Vault Deployment Architecture for Vault Associate (003)

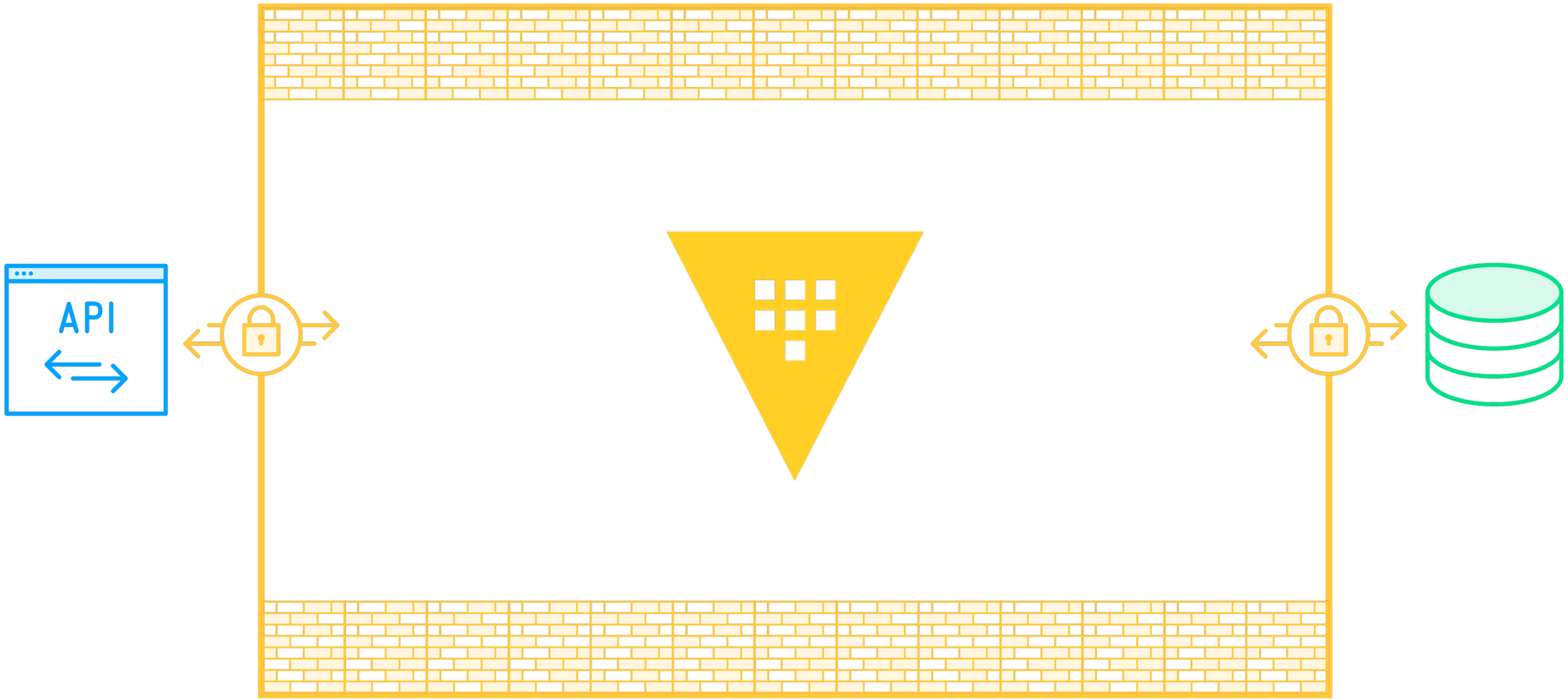
Vault Physical Architecure

Ned Bellavance

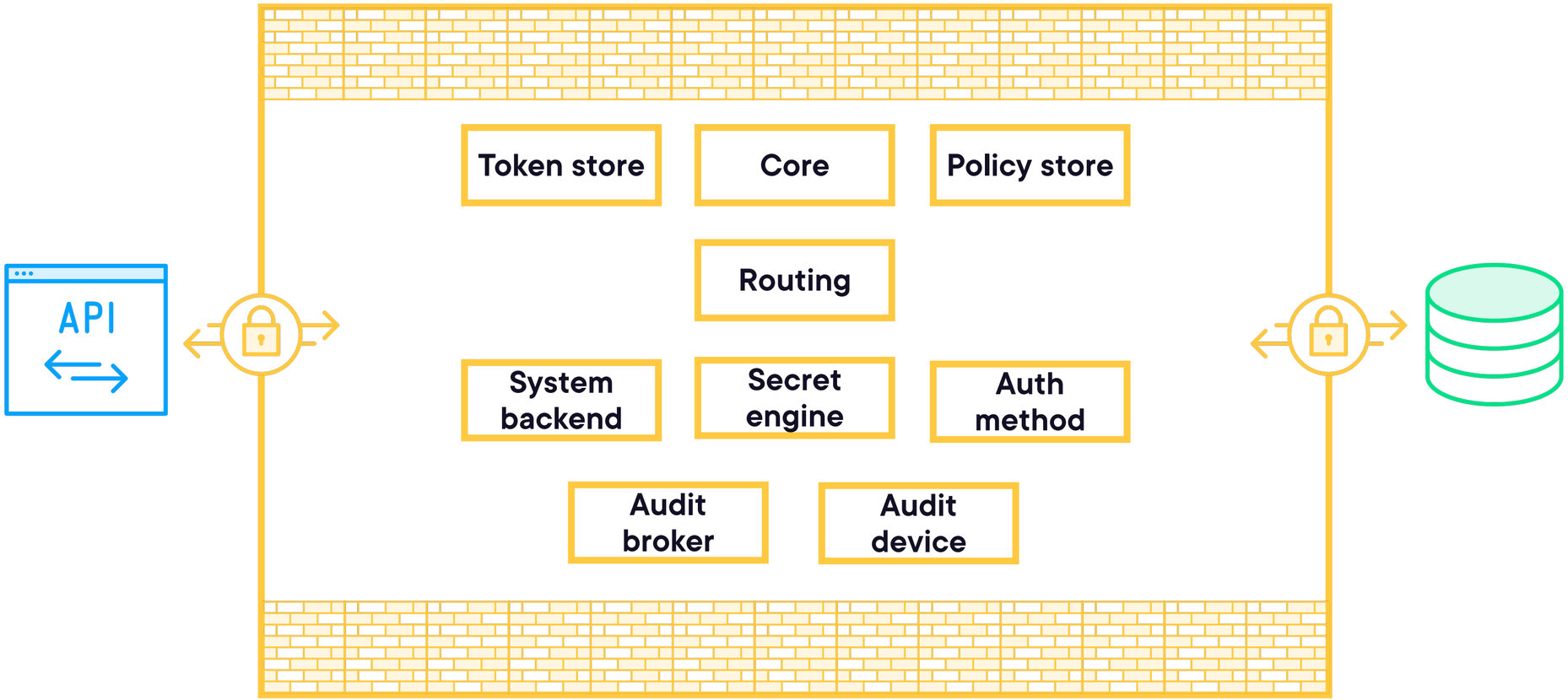
HashiCorp Certified Instructor

@nedinthecloud | nedinthecloud.com

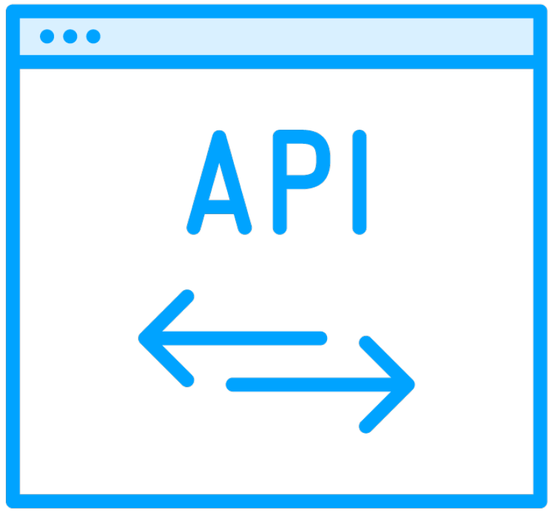
Vault Architecture



Vault Architecture



API Frontend



Gateway for all client communication

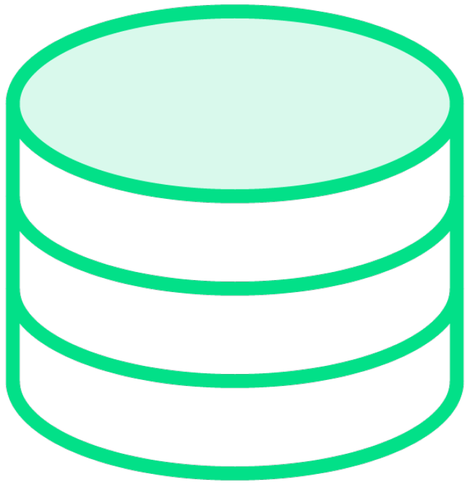
- CLI and UI

Authentication methods

ACL policies

TLS certificates

Storage Backend

