



Streaming Readout for SVT

K. Oyama (RCNP, the University of Osaka)
and
T. Gunji (QNSI, the University of Tokyo)

SVT Working Meeting, Oct. 9, 2026 @ LBNL

SPADI Alliance

Signal processing and data acquisition infrastructure alliance

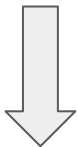


Paradigm Transition to Streaming Readout (SRO)

2

Paradigm Transition from Triggered to Streaming Readout (SRO)

- Modern complex and high-rate, high-lumi. experiments require streaming architectures
 - complex trigger for rare signal under high background
- All data from FEE are **continuously digitized, tagged with high-precision time-stamps**, and pushed upstream = **streaming**
- In streaming, event building and selection (filtering) are executed on computing clusters



- **NestDAQ** provides a standardized, modular, open-source, and lightweight software framework to realize trigger-less streaming readout across diverse physics experiments
 - **RCNP and SPADI-Alliance propose to include the NestDAQ in SVT and ePIC DAQ framework**



SPADI Alliance Initiative

- Community-driven collaborative alliance
- Standardizing all aspects necessary for next decades data acquisition framework i.e. hardware, firmware, software, clock distribution, filtering, logging, control & monitoring, and etc

Present Key Components of the SPADI Framework

- AMANEQ FEE board: FPGA TDC with sub-30 ps resolution
- MIKUMARI & LACCP: clock synchronization & sub-nanosecond time alignment
- **NestDAQ: streaming DAQ engine**
- ARTEMIS: ROOT-based online/offline analysis and QA framework

SPADI Alliance

Signal processing and data acquisition infrastructure alliance

> 200 researchers (32 institute)

