

- **Station based approach**

- The station is tasked with the coordination of movement
- The station carries the geometric information on the track.
- The stations' list carries the mover information (ticket)
- The geometric information has influence on the stations' behaviour
  - How many nests?
  - How long the distance until release
- The geometric information has influence on the movers' behaviour within a station
  - Forward move
  - Backward move

## ▪ Station based approach

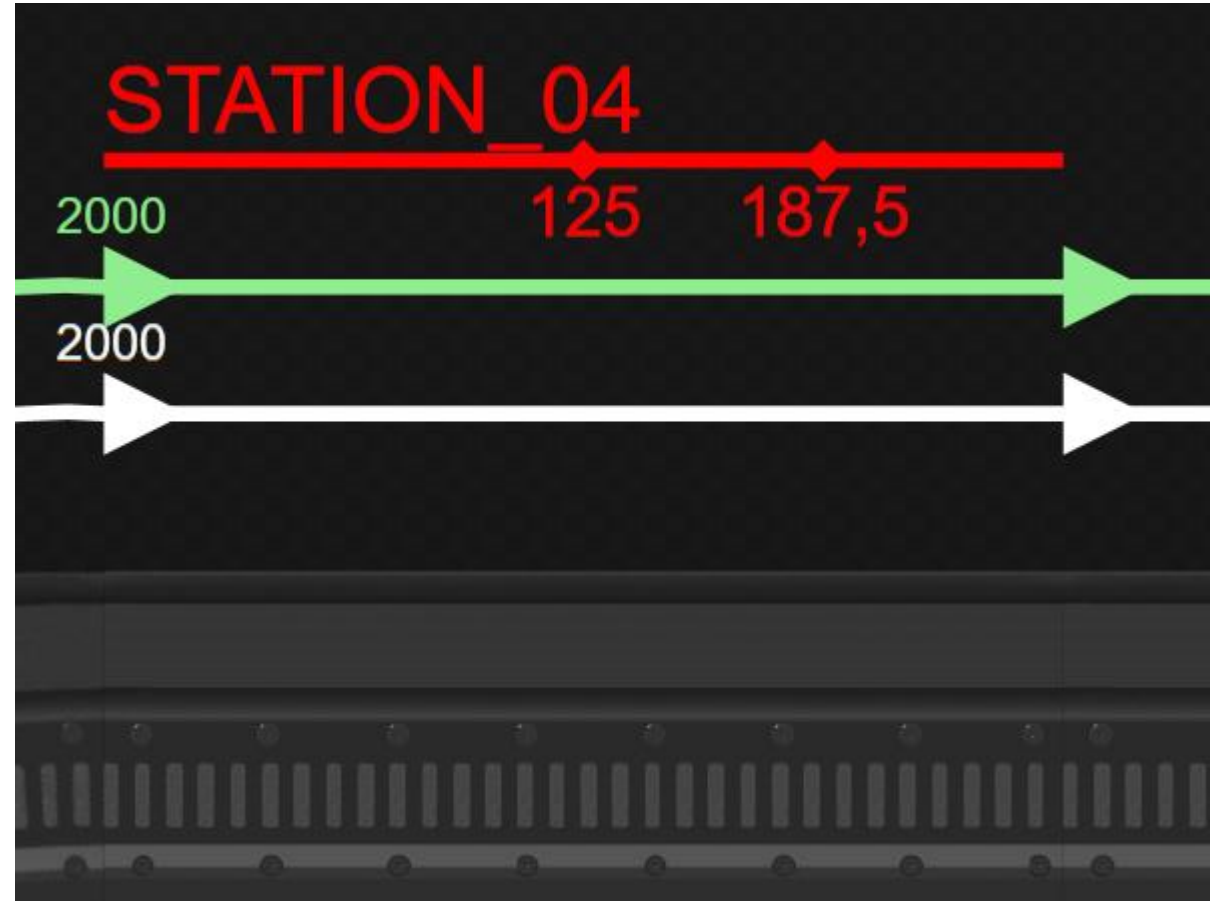
- The station is tasked with the coordination of mover targets
  - Halting in the states you answer to.
    - The station will wait here forever, isolated, doing nothing until commanded
  - Fall-through states you only see in the logs.
    - The station must be able to execute without delay until the next roadblock (either a state you answer to or an error exit)
- Handover is atomic and is either successful or leads to a fatal error
  - The linked lists are not using dynamic memory allocation.
    - Static memory allocation [0.. MAX\_LIST\_NODES] is ensuring that no memory fault or heap fragmentation can occur.

