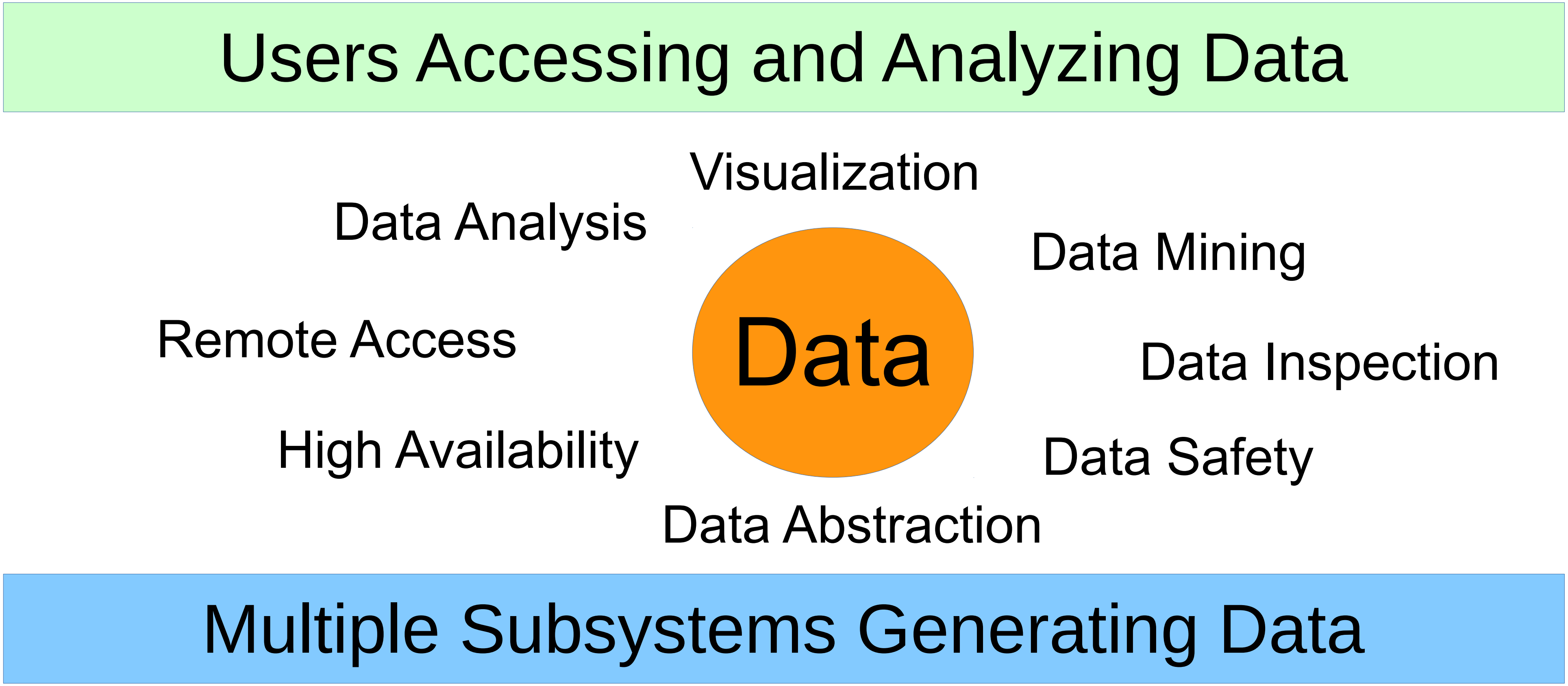
UFO Cloud

Data-Acquisition-as-a-Service

Suren A. Chilingaryan
Institute for Data Processing and Electronics at KIT



Users Accessing and Analyzing Data

Visualization

Data Analysis

Data Mining

Remote Access

Data

Data Inspection

High Availability

Data Safety

Data Abstraction

Multiple Subsystems Generating Data

New challenges for DAQ Software

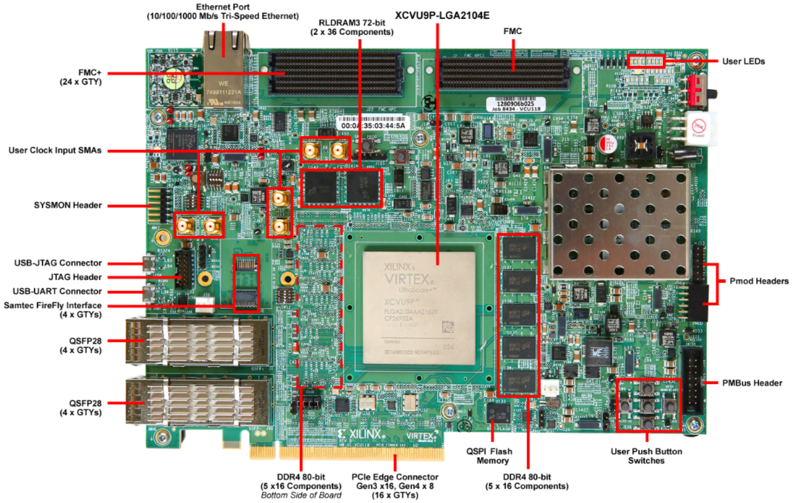
- **New detectors: Extreme data rates**
  - Can’t store all the data: online data reduction is needed
  - Moving between sites is slow: remote analysis services are needed
- **Increased automation: High throughput of samples/runs**
  - Detect the problems already during acquisition
  - Automate curation of the stored data
- **Uneven resource utilization: High investments and power balance**
  - Multiple experiment phases: Acquisition, analysis, curration, etc.
  - Huge load spikes before meetings and conferences
- **More complex data processing chains**

Detectors meet HPC Cloud

- Direct ingress of real-time data in the HPC environment over tge fast Ethernet fabric
- Move control tasks to the HPC and rely on Cloud technologies (IaaS/PaaS) to improve scalability and reliability of the service
- Utilize available hardware accelerators (GPUs, FPGAs, Many-core CPUs, …) to improve performance
- Use Scientific Workflow Engines to simplify development of distributed data processing software
- Integrate automated data quality verification based on statistical and AI-based methods
- Offer long-term storage facilities to the users and provide remote data visualization and analysis services.

Data Ingress

- Ethernet interfaces are nowadays faster than PCIe links
- Ethernet cables up to 100m are readily available. Ranges up to 10-50 km can be covered with Fiber cables.
- ROCe (UDP-based) and iWARP (TCP-based) extensions allow to RDMA data directly in the system or GPU memory



