Tutorial
MicroTCA Management

"How to become a MicroTCA expert – this year: within 25 minutes"

1st MicroTCA Workshop

at

Large Scientific Facility Control Workshop

by

University of Science and Technology of China

Heifei, September 18th – 21st, 2024

UNCLASSIFIED

Agenda

- **About N.A.T.**
- From ATCA to MTCA
- Why do we need management?
- What is behind the management?
- How does it work?
- What can you do?

About N.A.T. – who we are

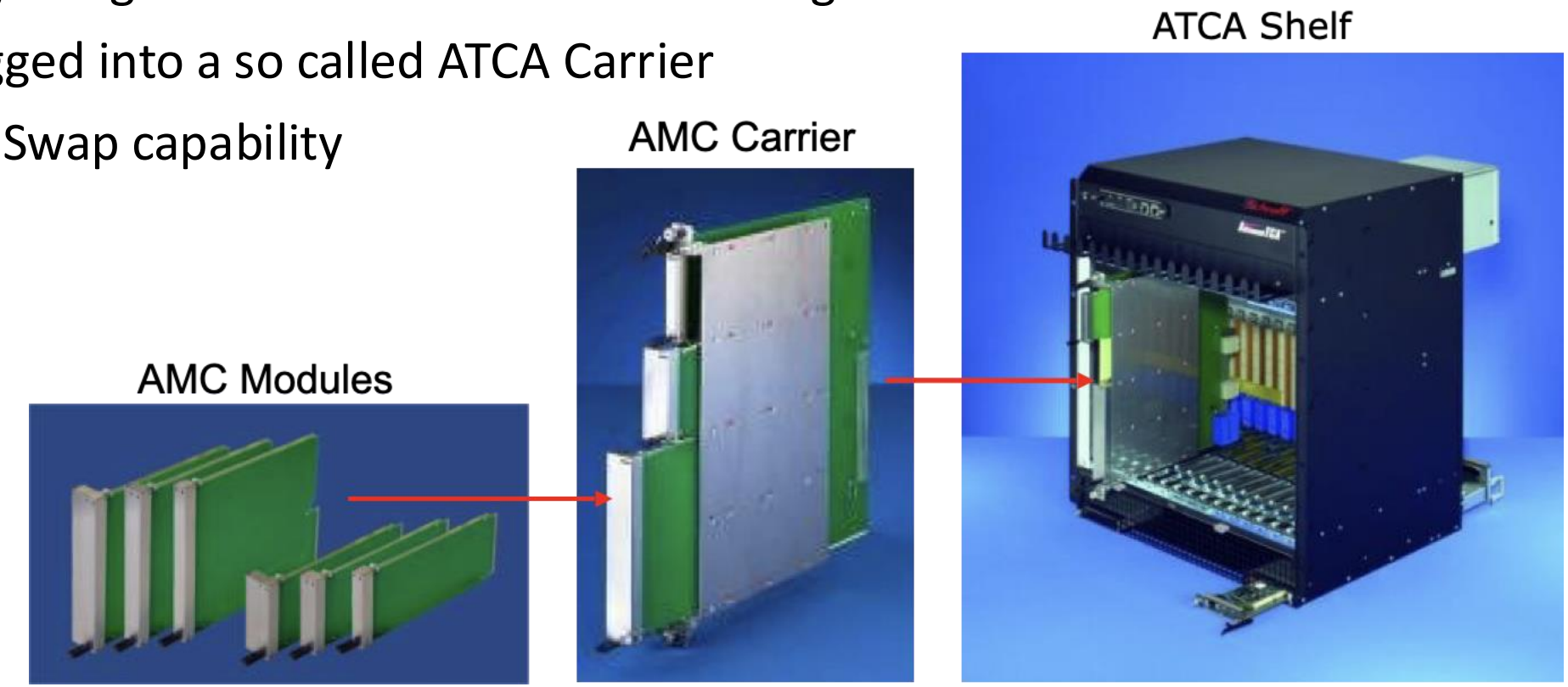
- Gesellschaft für **N**etzwerk- und **A**utomatisierungs-**T**echnologie mit beschränkter Haftung => **N.A.T.**
- Founded in 1990
- Proud to provide quality ”made in Germany”
  - since more than 33 years by 25 highly professional employees
- Privately owned and owner lead business
- Own purpose-built building of more than 1,600m^2 (17,222ft^2) with on-site centers for
  - hardware and software design
  - pre-manufacturing , test + repair

Agenda

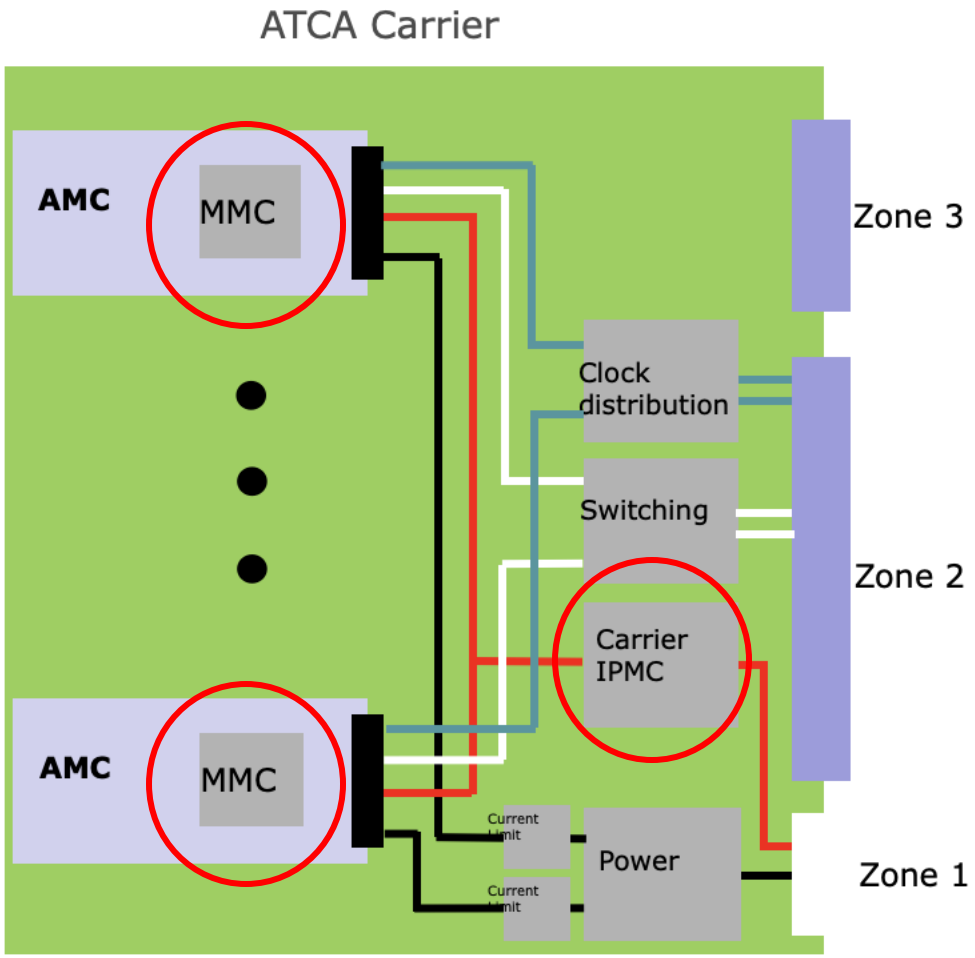
- About N.A.T.
- **From ATCA to MTCA**
- Why do we need management?
- What is behind the management?
- How does it work?
- What can you do?

Quick Look At ATCA History for AMC (mezzanine cards)

- Initially developed as function extension for ATCA Boards
- Fully integrated into the ATCA IPMI management structure
- Plugged into a so called ATCA Carrier
- Hot Swap capability

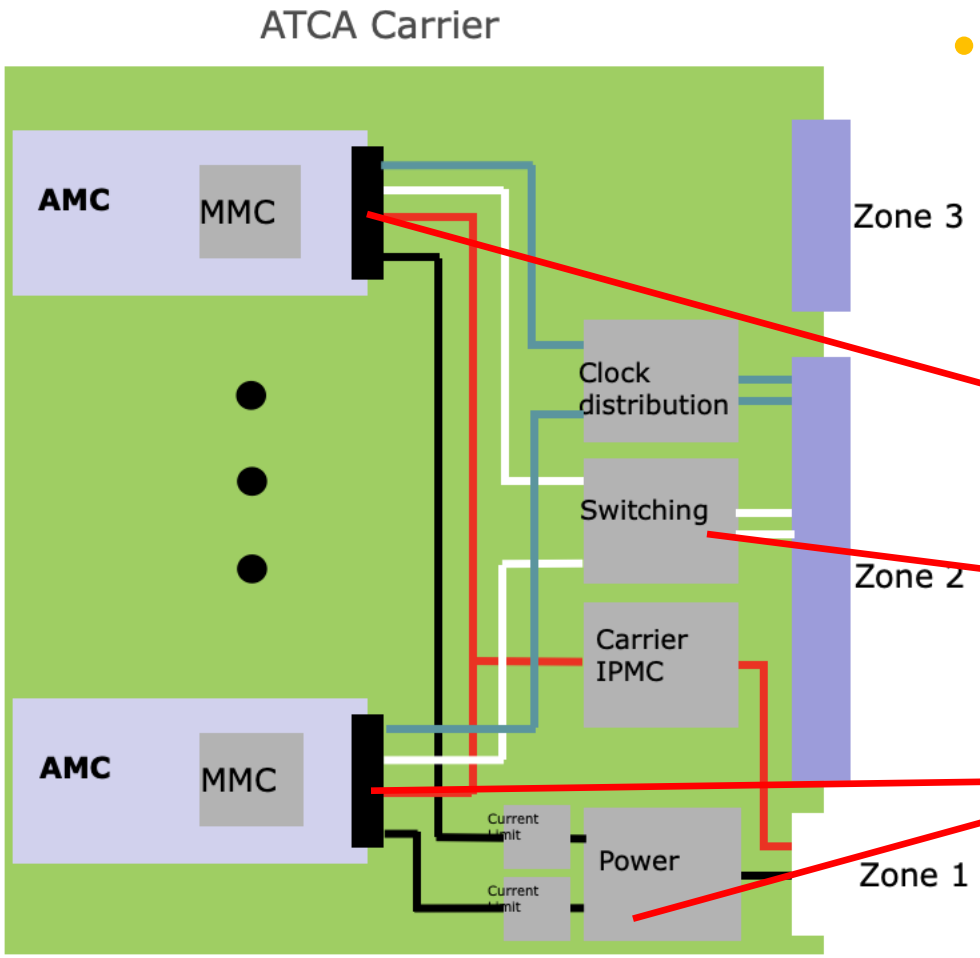


Quick Look At ATCA ATCA carrier – the environment an AMC module lives in



- Idea for own system like MTCA:
  - directly plug AMC onto backplane
- Requires "virtual carrier" providing
  - Power, management, CLK, Switching

Quick Look At ATCA How to migrate from ATCA to MTCA



- "virtual carrier" consists of
  - MTCA chassis incl. backplane and CUs
  - Power Module
  - MCH for management, CLK, switching

