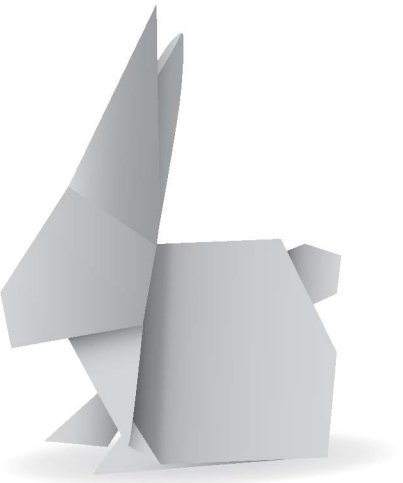
A tale of
daemons, wizards
and a magic carpet



Oliver Welter, 2015/10

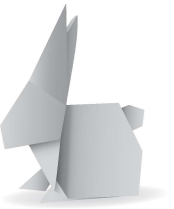
WhiteRabbitSecurity

Agenda

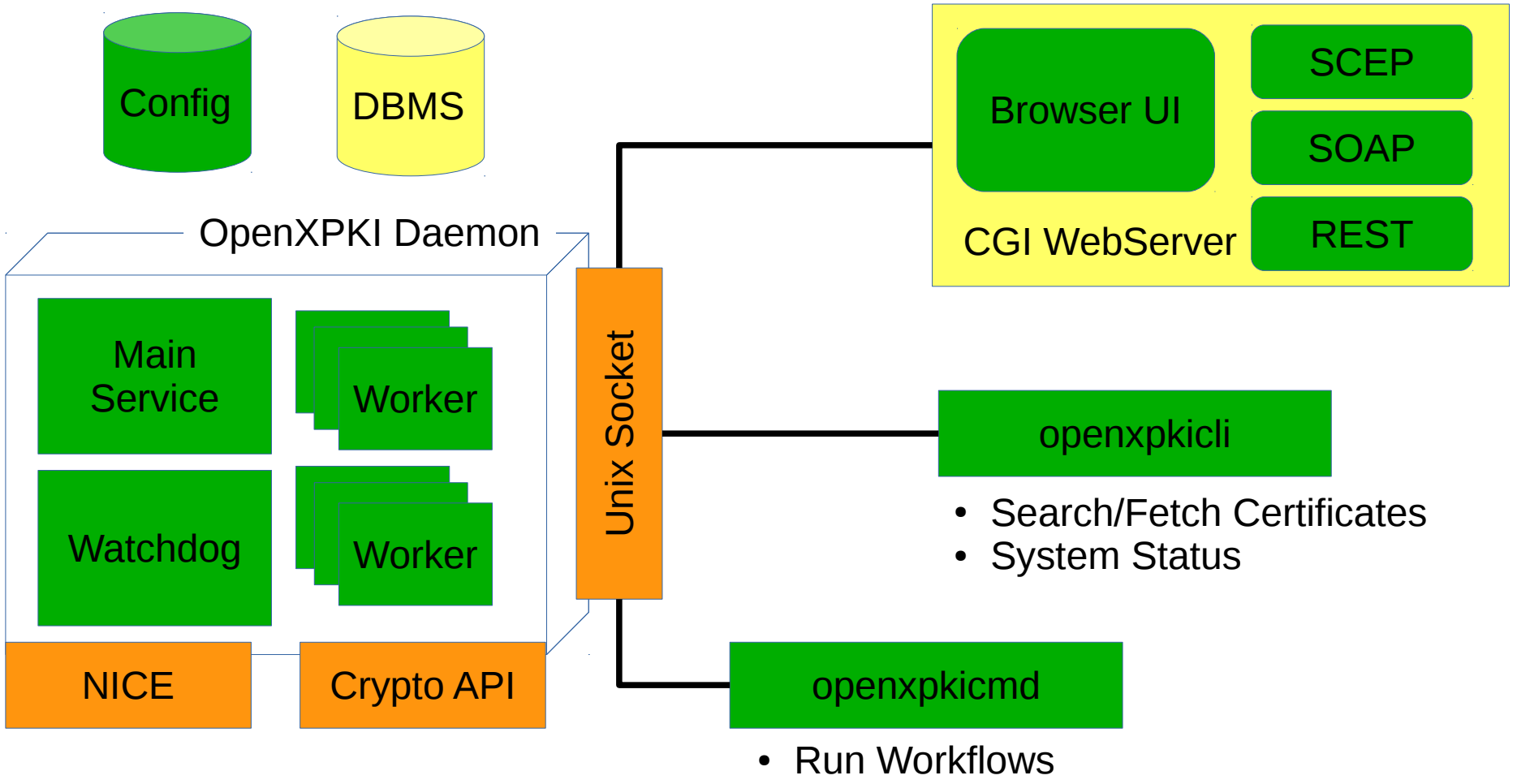
- OpenXPKI Core Architecture
- Client API and Interfaces
- NICE Backend API

