Key Management for Data-at-Rest Encryption in Percona Server for MongoDB

Agenda

- Percona Server for MongoDB
- encryption
- key management systems overview
- data-at-rest encryption setup
- master key rotation
- key state polling for KMIP
- secret versioning in Vault
- closing thoughts
- Q&A

Percona Server for MongoDB

- drop-in replacement for MongoDB Community Edition
- implements additional features:
  - in-memory storage engine
  - authentication and authorization: LDAP, OIDC, etc.
  - backups
  - data-at-rest encryption

Encryption Types

- in-transit: encrypted connections
  - client⇔node
  - node⇔node
- in-use: encrypted data in memory
- at-rest: encrypted data files on disk

Disk Encryption vs Data-at-Rest Encryption

Disk Encryption:

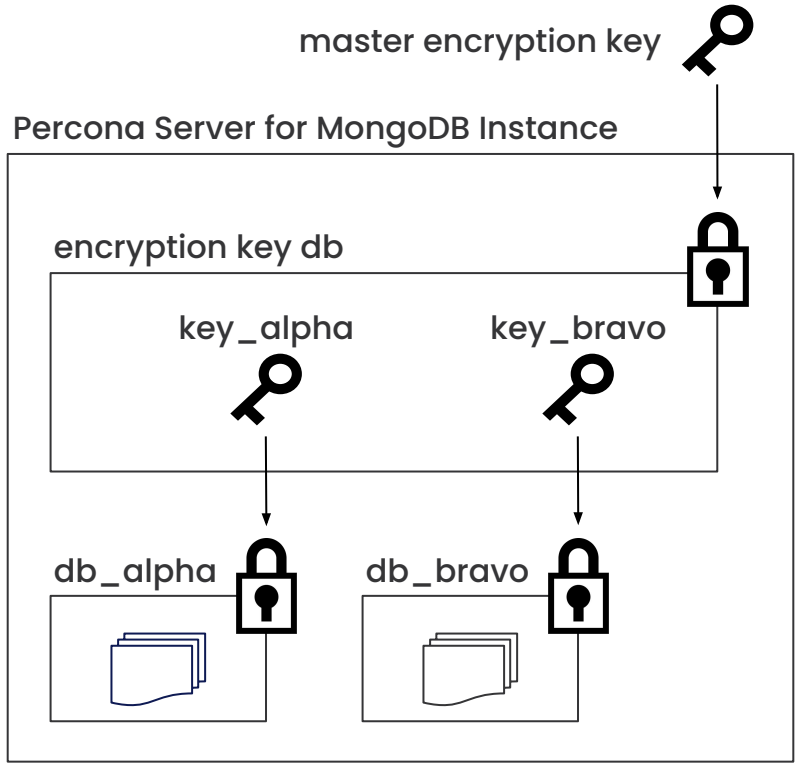
- All files on the disk are encrypted
- Copying: target media needs encryption
- KMS integration = ?

Data-at-Rest Encryption:

- Only database files are encrypted
- Copying: the files remain encrypted
- KMS integration: implemented in Percona Server for MongoDB

Data-at-Rest Encryption

- document ∈ collection ∈ database
- AES-256-CBC or AES-256-GCM
- one key per database
- encryption key database
  - keeps per-database keys
  - encrypted with a master key
- clean data directory



Key Management

- encryption key file
  - base64-encoded key
  - not for production
- KMIP server
- HashiCorp's Vault

configuration excerpt

```
security:
  enableEncryption: true
  encryptionKeyFile: /path/to/key/file
  kmip:
    <kmip_settings>
  vault:
    <vault_settings>
```

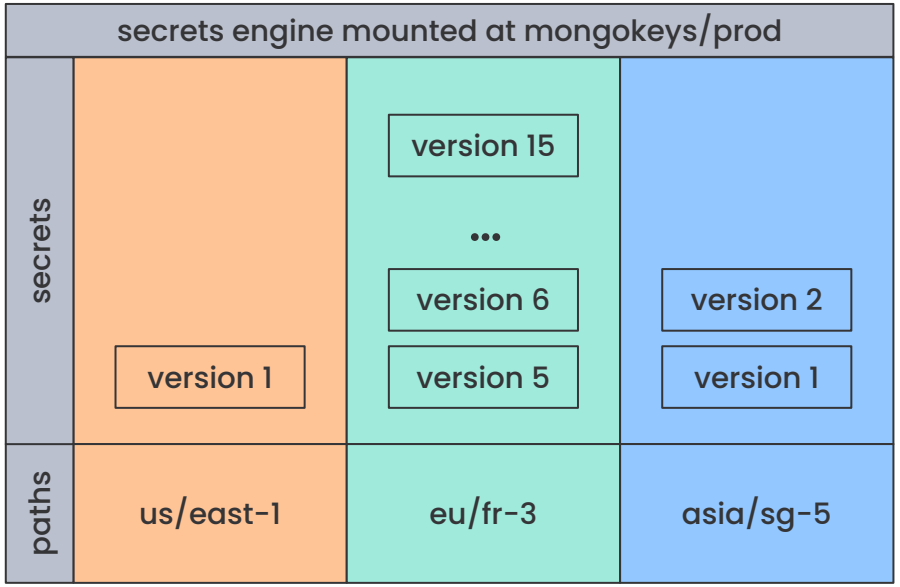
KMIP: Key Management Interoperability Protocol

