

ONDA

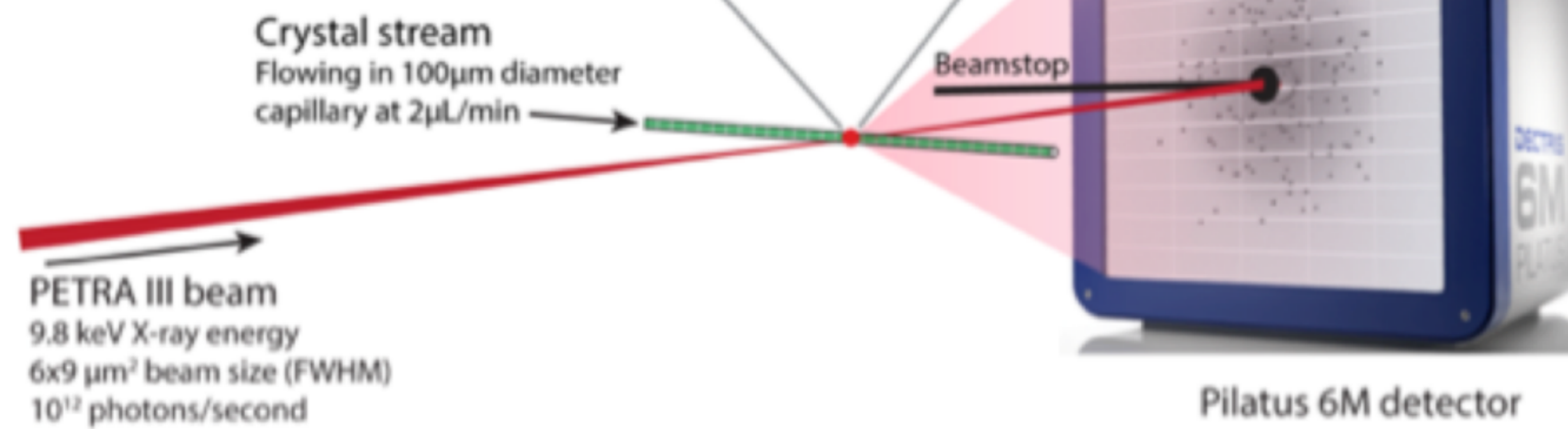
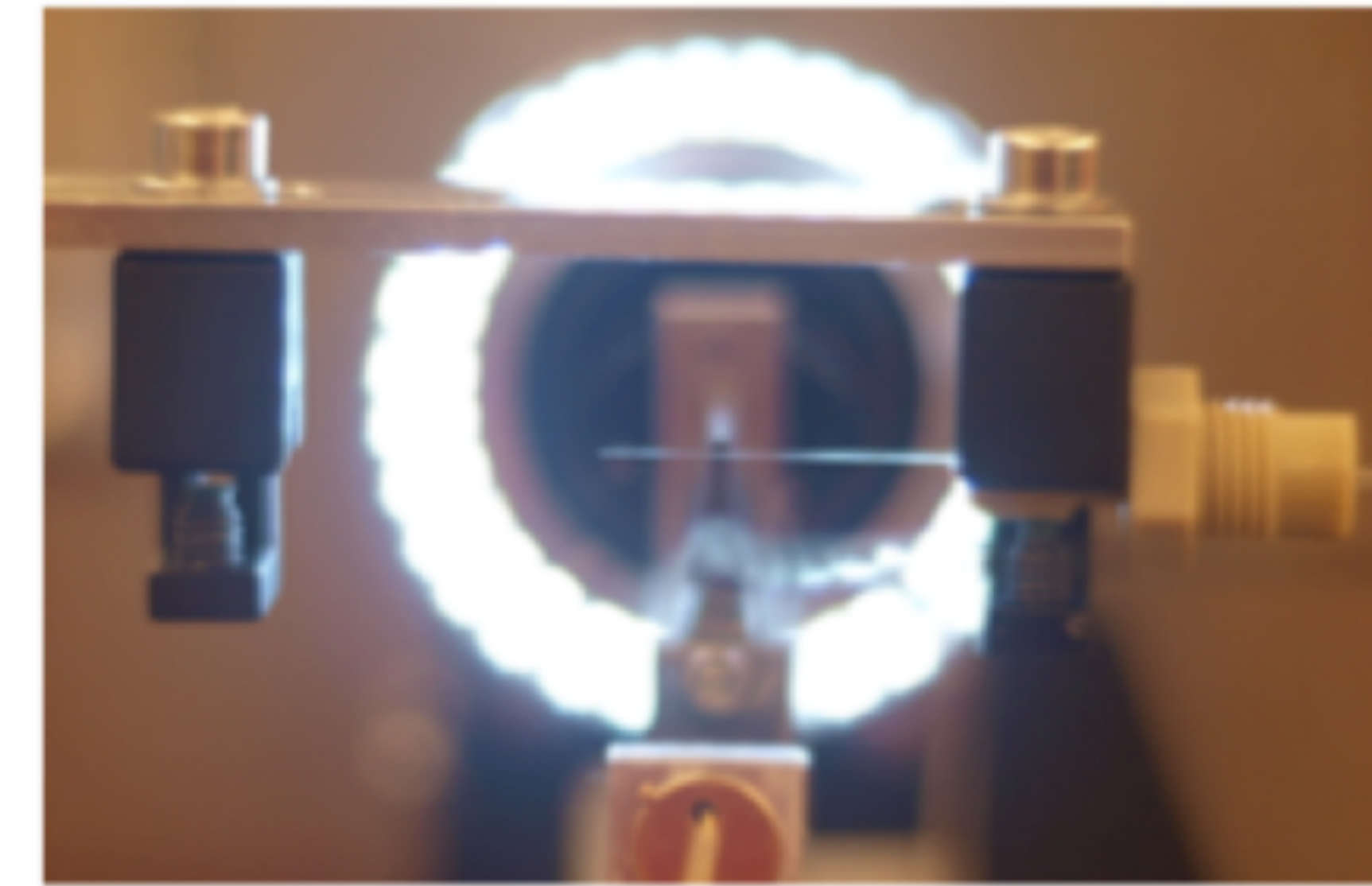
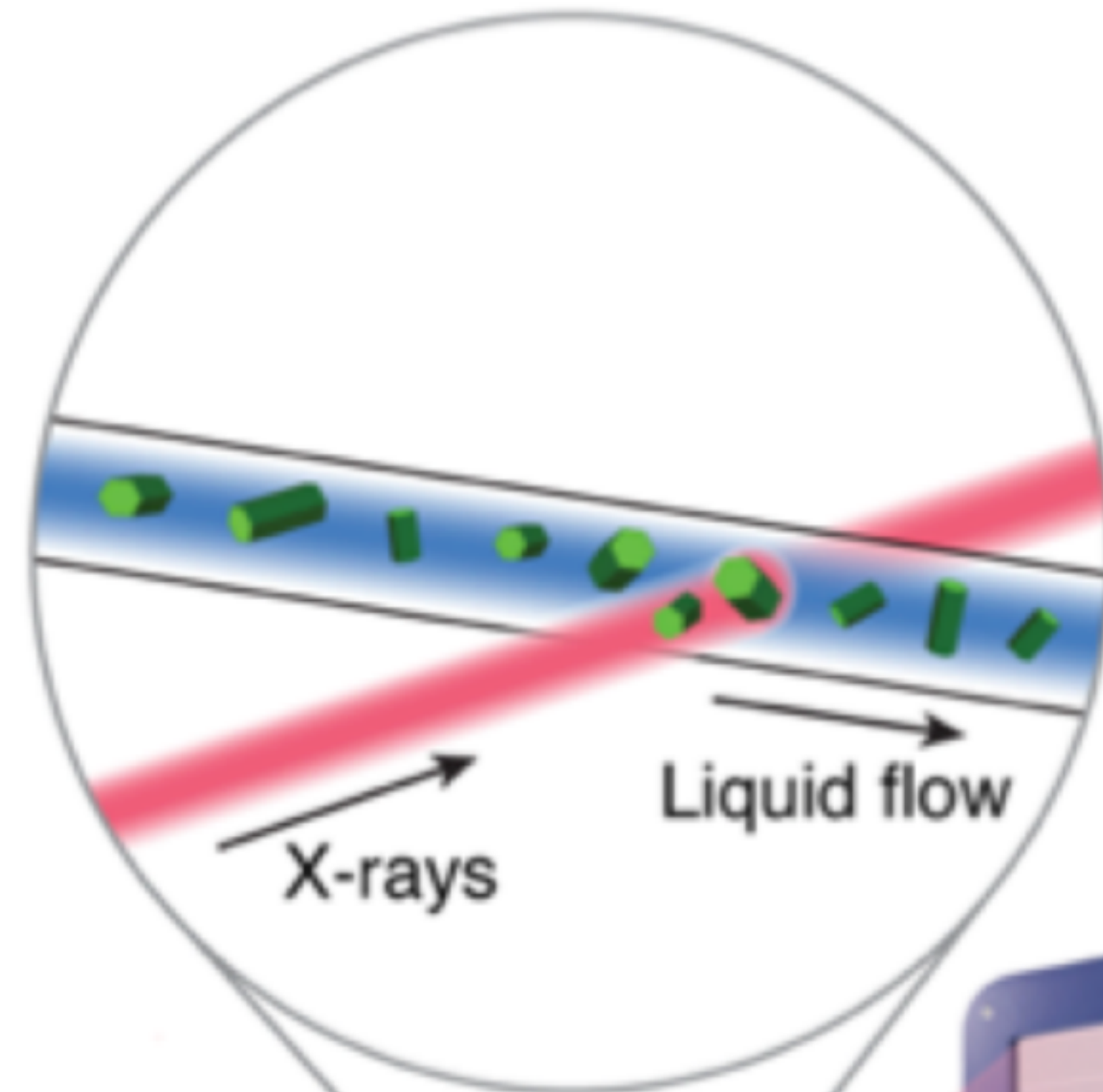
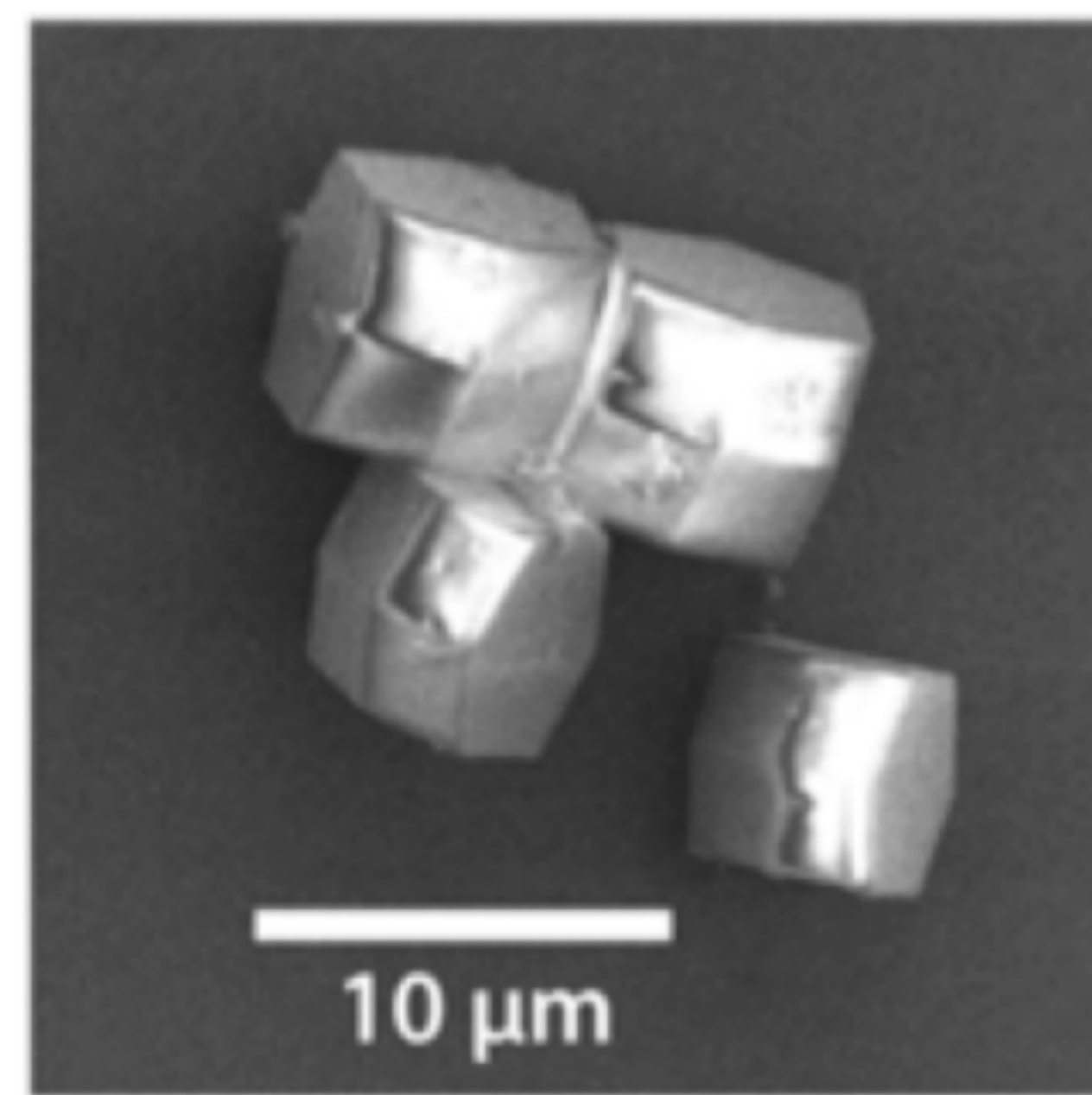
Online Data Analysis

