advanced TCA® summit

Open Source Tools for AdvancedTCA

coreIPM

An Open Source IPMI Implementation

Gokhan Sozmen

coreIPM Project Leader

Santa Clara, CA   USA
October 2007

Agenda

- Platform management refresher. Introduction to coreIPM.
- System design for ATCA platform management, using the coreIPM open source IPMI software on an ARM7 microcontroller.

Participants will (hopefully) leave with sufficient information to implement a working ATCA management solution.

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.

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 architecture ..components.

coreIPM architecture is comprised of:

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

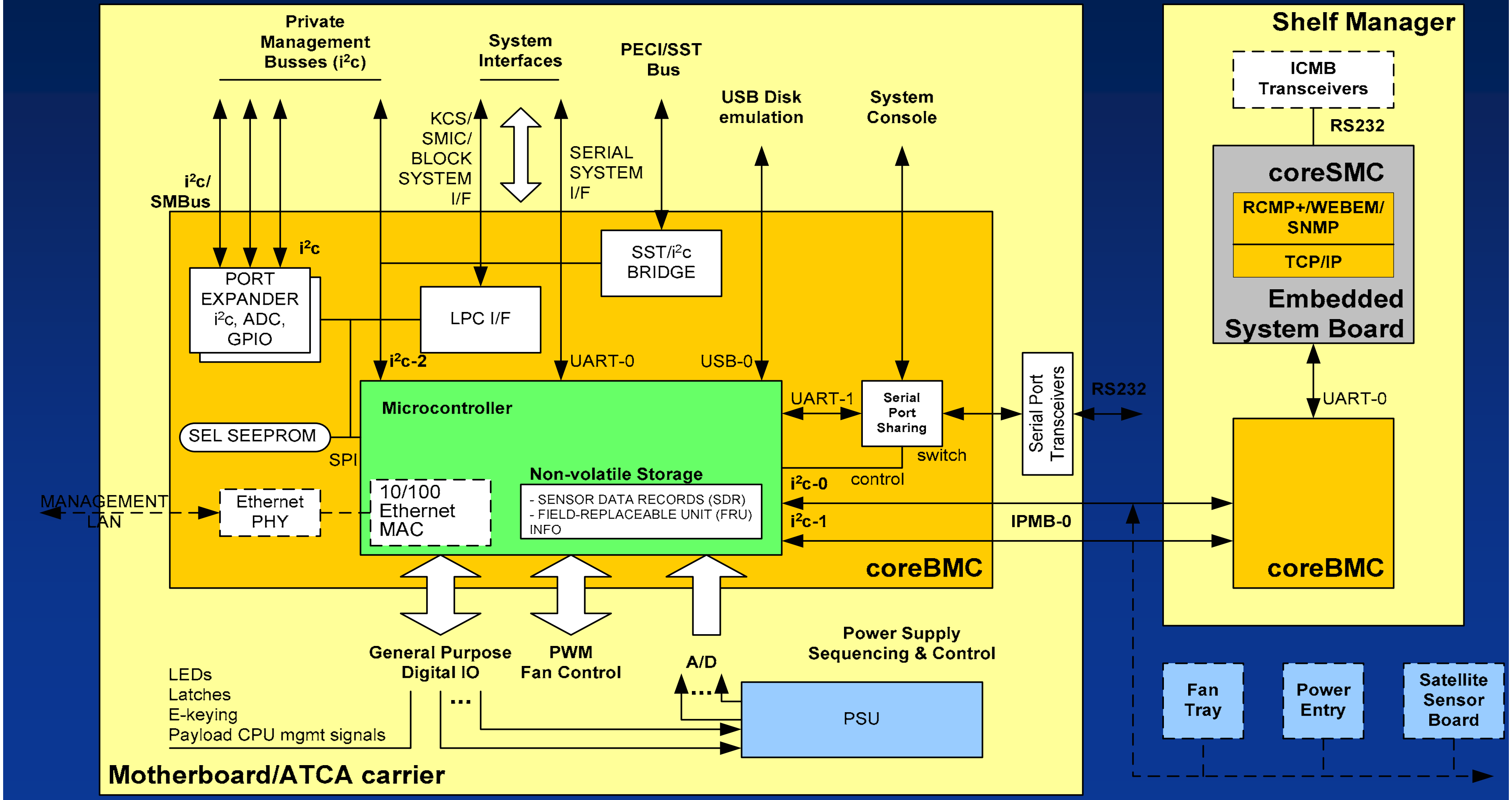
coreBMC is fully sufficient by itself to provide board management functions for ATCA, AMC, microTCA (Carrier manager, Power module, Fan), OPMA, motherboard, appliance applications. Coupled With ***coreSMC*** it's possible to build shelves and higher hardware hierarchies such as an ATCA chassis.

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.

