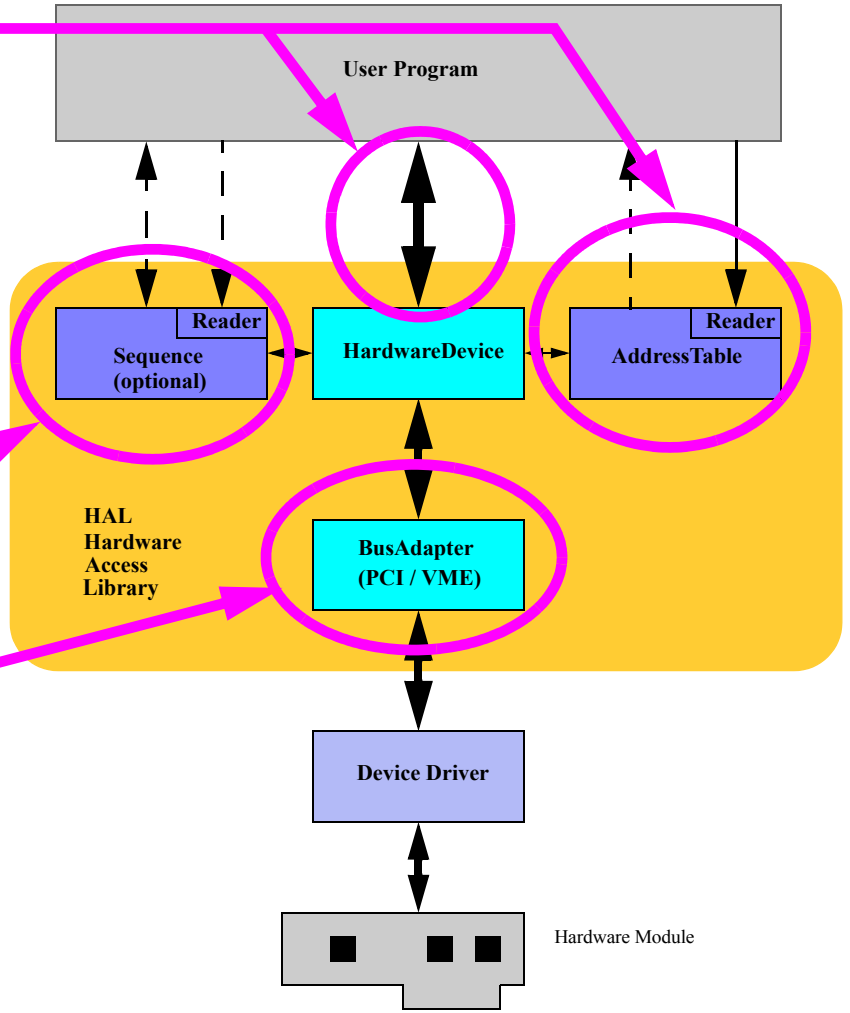
Recent HAL developments

- HAL review
- Database access
- VME64x configuration
- Pending items

HAL review

- Purpose:
  - Abstraction of Hardware Access in general API
  - User friendly API and “easy to read / maintain” code
    - concept of item, identified by strings
  - Higher- (better: medium-) level functionality
    - Bitfields, automatic checks
  - Hardware debugging facilities
  - Driver independent user code
    - concept of BusAdapter class
  - Technology independent user code
    - PCI or VME

