Managing Secrets with Confidence –
A Comprehensive Guide to Using Hashicorp Vault in Kubernetes Clusters

ISO GRUPPE

Intro –
Who, What, Why?

ISO GRUPPE

Alain Lompo

Senior Software Developer, DevOps
& Security Consultant in the Public Sector

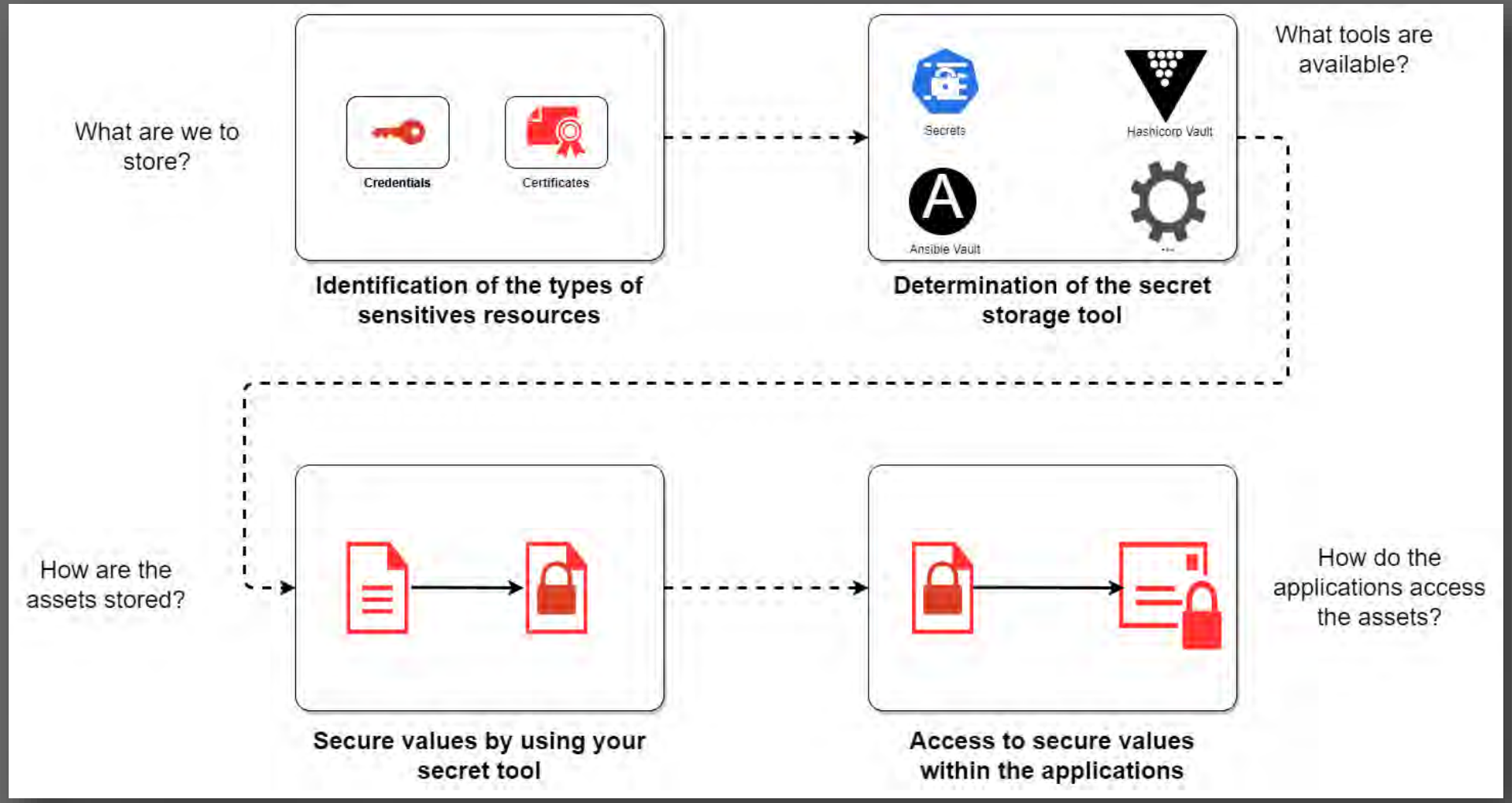
20 years of digital product development, training and consulting

Agenda

- Handling Sensitive Resources in Kubernetes
- Quick Intro to Hashicorp Vault
- Overview of Vault Architecture
- Integrating Vault with Kubernetes
- Real World Scenarios
- Demos
- Wrap-Up

Handling Sensitive Resources

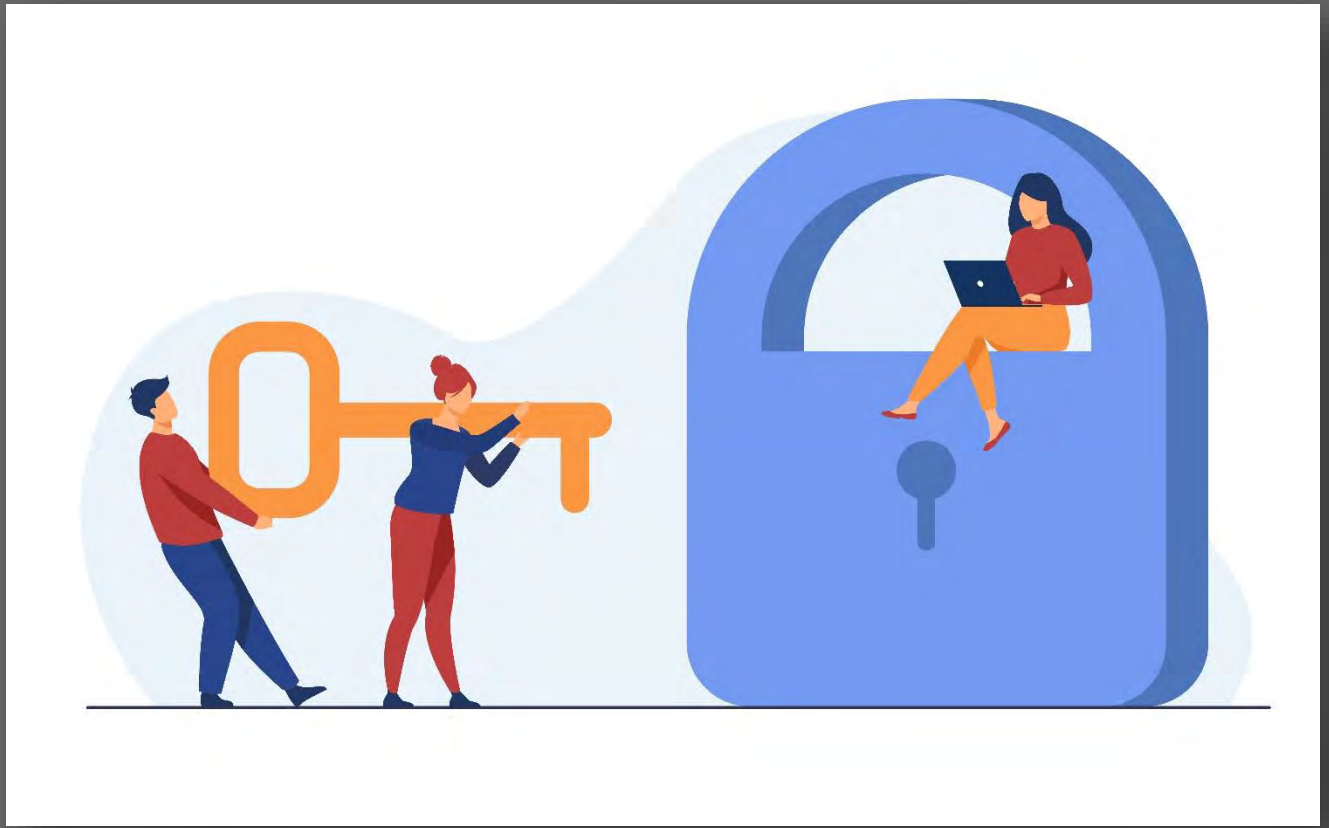
Handling Sensitive Resources in Kubernetes



ISO GRUPPE

Type of Sensitive Resources

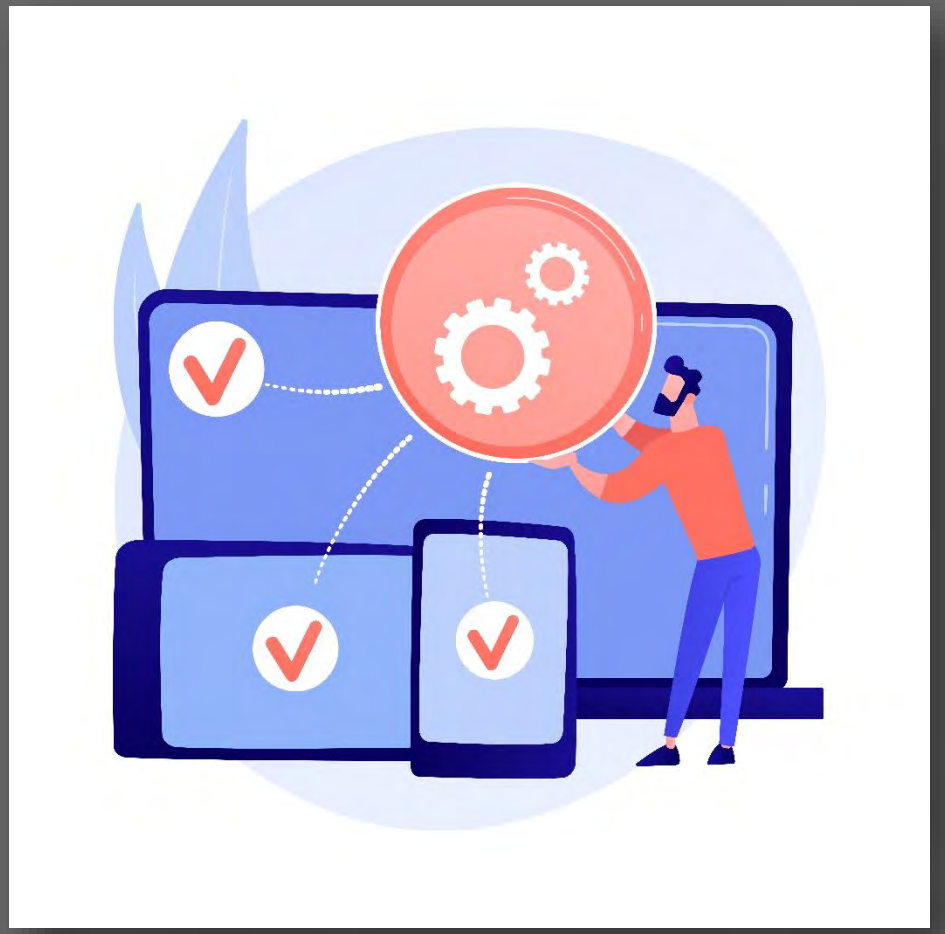
- Credentials
  - Passwords
  - API keys
- Certificates
  - TLS
  - GPG Keys



ISO GRUPPE

Type of Sensitive Resources

- Application Configurations
  - Runtime Arguments
- ...



