ISGS 2016

Are your keys safe?

Marcel Dasen
VP Engineering

Securosys SA

Keys ?

Encryption keys

- asymmetric e.g. RSA, ECC public/private **key** pairs for wrapping
- symmetric e.g. AES, RC4, 3DES **keys**

Signature keys

- asymmetric e.g. DSA, ECDSA public/private **key** pairs for signing (of object hashes)
- hash algorithms e.g. SHA-2, SHA-3

Certificates

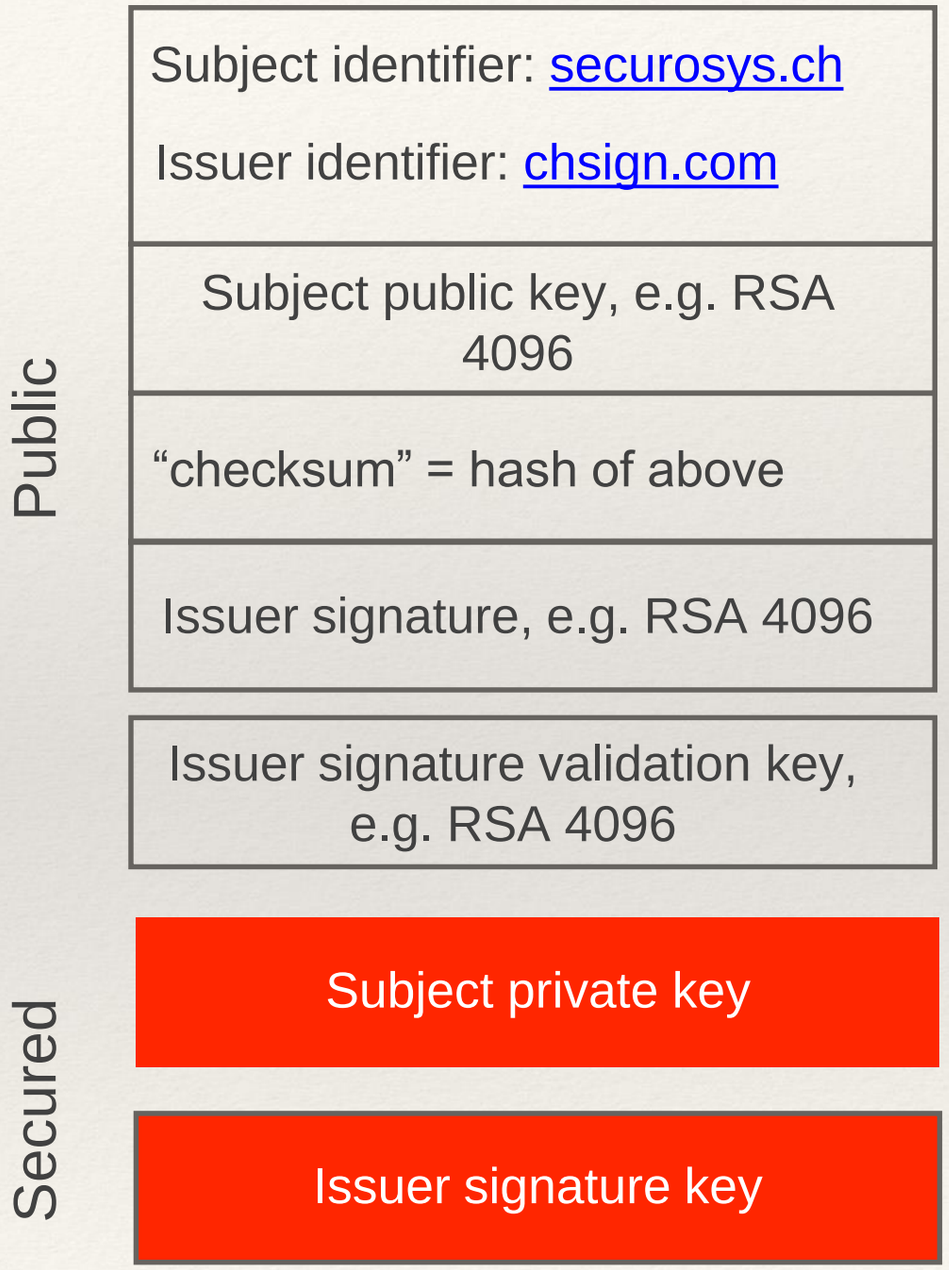
- digitally signed **public keys**

Where are cryptographic keys used?