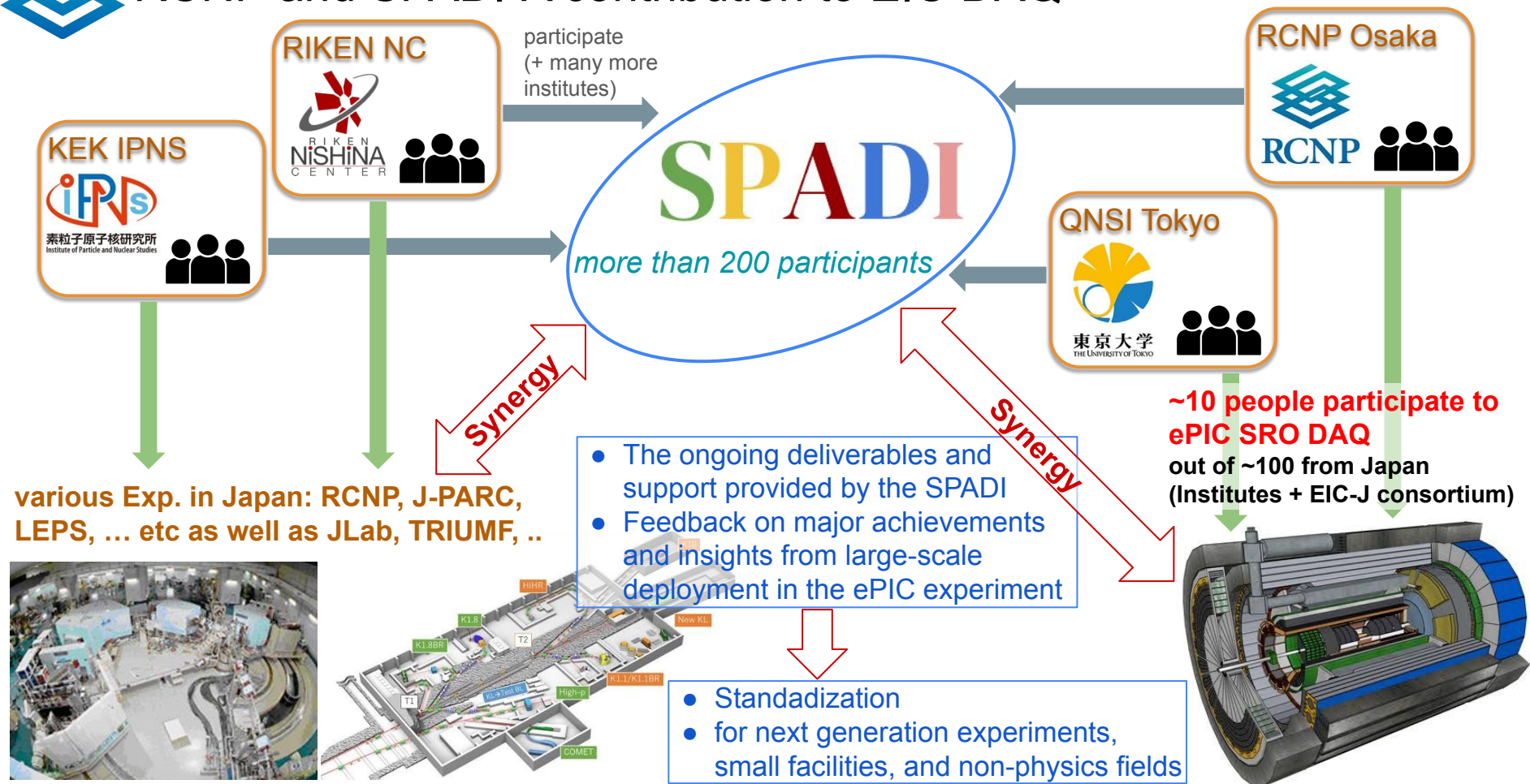


Open also for abroad



RCNP and SPADI-A contribution to EIC DAQ

4

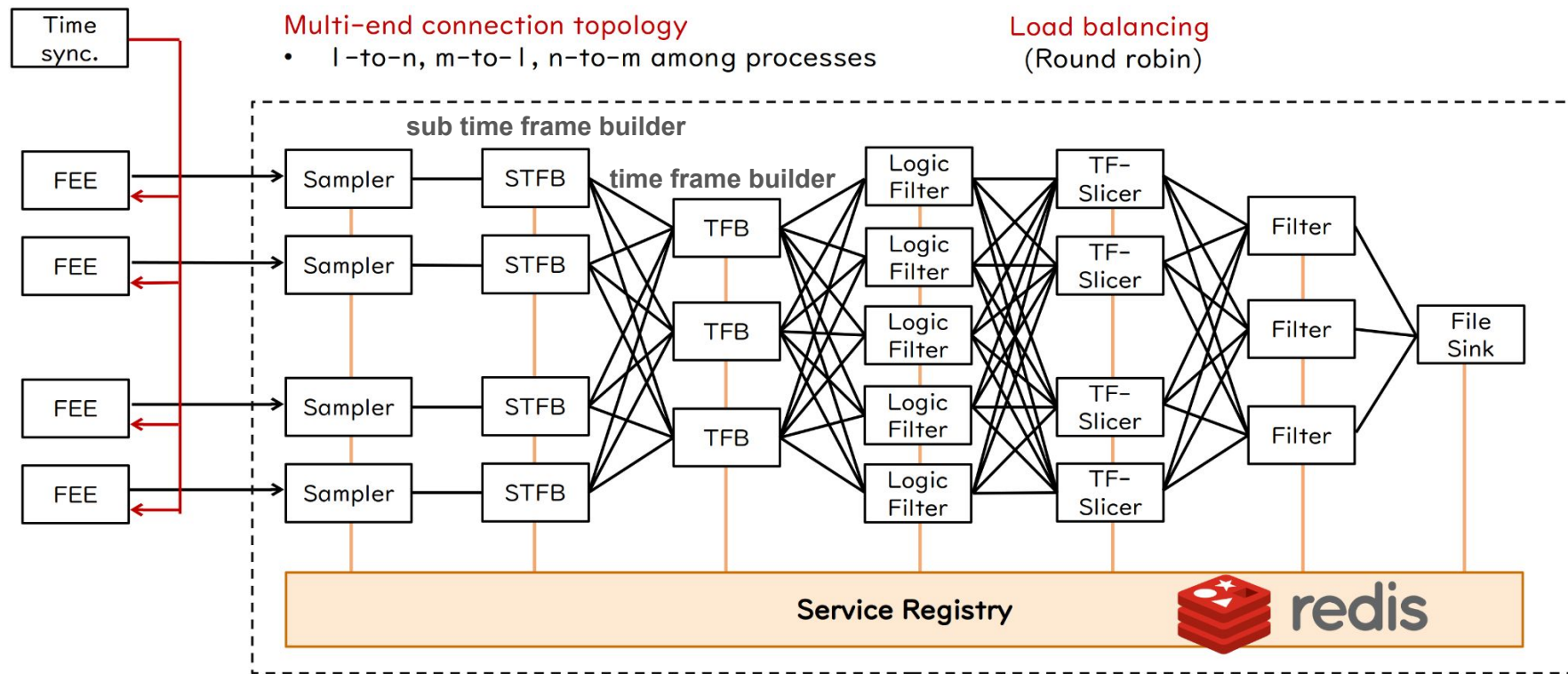




NestDAQ Basic Concept (2)

5

- A highly scalable, network-distributed data acquisition (DAQ) framework
 - dead-time less streaming by multi-staged pipeline data processing model with fully configurable topology





Flexible Network Based Pipeline for High-Performance, Hardware-Triggerless Streaming DAQ

- **Timeframe Based** Data Streaming
 - Synchronized data from detectors over a fixed time interval is treated as a fundamental unit i.e. timeframe
 - Essential that **SVT MAPS data does so** (discuss later)
- **Software-Based High Level Filtering** Instead of Hardware Trigger
 - Timeframes gathered from multiple detectors are filtered by a software trigger algorithm
 - Allowing any complex trigger logic as it is software trigger → drastic data reduction prior to writing
 - Scalable by increasing number of WNs according to computing complexity
 - Multiple stages filtering
 - GPU offload
- **Load Balancing & Scalability**
 - High-throughput data streaming distributed dynamically across multiple WNs
- **Observability**
 - Integrated logging, real-time data flow metrics, error analysis, and anomaly detection for reliable operation.
- **Extensibility**
 - Agnostic to underlying front-end hardware; adaptable to various detector FEEs
 - **SVT specific input interface (firmware, device driver & software) to be prepared**
 - e.g. Describing the “Sampler” component allows users to accommodate any readout interface.



bTOF test beam experiment with NestDAQ (cont.)

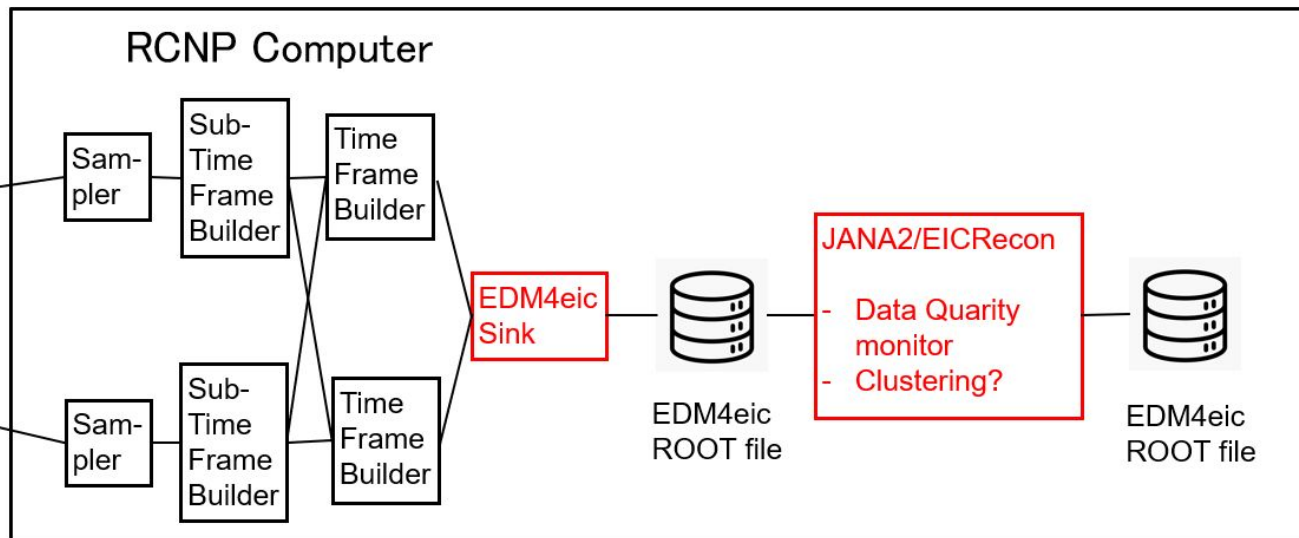
7

Courtesy of T. Kumaoka

Established and demonstrated an end-to-end data path from NestDAQ to EICRecon

- data produced by NestDAQ are transferred via ZeroMQ, converted into the EDM4eic event data model, and read by EICRecon successfully
 - EDM4eic&podio library linked in EDM4eicSink process
 - NestDAQ run in EICrecon container

