FairRoot Database Interface

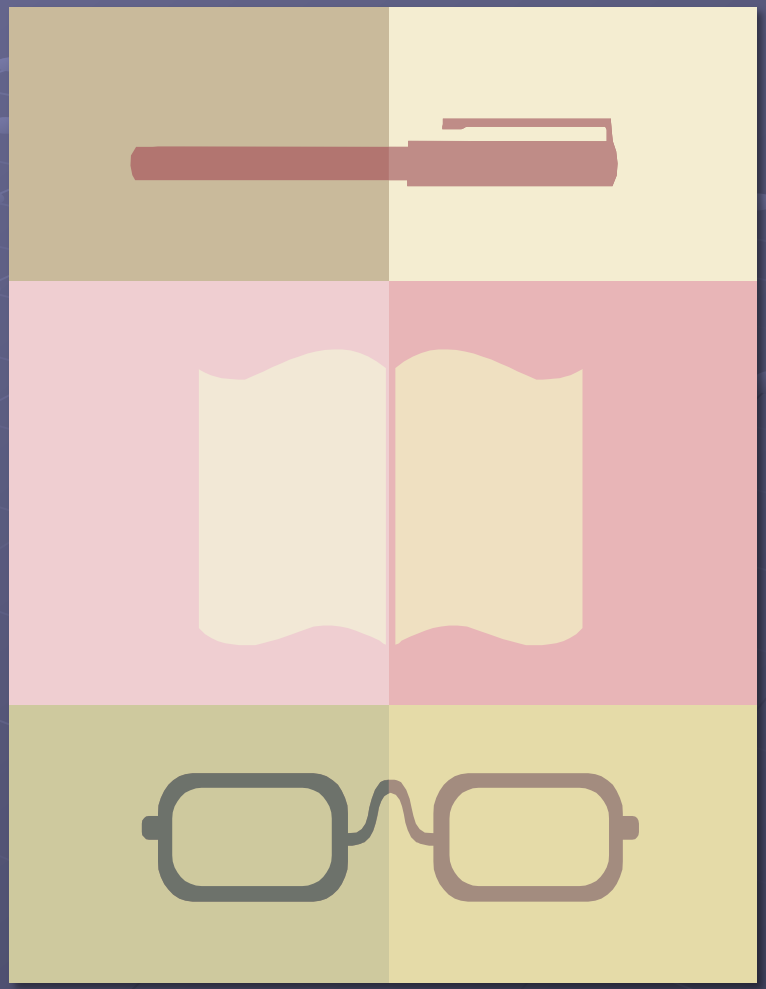
Denis Bertini (IT-GSI)
Mohammad Babai (KVI)

Outline

- Why we don’t want a database ?
- Why do we want a database ?
- Where do we start ?
- Database connectivity
- Version Management
- Database migration
- I/O: Generic parameter container

