HELMHOLTZ ASSOCIATION

Work package 2:

Real-time Data Processing

Andreas Kopmann (KIT)

HDRI / PanData Workshop 27./28.2.2012, DESY

DESY · GSI · HZB Helmholtz Zentrum Berlin · Helmholtz-Zentrum Geesthacht Zentrum für Material- und Küstenforschung · KIT Karlsruhe Institute of Technology

Motivation

1. DAQ for pixel detectors
   Handling high-throughput

2. Monitoring of “data streams”
   Fast turn-around time

3. Data quality checks
   Efficient use of beam time

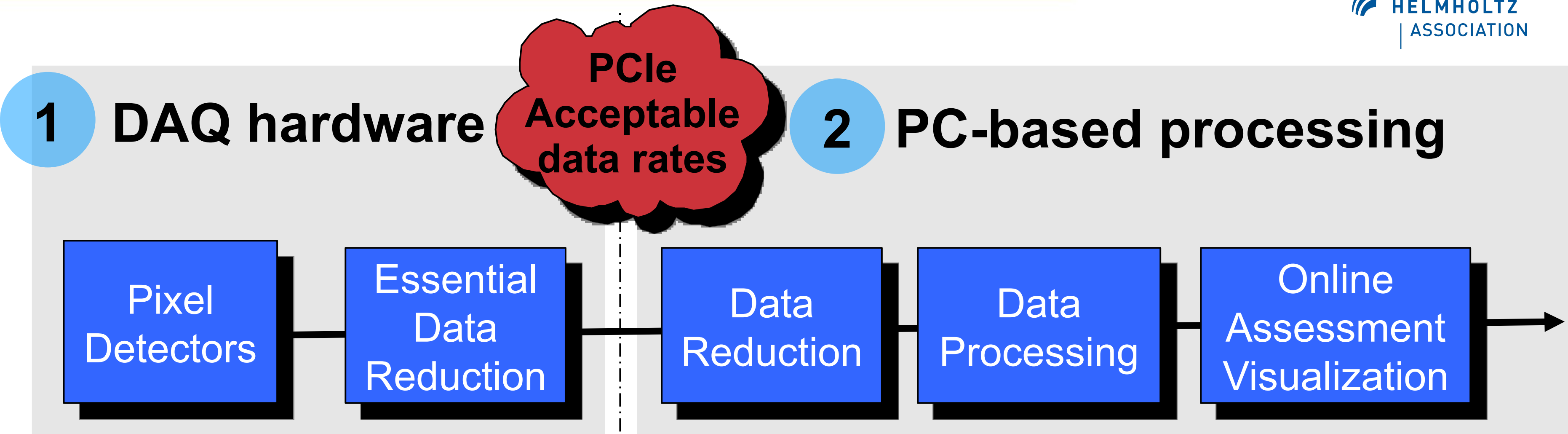
4. Online data reduction
   Save disk space