Storing sensitive information

- What are the available Secret Management Storage solutions?
- How is sensitive asset intended to be used?
- What application framework is being used?
  - What are the options for injecting external config?

The Kubernetes Secret Resource

- Natural default (included in every Kube distribution)

- Provide some “**form**” of protection
  - Values not encrypted
  - Instead Base64 encoded

The Kubernetes Secret Resource

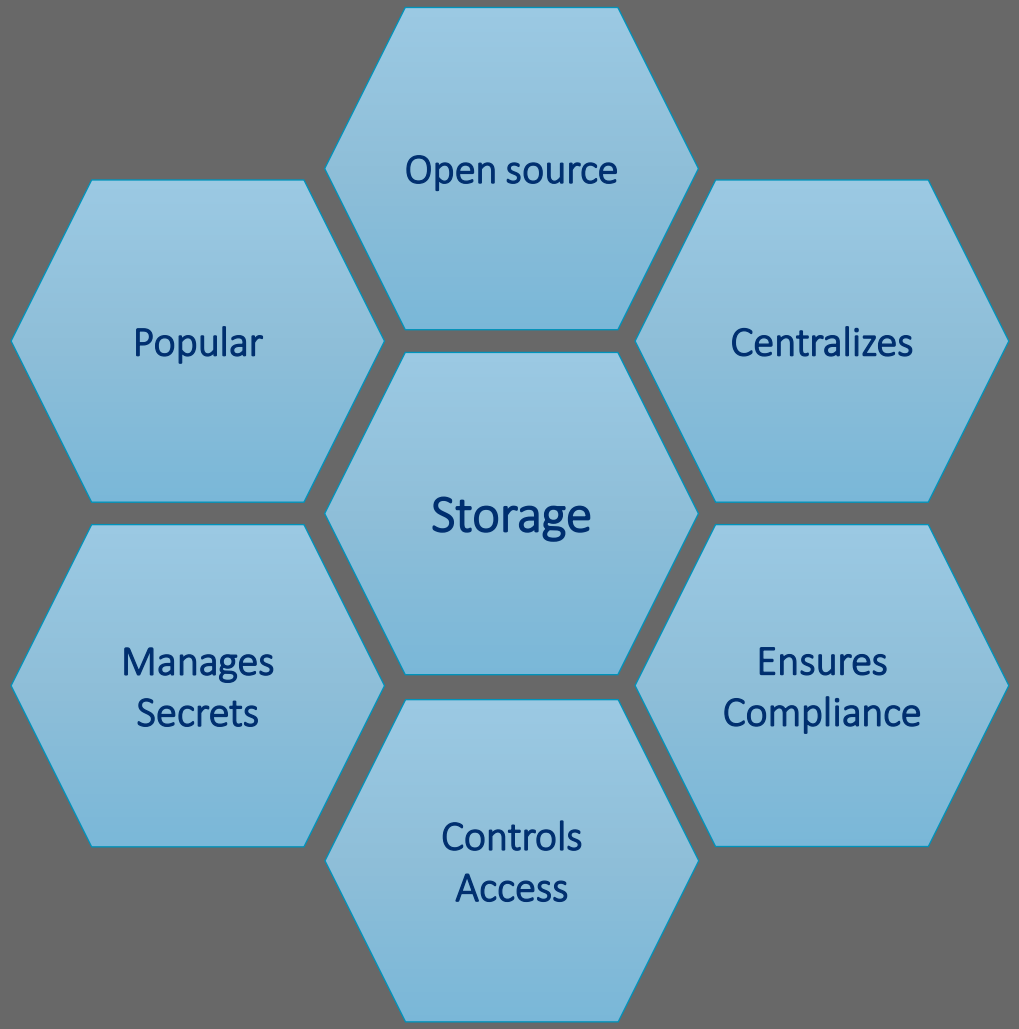
```
apiVersion: v1
kind: Secret
metadata:
      name: hello-secret
type: Opaque
stringData:
    hello.message: Hello
Conf42
```

kubectl apply -f hello-secret-config.yaml

ISO GRUPPE

Quick Intro to Hashicorp Vault

What ist Vault?



ISO GRUPPE

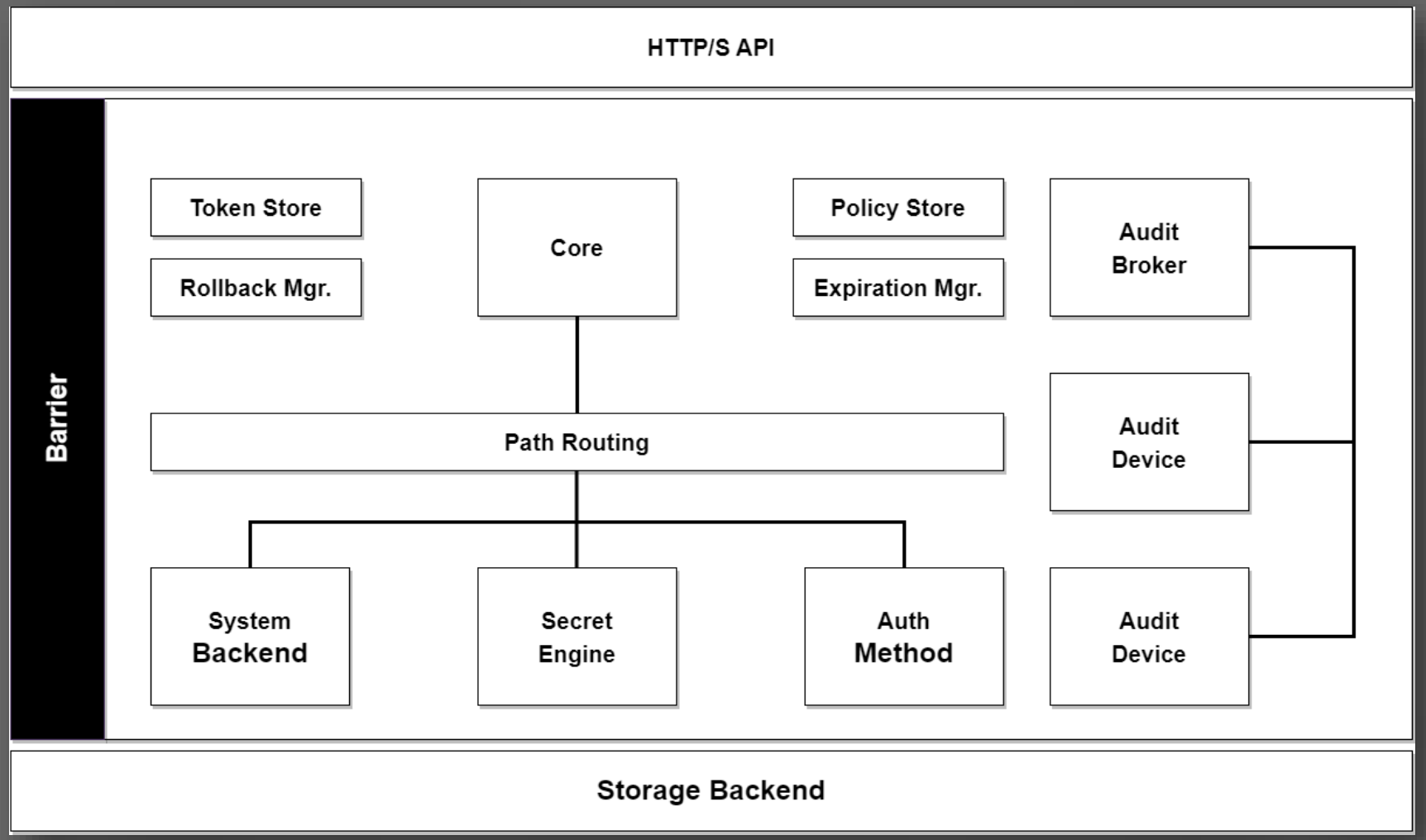
Quick Intro to Hashicorp Vault...

High level features

- Simplify Secret Management.
- Protect Sensitive Data.
- Can be integrated with various cloud and infrastructure platforms.
- Provide a unified and secured solution for Secret Management.

