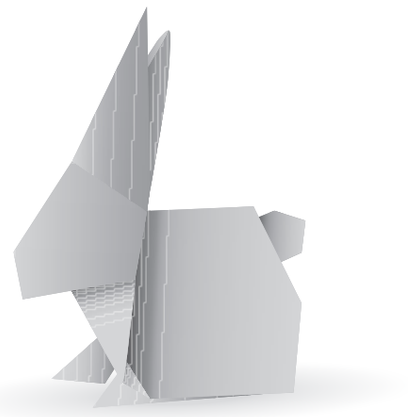
OpenXPKI

Workshop 2018

Martin Bartosch



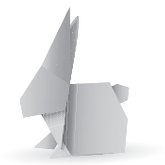
WhiteRabbitSecurity

PKI Goals Recap

WhiteRabbitSecurity

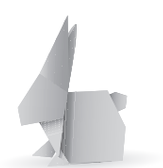
Public Key Cryptography Benefits and Challenges

- ☑ Can provide authenticity, integrity and confidentiality
- Requires reliable association of subject and key
- Requires key distribution
- Requires key life cycle management



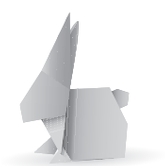
PKI:
Making Public Keys Manageable

- Associate keys with their holders
- Distribute public keys
- Manage key life cycle

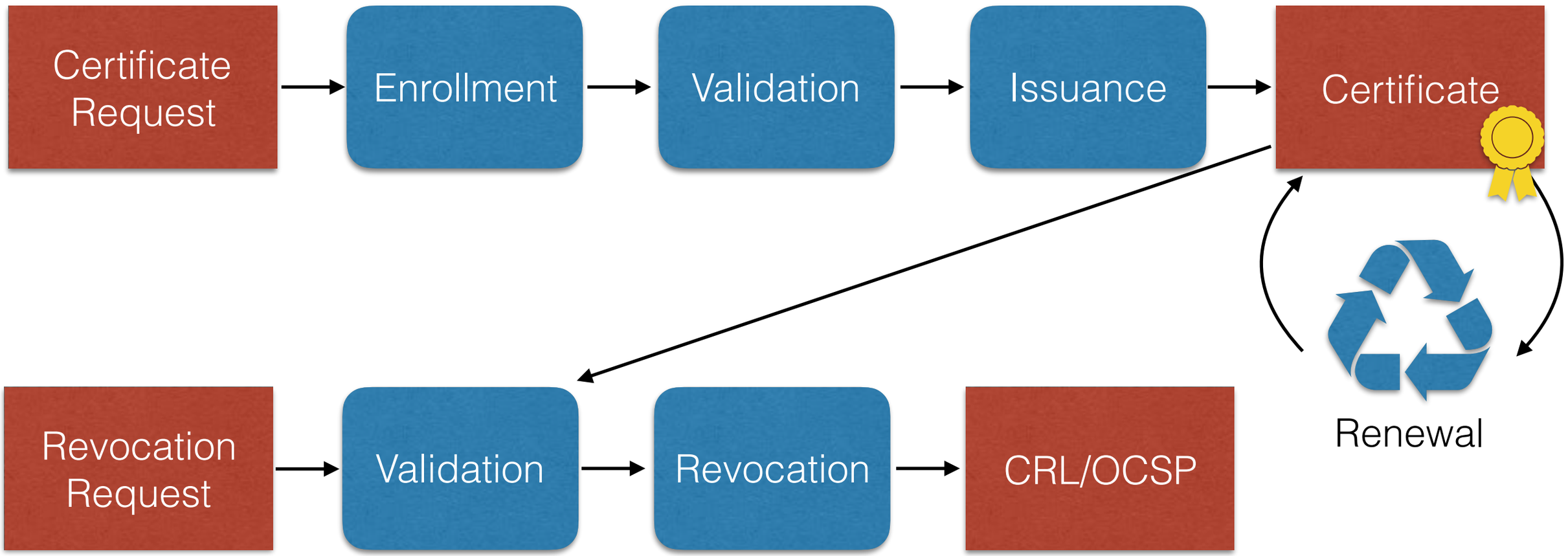


Digital Certificates: Building Blocks of PKI

- Digitally signed data structure
- *Binds* public key to an entity (person, system)
- *Chains* trust back *to a trusted authority*
- Provide *reliable naming* to identify entity
- *Limits* public key *validity*
- *Limits* public key *usage*



