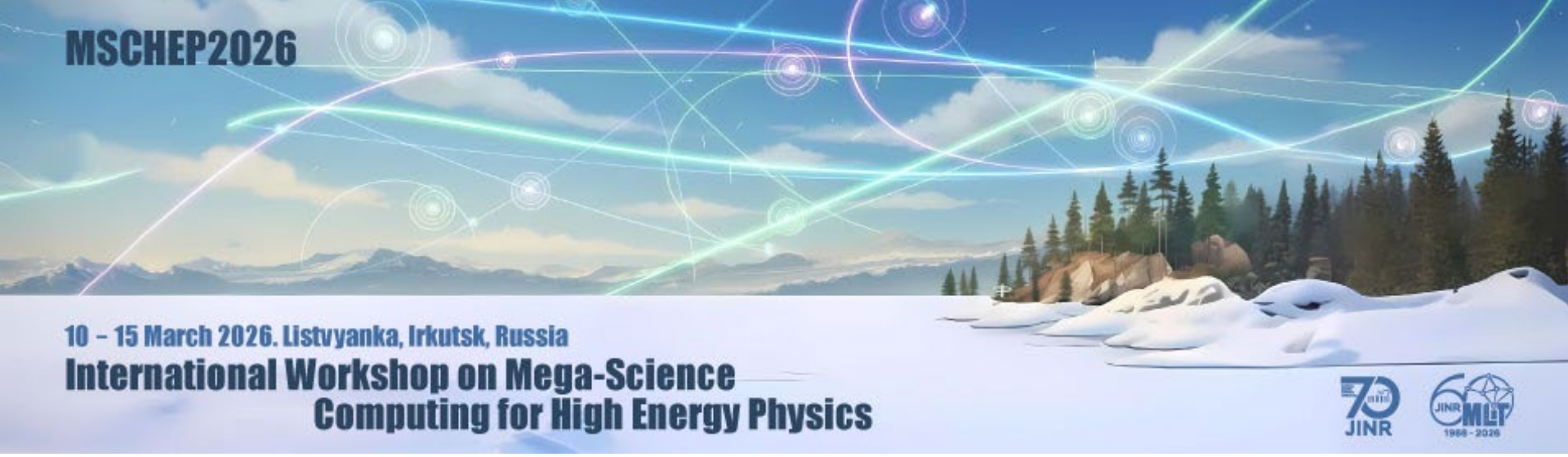


Automated Offline Data Processing in JUNO

Tao Lin

IHEP

March 10, 2026

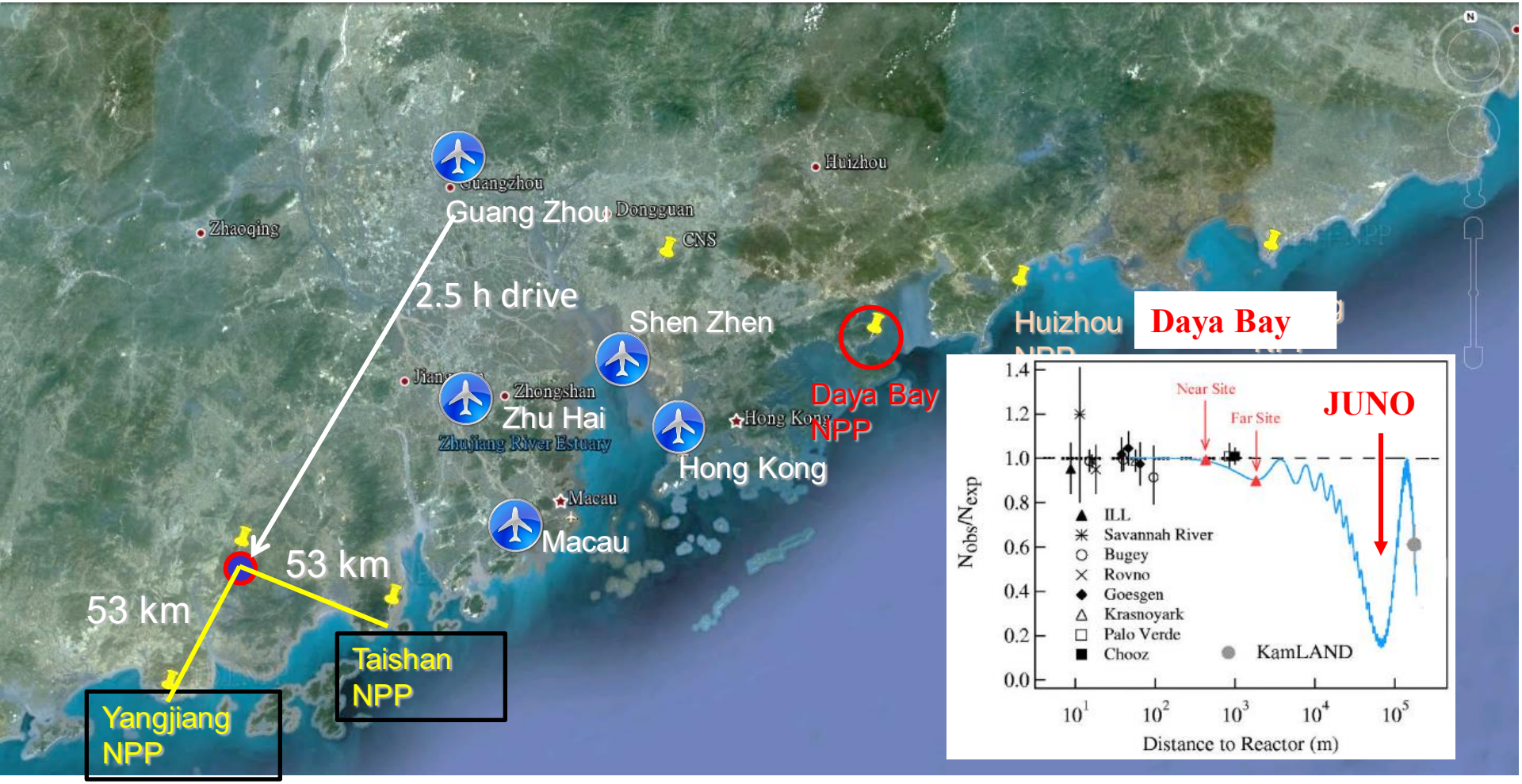
Outline

- Overview of JUNO data processing
- Design of Keep Up Production pipeline
- Micro-service based KUP software
- Application in JUNO and JUNO-TAO
- Summary

Outline

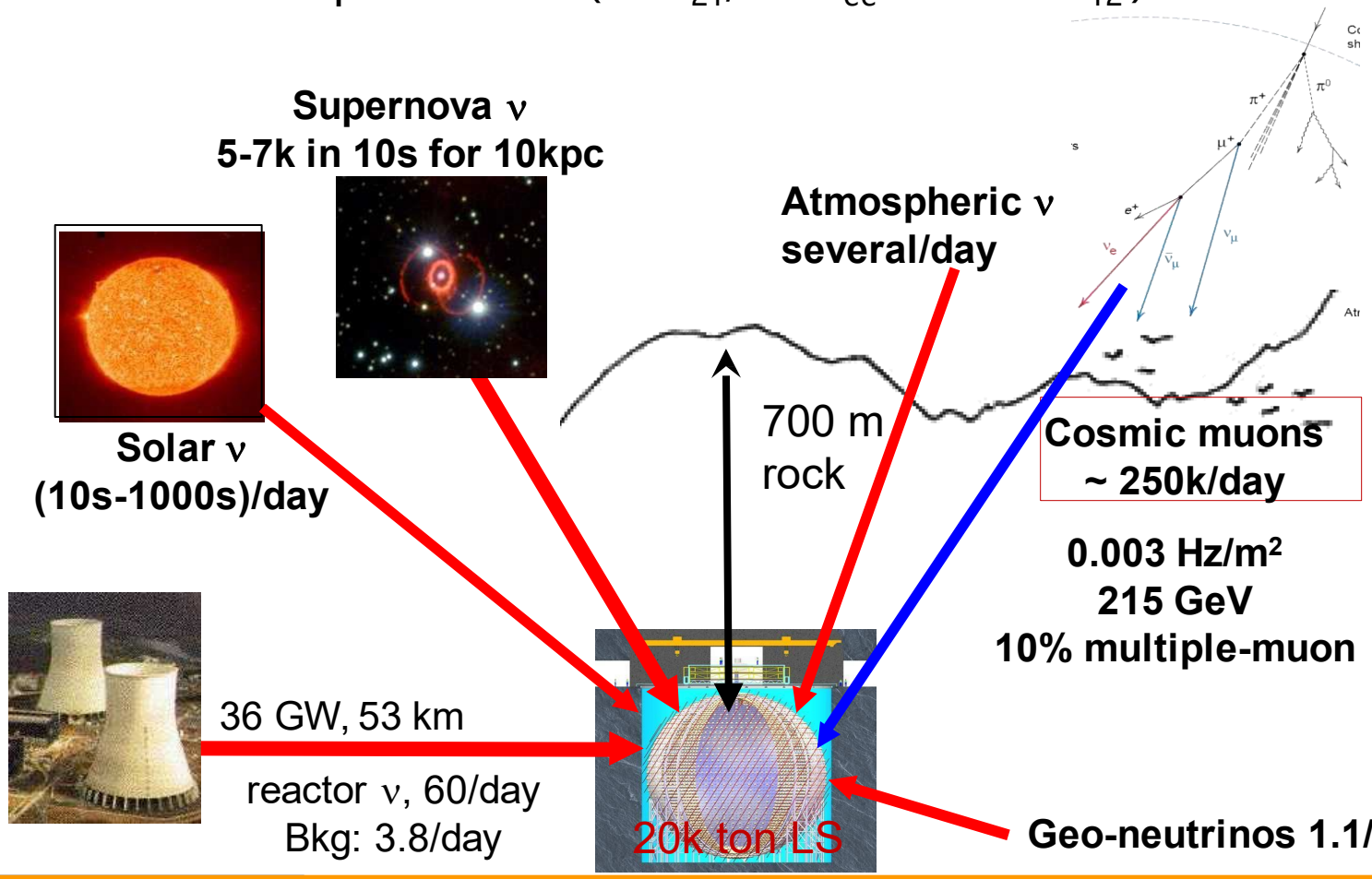
- Overview of JUNO data processing
- Design of Keep Up Production pipeline
- Micro-service based KUP software
- Application in JUNO and JUNO-TAO
- Summary