- Wizard-Like User Interface
- Workflow Engine – OpenXPKI's magic carpet

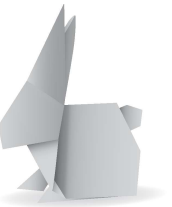
WhiteRabbitSecurity



OpenXPKI Core Architecture

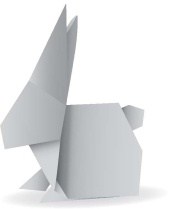


WhiteRabbitSecurity



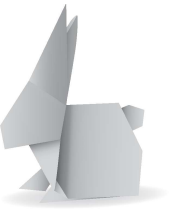
Client API and Interfaces / SCEP

- „Industry Standard“, invented by Cisco
- anonymous initial enrollment
- initial enrollment with password, OnBehalf certificate
- signed renewal
- support for revocation information (GetCRL)
- support for root CA rollover (GetNextCA)



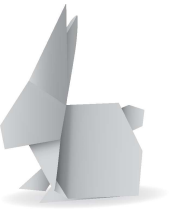
Client API and Interfaces / SOAP

- automated revocation of certificates
- anonymous
- authenticated via HTTPS with client certificate
- revocation by certificate identifier or IssuerSerial
- easily extensible (e.g. revoke by Smartcard ID)
- planned: request certificates via SOAP



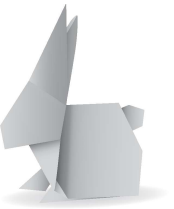
Client API and Interfaces / REST

- manage certificates using REST over HTTPS
- fetch certificate details using GET
- request/renew certificate using POST/PUT
- revoke certificate using DELETE
- easy integration of full certificate lifecycle into 3rd party applications with few knowledge on „crypto things“



Client API and Interfaces / Browser UI

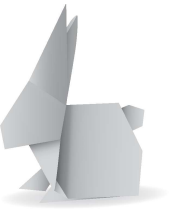
- works in any modern browser (based on Ember.js)
- login using different authentication backends
- download of entity and ca certificates, revocation lists
- search for certificates and workflows
- request / revoke certificates
- reports and statistics



N I C E

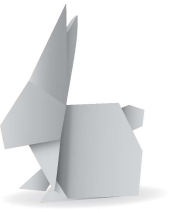
Nice **I**nterface for **C**ertificate **E**nrollment

WhiteRabbitSecurity



NICE Backend API

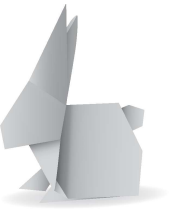
- abstraction layer for „atomic“ CA operations
- default backend uses local OpenSSL
- can delegate CA operations to a remote system
- 3rd party connectors (available on request)
  - SwissSign (server and user certificates)
  - Verisign (server certs only, NDA required)
  - PSW Group (reseller of several public CAs)



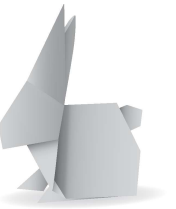
NICE Backend API

- why use OpenXPKI to talk to 3rd party CAs?
- accessible inside company network
- automation interfaces available (SCEP, SOAP)
- enforce process and policies
- easy monitoring of existing / expiring certificates

