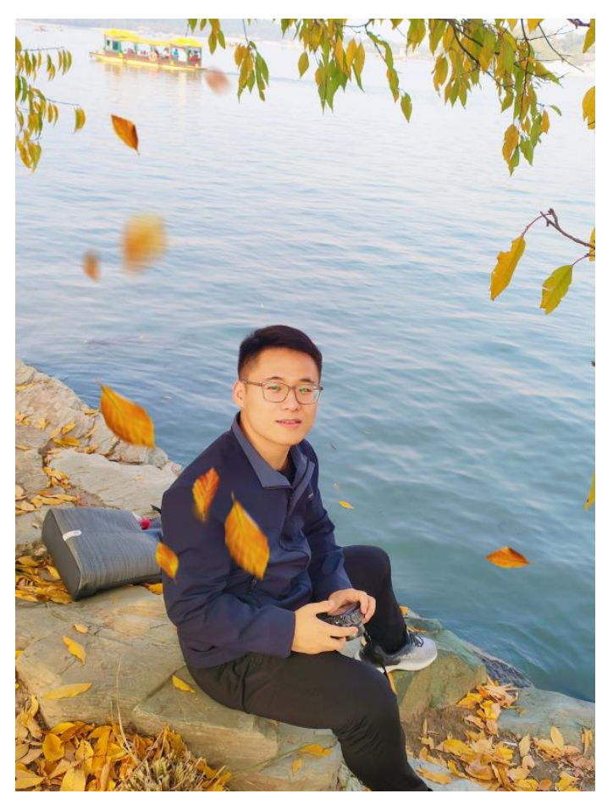
中国科学院高能物理研究所
Institute of High Energy Physics, Chinese Academy of Sciences

24th IEEE Real Time Conference
2024/04/22 – 2024/04/26 – ICISE, Quy Nhon, Vietnam

A Plugin-Based Software Framework for Data Acquisition and Processing

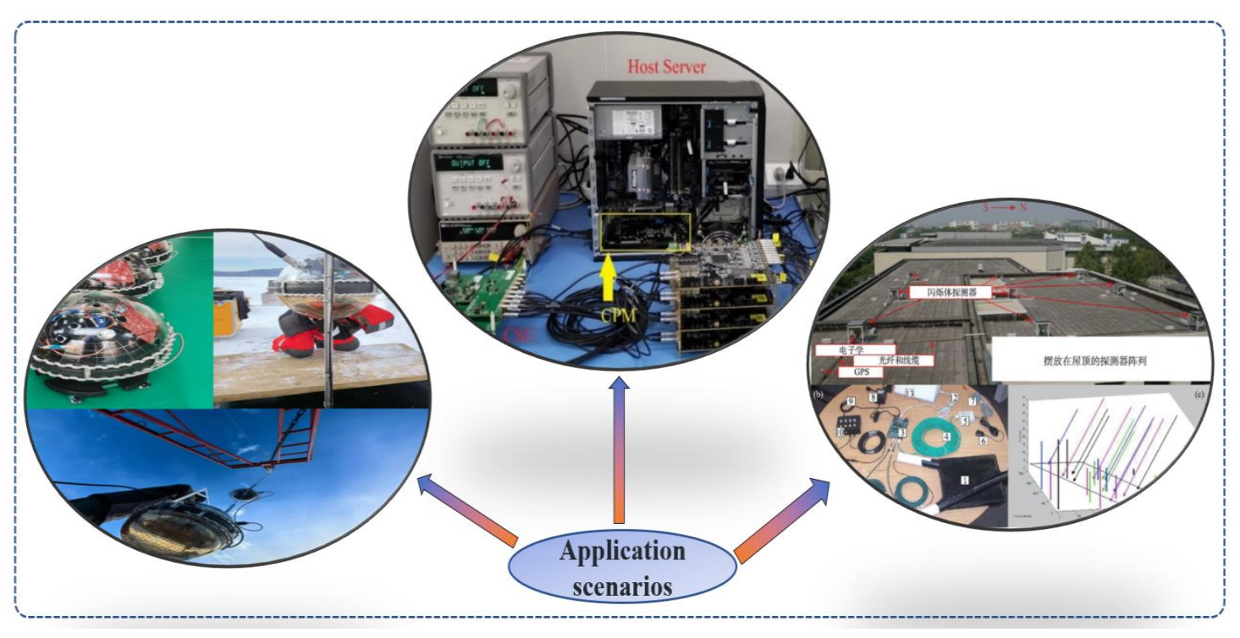
Shaoshuai Fan(1,2,3), Minhao Gu(1,2,3), Hangchang Zhang(1,2,3)

