {
      "username": "sanjary",
      "uid": "1234",
      "groups": [ "bikerental:admin" ]
    }
  }
}

Deny:
{
  "apiVersion": "authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "status": {
    "authenticated": false
  }
}
```

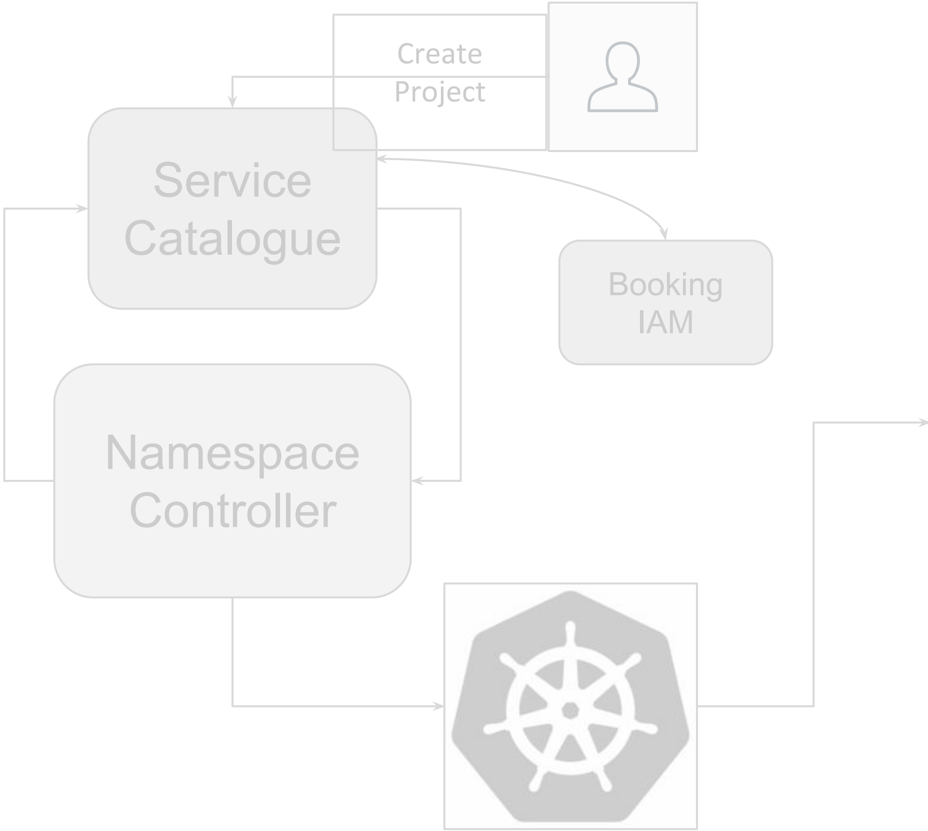
Auth Webhook + Namespace Controller



Rolebinding:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: admin
  namespace: bikerental
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: admin
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: bikerental:admin
```

Auth Webhook + Namespace Controller



Create Project

Service Catalogue

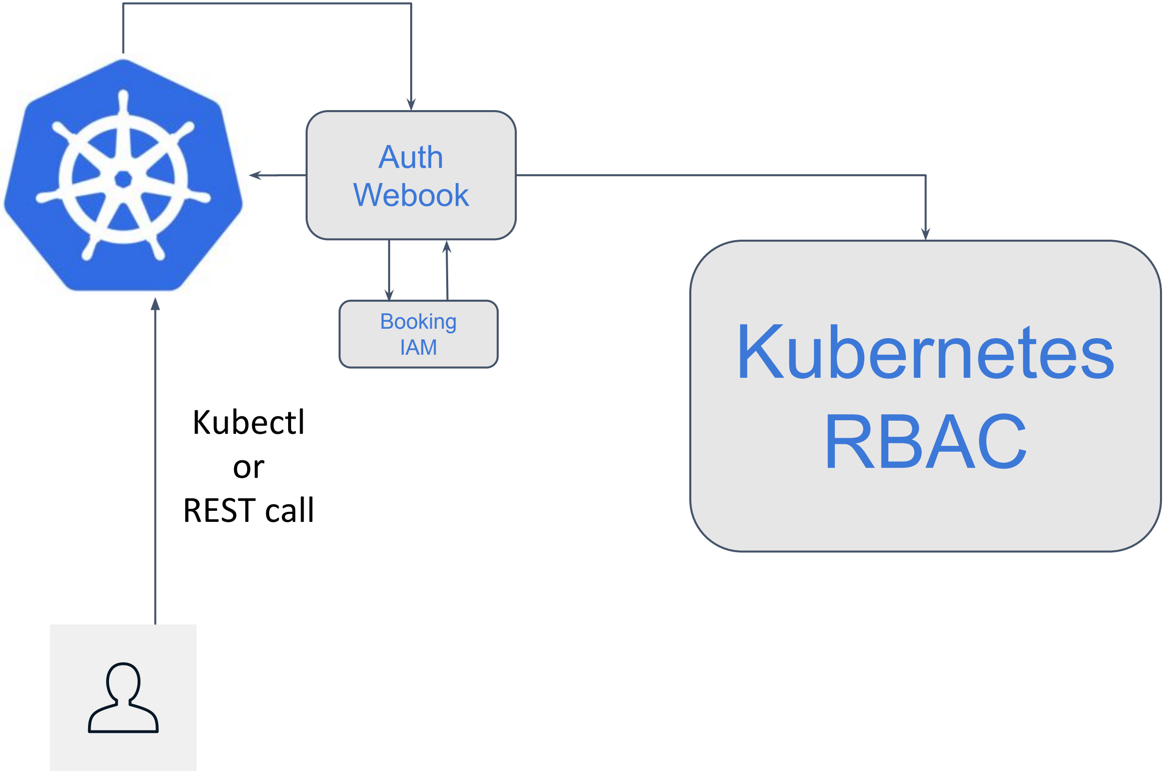
Booking IAM

Namespace Controller

Rolebinding:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: admin
  namespace: bikerental
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: admin
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: bikerental:admin
```

Lifecycle of a kubectl command



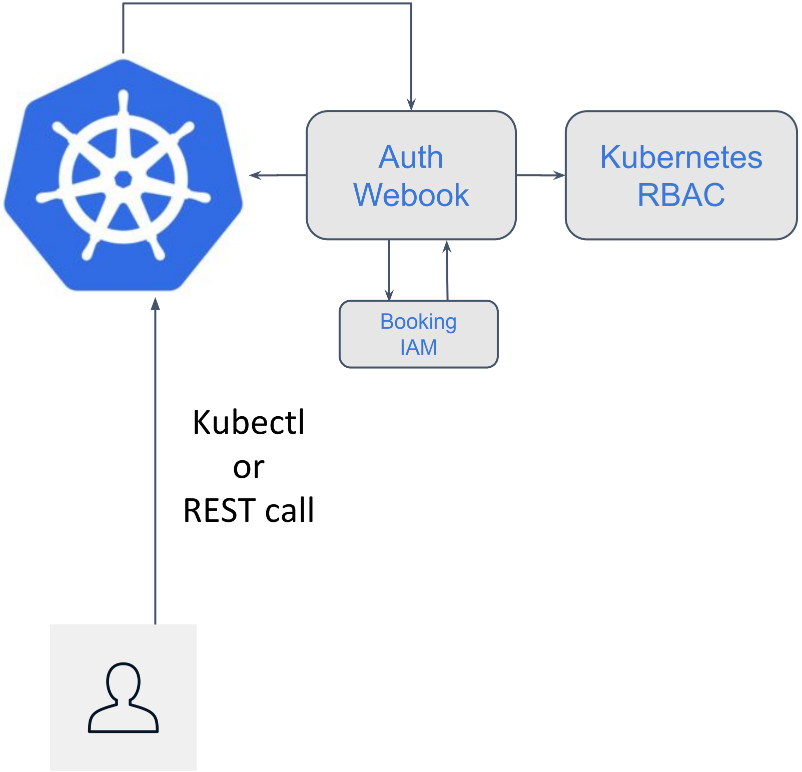
Auth Webook

Booking IAM

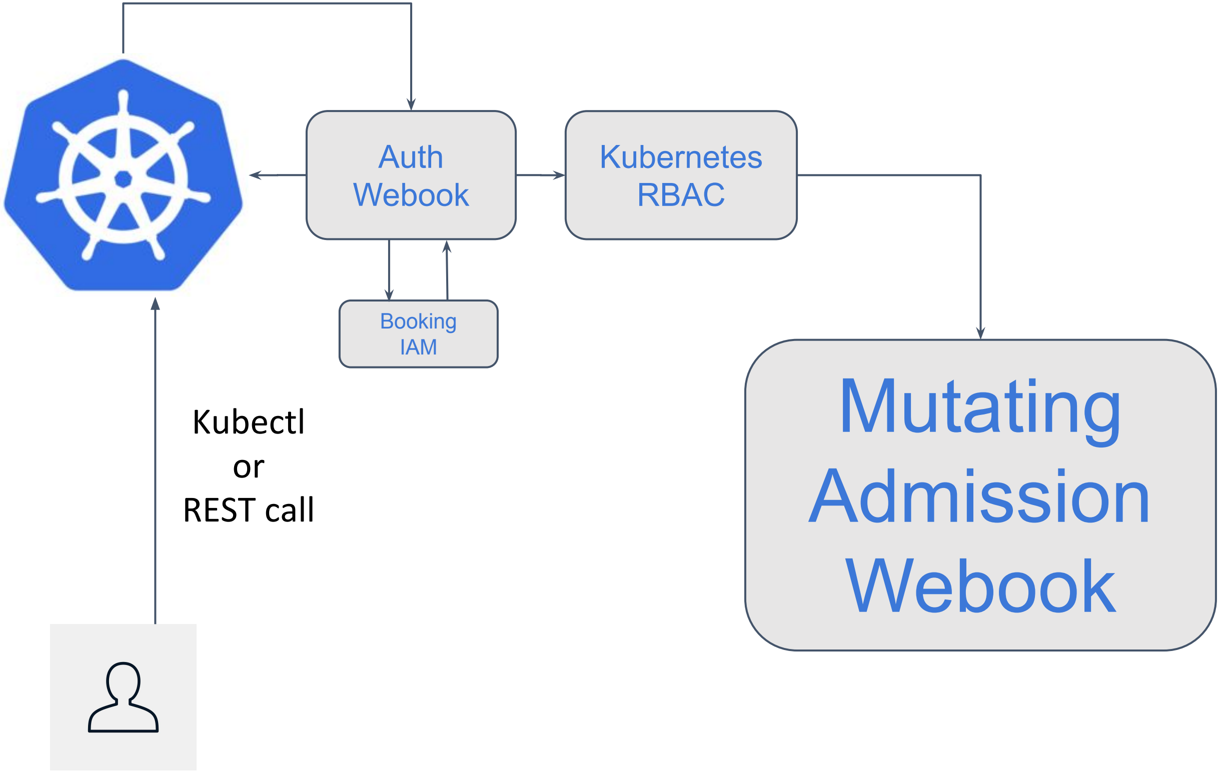
Kubernetes RBAC

Kubectl or REST call

Lifecycle of a kubectl command



Lifecycle of a kubectl command



Admission Webhook (Mutation)

Example use cases:

Admission Webhook (Mutation)

Example use cases:

- Assign random non-root UID to pod

Admission Webhook (Mutation)

Example use cases:

- Assign random non-root UID to pod
- Inject environment variables in pod

Admission Webhook (Mutation)

Example use cases:

- Assign random non-root UID to pod
- Inject environment variables in pod
- Inject labels on pod

Admission Webhook (Mutation)

Example use cases:

- Assign random non-root UID to pod
- Inject environment variables in pod
- Inject labels on pod
- Inject init-containers/sidecars in pod

Admission Webhook (Mutation)

Example request:

```
{
        "uid": "695e2fed-6e50-4ca1-9042-aadbacfb8fb0",
         "resource": {group: "", version: "v1", resource: "pods"},
        "name": "busybox",
        "namespace": "bikerental",
        "operation": "CREATE",
        "userInfo": {
                "username": "sanjary",
                "uid": "1234",
                "groups": "bikerental:admin"
        }
        "object": ".....",
        "oldObject": "..."
}
```

Admission Webhook (Mutation)

Example request:

```
{
      "uid": "695e2fed-6e50-4ca1-9042-aadbacfb8fb0",
       "resource": {group: "", version: "v1", resource: "pods"},
      "name": "busybox",
      "namespace": "bikerental",
      "operation": "CREATE",
      "userInfo": {
            "username": "sanjary",
            "uid": "1234",
            "groups": "bikerental:admin"
      }
      "object": ".....",
      "oldObject": "..."
}
```

Admission Webhook (Mutation)

KubeCon CloudNativeCon Europe 2019

Example request:

```
{
	"uid": "695e2fed-6e50-4ca1-9042-aadbacfb8fb0",
	 "resource": {group: "", version: "v1", resource: "pods"},
	"name": "busybox",
	"namespace": "bikerental",
	"operation": "CREATE",
	"userInfo": {
		"username": "sanjary",
		"uid": "1234",
		"groups": "bikerental:admin"
	}
	"object": ".....",
	"oldObject": "..."
}
```

Admission Webhook (Mutation)

Example request:

