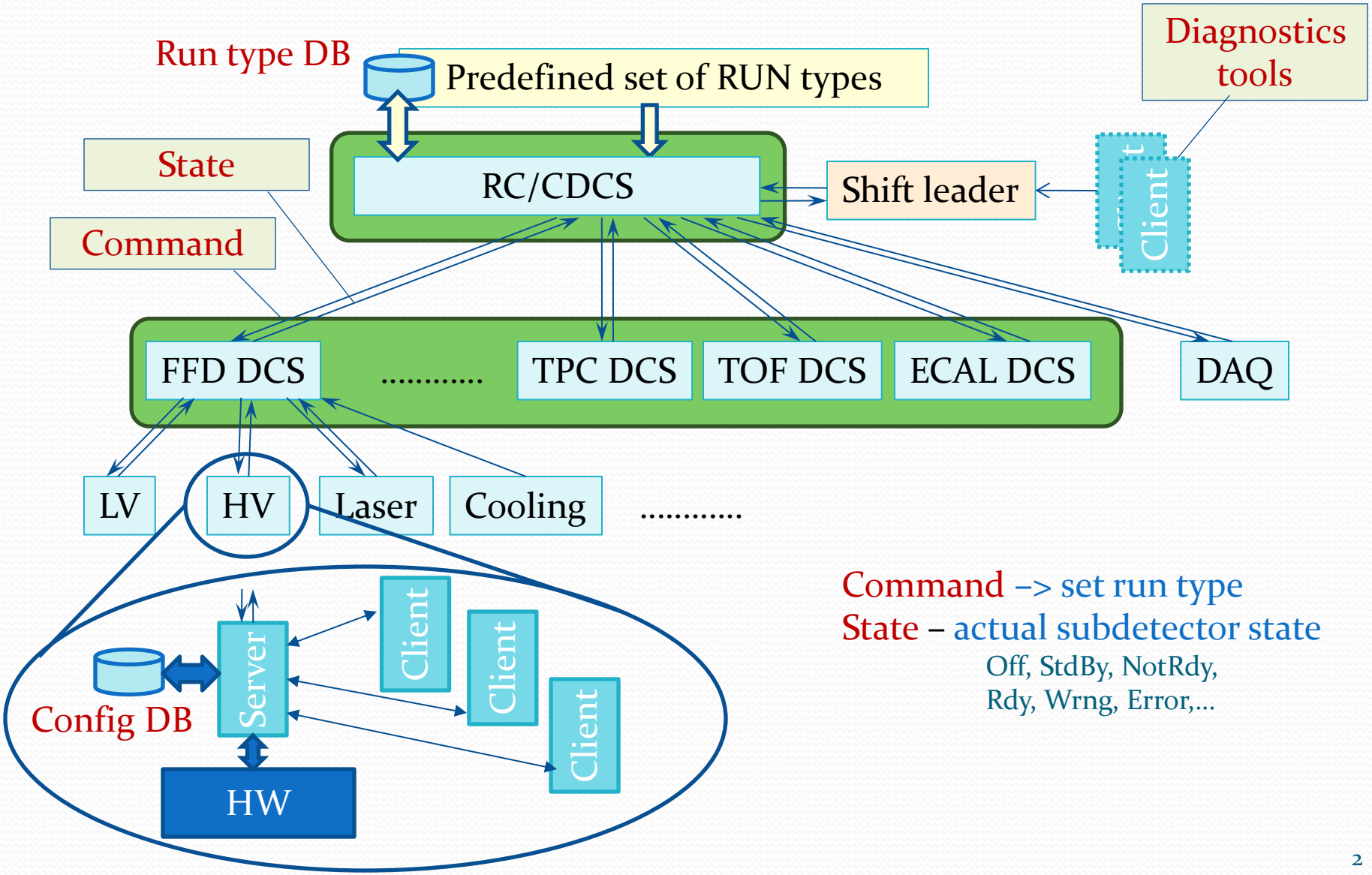
State of MPD DCS and plans for 2025

S.Sergeev

MPD DCS structure



Command –> set run type

State – actual subdetector state

Off, StdBy, NotRdy,
Rdy, Wrng, Error,...

