

MicroTCA Platform Management with Free and Open Source Software

*Gokhan Sozmen
coreIPM Project Leader*

Tutorial Outline

- Building management into your product
- Introduction to IPMI & ATCA/MicroTCA extensions
- MicroTCA management issues
- Putting together a system from scratch
- Interfacing to upper level tools & interfaces

Building management into your product ... basic requirements.

- You need to locally and remotely monitor system hardware and sensors, control system components, and log important system events.
- The management scheme needs to be independent of a main motherboard processor (the payload or host) and its peripherals, BIOS, OS, payload power and system management software that runs on the payload processor.
- The management scheme should provide out-of-band access to management functions at all times.

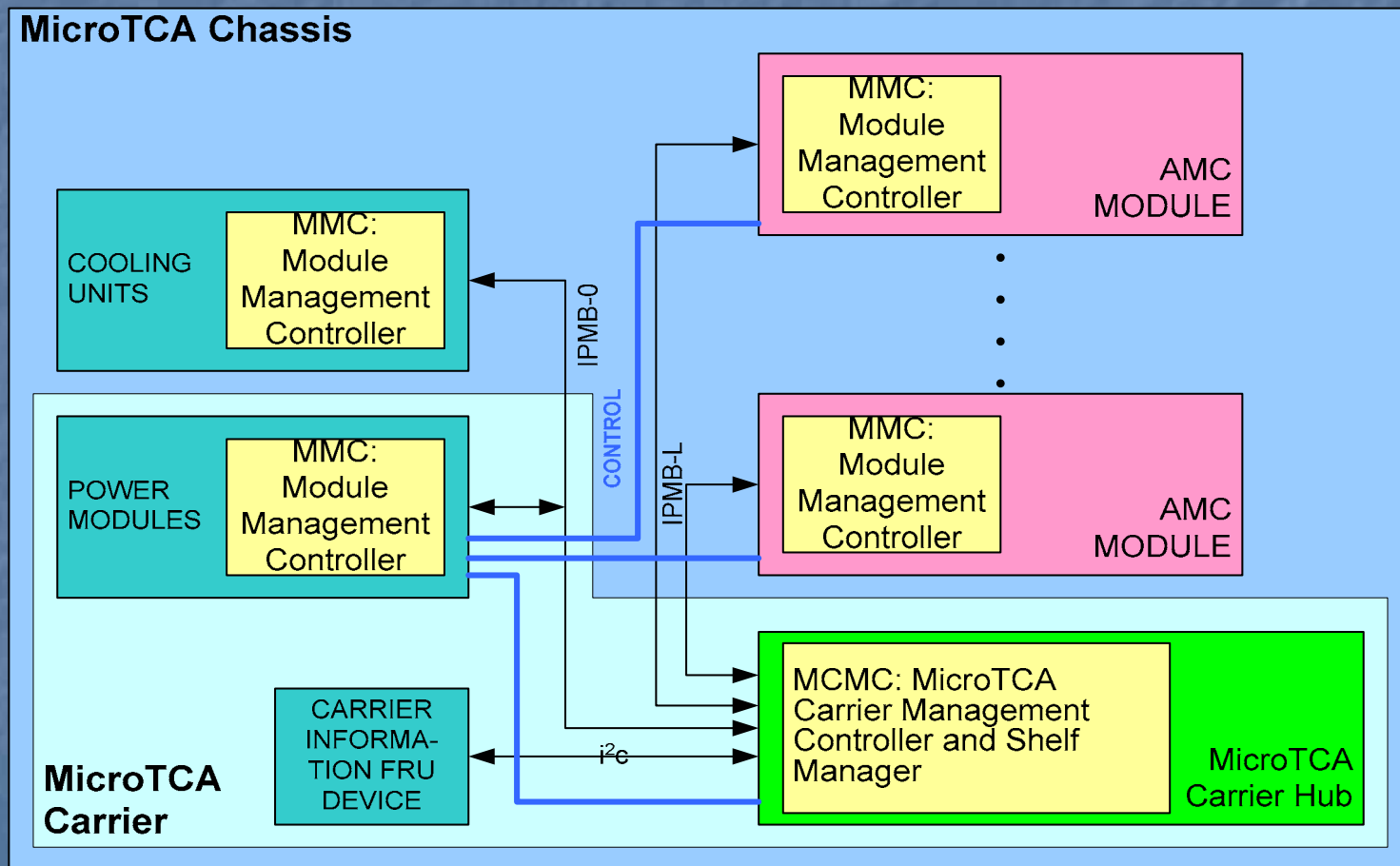
What is the Intelligent Platform Management Interface (IPMI) ?

- An open standard platform management specification.
- IPMI defines how users can monitor system hardware and sensors, control system components, and log important system events.
- It is independent of a main motherboard processor (the payload or host) and its peripherals, BIOS, OS, payload power and system management software that runs on the payload processor.
- It provides out-of-band access to management functions at all times.

ATCA and uTCA extensions to IPMI

- Single board model extended to Blade model
- Chassis, Backplane, & Shelf Management subsystem
- Switching infrastructure
- Hot swapping
- Scaleable & customizable platform

MicroTCA Management Infrastructure



Putting together a uTCA system using Free & Open Source components

Key components:

- coreIPM
 - Board & Chassis Management Software
 - HW Reference design
 - Development system
- NetSNMP
- OpenHPI
- OpenPegasus

What is coreIPM ?

coreIPM is an IPMI based comprehensive open source management architecture for platform management.

It can manage:

- a single motherboard
- a blade
- an enclosure
- configurations that can be made up of any combination of boards, blades, enclosure controllers, and multiple enclosure aggregations.

coreIPM is compliant with the Intelligent Platform Interface Management (IPMI) specification v2.0 + PICMG ® 3.0 R2.0 AdvancedTCA, PICMG® AMC.0 R2.0 Advanced Mezzanine Card and PICMG ® MTCA.0 MicroTCA R1.0 specification extensions for blades and micro blades.

coreIPM is the final piece of the puzzle for F/OSS based platform management.

coreIPM features

- Spans the whole IPMI universe, not just tied to ATCA:
 - Bigger pool of potential users and developers
 - Easy to retarget/repackage your solution: go from an ATCA blade to a MicroTCA or an appliance solution with the same management architecture. Expertise portable across platforms.
- Not tied to a management chip vendor.
- Based on a popular 32 bit architecture. Migrate paths from ARM7 to ARM9 for higher performance or to ARM Cortex architecture for low power & cost.
- 32 bit performance at an 8 bit cost. Processing power and large on chip storage options provide headroom for growth. Powerful enough to run a network stack, encryption, KVM over IP etc.
- Shelf management functions are separated between an coreBMC ARM core processor and an embedded system that handles the network centric functions such as RMCP+, SNMP, Web I/F. Hardware independent code enables porting and usage of embedded platform and OS of choice for the shelf manager.
- Hardware independent IPMI functionality is easily portable.
- Architectural support for port expanders provide an unlimited number of peripheral ports; no limitation on i2c, GPIO, ADC etc. interfaces.

coreIPM features..cont.

