



DevOps
Conf **2022**

Мультиластерный масштабируемый Vault на тысячи сервисов – это ОК

Иван Буймов
Одноклассники



- ~10 лет в ИТ
- 4 года в ОК
- Занимаюсь автоматизацией
- Много пишу на Python
- Немного пишу на Go

На фото я чиню сервера во время летнего корпоратива на прошлом месте работы

Иван Буймов
Одноклассники

О чём сегодня поговорим

1

Инфраструктура
ОК

2

Интеграция с
нашими системами
и аутентификация

3

Отказ ДЦ и
масштабирование

4

Решение проблем,
что получилось

5

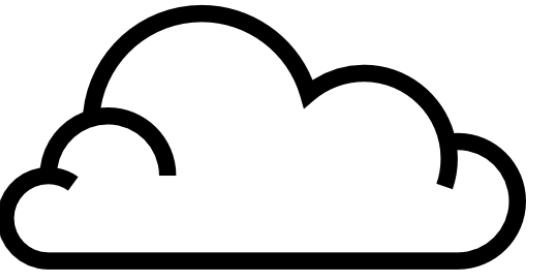
Вопросы

Одноклассники сегодня

о
х

Одноклассники сегодня

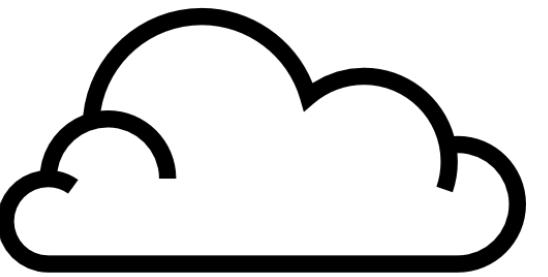
о
х



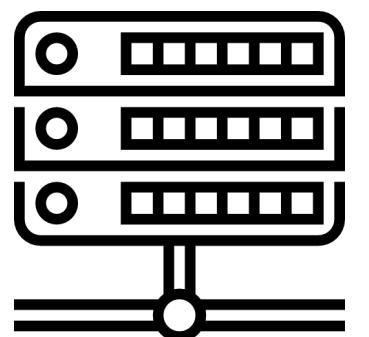
Десятки тысяч
контейнеров
в облаке one-cloud

Одноклассники сегодня

ОК

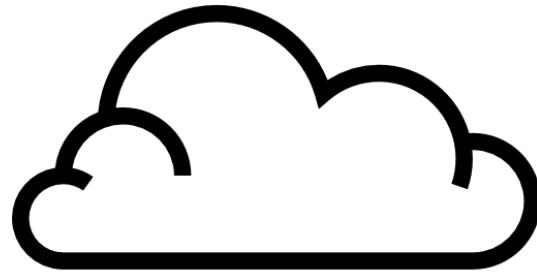


Десятки тысяч
контейнеров
в облаке one-cloud

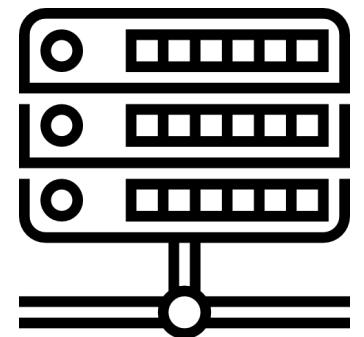


10k серверов

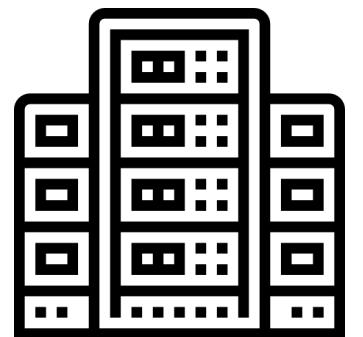
Одноклассники сегодня



Десятки тысяч
контейнеров
в облаке one-cloud



10k серверов

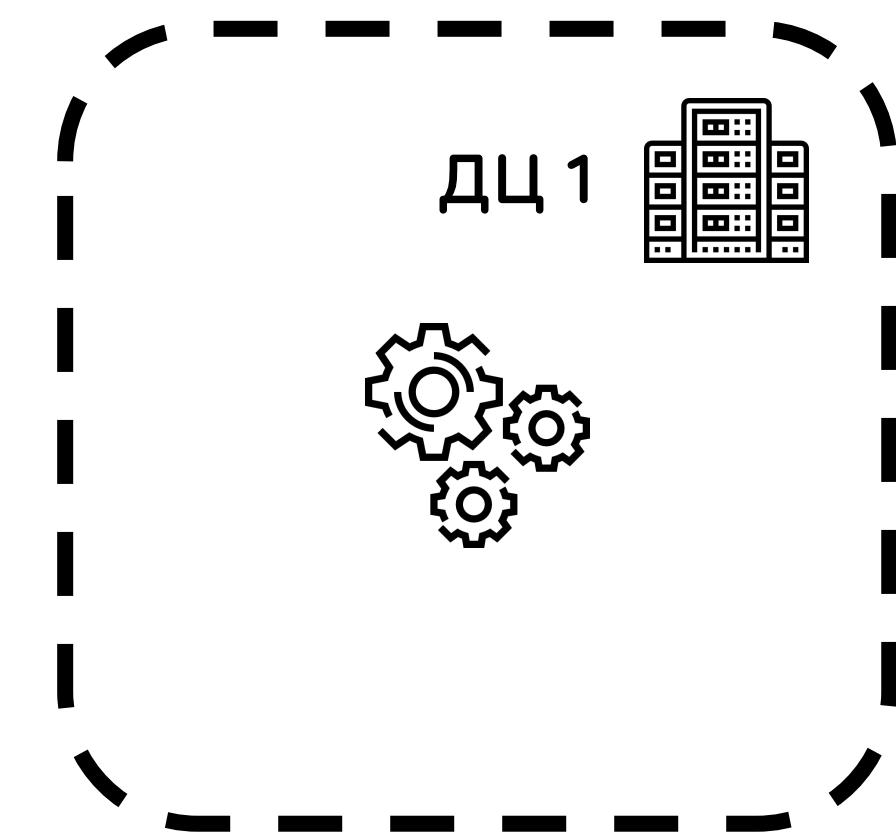


7 дата-центров

Наши стандарты

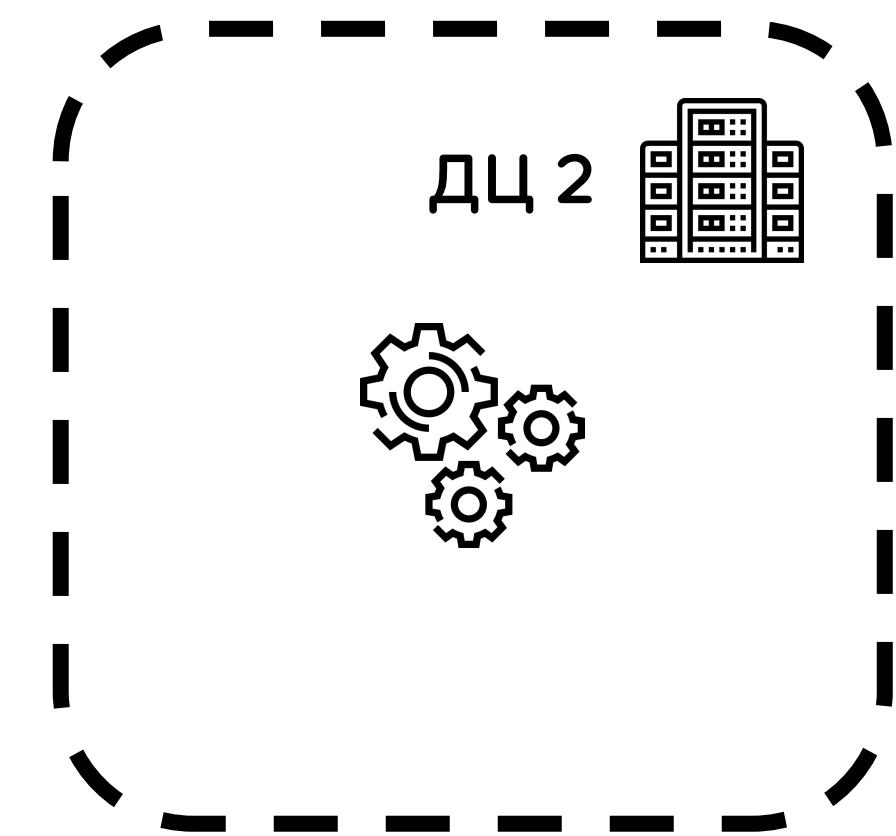
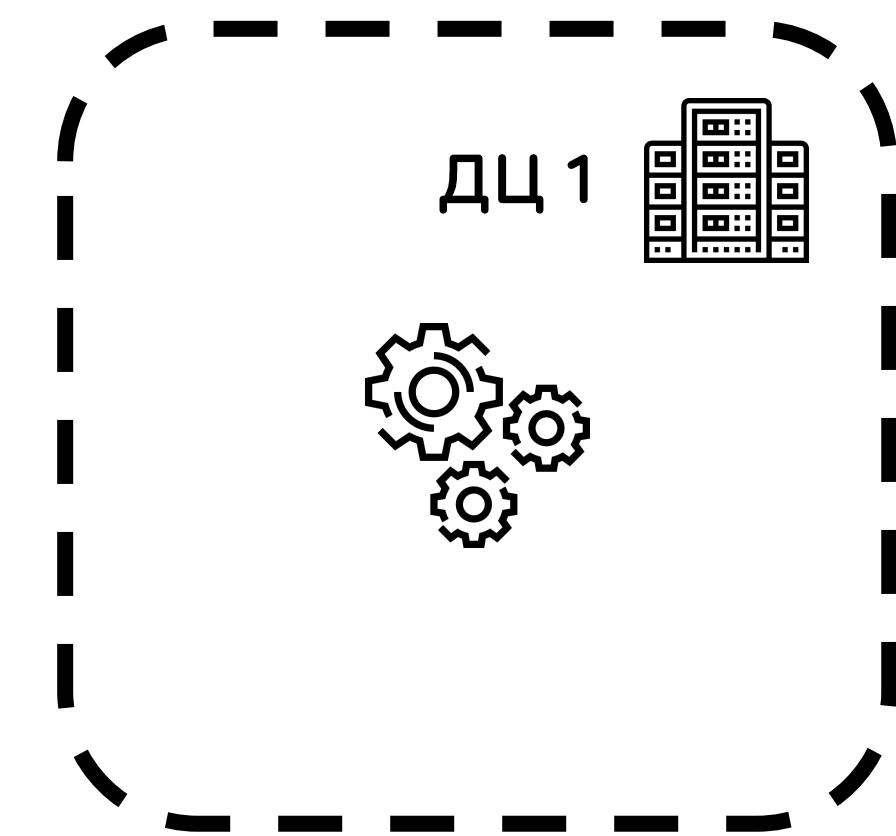
Наши стандарты

RF3 – обязательно



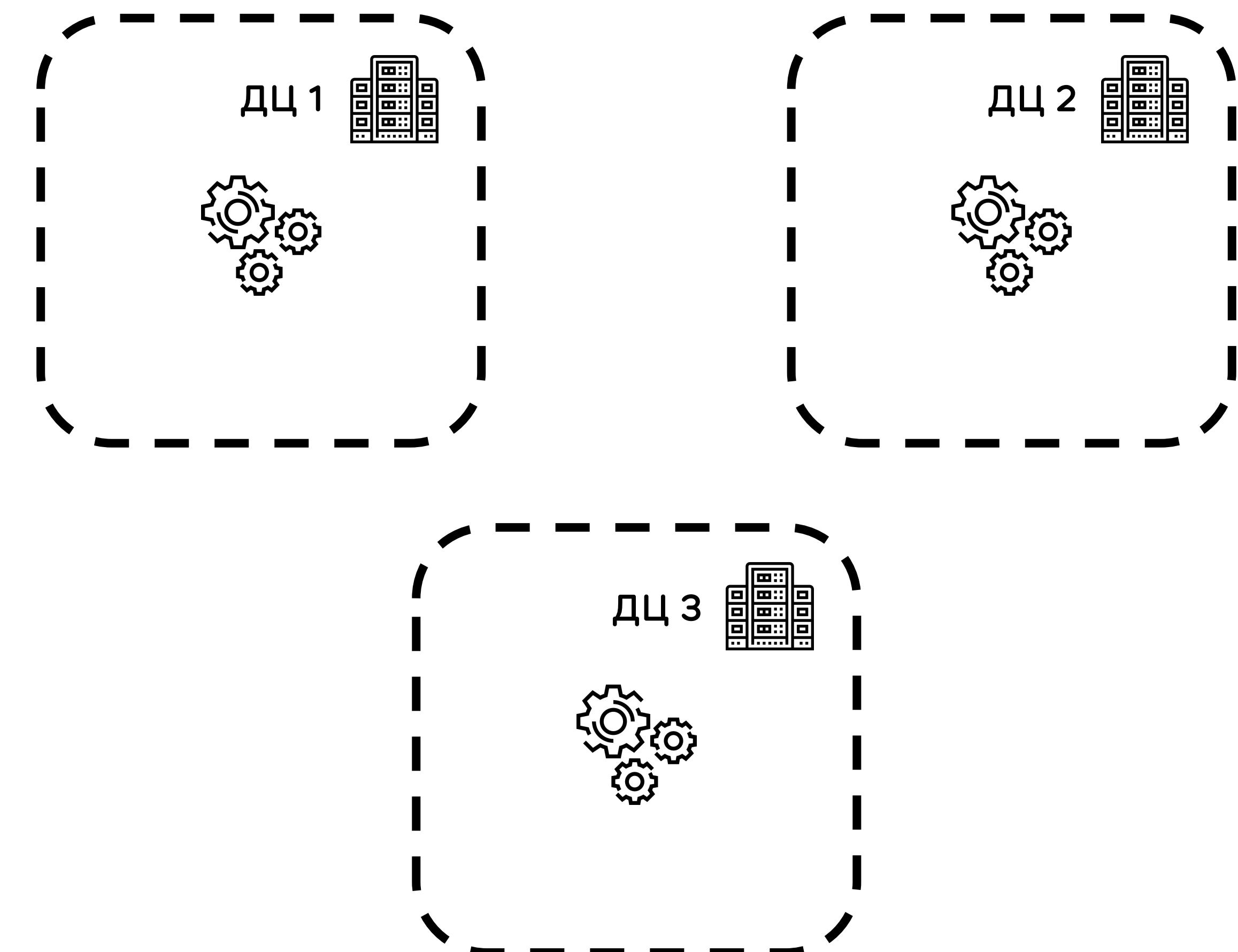
Наши стандарты

RF3 – обязательно



Наши стандарты

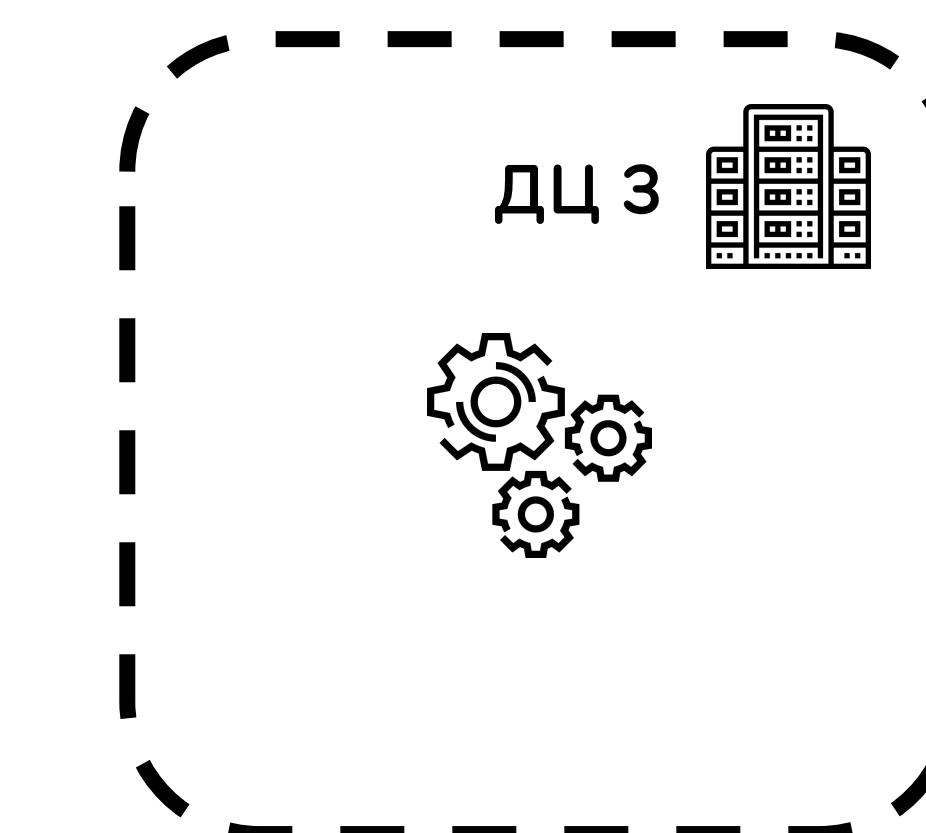
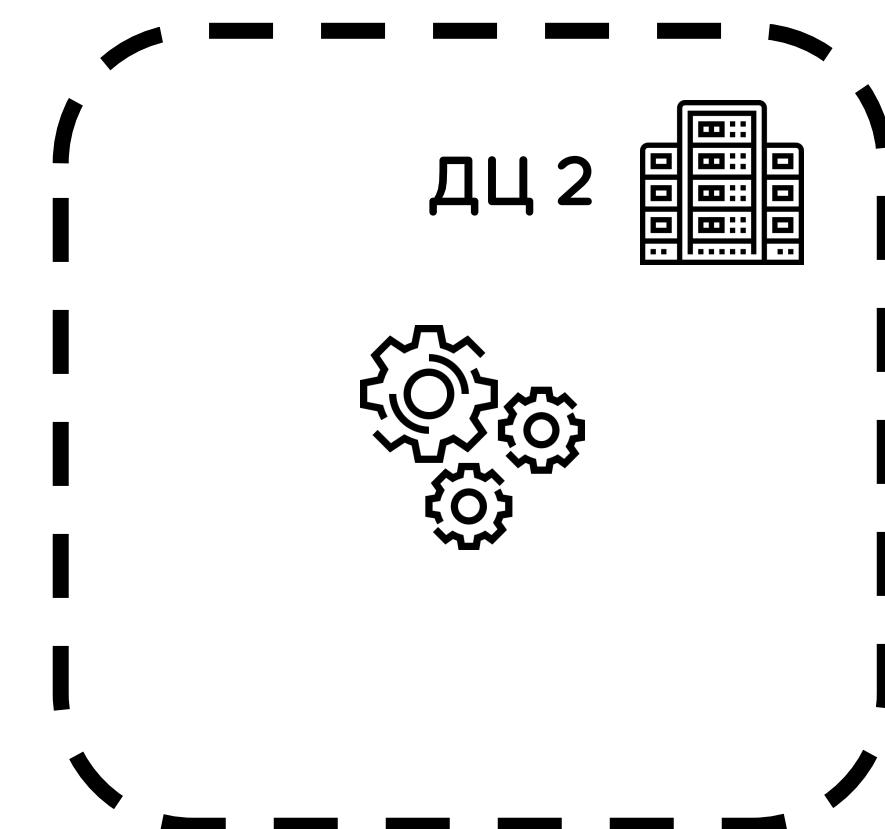
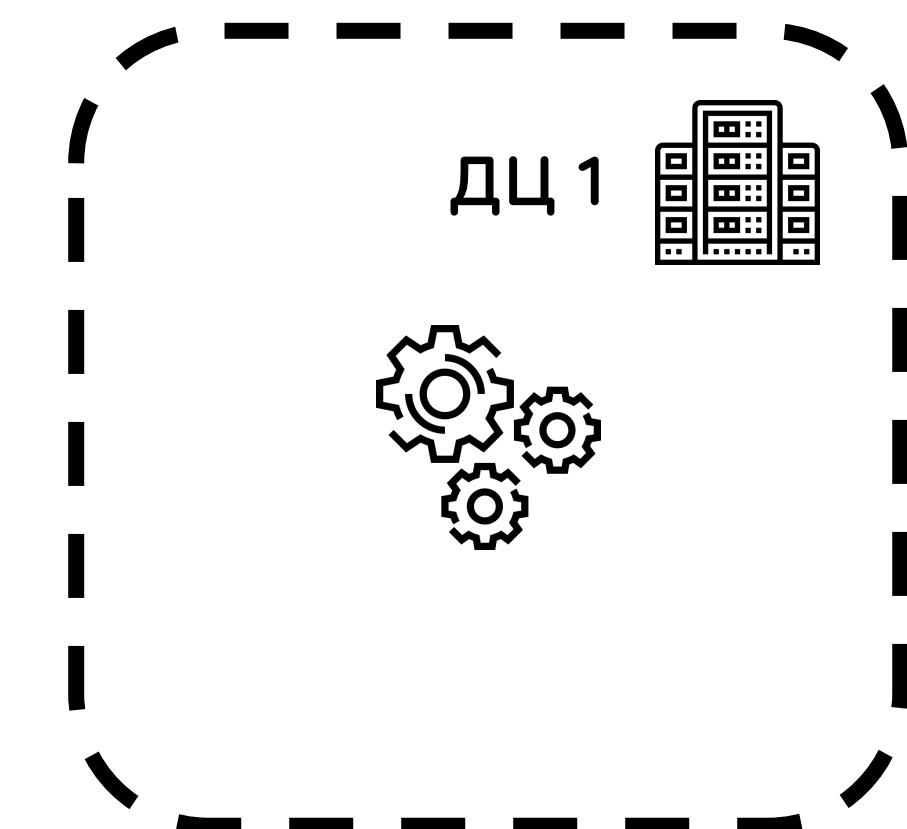
RF3 – обязательно



Наши стандарты

RF3 – обязательно

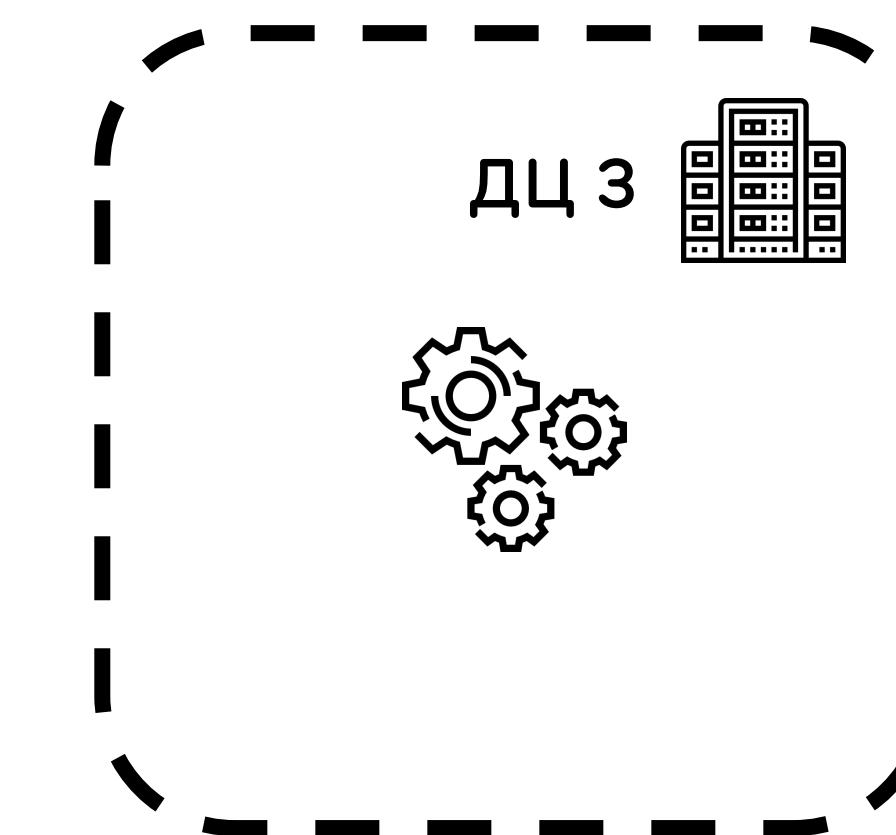
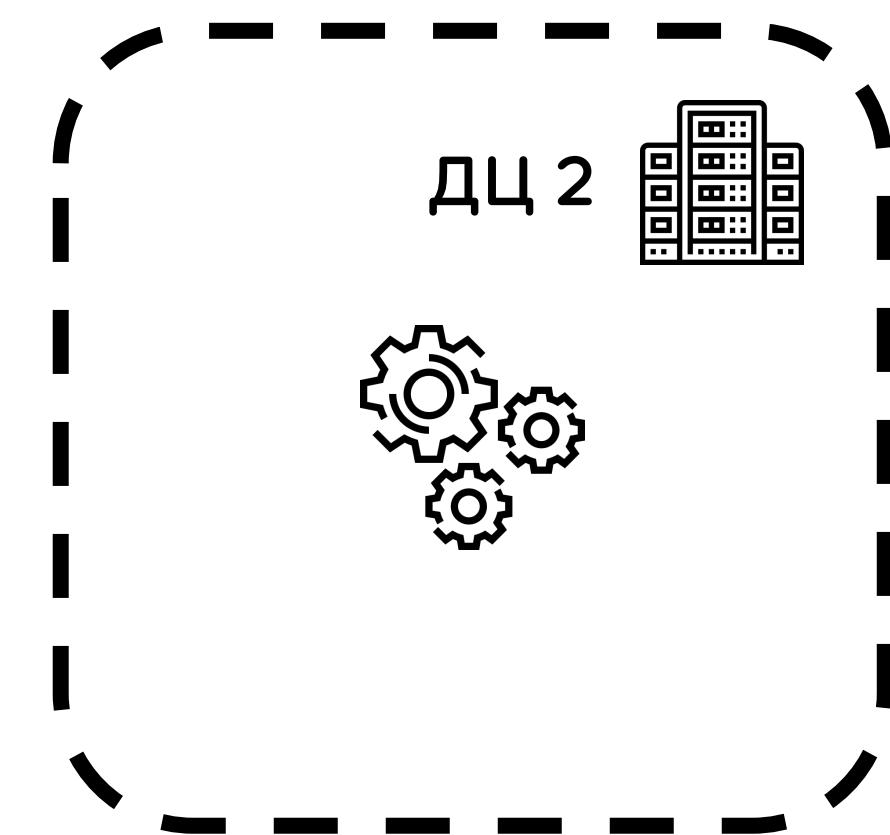
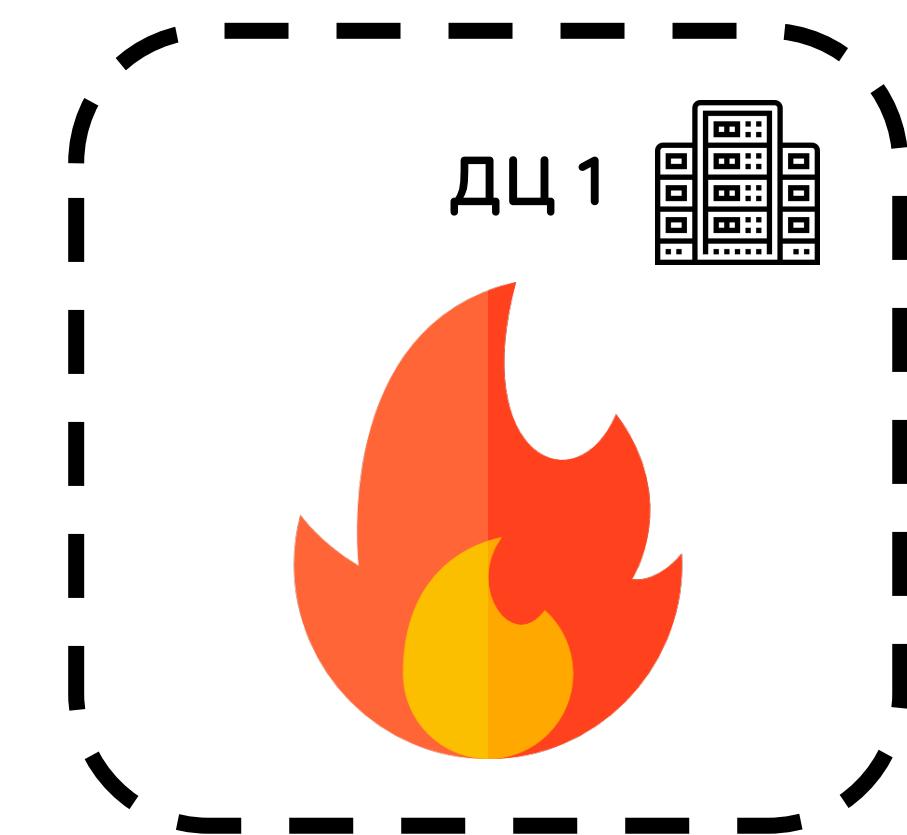
Отказ дата-центра



Наши стандарты

RF3 – обязательно

Отказ дата-центра

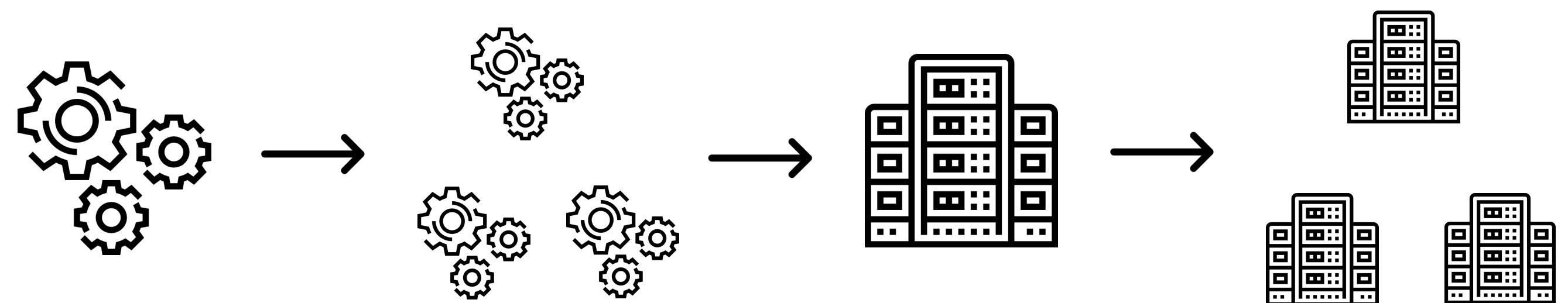


Наши стандарты

RF3 – обязательно

Отказ дата-центра

Плавное применение
изменений

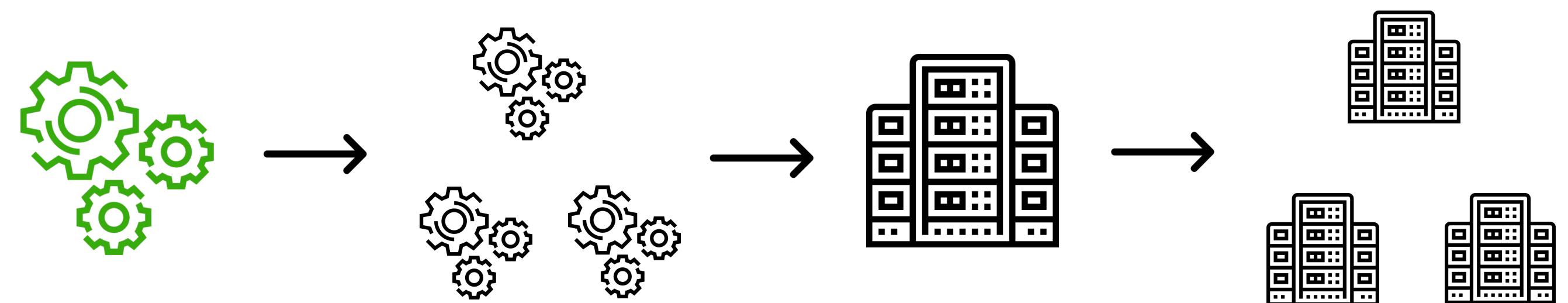


Наши стандарты

RF3 – обязательно

Отказ дата-центра

Плавное применение
изменений

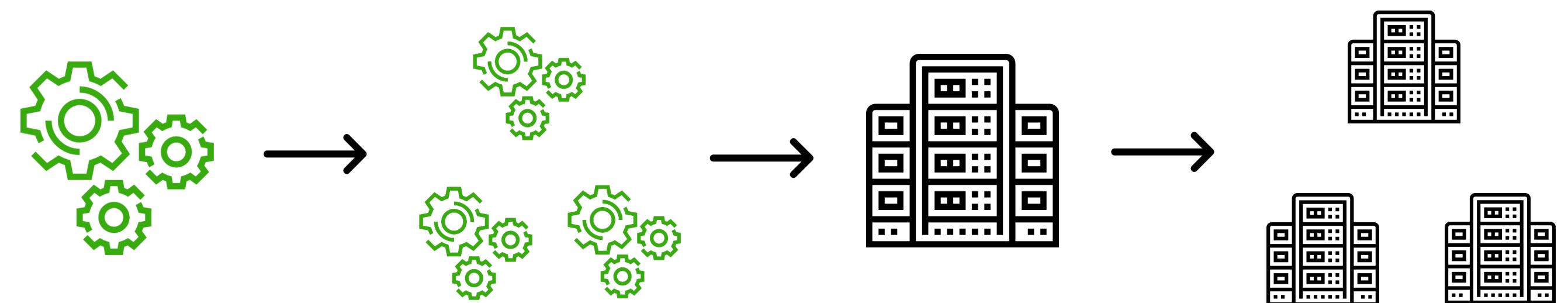


Наши стандарты

RF3 – обязательно

Отказ дата-центра

Плавное применение
изменений

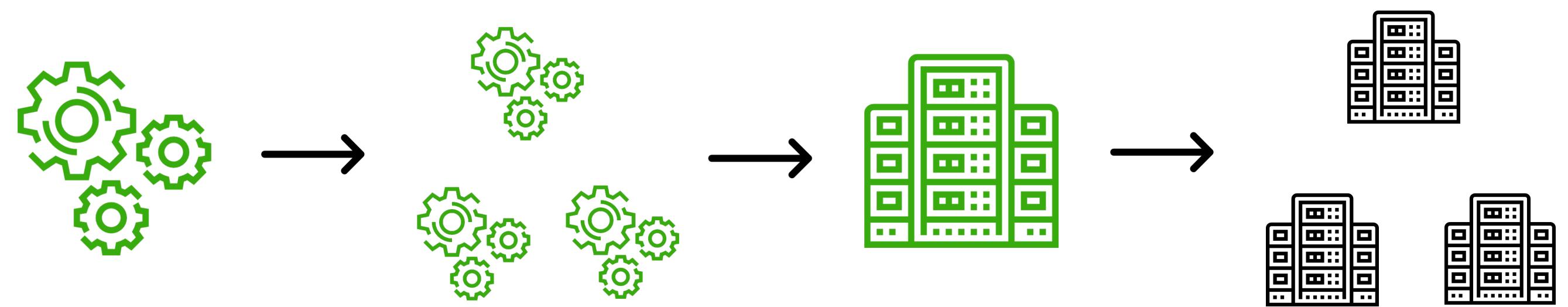


Наши стандарты

RF3 – обязательно

Отказ дата-центра

Плавное применение
изменений

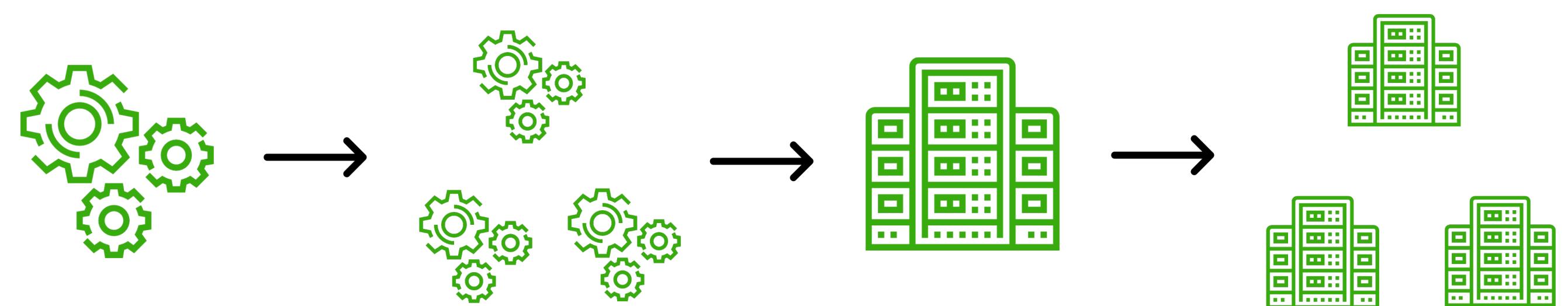


Наши стандарты

RF3 – обязательно

Отказ дата-центра

Плавное применение
изменений



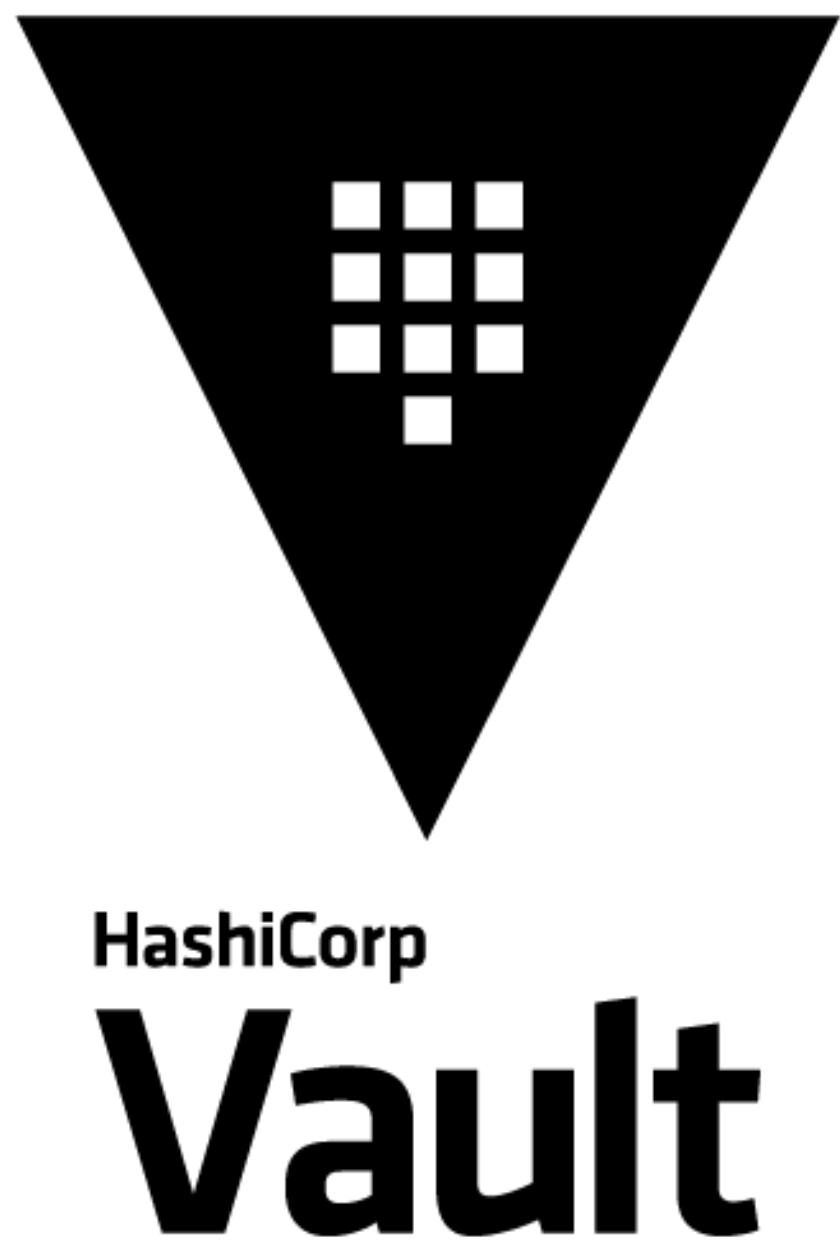
Требования

- Шифрование
- Аудит
- Удобный доступ для людей и машин (UI, API, cli)
- Версионирование секретов
- Интеграция с нашими системами: CMDB, CM
(Ansible CFEngine), one-cloud
- Наши стандарты

HashiCorp Vault

ХО

- Де-факто является стандартом в индустрии
- Поддержка в стороннем ПО (TeamCity, Ansible)
- OpenSource
- Web UI, REST API, cli, libs





Отлично! Но чего-то
не хватает...