- **Station based approach**
  - Closed loop trade offs
    - No passing at all times.
      - This constraint is the main reason why lists may stay sorted simply by mapping the topology onto the logic.
  - High throughput and high speed
    - While a track switch enables highest flexibility regarding the mover targets
      - A track switch introduces a bottleneck which must be accounted for.

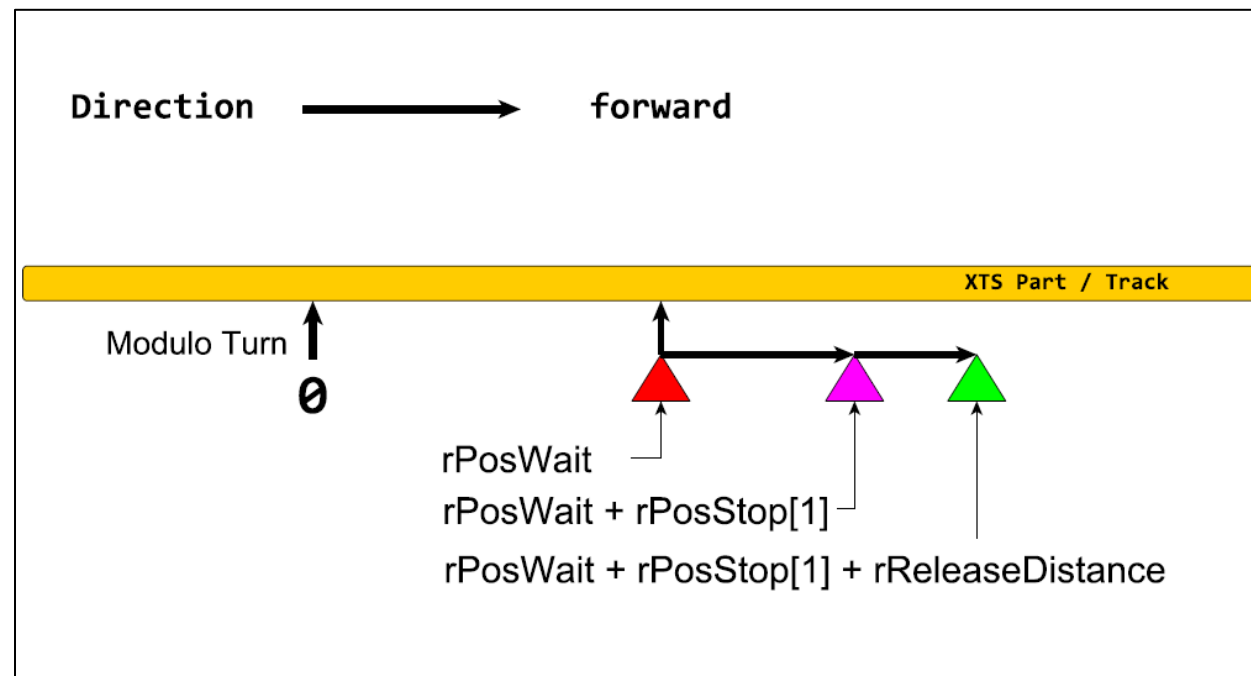
- Station based approach
- Example of a closed loop XTS:
  - **PROCESS** may have multiple stations
  - **STATION** may have multiple nests
  - **NEST** is a stop position for a mover