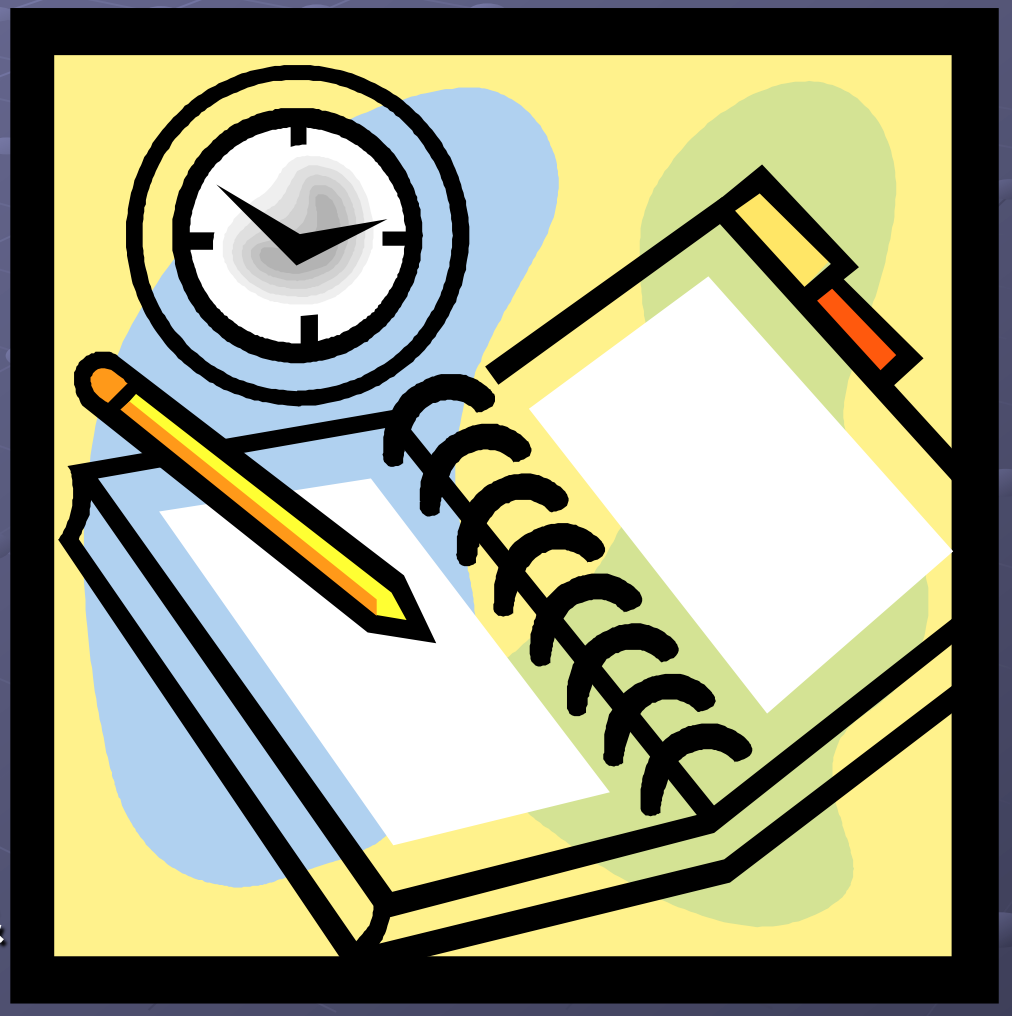
Why we don’t want a database?

- **DBs are complex**, difficult and time-consuming to design
- **“garbage in – garbage out”.**
  - Who will perform data entry?
  - Learning +Training? On-line help?
  - How data entry will be performed?
- Initial training required for all users
- **Damage to Dbs** will affect everybody
- Extensive conversion costs in moving from a DBs to another
- Substantial hardware and software start-up costs



Why do we want a database?