WhiteRabbitSecurity

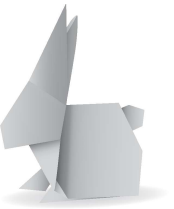


User Interface

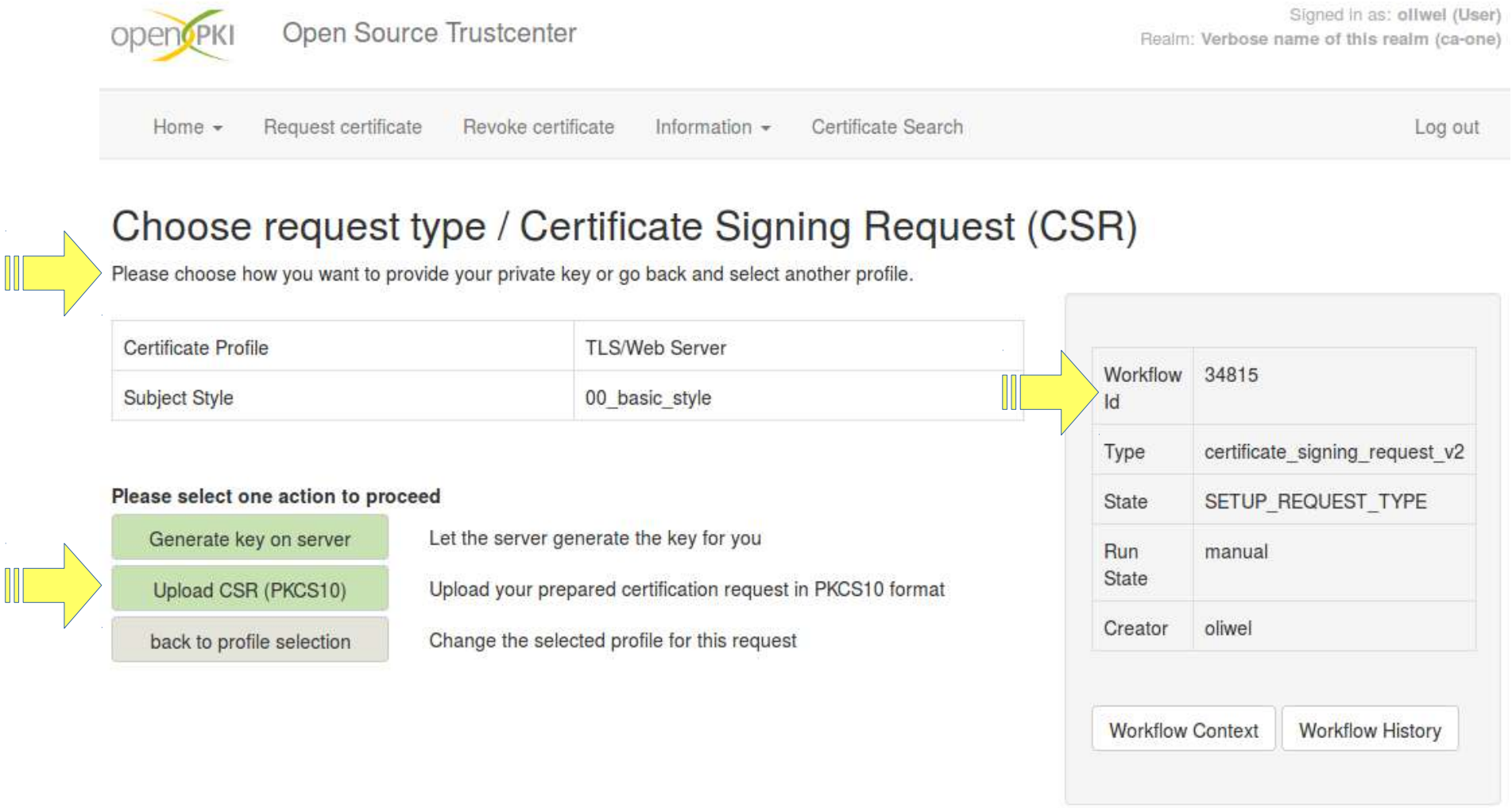


User Interface

- all logic is done on the server
- many aspects can be configured
- extension possible without „hacking the core“
- „step by step“ user guidance



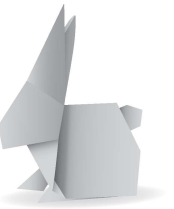
User Interface / Certificate Request



WhiteRabbitSecurity

User Interface / Certificate Request

WhiteRabbitSecurity



User Interface / Certificate Request

Review Request / Certificate Signing Request (CSR)

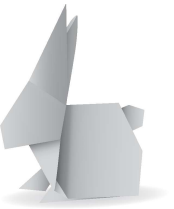
Your request is ready to be submitted. Please check if the information shown is correct. You can update information using the buttons bel
button to enqueue the request for further processing.

Certificate Subject	CN=www.example.com,DC=Test Deployment,DC=OpenXPKI,DC=org
Subject Alternative Name	DNS: www.example.com (ok) DNS: www.example.dw (FAIL)
Failed Policy Check DNS	www.example.dw (FAIL)
Certificate Profile	TLS/Web Server
Requestor Information	**Firstname** T **Lastname** T **Email address** mail@oliwel.de **Affiliation** System Owner