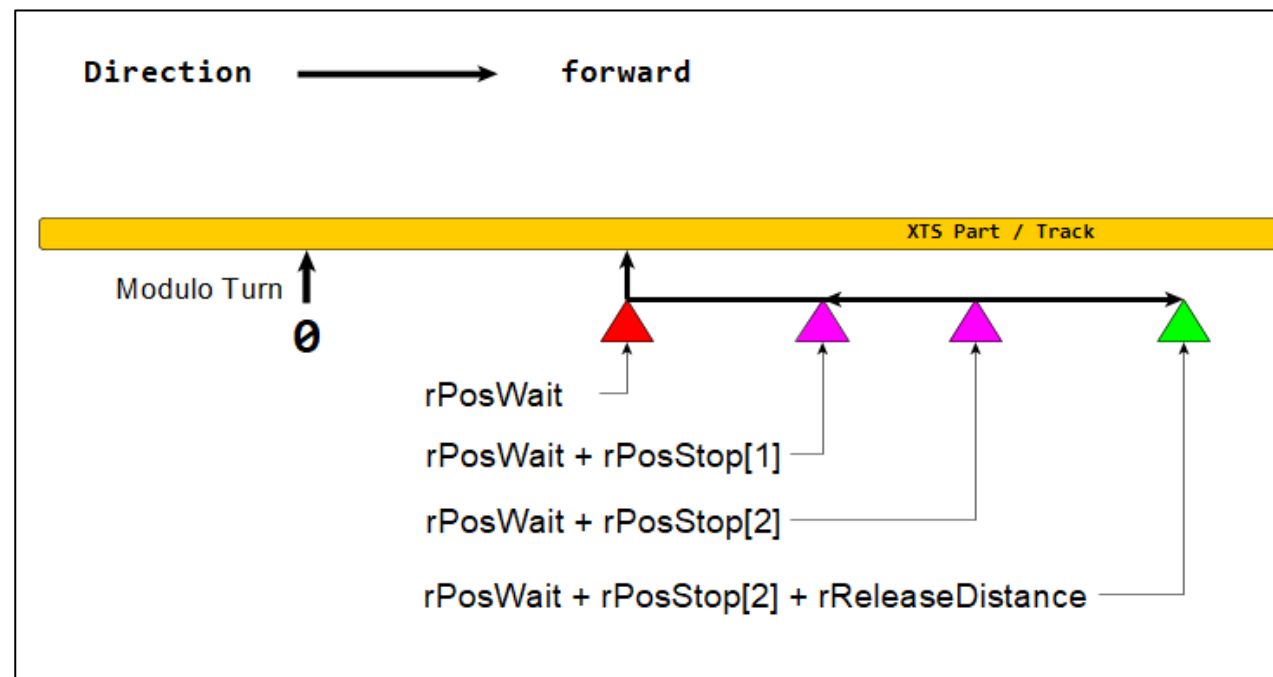
- Station based approach
- STATION geometry:
  - **WaitPos**: a position on the track where the station starts, and any other station may send a mover to.
  - **StopPos**: 1 to 8 possible **relative** positions a mover may stop at.
  - **ReleaseDistance**: distance a mover has to travel to logically leave a station.



- **Station based approach**
- **STATION geometry:**
  - **WaitPos:** a position on the track where the station starts, and any other station may send a mover to.
  - **StopPos:** 1 to 8 possible **relative** positions a mover may stop at.
  - **ReleaseDistance:** distance a mover has to travel to logically leave a station.



- **Station based approach**
- **STATION geometry:**
  - **WaitPos:** a position on the track where the station starts, and any other station may send a mover to.
  - **StopPos:** 1 to 8 possible **relative** positions a mover may stop at.
  - **ReleaseDistance:** distance a mover has to travel to logically leave a station.

