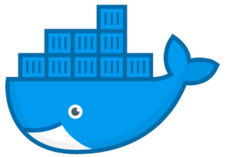
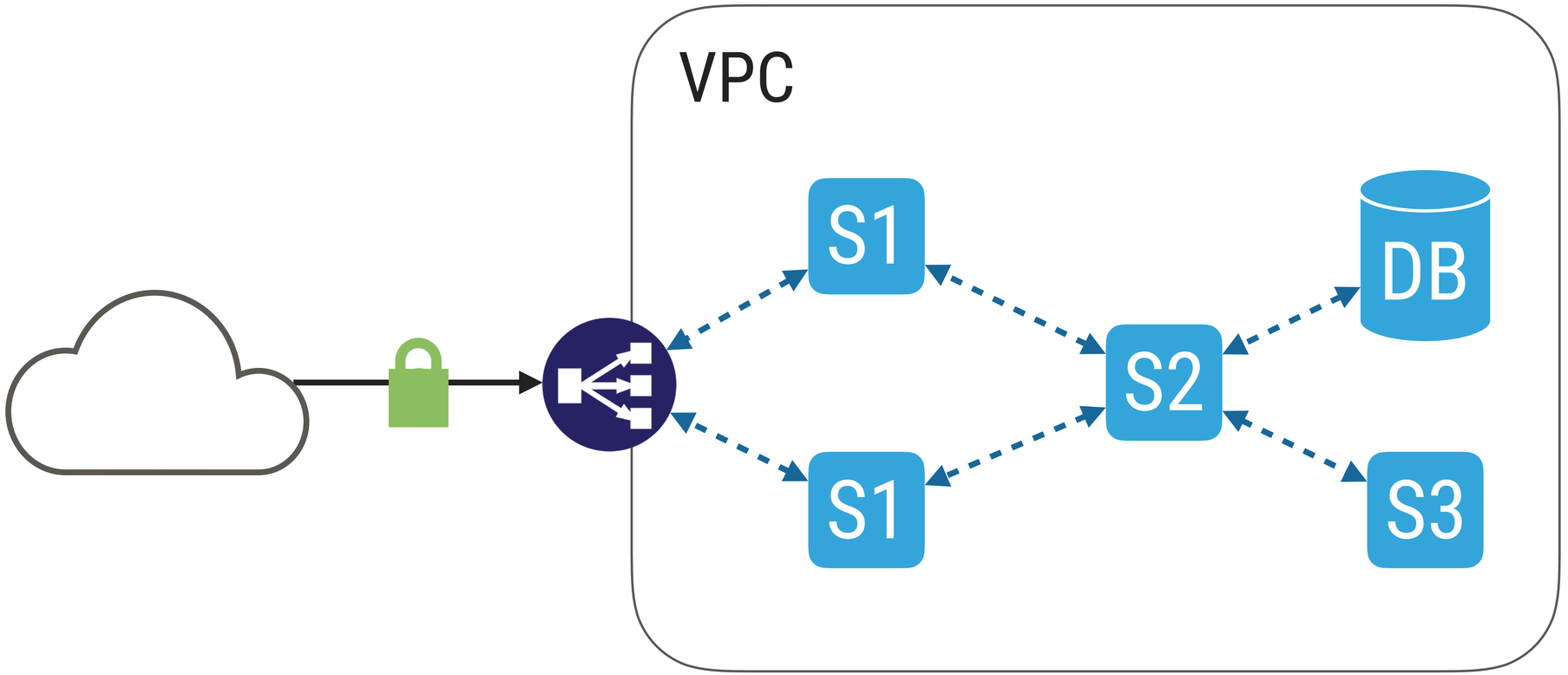
MINIMIZING THE WINDOW OF COMPROMISE

