KEKB Static Database Design

presented by

Noboru Yamamoto

at EPICS Collaboration meeting
Nov. 2000

Based on the talk by T-T. Nakamura at EPICS meeting in KEK, May 2000.

Contents

1. Overview
2. EPICS database configuration by the RDB
3. Example
4. Access from high-level applications

Overview

1. KEKB-DB is the Relational Database for the KEKB accelerator control
   - Relational Database Manager : Oracle7 on HP-UX
   - Contents: Static parameters of the accelerator components
     - Device parameters
     - Wiring information
     - Magnetic Field Measurement Data
     - BPM Mapping Data
     - Other Calibration Constants
     - Lattice information

Main usage of KEKB-DB

1. EPICS run-time database (process DB) are configured by the RDB
2. High-level application programs get information from the RDB
3. Human gets information from the RDB

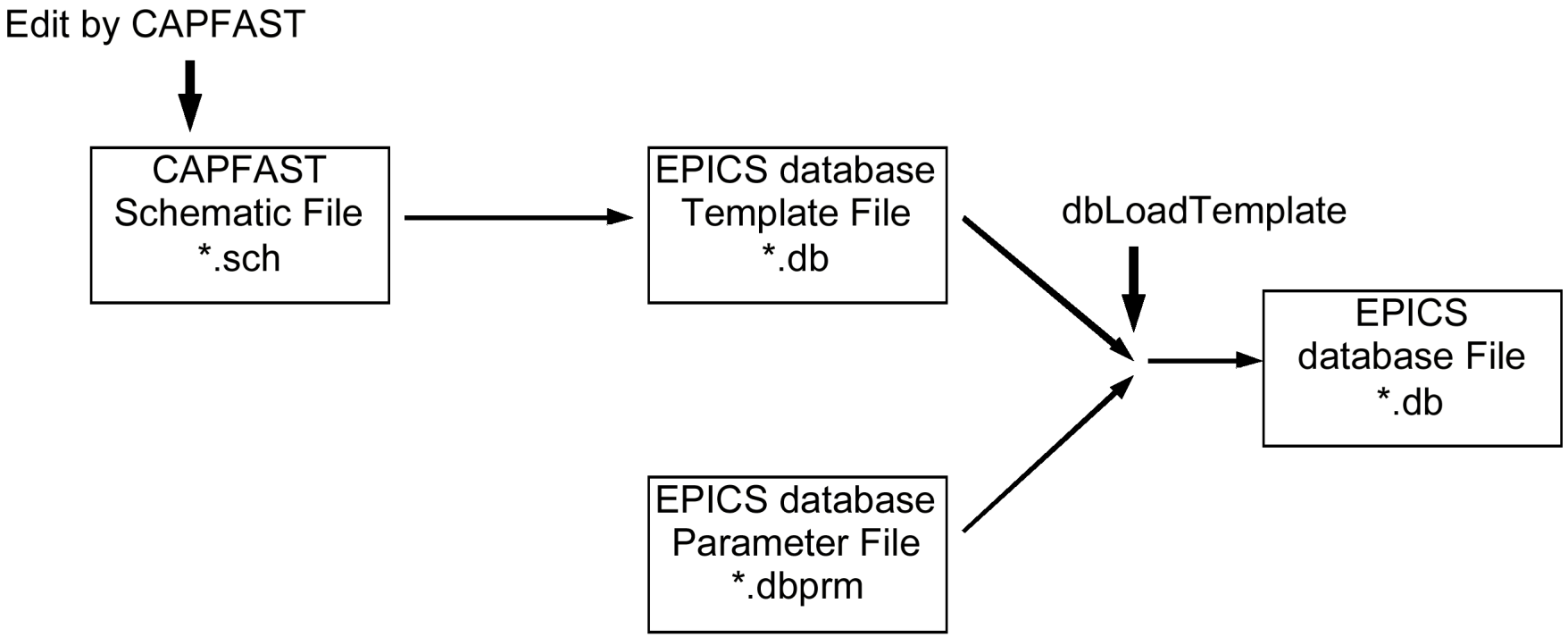
KEKB-DB is the unique provider of static information

- for IOCs (run-time database),
- for Application programs

  and
- for Humen.

EPICS database configuration by the RDB

How do you make EPICS run-time database for several thousands of devices ?



EPICS database Template File BTMGPS.db

```
record(ai,"$(HEAD):$(NAME):INP_AMP") {
field(DESC,"analog input record")
field(SCAN,"Passive")
field(PINI,"NO")
field(DTYP,"Soft Channel")
field(DISV,"1")
field(DISS,"NO_ALARM")
field(PRIO,"LOW")
field(FLNK,"$(HEAD):$(NAME):INP_AMP_PUT.VAL")
field(INP,"0.000000000000000e+00")
field(PREC,"0")
field(LINR,"NO CONVERSION")
field(EGU,"volts")
field(HOPR,"$(PS_MAX)")
field(LOPR,"$(PS_MIN)")
field(AOFF,"0.0000000e+00")
```

EPICS database Parameter File MGPS_B.dbprm

```
file BTMGPS.db {
    {
        CHANNEL="1",
        DV_ADR="1",
        DV_AO=".00827",
        DV_AS="1.0000389",
        DV_CH="101",
        DV_K="20",
        DV_P_HI="65",
        DV_P_LO="5",
        HEAD="BTePS",
        NAME="B0E",
        NODE_NO="1",
        PS_MAX="200",
        PS_MIN="0",
        THRU_COEFF=" 3.333333"
    }
}
```

How do you prepare and maintain the parameter files ?

