# CMS database software status

Zhen Xie
Princeton University

# Overview of CMS database applications (I)

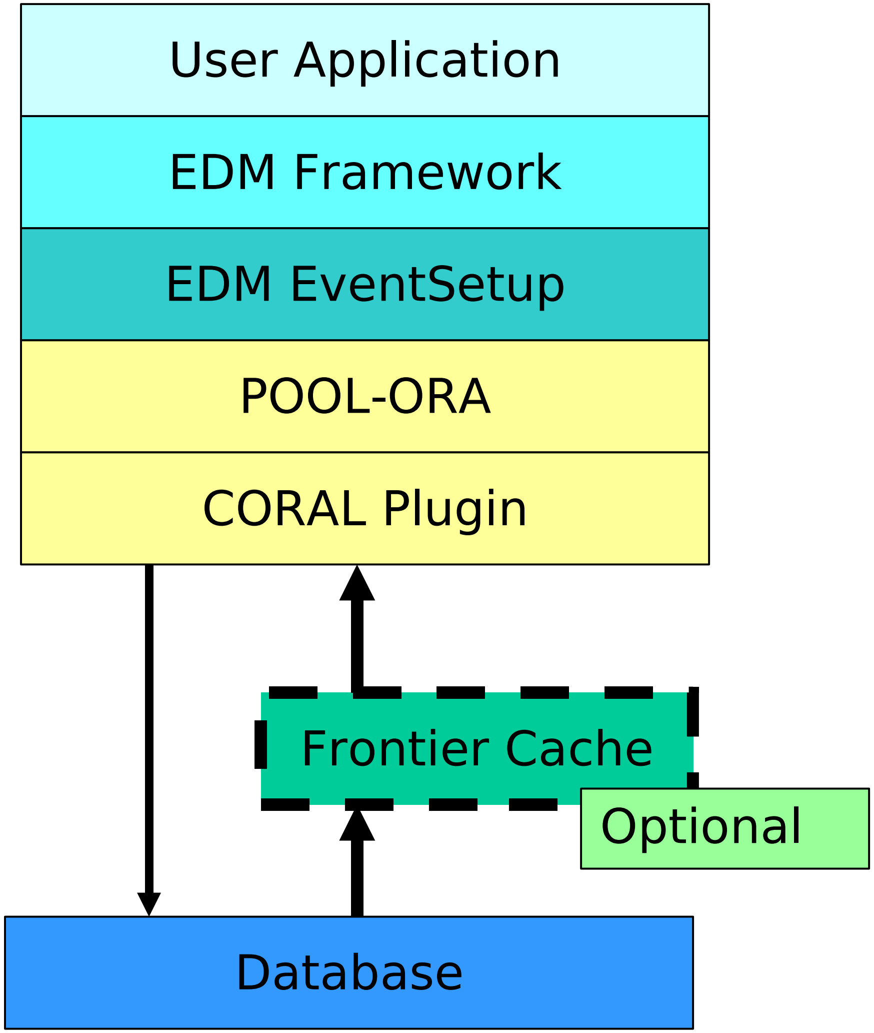
- File catalog, dataset catalog, data transfer and production bookkeeping system
  - DBS, DLS(uses LFC-catalog), RefDB, Dashboard, PhEDEx, ProductionAgent, BOSS
  - Applications identified but no clear definition of T0 workflow yet. Oracle service “at CERN” is required for these applications, MySQL deployment at T1.
    - Some legacy application will still use CMS-operated MySQL DB at CERN in 2006
  - At present there is no indication that replication or distribution are needed for 2006. DBS and DLS may need replication in the longer term, while the others no.
  - Applications in SC4.

S.Belforte, P.Elmer, L.Bauerdick, T.Wildish

# Overview of CMS database applications (II)

- Conditions db applications: calibration and alignment
  - In April-May Magnet-Test/Cosmic-Challenge(MTCC), not in SC4
  - Commissioning of conditions db is in the critical path of MTCC
  - Oracle service is required at P5 and T0
  - Oracle streaming is required at CERN from P5 to T0
  - FroNTier deployment is required at T1
- This talk will focus on the readiness of CMS conditions db application for MTCC