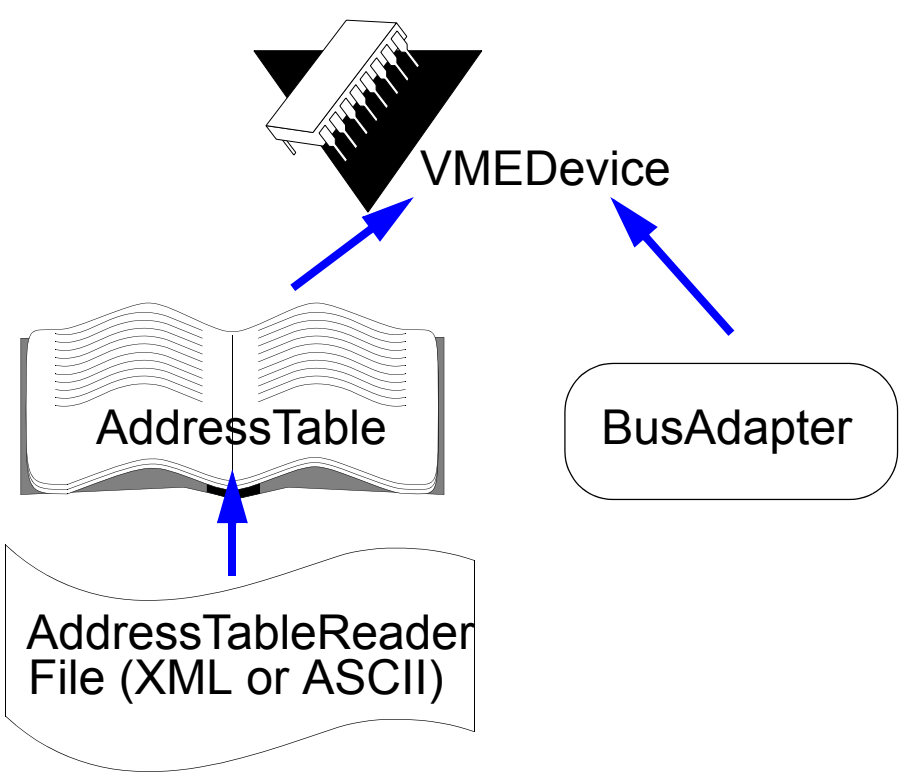


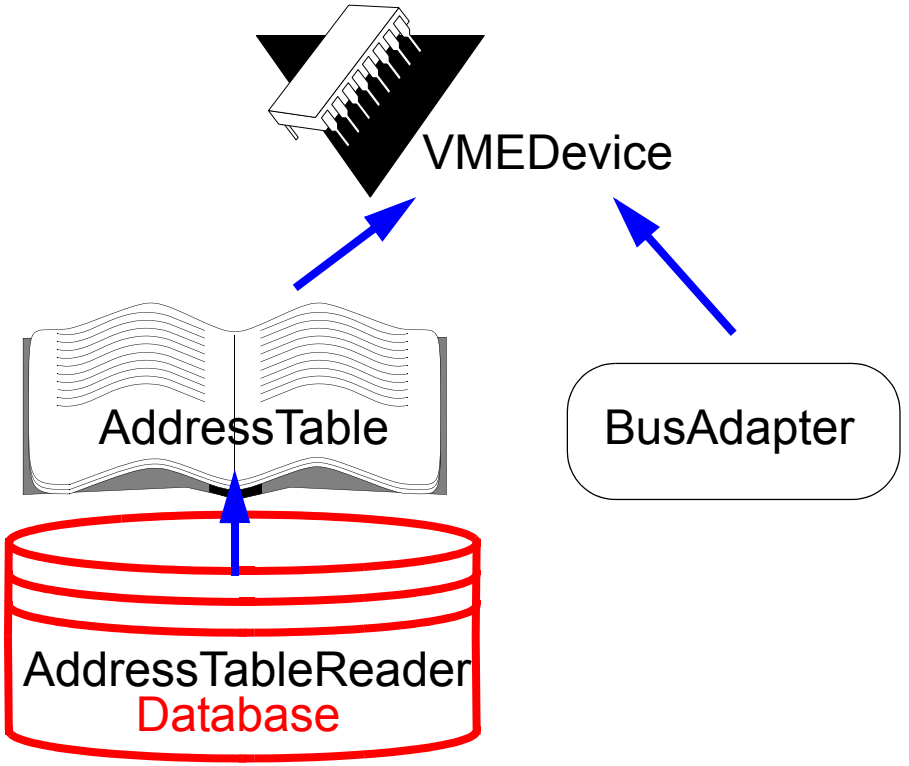
Use of the HAL in CMS

- Hardware debugging
  - Fast software development
  - Software independent of some hardware details
    - addresses are stored in configuration files
    - masks are stored in configuration files
    - changes in addresstable can be made without recompile
  - A sequencer allows to write small mini scripts
    - Allows to try out complex configuration procedures (example: setup up chained DMAs)
    - can be changed and reexcecuted without recompile
- XDAQ Environment
  - Configuration and startup of hardware equipment
  - Monitoring the status of hardware equipment during running
  - Change of running paramenters
  - Local DAQ / Test-beam : local (“medium speed”) readout of data via VME (block transfer)

Database Access

- So far :
  - AddressTables for HAL come from file (ASCII or XML format)
- New:
  - AddressTables can be read from Database (Oracle tested so far)



