Red Hat Identity Management

Certificate System Technical Overview

Shawn Wells, RHCE
Account Manager, Intelligence Programs
sdw@redhat.com / 443.534.0130

Agenda

- Overview of PKI
- Overview of Red Hat Certificate System
- Key Features of Certificate System:
  - Token Innovations
  - Scalability and Performance
  - High Availability and Disaster Recovery
  - Tools and SDKs
  - NSS Crypto Libraries
  - New Features
- Benefits and Roadmap
- Next Steps
- Questions

Objectives

This presentation will help you understand the following:

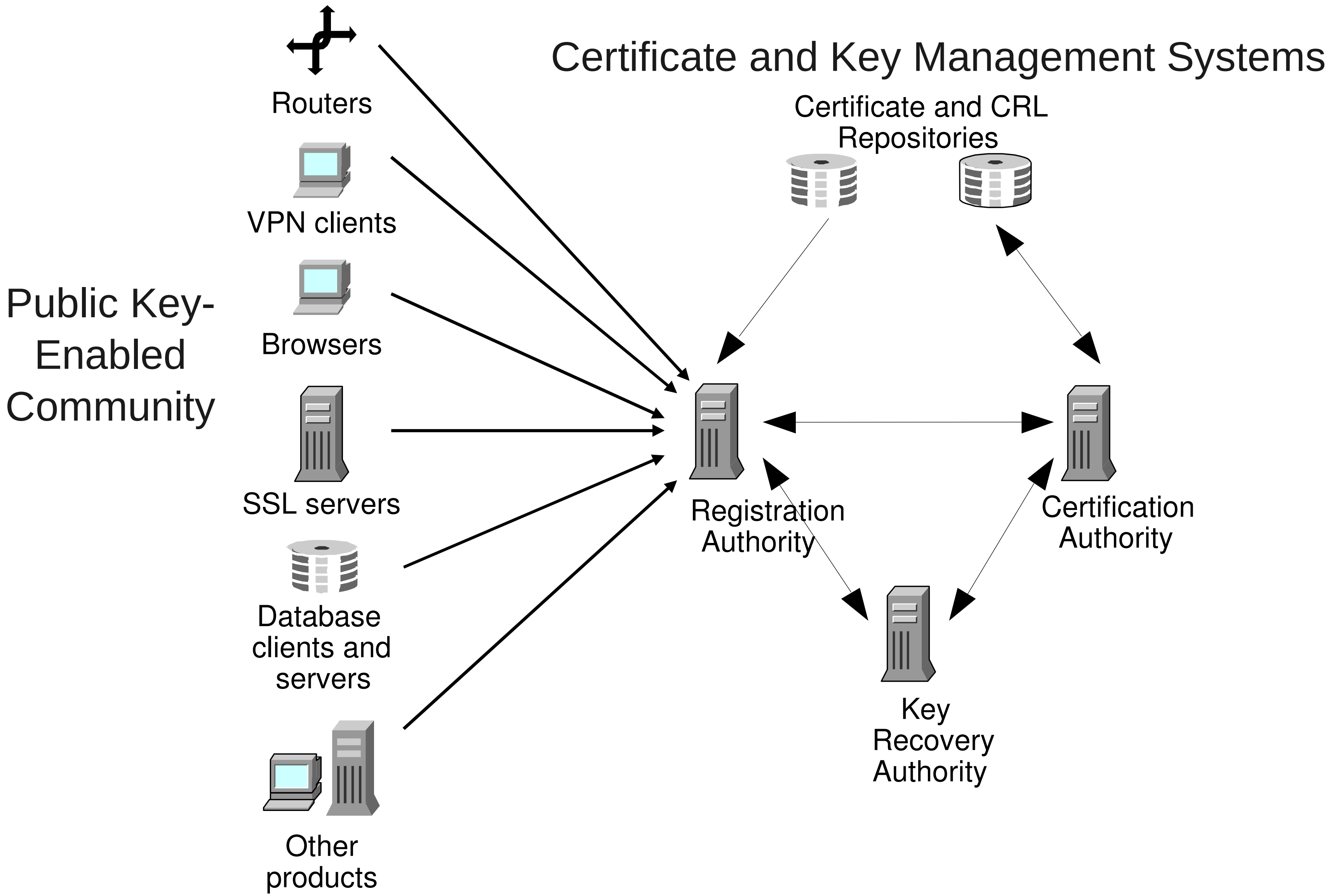
- Basic functions of a PKI
- Certificates, CAs and Certificate Hierarchies, Revocation and Verification
- Basic Red Hat Certificate System characteristics
  - Simplified Assurance/Basic Functions
  - Major components and ESC
  - Architecture
- Significance of key Certificate System features, including
  - Token Innovations
  - Availability and Disaster Recovery
  - Role of NSS
- Product benefits and roadmap

Overview of PKI

What Is PKI?

- **P**ublic **K**ey **I**nfrastructure
- Set of standards and services that facilitate the use of public-key cryptography in a networked environment
- SSL uses PKI: cornerstone of Internet commerce
- Benefits:
  - Allows two strangers to communicate in a secure fashion
  - Permits authentication without requiring user to send secret over the wire (unlike name & password)
  - Encryption protects confidentiality of sensitive information
- Problems:
  - Enrollment and initial application configuration has historically been a difficult problem to solve

What a PKI Looks Like


Certificate and Key Management Systems
Routers
VPN clients
Browsers
SSL servers
Database clients and servers
Other products
Public Key-Enabled Community
Certificate and CRL Repositories
Registration Authority
Certification Authority
Key Recovery Authority

Overview of Red Hat Certificate System

Red Hat Certificate System: Simplified Assurance

- Highly flexible, standards-based PKI solution
- Built on open source Network Security Services (NSS) crypto libraries used by Mozilla, all Netscape Servers, and Sun Directory Server.
- Unique approach with integrated smartcard deployment
- High scalability and performance via integrated Directory Server
- Unmatched availability and disaster recovery
- IPS-140-2 certification underway for NSS
- Common Criteria certification by NIAP (partnership between NSA and NIST) at Evaluation Assurance Level 4 augmented under CIMC protection profile
- Java SDK and tools

