LCG Distributed Deployment of Databases
A Project Proposal

Dirk Düllmann

LCG PEB

20th July 2004

Why a LCG Database Deployment Project?

- What's missing?
  - LCG provides an infrastructure for distributed access to file based data and data replication
  - Physics applications (and Grid services) require a similar infrastructure for data stored in relational databases
    - Several applications and services already use RDBMS
    - Several sites have already experience in providing RDBMS services
  - Need for some standardisation as part of LCG
    - To allow applications to access data in a consistent, location independent way
    - To allow to connect existing db services via data replication mechanisms
    - To simplify a shared deployment and administration of this infrastructure during 24*7 operation
    - To increase the availability and scalability of the total LCG system
- Need to bring service providers (site technology experts) closer to database users/developers to define a LCG database service for the upcoming data challenges in 2005

Project Goals

- Define distributed database services and application access allowing LCG applications and services to find relevant database back-ends, authenticate and use the provided data in a location independent way.
- Help to avoid the costly parallel development of data distribution, backup and high availability mechanisms in each experiment or grid site in order to limit the support costs.
- Enable a distributed deployment of an LCG database infrastructure with a minimal number of LCG database administration personnel.

Project Non-Goals

- Store all database data
  - Experiments are free to deploy databases and replicate data under their responsibility
- Setup a single monolithic distributed database system
  - Given constraints like WAN connections one can not assume that a single synchronously updated database would work or give sufficient availability.
- Setup a single vendor system
  - Technology independence and multi-vendor implementation will be required to minimize the long term risks and to adapt to the different requirements/constraints on different tiers.
- Impose a CERN centric infrastructure to participating sites
  - CERN is one equal partner of other LCG sites on each tier
- Decide on an architecture, implementation, new services, policies
  - Produce a technical proposal for all of those to LCG PEB/GDB

