ENDIT Plugins

NeIC NT1 Manager
Mattias Wadenstein
<maswan@ndgf.org>

2020-10-21
NDGF AHM, Zoom

NORDIC E-INFRASTRUCTURE COLLABORATION

Overview

- Design
- NDGF deployment
- Main challenges
- dCache endit plugin

SPEAKER | Mattias Wadenstein <maswan@ndg

What is ENDIT

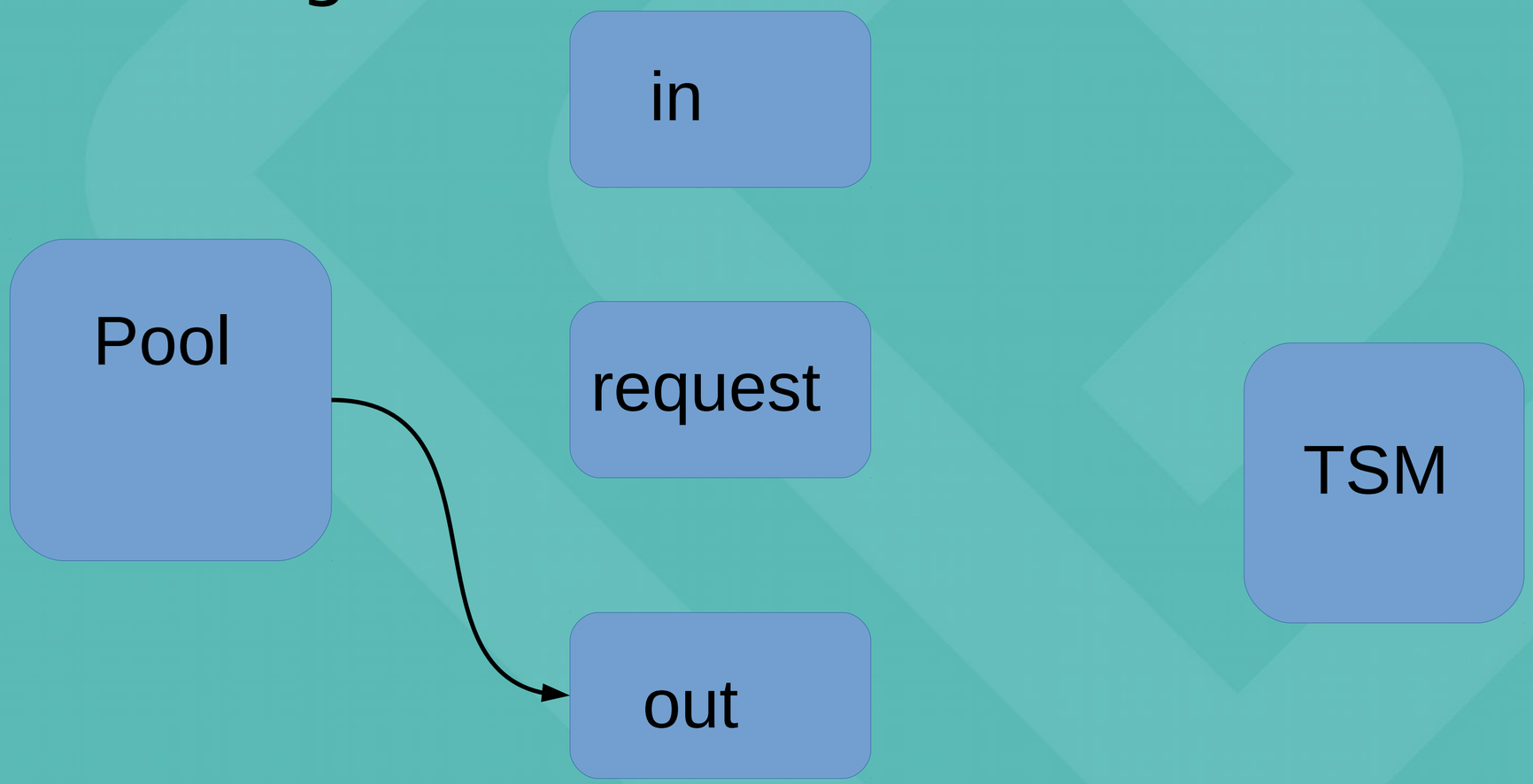
- Efficient Nordic Dcache Interface to TSM
  - Or, well, IBM Spectrum Protect as it is called these days
- A package to use a TSM controlled tape library as an HSM backend for dCache
  - Most of our sites run TSM for backups, so using existing infra
- Designed for efficiency
  - Now also for scalability
- In production use by NDGF-T1 for a decade

