

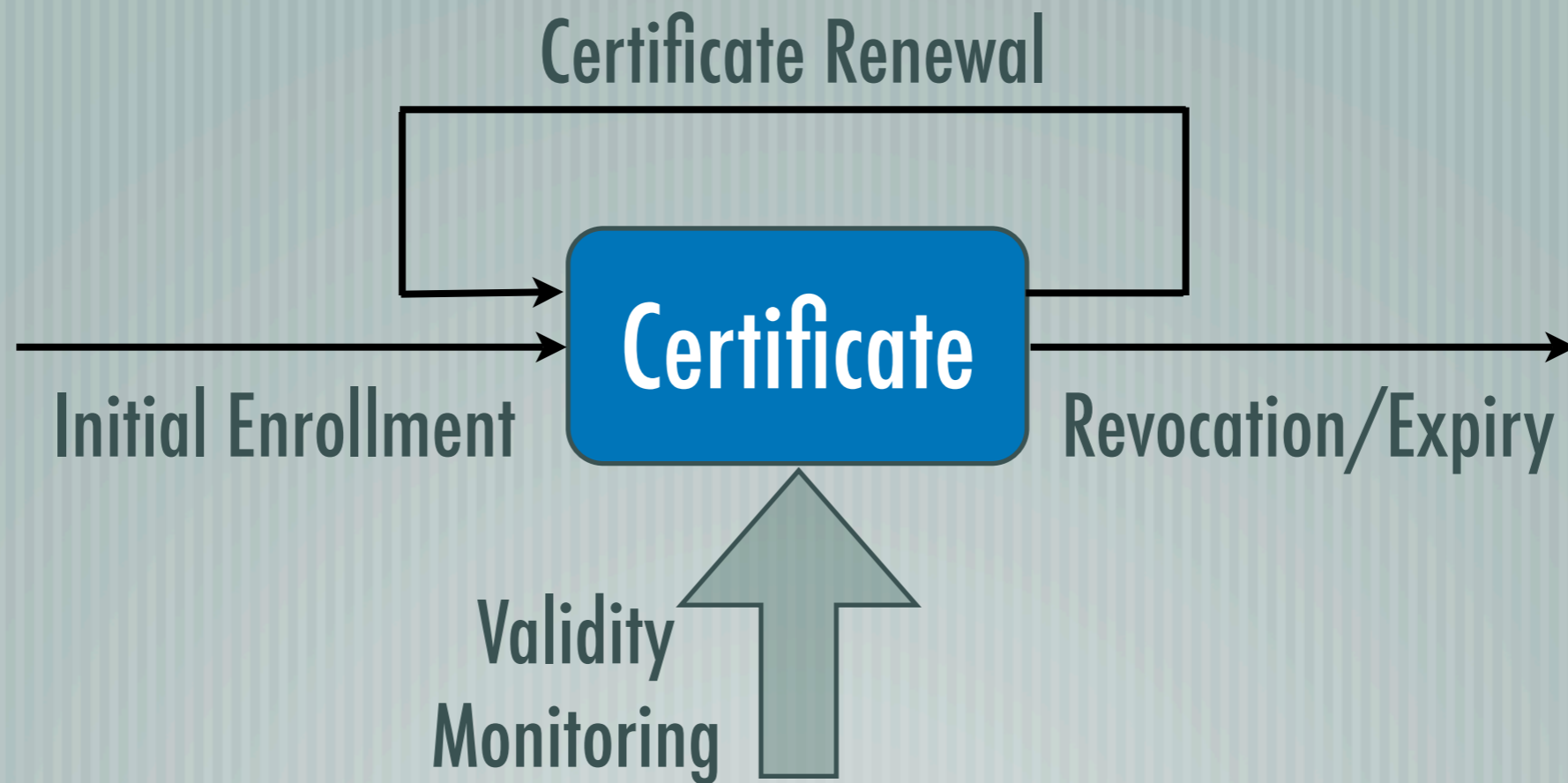


# Advanced SCEP Use

Certificate Lifecycle Management with SCEP

2005-10-18 – Martin Bartosch

# Certificate Lifecycle



# Real-life Certificate Use

- [ PKI enabled applications require (valid) certificates

- [ Obvious approach: manual administration

- Request certificate

- Install certificate

- Note expiry date in calendar and...

- ...don't forget to request new certificate prior to expiry

- Repeat. Don't make mistakes.