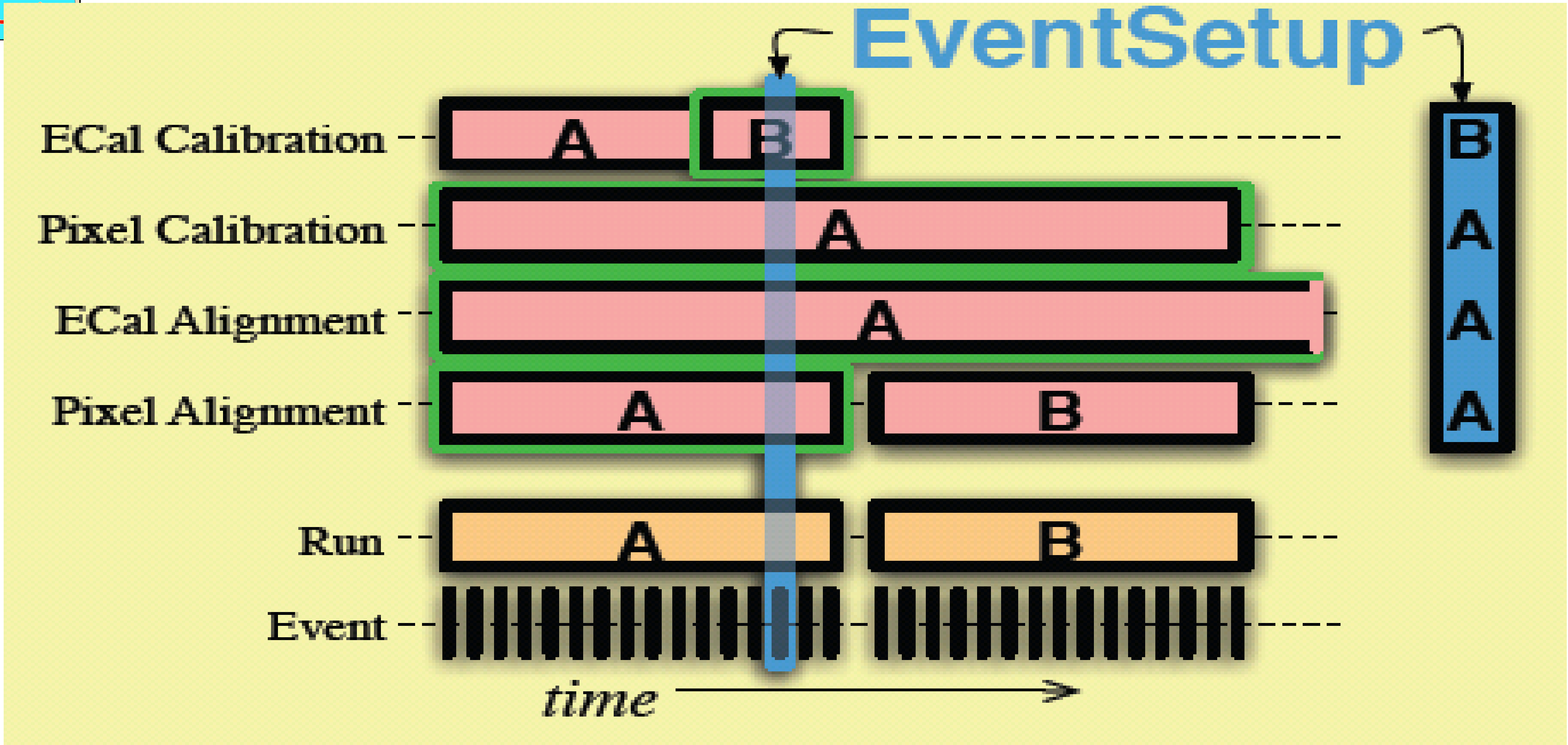
# Conditions db Software stack



- EDM EventSetup assures the correct non-event data is accessed and available for the user application.
- POOL-ORA (Object Relational Access) is used to map C++ objects to Relational schema.
- A POOL-RAL/FroNTier-Oracle plug-in is used to to enable a middle-tier proxy/caching service for read-only access.
- ORACLE is required at T0
- FroNTier is required for data distribution at T1 and beyond.
- Other technologies, e.g. MySQL, SQLite are required for application testing in the development process.