ENDIT design

- Using the dsmc command line client to get/put/rm
  - Assumption: Unlikely to lose data due to weird corner cases
  - Using intermediate directories to create batching for efficiency
- Thresholds for when to stage in size, time, etc
- Use of dedicated tape read and write nodes
  - Mostly a consideration for performance
  - At NDGF we do a pool2pool copy for reads, so the clients hit the same disk pools as disk data

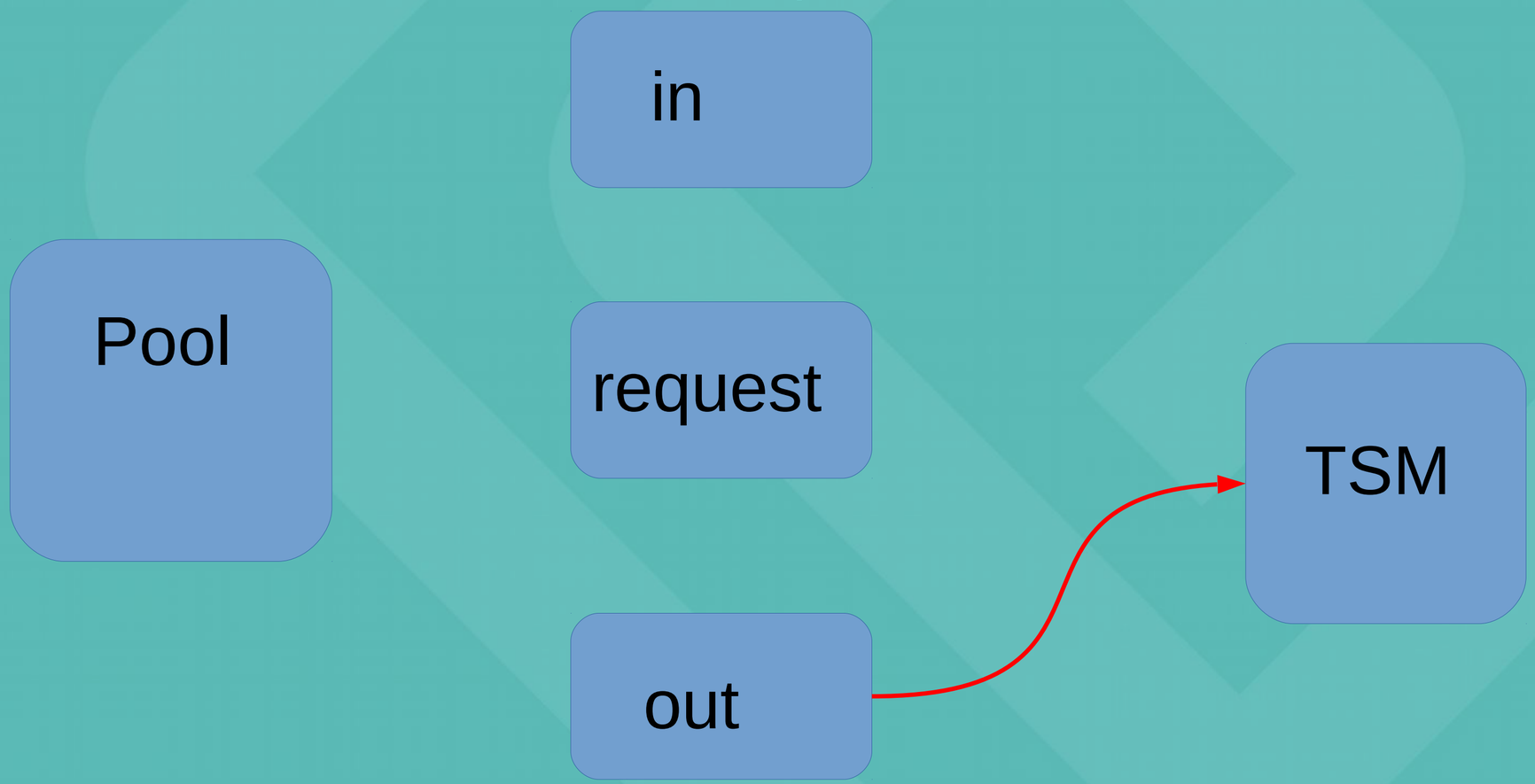
ENDIT design

- Put, step 1: A hardlink is created in “out” for the file staged when dCache flushes it



