Authentication and Authorisation for Research and Collaboration

The AARC 'CILogon-like' Pilot

using an IOTA CA and Credential Store for Europe

David Groep
AARC NA3 Activity Lead
Nikhef, Physics Data Processing Group

EUGridPMA 36 Bratislava meeting
18 January 2016

Some AARC Facts

Authentication and Authorisation for Research and Collaboration

- Two-year EC-funded project
- 20 partners
  - NRENs, e-Infrastructure providers and Libraries as equal partners
- About 3M euro budget
- Starting date 1st May, 2015
- https://aarc-project.eu/

AARC Vision and Objectives

Improve federated access by addressing current challenges

Integrate existing R&E AAIs to create a highway for identities

Avoid the creation of project-specific AAIs by enabling researchers to use their existing credentials to access different resources

Harmonise policies among e-Infrastructures to ease service delivery

Define a training package for institutions and services to support federated access

AARC Structure

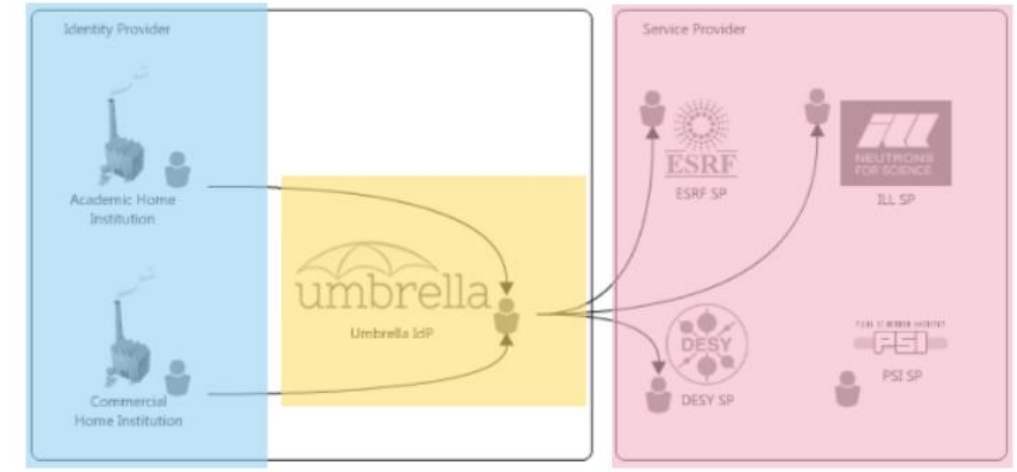
AARC Pilots and the EGI AAI evolution

- Most infrastructures move to completely hiding PKIX from the end-user
  - “we’ve found all people in the world who understand PKI” (and they by now all got a certificate ;-)
  - EEPKI + RFC3820 proxies do solve both the CLI and delegation use case rather nicely

- Bridging and translation is the pragmatic approach
  - Does not require major technical changes in existing R&E federations
  - Allows for community-centric identities-of-last-resort (or first resort, for that matter!)
  - Time line is more predictable, because fewer entities are involved –
    and those entities have a stake in and the benefits off the results

- Emerging as a pattern in many Research Infrastructures that use CLI or brokerage
  - ELIXIR, UMBRELLA, WLCG, INDIGO DC
  - SAML->OIDC, SAML->X509, X509->OIDC, X509->SAML, OIDC->X509, …

Example: Umbrella



Marcus Hardt, KIT and AARC

https://aarc-project.eu/wp-content/uploads/2015/11/20151102-JRA1.2-Blueprint-Status-Hardt.pdf

Translation services – an overview

CILogon Service

NCSA (IL,USA) operated service and project

InCommon backed MICS and IOTA

Generic ‘opaque’ certificate in Europe

Helps with PII data protection and integration with ESFRIs and e-Infrastructure

CERN wLCG IOTA CA

eduGAIN backed with added CERN HR DB controls

GÉANT

GEANT Trusted Certificate Service TCS

could be turned into a translation service, when each subscriber would enable that since it has a subscriber-centric validation model

AARC engagement process: operational pilots

- AARC *will not operate* any long-term services (that's for GEANT, EGI, PRACE, EUDAT …)
- But *will pilot* actual technology combinations that are useful to (research) communities

Proposal-identified pre-pilot: certificate-less access to existing services with "CILogon for Europe"

- Driven by Mischa Sallé and Tamas Balogh (Nikhef)
- Aligned with the EGI "JRA1" activity around the evolution of the AAI technology (ChristosK)
- Using actual use cases from EGI competence centres and AARC communities

"It's always a challenge to pilot something with a real community – the expectations are usually much higher than what can be provided in a pilot …"

AARC SA1 “CIlogon-like Pre-Pilot”

Establish a CILogon (like) service in Europe

- Integrated closely with R&E federation landscape (with all of full-mesh, H&S, mixed-models)
- Integration with user community services and attribute services
- Close co-operation with the CILogon Project (Jim Basney et al.)

Pre-pilot work, so based on pre-AARC requirements gathering

- FIM4R requests, alignment with known user communities (EGI evolution, ELIXIR)
- Potential to support the EGI ENGAGE community ‘competence centre’ work
- Leveraging existing components and services: CIlogon + ‘OAuth4MyProxy’ components, VOMS Attribute Certificate service, OIDC libraries, …
- Try to fit first in the existing policy framework: Approved Robots (and “PUSPs”), Trusted Credential Stores, PKP Guidelines, IGTF ‘DOGWOOD’ – unless the pilot runs aground …

Desired features set

- Certificate or proxy retrieval possible for federatively-authenticated end-user inside a community (VO) portal or science gateway
- Work with the existing (SAML2) R&E federations
- Credential repository feature: manage credentials on behalf of the user
- Provide – on the user's request – delegated credentials to science gateways
- Make end-user facing science gateways *really* light-weight: VOs should not need to know about protecting long-term secrets (and need a way to authenticate users)
- Support both certificate and non-certificate science gateway use cases in the same way
- Provide simple way for users to obtain 'opaque' CLI credentials (proxies) on their own system

Constraints:

- No new software components (only limited glue)
- Deployable in a scalable way – with a sustainability model behind it
- As few CAs as possible (preferably: just one)