Basic Functions

- Issues certificates
- Issues CRLs
- Modular deployment – web based
- Archives user’s private keys (optional)
- Lots of auditing
- Flexible access control
- Provides a management interface

Supported Platforms (RHCS 8.0)

- Supported Platforms for Certificate Server
  - RHEL 5.3+
- Supported Platforms for Enterprise Security Client
  - Microsoft Windows Vista 32-bit
  - Microsoft Windows Vista 64-bit
  - Microsoft Windows XP 32-bit
  - Microsoft Windows XP 64-bit
  - Red Hat Enterprise Linux 5.3 x86
  - Red Hat Enterprise Linux 5.3 x86_64

IMPORTANT: The Enterprise Security Client was supported on Apple Mac for Red Hat Certificate System 7.x, but is not supported on Mac for 8.0.

Supported Platforms (RHCS 8.0)

- Supported Smart Cards
  - The Enterprise Security Client supports Global Platform 2.01-compliant smart cards and JavaCard 2.1 or higher.
  - The Certificate System subsystems have been tested using the following tokens:
    - Gemalto TOP IM FIPS CY2 64K token, both as a smart card and GemPCKey USB form factor key
    - Gemalto Cyberflex e-gate 32K token (Red Hat Enterprise Linux only)
    - Safenet 330J Java smart card
  - Smart card testing was conducted using the SCM SCR331 CCID reader.
  - The only card manager applet supported with Certificate System is the CoolKey applet which ships with Red Hat Enterprise Linux 5.3.

- Supported Hardware Security Modules (HSM)
  - Safenet Chrysalis-ITS LunaSA
  - nCipher netHSM 2000

Main Components

- **Certificate Authority (CA):** Issues X.509 digital certificates and CRLs
- **Token Management System (TMS):**
  - Supports Global Platform smartcards & software tokens
  - Makes smartcards as easy to use as an ATM
- **Registration Authority (RA):** Supported for the benefit of pre-7.0 deployments
- **Data Recovery Manager (DRM):**
  - Secure repository for backup/recovery of user's private keys
  - Configurable multi-person approval for recovery
- **Online Certificate Status Protocol (OCSP) Responder:**
  - Responds to OCSP requests to verify certificate validity in real time
- **Token Key Service (TKS)**
  - Manages symmetric keys for securing communication between subsystems and tokens
- **Registration Authority (RA)**
  - Front-end subsystem to offload certain CA tasks such as local authentication, requestor information gathering and request validation

Key Features of Red Hat Certificate System

Key Features: Token Innovations

- Certificate System works with Global Platform compatible smartcards (tokens)
  - Greatly simplifies enrollment and all other aspects of token management
  - Customizable enrollment process
  - First to market with integrated soft certificate/hardware token solution
- Enterprise Security Client:
  - Runs on RHEL, Windows
  - Facilitates communication between Certificate System back end and token
- Firefox and Thunderbird “do the right thing” with tokens
  - We have built special versions with improved token support
  - Code contributed to Mozilla projects

Key Features: Scalability & Performance

- Solid HTTP Engine Based on Netscape Enterprise Server
- Database optimizations
- Lab tests:
  - Issued over 12 million certificates from single server in less than 35 days (~14,000 certificates/hour)
  - Simultaneously published to Directory Server and archived private keys
  - 10% of certificates revoked, resulting in 1.2-million-entry CRL
  - Generated CRL in less than 30 minutes

High Availability & Disaster Recovery

Cloning/failover mechanism:

- Reduces unplanned outages by making one or more subsystem clones available for failover
- CA, DRM, and OCSP Responder can be cloned
  - CA key material available 24x7
  - Data sources for cloned systems are replicated, so data is shared seamlessly between subsystem databases
- Master and cloned instances typically installed on different machines behind a load balancer
- When a failure occurs, load balancer transparently redirects all requests to a clone that's still running, without any service interruption

Tools & SDKs

- Java SDK for integrating with other enterprise applications
  - Documentation for creating plug-ins
  - Bootstraps authentication mechanisms using existing databases and other applications
  - Facilitates customized publishing, e.g. to trigger billing when a certificate is published
- Uses Console, a GUI application for typical admin tasks
- Command-line administrative and testing tools for additional tasks

NSS Crypto Libraries

- Open source C libraries designed to support cross-platform development of security-enabled client and server applications
  - Tri-license: GPL, LGPL, MPL
- Underlies crypto features of Mozilla clients, all Netscape servers, Sun Directory
- Highly portable codebase: supports 20+ platforms
  - Available as RPMs on Red Hat Linux
- Crypto algorithms, X.509 v3 certificates, CRLs, OCSP, SSL/TLS, S/MIME, PKCS #5, PKCS #7, PKCS #10, PKCS #11, PKCS #12, etc.
- Smartcard and other hardware crypto device support
- JSS: open source Java bindings for NSS
  - Gives Java programs access to NSS via JNI

Government Support

- Fully compatible with Federal Bridge
  - Gateway mechanism used by government agencies
- FIPS-140-2 certified
  - Third-party crypto certification required for government contracts
- Support for certificate issuance with Windows extensions for Windows Smartcard Logon

New Features of 7.2 & 7.3

Modularization

- Red Hat Certificate System 7.2 consists of ~75 separate RPM modules.
  - Core RHCS packages (CA, OCSP, TPS, Common tools, etc.)
  - Apache
  - Tomcat
- Easier to maintain, easier to support
- Modular architecture based upon Filesystem Hierarchy Standard (FHS) 2.3

Auto Enrollment Proxy Operation

- Red Hat Certificate System Proxy agent running within a Windows Domain
  - Users & Computers registered in a domain can automatically discover the Auto enrollment services
  - Certificate requests from Windows computers are authenticated and then automatically forwarded to a Red Hat Certificate Authority
  - After issuance the certificates are returned via the proxy
  - Certificate will automatically be installed into the requesting application
- Operation completely transparent to Windows clients