- ***coreSMC*** incorporates networking stack + RCMP + SNMP + WEBEM + access control + high availability functions.

coreIPM architecture ..Hardware



Shelf Manager

ICMB Transceivers

RS232

coreSMC

RCMP+/WEBEM/ SNMP

TCP/IP

Embedded System Board

UART-0

coreBMC

Private Management Busses (i^2c)

System Interfaces

PECI/SST Bus

USB Disk emulation

System Console

i^2c/ SMBus

KCS/ SMIC/ BLOCK SYSTEM I/F

SERIAL SYSTEM I/F

i^2c

PORT EXPANDER i^2c, ADC, GPIO

SST/i^2c BRIDGE

LPC I/F

i^2c-2

UART-0

USB-0

Microcontroller

UART-1

Serial Port Sharing

switch

control

Serial Port Transceivers

RS232

SEL SEEPROM

SPI

MANAGEMENT LAN

Ethernet PHY

10/100 Ethernet MAC

Non-volatile Storage

- SENSOR DATA RECORDS (SDR)
- FIELD-REPLACEABLE UNIT (FRU) INFO

i^2c-0

i^2c-1

IPMB-0

coreBMC

General Purpose Digital IO

PWM Fan Control

A/D

Power Supply Sequencing & Control

…

…

PSU

LEDs
Latches
E-keying
Payload CPU mgmt signals

Motherboard/ATCA carrier

Fan Tray

Power Entry

Satellite Sensor Board

coreBMC architecture

- Small footprint designed to be embedded in a single chip microcontroller.

- Initial support for the NXP 32-bit LPC2000 microcontroller family based on ARM7 core. Use the controller best suited for the task. Currently 32+ microcontroller devices to choose from. High performance and low power consumption in a cost-effective package. Cost competitive with 8 and 16 bit microcontrollers.

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 RCMP+, SNMP, Web I/F. Hardware independent code enables porting and usage of embedded platform and OS of choice for the shelf manager.

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.

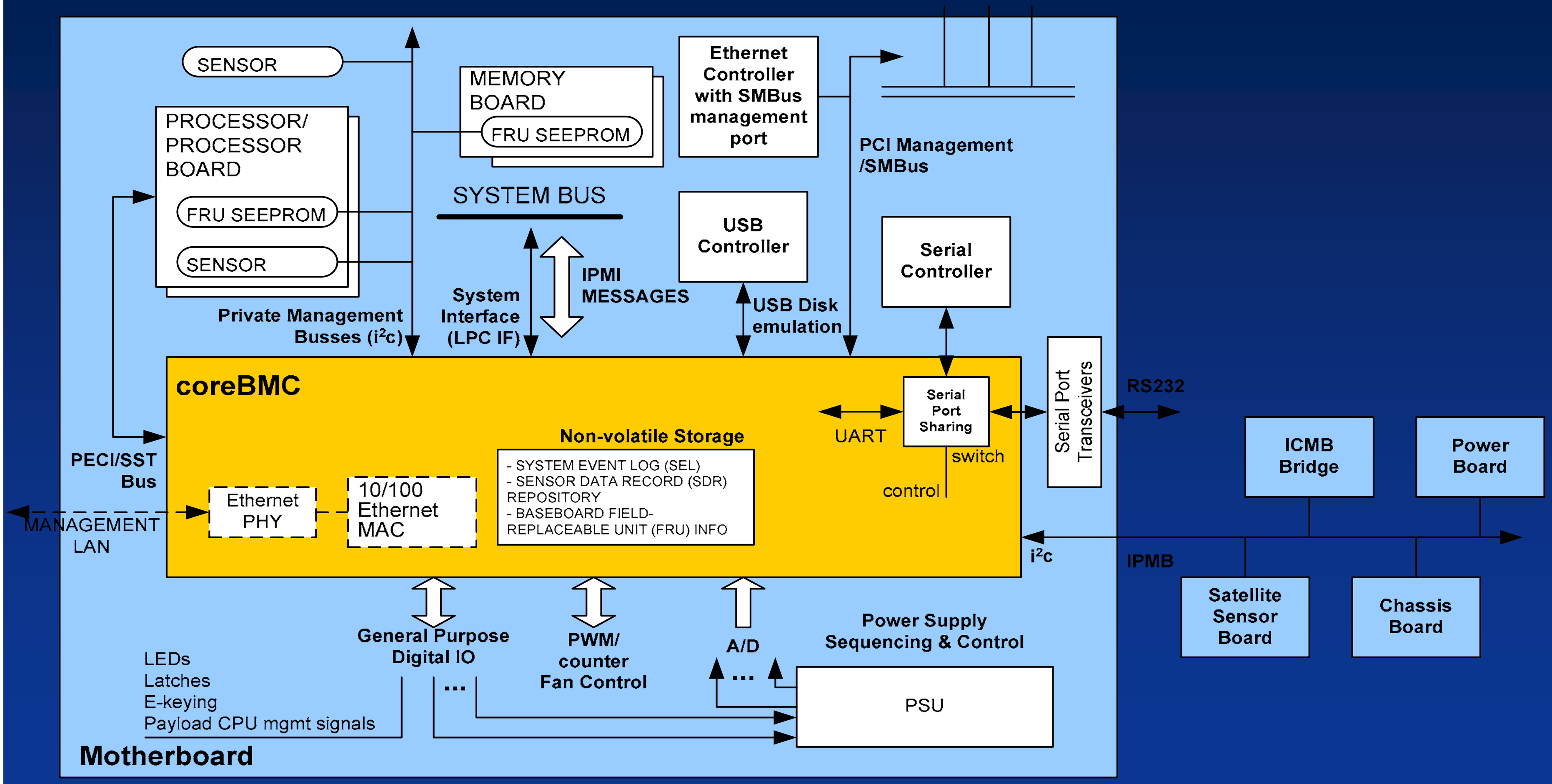
Why does Open Source matter ?

