HELMHOLTZ

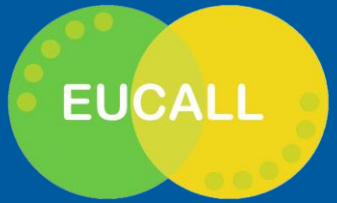
HZDR
HELMHOLTZ ZENTRUM DRESDEN ROSSENDORF

ScaDS DRESDEN LEIPZIG

TECHNISCHE UNIVERSITÄT DRESDEN

RESEARCH FOR GRAND CHALLENGES

ONLINE DATA PROCESSING GPU VS. FPGA

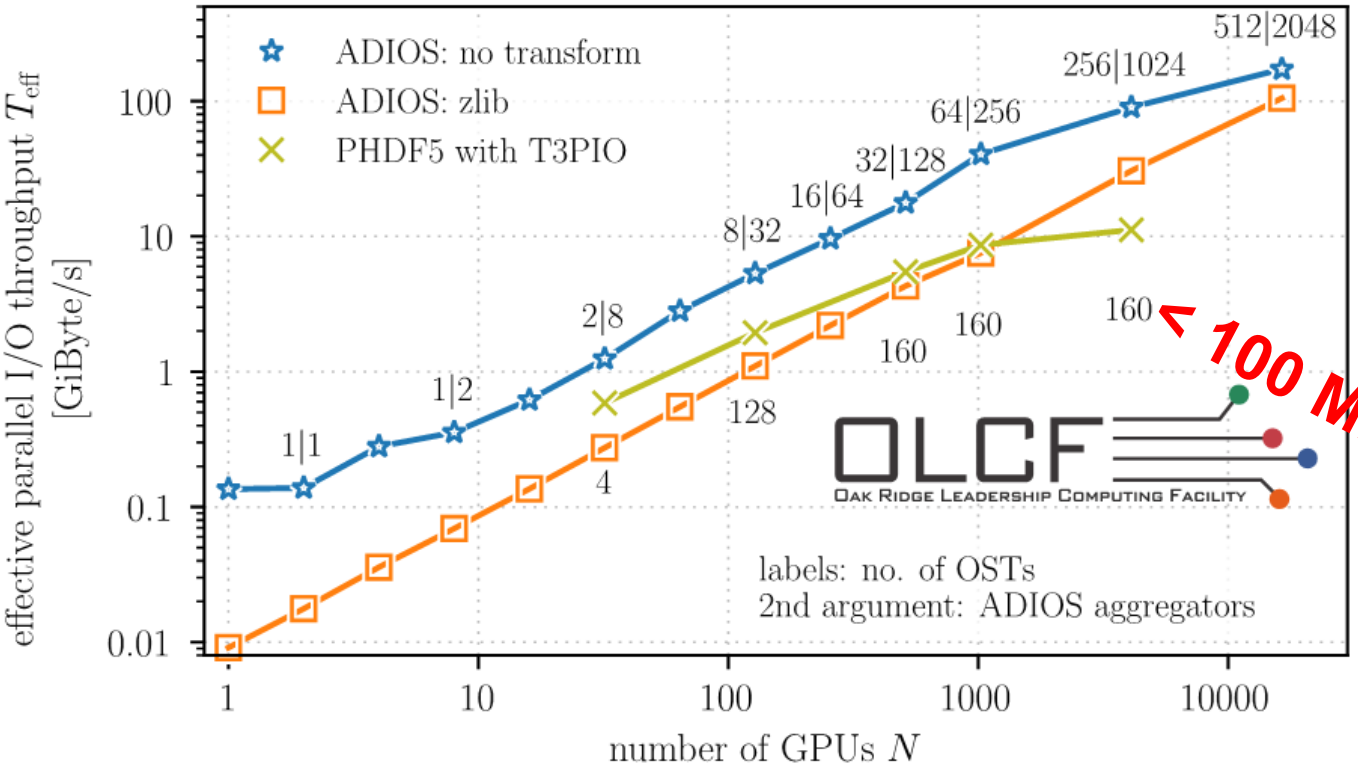


Michael Bussmann
Helmholtz-Zentrum Dresden – Rossendorf

HOW DO YOU BEST TRANSPORT DATA? (ABRIDGED VERSION)

DON’T

DATA NEEDS TO BE STORED FOR LATER ANALYSIS

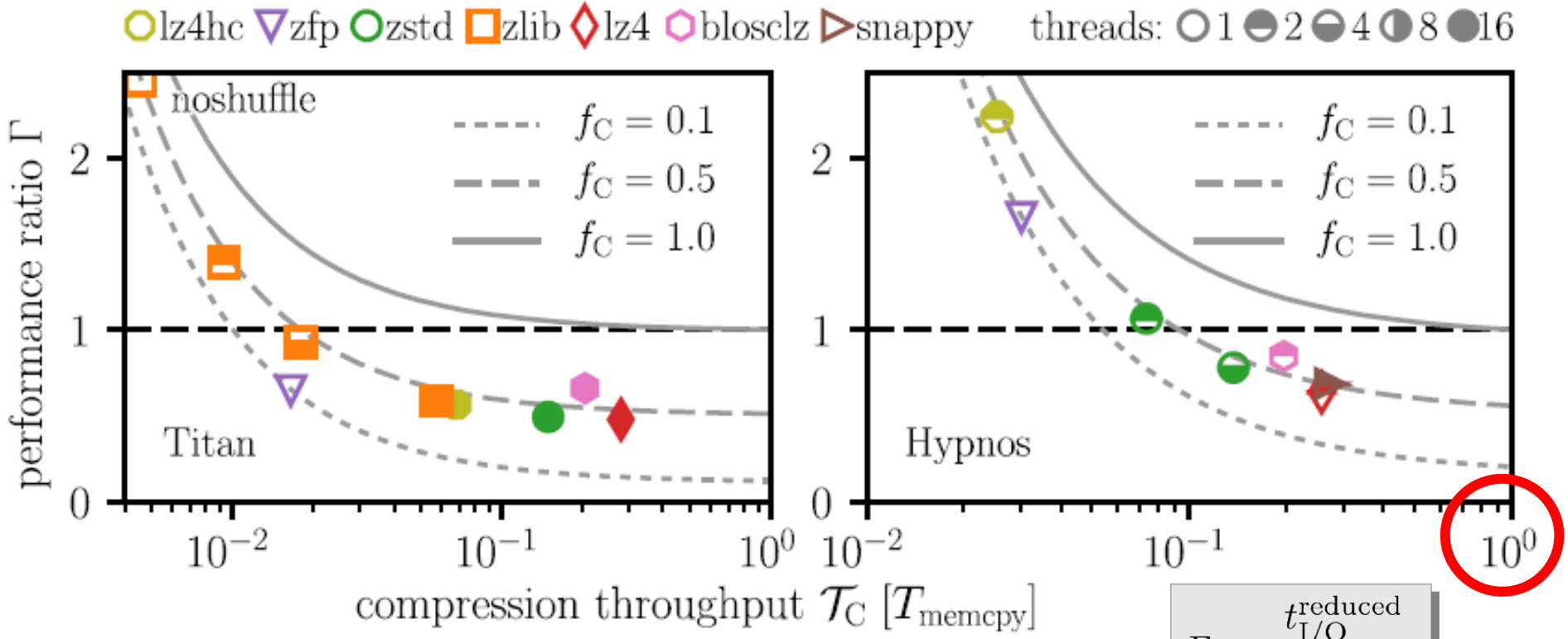


- I/O is shared
- BW hierarchy
- Analysis needs BW

< 100 MB / (Node Second)

