Secret management with Vault

Vault at IT-DESY

Kai Wiemann
with Maximilian Kölpin, Thomas Hartmann, Krunoslav Sever, Sven Sternberger
Paris, 19th April 2024

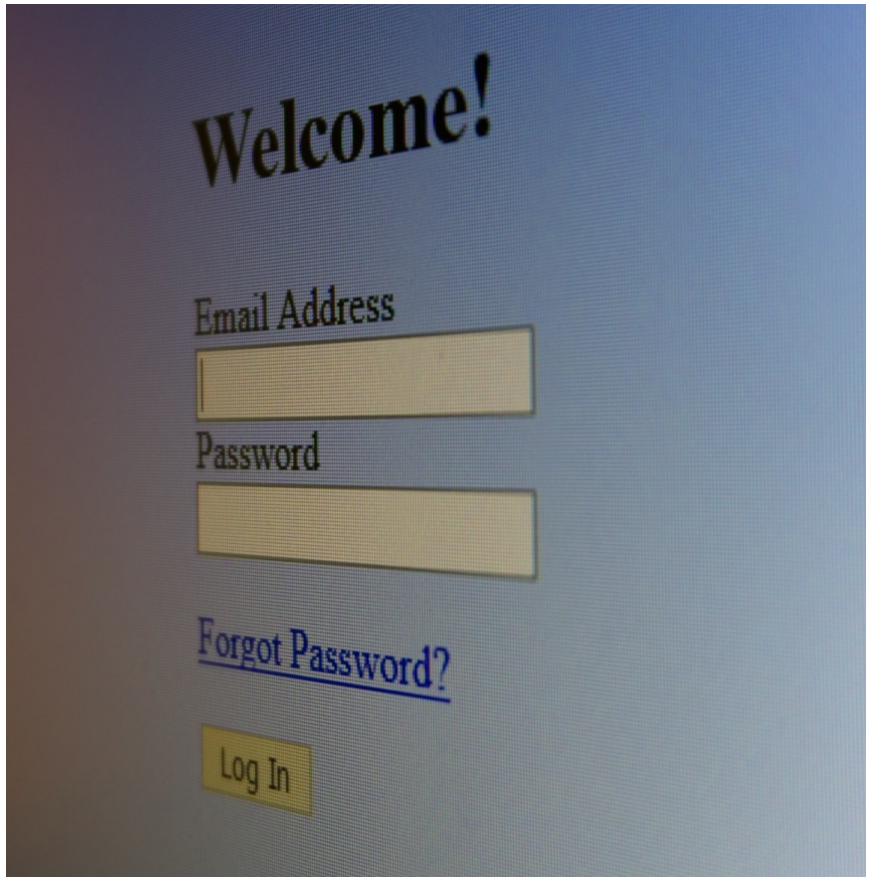
HELMHOLTZ

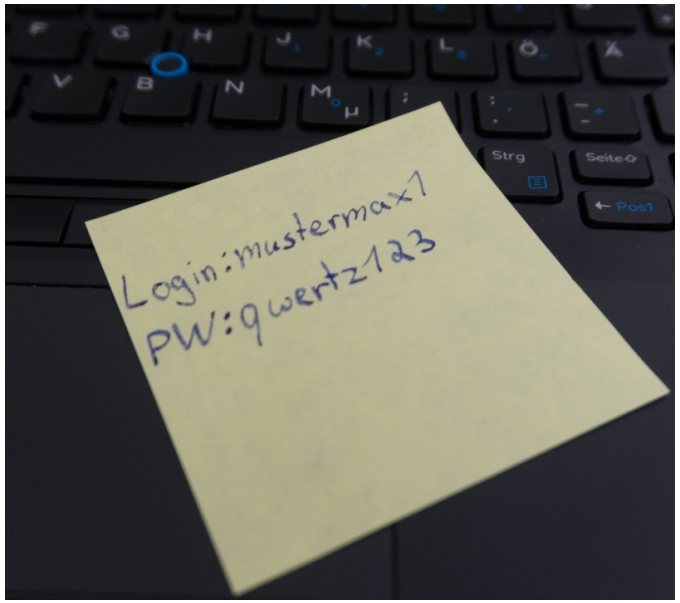
DESY.

Why secret management?

Why a secret management at all?