JUNO: Jiangmen Underground Neutrino Observatory



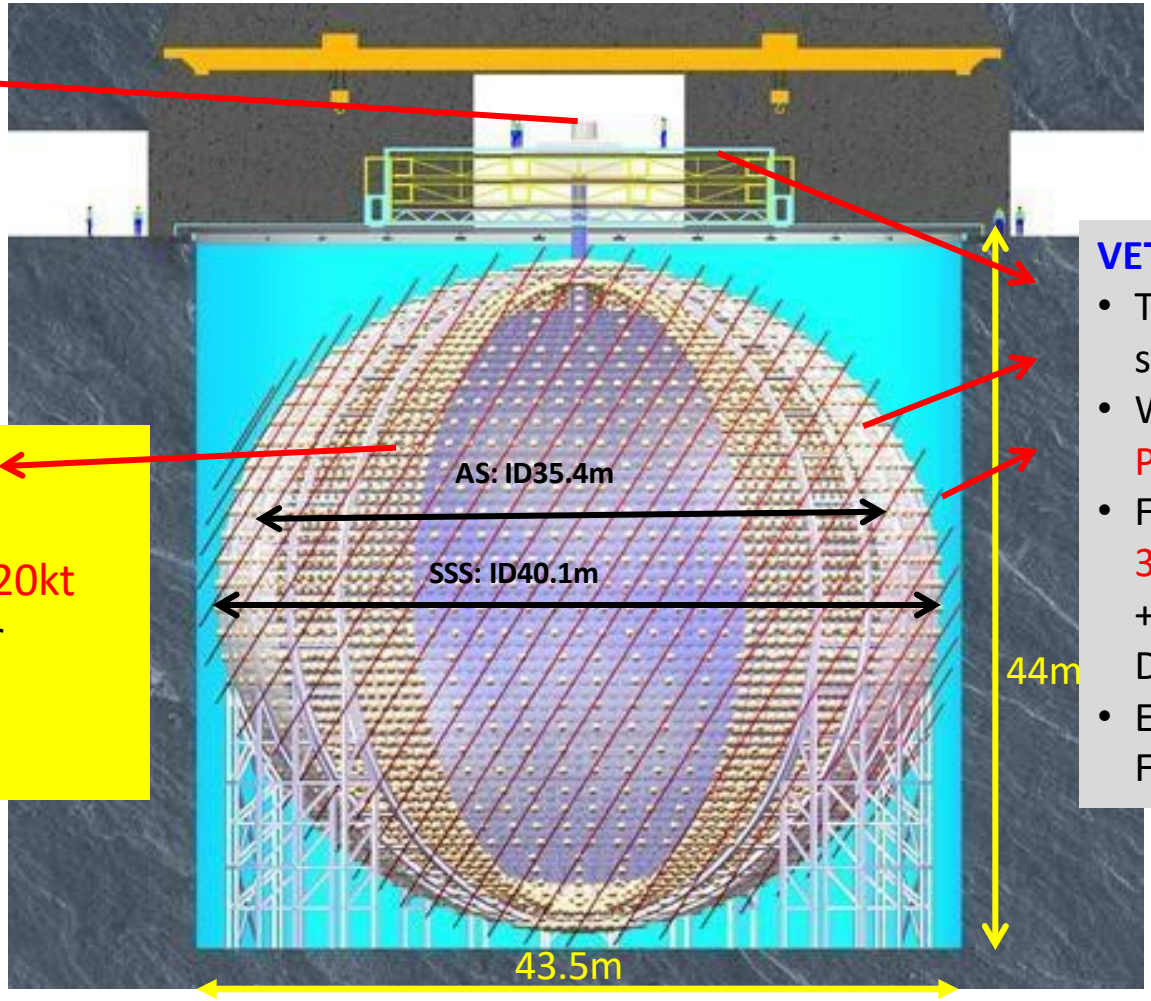
Physics at JUNO

- Determination of mass hierarchy (MH)
- Precision of three parameters (Δm^2_{21}, Δm^2_{ee} and $\sin^2\theta_{12}$)



JUNO Detector Design

Calibration



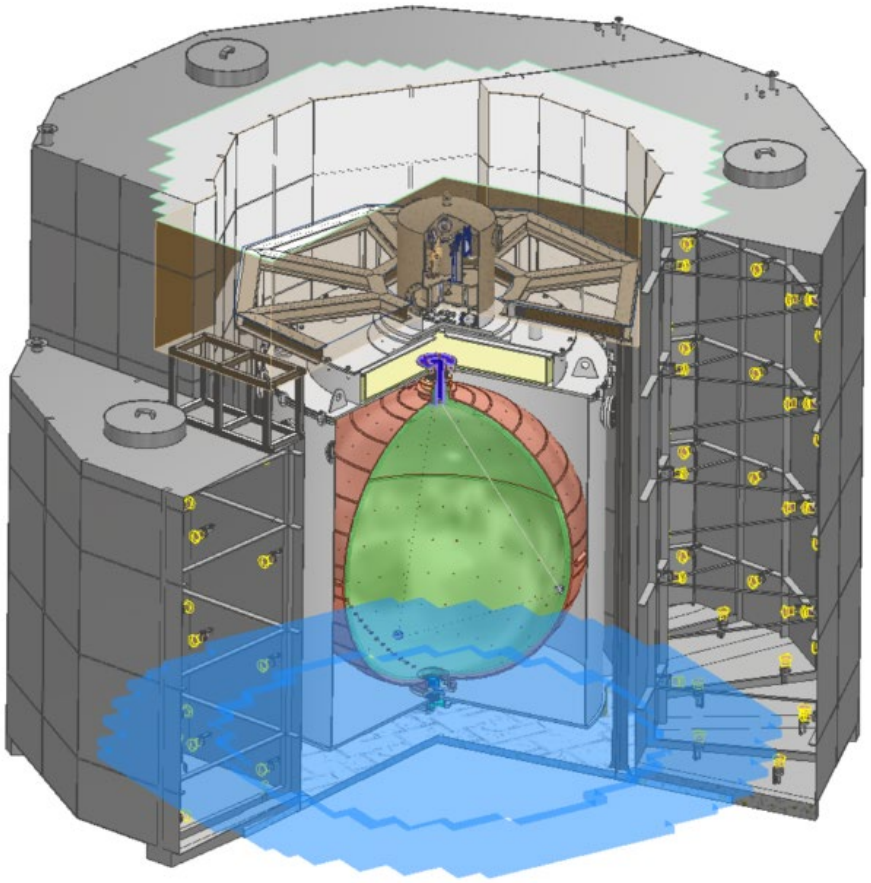
Central detector
- Steel structure
- Acrylic sphere + 20kt Liquid scintillator
- 17612 20" PMT
- 25600 3'' PMT

VETO system
- Top Tracker: plastic scintillator
- Water + 2400 20'' PMT
- For atmospheric ν: 348 20" spare PMTs + 600 8" PMTs from Daya Bay
- Earth Magnetic Field shielding coils

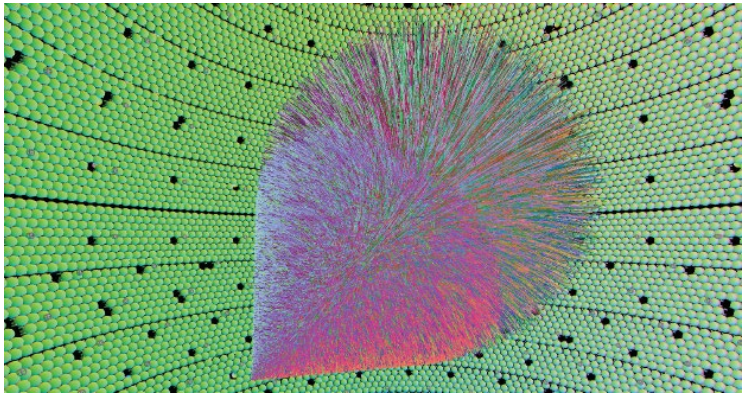
arXiv:1508.07166
Prog. Part. Nucl. Phys. 123(2022), 103927

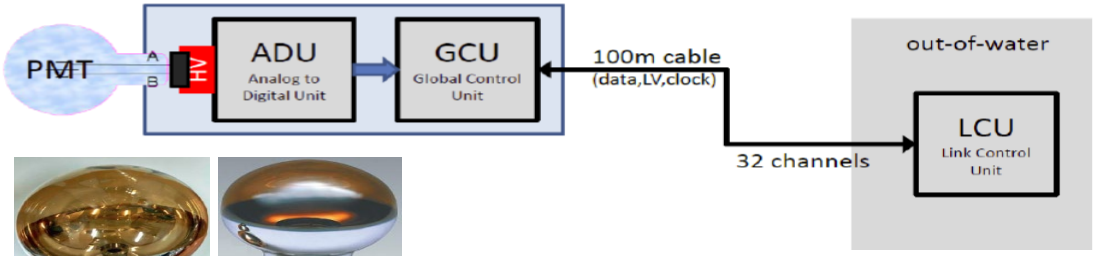
JUNO-TAO Detector Design

- Central Detector (CD)
  - 2.8 tons Gd-LS in a spherical vessel, 1-ton FV, ~2000 IBDs/day
  - 10 m^2 SiPMs to collect scintillation light
  - 8048 readout channels
- Water Tank (WT)
- Top Veto Tracker (TVT)
- Data throughput
  - ~100 Mb/s, about 1/5 of JUNO