Pilatus 6M
6 M pixels
12 Hz frame rate
180 MB/s
630 GB/h
DS: 10 - 30 GB

pco.edge
5.3 M pixels
100 Hz frame rate
750 MB/s
2,5 TB/h
DS: 10 - 30 GB

Real-time Data Processing: Data Life-cycle

HELMHOLTZ ASSOCIATION



Technologies

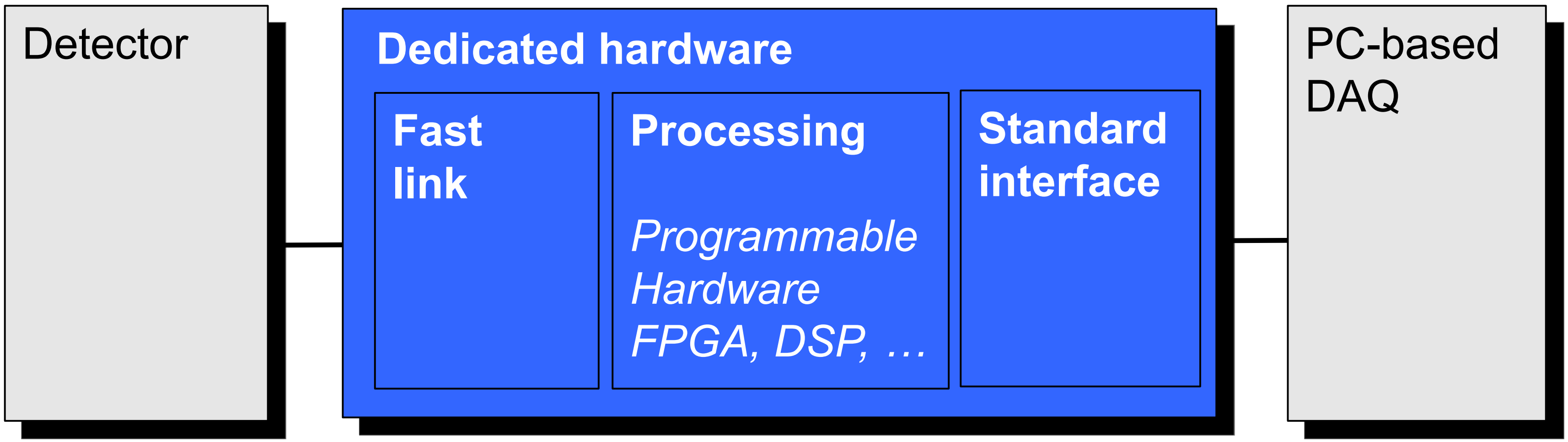
Programmable hardware	**„Parallel Computing“:**
FPGA, DSP, embedded GPUs	PC + Coprocessing / GPU

Tasks

Selected commercial hardware Standardized custom hardware	Hardware independant programming (e.g. OpenCL) Library of common algorithms

Task 1: Dedicated Hardware

General concept:



- For highest data rates, off-the-shelf PC technology is too slow
- Necessary data reduction can be realized by fast links combined with programmable hardware

Solution: **Standardized PNI hardware box**

  - Will avoid dublication of work and excessive development effort

Task 1: Dedicated Hardware

HELMHOLTZ ASSOCIATION

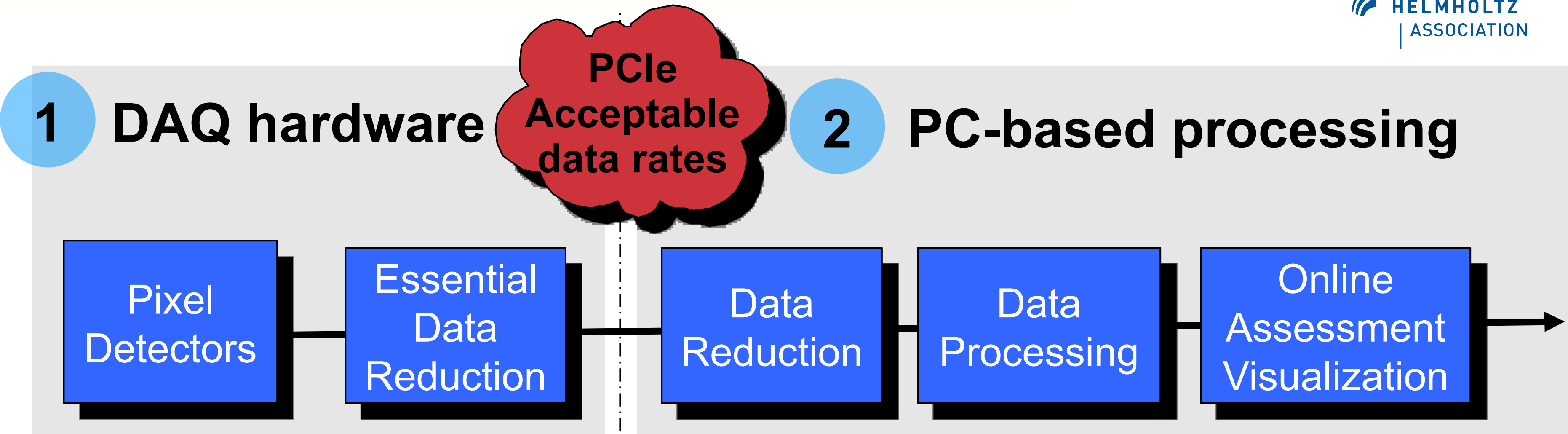
Actions:

- Survey of available solutions for programmable signal processing hardware
- Adopt existing solutions to the needs of PNI or join a running HGF or other development

HGF labs have substantial competence in this field

→ Talk: M Balzer

Real-time Data Processing: Data Life-cycle



Technologies

Programmable hardware

FPGA, DSP, embedded GPUs

„Parallel Computing“:

PC + Coprocessing / GPU

Tasks

Selected commercial hardware
Standardized custom hardware

Hardware independant programming (e.g. OpenCL)
Library of common algorithms

Task 2: Real-time Data Assessment

HELMHOLTZ ASSOCIATION

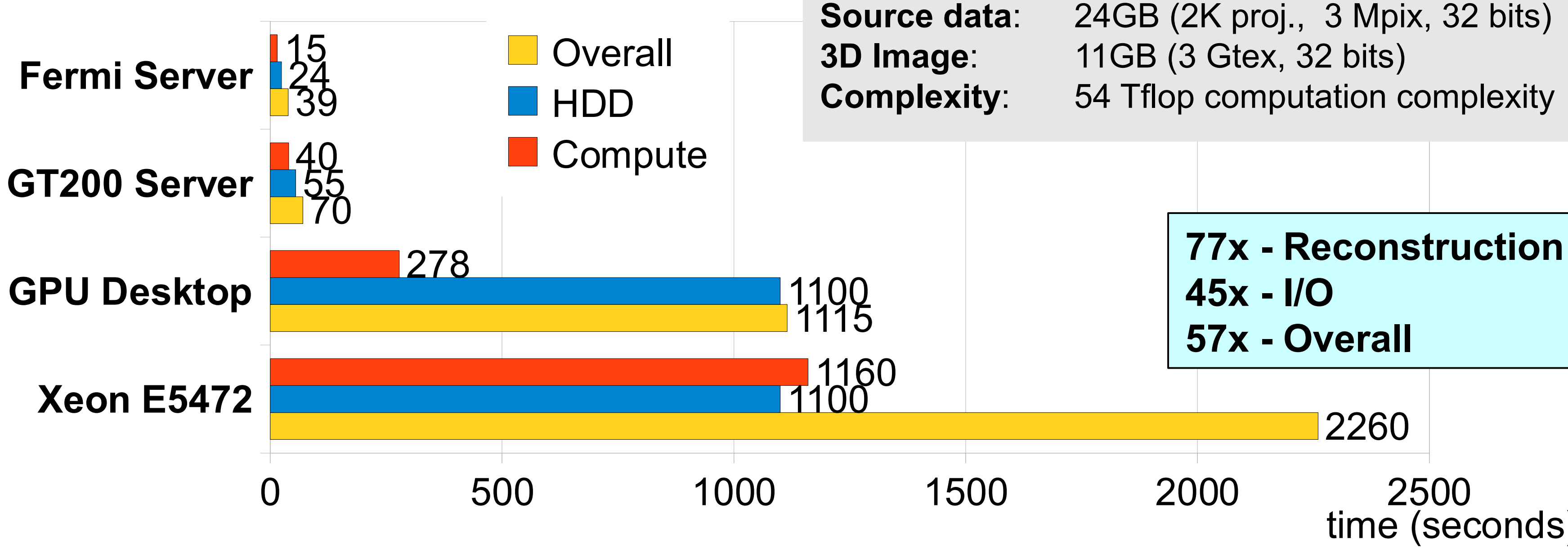
Actions:

- Computation system based on GPU co-processors
- Online tomographic reconstruction
- Prototype adoption of a complete crystallography data flow using the HDRI standard format
- General environment for parallel image processing
    - a) Independent from available hardware (e.g. OpenCL)
    - b) Library of standard algorithms
    - c) Easy adaption to new problems

Status GPU Hardware + Programming

HELMHOLTZ ASSOCIATION

Source data: 24GB (2K proj., 3 Mpix, 32 bits)
3D Image: 11GB (3 Gtex, 32 bits)
Complexity: 54 Tflop computation complexity



77x - Reconstruction
45x - I/O
57x - Overall

	Xeon Server	GPU Desktop	GT200 Server	Fermi Server
Computati onal Units	Xeon E5472 8 core, 3 GHz	**GTX 280** **1 core**	**3 x GTX295** **6 cores**	**6 x GTX580** **6 cores**
CPU	2 x Xeon E5472	Core2 E6300	2 x Xeon E5540	2 x Xeon E5540
Memory	16GB DDR3	4GB DDR2	96G[illegible]	[illegible]
HDD	WDC5000AACS	WDC5000AACS	**2 x** [illegible]	[illegible]

→ Talk: S Chilingaryan

Parallel Processing Framework

Applications
(E.g. tomographic reconstruction, ...)

Load balancing + management
(CPU ⇔ GPU ⇔ frame grabber / Single ⇔ double precision)

Primitives for image processing
(2d FFT, filters, Radon transform, image conversion, …)

Core functions, hardware access
(data transfer, file storage, camera buffering, ROI, …)

Framework: Basics

HELMHOLTZ ASSOCIATION

Image processing = composition of filter nodes

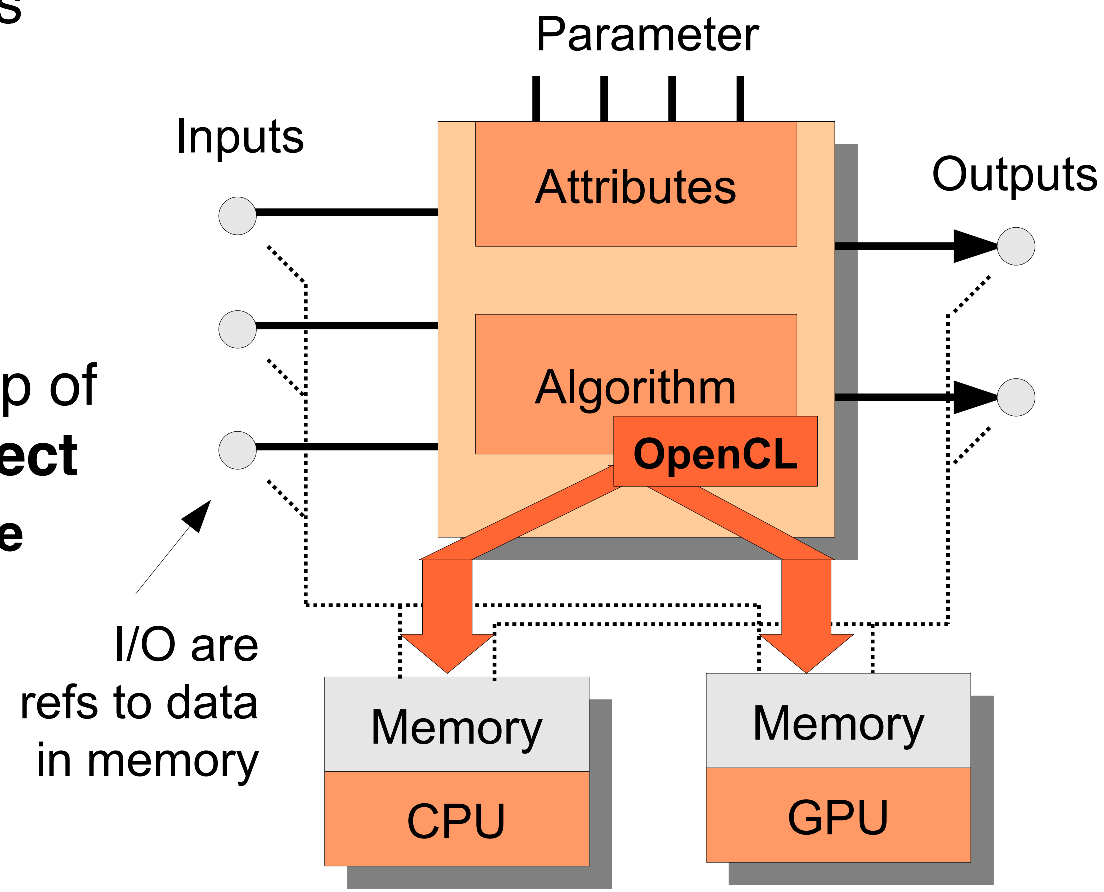
- One thread per node
- Mapping to **CPU or GPU**
- Encourages **recycling** of tested components

