

Future plans for CORAL and COOL

Andrea Valassi (IT-ES)

For the Persistency Framework team

Database Futures Workshop, 7th June 2011

- **Overview**
- **COOL**
- **CORAL**
- **CORAL Server**
- **Oracle client issues**
- **Conclusions**



- **CORAL**

- Abstraction of access to relational databases
 - Support for Oracle, MySQL, SQLite, FroNTier
 - *CORAL: generic interfaces (no schema/query design)*
- Used directly or indirectly via COOL/POOL
 - *COOL/POOL: specific applications (with schema/query design)*

- **COOL**

- Conditions database (relational)
 - Conditions object metadata (interval of validity, version)
 - Conditions object data payload (user-defined attributes)

- **POOL**

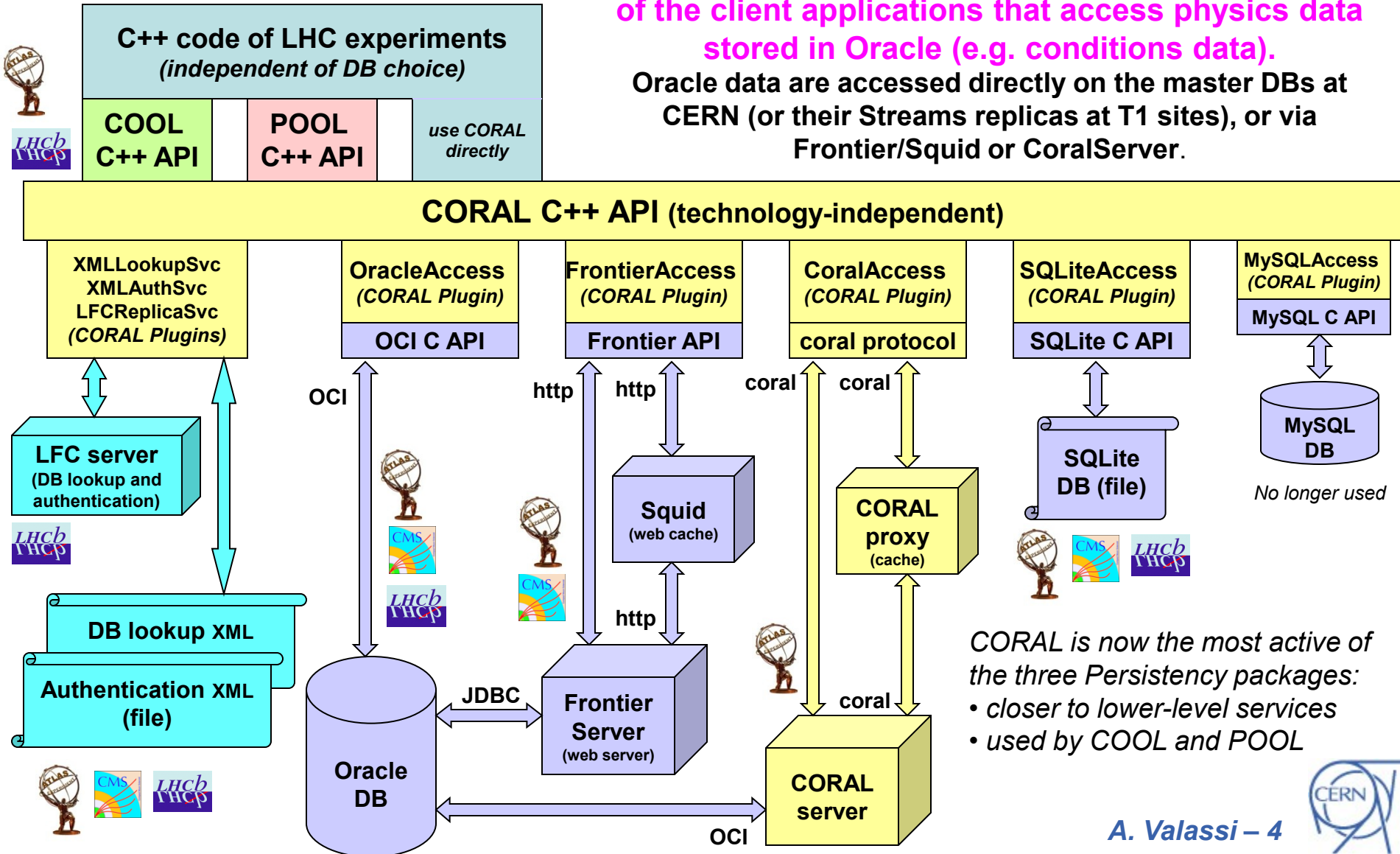
- Object streaming and metadata catalogs (ROOT/relational)
 - Event collections: tags database (ATLAS)
 - Object relational mapping (CMS, no longer used)
 - Event storage to ROOT files (not relevant for this workshop)