Vincenzo Innocente

# non-event data framework



- Provides a unified access mechanism for non-Event data
- **EventSetup** “snapshot” of detector at an instant in time
- **Record**: holds data with same interval of validity
- Not a new idea: has been used by CLEO experiment since 1998

**Chris Jones**

# Interval Of Validity (IOV) model

- Interval of Validity(IOV) – the range of time for which a set of non-event data is valid
- IOV is pure offline concept
  - Important concept because online and offline db are distinct in CMS
  - The assignment of IOV is carried out offline by algorithm or person in charge of providing a certain conditions data set for the offline (and HLT) operation
  - Data stored in the online db, such as data taking time, may be used to generate IOV
- Modifying IOV means a full new IOV set is created
  - Do not store delta in time
  - No update of the old values
  - Light management

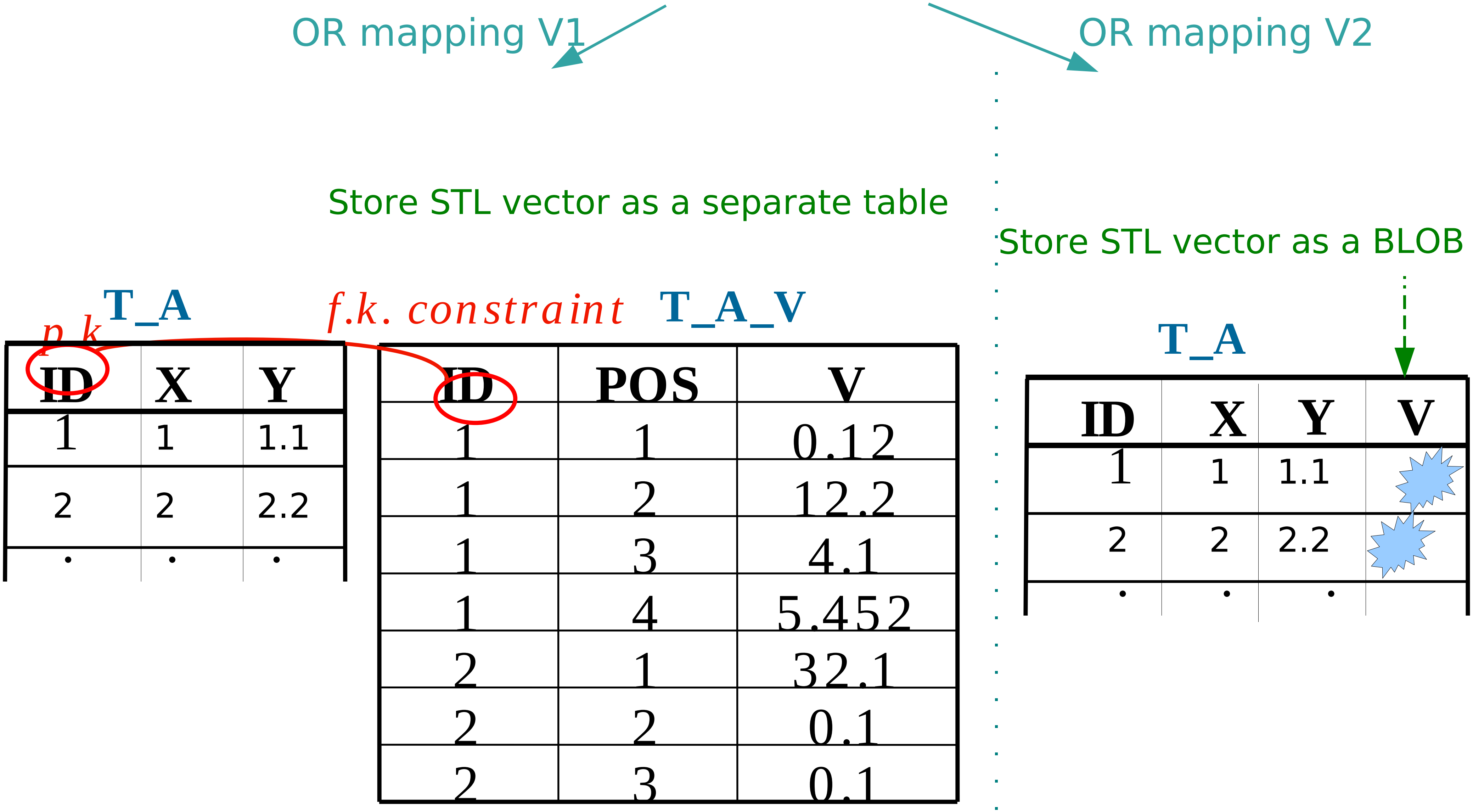
# POOL-ORA for offline db

- Relational backend of POOL
- POOL-ORA provides an object (as in C++) to relational (as in DB) mapping. The schema in the DB is “generated” based on the “shape” of the object.
  - Normal C++ object, no database related info in object itself
  - Various tools are being provided that facilitate managing POOL objects and OR mappings
- POOL-ORA used for IOV management
  - One central module interfaced with the EventSetup
- POOL-ORA used for data payload
  - Each detector has to model its data as C++ objects