- managed objects:
  - public/private/symmetric key
  - certificate/etc
- operations
  - create/register/get/etc
  - activate/revoke/destroy/etc
- binary transfer encoding
- identifier ⇨ managed object
- state attribute

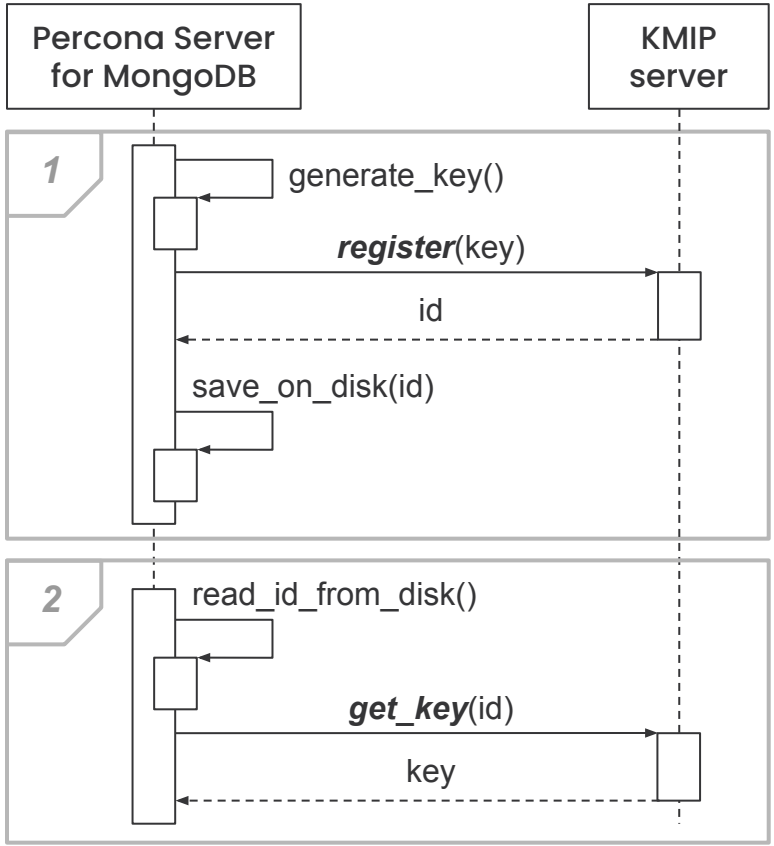
id	symmetric key
4f23b48f…	0a48f6e51c202158…
db2dc512…	d955cb3dc35ba0e8…
10123ad0…	9c8ae5582666005a…
71e040a2…	c8234501e2ef0f89…

HashiCorp's Vault

- versioned key-value secrets engine
- secret = set of key-value pairs
  - {<smth>: <master_key_base64}
- absolute secret path:
  - mongokeys/prod/data/asia/sg-5
- {absolute_path, version} ⇨ secret



