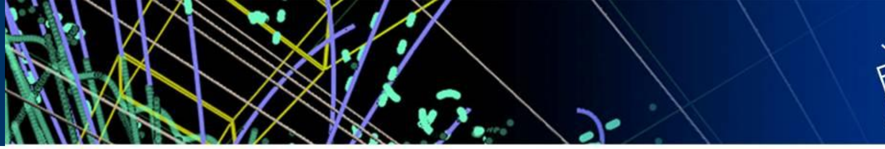
Identity Management

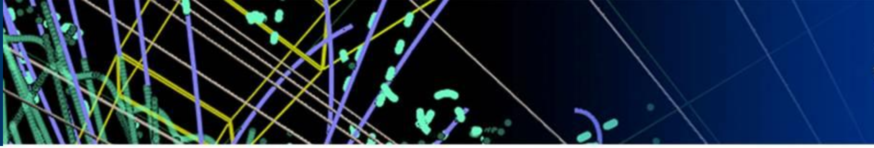
J Jensen, STFC

hepsysman, June 2017



Contents

- Background – federated identity management
- Certificates (hosts, in particular)
- Pathfinder
- RCauth
- Future



Fed. IdM

- SSO: single account, used everywhere
- Single login: only need to log in once
- Federated identity: identity comes from home IdP
  - E.g. UK Access Management Federation,
  - eduroam,
  - eduGAIN “superfederation”,
  - Assent
- AARC project “blueprint architecture”
  - Login is typically via portal but there is (some) tech for CLI

Host Certificates

Brian: - you are all individuals!
Crowd (in unison) - we are all individuals!
Lone voice: - I’m not!

Certificates are issued to a private key
The private must not be shared
… but the

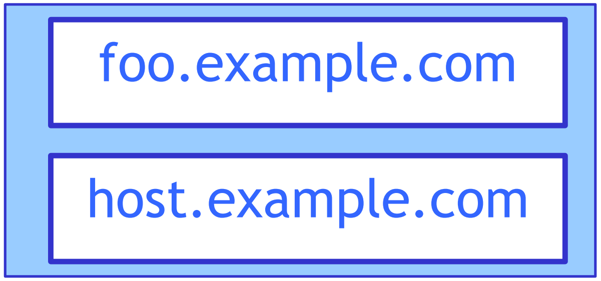
Multiple SANs

- A certificate contains:
  - A *subject name* (which names the subject)
    - The DN
    - For hosts, the CN is the hostname
  - *Subject alternative name(s)* – hereinafter “SAN” – which also name the subject
    - E.g. email address(es)
    - Host name(es)
    - IP address(es)

Host Naming

- Example: host.example.com. has alias foo.example.com.
- The client accesses foo.example.com.
- The certificate is (usually) issued to the alias

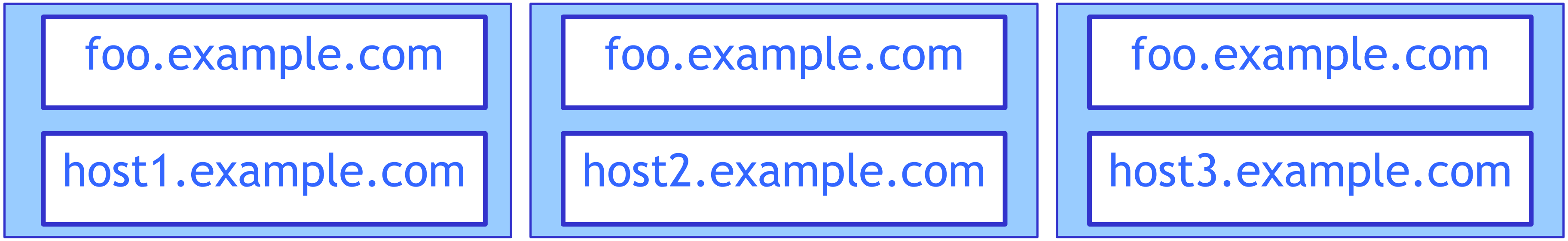
- RFC 2818, section 3.1 (server identity):
  - Client MUST check hostname (in URL) against server identity (in cert)
- Globus (till recently) didn’t need this because it resolved the names through DNS
  - However, that is a security risk since DNS could be tampered with



