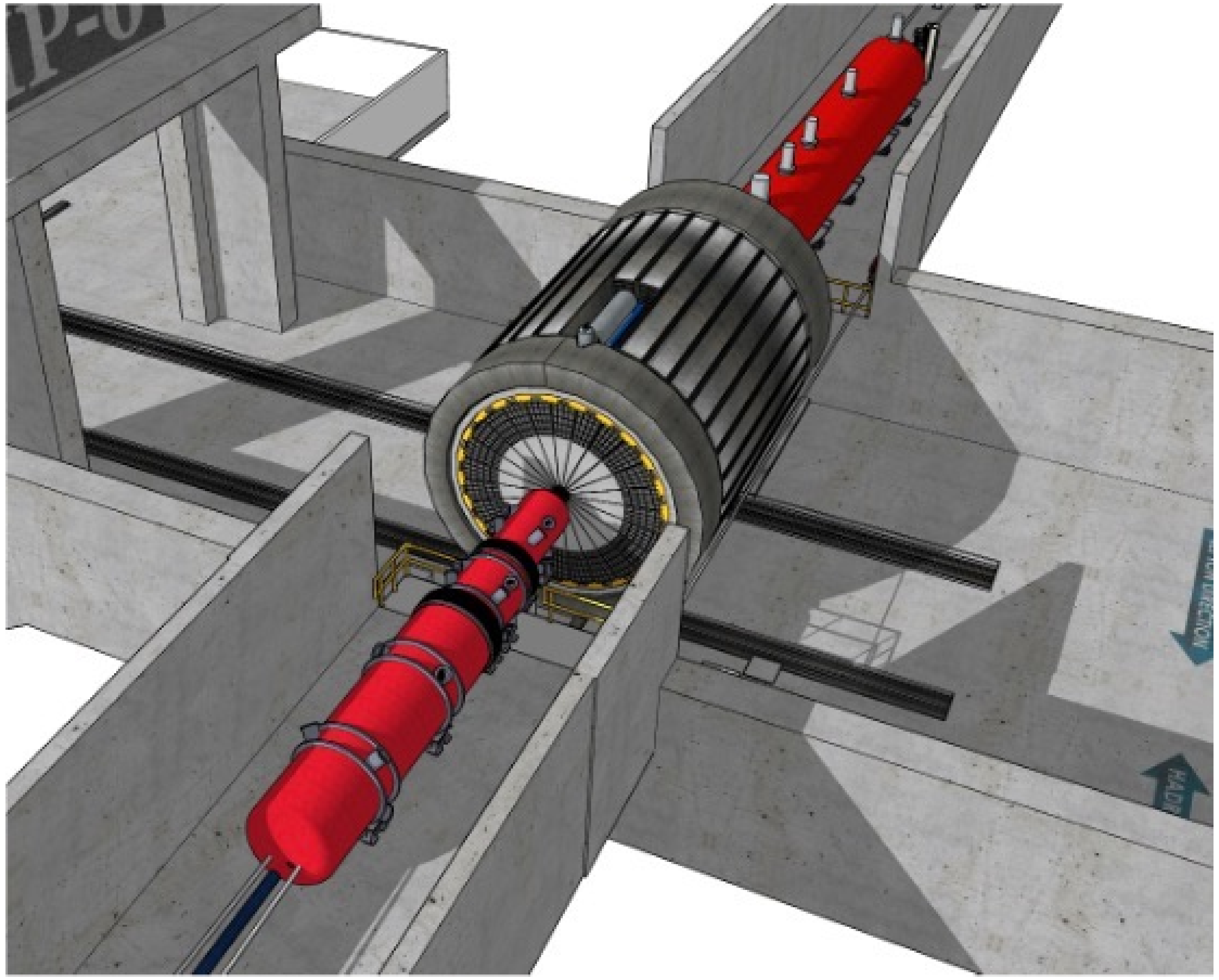
Reconstruction Framework Advancements for Streaming Readout for the ePIC Experiment at the EIC

Torri Jeske, Nathan Brei (nbrei@jlab.org)

CHEP 2024



ePIC is the primary experiment at the Electron-Ion Collider (EIC)

It is a **highly integrated**, **multi-purpose** experiment that is being realized by the **international ePIC collaboration** (formed in 2022-2023) jointly with the EIC Project.

State of the art detectors and computing

ensure that ePIC will deliver on EIC science from the start of operations

Compute-Detector Integration

Seamless data processing from detector readout to analysis using streaming readout and streaming computing. **Provide rapid turnaround** of 2-3 weeks for data **for physics analyses**, with the timeline driven by alignment and calibrations.

Compute-Detector Integration to Accelerate Science

Streaming Readout

Data is read out in continuous parallel streams that are encoded with information about when and where the data was taken

Artificial Intelligence

for autonomous alignment and calibration, validation, and rapid processing

Heterogeneous Computing

for acceleration (CPU, GPU)

Streaming Data Processing: Comparison

Traditional

- Data acquired in online workflows
- Data is stored as large files in hierarchical storage
- offline workflows process data
- Batch queue-based resource provisioning
- Discrete, coarse-grained processing units (files and datasets)
- Decoupling from real-time data acquisition

Streaming

- Quasi-continuous flow of fine-grained data
- Dynamic flexibility to match real-time data inflow
- Prompt processing is crucial for data quality and detector integrity
- Processing full data set quickly to minimize time for detector calibration and deliver analysis ready data

Advantages of Streaming Data Processing

Simplified readout

No custom trigger hardware and firmware

Holistic detector information

Build events with holistic detector information

Continuous data flow

Detailed knowledge of backgrounds and enhanced control of systematics

JANA2

The ePIC collaboration chose **JANA2** as its reconstruction framework. It is a **scalable, modern c++ reconstruction framework** for both **traditional and streaming event processing**. It has evolved substantially in response to ePIC's needs.