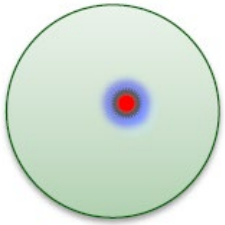
JUNO offline data processing



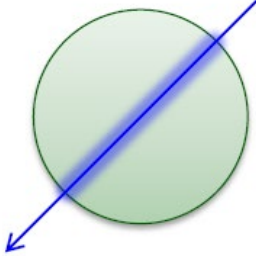


NNVT MCP-PMT **Hamamatsu R12860**

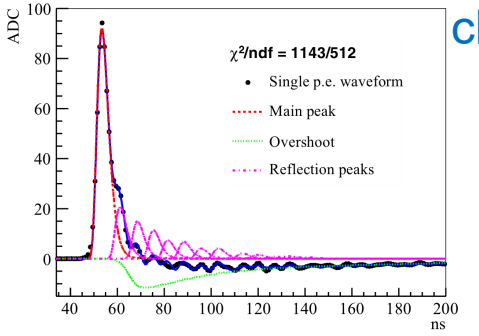
Waveform
ADC/TDC



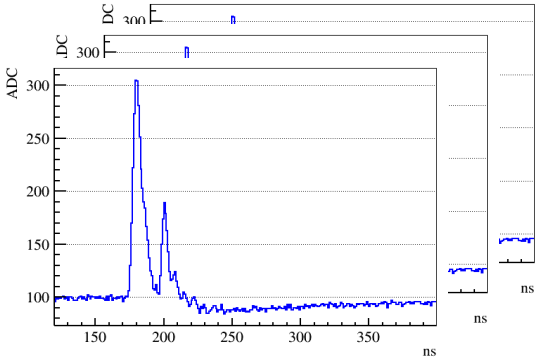
Point-like
Energy/Vertex



Tracking
Direction/Track Length

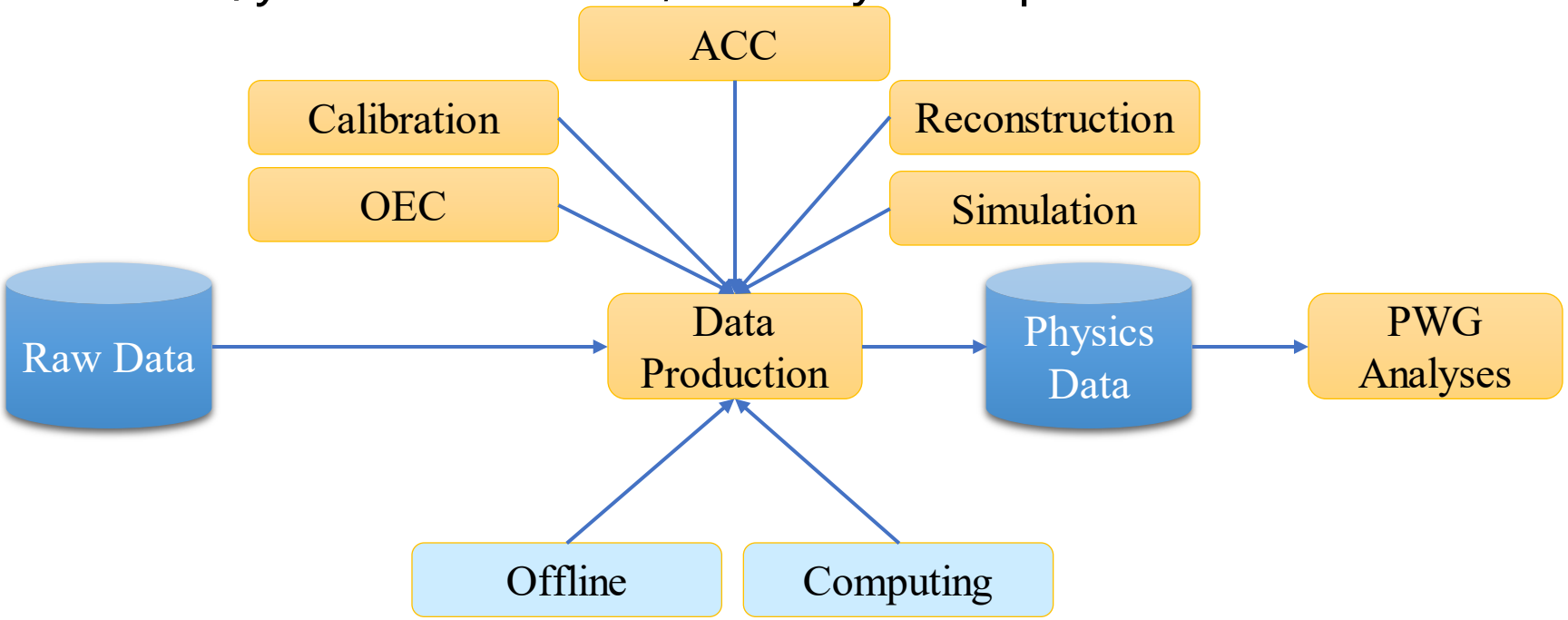


PMT hits
charge/time

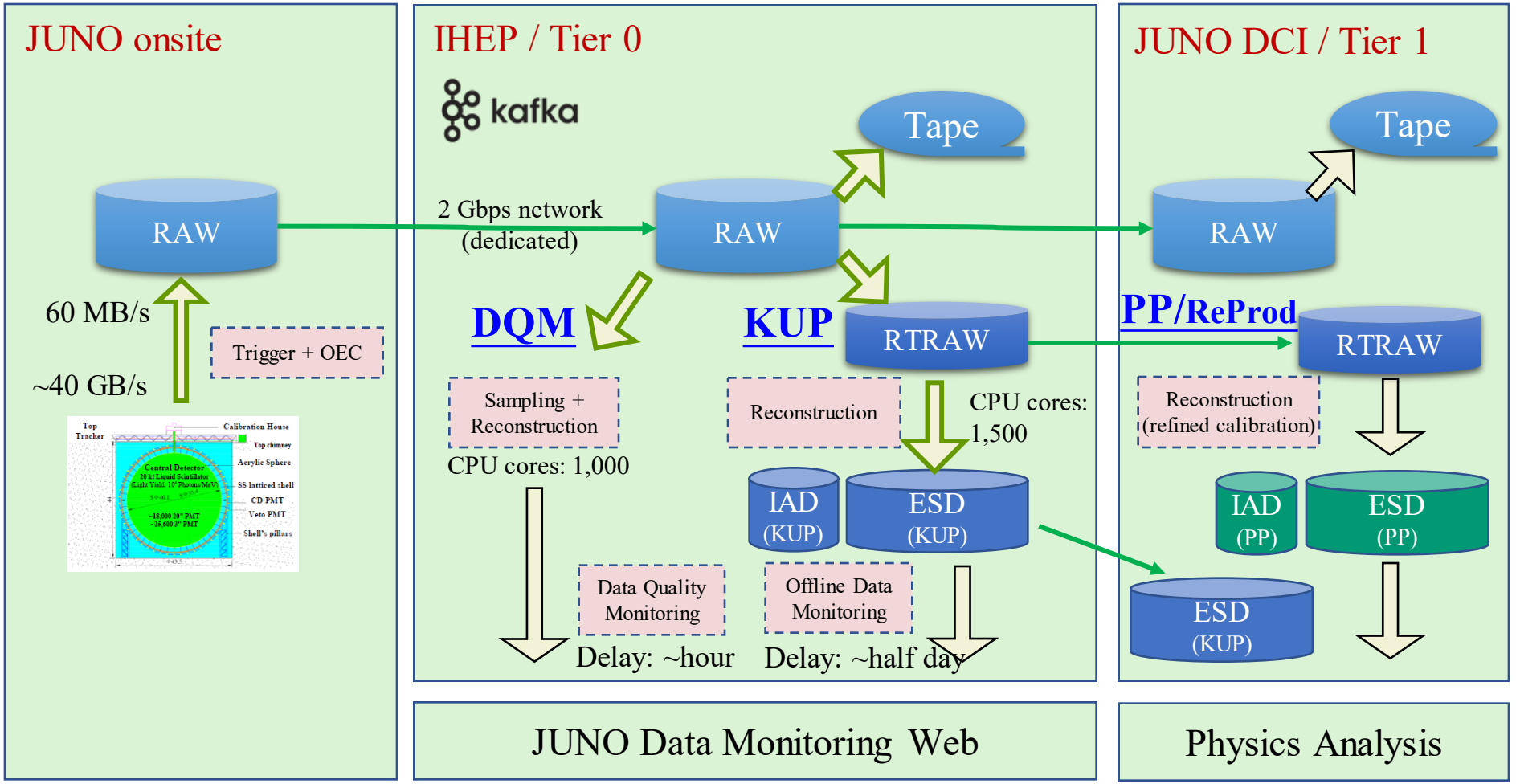


Data production

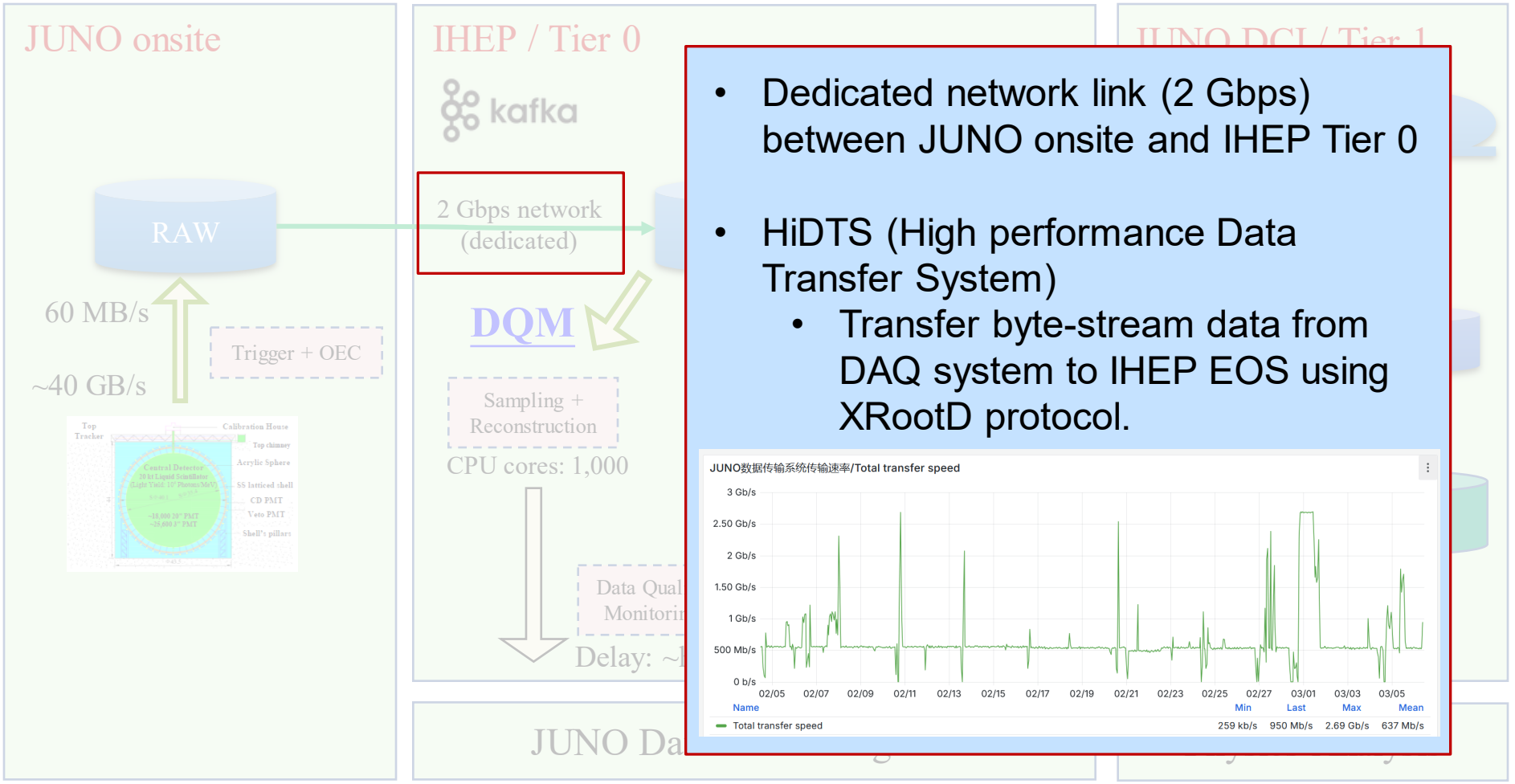
- Data production group is in charge of offline data processing.
- Challenge:
  - 2 PB/year of RAW data, 20~30 years operation.