A Huebl, …, M Bussmann, High Performance Computing 2 (2017)

COMPRESSION IS NEEDED, BUT NEEDS TIME



$$\Gamma \equiv \frac{t_{I/O}^{reduced}}{t_{I/O}^{unreduced}}$$

A Huebl, …, M Bussmann, High Performance Computing 2 (2017)

I/O ON THE 3RD FASTEST SUPERCOMPUTER IN THE WORLD

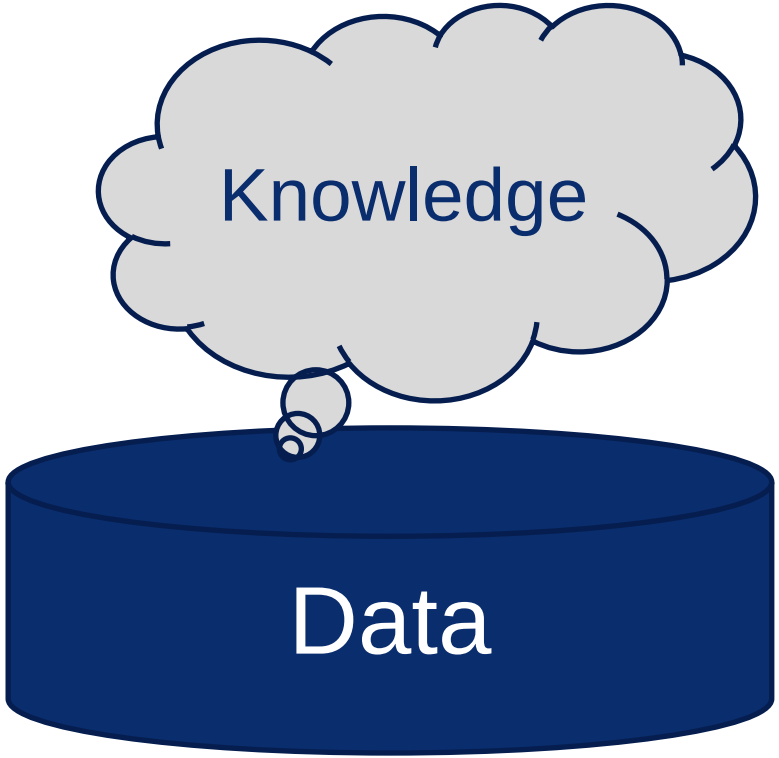
PRACE

"Overall , this is an outstanding proposal. […]
The HPC resource request are appropriate.
The PIs should try to reduce the data requirements
***and try to find a solution that is technically possible for CSCS**."*

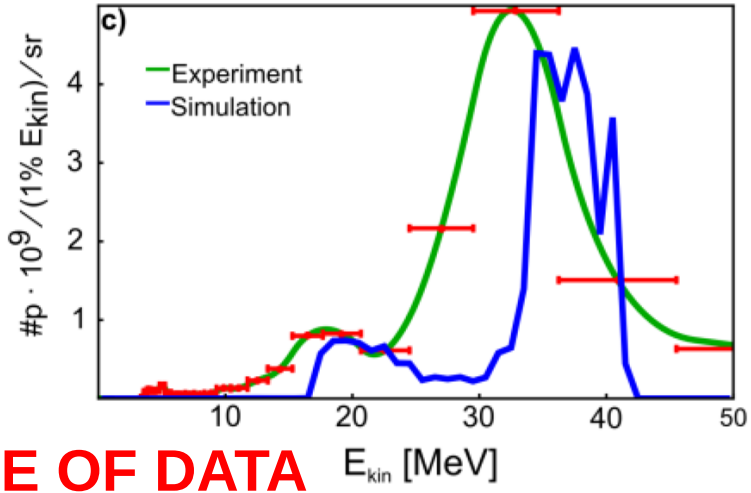
1 Pbyte in < 1 week

"The TNG simulations produced more than 500 Terabyte […] The full analysis will keep the participating scientists busy for many years to come"

BIG DATA IS ALL ABOUT THROWING STUFF AWAY



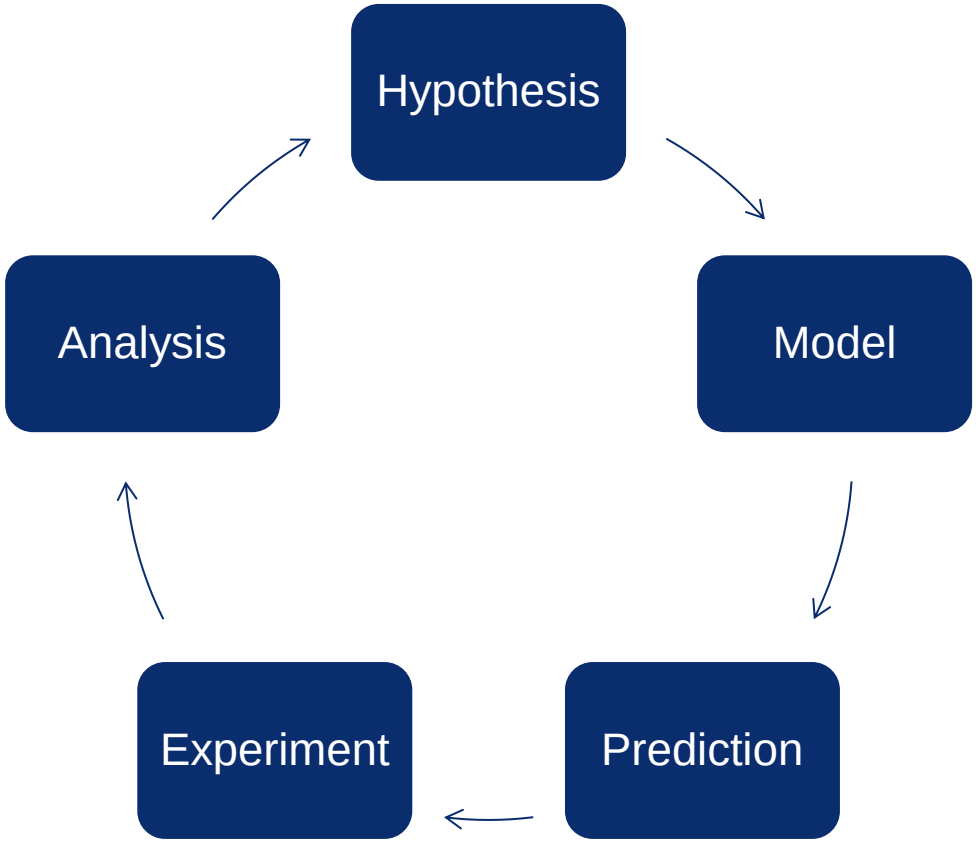
- The amount of scientific data grows
- As do data rates
- Scientists need to understand their data



To be published in Nature Comm.

