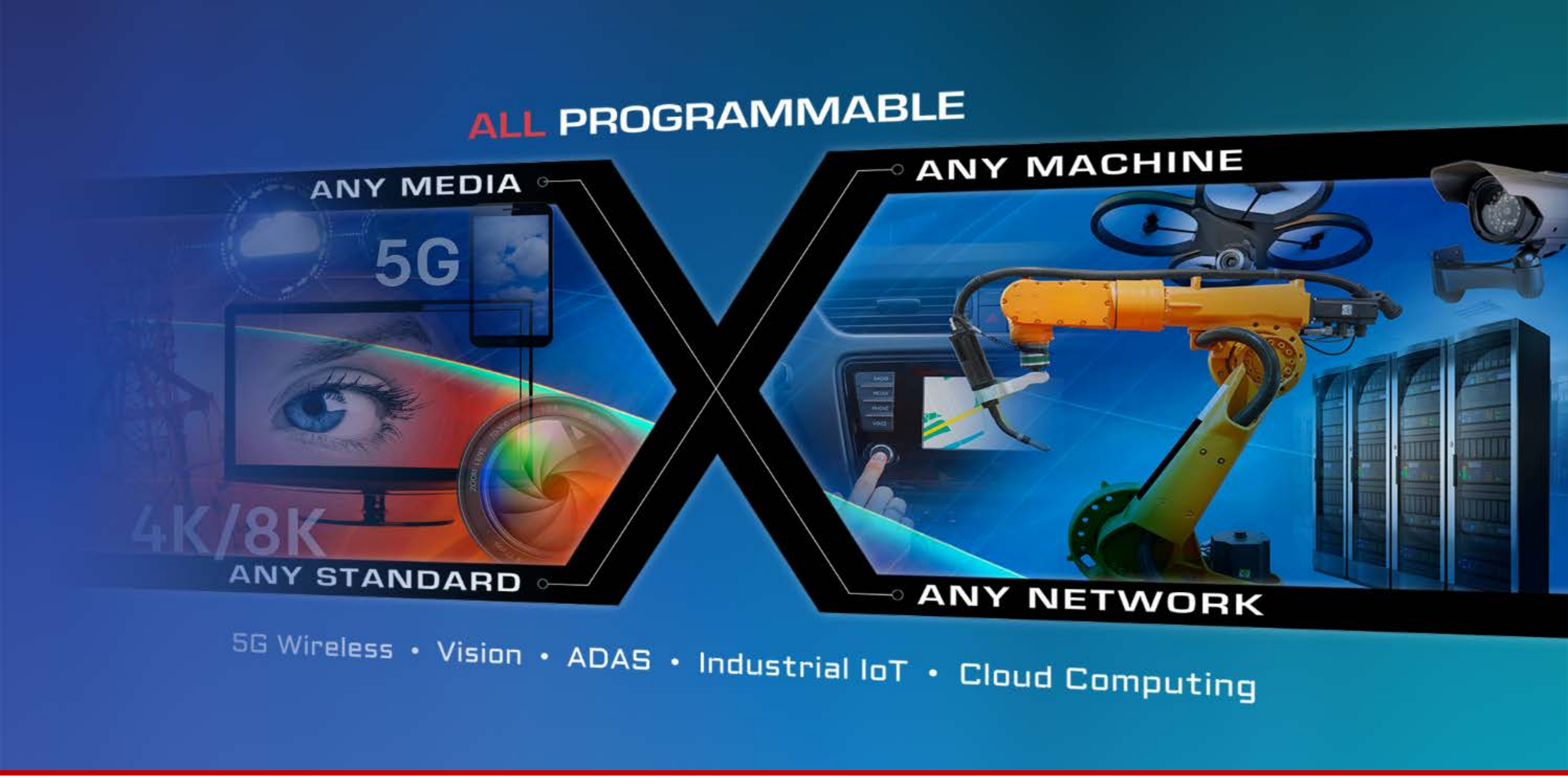


XILINX
ALL PROGRAMMABLE.

mle
missing link electronics

Heterogeneous Multi-Processing for SW-Defined Multi-Tiered Storage Architectures

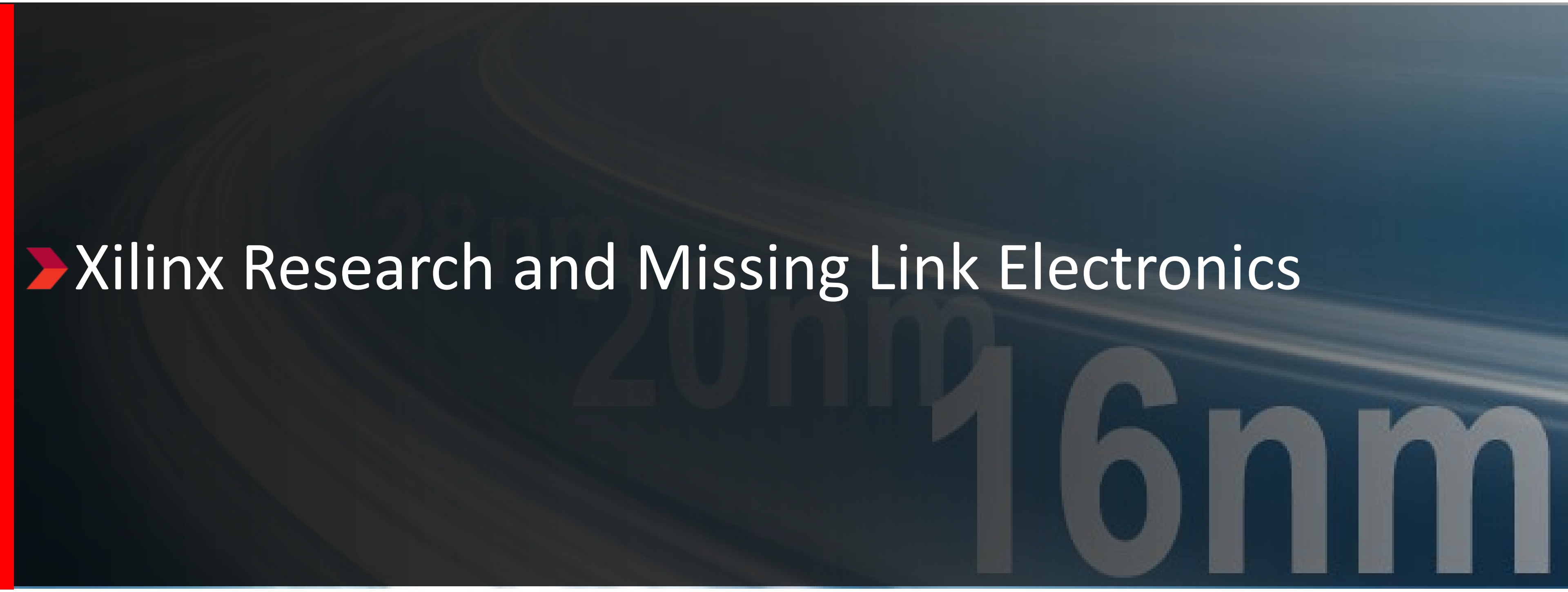
Endric Schubert (MLE)
Ulrich Langenbach (MLE)
Michaela Blott (Xilinx Research)

SDC, 2017

Content

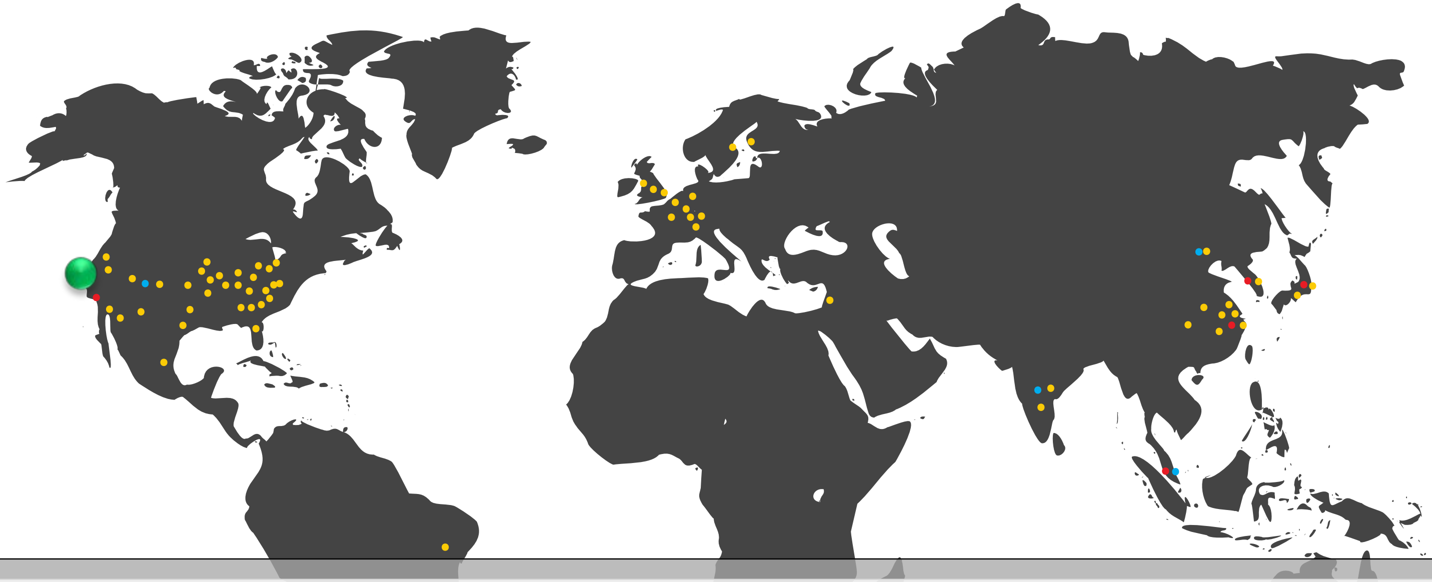
Heterogeneous Multi-Processing for
Software-Defined Multi-Tiered Storage Architectures

- Who – Xilinx Research and Missing Link Electronics
- Why – Multi-tiered storage needs predictable performance scalability, deterministic low-latency and cost-efficient flexibility / programmability
- What – Tera-OPS processing performance in a single-chip heterogeneous compute solution running Linux
- How – Combine “unconventional” dataflow architectures for acceleration & offloading with Dynamic Partial Reconfiguration and High-Level Synthesis



Xilinx Research and Missing Link Electronics

Xilinx – The All Programmable Company



XILINX
ALL PROGRAMMABLE™

XILINX - Founded 1984

🟢 Headquarters	🟡 Sales and Support
🔵 Research and Development	🔴 Manufacturing

$2.38B FY15 revenue

>55% market segment share

3,500+ employees worldwide

20,000 customers worldwide

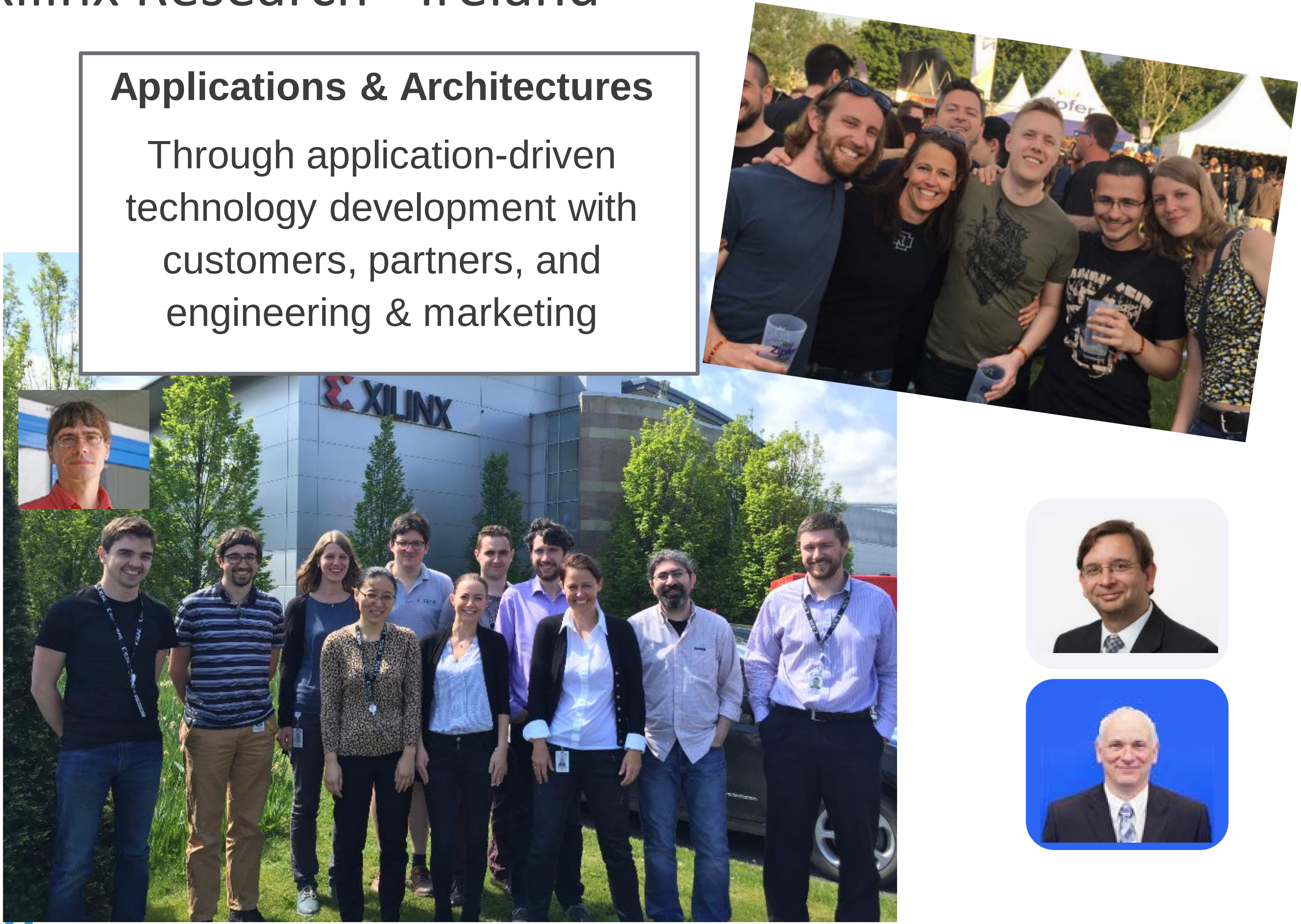
3,500+ patents

60 industry firsts

Xilinx Research - Ireland

Applications & Architectures

Through application-driven technology development with customers, partners, and engineering & marketing



Missing Link Electronics

Xilinx Ecosystem Partner

Vision: The convergence of software and off-the-shelf programmable logic opens-up more economic system realizations with predictable scalability!

Mission: To de-risk the adoption of heterogeneous compute technology by providing pre-validated IP and expert design services.

Certified Xilinx Alliance Partner since 2011, Preferred Xilinx PetaLinux Design Service Partner since 2013.

Missing Link Electronics Products & Services

TCP/IP & UDP/IP Network Protocol Accelerators at 10/25/50 GigE line-rate.

Patented Mixed Signal systems solutions with integrated Delta-Sigma converters in FPGA logic.

SATA Storage Extension for Xilinx Zynq All-Programmable Systems-on-Chip.

Low-Latency Ethernet MAC form German Fraunhofer HHI.

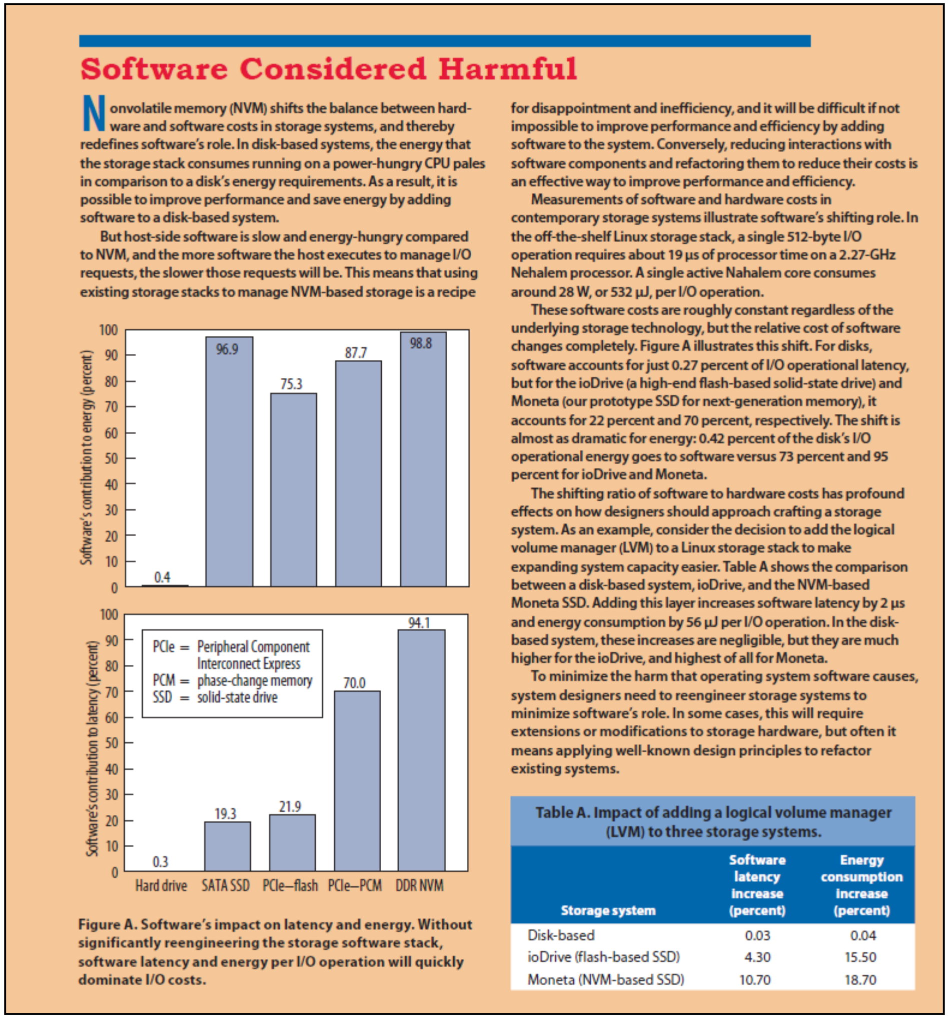
Key-Value-Store Accelerator for hybrid SSD/HDD memcached and object storage.

