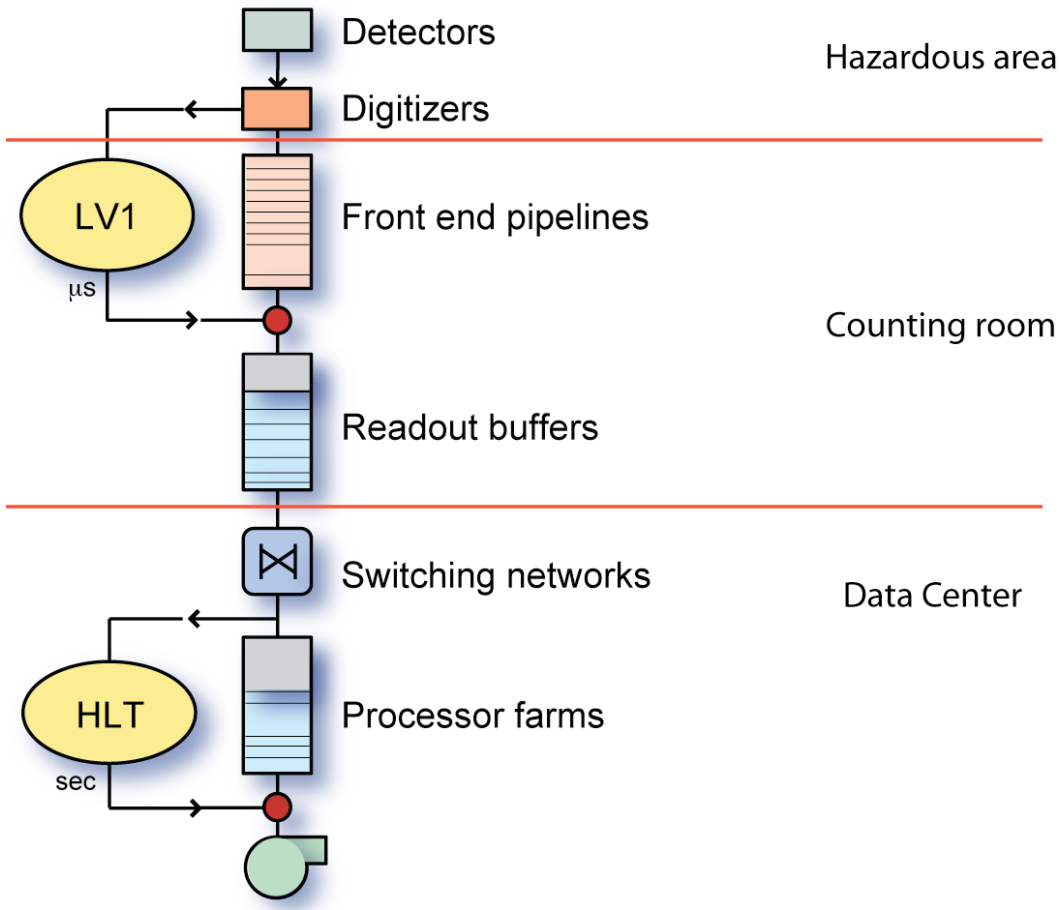
SuperB Readout R&D

M. Bellato

INFN Padova

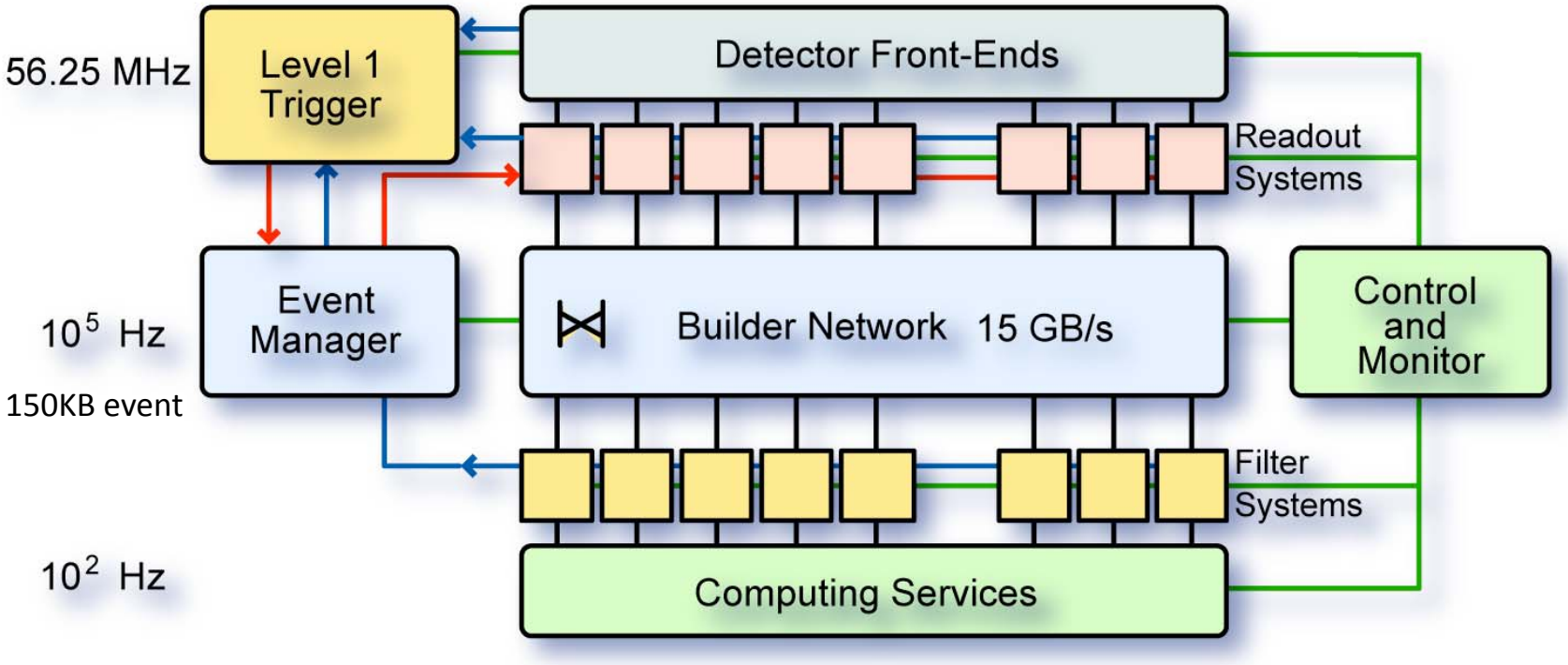
Readout Column



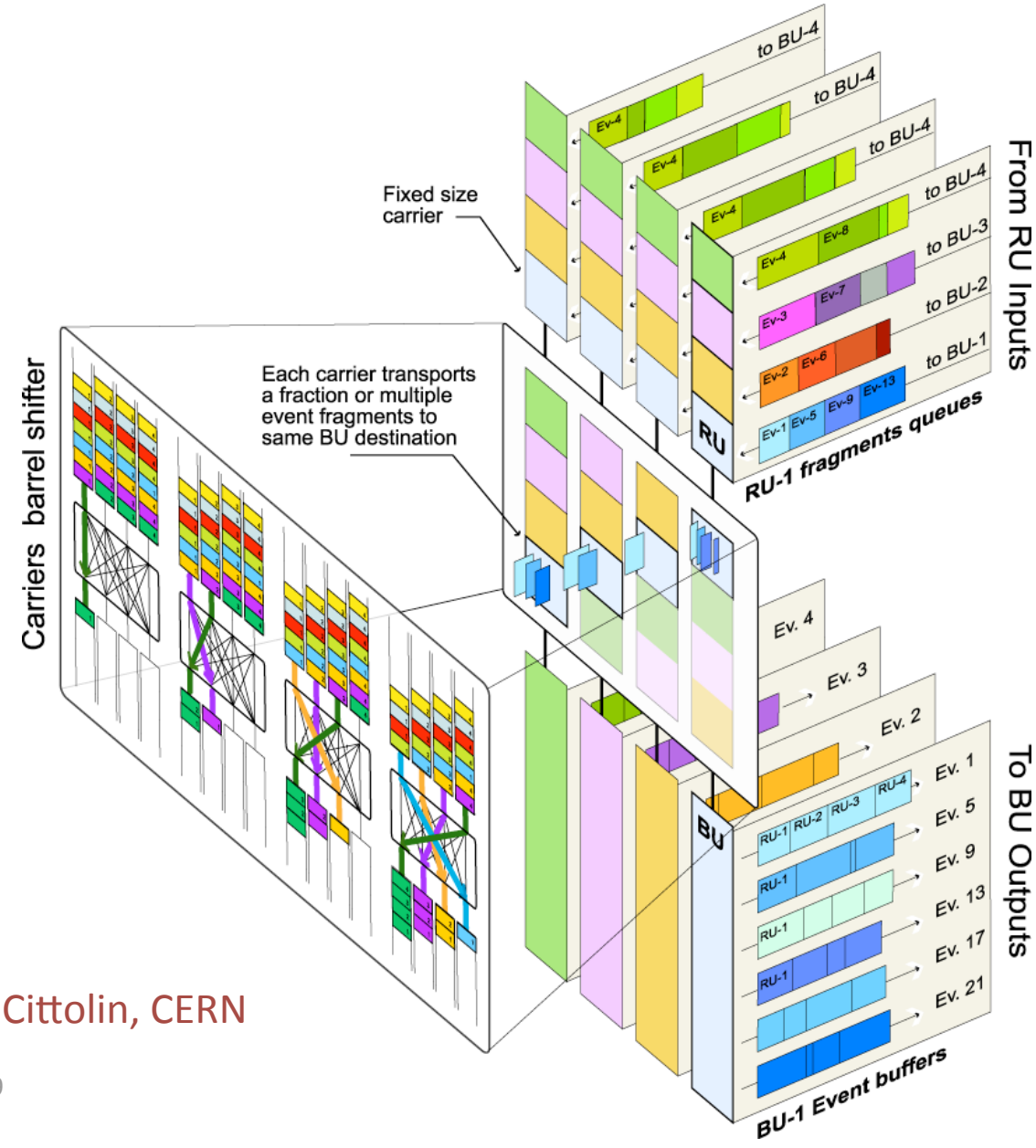
The (usual) problem

- Merge many inputs (front-end data sources) to one selected output (a server in the EVB farm)
- Then, change the output according to a strategy
- N:1 traffic shaping across a switch is natural and often the preferred choice

