Introductory Tutorial

MicroTCA Management

Dariusz Makowski

dmakow@dmcs.pl
dariusz.makowski@desy.de

Agenda

- Introduction and Overview
- Shelf Management in xTCA Systems
- xTCA for Physics Extension
- IPMI Implementation for MMC and RTM
- IPMI MMC 1.00 Framework

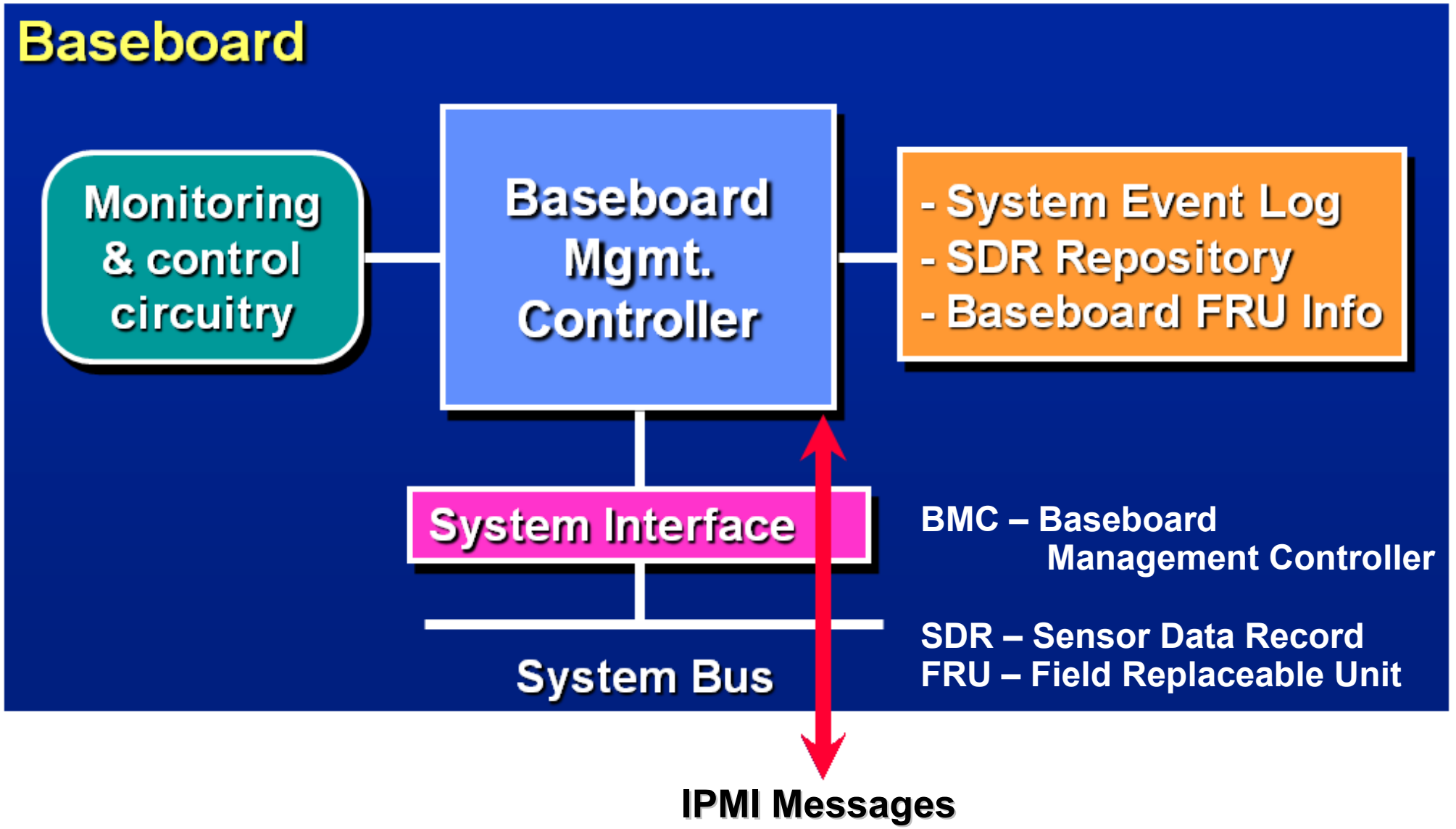
Introduction and Overview

Keep it Running...

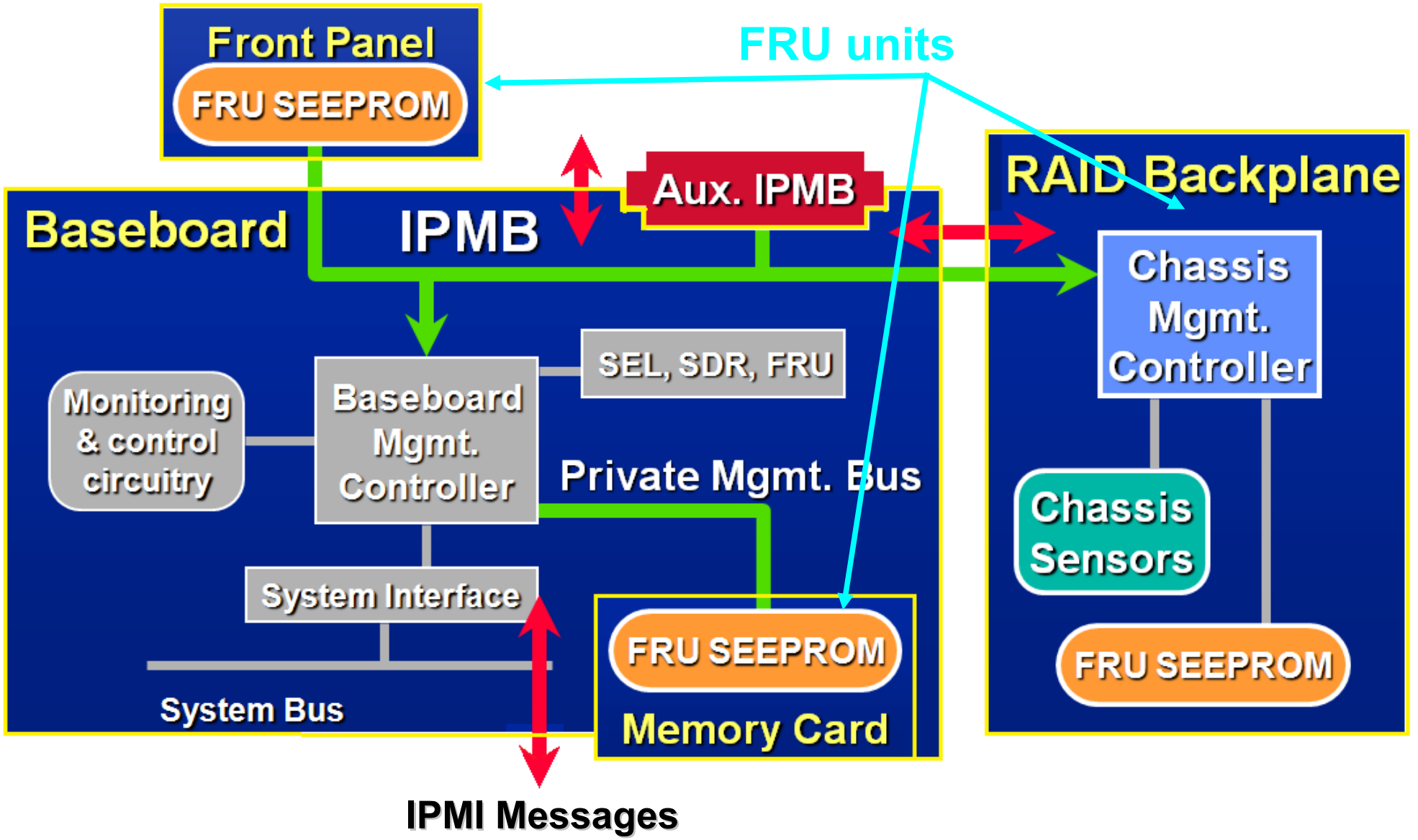
- **Intelligent Platform Management Interface** – protocol initially developed by Intel, Hewlett-Packard, NEC and DELL consortium
- Used by system administrators for out-of-band management of computer systems and monitoring of their operation
- First draft available in Spring '98 (IPMI v0.9)
- **RAS** Features Focus:
  - **R**eliability
  - **A**vailability
  - **S**erviceability
- Server oriented:
  - Remote administration
  - Expensive hardware
  - High costs of downtime and repair
- Plug and Play, Hot-swap

SuperBlade supercomputer
source: www.supermicro.com

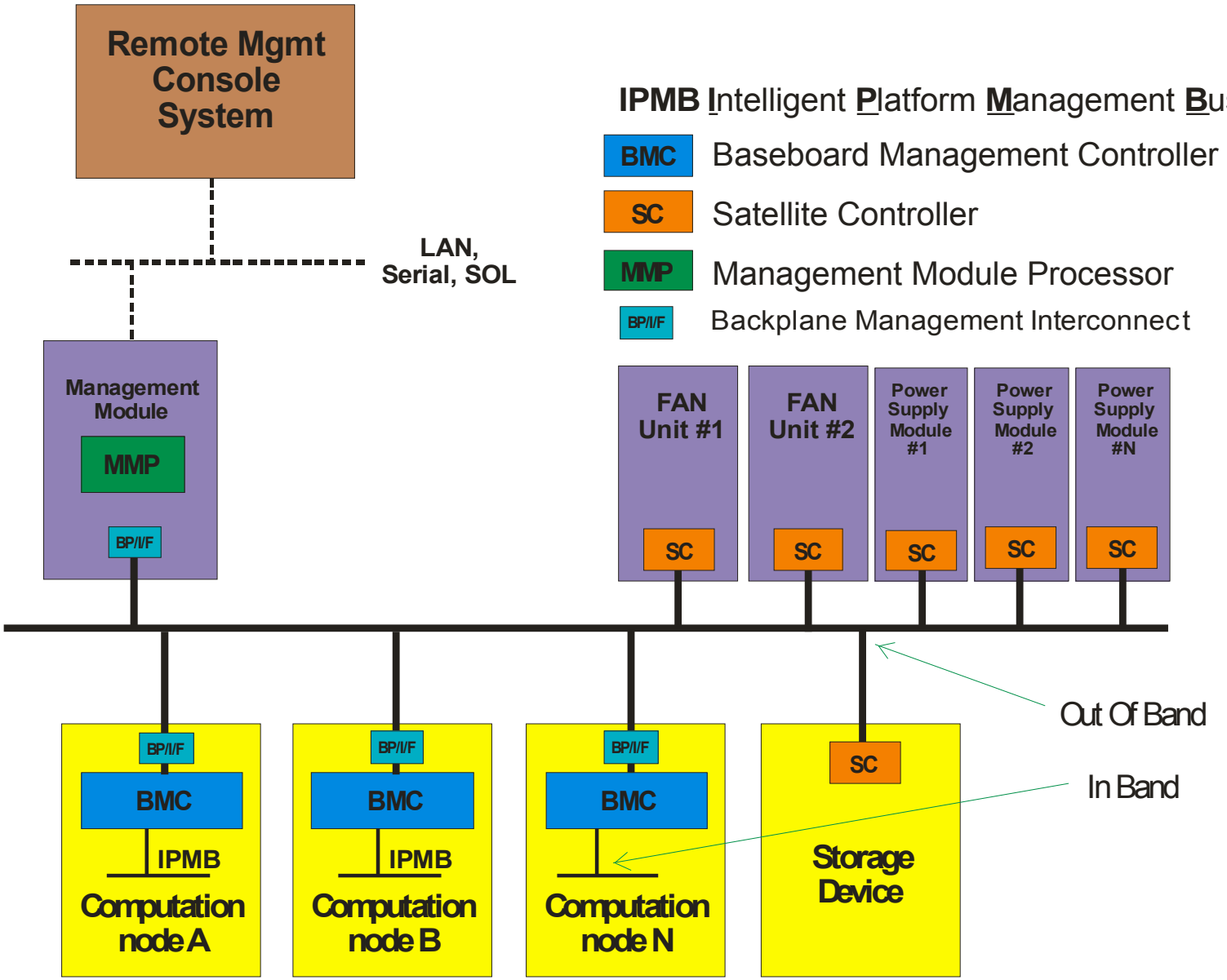
IPMI Components



Computer Node - IPMI Implementation



Chassis Implementation



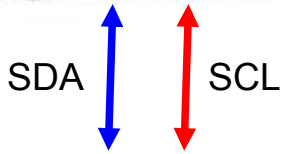
IPMI Elements (1)

- **Baseboard Management Micro-Controller provides intelligence for IPMI**
  - Out Of Band (OOB) access to computer
  - Autonomous monitoring and logging
  - System interface to internal IPMB (I2C)
  - Interface to IPMI Storage
  - Receives and logs events messages
  - Power control functions
  - BIOS access
  - OEM defined functions
  - System Watch-dog Timer

IPMI Elements (2)

- **Field Replaceable Unit (FRU)**

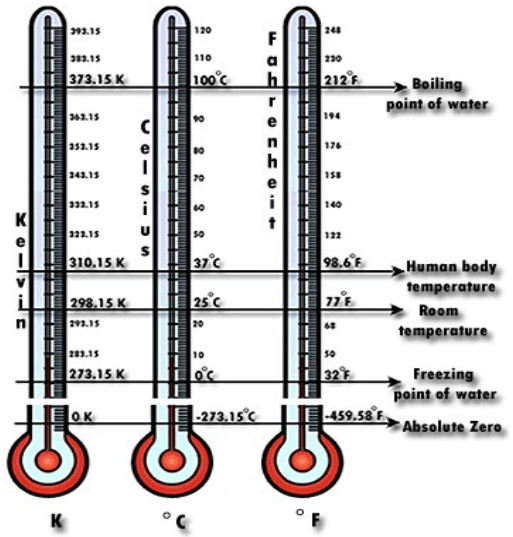
Field replaceable components of the system such as a board, module, fan unit, power supply module, raid matrix, etc.
FRU records are stored in a non-volatile memory.

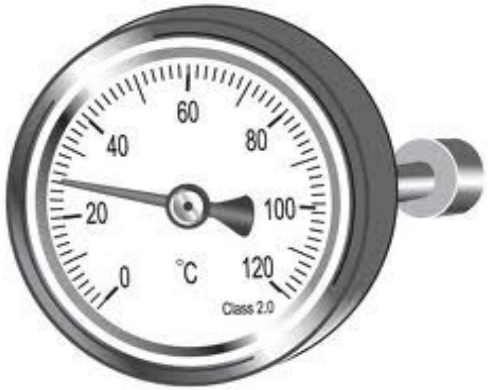


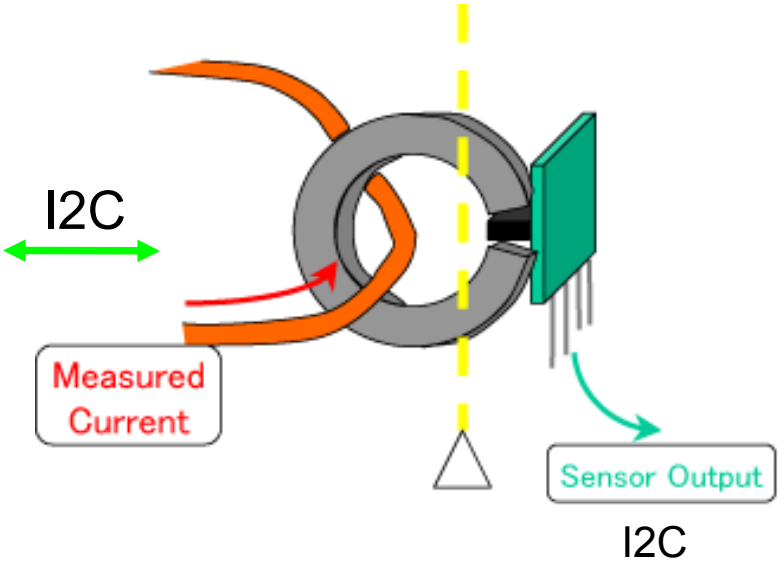
I2C

- **Sensor Data Record (SDR)**

Provide information about available on FRU sensors, events, management controllers, e.g. temperatures, voltages, sensors, etc.