It seems not good idea to maintain the file itself.

Some parameters are common to the same device type.

Some parameters appear in the different files.

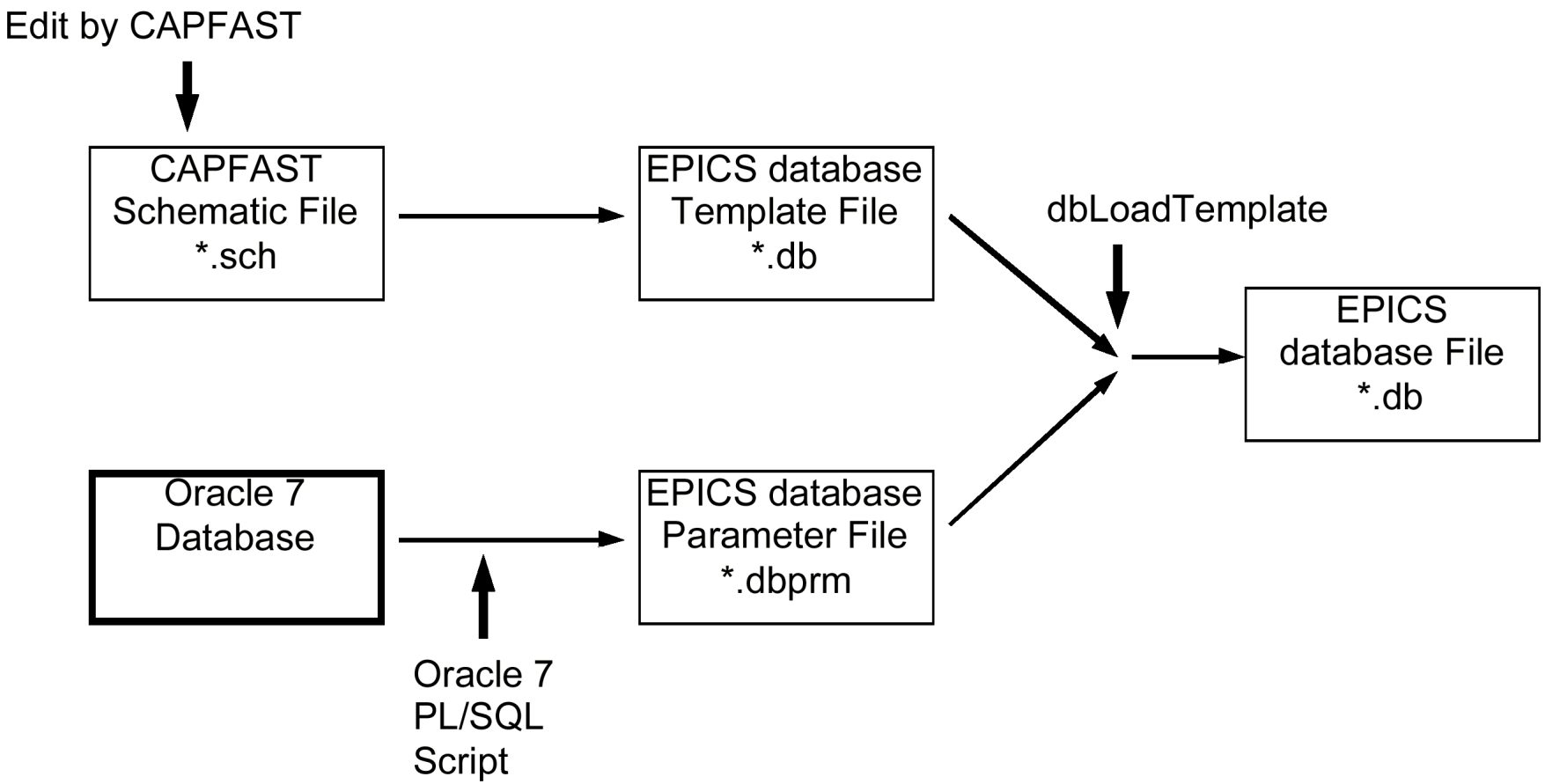
Maintain consistency of these parameters

---- > Use a RDB.

KEKB-DB is the source of information.

All of the secondary data should be derived from KEKB-DB.

Maintenance of the parameters is concentrated on the KEKB-DB.


Edit by CAPFAST
CAPFAST
Schematic File
*.sch
EPICS database
Template File
*.db
dbLoadTemplate
EPICS
database File
*.db
Oracle 7
Database
EPICS database
Parameter File
*.dbprm
Oracle 7
PL/SQL
Script

Example (Magnet system)

1. There are >30 tables for the magnet system
2. Tables for magnets
   - MG.MAGNET_ALL
   - MG.MAGNET_ENTITY
   - MG.MAGN ET_TYPE
3. Tables for magnet power supplies
   - MG.MAGNET_PS_ALL
   - MG.MAGNET_PS_ENTITY
   - MG.MAGNET_PS_TYPE
4. In RDB, a table is a kind of 2-D array (column x row).
5. Each column has own name and data type.

