

Slow Control Status

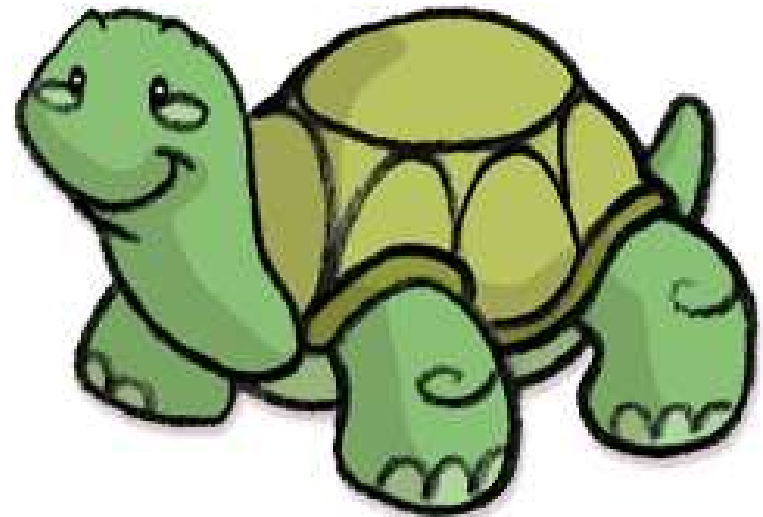
Mikihiko Nakao (KEK)

`mikihiko.nakao@kek.jp`

22 January 2015

7th Belle II VXD workshop

Praha



Purpose of this talk

- Status of VXD slow control status was already nicely summarized by M.Ritzert
- Outline
 - NSM2/EPICS/CSS interface
 - NSM2 based control systems

NSM2/EPICS/CSS packages

- **NSM** (version 1.9.41)

- tiny package (<1MB source, 4MB binaries, built in 2 seconds)

- bdaq SVN, see <https://belle2.cc.kek.jp/~twiki/bin/view/Detector/DAQ/NSM2>

- **EPICS** (Release 3.14.12.4)

- small package (12MB source, 132MB binaries, built in 81 seconds)

- official site: <http://www.aps.anl.gov/epics/>

- **CSS on eclipse** (cs-studio branch 3.3.x)

- huge package (1.25GB git clone)

- official site: <https://github.com/ControlSystemStudio/cs-studio.git>

- on top of eclipse 3.7.2 + delta-pack 3.7.2 (binaries)