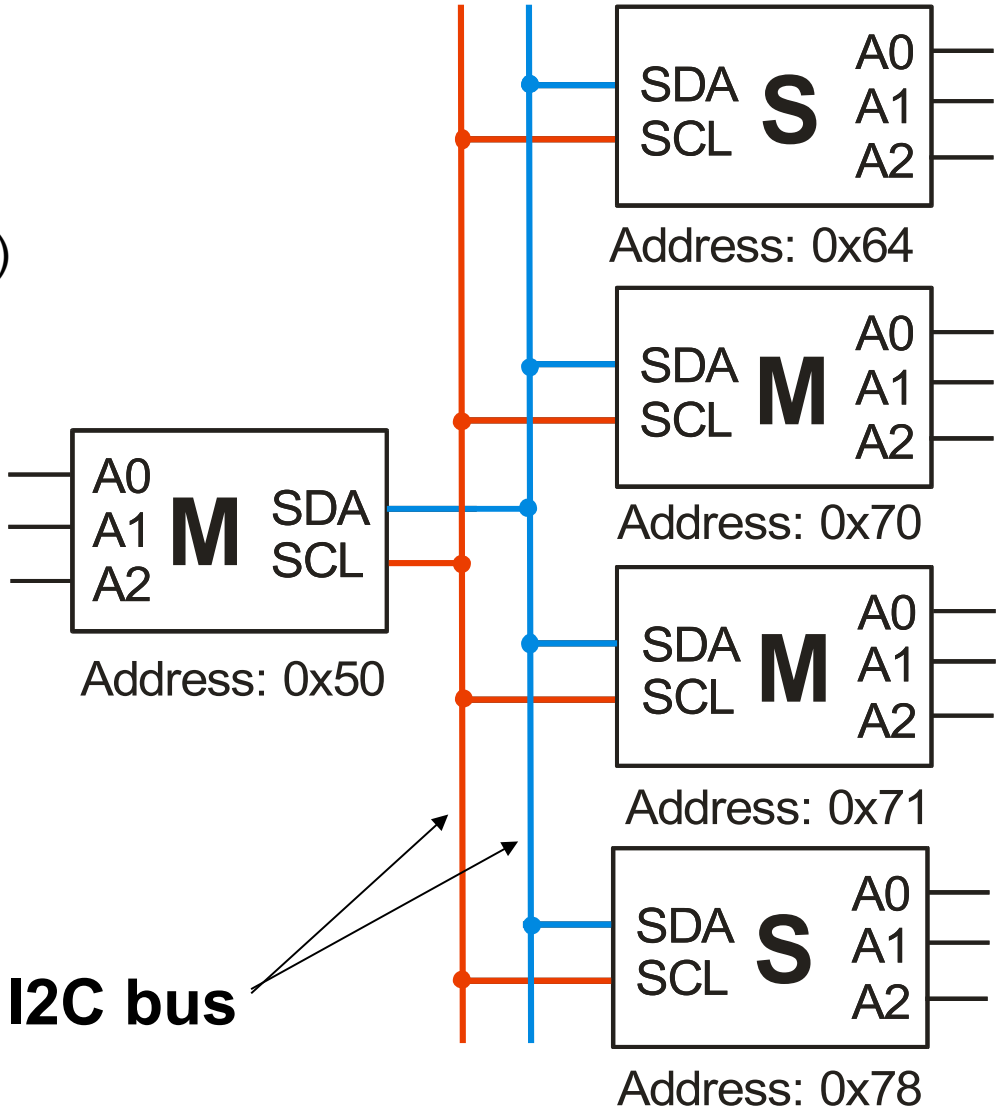




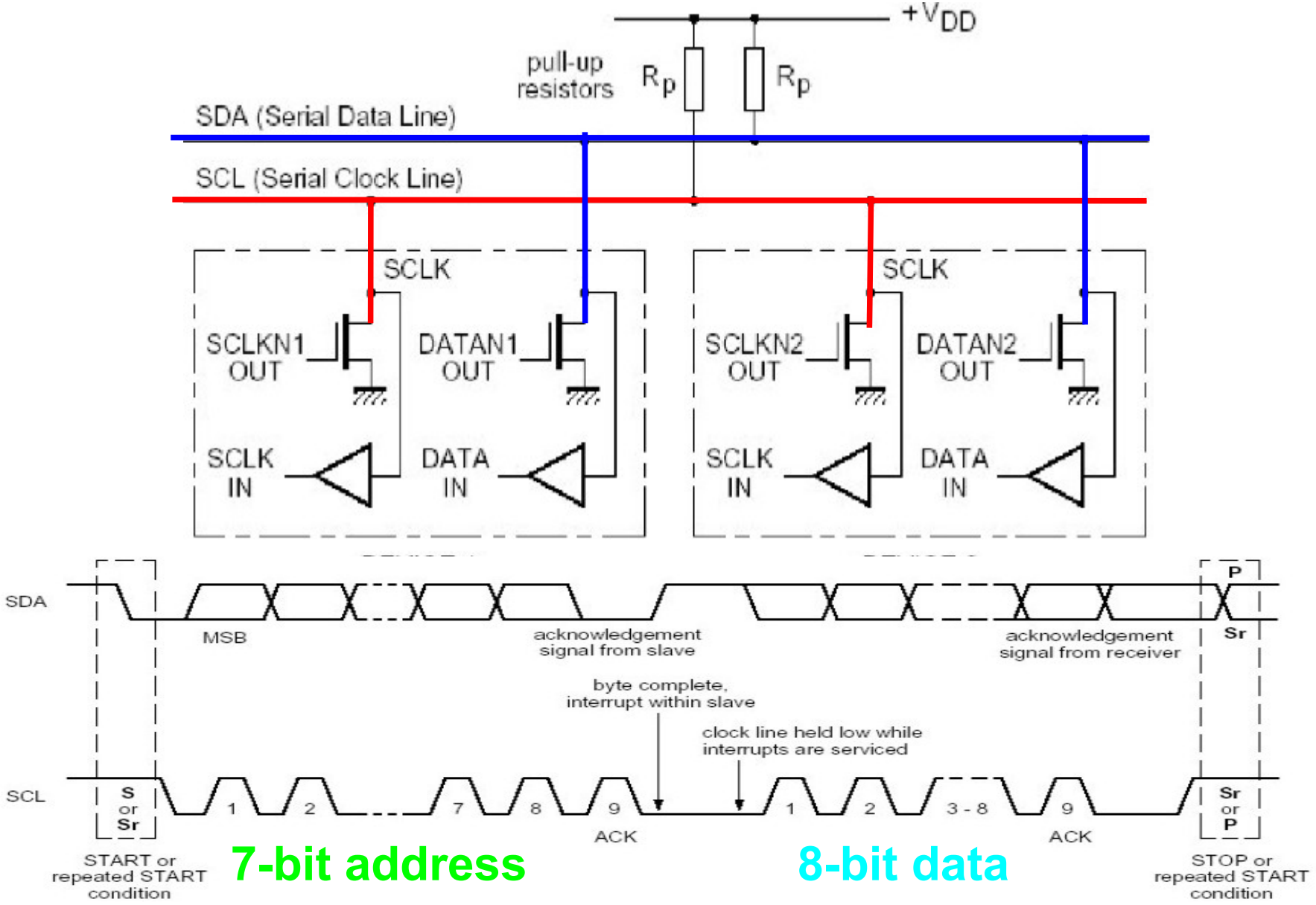
I2C

I2C Interface – IPMI Backbone (1)

- I2C – Inter-Integrated Circuit bus
- Standard developed by Philips on early 80s
- Two wire synchronous, half-duplex interface (SDA – data line, SCL – clock line)
- Bidirectional multi master-slave transfers, 8-bit frames
- Transmission speed: 100 kbps, 400 kbps, 3.4 Mbps
- 7-bit or 10-bits device address
- Arbitration used for multi-master transmission



I2C Interface – IPMI Backbone (2)



Shelf Management in xTCA Systems

xTCA and IPMI

- IPMI introduced in PICMG 2.16 backplanes (CompactPCI systems) and later in AMC, ATCA and MTCA standards
- IPMI enables "diagnose-before-dispatch" automation
- Required for 99.999 percent high availability (HA) mark
- IPMI controller (shelf manager) is responsible for:
  - Monitoring overall shelf health
  - Communicating with remote System Management Software (SMS)
  - Hot-swap events (e.g. hardware component entry-removal events)
  - Latch/lock management
  - Power budgeting
  - In-rush current sequencing
  - Electronic keying (E-keying)
- PICMG 3.0 extension commands
- HPM.1 management firmware upgrade capability

