



SoC Developments for the Detector Control System of ATLAS Tile Calorimeter at the HL-LHC



Mpho Gift Doctor Gololo, Fernando Carrio Argos
University of the Witwatersrand, Instituto de Fisica Corpuscular
On behalf of South Africa and Valencia TileCal Upgrade group

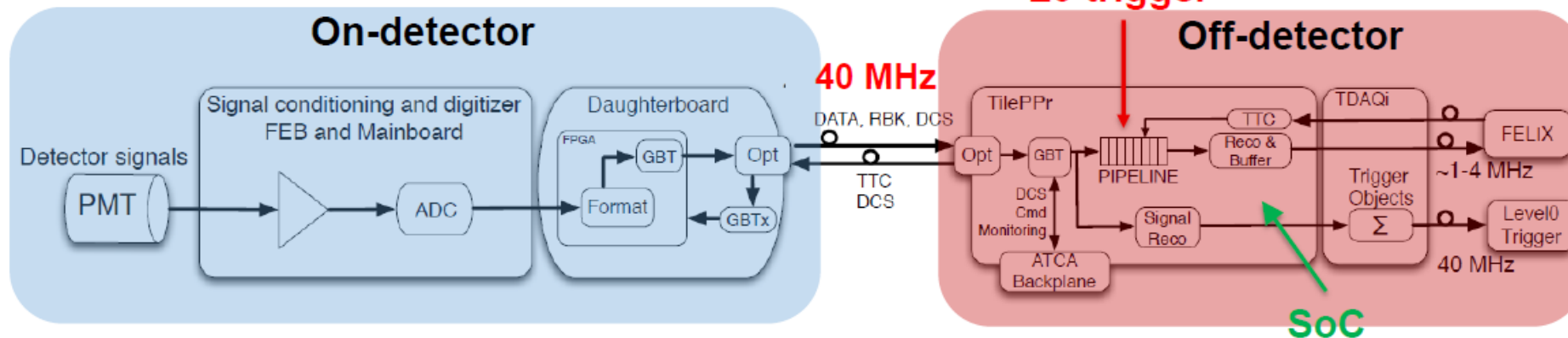
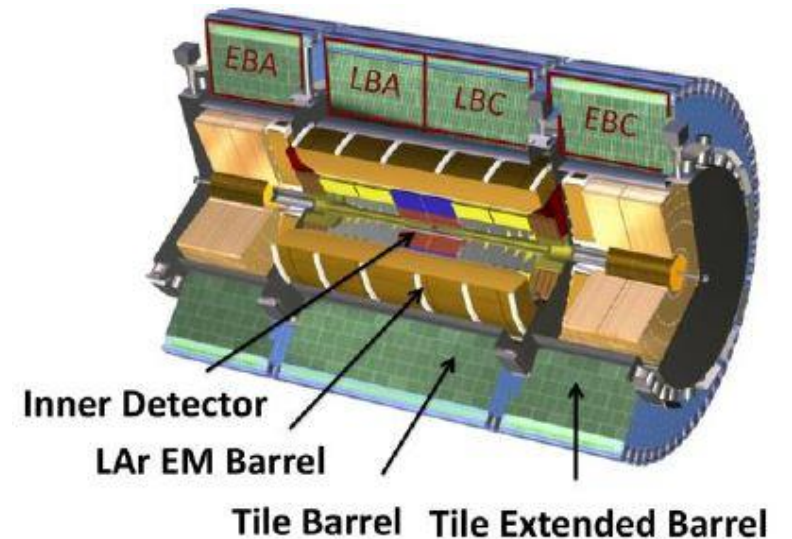


Contents

- ATLAS TileCal Phase II Upgrade
- Full-size Pre-Processor
- ATCA carrier base board
- TileCoM PCB production
- Software developments on TileCoM
 - OpcUaXADC developments as a generic project
 - OpcUaTileCoM developments specifically for the Tile Group
 - Ironman Ipbus development for the TDAQ interface
- Conclusion