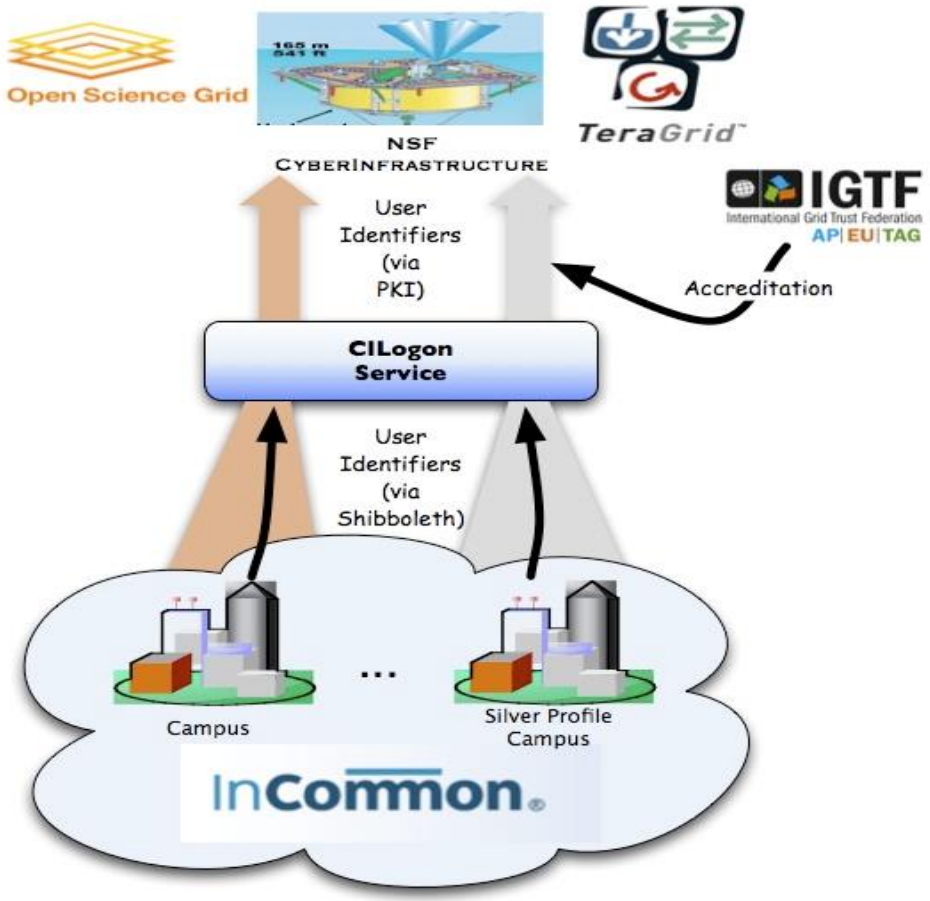
CILogon service and project (Jim Basney et al.)

CILogon Service

- Enable campus logon to CyberInfrastructure (CI)
  - Use researchers' existing security credentials at their home institution
  - Ease credential management for researchers and CI providers

Multiple interfaces

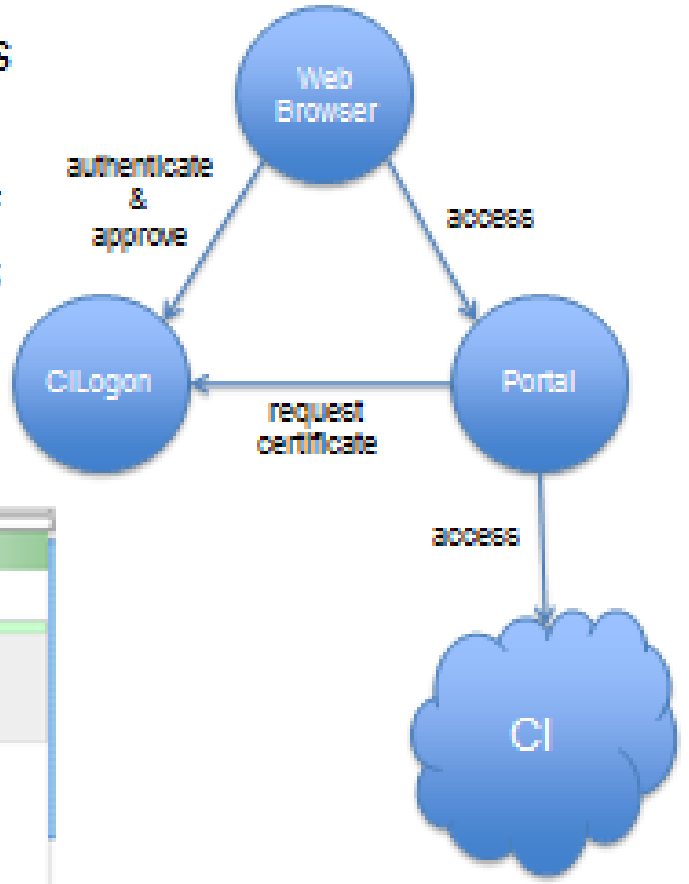
- SAML/OpenID Web Browser SSO
  - PKCS12 certificate download
  - Certificate issuance via OAuth
  - OpenID Connect token issuance
  - SAML ECP for CLI issuance



Slide content: Jim Basney, based on http://www.cilogon.org/docs/201106-cilogon-cern.pptx?attredirects=0&d=1 and http://www.cilogon.org/docs/20141030-basney-cilogon.pdf?attredirects=0&d=1

CILogon Portal Delegation

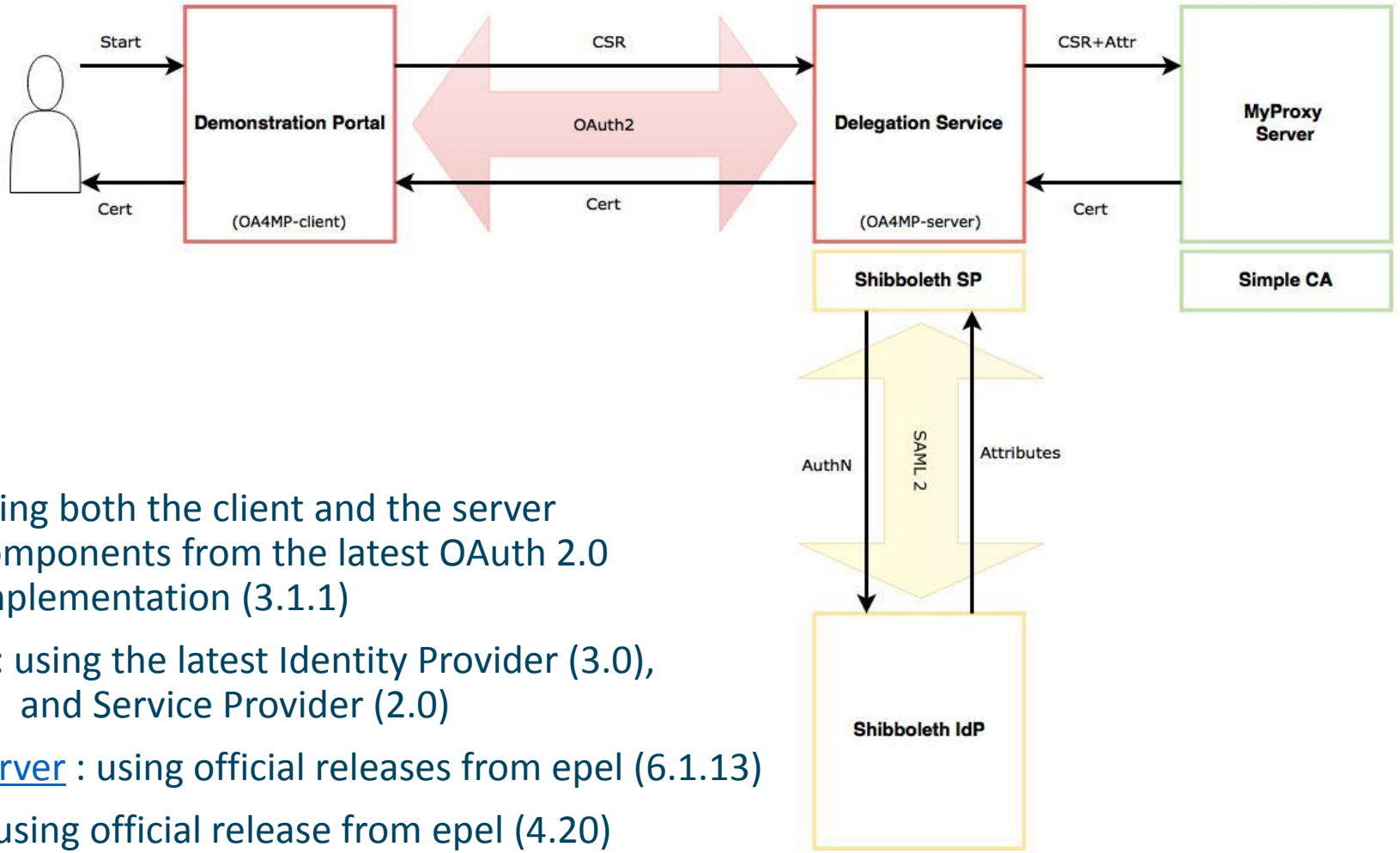
- Grid Portals and Science Gateways provide web interfaces to CI
  - Portals/Gateways need certificates to access CI on researchers' behalf
- CILogon Delegation Service allows researchers to approve certificate issuance to portals (via **OAuth**)
- www.cilogon.org/portal-delegation