Agenda

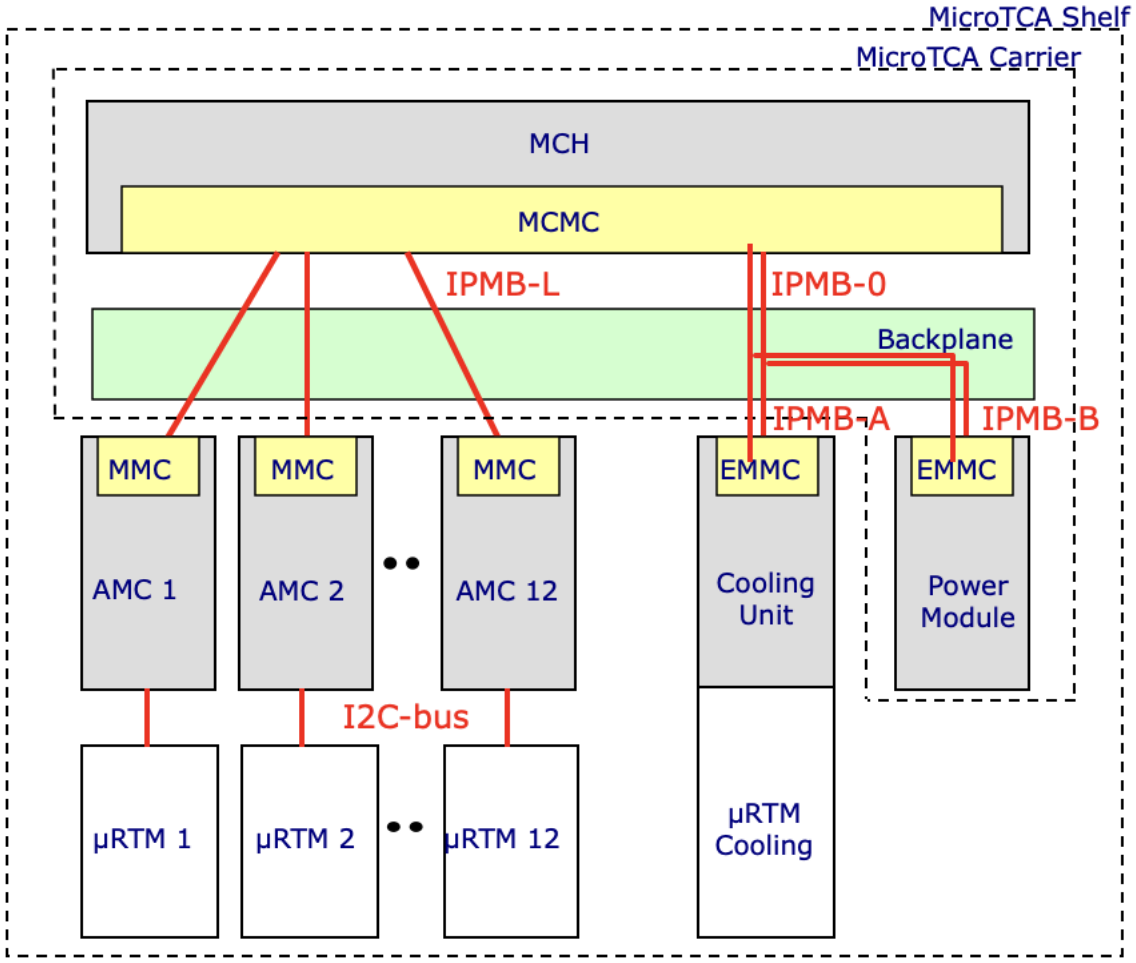
- About N.A.T.
- From ATCA to MTCA
- **Why do we need management?**
- What is behind the management?
- How does it work?
- What can you do?

Management Why do we need it?

- “Who” is in my system?
  - i.e. list of devices (aka “FRU” for **F**ield **R**eplaceable **U**nit)
- What capabilities does the FRU have?
  - i.e. active connections (AMCs) or RPMs (CUs)
- How healthy is my system?
  - i.e. sensors for current, voltage, temperature
  - i.e. events
- How can I talk to my FRUs?
  - i.e. manipulation of sensors
- How can I service my system?
  - i.e. hot-swap FRUs

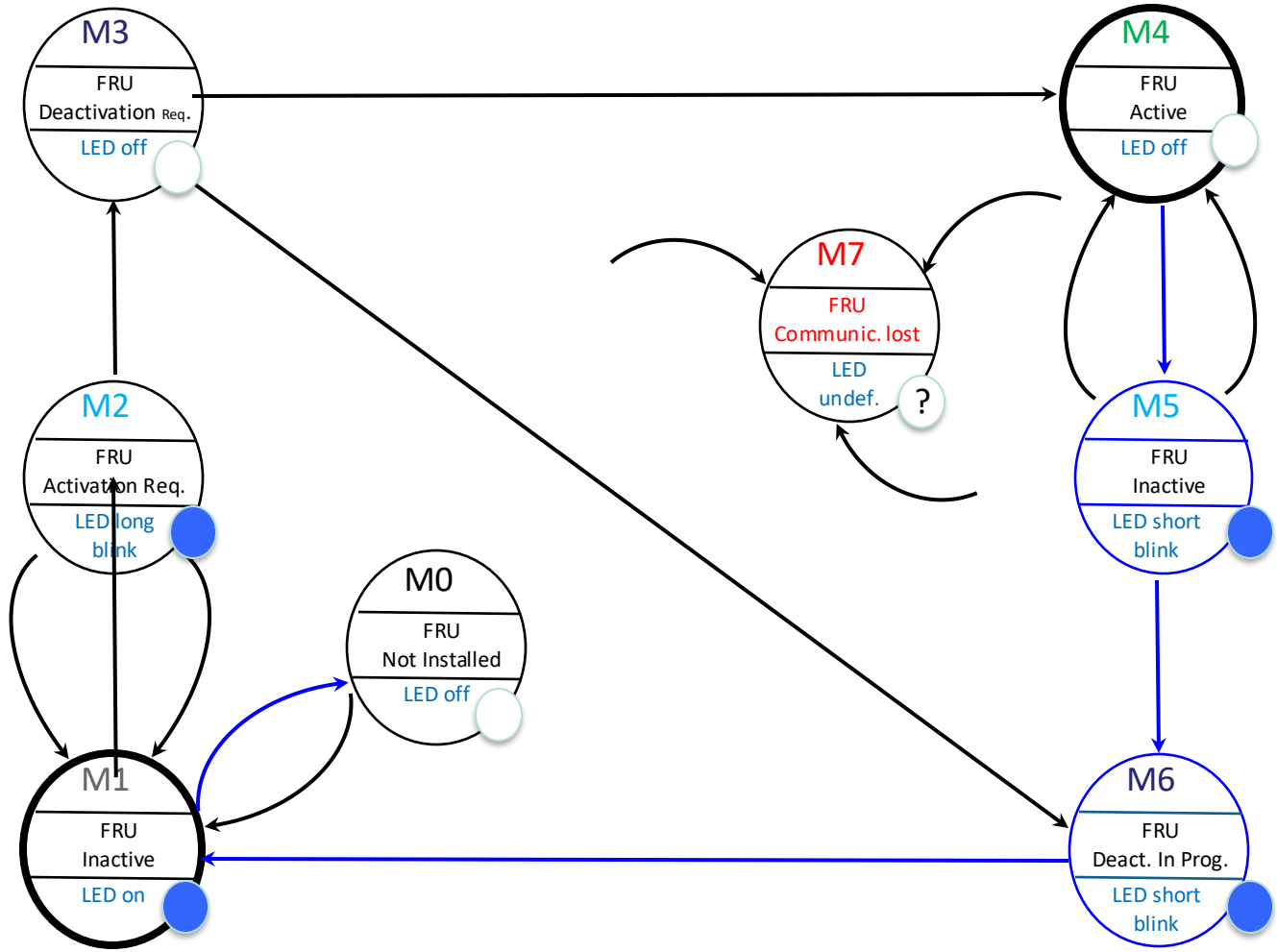
Management in MTCA Physical Connections And Controllers

- IPMB-L
  - connects the MCMC on the MCH to the MMC on the AMC Modules
  - radial architecture
- IPMB-0 .1
  - connects the MCMC on the MCH to the EMMC on the PMs and CUs
  - bussed architecture
- I2C-bus
  - connects the AMC to its µRTM
  - the µRTM is treated as managed FRU of the AMC

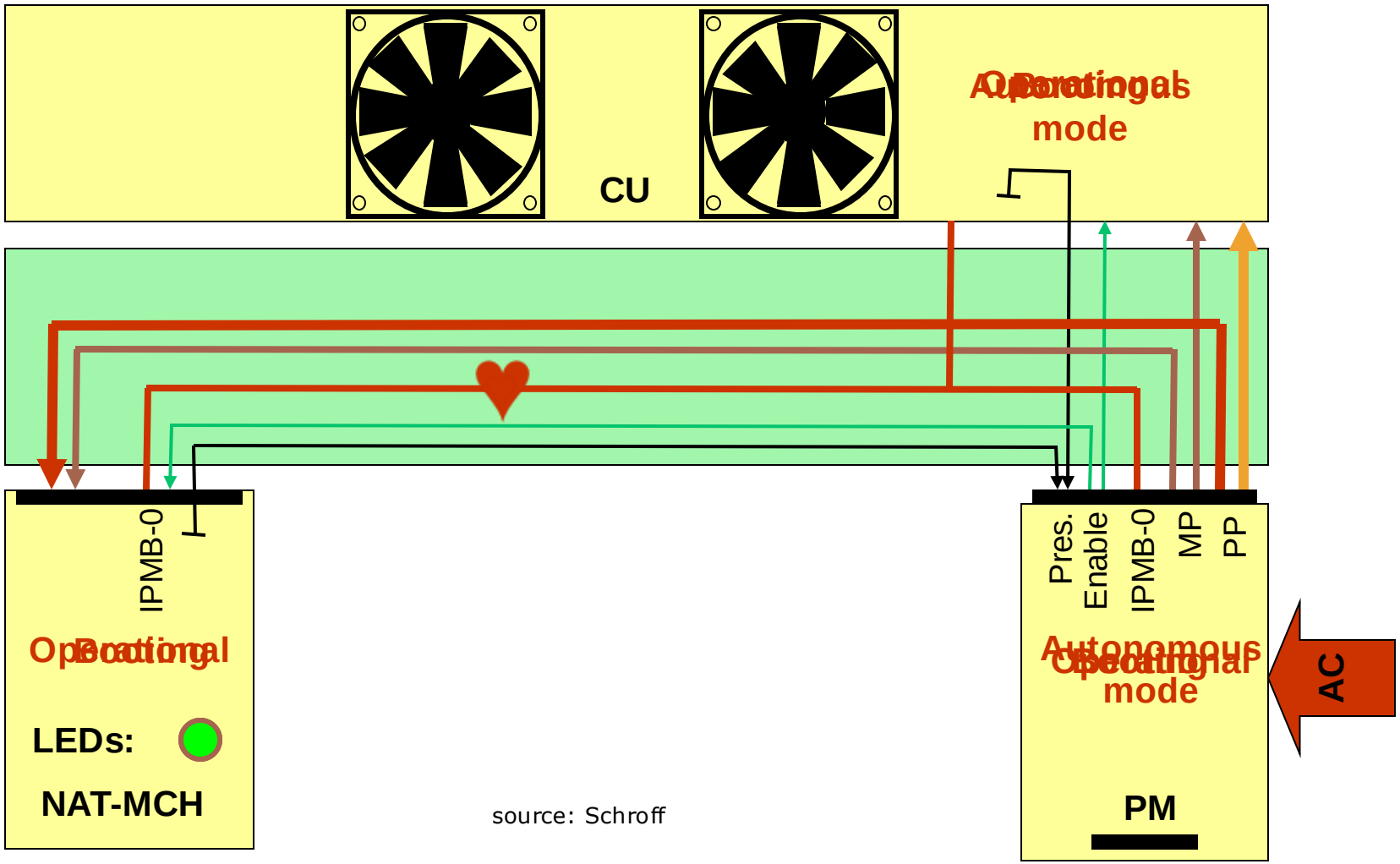


Management in MTCA FRU M states

- PICMG 3.0 and AMC specifications define FRU states, aka „M states“
  - Activation
    - FRU proceeds to state M4
  - Deactivation
    - FRU proceeds to state M1
  - Error (coms lost)
    - FRU moves to state M7
- MCH decides if and when module can reach M4
- MMC uses a state machine to control hot-plug/swap procedure

