RCauth Online CA service

Distributed operations and plans

rcauth.eu

@eoscfuture

Dissemination level: Public

EOSC-hub receives funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 777536.
and EOSC Future grant agreement id 101017536

Usual reminder - for n00bs and people with dodgy memories

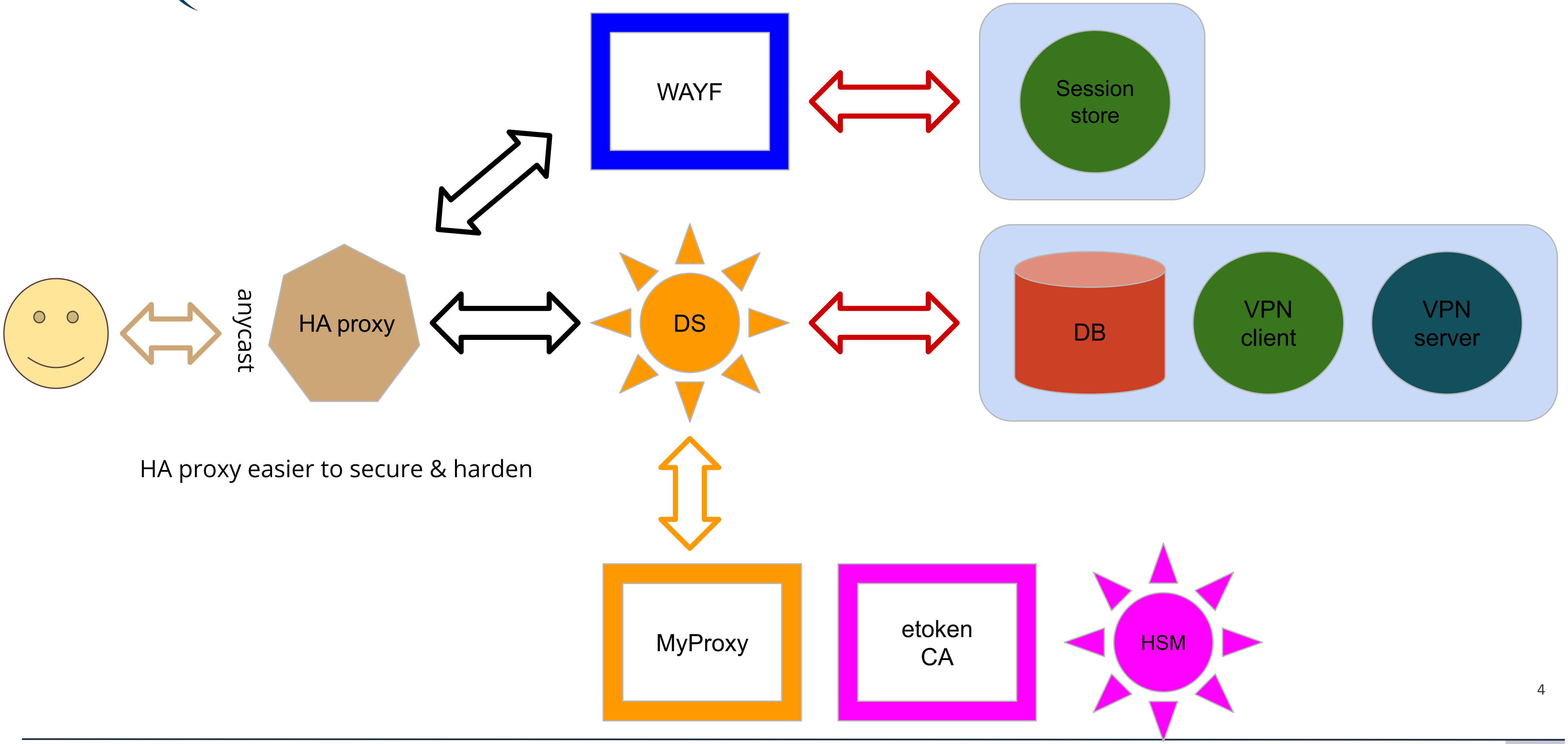
- RCauth is an IGTF accredited IOTA (DOGWOOD class) CA
  - Online credential conversion
  - Connected to eduGAIN (R&S+Sirtfi) plus direct, e.g. EGI Check-in and eduTEAMS
- EOSC Hub and EOSC Future implementing a **High Availability setup across 3 sites**