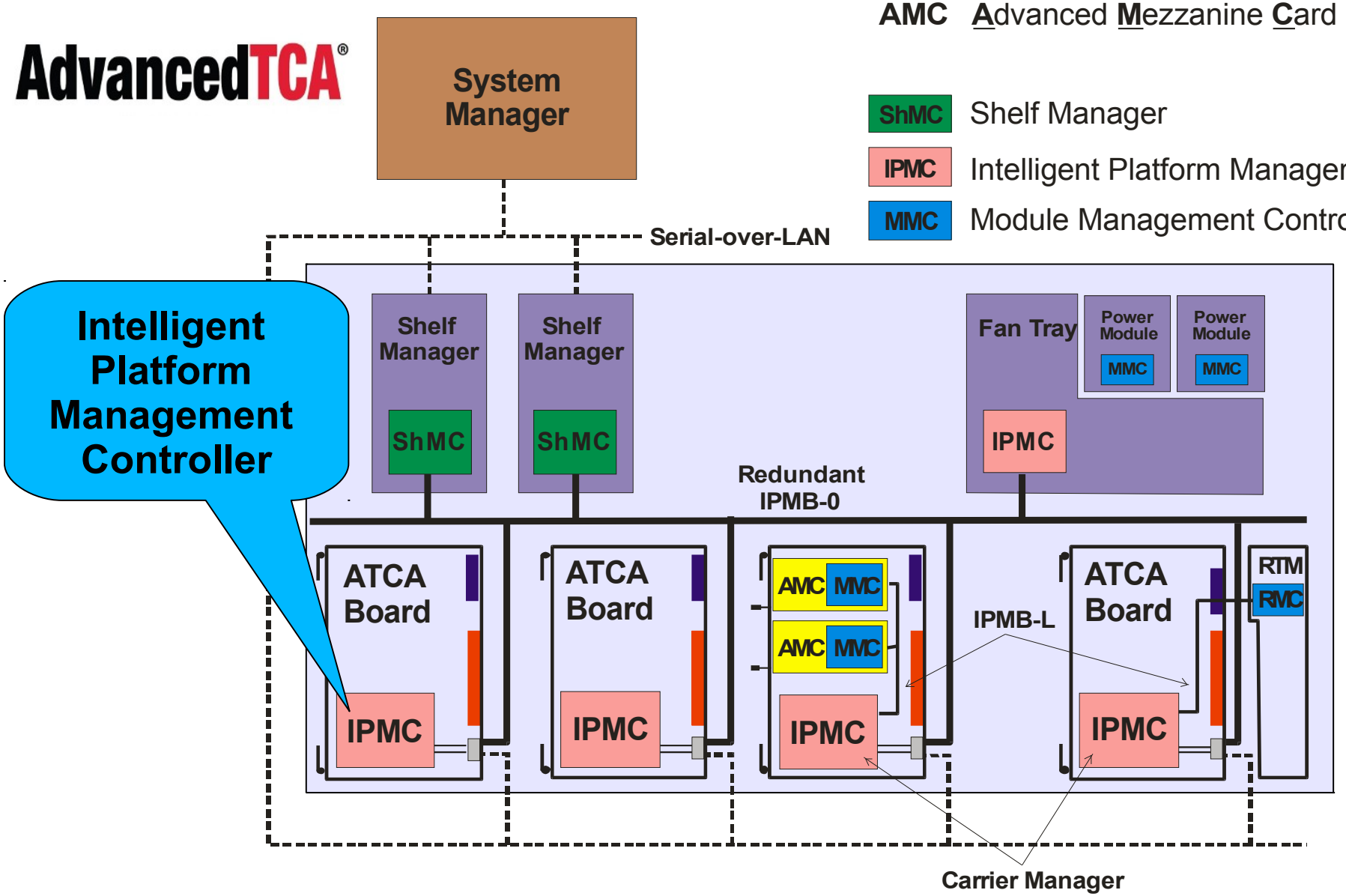
ATCA Shelf Management System

AMC Advanced Mezzanine Card

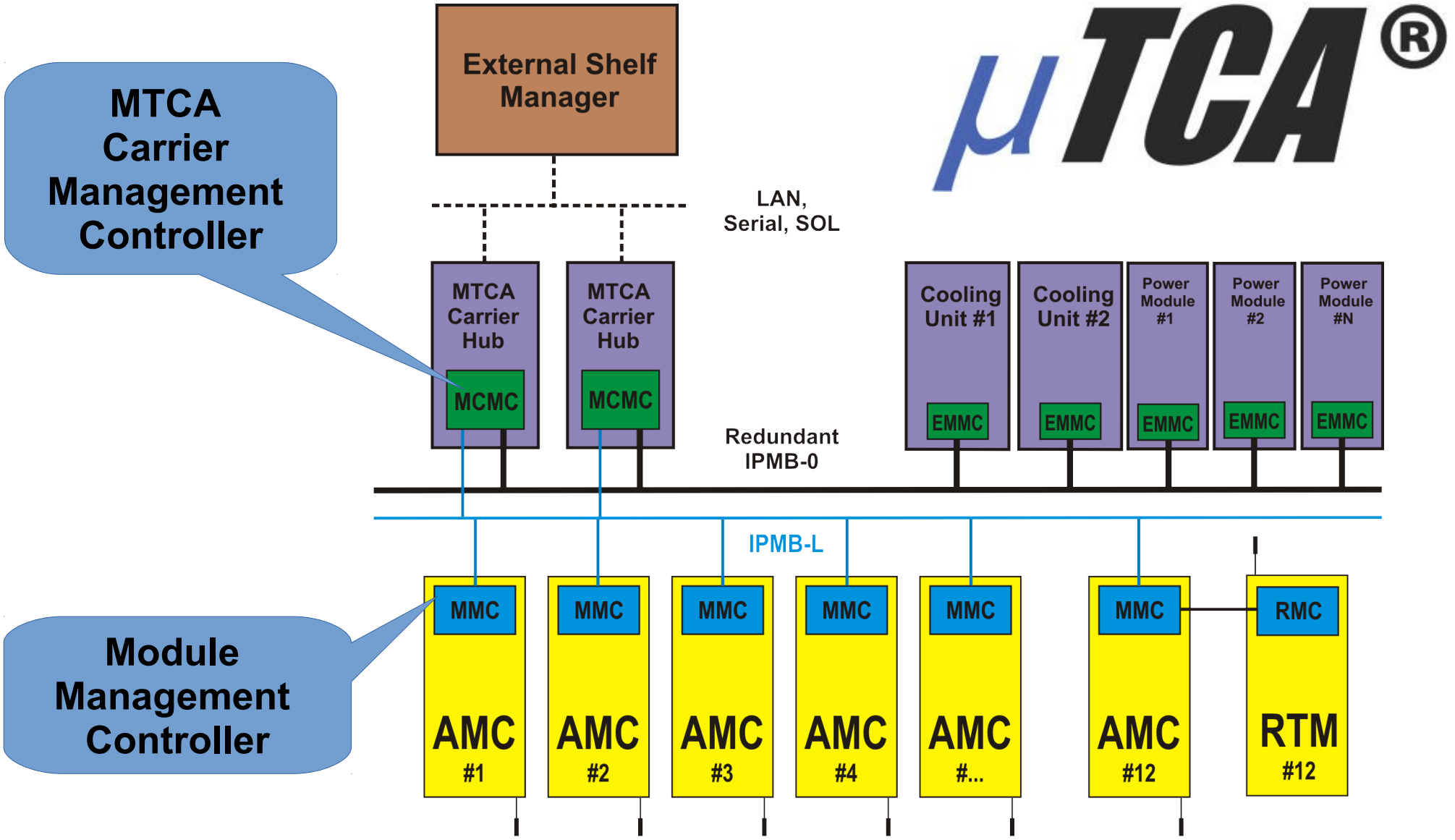
ShMC Shelf Manager

IPMC Intelligent Platform Management Controller

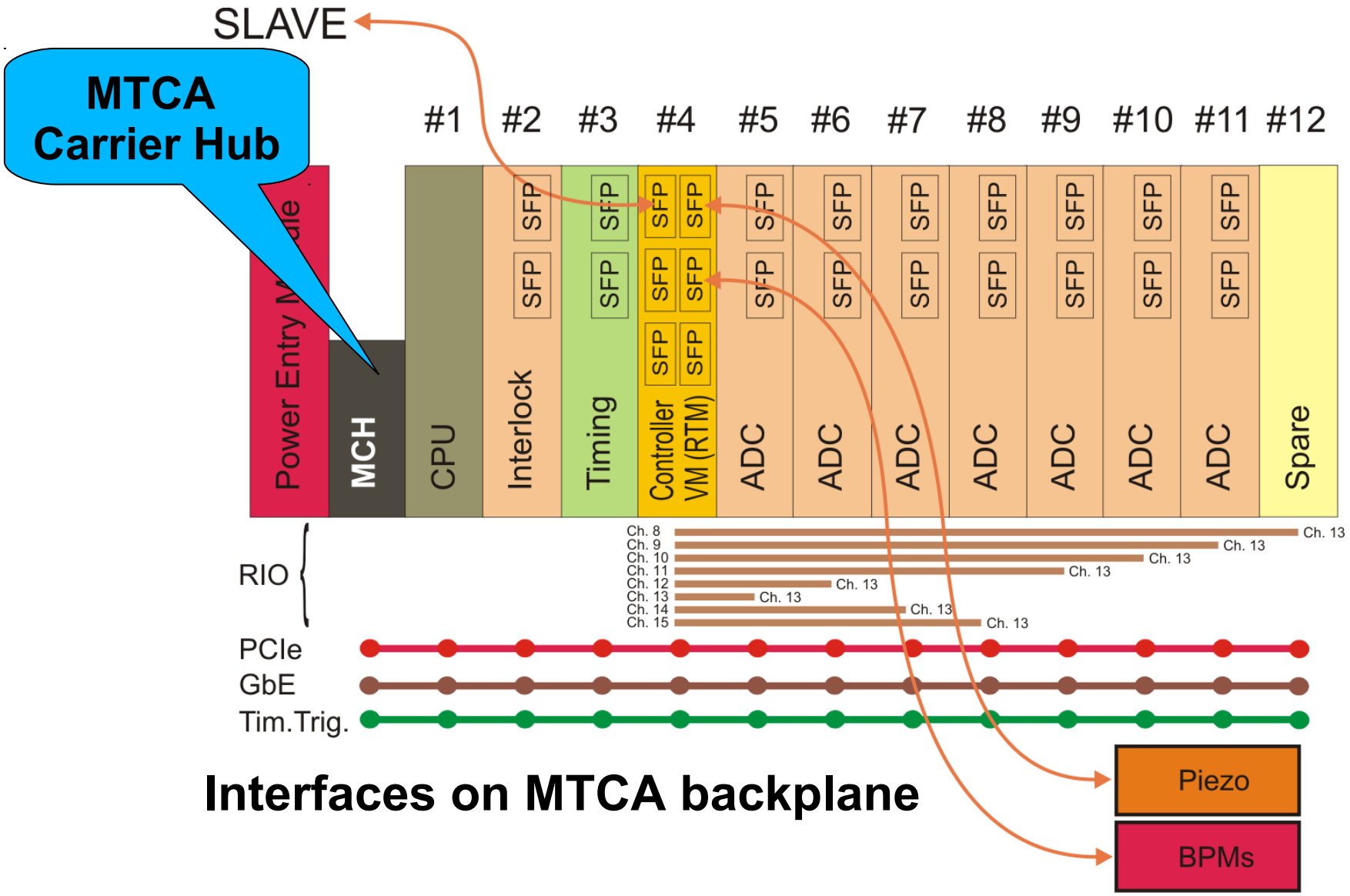
MMC Module Management Controller



MTCA Shelf Management System



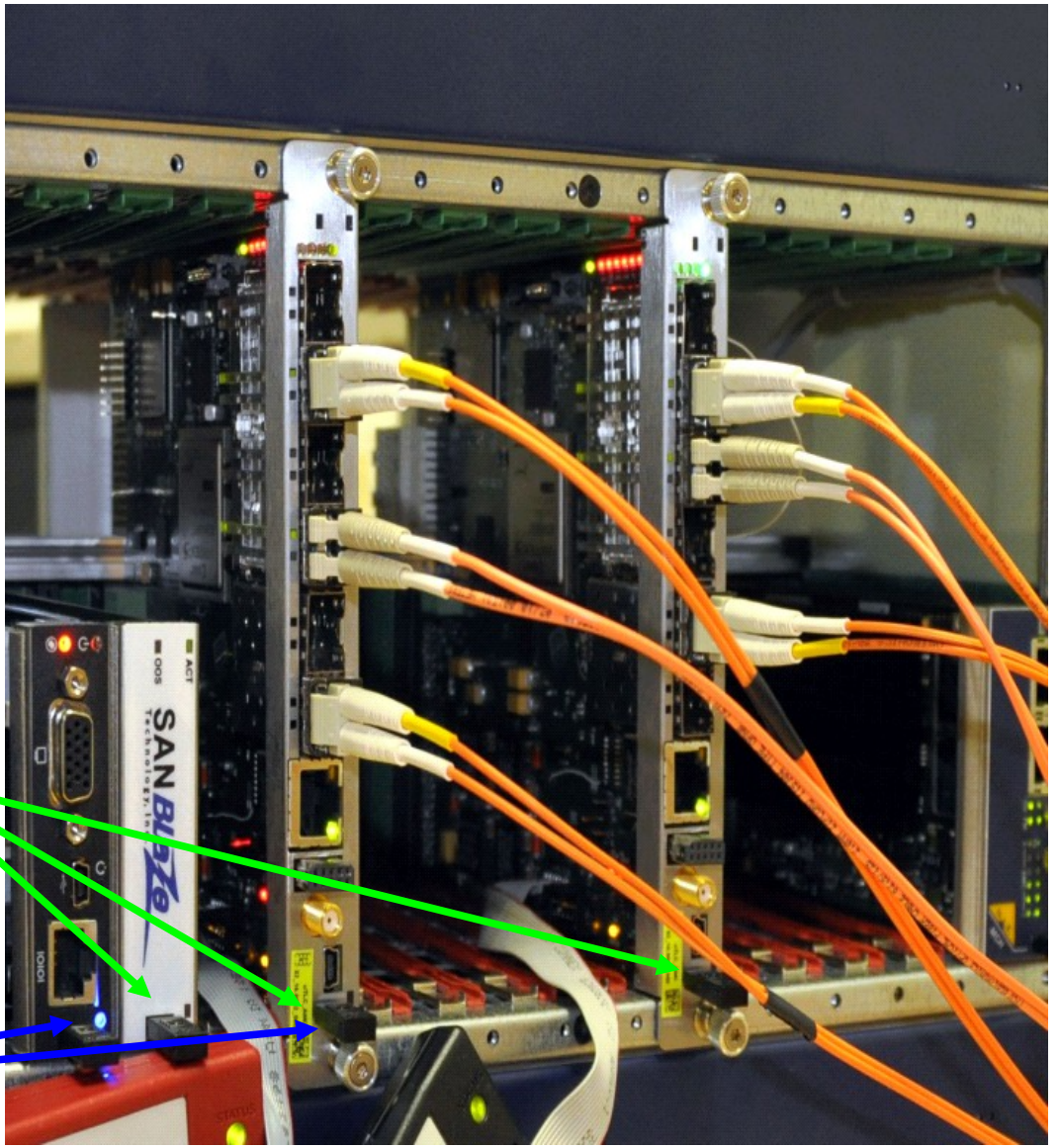
uTCA-based LLRF Control System of Accelerator



Interfaces on MTCA backplane

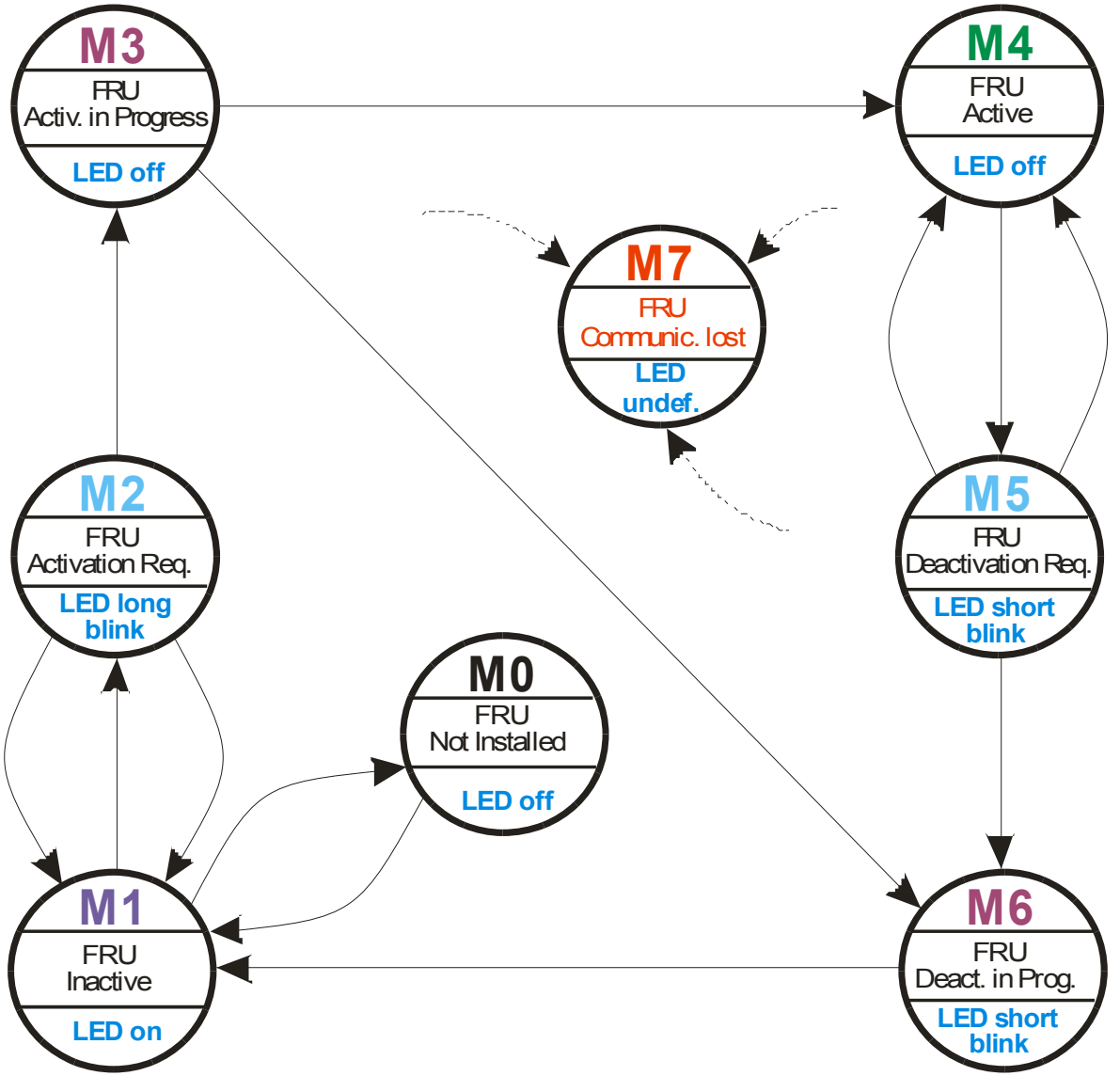
AMC Module Hot-plug

- What really happens when you plug AMC card into MTCA chassis?
- Module Activation procedure?
- Module Deactivation?
- How IPMI automates this process?
- What if something goes wrong?