Core system is built on top of **OpenCL** and **GLib/GObject**

- **Bindings to any language**

Documented using Sphinx and Gtk-Doc

Filter object = node

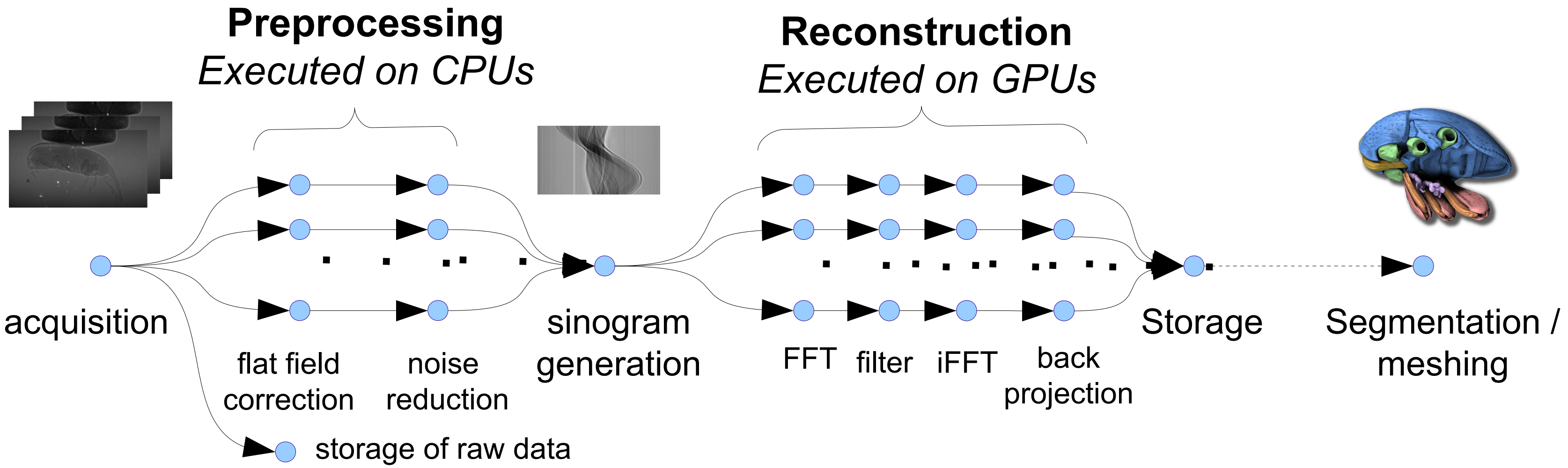


Framework: Application layer

An **application =** composition of filter nodes in a **graph**-like structure

- Interactive development with Python
- Storage as JSON files
- Parallel execution