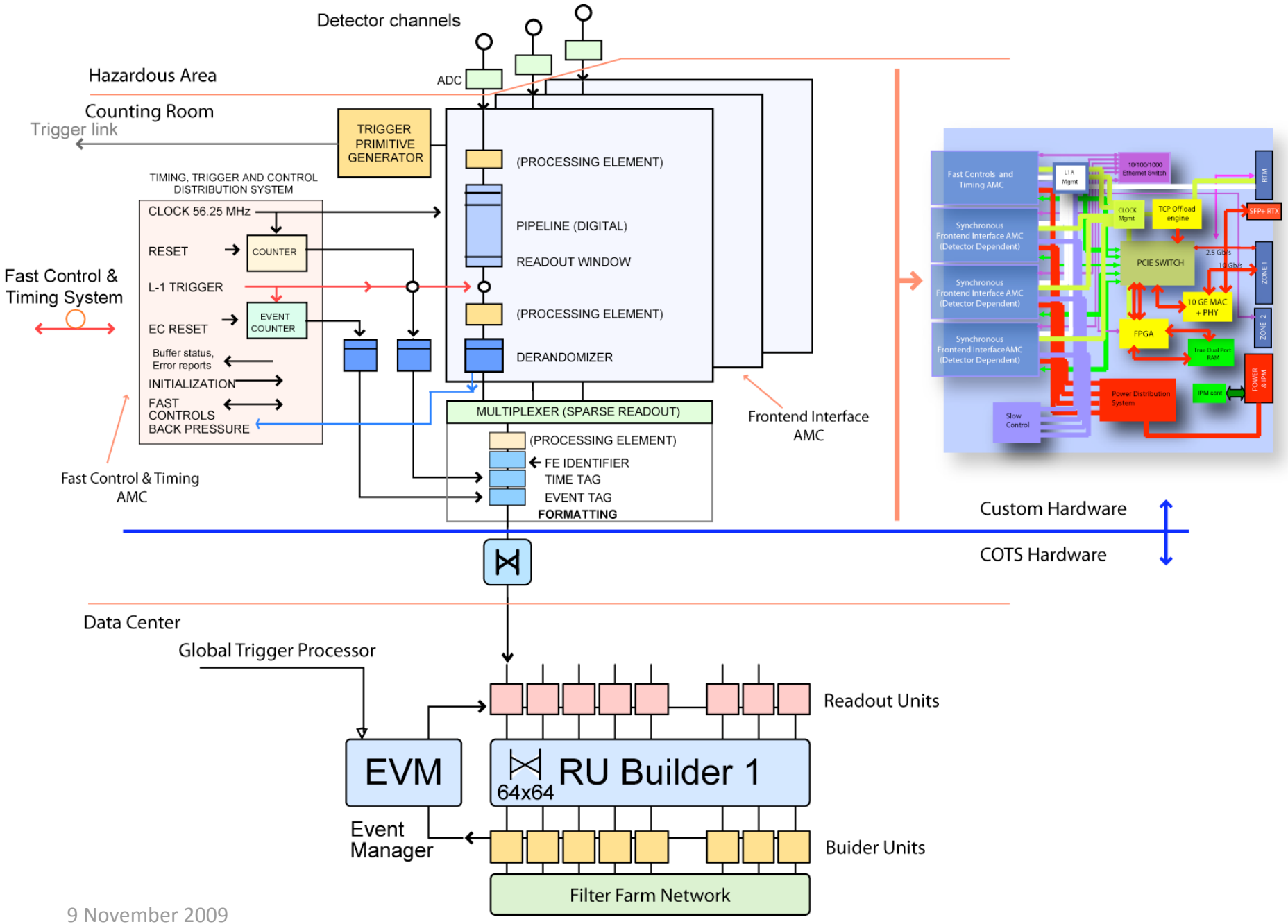
Event Building Network



Buffering in the readout

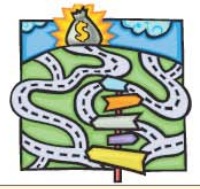


Courtesy of S. Cittolin, CERN


Detector channels
Hazardous Area
ADC
Counting Room
Trigger link
TRIGGER PRIMITIVE GENERATOR
TIMING, TRIGGER AND CONTROL DISTRIBUTION SYSTEM
CLOCK 56.25 MHz
RESET
COUNTER
L-1 TRIGGER
EC RESET
EVENT COUNTER
Buffer status, Error reports
INITIALIZATION
FAST CONTROLS BACK PRESSURE
Fast Control & Timing System
Fast Control & Timing AMC
(PROCESSING ELEMENT)
PIPELINE (DIGITAL)
READOUT WINDOW
(PROCESSING ELEMENT)
DERANDOMIZER
MULTIPLEXER (SPARSE READOUT)
(PROCESSING ELEMENT)
FE IDENTIFIER
TIME TAG
EVENT TAG
FORMATTING
Frontend Interface AMC
Fast Controls and Timing AMC
Synchronous Frontend Interface AMC (Detector Dependent)
Synchronous Frontend Interface AMC (Detector Dependent)
Synchronous Frontend InterfaceAMC (Detector Dependent)
L1A Mgmt
10/100/1000 Ethernet Switch
CLOCK Mgmt
TCP Offload engine
PCIE SWITCH
2.5 Gb/s
10 Gb/s
10 GE MAC + PHY
FPGA
True Dual Port RAM
Power Distribution System
IPM cont
Slow Control
RTM
SFP+ RTX
ZONE 1
ZONE 2
POWER & IPM
Custom Hardware
COTS Hardware
Data Center
Global Trigger Processor
Readout Units
EVM
64x64
RU Builder 1
Event Manager
Buider Units
Filter Farm Network

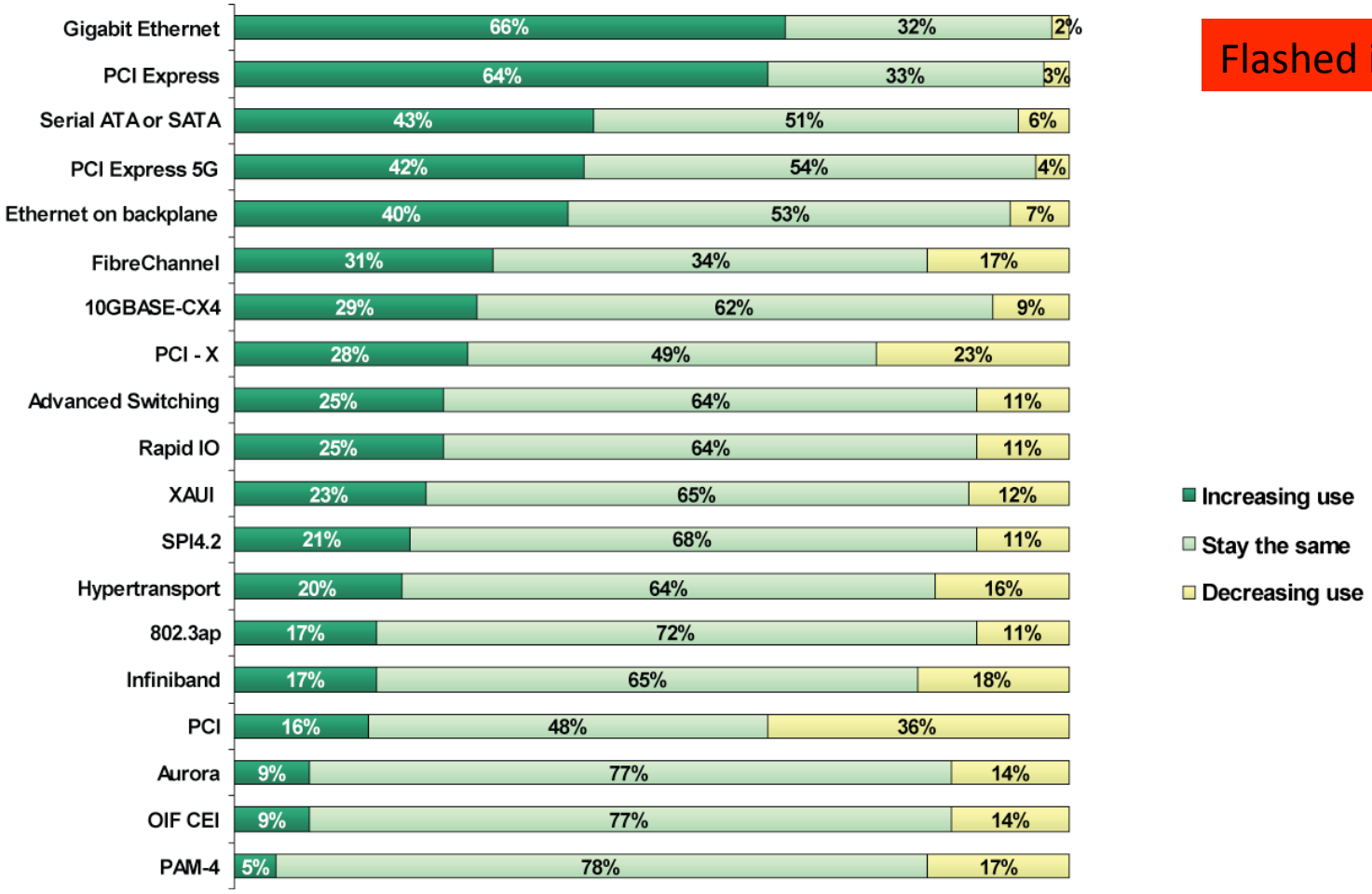
Comparison of Serial Links






	Dedicated Point-to-Point	Manually Switched Point-to-Point	Memory Mapped Fabric	Packet Switched Fabric
Software Reconfigurable Paths	No	Yes	Yes	Yes
Self-Routing Packets	No	No	No	Yes
Automatic Path Re-Routing	No	No	No	Yes
Packet Overhead Required	Low	Low	Med	High
Payload Data Efficiency	High	High	Med	Low
Software Driver Complexity	Low	Low	Med	High
FPGA Interface Complexity	Low	Low	Med	High
Protocol Transparent	Yes	Yes	No	No
Protocols Supported	Aurora VITA 49 PCIe SRIO	Aurora VITA 49 PCIe SRIO	PCIe	SRIO Ethernet