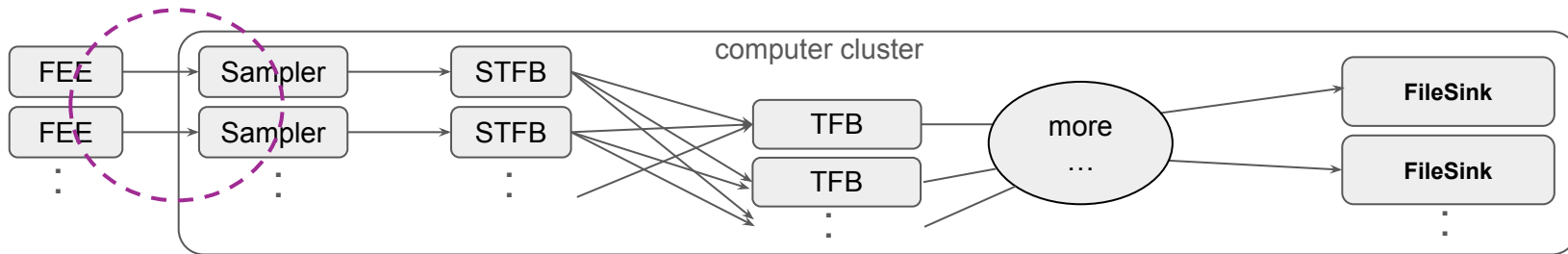
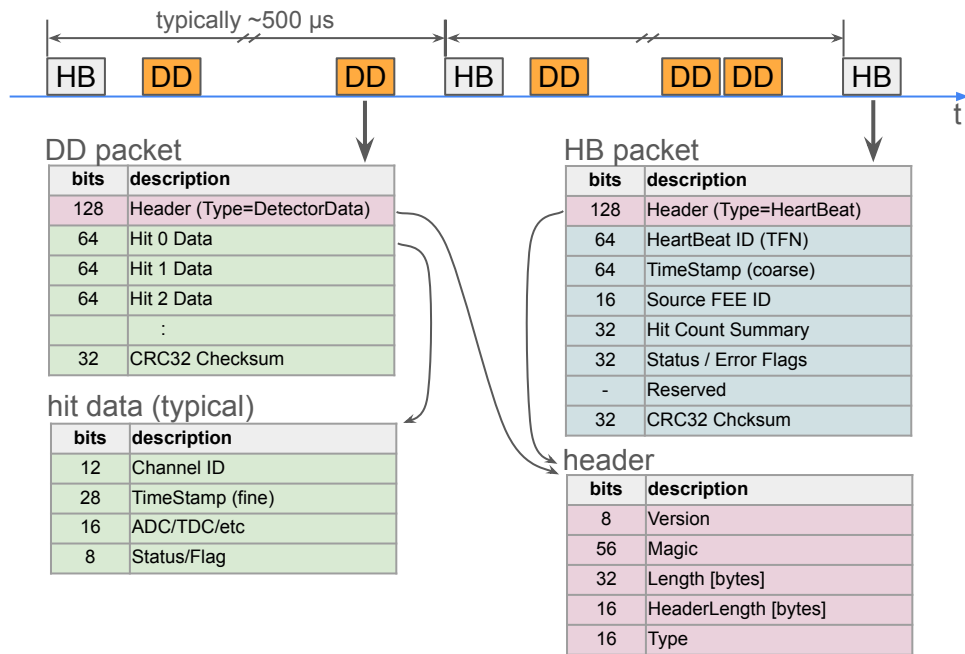
AMANEQ HR-TDC



Data Format and Processing: Detector Data Sampling

FEE sends two types of data packets to “Sampler”

- HeartBeat packet (HB)
 - periodically sent as timeframe divider
 - Header + lightweight payload
- Detector Data packet (DD)
 - interleaved between two HB packets
 - ADC/TDC etc.
 - Header + payload
 - payload format is detector specific but it **must include fine timestamp**
 - packet drop reported by the next HB
- Presently: all FEEs speak TCP/IP UDP**

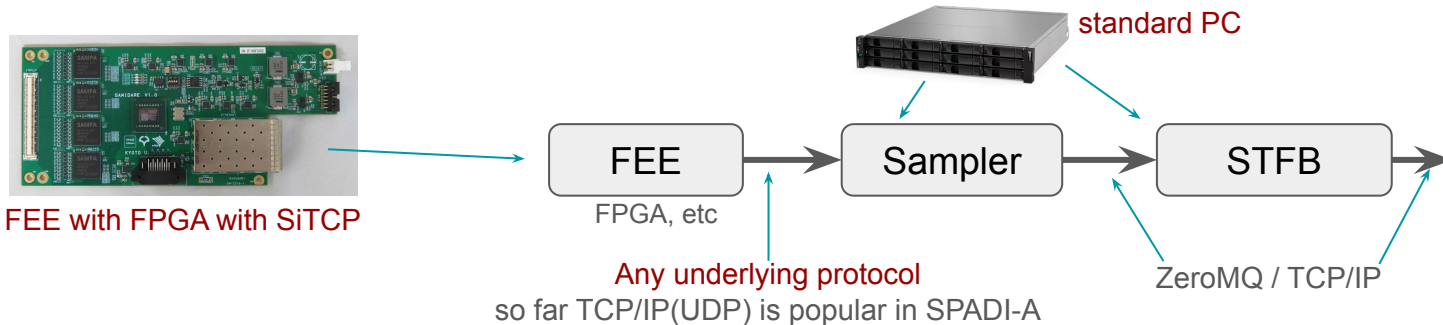




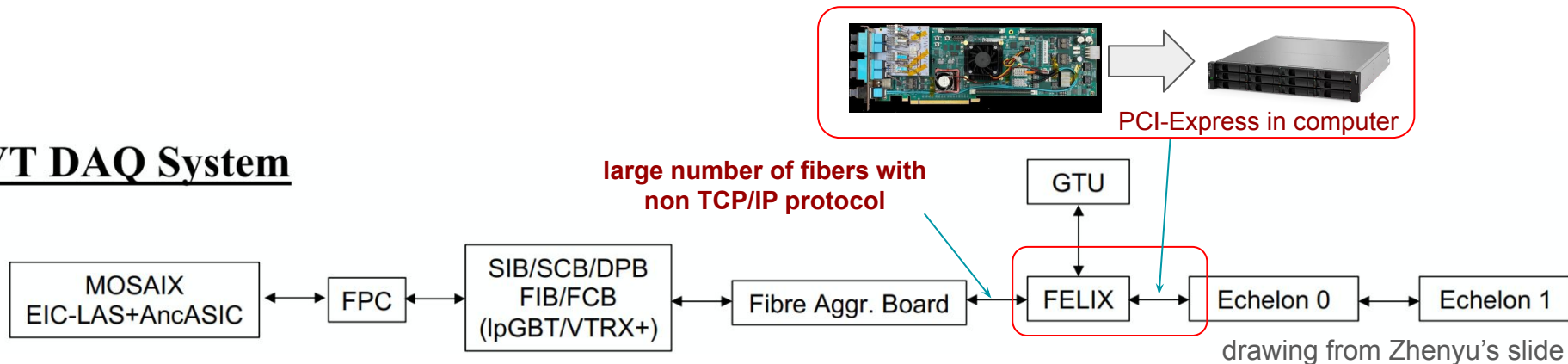
Our interest: NestDAQ Adaptation for SVT