Interfaces actual state

Agreed with subdetector teams and tested

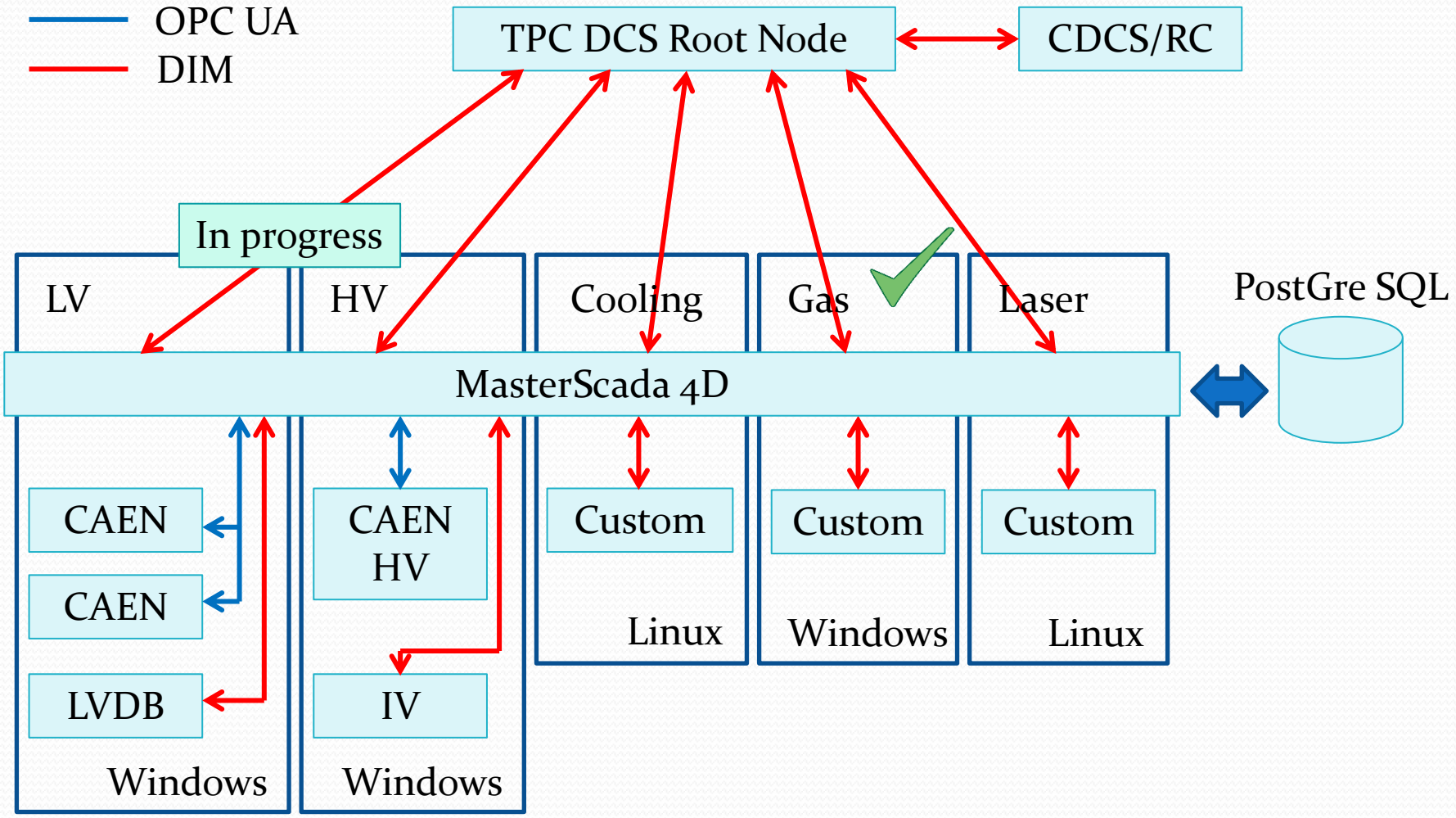
- Solenoid -> MS4D -> DIM -> CDCS
- DAQ -> DIM -> CDCS
- ITS -> WINCC OA -> DIM -> CDCS
- TPC -> MS4D -> DIM -> CDCS
- FFD -> home-made hw/sw -> DIM -> CDCS
- ECAL -> Tango -> DIM -> CDCS
- TOF-> Tango -> DIM -> CDCS
- fHCAL -> DIM -> CDCS
- Lumi -> Tango -> DIM -> CDCS

DCS framework

- **Central DCS application** – prototype runs
- **Subdetector root node** - prototype runs
  - Has GUI
  - Adjustable for subdetector architecture
  - Joins Linux and Windows computers
- **Converter local files <->DIM** – final version runs
- **Hardware Server Emulator** – runs
- **DIMTree** - DIM address space browser (runs at Windows and Linux)

