Streaming Readout Workshop SRO XII
University of Tokyo, 12/02 – 12/04, 2024

RCNP

SPADI Alliance

SPADI Alliance and ARTEMIS for SRO processing

Shinsuke OTA

Research Center for Nuclear Physics

Contents

- SPADI Alliance
  - An alliance for the development and standardization of streaming data acquisition system
- Artemis
  - An analysis framework for the online-offline analysis with minimum-labor coding

ARTEMIS

A RooT Extension with Modular processor for Instant Switching

https://github.com/artemis-dev/artemis/tree/develop

1 experiment per 1-2 weeks

Slightly or Largely DIFFERENT detector setup

History

- ~2002 Using PAW based framework (widely used in old RIKEN Acc. Facility)
- ~2008 Start discussion on the new data format of RI Beam Facility in RIKEN
- ~2013 First implementation of artemis and upgrading accordingly
- 2013 – Used for RIBF, HIMAC experiments and recently RCNP, J-PARC
- 2022 Adaptation to the NestDAQ data format

Concept

How can we reduce the debug process and share the same software for many of experiments?

Analysis procedure

Quick and easy modification of mapping and procedure

User interface

Easy operation without C++, but

Data adaptation

Any types of event sources

Output

Create histogram without compilation

Feature

Analysis procedure

No code flow building

User interface

CUI base Shorthand commands

Data adaptation

Prepare only input adapter
(and well-structured data format)

Output

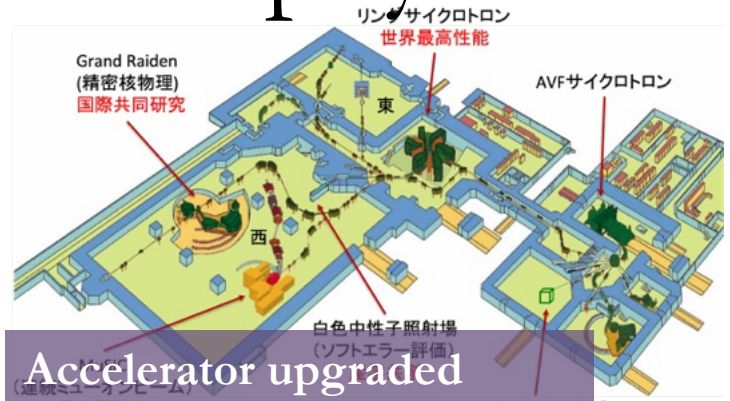
Hack the dictionary of ROOT

Why?
Short lived setup in low-energy nuclear physics

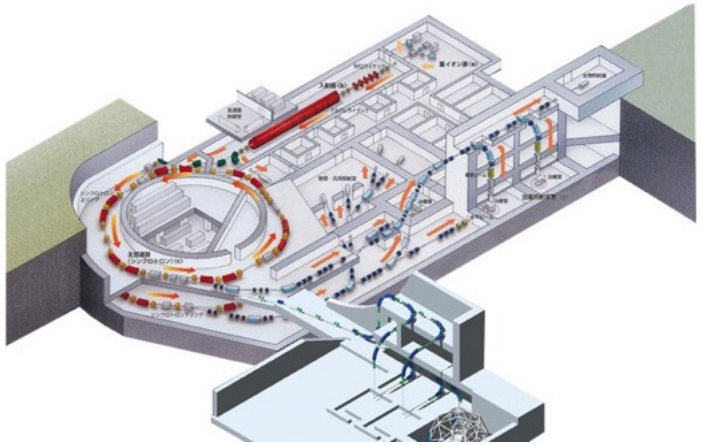
Typical beamtime length : **1 day – a few week**
Typical preparation time : **2 – 3 week (sometimes a few days)**
Similar but **different** detector setup
⇒ DAQ system is also modified in the short-term
⇒ Online monitor…?

RCNP
Z=0-30
0.1<β<0.7

A few concurrent staffs



Accelerator upgraded
10 times intensity

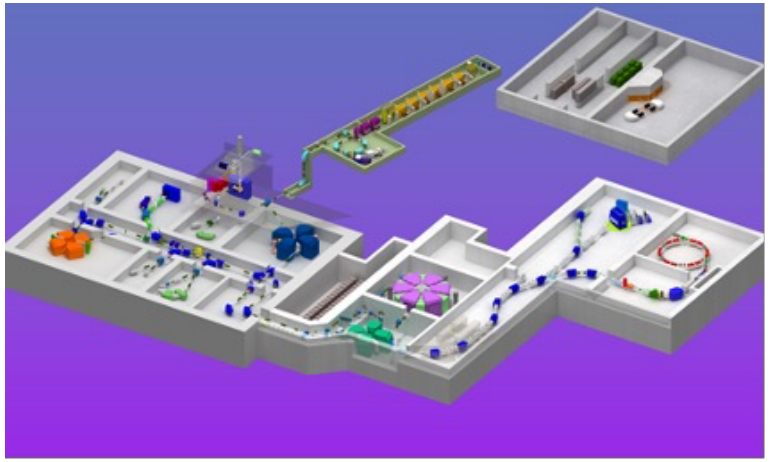


Scrap and build

A few staffs

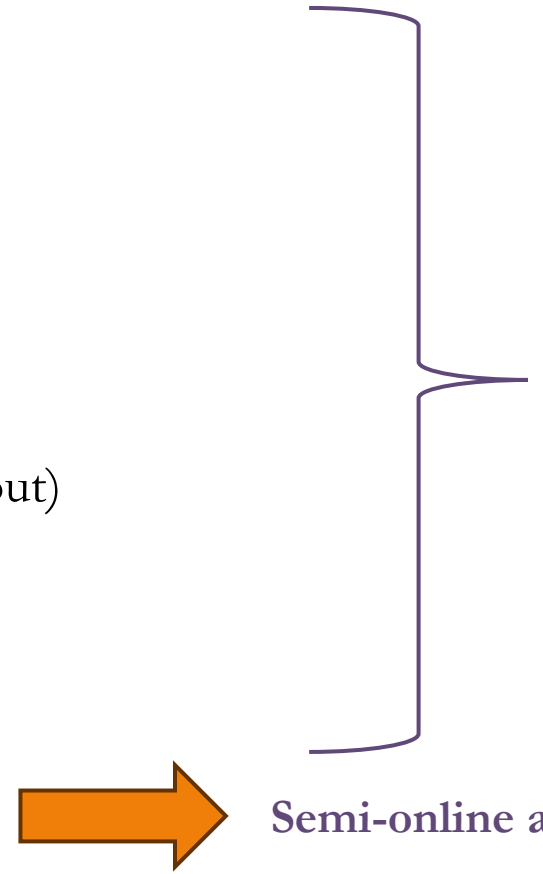
HIMAC
Z=0-54
0.01<β<0.7

RIBF
Z=0-82
0.01<β<0.7



Data and analysis processes there

- Limited types of raw data (at that time)
  - ADC
  - TDC
  - Samples and clocks
  - Timestamp
- **Limited types of detector output**
  (but different # and places of detectors)
  - Scintillator with some PMTs
  - Parallel plate avalanche counters (Delay-line readout, strip readout)
  - Multi-wire drift chambers (but different # of planes and wires)
  - Ion chambers (but different # of readouts)
  - Time projection chambers (but different configurations)
- Common processes of calibration
- Specific processes of calibration and reconstruction



Simple online monitor (very common)

