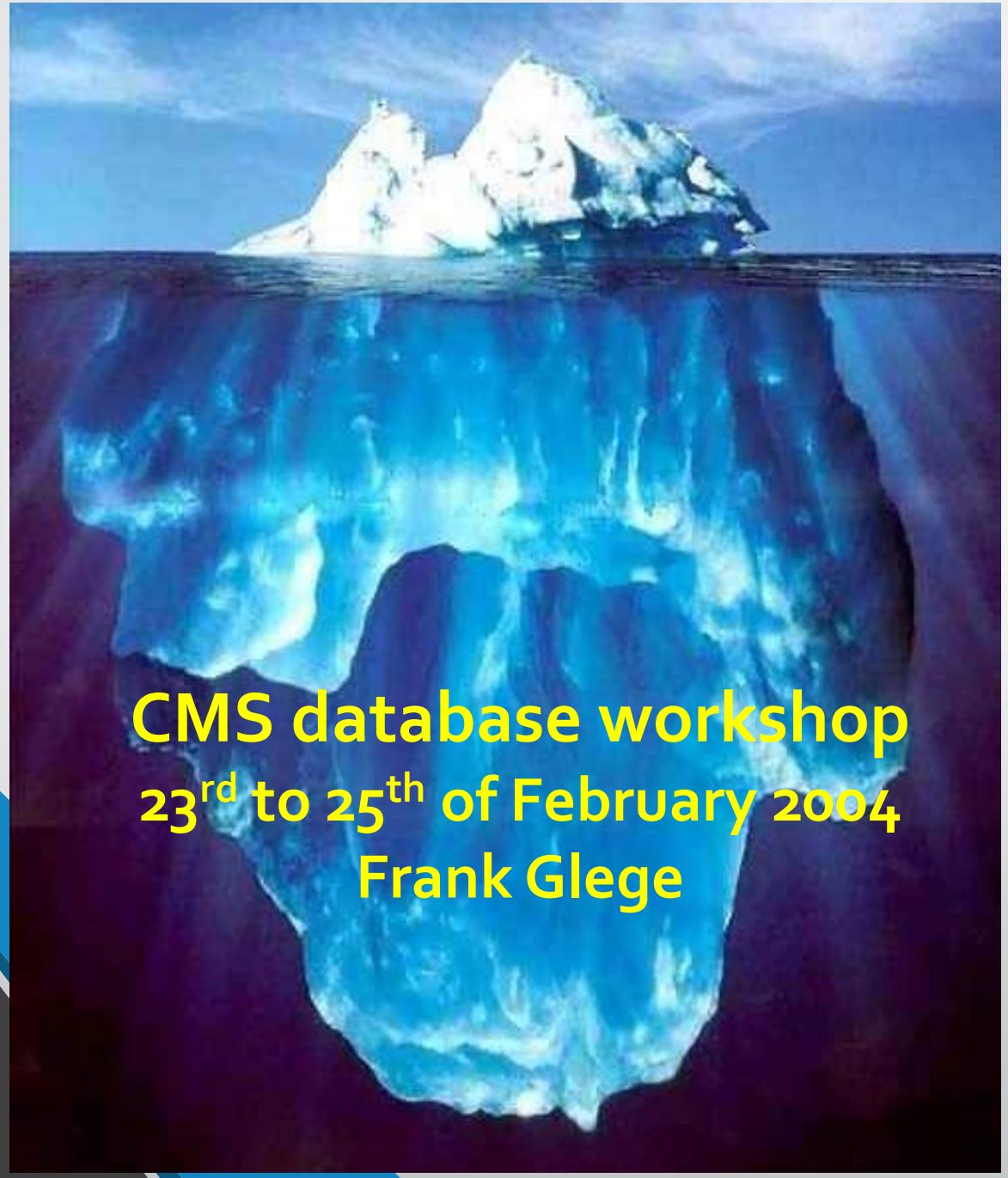
Online DBs in run 3

DAQ@LHC 2016

Frank Glege on behalf of several contributors of the LHC experiments

Scope

- Configuration DBs DAQ and DCS
- On line conditions DB DAQ and DCS
- No off line conditions DB
- No auxiliary, technical DBs like asset management and tracking, geometry, materials, etc.