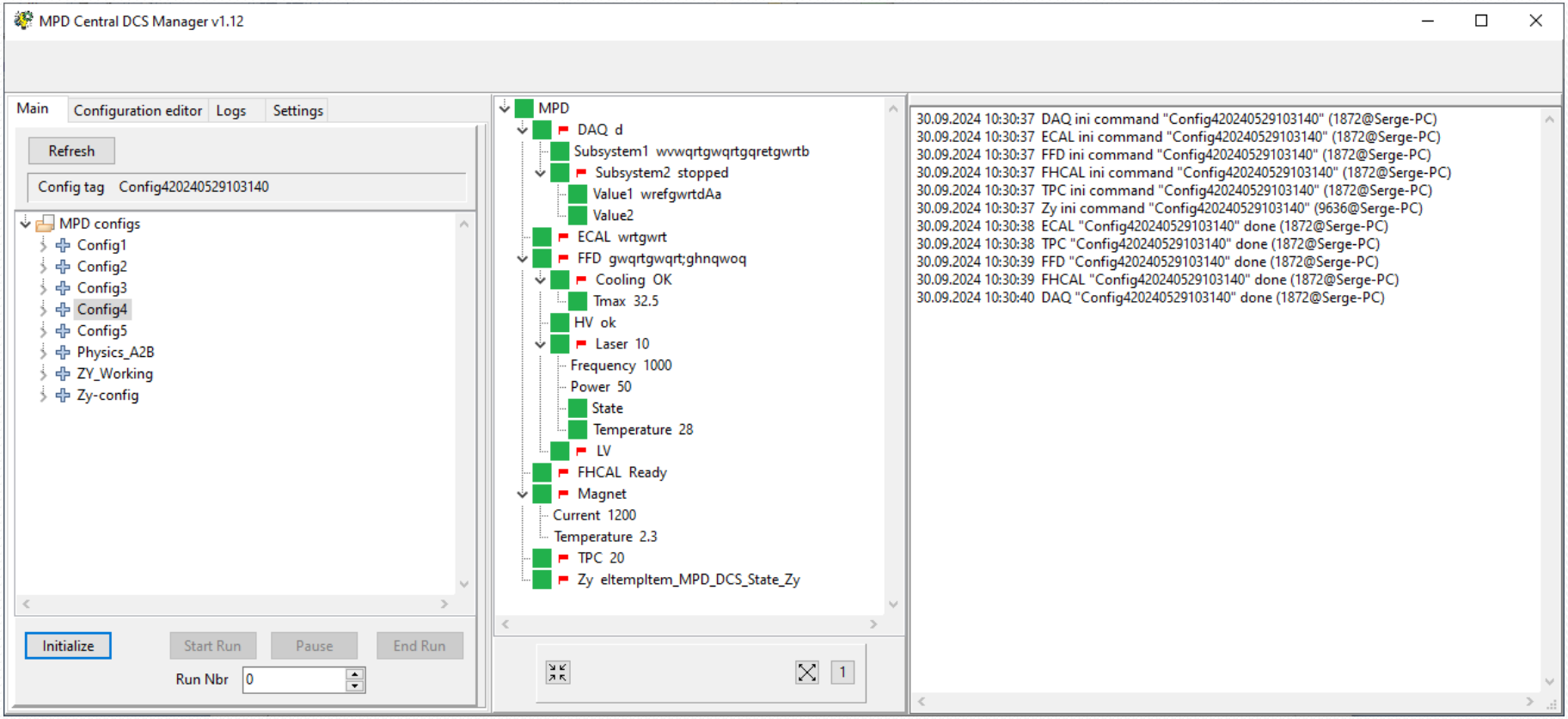
- **Started TPC integration**. During this process the Root node application to be modified

