DAQ System for CEPC Detector

Xiaolu Ji

On behalf of CEPC Detector TDAQ group

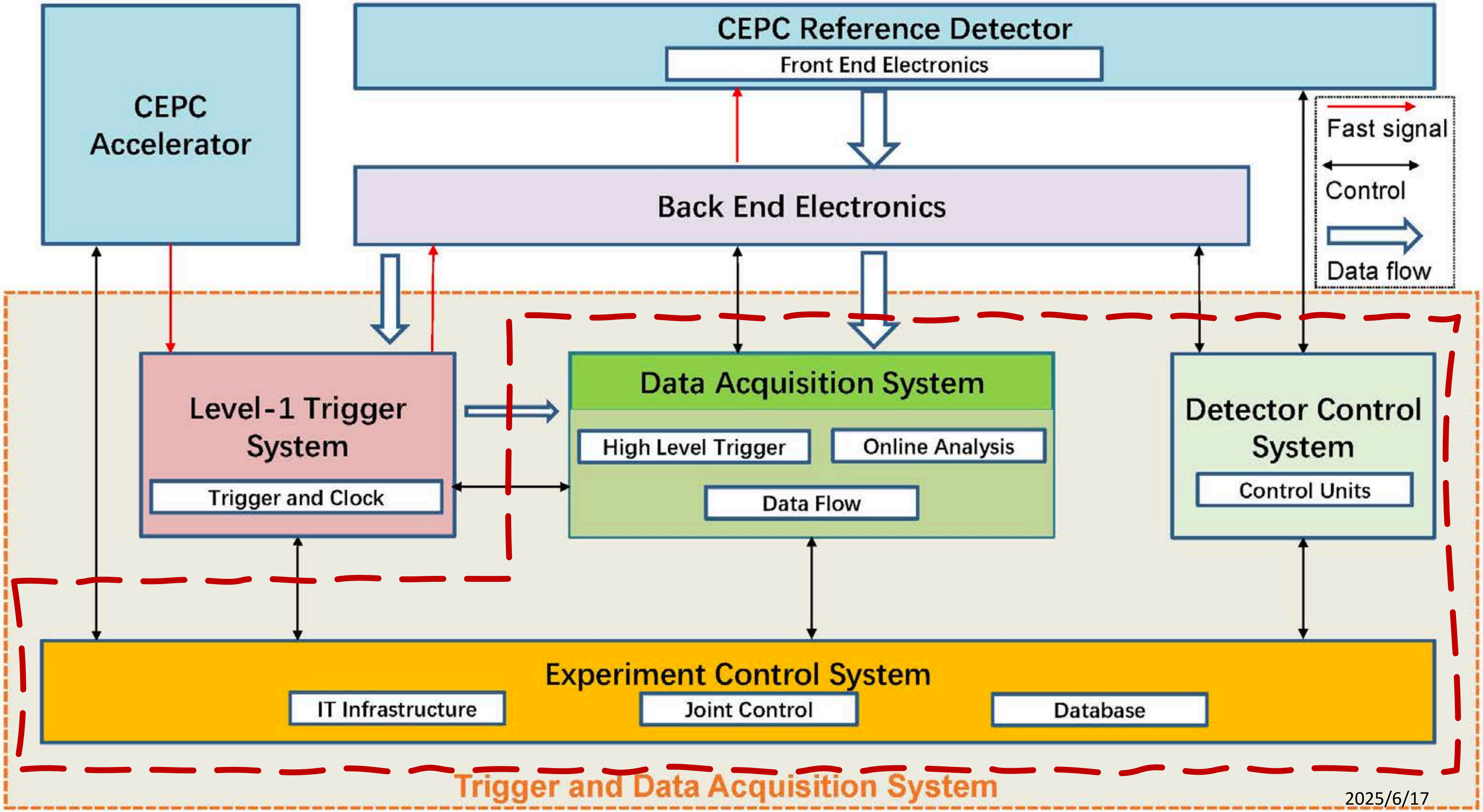
2025.6.17 Barcelona

CEPC 2025 International Workshop

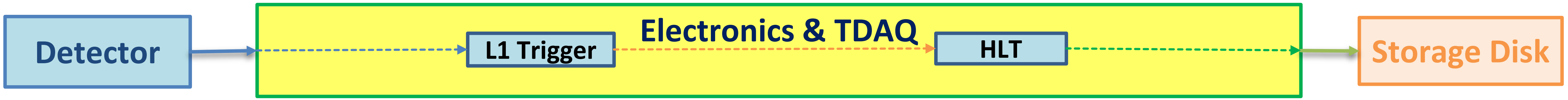
Outline

- **DAQ**
  - Requirement
  - Architecture design
  - Software (Radar) development
    - V2: ✔ implemented features
    - V3: ⌛ongoing research activities
- **DCS**
- **ECS**
- **Summary**

Overall Design of TDAQ and Online



CEPC Detector Requirement – Data Rate



Bunch crossing rate(Beam background data rate)	L1 trigger rate(Readout data rate)	HLT rate(Storage rate)
Higgs mode: 1.3 MHz (600 GB/s)	50 kHz(49 GB/s)	1 kHz(1.5 GB/s)
Low Lumi. Z: 12 MHz (1.4 TB/s)	120 kHz(73 GB/s)	20 kHz(16 GB/s)
High Lumi. Z: 39.4 MHz (4.0 TB/s)	400 kHz(555 GB/s)	80 kHz(163 GB/s)

* Event storage ratio for bkg/phys < 1

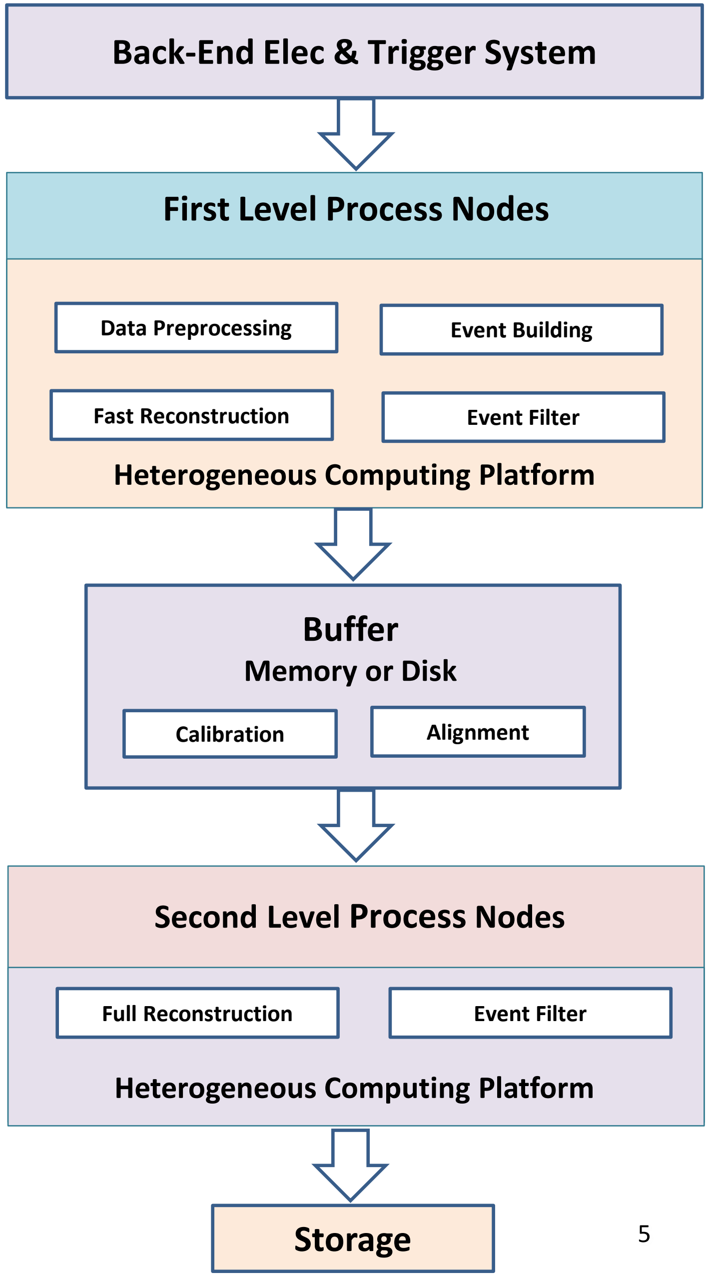
Table 12.4: Trigger rate estimation table for different run conditions. The physical event rate is introduced in Section 12.1.1.1. The detailed readout rates of each sub-detector are shown in Table 12.12.