4 PBYTE OF DATA

ALL SCIENCE IS DATA DRIVEN & DATA IS GROWING FAST

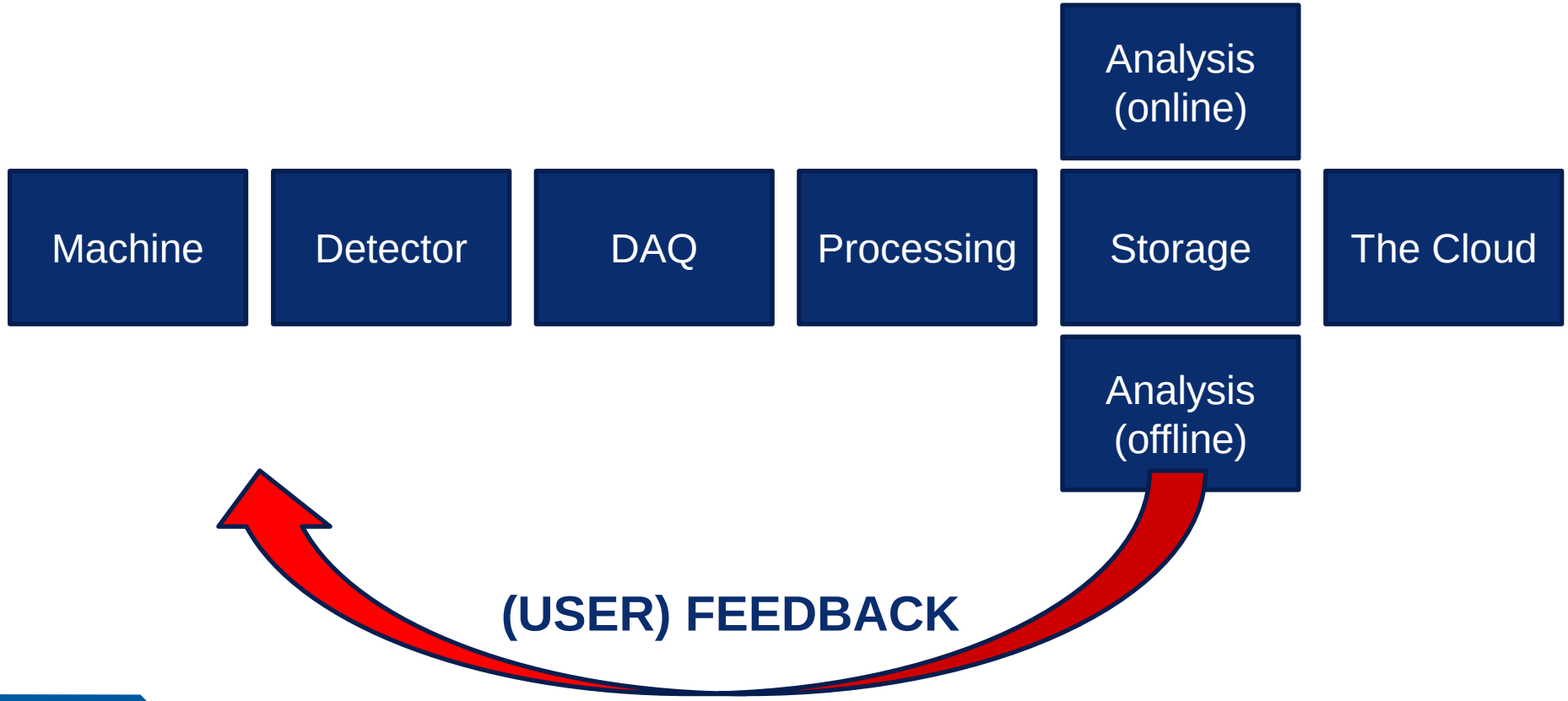


The scientific workflow stands as it is.

But it has become HPC-dependend.

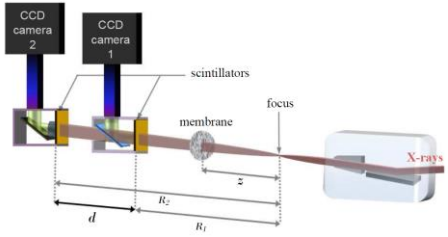
We have (not yet) embraced this!

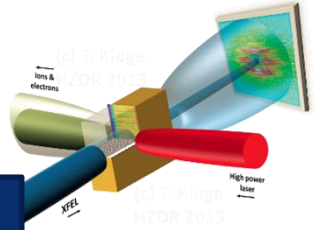
TYPICAL CHAIN OF INFORMATION



OUR INVOLVEMENT

WAVEFRONT RECONSTRUCTION (WITH ESRF + EUXFEL)



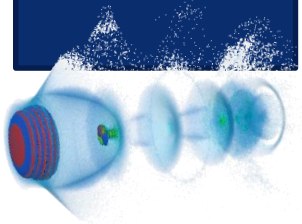


CLOUD COMPUTING DATA REPOS (EUPRAXIA, HGF)

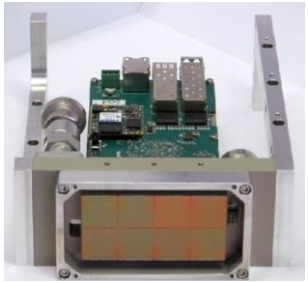
Machine | Detector | DAQ | Processing | Storage | The Cloud

Analysis (online)

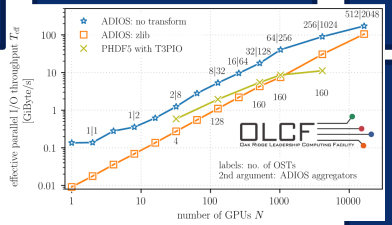
Analysis (offline)



FEMTOSECOND DIAGNOSTICS (EUPRAXIA, HGF)

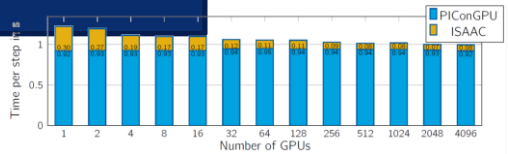


PHOTON COUNTING (WITH PSI)