PRACTICAL MTLS

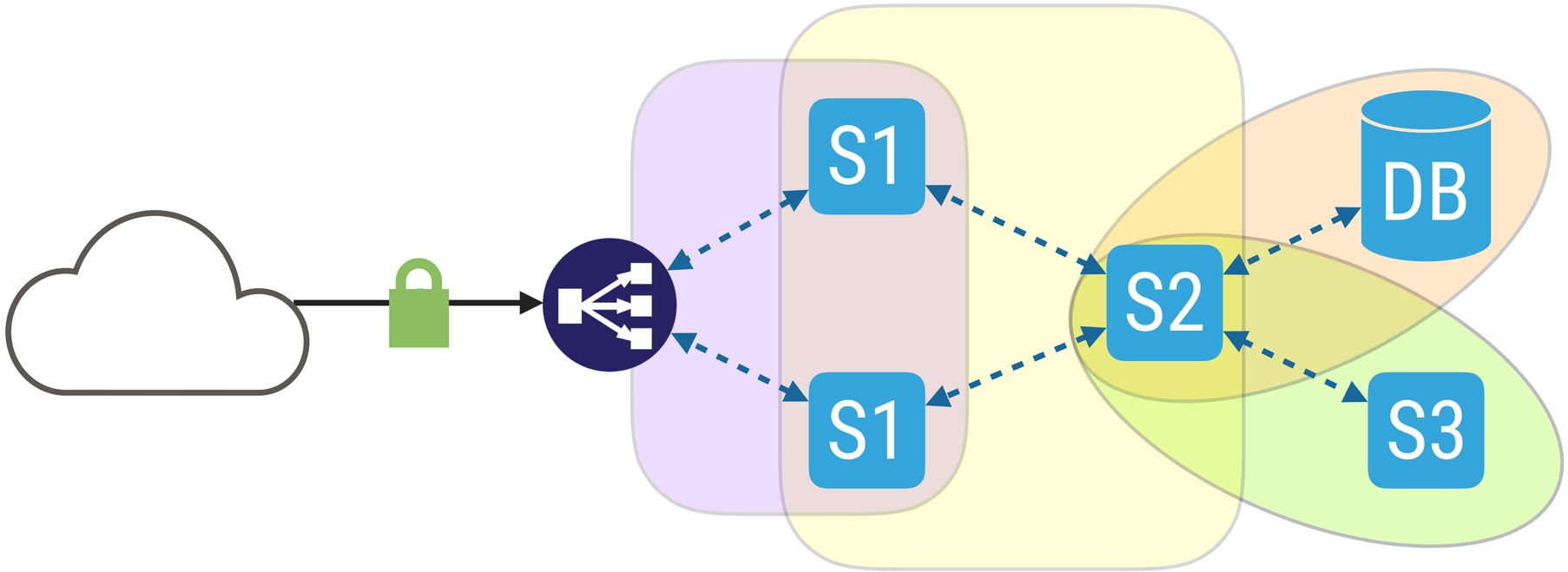


Ying Li
@cyli

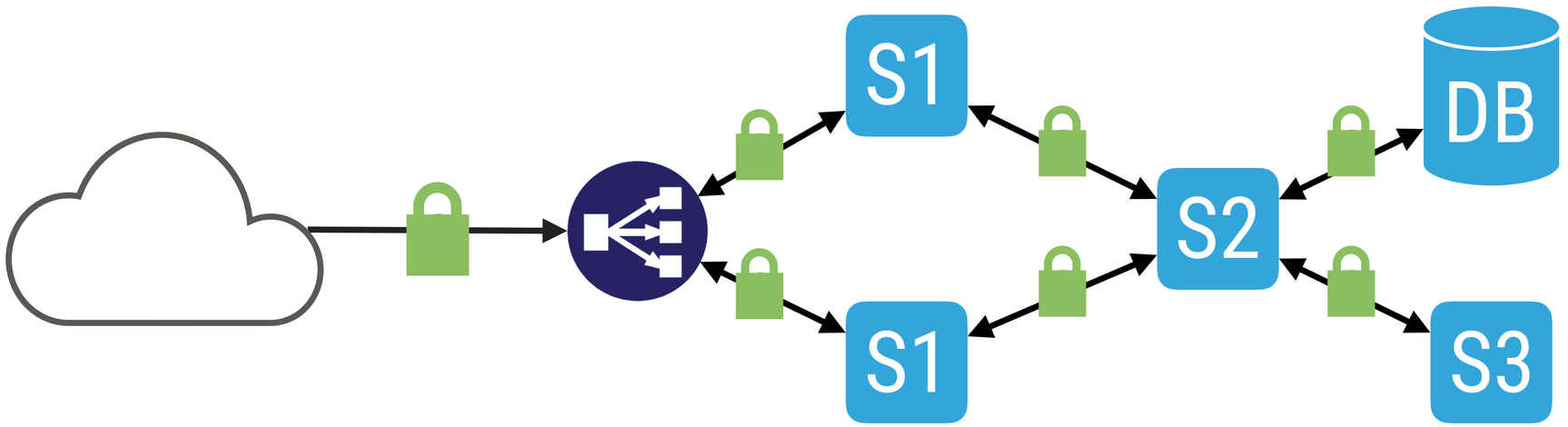
TYPICAL MICROSERVICE ARCHITECTURE



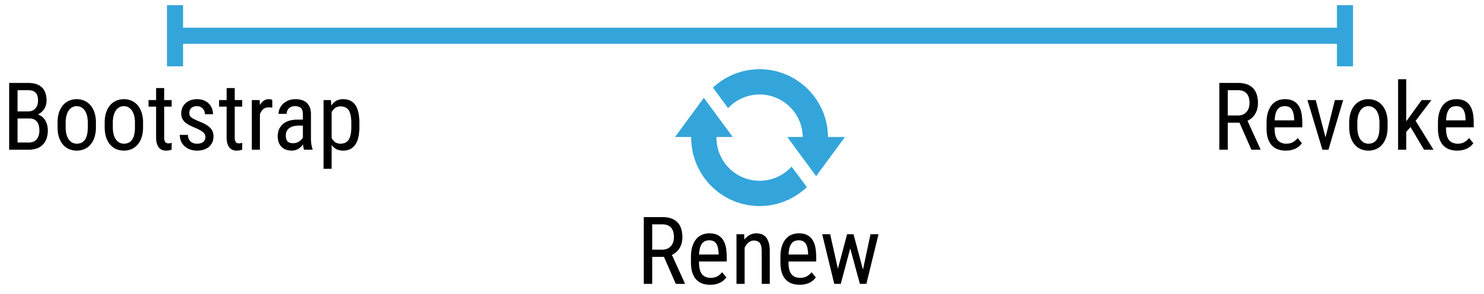
VLAN-TASTIC MICROSERVICE ARCHITECTURE



CORRECT MICROSERVICE ARCHITECTURE



APPLICATION TLS LIFECYCLE



BOOTSTRAP

- CSR ➡ CA
- Configuration

RENEW

- Schedule

RENEW

- Schedule
- CSR ➡ CA
- Configuration

RENEW

- Schedule
- CSR ➡ CA
- Configuration
- Restart

REVOKE

- CRL
- OCSP [Stapling]

AUTOMATE, AUTOMATE, AUTOMATE

- Promotes adoption of mTLS

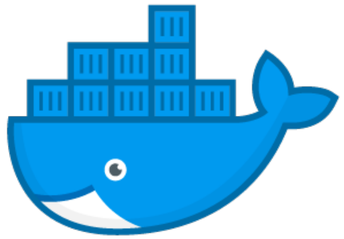
AUTOMATE, AUTOMATE, AUTOMATE

- Promotes adoption of mTLS
- Single location for private key

AUTOMATE, AUTOMATE, AUTOMATE

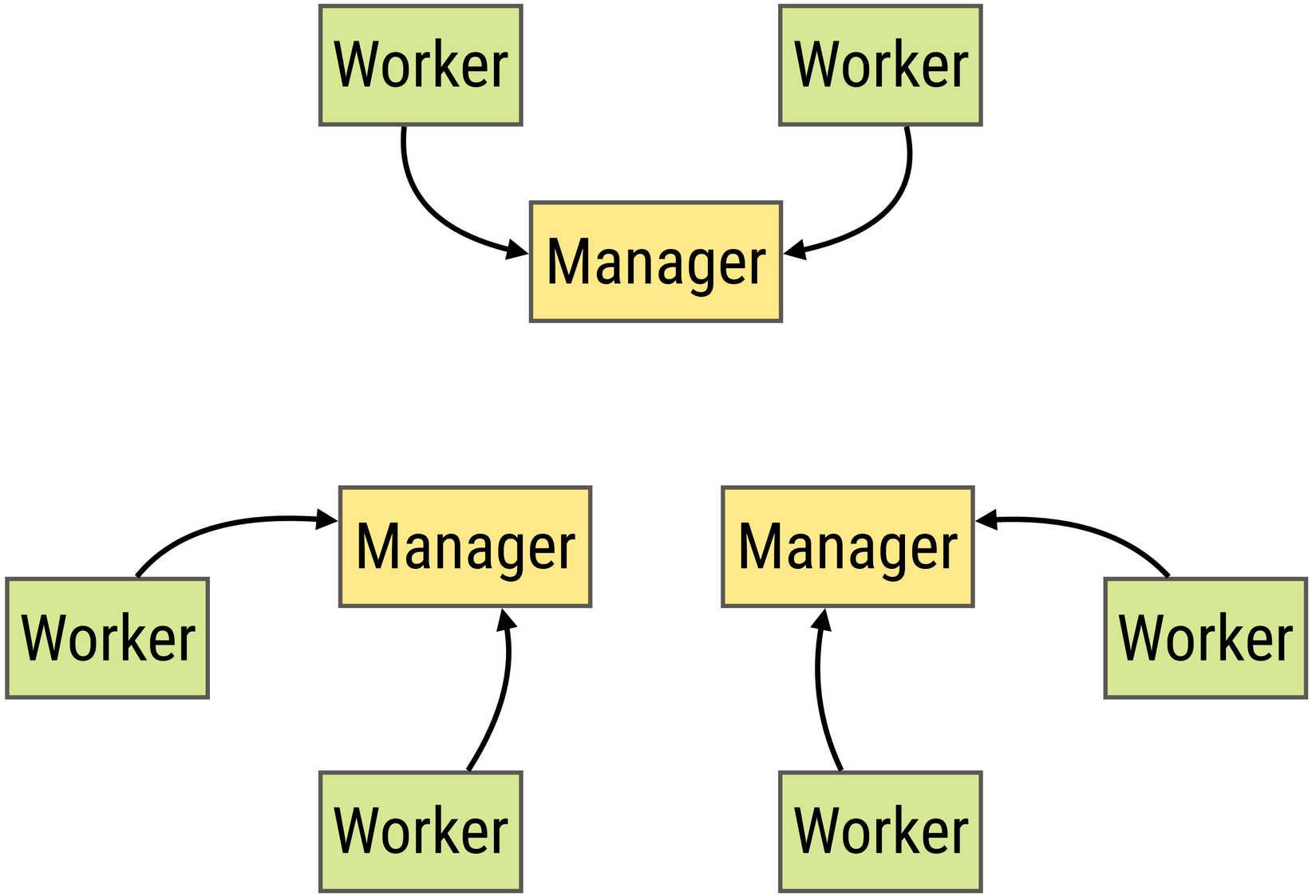
- Promotes adoption of mTLS
- Single location for private key
- Shorter certificate expiry