Operation phase / **Condition**	**Higgs**	**I** / **Z (12.1 MW)**	**W**	**II** / **Z (50 MW)**	**III** / **$t\bar{t}$**
Non-empty bunch crossing rate(MHz)	1.34	12	6.5	39.4	0.17
Luminosity ($10^{34}/cm^2/s$)	8.3	26	26.7	95.2	0.8
Physical event rate (kHz)	0.5	10	1.1	40	5.7×10^{-2}
L1 triger rate (kHz)	50	120	65	400	2
DAQ readout rate (Gbyte/s)	49.0	72.8	-	555	-
HLT rate (kHz)	1	20	2	80	1
Raw event size (kbyte)	1453.5	801.1	1500	2042	1000
DAQ storage rate (Gbyte/s)	1.5	16	3	163	1

- **Readout** 50-550 GB/s
- **Storage** 1.5-160 GB/s
- **Online processing** (HLT included)
- ***Scalable, flexible*** for possible **full software trigger upgrade**

DAQ Architecture Design

- ✓ **Seamless HW/SW trigger compatibility**
  - Smooth upgrade to full-software triggering
- **Infrastructure Foundation**
  - Full COTS(commercial-off-the-shelf) ***hardware***
  - RADAR ***software framework***
  - Heterogeneous ***computing platform***
- **Two-stage HLT design**
  - 1st: fast reconstruction
  - 2nd: full reconstruction with online calibration
- **Decoupled scheme**
  - Memory/disk buffering pools
  - Environment-agnostic compute
  - Online integration of offline algorithms



Streaming Readout Framework – Radar

heteRogeneous Architecture of Data Acquisition and pRocessing

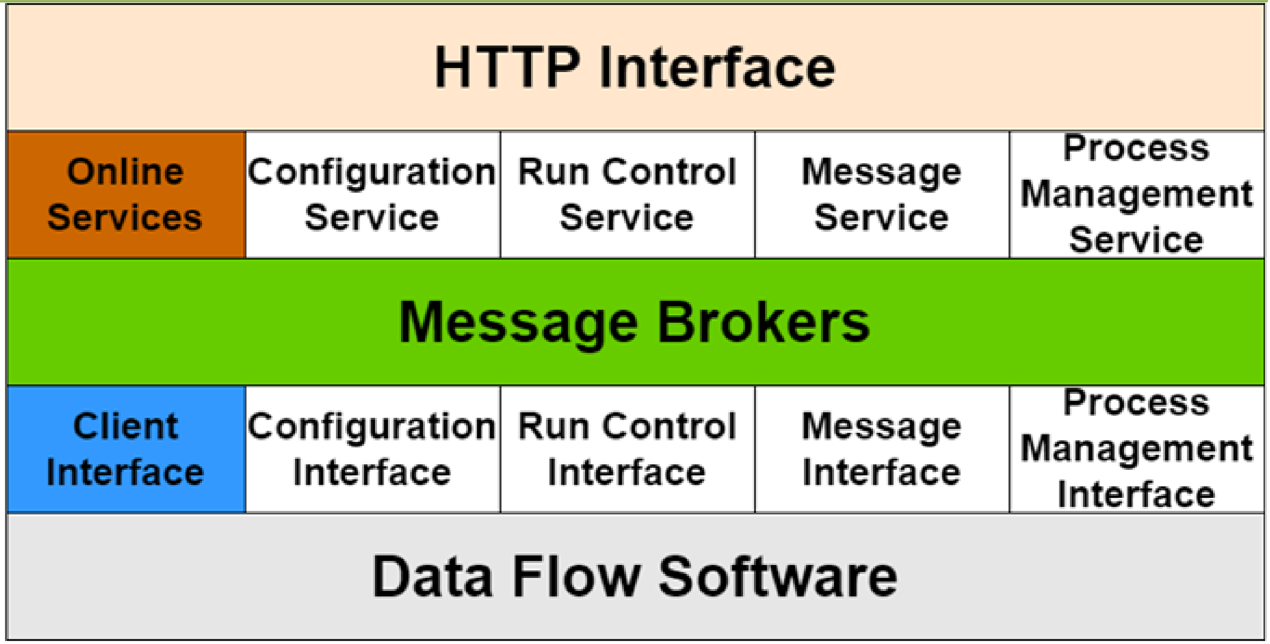
Version	Deployment	Data rate	Trigger mode	Key updates
V1	LHAASO	~ 5 GB/s	SW Only	- Baseline implementation
V2	JUNO	~ 50 GB/s	Hybrid	• Containerized • Partial HA
V3 (Dev)	CEPC detector (Future)	~ TB/s	Default HW + Possible SW	• Heterogeneous processing • Full HA