Slide: Jim Basney, NCSA and CILogon

CILogon www.cilogon.org

CILogon demo portal: Delegation of credentials using OAuth4MyProxy



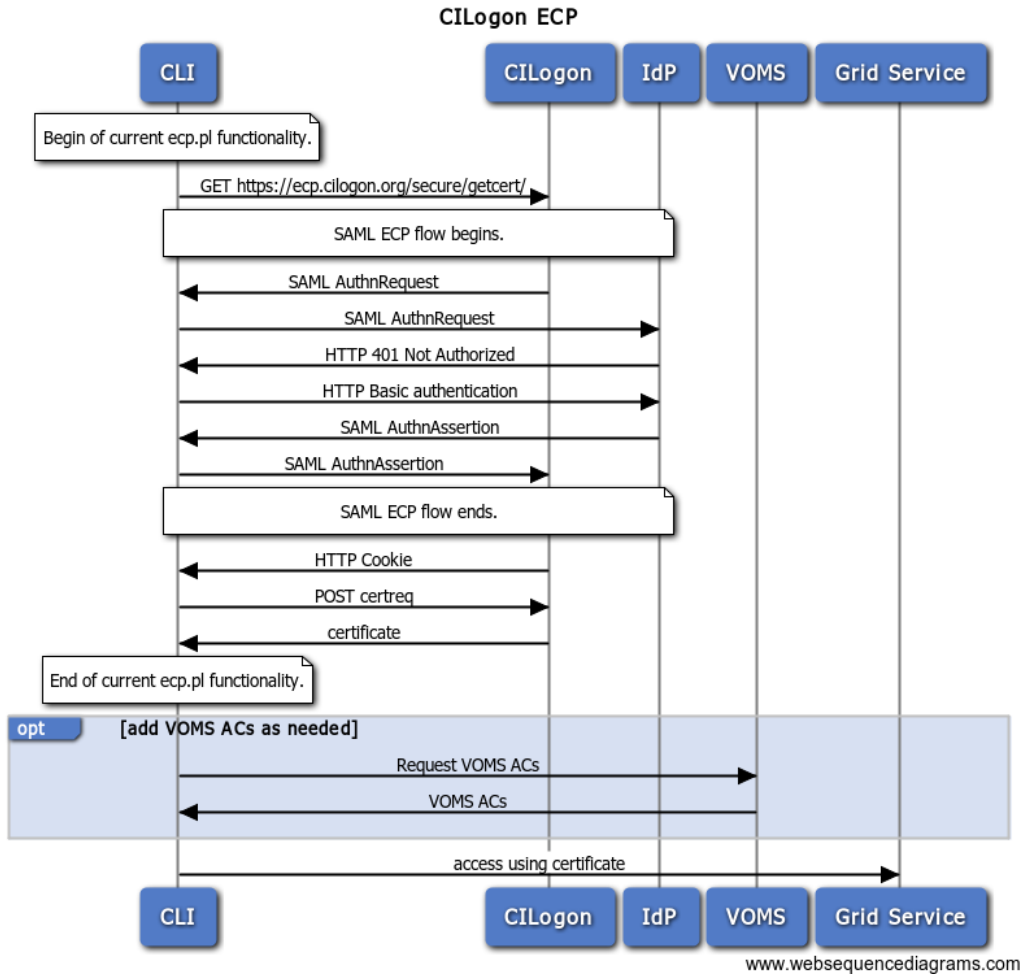
- OA4MP : using both the client and the server components from the latest OAuth 2.0 implementation (3.1.1)
- Shibboleth : using the latest Identity Provider (3.0), and Service Provider (2.0)
- MyProxy Server : using official releases from epel (6.1.13)
- SimpleCA : using official release from epel (4.20)

CILogon and SAML ECP

- SAML federated login via ‘Enhanced Client’ (read: CLI) or ‘Proxy’ (read: brokered access)
- https://wiki.oasis-open.org/security/SAML2EnhancedClientProfile

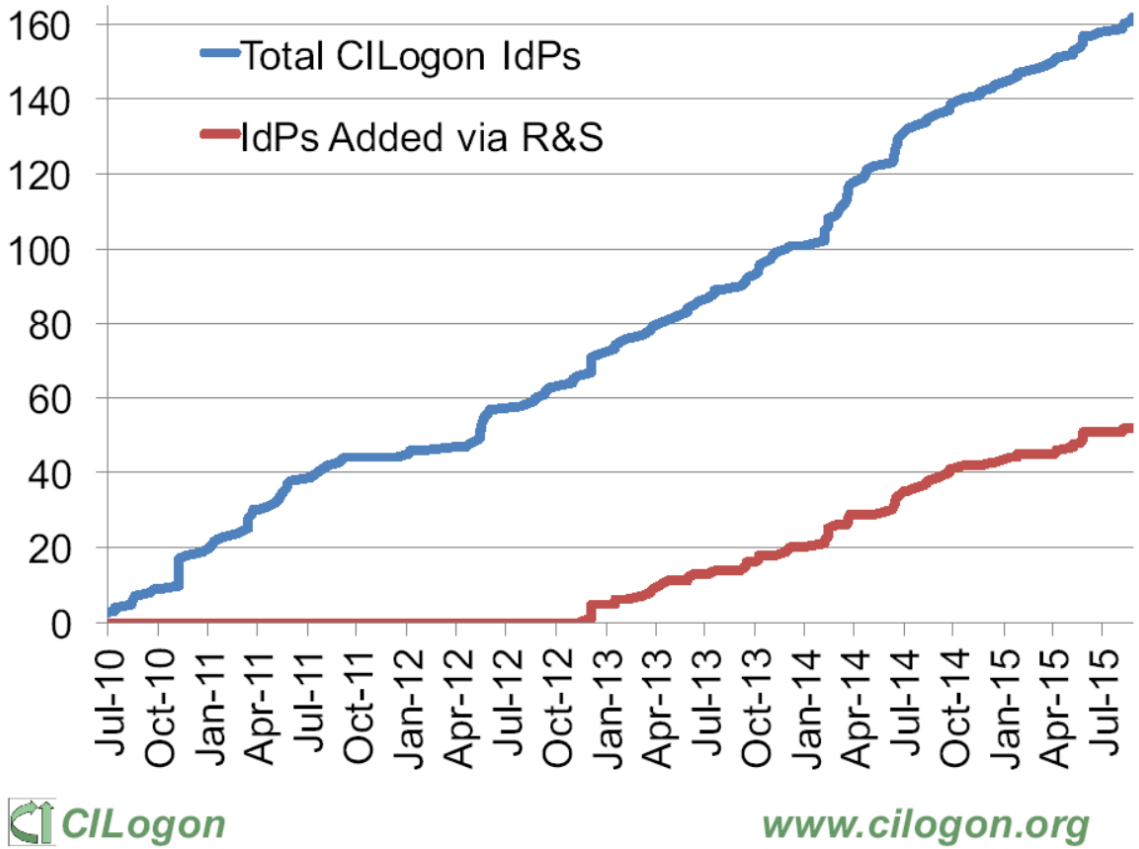
- A heavy (**trusted**) client sends credentials and receives assertions from a specific IdP ECP end-point
- Supports non-web
  … if it were supported by the IdP
- Most prominent use case: Office365
- Limited update & support (only in Shib, and only v3+)

