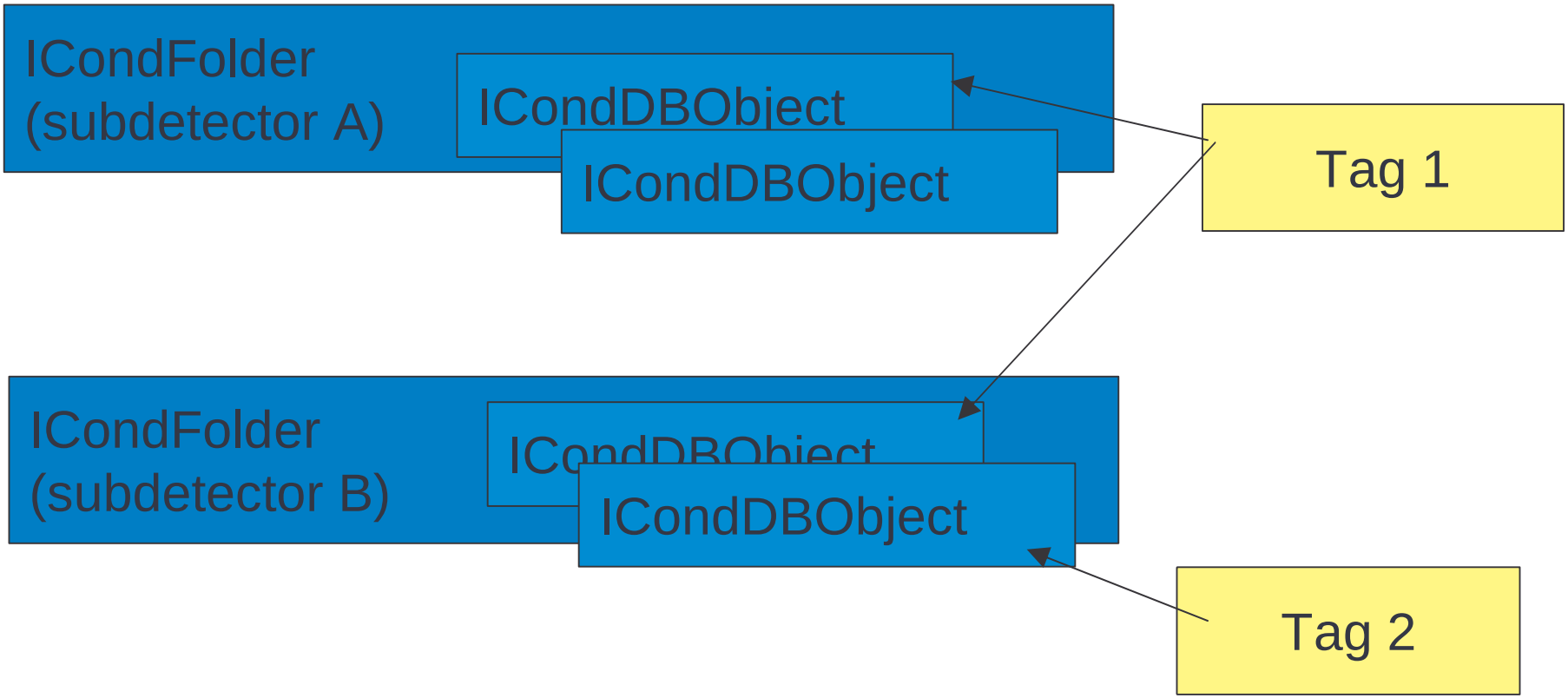
ATLAS Online Software

ConditionsDB MySQL Backend Implementation

ConditionsDB terminology

- Filesystem like hierachy



ConditionDB Overview

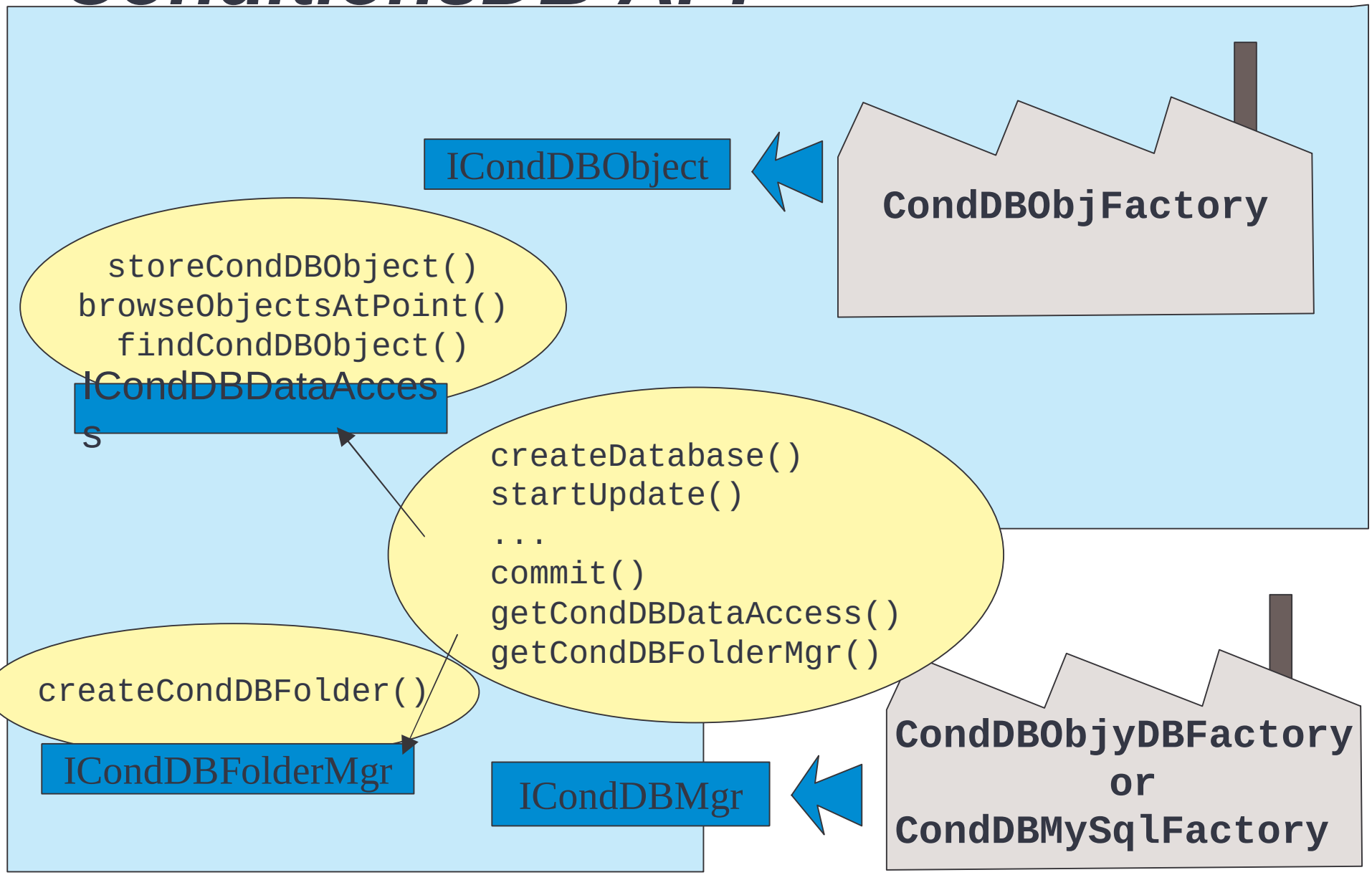
- Object description

ICondFolder : "*detector element*"; "detector element property"; "*data type*"

ICondDBObject : "*time validity range*"; "*insertion time*"; "*data block or hash key*"

