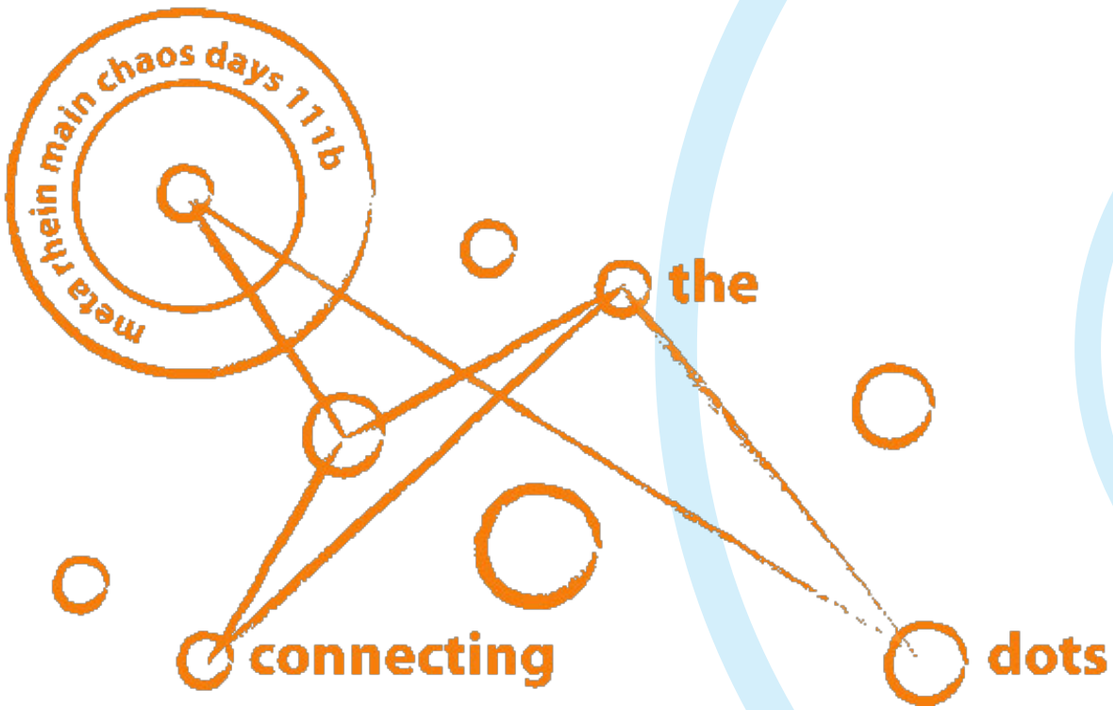
# Open Source Public Key Infrastrukturen mit OpenXPKI

Alexander “alech” Klink, Cynops GmbH



# Übersicht

- Kurze Einführung in PKI
- OpenXPKI Features und Konzepte
- Workflows am Beispiel Zertifikatsantrag
- Simple Certificate Enrollment Protocol (SCEP)
- Smartcard-Personalisierung
- Support
- Hackers wanted!

# Public Key Infrastrukturen

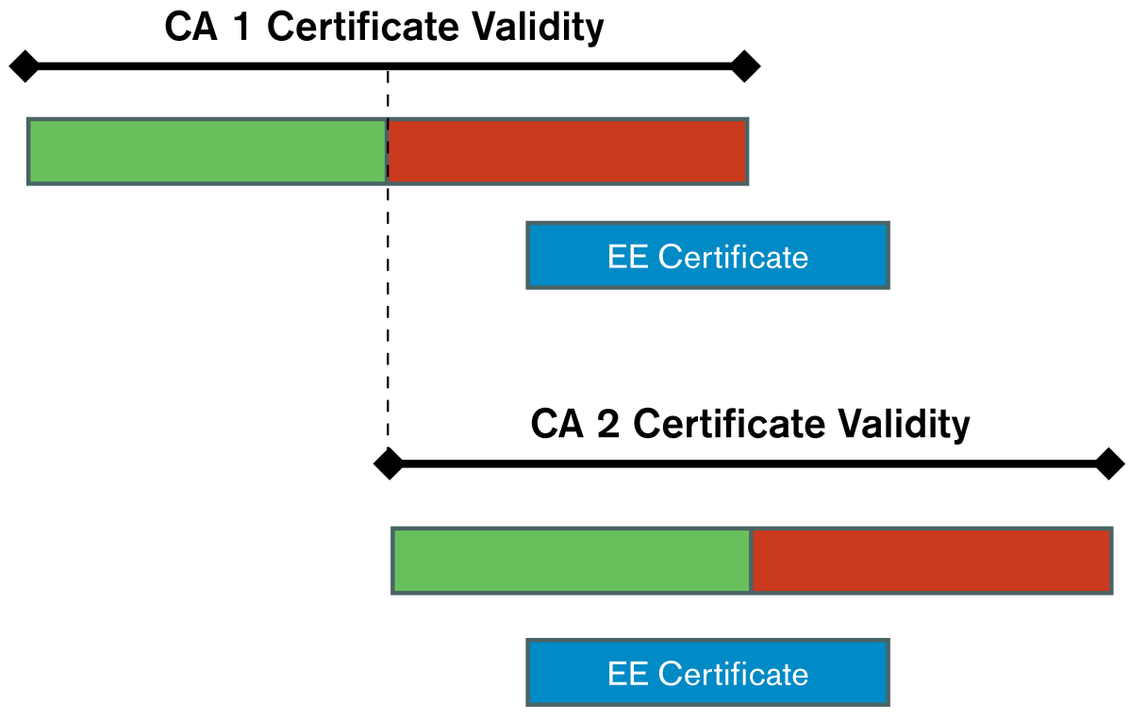
## Protecting your private bits

- Alice und Bob, oder auch nicht
- Certificate Authorities (CAs) stellen Zertifikate aus – an end entities (User/Server) oder CAs
- Zertifikatsantrag + Vertrauen + Freigabe = Zertifikat
- hierarchische Strukturierung von CAs

# OpenXPKI features

## All the best, for free

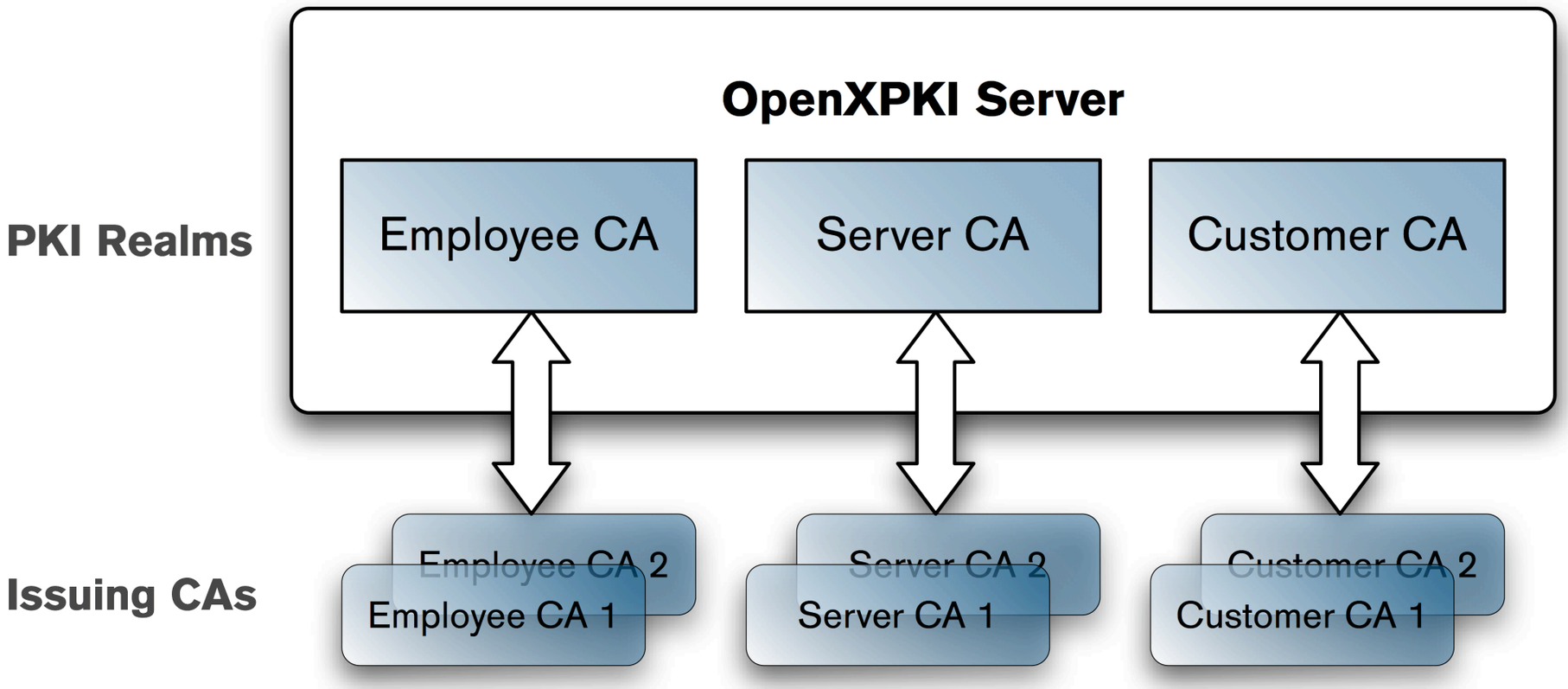
- Enterprise Features, nicht die typische 5-Minuten CA für das Heim-VPN
- CA rollover für kontinuierlichen Betrieb