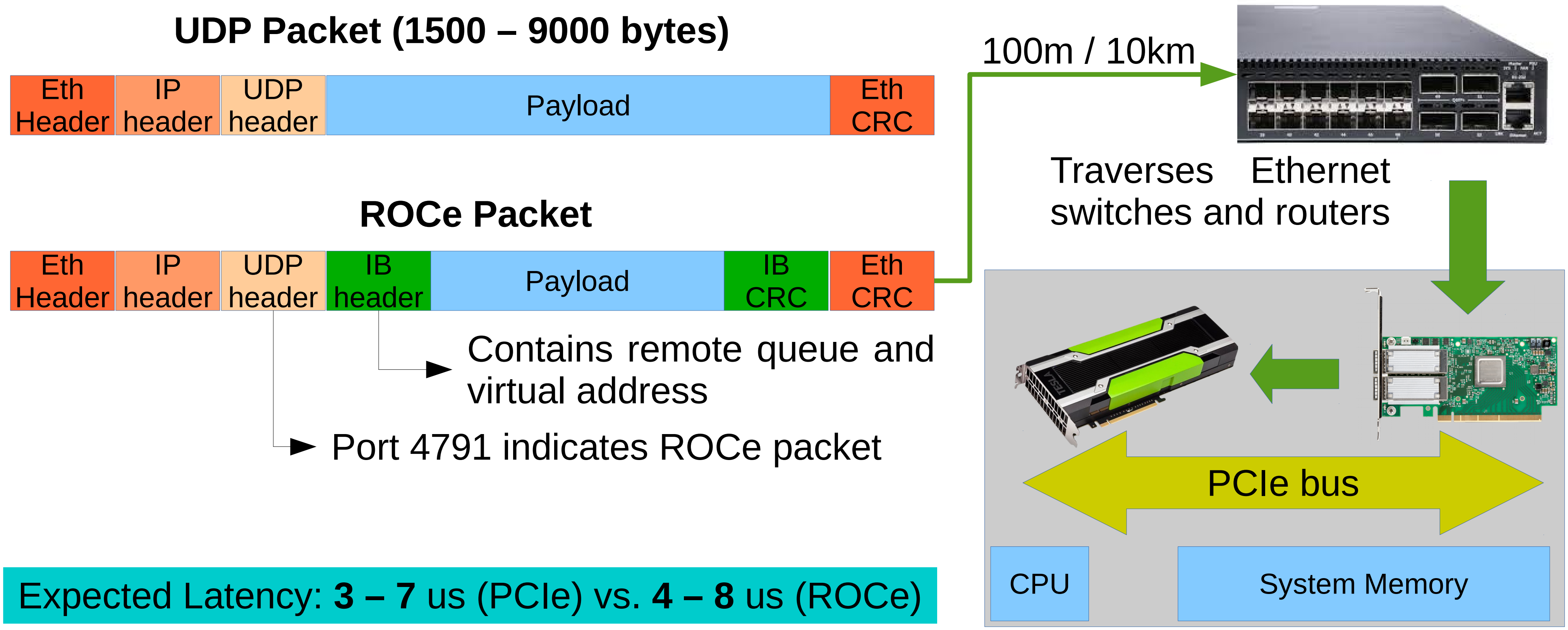
Xilinx VCU 118

Interface performance

PCI express gen. 3 x16	~ 12 GB/s
2x Ethernet QSFP28 (100 Gbit/s)	~ **25 GB/s**

ROCe extension

- ROCe encapsulates Infiniband headers in the payload of UDP packet which can traverse standard Ethernet infrastructure.
- 4791 port in UDP header indicates ROCe packet and the UDP payload, then, includes additionally an Infiniband header and checksum
- Infiniband header contains ID of remote queue and a virtual address to read/write the data from/to.



UDP Packet (1500 – 9000 bytes)

Eth Header	IP header	UDP header	Payload	Eth CRC

ROCe Packet

Eth Header	IP header	UDP header	IB header	Payload	IB CRC	Eth CRC

Contains remote queue and virtual address

Port 4791 indicates ROCe packet

100m / 10km

Traverses Ethernet switches and routers

PCIe bus

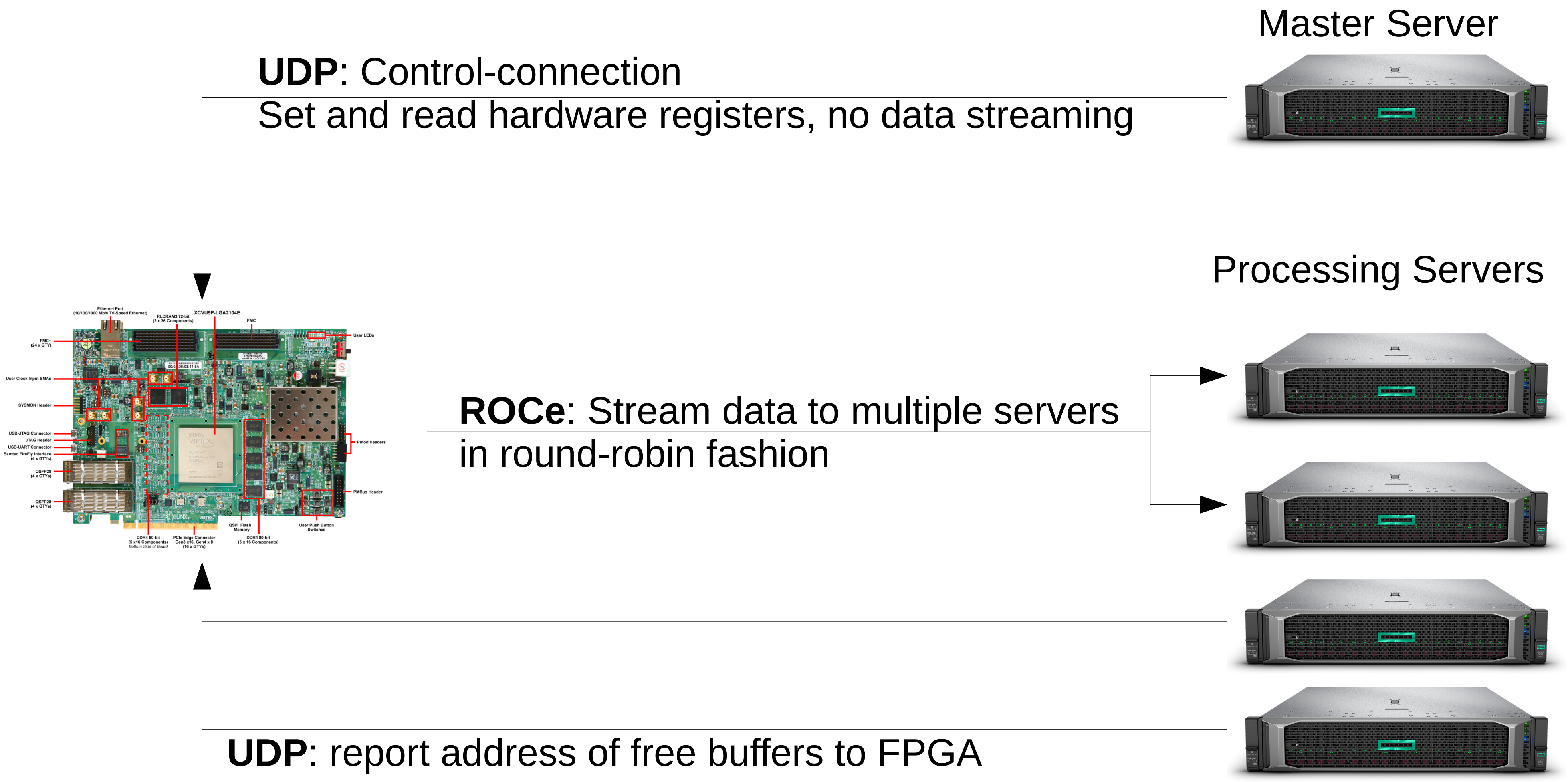
CPU

System Memory

Expected Latency: **3 – 7** us (PCIe) vs. **4 – 8** us (ROCe)

Processing Cluster

- Master server only configures the hardware and the data processing-nodes, but doesn't receive any data
- Processing nodes send UDP packet with buffer addresses to the FPGA which responds with the data in round-robin fashion.



Master Server

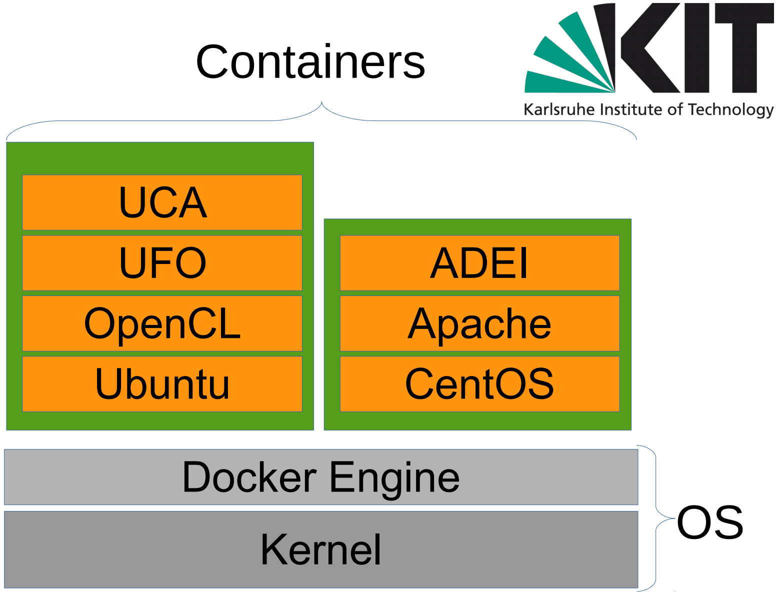
UDP: Control-connection
Set and read hardware registers, no data streaming