TPC DCS

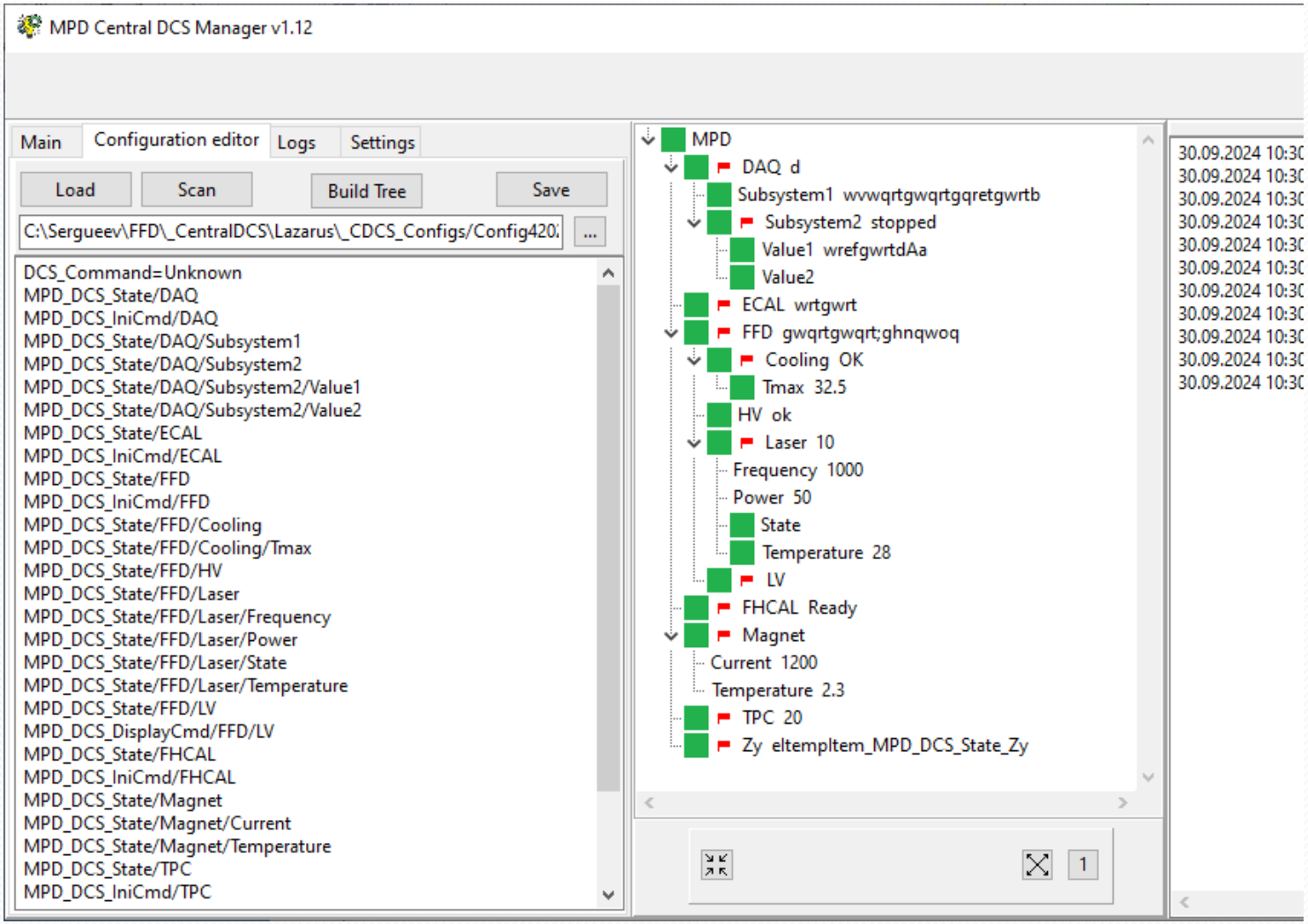


Interface MasterScada_4D <-> DIM ready

CDCS manager I



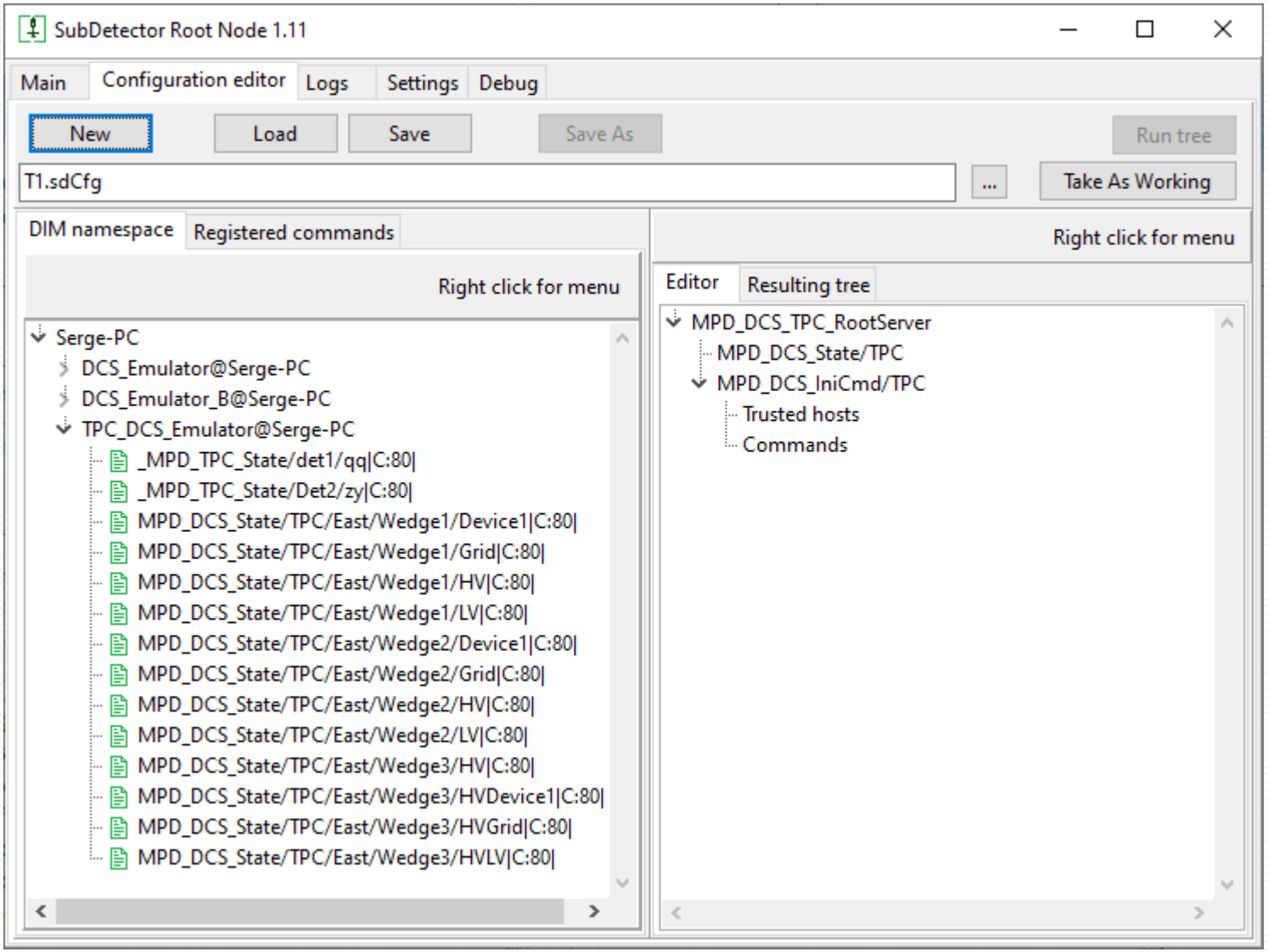
CDCS manager II



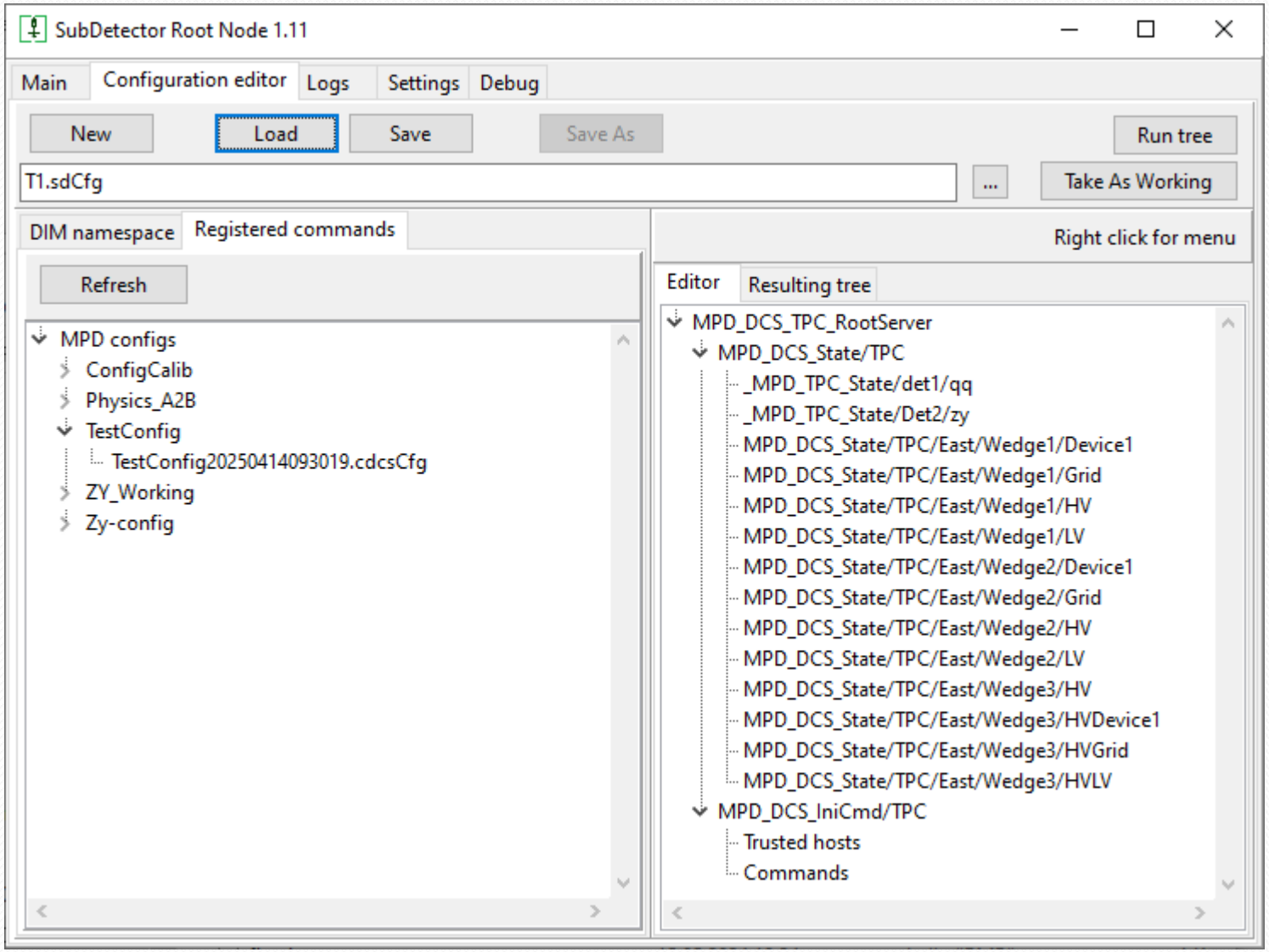
SubDetector Root Node I

- Receives states of subdetector subsystems using DIM Info Services
- Builds resulting subdetector state
- Receives commands from CDCS.
- Parses CDCS commands and converts to subdetector subsystems commands.
- Sends commands to subdetector subsystems using DIM Command Services
- GUI for configuration command editing
- Adjustable for MPD subdetectors