<https://www.ondamonitor.com>

V. Mariani and others

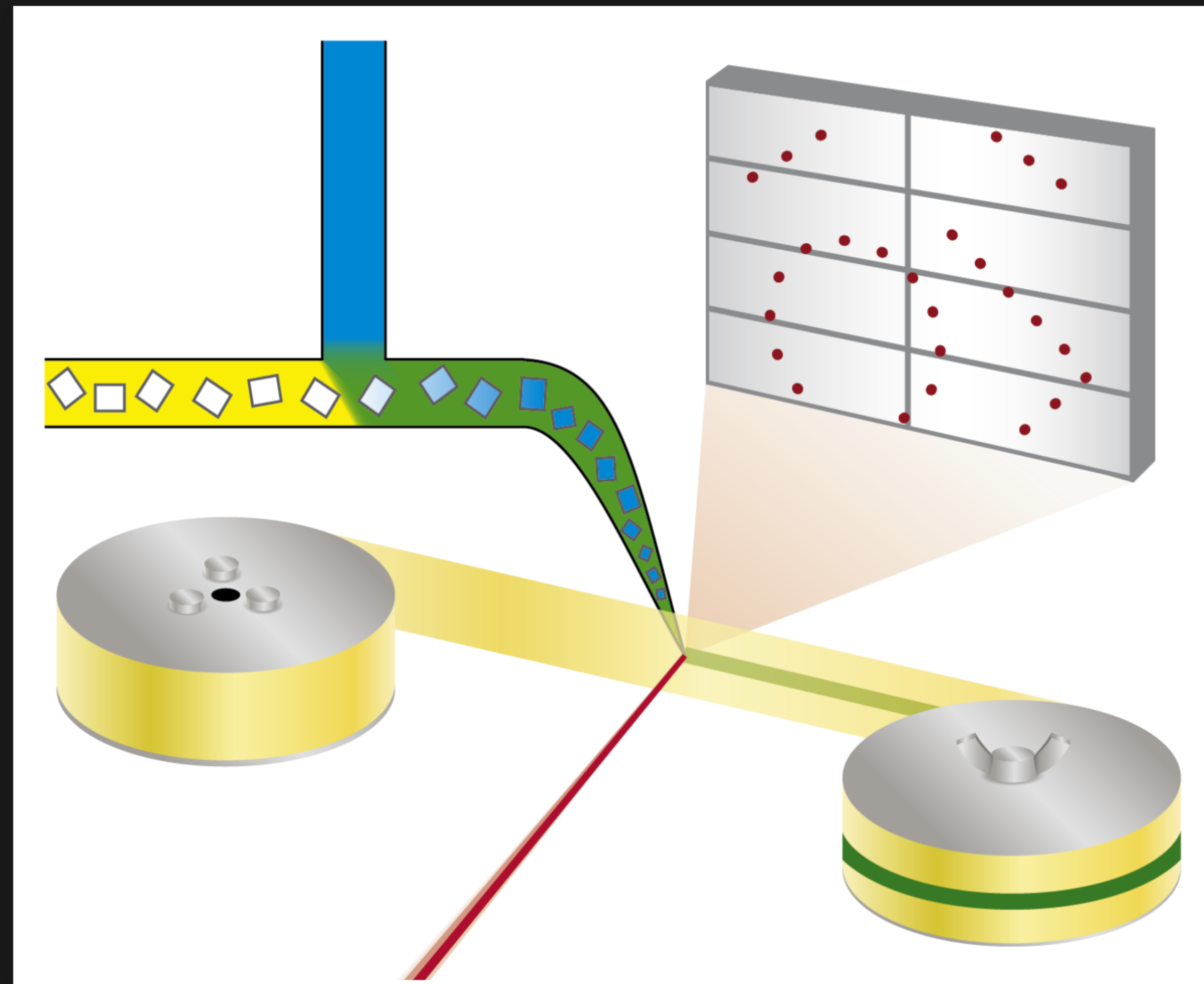


SERIAL CRYSTALLOGRAPHY (LIQUID JET)

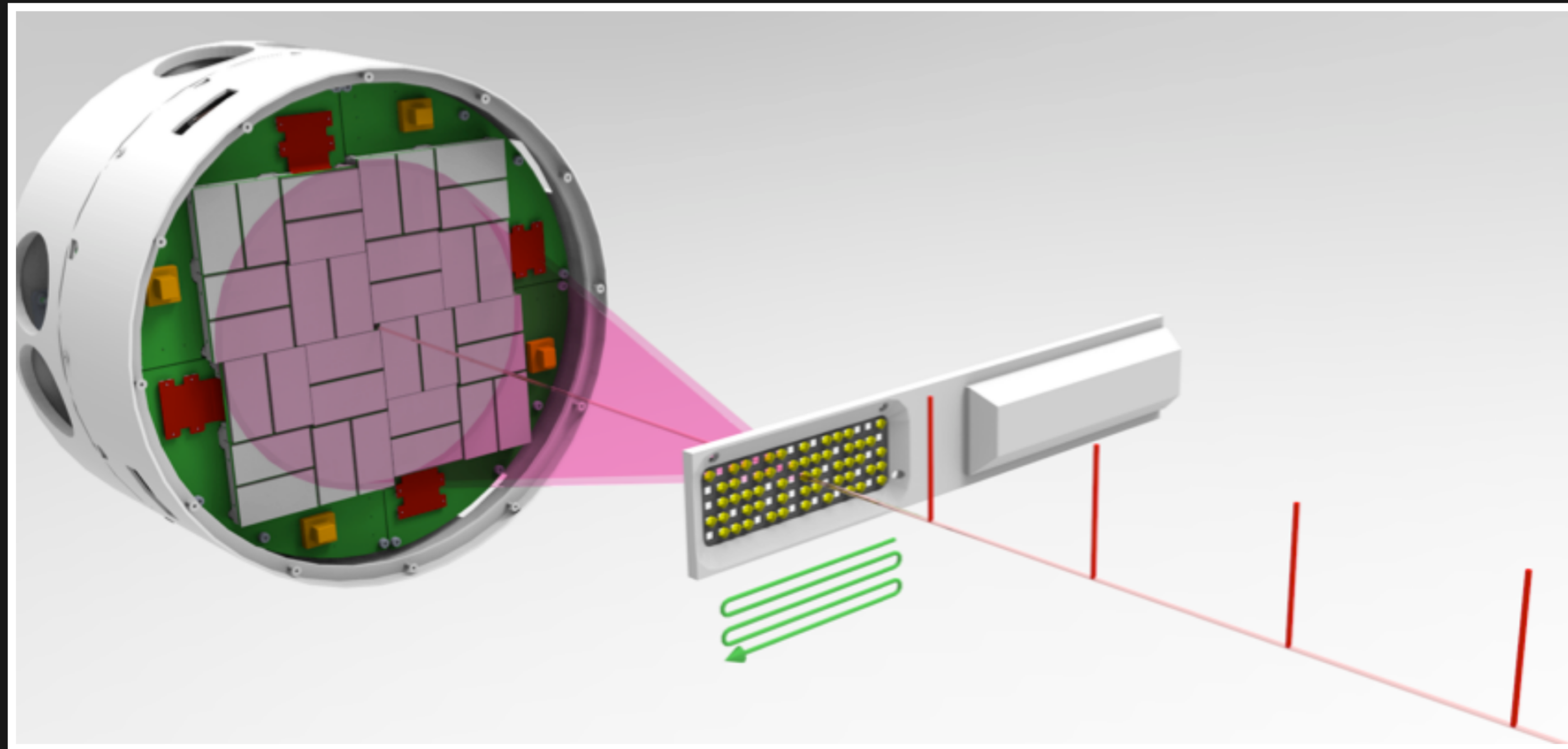


6M at 100 Hz
360k frames / hr
4 TB per hour

SERIAL CRYSTALLOGRAPHY (TAPE DRIVE)



SERIAL CRYSTALLOGRAPHY (CHIP)



DURING THE EXPERIMENT...

- We need data that help us take **immediate** decisions (within **seconds**)
- Examples:
 - Are we hitting the sample? (Move sample)
 - Is the injector clogged? (Stop run)
 - Did the injector liquid jet move? (Adjust injector)
 - Are there crystals on this chip? (Change chip)
 - Are there saturated pixels? (Attenuate beam)

FOR QUICK DECISIONS...

- Accuracy is not a strong requirement. Low latency is
- Low latency is more important than Completeness

WHAT IS ONDA?