9

- So far, NestDAQ have experiences only with FEEs speaking TCP/IP
- This is not the case in SVT and many of recent high channel density detectors with FPGA card to receive detector & ASIC data



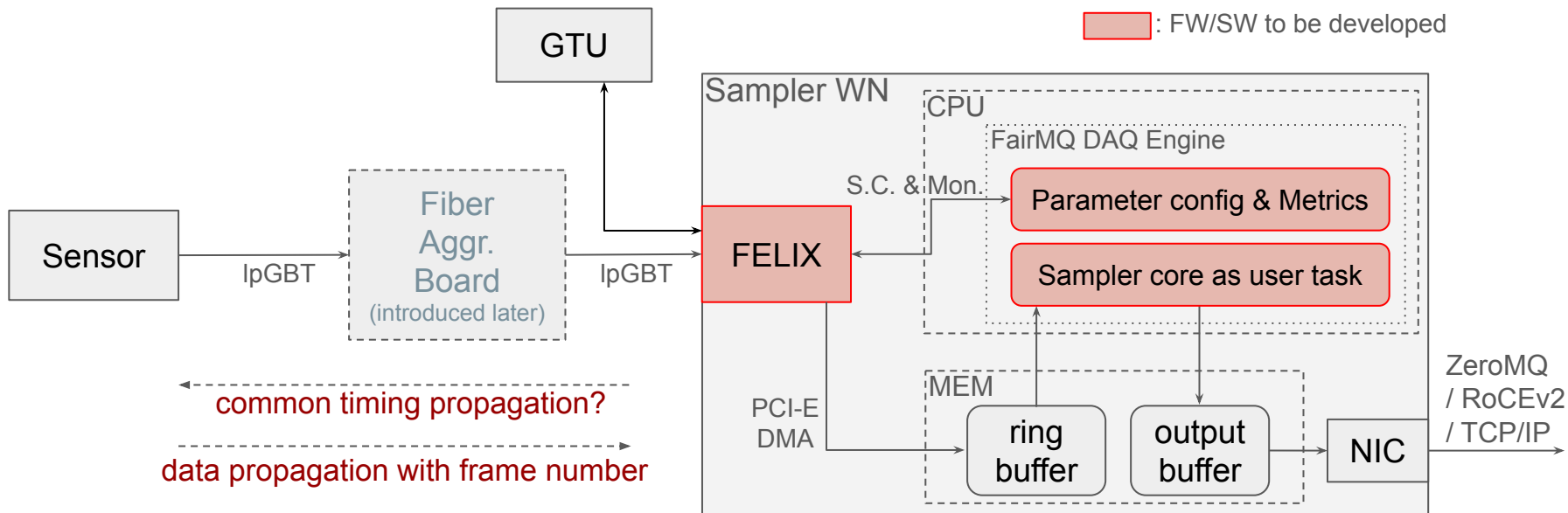
SVT DAQ System



drawing from Zhenyu's slide



- Develop “Sampler” to read data from FELIX via PCI-Express instead of NIC
 - device driver and user code to read data via IpGBT and put in mem.
 - similar experience in J-PARC E16 by T. Takahashi
 - the heartbeat packet and timestamp attached to the data should be produced before non-deterministic latency buffer → MOSAIX frame counter is ok if it is synchronized
 - FELIX to attach absolute time-stamp to data, then forward data to CPU



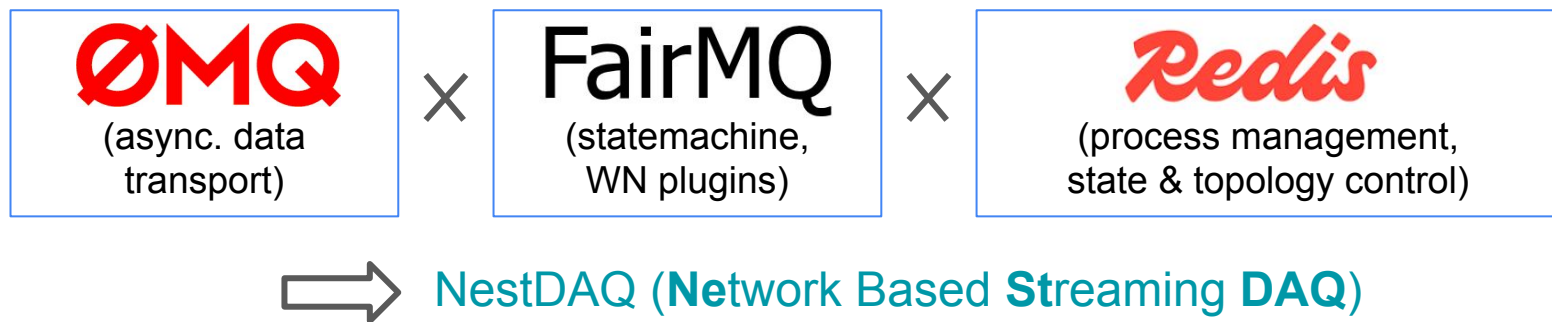


NestDAQ

- NestDAQ provides a lightweight, highly scalable, and trigger-less streaming DAQ engine for modern physics experiments
- Strong support of community via SPADI-Alliance framework in Japan
- Built on ZeroMQ, FairMQ, and Redis, it achieves zero-copy data transport, transparent multi-stage software filtering, and dynamic topology control.
- Proven through successful high-rate beam tests (RCNP, J-PARC) and end-to-end data processing integrated with EICRecon.

NestDAQ for SVT

- For SVT by developing FELIX firmware and Sampler user code, NestDAQ can be adapted
- However, careful consideration and implementation where attach timestamp to data is important
- *Not discussed*: filtering in Echelon0 stage is also to be discussed (GPU offload as well)



- Message-Queue Driven Architecture
 - Built upon distributed message-passing middleware (FairMQ / ZeroMQ) for multi-process communication.
- Dynamic Topology & Configuration Management
 - NoSQL/Key-Value database (Redis) for
 - centralized configuration,
 - state control (Pub/Sub),
 - automatic topology construction, and
 - monitoring all processes of distributed computers.