Processing Servers

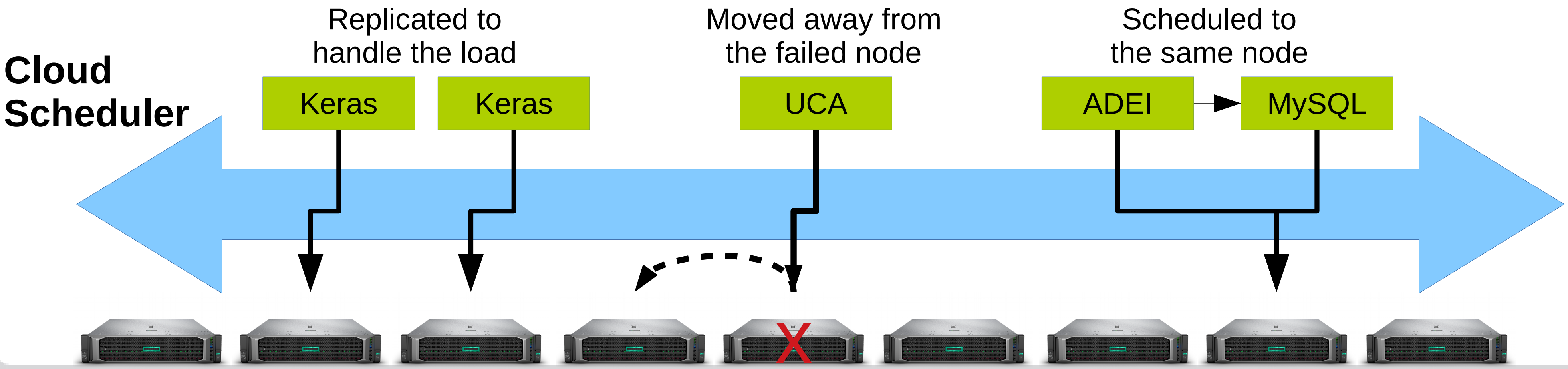
ROCe: Stream data to multiple servers in round-robin fashion

UDP: report address of free buffers to FPGA

Software Platform

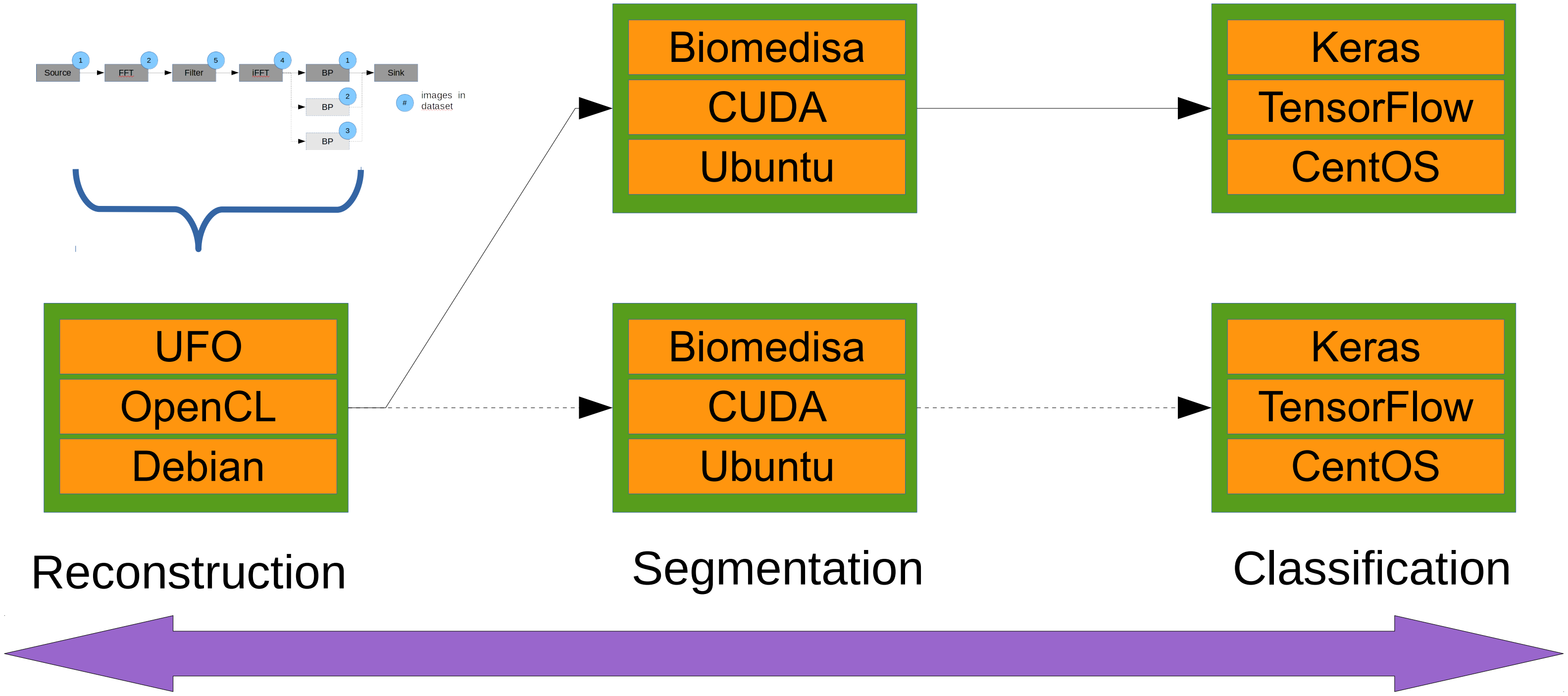


- **Containers**
  - Pack application with all dependencies
  - Isolation (problems & resources)
  - Low overhead
- **Private Cloud Infrastructure**
  - Load-balancing: Stops / starts additional replicas according to the load
  - High-availability: Restart failed services, migrate from failed nodes
  - Resource management: Allocate nodes to apps, set memory/cpu limits
  - Security: Allows to share hardware without sharing the data


Cloud Scheduler
Replicated to handle the load
Keras
Keras
Moved away from the failed node
UCA
Scheduled to the same node
ADEI
MySQL

Container-Native Workflow

- Connect containers to achieve the desired data flow and results
- Each filter may process data on a GPU or CPU with CUDA/OpenCL/OpenMP/...
- Scientific Workflow Engine schedules execution of task
- Execution is distributed for efficient use of available nodes and network bandwidth
- Duplicates sub path for multi-node execution
- Base on the existing Scientific Workflow Engine, like Project Argo or Pegasus