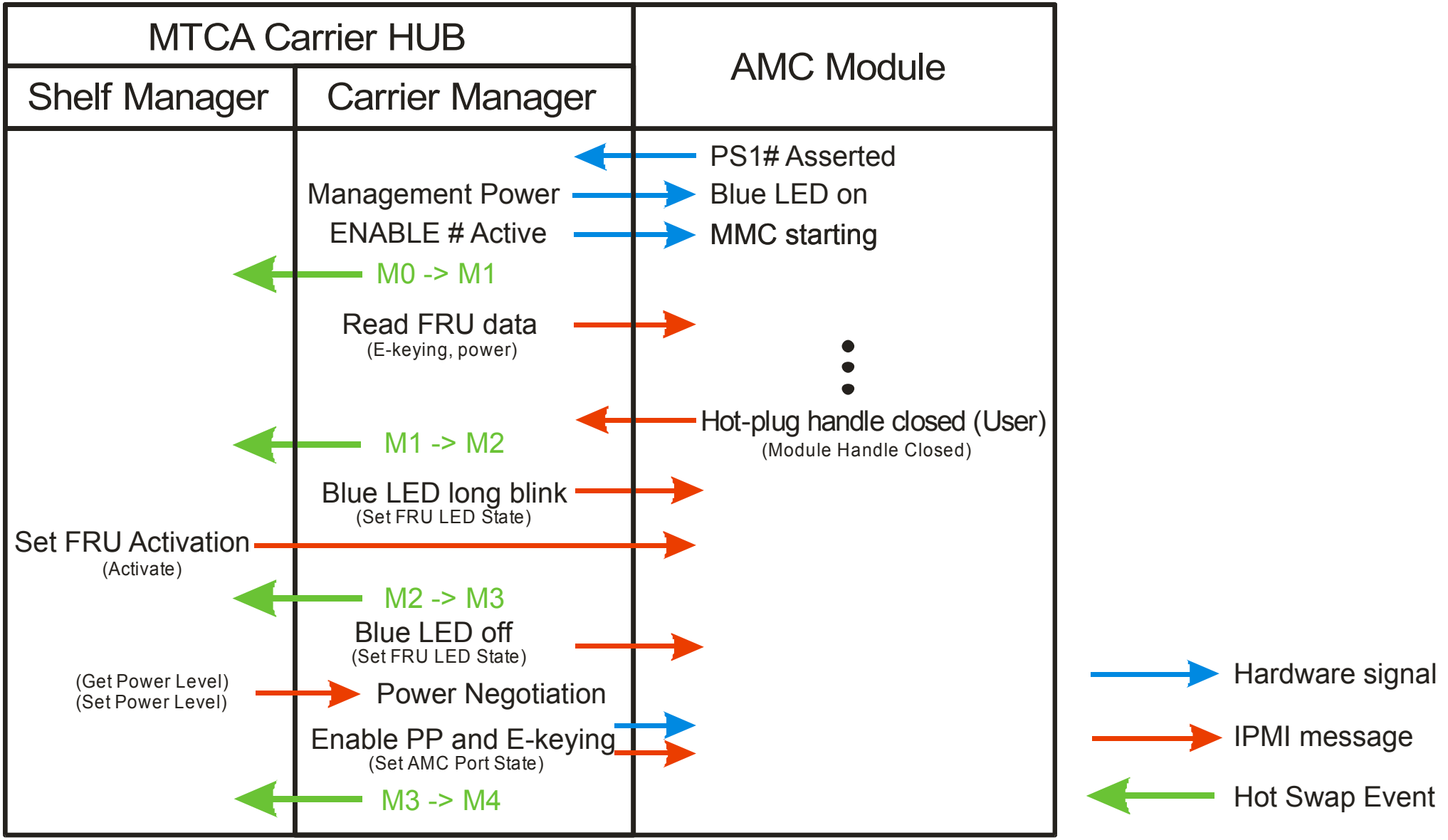


Module Activation/Deactivation

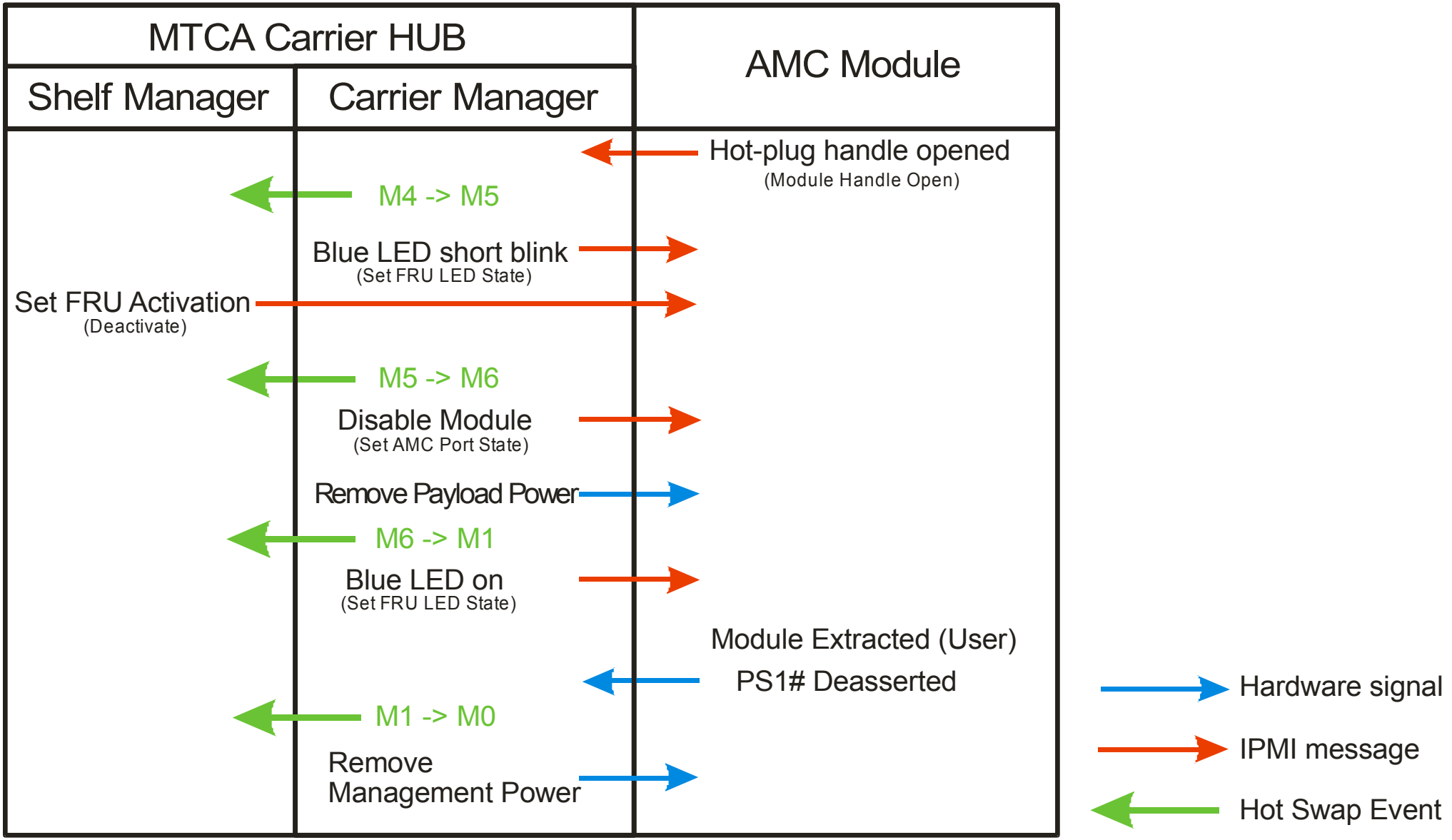
- PICMG 3.0 and AMC specifications define FRU states
- Activation pushes FRU into M4 state
- Deactivation moves FRU into M1 state
- If something wrong happen module goes into M7 state
- MCH decides if and when module can reach M4
- MMC uses a state machine to control hot-plug procedure



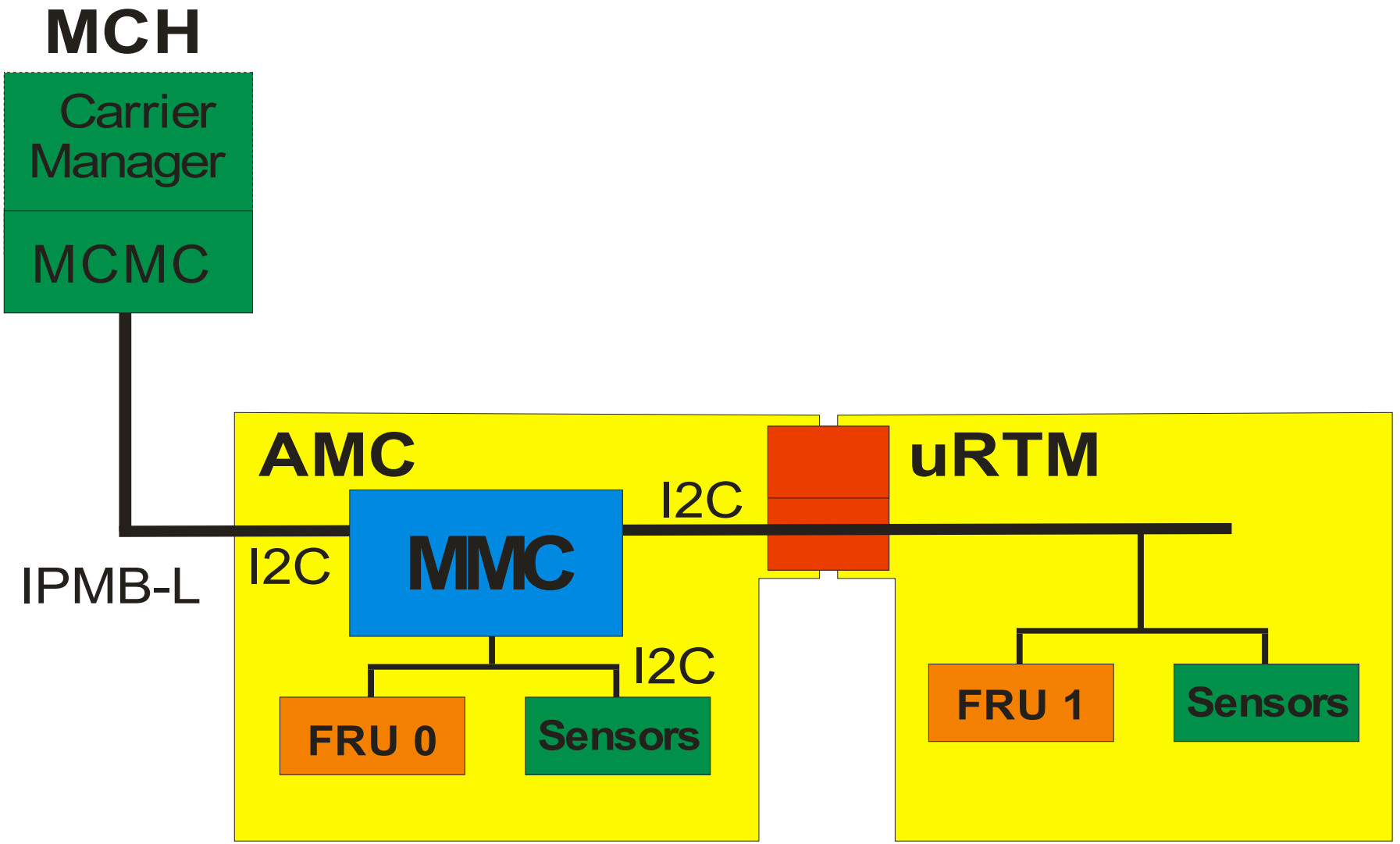
AMC Module Insertion



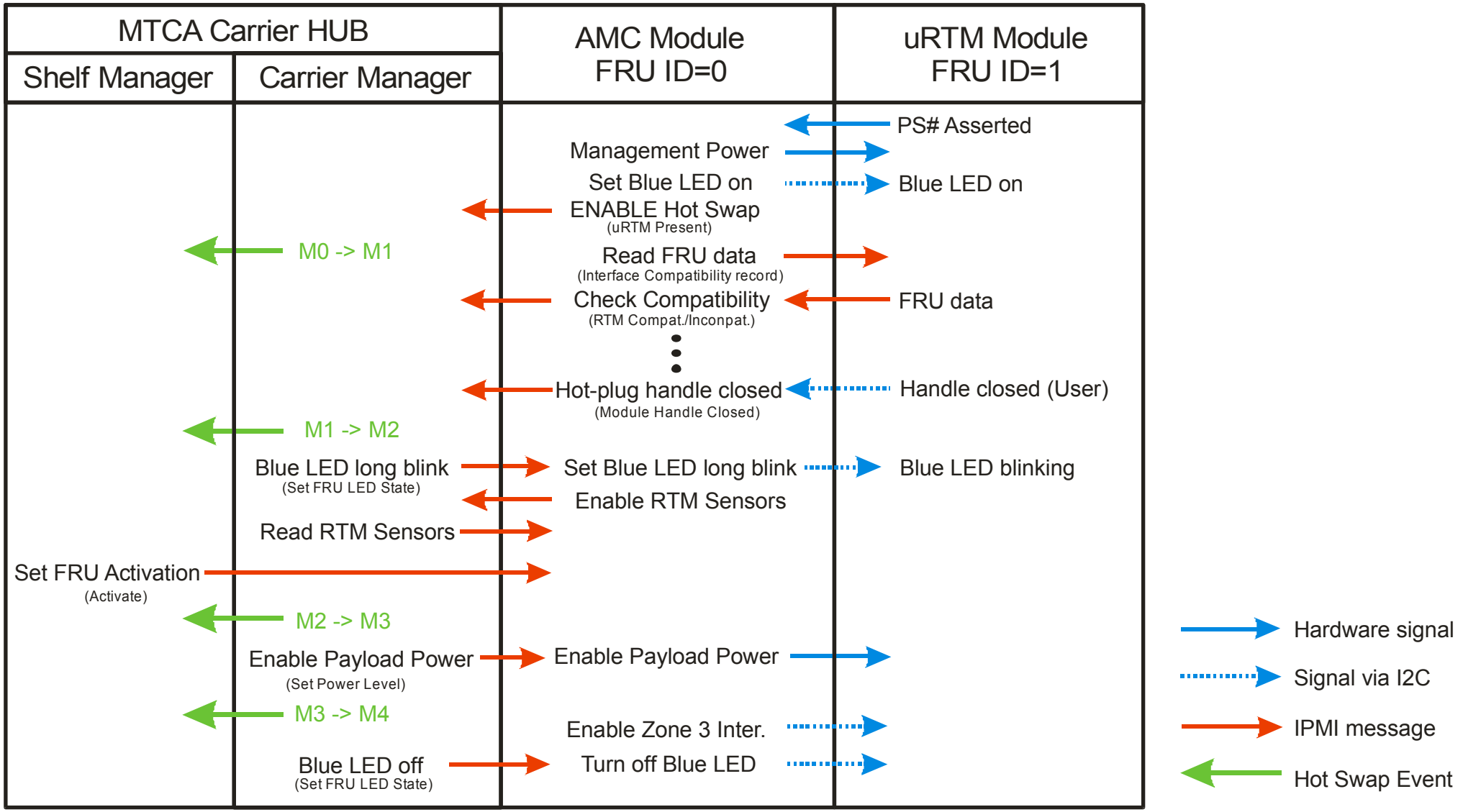
AMC Module Extraction



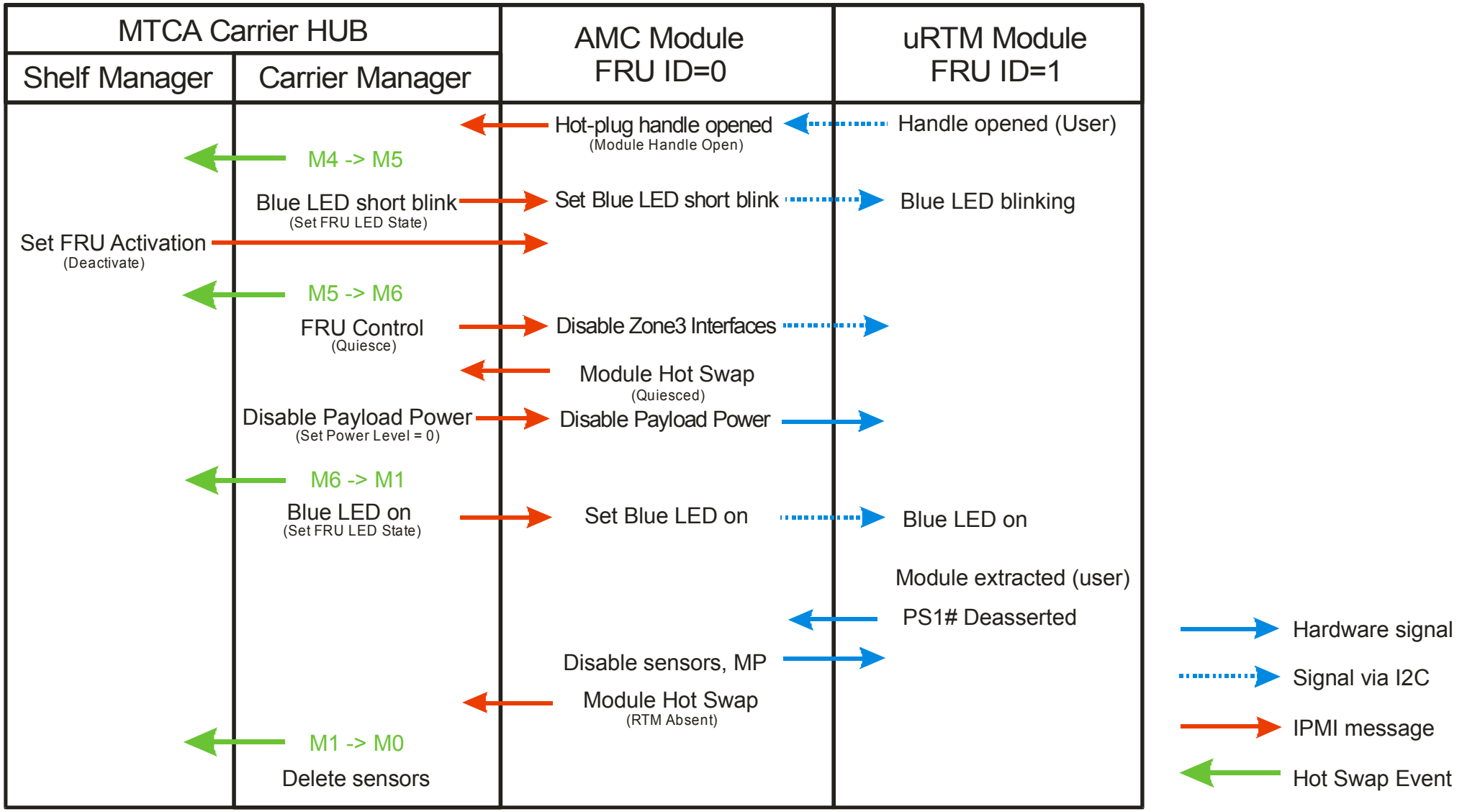
MTCA.4 – Hardware Management



RTM Module Insertion

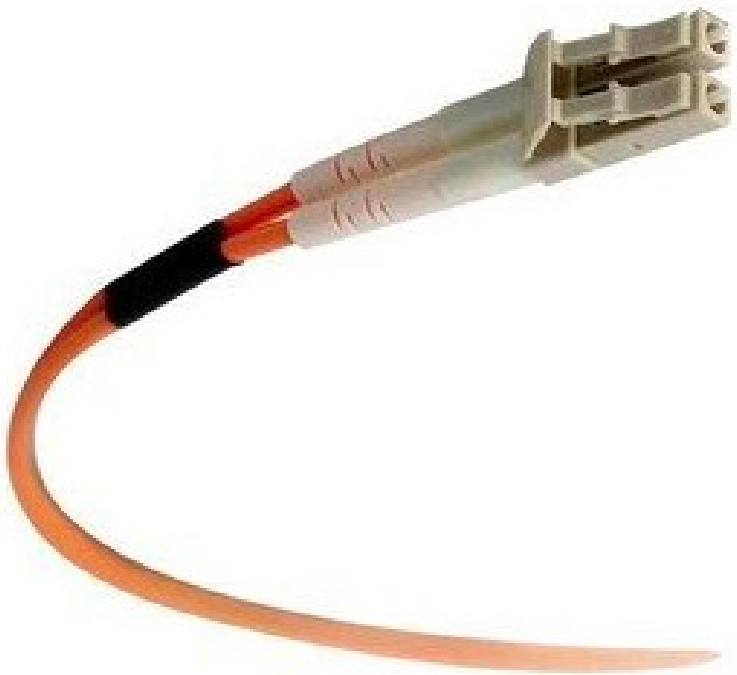


RTM Module Extraction



Electronic-keying – Connection Compatibility Management

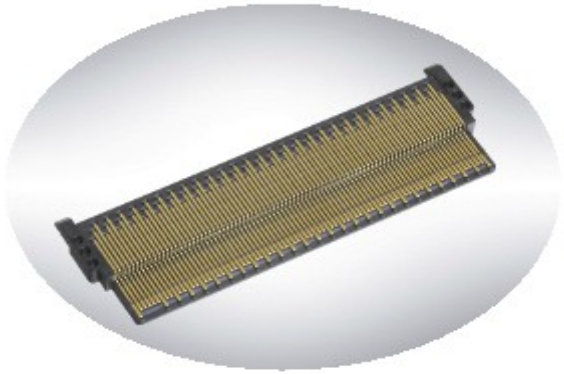
- Allow using different standards (1/10 GbE, PCIe, SRapidIO, Inifiniband, Custom links)
- Provide connections compatibility
- Protect hardware



