

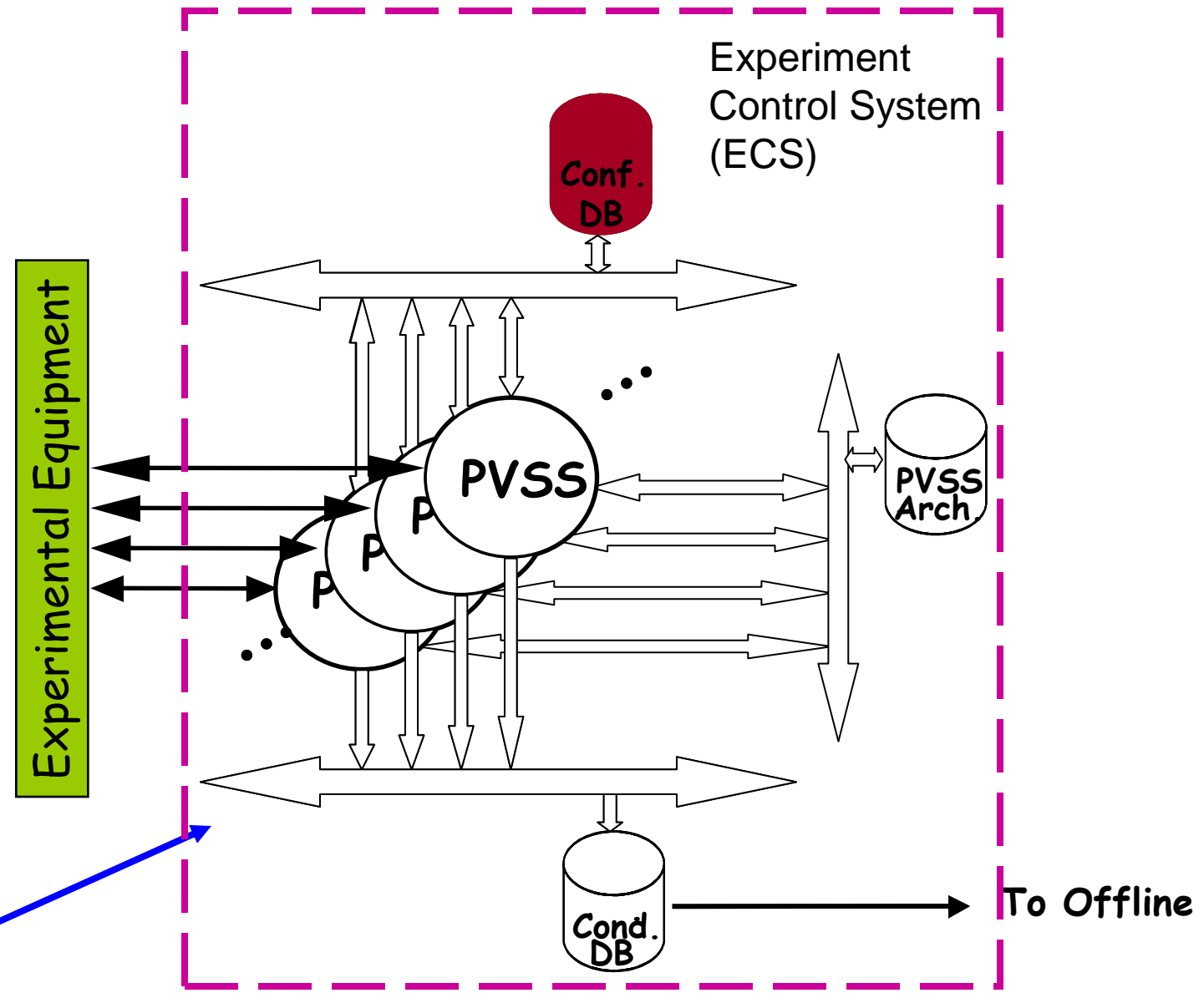


LHCb Configuration Database

Lana Abadie, PhD student
(CERN & University of Pierre et Marie Curie (Paris VI), LIP6)

Context

3 Oracle
Databases



Problem to solve

The ECS will use the configuration DB to get all information necessary to configure and monitor the detector according to:

- **A partition** : part of the detector which can run independently and concurrently

Which subsystems? How are they interconnected?

- **A running mode or an activity** : what settings for the devices in the given partition?

Need to find what and how to store them

Configuration DB: what to store?