```
{
        "uid": "695e2fed-6e50-4ca1-9042-aadbacfb8fb0",
         "resource": {group: "", version: "v1", resource: "pods"},
        "name": "busybox",
        "namespace": "bikerental",
        "operation": "CREATE",
        "userInfo": {
                "username": "sanjary",
                "uid": "1234",
                "groups": "bikerental:admin"
        }
        "object": ".....",
        "oldObject": "..."
}
```

Admission Webhook (Mutation)

Example request:

```
{
      "uid": "695e2fed-6e50-4ca1-9042-aadbacfb8fb0",
       "resource": {group: "", version: "v1", resource: "pods"},
      "name": "busybox",
      "namespace": "bikerental",
      "operation": "CREATE",
      "userInfo": {
            "username": "sanjary",
            "uid": "1234",
            "groups": "bikerental:admin"
      }
      "object": ".....",
      "oldObject": "..."
}
```

Admission Webhook (Mutation)

Example request:

```
{
    "uid": "695e2fed-6e50-4ca1-9042-aadbacfb8fb0",
     "resource": {group: "", version: "v1", resource: "pods"},
    "name": "busybox",
    "namespace": "bikerental",
    "operation": "CREATE",
    "userInfo": {
        "username": "sanjary",
        "uid": "1234",
        "groups": "bikerental:admin"
    }
    "object": ".....",
    "oldObject": "..."
}
```

Admission Webhook (Mutation)

Example request:

```
{
     "uid": "695e2fed-6e50-4ca1-9042-aadbacfb8fb0",
     "resource": {group: "", version: "v1", resource: "pods"},
     "name": "busybox",
     "namespace": "bikerental",
     "operation": "CREATE",
     "userInfo": {
          "username": "sanjary",
          "uid": "1234",
          "groups": "bikerental:admin"
     }
     "object": ".....",
     "oldObject": "..."
}
```

Admission Webhook (Mutation)

Example request:

```
{
        "uid": "695e2fed-6e50-4ca1-9042-aadbacfb8fb0",
         "resource": {group: "", version: "v1", resource: "pods"},
        "name": "busybox",
        "namespace": "bikerental",
        "operation": "CREATE",
        "userInfo": {
                "username": "sanjary",
                "uid": "1234",
                "groups": "bikerental:admin"
        }
        "object": ".....",
        "oldObject": "..." // (only for UPDATE operation)
}
```

Admission Webhook (Mutation)

Example request:

```
{
      "uid": "695e2fed-6e50-4ca1-9042-aadbacfb8fb0",
       "resource": {group: "", version: "v1", resource: "pods"},
      "name": "busybox",
      "namespace": "bikerental",
      "operation": "CREATE",
      "userInfo": {
            "username": "sanjary",
            "uid": "1234",
            "groups": "bikerental:admin"
      }
      "object": ".....",
      "oldObject": "..." // (only for UPDATE operation)
}
```

Admission Webhook (Mutation)

Example response:

```
{
  "response": {
    "uid": "695e2fed-6e50-4ca1-9042-aadbacfb8fb0",
    "allowed":true,
    "patch":"asdfbASCiJhZGQiLCJwYXRoIjoiL21ldGFkYXbSJ9fV0=",
    "patchType":"JSONPatch"
  }
}
```

Admission Webhook (Mutation)

Example response:

```
{
   "response": {
      "uid": "695e2fed-6e50-4ca1-9042-aadbacfb8fb0",
      "allowed":true,
      "patch":"asdfbASCiJhZGQiLCJwYXRoIjoiL21ldGFkYXbSJ9fV0=",
      "patchType":"JSONPatch"
   }
}
```

Admission Webhook (Mutation)

Example response:

```
{
  "response": {
    "uid": "695e2fed-6e50-4ca1-9042-aadbacfb8fb0",
    "allowed":true,
    "patch":"asdfbASCiJhZGQiLCJwYXRoIjoiL21ldGFkYXbSJ9fV0=",
    "patchType":"JSONPatch"
  }
}
```

Admission Webhook (Mutation)

Example response:

```
{
  "response": {
    "uid": "695e2fed-6e50-4ca1-9042-aadbacfb8fb0",
    "allowed":true,
    "patch":"asdfbASCiJhZGQiLCJwYXRoIjoiL21ldGFkYXbSJ9fV0=",
    "patchType":"JSONPatch"
  }
}
```

Patches:

```
[ {  "op": "add",   "path": "/spec/containers/0/imagePullPolicy", "value": "Always" } ]
```

Admission Webhook (Mutation)

Example response:

```
{
  "response": {
    "uid": "695e2fed-6e50-4ca1-9042-aadbacfb8fb0",
    "allowed":true,
    "patch":"asdfbASCiJhZGQiLCJwYXRoIjoiL21ldGFkYXbSJ9fV0=",
    "patchType":"JSONPatch"
  }
}
```

pod.yaml (before mutation)

```
apiVersion: v1
kind: Pod
metadata:
  name: busybox
  namespace: bikerental
spec:
  containers:
  - image: busybox
    imagePullPolicy: IfNotPresent
    name: busybox
```

pod.yaml (after mutation)

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: busybox
  namespace: bikerental
spec:
  containers:
  - image: busybox
    imagePullPolicy: Always
    name: busybox
```

Admission Webhook (Mutation)

Configure kube-apiserver
with the webhook flag

Admission Webhook (Mutation)

Configure kube-apiserver with the webhook flag

--enable-admission-plugins= MutatingAdmissionWebhook,...
.

Admission Webhook (Mutation)

Configure kube-apiserver with the webhook flag

--enable-admission-plugins=
MutatingAdmissionWebhook,..
.

```
apiVersion: admissionregistration.k8s.io/v1beta1
kind: MutatingWebhookConfiguration
metadata:
  name: pod-mutation
webhooks:
- clientConfig:
    caBundle: <CA bundle>
    service: null
    Url: https://mutate.example.com/v1/mutate-pods
  failurePolicy: Fail
  name: pod-mutation
  rules:
  - apigroups:
    - '*'
    apiVersions:
    - '*'
    operations:
    - CREATE
    resources:
    - pods
```

Admission Webhook (Mutation)

Configure kube-apiserver with the webhook flag

--enable-admission-plugins=
MutatingAdmissionWebhook,..
.

```
apiVersion: admissionregistration.k8s.io/v1beta1
kind: MutatingWebhookConfiguration
metadata:
  name: pod-mutation
webhooks:
- clientConfig:
    caBundle: <CA bundle>
    service: null
    Url: https://mutate.example.com/v1/mutate-pods
  failurePolicy: Fail
  name: pod-mutation
  rules:
  - apigroups:
    - '*'
    apiVersions:
    - '*'
    operations:
    - CREATE
    resources:
    - pods
```

Admission Webhook (Mutation)

Configure kube-apiserver with the webhook flag

--enable-admission-plugins=
MutatingAdmissionWebhook,..
.

```
apiVersion: admissionregistration.k8s.io/v1beta1
kind: MutatingWebhookConfiguration
metadata:
  name: pod-mutation
webhooks:
- clientConfig:
    caBundle: <CA bundle>
    service: null
    Url: https://mutate.example.com/v1/mutate-pods
  failurePolicy: Fail
  name: pod-mutation
  rules:
  - apigroups:
    - '*'
    apiVersions:
    - '*'
    operations:
    - CREATE
    resources:
    - pods
```

Admission Webhook (Mutation)

Configure kube-apiserver with the webhook flag

--enable-admission-plugins=
MutatingAdmissionWebhook,..
.

```
apiVersion: admissionregistration.k8s.io/v1beta1
kind: MutatingWebhookConfiguration
metadata:
  name: pod-mutation
webhooks:
- clientConfig:
    caBundle: <CA bundle>
    service: null
    url: https://mutate.example.com/v1/mutate-pods
  failurePolicy: Fail
  name: pod-mutation
  rules:
  - apigroups:
    - '*'
    apiVersions:
    - '*'
    operations:
    - CREATE
    resources:
    - pods
```

Admission Webhook (Mutation)

Configure kube-apiserver with the webhook flag

--enable-admission-plugins=
MutatingAdmissionWebhook,..
.

```
apiVersion: admissionregistration.k8s.io/v1beta1
kind: MutatingWebhookConfiguration
metadata:
  name: pod-mutation
webhooks:
- clientConfig:
    caBundle: <CA bundle>
    service: null
    Url: https://mutate.example.com/v1/mutate-pods
  failurePolicy: Fail
  name: pod-mutation
  rules:
  - apigroups:
    - '*'
    apiVersions:
    - '*'
    operations:
    - CREATE
    resources:
    - pods
```

Admission Webhook (Mutation)

Configure kube-apiserver with the webhook flag

--enable-admission-plugins=
MutatingAdmissionWebhook,..
.

```
apiVersion: admissionregistration.k8s.io/v1beta1
kind: MutatingWebhookConfiguration
metadata:
  name: pod-mutation
webhooks:
- clientConfig:
    caBundle: <CA bundle>
    service: null
    Url: https://mutate.example.com/v1/mutate-pods
  failurePolicy: Fail
  name: pod-mutation
  rules:
  - apigroups:
    - '*'
    apiVersions:
    - '*'
    operations:
    - CREATE
    resources:
    - pods
```

Admission Webhook (Mutation)

Configure kube-apiserver with the webhook flag

--enable-admission-plugins=
MutatingAdmissionWebhook,..
.

```
apiVersion: admissionregistration.k8s.io/v1beta1
kind: MutatingWebhookConfiguration
metadata:
  name: pod-mutation
webhooks:
- clientConfig:
    caBundle: <CA bundle>
    service: null
    Url: https://mutate.example.com/v1/mutate-pods
  failurePolicy: Fail
  name: pod-mutation
  rules:
  - apigroups:
    - '*'
    apiVersions:
    - '*'
    operations:
    - CREATE
    resources:
    - pods
```

Admission Webhook (Mutation)

Configure kube-apiserver with the webhook flag

--enable-admission-plugins=
MutatingAdmissionWebhook,..
.

```
apiVersion: admissionregistration.k8s.io/v1beta1
kind: MutatingWebhookConfiguration
metadata:
  name: pod-mutation
webhooks:
- clientConfig:
    caBundle: <CA bundle>
    service: null
    Url: https://mutate.example.com/v1/mutate-pods
  failurePolicy: Fail
  name: pod-mutation
  rules:
  - apigroups:
    - '*'
    apiVersions:
    - '*'
    operations:
    - CREATE
    resources:
    - pods
```

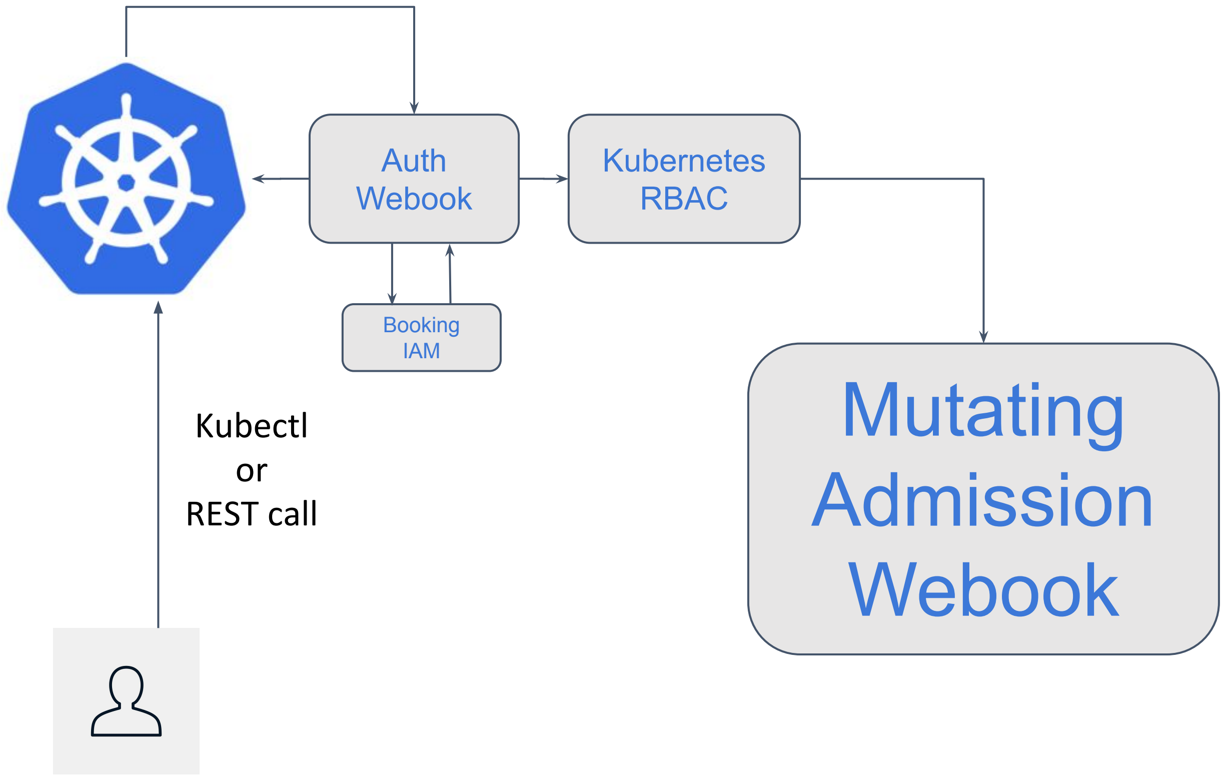
Admission Webhook (Mutation)

