ATLAS Online Software

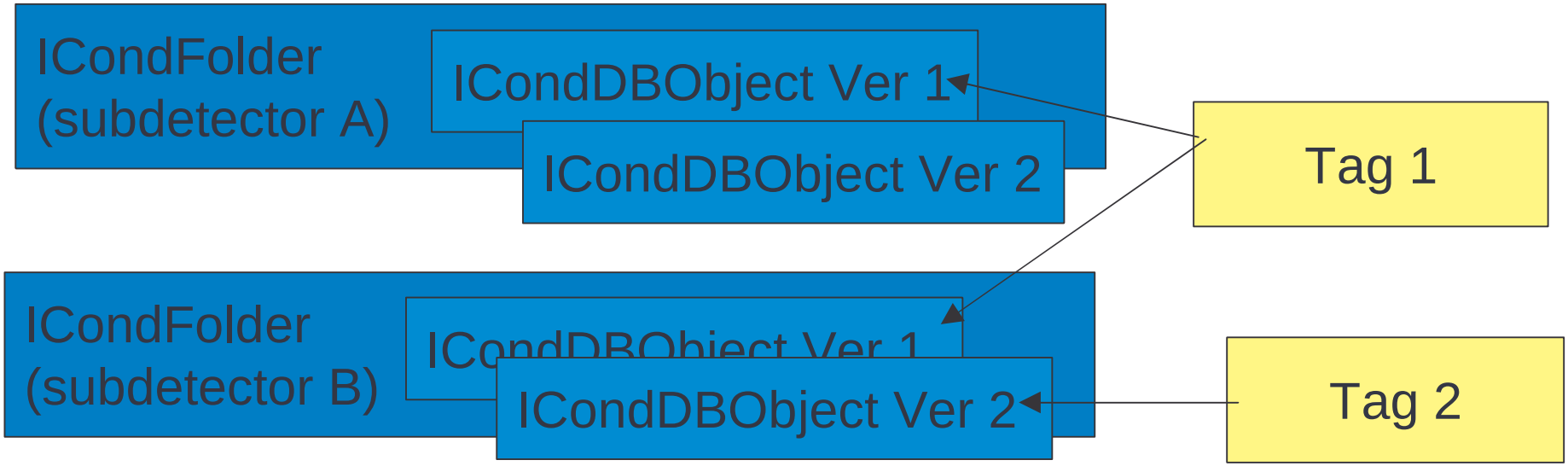
ConditionsDB MySQL Backend Implementation