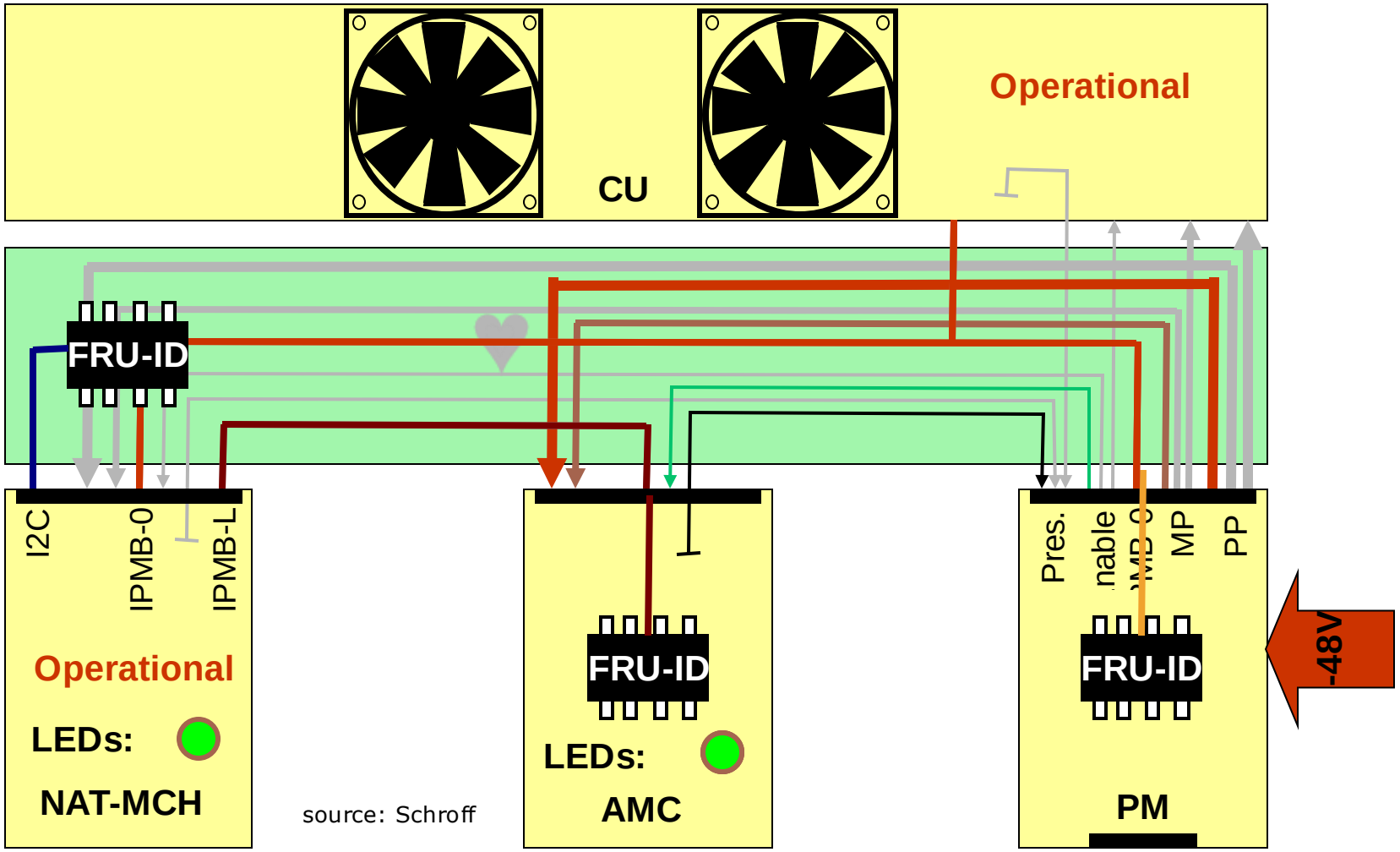


Infrastructure Power up



source: Schroff

Payload power up



 © 2023 N.A.T. GmbH | UNCLASSIFIED | All trademarks, brands and logos are property of their respective owners

Agenda

- About N.A.T.
- From ATCA to MTCA
- Why do we need management?
- **What is behind the management?**
- How does it work?
- What can you do?

Management in xTCA What is behind

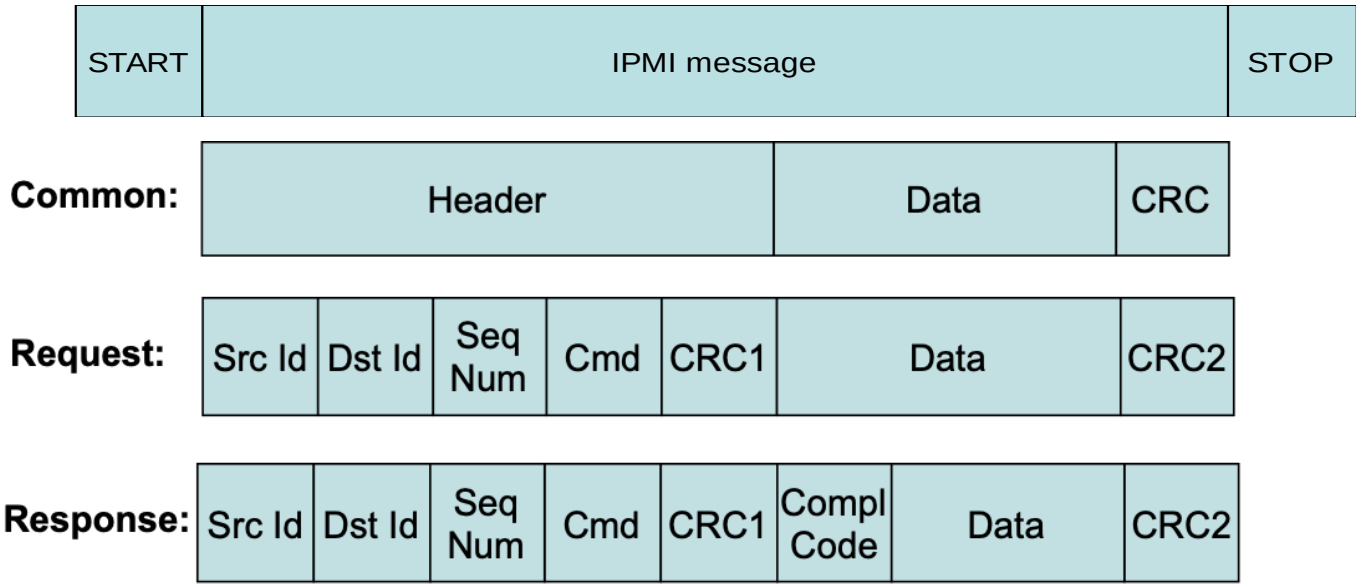
- Idea of management:
  - Hardware supervision by software (remote control and monitoring)
  - Intelligent handling of events and actions
  - Abstraction of hardware functionality
  - Operating system independent

=> I^2C (**I**nter **I**ntegrated **C**ircuit)

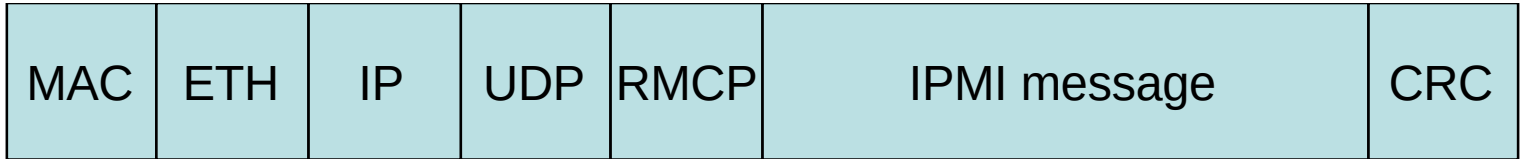
=> IPMI (**I**ntelligent **P**latform **M**anagement **I**nterface)

Management in xTCA IPMI

- I²C (**I**nter **I**ntegrated **C**ircuit): two wire multi-master capable bus
- IPMI protocol



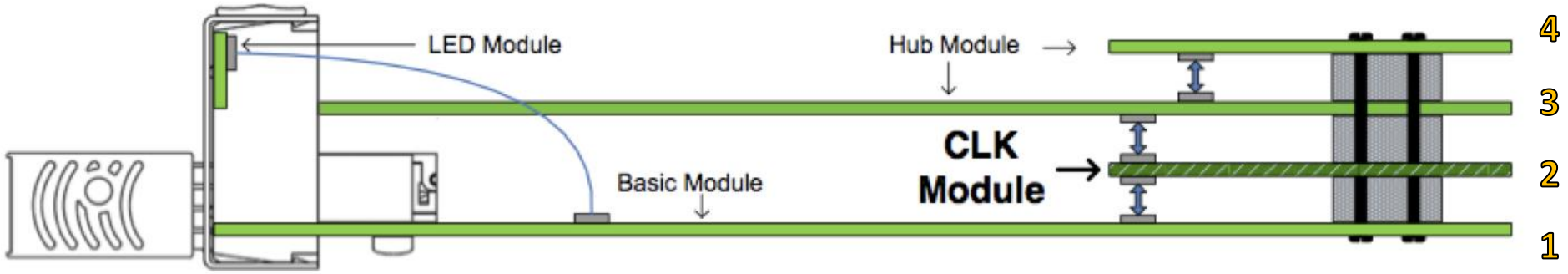
- RMCP (**R**emote **M**anagement **C**ontrol **P**rotocol)