Introduction to Radar V2

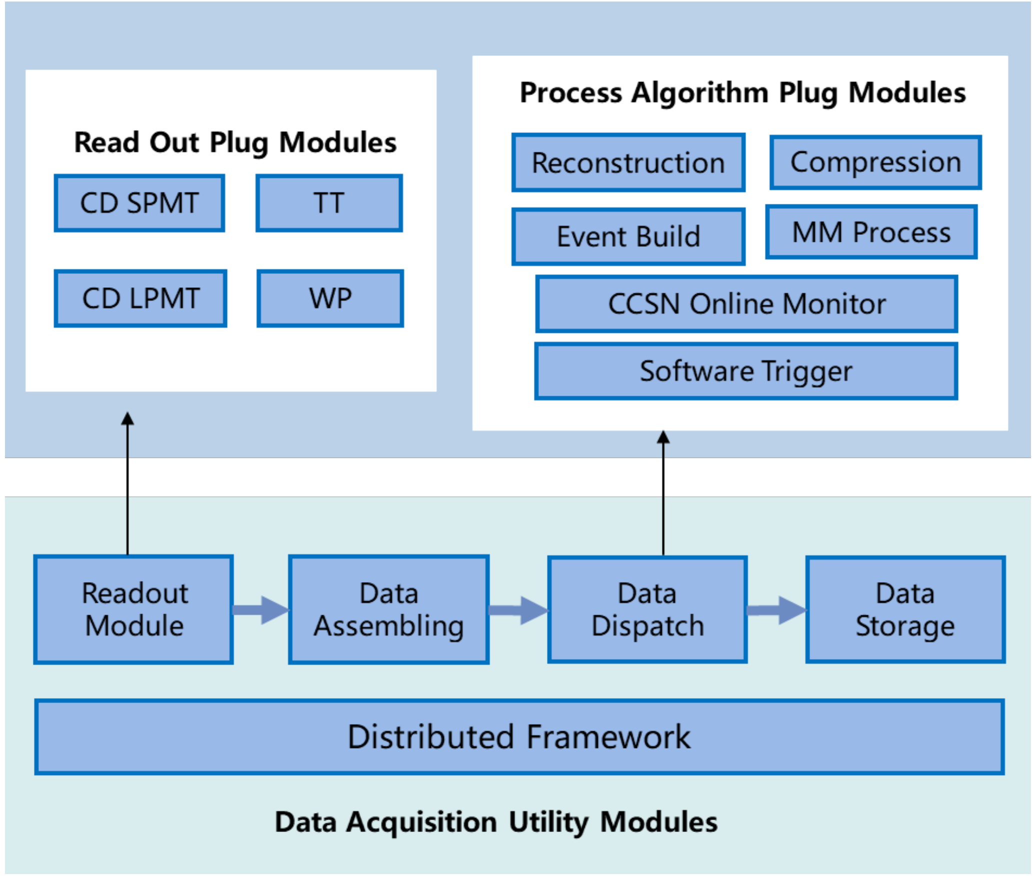
General-purpose distributed framework

- Transport layer – ZeroMQ
- Services – Kafka / ZooKeeper based

Online Software: management & services

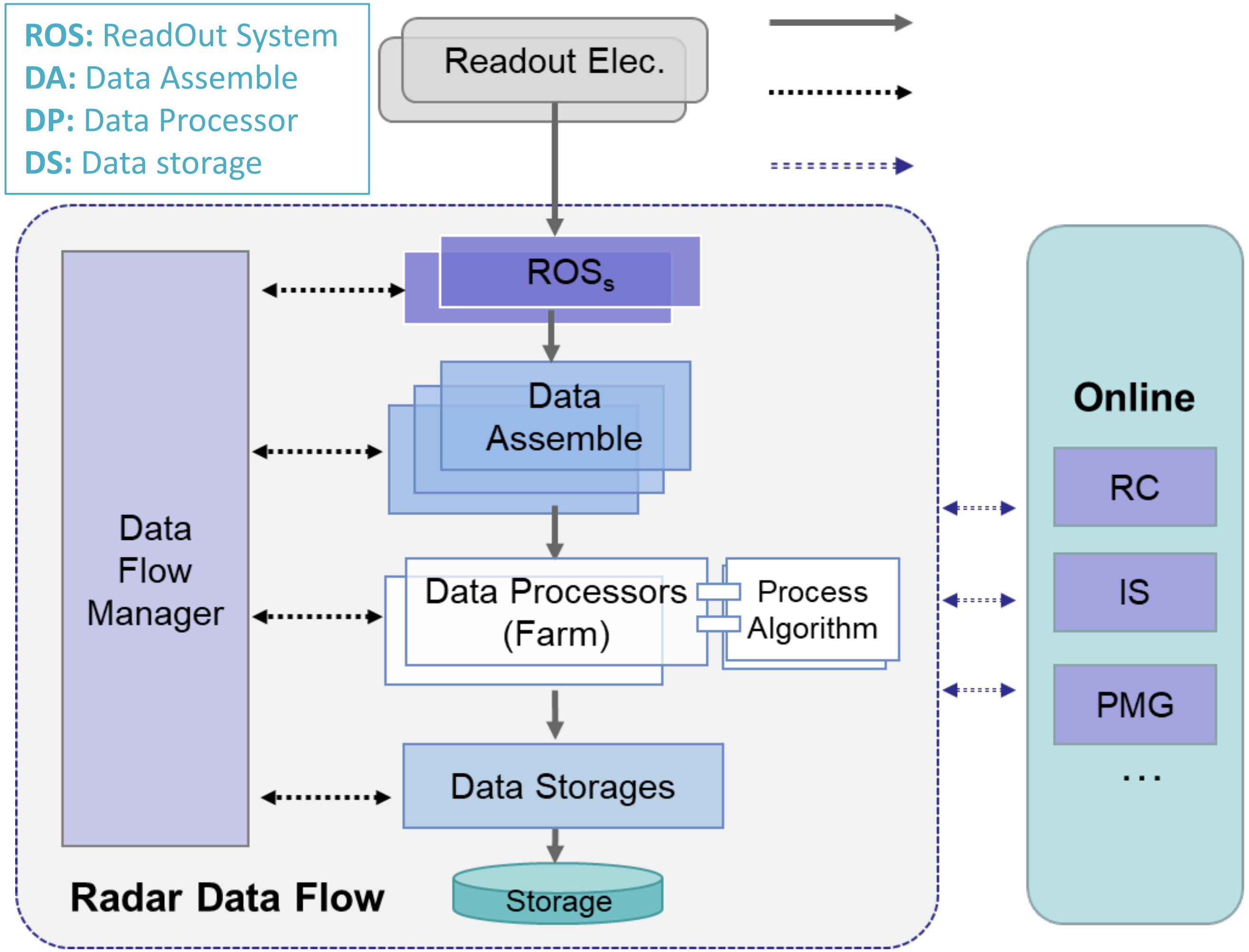


Data Flow Software: process data



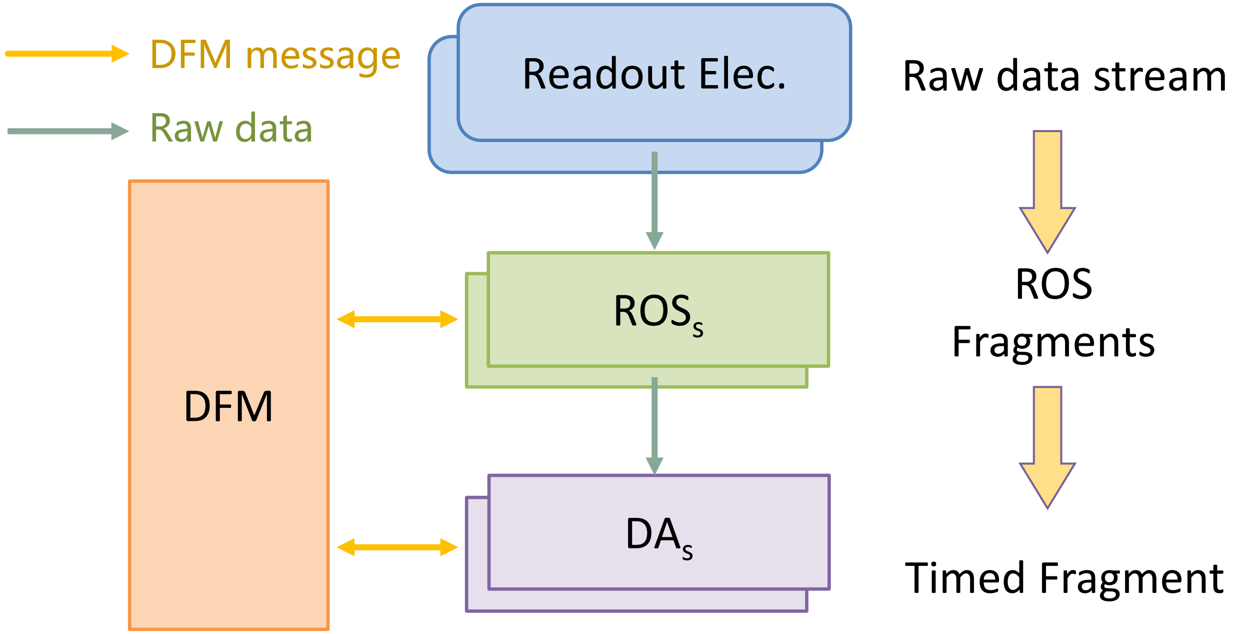
Running at JUNO for > 0.5y for commissioning

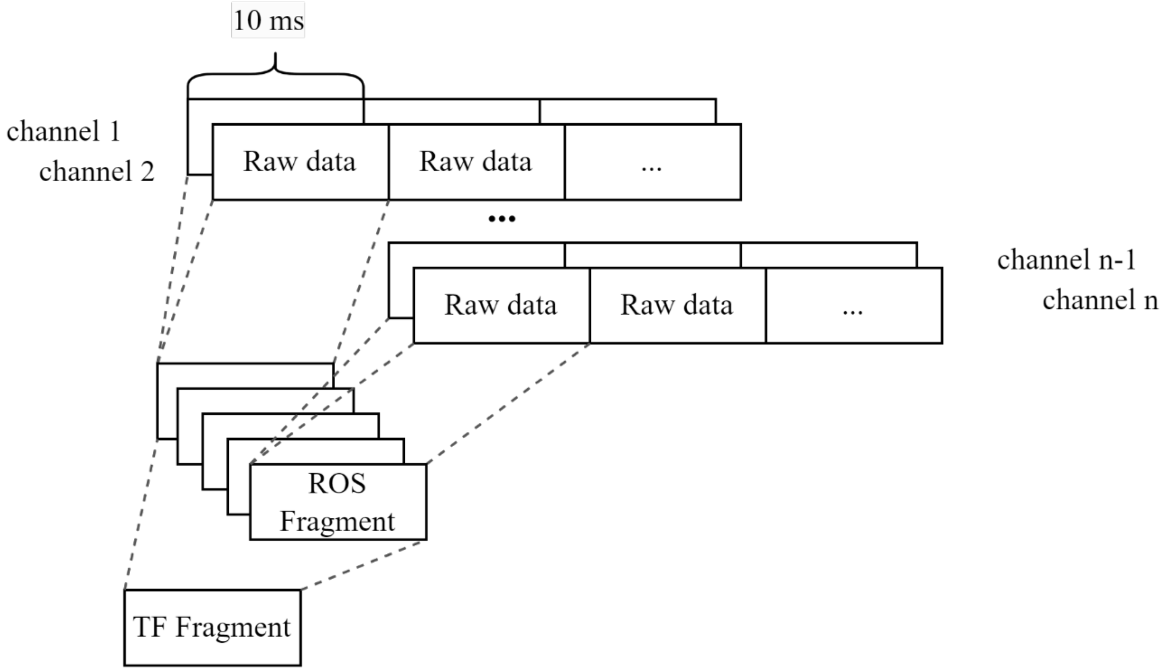
Data Flow Software



- Lightweight structure
- ROS + DA + DP + DS
- **Plug-in modules design**
- Integrate customized readout / processing modules

DAQ Data Assembly

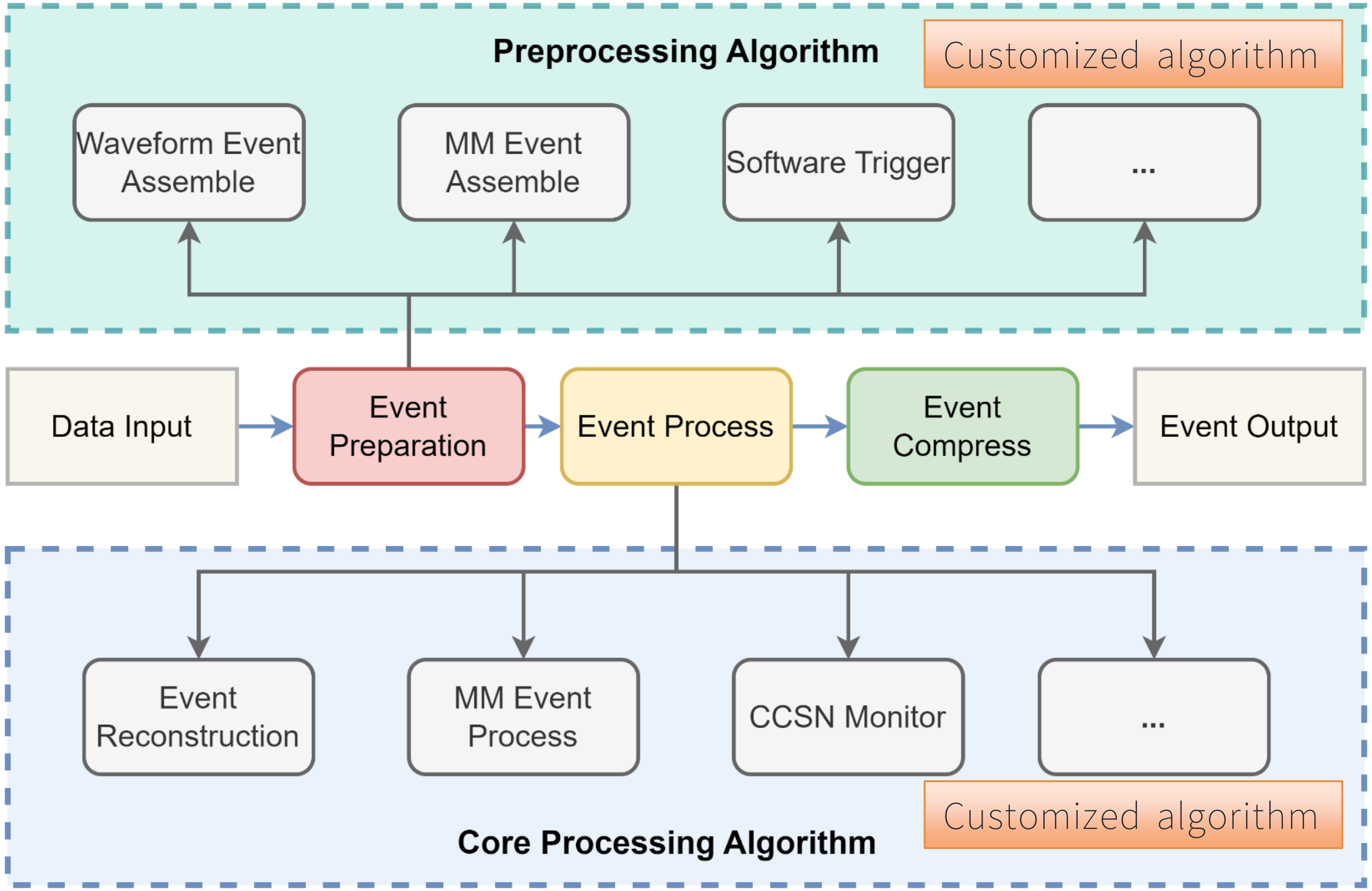




- Assembled by **trigger ID or timestamp**
  - **HW / SW trigger supported**
- **Uniform processing** triggered & trigger-less data
- **Multi-level assemble** – ROS + DA