UFO
OpenCL
Debian

Reconstruction

Biomedisa
CUDA
Ubuntu

Biomedisa
CUDA
Ubuntu

Segmentation

Keras
TensorFlow
CentOS

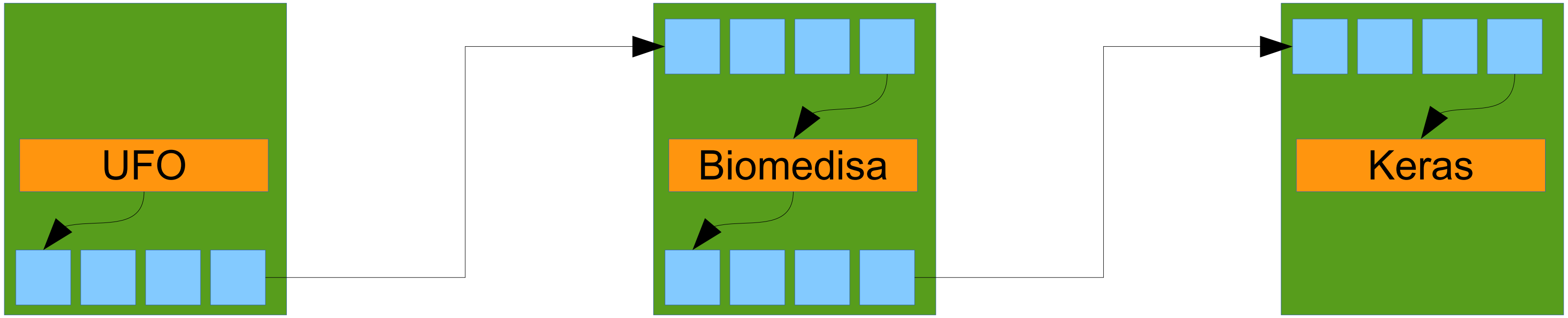
Keras
TensorFlow
CentOS

Classification

Executed on different nodes

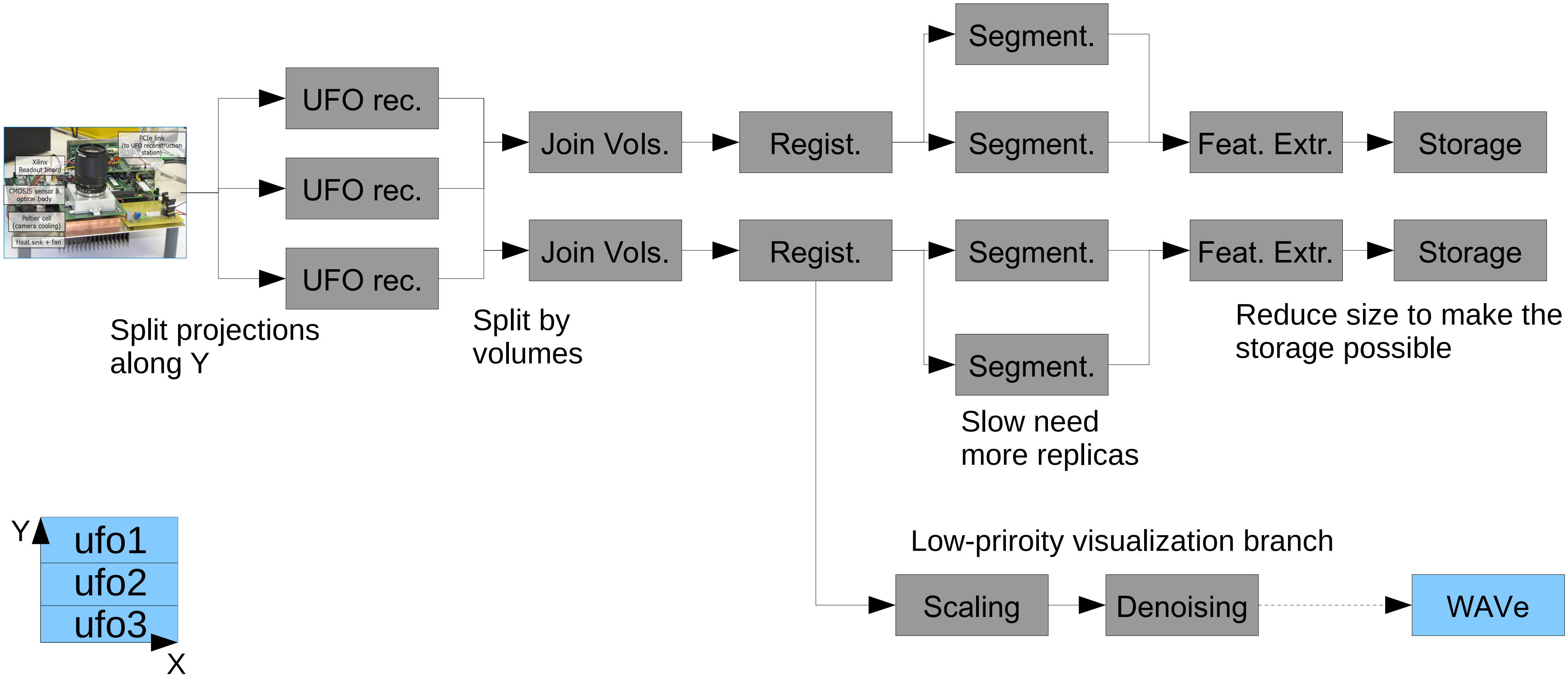
Container Communication

- The RDMA mechanism is used to transfer data between containers
- Container passes the produced data to the scheduler for delivery while it produces the next data set
- Scheduler requests a new buffer in the memory of a next container in queue and starts RDMA transfer. Upon completion the buffer is put in the processing queue
- If too many buffers are in queue, a new replica of the container may be started



Sample Tomography Workflow

- Multiple reconstruction nodes used to handle the load
- Data from all reconstructors is combined in a single volume and distributed further on split on the volume basis
- For compute-intensive tasks more replicas are launched
- Subset of volumes is prepared for online visualization



Experiment Life-cycle

- **Data Acquisition Phase**
  - Data reduction
  - Real-time reconstruction
  - Monitoring
  - Slow control

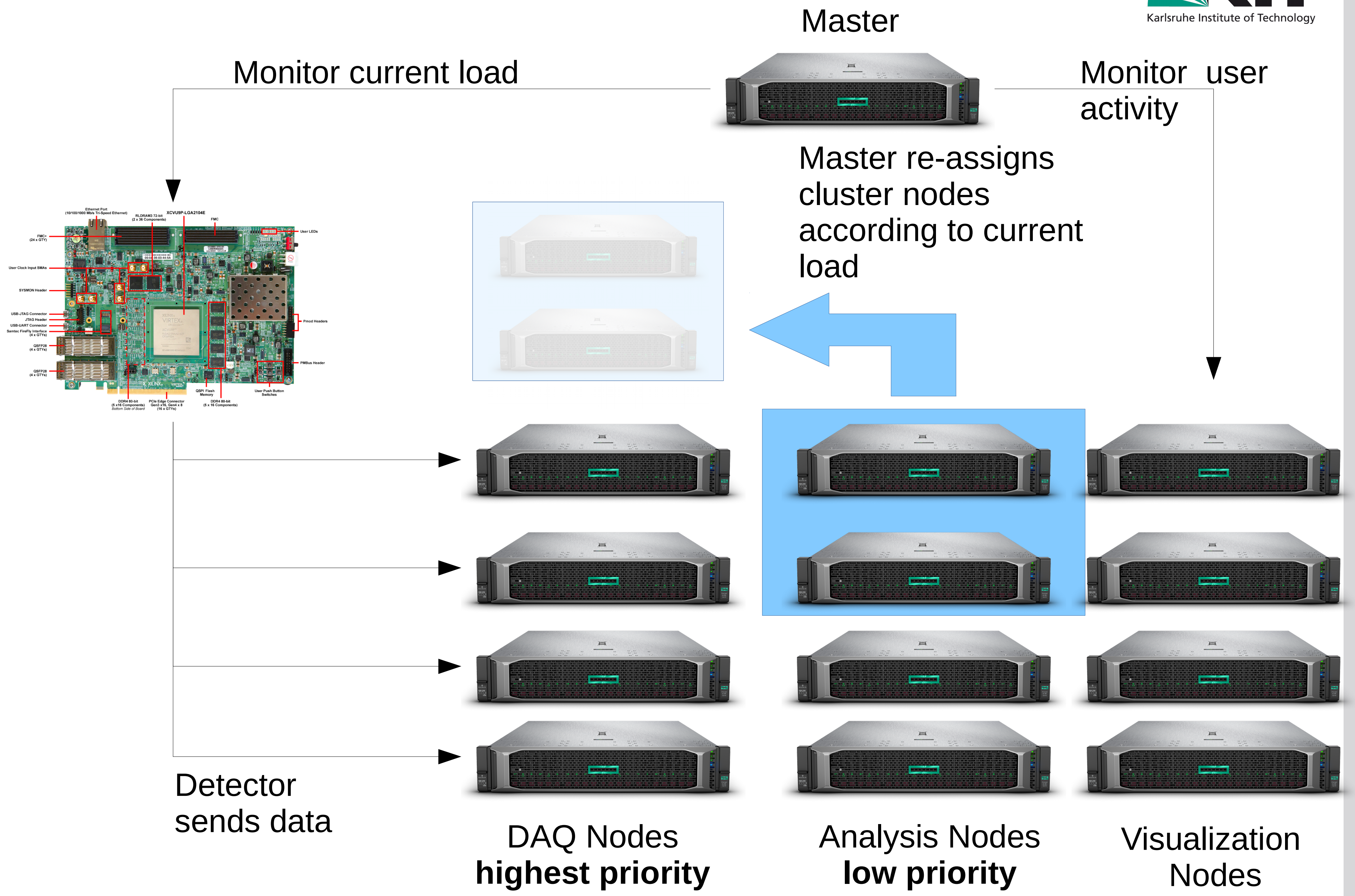
- **Offline Data Analysis Phase**
  - Quality control and automated data preparation (i.e. Registration, fully automated segmentation, generation of previews, etc.)