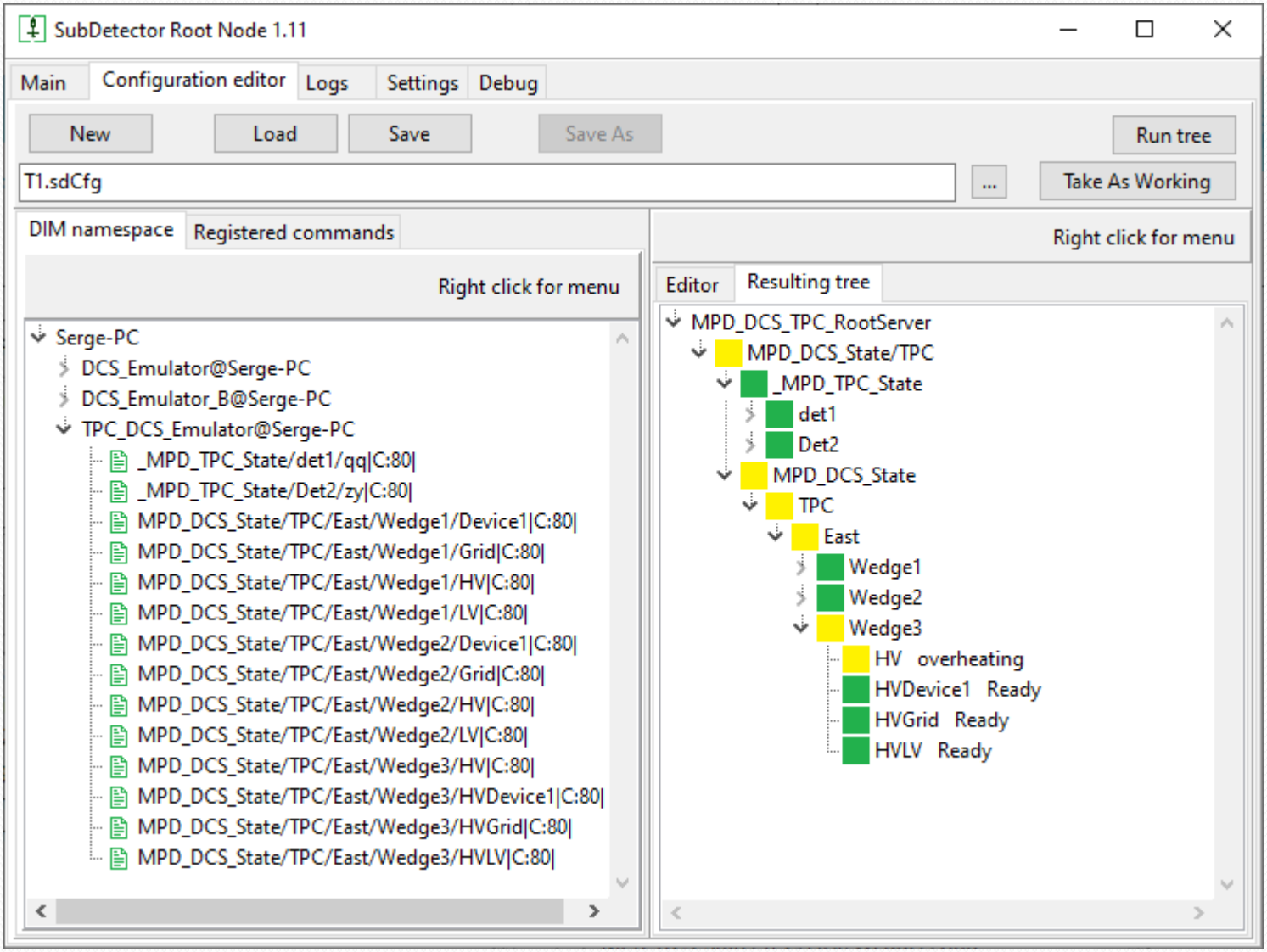
Root Node, DIM item selector



Root Node, Command selector



Root Node, Test run



Disk file to DIM Converter

- Reads local files containing subsystem state with refresh period ~1 sec. Converts content of file to DIM Info item
- Checks “heart beat” and generates OFF state if no data refresh occurred during predefined time
- Receives DIM commands from subdetector Root Node and records command content to a disk file
- One file – one DIM item
- One subdetector could run Converter copies at different computers
- To be ready in mid-November for Windows and by the end for Linux

Safety issues