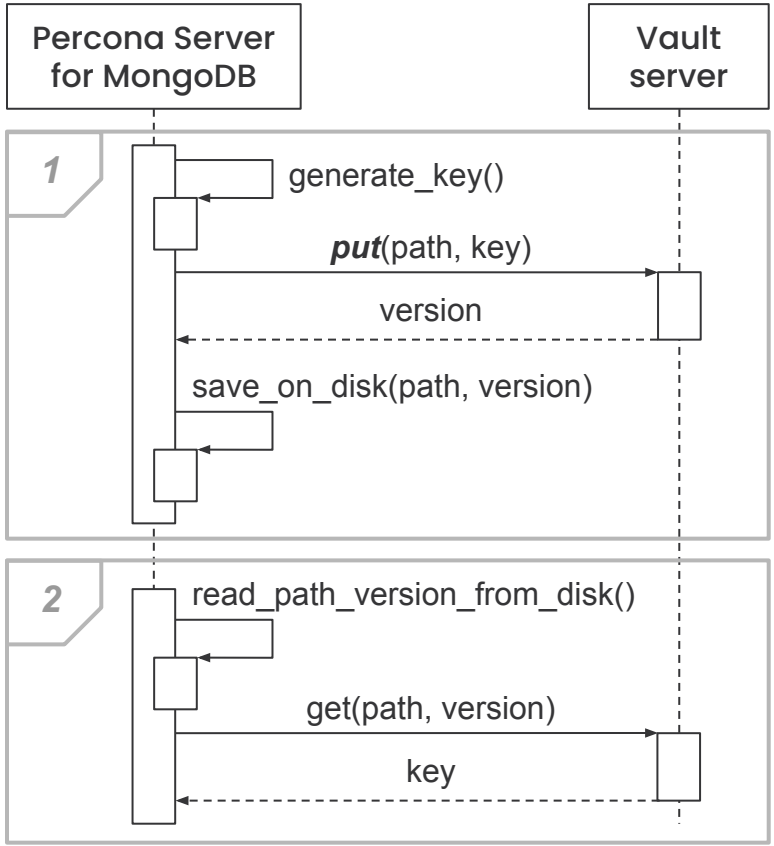
KMIP: Initial & Subsequent Starts



configuration excerpt

```
security:
  enableEncryption: true
  kmip:
    serverName: my-kmip.acme.com
    port: 5696
    serverCAFile: /path/to/ca.crt
    clientCertificateFile: client.pem
```

Vault: Initial & Subsequent Starts



configuration excerpt

```
security:
  enableEncryption: true
  vault:
    serverName: my-vault.acme.com
    port: 8200
    serverCAFile: /path/to/ca.crt
    tokenFile: /path/to/vault.token
    secret: mongokeys/prod/data/asia/sg-5
```

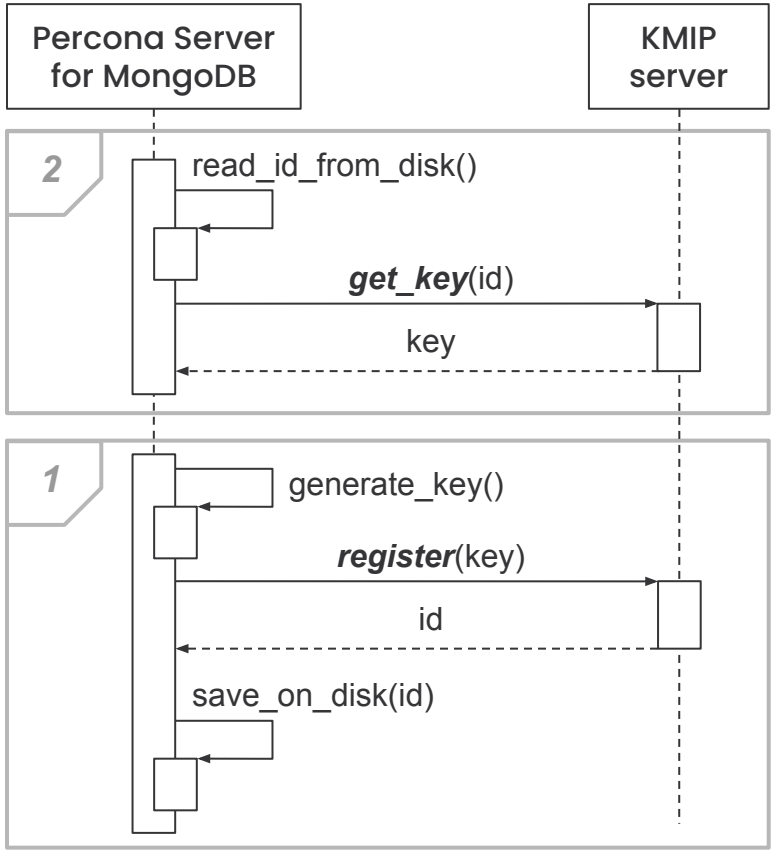
KMIP: Master Key Rotation

- stop the service
- set `rotateMasterKey`
- start the service
  - no query serving
- wait until the service exits
- unset `rotateMasterKey`
- start the service

configuration excerpt

```
security:
  enableEncryption: true
  kmip:
    serverName: my-kmip.acme.com
    port: 5696
    serverCAFile: /path/to/ca.crt
    clientCertificateFile: client.pem
    rotateMasterKey: true
```

KMIP: Master Key Rotation (cont.)