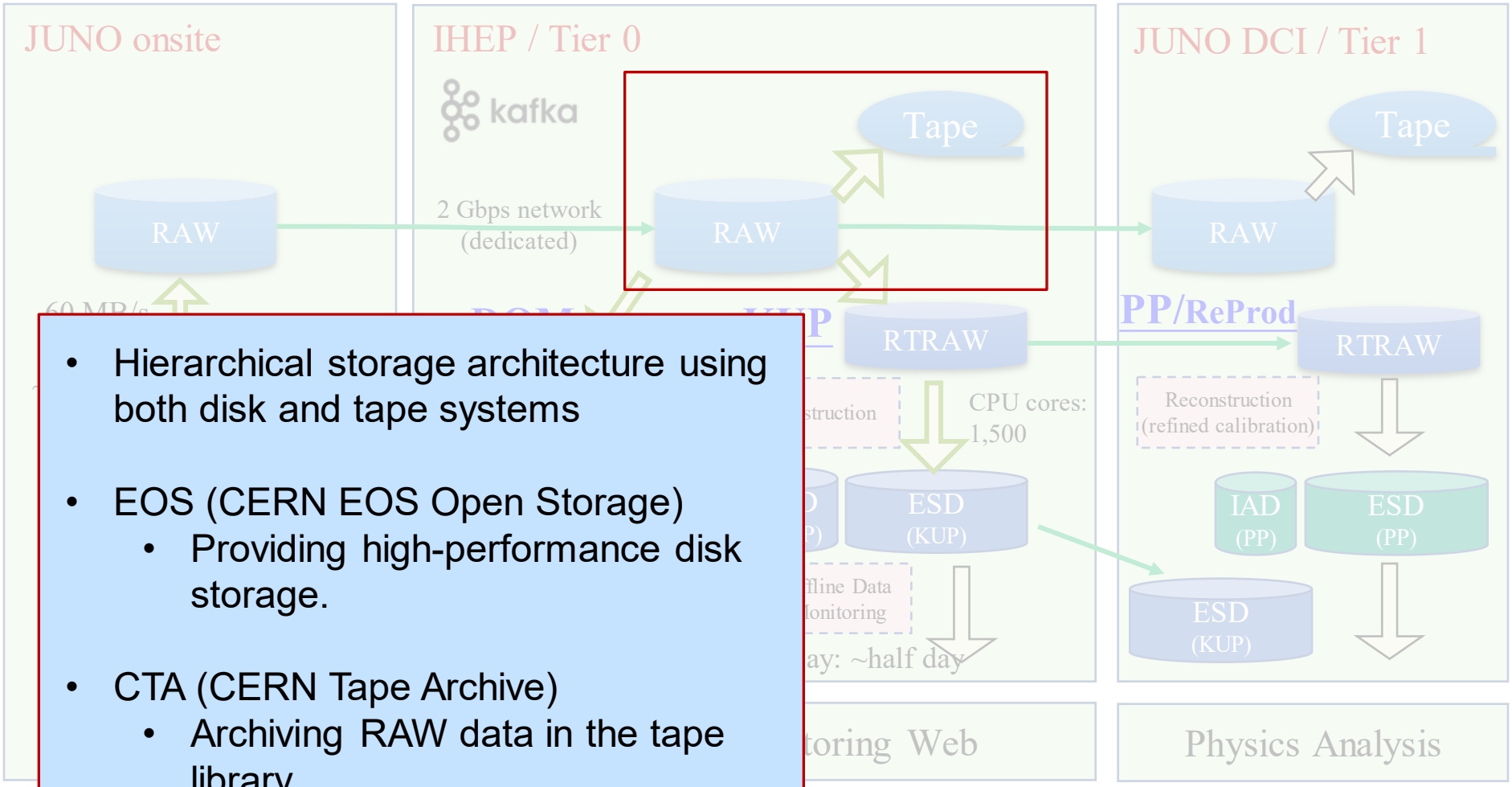
Overview of JUNO data processing



Data transfer from onsite to Tier-0

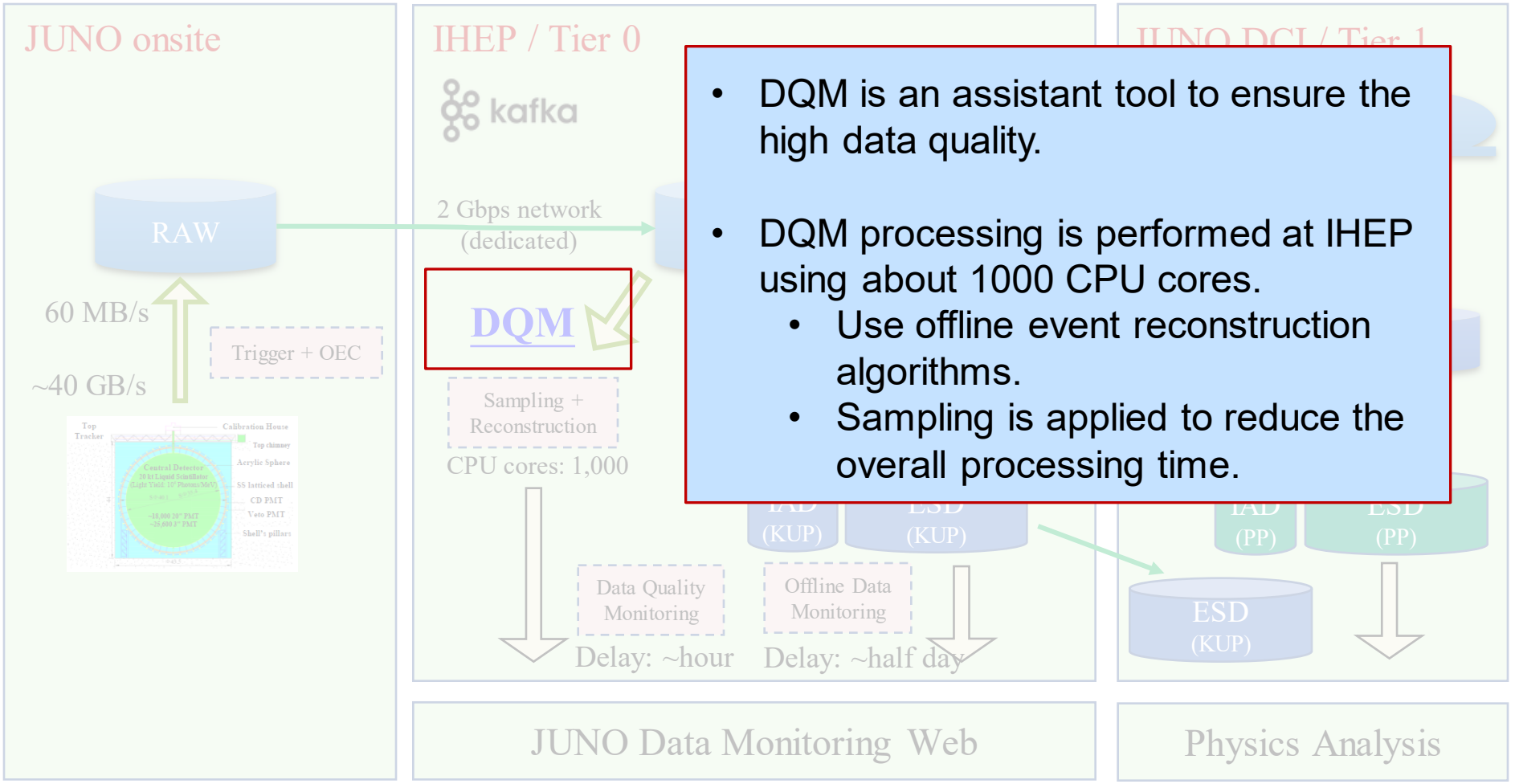


Storage at Tier-0



- Hierarchical storage architecture using both disk and tape systems
- EOS (CERN EOS Open Storage)
  - Providing high-performance disk storage.
- CTA (CERN Tape Archive)
  - Archiving RAW data in the tape library.

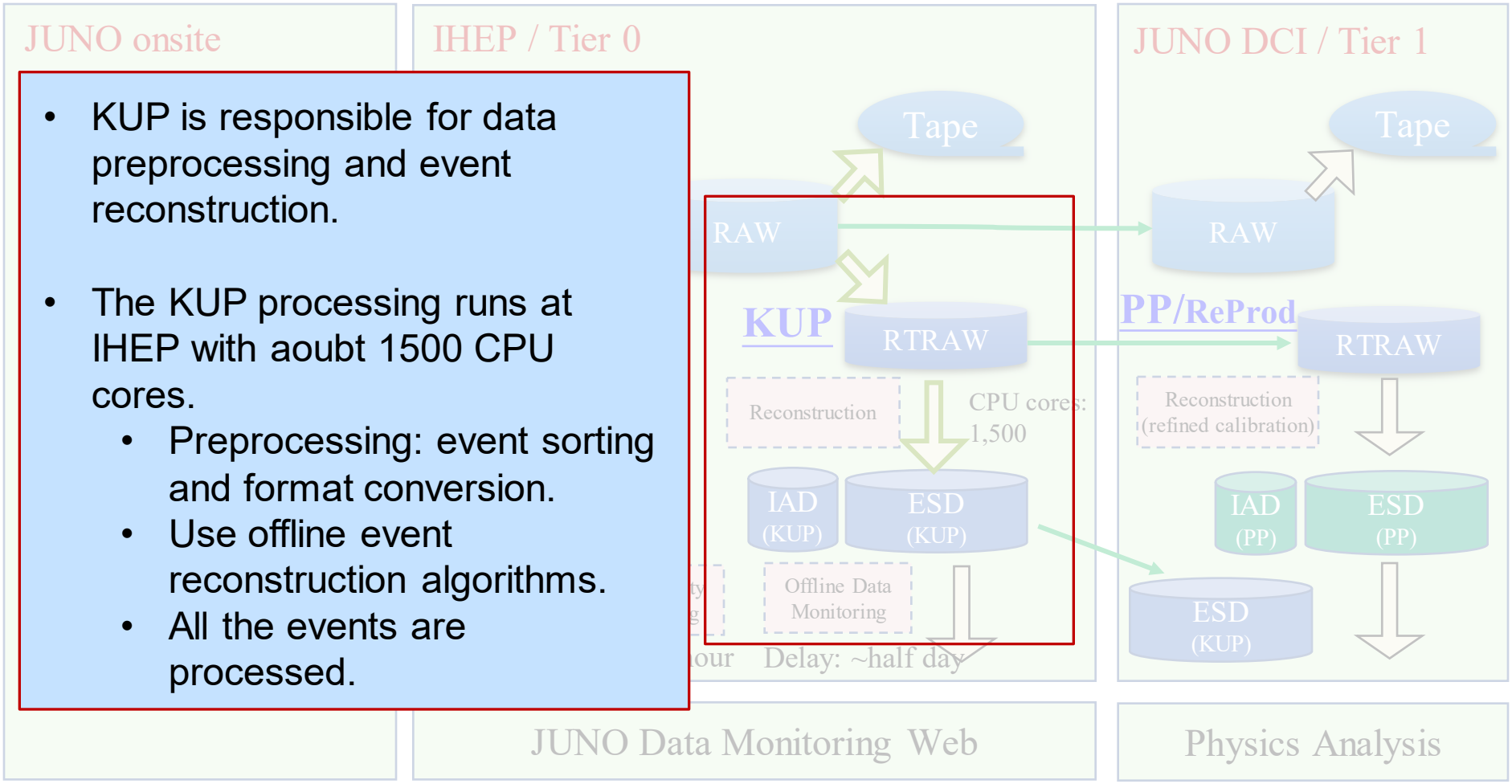
Data Quality Monitoring



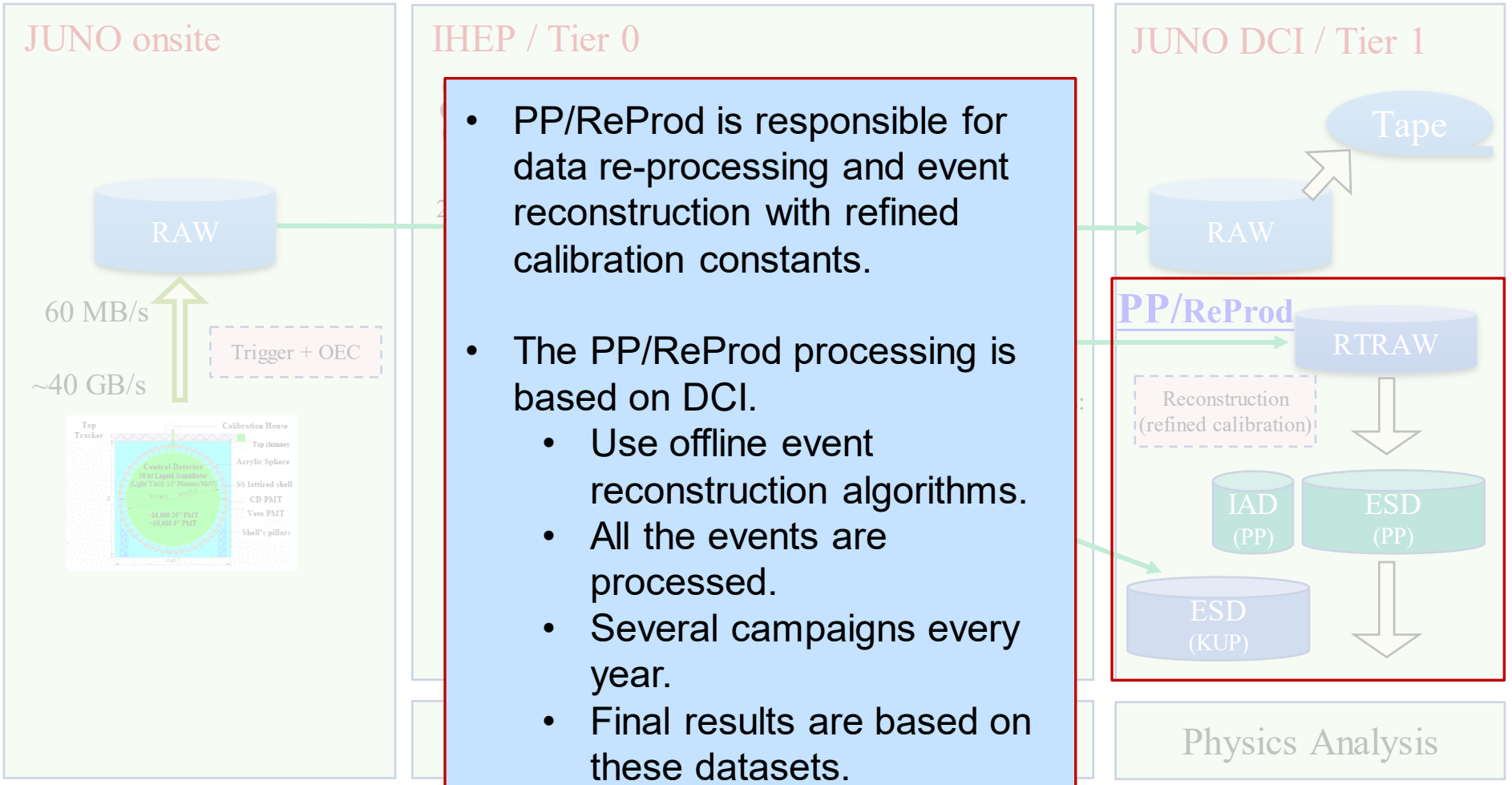
- DQM is an assistant tool to ensure the high data quality.
- DQM processing is performed at IHEP using about 1000 CPU cores.
  - Use offline event reconstruction algorithms.
  - Sampling is applied to reduce the overall processing time.

Keep Up Production

- KUP is responsible for data preprocessing and event reconstruction.
- The KUP processing runs at IHEP with aoubt 1500 CPU cores.
  - Preprocessing: event sorting and format conversion.
  - Use offline event reconstruction algorithms.
  - All the events are processed.