# Real-life Certificate Use (2)

- [ Manual certificate administration

- does not scale

- is error prone

- is a management and support nightmare

- [ Enter automatic certificate management

# Certificate Management Protocols

- [ PKIX-CMP (Certificate Management Protocol)

- RFC 2510, RFC 2511; not widely used

- [ CMC (CM over CMS, no transport defined)

- RFC 2797, used by Microsoft CA (with COM/DCOM as proprietary transport protocol)

- [ SCEP (IETF Draft version 12, PKCS#10 in PKCS#7 over HTTP)

- de-facto standard for network appliances (Cisco etc.)

- **supported by OpenCA**

# OpenCA SCEP Support

— [ Certificate Lifecycle Management possible via SCEP:

— Initial enrollment

— Renewal

# Initial Enrollment

— [ Anonymous enrollment (no authentication)

— **supported** by OpenCA (as of today)

— [ Pre-authentication (preshared key)

— **not supported**

— [ Signature with existing certificate issued by CA

— not explicitly defined by SCEP draft (as of version 12)

— currently being tested (**to be supported soon**)

# Certificate Renewal

- [ Authenticate new request by signing with existing certificate

- Signature and signer certificate must be valid **and**

- Signer DN == requested DN **and**

- Not more than two valid certs with the same DN

- [ Automatic approval possible

- [ Renewal requests inherit original RA and Role

# SCEP Server Configuration

## Policy definition

Allow Enrollment

Allow Renewal

Automatic Renewal  
Approval

etc/servers/scep.conf:

```
# ScepAllowEnrollment: if set to "NO"  
# the SCEP server will not accept requests  
# for certificate DNSs that don't exist yet.  
ScepAllowEnrollment      "YES"
```

```
# ScepAllowRenewal: if set to "YES" the SCEP  
# server will allow renewal requests for  
# existing certificates.  
ScepAllowRenewal        "YES"
```

```
# ScepAutoApprove: if set to "YES" and  
# SCEP request is signed with already existing  
# end entity certificate the request is  
# automatically approved in the RA.  
ScepAutoApprove        "NO"
```

# SCEP Server Configuration

Request processing

Accept request extensions

Default request settings

etc/servers/scep.conf:

```
# ScepKeepSubjectAltName: parse incoming  
# request and keep supplied SubjectAltName  
ScepKeepSubjectAltName "YES"
```

```
# Defaults for initial enrollment  
# Change these according to your setup  
ScepDefaultRole "VPN Server"  
ScepDefaultRA "Trustcenter itself"
```

# SCEP Server Configuration

Request processing

Matching renewal requests with existing certs

Request selection of Certificate Role (via PKCS#10 attribute, initial enrollment, currently testing)

etc/servers/scep.conf:

```
# ScepRenewRDNMatch: List of request RDNs that
# must match an existing certificate to
# identify the request as a renewal
# Example: "CN,O,C"
# Note: CN might not be enough for your
# case if your CNs are not unique. In
# this case add additional RDN components,
# such as OU, O or DC in order to allow
# a match.
```

```
ScepRenewalRDNMatch "CN"
```

# Certificate Monitoring

- [ CA based monitoring is not useful

- CA cannot easily keep track of responsible persons

- [ Client based approach

- on each client deploy some “agent” software

- invoked daily (cron job)

- monitor all local keystores

- check remaining certificate validity

- automatically enroll renewal requests

# SCEP Clients

— [ **sscep**: Unix SCEP client written in C

— handles raw SCEP communication

— no workflow handling

— <http://www.klake.org/~jt/sscep/>

— [ **autoscep**: Unix SCEP renewal client written in C

— based on sscep

— limited keystore support (PEM format only)

— <http://autosscep.spe.net/>

— [ **scepclient**: Java SCEP client

— <http://www.urut.ch/scep/>

# SCEP Clients: CertMonitor

- [ **CertMonitor**: Certificate monitoring agent (Perl)

- GPL'ed code

- handles renewal workflow

- completely automatic operation (cron)

- requires and encapsulates OpenSSL and scep

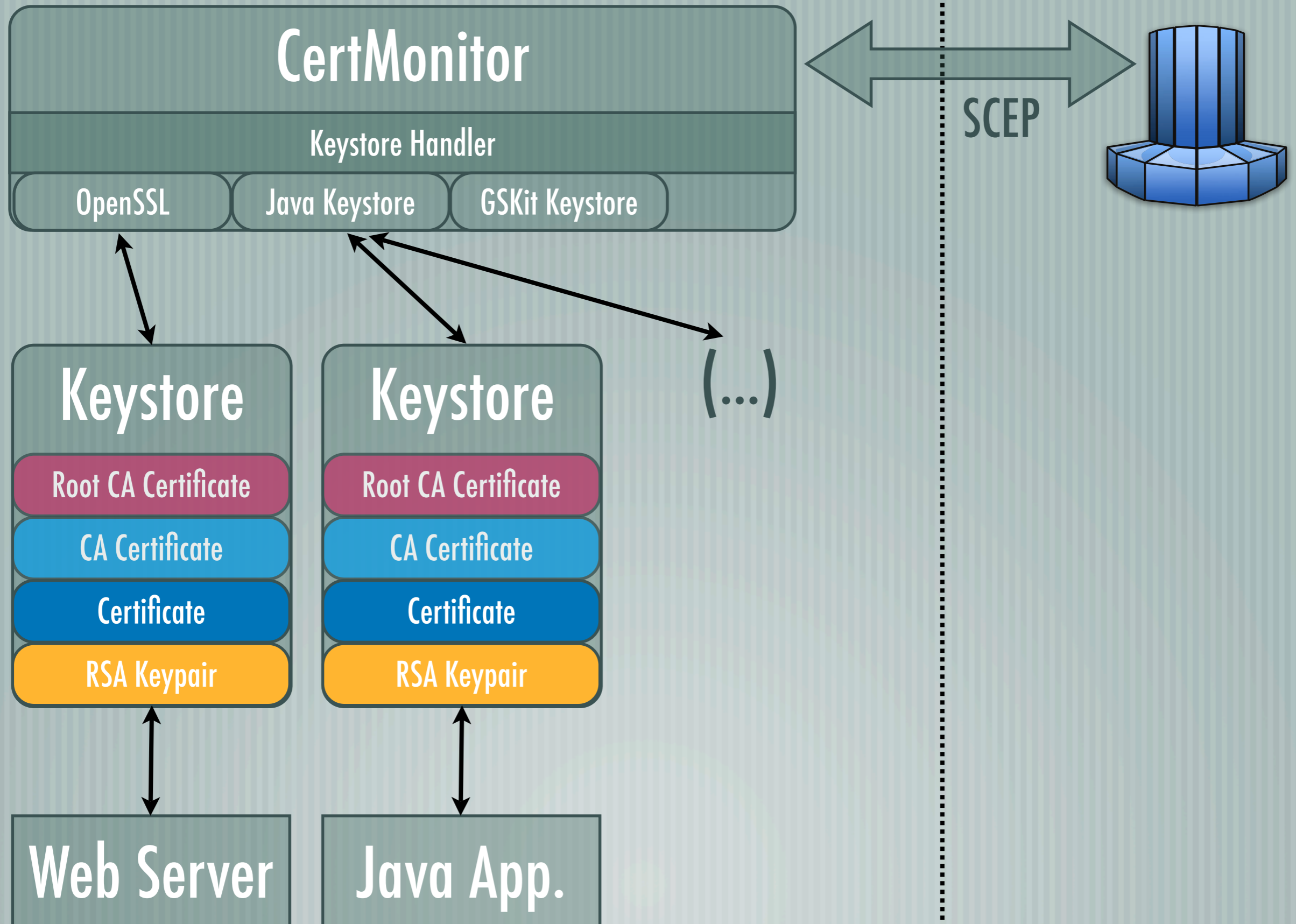
- can manage an arbitrary number of local keystores

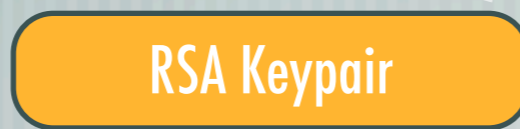
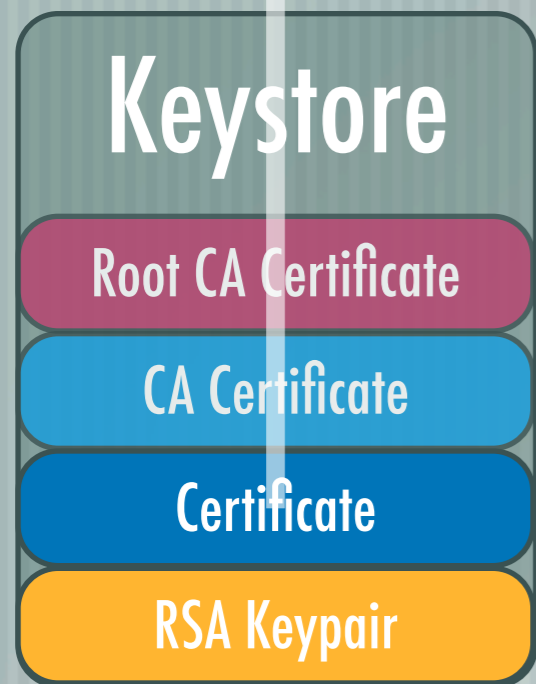
- multi-platform support

- multi-keystore-type support

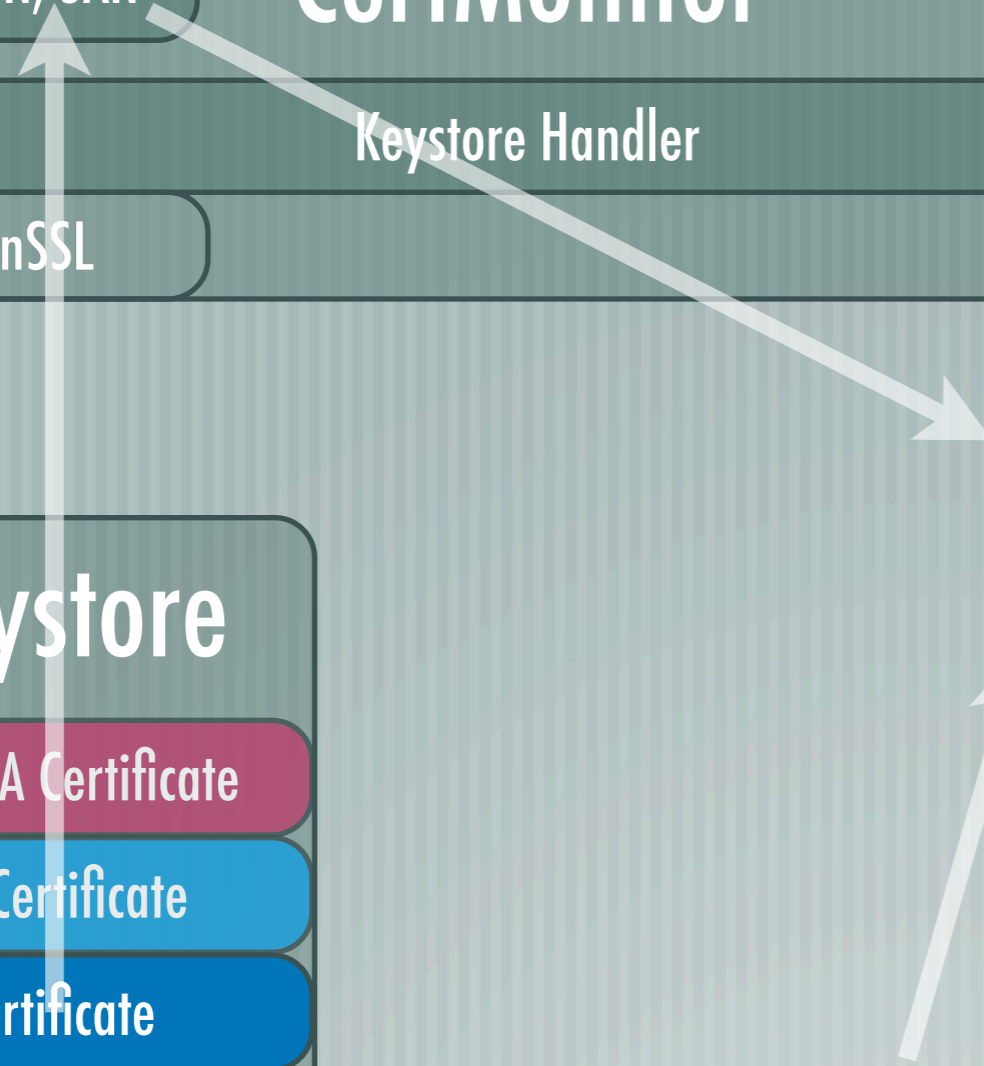
# End Entity System

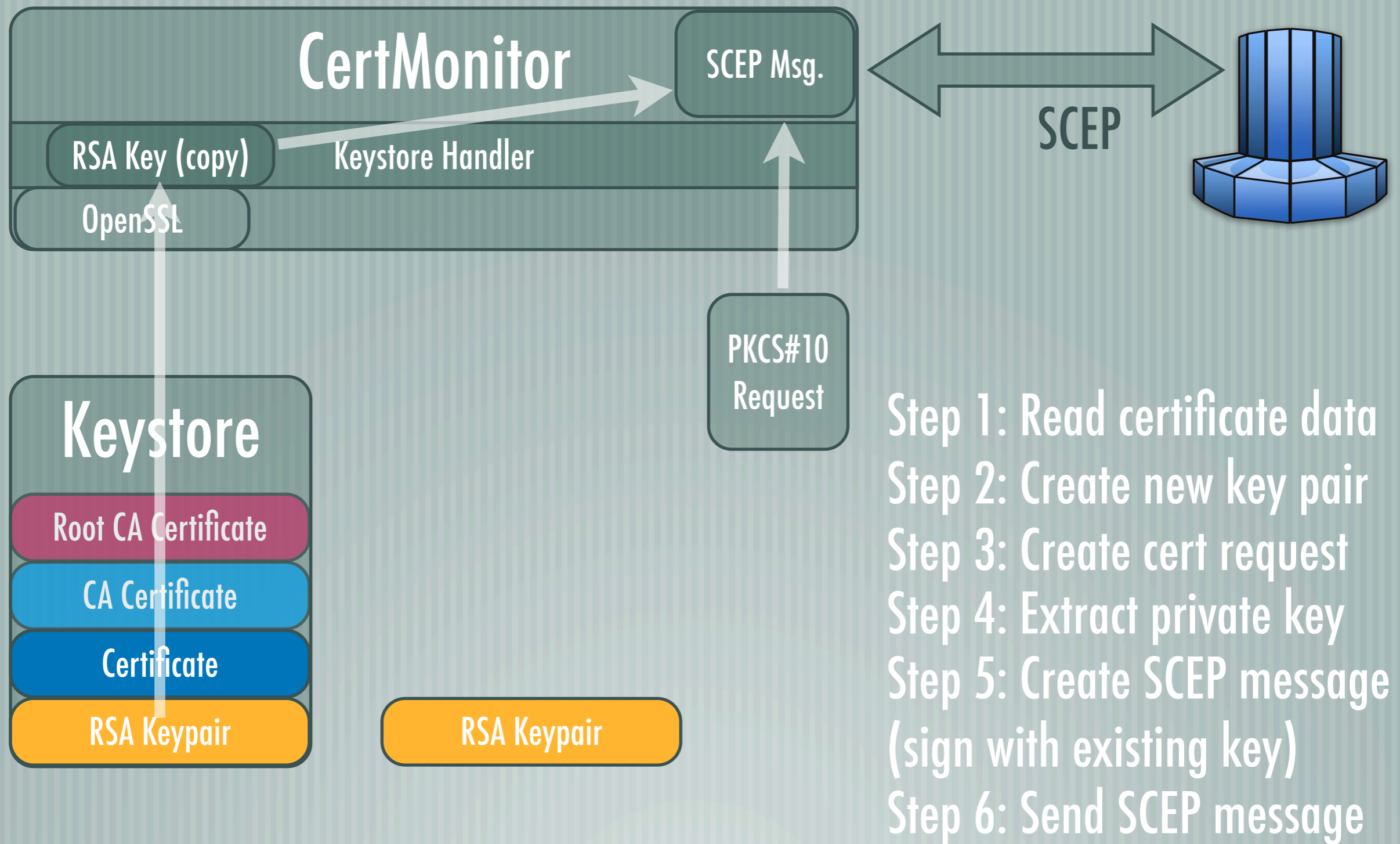
# SCEP Server

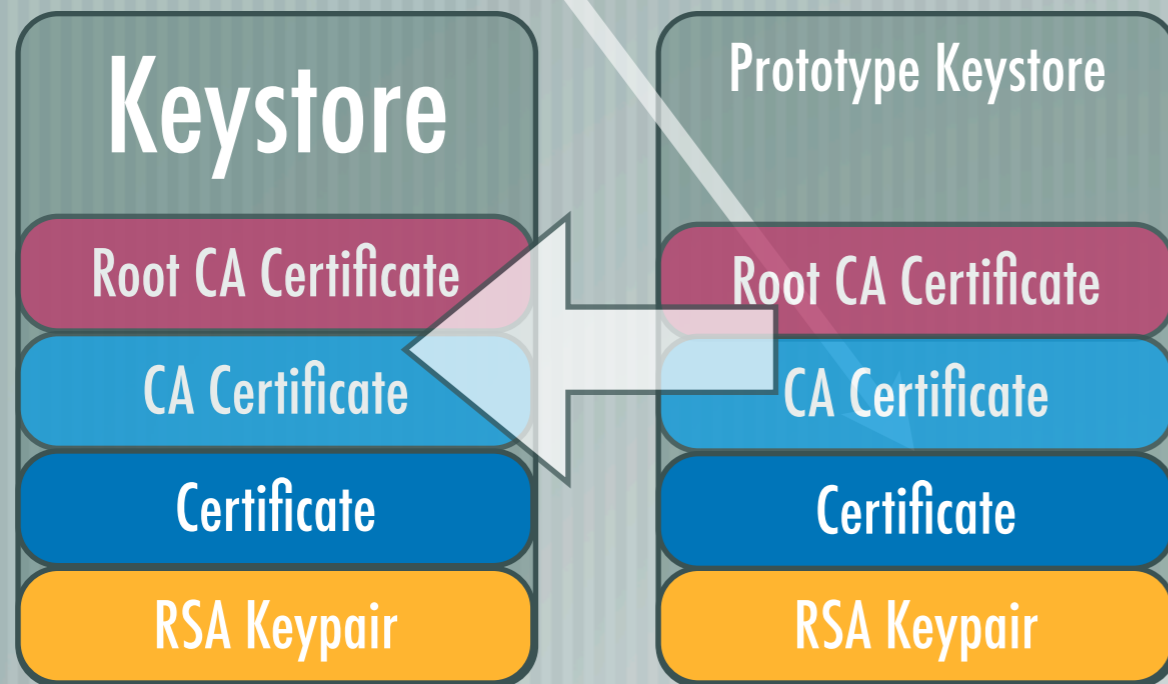




- Step 1: Read certificate data
- Step 2: Create new key pair
- Step 3: Create cert request







- Step 7: Poll SCEP Server
- Step 8: Receive certificate
- Step 9: Create new keystore
- Step 10: Replace old keystore

# CertMonitor

- [ Platforms supported: Unix (Linux, AIX, Solaris, Mac OS X...)

- Support planned for: Windows, z/OS, Tandem

- [ Multi-Keystore support:

- OpenSSL format (PEM encoded certs/keys)

- IBM GSKit Keystore format (MQ Series)

- Java Keystore (planned)

- RACF (planned: access from USS via REXX)

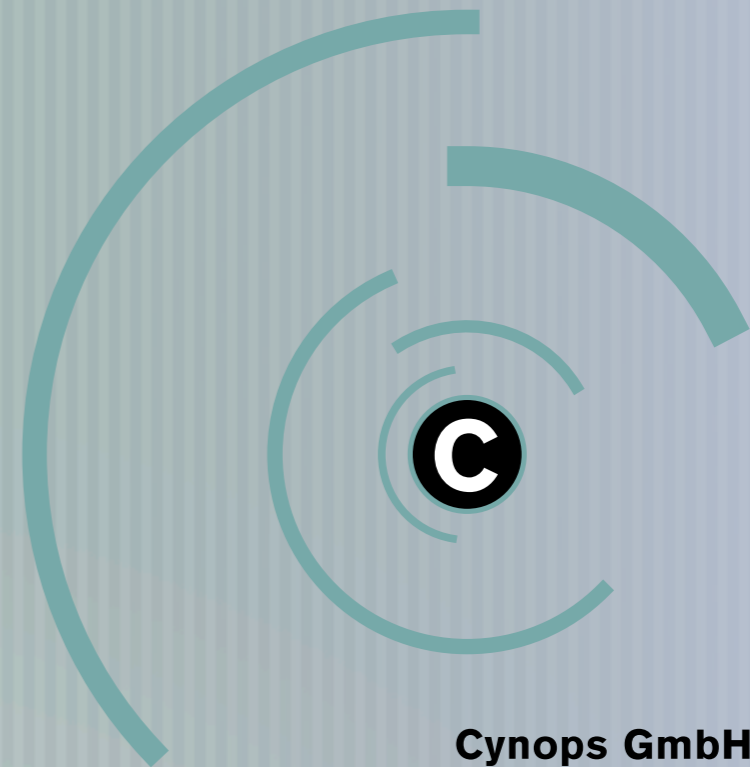
# Infrastructure Resiliency

— [ CA side support for Rollover is required for infrastructure resiliency

— Enter Multi-CA and CA Rollover Support

Thanks for your  
attention!

Martin Bartosch



**Cynops GmbH**

info@cynops.de

Kirchgasse 10c

61449 Steinbach (Taunus)

T (+49) 0 61 71.6 98 18 03

F (+49) 0 61 71.6 98 18 09

<http://www.cynops.de/>