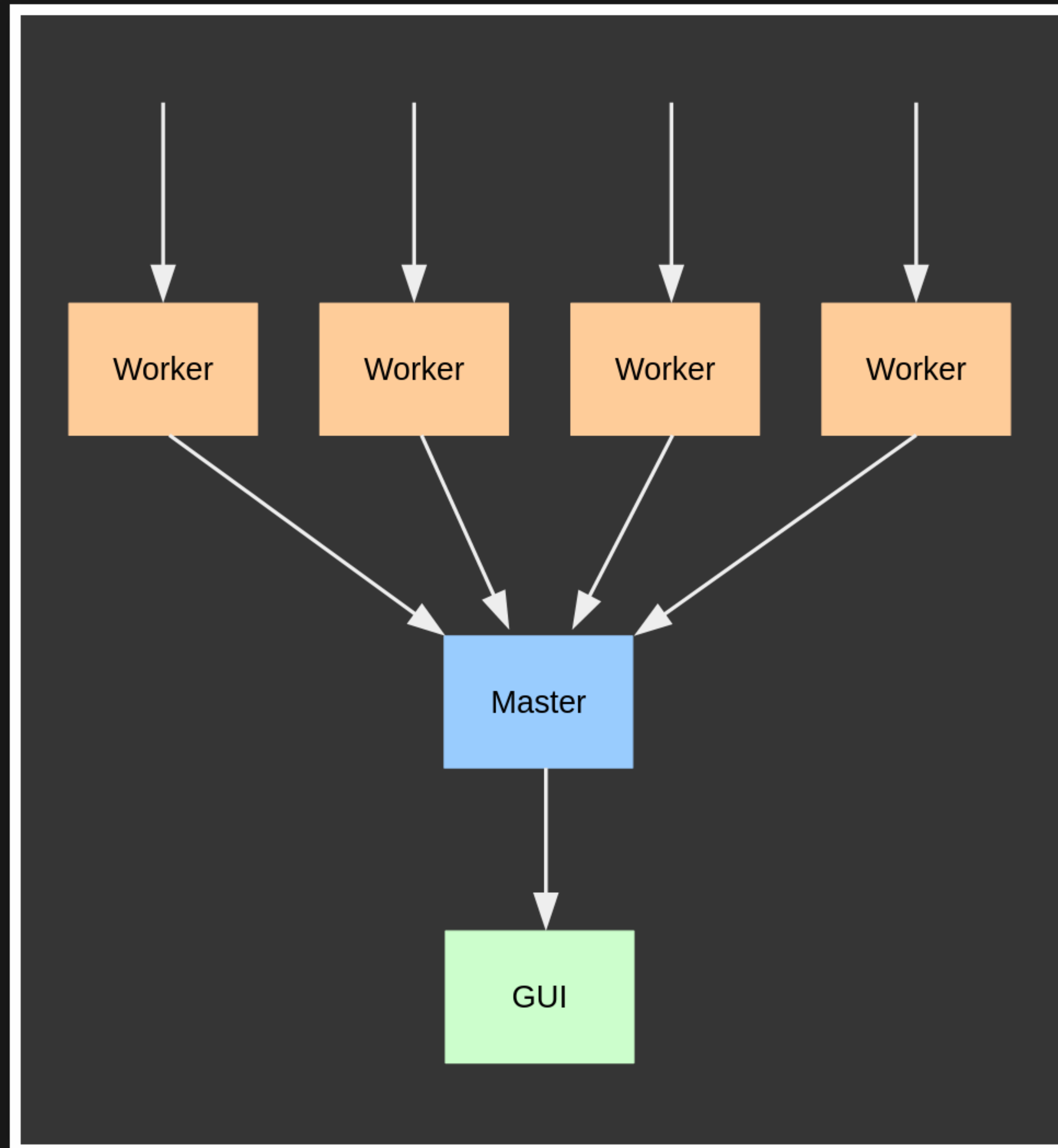
- OnDA is an **online** monitor
- Up to date information for real-time human feedback
- Keep up with data flow (Fresh data)
- Speed is paramount
- Accuracy & completeness are secondary

WHAT IS ONDA?

- Online monitoring framework
- Set of pre-packaged online monitors

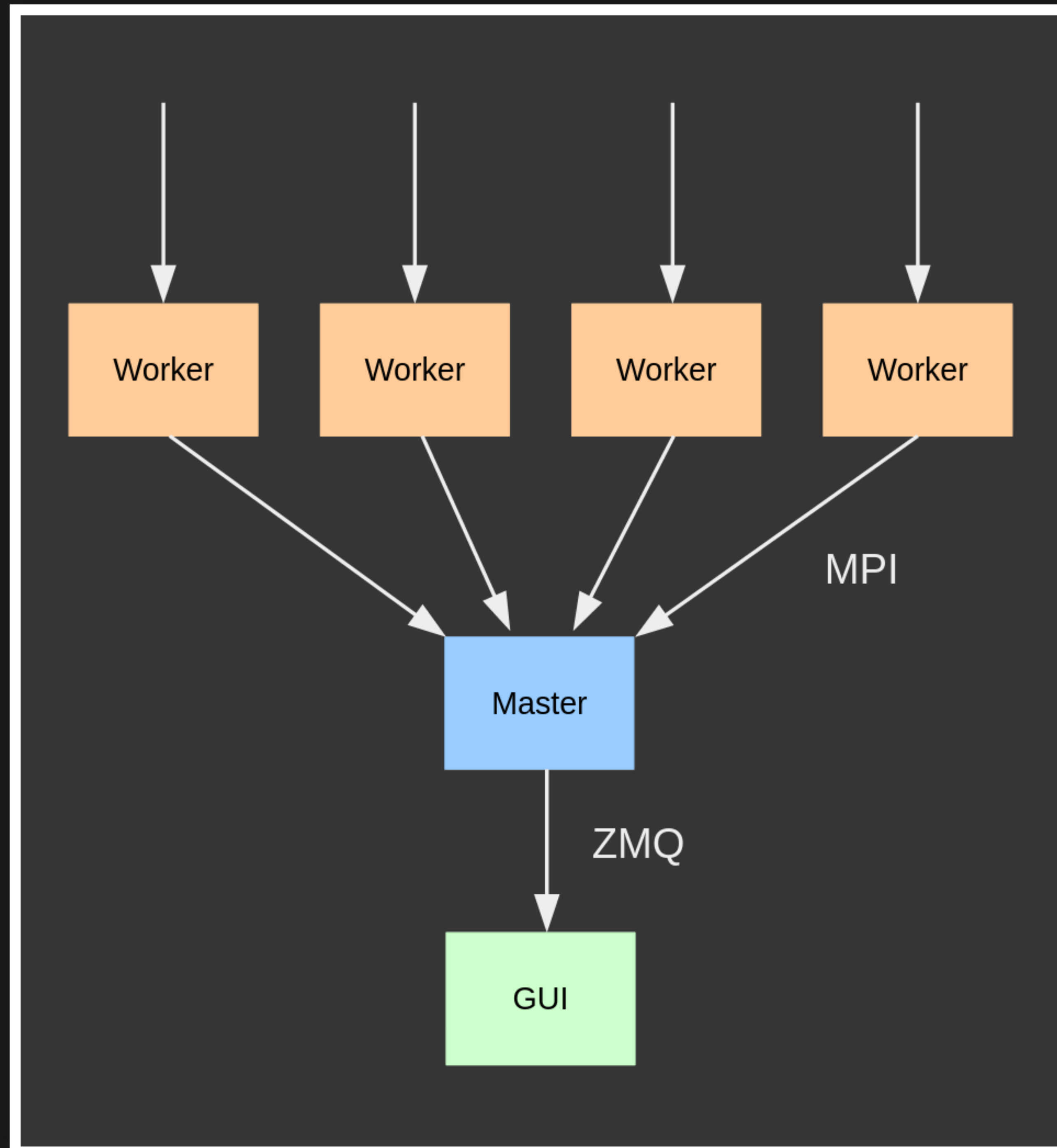
DESIGN GUIDELINES

- Minimalistic design (K.I.S.S.)
- Speed and stability
- Intrinsic parallelization
- Portability
- Flexibility



PARALLELIZATION IN ONDA (CURRENTLY)

- Workers process data
- Workers send data to Master
- Master aggregates data
- Master sends data to GUI(s)



PARALLELIZATION IN ONDA (CURRENTLY)

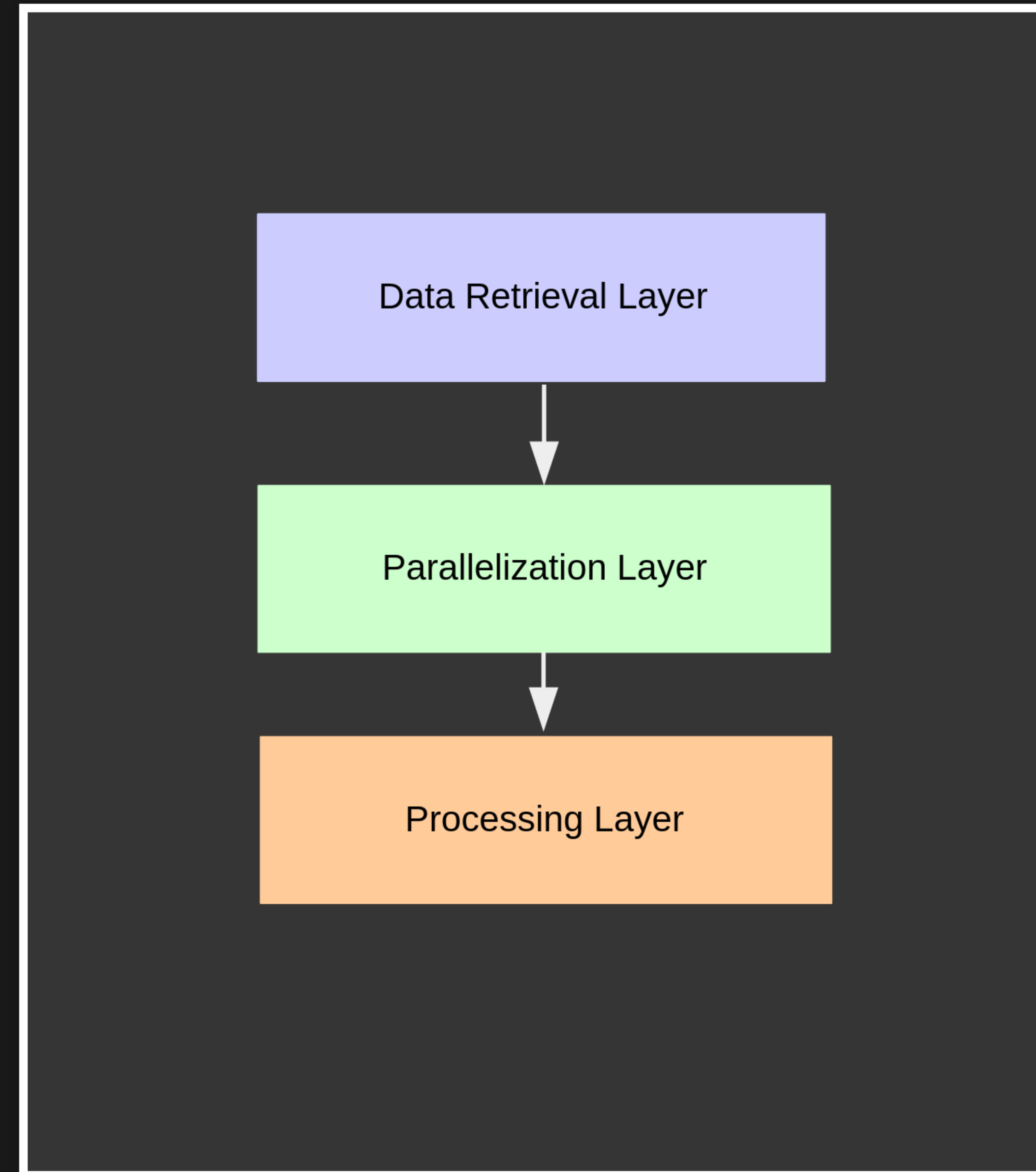
- Worker > Master: MPI
- Master > GUI: ZeroMQ
- Implementation: Python (and C/C++)