Configure kube-apiserver with the webhook flag

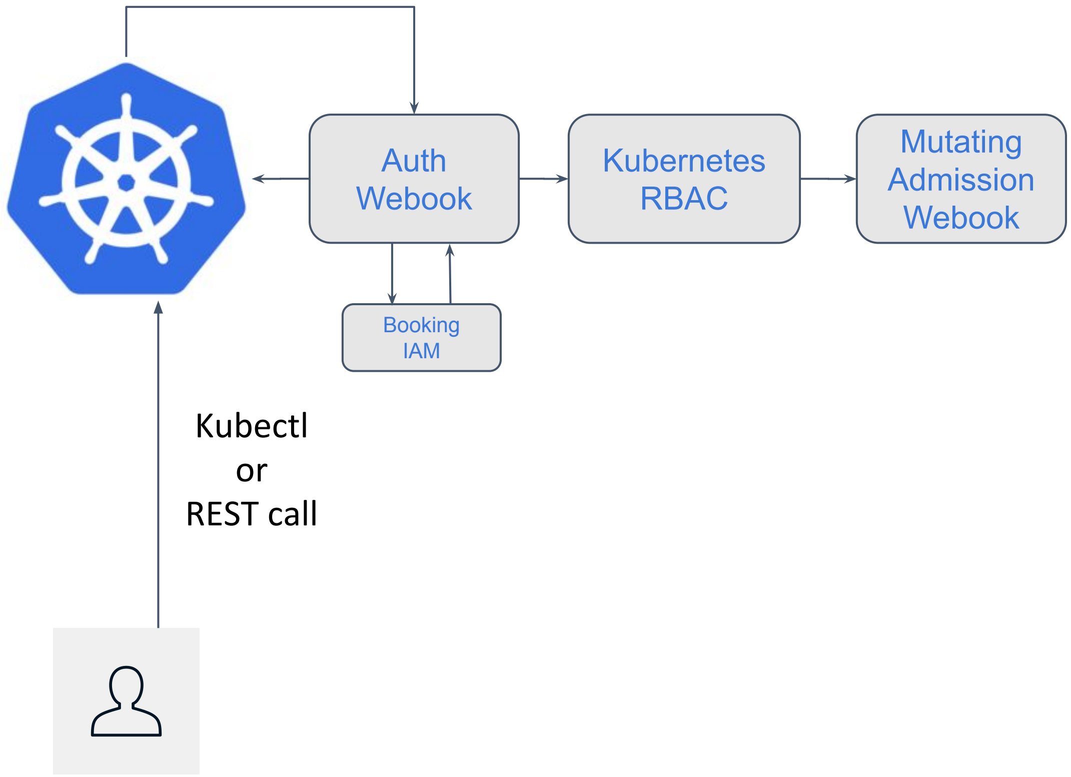
--enable-admission-plugins=
MutatingAdmissionWebhook,..
.

```
apiVersion: admissionregistration.k8s.io/v1beta1
kind: MutatingWebhookConfiguration
metadata:
  name: pod-mutation
webhooks:
- clientConfig:
    caBundle: <CA bundle>
    service: null
    url: https://mutate.example.com/v1/mutate-pods
  failurePolicy: Fail
  name: pod-mutation
  rules:
  - apigroups:
    - '*'
    apiVersions:
    - '*'
    operations:
    - CREATE
    resources:
    - pods
```

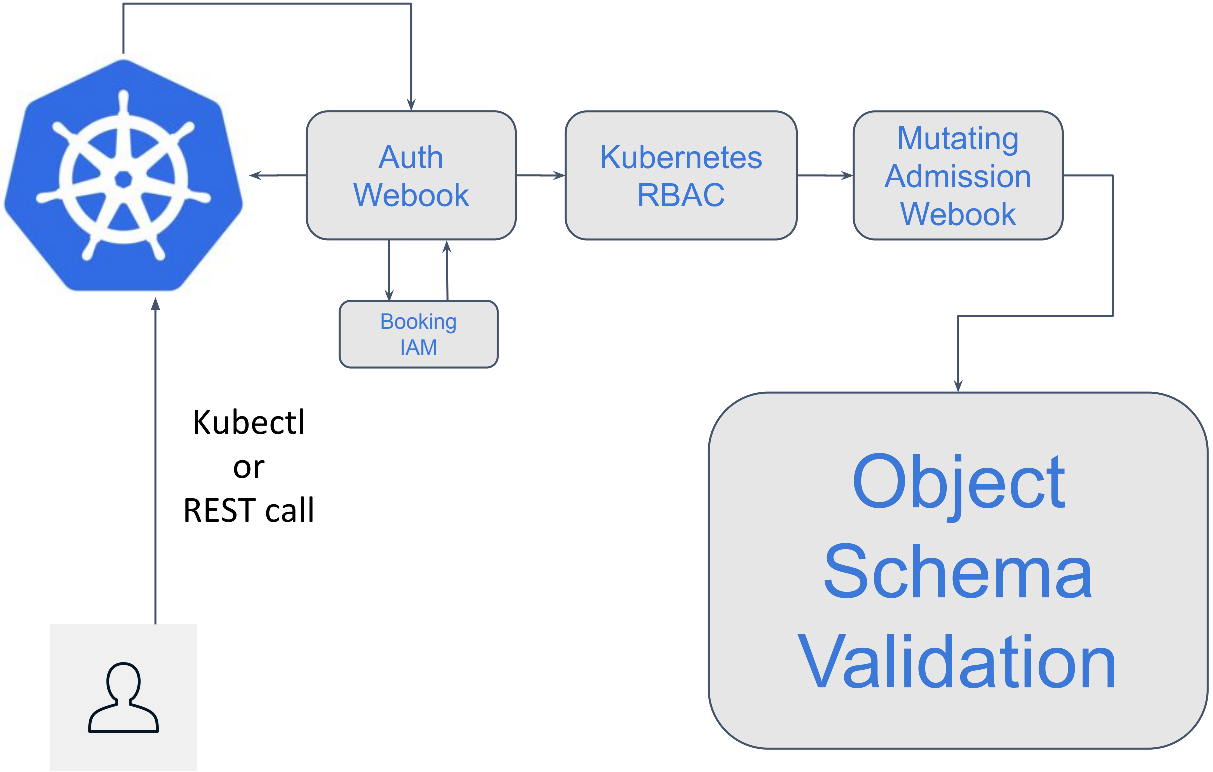
Lifecycle of a kubectl command



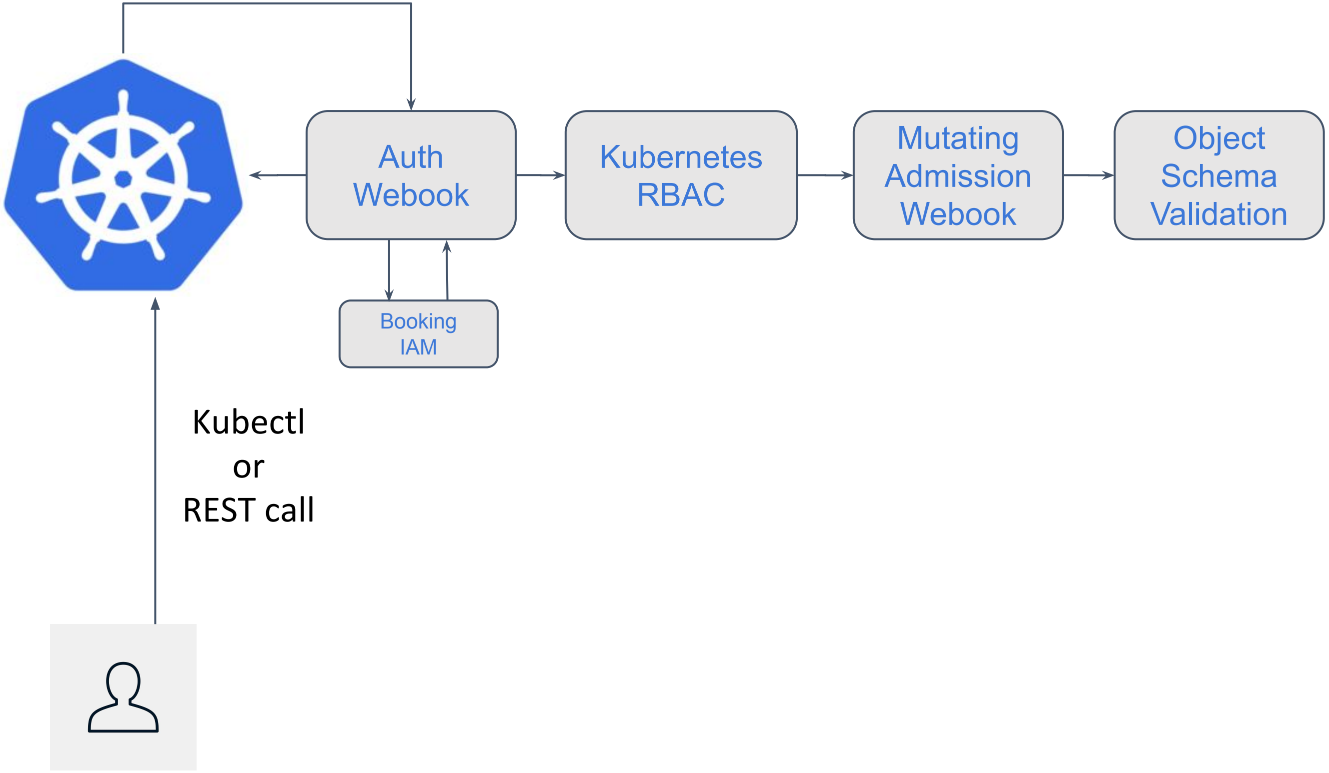
Lifecycle of a kubectl command



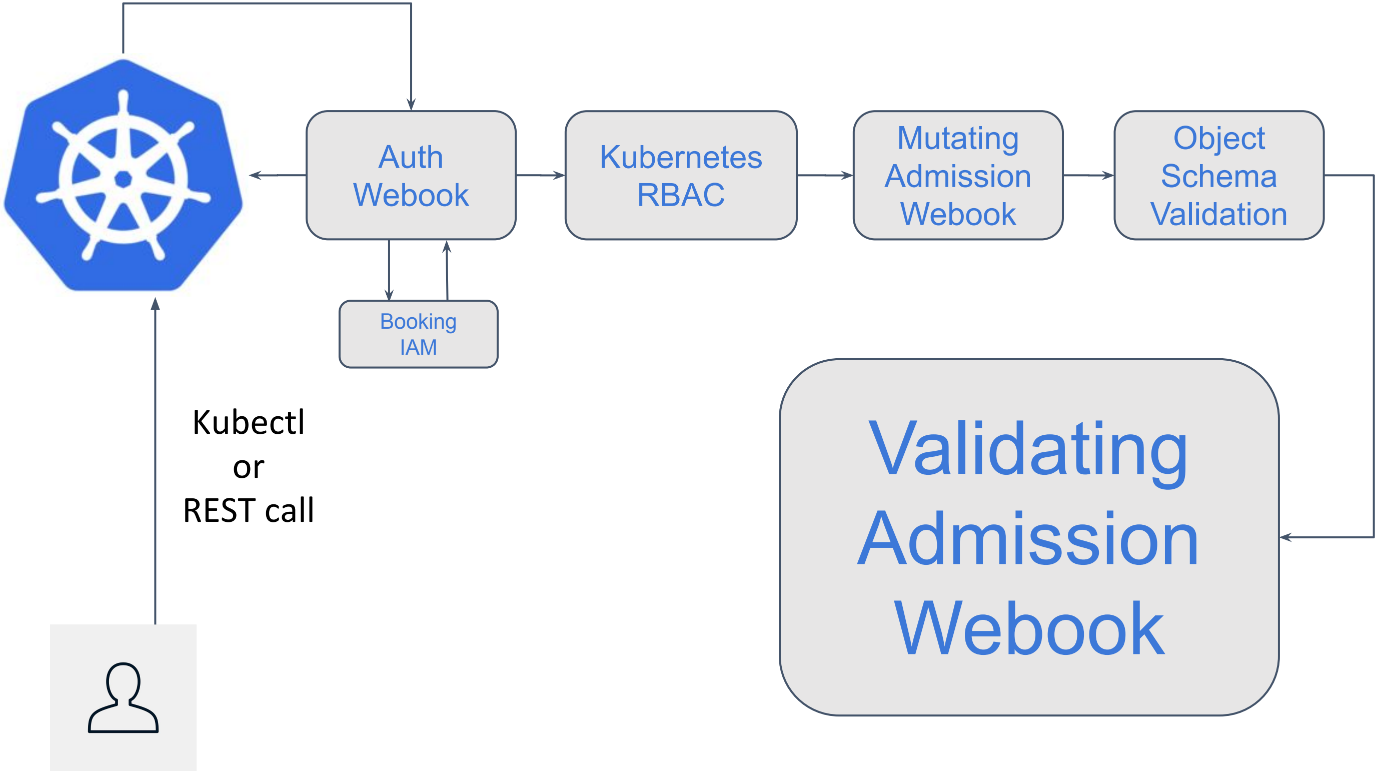
Lifecycle of a kubectl command



Lifecycle of a kubectl command



Lifecycle of a kubectl command



Admission Webhook (Validation)

Example use cases:

Admission Webhook (Validation)

Example use cases:

- Check legitimacy (eg. registration in Service Catalogue in our case)