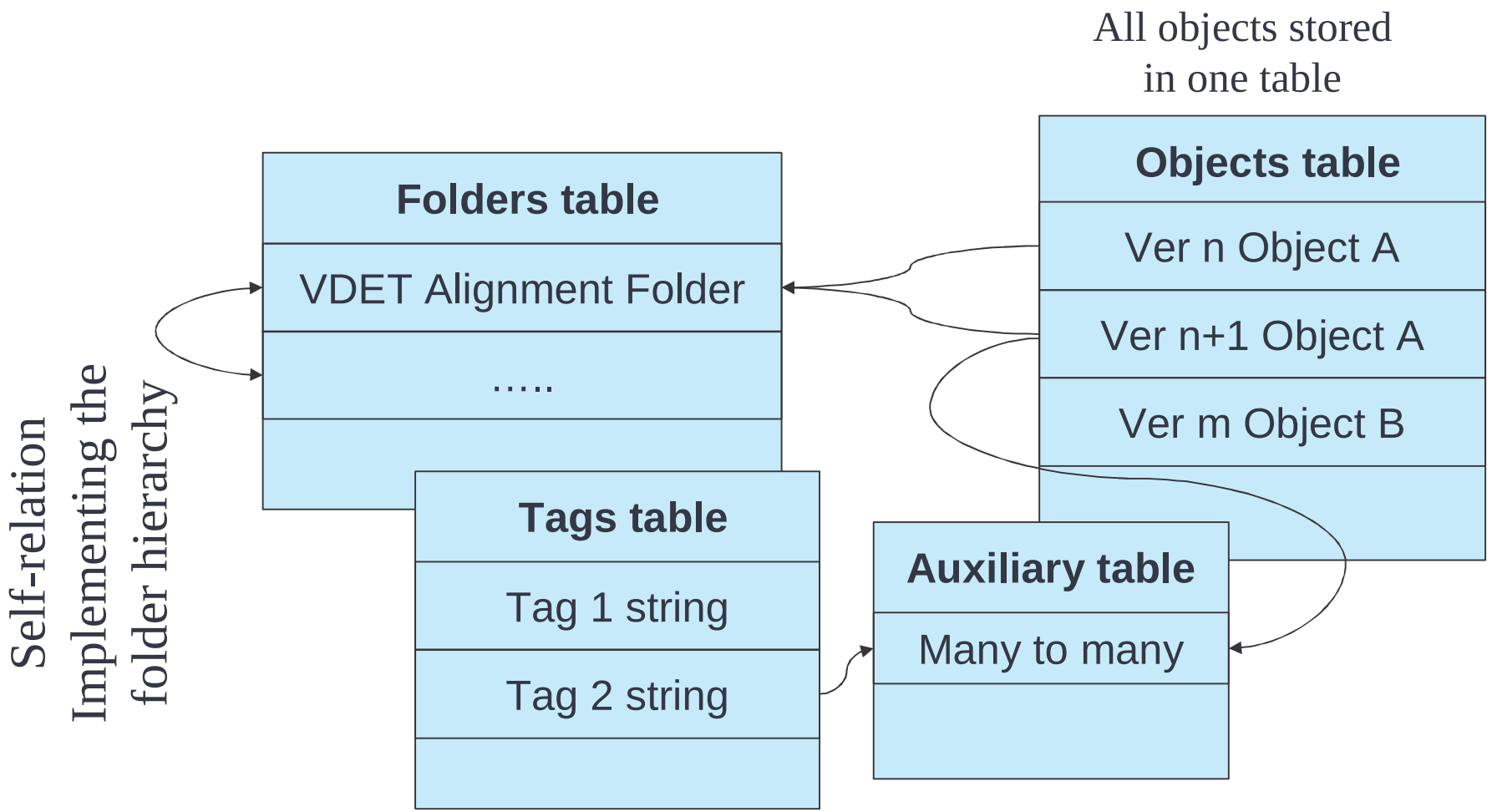
Tag 2 : "*description string*"