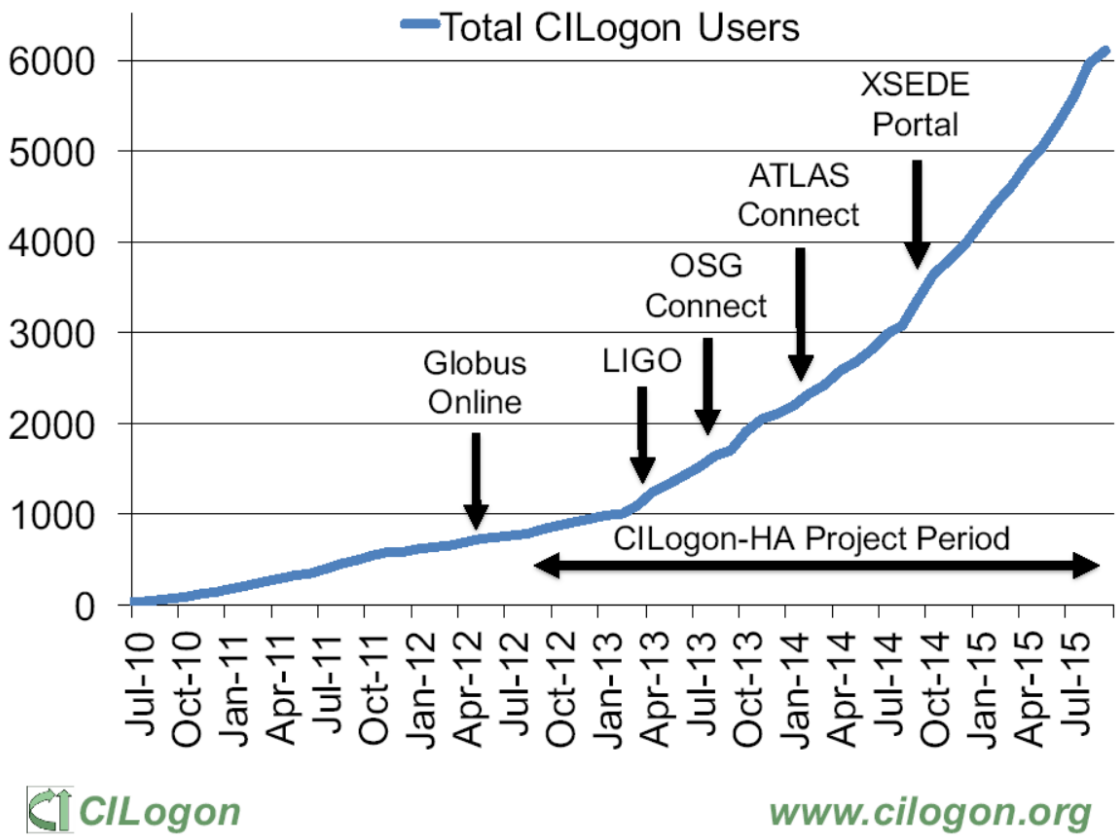
Unlikely to fly in Europe – but there many alternatives like Moonshot, but also a move to OIDC, OAuth2, …



Flowchart: from CILogon

CILogon adoption in the US/InCommon





http://www.cilogon.org/docs/2015-09-cilogon-ha.pdf?attredirects=0&d=1

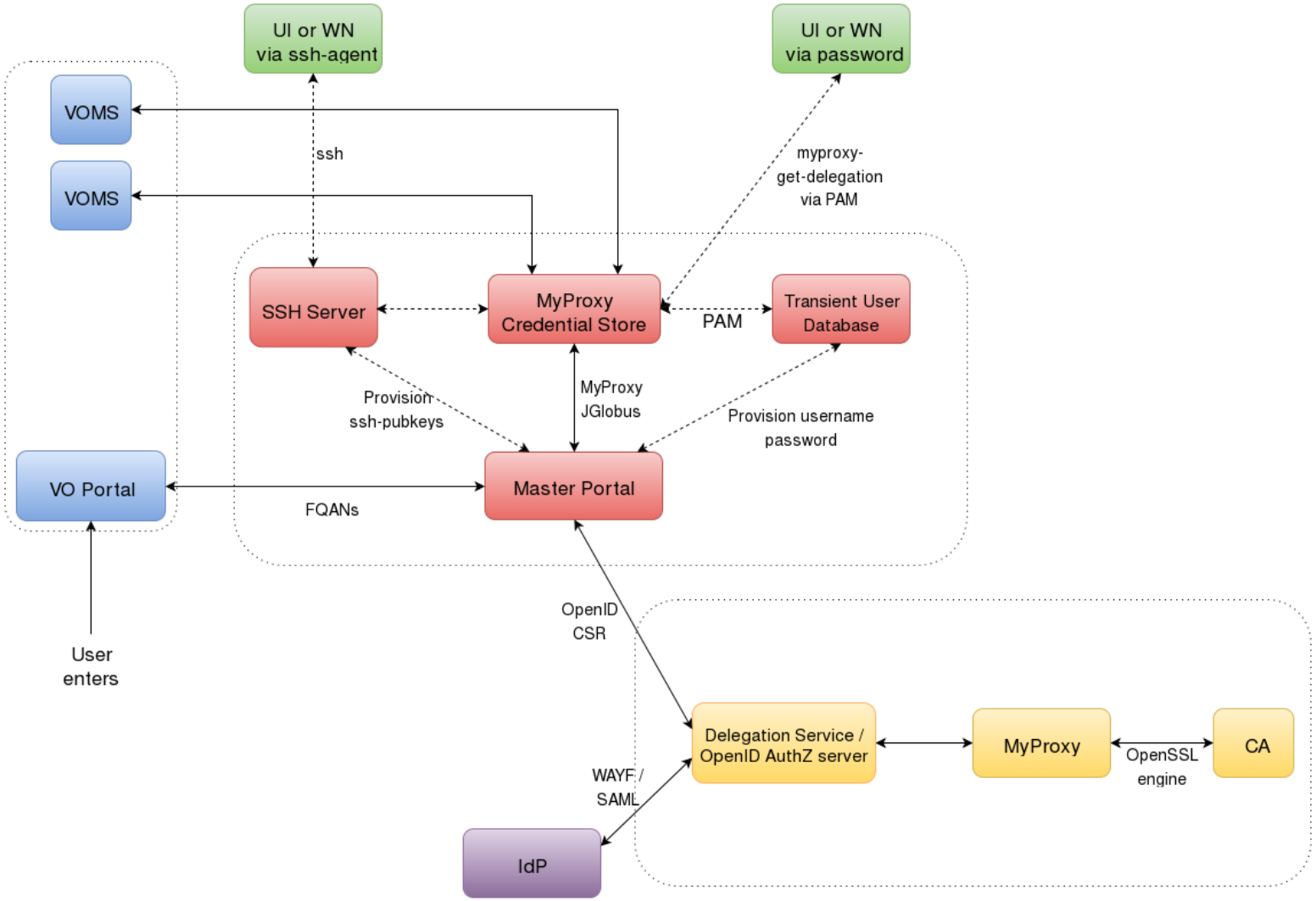
End-user credential hiding in the AARC CILogon-like Pilot