Overview of usage and architecture

CORAL is used in ATLAS, CMS and LHCb by many of the client applications that access physics data stored in Oracle (e.g. conditions data).

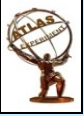


Oracle data are accessed directly on the master DBs at CERN (or their Streams replicas at T1 sites), or via Frontier/Squid or CoralServer.



CORAL is now the most active of the three Persistency packages:

- closer to lower-level services
- used by COOL and POOL

Details of CORAL, COOL, POOL usage

Persistency Framework in the LHC experiments	 ATLAS	 CMS	 LHCb
CORAL (Oracle, SQLite, XML authentication and lookup)	Conditions data (COOL) Geometry data (detector descr.) Trigger configuration data Event collections/tags (POOL)	Conditions data Geometry data (detector descr.) Trigger configuration data	Conditions data (COOL)
CORAL + Frontier (Frontier/Squid)	Conditions data (R/O access in Grid)	Conditions, Geometry, Trigger (R/O access in Grid, HLT, Tier0)	—
CORAL Server (CoralServer/CoralServerProxy)	Conditions, Geometry, Trigger (R/O access in HLT)	—	—
CORAL + LFC (LFC authentication and lookup)	—	—	Conditions data (authentication/lookup in Grid) <i>(only until 2010)</i>
COOL	Conditions data	—	Conditions data
POOL (relational storage service)	—	Conditions, Geometry, Trigger <i>(only until 2010)</i>	—
POOL (collections – ROOT/relational)	Event collections/tags	—	—

- CORAL, COOL and POOL are a joint development of IT-ES, ATLAS, CMS and LHCb
- For POOL: only included the components using relational databases, relevant for this works
- CORAL and COOL are also used by non-LHC experiments (Minerva at FNAL, NA62 at CERN)



- **Software maintenance**
 - Regular software releases of the stack (~one per month)
 - New platforms/compilers, new external software versions
 - Including selection/installation/patching of Oracle (OCI) client
 - Infrastructure: repository, build, test (e.g. SVN, cmake, valgrind...)
- **Operation and support**
 - User support and bug fixing
 - Debugging of complex service issues (~from a client perspective)
- **A few enhancements and new features**
 - See next slides for details on CORAL and COOL
 - About POOL: may transfer support to ATLAS if LHCb drops it



- **Performance optimizations and related new features**
 - e.g. speed up BLOB access through Python API (PyCool)
 - e.g. new API (and SQL queries) for fetching first/last IOV in a tag
 - All access (read/write) to the COOL database goes via the API
 - No plans for significant SQL/database performance optimizations
 - Major optimizations a few years ago (queries rewritten, indexes...)
 - Hints added (in dynamic query creation) to stabilize execution plans
- **New software features and database schema extensions**
 - e.g. COOL user control over CORAL sessions and transactions
 - e.g. better storage of 'vector' payload for IOVs and of DATE types
 - Some of these are already coded but need to be tested/released
- **In summary: fewer tasks planned for COOL than CORAL**