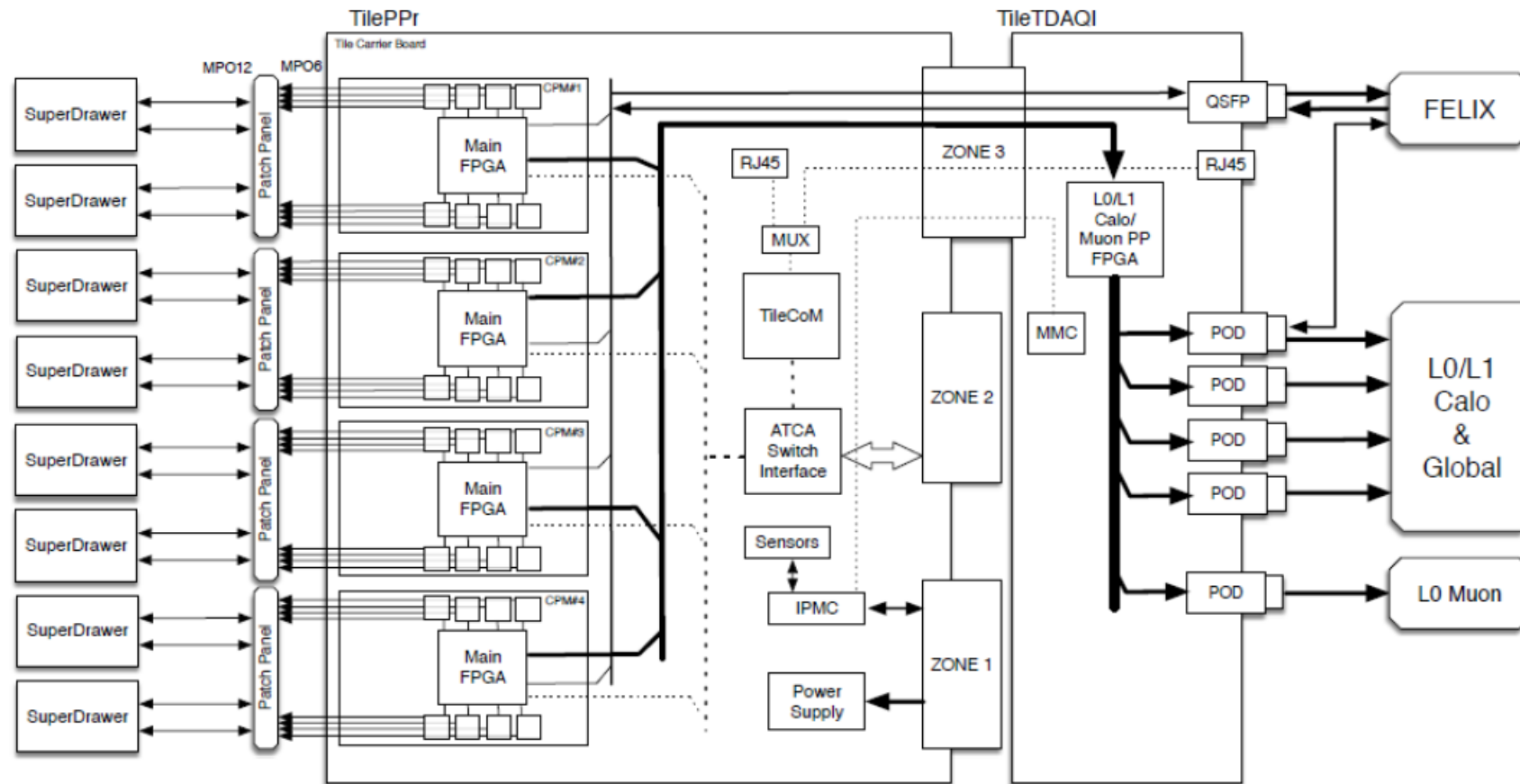
ATLAS TileCal Phase II Upgrade

- Complete replacement of on-detector and off-detector readout electronics
 - Aging of electronics due to time and radiation
 - Current readout system is not compatible with Phase II TDAQ architecture
- New readout strategy for HL-LHC
 - On-detector electronics will transmit digitized data to the off-electronics at the LHC frequency
40 Tbps to read out the entire detector!



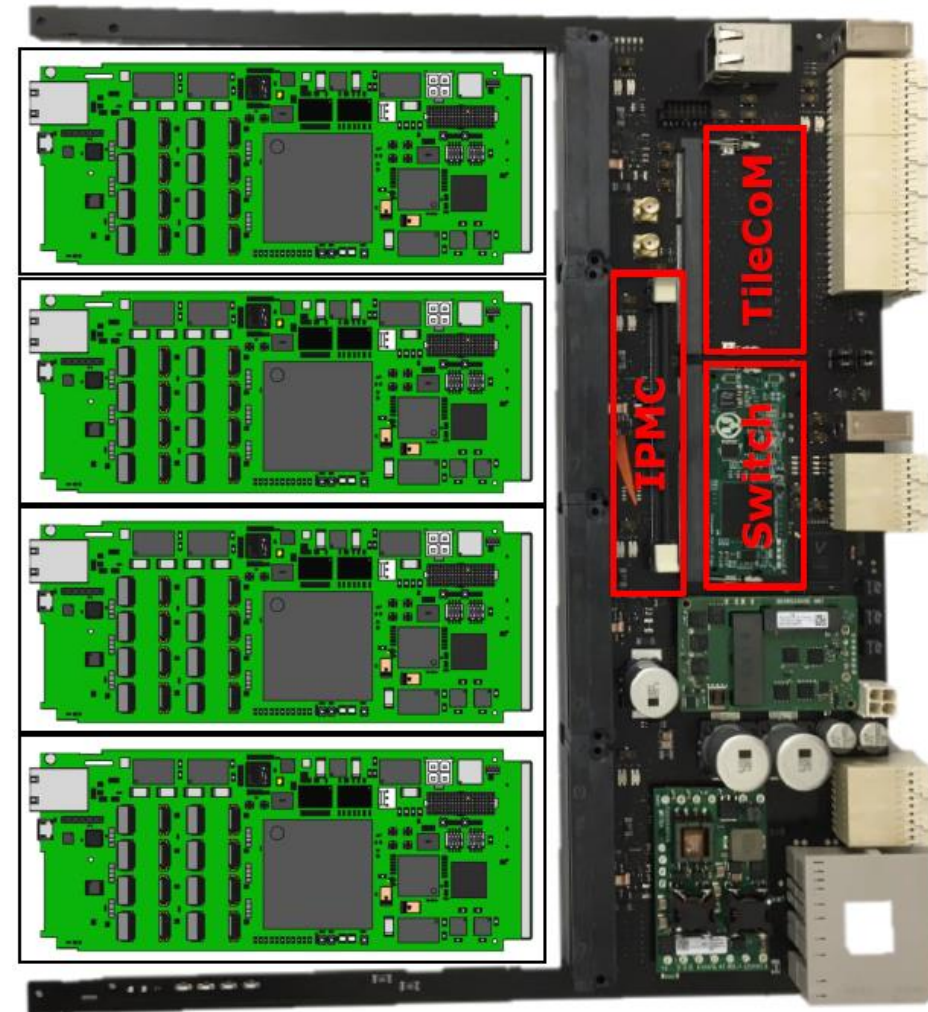
Full-size Pre-Processor

- **32 PPr boards** in ATCA format: ATCA carrier + 4 Compact Processing Modules
- **32 TileTDAQ-I**: Preprocesses trigger data and interfaces with L0Calo, L0Muon, Global and FELIX system