Online Data Processing

Data processing is flexible, configurable



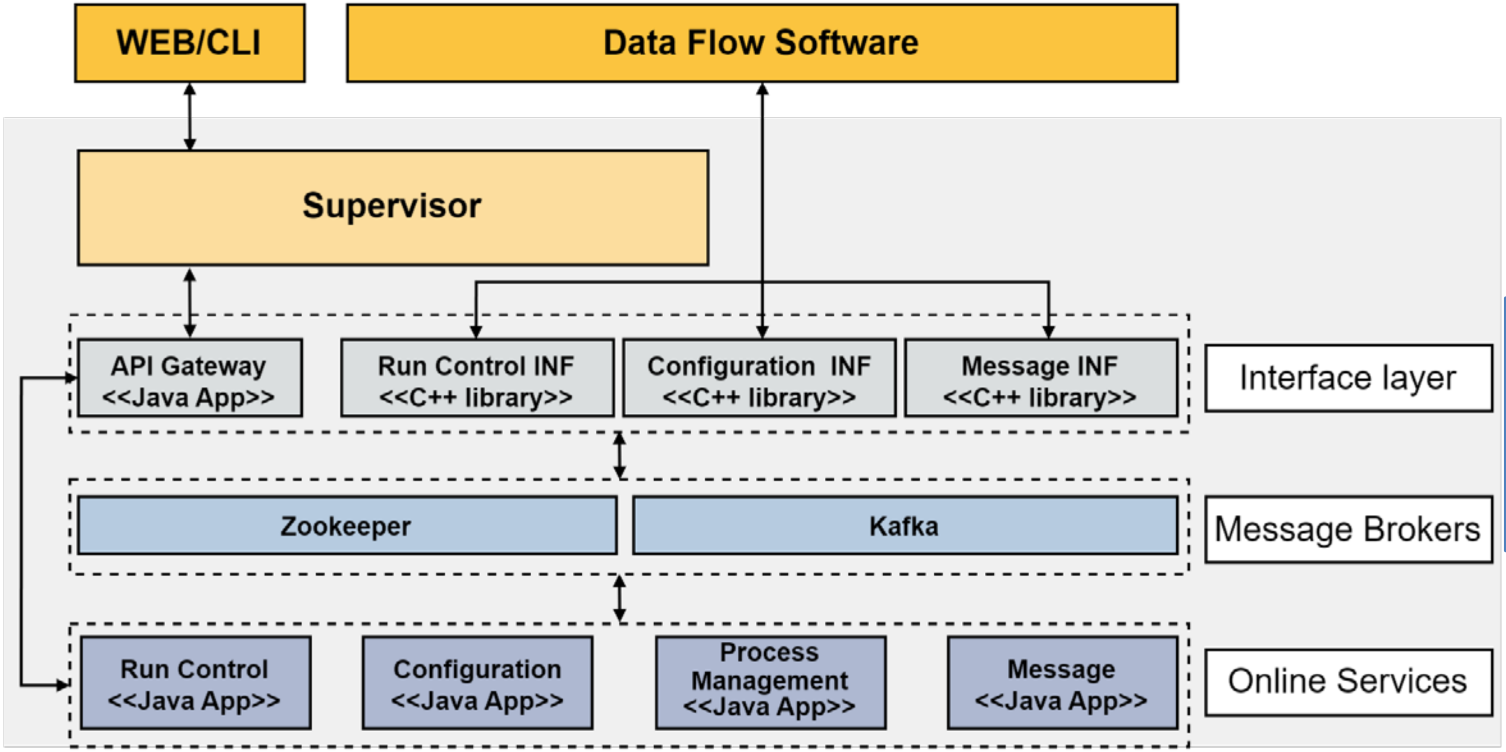
Multi-stage data processing:

- Data preparation / preprocessing
- Core event processing
  - **HLT algorithm** can be integrated

- **Parallel processing**
- **Interface provided**
- **Plugin algorithm** deployment

- **CCSN:** Core Collapse SuperNova
- **MM:** Multi-Messenger

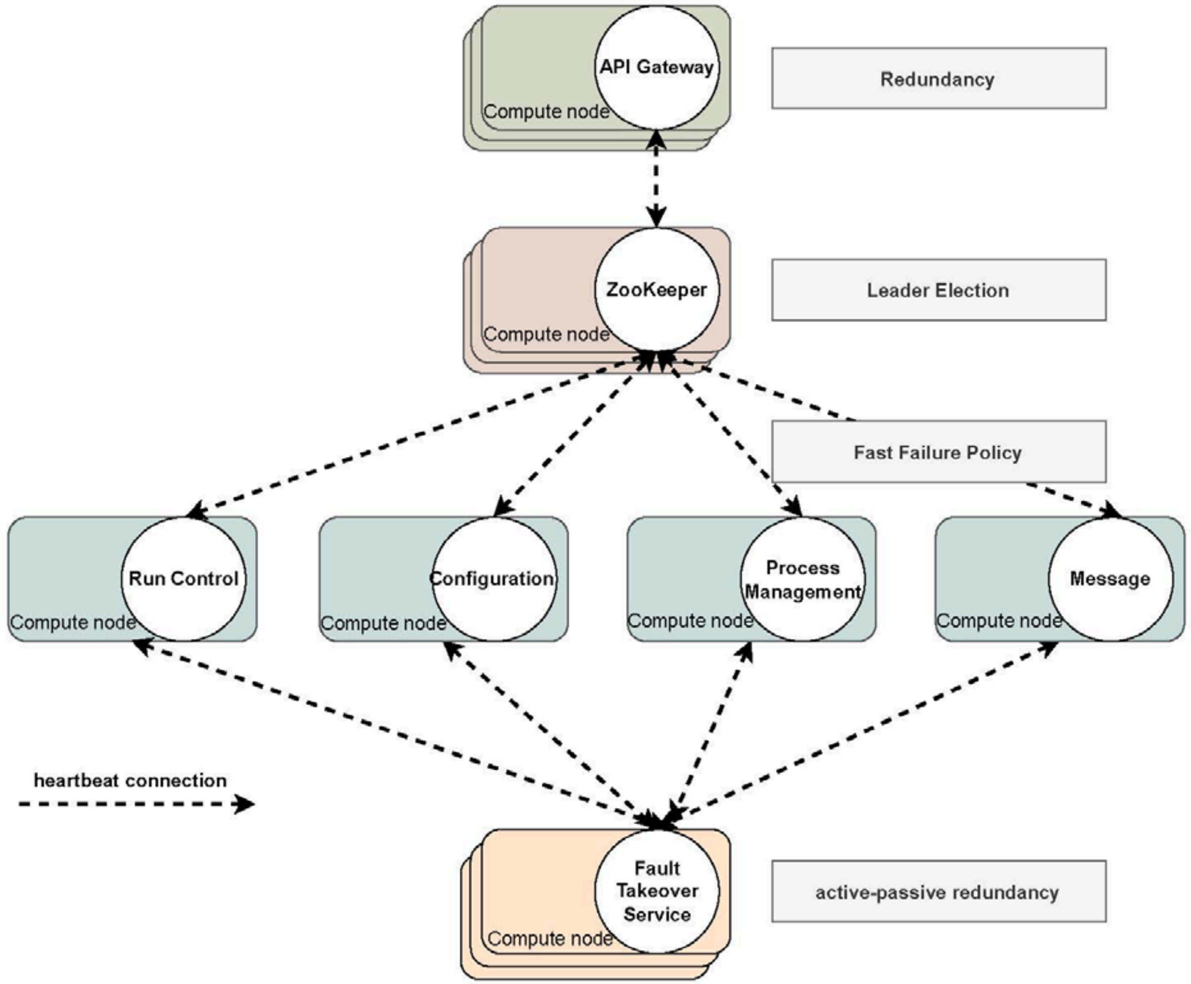
Online Software Overview



provide management, interfaces and services

- **Centralized messaging topology** → message brokers **decouple** online - data flow
- **Microservices architecture** → Independent & Loosely-Coupled Services
- **Layered component design** – interface, message, online services and supervisor
- **Kubernetes-Managed Containers** → Orchestrated Operations

Online HA Design & Implementation



Failure Detection

Heartbeat Detection

- Inter-Process: Based on ZooKeeper
- Inter-Node: Based on ICMP Protocol

Failover

- Redundancy + Master-Slave Election
- Takeover scheme based on Kubernetes
- Solutions for rapid restart of services

✓ Anomaly detection and recovery fully implemented

✓ Services recovered within seconds, ensuring overall reliability

Radar V3 – Current R&D

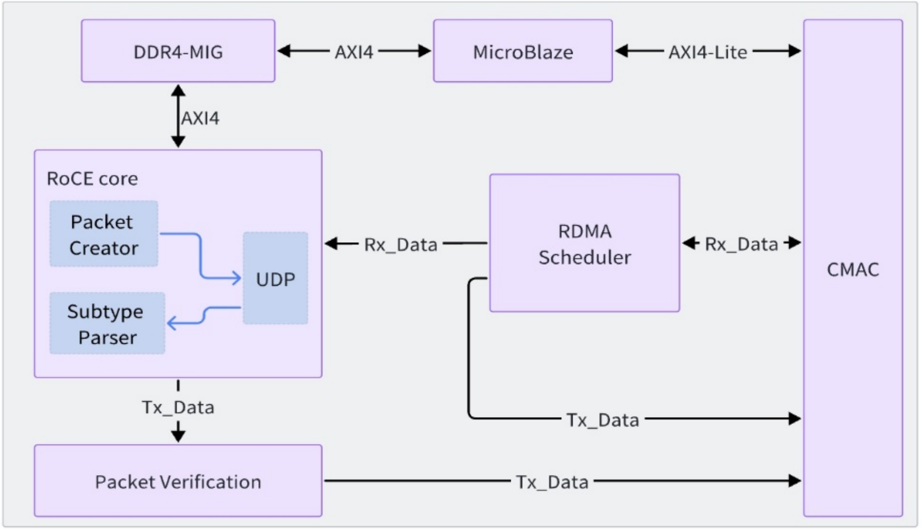
Readout protocol	RDMA research
Dataflow	HA design
Online processing	Heterogeneous computing platform
Online cache	Distributed memory cache pool

Implement RDMA on FPGAs

DAQ readout protocol

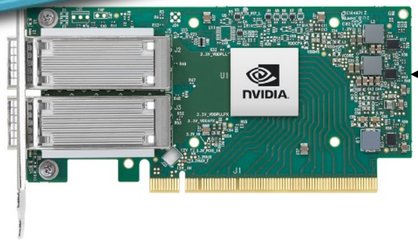
- **Alt. to TCP for possible software trigger upgrade**

- **Protocol Selection**
  - RoCE v2: low cost, high perf.
- **Firmware Design: Integrate RDMA Logic with UDP**
  - RoCE Core: UDP Parsing/Generation
  - RDMA Scheduler: RoCE packet Verification & ARP Handling
  - Packet Verification: Validates data packets
  - DDR4 MIG: Data cache management
- **Point-to-point test platform built**
- **Development status**
  - √ ICMP & ARP
  - √ RDMA send / write + connection management
  - RDMA recv / read in progress



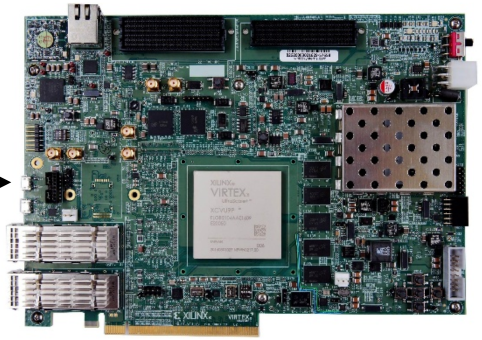
RDMA FPGA Firmware Design.