Documentation

JANA2 Concepts

JEvent

Container for data that can be processed as a discrete unit, independently from the rest of the stream.

JEventSources

Take a file or messaging producer which provides raw event data and exposes it to JANA2 as a stream

JFactories

Calculate a specific result on an event-by-event basis. Inputs can come from an event source or may be computed via other JFactories.

JEventProcessors

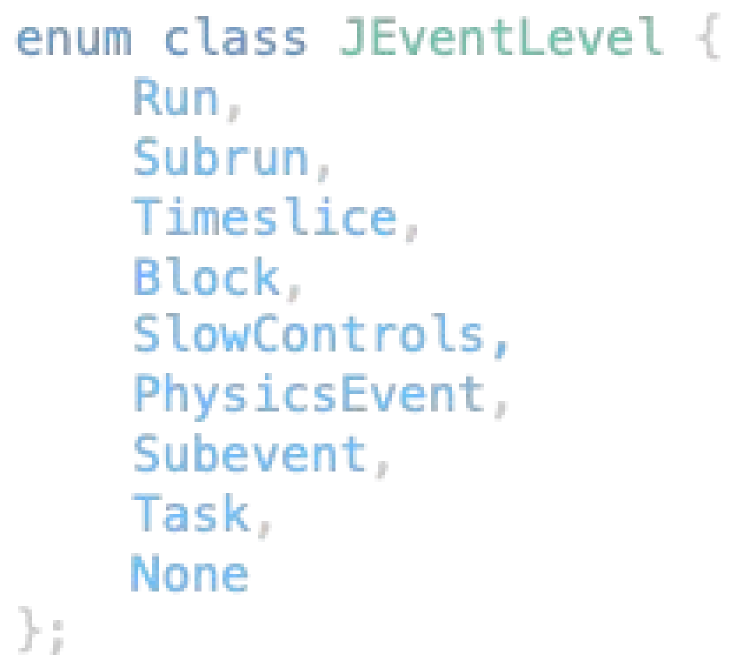
Run desired JFactories over teh event stream and write the results to an output file

JANA2 New Feature

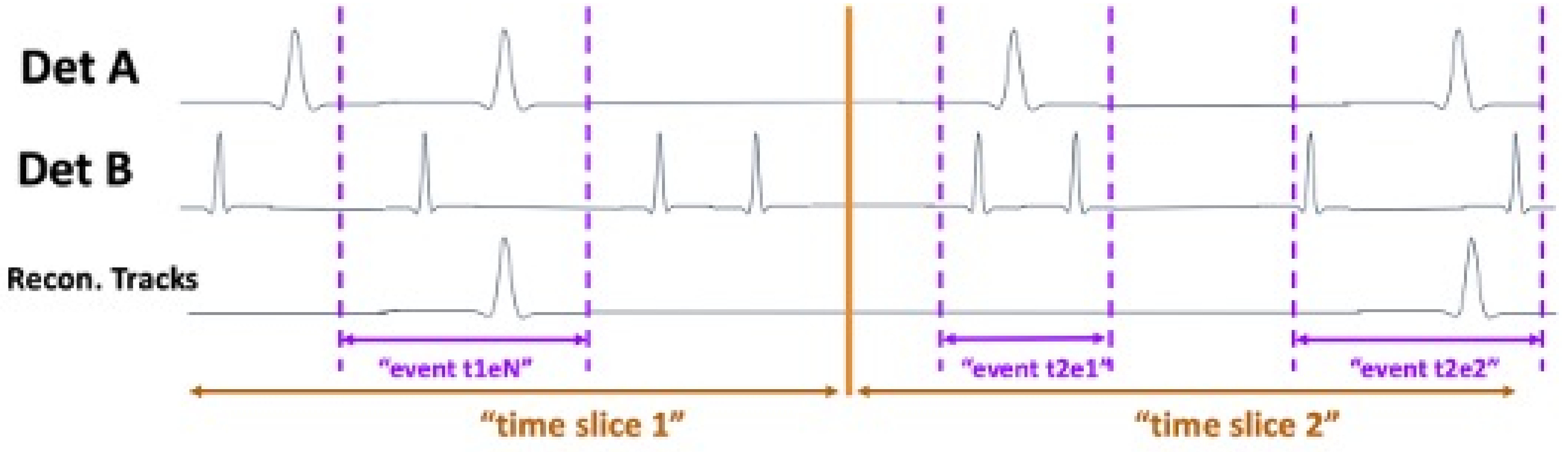
Event levels

Timeslices form the basic unit in a streaming system

JANA2 supports different event levels, where each JEvent is tagged as belonging to some JEventLevel