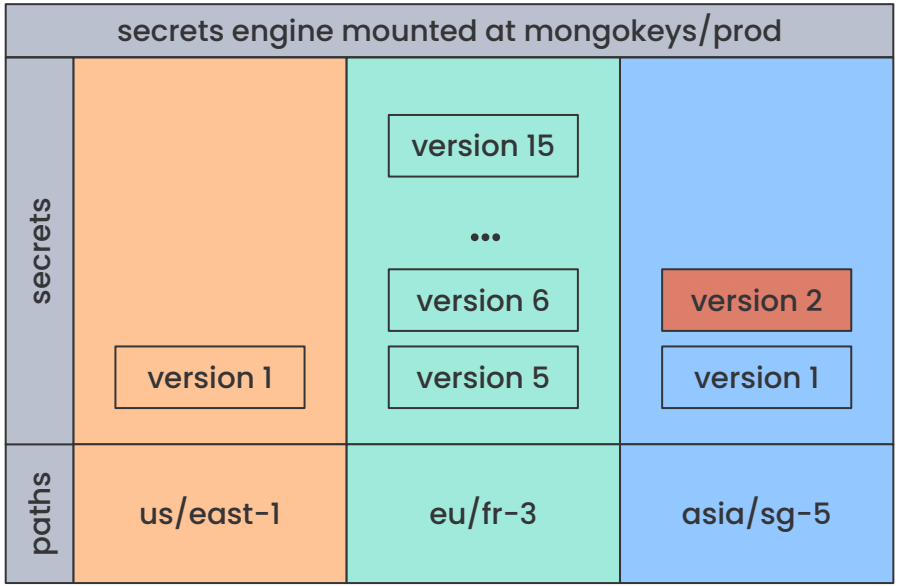


configuration excerpt

```
security:
  enableEncryption: true
  kmip:
    serverName: my-kmip.acme.com
    port: 5696
    serverCAFile: /path/to/ca.crt
    clientCertificateFile: client.pem
    rotateMasterKey: true
```