April, 23, 2002

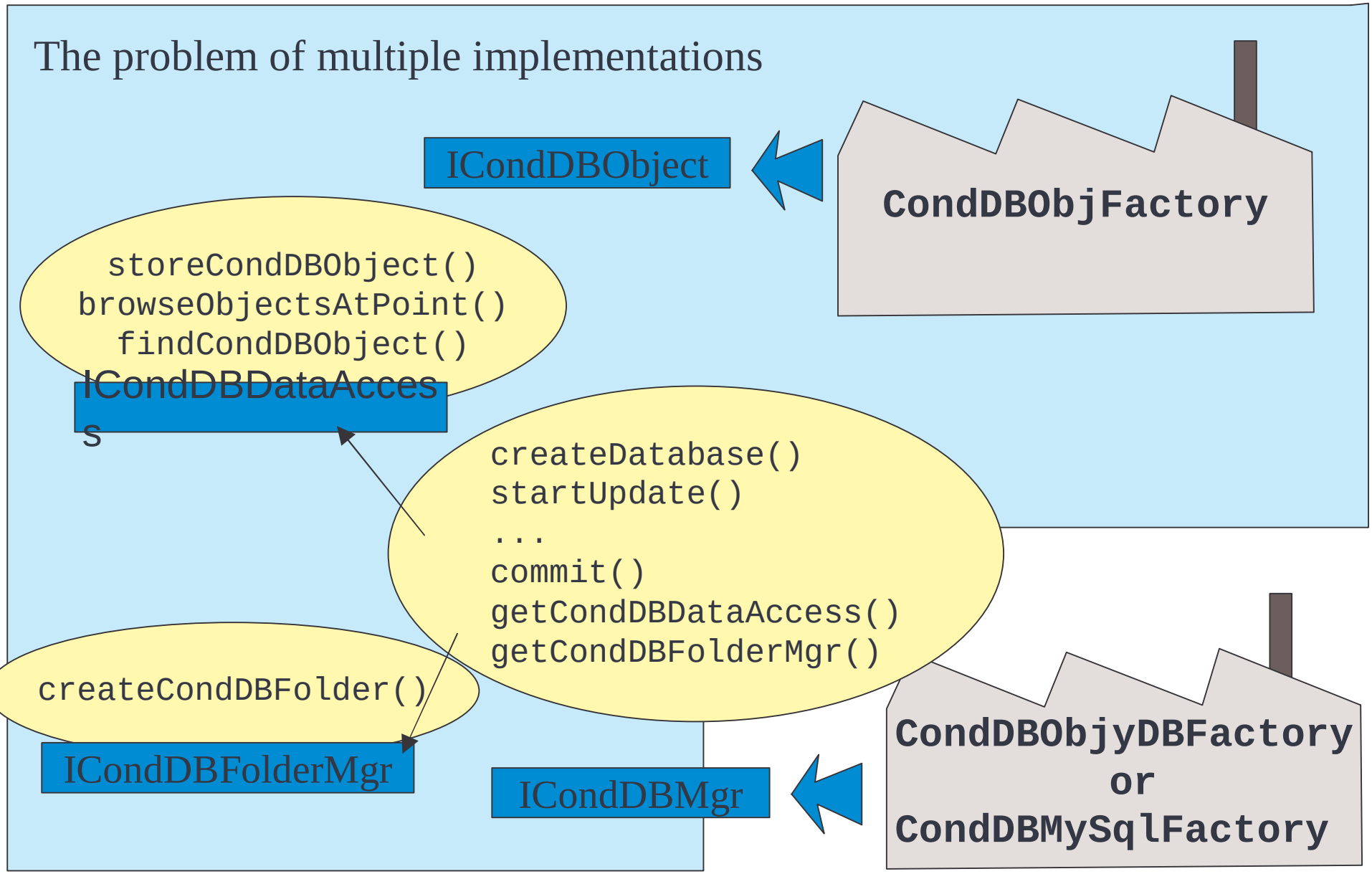
Jorge Lima – FCUL (Jorge.Lima@fc.ul.pt)

ConditionsDB terminology

- The Folders and FolderSets are organised in a filesystem like hierarchy.
- FolderSet is analogous to a directory.
- Folder is analogous to a file.