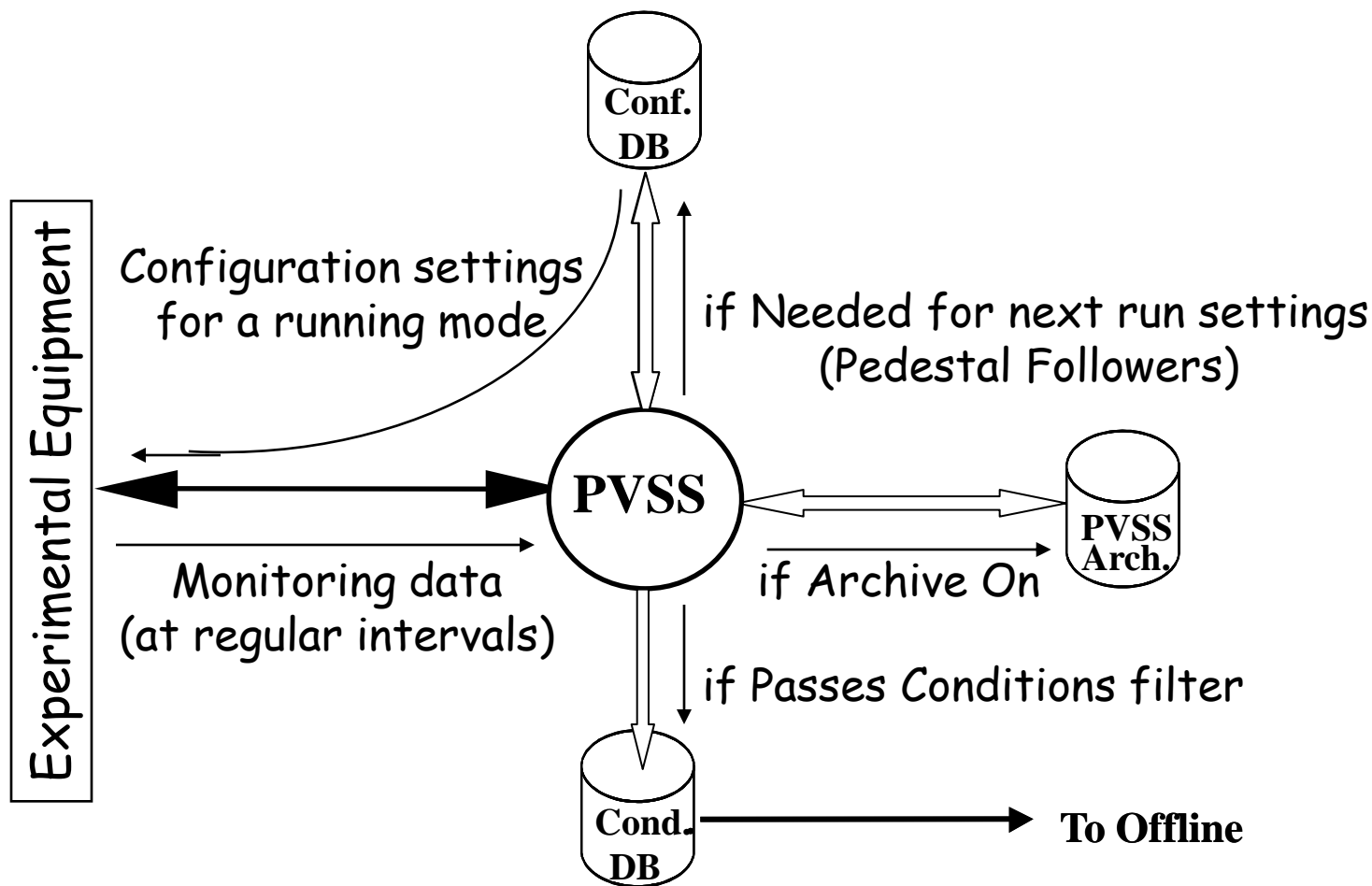
Settings of controllable devices according to a running mode such as:

- All Electronics Boards (Register settings, etc.)
- HV, LV Power Supplies (Voltage settings, etc.)
- Trigger Algorithms (Job options, etc.)

Information about partitioning

- All connectivity between devices per subsystem
- Destination and routing tables
- Configuration files