A team of FPGA and Linux engineers to support our customer's technology projects in the USA and Europe.

Motivation

Technology Forces in Storage

- **Software significantly impacts latency and energy efficiency in systems with nonvolatile memory**
- **However, software-defined flexibility is necessary to fully utilize novel storage technologies**
- **Hyper-capacity hyper-converged storage systems need more performance, but within cost and energy envelopes**



Software Considered Harmful

Nonvolatile memory (NVM) shifts the balance between hardware and software costs in storage systems, and thereby redefines software's role. In disk-based systems, the energy that the storage stack consumes running on a power-hungry CPU pales in comparison to a disk's energy requirements. As a result, it is possible to improve performance and save energy by adding software to a disk-based system.

But host-side software is slow and energy-hungry compared to NVM, and the more software the host executes to manage I/O requests, the slower those requests will be. This means that using existing storage stacks to manage NVM-based storage is a recipe for disappointment and inefficiency, and it will be difficult if not impossible to improve performance and efficiency by adding software to the system. Conversely, reducing interactions with software components and refactoring them to reduce their costs is an effective way to improve performance and efficiency.

Measurements of software and hardware costs in contemporary storage systems illustrate software's shifting role. In the off-the-shelf Linux storage stack, a single 512-byte I/O operation requires about 19 µs of processor time on a 2.27-GHz Nehalem processor. A single active Nahalem core consumes around 28 W, or 532 µJ, per I/O operation.

These software costs are roughly constant regardless of the underlying storage technology, but the relative cost of software changes completely. Figure A illustrates this shift. For disks, software accounts for just 0.27 percent of I/O operational latency, but for the ioDrive (a high-end flash-based solid-state drive) and Moneta (our prototype SSD for next-generation memory), it accounts for 22 percent and 70 percent, respectively. The shift is almost as dramatic for energy: 0.42 percent of the disk's I/O operational energy goes to software versus 73 percent and 95 percent for ioDrive and Moneta.

The shifting ratio of software to hardware costs has profound effects on how designers should approach crafting a storage system. As an example, consider the decision to add the logical volume manager (LVM) to a Linux storage stack to make expanding system capacity easier. Table A shows the comparison between a disk-based system, ioDrive, and the NVM-based Moneta SSD. Adding this layer increases software latency by 2 µs and energy consumption by 56 µJ per I/O operation. In the disk-based system, these increases are negligible, but they are much higher for the ioDrive, and highest of all for Moneta.

To minimize the harm that operating system software causes, system designers need to reengineer storage systems to minimize software's role. In some cases, this will require extensions or modifications to storage hardware, but often it means applying well-known design principles to refactor existing systems.

Software's contribution to energy (percent)
100
90
80
70
60
50
40
30
20
10
0
0.4
96.9
75.3
87.7
98.8
Software's contribution to latency (percent)
PCIe = Peripheral Component Interconnect Express
PCM = phase-change memory
SSD = solid-state drive
0.3
19.3
21.9
70.0
94.1
Hard drive
SATA SSD
PCIe—flash
PCIe—PCM
DDR NVM

Figure A. Software's impact on latency and energy. Without significantly reengineering the storage software stack, software latency and energy per I/O operation will quickly dominate I/O costs.

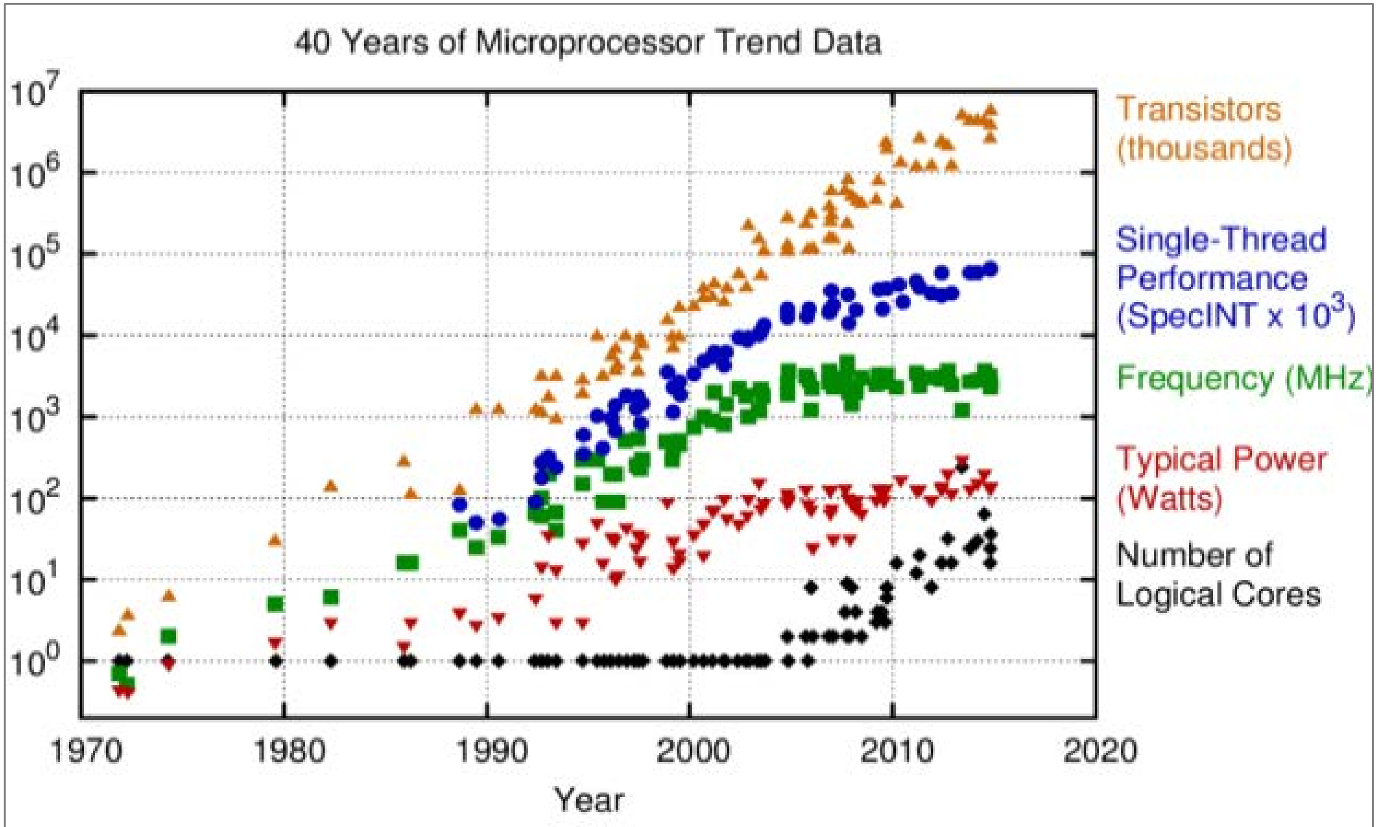
Table A. Impact of adding a logical volume manager (LVM) to three storage systems.

Storage system	Software latency increase (percent)	Energy consumption increase (percent)
Disk-based	0.03	0.04
ioDrive (flash-based SSD)	4.30	15.50
Moneta (NVM-based SSD)	10.70	18.70

Source: Steven Swanson and Adrian M. Caulfield, UCSD
IEEE Computer, August 2013

The Von Neumann Bottleneck [J. Backus, 1977]

- **CPU system performance scalability is limited**



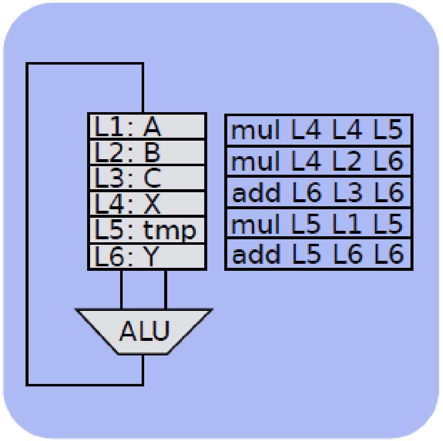
40 Years of Microprocessor Trend Data

Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2015 by K. Rupp

New Compute Architectures are needed

Spatial vs. Temporal Computing

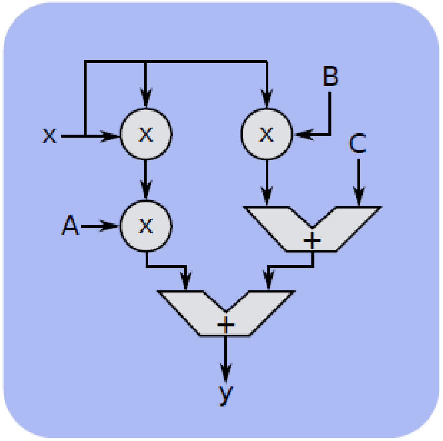
Sequential Processing
with CPU



L1: A
L2: B
L3: C
L4: X
L5: tmp
L6: Y

```
mul L4 L4 L5
mul L4 L2 L6
add L6 L3 L6
mul L5 L1 L5
add L5 L6 L6
```

Parallel Processing
with Logic Gates

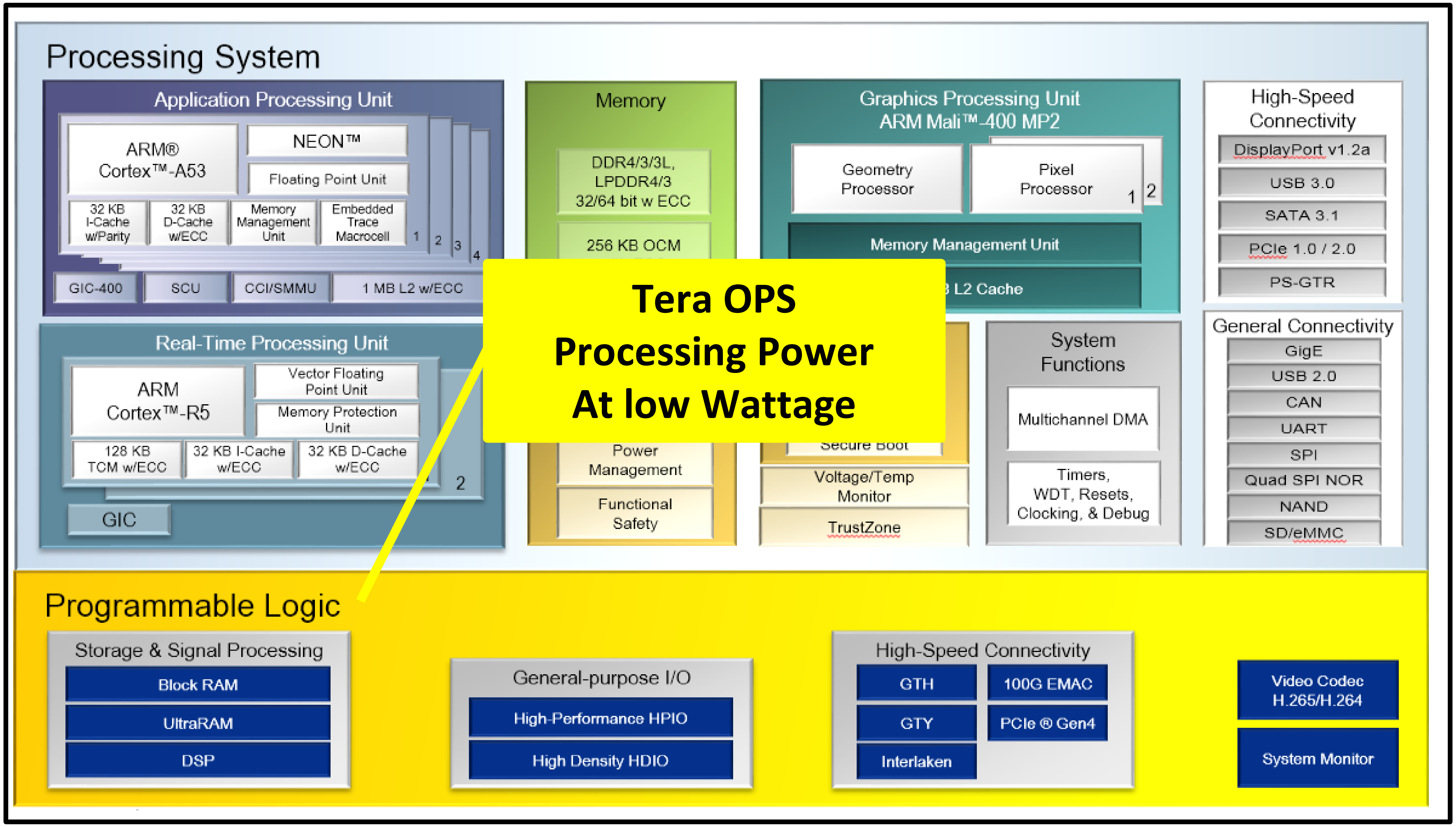


Source: Dr. Andre DeHon, Upenn: “Spatial vs. Temporal Computing”

- **CPU system performance scalability is limited**
- **Spatial computing offers further scaling opportunity**