ConditionsDB API



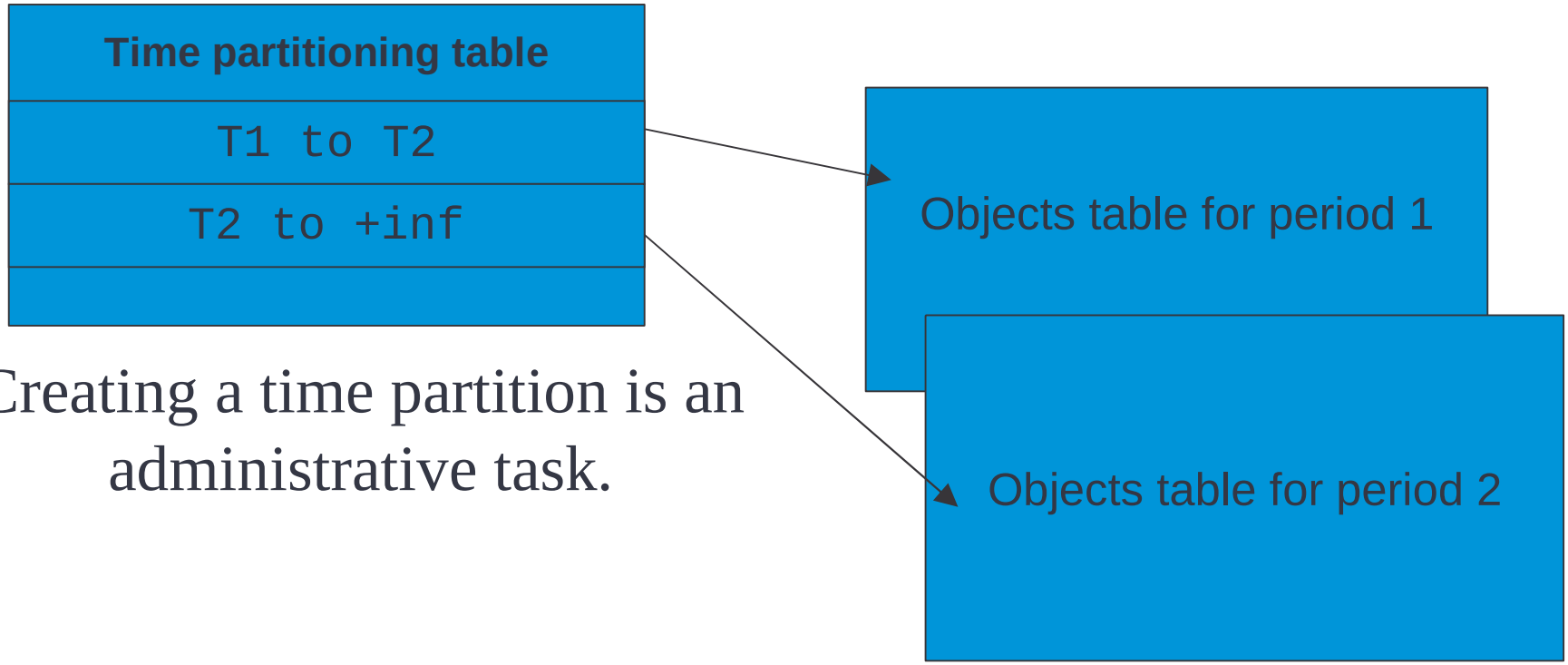
Relational DB model

- Simplified tables relationship



Relational DB model

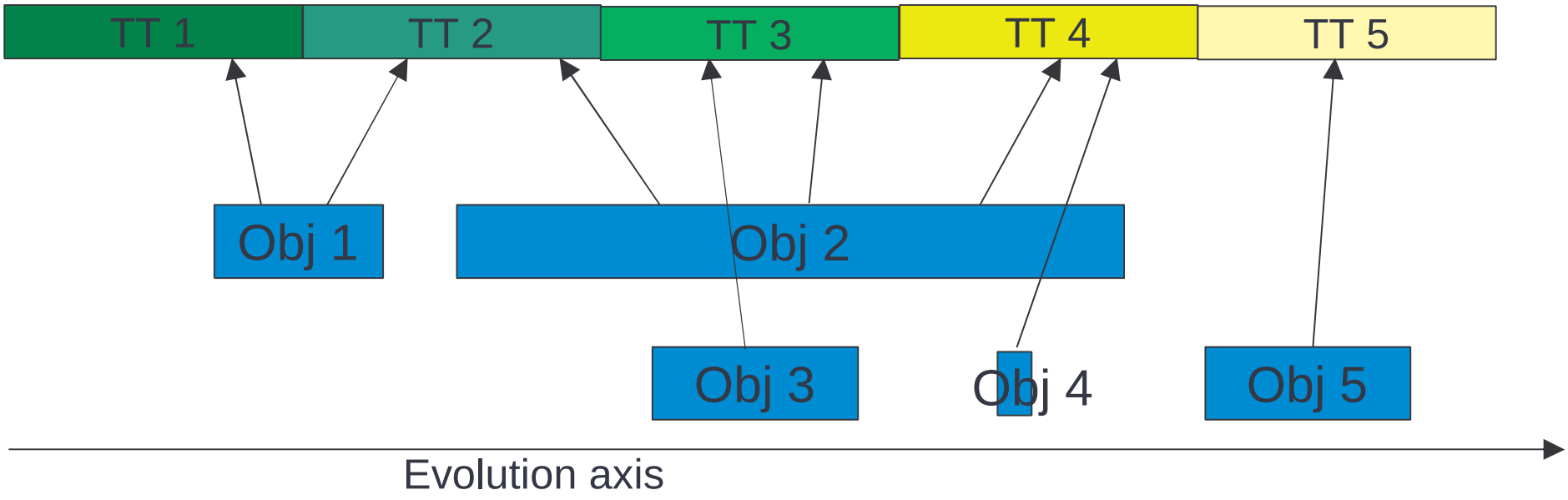
- No clustering on cathegory ?
- Clustering over large time intervals.
- Scalability allowing different databases and database servers.



