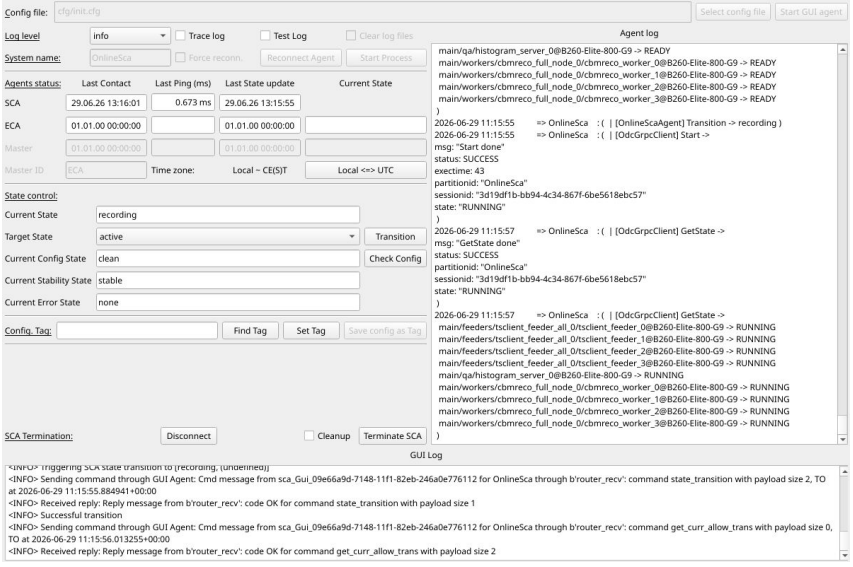
VT26 from DDP perspective

- Focus on management and control sequence
- First *production* use of the new SCA and integration layers
- Components used
  - GSI IT tools - ODC, DDS, FairMQ
  - Online SCA
    - ECS/SCA GUI *(P-A.Loizeau)*
  - FairMQ integration layer
    - **`cbmreco`** and **`timeslice_forwarder`**

Control system

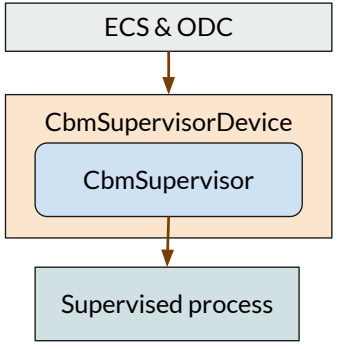
First version of Online SCA

- Connection to ECS
- Communication with ODC via gRPC
- Basic GUI
- Functionally worked very well
- Many things to polish
  - State transitions are blocking in GUI
  - Error handling is rough



FairMQ devices

- **`CbmRecoDevice`** - online binary (**`cbmreco`**) functionality as FairMQ device
  - Utilizes the same steering class as **`cbmreco`**, but replaces **`main()`**
  - **TODO** Currently, **`fles::TimesliceSource::get()`** is blocking when no data is coming, hence *Recording -> Active -> Initialized* transition not always reliable (transition not used in VT26)
- **`CbmTsForwarderDevice`** - **`timeslice_forwarder`** process managed with CbmSupervisor (via **`boost::process::v2`**)
- FairMQ is, at this, point not used for any data movement, only for ODC communication and state machine functionality



Resource utilization

- We had to heavily oversubscribe on resources to achieve high CPU utilization
  - 16x 32-thread **`cbmreco`** processes per compute node (+ 1x **`timeslice_forwarder`**)
  - Each compute node had 128 physical cores and 256 CPU-threads
  - Ideally, we should achieve high CPU load with half the number of processes when saturated with timeslices
- I think cbmreco is still leaking memory over time, investigation ongoing alongside https://redmine.cbm.gsi.de/issues/4156