- Users and developers are partners, success is based on cooperation. No adversarial relationship
- Community support builds a distributed network of experts. People will help you build your solution. Skilled manpower easier to find. Collective wisdom of demanding users aggregated in forums.
- Everyone can participate in testing the next release. No surprises.
- No hassle evaluation, no time limits
- No messy legal paperwork
- Commercial 3rd party support & development: support not tied to the viability of a particular vendor. Support business is open to competition and market forces. Service suppliers have to stay the best. They do not own anything you depend on.
- Code usage flexibility: port, re-use, re-target
- Open for peer review; critisism welcome, encouraged
- No forced ugrades; upgrading a technical decision not a financial one
- Insight to the development process. Monitor development activities. No reliance on a vendor's delivery estimate. You can influence schedules and priorities of the development process by contributing.
- Direct access to developers

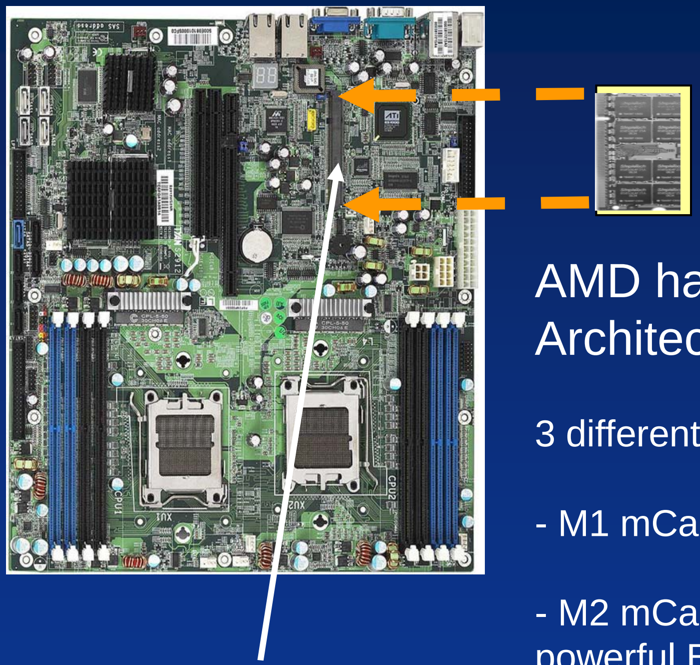
Where is coreIPM used ?

- In the following slides we show some applications for coreIPMI controllers. Functions shown in yellow boxes are handled by coreIPM.

Motherboard application of coreIPM



Motherboard application of coreIPM cont…



OPMA plug-in management controller (mCard)

SODIMM OPMA connector

AMD has defined the Open Platform Management Architecture (OPMA) based on IPMI

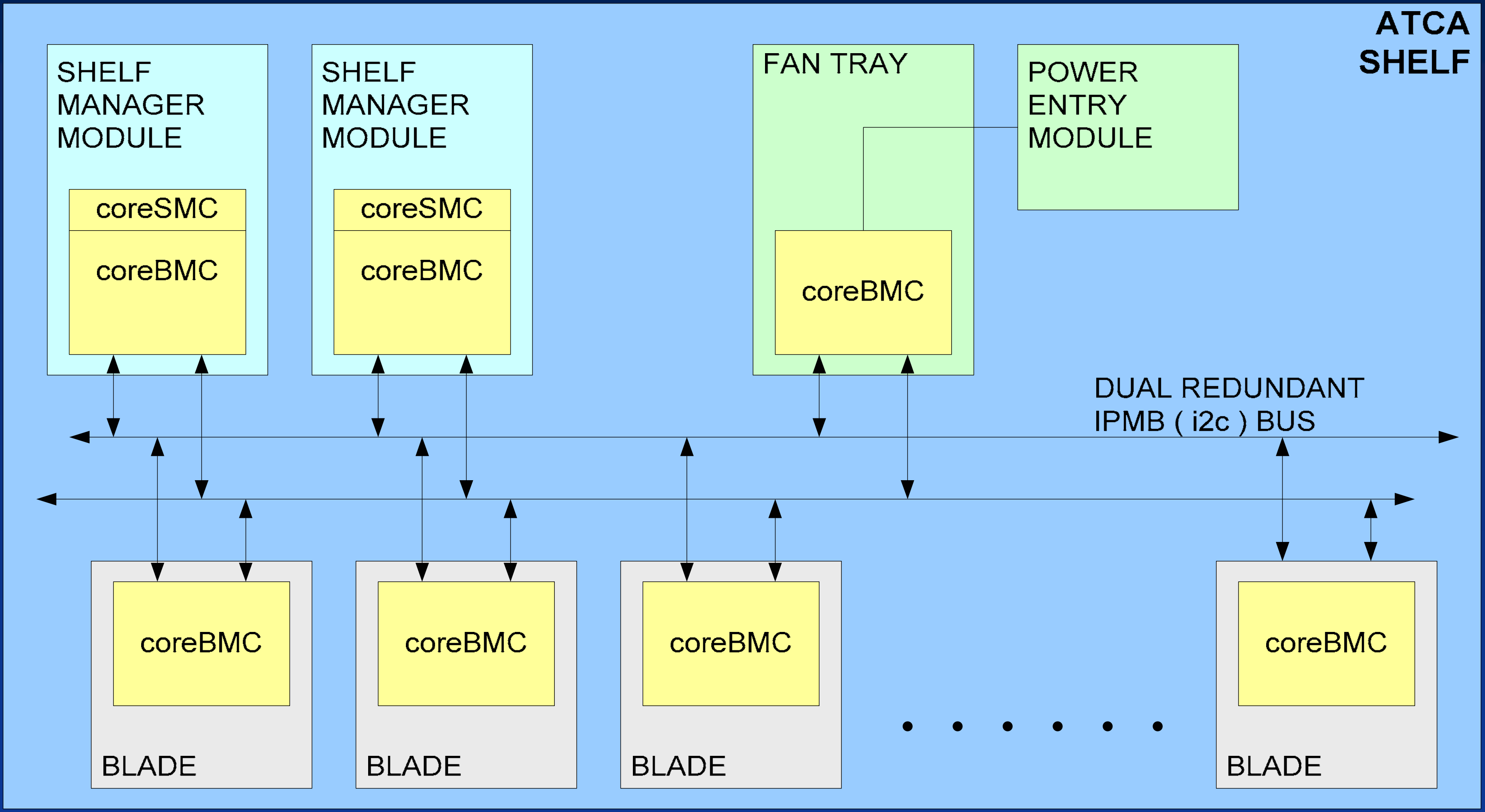
3 different mCard types are defined:

- M1 mCard uses the motherboard NIC. Basic solution

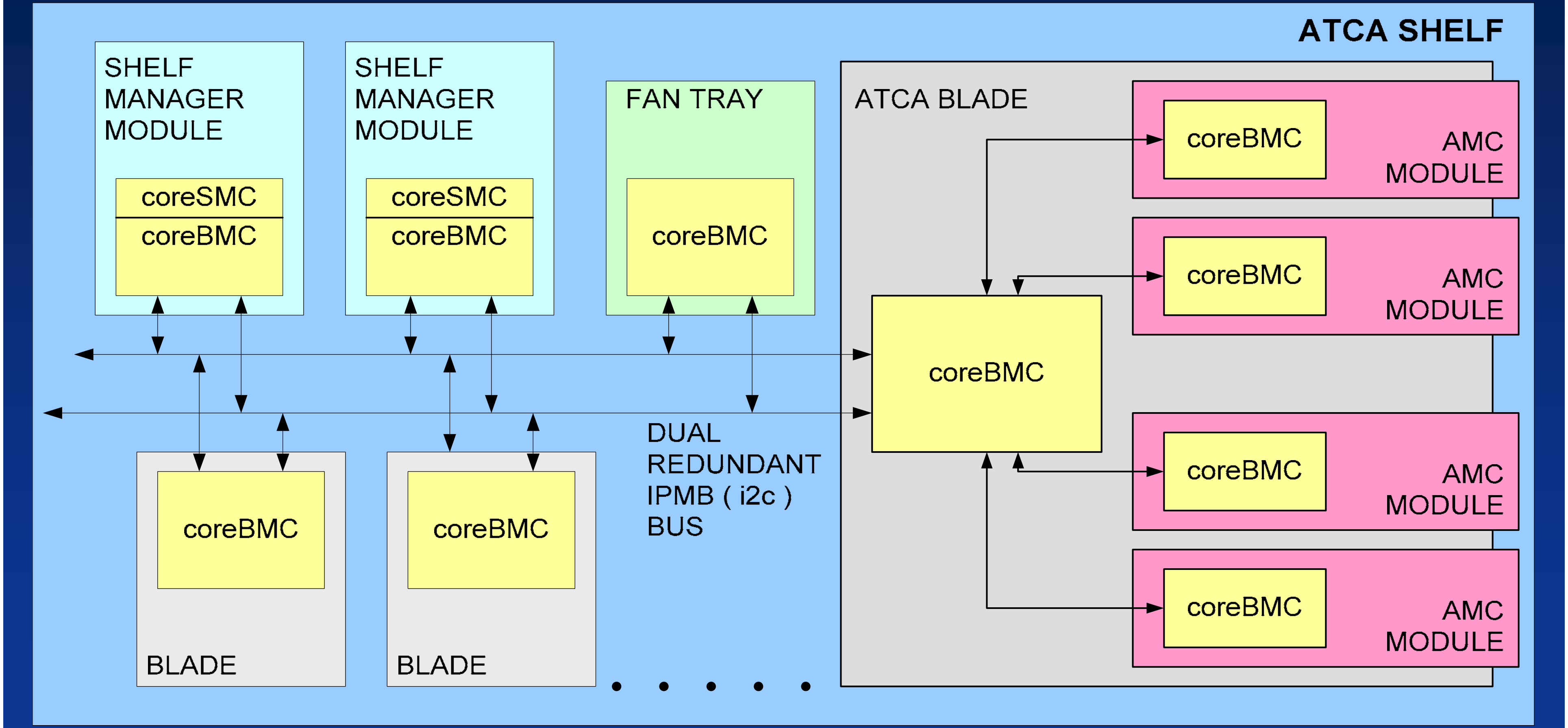
- M2 mCard uses it's own dedicated management NIC. Implies a more powerful BMC CPU core, larger address space, more flash and RAM. Possibly implements remote virtual mass storage.

- M3 mCard Adds KVMoverIP. Requires video capture and compression capability. The M3 solution provides KVMoIP without consuming a PCI slot, giving it an advantage over current solutions existing in today's market.

