

CORAL Server Development Status

Zsolt Molnár - CERN-IT-DM
DM Meeting,
14 October, 2008

Zsolt.Molnar@cern.ch

CORAL

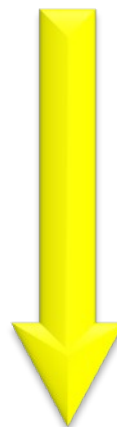
```
// stop all pools,  
for(tp = m...  
if(tp->second-  
busyTPools.p  
  
// Reap child pr  
pid_t pid;  
while ((pid = w  
if(!beGraceful)  
// on a SIGINT  
return; it  
}  
  
// now loop wait  
while(busyTPool  
sleep(1); // S  
for(unsigned i  
if(busyTPools  
// it's file no  
busyTPools.  
  
else
```

Data Management

Physics applications
common or experiment specific



COOL
validity
intervals
conditions data



POOL
object storage
metadata
collections, file catalogs

CORAL
RDBMS access, indirection,
authentication/authorization



Computing Services
files, databases, grid services



- **Access and security problems**
 - Oracle does not support grid authentication
 - Exposed security holes
- **Software licensing problems**
 - CORAL clients link against DB access libraries
 - Configuration management, export limitations
- **Performance problems**
 - Oracle spawns many idle processes
 - Waste of CPU and server resources

