CCLRC
Rutherford Appleton Laboratory

Data GRID

Alternative CA software

Jens G Jensen
UK e-Science CA
Rutherford Appleton Laboratory

A talk in three parts

- Part one being about Baltimore uniCert
- Part two, being the second part, about pyCA
- Part three, being the third and final part, about the Java based solution that we’re working on

Part one

Baltimore uniCert

Jens G Jensen
UK e-Science

Baltimore uniCert

- Spent a day talking with Baltimore techies
- We haven’t actually tested it yet…
- …so presentation will be *salvo errore et omissione*…
- You can get more information from the Baltimore web site (but will have to register to get it ☹)
- And we also know people you can ask…

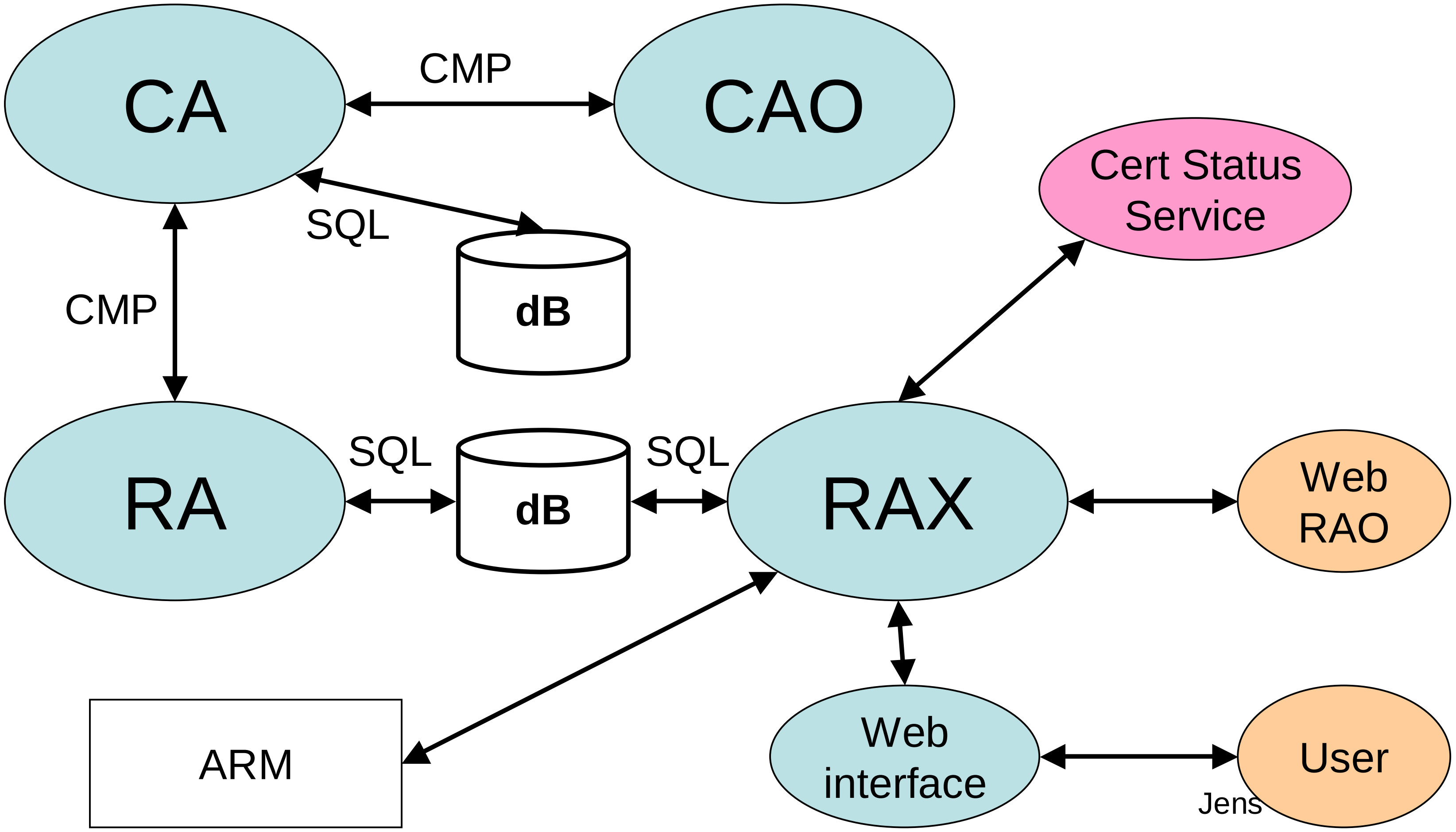
uniCert, technical requirements

- Root CA is *online* – works with FIPS 140 level 3 or 4 HSM
- Must use Oracle as underlying database (comes with licence)
- CA Operator (see later) must run on Microsoft Windows
- All other parts of the CA run on Solaris (two boxes required)

uniCert, terminology

- “CA” – refers to online *signing* system
- “RA” – refers to online request management system
- “RA Operator” (“RAO”) – the (human) RA
- “CA Operator” (“CAO”) – the signing module
- “ARM” – advanced registration module – sort of an “automated RAO”

Schematics


CA
CMP
CAO
SQL
dB
Cert Status Service
CMP
RA
SQL
dB
SQL
RAX
Web RAO
ARM
Web interface
User

uniCert, additional comments

- Can modify contents of certificates easily
- Point-and-click CA “policies” – also very easy to manage sub-CAs with different policies
- Can have different policies for different RAs
- Can do automatic renewal (on old keys)
- Cannot do automatic re-key (i.e. re-key is like initial request – have to go through RAs again)

Baltimore Tech

- I quote: “Full development roadmap and commitment”
- Standard protocols used whenever possible (CMP, OCSP, LDAP, SQL) – not for RAO, though
- 30 day evaluation licence available
    - (of course this requires 30 consecutive days of my time…)

uniCert in e-Science?

- We decided not to evaluate it for now…
- …too much work to migrate from existing solution (uniCert mostly assumes you start from scratch)
- …too much work to adopt “weird” UK namespace requirements (OU and L identify RA) – may be possible with ARM but will probably be a lot of work