- Do not assume any changes in the IdPs: no ECP, no new policies, no nothing (reality, sorry!)
- Assume no major changes in the e-Infrastructures: interfaces remain a mix of Web and PKIX, policies remain mostly as-is
- Should show results ‘fairly soon’ (i.e. in a few months work) for a broad audience
- Leverage existing CILogon and MyProxy, thanks to the collaboration of AARC-CTSC/MyProxy

Beyond CILogon

- CILogon assumes the e-Infrastructures (CIs) build the portals and interfaces
- CILogon assumes that users in the end might retrieve certificates explicitly
- Larger RIs and e-Infra in Europe could do it, but not the large number of small communities
- So the AARC Pilots adds additional control elements: credential management, light-weight portal interfacing, (VOMS) attribute management, *optional: opaque credential retrieval*

Components



http://wiki.nikhef.nl/grid/CILogon_Pre-Pilot_Work

Authorization at the VO level

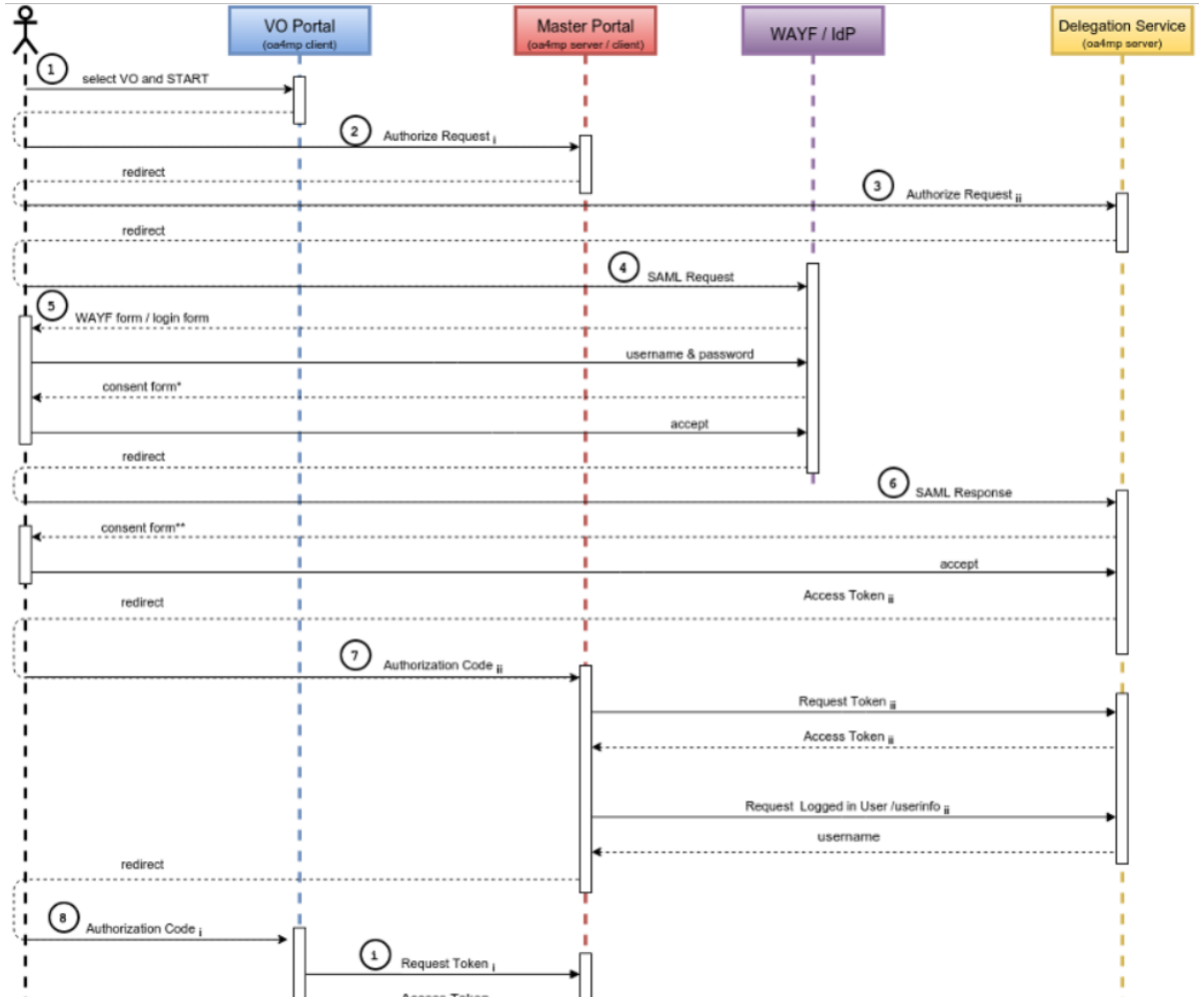
- The VO light-weight portals (gateways) can re-use this system for both AuthN and AuthZ
- Can be used besides a conventional (SAML) login to science gateway when a proxy is needed