Agenda

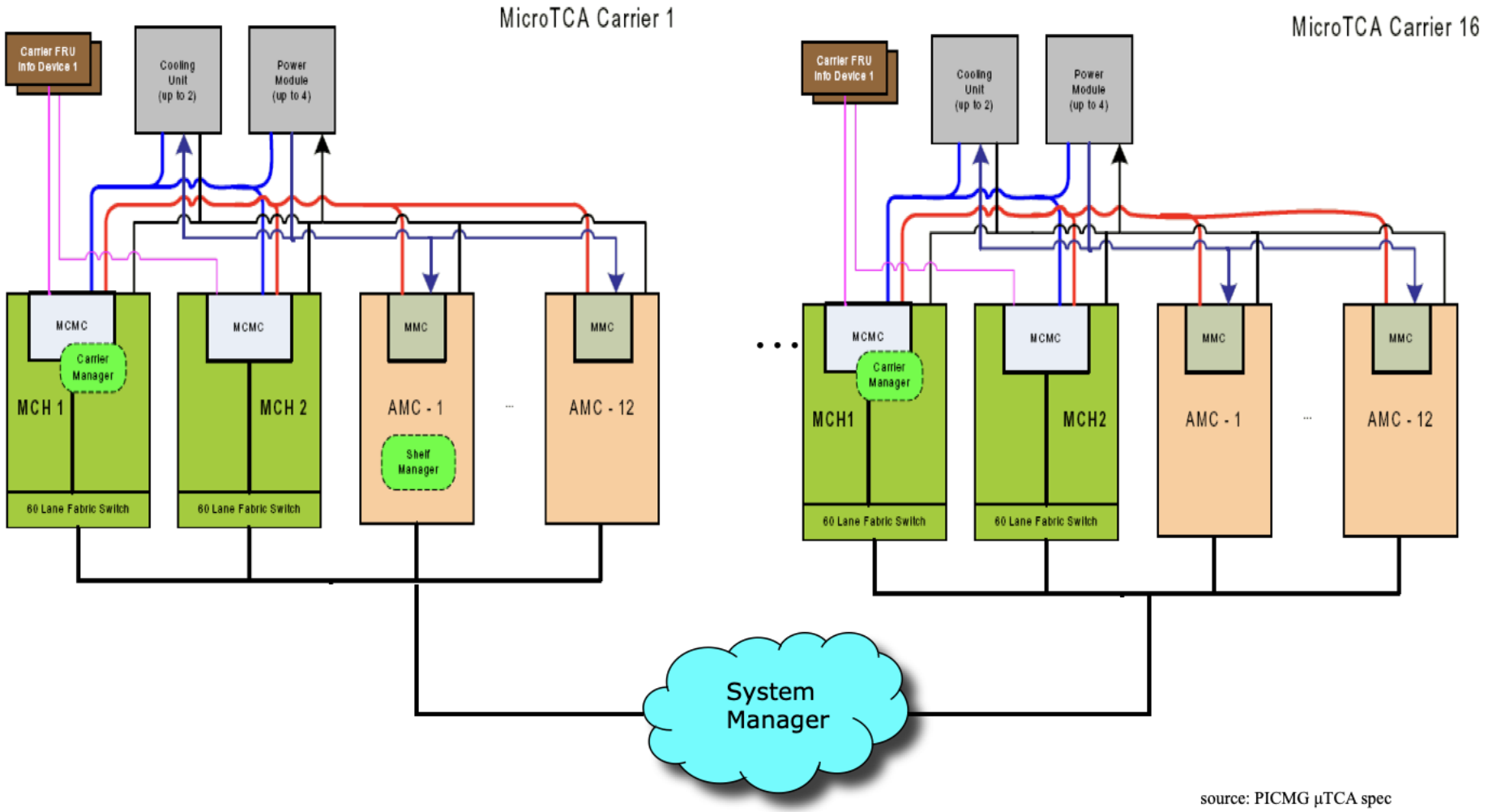
- About N.A.T.
- From ATCA to MTCA - two well connected standards
- Why do we need management?
- What is behind the management?
- **How does it work?**
- What can you do?

MTCA Carrier Hub (MCH) Adaptable to application demands



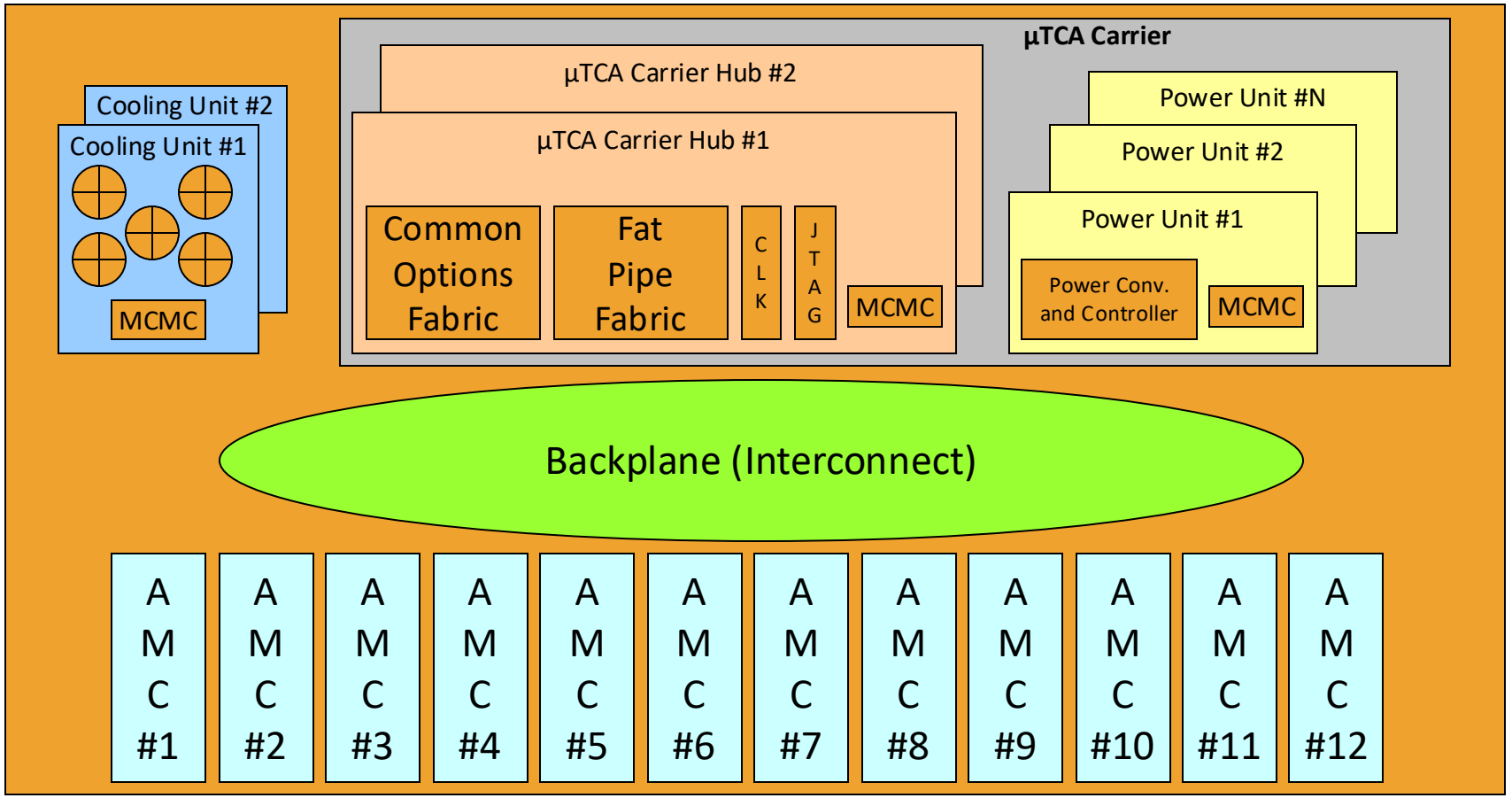
1. Basic Module with GbE-Switch to all AMC slots and **Management: carrier manager, shelf manager, system manager**
2. Clock Module for CLK #1-3/ #1,2,F to all AMC slots
3. Fat Pipe Hub Module (3) for AMC slots #1-6
4. Fat Pipe Hub Module (4) for AMC slots #7-12

Management in MTCA Management Structure



source: PICMG µTCA spec

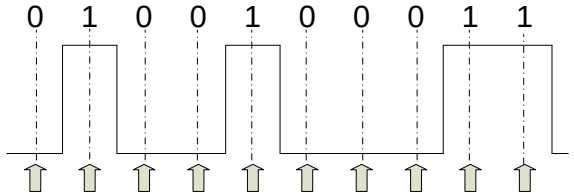
Excursus: fat pipes and clocks within a MicroTCA system



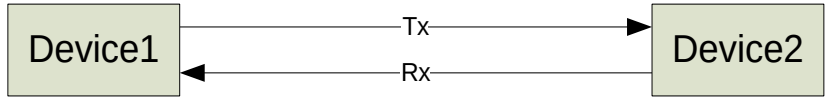
Excursus: fat pipes

- Bits are transmitted one after the other, over single data line
- Every data byte (8bit) is transformed to 10bit symbol that contains enough transitions

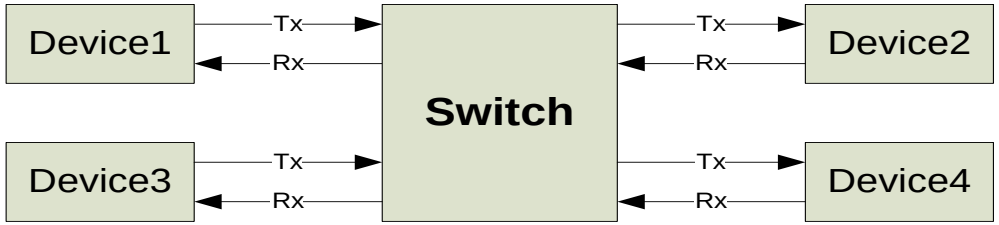
  → **8B/10B Coding**



- Clock is recovered from serial stream
- Bidirectional transmission via dedicated Tx and Rx lines
  - One Tx/Rx pair is called "**Lane**"



- Multiple Devices interconnect by **switches**



Excursus: fat pipes within a MicroTCA system

- Fat pipes aka fabrics
- Defined by PICMG AMC.x series
  - AMC.0 – Base Specification (with MTCA.0 Ref 3 some changes)
  - AMC.1 – PCI Express (PCIe): gen 1, gen 2, gen 3, (gen 4/5)
  - AMC.2 – Ethernet: 1GbE, XAUI, 10GbE, 40GbE
  - AMC.3 – Storage (SAS)
  - AMC.4 – Serial RapidIO (SRIO)
- Link width: x1, x2, x4, lanes aka “ports”
- Compatibility between AMC and switch on MCH ensured by e-keying
- All signal levels are LVDS => incompatibility could not cause damage