Starting situation



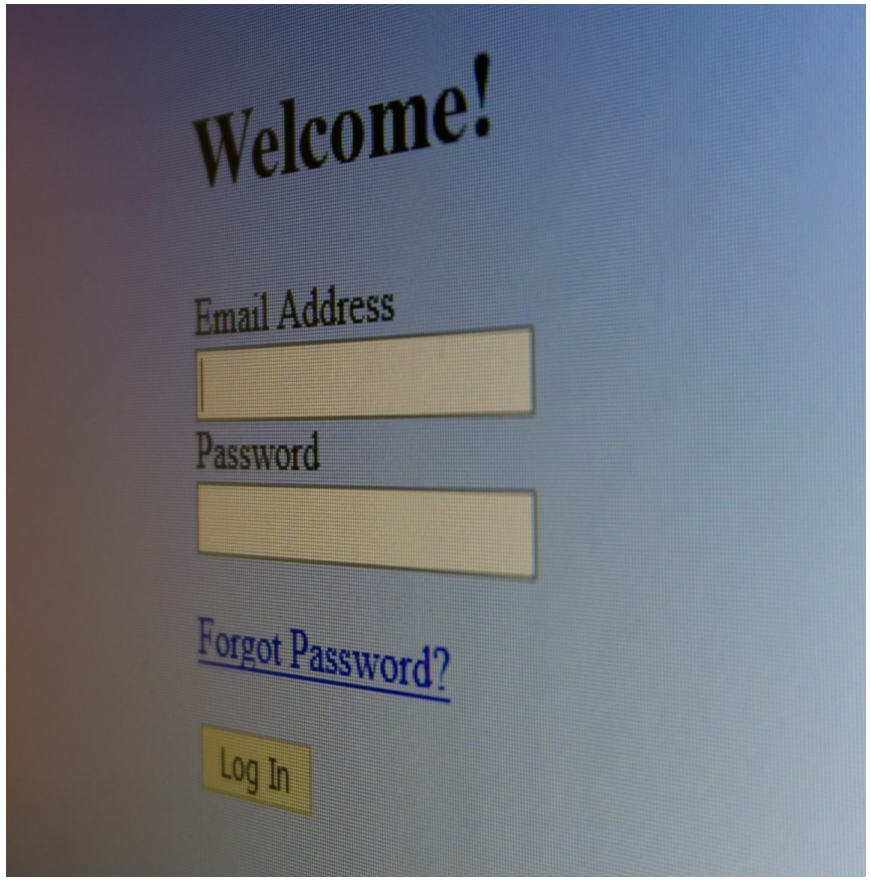


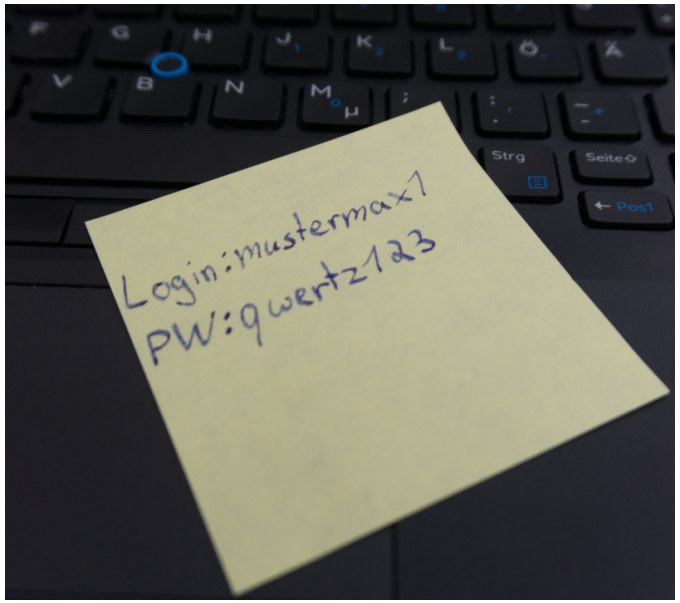

```
~% mysql -u root -p
Enter password:


# sensitive files
/etc/krb5.keytab
/etc/ssh/ssh_host_ed25519_key
/etc/ssh/ssh_host_rsa_key
```

```
~% ssh root@critical-host
Password:

~% curl -H "Authorization: Bearer $TOKEN" "https://.."
```

Why a secret management at all?

Starting situation






```
~% mysql -u root -p
Enter password:


# sensitive files
/etc/krb5.keytab
/etc/ssh/ssh_host_ed25519_key
/etc/ssh/ssh_host_rsa_key
```

```
~% ssh root@critical-host
Password:

~% curl -H "Authorization: Bearer $TOKEN" "https://.."
```

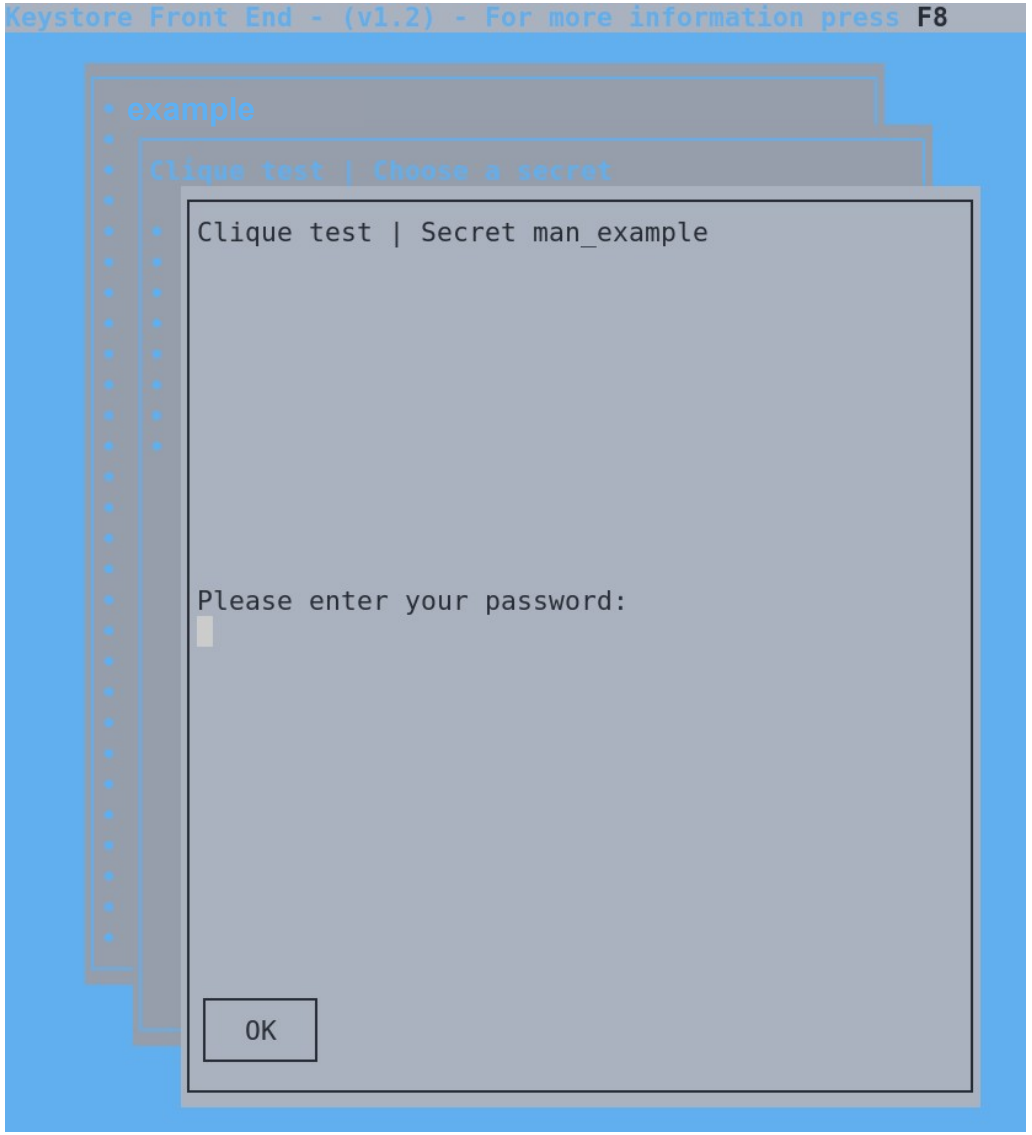
Central tool to manage secrets

Why a new secret management software?

Current tools

Keystore

- Interactive usage via CLI, TUI
- Authentication via public / private key
- ILO, root passwords, …
- Drawback: Only interactive, home development



Why a new secret management software?