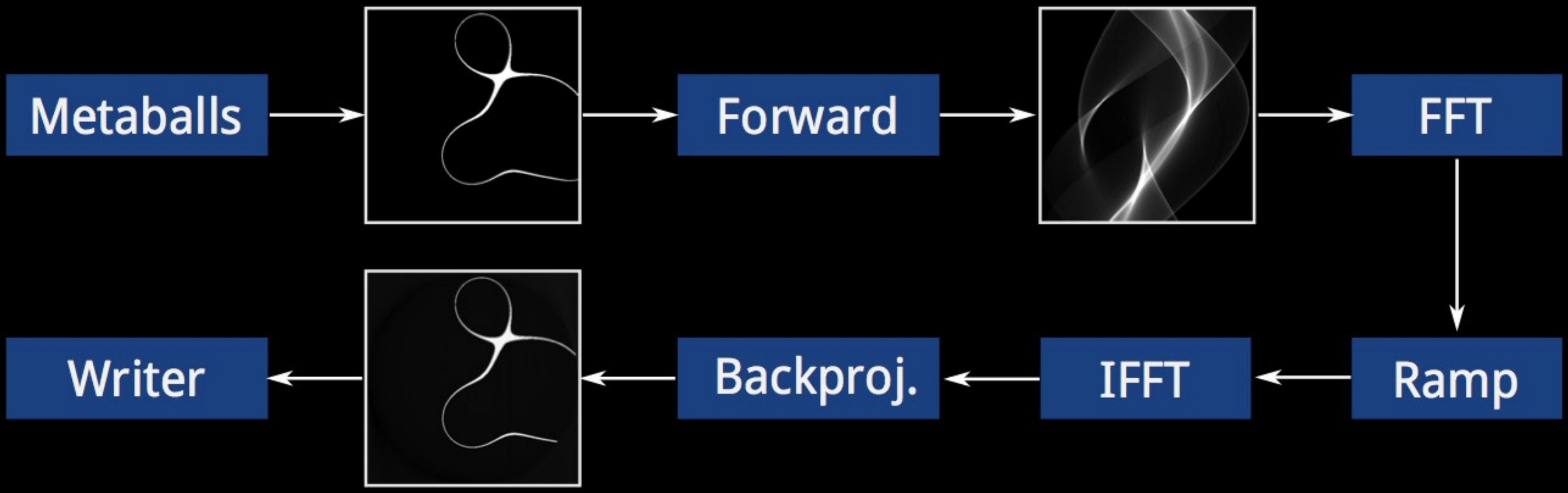
Example: Filtered Back projection



Simulating tomography

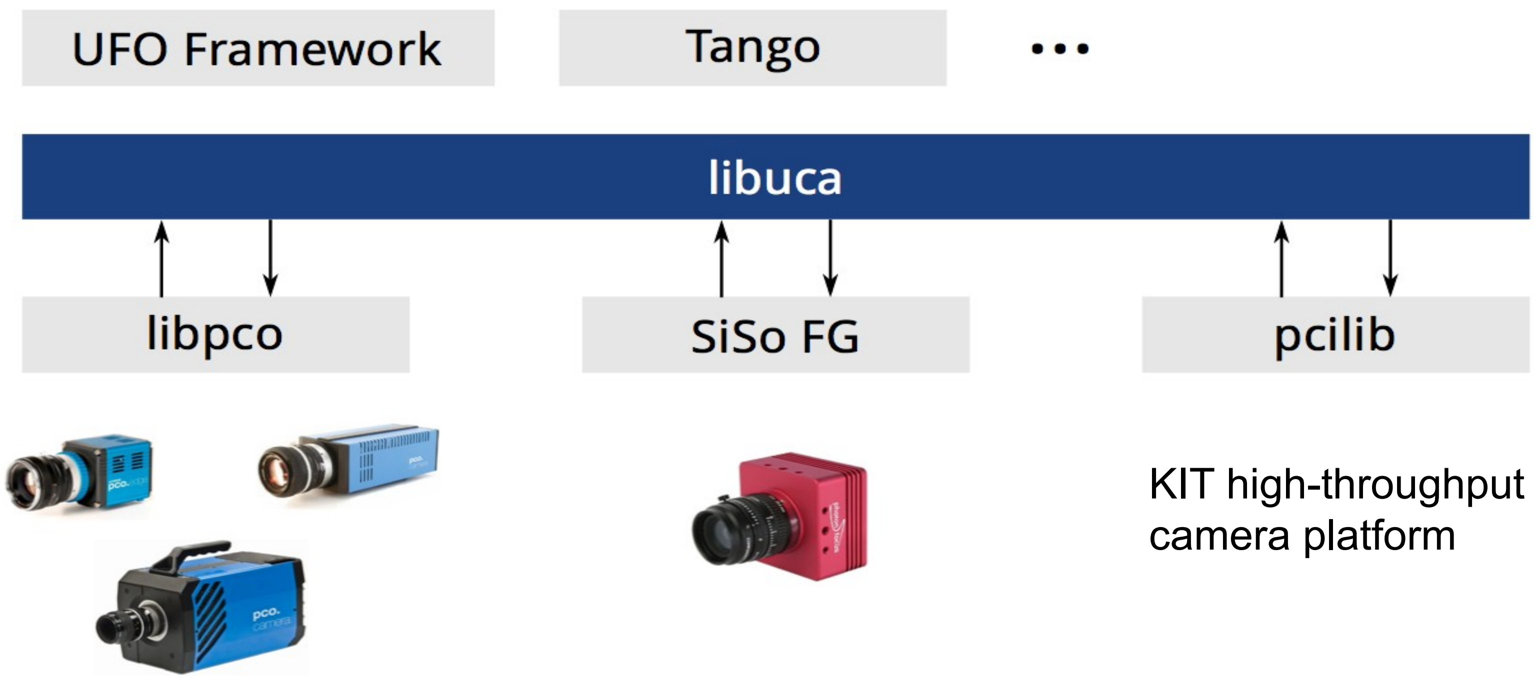
Filter the sinogram for correct back projection

```
metaballs.connect_to(forward)
forward.connect_to(fft)
fft.connect_to(ramp)
ramp.connect_to(ifft)
ifft.connect_to(backproject)
backproject.connect_to(writer)
```



Camera Abstraction

- Generalized access to streaming cameras (C-API)
- 64-bit linux support for PCO cameras
- Licensed under LGPL with permission from PCO