- Web server authentication (e.g. https), “SSL” certificates, EV-certificates
- VPN client and concentrator authentication (IPSEC, SSL VPN)
- Email signatures (S/MIME)
- Data volume encryption (e.g. MS virtual smart card)
- Document signing
- SQL server
- Service authentication (mail - mail server, ssh, sDNS,…)

Definition “certificate”

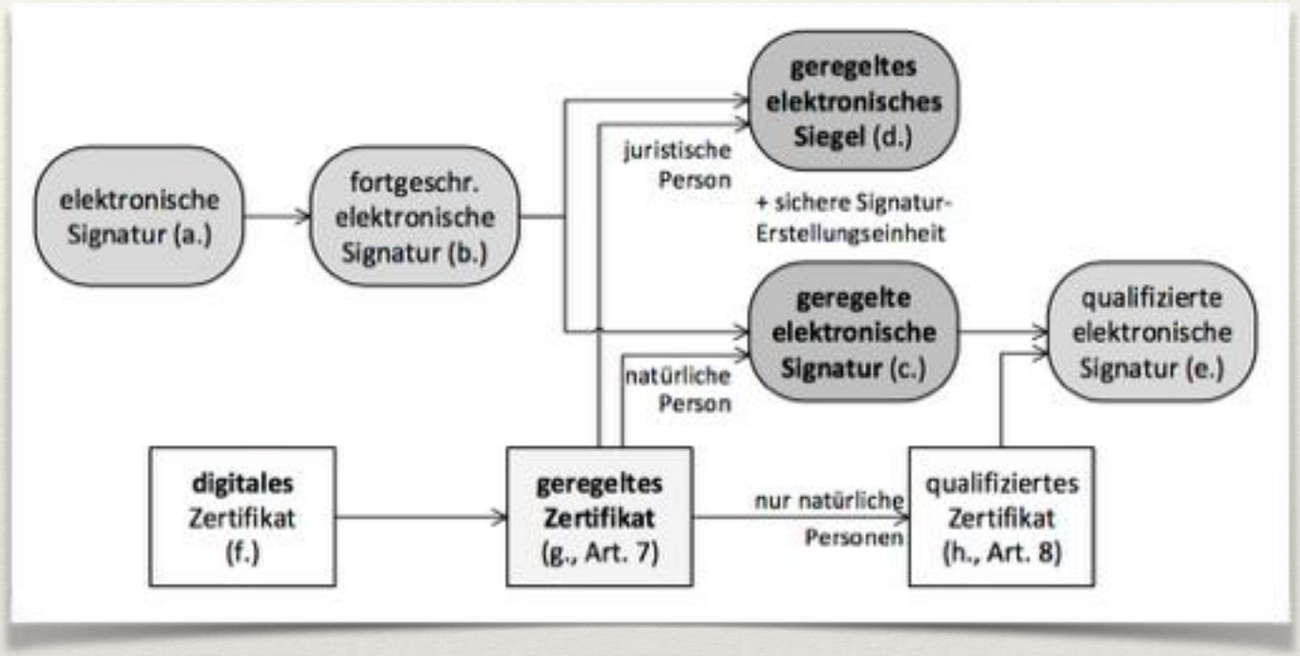
- In general a certificate is an object, which a trusted entity certifies its origin or contents
- a “digital certificate” typically refers to a
  - “public key”
  - subject unique information (e.g. securosys.ch)
  - issuer unique information (e.g. VeriSign)
  - validity period
  - “digital signature”



securosys

Special case example: Digital signature

“qualifizierte” elektronische Signatur



Equivalent to personal hand written signature
==> legally binding contracts

Codified in ZertES (SR 943.02), equivalent codifications in all EU countries [3]

Requires a certified certificate issuing device (CC EAL4+, FIPS140-2 L3, or equivalent) and audited procedures (CP/CPS)