In fact...

coreIPM provides a plug-in solution not only for system management but for any embedded application. You can modify the code freely and with a rich set of peripheral drivers and an operating environment, you have a platform to build and ship your applications quickly. There are no licensing encumbrances or royalties.

coreIPM architecture ..components.

coreIPM architecture is comprised of:

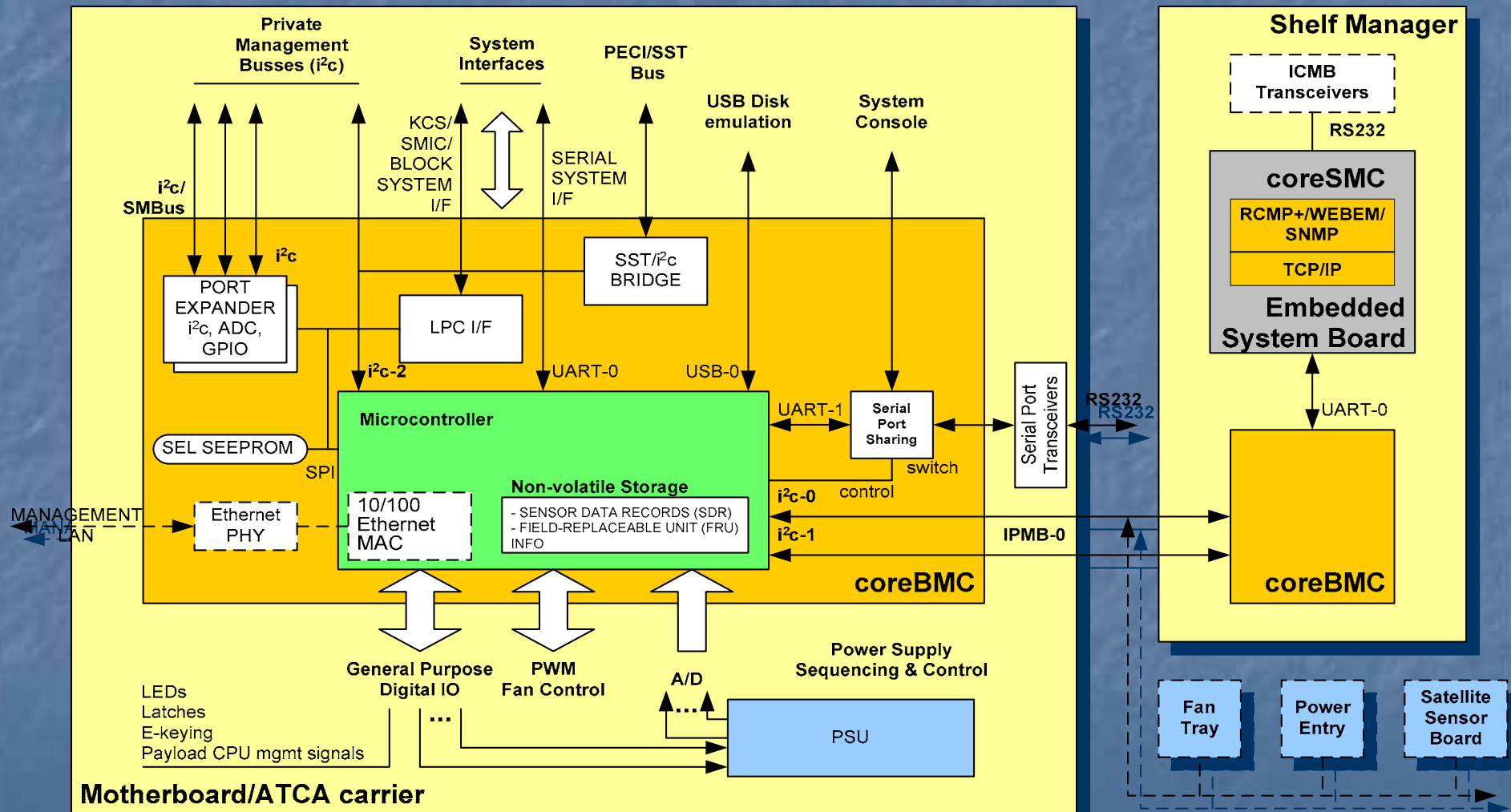
- *coreBMC* (Baseboard Management Controller) and
 - *coreSMC* (Shelf Management Controller)
- hardware & software modules.

coreBMC is fully sufficient by itself to provide board management functions for ATCA, AMC, microTCA (Carrier manager, Power module, Fan), OPMA, motherboard, appliance applications. Coupled With *coreSMC* it's possible to build shelves and higher hardware hierarchies such as an ATCA chassis.

coreIPM architecture ..components.

- *coreBMC* incorporates all shelf management functions except the higher level protocols. It handles all power, cooling, e-keying and state management. *coreBMC* has IPMI, ATCA, AMC, MicroTCA support built in the code base. Specific features are enabled at build time. Supports Terminal mode & KCS system interfaces.
- *coreBMC* also includes
 - BSD licensed lwIP TCP/IP stack.
 - RMCP+, SNMP Traps, Serial over LAN & crypto support on an ARM7 microcontroller.
 - For low cost shelf manager, motherboard BMC/OPMA applications
- *coreSMC* incorporates networking stack + RMCP + SNMP + HPI + WEBEM + access control + high availability functions.

coreIPM architecture ..Hardware



coreBMC architecture..Processor Support

- Small footprint designed to be embedded in a single chip microcontroller.
- Initial support for ARM7 cores. Use the controller best suited for the task. High performance and low power consumption in a cost-effective package. Cost competitive with 8 and 16 bit microcontrollers.

coreBMC Current Command Support

IPMI Device "Global" Commands

- Get Device ID
- Cold Reset
- Warm Reset

BMC Device and Messaging Commands

- Set BMC Global Enables
- Get BMC Global Enables
- Clear Message Flags
- Get Message Flags
- Get Message
- Send Message
- Master Write-Read
- Get Channel Authentication Capabilities
- Get Session Challenge
- Activate Session
- Set Session Privilege Level
- Close Session
- Get Session Info
- Get AuthCode
- Set Channel Access
- Get Channel Access
- Get Channel Info
- Set User Access
- Get User Access
- Set User Name
- Get User Name
- Set User Password

BMC Watchdog Timer Commands

- Reset Watchdog Timer
- Set Watchdog Timer
- Get Watchdog Timer

Chassis Device Commands

- Get Chassis Capabilities

Event Commands

- Set Event Receiver
- Get Event Receiver
- Platform Event (a.k.a. "Event Message")

PEF and Alerting Mandatory Commands

- Get PEF Capabilities
- Arm PEF Postpone Timer
- Set PEF Configuration Parameters
- Get PEF Configuration Parameters
- Set Last Processed Event ID
- Get Last Processed Event ID

Sensor Device Commands

- Get Device SDR Info
- Get Device SDR
- Reserve Device SDR Repository
- Get Sensor Reading

FRU Device Commands

- Get FRU Inventory Area Info
- Read FRU Data
- Write FRU Data

coreBMC Current Command Support

AdvancedTCA® and PICMG® specific request commands

- Get PICMG Properties
- Get Address Info
- FRU Control
- Get FRU LED Properties
- Get LED Color Capabilities
- Set FRU LED State
- Get FRU LED State
- Set IPMB State
- Set FRU Activation Policy
- FRU Activation Policy
- Set FRU Activation
- Get Device Locator Record ID
- Set Port State
- Get Port State
- Compute Power Properties
- Set Power Level
- Get Power Level
- Renegotiate Power
- Get Fan Speed Properties
- Set Fan Level
- Get Fan Level