New Compute Architectures are needed to take advantage of this

Architectural Choices for Storage Devices

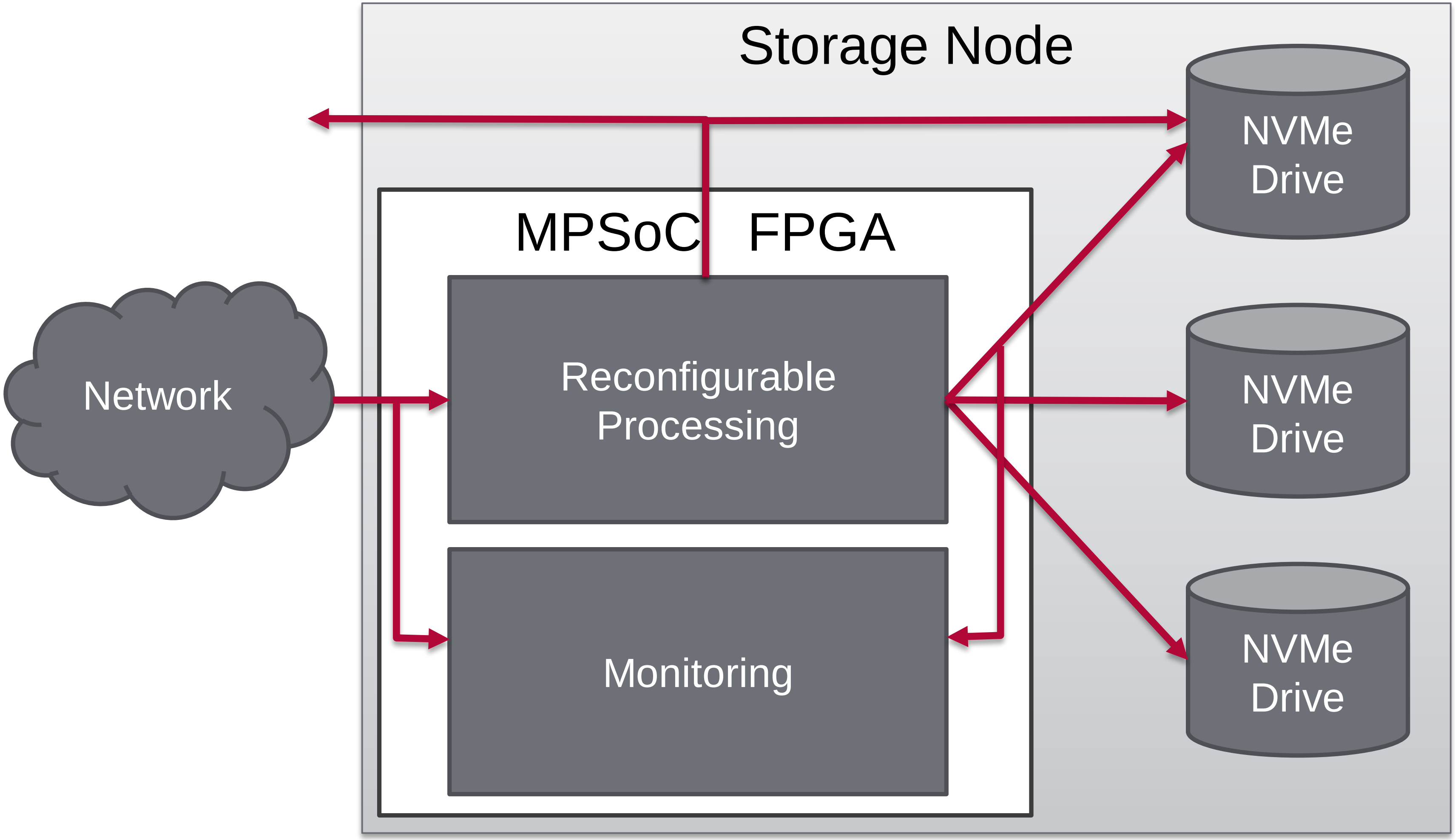


Tera OPS Processing Power At low Wattage

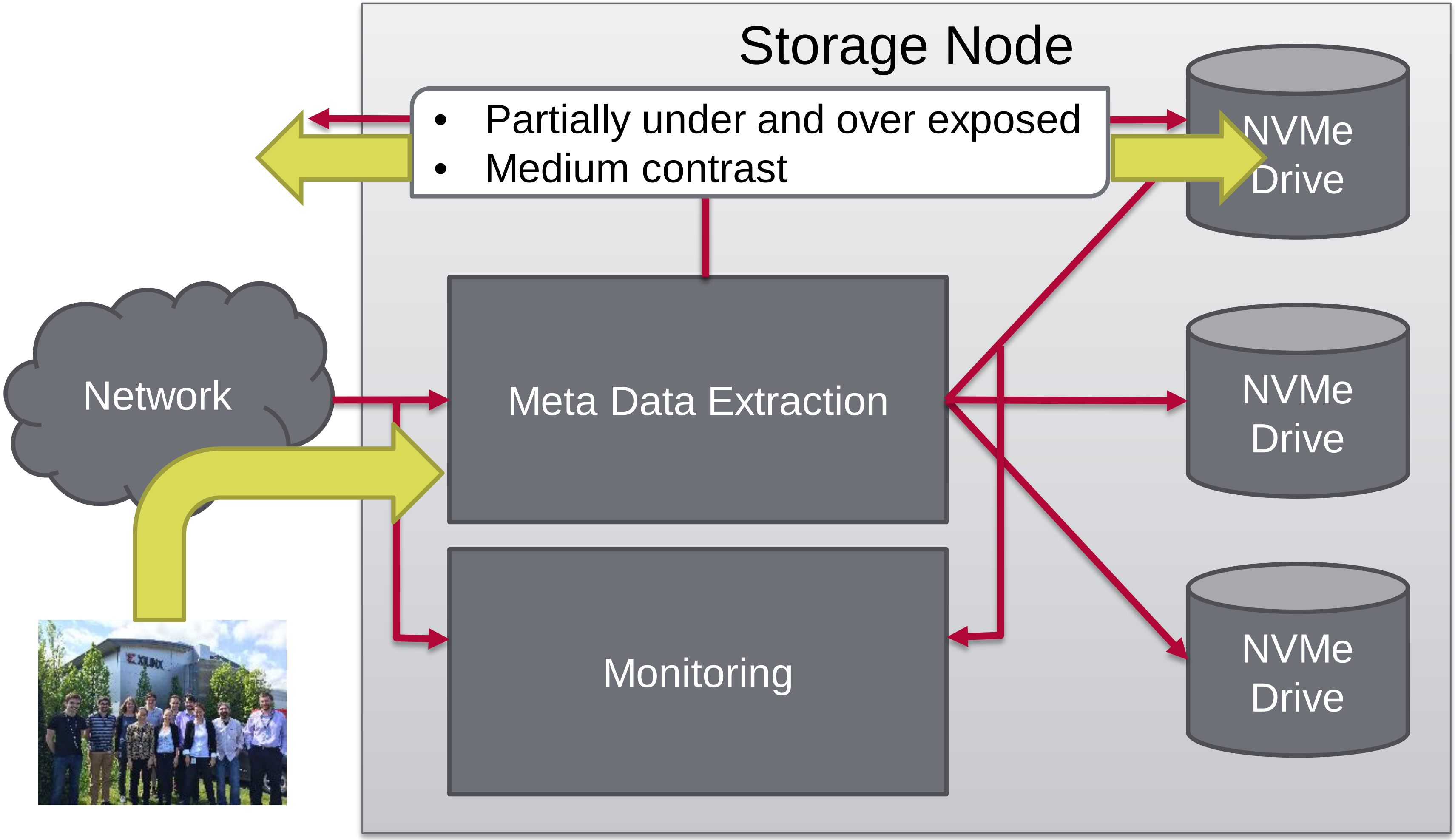
Source: T.Noll, RWTH Aachen

Use Case: Image/ Video Storage

A Flexible All Programmable Storage Node

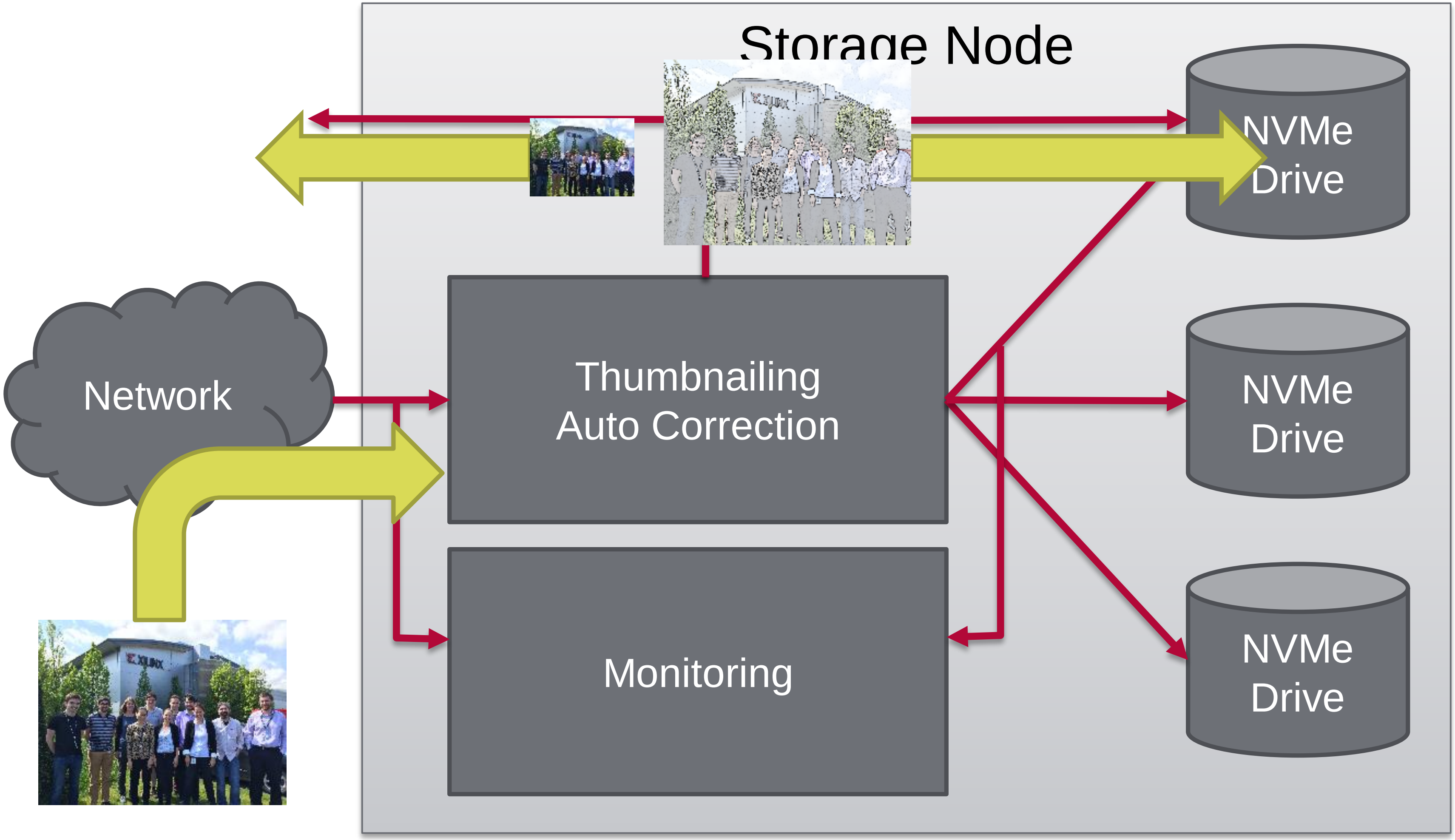


Meta Data Extraction, e.g. Image Quality Metrics

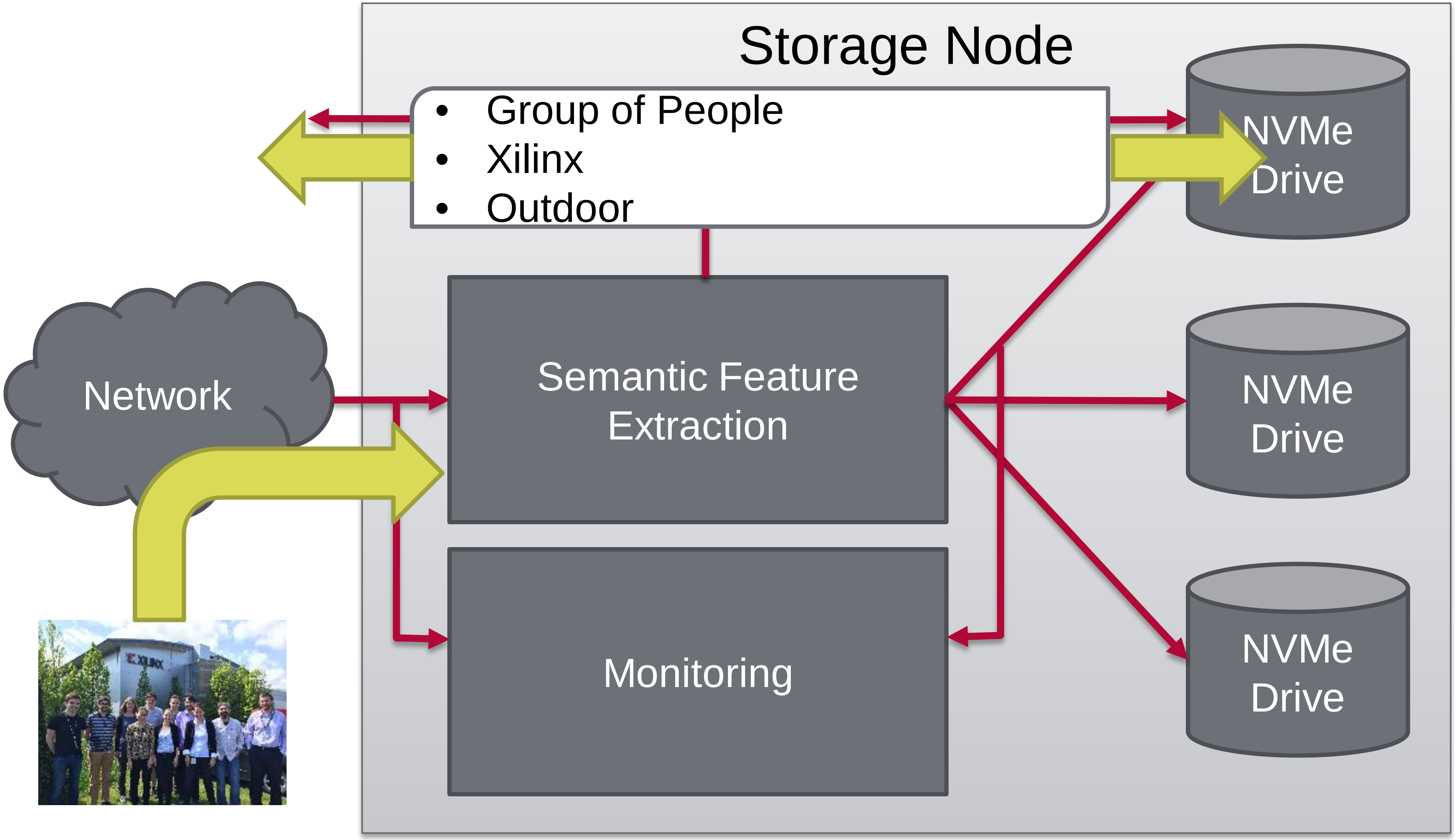


- Partially under and over exposed
- Medium contrast

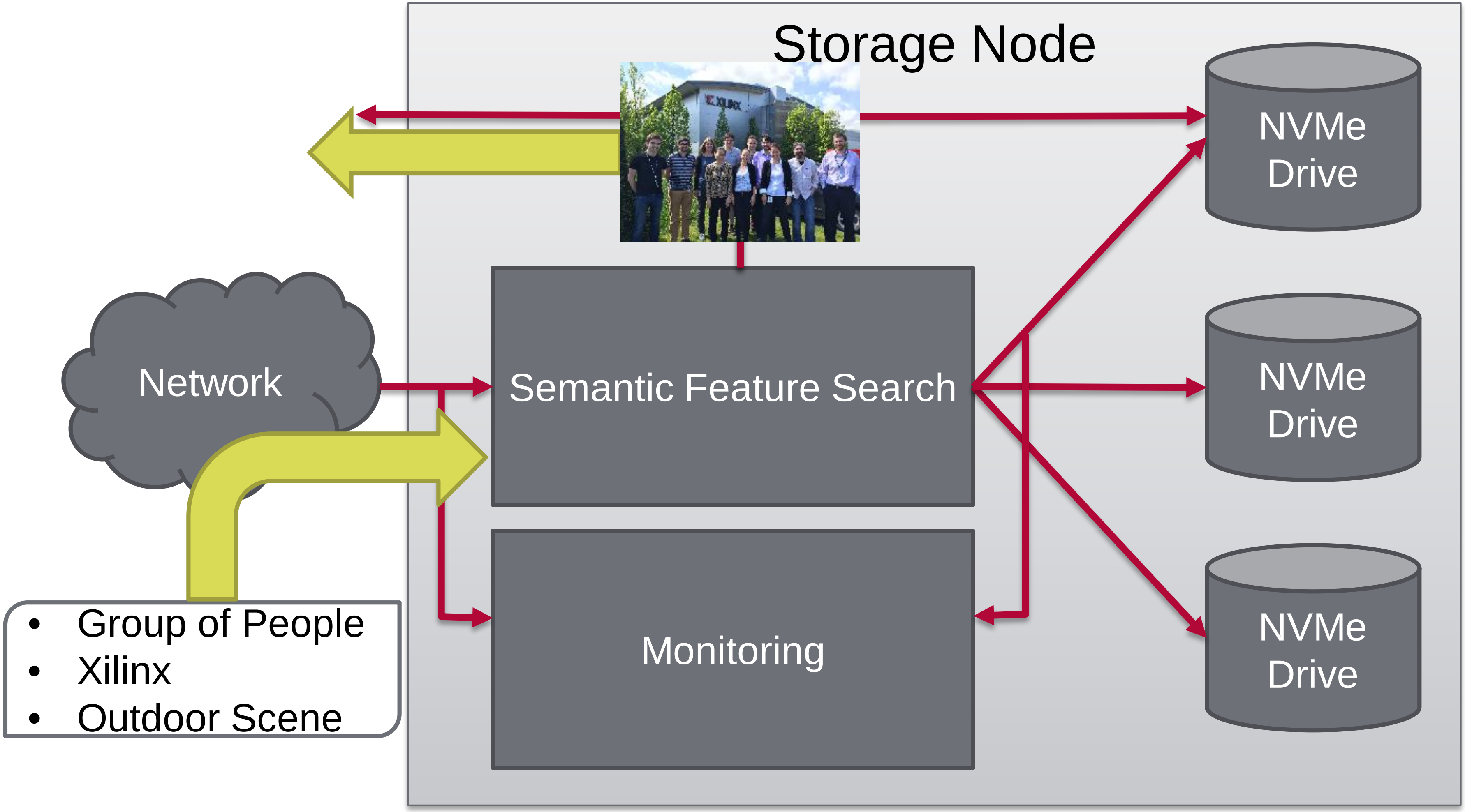
Processing, e.g. Thumbnailing, Auto-Correction