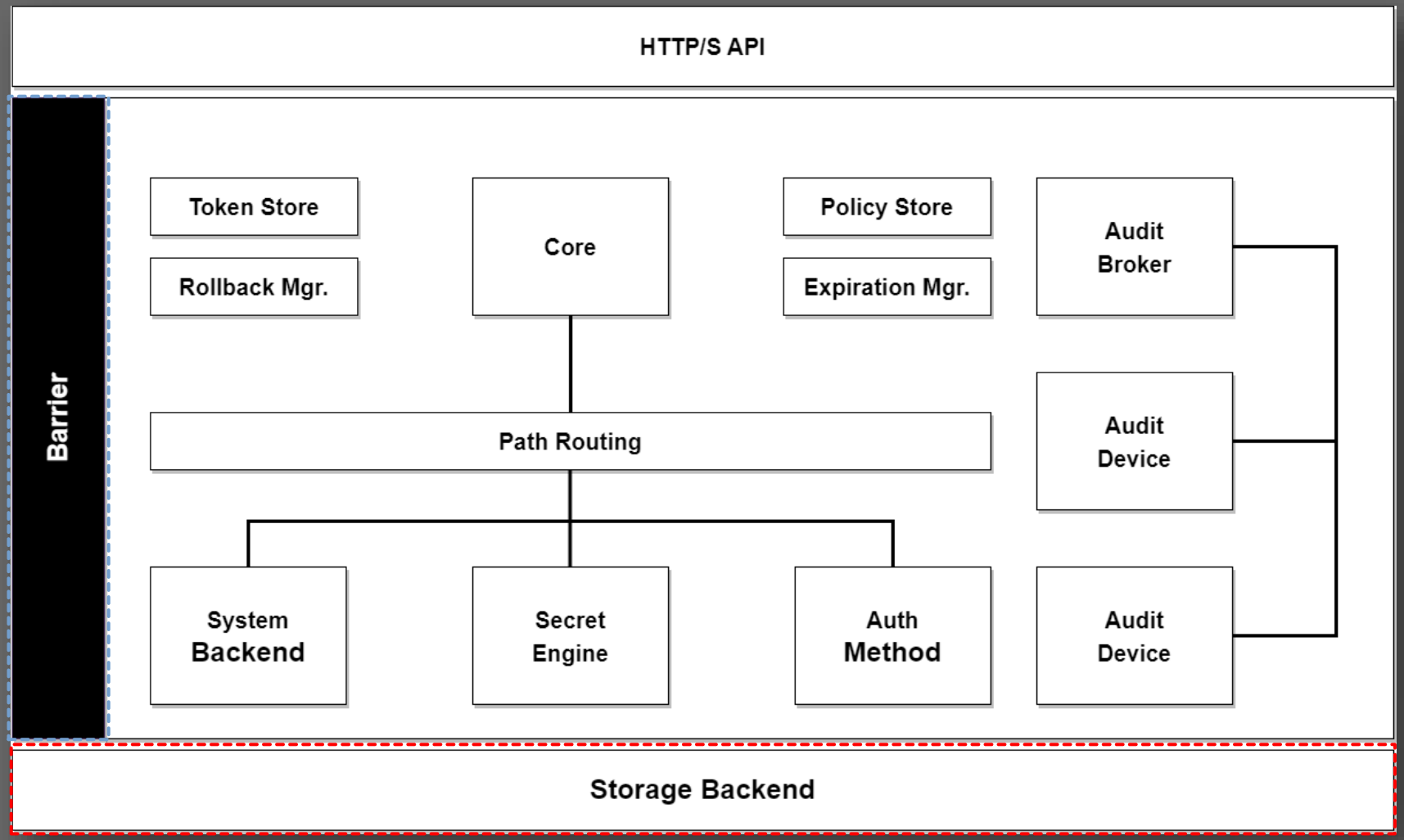
Overview of Vault Architecture

Overview of Vault Architecture



ISO GRUPPE

The Vault Encryption Layer

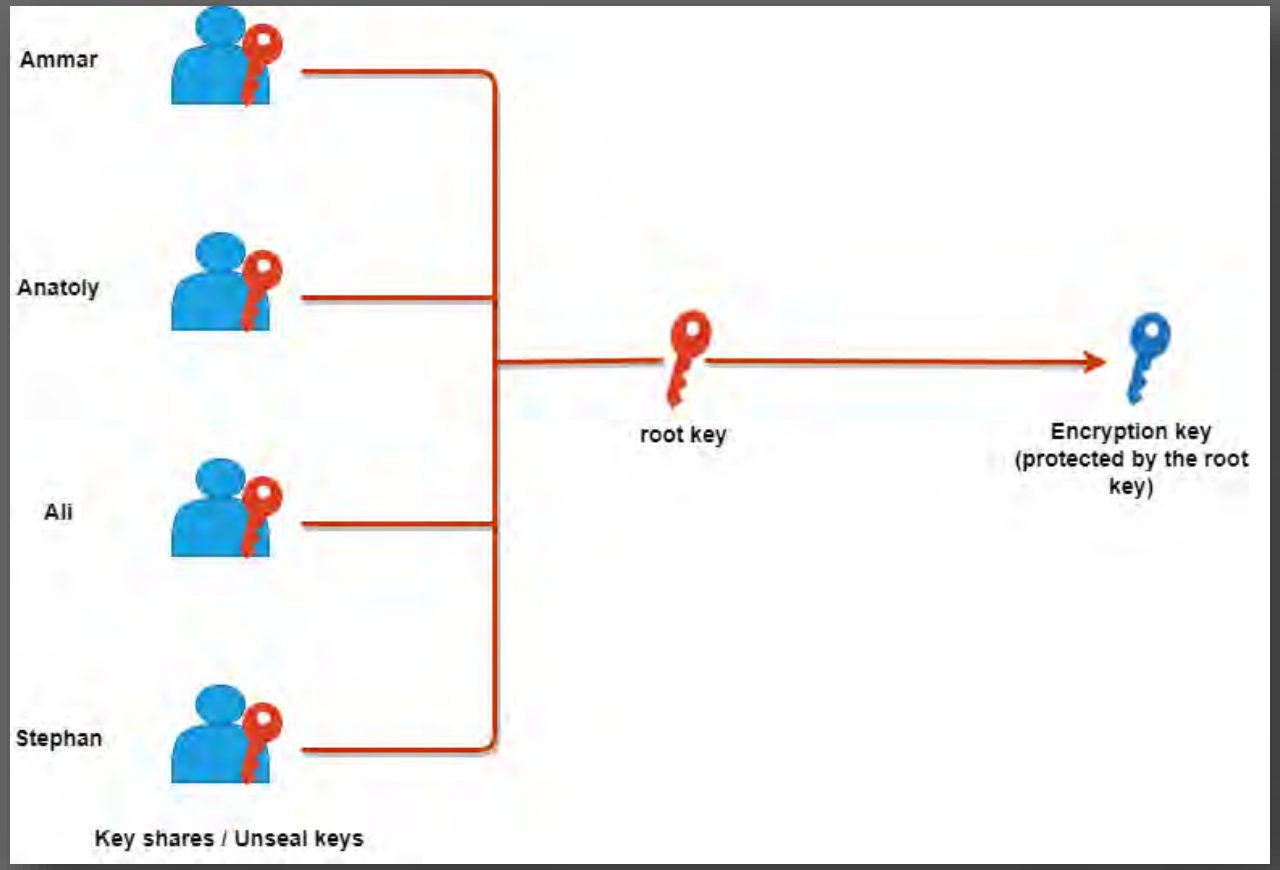


ISO GRUPPE

Unsealing Vault

- At start a Vault Server begins in a **sealed state**:
  - It **MUST** be unsealed before any operation can be performed
  - Unsealing is done by providing the unseal keys.
  - At initialization Vault generates an encryption key used to protect all Vault datas
  - The encryption key is protected by a root key that is stored alongside all other Vault data, but encrypted by another mechanism: the unseal key

Vault and Shamir’s secret sharing

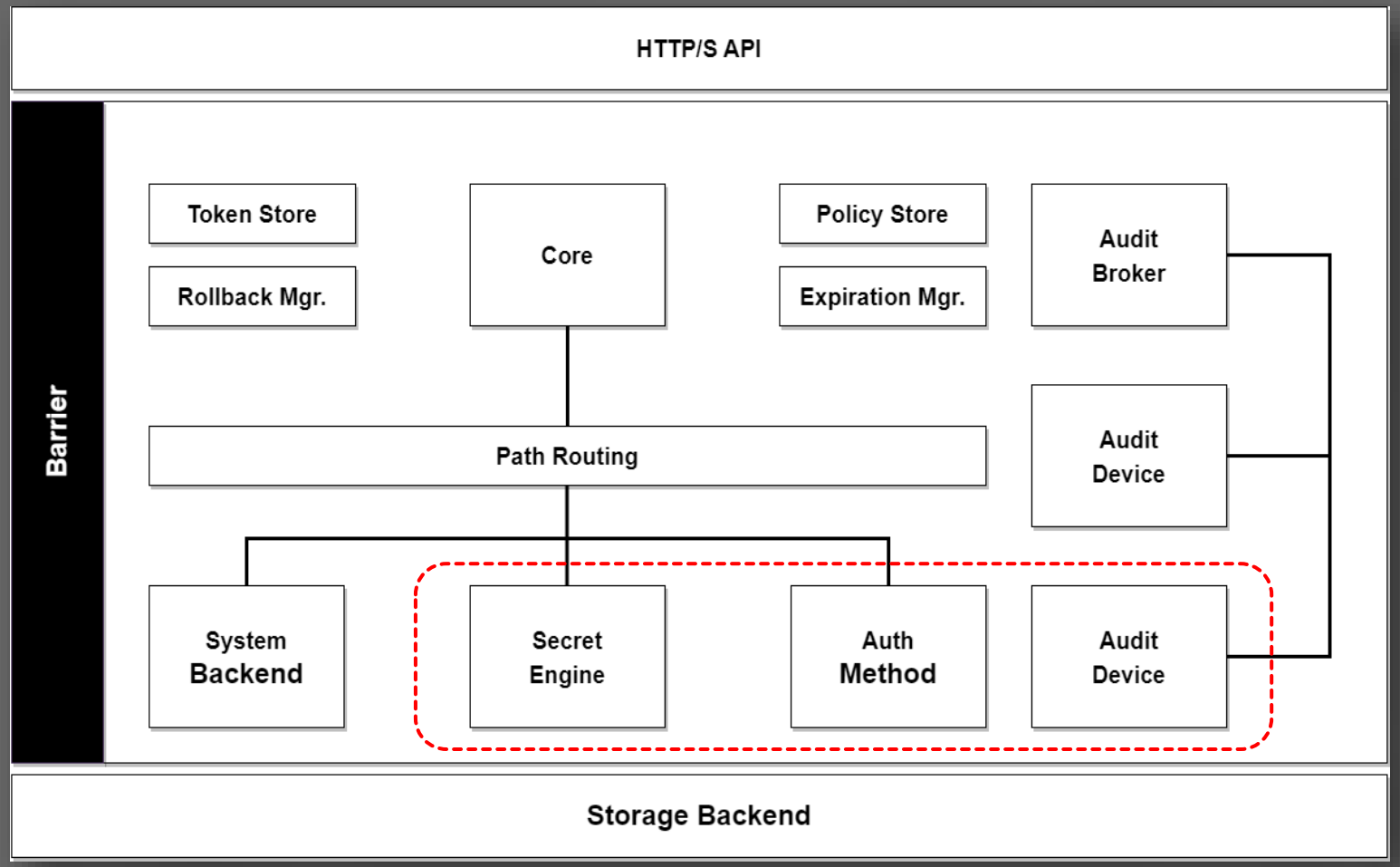


ISO GRUPPE

Use of the encryption key

- Once Vault retrieves the encryption key:
  - It decrypts the data in the storage backend
  - It enters in the unsealed state
  - Once unsealed, it loads the configured **audit devices**, **auth methods** and **secret engines**.