- Интеграция с нашими системами
- Выписка сертификатов
- Отказ дата-центра
- Масштабирование

ХО

Что нужно для чтения секрета?

ХО

Vault-токен

Что нужно для чтения секрета?



Vault-токен

Как получить токен?

Что нужно для чтения секрета?



Vault-токен

Как получить токен?

Пройти аутентификацию в Vault

Что нужно для чтения секрета?



Vault-токен

Как получить токен?

Пройти аутентификацию в Vault

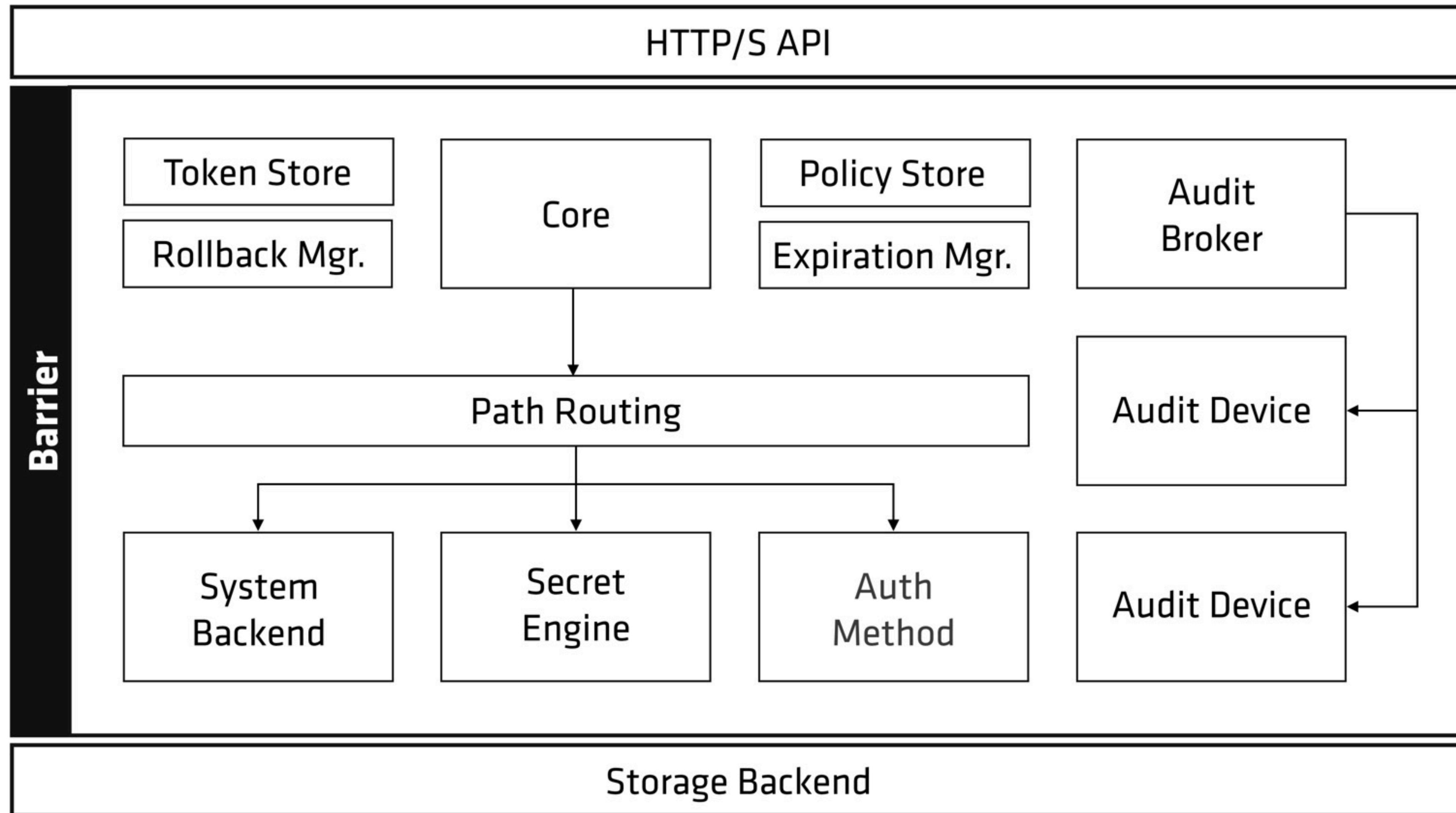
Как пройти аутентификацию?

Секрет для доступа к секретам

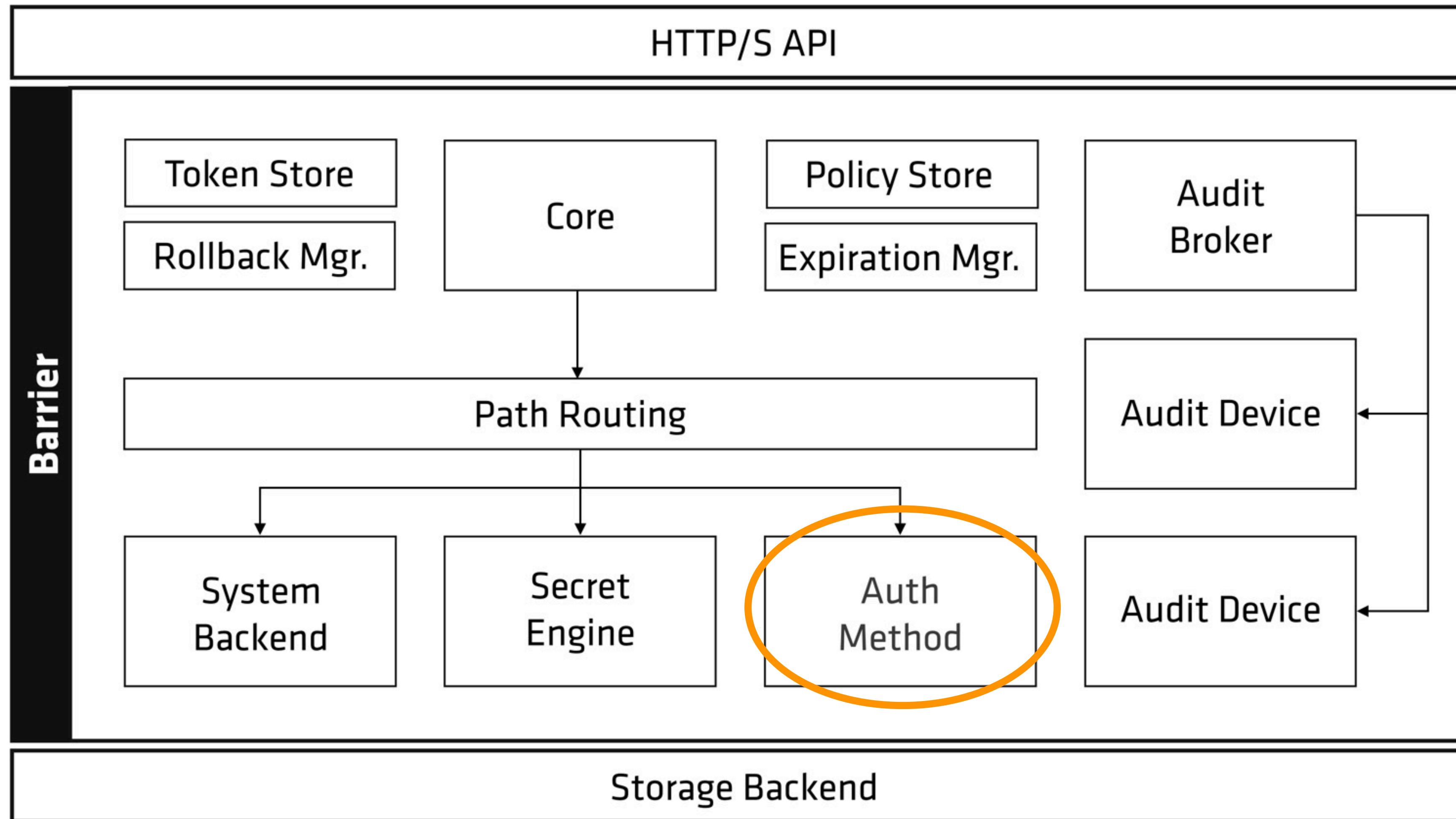
- Нужен доверенный источник, обычно это системы **CI/CD** или **Configuration Management**



Архитектура Vault



Архитектура Vault



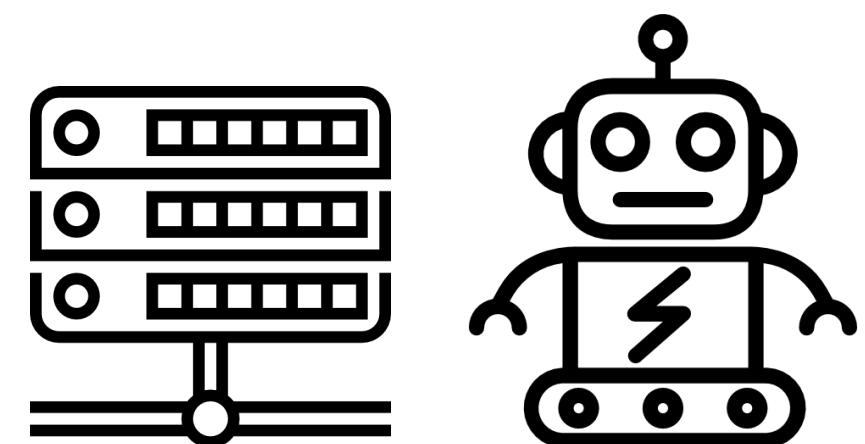
Как пройти аутентификацию?

ХО



Как пройти аутентификацию?

ХО

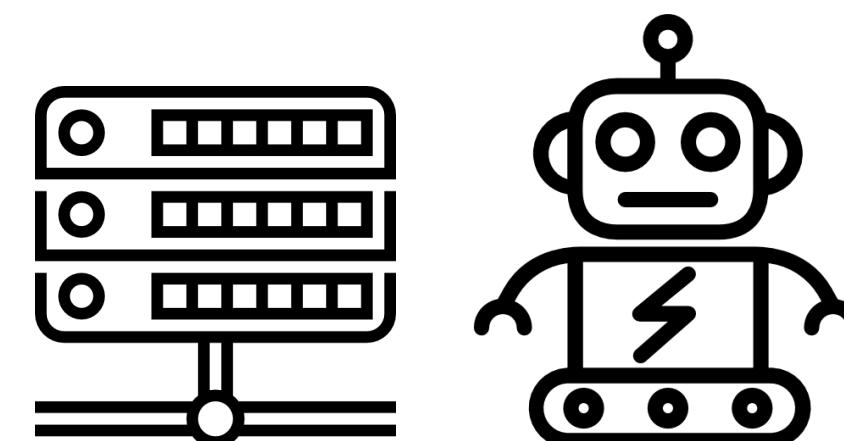


Как пройти аутентификацию?

ХО



LDAP

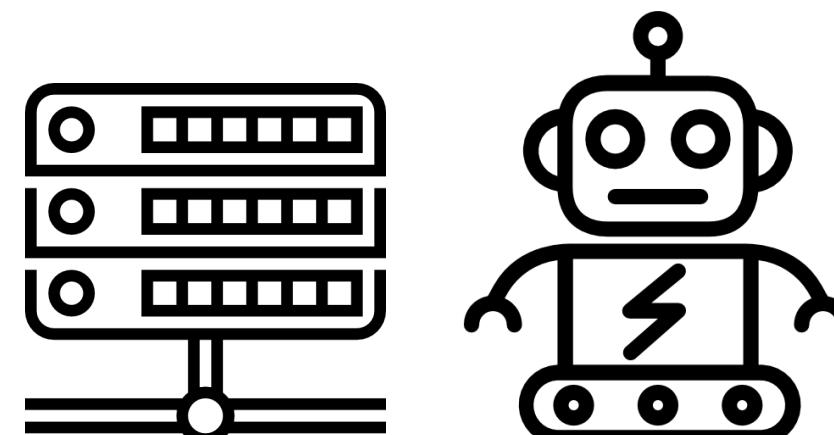


Как пройти аутентификацию?

ХО



LDAP



?



Auth Methods

• Overview

AppRole

AliCloud

AWS

Azure

Cloud Foundry

GitHub

Google Cloud

› JWT/OIDC

Kerberos

Kubernetes

LDAP

› Login MFA

Oracle Cloud Infrastructure

Okta

RADIUS

TLS Certificates

Tokens

Username & Password

Kubernetes

AliCloud

LDAP

TLS Certificates

AWS

Cloud Foundry

GitHub

Tokens

Google Cloud

JWT/OIDC

Username & Password

Oracle Cloud Infrastructure

AppRole

RADIUS



Auth Methods

Overview

AppRole

AliCloud

AWS

Azure

Cloud Foundry

GitHub

Google Cloud

JWT/OIDC

Kerberos

Kubernetes

LDAP

Login MFA

Oracle Cloud Infrastructure

Okta

RADIUS

TLS Certificates

Tokens

Username & Password

Kubernetes

AliCloud

LDAP

Tokens

TLS Certificates

AWS

Cloud Foundry

GitHub

Google Cloud

Username & Password

AppRole

RADIUS

JWT/OIDC

JWT: json web token

Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ  
JoZWxsbyI6IndvcmxkIiwiZGF0YSI6InNvbWUgZ  
GF0YSIsImZvbyI6ImJhciJ9.0mtc-  
LssSGdoVj9Tzx1F2J3f7-guDvgWaGaKtGB3D_U
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{  
  "alg": "HS256",  
  "typ": "JWT"  
}
```

PAYOUT: DATA

```
{  
  "hello": "world",  
  "data": "some data",  
  "foo": "bar"  
}
```

VERIFY SIGNATURE

```
HMACSHA256(  
  base64UrlEncode(header) + "." +  
  base64UrlEncode(payload),  
  very-secure-key  
)  secret base64 encoded
```

JWT: json web token

Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ  
JoZWxsbyI6IndvcmxkIiwiZGF0YSI6InNvbWUgZ  
GF0YSIsImZvbyI6ImJhciJ9.0mtc-  
LssSGdoVj9Tzx1F2J3f7-guDvgWaGaKtGB3D_U
```

Decoded EDIT THE PAYLOAD AND SECRET

```
HEADER: ALGORITHM & TOKEN TYPE
{
  "alg": "HS256",
  "typ": "JWT"
}

PAYLOAD: DATA
{
  "hello": "world",
  "data": "some data",
  "foo": "bar"
}

VERIFY SIGNATURE
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  very-secure-key
)  secret base64 encoded
```

Заголовок

JWT: json web token

Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ  
JoZWxsbyI6IndvcmxkIiwiZGF0YSI6InNvbWUgZ  
GF0YSIsImZvbyI6ImJhciJ9.0mtc-  
LssSGdoVj9Tzx1F2J3f7-guDvgWaGaKtGB3D_U
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{  
  "alg": "HS256",  
  "typ": "JWT"  
}
```

Заголовок

PAYOUT: DATA

```
{  
  "hello": "world",  
  "data": "some data",  
  "foo": "bar"  
}
```

Данные (payload)

VERIFY SIGNATURE

```
HMACSHA256(  
  base64UrlEncode(header) + "." +  
  base64UrlEncode(payload),  
  very-secure-key  
)  secret base64 encoded
```

JWT: json web token

Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ  
JoZWxsbyI6IndvcmxkIiwiZGF0YSI6InNvbWUgZ  
GF0YSIsImZvbyI6ImJhciJ9.0mtc-  
LssSGdoVj9Tzx1F2J3f7-guDvgWaGaKtGB3D_U
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{  
  "alg": "HS256",  
  "typ": "JWT"  
}
```

Заголовок

PAYOUT: DATA

```
{  
  "hello": "world",  
  "data": "some data",  
  "foo": "bar"  
}
```

Данные (payload)

VERIFY SIGNATURE

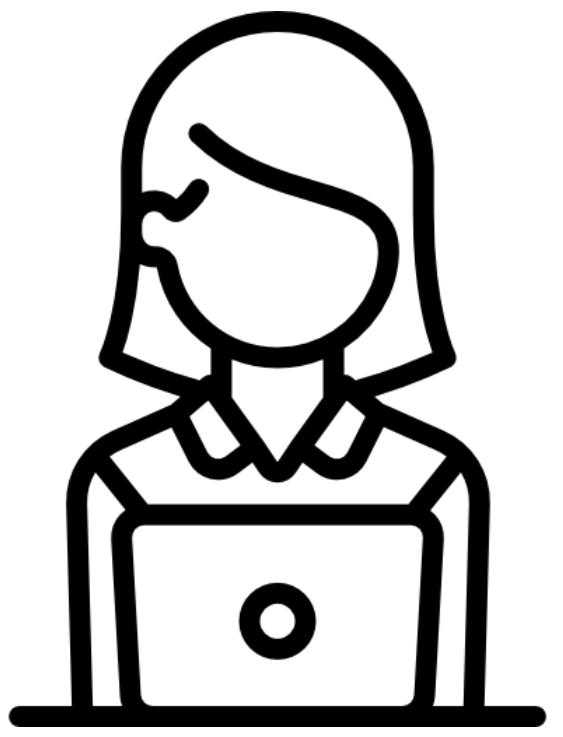
```
HMACSHA256(  
  base64UrlEncode(header) + "." +  
  base64UrlEncode(payload),  
  very-secure-key  
)  secret base64 encoded
```

Подпись

JWT: json web token

ХО

Алиса

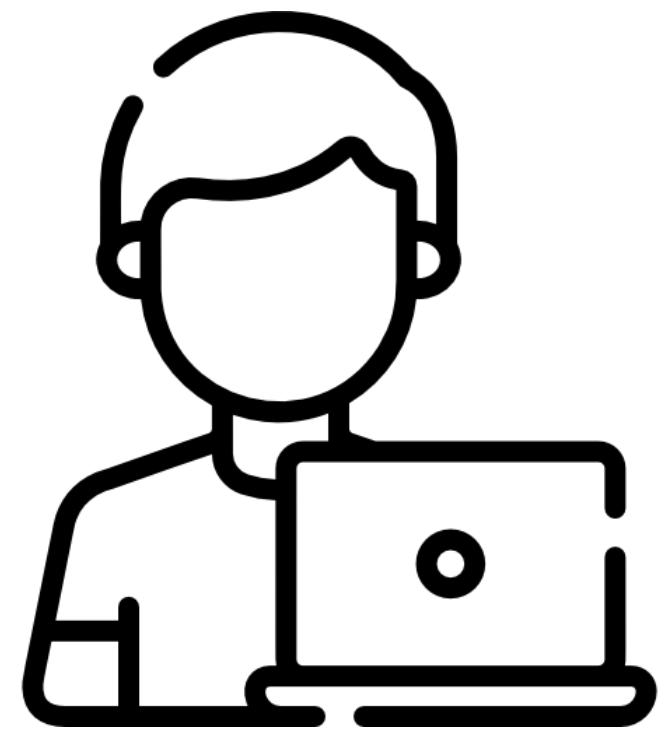


write data
sign with **private key**



read data
check sign with **public key**

Боб

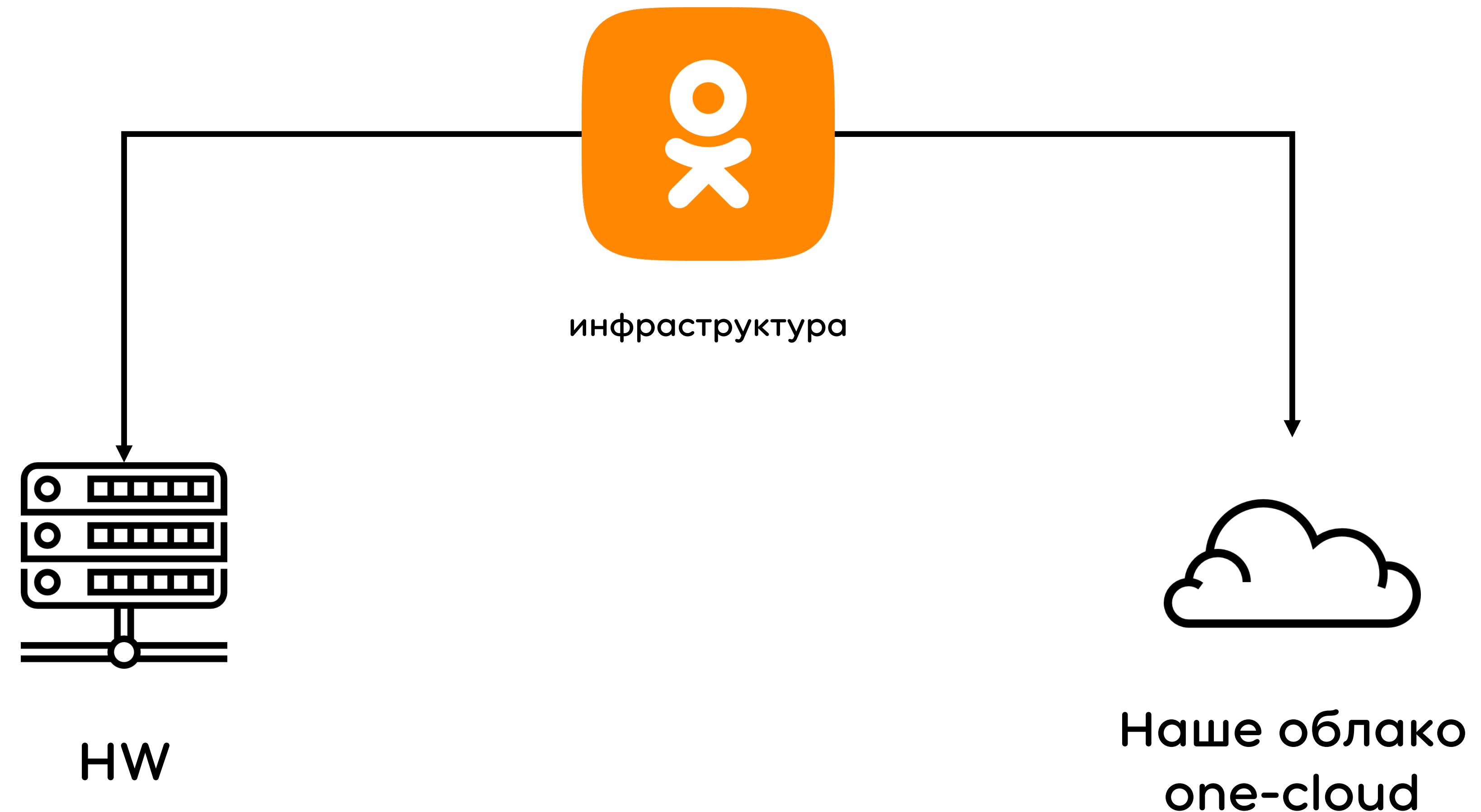


```
{  
  "data": {  
    "name": "Ivan",  
    "Age": 31  
  }  
}
```

```
{  
  "data": {  
    "name": "Ivan",  
    "Age": 31  
  }  
}
```