Current tools

Keystore

- Interactive usage via CLI, TUI
- Authentication via public / private key
- ILO, root passwords, …
- Drawback: Only interactive, home development

Host Key Distribution service

- Automated storage, retrieval of host based secrets
- Authentication via IP, SFTP
- Kerberos keytabs, SSH host keys, Puppet certificates
- Drawback: Only host based, home development

Why a new secret management software?

Current tools

Keystore

- Interactive usage via CLI, TUI
- Authentication via public / private key
- ILO, root passwords, …
- Drawback: Only interactive, home development

Host Key Distribution service

- Automated storage, retrieval of host based secrets
- Authentication via IP, SFTP
- Kerberos keytabs, SSH host keys, Puppet certificates
- Drawback: Only host based, home development

Hashes in Puppet

- Set secrets via Puppet
- root passwords, database credentials
- Drawback: Hash must be used and is visible

Why a new secret management software?

Current tools

Keystore

- Interactive usage via CLI, TUI
- Authentication via public / private key
- ILO, root passwords, …
- Drawback: Only interactive, home development

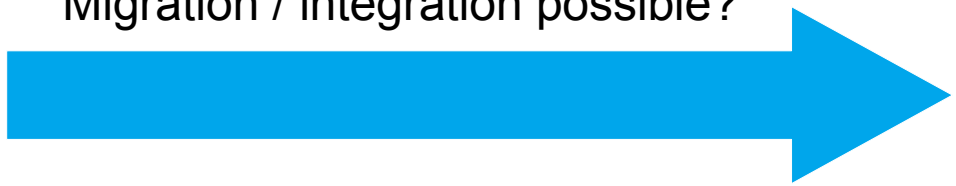
Host Key Distribution service

- Automated storage, retrieval of host based secrets
- Authentication via IP, SFTP
- Kerberos keytabs, SSH host keys, Puppet certificates
- Drawback: Only host based, home development

Hashes in Puppet

- Set secrets via Puppet
- root passwords, database credentials
- Drawback: Hash must be used and is visible

Migration / integration possible?



Why Vault?

Requirements

- Trusted, established software
- Options for integration, automation
- Authorization management
- Different methods to manage secrets

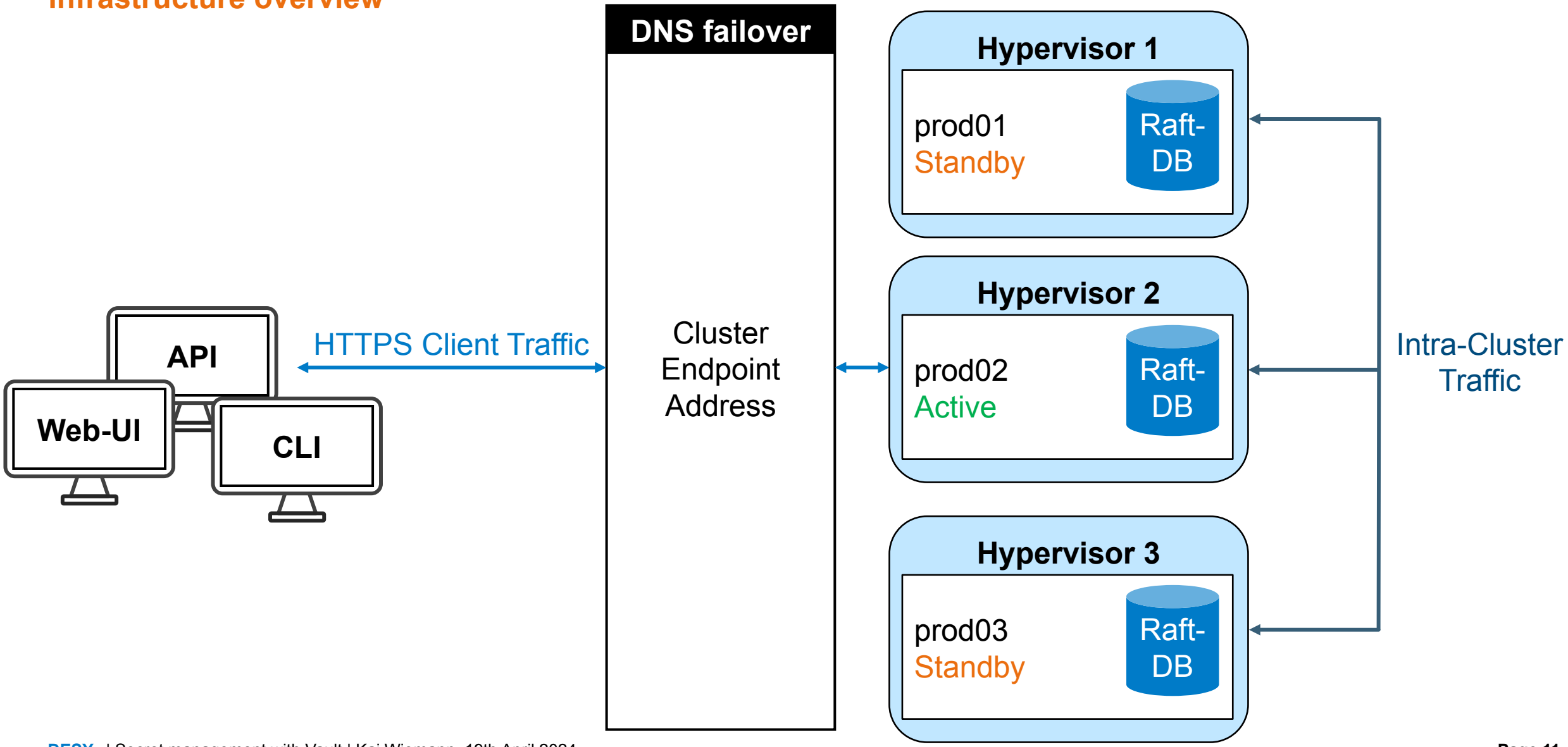
Vault

- Used by large companies, Open-Source*, audited, ISO-certified
- REST-API, addable modules called “engines”
- Based on policies; authentication via OIDC, JWT, certificates, ...
- Web interface, CLI, REST-API