# Data Objects and Object relational Mapping

- Objects must be described for each kind of non-event data
  - Calibration: pedestals, gains, crosstalk, etc
  - Geometry and alignment
- The description of the data object consists of its C++ class in a header file.
- POOL-API generates the schema from the C++ header of the object, and subsequently store the data in the database
- XML files are used to guide the object relational mapping process
  - The same C++ object can be mapped to different database structures and storage type
  - The mapping has impact on performance!

```
struct A{
 int x; float y; vector<float> v;
};
```



OR mapping V1

Store STL vector as a separate table

p.k.

T_A

| ID | X | Y |
|---|---|---|
| 1 | 1 | 1.1 |
| 2 | 2 | 2.2 |
| . | . | . |

*f.k. constraint*

T_A_V

| ID | POS | V |
|---|---|---|
| 1 | 1 | 0.12 |
| 1 | 2 | 12.2 |
| 1 | 3 | 4.1 |
| 1 | 4 | 5.452 |
| 2 | 1 | 32.1 |
| 2 | 2 | 0.1 |
| 2 | 3 | 0.1 |

OR mapping V2

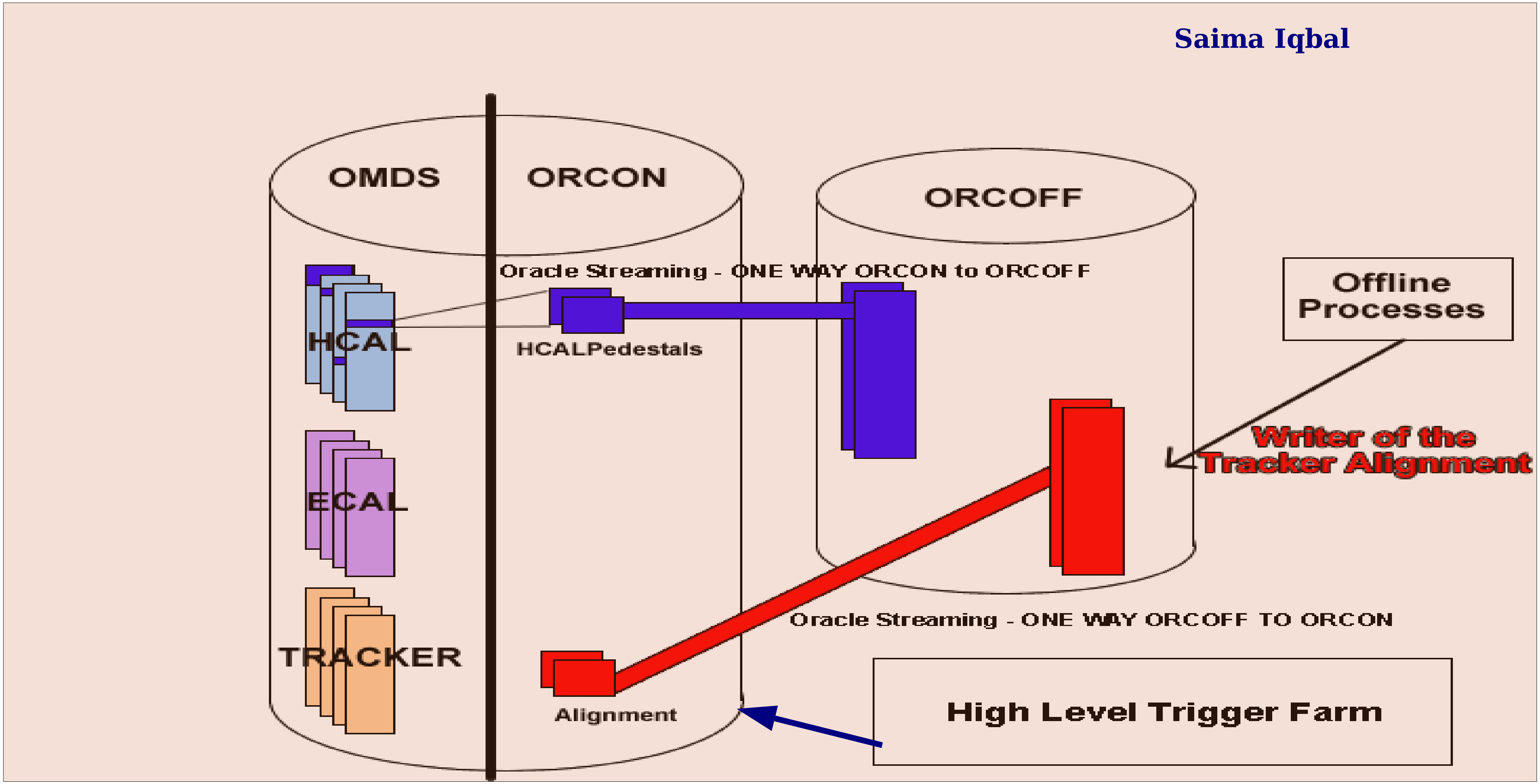
Store STL vector as a BLOB

T_A

| ID | X | Y | V |
|---|---|---|---|
| 1 | 1 | 1.1 | ✹ |
| 2 | 2 | 2.2 | ✹ |
| . | . | . | |

# Online to offline data transfer

**RDBMS (online)** → transformation → **POOL-ORA (nearline)** → streaming → **POOL-ORA (offline-T0)**



**Saima Iqbal**

OMDS
ORCON
ORCOFF
Oracle Streaming - ONE WAY ORCON to ORCOFF
HCAL
HCALPedestals
Offline Processes
Writer of the Tracker Alignment
ECAL
TRACKER
Oracle Streaming - ONE WAY ORCOFF TO ORCON
Alignment