1. MG.MAGNET_PS_ALL

- This table describes functional properties.
- Each row corresponds to a magnet power supply.
- The columns are:
  - ATTRIB beam line identifier (LER, HER, e+ BT, e- BT, LINAC, PF-AR, AR-BT,...)
  - NAME functional name of the power supply
    - (The name comes from lattice design)
  - COMMENTS comments
  - HW_NAME name (alias) mainly used in hardware groups
  - LOCATION power supply building
  - SAFETY_PS interlocked with radiation safety system or not
  - PS_TYPE_ID type identifier of the power supply
  - SERIAL product serial number
  - etc.

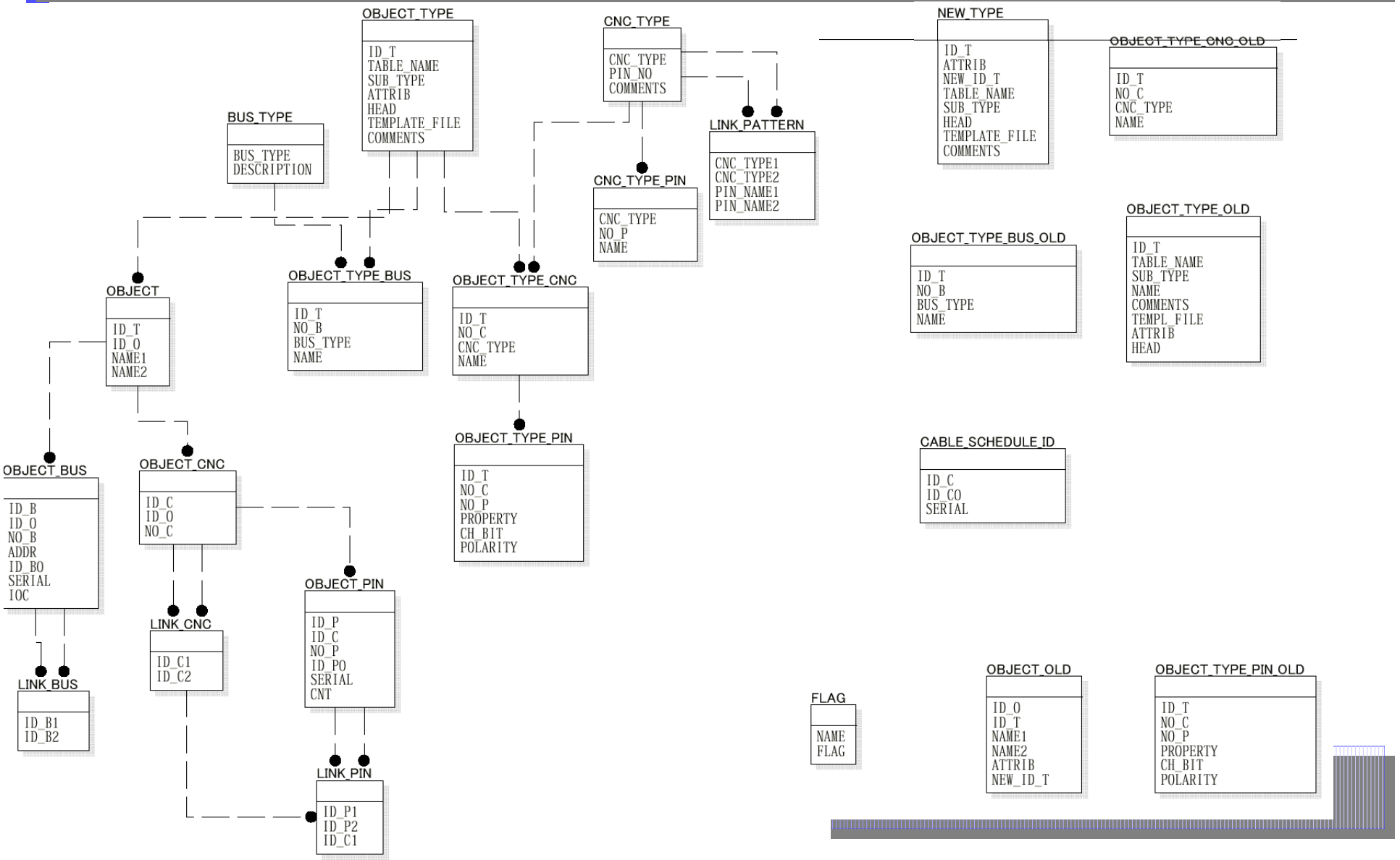
1. MG.MAGNET_PS_ENTITY

- This table describes physical properties .
- Each row corresponds to a magnet power supply.
- The columns are:
  - PS_TYPE_ID type identifier of the power supply
  - SERIAL product serial number
  - COMMENTS comments
  - J_REG , K_REG calibration constants for current setting
  - (DAC current)
  - MJ_REG , MK_REG calibration constants for current monitor
  - (monitored voltage current)
  - etc.