… so the certificate MUST be issued to the alias…

Naming Hosts

- One “server” spread across multiple hosts
- Client calls – and sees only – foo.example.com
- Each host has an individual CNAME
- The certificate is issued to the CNAMEs because they are all individuals
- (Instead of getting a cert for foo.example.com. and copying it to all)
- (or 3 certs for foo.example.com. which is slightly less bad but still bad)



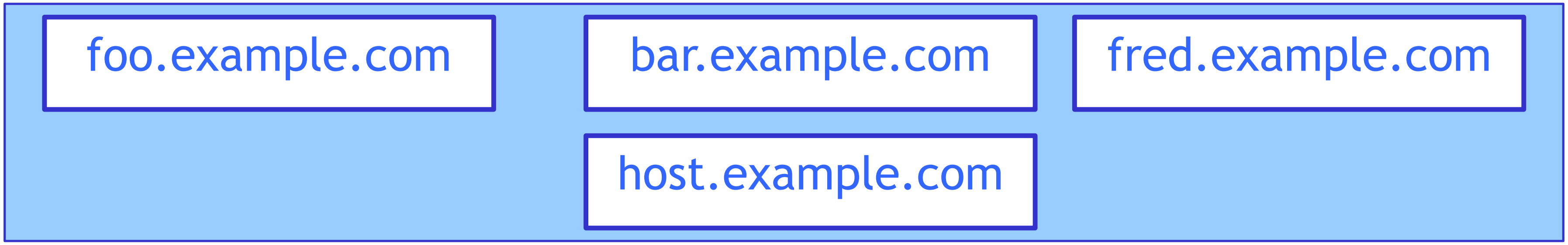
foo.example.com
host1.example.com

foo.example.com
host2.example.com

foo.example.com
host3.example.com

Naming Hosts

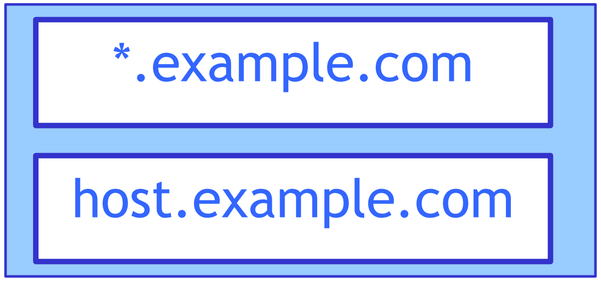
- Multiple servers on a single host
- A single host fronts multiple hosts
- Certificate is typically issued to the CNAME



Example: srm-gen.gridpp.rl.ac.uk.

Naming Hosts

- Multiple servers on a single host
- A single host fronts multiple hosts
- Certificate is issued with wildcard
- Matches {foo,bar,fred}.example.com.
- But also mail.example.com.
- Should be used with some caution (if at all)



Back to RFC 2818

- Client MAY perform its own non-std check
  - E.g. check the server’s key
- If not, client extracts SANs
  - Checks if one matches the requested hostname
  - or IP address
- Iff there are no SANs, client checks CN
  - If no alternative names are present

Supporting Multiple SANs (in Host Certs)

- Currently:
  - A 500 line Perl script (lots of error checking)
  - Which “edits” the request
  - Must be run by CA operator
  - Before the certificate is signed, ideally
- Status
  - Error prone: both CA op and requester must coordinate
  - RA op may not see alternative names

Supporting Multiple SANs (in Host Certs)

- Requester adds requested SANs to request
  - See GridPP Wiki for instructions
  - Must have the name in CN (typically CNAME) also in SAN (conventionally the first SAN)
  - Have to use PeCR to submit – JK working on this
  - Must use PKCS#10
- RA op sees extra SANs and approves them
  - …hopefully
- Signing system honours extra SANs?
  - You can put anything in a request
  - But by default almost everything is ignored!