Current Vault setup

Current Vault setup

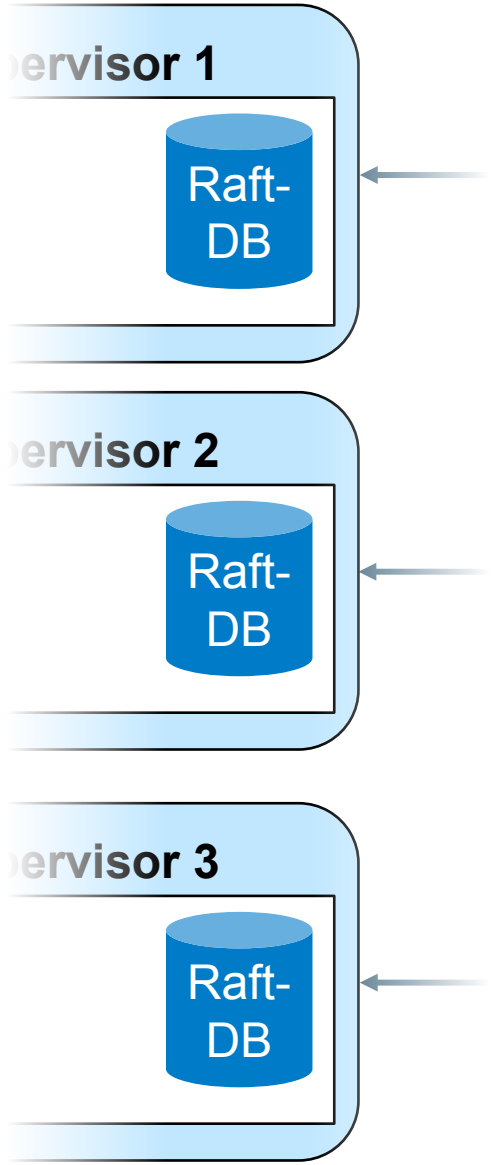
Infrastructure overview



Current Vault setup: Storage

Integrated Storage (Raft) as Storage Backend

- Recommended by HashiCorp and supported officially
- No additional software or clusters required
- Less administrative effort
- High availability due to Raft consensus algorithm
- Online backups possible with atomic snapshots



Current Vault setup: Security measures

Security measures – encrypted swap

- Vault usually prevents memory from being swapped to disk via `mlock` syscall
- Problem: Raft does not interact well with `mlock`
  - Vault documentation strongly recommends disabling usage of `mlock` in combination with Raft

➡ Deploy VMs with encrypted swap

```
~# dmsetup ls --target crypt
cryptswap       (253, 0)
```

> `disable_mlock (bool: false)` – Disables the server from executing the mlock syscall. mlock prevents memory from being swapped to disk. Disabling mlock is not recommended unless using integrated storage. […]
>
> Source: https://developer.hashicorp.com/vault/docs/configuration#disable_mlock

Current Vault setup: Security measures

Security measures – systemd hardening

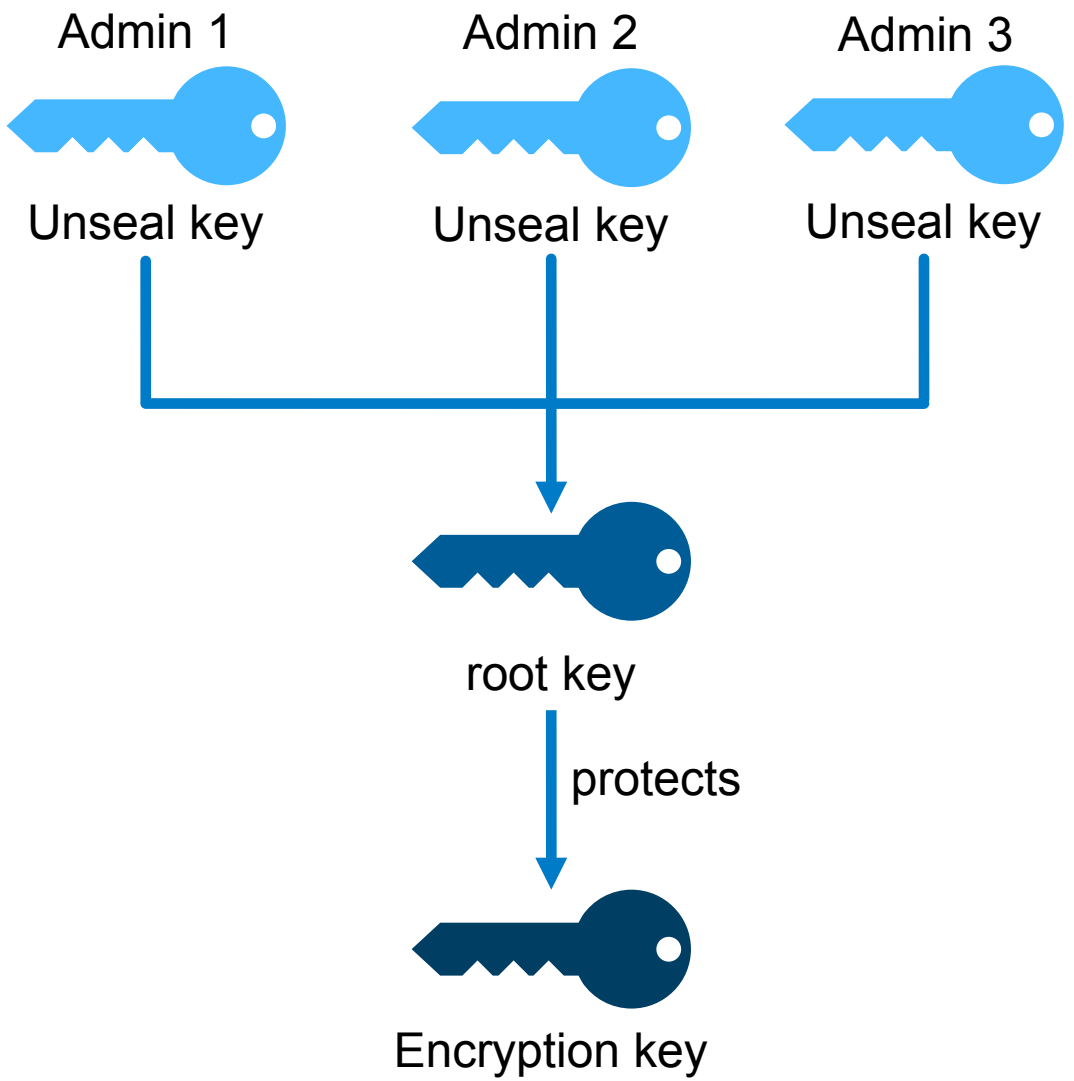
- Vault managed via systemd
- Use systemd to restrict capabilities of vault process
- `systemd-analyze security` helps
- Determining the minimum required capabilities with a bit of trial and error

```
~# systemd-analyze security --no-pager vault.service
  NAME                                    DESCRIPTION                                    EXPOSURE
✗ PrivateNetwork=                         Service has access to the host's network         0.5
✓ User=/DynamicUser=                      Service runs under a static non-root us…
✓ CapabilityBoundingSet=~CAP_SET(UID|GID|… Service cannot change UID/GID identitie…
[…]
→ Overall exposure level for vault.service: 2.0 OK 🙂
```

