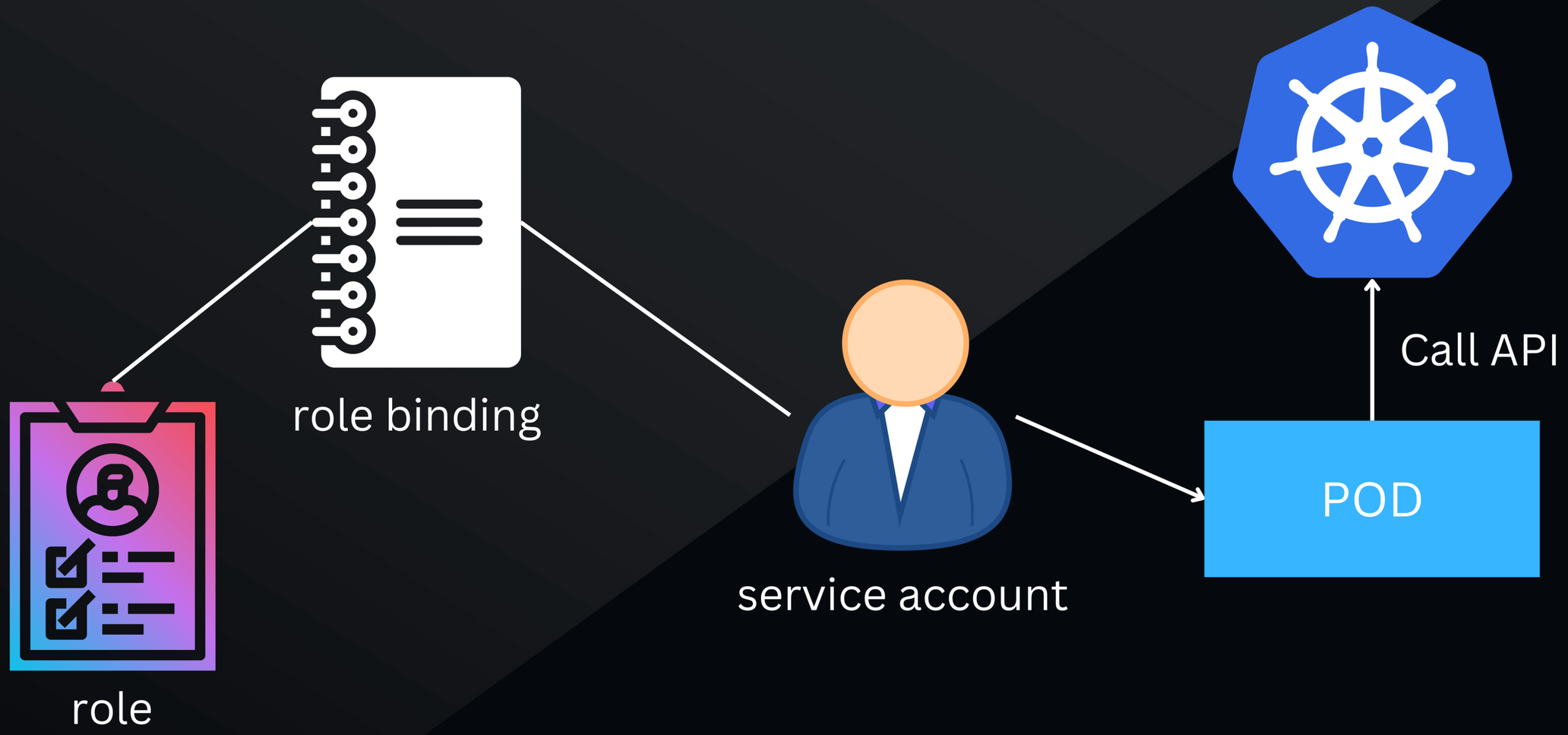
SERVICE ACCOUNT

- Service account is an identity that is **used by pods to access the Kubernetes API and other Kubernetes resources**
- It has **limited permissions** and can only access resources that are explicitly allowed by Kubernetes **RBAC (Role-Based Access Control) policies**
- **E**very pod in the cluster needs to have **one (and only one) service** account
- Usage
  - **Accessing the Kubernetes API** to read or modify configuration information
  - **Creating or modifying Kubernetes resources**, such as pods, services, or deployments
  - **Authenticating with external systems**, such as cloud providers or databases, using credentials stored in a Kubernetes secret

