Secret Management with Vault (and more)

Brian Nuszkowski

- Philosophy
- Architecture
- Components
- Implementation and Scale

“don't put your eggs in one basket”

“a bird in the hand is worth two in the bush”

“not perfect, but better”

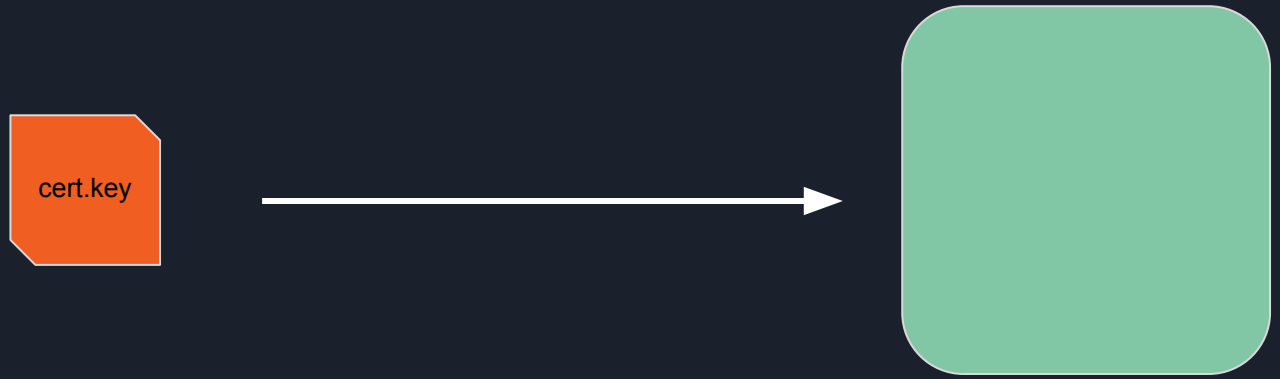