Securing Audit devices, Auth methods and Secret engines



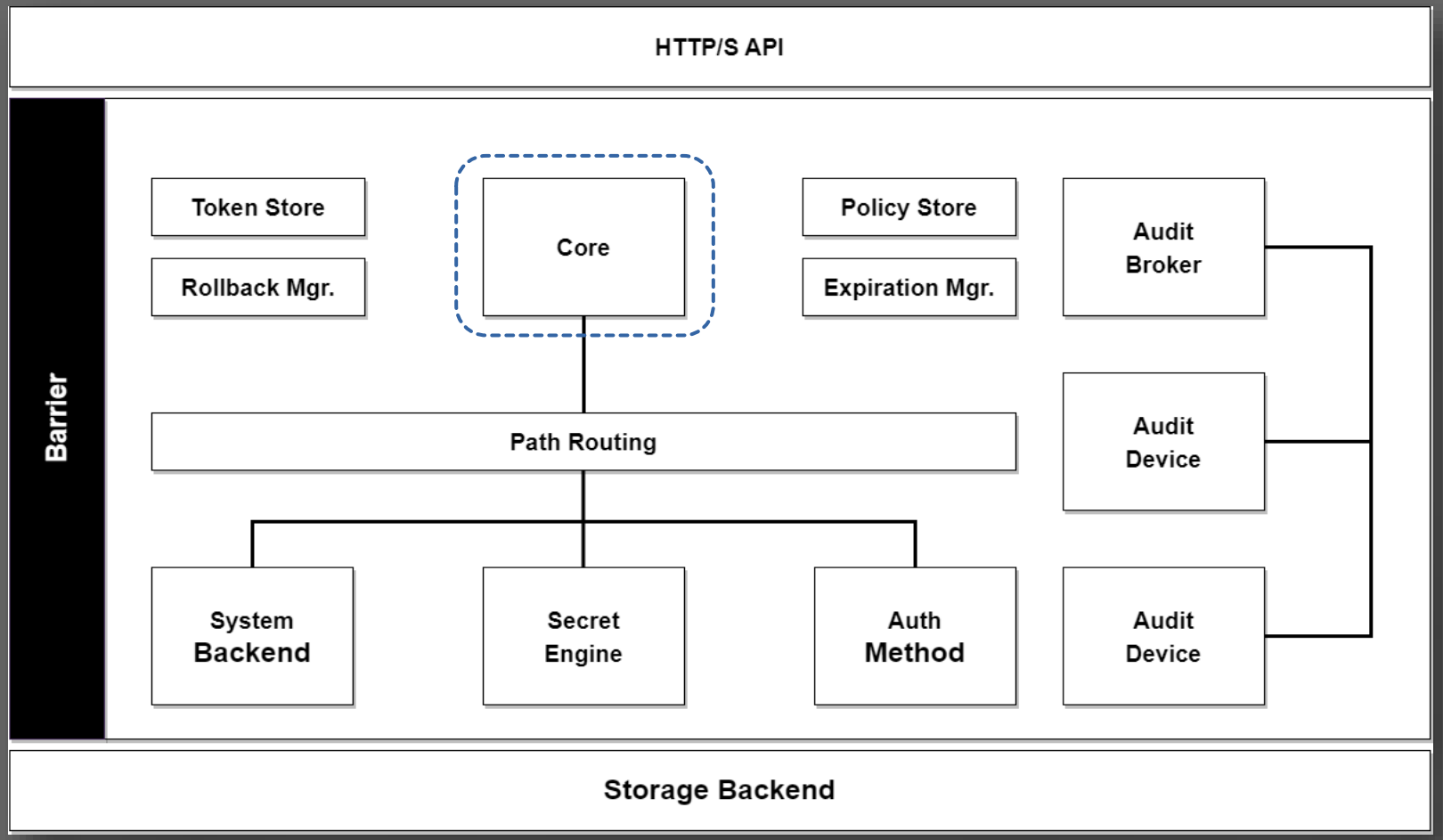
ISO GRUPPE

Securing Audit devices, Auth methods and Secret engines

Configuration

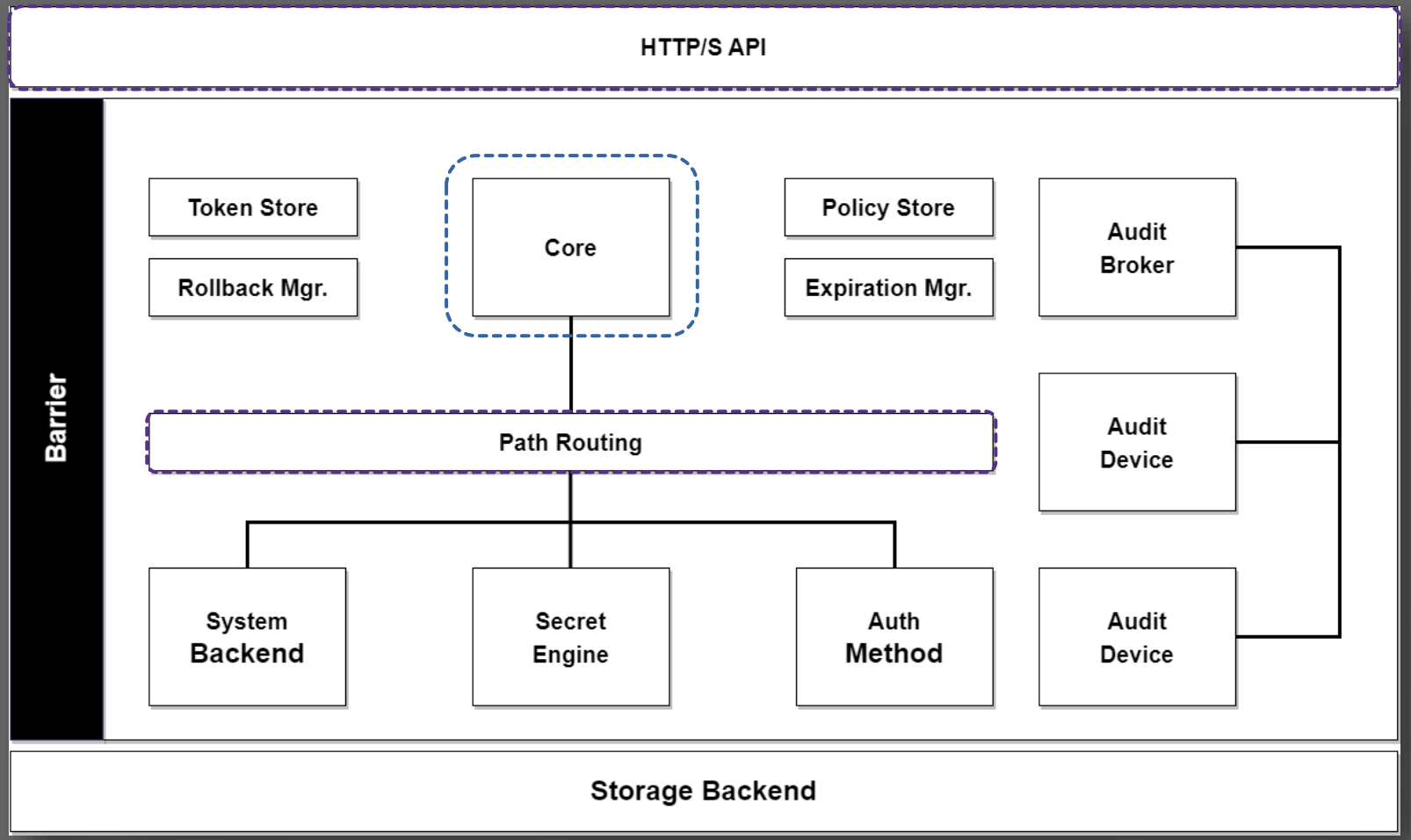
- Security sensitive.
- Stored in Vault.
  - ✓ Changes are protected by ACL
  - ✓ And tracked by Audit Logs
- Cannot be specified outside of Vault.
- Users with permissions can modify them.

Role of the Core



ISO GRUPPE

Role of the Core



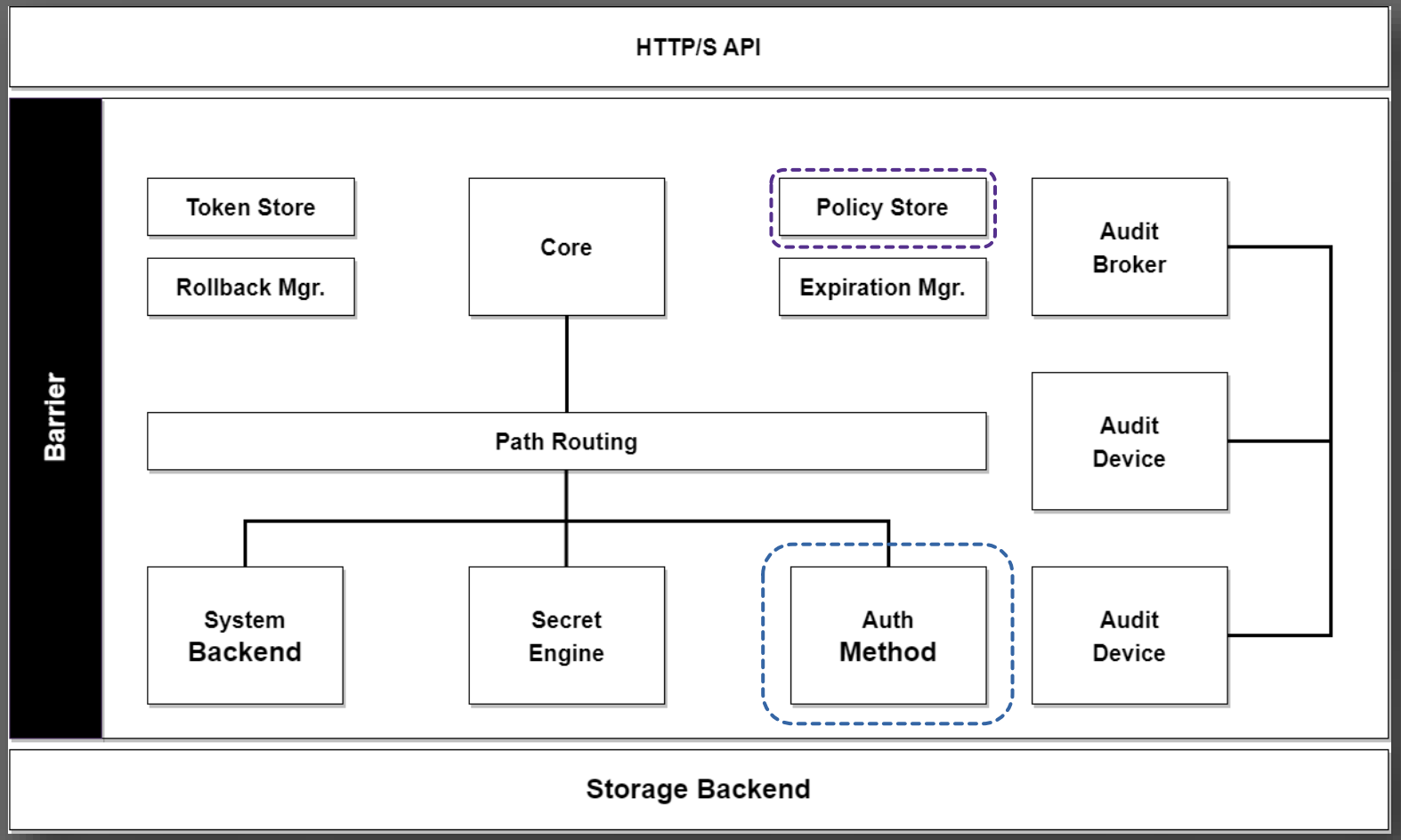
ISO GRUPPE

Role of the Core

- Once Vault is unsealed, request may be processed from HTTP/S API to the core

- The Core:
  - Manage the flow of requests through the System.
  - Enforce ACL.
  - Ensure Audit Logging is done.