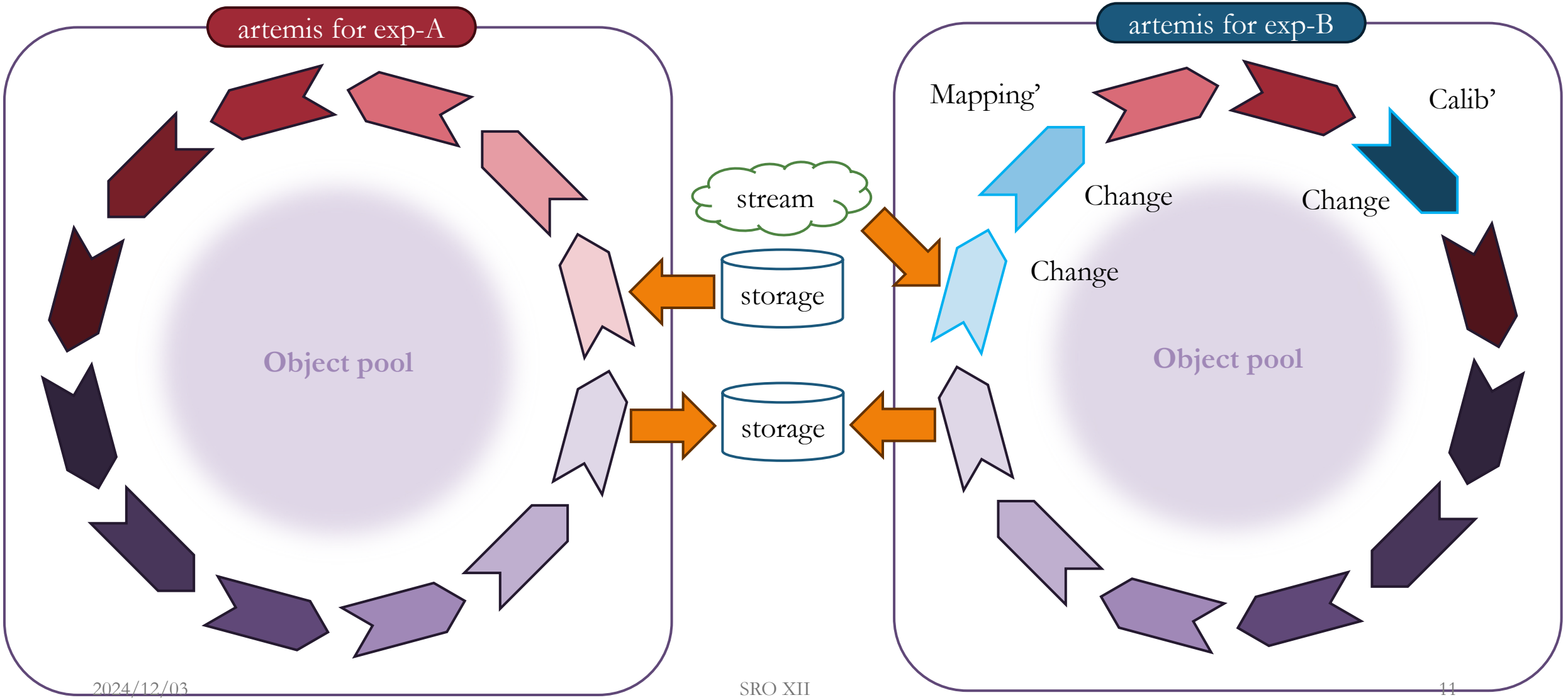
Semi-online and offline

Typical analysis flow

Assumption: ”Event data is built”

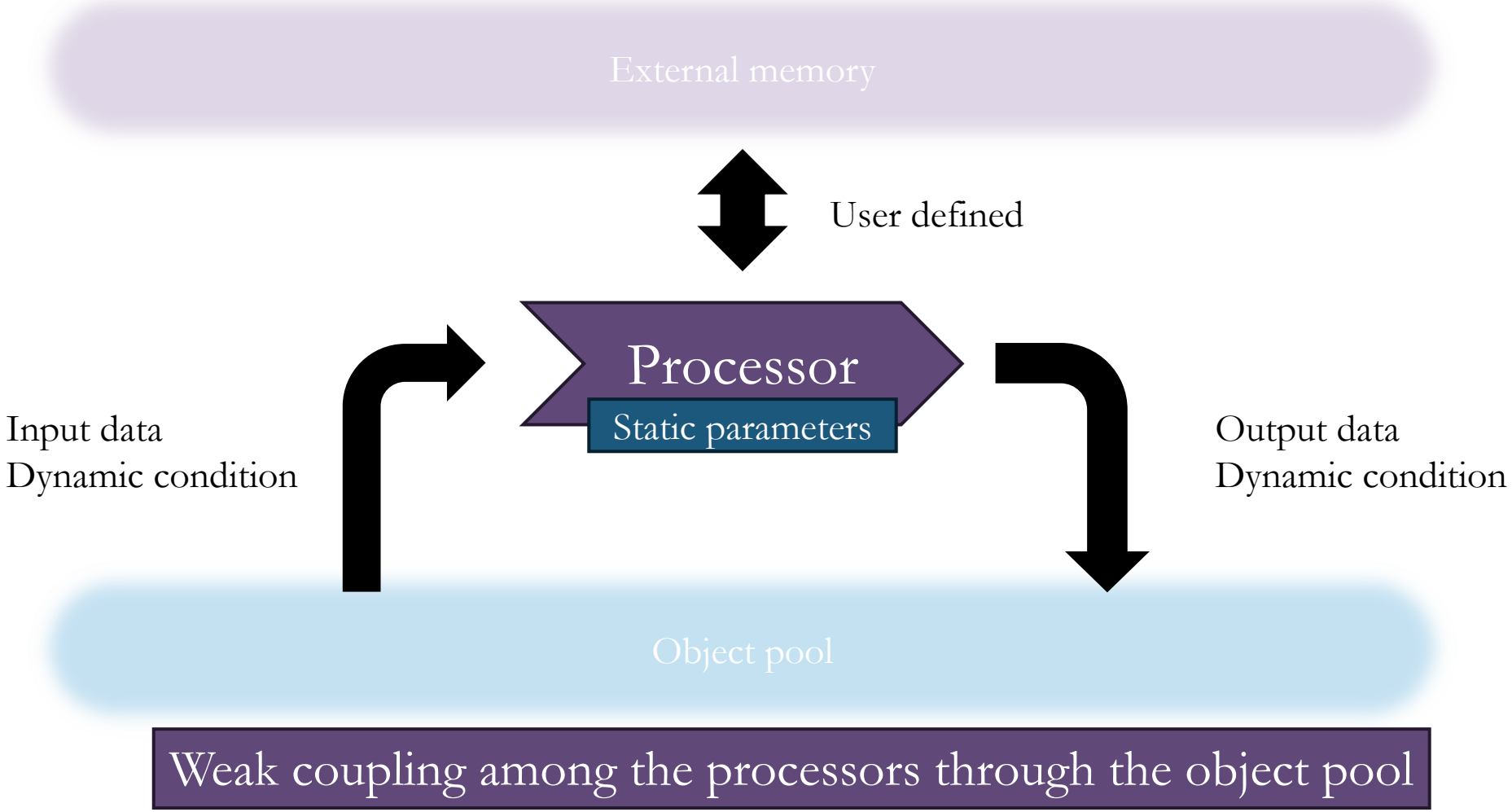
- The flow of the data analysis is logically the same in most experiments.
- Global calibration and reconstruction are specific for each experiment.
- We have to maintain many similar procedures, codes, and programs if we prepare for the analysis programs…

Chain of processes: reuse of the pieces



Processor: the building block of flow

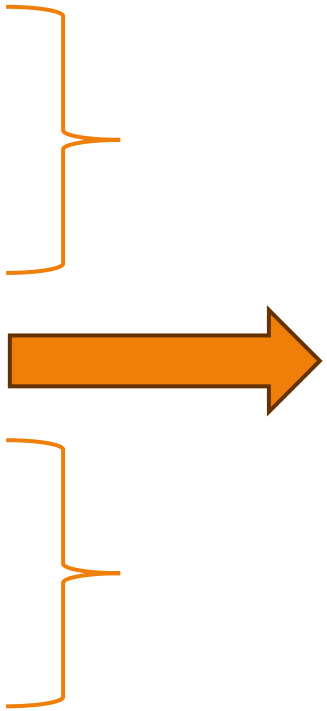
Disk, Streaming socket, Memory…



Weak coupling among the processors through the object pool

TProcessor

- virtual void Init();
- virtual void PreRun();
- virtual void PreLoop();
- virtual void Process();
- virtual void PostLoop();
- virtual void Clear();
- virtual void PostRun();



Preparation of input, output, and parameters.

Actual processing of input data, condition, and output

Clean up of temporal objects.