SDR Device Commands

- Get SDR Repository Info
- Reserve SDR Repository
- Get SDR

LAN Device Commands

- Set LAN Configuration Parameters Transport
- Get LAN Configuration Parameters Transport
- Suspend BMC ARPs

core/PM Security

Multi-session access

User Privilege Levels

Cipher Suites

Authentication Algorithms

- HMAC-SHA1
- HMAC-MD5

Integrity Algorithms

- HMAC-SHA1-96
- HMAC-MD5-128
- MD5-128

Confidentiality Algorithms

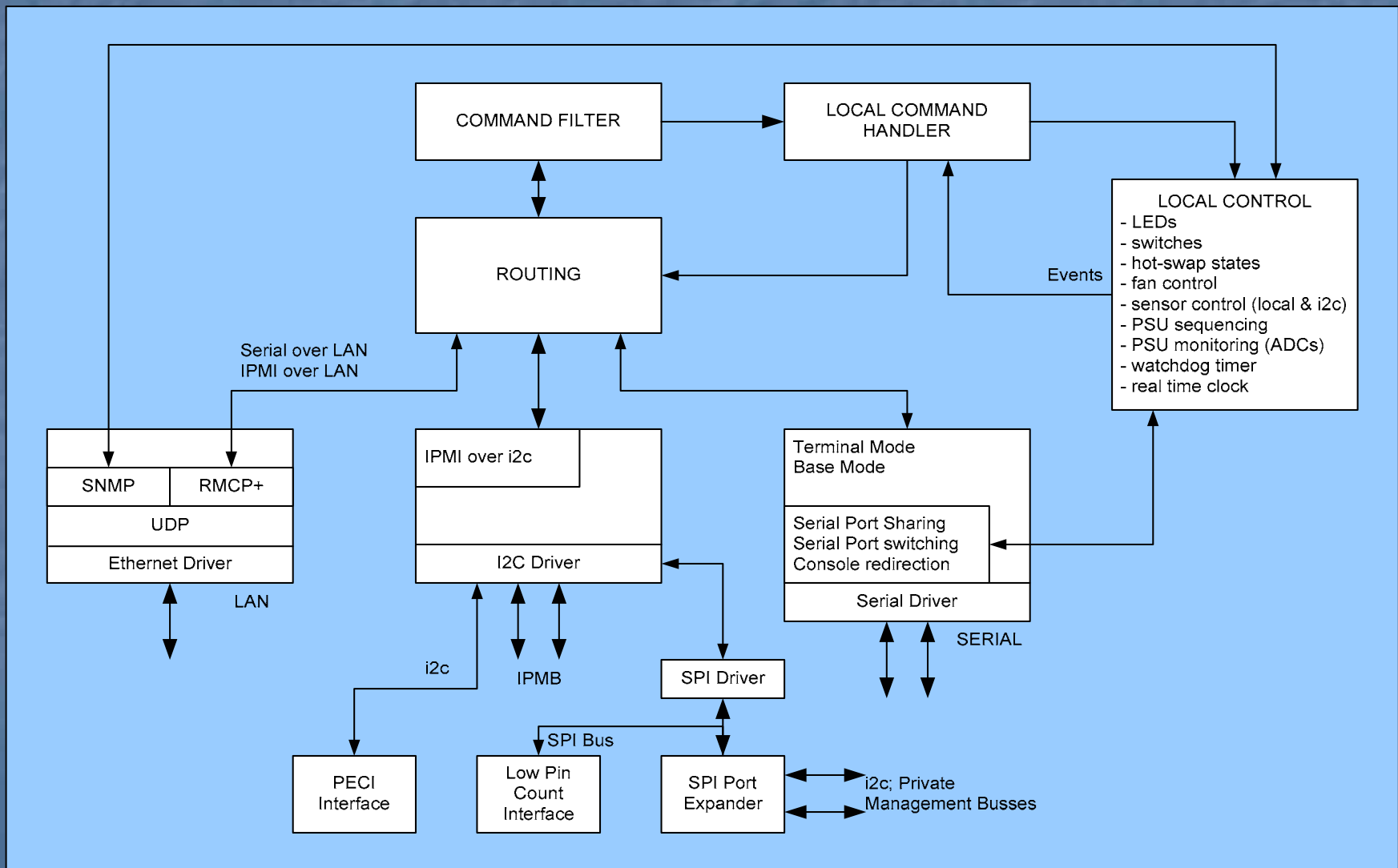
- AES-CBC-128

coreIPM architecture ..HW features.

Supported hardware features

- i²c
- Timers
- General purpose IO pins for latches, LEDs, E-keying and other uses
- RS232 with port switching
- SPI : port expanders, SEEPROM, LPC
- ADC
- PWM/tach fan control
- Watchdog Timer and Real-Time Clock with battery backup
- Edge or level sensitive interrupt pins
- Ethernet (on select microcontrollers) supporting RMCP+, Serial over LAN
- USB (on select microcontrollers)
- Field firmware upgrades over serial, LAN or i2c interfaces.