# OpenXPKI features

## All the best, for free

- PKI realms ≈ Multimandantenfähigkeit



# OpenXPKI Features

## All the best, for free

- Support für professionelle Backends: HSM, RDBMS, LDAP, ...
- I18N: Englisch, Deutsch, Russisch
- Konfigurationsvererbung (`<tag super="../../foo">`)
- Request-Tracker-Anbindung
- Flexibilität durch Workflows
- Interfaces: Web, SCEP, Perl
- Smartcard-Personalisierung

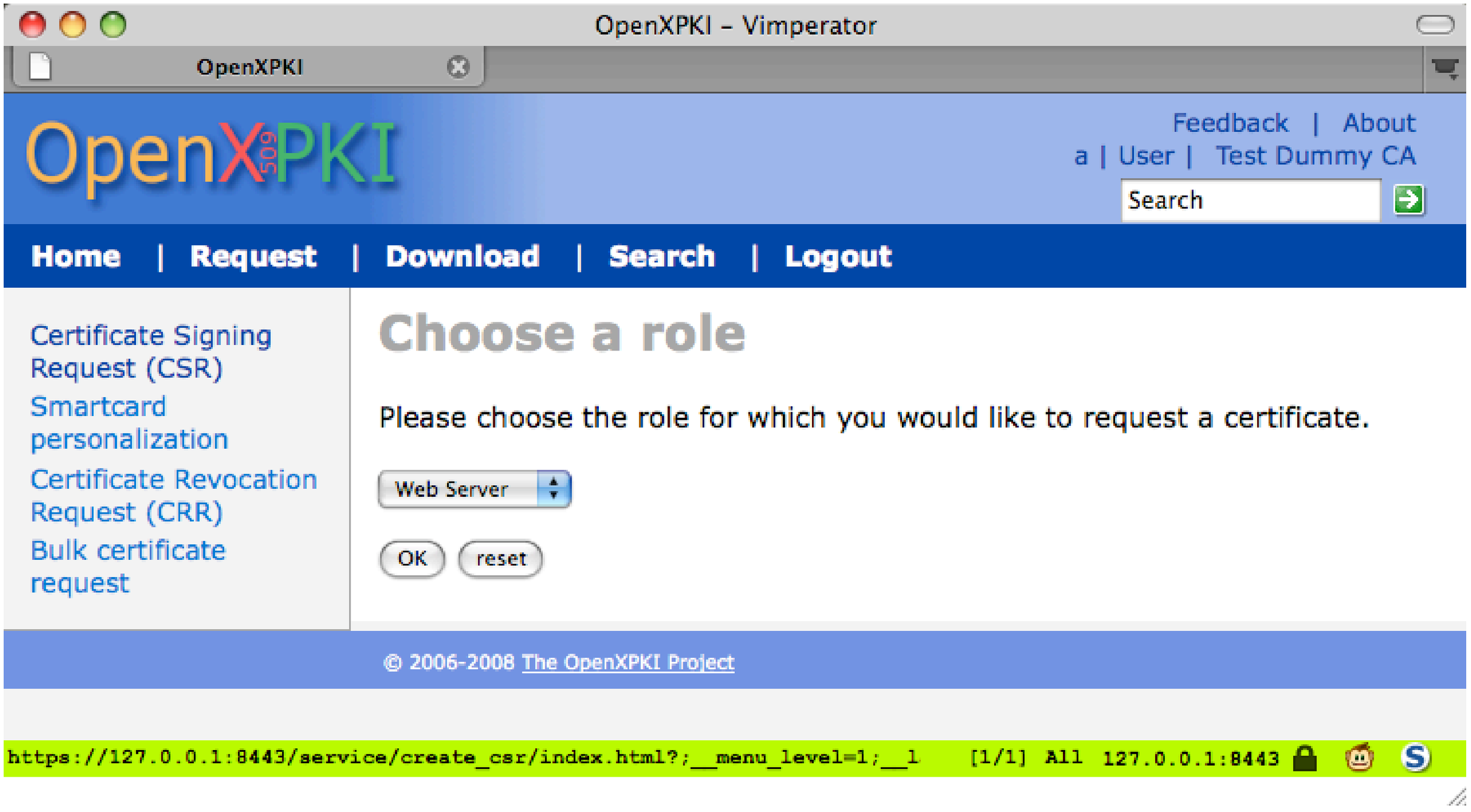
# Workflows

## Beispiel Zertifikatsantrag

- Wie sieht es für den User aus?
- Zugriff über ein Webinterface
  - Erstellung durch Browser (SPKAC/XEnroll)
  - PKCS#10-Upload
  - Schlüsselgenerierung auf der CA
- Und so sieht's aus (ungefähr) ...