Master Encryption Key to Node Backward Mapping

Vault

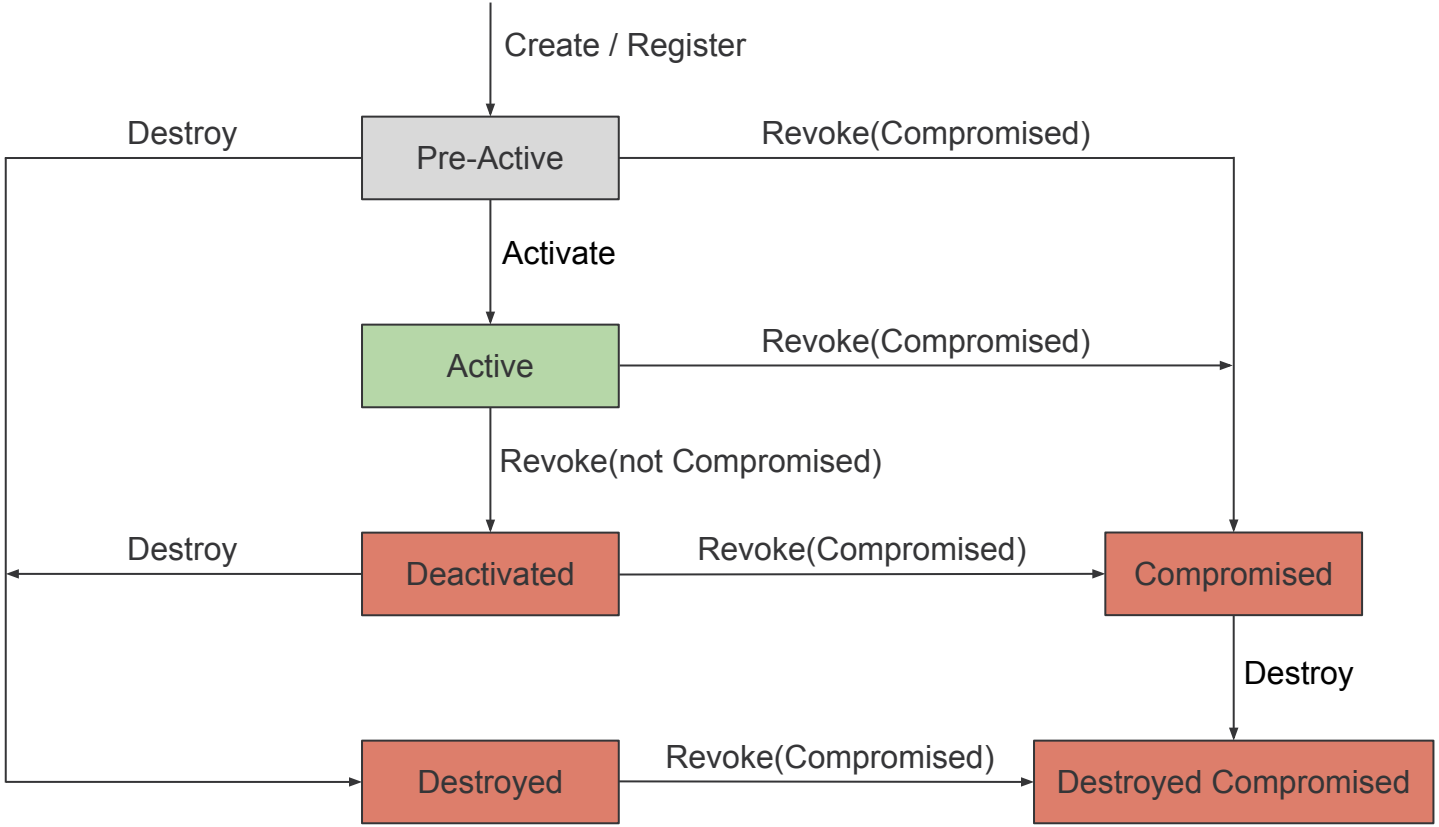


KMIP

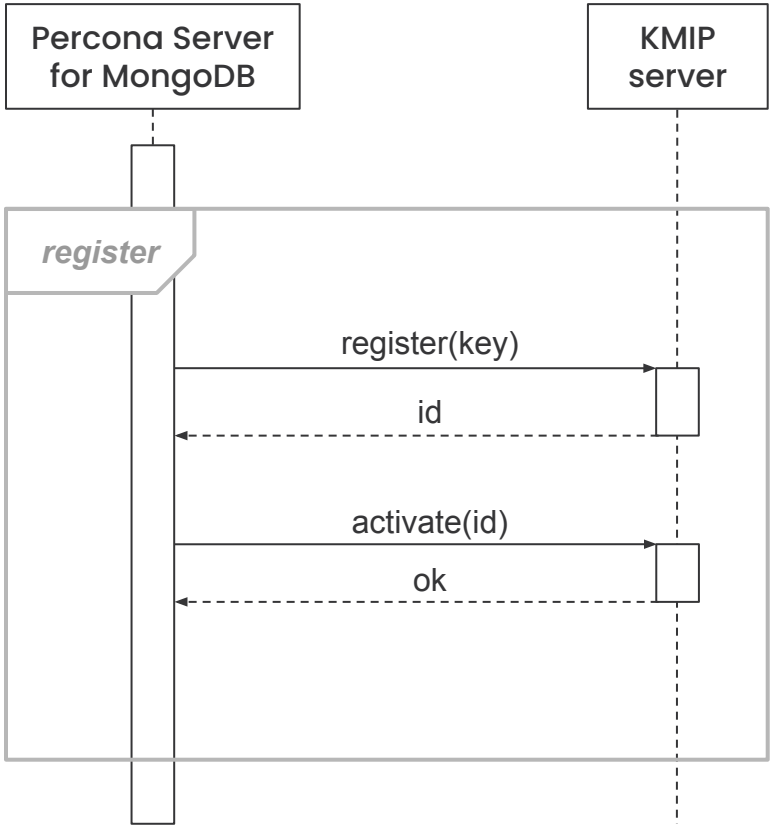
id	symmetric key
4f23b48f…	0a48f6e51c202158…
db2dc512…	d955cb3dc35ba0e8…
10123ad0…	9c8ae5582666005a…
71e040a2…	c8234501e2ef0f89…