Current Vault setup: Security measures

Security measures – Unsealing Vault

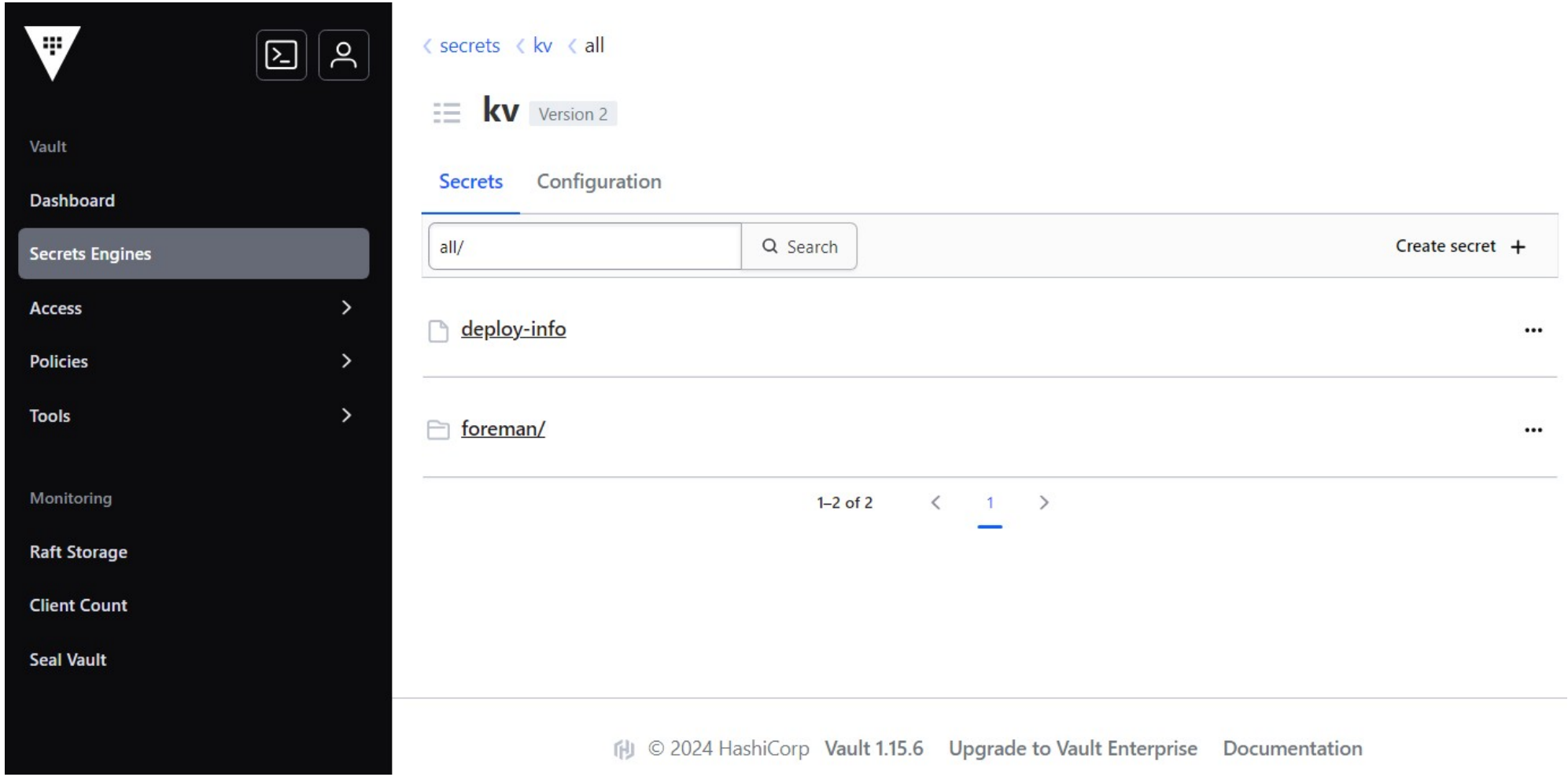
- Vault started sealed
  - Data encrypted
  - Vault unable to decrypt it by itself
  - Even after reboot, node failure
- Multiple unseal keys required to reconstruct the root key, unseal Vault
- PGP keys used to encrypt the unseal keys when Vault is initialized
- Also stored in Keystore



Managing secrets with Vault

Managing secrets with Vault

Interactive usage – Web interface



Managing secrets with Vault

Interactive usage – Command line interface / REST-API

```
❯ vault kv get kv/all/deploy-info
===== Secret Path =====
kv/data/all/deploy-info

======= Metadata =======
Key                Value
---                -----
created_time       2024-04-09T13:51:16.604114782Z
custom_metadata    <nil>
deletion_time      n/a
destroyed          false
version            2
====== Data ======
Key         Value
---         -----
password    foo
username    bar
```

Managing secrets with Vault

Interactive usage – Command line interface / REST-API

```
❯ vault path-help kv | grep --fixed-strings --after-context=1 '<path>'
    ^data/(?P<path>.*?[^/]$)$
        Write, Patch, Read, and Delete data in the Key-Value Store.
--
    ^delete/(?P<path>.*)$
        Marks one or more versions as deleted in the KV store.
--
    ^destroy/(?P<path>.*)$
        Permanently removes one or more versions in the KV store
[…]

❯ vault kv get -output-curl-string kv/all/deploy-info
[…]
❯ curl -H "X-Vault-Request: true" \
       -H "X-Vault-Token: $(vault print token)" \
       "https://$VAULT_ADDR/v1/kv/data/all/deploy-info" | jq '.data.data'
{
  "password": "foo",
  "username": "bar"
}
```