See presentation by T. Takahashi in 14th edition of the Streaming Readout (SRO-XIV) workshop for detail
<https://indico.frib.msu.edu/event/96/overview>

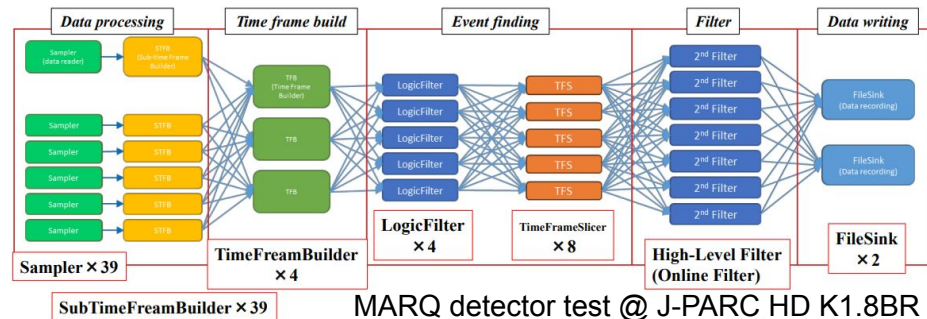
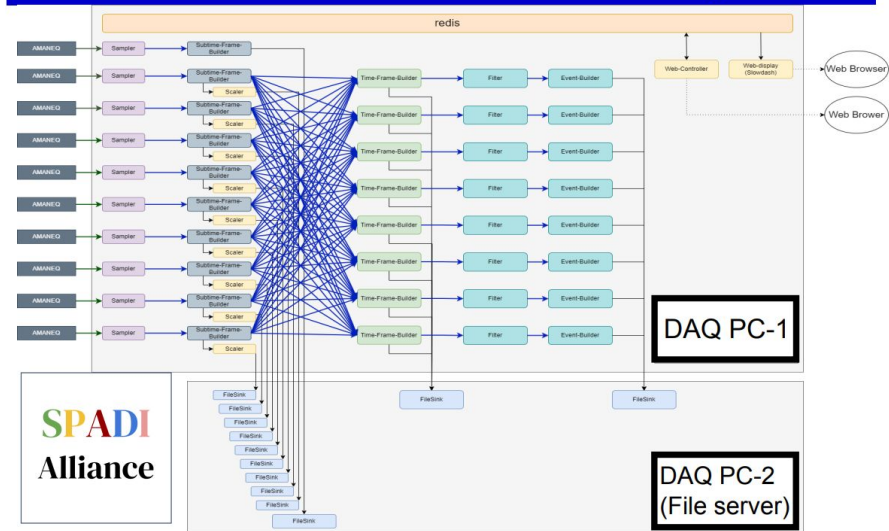


Field Demonstrations at RCNP & J-PARC

x40 improvement in rate capability w.r.t. traditional triggered systems, with sustained data throughput surpassing ~1 GB/s.

- RCNP Grand Raiden & Large Acceptance Spectrograph
 - Throughput: ~300 Mbps continuous streaming readout.
 - Event Rate: Achieved 150 - 200 kcps (>10× faster than legacy VME systems), enabling rare
- J-PARC Hadron Facility (E50 T103 Test Experiment)
 - Stable operation at 260 MB/s sustained average transfer rate over 10-hour hadron beam extraction without data loss.
 - Network Scalability: Multi-server configuration demonstrated load-balanced transfer up to 9.2 Gbps.
 - 2 stages HLF achieved 600 MB/s processing throughput with 1750 MB/s TOF calculation speed.