Prospective use of standards

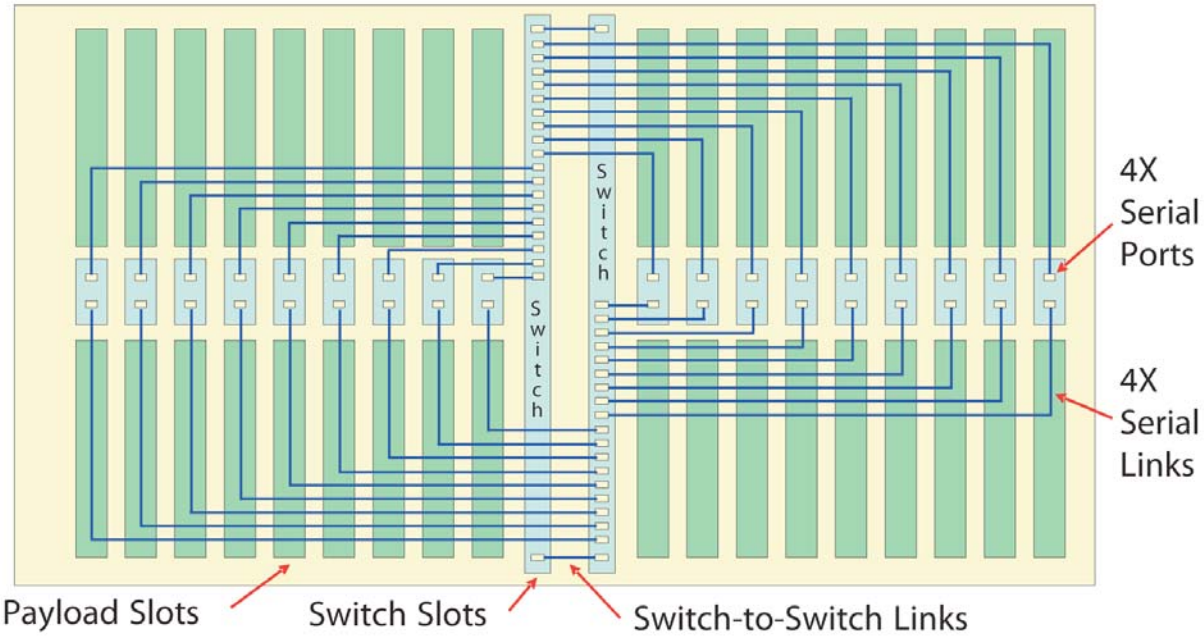


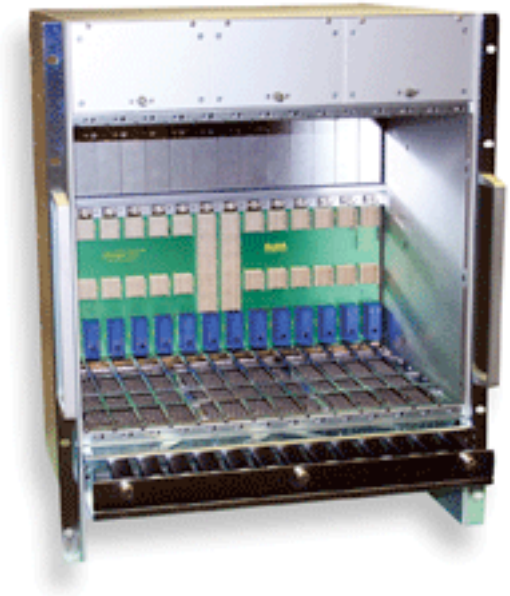
Flashed in 2006

Case study: the Agata Readout – 3 AdvancedTca/MicroTCA Serial Backplanes

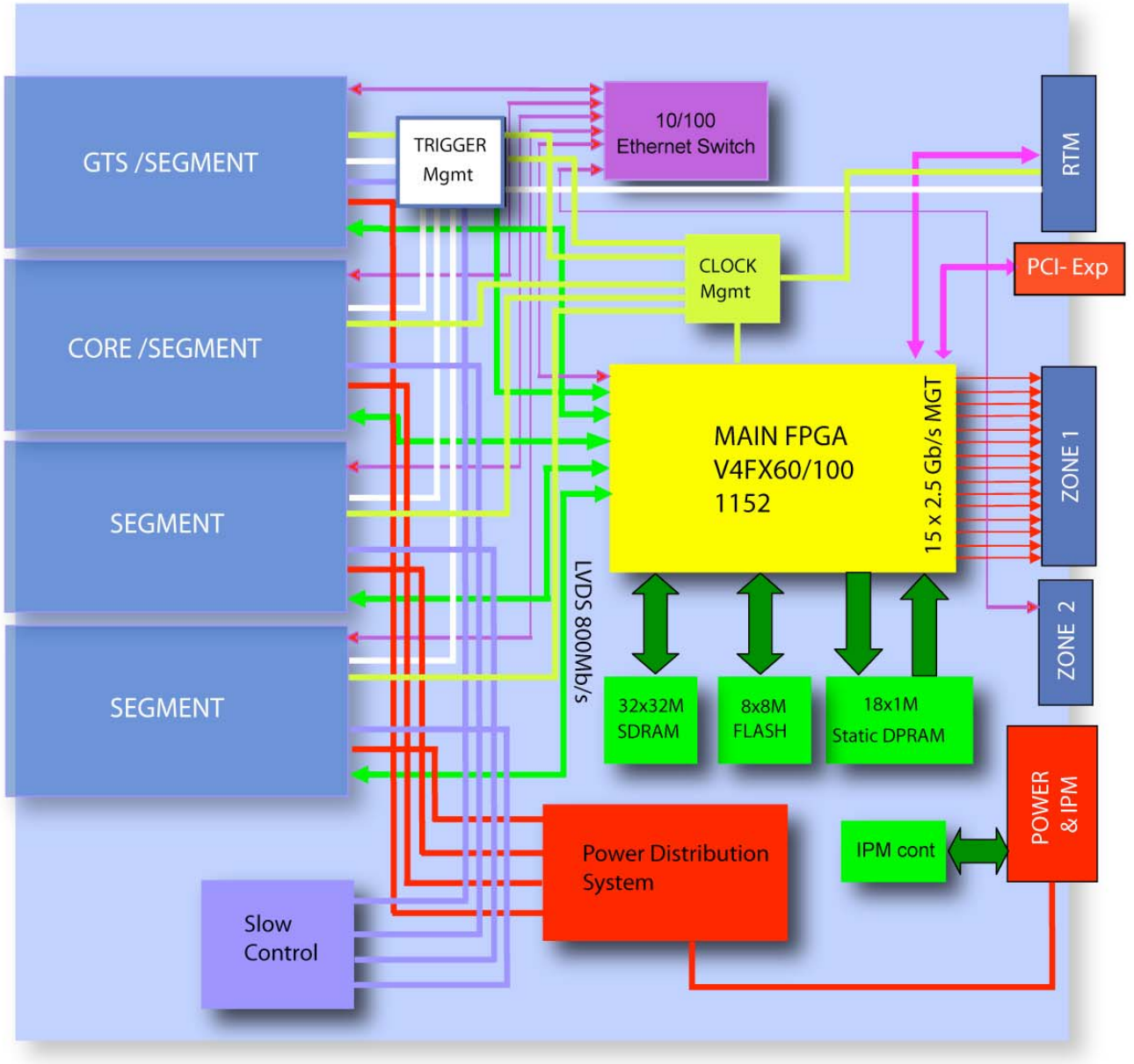
Full mesh

Dual star

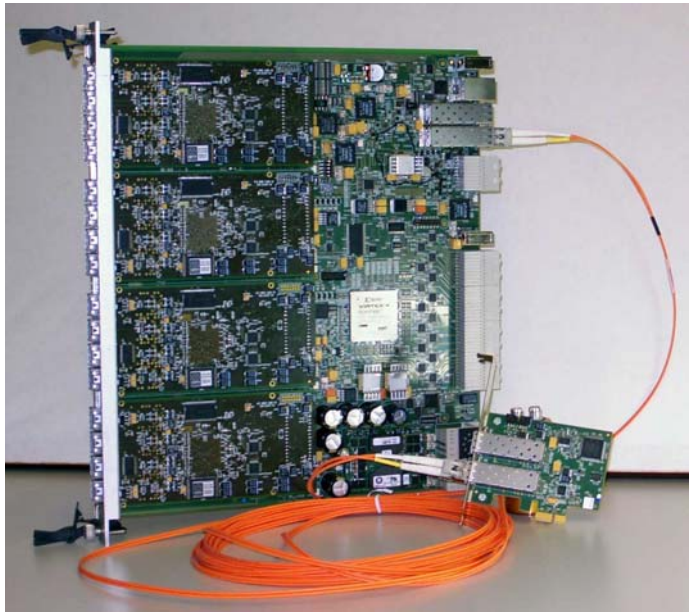


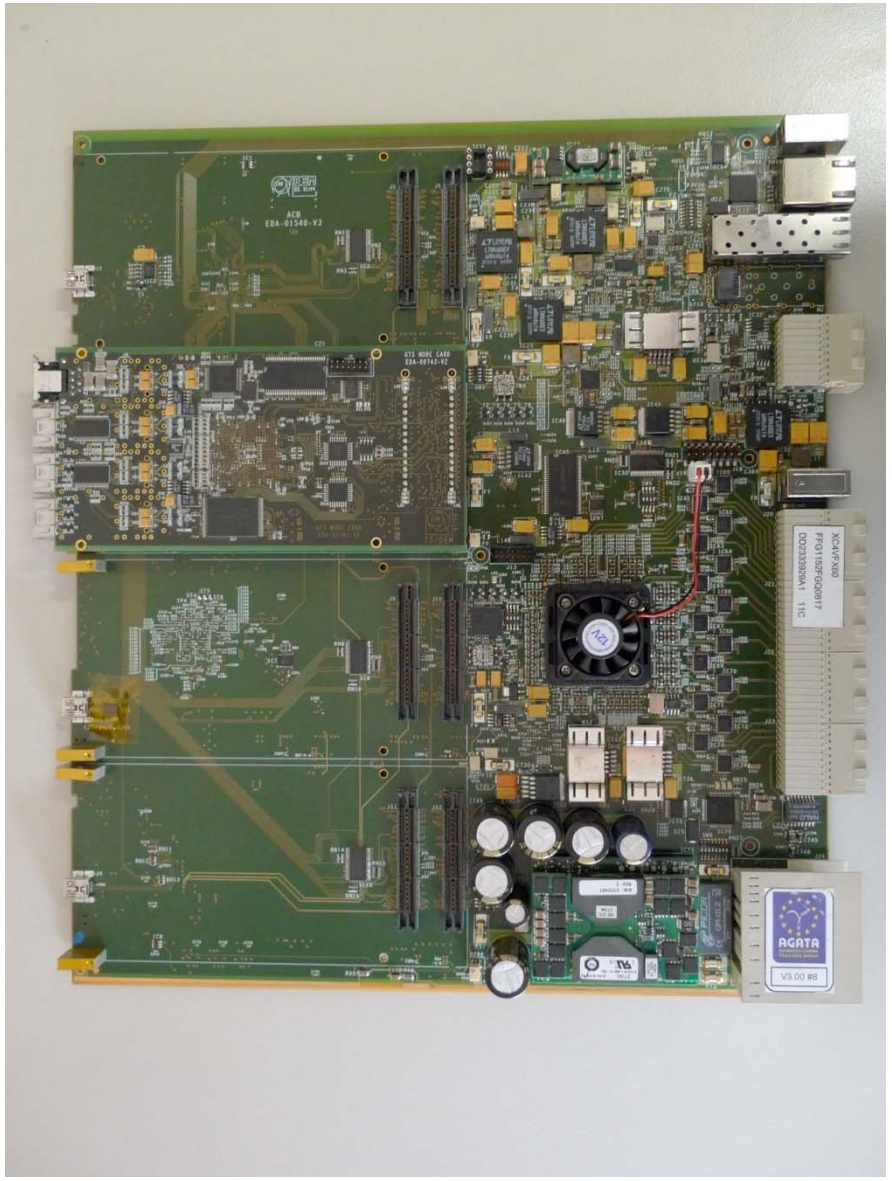