Outline (very, very approximate)

- Overview of architecture (reminder)
- Reusing RCauth
- HAHA proxy

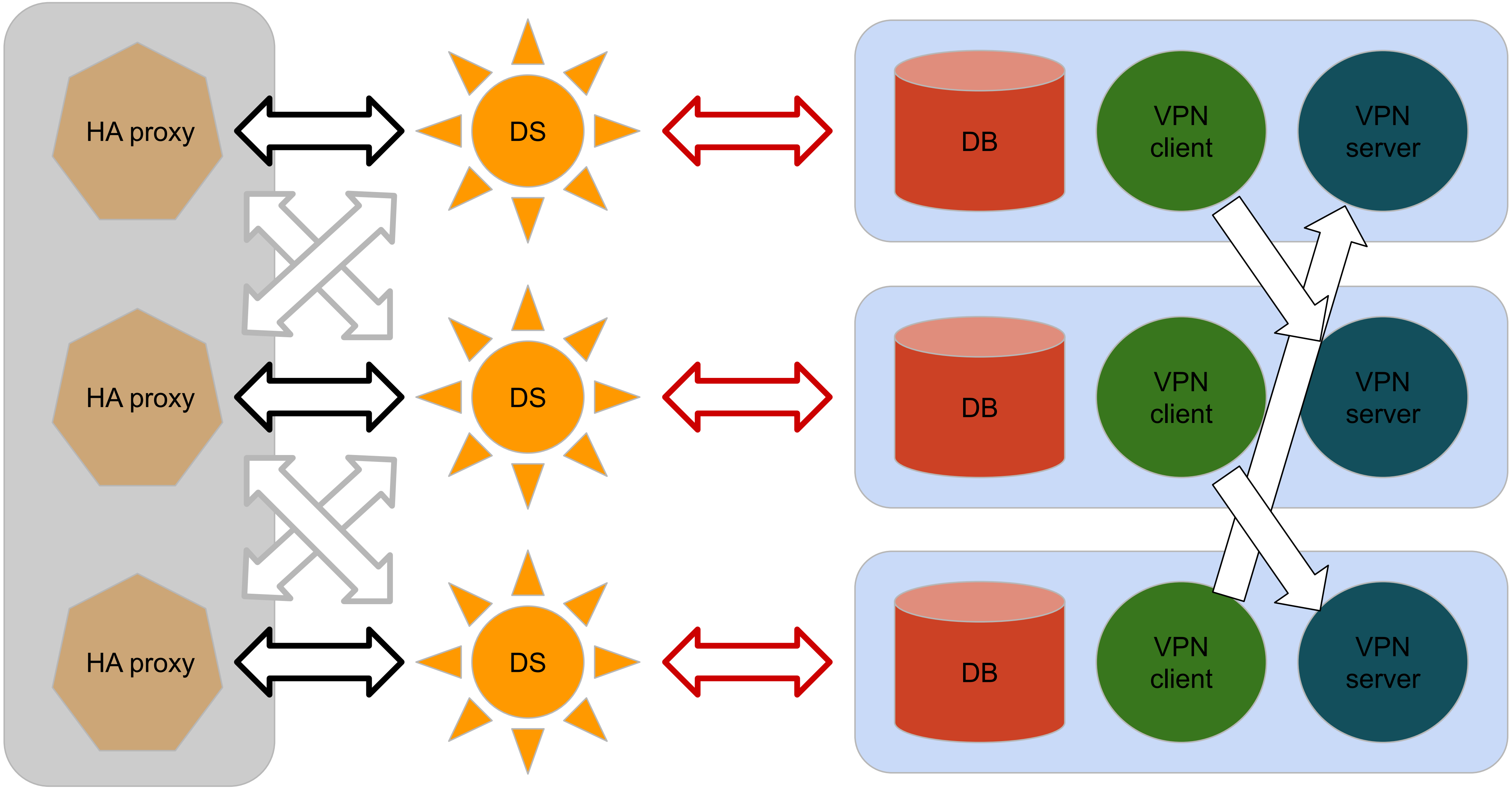
RCauth.eu

Final(ish) Architecture (1/3) - site view



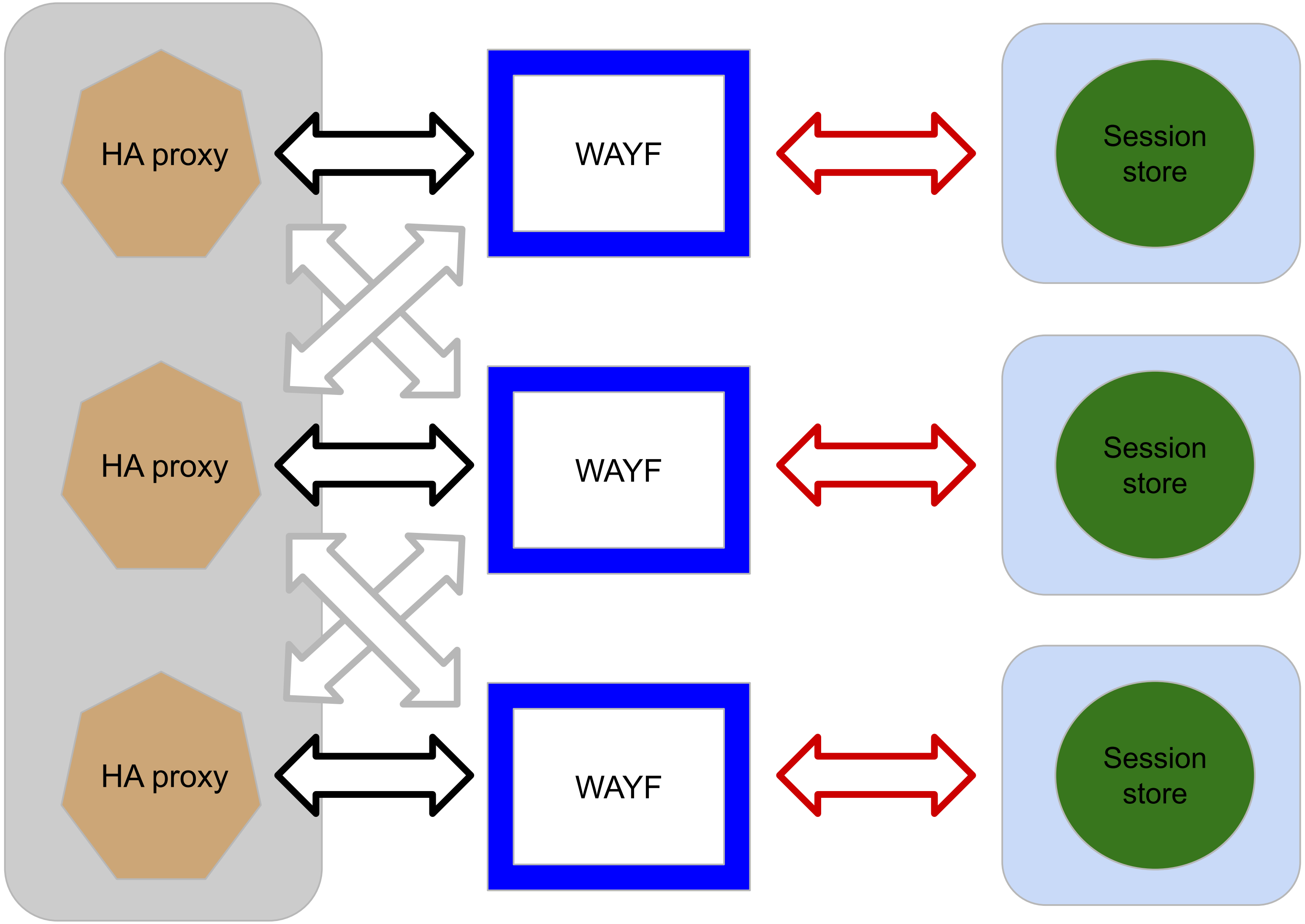
HA proxy easier to secure & harden

Final(ish) Architecture (2/3) - global DS view



RCauth.eu

Final(ish) Architecture (3/3) - global WAYF view



Reusing RCauth Researched Resources

- 3x node peer-peer redundant VPN
  - In principle extensible to >3 but what topology?
- Galera cluster is kind of old hat
  - Although using MySQL/MariaDB has certain (dis)advantages
- Splitting secrets - theory and practice
  - The difference between theory and practice is that, in theory, there is no difference
- High availability high availability proxy (HAHA proxy, more on this later)
- Distributed CRL updates
  - Lower latency-to-revoke => higher LoA (well, slightly)

Towards Tomorrow's Technology Trials?