Vault auth mechanism



ISO GRUPPE

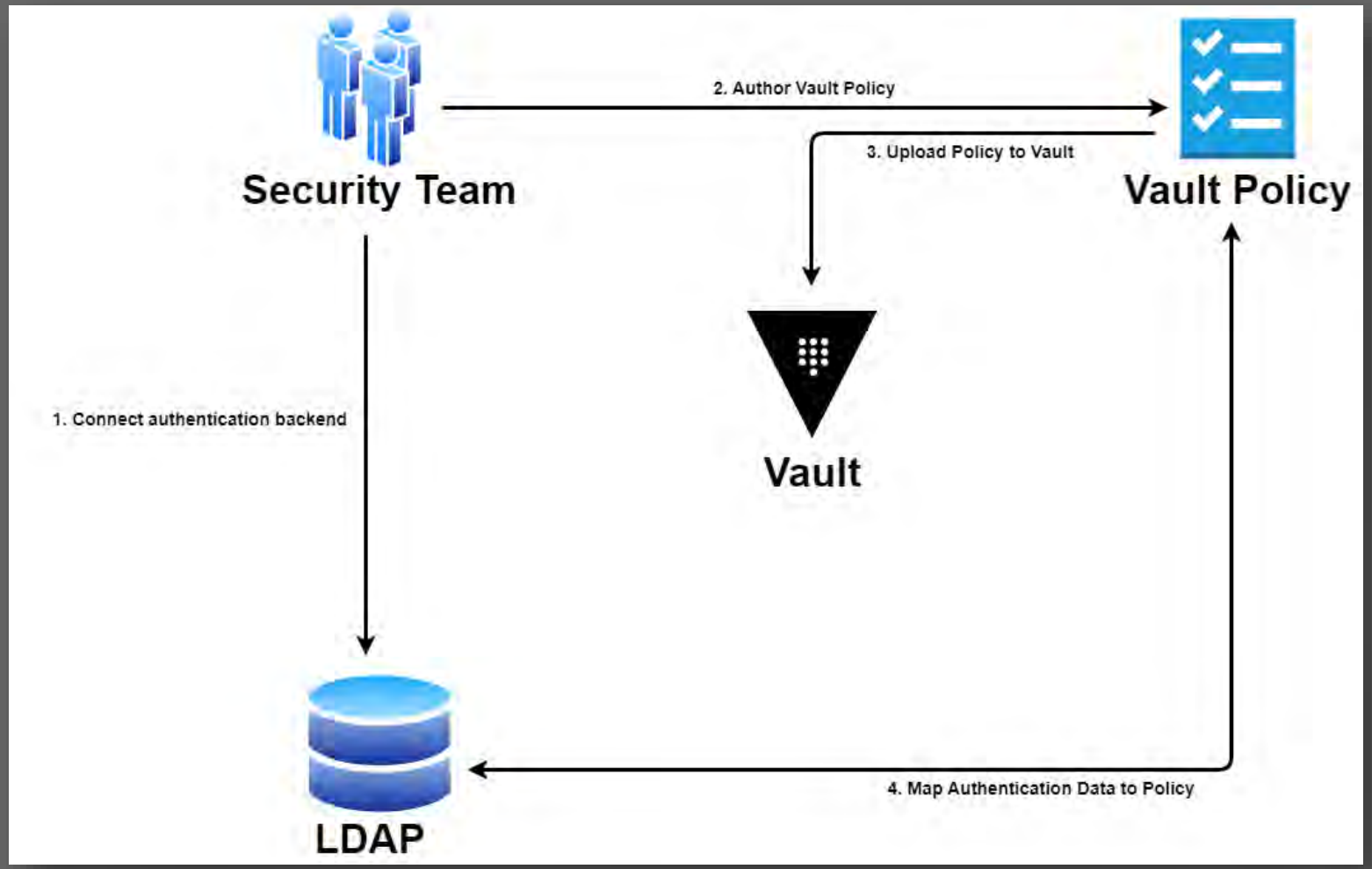
Vault auth mechanism

- A client that connects to Vault needs to authenticate

- Vault provide:
  - Configurable auth methods
  - Offers flexibility within the auth mechanism used
  - Operators may use username/password or Github (for example)
  - Applications may use private/public keys or tokens to authenticate

- An auth request that flows through the core and into an auth method:
  - Determines if the request is valid
  - Returns a list of associated policies

Vault Policies



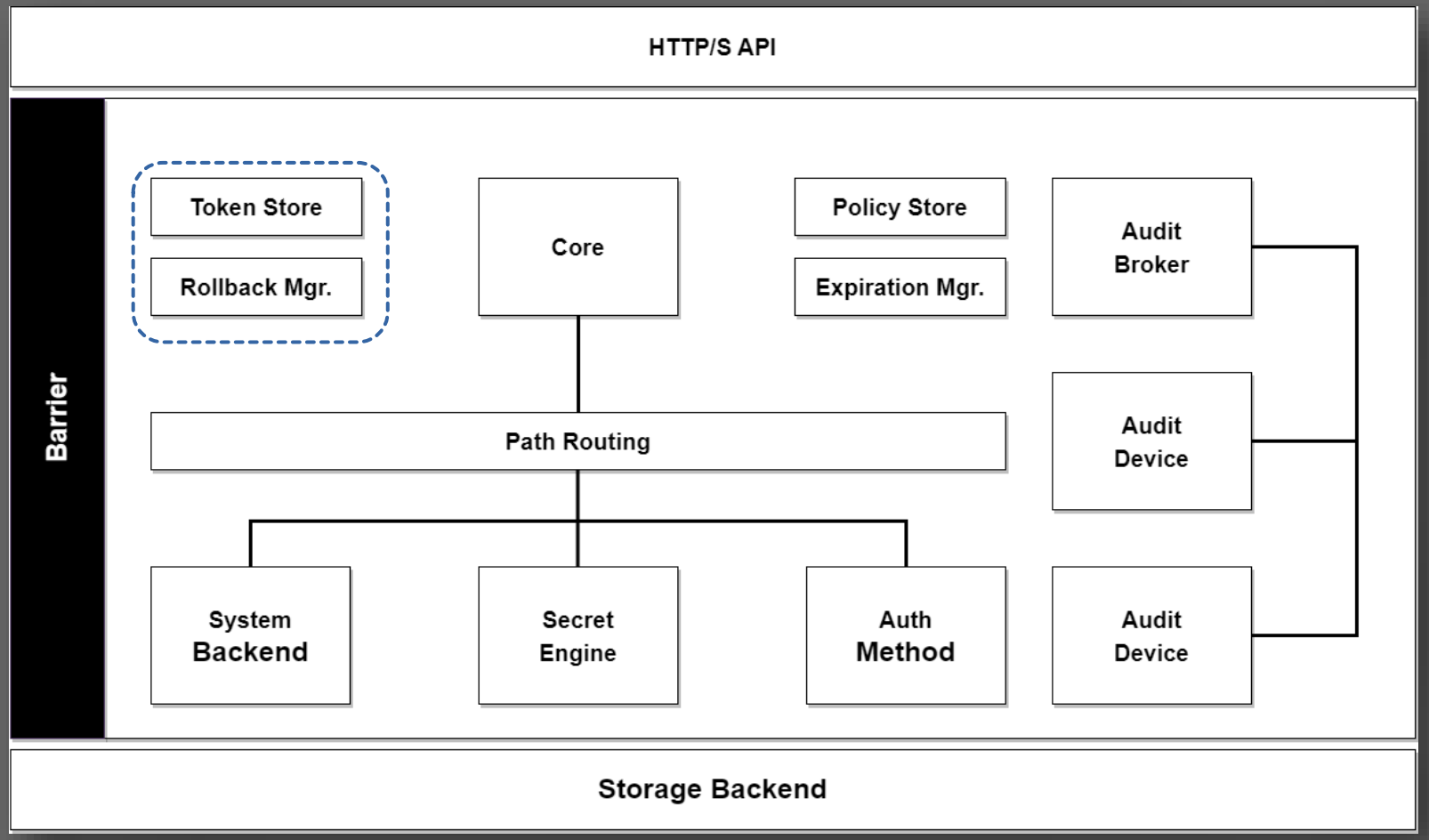
ISO GRUPPE

Vault Policies

- Policies are named ACL rules:
  - Example: the “root” policy built-in and permit access to all resources.
  - Any number of policies can be created
  - It allows fine grained control over paths

- Vault operates in an allowed-access mode

Client Token



ISO GRUPPE

Client Token

- After authentication, an auth method provides a set of applicable policies
- A new client token is generated and managed by the token store
- The client token is used for making future request:
  - Approach similar to auth cookies
- Client token may have an associated lease:
  - May need to be renewed periodically to avoid invalidation

Request authorization flow

- Authenticated requests are made by providing the client token.

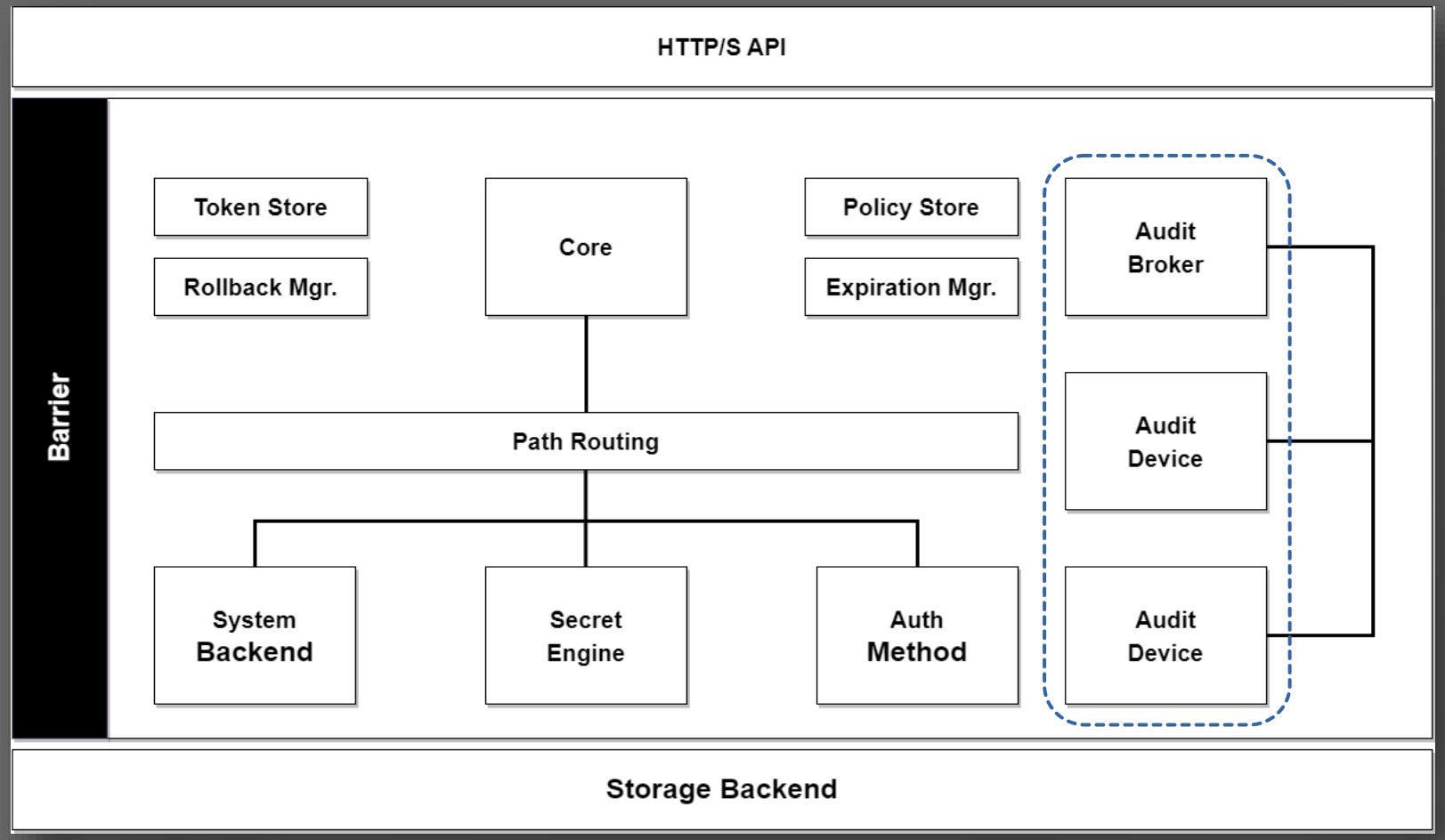
- The client token is used to verify the client
  - Ensuring they are authorized
  - In conformance with the relevant policies
  - Policies (named ACL) are used to authorize the client request

The Secret Engine

- The authorized request is then routed to the secrets engine:
  - It is processed depending on its type

- When the secret engine returns the secret:
  - The Core registers it with the expiration manager
  - It attaches a lease ID
  - Clients uses the lease ID to renew/revoke their secret.
  - The expiration manager automatically revokes the secret if a client allows the lease to expire

The Audit Broker and devices



ISO GRUPPE

The audit broker and devices

- The Core:
  - Logs requests and responses to the audit broker
  - Thus, distributing the requests to all configured audit devices

Other Vault Activities

- Outside of the request flow:
  - The core performs specific background activities
    - Such as lease management
  - Also:
    - Vault handles specific partial-failure cases by using write-ahead logging with a rollback manager.
    - This is transparent to the user and done within the core.