RCNP GR/WS E585 software configuration

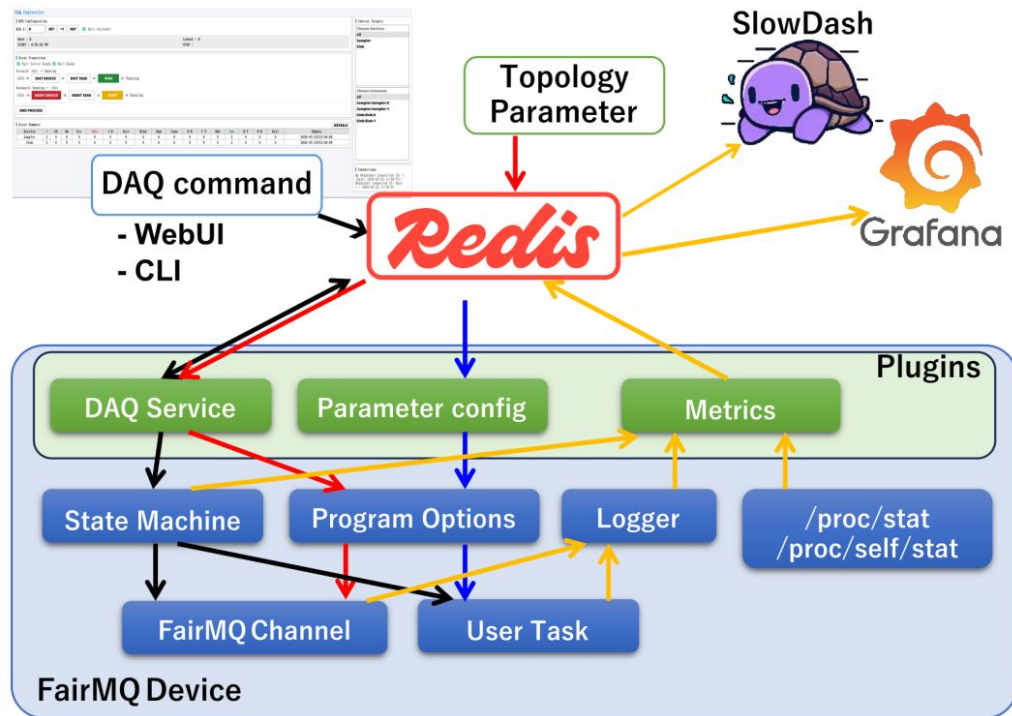


MARQ detector test @ J-PARC HD K1.8BR



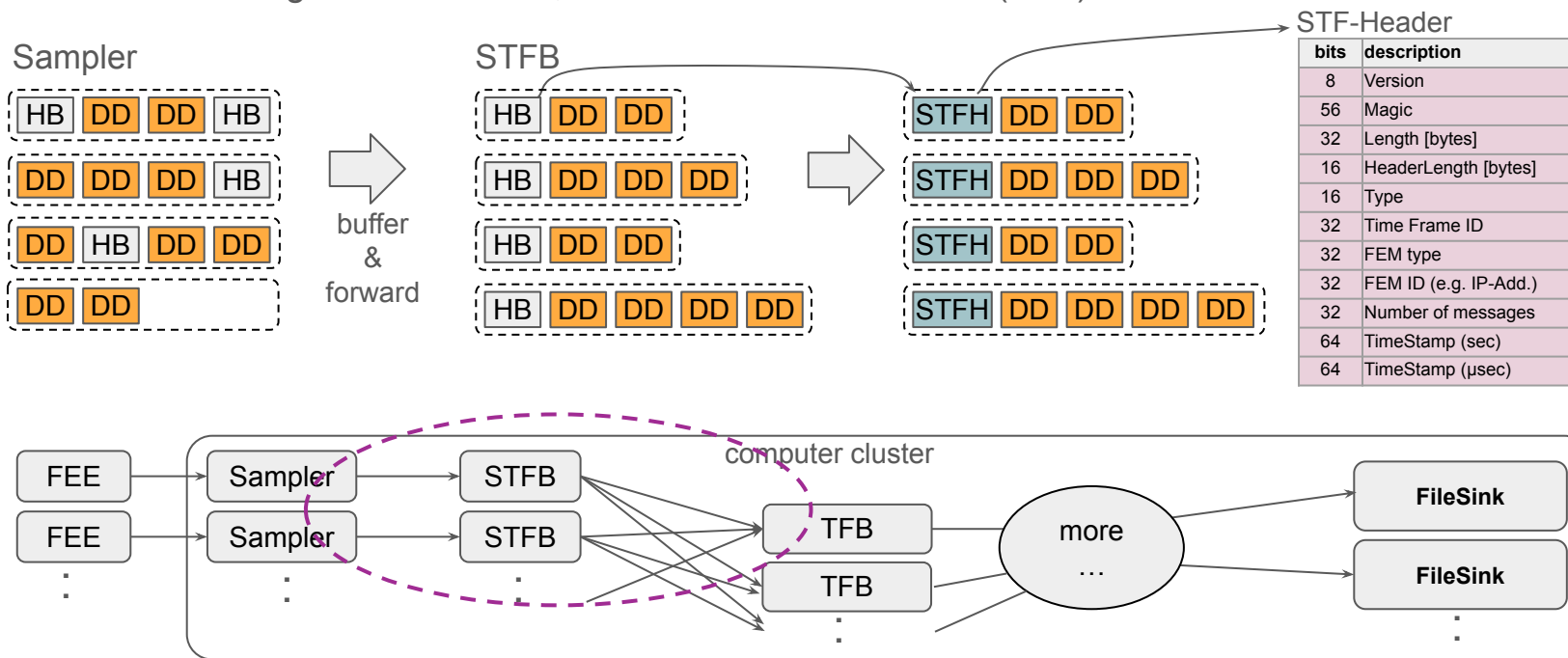
FairMQ with Plugins Device as DAQ engine of all WNs

- DAQ Service Plugin
 - Run control
 - user Message Queue
 - Control the state machine
 - Service discovery
 - use KVS (key value store) function
 - Semi-automatic connection configuration
- Metrics Plugin
 - use Time Series DB
 - Grasping the processes statuses
 - Visualization using Grafana or any
- Parameter config Plugin
 - user KVS
 - Read device initialization parameters from the database.



Data Format and Processing: SubTimeFrame Building

- Sampler captures HB and DD, buffers them, then push them to STFB as ZeroMQ multipart message (zero copy, no sub timeframe constructed yet) without considering timeframe border yet
- “SubTimeFrameBuilder (STFB)”, prepared for each Sampler, build up sub-timeframe, attach a STF-Header using HB information, send to TimeFrameBuilder(TFB)

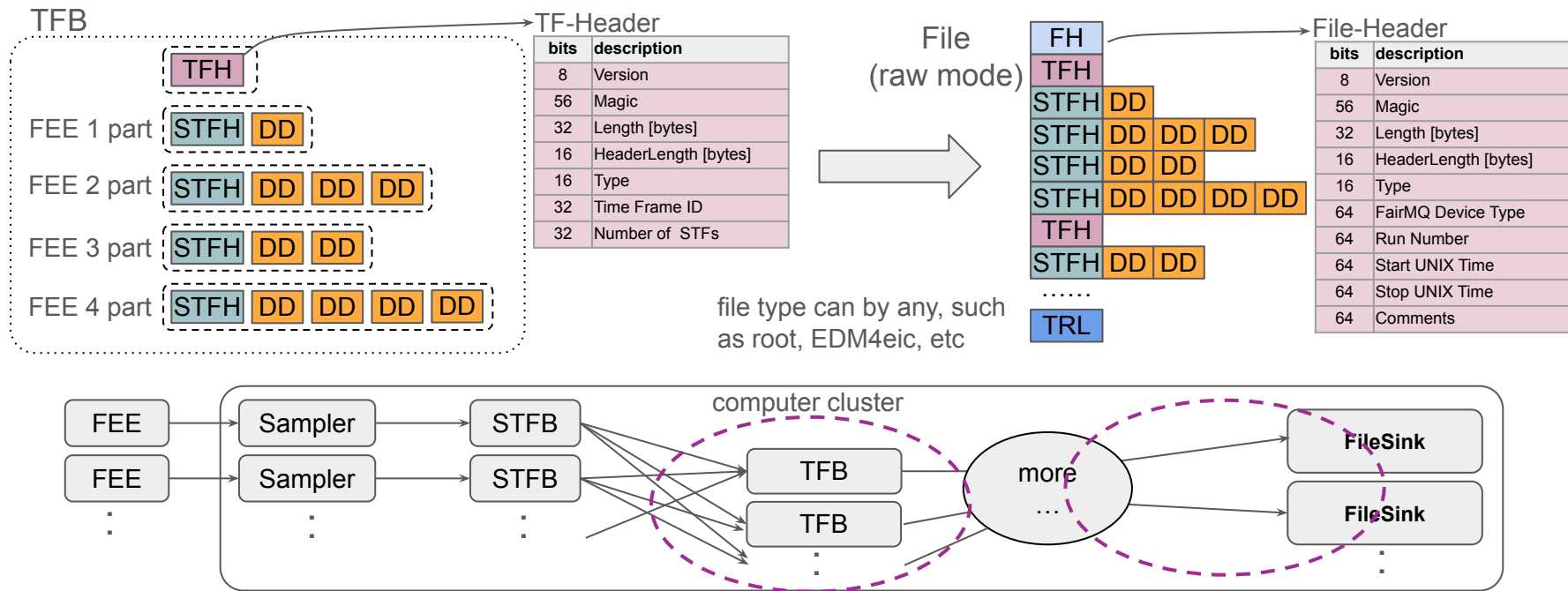




Data Format and Processing: TimeFrame Building

16

- TFB(Time Frame Builder) receives STF from all subsystem STFBs but for the same time interval
 - round robin (implemented) or queue-based load balancing (in preparation)
 - Zero-copy and data transparency are always kept
- Finally, data is sent to FileSink, and ZeroMQ multipart message is finally cupsulated, written to file.