Посмотрим по сторонам



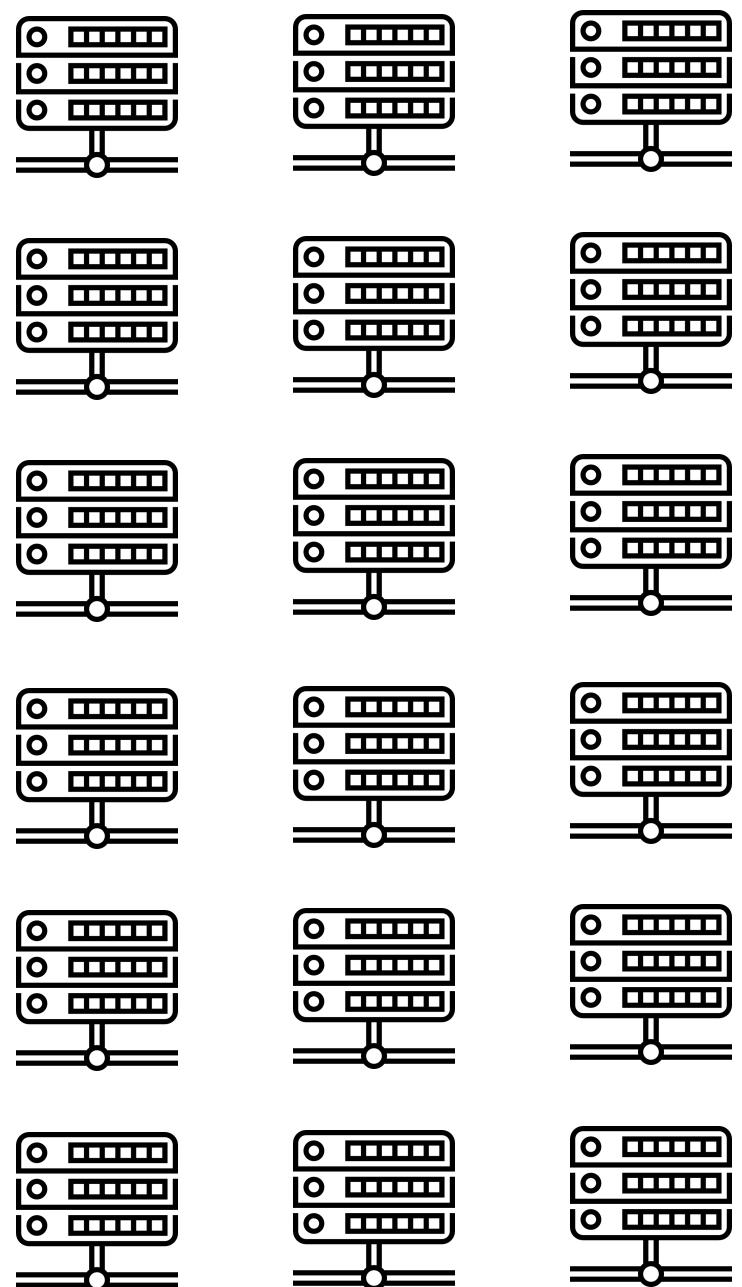
Посмотрим по сторонам: HW

ХО

HW

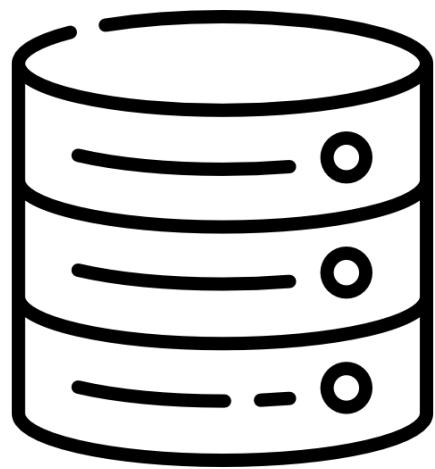
Посмотрим по сторонам: HW

HW

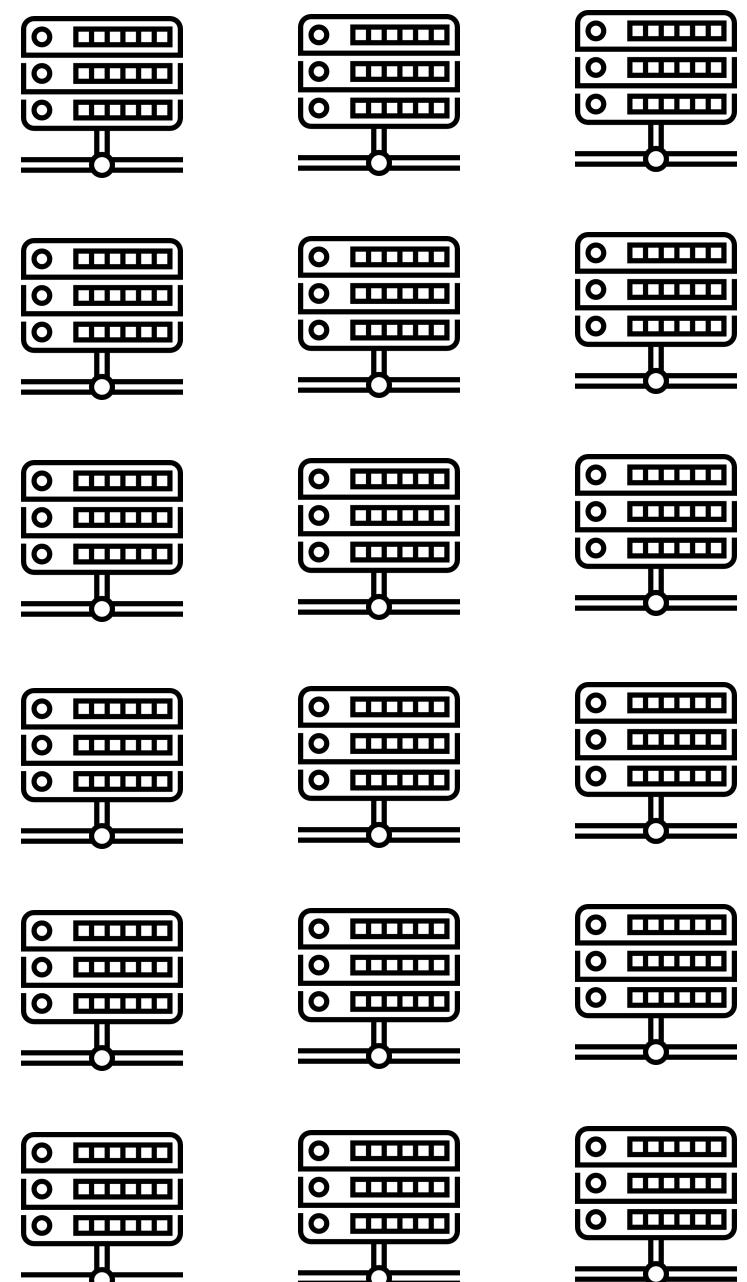


Посмотрим по сторонам: HW

HW



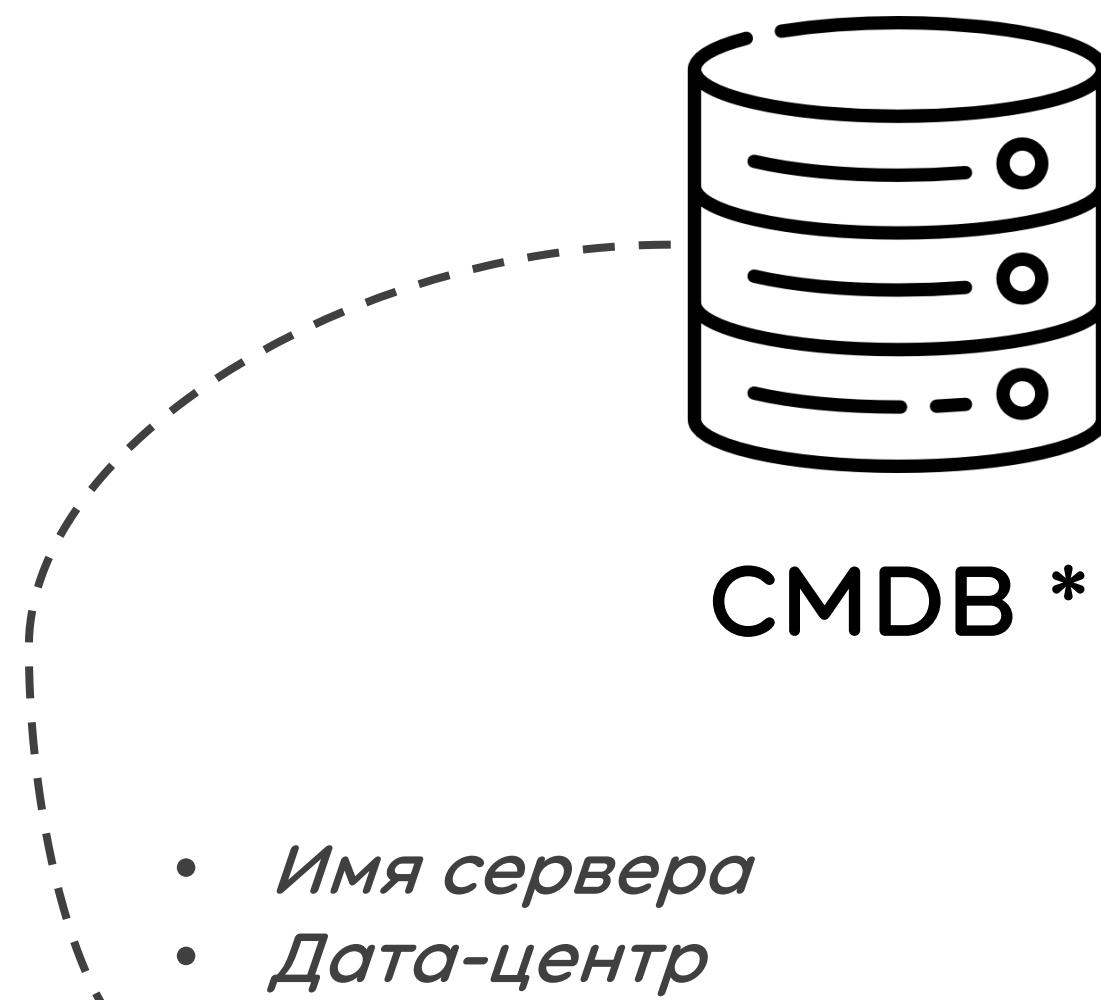
CMDB *



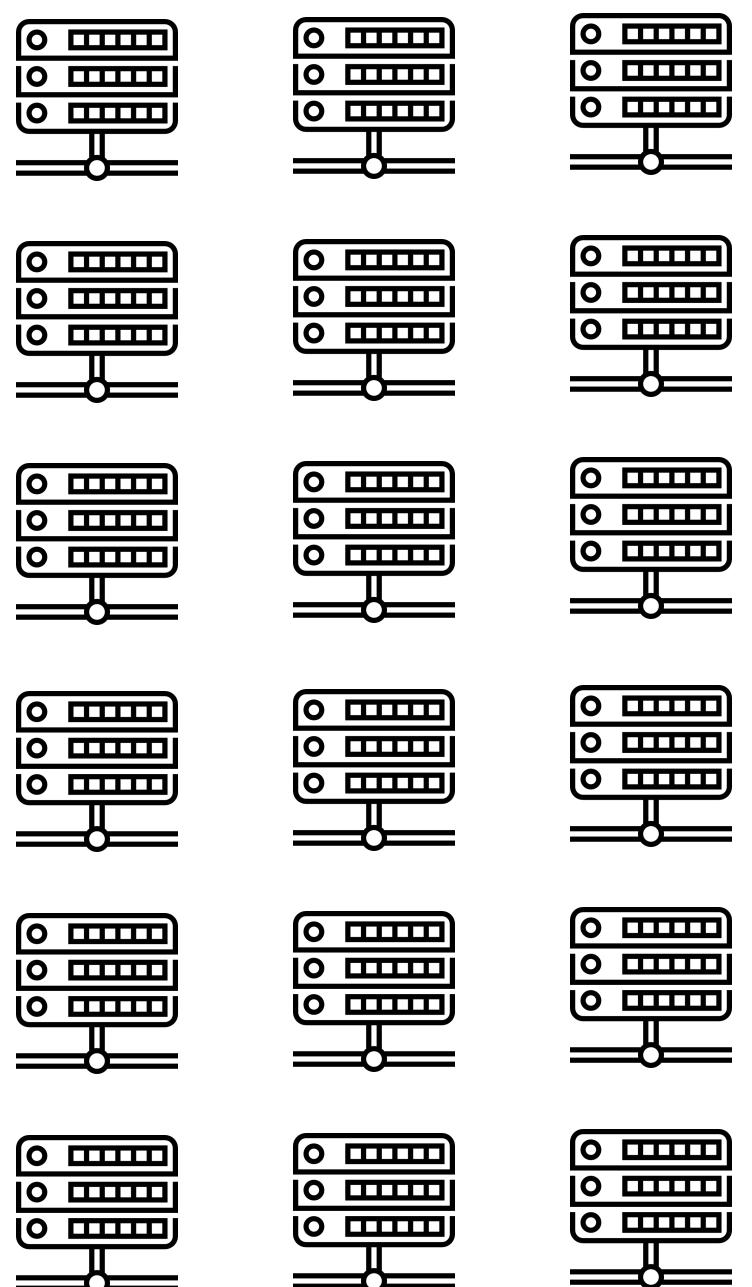
* Configuration management database

Посмотрим по сторонам: HW

HW



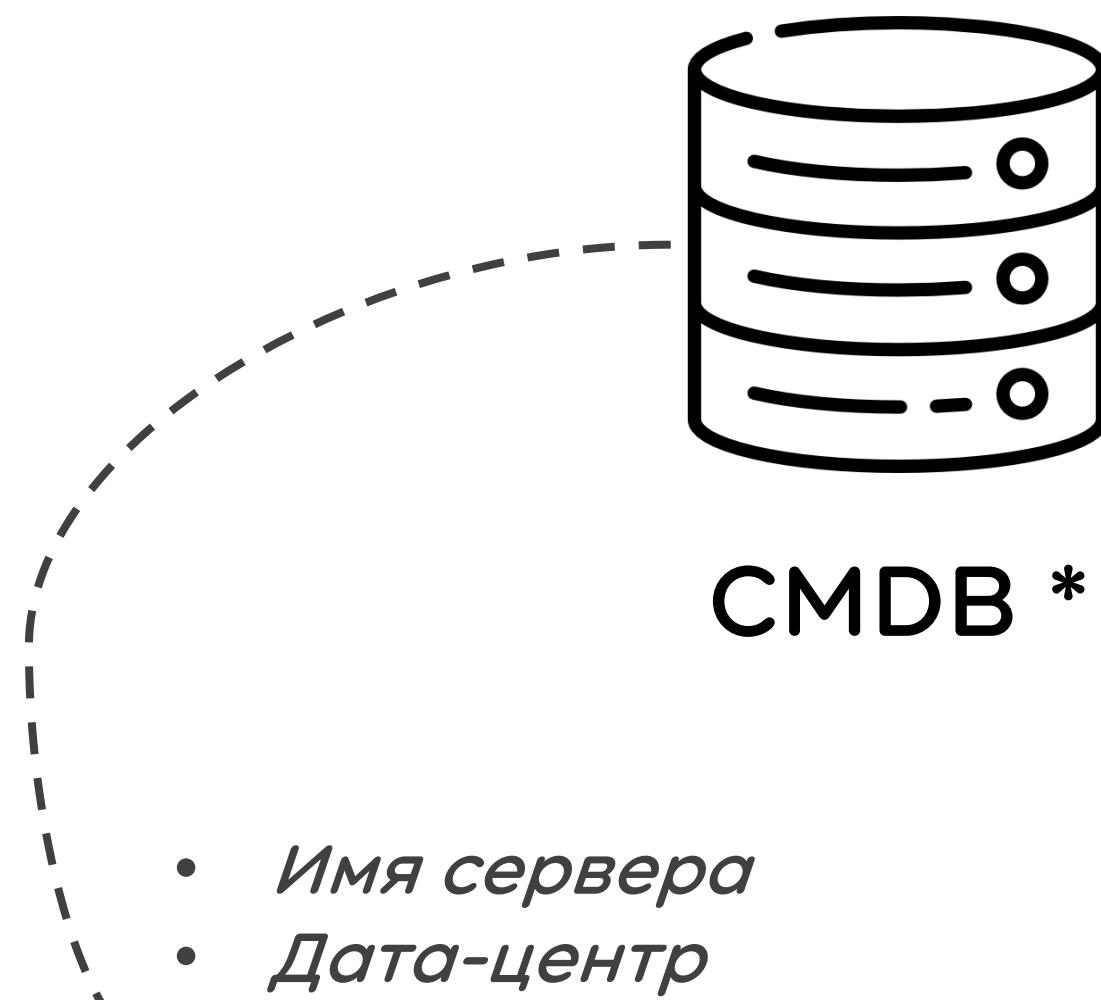
- Имя сервера
- Дата-центр
- Настройки сети
- Группа (*cloud-minion, cdb, video-download*)



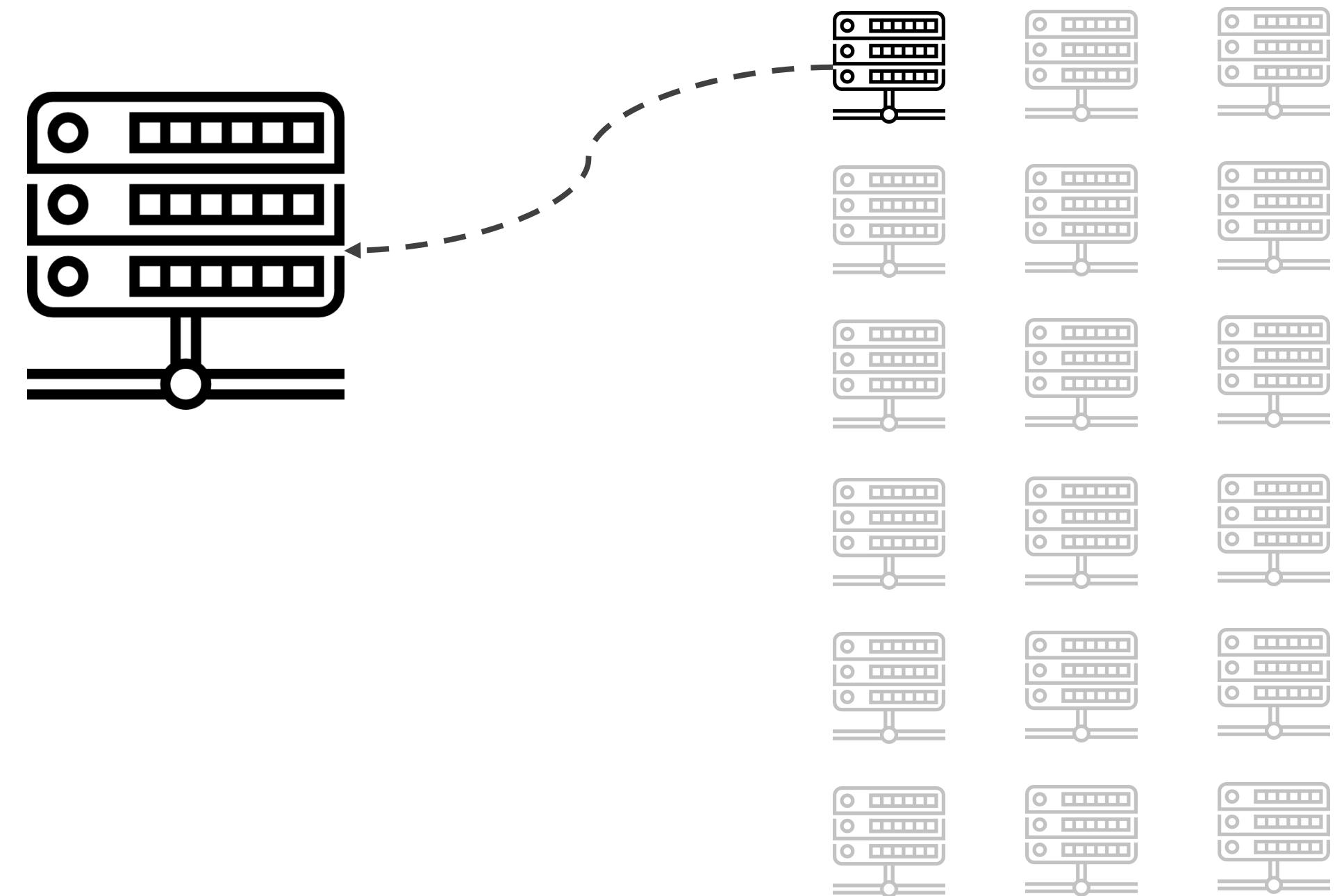
* Configuration management database

Посмотрим по сторонам: HW

HW

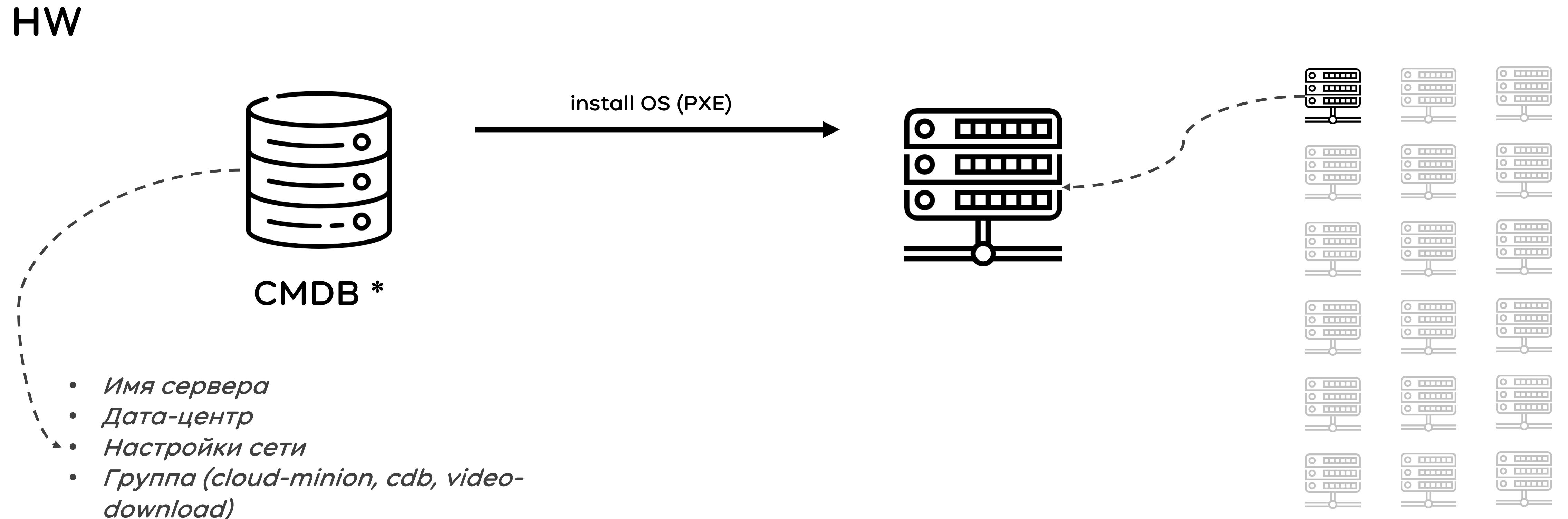


- Имя сервера
- Дата-центр
- Настройки сети
- Группа (*cloud-minion, cdb, video-download*)



* Configuration management database

Посмотрим по сторонам: HW

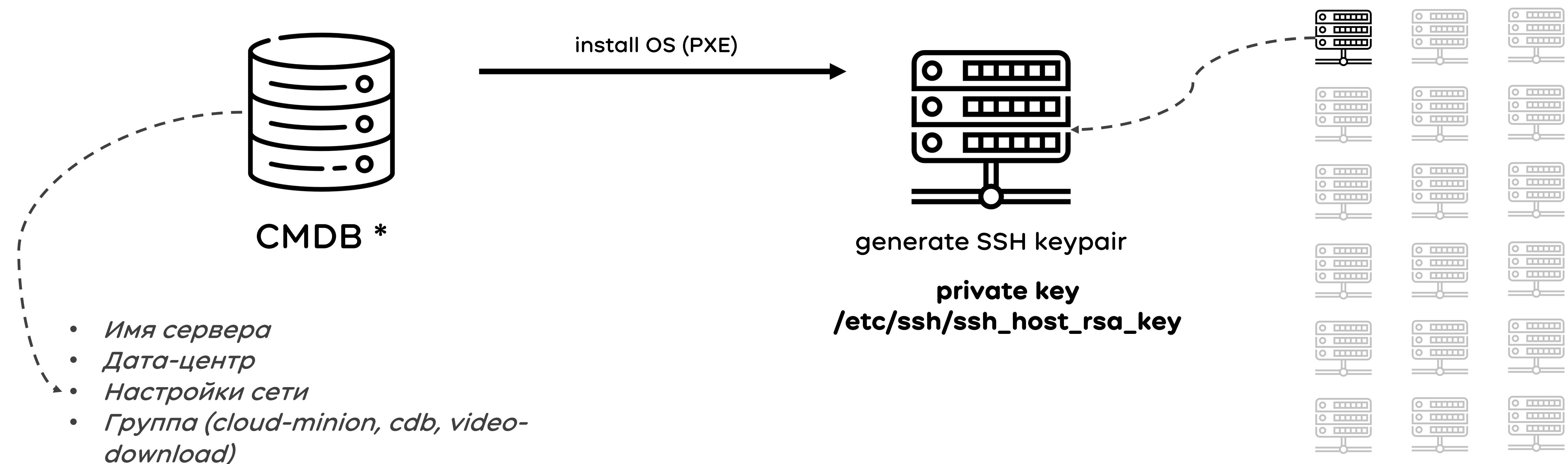


* Configuration management database

Посмотрим по сторонам: HW



HW

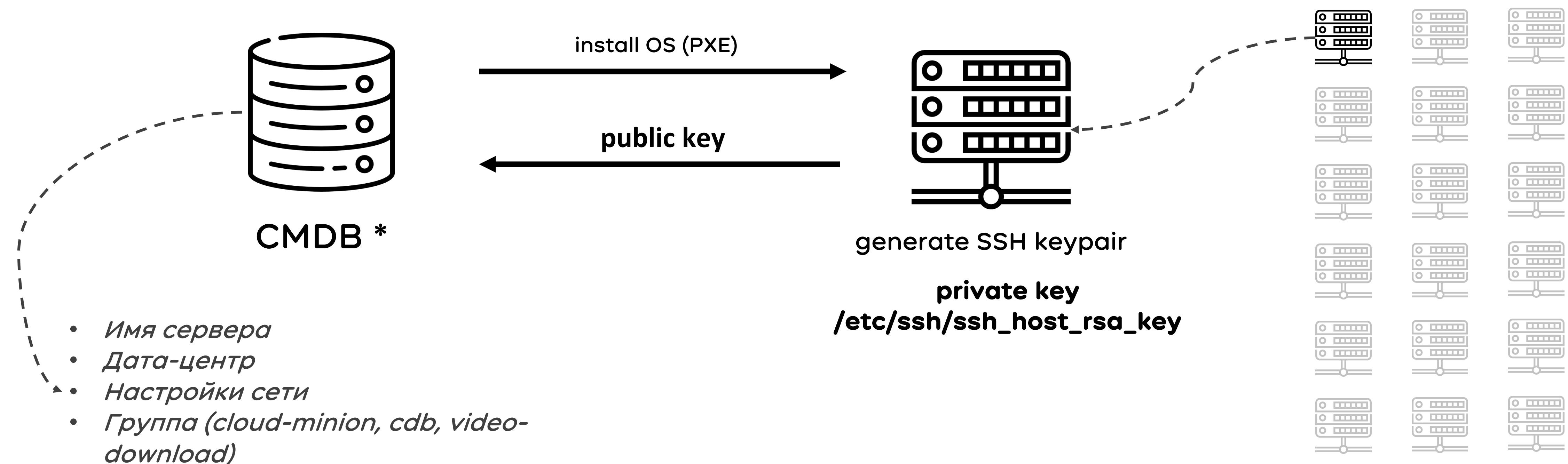


* Configuration management database

Посмотрим по сторонам: HW



HW



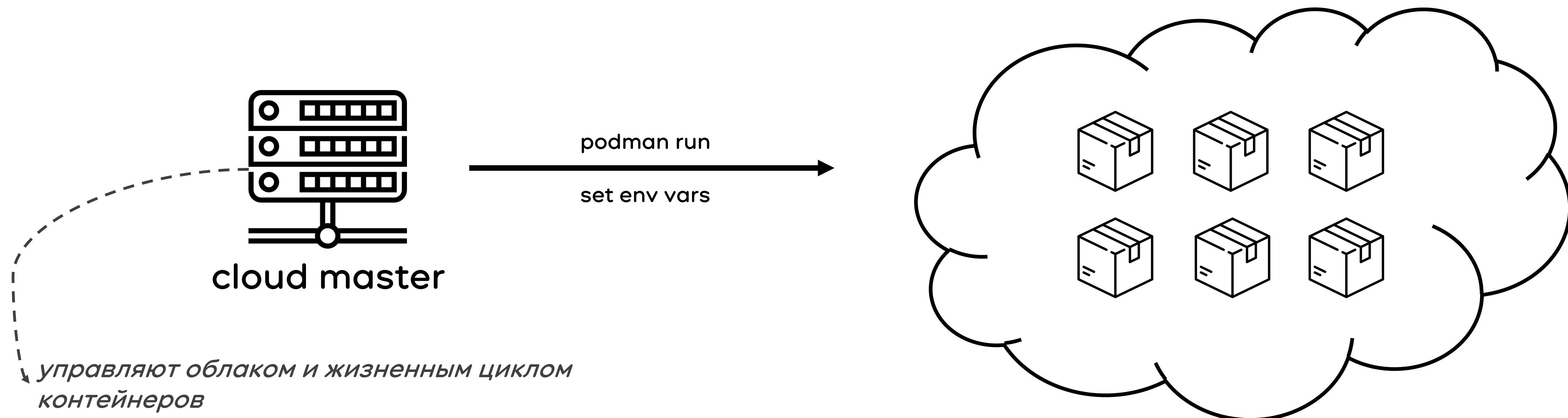
* Configuration management database

Посмотрим по сторонам: Облако



One-cloud

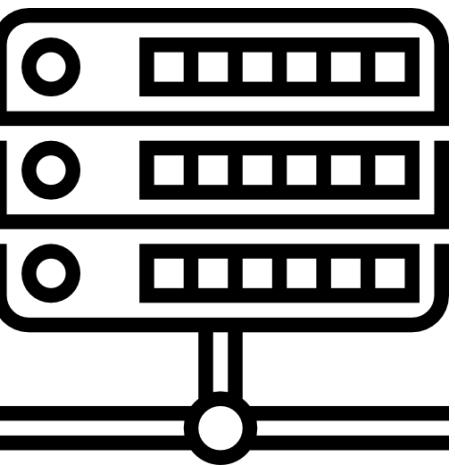
- У контейнеров нет процедуры деплоя ОС
- В контейнерах нет приватного ключа (готового секрета)



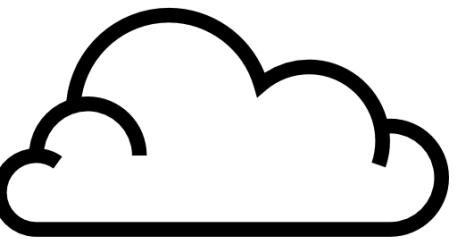
Есть все для JWT!



- На всех серверах есть **приватные ключи**
- Все **публичные ключи** есть в CMDB
- **Полезная нагрузка** – информация из CMDB



HW-сервер может сам создать свой JWT



Мастер облака – тоже HW-сервер

Создает JWT для контейнеров и кладет в ENV при запуске контейнера

Есть все для JWT!

ХО

- Секрет для доступа к секретам – **JWT**
- Доверенные источники – **CMDB** и мастера **One-cloud**
- Нет новых сущностей
- Не нужно поддерживать новые решения





Auth Methods

Overview

AppRole

AliCloud

AWS

Azure

Cloud Foundry

GitHub

Google Cloud

JWT/OIDC

Kerberos

Kubernetes

LDAP

Login MFA

Oracle Cloud Infrastructure

Okta

RADIUS

TLS Certificates

Tokens

Username & Password

Kubernetes

AliCloud

LDAP

Tokens

TLS Certificates

AWS

Cloud Foundry

GitHub

Google Cloud

Username & Password

Oracle Cloud Infrastructure

AppRole

JWT/OIDC

RADIUS

JWT в Vault «из коробки»

```
> vault auth enable -path=jwt jwt  
Success! Enabled jwt auth method at: jwt/
```

Три метода проверки подписи:

- **Static Keys** ----->

```
> vault write auth/jwt/config jwt_validation_pubkeys=@key.pub.pem  
Success! Data written to: auth/jwt/config  
> vault read auth/jwt/config  
Key Value  
---  
...  
jwt_validation_pubkeys [-----BEGIN PUBLIC KEY-----  
MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAE4+SFvPw0y0miy/FiT  
T05HnwjpEbSq+7+1q9BFxAkzjgKnIkXk5qxhzXQvRmS4w9ZsskoTZtu  
UI+XX7conJhzCQ==  
-----END PUBLIC KEY-----]  
...
```

- Нужно поддерживать список в vault

JWT в Vault «из коробки»

```
› vault auth enable -path=jwt jwt  
Success! Enabled jwt auth method at: jwt/
```

Три метода проверки подписи:

- Static Keys
- **JWKS (JSON Web Key Set)**

```
› vault write auth/jwt/config jwks_url="http://127.0.0.1:8000"  
Success! Data written to: auth/jwt/config
```

Key	Value
jwks_url	http://127.0.0.1:8000

- Нужно поддерживать список
- Нужно гонять данные по сети (все ключи)

JWT в Vault «из коробки»



Сомнительный механизм проверки подписи:

JWT в Vault «из коробки»

Сомнительный механизм проверки подписи:

```
var valid bool
for _, key := range config.ParsedJWTPubKeys {
    if err := parsedJWT.Claims(key, &claims, &allClaims); err == nil {
        valid = true
        break
    }
}
```

тысячи проходов



JWT в Vault «из коробки»

```
› vault auth enable -path=jwt jwt  
Success! Enabled jwt auth method at: jwt/
```

Три метода проверки подписи:

- Static Keys
- JWKS (JSON Web Key Set)
- **OIDC Discovery**

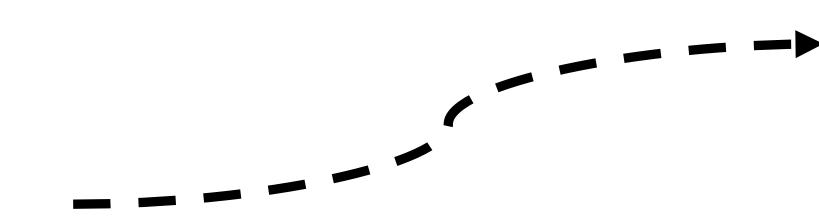
- Рассматривали как основное решение
- Не нужно перебирать ключи
- **Нужно написать свой OIDC-сервер**
- **Нужен умный клиент**

JWT в Vault «из коробки»

```
› vault auth enable -path=jwt jwt  
Success! Enabled jwt auth method at: jwt/
```

Три метода проверки подписи:

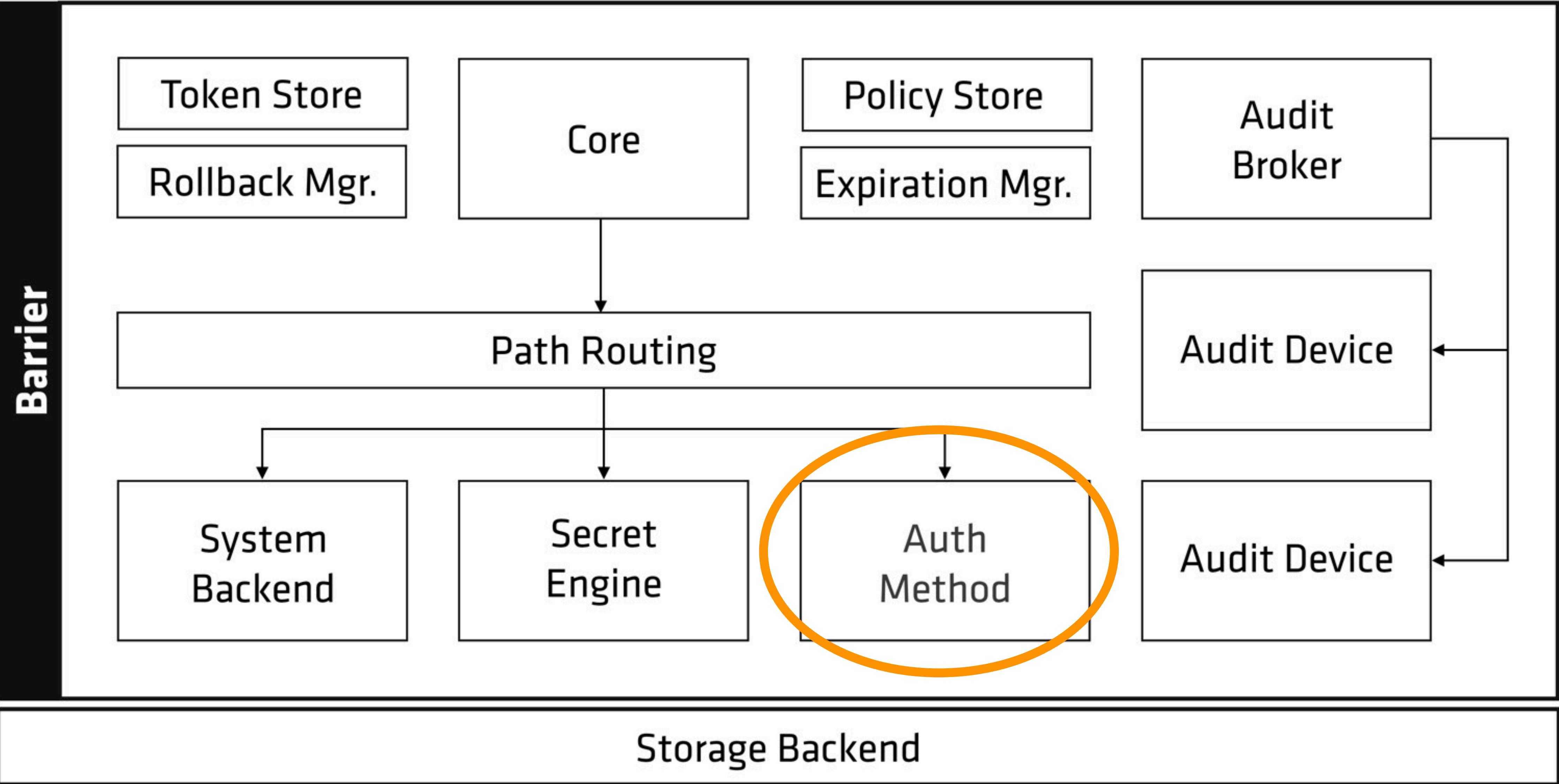
- Static Keys
- JWKS (JSON Web Key Set)
- OIDC Discovery



- Нужно заранее создавать роли
- Набор политик задается в ролях при их создании
- Роли нужно поддерживать



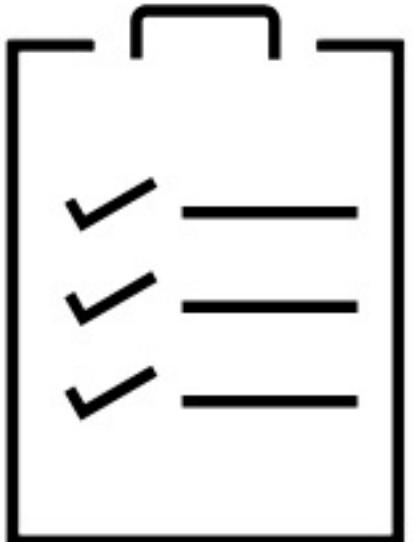
Сделаем аутентификацию сами!



Политики

После аутентификации в Vault клиент получается **Vault-токен**

У Vault-токана есть **метаданные и политики**

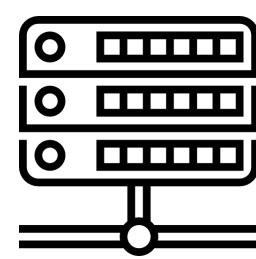


```
› vault token lookup s.wFCv4MhhScDKaSuwhNZRF4yv
```

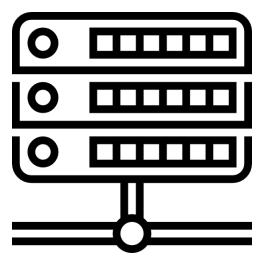
Key	Value
---	-----
...	
accessor	cWwjU76HZsIXhPIdBfUI547X
creation_ttl	24h
meta	map[cmdb_group:web service:web-1 hostname:srvd1234]
display_name	jwt-srvd1234
path	auth/jwt/login
policies	[hw_host, hw_host_srd1234, cmdb_group_web, development]
...	

Политики

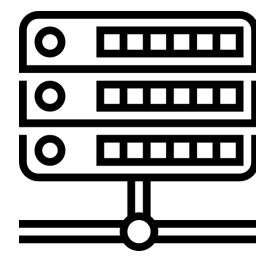
В политиках описаны пути и права доступа



srvd1234

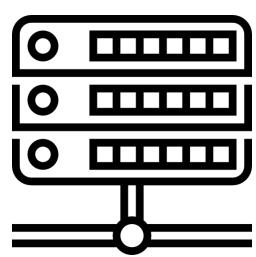


srvd4567



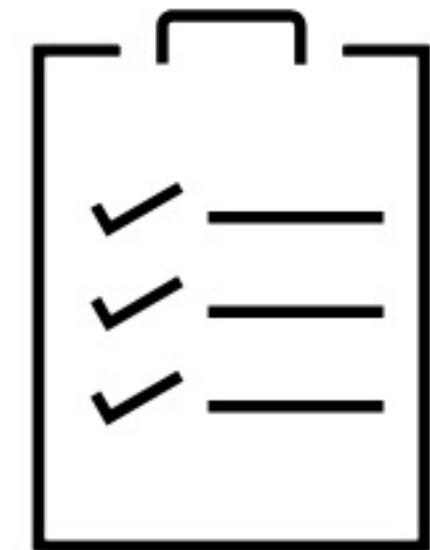
srvd7890

...



srvdXXX

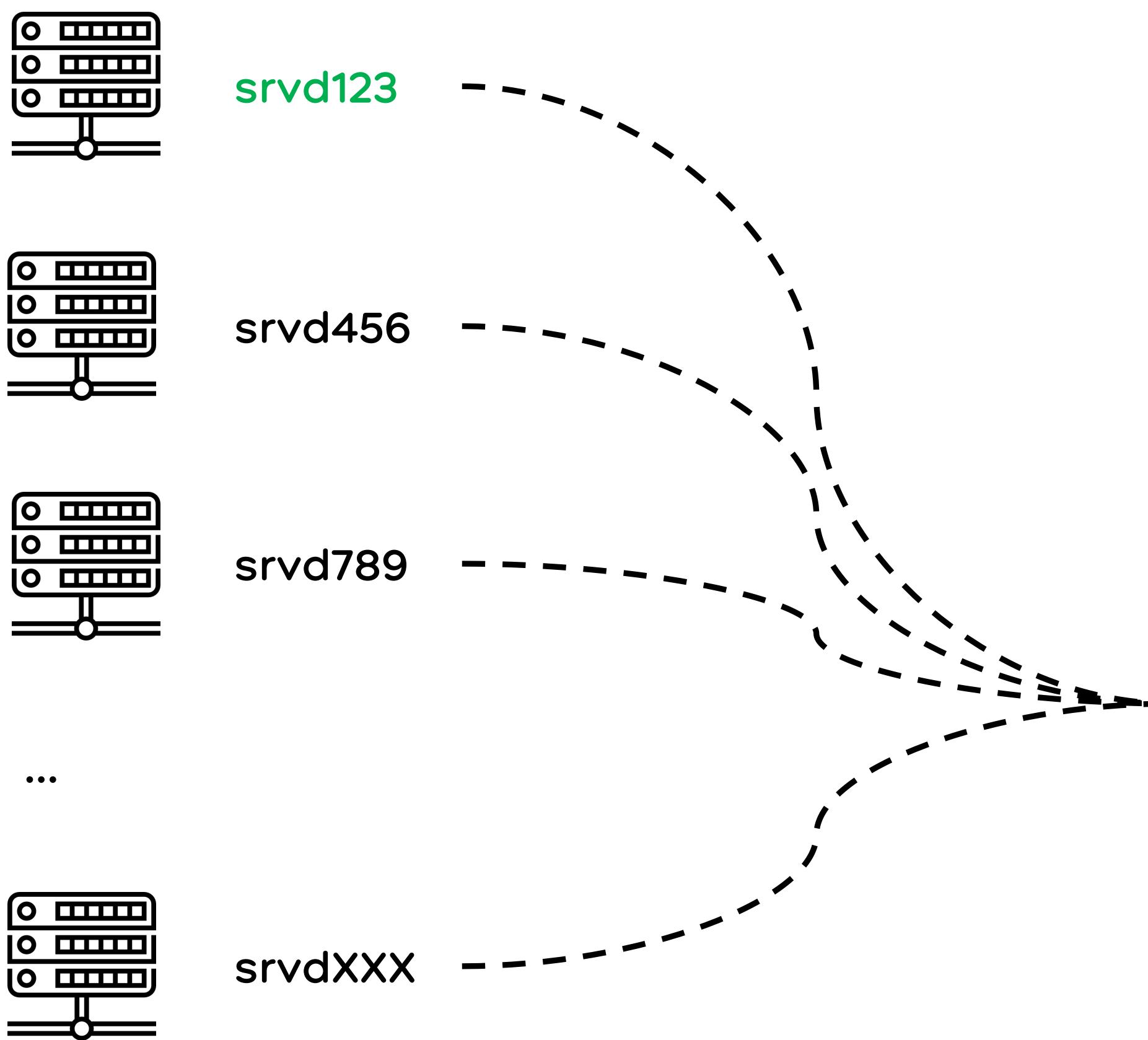
```
> cat policies/srvd1234.yml
---
paths:
  - path: kv/secrets/srvd1234/*
capabilities:
  - read
  - update
```



Писать 100500 политик для каждого хоста?..