- Partial implementation achieved
- Further development required



100G link

QSFP28



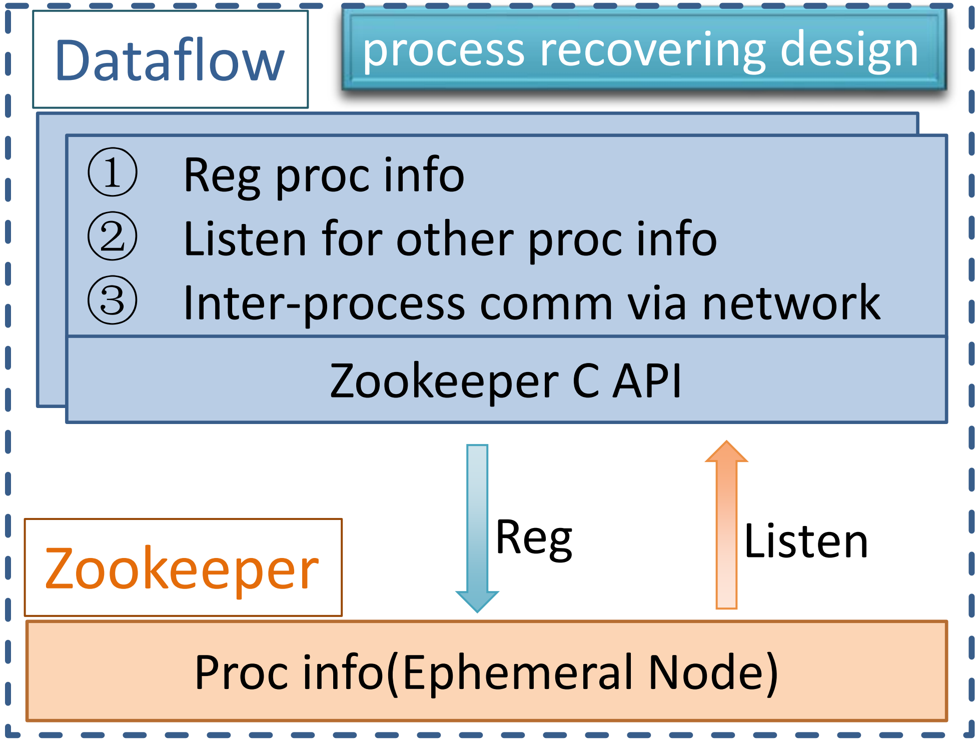
Nvidia Mellanox Connectx-5.

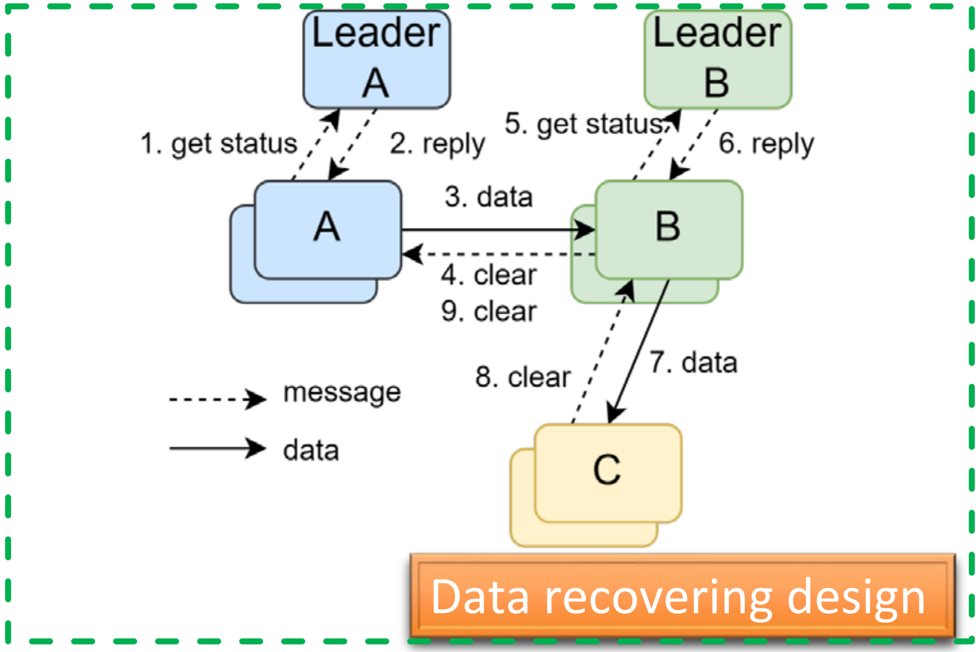
AMD FPGA VCU118 Evaluation Kit.

Dataflow HA Design

- Goal: Auto-recover + minimum data loss
- Develop process recovering scheme
  - Dynamic update
  - State machine path update
  - Test passed in dataflow
- Data recovery solutions
  - From backup node | From upstream node

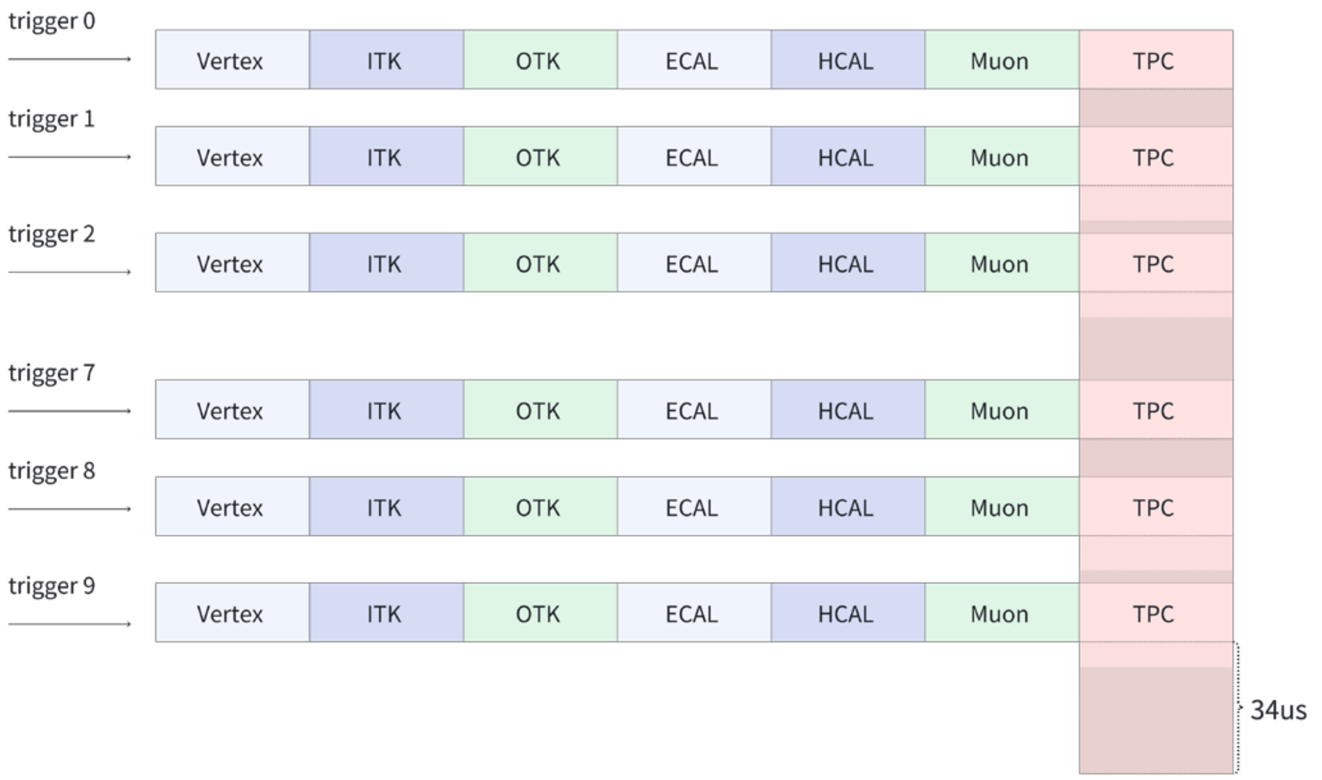
[ONLINE HA] ● Implemented in Radar V2
[DATAFLOW HA] ◎ Developing
➜ VISION: Radar v3 → Full HA DAQ framework





Processing - Data Assembly

- **L1 trigger ID based** event building
- **Batch pack** for parallel processing
  - ↑ GPU utilization
  - ↓ TPC data overlap



