Develops global
conditions groups one
calibration/alignment new promotes quality
processing projects knowledge shortest
alignment DB monitoring tools TSG well
accessing aspects end expertise oversee i.e Ensures POGs various worked
AlCa calibration oversight users POG representatives
addition provides updates
AlCaDB cover defines deals optimizations operational services
CMS data databases offline service recognizes
CERN constants management
handle Oversees
database DPGs group objects

Overview of Database systems in CMS

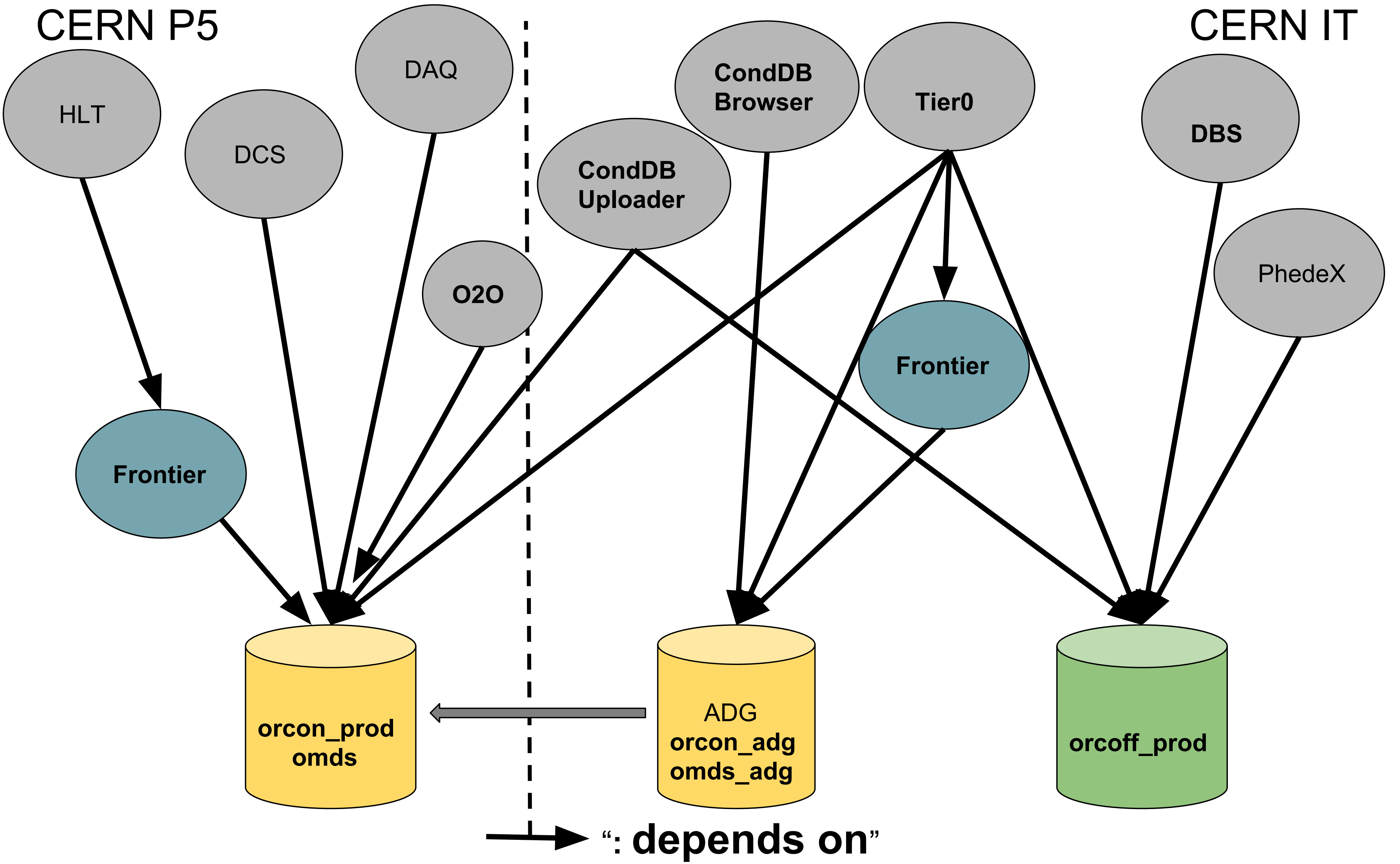
G. Govi (FNAL)
On behalf of the CMS experiment