(1) Institute of High Energy Physics, Beijing 100049, People's Republic of China
(2) University of Chinese Academy of Sciences, Beijing 100049, People's Republic of China
(3) State Key Laboratory of Particle Detection and Electronics, Beijing 100049, People's Republic of China



INTRODUCTION

Compared to the data acquisition system(DAQ) of large-scale high-energy physics experiments, small-scale experiments, such as preliminary research experiments for large-scale experiments, educational instruction, engineering development, and medical applications, have the following characteristics: They have smaller data volumes and face a greater variety of front-end electronics readout and data processing needs. To meet the requirements of the above scenarios, we have proposed a DAQ framework—ExpDAQ, which offers flexibility in data flow and reusability of processing algorithms.

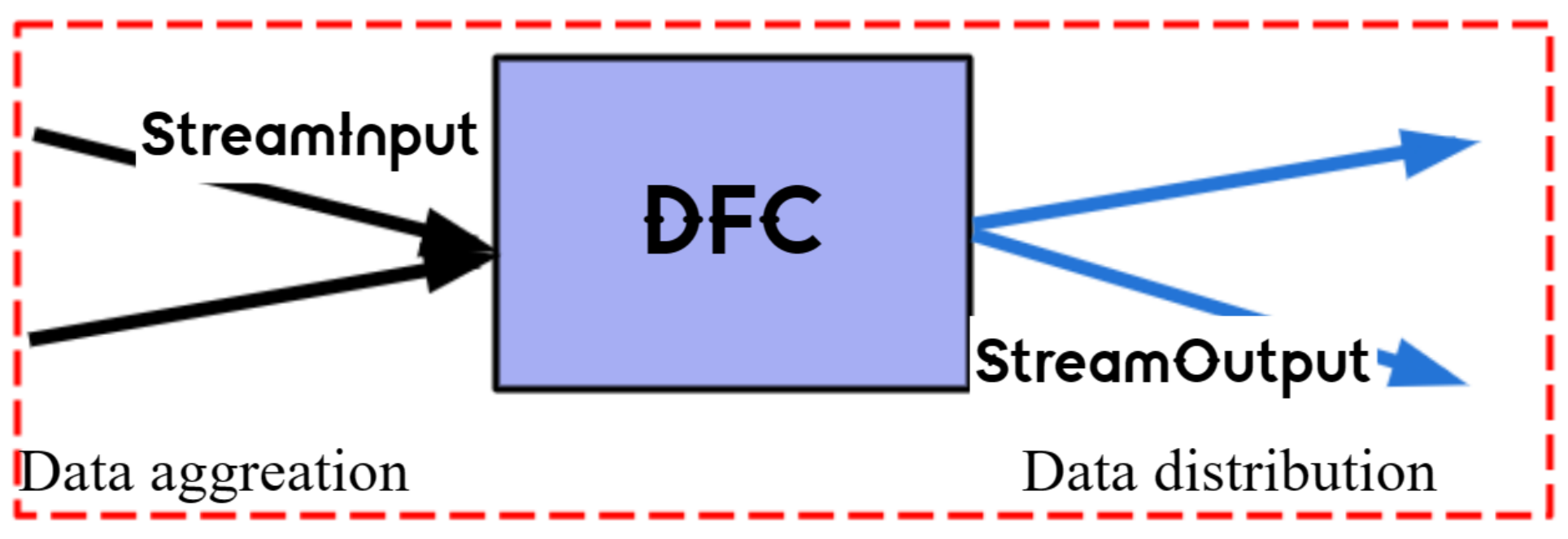


FEATURES

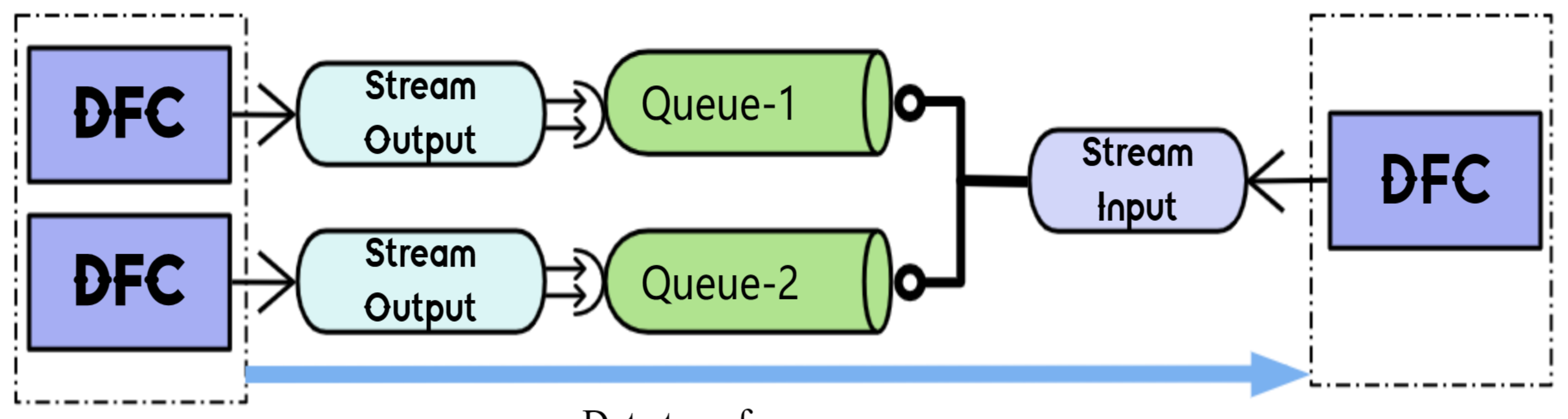
- In the ExpDAQ framework, the data flow process is scalable.
  - › A C++ plugin manager has been designed to manage different plugins, allowing for the addition of modules and system scalability.
  - › The data flow can dynamically change based on the configuration file, enabling dynamic variations in the processing flow.
  - › Define a standard data header format to label data fragments, enabling data aggregation and routing of fragments to specific processing modules.
- In the ExpDAQ framework, traditional state machines have been replaced. Plugins now manage their states independently by receiving commands.

METHOD

- Data Flow Component (DFC): A fundamental processing unit capable of being utilized to form various data streams.
- Composition of DFC
  - **StreamInput**
  - **StreamOutput**
  - **Data Process**



- The basic processing unit DFC's I/O
- StreamOutput: Distribute data into the next DFC based on different strategies
- StreamInput: Aggregate data from different DFCs based on different strategies