Example of processor (conceptual)

Input	Processor	Output
No input object	Read from disk or socket **TStreamingDataEventStore**	Rawdata objects
	Read from storage **TParameterLoader**	Calibration parameter set
Calibration parameter set Rawdata w/ Timing and Charge	**TTimingChargeCalibrationProcessor**	Calibrated Timing and Charge
TPC hit cluster TPC Gas Property TPC readout pad response	**TTPCTrackFittingWithDiffusionProcessor**	Track
Histogram Definition Relating objects	Write to memory and/or disk **TTreeProjectionProcessor**	No output object

No code flow building

- Various “Microcode” as a processor
- Human readable configuration file with YAML format
- Automatic instantiation of processes inside the ARTEMIS.

```yaml
Anchor:
  - &input blddata/run@NUM@.bld
#  - &input /home/e483/data/run2005.bld
  - &output output/run@NUM@.root

Processor:
  - name: timer
    type: art::TTimerProcessor

  - name: es_rcnp
    type: art::TRCNPEventStore_ts
    parameter:
      OutputTransparency: 1
      InputFiles:
         - *input

  - name: mapper
    type: art::TMappingProcessor
    parameter:
      OutputTransparency: 1
      MapConfig: mapper_tamidaq.conf

  - include: tref.yaml
  - include: pla_gr.yaml
  - include: vdc_gr.yaml

  - name: hist_pla_gr
    type: art::TTreeProjectionProcessor
    parameter:
      Type: art::TTreeProjection
      FileName: hist/pla/pla_gr.hist.yaml

  - name: outputtree
    type: art::TOutputTreeProcessor
    parameter:
      FileName: *output
```

ARTEMIS user interface

- Linux-command-line-like experience
- Easy to add new commands
  - Packaging the well-used procedures

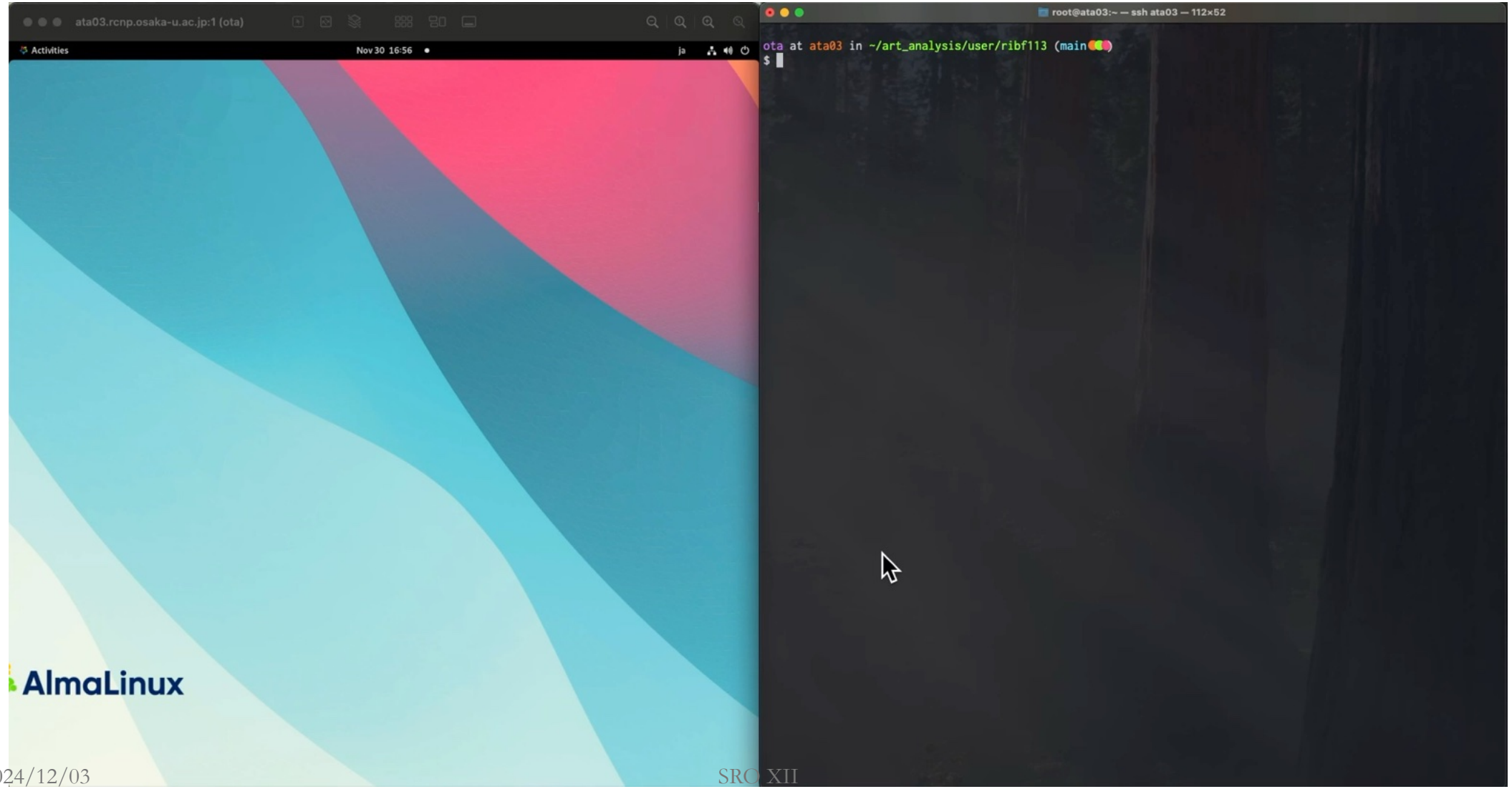
```
artemis [54] ls

 output/hoge.root
    0 art::TAnalysisInfo analysisInfo
    1 art::TArtTree tree                tree
>   2 TH2F hh                vdc_gr_clust_x2.cluster_angle:vdc_gr_clust_x2.cluster_pos

artemis [55] ht 2
artemis [56]
```

Booking the histograms

- Human readable variables
- Applying cut

```
group:
  - name: beam
    title: beam profile
    contents:
      - name: diaf3_v1290_all
        title: diaf3 all
        x: ["diaf3_v1290_all.fTiming",1200,-11000.,1000.]
      - name: diaf7_v1290_all
        title: diaf7 all
        x: ["diaf7_v1290_all.fTiming",1200,-11000.,1000.]
```

How it works


ata03.rcnp.osaka-u.ac.jp:1 (ota)
Activities
Nov 30 16:56
ja
AlmaLinux
root@ata03:~ — ssh ata03 — 112×52
ota at ata03 in ~/art_analysis/user/ribf113 (main
$

ARTEMIS: Requirements

- Platform: Linux, Mac
- Language: C++17
- Dependency
  - cmake
  - ROOT 6.32.0 (especially for Mac)
  - yaml-cpp 0.6.3
  - GETDecoder (optional)
  - zmq (optional)
  - redis++/hiredis (optional)

ARTEMIS: Implementation of framework

- Object oriented programing
  - Using Dictionary of ROOT.
  - Future: less dependent of the ROOT.
- Monitoring with web browser using JSROOT.
- Parallel processing with Open-MPI
- CUI based control.
- AI/ML can be supported through the implementation of the corresponding processor

Available Event Sources

- RIBF Data Format
- RCNP Data Format
- ROOT Tree Input
- Some Monte Carlo Generators for TPCs
- **Streaming Data Format (and online streaming via ZMQ)**

Adapted detectors and physics analysis

- Local calibrations and global corrections for
- Scintillator detectors
- Multi wire drift chambers
- Cathode readout chambers
- Strip readout parallel plate avalanche counters
- Time projection chambers

SPADI Alliance

Signal Processing and Data Acquisition Infrastructure Alliance
for the standard data acquisition and processing system

World-wide common issue

Treatment of quantum big data

Quantum beam
10 times and more

×

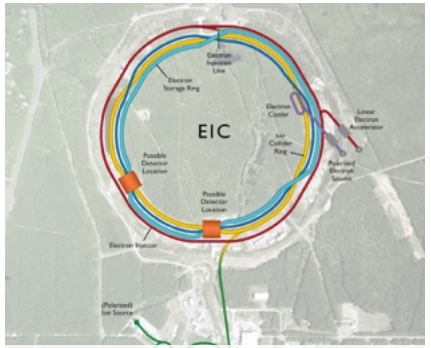
High granularity
for the high resolution and high intensity

