

## server

- gaspi context **fixed port**
    - proc\_init/term inter-rank via gaspi
    - configuration/information queries
    - setup of topology rpc ports
  - container manager
    - rpc to client (via socket)
    - “global topology” inter-rank via rpc
      - communication for global segments/allocs
    - memory manager
      - task queue + threads for transfers **#thread = #gaspi queue**
      - areas (segments)
        - beegfs **#fd fixed, lock instead of flush**
        - gaspi **comm buffers in user segment, non-shared**
        - shmem **local handled like global**
- uses

local via socket

**home-grown rpc  
fixed path**

inits

## client

- api (via socket)
    - global alloc: segment & alloc
    - put/get/stream
    - local segments **1 / worker, non-shared**
    - scheduling info (transfer cost)
- worker apps {  
-  
agent }

## issues

- inconsistent naming (local, global, segment, alloc, memory)
- segment parameters handled horribly in server, fine in client
- magic constants (timeouts, #threads, #fd)
- allocation policies for segments/allocations missing (multilevel, only-on-one-rank, auto, ...)