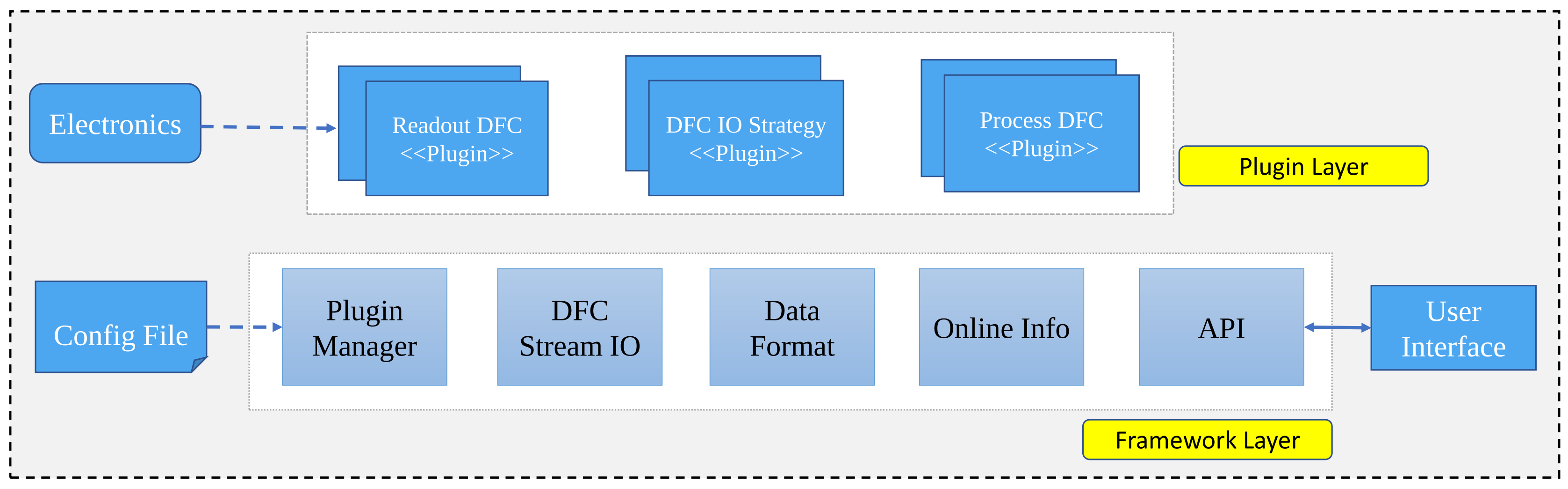


Data transfer

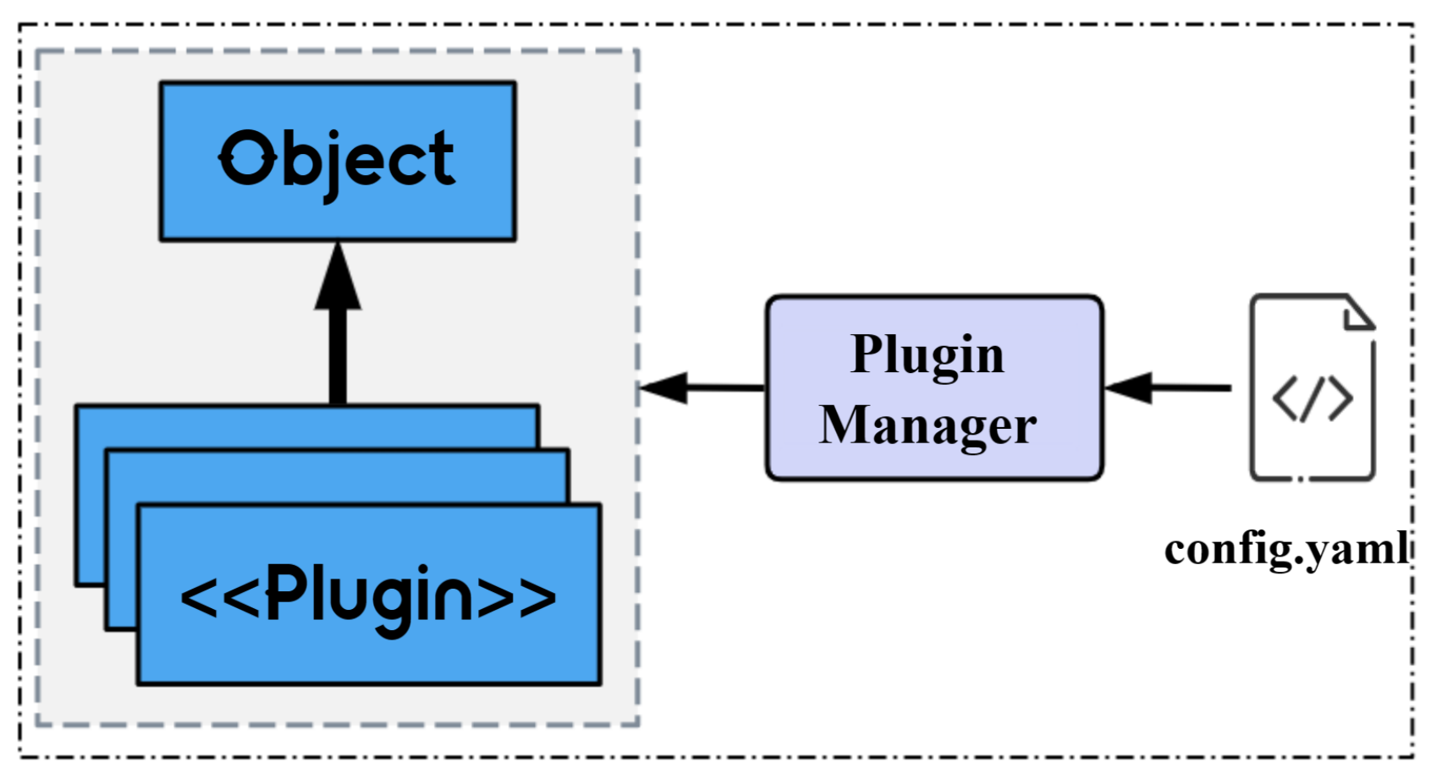
- Through the functions, data can be freely transferred between different DFCs.