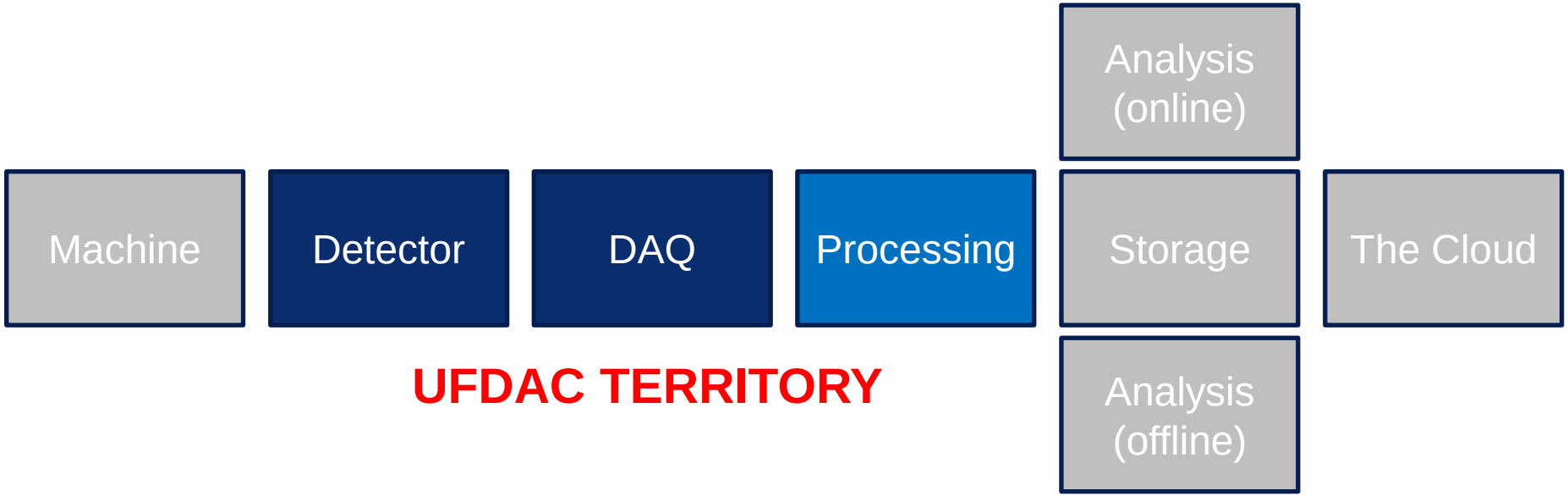
FAST I/O, STAGING (WITH ORNL)

OPEN FORMATS (EUXFEL,ELI,LBNL,…)



IN-SITU VISUALIZATION, STEERING & FEEDBACK (SCADS, TU DRESDEN)