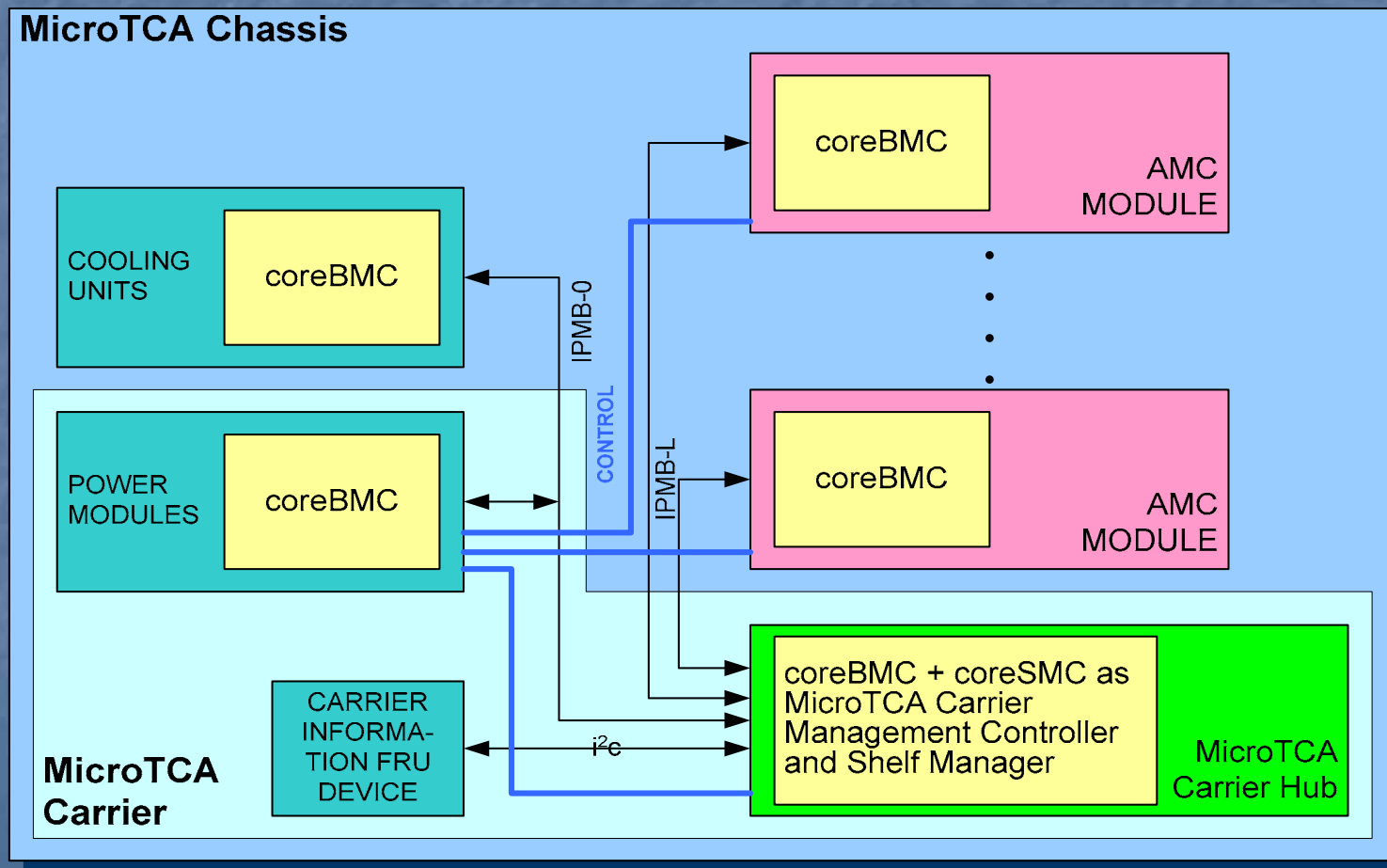
coreBMC architecture .. SW modules



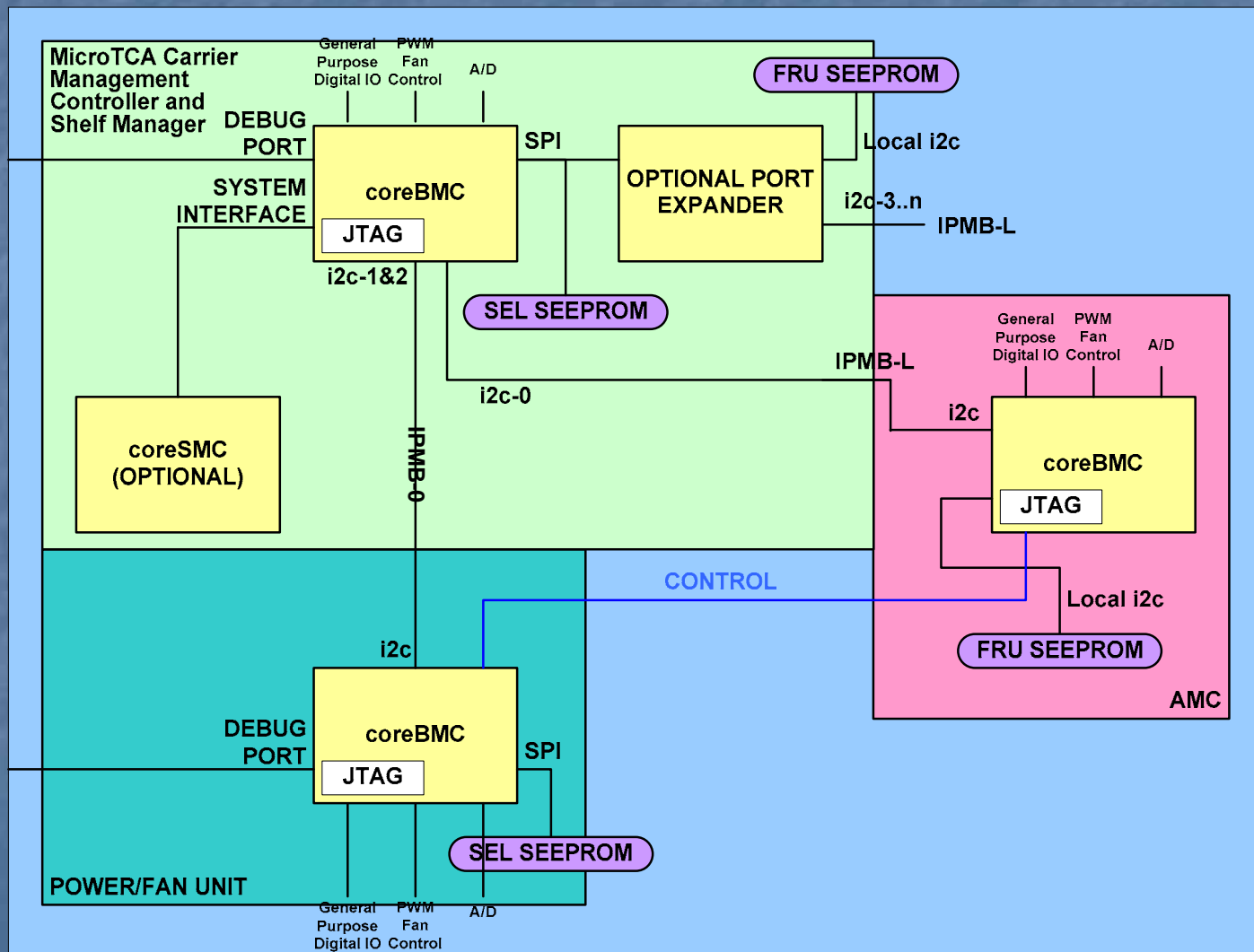
coreIPM licensing

- Source code is available under the Open Source GNU General Public License (GPLv2) and commercial licensing.
- coreIPM project has full ownership of the source and can provide alternative licensing.
- Commercial licensees get a commercially supported product without a requirement that their coreIPM-based software be open sourced.

Blade application of coreIPM... MicroTCA Management Infrastructure



coreIPM logical reference design configured for MicroTCA Management

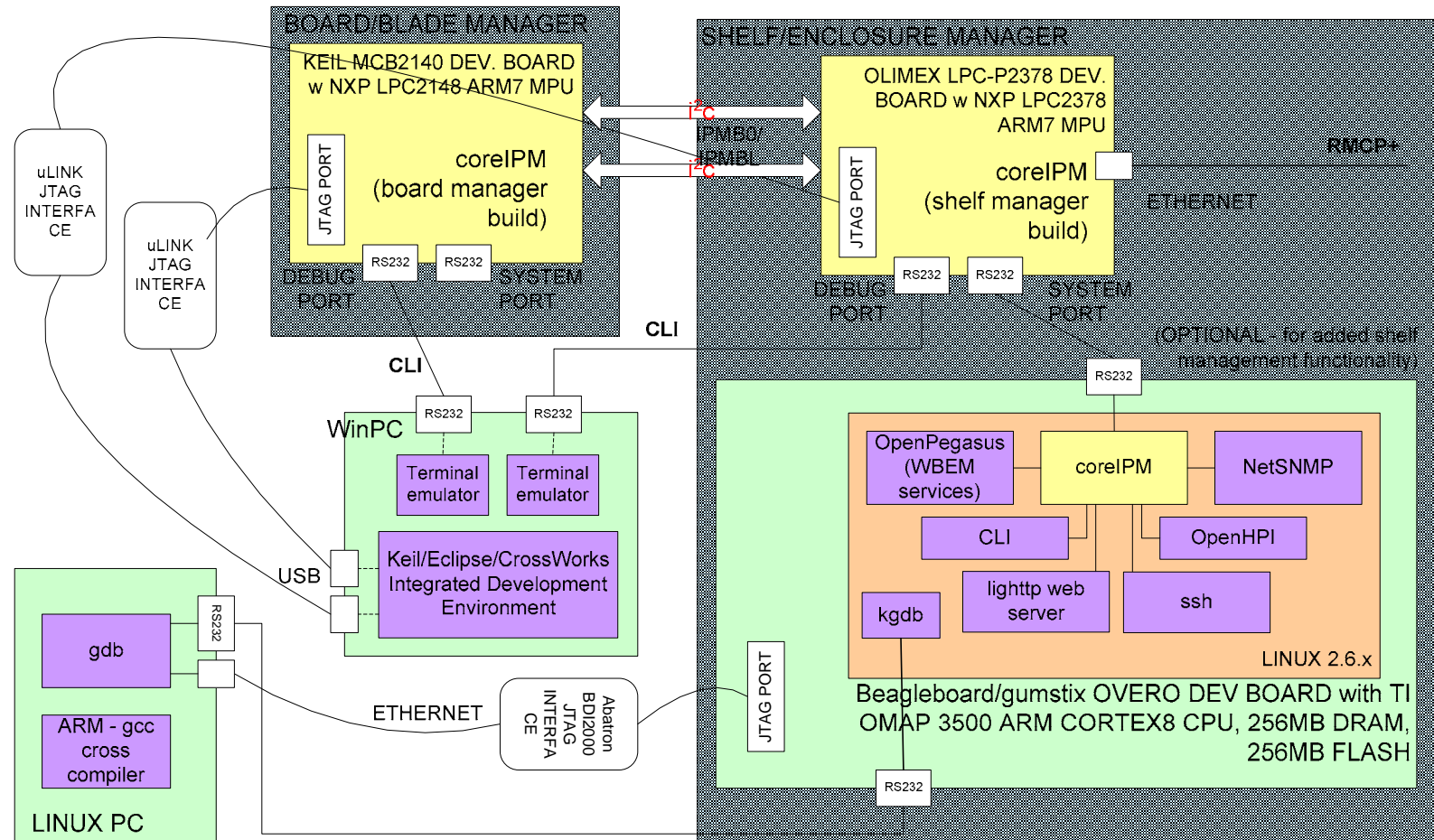


Building a platform management subsystem

www.coreipm.com has the sources, schematics, build tools and latest instructions.