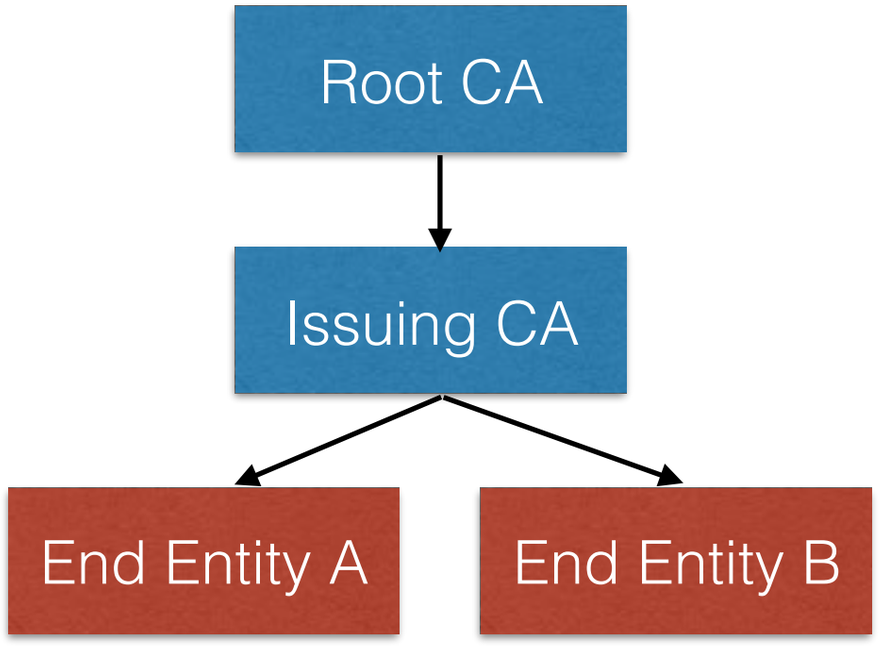
Digital Certificates Lifecycle

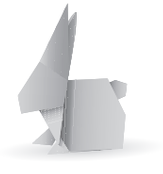


WhiteRabbitSecurity

PKI Hierarchies
Certificate Chaining

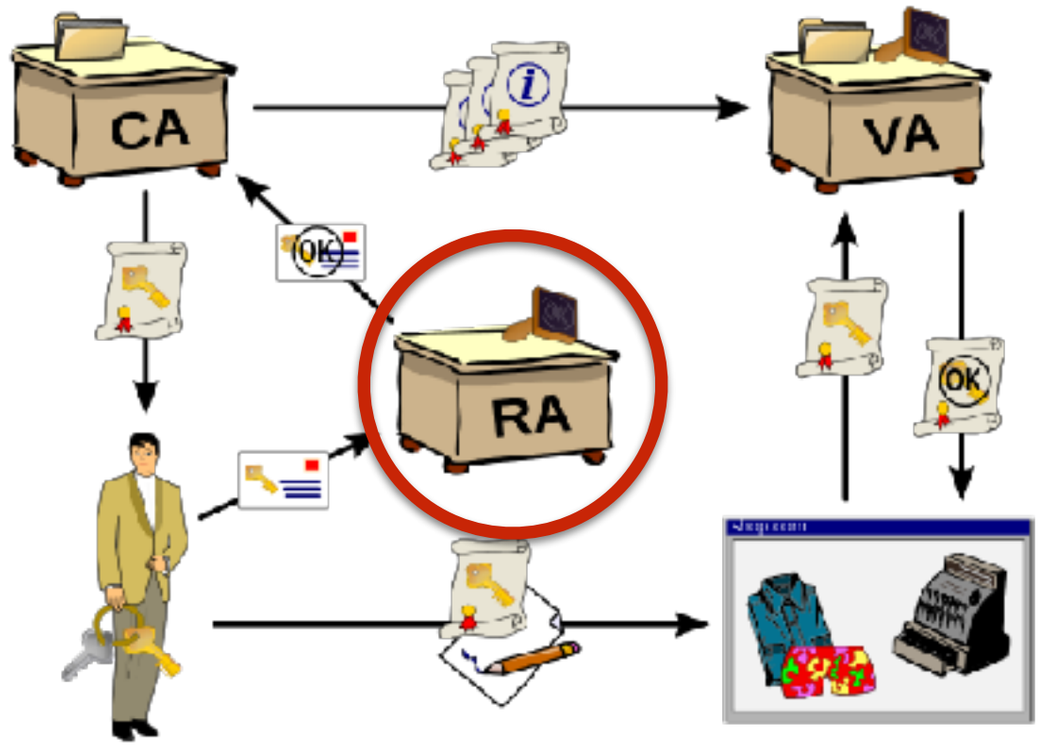
- Allows creation of Trust Hierarchies
- Separation of Root CA (Trust Anchor) and Issuing CA
  - Low frequency of Root CA key operations
  - Higher security level possible (offline)
  - Subordinate Issuing CA responsible for issuance of End Entity Certificates





Traditional PKI components: RA - Registration Authority

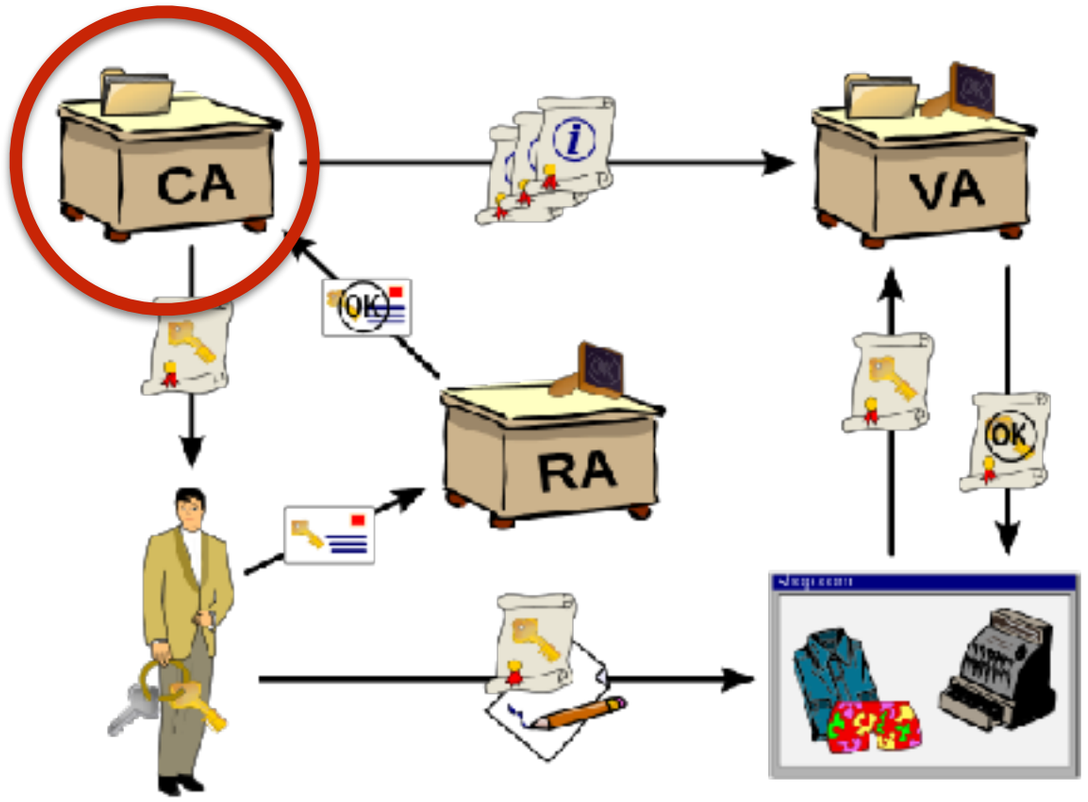
- Customer facing online component
- Receives enrollment/revocation requests
- Performs verification/validation on requests
  - Authentication of applicant
  - Validation of request data
  - Request authorization
- Forwards approved requests to CA