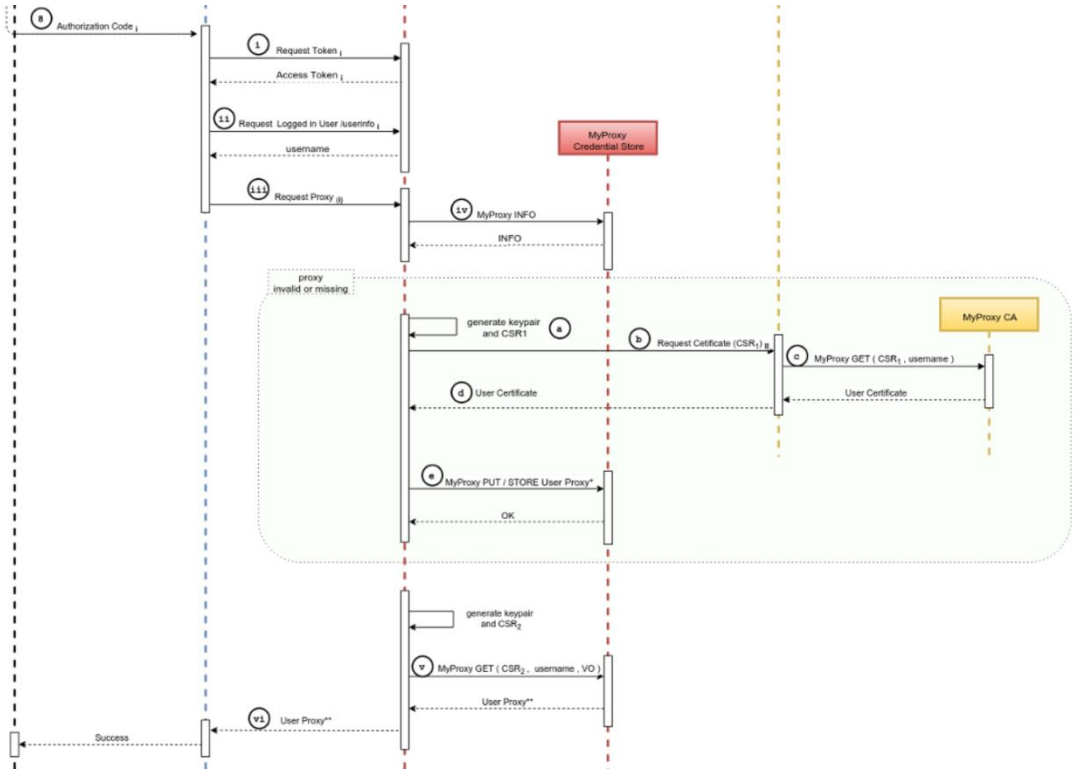
Or even …

- *User ability to complete the OIDC login to the VO web portal (each time) does AuthN*
- *Ability of the portal to successfully request VOMS attributes for an AuthZ/membership check*
- *Successful authN and failed AuthZ? Suggest enrolment or auto-enrol members!*

- VO portal must be on a trusted list of the Master Portal
  - Needs to be able to do OIDC in a trusted way
  - Using a VO portal *client ID + client secret* (but there are server certs as well for the web site itself)
  - User must be able to trust that the Master Portal will only relinquish user credentials to intended places
  - OIDC consent mechanism informs the user of where the user credentials are sent

It’s a complex flow … because of a double OIDC + SAML + Online CA





X509 (proxy) certificates as opaque access tokens

- VO or Master Portal can offer user to register user's SSH pubkey with Master Portal
- Master Portal can store (uid, pubkey) pairs in a directory service (e.g. LDAP) associated with the MyProxy user identifier (username)
- SSH Server runs cron job and creates authorized_keys file:
  - Using a single special account, runs a *myproxy-logon* wrapper, similar to what SVN servers do*
- SSH-agent forwarding: central login node, client UI, VDI server, laptop retrieves proxy
- Any script wrapper to save proxy: looks similar to a Kerberos ticket
  - No need for either extra password, ECP, Moonshot etc.
  - Very similar to GitHub, SourceForge, etc.

```
* /usr/local/sbin/mkgroup-sshlpk \
    -c 'command="svnserve -t -r /srv/svn --tunnel-user=@UID@", no-pty' \
    -o ~svn/.ssh/authorized_keys --filter '(authorizedService=ndpfsvn)' nDPFSubversionUsers
... [gives] ...
command="svnserve -t -r /srv/svn --tunnel-user=tsuerink",no-pty ssh-rsa AAAAB3...2w== t@net
```

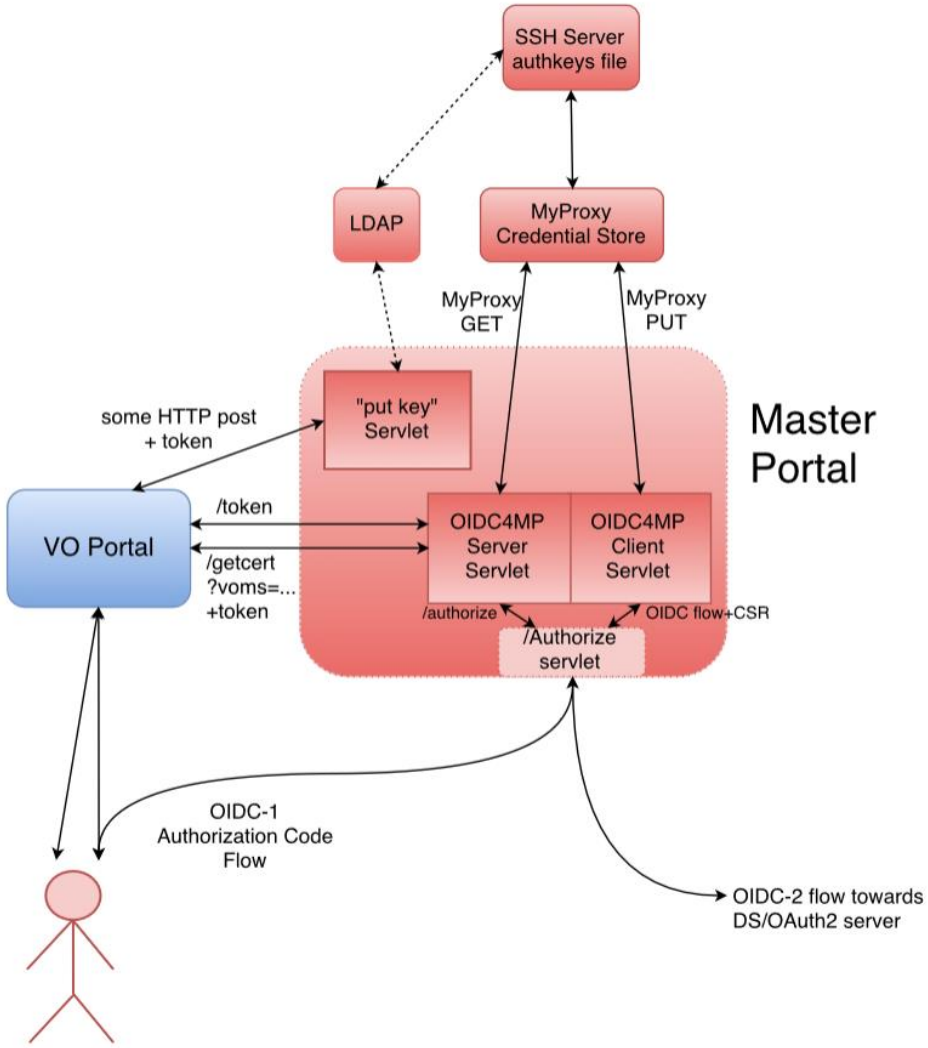
Master portal is a rather critical service

- This is the component that – with a credential store and an (OIDC) authentication interface – takes care of the user credential management

- The back-end CA provides
  - Identifier uniqueness
  - Revocation capability

  but not much more!