CMS database workshop
23rd to 25th of February 2004
Frank Glege

Databases at LHC experiments

- The usage of relational DBs, at least to the current extend, is new
- DBs have become a vital part of the complete experiment system
- DBs are often used as data dump rather than intelligent data store
- Not all DB developers know the concept of normalization and differences in read/write optimized structures
- DBAs at CERN play a very important role by monitoring performance, consulting and proposing improvements

Online Master Data Storage

Create rec conDB data set

Offline Reconstruction
Conditions DB
ONline subset

Configuration

Conditions

Calibration

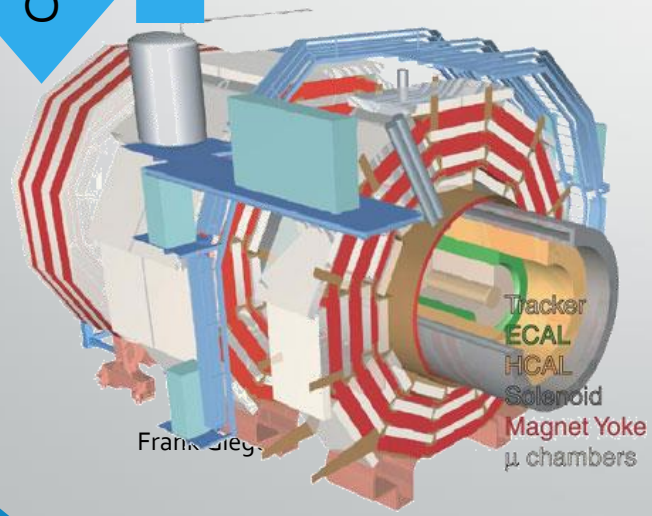
Conditions

Bat 513

Master copy

Offline Reconstruction
Conditions DB
OFfline subset

LCG



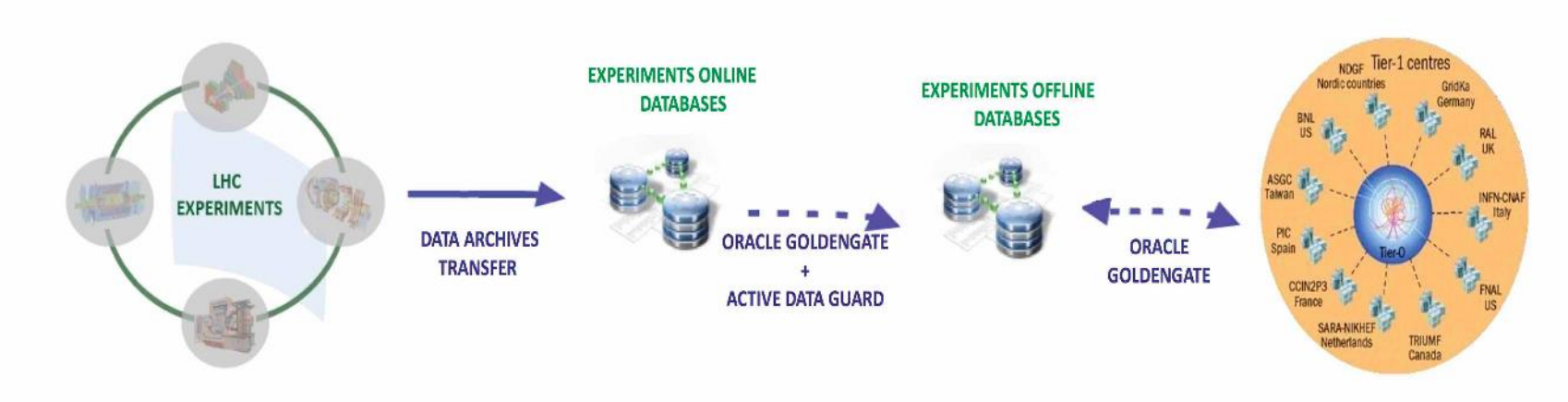
LCG LCG LCG LCG

Database technology

- Online DBs fit well into relational scheme
  - Conditions are written sequentially but read randomly
  - Configuration is highly relational and requires integrity to be ensured
- ORACLE RAC is a good solution for online DBs. No need to change.
- Current sizes:
  - ALICE: 7TB
  - ATLAS: 12.5TB
  - CMS: 11.5TB
  - LHCb: 15TB