Source: Wikipedia

Traditional PKI components: CA - Certificate Authority

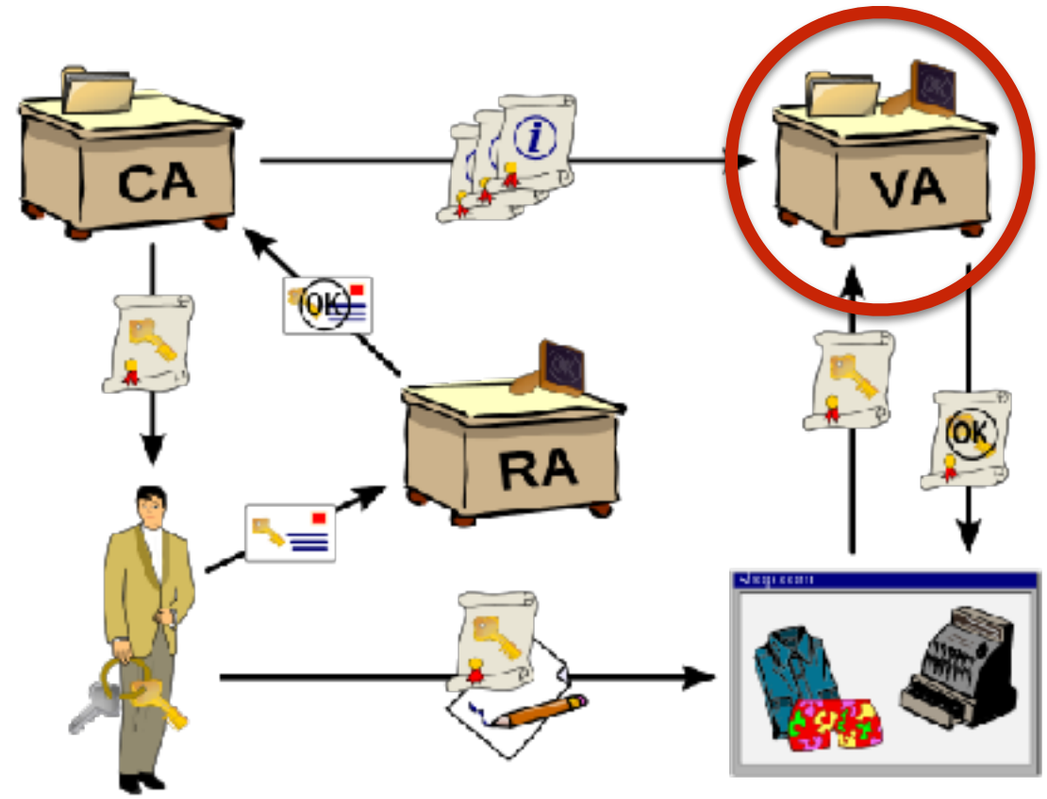
- Environment primarily dedicated to certificate issuance and revocation
- Often segregated via Firewalls, network segmentation or operated offline
- Receives approved requests from RA
- Performs requested infrastructure key activity (issuance, revocation, CRL issuance)



Source: Wikipedia

Traditional PKI components: VA - Validation Authority

- Services for certificate validity verification
- CRL hosting
- OCSP operation



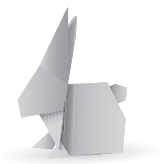
Source: Wikipedia

PKI Integration

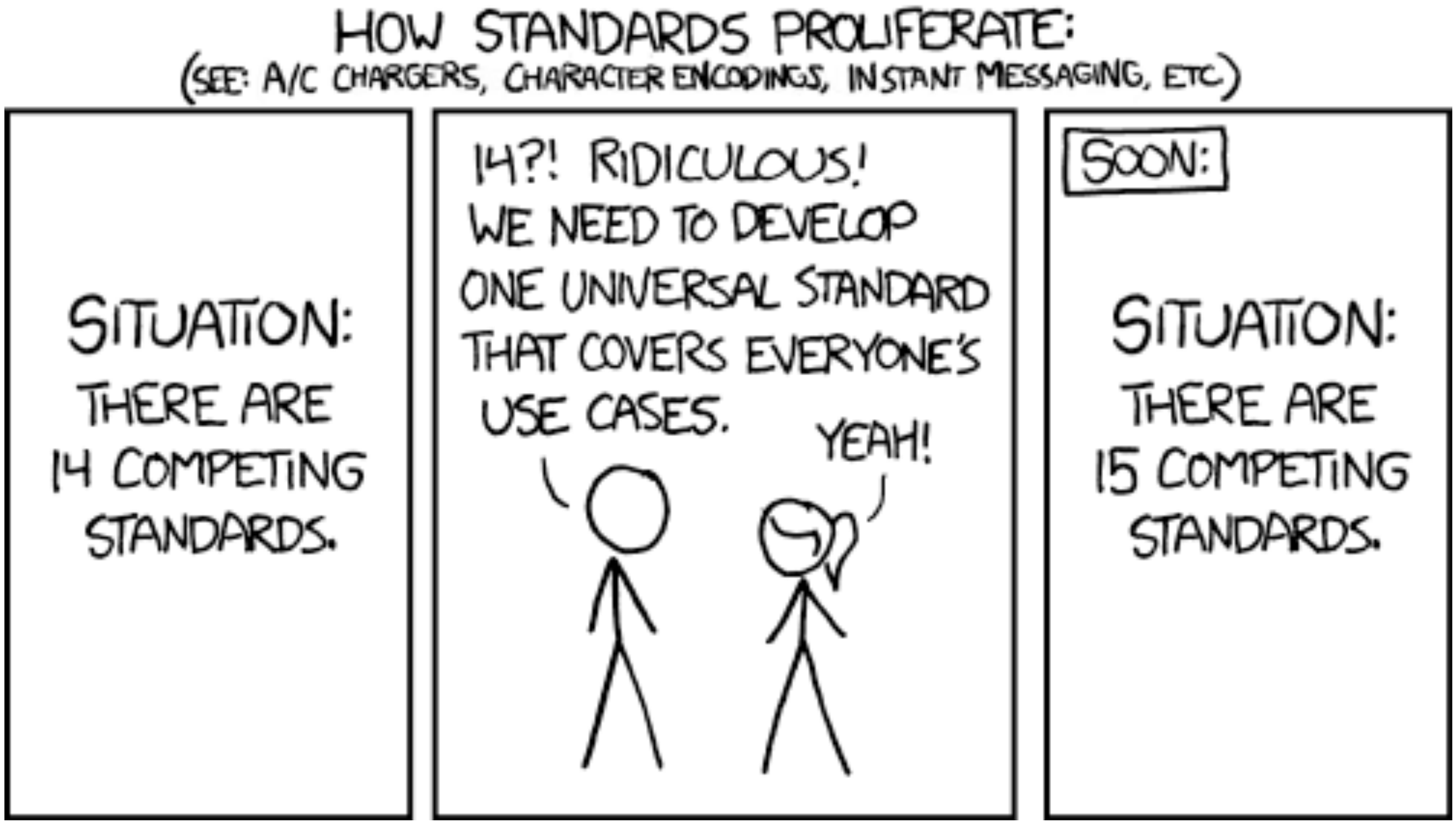
- Manual enrollment, approval and issuance causes effort and is error-prone

