Databases in CMS

Conditions DB workshop

8th/9th December 2003

Frank Glege

Database types

- We will have 4 kinds of databases in CMS:
  - Construction databases
    - Holds all information about the sub detector construction up to the start of integration
  - Equipment management database
    - Holds all information to physically set up the detector and is used for asset tracking
  - Configuration database
    - Holds all information required to bring the detector in any running mode
  - Conditions database
    - Holds all information needed for event reconstruction

Construction DBs

- Sub detector specific
- Sub detector responsibility
- Already existing for most sub detectors.
- Have to be available for the lifetime of CMS
- Data: construction info, initial calibration data, work flow info, etc.
- Data will partially be copied into the 3 other DBs.

Equipment management DB

- Common for all CMS
- Set up by CMS integration, data entered and maintained by sub detectors
- Partially existing
- Data: detector and electronics parts, cables, racks, crates, location history of all items, etc…
- User interface existing (rack wizard) to enter required information.
- Needs time stamping.
- Few 100MB of live data

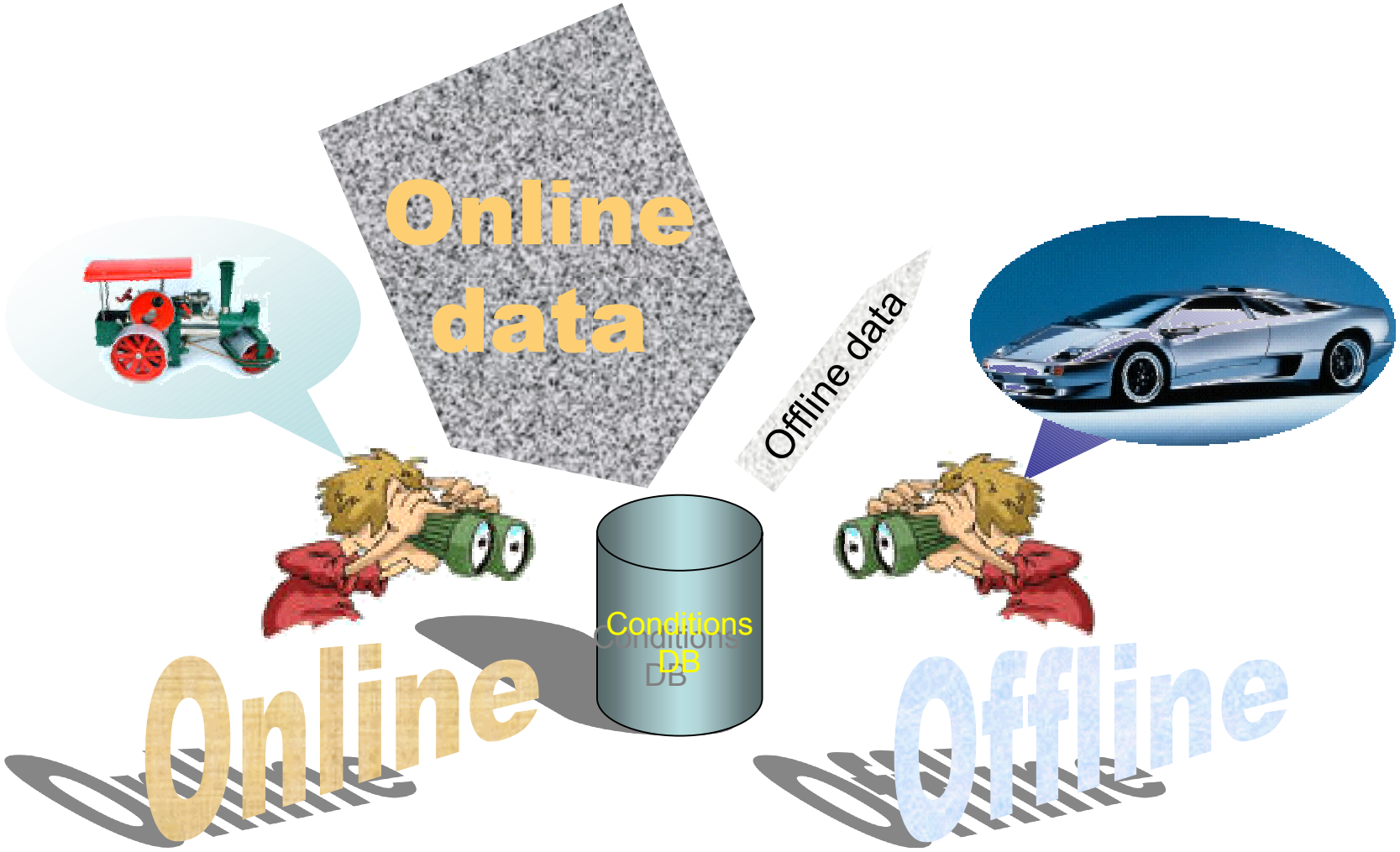
Configuration DB

- Sub detector specific parts on front end electronics configuration/calibration, combined DAQ, Trigger and DCS info.
- Set up by corresponding groups
- Very few parts existing
- Data: see above
- Interface to online software framework existing
- First prototype of interface to DCS existing
- Needs time stamping, versioning and tagging.
- Several tens of MB of live data.

Conditions DB

- Data: all parameters describing the run conditions needed for:
  - offline reconstruction (minor part of conditions data).
  - error tracking in the online system
- DB implementation should be relational.
- Tools for data maintenance provided by LCG?

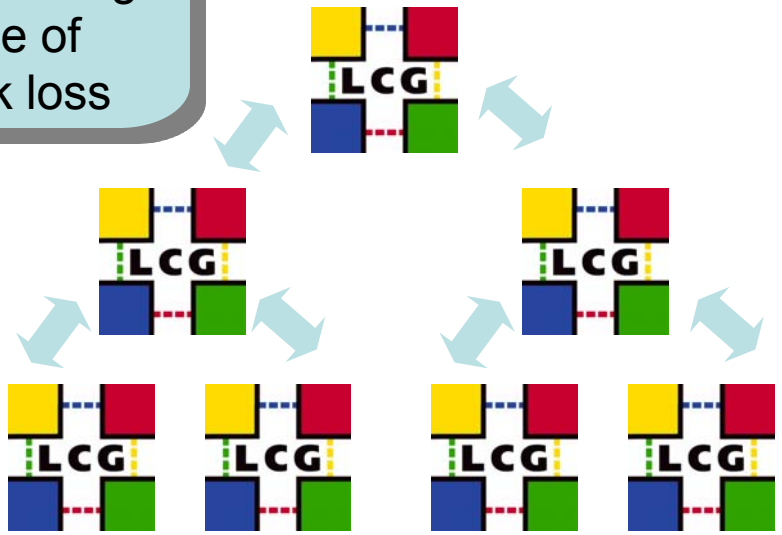
Conditions DB is an o?line DB



Constraints on physical implementation

DB buffer

DB buffer to ensure running in case of network loss



Self contained persistent storage region

The conditions DB data…