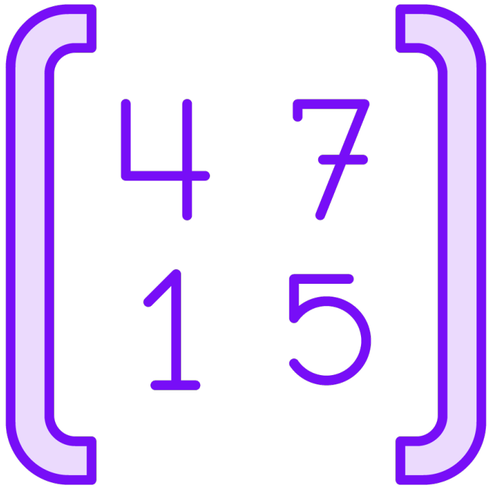


Data encrypted before leaving barrier

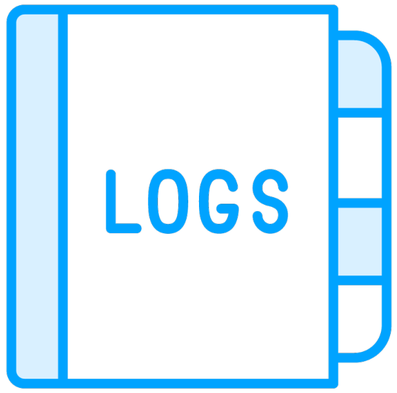
Encryption keys protected by root key

Root key protected by unseal key

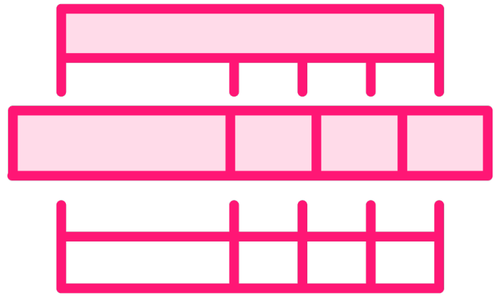
Other Vault Objects



Metrics



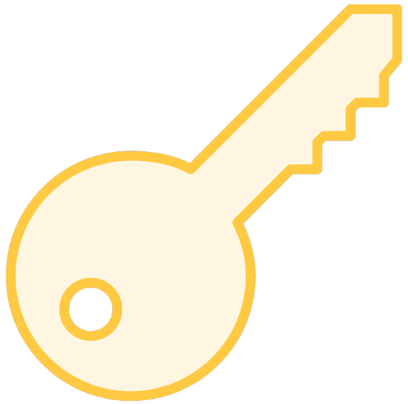
Logs



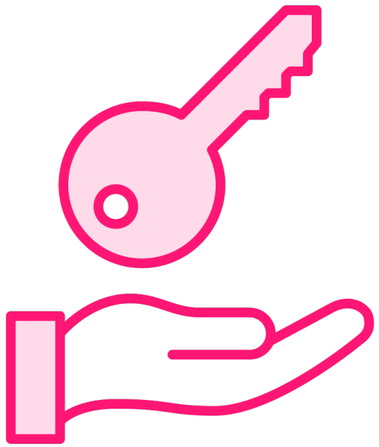
Audit devices

Encryption Keys

Encryption keys
Protect data written to storage
Stored on disk

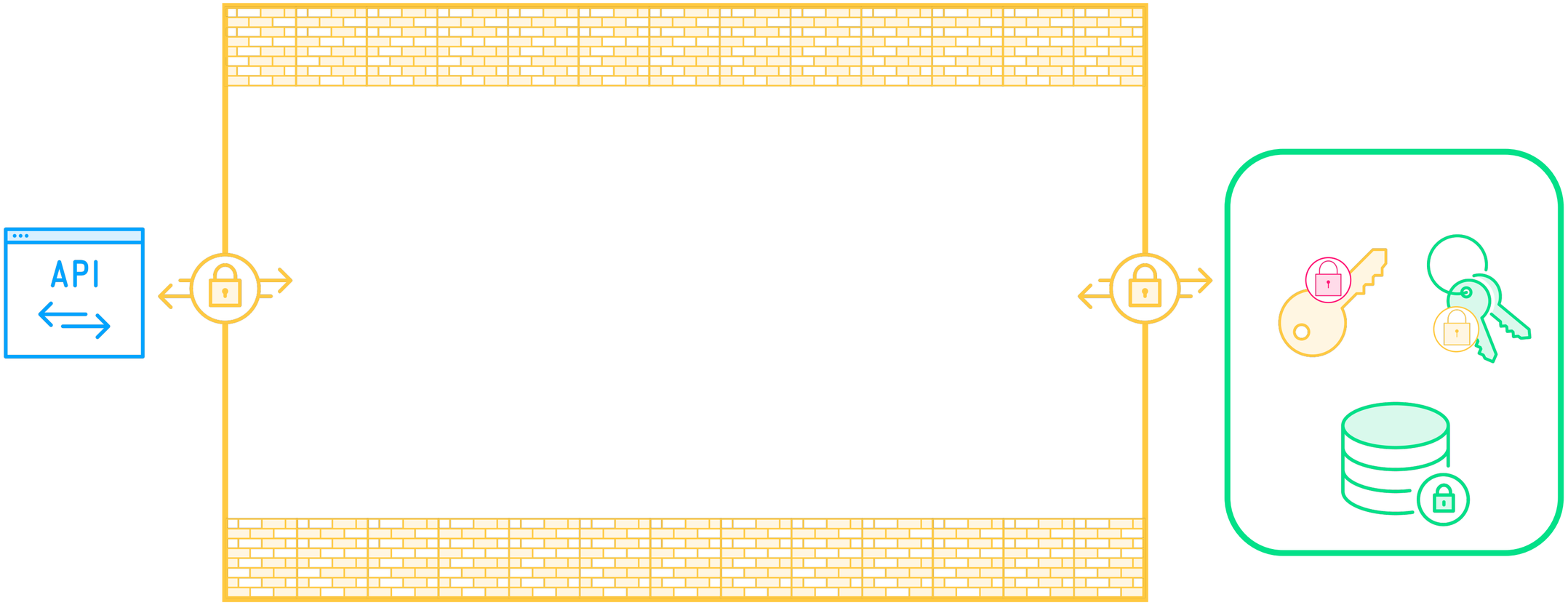


Root key
