ENDIT 1.0

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

2018-05-29
dCache workshop
Hamburg, Germany

NORDIC E-INFRASTRUCTURE COLLABORATION

Overview

- What is ENDIT?
- Main challenges
- New dCache endit plugin
- Updates in the daemons

What is ENDIT

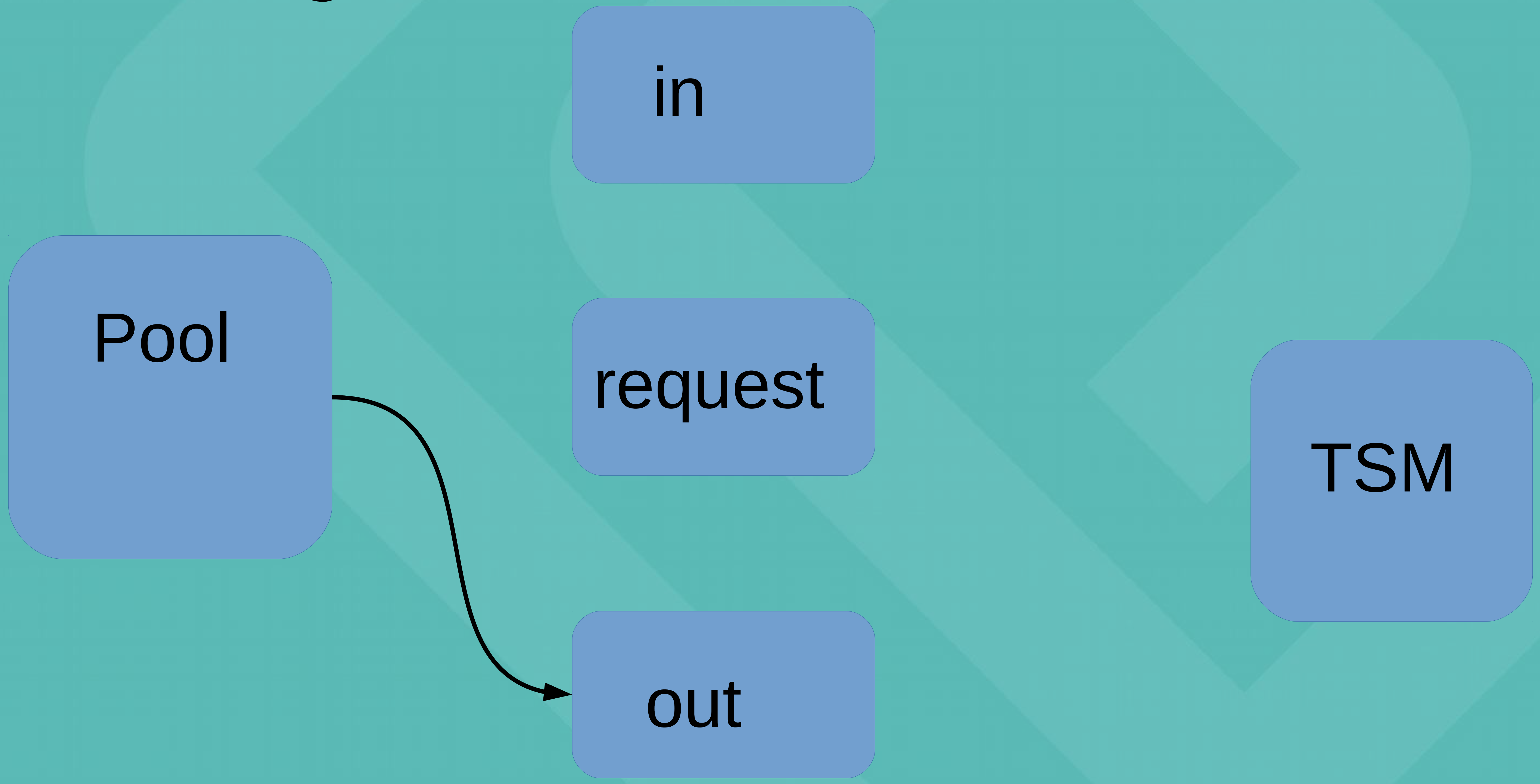
- Efficient Nordic Dcache Interface to TSM
  - Or, well, IBM Spectrum Protect as it is called these days
- A package to use a TSM controlled tape library as an HSM backend for dCache
- Designed for efficiency
  - Now also for scalability
- In production use by NDGF-T1 for a decade
  - Latest version for a month or three