Dataflow Example



The Conf DB : inside

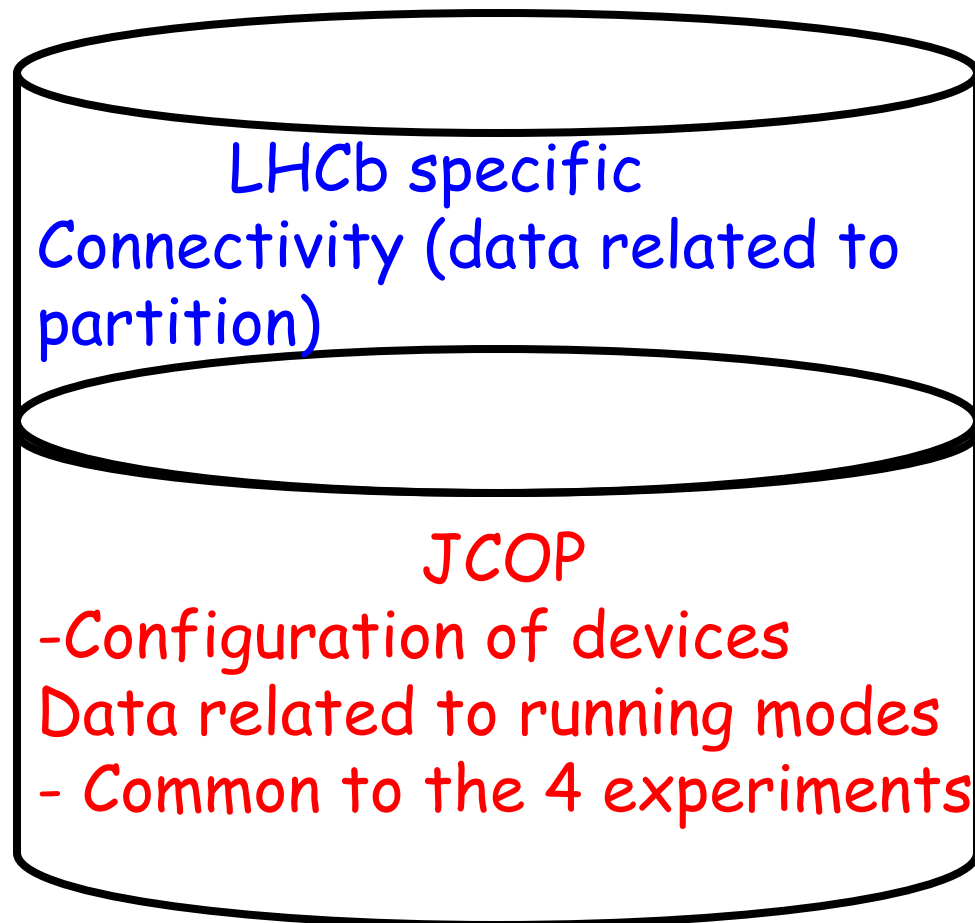
LHCb specific :

Devices in the partition?

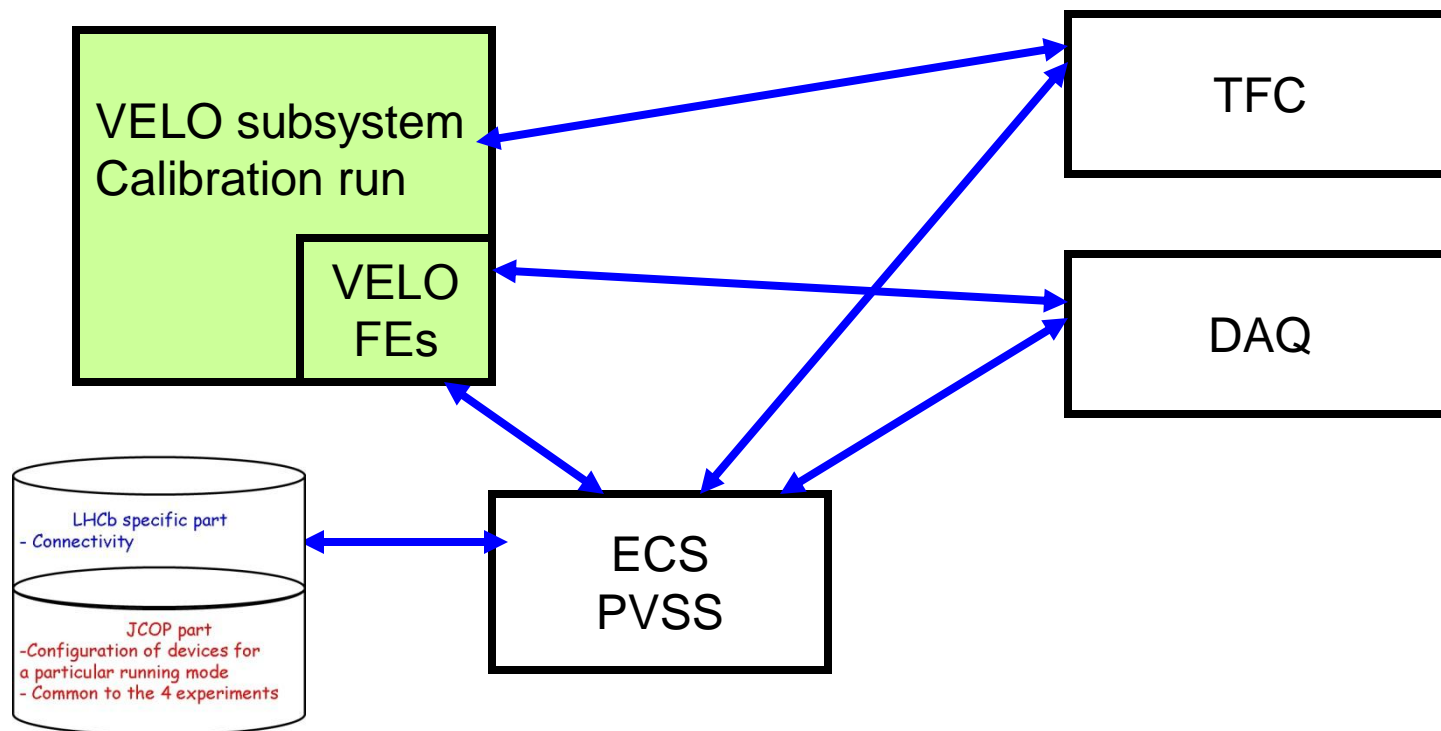
How to interconnect the subsystems ?