Case study: the Agata Readout – 4
Frontend Readout Card

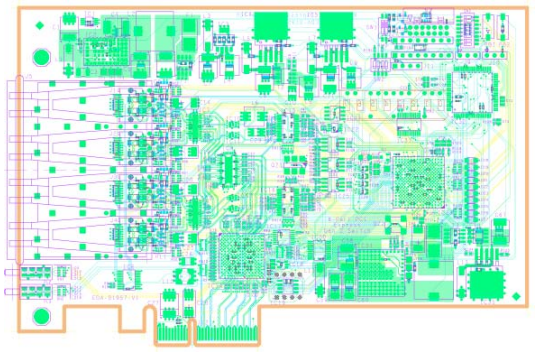


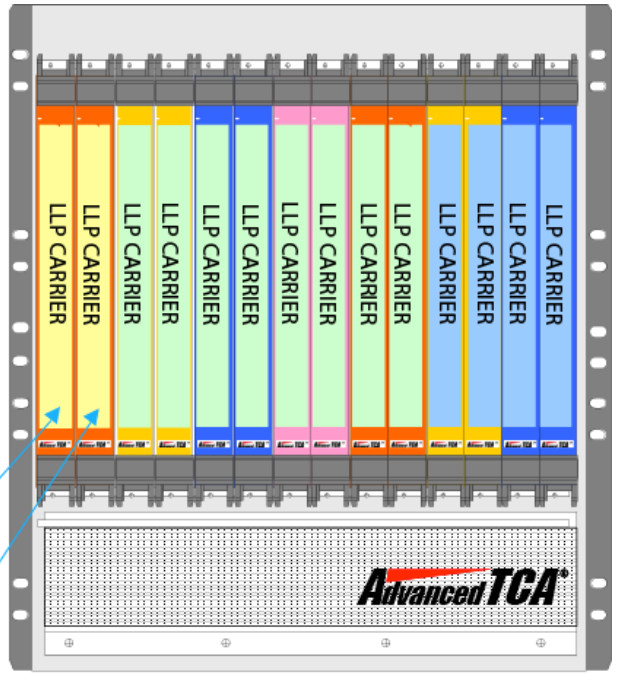
Agata Readout Card





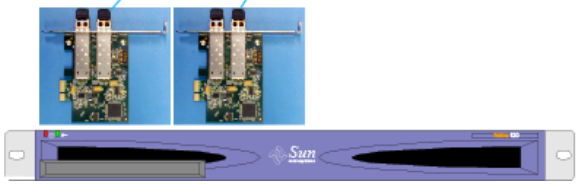
Case study: the Agata Readout – 5
Event builder I/F





BW : 200 Mbytes/s/link

PCI Express Optical link



G-ethernet Rationale