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

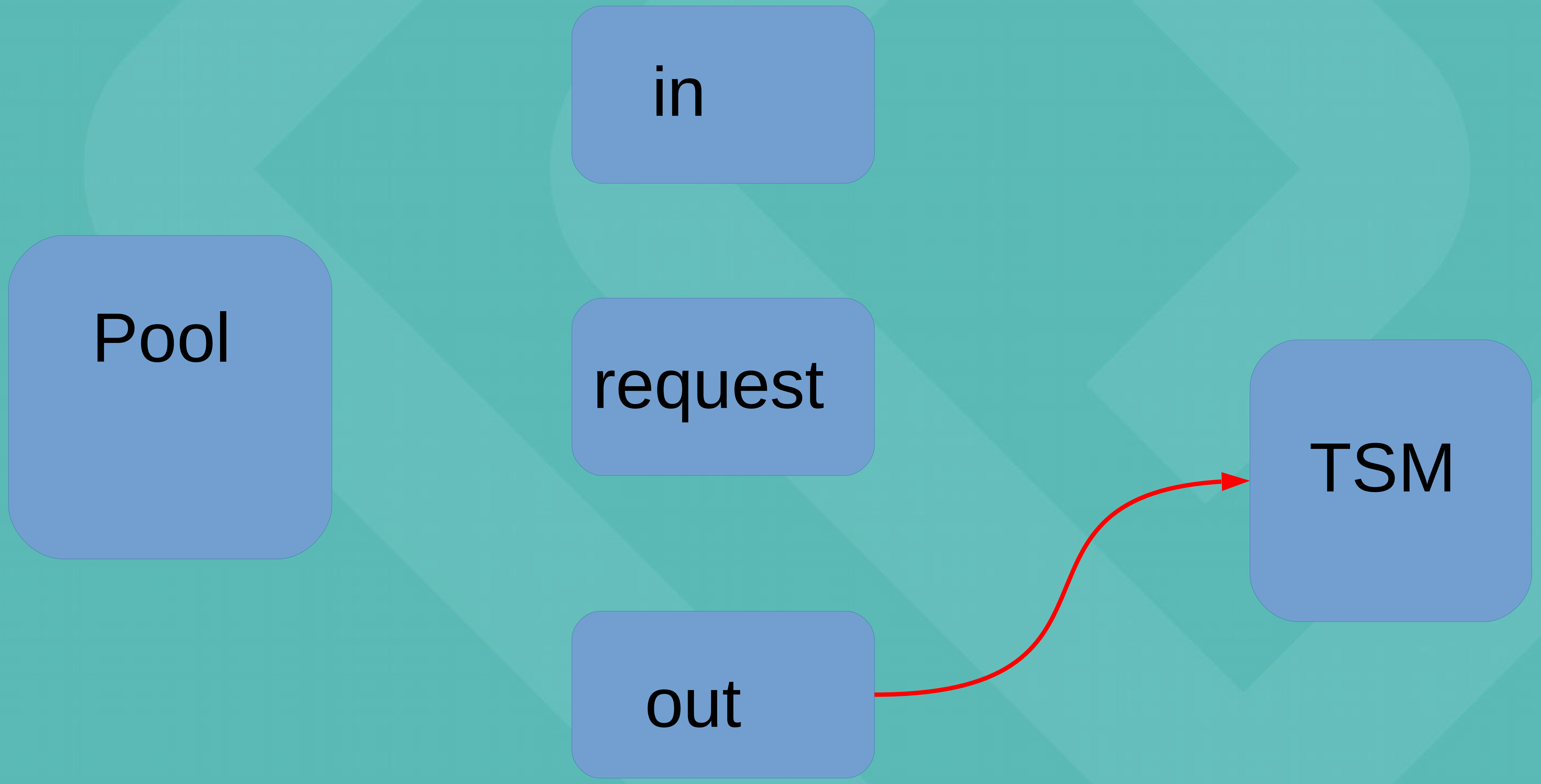
ENDIT design

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



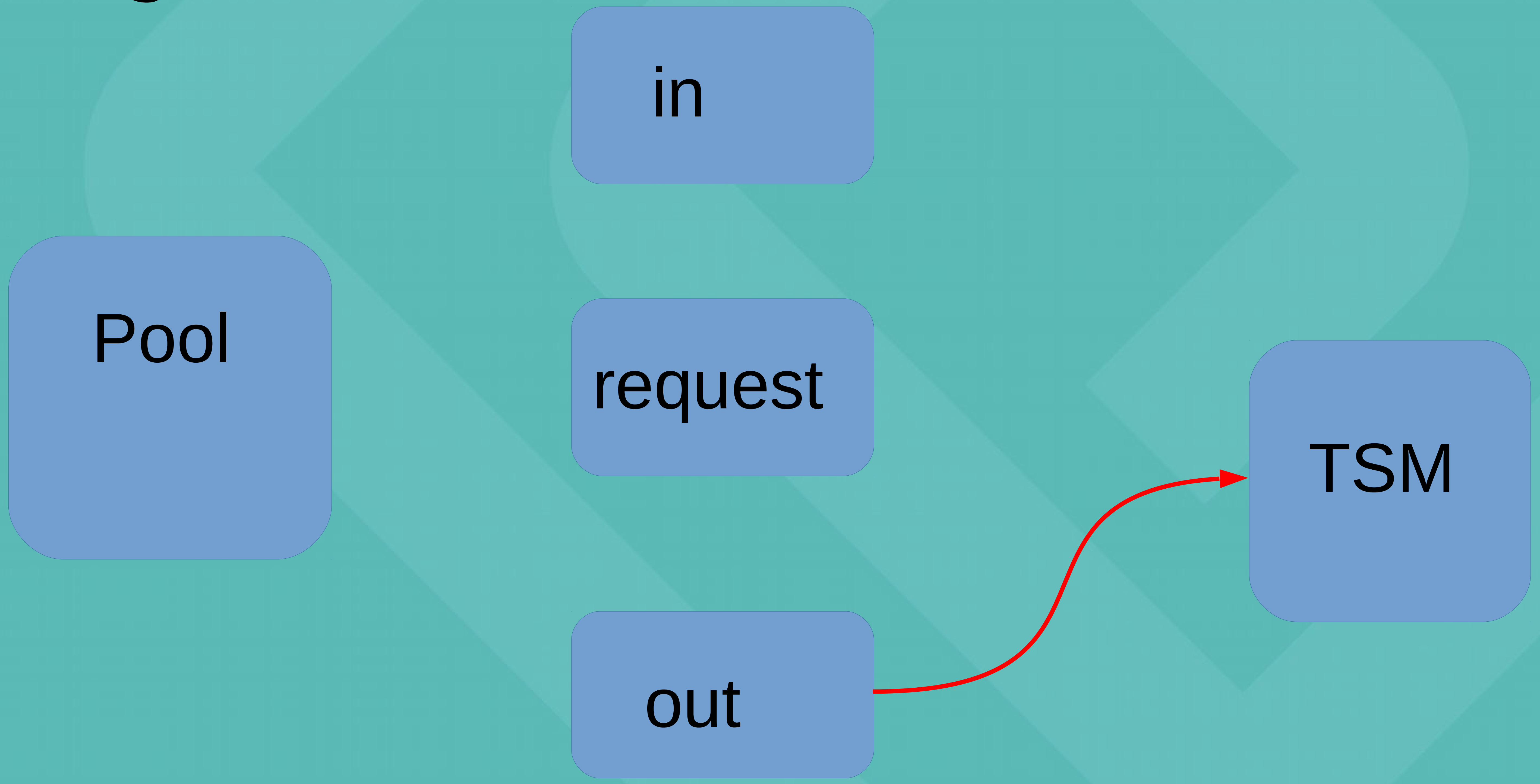
ENDIT design

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



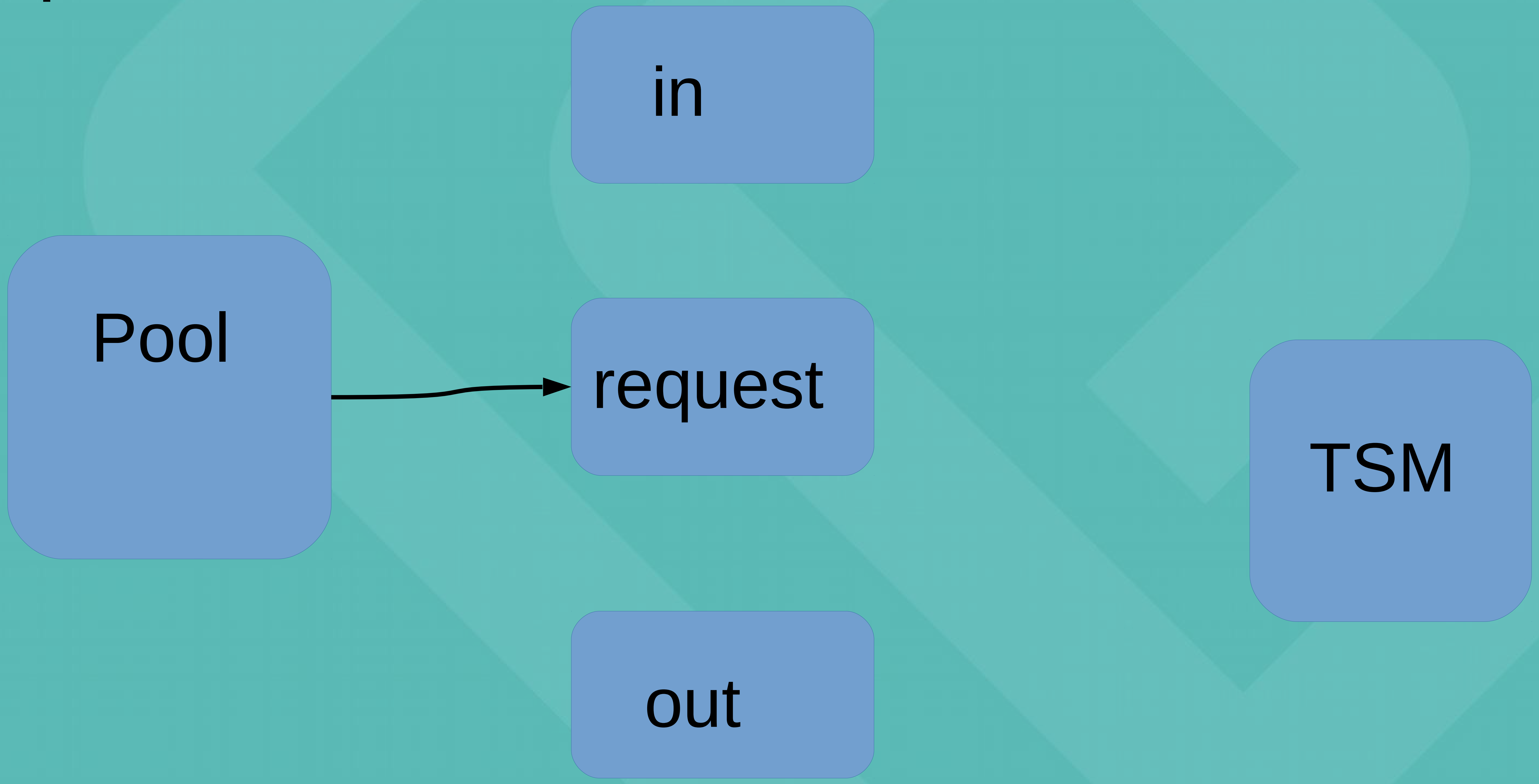
ENDIT design

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



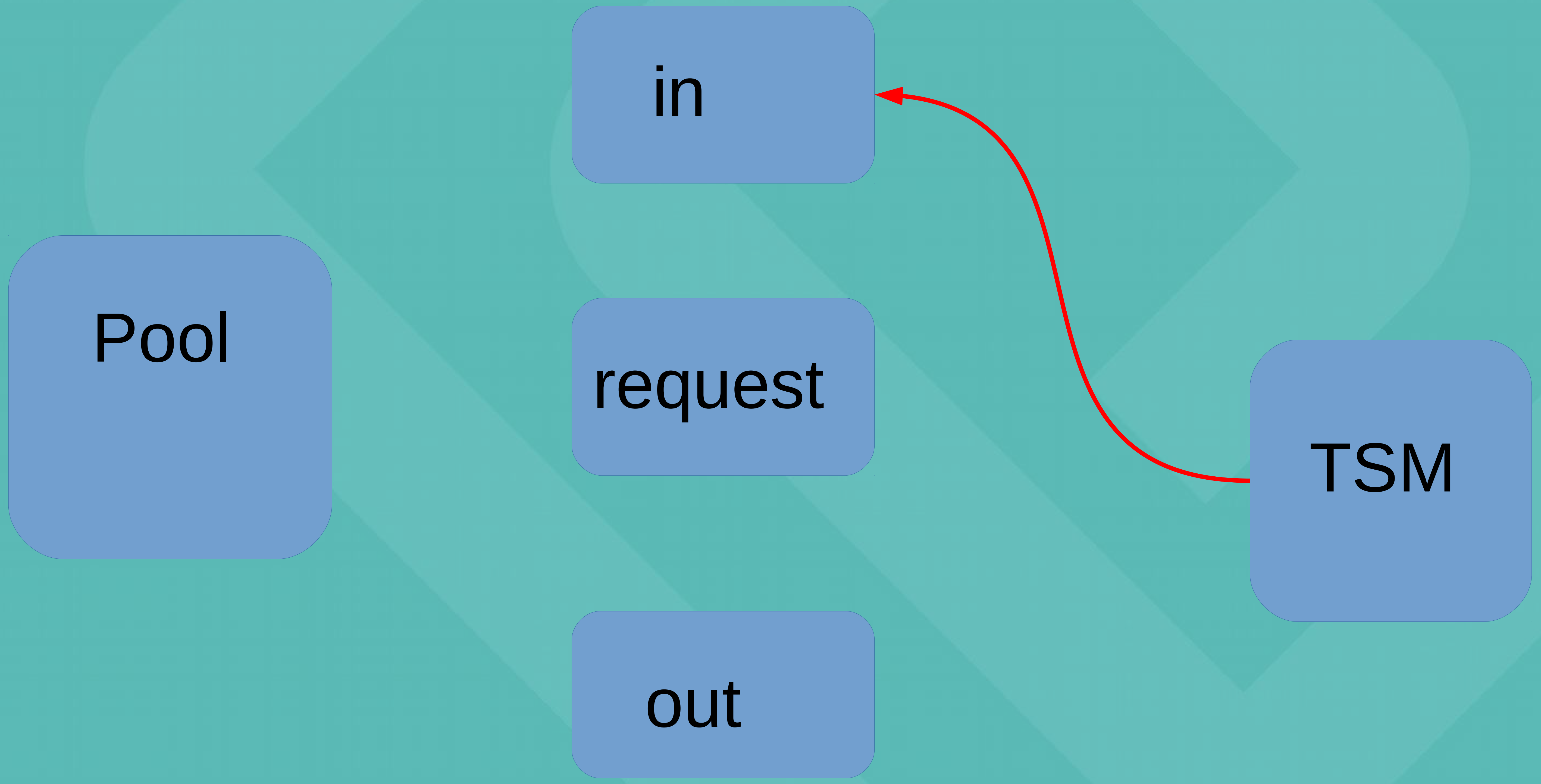
ENDIT design

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



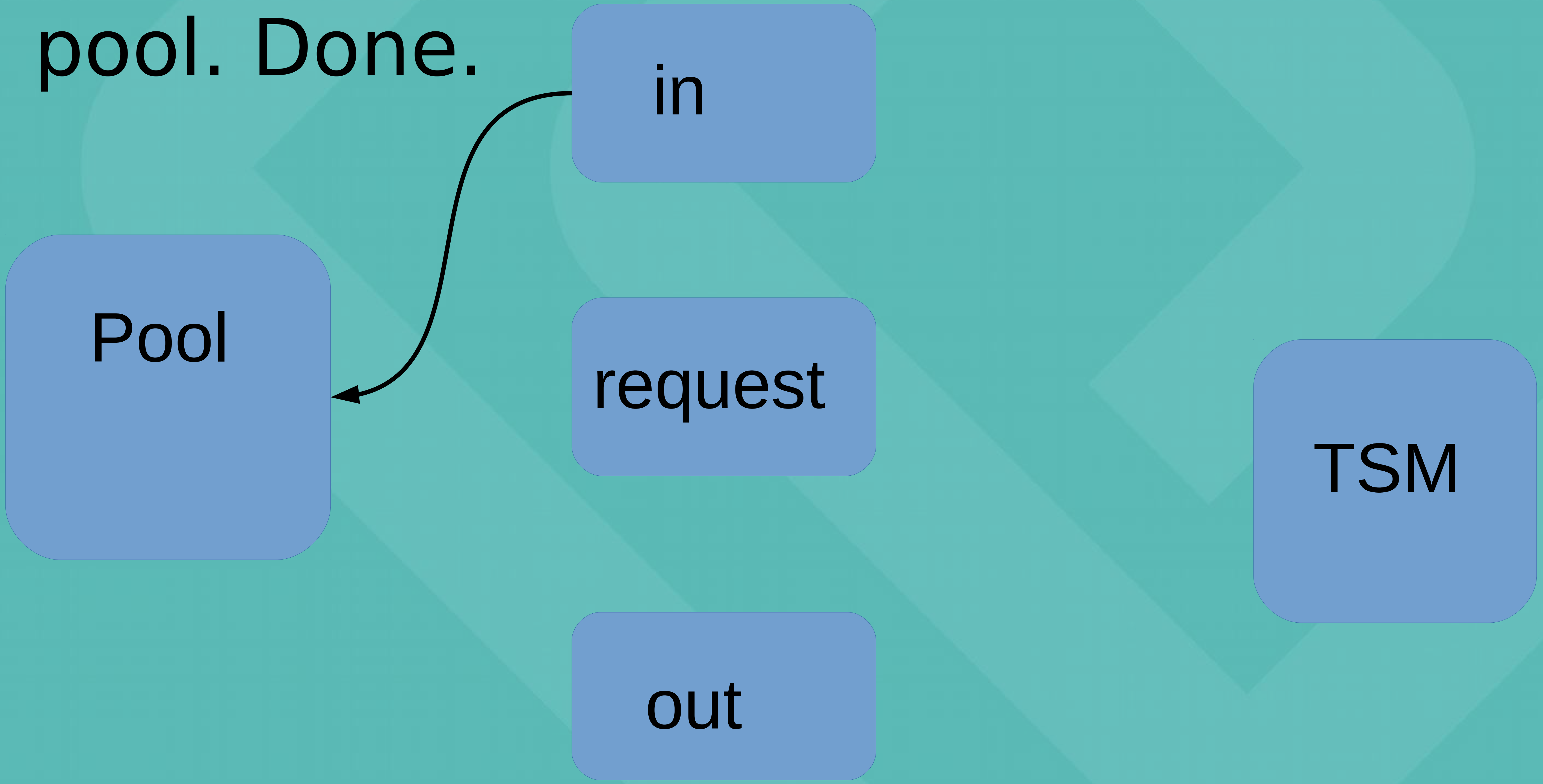
ENDIT design

- 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.



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

dCache plugin

- Instead of a HSM script, we now 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
  - Can do restores as fast as the rest of dCache can handle it (probably latency bound to namespace from a single pool)

ENDIT daemons

- A set of daemons that does the TSM interaction
  - https://github.com/neicnordic/endit/
  - GPLv3
  - Perl, using IPC::Run3 to run external commands
  - One each for archive, retrieve, remove
- Configuration in a common config file
- Needs to run as the same user as the pool

Daemon news

- Template based configuration
  - Most config renamed, so please generate a new one and re-fill relevant bits when upgrading from old endit
- tsmretriever.pl cleans up old files in “in/” during startup
- Information logged for statistics use
  - Consuming them is future work
- Parallelism in writes
  - <x GB: n=1; >x <y: n=2, >y: n=3, etc

Daemon news

- Parallelism in reading based on a tapes.hint file
  - File needs to be produced in close cooperation with the TSM server admins and regularly updated
  - One dsmc retrieve session for each tape up to config limit
  - Requests for files not in the hints file is handled by fallback of all such files being considered to be on the “default” tape
- Throttling for trickle reads
- Lots of bugfixes
- https://wiki.neic.no/wiki/DCache_TSM_interface

Questions?

NORDIC E-INFRASTRUCTURE COLLABORATION