- plugins / patches by M.Ritzert / T.Roeder

- **CSS standalone application**

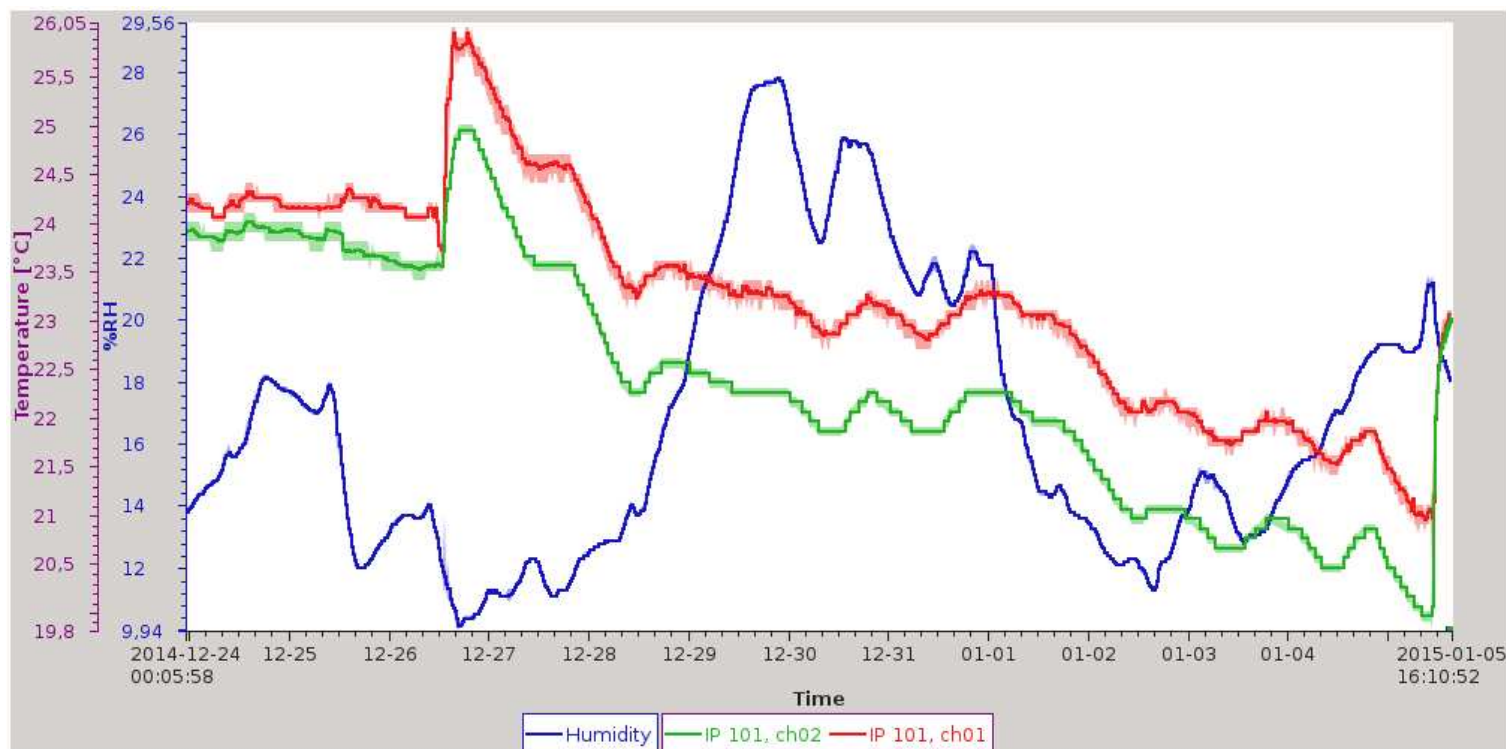
- script provided by M.Ritzert, even bigger? (3GB in total)

Running NSM2/EPICS/CSS at KEK

- Setting up NSM2 and EPICS is easy (single “make” command), setting up CSS is more challenge
- Installed on two PCs in KEK during visit of M.Ritzert/T.Roeder for Scientific Linux 6.5 (b2epics, b2stone)
- I went through the installation procedure again on my Ubuntu 14.04 laptop, and now the procedure is written down at <https://belle2.cc.kek.jp/~twiki/pub/Detector/SlowControl/CSS/HOWTO.css-in-ubuntu14.04.txt> to be able to work even during a flight
- Still on learning curve at KEK

E-hut monitor with EPICS

- Temperature / humidity monitor based on Yokogawa MW100
- Channel assignment is still temporary, but hardwired interlock and alarm are already in operation
- EPICS IOC using modbus
- Set up in October, running since then without break (trend during new year power shutdown is recorded in CSS Archiver)



(M.Ritzert)

NSM2 plugin to CSS

- nsm2socket: process to convert native NSM2 data into socket based access, needed for any java or CSS application

- CSS plugin (org.csstudio.belle2.pvmanager.nsm) handles PV

```
nsm://STORAGE_STATUS:storage_status:node[1]:flowrate_in
```

in a same way that it access EPICS PV, for example

```
epics://MW100_1_100:ch01
```

- Homeworks

- nsm2socket program to be included in NSM2 package, i.e., to make it independent of slow control software tree in Belle II software SVN
- there is a delay in updating NSM2 data, which can be removed
- plugin for NSM2 request from/to CSS is yet to be established, or otherwise a CSS GUI cannot control NSM2 program

