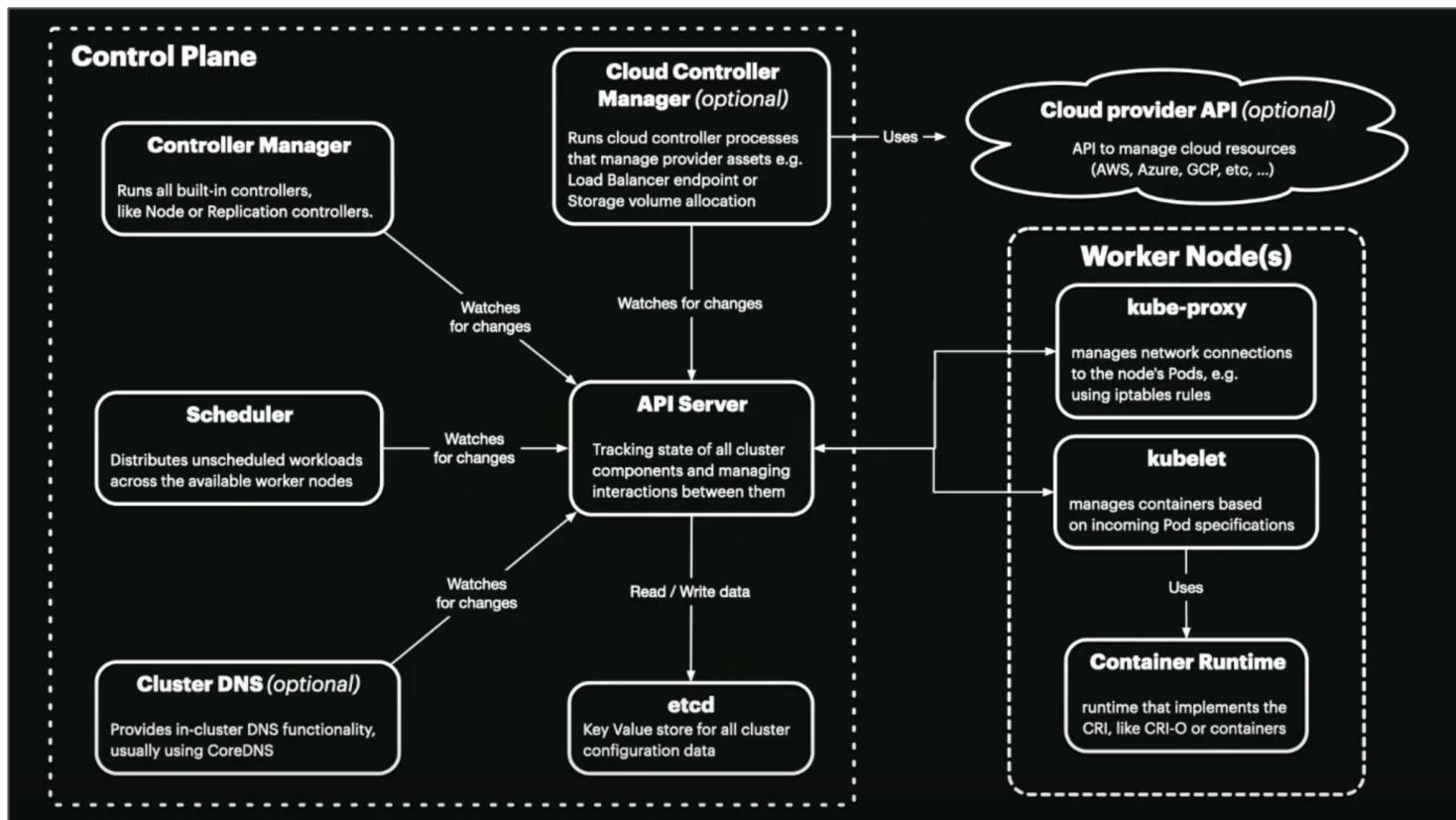
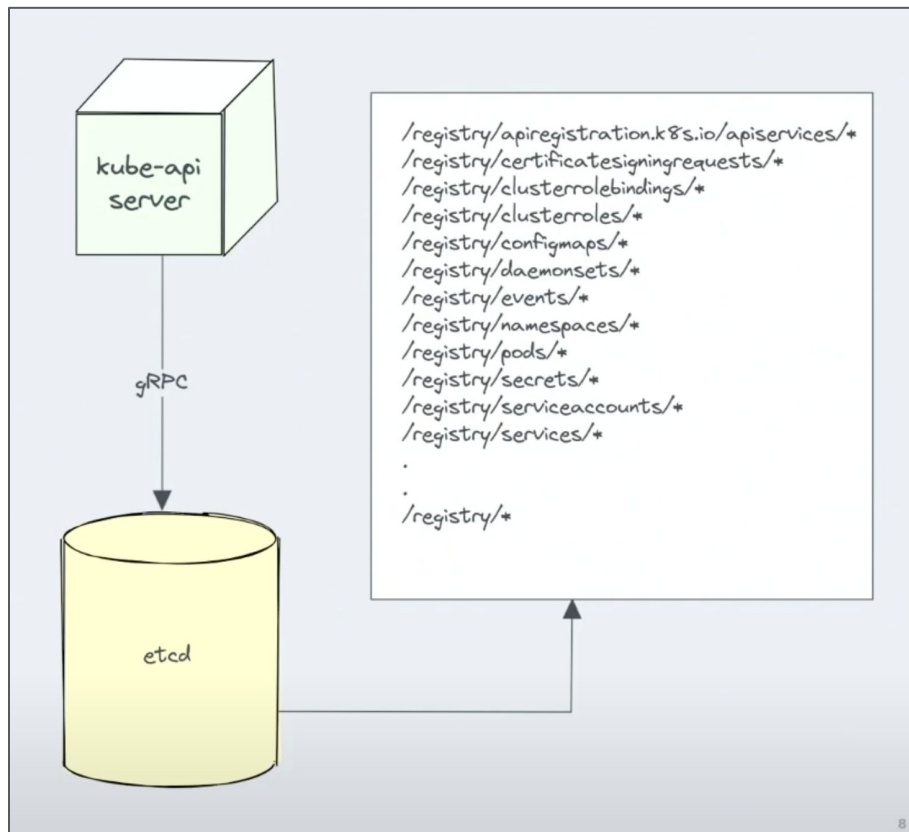