LAYERS

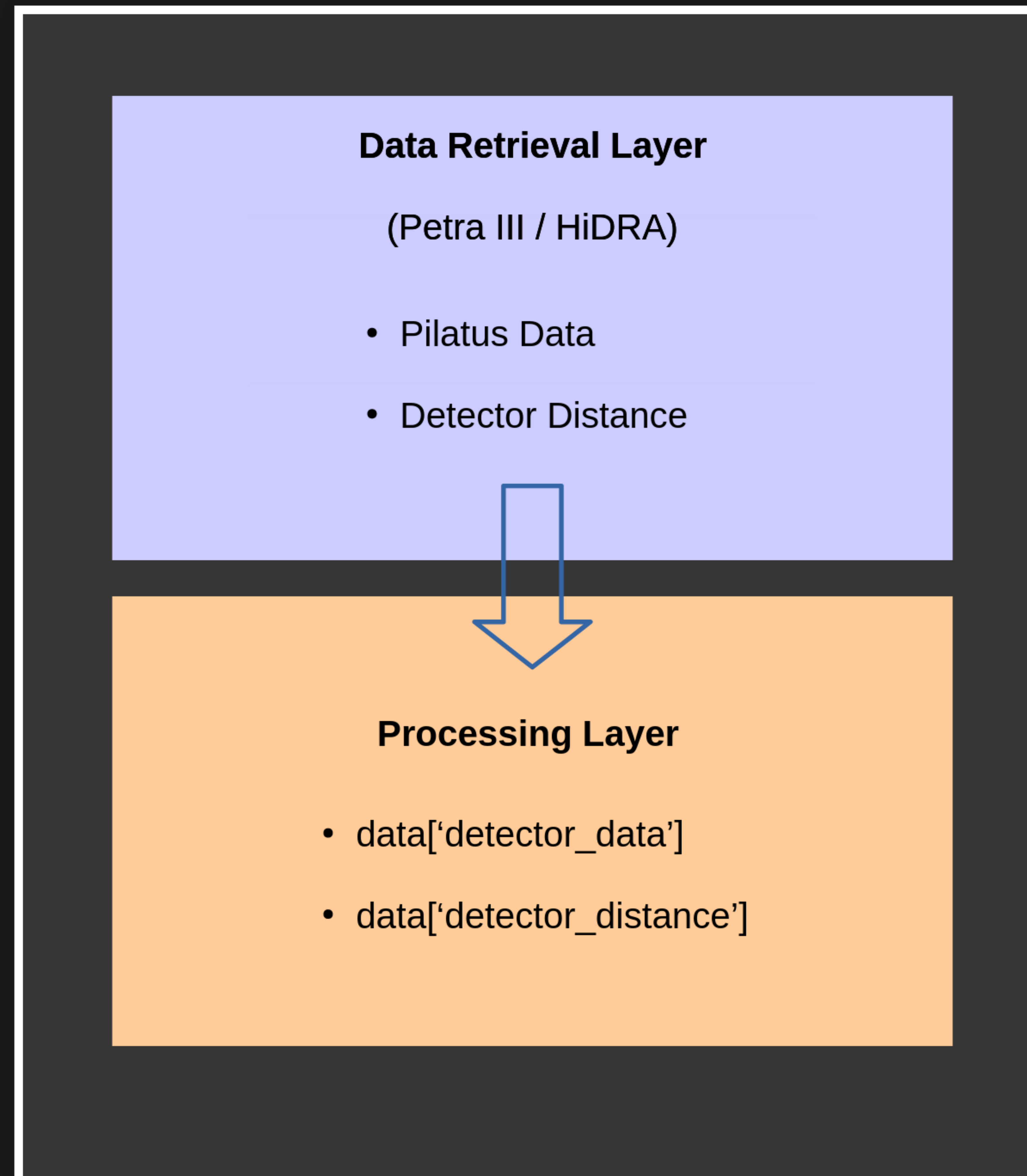
Data Retrieval Layer:
data extraction

Parallelization Layer:
master/worker comm

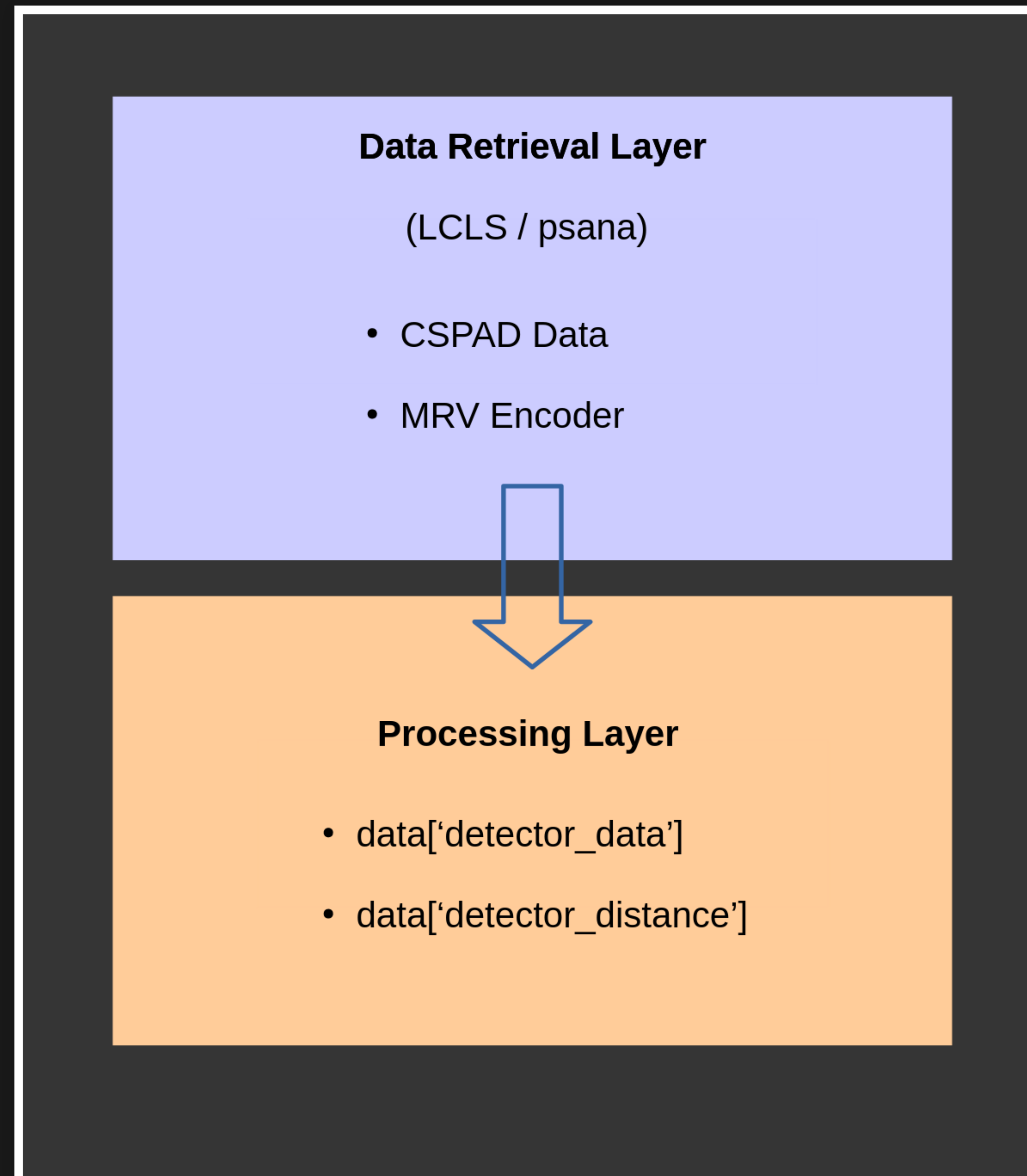
Processing Layer:
data processing



DATA RETRIEVAL LAYER

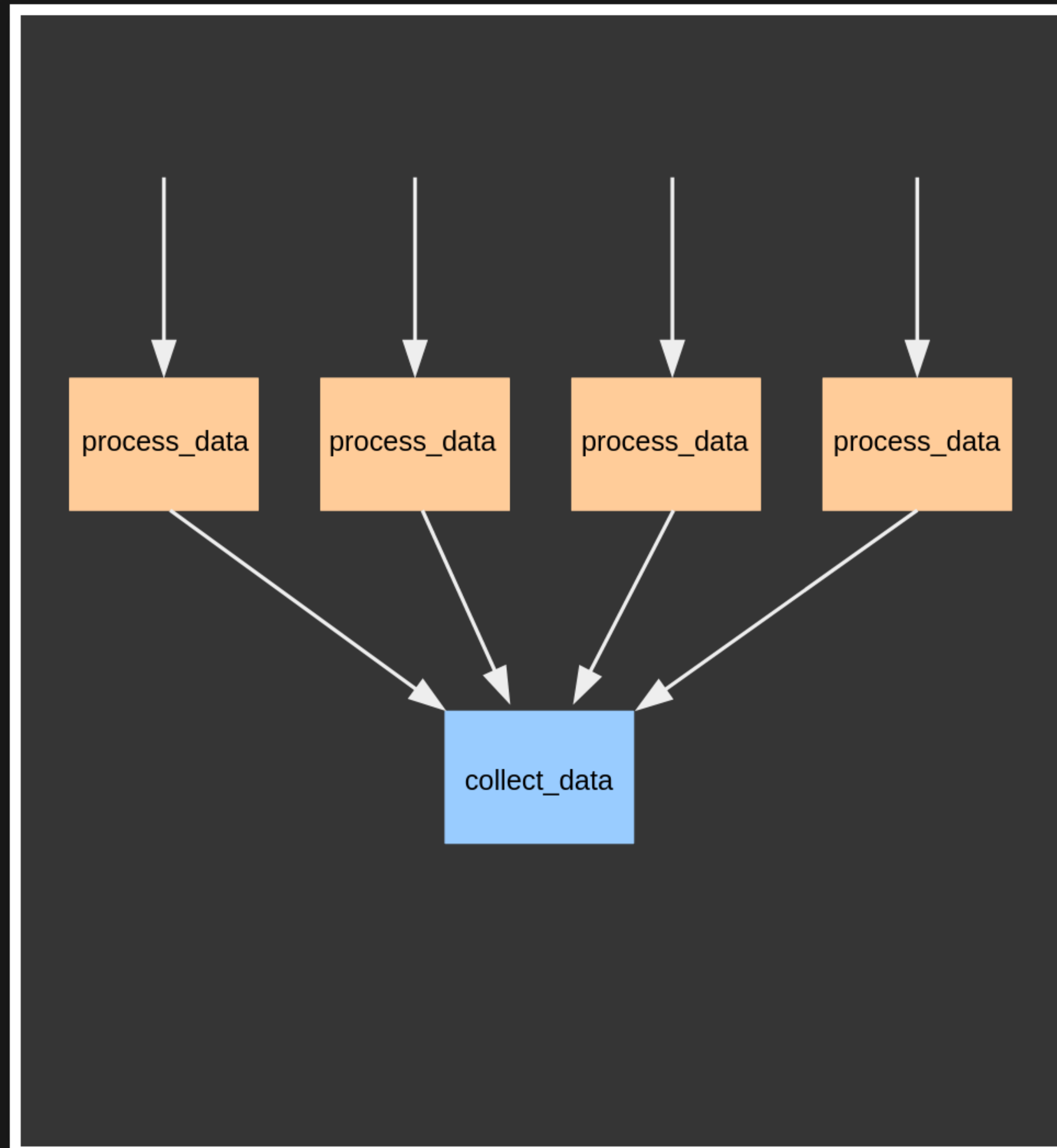


DATA RETRIEVAL LAYER



DATA RETRIEVAL LAYER

- "Core Developers"!
- Real-time recovery of data (facility-dependent)
- Close contact with facilities (often on site!)
- Turning facility-dependent data formats into...
- ...facility-independent abstractions