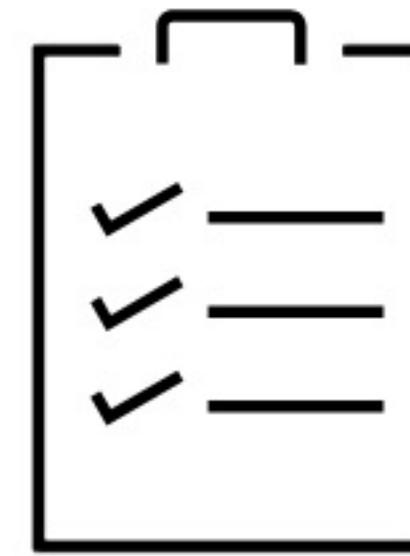
Политики

В политиках можно использовать шаблоны



```
> cat policies/srvd1234.yml
---
paths:
  - path: kv/secrets/srvd1234/*
capabilities:
  - read
  - update

> cat policies/hw_host.yml
---
paths:
  - path: kv/secrets/{{ hostname }}/*
capabilities:
  - read
  - update
```



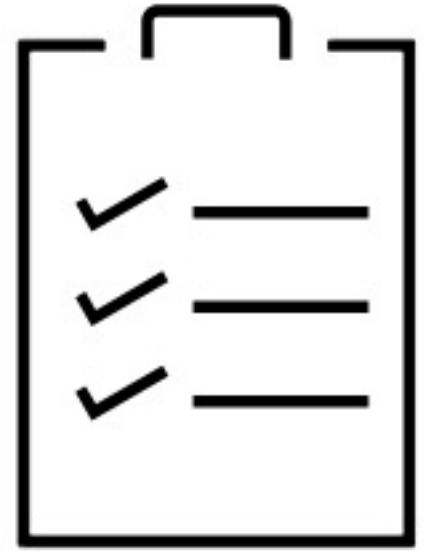
Политики

В политиках можно использовать шаблоны

```
> cat policies/hw_host.yml
---
paths:
  - path: kv/secrets/{{ hostname }}/+
capabilities:
  - read
  - update

> vault token lookup s.wFCv4MhhScDKaSuwhNZRF4yv

Key          Value
---
...
accessor    cWwjU76HZsIXhPIdBfUI547X
creation_ttl 24h
meta        map[cmdb_group:web service:web-1 hostname:srvd1234]
display_name jwt-srvd1234
path        auth/jwt/login
policies   [hw_host, hw_host_srd1234, cmdb_group_web, development]
...
```



Политики

HCL

```
> cat development.hcl  
  
path "kv/dev/users/+" {  
    capabilities = [  
        "read",  
        "update",  
    ]  
}  
  
path "kv/users/{{ service }}" {  
    capabilities = [  
        "read",  
    ]  
}
```



YAML

```
> cat development.yaml  
  
- path: kv/dev/users/+  
  capabilities:  
    - read  
    - update  
  
- path: kv/users/{{ service }}
```

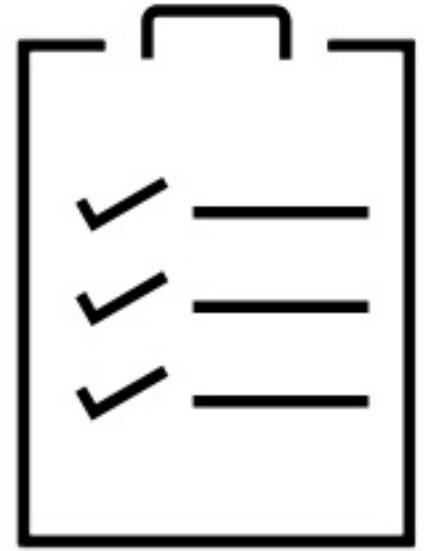
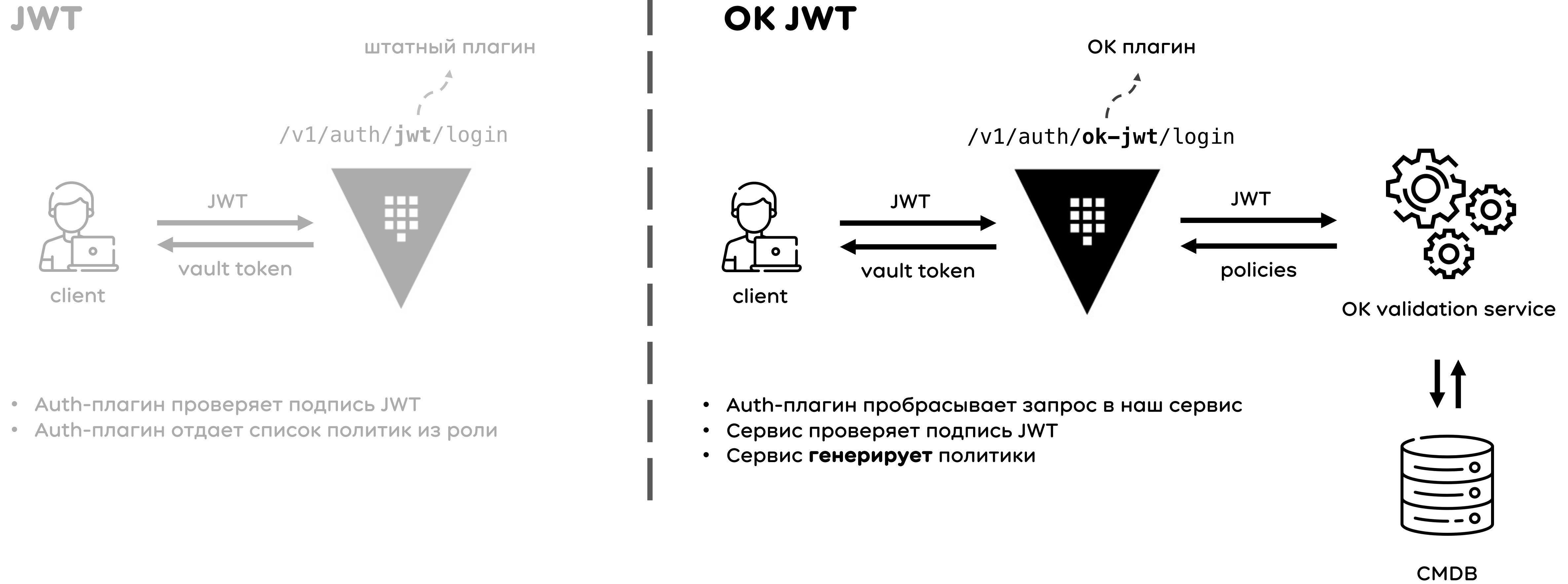


Схема аутентификации



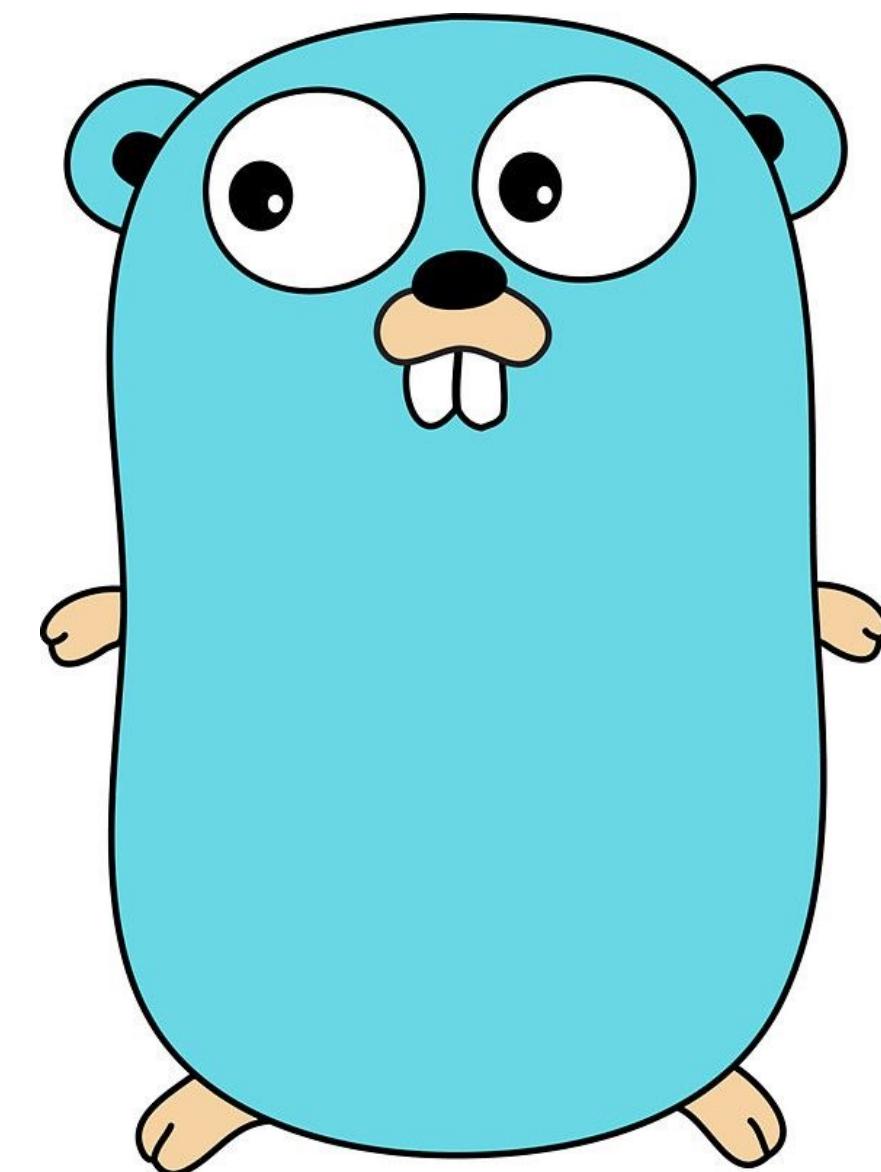
Аутентификация: OK JWT

OK

Auth-плагин для Vault

```
> vault write sys/plugins/catalog/auth/ok-jwt command="ok-jwt" sha_256=$SHA  
> vault auth enable -path=ok-jwt ok-jwt
```

- Бинарный файл (Go)
- Vault проверяет SHA плагинов
- Основан на [hashicorp/vault-plugin-auth-jwt](#)
- Проксирует запросы в наш сервис валидации
- Добавляет в токен **политики и метаданные**

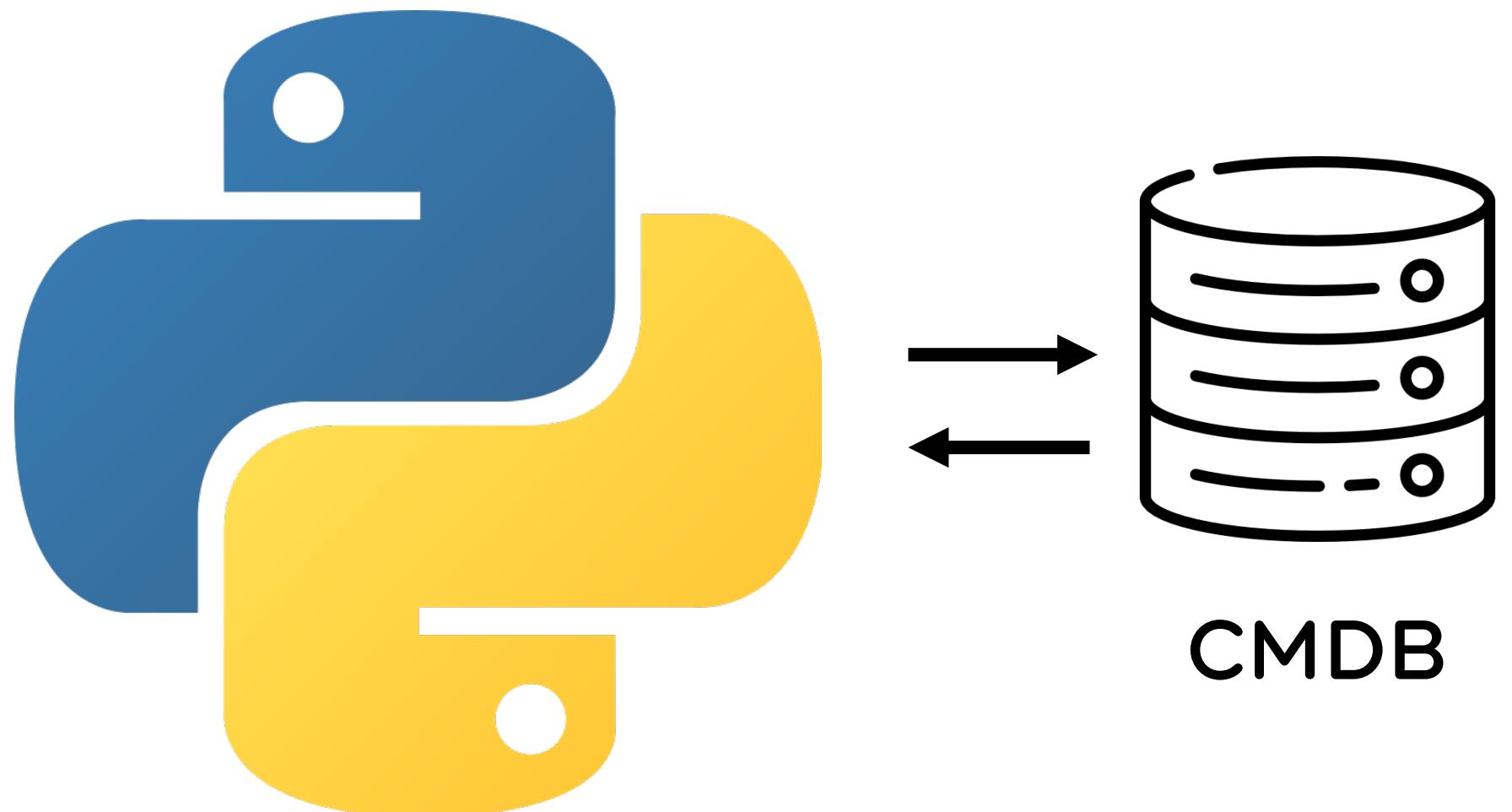


Аутентификация: OK JWT

Х

Сервис для валидации

- Python
- Проверяет подпись JWT
- Сверяет данные с информацией в CMDB
- Генерирует политики по нашим правилам
- Отдает список политик в Vault (наш плагин)



Аутентификация: OK JWT



```
func validateByOKService(token, OKAuthServiceURL string) ([]string, error) {  
  
    type response struct {  
        Policies []string `json:"policies"  
        Message  string   `json:"message"  
    }  
    var respData response  
    log.Printf("Send JWT for validation to: %s", OKAuthServiceURL)  
  
    var jsonStr = []byte(fmt.Sprintf(`{"JWT": "%s"}`, token))  
  
    req, err := http.NewRequest("POST", OKAuthServiceURL, bytes.NewBuffer(jsonStr))  
    if err != nil {  
        log.Printf("Can't create http request: %s", err)  
        return nil, err  
    }  
    ...  
    return respData.Policies  
}
```

Аутентификация: OK JWT



```
func validateByOKService(token, OKAuthServiceURL string) ([]string, error) {  
  
    type response struct {  
        Policies []string `json:"policies"  
        Message  string   `json:"message"  
    }  
    var respData response  
    log.Printf("Send JWT for validation to: %s", OKAuthServiceURL)  
  
    var jsonStr = []byte(fmt.Sprintf(`{"JWT": "%s"}`, token))  
  
    req, err := http.NewRequest("POST", OKAuthServiceURL, bytes.NewBuffer(jsonStr))  
    if err != nil {  
        log.Printf("Can't create http request: %s", err)  
        return nil, err  
    }  
    ...  
    return respData.Policies  
}
```

Аутентификация: OK JWT



```
func validateByOKService(token, OKAuthServiceURL string) ([]string, error) {  
  
    type response struct {  
        Policies []string `json:"policies"  
        Message  string   `json:"message"  
    }  
    var respData response  
    log.Printf("Send JWT for validation to: %s", OKAuthServiceURL)  
  
    var jsonStr = []byte(fmt.Sprintf(`{"JWT": "%s"}`, token))  
  
    req, err := http.NewRequest("POST", OKAuthServiceURL, bytes.NewBuffer(jsonStr))  
    if err != nil {  
        log.Printf("Can't create http request: %s", err)  
        return nil, err  
    }  
    ...  
    return respData.Policies  
}
```



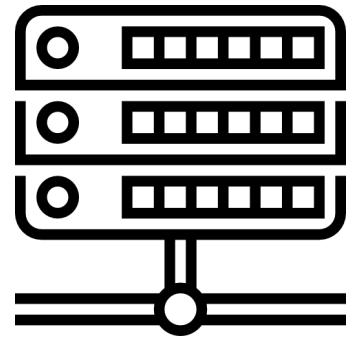
Откуда берется JWT?

Библиотека для работы с Vault



- Python-библиотека
- Генерация JWT, аутентификация, продление токенов
- Ставим на все сервера и в контейнеры
- Используется повсеместно: Ansible (lookup plugin), CFEngine, скрипты и т.д.

Откуда берется JWT?



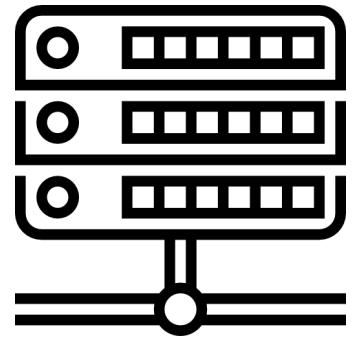
HW

Сервер с помощью `ok-pyvault` генерирует JWT и подписывает его своим приватным ключом: `/etc/ssh/ssh_host_rsa_key`

```
[root@srvd1234 ~]# vault-login -v
DEBUG: Generating JWT and sign it by ssh private key.
DEBUG: Read private key from /etc/ssh/ssh_host_rsa_key
DEBUG: JWT generated. Payload: {'group': 'development',
'aud': 'hw', 'service': 'maintanence', 'iss': 'srvd1234',
'exp': 1636627489, 'iat': 1636626889, 'nbf': 1636626889}
DEBUG: "POST /v1/auth/jwt/login HTTP/1.1" 200 642
DEBUG: Logged to Vault via JWT. Policies:
['cmdb_group_development', 'cmdb_service_maintanence',
'default', 'hw_host', 'hw_host_srvd1234']
INFO: Write file at '/root/.vault-token'
INFO: Authenticated in https://dl.vault.ok.ru
```

```
[root@srvu1234 ~]# cat .vault-token
s.nnKD0PGpGt987Nb6lvQ0IbXU
```

Откуда берется JWT?



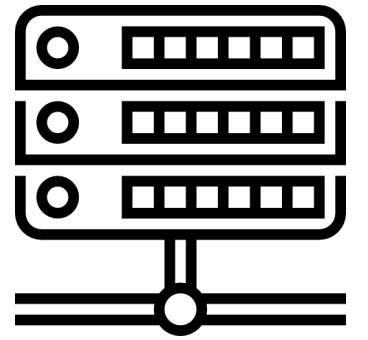
HW

Сервер с помощью `ok-pyvault` генерирует JWT и подписывает его своим приватным ключом: `/etc/ssh/ssh_host_rsa_key`

```
[root@srvd1234 ~]# vault-login -v
DEBUG: Generating JWT and sign it by ssh private key.
DEBUG: Read private key from /etc/ssh/ssh_host_rsa_key
DEBUG: JWT generated. Payload: {'group': 'development',
'aud': 'hw', 'service': 'maintanence', 'iss': 'srvd1234',
'exp': 1636627489, 'iat': 1636626889, 'nbf': 1636626889}
DEBUG: "POST /v1/auth/jwt/login HTTP/1.1" 200 642
DEBUG: Logged to Vault via JWT. Policies:
['cmdb_group_development', 'cmdb_service_maintanence',
'default', 'hw_host', 'hw_host_srvd1234']
INFO: Write file at '/root/.vault-token'
INFO: Authenticated in https://dl.vault.ok.ru
```

```
[root@srvu1234 ~]# cat .vault-token
s.nnKD0PGpGt987Nb6lvQ0IbXU
```

Откуда берется JWT?



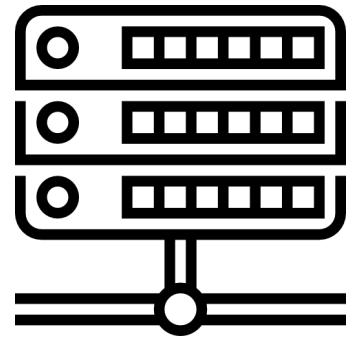
HW

Сервер с помощью ok-pyvault генерирует JWT и подписывает его своим приватным ключом: **/etc/ssh/ssh_host_rsa_key**

```
[root@srvd1234 ~]# vault-login -v
DEBUG: Generating JWT and sign it by ssh private key.
DEBUG: Read private key from /etc/ssh/ssh_host_rsa_key
DEBUG: JWT generated. Payload: {"group": "development",
'aud': 'hw', 'service': 'maintanence', 'iss': 'srvd1234',
'exp': 1636627489, 'iat': 1636626889, 'nbf': 1636626889}
DEBUG: "POST /v1/auth/jwt/login HTTP/1.1" 200 642
DEBUG: Logged to Vault via JWT. Policies:
['cmdb_group_development', 'cmdb_service_maintanence',
'default', 'hw_host', 'hw_host_srvd1234']
INFO: Write file at '/root/.vault-token'
INFO: Authenticated in https://dl.vault.ok.ru
```

```
[root@srvu1234 ~]# cat .vault-token
s.nnKD0PGpGt987Nb6lvQ0IbXU
```

Откуда берется JWT?



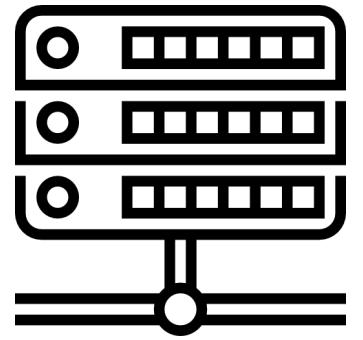
HW

Сервер с помощью ok-pyvault генерирует JWT и подписывает его своим приватным ключом: **/etc/ssh/ssh_host_rsa_key**

```
[root@srvd1234 ~]# vault-login -v
DEBUG: Generating JWT and sign it by ssh private key.
DEBUG: Read private key from /etc/ssh/ssh_host_rsa_key
DEBUG: JWT generated. Payload: {'group': 'development',
'aud': 'hw', 'service': 'maintanence', 'iss': 'srvd1234',
'exp': 1636627489, 'iat': 1636626889, 'nbf': 1636626889}
DEBUG: "POST /v1/auth/jwt/login HTTP/1.1" 200 642
DEBUG: Logged to Vault via JWT. Policies:
['cmdb_group_development', 'cmdb_service_maintanence',
'default', 'hw_host', 'hw_host_srvd1234']
INFO: Write file at '/root/.vault-token'
INFO: Authenticated in https://dl.vault.ok.ru
```

```
[root@srvu1234 ~]# cat .vault-token
s.nnKD0PGpGt987Nb6lvQ0IbXU
```

Откуда берется JWT?



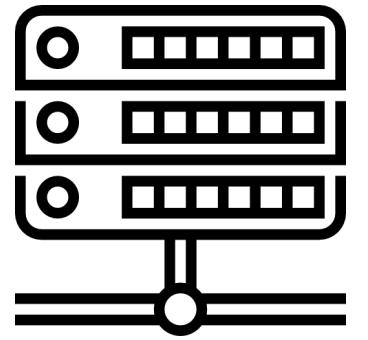
HW

Сервер с помощью `ok-pyvault` генерирует JWT и подписывает его своим приватным ключом: `/etc/ssh/ssh_host_rsa_key`

```
[root@srvd1234 ~]# vault-login -v
DEBUG: Generating JWT and sign it by ssh private key.
DEBUG: Read private key from /etc/ssh/ssh_host_rsa_key
DEBUG: JWT generated. Payload: {'group': 'development',
'aud': 'hw', 'service': 'maintanence', 'iss': 'srvd1234',
'exp': 1636627489, 'iat': 1636626889, 'nbf': 1636626889}
DEBUG: "POST /v1/auth/jwt/login HTTP/1.1" 200 642
DEBUG: Logged to Vault via JWT. Policies:
['cmdb_group_development', 'cmdb_service_maintanence',
'default', 'hw_host', 'hw_host_srvd1234']
INFO: Write file at '/root/.vault-token'
INFO: Authenticated in https://dl.vault.ok.ru
```

```
[root@srvu1234 ~]# cat .vault-token
s.nnKD0PGpGt987Nb6lvQ0IbXU
```

Откуда берется JWT?



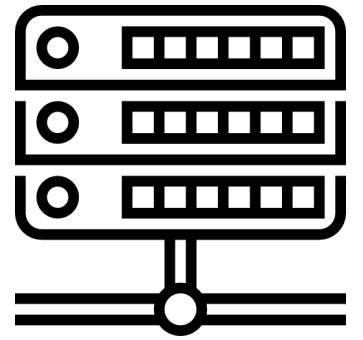
HW

Сервер с помощью `ok-pyvault` генерирует JWT и подписывает его своим приватным ключом: `/etc/ssh/ssh_host_rsa_key`

```
[root@srvd1234 ~]# vault-login -v
DEBUG: Generating JWT and sign it by ssh private key.
DEBUG: Read private key from /etc/ssh/ssh_host_rsa_key
DEBUG: JWT generated. Payload: {'group': 'development',
'aud': 'hw', 'service': 'maintanence', 'iss': 'srvd1234',
'exp': 1636627489, 'iat': 1636626889, 'nbf': 1636626889}
DEBUG: "POST /v1/auth/jwt/login HTTP/1.1" 200 642
DEBUG: Logged to Vault via JWT. Policies:
['cmdb_group_development', 'cmdb_service_maintanence',
'default', 'hw_host', 'hw_host_srvd1234']
INFO: Write file at '/root/.vault-token'
INFO: Authenticated in https://dl.vault.ok.ru
```

```
[root@srvu1234 ~]# cat .vault-token
s.nnKD0PGpGt987Nb6lvQ0IbXU
```

Откуда берется JWT?



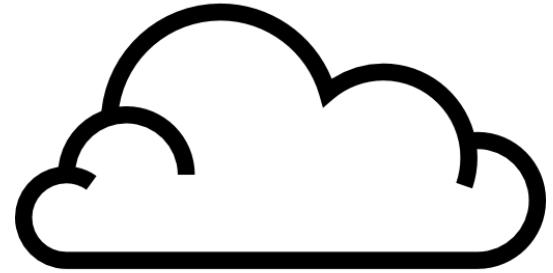
HW

Сервер с помощью `ok-pyvault` генерирует JWT и подписывает его своим приватным ключом: `/etc/ssh/ssh_host_rsa_key`

```
[root@srvd1234 ~]# vault-login -v
DEBUG: Generating JWT and sign it by ssh private key.
DEBUG: Read private key from /etc/ssh/ssh_host_rsa_key
DEBUG: JWT generated. Payload: {'group': 'development',
'aud': 'hw', 'service': 'maintanence', 'iss': 'srvd1234',
'exp': 1636627489, 'iat': 1636626889, 'nbf': 1636626889}
DEBUG: "POST /v1/auth/jwt/login HTTP/1.1" 200 642
DEBUG: Logged to Vault via JWT. Policies:
['cmdb_group_development', 'cmdb_service_maintanence',
'default', 'hw_host', 'hw_host_srvd1234']
INFO: Write file at '/root/.vault-token'
INFO: Authenticated in https://dl.vault.ok.ru
```

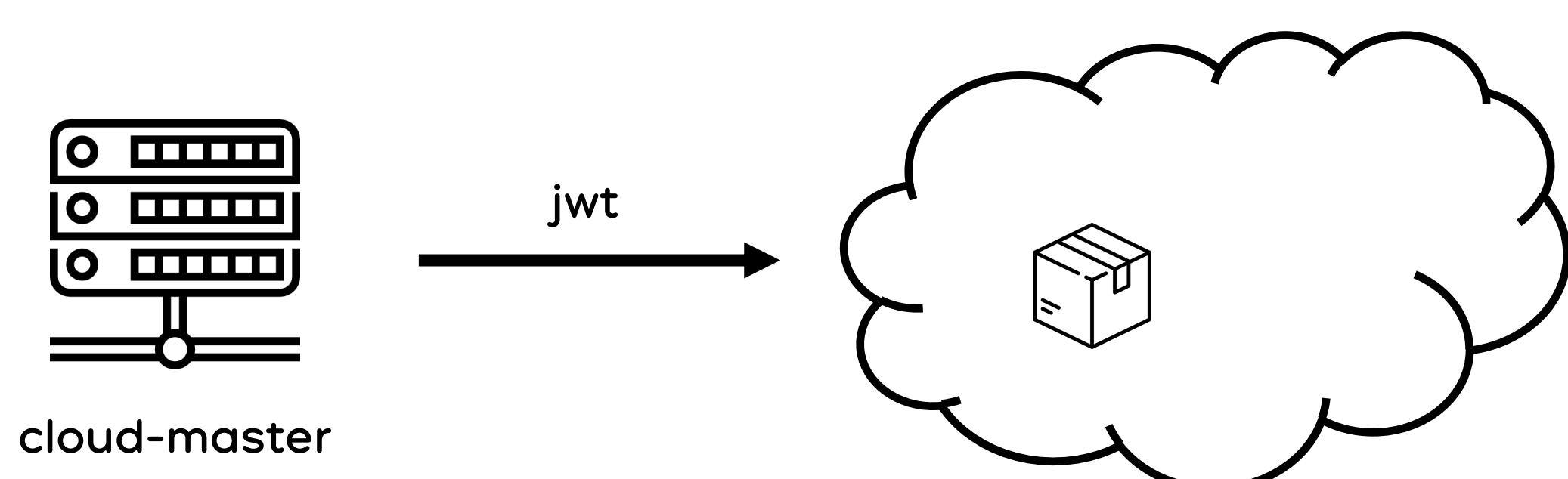
```
[root@srvu1234 ~]# cat .vault-token
s.nnKD0PGpGt987Nb6lvQ0IbXU
```

Откуда берется JWT?



One-cloud

Мастер облака подписывает JWT своим
ключом и кладет в ENV контейнера

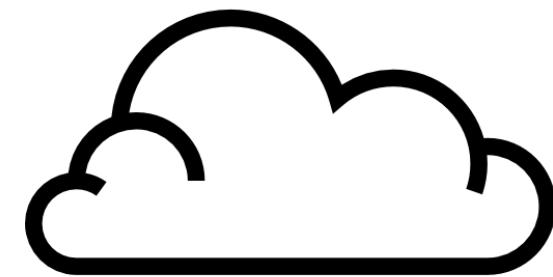


```
1.web.service.dc.ok.ru: /# echo $cloud_jwt_token  
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdWQiOi...jbG91ZCI  
sIm5iZiI6MTYzMjYxOTM5NywiY2xvdWfaG9zdG5hbWUiOixLm1vbmd  
vLmF1dG...90ZXN0LmRjIiwiN30.LeFxQr_3NpIuXY
```

```
1.web.service.dc.ok.ru: /# vault-login -v  
DEBUG: "POST /v1/auth/jwt/login HTTP/1.1" 200 923  
DEBUG: Logged to Vault via JWT. Policies:  
['cloud_instance', 'cloud_dc', 'cloud_web.service',  
'cloud_web.service.dc']  
INFO: Write file at '/root/.vault-token'  
INFO: Authenticated in https://dl.vault.ok.ru
```

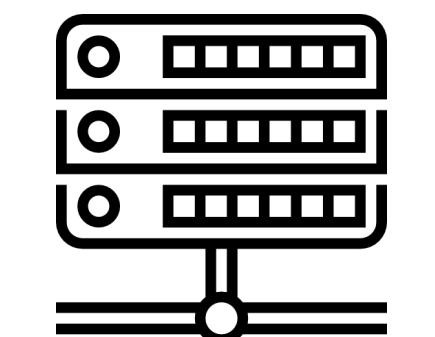
```
1.web.service.dc.ok.ru: /# cat ~/.vault-token  
s.oeX28WXHGoVwIh8FLE1iWgTz
```

Откуда берется JWT?

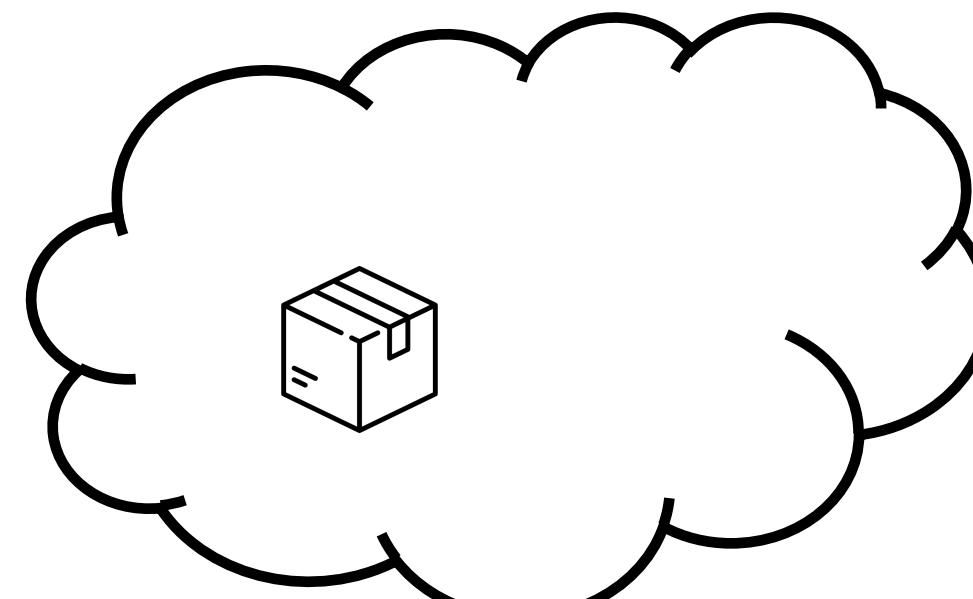


One-cloud

Мастер облака подписывает JWT своим
ключем и кладет в ENV контейнера



jwt

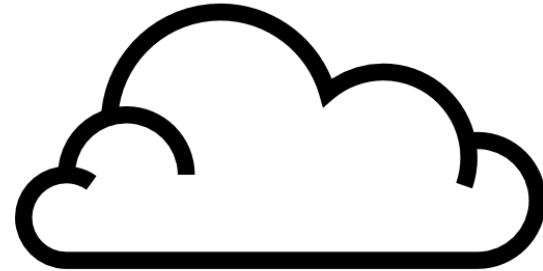


```
1.web.service.dc.ok.ru: /# echo $cloud_jwt_token  
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdWQiO...jbG91ZCI  
sIm5iZii6MTYzNjYxOTM5NywiY2xvdWRfaG9zdG5hbWUiOiIxLm1vbmd  
vLmF1dG...90ZXN0LmRjIiwiN30.LeFxQr_3NpIuXY
```

```
1.web.service.dc.ok.ru: /# vault-login -v  
DEBUG: "POST /v1/auth/jwt/login HTTP/1.1" 200 923  
DEBUG: Logged to Vault via JWT. Policies:  
['cloud_instance', 'cloud_dc', 'cloud_web.service',  
'cloud_web.service.dc']  
INFO: Write file at '/root/.vault-token'  
INFO: Authenticated in https://dl.vault.ok.ru
```

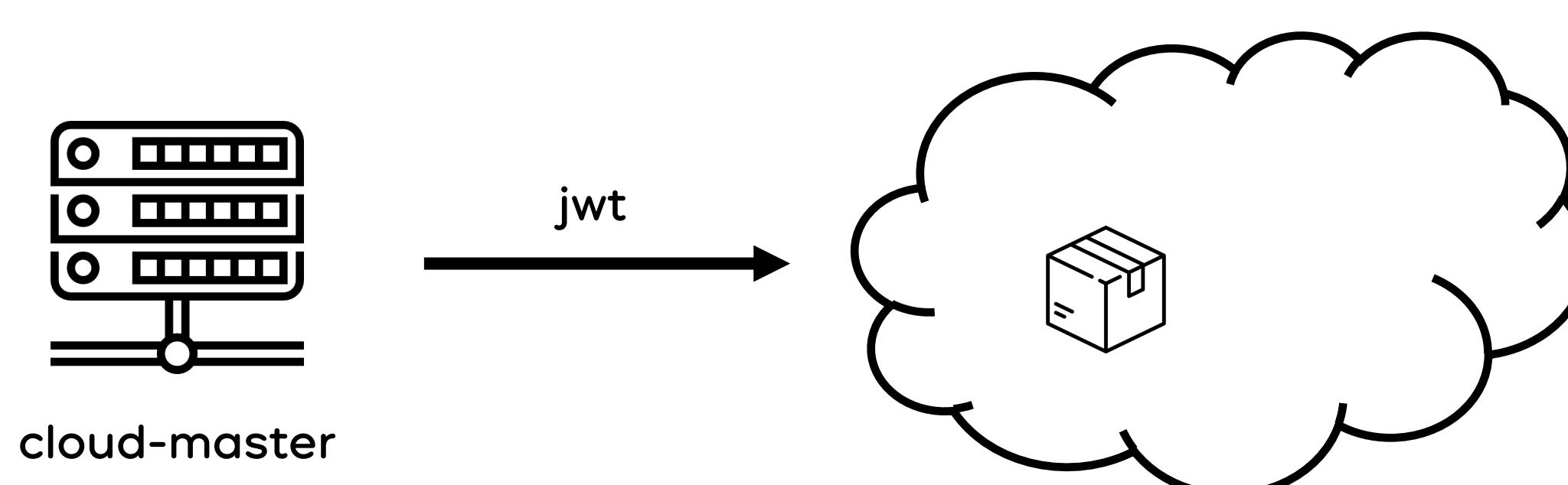
```
1.web.service.dc.ok.ru: /# cat ~/.vault-token  
s.oeX28WXHGoVwIh8FLE1iWgTz
```

Откуда берется JWT?