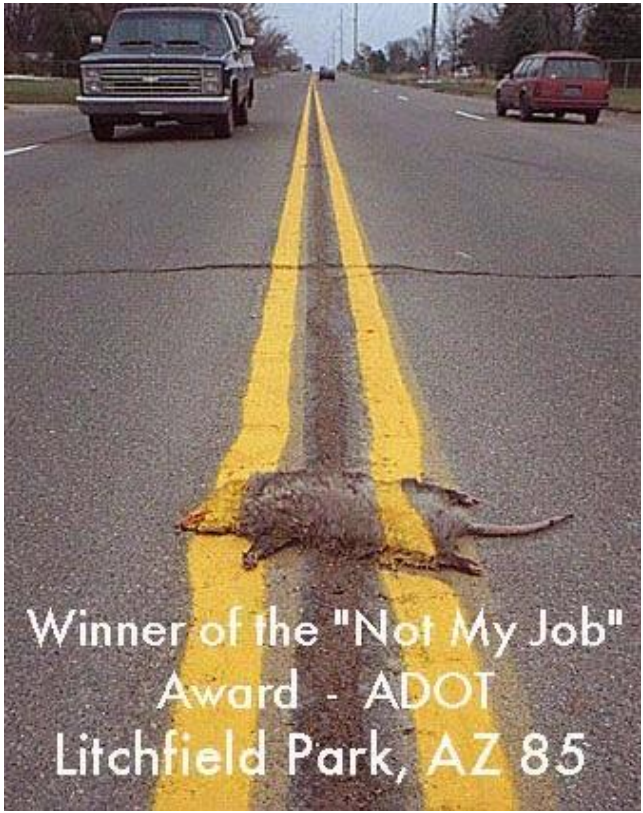
TYPICAL CHAIN OF INFORMATION



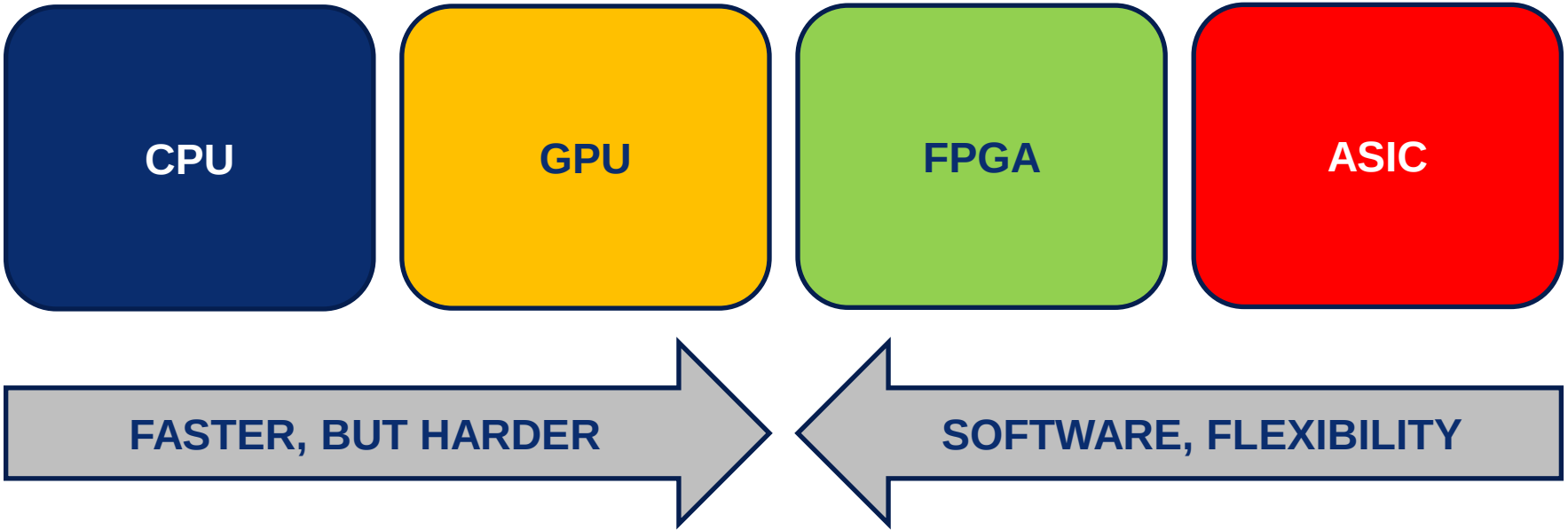
RESPONSIBILITIES AND THE DATA LANDSLIDE

Machine | Detector | DAQ | Processing

UFDAC TERRITORY

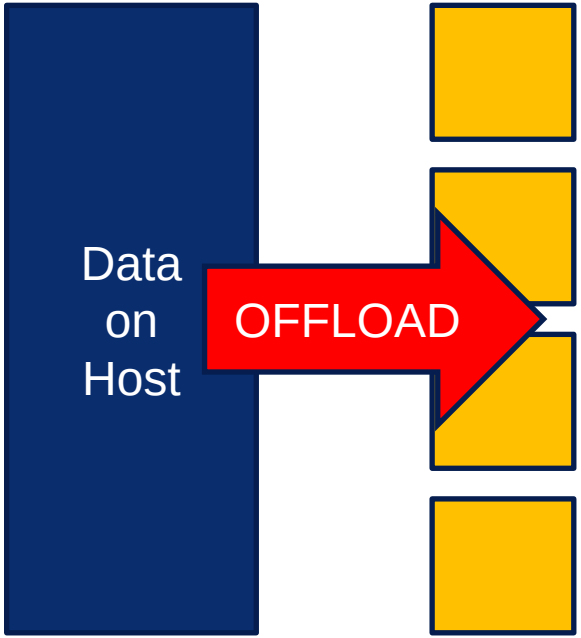


IT’S NOT JUST ENGINEERING

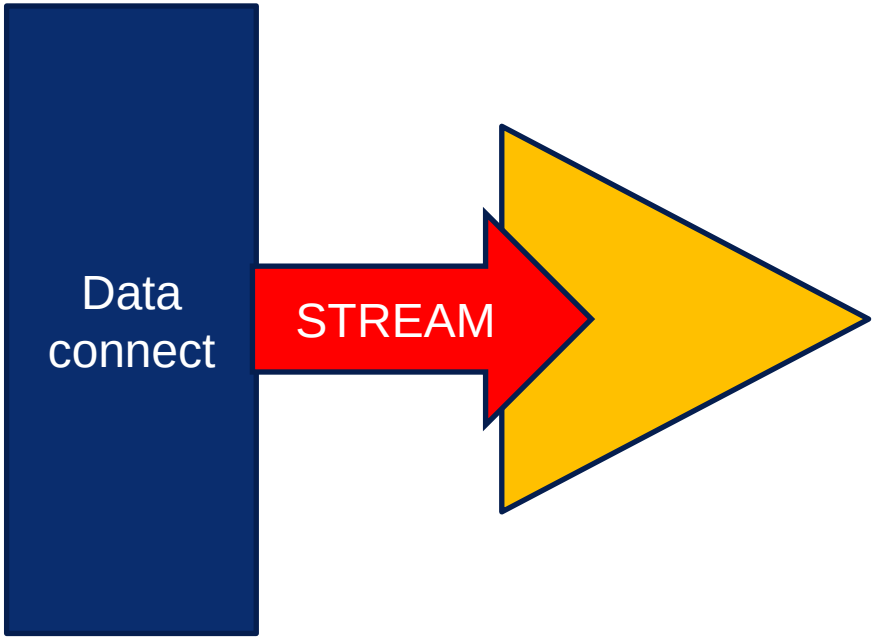


GPUS VS. FPGAS – MODE OF DATA TRANSPORT

GPU (HIDE LATENCY)



FPGA (OPTIMIZE THROUGHPUT)