DESIGN & IMPLEMENTATION

- **Architecture of ExpDAQ framework**



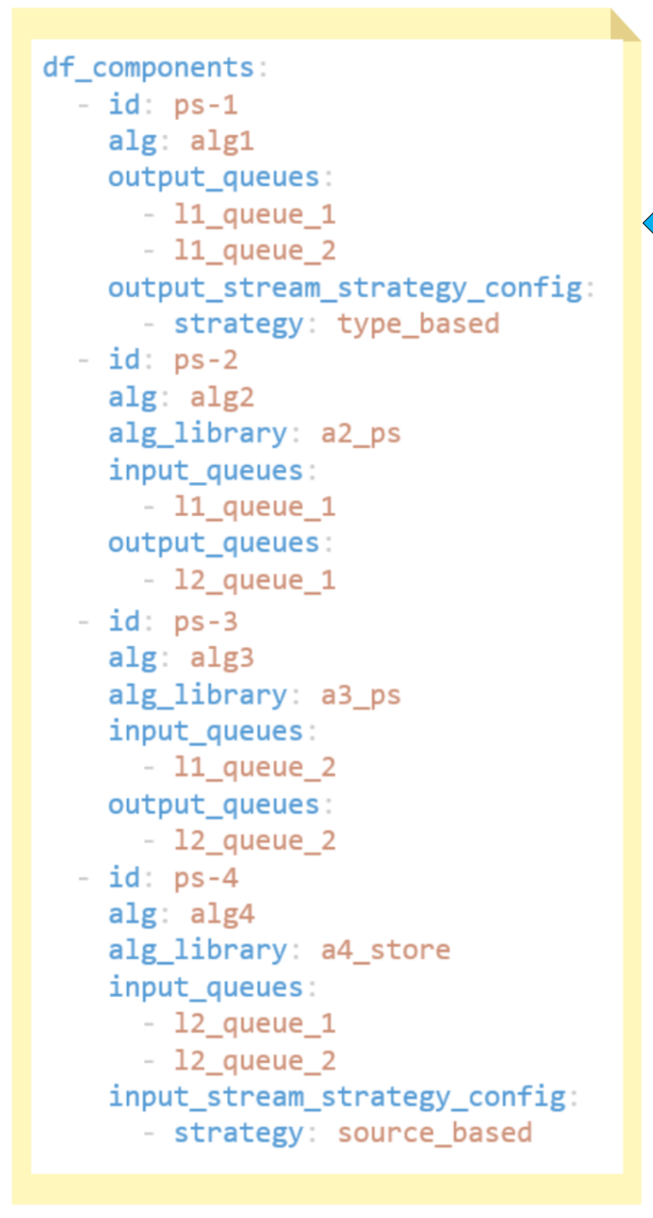
- The ExpDAQ comprises two layers: the plugin layer and the framework layer.
- Different functionalities of DFC call Stream IO and strategies to achieve the transfer of data packets between DFCs.
- The plugin layer includes two types of DFC with different functionalities:
  - Readout: Electronic configuration, readout, and data packetization, serving as the data source
  - Process: The fundamental unit for data processing
- Online Info & API：Monitor info publish & Control
- **Plugin Manager**



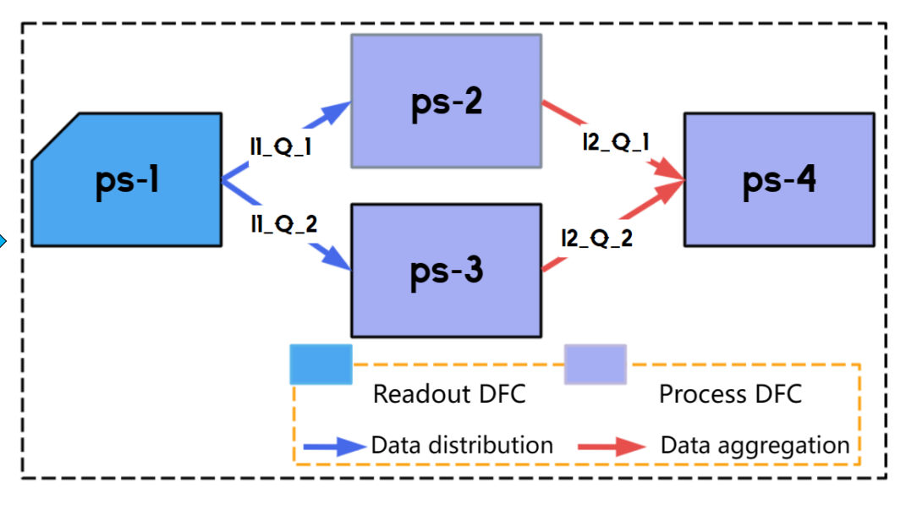
- Inherited from a same base class "Object"
- Object class can inject the required dependencies based on configuration information
- Dynamic loading and dependency injection of C++ base plugins.