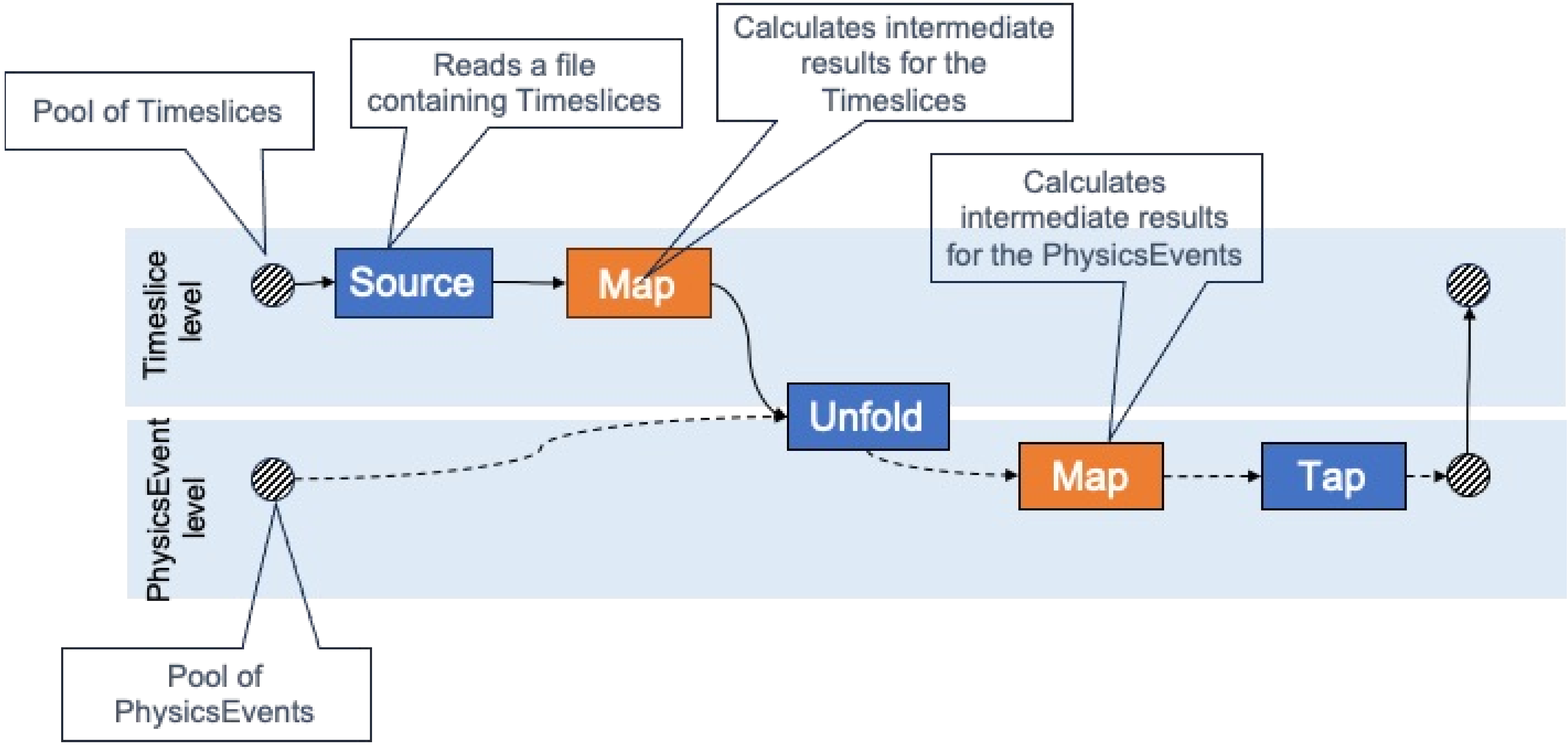


```
enum class JEventLevel {
    Run,
    Subrun,
    Timeslice,
    Block,
    SlowControls,
    PhysicsEvent,
    Subevent,
    Task,
    None
};
```