Protects encryption keys
Stored on disk

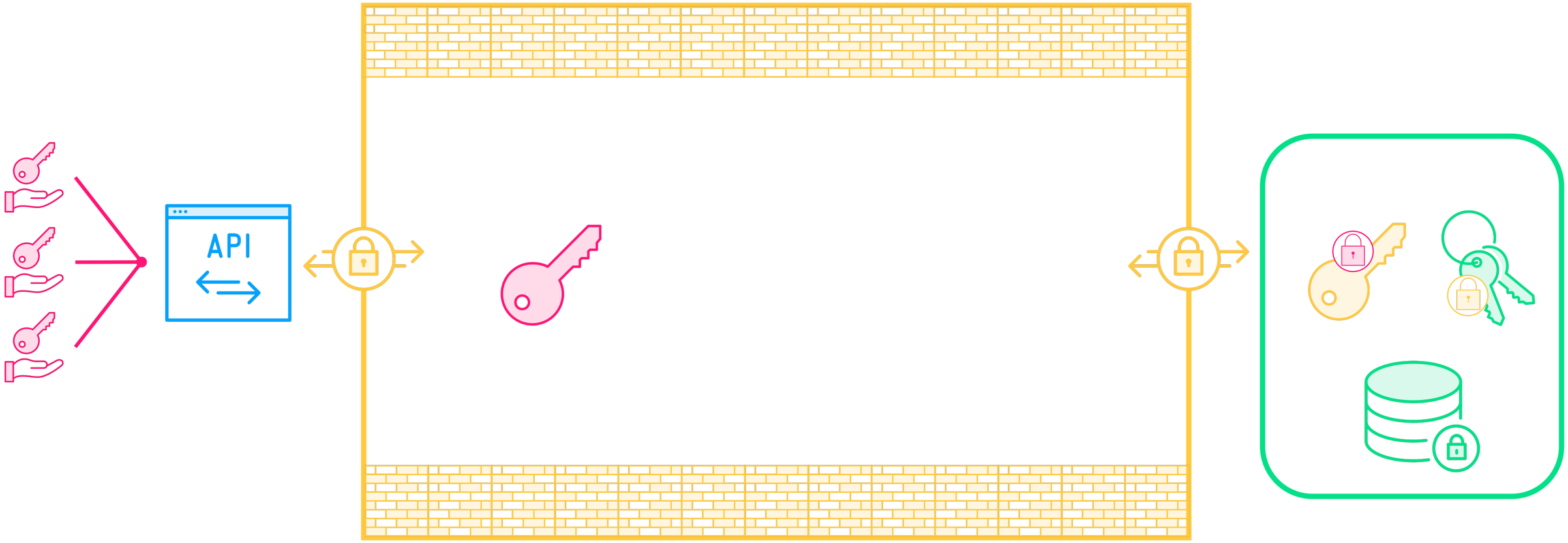


Unseal key
Protects root key
Stored externally

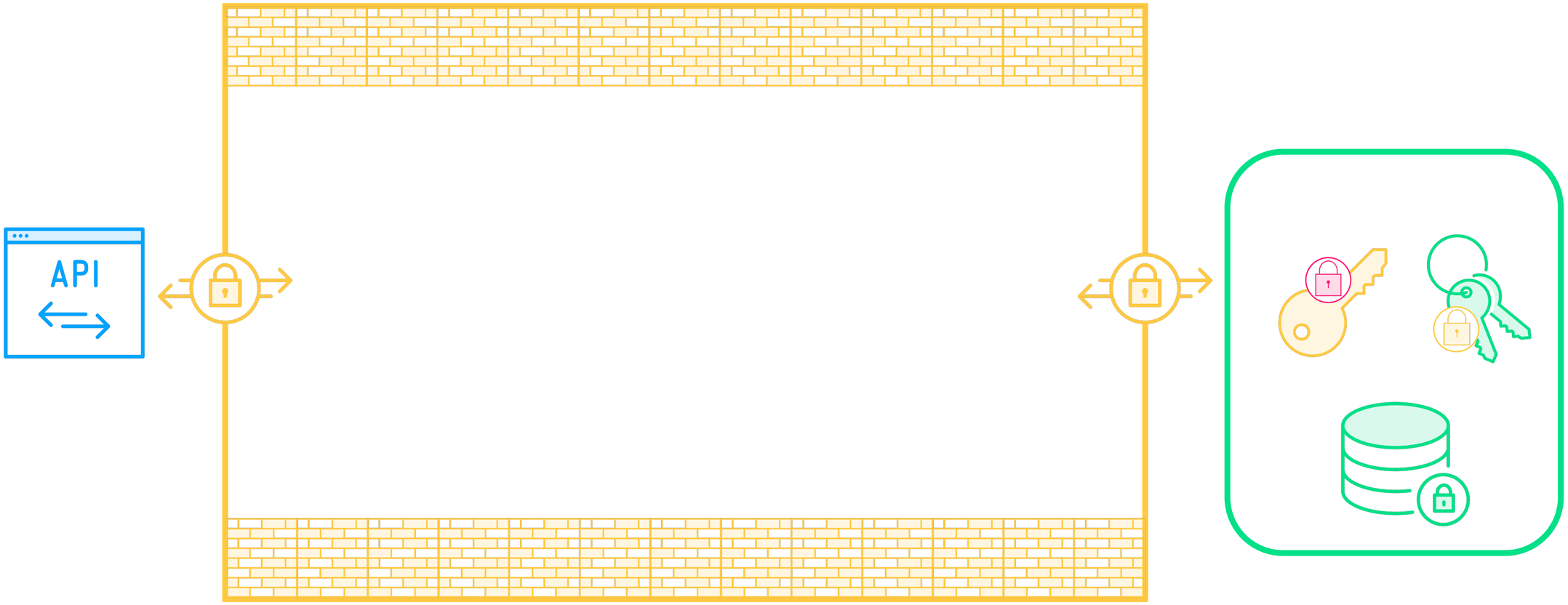
Unsealing Vault



Unsealing Vault



Unsealing Vault



Storage Backend Types

Object

Database

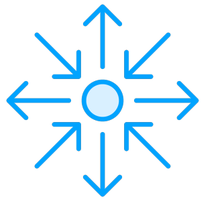
Key/value

Local file

In-memory

Storage Decision Points

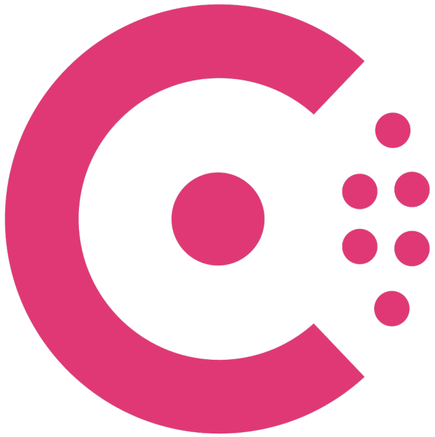
Support - HashiCorp or community



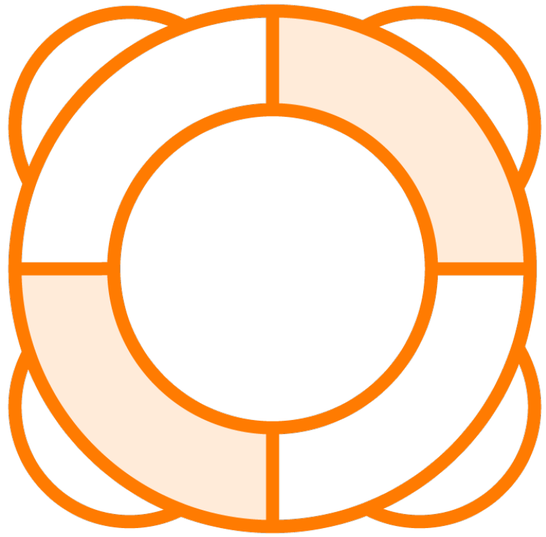
High availability support

Level of access and comfort

