
CERN

Kerberos token renewal & HTCondor

Ben Jones

Why do we need token renewal?

- Broadly speaking, there are two submission methods for batch compute at CERN
  - Grid submissions (no need for kerberos)
  - Local submissions
- Local submissions have always relied on AFS for shared storage – AFS means AFS tokens (and more or less means kerberos too)
- Kerberos / AD ticket renewal policy doesn't match job queue + run length
  - 24h tokens renewable for 7 days

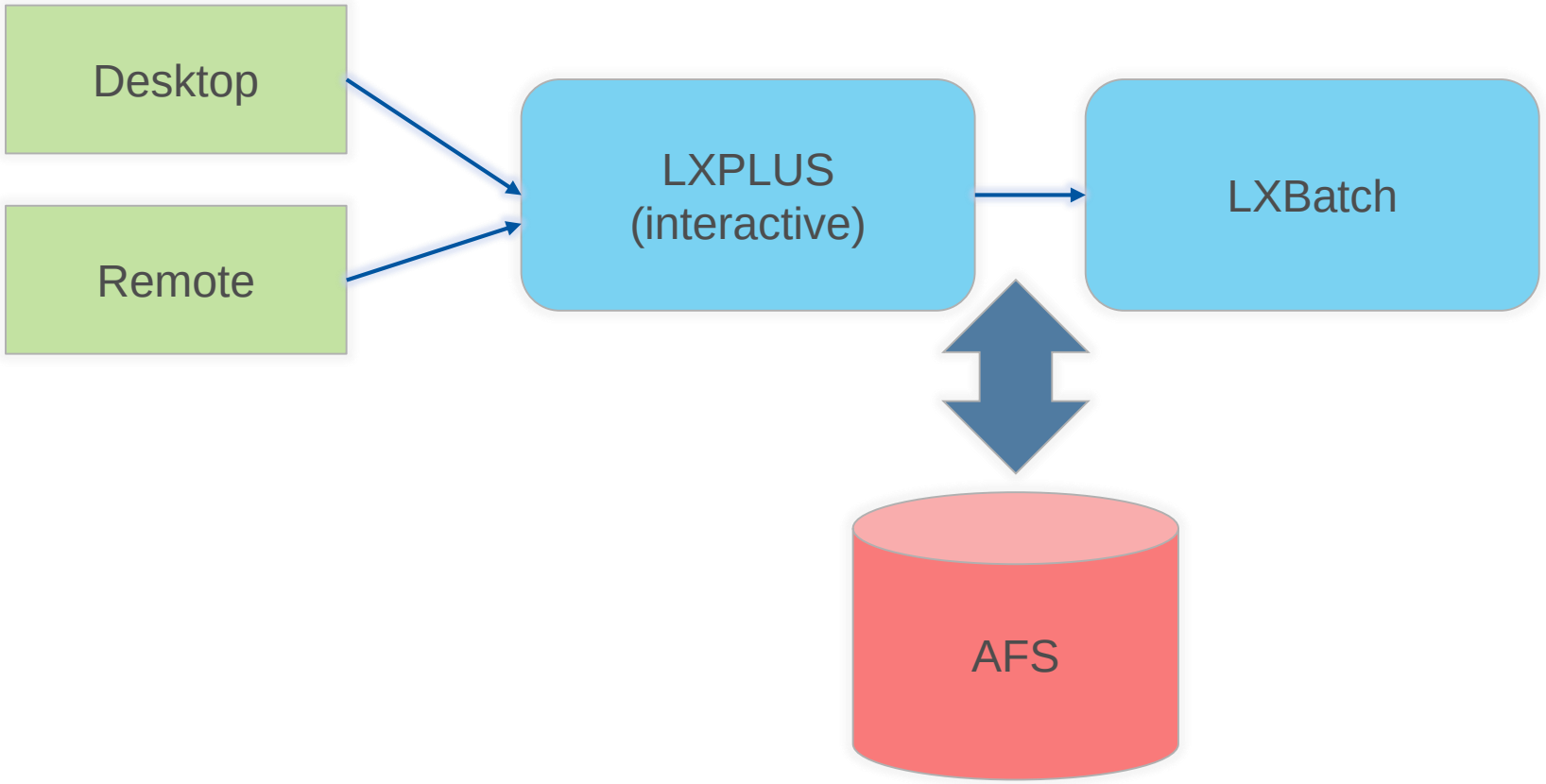
Wait… isn't CERN deco'ing AFS?!