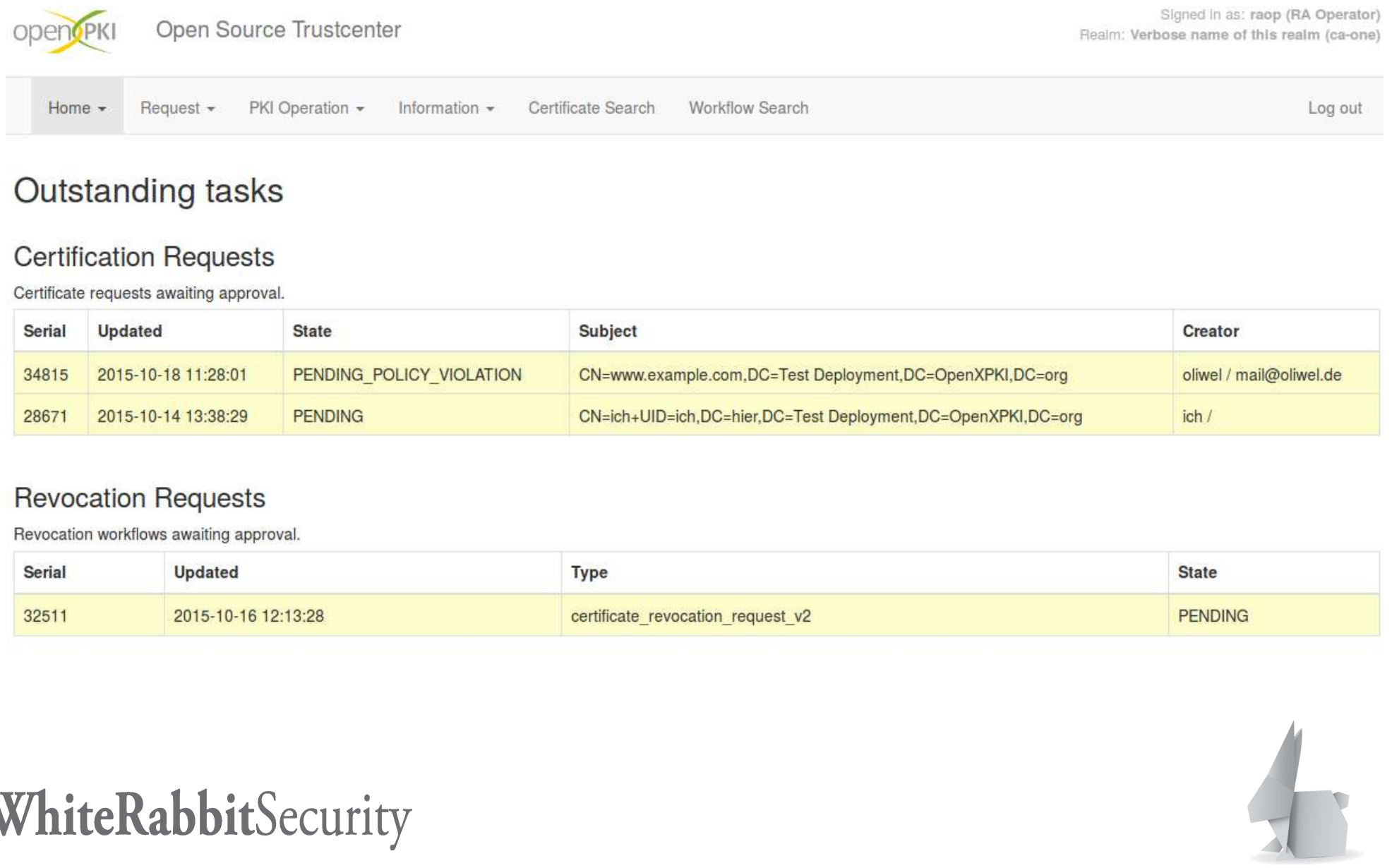
Submit with Policy Exception | Re-Evaluate Policy | Edit Subject | Edit SAN | Edit Certificate Info | Cancel Request

- make problems visible to the user early!
- catch errors before they reach your backoffice
- reduce support effort

WhiteRabbitSecurity



User Interface / Operator View



WhiteRabbitSecurity

User Interface / Operator View

Certification Requests

Certificate requests awaiting approval.

Serial	Updated	State	Subject	Creator
28671	2015-10-14 13:38:29	PENDING	CN=ich+UID=ich,DC=hier,DC=Test Deployment,DC=OpenXPKI,DC=org	ich /

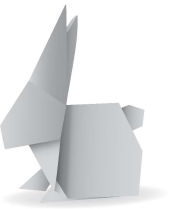
Requests with Policy Expcetion

Those requests violate the PKI policy and await exceptional approval.

Serial	Updated	User Comment for Exceptional approval	Subject	Creator
34815	2015-10-18 11:28:01	New Domain not in DNS	CN=www.example.com,DC=Test Deployment,DC=OpenXPKI,DC=org	oliwel / mail@oliwel.de

- trim the frontend to fit your support needs
- faster turnaround time, less errors
- it's only config (auditors love it!)