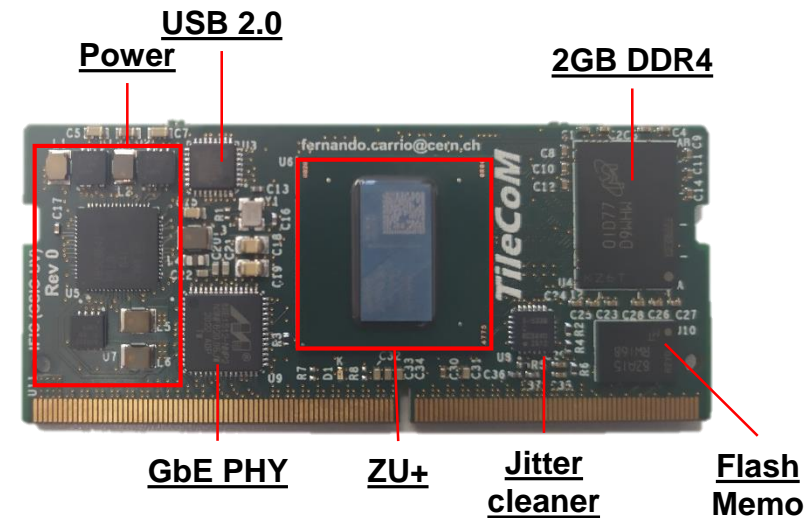
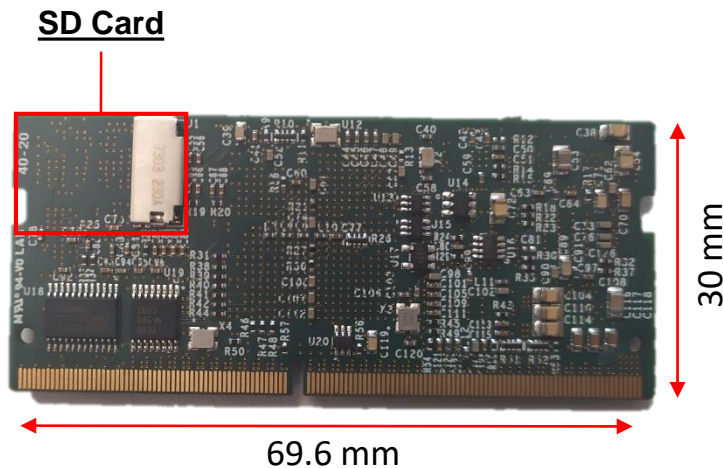
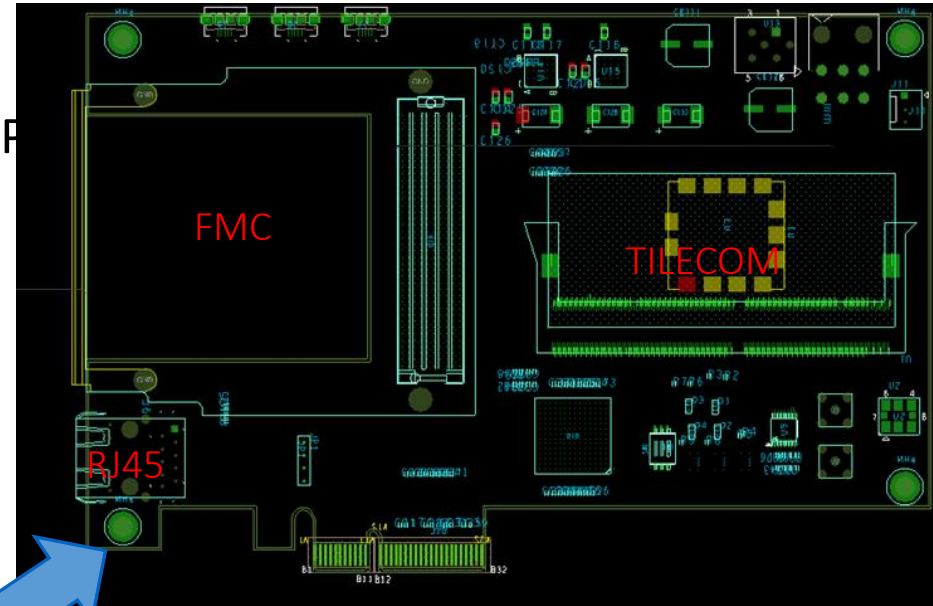
ATCA carrier base board

- Zone 1: Power distribution to CPMs and TDAQ-I
 - Max power of 400 W
- Zone 2: GbE + XAUI 10G
 - Base & Fabric
- Zone 3: Communication
 - Between CPMs and TDAQ-I
- Three on-board mezzanines
 - CERN IPMC board
 - TileCoM board
 - 16 GbE port switch module



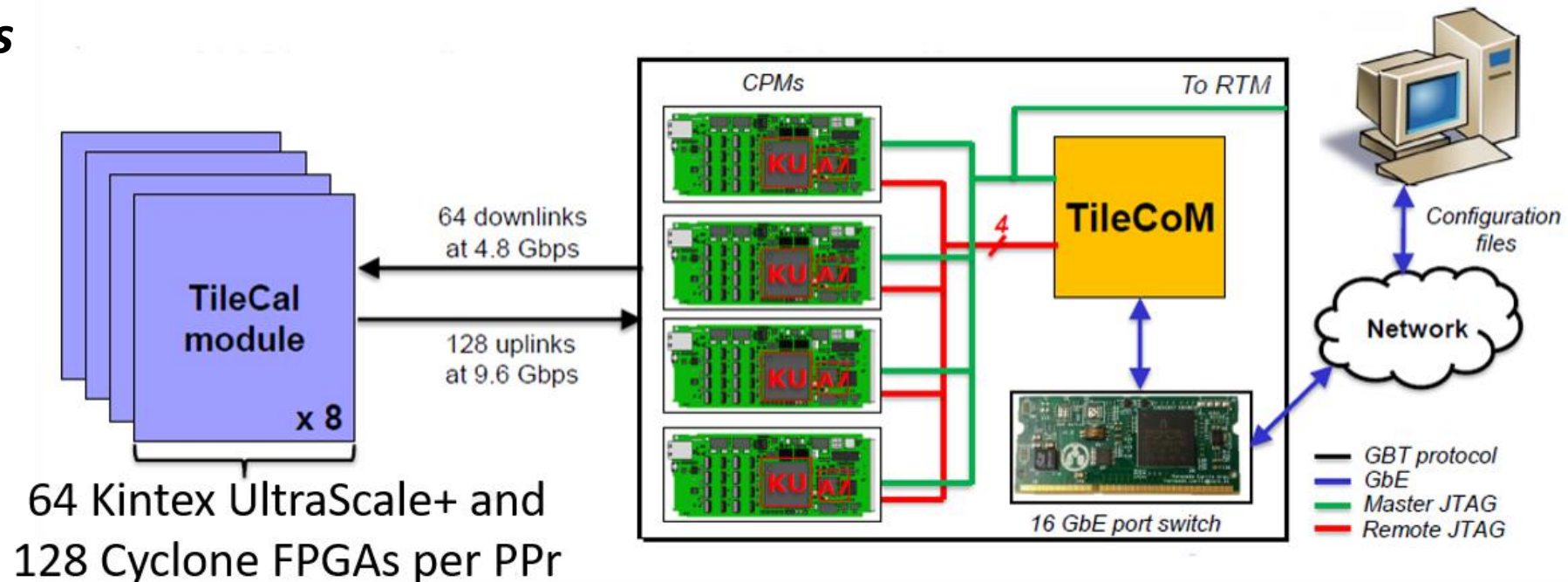
TileCoM Hardware Components

- Main components:
 - Zynq UltraScale+ ZU2CG, 2GB DDR4
 - SD card, GbE, USB 2.0, SPI, I²C, UART, 82 x GF
 - PCB with 10 layers (1.2 mm), FR4
- Test board to validate all functionalities
 - Layout stage. PCB to be produced in Feb
 - All GPIO routed to a HPC FMC connector
 - Emulates carrier's clocking schema
 - Ethernet RJ45 (DCS interface)
 - USB, UART, PCIe x4 ports