PROCESSING LAYER

Functions

- Worker > process_data
- Master > collect_data

PROCESSING LAYER

Example: Worker

```
def process_data(self, data):  
    # DATA IS AVAILABLE IN THE data DICTIONARY  
  
    img = data['detector_data']  
    peaks = self.peakfinder.find_peaks(img)  
  
    # PUT EVERYTHING THAT MUST BE SENT TO THE MASTER NODE  
    # IN processed_data.  
  
    processed_data = {'peaks': peaks}  
    return processed_data
```


PROCESSING LAYER

Example: Master

```
def collect_data(self, data):  
  
    # processed_data FROM THE WORKER NODES IS AVAILABLE  
    # AS data  
  
    # ADD YOUR CODE FOR DATA AGGREGATION HERE. FOR EXAMPLE:  
  
    self.peak_accumulator.add_peaks(data['peaks'])  
    collected_data = {  
        'max_resolution': peak_accumulator.get_max_res()  
    }  
    self.broadcast_socket.send_data(collected_data)
```

LAYERS

From configuration file

```
processing_layer = 'crystallography'  
parallelization_layer = 'mpi'  
data_retrieval_layer = 'petra3_p11'
```

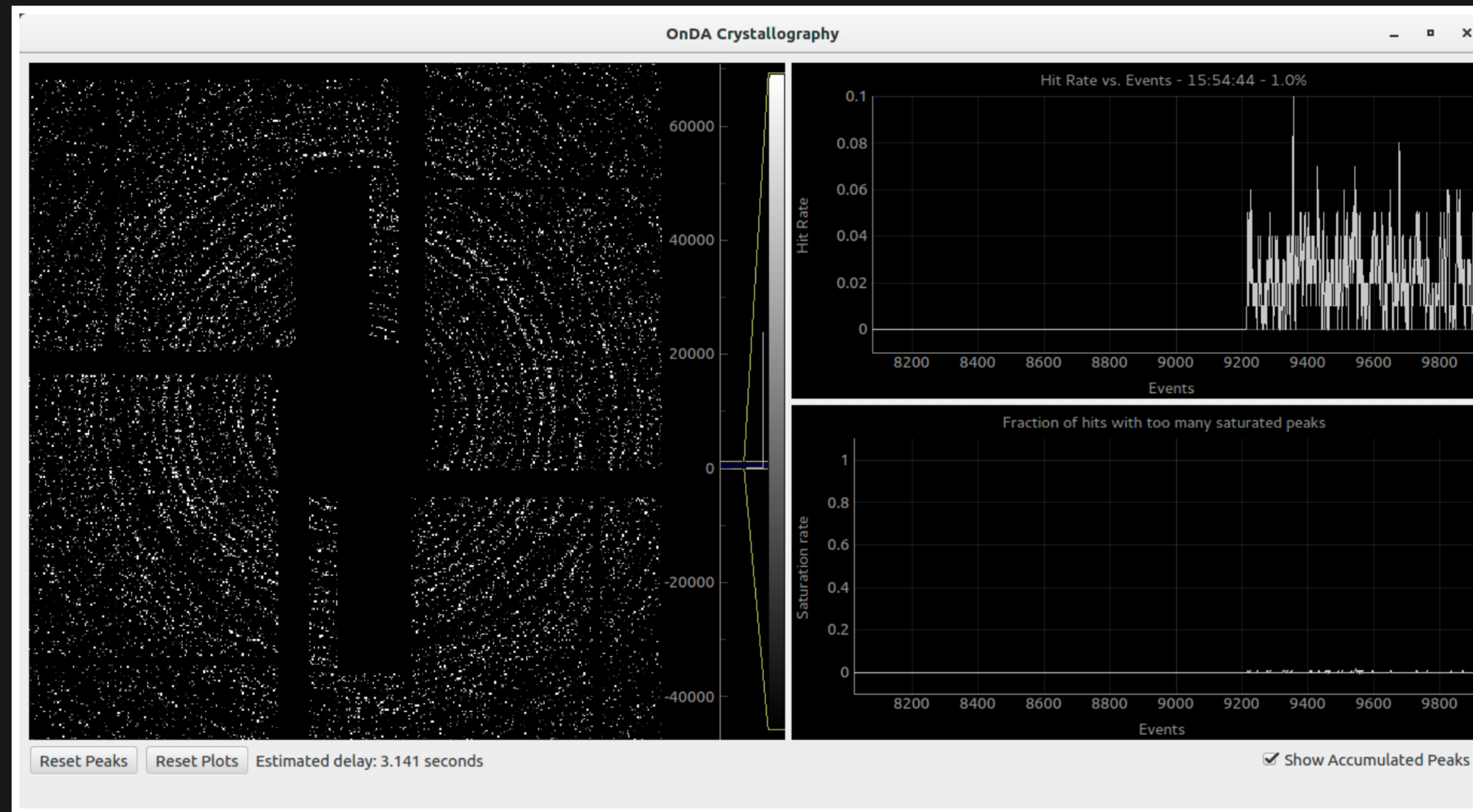
LAYERS

From configuration file

```
processing_layer = 'crystallography'  
parallelization_layer = 'mpi'  
data_retrieval_layer = 'lcls_cxi'
```

