HTCondor - KRB integration

First experiments and plans at DESY

Christoph Beyer & Thomas Finnern
With 1 slide from Thomas Hartmann

HTCondor Workshop DESY
Hamburg, 6-9 Jun 2017

Why KRB / AFS Support for local users ?

- AFS is still the primary $HOME for local users at DESY
- Heavily used in BIRD/NAF
- Users have legacy code, scripts, data,...
- Need for a stable shared filesystem
- Job should be independent from submit host
- Would at least be nice to keep it during migration to HTCondor
- KRB ticket handling can become handy for other services
- Future of (Open)AFS at DESY under discussion

KERBEROS – it’s easy in a kerberised ENV :)

- All you need is a ticket:

```
[chbeyer@pal44]~% echo $KRB5CCNAME
FILE:/tmp/krb5cc_4293_JNmh89
```

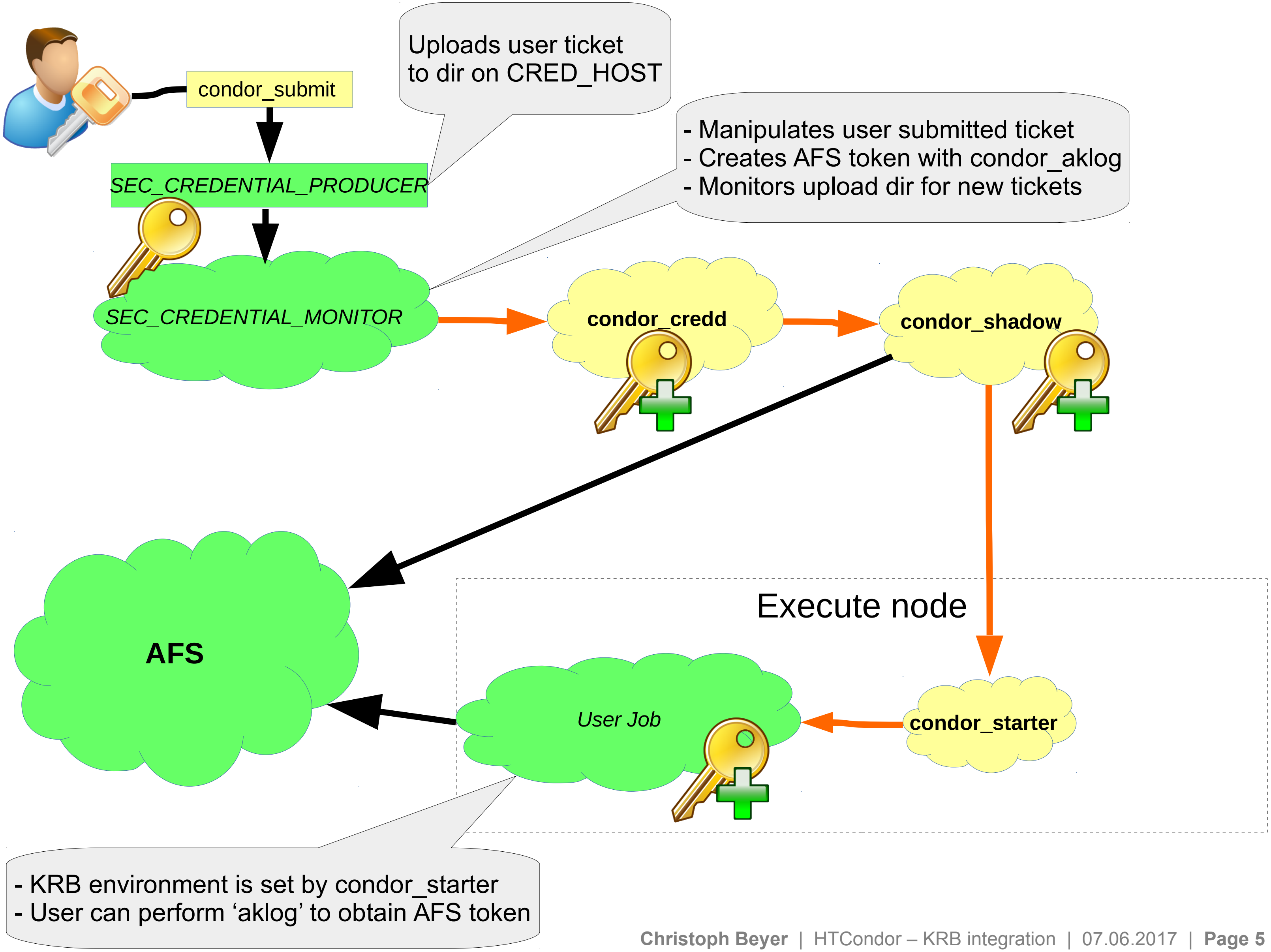
- Creating an AFS token is a piece of cake too:

```
[chbeyer@pal44]~% aklog
[chbeyer@pal44]~% klist
Ticket cache: FILE:/tmp/krb5cc_4293_JNmh89
Default principal: chbeyer@DESY.DE
Valid starting     Expires            Service principal
06/02/17 09:58:18  06/03/17 09:58:18  krbtgt/DESY.DE@DESY.DE
renew until 06/04/17 09:58:18
06/02/17 10:03:39  06/03/17 09:58:18  afs/desy.de@DESY.DE
renew until 06/04/17 09:58:18
```

So what’s all the fuzz about ?

- KRB ticket handling looks easy at first glance but gets more complicated when you want it to be secure, reliable and easy to use
- Creating an AFS token is easy inside your job when the KRB environment is set, hence writing to afs from inside the job is easy
- Having job output, log- and errorfiles in AFS means that the HTCondor daemons on the workernode and on the scheduler need AFS tokens too
- The token should be transferred with the job for not having all available tokens on all workernodes
- You don’t want to keep credentials of users forever unless they do have running or hold jobs and will need a credential later on
- The lifetime of a standard KRB/AFS ticket is usually 24 hours and some condor jobs may run longer than that, currently guaranteed 1 week ticket lifetime

KRB handling in HTCondor



Uploads user ticket to dir on CRED_HOST

- Manipulates user submitted ticket
- Creates AFS token with condor_aklog
- Monitors upload dir for new tickets

- KRB environment is set by condor_starter
- User can perform ‘aklog’ to obtain AFS token

SEC_CREDENTIAL_PRODUCER & SEC_CEREDENTIAL_MONITOR

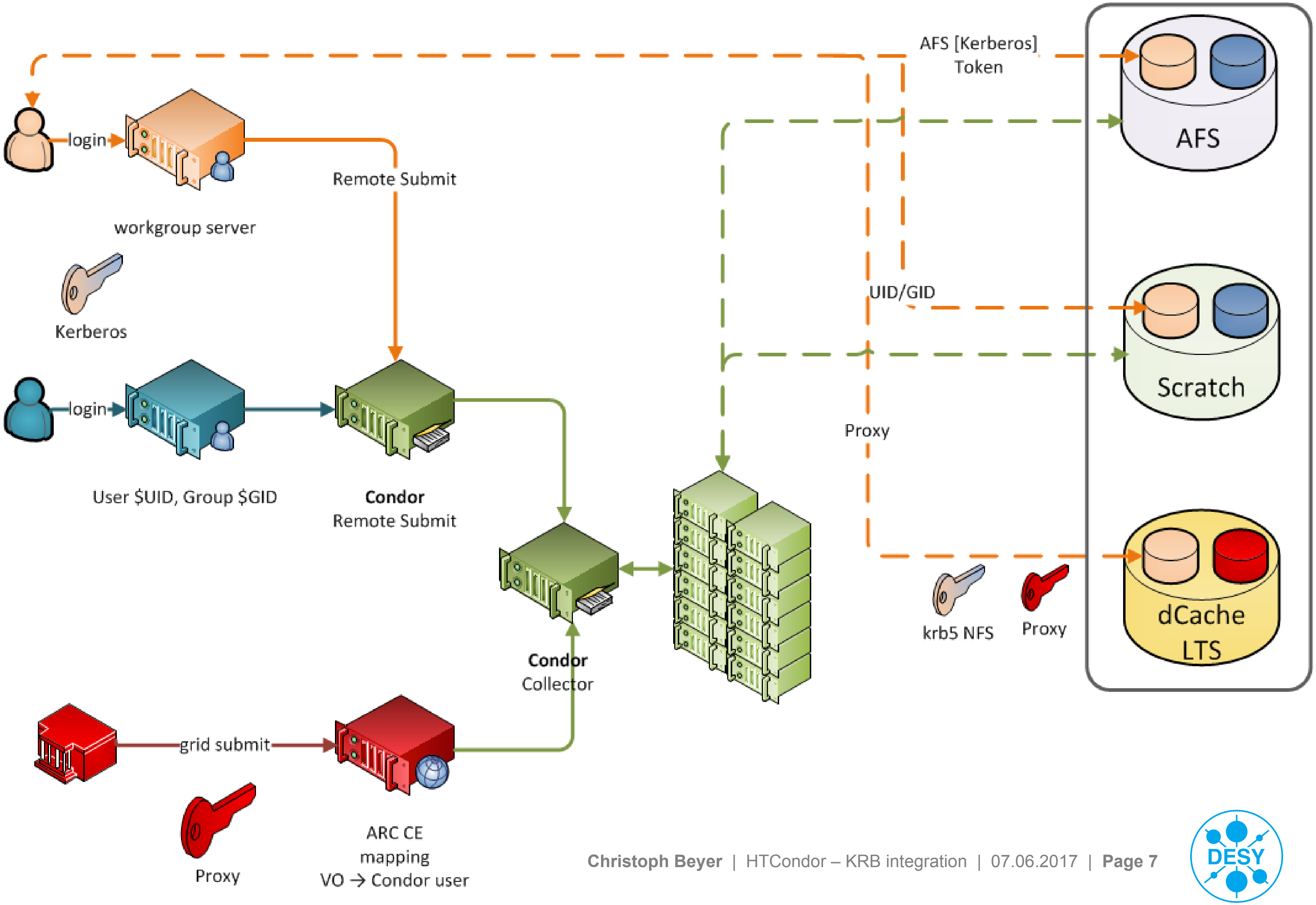
SEC_CREDENTIAL_PRODUCER

- Shellscript
- Performs some sanity checks on the users KRB ticket
- Runs ‘condor_aklog’
- Writes KRB ticket to <STDOUT>
- Writes job log (?)

SEC_CREDENTIAL_MONITOR

- Shellscript
- Checks for KRB tickets uploaded by SEC_CREDENTIAL_PRODUCER
- On the master: Replaces uploaded tickets by identical usertickets with a longer time to live (2 weeks renewable)
- Renews ticket via ARC (authenticated remote command) to ensure at least 1 week ticket lifetime
- On the workernode Prolongs tickets (no special authorization needed)
- Runs ‘condor_aklog’

Layout for a combined batch facility at DESY



The End

- Questions ?
- SendCredential = True/False ?
- CREDD enhanced token management possible ?