



# A Conditions Data Management System for HEP Experiments

Paul Laycock, on behalf of  
the HSF Conditions DB WG

# Overview

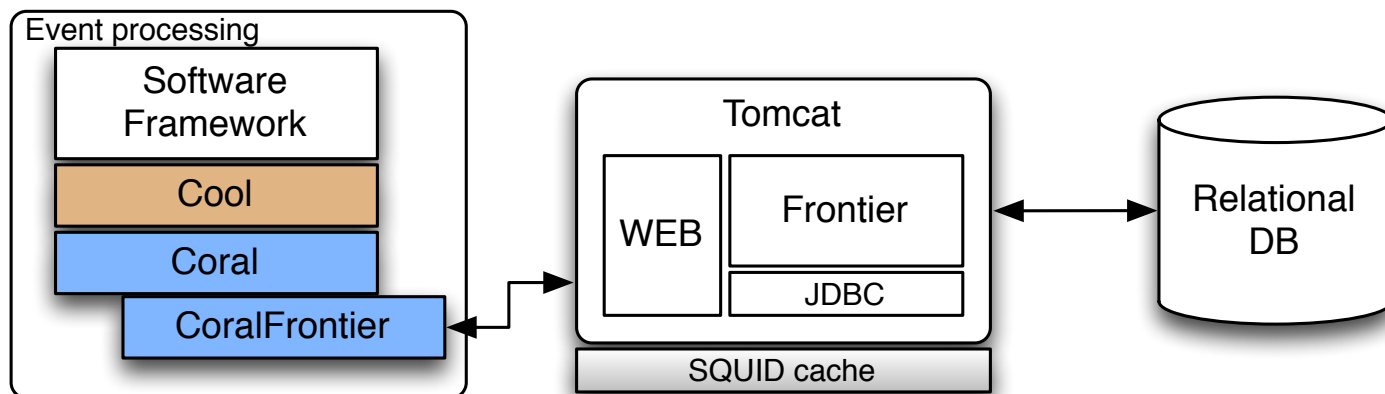
- Conditions DB infrastructure today
  - Mainly from the ATLAS/CMS perspective
- Convergence
  - Also for non-LHC experiments, in particular Belle II and NA62
- Summary of design principles
- Outlook

# Conditions DB infrastructure today

- A lot of good experience with COOL
  - A versatile tool supporting many workflows
- Nevertheless there are limitations
  - Caching is problematic
- Some common solutions
  - Frontier solved the caching problem
- Can we go further?

# Conditions DB infrastructure today

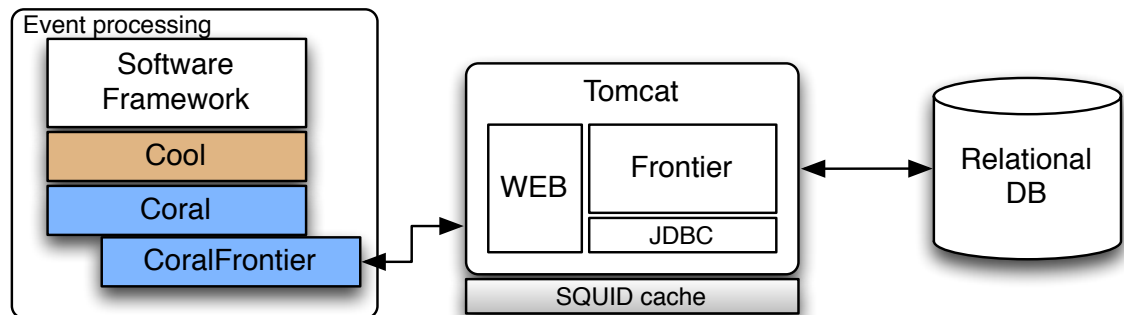
- Schematic of today's infrastructure, based on Cool on the client-side