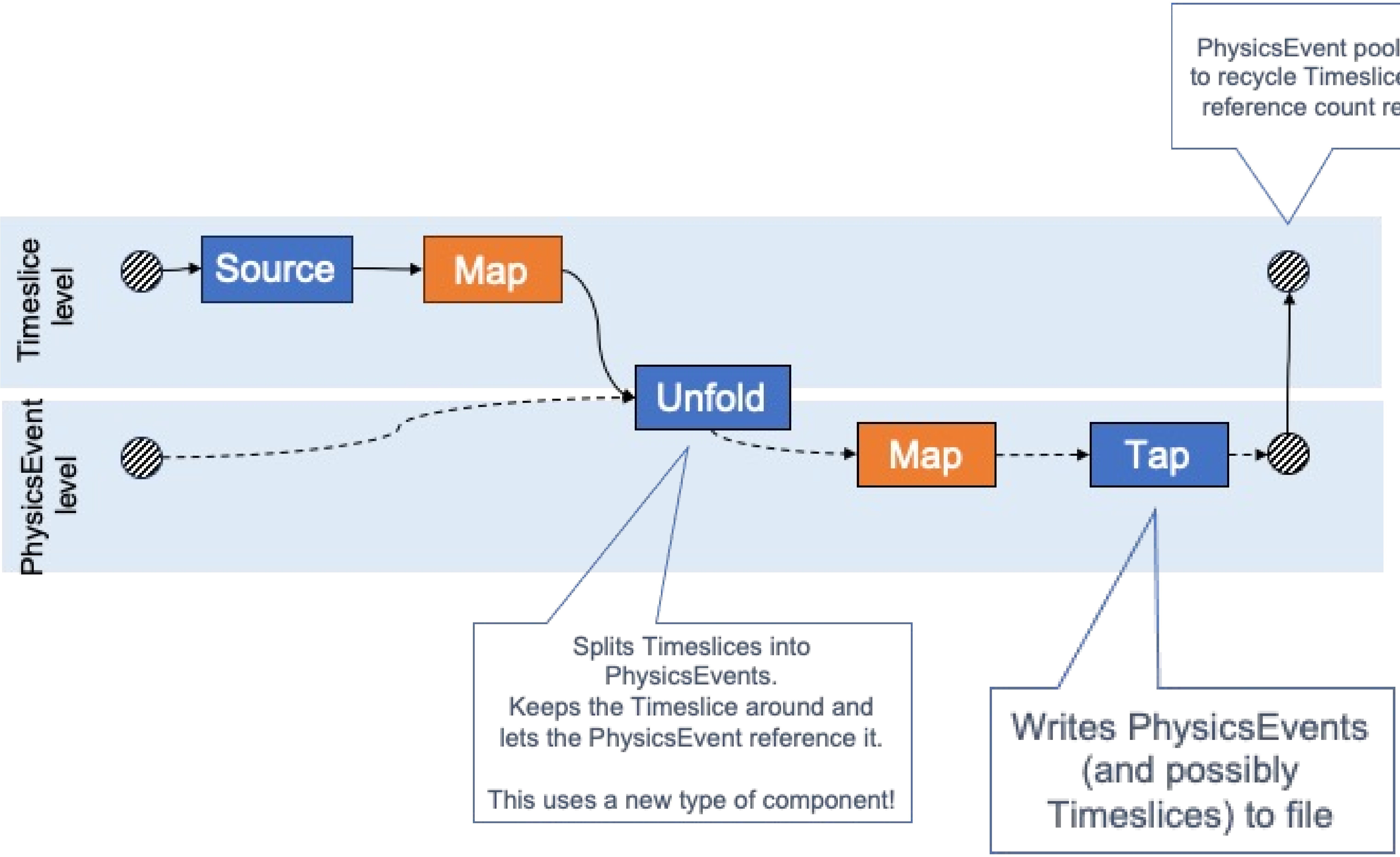


JANA2 should handle these cases symmetrically to the maximum extent possible.

Extracting Physics Events from Timeslices


Calculates intermediate results for the Timeslices
Reads a file containing Timeslices
Pool of Timeslices
Calculates intermediate results for the PhysicsEvents
Timeslice level
Source
Map
Unfold
PhysicsEvent level
Map
Tap
Pool of PhysicsEvents

Extracting Physics Events from Timeslices



PhysicsEvent pool knows how to recycle Timeslices once their reference count reaches zero

Splits Timeslices into PhysicsEvents.
Keeps the Timeslice around and lets the PhysicsEvent reference it.

This uses a new type of component!