- It is a highly trusted component, of which there should not be many
- But it may still be better then end-user managed keys – for *authentication*, that is …



Distribution of Roles in a Sustainable Model

VO Portal (Science GW)

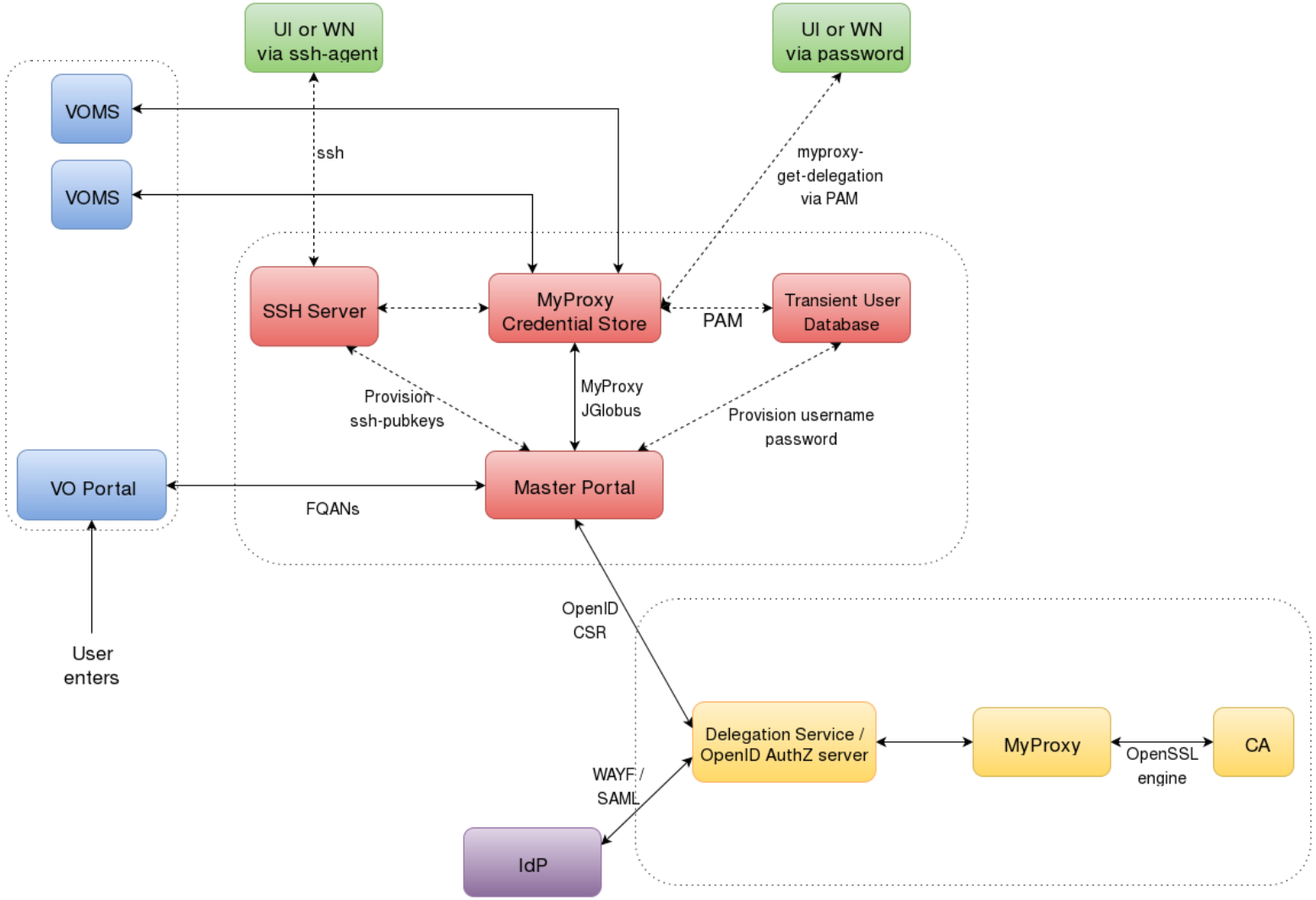
- One per application
- Many deployed throughout
- Reduced policy and compliance burden

Master Portal

- One per country, ESFRI
- Must be well-managed
- Can be managed because there are few

CA and Delegation Service

- As few as possible: just one!



Relevant (IGTF) policies

- IOTA AP: LoA DOGWOOD and PKI Technology Guidelines
  - https://www.igtf.net/ap/iota
  - https://www.igtf.net/ap/loa
  - https://wiki.eugridpma.org/Main/PKITechnologyGuidelines
- PKP Guidelines
  - https://www.eugridpma.org/guidelines/pkp
- Guidelines on the Operation of Credential Stores (draft)
  - https://www.eugridpma.org/guidelines/trustedstores/

Towards a CILogon CA for Europe

As a first phase, Jim may actually just open up the existing CILogon to ‘eduGAIN’

- Once InCommon has also technically joined eduGAIN
- For qualified entities in eduGAIN: R&S + SirTFi
- Uniqueness enforced by the CILogon CA itself (as long as there’s no true ‘ePUniqueID’)

Aim for (a single) IOTA CA in Europe (EU/EEA) to back the Master Portals

- This would be a generic IOTA CA, but it can be modeled closely on the existing ones
- Model yet to be worked out (extend CERN’s IOTA CA? A new one?)
- Support as many (European) eduGAIN IdP as feasible
- Potentially including qualified *IdPs of last resort* operated by RI/e-Infrastructures, or qualified proxy services like the VOPaaS IdP gateway (with LoA support)
- Having it issue only short-lived credentials would make things like DPCoCo compliance easier

Operating an generic European IOTA CA

- It is a SPOF, so needs a high service level
- Should be within the EU to ease transfer of personal data
- Needs a sustainability model

Candidates to run this?

- joint e-Infrastructure operation?
- source it to a dedicated company under a strong SLA + PII protection?

'Worst case' would be to get one per country … and we still need a business model

- In AARC DAASI is tasked to research possible sustainability & operating models

Protecting credentials, CS Operators

http://wiki.eugridpma.org/Main/CredStoreOperationsGuideline

- data needed to activate and use credential material must not be held by the system on persistent storage, and must not be held by the system administrators. These must only be present in the system as a result of a user action, and only for as long as the user is using the system.
- The activation data and any plain text private keys should be removed as soon as the user stops using the service, and should not be kept past 24 hours of inactivity exclusive use of confidential, integrity protected, and authenticated channels for the transfer of activation data and any private key material.
- The keys used to protect the channel must have a strength equivalent to or better than an 2048 bit RSA key.
- The keys must be suitably protected by the operating system or an HSM, and must only be accessible by the service and trained personnel with procedural controls.

PKP Guidelines

Section: Generation of private keys

- A system SHOULD NOT persistently keep pass phrases or plain text private keys for longer than 24 hours, unless the key pair is used solely as a basis for Short Lived credentials, i.e. the certificate has a total validity of less than 1 Ms.