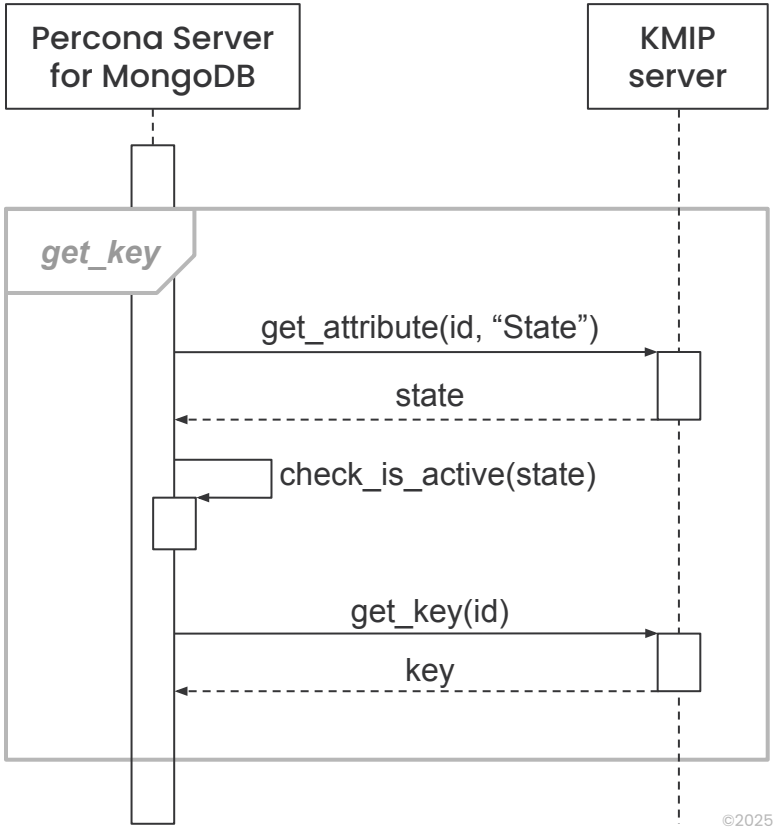
`mongokeys/prod/data/asia/sg-5`

KMIP: State Attribute

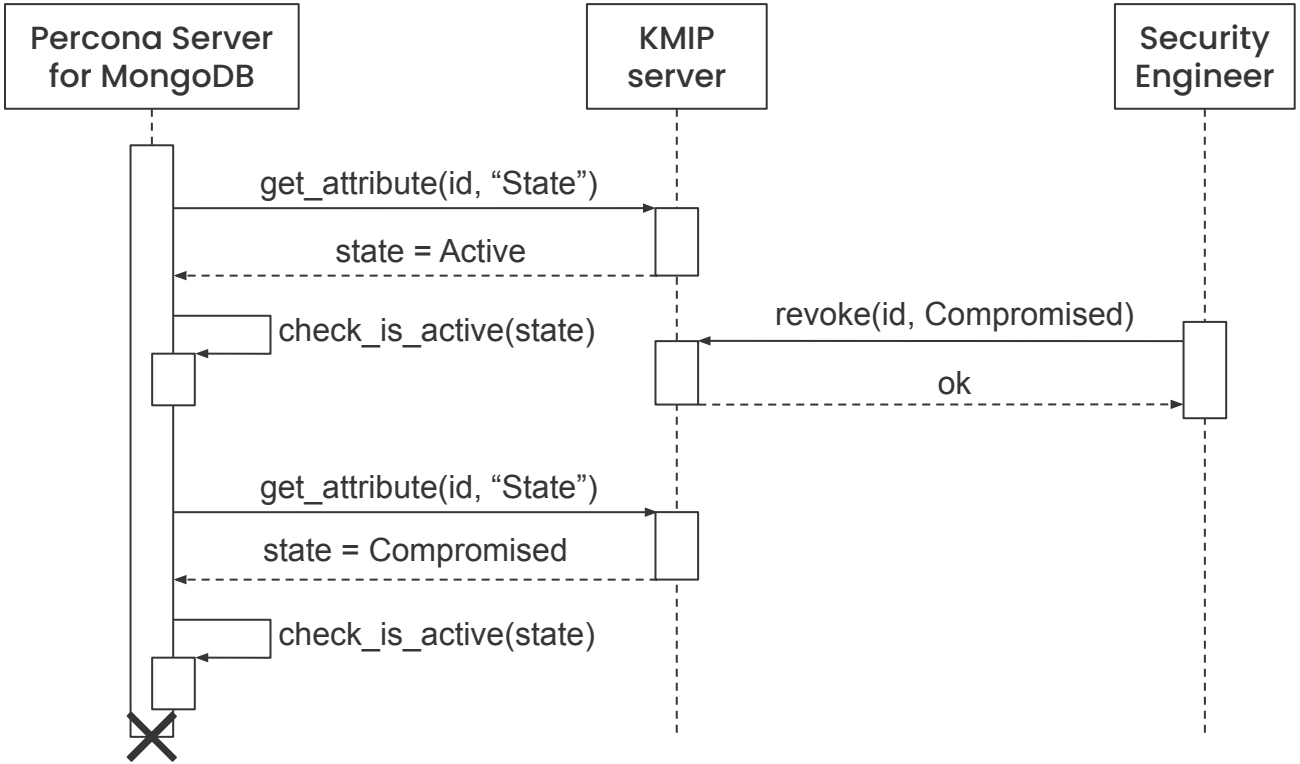


KMIP: Register & Get Key



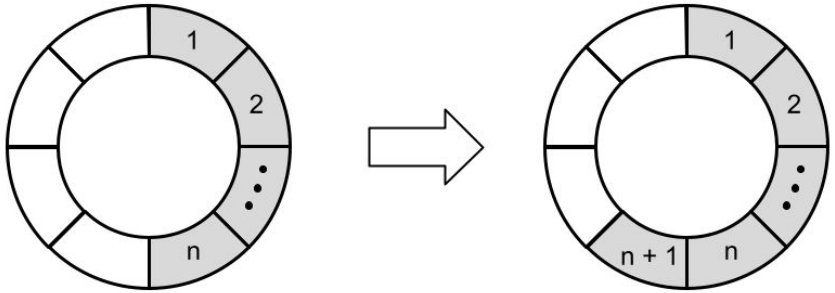


KMIP: Key State Polling

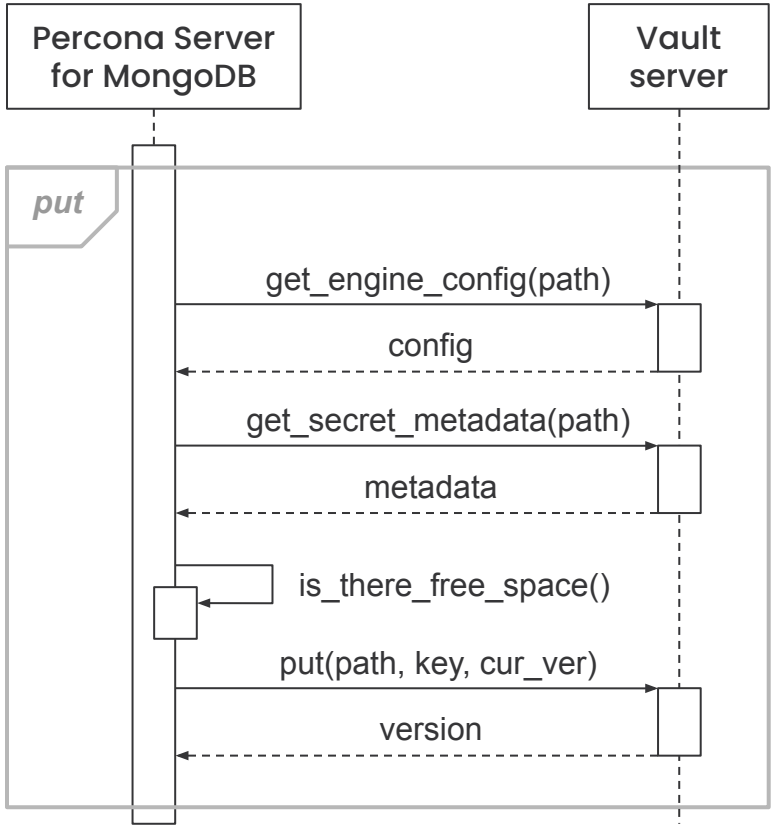


Vault: Max Versions and Overwriting

- removes the oldest version after `max_versions` is reached
- `max_versions = 10` by default
- may result in data loss
- set `max_versions` in:
  - secret engine's config
  - secret's metadata
- `max_versions = 24,000`



Vault: Check Max Versions



configuration excerpt

```
security:
  enableEncryption: true
  vault:
    serverName: my-vault.acme.com
    port: 8200
    serverCAFile: /path/to/ca.crt
    tokenFile: /path/to/vault.token
    secret: mongokeys/prod/data/asia/sg-5
    checkMaxVersions: true
```

Vault: Permission and Max Versions Configuration

- grant permissions for engine's config and secret's metadata:

```
path "mongokeys/prod/data/*" {
  capabilities = ["create", "read", "update"]
}
path "mongokeys/prod/metadata/*" {
  capabilities = ["read"]
}
path "mongokeys/prod/config" {
  capabilities = ["read"]
}
```

- configure max versions:
  - `vault kv metadata put -max-versions=24000 \ -mount=mongokeys/prod asia/sg-5`