=

Quantum bigdata
1000 times larger amount data

Upgrade of the AVF Cyclotron 10 times larger intensity is available

400 Tbps data generation at the FEE



400Tbps

EIC (commitment from JAPAN)

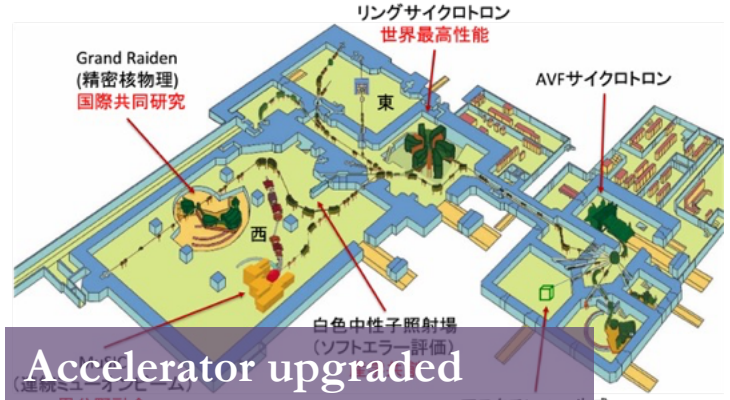
Beyong the limitation of the present data acquisition and processing