Signing Multi-SAN Requests

- Introduces extra risk
  - Typically, someone sneaking in a dodgy name
  - Or applies for something bad in good(ish) faith
- Implementation
  - Extracted and checked, and added to cert
  - Instead of the ‘copy’ and ‘copyall’ semantics of openssl ca
- Help to introduce rules…?
  - Maybe restrict to “trusted” admins (by DN) – would only work for *new* reqs.
  - No bulk…?
- Wildcards need checking (e.g. *.example.com)
  - Restricting domains (by admin)

Renewing Multi-SAN certs

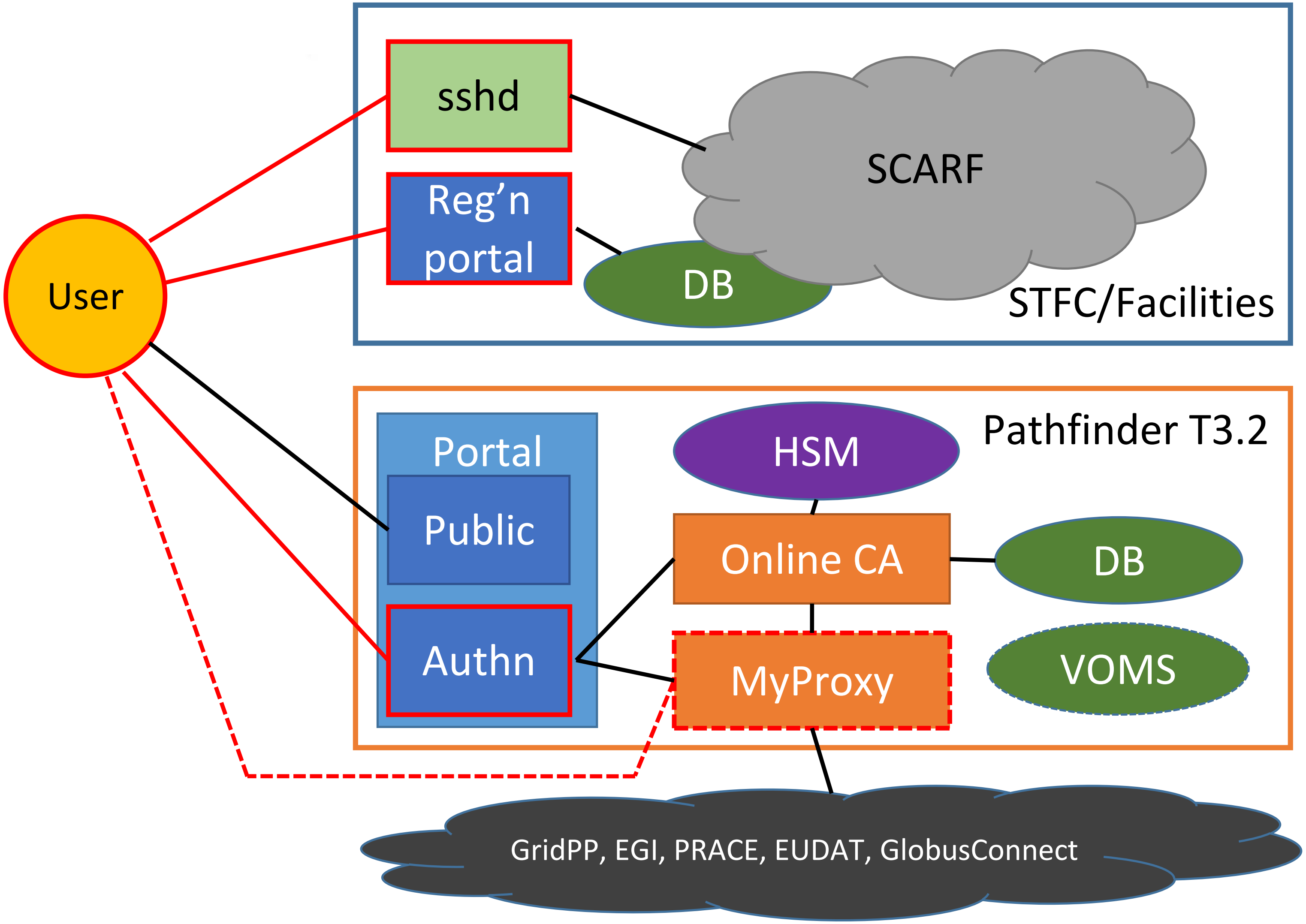
- (Even) less mature
  - Not much time/effort to develop CA
  - Needs to copy stuff over from previous
    - Some exploratory work in this area but less mature
    - Some times changes are need (= CCRs)
    - (Certificate Change Request)
  - With PeCR it may be just as easy to apply for new?