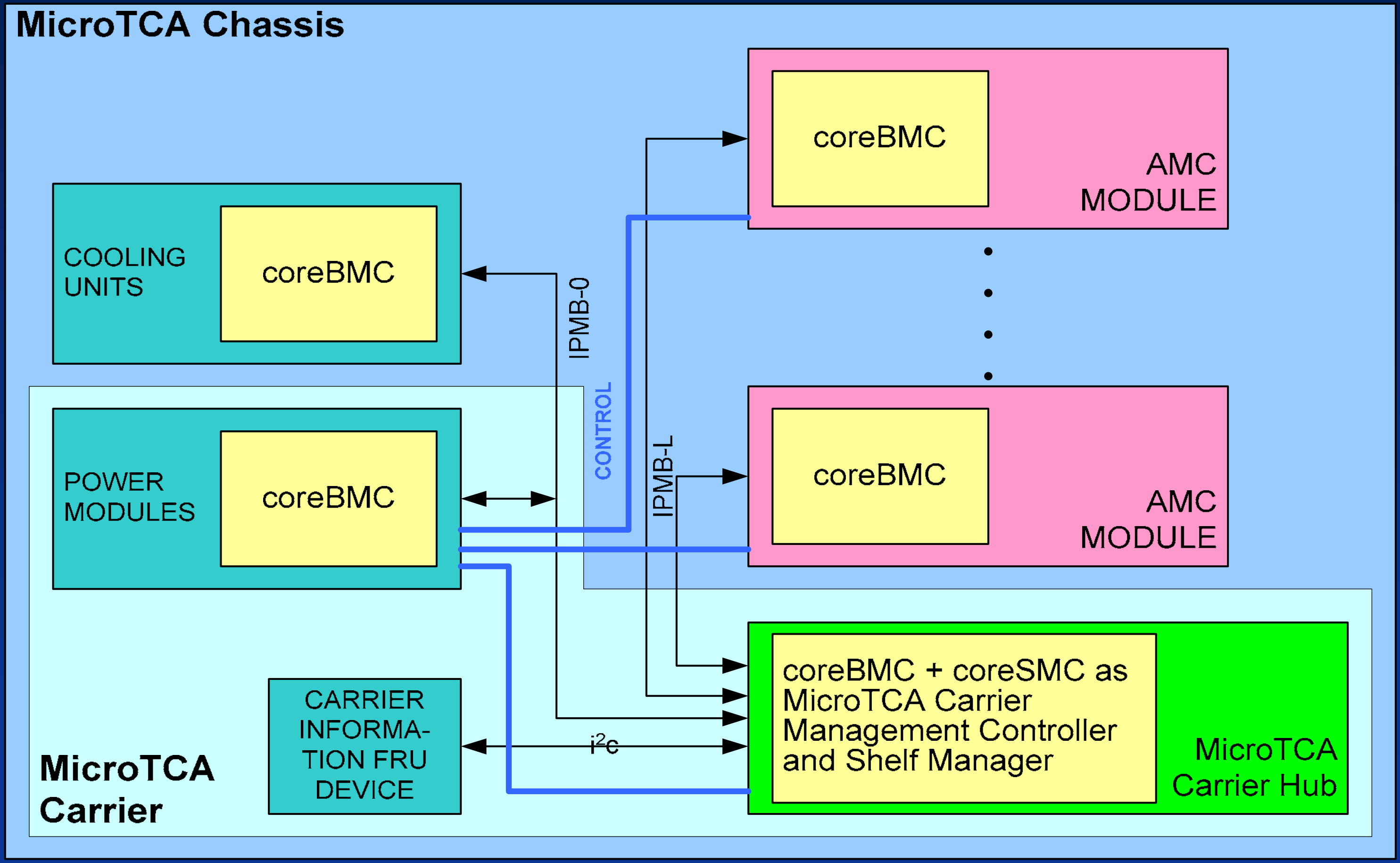
Blade application of coreIPM
ATCA Management Infrastructure