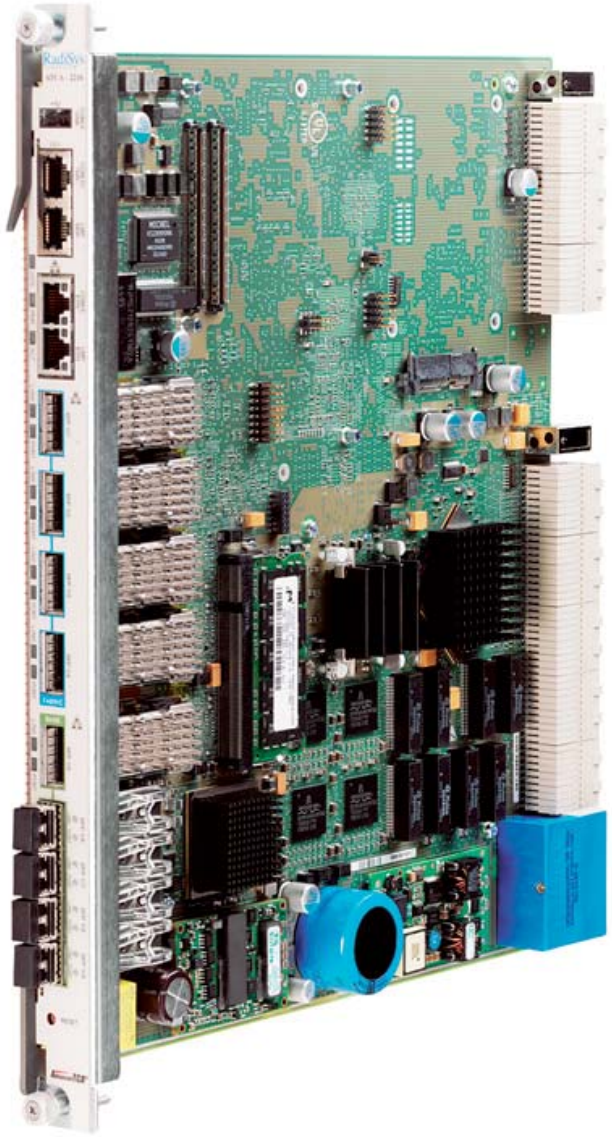
- Ubiquitous, low cost, long life, vast ecosystem
- Scalable to thousands of nodes
- Can feed the EVB directly from readout cards
- Well established for 1Gigathernet
- 10Gethernet not yet well established in the backplane
- Need of TCP offloading engines (TOE's)
- Need to account for latency across switches