Philosophy

Problem Statement #1: Managing Secrets

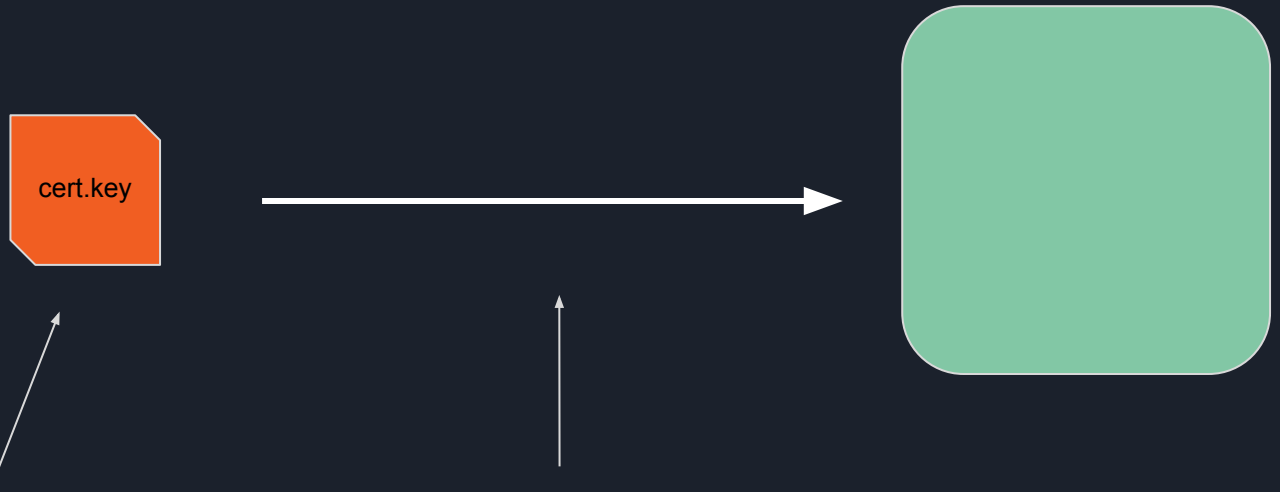
Move something,
from someplace

To
Somewhere

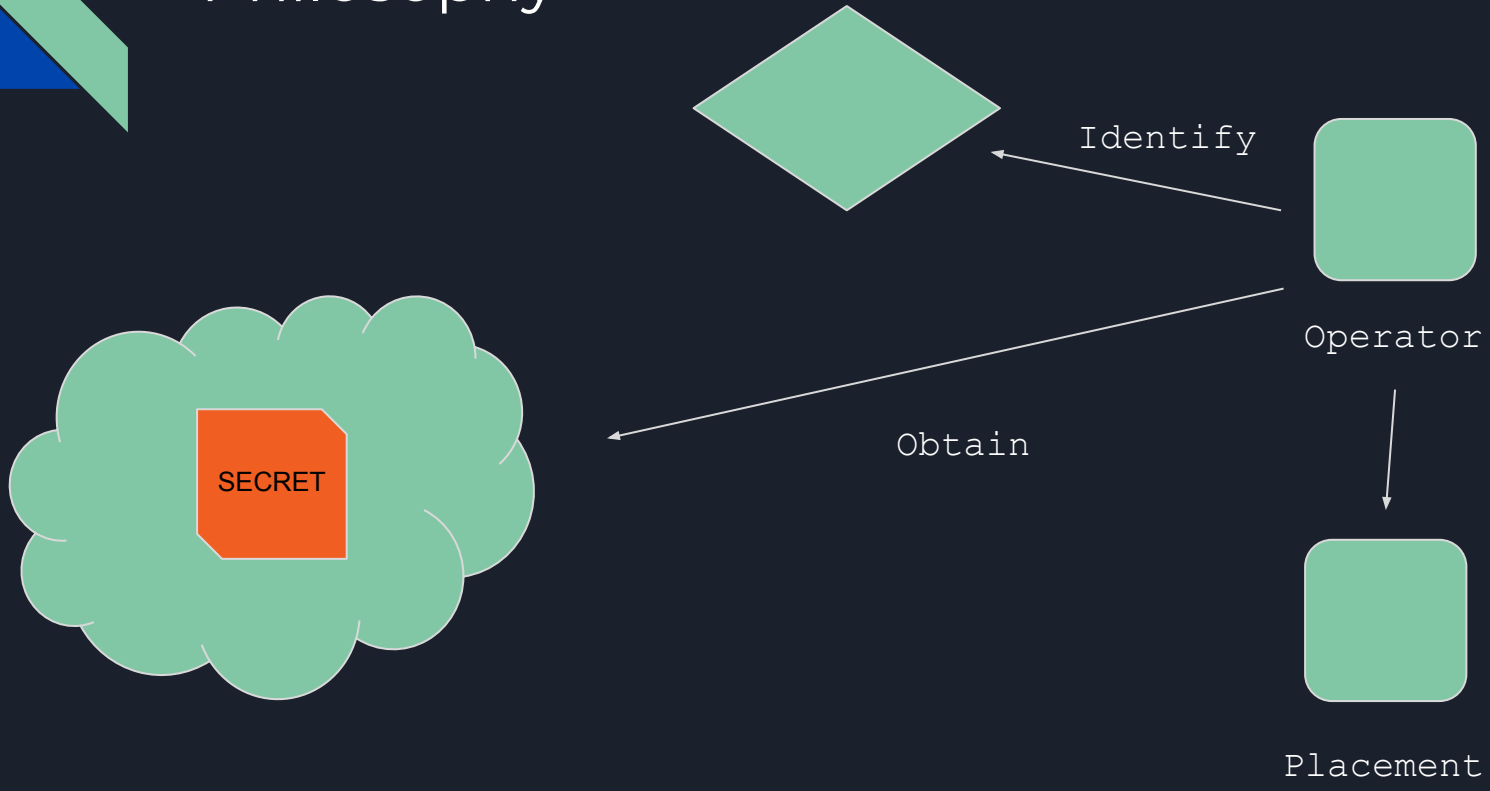
Operation: Move The Keys



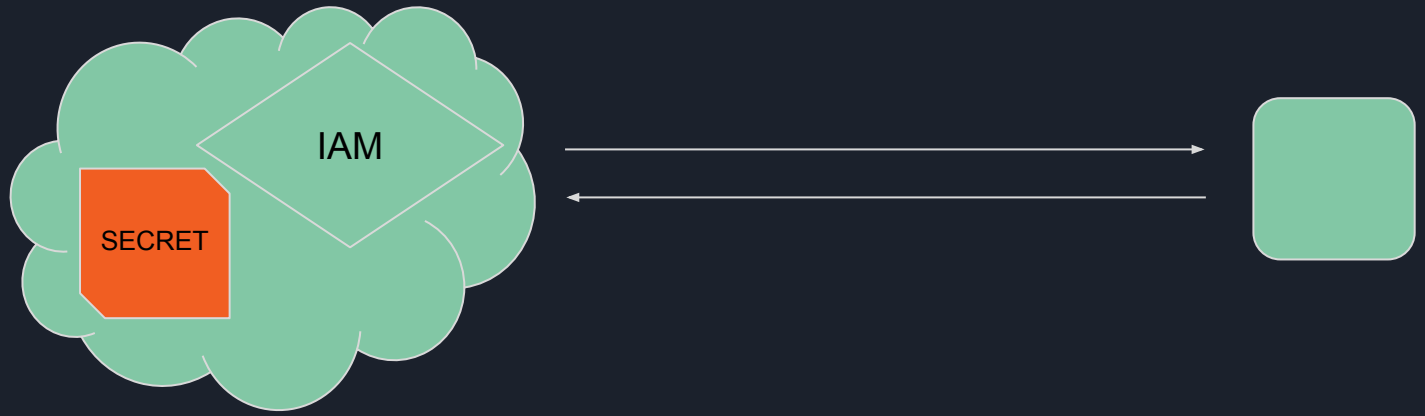
Operation: Move The Keys