Other technologies have been discussed and investigated - but not tested (by us (yet)):

- HA proxy: Leif, Niels, *et al*: the solution known as the "InAcademia solution"
  - On resolving the DNS name, a dynamic DNS returns a response pointing to the "closest" "alivest" node
  - relying on an *anycast*ed DNS
- Distributed services: memcached

Would R&S 2.0 be useful for RCauth?

Could we implement an ASPEN or BIRCH flavour/branch? (à la Pathfinder)

- And if we did, would we lose/gain users - and RPs?

R&S 2.0 based on profiles:

- anonymous coward profile
- pseudonymous auz profile
  - Approximately status quo
- personalized auz profile (in progress)
  - name/mail, org./aff., assurance

It needs to be implemented to be useful!

Failure modes of backend signing (high to low)

- Site X loses network access
  - HAP is not reachable, or DS not reachable
  - Site Y, Z’s HAPs run service, omitting routing to X’s DS
- Site X loses its DS
  - Site X’s HAP notices its preferred DS is unreachable and route to Y, Z
  - Site Y and Z’s HAPs notice X’s DS is unavailable and don’t route to it
- Site X loses backend MyProxy/signing
  - Its DS must notice and flag itself as down (firewall port 443)

Failure and Recovery

User view

- Individual transaction may fail ↯
- Try again and it should work...

Site view

- Monitoring should let site know a service is down
- RCauth should notice when site/service is back

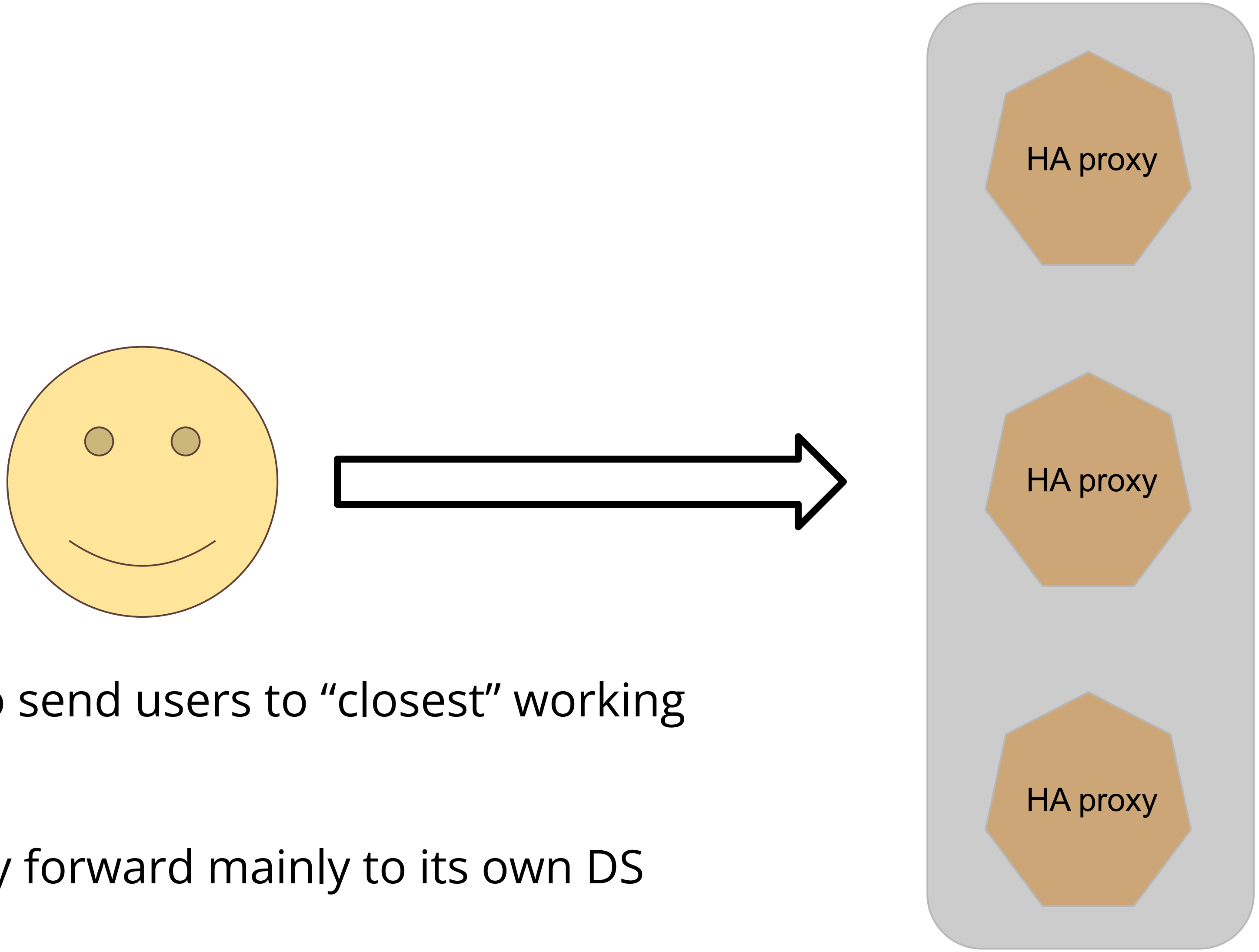
While we have done lots of testing, probably still some edge cases