JCOP :

Device settings for the selected **activity** such as physics, cosmics ?



Ex: calibration run for VELO LHCb tool

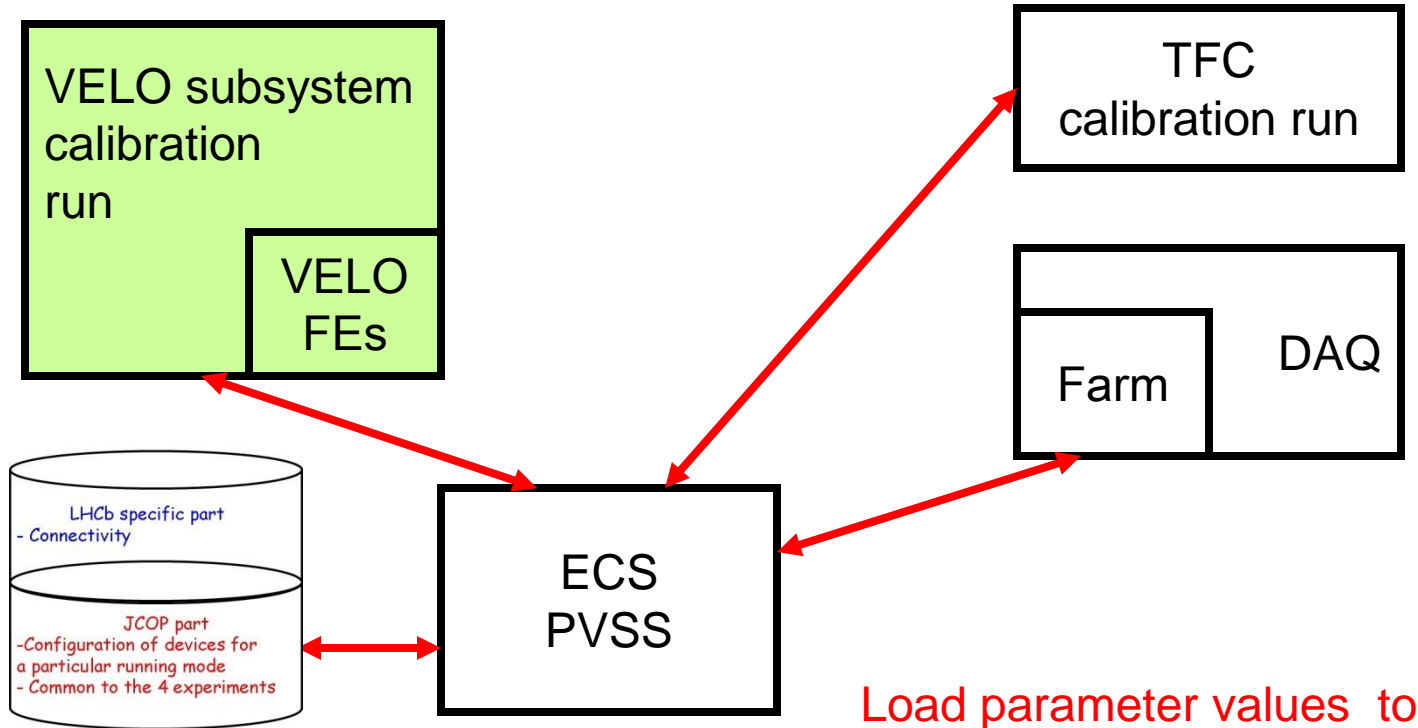


Which devices are involved in the partition? Here VELO

Get connectivity between the VELO and TFC to send the clock.