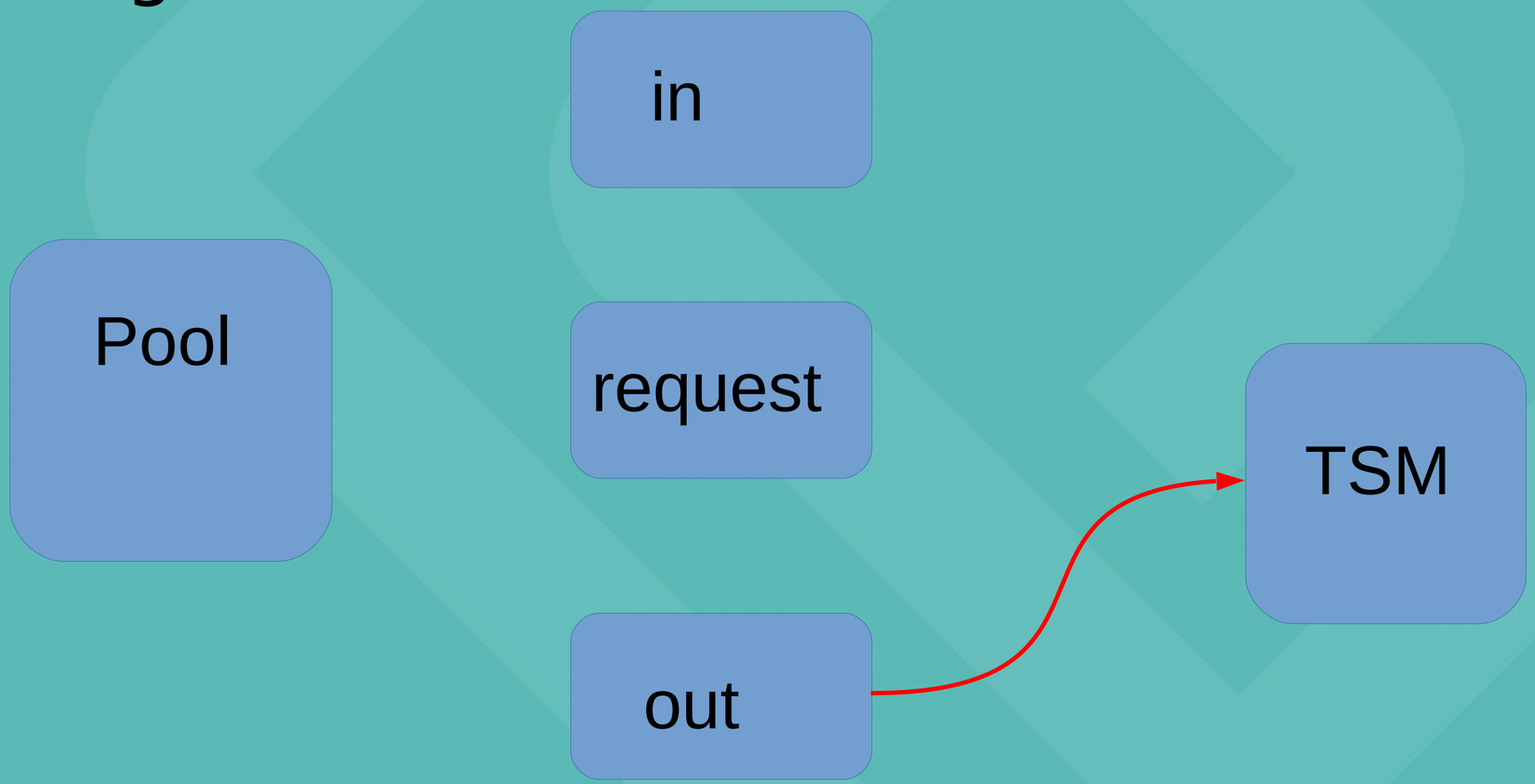
ENDIT design

- Put, step 2: Time passes. When there is more than X GB files or Y time, dsmc archive -delete out/*