TCP/IP offloading - 1

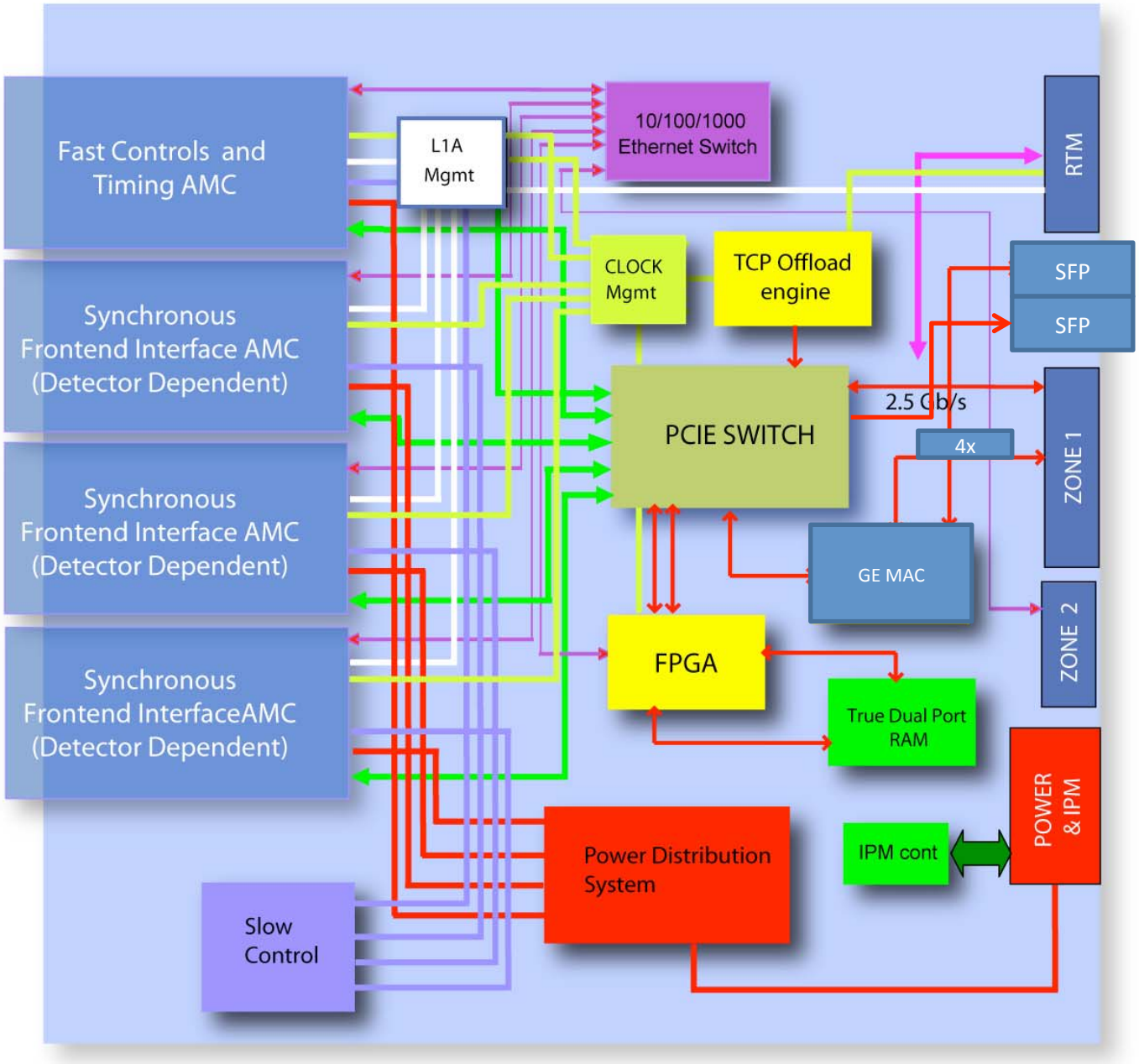
- 2 flavours :
  - Full state offload (chimney)
    - Complex
    - > 70% cpu savings
  - Stateless offload
    - Mainly Receive side scaling and checksum computation
    - ~ 20% cpu savings
- Usually implemented with custom processor(s) in VLSI
- Low acceptance by the Linux community
- All major telecom chipmakers have NIC's with TOE
- We need an R&D phase to test TOEs in event building

COTS : 1/10 Gethernet switches

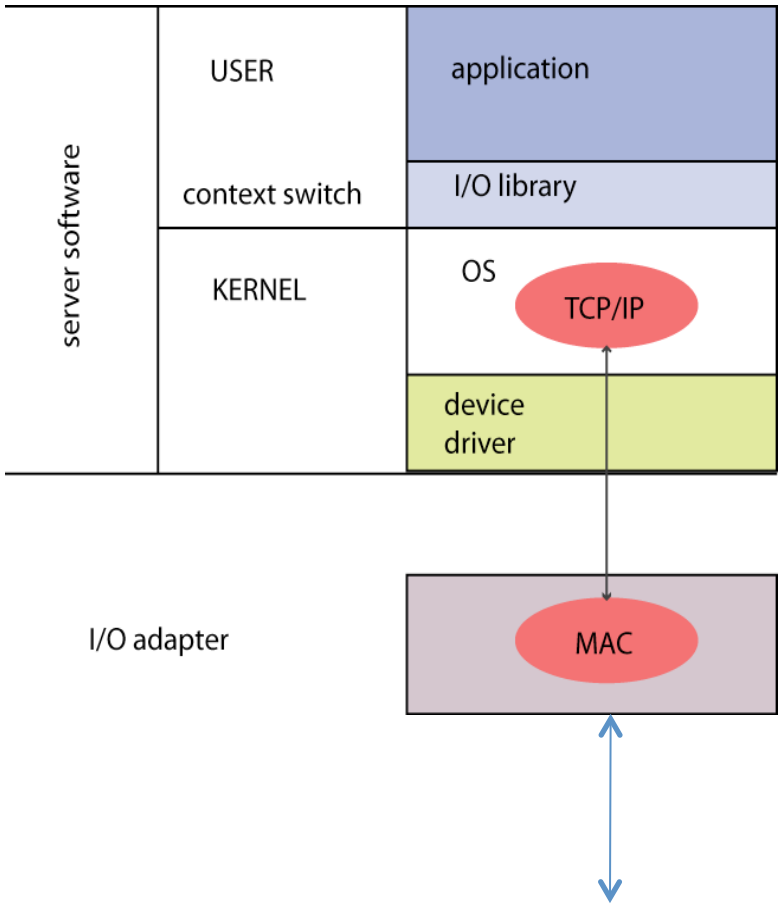
- ✓Atca ethernet switches are readily available
- ✓Mixed operation of 1Ge and 10Ge in the same card
  - ✓Useful in aggregation : 10 x1Ge -> 1x 10Ge
  - ✓Typical in large EVBs