ORACLE evolution

- Current versions: Oracle 11.2 and 12.1
  - Supported until 2020 and 2021 respectively
- LS2 recommended upgrade to Oracle 12.2 (release date 2016)
  - Main new feature: In-Memory database bringing real-time responsiveness
  - Bug fixes
- Replication scheme unchanged:



Additional DB services

- DB on demand
  - MySQL CE 5.7
    - Q1/Q2 2016
    - To stay for next 3 years
  - PostgreSQL 9.4.5
  - Oracle application server for General Service
- Hadoop Service in IT
  - Setup and run the infrastructure
  - Provide consultancy
  - Build the community

DCS conditions DB

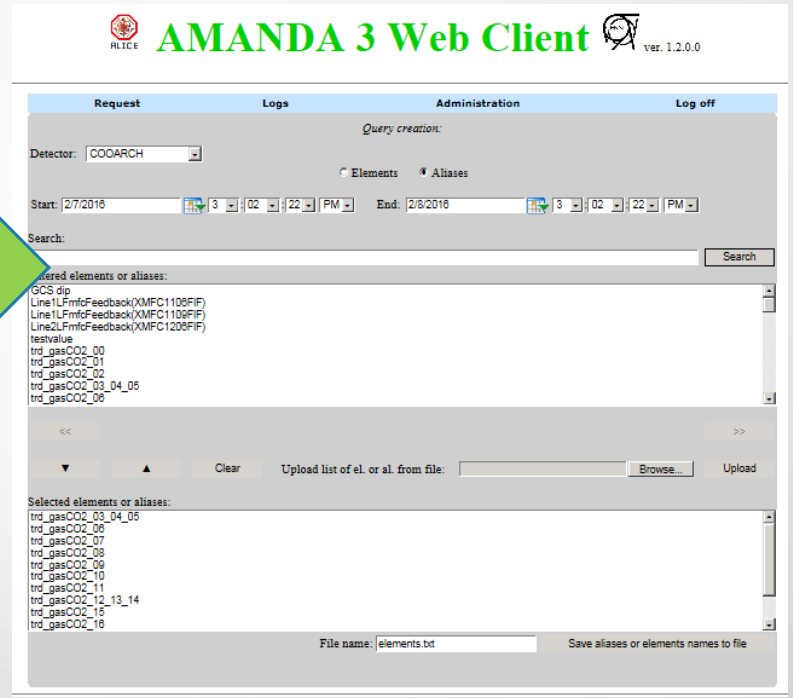
- Holds mainly the status of the detector parts at any time
- Most of the condition data is produced by the DCS.
- All LHC experiments use ORACLE DBs with Win CC OA schemas for primary storage of DCS conditions data (pure relational).
- Transfer/conversion to off-line conditions
  - ALICE: WinCC OA-> ADAPOS(ALICE datapoint server)-> data stream
  - ATLAS: ATLAS: WinCC OA -> COOL
  - CMS: WinCC OA -> O2O -> CMS offline conditions format (ConDB v2.) / CORAL
  - LHCb: WinCC OA-> WinCC OA-> DB Writer -> COOL

ALICE DB infra structure layout

DCS Cluster
(~100 DB clients)