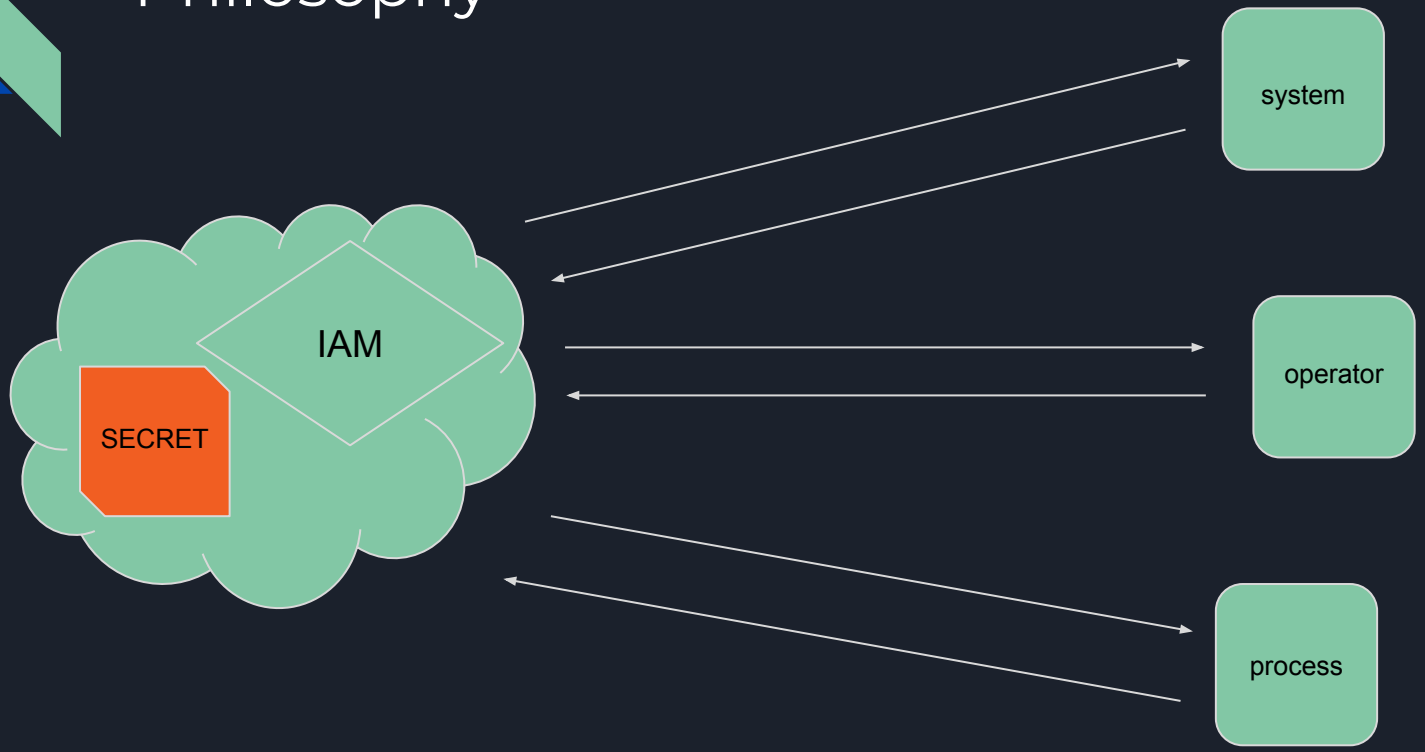
Philosophy



Philosophy



Philosophy

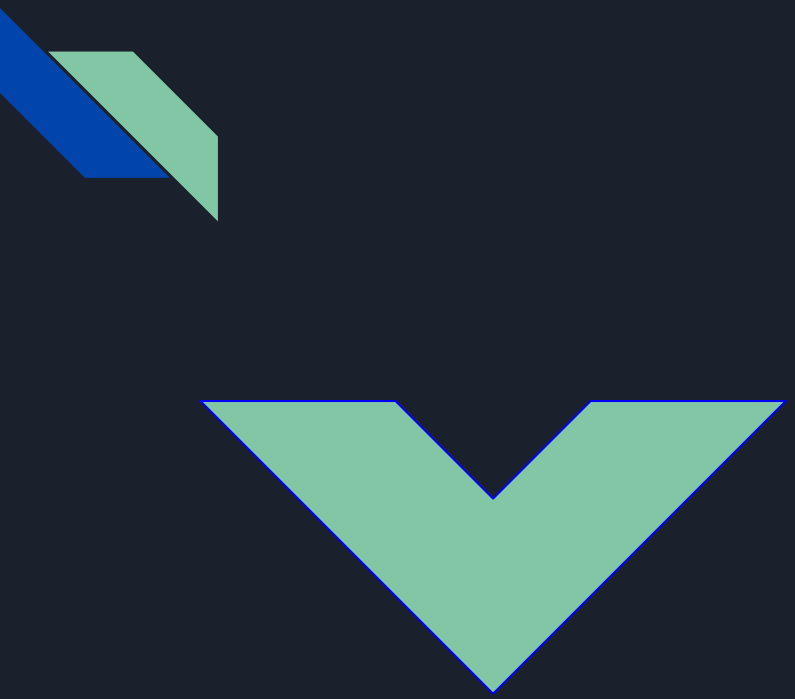


Recap

Problem Statement #1: Managing Secrets

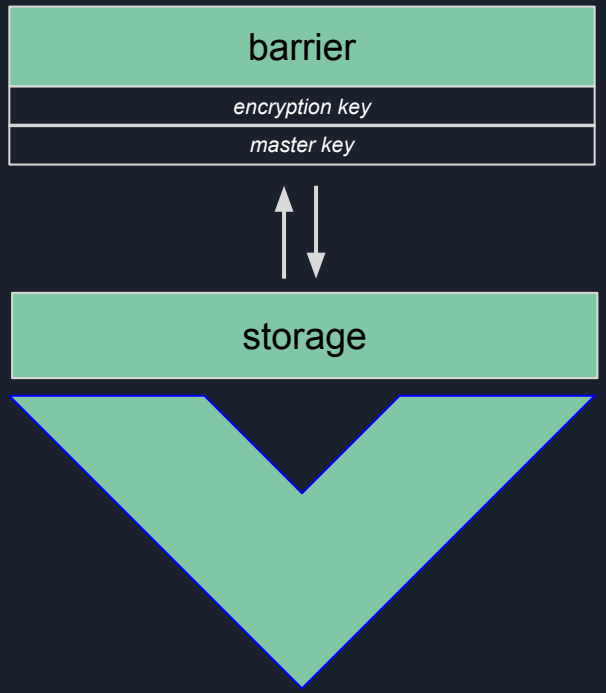
Problem Statement #2: Secure Operations can be Complicated