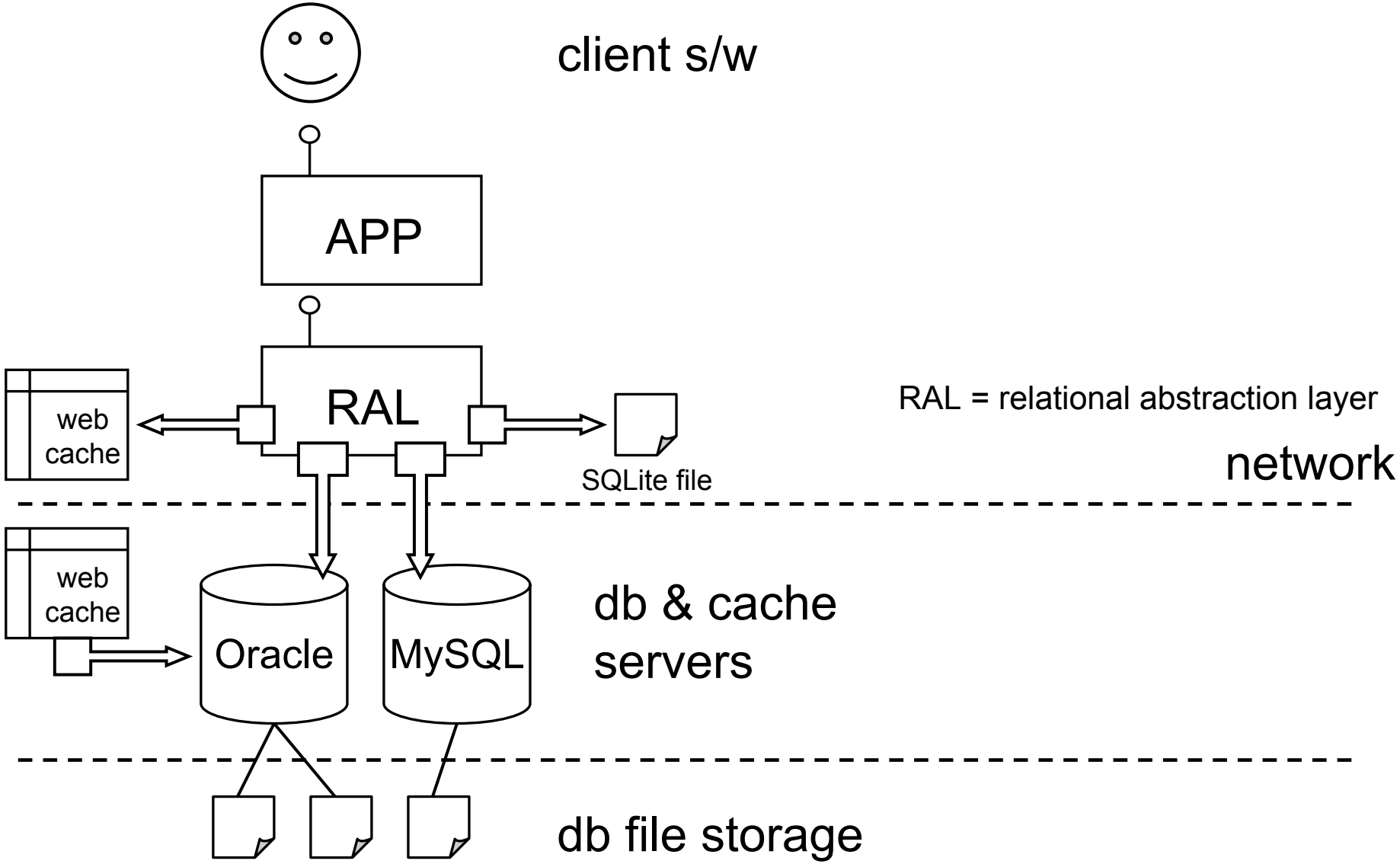
Database Service Situation at LCG Sites

- Several sites run Oracle based services for HEP and non-HEP applications
  - Deployment experience and procedures exists
  - … and can not be changed easily without affecting other site activities
- MySQL is very popular in the developer community
  - Used for some production purposes in LHC, though not at large scales.
  - Expected (just by developers?) to be easy to deploy down to T2 sites where db administration resources are very limited
  - So far no larger scale production service exists at LCG sites
    - but many applications which are bound to MySQL
- Expect a significant role for both database flavors
  - In different areas of the LCG infrastructure

Situation on the Application Side

- Databases are used by many applications in the physics production chain
  - Currently many of these applications are centralized ones
    - One (or more) copies of an application run against a single database
    - Many of these applications expect to move to a distributed model for scalability and availability reasons
    - This move can be simplified by an LCG database replication infrastructure - but this does not happen by magic
  - Selection of the db vendor is often made by application developers
    - Not necessarily yet with the full deployment environment in mind
- Need to continue to make central applications vendor neutral
  - DB abstraction layers exist or are being implemented in many foundation libraries
    - OGSA-DAI, ODBC, JDBC, ROOT, POOL, … are steps in this direction
    - Degree of the achieved abstraction varies
  - Still many applications which are only available for one vendor
    - Or have significant schema differences which forbid DB<->DB replications

Application s/w and Distribution Options



Distribution Options - and Impact on Deployment and Apps

- DB Vendor native replication
  - Requires same (or at least similar) schema for all applications running against replicas of the database
- Commercial heterogeneous database replication solutions
- Relational Abstraction based replication
  - Requires that applications are based on an agreed mapping between different back-ends
  - Possibly enforced by the abstraction layer
    - Otherwise by the application programmer
- Application level replication
  - Requires common API (or data exchange format) for different implementations of one application
    - Eg POOL File catalogs, ConditionsDB (MySQL/Oracle)
  - Free to choose backend database schema to exploit specific capabilities of a database vendor
    - Eg large table partitioning in the case of the Conditions Database

Candidate Distribution Technologies

Vendor native distribution

- Oracle replication and related technologies
  - Table-to-Table replication via asynchronous update streams
  - Transportable tablespaces
  - Little (but non-zero) impact on application design
  - Potentially extensible to other back-end database through API
  - Evaluations done at FNAL and CERN
- MySQL native replication
  - Little experience in HEP so far
    - ATLAS uses replicates databases to multiple sites, but replication is largely static and manual
  - Feasible (or necessary) in the WAN?