Get connectivity between VELO and DAQ to configure the network (data path)

Ex: calibration run for VELO (JCOP tool)

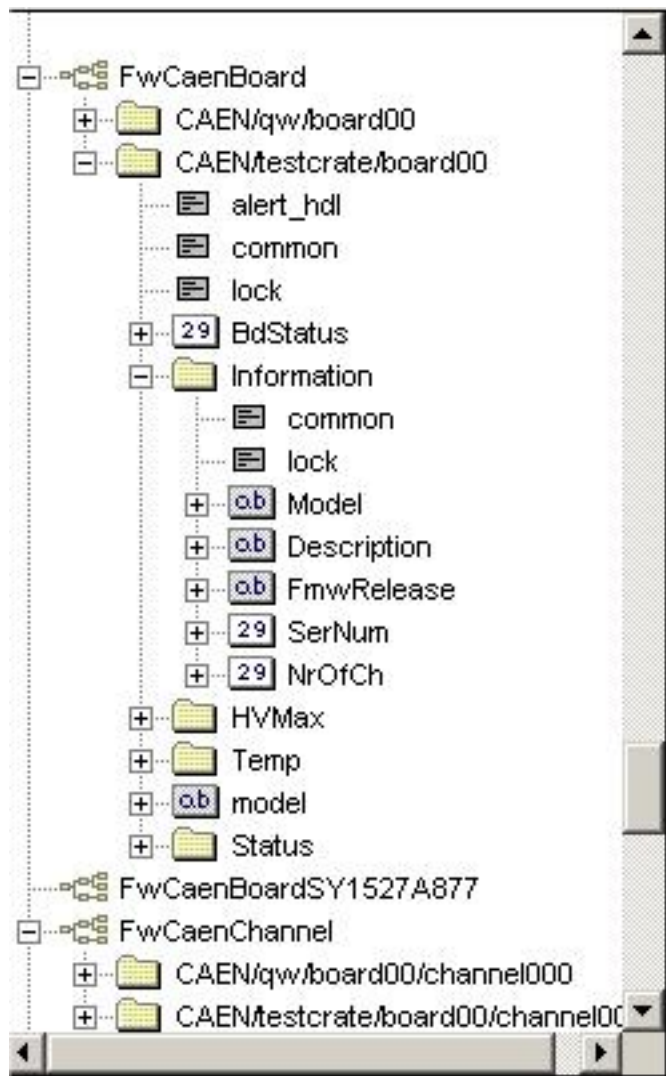


Load parameter values to configure VELO devices, Farm nodes and TFC devices for a calibration run

JCOP tool : Functions

- Framework to use with PVSS (SCADA system)
- Concept of **recipes** : snapshot of a set of parameters and their **values** which change with the **running mode**.
- Save and load recipes for a device or a set of devices (hierarchy) into/from the configuration DB
- Possibility of **tagging** recipes (versioning)
- **Recipe cache** : save and load recipes without connecting to the DB. Useful for
 - Testing setting values
 - Load recipes for successive runs once.
- More info about the JCOP tool...