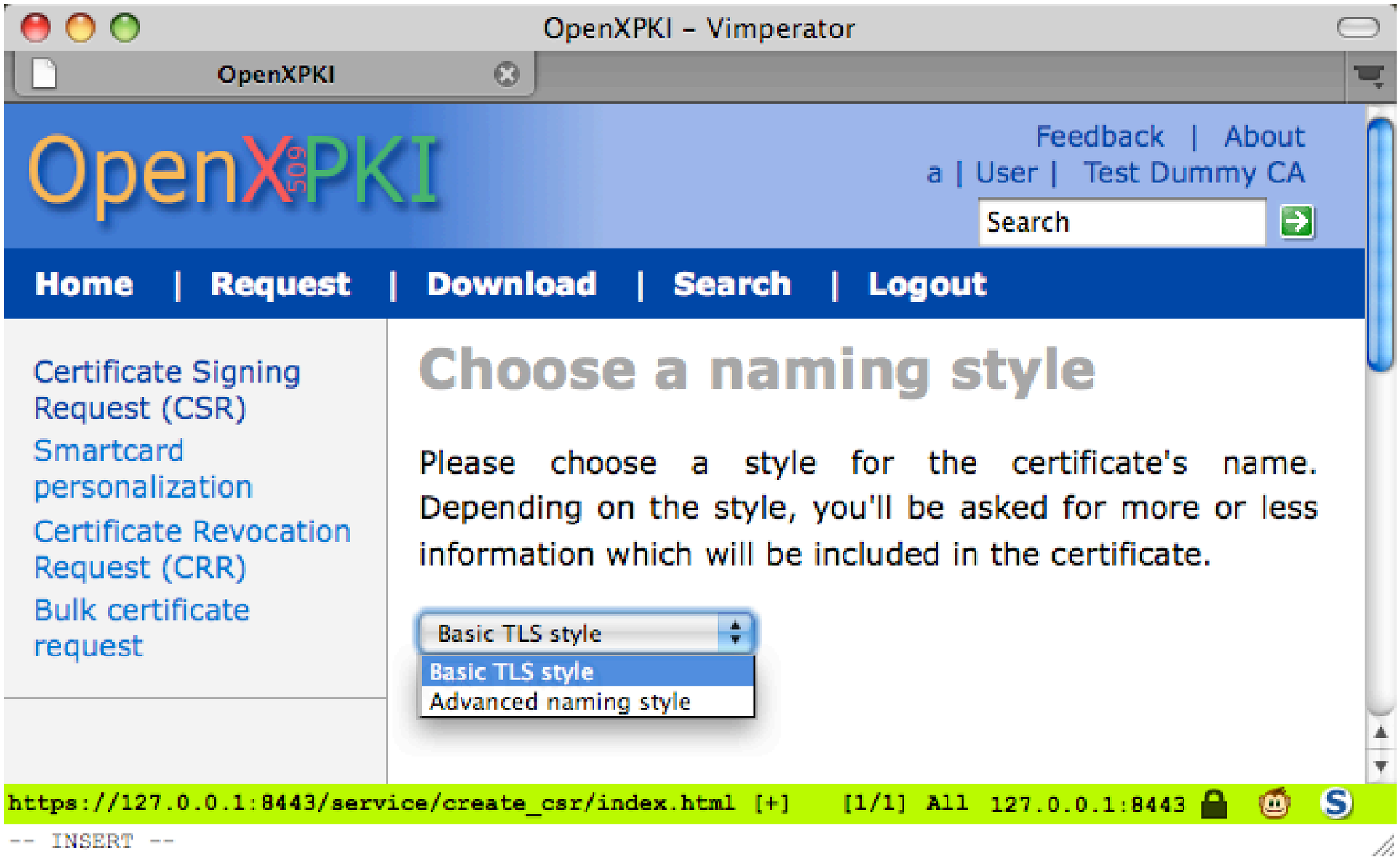
# Workflows

## Zertifikatsantrag: User Interface



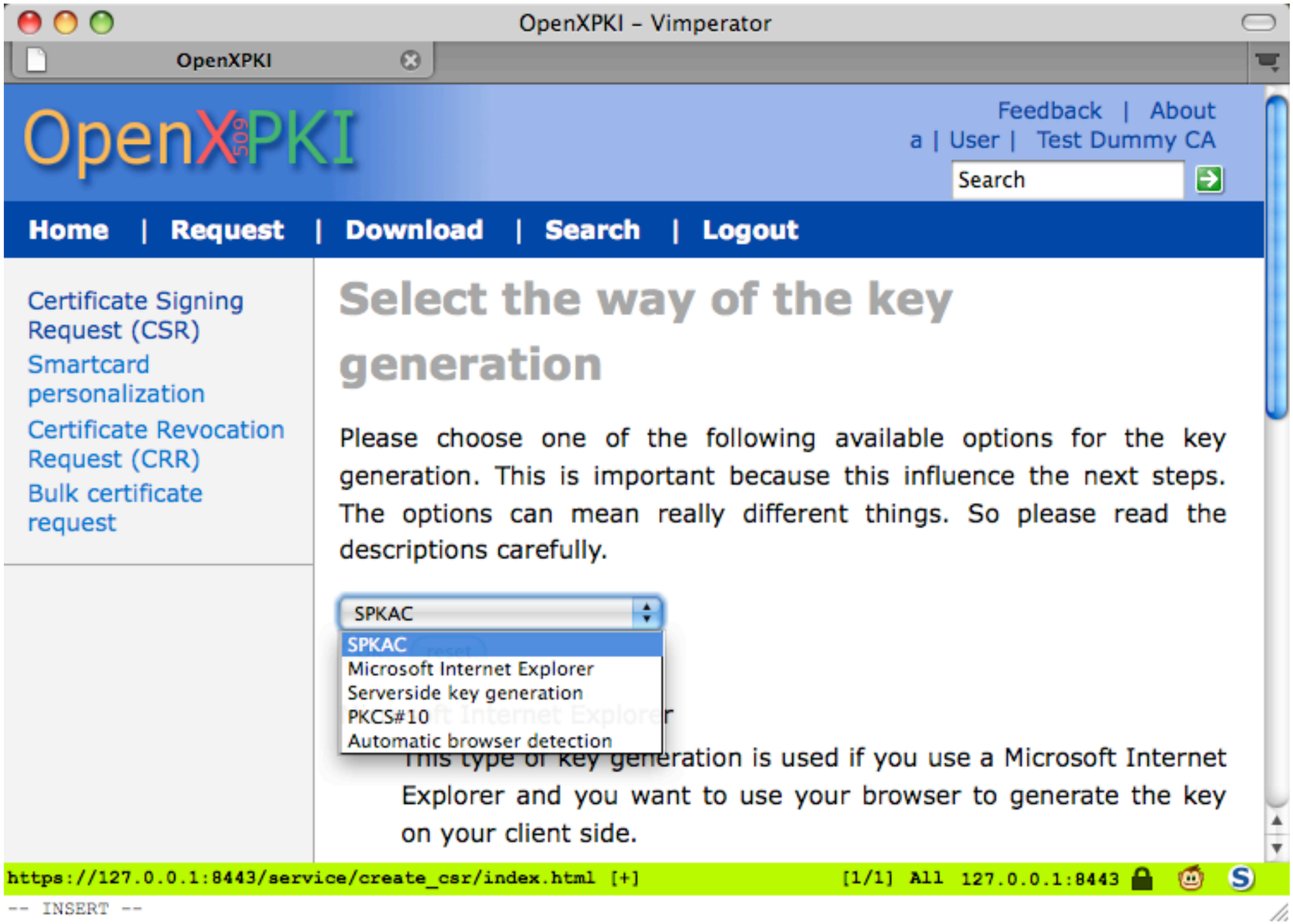
# Workflows

## Zertifikatsantrag: User Interface



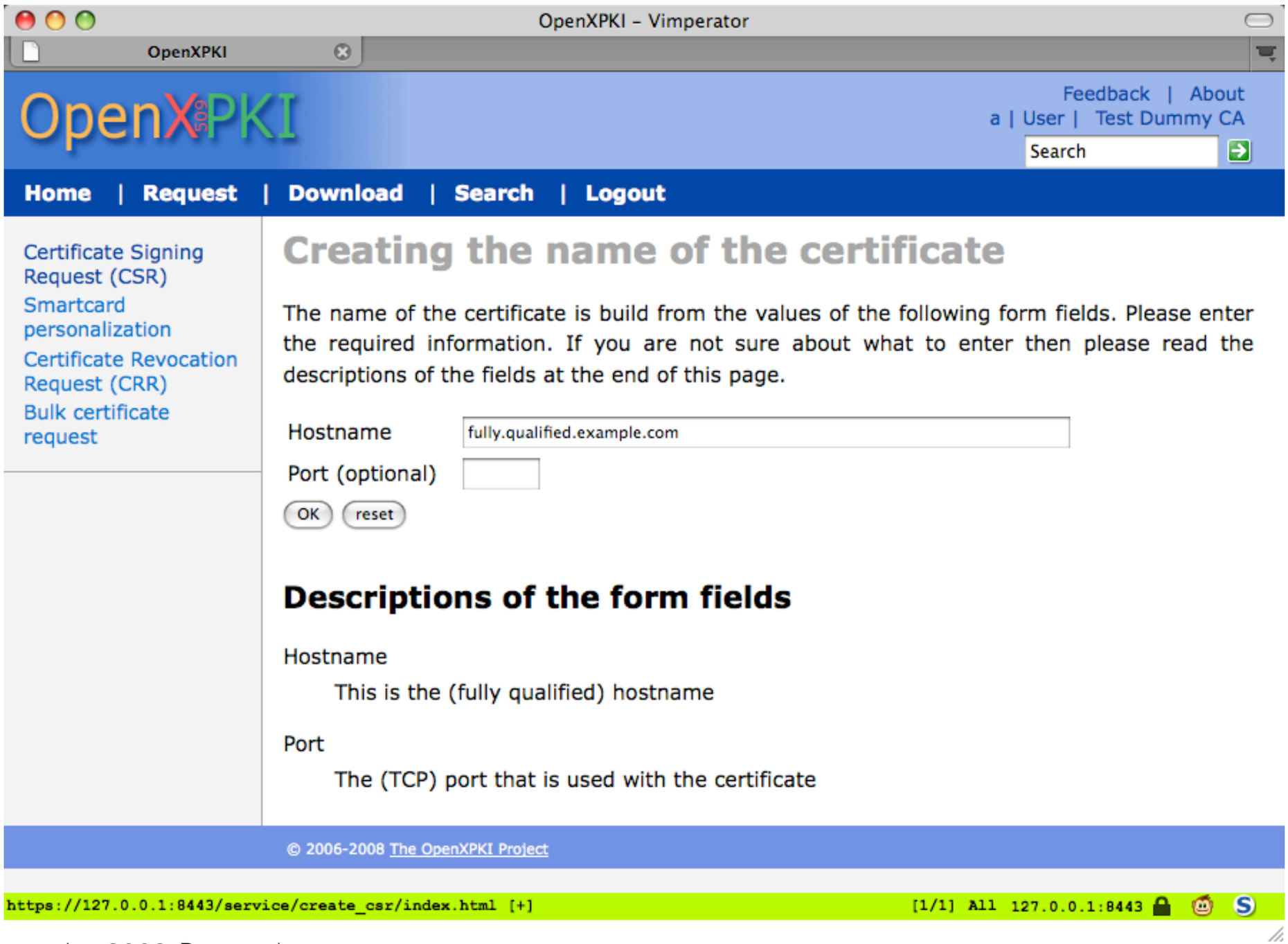