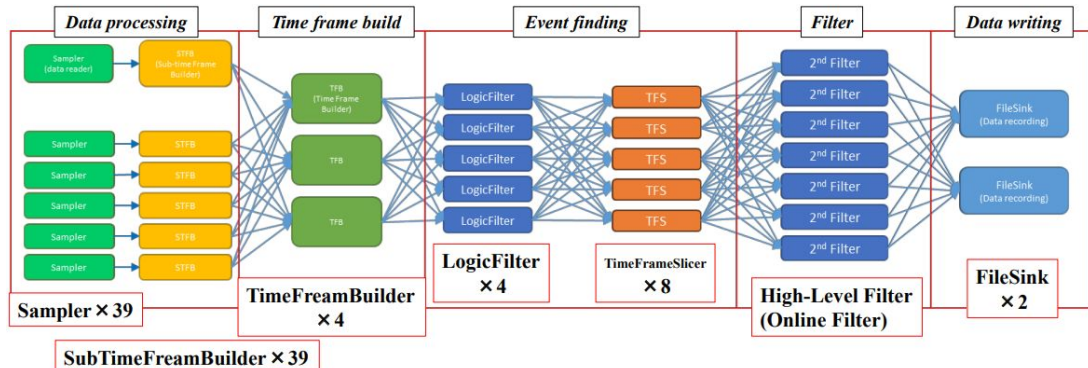




Data Format and Processing: Filtering & Slicing

17

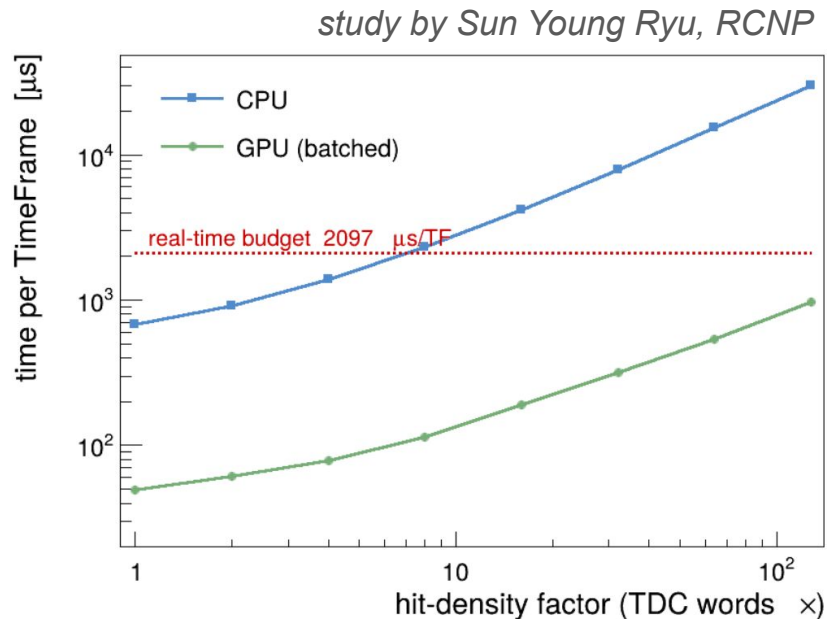
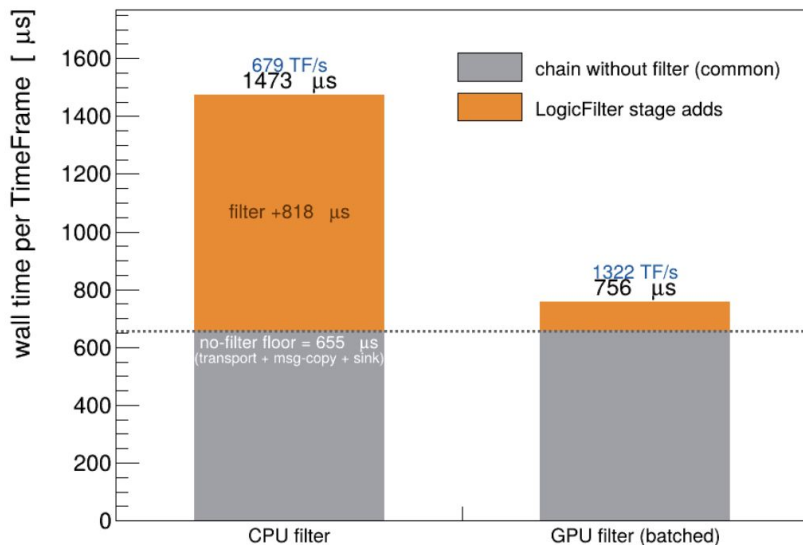
- Any number of Filter Processors can be inserted between the TFB and FileSink to enable software triggering, data compression, and reduction
 - e.g. MARQ test exp. at J-PARC implemented two levels
 - Logic Filter: low level fast selection such as coincidence, number of hits (LUT)
 - remove unnecessary detector data
 - attach trigger decision information as meta part of ZeroMQ
 - 2nd Filter (High Level Filter): clustering, tracking, waveform fitting, etc
- The Time Frame Slicer (TFS) extracts specific time windows of interest from large Time Frames based on logic filter decision.
 - It slices continuous raw data streams around target events without modifying payload structures, drastically reducing data volume and enabling efficient, zero-copy pipeline processing for further downstream



MARQ detector test
@ J-PARC HD K1.8BR



- GPU Acceleration for Logic Filter
 - implementation tested on E50 data
 - 4 ns slice bin $\rightarrow$ 1 thread of GPU for each time bin of TF (131k bins), apply LUT
 - GPU >10 times faster than CPU for filter calculations
 - CPU: i7-14700K/(8P+12E), 5.6GHz, 64GB
 - GPU: RTX4060Ti 16GB, 4352 CUDA