Semantic Feature Extraction, e.g. Classification



Semantic Search Support



Storage Node

Network

Semantic Feature Search

Monitoring

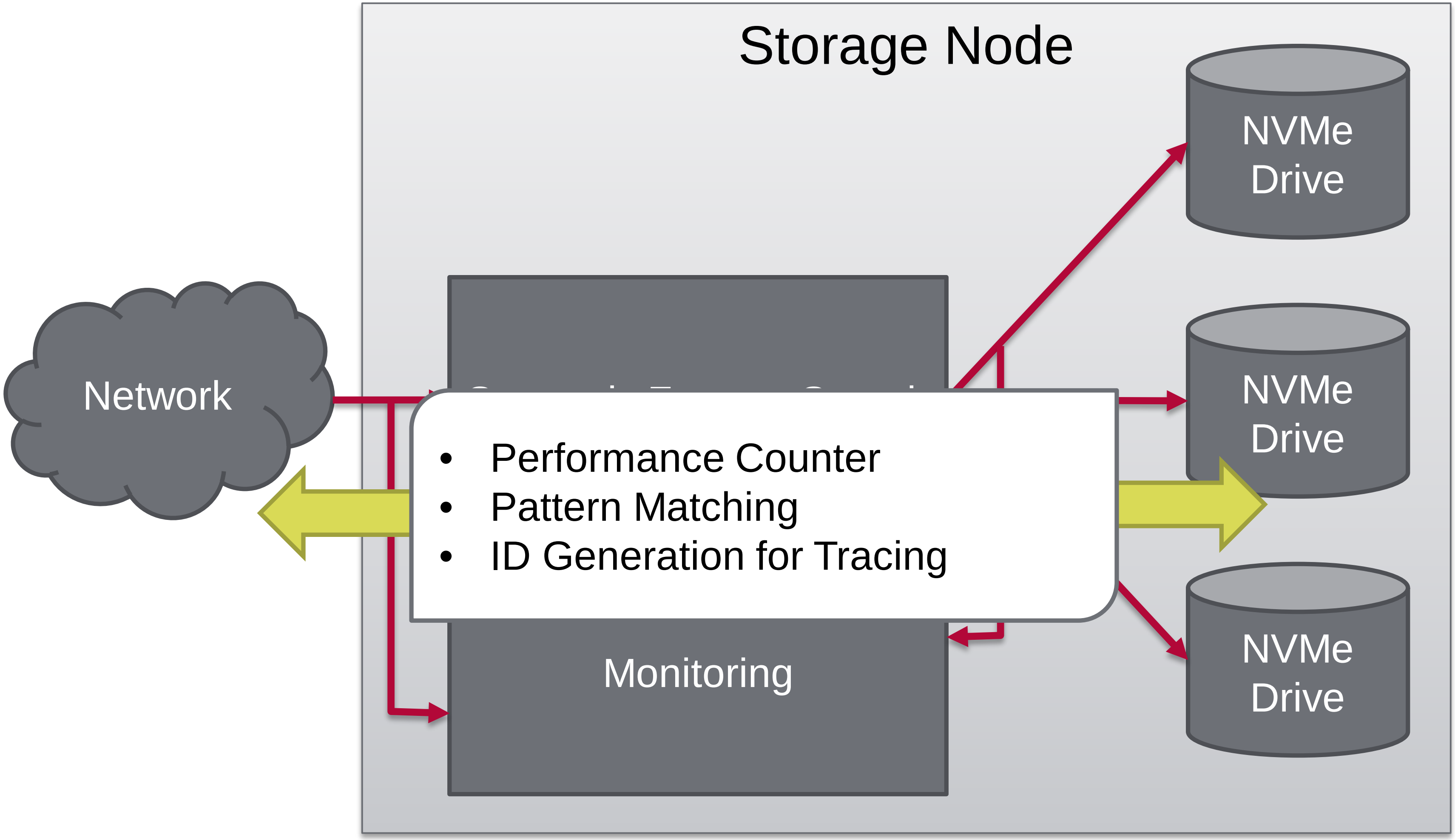
NVMe Drive

NVMe Drive

NVMe Drive

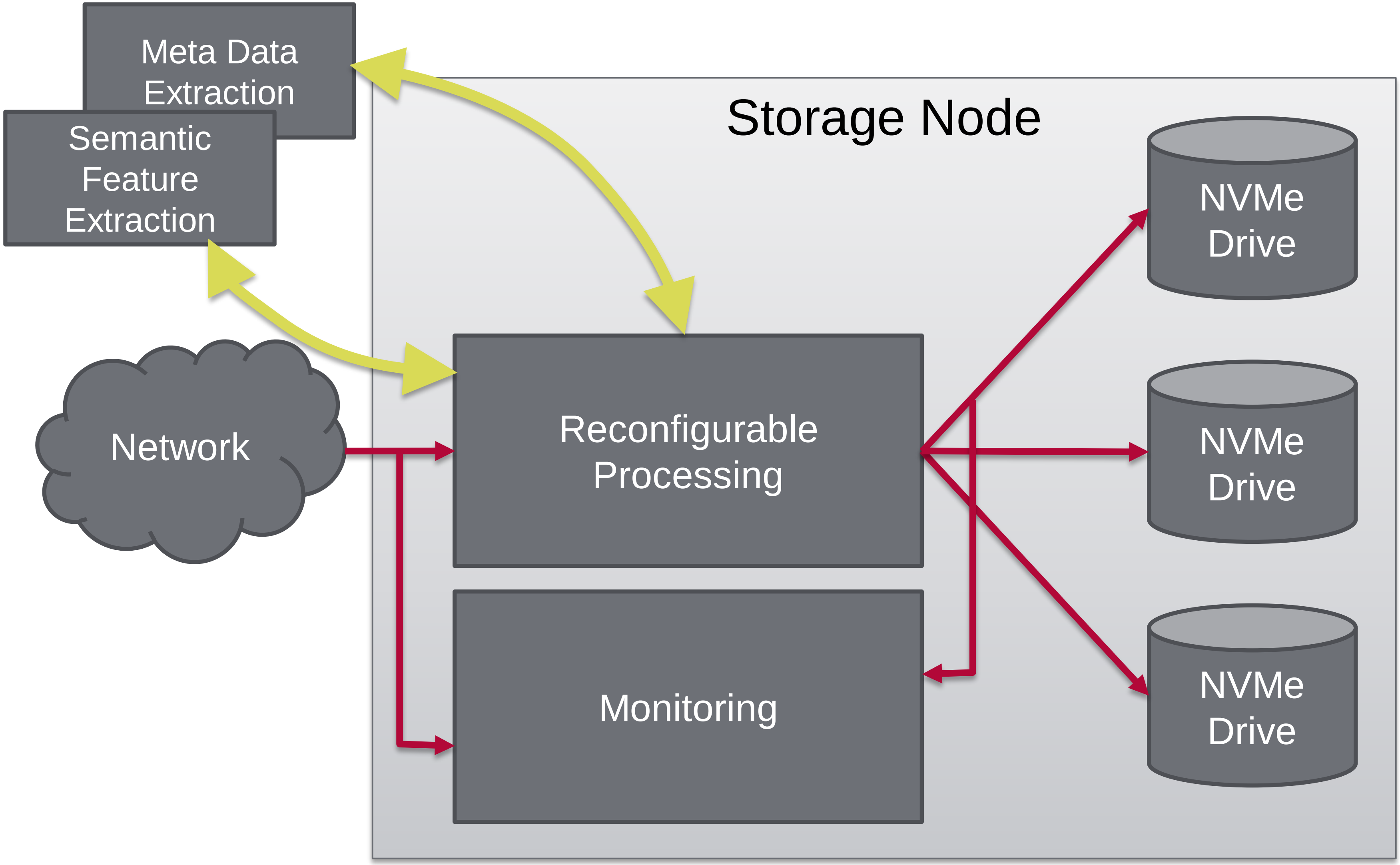
- Group of People
- Xilinx
- Outdoor Scene

Performance Metrics, e.g. Bandwidth, Latency



- Performance Counter
- Pattern Matching
- ID Generation for Tracing

Runtime Programmability

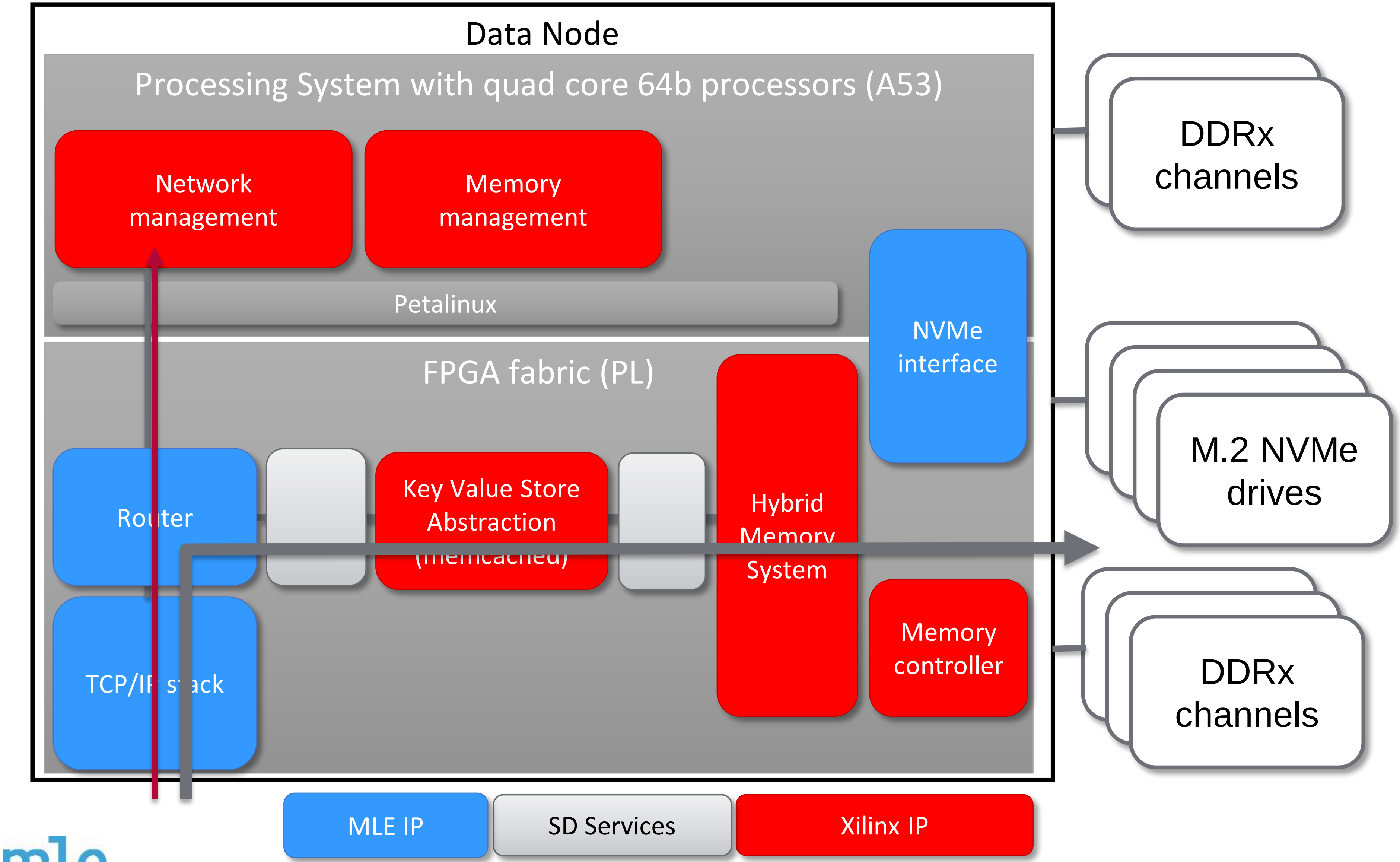


Architectural Concepts

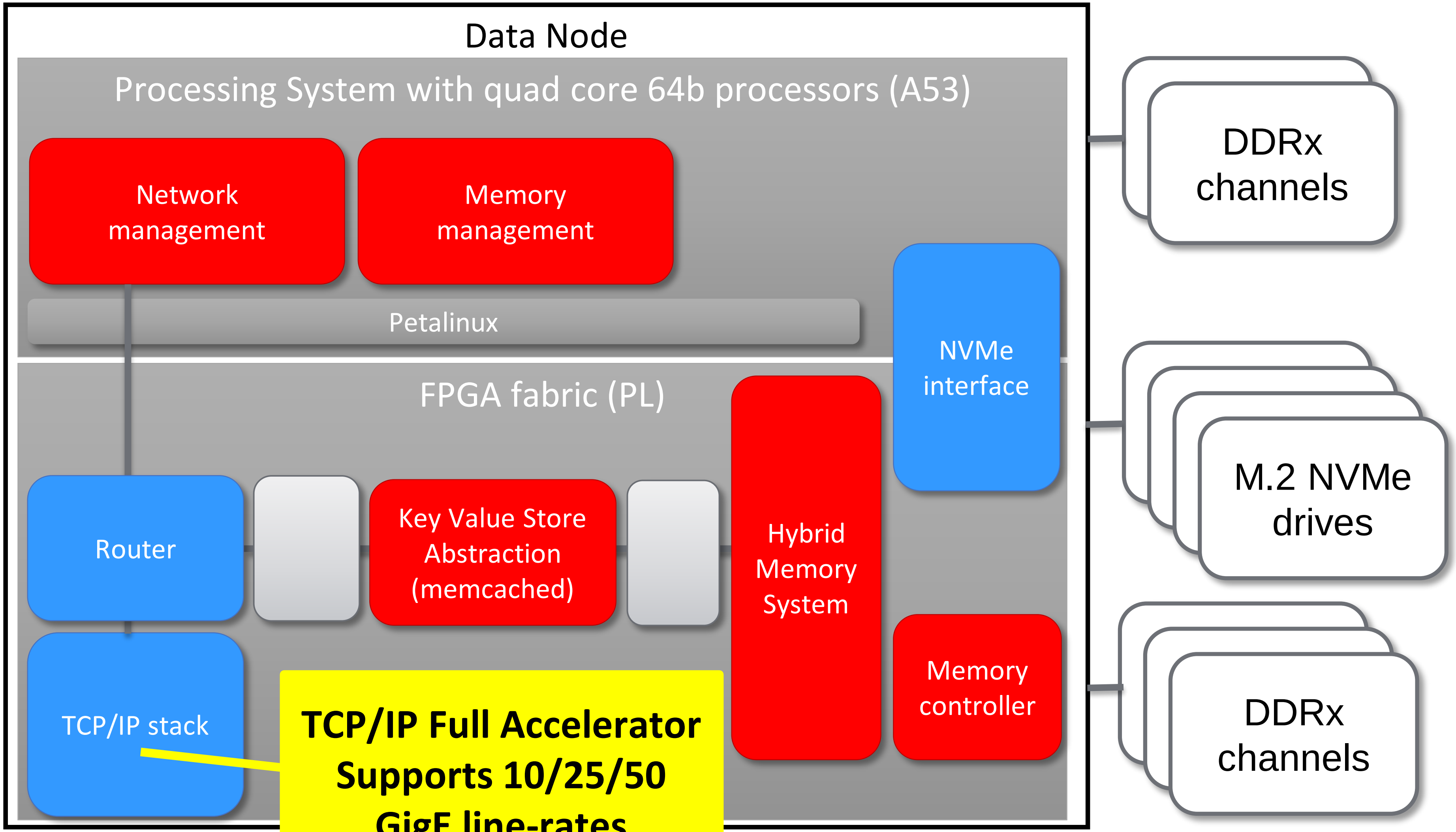
Key Concepts Presented at SDC-2016

- Heterogeneous compute device as a single -chip solution
- Direct network interface with full accelerator for protocols
- Performance scaling with dataflow architectures
- Scaling capacity and cost with a Hybrid Storage subsystem
- Software-defined services

SDC-2016: Single-Chip Solution for Storage

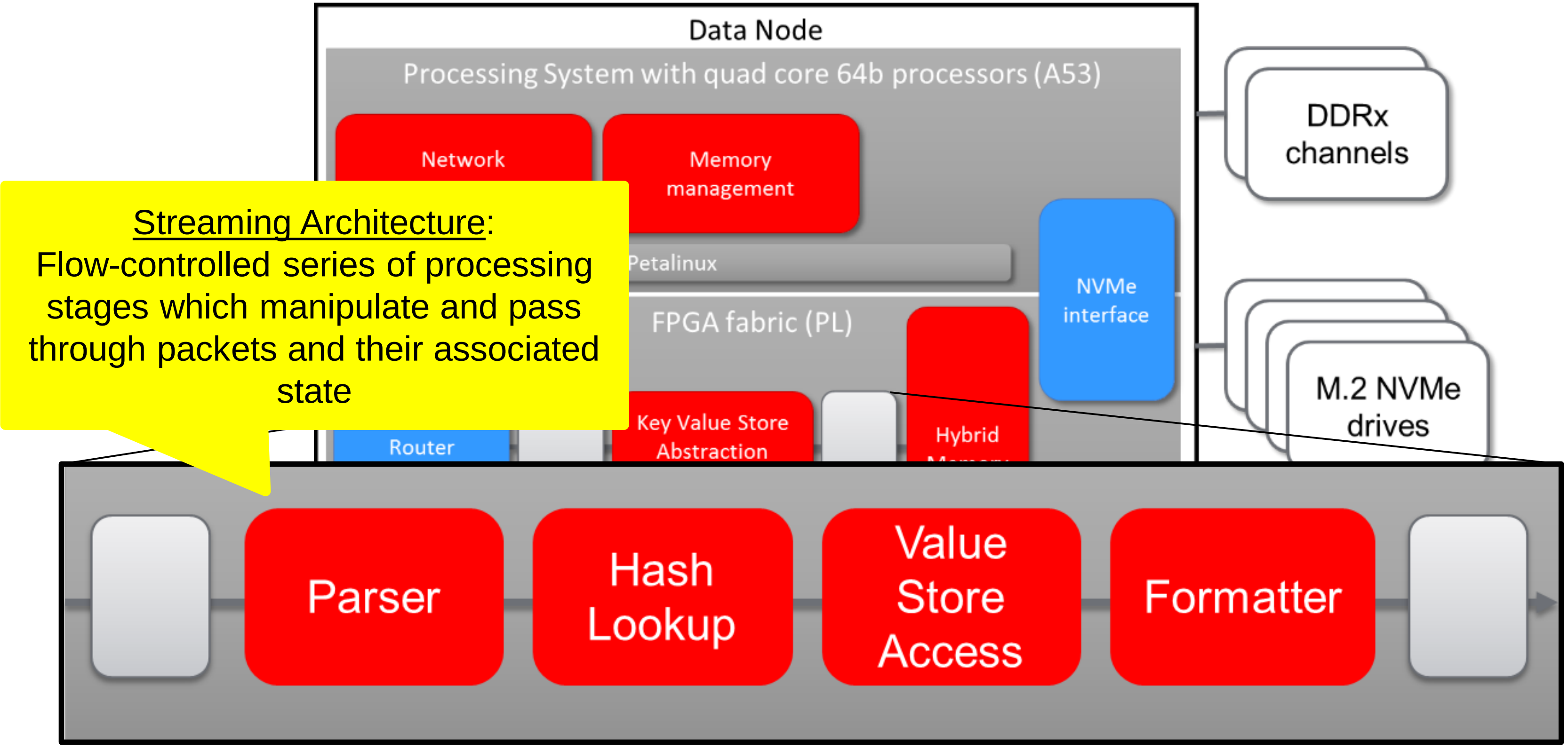


SDC-2016: Hardware Accelerated Network Stack



TCP/IP Full Accelerator Supports 10/25/50 GigE line-rates

SDC-2016: Dataflow architectures for performance scaling



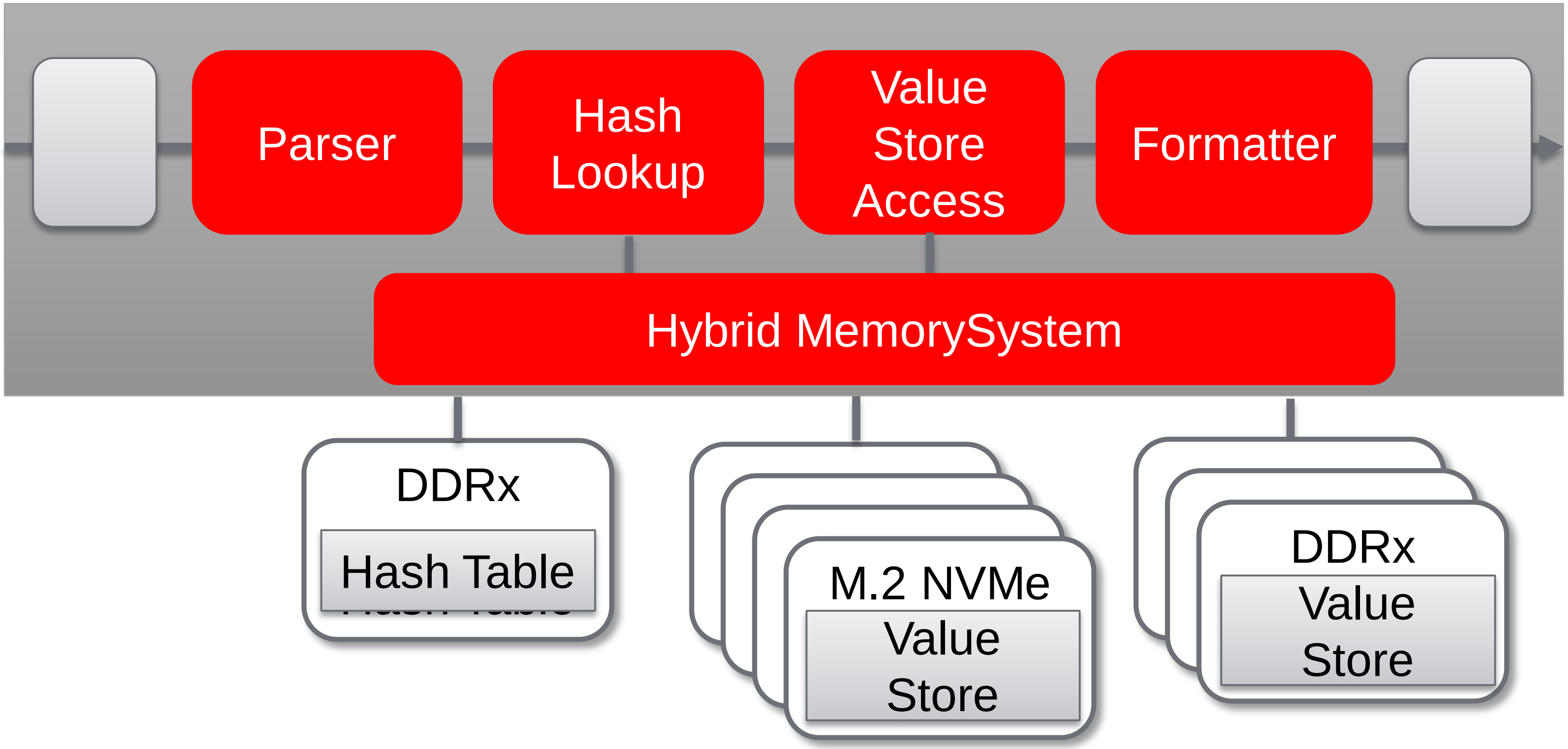
Streaming Architecture: Flow-controlled series of processing stages which manipulate and pass through packets and their associated state