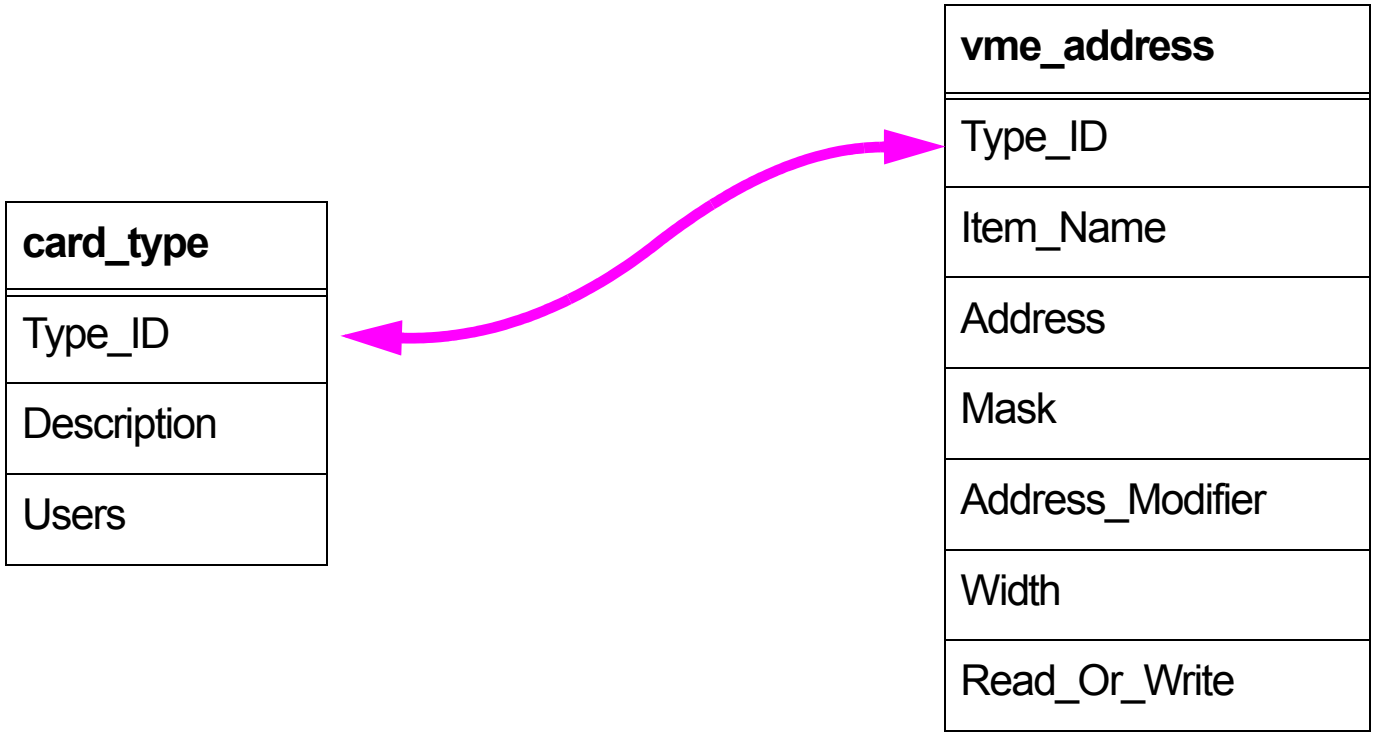


Database Access: Details

- Uses XDAQ Data-store (Dstore) application (Frank Glege)
  - Dstore performs access to database
  - Dstore provides mapping XML <--> SQL-tables
  - Allows to access ORACLE and MYSQL databases
- Consequences for HAL:
  - XML file format will change completely
    - XML format is defined by DStore application
    - it does not make sense to keep 2 XML formats for the same scope
  - Scripts will be provided
    - Scripts to generate ORACLE tables
    - Conversion script:
      Input: Printout of AddressTable (“print” - method) or ASCII address table
      Output: XML AddressTable in new format
      SQL script with commands to generate Oracle table entries