High Level Trigger Farm

# CMS Online to offline transfer activities

- **Development of Individual Sub-detector's Online Schema**: Provide support to individual sub-detector to develop their online databases at Pit-5 Oracle server

- **Data Transformation Test**: Provide support to individual sub-detector to filter the data from online database which they needed to generate POOL-ORA objects and transfer to Offline database

- **Data Transfer Functionality Test**: Data transfer by using Oracle Stream in between CMS online Oracle database server i.e. Pit-5 server and an Oracle database server at IT

- **Scalability/Performance Test**: Test Scalability/Performance of developed application and Oracle Stream for the data of the range of GB (*at least*)

- **Data Transfer Monitoring**: Test the reliability of data transfer in between Pit-5 Oracle server and server at IT

- **Failure Mode Tests**: Test Online-to-Offline data transfer with different failure modes

- **Reverse Streaming**: Test data transfer from Offline-to-Online database

# POOL DB Data Access pattern

- Data access mode
  - Payload objects are never updated
  - IOV objects can be rewritten but are never updated
- Connection
  - Two connections open per calibration task: one for IOV lookup, one for payload retrieval
- Transaction
  - The entire payload object(all electronic channels) is loaded in memory in one go
  - Channel lookup is in-memory C++ operation
  - Keep db interactions to minimum

