

Progress in Defining Physics Software Architectures and Protocols for ATCA and MTCA.4 Standards and Guidelines

A.P. Lowell

Triple Ring Technologies, Newark, CA

alowell@TripleRingTech.com

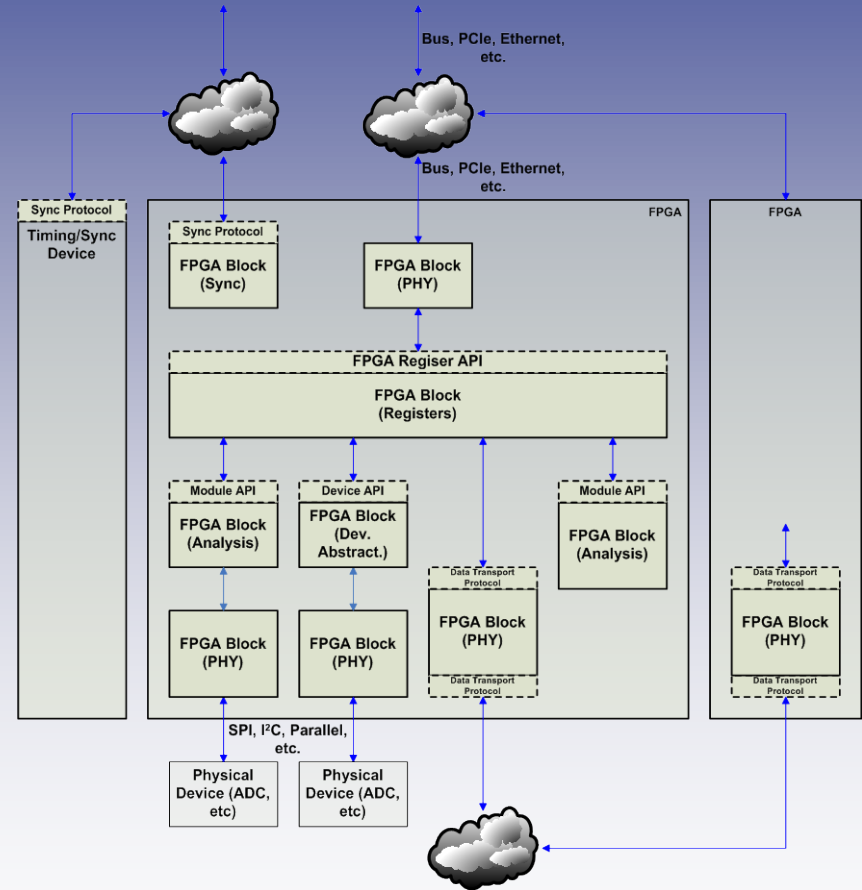
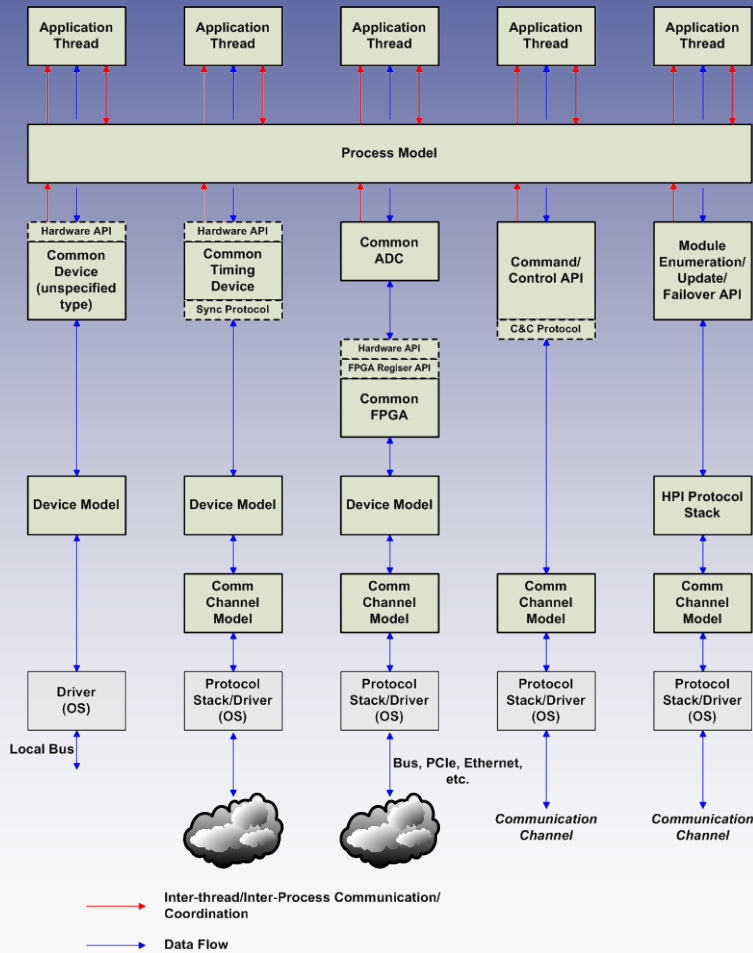
Committee Charter