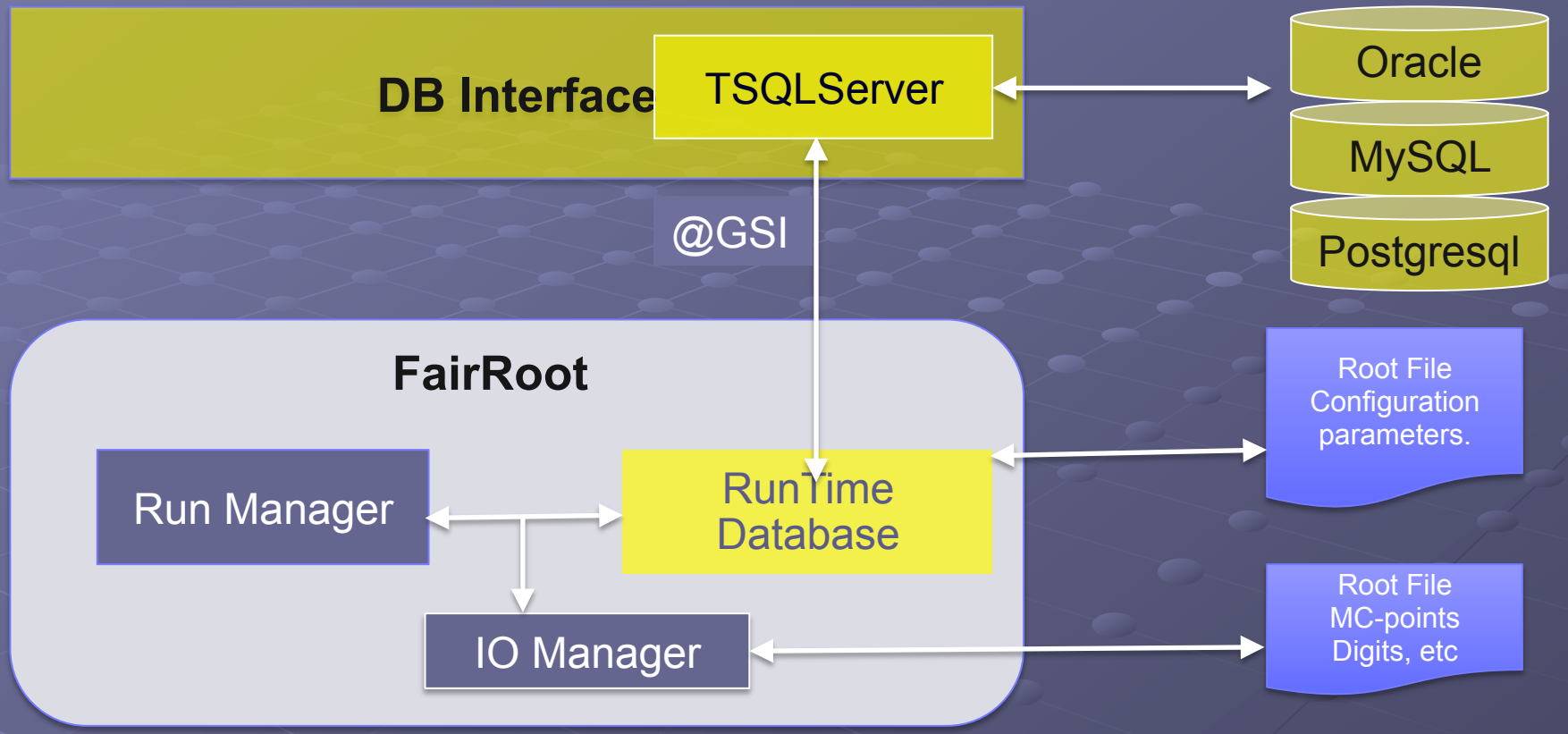
- To keep an “history” of records
  - Online / Offline Parameters
  - Experimental setups
  - etc ..
- To share these records between users
- To able users to access the same records concurrently
- To provide efficient access to large amount of data without caring about physical storage format
- To ease communication between users
- To ensure Data consistency & integrity



Where do we start?

- **Database Interface in FairRoot using TSQLServer**
  - (MySQL, Oracle, PostGre,...)
- Allows multiple connections to Dbs at runtime
- Adds **Version Management**
  - Data type: Real and/or MC
  - Detector type
  - Date and Time Range
- Reduces SQL coding
  - Simple Predefined Table
  - Only Simple SQL used
  - Ultimately **Generic Container**
- Handles Write/Read access