NSM2-EPICS communication

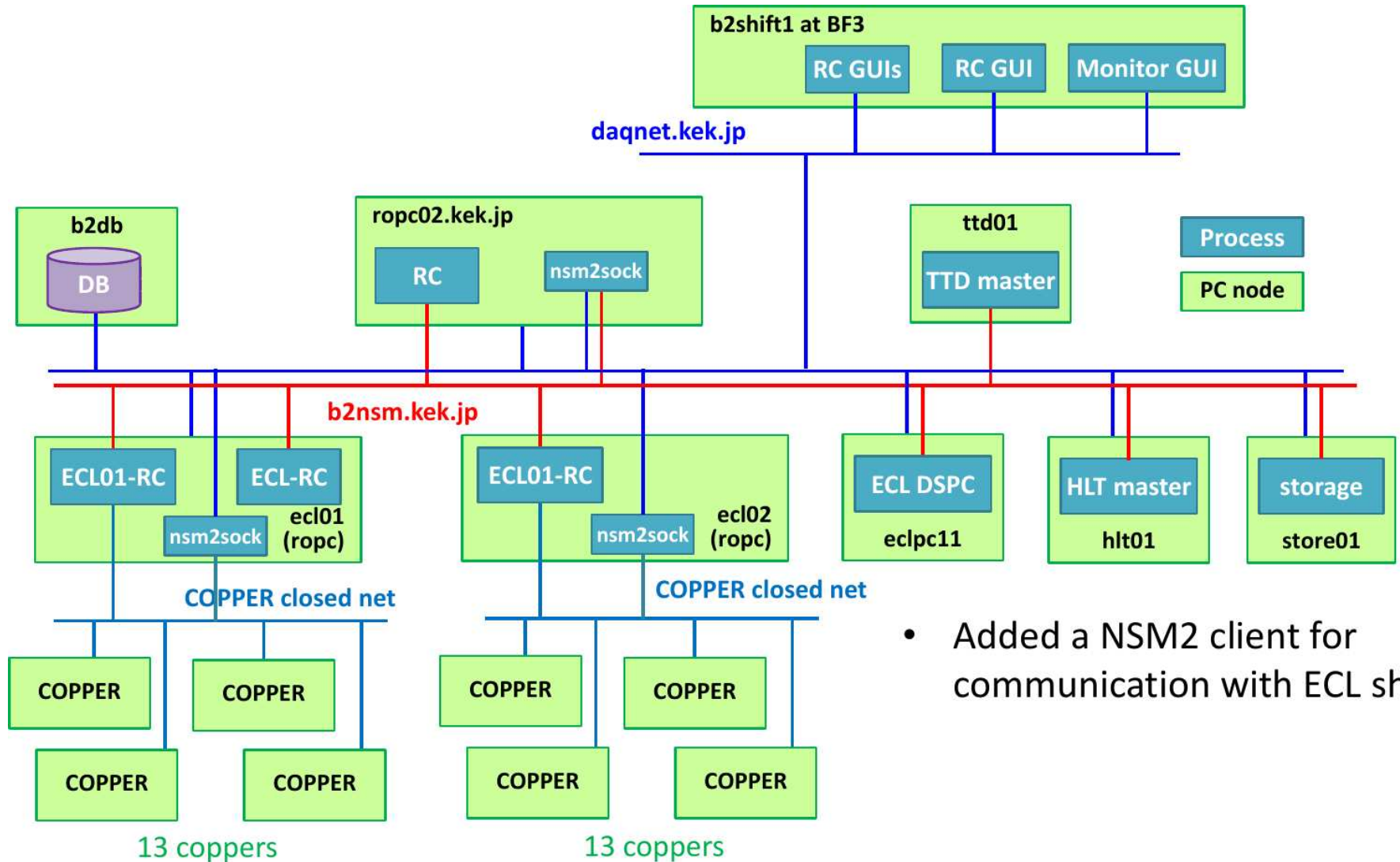
- Process to process communication does not need to go through CSS, but then NSM2 process and EPICS process (IOC) have to talk each other
- devNSM IOC: NSM2 data and request accessible by EPICS
 - Still under working directory with example NSM data
 - PV records are defined for test purpose
 - Homework: Framework and PV definitions have to be separated, and repository to be decided
- Homework: EPICS PV to be accessed by NSM2