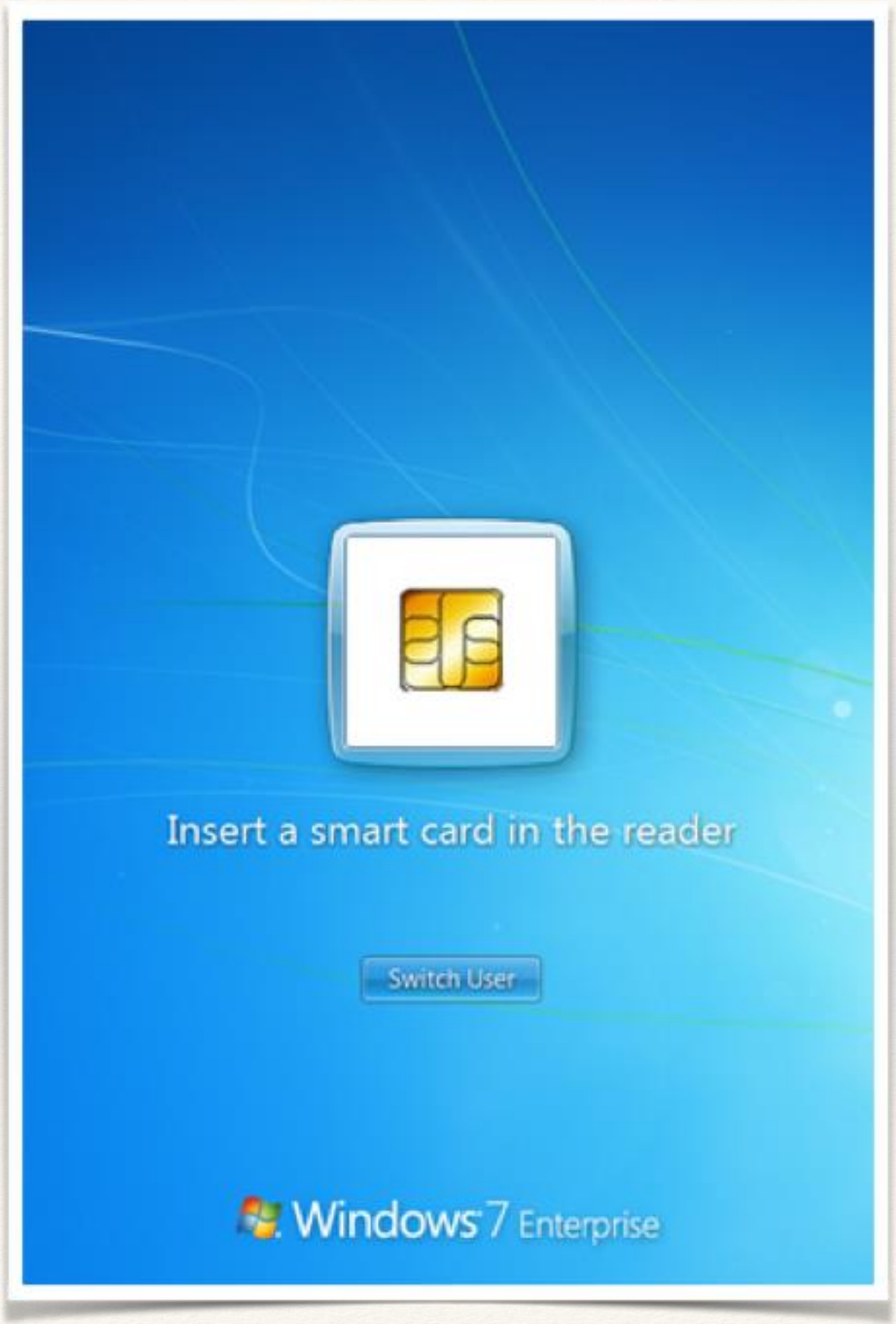
securosys

Example: Smart card login

Two factor login

Certificate and private key stored on smart card

Pin “stored” in user’s brain to unlock smart card



securosys

Web server “SSL” certificate store

Example: Web server

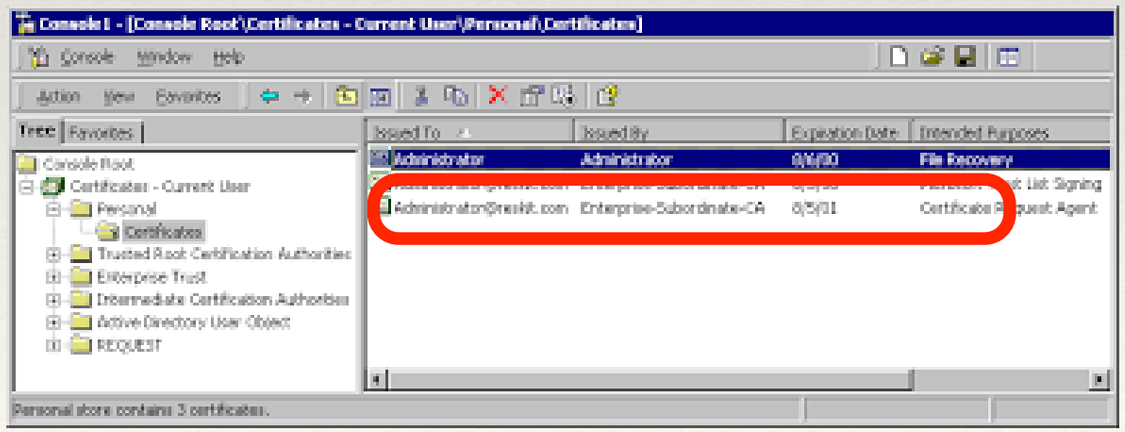
- Apache
  - file system

```
<VirtualHost 192.168.0.1:443>
DocumentRoot /var/www/html2
ServerName www.yourdomain.com
SSLEngine on
SSLCertificateFile /path/to/your_domain_name.crt
SSLCertificateKeyFile /path/to/your_private.key
SSLCertificateChainFile /path/to/DigiCertCA.crt
</VirtualHost>
```