FairRoot DB extended

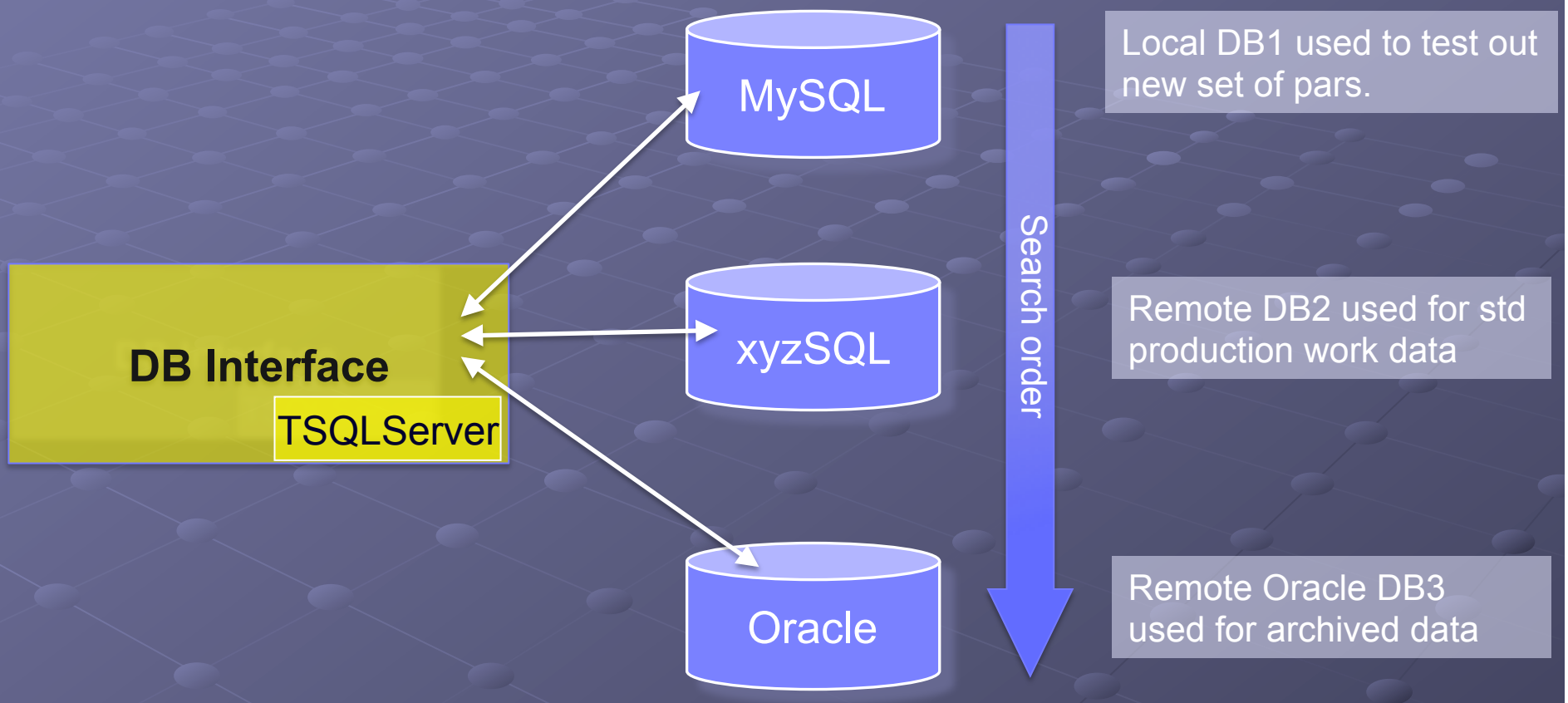


Runtime Database

- *RootFileIo*
- AsciiFileIO
- TSQLIO(Generic & detector pars) ?