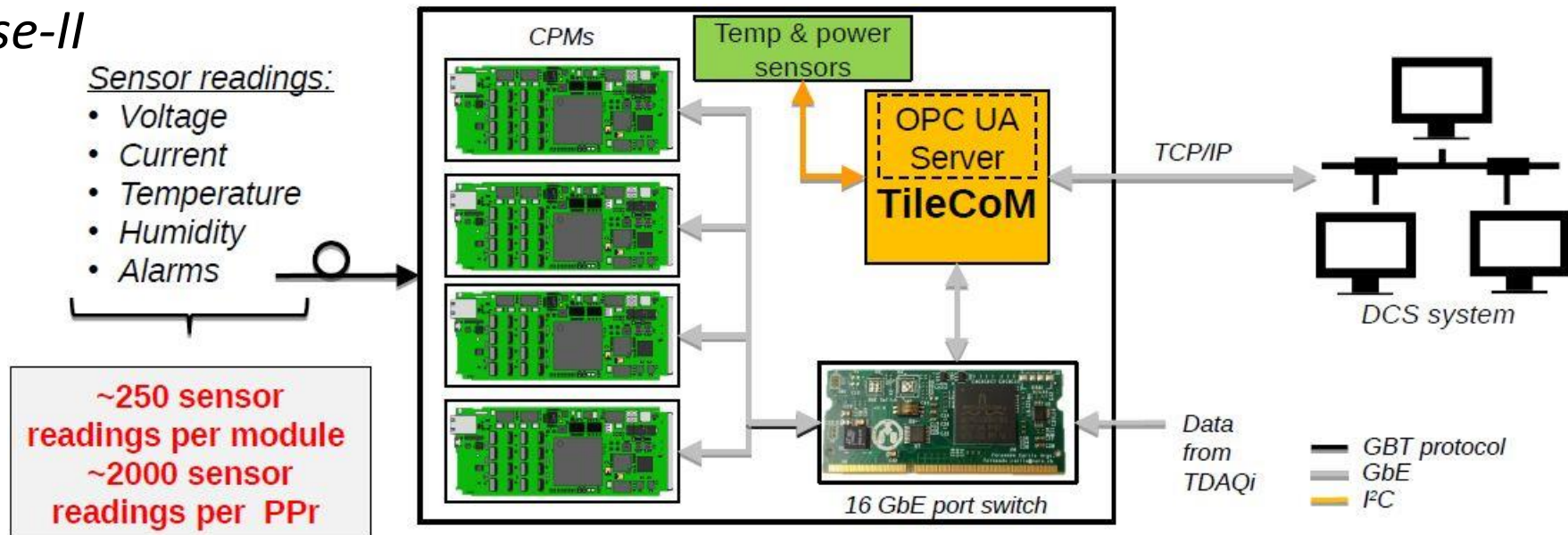
Functionality #1: FPGA programming

- Configuration of CPM, TDAQi and on-detector FPGAs
 - **4 Remote** JTAG chains for on-detector KU+ and Cyclone IV FPGAs
 - **1 Master** JTAG chain for the 4 CPMs + TDAQi
- Two options under evaluation
 - Implementation of Xilinx Virtual Cable IP cores
 - Custom JTAG controller without TDO check (mods to disable TDO checkings)
- ***Update the entire detector electronics configuration at the same time in few minutes***



Functionality #2: PPr and DCS system

- Interface with the DCS system through dedicated GbE port
 - TileCoM runs an OPC UA Server as middleware
- Sensor monitoring reading from ATCA carrier, RTM and on-detector electronics
 - Power consumption, optics diagnostics, temperature sensors, ...
 - I²C interface for reading local ATCA carrier sensors
 - CPMs retrieve sensor data from on-detector through GBT links
 - TileCoM reads out sensor data from CPM and RTM through GbE (IPbus)
- *Control and monitoring of the HV distribution (HV Opto) → not baseline for Phase-II*

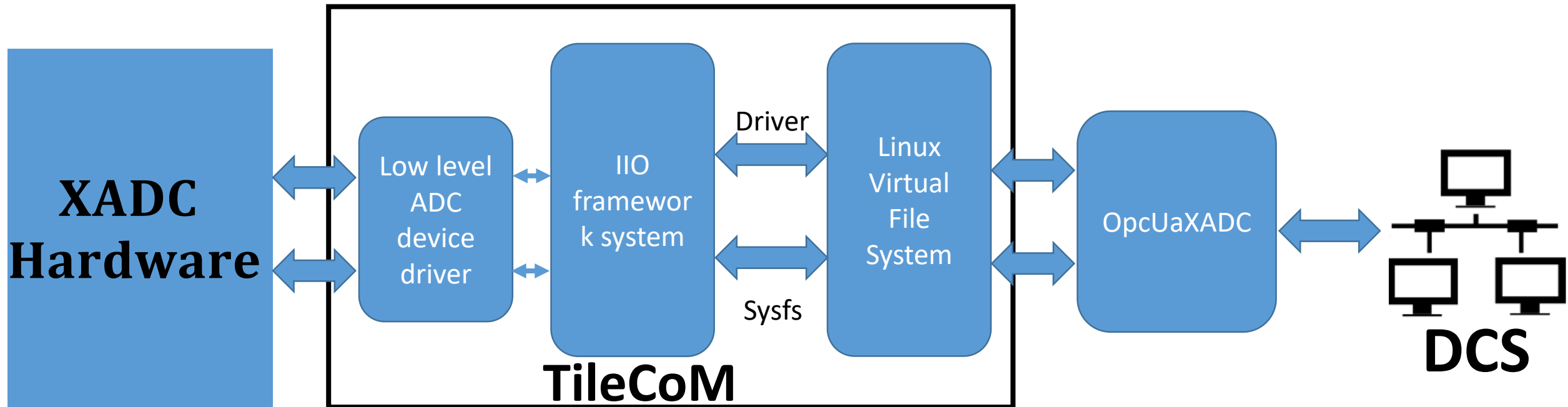


Functionality #2: PPr and DCS system

- This Functionality builds a server on the TileCoM board to read sensors through the GbE
- Focus is directly on the off-detector electronics before on-detector electronics
 - PPr FPGAs – 4 CPMs that connects to the TileCoM through the GbE
 - TDAQi FPGAS – 4 Kintex FPGAs that connects to the TileCoM through the GbE
- This functionality is divided into two projects using two different methods to interact with sensors
- Each class represent a sensor on the board current being monitored:
 - Following slide shows different classes that represent different sensors
 - For the evaluation board, current focus is based on XADC
 - Voltages, Temperatures, Alarms and BoardHealthFlags
- Each class in quasar represents the sensor readings from the device
 - Approximately 2000 sensors per CPM to be read by this development