# Workflows

## Zertifikatsantrag: User Interface



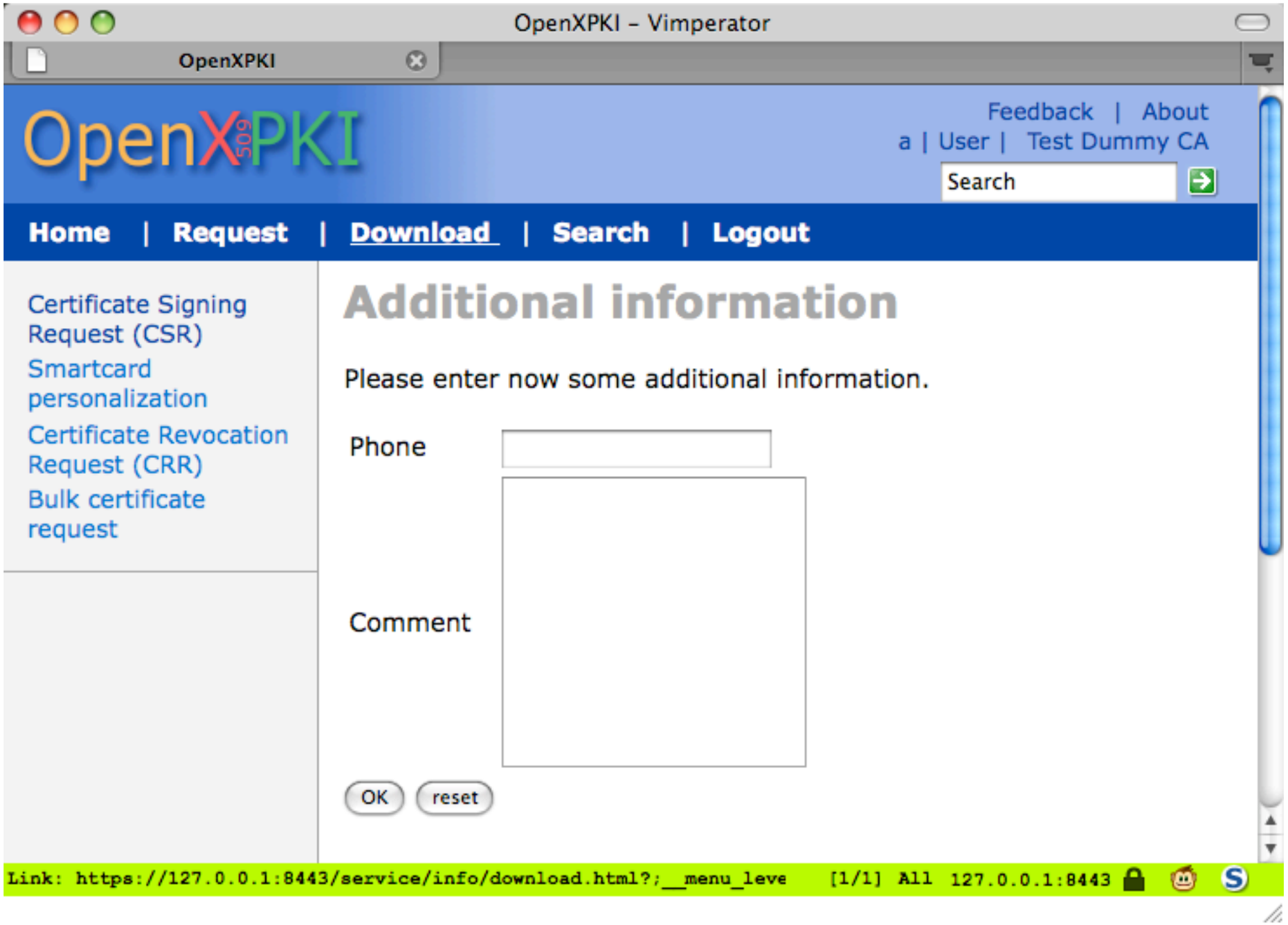
# Workflows

## Zertifikatsantrag: User Interface



# Workflows

## Zertifikatsantrag: User Interface

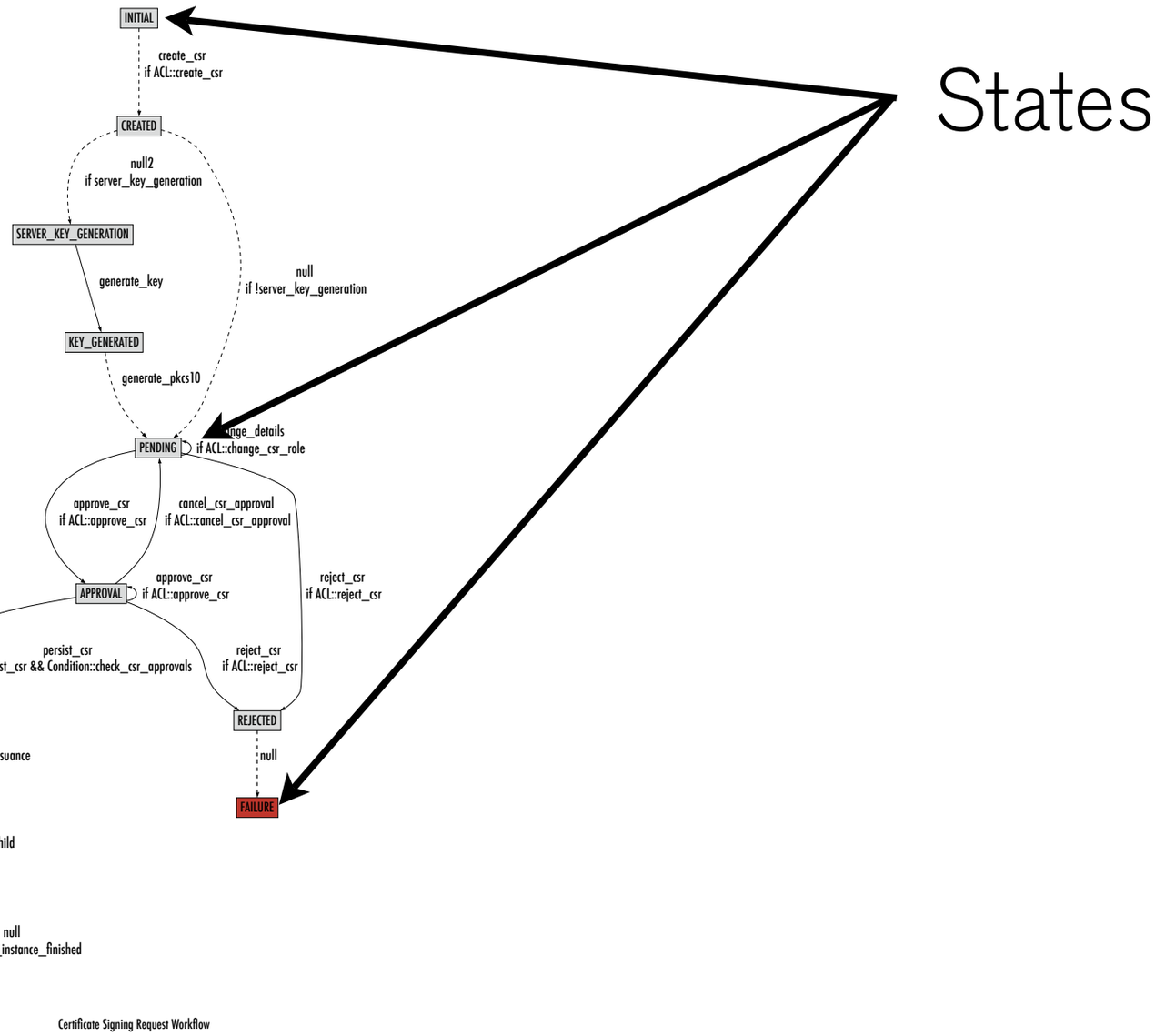


# Workflows

## Zertifikatsantrag: User