E-keying for MTCA Zone 1 Connector

MTCA Zone 1 connector divided into zones:

- Each module can receive or drive clock signals
- Star connection on ports 0 and 1
- P2P (Peer-to-Peer) connection on ports 2 and 3
- Star connection via switch on ports 4-7
- Star connection or P2P on ports 8-11
- P2P connection on ports 12-15
- Bus connection on ports 17-20



AMC 170 pin connector

Region	Interface	Port
Clocks	CLK1	TCLKA
	CLK2	TCLKB
	PCIe Clock	FCLKA
Common Option	1 GbE	0
		1
	SAS/SATA	2
		3
Fat Pipe	PCIE 10 GbE SRIO	4
		5
		6
		7
Ext. Fat Pipe	PCIE 10 GbE SRIO P2P	8
		9
		10
		11
Extended Options	P2P	12
		13
		14
		15
	CLK	TCLKC/D
	Triggers Interlocks Clocks	17
		18
		19
		20

E-keying in MTCA

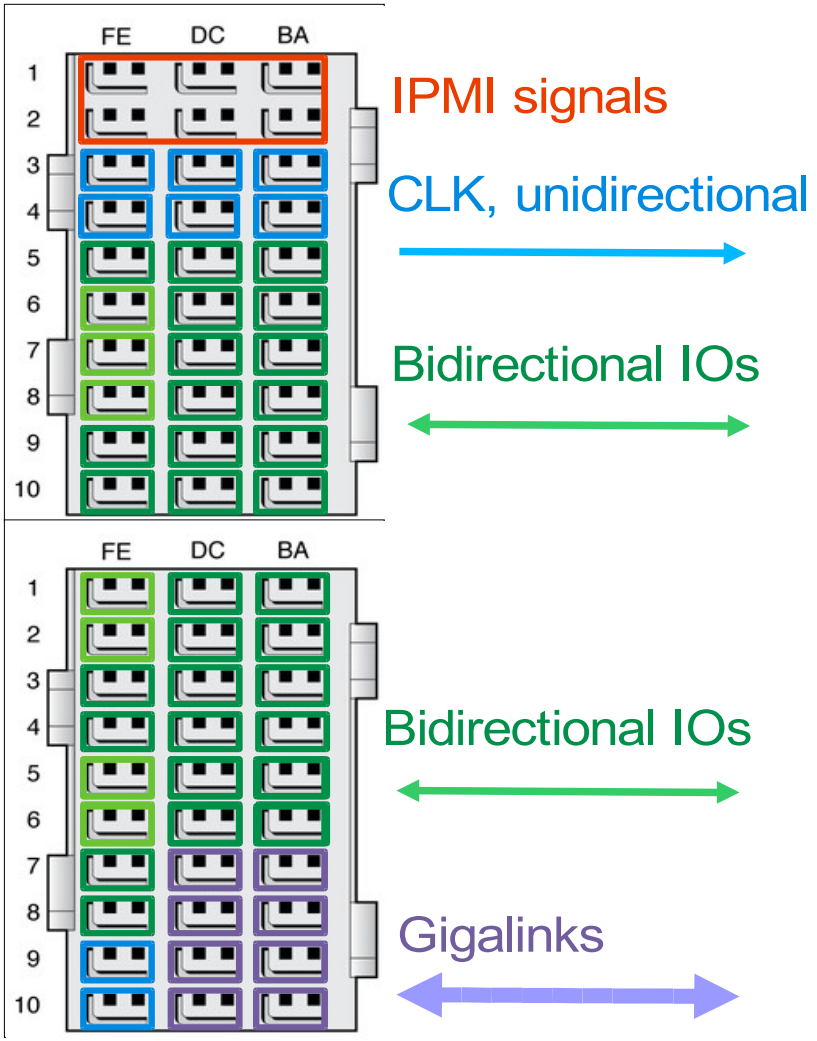
- MCH will read FRU records from both modules
- Compare E-keying records of MCH and AMC3/AMC4
- Compare E-keying records of AMC3 and AMC4
- Activate interfaces with compatible protocols

AMC in slot 3			Decision		AMC in slot 4		
Region	Interface	Port	AMC 3	AMC 4	Region	Interface	Port
Clocks	CLK1	TCLKA	☑	☑	Clocks	CLK1	TCLKA
	CLK2	TCLKB	☑	☑		CLK2	TCLKB
	PCIe Clock	FCLKA	☑	☑		PCIe Clock	FCLKA
Common Option	1 GbE	0	☑	☑	Common Option	1 GbE	0
		1	☑	☑			1
	SATA	2	☒	☑		SAS	2
	SATA	3	☑	☑		SATA	3
Fat Pipe	PCIE 1	4	☑	☑	Fat Pipe	PCIE 1	4
	PCIE 2	5	☑	☒			5
	PCIE 3	6	☑	☒			6
	PCIE 4	7	☑	☒			7
Ext. Fat Pipe	10 GbE	8	☒	☑	Ext. Fat Pipe	SRIO	8
	10 GbE	9	☒	☑		SRIO	9
	10 GbE	10	☒	☑		SRIO	10
	10 GbE	11	☒	☑		SRIO	11
Extended Options	GTX	12	☒	☒	Extended Options	P2P - LVDS	12
		13	☒	☒			13
	P2P - LVDS	14	☑	☑		P2P - LVDS	14
	P2P - LVDS	15	☑	☑		P2P - LVDS	15
	CLK	TCLKC/D					
	Triggers Interlocks Clocks	17	☑	☑		Triggers Interlocks Clocks	17
		18	☑	☑			18
		19	☑	☑			19
		20	☑	☑			20

Zone 3 E-Keying

- DESY Zone 3 recommendation for digital and analogue classes
- Various, bidirectional analogue and digital signals
- MTCA.4 keying only verifies voltage levels
- Require E-Keying with MMC and RTM controller support to avoid signal collisions

N	Data Signal in Volts
1	LVDS
2	0 – ±1
3	>±1 – ±3.3
4	>±3.3 – ±10
5	>±10
6	Reserved
7	Reserved
8	Reserved

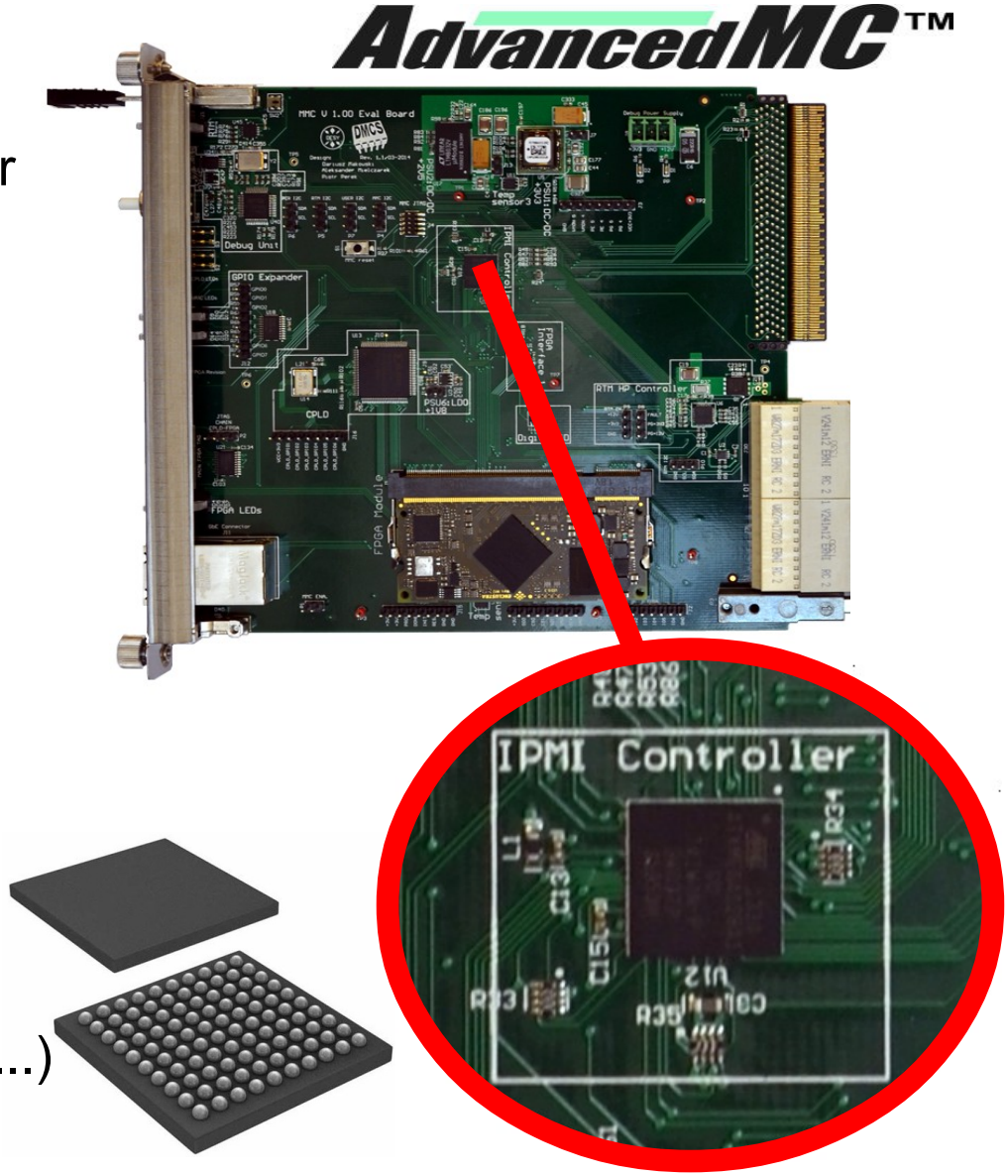


F. Ludwig, '...Zone 3 classes...'

Module Management Controller (AMC Module)

Tasks of MMC

- Required on each AMC Module
- Communication with the Carrier Manager
- Module management:
  - Module activation and deactivation
  - Warm and cold module reset
  - Power supply management
- Monitoring of module crucial parameters
  - Temperature
  - Supply voltages
  - Currents
  - Clocks, etc.
- E-keying mechanism (PCIe, GbE, sRIO,...)
- Supervision of µRTM module (MTCA.4)

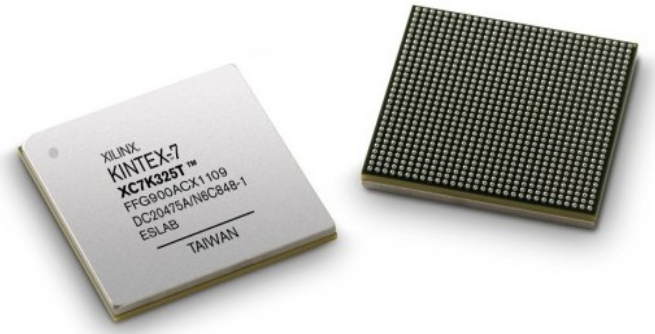


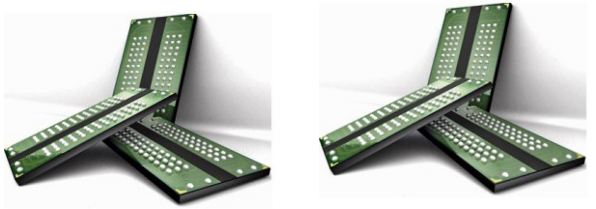
DAMC-TCK7 - LLRF Controller (1)

- Two modes of operation:
  - Low Level Radio Frequency controller
- Enough processing power for execution of the cavity field stabilizing algorithms (PID, feed-forward) in the feedback loop
  - FPGA
  - Memory
- Dedicated low-latency connections
- PCIe and Gigabit Ethernet
- Firmware upgrade feature
- Compliance with MTCA.4 specification

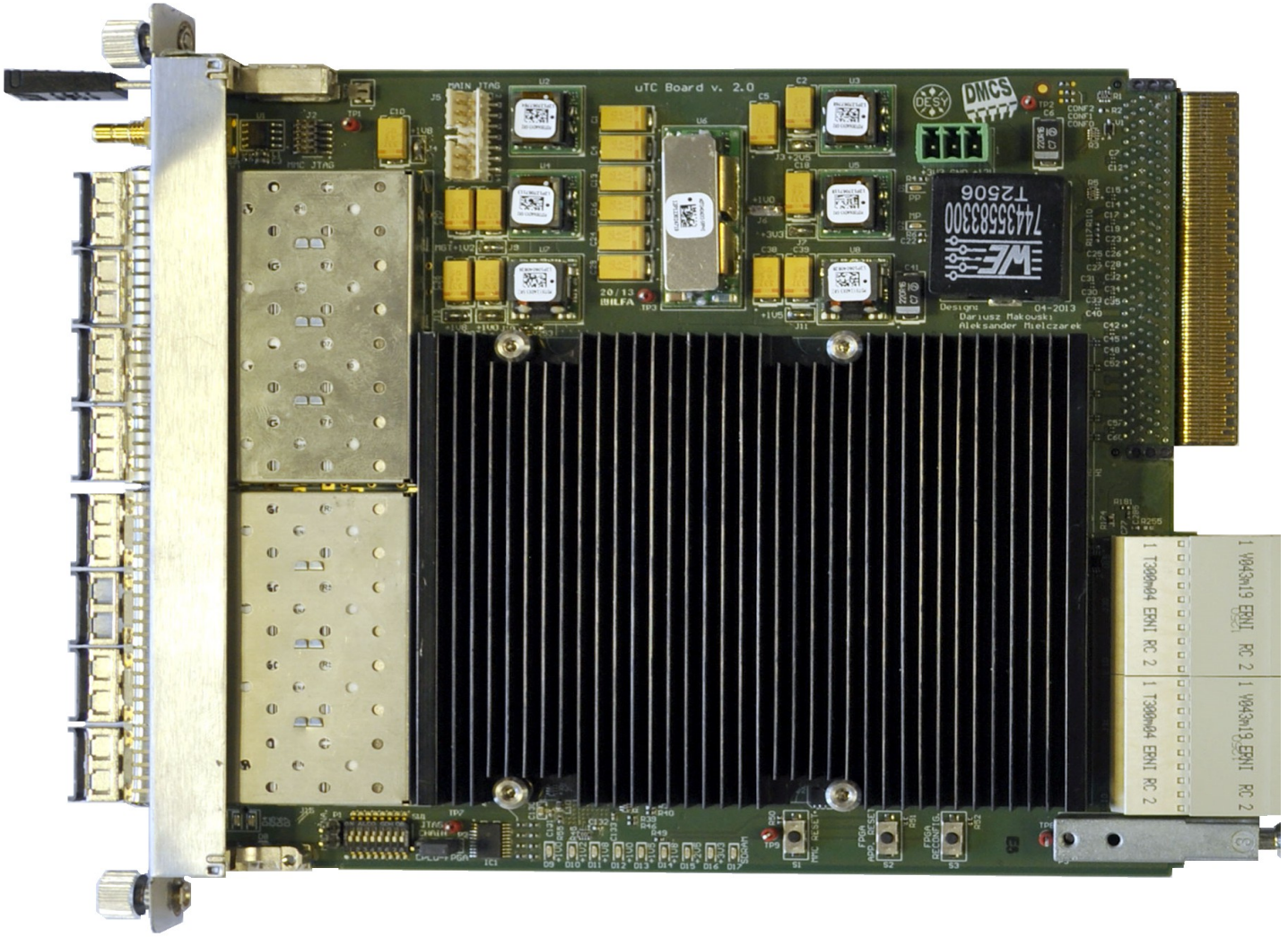
DAMC-TCK7 – LLRF Controller (1)

- Provides computation power for LLRF algorithms
- Provides interface to VM,diagnostics,management
- Module Requirements:
  - PCIe - main parameters of the computation module, latency <50 us, throughput ~16 Gbps
  - LLL- data from DAQ modules and to Vector Modulator, latency <200 ns, throughput >6 Gbps
  - Optical connection for communication with other LLRF subsystems
- Trigger and interlock signals provided on backplane
- Compliant with PICMG MTCA.4 specification
- Module management, diagnostics, high availability