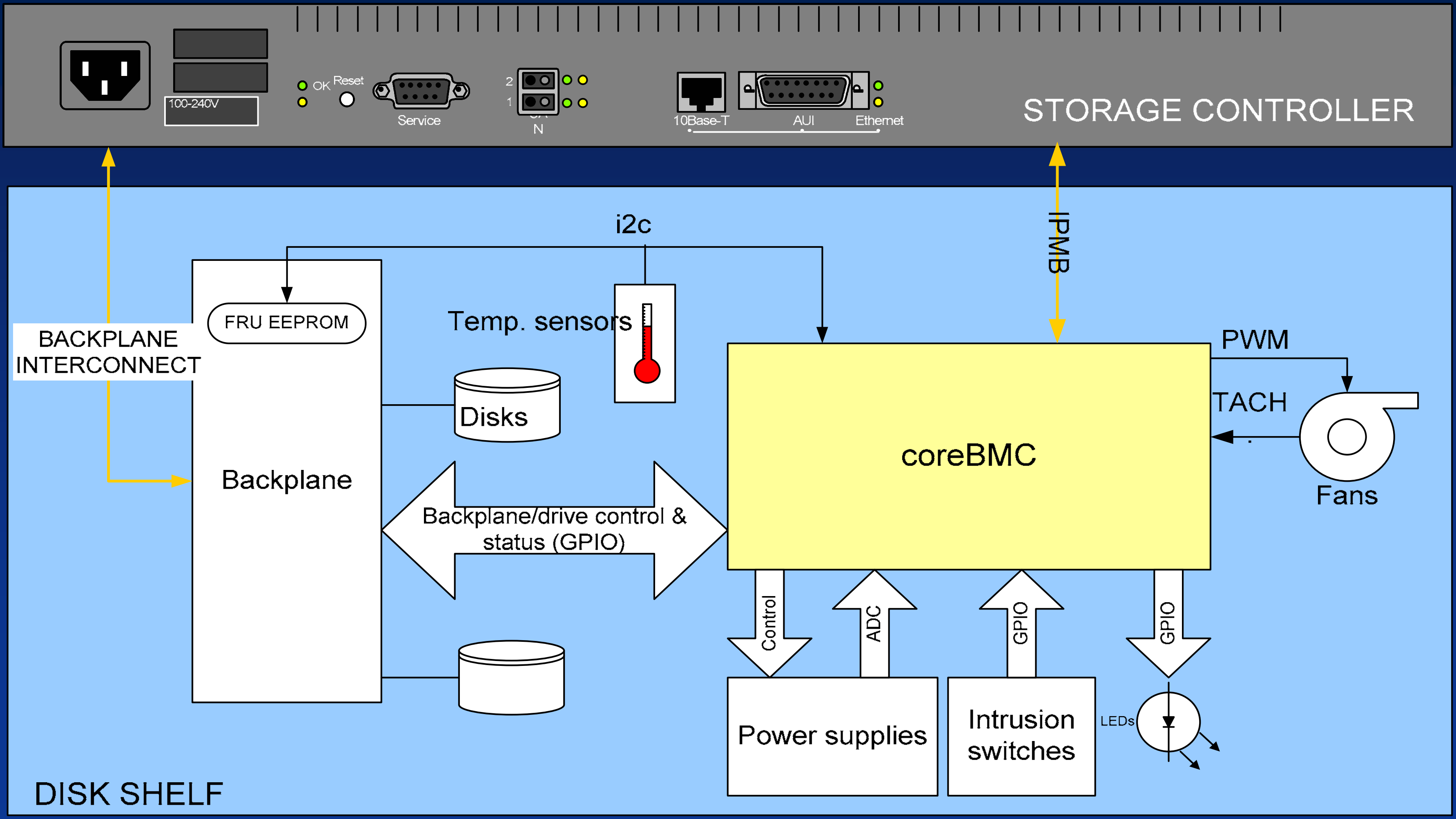
Blade application of coreIPM cont…
AMC Management Infrastructure



Blade application of IPMI cont…
MicroTCA Management Infrastructure



Enclosure management using coreIPM



coreIPM architecture ..HW features.

Supported hardware features

- i^{2}c
- Timers
- General purpose IO pins for latches, LEDs, E-keying and other uses
- RS232
- SPI : port expanders, SEEPROM
- ADC
- PWM/tach fan control
- Watchdog Timer and Real-Time Clock with battery backup
- Edge or level sensitive interrupt pins
- Ethernet (on select microcontrollers)
- USB (on select microcontrollers)

coreIPM architecture ..Messaging.