ConditionsDB API

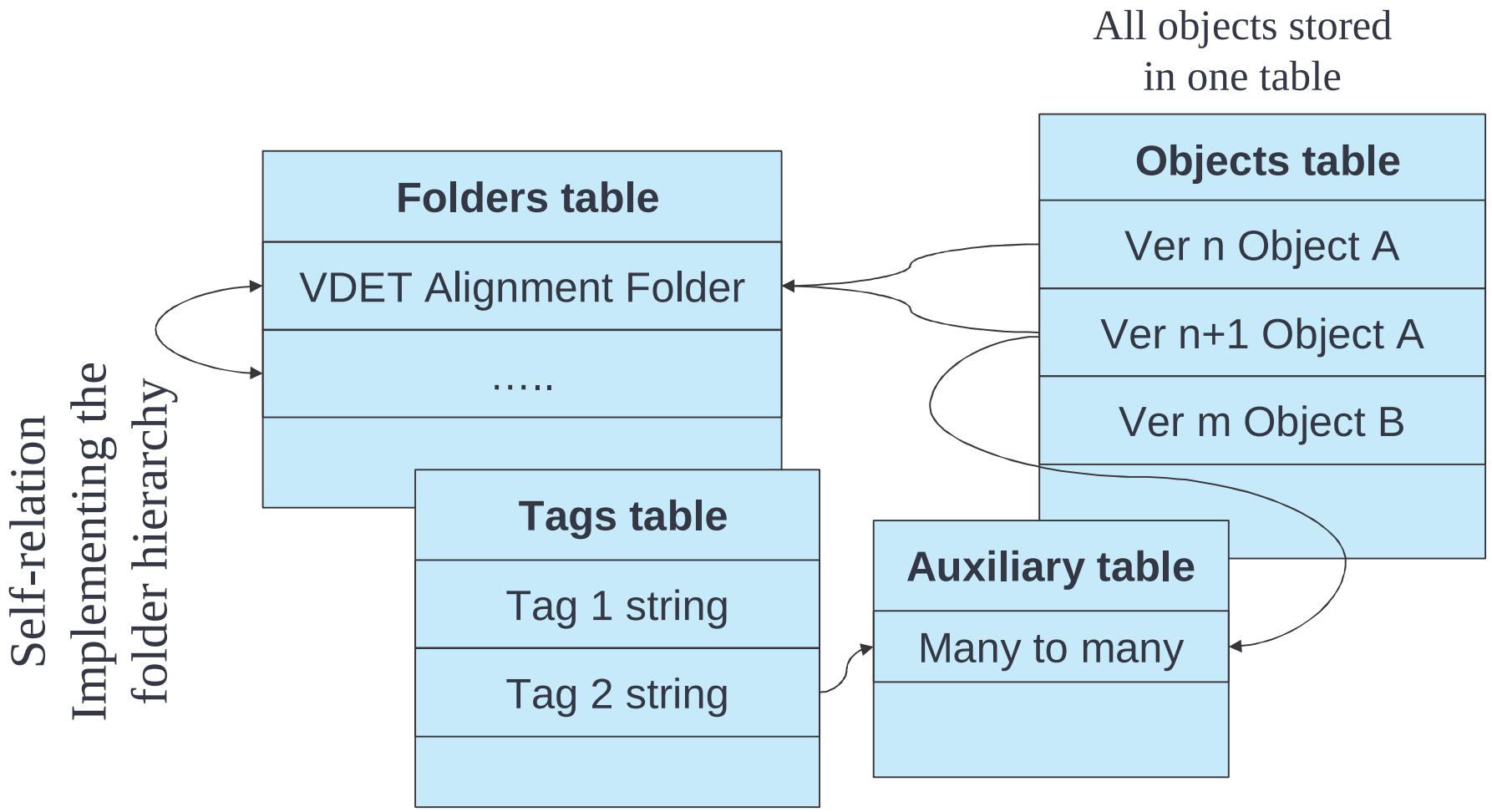


Code example (DBMS independence)

```
#include <ConditionsDB/CondDBMySQLMgrFactory.h>

...

    ICondDBMgr* CondDBmgr =
    CondDBMySQLMgrFactory::createCondDBMgr();

    ...
    CondDBmgr->init();
    ...
    CondDBmgr->startUpdate();
    CondDBmgr->createCondDB();
    CondDBmgr->commit();
    ...

    CondDBMySQLMgrFactory::destroyCondDBMgr( CondDBmgr );
```