Interface

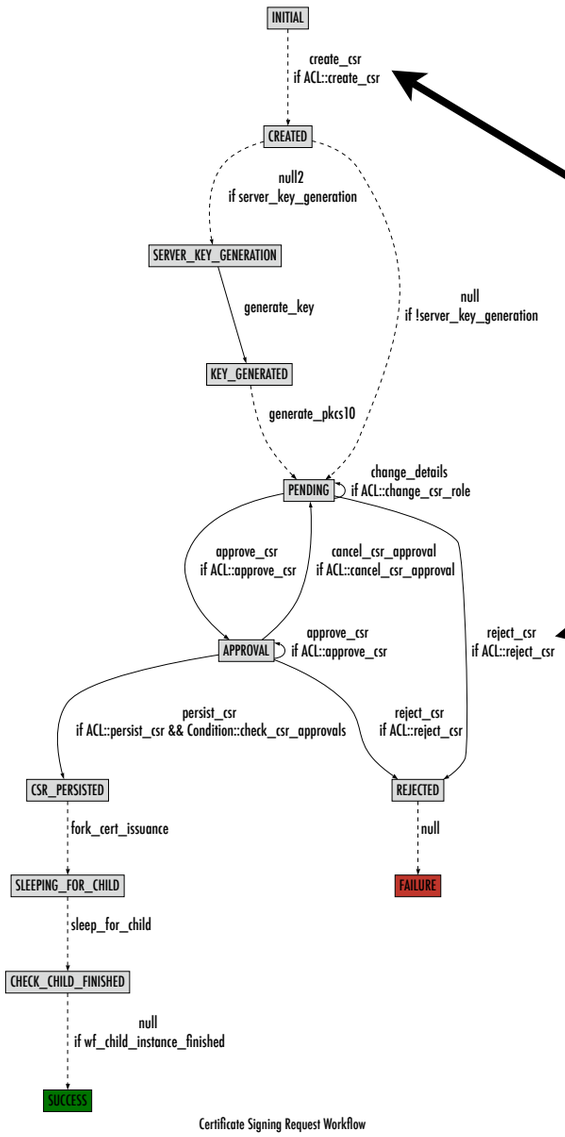

# Workflows

## Visualisierung mit GraphViz



# Workflows

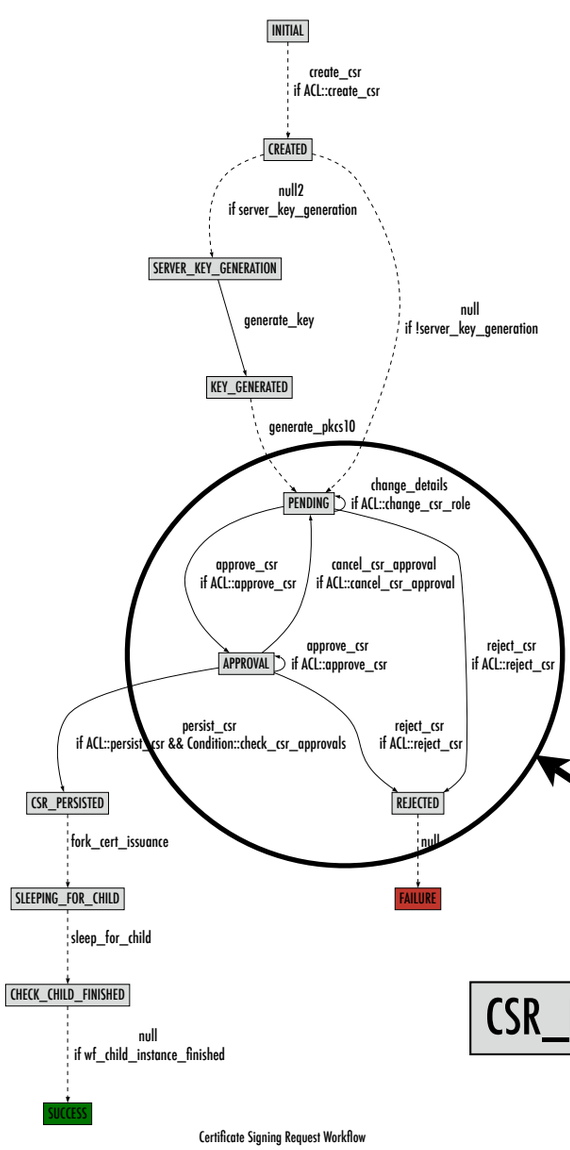
## Visualisierung mit GraphViz

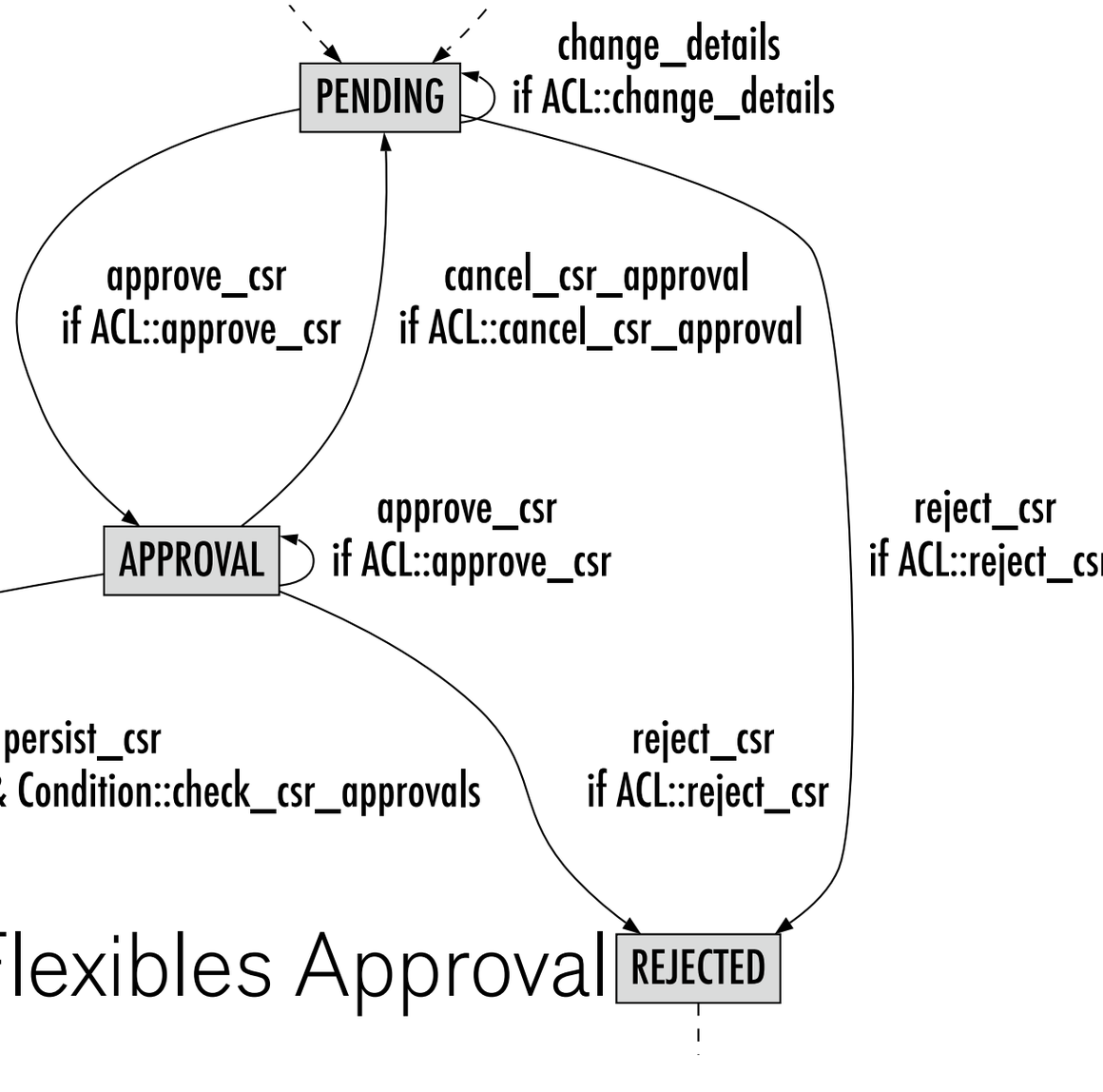



Activities (if condition)