Summary

HELMHOLTZ ASSOCIATION

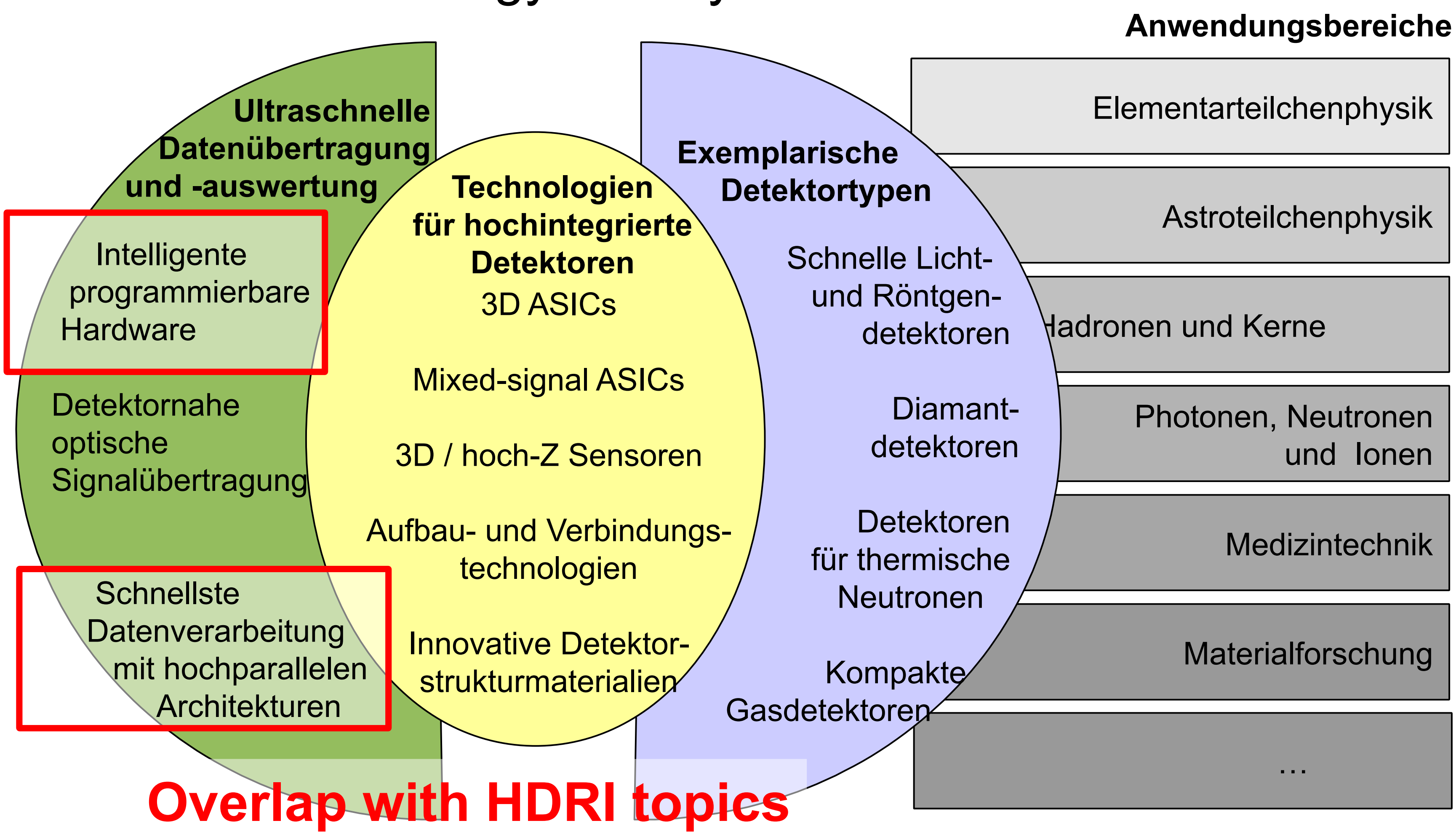
- Search for common DAQ platform (DESY, GSI, KIT)
- Evaluation of GPU platforms
- Tomographic reconstruction (PyHST by ESRF, KIT)
- GPU optimizations for PX (KIT, DESY)
- Linux drivers for PCO cameras (libuca by KIT)
- Parallel processing framework for streamed data (KIT, HZG, Soleil)
  - Simplifies GPU programming
  - Possible standard interfaces for GPU algorithms ?

Web	www.pni-hdri.de
Wiki	https://wiki.pni-hdri.de
Mailing list	wp2@pni-hdri.de
	https://lists.desy.de/sympa/info/pni-hdri-wp2

HGF-Portfolio Process

HELMHOLTZ ASSOCIATION

Detector Technology and Systems Platform



HGF-Portfolio Process

Detector Technology and Systems Platform

Kick-off Meeting 21.-22.2.2012 at KIT:

Partners:
IN2P3, FNAL, STFC, PSI
MPI Physik

HELMHOLTZ ASSOCIATION

DESY

GSI

HZB Helmholtz Zentrum Berlin

HZDR

Helmholtz-Zentrum Geesthacht
Zentrum für Material- und Küstenforschung

KIT
Karlsruhe Institute of Technology

JGU

seit 1558

HGF-Portfolio Process

Detector Technology and Systems Platform

Demonstrator „HGF-Cube“: