

Queueing in dCache

Paul Millar

Berlin, 2013.05.28



HELMHOLTZ
ASSOCIATION



Mythical self-organising users?



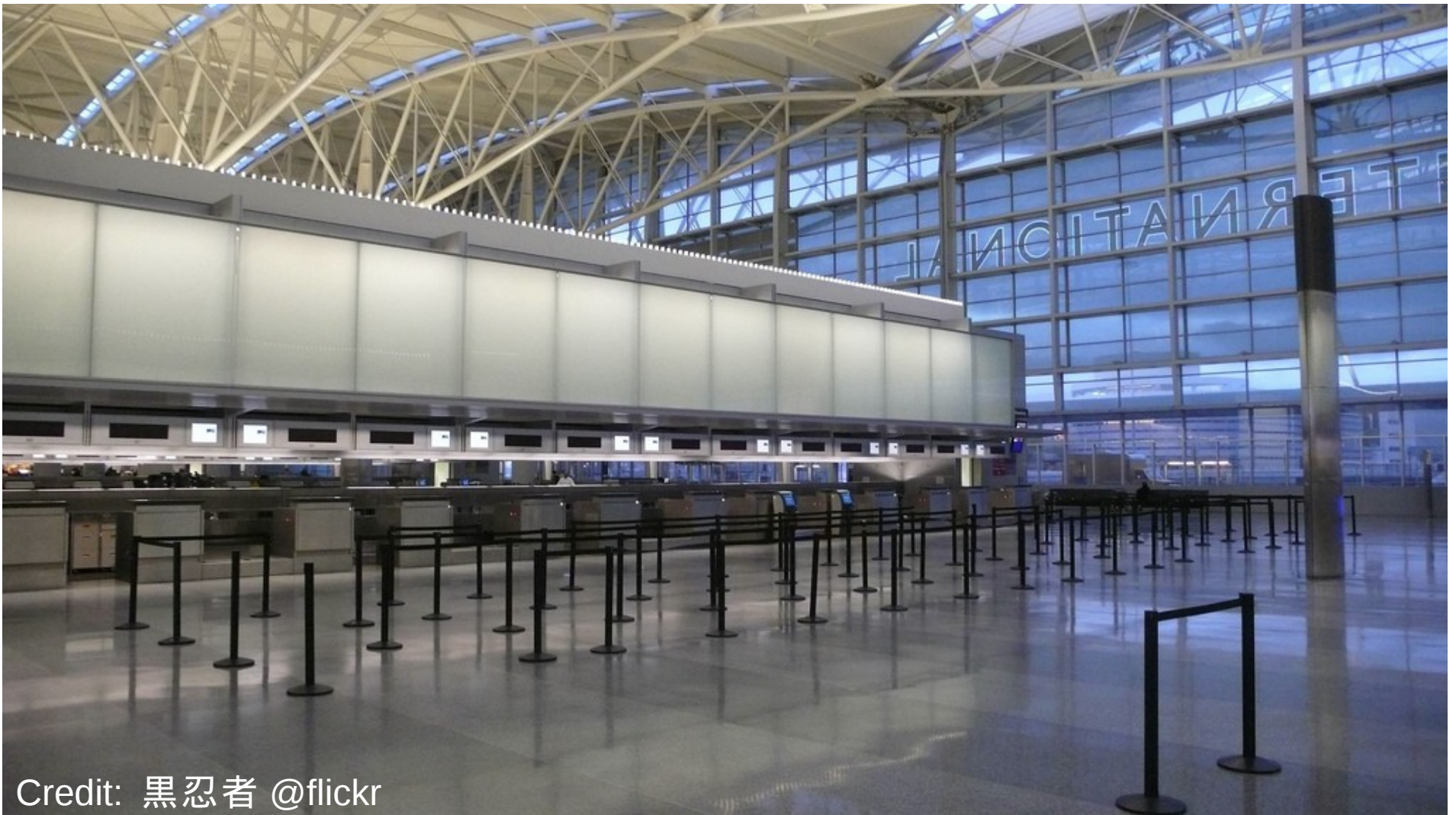
Credit: Florenz Kley@flickr

Queues: dealing with uneven loads



Credit: [toronto_pcu@flickr](#)

Always fast-enough too expensive



Credit: 黒忍者 @flickr

Threads and ThreadPools

- A thread is the smallest unit of independent CPU activity supported by Java (and most OSes)
 - Creating a thread is (relatively) expensive
 - A thread-pool allows a thread to be used for multiple tasks
-

Queue + ThreadPool



Credit: [mimi anderson@flickr](#)

Queues and dCache



Credit: **Sarah Macmillan**@flickr

Queue overflowing are a symptom



Credit: Alexandre Duret-Lutz@flickr

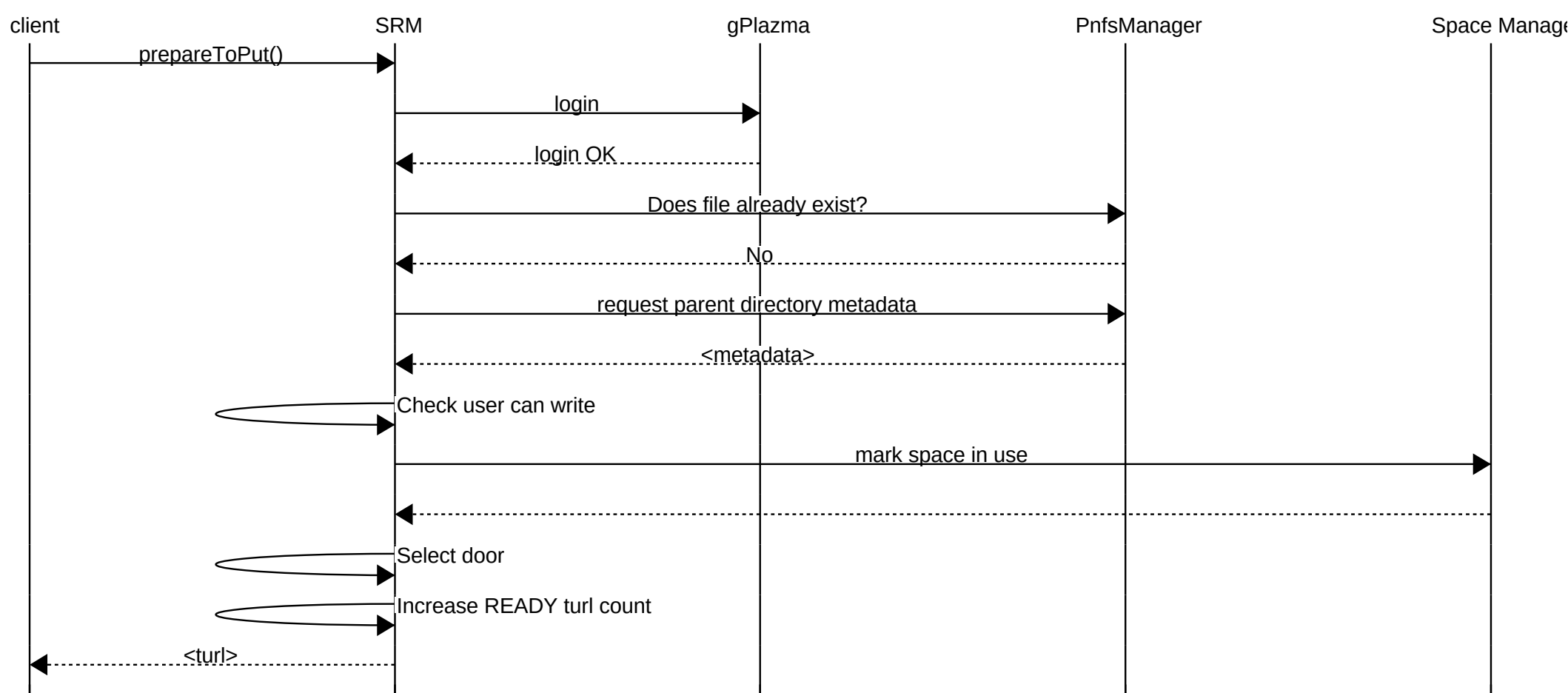
Queues and dCache

- Impossible to get a comprehensive talk on this subject
 - It's far, far to big a topic!
 - Instead, take a **worked example**:
 - Uploading a file using SRM and FTP
 - Client chooses to write file into a specific space-reservation
 - File is custodial/nearline, so flushed to tape
 - With the following limitations:
 - Everything runs smoothly (no errors)
 - only presenting some of the interactions between components
 - Skip over some details, when they are unenlightening.
 - No advise on tuning
 - Many details are specific for this worked example.
-

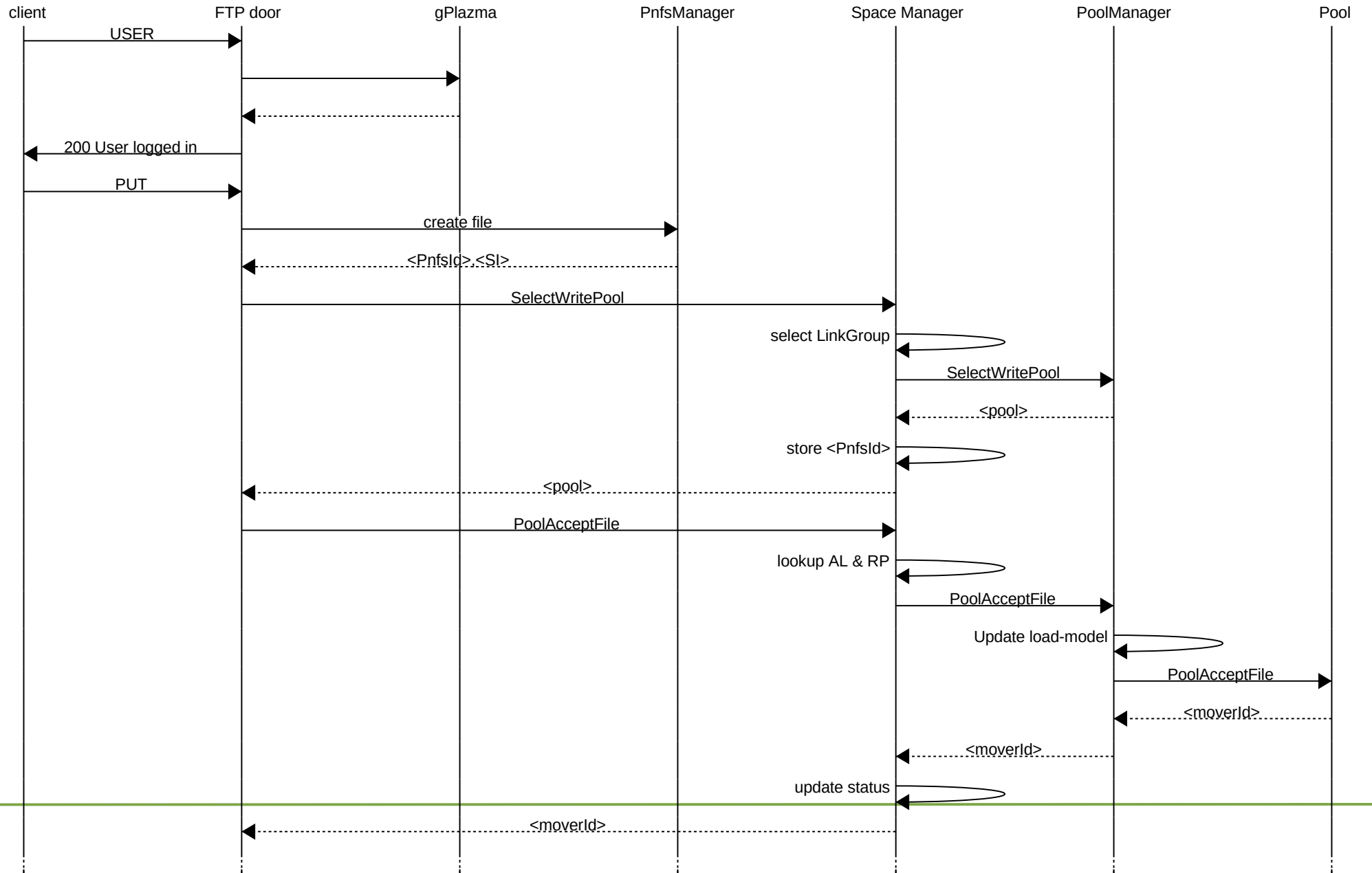
Helicopter view

- Client connects to...
 - SRM and issues a prepareToPut.
 - GridFTP door, preparing dCache for upload
 - Pool, delivering data
 - SRM and issues a putDone command.
 - Independent from the putDone, the pool flushes file to tape
-

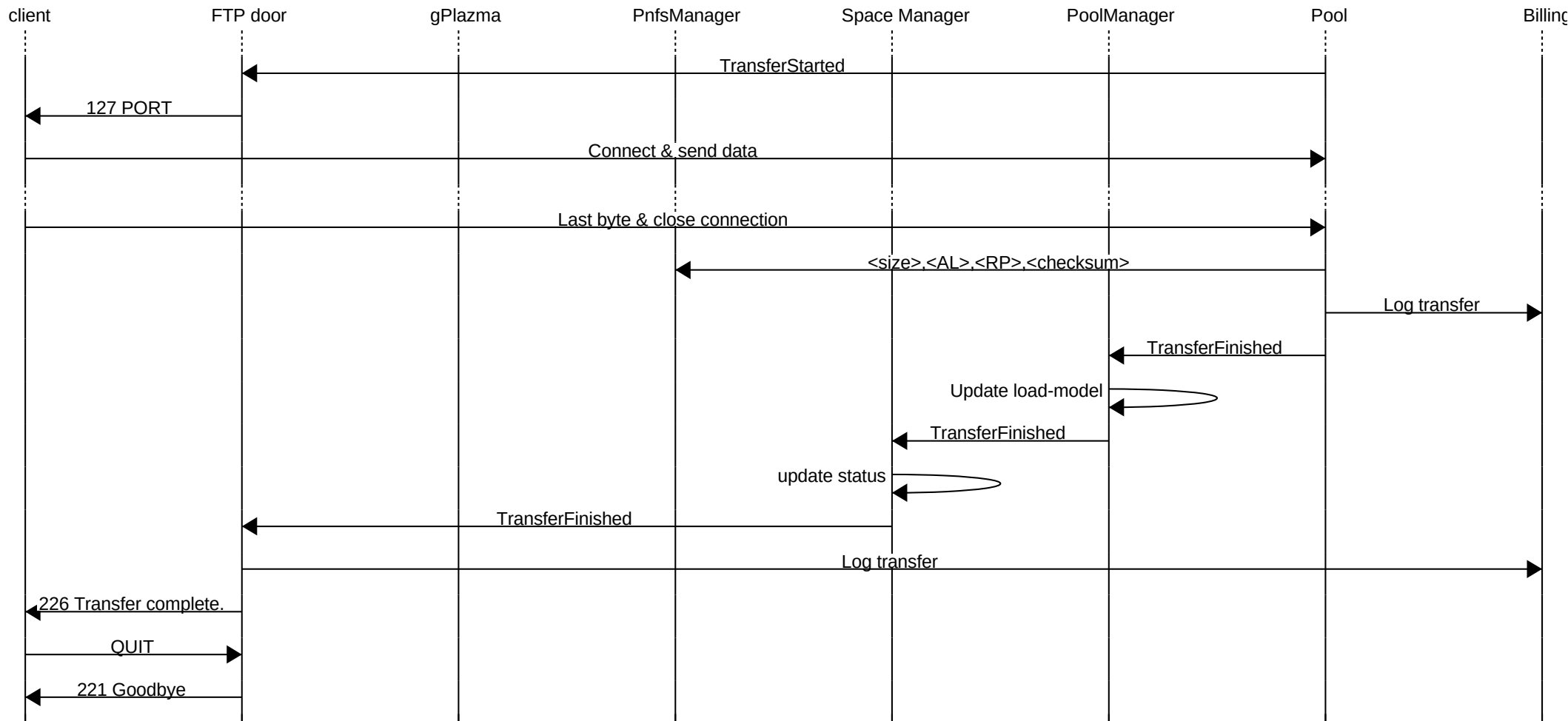
The communication: srmPrepareToPut



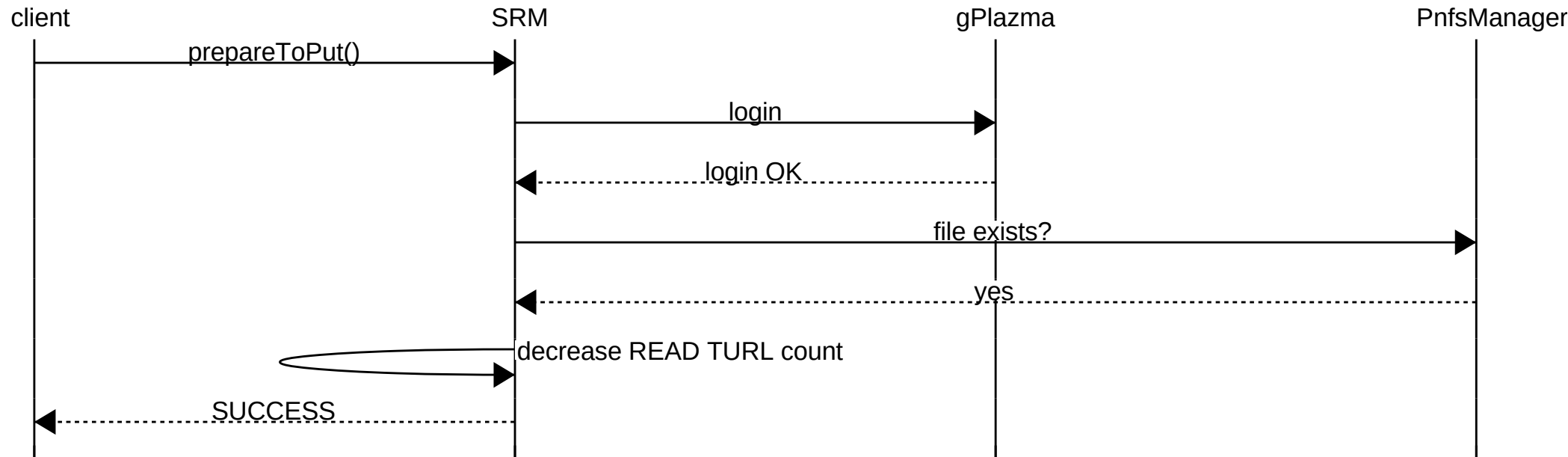
Communication: FTP (part 1)



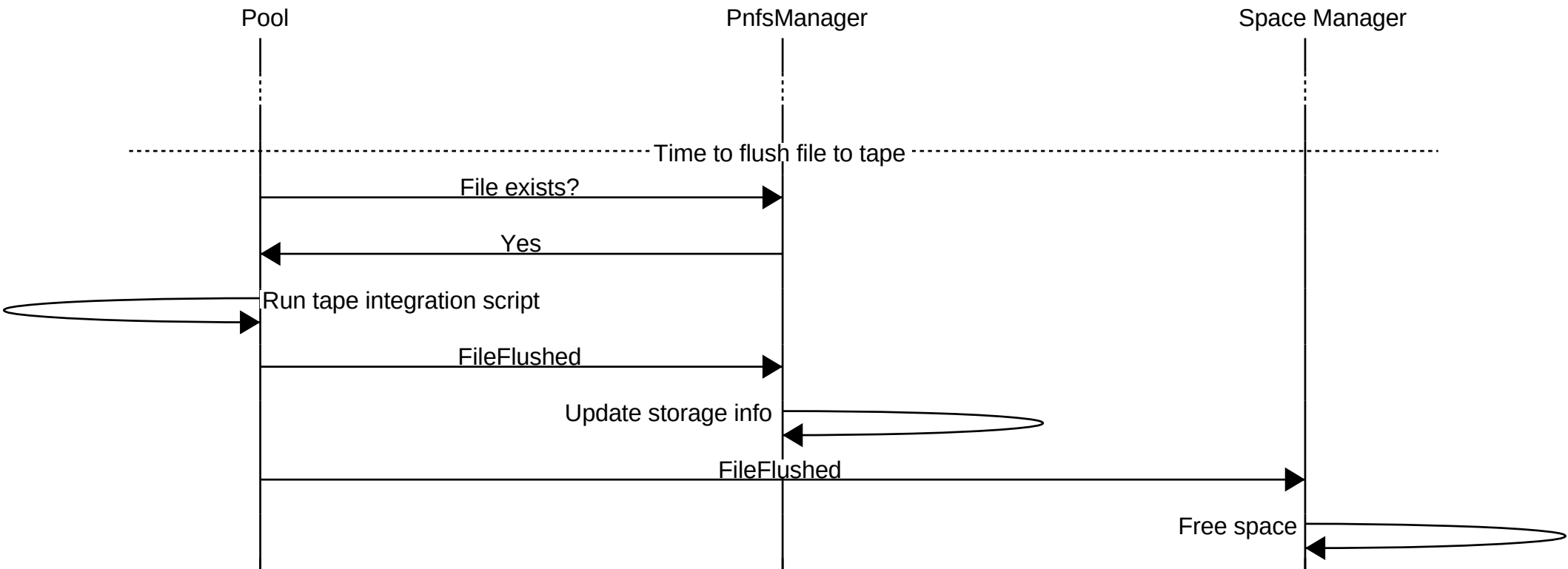
Communication: FTP (part 2)



Communication: srmPutDone

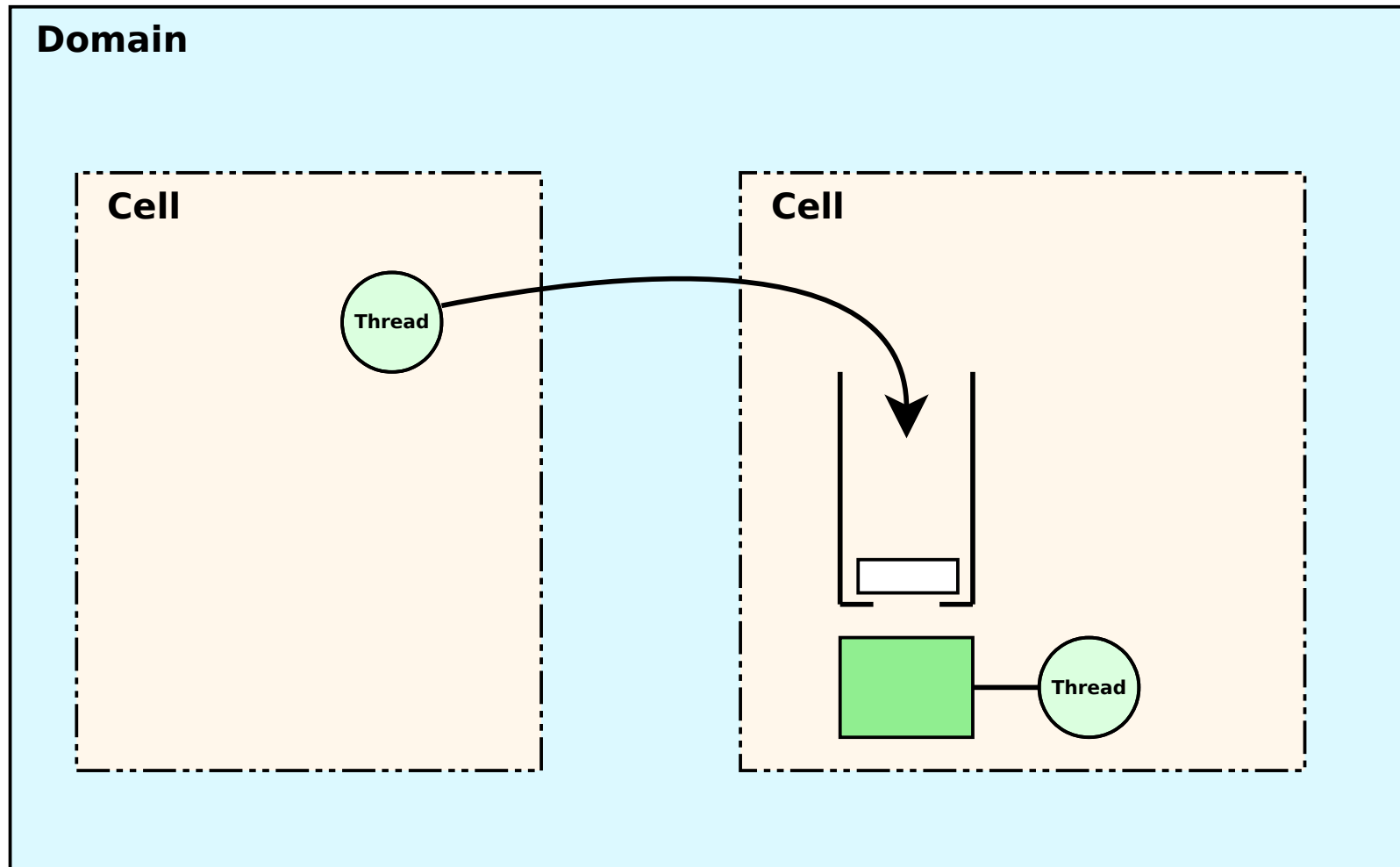


Communication: pool-flush

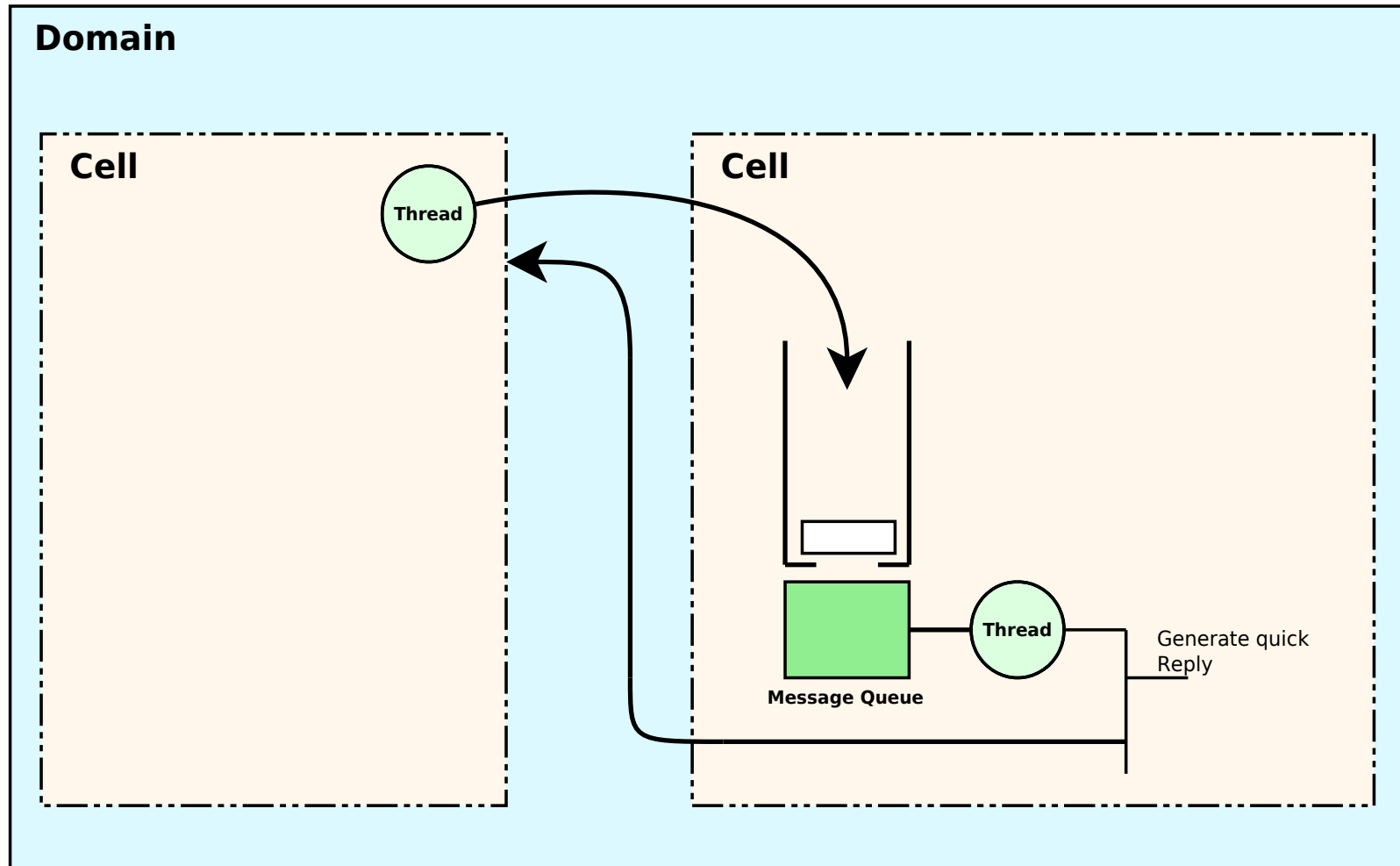


Queues: messages

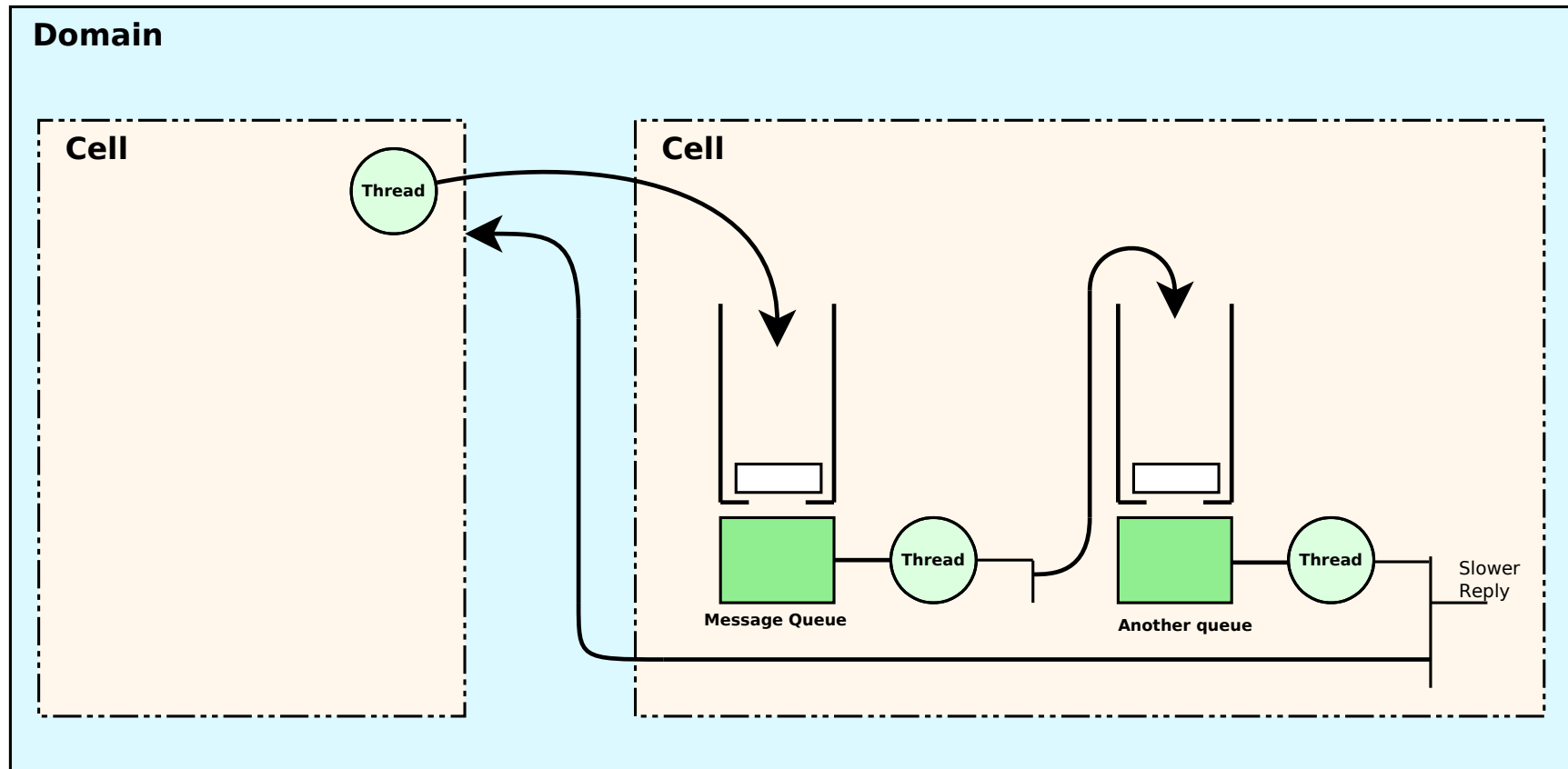
Messaging



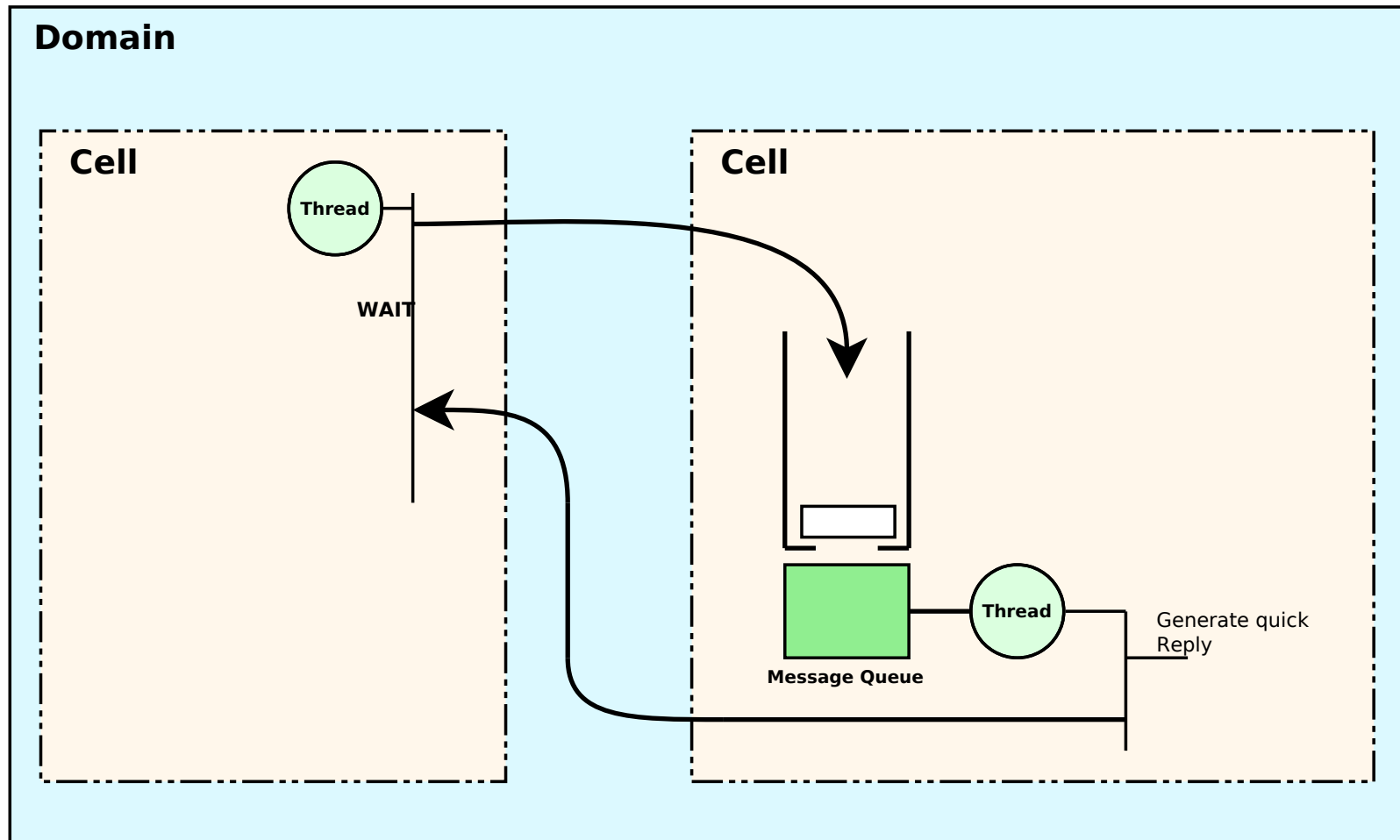
Messages: generating quick reply



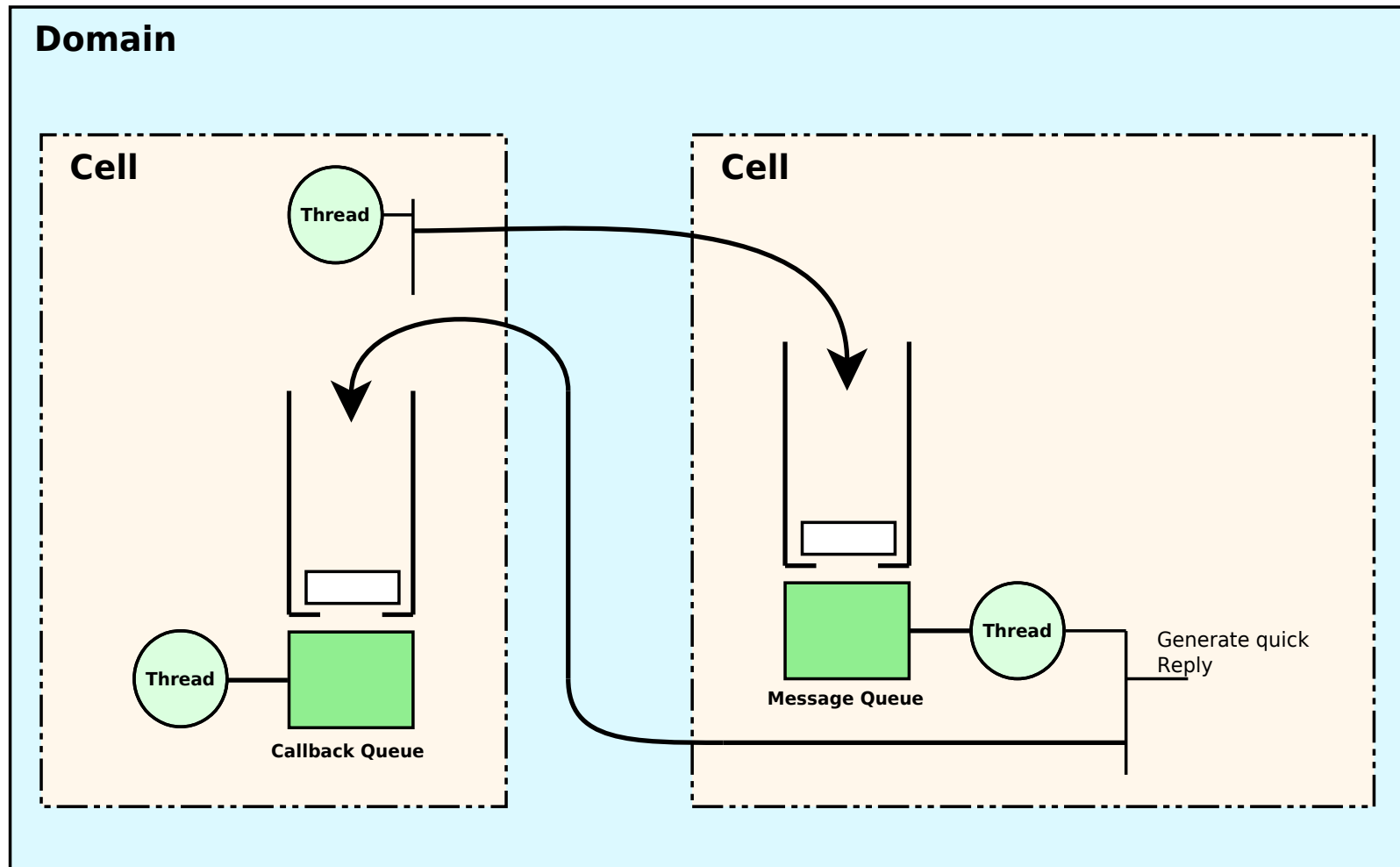
Messages: slower replies



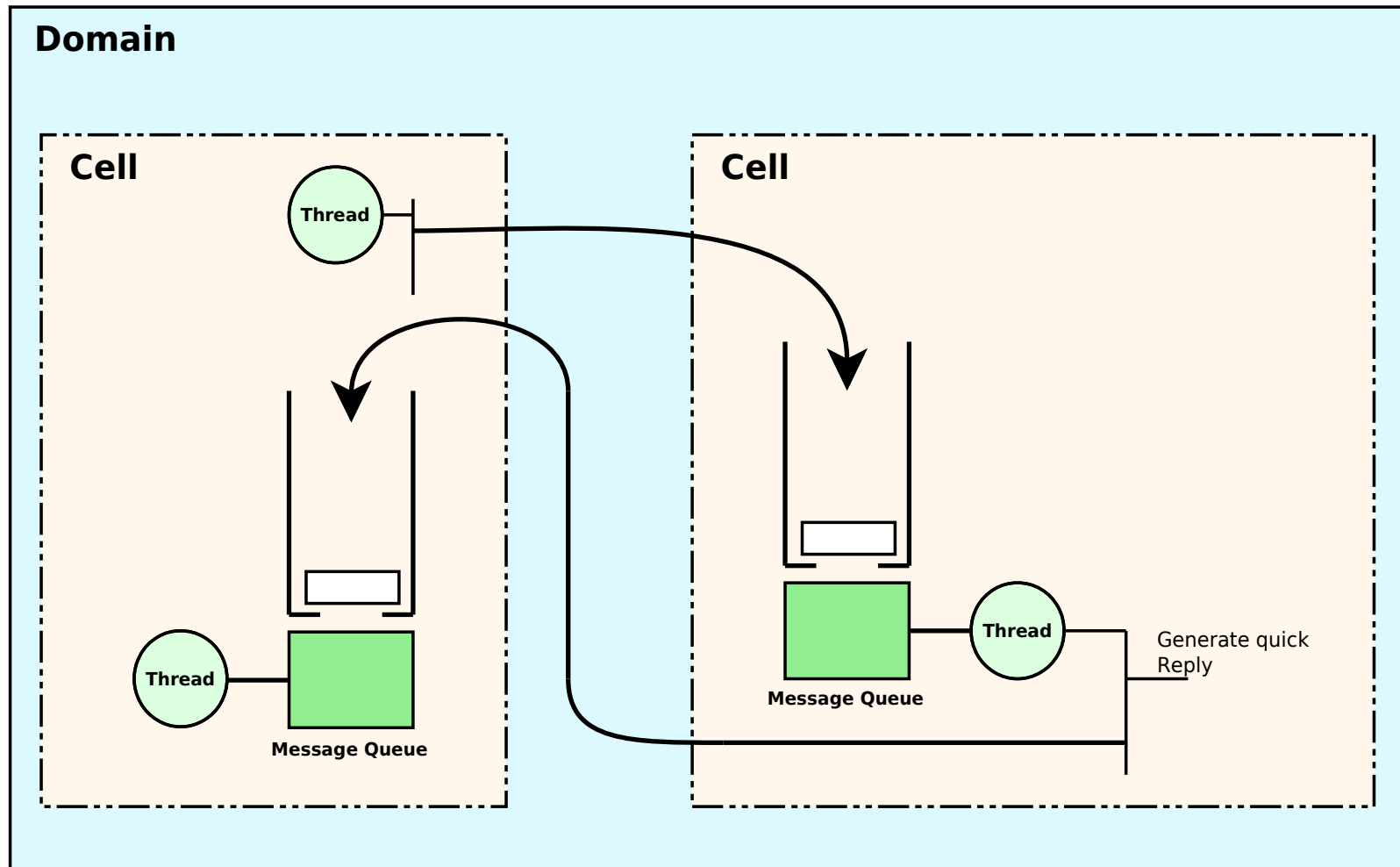
Messages: replying (blocking)



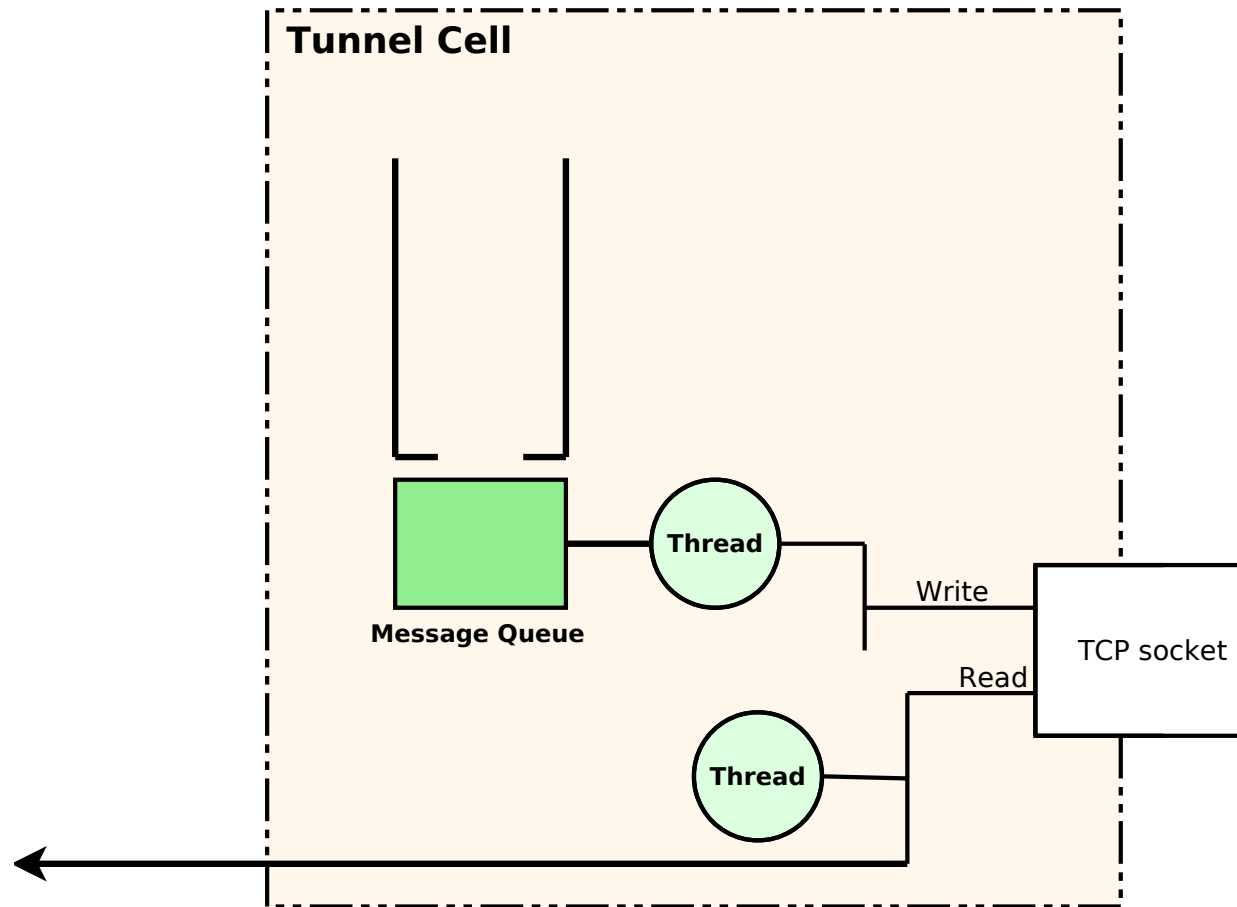
Messages: registered call-back



Messages: pure asynchronous

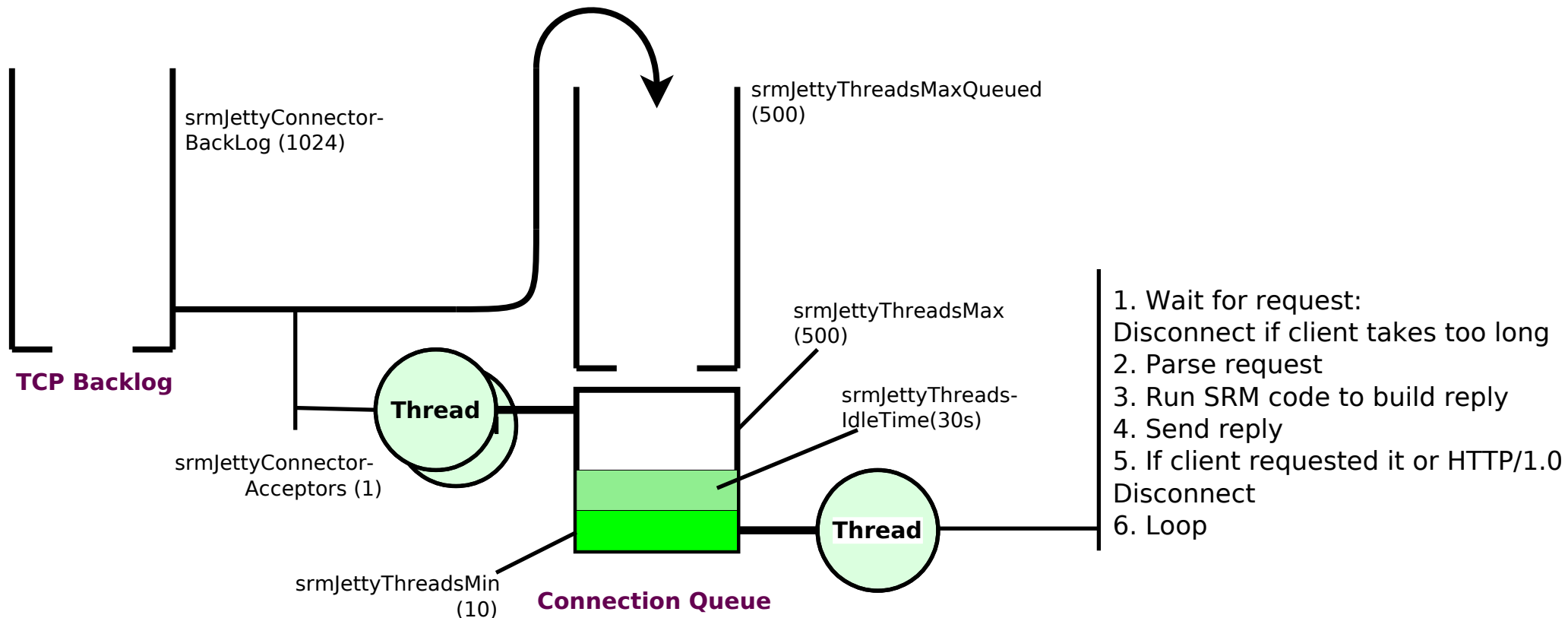


Messages: tunnel



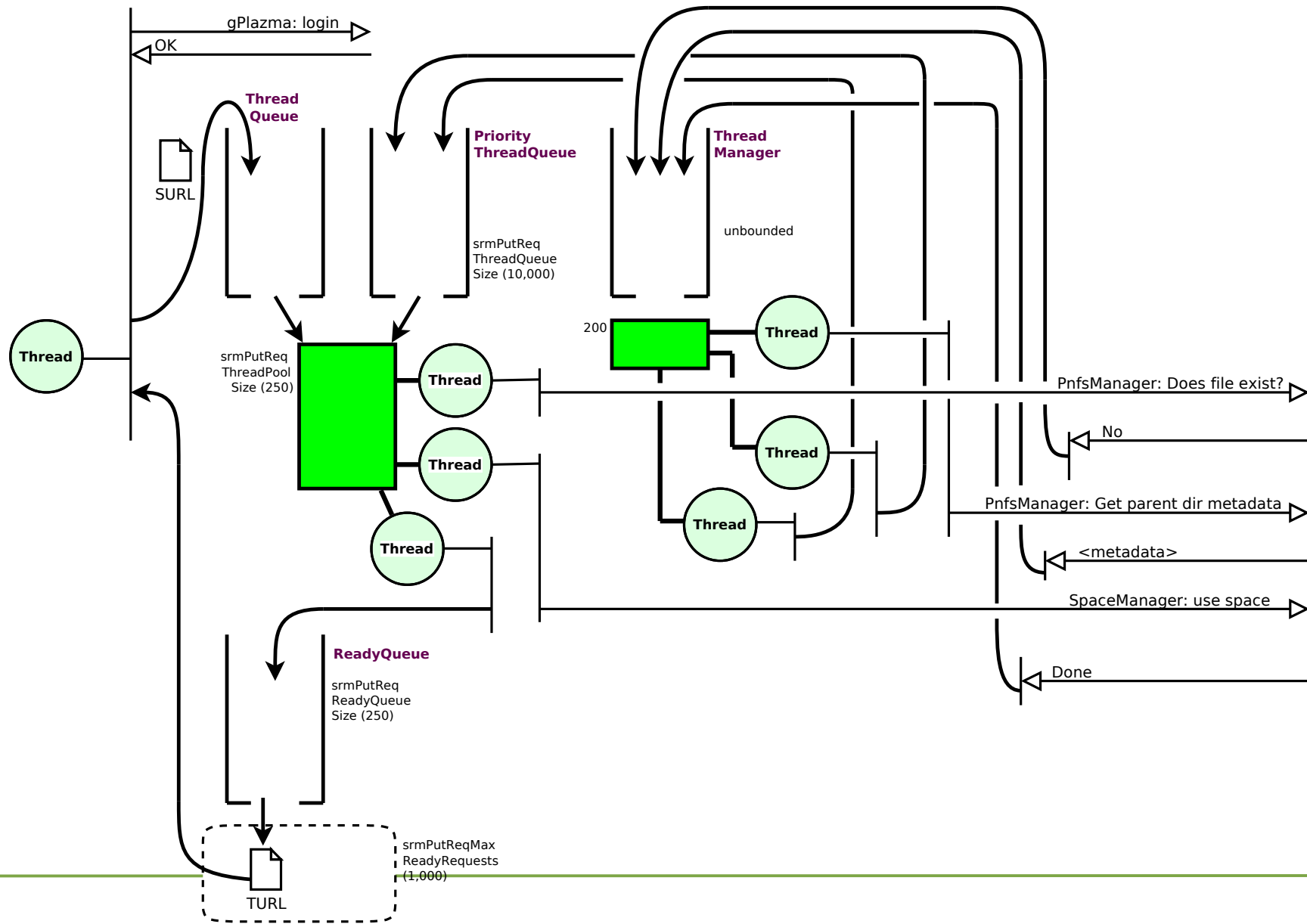
SRM: srmPrepareToPut

srmPrepareToPut: Jetty server

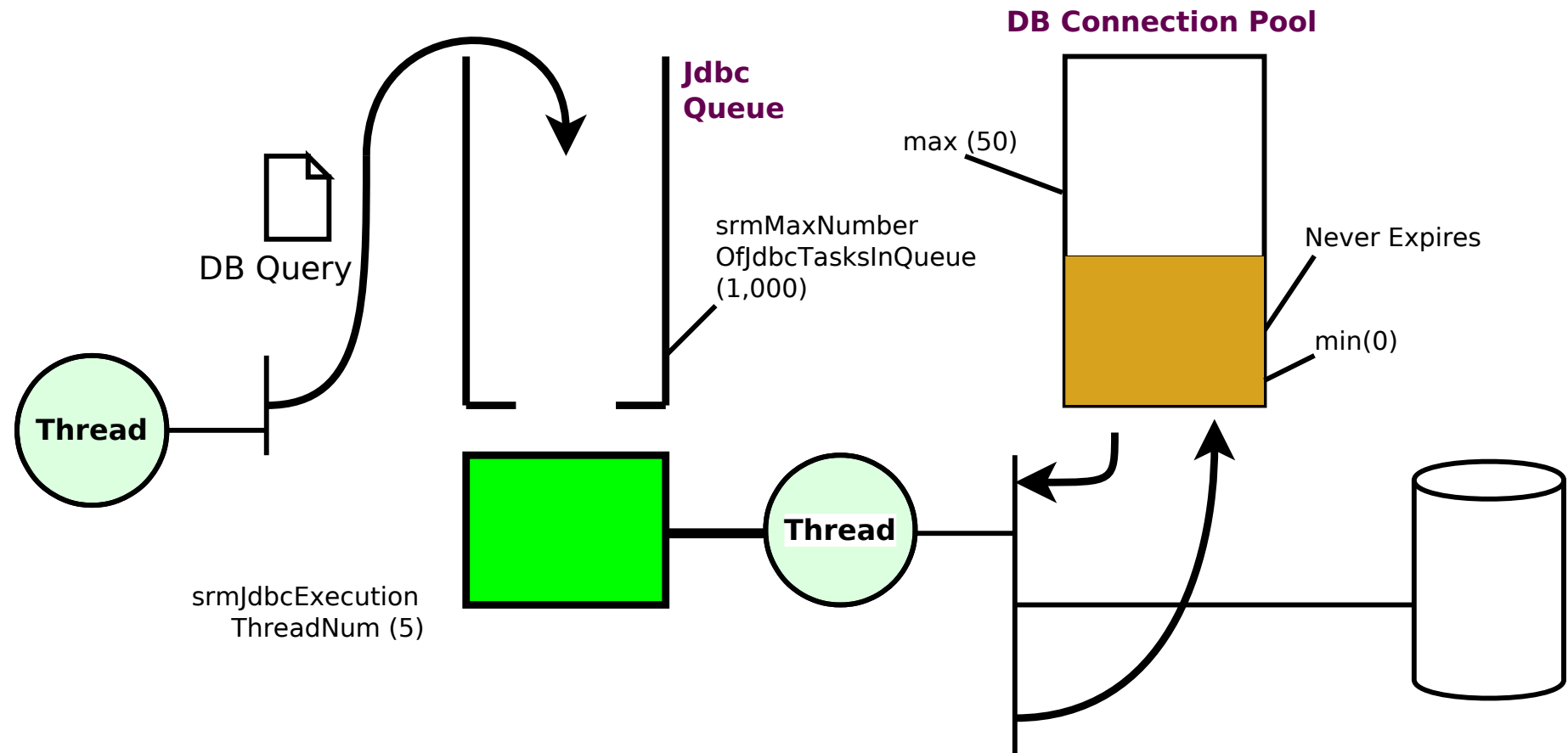


```

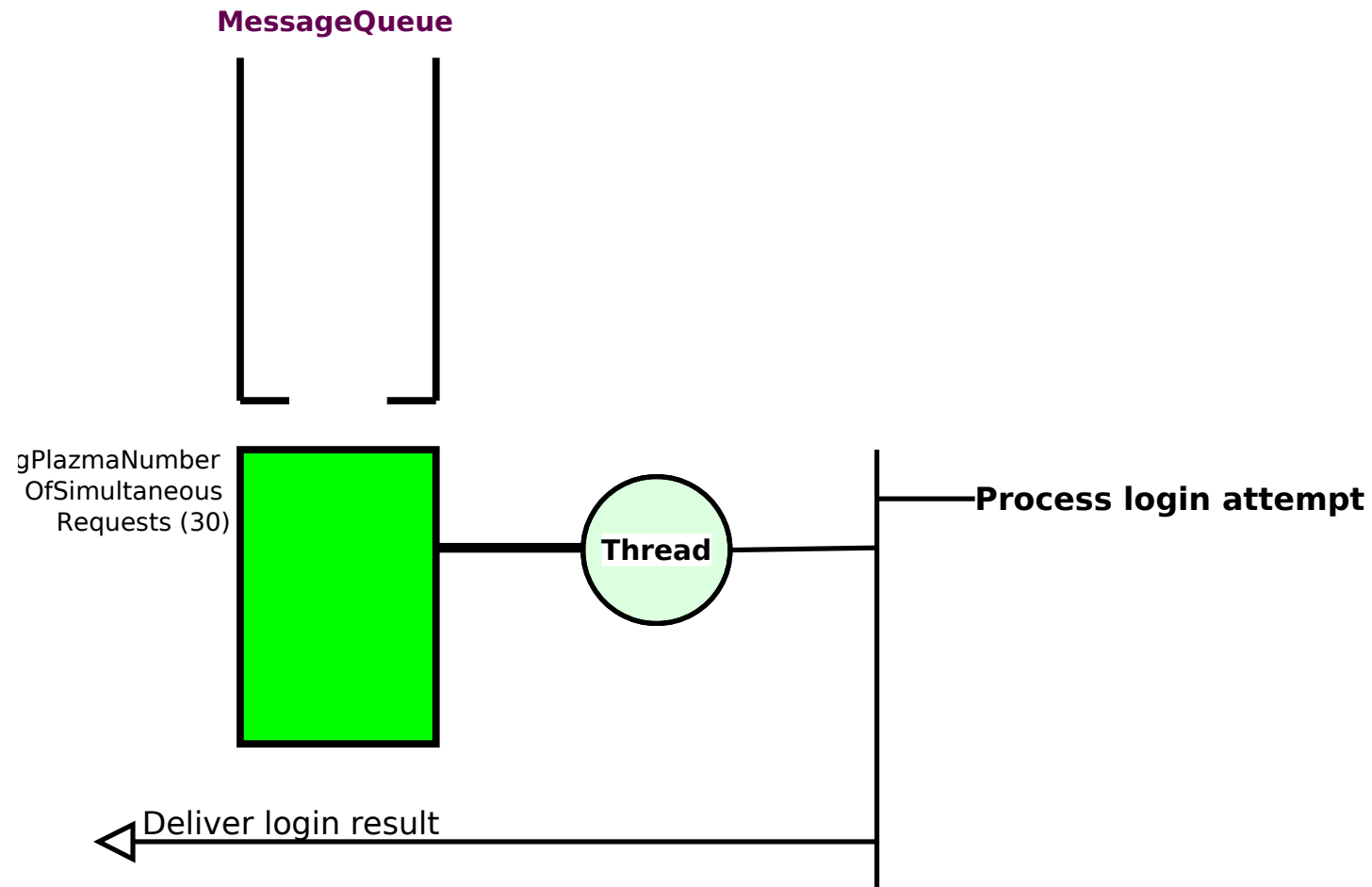
sequenceDiagram
    participant A
    participant B
    A->>B: gPlazma: login
    B-->>A: OK
  
```



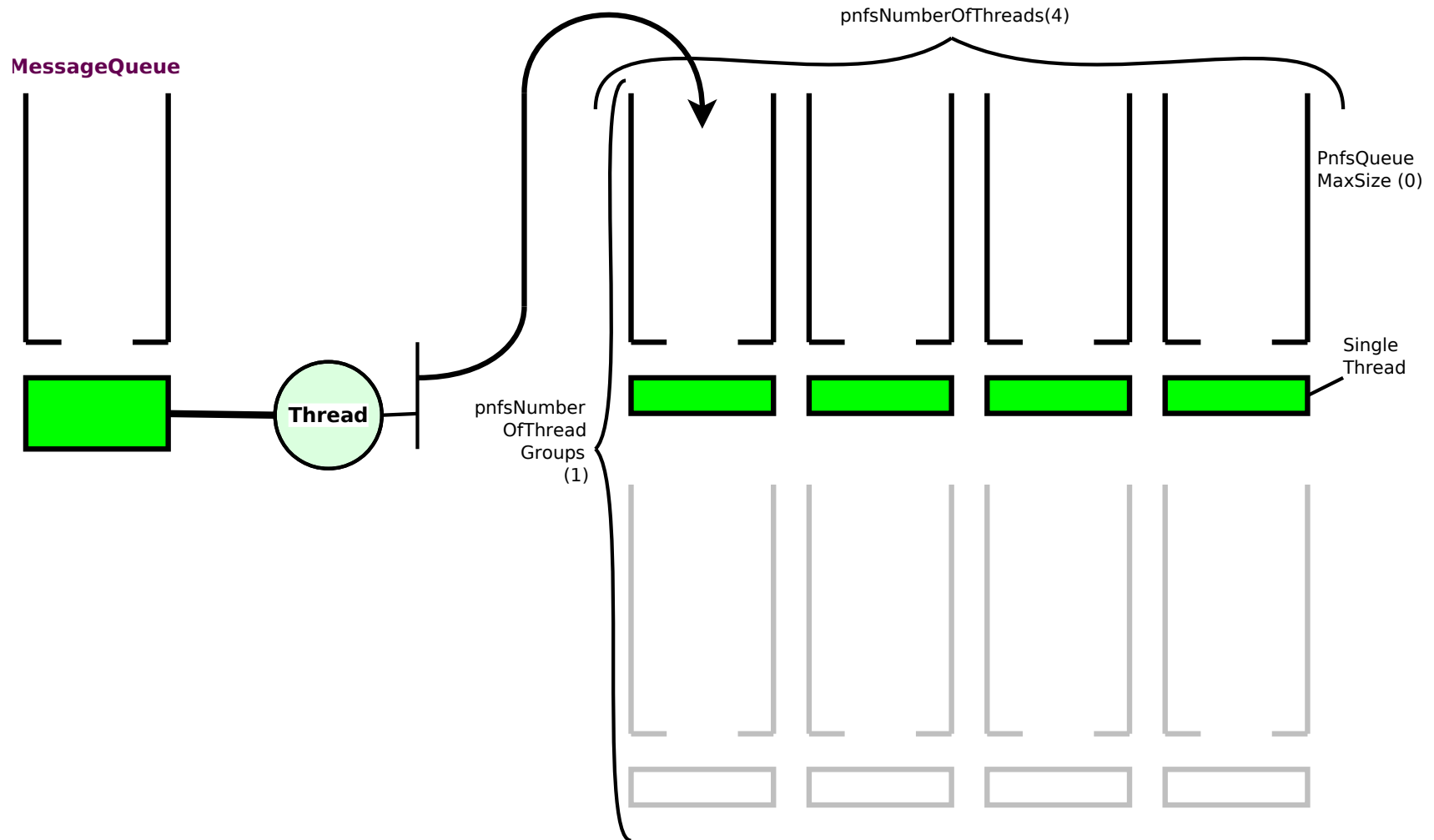
SRM and databases



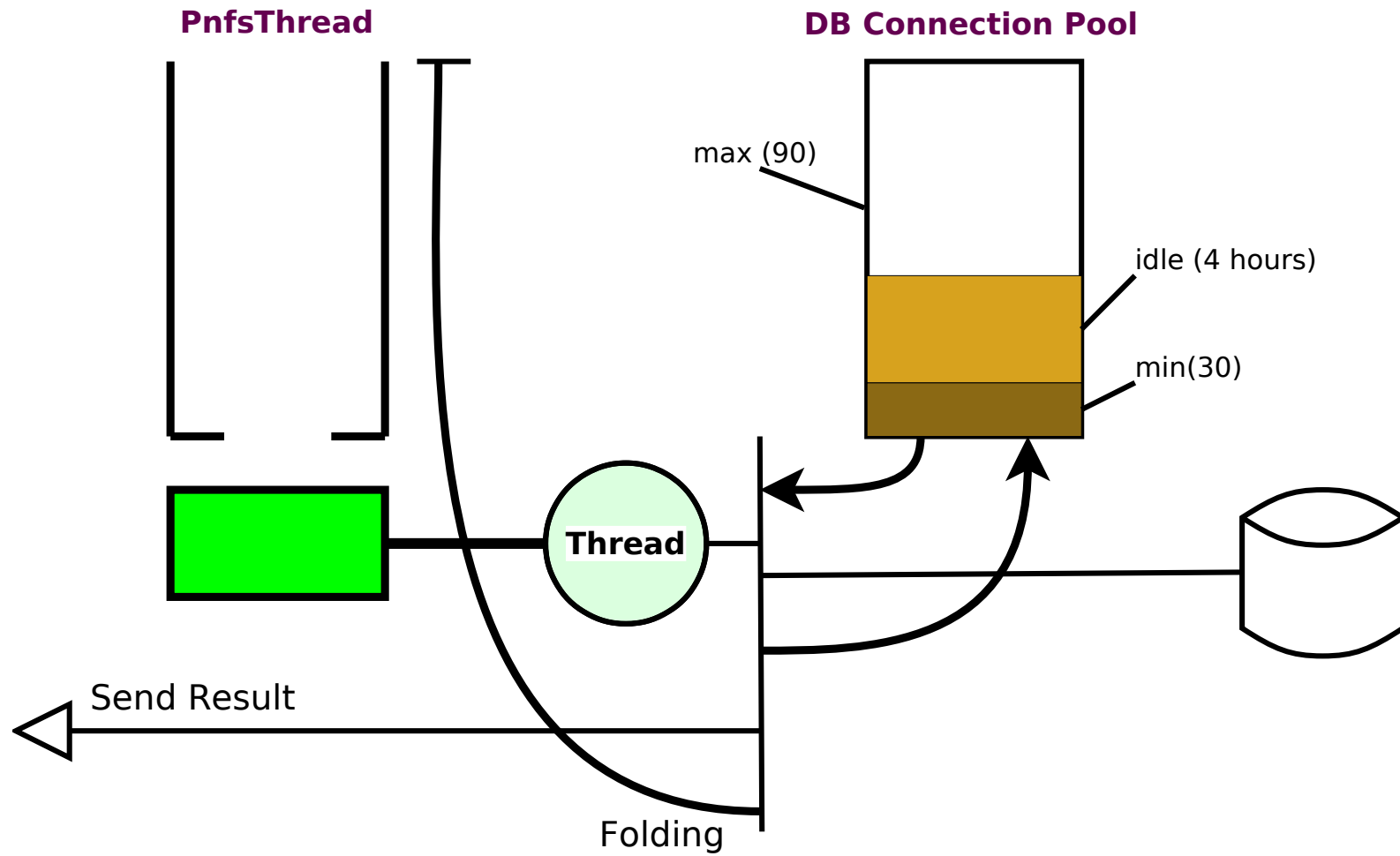
gPlazma



PnfsManager

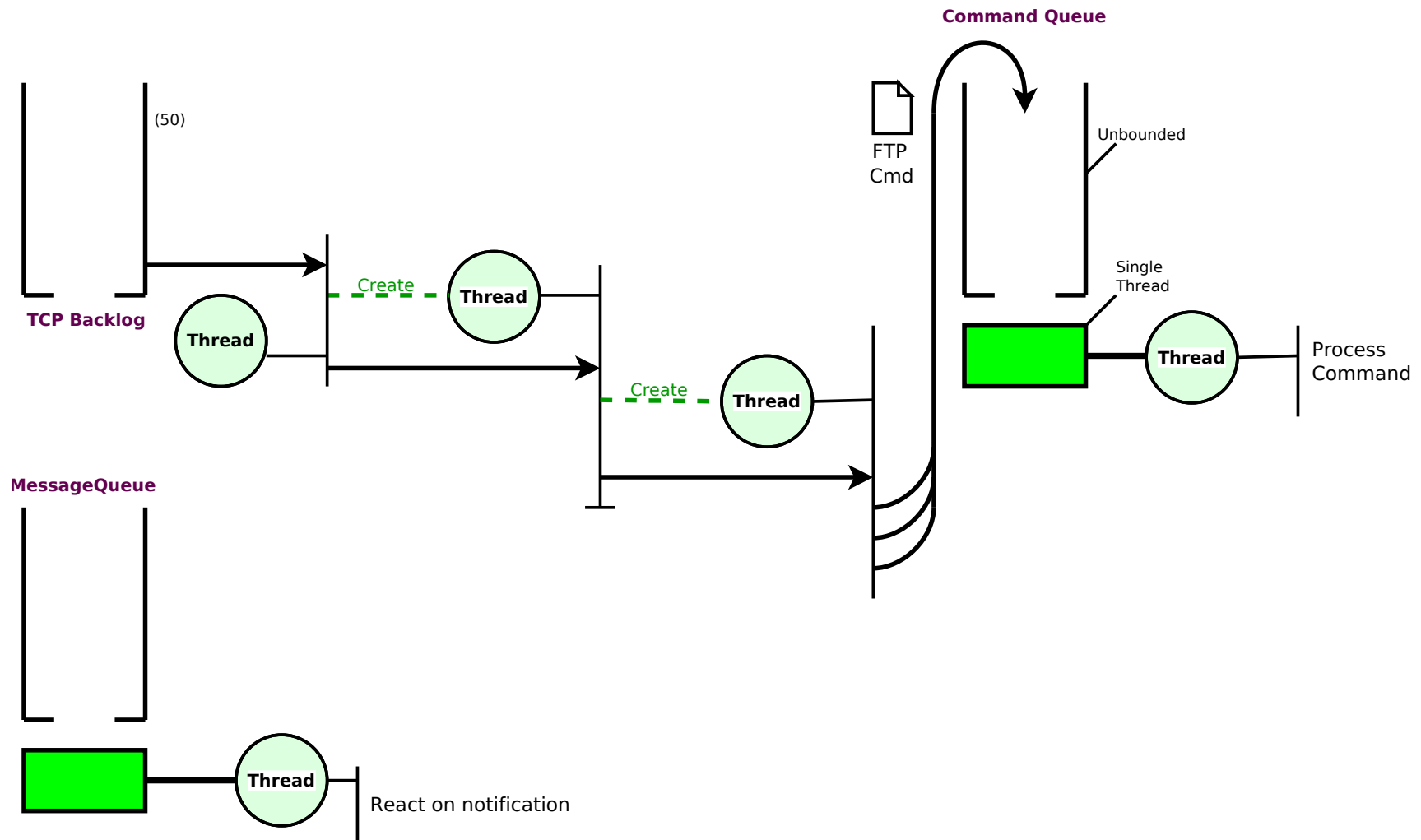


PnfsManager



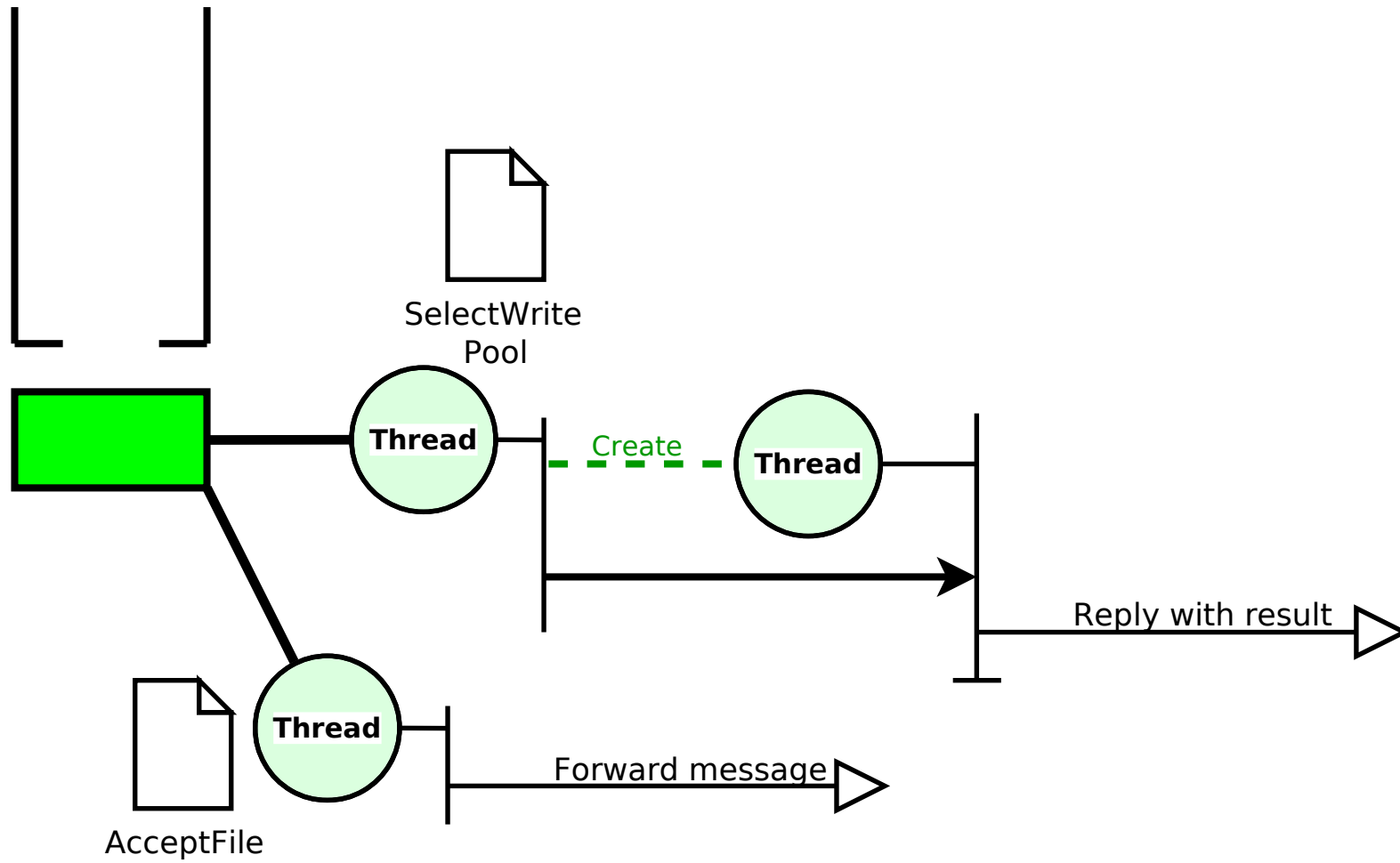
FTP upload and pool flush

FTP door

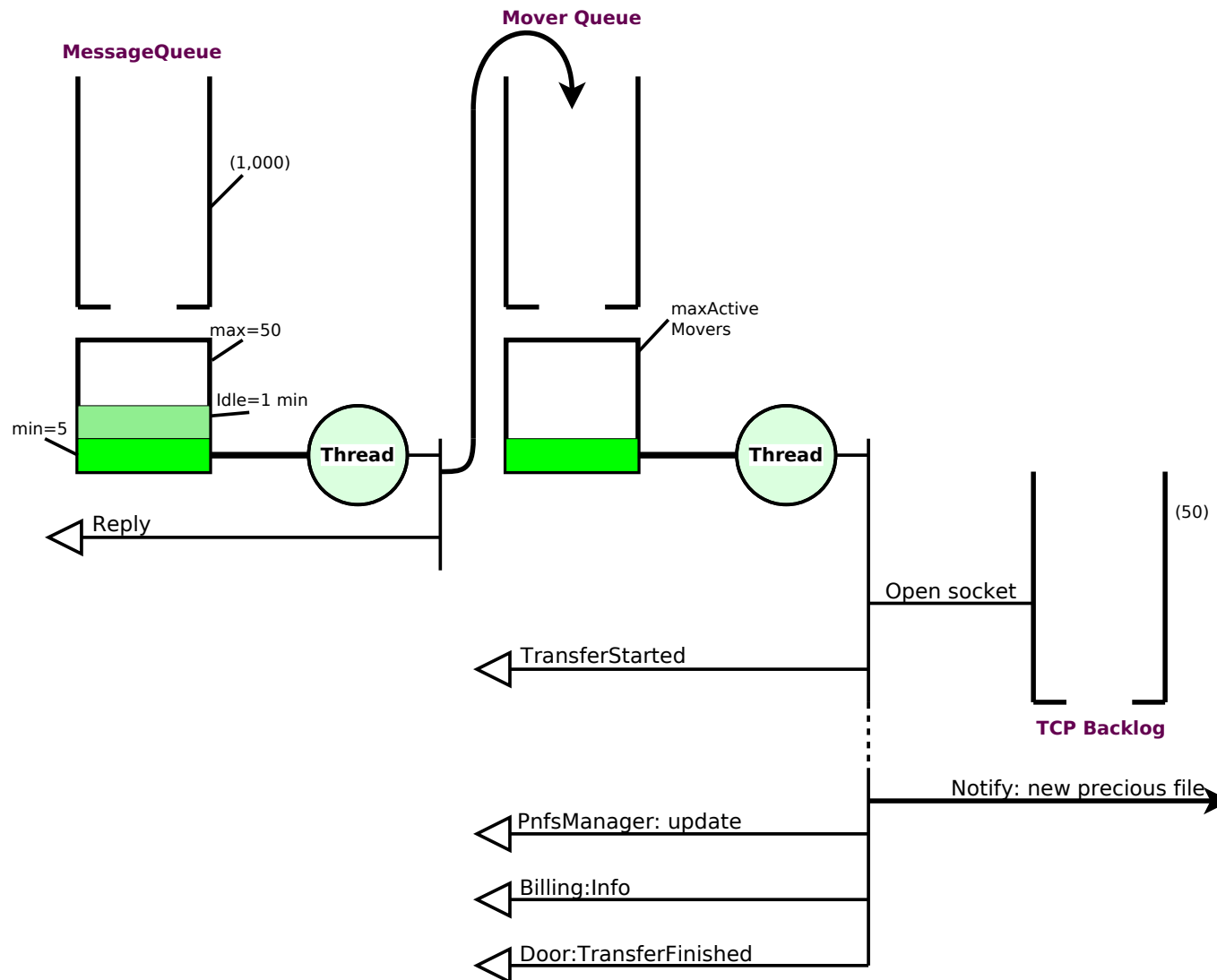


PoolManager

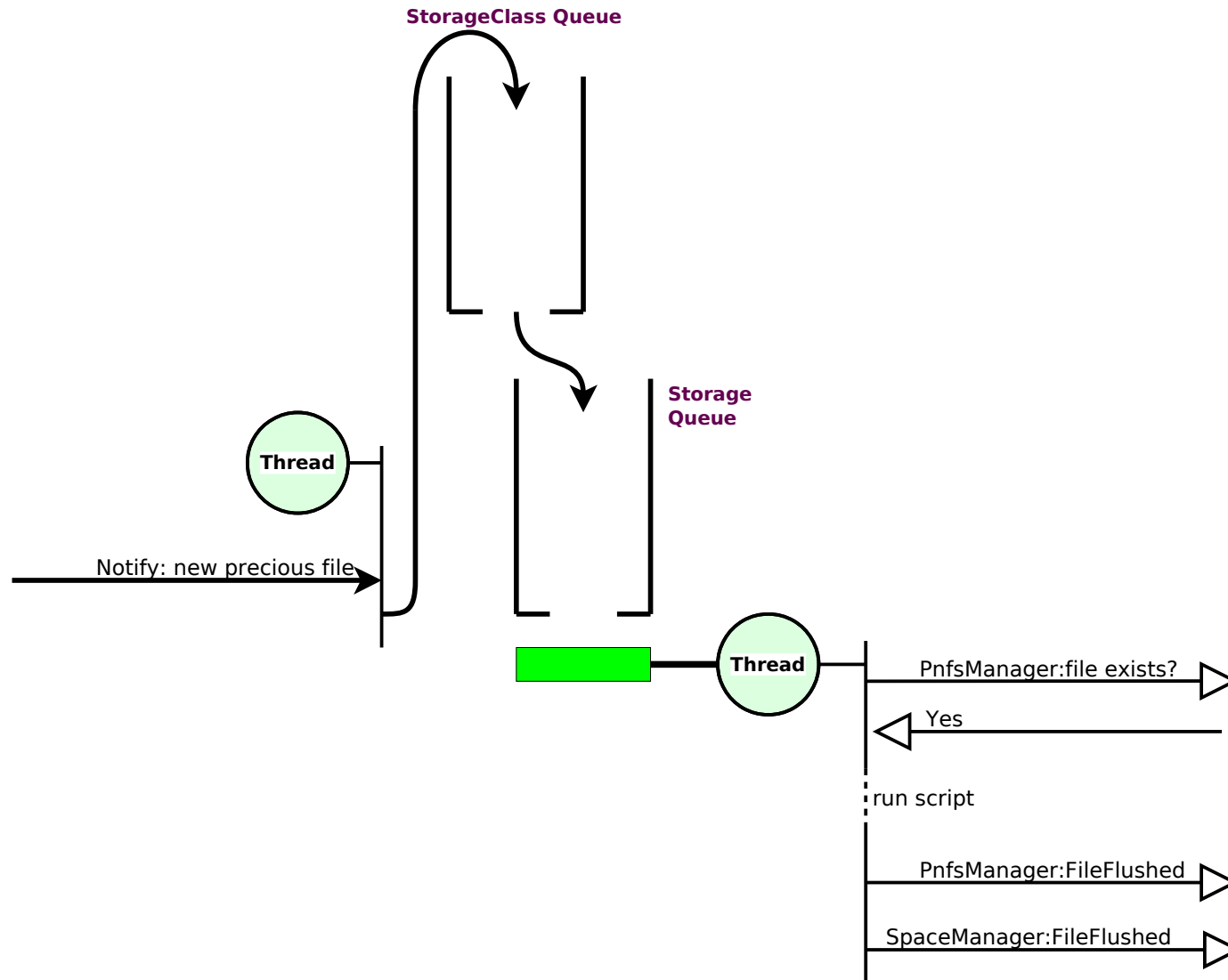
MessageQueue



Pool: AcceptFile

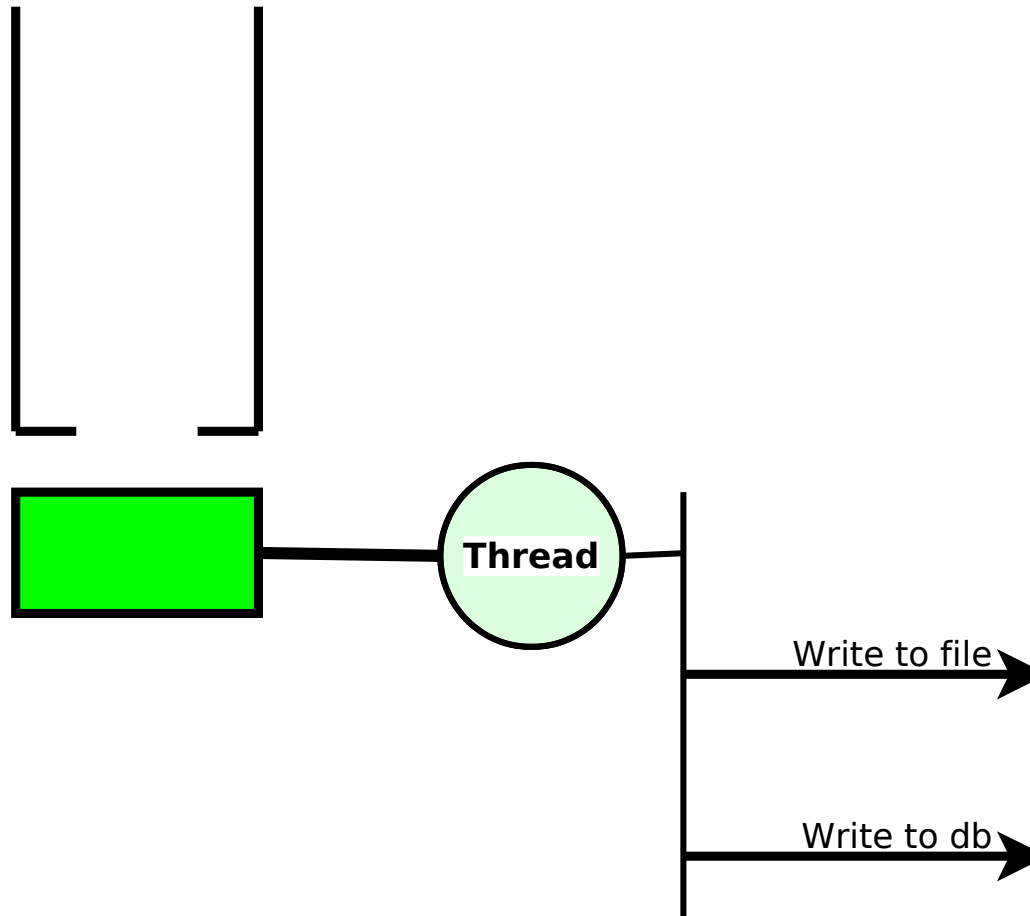


Pool: flush



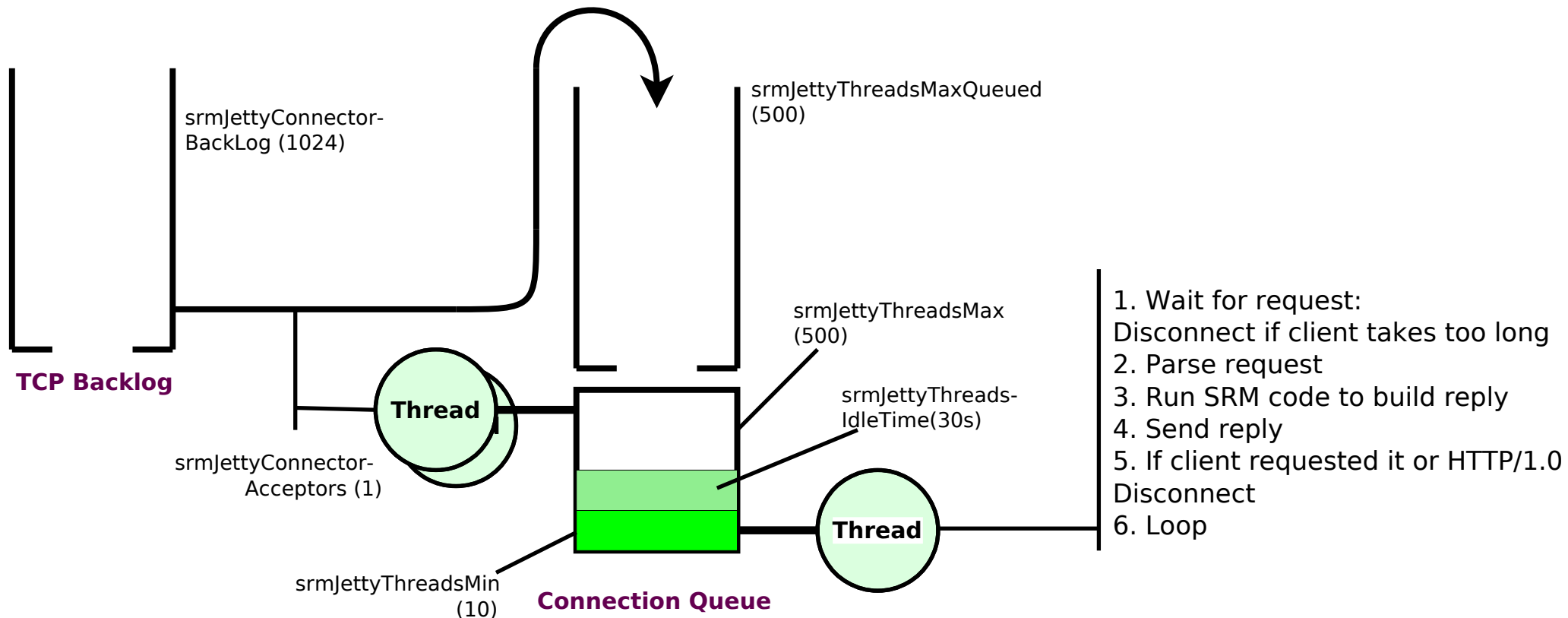
Billing

MessageQueue

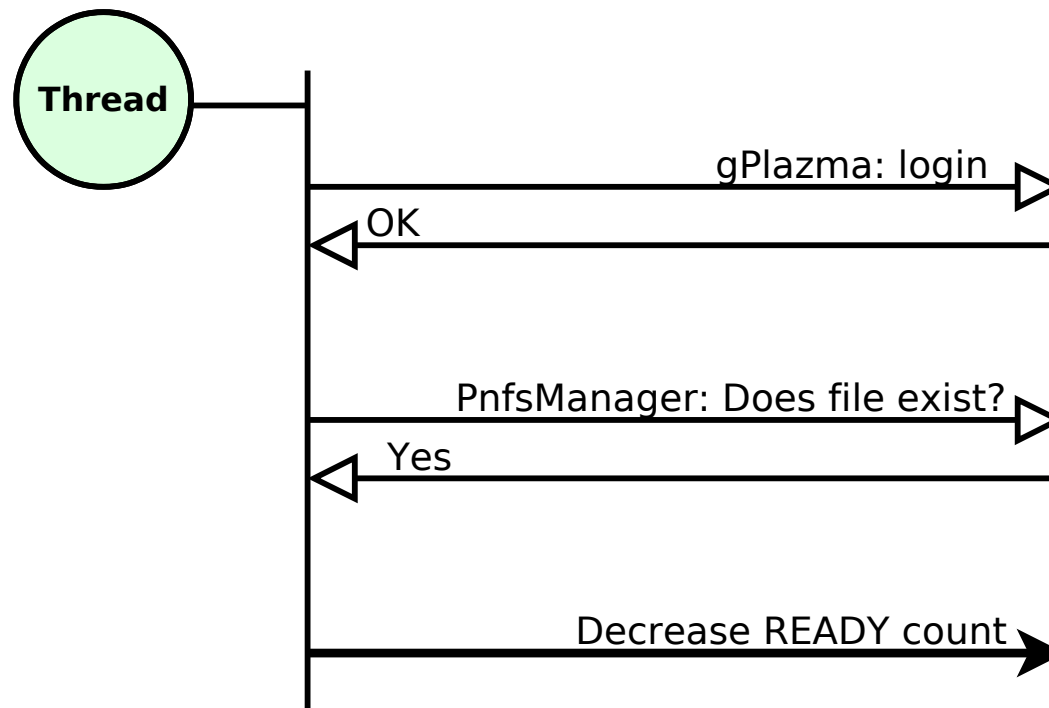


SRM srmPutDone

srmPutDone: Jetty server



srmPutDone: generating reply



Summary

- Queues are not evil
- Queues should be tuned for good performance, even under heavy load
 - Simply increasing queues isn't always optimal
- Quite a number of queues in dCache:
 - not all are obvious, not all are tunable