Ion accelerators and DAQ staffs in Japan

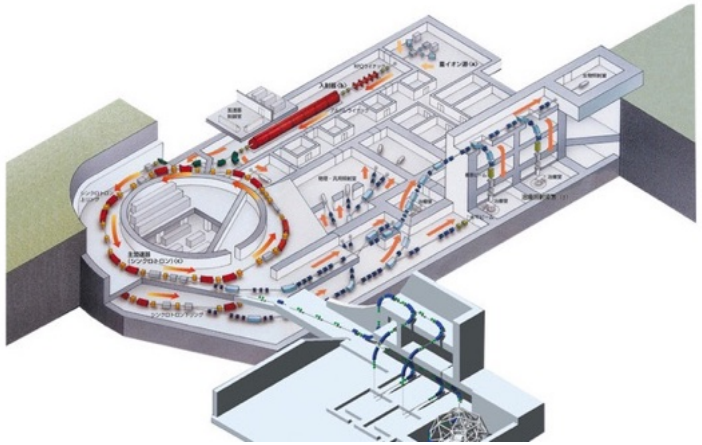
J-PARC
Z=0,1
MIP

RCNP
Z=0-30
0.1<β<0.7

Within the collaboration	A few concurrent staffs



Accelerator upgraded
10 times intensity



Scrap and build	A few staffs

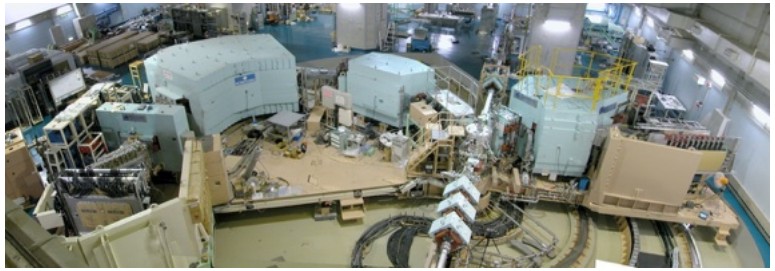
HIMAC
Z=0-54
0.01<β<0.7

RIBF
Z=0-82
0.01<β<0.7

Scales in our plan

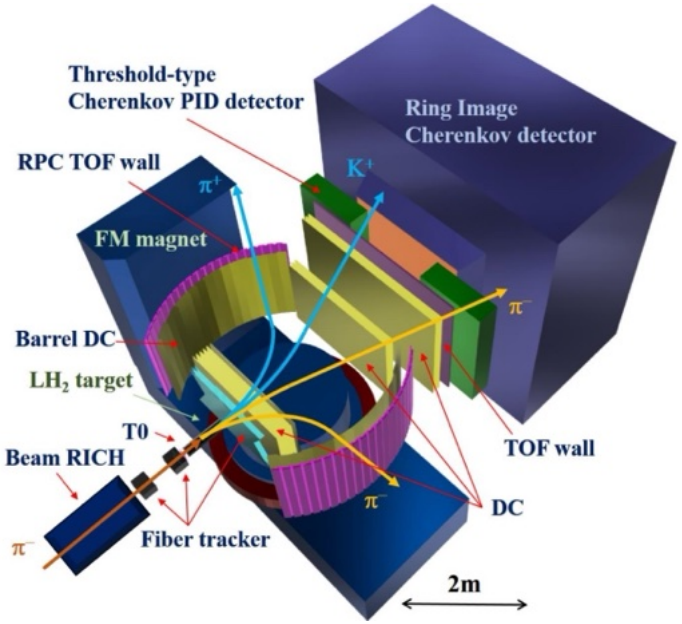
2k channels

double-arm spectrometer
@RCNP



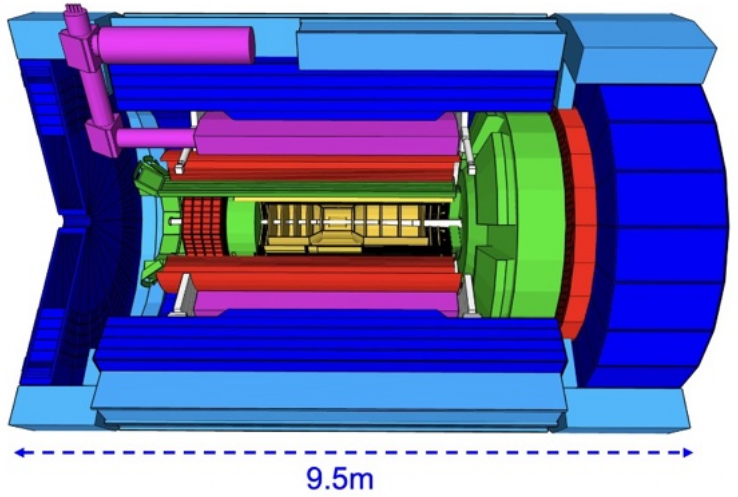
30k channel

complex spectrometer
@J-PARC



10M channels

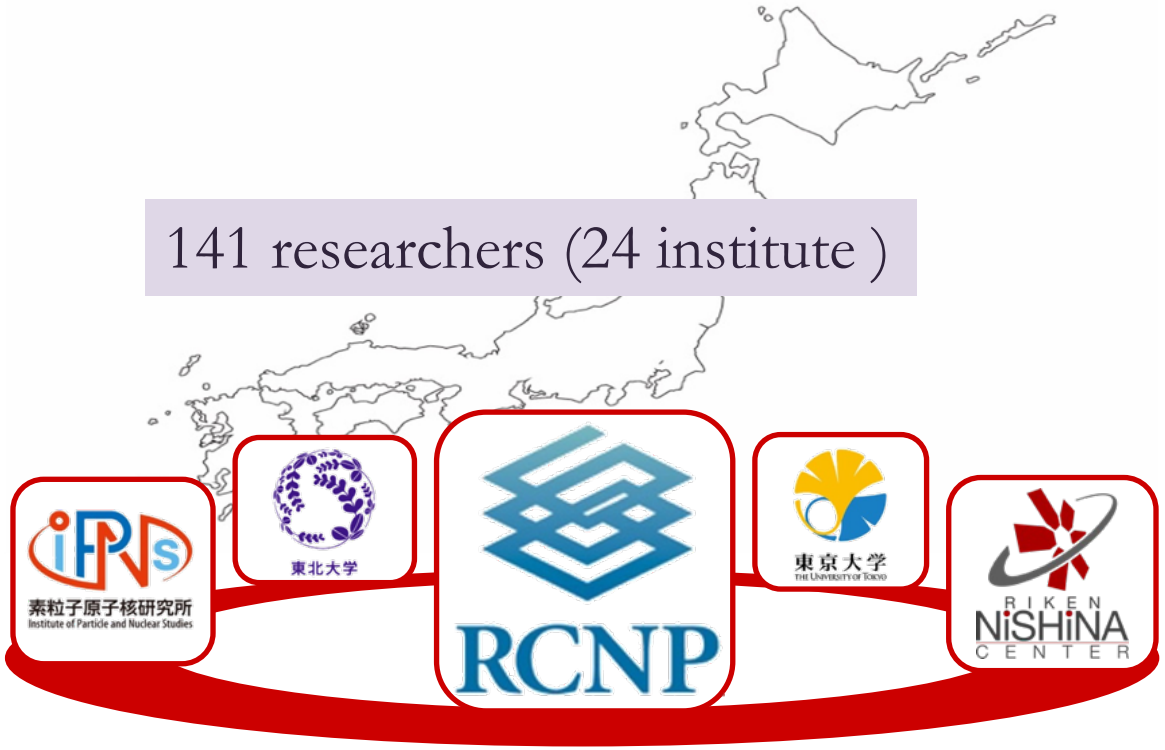
complex spectrometer
@EIC



from small to large

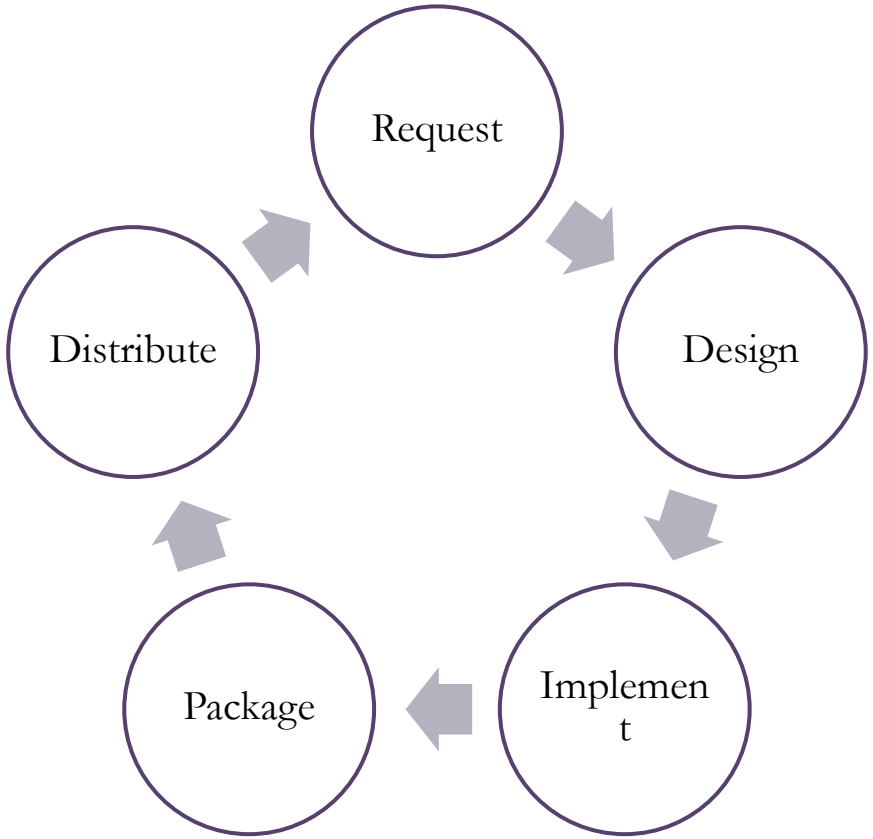
SPADI Alliance

Signal processing and data acquisition infrastructure alliance
toward the standardization for sustainable developments