- Most of the conditions data is not used for offline reconstruction
- Most of the conditions data stays at the experiment
- Conditions data is the logging of the online system
- Conditions data will be used for error tracking in the online system
- Conditions data will be related to
  - Other conditions data
  - Event quality monitoring
  - Configuration data
  - …

CMS wish list for Conditions DB

- Conditions database shall be relational
- Tools/features of RDB shall be used as much as possible
- Conditions DB data management tools should be provided by LCG (many provided anyhow by RDBMS or existing for RDBs)
- Conditions data handling tools should be provided by LCG. Ideally generic enough to be used with configuration DB (many existing for RDBs)

Conditions DB API **definition**

- The conditions DB API definition should
  - Reflect the need for time stamping, tagging and versioning
  - Allow for a *high performance* implementation
  - Be independent of the DB implementation

BUT

- Is a classical API the way to go? (needs multiple implementations, etc.)
- Do alternatives exist? (web services, etc.)

Conditions DB API implementation

- CMS has no experience with the current API implementations.
- Assuming a RDB implementation with intrinsic meta data handling and data management functionality
  - *Will the actual implementation be less work than the adaptation of the reconstruction framework and the persistent storage system to the API?*

CMS Conditions DB plans

- Ideally we would have a conditions DB prototype for DC04
- More realistically we should have conditions DB prototypes for next years test beams
- Even more realistically we should have one for our slice test end of 2004
- Indispensable: we have to have the conditions DB somewhere in 2005

Manpower

- CMS would need 2 to 3 people more to work on databases 
- Current manpower situation:
  - 2 FTE at 10%
  - 1 FTE at 20%
  - 1 FTE 100% (will leave soon)
  - ➢~0.5 FTE

Conclusions

- The conditions DB will be used by on- and offline
- DB implementation should be relational
- Logistics for conditions DB should be provided by LCG
- Existing tools should be used as much as possible
- Man power situation in CMS is very difficult