ROM Block Diagram



Legacy TCP

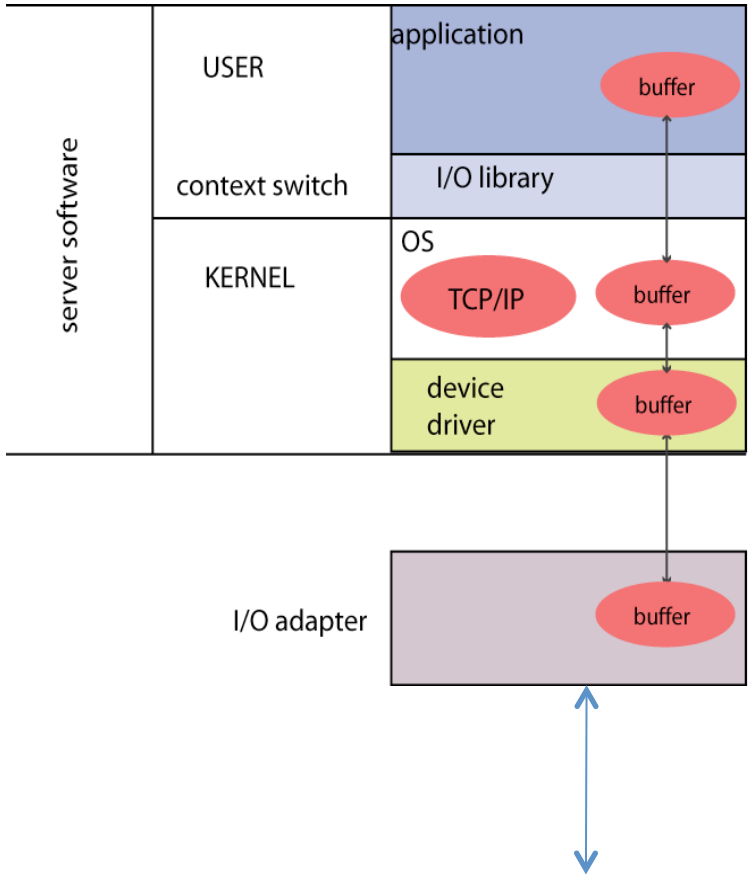


10Gethernet line rate is obtained (through TCP) with a ~10GHz CPU @ 100% load

Ethernet switch

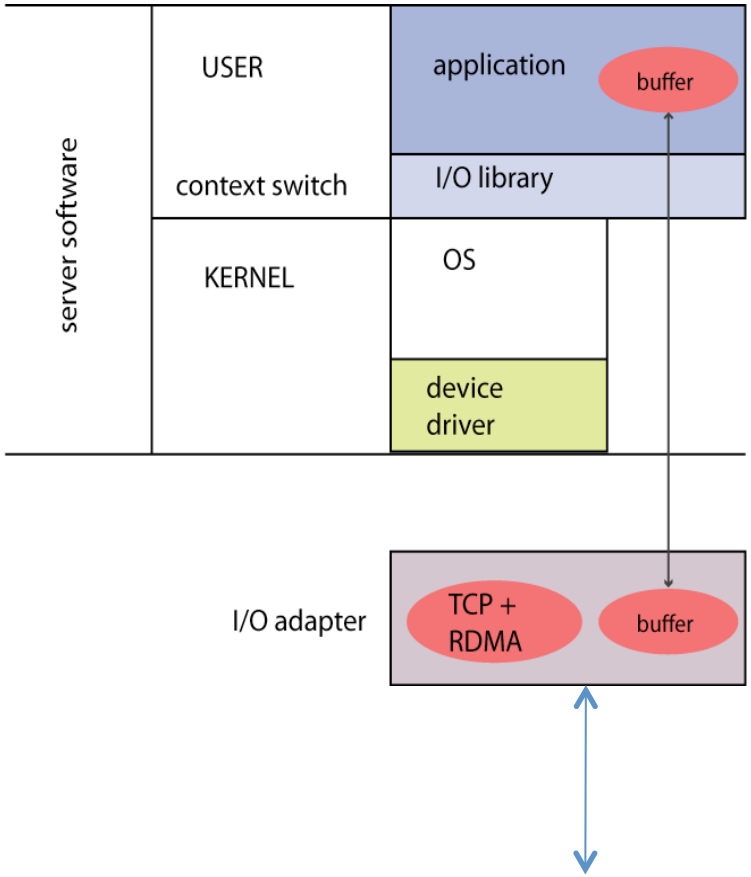
TCP + RDMA

Standard implementation



Ethernet switch

TCP Offload



Ethernet switch

Why TCP ?

✓Reliable (flow control)

✓Congestion avoidance

✓Switchable

✓Scalable and low cost

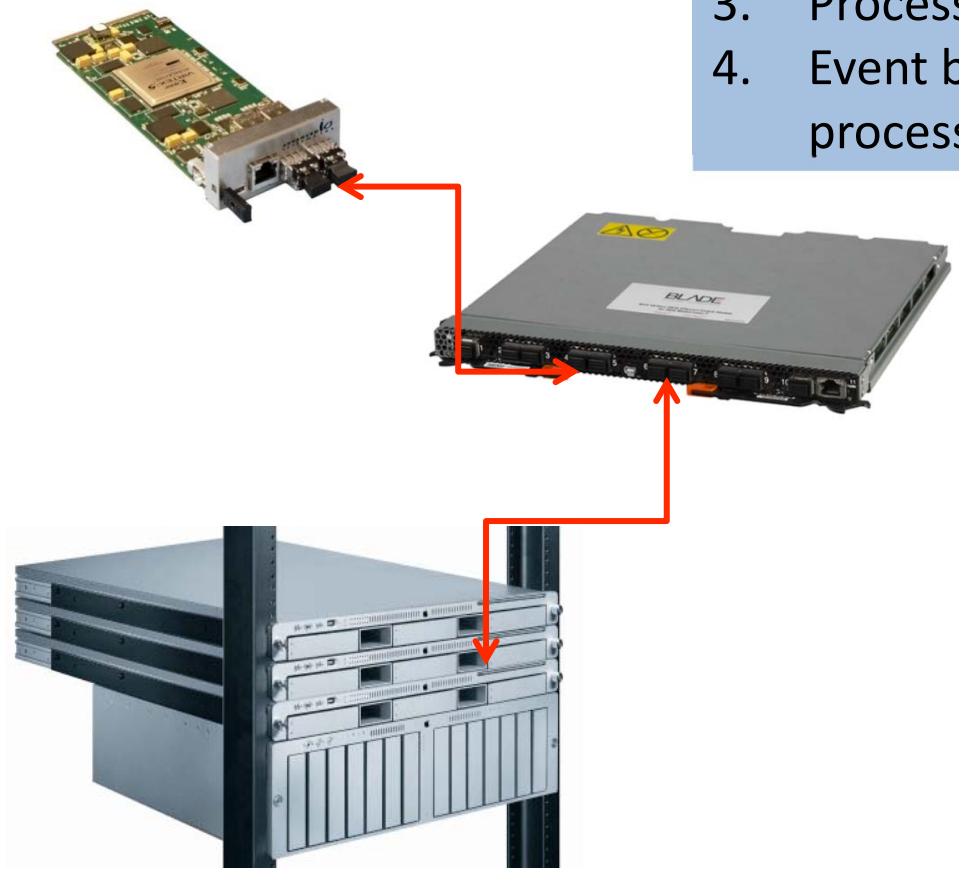
✓Efficient (not when driven by software)

✓Transport layer for RDMA applications

✓R&D will be devoted to build a TCP Offload Engine (TOE) with RDMA support

Hardware Setup

1. FPGA with 10GE MAC + TOE
2. 10 GE switch
3. Processor farm with 10GE adpaters
4. Event builder based on RDMA on processor farm



Conclusions

- Proposal for a Readout board based on a modular approach
  - Adherence to standards as much as possible
- R&D on readout protocol based on RDMA and offloaded TCP