  - `vault write mongokeys/prod/config max_versions=24000`

Implementations and Licensing

KMIP:

- Multiple commercial implementations
- PyKMIP: Apache 2.0
  - only for testing
  - not maintained
- OVH kmip-go: Apache 2.0

Vault:

- HashiCorp's Vault CE:
  - Business Source License
  - use grant for production
- HashiCorp's Vault EE:
  - commercial license
  - implements KMIP
- OpenBao: Mozilla Public License

Clothing thoughts

- factor in data-at-rest encryption when planning a deployment
- do master key rotations at least once a year
- be prepared for an unplanned master key rotation
- Vault: tune permissions and increase maximum number of versions
- contact Percona for assistance

Resources

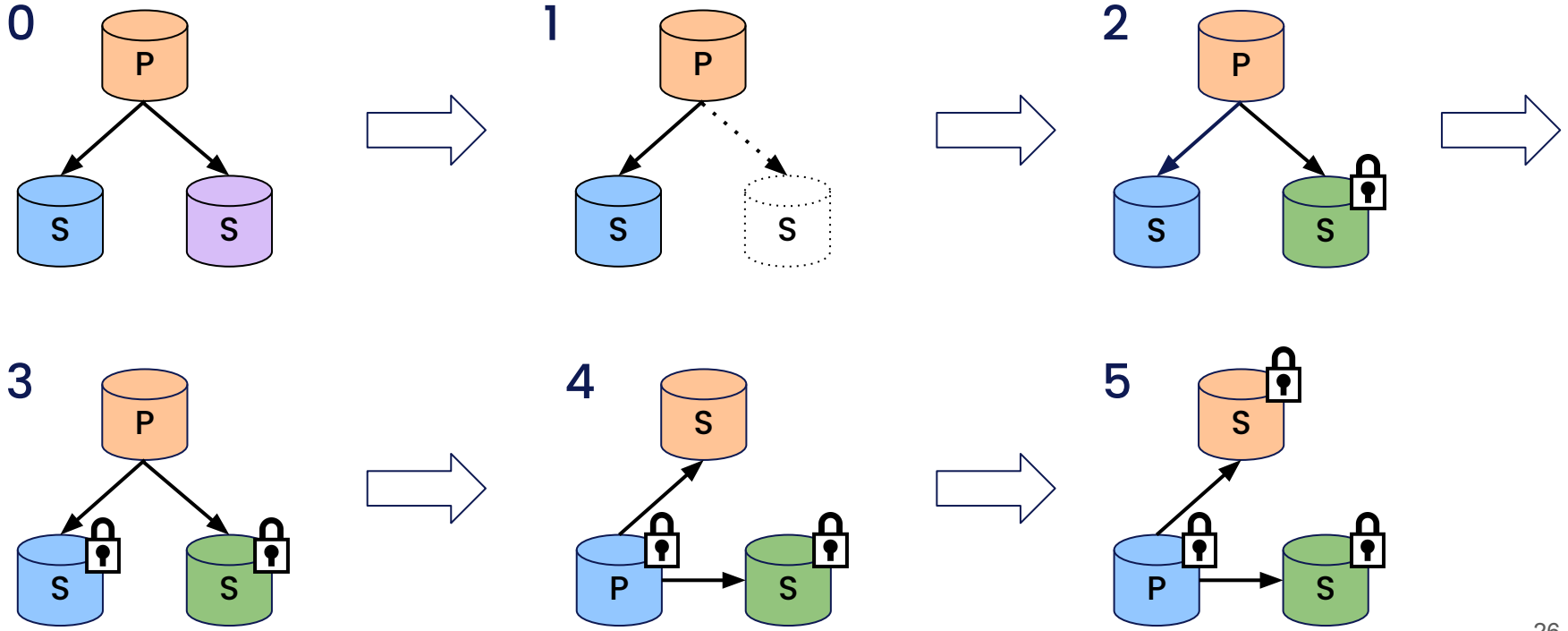
- KMIP: https://bit.ly/3Pt1Bhw
- OVH KMIP: https://github.com/ovh/kmip-go
- HashiCorp's Vault: https://www.hashicorp.com/en/products/vault
- OpenBao: https://openbao.org
- Percona Blog: https://www.percona.com/blog/
  - key state polling: https://bit.ly/3UxF5pJ
  - vault key overwriting prevention: https://bit.ly/3JmfoGe
- Percona Server for MongoDB:
  - documentation: https://bit.ly/3TpmlHQ
  - community forum: https://bit.ly/3x1ROZl