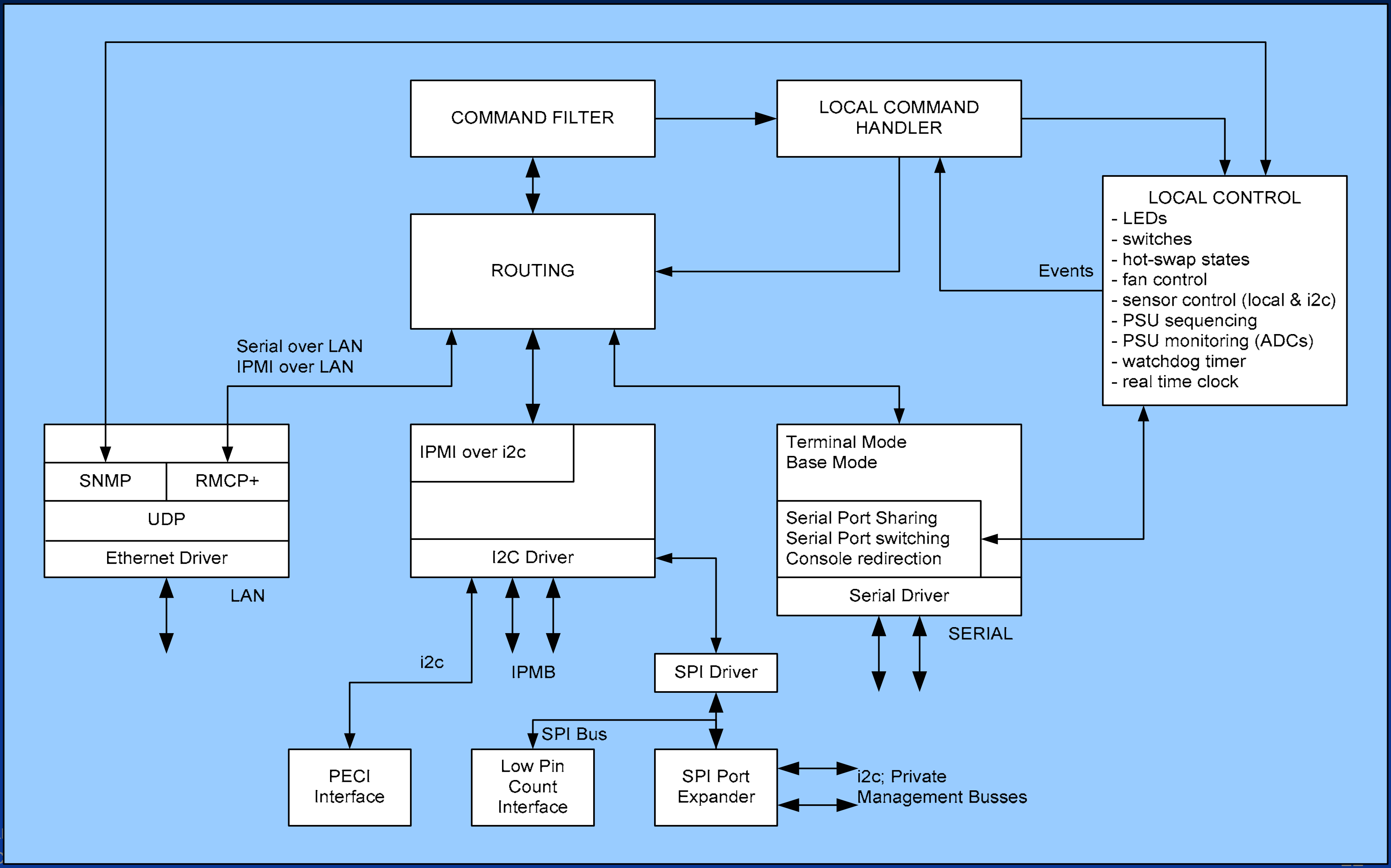
- ***coreBMC Messaging interfaces***
  - System Interface - Terminal mode (to payload or shelf manager).
    Supported by OpenIPMI
  - Serial Interface - Terminal & Base mode
    Serial port sharing
    Serial port switching
    Console redirection
  - IPMB, IPMB-0, IPMB-L (i^{2}c)
  - LAN

- ***Message Bridging***
  Routing between System Interface, Serial Interface, IPMB and LAN.

coreIPM architecture ..Diagnostics.

- ***Diagnostics***
  - coretest: Linux based test program for exercising IPMI features. Also used for firmware updates.
  - 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.

coreIPM architecture .. SW modules



COMMAND FILTER

LOCAL COMMAND HANDLER

ROUTING

LOCAL CONTROL
- LEDs
- switches
- hot-swap states
- fan control
- sensor control (local & i2c)
- PSU sequencing
- PSU monitoring (ADCs)
- watchdog timer
- real time clock

Events

Serial over LAN
IPMI over LAN

SNMP

RMCP+

UDP

Ethernet Driver

LAN

IPMI over i2c

I2C Driver

Terminal Mode
Base Mode

Serial Port Sharing
Serial Port switching
Console redirection

Serial Driver

SERIAL

i2c

IPMB

SPI Driver

SPI Bus

PECI Interface

Low Pin Count Interface

SPI Port Expander

i2c; Private Management Busses

Building a platform management subsystem

- Download source and tools
  - www.coreipm.com has pointers to the source, build tools and latest instructions.
- Configure
- Build
- Debug
- Test

Configuring coreBMC

Need to interconnect the hardware, device drivers and target requirements:

- Hardware: microcontroller type, board interconnections …
- Drivers: i^2c, fan control, SPI, sensors, serial, GPIO …
- Targets: ATCA, AMC, MicroTCA, OPMA, Power entry module, Fan tray, Smart power strip, Appliance etc.

Configuring coreBMC .. Interconnections

- WIRING TABLE
  Defines logical to physical mapping

```
#define I2C0        LPC2138_I2C_BASE
#define SERIAL_CONSOLE     ...
#define IPMB0_A…
#define BLUE_LED        …
etc
```

Configuring coreBMC .. Sensors

Sensors currently supported:

- On chip ADC (1-n)
- On chip GPIO
  - Hot swap switches
  - AMC signals
- i^{2}c based temperature sensors

To configure sensors, make an entry in the Sensor Data Record table and add a corresponding entry to the sensor table

```
SENSOR_TABLE sensor[] = {
      sensor_id;
      scan_handler = adc_sensor/i2c_sensor/gpio_sensor
}

SDR sdr_entry_table [] = {
}
```

Configuring coreBMC .. Cont.

Similar table entries required for configuring:

- LEDs
- Fans
- Platform Event Filtering
- Serial ports

Refer to the coreIPM User’s Manual.

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.