  - “This text is written such that it allows for a portal to request a certificate on the user’s behalf (e.g. by redirecting the users to a, potentially federated, SLCS service) and keep the key material in the portal. To off-set the risk of keeping unencrypted private keys on disk for long periods of time, the mechanism as used by, e.g., the ssh-agent system is intended to be used for protection: The portal can itself encrypt it with some other pass phrase and store the key on disk, but keep the (portal-private) activation data to re-read the private key only in-memory (so that it becomes a lot harder to sniff in case the box is broken, in the same way that ssh-agent does it and for the same reasons).”

PKP Guidelines

But in the “Storage of key material” section …

- Data needed to decrypt or use the private key MUST not be held by the system on persistent storage, and MUST not be held by the system administrators. It MUST only be present in the system as a result of a user action, and only for as long as the user is using the system. The activation data and any plain text private keys SHOULD be removed as soon as the user stops using the service, and MUST NOT be kept past 24 hours of inactivity.

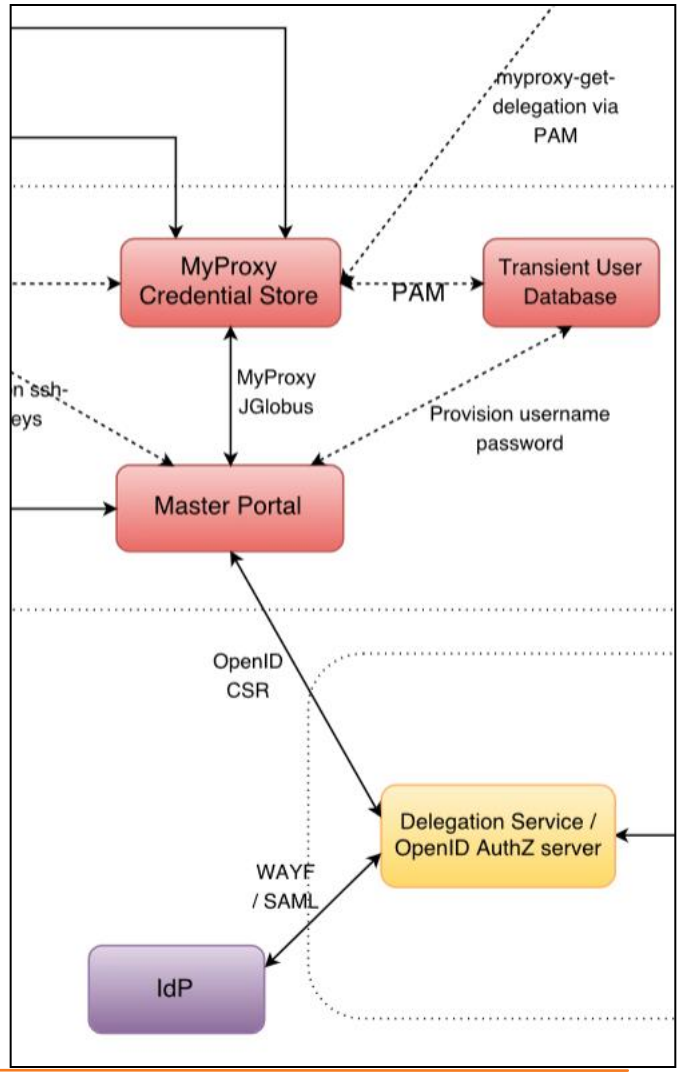
  - “This text specifically allows for long-running and multi-step work flows to continue in the absence of physical user presence at the portal. The word 'inactivity' should be interpreted as “if a user logs in and starts a long work flow at 3PM, leaves the portal and goes home at 5 PM, but the work flow completes only 48 hours after that, it is perfectly legitimate for this third-party system to hang on to the private key activation data in memory for 56 hours”. If we were to limit the caching of activation data to just 6 (or 24) hours after the user as stopped clicking on the portal (i.e. at 11PM), we would never get any real work done. But if the portal gets rebooted, the activation data is lost and the work flow will terminate once the pending proxies expire (after ~ 12-24 hrs). The 6 or 24-hour period is somewhat arbitrary, and should be synchronised to the characteristic ‘session expiration’ period for most portal applications.”

PKP Guidelines – the Challenge

The PKP Guidelines assume the user is present – *somewhere* in the workflow – to provide a unique secret (activation data) with which to protect the user's credential

But:

- in the entire federated workflow, *there is no such secret to be found*
- the SAML assertion, the OIDC access tokens, the authorization codes: all are generated by servers
- the one unique user secret is hidden – rightfully and only ever exposed to the federated IdP
- the only place from which to get a unique bit of private data close to the credential store … is a single common place: the Master Portal ☹



Building the Pilot

The AARC CILogon-like Pilot works around this in the ‘trivial’ way

- It requests a certificate when the user logs in via the portal (OIDC->OIDC->SAML/R&E)
- Generates a unique key pair in the Master Portal memory, making a CSR on behalf of the user
  OK, since *“Key material MUST only be generated in the system as a result of a user action”*
- It then delegates a *proxy* to the Master Credential Store (a protected MyProxy)
- And securely deletes the key pair in memory

By storing only a proxy in the Master Credential Repository it escapes the PKP and CS guidelines

Another alternative: replace the CA with a single Robot & ‘Per-User sub-proxies’ (PUSPs)

- PUSPs are already used by the EGI “Long Tail of Science” gateways
- RFC3820 proxy certificates generated from a single Approved Robot:
  - embedded in the naming is a unique identifier
  - the generator (portal) can associate the identifier with an individual Web User

```
"/C=IT/.../CN=Robot: Catania Science Gateway - Roberto Barbera/CN=user:jdoe" jdoe_localuser
"/C=IT/.../CN=Robot: Catania Science Gateway - Roberto Barbera/CN=*" .portal_users
```

- This can also replace the CA it ‘just’ take the portal setup outside the PKP scope

but … is this the best way to do things??

https://wiki.egi.eu/wiki/Fedcloud-tf:WorkGroups:Federated_AAI:per-user_sub-proxy

What is the ‘right’ way of doing this? Considerations

Can the user – with duly informed consent – actually delegate credential management?

Pros

- Good for usability and recoverability of user credentials (no separate passwords)
- Custodianship is clearly identified (at the master portal operator)
- Users are quite fed up with credential management and use *any* solution, so why not this?
- Short life time can help limit the risks

Cons

- Custodianship is clearly identified – who will want to run the master portal and take the risks?
- Master portal operator can act as the user –
- Short life time impairs the user for long-running jobs – automatic credential renewal is non-trivial since you need the user in the loop every time
  … and the master portal does not know the workflow

Short-lived credentials?

- Use of credentials with a life time < 1Ms (11days) does allow unencrypted storage
- So the master portal could ‘just’ hold them

But

- Is that indeed better?
- The work flow may run for longer than 11 days
- master portal cannot renew the credential on behalf of the user (so needs to warn the user, and thus collect more PII than otherwise needed)
- master portal does not know the workflow, so cannot ‘renew’ credentials in the VO portal
- VO portal might renew, but then needs to be fully trusted to as to refresh based on existing (proxy) credential of the end-user at the master portal credential store

Thanks to all AARC folk whose slides and work I used in here –
esp. Mischa Sallé, Tamas Balogh, Jim Basney, Paul van Dijk

Thank you

Any Questions?

davidg@nikhef.nl

https://aarc-project.eu

The work leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 653965 (AARC).