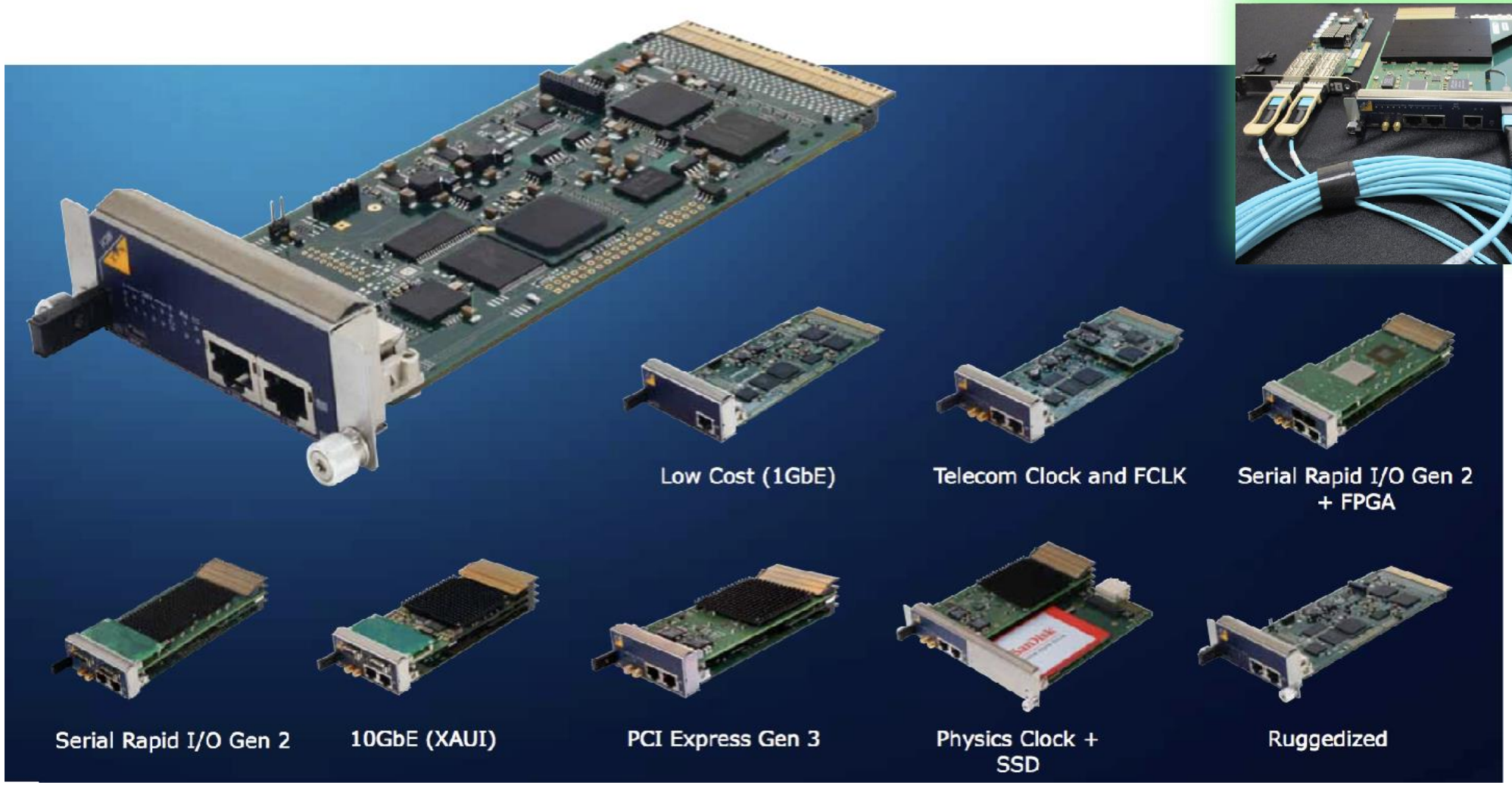
Excursus: clocking within a MicroTCA system

- Defined by PICMG MTCA.0 and AMC.0
  - frequency limited to 100MHz by spec
  - from an MCH perspective: CLK1, CLK2, CL3
  - from an AMC perspective: TCLKx and FCLK
  - mapping between CLK1-2 and TCLKx/FLCK provided by the backplane
  - Compatibility between AMC and switch on MCH ensured by e-keying

 © 2023 N.A.T. GmbH | UNCLASSIFIED | All trademarks, brands and logos are property of their respective owners

Con.	Region	Port	Function	non-redundant system (1x MCH)	redundant system (2x MCH)
Basic Side	Clock	TCKLA	Telecom Clock A	CLK1	1-CLK1
		TCLKB	Telecom Clock B	CLK2	1+2-CLK2
		FCLK	Fabric Clock or redundant TCLKA	CLK3	2-CLK1
	Common Options	0	1GbE	1-A	1-A
		1	redundant 1GbE		2-A
		2	SAS/SATA	1-B	1-B
		3	redundant SAS/SATA		2-B
	Fat Pipe	4	PCIe or 1GbE or 10GbE or SRIO	1-D	1-D
		5	PCIe or 1GbE or 10GbE or SRIO	1-E	1-E
		6	PCIe or 1GbE or 10GbE or SRIO	1-F	1-F
		7	PCIe or 1GbE or 10GbE or SRIO	1-G	1-G
	Extended Fat Pipe	8	1GbE or 10GbE or SRIO		2-D
Extended Side		9	1GbE or 10GbE or SRIO		2-E
		10	1GbE or 10GbE or SRIO		2-F
		11	1GbE or 10GbE or SRIO		2-G
	Extended Options	12	APS or Point-2-Point or rearIO		
		13	Point-2-Point or rearIO		
		14	Point-2-Point or rearIO		
		15	Point-2-Point or rearIO		
		TCLKC+D	Telecom Clocks C+D		
		17	1GbE or 10GbE or SRIO		
		18	1GbE or 10GbE or SRIO		
		19	1GbE or 10GbE or SRIO		
		20	1GbE or 10GbE or SRIO		

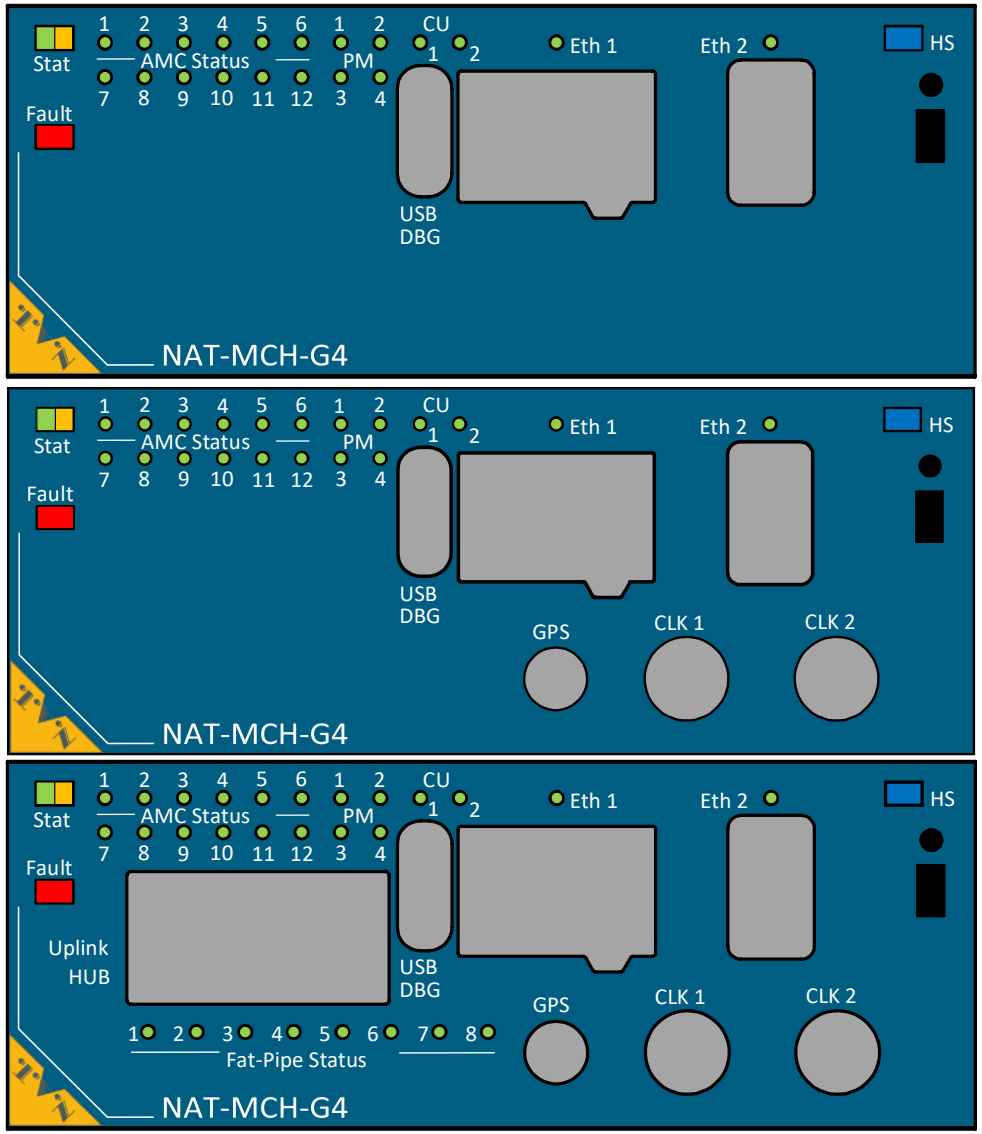
MTCA Carrier Hub (MCH) Adaptable to application demands



NAT-MCH-Gen 4

Standard MCH (SFS)

- MCH-G4
  - 3 different uplink version
    - RJ45/SFP/ix
    - IEEE1588 boundary switch
  - CLk Module
    - CLK 12F and option OXCO
  - HUB EX with Uplink 40/100G
  - HUB E with Uplink 40/100G
  - HUB PCIe with uplink x8/x16
- eMCH-G4 part of NATIVEServer



Main differences between Gen3 and Gen4 (excerpt)

Function	NAT-MCH G-3	NAT-MCH G-4 (improvements)
CPU + O/S + memory	Single core NXP Coldfire + OK1 + 64Mb	**Dual** ARM core (A9, Xilinx Zynq) + FreeRTOS + **1GB**
Base Switch	Broadcom 1GbE	Microchip 1/**10**/**40**GbE, **dual ARM core** (A53)
Base Fabric + Uplinks	12x 1GbE + 2x 1GbE (RJ45)	12x 1/**2.5**/**10**GbE + 2x 1/**10**GbE (RJ45/**iX**/**SFP-DD**)
Clock Module + ext. Input/output	CLK123, CLK12F, CLK-PHYS + dual input/output	CLK-G4 + dual input/output and **GPS**
IEEE1588/SyncE + TSN support + OXCO	Not supported+ Not supported + NAMC-PTM	**Supported** + **Supported** + **OCXO**
Fat pipe Ethernet switch	Marvell Amstrong-LP 40GbE	Marvell Amstrong-LP 40GbE
Fat Pipe + Uplinks	12x XAUI + MPO	12x XAUI/**10**/**40**G + **SFP-DD**
Fat pipe PCIe Switch + PCIe Gen	PLX + Gen3	MicroChip + **Gen4**
Fat Pipe + Uplinks	12x PCIe Gen3 + Finisar BOA (NAT-MCH-PHYS80)	12x PCIe **Gen4** + **SFP-DD**
Fat pipe SRIO Switch + SRIO Gen	IDT + Gen2	?
Fat Pipe + Uplinks	12x SRIO Gen2 + Infiniband	?
User Interfaces	CLI, Web (GoAhead) => Update with 2.22.x	**unified** CLI, **reworked** Web (Mongoose) **incl. CLI**
NATView: HPM update + backplane viewer + FRU-Ed	JRE on external device	**Integrated into Web interface** (excl. FRU-Editor)

Agenda

- About N.A.T.
- From ATCA to MTCA - two well connected standards
- Why do we need management?
- What is behind the management?
- How does it work?
- **What can you do?**

Management in xTCA What can you do?

- "Who" is in my system?
  - i.e. list of devices (aka "FRU")
- What capabilities does the FRU have?
  - i.e. active connections (AMCs) or RPMs (CUs)
- How healthy is my system?
  - i.e. sensors for current, voltage, temperature
  - i.e. events
- How can I interfere with my FRUs?
  - i.e. manipulation of sensors
- How can I service my system?
  - i.e. hot-swap FRUs

Summary

- About N.A.T.
- From ATCA to MTCA - two well connected standards
- Why do we need management?
- What is behind the management?
- How does it work?
- What can you do?

Thank you for your attention !