SWARMKIT OVERVIEW

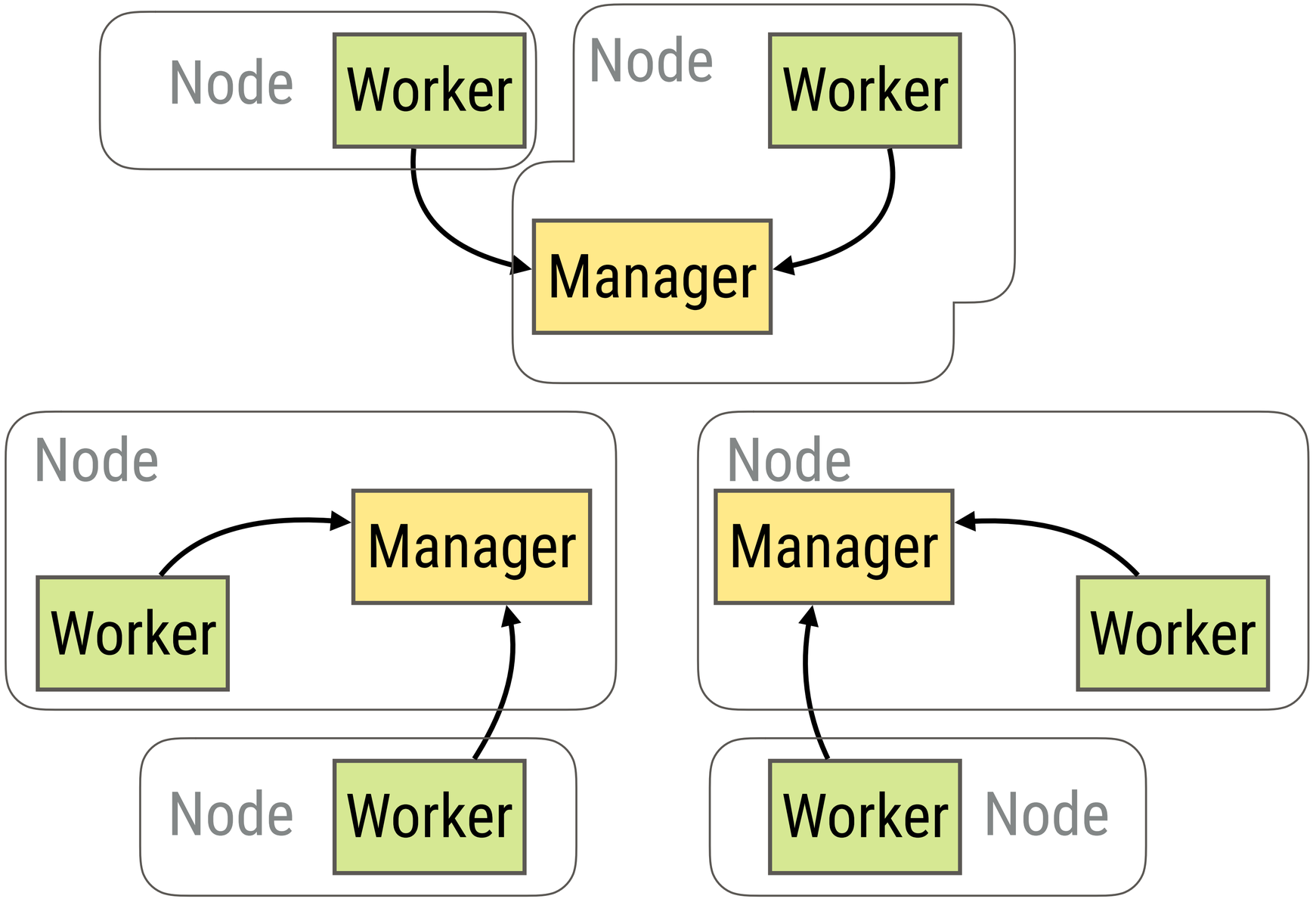


https://github.com/docker/swarmkit

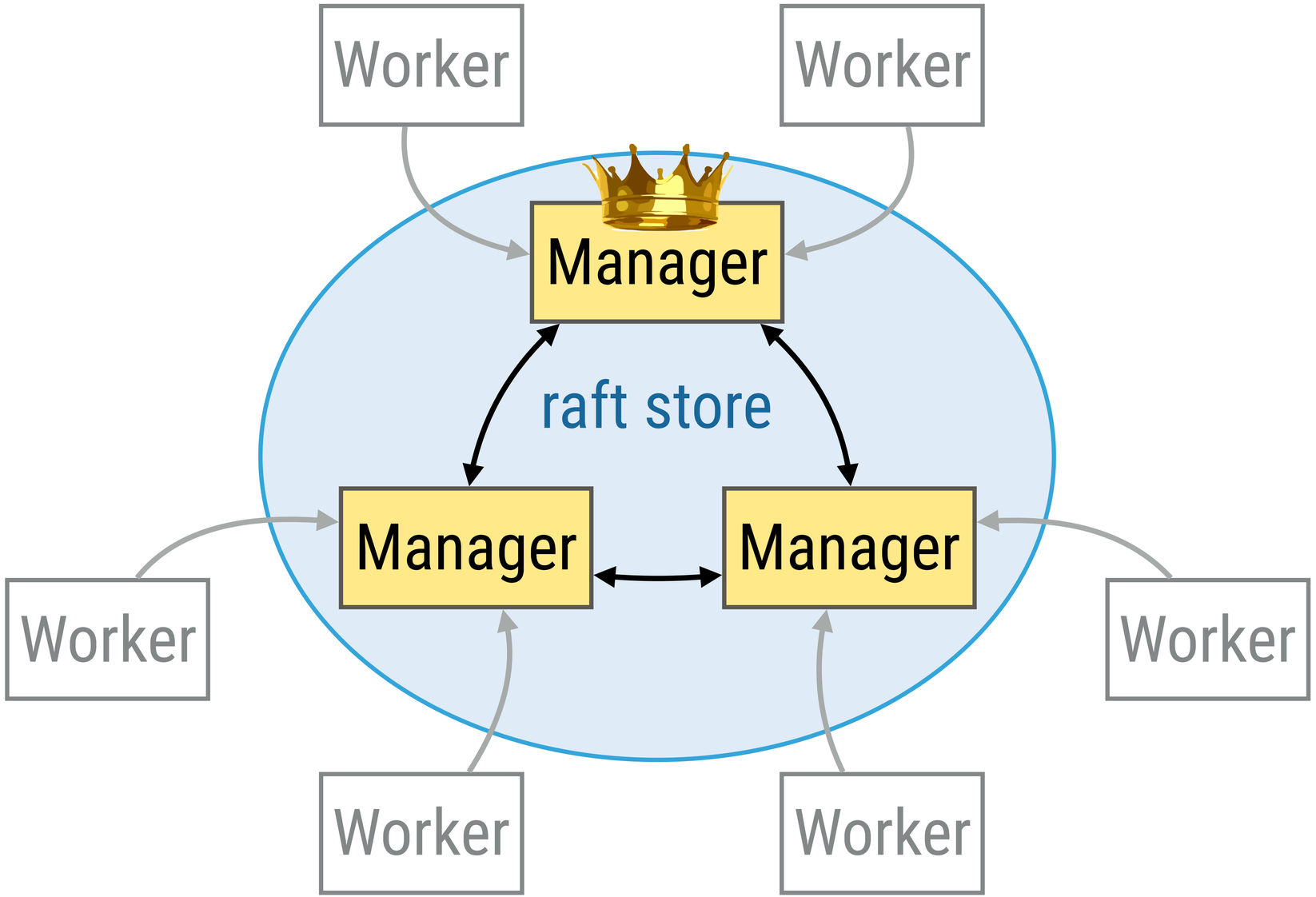
CLUSTER


Worker
Worker
Manager
Worker
Manager
Worker
Manager
Worker
Worker
Worker

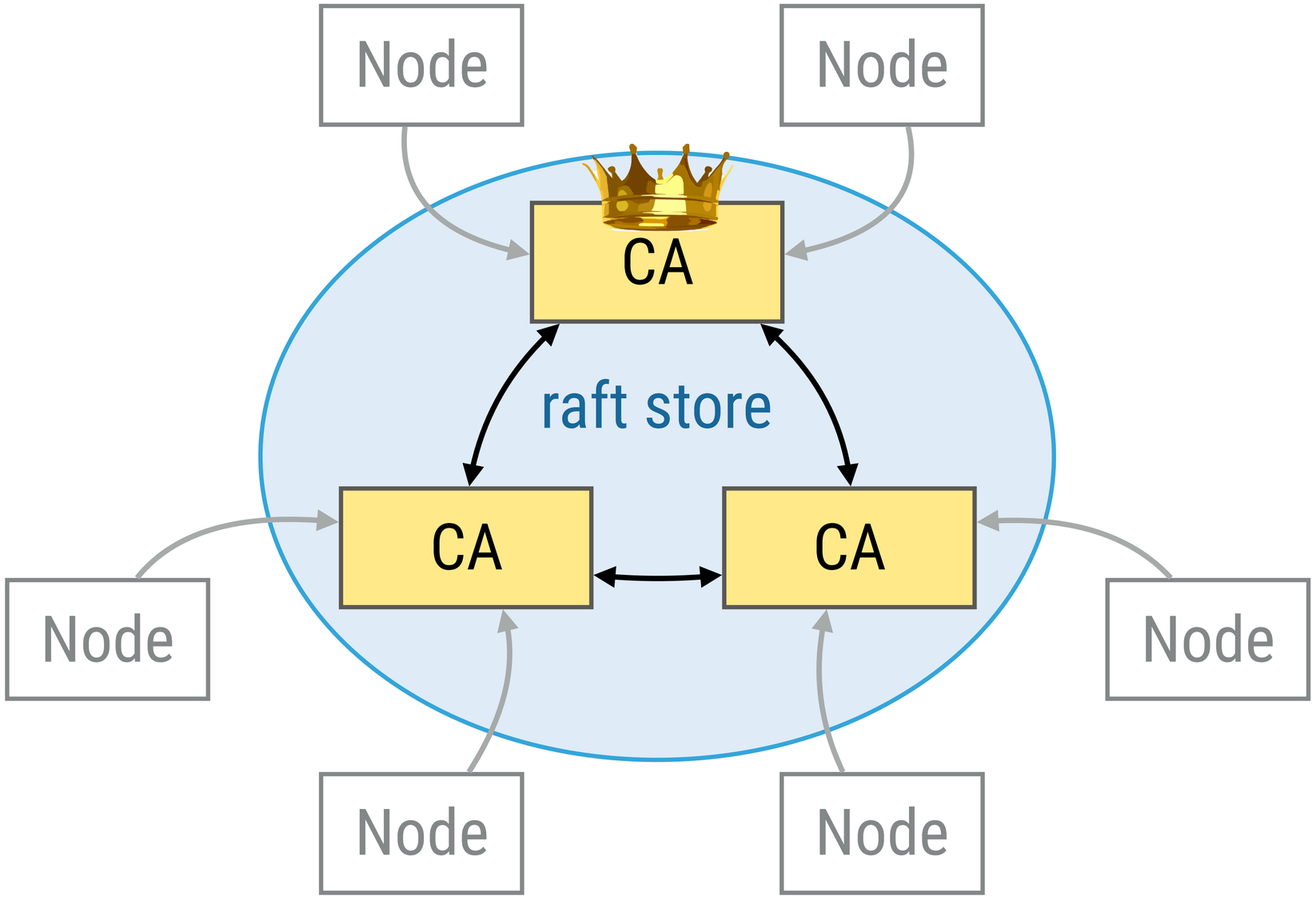
CLUSTER


Node
Worker
Node
Worker
Manager
Node
Manager
Worker
Node
Manager
Worker
Node
Worker
Node

CLUSTER


Worker
Worker
Manager
raft store
Manager
Manager
Worker
Worker
Worker
Worker

CLUSTER



SWARMKIT'S IMPLEMENTATION

```
$ openssl x509 -in
/var/lib/docker/swarm/certificates/swarm-node.crt