# Workflows

## Visualisierung mit GraphViz



# Workflows

## XML-Konfiguration

```
50   <state name="APPROVAL">
51     <action name="persist_csr"
52             resulting_state="CSR_PERSISTED">
53       <condition name="ACL::persist_csr"/>
54       <condition name="Condition::check_csr_approvals"/>
55     </action>
56     <action name="approve_csr"
57             resulting_state="APPROVAL">
58       <condition name="ACL::approve_csr"/>
59     </action>
60     <action name="cancel_csr_approval"
61             resulting_state="PENDING">
62       <condition name="ACL::cancel_csr_approval"/>
63     </action>
64     <action name="reject_csr"
65             resulting_state="REJECTED">
66       <condition name="ACL::reject_csr"/>
67     </action>
68   </state>
```

a snippet from workflow_def_certificate_signing_request.xml

# Workflows

## ganz unten: Perl & Workflow.pm

```perl
sub execute {
    my $self        = shift;
    my $workflow    = shift;
    my $context     = $workflow->context();

    my $type    = $context->param('csr_type');
    my $profile = $context->param('cert_profile');

    [ ... ]

    $dbi->insert(
        TABLE => 'CSR',
        HASH  => {
            'PKI_REALM'   => $pki_realm,
            'CSR_SERIAL'  => $csr_serial,
            'DATA'        => $data,
            [ ... ]
        },
    );
    $dbi->commit();
    $context->param('csr_serial' => $csr_serial);
}
```

a snippet from Workflow/Activity/CSR/PersistRequest.pm

# SCEP

## Worum geht's?

- SCEP = Simple Certificate Enrollment Protocol
- IETF-Draft ausgehend von Cisco
- Protokoll um automatisch Zertifikate zu beantragen oder zu erneueren
- PKCS#10 in PKCS#7 über HTTP (könnte also schlimmer sein ...)
- In vielen Hardware-Devices eingebaut

# SCEP

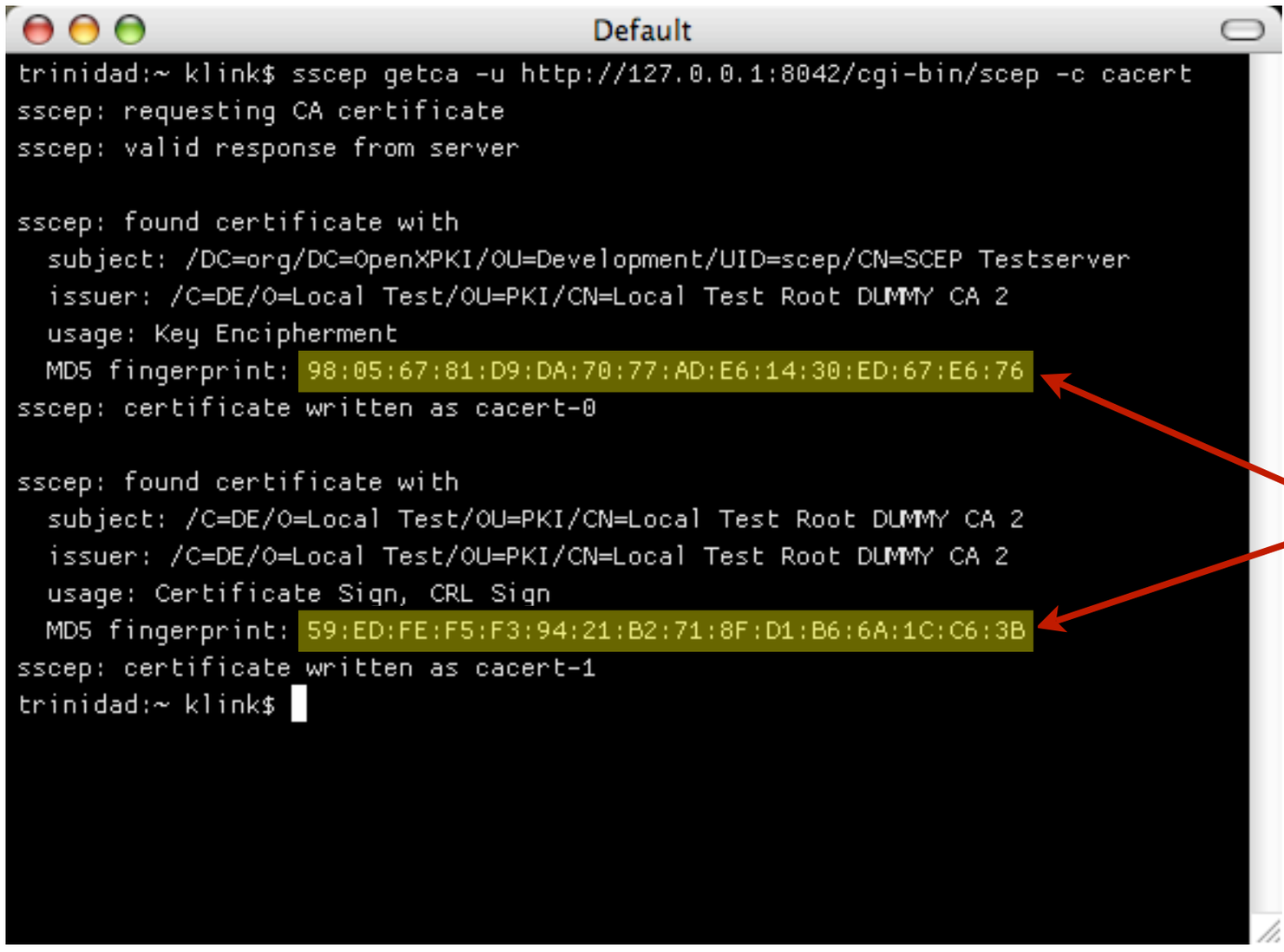
## Die Motivation

- Alternative: manueller Prozess:
  - Zertifikat anfordern, installieren, Ablaufdatum in den Kalender eintragen (wahrscheinlich mindestens den vom nächsten Jahr)
  - Vor dem Ablaufdatum: gleiche Prozedur
  - wiederholen ...
  - ziemlich unmöglich für große Installationen