- Automation use cases
  - Initial enrollment
  - Renewal
  - Validity checking

➡ PKI Interface required



PKI Interface Standards



Source: xkcd.com (https://xkcd.com/927/)

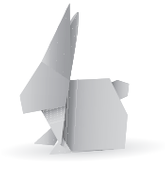
PKI Interface Standards

	Data format	Transport	Use cases	RL relevance
CMC	CMS, PKCS#10 ASN.1	n/a	certificate lifecycle	lacking transport protocol def.
SCEP	PKCS#7, PKCS#10 ASN.1	HTTP	enroll, renew	high
EST	CMC	HTTPS	enroll, renew	growing
MS Autoenrollment	CMC	DCOM	enroll, renew	Microsoft: high others: zero
CMP	CMP ASN.1	HTTP (…)	certificate lifecycle	low, academical
XKMS	XML	HTTP (…)	certificate lifecycle	low
KMIP	KMIP TTLV (XML, JSON)	TLS, HTTPS (…)	key management	situational, „product glue“
SCVP	SCVP ASN.1	HTTP (…)	validation (policy)	low (zero?)
OCSP	OCSP ASN.1	HTTP	real time validation	situational

OpenXPKI

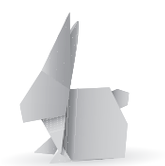
Design Principles and Features

WhiteRabbitSecurity

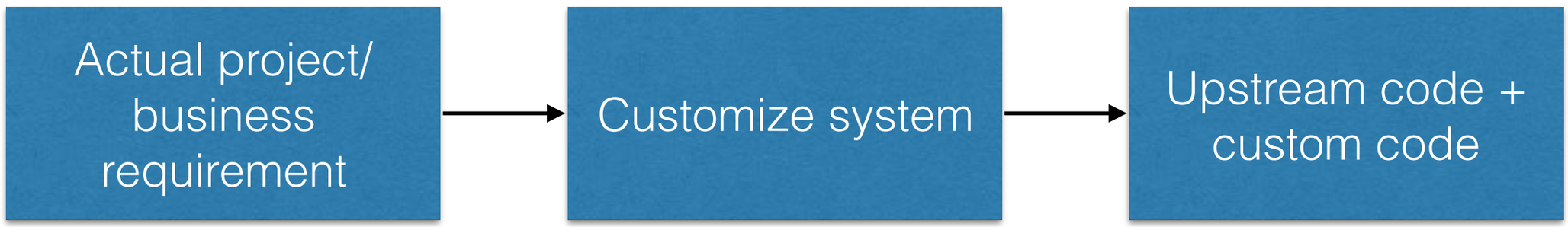


OpenXPKI Goals

- Primary use case: Online Issuing CA
- Strong focus on
  - RA functionality
  - automation support
  - flexibility
  - infrastructure integration

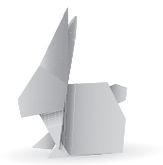


Standard project methodology

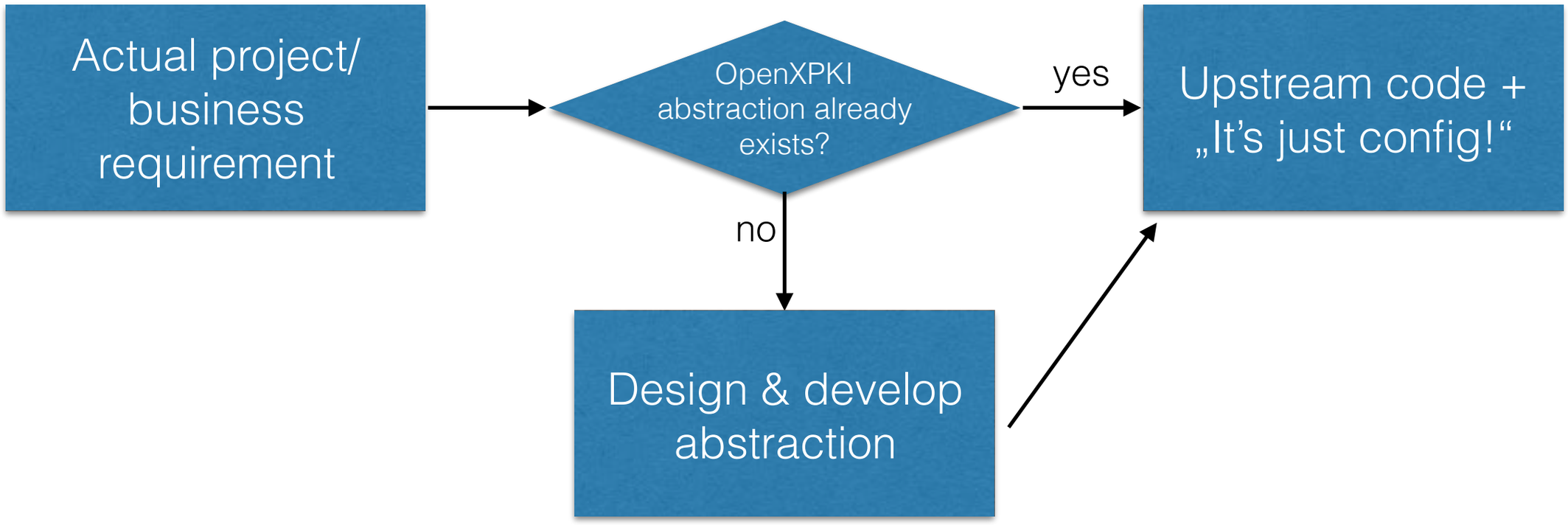


Customer specific code customization (e. g. local Workflow activities).

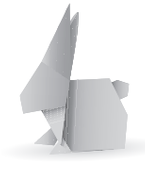
Technical debt: maintain customizations.



OpenXPKI Dev Team approach: Customer requirement driven architecture

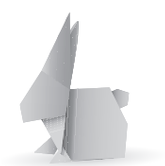


Take two steps back, abstract problem, implement abstraction, configure system. Maintain configuration.