Writes PhysicsEvents (and possibly Timeslices) to file

JANA2 New Feature

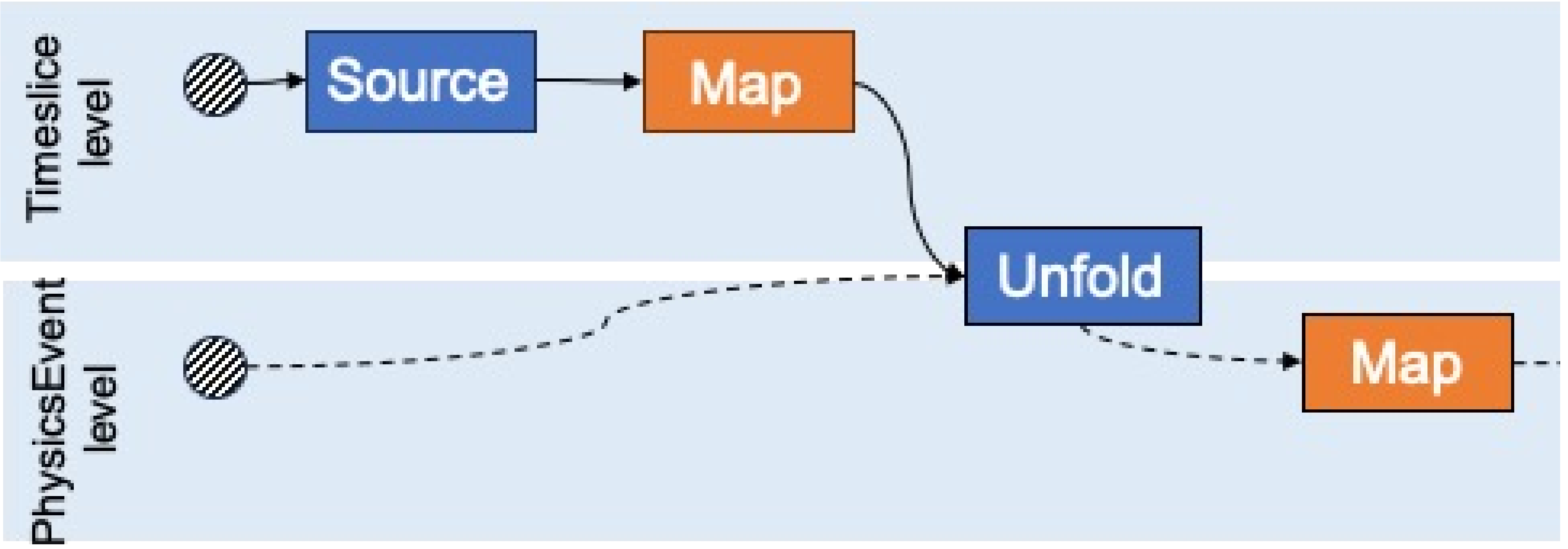
JEventUnfolder

JEventUnfolder is a special class that defines an algorithm for *unfolding**

Inputs come from Timeslice (*parent*)
Outputs are PhysicsEvents (*child*)

This replaces the traditional hardware trigger

*terminology borrowed from functional programming and stream processing

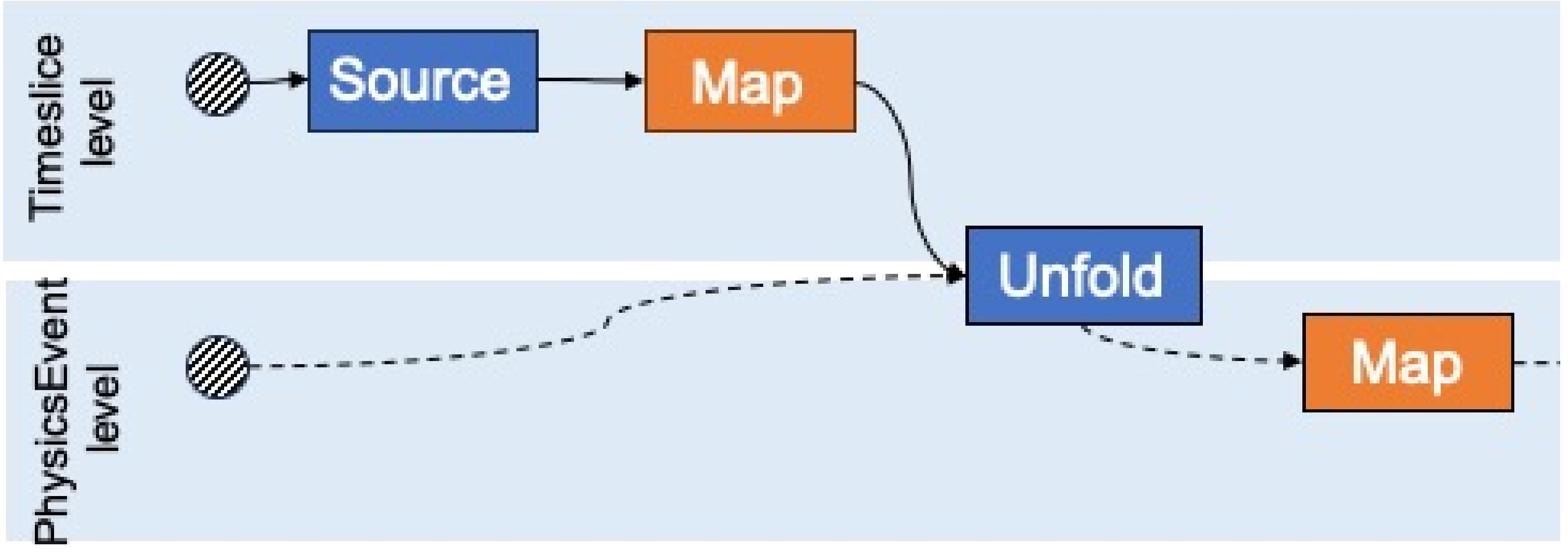


Implications for Factories (algorithms)