Local Database vs. Local Cache

- FNAL experiments deploy a combination of http based database access with web proxy caches close to the client
  - Significant performance gains
    - reduced real database access for largely read-only data
    - reduced transfer overhead than SOAP RPC based approaches
  - Web caches (eg squid) are much simpler to deploy than databases
    - could remove the need for a local database deployment on some tiers
    - no vendor specific database libraries on the client side
  - “Network friendly” tunneling of requests via a single port
- Expect caching technology to play a significant role especially towards the higher Tiers which may not have the resources to maintain a reliable database service

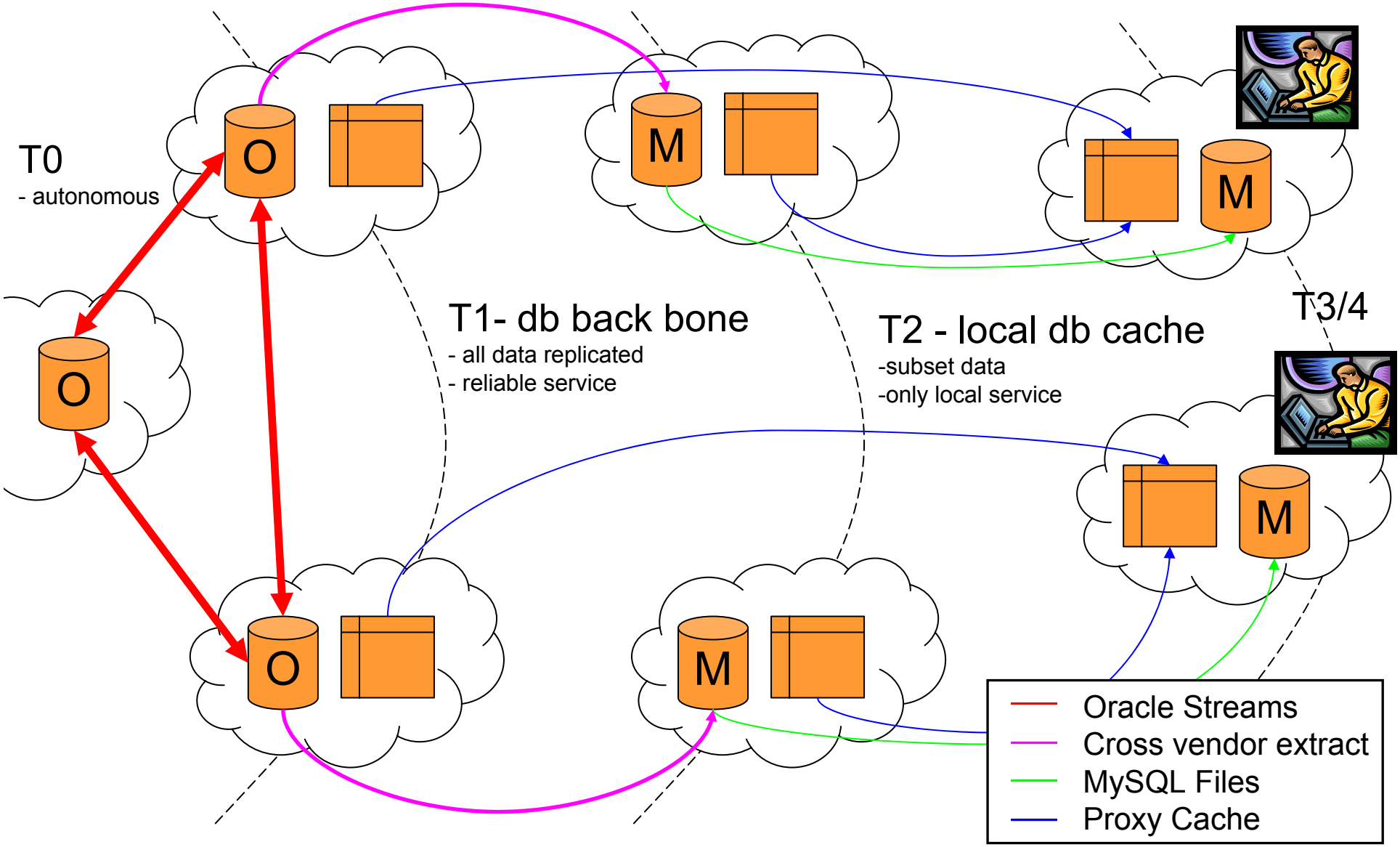
Tiers and Requirements

- Different requirements and service capabilities for different Tiers
  - Database Backbone (Tier1 <-> Tier1)
    - High volume, often complete replication of RDBMS data
    - Can expect good, but not continuous network connection to other T1 sites
    - Symmetric, possibly multi-master replication
    - Large scale central database service, local dba team
  - Tier1 <->Tier2
    - Medium volume, sliced extraction of data
    - Asymmetric, possibly only uni-directional replication
    - Part time administration (shared with fabric administration)
  - Tier1/2 <->Tier4 (Laptop extraction)
    - Support fully disconnected operation
    - Low volume, sliced extraction from T1/T2
- Need a catalog of implementation/distribution technologies
  - Each addressing parts of the problem
  - But all together forming a consistent distribution model

Where to start?

- Too many options and constraints to solve the complete problem at once
  - Need to start from a pragmatic model which can be implemented by 2005
  - Extend this model to make more applications distributable more freely over time
    - This is an experiment and LCG Application Area activity which is strongly coupled to the replication mechanisms provided by the deployment side
- No magic solution for 2005
  - Some applications will be still bound to a database vendor
    - and therefore LCG to a particular tier
  - Some site specific procedures will likely remain

Starting Point for a Service Architecture?



Staged Project Evolution

- Proposal Phase 1 (in place for 2005 data challenges)
  - Focus on T1 back-bone understand the bulk data transfer issues
    - Given the current service situation a T1 back-bone based on Oracle with streams based replication seems the most promising implementation
    - Start with T1 sites who have sufficient manpower to actively participate in the project
  - Prototype vendor independent T1<->T2 extraction based on application level or relational abstraction level
    - This would allow to run vendor dependent database applications on the T2 subset of the data
  - Define a MySQL service with interested T2 sites
    - Experiments should point out their MySQL service requirements to the sites
  - Start with T2 sites which are interested in providing a MySQL service and are able to actively contribute its definition
- Proposal Phase 2
  - Try to extend the heterogeneous T2 setup to T1 sites
    - By this time real MySQL based services should be established and reliable
    - Cross vendor replication based on either Oracle streams bridges or relational abstraction may have proven to work and to handle the data volumes

Proposed Project Structure

WP1 -Data Inventory and Distribution Requirements

- Members are s/w providers from experiments and grid services based on RDBMS data
- Gather data properties (volume, ownership) requirements and integrate the provided service into their software

WP2 - Database Service Definition and Implementation

- Members are site technology and deployment experts
- Propose an agreeable deployment setup and common deployment procedures

WP3 - Evaluation Tasks

- Short, well defined technology evaluations against the requirements delivered by wp1
- Evaluation are proposed by people WP2 (evaluation plan) and typically executed by the people proposing a technology for the service implementation and result in a short evaluation report

Data Inventory

- Collect and maintain a catalog of main RDBMS data types
  - Select from catalog of well defined replication options
    - which can be supported as part of the service
  - Conditions and Collection/Bookkeeping data are likely candidates
- Ask the experiments and s/w providers to fill a simple table for each main data type which is candidate for storage and replication via this service
  - Basic storage properties
    - Data description, expected volume on T0/1/2 in 2005 (and evolution)
    - Ownership model: read-only, single user update, single site update, concurrent update
  - Replication/Caching properties
    - Replication model: site local, all t1, sliced t1, all t2, sliced t2 …
    - Consistency/Latency: how quickly do changes need to reach other sites/tiers
    - Application constraints: DB vendor and DB version constraints
  - Reliability and Availability requirements
    - Essential for whole grid operation, for site operation, for experiment production,
    - Backup and Recovery policy
      - Acceptable time to recover, location of backup(s)

DB Service Definition and Implementation

- Service Discovery
  - How does a job find a replica of the database it needs?
    - Transparent relocation of services?
- Connectivity, firewalls and constraints on outgoing connections
- Authentication and authorization
  - Integration between DB vendor and LCG security models
- Installation and configuration
  - Database server and client installation kits
    - Which client bindings are required?
      - C, C++, Java(JDBC), Perl, ..
  - Server administration procedures and tools
    - Monitoring and statistics gathering
    - Basic agreements to simplify the distributed operation
  - Server and client version upgrades (eg security patches)
    - How, if transparency is required for high availability?
- Backup and recovery
  - Backup policy templates, responsible site(s) for a particular data type
  - Acceptable ▪ latency for recovery

Initial list of possible evaluation tasks

- Oracle replication study
  - Eg Continue/extend work started during CMS DC04
  - Focus: stability, data rates, conflict handling
- DB File based distribution
  - Eg shipping complete MySQL DBs or Oracle tablespaces
  - Focus: deployment impact on existing applications
- Application specific cross vendor extraction
  - Eg Extracting a subset of Conditions Data to a T2 site
  - Focus: complete support of experiment computing model use cases
- Web Proxy based data distribution
  - Eg Integrate this technology into relational abstraction layer
  - Focus: cache control, efficient data transfer
- Other Generic Vendor-to-Vendor bridges
  - Eg Streams interface to MySQL
  - Focus: feasibility, fault tolerance, application impact

Proposed Mandate, Timescale & Deliverables

- Define in collaboration with the experiments and Tier0-2 service providers an reliable LCG infrastructure which allows to store the database data and distribute it (if necessary) for use from physics applications and grid services. The target delivery date for a first service should be in time for the 2005 data challenges.
- The project should run as part of the LCG deployment area in close collaboration with the application area as provider of application requirements and db abstraction solutions.
- Main deliverables should be
  - An inventory of data types and their properties (incl. distribution)
  - A service definition document to be agreed between experiments and LCG sites
    - Including required s/w installations to access to LCG db services and to distribute the data
  - A service implementation document to be agreed between LCG sites
- Status reports to the established LCG committees
  - Final decisions are obtained via PEB and GDB

Summary

- Sufficient agreement about necessity of a db deployment service in ATLAS, CMS and LHCb and indirectly (via gLite) from ALICE
  - Propose to start database deployment project now as part of the LCG deployment area with strong application area and Tier 0/1/2 site involvement
- First step would be to nominate participants and setup regular meetings
  - Two participants who represent the experiment s/w development and deployment activities in data inventory discussions
    - Participants from major grid services (LCG RLS, EGEE)
  - Participants from T1 & T2 sites which can put in technology expertise and some effort to participate in technology evaluations
  - Both will require significant work (>0.3 FTE)
- Present a first data inventory and a more detailed work plan at a GDB/PEB 2 month after project start
  - Validate proposed distribution model against requirements from data inventory
  - Confirm feasibility of model components in evaluation tasks with sites
  - Understand model consequences with developers of key applications