RUN 2 DB Service
Primary DB

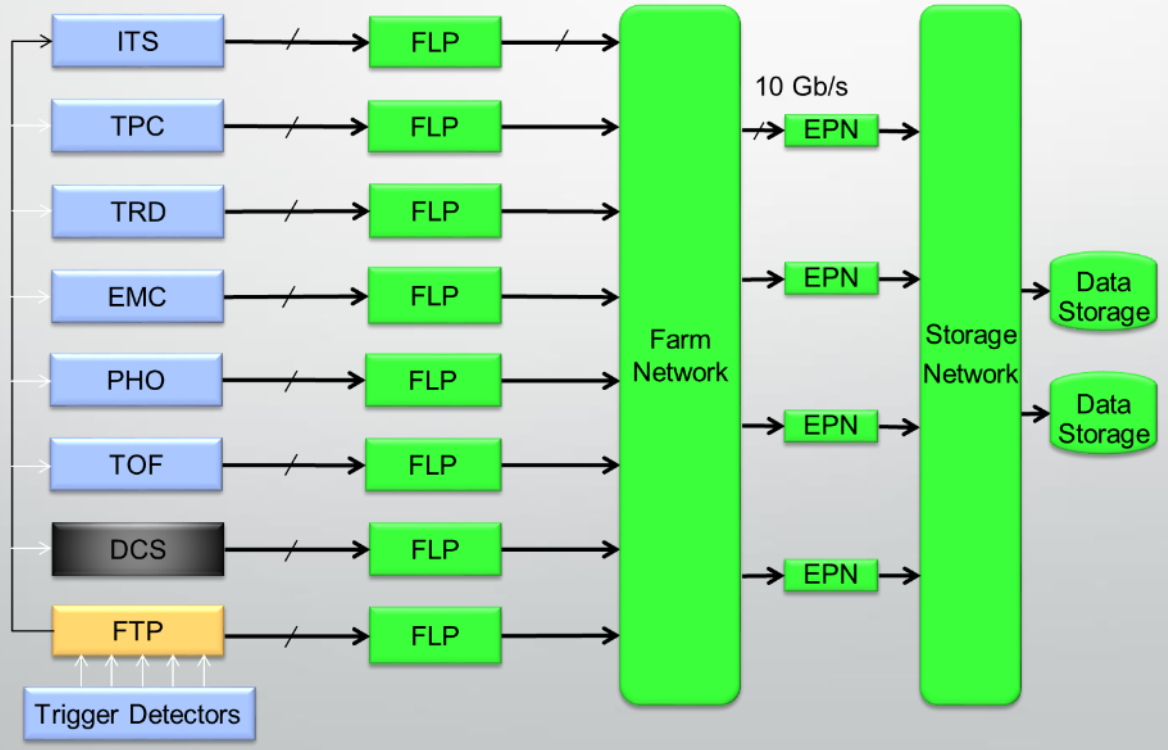
DB Replica in IT
Standby DB



DCS GPN

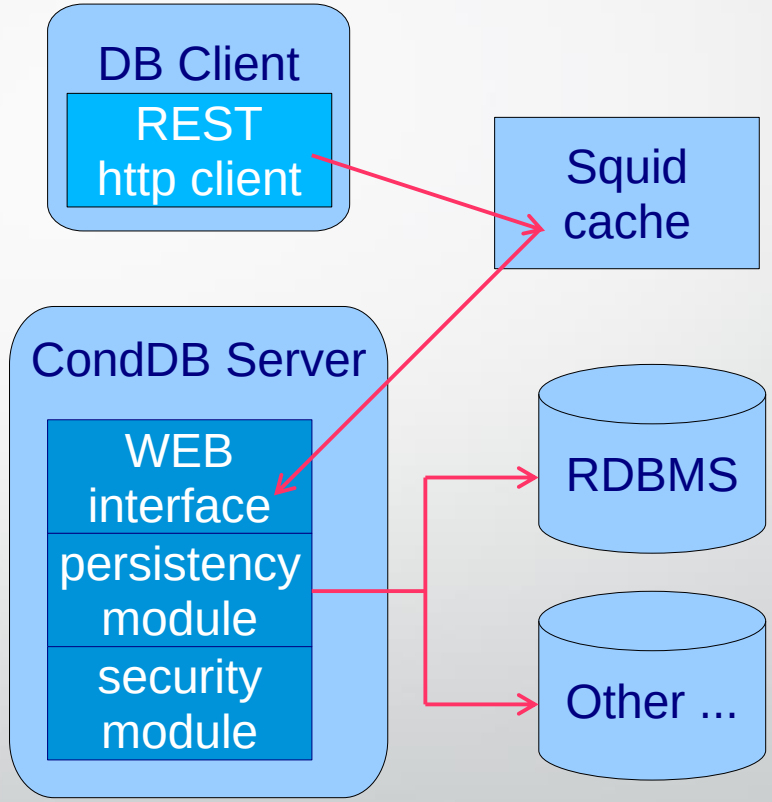
DAQ conditions DB

- Holds mainly the condition of the DAQ system during data taking
- ALICE
  - Using the O2 infrastructure to add conditions to the data stream



ATLAS DAQ conditions DB

- ATLAS currently uses LCG COOL database. This might be replaced by a new system
- Conditions DB server provides only REST API
  - Use Squid cache for scalability
  - No relational DB libraries for clients
- Simplify database schema based on architecture of CMS conditions DB: few tables instead of O(1K)
  - Payload storage (a BLOB) is opaque to the conditions server => cannot perform payload queries
  - Open ended IoV (no *until*) => no holes in intervals vs. COOL
  - No channels, handled inside payload
- 2016 – proto, 2017 – development & migration, 2018 – tests / validation