- Assemble your development environment. Schematics for all HW components and development boards are available for free.
- Download source and tools .. again for free.
- Configure. (All done for the reference design.)
- Build.
- Download to hardware.
- Test.

coreIPM reference dev. environment



REFERENCE DEVELOPMENT ENVIRONMENT

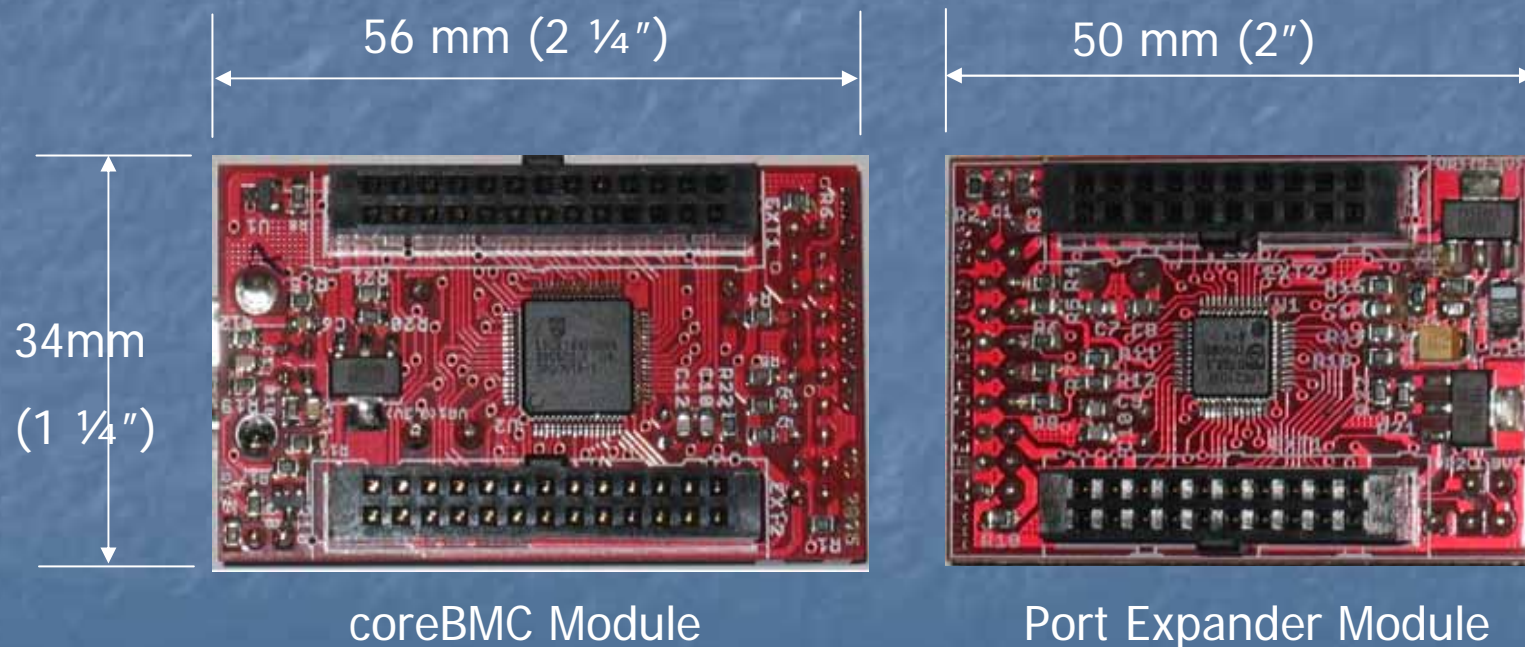
Configuring coreBMC

There is a common code base. Certain configurations are pre-defined with make/project files, use the one for your build:

- IPMC: ATCA blade
- MMC : AMC blade
- MCMC: MicroTCA Carrier Management Controller
- PU: MicroTCA Power Unit Enhanced MMC
- BMCXL: BMC, OPMA controller
- PE: SPI to i2c/ADC/GPIO
- INFRASTRUCTURE: no IPMI or ATCA, for embedded control applications.

coreIPM quick start implementation with plug-in building blocks

- System builders can build all configurations with off-the-shelf building blocks



coreIPM quick start implementation with plug-in building blocks .. cont.