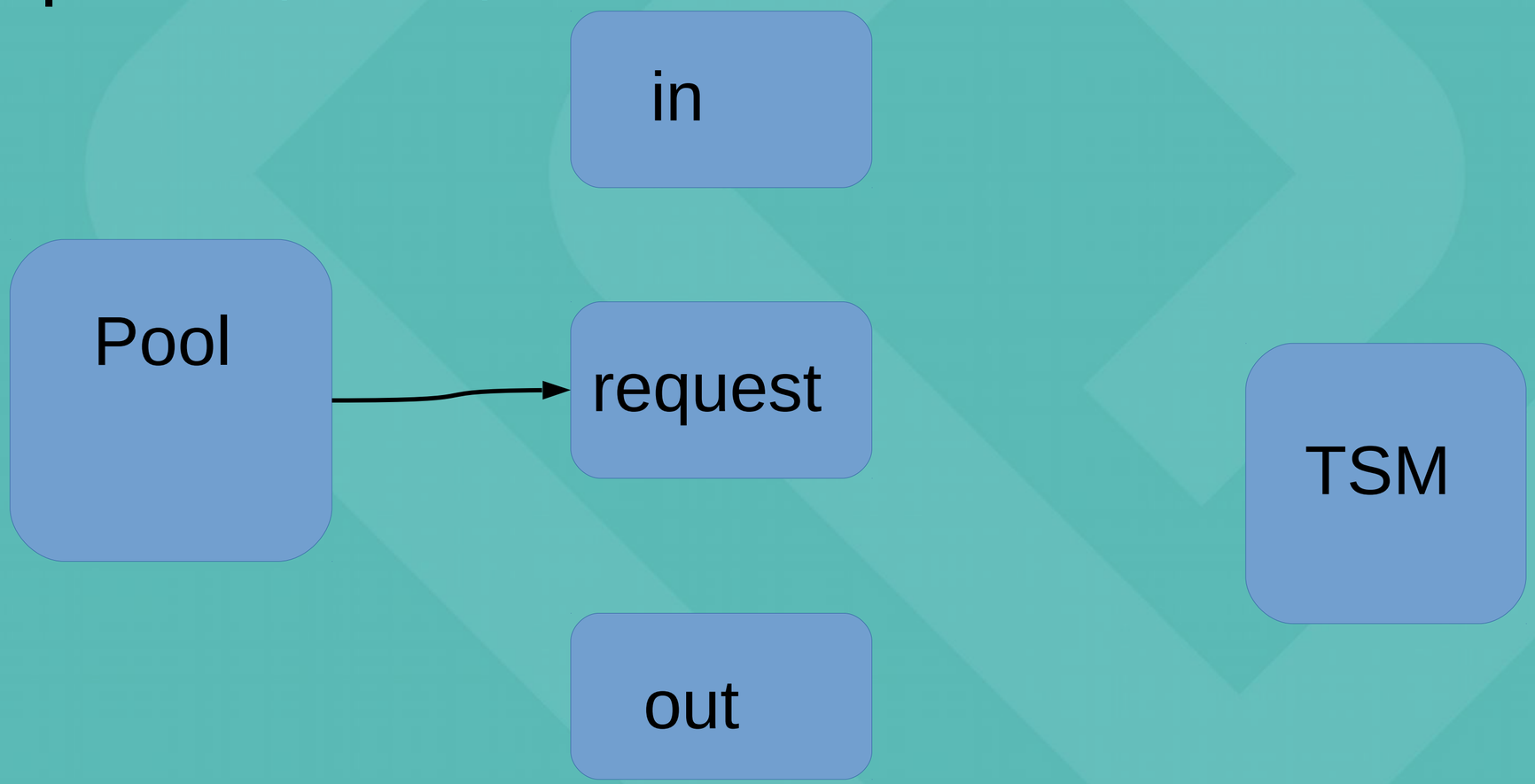
ENDIT design

- Put, step 3: the ENDIT plugin discovers that the file is gone from out and considers it successfully put