Herbert Erd
Strategic Business Development Manager

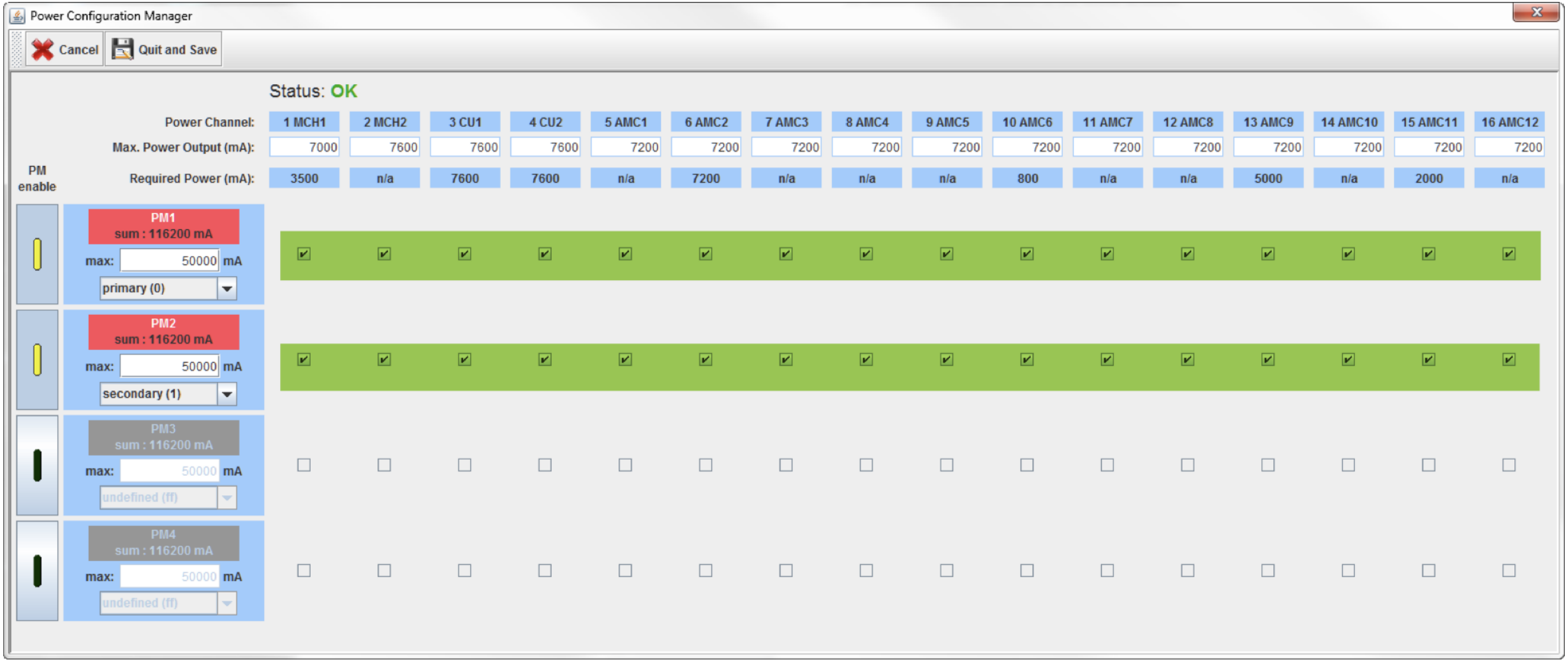
herbert.erd@nateurope.com

N.A.T. GmbH
Konrad-Zuse-Platz 9
53227 Bonn, Germany

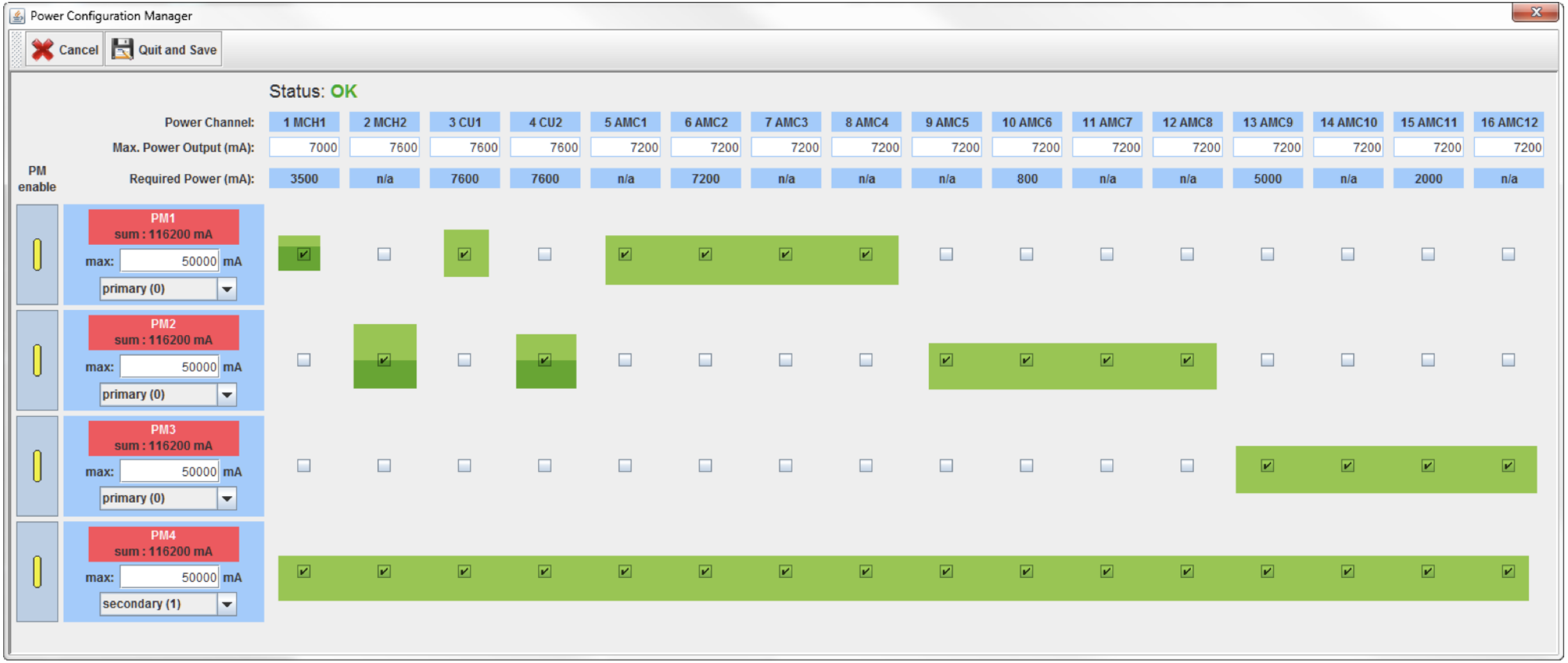
www.nateurope.com

Repository

NATview Power Configuration Manager Redundancy



NATview Power Configuration Manager n+1 Redundancy



Important CLI commands of NAT-MCH example and excerpt

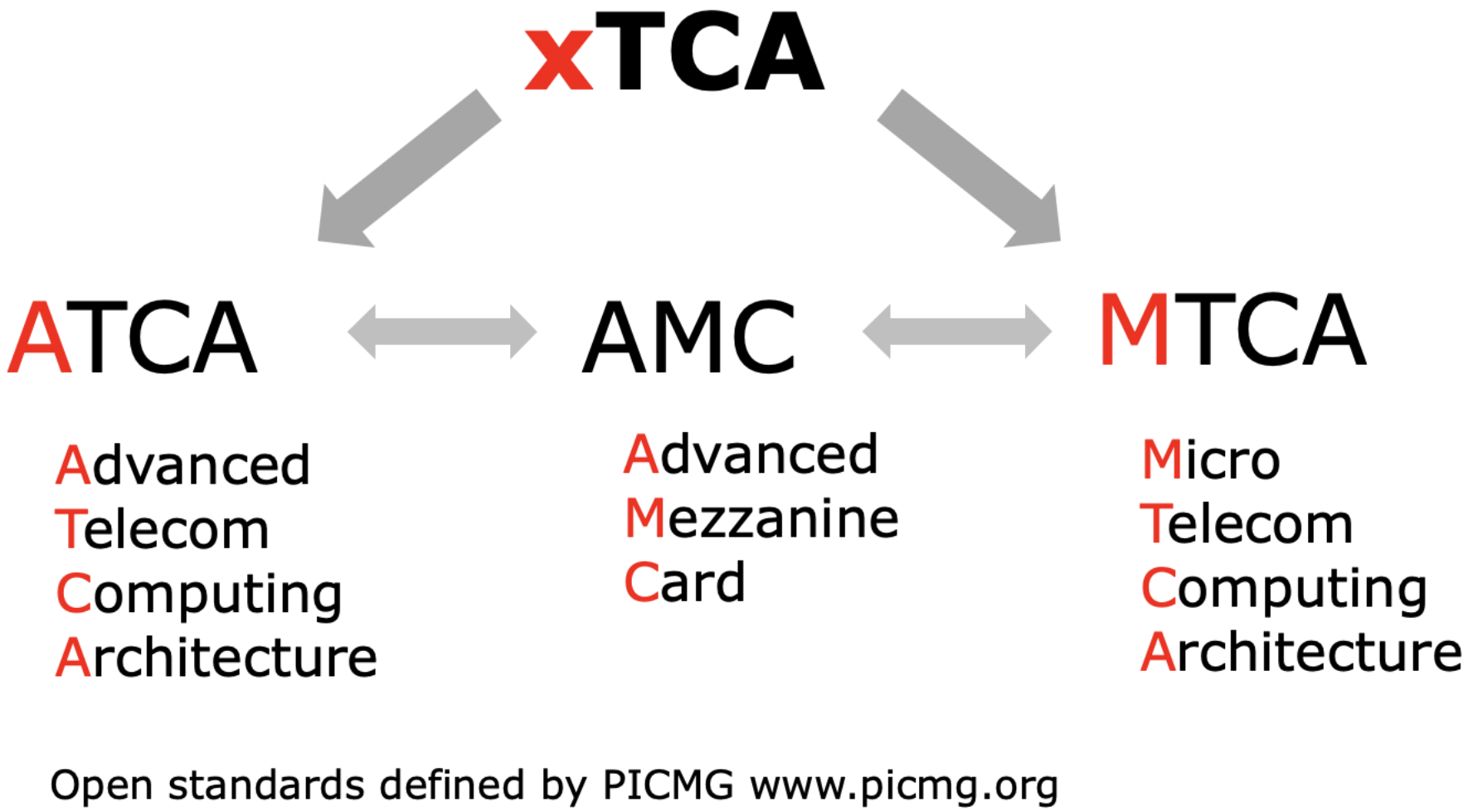
- **imsg_info** – lists all **IPMI messages** supported by MCH
  - Prints the implementation status for the supported IPMI messages on the Host (RMCP) interface and can be used to print a list of IPMI messages that are supported by the MCH.
- **show_ekey -** shows **results of e-keying process**, i.e. all activated connections
- **show_fru -** lists **all FRUs** in system
- **show_fruinfo –** lists **contents** for given **FRU**
- **show_cu -** shows status of all **cooling units**
- **show_pm -** shows status of all **power modules**
- **show_sensorinfo** fru_id – shows details for **all sensors** of given **FRU**
- **fan_ctl -** allows **manual control** of **cooling units**
- **shutdown** fru_id/all – initiates **gracefull shutdown** of single or all **FRU**s
- **fru_start** fru_id – initiates **gracefull start** of given **FRU**