- **Now: 10 Gbps demonstrated with a 64b data path @ 156MHz using 20% of FPGA**
- **Next: 100 Gbps can be achieved by using a 512b @ 200MHz pipeline for example**

Source: Blott et al: Achieving 10Gbps line-rate key-value stores with FPGAs; HotCloud 2013

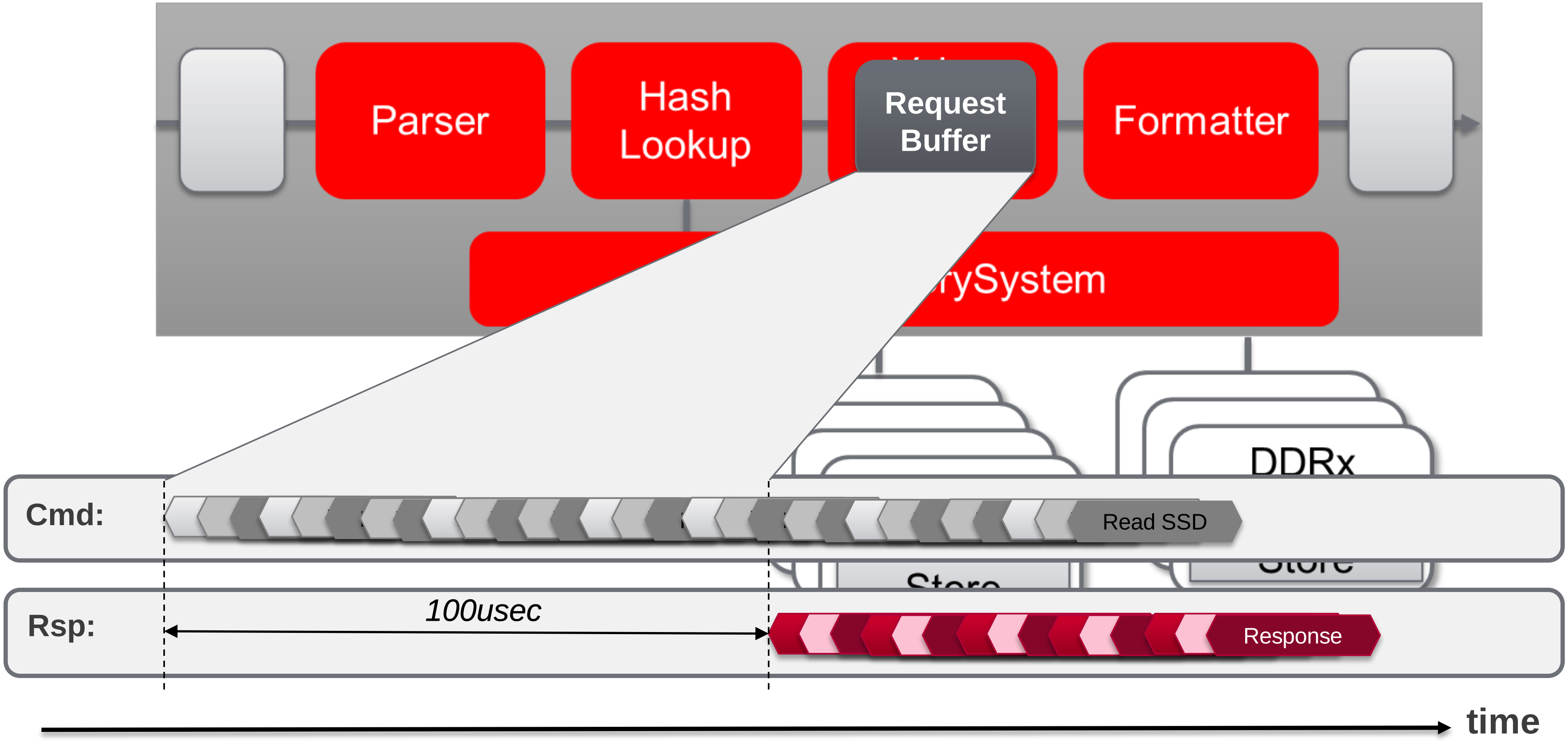
SDC-2016: Scaling Capacity via hybrids

- SSDs combined with DDRx channels can be used to build high capacity & high performance object stores
- Concepts and early prototype to scale to 40TB & 80Gbps key value stores



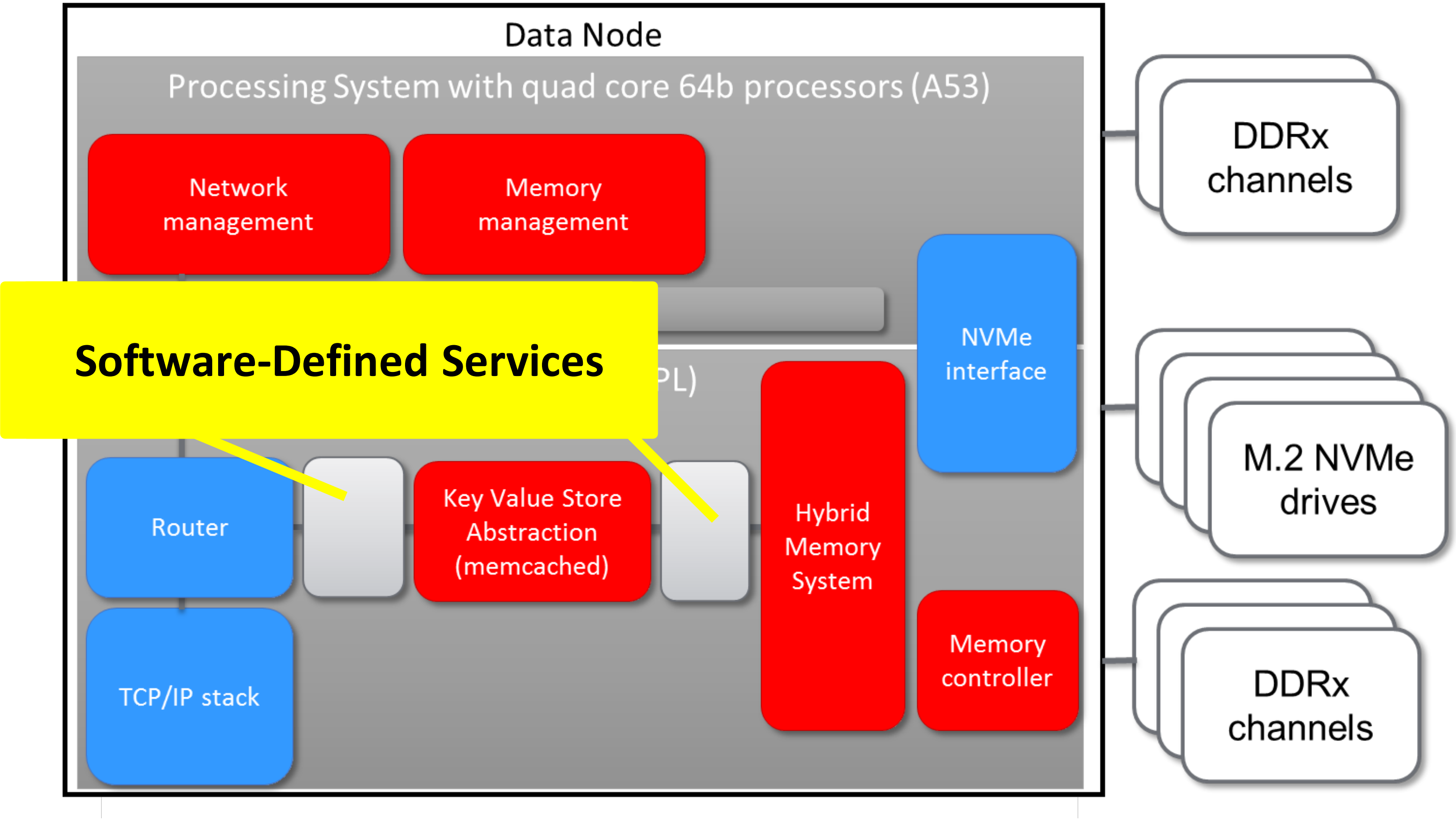
Source: HotStorage 2015, Scaling out to a Single-Node 80Gbps Memcached Server with 40Terabytes of Memory

SDC-2016: Handling High Latency Accesses without Sacrificing Throughput



- **Dataflow architectures: no limit to number of outstanding requests**
- **Flash can be serviced at maximum speed**

Software-Defined Services



- **Spatial computing of additional services at no performance cost until resource limitations are reached**

Software-Defined Services

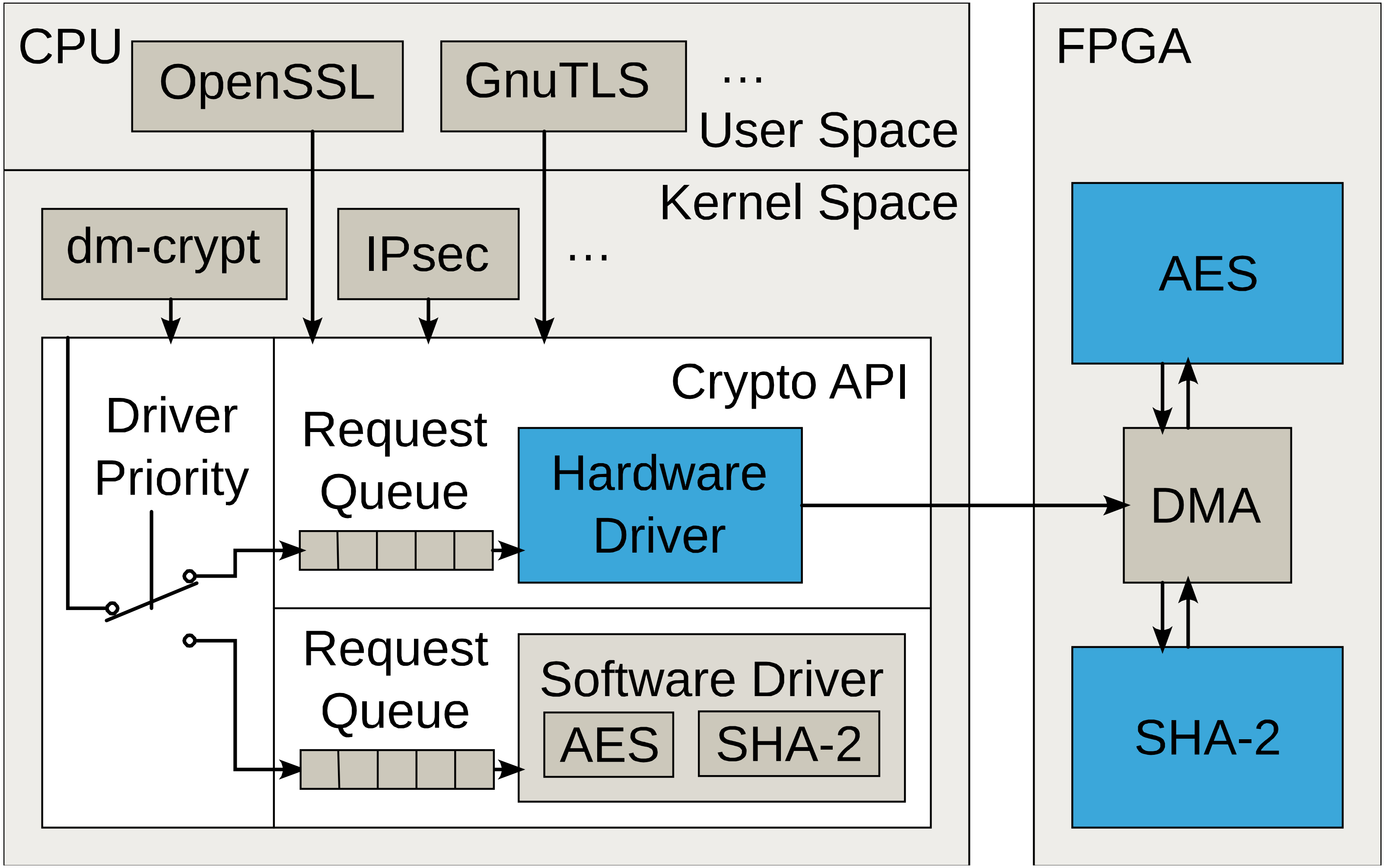
Software-Defined Services – Proof-of-Concepts

- **Offload engines for Linux Kernel Crypto-API**
- **Non-intrusive latency analysis via PCIe TLP “Tracers”**
- **Inline processing with Deep Convolutional Neural Networks**
- **Declarative Linux Kernel Support Partial Reconfiguration**

Software-Defined Services - Example 1) Accelerating the Linux Kernel Crypto-API

- **Crypto-API is a cryptography framework in the Linux kernel used for encryption, decryption, compression, de-compression, etc.**
- **Needs acceleration to support processing at higher line-rates (100 GigE).**
- **Open Source software implementation that follows a streaming dataflow processing architecture**
  - Hardware Interface: AXI Streaming
  - Software/ Hardware Interface: SG-DMA in, SG-DMA out
- **High-Level Synthesis generated accelerator blocks from reference C code**

System Architecture of Crypto-API Accelerator

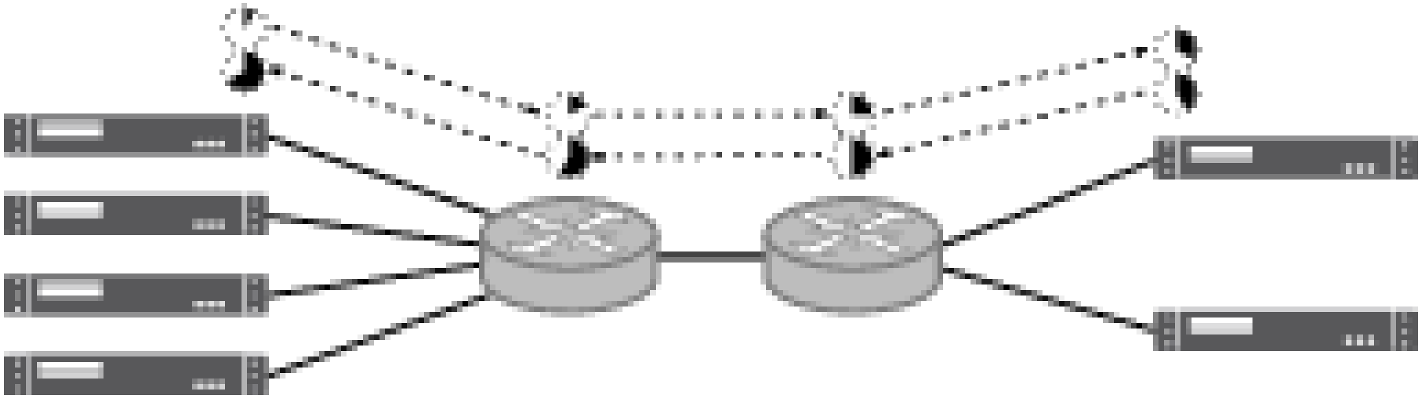


Software-Defined Services - Example 2) Non-Intrusive Latency Analysis via PCIe TLP Tracers

- **Performance analysis and ongoing monitoring of bandwidth and latency in distributed systems is difficult.**
  - – Round-trip times
  - – Time-outs
  - – Throttling
- **When done in software, results get distorted by additional compute burden.**
- **When done in Programmable Logic, it can be (clock cycle) accurate and non-intrusive via adding so-called "Tracers" into the dataflow.**