RCauth operations

- Weekly meetings
  - notes and actions in wiki (prev. EOSC Hub now EOSC Future, in private area)
- Several repositories:
  - Public (software) https://github.com/rcauth-eu
  - Private keybase repo (medium-sensitive info)
  - Site-specific repos (for site-specific stuff)
- **What else do we need - to share knowledge with the AAAI community?**
  - Considering a paper describing the technologies

RCauth.eu

HAHA proxy



Need a way to send users to “closest” working service

Each HA proxy forward mainly to its own DS

If a HA loses its backend DS, it can still route to the other DSes

HA setup suggestions/discussion

- Christos: SUNET DNS magic used for InAcademia.org
- David: ANYCAST routing between the AS (Autonomous Systems) of Nikhef, JANET, GRNET using BGP (Border Gateway Protocol)

- What will work best for RCauth?
  - What will work best for others? (e.g. if inside an AS)
  - It would be good to collect experiences with both
- Try first with ***anycast***:
  simplest and most reliable when possible

RCauth.eu

ANYCAST / HAproxy 1/2

- e.g. used by Google and CloudFlare DNS 8.8.8.8 / 1.1.1.1
- single IP address for all three HAproxies:

  using BGP routing rules to get optimal route and to automatically failover
- GRNET and Nikhef have successfully deployed
- using bird and anycast-healthchecker
  - bird: user-space BGP client, announces anycast IP to router
  - anycast-healthchecker: monitors HAproxy and updates bird config
- easy to set up
- anycast does require collaboration with network engineers & NREN
- STFC investigating: meanwhile, can use Nikhef HAproxy instead

ANYCAST / HAproxy 2/2

- use same HAproxy for Delegation Servers and WAYFs
  - “vhost” config, straightforward in HAproxy
  - easier since we can reuse the anycast
  - remember: each HAproxy has its preferred backend, plus two failovers
- anycast-healthchecker checks that 3 conditions are satisfied:
  - HAproxy is running
  - at least 1 of the 3 DSes is up or starting
  - at least 1 of the 3 WAYFs is up or starting
- if check fails twice in a row -> update bird to remove site
- likewise for recovering from down state

Key Recovery Experiences

Spot the difference:

```
XOR=3Da3defc013510ff3a...
```

Key Recovery Experiences

Using FIFOs to guard secrets:

```
mkfifo sklyp
openssl rsa -check -noout -in sklyp &
./convert_revert.py secret1 0 secret2 0 >sklyp
```

This works, but the script reads each input twice, so won’t read input from a FIFO. So `secret1` or `secret2` cannot be FIFOs.

Thank you for your attention!

Questions?

Contact

RCauth Operations team
`ops-management(AT)rcauth.eu`

rcauth.eu

@eoscfuture