Database Table Format

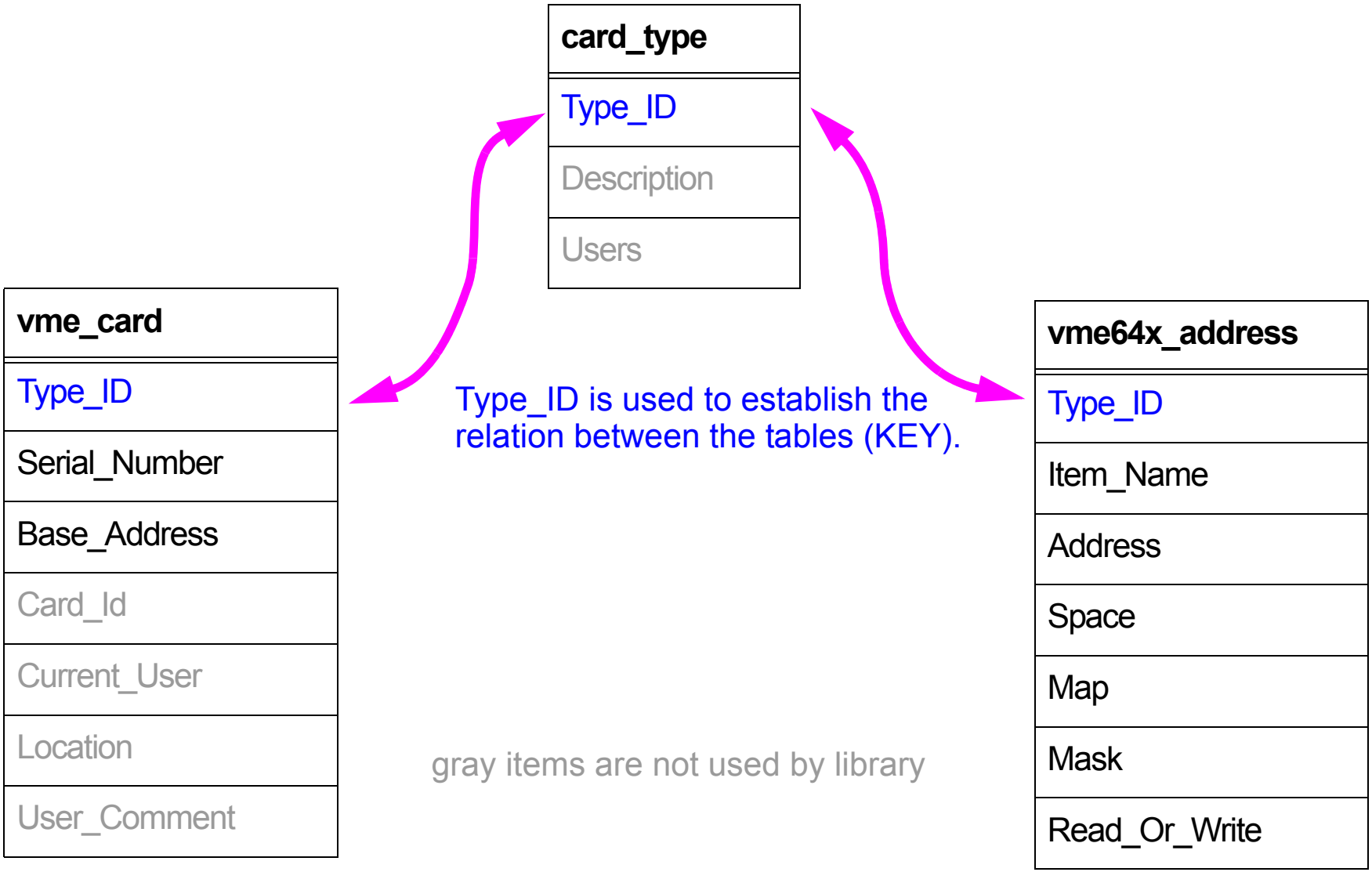
- More entries can be added



VME64x Plug and Play configuration

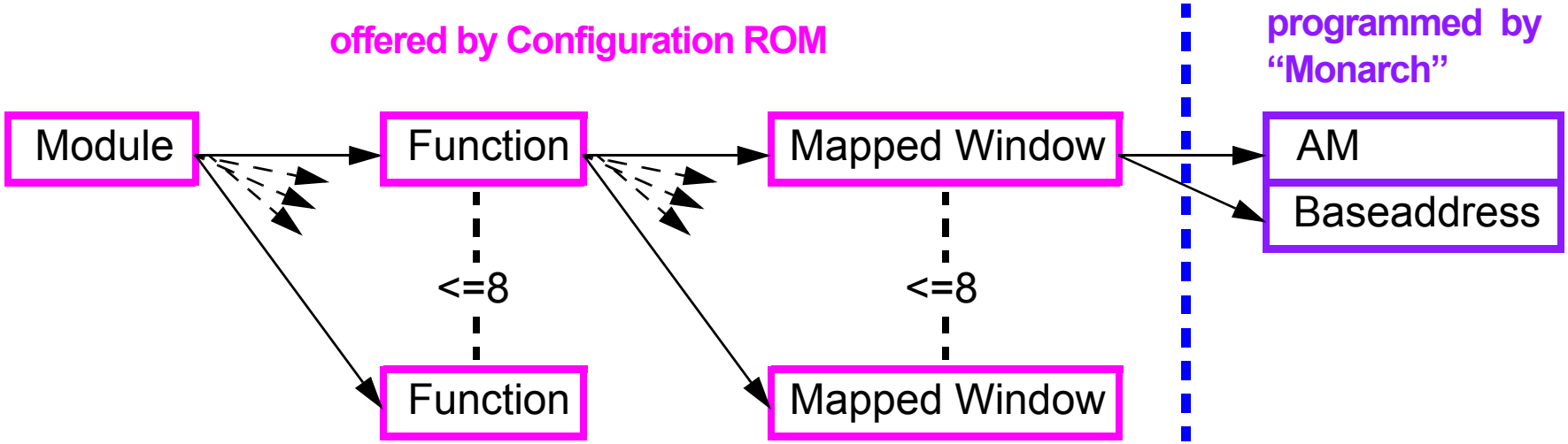
- It has been proposed to design custom modules in CMS according to VME64x to allow for “Plug and Play”.
  - document available at http://cmsdoc.cern.ch/~cschwick/VME/index.html
- Software to support Plug and Play is now under development
  - As add-on package to HAL.
  - Needs to run in XDAQ environment in order to access database.
- Features of the package:
  - Compatibility with “Standard VME” modules and “mixed” systems
    - Information on non - plug and play modules are given to the library before configuration
  - So far support for A16, A24 and A32 addressiing and D8, D16 and D32 data width.
  - Serial Number in Configuration ROM is used to retrieve data base information
  - AddressTables are automatically retrieved.
  - Checks consistency between AddressTable and configuration ROM entries.
  - Automatic mapping of the address space.

VME64x: Database Table Structure



VME64x : steps to configure a crate

1) read static configuration of “Standard” VME Modules
2) check if there is a module on the “Amnesia address” 0x1e
   (throws exception if module found)
3) probe each slot without static configuration for VME64x module
   - check Identification bytes and VME-version
   - verify the checksum of the Configuration ROM
   - read out serial number and retrieve module TypeId from database
   - read out address space requirements and addressing capabilities
4) Map the address space of all VME64x modules found
   - consider the static mapping of standard VME modules.
   - choose addressModifier using a simple priority scheme.



Standard VME : A mapped region is addressed with various AMs

VME64x : A mapped region is addressed with a single AM

5) Enable the VME64x modules

- this makes the VME64x modules start answering to VME requests

6) Obtain VMEDevices form configuration:

- AddressTables are retrieved from Database
- AddressTables are checked against the mapped address space (for consistency)
- VMEDevice is constructed and returned

Pending Items

- Concept of a channel
  - Often VME Address Tables contain repetitive structures (n - channel ADC, TDC, ...)
  - Would be nice to have the concept of a channel in the HAL.
    - Eases loops over channels for setup procedures
    - Avoids using offset (unsafe, error prone)

Please comment if you have suggestions or special needs.

- # Review of the Sequencer

  - Existing Sequencer
    - Only deal with one Hardware Device in one sequence
    + Sequences can be called from C++ application (e.g. XDAQ application)
  - Possibility to incorporate HAL in TCL/Tk.
    + A prototype has been built (by Akos) and is being used by Janosh (Global Trigger)
    + Allows to use full TCL scripting
    + Allows to write scripts which handle more than one VME-device
    - Can NOT be called from C++ application (TCL/Tk is the Main Program)
  - An XML scripting facility has been devloped by Ildefons and is currently under test
    o Results of the evaluation will be available in a month

  May be the sequencer becomes obsolete, but something at least as powerful will be delivered.

  Is this inconvenient for the users ? (Are there existing huge sequences?)

  Please give feedback if you have needs, ideas concerning the sequencer.