Strategy comittee

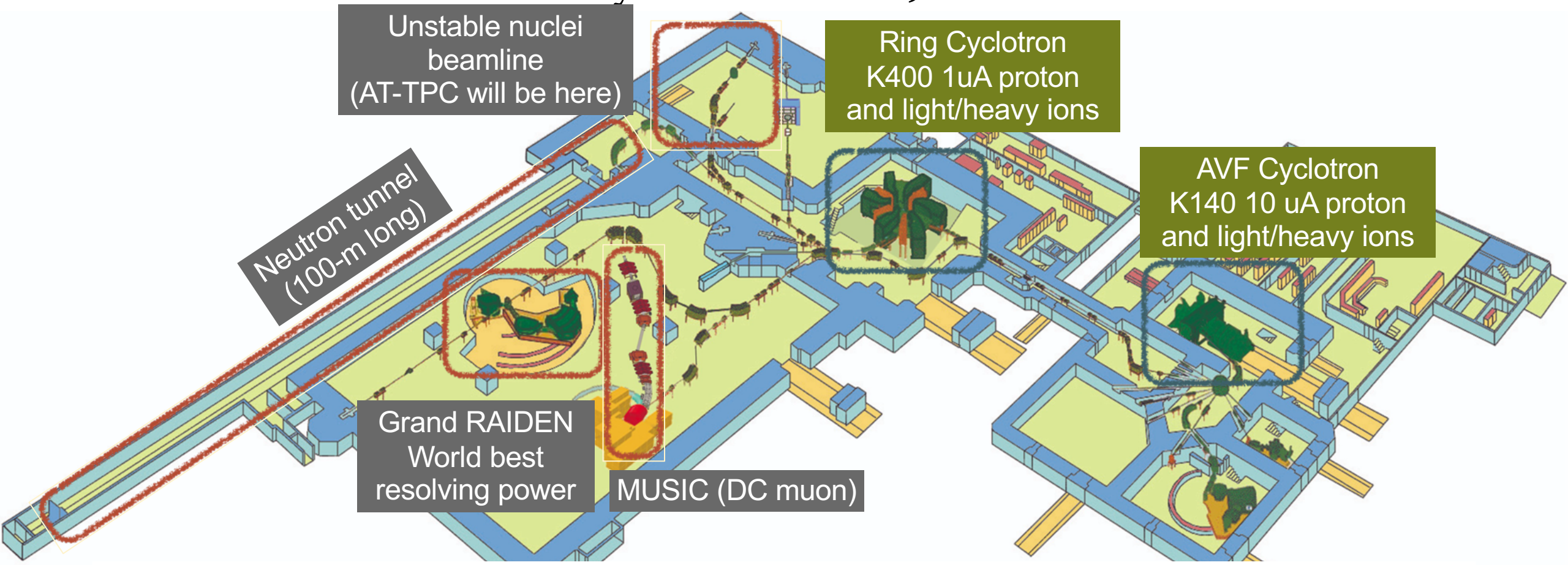
Process to standardization



History of SPADI Alliance

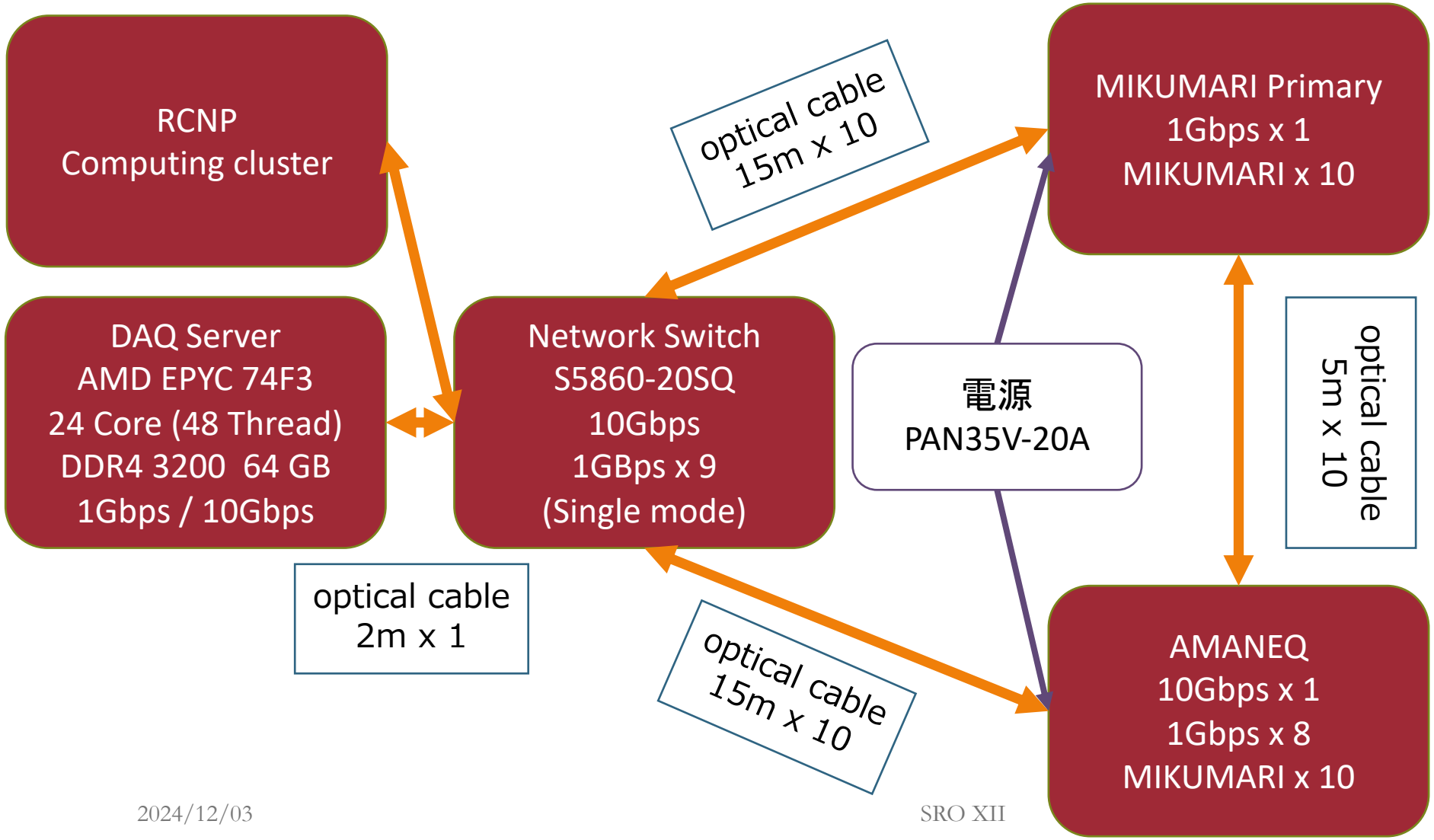
- 2022.05 Initiated
  - FY2022 Discussion for the implementation and FEE developments
- 2023.03 Town meeting in JPS and Annual workshop
- 2023.03 Test Implementation of S-DAQ at RCNP
- 2023.06 Test Implementation of S-DAQ at J-PARC
- 2023.06 Bylaws are issued
- 2023.07 First physics experiment with S-DAQ
- 2023.09 Laboratory exercise of DAQ implementation
- 2024. Implementations in many institutes and experiments…

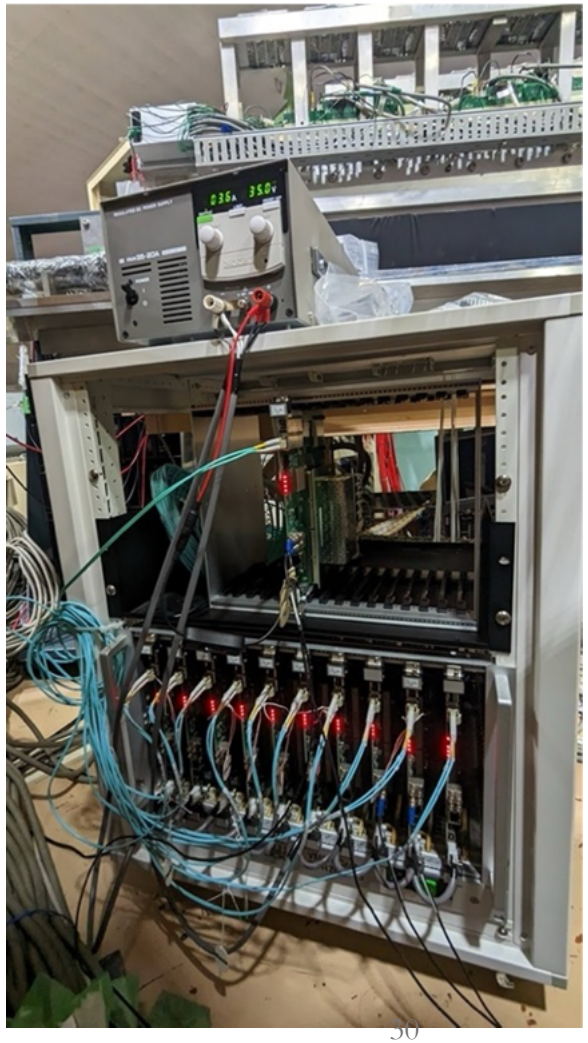
Accelerator facility in RCNP, Osaka



Ion beams (protons, deuterons, and heavy ions), continuous muon beams, and white neutrons available
Used not only in basic physics, but also in medicine (RI drugs), archaeology, engineering, etc.