WhiteRabbitSecurity

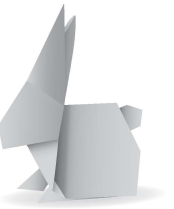


Workflow Engine

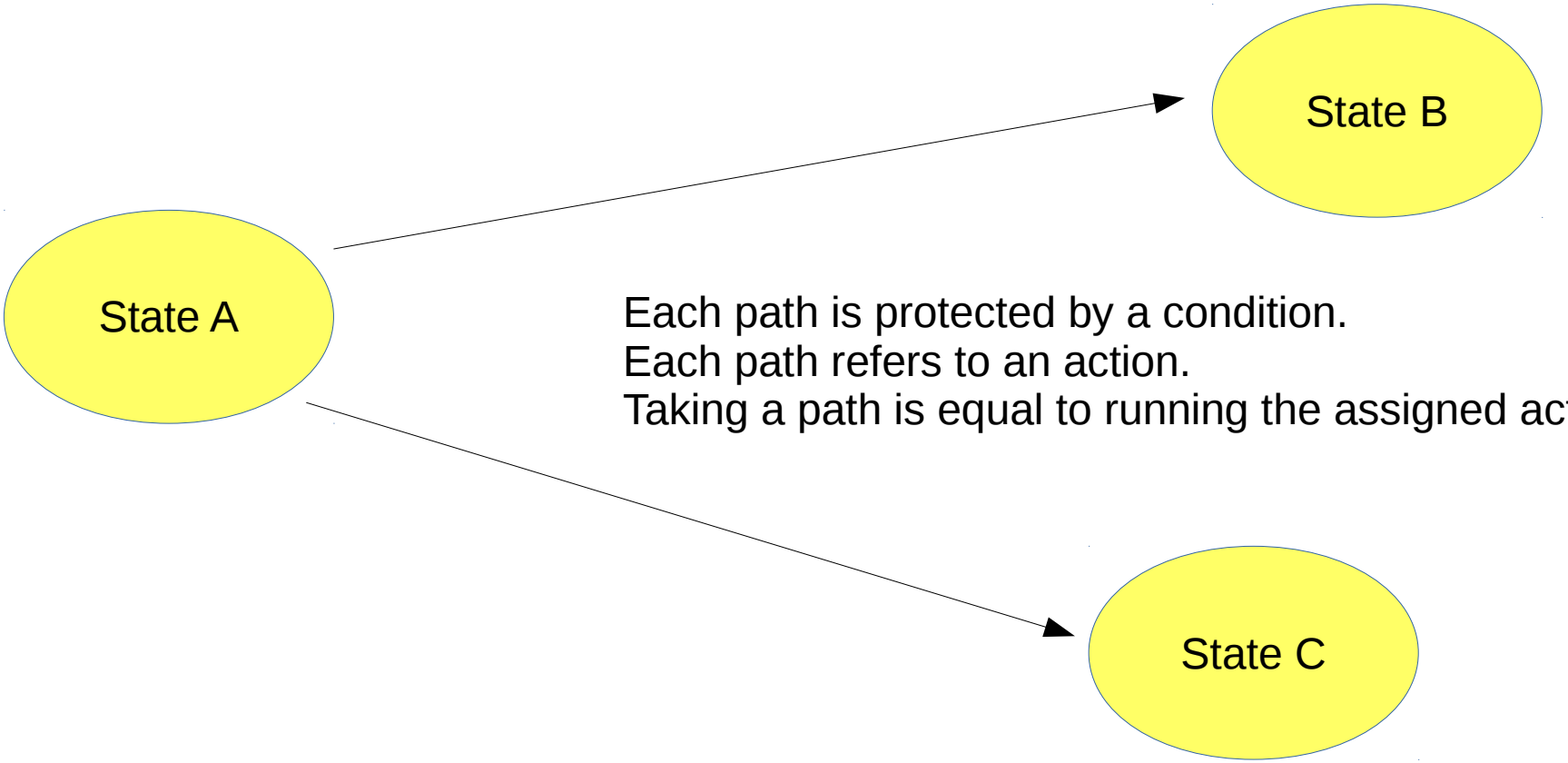
A magic carpet … can be used to transport humans … instantaneously or quickly to their destination.