OpenXPKI abstractions: Results from 13 years of development

- Workflow engine and generic workflows with „hooks“ for customization (implement business logic)
- Workflow „datapool“ (tuple storage, data encryption)
- Generic, adaptive user interface (implicit via Workflow definition)
- Connector (abstract external data sources/sinks)
- Template::Toolkit support (propagate/reuse data)
- RPC interface (expose Workflows)
- NICE interface (chain CA backends)



OpenXPKI Core Properties

- Multitenancy:
  Support multiple logical CAs in one CA instance („PKI Realms“)

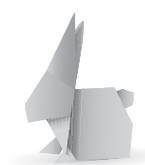
- Perpetual PKI operation:
  Support fully automatic CA rollover within a PKI Realm

- Extensive support of automatic enrollment interfaces

- Crypto backend abstraction

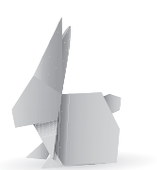
- HSM support

- RDBMs support



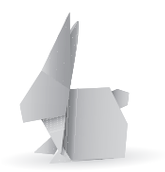
OpenXPKI Core Properties (2)

- Powerful RA features: highly flexible request authentication/authorization
- Certificate lifecycle management
  - Notifications
  - Renewal
  - Inventory and reporting



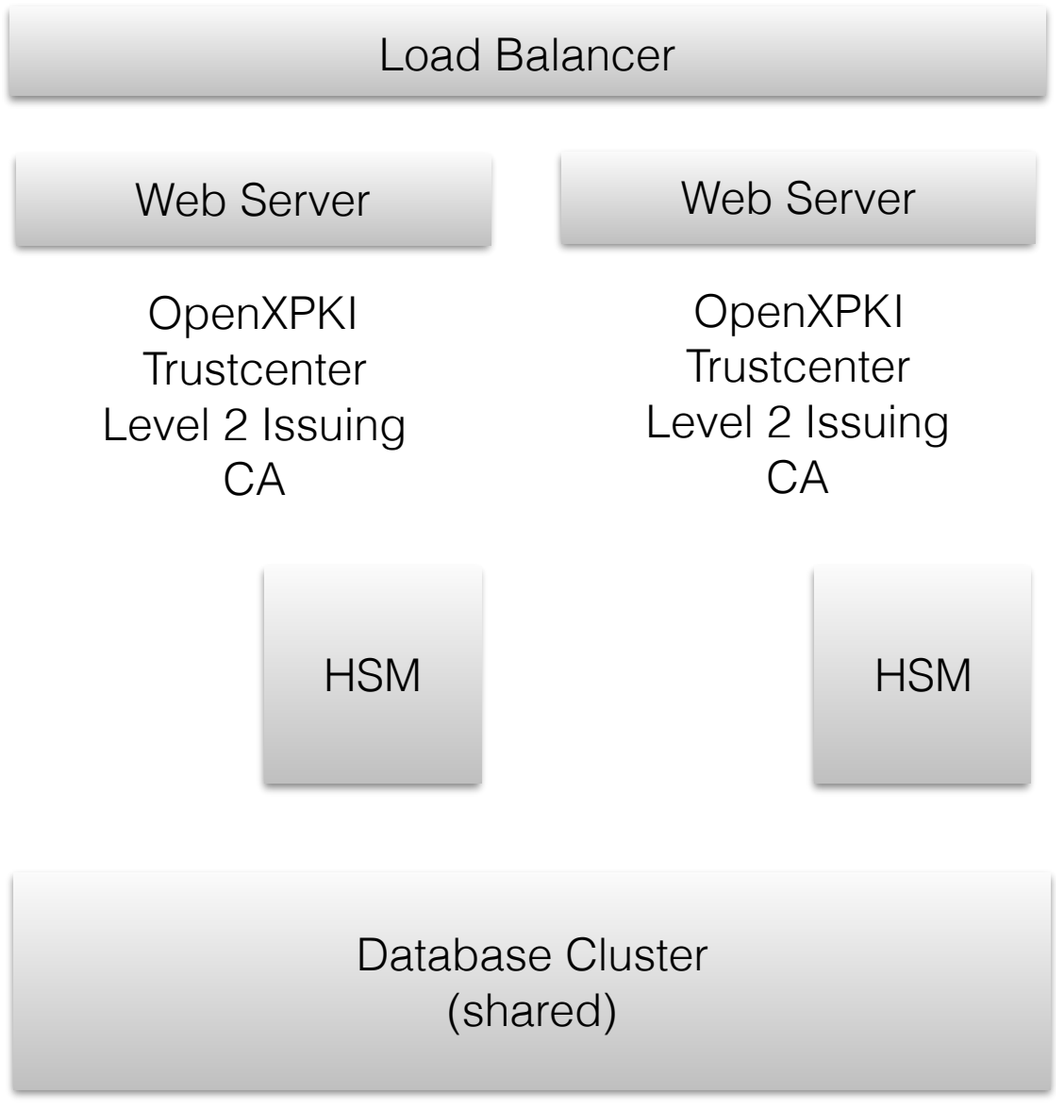
OpenXPKI Core Properties (3) Configuration

- Text-based, hierarchical configuration
- OpenXPKI configuration layer uses an abstraction for arbitrary data sources/sinks („Connector“)
  - Connectors can be used *anywhere*, for *any* configuration item
  - Replace literal configuration value with a reference to a Connector returning the value
  - OpenXPKI PKI Realm and workflow configuration extensively supports templating (via Template::Toolkit)



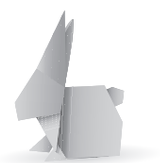
Horizontal Scaling: Active/active Clustering

- Multiple OpenPKI nodes with identical configuration
- Shared, redundant database required (for redundant A/A cluster)
- Load balancer required (scheduling algorithm should favor the same node a client has been accessing previously)



Vertical Segregation: Chaining PKI components

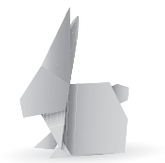
- Traditional PKI architecture: RA - CA - VA
- OpenXPKI is RA & CA in one application
- Functionality can be separated:
  - OpenXPKI can act as a RA
  - Delegate actual CA operations to „Backend CA“ (online, offline)
- OpenXPKI can integrate with external asset databases, change tools or workflow systems



OpenXPKI concepts:

PKI Realms

WhiteRabbitSecurity



PKI Realms: CA Multitenancy

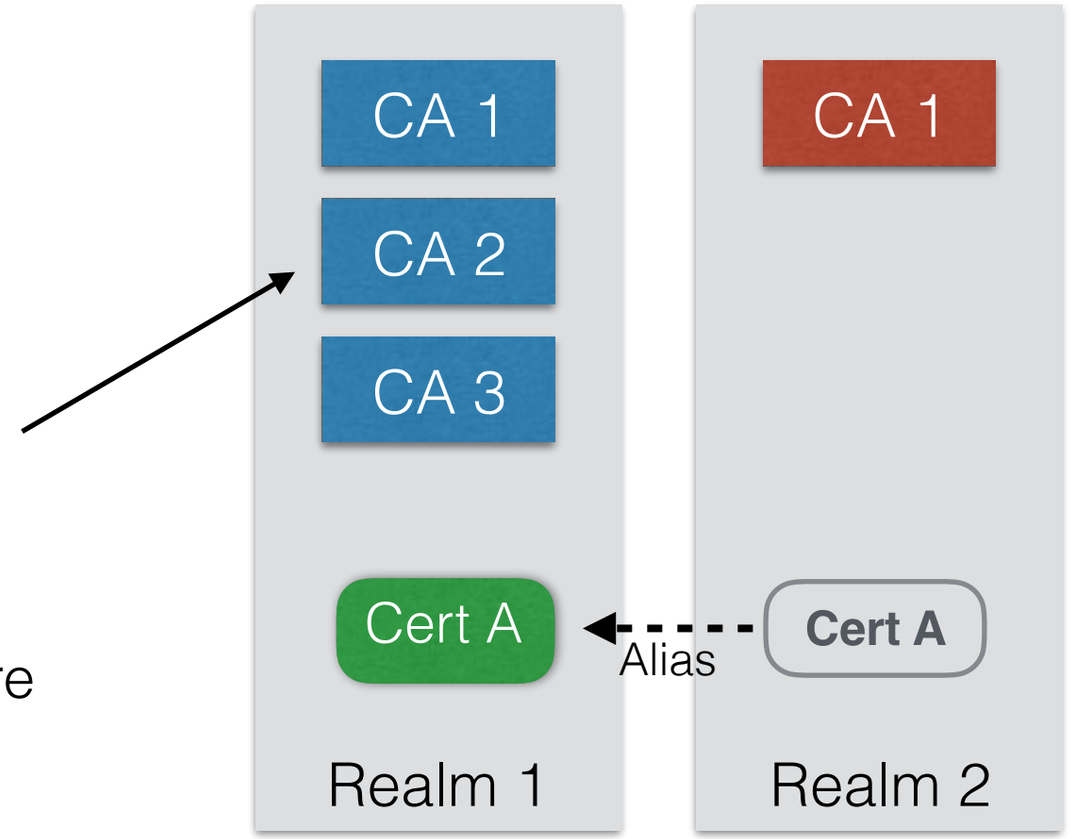
- PKI Realms: OpenXPKI supports running multiple, separated logical CAs in one single instance (same database)

- Can be used for multitenancy or running CAs for distinct purposes on the same OpenXPKI instance

Logical CAs

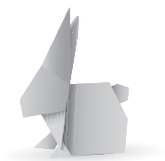
PKI Realms: Logical CAs

- PKI Realm: Logical CA (namespace):
  - common set of policies, certificate profiles and enrollment/issuance/ revocation workflows
    - provides common namespace for all end entity certificates
- PKI Realm contains arbitrary number of Issuing CAs
- Certificates in different PKI Realms are only visible inside their own Realm
- Certificates can be linked from one PKI Realm to another realm



Shared configuration

- All OpenXPKI PKI Realms share
  - Database configuration
  - Daemon configuration
  - Watchdog configuration



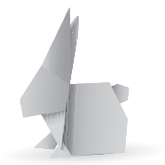
PKI Realms:
Realm specific configuration

- Interfaces
  - Authentication
  - APIs (SCEP, EST, RPC, SOAP)
  - User Interface hints
- CA processing
  - Certificate profiles
  - Crypto token/crypto backend
- Workflows
- Artefact handling
  - Publishing (CRL, Certificates)
  - Notification
  - Metadata management
  - Reporting

OpenXPKI concepts:

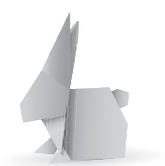
CA Rollover

WhiteRabbitSecurity



Problem: CAs expire

- All X.509-Zertifikate have a built-in expiry date, CAs included
- (Useful) purpose: control active usage period cryptographic keys
- Common approach: looong CA certifiate validity (2040, anyone…?)



Naive approach: Long lived CAs

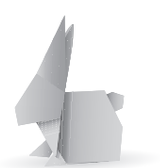
- The Problem: CAs *still* expire eventually
- When this is about to happen it is likely nobody will have a clue what to do
- No preparation, no plan

➡ Long CA validities only mask and postpone implicit problems

CA Expiration
The „sustainability“ approach

- Plan for perpetual operation of the PKI
- Make CA expiry a part of normal operation