- Migration away from AFS driven by concern at health of upstream project
  - releases, mail traffic, conferences, associated companies
  - no new features like IPv6, DES
  - ecosystem – little beyond two companies
- Slow migration – goal is no critical AFS deps by LHC Run 3 (2020+)
- No perfect drop-in replacement
- EOS for most data (via FUSE + CERNBox)
  - Perhaps for $HOME on LXPLUS

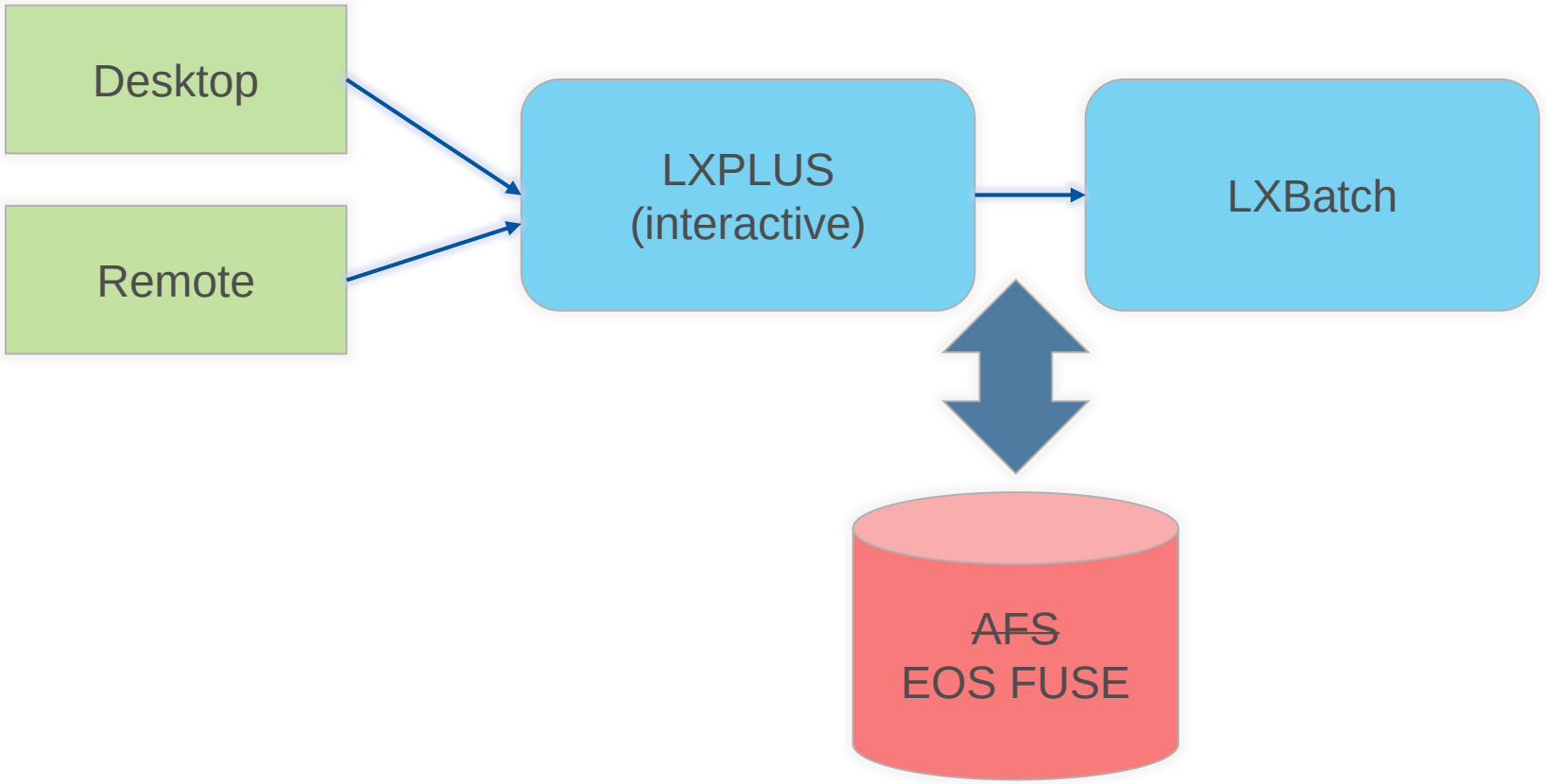
Other uses for tokens

- General purpose kerberos token & imaginative users == lots of kerberos dependencies
- One dependency we plan for: EOS FUSE uses kerberos tokens
- Others include other storage services, and any other service in CERN
- Even self contained user groups, such as ATLAS Tier-0 don’t understand all kerberos dependencies