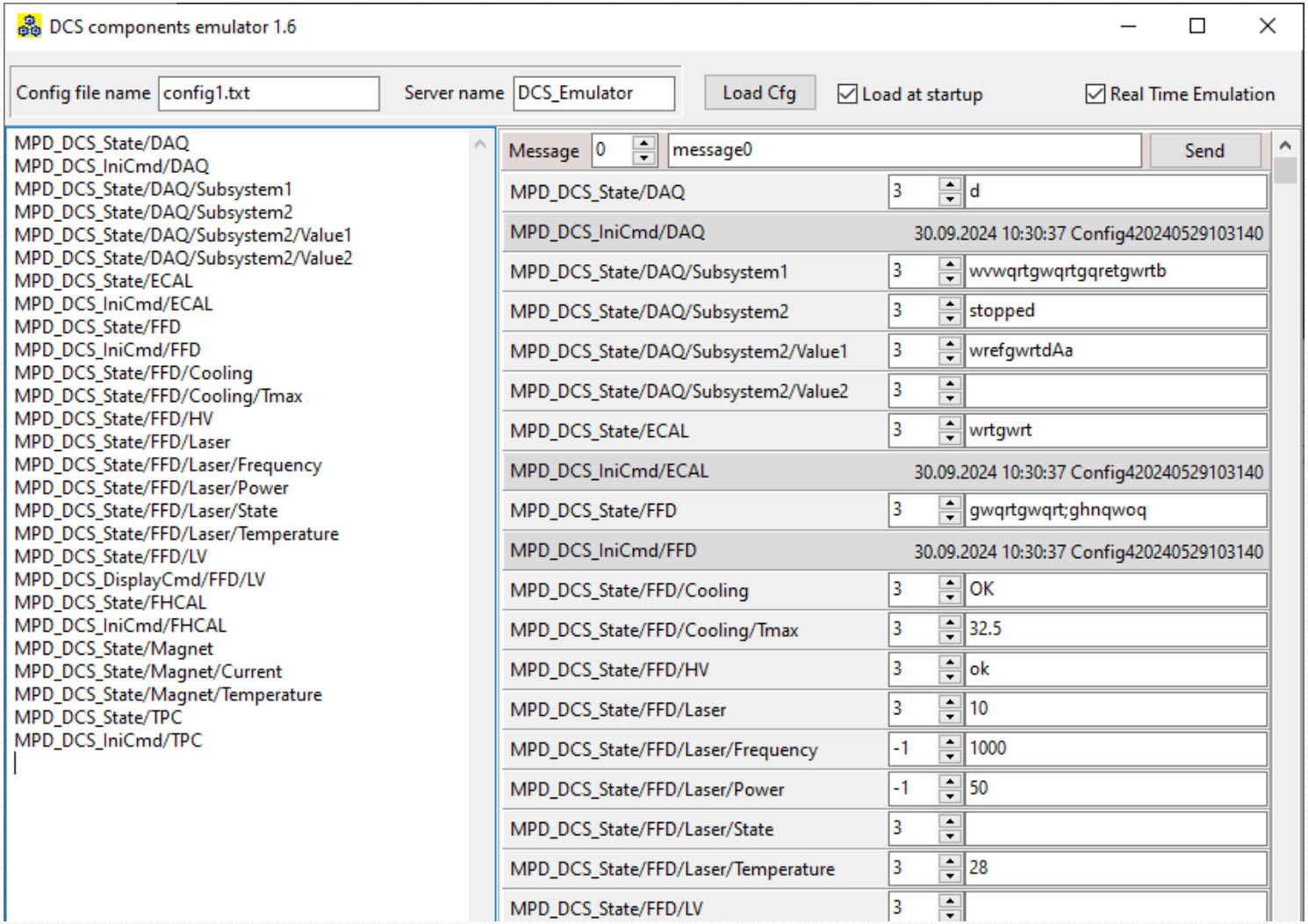
- Each DIM command contains a process ID and a name of a computer issued the command (Command Source ID, CSID)
- Root node has a possibility to create a list of trusted hosts which could send a command
- Each command service should check a CSID of a command
- The CDCS publishes its CSID and root nodes could get it in a real time mode
- The Root Node also publishes its CSID and hardware servers also could get it in a real time mode
- DIM protocol does not allow to have two items with the same name in a common name address space so there is no simple way to send a command by mistake