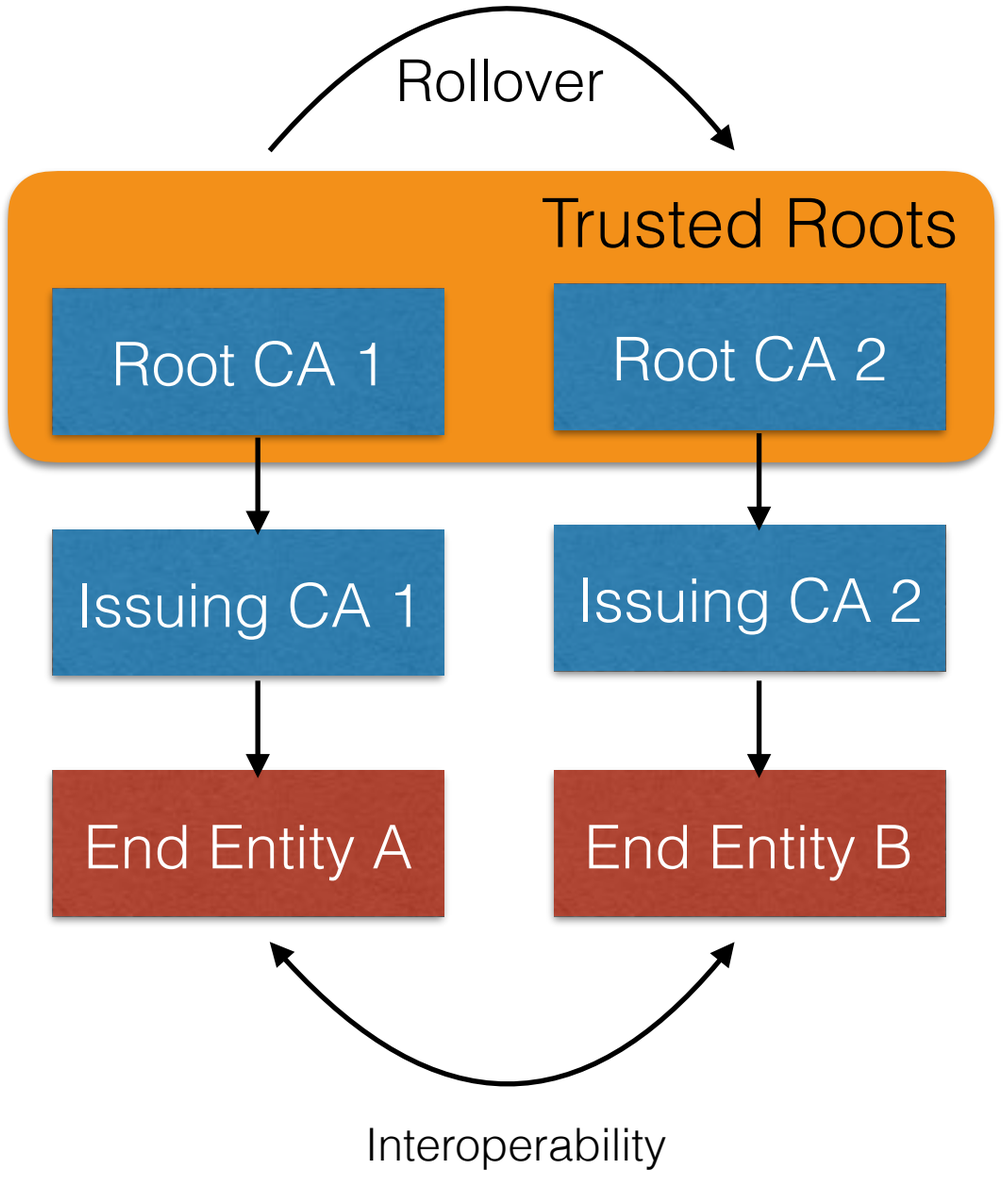
  ➡ CA Rollover



CA Rollover

Plan for perpetual operation of a PKI

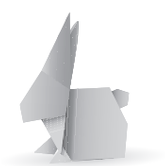
- Approach: a completely new CA generation assumes operation
- Identical configuration, identical certificate profiles
- Issuance of new certificates only with new hierarchy
- Existing certificates remain in operation



WhiteRabbitSecurity

Planning for CA Rollover

- Make CA validity long enough to support designated operation purposes, e. g.
  - End Entities validity: 1 - 3 years
  - CAs validity: ~ 6 - 12 years
    (rule of thumb: EE validity * 2 + 1 year)
- Purposefully group CA Rollover events
  - Vertically align CA validities
  - Rollover happens on all levels simultaneously



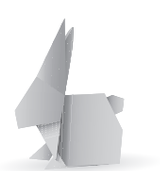
Planning for CA Rollover (cont)

- Distribute new Root CA long before performing CA Rollover

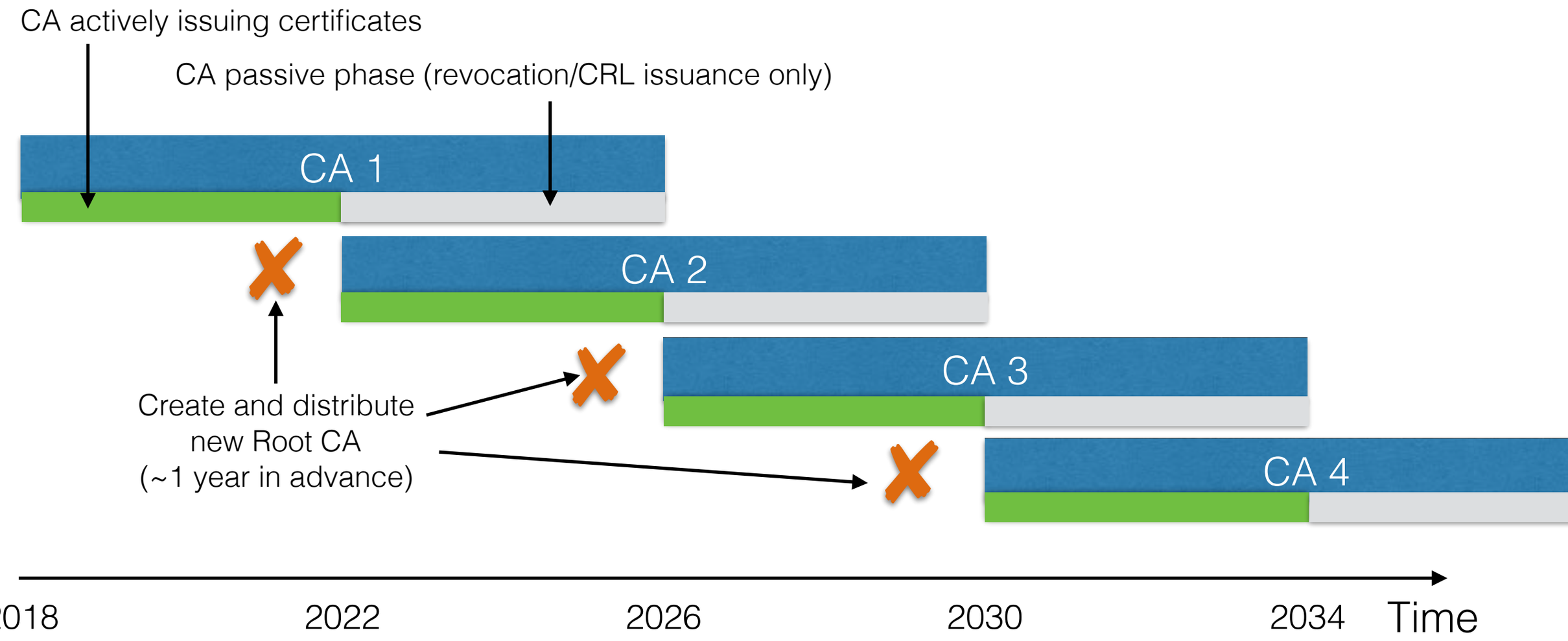
- Actively verify new Root CA „adoption“

- Practice CA Rollover:

  ➡ Reduce validity of your „Test CA“, forcing Test Certificate users to perform CA Rollover more often on their test systems

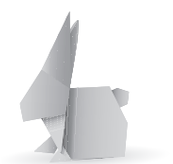


CA Rollover (cont.)