- Frontier caching layer required

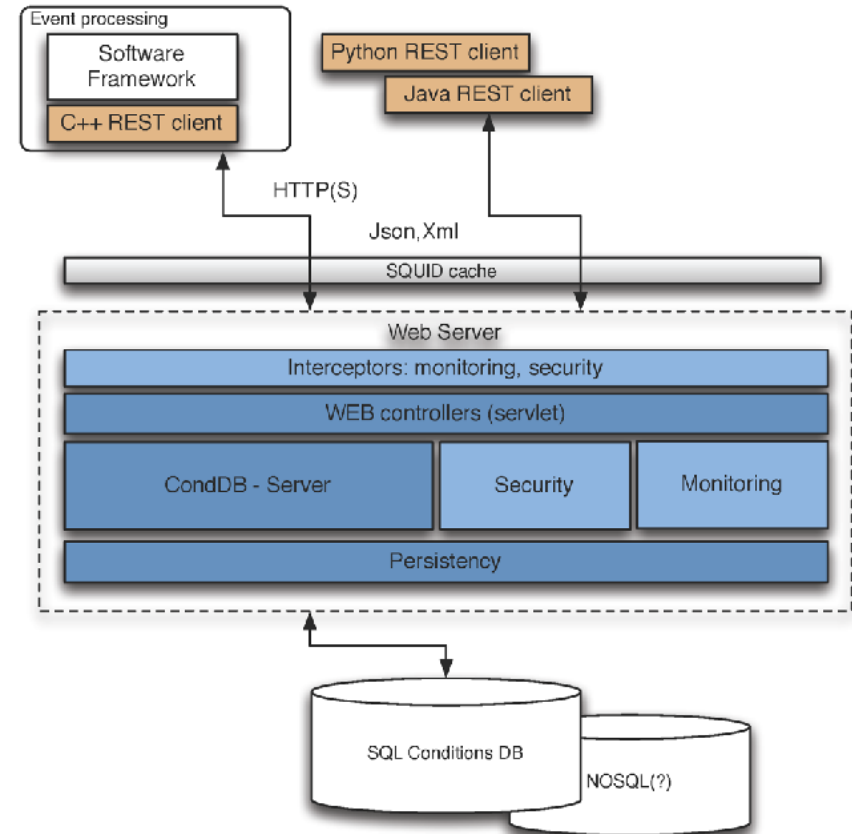
# Conditions DB infrastructure today

- Frontier model: Read-only, based on HTTP so allows squid-caching
- JDBC used to generate generic queries
- XML blobs sent back to the client, parsed by Cool-Coral-CoralFrontierPlugin



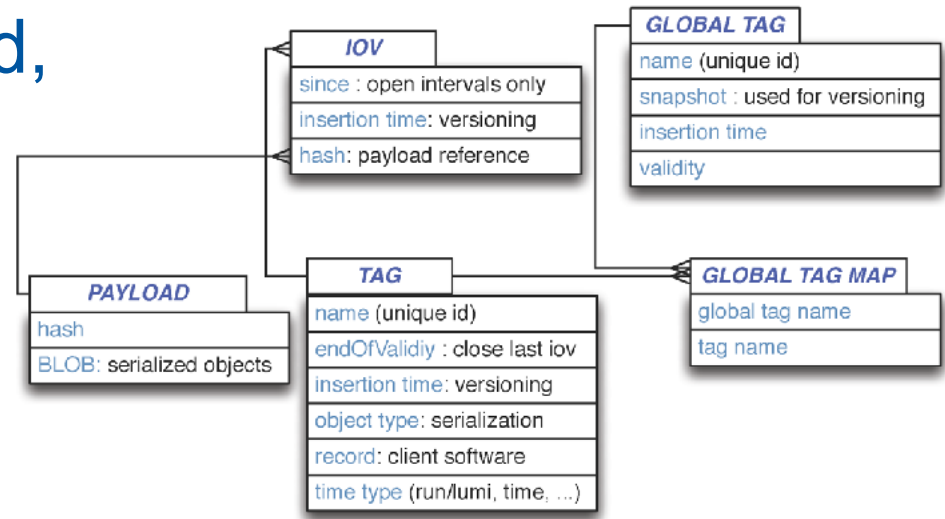
# Convergence

- REST communication between client and server
- Ensures simple interfaces
- Benefit from industry-standard solutions
- Loose coupling to allow component replacement



# Convergence

- Data Model: relational DB
- Single tables for payload, tags, IOVs
- IOVs and payloads resolved independently
- Cache-friendly design





# Convergence

- File-system approach is appealing, e.g. NA62 is migrating from files-in-the-release to a file-system approach:

/cvmfs/<experiment>/conditions/<gtag>/<system>/RunXYZ/payload

- Staged approach, first to a file-system, then to database (running experiment)
- Young experiments and e.g. HPC workflows, must keep cache-friendly feature
- Git as another alternative (c.f. LHCb)?

# Summary of design principles

- High degree of separation between client and server, client-side is simple but takes care of (de)serialisation
- Conditions accessed via a REST interface
- Caching must be built in, good experience using web proxy technologies. Clients should be able to deal with multiple proxies and servers
- Relational DB for data model is preferred
- File-system approach also interesting and can be very useful (though not generally considered to be the master payload storage)

# Outlook

- Conditions data infrastructure has been crucial to the successful running of the LHC experiments...
- ...and was also found to be a bottleneck
- Based on good collaborative experience starting with CMS and ATLAS, move towards CDB infrastructure that follows some good (!) design principles
- Caching built-in, and loosely coupled components to allow easier migration