Introduction

- The Database infrastructure plays a primary role in the CMS operations
  - Involved in all of the main production activities
  - Fairly complex architectures, with several dependent subsystems
  - Main backend choice is RDBMS/Oracle.
  - Went through several iterations of tuning over the past years

- About this overview
  - Does not cover the whole database usage in CMS
  - Focus is on the most critical systems for the database operation
  - Many other systems not mentioned are relying on Oracle
  - Notable cases of system adopting other types of RDBMS or No-SQL
    - iCMS recently moved to PostgreSQL
    - Prod Request Manager went to CouchDB

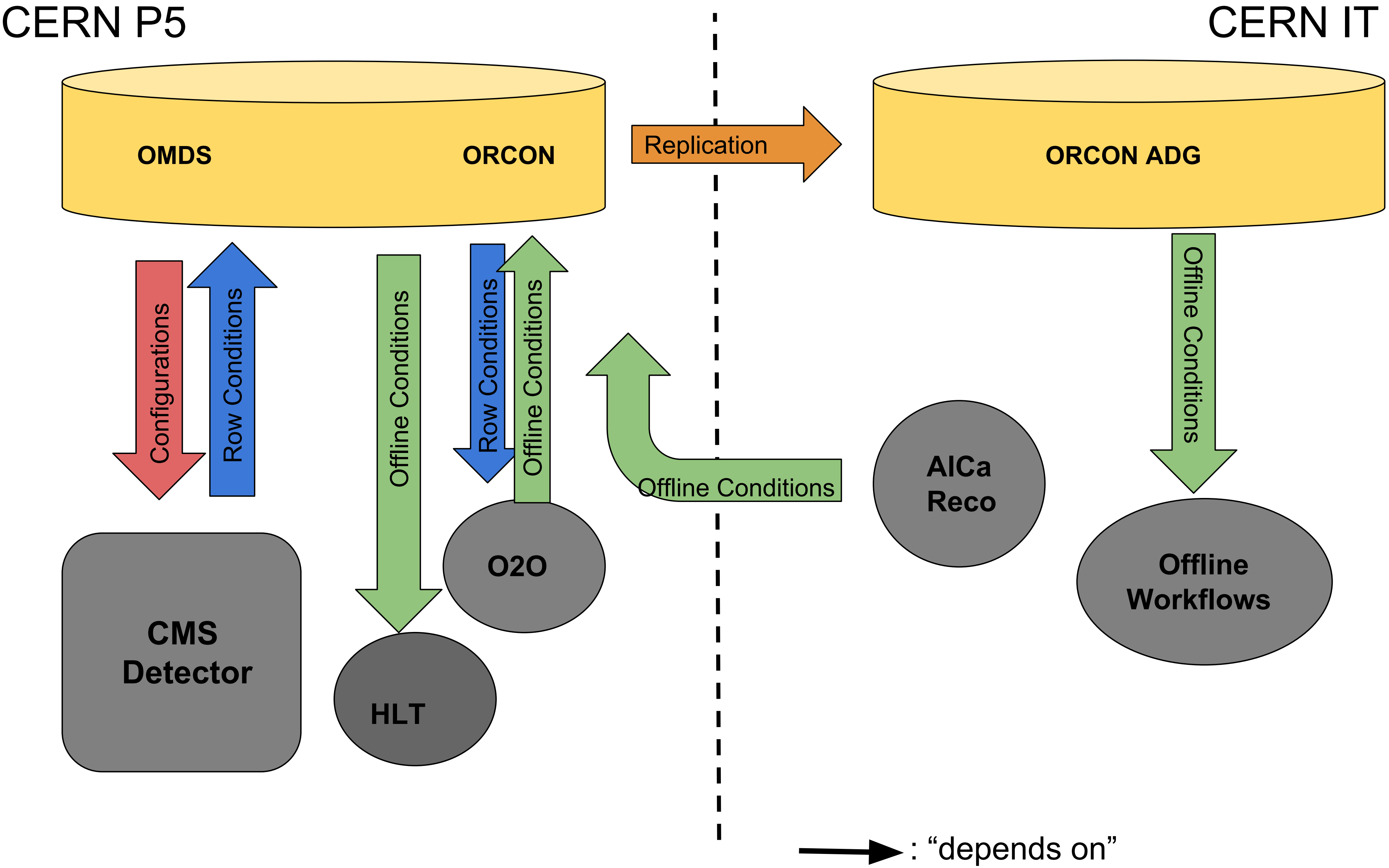
Oracle databases and applications



“: depends on”