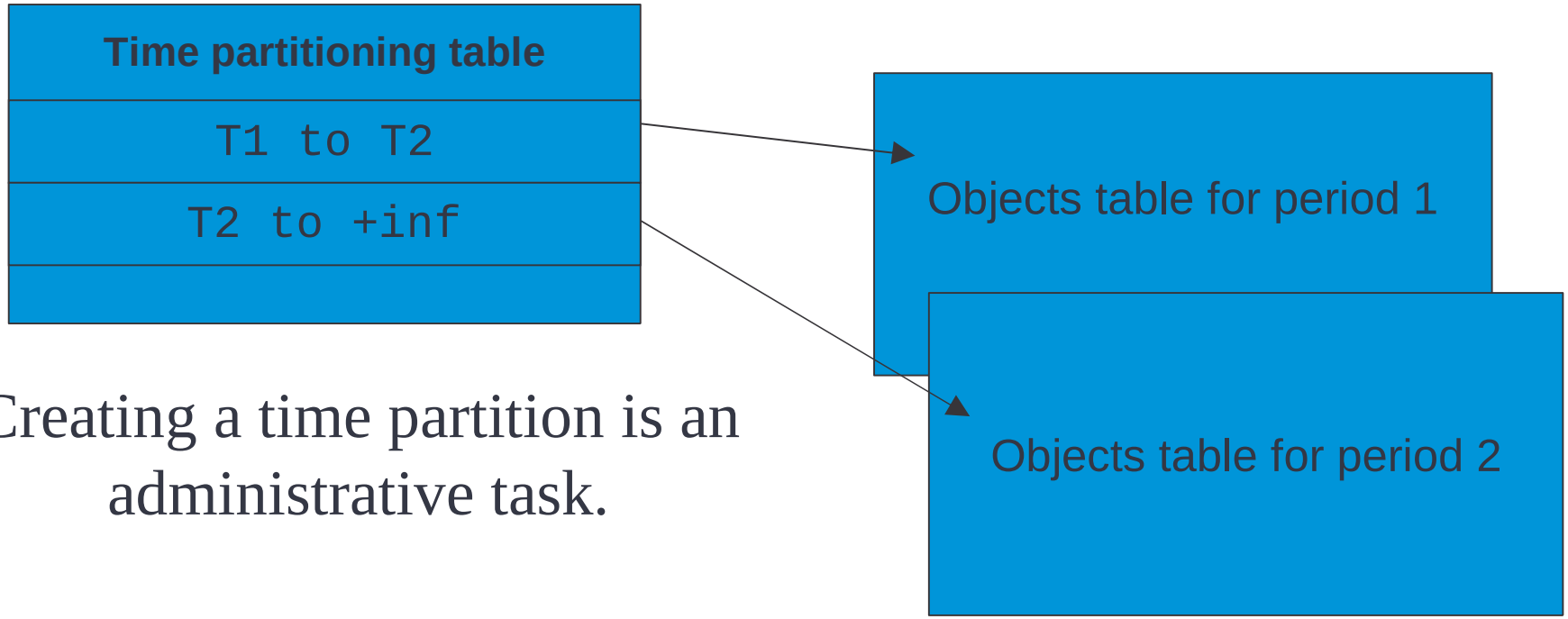
Relational DB model

- Simplified tables relationship



Relational DB model

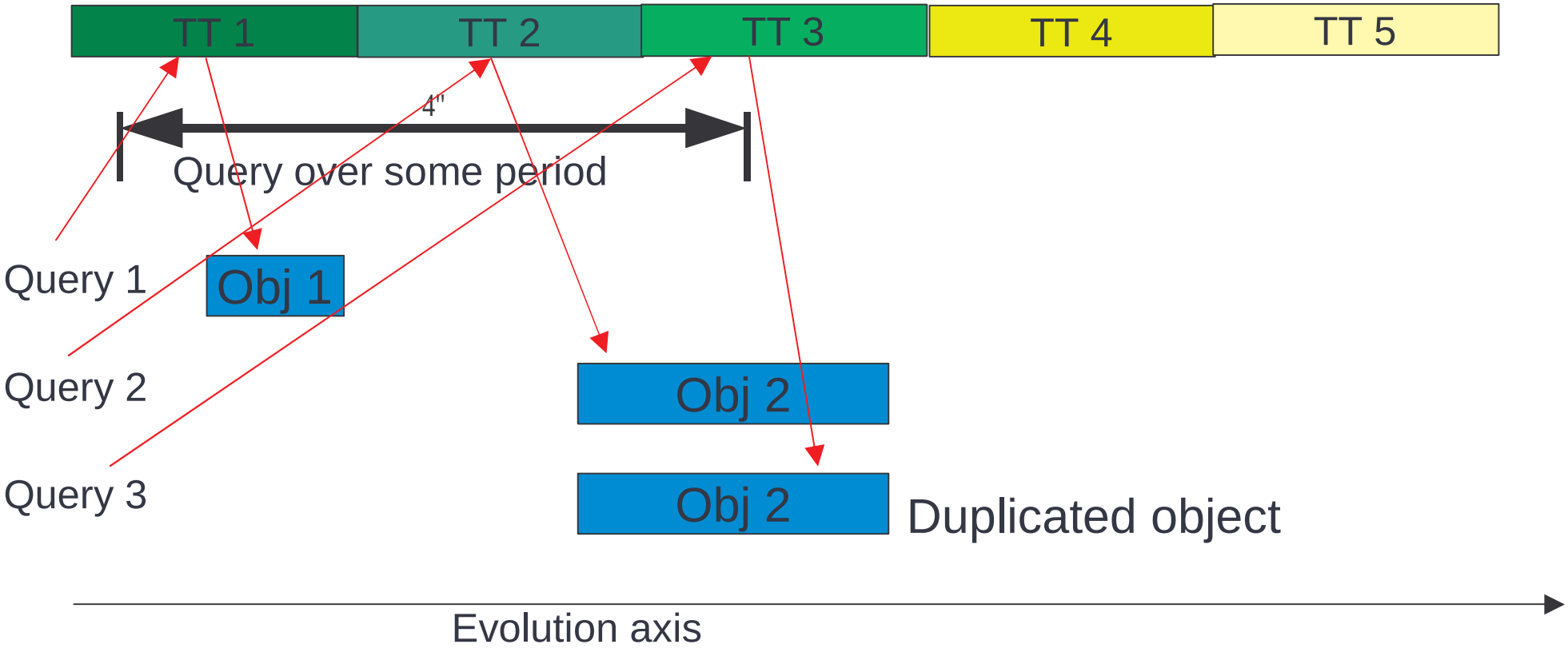
- No clustering on category ?
- Clustering over large time intervals.
- Scalability allowing different databases and database servers.