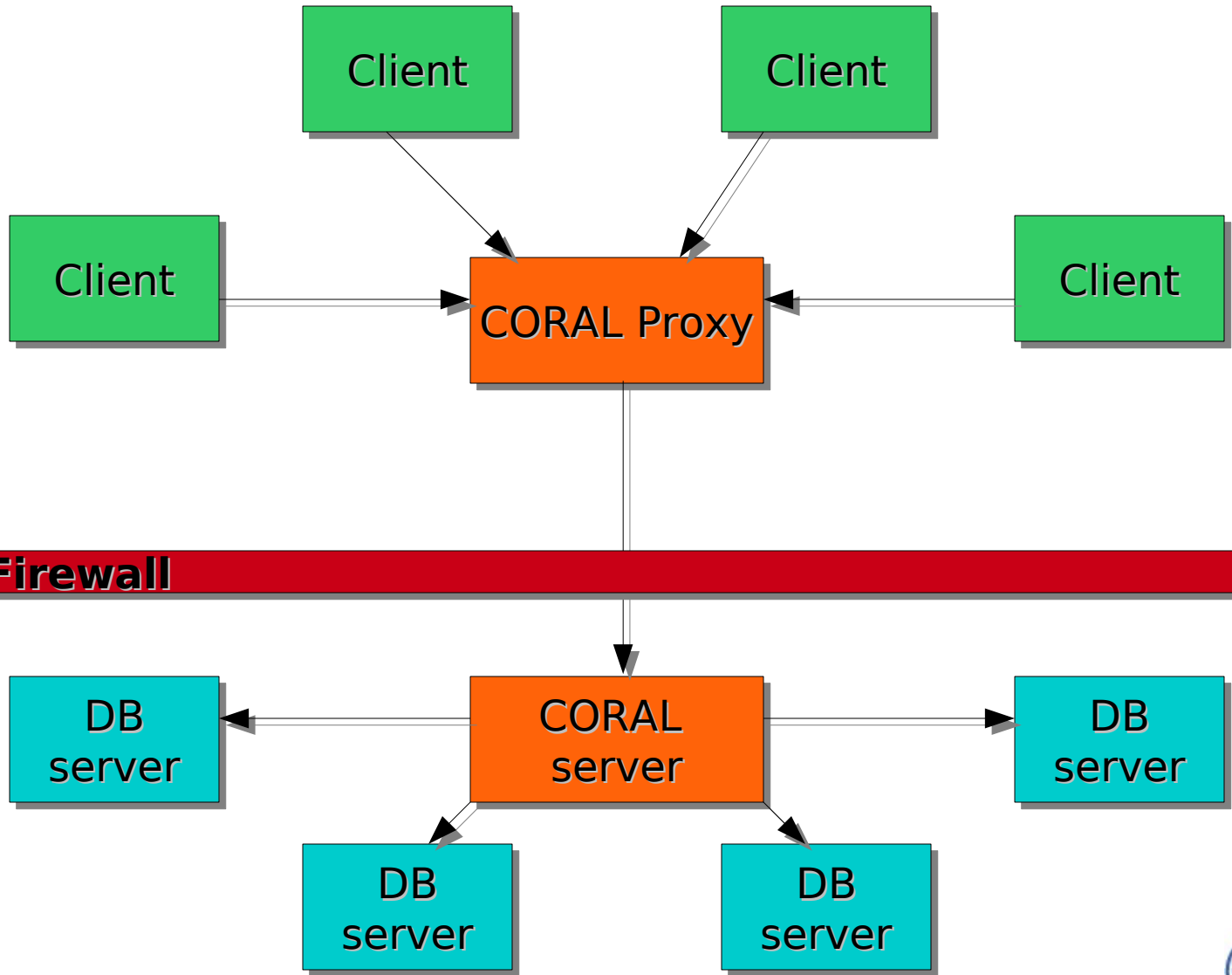


Data
Management

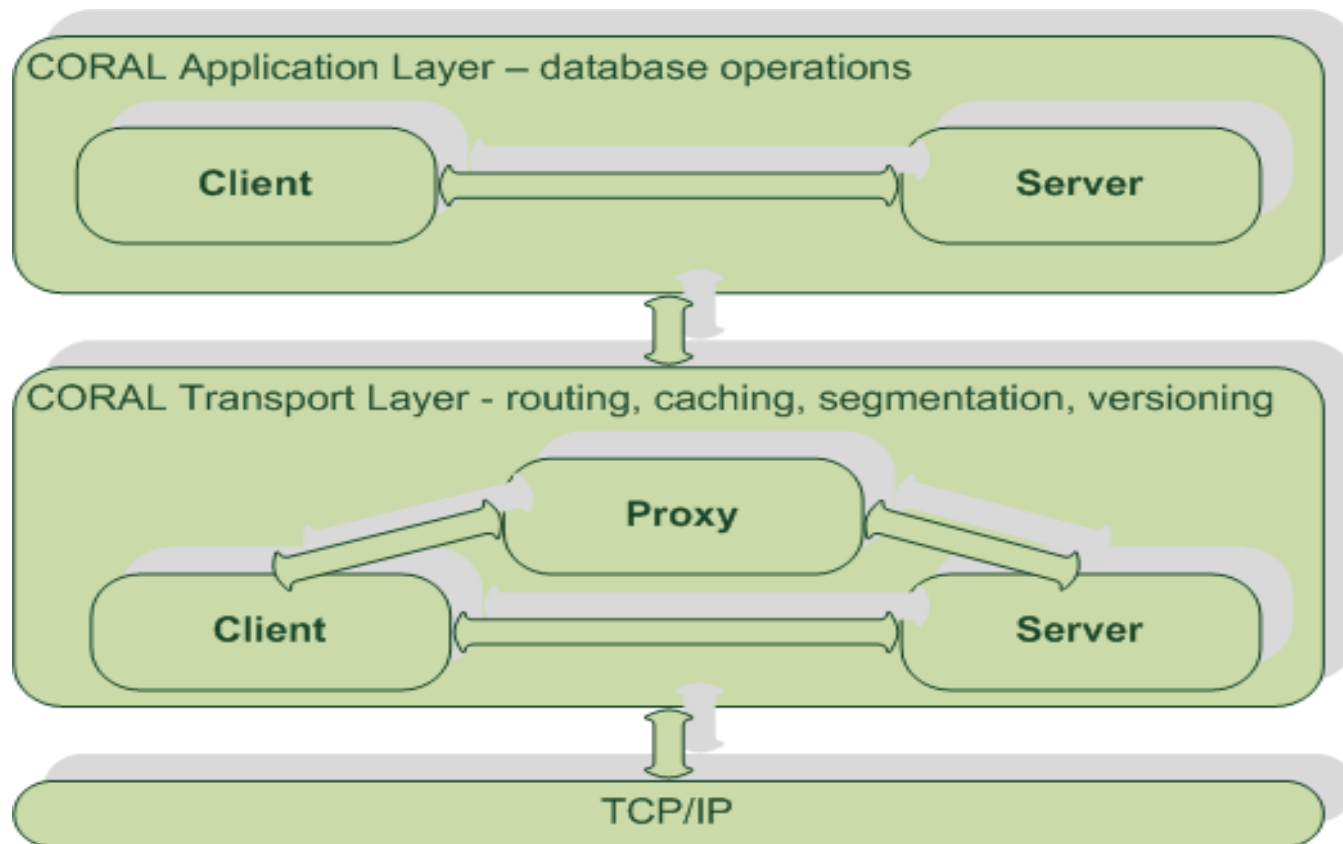
Solutions provided by CORAL Server

- **Access and security**
 - Database servers are not visible from Internet
 - CORAL server is the gateway, handling GSI
- **Software licensing**
 - One single plugin for all the database backends
 - Pure CORAL code
 - Transfers general representation of the relational operations through the network
- **Performance**
 - A network of caching proxy servers between the clients and the CORAL server
 - Developed at SLAC

Deployment example



Layered communication protocol



```
// stop all pools,  
for(tp = m...  
if(tp->second-  
busyTPools.p
```

```
// Reap child pr  
pid_t pid;  
while ((pid = w  
if(!beGraceful)  
// on a SIGINT  
return; it
```

```
// now loop wait  
while(busyTPool  
sleep(1); //  
for(unsigned i  
if(busyTPools  
// it's file no  
busyTPools
```

else
Data Management

- **Binary protocol**
- **Support of different architectures, translation**
- **Lead by cache-ability**
 - Responses are hash-ed by the corresponding requests
 - Determine semantically identical requests
 - Those lead to identical hash keys
 - Different architectures and clients should generate the same key
 - Open for future optimization
 - Did we achieve the above goals?
 - Compressing? Crypt-ing?