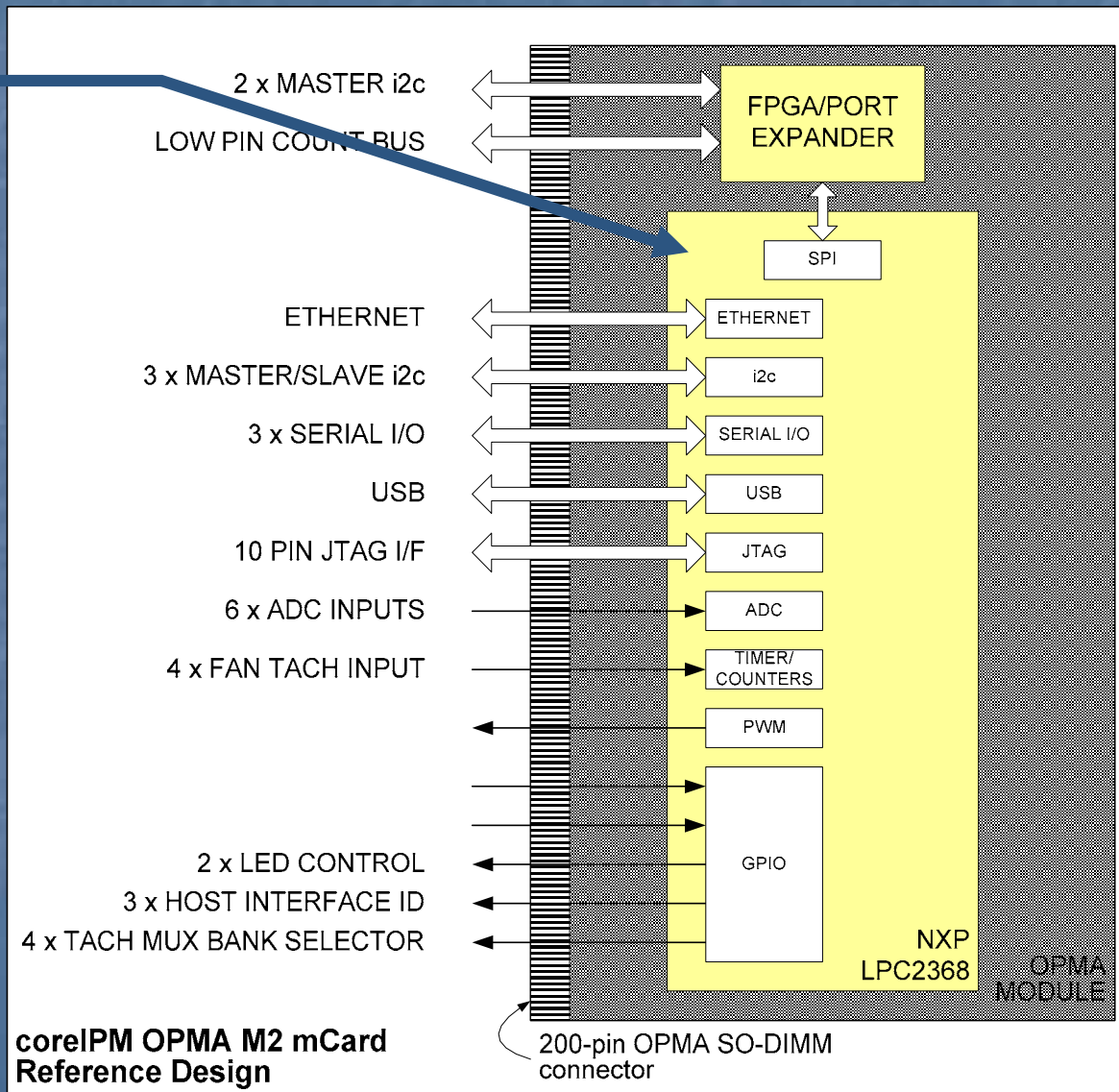


Shelf Management processor running coreIPM-LINUX release *Gumstix Overo Board*

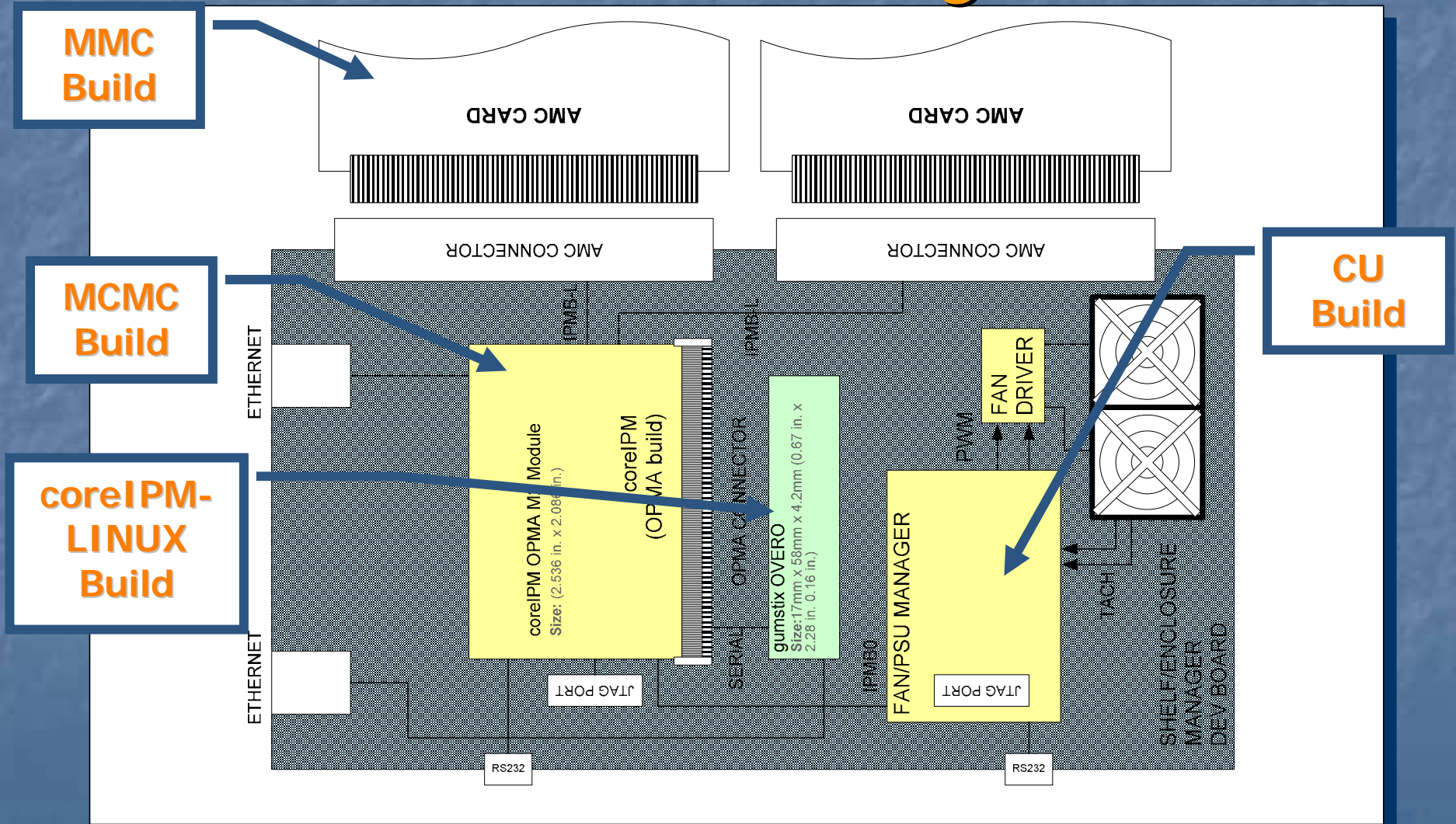
- Processor: OMAP 3503 with ARM Cortex-A8 CPU
- Clock(MHz):600 MHz
- Performance:1200 Dhrystone MIPS
- Memory:256MB RAM 256MB Flash
- Size:17mm x 58mm x 4.2mm (0.67 in. x 2.28 in. 0.16 in.)

coreIPM OPMA Reference Design

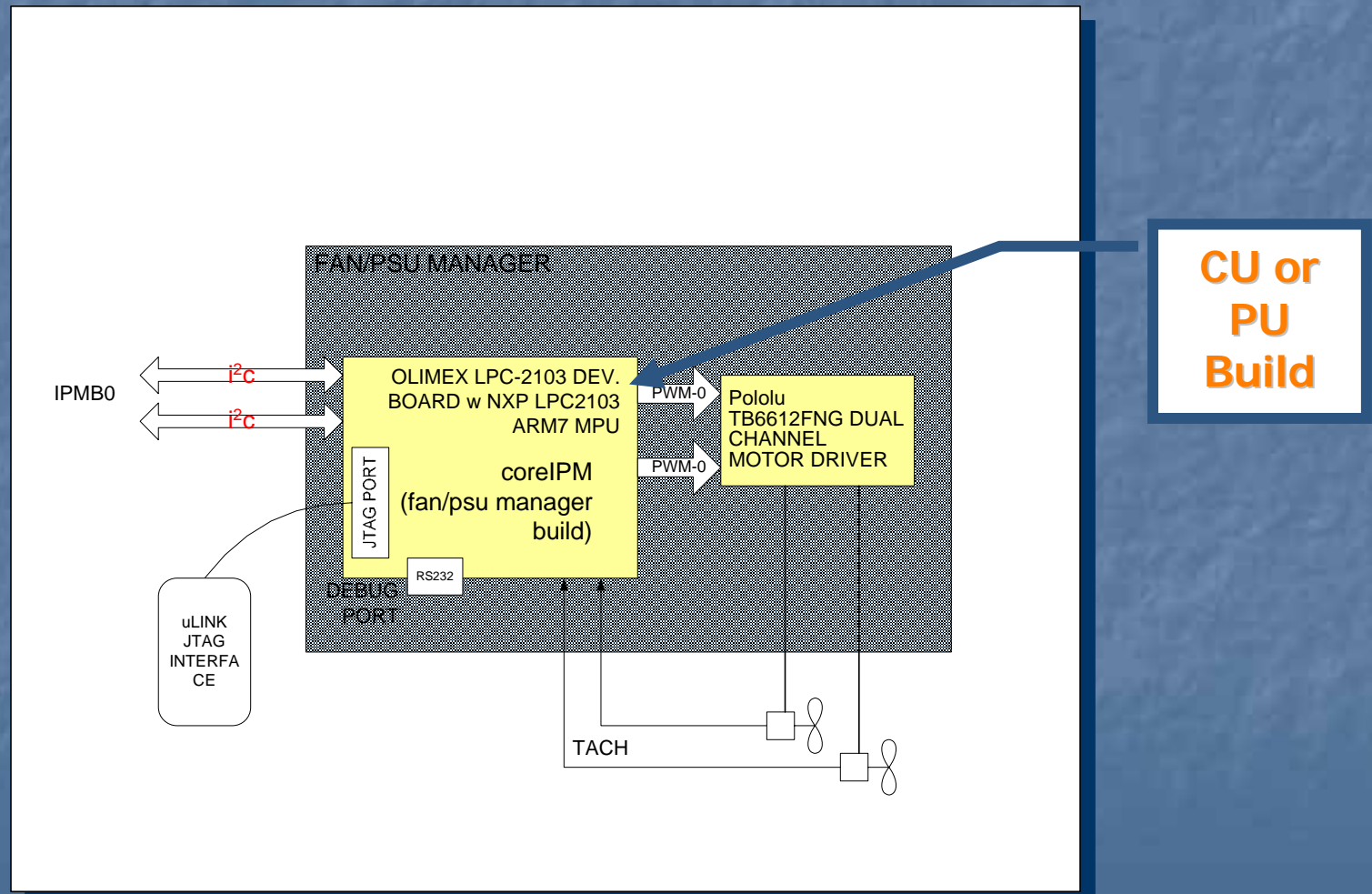
**MCMC
Or
OPMA
Build**



coreIPM dev board for MicroTCA & Generic Shelf Management



coreIPM Fan/PSU Manager reference dev. env.