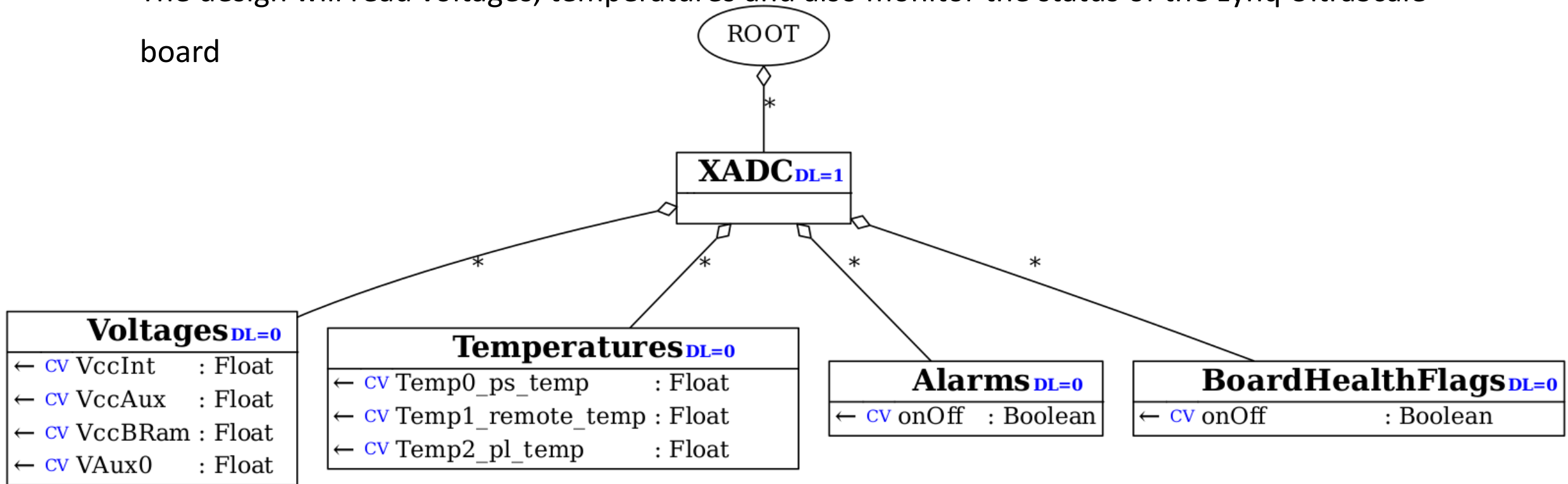
Functionality #2: PPr and DCS system – Project 1

- The first project only focuses only on XADC quasar server – client to readout necessary data
- This project will be made public and will be beneficial to most users with minimal design task
- The users of this project will only need to add variables on the configuration files



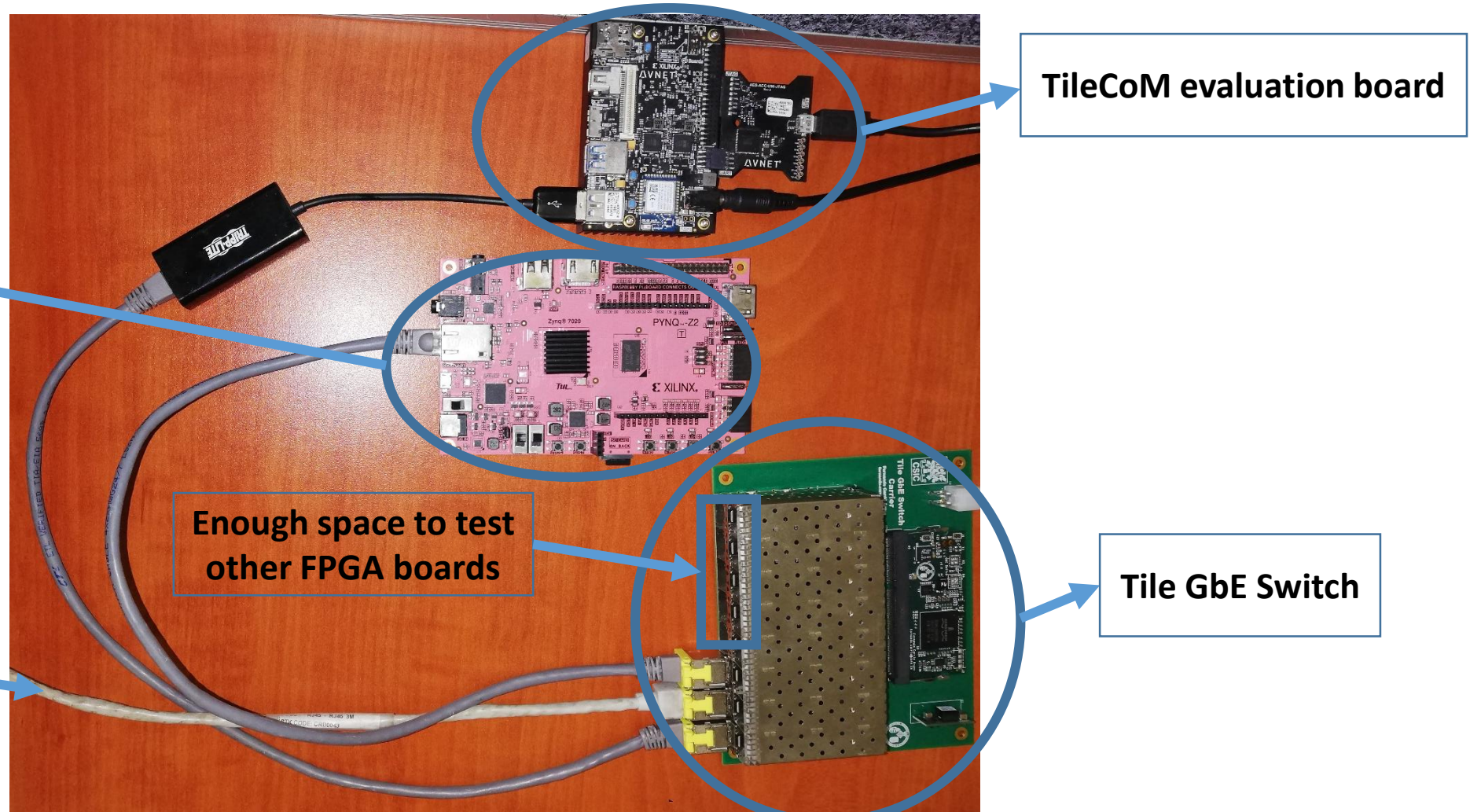
Functionality #2: PPr and DCS system – Project 1

- Design diagram for the TileCoM reading data from the XADC
- The XADC read data from the low level hardware using the IIO framework
- The design will read voltages, temperatures and also monitor the status of the zynq UltraScale board



Project 1: OpcUaXADC – Test station

- Test bench includes the TileCoM evaluation board, One Zynq-7000 board, Tile GbE Switch
- Other boards to be added – The server should read data from multiple boards through GbE



Project 1: OpcUaXADC – Test station

- The OpcUaXADC server has been deployed and tested on the TileCoM evaluation board
- <https://gitlab.cern.ch/mgololo/opcuaxadc>

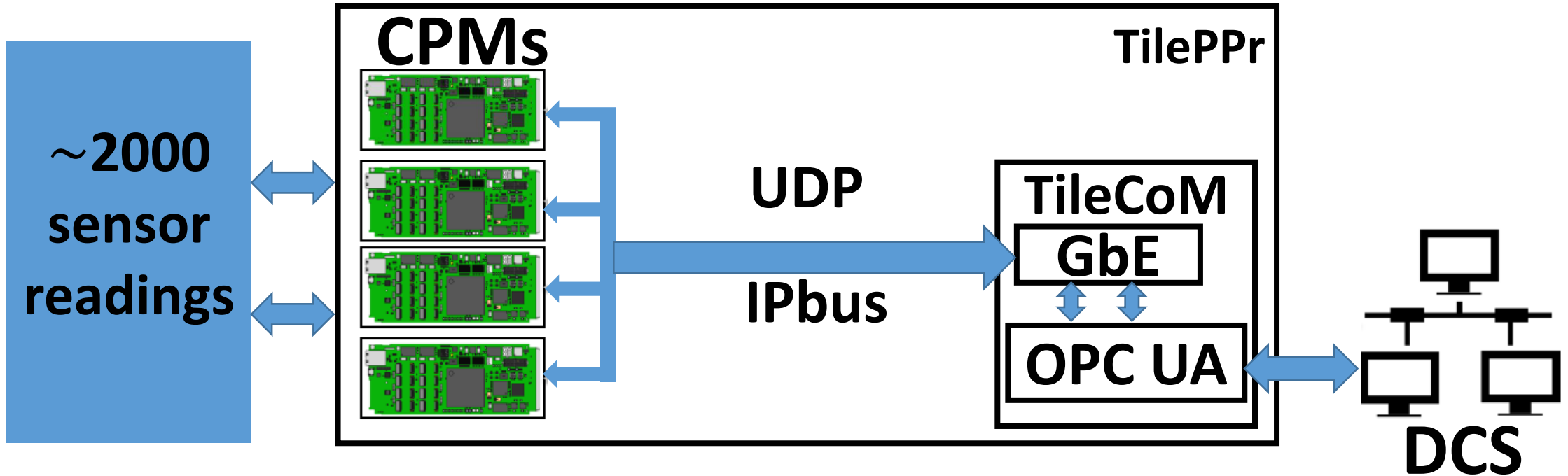
```
[2021-06-06 00:10:47.335 (UTC+0000)] info/network TCP network layer listening on opc.tcp://localhost.localdomain:4841/
2021-06-06 00:10.47.335718 [uaserver.cpp:71, INF] UA_Server_run_startup returned: Good (0x0), continuing.
2021-06-06 00:10.47.335936 [BaseQuasarServer.cpp:434, INF] *****
2021-06-06 00:10.47.336006 [BaseQuasarServer.cpp:435, INF] Press CTRL-C to shutdown server
2021-06-06 00:10.47.336070 [BaseQuasarServer.cpp:436, INF] *****
2021-06-06 00:10.48.336211 [DTemperatures.cpp:98, INF] TEMP_PS: 49
2021-06-06 00:10.48.336510 [DTemperatures.cpp:99, INF] TEMP_REMOTE: 49
2021-06-06 00:10.48.336623 [DTemperatures.cpp:100, INF] TEMP_PL: 49
2021-06-06 00:10.48.337790 [DVoltages.cpp:99, INF] VCC_INT: 1
2021-06-06 00:10.48.338919 [DVoltages.cpp:100, INF] VCC_AUX: 2
2021-06-06 00:10.48.340044 [DVoltages.cpp:101, INF] VCC_BRAM: 1
2021-06-06 00:10.48.340153 [DVoltages.cpp:102, INF] VCC_PSDDR: 1
2021-06-06 00:10.48.340261 [DVoltages.cpp:103, INF] VCC_VREF: 0
2021-06-06 00:10.48.340373 [DVoltages.cpp:104, INF] VCC_PSIO: 1
2021-06-06 00:10.48.340483 [DVoltages.cpp:105, INF] VCC_PSBATT: 2
2021-06-06 00:10.48.341607 [DVoltages.cpp:106, INF] VCC_PSINTfPDDR: 1
2021-06-06 00:10.48.342732 [DVoltages.cpp:107, INF] VCC_AMS: 2
2021-06-06 00:10.48.342840 [DVoltages.cpp:108, INF] VCC_PSDDRPLL: 2
2021-06-06 00:10.48.343961 [DVoltages.cpp:109, INF] VCC_PSMGTRAVCC: 1
2021-06-06 00:10.48.344067 [DVoltages.cpp:110, INF] VCC_PSINTLP: 1
2021-06-06 00:10.49.344241 [DTemperatures.cpp:98, INF] TEMP_PS: 49
2021-06-06 00:10.49.344390 [DTemperatures.cpp:99, INF] TEMP_REMOTE: 49
2021-06-06 00:10.49.344497 [DTemperatures.cpp:100, INF] TEMP_PL: 49
2021-06-06 00:10.49.345647 [DVoltages.cpp:99, INF] VCC_INT: 1
2021-06-06 00:10.49.346776 [DVoltages.cpp:100, INF] VCC_AUX: 2
2021-06-06 00:10.49.347898 [DVoltages.cpp:101, INF] VCC_BRAM: 1
2021-06-06 00:10.49.348003 [DVoltages.cpp:102, INF] VCC_PSDDR: 1
2021-06-06 00:10.49.348108 [DVoltages.cpp:103, INF] VCC_VREF: 0
2021-06-06 00:10.49.348219 [DVoltages.cpp:104, INF] VCC_PSIO: 1
2021-06-06 00:10.49.348324 [DVoltages.cpp:105, INF] VCC_PSBATT: 2
2021-06-06 00:10.49.349445 [DVoltages.cpp:106, INF] VCC_PSINTfPDDR: 1
2021-06-06 00:10.49.350563 [DVoltages.cpp:107, INF] VCC_AMS: 2
2021-06-06 00:10.49.350667 [DVoltages.cpp:108, INF] VCC_PSDDRPLL: 2
2021-06-06 00:10.49.351785 [DVoltages.cpp:109, INF] VCC_PSMGTRAVCC: 1
2021-06-06 00:10.49.351888 [DVoltages.cpp:110, INF] VCC_PSINTLP: 1
```

Functionality #2: PPr and DCS system – Project 2

- **Alternative 1: Native quasar using C++**
 - Pros:
 - Sample framework available and was used before with Ipbus
 - Possible option is to use Milkyway to read data from CPM and TDAQi
 - Cons: Project schedule might be changed because the old OpcUAipbus project is deprecated
- **Alternative 2: Hybrid OpcUA using quasar + Poverty**
 - Pros: Easier and faster development using python for Poverty with quasar
 - Cons: Limited to some data type it can read and publish to the client
- **Alternative 3: Quasar implementation using pure Python**
 - Cons:
 - This is not a complete framework yet.
 - Thus, the project cannot be used at this moment

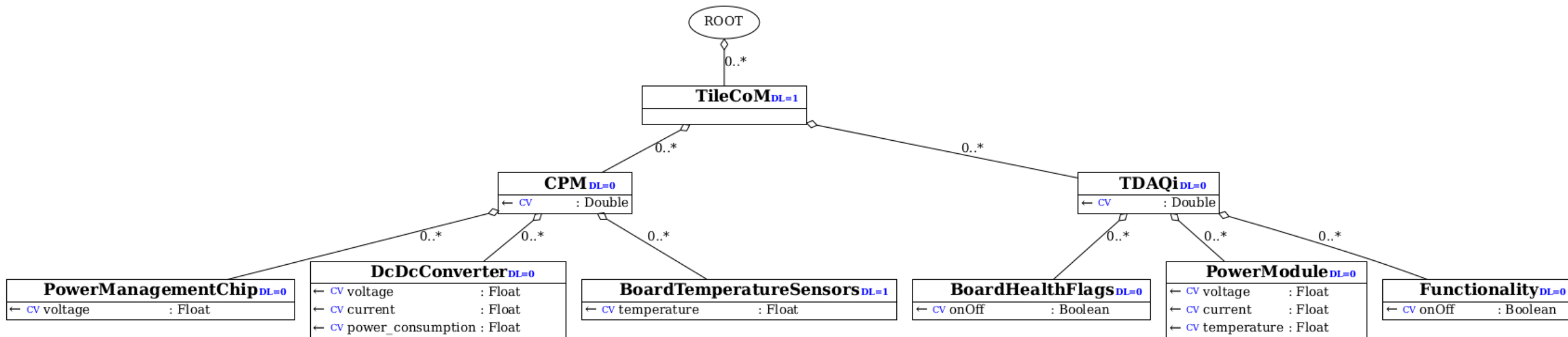
Functionality #2: PPr and DCS system – Project 2

- OpcUATileCoM is a separate OPC that will read monitoring data from the CPM through the Ipbus
- The Ipbus alternative was chosen to be used as a means of of communication between CPM and TileCoM because a similar project was done before and it can read all data types



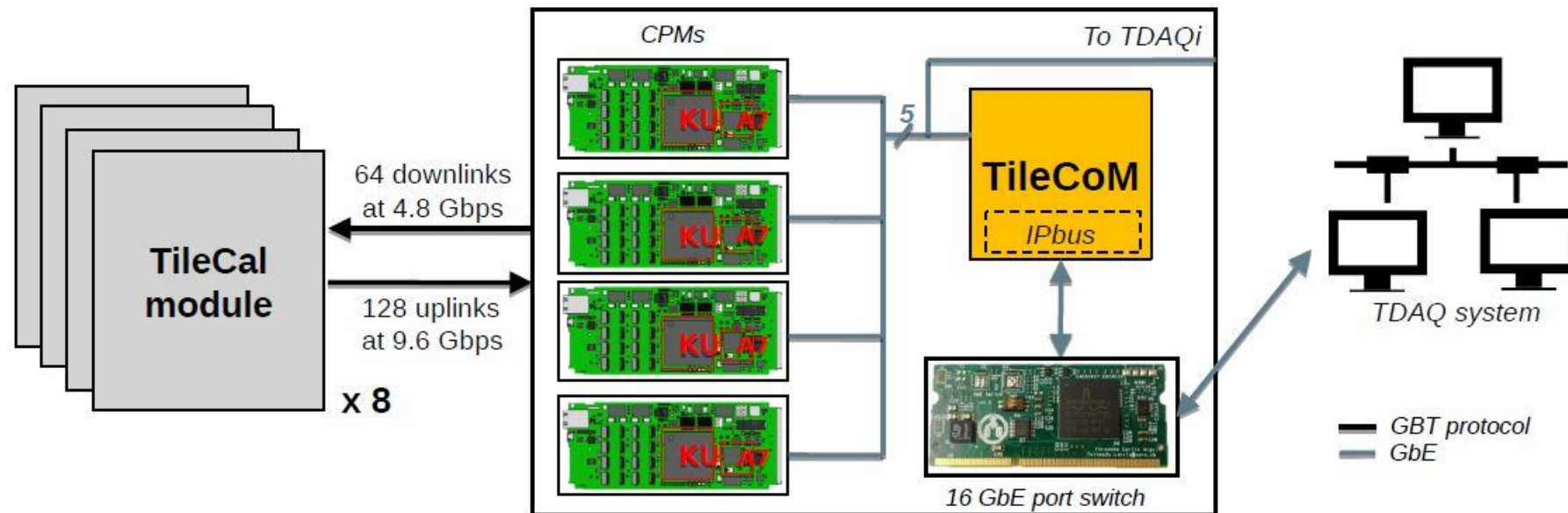
Functionality #2: PPr and DCS system – Project 2

- Design diagram for the TileCoM reading data from the CPM and TDAQi
- The diagram is generated by the quasar framework with CPM and TDAQi as root classes
- Both the CPM and the TDAQi will be connected to the TileCoM through the GbE port
- The following diagram only shows few sensors read by CPM and TDAQi



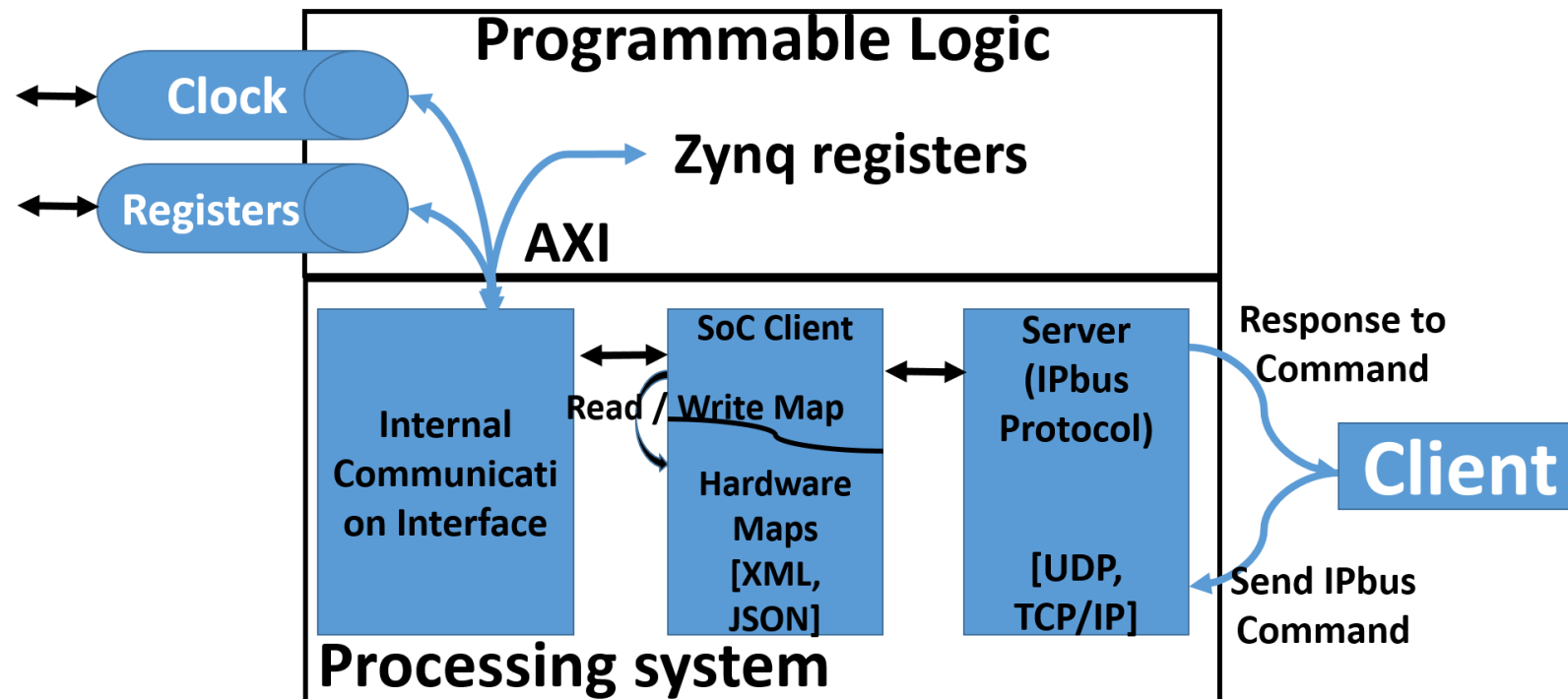
Functionality #3: TDAQ interface

- Interface to the TDAQ system for system monitoring / configuration
- Slow control and configuration of the PreProcessors
 - Configuration of the ATCA carrier and CPMs: clocking, power, optical modules, GbE switch, ...
 - Monitoring of ATCA on-board sensors, CPM sensors and on-detector electronics sensors through I2C and GbE



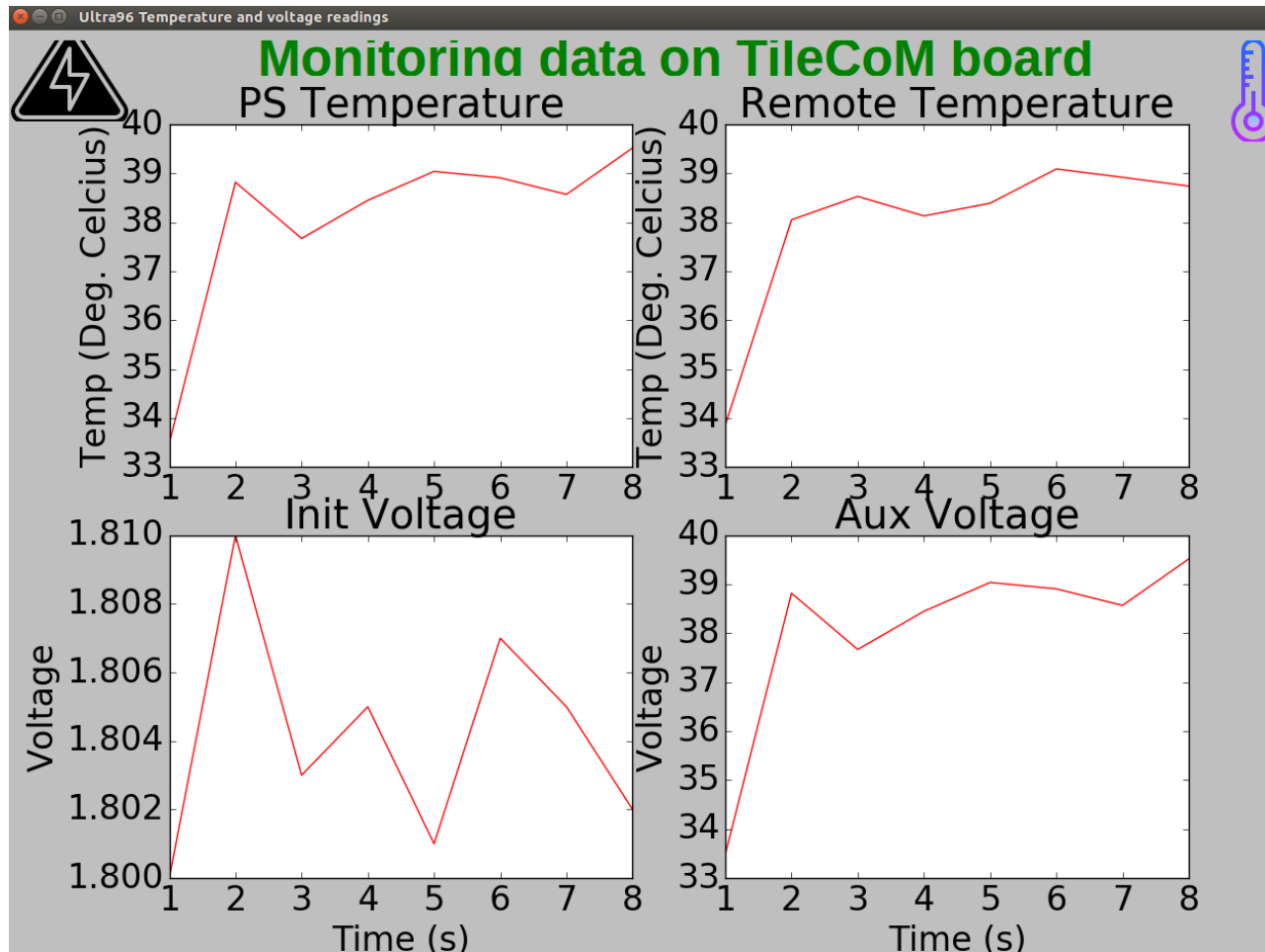
Ironman implementation for IPbus

- The IPbus server is implemented directly on the Ultra96-V2 Zynq UltraScale+ MPSoC TileCoM evaluation board.
- The server actively listens on the reactor, performs basic sanity checks, sends and receive packets from SoC client



Ironman implementation for IPbus - Client

The client is implemented with Twisted libraries and PyQt5 on Ubuntu 18.04 to read out Xilinx Analog-to-Digital Converters temperature and voltages.



Conclusion

- Complete redesign of the read out electronics for the HL-LHC
 - Total of 32 PreProcessors to readout TileCal
 - Each PreProcessor will host a TileCoM mezzanine board
- TileCoM based on Zynq UltraScale+MpSoC has been produced and it is currently under tests
- **IFIC** in charge of hardware and; **Wits** in charge of firmware and software
- TileCoM functionalities
 - Remote programming functionality was implemented before
 - Two OPC-UA servers are implemented on the TileCoM
 - One OPC-UA is a generic project and the other project will be for Tile
 - Ironman Ipbus software will be used for TDAQ interface functionality
- These functionalities will soon be implemented on the actual TileCoM
 - To remotely control and configure the on and off detector electronics
 - To read Temperature & humidity sensors, power monitoring, etc