-te
Cert
...
  Issuer: CN=swarm-ca
   Validity
     Not Before: Mar  9 15:21:00 2017 GMT
     Not After : Jun  7 16:21:00 2017 GMT
   Subject: O=lgz5xj1eqg..., OU=swarm-manager, CN=ofcm6bdy...
   ...
     X509v3 Subject Alternative Name:
       DNS:swarm-manager, DNS:ofcm6bdy..., DNS:swarm-ca
     ...
-----BEGIN CERTIFICATE-----
MIICNDCCAdugAwIBAgIUCoRaj23j4h5
...
```

Swarm ID

Node Role

Node ID

BOOTSTRAP

Token Version

Random Secret

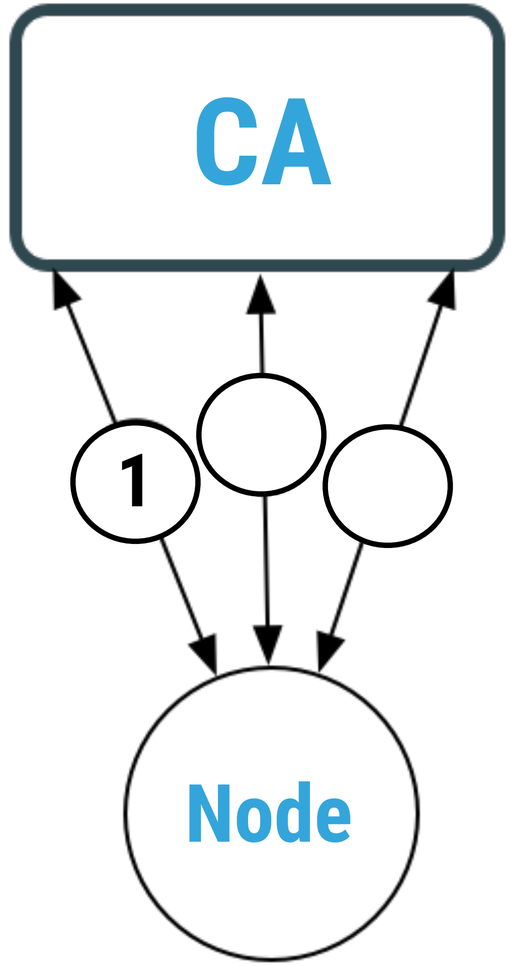
`SWMTKN-1-mx8suomaom825bet6-cm6zts22rl4hly2`