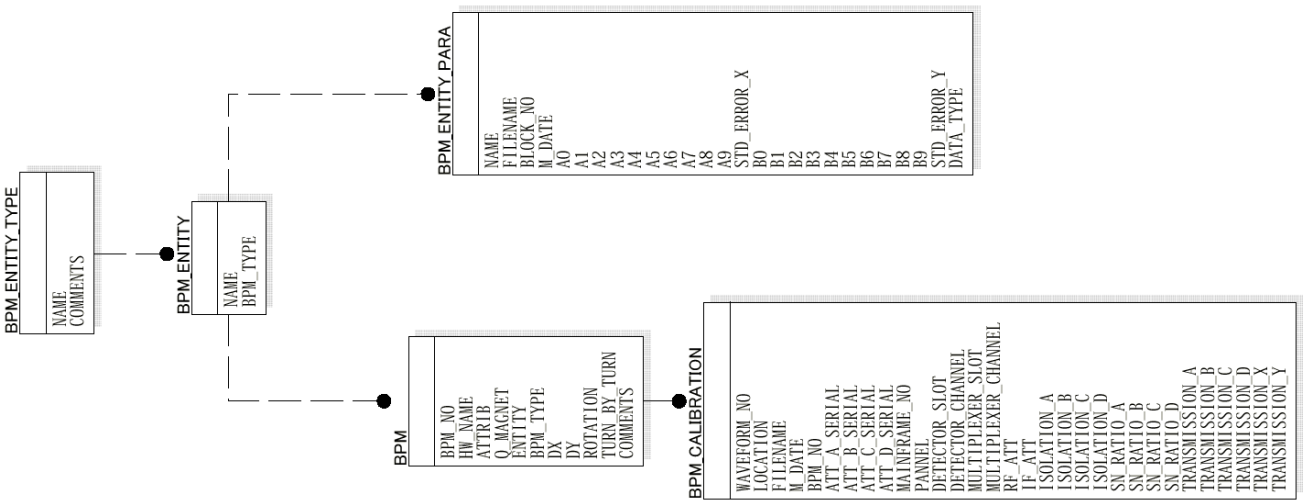
1. MG.MAGNET_PS_TYPE

- This table describes physical properties common to
- the same power supply type.
- Each row corresponds to a power supply type.
- The columns are:
  - ID type identifier of the power supply
  - MAKER manufacturer
  - BI_POLAR bipolar or not
  - A_MAX maximum output current
  - V_MAX maximum output voltage
  - REGULATOR type of current regulator
  - COMMENTS comments
  - etc.