- **Interactive Remote Analysis Phase**
  - Data Visualization
  - User-assisted analysis

Re-balancing Load

- **Improving utilization of IT infrastructure**
  - Similar resources required during all phases: GPUs, Storage, ….
  - Readout nodes can be used for offline analysis when detector is not streaming data
  - Not critical if offline analysis is executed few hours or days later
- **Priorities**
  - **Highest**: Readout
  - **Normal**: Monitoring and serving interactive user requests
  - **Idle**: Offline analysis and data pre-processing.

Re-balancing load



Master

Monitor current load

Monitor user activity

Master re-assigns cluster nodes according to current load

Detector sends data

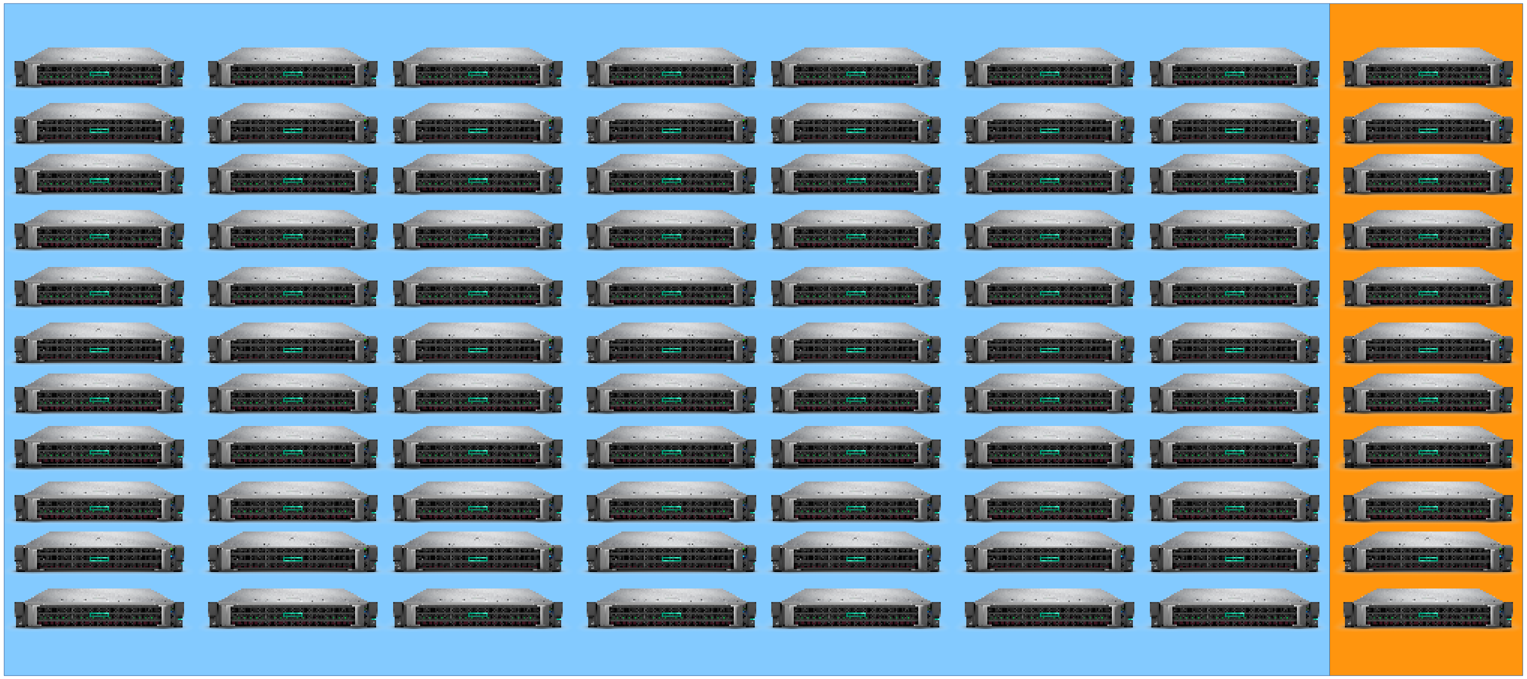
DAQ Nodes
highest priority

Analysis Nodes
low priority

Visualization Nodes

Ethernet Port (10/100/1000 Mb/s Tri-Speed Ethernet)
RLDRAM3 72-bit (2 x 36 Components)
XCVU9P-LGA2104E
FMC
User LEDs
FMC+ (24 x GTY)
User Clock Input SMAs
SYSMON Header
USB-JTAG Connector
JTAG Header
USB-UART Connector
Samtec FireFly Interface (4 x GTYs)
QSFP28 (4 x GTYs)
QSFP28 (4 x GTYs)
Pmod Headers
PMBus Header
DDR4 80-bit (5 x16 Components) Bottom Side of Board
PCIe Edge Connector Gen3 x16, Gen4 x 8 (16 x GTYs)
QSPI Flash Memory
DDR4 80-bit (5 x 16 Components)
User Push Button Switches