Physics Production



- PP/ReProd is responsible for data re-processing and event reconstruction with refined calibration constants.
- The PP/ReProd processing is based on DCI.
  - Use offline event reconstruction algorithms.
  - All the events are processed.
  - Several campaigns every year.
  - Final results are based on these datasets.

Outline

- Overview of JUNO data processing
- Design of Keep Up Production pipeline
- Micro-service based KUP software
- Application in JUNO and JUNO-TAO
- Summary

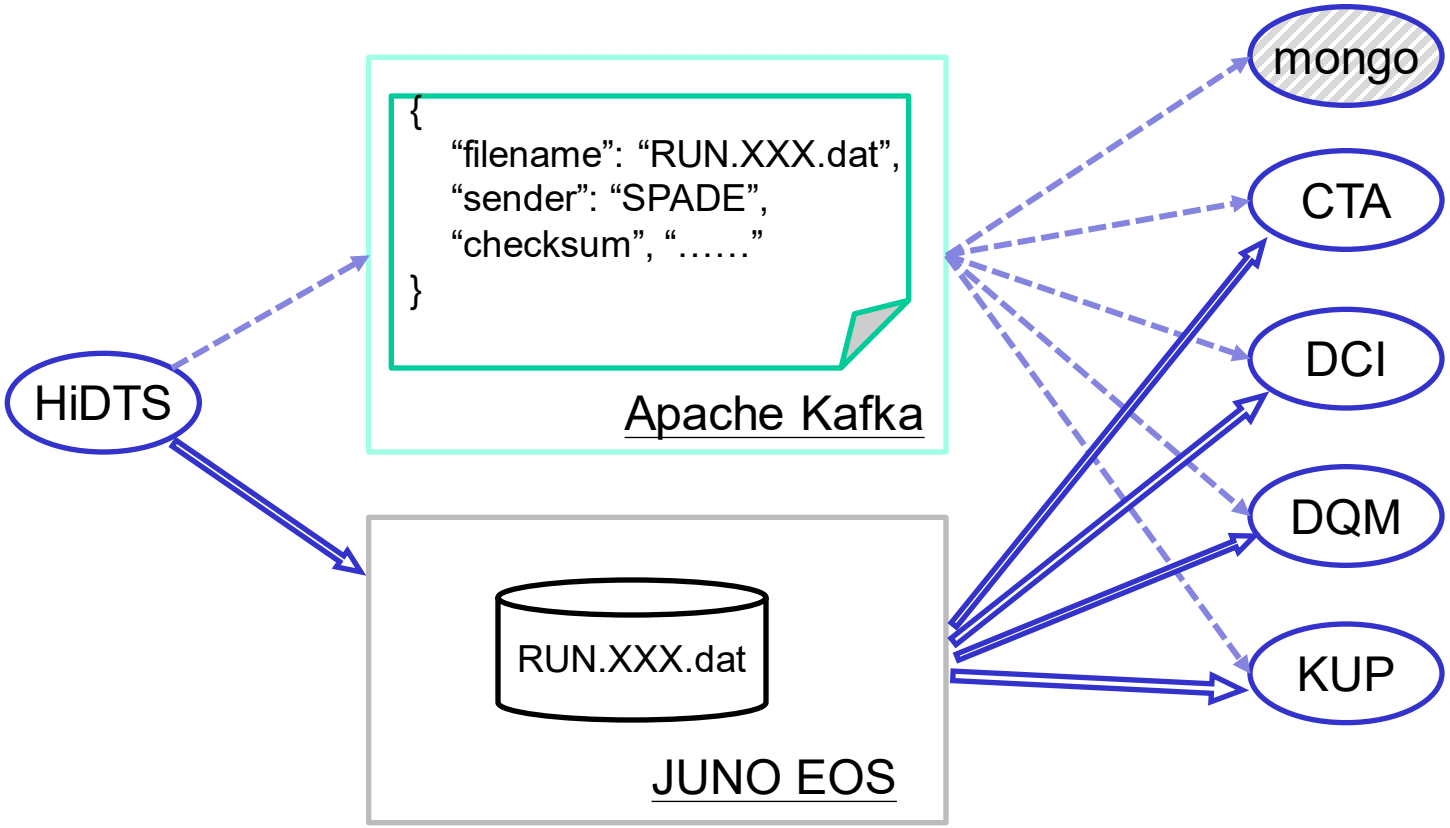
Motivation and requirements

- Motivation
  - Data volume: ~2 PB per year (operate for 20-30 years).
  - Complex processing workflow: offline data processing consists of multiple stages with dependencies.
  - Operational efficiency: Continuous data taking requires reliable and automated processing
- Requirements
  - Stability: Long-term reliable operation
  - Scalable: Ability to handle increasing data volume
  - Automation: Automatic workflow management
  - Monitoring: Failure detection and automatic resubmission

Design Concept

- Event-driven workflow architecture
- Apache Kafka message queue
  - Enables asynchronous communication between processing components
  - Decouples different workflow stages
  - Provides reliable message buffering and delivery
- MongoDB backend database
  - Stores workflow states, task metadata, and processing records
  - Supports efficient querying and system monitoring
- Automated workflow management for preprocessing, reconstruction and analysis.

System Design



- A file is created in EOS disk storage, then a message is sent to message queue to trigger others.
- A unified message schema is adopted across all subsystems in the pipelines.

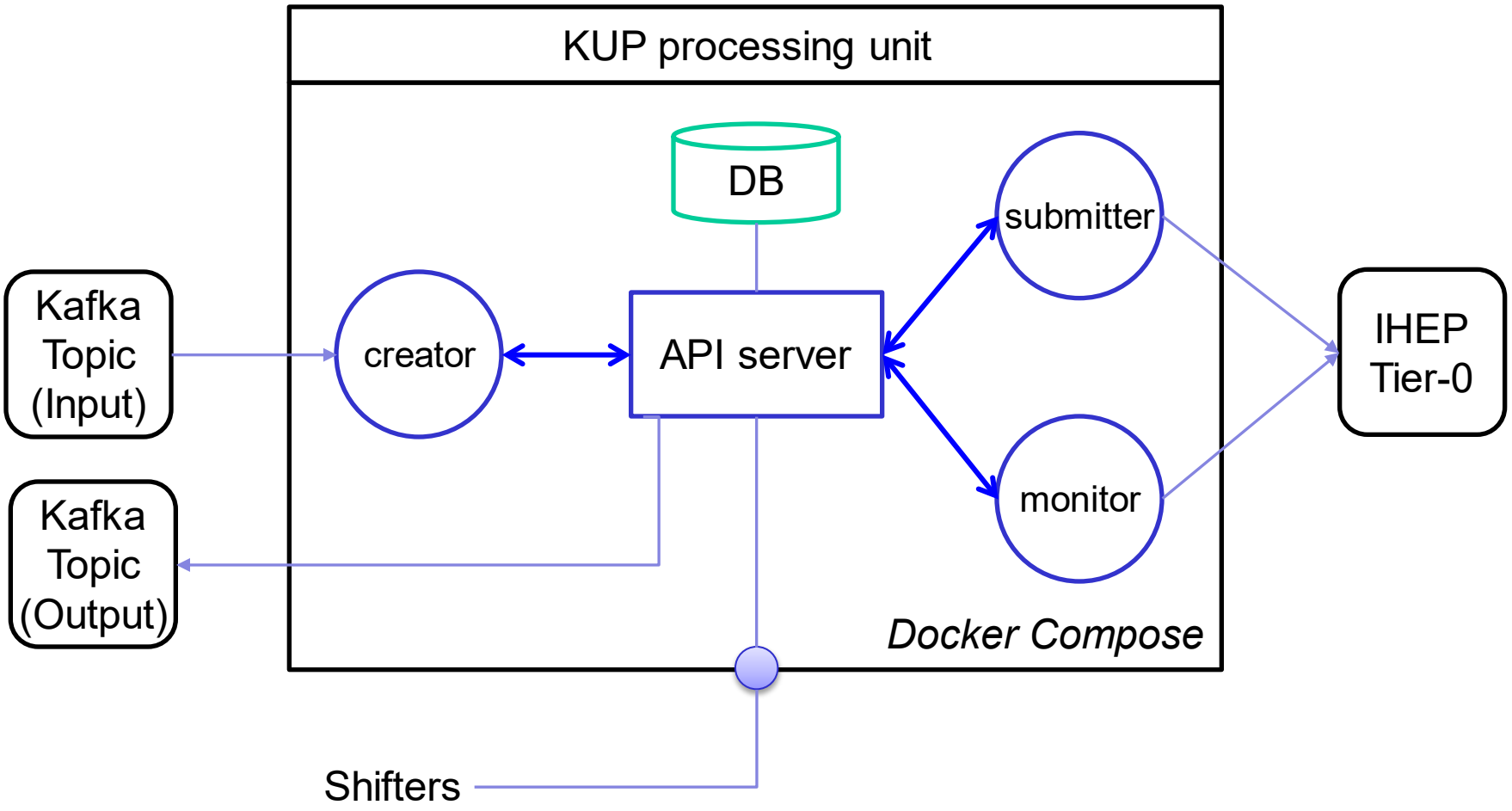
Outline

- Overview of JUNO data processing
- Design of Keep Up Production pipeline
- Micro-service based KUP software
- Application in JUNO and JUNO-TAO
- Summary

Micro-service based KUP software

- The KUP pipeline consists of multiple processing tasks
  - index generation, event sorting, data format conversion, event reconstruction and analysis.
  - More tasks could be added according to the requirements
- Each processing step could be implemented as a standardized processing unit.
- Configuration-driven execution
  - Different pipeline stages share the same execution framework
  - Task behavior is controlled through the configuration.
- Containerized deployment
  - These units are managed and deployed using Docker.

Architecture



API server

- The API server is the central coordination component, providing RESTful APIs for all services.
- It manages workflow state and task metadata.
- Main responsibilities
  - Provide YAML based configuration for each run.
  - Manage the tasks within a run.
  - Handle task failures and resubmission: when a task fails, a new job is created for reprocessing.
  - Track job status on the computing cluster.
  - It is exposed to shifters, allowing them to manage the run configurations and trigger reprocessing of failed tasks.

Creator service

- The creator service listens to messages from the Kafka topic and create new processing tasks.
  - Extended to support event sorting and batch processing.
- Workflow
  - A message arrives and is parsed to extract metadata, such as run, stream, tag, sequence number, date and time.
  - Apply filters and skip the message if necessary.
  - Query the YAML-based configuration according to the run number from API server.
  - Generate a processing script based on the configuration and input metadata.
  - Register the task information with the API server.