Compute workflow



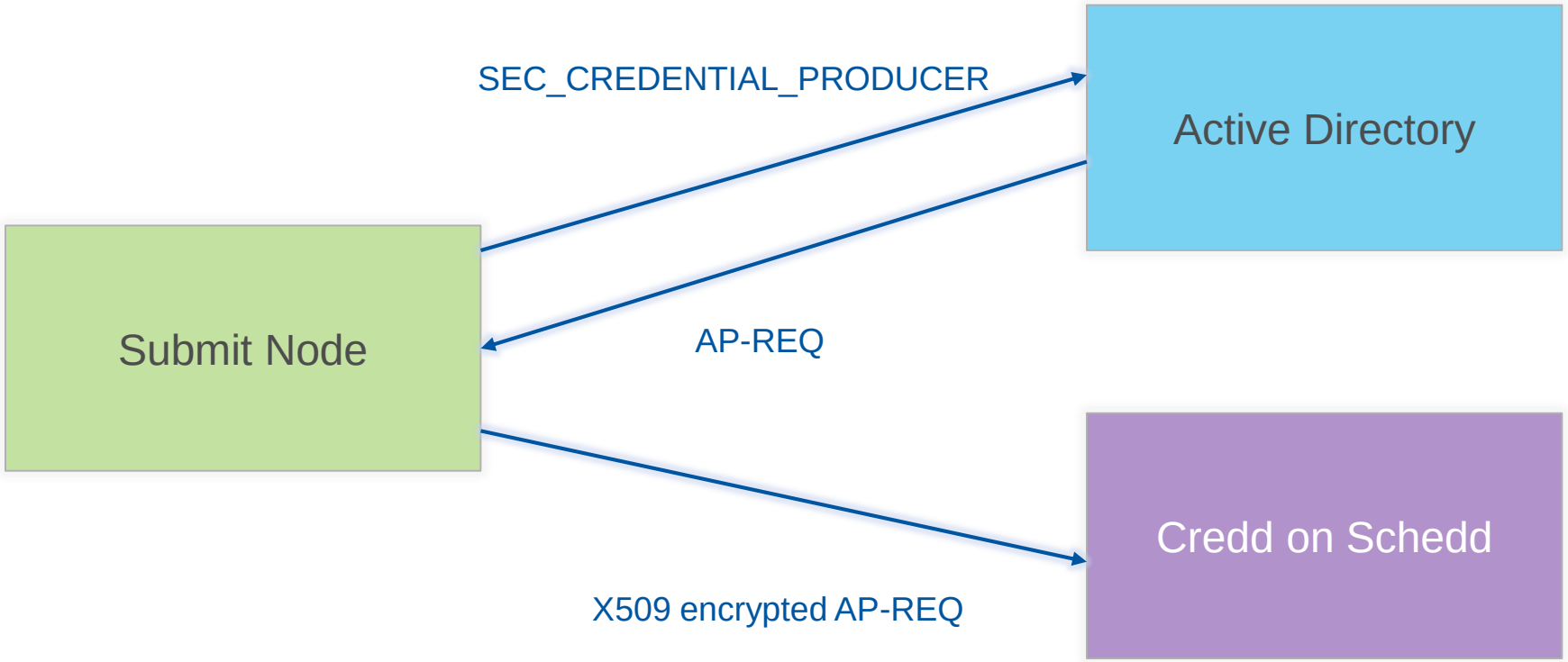
Compute workflow



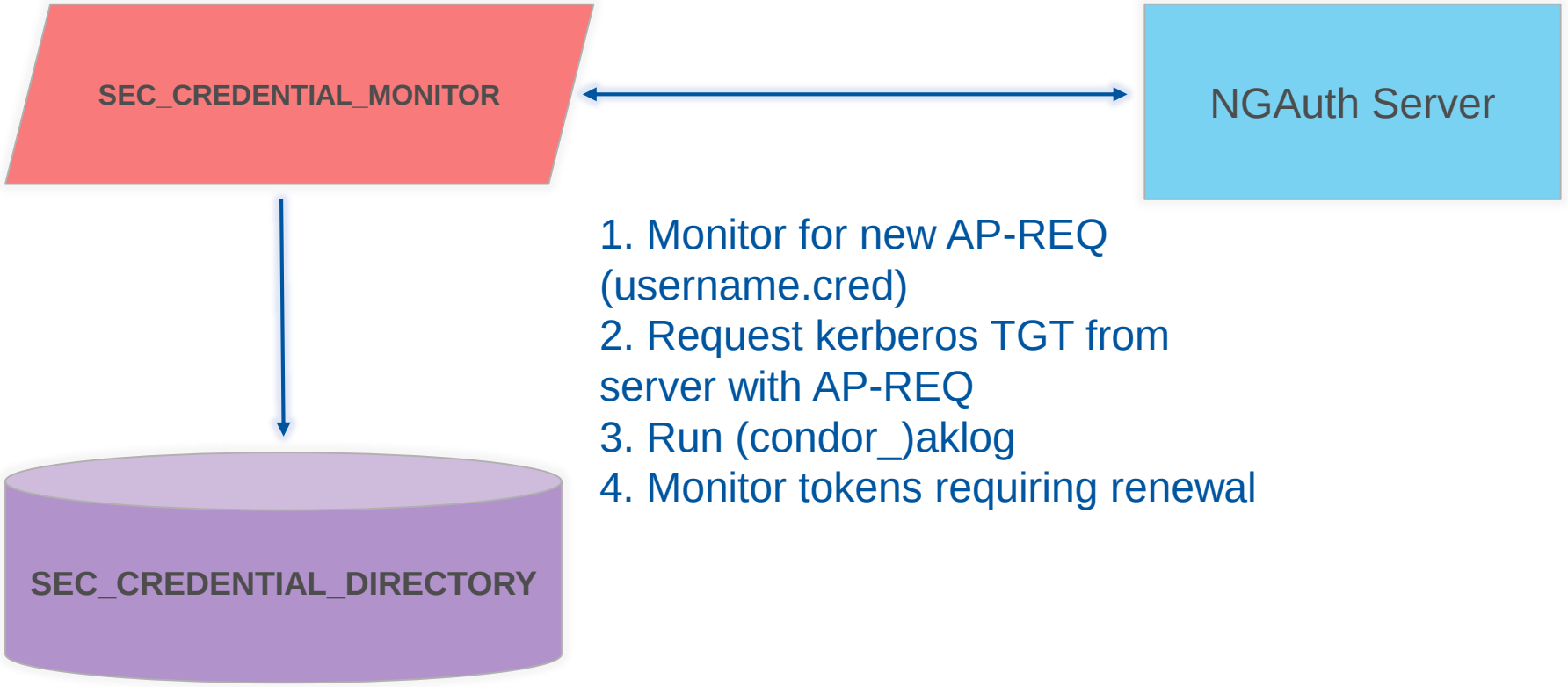
HTCondor integration

- The following touchpoints govern the integration with condor:
  - SEC_CREDENTIAL_PRODUCER
  - SEC_CREDENTIAL_MONITOR
  - SEC_CREDENTIAL_DIRECTORY
- Submit node is responsible for obtaining a token and sending to schedd
- Schedd turns submit token into kerberos TGT
- Execute node needs token, maintained as kerberos TGT