Architecture



- Written in Golang
- Single binary for client and server
  - Other clients available
- HTTP API Interface
- Interaction Languages
  - JSON
  - HCL (HashiCorp Configuration Language)


plugin
audit
secret
auth + policies
barrier
storage*
configuration


barrier
encryption key
master key
storage

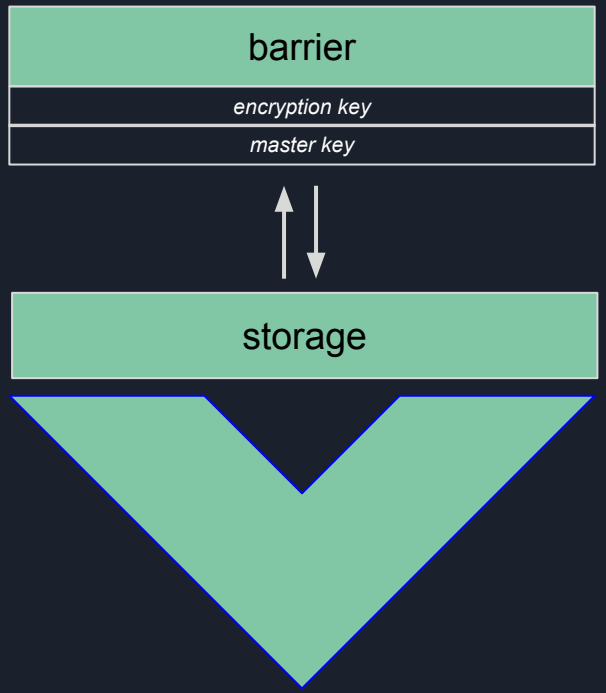
Master Key → Encryption Key → Data

key encryption key (KEK)

data encryption key (DEK)