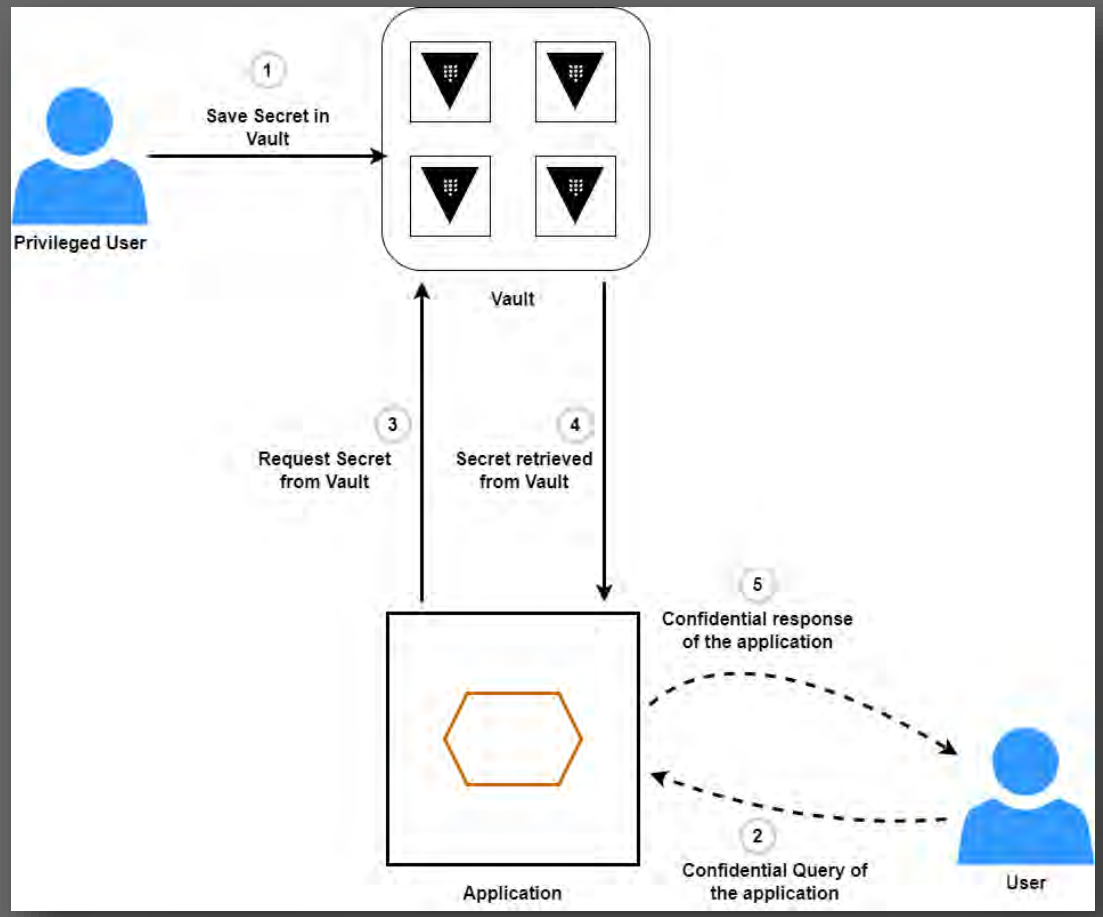
Integrating Vault with Kubernetes

Integrating Vault with Kubernetes

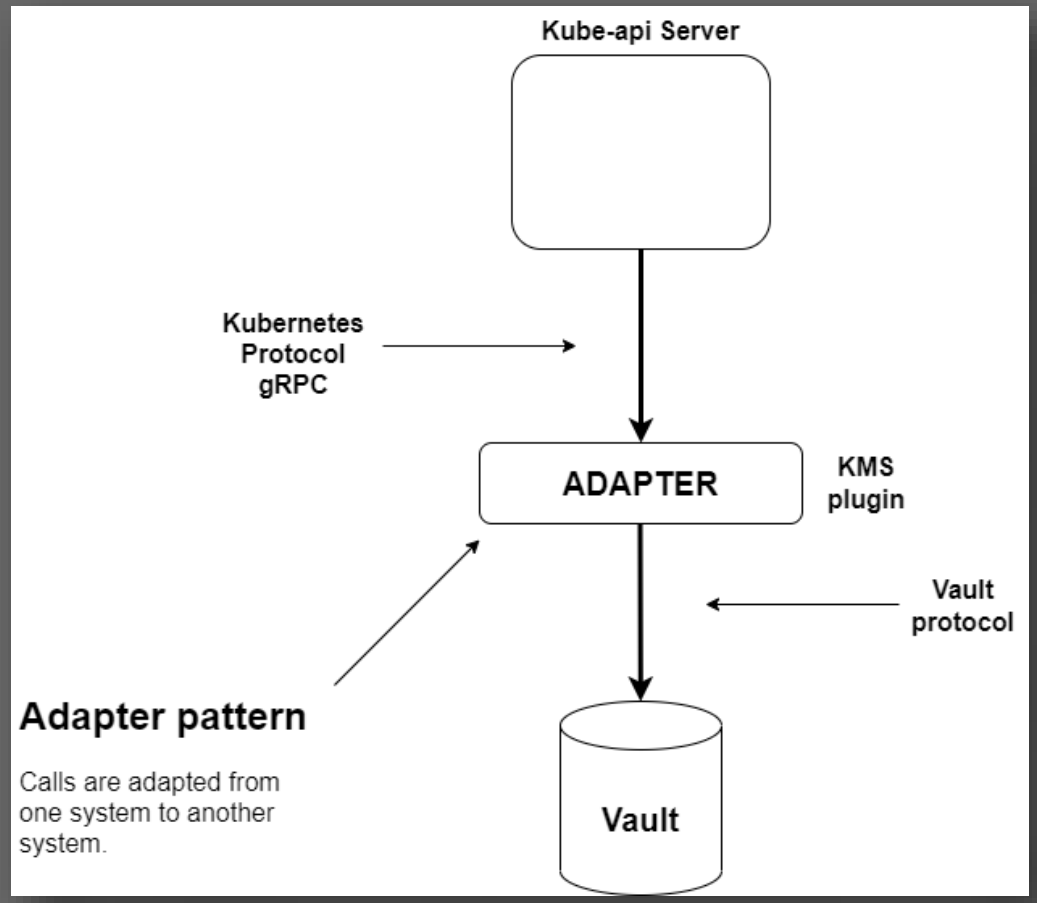
- Why is this important ?
  - Containers are ephemeral
  - Centralized Secret management
  - Access control and auditing
  - Integration with other tools

- Various approaches each with their own benefits and limitations

Vault application access model

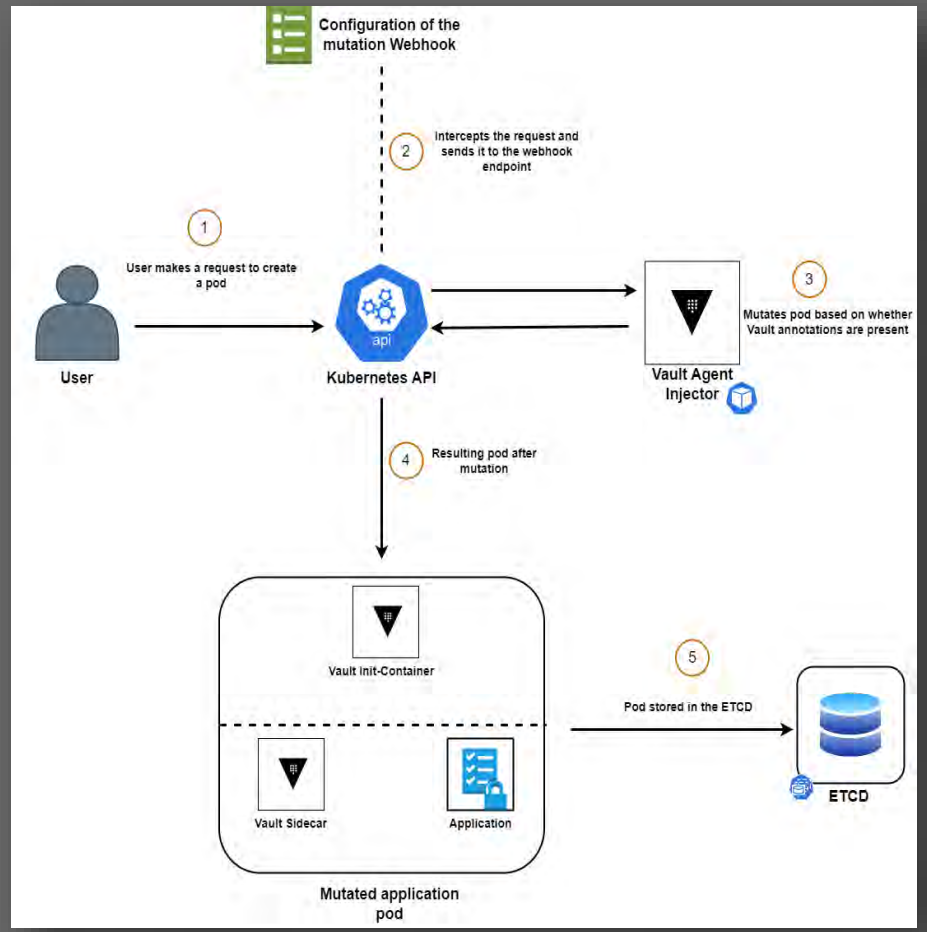


Vault as KMS



ISO GRUPPE

Vault with Agent Injector and sidecar container



Kubernetes Auth Backend

ISO GRUPPE

Kubernetes Auth Backend

```
$ vault auth enable kubernetes
Success! Enabled kubernetes auth method at: kubernetes/
```

```
$ vault write auth/kubernetes/config \
      kubernetes_host=„https://$KUBERNETES_PORT_443_TCP_ADDR:443“
Success! Data writen to: auth/kubernetes/config
```