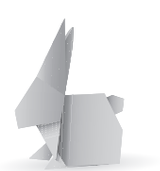
CA Rollover with OpenXPKI

- CA Rollover is a core feature OpenXPKI
- Can work fully automatically within a PKI Realm
- Can be administratively postponed or triggered
- Seamless, no system downtime
- CA Rollover criterium
  - Highest NotBefore date of all valid CA Certificate
  - Or optional administrative Rollover Date



CA Rollover: Preparation

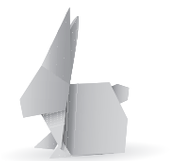
- Administratively *import* the new CA certificate
  - Possible at any time before the actual rollover
  - System keeps running, no interruption
  - Importing new CA certificate is possible without configuration modification



CA Rollover Administration

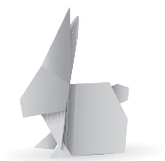
```
$ openxpkiadm certificate import  --file ca-one-signer-1.crt \
  --realm ca-one --token certsign
```

```
$ openxpkiadm alias --realm ca-one
=== functional token ===
ca-one-signer (certsign):
  Alias     : ca-one-signer-1
  Identifier: _2qf_ZTfJgXgh4xWm3WYr0IgYN8
  NotBefore : 2018-02-07 13:21:52
  NotAfter  : 2023-02-09 13:21:52
```

CA Rollover: Execution

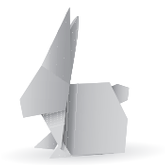
- For any certificate issuance operation OpenXPKI automatically
  - determines all valid CAs capable of issuing certificates for the PKI Realm
  - picks the CA certificate with the highest NotBefore date
  - issues the certificate with this CA certificate
- All other CA certificates within the PKI Realm remain valid, but are only used to issue CRLs during their remaining lifetime
- OpenXPKI correctly builds certificate chains for enrollment interfaces



OpenXPKI concepts:

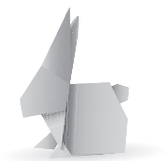
Interfaces

WhiteRabbitSecurity



OpenXPKI interfaces
Support of commonly used standard enrollment protocols

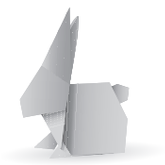
- SCEP
- EST
- Why no other interfaces?
  Nobody asked for them yet…



OpenXPKI interfaces
Non-standard interfaces: RPC

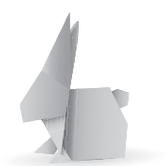
- Generic RPC and SOAP interface

  ➡ can be used to expose OpenXPKI workflows



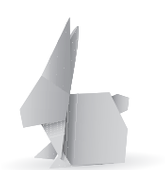
Enrollment interfaces
Native Windows autoenrollment

- Domain-joined Windows clients automatically enroll against a Windows CA (AD Certificate Services)
- Proprietary transport protocol (DCOM), hence…
- … no native support in OpenXPKI…
- … but possible via Secardeo CertEP (uses RPC enrollment interface)



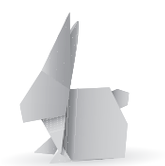
Enrollment interfaces Features

- All enrollment interfaces include built-in hooks for „eligibility checks“
  - Request authentication
  - Request authorization (e. g. check against asset database/CMDB)
- Enrollment interfaces support authentication based on existing certificate



OpenXPKI
Non-standard interfaces: NICE

- NICE Interface
  - CA operation abstraction layer
    - support synchronous and asychronous operation
    - support local and remote CAs
    - Possible use case:
      delegate certificate issuance/revocation to
      remote CA (e. g. SwissSign, Comodo…)

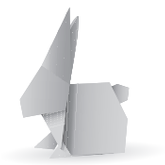


Validation: CRLs

- CRLs can be published to local and remote targets
  - local file system
  - remote systems (e. g. scp, HTTP/REST)
  - LDAP
- arbitrary number of publishing endpoints

Validation: OCSP

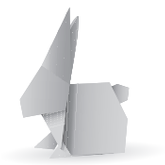
- OCSP support not integrated in OpenXPKI CE
- 3rd party OCSP responders can be integrated
- High performance OCSP responder support available with OpenXPKI Enterprise Edition



OpenXPKI

Enterprise Edition

WhiteRabbitSecurity



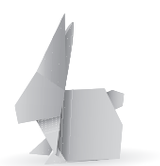
OpenXPKI Code Base

Community Edition:
https://github.com/openxpki/openxpki

Primarily maintained by White Rabbit Security Dev Team
Packages for Debian

OpenXPKI Enterprise Edition :=

Community Edition code base
+ Packaging for other OS (RHEL, SLES)
+ Professional Services
+ Advanced Features & Workflows



White Rabbit Security
OpenXPKI Professional Services

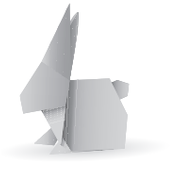
Operation Support

- Privileged support channels
- Software maintenance
  - „Care & Feeding“ of OpenXPKI installation
  - Software updates
- PKI administration support

Customer Projects

- Business analysis
- Architecture & design
- Configuration and workflow customization
- Custom development
- Configuration management
- Infrastructure integration

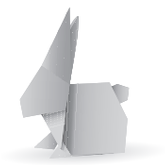
WhiteRabbitSecurity



White Rabbit Security
OpenXPKI Advanced Features

- Advanced Workflows
- Improved statistics and reporting engine
- External CA Backends
- VA Module
  - Improved CRL processing
  - High Performance OCSP Responder

WhiteRabbitSecurity



White Rabbit Security Workshops & Training

OpenXPKI

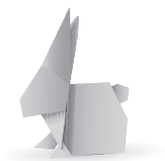
- OpenXPKI Registration Officer
- OpenXPKI System Administrator
- OpenXPKI Developer

WhiteRabbitSecurity

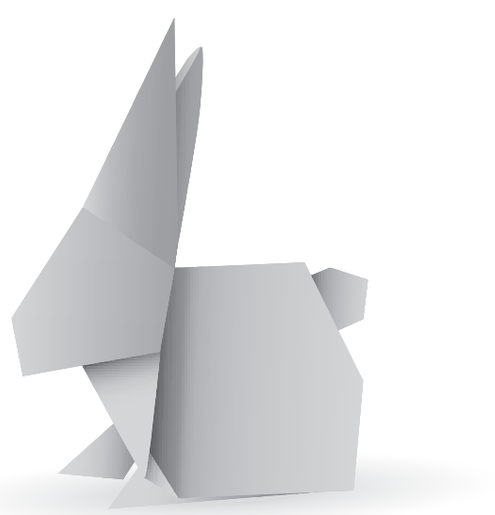
White Rabbit Security Consulting

- Business and requirements analysis
- Security architecture and systems design
- Cryptographic key management solutions
- PKI architecture and implementation

WhiteRabbitSecurity



“We pull a rabbit
out of a hat - and show you
how the trick works”



WhiteRabbitSecurity