Admission Webhook (Validation)

Example use cases:

- Check legitimacy (eg. registration in Service Catalogue in our case)

- Ensure running images only from trusted sources

Admission Webhook (Validation)

Example use cases:

- Check legitimacy (eg. registration in Service Catalogue in our case)
- Ensure running images only from trusted sources
- Check number of containers in pod

Admission Webhook (Validation)

Example use cases:

- Check legitimacy (eg. registration in Service Catalogue in our case)
- Ensure running images only from trusted sources
- Check number of containers in pod
- Check presence of certain labels on pod

Admission Webhook (Validation)

Example use cases:

- Check legitimacy (eg. registration in Service Catalogue in our case)
- Ensure running images only from trusted sources
- Check number of containers in pod
- Check presence of certain labels on pod
- Enforce certain best practices for kubernetes resource declaration

Admission Webhook (Validation)

Example request:

Same as mutation webhook

Admission Webhook (Validation)

Example response (Accept):

```
{
  "response": {
    "uid": "695e2fed-6e50-4ca1-9042-aadbacfb8fb0",
    "allowed":true,
  }
}
```

Admission Webhook (Validation)

Example response (Deny):

```
{
  "response": {
    "uid": "695e2fed-6e50-4ca1-9042-aadbacfb8fb0",
    "allowed": false,
    "status": {
      "message": "this resource name is not allowed",
      "code": 400
    }
  }
}
```

Admission Webhook (Validation)

Example response (Deny):

```
{
  "response": {
    "uid": "695e2fed-6e50-4ca1-9042-aadbacfb8fb0",
    "allowed": false,
    "status": {
      "message": "this resource name is not allowed",
      "code": 400
    }
  }
}
```

Admission Webhook (Validation)

Example response (Deny):

```
{
  "response": {
    "uid": "695e2fed-6e50-4ca1-9042-aadbacfb8fb0",
    "allowed": false,
    "status": {
      "message": "this resource name is not allowed",
      "code": 400
    }
  }
}
```

Admission Webhook (Validation)

Configure kube-apiserver with the webhook flag

--enable-admission-plugins=
ValidatingAdmissionWebhook,
...

Admission Webhook (Validation)

Configure kube-apiserver with the webhook flag

```
--enable-admission-plugins=
ValidatingAdmissionWebhook,
...
```

Admission Webhook (Validation)

```
apiVersion: admissionregistration.k8s.io/v1beta1
kind: ValidatingWebhookConfiguration
```

Configure kube-apiserver with the webhook flag

--enable-admission-plugins=
ValidatingAdmissionWebhook,
...

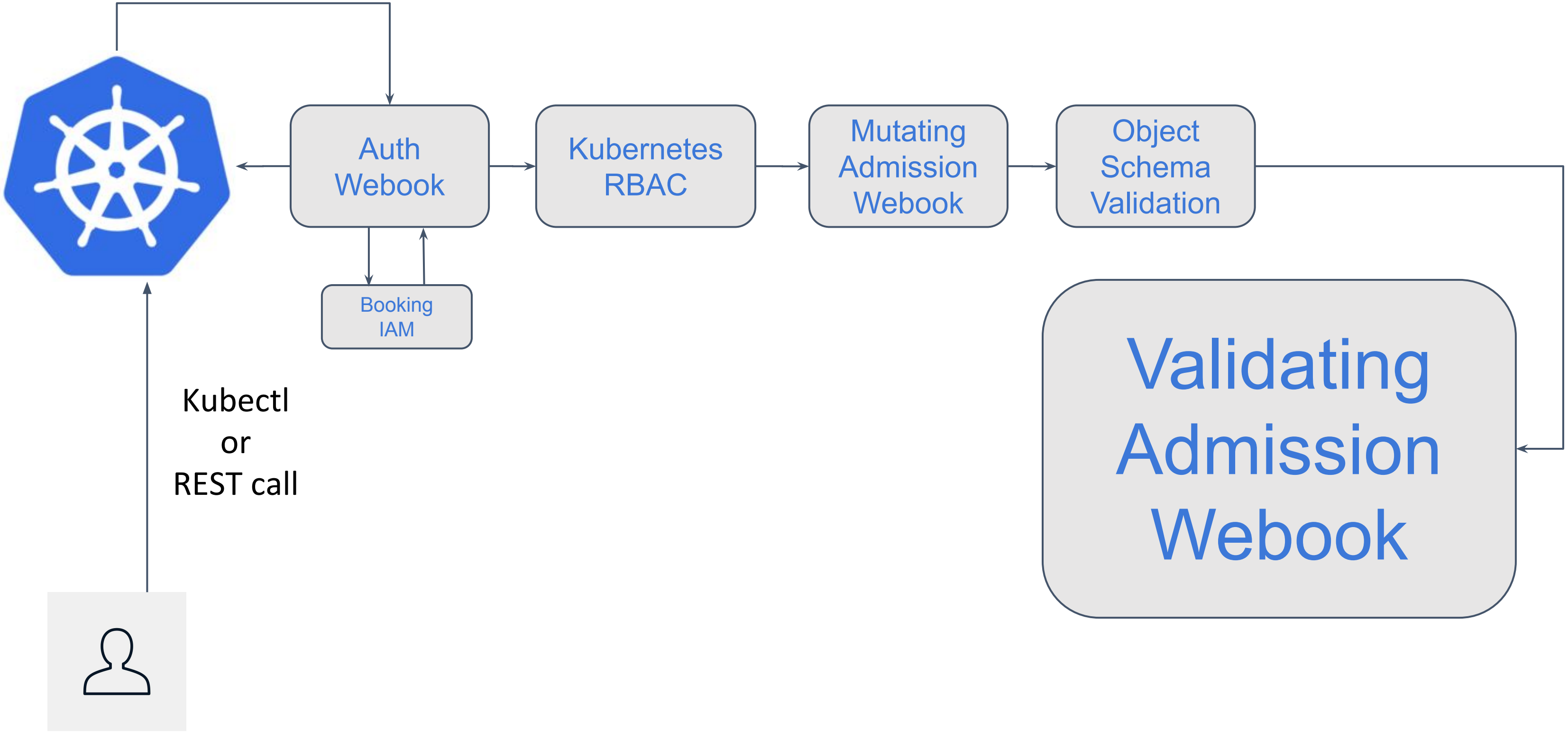
Admission Webhook (Validation)

Configure kube-apiserver with the webhook flag

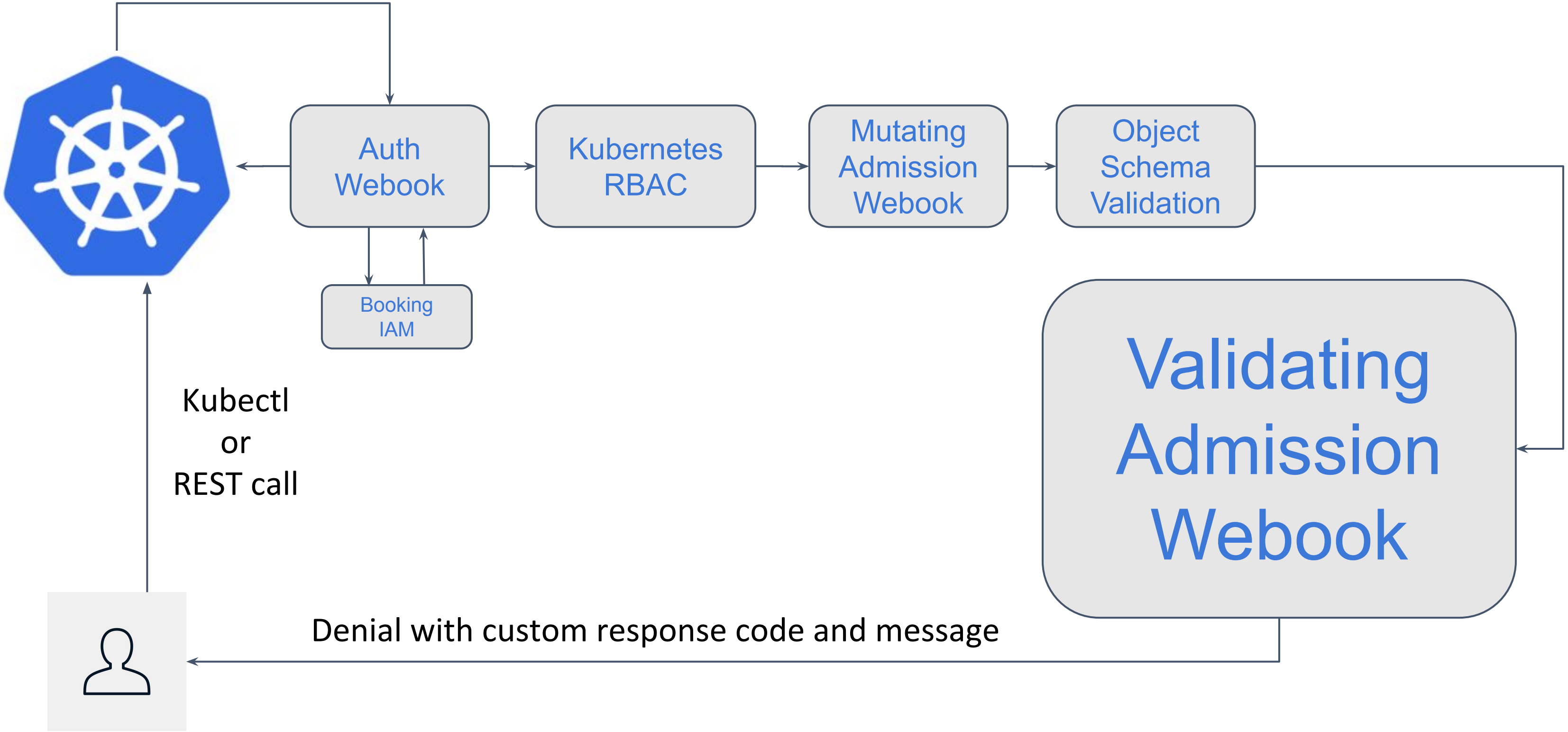
--enable-admission-plugins=
ValidatingAdmissionWebhook,
...

```
apiVersion: admissionregistration.k8s.io/v1beta1
kind: ValidatingWebhookConfiguration
. . . . . . .
. . . . . . .
. . . . . . .
. . . . . . .
Rest is same as Mutating
Webhook Configuration
. . . . . . .
. . . . . . .
. . . . . . .
. . . . . . .
. . . . . . .
```