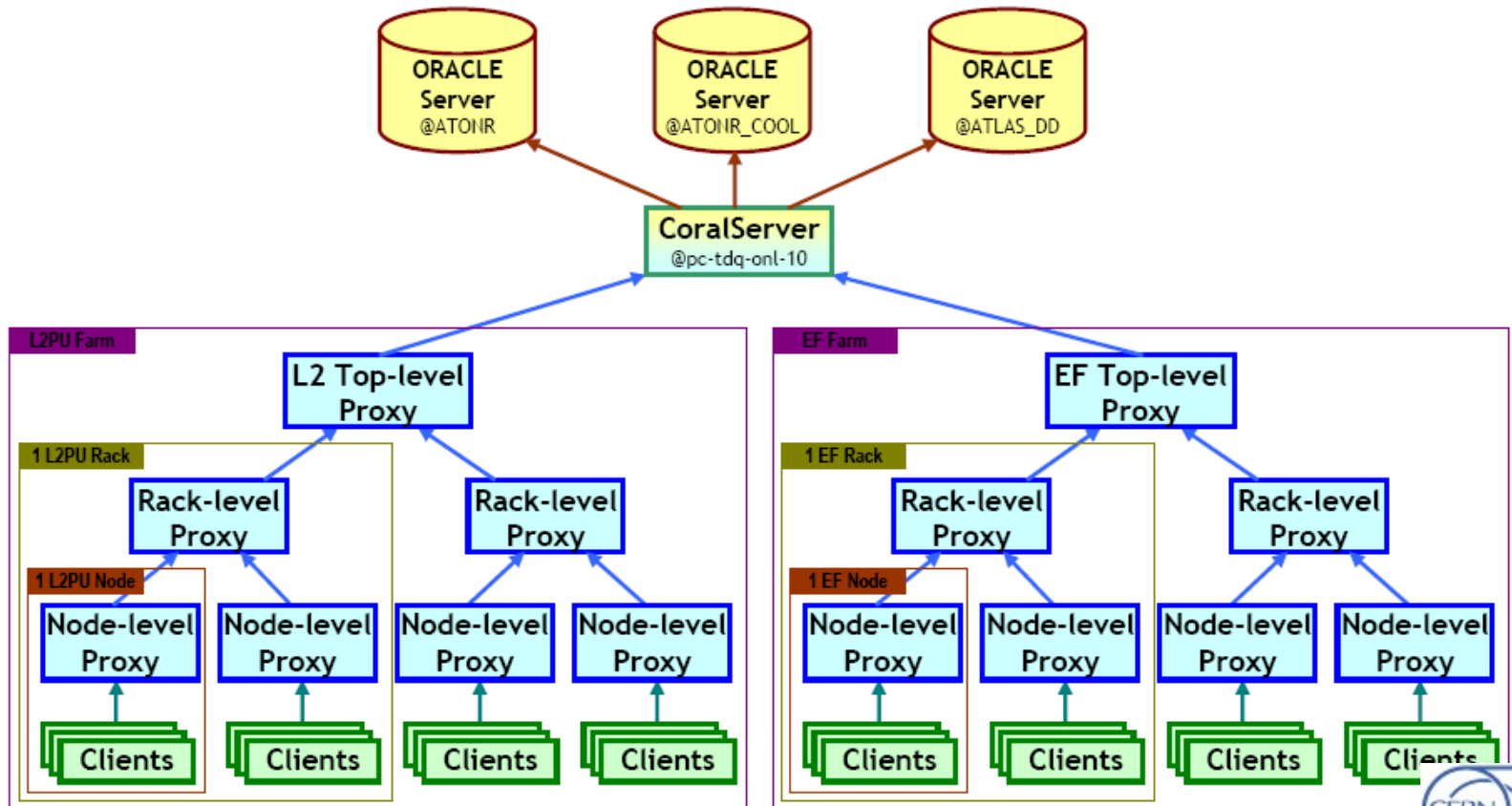
- **Reconnection after network/database glitches**
 - *Details in the next slide*
- **More robust tests (spot bugs before users report them)**
 - Was heavily relying on COOL test suite so far
- **Performance studies and optimizations**
 - e.g. reduce data dictionary queries in the Oracle plugin
 - compare Oracle, Frontier, CoralServer to find any other such issue
 - also related to CORAL handling of read-only transactions (serializable)
- **Later: enhance/redesign monitoring functionalities**
 - Interest by ATLAS online (CORAL server) and CMS too
 - In practice: need better code instrumentation for DB queries
 - See Cary Millsap's recommendations at his seminar last week!
- **In parallel: a few minor feature enhancements**
 - Support for sequences, FKs in SQLite, multi-schema queries...