JCOP Tool : in practice

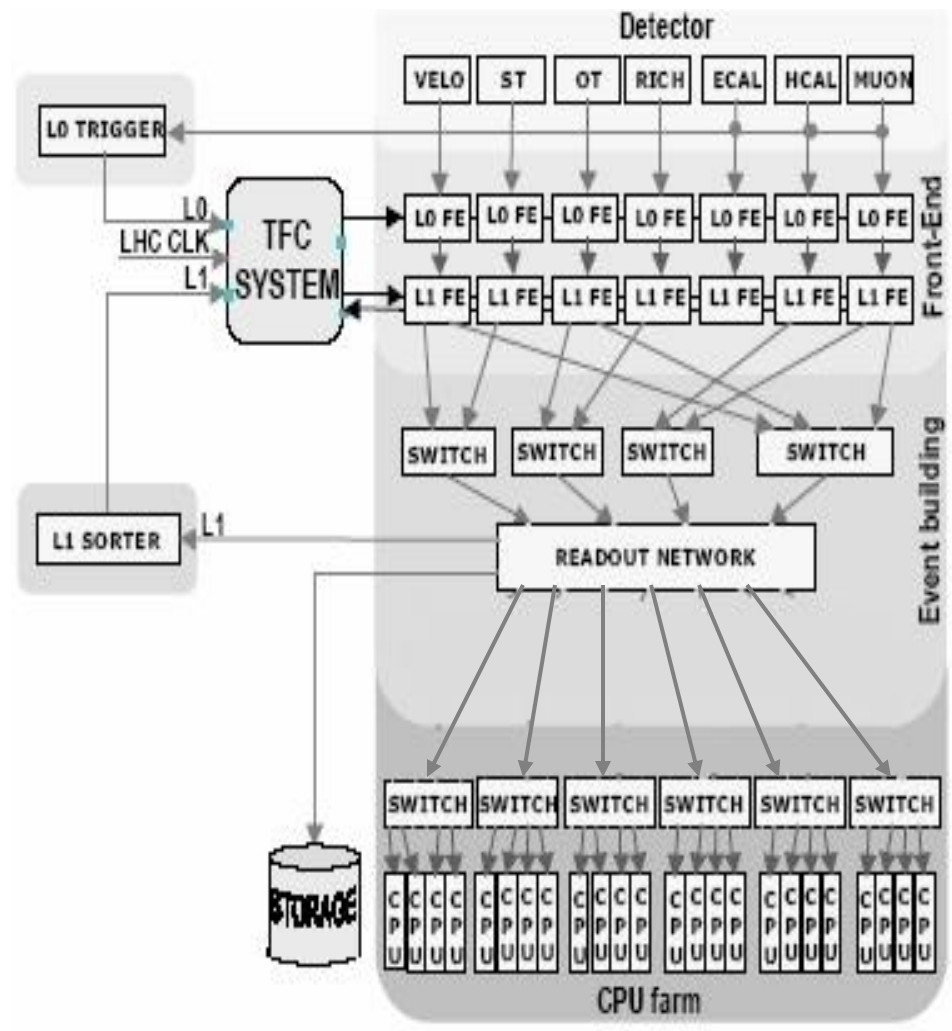


Properties of Framework device: CAEN Channel (FwCaenChannel)

Property names	Show DPT Elements	
	Store Value	Store Alert
Voltage software limit readback		
Voltage max soft value		
v1 set point readback		
v1 set point	YES	
v0 set point readback		
v0 set point	YES	
Under voltage		YES
Trip time readback		
Trip time	YES	
Trip		YES
Switch on/off	YES	
Survey on/off		
Status		YES
Software enabled		

Connectivity

- Description of the physical links between devices on its output and its input using the system dataflow
- Useful to determine
 - Exactly the data path
 - Destination/routing tables
 - Config files (e.g. DHCP)
 - What devices need to be configured given a **partition**
 - How to interconnect all the subsystems with TFC, DAQ and ECS