USE GPU (R)DMA WISELY!

GPUS VS. FPGAS – EFFORT, DEBUGGING, PORTABILITY, …

GPU

- CUDA / AMD HIP
- OpenCL
- OpenACC
- Others

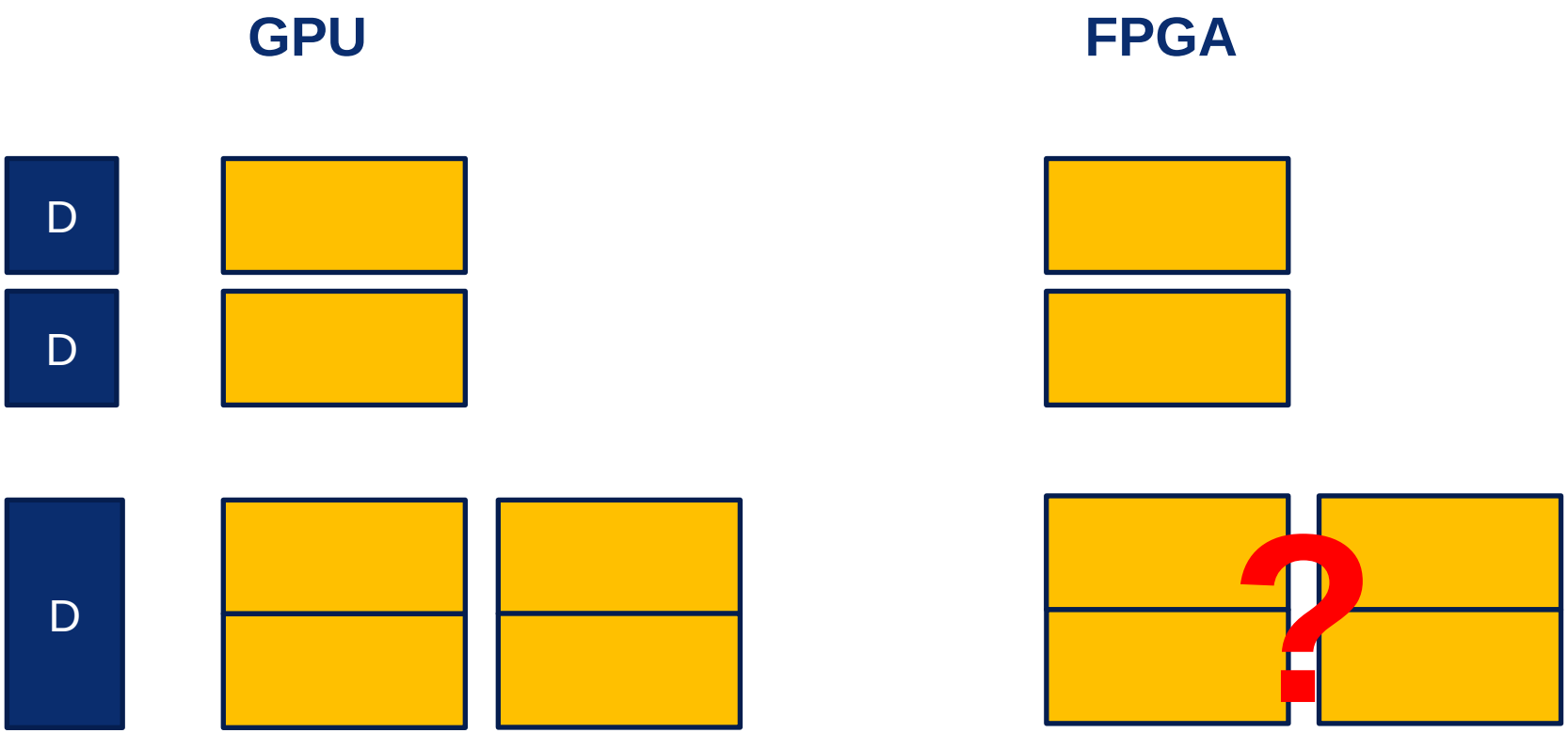
FPGA

- VHDL
- OpenCL
- Others

- UDP, MPI, TCP
- Ethernet, Infiniband
- PCIe, NVLink

VERY LIMITED PERFORMANCE PORTABILITY
DEBUGGING NOT EASY
EFFORT DEPENDS ON MINDSET & EXPERIENCE
FAST PACE OF DEVELOPMENT

GPUS VS. FPGAS – SCALABILITY (COMPUTE COMPLEXITY D^2)