PERCONA

Thank you

Extra

Encrypting a Running Replica Set



KMIP: Master Key Rotation & Existing Key

- stop the service
- set `rotateMasterKey`
- set `keyIdentifier`
- start the service
- wait until the service exits
- unset `rotateMasterKey`
- remove `keyIdentifier`
- start the service

configuration excerpt

```
security:
  enableEncryption: true
  kmip:
    serverName: my-kmip.acme.com
    port: 5696
    serverCAFile: /path/to/ca.cert
    clientCertificateFile: client.pem
    rotateMasterKey: true
    keyIdentifier: foobar
```

Vault: Master Key Rotation & Existing Key

- stop the service
- set `rotateMasterKey`
- set `secret` and `secretVersion`
- start the service
- wait until the service exits
- unset `rotateMasterKey`
- remove `secretVersion`
- start the service

configuration excerpt

```
security:
  enableEncryption: true
  vault:
    serverName: my-vault.acme.com
    port: 8200
    serverCAFile: /path/to/ca.cert
    tokenFile: /path/to/vault.token
    secret: mongokeys/prod/data/asia/jp-7
    rotateMasterKey: true
    secretVersion: 42
```

Vault: Master Key Rotation & New Secret Path

- stop the service
- set `rotateMasterKey`
- change `secret`
- start the service
- wait until the service exits
- unset `rotateMasterKey`
- start the service

configuration excerpt

```
security:
  enableEncryption: true
  vault:
    serverName: my-vault.acme.com
    port: 8200
    serverCAFile: /path/to/ca.cert
    tokenFile: /path/to/vault.token
    secret: mongokeys/prod/data/asia/am-9
    rotateMasterKey: true
```

KMIP: Migration from Key File

- decode key from the key file
- register and active the key on a KMIP server
- set the `kmip` section in config including `keyIdentifier`
- remove `encryptionKeyFile` from config
- restart the service
- remove `keyIdentifier`

configuration excerpt

```
security:
  enableEncryption: true
  kmip:
    serverName: my-kmip.acme.com
    port: 5696
    serverCAFile: /path/to/ca.crt
    clientCertificateFile: client.pem
    keyIdentifier: foobar
```

Vault: Migration from Key File

- put the base64-encoded key form a key file to the Vault server
- set the `vault` section in config including `secret` and `secretVersion`
- remove `encryptionKeyFile` parameter
- restart the service
- remove `secretVersion`

configuration excerpt

```
security:
  enableEncryption: true
  vault:
    serverName: my-vault.acme.com
    port: 8200
    serverCAFile: /path/to/ca.crt
    tokenFile: /path/to/vault.token
    secret: mongokeys/prod/data/asia/sg-5
    secretVersion: 1
```

KMIP: Multiple Servers & Connect Retries

- total tries = `connectRetries + 1`
- sequence of connect attempts:
  - kmip-1.com
  - kmip-2.com
  - kmip-1.com
  - kmip-2.com
  - kmip-1.com
  - kmip-2.com

configuration excerpt

```
security:
  enableEncryption: true
  kmip:
    serverName: kmip-1.com, kmip-2.com
    port: 5696
    connectTimeoutMS: 15000
    connectRetries: 2
    serverCAFile: /path/to/ca.crt
    clientCertificateFile: client.pem
```