Submit node

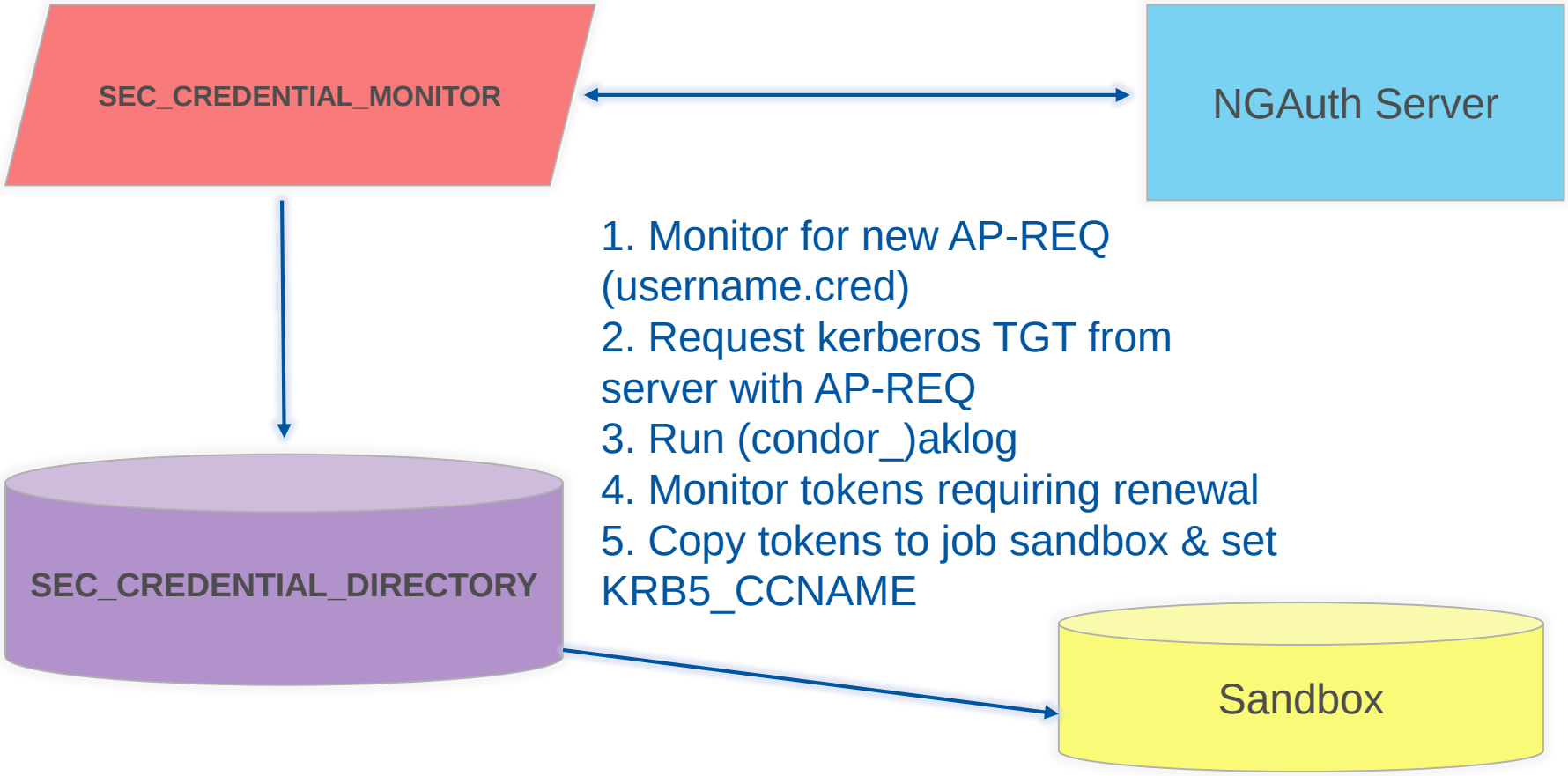

SEC_CREDENTIAL_PRODUCER
Active Directory
Submit Node
AP-REQ
Credd on Schedd
X509 encrypted AP-REQ

Schedd



1. Monitor for new AP-REQ (username.cred)
2. Request kerberos TGT from server with AP-REQ
3. Run (condor_)aklog
4. Monitor tokens requiring renewal

Execute node



NGAuth

- Service takes AP-REQ from credmon, extracts user info
- Uses privileged AD certificate to request kerberos TGT
- Encrypts TGT to x509 key deployed on schedds & worker nodes
- In principle multiple ngauth servers can be used
  - in practice some care is needed for DNS aliased names (rdns = false in krb5.conf)

Constrained Delegation (KCD)

- Rather than general purpose token, provides token that can be used with pre-defined subset
  - Less scary from security perspective
- Would require users to pre-define which services they want to use
  - This probably means we can’t do it
- Extension to KCD where services register
  - Not available

NGAuth issues

- Server with privilege to acquire tickets for any user
  - Could improve, but only to any user who opts in to batch service
- AP-REQ is effectively immortal
- No longer tied directly to AFS as per previous LSFAuth, but code & approach similar

Other options

- Certificates
  - Rather than AP-REQ, request certificate or proxy
  - Certificate could be used to then request the TGT
  - Easier to revoke certificates?
  - Time limited proxy?
- Longer expiry on kerberos tokens
  - Team responsible for AAA now responsible for NGAuth
    - Conway's Law might help improve the situation

Experience

- If ngauth server is overloaded, and no TGT is produced:
  - schedd – submission fails as log/out/err can't be initialised
  - execute – random job failures as paths can't be accessed
  - transient failures on exectute nodes can black hole jobs till detection
- Deploying new versions of credmon can be painful
- Can't set remote initialdir if it requires auth
- In general, mechanism works well

Links

- NoAFS:
  - CHEP-2016 - https://indico.cern.ch/event/505613/contributions/2230944/
  - HEPIX-2016-2 - https://indico.cern.ch/event/531810/contributions/2326350/

Questions?