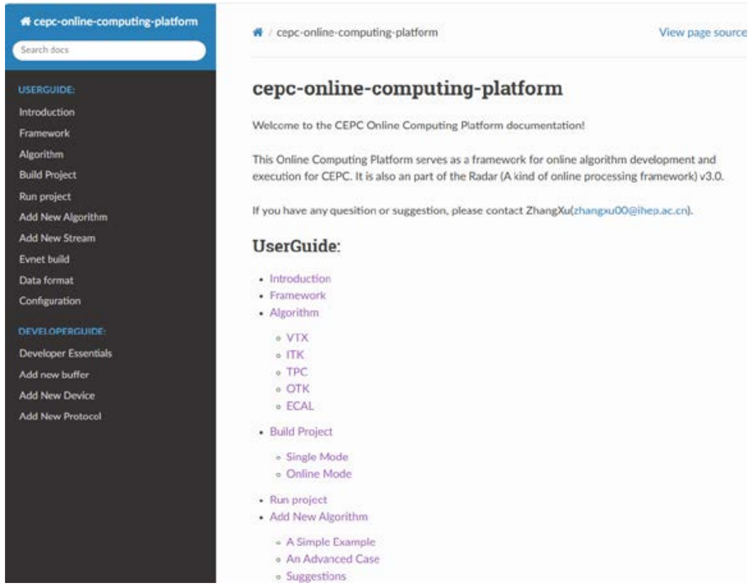
Processing - Heterogeneous Computing Platform

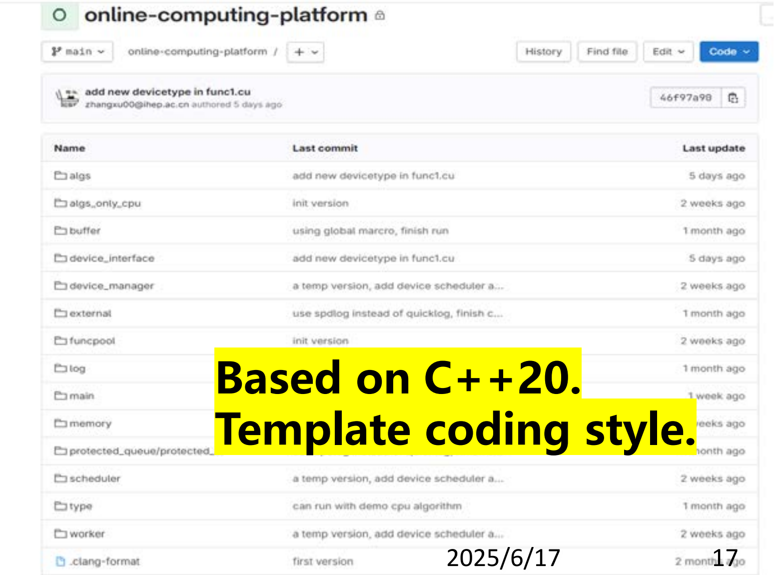
Support HLT or full-software trigger mode

- **Heterogeneous scheduler**
  - **Balanced Loading**: schedule across devices
  - **Data-aware Strategy:** co-locate dependent tasks
  - **Dev-Friendly**: unified memory management
- **Compatibility Matrix**

Type	Supported	To be supported
Buffer	Disk	Shared Memory • Memory Buffer
Protocol	TCP	RDMA
Compute	CPU • GPU	FPGA
Memory	CPU • GPU	FPGA

Status: V1 Released • Internal Validated





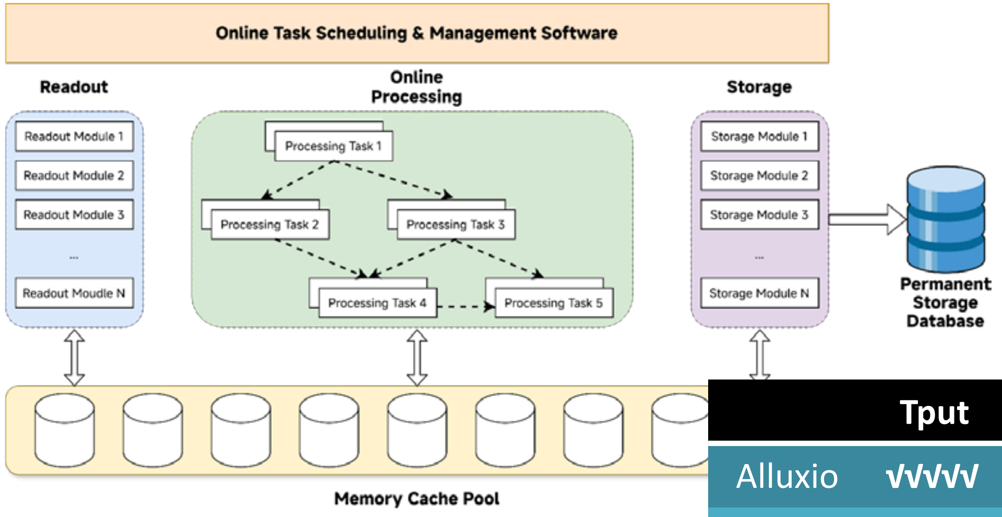
Based on C++20.
Template coding style.

MemIO – Distributed Memory Cache Pool

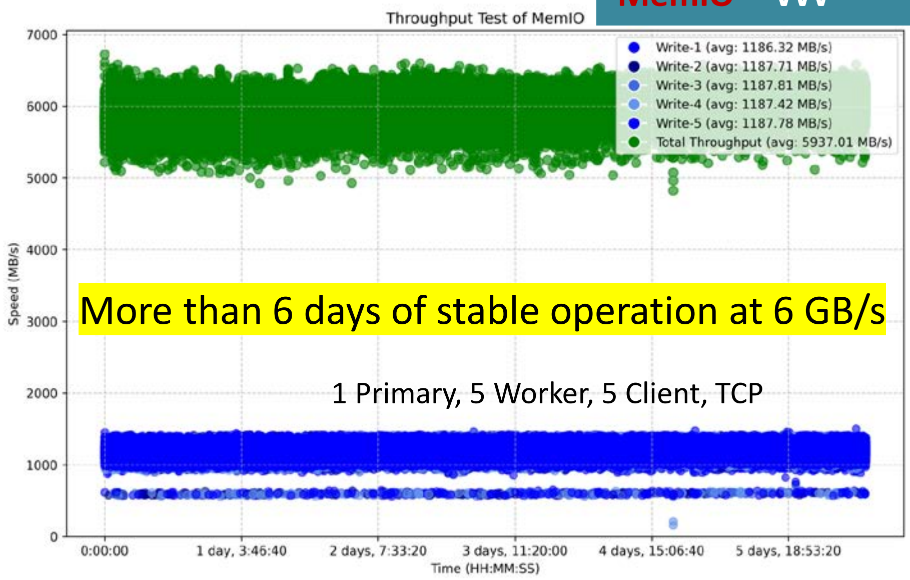
- ***Why***
  - **Decoupling**
  - → Dynamic Readout + Heterogeneous Compute
- ***Memory Aggregation***
  - **Speed >> Disk**
  - Online compute focus: **Throughput > Capacity**
- ***Core advantage***
  - **High Tput| High Eff. | High Rel.**
- ***Choice***
  - Trade-off: Perf. + Avail. **VS** Cost
  - Strategy: OR/AND Buffers

Research Progress: V1 ✓ | LHAASO Test ⌛

▸ Next: RDMA ⇩ CPU | MemFS ⚡ | Optimize



	Tput	Eff.	Rel.
Alluxio	√√√√√	√√√	×
MinIO	√√√√√	√	√
MemIO	√√√	√√√	√



Detector Control System: Requirement

Maximize experimental efficiency & data quality via stable, reliable, secure operation