HashiCorp Supported Storage

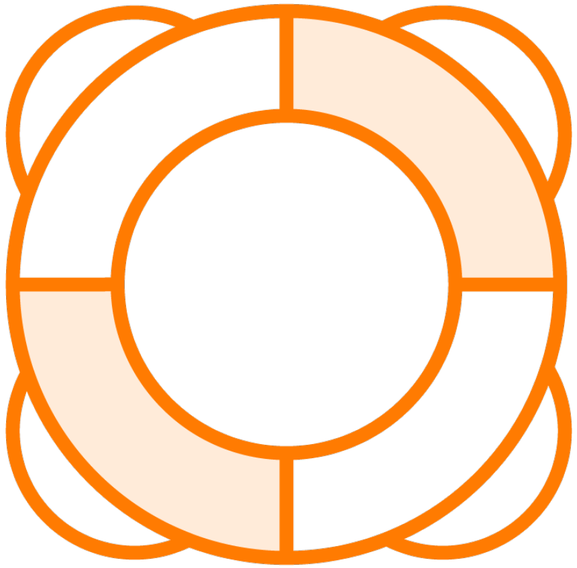


Consul



Integrated storage
(Raft)

Integrated Storage



Uses local storage

Raft consensus protocol replication

Key benefits

- Data location
- Operational simplicity
- Cluster HA
- HashiCorp support
- Troubleshooting

Preferred option

Vault Server Configuration

HCL or JSON format

Multiple files supported

Passed with one or more –config flags

Loaded during startup only

Configuration Parameters

General

High availability

Listener

Storage

Seal

Telemetry

vault.hcl

```
ui = [ true | false ]

disable_mlock = [ true | false ]

log_level = [ "error" | "warn" | "info" | "debug" | "trace" ]

log_file = "<absolute file path>"

log_format = [ "standard" | "json" ]
```

What's up with mlock?

Prevents memory swap to disk

Defaults to false (enabling mlock)

Set to true for integrated storage

Disable or encrypt swap storage

disable_mlock

vault.hcl

```
ui = [ true | false ]

disable_mlock = [ true | false ]

log_level = [ "error" | "warn" | "info" | "debug" | "trace" ]

log_file = "<absolute file path>"

log_format = [ "standard" | "json" ]
```

Listener Stanza

```
listener "tcp" {
  address       = "0.0.0.0:8200"
  tls_cert_file = "path/to/public/cert.crt"
  tls_key_file  = "path/to/private/cert.key"
}
```

Storage Stanza

```
storage "file" {
  path = "/vault/data"
}

storage "raft" {
  path = "/vault/data"
  retry_join {
    leader_api_addr     = "https://leader_ip_address:8200"
    leader_ca_cert_file = "/path/to/ca/cert.pem"
  }
}
```