Submitter service

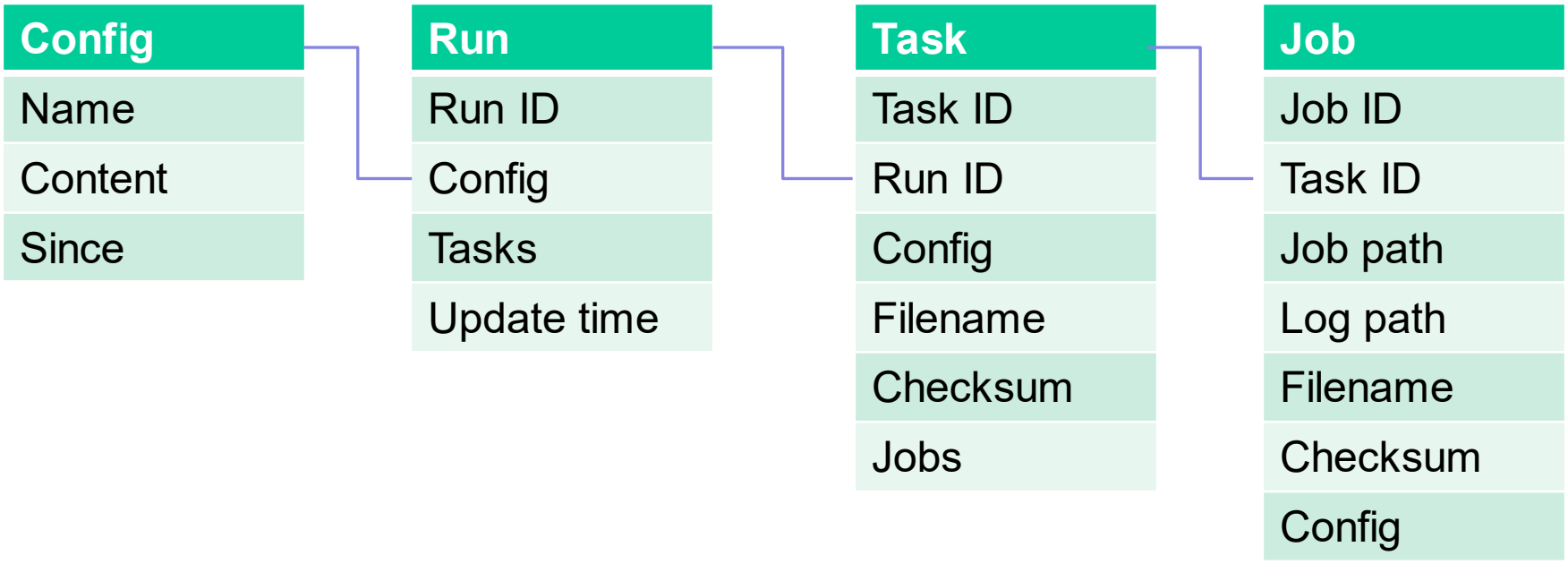
- The submitter service is responsible for job submission.
- Workflow
  - Retrieve a pending task from the API server.
  - Create a new job and update the job submission counter in task.
  - Submit the processing job to the IHEP Tier-0 via SSH.
  - Retrieve the job ID and update the job status.
  - If the job submission is failed, the service automatically retries.

Monitor service

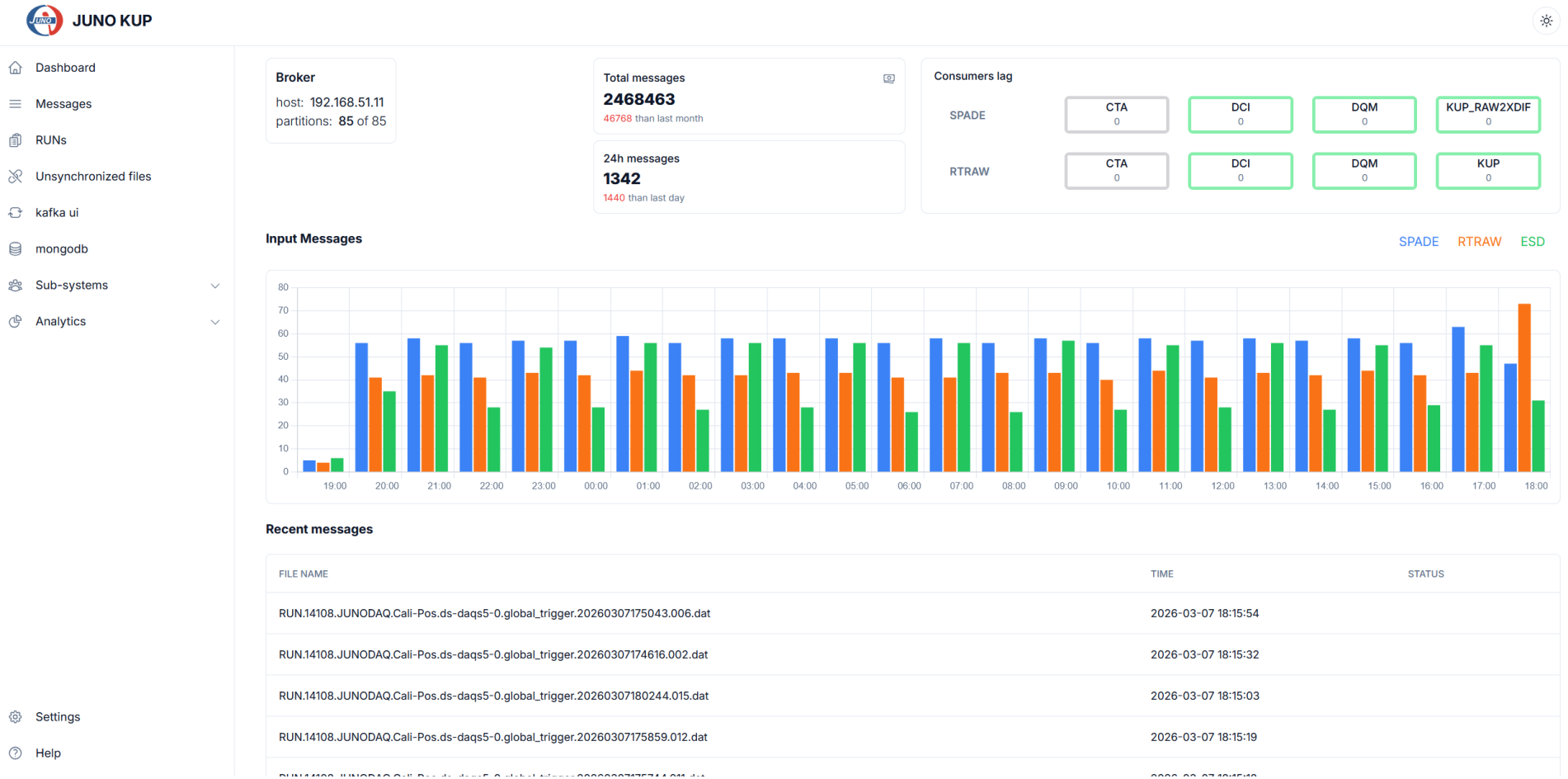
- This service tracks the execution status of jobs.
- Workflow
  - Check job status on IHEP Tier-0 via SSH.
  - If the job ID is still idle or running, skip further checks.
  - Verify whether the stdout and stderr log files exist on the cluster.
    - If not, the job is marked as failed.
  - Parse the log files to detect possible failures.
    - If a failure is found, the job is marked as failure.
  - If no failure is detected, the job is marked as done.
    - The API server will send a message to Kafka.

Database

- MongoDB is used to store all system information.
- Each processing unit consists of the following four collections.



Web portal


JUNO KUP
Dashboard
Messages
RUNs
Unsynchronized files
kafka ui
mongodb
Sub-systems
Analytics
Settings
Help
Broker
host: 192.168.51.11
partitions: 85 of 85
Total messages
2468463
46768 than last month
24h messages
1342
1440 than last day
Consumers lag
SPADE
CTA 0
DCI 0
DQM 0
KUP_RAW2XDIF 0
RTRAW
CTA 0
DCI 0
DQM 0
KUP 0
Input Messages
SPADE RTRAW ESD
80 70 60 50 40 30 20 10 0
19:00 20:00 21:00 22:00 23:00 00:00 01:00 02:00 03:00 04:00 05:00 06:00 07:00 08:00 09:00 10:00 11:00 12:00 13:00 14:00 15:00 16:00 17:00 18:00
Recent messages
FILE NAME
TIME
STATUS
RUN.14108.JUNODAQ.Cali-Pos.ds-daqs5-0.global_trigger.20260307175043.006.dat
2026-03-07 18:15:54
RUN.14108.JUNODAQ.Cali-Pos.ds-daqs5-0.global_trigger.20260307174616.002.dat
2026-03-07 18:15:32
RUN.14108.JUNODAQ.Cali-Pos.ds-daqs5-0.global_trigger.20260307180244.015.dat
2026-03-07 18:15:03
RUN.14108.JUNODAQ.Cali-Pos.ds-daqs5-0.global_trigger.20260307175859.012.dat
2026-03-07 18:15:19

Web portal: message monitoring

Recent messages

Topic RTRAW

FILENAME	FILEPATH	SENDER	TIME	CHECKSUM
RUN.14108.JUNODAQ.Cali-Pos.ds-daqs5-0.global_trigger.20260307180455.017_J25.7.1.rtraw	root://junoeos01.ihep.ac.cn//eos/juno/juno-rtraw/J25.7.1/global_trigger/00014000/00014100/14108/	KUP	2026-03-07 18:26:16	958ad099
RUN.14108.JUNODAQ.Cali-Pos.ds-daqs5-0.global_trigger.20260307180244.015_J25.7.1.rtraw	root://junoeos01.ihep.ac.cn//eos/juno/juno-rtraw/J25.7.1/global_trigger/00014000/00014100/14108/	KUP	2026-03-07 18:25:15	3f8c8220
RUN.14108.JUNODAQ.Cali-Pos.ds-daqs5-0.global_trigger.20260307180349.016_J25.7.1.rtraw	root://junoeos01.ihep.ac.cn//eos/juno/juno-rtraw/J25.7.1/global_trigger/00014000/00014100/14108/	KUP	2026-03-07 18:24:45	91664ab4
RUN.14108.JUNODAQ.Cali-Pos.ds-daqs5-0.global_trigger.20260307174459.001_J25.7.1.rtraw	root://junoeos01.ihep.ac.cn//eos/juno/juno-rtraw/J25.7.1/global_trigger/00014000/00014100/14108/	KUP	2026-03-07 18:24:45	d747a518
RUN.14108.JUNODAQ.Cali-Pos.ds-daqs5-0.global_trigger.20260307180143.014_J25.7.1.rtraw	root://junoeos01.ihep.ac.cn//eos/juno/juno-rtraw/J25.7.1/global_trigger/00014000/00014100/14108/	KUP	2026-03-07 18:24:14	32505d02
RUN.14108.JUNODAQ.Cali-Pos.ds-daqs5-0.global_trigger.20260307180622.018_J25.7.1.rtraw	root://junoeos01.ihep.ac.cn//eos/juno/juno-rtraw/J25.7.1/global_trigger/00014000/00014100/14108/	KUP	2026-03-07 18:23:44	2d33e580
RUN.14108.JUNODAQ.Cali-Pos.ds-daqs5-0.global_trigger.20260307175859.012_J25.7.1.rtraw	root://junoeos01.ihep.ac.cn//eos/juno/juno-rtraw/J25.7.1/global_trigger/00014000/00014100/14108/	KUP	2026-03-07 18:23:44	6af93f9f
RUN.14108.JUNODAQ.Cali-Pos.ds-daqs5-0.global_trigger.20260307175457.009_J25.7.1.rtraw	root://junoeos01.ihep.ac.cn//eos/juno/juno-rtraw/J25.7.1/global_trigger/00014000/00014100/14108/	KUP	2026-03-07 18:23:44	7f9b5db3
RUN.14108.JUNODAQ.Cali-Pos.ds-daqs5-0.global_trigger.20260307175335.008_J25.7.1.rtraw	root://junoeos01.ihep.ac.cn//eos/juno/juno-rtraw/J25.7.1/global_trigger/00014000/00014100/14108/	KUP	2026-03-07 18:23:44	0db53658
RUN.14108.JUNODAQ.Cali-Pos.ds-daqs5-0.global_trigger.20260307175209.007_J25.7.1.rtraw	root://junoeos01.ihep.ac.cn//eos/juno/juno-rtraw/J25.7.1/global_trigger/00014000/00014100/14108/	KUP	2026-03-07 18:23:44	50fc6e16

Web portal: data processing monitoring

Run List

rtraw2esd | Start Run | End Run

RUN 14108 — 0 — 19 — 0

14108 — Collapse

filename	input_checksum	update_time	status
RUN.14108.JUNODAQ.Cali-Pos.ds-daqs5-0.global_trigger.20260307180014.013_J25.7.1.rtraw	568913a6	2026-03-07T18:30:12.922000	
RUN.14108.JUNODAQ.Cali-Pos.ds-daqs5-0.global_trigger.20260307175625.010_J25.7.1.rtraw	898c0e4c	2026-03-07T18:28:40.504000	
RUN.14108.JUNODAQ.Cali-Pos.ds-daqs5-0.global_trigger.20260307174701.003_J25.7.1.rtraw	bb3dc959	2026-03-07T18:31:44.010000	
RUN.14108.JUNODAQ.Cali-Pos.ds-daqs5-0.global_trigger.20260307175744.011_J25.7.1.rtraw	604af7f2	2026-03-07T18:28:40.664000	
RUN.14108.JUNODAQ.Cali-Pos.ds-daqs5-0.global_trigger.20260307174754.004_J25.7.1.rtraw	44dae67a	2026-03-07T18:29:11.566000	
RUN.14108.JUNODAQ.Cali-Pos.ds-daqs5-0.global_trigger.20260307174616.002_J25.7.1.rtraw	ffc8eb55	2026-03-07T18:33:15.209000	
RUN.14108.JUNODAQ.Cali-Pos.ds-daqs5-0.global_trigger.20260307180455.017_J25.7.1.rtraw	958ad099	2026-03-07T18:31:13.662000	
RUN.14108.JUNODAQ.Cali-Pos.ds-daqs5-0.global_trigger.20260307180143.014_J25.7.1.rtraw	32505d02	2026-03-07T18:32:44.655000	