NSM2-based Run/Slow-control

- Full data acquisition chain for unified Belle2link systems
HSLB ⇒ **COPPER** ⇒ **ROPC** ⇒ **EB1** ⇒ **HLT** (⇒ EB2) ⇒ **storage**
- Nested run control process,
e.g., ECL has **2** ROPCs and each ROPC has **13** COPPERs to control
- Established at DESY beam test, now heavily used in ECL cosmic test
- Timing distribution system (**TTD**)
- Frontend Electronics, through Belle2link or other path
- Power supplies (EPID, KLM, CDC, ...)

All pieces are getting ready if we reside in the NSM2 world, although some nice features are missing such as CSS Archiver...

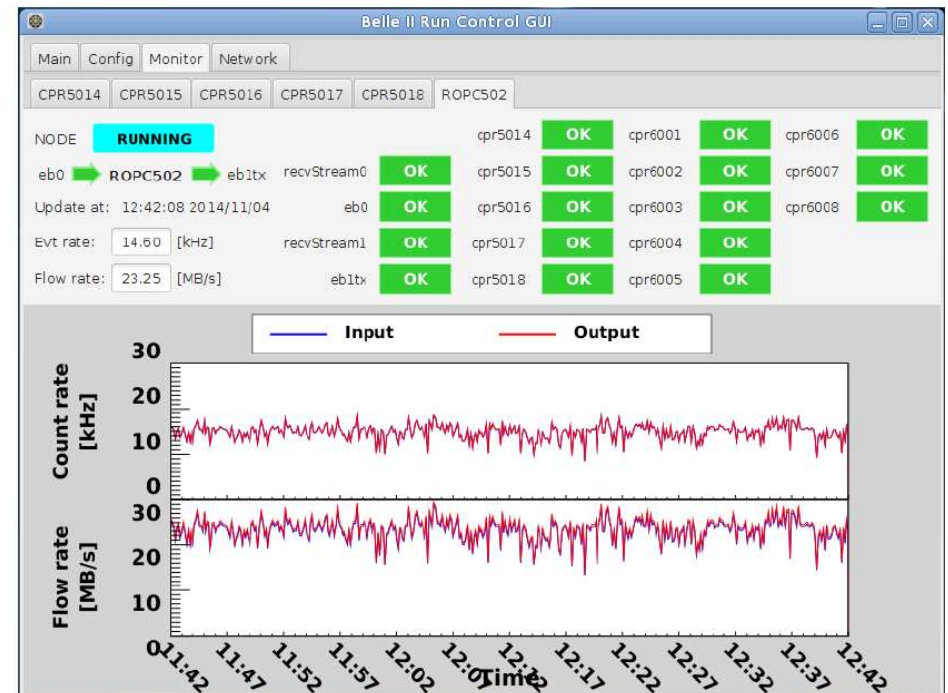
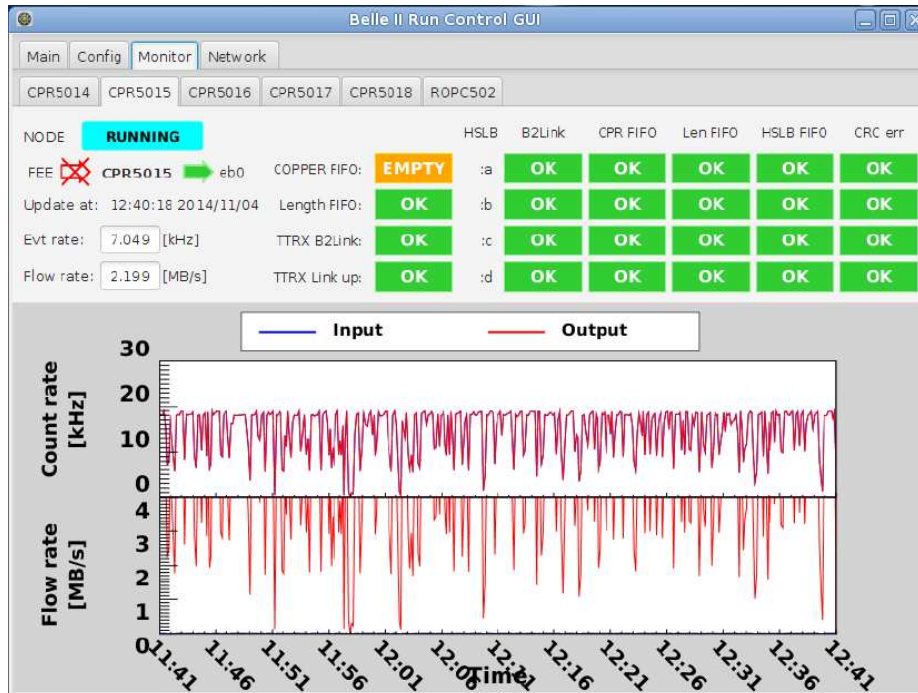
Run control for ECL cosmic ray test