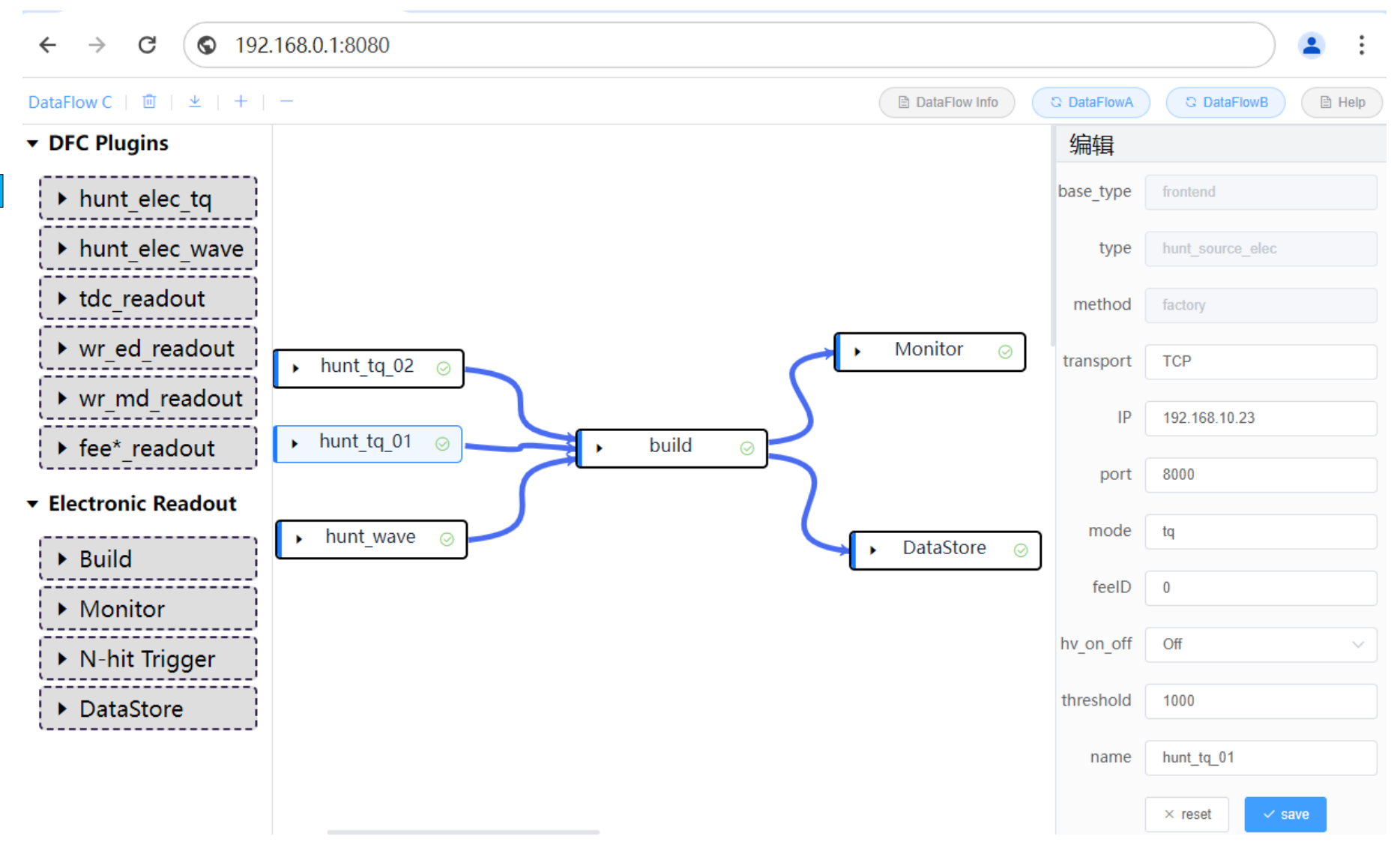
EXAMPLE

- The ExpDAQ framework implements the following functions
- Dynamically load processing plugins based on configuration file
- Stream I/O distributes or aggregates data based on strategy plugins



```
df_components:
  - id: ps-1
    alg: alg1
    output_queues:
      - l1_queue_1
      - l1_queue_2
    output_stream_strategy_config:
      - strategy: type_based
  - id: ps-2
    alg: alg2
    alg_library: a2_ps
    input_queues:
      - l1_queue_1
    output_queues:
      - l2_queue_1
  - id: ps-3
    alg: alg3
    alg_library: a3_ps
    input_queues:
      - l1_queue_2
    output_queues:
      - l2_queue_2
  - id: ps-4
    alg: alg4
    alg_library: a4_store
    input_queues:
      - l2_queue_1
      - l2_queue_2
    input_stream_strategy_config:
      - strategy: source_based
```





编辑

base_type	frontend
type	hunt_source_elec
method	factory
transport	TCP
IP	192.168.10.23
port	8000
mode	tq
feeID	0
hv_on_off	Off
threshold	1000
name	hunt_tq_01

In order to facilitate the design of data flows, we have developed an online flow design tool to assist in generating configuration files!