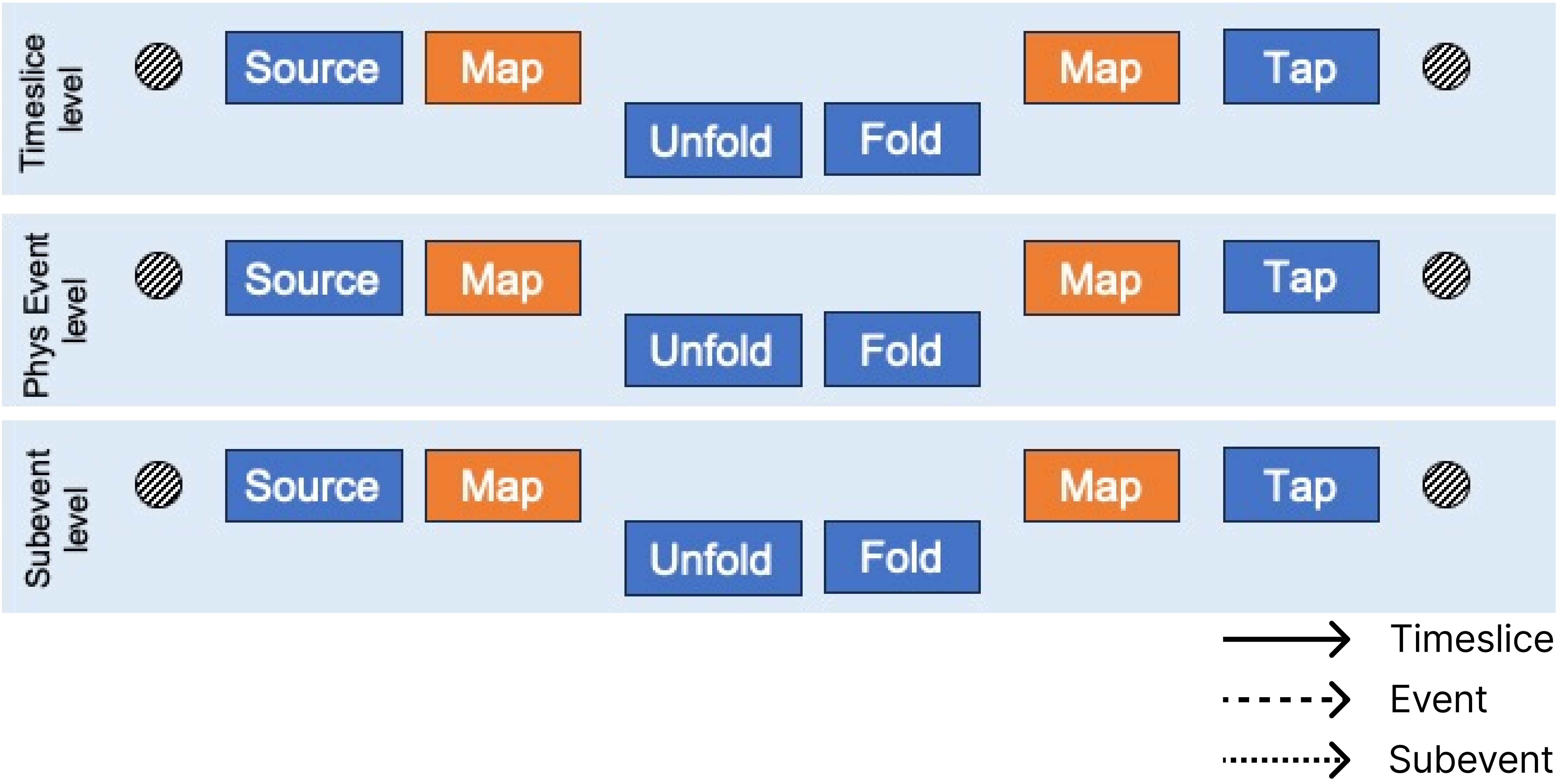
Factories are written for a specific event level

Timeslice-level factories, event-level factories

Event-level factories have access to timeslice-level results



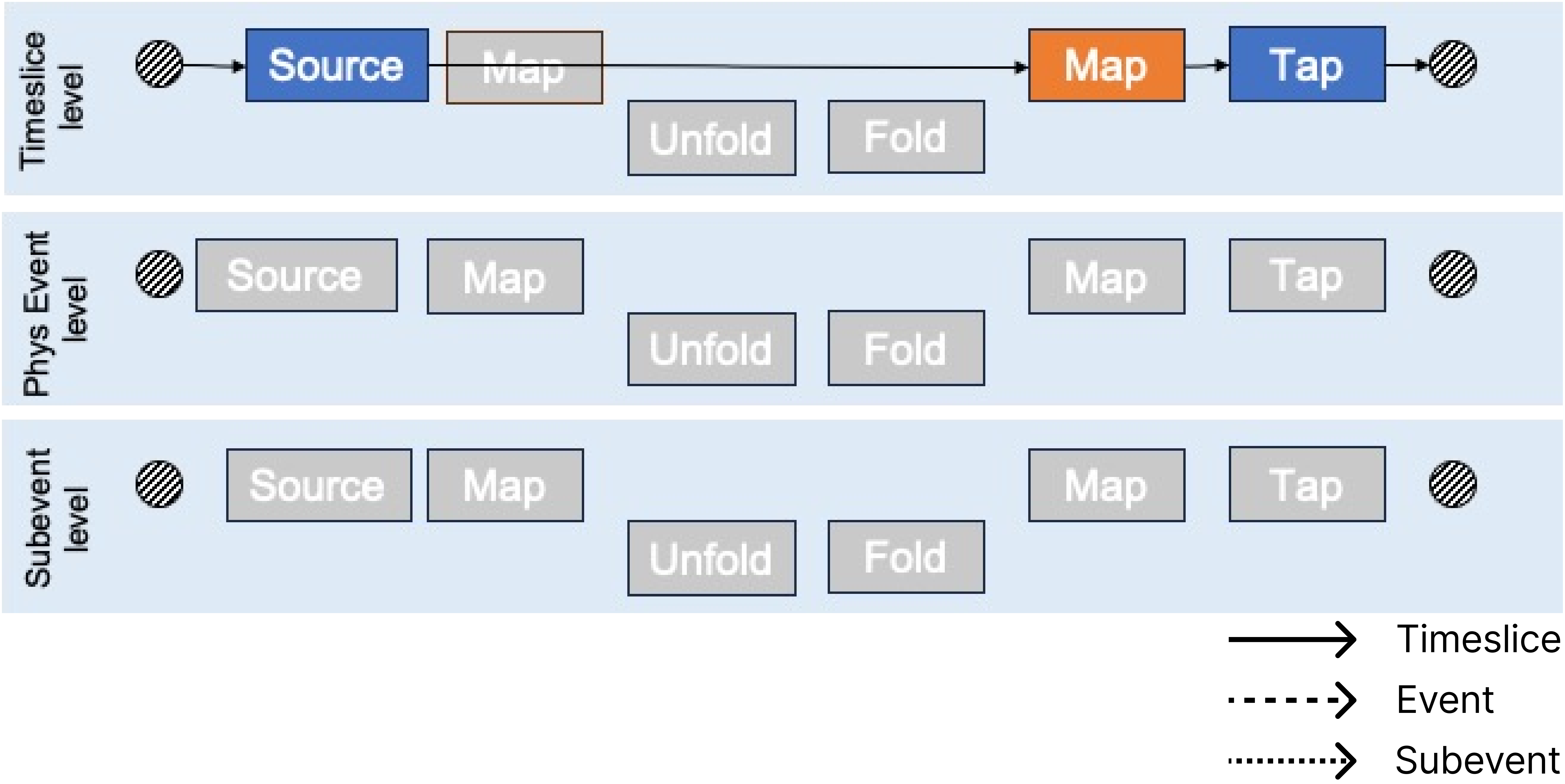
Generalizing further..