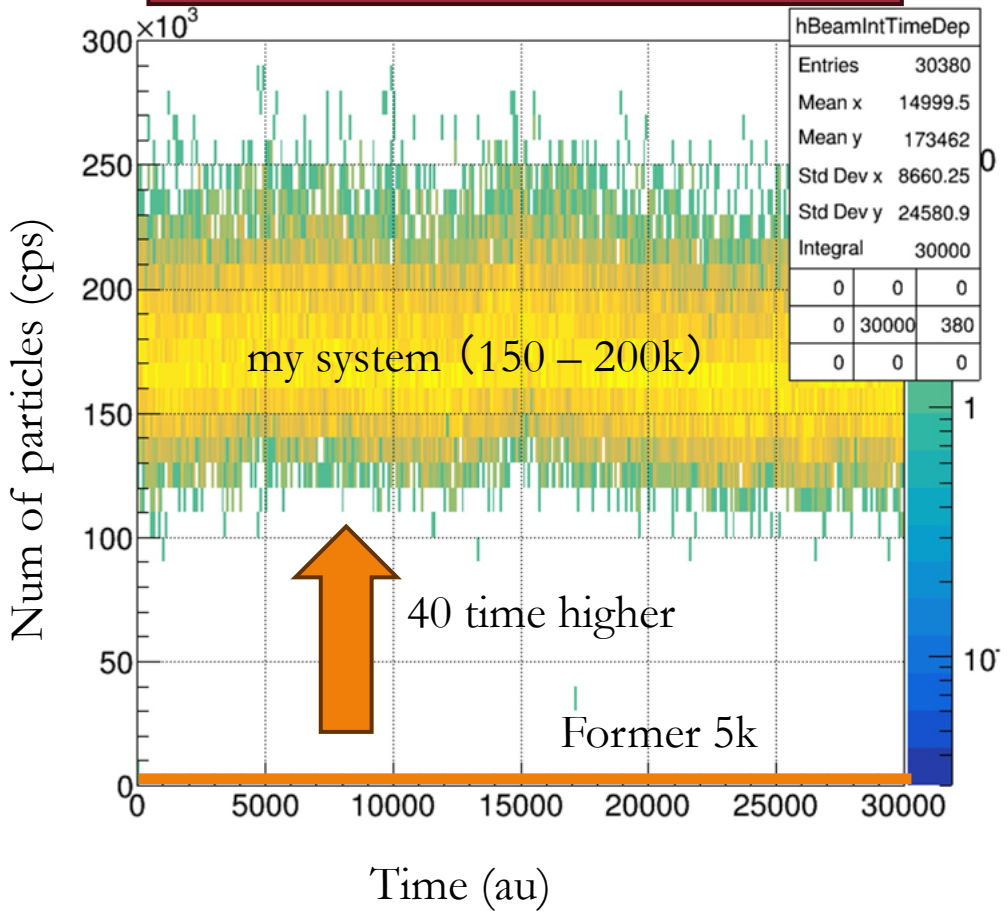
Network configration

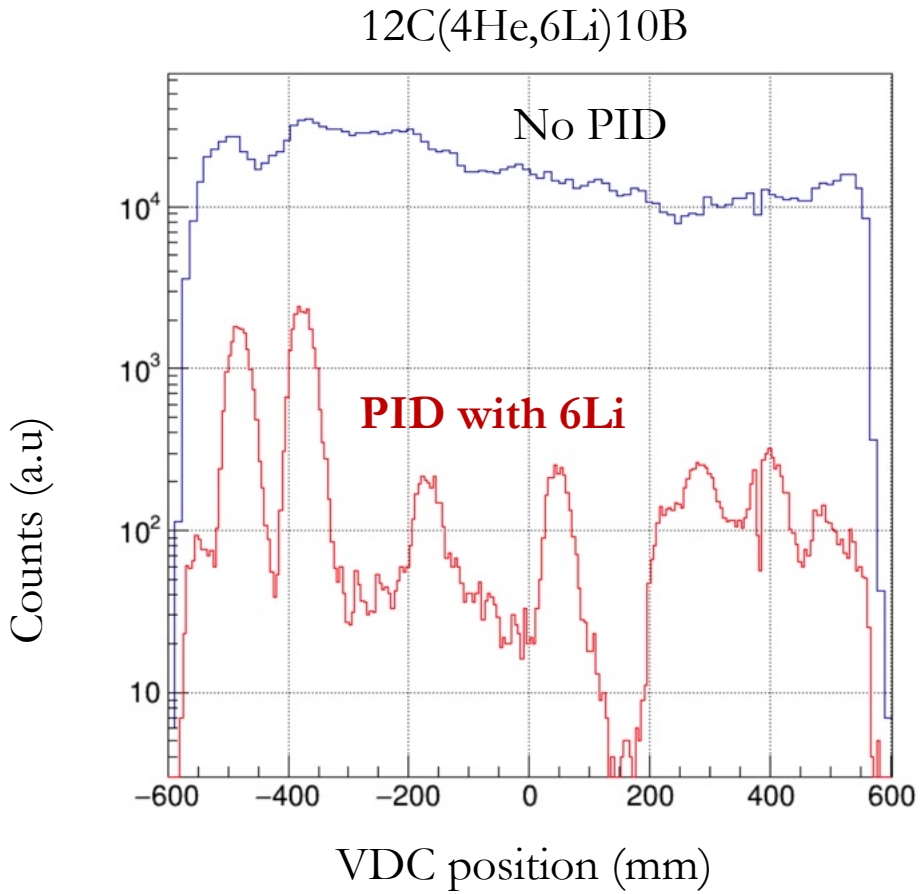




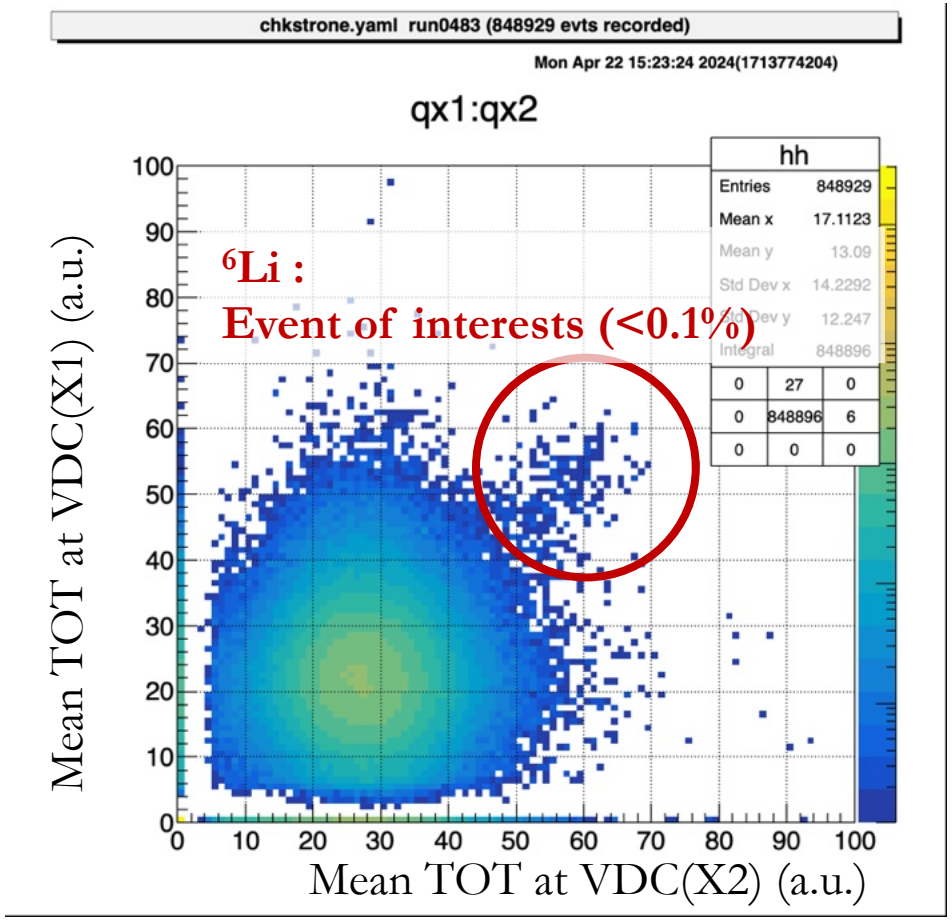
Example of the reaction measurement

ARTEMIS Analyzed Timeframe Data



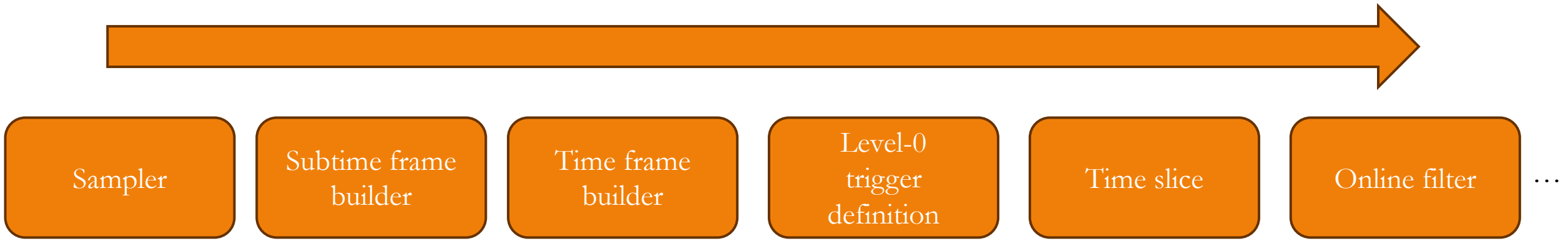


Needs more development : online filter



- Data can be recorded but with S/N < 0.1%
- Consume the storage and analysis time by meaningless data…
  - ex.) 17 TB for 5 days
- Online filtering is strongly required.
  - Particle identification
  - Tracking and vertex reconstruction
  - Physics observables

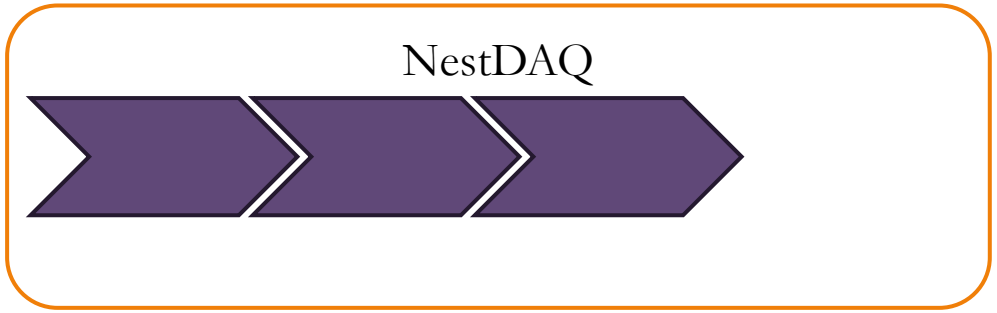
Filtering process can be prepared for NestDAQ independently but there is already…

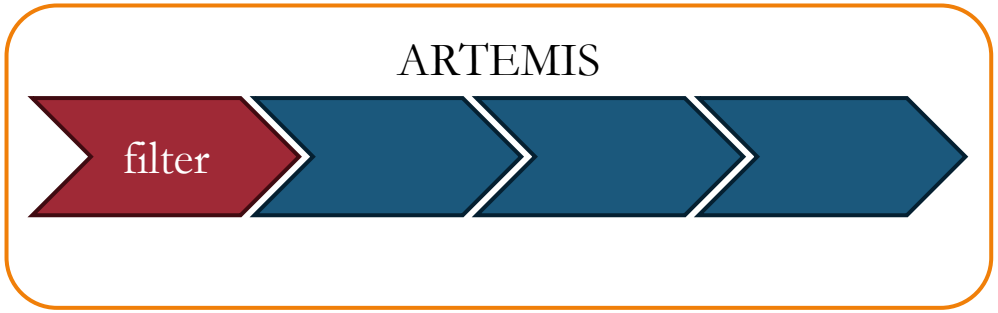


Can we share the algorithm?