ALWAYS CHUNK DATA

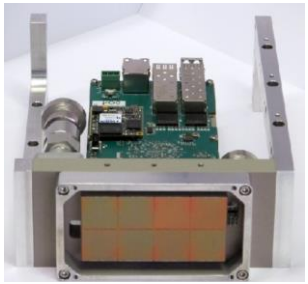
GPUS VS. FPGAS – DOING IT WRONG (~ OK FOR DAQ)

Stream in

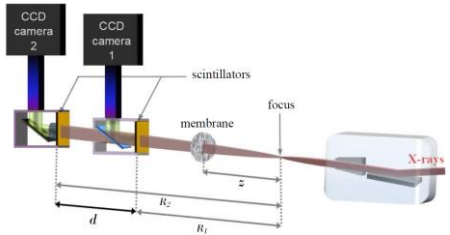
DATA PROCESSING

Stream out

PHOTON COUNTING

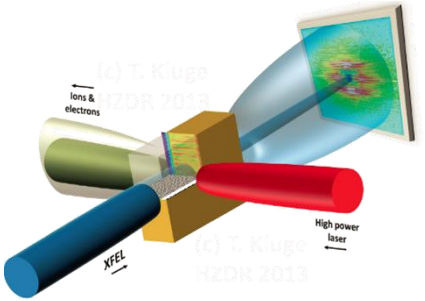


+



WAVEFRONT RECONSTRUCTION

+

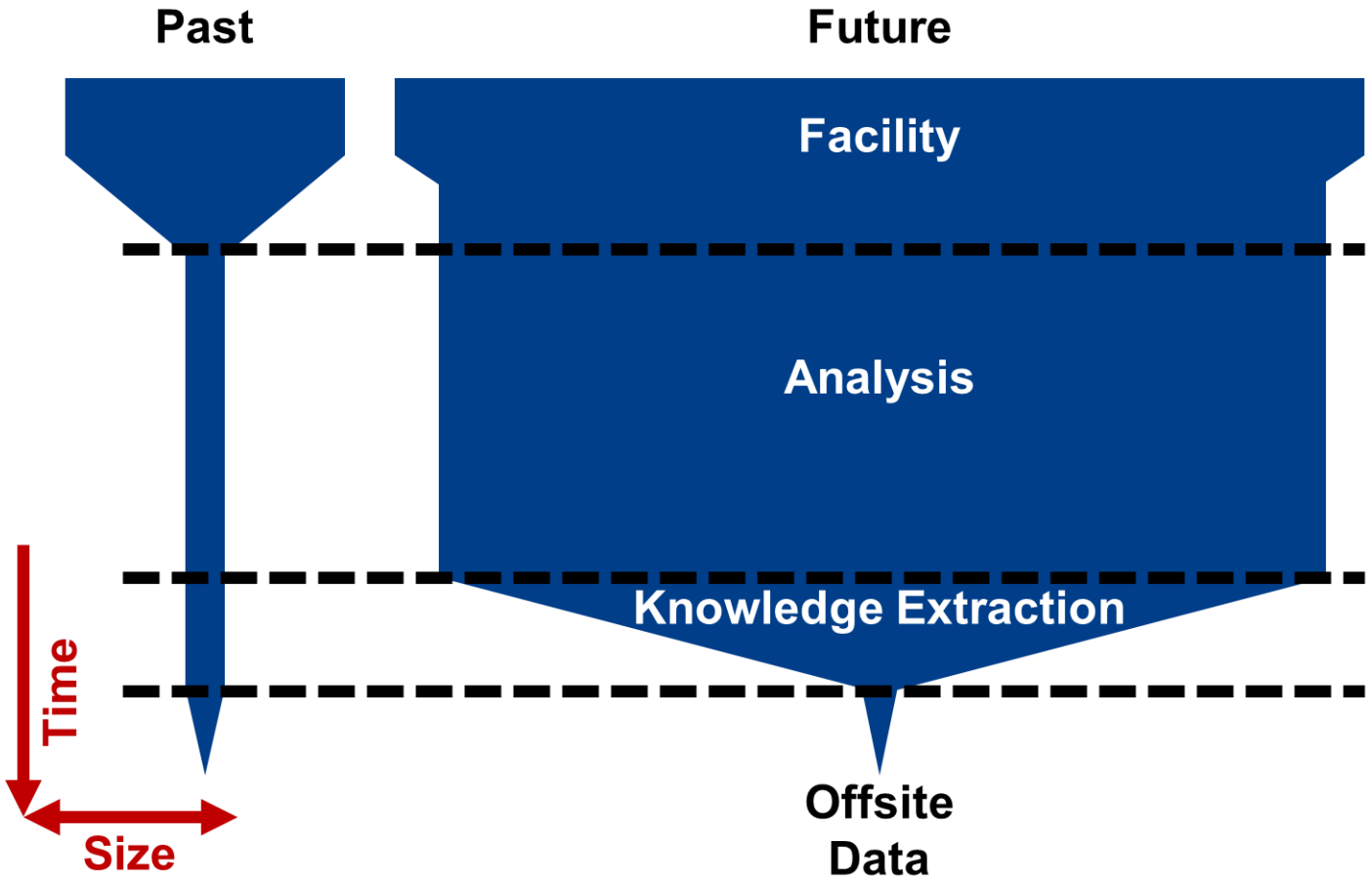


ONLINE ANALYSIS

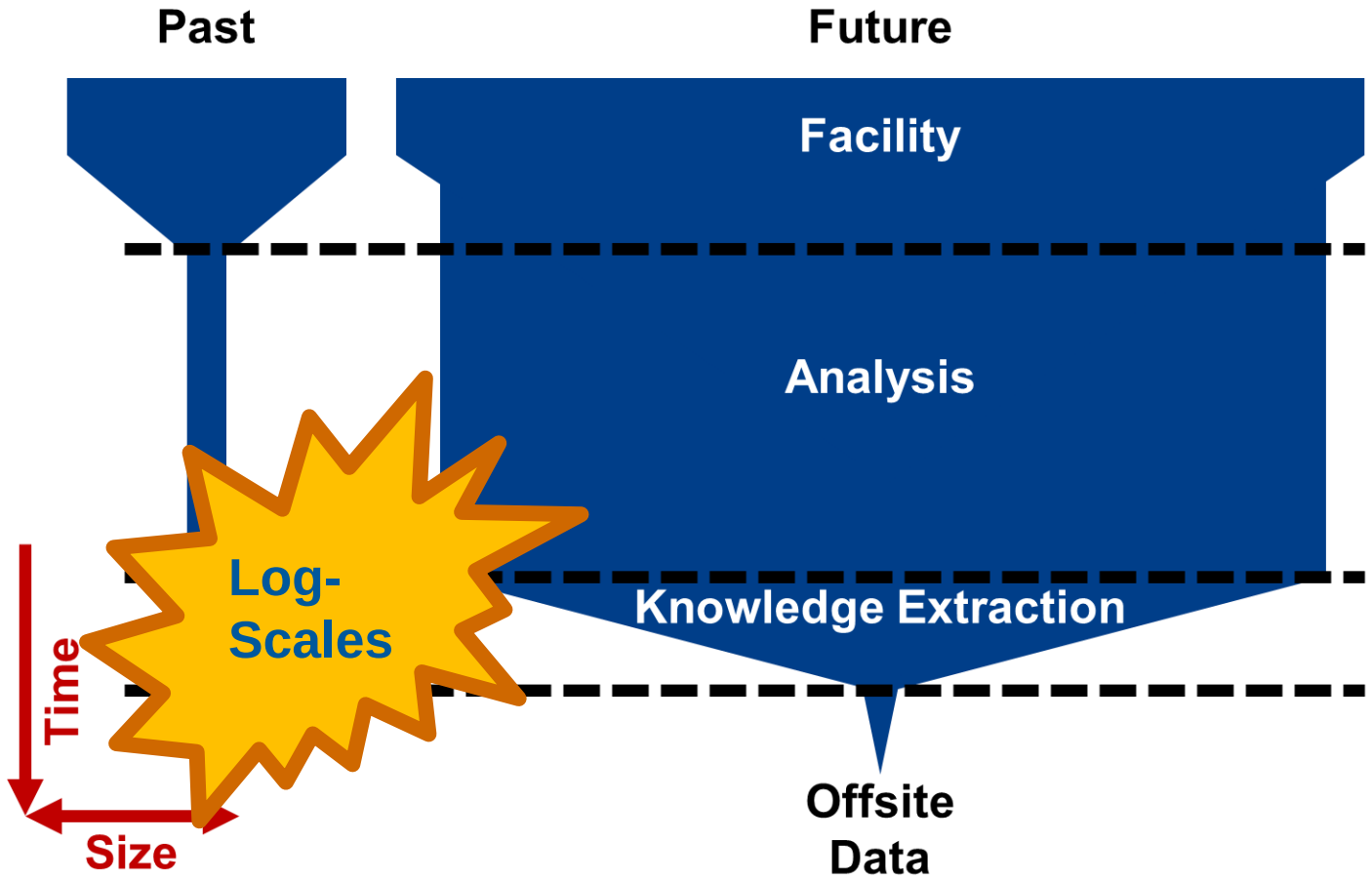
GPUS VS. FPGAS VS. CPUS VS. ASICS

	CPU	GPU	FPGA	ASIC
MODE	BATCH,STREAM	BATCH	STREAM,BATCH	STREAM
LATENCY	LARGE	OFFLOAD	~ ZERO	~ ZERO
BW	MEDIUM	HIGH	HIGH	HIGH
EFFORT	LOW	MEDIUM	HIGH	HIGH
DEBUGGING	EASY	HARD	HARD	HARD
PORTABILITY	HIGH	LOW	LOW	LOW
LIFETIME	HIGH	LOW	MEDIUM	HIGH
SCALABILITY	HIGH	MEDIUM	LOW	LOW