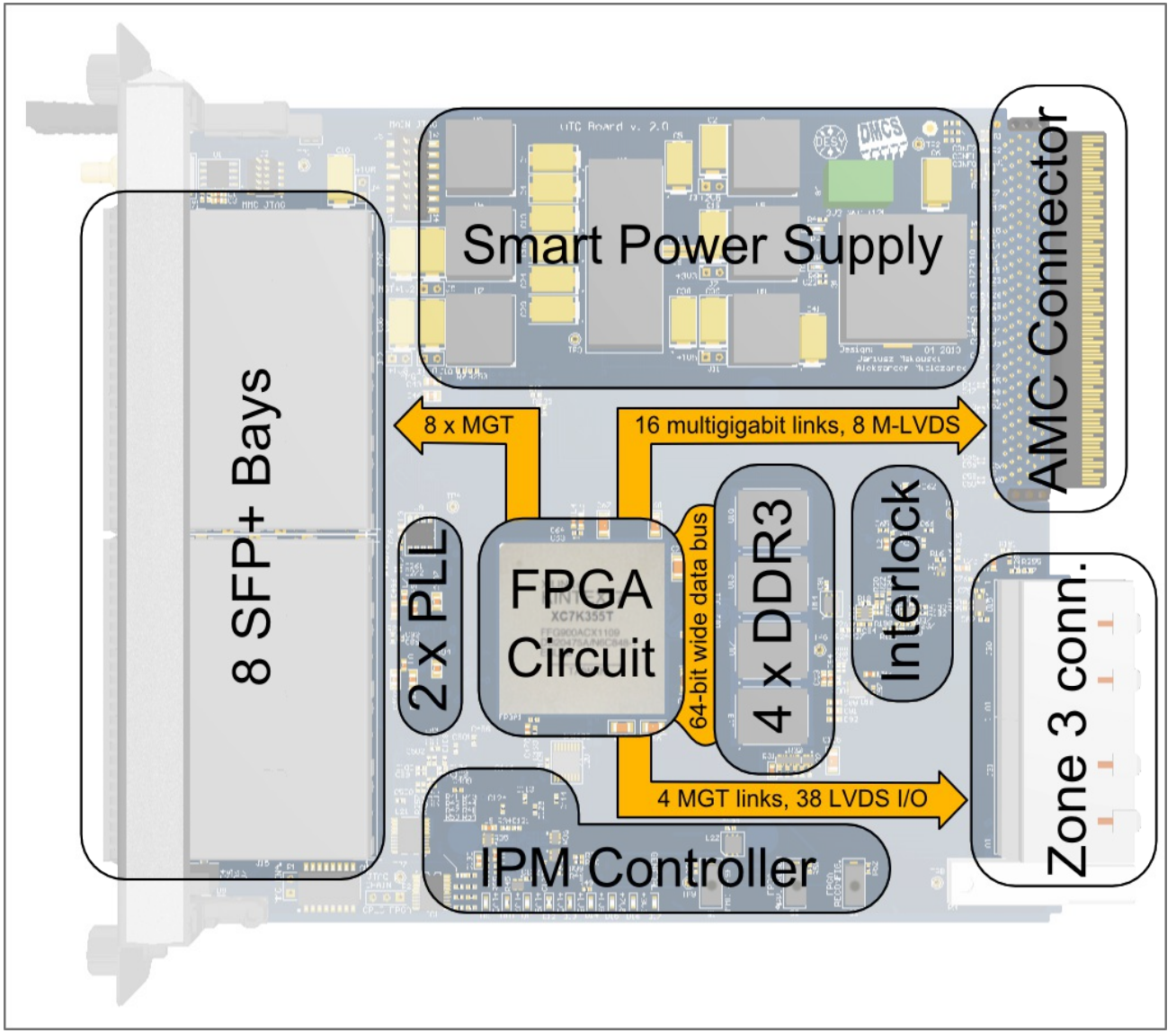




DAMC-TCK7 – LLRF Controller (2)

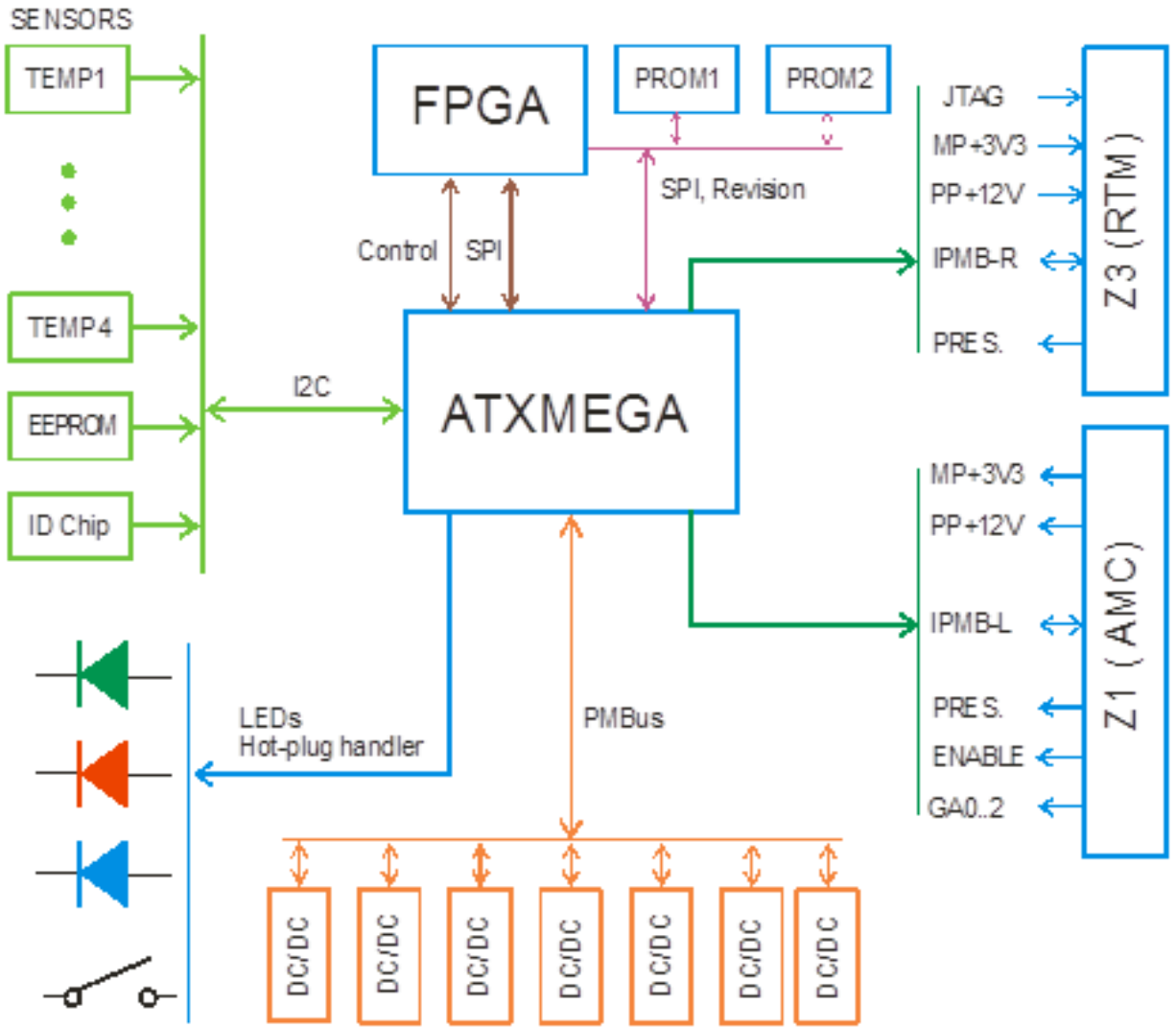


DAMC-TCK7 – LLRF Controller (3)



Zone 3
Class D1.2

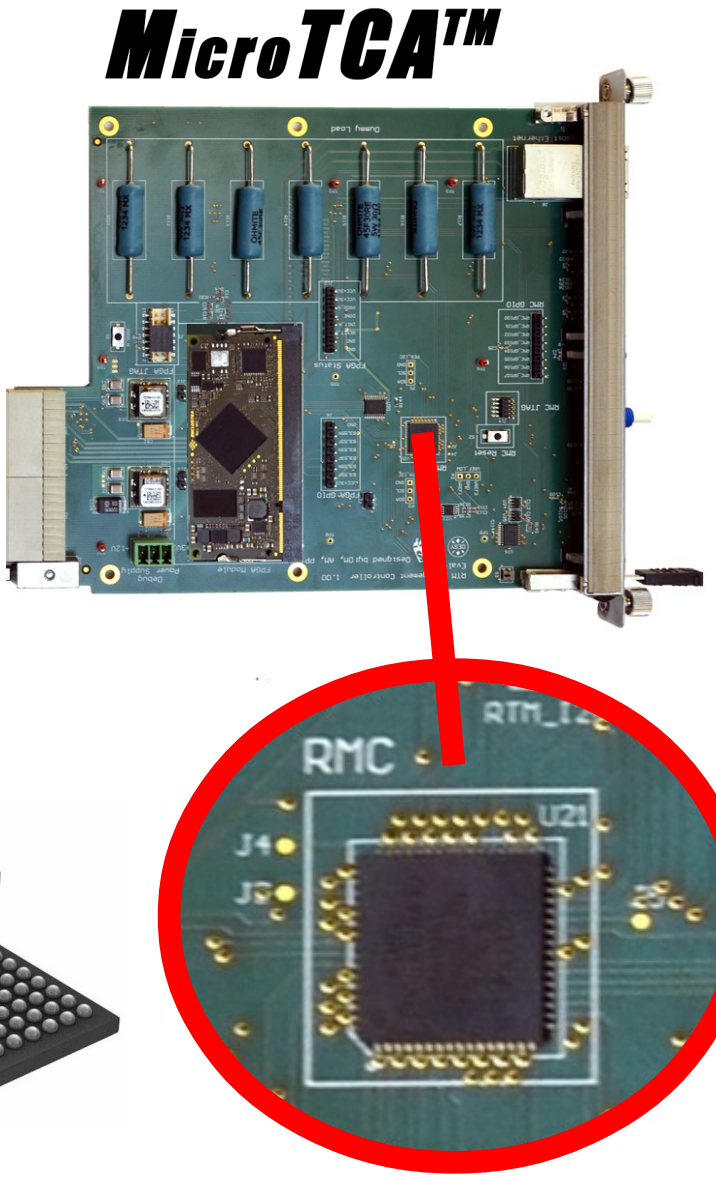
Block Diagram of MMC

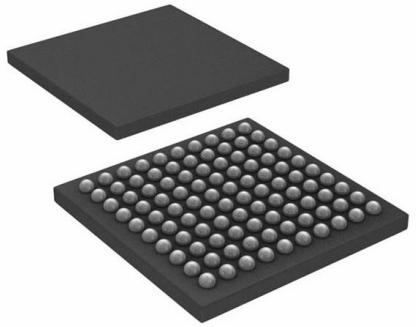


RTM Management Controller

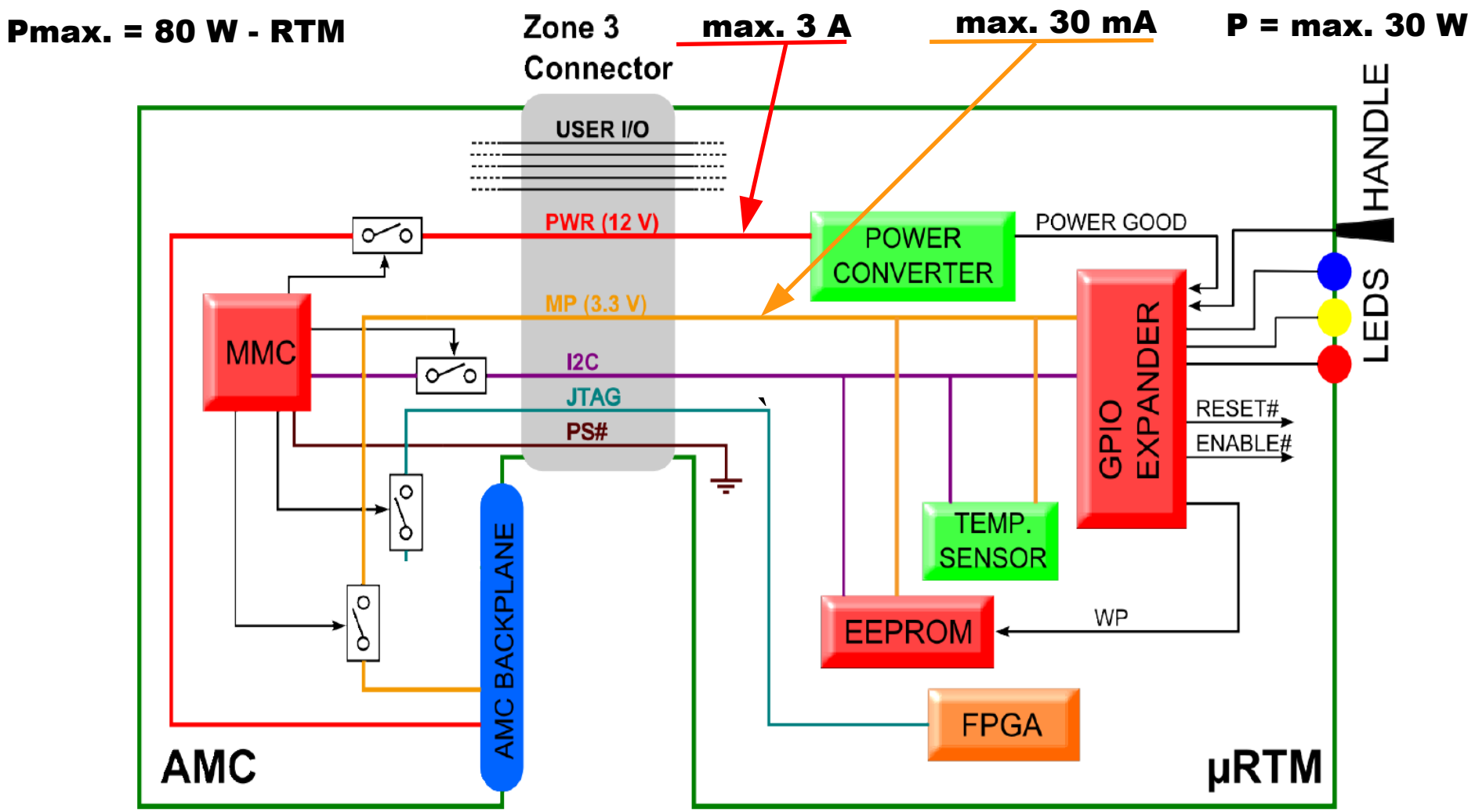
RTM Management Controller

- Required on each RTM Module
- Simple and Advanced solution
- Communication with MMC
- RTM management:
  - Module activation and deactivation
  - Warm and cold module reset
  - Power supply management
- Monitoring of module crucial parameters
  - Temperature
  - Supply voltages
  - Currents
  - Clocks, etc.
- E-keying mechanism (Zone 3)