# SCEP

## ... unter Unix mit sscep

CA-Zertifikate vom SCEP-Server anfordern

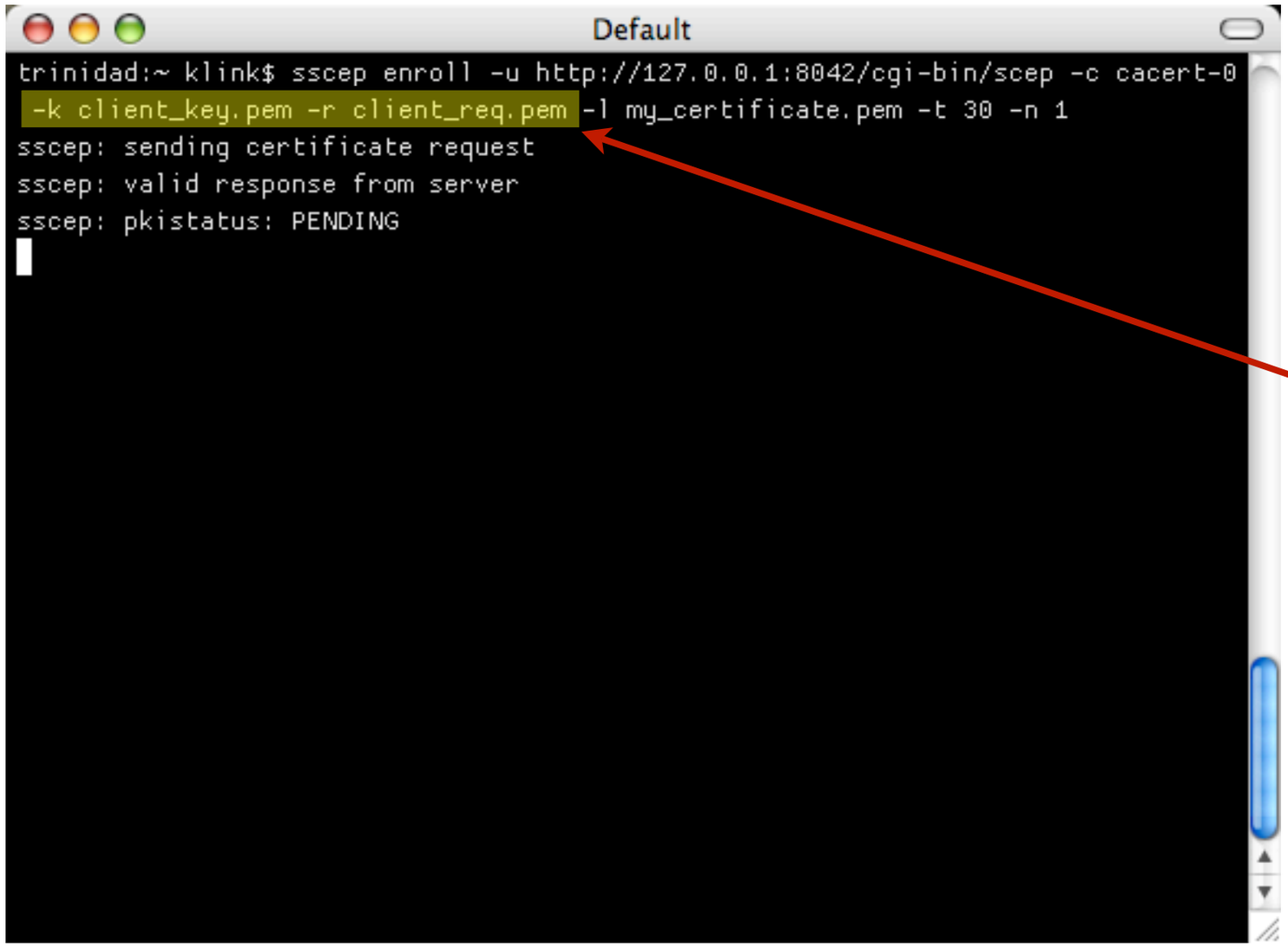



Fingerprints überprüfen (out of band)