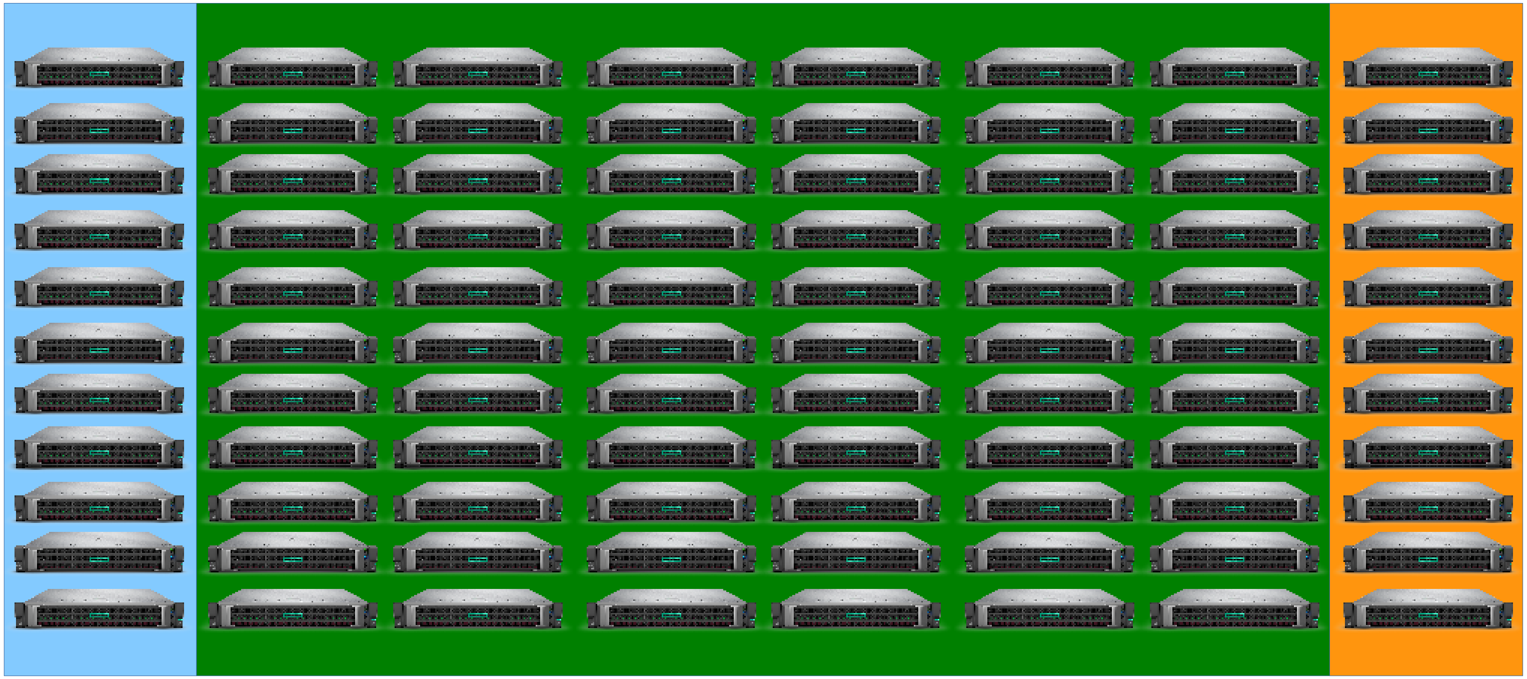
Data Acquisition Phase



DAQ Nodes

Interactive Nodes

Night

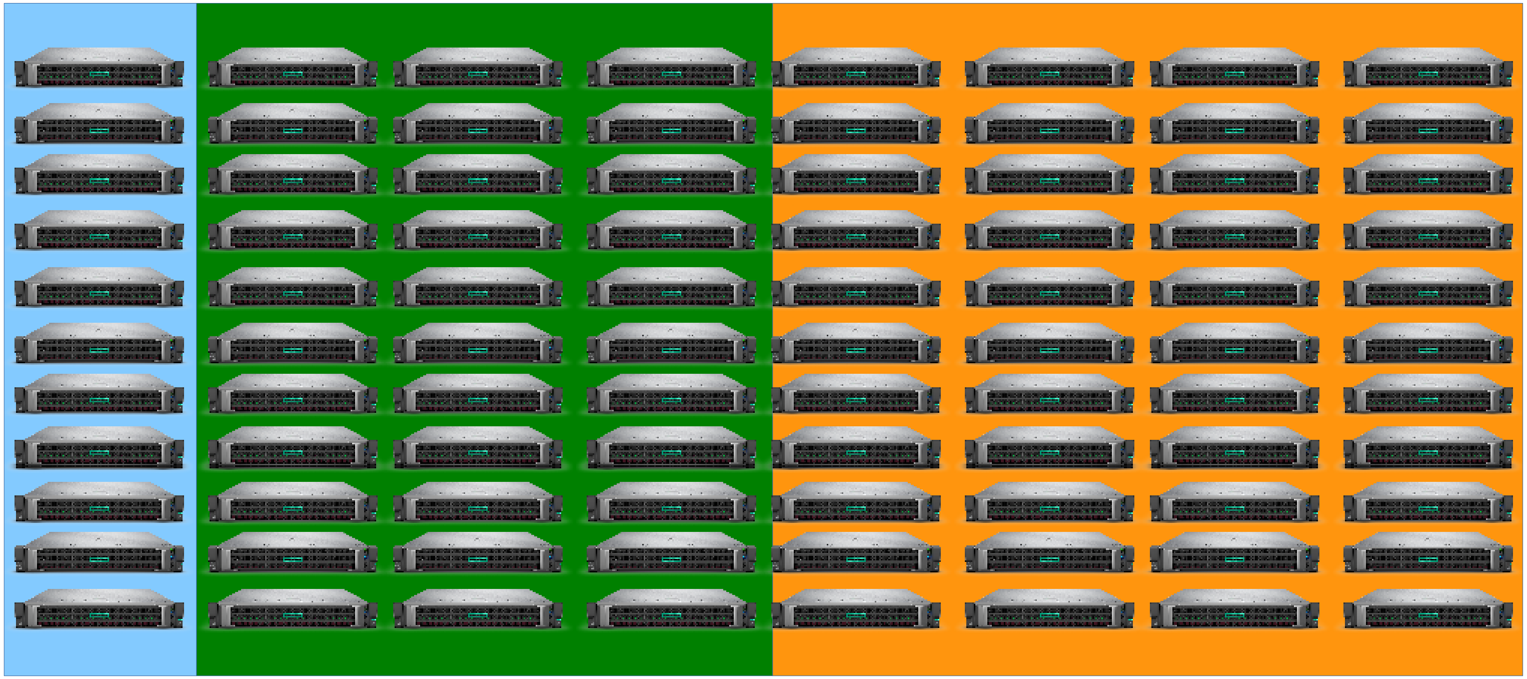


DAQ Nodes

Analysis Nodes

Interactive Nodes

Remote users connect



DAQ Nodes

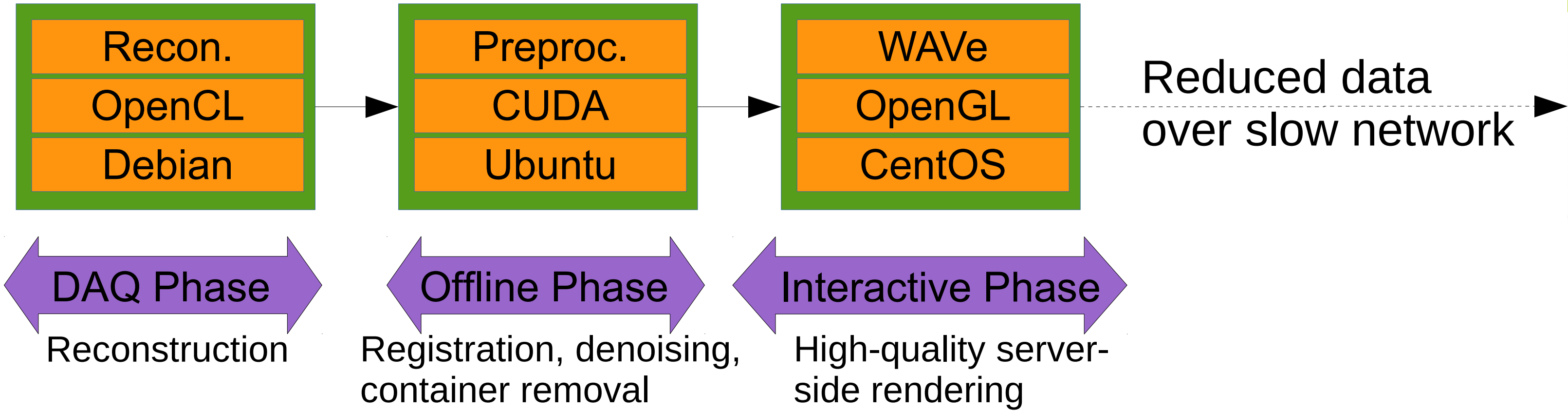
Analysis Nodes

Interactive Nodes

Remote Analysis

Remote Visualization with WAVe

- Multiple phases of data processing and preparation (different priorities)
- Hybrid Server-side (high quality) and client-side (interactive) rendering



Recon.
OpenCL
Debian

Preproc.
CUDA
Ubuntu

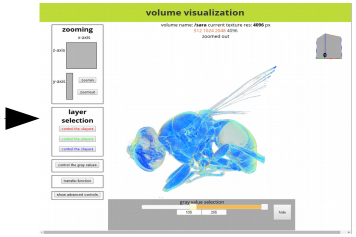
WAVe
OpenGL
CentOS

Reduced data over slow network

DAQ Phase
Reconstruction

Offline Phase
Registration, denoising, container removal

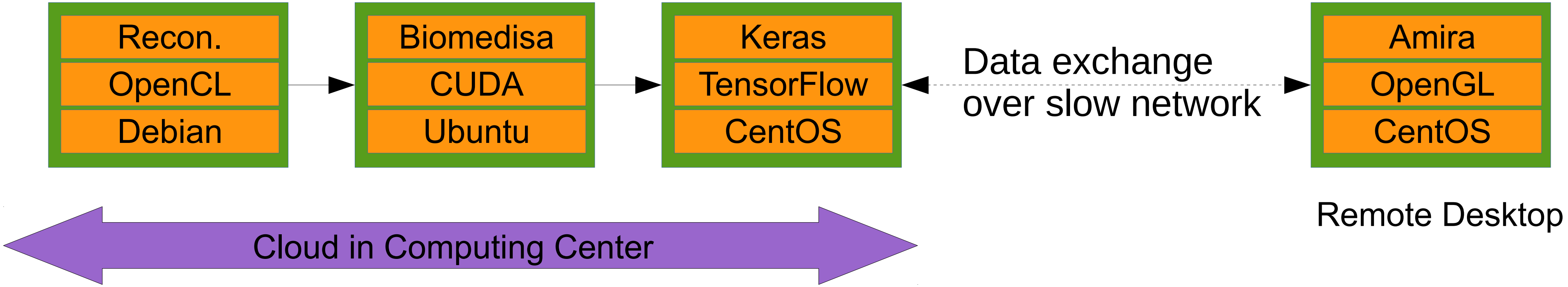
Interactive Phase
High-quality server-side rendering



volume name: /sara current texture res: 4096 px
512 1024 2048 4096
zoomed out

Complex client-side applications using new container features

- Support Linux containers is provided on Windows and OS X
- Run Desktop Applications in the container