Creating a time partition is an administrative task.

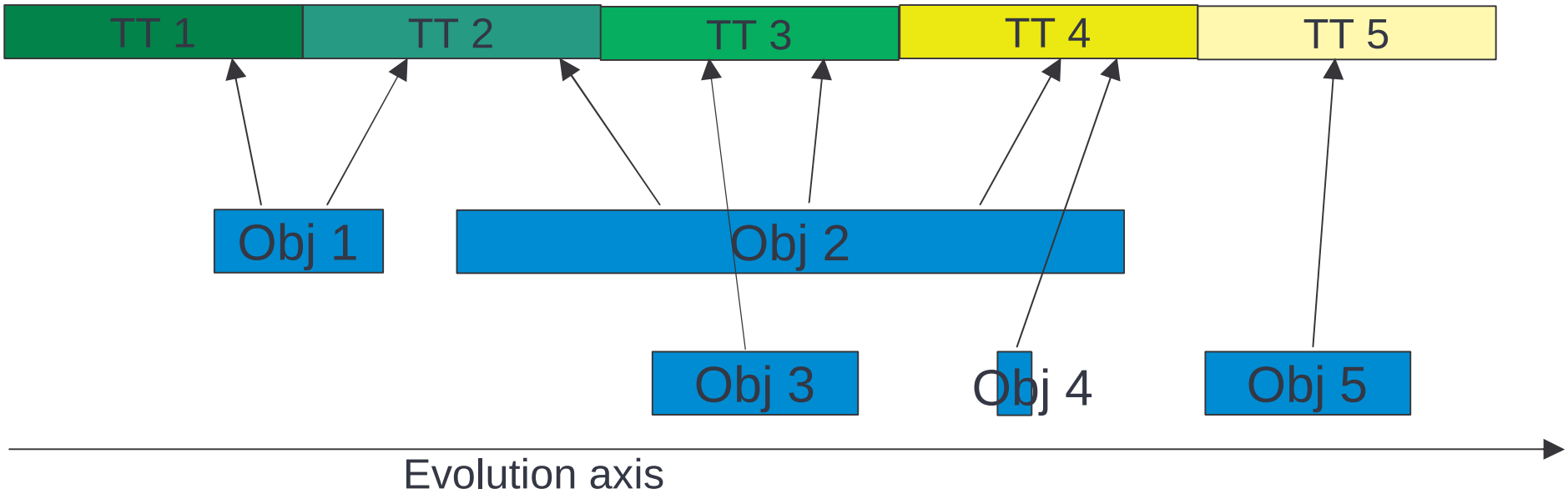
Object retrieval in a time partitioned environment

- Retrieving in a particular point is a trivial case.
- Retrieving in a time interval is transparent with some overhead.



Object storage in a time partitioned environment

- An object that crosses a partition boundary is replicated
- Overhead associated with object duplication.
- Online performance shouldn't be affected (starting now).
- Acceptable offline overhead.