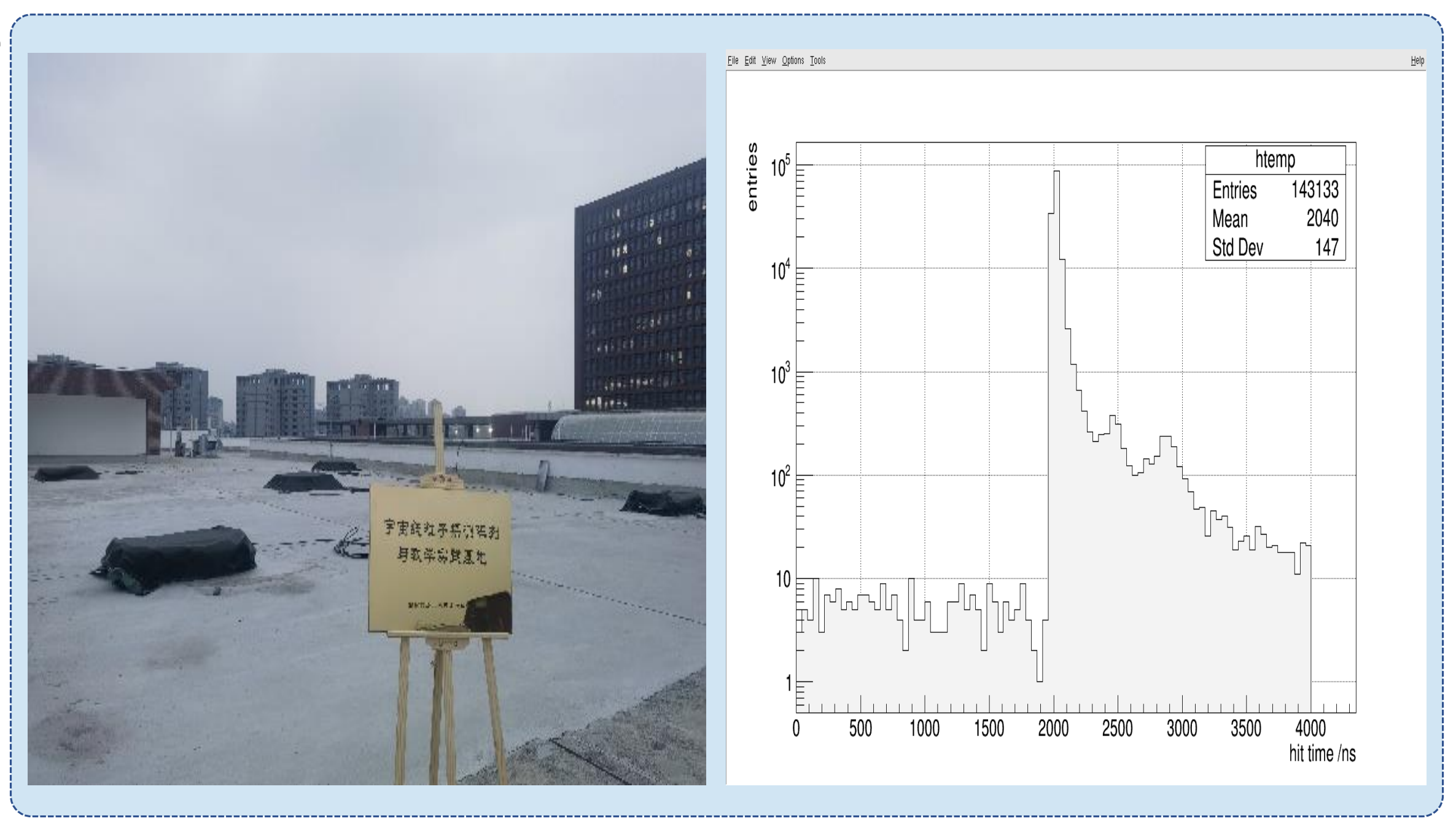
SUMMARY

The ExpDAQ framework allows processing units to freely combine processing flows through configuration files, providing a more convenient solution for data acquisition and online data processing in high-energy physics experiments.