scalable architecture using Squid cache for read-only access

DAQ conditions DB

- CMS
  - Run control (JAVA based) writes directly to ORACLE partly using name value pairs
  - Core DAQ SW (XDAQ) provides an DB abstraction layer to write to DB forcing a normalized relational structure

DAQ conditions DB

- LHCb
  - ECS (WinCC OA) writes to ORACLE via a DB Writer Process

DCS configurations DB

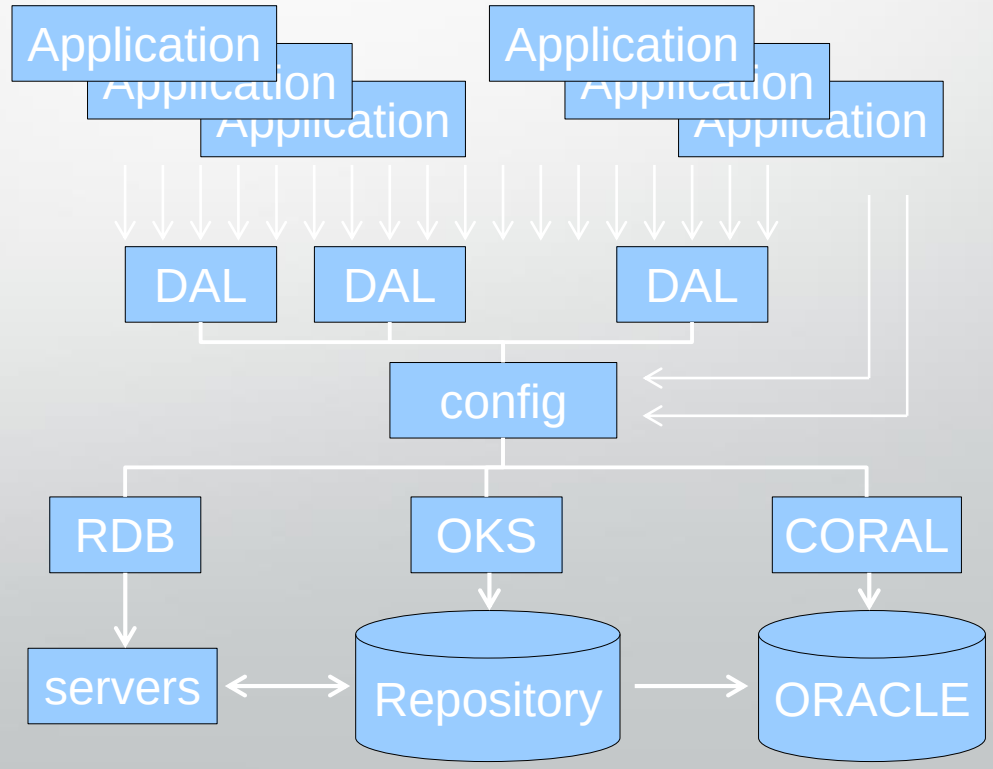
- Configuration data read and interfaced by DCS has different scopes in the different experiments
- ALICE
  - Configuration of classical slow control and parts of detector front ends
  - Stored in ORACLE read by proprietary mechanisms
- ATLAS
  - Uses JCOP FW tools for configuration
- CMS
  - Only configuration of classical slow control
  - Uses the JCOP FW confDB component
- LHCb
  - Experiment Control System (ECS) manages the whole experiment (DAQ+DCS)
  - Uses the JCOP FW confDB component