https://en.wikipedia.org/wiki/Magic_carpet

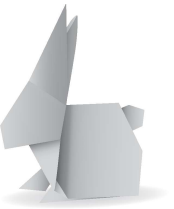
WhiteRabbitSecurity



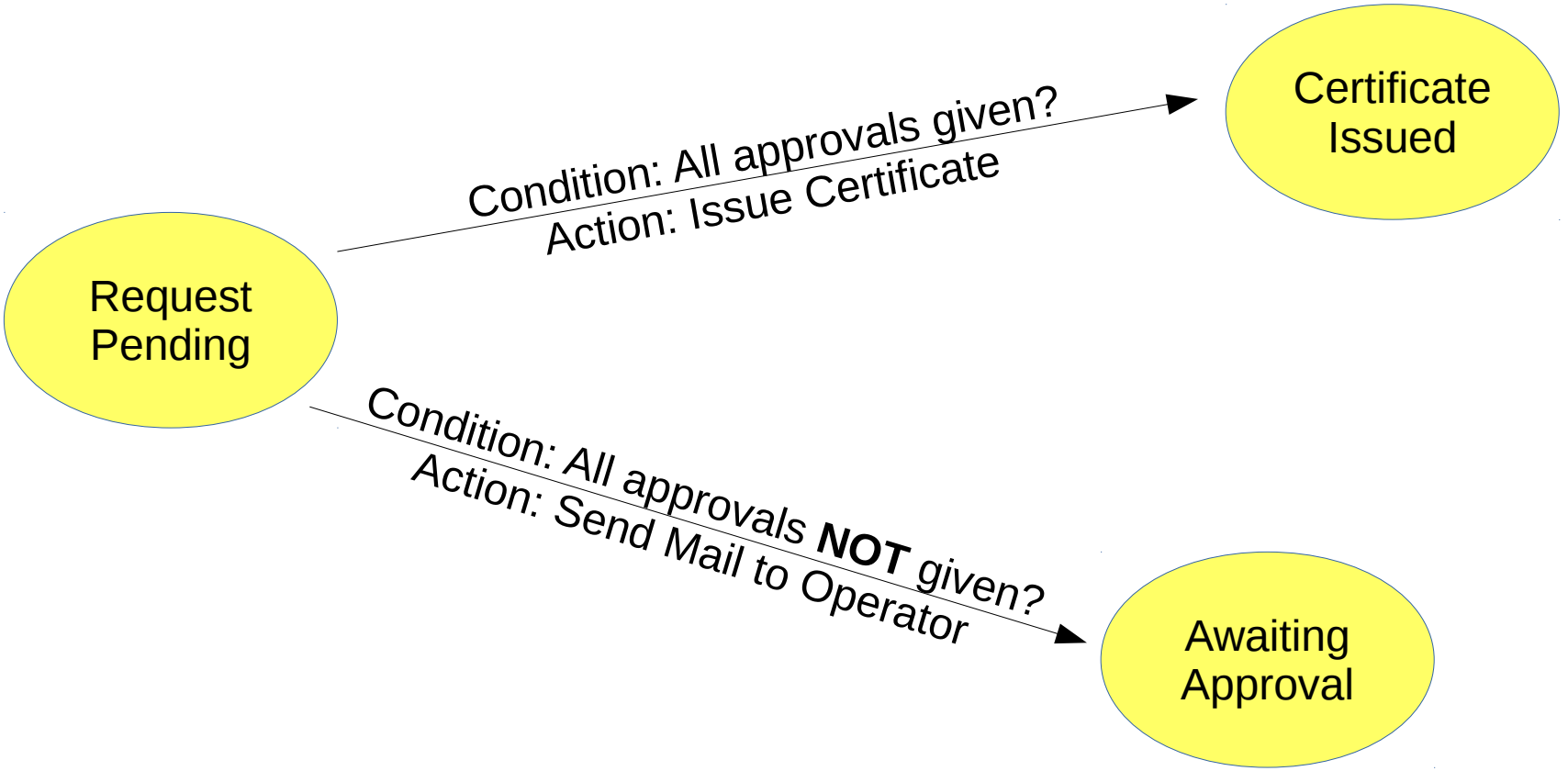
Workflow Engine / How it works

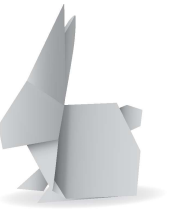


Each path is protected by a condition.
Each path refers to an action.
Taking a path is equal to running the assigned action.



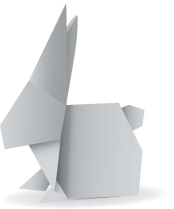
Workflow Engine / How it works





Workflow Engine / Building Blocks

- state machine
- workflow graph
- pre-defined code blocks
- configuration layer
- workflow context (scratchpad to store key/value pairs)
- scheduled execution in background



Workflow Engine / Building Conditions

- simple inline evaluation of context values

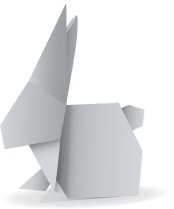
```
condition:
    cert_export_export_successful:
        class: Workflow::Condition::Evaluate
        param:
            test: not $context->{error_code};
```