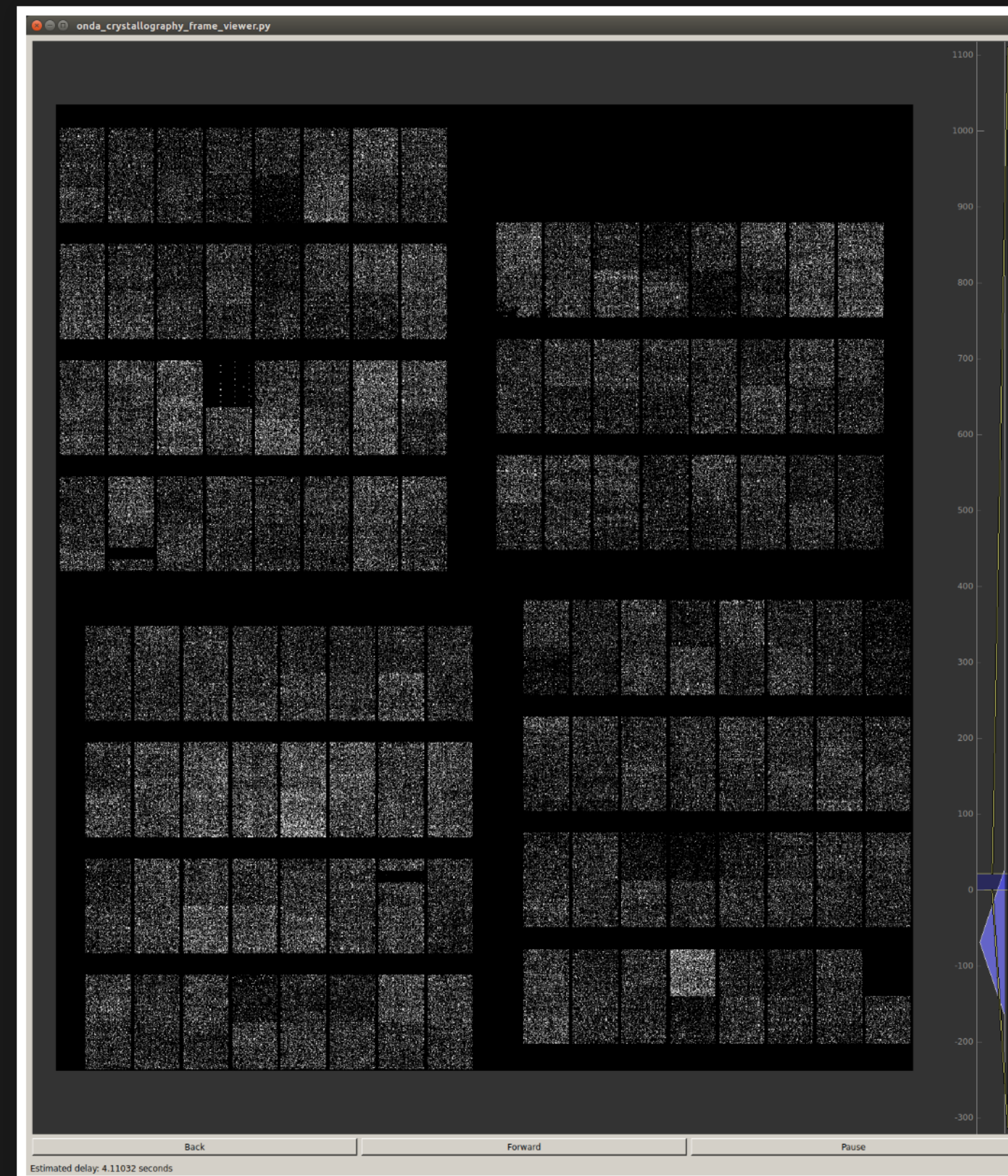
ONDA IN ACTION

ONDA SERIAL CRYSTALLOGRAPHY



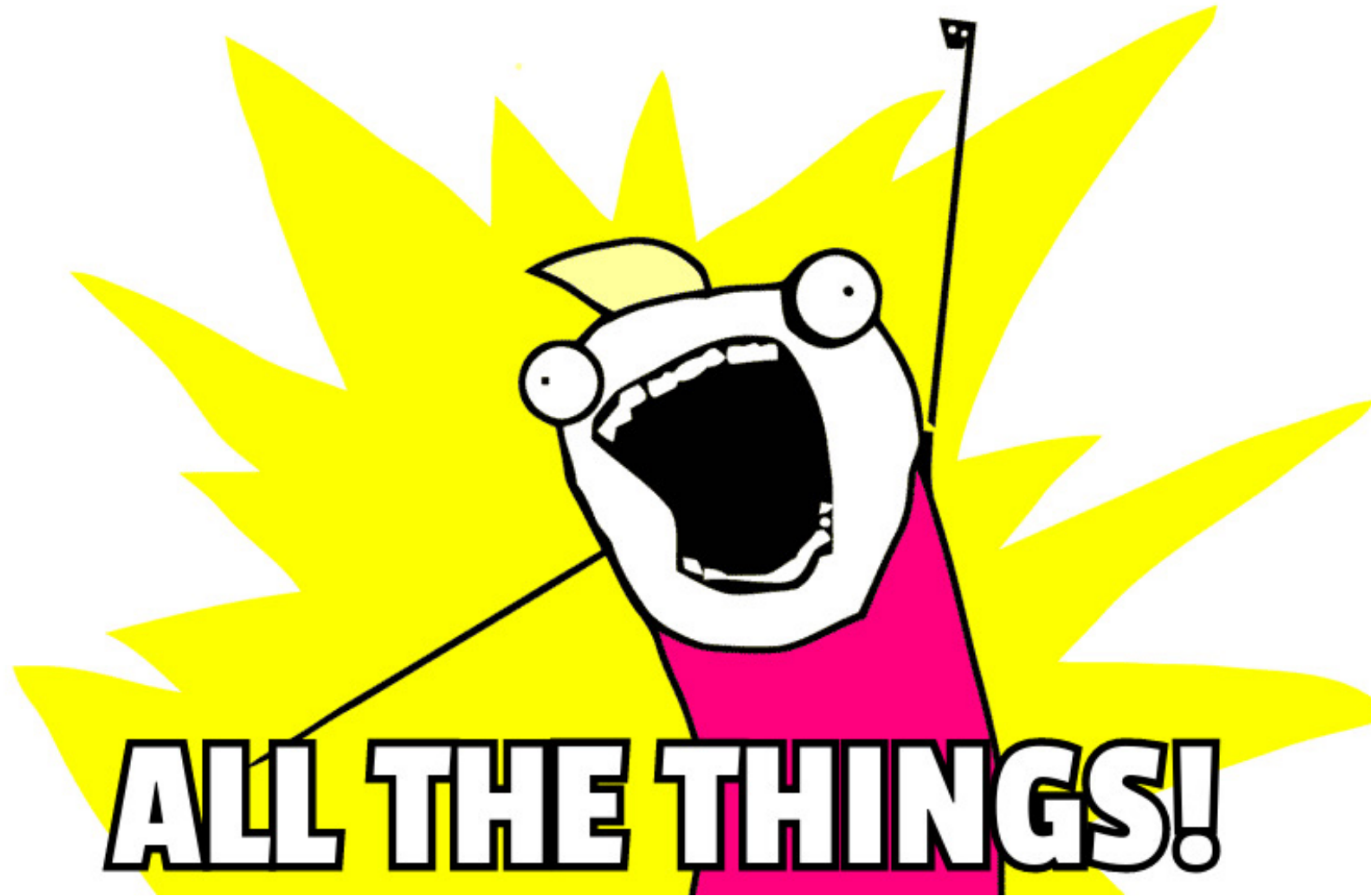
ONDA IN ACTION

ONDA SERIAL CRYSTALLOGRAPHY



ONDA ALL THE THINGS!

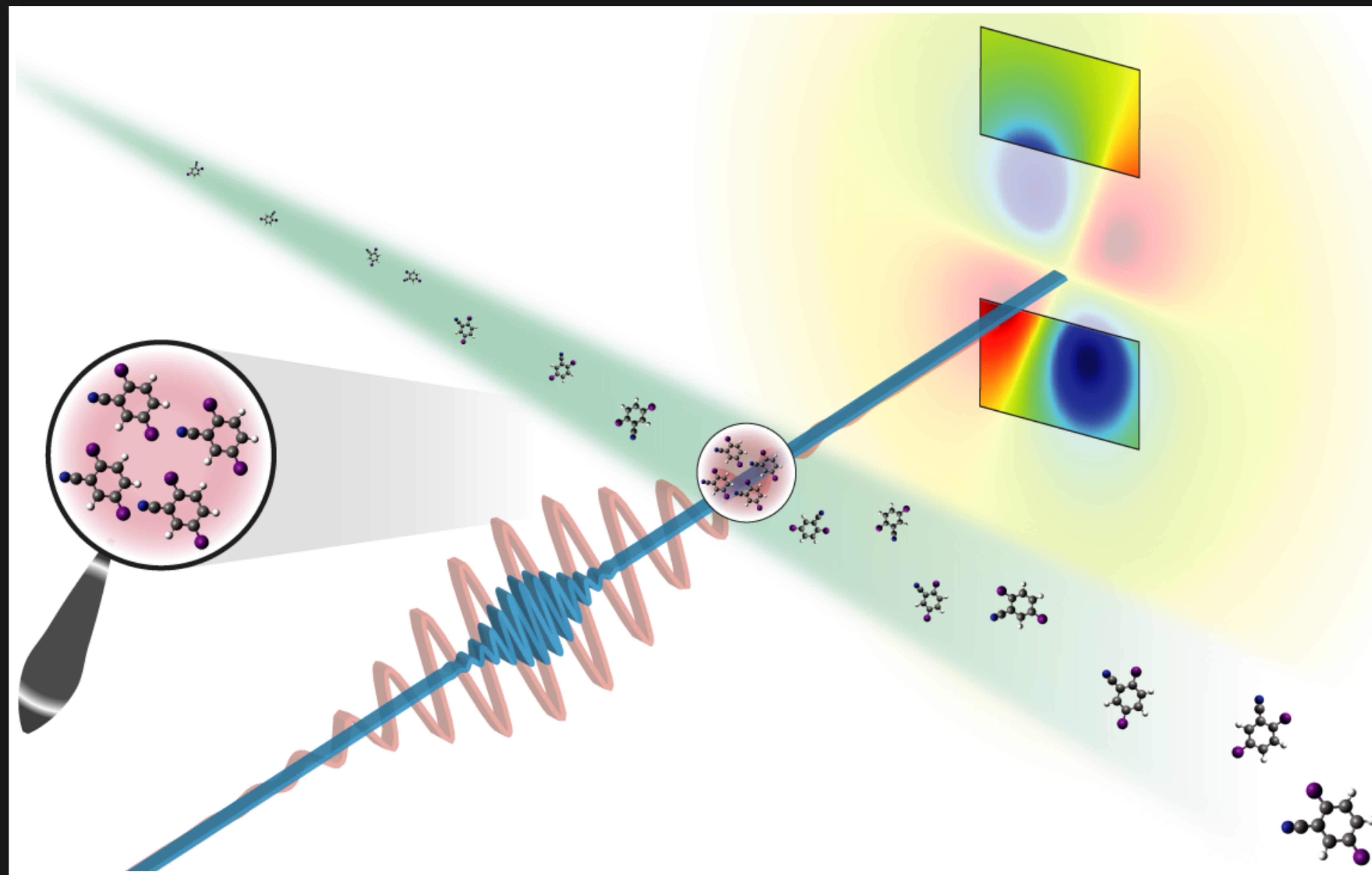
REAL-TIME MONITOR



ALL THE THINGS!