- Added a NSM2 client for communication with ECL shapers

● All Barrel ECL is read out, with 18 COPPER modules

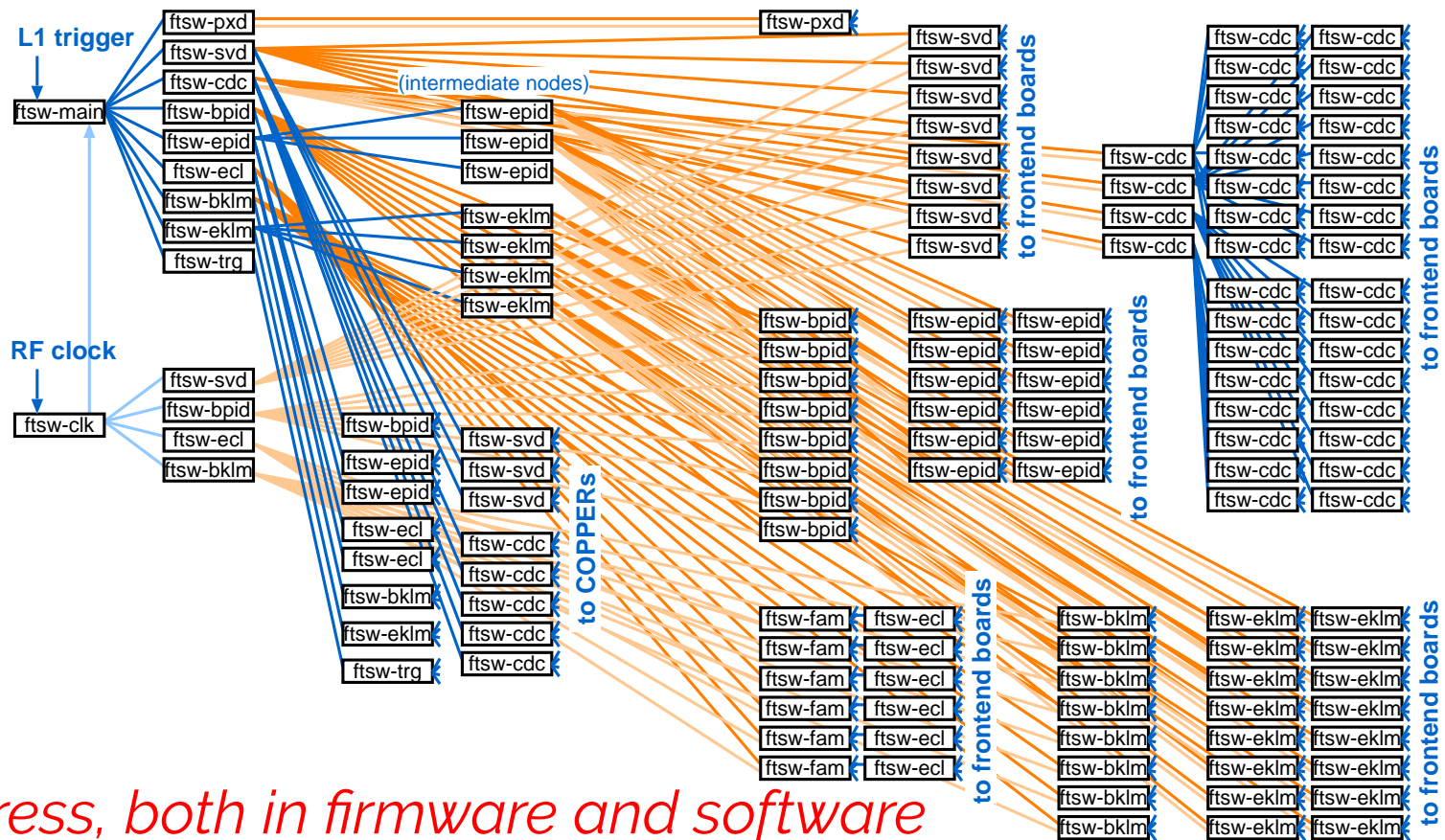
Screen shot



- This GUI is based on Java-8
- Plan is to switch to CSS, after developing the CSS plugin for NSM2 request

TTD control

- ~600 FEEs, ~200 COPPERs, ~140 FTSW has to be controlled
- Partitioned operation (e.g., ECL and KLM running separately)
- Error location has to be identified
- FEE FPGA programming through JTAG over b2tt



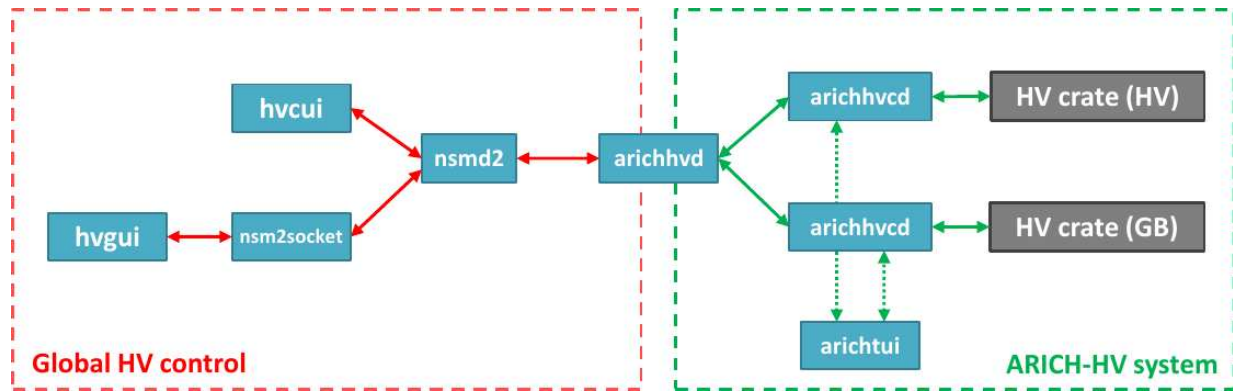
Work in progress, both in firmware and software

Configuration

- ECL configuration: shaper/ADC via Ethernet
 - Which shapers to be used
 - Readout mode / threshold setting
- KLM configuration
 - Belle2link to download **millions** of parameters
 - Belle2link stream mode for most of them as it is too many to be mapped on register address space
 - Database and handling scheme is under discussion now

PS control

- PS hardware controlled via Ethernet
- Setting voltages and V/I limits
- Monitor V/I
- Implemented for EPID
- Test started for KLM and CDC
- curses based CUI, too



tkonno@b2slow2:~/slc

ALL	[ON]:[OFF]	Store [0]	Recall [0]								
Crate	Slot	Ch	Switch	RampUp[V]	RampDown[V]	Vdemand[V]	Vlimit[V]	Climit[mA]	State	Vmon[V]	Cmon[mA]
01	01	00	ON	200	100	2000	9000	1000	ON	1998.0	399.0
01	02	00	ON	200	100	2100	9000	1000	ON	2101.0	420.0
01	03	00	ON	200	100	1890	9000	1000	ON	1890.0	378.0
01	04	00	ON	200	100	1900	9000	1000	ON	1899.0	379.0
01	05	00	ON	200	100	2010	9000	1000	ON	2007.0	401.0
01	06	00	ON	200	100	2030	9000	1000	ON	2032.0	406.0
01	07	00	ON	200	100	2010	9000	1000	ON	2010.0	402.0
01	08	00	ON	200	100	2040	9000	1000	ON	2038.0	407.0
01	09	00	ON	200	100	2050	9000	1000	ON	2051.0	410.0
01	10	00	ON	200	100	2120	9000	1000	ON	2118.0	423.0
01	11	00	ON	200	100	2010	9000	1000	ON	2009.0	401.0
01	12	00	ON	200	100	2000	9000	1000	ON	1996.0	399.0
01	13	00	ON	200	100	2000	9000	1000	ON	1999.0	399.0
01	14	00	ON	200	100	2000	9000	1000	ON	1996.0	399.0
01	15	00	ON	200	100	2000	9000	1000	ON	1999.0	399.0
01	16	00	ON	200	100	2000	9000	1000	ON	1999.0	399.0
01	17	00	ON	200	100	2000	9000	1000	ON	1998.0	399.0
01	18	00	ON	200	100	2000	9000	1000	ON	2002.0	400.0
02	01	01	ON	200	100	2000	9000	1000	ON	1999.0	399.0
02	01	02	ON	200	100	2000	9000	1000	ON	1999.0	399.0
02	01	03	ON	200	100	2000	9000	1000	ON	1999.0	399.0
02	01	04	ON	200	100	2000	9000	1000	ON	1999.0	399.0
02	01	05	ON	200	100	2000	9000	1000	ON	1999.0	399.0
02	02	01	ON	200	100	1900	9000	1000	ON	1902.0	380.0
02	02	02	ON	200	100	1900	9000	1000	ON	1902.0	380.0
02	02	03	ON	200	100	1900	9000	1000	ON	1902.0	380.0
02	02	04	ON	200	100	1900	9000	1000	ON	1902.0	380.0
02	02	05	ON	200	100	1900	9000	1000	ON	1902.0	380.0
02	03	01	ON	200	100	1800	9000	1000	ON	1797.0	359.0
02	03	02	ON	200	100	1800	9000	1000	ON	1797.0	359.0
02	03	03	ON	200	100	1800	9000	1000	ON	1797.0	359.0
02	03	04	ON	200	100	1800	9000	1000	ON	1797.0	359.0
02	03	05	ON	200	100	1800	9000	1000	ON	1797.0	359.0
02	04	01	ON	200	100	1870	9000	1000	ON	1858.0	373.0
02	04	02	ON	200	100	1870	9000	1000	ON	1858.0	373.0
02	04	03	ON	200	100	1870	9000	1000	ON	1858.0	373.0
02	04	04	ON	200	100	1870	9000	1000	ON	1858.0	373.0
02	04	05	ON	200	100	1870	9000	1000	ON	1858.0	373.0

Summary

- NSM2/EPICS/CSS are now used in KEK on regular basis
- NSM2 based run control chain is now established
- We are now making a new step for the CSS unification, but only after Feb. B2GM
 - A few more development steps is needed to use CSS in the NSM2 based control system
 - Not much before Feb B2GM (too soon), next target is March DAQ WS

More general slides...

Glossary

● **Slow control**

- (in a wide sense) everything listed below
- (in a narrow sense) everything other than run control
- (in a narrowest sense) everything other than run/PS control
which do not change states during regular run period