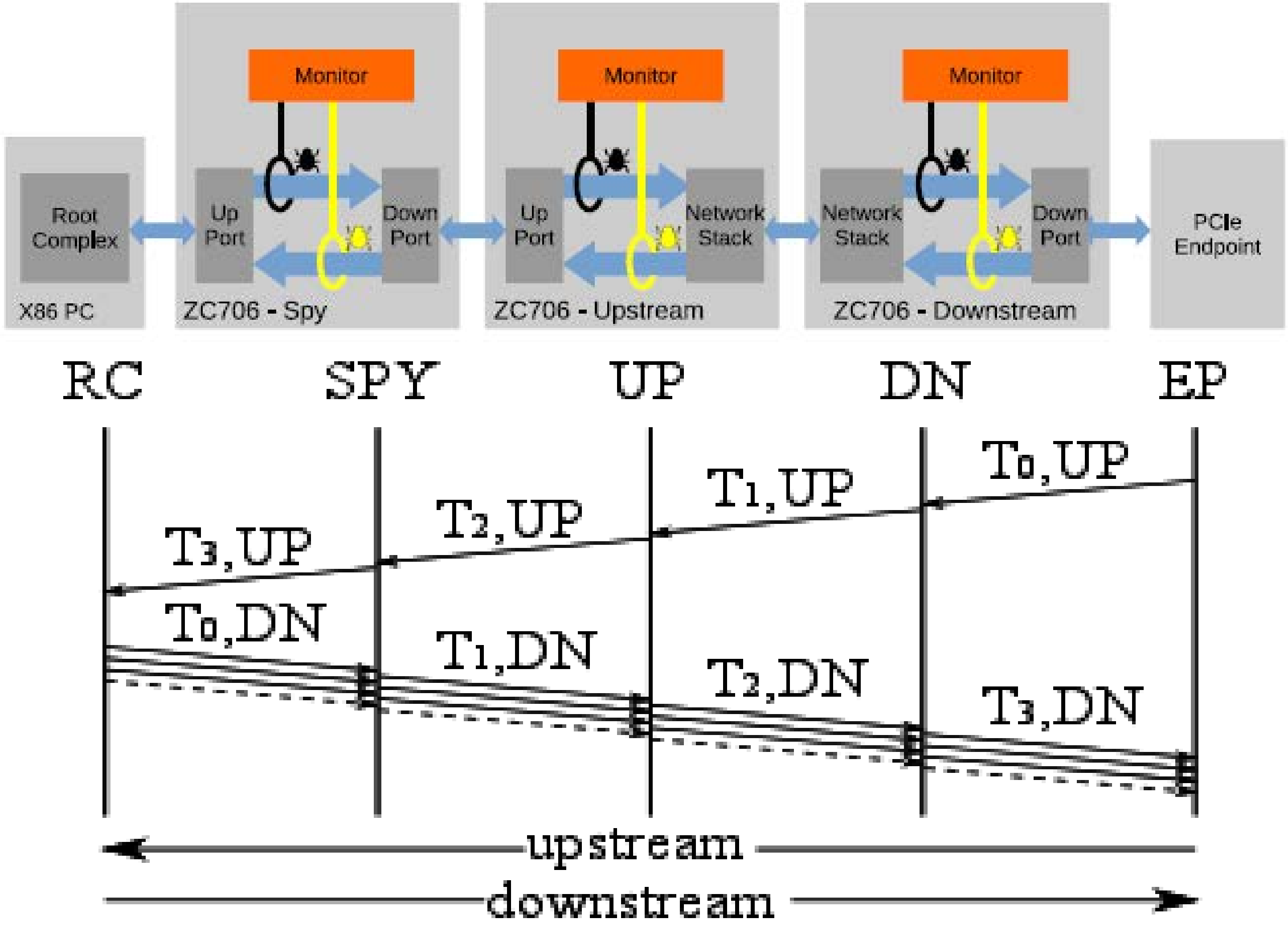
Tracer-Based Performance Analysis

- **Tracers within PCIe Transaction Layer Packets (TLP)**
  - – Based on addresses/ IDs, detected at PCIe switches and endpoints
  - – Transparent for transport layer (Ethernet, etc)

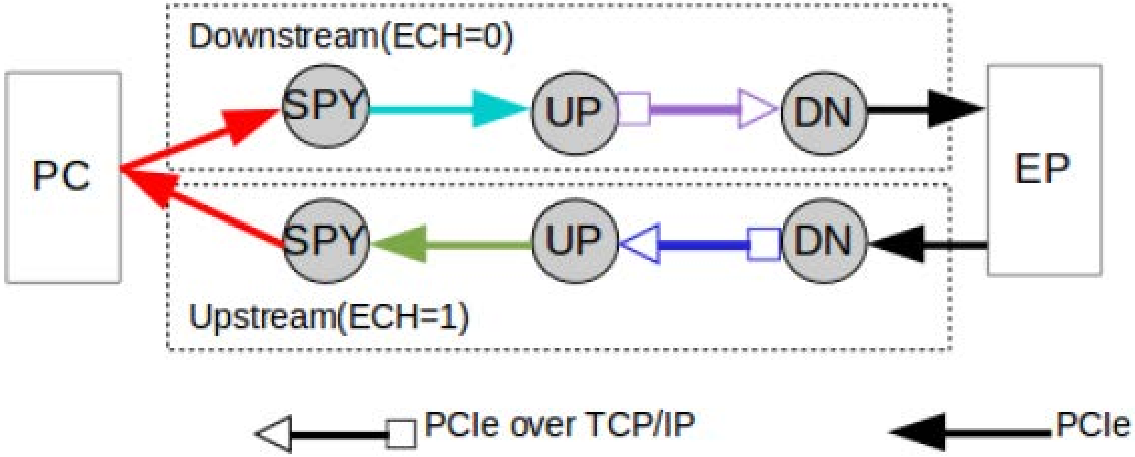


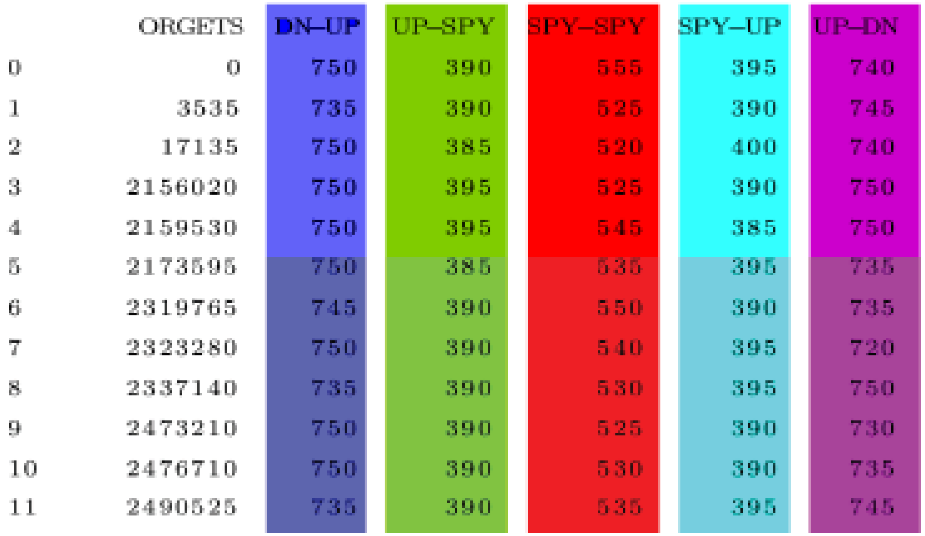
Proof-of-Concept Implementation

- **Full implementation on network with multiple boards**



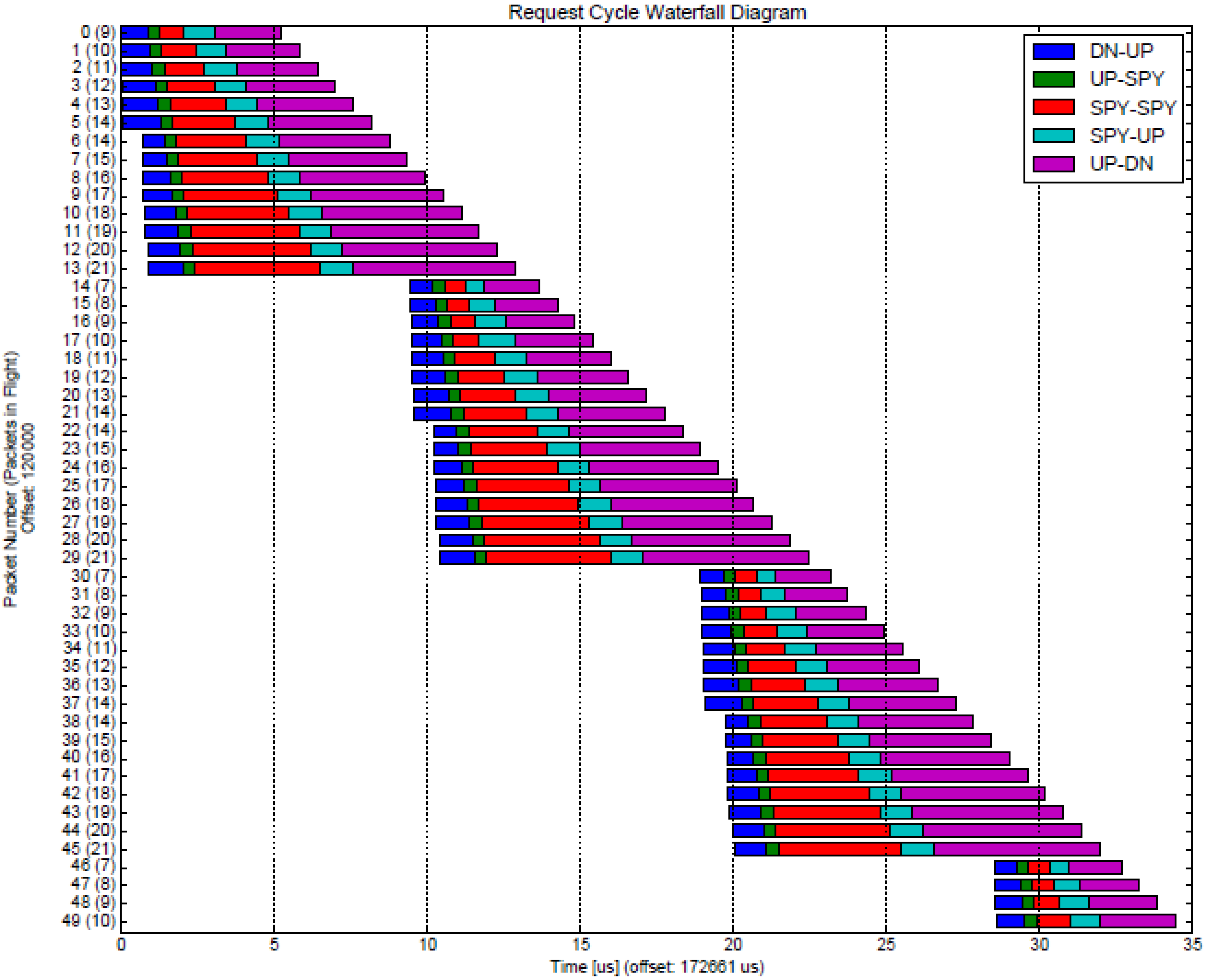
Latency Monitoring WithTracers - Overview





	ORGETS	DN–UP	UP–SPY	SPY–SPY	SPY–UP	UP–DN
0	0	750	390	555	395	740
1	3535	735	390	525	390	745
2	17135	750	385	520	400	740
3	2156020	750	395	525	390	750
4	2159530	750	395	545	385	750
5	2173595	750	385	535	395	735
6	2319765	745	390	550	390	735
7	2323280	750	390	540	395	720
8	2337140	735	390	530	395	750
9	2473210	750	390	525	390	730
10	2476710	750	390	530	390	735
11	2490525	735	390	535	395	745

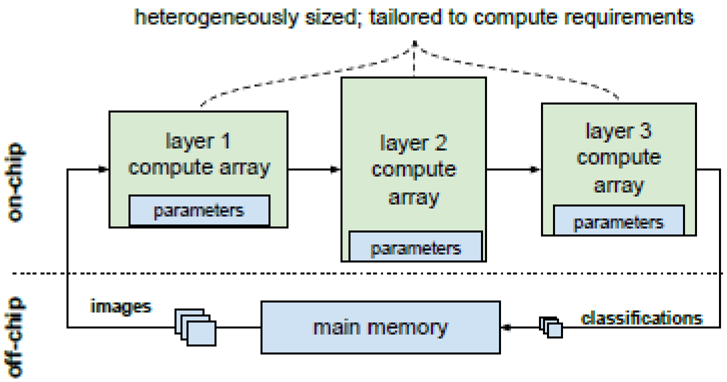
Latency Monitoring with Tracers - Results

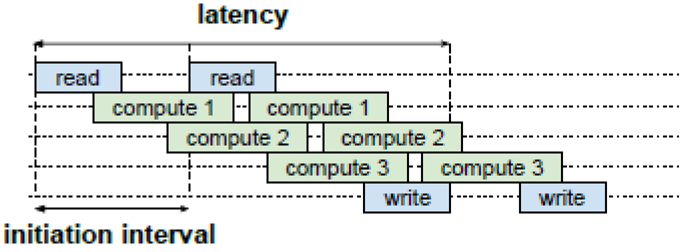


Software-Defined Services - Example 3) Inline Processing w/ Neural Networks

- **Deep Convolutional Neural Networks (CNN) have demonstrated values in classification, recognition and data-mining.**
- **However, CNN can be very compute intensive, when done at single or double float precision.**
- **Recent approaches involve reduced precision (INT8, or even less), as well as dataflow-oriented compute architectures.**
  - Taps into tremendous compute power within Programmable Logic
- **What if, CNN can be run close to the data, within the storage node?**

Streaming Dataflow Processing in BNN Inference





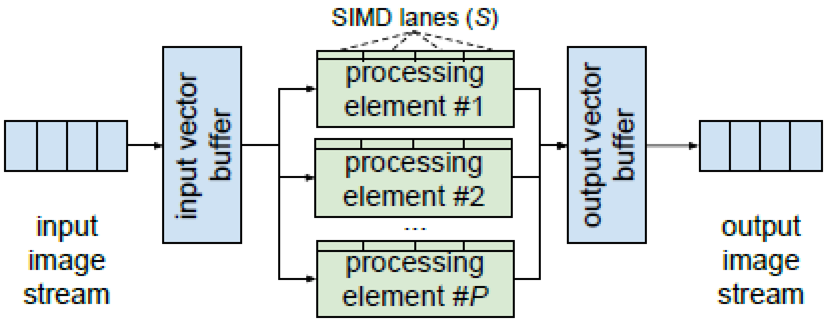
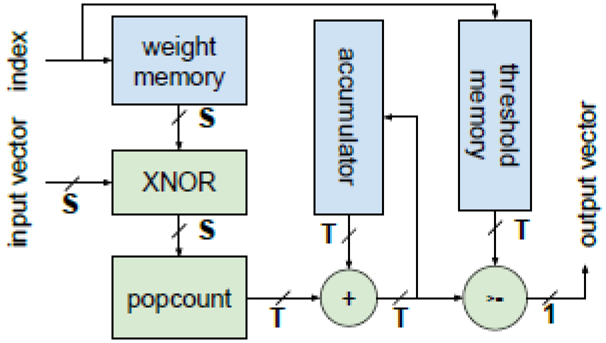


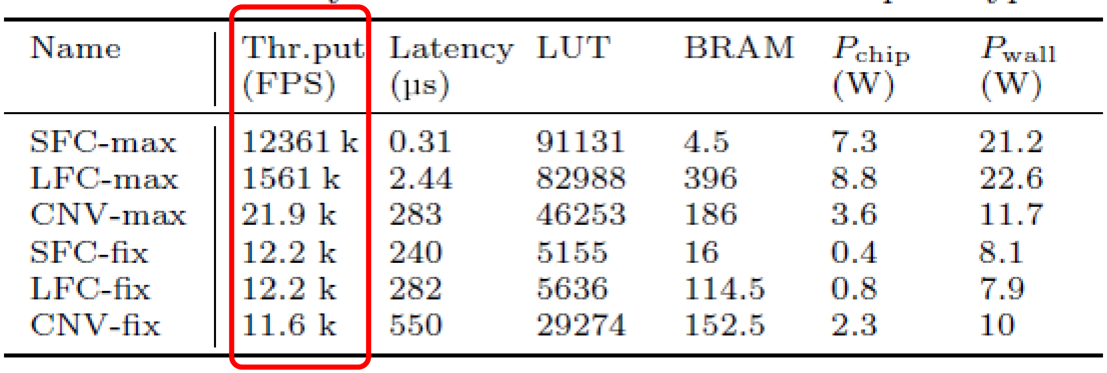
Figure 5: Overview of the MVTU.



Courtesy "FINN: A Framework for Fast, Scalable Binarized Neural Network Inference", Umuroglu, Fraser, Blott et al., 25th Symp. on FPGA, 2017

BNN Results

Table 3: Summary of results from FINN 200 MHz prototypes.