- **Different issues reported by all experiments**
 - e.g. ORA-24327 “need explicit attach” in ATLAS/CMS ([bug #58522](#))
 - Fixed with a workaround in CORAL 2.3.13 (released in LCG 59b)
 - e.g. OracleAccess crash after losing session in LHCb ([bug #73334](#))
 - Fixed in current CORAL 2.3.16 candidate (see below)
 - Similar crashes can also be reproduced on all other plugins
- **Work in progress since a few months** (*A.Kalkhof, R.Trentadue, A.V.*)
 - Catalogued different scenarios and prepared tests for each of them
 - Prototyped implementation changes in ConnectionSvc and plugins
- **Current priority: fix crashes when using a stale session**
 - May be caused both by network glitch and user code ([bug #73834](#))!
 - A major internal reengineering of all plugins is needed (replace references to SessionProperties by shared pointers)
 - Done for OracleAccess ST in 2.3.16 candidate, pending for other plugins
 - The patch fixes single-thread issues; MT issues are still being analyzed
- **Next: address actual reconnections on network glitches**
 - e.g. non serializable R/O transaction: should reconnect and restart it
 - e.g. DDL not committed in update transaction: cannot do anything



- **CoralServer deployed for HLT in October 2009**
 - Smooth integration, used for LHC data taking since then
 - No problems except for handling of network glitches



- **Support usage in the ATLAS online system**
 - Requests and plans are in line with more general CORAL needs
 - Fix for the network glitch issue
 - CORAL monitoring of DB queries (both in server and proxies)
 - More detailed performance analysis and optimizations
- **Work on further extensions (e.g. for offline) is now frozen**
 - Interest from offline communities is limited or inexistent
 - Frontier is used for read-only use cases by both ATLAS and CMS
 - Also, likely synergy with CVMFS in the future for Squid deployment (http)
 - Possibly larger potential for extension than Frontier in other use cases, but no real need/request for these extended features
 - Authentication & authorization via X509 Grid proxy certificates
 - Already in CVS: will be released after cleanup of Globus integration
 - Database update functionalities (DDL/DML) – wont do
 - Disk resident cache (a la Squid) – wont do