Connectivity design

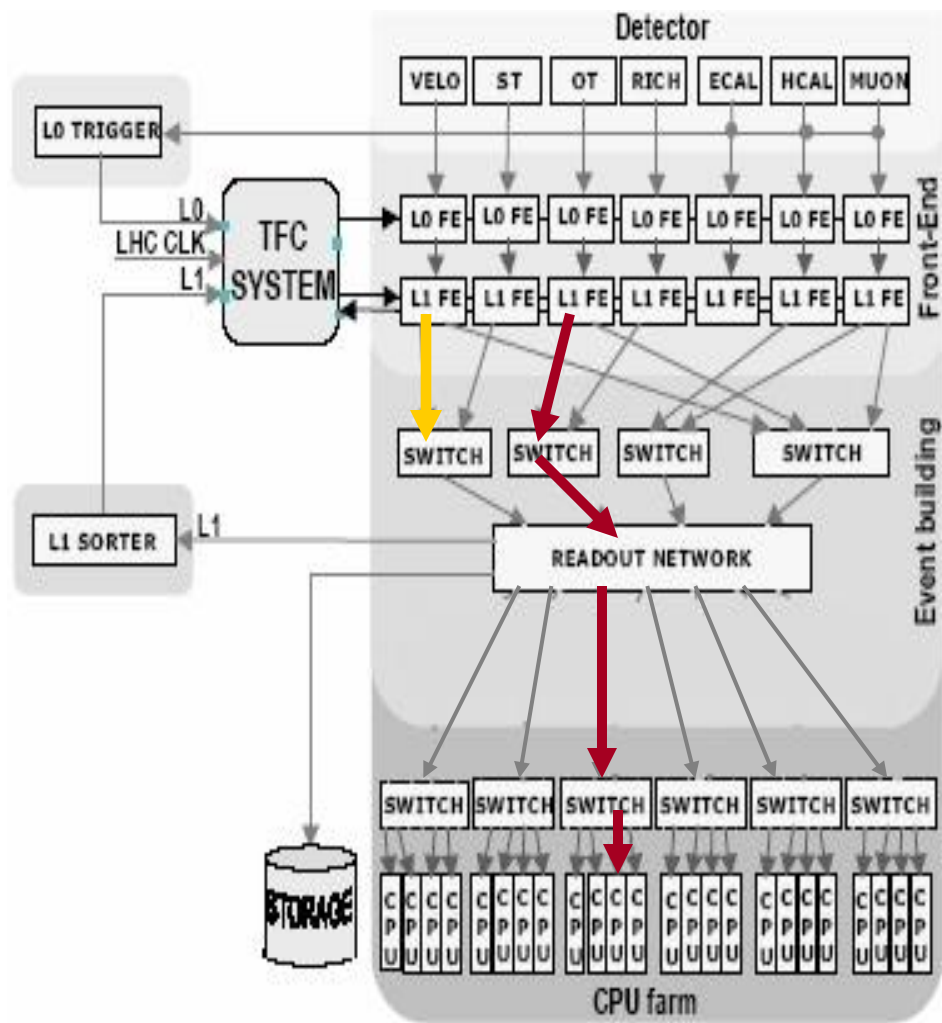
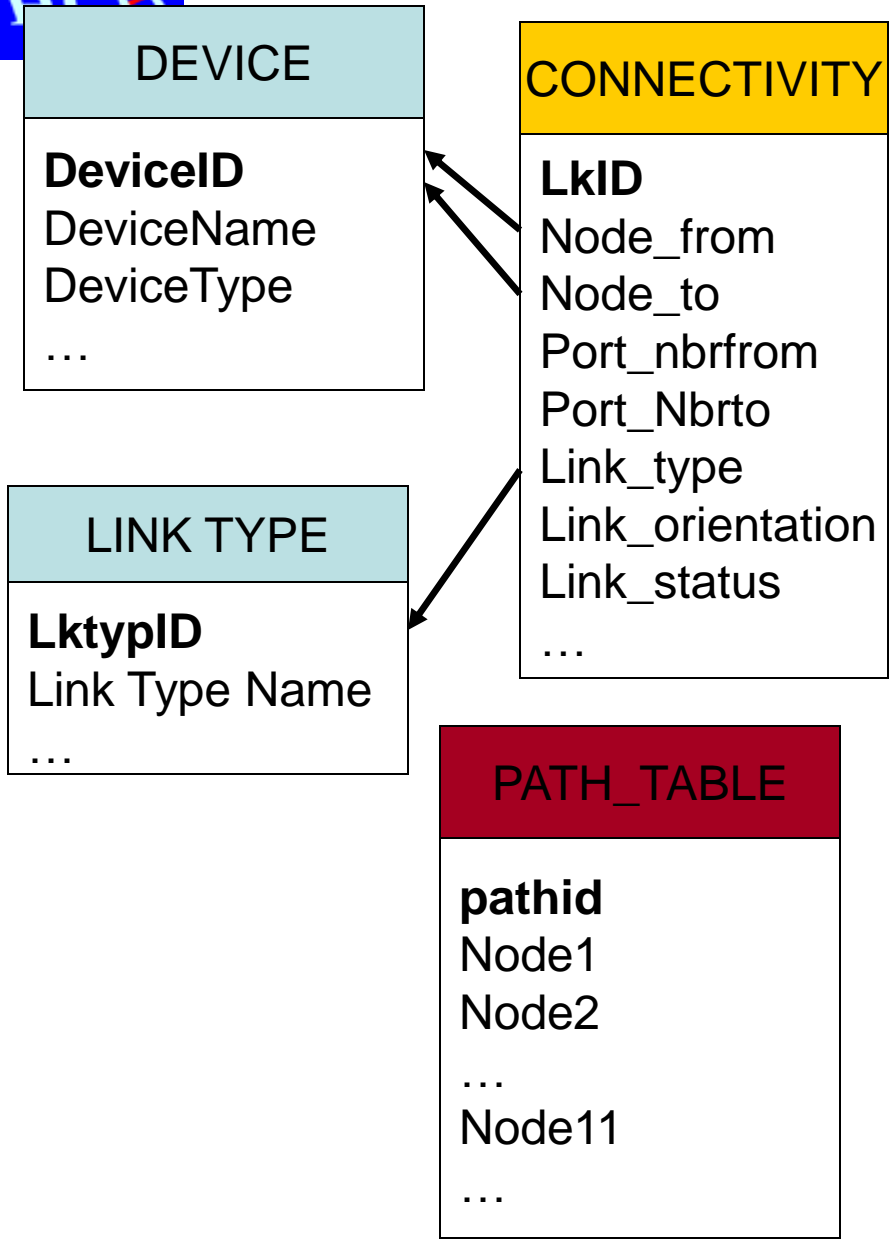
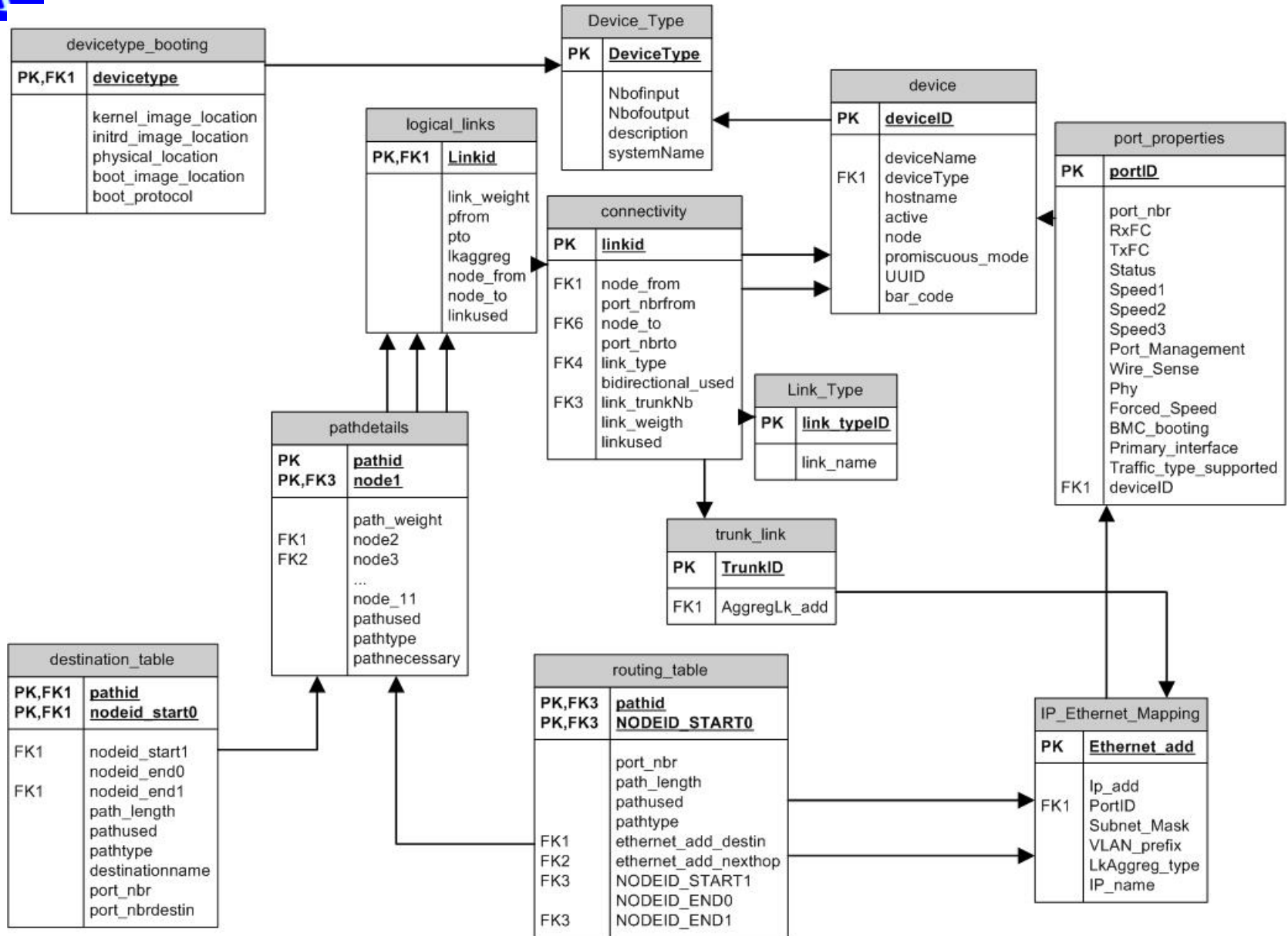