- query external datasources using values from context

```
profile_has_san_section:
    class: OpenXPKI::Server::Workflow::Condition::Connector::Exists
    param:
        _map_config_path: profile.[% context.cert_profile %].style.[% context.cert_subject_style %].ui.san
```

- run custom code (defined in a custom class)
- conditions must give boolean answers (yes/no)

WhiteRabbitSecurity



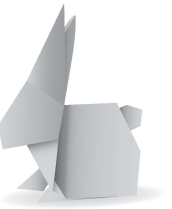
Workflow Engine / State-Action-Graph

- state with single condition and two paths

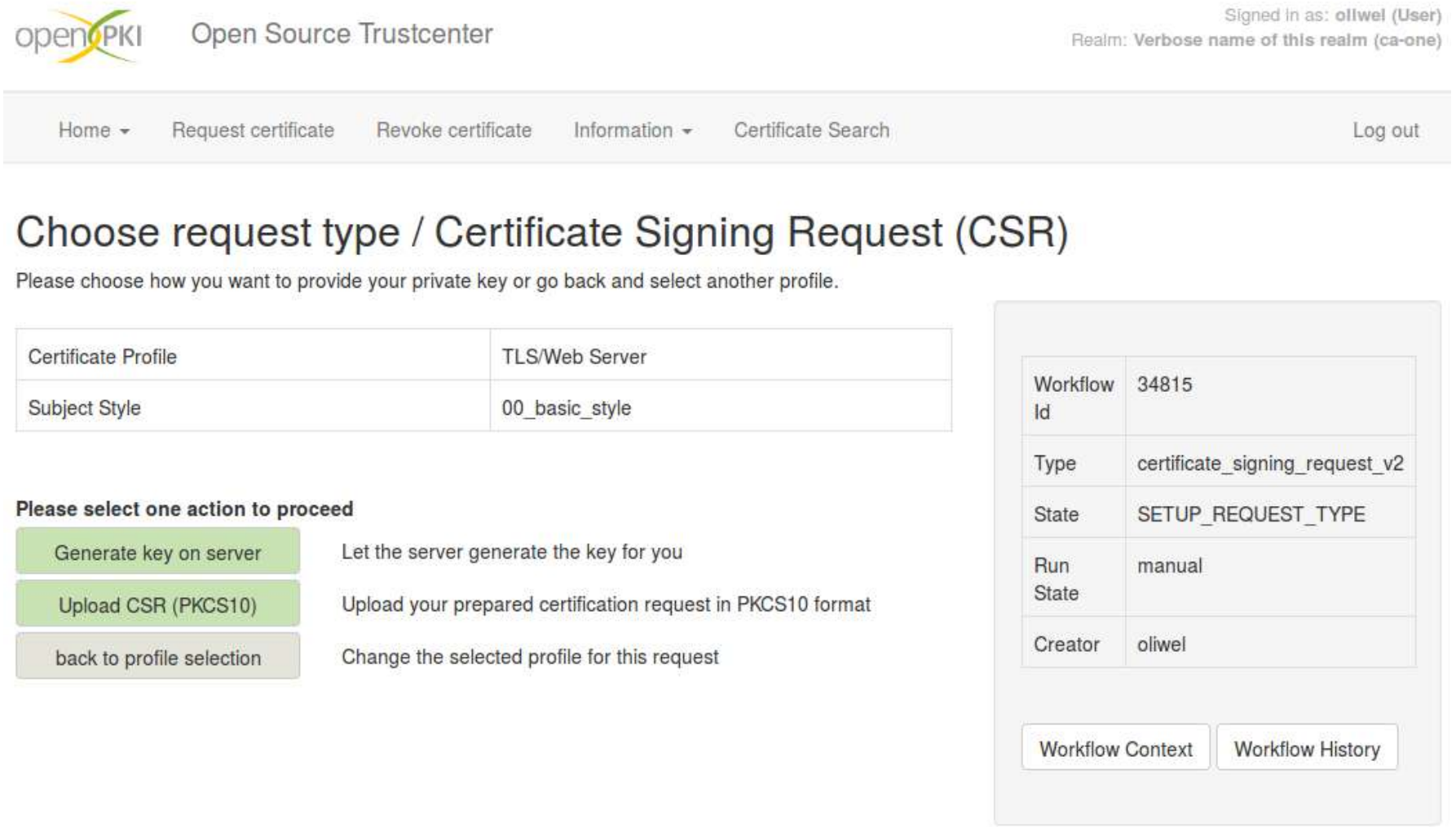
```
LOAD_NEXT_CA:
    autorun: 1
    action:
      - get_next_ca > ISSUE_CRL ? !is_ca_list_empty
      - global_noop > SUCCESS ? is_ca_list_empty
```