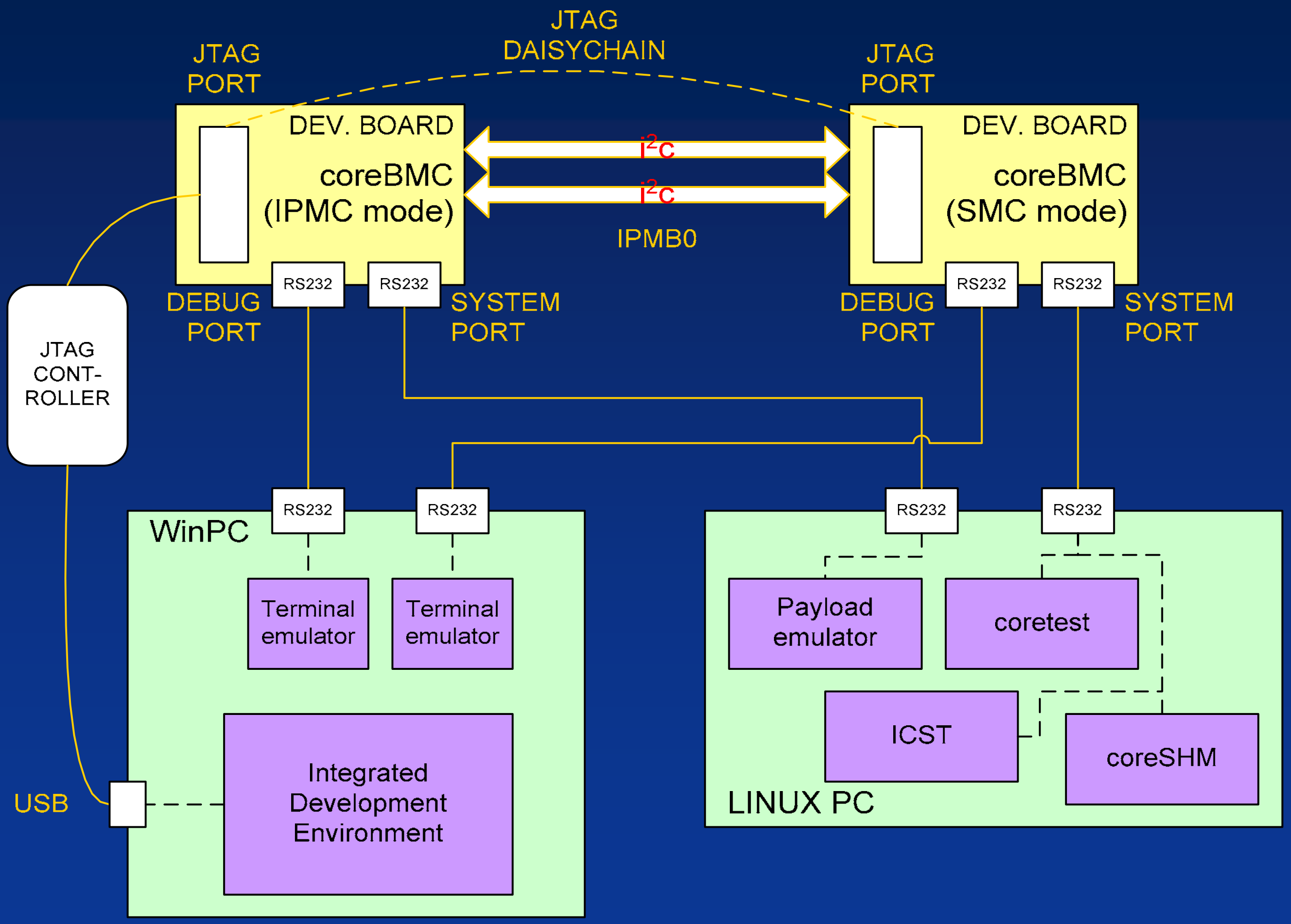
We provide support files for 3 build & debug environments:

- Eclipse IDE + gcc (free)
- Keil IDE + CARM/RealView (commercial - $$) (Note: For evaluation purposes the free Keil evaluation version will build & debug coreBMC)
- Rowley CrossWorks IDE + gcc (commercial - $)

Debugging & Testing coreIPM

- We use inexpensive development boards for the microcontroller to build a target configuration. Target code type is selected by compile switches.
- Linux based coretest application is part of the source release. CLI based, easily extensible.
- 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

Debugging & Testing coreIPM



ATCA CONFIGURATION

Project status

- coreBMC initial alpha released June ‘07
- AMC, MicroATCA extensions due Nov. ’07
- Estimated Beta release 1Q08
- First design win in Sept ’07, ongoing commercial development. Project is funded and viable.
- 20K+ lines of code
- < 32K Flash, < 4K RAM required
- All mandatory commands supported + more
- System testing & development ongoing on a daily basis.

Work in Progress

- coreSHM shelf manager
  - Linux based, runs on any platform. Develop, debug & test on a desktop, transfer to an embedded system.
  - Uses coreBMC as a front end
  - RCMP+, CLI, SNMP, HPI plugin, web browser interfaces

Work in Progress .. Cont.

- coreBMC-XL
  - TCP/IP stack with RCMP+, SNMP, Serial over LAN, httpd on the ARM7 microcontroller.
  - Low cost shelf manager
  - Needed for motherboard/OPMA support

Work in Progress .. Cont.

- coreBMC-XL server motherboard support
  - LPC (Low pin count) Interface for system interface.
  - Platform Environment Control Interface (using Maxim PECI-to-i^{2}c translators)
  - USB disk emulation for remote boot & remote keyboard, mouse support.

coreIPM .. Conclusion

- Join the project ! Developers & users wanted.

Utilize coreIPM Group Expertise

We provide

- Software solutions, commercial licensing, support, and custom engineering services for platform management.
- ATCA/microTCA Hardware design
- Embedded systems consulting

Thank you !

coreIPM

www.coreipm.com

info@coreipm.com