Seal Stanza

```
seal "azurekeyvault" {
  tenant_id     = "55555-6666-77778888"
  client_id     = "11111-2222-33334444"
  client_secret = "super_secret_password"
  vault_name    = "prod-vault-name"
  key_name      = "prod-key-name"
}
```

Telemetry Stanza

```
telemetry {
  statsite_address        = "statsite.server.local:8125"
  prefix_filter           = [ "+vault.token", "-vault.expire" ]
  enable_hostname_label = true
  metrix_prefix           = "vault-prod"
}
```

Environment Variables

VAULT_UI – Enable the UI

VAULT_LOG_LEVEL – Set log level

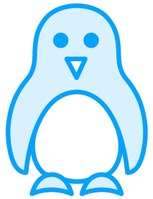
VAULT_LICENSE_PATH – Enterprise license

VAULT_RAFT_PATH – Data path on filesystem

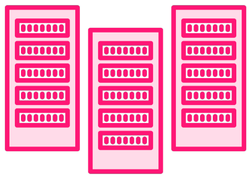
Seal Stanza

```
seal "transit" {
  address    = "https://transit.vault.local:8200"
  key_name   = "transit_key_name"
  mount_path = "transit/"
  token      = "env://TRANSIT_TOKEN"
}
```

Deployment Options



Operating system

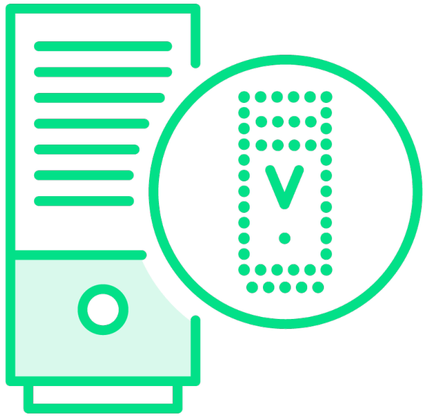


Compute platforms

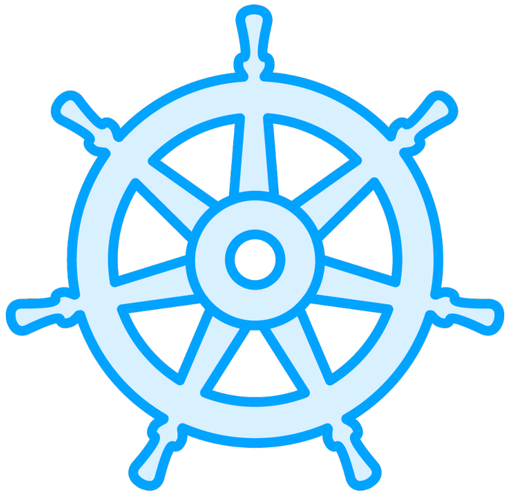
HCP Vault Dedicated

Compute Platforms

Bare metal
Most secure
High cost
Admin overhead

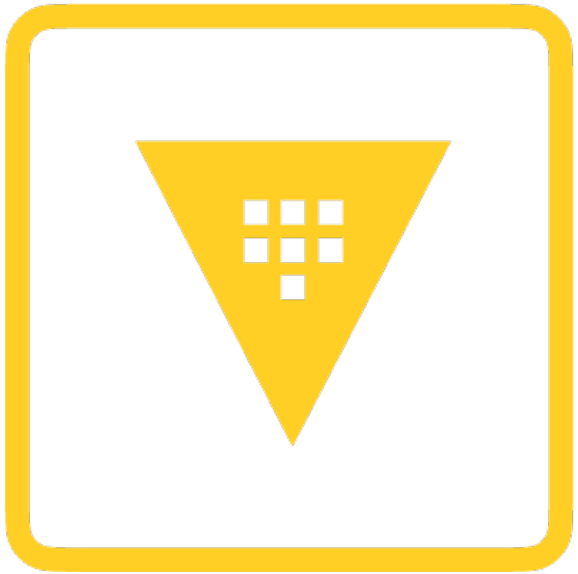


Virtualization
Good isolation
Simple deployment
Reduced overhead



Container
Isolation concerns
Ephemeral nature
Detailed guide

HCP Vault Dedicated



HashiCorp hosts and manages Vault

Deploy on supported cloud provider

- AWS or Azure

Dynamically scalable

Includes enterprise features

Creating a Vault Server

Create a Vault server using integrated storage, Shamir seal, and Docker