Web portal: unprocessed file monitoring

Unsynchronized Data

Stream: global_trigger

RUN	STREAM	SEQ	DATA FILE	RAW	RTRAW	ESD	CREATE TIME	UPDATE TIME
▶ Run 11833 (22 files)								
▼ Run 13893 (1 files)								
13893	global_trigger	001	RUN.13893.JUNODAQ.Calib-ACU-Cs137-Global-Moving.ds-daqs5-0.global_trigger.20260227165025.001.dat	✔	✗	✗	2026-02-27 16:52:07	2026-02-27 16:52:07
▶ Run 13920 (16 files)								
▶ Run 13921 (1 files)								
▶ Run 13922 (13 files)								
▶ Run 13923 (1 files)								
▶ Run 13924 (13 files)								
▶ Run 13925 (1 files)								
▶ Run 13926 (28 files)								
▶ Run 13927 (1 files)								
▶ Run 13928 (17 files)								
▶ Run 13929 (1 files)								
▶ Run 13930 (8 files)								
▶ Run 13931 (1 files)								

Outline

- Overview of JUNO data processing
- Design of Keep Up Production pipeline
- Micro-service based KUP software
- Application in JUNO and JUNO-TAO
- Summary

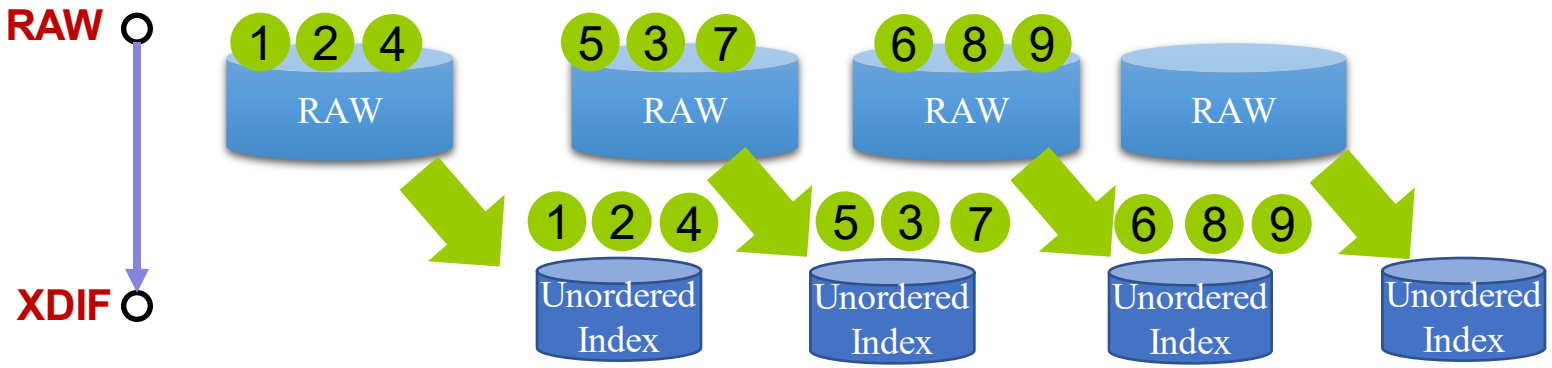
Pre-processing with event sorting

RAW



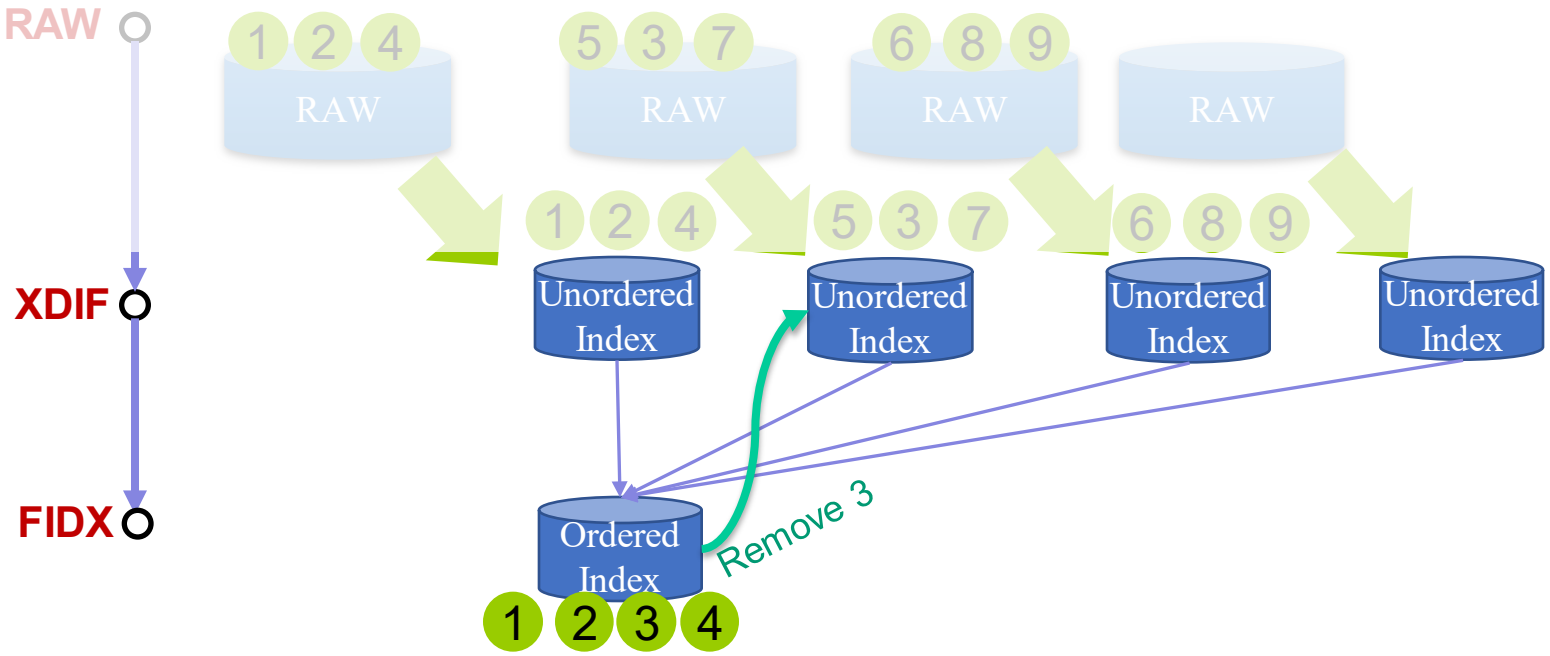
- Events in a RAW file must be sorted using data from the current file and the subsequent files.
- The sorting step could not be parallelized.
- The entire workflow needs to be implemented in KUP.

Pre-processing with event sorting



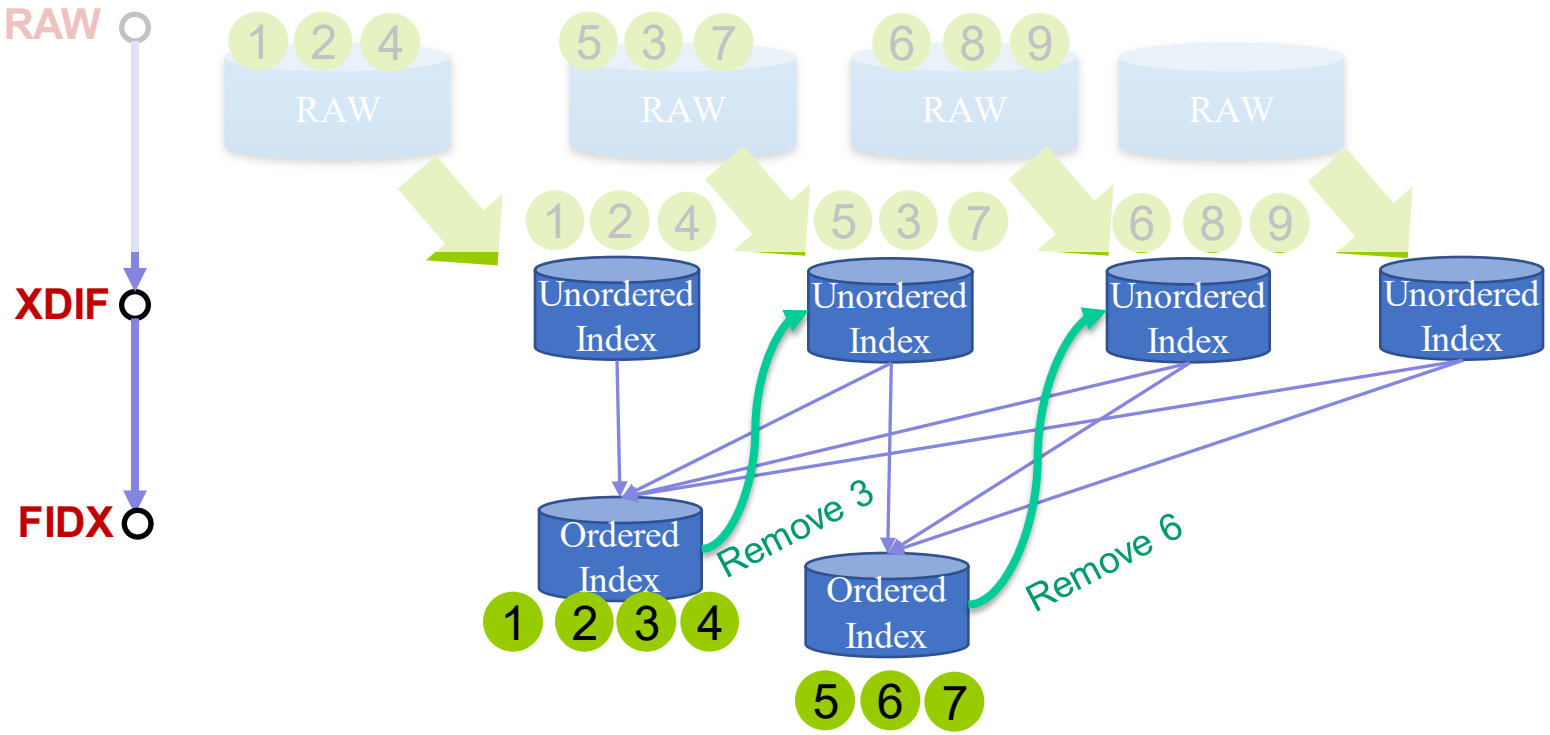
- The RAW file is first uncompressed, and an unordered index is created.
- Each index entry refers to a file and an offset within file.
- Flags in unordered index are updated after sorting.