Barrier Sealed State

Barrier Unsealed State


barrier
encryption key
master key
storage

Master Key

c2fc4cc7f02da94cd1d621921dff56dc2485407cd0c39b5e5c7422f7d806fe8f

Problem Statement Infinity: Managing Master Key

Move something,
from someplace

To
Somewhere

Shamir’s Secret Sharing

c2fc4cc7f02da94cd1d621921dff56dc2485407cd0c39b5e5c7422f7d806fe8f

Unseal keys

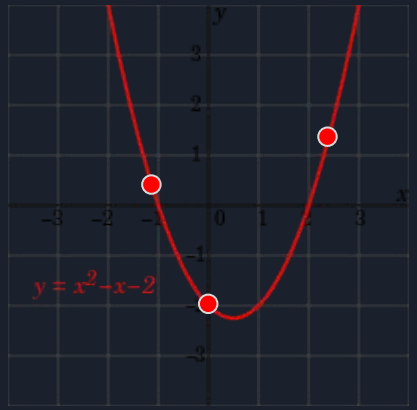
5d9be2c763a2e90ecdd4ccfc364d0ab4

37c0fcbda5a4dc65b439da18a43ce7fd

0be3021b7874b2e616504e20fa842330

019aa865d9b8449b7fc3bb64b6341bd0

ed8b816a75c88c55a75f21b5bb456cc9



m of n

Key Notes

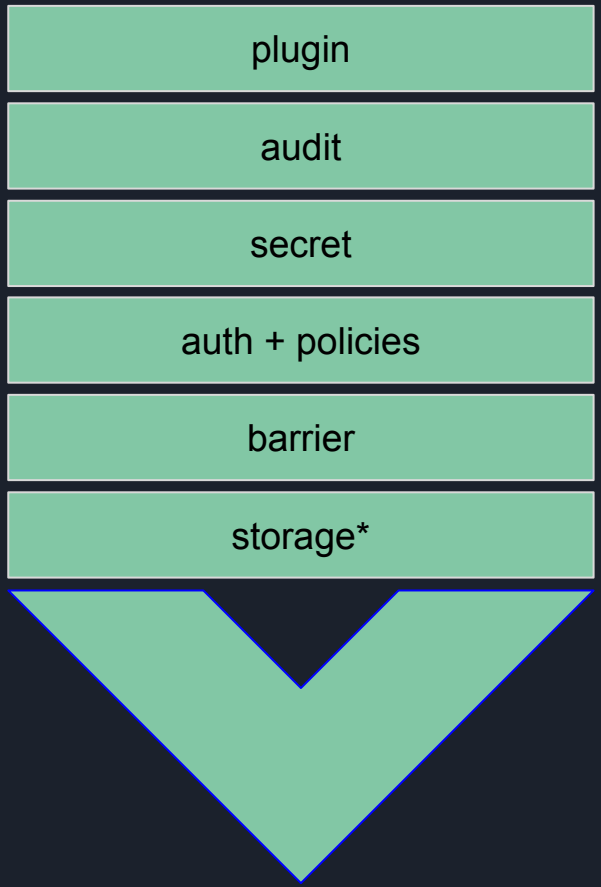
Unseal Philosophy

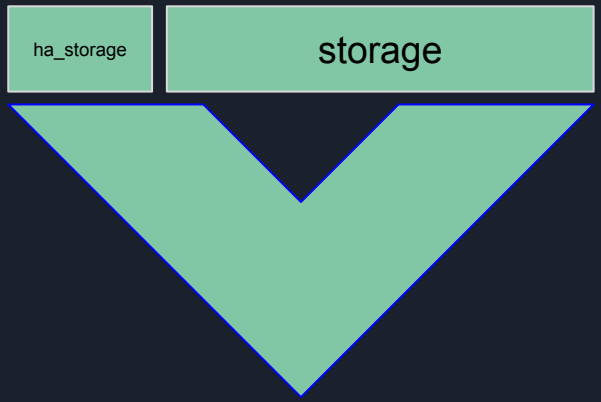
- Manually
- Sidecar
- Auto Unseal*
- HSM**

Maintenance

- Rekey
- Rotation

Components


plugin
audit
secret
auth + policies
barrier
storage*


ha_storage
storage

Storage Backends

- All storage backends store data
- Not all storage backends support HA
- Mix-n-Match
- Community Supported

Backendends*
Azure
CockroachDB
Consul
CouchDB
DynamoDB
Etcd
Filesystem
Google Cloud Storage
Google Cloud Spanner
In-Memory
Manta
MySQL
PostgreSQL
Cassandra
S3
Swift
Zookeeper
Write Your Own!

*Subject to change

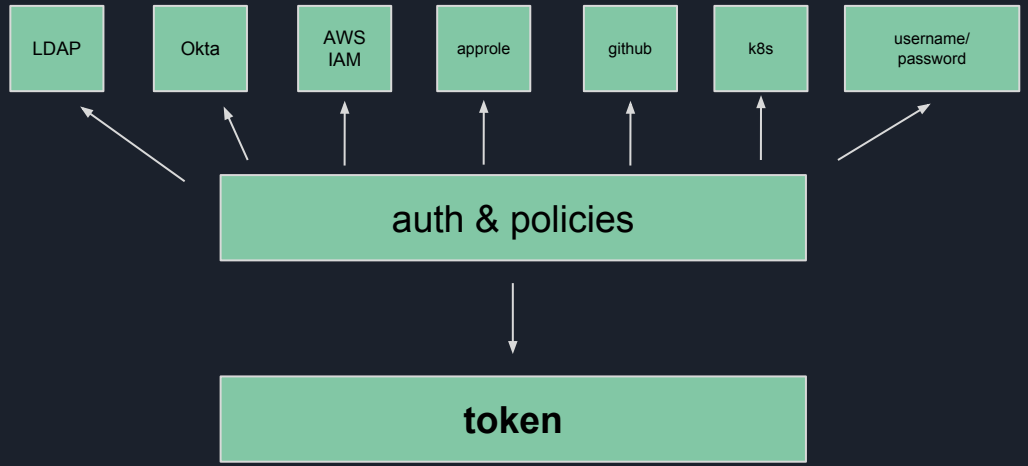
Storage Backends

```
storage "mysql" {
  address = "rds-thingy-us-east-2.aws.computer.net"
  username = "hugo"
  password = "stiglitz"
  database = "vault"
}

ha_storage "dynamodb" {
  ha_enabled = "true"
  region     = "us-west-2"
  table      = "vault-data"
}
```

Auth Backends

Auth Backends




```
X-Vault-Token: <YOURTOKEN>
```

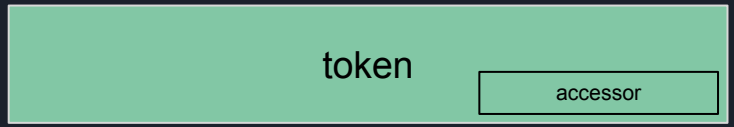
Auth Backends - Token

“The token is the back on which all other authentication is built”

```
Key                Value
---                -----
accessor           031b86b6-479c-6d60-c259-adfe3cde7579
creation_time      1505450722
creation_ttl       300
display_name       token-cool-token
expire_time        2017-09-15T00:50:22.010528189-04:00
explicit_max_ttl   0
id                 b9c958e9-0087-14f1-99b0-07710841f7f9
issue_time         2017-09-15T00:45:22.010527976-04:00
meta               map[department:computer]
num_uses           0
orphan             false
path               auth/token/create
policies           [default the-ting-go]
renewable          true
ttl                276
```

Special tokens:

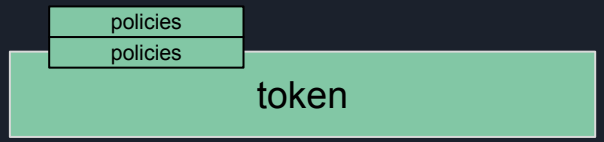
- Root
- Periodic
- Orphan
  - Parent
    - Child1
    - Child2
      - GrandChild
    - Child3
- Lease, Renew, and Revoke
- TTL
  - System
  - Mount
  - At-creation
  - Explicit max