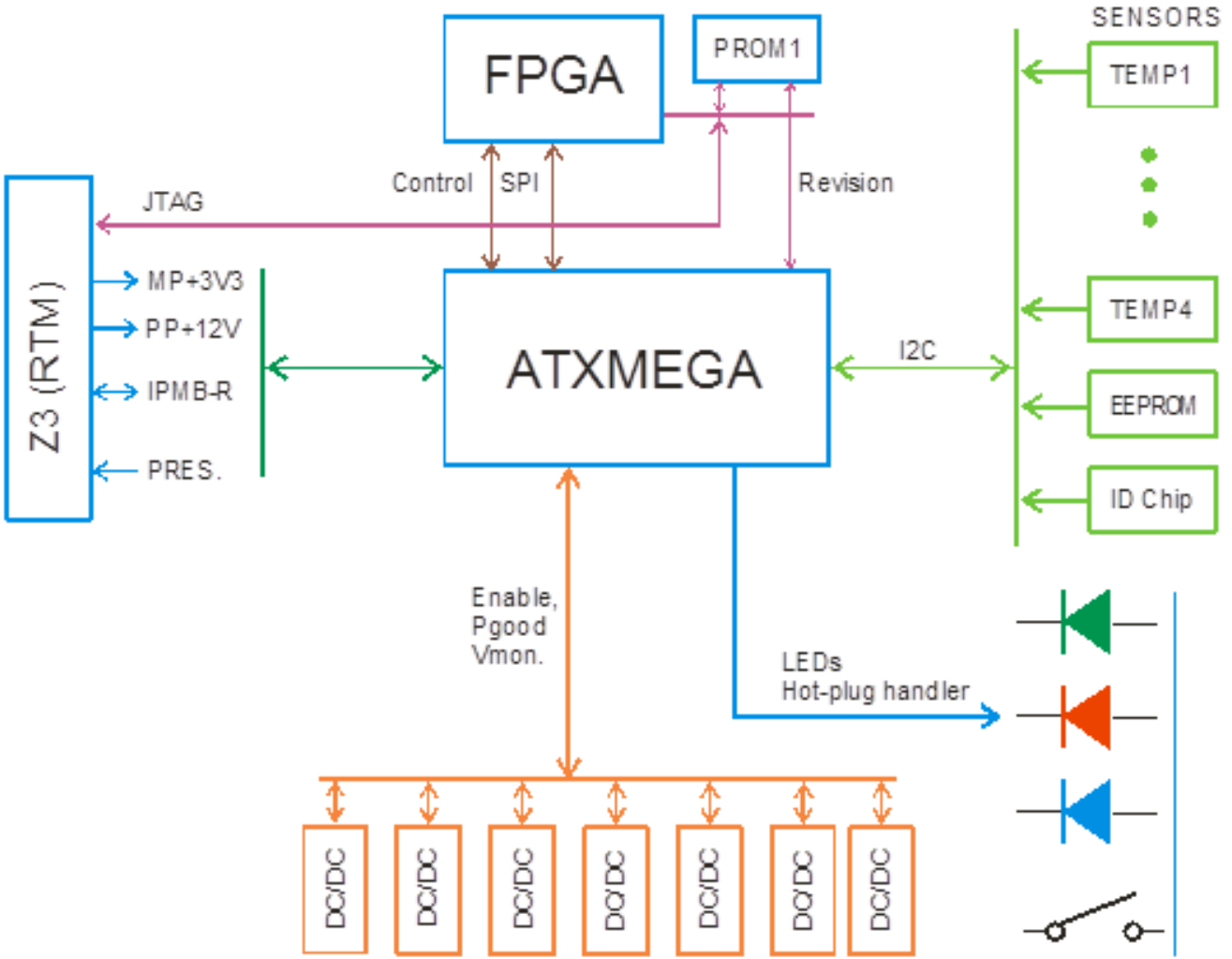




RTM Management – Simple Solution

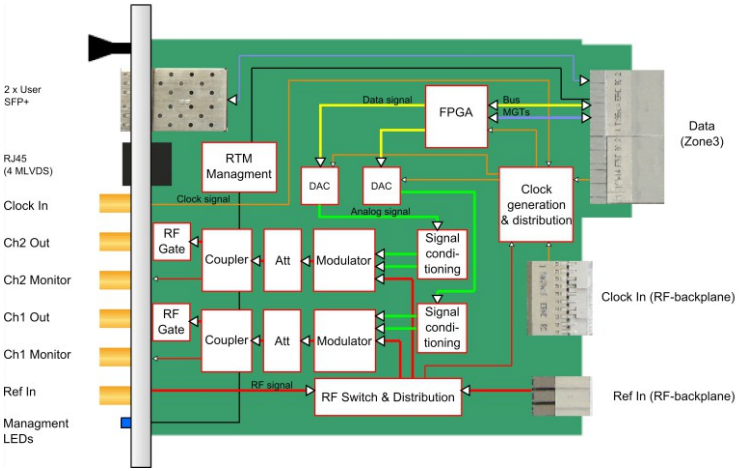


RTM Management – Advanced Solution

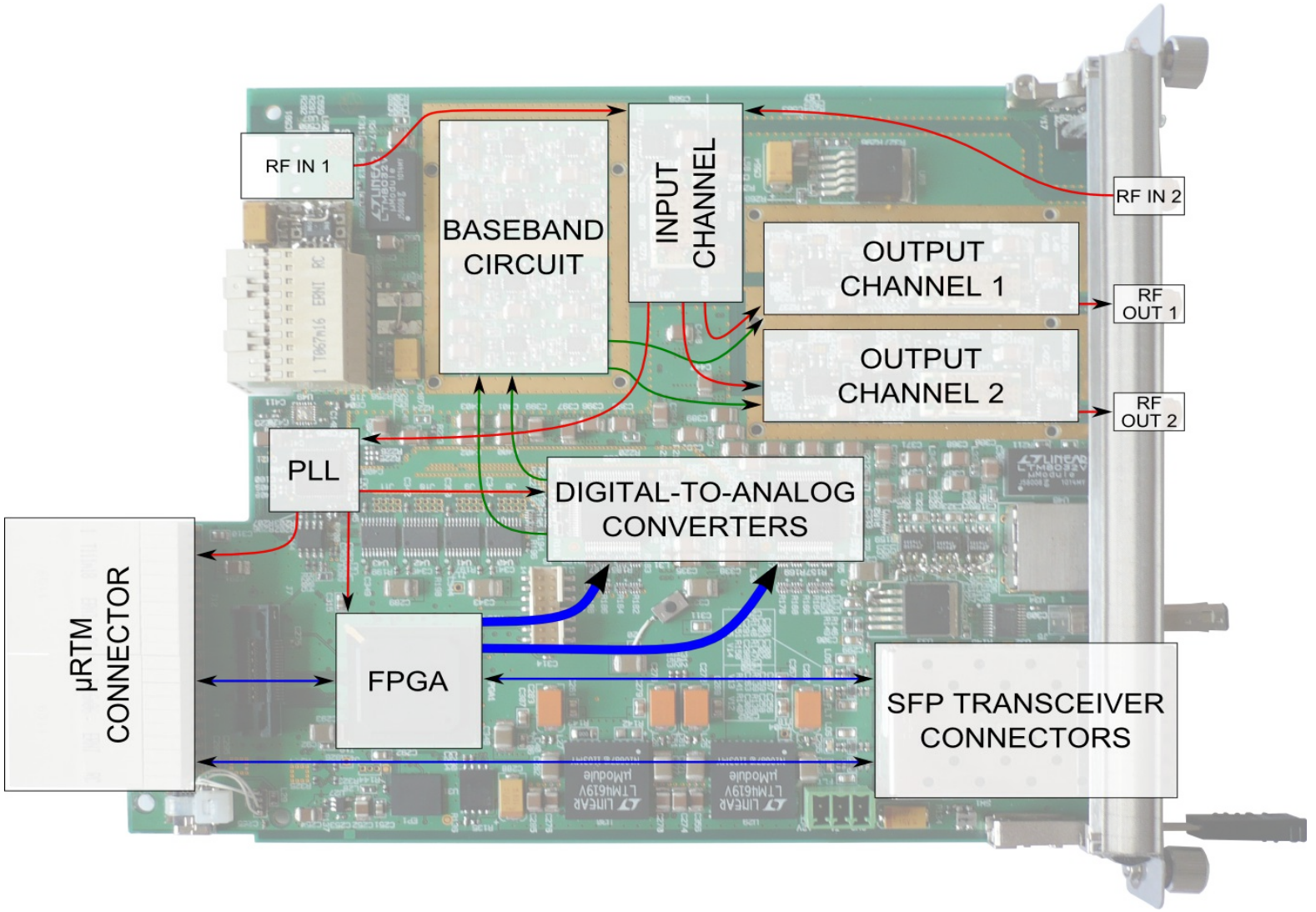


Vector Modulator Module

- Performs amplitude and phase modulation of RF reference signal to drive the klystron
- Module Requirements:
  - 2 channel high-frequency vector modulation at 1.3 GHz, 3.9 GHz
  - Non linearity error <-55 dBc(0.2%)
  - Short-term PM stability <10 fs(RMS) [10 Hz,100 kHz]
  - Spectral purity -155dBc/Hz
  - Calibration capability
  - 16-bit resolution on each I and Q channel
  - Programmable attenuators
  - Zone 3 D1.2 Class compatible
  - E-keying for Zone 3
  - Compliant with MTCA.4



DRTM-VMLF – Vector Modulator



IPMI MMC 1.00 Framework – Firmware for MTCA.4 Management

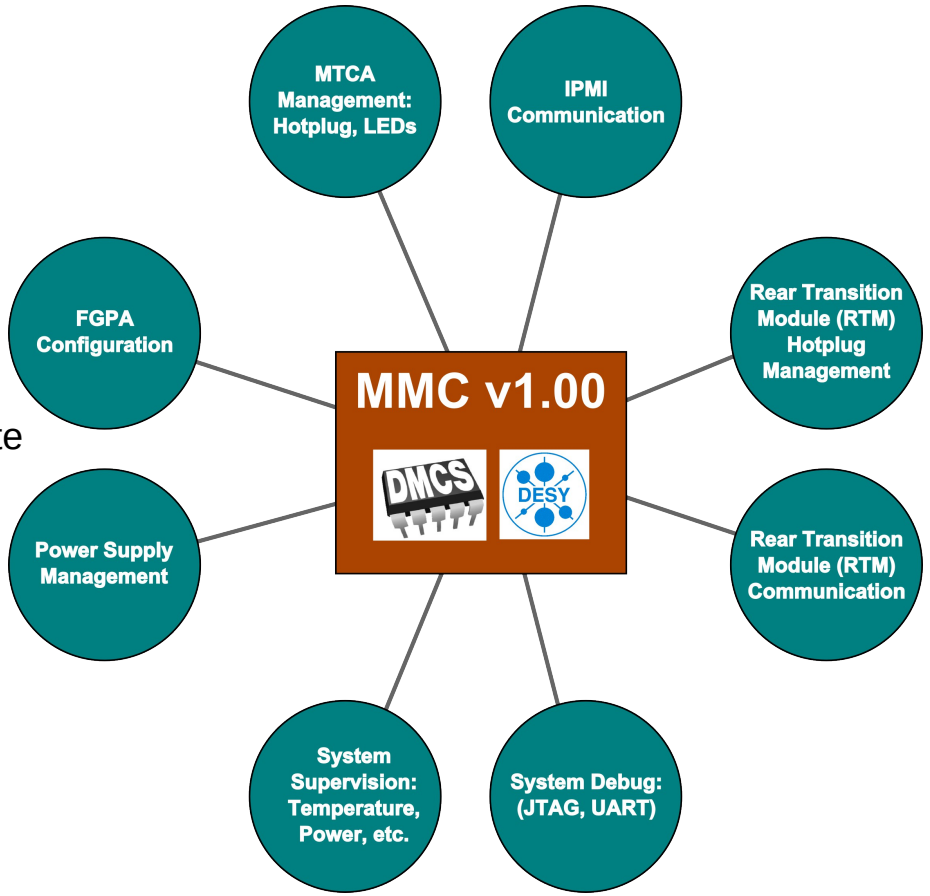
MMC v1.00 Hardware-Firmware – Unified Solution

MMC for AMC:

- Standard MTCA.4 compliant functions
- Monitoring of on-board voltages
- Protection in case of overheating
- UART for debugging and local monitoring
- Zone3 Isolation functionality
- Hot-plugging controller for µRTM Modules
- Management of on-board FPGA/CPU/DSP and firmware update
- MMC firmware upgrade using HPM.1

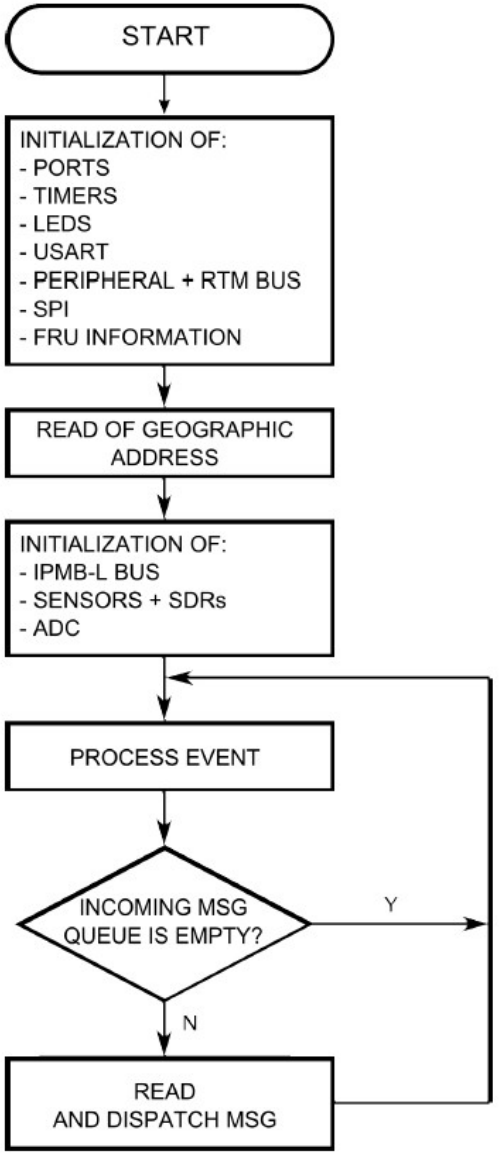
MMC for µRTM:

- Standard MTCA.4 compliant functions
- Monitoring of on-board voltages
- Protection in case of overheating
- UART for debugging and local monitoring
- Zone3 Isolation functionality
- Management of payload reconfiguration and firmware update



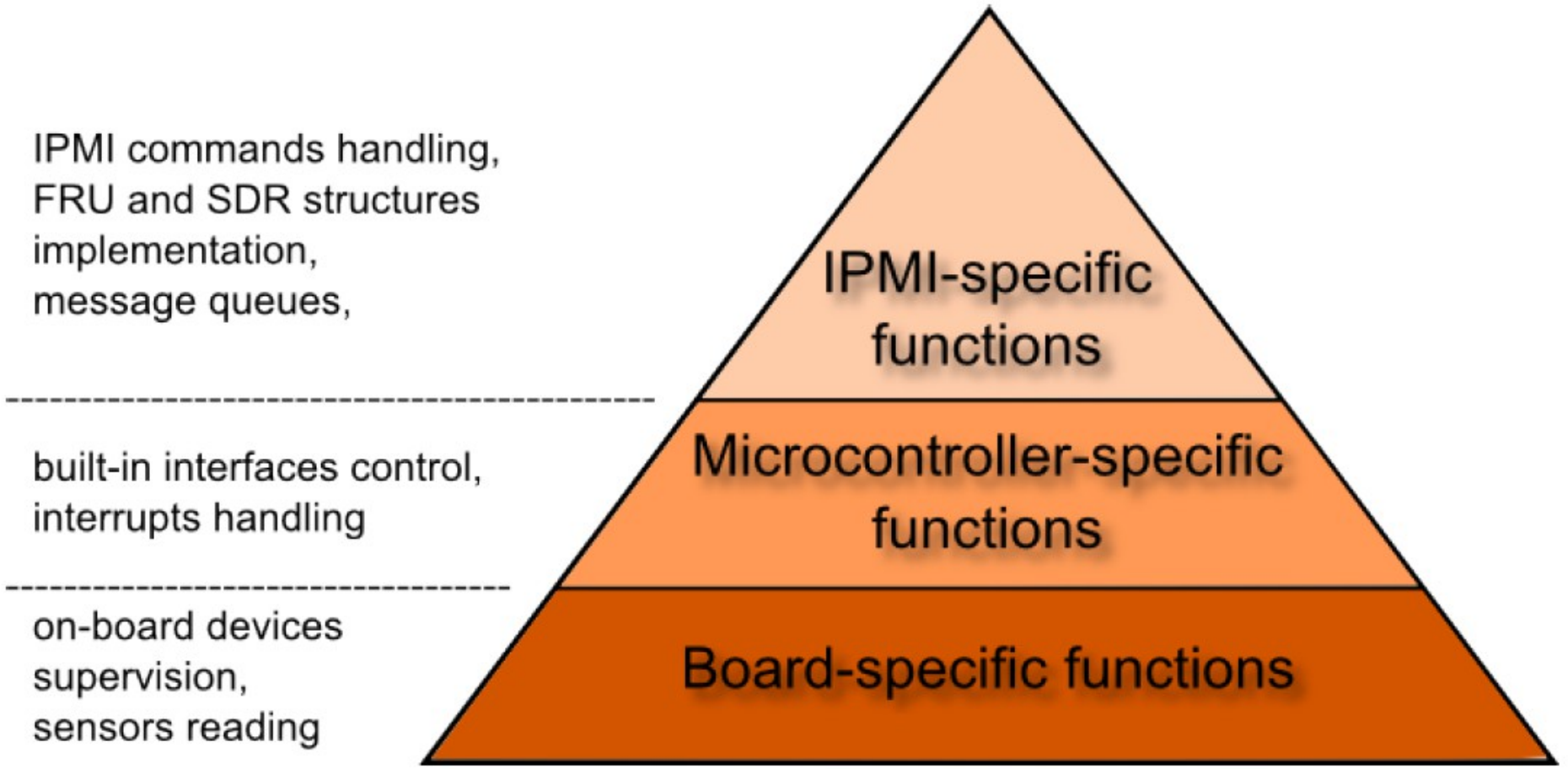
MMC – Implementation (1)