JANA2 can “wire” the arrows automatically

arrows abstractly form a grid
wiring can also be specified manually

Basic Topology



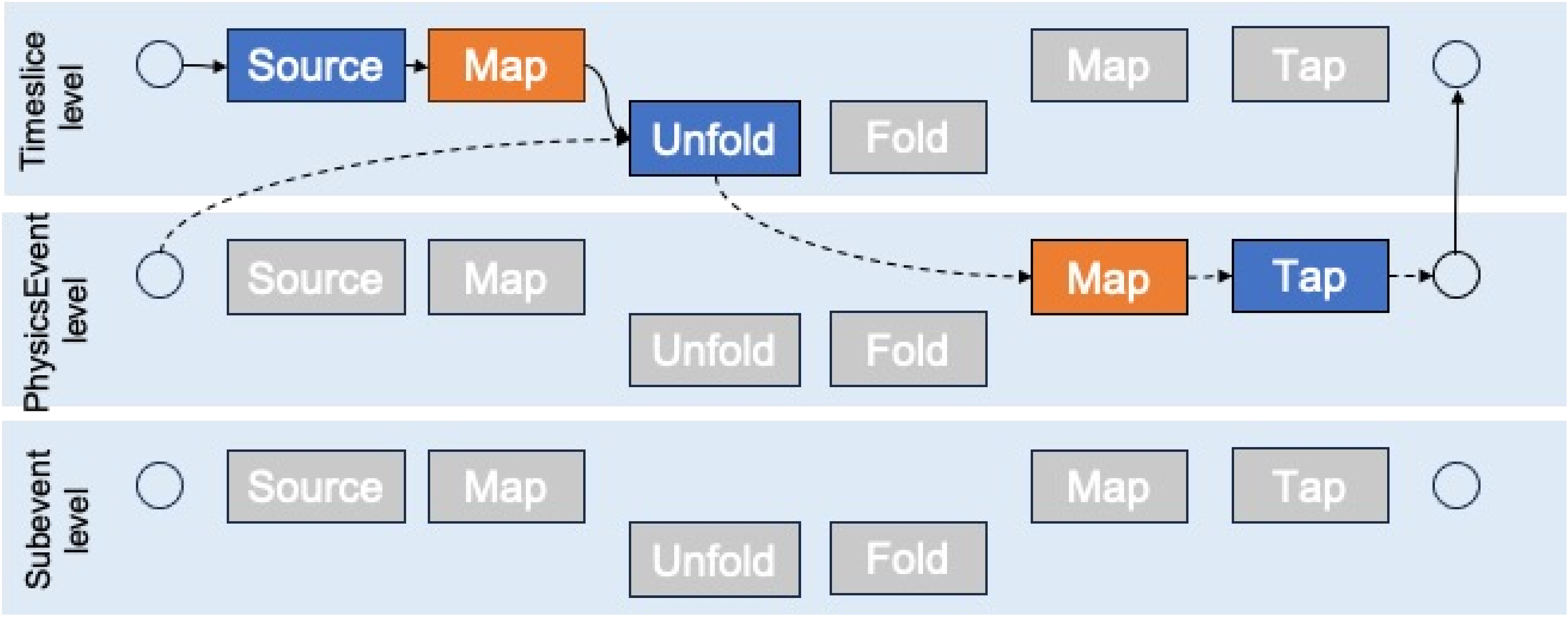
User Provides:

JEventSource (Timeslice)

JEventProcessor

JFactory

Timeslice Splitting Topology



User Provides:

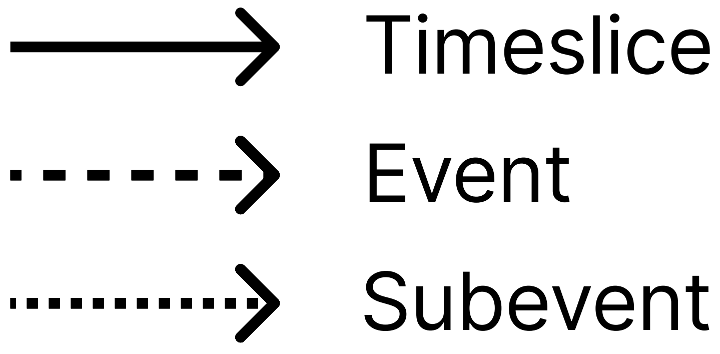
JEventSource (T)