# SCEP

## ... unter Unix mit sscep

Zertifikatsantrag an die CA schicken

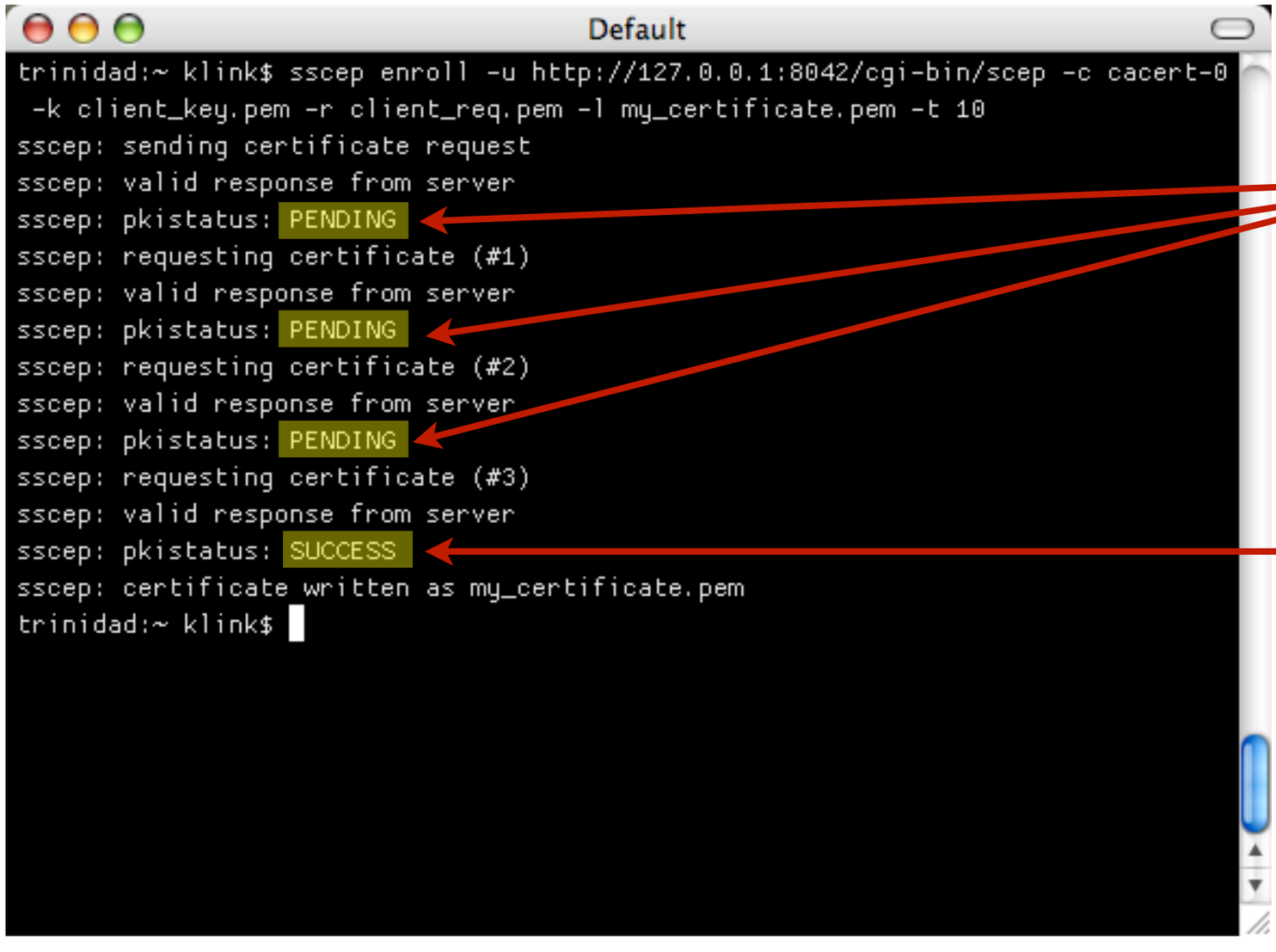



Mit einem vorher generierten Schlüssel und PKCS#10-File

# SCEP

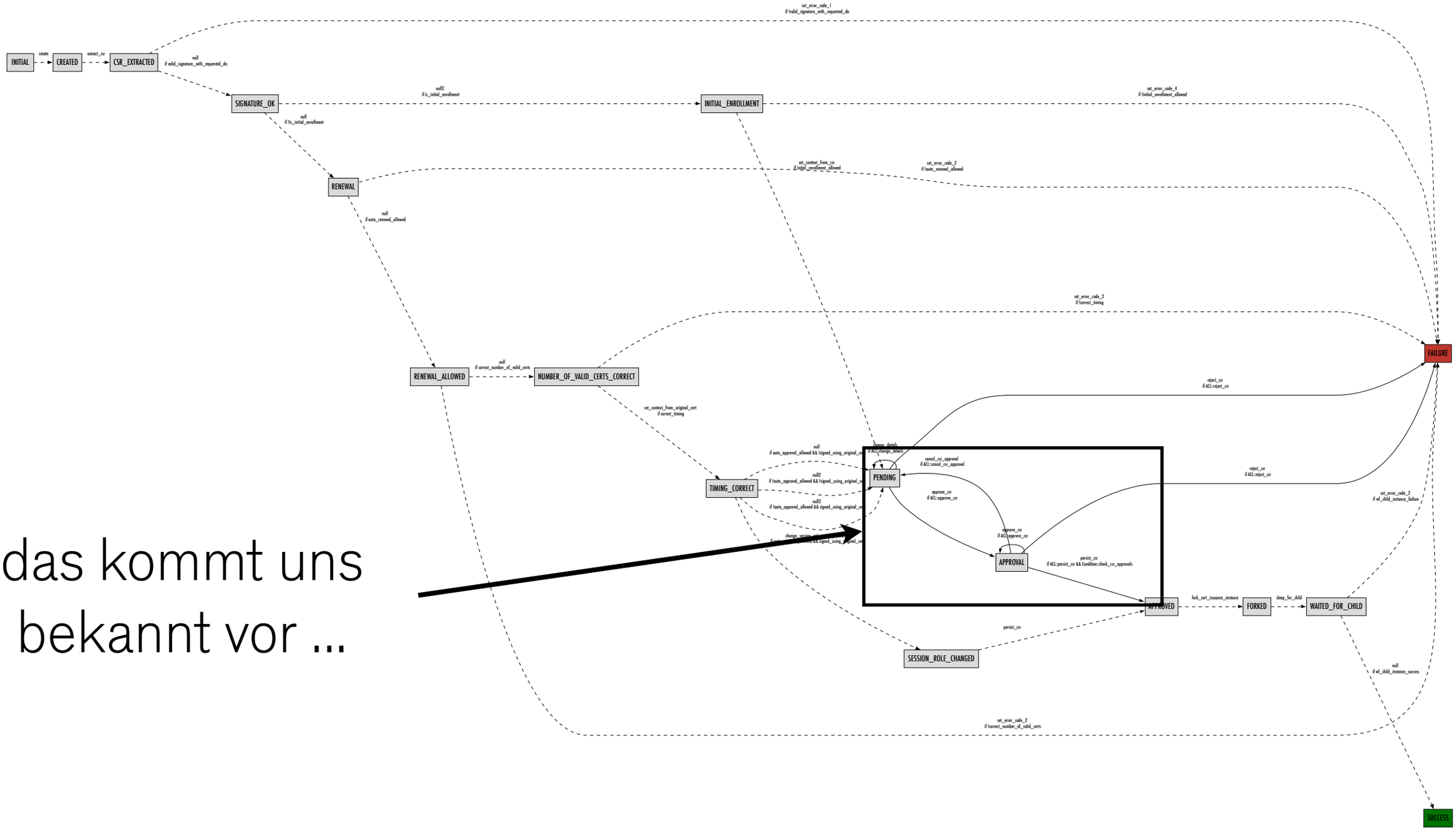
## ... unter Unix mit sscep

Zertifikatsantrag an die CA schicken



# SCEP

## ... schon wieder Workflows

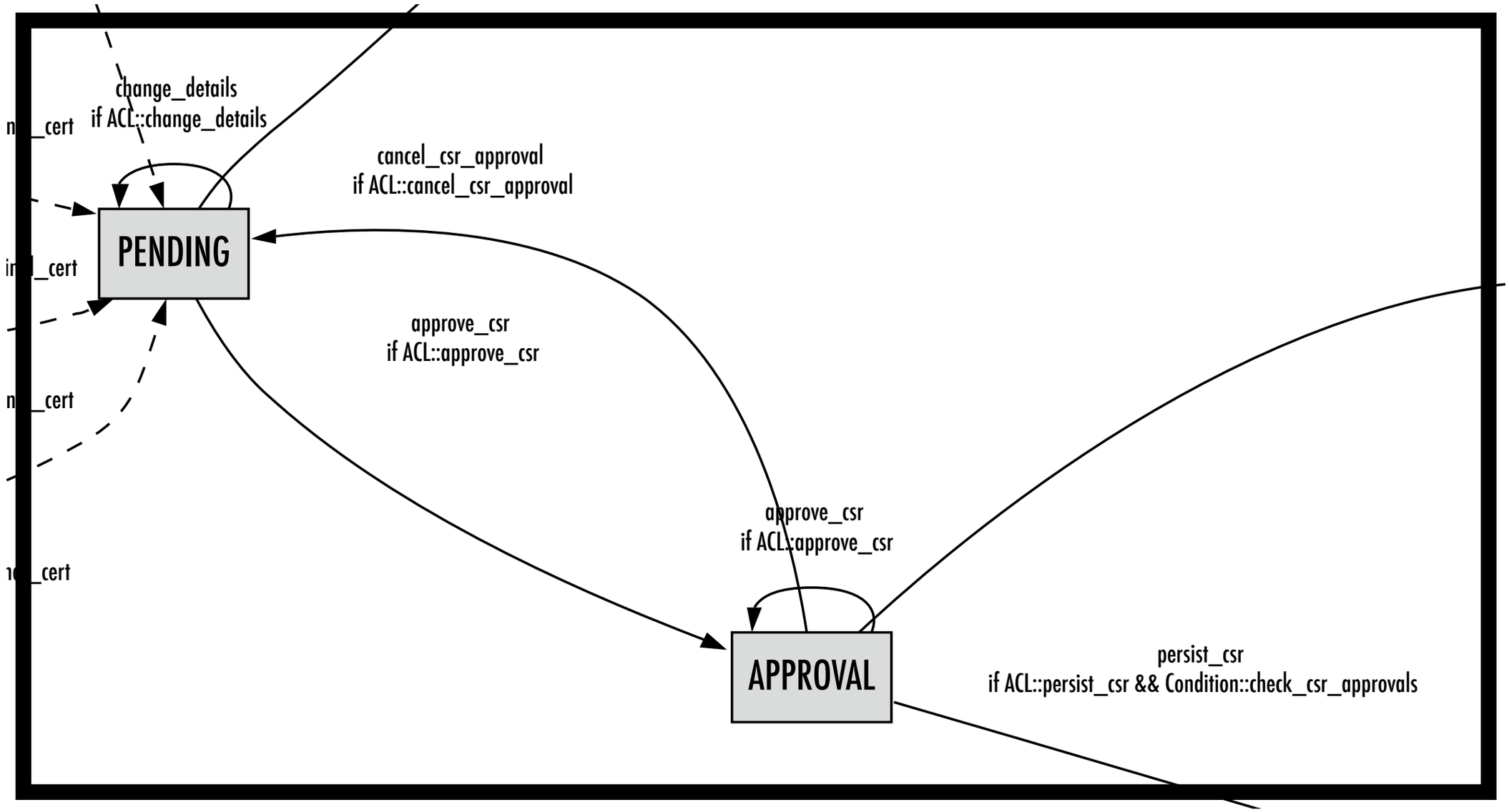



das kommt uns
bekannt vor ...

# SCEP

## ... schon wieder Workflows

Approval:



# SCEP

## Alternativen

- PKIX-CMP (Certificate Management Protocol)
  - standardisiert in RFC 4210, 4211, aber sehr wenig benutzt ...
- CMC (CM over CMS, kein Transport definiert)
  - RFC 2797, wird von Microsoft CA mit COM/DCOM als proprietärem Transport genutzt
  - Support in OpenXPKI wäre super, da wir so einfach eine MS CA für Domain-Controller-Enrollment ersetzen könnten

# SCEP

## Hilfswerkzeug: CertNanny

- Automatisches Renewal von Zertifikaten und Austausch im Keystore (SCEP + Signatur mit dem “alten” Zertifikat)
- Verfügbar auf Unix (getestet: Linux, AIX, Solaris, Darwin), Win32, Tandem NonStop
- Arbeitet mit OpenSSL-Schlüsseln, PKCS#8, Java Keystores, IBM GSKit, Windows Certificate Store, PKCS#12
- weit verbreitet in einer großen Bank

# Smartcard-Personalisierung

## Wofür eigentlich Smartcards?

- Authentisierung:
  - Smartcard logon
  - (W)LAN (802.1x)
  - Fileserver-Verschlüsselung (outsourcing!)
- (E-Mail)-Encryption
  - ...aber: Problem Key Recovery

# Smartcard-Personalisierung

## Warum Self-Service-Personalisierung?

- Eine große Anzahl Karten in einen funktionsfähigen Zustand bringen:
  - selbst machen ...
  - ... oder es die Benutzer machen lassen
- Da fällt die Entscheidung leicht :-)
  - User braucht nur IE, Smartcard-Treiber und ein bisschen Zeit → screencast

# Die Zukunft von OpenXPKI

## Releases und neue Ideen

- Erstes Release “real soon now” (dieses Jahr)
- Dokumentation (Wiki statt DocBook)
- Testen, testen, testen
- mittelfristig:
  - Anbindung an Tivoli, Nagios, ...
  - Cluster-Support
- Neue Live-CD-Version

# Support

## Wenn RTFM nicht weiterhilft ...

- Leider ist die Dokumentation noch nicht fertig ...
- ... aber selbst wenn sie es irgendwann ist, ist OpenXPKI immer noch ziemlich komplex
- Support über die Mailingliste: openxpki-users@lists.sf.net, Sourceforge Tracker
- #openxpki im Freenode-Netz
- Cynops GmbH bietet auch kommerziellen Support an ... :-)

# Hackers wanted

## Use it, break it, enhance it ...

- Wir sind immer auf der Suche nach neuen Leuten
  - Entwicklern
  - Bug^H^H^H Feature-Reportern :-)
  - Leute, die gerne WTFM betreiben
  - Leute, die gerne Exploits finden
  - Leute, die SCEP-Hardware besitzen
- Gleich, oder später: openxpki-devel@lists.sf.net

# Fragen?

## Kommentare? Konfusion?

- Jetzt ist Zeit dafür ...
- Aber gerne auch noch später:
  - hier oder via a.klink@cynops.de
- Danke für Eure Zeit!