Table Schema



Implementation

- Library **confDB** in *C* using *OCI*, to query information related to connectivity
- **confDB** library has been extended to Python using *BOOST* (**confDBpython** module).
- **confDB** library has been extended to *PVSS* using the *Generic External Handler* (module provided by *JCOP*)

CDbVis

- Python tool (using confDB library)
- Nice view of the content of the database
- Check consistency (error when inserting connectivity between devices)
- Not all the features are implemented yet:
 - You can view the connectivity and get some information about the devices
 - But you can't insert anything

Component View

CdbVis

File View Test Help

Select System
tfc

Select Device
TTCx

TTCx_00
TTCx_01
TTCx_02
TTCx_03
TTCx_04
TTCx_05
TTCx_06
TTCx_07
TTCx_08
TTCx_09
TTCx_10
TTCx_11
TTCx_12
TTCx_13

Device Information

Attribute	Value
DEVICENAME	TTCx_01
DEVICE_DESCRIPTION	TTC transmitter
USER_NAME	None
DEVTYPE	TTCx
ACTIVE	0.0
NBROFOUTPUT	14
NBROFINPUT	16

16 devices found

#: 480 avg: 13.12

The diagram shows a central component labeled THOR_00 (yellow) connected to a component labeled TTCx_01 (green). From TTCx_01, 14 lines radiate outwards to 14 components labeled TTCoc01_00 through TTCoc01_13 (green). The background is blue.

TFC Switch

Transmitter (opt)

Coupler (opt)

Conclusions

- Prototype exists
- Possibility to save and load recipes in/from DB and recipe cache for one or a set of devices.
- Possibility to save and query connectivity information
- Need to finish the implementation of CDBVis
- Need to implement update functions for the confDB library