Auth Delegation Implementations + Implications

policies

Attached to the user token as explicitly defined or via group membership



policies

- default
- root
- HCL or JSON

```
path "secret/*" {
  capabilities = ["create"]
}

path "secret/foo" {
  capabilities = ["read"]
}

path "auth/token/lookup-self" {
  capabilities = ["read"]
}
```

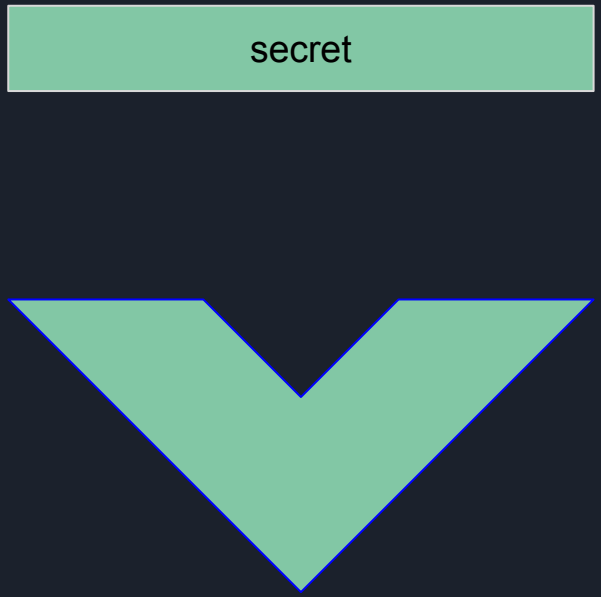
policies

- create (POST/PUT)
- read (GET)
- update (POST/PUT)
- delete (DELETE) - Allows deleting the data at the given path.
- list (LIST)
- sudo
- deny

policies

```
path "secret/thing" {
  capabilities = ["create"]
  allowed_parameters = {
    "bar" = ["zip", "zap"]
    "whatever" = []
  }
}
```

Backends



Secret Engines

AWS
Consul
Cubbyhole
Databases
Google Cloud
Key/Value
Identity
Nomad
PKI (Certificates)
RabbitMQ
SSH
TOTP
Transit

Secret Engine - Key/Value

kv/ (aka secret/)

```
› vault write secret/something username=whatever password=thing
Success! Data written to: secret/something

› vault read secret/something
Key                 Value
---                 -----
refresh_interval    768h0m0s
password            thing
username            whatever

 › vault write secret/something password=newpw
Success! Data written to: secret/something

 › vault read secret/something
Key                 Value
---                 -----
refresh_interval    768h0m0s
password            newpw
```

Secret Engine - PKI

Old Way:

- Create Private Key
- Create CSR
- Submit CSR
- WAIT

New Way:

- `vault write pki/issue/standard common_name=svc.company.com`
- Good for End Users
- Even Better for Systems

Secret Engines

SSH

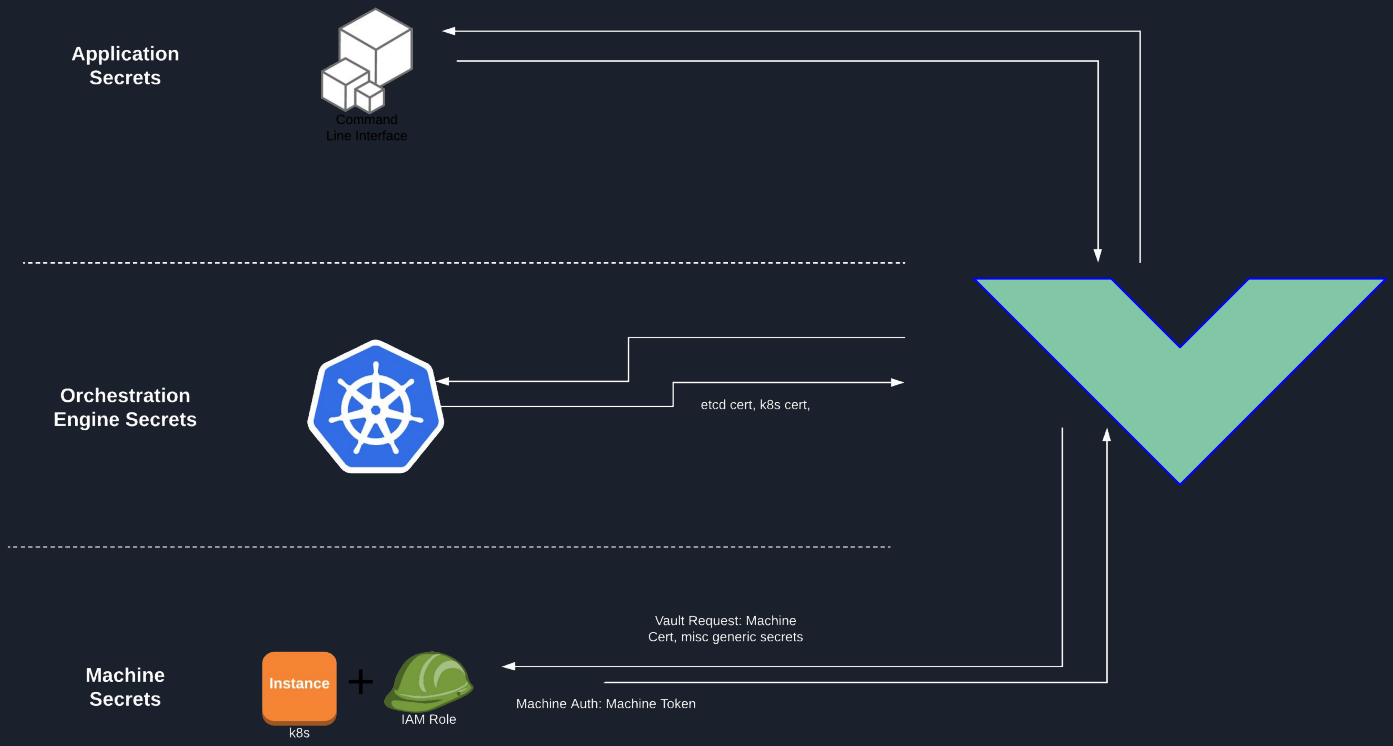
- Signed SSH Certificates
- One Time SSH Passwords