Pre-processing with event sorting



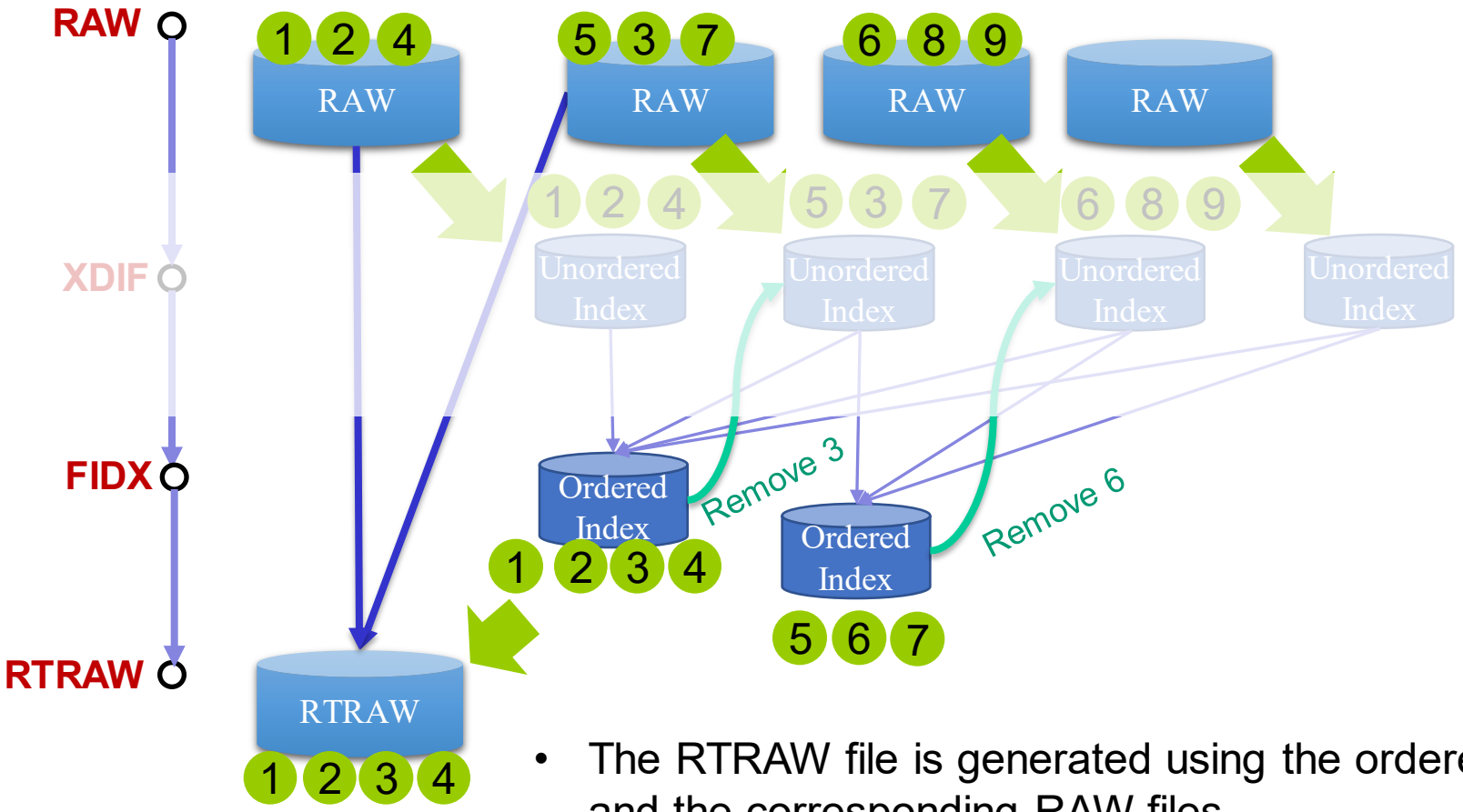
- To produce the ordered index, the current unordered index and the subsequent indices are required.
- After sorting, the corresponding events are marked as used in the unordered indices.

Pre-processing with event sorting



- The next ordered index must be generated after the previous one.

Pre-processing with event sorting

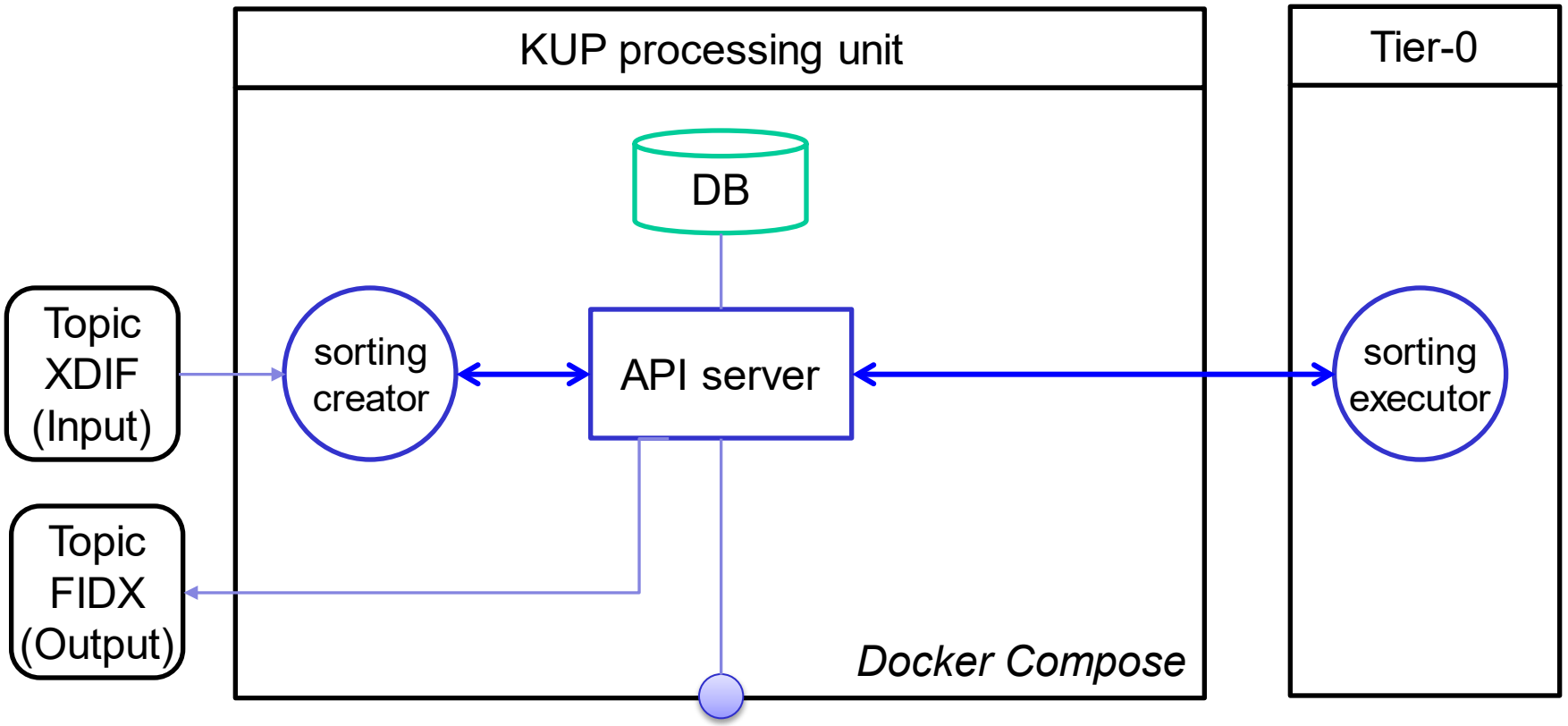


- The RTRAW file is generated using the ordered index and the corresponding RAW files.

Implementation of sorting in pipeline

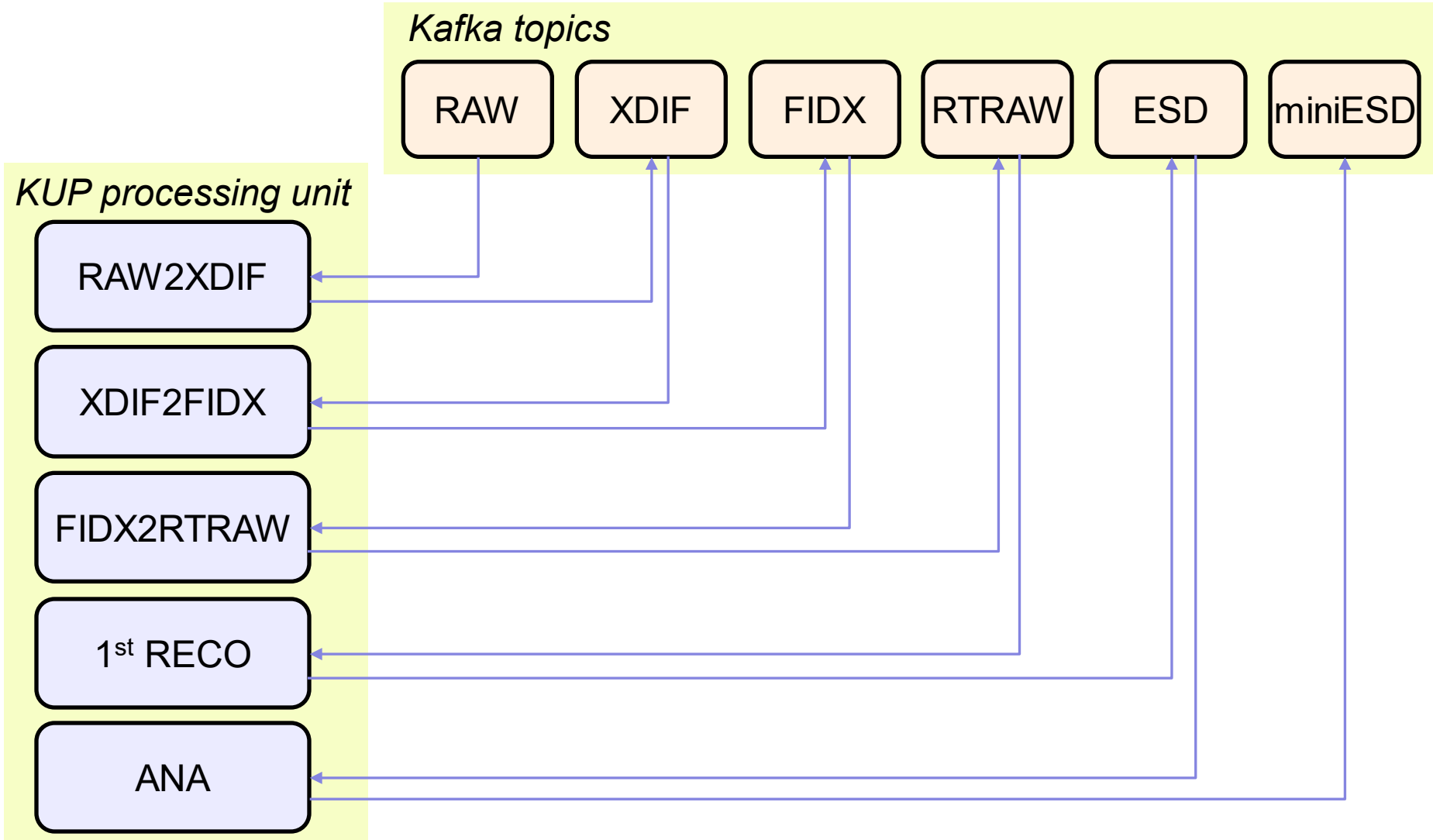
- To reduce latency, the preprocessing stage is split into three steps:
  - RAW2XDIF: generates unordered indices in parallel.
  - XDIF2FIDX: performs **event sorting sequentially**.
  - FIDX2RTRAW: produces RTRAW files in parallel.
- The creator service is extended to support event sorting.
  - Run and file information are cached in a local SQLite database.
  - Sorting requires the continuous of file sequence number.
  - DAQ database is queried to obtain the status of a run and number of files in the run.

Deployment of sorting unit



A dedicated service is used to execute sorting tasks instead of submitting them as jobs.

Final deployment in JUNO (TAO)



Outline

- Overview of JUNO data processing
- Design of Keep Up Production pipeline
- Micro-service based KUP software
- Application in JUNO and JUNO-TAO
- Summary

Summary

- An automated offline data processing system has been developed for JUNO Tier-0.
- The system adopts a microservice architecture, using Apache Kafka for message communication.
- The KUP pipeline supports multiple preprocessing and reconstruction tasks through configuration-driven processing units.
- A cross-file event sorting workflow is implemented to produce ordered event data.
- The system has been in operation since the JUNO commissioning phase.
- It enables scalable, reliable, and automated data processing for JUNO and TAO operations.

Thank you for your attention