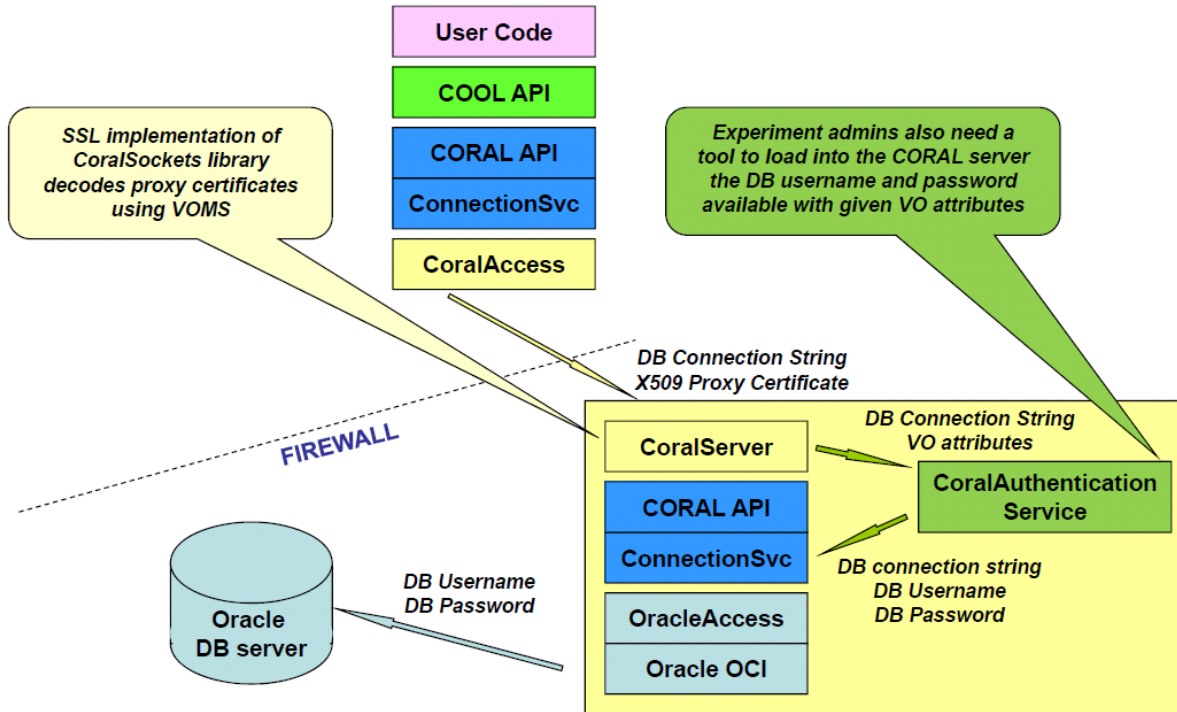
- **Oracle client for CORAL is maintained by the CORAL team**
 - Different installation (consistent with AA 'externals' on AFS)
 - Tailor-made contents
 - e.g. 11.2.0.1.0 patches to fix selinux and AMD quadcore bugs
 - e.g. sqlnet.ora customization for 11g ADR diagnostics
 - Close collaboration with Physics DB team in IT-DB on these issues
- **Two open issues in Oracle – plus a similar one in Globus**
 - All three are conflicts with the default Linux system libraries
 - Should either use the system libraries or use 'versioned symbols'
 - **Globus redefines gssapi symbols ([bug #70641](#))**
 - Suggested to use versioned symbols: will be in the 2011 EMI release
 - Workaround: disabled gssapi from Xerces (used by CORAL)
 - **Oracle client redefines gssapi symbols ([bug #71416](#))**
 - SR 3-1977807081 – gssapi in libclntsh.so conflicts with libgssapi_krb5.so
 - Suggested to use versioned symbols (Oracle bug 10184681)
 - No workaround needed so far (problem not yet seen in production...)
 - **Oracle client redefines kerberos symbols ([bug #76988](#))**
 - SR #3-3620145421 – krb5 symbols in libclntsh.so conflict with libkrb5.so
 - Suggested to use versioned symbols (Oracle bug 12557209)
 - Workaround: will customize kerberos parameters in sqlnet.ora

- **ATLAS, CMS and LHCb access Oracle via CORAL**
 - For conditions data (e.g. COOL in ATLAS/LHCb) and more
- **Support for many backends allows many options**
 - ATLAS switch to Frontier was painless from a software POV
- **Highest load is from software and service support**
 - Releases, Oracle client issues, debugging of complex service issues
 - A few new developments too, in parallel

*Thanks to the Physics DB team in IT-DB
and to the experiment users and DBA's
for their help and very good collaboration!*



Reserve slides



- **For comparison, if authentication uses the LFC replica service:**
 - Credentials are stored in LFC server (here: in Coral server)
 - Credentials are retrieved onto client by LFC plugin (here: stay in Coral server)
 - Credentials are sent directly by client to Oracle (here: sent by Coral server)
 - In both cases, credentials for Oracle authentication are username & password
 - No support of Oracle server for X509 proxy certificates
 - Could try using Kerberos authentication on Oracle server otherwise?