TOTP

```
❯ vault read totp/code/brian
Key     Value
---     -----
code    666989
```

Transit

- Encrypt/Decrypt
- Random Byte Generator
- Hashing
- HMAC
- Signatures


Application Secrets
Command Line Interface
Orchestration Engine Secrets
etcd cert, k8s cert,
Machine Secrets
Instance
k8s
IAM Role
Vault Request: Machine Cert, misc generic secrets
Machine Auth: Machine Token

Backends

audit

Backend - Audit


```
{"time":"2018-04-25T18:43:32.6798613Z","type":"request","auth":{"client_token":"hmac-sha256:0960371a
906ddf61e89d4d9246259ee9d4dac0f18880f3ecceada5e4184b7a44","accessor":"hmac-sha256:62c0c468c8fc68fa9d
79f55ebb2dee740c6090c9435acb89fd81778aee8f1385","display_name":"root","policies":["root"],"metadata"
:null,"entity_id":""},"request":{"id":"d7efb81a-9457-0f19-848a-fef3a210c799","operation":"update","c
lient_token":"hmac-sha256:0960371a906ddf61e89d4d9246259ee9d4dac0f18880f3ecceada5e4184b7a44","client_
token_accessor":"hmac-sha256:62c0c468c8fc68fa9d79f55ebb2dee740c6090c9435acb89fd81778aee8f1385","path
":"secret/data/hello","data":{"data":{"mysecret":"hmac-sha256:189578c2e7314c850f63071e7842f643b1d8c1
b6380c8d5599c9d674ff284997"},"options":{}},"policy_override":false,"remote_address":"127.0.0.1","wra
p_ttl":0,"headers":{}},"error":""}

{"time":"2018-04-25T18:43:32.680390805Z","type":"response","auth":{"client_token":"hmac-sha256:09603
71a906ddf61e89d4d9246259ee9d4dac0f18880f3ecceada5e4184b7a44","accessor":"hmac-sha256:62c0c468c8fc68f
a9d79f55ebb2dee740c6090c9435acb89fd81778aee8f1385","display_name":"root","policies":["root"],"metada
ta":null,"entity_id":""},"request":{"id":"d7efb81a-9457-0f19-848a-fef3a210c799","operation":"update"
,"client_token":"hmac-sha256:0960371a906ddf61e89d4d9246259ee9d4dac0f18880f3ecceada5e4184b7a44","clie
nt_token_accessor":"hmac-sha256:62c0c468c8fc68fa9d79f55ebb2dee740c6090c9435acb89fd81778aee8f1385","p
ath":"secret/data/hello","data":{"data":{"mysecret":"hmac-sha256:189578c2e7314c850f63071e7842f643b1d
8c1b6380c8d5599c9d674ff284997"},"options":{}},"policy_override":false,"remote_address":"127.0.0.1","
wrap_ttl":0,"headers":{}},"response":{"data":{"created_time":"hmac-sha256:4452da60561049f58d83f91796
065a79465efcc8211f34e89a568821a96af0f0","deletion_time":"hmac-sha256:43ea9f7f97a2deeade77fcc2ff3ad7e
a27ad531a6b30a990c99d2861ba9dd289","destroyed":false,"version":2}},"error":""}
```