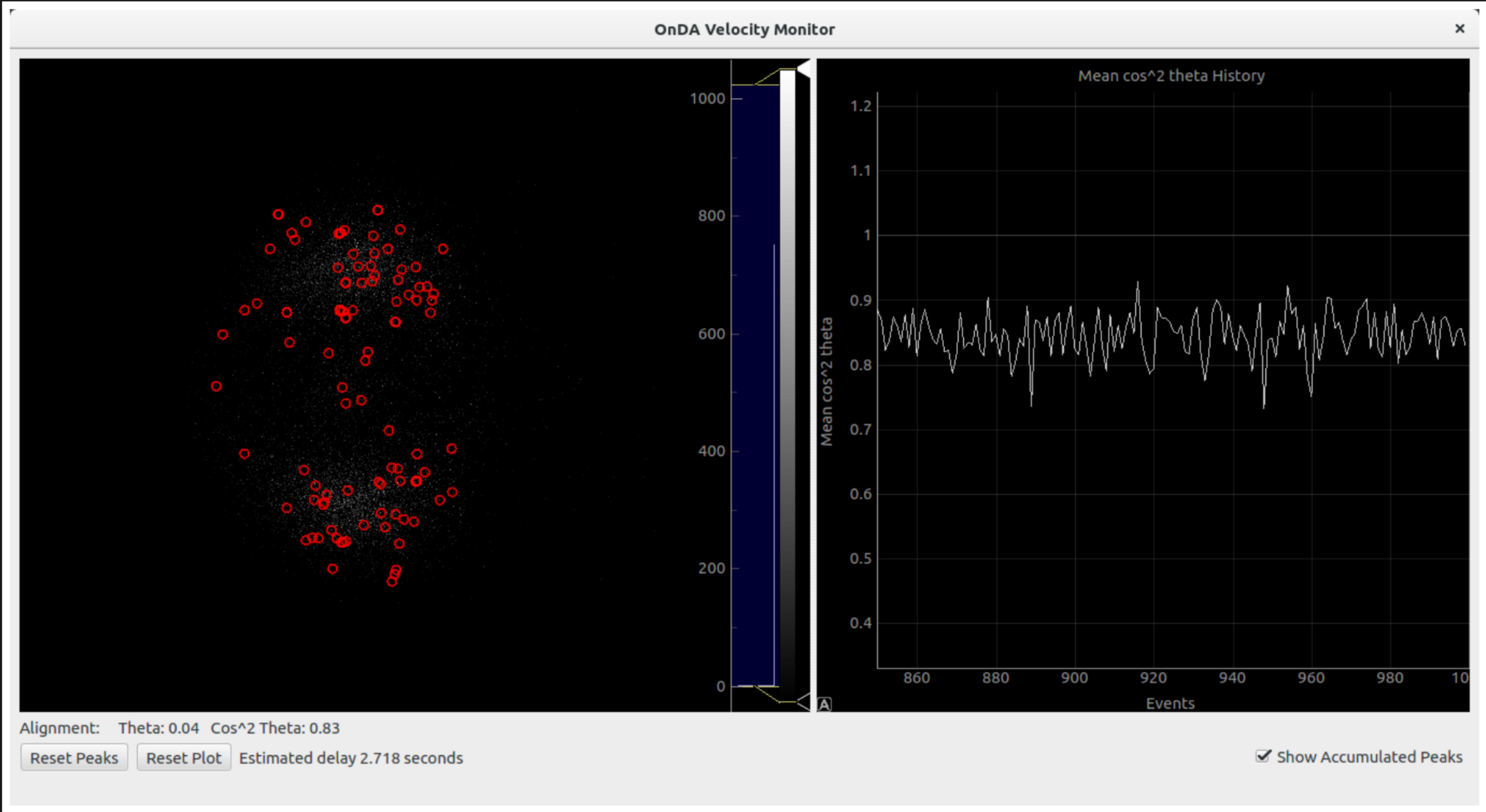
ONDA IN ACTION

ONDA VELOCITY MONITOR IMAGING



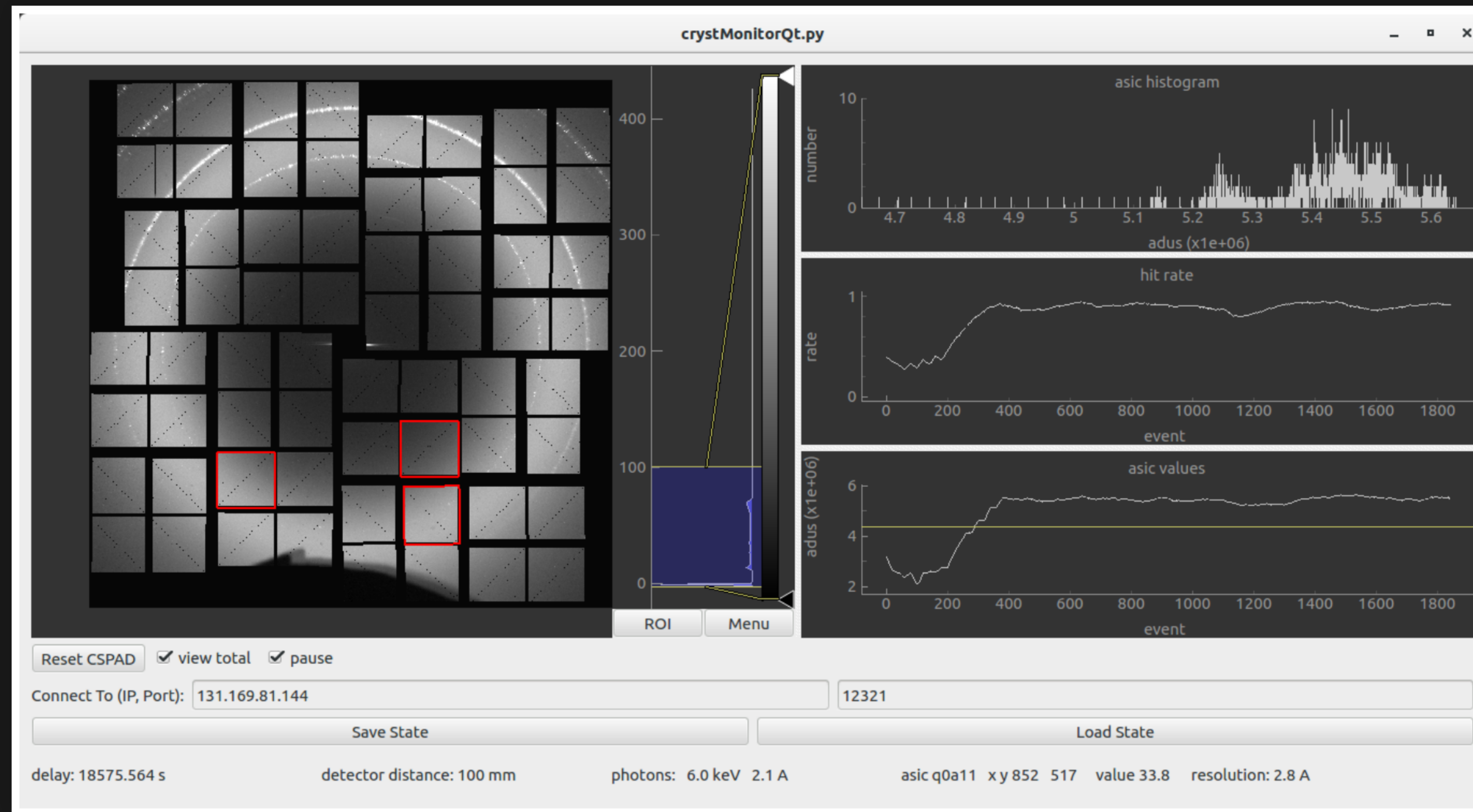
ONDA IN ACTION

ONDA VELOCITY MONITOR IMAGING



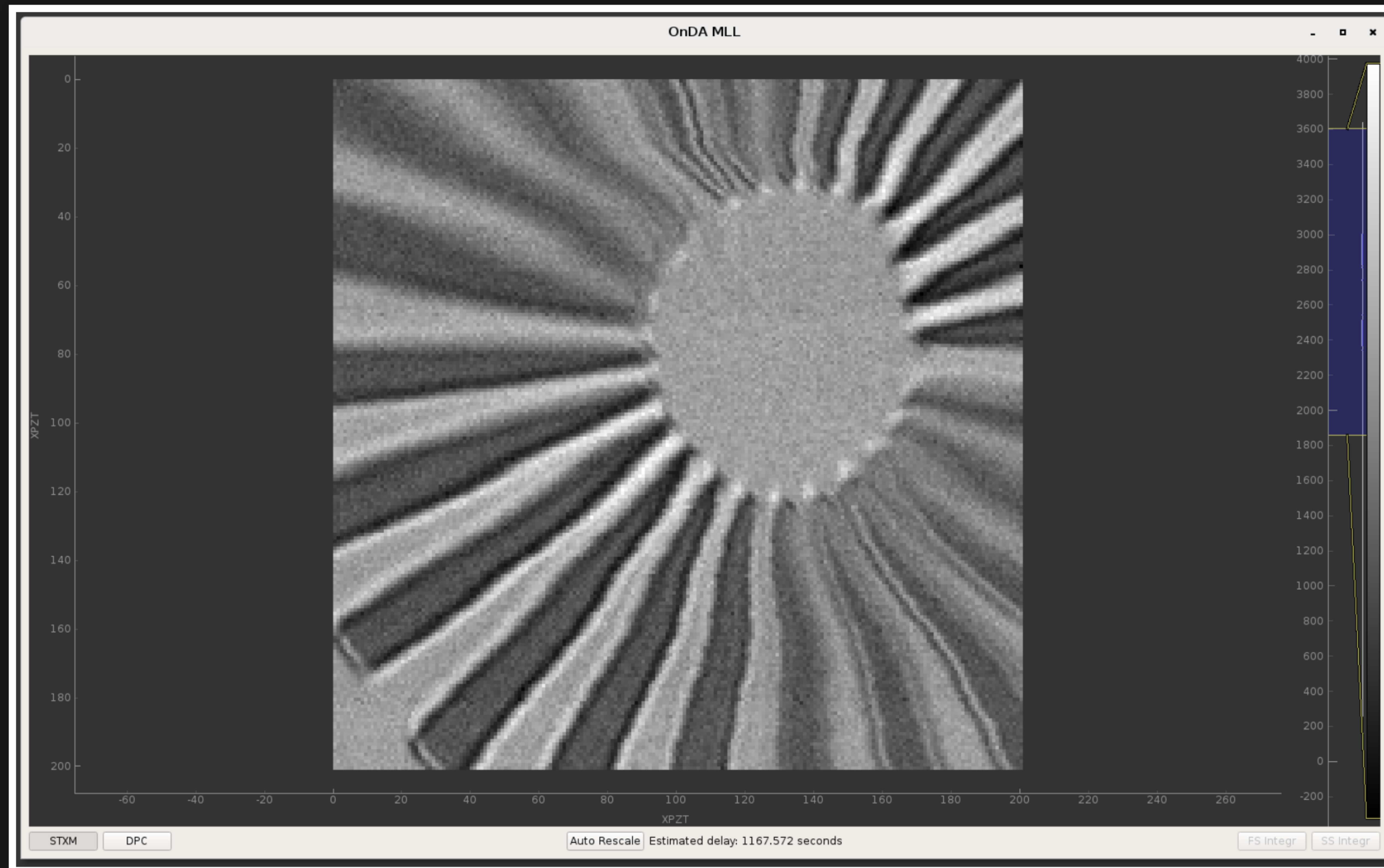
ONDA IN ACTION

ONDA FIBER SCATTERING



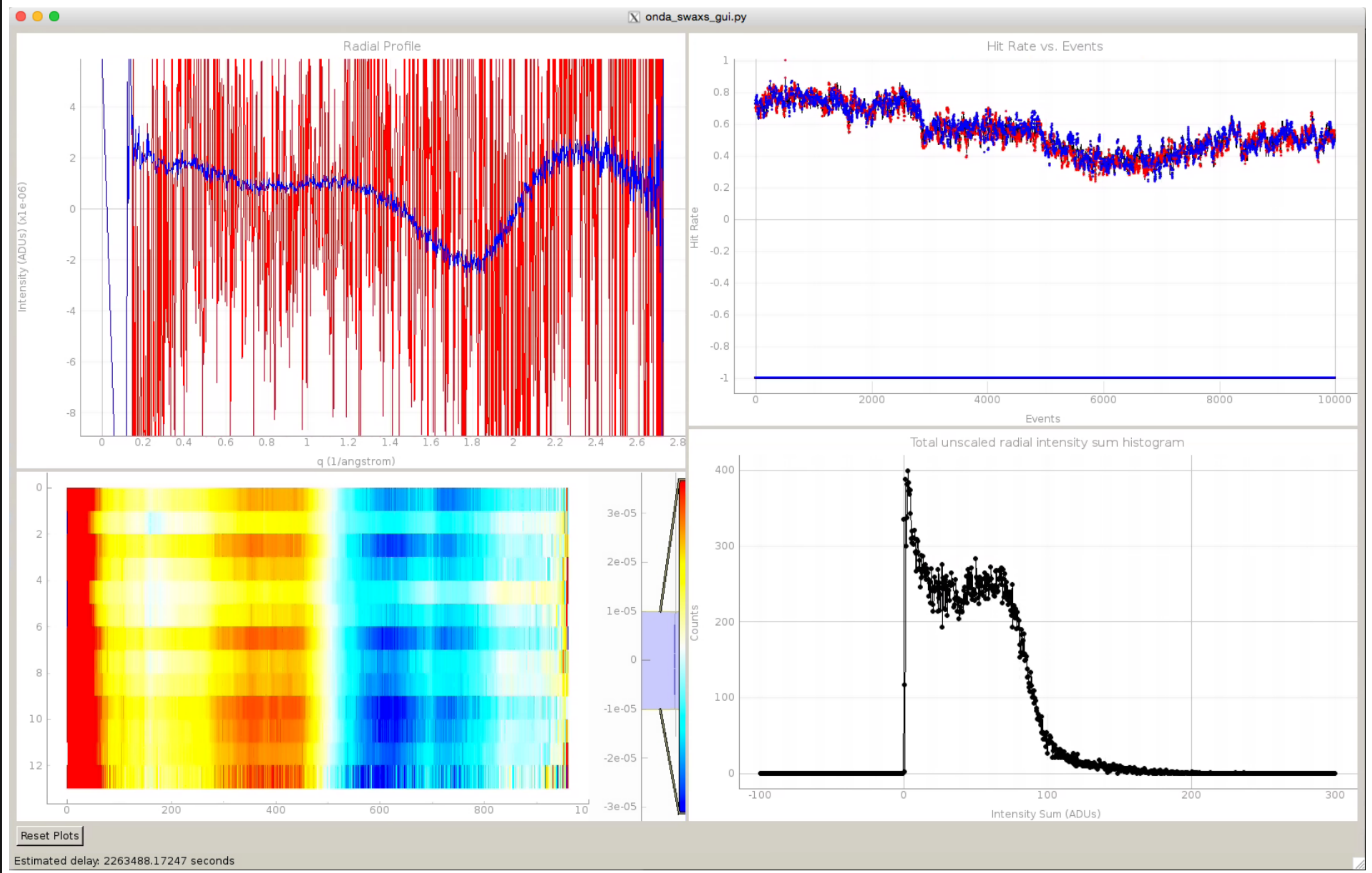
ONDA IN ACTION

ONDA MULTILAYER LENSES



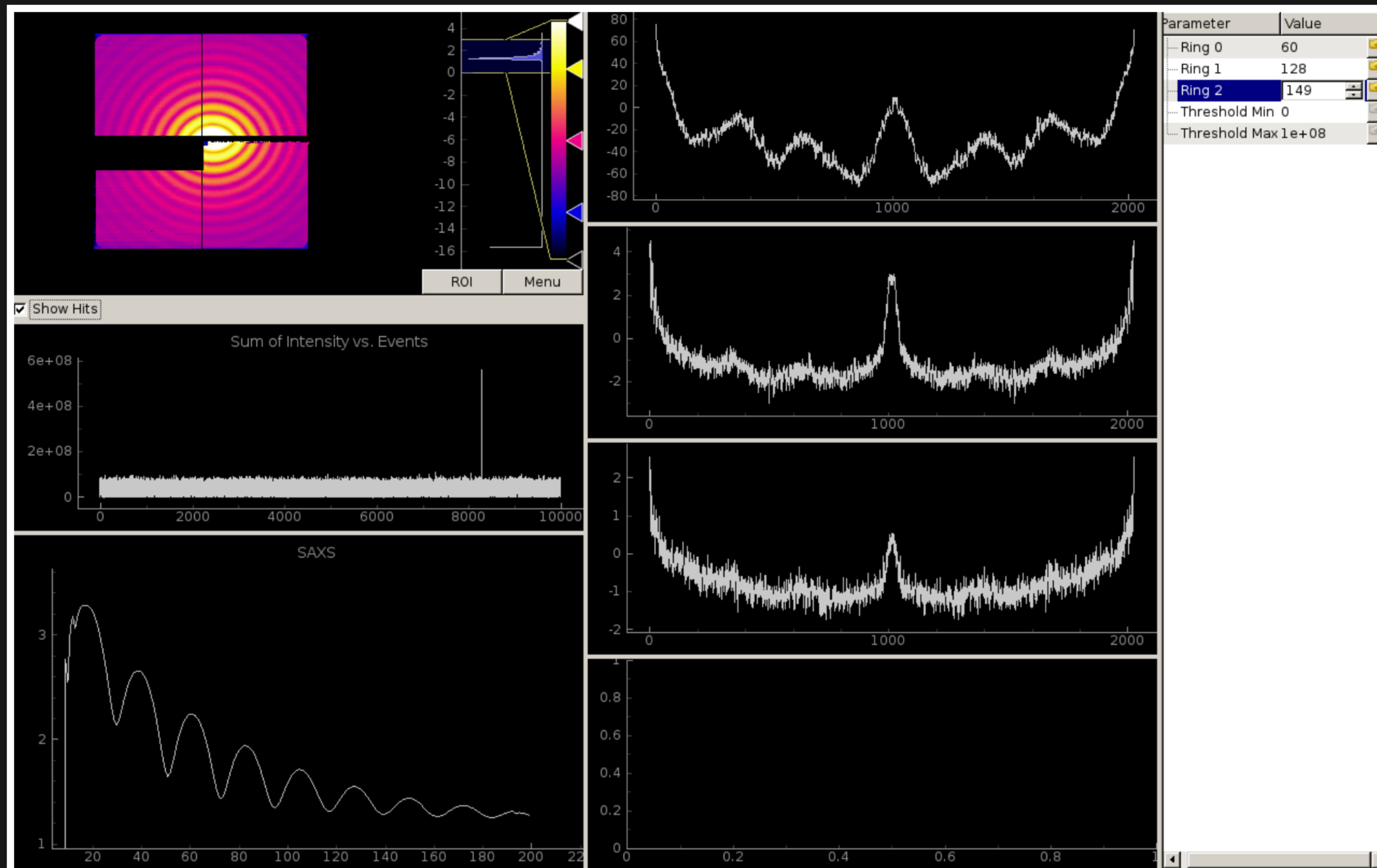
ONDA IN ACTION

ONDA SWAXS



ONDA IN ACTION

ONDA FXS



WHERE CAN ONDA BE USED?

- [Psana](#) (SLAC)
- Manuela Kuhn's [HiDRA](#) (PETRA III but facility-agnostic)
- [Files](#) (HDF5, CBF, NeXus)
- In the future: [Karabo](#) (European XFEL)

SUPPORTED DETECTORS - X-RAY

- CSPAD
- Rayonix
- pnCCD
- PILATUS
- EIGER
- LAMBDA
- ePix
- ePix100
- JUNGFRAU
- AGIPD 1M

SUPPORTED DETECTORS - OTHERS

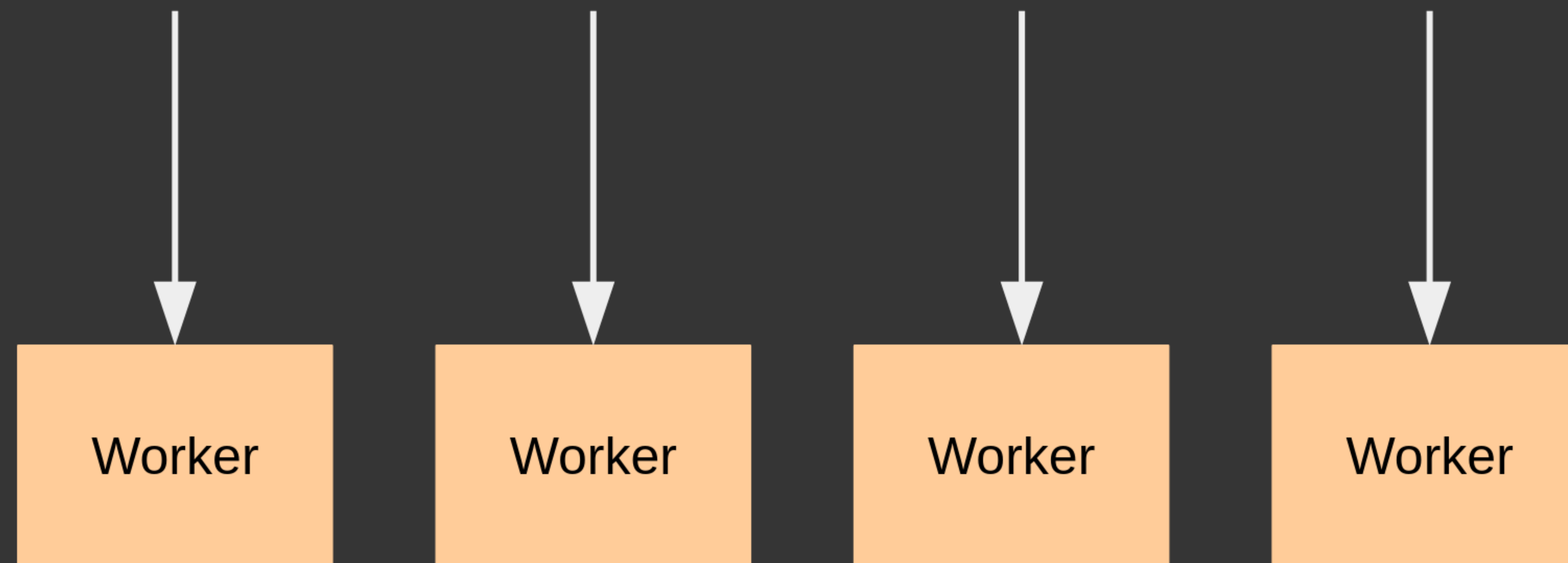
- VMI
- Delay-Line VMI
- Aquiris Digitizers
- Cameras (Opal, etc.)

FUTURE DEVELOPMENTS