- Initialization
  - Initialization of software structures
  - Configuration of peripheral devices
- Main loop
  - Event processing
  - Messages handling
  - Updating sensors
  - Payload Control
  - Command Line Interface



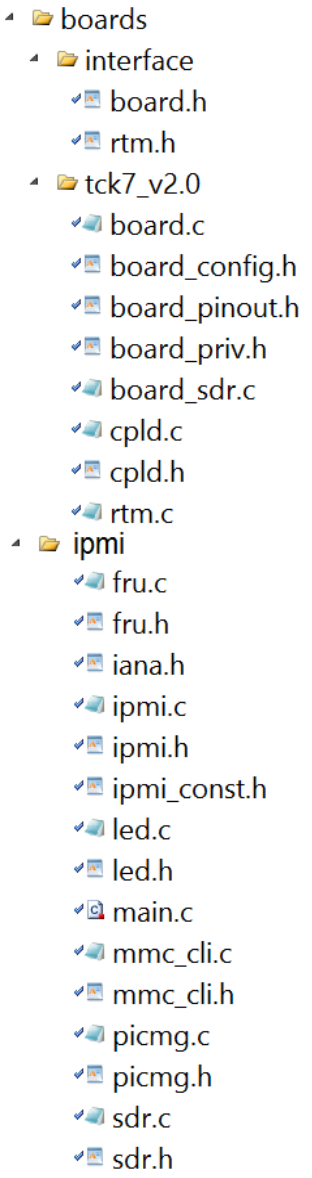
MMC – Implementation (2)

- Works aimed to code portability
- Software layers



MMC 1.00 Framework

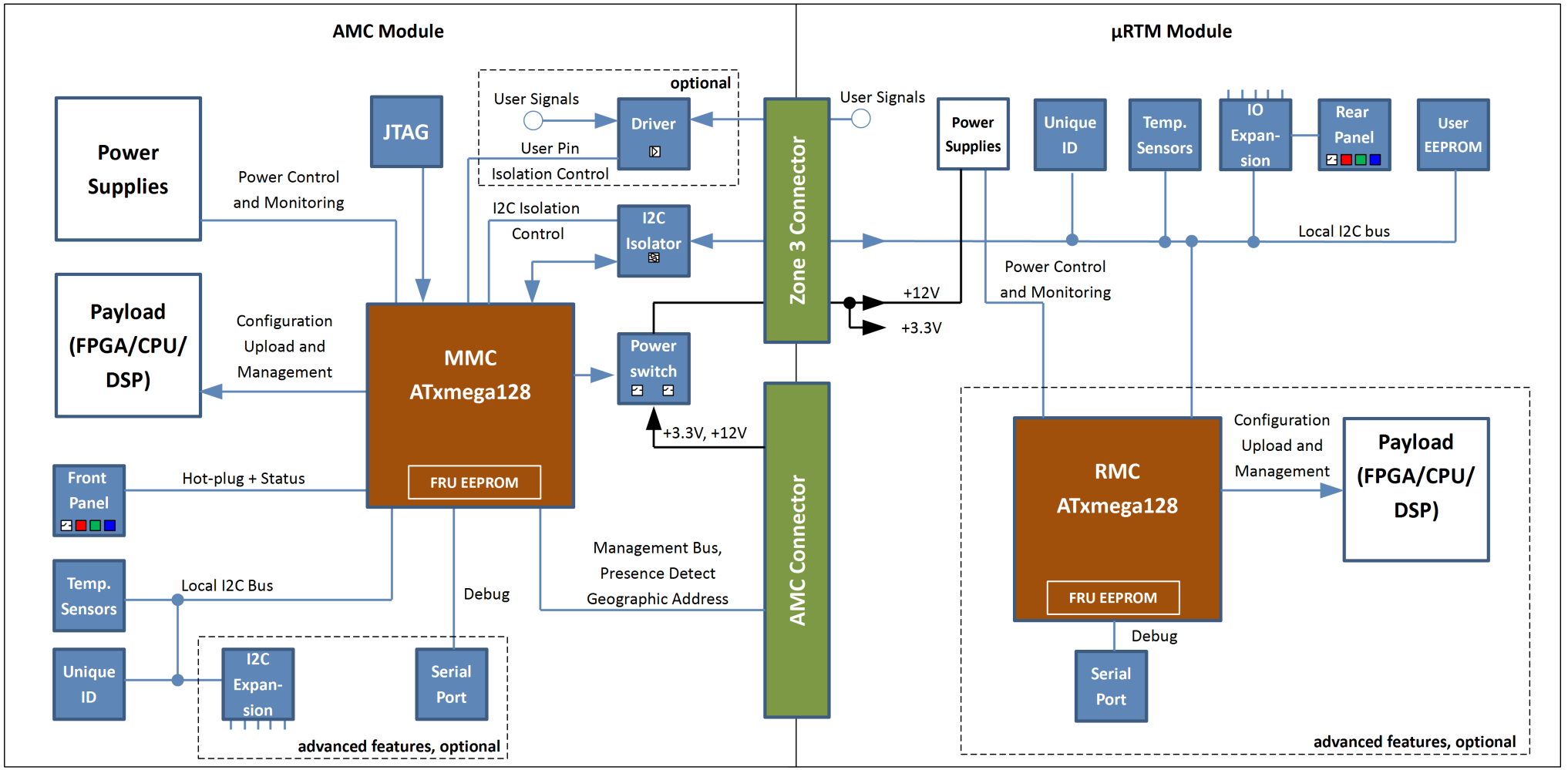
- IPMI core
  - IPMI events service
  - SDR, FRU, LEDs
  - PICMG
  - RTM
- Microcontroller specific functions
  - Firmware written using Atmel Software Framework (ASF)
  - Drivers and services for ATXmega microcontroller
  - Drivers for peripheral devices
  - Start-up code
- Board specific functions
  - Device drivers
  - Sensors
  - Payload management



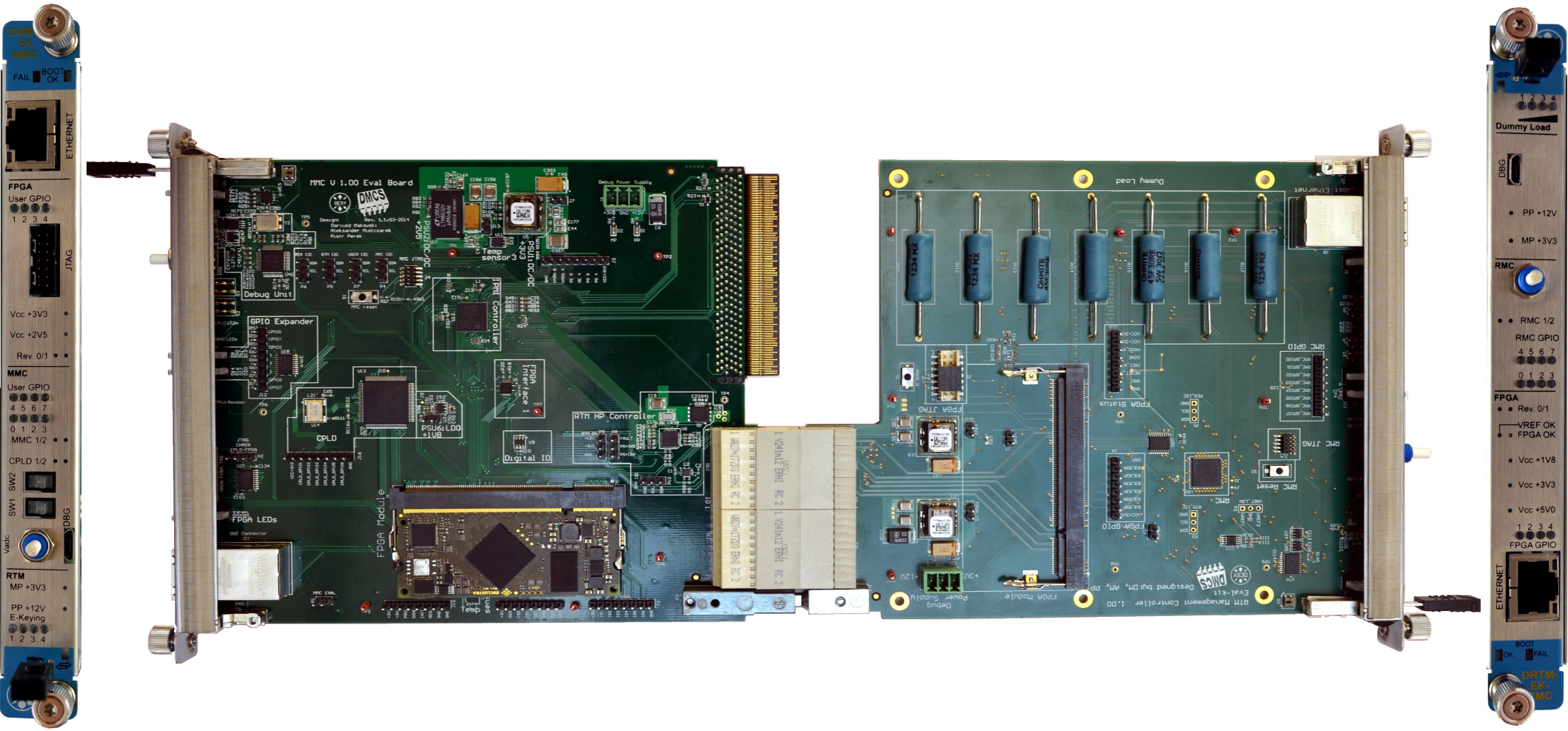

```
* @file board.c
* @brief Board implementation of board specif
*
* @copyright (C) 2014 Deutsches Elektronen-S
* @copyright (C) 2014 Lodz University of Tech
```

EvalKits for Fast Development of IPMI

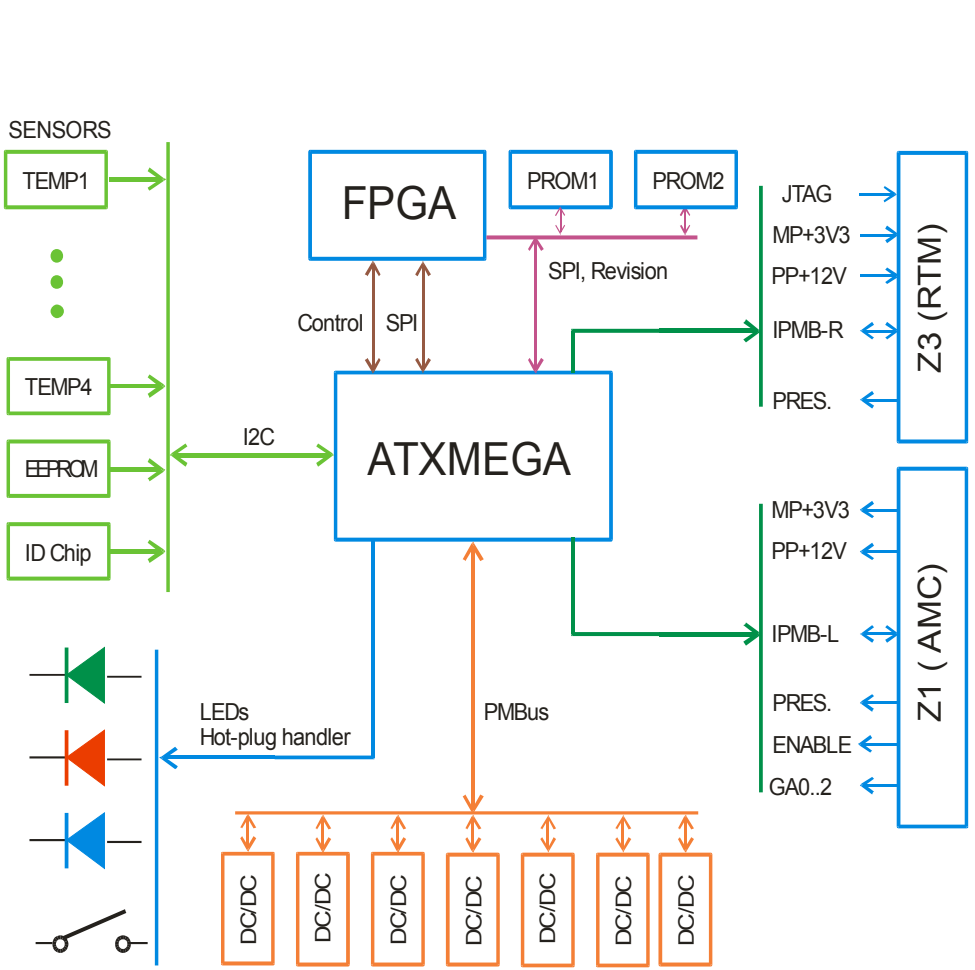
Block Diagram of MMC and RMC Eval-kits

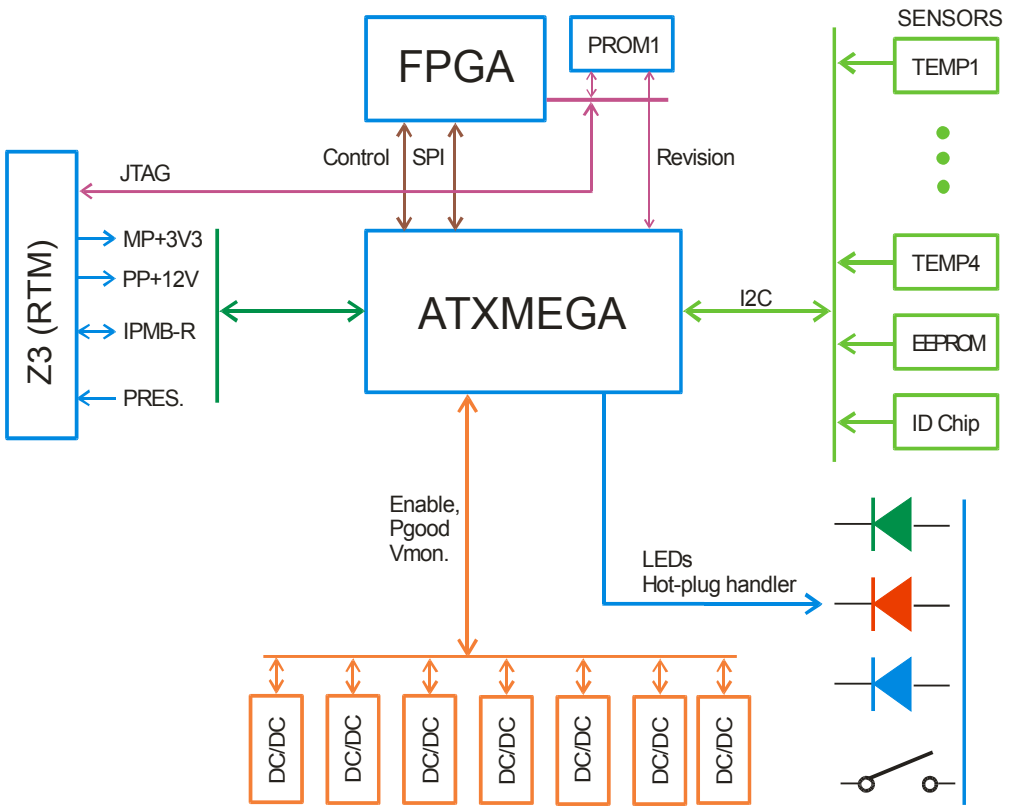


MMC and RMC Eval-kits Hardware



AMC and RTM Management





Thank you for your attention

Questions ?
Comments ?