One-cloud

Мастер облака подписывает JWT своим
ключем и кладет в ENV контейнера

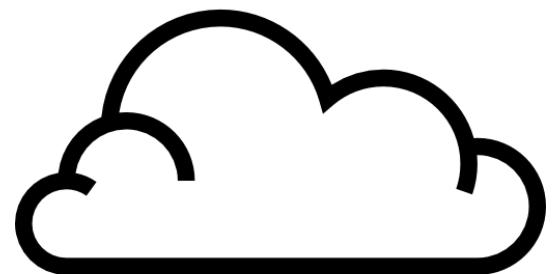


```
1.web.service.dc.ok.ru: /# echo $cloud_jwt_token  
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdWQiO...jbG91ZCI  
sIm5iZiI6MTYzMjYxOTM5NywiY2xvdWRfaG9zdG5hbWUiOiIxLm1vbmd  
vLmF1dG...90ZXN0LmRjIiwiN30.LeFxQr_3NpIuXY
```

```
1.web.service.dc.ok.ru: /# vault-login -v  
DEBUG: "POST /v1/auth/jwt/login HTTP/1.1" 200 923  
DEBUG: Logged to Vault via JWT. Policies:  
['cloud_instance', 'cloud_dc', 'cloud_web.service',  
'cloud_web.service.dc']  
INFO: Write file at '/root/.vault-token'  
INFO: Authenticated in https://dl.vault.ok.ru
```

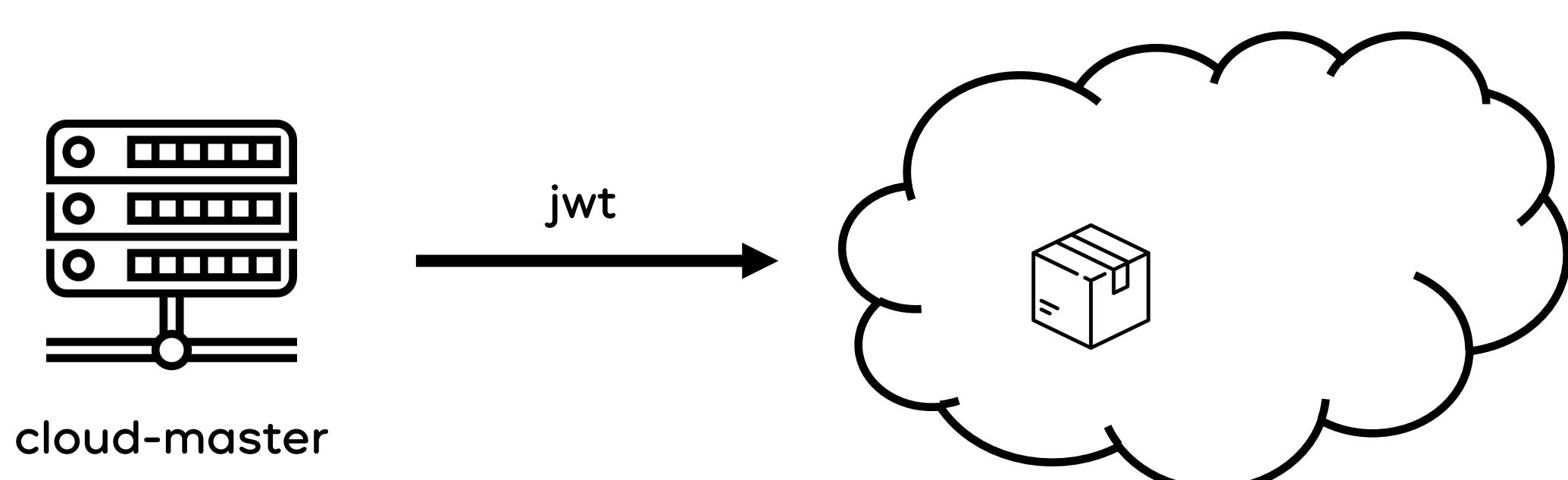
```
1.web.service.dc.ok.ru: /# cat ~/.vault-token  
s.oeX28WXHGoVwIh8FLE1iWgTz
```

Откуда берется JWT?



One-cloud

Мастер облака подписывает JWT своим
ключем и кладет в ENV контейнера

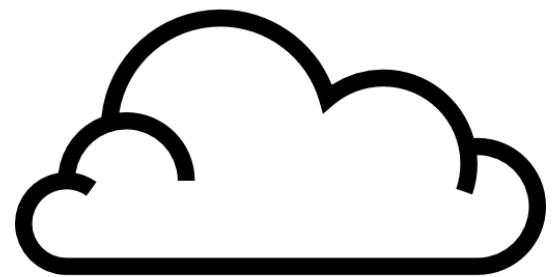


```
1.web.service.dc.ok.ru: /# echo $cloud_jwt_token  
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdWQiO...jbG91ZCI  
sIm5iZiI6MTYzMjYxOTM5NywiY2xvdWfaG9zdG5hbWUiOiIxLm1vbmd  
vLmF1dG...90ZXN0LmRjIiwiN30.LeFxQr_3NpIuXY
```

```
1.web.service.dc.ok.ru: /# vault-login -v  
DEBUG: "POST /v1/auth/jwt/login HTTP/1.1" 200 923  
DEBUG: Logged to Vault via JWT. Policies:  
['cloud_instance', 'cloud_dc', 'cloud_web.service',  
 'cloud_web.service.dc']  
INFO: Write file at '/root/.vault-token'  
INFO: Authenticated in https://dl.vault.ok.ru
```

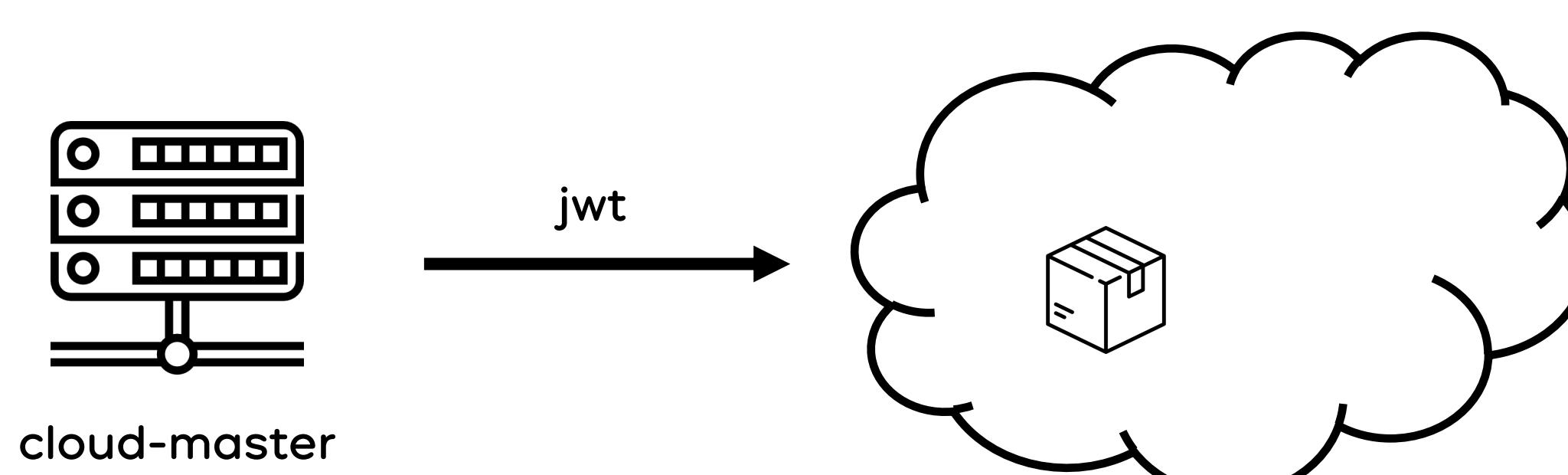
```
1.web.service.dc.ok.ru: /# cat ~/.vault-token  
s.oeX28WXHGoVwIh8FLE1iWgTz
```

Откуда берется JWT?



One-cloud

Мастер облака подписывает JWT своим
ключем и кладет в ENV контейнера

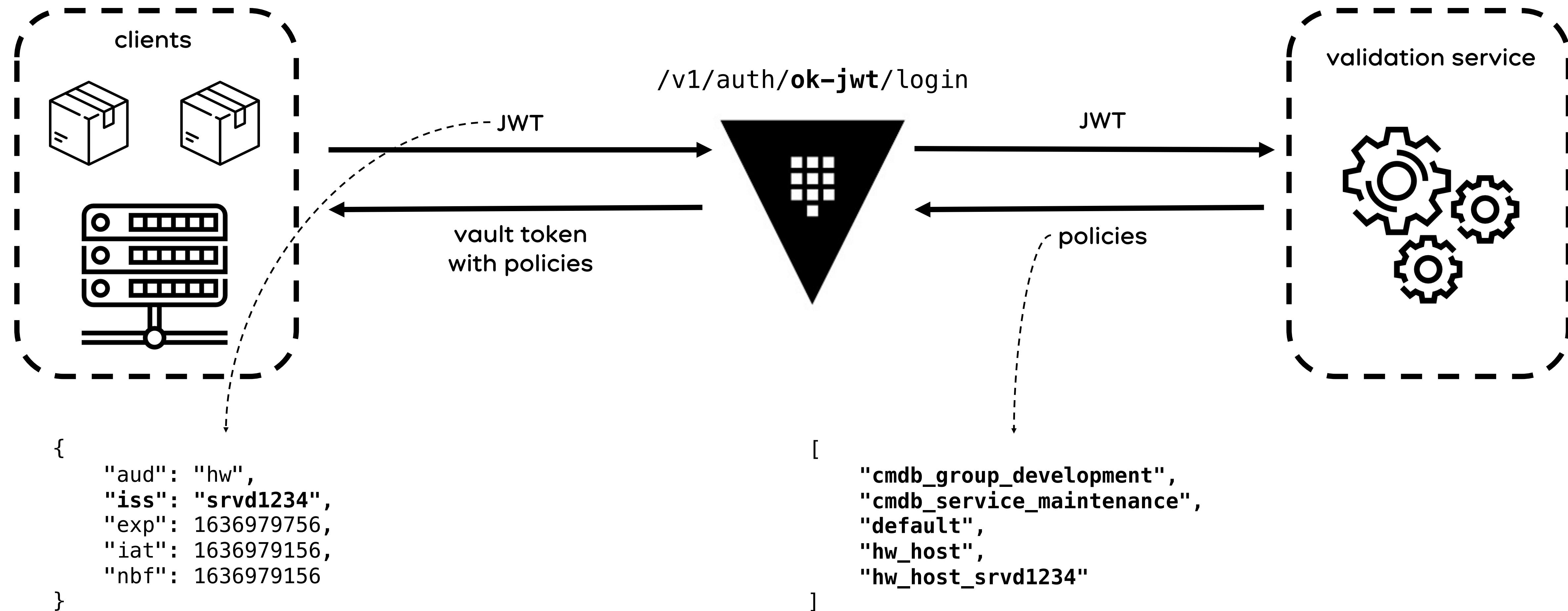


```
1.web.service.dc.ok.ru: /# echo $cloud_jwt_token  
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdWQiO...jbG91ZCI  
sIm5iZiI6MTYzMjYxOTM5NywiY2xvdWRfaG9zdG5hbWUiOiIxLm1vbmd  
vLmF1dG...90ZXN0LmRjIiwiN30.LeFxQr_3NpIuXY
```

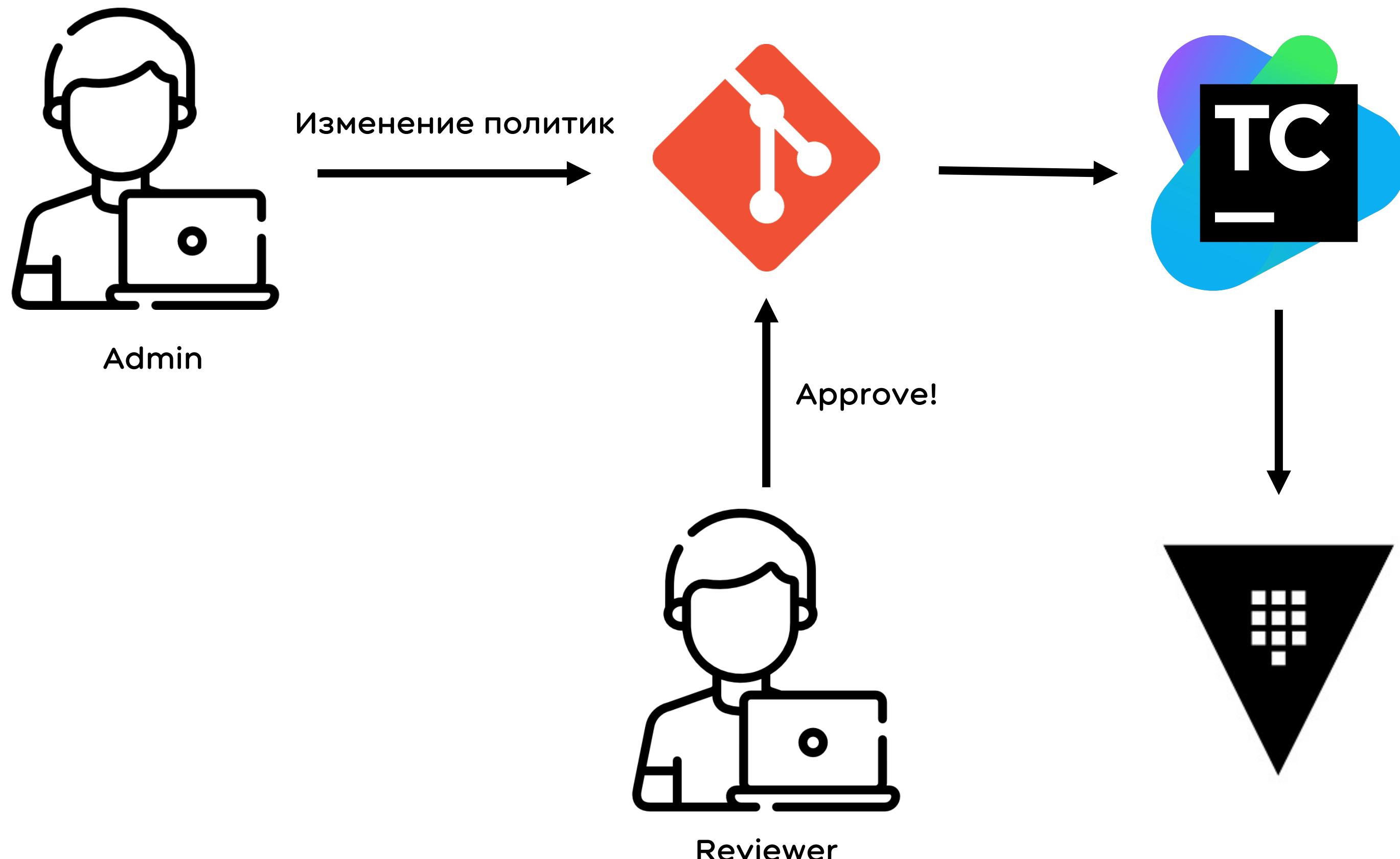
```
1.web.service.dc.ok.ru: /# vault-login -v  
DEBUG: "POST /v1/auth/jwt/login HTTP/1.1" 200 923  
DEBUG: Logged to Vault via JWT. Policies:  
['cloud_instance', 'cloud_dc', 'cloud_web.service',  
'cloud_web.service.dc']  
INFO: Write file at '/root/.vault-token'  
INFO: Authenticated in https://dl.vault.ok.ru
```

```
1.web.service.dc.ok.ru: /# cat ~/.vault-token  
s.oex28WXHGoVwIh8FLE1iWgTz
```

Наша аутентификация: OK JWT



Как применять политики



Наша авторизация: что получилось



- Нет ручных операций по созданию ролей
- Не нужно менять код приложений
- Динамические политики по нашим правилам
- Удобный аудит
- Интеграция со всеми нашими системами



- Интеграция с нашими системами
- Выписка сертификатов
- Отказ дата-центра
- Масштабирование



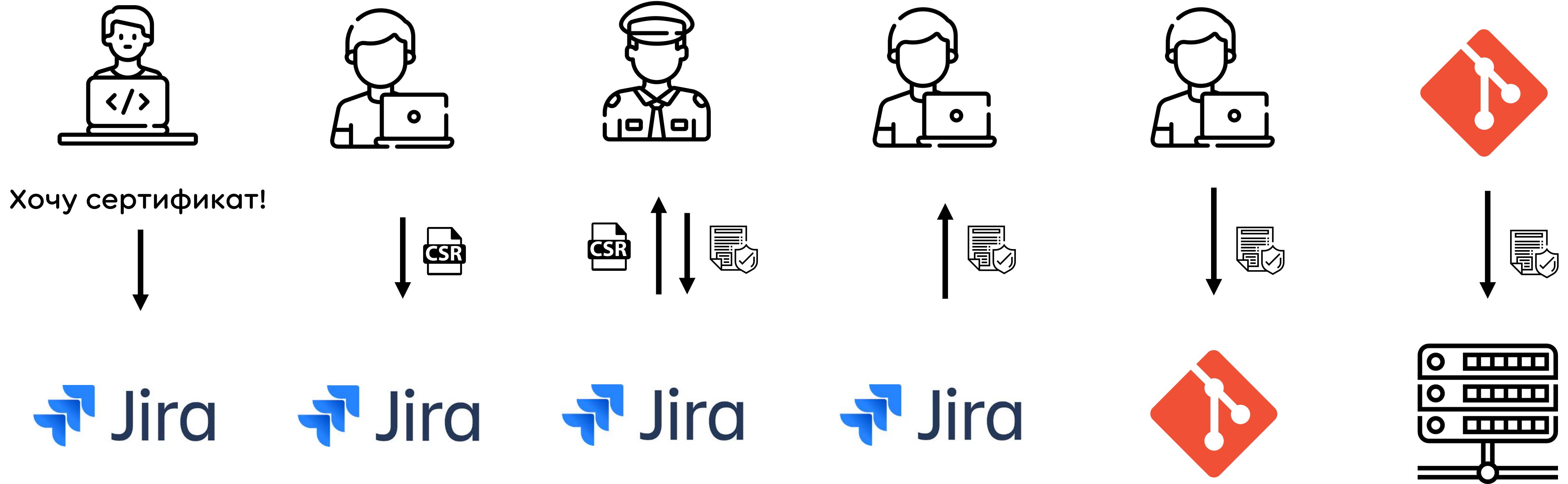
ХО



Выписка сертификатов: собственный «let's encrypt»

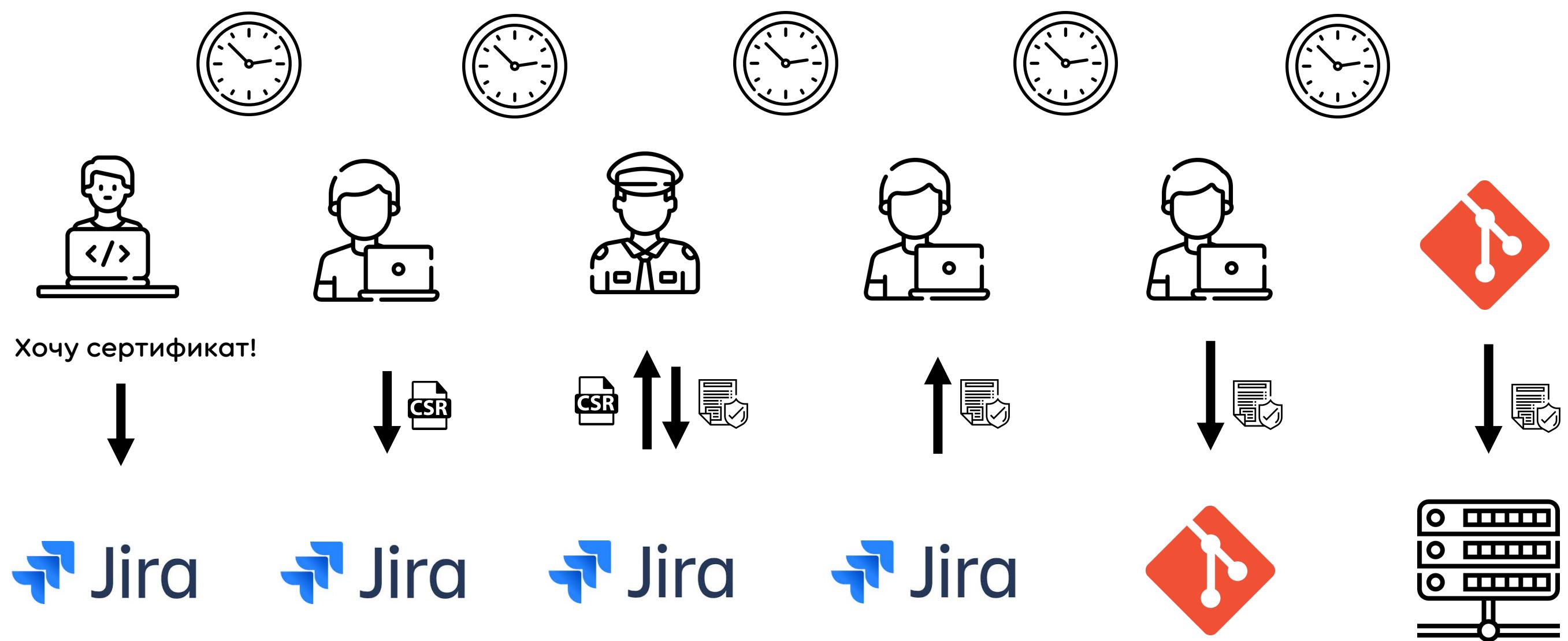
Жизнь до Vault

OK



Жизнь до Vault

Х



З человека

Много ручных операций

Долго

Сертификаты – это ОК



Vault умеет выписывать сертификаты



Шаблоны и переменные в путях к секретам

```
- path: /kv/data/+username  
- path: /kv/data/{{ dc }}/{{ service }}/password  
- path: /kv/data/{{ hostname }}/*
```



Шаблоны и переменные в именах доменов

```
allowed_domains:  
- *.ok.ru  
- web-*.ok.ru  
- "{{ service }}.ok.ru"  
- "{{ service }}.{{ dc }}.ok.ru"  
- "{{ hostname }}"
```

Нет поддержки шаблонов в именах доменов

`{{ <service> }}.{{ dc }}.ok.ru`

`web-1.dl.ok.ru`

`web-2.ec.ok.ru`

...

`web-N.xx.ok.ru`

Сертификаты – это ОК

- Хм, странно
- Поискали решение
- Нашли похожий PR
- Помогли его доработать

pki: Allow to use not only one variable during templating in allowed_domains
#9498

Merged calvn merged 2 commits into hashicorp:master from qk4l:pki_template_improve on 17 Aug 2020

Conversation 1 Commits 2 Checks 5 Files changed 2

qk4l commented on 16 Jul 2020

Contributor ...

Reviewers calvn

Assignees No one assigned

```
- matched, _ := regexp.MustCompile(`^{\.{.+?}}$`, currDomain)
- if matched && data.req.EntityID != "" {

+ isTemplate, _ := framework.ValidateIdentityTemplate(currDomain)
+ if isTemplate && data.req.EntityID != "" {
```

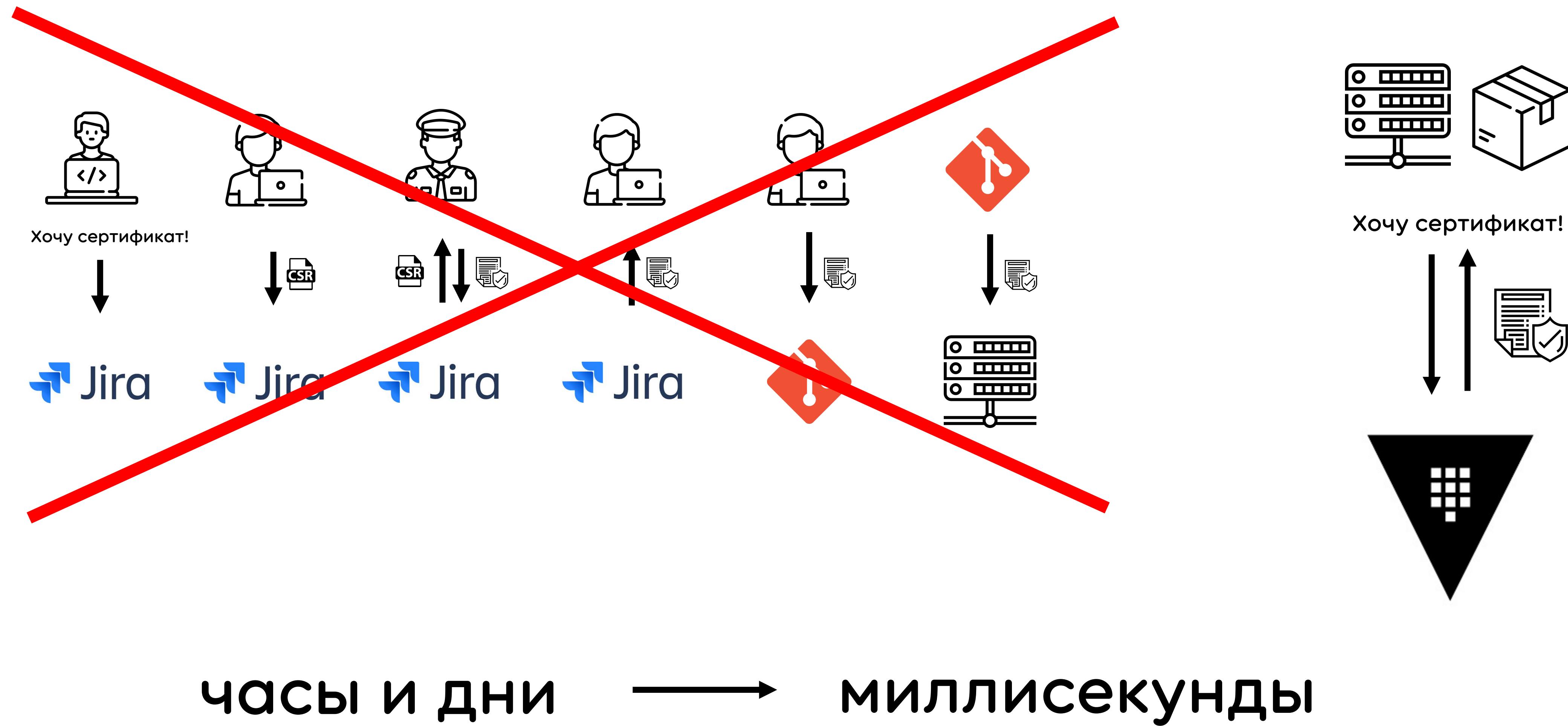
Сертификаты – это ОК

- Фича оказалась очень востребована
- HashiCorp сделали backport в прошлые версии (с 1.5.1)
- Сертификаты для всех, даром, и пусть никто не уйдет обиженный!

`allowed_domains:`

- "`{{ hostname }}.ok.ru`"
- "`{{ cloud_service }}.{{ dc }}.ok`"
- "`{{ cloud_service }}.clouds.ok.ru`"
- ...

Сертификаты – это ОК





- Интеграция с нашими системами
- Выписка сертификатов
- Отказ дата-центра
- Масштабирование

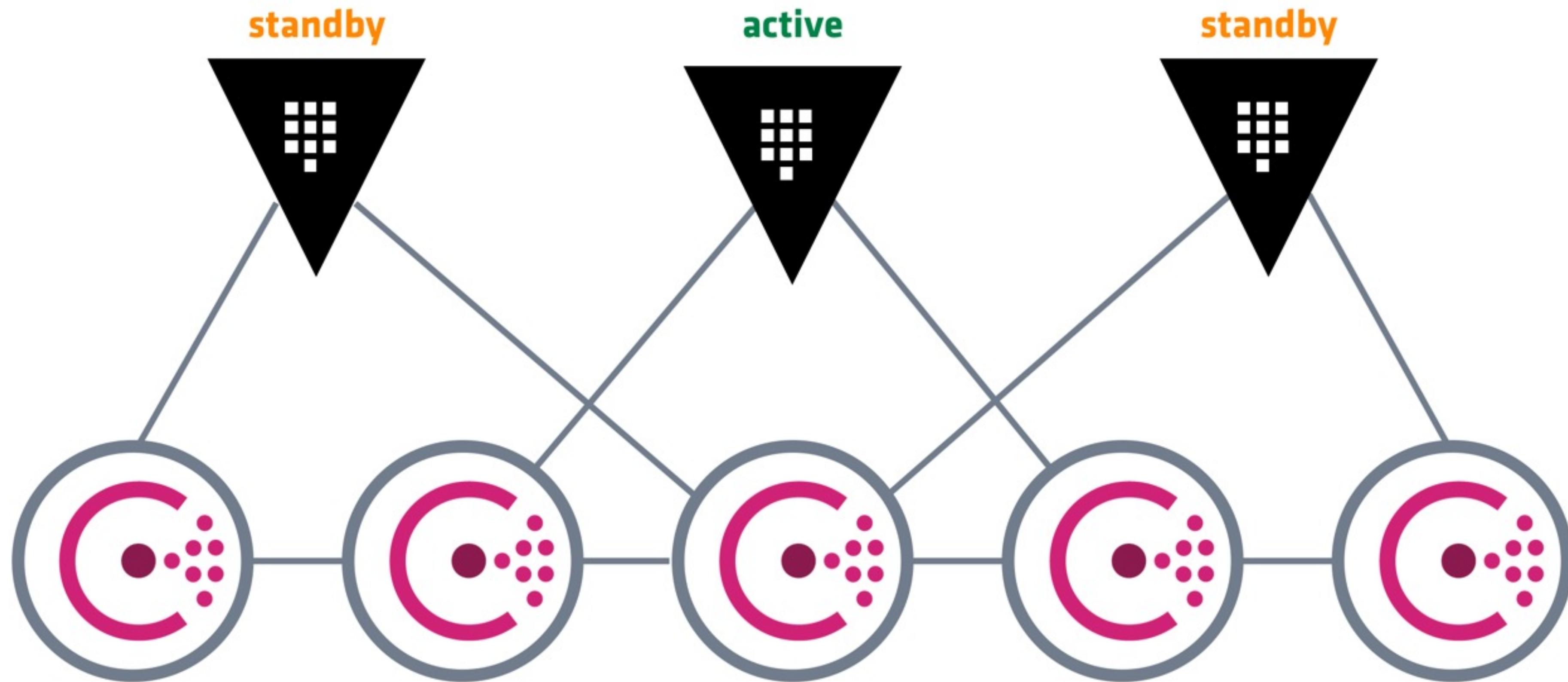


Х

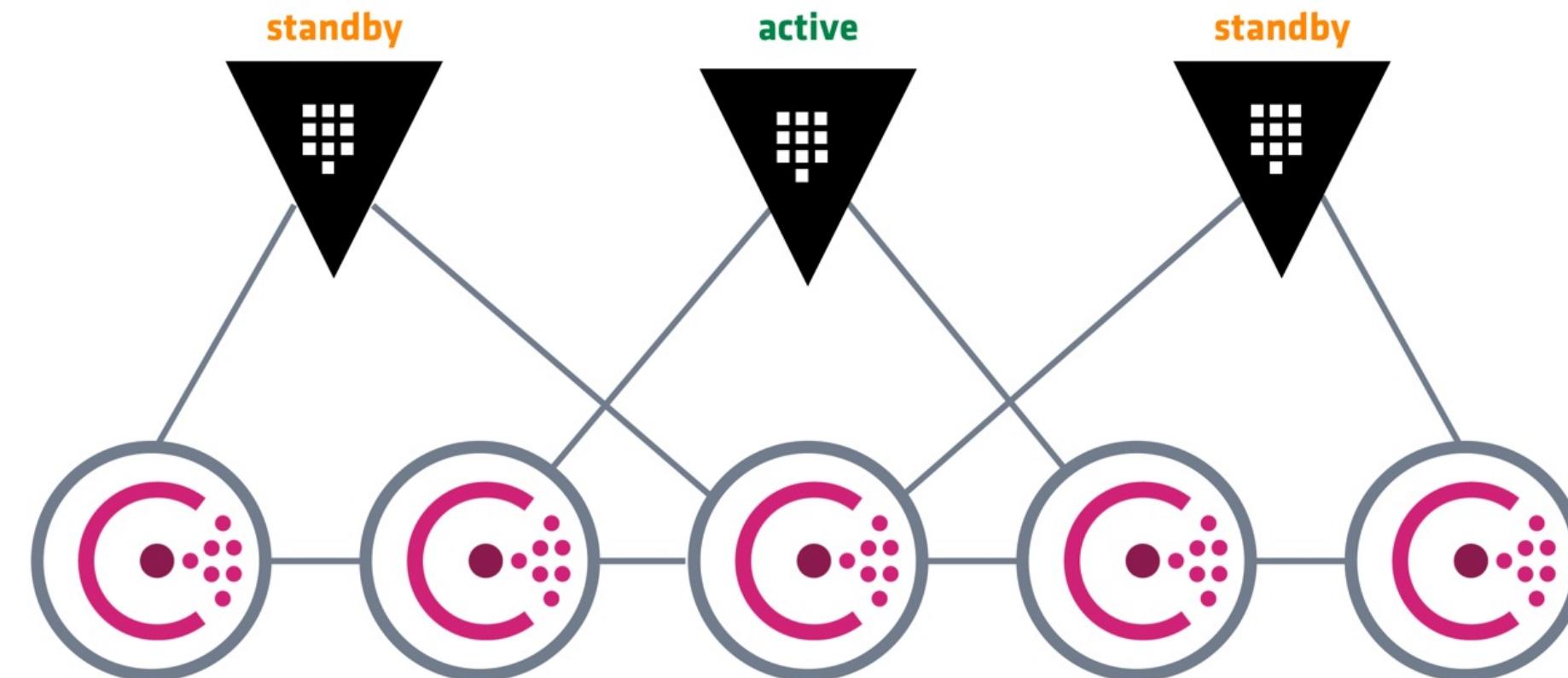


Отказ дата-центра и масштабирование

«Стандартный» кластер

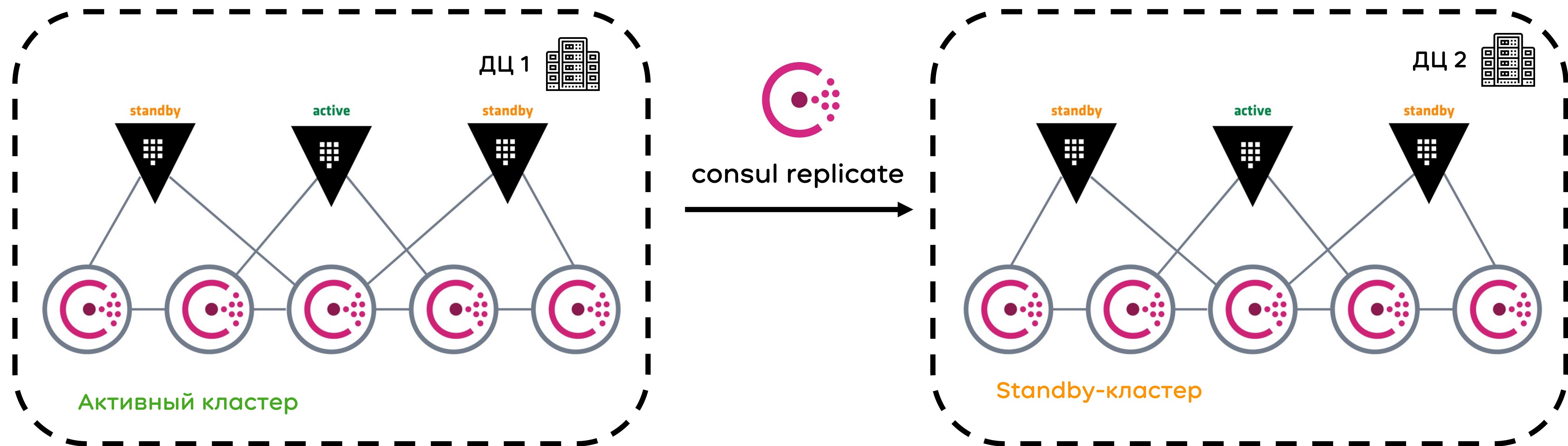


«Стандартный» кластер



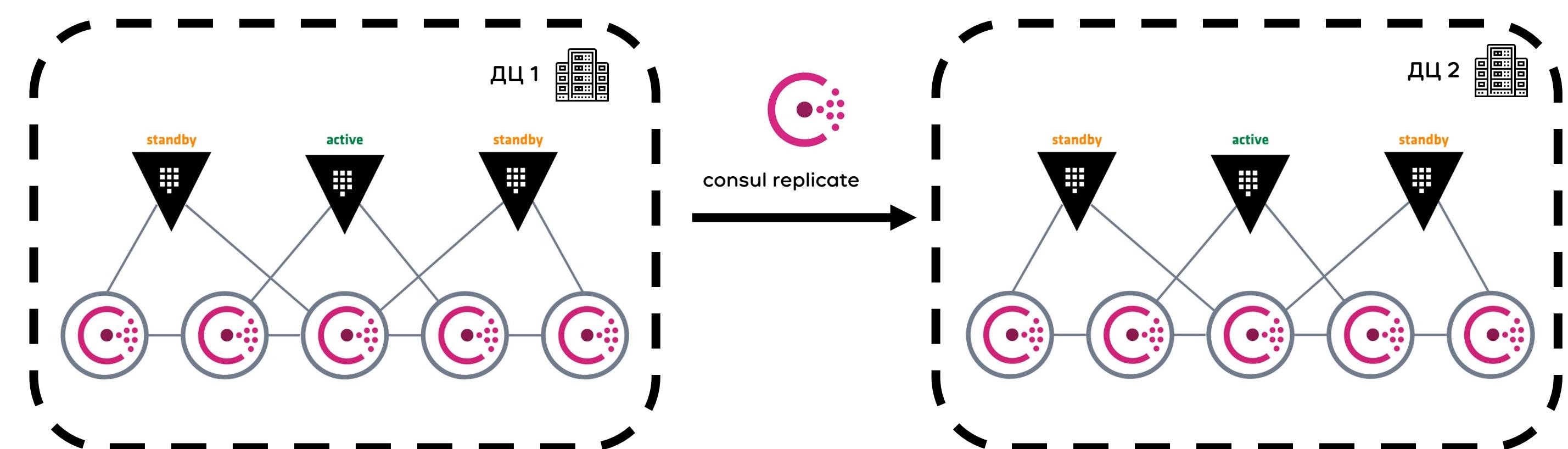
- Один активный сервер
- Standby-ноды не обрабатывают запросы
- Нет масштабирования
- Нет возможности обновить частично

Первые шаги



Уже лучше, но...