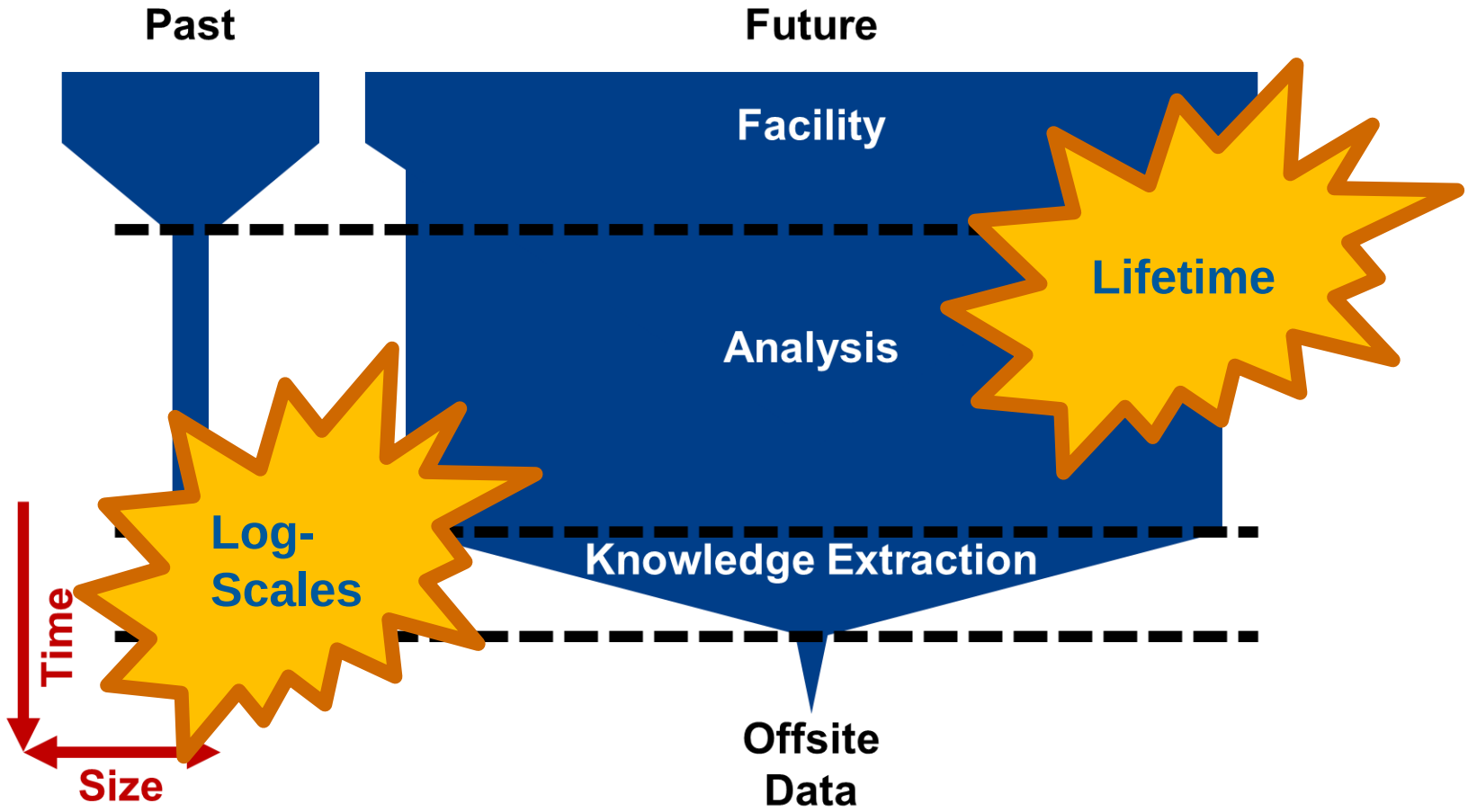
THE PROBLEM WITH IMAGES



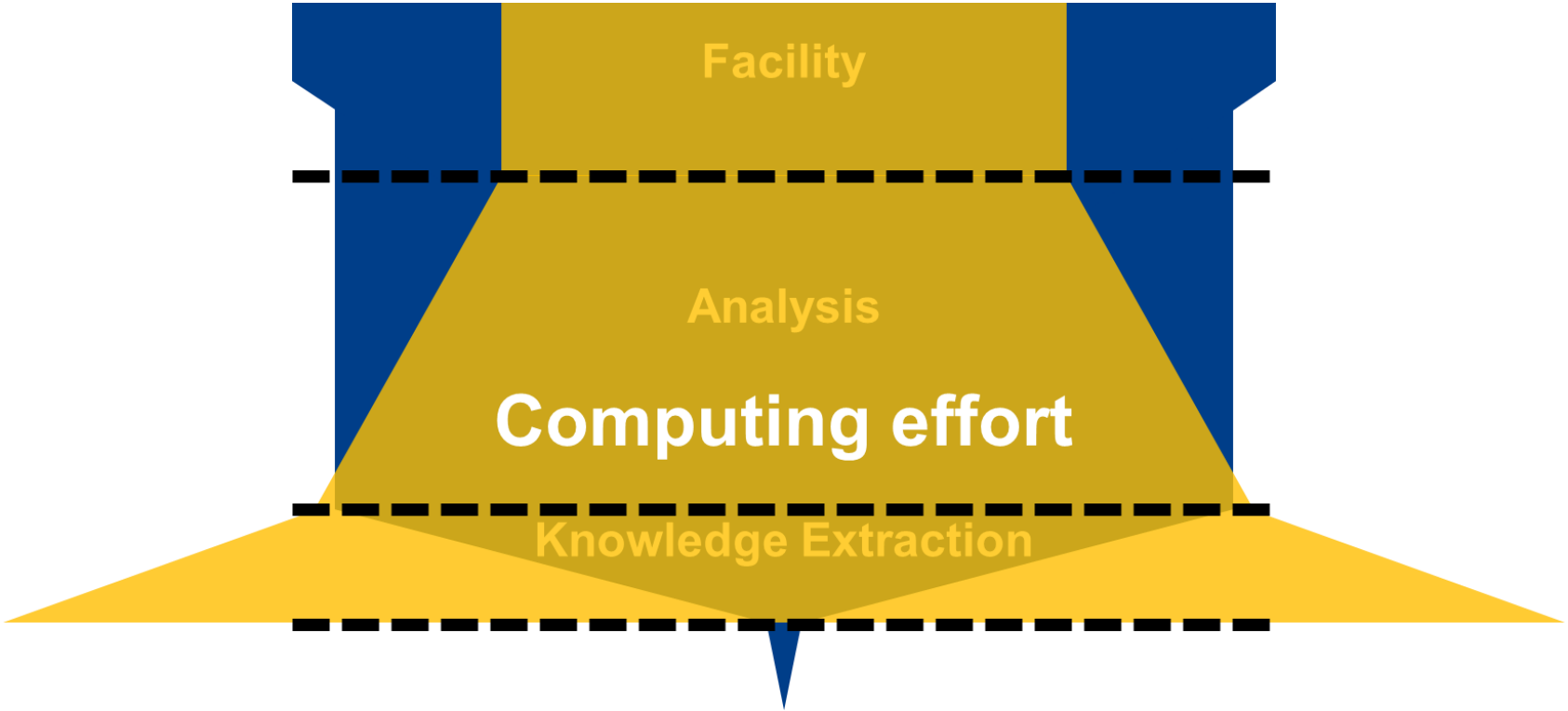
THE PROBLEM WITH IMAGES



THE PROBLEM WITH IMAGES



MORE COMPUTING FOR LESS DATA



WHAT WE NEED – SOFTWARE ENGINEERING

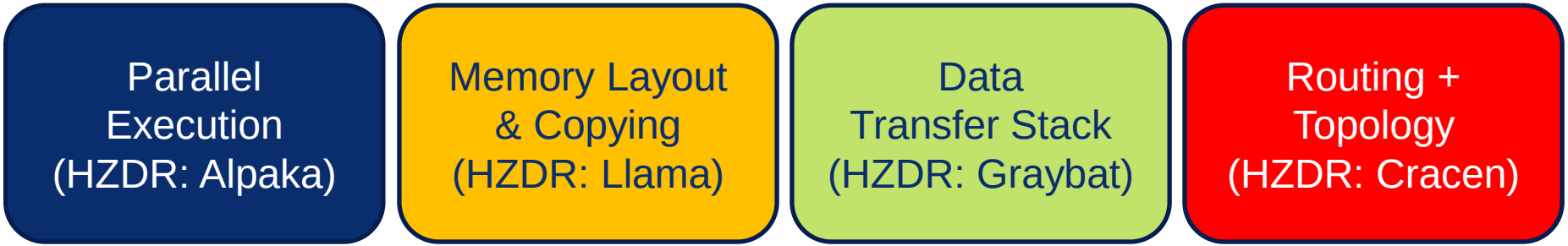
- Fast production cycles (A new architecture every year)
- Streaming translates to „not my job“, need integration (interfaces!)
- Reduce data movement to the maximum

FROM „ASIC MINDSET“ TO „CPU MINDSET“

HARDWARE ENGINEERING IS SOFTWARE ENGINEERING

WHAT WE NEED – PORTABILITY & FLEXIBILITY

- Avoid Vendor or Release Lock-in
- Choose best Hardware for the Job (Scalability, Energy, Prize)
- Units of Responsibility ≠ Software Interfaces



WHAT WE NEED – SCALABILITY (PROBLEM: USERS)

- Human in the Loop is the main problem!
- Algorithms have nonlinear compute dependency
- Data transfer causes problems: Throughput, Load balancing, Resilience

Memory (I/O) Staging (HZDR/ORNL)	Virtualization, Containers (HZDR/EUXFEL)	In-situ Visualization, Feedback (HZDR,TUD)	Just-in-time Compilation (HZDR+CERN)

WHAT WE WANT FROM VENDORS

- Optimize for Throughput, NOT for FLOPs/s
- See Data Parallelism + Task Parallelism + Memory Transfer + Staging as one
- Portable Parallel Programming + Intelligent Routing + Configurable Transfer

DON'T ➔ CAN