Name	Thr.put (FPS)	Latency (μs)	LUT	BRAM	P_{chip} (W)	P_{wall} (W)
SFC-max	12361 k	0.31	91131	4.5	7.3	21.2
LFC-max	1561 k	2.44	82988	396	8.8	22.6
CNV-max	21.9 k	283	46253	186	3.6	11.7
SFC-fix	12.2 k	240	5155	16	0.4	8.1
LFC-fix	12.2 k	282	5636	114.5	0.8	7.9
CNV-fix	11.6 k	550	29274	152.5	2.3	10

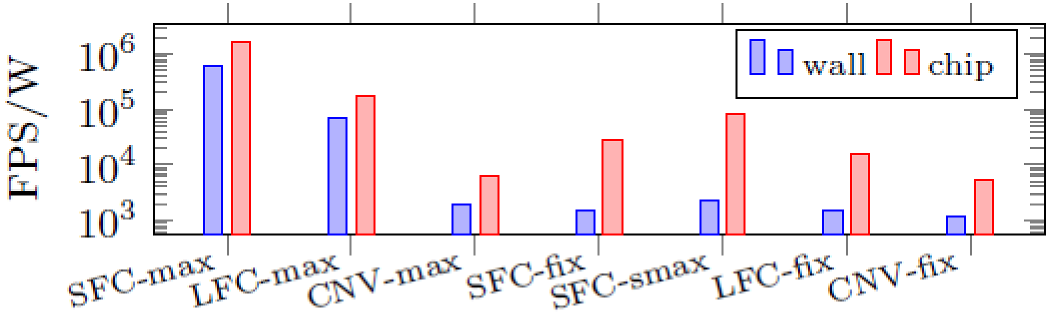


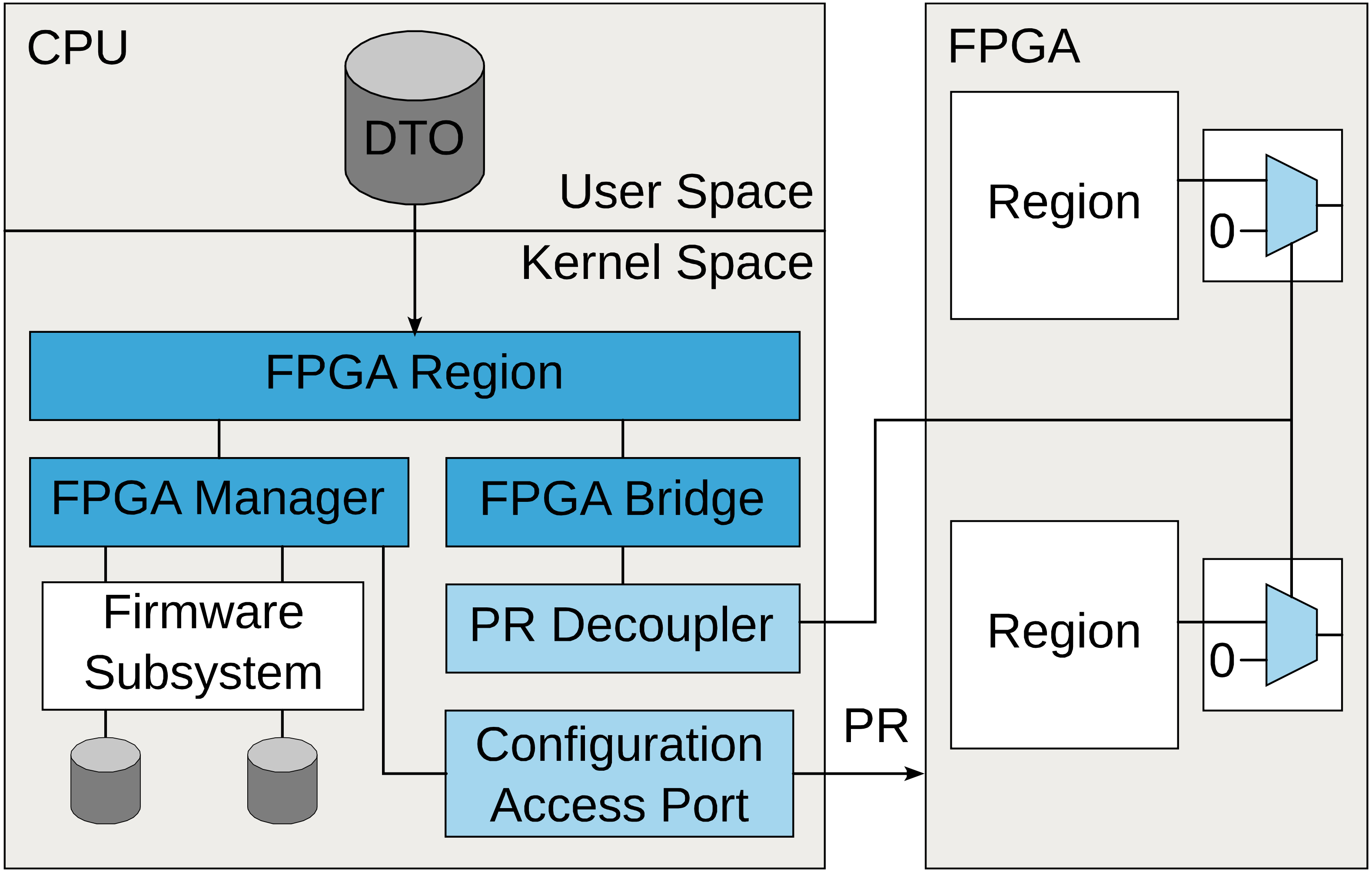
Figure 10: Prototype energy efficiency.

Courtesy “FINN: A Framework for Fast, Scalable Binarized Neural Network Inference”, Umuroglu, Fraser, Blott et al., 25th Symp. on FPGA, 2017

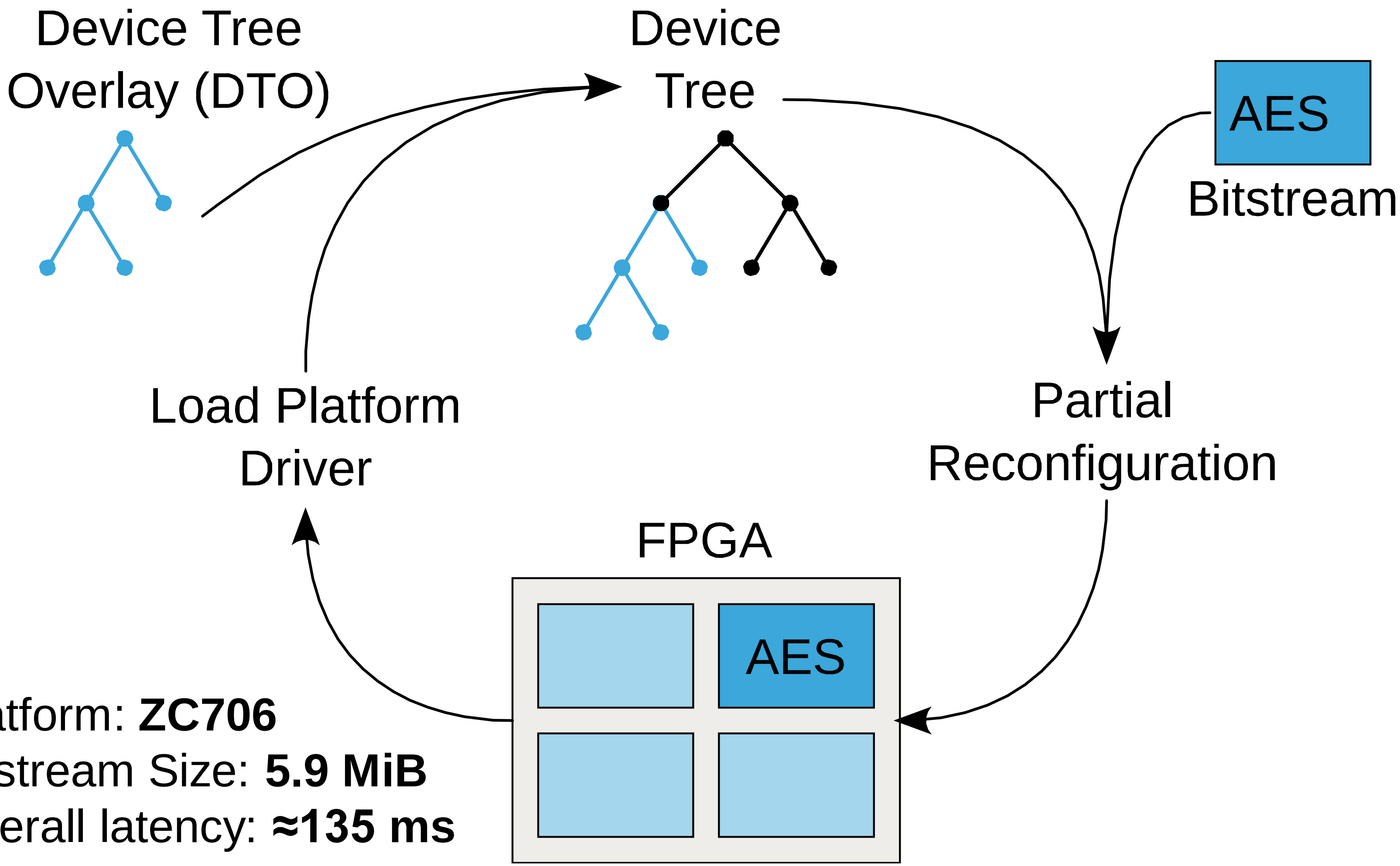
Software-Defined Services – Infrastructure Linux Kernel FPGA Framework

- **Supports both full and partial reconfiguration of FPGAs**
- **Adds a device tree interface for controlling the partial reconfiguration process**
- **Handles all FPGA internal processes**
- **Abstract device and vendor neutral interface**

Linux FPGA Framework Architecture

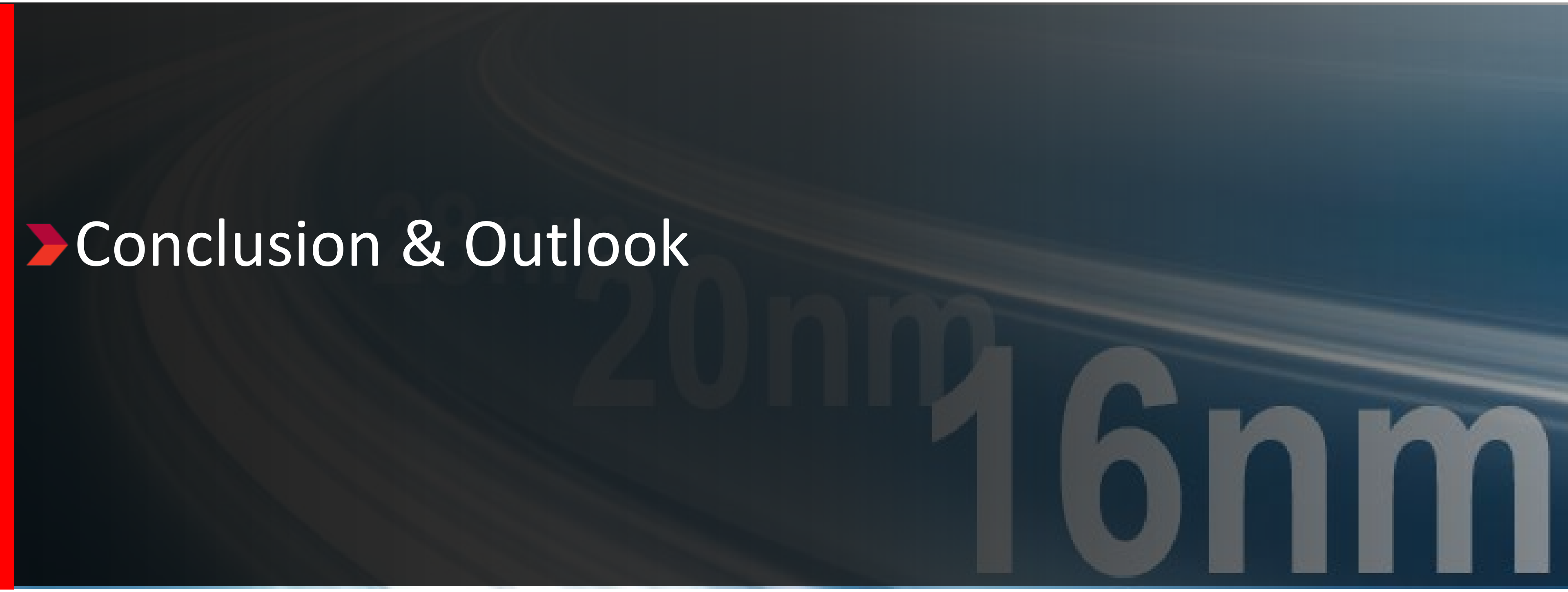


A Declarative Partial Reconfiguration Framework



Platform: **ZC706**
Bitstream Size: **5.9 MiB**
Overall latency: **≈135 ms**

Conclusion & Outlook

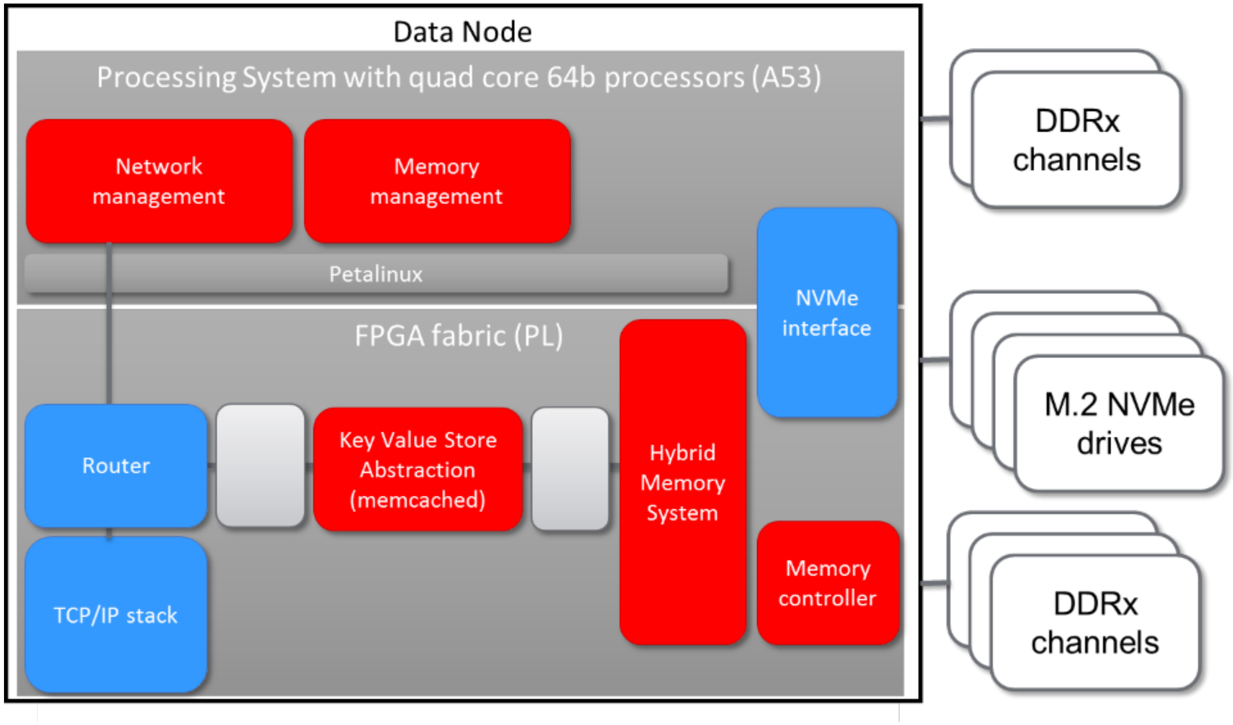


Conclusion

- **Trend towards unconventional architectures**
  - A diversification of increasingly heterogeneous devices and systems
  - Convergence of networking, compute and storage within single nodes
  - CPU-only processing runs out of steam
- **Key concepts for demonstrating Software-Defined Services**
  - Offload engines for Linux Kernel Crypto-API
  - Non-intrusive latency analysis via PCIe TLP "Tracers"
  - Inline processing with Deep Convolutional Neural Networks
- **Results:**
  - On commercially available hardware
  - Available for collaboration or in-house development

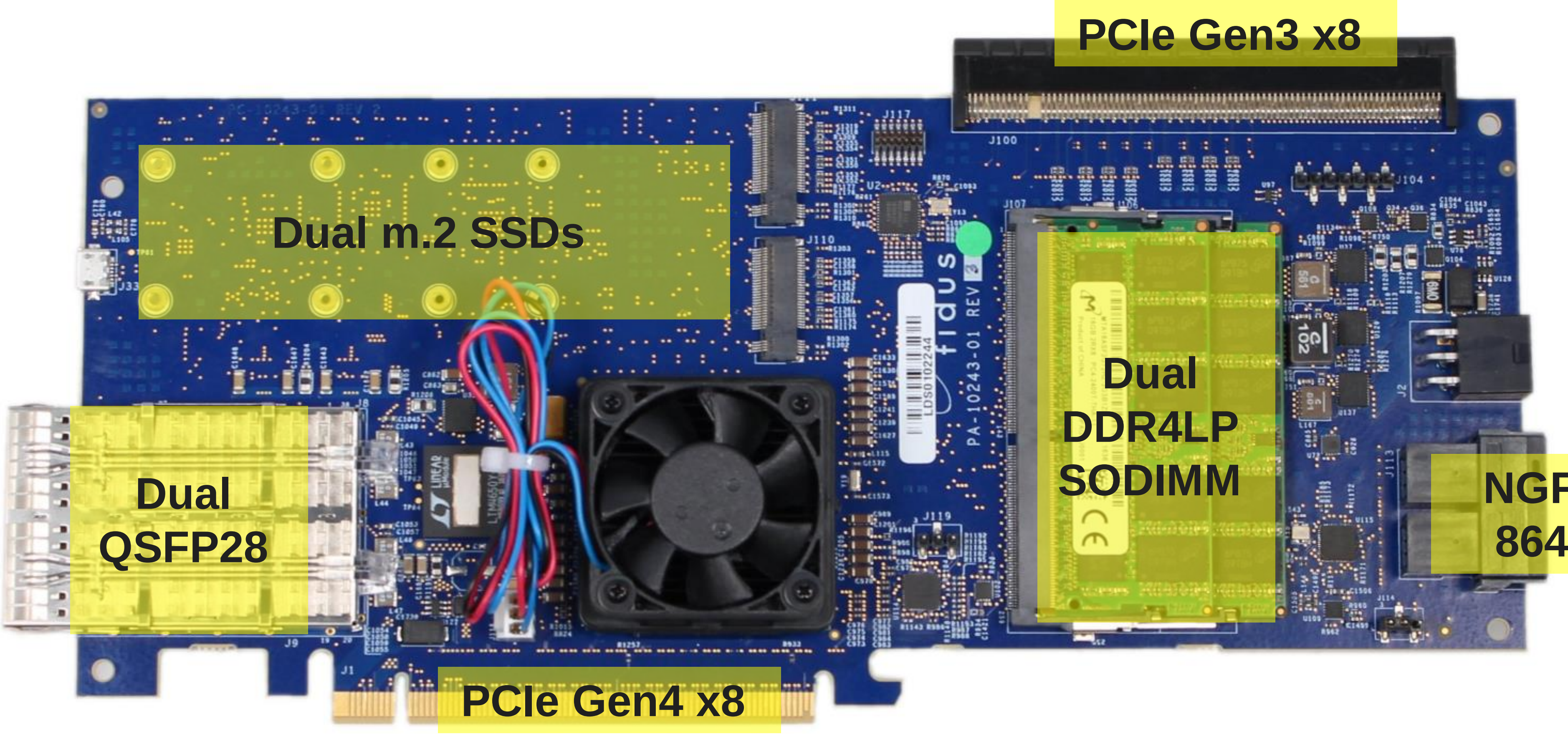
Single-Chip Implementation

- **Xilinx Zynq UltraScale+ MPSoC (XCZU19EG)**
  - ARM Cortex A-53 quad-core, ARM Coretx R5 dual-core, 1,968 DSP slices
  - 1.1 million system logic cells, 34Mbit BRAM, 36Mbit UltraRAM
  - 5x PCIe Gen3/4, 4x 100GigE, 44x 16.3Gbps, 28x 32.72Gbps