- nginx
  - file system

```
server {

listen   443;

ssl    on;
ssl_certificate    /etc/ssl/your_domain_name.crt;
ssl_certificate_key    /etc/ssl/your_domain_name.key;

server_name your.domain.com;
access_log /var/log/nginx/nginx.vhost.access.log;
error_log /var/log/nginx/nginx.vhost.error.log;
location / {
      root   /home/www/public_html/your.domain.com/public/;
      index  index.html;
}

}
```

- MS IIS
  - certificate store



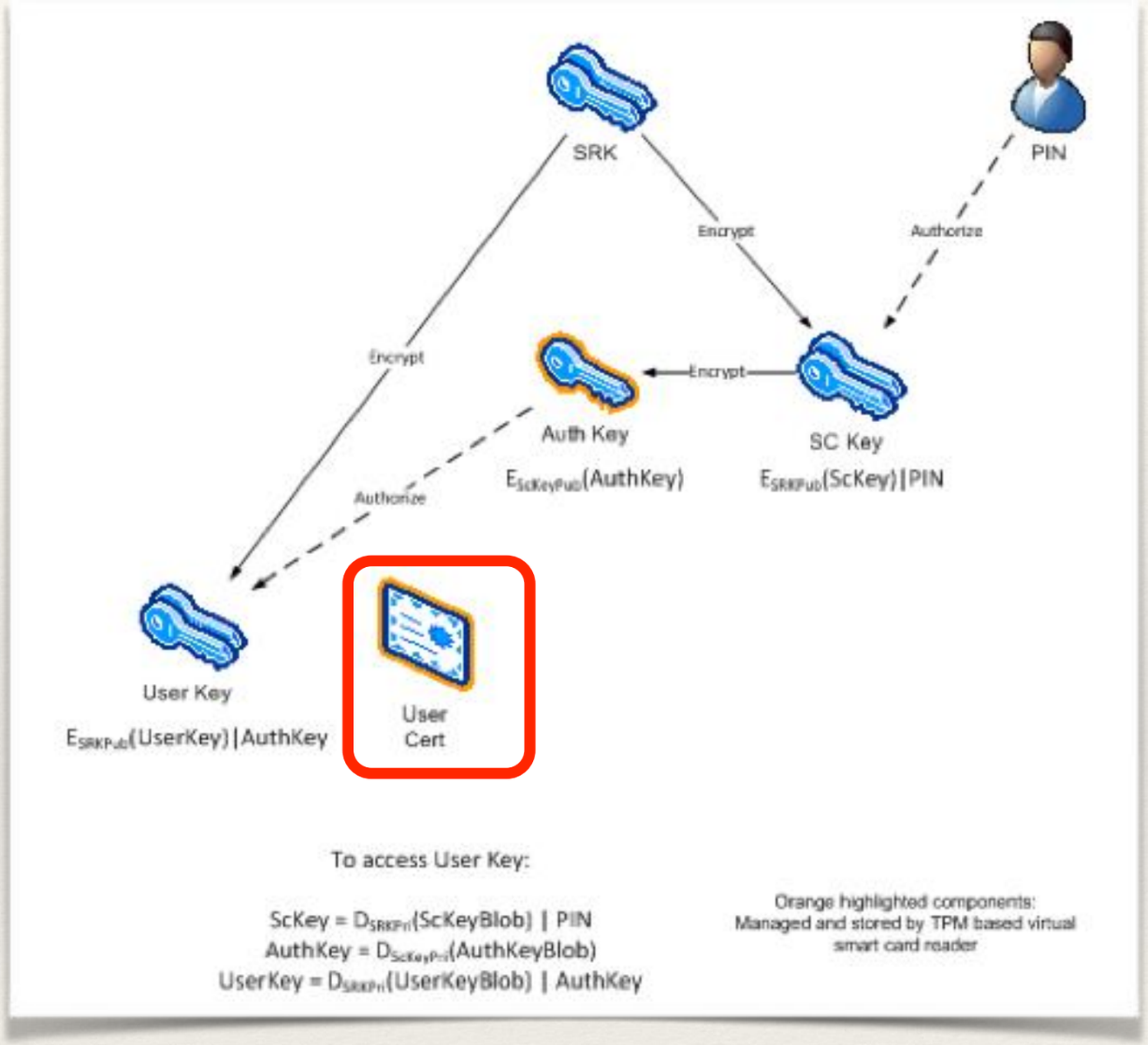
securosys

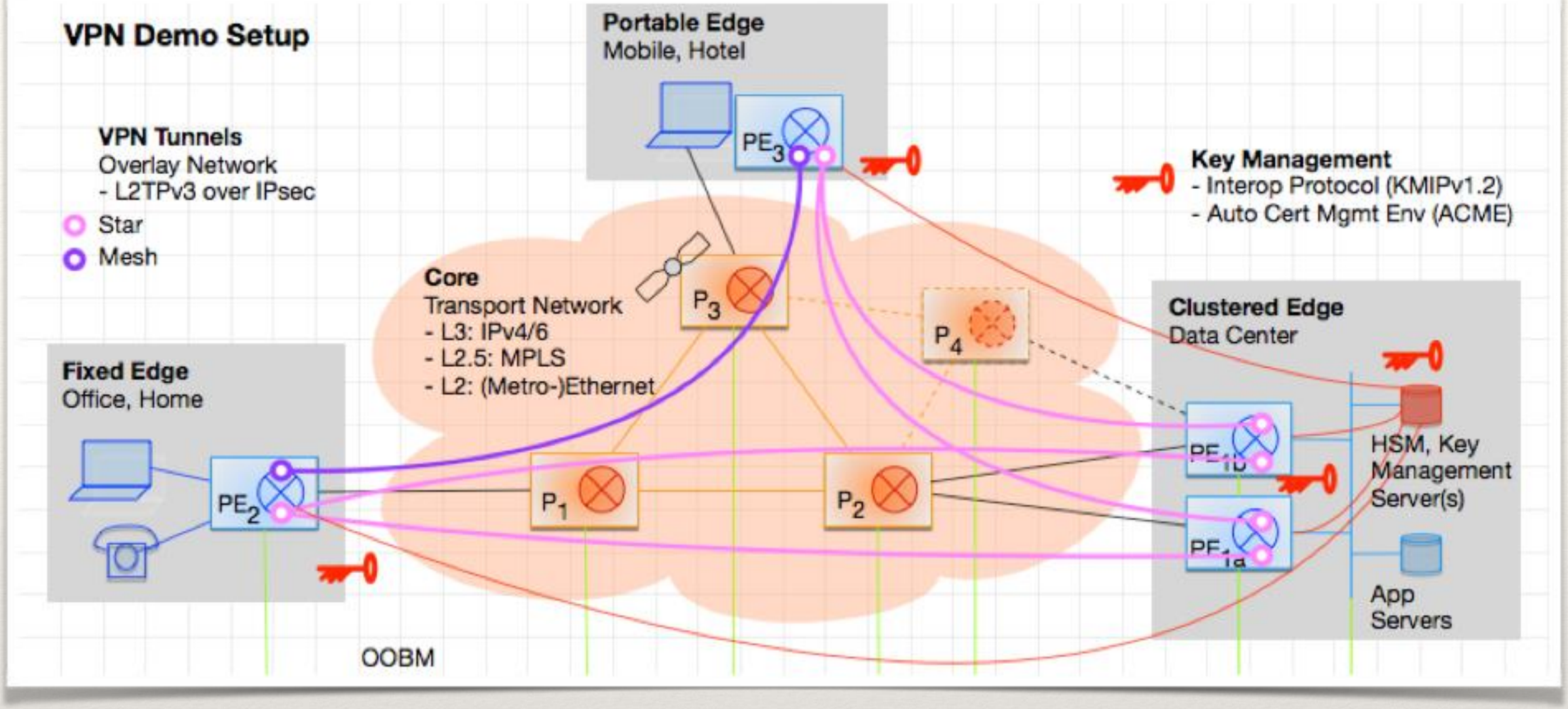
Example Microsoft storage encryption

Virtual smart card

Stores keys on hard drive

- User key (possibly stored in TPM)
- Smart card key, which is encrypted by the storage root key
- Authorization key for the user key decryption, which is encrypted by the public portion of the smart card key





Example keys in a virtual private network setup

VPN keys

How are my keys protected ?

- Different OS have different strategies
  - By software encryption (but where is the encryption key stored ?)
  - Trusted platform module (just one per machine)
- **Software cannot protect your keys**
  - A trust anchor is required

Do I care? What are the risks?

- **Loss of** “qualifiziertes Zertifikat”, “geregeltes Zertifikat” **signature certificate private key**
  - Legally binding signatures can be issued
- **Loss of communication certificate private keys**
  - interception possibility (eaves dropping, man in the middle)
  - potentially conflict with data protection laws
- **Loss of storage keys**
  - loss of intellectual property,
  - business continuity cost (e.g. after loss of credit card holder data)
  - potentially conflict with data protection laws
- **Loss of “SSL” certificate private key** (e.g. EV certificate)
  - Web site, email impersonation => attack vector
  - Risk for reputation with customers / partners

Linux (and various Unix)

