LHCb Online

&

the Conditions DB

Clara Gaspar, November 2003

Online Usage of Cond. DB

- **Two completely independent users:**
  - The Experiment Control System
    - Writes Online Conditions in the DB
  - The Event Filter Farm Algorithms
    - Need the Conditions for their processing/filtering tasks

LHCb Online

Experiment Control System

DCS Devices (HV, LV, GAS, Temperatures, etc.)

Detector Channels

Trigger

Front End Electronics

Readout Network

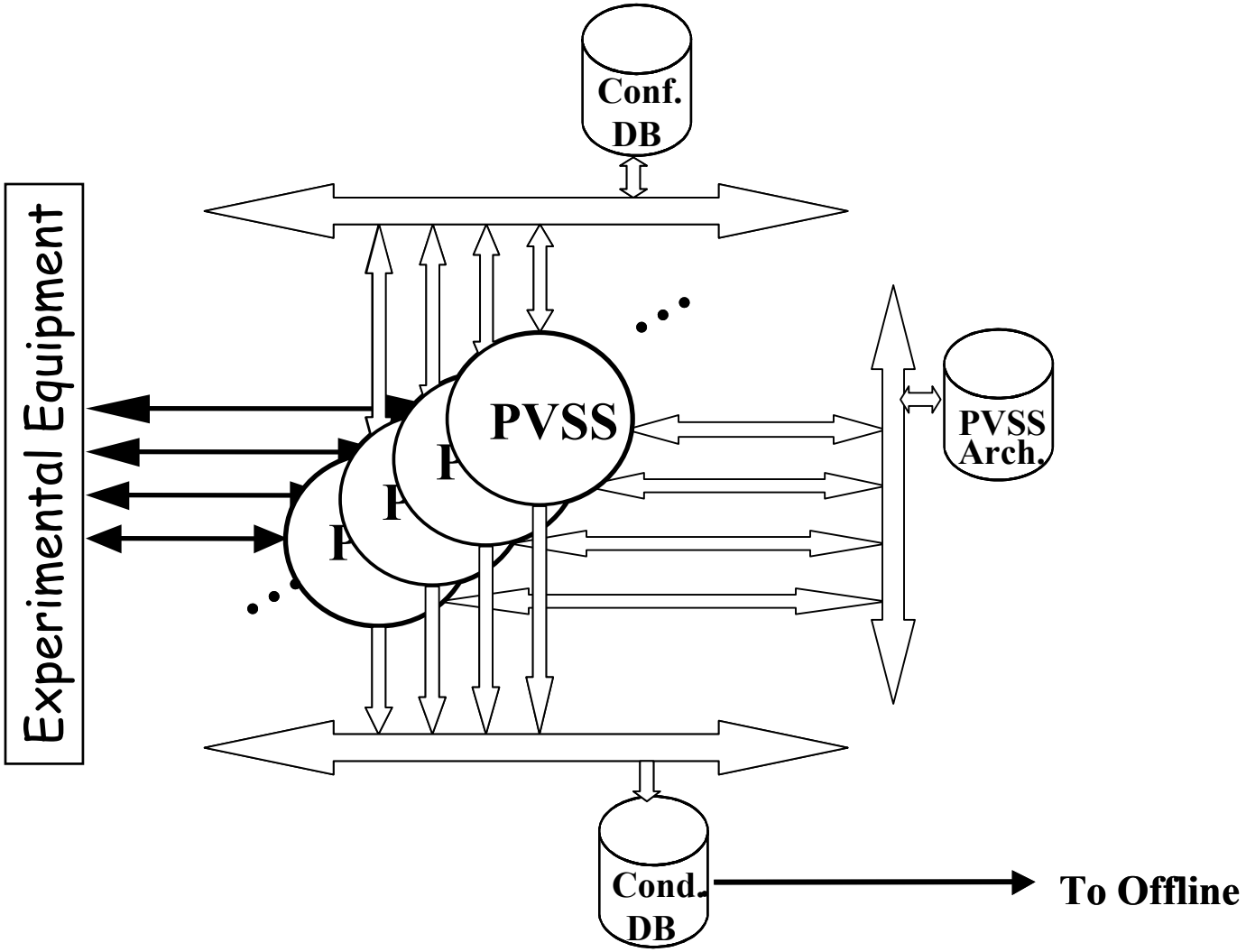
Processing/Filtering Farm

Storage

DAQ

External Systems(LHC,Technical Services,Safety,etc)

Data Handling Architecture



Conditions DB

- **From the Control System point of view:**
  - Only output
  - Only one interface:
    - from the ECS i.e. PVSS
  - Clients (offline algorithms) determine:
    - Data organization
    - Data update rate

Conditions Data Sources

- **DCS**
  - High Voltages, temperatures, pressures, etc.
- **DAQ & Trigger**
  - Pedestals, Errors, Counting rates, noise rates, etc.
- **EFF (Event Filter Farm)**
  - Pedestals, Thresholds, Gain Calibration, Alignment Constants, etc.
- **External Systems**
  - Accelerator data, bunch currents, etc.

Conditions Data Types

- **Raw Data**
  Values read directly from hardware (ex.: HV readings, Temperatures, raw alignment data, etc.)
  - ➡ No Versions, no Tags
- **Processed Data**
  Results of calculations done on raw data (ex.: calibration constants, alignment, etc.)
  - ➡ Automatic Versioning, User Tags

Conditions Data Format

- All data to be stored as data blobs (i.e. no explicit schema)
- But XML (non verbose mode), ex:

```
<tempblock len="150">
    <names>  devno x y mag </names>
    <format>  %d %f %f %f </format>
    1 2.5 3.4 6.7
    ...........
</tempblock>
```

Event Filter Farm Nodes

- **Offline Algorithms (running Online):**
  - Run within the Gaudi Framework.
  - Need Conditions data
  - But:
    - Offline all Conditions are available through the standard mechanisms.
    - Online the Conditions need updating while running and with special constraints:
      for ex. not while an event is being processed.

Conditions in EFF (current ideas)

- Separate Conditions data in blocks:
  - Constant during a run/fill
    (for example detector description)
  - Very slowly varying (very few times per hour)
  - Slowly varying parameters (every few mins)
- Store Condition blocks in memory
  (and provide for each block a shadow block)
  - Use Current Block for processing
  - Fill shadow block with new conditions when necessary
  - At next event if new shadow block -> switch block pointers

➡ **Re-implement Conditions API (from memory)**

Update Mechanism