# K8S CRDs and Controller

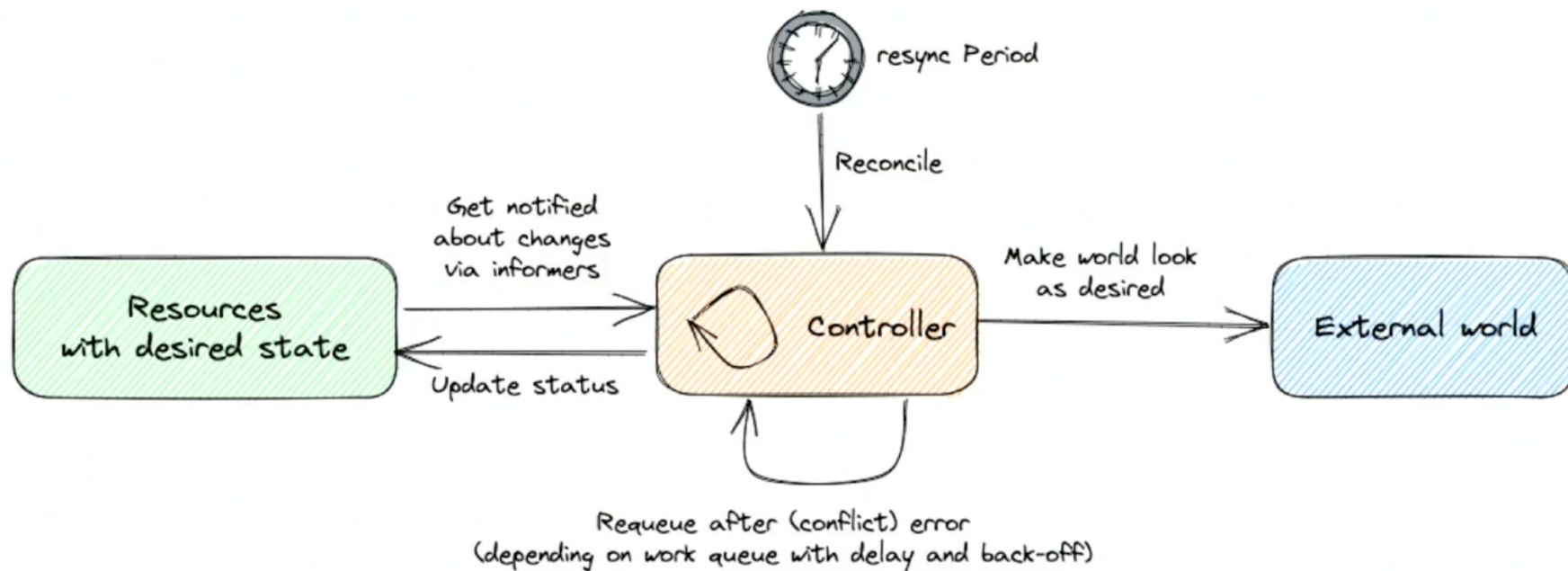
# Moving parts



# API Server



# Reconciliation loop



# Why Python?

- The majority of K8s controllers and operators are written in Go, which is the easy path
- Kubernetes is complicated and operator-sdk generate too much items to easily grasp
- Kopf a framework for making K8s operator development easier (started Zalando)
- IMO Python is simpler and has great existing tooling



# What are we using?

- [kopf](#), operator framework, handles events, retry and much more
- [kr8s](#), kubernetes client that feels familiar for folks who already know how to use kubectl
- [kubecrd](#), simplify working with CRDs

# Example CRD

```
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  name: mongouserresources.mkramb.com
spec:
  group: mkramb.com
  names:
    kind: MongoUserResource
    plural: mongouserresources
    singular: mongouserresource
  scope: Namespaced
  versions:
    - name: v1
      schema:
        openAPIV3Schema:
          properties:
            spec:
              properties:
                password:
                  type: string
                roles:
                  default: []
                  description: List of User roles
                  items:
                    type: string
                  type: array
                  uniqueItems: false
                username:
                  type: string
              required:
                - username
                - password
              type: object
            type: object
          served: true
          storage: true
```

# Example CRD item

apiVersion: mkramb.com/v1

kind: MongoUserResource

metadata:

name: mongouserresource123

spec:

username: testuser

password: secret-password

roles:

- read
- readWrite