Kubernetes Auth Backend Benefits

Benefits

No need for manual token distribution and management

Tight control of access based on pod's identity and permission

Seamless integration with Kubernetes

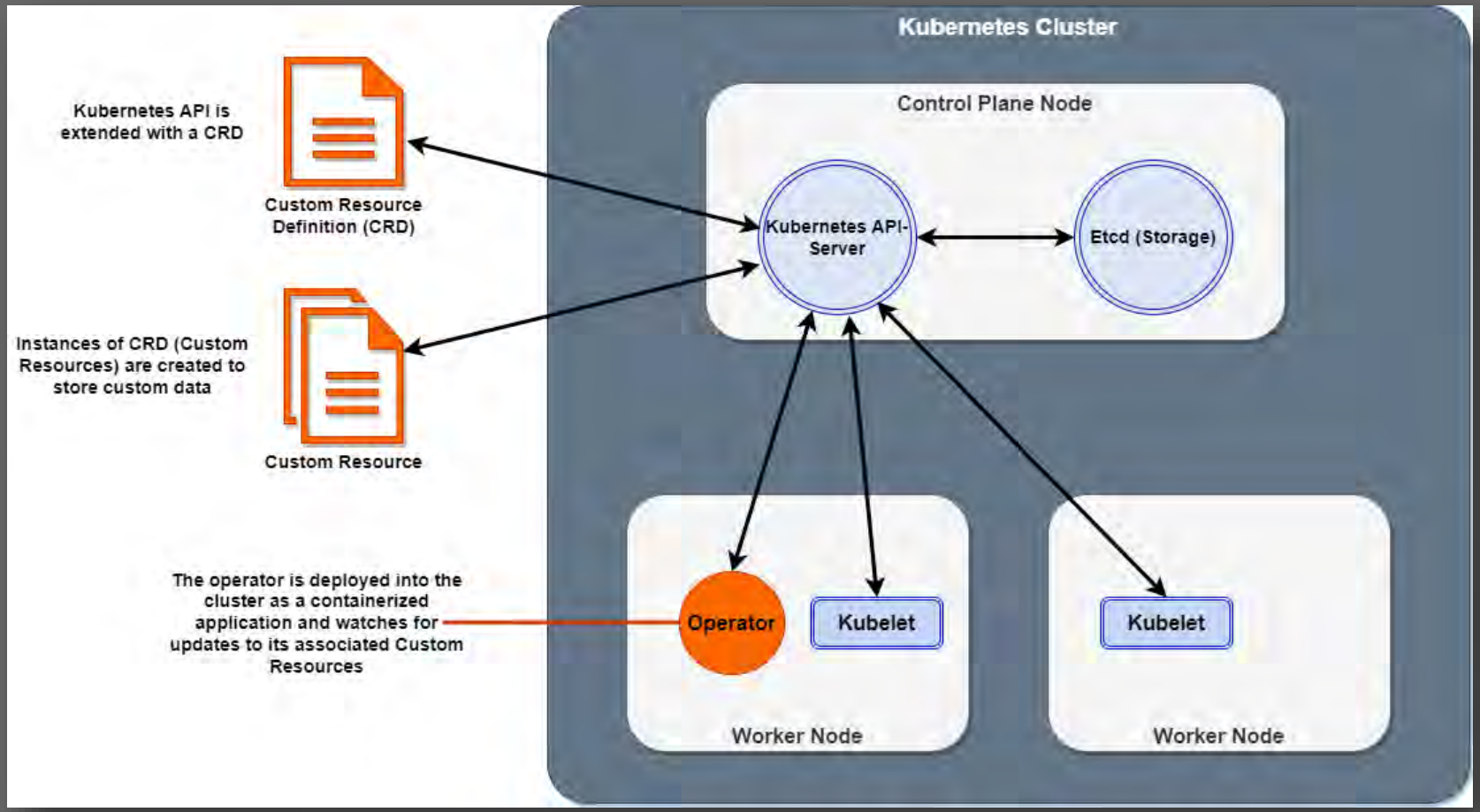
Kubernetes Auth Backend Benefits

Benefits

Automatic token renewal

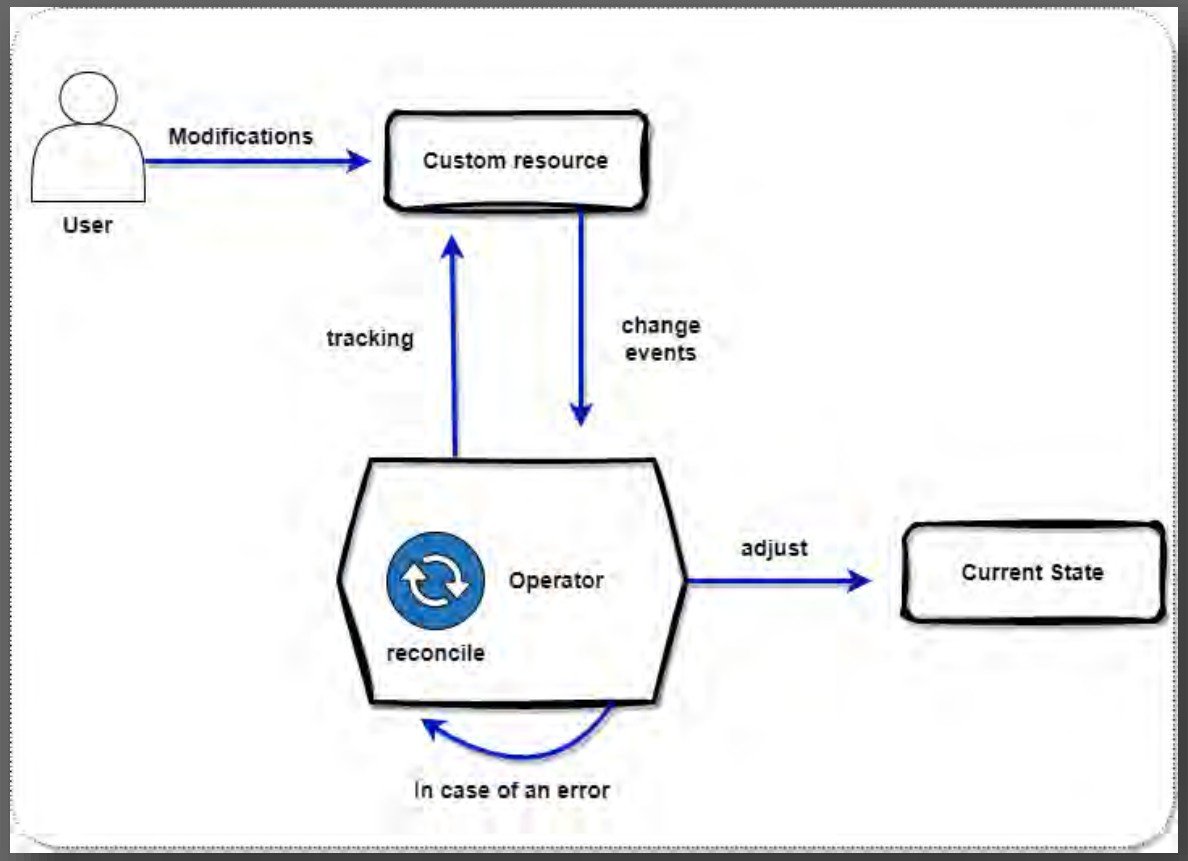
Manage secrets in a secure and scalable fashion in a Kube environment

Custom Resource Definitions (CRDs)...



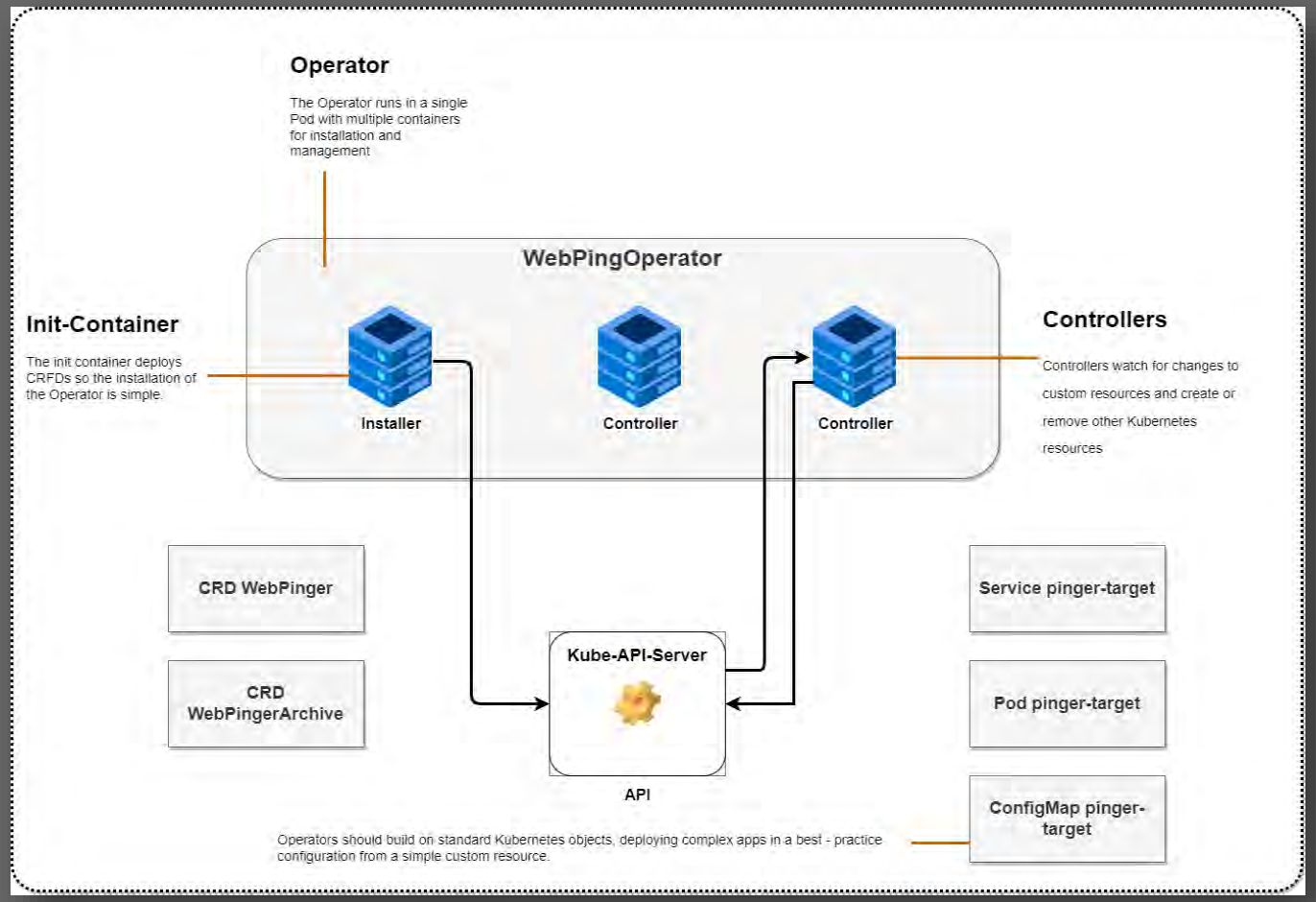
ISO GRUPPE

Custom Resource Definitions (CRDs)...



ISO GRUPPE

Custom Resource Definitions (CRDs)...



ISO GRUPPE

Custom Resource Definitions (CRDs)...