Part two

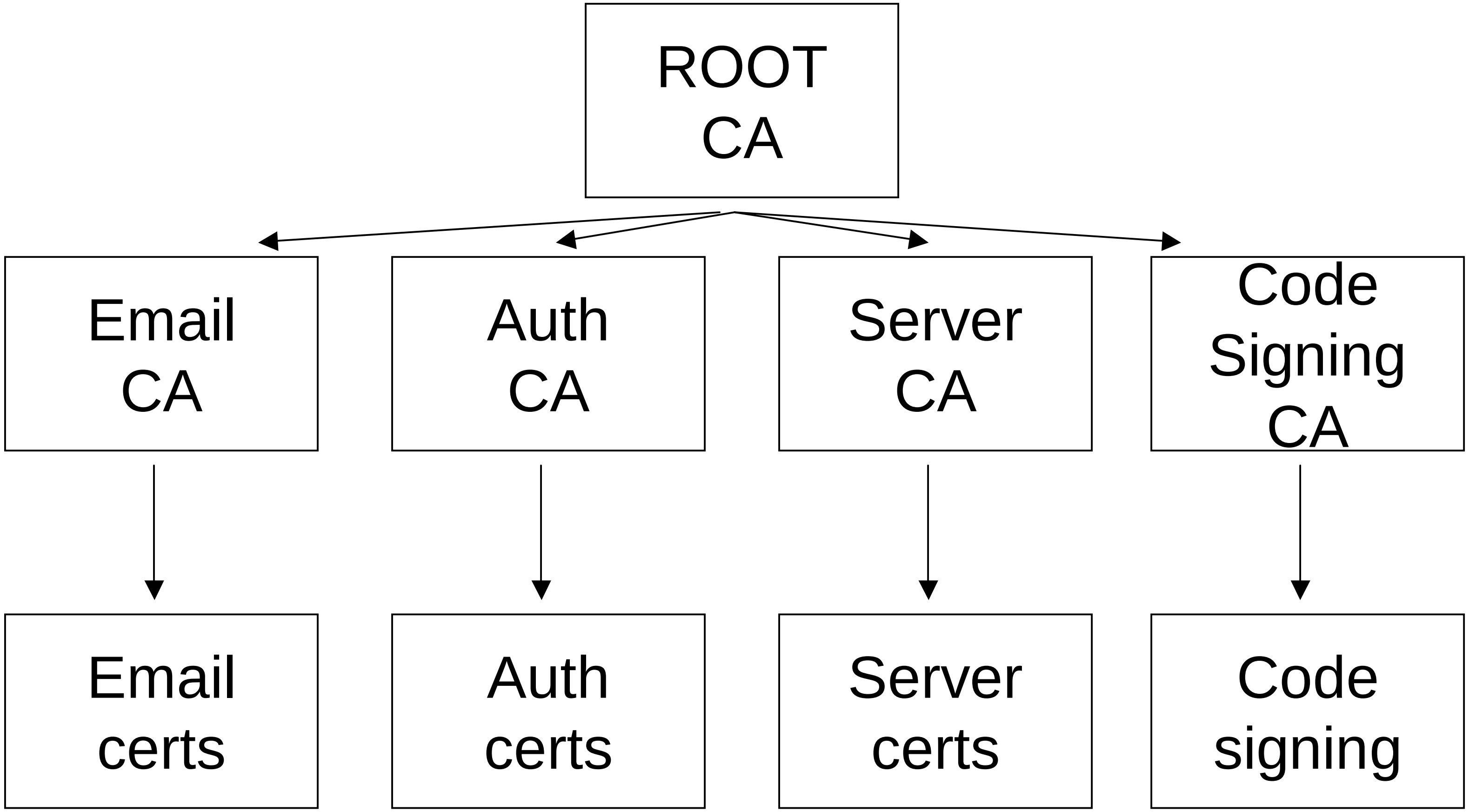
pyCA

Jens G Jensen
UK e-Science

Overview

- Written in python
- Runs as CGI programs under Apache
- Front end to OpenSSL
- LDAP support
- http://www.pyca.de/
- Not being actively developed at the moment – the author "does not have time but will bugfix"

(Default) Certificate Hierarchy



Part three

UK e-Science
Java solution

Jens G Jensen
UK e-Science

Overview

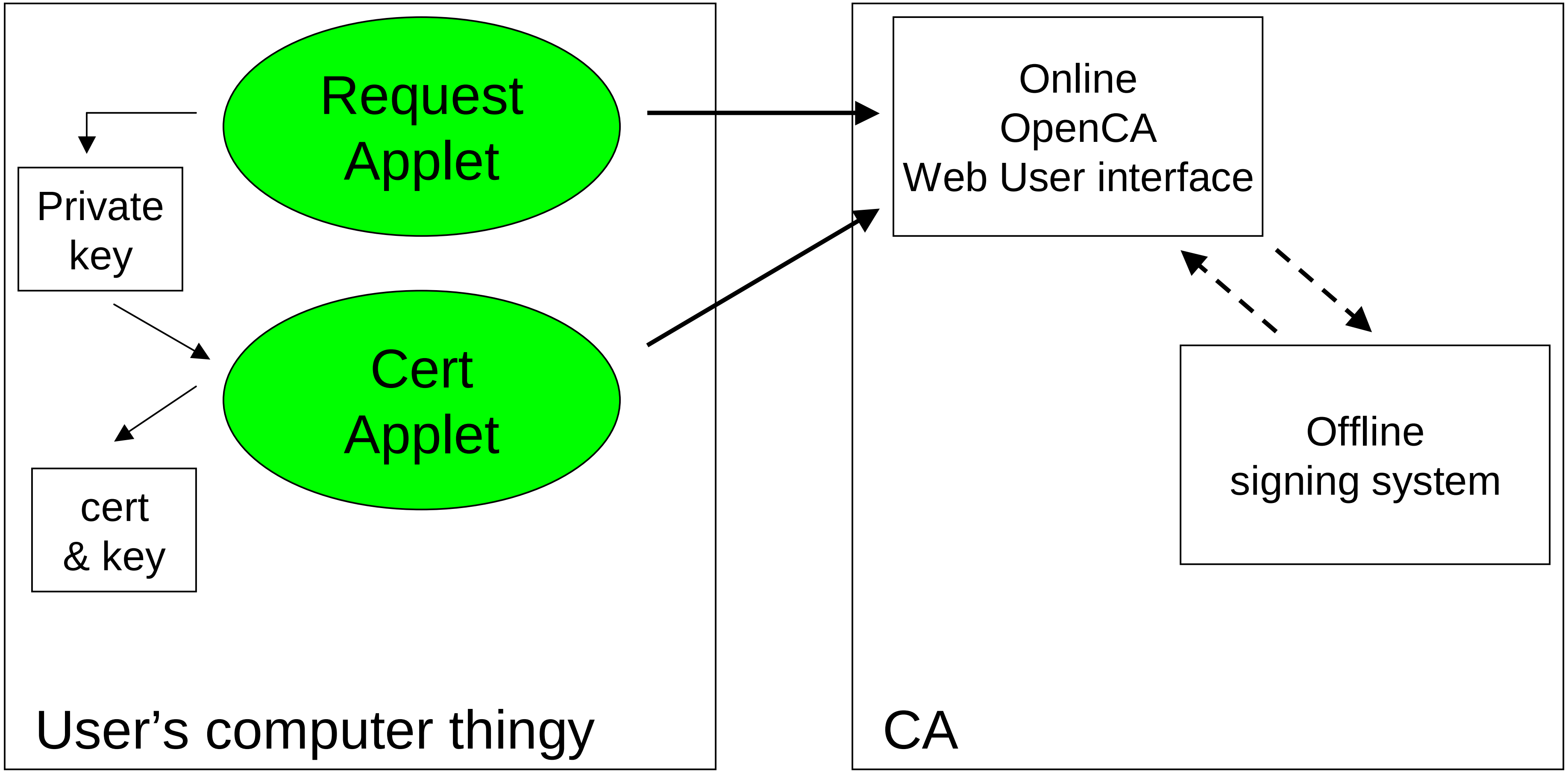
- Submits request to our current OpenCA system
- Written in Java as *signed applets*
- Crypto based on the `BouncyCastle` and `jcetaglib` libraries

  http://www.bouncycastle.org/

  http://jcetaglib.sourceforge.net/

- Still under development

Obligatory Diagram



Jens G Jensen
UK e-Science

PCKS#12

- Problems using `KeyStore` class from applet – not from java application
  - Applet complains of invalid signature on provider
  - Problem is with JCE 1.4, works with 1.3
- The `KeyStore` class is used to generate the PKCS#12 file

Browser support

- Browsers generally come equipped with JCE 1.1 or similar
- Currently users must install 1.4

Portability

- Not very…
- Written to take some of e-Science’s peculiarities into account
  - Namespace: OU and L, requirements on name forms
- Written to submit requests into OpenCA
- In the (near) future, can provide more generally useful CA software

Future developments

- Need to review the code, and clean it up
- Can replace OpenCA: since applets provide the user friendly interface, no need for OpenCA
  - Plan to replace system with a simpler Apache/mod_ssl/Perl-CGI/OpenSSL system using a PostgreSQL database
- Produce general non-eScience software?