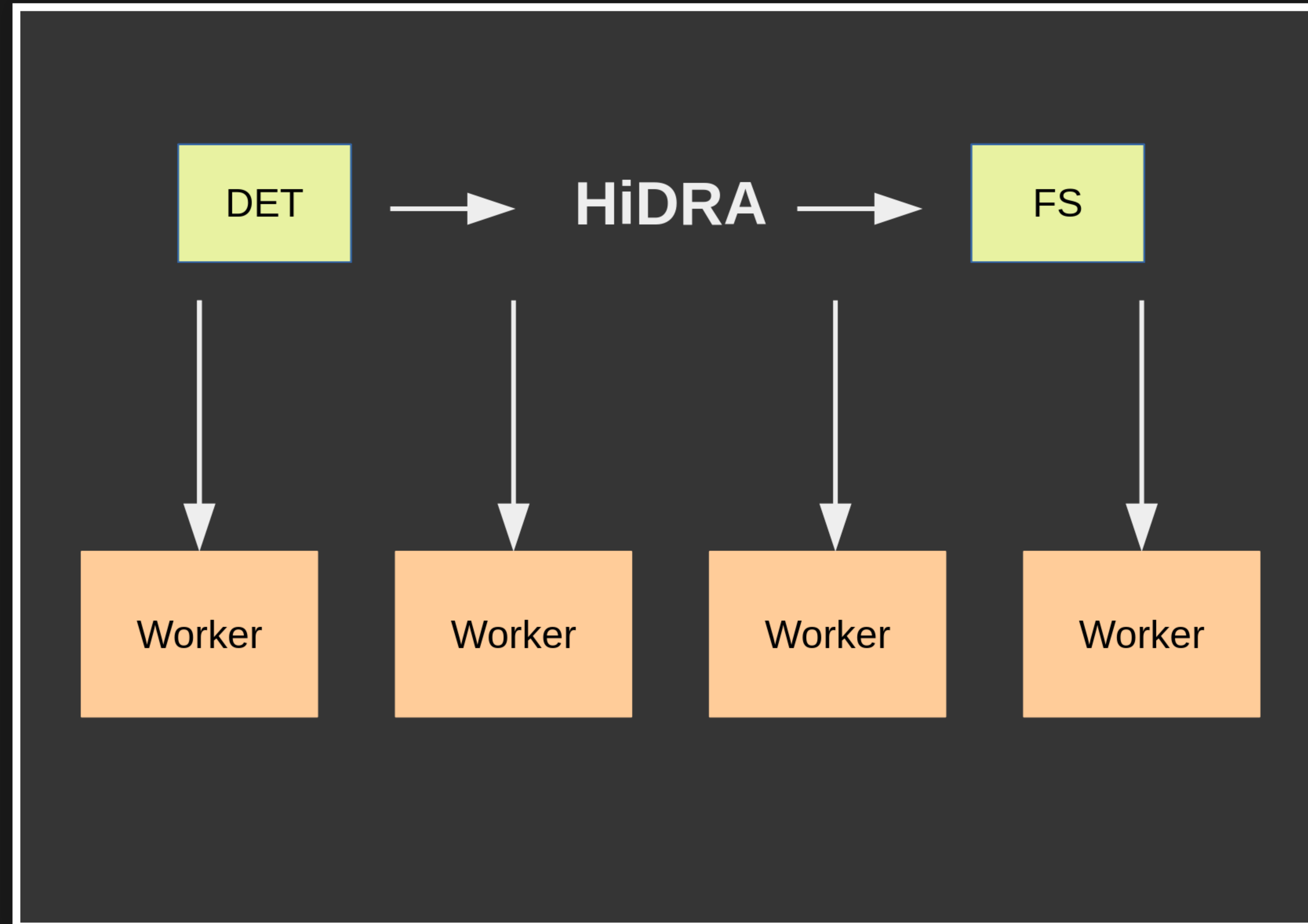
- More detectors (LPD, etc.)
- More features (Streak masking, etc.)
- Run OnDA at scale (Clusters)
- More facilities (ESRF....?)

HDF5 IN ONDA

Facility (Shared Memory, DAQ, etc.)



HDF5 IN ONDA



HDF5 IN ONDA

- HiDRA: file-content agnostic
- Streams binary
- Cannot read HDF5 stream:
 - Path only: Tricky!
 - Ram disk: Clumsy
 - And file needs to be closed!

ACKNOWLEDGEMENTS

HENRY CHAPMAN (CFEL, DESY)

- **CFEL:** Thomas White, Anton Barty, Alexandra Tolstikova, Andrew Morgan, Jochen Kuepper
- **DESY** Manuela Kuhn
- **University At Buffalo:** Thomas Grant
- **Berkeley Lab:** Kanupriya Pande
- **SLAC:** Chris O'Grady, Chun Hong Yoon, Jason Koglin, Thomas J. Lane

<https://www.ondamonitor.com>

Current Release: 19.09.0.1

Mariani et al., "OnDA: online data analysis and feedback for serial X-ray imaging", J. Appl. Cryst. (2016). 49, 1073–1080