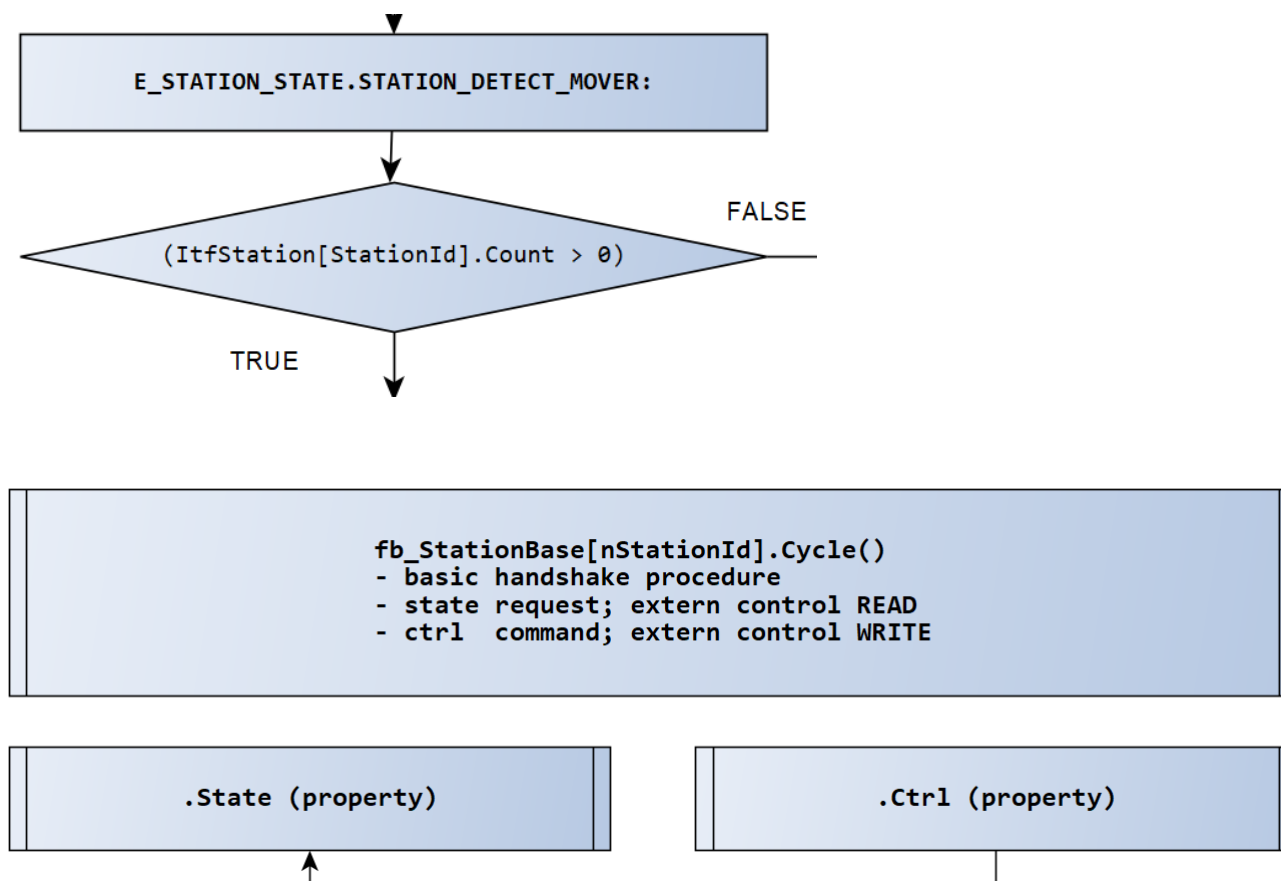




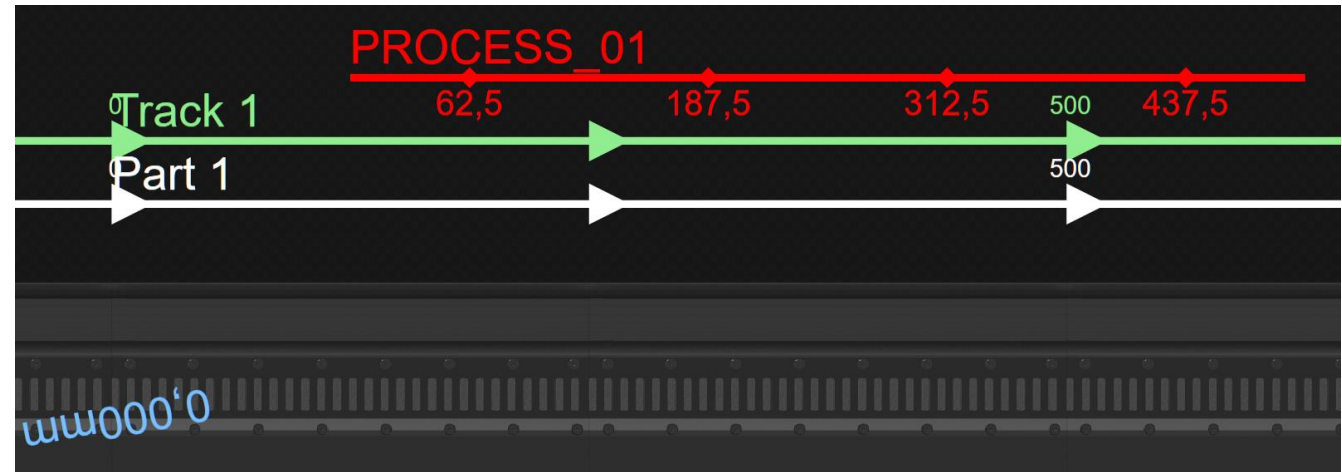
## ▪ Station based approach

## ▪ STATION operation:

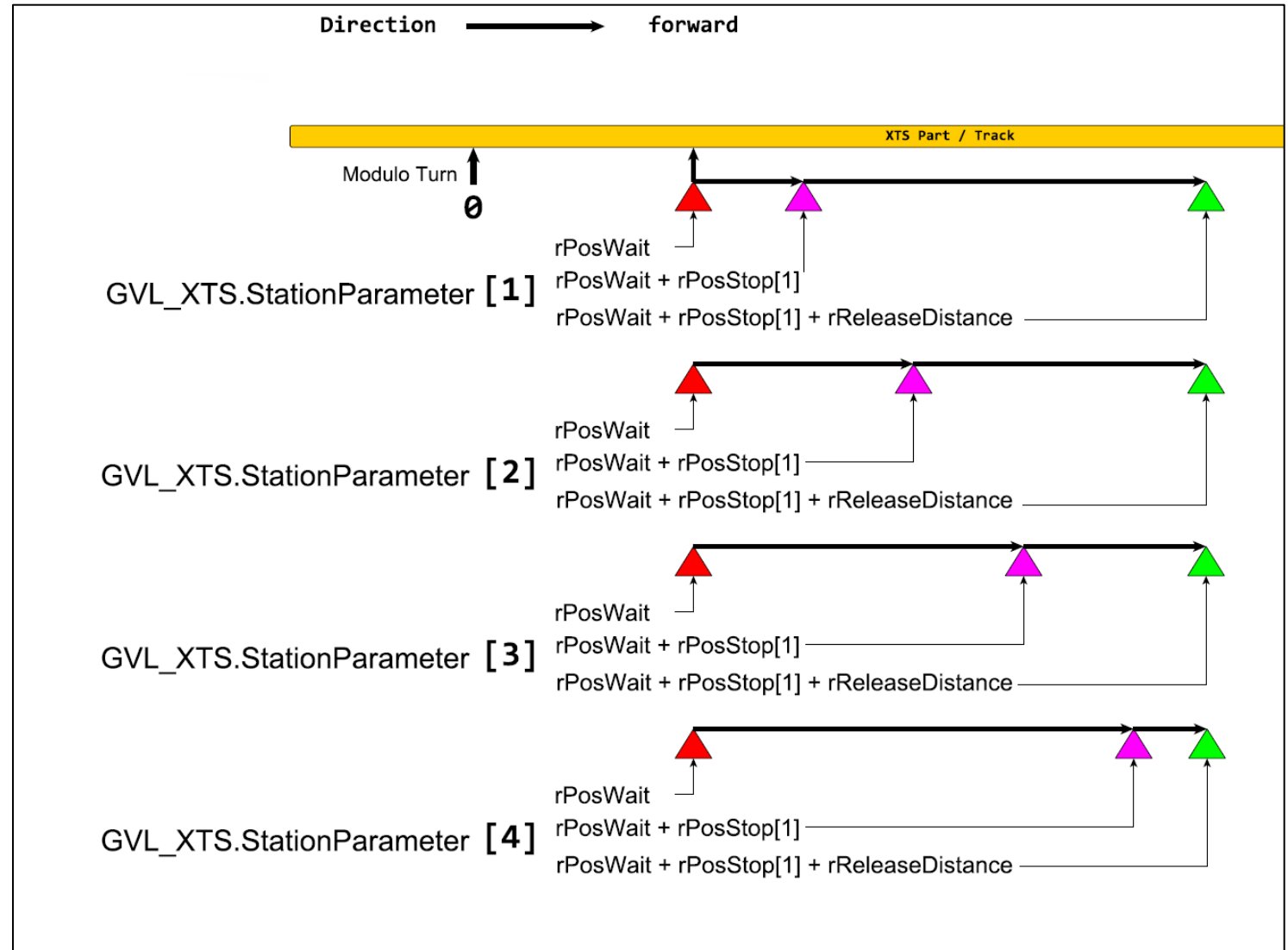
- **StationList**: a list in which a sending stations writes the mover ticket for this station. Station checks its own list cyclically and reports detection of a new mover.
- **StationCtrl**: control struct to command station
- **StationState**: state information you have to react to.