JFactory (T)

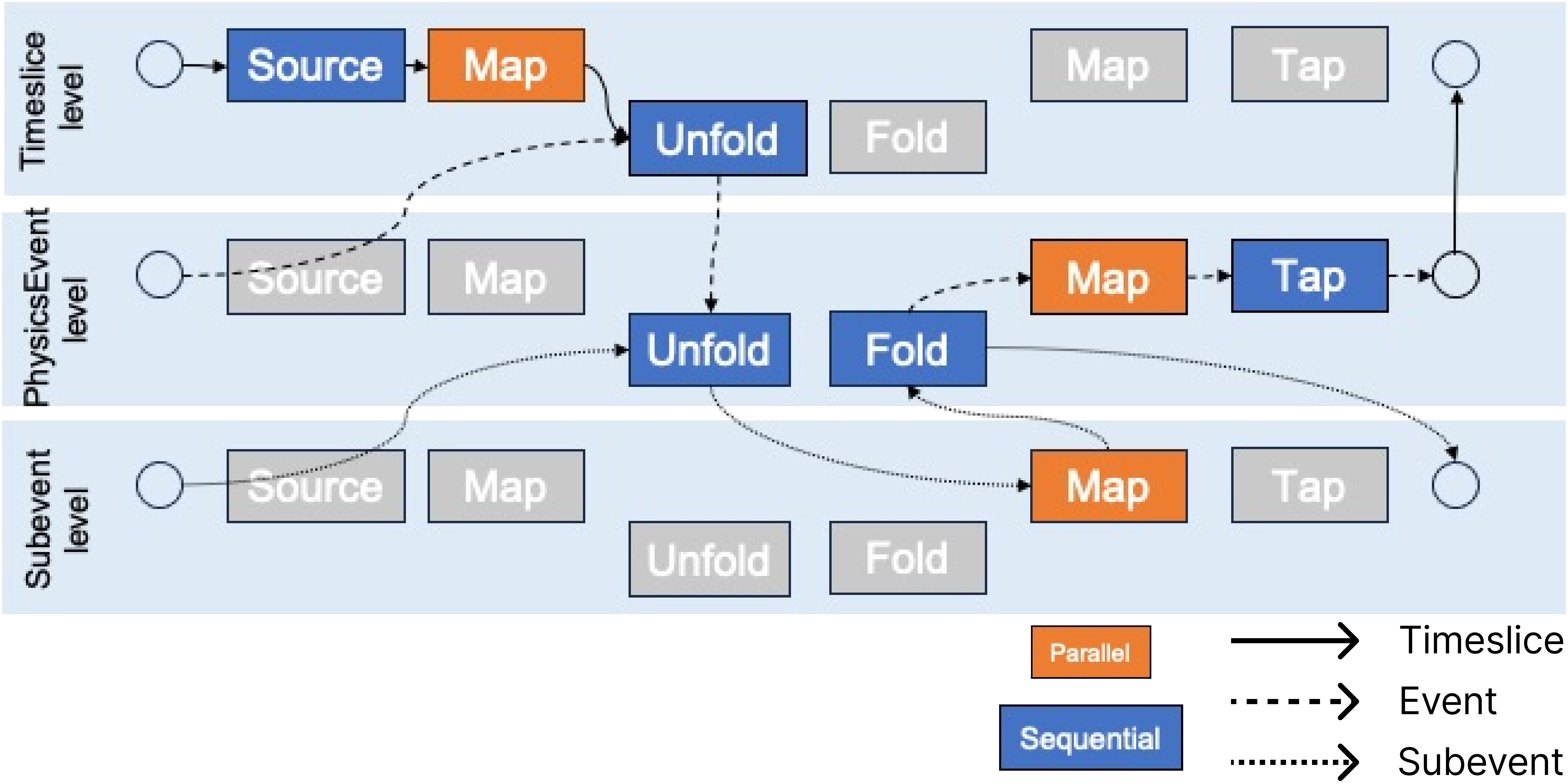
JEventUnfolder (T→ P)

JEventProcessor (P)

JFactory (P)



Timeslice + Subevents Splitting Topology



User Provides:

JEventSource (T)

JEventProcessor (P)

JEventUnfolder (T→ P)

JEventUnfolder (P→S)

JEventFolder (S→P)

JFactory (T)

JFactory (P)

JFactory (S)

Implications for EICrecon

Define many things once

Factories, PODIO event sources, and processors can be defined once.

Same clustering algorithm on timeslices, physics events, etc

Supports multiple topologies

User provides components

Topology generated from user input

Topology is built based on the components that the user defines

Summary

ePIC detector will use streaming readout

This will provide a simplified readout and holistic detector information with detailed knowledge of backgrounds and enhanced control of systematics

JANA2 was chosen as the reconstruction framework

Actively developed to support features needed for streaming readout

Introduced two new component abstractions: Folder and Unfolder

Users can write an algorithm once and configure it at runtime to operate on timeslices or physics events