- Chartered as PICMG Committee
 - Coordinates with Steering Committee and Hardware Committee
 - Purpose is:
 - *To facilitate inter-operability/interchangeability between facilities and projects by defining standard interfaces*
 - *To facilitate availability of COTS solutions*
 - *To reduce software development time and cost*
- by defining common techniques and modules for software development

Committee Topics

- *Primary Focus:* In-rack and embedded components
- *Preferred Approach:* Identify, integrate, and provide examples for existing industry standards
- *Expected Products:* Guidelines (not standards) and reference implementations
- *General Topics*
 - System/Shelf/Module management
 - Routing and Protocols
 - Operating systems and infrastructure
 - Common processing and operations libraries

Application Example



System/Shelf/Module Management

- Remote management functions
 - Shelf/Module identification
 - Health/Status monitoring
 - Redundancy/Failover management
 - Operating resource management
 - In-field application software/firmware upgrade
- *Primary Effort:* Document use of existing IPMI/HPM facilities for novice developers
- Example extensions
 - E-keying for analog signal formats
 - SW/FW update protocol via base/data fabrics

Routing and Protocols

- Data Protocols
 - Low/Medium/High latency
 - Identify standard protocols for common use
 - Generate low-latency protocol(s) as necessary
 - Software: connection & traffic mgmt.
- Timing/Synchronization Protocol
 - Clocks/Triggers/Interlocks: ps jitter, ns resolution
 - Distributed on ATCA/ μ TCA fabric channels
 - Software: connection and configuration mgmt.
- Command/Control Protocol
 - Application Oriented: augments IPMI functions
 - Channel-agnostic protocol via base fabric

- “Virtual Machine” for applications
 - Goal: OS- and hardware-independence
 - Standardized Components
 - Process/Thread management model
 - Hierarchical IO/Comm/Timing access model
 - Hardware access model for board development
- Utilize existing standards where available and applicable
- Reference implementations
 - Common OS/Platform environments
 - Common devices

- Commonly-used components
 - Virtualized Device APIs (eg. ADC, DAC, Sync)
 - Standard hardware API (eg. Verilog)
 - Standardized Components
 - Process/Thread management model
 - Hierarchical IO/Comm/Timing access model
 - Hardware access model for board development
 - Standard functions (eg. C&C parser, signal filters)
- Reference Designs (eg. thread-based control loop, signal acquisition loop)

- *Data Transport*: protocols selected for eval.
- *Timing/Sync*: Starting now, post-HW spec
- *Command/Control*: At proposal stage
- *Component Management*: At discussion stage; no OPR assigned
- *Hardware API*: Guideline in-work
- *Thread Model*: Guideline & Ref.Dsn. in-work
- *Device Model*: Guideline & Ref.Dsn. in-work
- *Libraries*: No activity