- define action

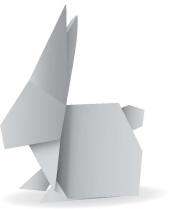
```
get_next_ca:
    class: OpenXPKI::Server::Workflow::Activity::Tools::WFArray
    param:
        array_name: ca_alias_list
        context_key: ca_alias
        function: shift
```

Workflow Engine / User Interaction

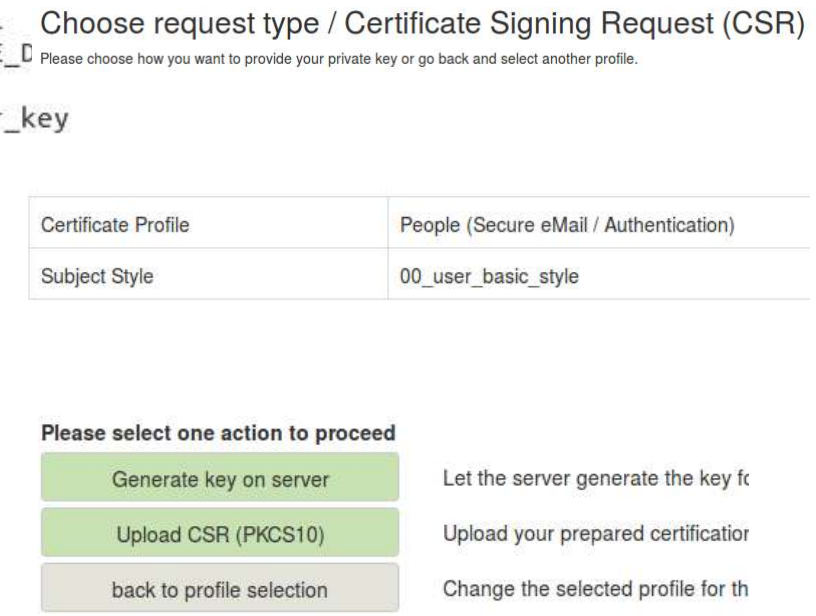


WhiteRabbitSecurity

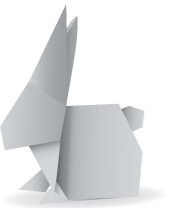


Workflow Engine / User Interaction

```
SETUP_REQUEST_TYPE:
    label: I18N_OPENXPKI_UI_WORKFLOW_STATE_CSR_SETUP_REQUEST_TYPE_LABEL
    description: I18N_OPENXPKI_UI_WORKFLOW_STATE_CSR_SETUP_REQUEST_TYPE_D
    action:
      - provide_server_key_params > ENTER_KEY_PASSWORD ? can_use_server_key
      - upload_pkcs10 > ENTER_SUBJECT ? can_use_client_key
      - select_profile > SETUP_REQUEST_TYPE
    output:
      - cert_profile
      - cert_subject_style
    button:
      _head: I18N_OPENXPKI_UI_WORKFLOW_HINT_SELECT_TO_PROCEED
      provide_server_key_params:
        description: I18N_OPENXPKI_UI_WORKFLOW_HINT_SERVER_KEY_PARAMS
        format: expected
      upload_pkcs10:
        description: I18N_OPENXPKI_UI_WORKFLOW_HINT_PKCS10_UPLOAD
        format: expected
      select_profile:
        label: I18N_OPENXPKI_UI_WORKFLOW_HINT_CHANGE_PROFILE_LABEL
        description: I18N_OPENXPKI_UI_WORKFLOW_HINT_CHANGE_PROFILE
        format: optional
```

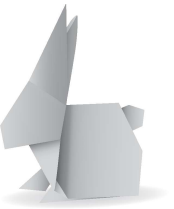


WhiteRabbitSecurity

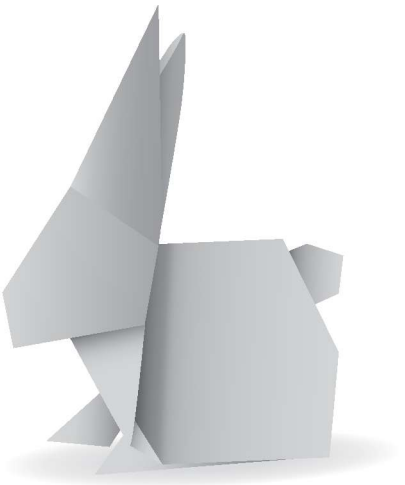


Workflow Engine / Summary

- customization of shipped workflows possible
- write your own workflow using pre-defined code blocks
- create new code for actions and conditions
- user interface auto-generated from workflow config



Thank You!



WhiteRabbitSecurity

Oliver Welter, 2015/10