Further Reading

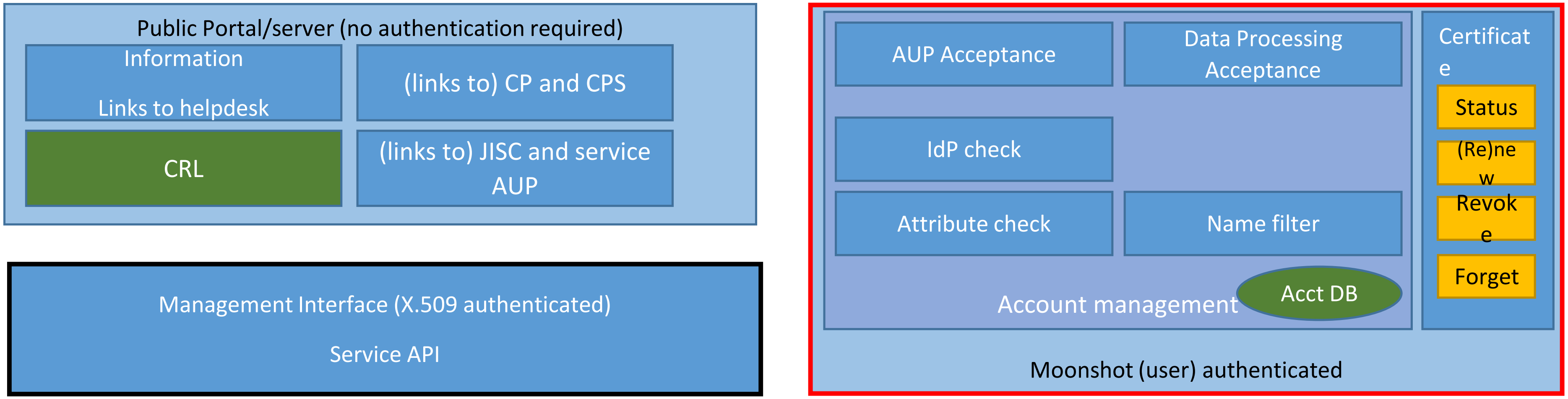
- RFC 2818
  - http://www.rfc-editor.org/rfc/rfc2818.txt
- RFC 5280
  - http://www.rfc-editor.org/rfc/rfc5280.txt
- GFD.225
  - http://www.ogf.org/documents/GFD.225.pdf

Federated access to certificates

PATHFINDER MICS RCAUTH IOTA


sshd
Reg'n portal
SCARF
DB
STFC/Facilities
User
Portal
Public
Authn
HSM
Pathfinder T3.2
Online CA
DB
MyProxy
VOMS
GridPP, EGI, PRACE, EUDAT, GlobusConnect

Front End(s)



Public Portal/server (no authentication required)

Information

Links to helpdesk

(links to) CP and CPS

CRL

(links to) JISC and service AUP

Management Interface (X.509 authenticated)

Service API

AUP Acceptance

Data Processing Acceptance

IdP check

Attribute check

Name filter

Account management

Acct DB

Certificate

Status

(Re)new

Revoke

Forget

Moonshot (user) authenticated

Red outline = Moonshot authenticated
Black outline = certificate authenticated

RCauth

- CA set up by NIKHEF as an AARC pilot activity
- Allows selected IdPs to get certs à la SARoNGS
- Will be run by EGI and EUDAT in EINFRA-12
  - The EUDAT one run by STFC, EGI by GRNET
- IOTA profile: OK for WLCG…
  - Pathfinder will be MICS (hopefully)

THANKS