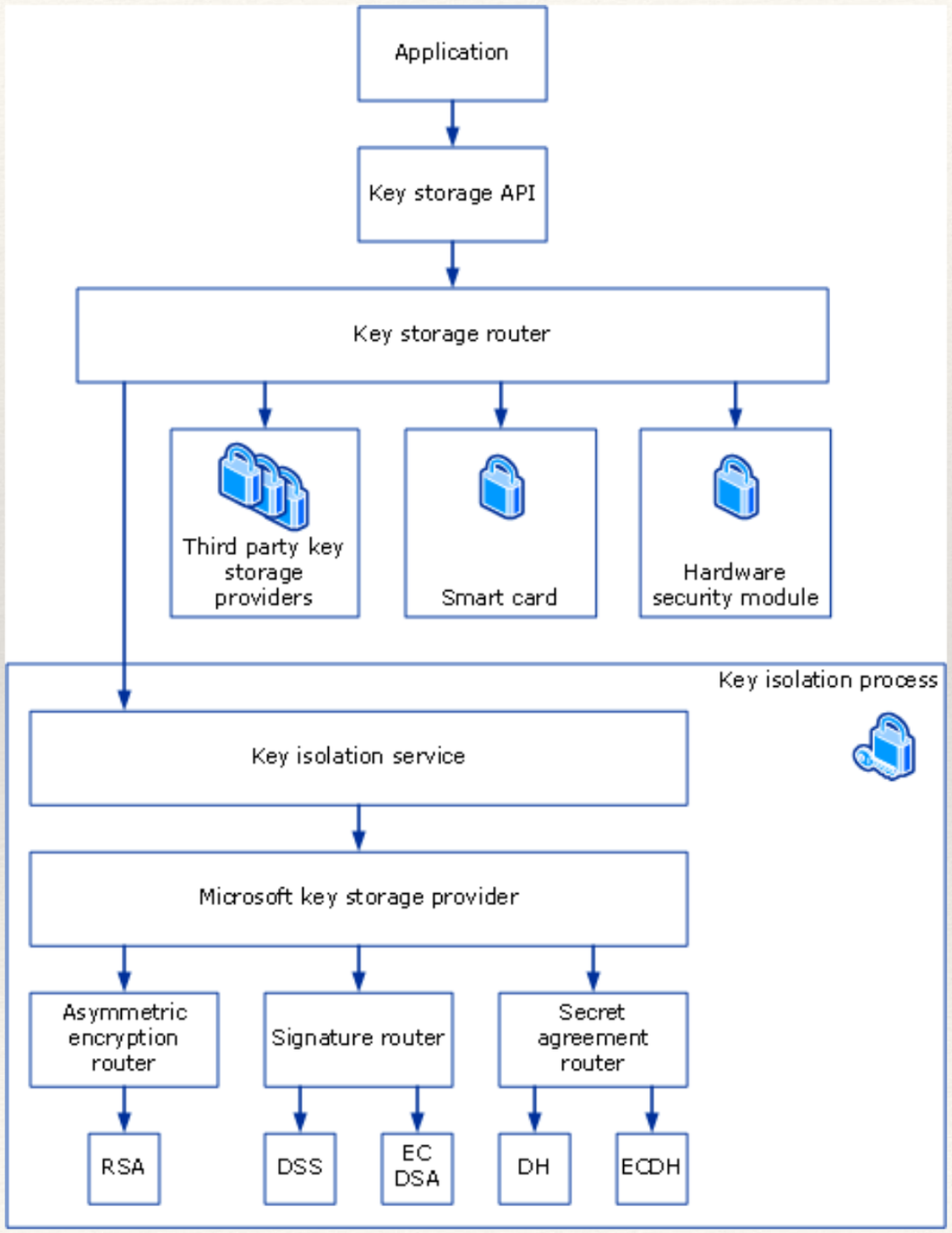
- Keys are protected by user / group file permissions
  - privileged users (e.g. “root”) can access everything
- Keys are typically stored in files, e.g. in users home directory

```
dasen$ ls -al .ssh
drwx------  1 dasen  staff   264 Jan 27  2015 .
drwxr-xr-x+ 1 dasen  staff  1792 May 12 16:21 ..
-rw-------  1 dasen  staff  1766 Jan 27  2015 id_rsa
-rw-r--r--  1 dasen  staff   400 Jan 27  2015 id_rsa.pub
```

Microsoft

- Keys are stored encrypted in
  - file system for standalone,
  - user profile for domain
- Encryption key per user (user master key)
  - rolled over regularly
  - stored in profile encrypted with logon password and user SID.