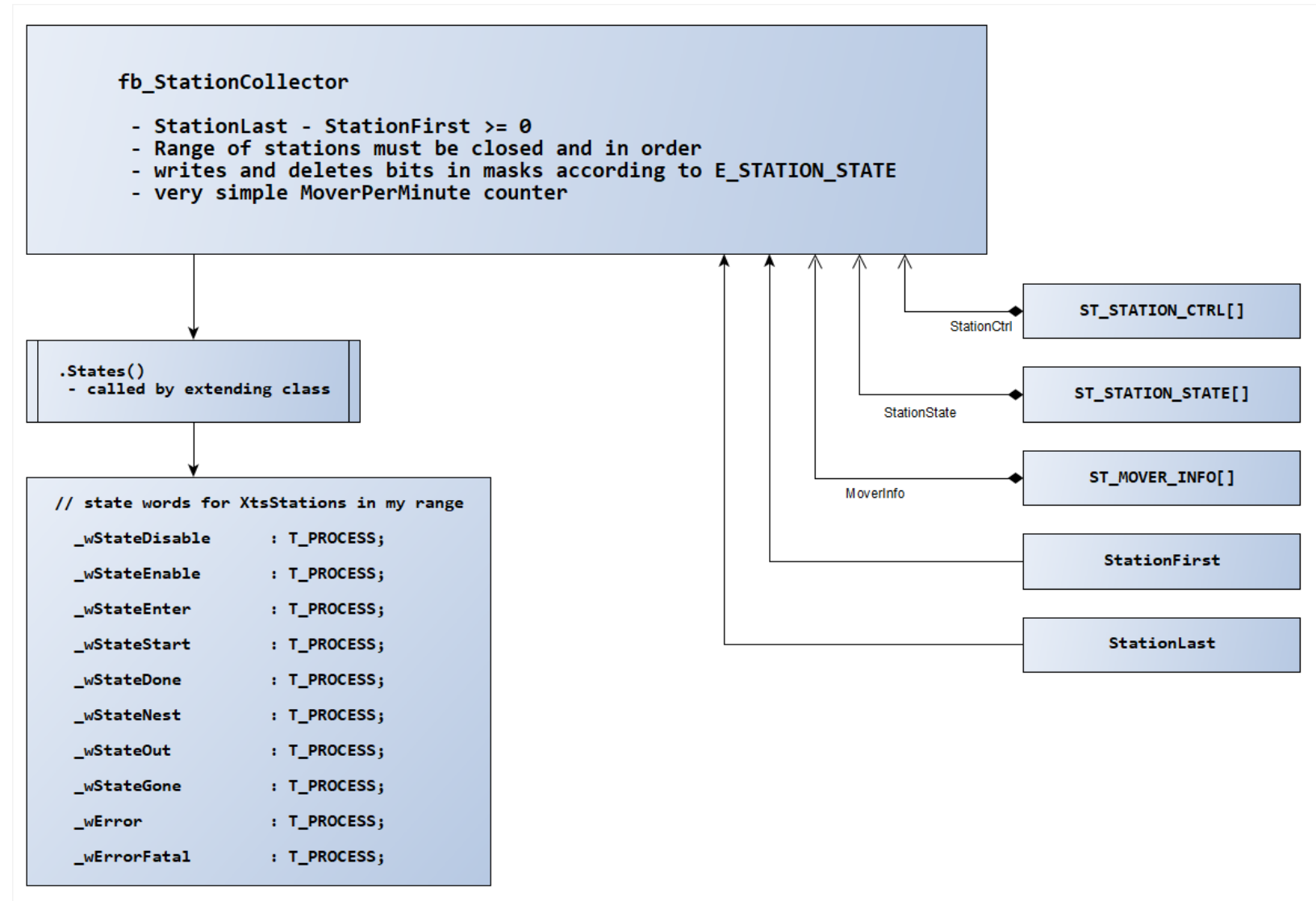
- **Station based approach**
- **PROCESS:**
  - may have one or many stations
  - works stations simultaneously
  - may mute stations
  - Stations in processes may have multiple nests



- **Station based approach**
- **PROCESS:**
  - may have one or many stations
  - works stations simultaneously
  - may mute stations
  - Stations in processes may have multiple nests