Plans for 2025

- TPC integration
- Other subdetectors?
- Migration of CDCS manager to Linux
- Modification of the RootNode application for Linux (to have common version for both Windows and Linux)

Thank you

DCS emulator



CDCS interface I

- RC/CDCS subscribes to published by subdetectors **state** InfoItems with names

  MPD_DCS_State/<subdetector name>

- Run configuration contains subdetectors list used in a run
- RC/CDCS sends run type name (text) to all CommandItems of subdetectors being in a list. CommandItems should have a name like

  MPD_DCS_IniCmd /<Subdetector name>

CDCS interface II

- Each subdetector DCS root node could (should?) have a CommandItem with name

  MPD_DCS_DisplayCmd/<subdetector node name>

  A command received by this CommandItem should start diagnostic tool (see below)
- DAQ should have additional Info/Command items to provide vital information to/from the RC/CDCS (to be discussed with DAQ team)

CDCS interface III

- RC/CDCS has a CommandItem MPD_DCS_Messages to receive messages from subsystems/subdetectors
- Format of message should be like (to be discussed)
  - <subdetector name>_<severity level>_<message text>
    - <severity level> defines a way to process the message
      - 0 -> just to show in a window. Could be scrolled by messages arriving later
      - 1 -> stays at the screen until confirmed
      - 2 -> stays at the screen until confirmed + sound alarm if not confirmed during defined time (1 min as an example)
      - 3 -> stays at the screen until confirmed + instant sound alarm
- All messages have a text content

States and colors


State=-1, Item does not have a state, no color to be displayed


State=0, OFF - any of sub-elements does not respond


State=1, StdBy – any of sub-elements is in stand-by mode

State=2, NotRdy – any of element is in transition state (Time-out should be implemented)


State=3, Ready – all elements are OK


State=4, Wrng – any of elements is in Warning state


State=5, Error – any of elements is in Error state


State=6, Ignrd – node in Partitioned state

Extended display (to be discussed)

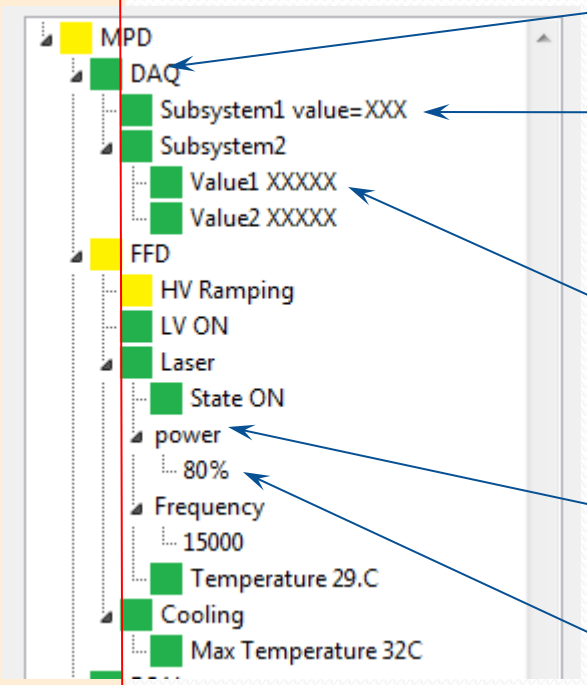
- A subdetector should provide a set of diagnostics tools stated by a CommandItem

  MPD_DCS_DisplayCmd/<subdetector name>

- This should be an application running at a CDCS PC or a web-page running AJAX script (?). The web server could be provided by a CDCS. Page content should be developed by the subdetector team and could be located at a common disk space
- Start parameters are defined in the CommandItem command content

Extra parameters interface

MPD state is built using 1-st level nodes



"MPD_DCS_State/DAQ", content="3"

"MPD_DCS_State/DAQ/Subsystem1", content="3_value=XXX"

"MPD_DCS_State/DAQ/Subsystem1/Value1", content="3_XXXXX"

"MPD_DCS_State/FFD/Laser_power", content="-1"

"MPD_DCS_State/FFD/Laser_power/", content="-1_80%"

Obligatory

Extra info to display in the tree, defined by subsystem/subdetector