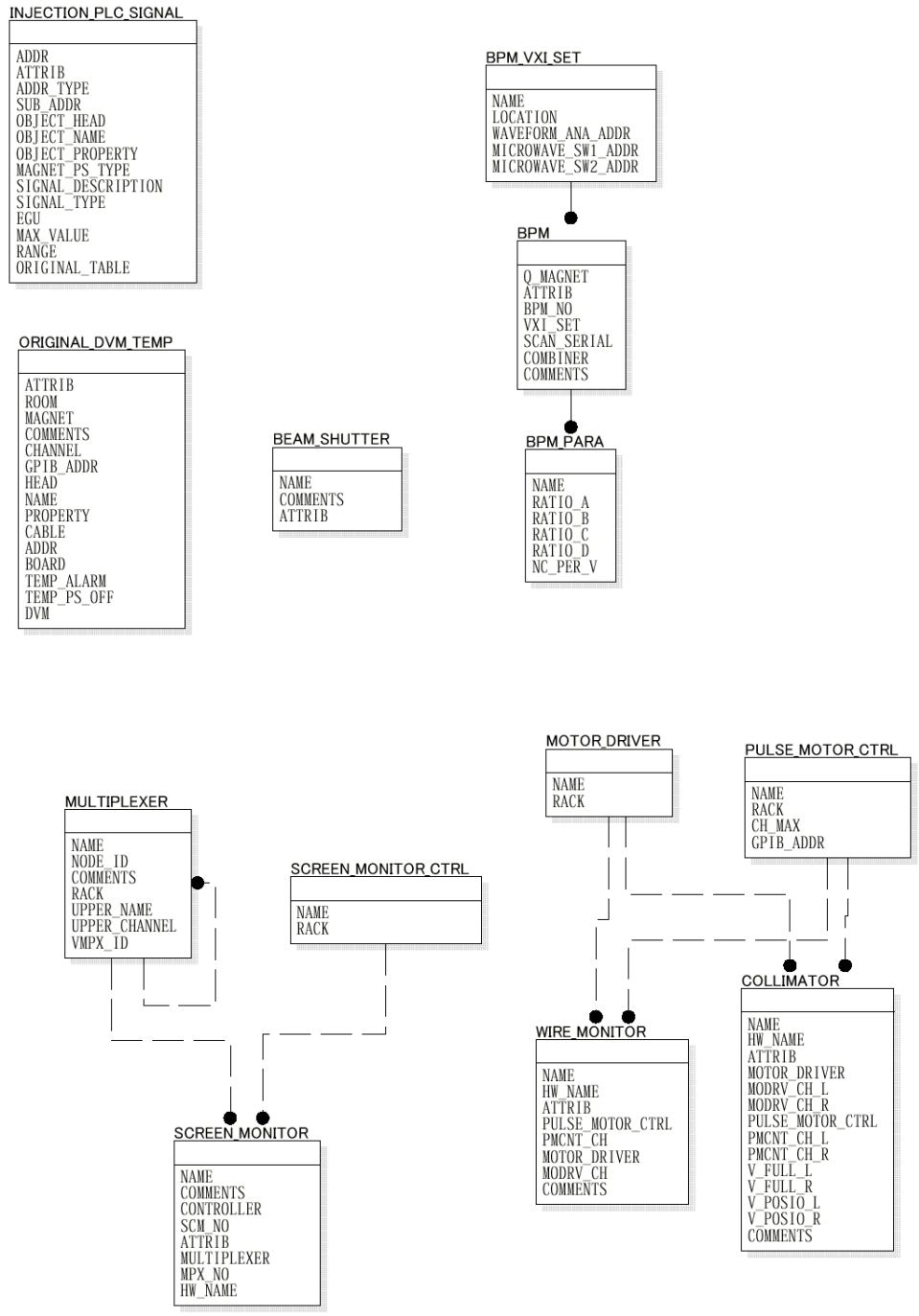
ER diagram : Wiring Database



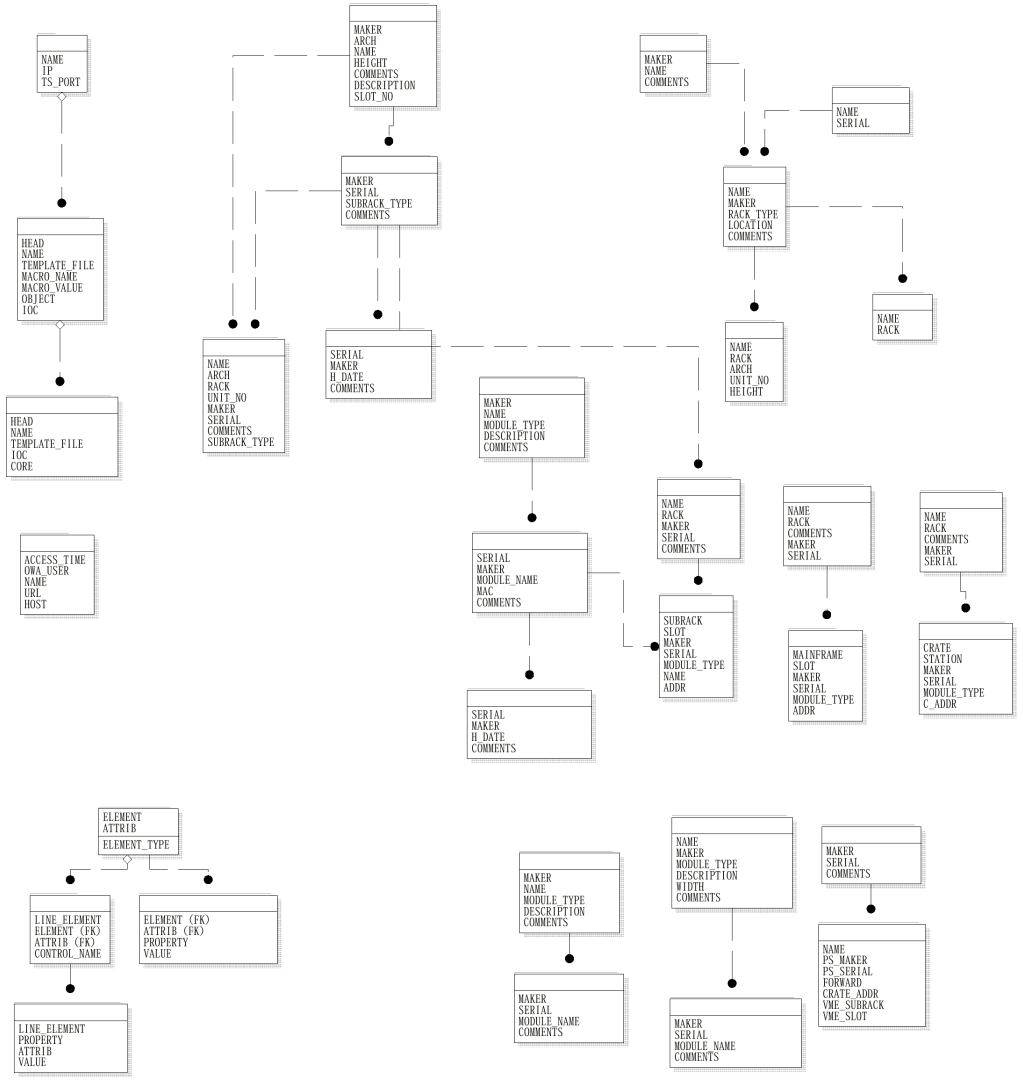
ER Diagram : Beam Instrumentation



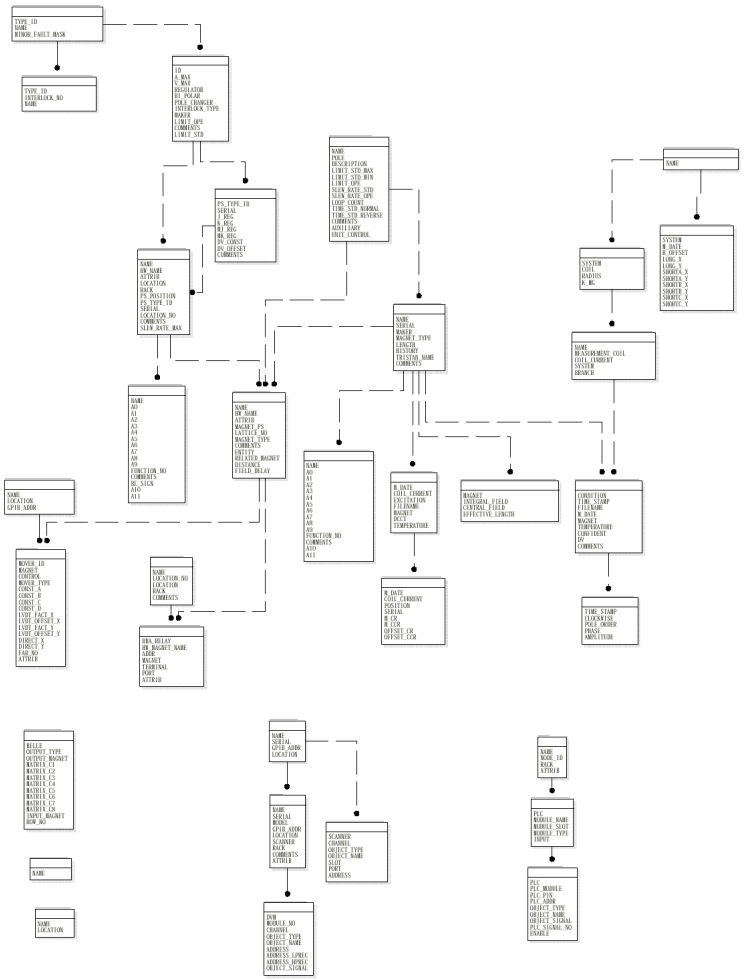
ER Diagram : Beam Transport



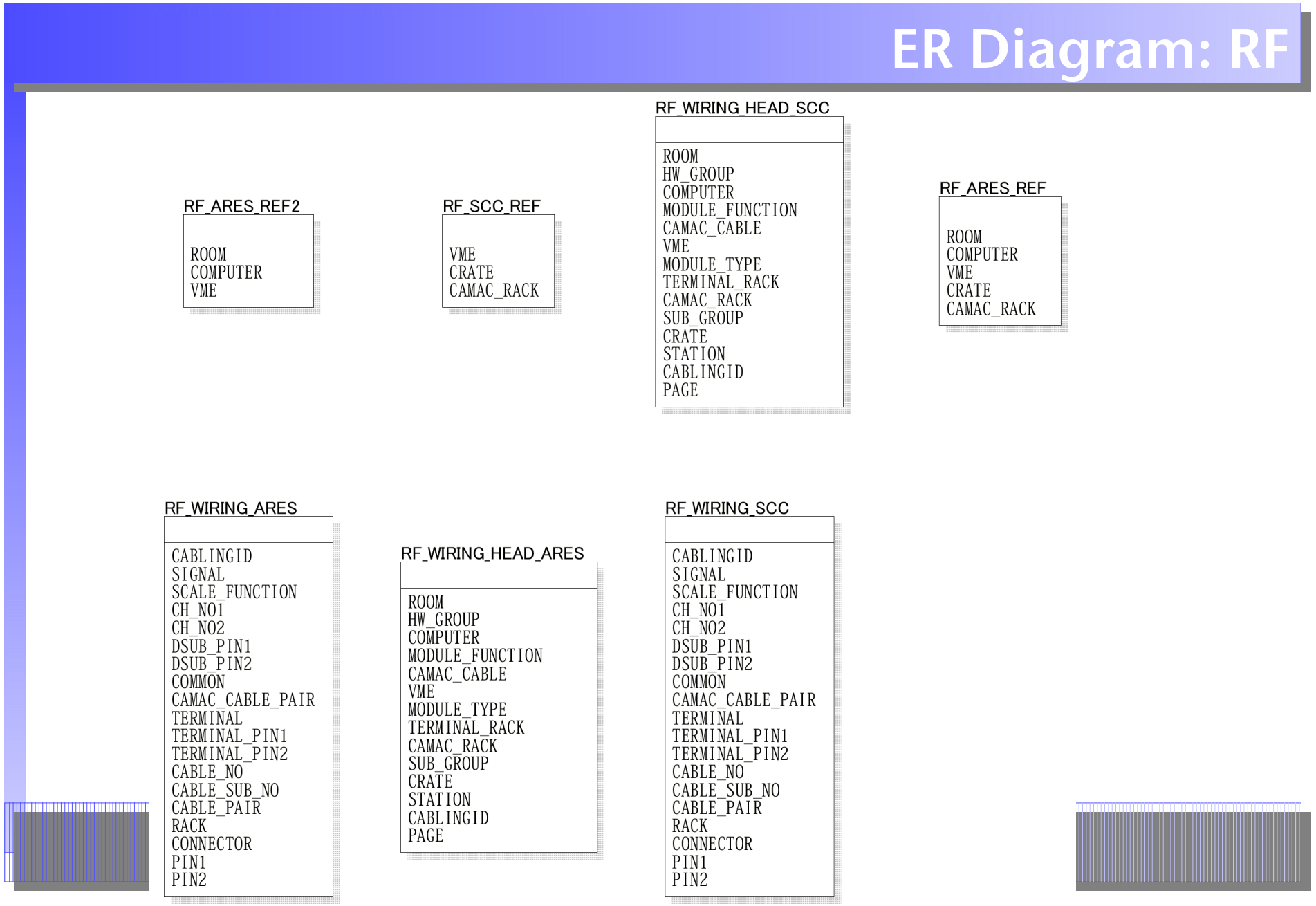
ER Diagram : Control



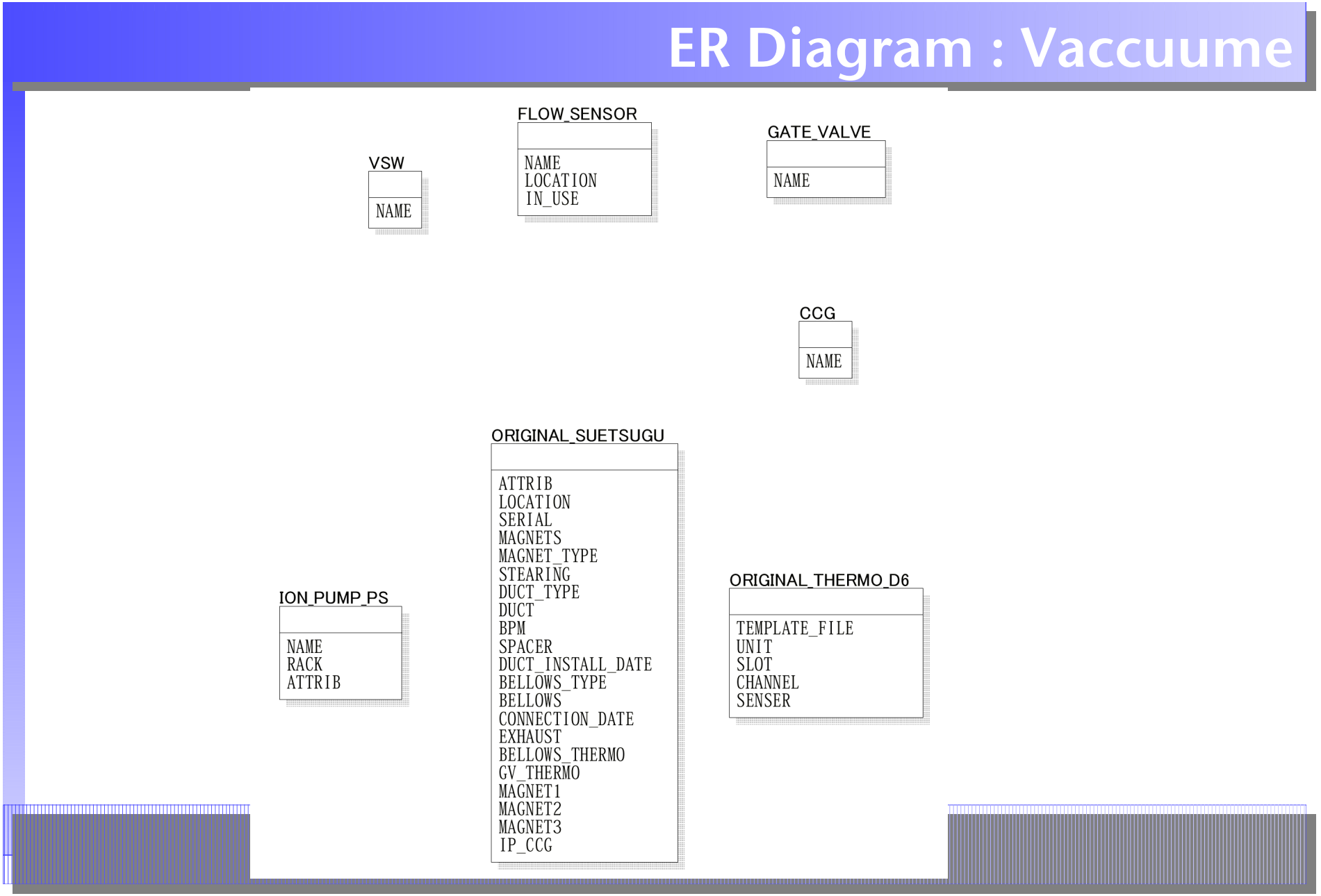
ER Diagram : Magnet/Power Supply



ER Diagram: RF



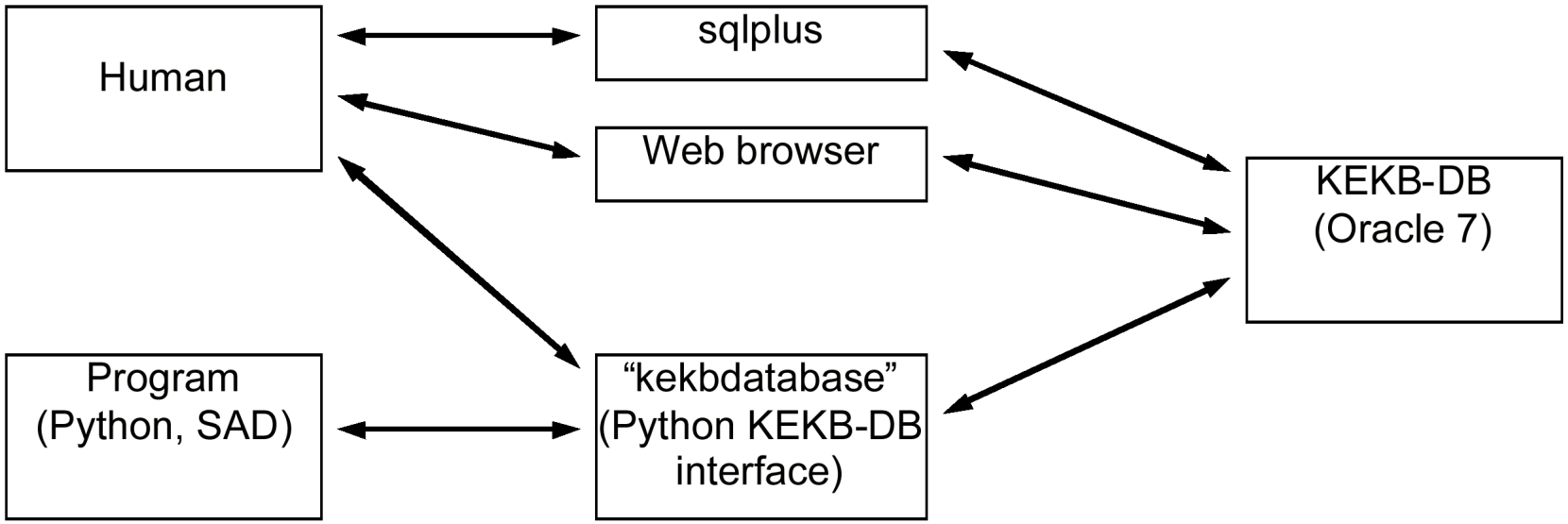
ER Diagram : Vaccuume



Data Source

1. In the most of cases, Original data came from Hard Ware group as Excel file.
2. WWW interface was developed to maintain data on RDB, but rarely used actually.

4. Access from high-level applications



kekbdatabase - Python KEKB-DB interface module

Using this module, whole table can be read easily

```
table = kekbdb("MG.MAGNET_PS_ALL")
```

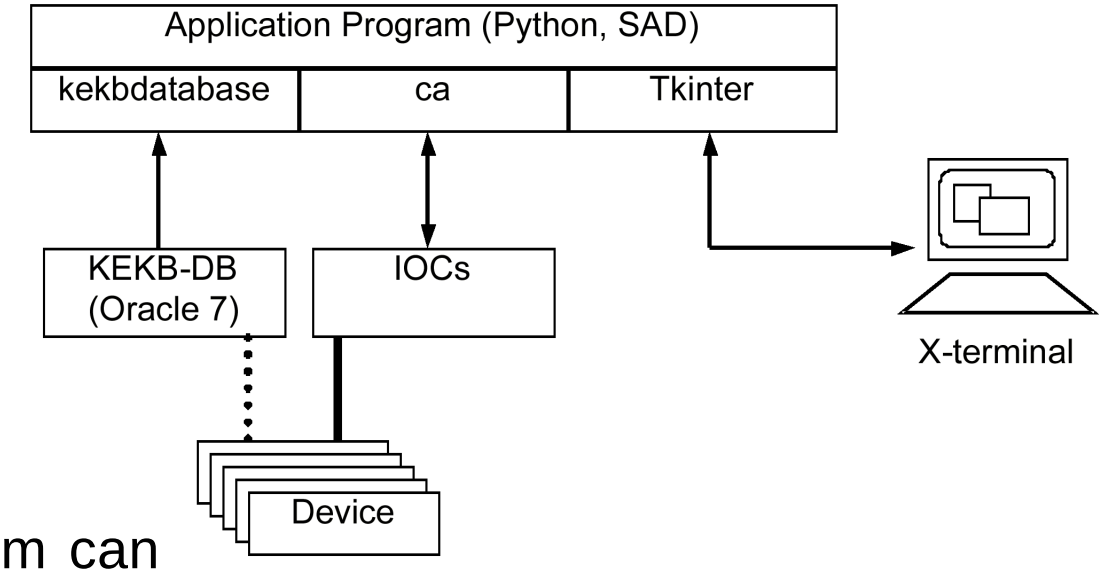
Any processed data can be read with arbitrary SQL statement

```
table = kekbdb("select * from MG.MAGNET_PS_ALL
where NAME='QC1RE'")
```

table is essentially an array (column x row).
But it is also a class instance with useful methods.

Data Access methods from an application

Communication methods between the application program and its oute world



The application program can
read **static properties** of the devices through **RDB**
and
read/write **dynamic properties** of the devices through **CA**.

Summary

1. Maintenance of the device parameters is concentrated on the KEKB-DB.
2. The parameter files for the configuration of the EPICS process DB are automatically generated from KEKB-DB by PL/SQL scripts.
3. KEKB-DB is also accessed from the high-level application programs written in Python or SAD.