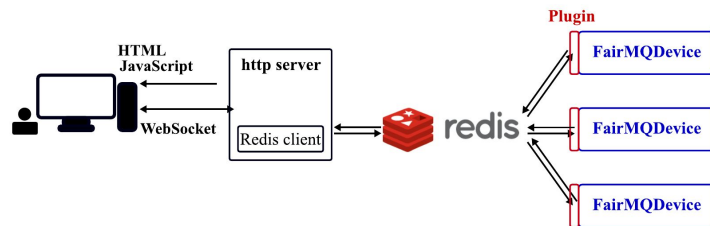
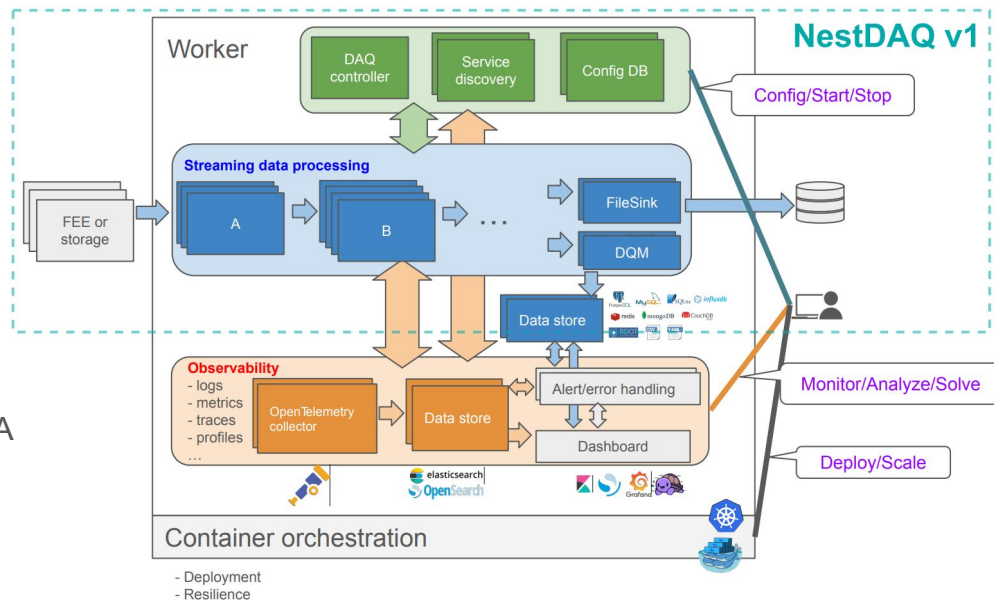


Future Roadmaps of NestDAQ

19

- Widespread Facility Deployment
 - Operational across Japanese hadron/nuclear facilities: RCNP, J-PARC, RARiS, and HIMAC.
 - JLab beam dump experiment, TRIUMF TUCAN and EIC
- Ongoing Developments
 - Enhancing containerized deployment (container)
 - Scheduler update (queue-based load balancing)
 - Observability (logs, metrics, visualization)
 - Orchestration and Agentic-AI driven control
 - Integrating heterogeneous computing (GPU, FPGA acceleration) for real-time online event reconstruction and AI-driven filtering
 - **Zero copy by RDMA** (RoCEv2) as transport layer and GPU filtering by GPUDirect
 - replace ZeroMQ with libfabric?
 - Partitioning & multiple actors
 - and more ...

NestDAQ v2

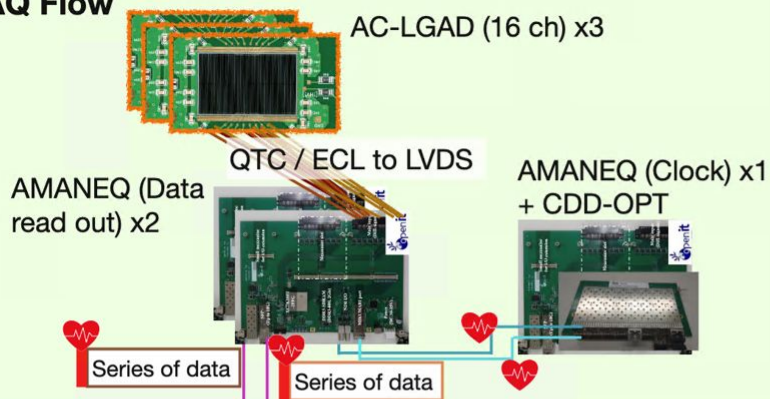




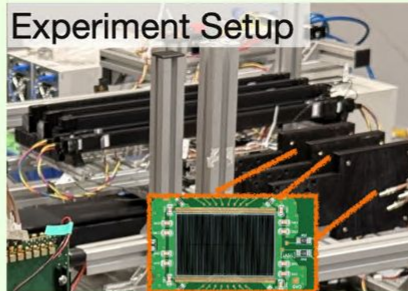
Courtesy of T. Kumaoka

Tried mini ePIC streaming orchestration (Echelon0-Echelon2)

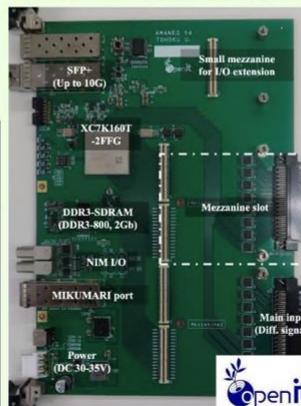
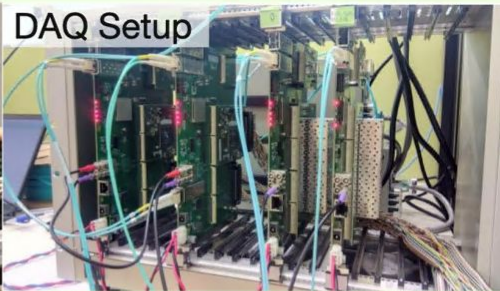
DAQ Flow



Experiment Setup



DAQ Setup



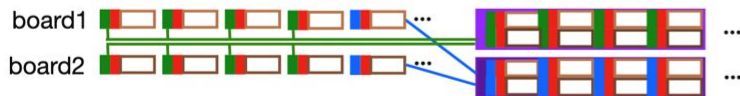
AMANEQ

(A main electronics for network oriented trigger-less data acquisition system)

- Kintex7 with speed grade -2
 - Transceiver bandwidth up to 10 Gbps (10 GbE)
- Two mezzanine slot for functionality extension
 - HR-TDC
 - Clock distribution
 - Input-channel extension
 - etc...
- A jitter cleaner (CDCE62002)
- DDR3-SDRAM as a de-randomizer

Desktop PC (NestDAQ)

- Sampler
- SubTimeFrameBuilder - TimeFrameBuilder



- Output

1. File sink (EDM4eic) → XRootD → Univ Osaka
2. Send a data as message queue

<https://agenda.infn.it/event/47630/contributions/279107/attachments/143691/218225/sroxiii-honda.pdf>