Registration Authority

- Accepts enrollment requests and authenticates them in a local context
- Allows CA to run behind a firewall
- Allows delegation of Certificate approvals
- Email notification of certificate status
- Pluggable interface for additional authentication mechanisms or work flow

Other New Features

- Improved Token Support
  - DRM-generated private keys, archival, and key recovery for tokens
  - Injection of wrapped private key from DRM during enrollment/recovery
- ESC Improvements:
  - Support for key recovery, PKCS #11 interface
  - Client installers (including security libraries)
- Improved migration support, including to Red Hat Enterprise Linux
- SHA-256 and SHA-512
- HSM Support: nCipher nShield 9.01, Chrysalis Luna SA 3.1

Certificate System 8.0

Certificate System 8.0

- **Certificate Renewal**
  Certificate renewal for all Certificate System-issued certificates has been reintroduced using the new profile framework. There are a number of new profiles to use for renewal, including encryption and signing certificates for both standard use and on tokens, and server certificate renewal. New inputs have been added to manage certificate renewal, so corresponding renewal profiles can be created for custom enrollment profiles.

- **Improved Subsystem Cloning**
  Cloning has been enhanced with distributed numeric assignments logic so that cloned CAs can efficiently divide and use serial numbers for certificates without becoming blocked because of inadequate serial number ranges.

- **Stronger SELinux Policies**
  SELinux policies are now required for every subsystem and run in enforcing mode by default, providing much more protection for Certificate System processes.

Certificate System 8.0

- **Improved UTF8 Support**
  The CA, OCSP, and DRM subsystems fully accept and interpret certificate requests generated using UTF-8 characters, both in the console and in the agent services pages. This support is for specific fields.

  End users can submit certificate requests with UTF-8 characters in those fields and end users and agents can search for and retrieve certificates and CRLs in the CA and retrieve keys in the DRM when using those field values as the search parameters.

  Four fields fully-support UTF-8 characters:
  - Common name (used in the subject name of the certificate)
  - Organizational unit (used in the subject name of the certificate)
  - Requester name
  - Additional notes (comments appended by the agent to the certificate)

- NOTE: This support does not include supporting internationalized domain names, like in email addresses.

Certificate System 8.0

- **Enhanced Support for Third-Party ECC Modules**
  Certificate System 8.0, although it does not ship with an ECC module, does support loading and using third-party ECC PKCS#11 modules with the CA. The console can handle ECC-based SSL sessions, and the server generates and supports ECC certificates.

- **Simplified Signed Audit Logging**
  Audit log signing certificates are now created with all of the other default subsystem certificates as soon as a CA, DRM, OCSP, TKS, or TPS subsystem is configured. The log is also already configured and can be very easily enabled. Signed audit logs can be verified by auditors using the included AuditVerify script.

- **New Windows Smart Card Login Profile for Tokens**
  A new example profile is included with the regular CA profiles list which enabled the CA and TPS to issue certificates and enroll tokens that can be used to log into Windows systems.

Certificate System 8.0

- **Expanded TPS Roles**
  A new role, the operator role has been added to the TPS subsystem. This role can view and search all tokens, certificates, and activities within the Token Processing System (TPS) but cannot edit any entries.

  Additionally, the administrator role interface has been enhanced to allow administrators to create and edit users, assign profiles, and delete users directly.

- **Added IPv6 Support**
  The Certificate System 8.0 services can accept requests from all supported browsers, from other Certificate System subsystems, and from the administrative console over IPv6. The server also supports using IPv6 addresses in the Subject Alt Names of certificates, with certificate extensions, and with Certificate System scripts and tools.

Product Benefits & Roadmap

Benefits

- Mature product: 10+ years of specialized engineering expertise
- End-to-end solution
  - Uses same NSS crypto libraries as Mozilla products
  - Leverages Red Hat Directory Server features and performance
  - Supports heterogeneous environments
  - Key element of cross-platform Red Hat identity management solution
- Easy for users
  - Mature life cycle management
  - Protected from complexity of PKI
  - Military-grade crypto that just works
  - Fewer calls to Help desk
  - Simplified smartcard deployment and usability: Functions like an ATM card

Benefits (Cont'd)

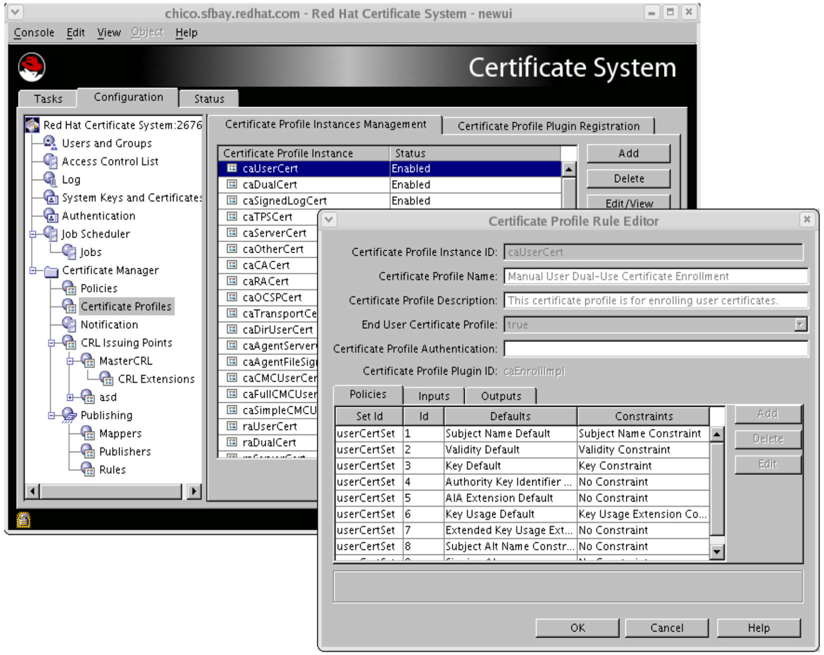
- Robust administration
  - High availability and automated disaster recovery through Directory-based Multi-Master replication, cloning, and failover
  - Console application provides GUI for routine tasks
  - Mature command-line tools permit specialized or bulk operations
  - Remote smartcard administration
  - Hides complexity without sacrificing flexibility
- Consistent, reliable maintenance after initial rollout
  - Red Hat commitment to support and training
  - Red Hat Network for hot fixes, updates, new releases

Configuration

- Config files:

```
internaldb.ldapauth.bindPWPrompt=Internal LDAP Database
internaldb.ldapauth.clientCertNickname=
internaldb.ldapconn.host=localhost
internaldb.ldapconn.port=3100
internaldb.ldapconn.secureConn=false
jobsScheduler.enabled=false
jobsScheduler.interval=1
```

- Graphical Console:



Certificate Profile Instance	Status
caUserCert	Enabled
caDualCert	Enabled
caSignedLogCert	Enabled
caTPSCert	
caServerCert	
caOtherCert	
caCACert	
caRACert	
caOCSPCert	
caTransportCe	
caDirUserCert	
caAgentServer	
caAgentFileSig	
caCMCUserCer	
caFullCMCUser	
caSimpleCMCU	
raUserCert	
raDualCert	

Certificate Profile Instance ID: caUserCert

Certificate Profile Name: Manual User Dual-Use Certificate Enrollment

Certificate Profile Description: This certificate profile is for enrolling user certificates.

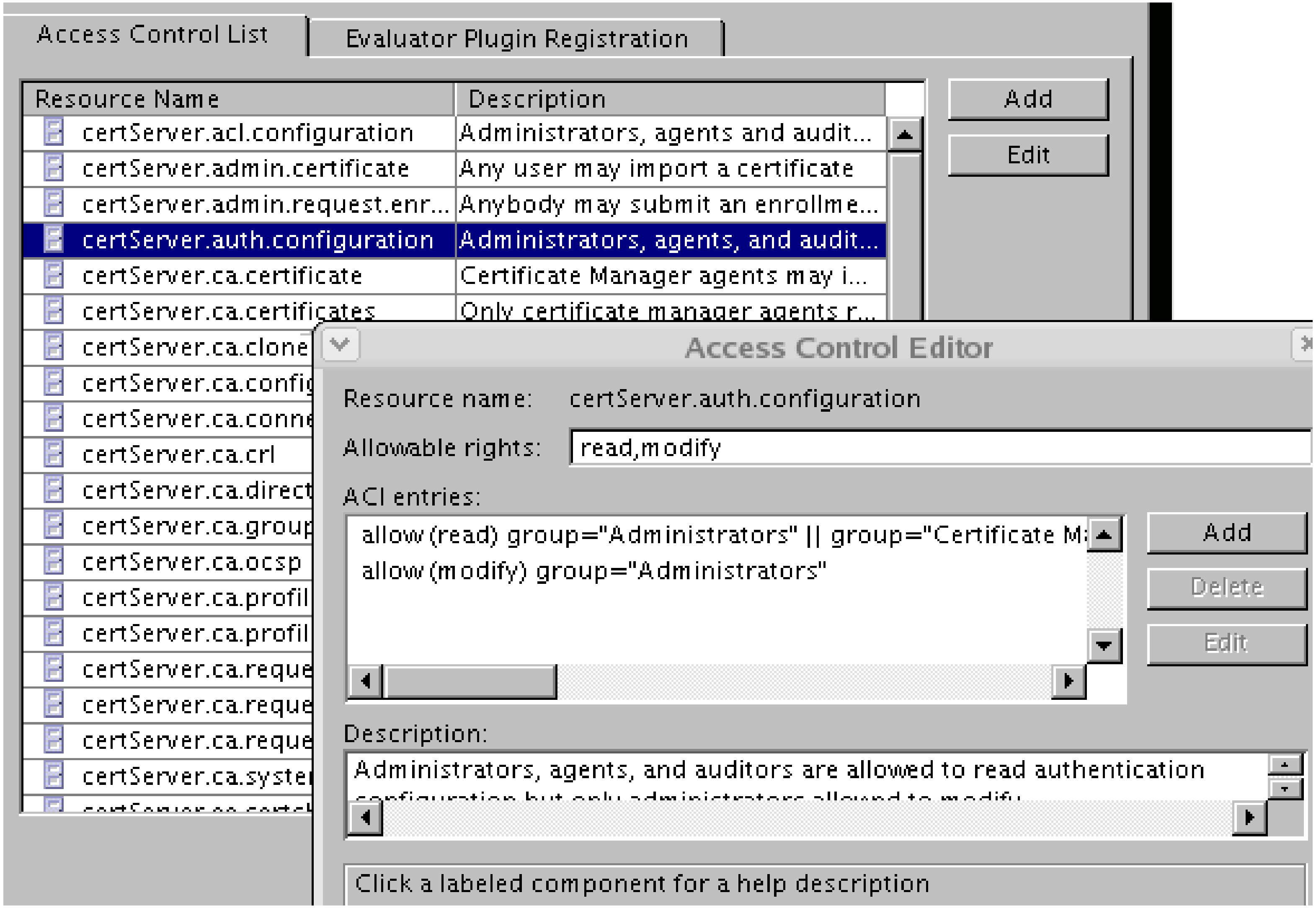
End User Certificate Profile: true

Certificate Profile Authentication:

Certificate Profile Plugin ID: caEnrollImpl

Set Id	Id	Defaults	Constraints
userCertSet	1	Subject Name Default	Subject Name Constraint
userCertSet	2	Validity Default	Validity Constraint
userCertSet	3	Key Default	Key Constraint
userCertSet	4	Authority Key Identifier ...	No Constraint
userCertSet	5	AIA Extension Default	No Constraint
userCertSet	6	Key Usage Default	Key Usage Extension Co...
userCertSet	7	Extended Key Usage Ext...	No Constraint
userCertSet	8	Subject Alt Name Constr...	No Constraint

Access Control Lists



Resource Name	Description
certServer.acl.configuration	Administrators, agents and audit...
certServer.admin.certificate	Any user may import a certificate
certServer.admin.request.enr...	Anybody may submit an enrollme...
certServer.auth.configuration	Administrators, agents, and audit...
certServer.ca.certificate	Certificate Manager agents may i...
certServer.ca.certificates	Only certificate manager agents r...
certServer.ca.clone	
certServer.ca.confic	
certServer.ca.conne	
certServer.ca.crl	
certServer.ca.direct	
certServer.ca.group	
certServer.ca.ocsp	
certServer.ca.profil	
certServer.ca.profil	
certServer.ca.reque	
certServer.ca.reque	
certServer.ca.reque	
certServer.ca.syste	

Resource name: certServer.auth.configuration

Allowable rights: read,modify

ACI entries:

```
allow (read) group="Administrators" || group="Certificate M:
allow (modify) group="Administrators"
```

Description:

Administrators, agents, and auditors are allowed to read authentication configuration but only administrators allowed to modify

Click a labeled component for a help description

Backup - Architecture

Certificate System Architecture

- Common Architectural Features
  - Web-based, configuration, subsystem connections, database, privileged user management, logging, cron jobs, user interface framework, plugin architecture
- Subsystem-Specific
  - CA Signing Unit, profile configuration
  - DRM Key Archival/Recovery Unit
  - OCSP protocol
- Multiple subsytems can be installed on same or different machines, and connected together

Certificate System Architecture

- Web-server based application
  - Java – CA, DRM, TKS components
  - C++ - TPS
- Opens multiple in-bound ports
  - Administration
  - Agent
  - End-Entity – end users
- Opens out-bound ports to other servers
  - LDAP
  - HTTP (e.g. CA->DRM connector)

Authentication/Authorization

- CS subsystem maintains list of privileged users
- Users can have one or more rules
  - Administrator
  - Auditor
  - Agent
  - Trusted Manager (Affiliated Subsystem)
- Users can authenticate with Password or Certificate depending on Role

Internal Database

- All CS data is stored in an LDAP database
- Types of data
  - Users
  - Certificates
  - Certificate Requests
  - CRLs
  - Private keys
  - Key Recovery Requests
  - Smartcard records

Logging

- HTTP transactions
- System
- Debug
- Signed Audit Log
  - 21005.Thread-19 - [19/Jul/2005:10:05:34 PDT]
  - [14] [6]   (source, level)
  - [AuditEvent=CERT_REQUEST_PROCESSED][SubjectID=tps][Outcome=Success][ReqID=170][InfoName=certificate][InfoValue=null] certificate request processed

Jobs

- Periodic 'cron'-style jobs which get run at intervals
- Java plugins get invoked
- Access to internal CA structures
- Example:
  - Send mail to users who certs are about to expire

SDK

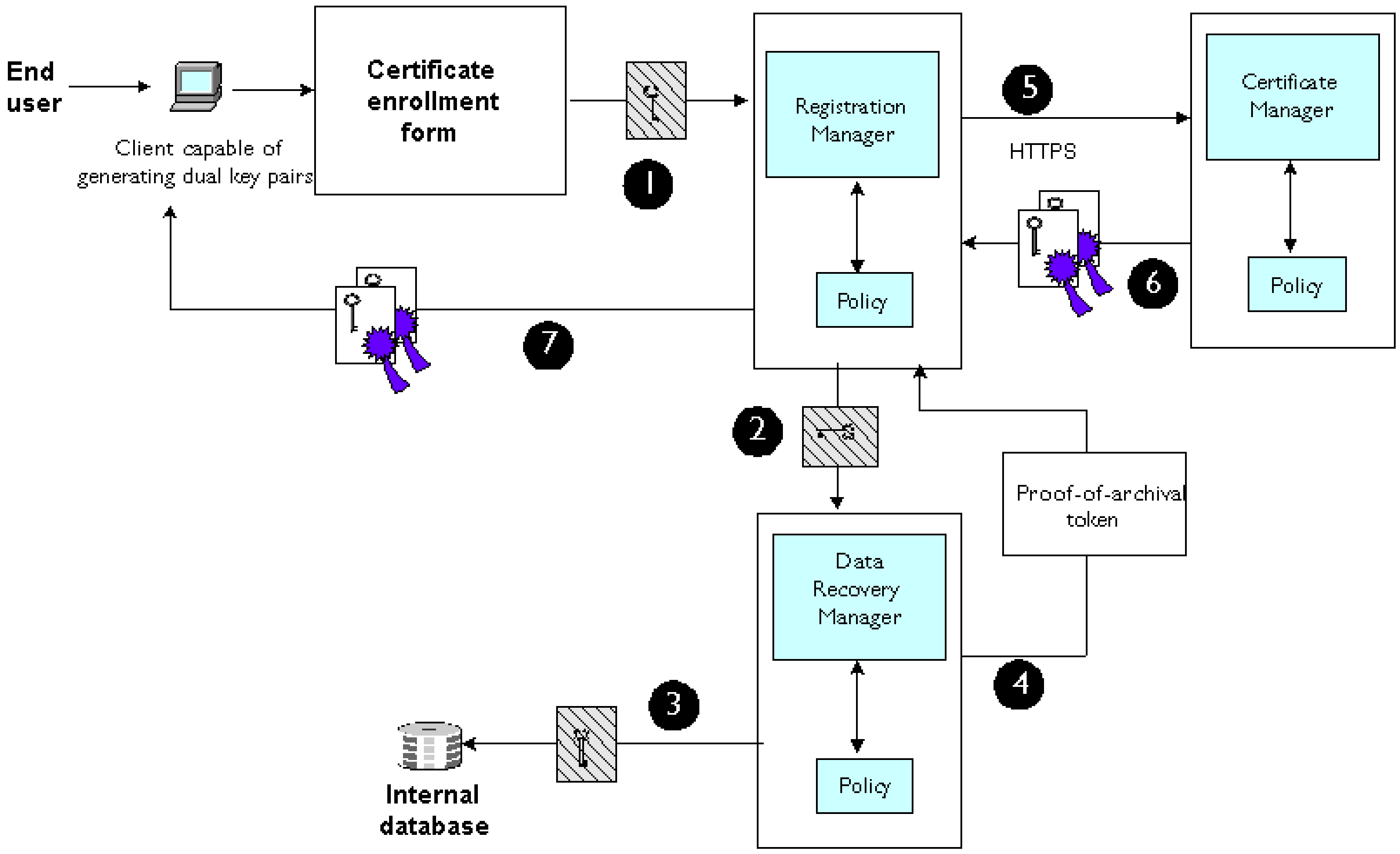
- Java plugins:
  - Jobs
  - Authentication plugins (LDAP, NIS, one-time PIN etc)
  - Certificate Extensions
  - Listeners
- HTTP interface

  -
    ```
    http://testca.redhat.com:5303/enrollment?csrRequestorName=++&CN=steve&UID=steve&E=steve
    %40redhat.com&OU=engineering&O=redhat&C=US&email=true&ssl_client=true&digital_signature=true&non_repud
    iation=true&key_encipherment=true&challengePassword=&confirmChallengePassword=&csrRequestorEmail=steve
    %40redhat.com&csrRequestorPhone=&csrRequestorComments=&subjectKeyGenInfo=MIG0MGAwXDANBgkqhkiG9w0BAQEFA
    ANLADBIAkEA0a195K0sThuui7D5T4DA3Eki%0AKXJEW7lrjA9bnlivjKpjjN5tEPBSBFzSGph%2FJ81Z5kVsz%2FJ
    %2FqLYLbAIQIyl7HQID%0AAQABFgAwDQYJKoZIhvcNAQEEBQADQQAhLPBeNqr7hIHFEKgmH0qV%2F7lu5%2FsegPys
    %0AZ2b3zOW2%2BTiHJNAinhs%2FtMlKUnOu4LB3%2FCbASLJIqLn81Rgpjs11&submit=Submit&subject=E%3Dsteve
    %40redhat.com%2C+CN%3Dsteve%2C+UID%3Dsteve%2C+OU%3Dsteve%2C+O%3Dsteve%2C+C
    %3DUS&requestFormat=keygen&certType=client
    ```

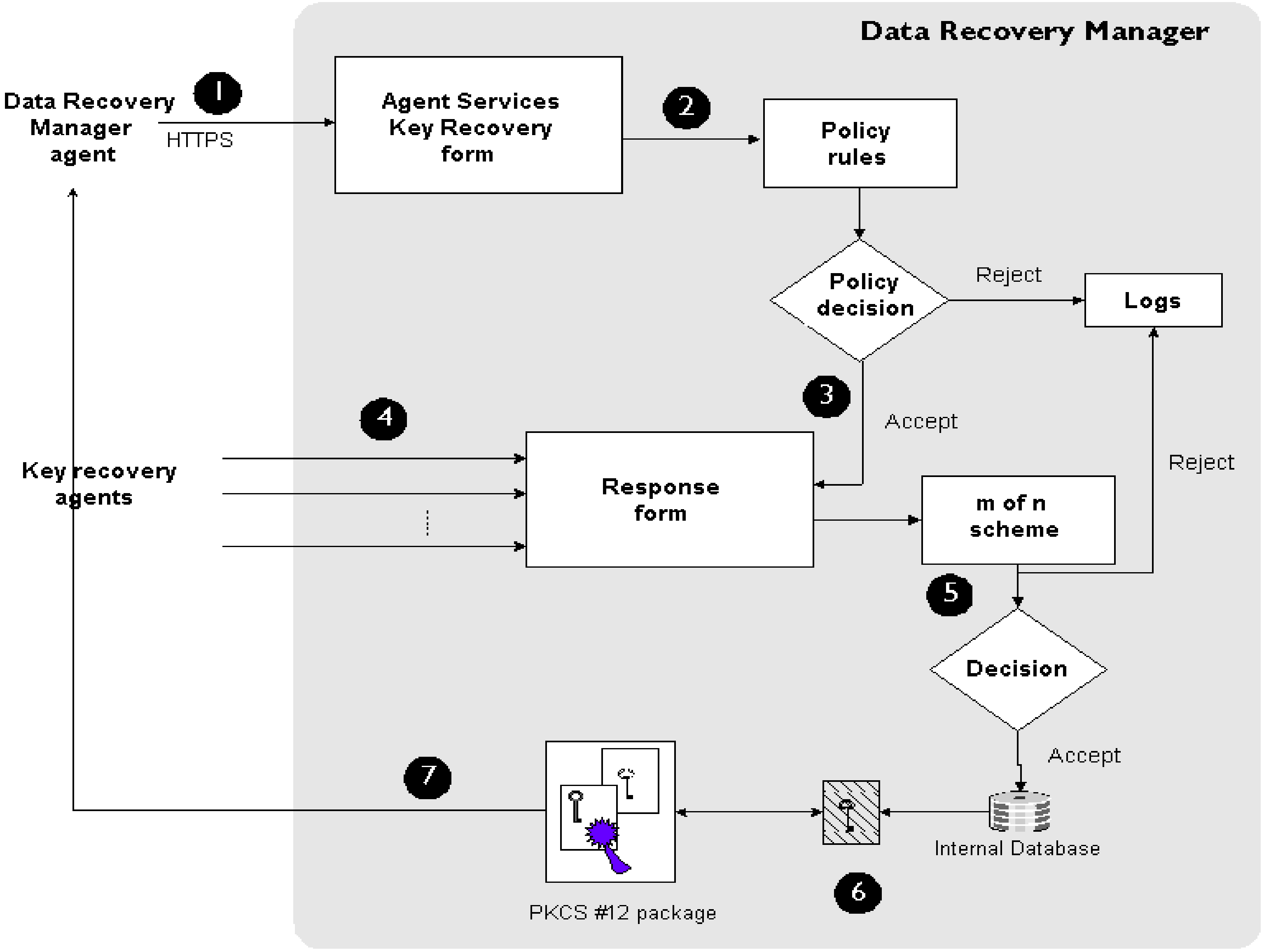
Backup Slides

Key Archival

DRM: Key Archival



DRM: Key Recovery



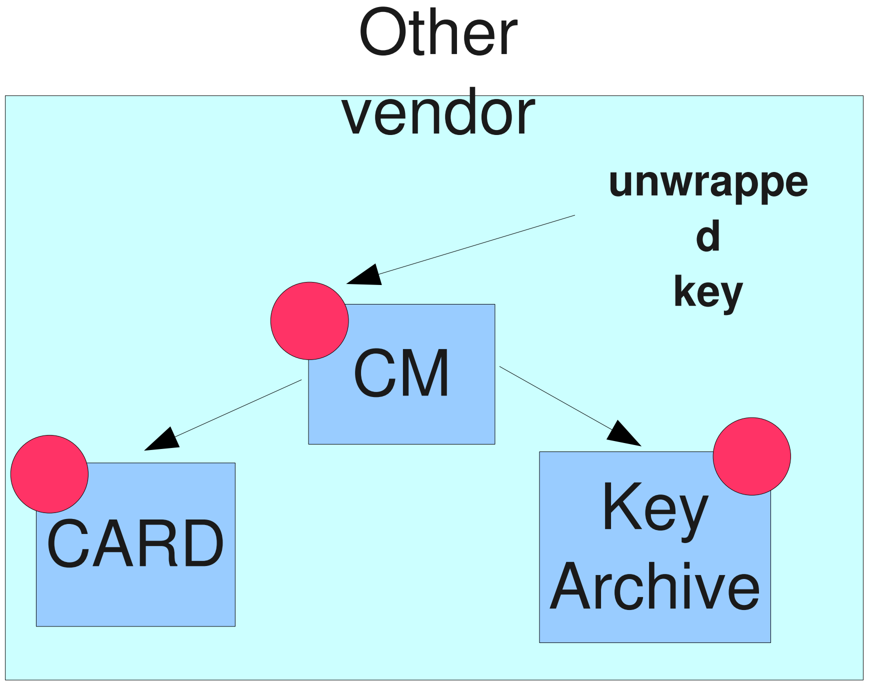
Key Recovery

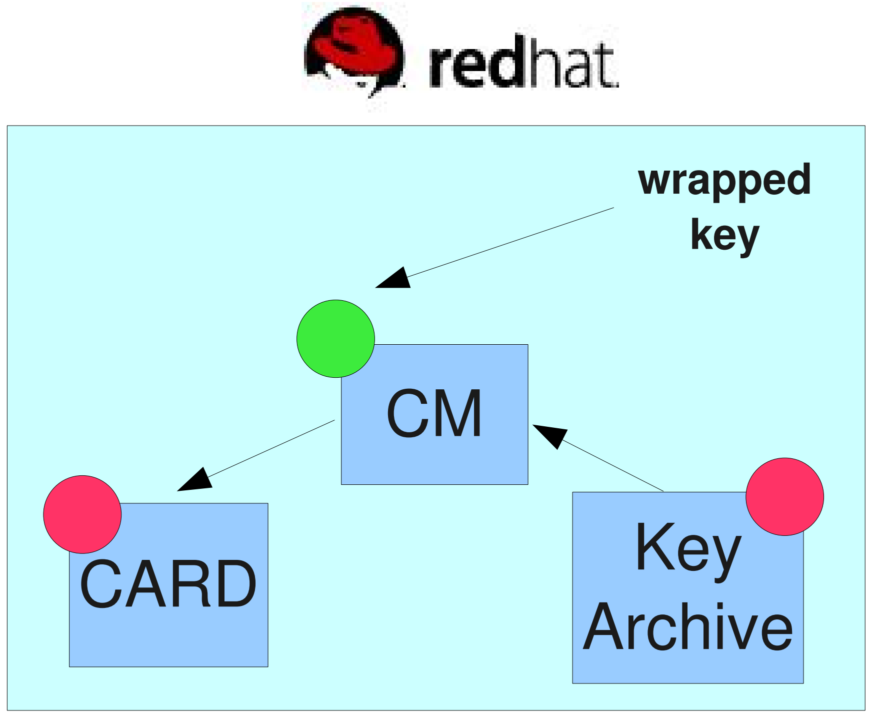
- Human Factors
- Types of lost token
  - Left token at home
  - Stolen
  - Don't know where it is
- Goal:
  - Back up keys automatically for users
  - Keys never appear on disk, always on the card
    - More secure
    - Less complex for user

Constraints

- To provide a backup of user's private key, a copy must be made
- Once a private key is generated in the smartcard, it cannot be extracted
  - The applet does not provide an API to extract key data from the card
- Only option is to generate keys off-card
  - Then, archive the keys
  - Then, import the keys into the card

Comparison with other products





Key Archival Flow

- During Secure Channel handshake, TKS generates extra key:
  - Key Transport Key (KTK)
- KTK is wrapped with KEK key, for card
- KTK is wrapped with DRM Transport Key, for the DRM
  - TPS asks DRM to generate a key
  - DRM unwraps the wrapped KTK
  - Generate a new key pair, archive it
  - Wrap the resulting private key with the KTK
  - Return the public key and wrapped private key to the TPS
- TPS injects the wrapped private key onto the card

Key Recovery Scenarios

- Signing keys:
  - Always get a new signing key/cert
- Encryption keys
  - Choice:
    - recover old key/cert, don't revoke
    - recover old key/cert, revoke AND get new key/cert
- Card capacity:
  - 32k Card can support at least 3 keys/certs
  - We compress some management data on the card
  - Exploring other schemes to support more keys

Card Loss

- TPS can enforce “1 active card per user” policy.
- TPS tracks cards assigned to users, and their status
- Administrator must change status of token to allow re-enrollment:
  - Destroyed -> implies revocation not necessary
  - Lost -> implies unknown status, revocation necessary
  - Temporarily lost -> implies left at home
- Temporary loss:
  - Can be issued a new type of card for that day,
  - e.g. ID / signing-only, short-lived certificate
  - old certs put 'on hold'
  - when original card is found, certs are taken off-hold

Backup - NSS

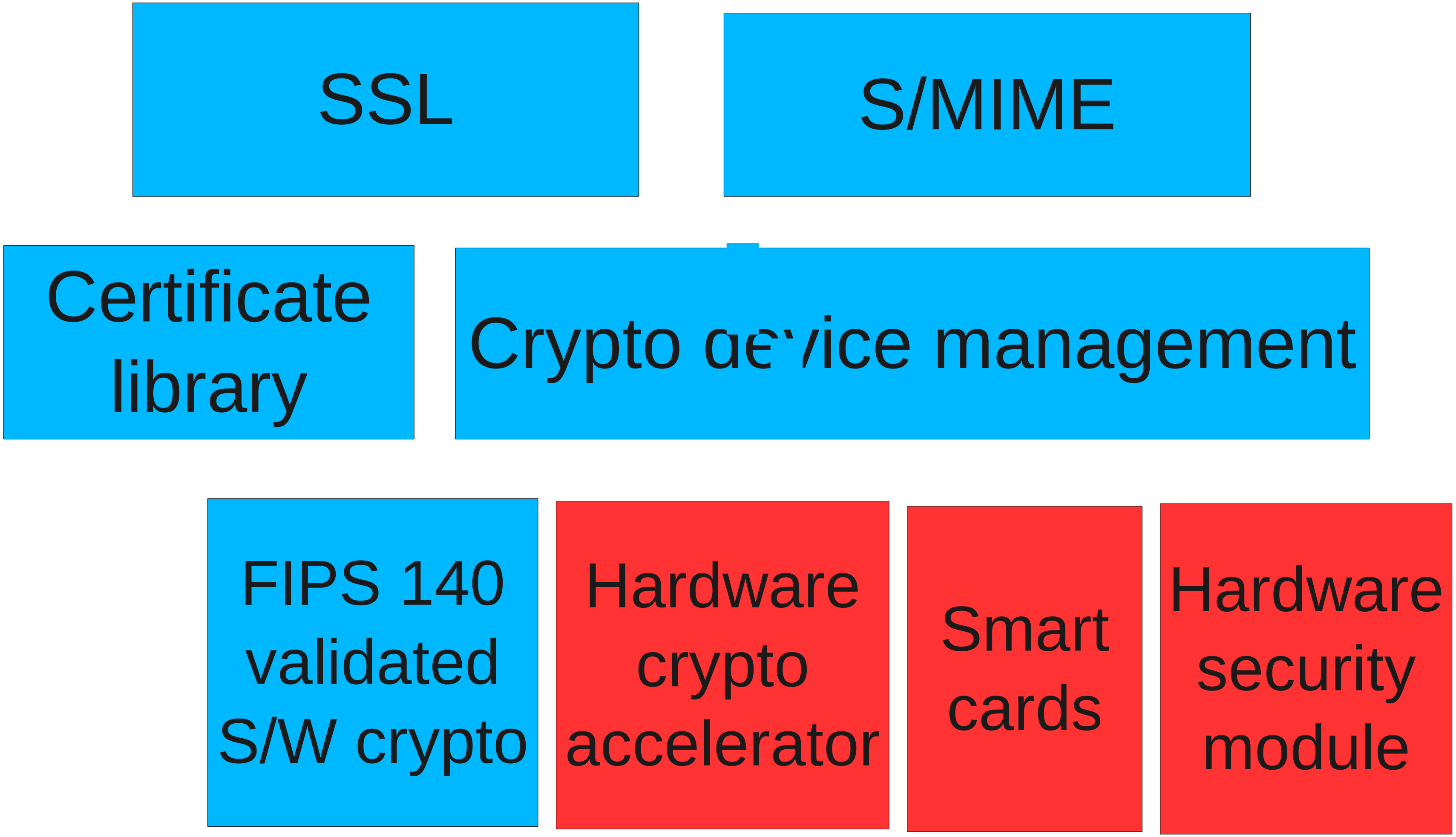
What is NSS?

- The crypto and PKI implementations in Red Hat Certificate System and many other products
- Open source C libraries
  - Can be used by C and C++ applications
  - Can be used by client and server applications
  - Cross-platform: ported to 20+ platforms
- Support major crypto algorithms and security standards
- Support smartcards and other hardware crypto devices
- JSS: open source Java bindings for NSS
  - Gives Java programs access to NSS via JNI

Open Source Licenses

- NSS can be used free of charge under the Mozilla Public License (MPL), GNU General Public License (GPL), or GNU Lesser General Public License (LGPL).
- The triple license allows both open source and closed source applications to use NSS.
    - MPL: recommended for most scenarios
    - GPL: for use by GPL open source products
    - LGPL: for copying NSS code into LGPL open source products
- Most common scenario: you use NSS without modifications under MPL. The only obligation is to include a notice in your product or documentation:
    - This product contains the NSS libraries. The source code for NSS is available under the Mozilla Public License at http://www.mozilla.org/.

NSS Architecture Diagram



SSL

S/MIME

Certificate library

Crypto device management

FIPS 140 validated S/W crypto

Hardware crypto accelerator

Smart cards

Hardware security module

Features: NSS Crypto Library

- Software crypto module
- FIPS 140 Level 2 validated
- Public key: RSA (PKCS #1 v1.5), DSA, Diffie-Hellman
- Symmetric key: DES, Triple DES, AES, RC2, RC4
- Hash: SHA-1, SHA-256, SHA-384, SHA-512, MD2, MD5
- Keyed hash: HMAC
- FIPS 186 pseudorandom number generator (PRNG)
- Storage of certificates and keys

Features: NSS Base Library

- ASN.1 encoder and decoder
- X.509 v3 certificates
  - Certificate path validation
  - Certificate revocation checking: CRLs and OCSP
- PKCS #11 device support
  - Smartcards
  - Hardware crypto accelerators
  - Hardware security modules

Features: NSS High-Level Libraries

- S/MIME (CMS): encrypted and signed email
  - PKCS #7 certificate chains
  - PKCS #12 private key backup
- SSL and TLS
- XML Digital Signature and Encryption: with the xmlsec library