Х



- Нет «честного» отказа ДЦ: требуется участие нескольких человек для unseal, переключение репликации и т.д.
- Репликация – небесплатно
- Consul – мы не используем, и на внедрение нужно время
- Масштабирование, обновление

Есть ли решение от HashiCorp? Таки есть

Х



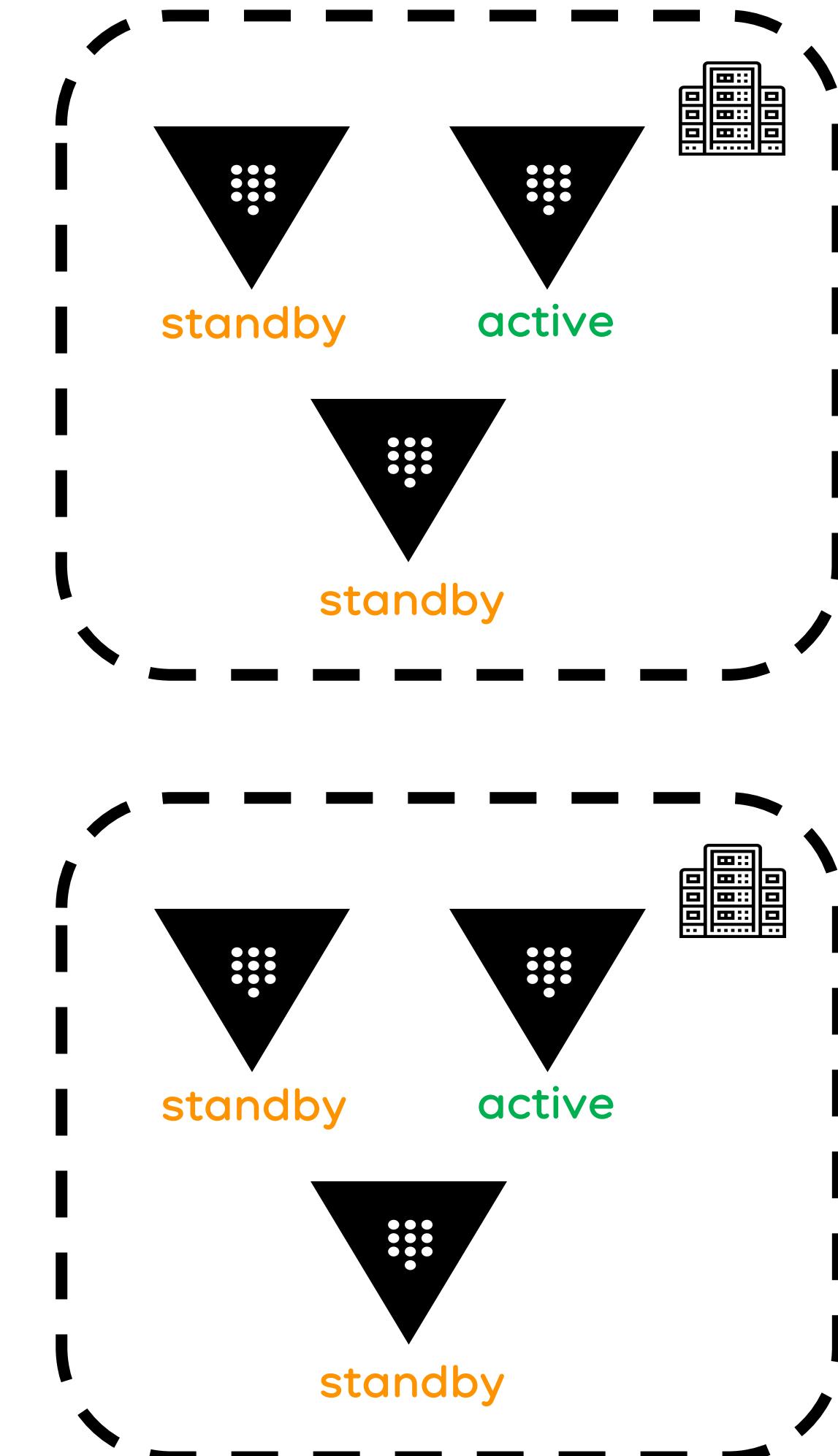
Vault	Open Source	Enterprise PLATFORM	MODULES
	Secrets management and data protection Download	Collaboration and operations features for teams Request a demo	Multi-datacenter, Scale, Governance and Policy features for organizations
ENTERPRISE PLATFORM			
Disaster Recovery	?	✓	✓
Namespaces	?	✓	✓
Monitoring	?	✓	✓
MULTI-DATACENTER & SCALE			
Replication	?		✓

imgflip.com

ЧТО МЫ ХОТИМ?



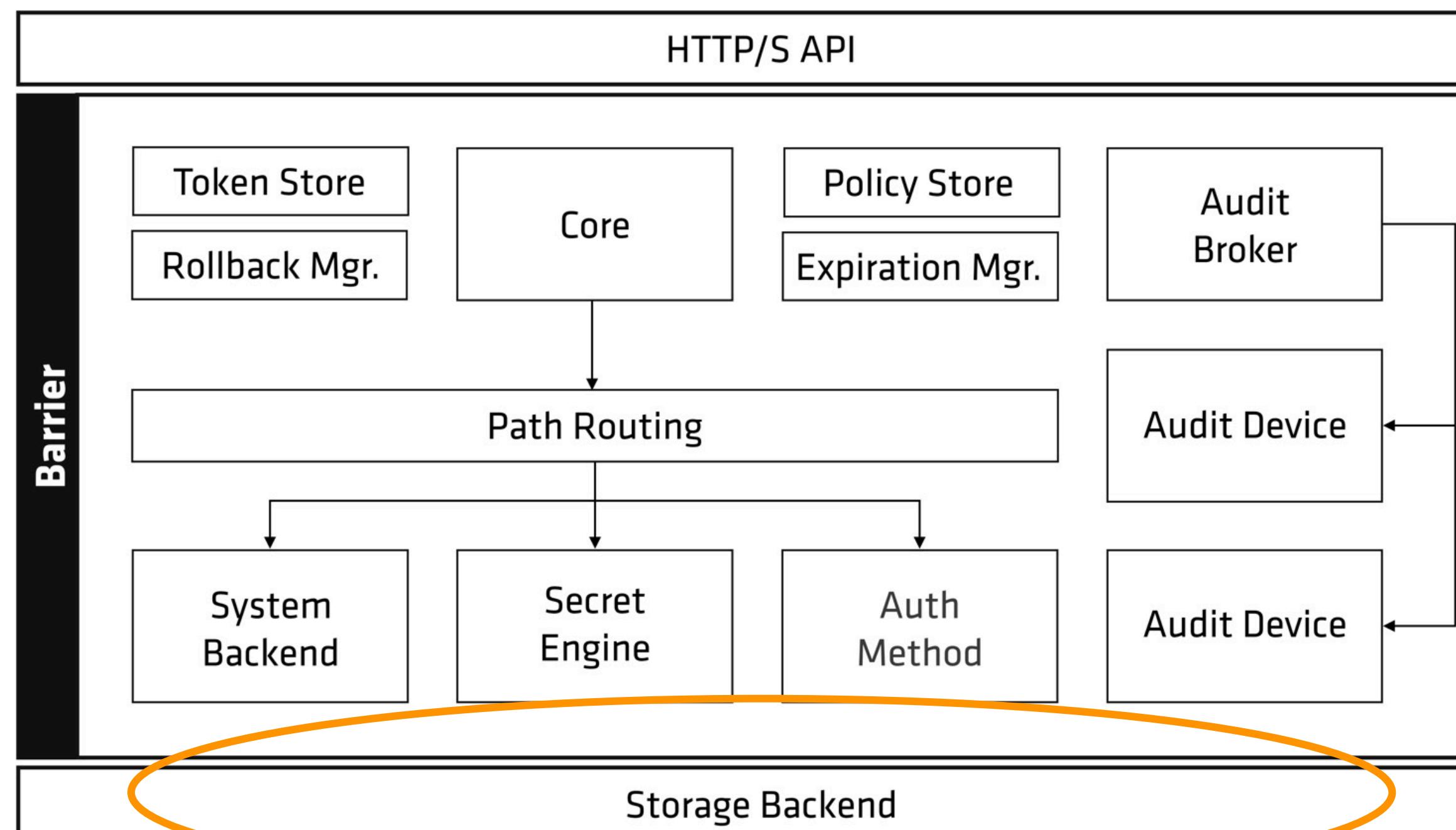
- Несколько кластеров
- Общие данные
- Прозрачно для клиентов



Как Vault хранит данные?

Формат зависит от реализации Storage Backend

Написать свой backend – несложно



```
type Backend interface {
    // Put is used to insert or update an entry
    Put(ctx context.Context, entry *Entry) error

    // Get is used to fetch an entry
    Get(ctx context.Context, key string) (*Entry, error)

    // Delete is used to permanently delete an entry
    Delete(ctx context.Context, key string) error

    // List is used to list all the keys under a given
    // prefix, up to the next prefix.
    List(ctx context.Context, prefix string) ([]string, error)
}
```

ЧТО ВНУТРИ?

Самый простой пример: **filesystem**

```
$ tree -L 2
.
├── auth
│   ├── 0cb9ba36-aa2e-5713
│   └── 7f7da874-13f4-fbfa
├── core
│   ├── _audit
│   └── wrapping
├── logical
│   ├── 23ab0741-b965-1668-fb05
│   └── da3951b5-c9eb-3f35-076f
└── sys
```

```
storage "file" {
    path = "/tmp/test-vault"
}
```

ЧТО ВНУТРИ?

Самый простой пример: **filesystem**

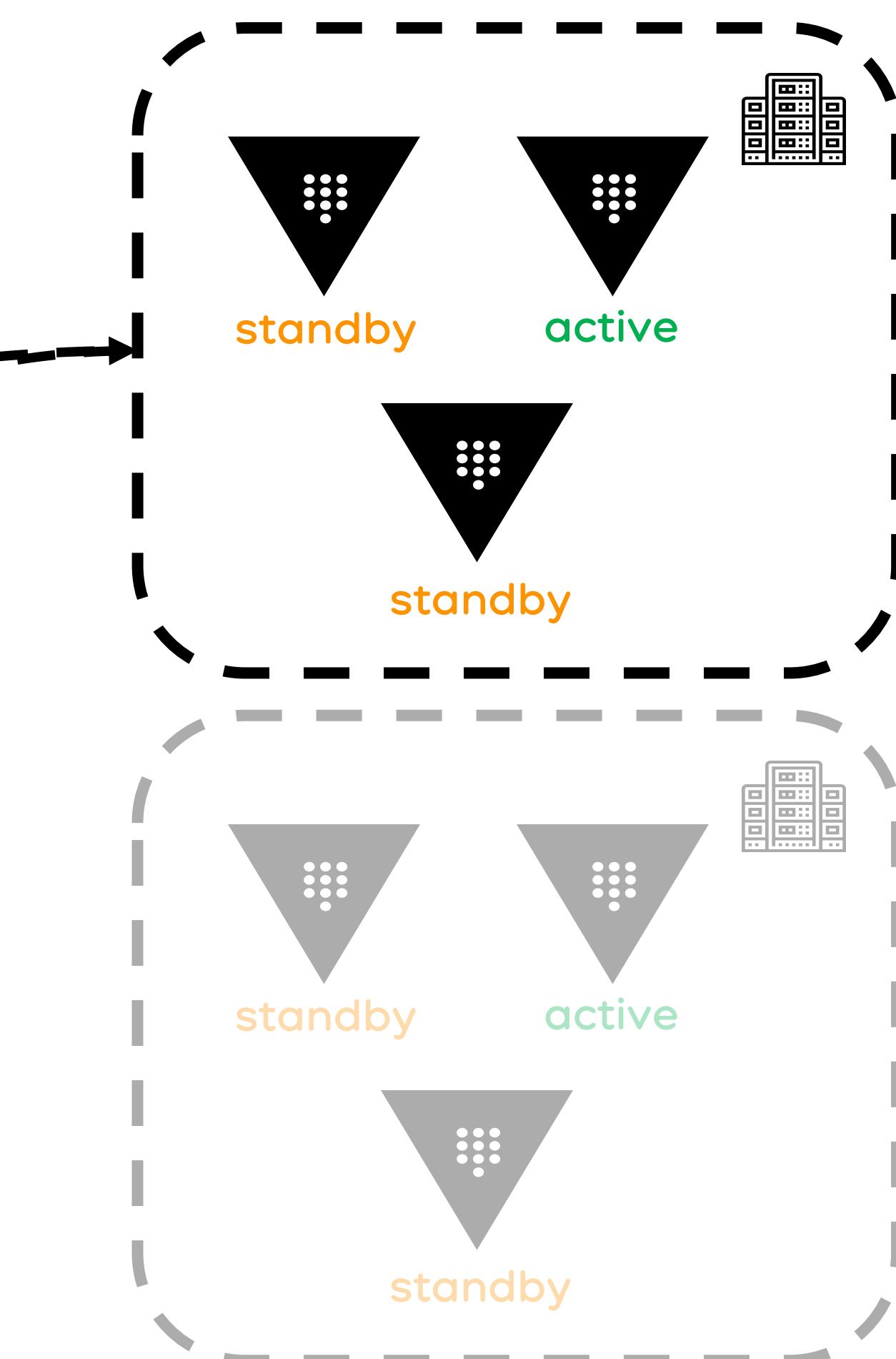
```
$ tree -L 2
```

«ПРИВАТНЫЕ» ДАННЫЕ

ЧТО ВНУТРИ?

Самый простой пример: filesystem

```
$ tree -L 2
.
├── auth
│   └── 0cb9ba36-aa2e-5713
│       └── 7f7da874-13f4-fbfa
├── core
│   ├── _audit
│   └── wrapping
├── logical
│   ├── 23ab0741-b965-1668-fb05
│   └── da3951b5-c9eb-3f35-076f
└── sys
    ├── counters
    ├── expire
    └── leader
```



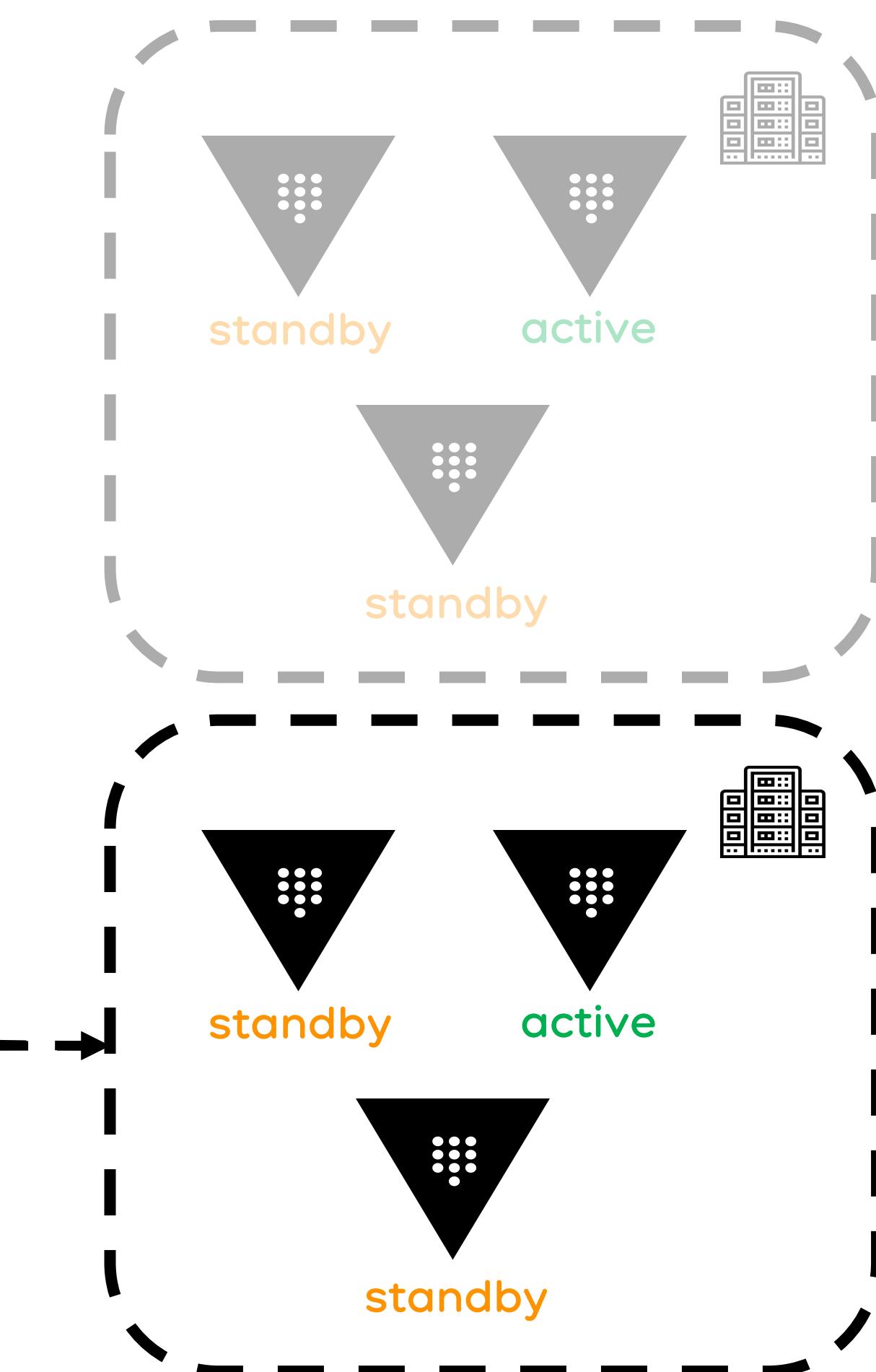
ЧТО ВНУТРИ?

Самый простой пример: filesystem

```
$ tree -L 2
```

```
.
```

- auth
 - 0cb9ba36-aa2e-5713
 - 7f7da874-13f4-fbfa
- core
 - _audit
 - wrapping
- logical
 - 23ab0741-b965-1668-fb05
 - da3951b5-c9eb-3f35-076f
- sys
 - counters
 - expire
 - leader
 - policy
 - token



ЧТО ВНУТРИ?

Самый простой пример: **filesystem**

```
$ tree -L 2
```

```
.
```

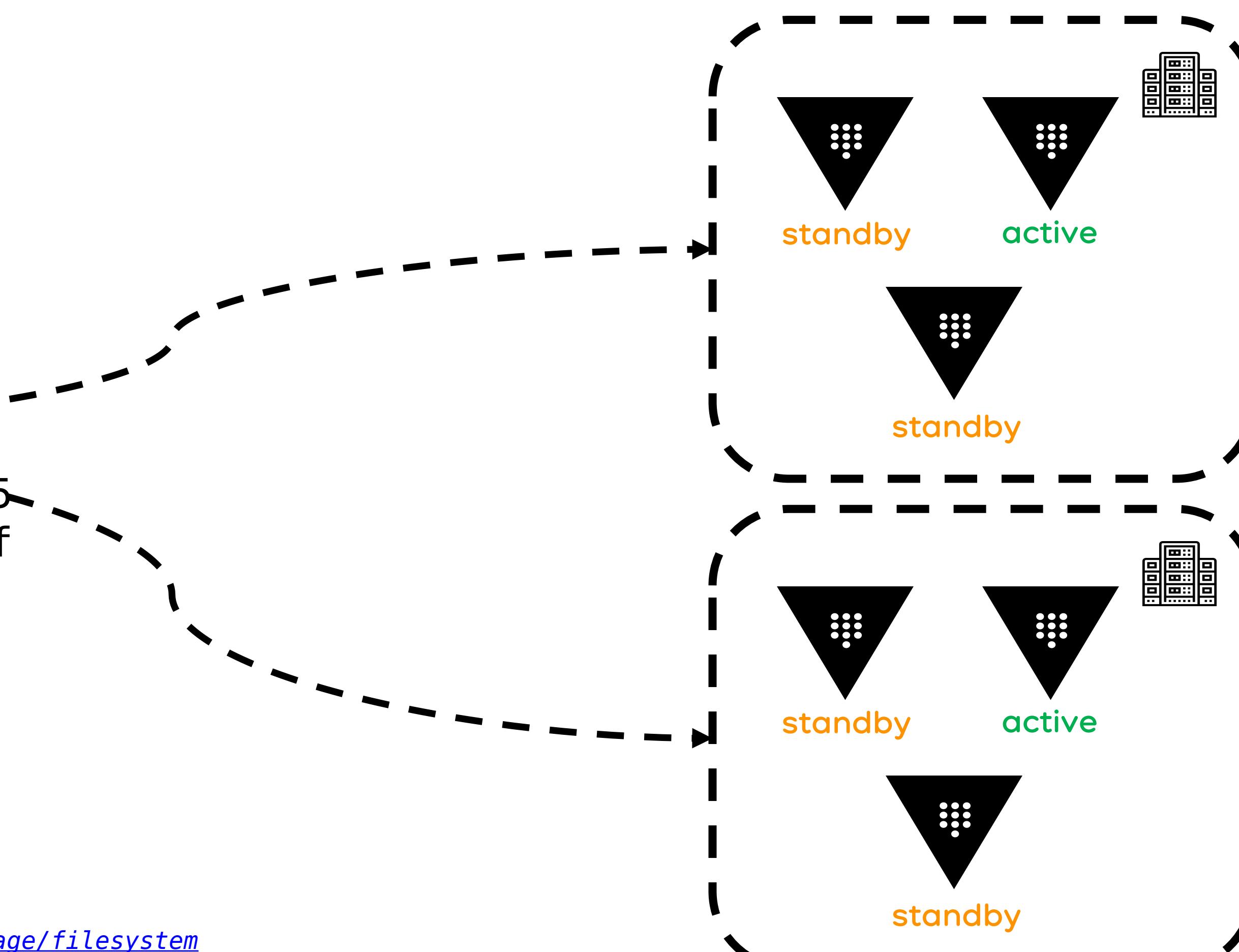
- auth
 - 0cb9ba36-aa2e-5713
 - 7f7da874-13f4-fbfa
- core
 - _audit
 - wrapping
- logical
 - 23ab0741-b965-1668-fb05
 - da3951b5-c9eb-3f35-076f
- sys
- counters
- expire
- leader
- policy
- token

«Общие» данные

ЧТО ВНУТРИ?

Самый простой пример: filesystem

```
$ tree -L 2
.
├── auth
│   └── 0cb9ba36-aa2e-5713
│       └── 7f7da874-13f4-fbfa
├── core
│   ├── _audit
│   └── wrapping
└── logical
    ├── 23ab0741-b965-1668-7b05
    └── da3951b5-c9eb-3f35-076f
├── sys
└── counters
```



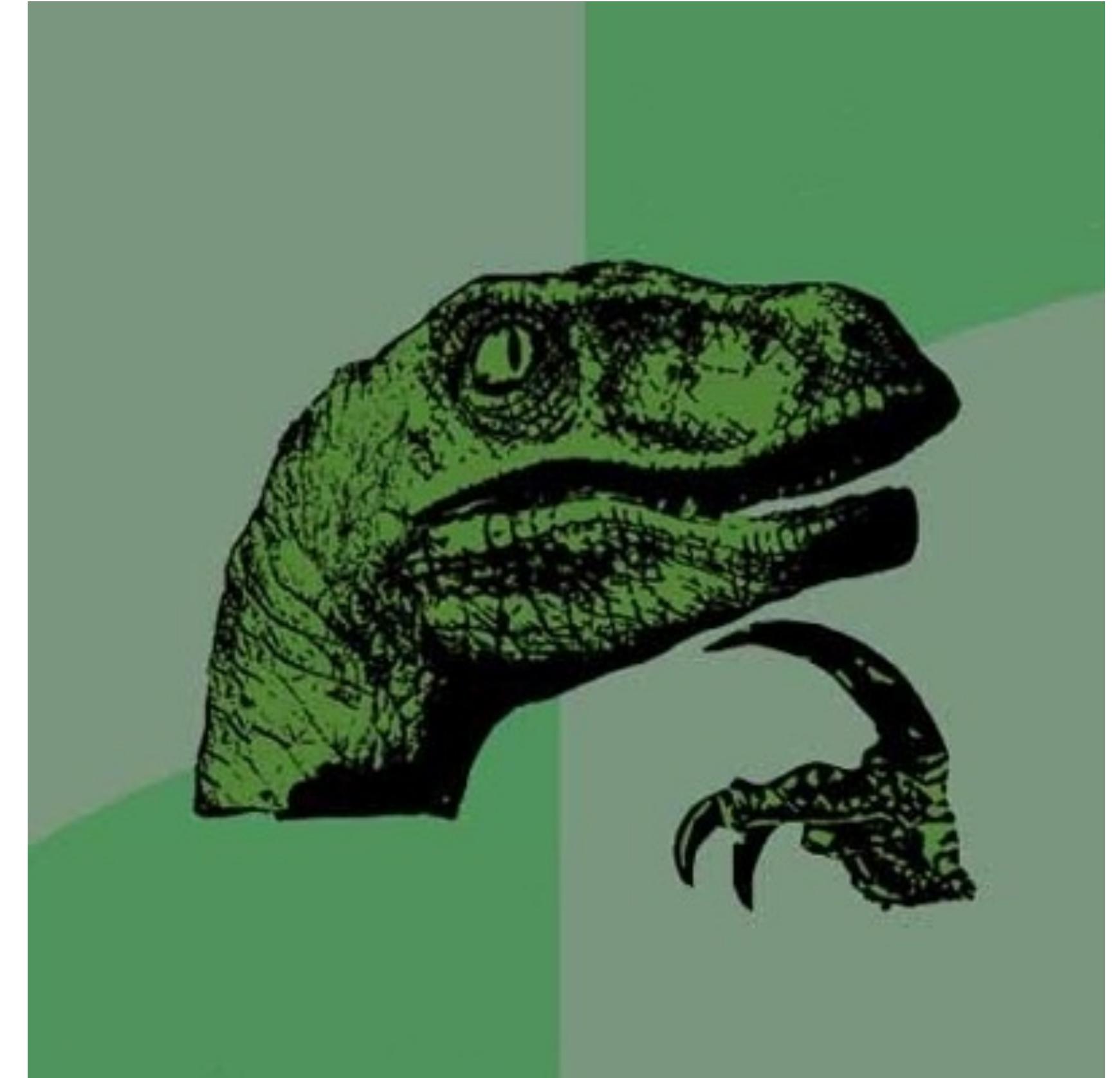
А что мешает сделать два активных
кластера?

х
о

А что мешает сделать два активных
кластера?

х
о

**Разным кластерам нужны
одинаковые данные**

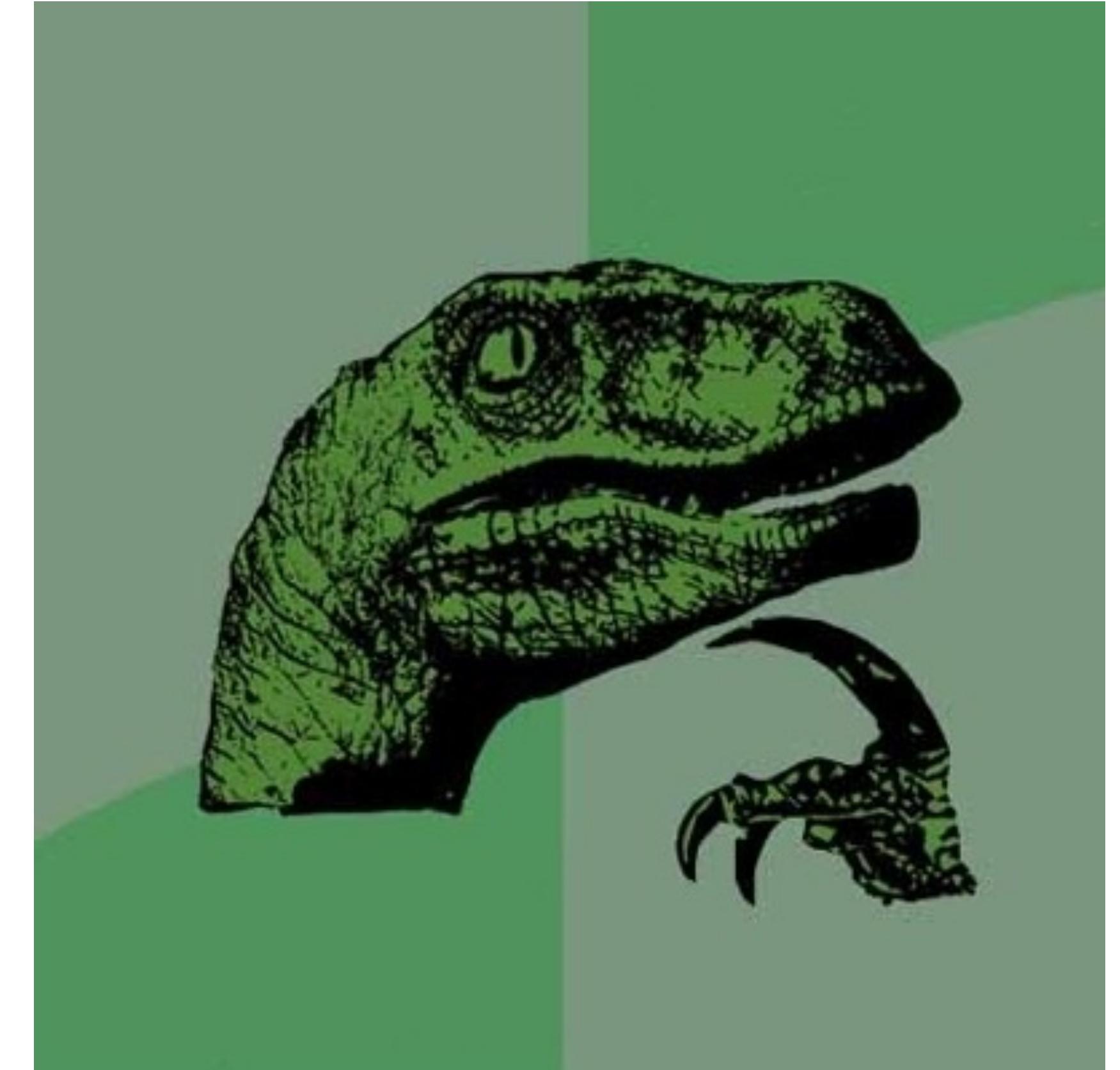


А что мешает сделать два активных кластера?

х

**Разным кластерам нужны
одинаковые данные**

**Разным кластерам нужны разные
данные**



Мы не первые задались вопросом



«Need better story for multiple datacenter scenarios #633»



jefferai commented on 22 Oct 2015

Member ...

It may still be worth some fiddling...there are two separate issues here:

One is that you're replicating *all* data. As I said before, you'd need to limit your replication to specific prefixes and keys. What's happening is that you have a local cluster in the target DC, the data is being sent over via consul-replicate, and along with that is coming the key with the information about which node is the leader. That's not the only key that will cause problems; you definitely do not want to replicate expiration information across. From some looking at a basic layout with the `file` physical backend, you'd want to copy `/sys/token` but not `/sys/expire`, yes to `/logical/`, and yes much of `/core` but not `/core/leader` and not `/core/lock`.

The other problem is the new value not showing up on the leaders in the other DCs. That *should* have been solved by disabling the cache, but are you sure that the right prefixes are being copied over with consul-replicate? Does it work after restarting Vault in those DCs?

Мы не первые задались вопросом

х



Мы не первые задались вопросом



«Need better story for multiple datacenter scenarios #633»

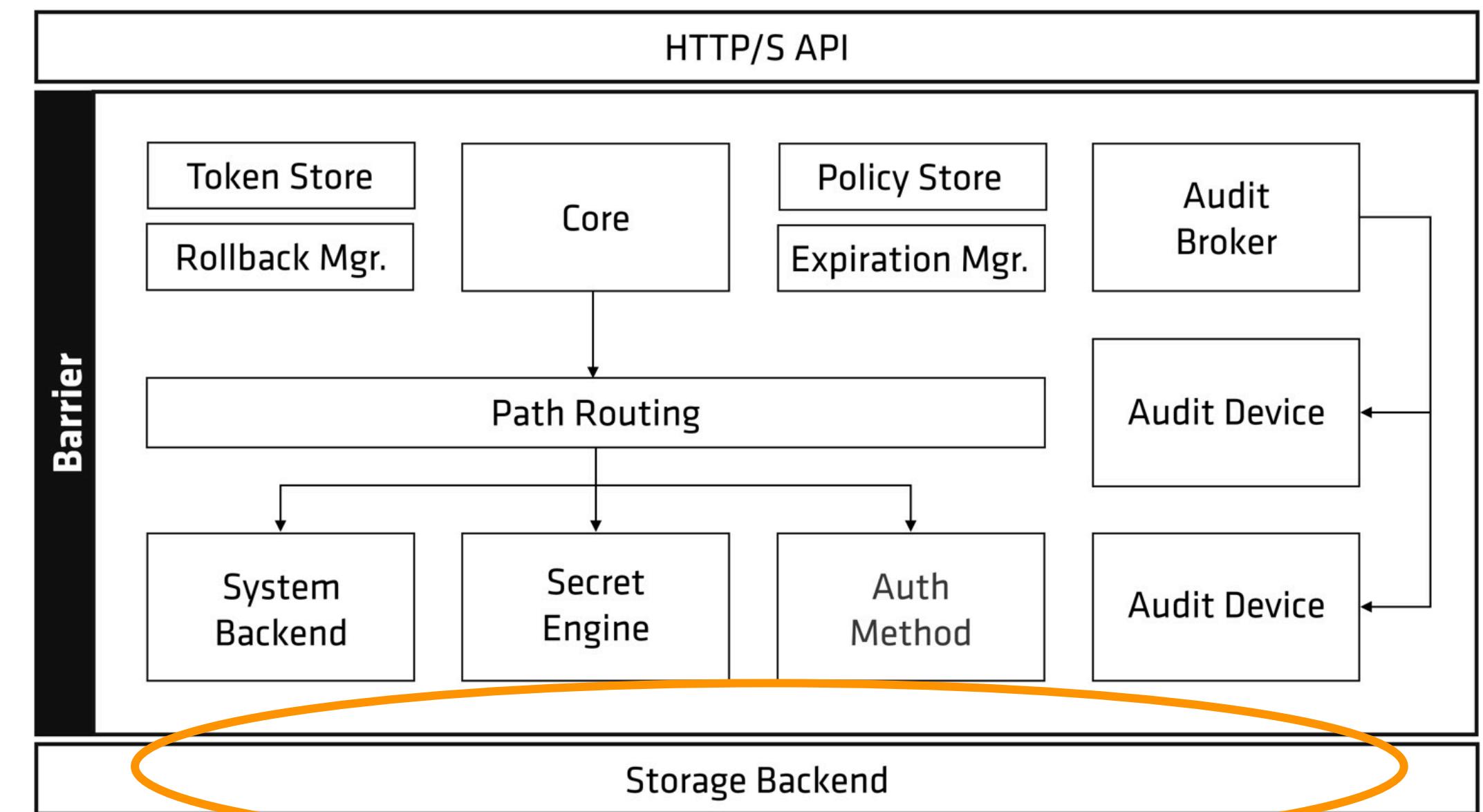


jefferai commented on 15 Mar 2018

Replication is a part of Vault Enterprise which solves this use-case, closing!