Benefits

Native Kubernetes Integration

Declarative configuration

ISO GRUPPE

Kubernetes Auth Backend Benefits

Benefits

Scalability

Extensibility

Real World Scenarios

Real world scenarios

Storing app secrets

Integrating with Kubernetes service account

Centralized Secret Store

Implement RBAC for Kubernetes Secrets

ISO GRUPPE

Real world scenarios

Providing secrets to legacy apps

Managing SSH keys for Kubernetes clusters

Integrating with other security and compliance tools

ISO GRUPPE

Demos

Enable the kv-v2 Secret Engine

```
$ vault secrets enable -path=internal kv-v2
Success! Enabled the kv-v2 secrets engine at: internal/
```

Create a Secret at a specific path

```
$ vault kv put internal/database/config username=„db-readonly-username“ password=„db-secret-password“
```

```
======= Metadata =======
Key                 Value
---                 -----
created_time        2023-03-19T17:55:25.05842502Z
custom_metadata     <nil>
deletion_time       n/a
destroyed           false
version             1
/ $
```

Verify the secret

```
$ vault kv get internal/database/config
```

```
...

====== Data ======
Key          Value
---          -----
password     db-secret-password
username     db-readonly-username
/ $
```

Configure Kubernetes Authentication

```
$ vault auth enable kubernetes
Success! Enabled kubernetes auth method at: kubernetes/
```

```
$ vault write auth/kubernetes/config \
      kubernetes_host=„https://$KUBERNETES_PORT_443_TCP_ADDR:443“
Success! Data writen to: auth/kubernetes/config
```

Creating Policy with Read Capabilities

```
$ vault policy write internal-app - <<EOF
  path „internal/data/database/config“ {
    capabilities = [„read“]
  }
  EOF

Success! Uploaded policy: internal-app
```

Creating a Kubernetes Authentication Role

```
$ vault write auth/kubernetes/role/internal-app \
    bound_service_account_names=internal-app \
    bound_service_account_namespaces=default \
    policies=internal-app \
    ttl=24h

Success! Data written to: auth/kubernetes/role/internal-app
```

Creating a Kubernetes Service Account

```
$ kubectl create sa internal-app
```

```
$ kubectl get serviceaccounts
```

```
NAME                 SECRETS   AGE
default              0         81m
internal-app         0         43s
vault                0         53m
vault-agent-injector 0         53m
```

Deploy a demo app

```
deployment-orgchart.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: orgchart
  labels:
    app: orgchart
spec:
  selector:
    matchLabels:
      app: orgchart
  replicas: 1
  template:
    metadata:
      annotations:
      labels:
        app: orgchart
    spec:
      serviceAccountName: internal-app
      containers:
        - name: orgchart
          image: jweissig/app:0.0.1
```

Deploy a demo app

```
$ kubectl get pods
```

```
NAME                                    READY   STATUS    RESTARTS   AGE
orgchart-f945d469-klzwj                 1/1     Running   0          44s
vault-0                                 1/1     Running   0          69m
vault-agent-injector-7dcd577577-mnqsx   1/1     Running   0          69m
```

Inject Secrets into the pod

```
$ kubectl cat patch-inject-secrets.yaml
```

```
spec:
  template:
    metadata:
      annotations:
        vault.hashicorp.com/agent-inject: "true"
        vault.hashicorp.com/role: "internal-app"
        vault.hashicorp.com/agent-inject-secret-database-config.txt:
        "internal/data/database/config"
```

Inject Secrets into the pod

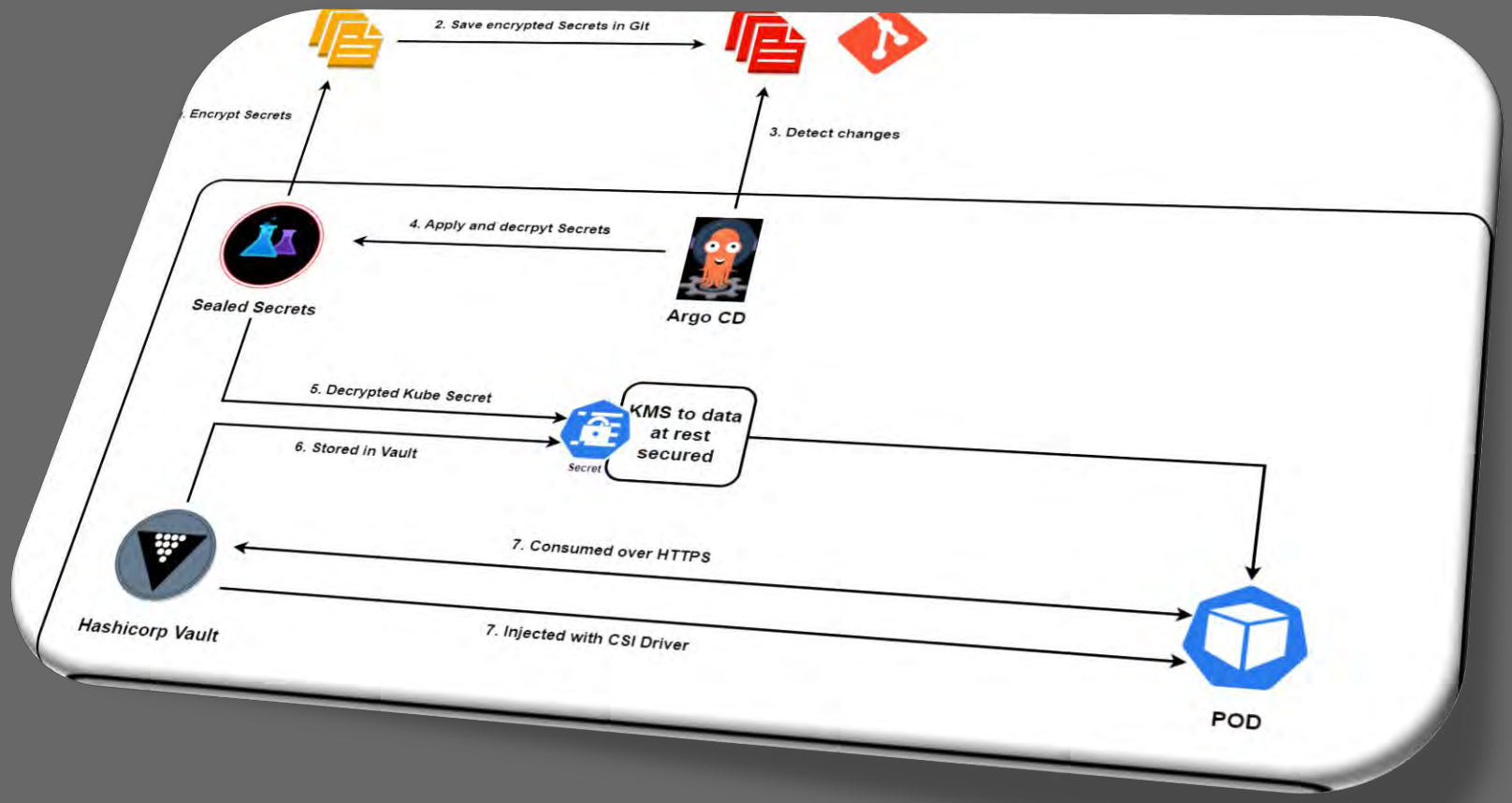
```
$kubectl patch deployment orgchart --patch "$(cat patch-inject-secrets.yaml)"
```

Verify the secret

```
$ kubectl exec $(kubectl get pod -l app=orgchart -o
jsonpath="{.items[0].metadata.name}") --container orgchart -- cat
/vault/secrets/database-config.txt
```

```
data: map[password:db-secret-password username:db-readonly-username]
metadata: map[created_time:2023-03-19T17:55:25.05842502Z custom_metadata:<nil>
deletion_time: destroyed:false version:1]
```

Usage in CI/CD context



ISO GRUPPE

Wrap-Up


THANK YOU

References

- What is Vault?: https://developer.hashicorp.com/vault/docs/what-is-vault
- Hashicorp Vault Architecture: https://developer.hashicorp.com/vault/docs/internals/architecture
- Kubernetes Secrets Management (Manning, ISBN 9781617298912)
- Whats HCP Vault: https://developer.hashicorp.com/vault/tutorials/cloud/vault-introduction
- CRDs: https://k21academy.com/docker-kubernetes/k8s-custom-resource-definition/
- CRDs, Operators, Custom Controllers: https://livebook.manning.com/book/learn-kubernetes-in-a-month-of-lunches/chapter-20/v-7/142

References

- Sample CRDs for a knative operator: https://github.com/knative/operator/tree/main/config/crd/bases
- Make custom resources of Kubernetes operator user-friendly: https://developer.ibm.com/articles/make-complex-custom-resources-of-kubernetes-operators-user-friendly/
- Apimatic codegen kubernetes operator: https://docs.apimatic.io/changelog/added-new-feature-apimatic-codegen-kubernetes-operator/
- Kubernetes operators – Automated Lifecycle Management: https://builders.intel.com/docs/networkbuilders/kubernetes-operators-automated-lifecycle-management-technology-guide.pdf
- Kubernetes operators automated lifecycle management technology guide: https://builders.intel.com/docs/networkbuilders/kubernetes-operators-automated-lifecycle-management-technology-guide.pdf
- Extend Kubernetes Api with Custom Resource Definitions: https://kubernetes.io/docs/tasks/extend-kubernetes/custom-resources/custom-resource-definitions/
- Vault Kubernetes raft deployment guide: https://developer.hashicorp.com/vault/tutorials/kubernetes/kubernetes-raft-deployment-guide
- Vault Kubernetes sidecar tutorial: https://developer.hashicorp.com/vault/tutorials/kubernetes/kubernetes-sidecar

ISO GRUPPE

Ressources

- Question question mark:https://pixabay.com/illustrations/question-question-mark-response-1015308/
- Thank you: https://unsplash.com/photos/pnGjbJEmU3o/download?ixid=MnwxMjA3fDB8MXxhbGx8fHx8fHx8fHwxNjc5MjEwNzUy&force=true&w=1920
- Old black background:https://www.freepik.com/free-photo/old-black-background-grunge-texture-dark-wallpaper-blackboard-chalkboard-room-wall_11712558.htm#query=solid%20background&position=1&from_view=keyword&track=ais
- Abstract secure technology background: https://www.freepik.com/free-vector/abstract-secure-technology-background_5850994.htm#query=security%20background&position=0&from_view=keyword&track=ais
- Tiny people carrying key to open padlock: https://www.freepik.com/free-vector/tiny-people-carrying-key-open-padlock_13683703.htm#query=password&position=3&from_view=search&track=sph
- Cross platform development: https://www.freepik.com/free-vector/cross-platform-development-abstract-concept-illustration_11667604.htm#query=app%20config&position=0&from_view=search&track=ais

Ressources

- Login, register: https://pixabay.com/illustrations/login-register-window-button-4387708/
- Container-ship: https://pixabay.com/photos/container-ship-container-transport-6631117/
- Safe Vault: https://pixabay.com/vectors/safe-vault-lock-metal-money-33270/
- Blockchain Handshake: https://pixabay.com/photos/blockchain-handshake-shaking-hands-2853046/
- App software: https://pixabay.com/illustrations/app-software-contour-settings-1013616/
- Vpn address: https://pixabay.com/illustrations/vpn-address-anonymous-security-4046047/
- Key house: https://pixabay.com/illustrations/key-house-house-keys-home-estate-2114455/