MS Key Storage Provider (KSP)



[1]

MS CNG private key storage

Key type	Directory
User private	%APPDATA%\Microsoft\Crypto\Keys
Local system private	%ALLUSERSPROFILE%\Application Data\Microsoft\Crypto\SystemKeys
Local service private	%WINDIR%\ServiceProfiles\LocalService
Network service private	%WINDIR%\ServiceProfiles\NetworkService
Shared private	%ALLUSERSPROFILE%\Application Data\Microsoft\Crypto\Keys

Keychain on OSX

- Keys (certificates, passwords, etc.) are stored via the keychain application
- The keychain is located in the users profile directory
  - ~/Library/Keychains
- Keychains are encrypted with password derived keys
- Login key chain can be accessed with “root” privileges

The trust anchor

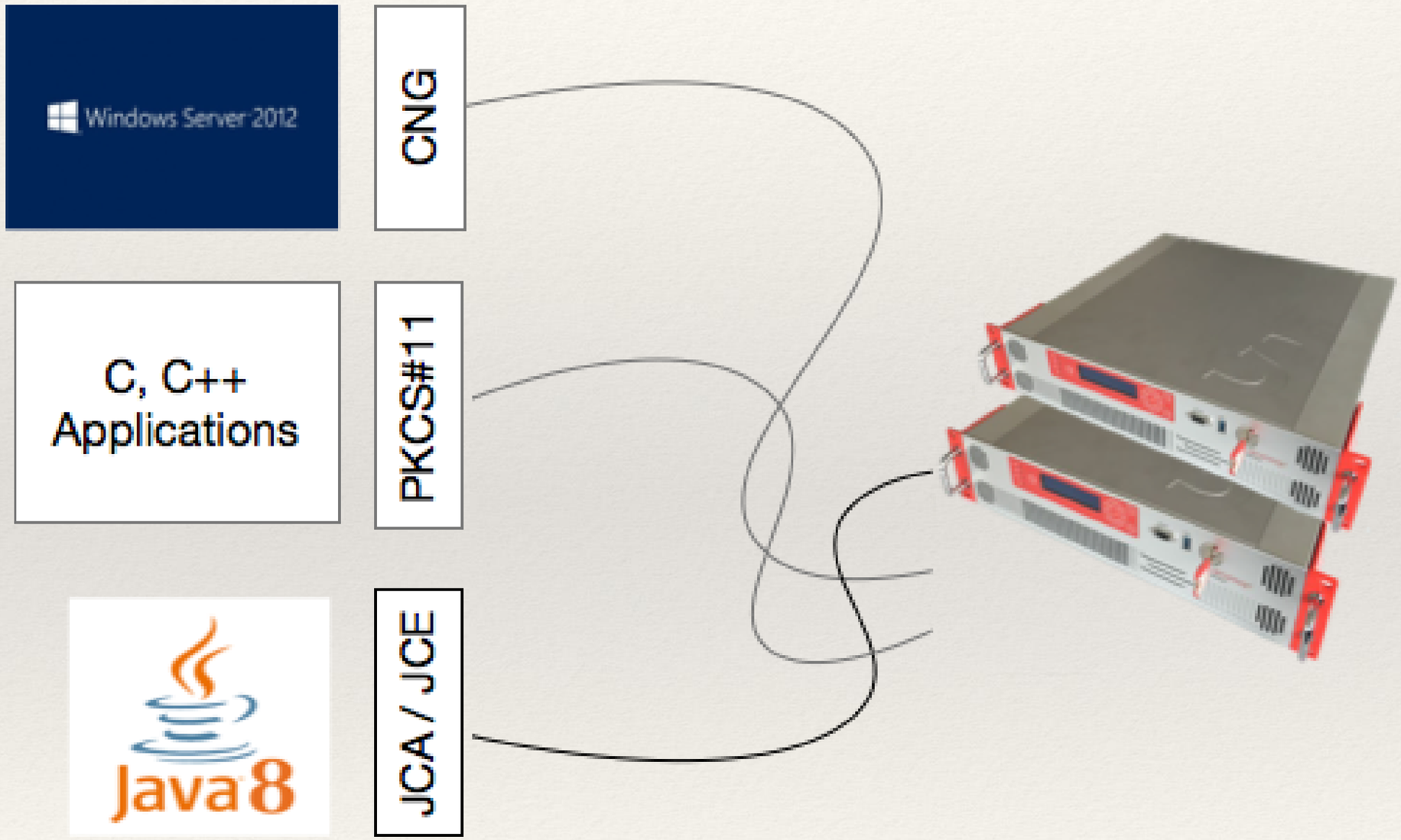
HSM is a Trust Anchor

- Generates trusted certificates and keys
- Safeguards keys = Key storage

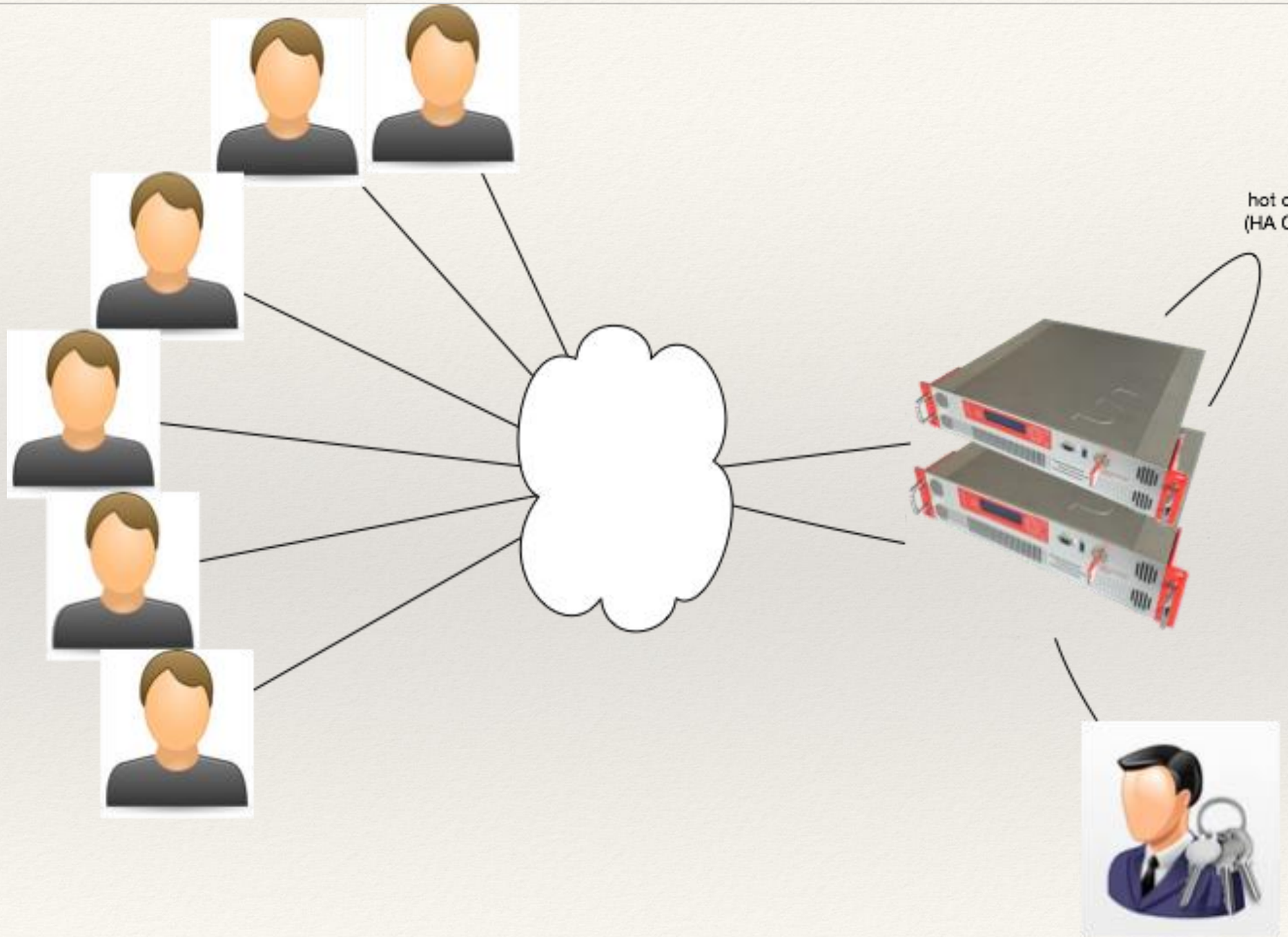
Different form factors

- Hardware security module (performance)
- Smartcard (portability)

Integration of a trust anchor



Multiple applications, one trust anchor



securosys

Generating and managing keys

- Use a secure key generation device (HSM)
  - Hardware random number generator
    - Validated encryption algorithms
- Keep private keys in protected storage