Интересно. Попробуем?

- **Идея: хранить «приватные» данные в отдельной таблице в базе данных**
- Взяли Cassandra
- Для Cassandra уже был готовый storage backend
- Дописали логику по выбору таблицы для чтения и записи в зависимости от пути запроса
- ...
- Это сработало!



Интересно. Попробуем?



```
CREATE KEYSPACE vault WITH replication = {  
    'class': 'NetworkTopologyStrategy',  
};
```



Интересно. Попробуем?



```
CREATE KEYSPACE vault WITH replication = {  
    'class': 'NetworkTopologyStrategy',  
};  
  
CREATE TABLE entries (  
    bucket text,  
    key text,  
    value blob,  
    PRIMARY KEY ((bucket), key), UPDATE TOKEN bucket  
)
```



Интересно. Попробуем?



```
CREATE KEYSPACE vault WITH replication = {  
    'class': 'NetworkTopologyStrategy',  
};  
  
CREATE TABLE entries (  
    bucket text,  
    key text,  
    value blob,  
    PRIMARY KEY ((bucket), key), UPDATE TOKEN bucket  
)  
  
CREATE TABLE entries_dc_1 (  
    bucket text,  
    key text,  
    value blob,  
    PRIMARY KEY ((bucket), key), UPDATE TOKEN bucket  
)
```



Интересно. Попробуем?



```
CREATE KEYSPACE vault WITH replication = {  
    'class': 'NetworkTopologyStrategy',  
};  
  
CREATE TABLE entries (  
    bucket text,  
    key text,  
    value blob,  
    PRIMARY KEY ((bucket), key), UPDATE TOKEN bucket  
)  
  
CREATE TABLE entries_dc_1 (  
    bucket text,  
    key text,  
    value blob,  
    PRIMARY KEY ((bucket), key), UPDATE TOKEN bucket  
)  
  
CREATE TABLE entries_dc_2  
  
CREATE TABLE entries_dc_N
```



Интересно. Попробуем?



```
func (c *CassandraBackend) Put(ctx context.Context, entry *physical.Entry) error {
-   stmt := fmt.Sprintf(`INSERT INTO "%s" (bucket, key, value) VALUES (?, ?, ?)` , c.table)
+
+   table := c.getTableByDC(c.cluster_name, entry.Key)
+   stmt := fmt.Sprintf(`INSERT INTO "%s" (bucket, key, value) VALUES (?, ?, ?)` , table)

+ func (c *CassandraBackend) getTableByDC(hostname, entryKey string) string {
+     buckets := c.buckets(entryKey)
+     for _, path := range privatePaths {
+         for _, bucket := range buckets {
+             if strings.Contains(bucket, path) {
+                 return c.table + "_" + c.cluster_name
+             }
+         }
+     }
+ }
+ return c.table
```

Успех!



- Можем использовать несколько активных кластеров
- Cassandra под капотом
- Легко переживаем отказы ДЦ
- Можем работать с кластерами независимо (*обновлять, проверять новые функции и т.д.*)

ХО



Конечно, не все
было так гладко...

«Сп» vs «Cn» vs «CN»

Vault некорректно работал с LDAP, если атрибут CN написан в другом регистре

Fix getCN during group fetching in auth/ldap #6518

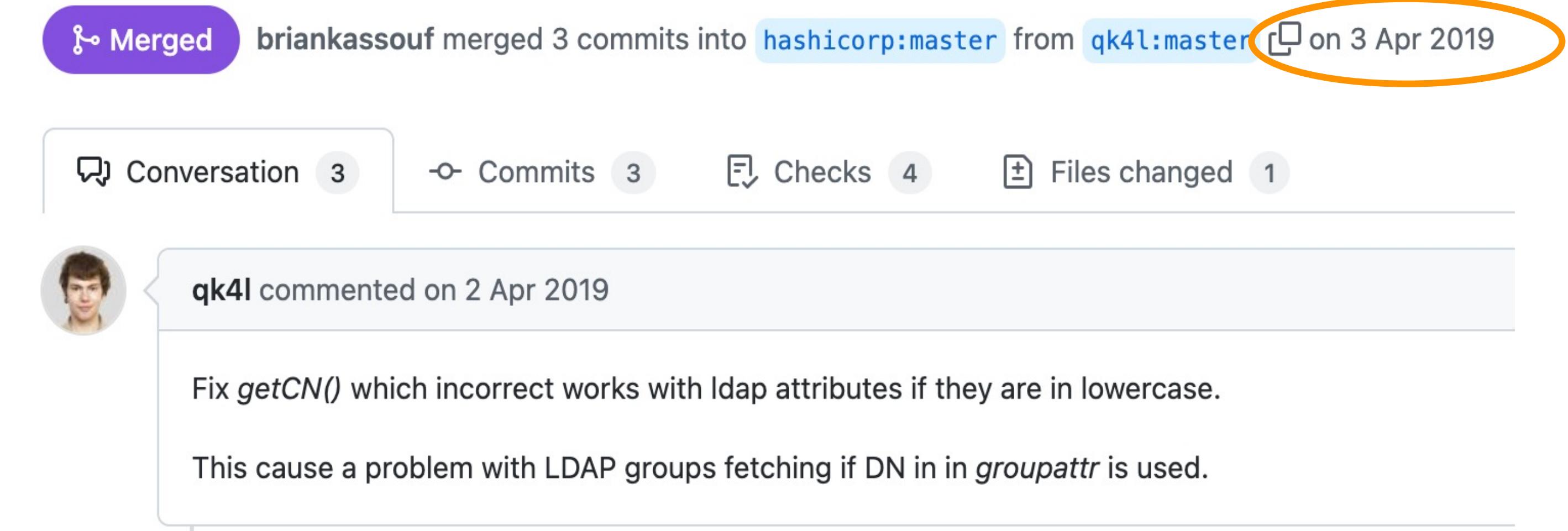
Merged briankassouf merged 3 commits into hashicorp:master from qk4l:master on 3 Apr 2019

Conversation 3 Commits 3 Checks 4 Files changed 1

qk4l commented on 2 Apr 2019

Fix `getCN()` which incorrect works with ldap attributes if they are in lowercase.

This cause a problem with LDAP groups fetching if DN in `groupattr` is used.



```
- if rdnAttr.Type == "CN" {  
+ if strings.EqualFold(rdnAttr.Type, "CN") {
```



🔒 gocql.github.io



GoCQL github.com/gocql

Under Development: The GoCQL package is currently actively developed and the API may change in the future.

Cassandra backend не работал с нашей версией

х

Починили обновлением зависимости gocql



Update gocql version for cassandra physical backend #9602

Merged ncabatoff merged 1 commit into hashicorp:master from byumov:update-gocql-version on 27 Jul 2020

Conversation 3 Commits 1 Checks 5 Files changed 27

byumov commented on 27 Jul 2020

Version of [gocql](#) library(using for Cassandra physical backend) more, than one year old now.
gocql [is in active developing](#) now and a lot of bugs were fixed.

For example, now we can't use Cassandra backend for Vault with old Cassandra version(2.0). It was fixed here:
[gocql/gocql#1292](#)

This PR just bump gocql version.

6

Нескучная документация

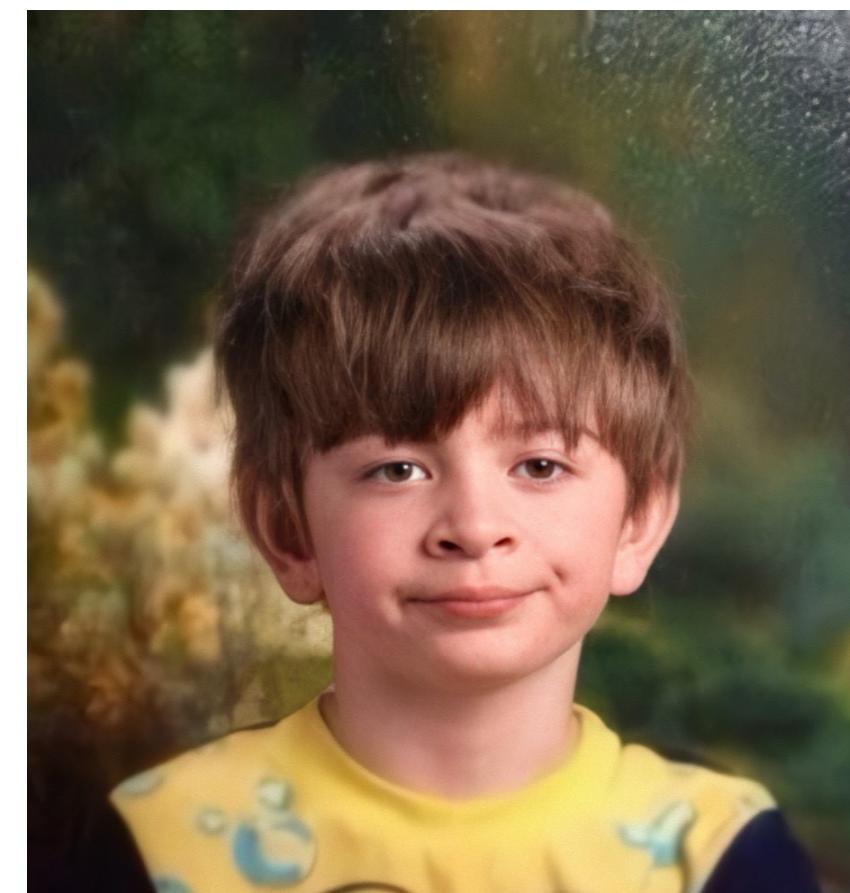
- `connection_timeout` (int: 0) - A timeout in seconds to wait until a connection is established with the Cassandra hosts.

Какой вы ожидаете timeout по умолчанию?

Может быть бесконечный?

Нет. 600 ms

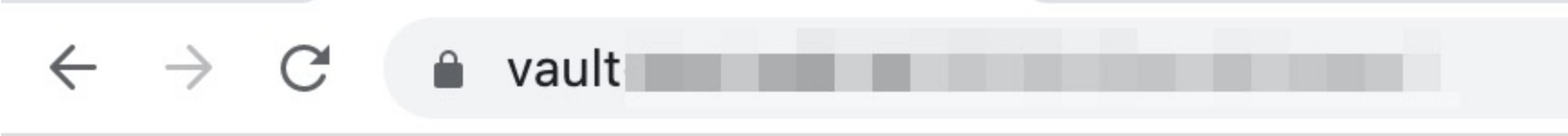
```
func NewCluster(hosts ...string) *ClusterConfig {  
    cfg := &ClusterConfig{  
        Hosts:           hosts,  
        CQLVersion:     "3.0.0",  
        Timeout:         600 * time.Millisecond,  
        ConnectTimeout: 600 * time.Millisecond,  
    }  
    return cfg  
}
```



Отложенные сюрпризы

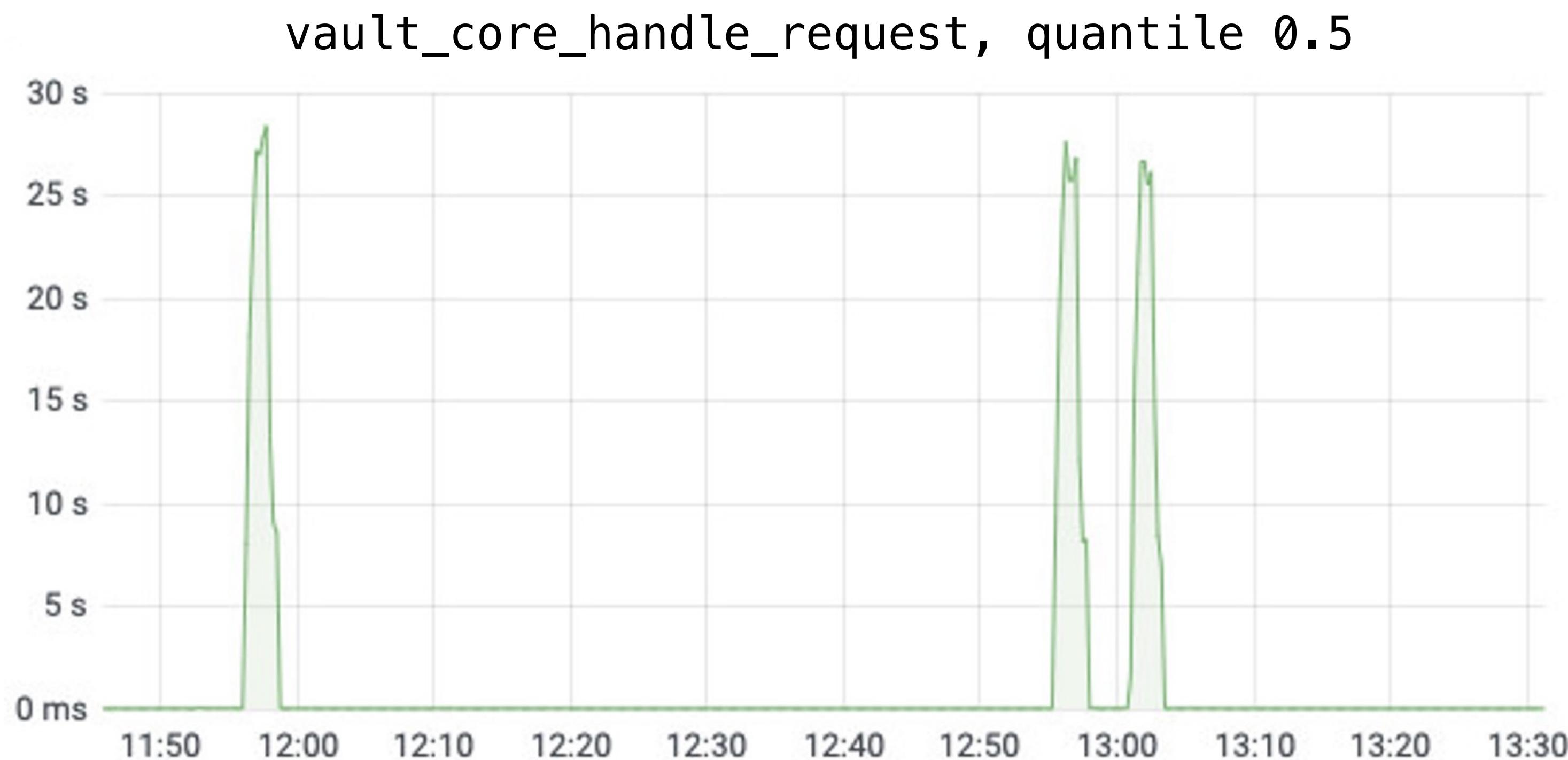
ХО

- Cassandra легко переживает отказ нод
- Однажды мы обновили кластер cassandra...



Живые ноды не возвращались, Vault работал до рестарта последней ноды Cassandra

Проблемы с нагрузкой



- Одновременно пришло много клиентов
- CPU на нодах Vault не вырос
- Время запросов в базу не увеличилось

Проблемы с нагрузкой

```
11:56:53 [DEBUG] policy: [WeKTEUP] PolicyStore.switchedGetPolicy: grab lock  
11:56:59 [DEBUG] policy: [WeKTEUP] PolicyStore.view.Get (db call): start
```

При отключенном кеше **все** запросы (*да, чтение тоже*) проваливаются в базу под **глобальным локом**

```
if grabLock {  
    ps.modifyLock.Lock()  
    defer ps.modifyLock.Unlock()  
}
```

Проблемы с нагрузкой

ХО

Выводы:

- Жить без кеша не получится
- Включили кеш
- Проблема решена!

Проблемы после включения кеша

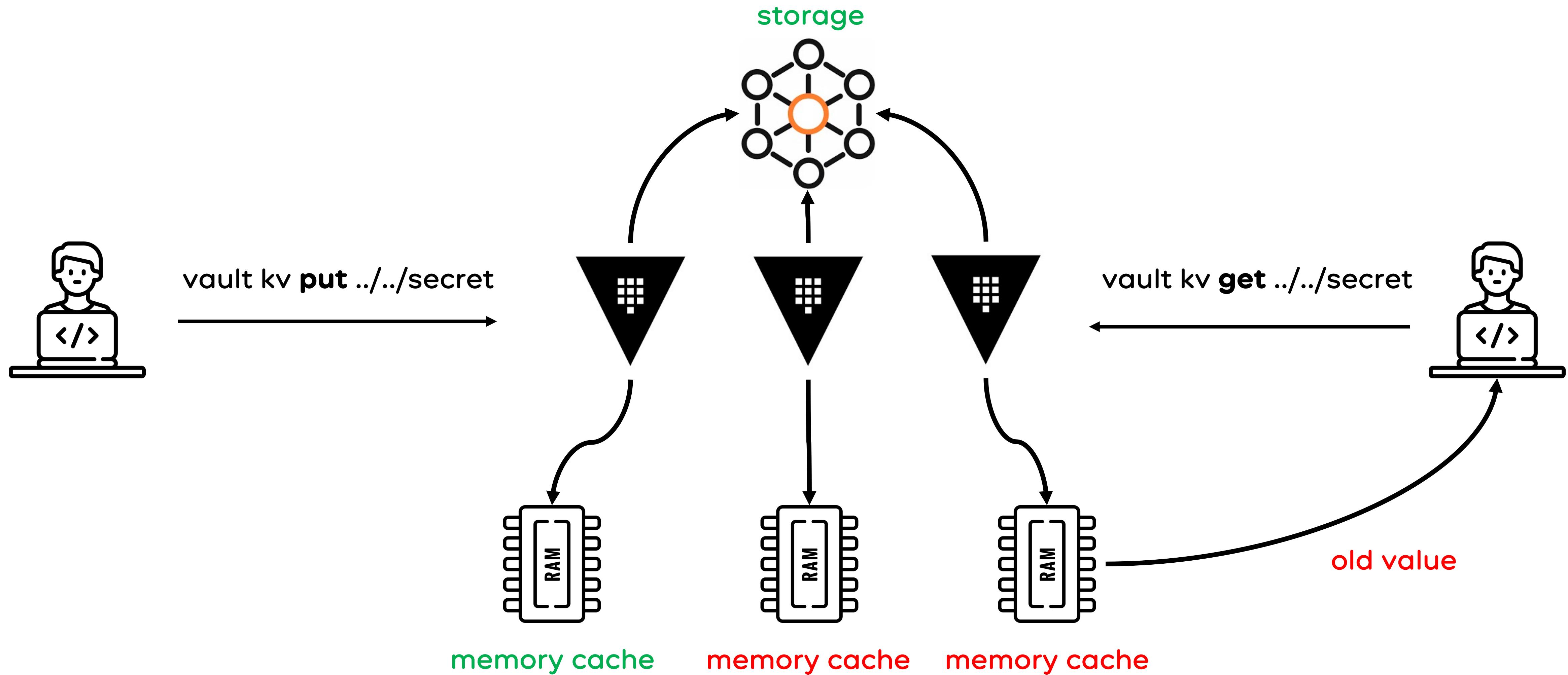
ХО

Мы осознанно отключали кеш, опасаясь проблем

Начали поступать жалобы:

- От админов: «*добавил новую политику, а доступ не появился*»
- От разработчиков: «*мы поменяли пароль, а Vault отдает старое значение*»

Посмотрим, как работает кеш



Как можно обойти?

Писать новые значения в каждый кластер?

```
› VAULT_ADDR=https://vault-dc1.ok.ru vault kv put ../../secret  
› VAULT_ADDR=https://vault-dc2.ok.ru vault kv put ../../secret  
› VAULT_ADDR=https://vault-dc3.ok.ru vault kv put ../../secret
```

- Во всех кластерах будет актуальный кеш 
- Заставлять клиентов писать в разные кластера не очень хорошо (*клиенты могут не знать про такую «особенность»*) 
- Непонятно, как решать проблемы с консистентностью (*какой-то кластер может быть недоступен при записи*) 

Как можно обойти?

ХО

Можно периодически менять мастера в кластере

› vault operator step-down

- У нового мастера будет чистый кеш 
- Смена мастера довольно тяжёлая операция (*до нескольких секунд, нужно получить с базы все leases и проверить их*) 
- Клиенты будут получать ошибки в момент смены мастера 

Элегантное решение 😊



Давайте сбрасывать кеш!

у Vault такого функционала не предусмотрено

```
+ go func() {
+     for _ = range time.Tick(time.Minute * 5) {
+         ps.tokenPoliciesLRU.Purge()
+         ps.egpLRU.Purge()
+         ps.logger.Warn("cache was purged")
+     }
+ }()
```

Побочный эффект: клиенты в течение 5-ти минут могут получать старые значения

С чем живем сейчас

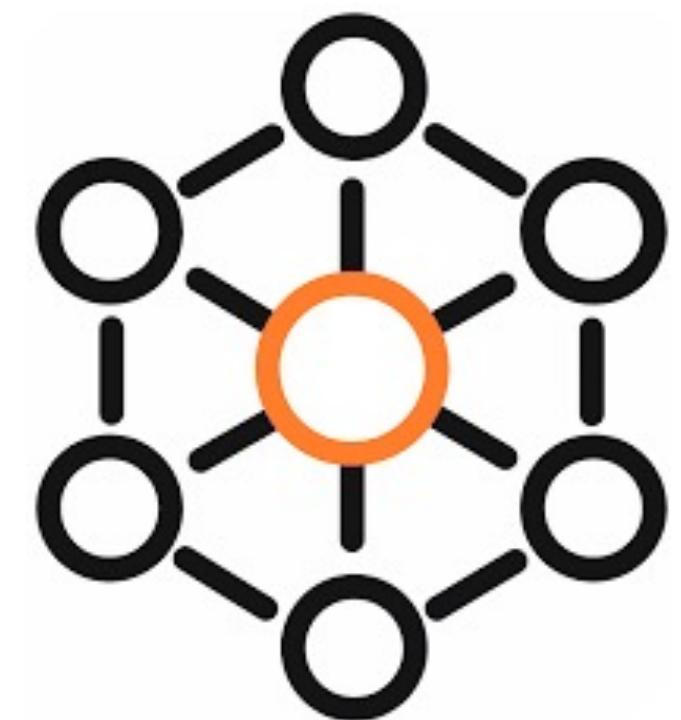


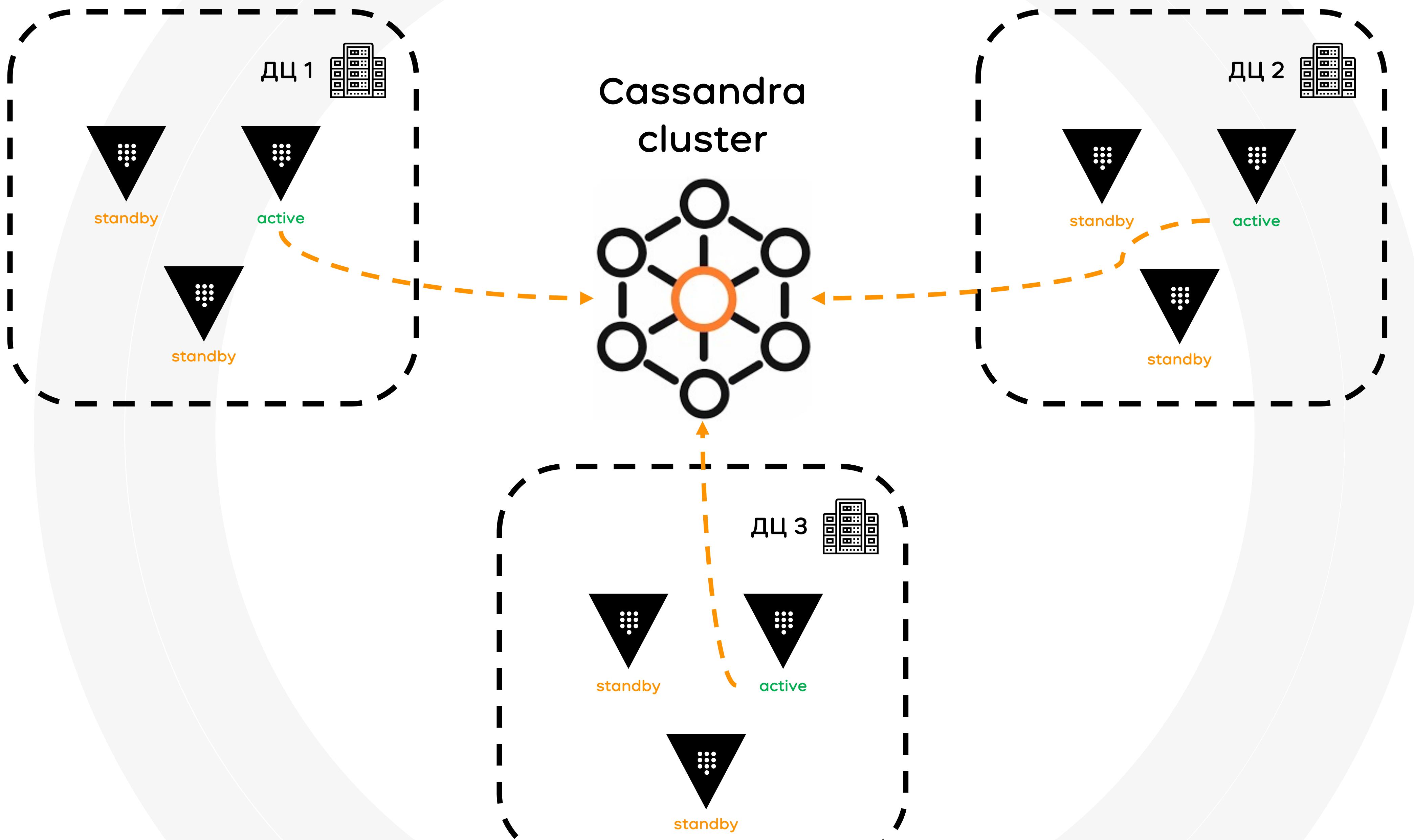
- Собираем Vault из исходников
- Собираем плагин для аутентификации
- Обновления сложнее, чем обычно (нужно наложить патчи, проверить, что всё хорошо)
- Теоретический риск, что HashiCorp поменяет структуру хранения данных

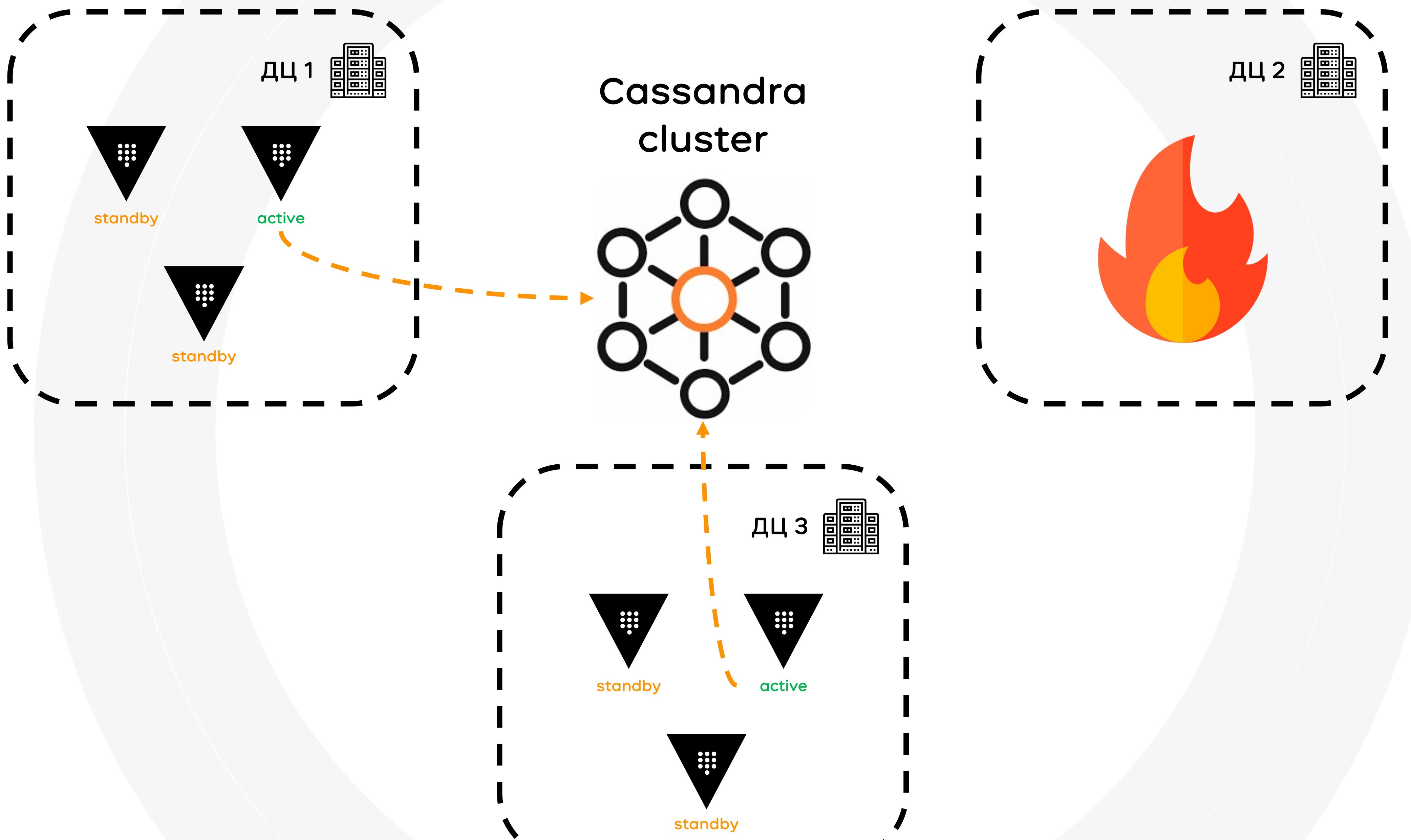


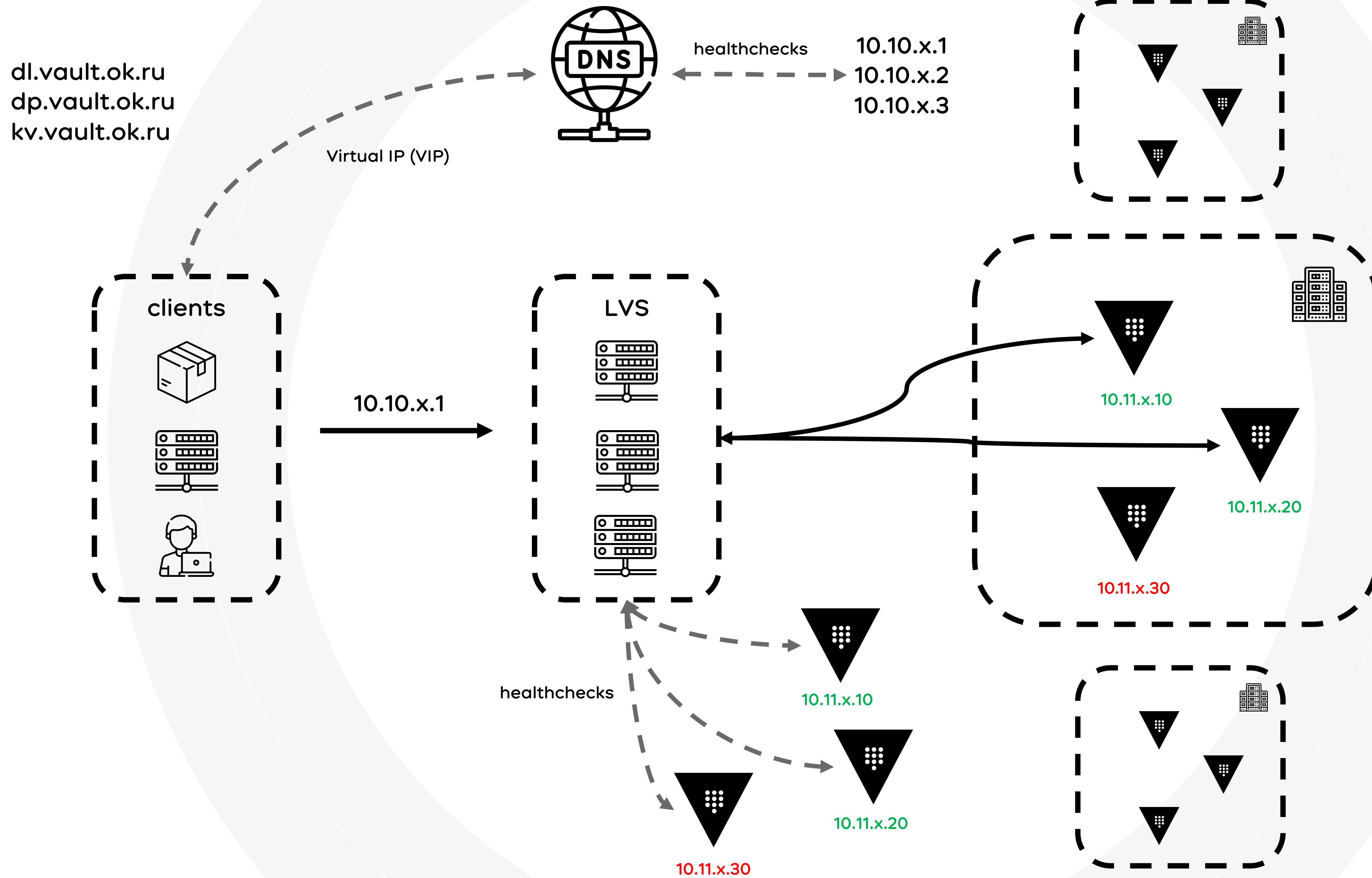
Что же получилось?