ATLAS DAQ configurations DB

- Used by T-DAQ and sub-detectors (100 MB of data / online configuration)
- Is based on OKS (Object Kernel System) supporting object data model (classes and named objects, attributes (primitive or class types), polymorphic multiple inheritance)
- OKS stores data in XML files supporting inclusion. Data integrity is protected by the OKS server containing full history of DB commits and write access restrictions using role based access manager policy.
- Used configurations are archived to Oracle using CORAL for offline.
- Tree of CORBA servers (RDB) is used for remote access. The root server distributes and controls current configuration. Multi-level hierarchy of servers addresses performance & scalability requirements of 10^4 clients.
- There are several generic schema and data editors (Motif and Qt based).

ATLAS DAQ configurations DB

- Programming access is implemented via abstract config interface supporting oks-xml, rdb and oracle archive implementations.
- On top of the config interface a number of DALs (data access libraries) are automatically generated for concrete DB schemas (sets of OKS classes) for C++ and Java. The Python binding creates classes on the fly reading the DB schema.
- Use template algorithms for similar objects (DQM, applications / segments) to reduce DB size and improve maintenance.
- The DALs and algorithms are mostly used by user applications accessing configuration description.



CMS DAQ configurations DB

- Run control configuration read by run control (JAVA) directly.
- FE configuration read from ORCALE using the XDAQ DB abstraction mechanism XDATA, a library for reflecting C++ data structures in various data serialization formats.

LHCb DAQ configuration DB

- LHCb has the Experiment Control System (ECS) based on the WinCC OA SCADA which controls the whole experiment:
  - Trigger
  - Front-end hardware control
  - Readout Supervisors
  - Event Builder control
  - HLT
  - Monitoring
  - Storage
- Configuration of all the sub-systems is treated in a similar manner

LHCb DAQ configuration DB

- DAQ configuration is done using standard JCOP Tools (fwConfDB)
- Configurations are stored in an Oracle DB (Configuration DB)
- Configurations are stored as named Recipes
  - Recipe types define for each configurable device type which settings are to be stored in a Recipe
  - Recipes implement the Recipe types with the valued parameters
  - The Recipe names follow a convention - hierarchy of activity type (e.g. “PHYSICS|pA|VdM”)
- The Recipes are applied automatically according to the currently set activity in the Run Control
  - Recipes to be loaded are matched by name to the set activity
  - Recipes loaded follow the activity type hierarchy
    - e.g. If the set activity name is “PHYSICS|pA|VdM”, the devices to be configured will check if a recipe that matches the name and apply it, if there isn’t, they’ll check if there’s a “PHYSICS|pA”, if there isn’t they’ll proceed to check for one named “PHYSICS”

Summary

- Databases are a vital part of the LHC experiments
- The chosen technology (ORACLE RAC) fits well for the on line DBs and will continue to be used
- The current DB implementations have proven to work well
- Service will continue as is in run 3 with minor changes