# POOL DB Data Volume (predictions...)

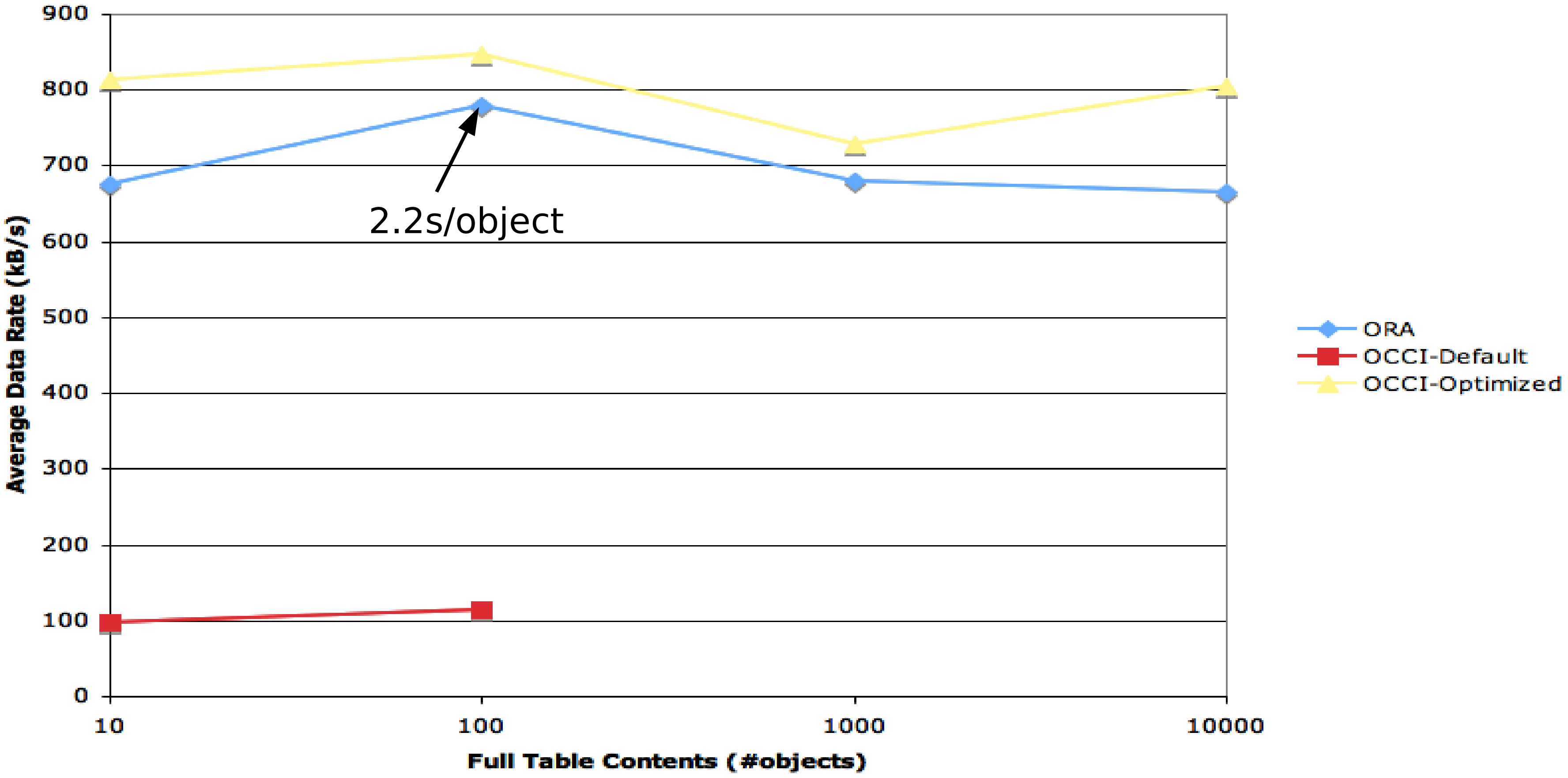
- Estimation for calibration payload.
  - Not a reference! Difficult to predict without real data taking.
- ECAL (R. Egeland)
  - 61,200 channels; 1.6 MB per pedestals object(per I/O)
  - 12,000 Pedestals in 6 months of 24/7 runtime (not for MTCC! )
- HCAL (F. Ratnikov)
  - 9,072 channels; 0.2 MB per pedestals object(per I/O)
  - A few other objects of similar size
- SiStrip Tracker (G. Bruno)
  - 40MB per pedestals object(per I/O)
- Muon detectors (U.Gasparini)
  - CSC: ~226,800 channels; 4 MB per crosstalk object. 1.2GB per year of LHC running (not for MTCC!)
  - DT: 192,000 channels
- Pixel
  - Not in MTCC

# Preliminary performance study

- POOL-ORA/oracle performance vs. vanilla OCCI studied by ECAL using pedestals object.
  - Later confirmed by HCAL
- Conclusion
  - The application is I/O bound (no surprise)
  - POOL has small overhead w.r.t vanilla OCCI
  - Current performance acceptable for ECAL
- There's room for improvement with new POOL version
  - User tunable prefetching parameter
  - Possibility of mapping stl containers to BLOB

# ECALPedestals offline db read performance

Average rate of reading 10 objects vs. database size



2.2s/object

Ricky Egeland

# Status

- POOL-ORA Object definition, eventsetup infrastructure testing completed for most detectors in MTCC
- Online schema to POOL-ORA schema transformation procedure is ready for most detectors
  - But everybody uses different procedures...
- Most detectors participating MTCC are able to serve existing test beam data from their own servers as conditions objects in offline calibration application
- Tuning object relational mapping and data compression to reduce I/O size for SiStrip tracker; performance so far acceptable for ECAL
- Oracle streaming infrastructure set up and tested

# Plan

- Integration
  - Full chain OMDS->ORCON->ORCOFF->calibration application tests with all detectors
- Offline Software
  - Enable blob support
  - Enable synchronization with MTCC timestamp
- Online to offline transfer
  - Harmonize schema transformation procedure of different detectors
  - transfer bookkeeping and monitoring
  - Test reverse data streaming from ORCOFF to ORCON
  - Set up individual stream for each detector