API layered structure

Upper layer: set of virtual classes imposed by the interface specification

Middle layer: implementation specific concrete classes, derived from the interface classes.

Bottom layer (NEW) : mySQL tight connected classes (replacing the old implementation's wrapper functions)

Why the additional bottom layer

- Allows one to plug/unplug/replace new features not foreseen in the interface:
  - Time partitioning management; administrative tasks
- Clearer code interfaces:
  - Code is easier to maintain or extend;
  - improved robustness.
- Painless integration with the rest of the code and possibly more efficient coding.

Model rules

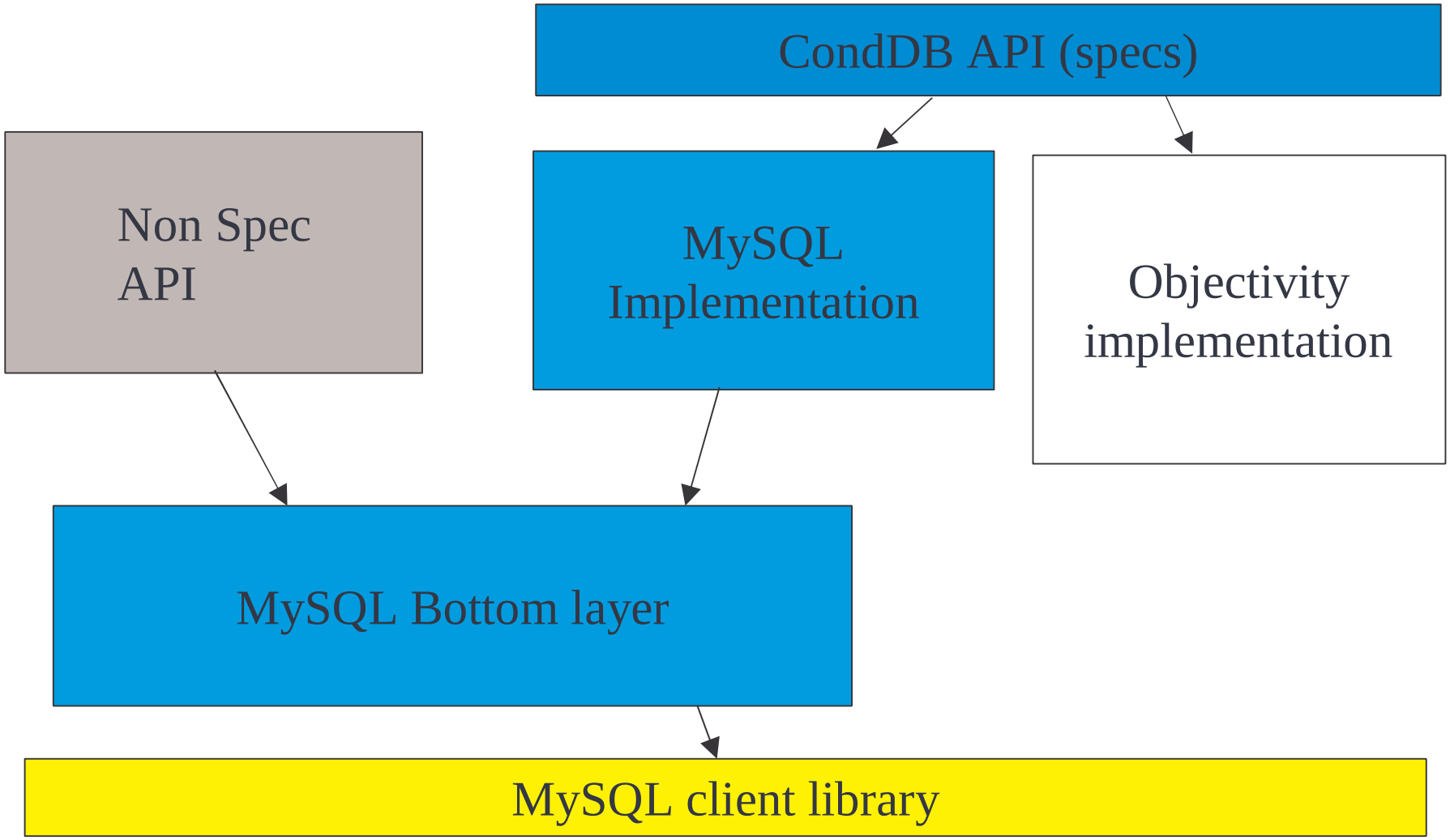
Upper layer: the overall database structure is a guarantied at this level

Middle layer: specific topics like the “*time validity range*”, the “*insertion time*”, the correctness of folderset/folder hierarchy are assured by this layer.

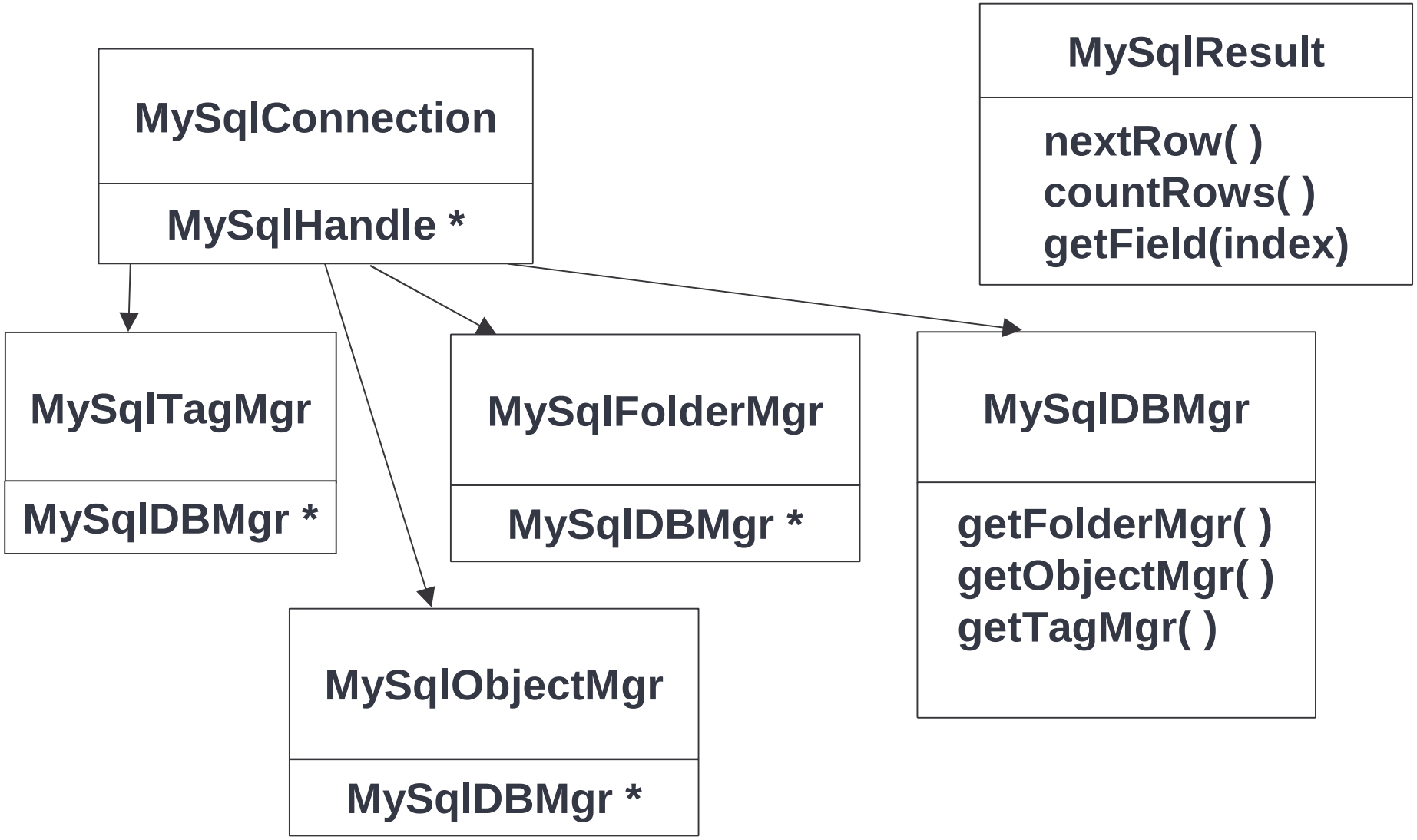
Bottom layer: Time partitioning policies and correctness

MySQL level: *define default values... what else?*

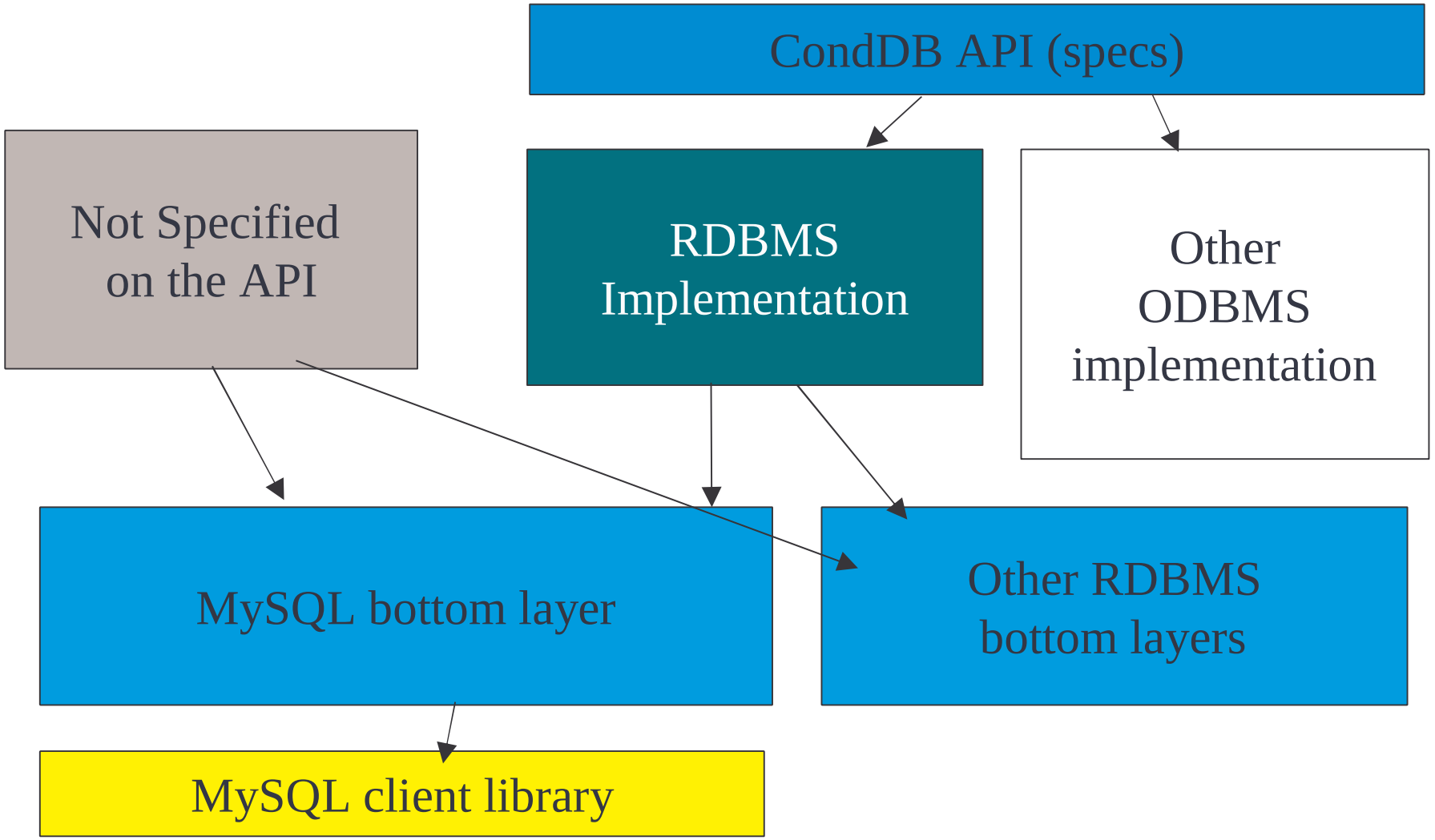
Architecture



Bottom Layer snapshot



Architecture



Bottom Layer snapshot