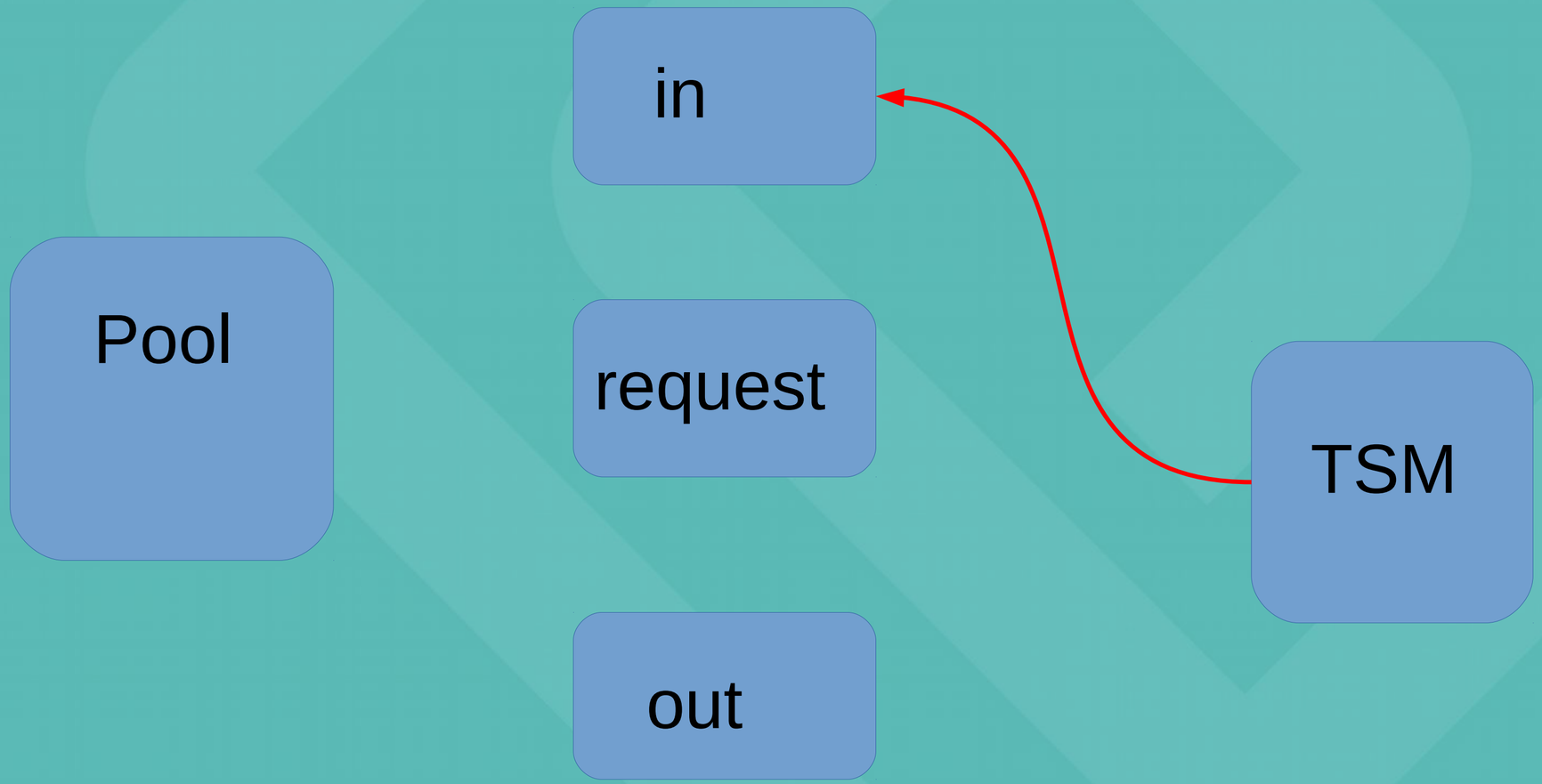
ENDIT design

- Get, step 1: The plugin creates a request file with pnfsid, size, etc



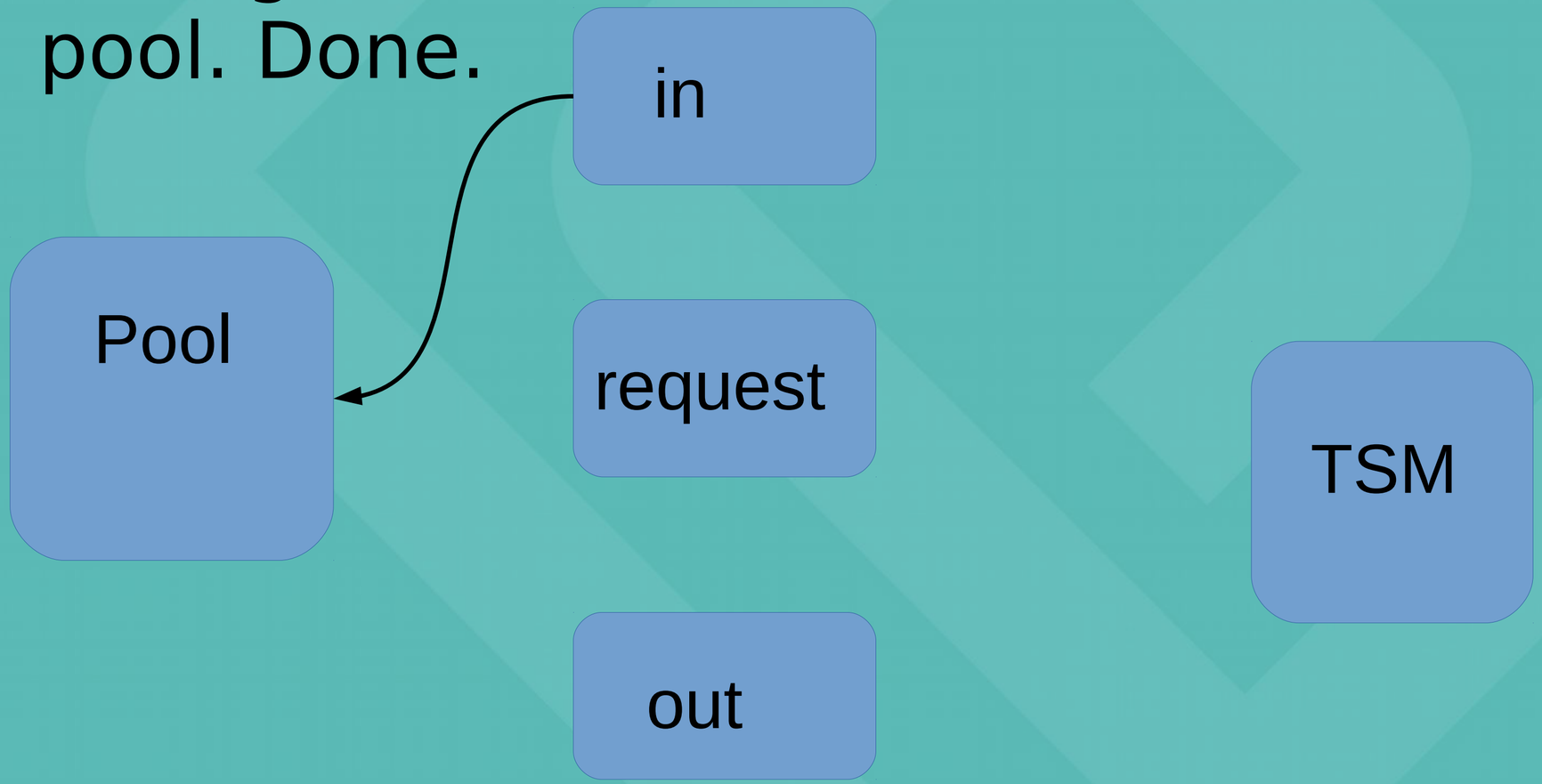
ENDIT design

- Get, step 2: Time passes, X or Y then the endit daemon retrieves the files from TSM to in/



ENDIT design

- Get, step 3: When the plugin discovers a file with the right name and size in “in/”, rename it into the pool. Done.



NDGF deployment

- We have multiple tape libraries
  - Each library and VO is their own hsminstance
    - alice.hpc2n.umu.se, atlas.hpc.ku.dk, etc
- Each has their own read and write pool
  - Fairly small buffer spaces
  - After restore, a read will trigger an automatic pool2pool transfer into the main disk poolgroups for reading
    - This way multiple reads of the same file won’t generate lots of extra traffic on the buffer space