Inventory: *show_fru*

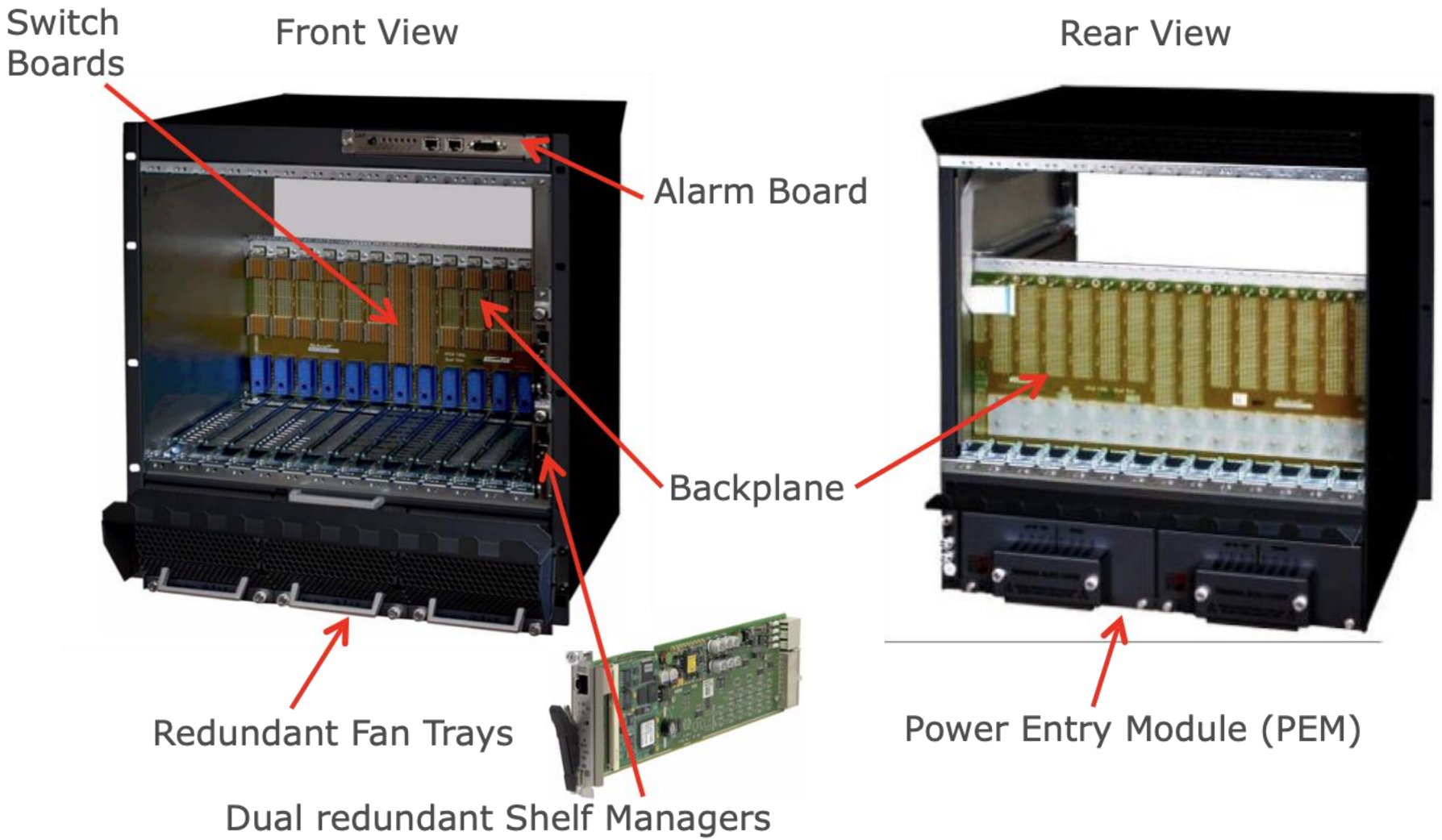
- Infrastructure
  - 0, 3, 60, 61 MCH1 modules
  - 1, 4, 62, 63 MCH2 modules
  - 40-41 Cooling Unit 12
  - 50-53 Power Modules 1, 2, 3, 4
- Payloads
  - 5-16: AMC1-12
  - 64, 65: MCH-RTM
  - 90-101: µRTM1-12
  - 102-104: eRTM13, 14, 15

```
FRU  Device    State  Name
=========================================
 0   MCH         M4    NMCH-CM
 3   mcmc1       M4    NAT-MCH-MCMC
 5   AMC1        M4    CCT AM 902/411
 6   AMC2        M4    X2TIMER
 7   AMC3        M4    DAMC2V3
 8   AMC4        M4    DAMC-TCK7
11   AMC7        M4    SIS8300L2 AMC
12   AMC8        M4    SIS8300L2 AMC
13   AMC9        M4    SIS8300L2 AMC
14   AMC10       M4    SIS8300L2 AMC
15   AMC11       M4    SIS8300L2 AMC
16   AMC12       M4    SIS8300L2 AMC
40   CU1         M4    Schroff uTCA CU
41   CU2         M4    Schroff uTCA CU
51   PM2         M4    PM-AC1000
60   Clock1      M4    MCH-Clock
61   HubMod1     M4    MCH-PCIe
64   MCH1-RTM    M4    MCH-RTM-ComEx
91   AMC2-RTM    M4    X2TIMERRTM
92   AMC3-RTM    M4    DAMC2RTM
93   AMC4-RTM    M4    DAMC-TCK7 RTM
96   AMC7-RTM    M1    SIS8300L2 RTM
97   AMC8-RTM    M1    SIS8300L2 RTM
98   AMC9-RTM    M1    SIS8300L2 RTM
99   AMC10-RTM   M1    SIS8300L2 RTM
100  AMC11-RTM   M1    SIS8300L2 RTM
101  AMC12-RTM   M1    SIS8300L2 RTM
104  eRTM15      M4    DRTM-LOG1300
```