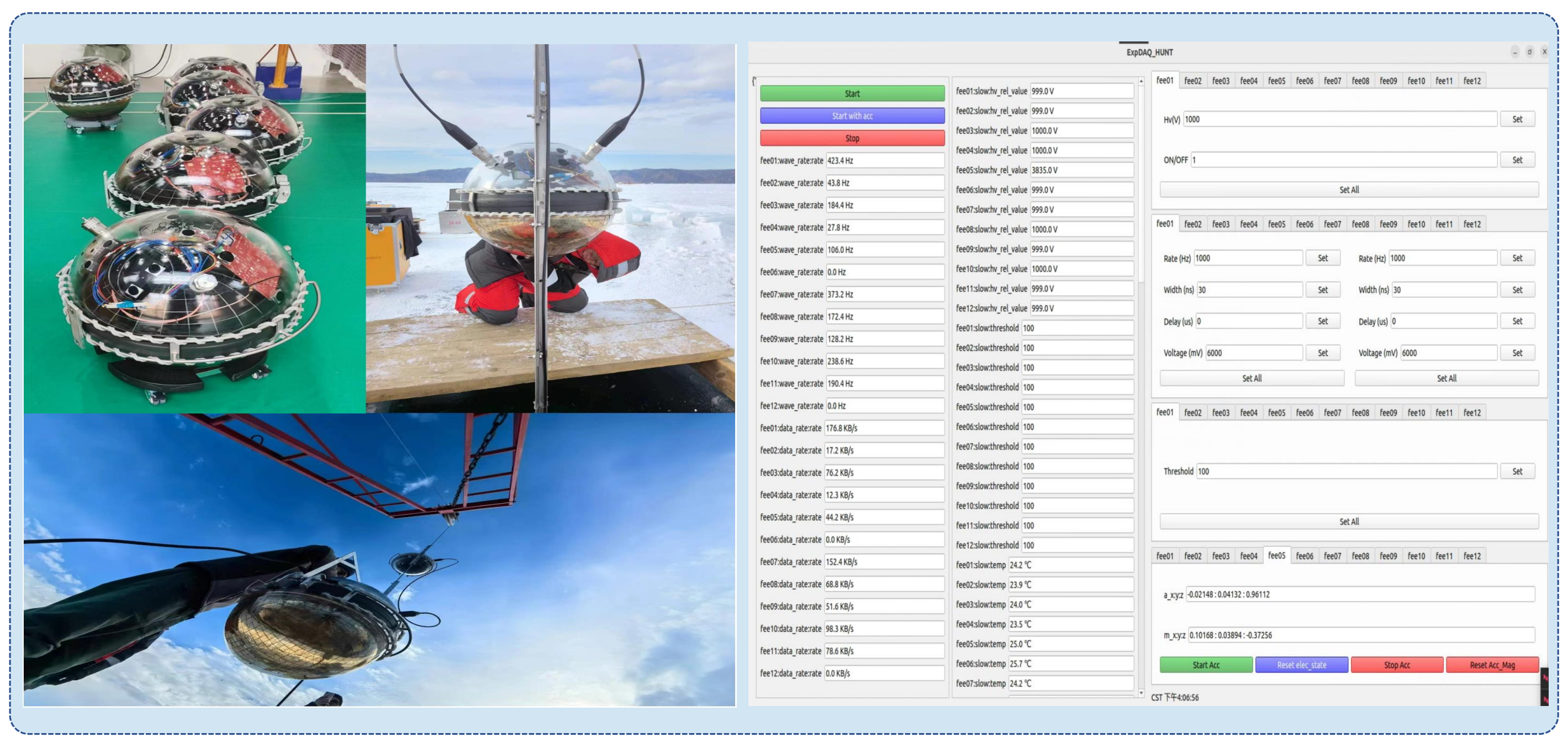
APPLICATION

The framework has been applied in multiple experiments.

- **Huge Underwater high-energy Neutrino Telescope prototype experiments**
- **LHAASO-ED detector testing**
- **Campus Cosmic-ray Observation Collaboration**
- **Time-to-Digital Converter Experiment**



The cosmic-ray particle detection array at Southwest Jiaotong University.



The HUNT prototype experiment located at Lake Baikal and its DAQ UI interface.