2017/05/29

Online/Offline interplay - conditions



Online applications: DCS

- Conditions
  - Logging
- Configurations
  - Slow control parameters
- Schema based on PVSS
  - many tables, many relations
- One schema per system
  - detectors + various h/w
- Massive amount of data
  - keeping the history
  - need to save *on change*!
  - currently 4.6 TB (1.5 for the Tracker!)
- Requirements
  - relational consistency features
  - limited transactional activity
  - storage scalability
    - partitioning
    - disk size expanded at every warranty period

Online applications: DAQ

- Conditions/Configuration
  - Run Control
    - Java-based system
    - Simple schema with name-value pairs
  - Core DAQ
    - XDAQ software DB abstraction (XDATA)
    - Normalized schema mapping C++ structures
  - Large amount of data
    - 660 Gb Core
    - > 1Tb with various subsystems
- Requirements
    - Relational features
    - Transactions

Condition Database

- Condition data
  - In the “offline” format
  - consumed by HLT and offline workflows => critical for data taking and data processing
  - described by very simple data model
  - complexity of data structure hidden in the payload details
- Access patterns
  - Write once, never update, read many times
  - Multi-source writing
  - Payloads are immutable
- Use cases
  - Write from online processes
  - Write from manual updates by the detector experts
  - Read from thousands of jobs running at P5 and on the Grid
    - Caching provided by Frontier
- Size
  - 227 Gb at the moment, mostly for the payloads

Condition Database

- Condition metadata
  - Identify/label the data set (TAG, GLOBAL TAG)
  - Define the time validity (IOV)
  - Operation/Data management
    - strategical data
  - Logging
    - Historical data
  - Design exploits RDBMS features
    - Data integrity (relations)
    - Transactions (concurrent writing, consistency)

Condition Database: future directions

- Going towards a multi-tier model
  - Adding a dedicated service for the database access
  - Clients become agnostic about the storage details
  - Simplifies/relaxes the requirements about Transactions
  - Allows smooth back-end technology evolution
  - A joint project with other experiments is active in this direction
    - See presentations by A.Formica and P.Laycock

- The Relational Storage could be enhanced with file-system solutions
  - Specific subset can be entirely exported in sqlite-files (data and metadata)
    - supports the use cases of data preservation
  - Payload data exported into files in CVMFS
    - providing easier and faster access

Computing applications I

- Tier0
  - Depends on the 3 databases Orcon, Orcon ADG and Orcoff
  - Communication with Storage Manager
    - orcon - orcon ADG
    - Ensures the data transfer from P5 => critical!
    - Does not need to archive - mainly strategic
    - Still quite large: 600 Gb!
  - Bookkeeping
    - orcoff
    - Track the activity state: files, jobs and associated metadata
    - Drives the processing and data handling (software is stateless)
    - Highly transactional
    - Internal data mostly not persistent
    - Except Some monitoring data for studies: 14Gb

Computing applications II

- DBS
  - Dataset Bookkeeping System
    - Catalogue by production and analysis operation
    - Runs and Lumi granularity
    - Growing continuously (currently 2 Tb)
    - Require to support a relevant load of user-defined queries
- PHEDEX
  - Data replication/ File transfer
    - Tracking the state of the files, driving the operation
    - Highly transactional
    - Currently 290 Gb

Outlook

- No big revolution planned…
  - Designs and implementations have been mostly targeted to RDBMS
    - Most of the schema are highly relational, with several levels of complexity
    - Transactions and consistency check play a key role to drive key applications
  - Expertise in this area is well established
    - ensure the reliability of the implementations
    - easy the development/deployment cycle of ‘small ‘ application
  - The main systems are tightly bound to the Oracle choice
    - Good connection with the excellent IT/DB team
    - No other service (perceived) with production-level available at Cern
  - The availability of a support service is a key aspect
    - To help in the optimization and performance tuning
    - To maintain the backend infrastructure
    - To intervene in the emergencies
- Condition Database special case
  - The metadata handling fit very well in a RDBMS model
  - The Payload storage is “weakly” relational
  - Export of on-demand of payloads to file systems may improve the caching