Known Prefix

Hash of Root CA

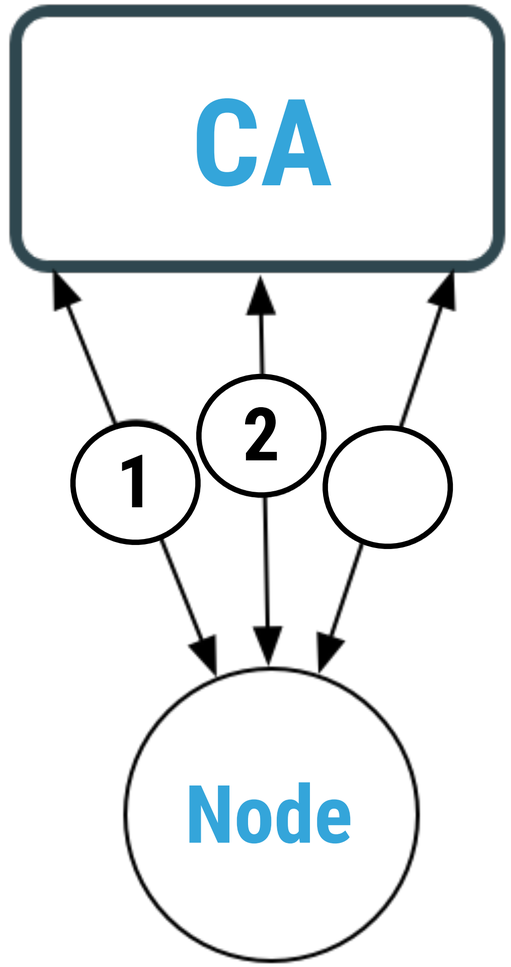
BOOTSTRAP

1. Retrieve, validate Root CA certificate.



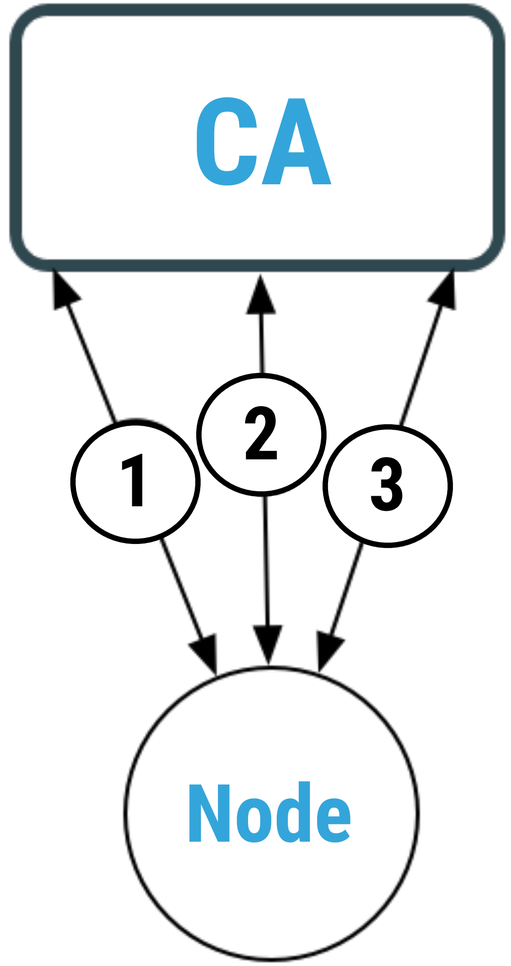
BOOTSTRAP

1. Retrieve, validate Root CA certificate.
2. CSR + secret token ➡ CA. (TLS)

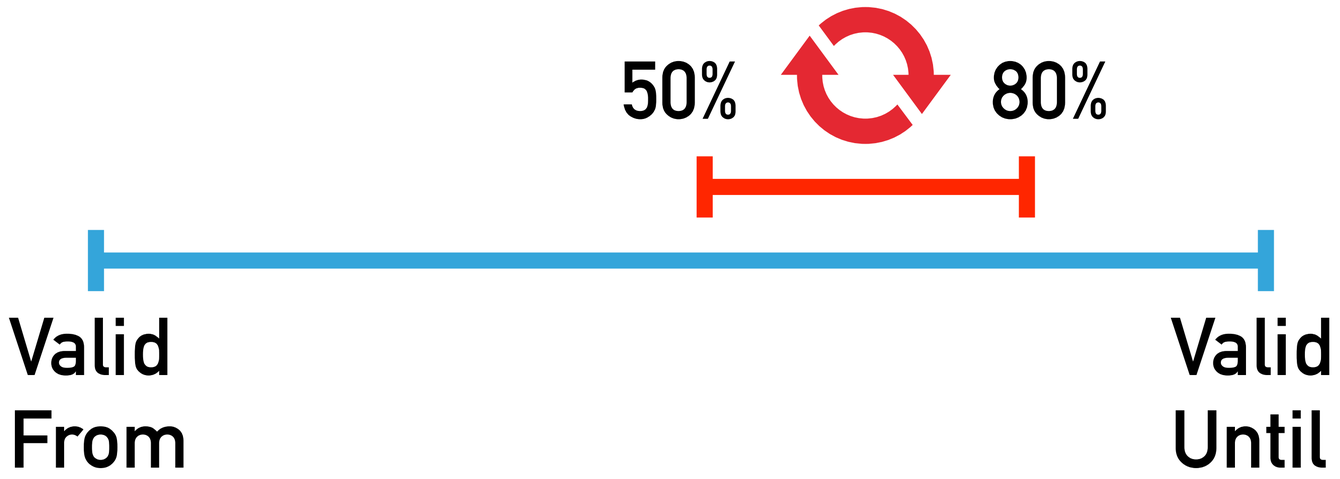


BOOTSTRAP

1. Retrieve and validate Root CA Public key material.
2. CSR + secret token ➡ CA. (TLS)
3. Get certificate. (TLS)



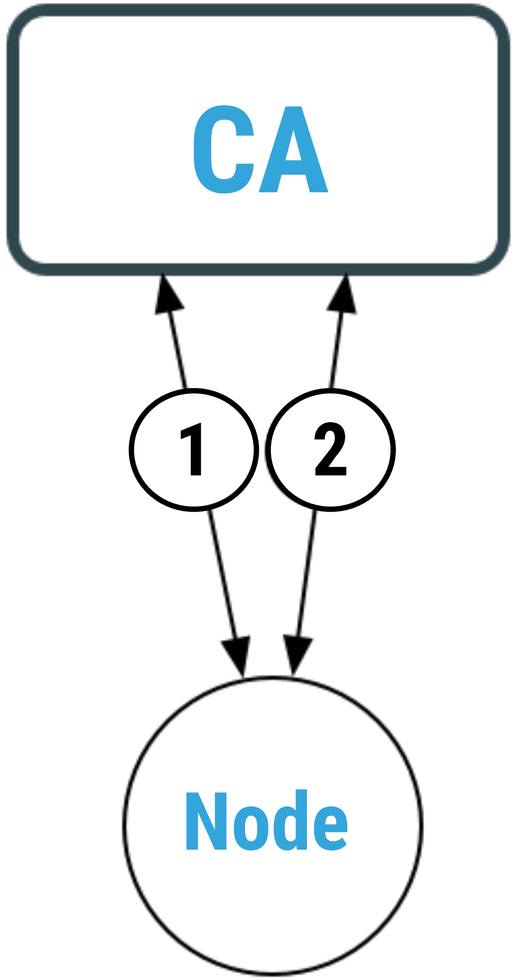
RENEW



RENEW

1. CSR + ➡ CA. (mTLS)

2.Get certificate. (mTLS)



RENEW

1. Trigger extra leader election
2. Workers all need to reconnect to managers
3. Reschedule work

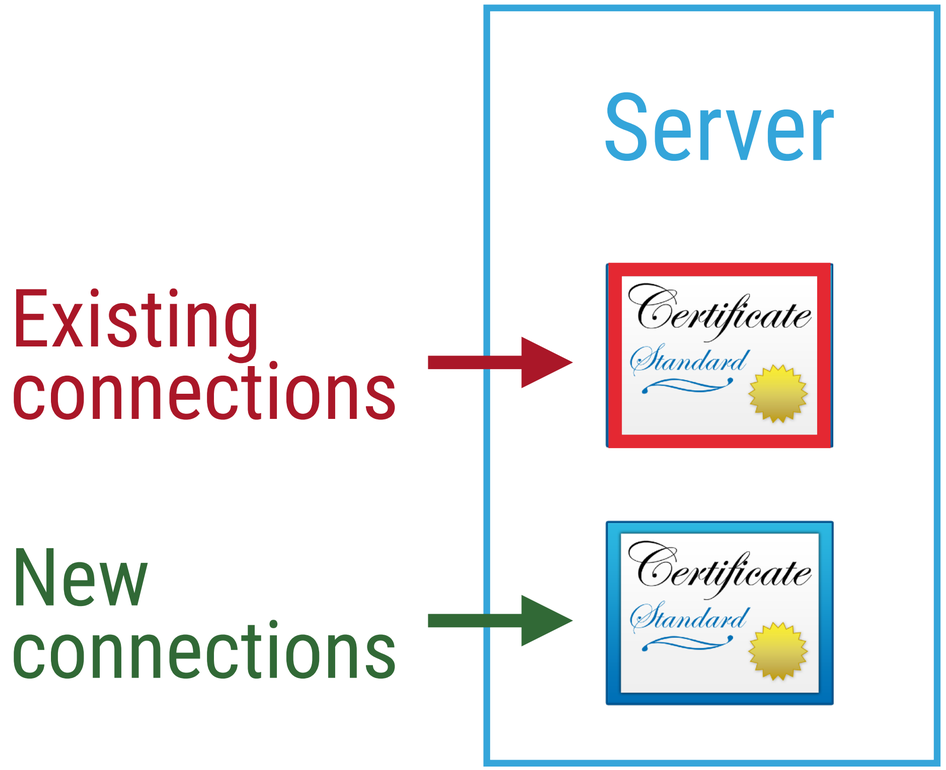
RENEW

```
type TransportCredentials interface {
    ClientHandshake(context.Context, string, net.Conn) (
        net.Conn, AuthInfo, error)
    ServerHandshake(net.Conn) (net.Conn, AuthInfo, error)
    Info() ProtocolInfo
    Clone() TransportCredentials
    OverrideServerName(string) error
}
```

RENEW

```
type TransportCredentials interface {
    ClientHandshake(context.Context, string, net.Conn) (
        net.Conn, AuthInfo, error)
    ServerHandshake(net.Conn) (net.Conn, AuthInfo, error)
    Info() ProtocolInfo
    Clone() TransportCredentials
    OverrideServerName(string) error
}
```

RENEW



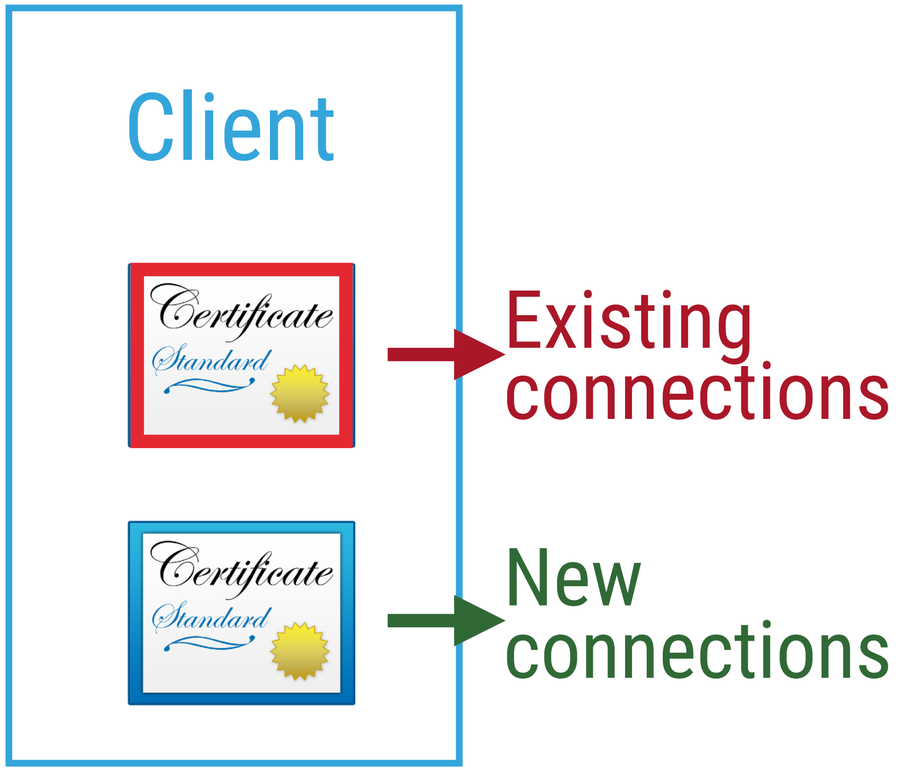

```
func (c *MutableTLSCreds) ServerHandshake(rawConn net.Conn) (
        net.Conn, credentials.AuthInfo, error) {
    c.Lock()
    conn := tls.Server(rawConn, c.config)
    c.Unlock()
    if err := conn.Handshake(); err != nil {
        rawConn.Close()
        return nil, nil, err
    }

    return conn, credentials.TLSInfo{
        State: conn.ConnectionState()}, nil
}
```

RENEW

```
func (c *MutableTLSCreds) ClientHandshake(ctx context.Context,
        addr string, rawConn net.Conn) (
        net.Conn, credentials.AuthInfo, error) {
    c.Lock()
    if c.config.ServerName == "" {
        colonPos := strings.LastIndex(addr, ":")
        if colonPos == -1 {
            colonPos = len(addr)
        }
        c.config.ServerName = addr[:colonPos]
    }

    conn := tls.Client(rawConn, c.config)
    c.Unlock()
    //....
    return conn, nil, nil
}
```



REVOKE

SWARMKIT'S IMPLEMENTATION

~~REVOKE~~

REMOVE

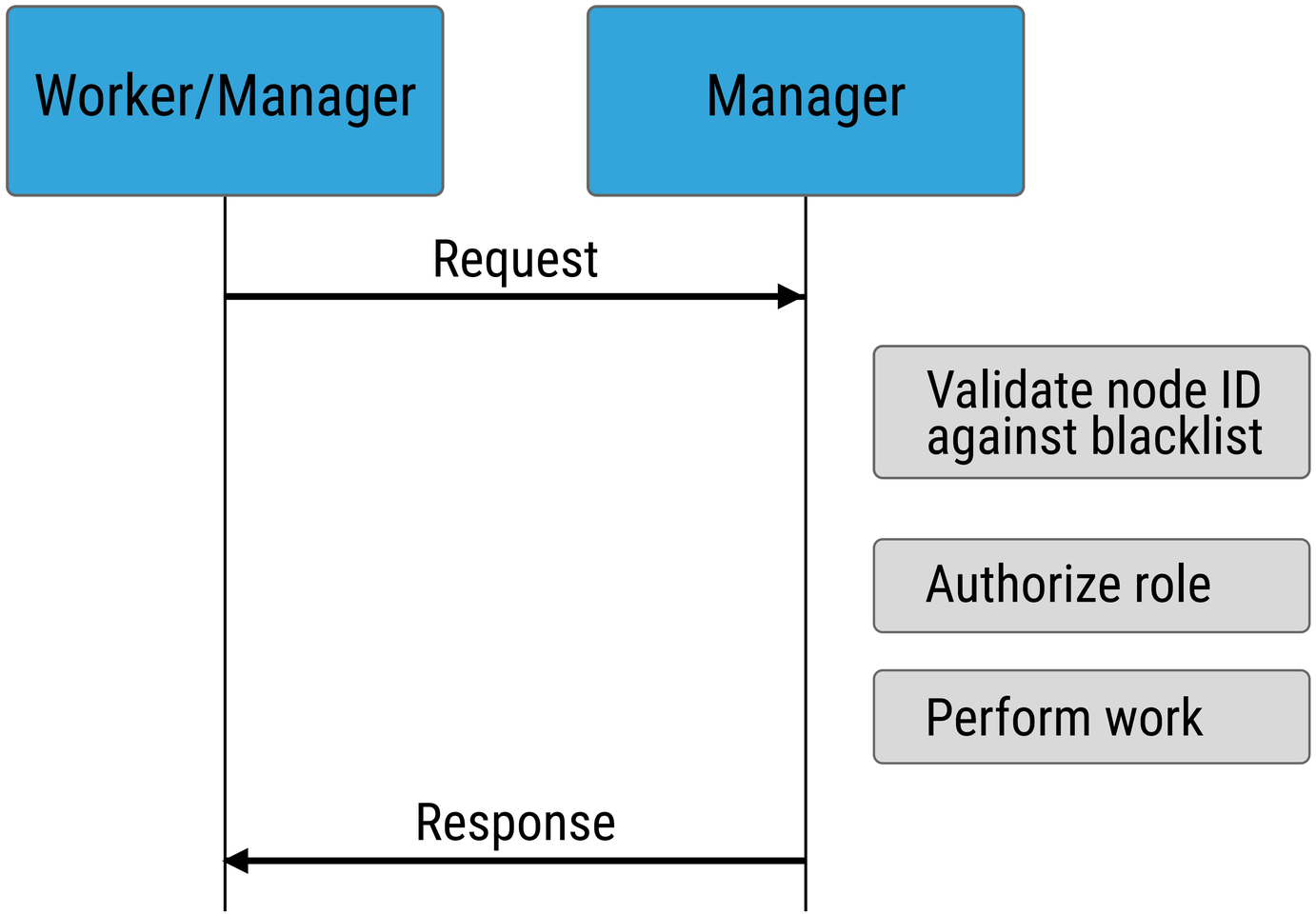
~~CRLS, OCSP [Stapling]~~

REMOVE

NODE BLACKLIST

Node ID	Certificate Expiry
a8h1vsk3k9o5nwea858ty9kma	2017-08-26 01:02:52 UTC
k80l2au3yq9f7x6r2oca13vwt	2017-07-15 11:35:23 UTC
n970d5be9ccgnreg4iti4jho3	2017-08-01 22:59:05 UTC

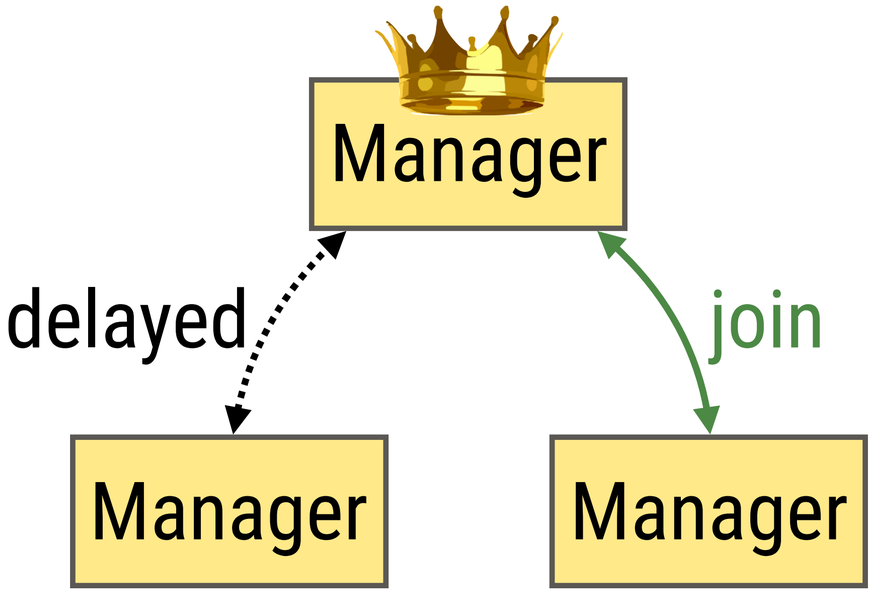
REMOVE


Worker/Manager
Manager
Request
Validate node ID against blacklist
Authorize role
Perform work
Response

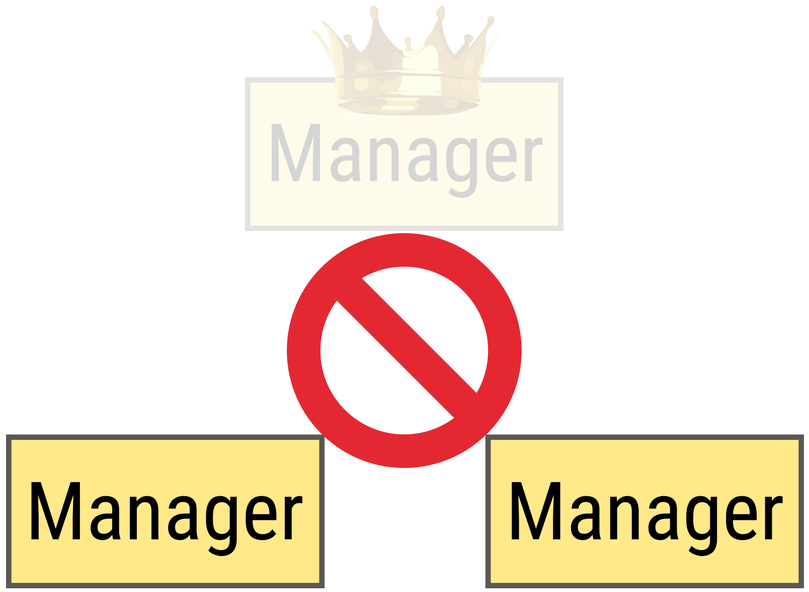
REMOVE

BLACKLIST VS WHITELIST

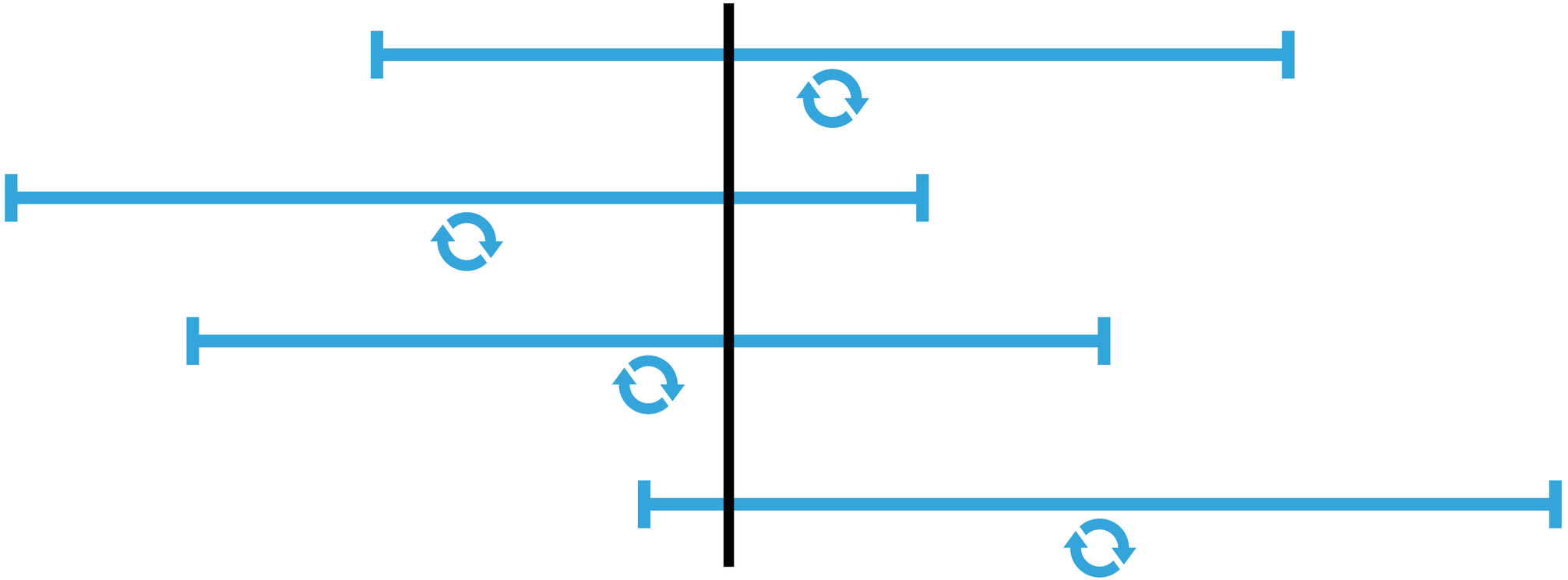
REMOVE



REMOVE



PROBLEM



Rotate CA

CA ROTATION

1
- (conf.) All nodes: trust old and new CA
- (wait.) Verify all nodes

CA ROTATION

1. (conf.) All nodes: trust old and new CA
2. (wait.) Verify all nodes
3. (conf.) All nodes: renew certificates
4. (wait.) Verify all nodes

CA ROTATION

1. (conf.) All nodes: trust old and new CA
2. (wait.) Verify all nodes
3. (conf.) All nodes: renew certificates
4. (wait.) Verify all nodes
5. (conf.) All nodes: trust new CA only
6. (wait.) Verify all nodes

CROSS-SIGNED INTERMEDIATE

Key Info: A
Signed by: A

Key Info: B
Signed by: B

Leaf cert: X
Signed by: B

Root: B ✓

CROSS-SIGNED INTERMEDIATE

Key Info: A
Signed by: A

Key Info: B
Signed by: B

Leaf cert: X
Signed by: B

Root: A ✗

CROSS-SIGNED INTERMEDIATE

Key Info: B
DN: B
Signed by: A

Key Info: B
DN: B
Signed by: B

Key Info: A
DN: A
Signed by: A

CROSS-SIGNED INTERMEDIATE

Leaf cert: X
Signed by: B’

Root: A ✓

CROSS-SIGNED INTERMEDIATE

Leaf cert: X
Signed by: B

Root: B ✓

CA ROTATION

- ~~(conf.) All nodes: trust old and new CA~~
- ~~(wait.) Verify all nodes~~
- Generate cross-signed intermediate

CA ROTATION

- Generate cross-signed intermediate
- (conf.) All nodes: renew certificates
- (wait.) Verify all nodes

1

CA ROTATION

- Generate cross-signed intermediate

1
- (conf.) All nodes: renew certificates
- (wait.) Verify all nodes

2
- (conf.) All nodes: trust new CA
- (wait.) Verify all nodes

- Throw away cross-signed intermediate

CA ROTATION: BEFORE ROTATION

Node Trust Root:

Node TLS Certificate:

Cluster Trust Root:

Cluster Cert Issuer:

CA ROTATION: START ROTATION

Node Trust Root:

Node TLS Certificate:

Cluster Trust Root:

Cluster Cert Issuer:

CA ROTATION: NODE CERT RENEWAL

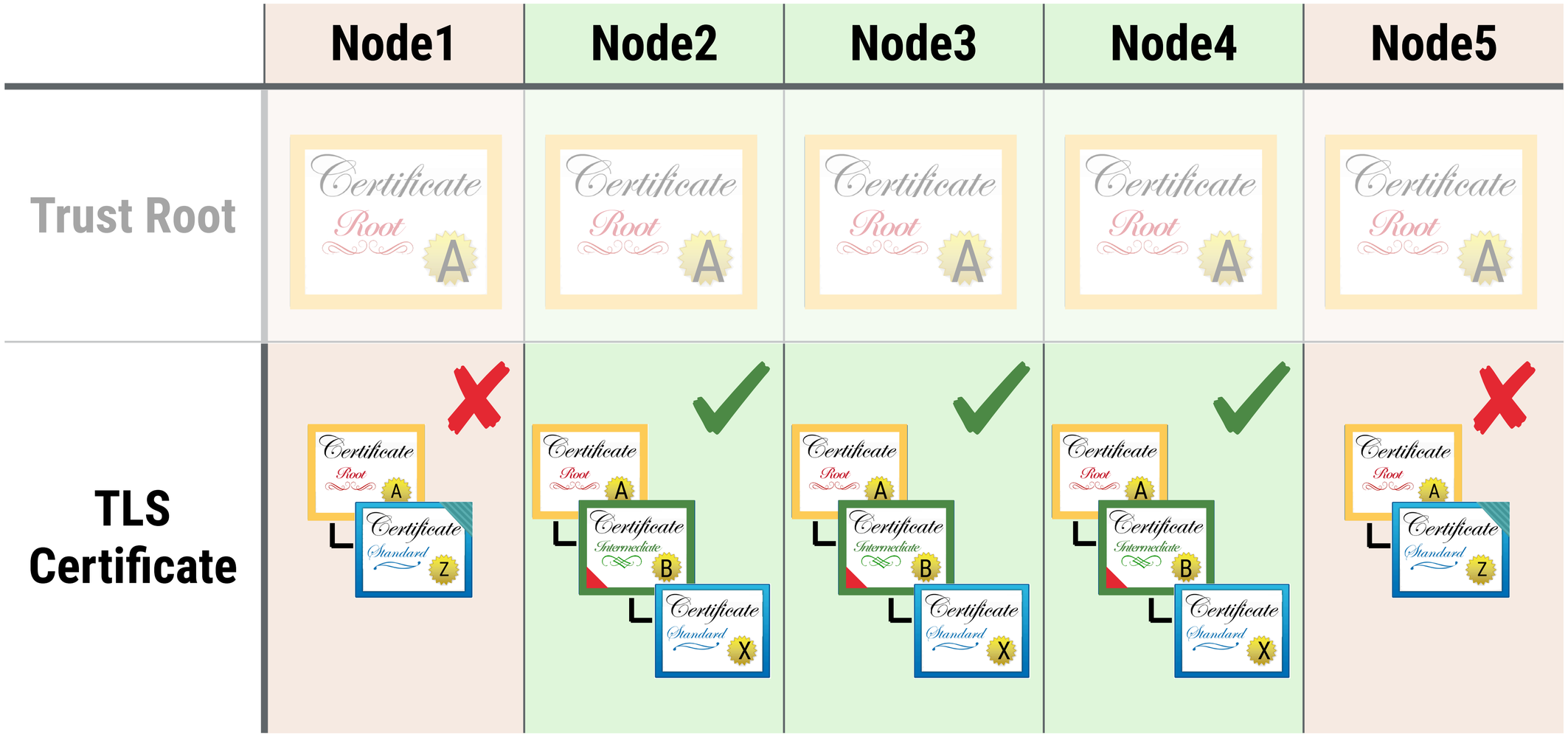
Node Trust Root:

Node TLS Certificate:

Cluster Trust Root:

Cluster Cert Issuer:

CA ROTATION: NODE CERT RENEWAL



CA ROTATION: ROTATE TRUST ROOT

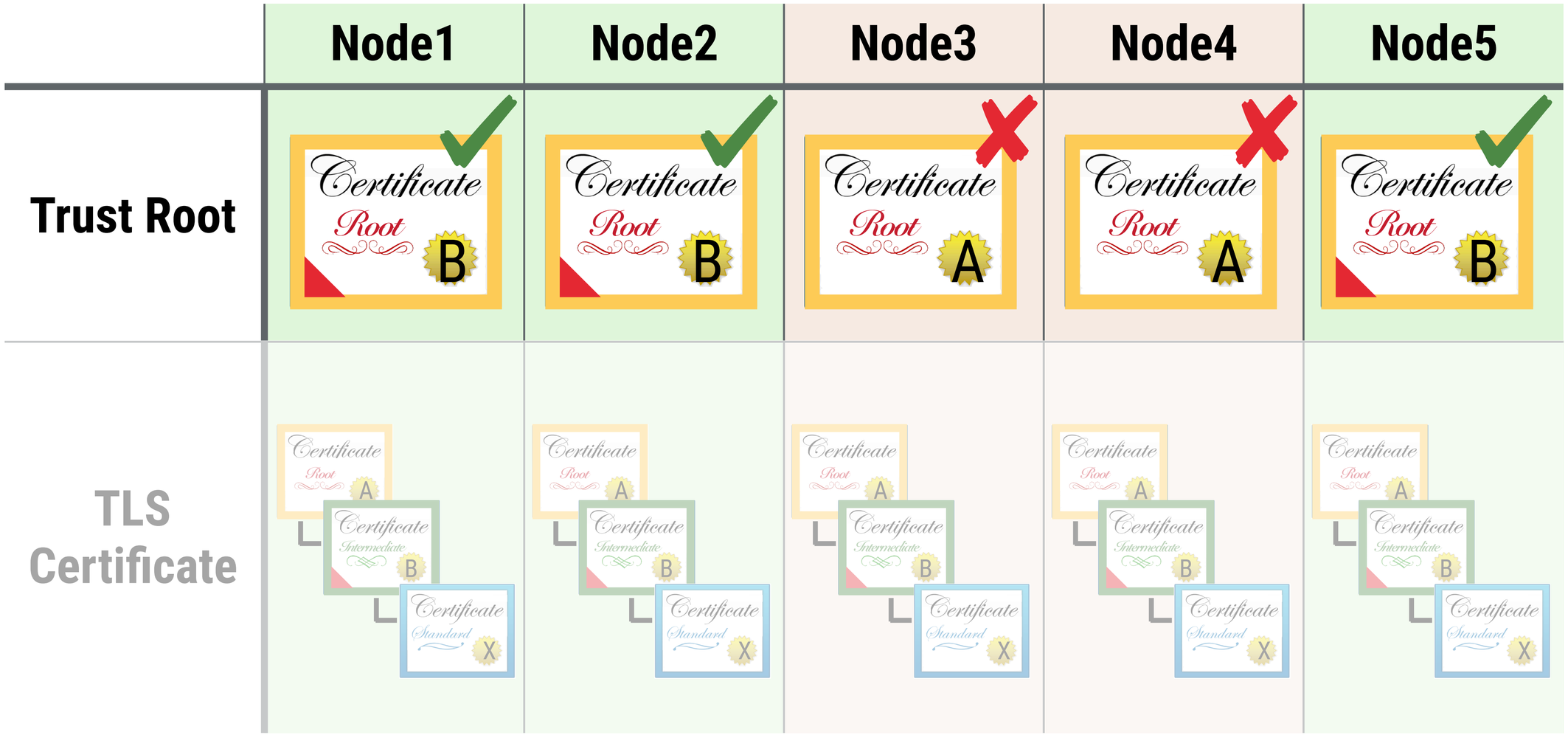
Node Trust Root:

Node TLS Certificate:

Cluster Trust Root:

Cluster Cert Issuer:

CA ROTATION: ROTATE TRUST ROOT



CA ROTATION: FINISH ROOT ROTATION

Node Trust Root:

Node TLS Certificate:

Cluster Trust Root:

Cluster Cert Issuer:

DEMO

MINIMIZING THE WINDOW OF COMPROMISE

MINIMIZING THE WINDOW OF COMPROMISE

- automatic bootstrap, renewal
- short certificate expiry

MINIMIZING THE WINDOW OF COMPROMISE

- automatic bootstrap, renewal
- short certificate expiry
- certificate revocation
- CA rotation

MORE INFORMATION

https://github.com/docker/swarmkit

https://diogomonica.com/2017/01/11/hitless-tls-certificate-rotation-in-go/

https://github.com/cloudflare/cfssl

(@cyli)