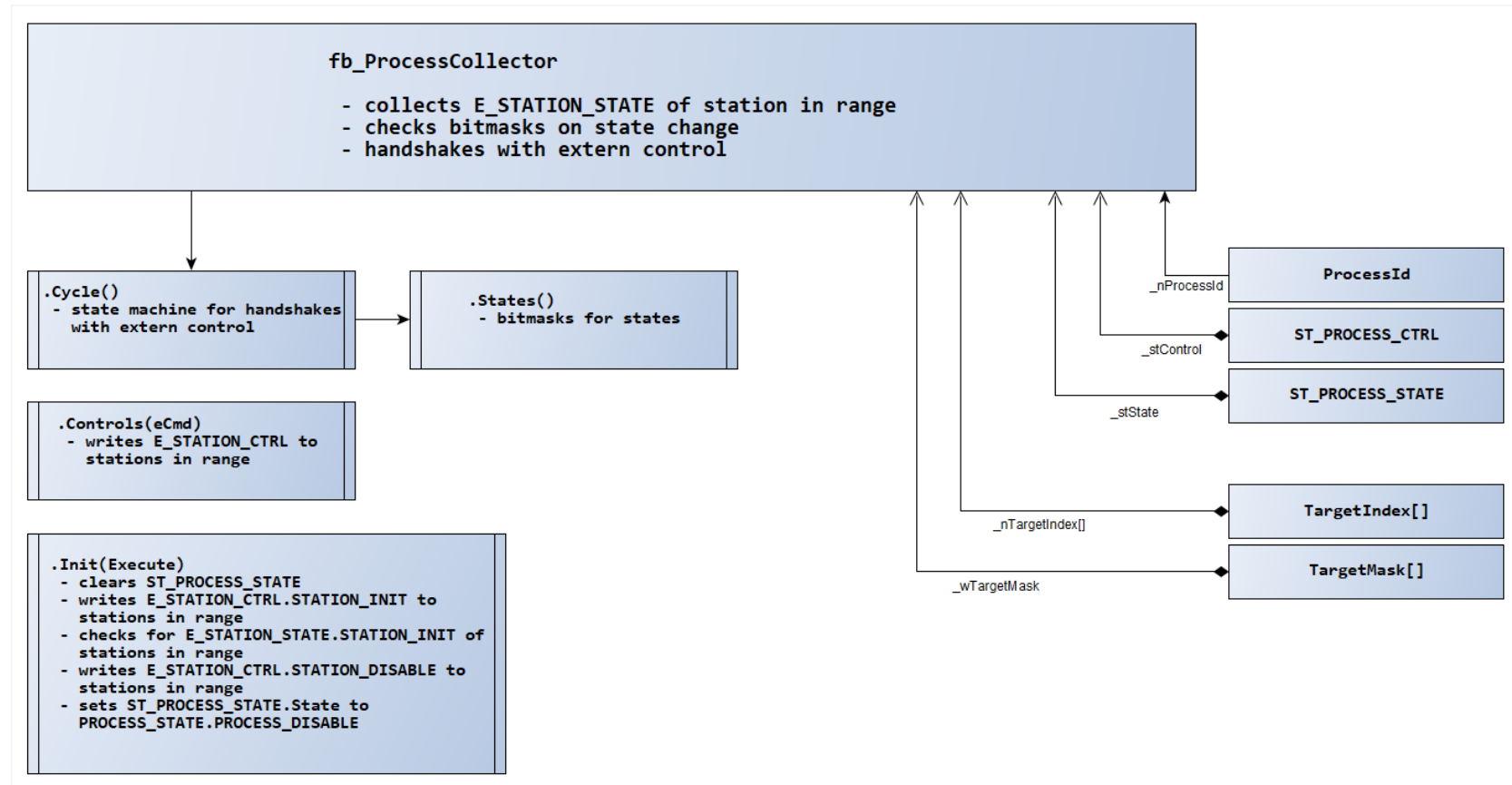
- Station based approach
- **PROCESS:**
  - collects information from the stations.



## ▪ Station based approach

## ▪ PROCESS:

- commands stations via dedicated structures



## XTS\_TRANSPORT\_LAYER project

MIT License

Copyright (c) 2026 HAUD

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

**The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.**

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.