SERVICE ACCOUNT DIAGRAM



SERVICE ACCOUNT & ROLE

```yaml
apiVersion: v1
kind: ServiceAccount
metadata:
  name: jarvis-svc-acct
  namespace: default

```

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: read-only-role
  namespace: default
rules:
- apiGroups: # core group
  - ""
  resources:
  - pods
  verbs:
  - get
  - list
  - watch
```

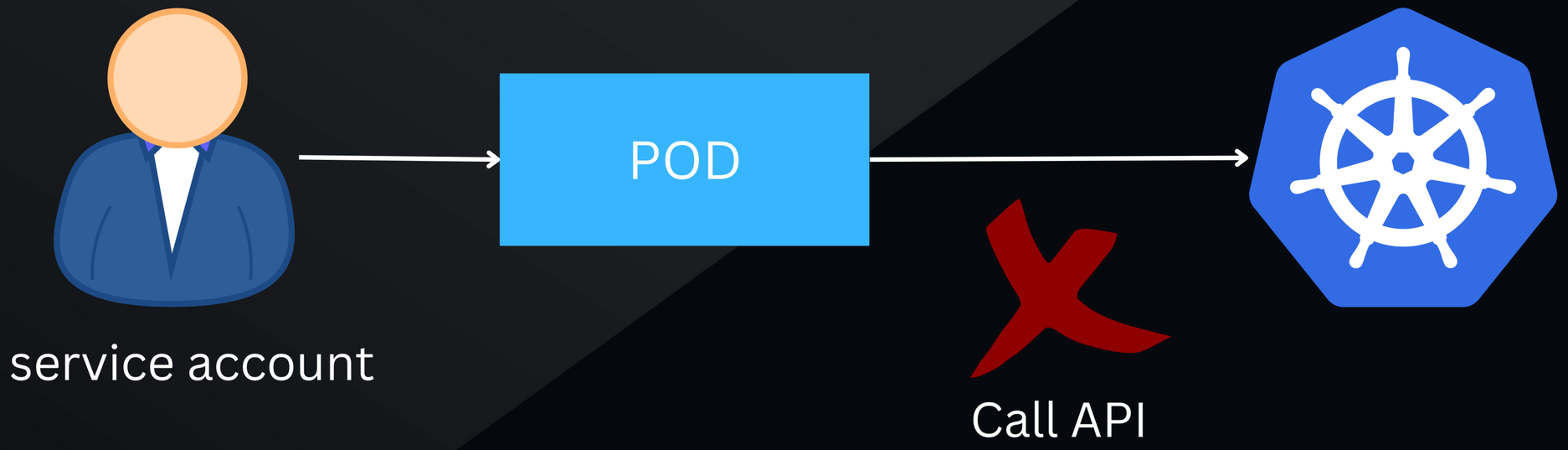
RB & ASSOCIATE TO POD

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: jarvis-rolebinding
  namespace: default
roleRef:
  kind: Role
  name: read-only-role
  apiGroup: rbac.authorization.k8s.io
subjects:
- kind: ServiceAccount
  name: jarvis-svc-acct
  namespace: default

```

```yaml
spec:
  # serviceAccountName: jarvis-svc-acct
  containers:
    - name: django-app
      image: joonlee0228/django:0.0.1
      resources:
        requests:
          cpu: "100m"
      env:
```

TEST W/O SERVICE ACCOUNT



TEST WITH SERVICE ACCOUNT