Configuring coreBMC .. Sensors

Sensors currently supported:

- On chip ADC (1-n)
- On chip GPIO
 - Hot swap switches
 - AMC signals
- i²c based temperature sensors (LM75 series)

Configuring coreBMC .. Cont.

You might need to customize :

- LEDs
- Sensor access functions
- Fans
- Platform Event Filtering
- Serial ports, etc
- FRU entries
- HW specific GPIO
- Pin mapping to match your HW design

We provide a configured system for our development environment.

Building coreBMC

- For coreBMC, full set of development tools for ARM are available using either gcc or commercial compilers and debuggers from multiple vendors. JTAG interface enables FLASH downloads and on-chip debugging.

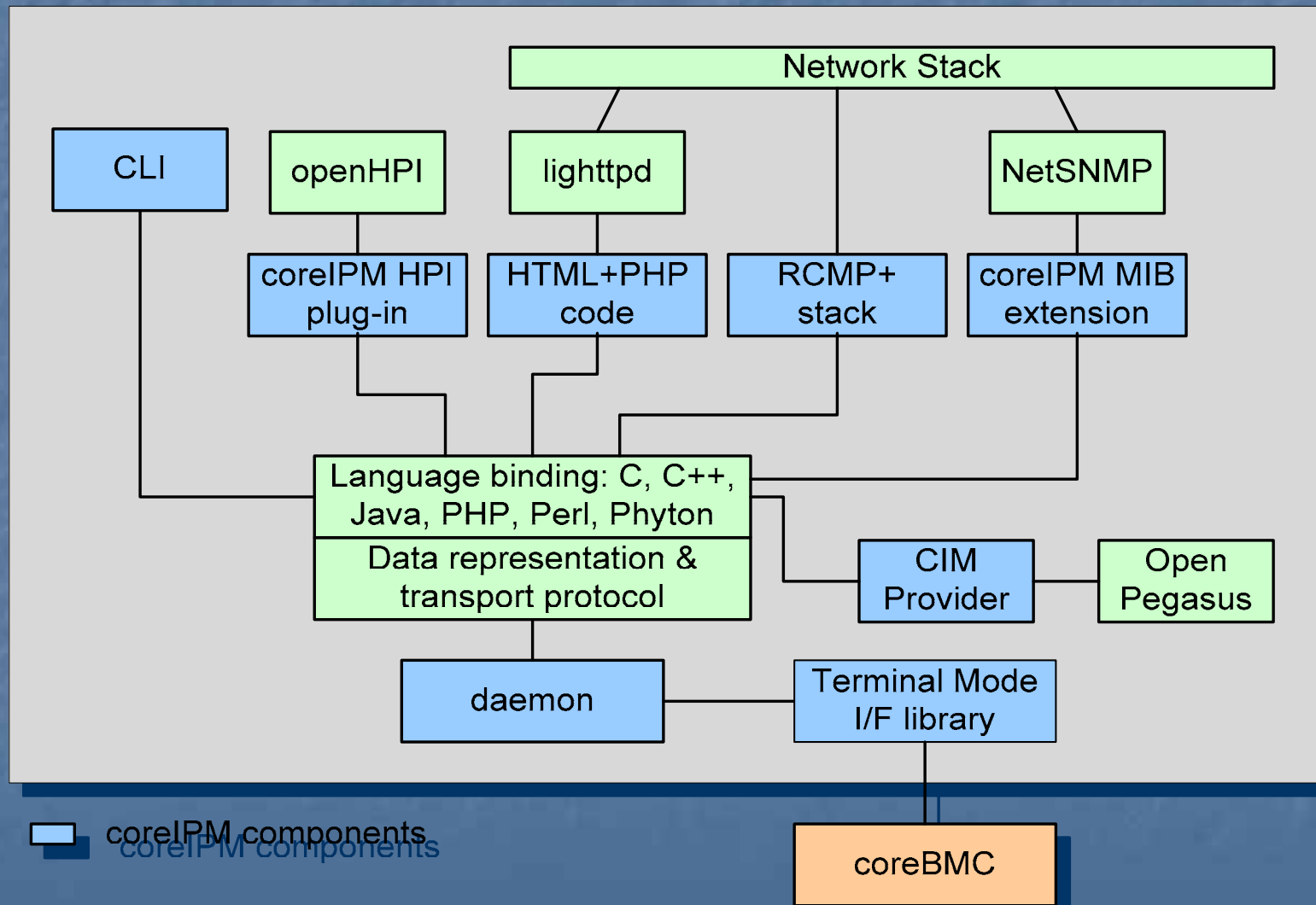
We are aiming to provide support files for 3 build & debug environments:

- Eclipse IDE + gcc (free)
- Keil IDE + CARM/RealView/gcc (commercial - \$\$)
- Rowley CrossWorks IDE + gcc (commercial - \$)

Debugging & Testing coreIPM

- Linux based coretest application is part of the source release. CLI based, easily extensible.
- CLI: a terminal connection is provided on both coreBMC and coreSHM that support IPMI “Terminal Mode” semantics. This enables you to issue IPMI packets and IPMI defined or proprietary control commands from a command line interface.
- ICTS: IPMI conformance test suite. Provides a system for checking pass/fail conformance with the Intelligent Platform Management Interface specification. Linux & windows versions available, from www.intel.com/design/servers/ipmi/tools.htm

coreSMC Architecture



coreIPM-LINUX Distribution

- *coreIPM-LINUX* is a fully fledged Linux distribution for OMAP 35xx & ARM9 processors with built in support for coreIPM management architecture.
- *coreIPM-LINUX* provides a ready to use, extremely compact drop in solution for platform management. It is specifically targeted towards shelf and appliance management.
- See “coreIPM-LINUX Release Notes” on our web site for building and usage.

coreIPM-LINUX Features

- coreIPM interface for controlling multiple i2c buses and general purpose IO pins for latches, LEDs, backplane addressing, HW control and other uses
- RS232 debug port & command line interface
- RMCP+ network interface for sending and receiving IPMI commands
- SNMP Trap generation, SNMP server & MIB for platform management.
- OpenHPI server with coreIPM OpenHPI plug-in.
OpenPegasus WBEM server with coreIPM provider plug-in.

SAF/HPI: Hardware Platform Interface

- SAF/HPI complements IPMI
- HPI provides a standard and hardware independent service to upper level management software to set and retrieve configuration or operational data about the hardware components, and to control the operation of those components.
- Standard library API for watchdogs, FRUs, SELs, events / traps, etc.
- Portable across OS's– Portable under all mngt services
- Abstracts away all layers under OS

OpenHPI

- HPI is defined as a library API of C-library functions
- OpenHPI provides an open source implementation of HPI.
- OpenHPI includes a Plug-in Application Binary Interface (ABI): an internal interface designed for developers to easily write modules for a specific platform with ease.
- There is a bundled plug-in based on the popular OpenIPMI library for low level IPMI access and supports rack-mount servers as well as AdvancedTCA.
- We have our own plug-in if you want to bypass OpenIPMI.