Recon.
OpenCL
Debian

Biomedisa
CUDA
Ubuntu

Keras
TensorFlow
CentOS

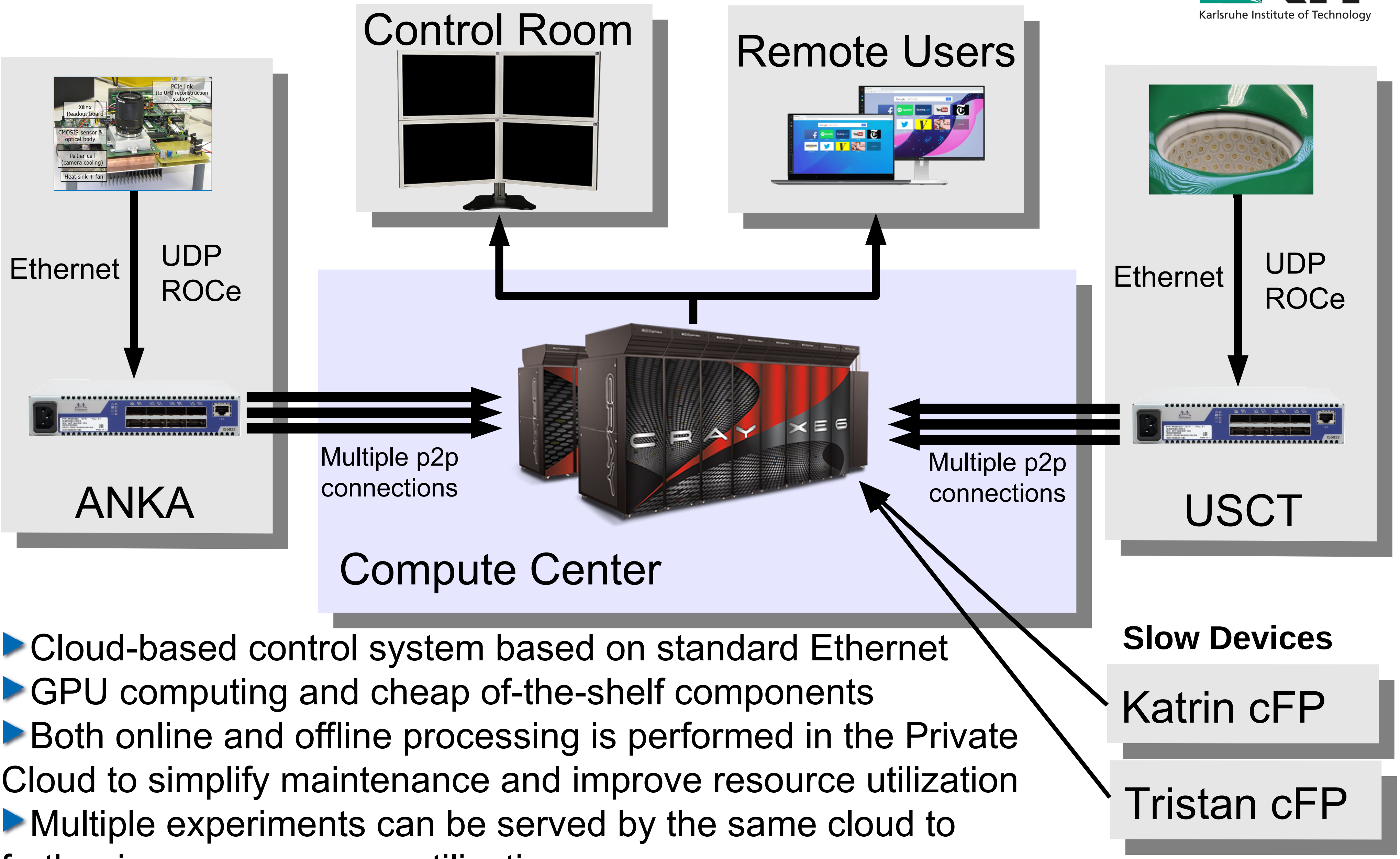
Data exchange over slow network

Amira
OpenGL
CentOS

Remote Desktop

Cloud in Computing Center

UFO Cloud Platform



PCIe link (to UFO reconstruction station)
Xilinx Readout board
CMOSIS sensor & optical body
Peltier cell (camera cooling)
Heat sink + fan

Control Room

Remote Users

Ethernet

UDP
ROCe

ANKA

Multiple p2p connections

Compute Center

Ethernet

UDP
ROCe

USCT

Multiple p2p connections

Slow Devices

Katrin cFP

Tristan cFP

- Cloud-based control system based on standard Ethernet
- GPU computing and cheap of-the-shelf components
- Both online and offline processing is performed in the Private Cloud to simplify maintenance and improve resource utilization
- Multiple experiments can be served by the same cloud to further improve resource utilization
- Distributed data processing framework based on the Containers and RDMA

Hardware support: What we need?

- **Ethernet-based register access**
  - Security-mechanism to prevent unauthorized-access
  - Protocol to read/write/modify registers over Ethernet
  - Batched reads/writes
- **High-speed data-streaming over the Ethernet**
  - ROCe v.2 extension support
  - Send data using multiple Ethernet ports
  - Multi-channel communication to many processing nodes
- **UFO Cloud Integration**
  - Interface to Cloud Master for requesting additional processing nodes or releasing not used ones
  - Extensive buffering capabilities to hold data until cluster is re-arranged to the increased load
  - Data and network packet awareness: for instance, camera frames are distributed between multiple nodes, but each node always gets a full frame

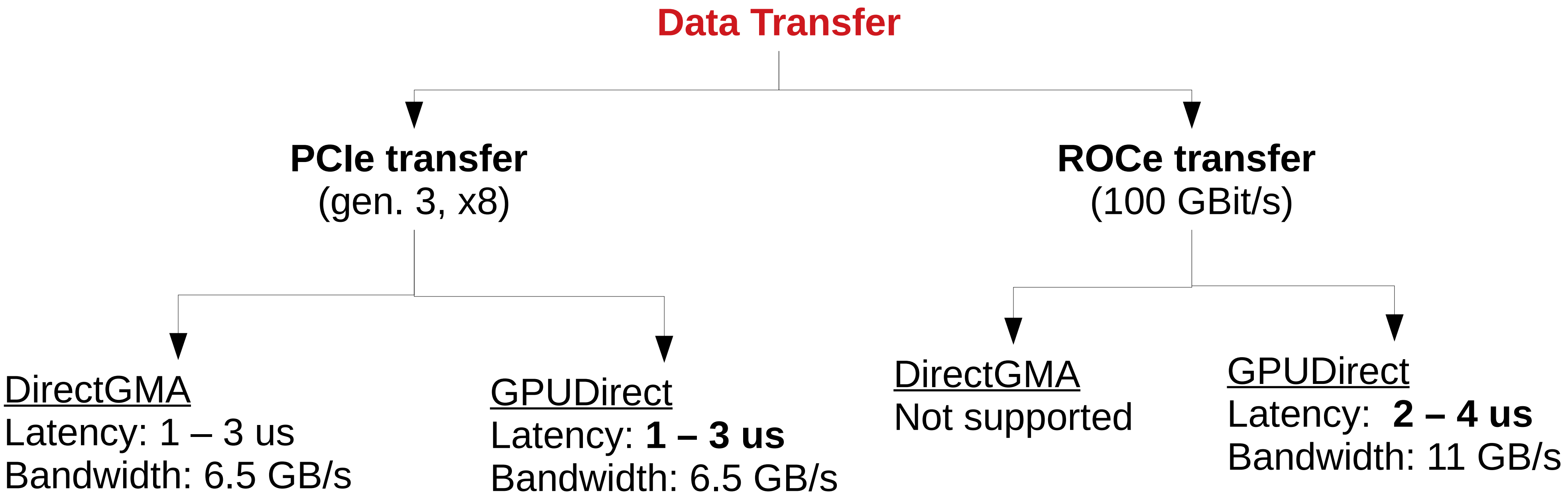
Summary: scale is the key

Major and ever growing investments are required to build high-speed DAQ systems. Running software on a common and shared platform may reduce costs for each participant and allows redistribute resources and adapt to spikes in the data taking.