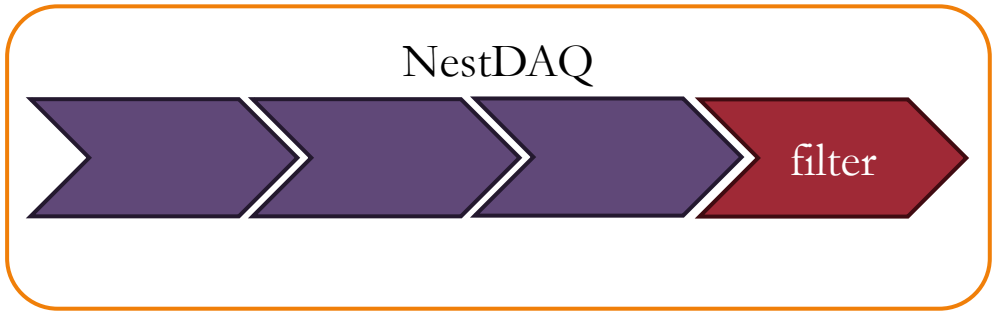
Many useful codes for various detectors already exist (not only in ARTEMIS)
Online processing algorithm should be checked in the offline (semi-online) analysis at some point.

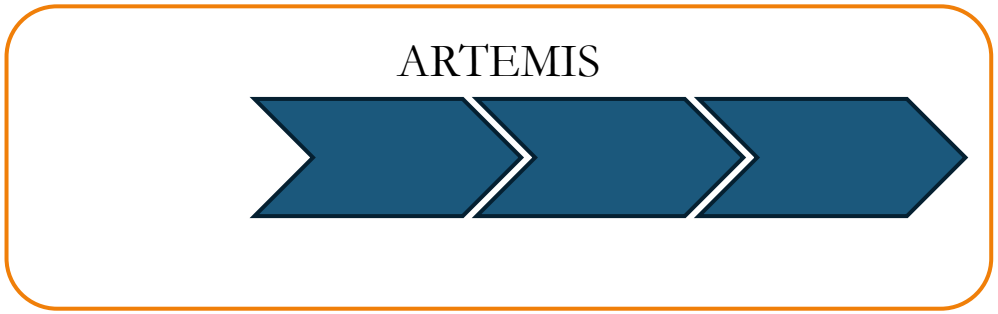
Debug phase





Production phase

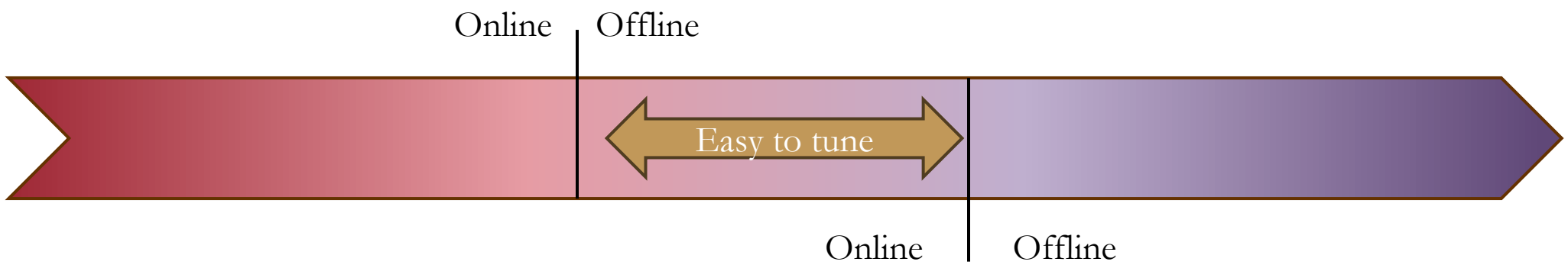




Multi process communication

Single/multi thread (no communication so far)

How to efficiently change the border ?



1: Algorithm package sharing

Mapping the algorithm, conversion of data object may be required but just we do it

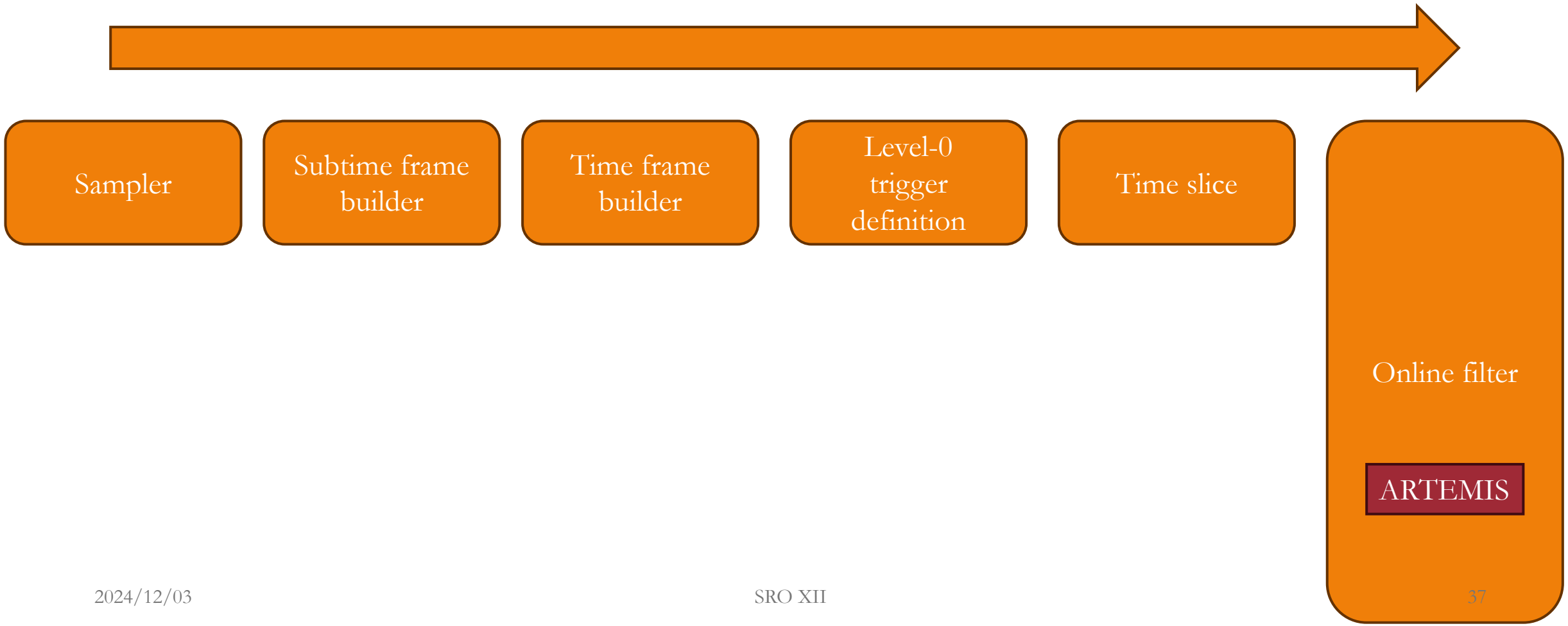
ARTEMIS Processor

- virtual void Init();
- virtual void PreRun();
- virtual void PreLoop();
- virtual void Process();
- virtual void PostLoop();
- virtual void Clear();
- virtual void PostRun();

NestDAQ Device

- virtual void InitTask()
- virtual void PreRun();
- virtual void ConditionalRun();
- virtua void PostRun();

2: Contain ARTMIS as a device of NestDAQ …



Summary

- SPADI Alliance for the standardization of the scalable streaming DAQ system
  - Domestic researchers have started considering using the same DAQ system
  - Preparing for international collaboration
- ARTEMIS for the easy and quick modification of online-offline analysis
  - No-code processing flow building
  - Many event sources: RIBF, RCNP, Streaming…
  - Linux-like experience in user interface
- Online-offline crossover by sharing the algorithm package
  - Start consideration in SPADI Alliance