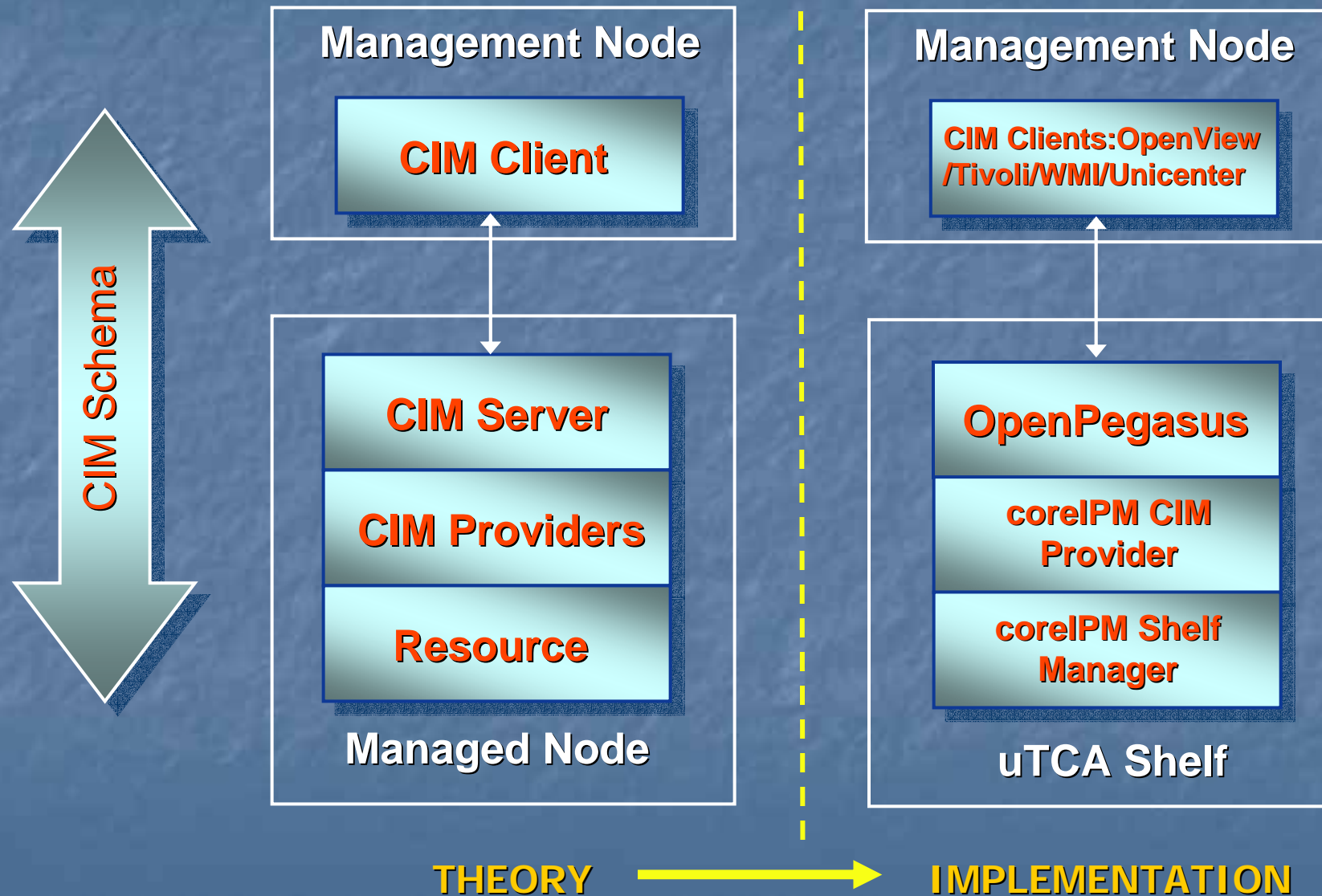
coreIPM HPI Plug-in Functions

chpi_open	chpi_get_sensor_event_masks	chpi_del_announce
chpi_close	chpi_set_sensor_event_masks	chpi_get_annunc_mode
chpi_get_event	chpi_get_control_state	chpi_set_annunc_mode
chpi_discover_resources	chpi_set_control_state	chpi_hotswap_policy_cancel
chpi_set_resource_tag	chpi_get_idr_info	chpi_get_hotswap_state
chpi_set_resource_severity	chpi_get_idr_area_header	chpi_set_autoinsert_timeout
chpi_get_el_info	chpi_add_idr_area	chpi_set_hotswap_state
chpi_get_el_caps	chpi_add_idr_area_id	chpi_request_hotswap_action
chpi_set_el_time	chpi_del_idr_area	chpi_get_autoextract_timeout
chpi_add_el_entry	chpi_get_idr_field	chpi_set_autoextract_timeout
chpi_get_el_entry	chpi_add_idr_field	chpi_get_power_state
chpi_clear_el	chpi_add_idr_field_id	chpi_set_power_state
chpi_set_el_state	chpi_set_idr_field	chpi_get_indicator_state
chpi_reset_el_overflow	chpi_del_idr_field	chpi_set_indicator_state
chpi_get_sensor_reading	chpi_get_watchdog_info	chpi_control_parm
chpi_get_sensor_thresholds	chpi_set_watchdog_info	chpi_load_id_get
chpi_set_sensor_thresholds	chpi_reset_watchdog	chpi_load_id_set
chpi_get_sensor_enable	chpi_get_next_announce	chpi_get_reset_state
chpi_set_sensor_enable	chpi_get_announce	chpi_set_reset_state
chpi_get_sensor_event_enables	chpi_ack_announce	chpi_inject_event
chpi_set_sensor_event_enables	chpi_add_announce	

WBEM/CIM

- Web-Based Enterprise Management (WBEM) is a platform and resource-independent standard that defines both a model (description) and protocol (interface) for monitoring and controlling diverse resources from diverse platforms.
- The Common Information Model (CIM) is the DMTF WBEM standard for describing data.
- www.dmtf.org

Common Information Model (CIM)



IPMI Mapping to CIM

- CIM_Sensor
- CIM_Associations
- CIM_FRU
- CIM_MessageLog
- CIM_LogRecord
- CIM_Watchdog

coreIPM CIM PROVIDER

- Sensor Data Records
- Entity Associations
- Field Replaceable Units
- System Event Log
- Watchdog
- Platform Control

- IPMI WBEM/CIM Provider exposes CIM Hardware profile schema. These are classes that allow administrators to access BMC information through the IPMI Driver.

IPMI Mapping to CIM..example

Each implemented class has methods that can be executed on the managed system and properties that return values. coreIPM CIM provider services these method and property requests in the coreSMC. Here's an example of methods and properties of the CIM_NumericSensor class.

CIM_NumericSensor

METHODS

- Reset
- RequestStateChange

PROPERTIES

- Name
- CurrentState
- PossibleStates
- EnabledState
- HealthState
- OperationStatus
- PollingInterval
- DeviceID
- SensorType
- CurrentReading
- BaseUnits
- UnitModifier
- MaxReadable
- MinReadable
- Resolution
- Accuracy
- IsLinear
- Hysteresis
- SupportedThresholds
- EnableThresholds
- SettableThresholds
- RateUnits
- NominalReading
- NormalMax
- NormalMin

CIM classes implemented by coreIPM Provider for OpenPegasus

- ComputerSystem
- AdminDomain
- SystemSpecificCollection
- RecordLog
- LogRecord
- Sensor
- NumericSensor
- CoolingDevice
- AlarmDevice
- PowerSupply
- Watchdog
- Sensor
- ManagementController

Thank you !

Visit our web site for the updated
version of this tutorial and
documentation.

coreIPM

www.coreipm.com

info@coreipm.com