  - Hardware security module
- Use a Software CA or Windows built-in CA

reverse proxies – nginx – Apache

Example HSM with open source

- To use an HSM, PKCS#11 API support is required. PKCS#11 library is vendor specific.
- Configure OpenSSL Engines library to use vendor PKCS#11 interface.
- Simplified support via “libp11”

Microsoft IIS

Example HSM with Windows server

- Microsoft supports a unified API for crypto operations via CNG (CryptoAPI next gen)
- CNG provider is vendor specific
- Key storage is implemented via a key storage provider (KSP)
  - MS default is an “isolated” process
  - Protected hardware can be attached by installing specific KSP
- IIS will use standard APIs.

Summary

- Digital keys are a crucial part of the digital world.
- Keys must be safeguarded. You want to know, where and when they are accessed
- Keys are trusted if generated with a trustworthy key generator.
- For trust you need a physical trust anchor.

References

1. https://msdn.microsoft.com/en-us/library/windows/desktop/bb204778(v=vs.85).aspx
2. https://technet.microsoft.com/en-us/library/bb895340.aspx
3. Botschaft zur Totalrevision des Bundesgesetzes über die elektronische Signatur (ZertES)
4. http://nginx.org

securosys

SWISS SECURITY TECHNOLOGIES
FOR COMMUNICATIONS SYSTEMS