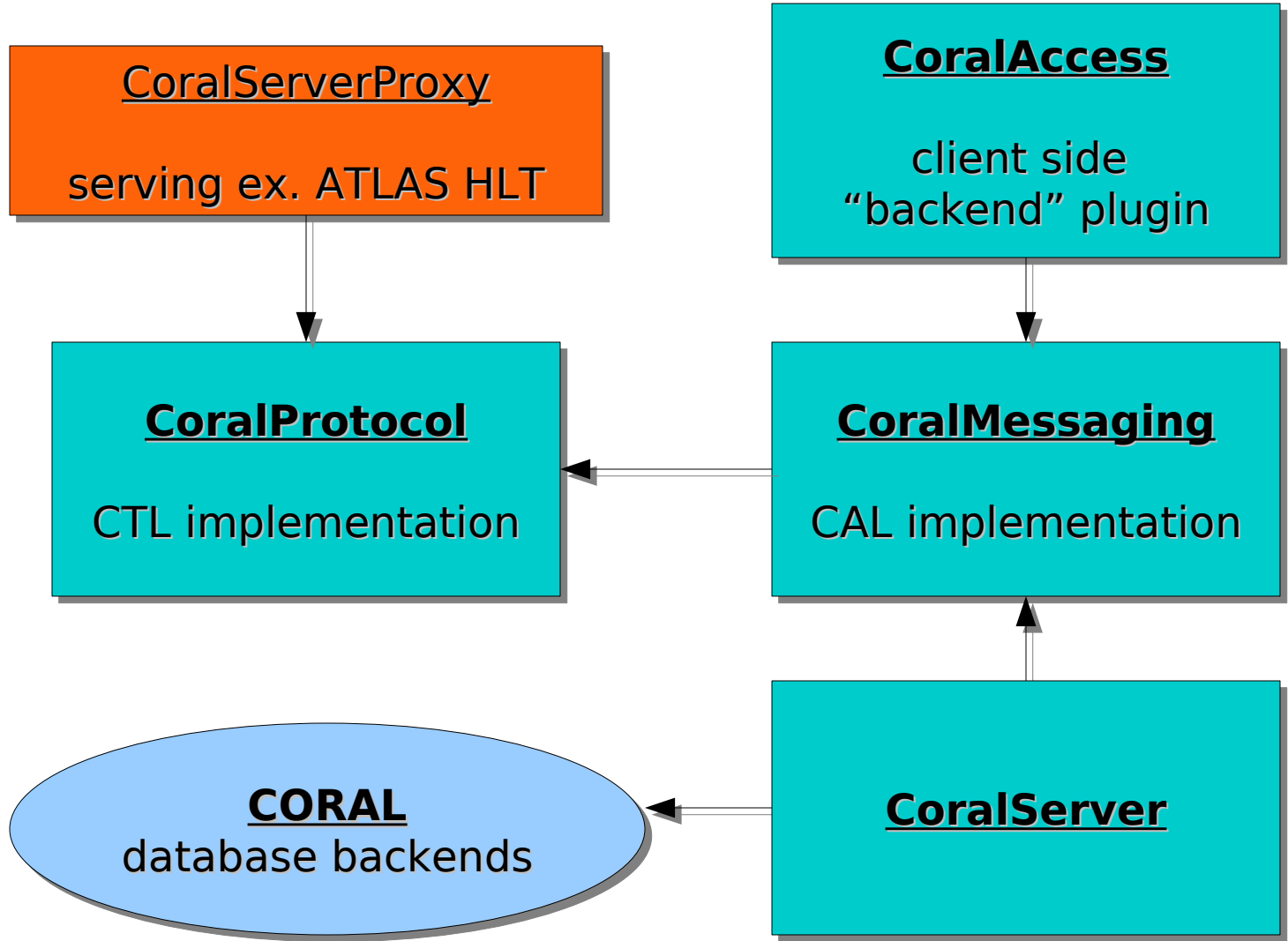


Communication protocol

- Joint development with SLAC
- CTL based on ATLAS DbProxy
- Multiple protocol version on server side
- Single protocol version on client side
- Updating operations are not cached
- Cache-ability can be controlled
- Based on message exchanges
 - Messages may be segmented
 - We call them packets...
 - The segments are individually hash-ed

Data
Management

New CORAL packages



The Server

- **Multi-threaded server**
- **Message handler plugins according to protocol version**
 - Determined by packet header
- **Multiplexing database sessions and transaction**
 - The proxy is a single client
- **Almost stateless**
 - Only the transaction state is shared between message handlers
- **The backend is encoded in the “schema”**
 - CORAL connection string:
oracle://oracleserver:port/schema
 - CoralAccess connection string:
coral://coralserver:port|oracle://oracleserver:port/schema



Development, deployment

- **SEAL free support only**
 - But temporary back-ports to SEAL-ed CORAL
 - Extra burden when releasing engineering versions
- **Client: planned support for all AA platforms**
- **Server: Linux/OSX support only**
- **~15000 lines of code**
 - With using extensive meta-programming techniques
- **Tests: CORAL Integration tests, ATLAS HLT tests, COOL test suite**
 - Still problems

Data
Management

Schedule, resources

- **December, 2008: first official release**
 - Part of CORAL 2.1.0
 - Read-only functionality only
 - Should already be used in data taking
- **Update, security functionality after that**
 - Security: ~February, 2009 ?
 - Update: ~April 2009 ?
- ***<http://pool.cern.ch/coral>***

Data
Management

People

- Contributing to CORAL: *Z. Xie* (CERN)
- Contributing to CORAL Server: *A. Kalkhof* (CERN)
- Contributing to planning, organizing: *D. Duellmann, A. Valassi* (CERN)
- CoralProxyServer: *R. Bartoldu, A. Salnikov* (SLAC)

Data
Management