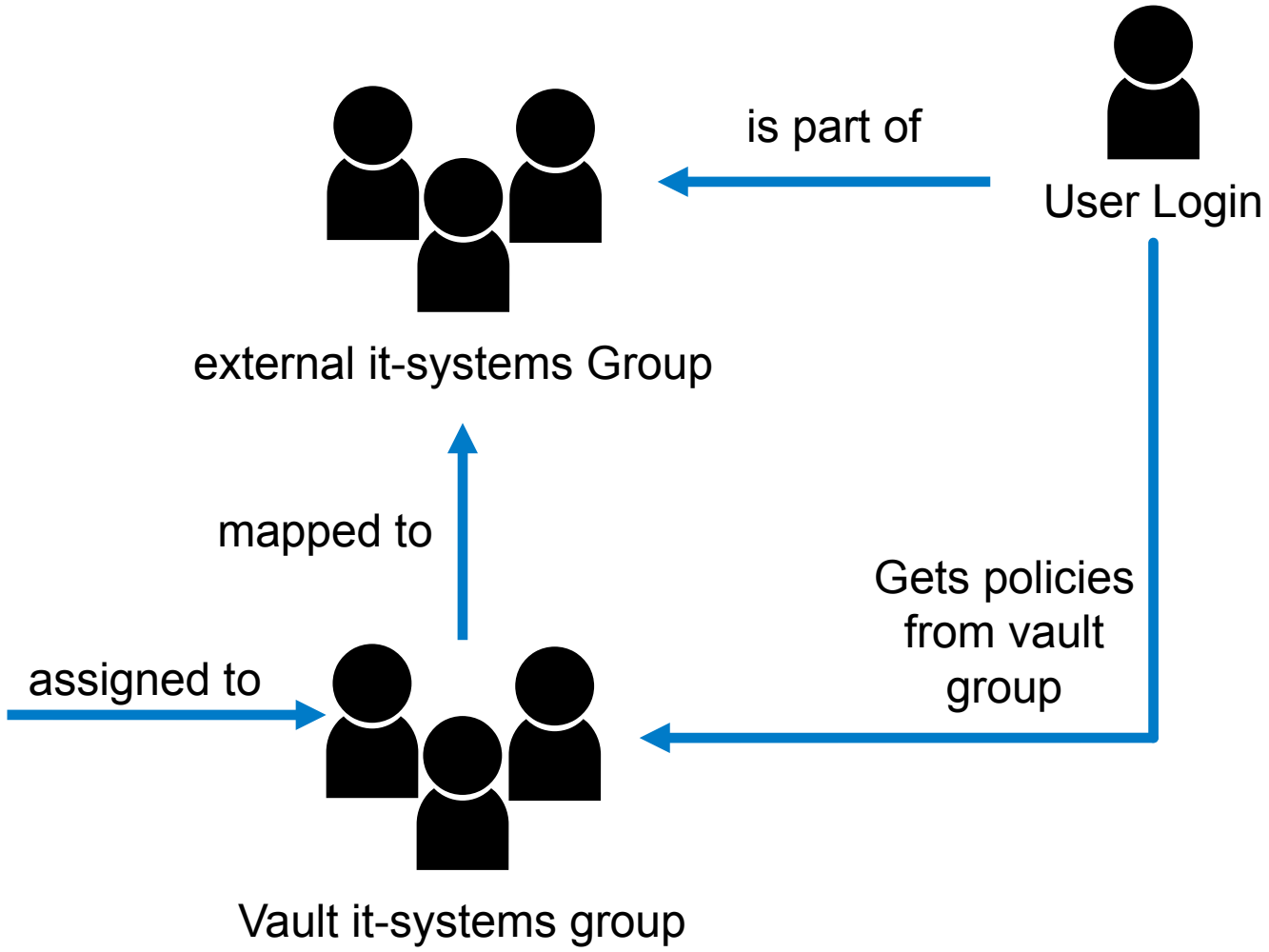
Managing secrets with Vault: Policies

Permission management via Policies

- Deny by default
- Templating possible
- Allows mapping to external groups

it-systems.hcl

```
[…]
path "kv/data/groups/it-systems/*" {
  capabilities = [ "read", "list", "create" ]
}
[…]
```



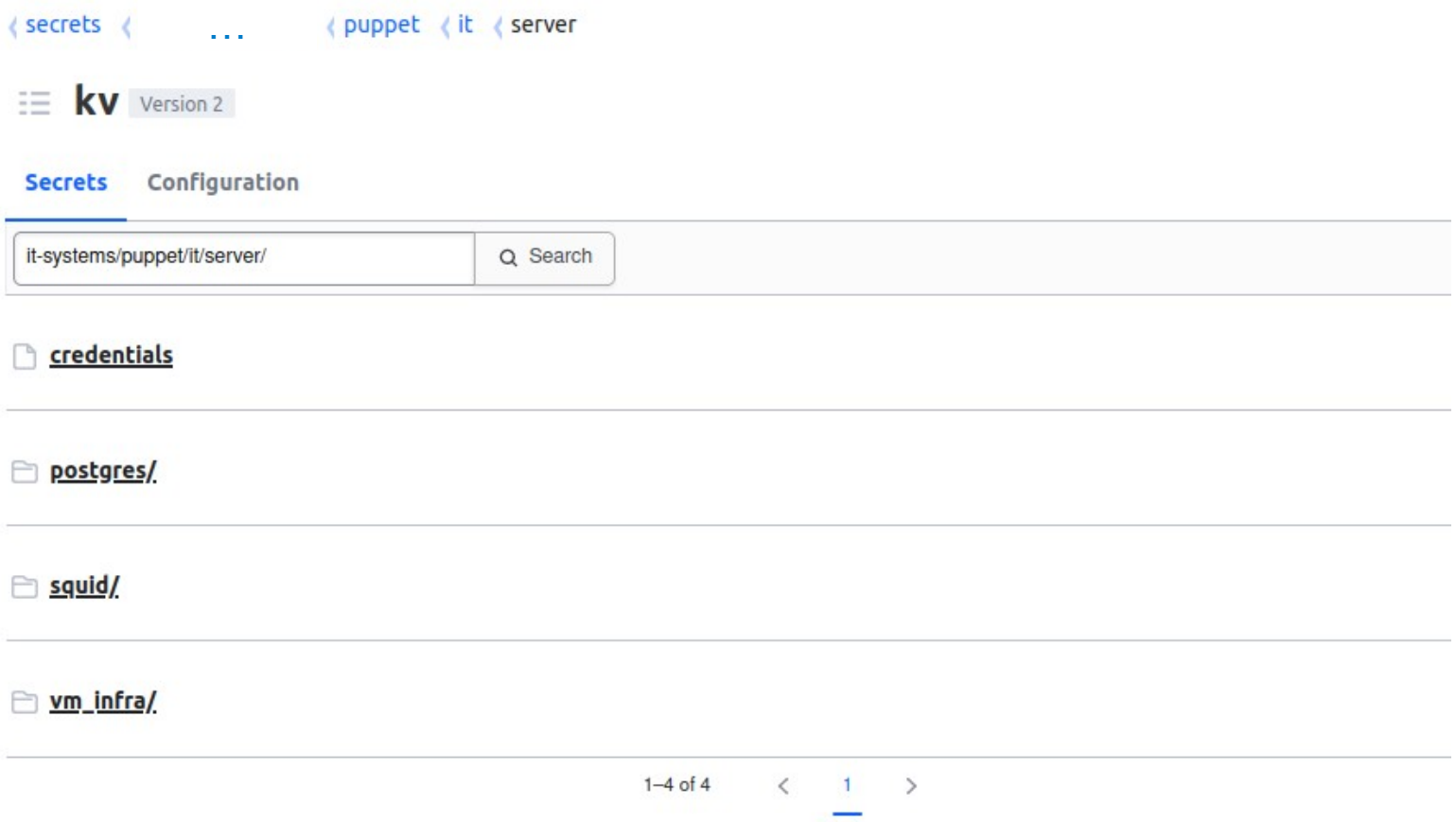
Managing secrets with Vault: Puppet integration