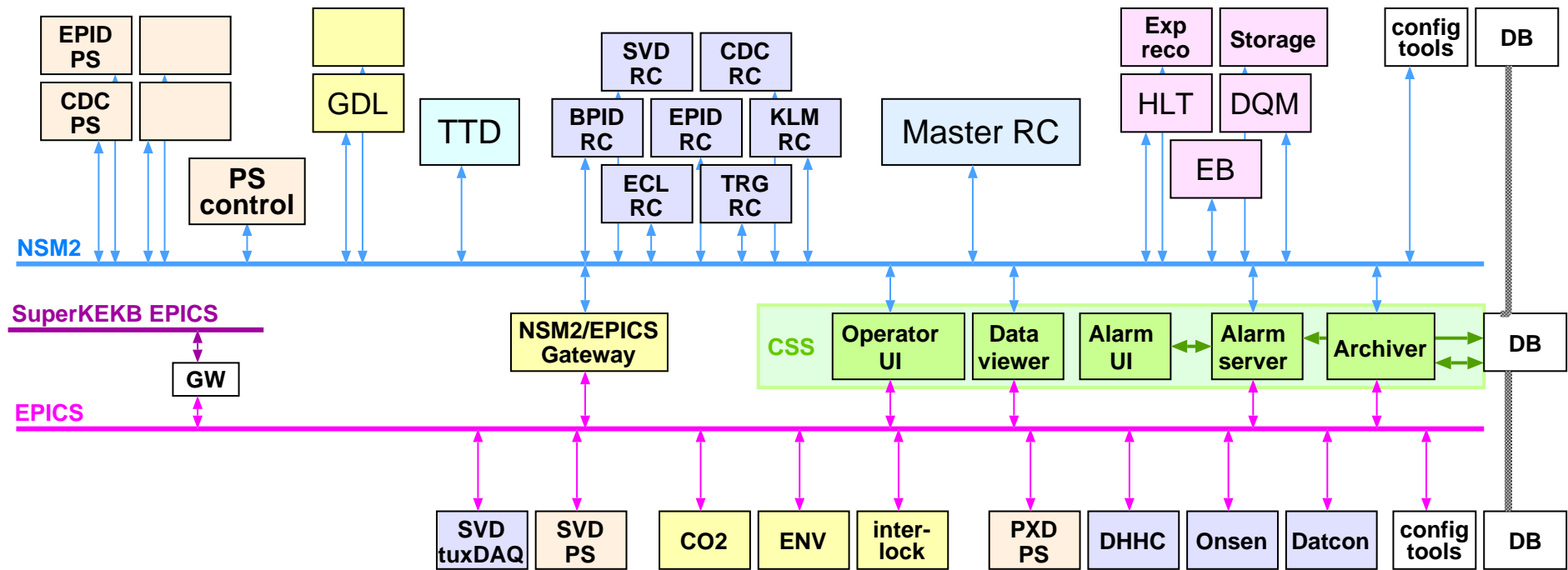
● **Run control**

- (in a wide sense) — to control systems that are involved in data-taking
- (in a narrow sense) — to control systems that are involved in data flow and changes the action when data taking starts

● **Power Supply (PS) control**

- power supplies that changes the states upon beam condition

CSS unifies the control systems



WANTED

- plugin to read NSM2 data by CSS
- plugin to send NSM2 request from CSS
- IOC to provide NSM2 data/requests to EPICS
- IOC to provide EPICS data/requests to NSM2

Unification of operation (state transition)

Run States



No response from the subsystem



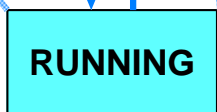
Cannot start run due to various reasons, but accepts messages from Run Control



Can change to RUNNING immediately



Should PXD have a separate state for dummy event running?



Can accept triggers and read out the detector or generate dummy events



(only for TTD and RC) trying to bring some of the subsystem to RUNNING

Power Supply States



No response from the subsystem



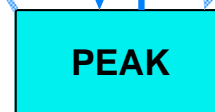
PS is turned off, which happens only during shutdown



PS voltage tolerable for beam injection, while as high as possible for a quick transition to PEAK



(optional) intermediate voltage to allow readout for background check with limited efficiency



Nominal voltage for data taking



PS trip state from which one has to go back to PEAK as quickly as possible

error

problem

Run Control Tree

- **Single master** — centralization of all actions taken during the run
- Run control and Power Supply (PS) control are the main players, others are more passive components
- Function-based partitioning — Data taking systems and Power Supply systems are independent