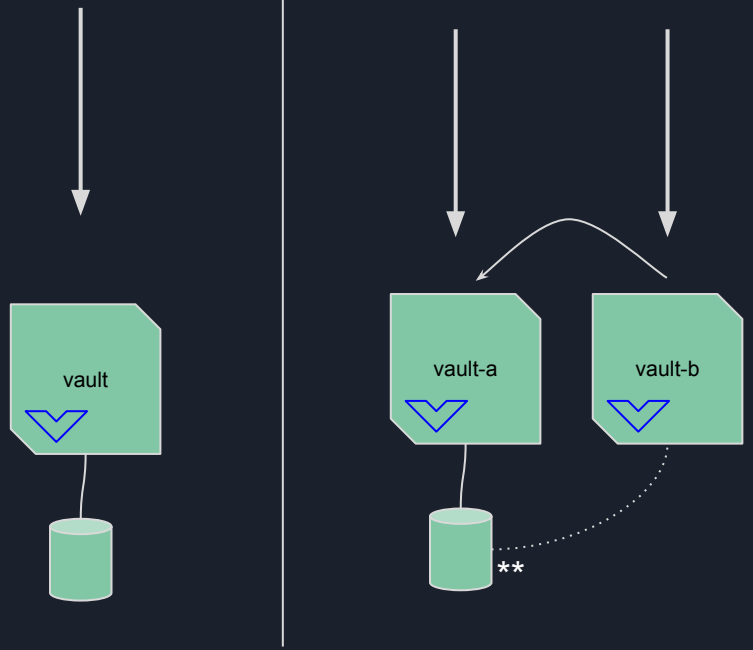
Audit Intelligence

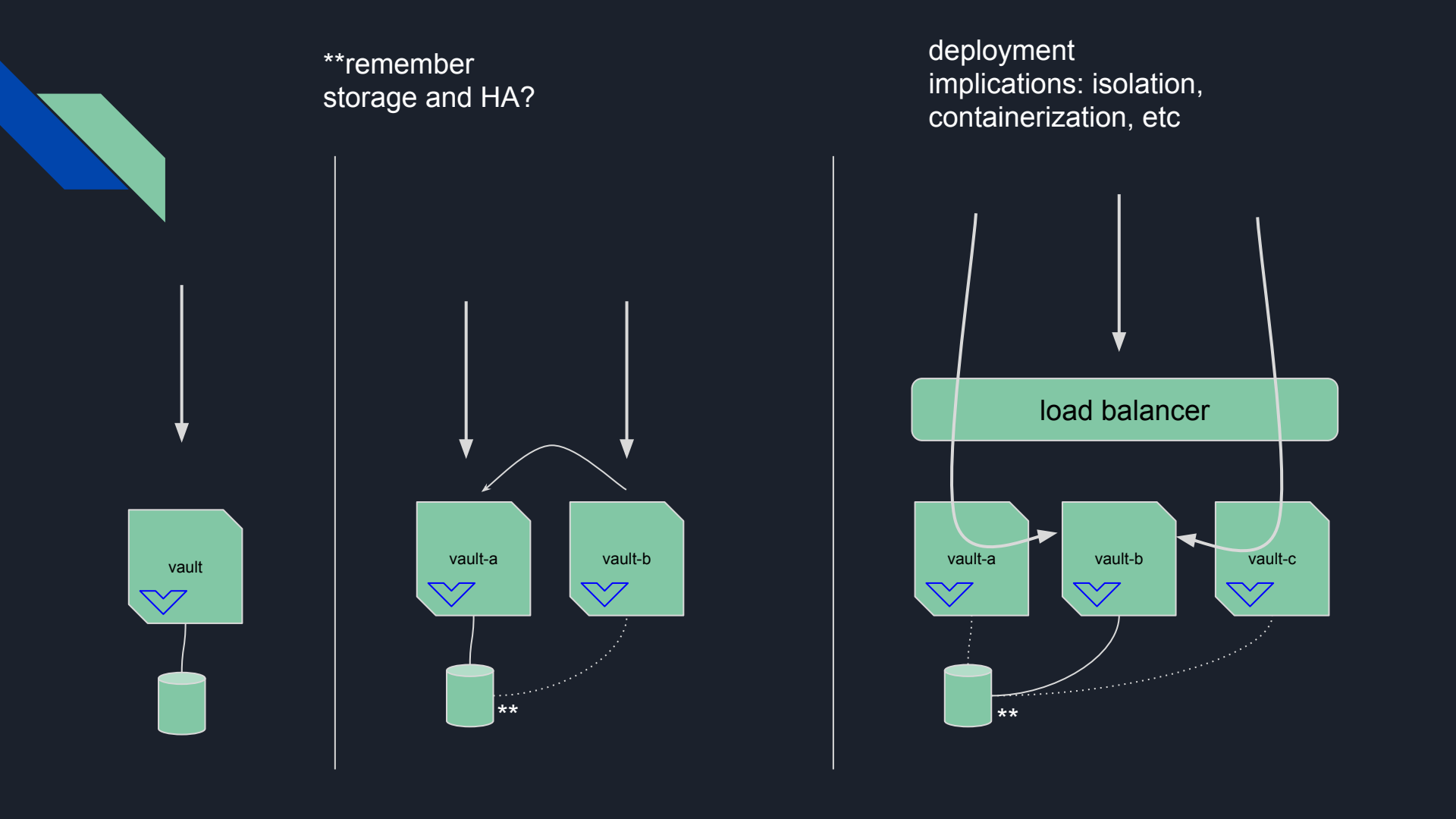
- Root Token Usage
- Raw Barrier Usage
- Usage Analytics
  - Request Breaktown
  - Active Users
  - Request Errors

Implementation & Scale

System Implementation/Design + Evolution

System Implementation/Design + Evolution




**remember
storage and HA?
deployment
implications: isolation,
containerization, etc
vault
vault-a
vault-b
**
load balancer
vault-a
vault-b
vault-c
**

Adoption and Scaling

- Technical Scaling is straightforward
- Scaling adoption and policy management is not
    - 4 questions for onboarding:

1. Identify **what** data will be stored in Vault
    a. Or if other secret backends need to be leveraged
2. Identify who will manage this data
3. Identify and document who (user) or what (robots, computers, applications, etc) will need access to read this data
4. Using #3, identify and implement an adequate authentication and retrieval method

Advanced Topics

- Plugins
- Sentinel
- Read the API Docs
- Read the Code

The End