Puppet secret layout

Puppet, Hostgroups

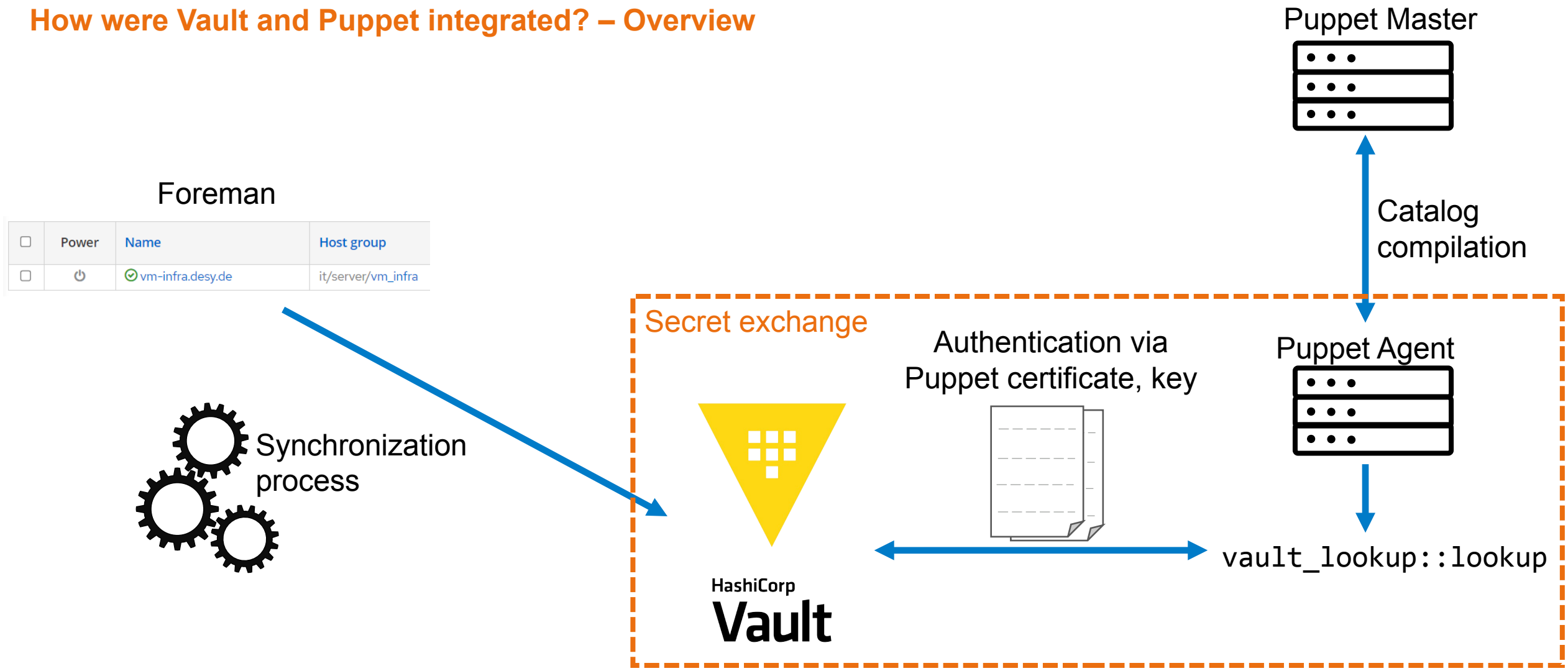
```
├── it
│   ├── server
│   │   ├── postgres
│   │   ├── squid
│   │   └── vm_infra
│   ├── vault
│   │   ├── prod
│   │   └── registry
│   ├── vaultui
│   └── wgs
├── maxwell
│   ├── infra
│   │   ├── jhub
│   │   └── web
│   ├── slurm
│   │   ├── admin
│   │   └── db
│   └── wgs
..
```