    - If the only read is an FTS copy to disk, this is an extra copy, but we have no way of enforcing this on our side

dCache plugin

- Instead of a HSM script like most dCache installations, we use a dCache plugin
  - https://github.com/neicnordic/dcache-endit-provider/
  - AGPL just like dCache
  - Just unpack the plugin in the plugin directory
  - Then configure through the dCache admin interface
- Much better scalability than the script
  - Tested to 100k outstanding read requests per pool
  - Can do restores as fast as the rest of dCache can handle it (probably latency bound to namespace from a single pool)

Main challenges

- dsmc only does reordering within a session
  - i.e. one invocation of dsmc retrieve -filelist=f.txt
  - Makes it tricky to build retrieve parallelism – we solved this with generating a mapping of filename → tape and run one session per tape
- Some quirks about dsmc and TSM
  - Like having to sleep for a second before renaming the file after it gets the correct size to avoid a race condition
- Pure TSM issues
  - No support for RAO yet
- dCache insists on reserving space for all reads
  - Limits the reordering to a small subset of total number of requests

Future work

- Create an option to not reserve space for files in the dCache pool until just before the rename() from the in/ directory
  - Necessary to be able to push sufficient number of requests to ENDIT daemons to get good throughput
- Other dCache features we’re thinking of:
  - Fallback tape write pools
  - Write overflow to _disk pool groups
  - The automatic pool to pool transfer to prefer within the same site
    - Shouldn’t be too hard to add to the zone concept

Questions?

NORDIC E-INFRASTRUCTURE COLLABORATION