- **Standard Ethernet is used everywhere**
  - Move all IT infrastructure to Computing Center. No additional equipment in the Lab, only an Ethernet switch.
  - No Linux driver required, easier debugging, etc.
- **Improved Scalability and Performance**
  - Low latency UDP/ROCe between detector and processing
  - No bottleneck because of Master server, i.e. better scalability
- **DAQ Software → Containers in the Cloud**
  - Simplified IT administration
  - Improved Data security and High Availability of the service
- **Institute-wide infrastructure for all DAQ and Analysis workloads**
  - Re-use resources, reduce movements of data
  - Short experiment can sustain very high rates as a significant share of institute resources can be temporarily allocated

Expected Latency

Total: **3 – 7** us (PCIe) vs. **4 – 8** us (ROCe)



Data Transfer

PCIe transfer
(gen. 3, x8)

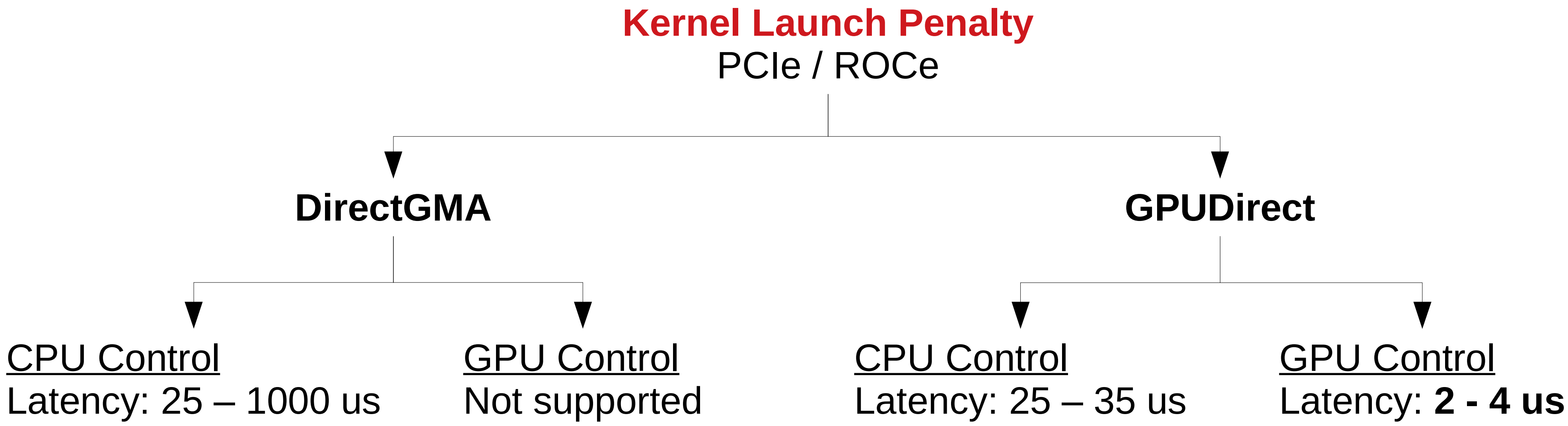
DirectGMA
Latency: 1 – 3 us
Bandwidth: 6.5 GB/s

GPUDirect
Latency: **1 – 3 us**
Bandwidth: 6.5 GB/s

ROCe transfer
(100 GBit/s)

DirectGMA
Not supported

GPUDirect
Latency: **2 – 4 us**
Bandwidth: 11 GB/s



Kernel Launch Penalty
PCIe / ROCe

DirectGMA

CPU Control
Latency: 25 – 1000 us

GPU Control
Not supported

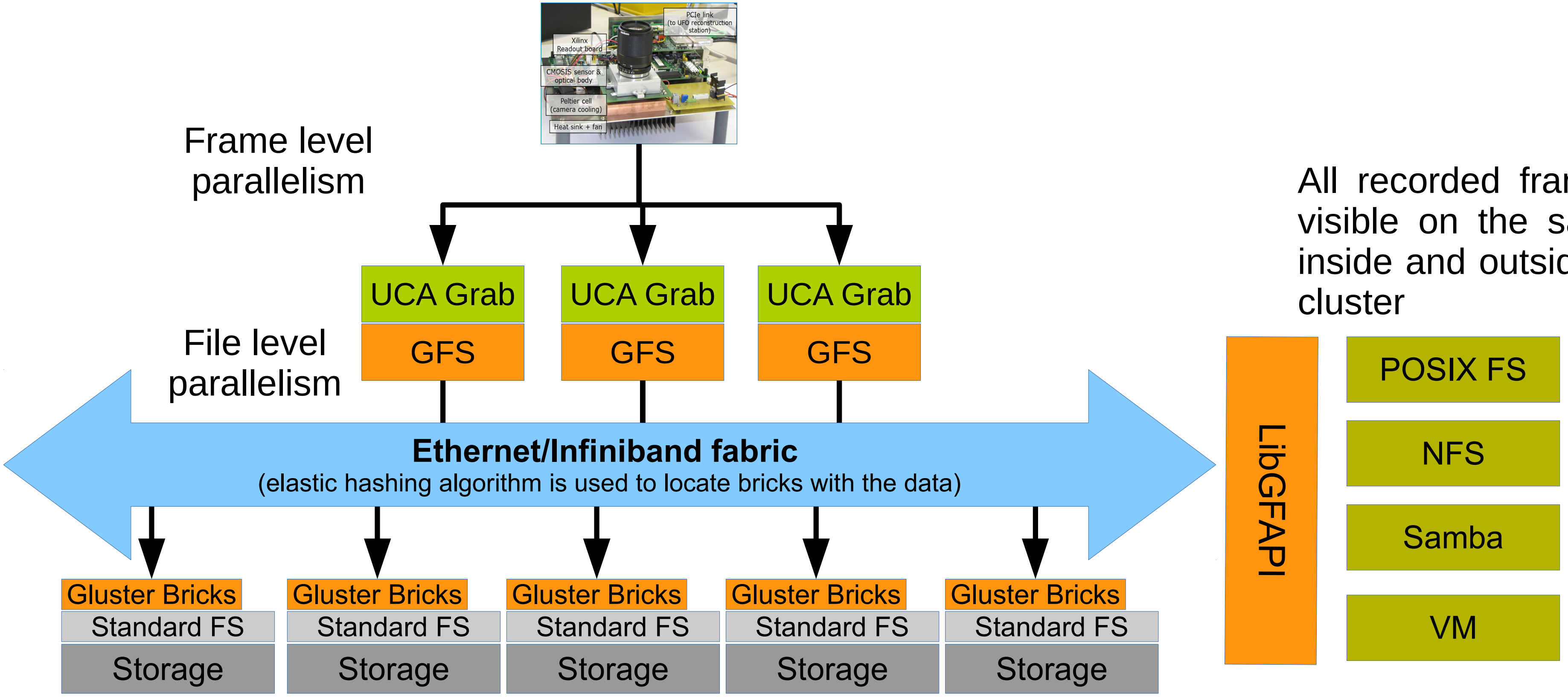
GPUDirect

CPU Control
Latency: 25 – 35 us

GPU Control
Latency: **2 - 4 us**

High Sped Storage with GlusterFS

- **Easy:** Runs in container on top of the Cloud Platform
- **Accessible:** POSIX FS in Containers; NFS/Samba – remotely
- **Fast**: Scalable to 1000 bricks as there is no metadata server, only P2P connections between clients and bricks
- **Secure:** Replication and geo-replication is supported


PCIe link
(to UFO reconstruction station)
Xilinx Readout board
CMOSIS sensor & optical body
Peltier cell (camera cooling)
Heat sink + fan

Frame level parallelism

UCA Grab | UCA Grab | UCA Grab

GFS | GFS | GFS

File level parallelism

Ethernet/Infiniband fabric
(elastic hashing algorithm is used to locate bricks with the data)

Gluster Bricks | Gluster Bricks | Gluster Bricks | Gluster Bricks | Gluster Bricks

Standard FS | Standard FS | Standard FS | Standard FS | Standard FS

Storage | Storage | Storage | Storage | Storage

All recorded frames are visible on the same FS inside and outside of the cluster

LibGFAPI

POSIX FS

NFS

Samba

VM

Up to 1000 bricks to boost performance