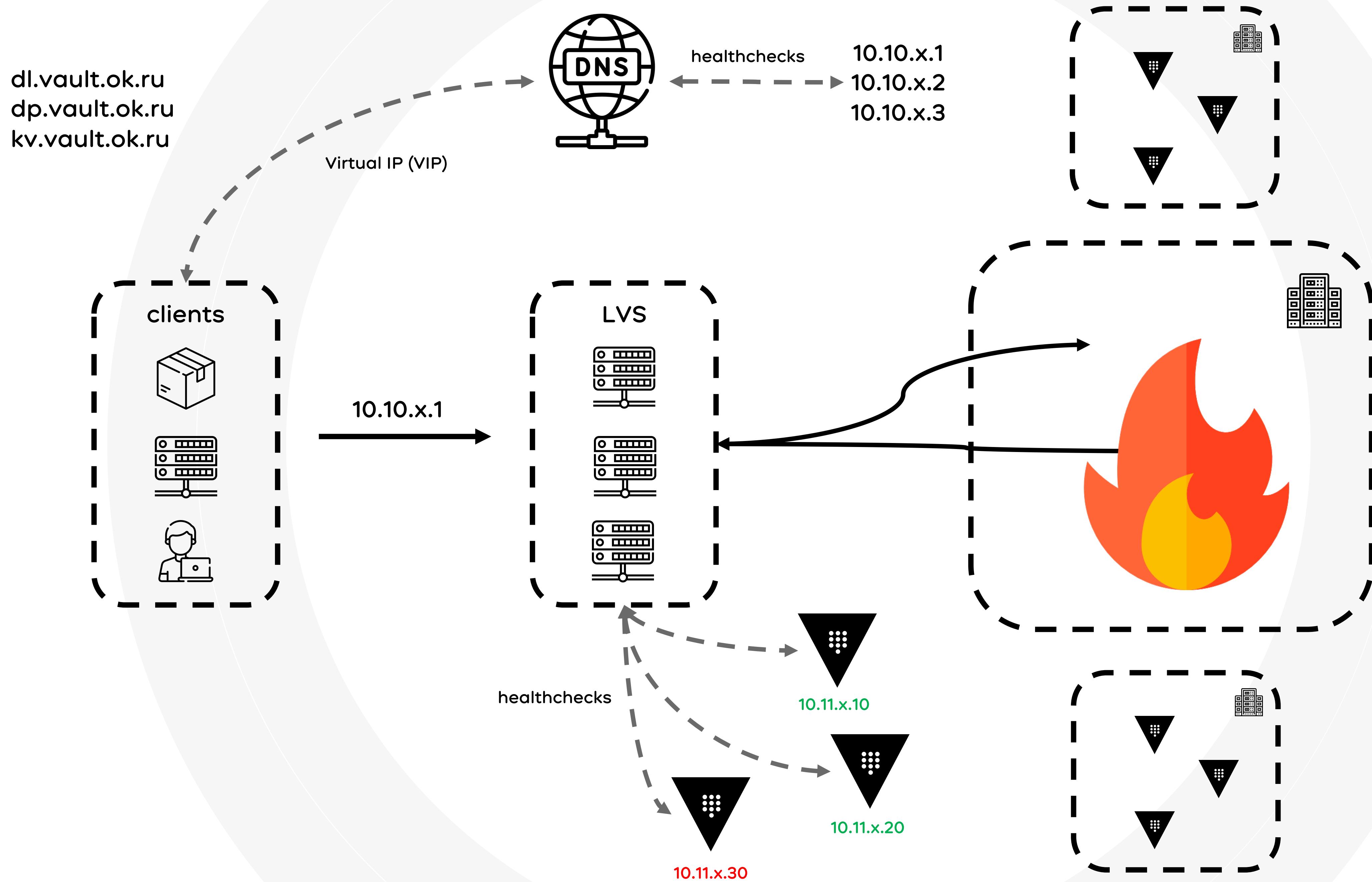
Cassandra ring













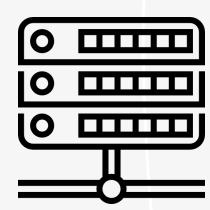
dl.vault.ok.ru
dp.vault.ok.ru
kv.vault.ok.ru

Virtual IP (VIP)



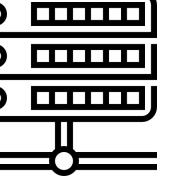
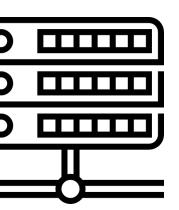
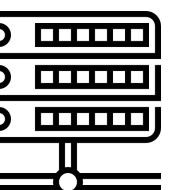
healthchecks
10.10.x.1 🔥
10.10.x.2
10.10.x.3

clients



10.10.x.1

LVS

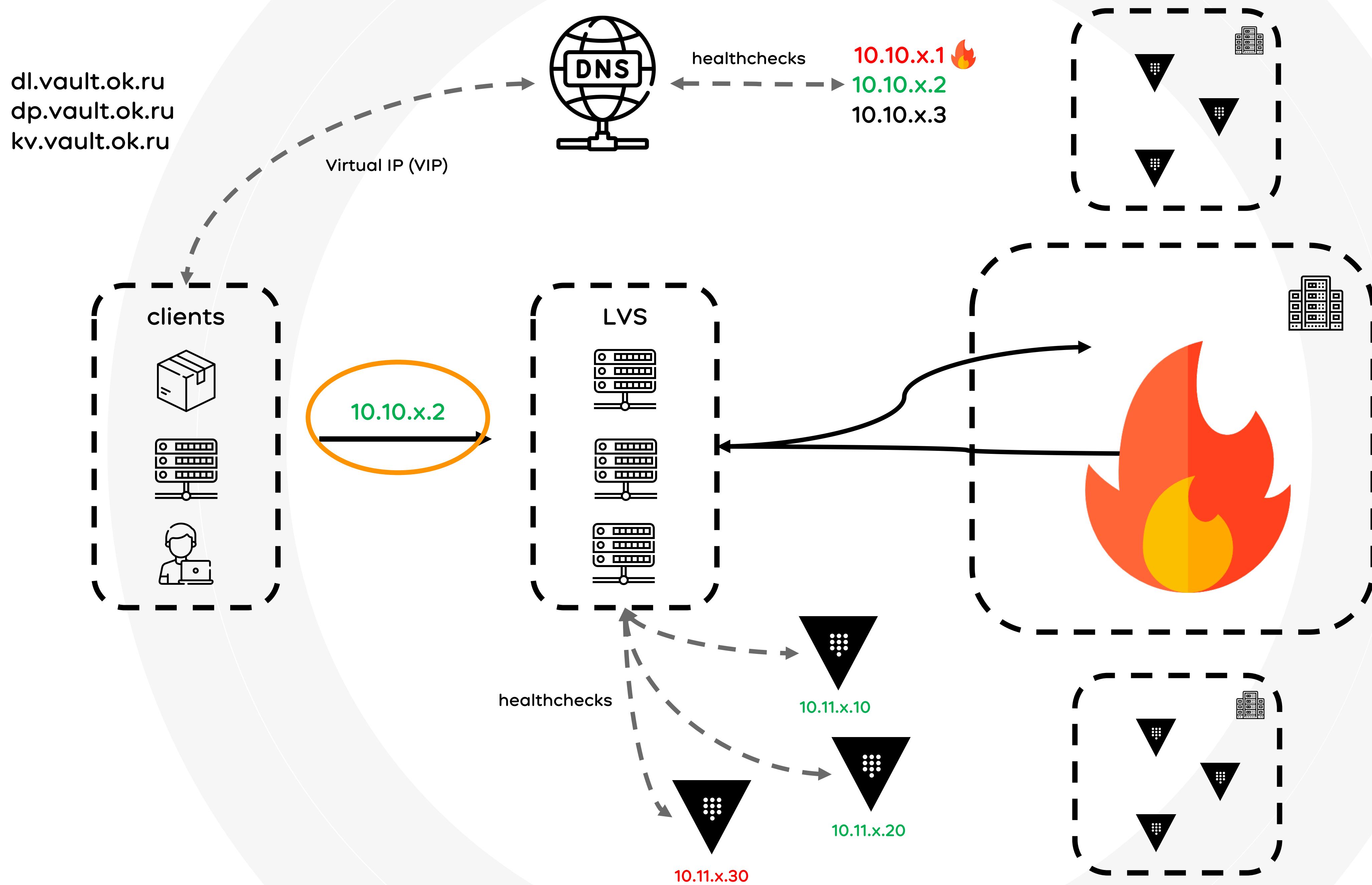


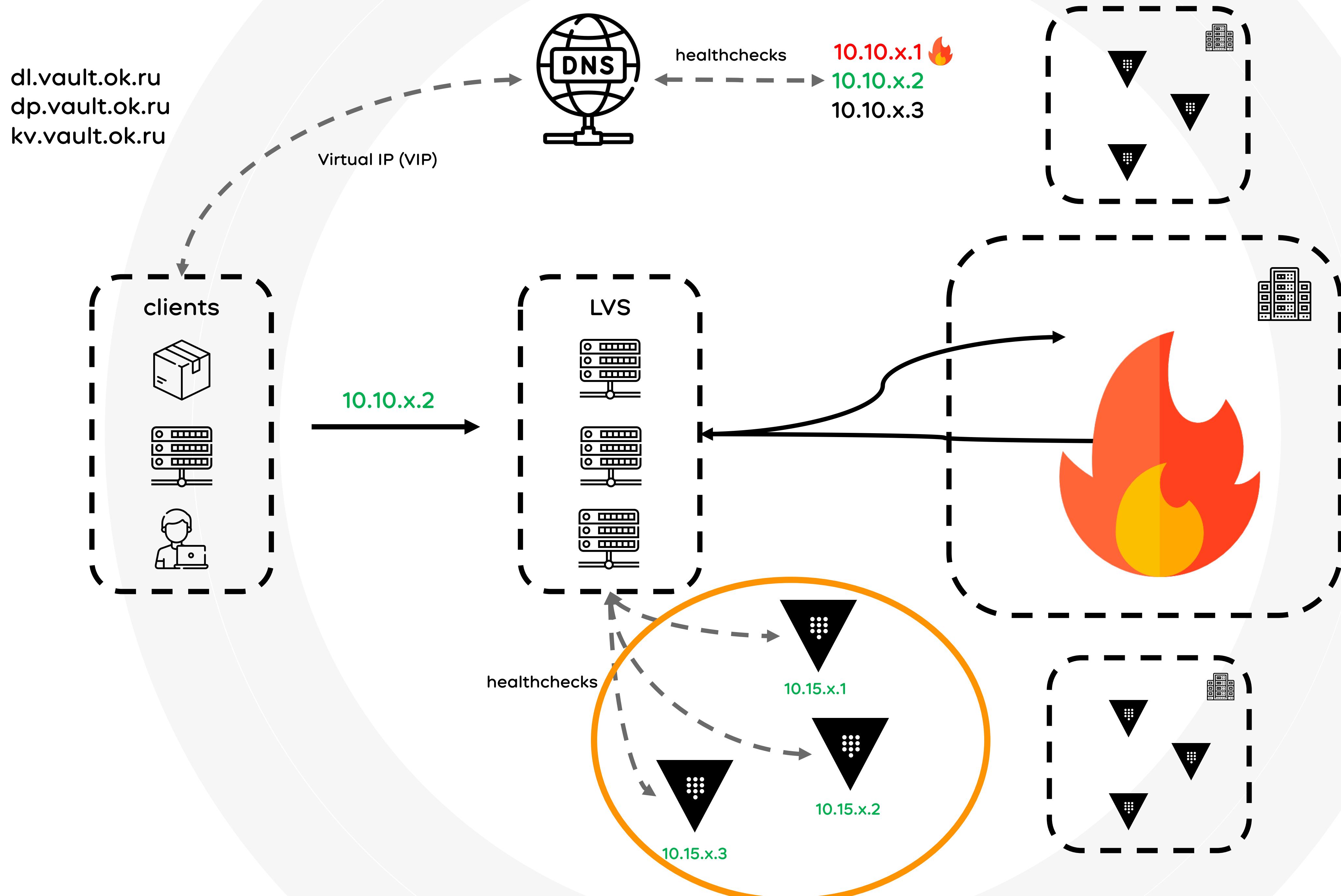
healthchecks

10.11.x.10

10.11.x.20

10.11.x.30





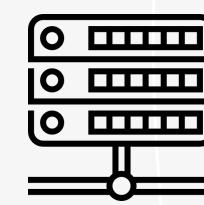
dl.vault.ok.ru
dp.vault.ok.ru
kv.vault.ok.ru

Virtual IP (VIP)

healthchecks

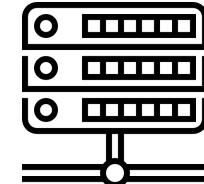
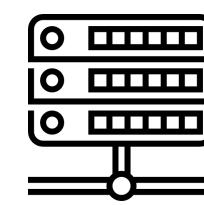
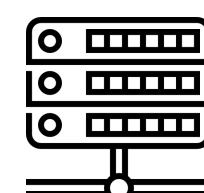
10.10.x.1 🔥
10.10.x.2
10.10.x.3

clients



10.10.x.2

LVS

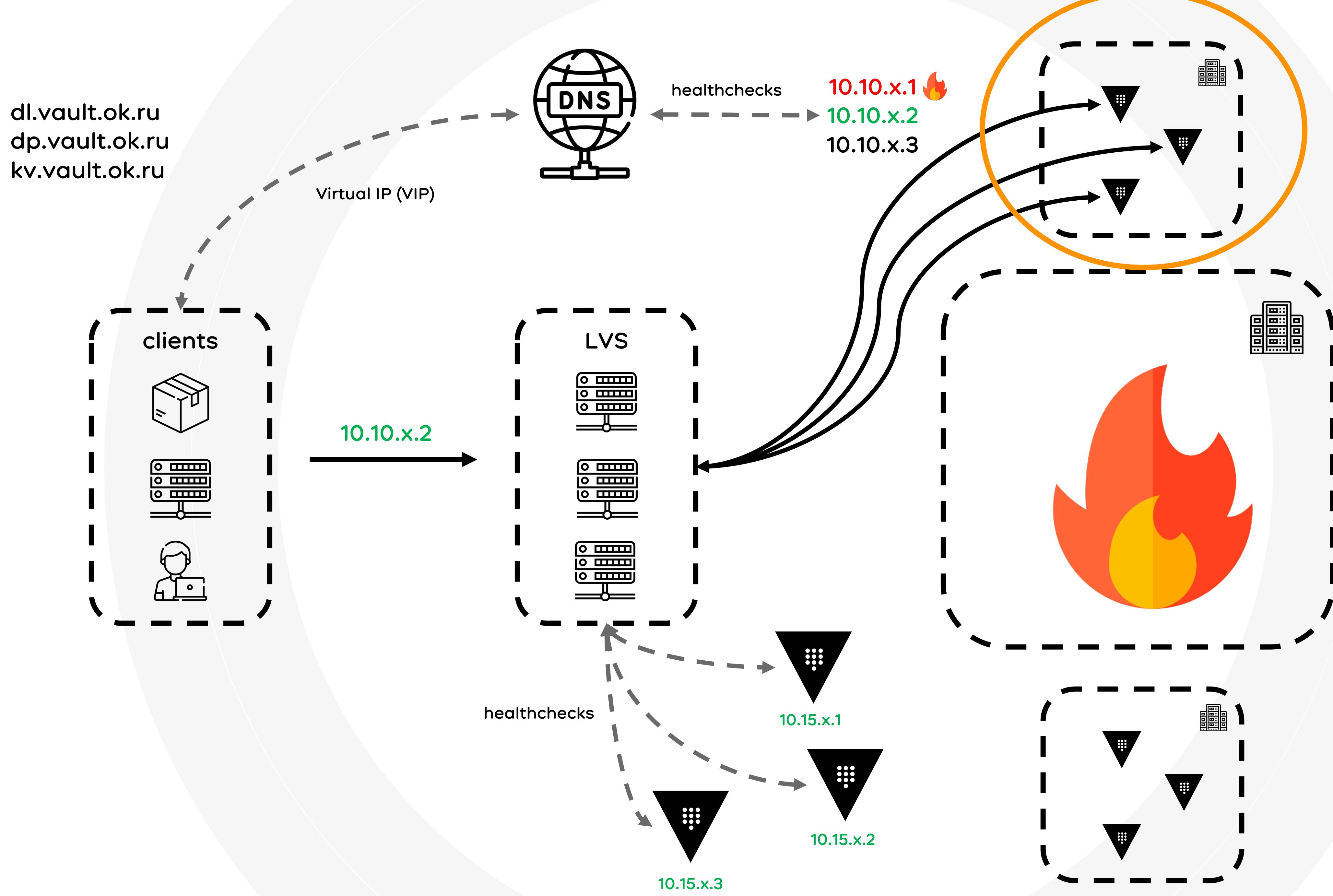


healthchecks

10.15.x.1

10.15.x.2

10.15.x.3

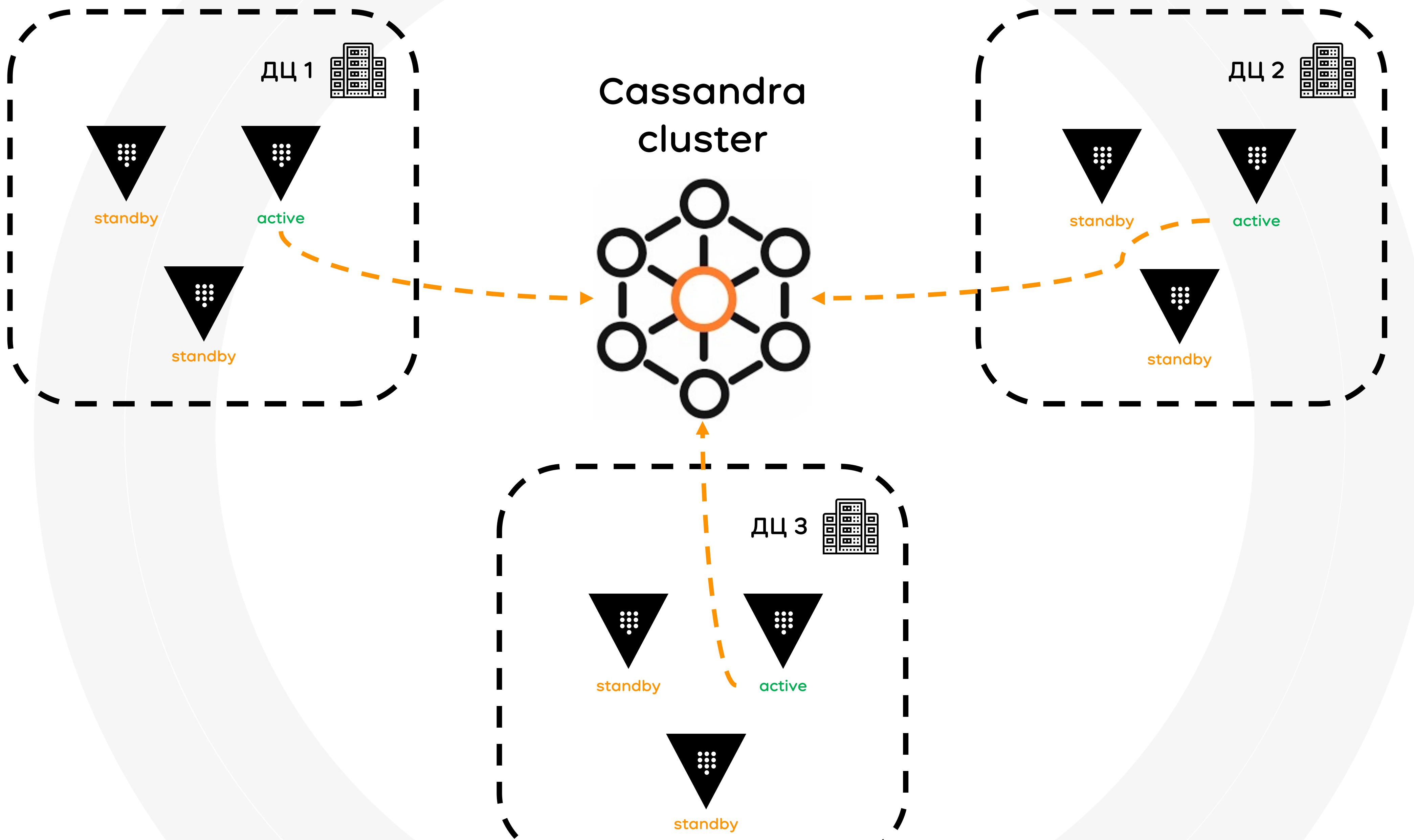


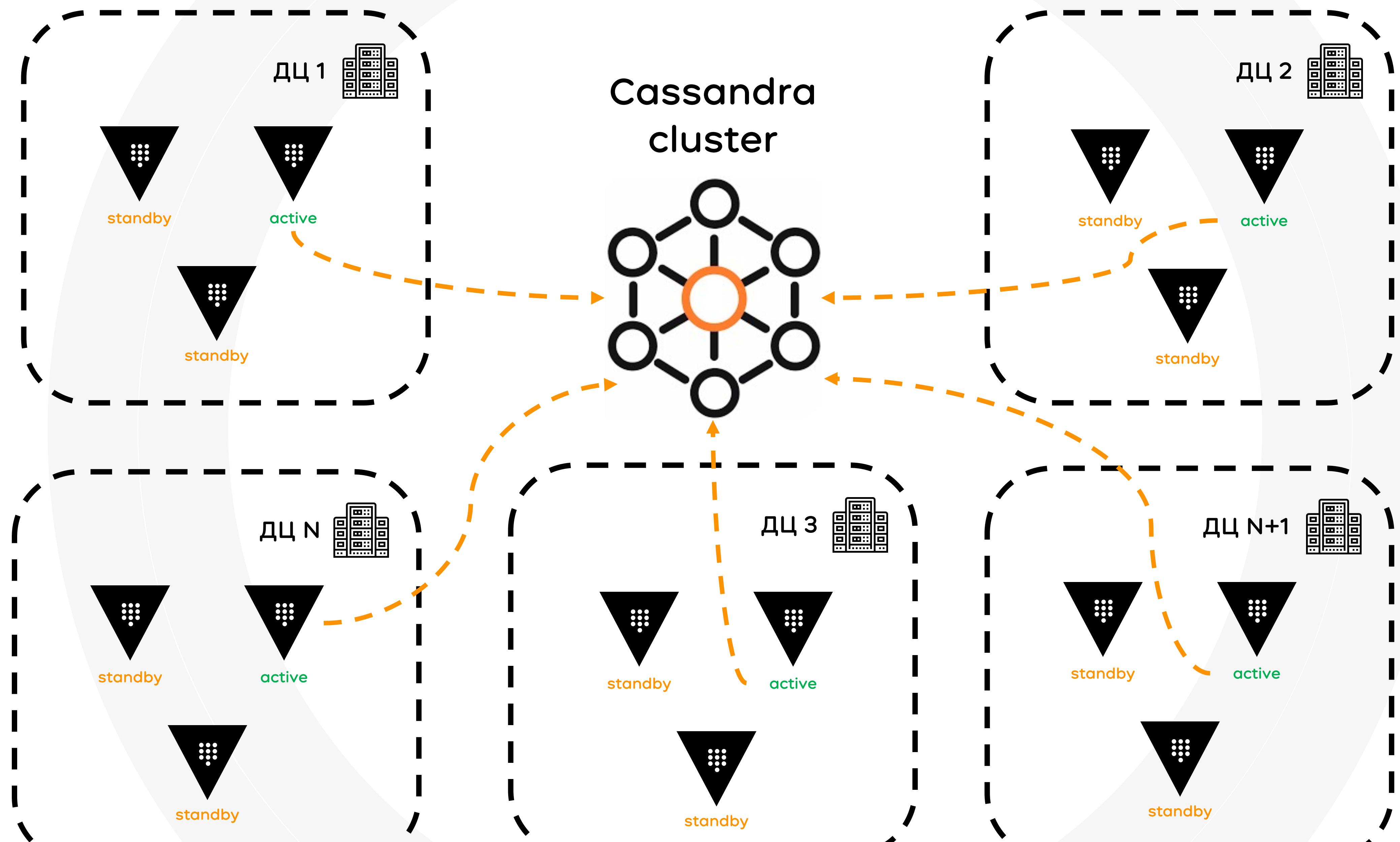


- Интеграция с нашими системами
- Выписка сертификатов
- Отказ дата-центра
- Масштабирование



OK







- Интеграция с нашими системами
- Выписка сертификатов
- Отказ дата-центра
- Масштабирование

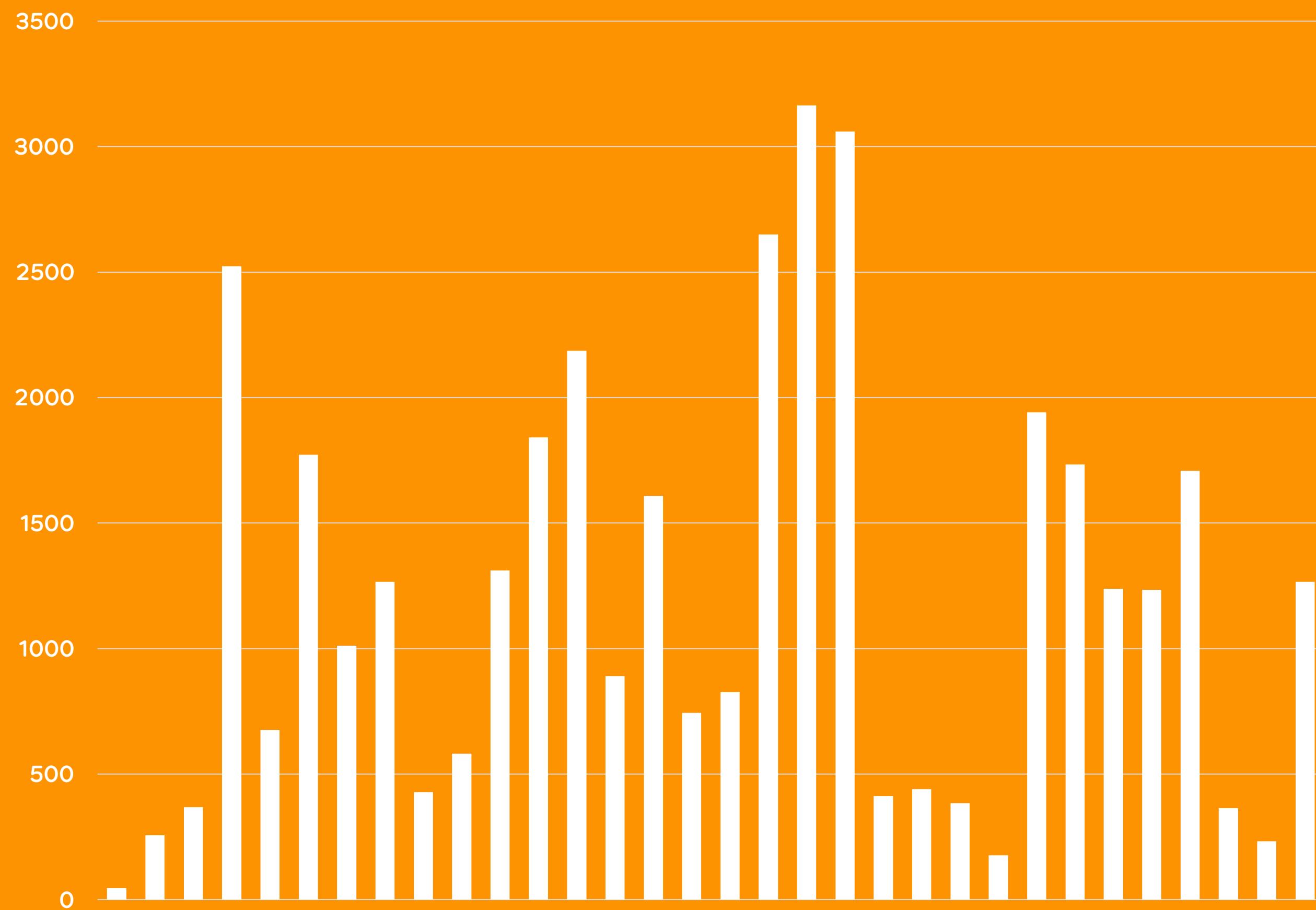




Немного цифр, куда же без них

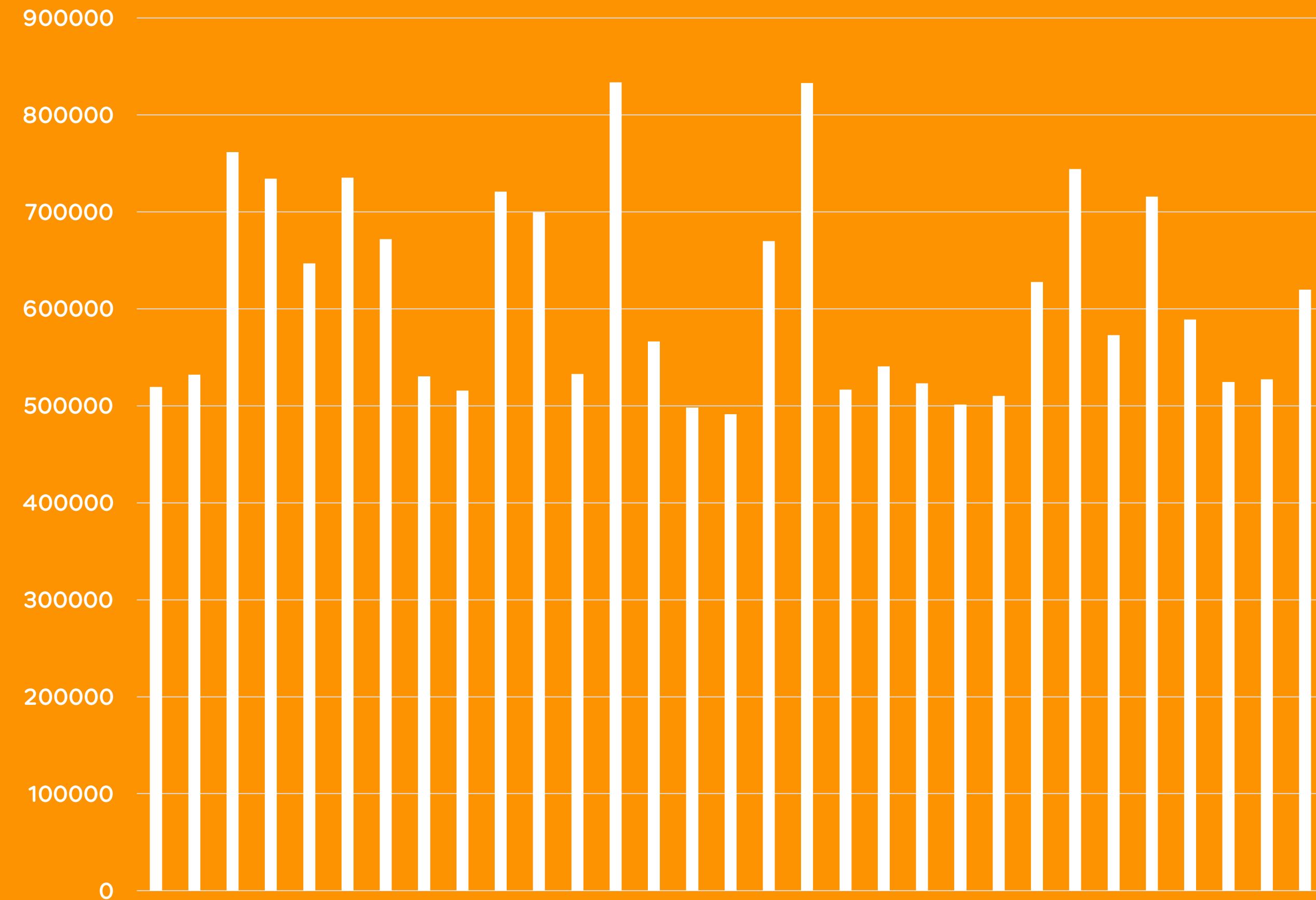
1.5K

сертификатов в сутки

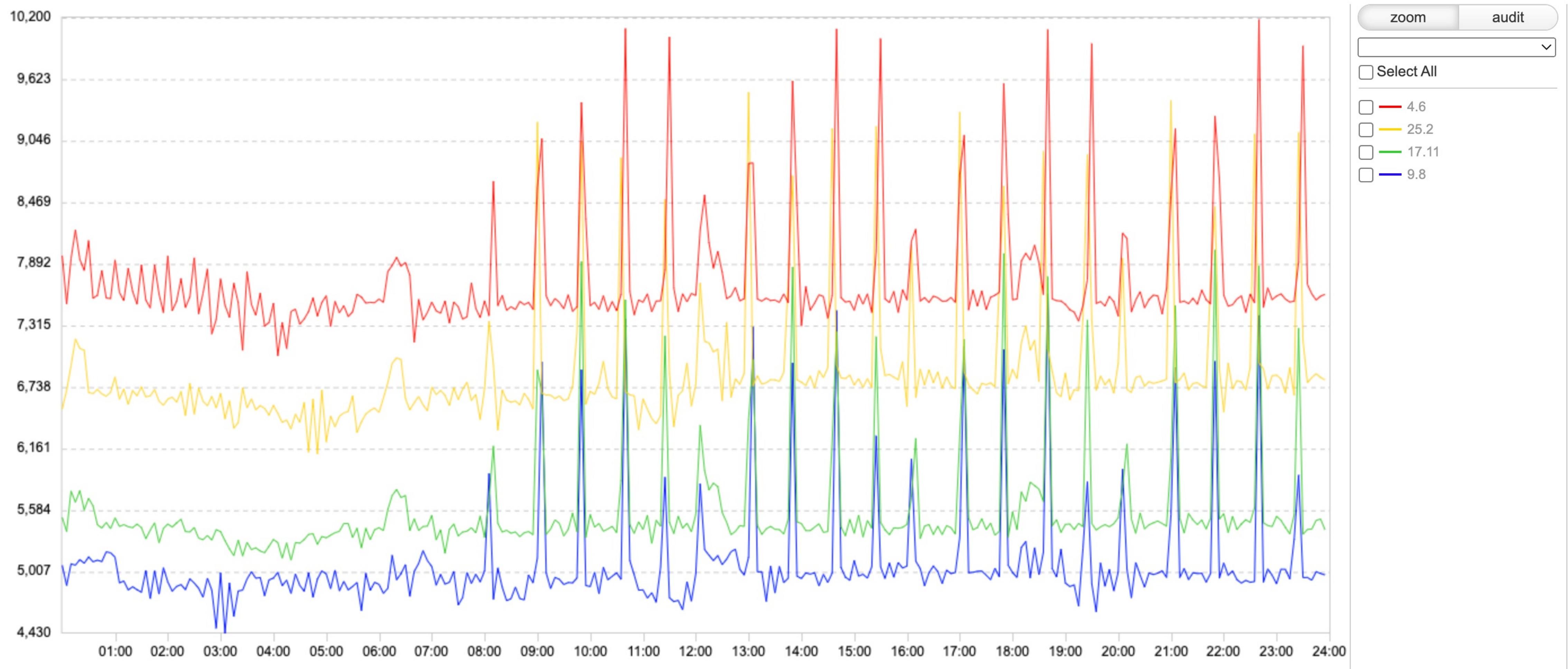


500K

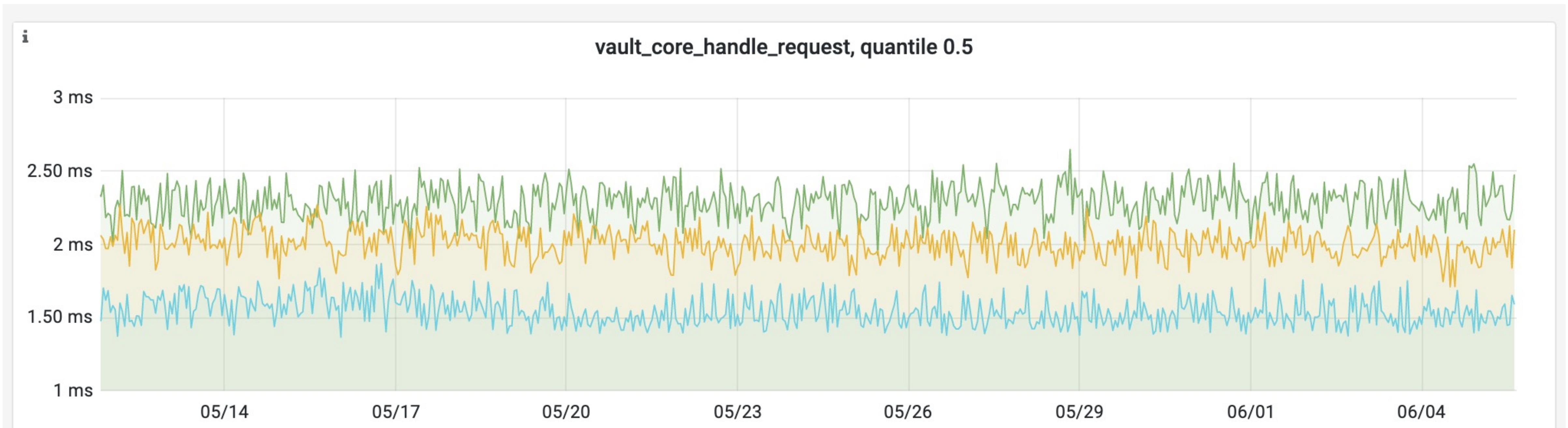
чтений секретов в сутки



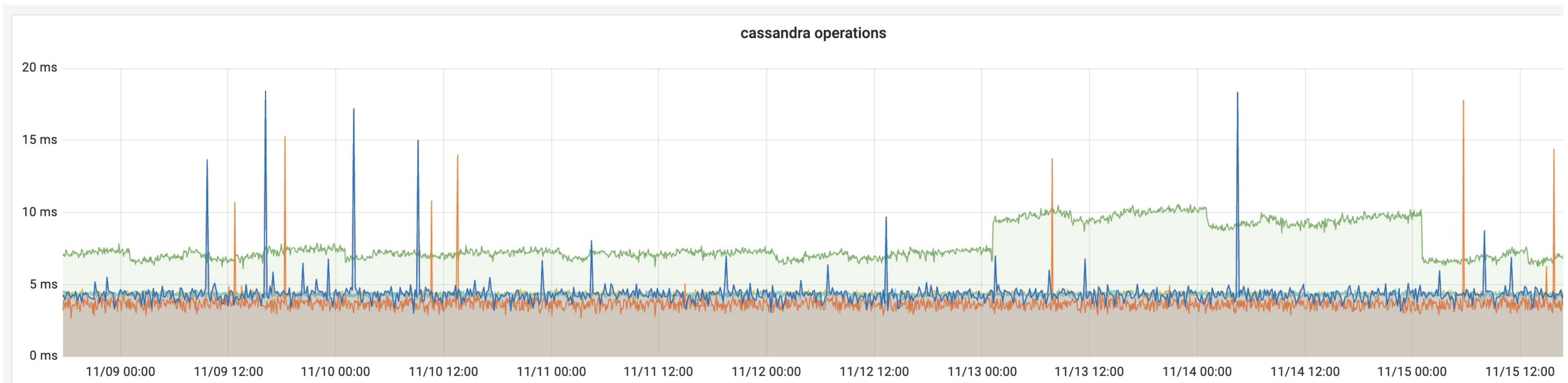
Популярность сервиса растет



Время обработки запроса



Время запроса в Cassandra





И напоследок

- HashiCorp Vault – отличный продукт с хорошей модульной архитектурой
- Vault помог нам повысить безопасность и сократить ручную работу
- Не бойтесь смотреть код продуктов, с которым работаете
- Не стесняйтесь модифицировать софт под свои задачи

Спасибо!



Иван Буймов



Артём Александров



Олег Анастасьев



Вадим Цесько



@byumov



@byumov



ivan@buymov.ru

Присоединяйтесь!



oktech.ru

Блог на Хабре



habr.com/ru/company/odnoklassniki/blog/





Слайды и ссылки



Оценить доклад