Commercially Available Development System

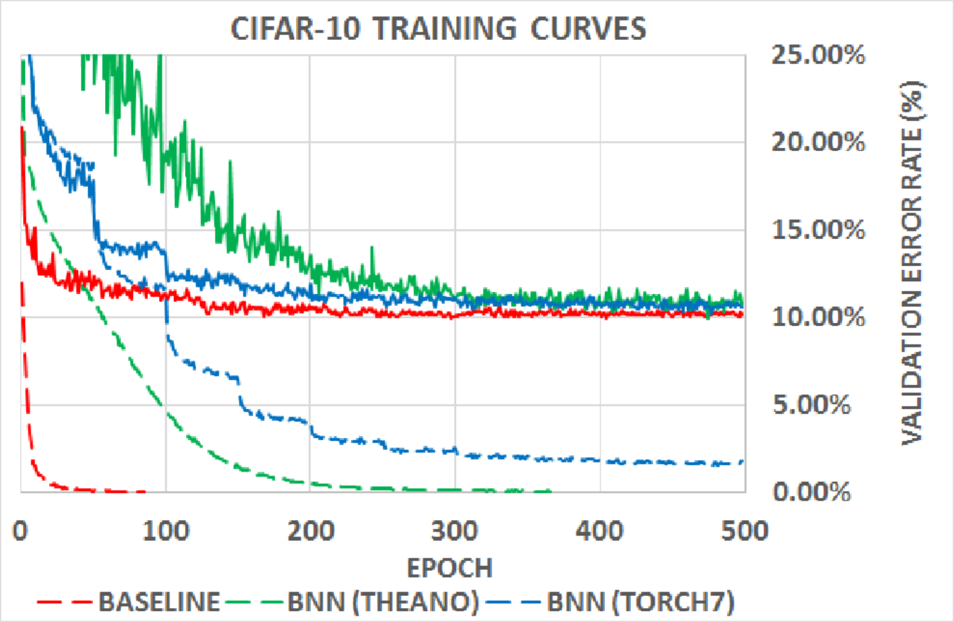
- Sidewinder-100 from Fidus Systems
- Accelerator IP and Linux BSP from MLE



Backup

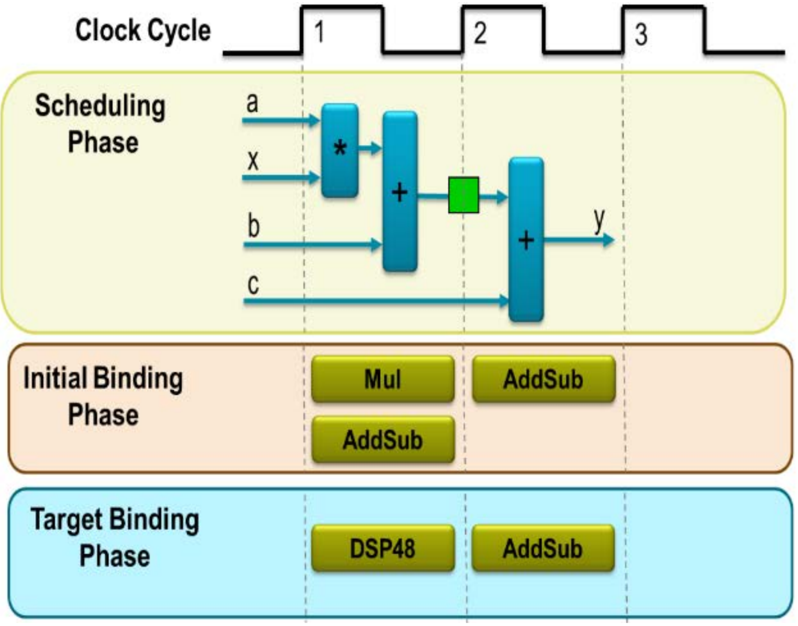
Reduced Precision Neural Networks

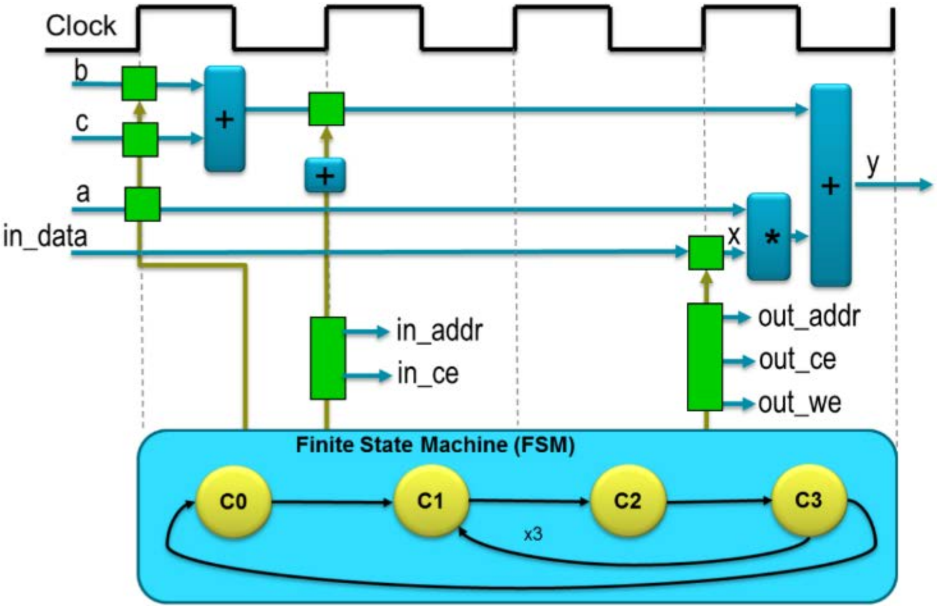
- **Binarized Neural Networks (BNN):**
  Training with float, CNN Inference runs at reduced precision
  – Less data (Mbytes) for parameters, less compute burdon.



Working Principles of High-Level Synthesis

- Design automation runs scheduling and resource binding to generate RTL code comprising data paths plus state machines for control flow

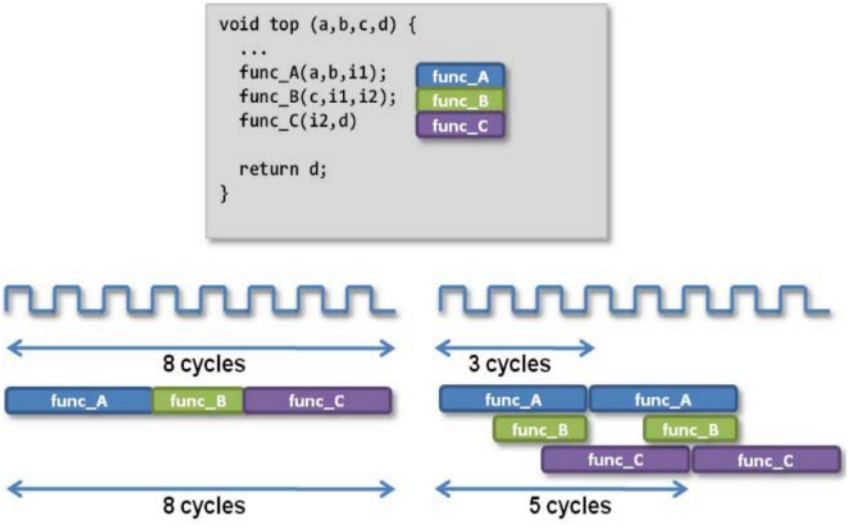




Benefits of HLS-Based C/C++ FPGA Design

- Automated performance optimizations via parallelization at dataflow level

- Automatic interface synthesis and driver code generation for HW/SW connectivity

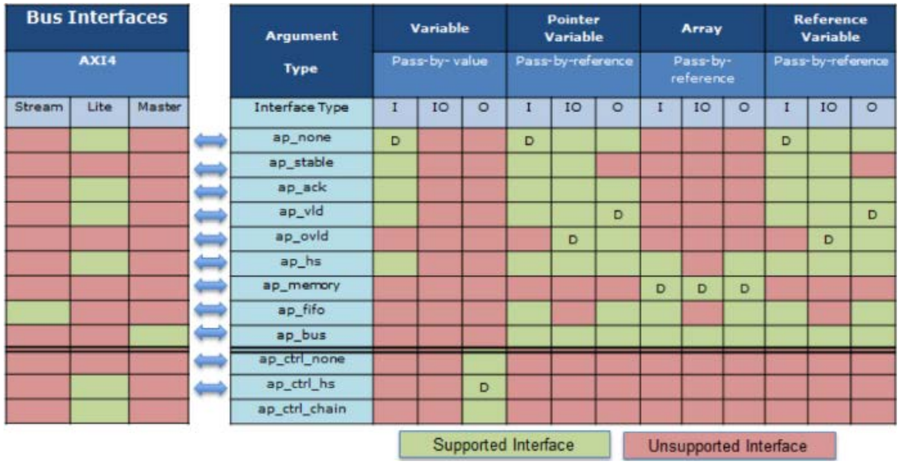


```
void top (a,b,c,d) {
  ...
  func_A(a,b,i1);
  func_B(c,i1,i2);
  func_C(i2,d)

  return d;
}
```

(A) Without Dataflow Pipelining

(B) With Dataflow Pipelining

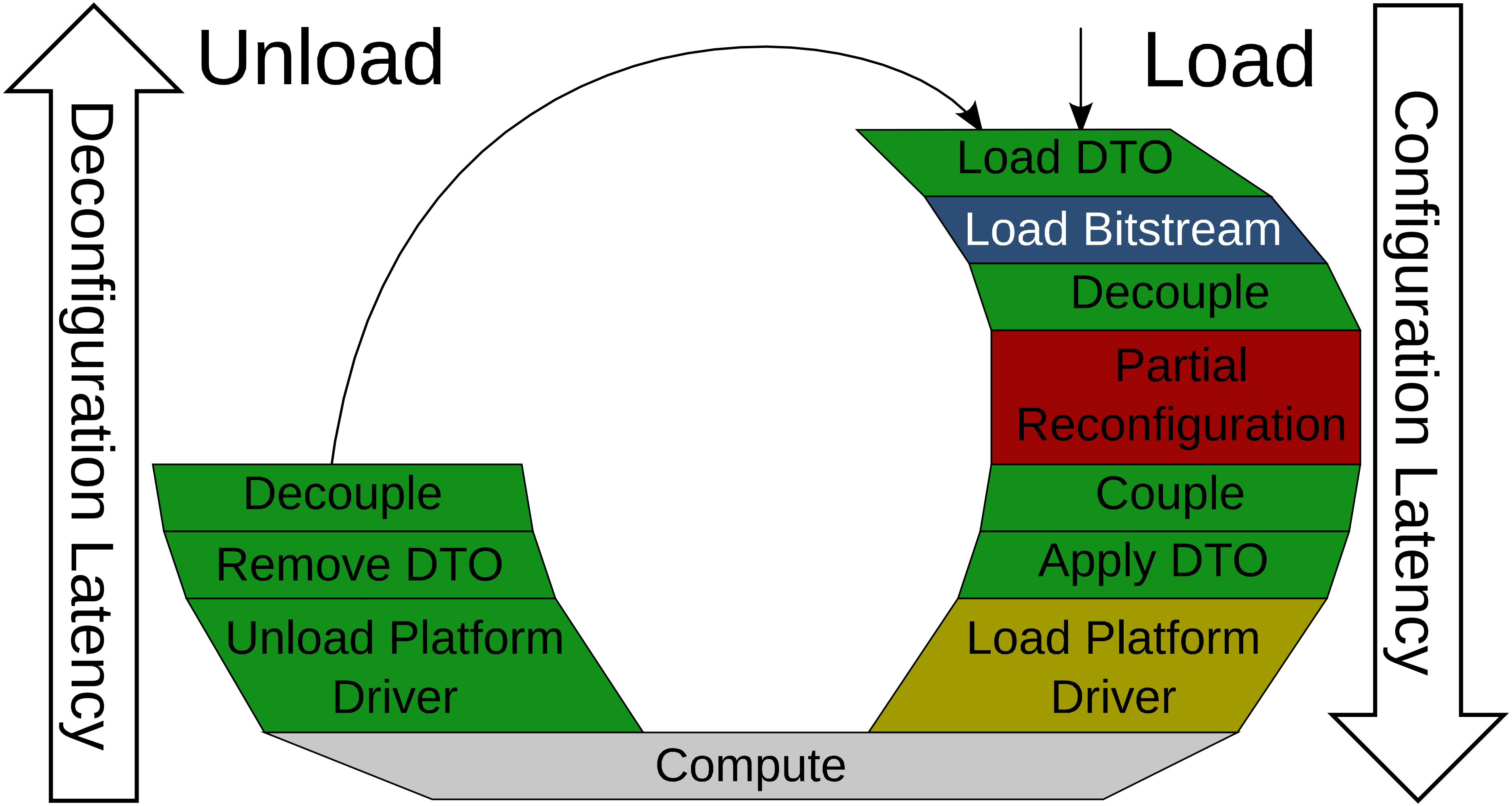


Bus Interfaces			Argument	Variable			Pointer Variable			Array			Reference Variable		
AXI4			Type	Pass-by- value			Pass-by-reference			Pass-by-reference			Pass-by-reference		
Stream	Lite	Master	Interface Type	I	IO	O	I	IO	O	I	IO	O	I	IO	O
🟥	🟩	🟥	ap_none	🟩 D	🟥	🟥	🟩 D	🟩	🟩	🟥	🟥	🟥	🟩 D	🟩	🟩
🟥	🟥	🟥	ap_stable	🟩	🟥	🟥	🟩	🟩	🟥	🟥	🟥	🟥	🟩	🟩	🟥
🟥	🟩	🟥	ap_ack	🟩	🟥	🟥	🟩	🟩	🟩	🟥	🟥	🟥	🟩	🟩	🟩
🟥	🟩	🟥	ap_vld	🟩	🟥	🟥	🟩	🟩	🟩 D	🟥	🟥	🟥	🟩	🟩	🟩 D
🟥	🟥	🟥	ap_ovld	🟥	🟥	🟥	🟥	🟩 D	🟩	🟥	🟥	🟥	🟥	🟩 D	🟩
🟥	🟩	🟥	ap_hs	🟩	🟥	🟥	🟩	🟩	🟩	🟩	🟥	🟩	🟩	🟩	🟩
🟥	🟥	🟥	ap_memory	🟥	🟥	🟥	🟥	🟥	🟥	🟩 D	🟩 D	🟩 D	🟥	🟥	🟥
🟩	🟥	🟥	ap_fifo	🟥	🟥	🟥	🟩	🟥	🟩	🟩	🟥	🟩	🟩	🟥	🟩
🟥	🟥	🟩	ap_bus	🟥	🟥	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
🟥	🟥	🟥	ap_ctrl_none	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
🟥	🟩	🟥	ap_ctrl_hs	🟥	🟥	🟩 D	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
🟥	🟩	🟥	ap_ctrl_chain	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥

🟩 Supported Interface

🟥 Unsupported Interface

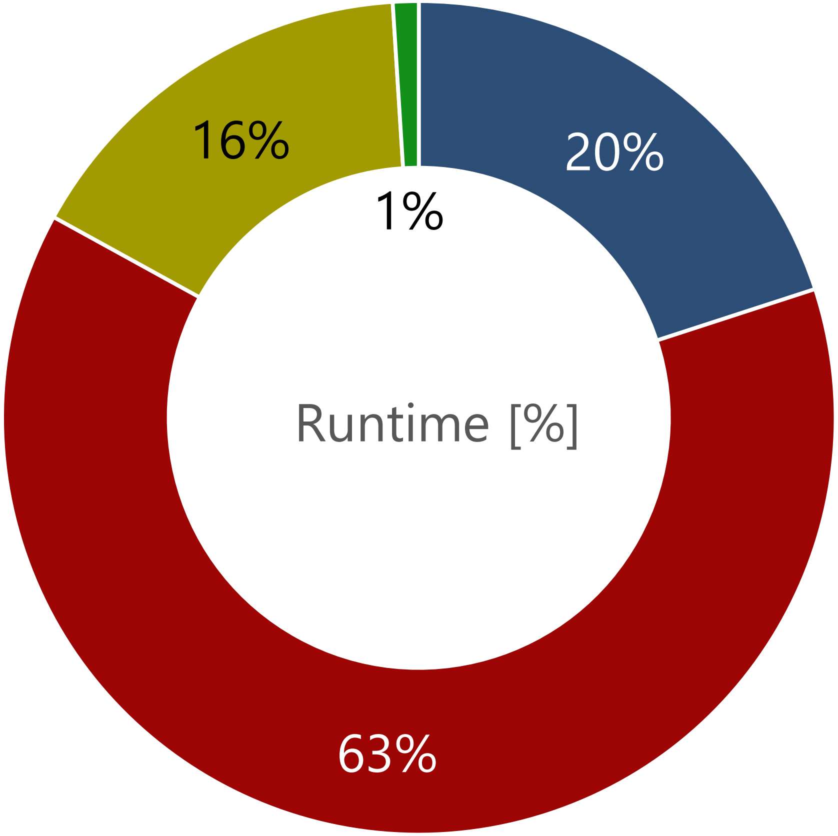
Reconfiguration Performance



Scheduling Latency - Profiling Results

- **Measurement of example system (AES accelerator on ZC706 board)**
- **Measured latencies via *ftrace* function entry and exit timestamps**

- **Bitstream Size: 5.9 MiB**
- **Overall latency: ≈135 ms**



Contact

- **Endric Schubert**
  Email: endric@mlecorp.com

- **Ulrich Langenbach**
  Email: ulrich@mlecorp.com

- **Missing Link Electronics**
  www.missinglinkelectronics.com
  Ph US: +1-408-475-1490
  Ph GER: +49-731-141149-0