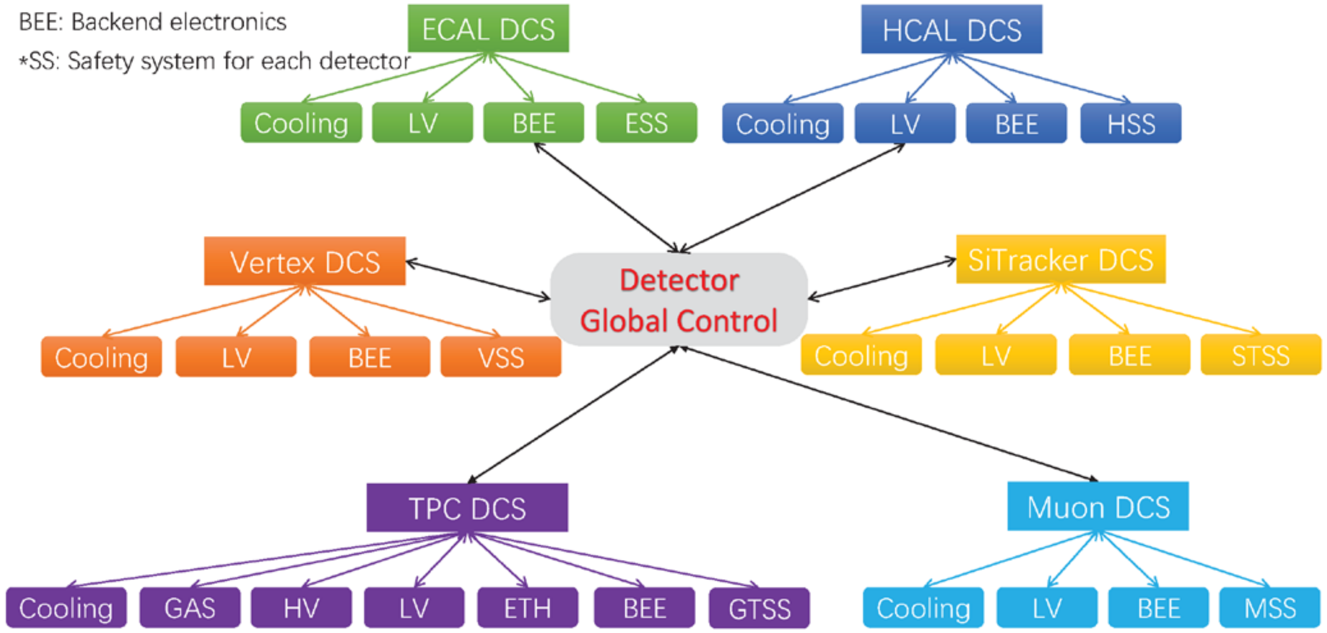
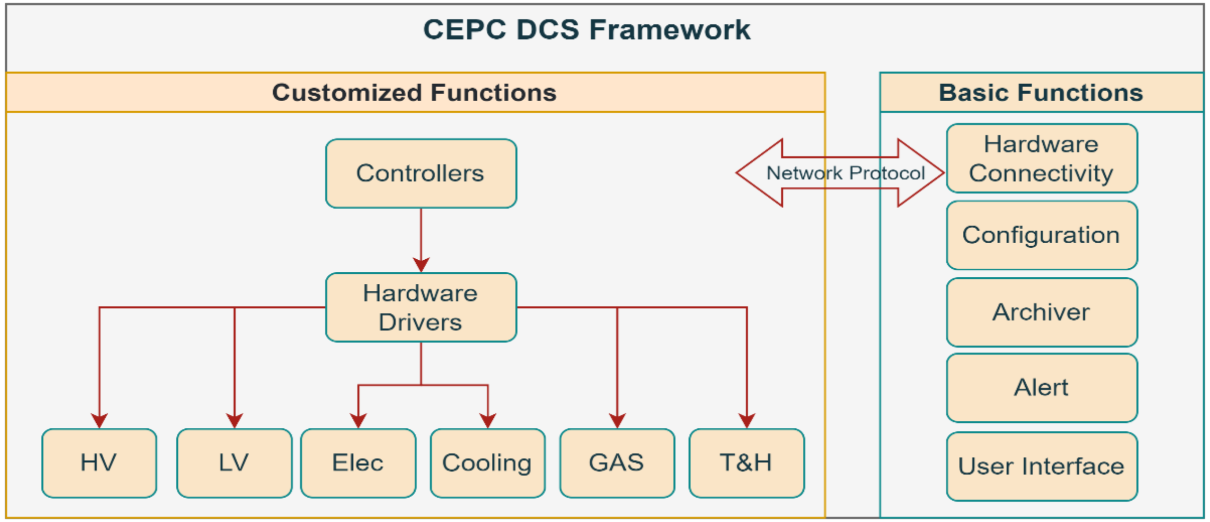
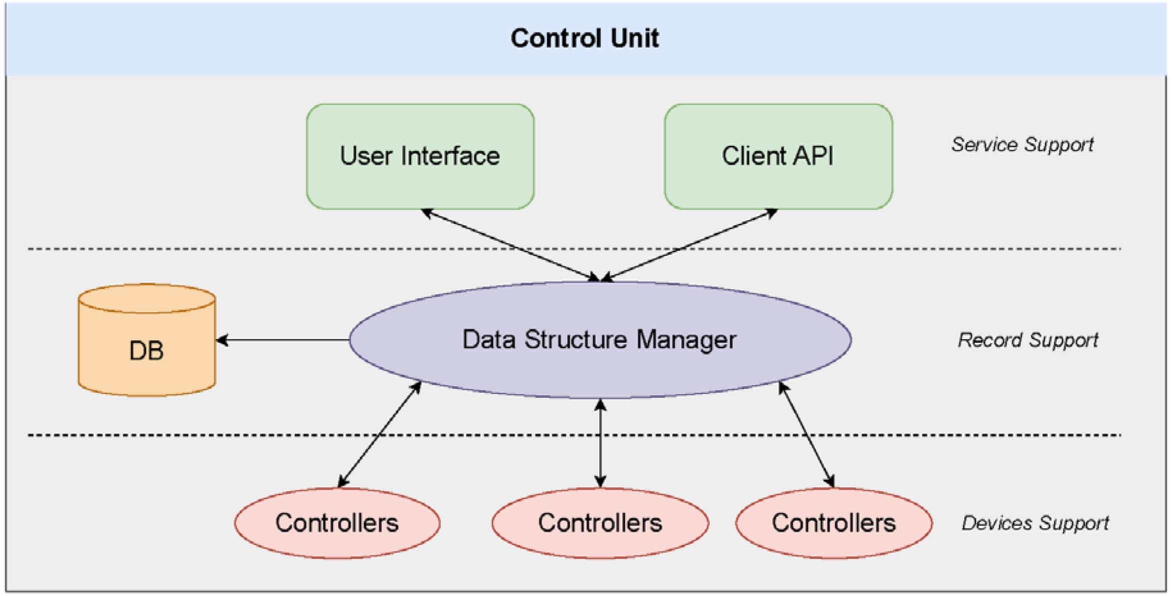


Table 12.16: Summary of detector control and monitoring quantities

Equipment	Channel Number	Control and Monitoring Items	Total Count
AC-DC Crate	55	Switch, voltage, current, etc., about 6 items	330
AD-DC Channel	550	Switch, voltage, current, etc., about 6 items	3300
FEE Power Crate	390	Switch, voltage, current, etc., about 6 items	2340
FEE Power Channel	17696	Switch, voltage, current, etc., about 6 items	106,176
FEE Board	27984	Voltage, current, temperature, etc., about 10 items	279,840
BEE Crate	110	Switch, fan, status, etc., about 10 items	1100
BEE Board	1018	Switch, voltage, current, etc., about 6 items	6,108
Trigger Crate	20	Switch, fan, status, etc., about 10 items	200
Trigger Board	150	Switch, fan, status, etc., about 10 items	1500
HV Crate	91	Switch, fan, status, etc., about 10 items	910
HV Channel	18694	Voltage, current, status, etc., about 6 items	112,164
Gas	1	Flow control, humidity, etc., about 30 items	30
Environmental	3568	Temperature, humidity	5448
Total			~520,000

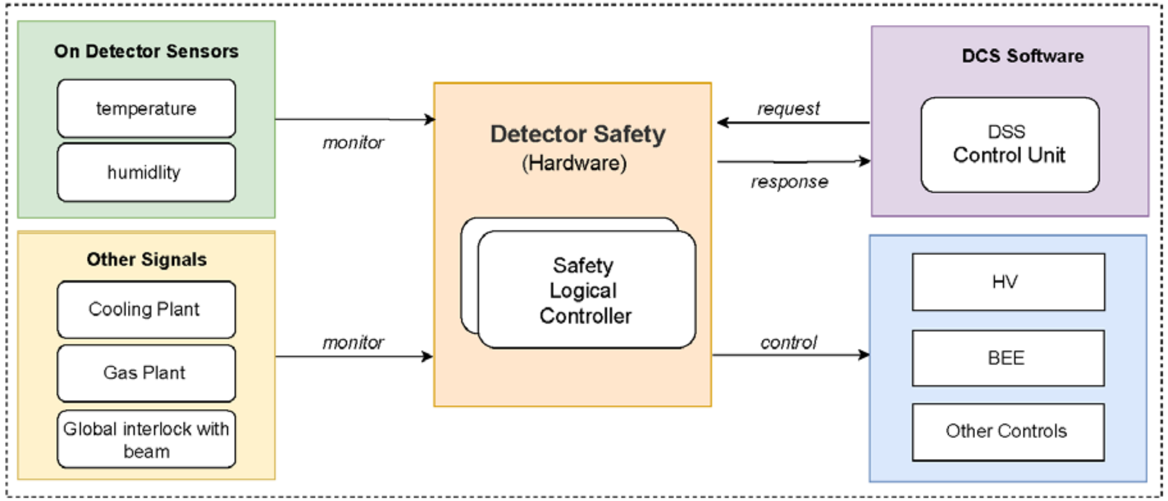
Detector Control System: Design





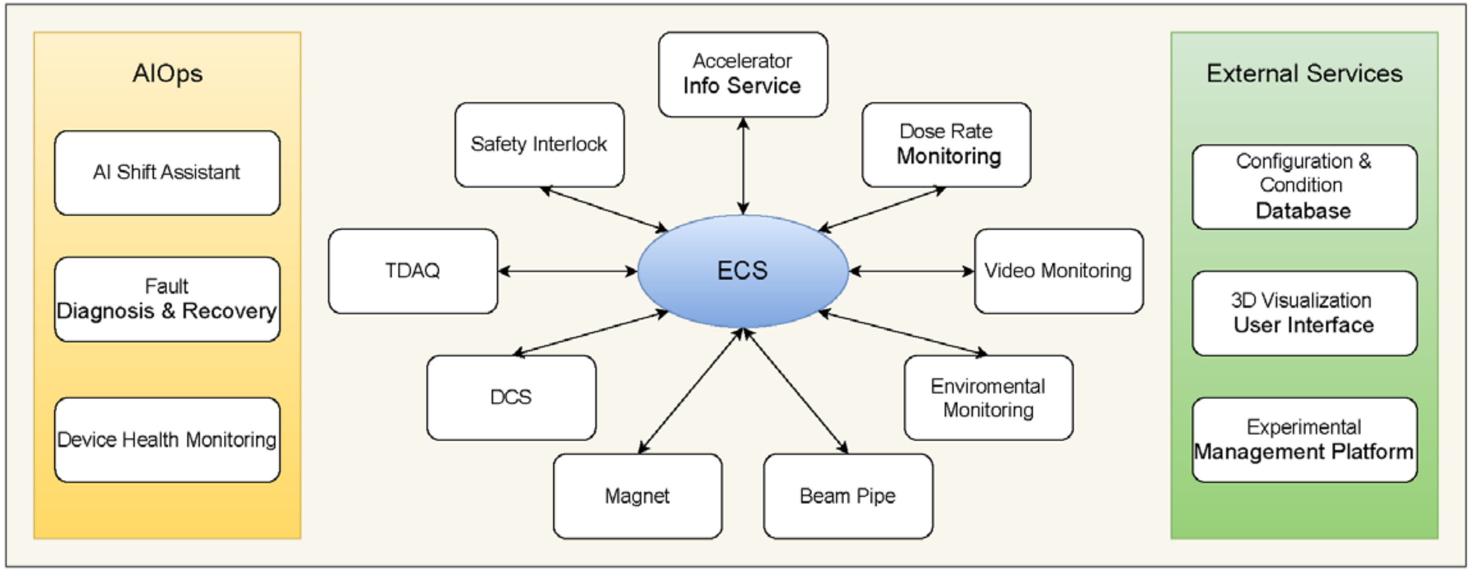
Key function modules

- Automated Process Control
- Real-time Monitoring & Alarms
- Detector safety interlock
- Remote Access