@(KVI+GSI)

Database connectivity(1)



Database Connectivity(2)

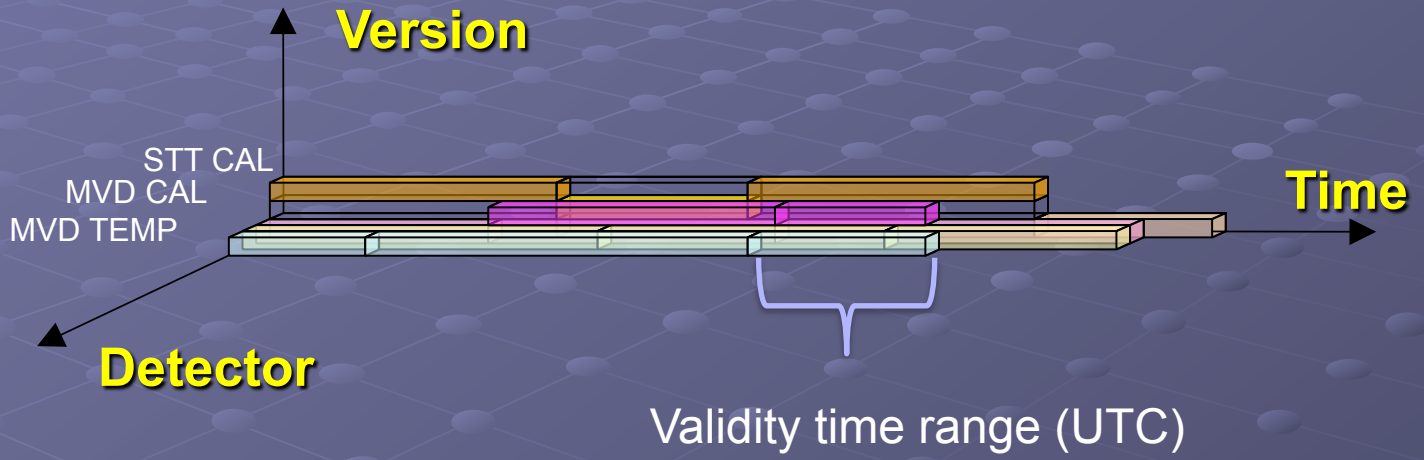
- Configuring connections (JDBC):

```
// db_config.sh
  export ENV_DB_URL="mysql://demac013.gsi.de/r3b;oracle://bka.oracle.gsi.de/db-test"
  export ENV_DB_USER="denis;scott"
  export ENV_DB_PSWD="mypasswd;tiger"
```

- Opening connections

```
// create a priority-list of databases
  FairDbMultConnector *fMultConn  = FairDbMultConnector();
  for (Int_t dbId = 0;dbId< fMultConn->GetDnNo(); ++dbId) {
     auto_ptr<FairDbStatement> fStmt(fMultConn->CreateStatement(dbId));
      // execute SQL stmt for dbId
   fStmt->ExecuteUpdate("DROP TABLE IF EXISTS R3BDBDEMODATA1");
...
}
```

Version Management



The Query process

1. Context (Timestamp,Detector,Version) is the primary key
2. Context converted to unique SeqNo
3. SeqNo used as keys to access all rows in main table
4. System gives user access of all such rows

I/0 using generic container

- Ultimately hides the SQL coding to the user, no DB expertise needed
- Allows simple type data member and also complex ones
- Stores as Table and/or BLOB (complex types)
- Schema evolution: uses streamer info , ROOT version

Database Migration

- Problem : “S” in SQLP does not mean Standard !!!
- Solution: keep the SQL code simple, reduce it to minimum
- Vendor-dependant SQL not allowed
- Database Interface supports a “on-the-fly” conversion

SQL conversion

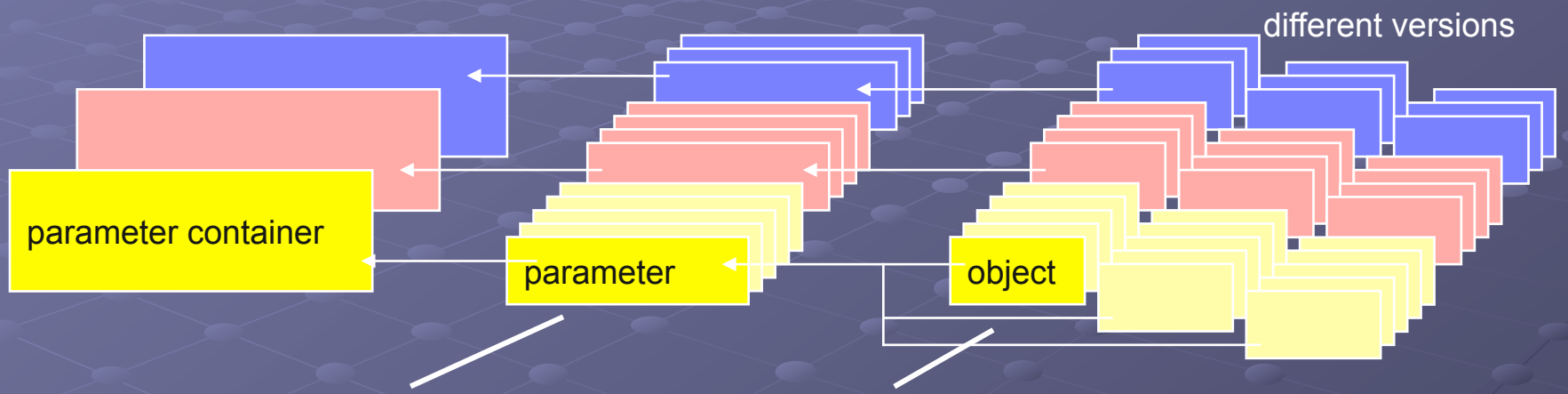
```
Recreating  FairDbTableMetaData for table R3DBDEMODATA3
  Column: 1 name SEQNO type Int precision 0
  Column: 2 name ROW_COUNTER type Int precision 0
  Column: 3 name MYBOOL_1 type String precision 0
  Column: 4 name MYBOOL_2 type Char precision 0
  Column: 5 name MYCHAR type Char precision 0
  Column: 6 name MYSTRING type Char precision 0
  Column: 7 name MYCHARS type String precision 20
  Column: 8 name MYCHARS2 type String precision 2
  Column: 9 name MY_SS_TINY_1 type Tiny precision 0
  Column: 10 name MY_SS_TINY_2 type Tiny precision 0
  Column: 11 name MY_SS_TINY_3 type Tiny precision 0
  Column: 12 name MY_SS_TINY_4 type Tiny precision 0
  Column: 13 name MY_UU_TINY_1 type Tiny precision 0
  Column: 14 name MY_UU_TINY_2 type Tiny precision 0
  Column: 15 name MY_UU_TINY_3 type Tiny precision 0
  Column: 16 name MY_UU_TINY_4 type Tiny precision 0
  Column: 17 name MY_US_TINY_1 type Tiny precision 0
  Column: 18 name MY_US_TINY_2 type Tiny precision 0
  Column: 19 name MY_US_TINY_3 type Tiny precision 0
  Column: 20 name MY_US_TINY_4 type Tiny precision 0
  Column: 21 name MY_SS_SHORT_1 type Short precision 0
  Column: 22 name MY_SS_SHORT_2 type Short precision 0
  Column: 23 name MY_SS_SHORT_3 type Short precision 0
  Column: 24 name MY_SS_SHORT_4 type Short precision 0
  Column: 25 name MY_UU_SHORT_1 type Short precision 0
  Column: 26 name MY_UU_SHORT_2 type Short precision 0
  Column: 27 name MY_UU_SHORT_3 type Short precision 0
  Column: 28 name MY_UU_SHORT_4 type Short precision 0
  Column: 29 name MY_US_SHORT_1 type Short precision 0
  Column: 30 name MY_US_SHORT_2 type Short precision 0
  Column: 31 name MY_US_SHORT_3 type Short precision 0
  Column: 32 name MY_US_SHORT_4 type Short precision 0
  Column: 33 name MY_SS_INT_1 type Int precision 0
  Column: 34 name MY_SS_INT_2 type Int precision 0
  Column: 35 name MY_SS_INT_3 type Int precision 0
  Column: 36 name MY_SS_INT_4 type Int precision 0
  Column: 37 name MY_US_INT_1 type Int precision 0
  Column: 38 name MY_US_INT_2 type Int precision 0
  Column: 39 name MY_US_INT_3 type Int precision 0
  Column: 40 name MY_US_INT_4 type Int precision 0
  Column: 41 name MYFLOAT_1 type Float precision 0
  Column: 42 name MYFLOAT_2 type Float precision 0
  Column: 43 name MYDOUBLE_1 type Double precision 0
  Column: 44 name MYDOUBLE_2 type Double precision 0
-I- FairDbStatement::TranslateSQL  sql: create table R3DBD
```

```
mysql> describe R3DBDEMODATA3;
+--------------+---------------------+------+-----+---------+-------+
| Field        | Type                | Null | Key | Default | Extra |
+--------------+---------------------+------+-----+---------+-------+
| SEQNO        | int(11)             | NO   | PRI | NULL    |       |
| ROW_COUNTER  | int(11)             | NO   | PRI | NULL    |       |
| MYBOOL_1     | char(1)             | YES  |     | NULL    |       |
| MYBOOL_2     | char(1)             | YES  |     | NULL    |       |
| MYCHAR       | char(1)             | YES  |     | NULL    |       |
| MYSTRING     | text                | YES  |     | NULL    |       |
| MYCHARS      | char(20)            | YES  |     | NULL    |       |
| MYCHARS2     | char(2)             | YES  |     | NULL    |       |
| MY_SS_TINY_1 | tinyint(4)          | YES  |     | NULL    |       |
| MY_SS_TINY_2 | tinyint(4)          | YES  |     | NULL    |       |
| MY_SS_TINY_3 | tinyint(4)          | YES  |     | NULL    |       |
| MY_SS_TINY_4 | tinyint(4)          | YES  |     | NULL    |       |
| MY_UU_TINY_1 | tinyint(3) unsigned | YES  |     | NULL    |       |
| MY_UU_TINY_2 | tinyint(3) unsigned | YES  |     | NULL    |       |
| MY_UU_TINY_3 | tinyint(3) unsigned | YES  |     | NULL    |       |
| MY_UU_TINY_4 | tinyint(3) unsigned | YES  |     | NULL    |       |
| MY_US_TINY_1 | tinyint(4)          | YES  |     | NULL    |       |
| MY_US_TINY_2 | tinyint(4)          | YES  |     | NULL    |       |
| MY_US_TINY_3 | tinyint(4)          | YES  |     | NULL    |       |
| MY_US_TINY_4 | tinyint(4)          | YES  |     | NULL    |       |
| MY_SS_SHORT_1 | smallint(6)        | YES  |     | NULL    |       |
| MY_SS_SHORT_2 | smallint(6)        | YES  |     | NULL    |       |
| MY_SS_SHORT_3 | smallint(6)        | YES  |     | NULL    |       |
| MY_SS_SHORT_4 | smallint(6)        | YES  |     | NULL    |       |
| MY_UU_SHORT_1 | smallint(5) unsigned | YES  |     | NULL    |       |
| MY_UU_SHORT_2 | smallint(5) unsigned | YES  |     | NULL    |       |
| MY_UU_SHORT_3 | smallint(5) unsigned | YES  |     | NULL    |       |
| MY_UU_SHORT_4 | smallint(5) unsigned | YES  |     | NULL    |       |
| MY_US_SHORT_1 | smallint(6)        | YES  |     | NULL    |       |
| MY_US_SHORT_2 | smallint(6)        | YES  |     | NULL    |       |
| MY_US_SHORT_3 | smallint(6)        | YES  |     | NULL    |       |
| MY_US_SHORT_4 | smallint(6)        | YES  |     | NULL    |       |
| MY_SS_INT_1  | int(11)             | YES  |     | NULL    |       |
| MY_SS_INT_2  | int(11)             | YES  |     | NULL    |       |
| MY_SS_INT_3  | int(11)             | YES  |     | NULL    |       |
| MY_SS_INT_4  | int(11)             | YES  |     | NULL    |       |
| MY_US_INT_1  | int(11)             | YES  |     | NULL    |       |
| MY_US_INT_2  | int(11)             | YES  |     | NULL    |       |
| MY_US_INT_3  | int(11)             | YES  |     | NULL    |       |
| MY_US_INT_4  | int(11)             | YES  |     | NULL    |       |
| MYFLOAT_1    | float               | YES  |     | NULL    |       |
| MYFLOAT_2    | float               | YES  |     | NULL    |       |
| MYDOUBLE_1   | double              | YES  |     | NULL    |       |
| MYDOUBLE_2   | double              | YES  |     | NULL    |       |
+--------------+---------------------+------+-----+---------+-------+
44 rows in set (0.02 sec)
```

```
Column: 3 name INSERTDATE type Date precision 0
-I- FairDbStatement::TranslateSQL  sql: CREATE TABLE R3DBDEMODATA2VAL(    SEQNO       INT NOT NULL PRIMARY KEY AUTO_INCREMENT,    TIMESTART    DATETIME NOT NULL,    TIMEEND      DATETIME NOT NULL,    DETECTORMASK  TINYINT,
Translates to 5 statements:-
-I- FairDbStatement::TranslateSQL    ALTER SESSION SET NLS_DATE_FORMAT='YYYY-MM-DD hh24:mi:ss'
-I- FairDbStatement::TranslateSQL    create table R3DBDEMODATA2VAL(SEQNO NUMBER(11) not null, TIMESTART DATE not null, TIMEEND DATE not null, DETECTORMASK NUMBER(4), SIMMASK NUMBER(4), TASK NUMBER(11), AGGREGATENO NUMBER(11), CR
-I- FairDbStatement::TranslateSQL     create public synonym R3DBDBDEMODATA2VAL for R3DBDBDEMODATA2VAL
-I- FairDbStatement::TranslateSQL     grant  select on R3DBDBDEMODATA2VAL to r3b_reader
-I- FairDbStatement::TranslateSQL     grant  select,insert,update on R3DBDBDEMODATA2VAL to r3b_writer
-I- FairDbStatement::ExecuteUpdate SQL:db-test:ALTER SESSION SET NLS_DATE_FORMAT='YYYY-MM-DD hh24:mi:ss'
-I- FairDbStatement::ExecuteUpdate SQL:db-test:create table R3DBDBDEMODATA2VAL(SEQNO NUMBER(11) not null, TIMESTART DATE not null, TIMEEND DATE not null, DETECTORMASK NUMBER(4), SIMMASK NUMBER(4), TASK NUMBER(11), AGGREGATENO NUMB
-D- R3BDbPrimer: Priming DB 0  SQL:CREATE TABLE R3BDBDEMODATA2(    SEQNO     INT,    ROW_COUNTER  INT,    AGGREGATENO   INT,    SUBSYSTEM  INT,    DATA      FLOAT);
```

HADES implementation

Oracle only solution base on server-side PL/SQL code



name
type: Int_t, Float_t, Double_t, Char_t, Text_t, UChar_t, class type

any class derived from TObject
decoded in the analysis interface
by ROOT streamer

stored as byte array (RAW or BLOB)
number of values (single value or array)
class version
streamer info, root version for ROOT classes
own version management

Conclusions

- FaiRoot Database Interface is in SVN
- Ongoing work on
  - RuntimeDB connection and initialisation scheme (GSI)
  - Implementation of Generic Parameter Container (KVI)
  - Preparation of realistic tests (R3B, Panda …)