Lifecycle of a kubectl command



Lifecycle of a kubectl command



Auth Webook

Kubernetes RBAC

Mutating Admission Webook

Object Schema Validation

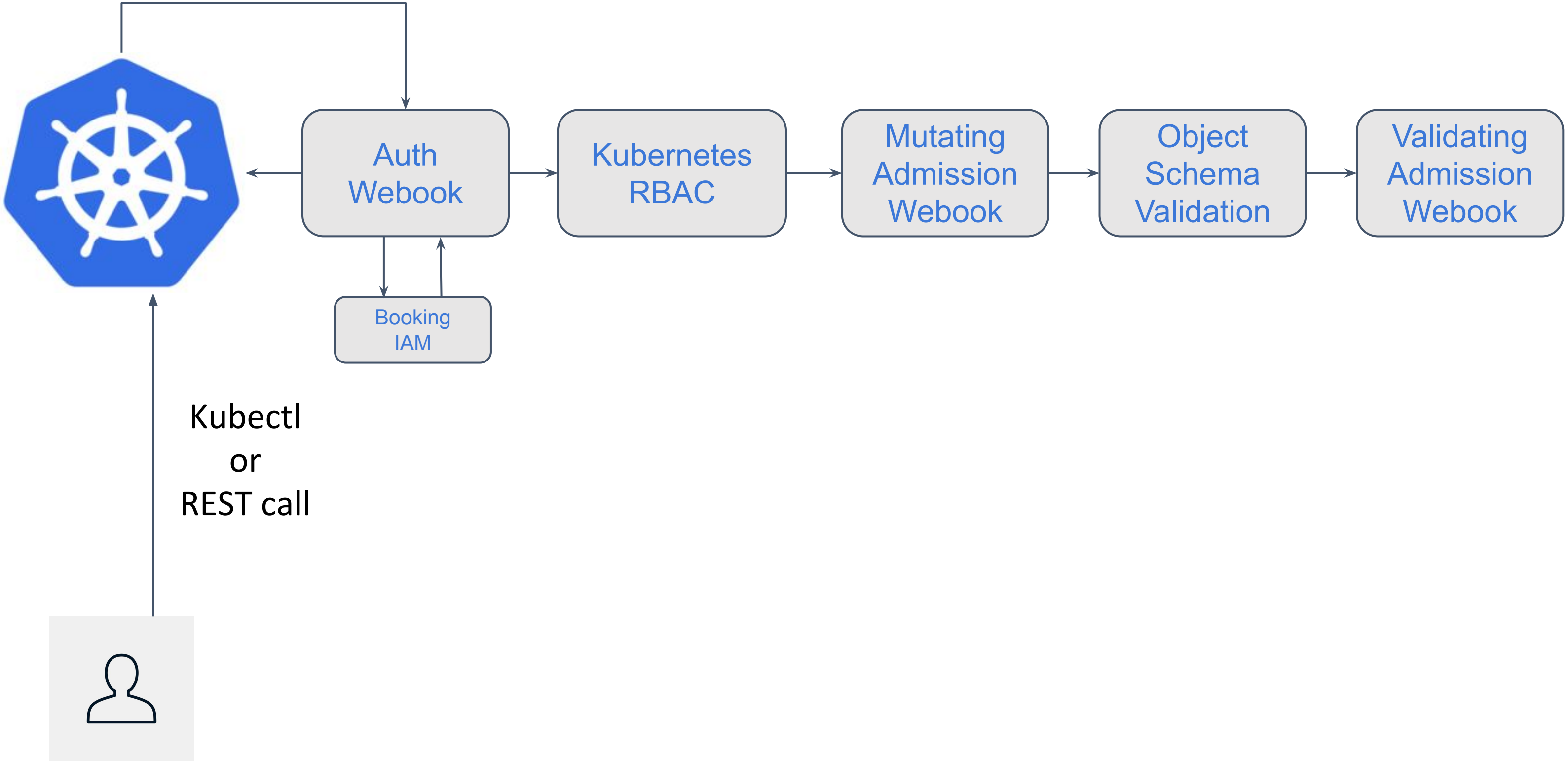
Booking IAM

Kubectl or REST call

Validating Admission Webook

Denial with custom response code and message

Lifecycle of a kubectl command



Auth Webook

Kubernetes RBAC

Mutating Admission Webook

Object Schema Validation

Validating Admission Webook

Booking IAM

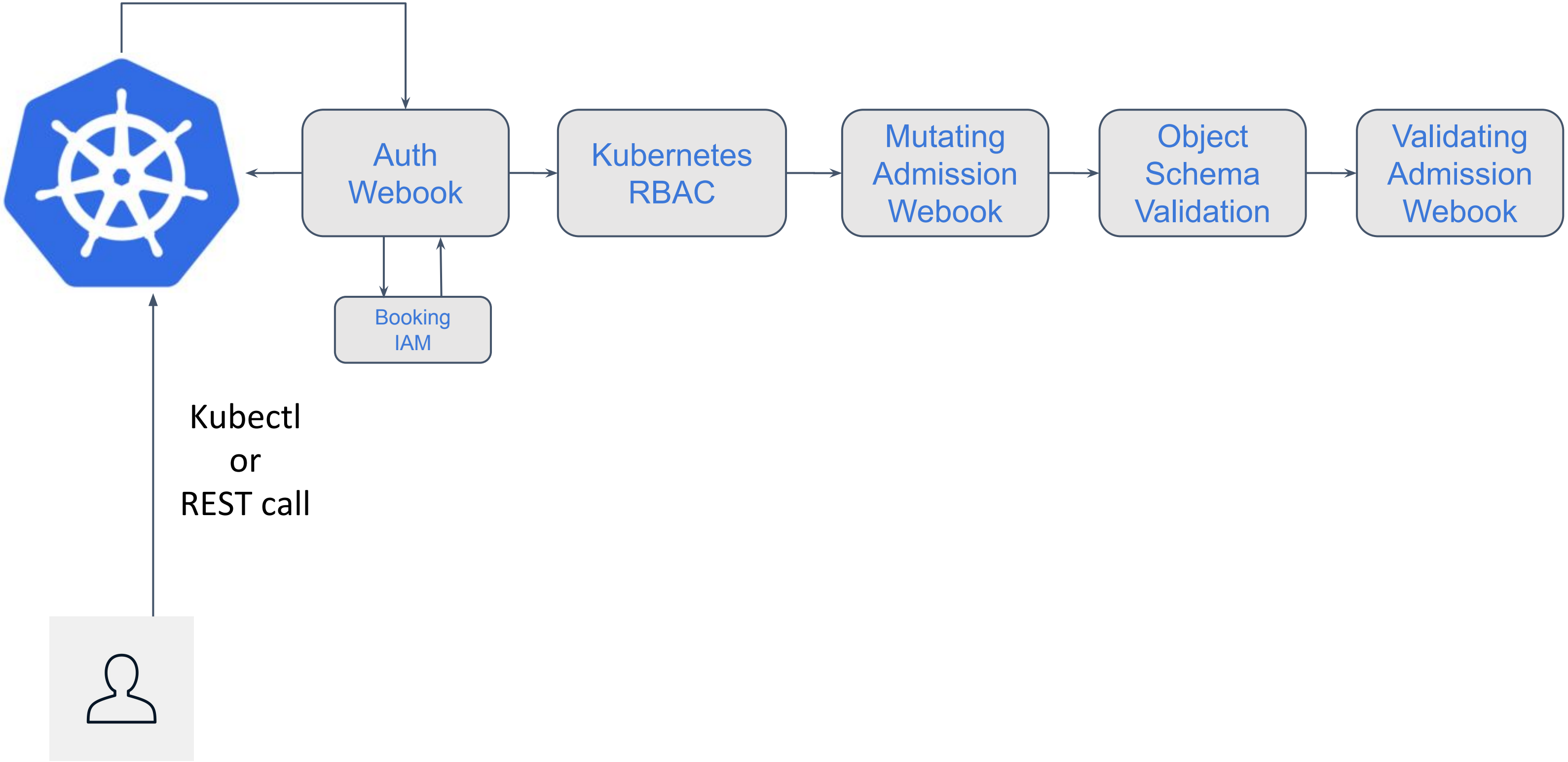
Kubectl or REST call

Admission Webhook

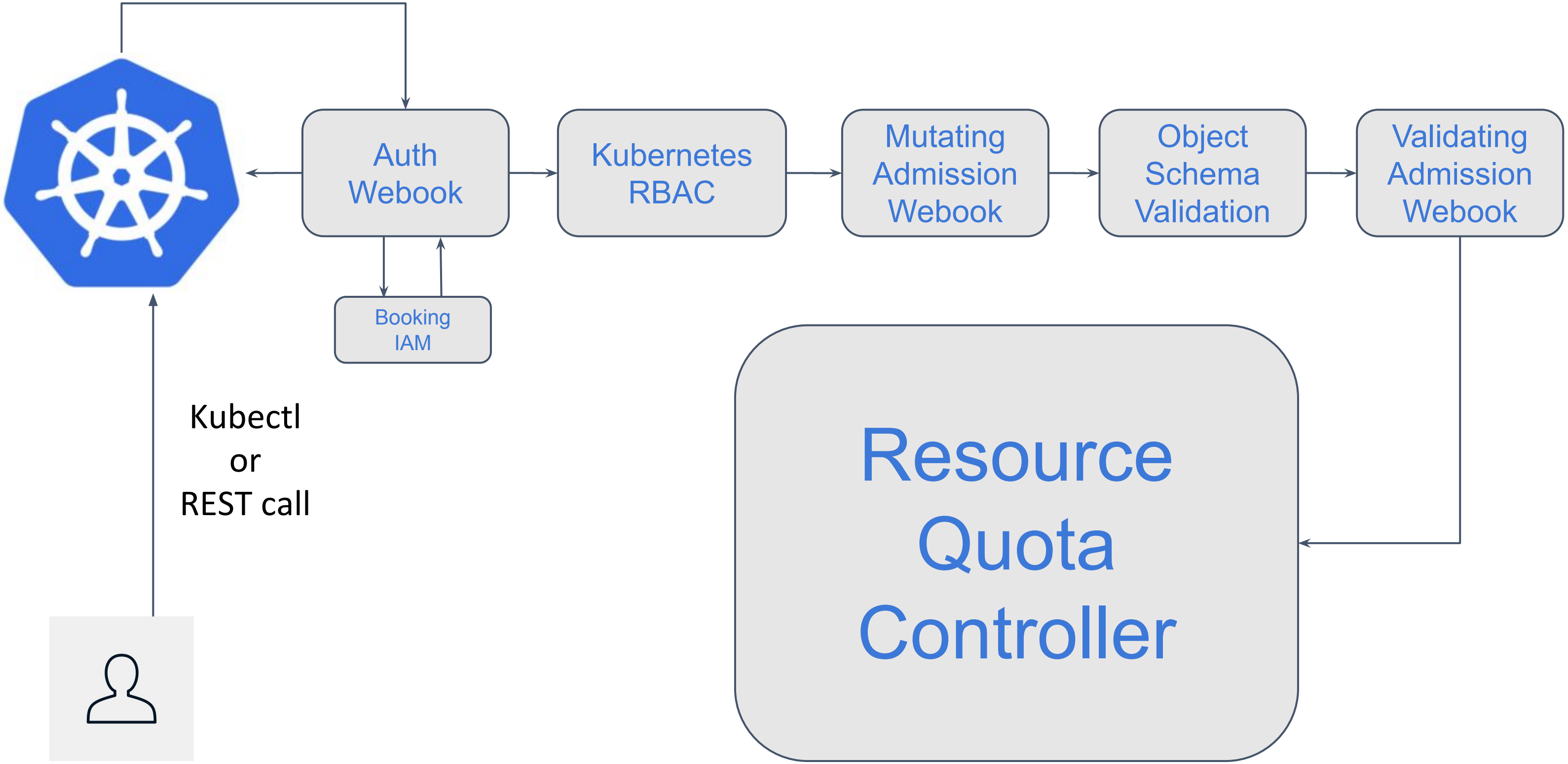
Full example implementation can be found here:

https://github.com/kubernetes/kubernetes/tree/v1.13.0/test/images/webhook

Lifecycle of a kubectl command



Lifecycle of a kubectl command



Kubectl
or
REST call

Auth Webook

Booking IAM

Kubernetes RBAC

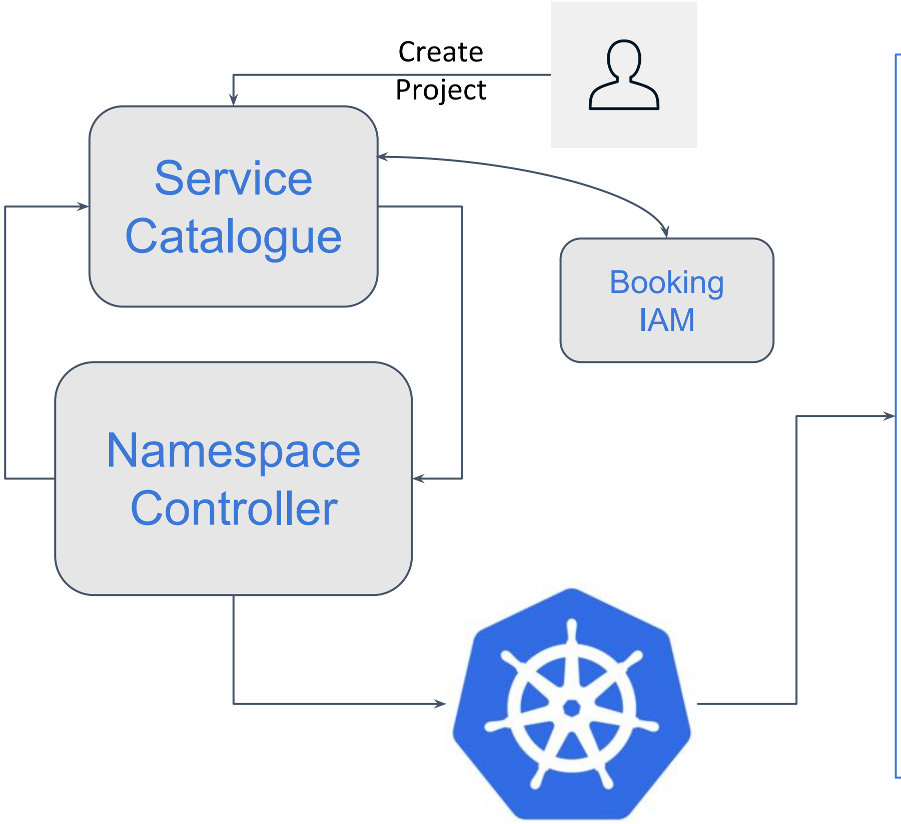
Mutating Admission Webook

Object Schema Validation

Validating Admission Webook

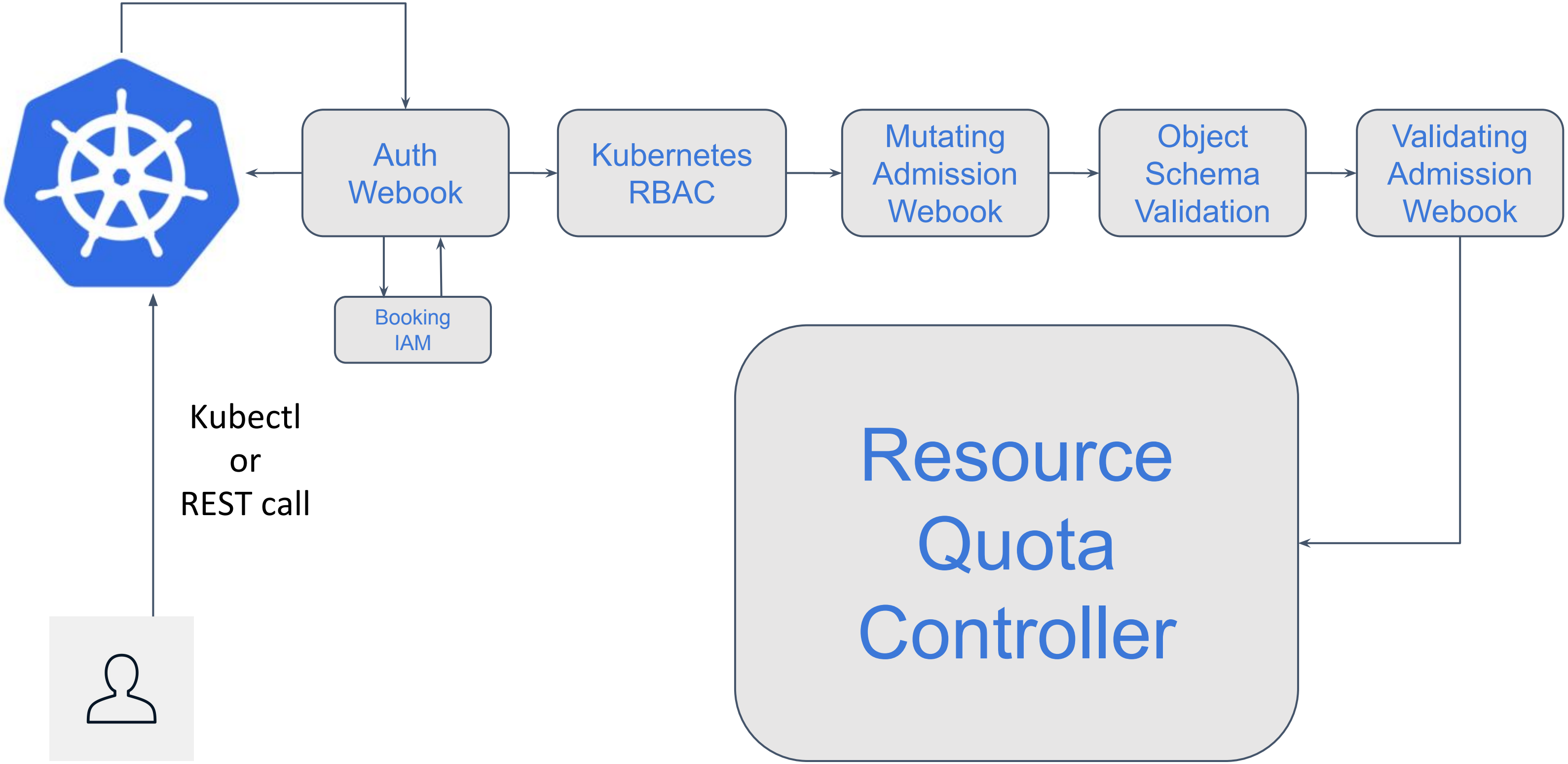
Resource Quota Controller

Workspace Provisioning (Recap)

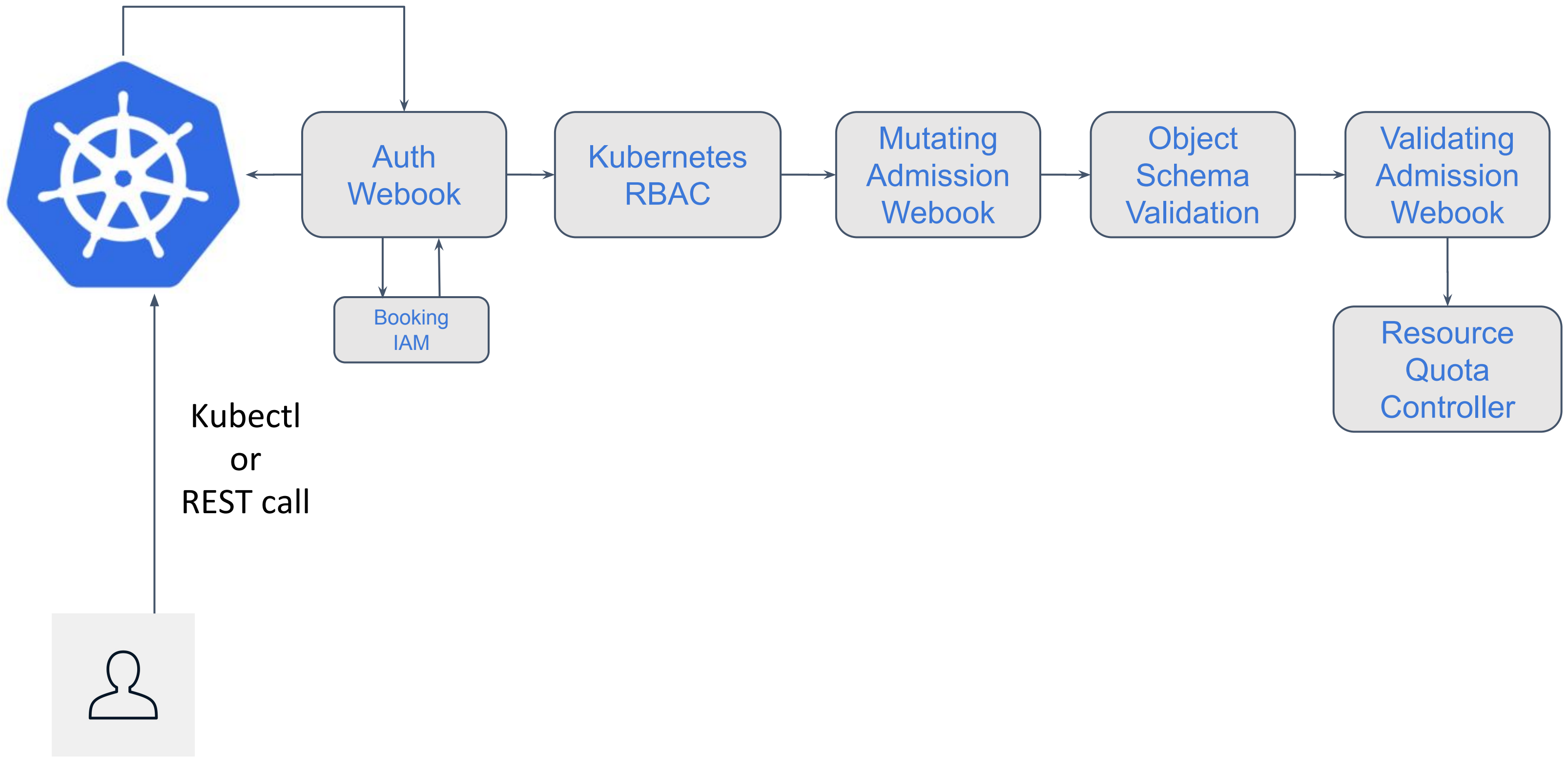


- Creates Namespace
- Creates Rolebinding
- Creates ResourceQuota
- Creates LimitRanges
- Creates Configmaps

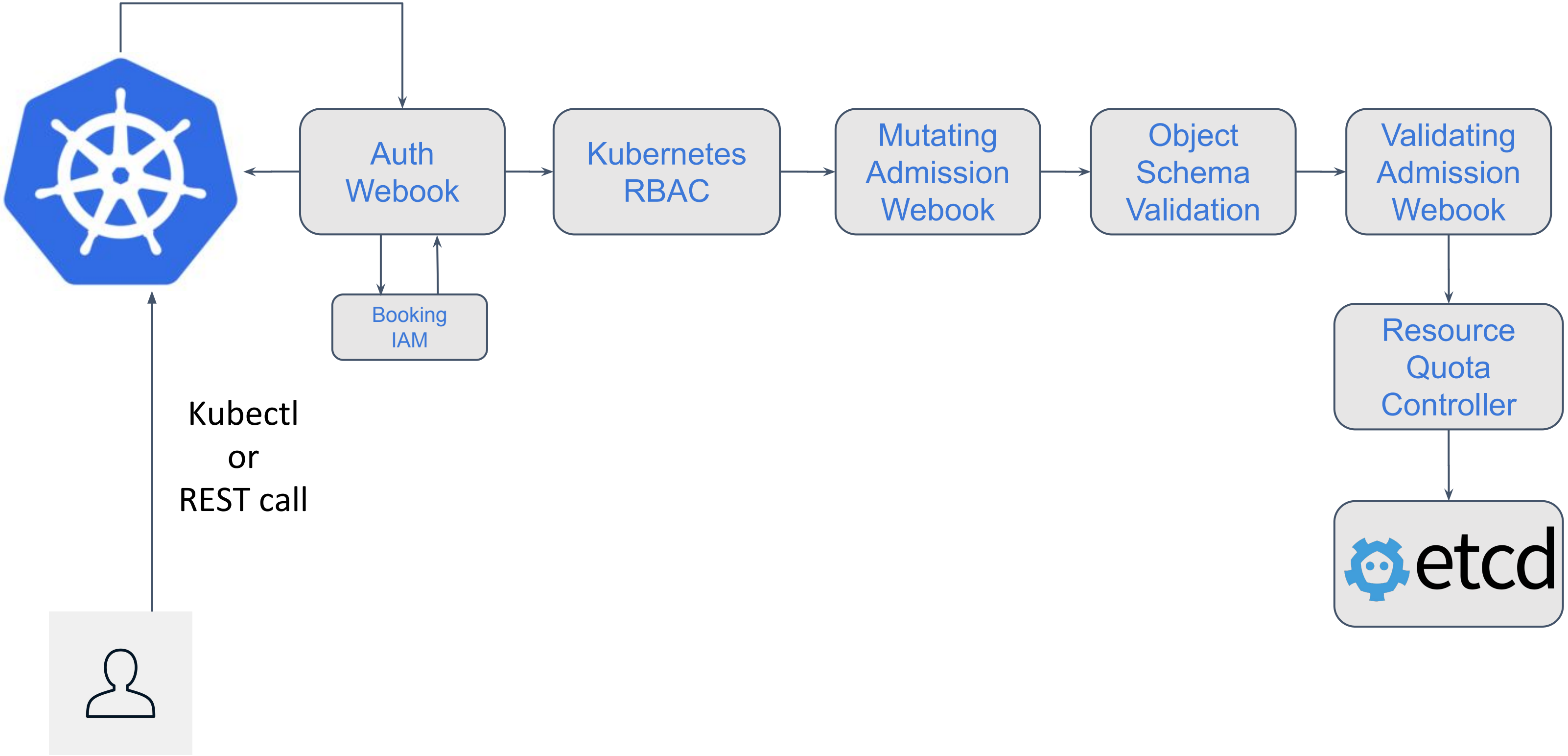
Lifecycle of a kubectl command



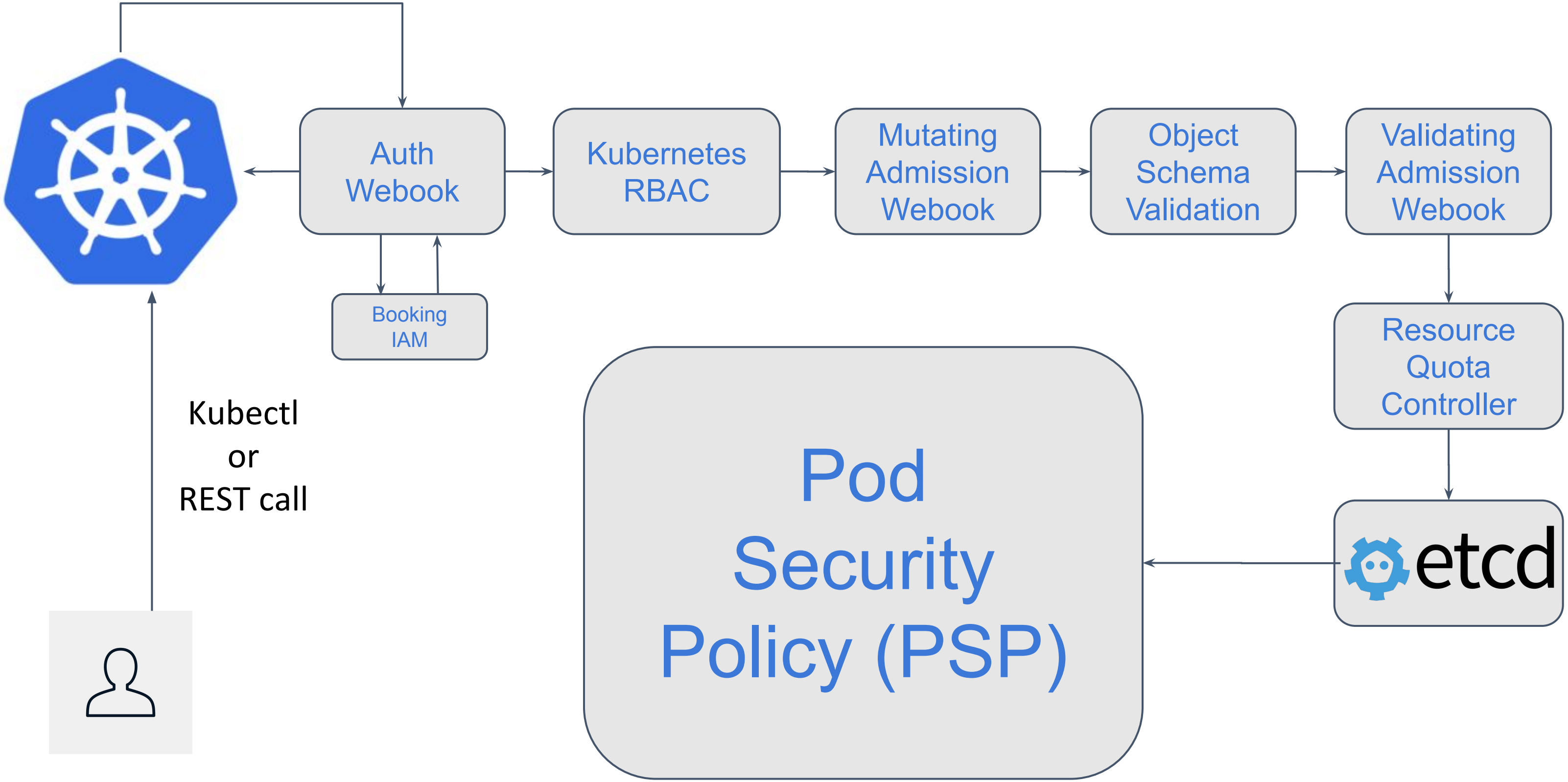
Lifecycle of a kubectl command



Lifecycle of a kubectl command



Lifecycle of a kubectl command



Pod Security Policy (PSP)

Example use cases:

Pod Security Policy (PSP)

Example use cases:

- Prohibit running pod with UID 0 (root)

Pod Security Policy (PSP)

Example use cases:

- Prohibit running pod with UID 0 (root)
- Allow certain or no host path to be mounted from within a pod

Pod Security Policy (PSP)

Example use cases:

- Prohibit running pod with UID 0 (root)
- Allow certain or no host path to be mounted from within a pod
- Do not allow containerized processes to share
  - Host network
  - Host IPC
  - Host Process ID Namespace

Pod Security Policy (PSP)

Example use cases:

- Prohibit running pod with UID 0 (root)
- Allow certain or no host path to be mounted from within a pod
- Do not allow containerized processes to share
  - Host network
  - Host IPC
  - Host Process ID Namespace
- Limit linux capabilities (eg. CAP_NET_ADMIN)

Pod Security Policy (PSP)

Example use cases:

Pod Security Policy (PSP)

Example use cases:

- Allow certain range of UIDs and GIDs to run containers with

Pod Security Policy (PSP)

Example use cases:

- Allow certain range of UIDs and GIDs to run containers with
- Allow certain types of volumes (eg. secret, pvc, configmaps, downward api etc.)

Pod Security Policy (PSP)

Example use cases:

- Allow certain range of UIDs and GIDs to run containers with
- Allow certain types of volumes (eg. secret, pvc, configmaps, downward api etc.)
- Allow only a certain range of host port access in the host network namespace

Pod Security Policy (PSP)

Example use cases:

- Allow certain range of UIDs and GIDs to run containers with
- Allow certain types of volumes (eg. secret, pvc, configmaps, downward api etc.)
- Allow only a certain range of host port access in the host network namespace
- Make root file system inside container read-only

Pod Security Policy (PSP)

- Configure kube-apiserver with the required flag

Pod Security Policy (PSP)

- Configure kube-apiserver with the required flag

--enable-admission-plugins= PodSecurityPolicy,...

Pod Security Policy (PSP)

- Configure kube-apiserver with the required flag

  --enable-admission-plugins= PodSecurityPolicy,...

- Create a common PSP which will be applied by default to any pod

Pod Security Policy (PSP)

- Create restricted PodSecurityPolicy object

```
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
  name: restricted
spec:
......
......
......
......
```

Fill with your requirements from example policy linked here

```
......
......
......
......
```

Pod Security Policy (PSP)

Create cluster role and cluster role binding for restricted PSP

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: restricted
rules:
- apigroups:
  - extensions
  resourceNames:
  - restricted
  resources:
  - podsecuritypolicies
  verbs:
  - use
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: restricted
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: restricted
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: system:serviceaccounts
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: system:authenticated
```

Pod Security Policy (PSP)

Create cluster role and cluster role binding for restricted PSP

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: restricted
rules:
- apigroups:
  - extensions
  resourceNames:
  - restricted
  resources:
  - podsecuritypolicies
  verbs:
  - use
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: restricted
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: restricted
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: system:serviceaccounts
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: system:authenticated
```

Pod Security Policy (PSP)

Create cluster role and cluster role binding for restricted PSP

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: restricted
rules:
- apigroups:
  - extensions
  resourceNames:
  - restricted
  resources:
  - podsecuritypolicies
  verbs:
  - use
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: restricted
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: restricted
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: system:serviceaccounts
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: system:authenticated
```

Pod Security Policy (PSP)

Create cluster role and cluster role binding for restricted PSP

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: restricted
rules:
- apigroups:
  - extensions
  resourceNames:
  - restricted
  resources:
  - podsecuritypolicies
  verbs:
  - use
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: restricted
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: restricted
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: system:serviceaccounts
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: system:authenticated
```

Pod Security Policy (PSP)

Create cluster role and cluster role binding for restricted PSP

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: restricted
rules:
- apigroups:
  - extensions
  resourceNames:
  - restricted
  resources:
  - podsecuritypolicies
  verbs:
  - use
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: restricted
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: restricted
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: system:serviceaccounts
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: system:authenticated
```

Pod Security Policy (PSP)

Create cluster role and cluster role binding for restricted PSP

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: restricted
rules:
- apigroups:
  - extensions
  resourceNames:
  - restricted
  resources:
  - podsecuritypolicies
  verbs:
  - use
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: restricted
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: restricted
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: system:serviceaccounts
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: system:authenticated
```

Pod Security Policy (PSP)

Create cluster role and cluster role binding for restricted PSP

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: restricted
rules:
- apigroups:
  - extensions
  resourceNames:
  - restricted
  resources:
  - podsecuritypolicies
  verbs:
  - use
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: restricted
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: restricted
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: system:serviceaccounts
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: system:authenticated
```

Pod Security Policy (PSP)

KubeCon CloudNativeCon Europe 2019

Create cluster role and cluster role binding for restricted PSP

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: restricted
rules:
- apigroups:
  - extensions
  resourceNames:
  - restricted
  resources:
  - podsecuritypolicies
  verbs:
  - use
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: restricted
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: restricted
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: system:serviceaccounts
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: system:authenticated
```

Pod Security Policy (PSP)

Create cluster role and cluster role binding for restricted PSP

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: restricted
rules:
- apigroups:
  - extensions
  resourceNames:
  - restricted
  resources:
  - podsecuritypolicies
  verbs:
  - use
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: restricted
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: restricted
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: system:serviceaccounts
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: system:authenticated
```

Pod Security Policy (PSP)

Create cluster role and cluster role binding for restricted PSP

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: restricted
rules:
- apigroups:
  - extensions
  resourceNames:
  - restricted
  resources:
  - podsecuritypolicies
  verbs:
  - use
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: restricted
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: restricted
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: system:serviceaccounts
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: system:authenticated
```

Pod Security Policy (PSP)

Create cluster role and cluster role binding for restricted PSP

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: restricted
rules:
- apigroups:
  - extensions
  resourceNames:
  - restricted
  resources:
  - podsecuritypolicies
  verbs:
  - use
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: restricted
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: restricted
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: system:serviceaccounts
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: system:authenticated
```

Pod Security Policy (PSP)

Create cluster role and cluster role binding for restricted PSP

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: restricted
rules:
- apigroups:
  - extensions
  resourceNames:
  - restricted
  resources:
  - podsecuritypolicies
  verbs:
  - use
```

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: restricted
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: restricted
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: system:serviceaccounts
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: system:authenticated
```

Pod Security Policy (PSP)

Create cluster role and cluster role binding for restricted PSP

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: restricted
rules:
- apigroups:
  - extensions
  resourceNames:
  - restricted
  resources:
  - podsecuritypolicies
  verbs:
  - use
```

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: restricted
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: restricted
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: system:serviceaccounts
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: system:authenticated
```

Pod Security Policy (PSP)

What about privileged pod is a valid requirement?

Pod Security Policy (PSP)

Create another privileged PSP the same way as restricted but relaxing the security constraints as per requirement

Pod Security Policy (PSP)

Create cluster role and cluster role binding for privileged PSP

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: privileged
rules:
- apigroups:
  - extensions
  resourceNames:
  - privileged
  resources:
  - podsecuritypolicies
  verbs:
  - use
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: privileged
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: privileged
subjects:
- kind: ServiceAccount
  name: default
  namespace: kube-system
```

Pod Security Policy (PSP)

Create cluster role and cluster role binding for privileged PSP

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: privileged
rules:
- apigroups:
  - extensions
  resourceNames:
  - privileged
  resources:
  - podsecuritypolicies
  verbs:
  - use
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: privileged
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: privileged
subjects:
- kind: ServiceAccount
  name: default
  namespace: kube-system
```

Pod Security Policy (PSP)

Create cluster role and cluster role binding for privileged PSP

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: privileged
rules:
- apigroups:
  - extensions
  resourceNames:
  - privileged
  resources:
  - podsecuritypolicies
  verbs:
  - use
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: privileged
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: privileged
subjects:
- kind: ServiceAccount
  name: default
  namespace: kube-system
```

Pod Security Policy (PSP)

Create cluster role and cluster role binding for privileged PSP

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: privileged
rules:
- apigroups:
  - extensions
  resourceNames:
  - privileged
  resources:
  - podsecuritypolicies
  verbs:
  - use
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: privileged
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: privileged
subjects:
- kind: ServiceAccount
  name: default
  namespace: kube-system
```

Pod Security Policy (PSP)

Create cluster role and cluster role binding for privileged PSP

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: privileged
rules:
- apigroups:
  - extensions
  resourceNames:
  - privileged
  resources:
  - podsecuritypolicies
  verbs:
  - use
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: privileged
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: privileged
subjects:
- kind: ServiceAccount
  name: default
  namespace: kube-system
```

Pod Security Policy (PSP)

Create cluster role and cluster role binding for privileged PSP

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: privileged
rules:
- apigroups:
  - extensions
  resourceNames:
  - privileged
  resources:
  - podsecuritypolicies
  verbs:
  - use
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: privileged
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: privileged
subjects:
- kind: ServiceAccount
  name: default
  namespace: kube-system
```

Pod Security Policy (Ordering)

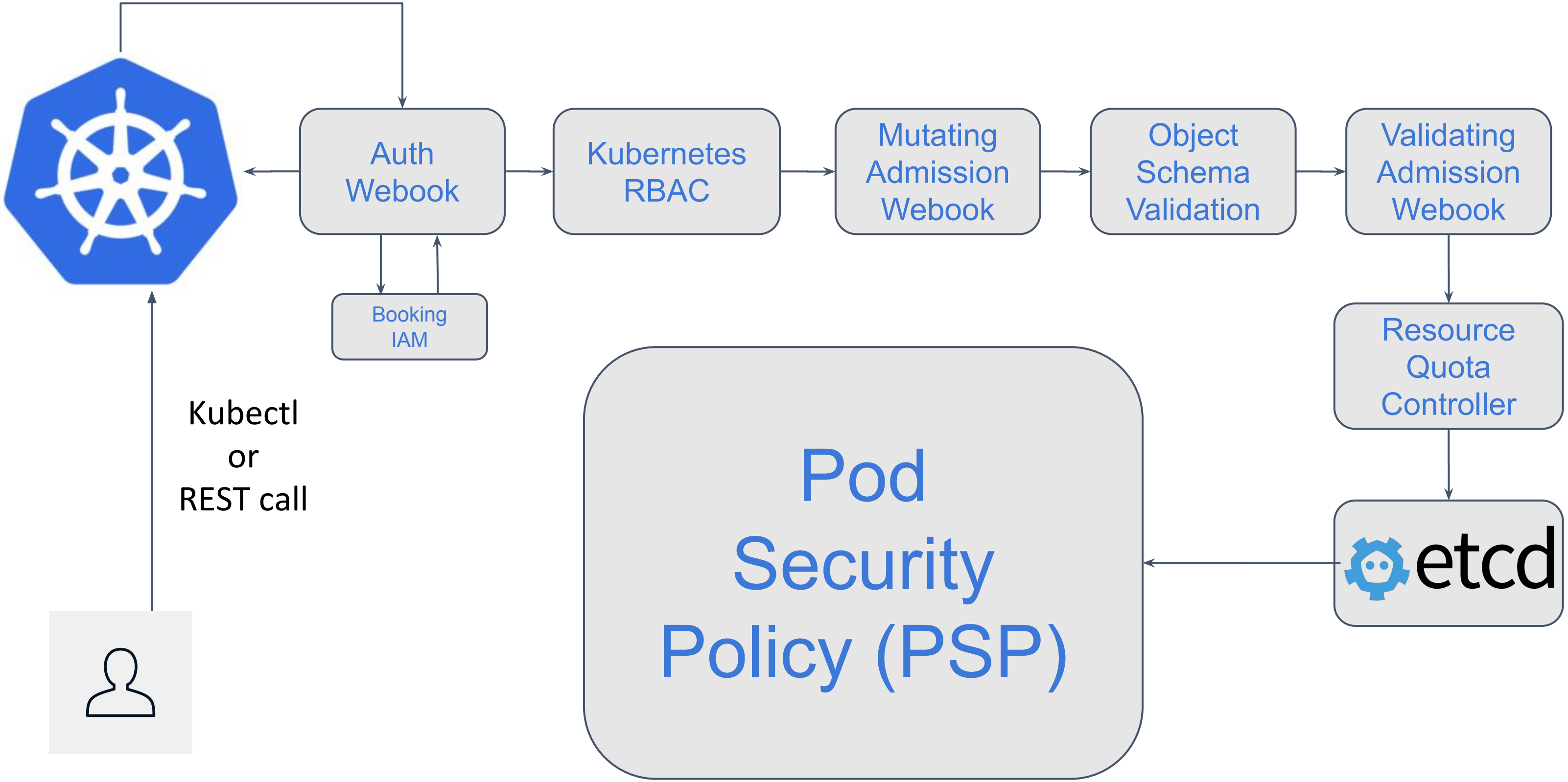
What if multiple policies are applicable to a pod creation request?

Pod Security Policy (Ordering)

What if multiple policies are applicable to a pod creation request?

- The first valid policy in alphabetical order is used

Lifecycle of a kubectl command



Auth Webook

Kubernetes RBAC

Mutating Admission Webook

Object Schema Validation

Validating Admission Webook

Booking IAM

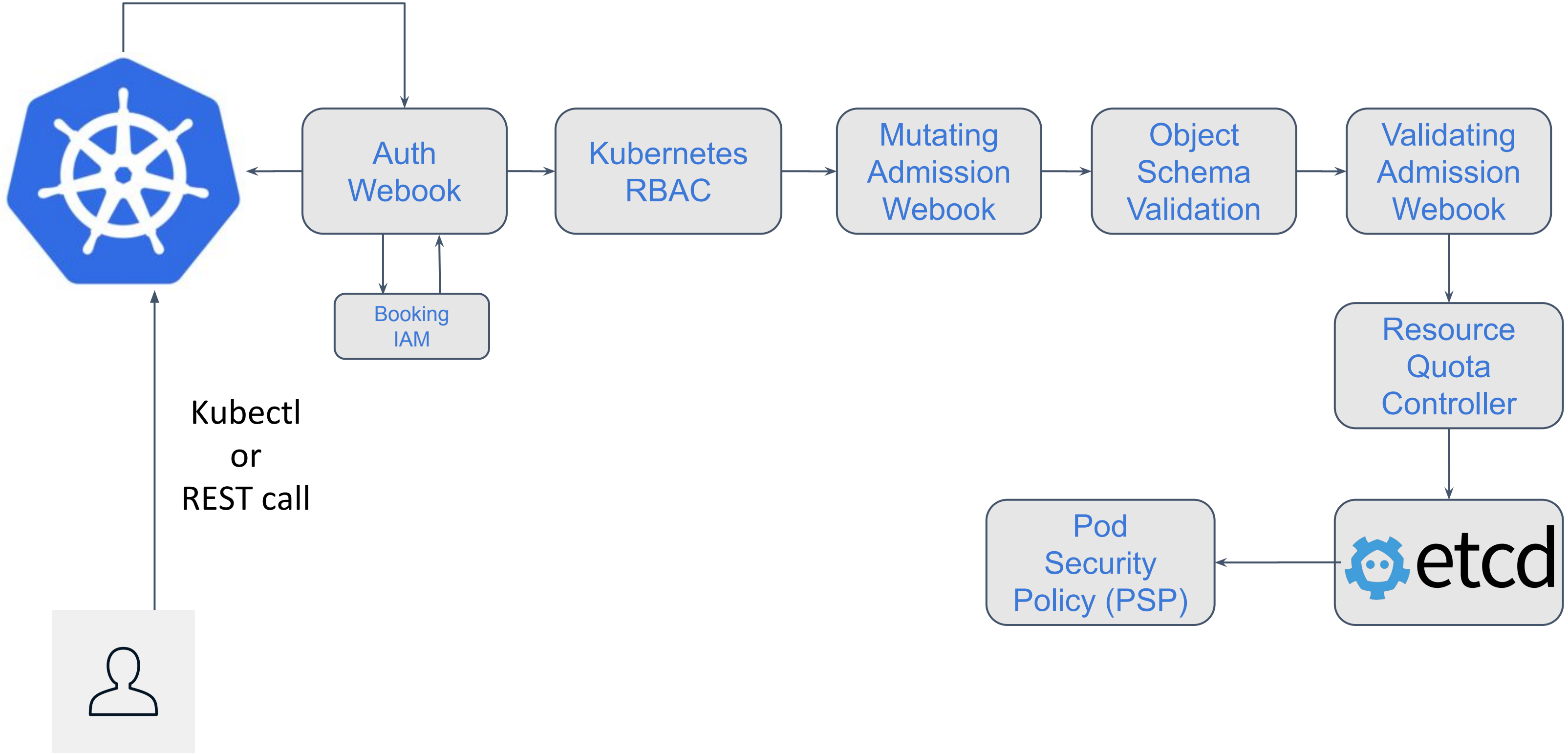
Resource Quota Controller

etcd

Kubectl or REST call

Pod Security Policy (PSP)

Lifecycle of a kubectl command



Auth Webook

Booking IAM

Kubernetes RBAC

Mutating Admission Webook

Object Schema Validation

Validating Admission Webook

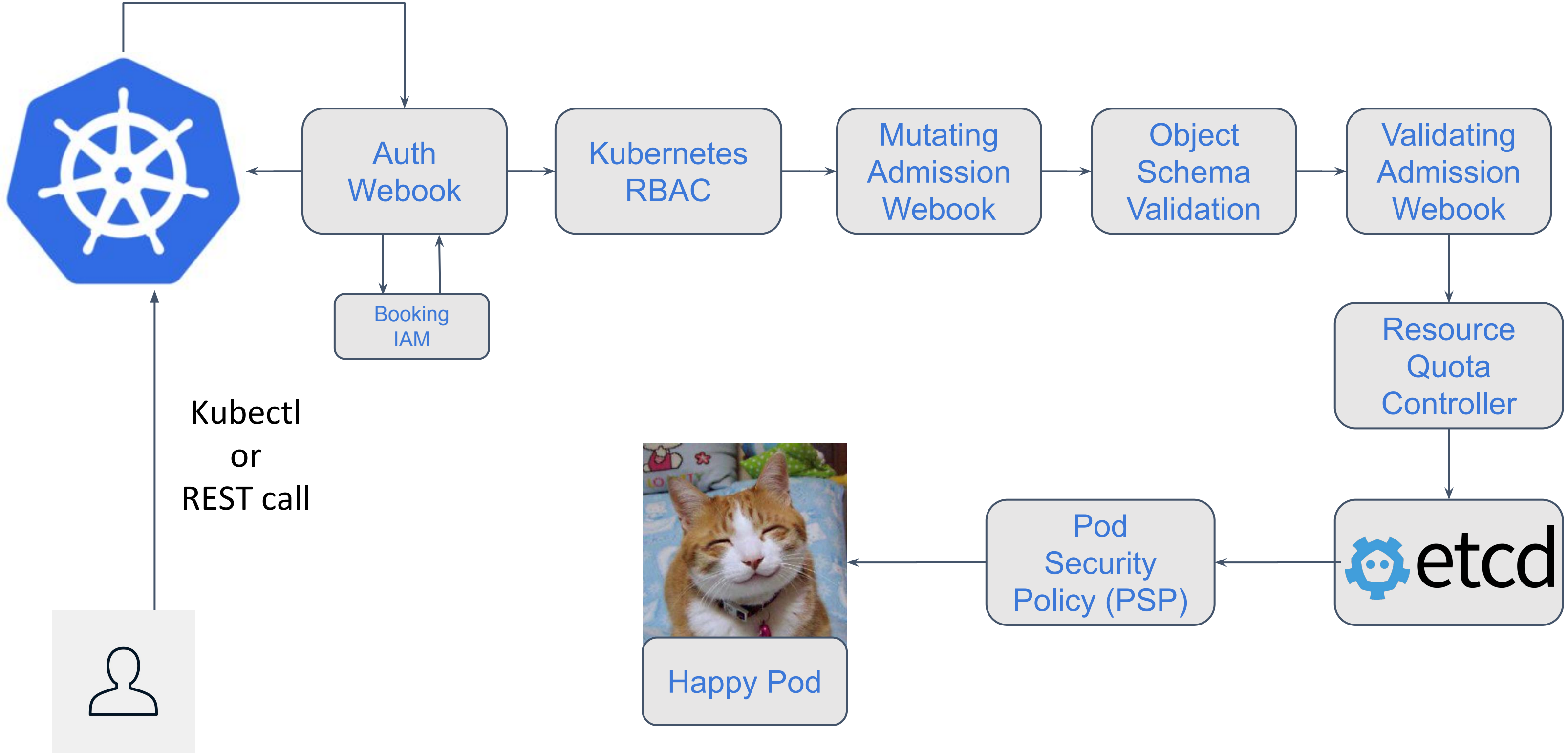
Resource Quota Controller

etcd

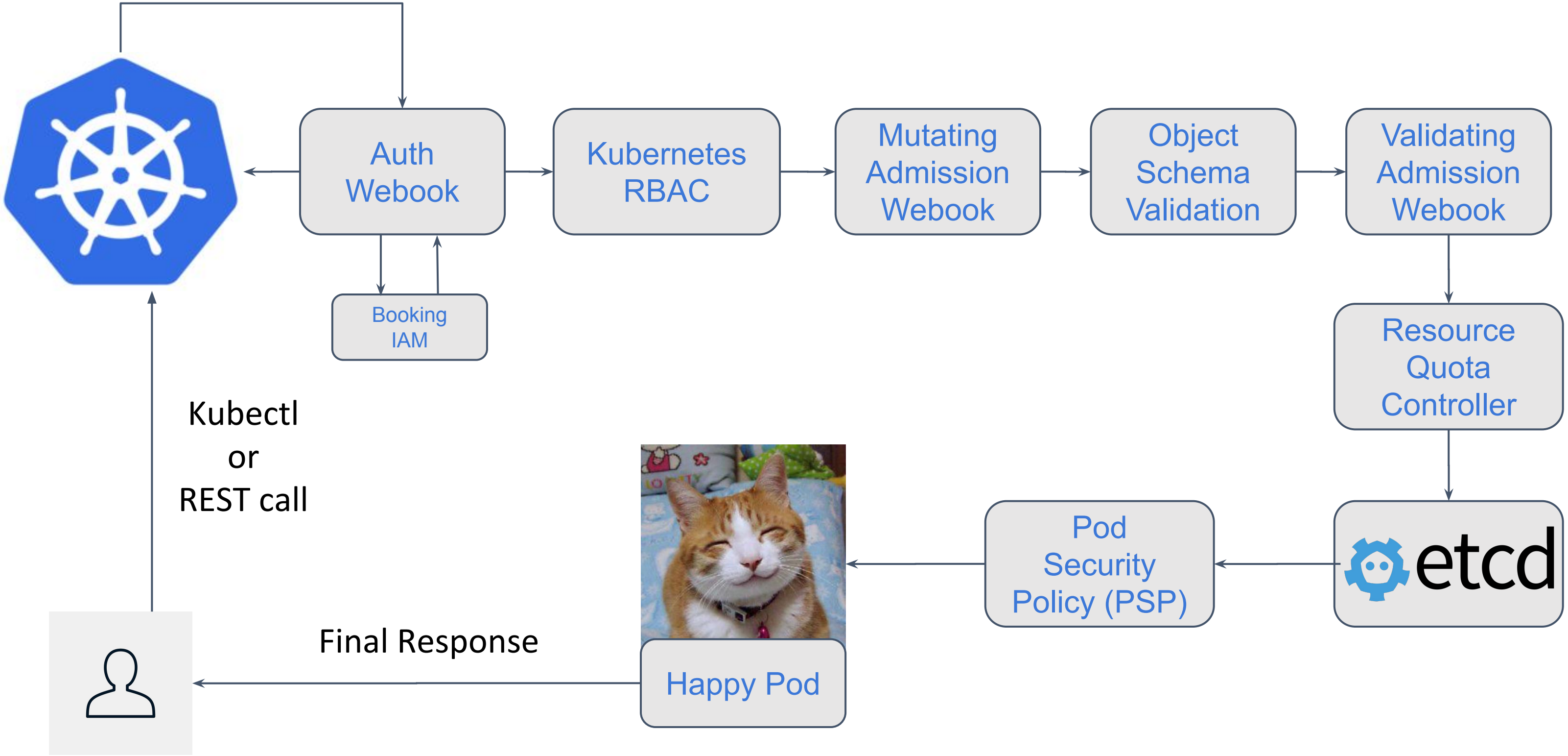
Pod Security Policy (PSP)

Kubectl or REST call

Lifecycle of a kubectl command



Lifecycle of a kubectl command



Key Takeaways

Key Takeaways

- Customize workflow using custom controllers (maybe using a framework), which opens the door to limitless automation

Key Takeaways

- Customize workflow using custom controllers (maybe using a framework), which opens the door to limitless automation
- Re-use your organization’s existing auth workflow with your kubernetes setup using Kubernetes auth webhook

Key Takeaways

- Customize workflow using custom controllers (maybe using a framework), which opens the door to limitless automation

- Re-use your organization’s existing auth workflow with your kubernetes setup using Kubernetes auth webhook

- Kubernetes admission controllers provide (using webhooks) a lot of opportunities to secure and customize resources being created in your kubernetes clusters

Key Takeaways

- Customize workflow using custom controllers (maybe using a framework), which opens the door to limitless automation
- Re-use your organization’s existing auth workflow with your kubernetes setup using Kubernetes auth webhook
- Kubernetes admission controllers provide (using webhooks) a lot of opportunities to secure and customize resources being created in your kubernetes clusters
- Take the opportunity to use PSPs (Pod Security Policies) to enforce a secure environment for your workloads to run

Thank you!

Thank you!

We’re Hiring

careers.booking.com

Booking.com

Questions

Q/A

Scan me