Experiment Control System

Deliver secure, efficient experimental control in demanding experimental environments



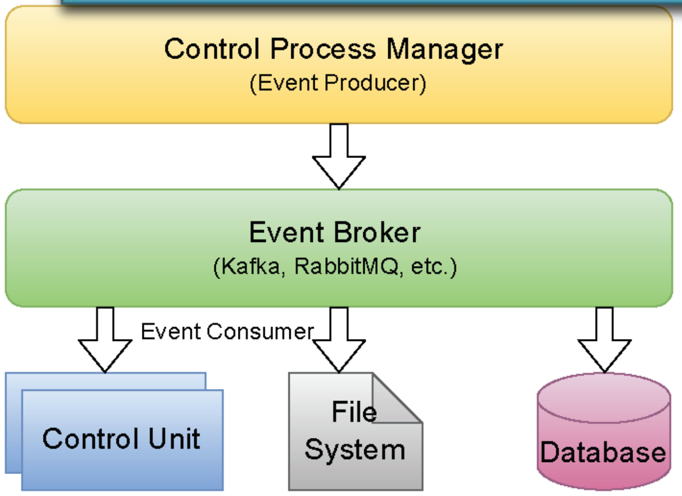
Scope	**Centralized**, higher-level than DCS
Integration	DCS + DAQ + other sub systems
Safety	Fault detection + Auto-recovery
Automation	AI analytics + Resource control
User Experience	Intuitive GUI + Unified access

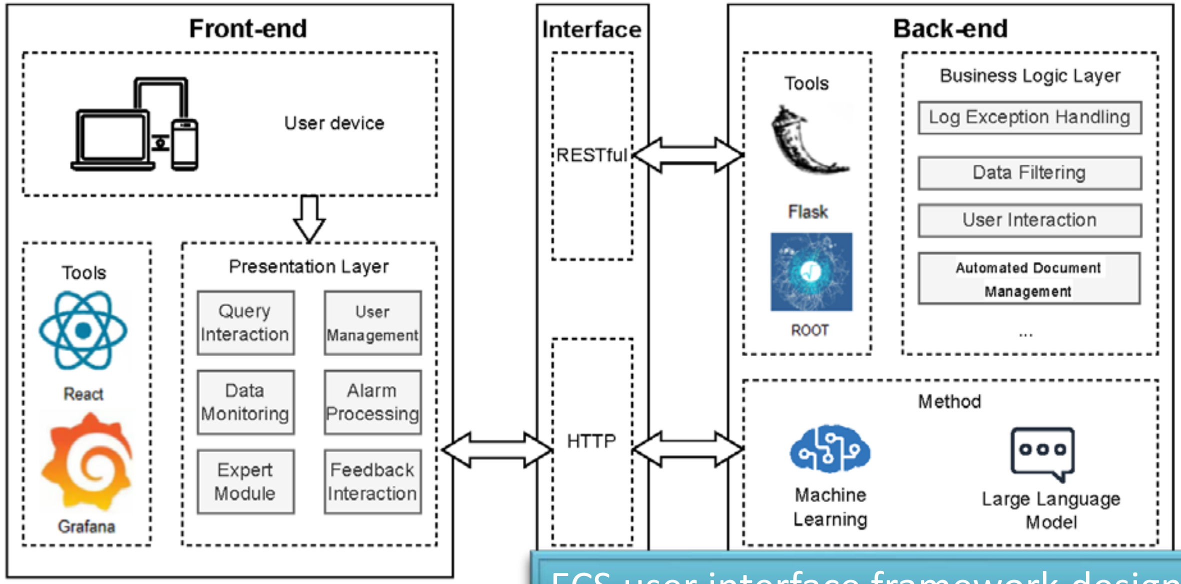
Experiment Control System: Design

Modular, standardized, AI-driven, resilient architecture

- Central Control Module
- Data Acquisition Module
- User Interface Module
- **AI Integration Layer**
- Safety Mechanisms

Central control software architecture

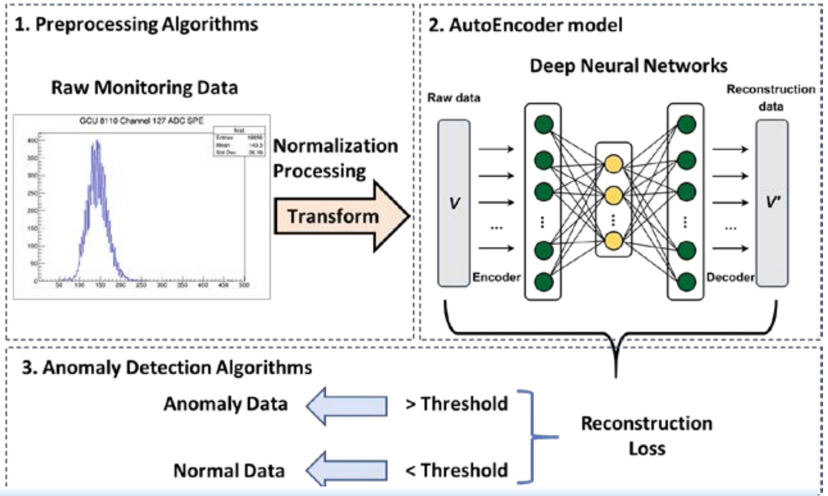




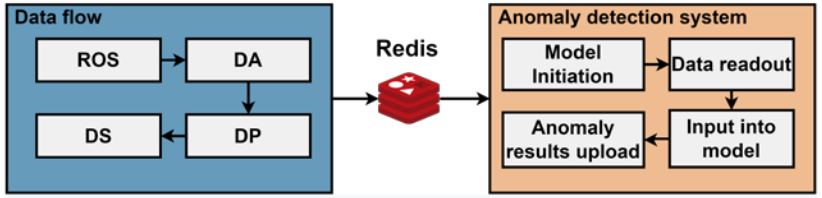
ECS user interface framework design

ML-based Anomaly Detection

More real-time and automatic monitoring, ensure the experiment efficiency



Architecture of Anomaly Detection Methods



Deployment Strategy

- **Realtime anomaly detection system**
  - **Unsupervised** Model: Autoencoder
  - **Supervised** Model: Neural Network
  - Model training & optimization
  - Integrated & tested with Radar

Model	Recall	Prec.
Autoencoder	~97%	~88%
Neural Network	~99%	~99%

- **Next:** Multi-model for scenarios

* Test done at JUNO, with SPMT ADC SPE

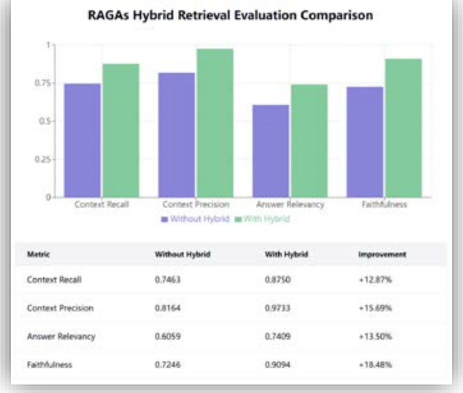
Provided a solution for real-time anomaly detection, has been validated in DAQ workflow

AI-based Intelligent Assistant

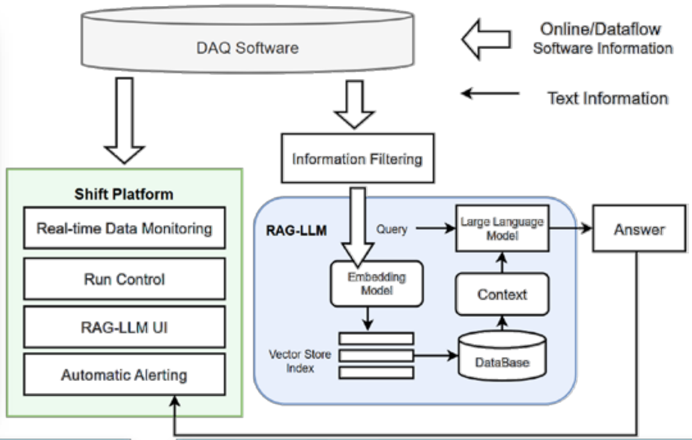
More efficient and intelligent information integration and interaction

- ⟐ **INFRASTRUCTURE**
  - ✓ Private KB & LLM Deployment
  - ✓ Stable Query API
- ⟐ **RAG INTELLIGENCE LAYER**
  - ✓ RAG-Powered Assistant
    - ⟐ Efficient Archive/Retrieval
  - ✓ RAG Evaluation Framework
    - ⟐ RAG chain tuning
- ⟐ **Application**
  - ✓ Auto-Anomaly Diagnosis Engine
- ⏭ **Next**
  - ⟐ Multi-modal extension
  - ⟐ Anomaly Detection Integration

Interaction interface display



Metric	Without Hybrid	With Hybrid	Improvement
Context Recall	0.7463	0.8750	+12.87%
Context Precision	0.8164	0.9733	+15.69%
Answer Relevancy	0.6059	0.7409	+13.50%
Faithfulness	0.7246	0.9094	+18.48%



Link optimization and evaluation

Anomaly Diagnosis Scheme

Summary

- Finished baseline technical design for DAQ & DCS & ECS
  - A scalable and flexible design
- Work in different directions is progressing
- Welcome to join us

Thanks!