Common Abbreviations ATCA AMC MicroTCA MTCA uTCA µTCA xTCA

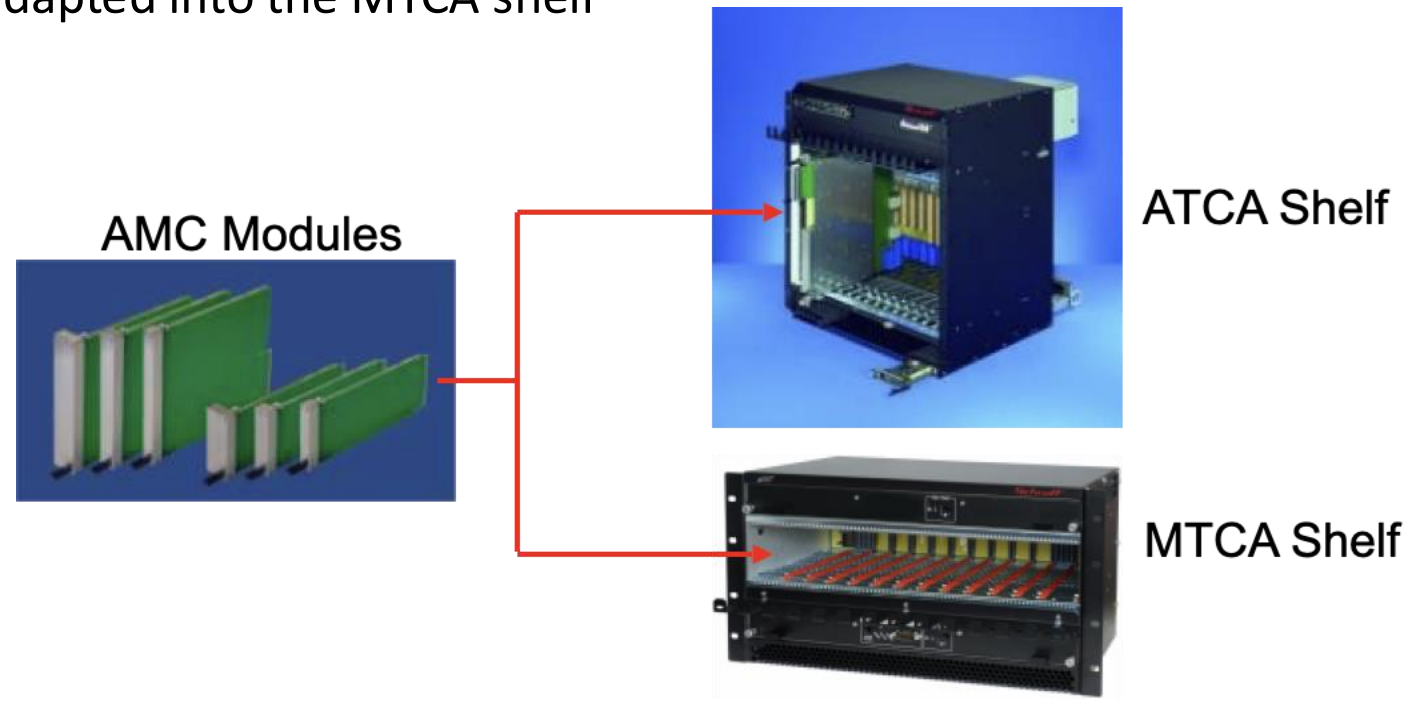


Quick Look At ATCA ATCA Shelf Elements

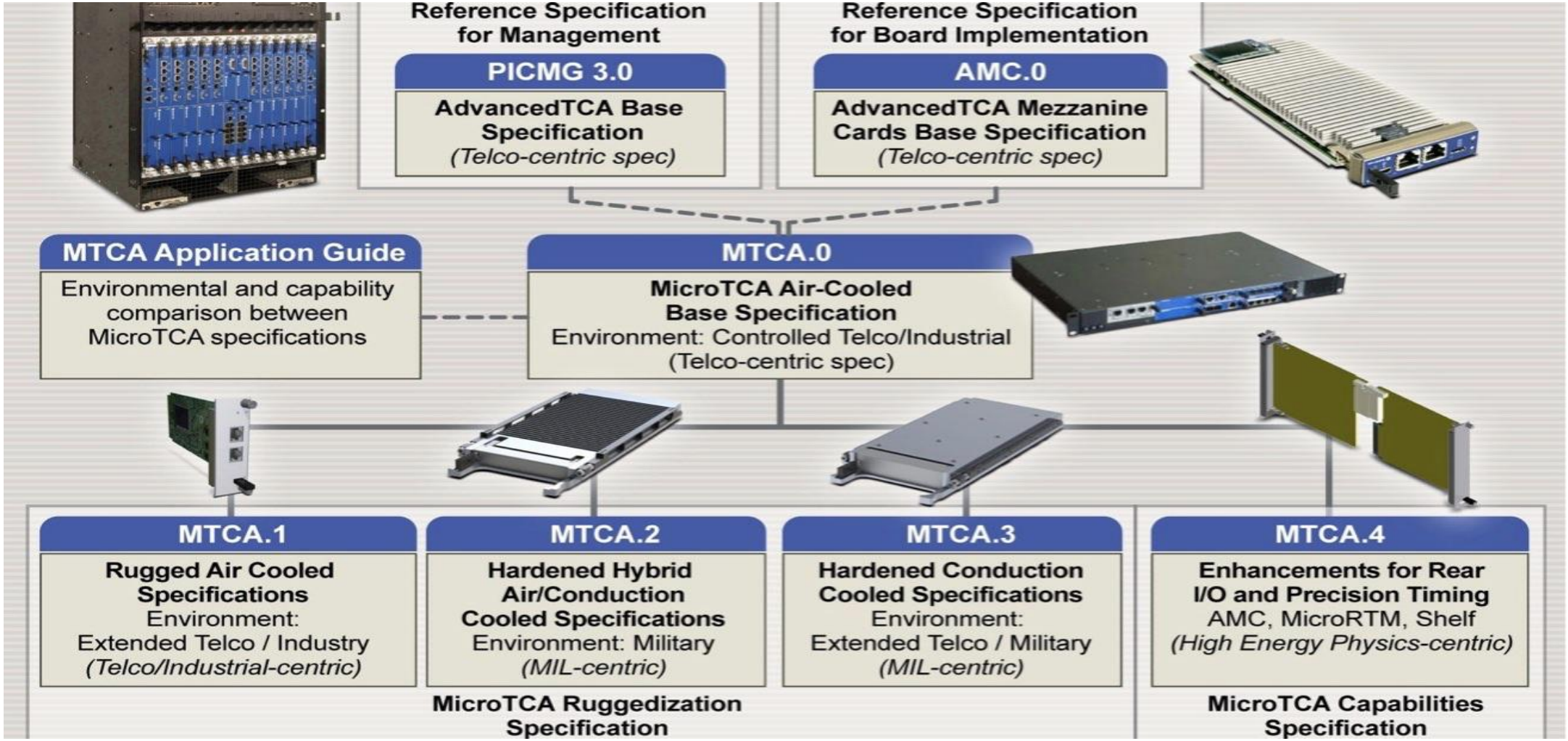


Quick Look At ATCA How to migrate from ATCA to MTCA

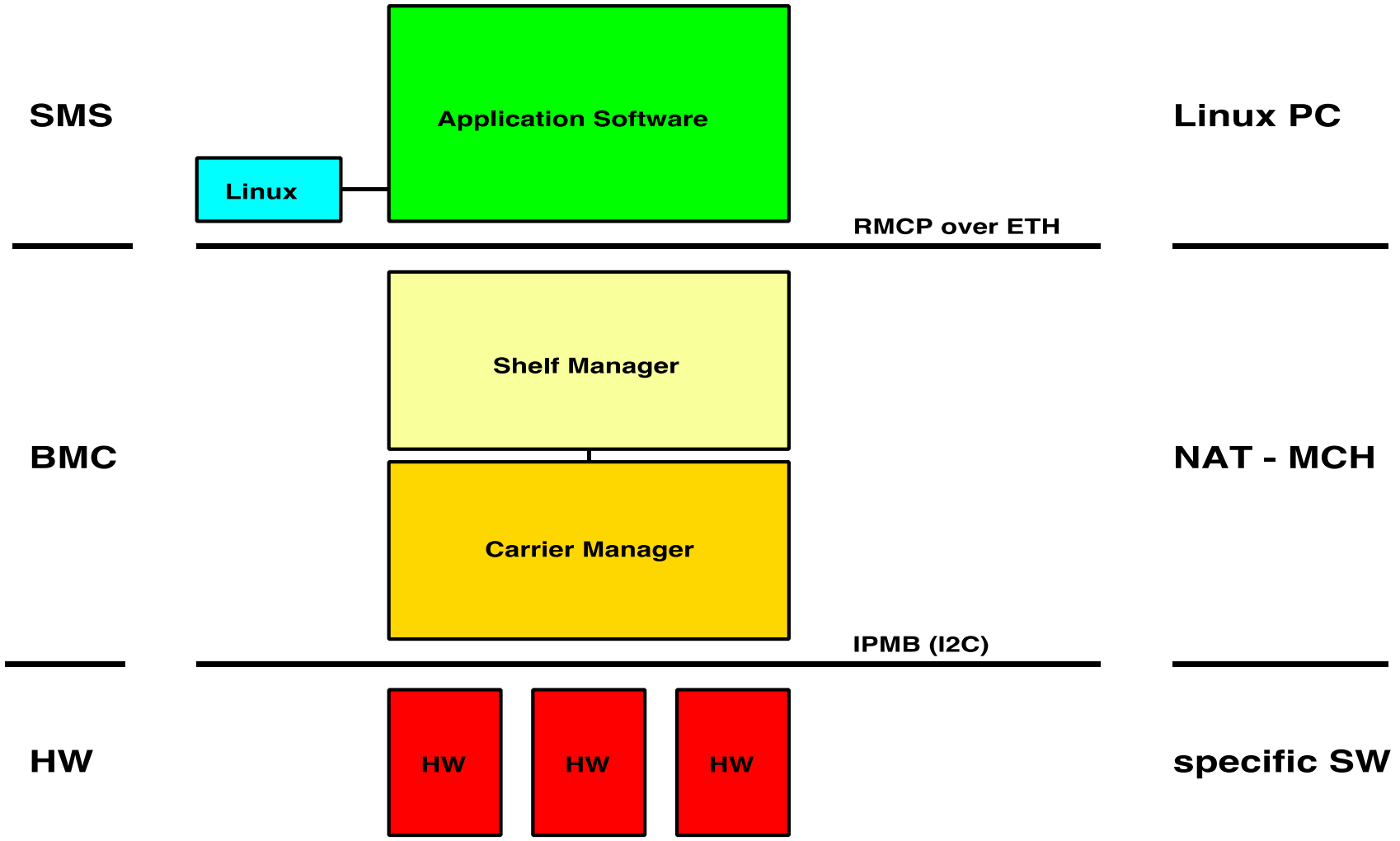
- The basic idea of MTCA is to have a shelf that contains just AMC modules
- Backplane directly accepts AMC modules
- AMCs are interchangeable between ATCA and MTCA
- The infrastructure of a ATCA Carrier was adapted into the MTCA shelf (power management switching)
- power input and all outputs to the front



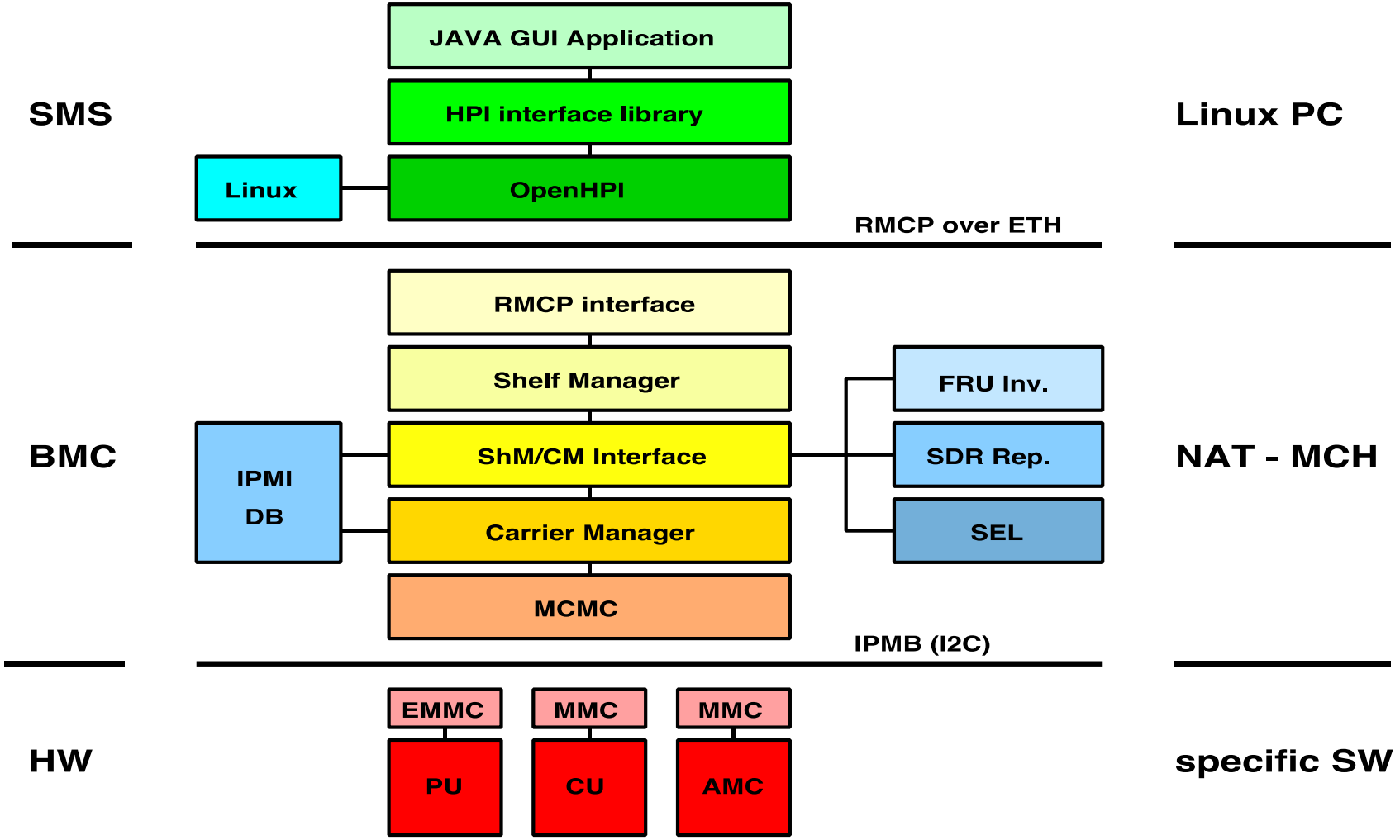
ATCA and MTCA Specifications



Management in xTCA Software structure



Management in xTCA Software structure



Useful numbers to remember FRU and I²C addresses

Device	Site No.	FRU ID	I²C Addr.
MCH-CM (Carrier Mngt.)	*1*	*0*	*0x20*
MCMC 1	*1*	*3*	*0x10*
MCMC 2	*2*	*4*	*0x12*
AMC 1-12 AMC 13	*1-12 13*	*5-16 29*	*0x72-0x88 0xa2*
CU 1 CU 2	*1 2*	*40 41*	*0xA8 0xAA*
PM 1 PM 2 PM 3 PM 4	*1 2 3 4*	*50 51 51 53*	*0xC2 0xC4 0xC6 0xC8*
MCH-CLK 1	*1*	*60*	*0x14*
MCH-CLK 2	*2*	*62*	*0x18*
MCH-Hub 1 (PCIe SRIO XAUI)	*1*	*61*	*0x16*
MCH-Hub 2 (PCIe SRIO XAUI)	*2*	*63*	*0x1a*
MCH-RTM 1	*1*	*64*	*0x1c*
MCH-RTM 2	*2*	*65*	*0x1e*
OEM 1-19	*1-19*	*60-78*	*0x42-0x66*
µRTM 1-12		*90-101*	*0x72 0x74-0x88*
Carrier FRU (backplane FRU)	*1*	*253*	*0xA4*

Useful numbers to remember FRU and I²C addresses

Device	Site No.	FRU ID	I²C Addr.
AMC13 in MCH 1 slot	*1*	*29*	*0xa2*
AMC13 in MCH 2 slot	*2*	*30*	*0xa4*
Telco alarm	*1*	*79*	*-*
Carrier Manager		*0*	*0x20*
physical Shelf FRU Info 1		*1*	
physical Shelf FRU Info 2		*2*	
Reserved for further AMCs		*17-28*	
Reserved for further RTMs		*102-124*	
Reserved		*125-127*	
Local Shelf Manager		*128*	
logical ShM (backplane FRU-Info)		*254*	
Implementation defined		*80-89*	
reserved for OEM modules		*66-78*	
reserved for PM		*54-59*	