

# Drivers and Software for MicroTCA.4.

M. Killenberg, M. Heuer, M. Hierholzer, L. Petrosyan, C. Schmidt, N. Shehzad, G. Varghese, M. Viti (DESY, Germany), S. Marsching (aquenos GmbH, Germany), M. Mehle, T. Sušnik, K. Žagar (Cosylab, Slovenia), A. Piotrowski (FastLogic, Poland), T. Kozak, P. Prędko, J. Wychowaniak (Łódź University of Technology, Poland)



## MicroTCA.4 Technology

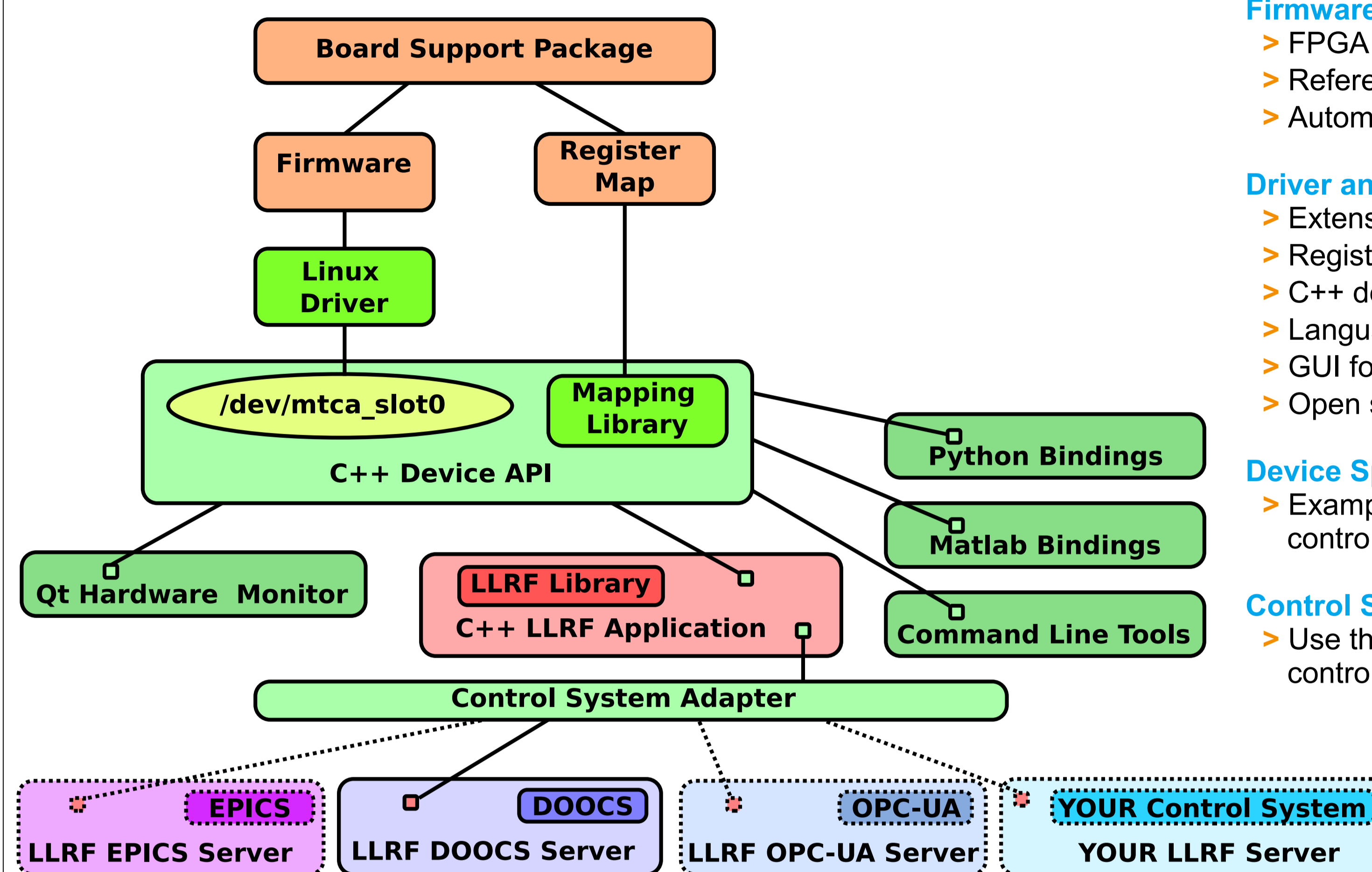
**Based on Advanced Telecommunications Computing Architecture (ATCA)**  
Widespread use in telecommunications since 2005

- > High speed serial bus topology
- > High modularity due to Advanced Mezzanine Cards (AMCs)
- > High availability due to redundancy
- > Reduced down-time due to hot-swap capability

## MicroTCA.4 Enhancements for Rear I/O and Precision Timing

- > Definition of Micro Rear Transition Modules (μRTMs)
- > Definition of AMC-μRTM connection
- > Radial clock lines for precision timing
- > Low latency point to point serial I/O
- > Advanced shelf management
- > High signal integrity by separation of analog and digital processing

## The DESY MicroTCA.4 User Tool Kit (MTCA4U)



### Firmware Board Support Package

- > FPGA abstraction layer
- > Reference firmware with demo application code
- > Automated generation of register map

### Driver and Basic Tools

- > Extensible universal driver
- > Register mapping library
- > C++ device API
- > Language bindings to Matlab and Python
- > GUI for convenient register monitoring/setting
- > Open source

### Device Specific Applications

- > Example: Low Level Radio Frequency (LLRF) control application for accelerators

### Control System Adapter

- > Use the same application code with multiple control systems

## Use Case

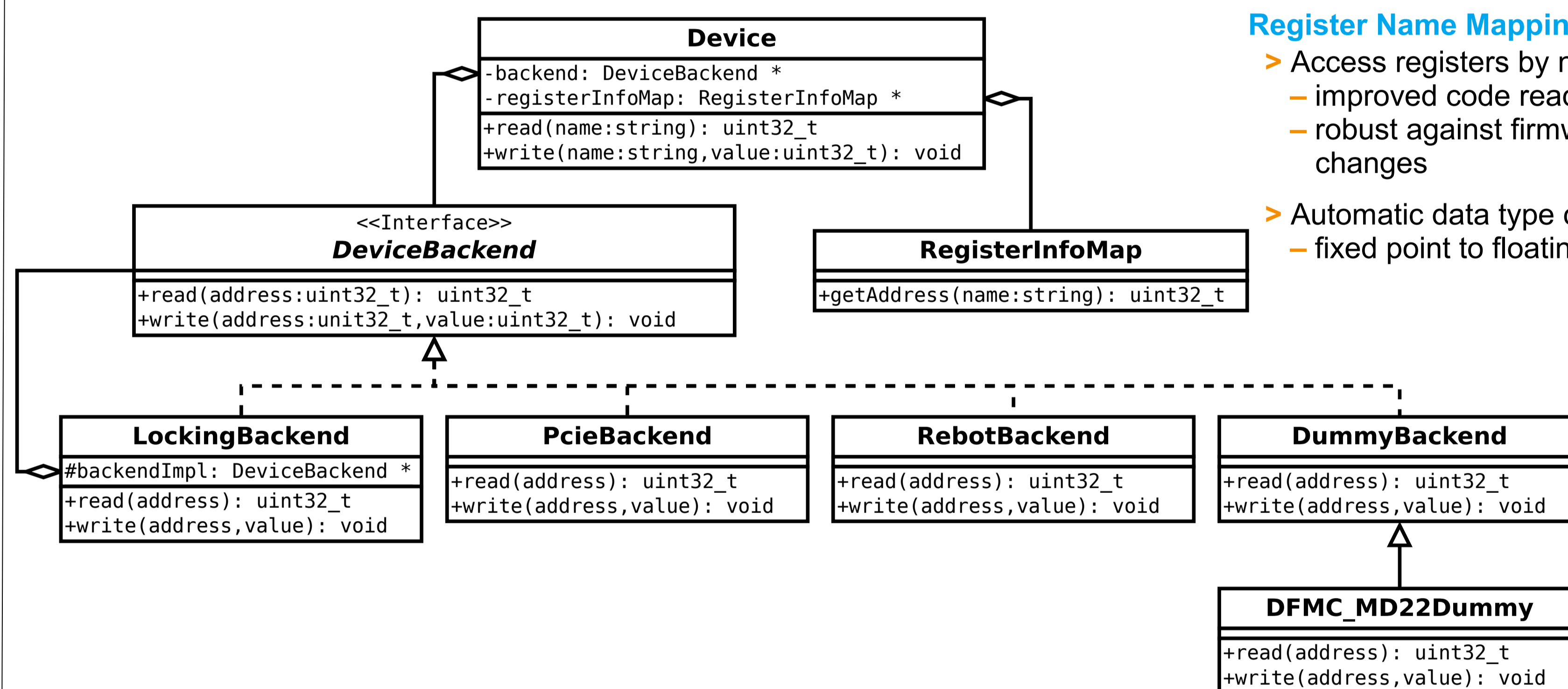
### Low Level Radio Frequency Control at the European XFEL and FLASH

- > Superconducting accelerators provide multi-GeV electron beams for Free Electron Lasers (FELs)
- > Digital low level radio frequency (LLRF) control based on MicroTCA.4
- > Pulsed operation (10 Hz)



MicroTCA.4 LLRF installation in the FLASH accelerator tunnel

## The C++ Device API



### Register Name Mapping

- > Access registers by name
- improved code readability
- robust against firmware changes
- > Automatic data type conversion
- fixed point to floating point

### DeviceBackend

- > Abstract interface
- > PCI Express
- > Register-based over TCP (Rebot)

### Back-End Factory

- > Automatically determine the type
- > Plugin mechanism
- add new back-ends at run time

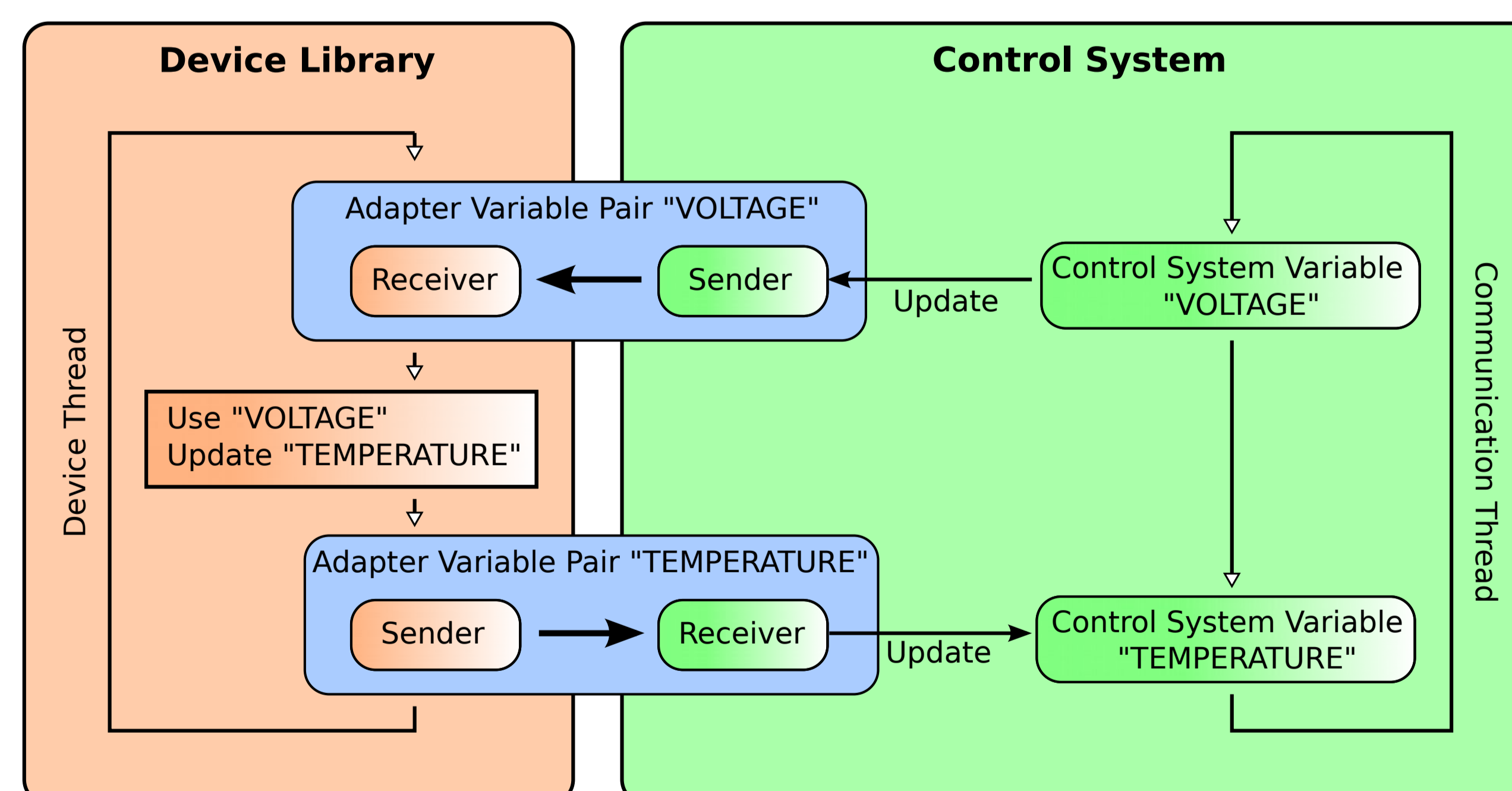
### DummyBackend

- > Simulate I/O address space in RAM
- > Callback functions on read/write
- implement firmware mock-ups

## Qt Hardware Monitor

- > Lists all hardware registers
- > Register names and properties
- > Read and modify register content
- > Basic plotting functionality

## Control System Adapter



### Task

- > Keep the application code (device library) independent from the control system
- > Minimise device-dependent code on the control system side

### Requirements

- > Thread safety
- > Real-time capability
- > Do not copy large data objects

Subversion Repository: <https://svnsvn.desy.de/public/mtca4u>