Creating a time partition is an administrative task.

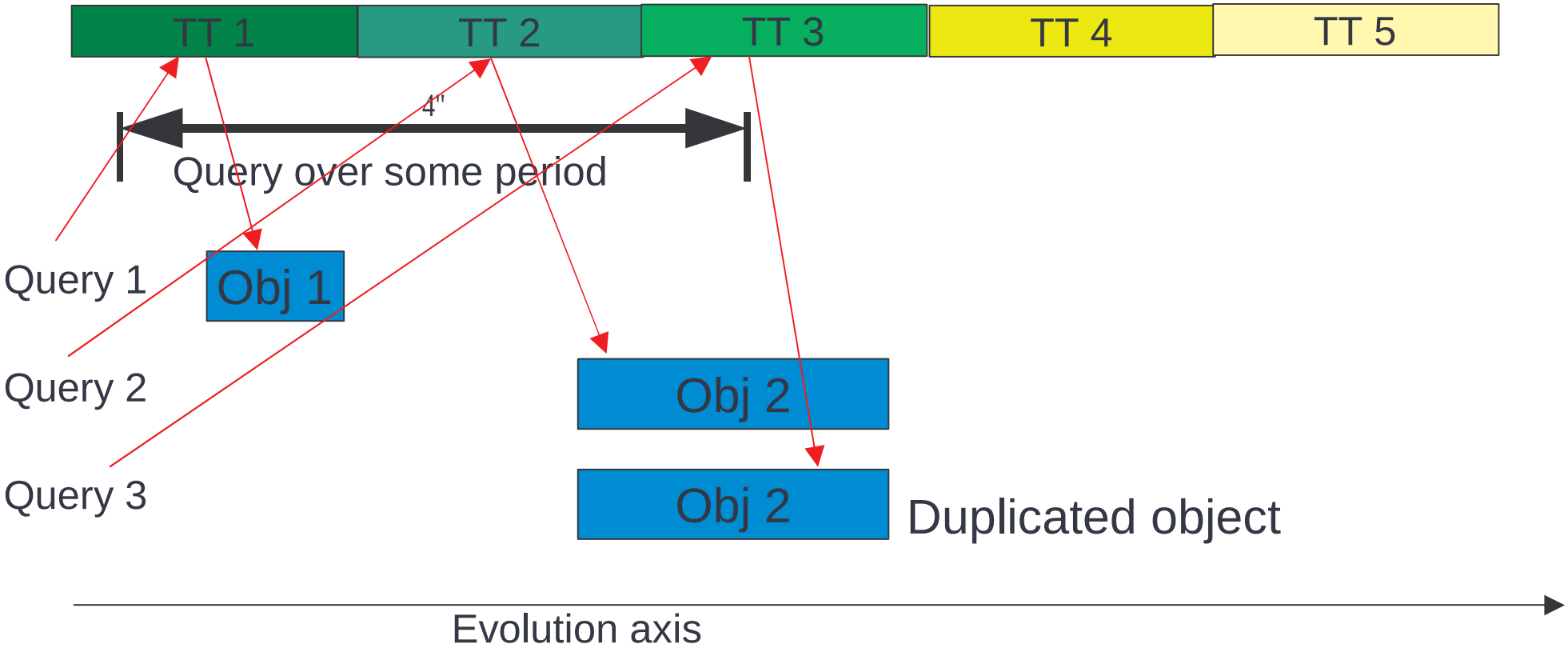
Object storage in a time partitioned environment

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



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.



Code example (DBMS independence)

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

...

  ICondDBMgr* CondDBmgr =
    CondDBMySQLMgrFactory::createCondDBMgr();

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

  CondDBMySQLMgrFactory::destroyCondDBMgr( CondDBmgr );
```