Vault



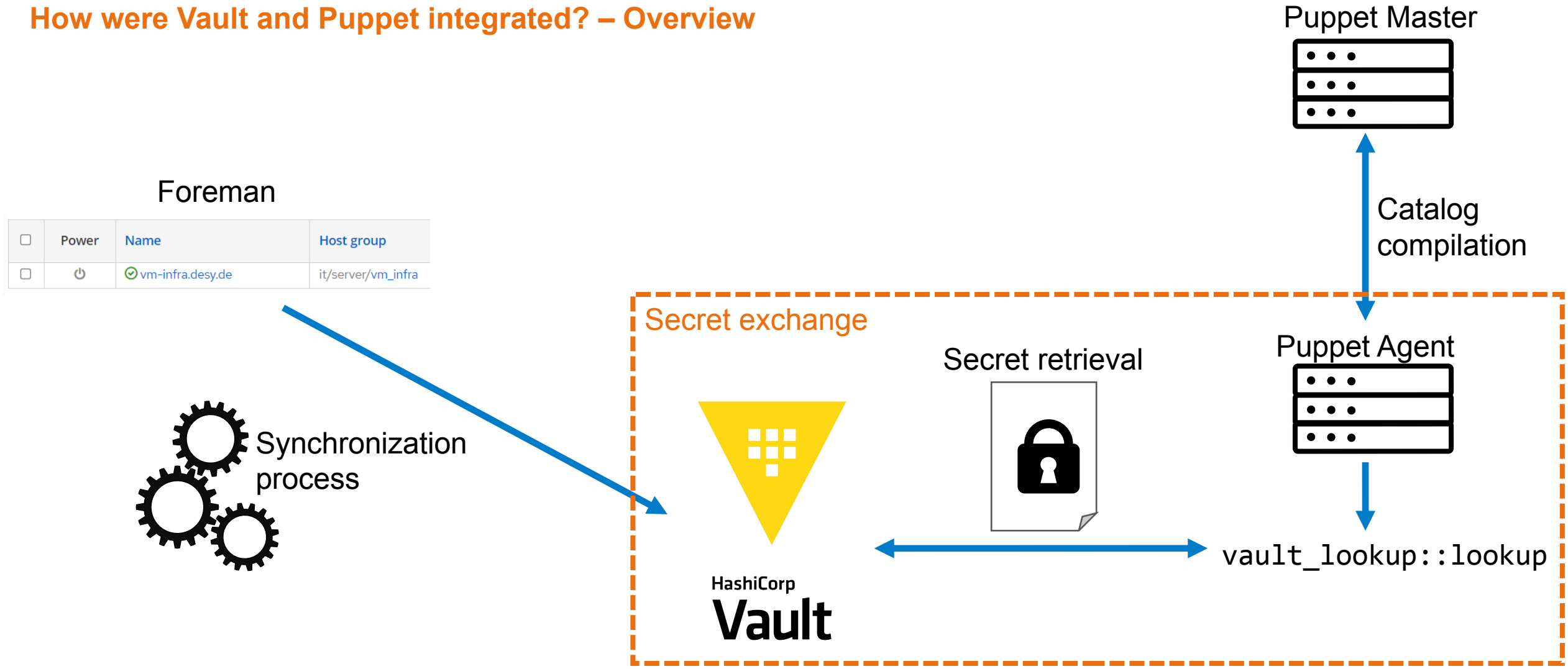
Managing secrets with Vault: Puppet integration

How were Vault and Puppet integrated? – Overview



Managing secrets with Vault: Puppet integration

How were Vault and Puppet integrated? – Overview



Managing secrets with Vault: Puppet integration

How were Vault and Puppet integrated? – Example

```
❯ cat manifests/init.pp
[…]
 $example_secret = Deferred('vault_lookup::lookup',
    ['[…]/puppet/it/server/credentials', {
      vault_addr => lookup('vault_lookup::vault_addr', Optional[String], undef, undef),
      field      => 'password',
  }])

  file { '/tmp/test-password':
    ensure  => file,
    content => $example_secret,
  }
[…]

~# puppet agent -t
[…]
Notice: /Stage[main]/Vm_infra/File[/tmp/test-password]/ensure: changed [redacted] to [redacted]
Notice: Applied catalog in 12.01 seconds
~# cat /tmp/test-password
test123
```

Managing secrets with Vault: Puppet integration

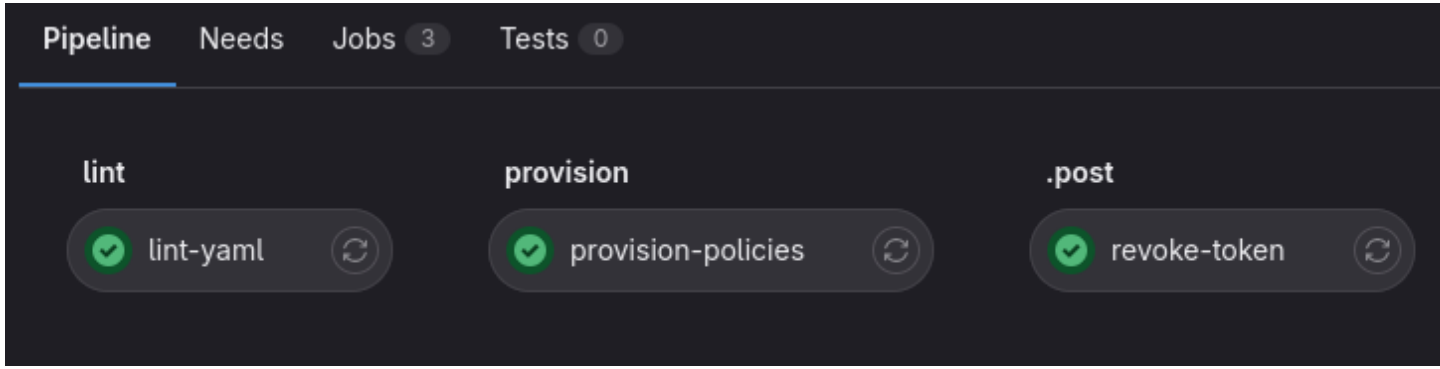
How were Vault and Puppet integrated? – Vault view

```
❯ vault read -format=json identity/entity/name/vm-infra.desy.de |
    jq '.data | {"metadata": .metadata, "name": .name}'
{
  "metadata": {
    "allow_delete_via_foreman_sync": "true",
    "hostgroup_0": "it",
    "hostgroup_1": "server",
    "hostgroup_2": "vm_infra"
  },
  "name": "vm-infra.desy.de"
}

❯ vault policy read puppetserver
[…]
path "[…]/puppet/{{identity.entity.metadata.hostgroup_0}}/{{identity.entity.metadata.hostgroup_1}}/+" {
  capabilities = [ "read", "list" ]
}
[…]
```

Managing secrets with Vault: Policy management

But how to manage policies?

```
data
├── configure
│   ├── auth
│   │   ├── approle
│   │   │   └── role
│   │   │       ...
│   │   │           └── vault-foreman-sync.yaml
│   │   ├── cert
│   │   │   └── certs
│   │   │       └── puppetserver.yaml
│   │   ...
│   └── sys
│       └── policies
│           └── acl
│               ...
│                   └── vault-foreman-sync.hcl
└── enable
    └── sys
        └── auth
            ├── cert.yaml
            ...
```



Outlook

Outlook

What are the next steps?

- Manage Kubernetes secrets with Vault
- Consolidate secrets stored in other tools
- Work on offline backup strategy and disaster recovery
- Evaluate OpenBao due to HashiCorp’s license change

Thank you for your time

Questions?

Acknowledgments

Thank you for your work and support

- Maximilian Kölpin
- Thomas Hartmann
- Krunoslav Sever
- Sven Sternberger

Appendix

Backup strategy

Offline backup

- Currently only daily backups via TSM
- Idea: Use WORM (Write once read many) USB drives for offline backup
- Write script to create Raft snapshot when USB drive is inserted