Adaptable data-processing for HEP computing frameworks

https://indico.fnal.gov/event/52666/contributions/231769/attachments/153101/198580/SCDProjects_LDRD_Knoepfel.pdf

Kyle J. Knoepfel

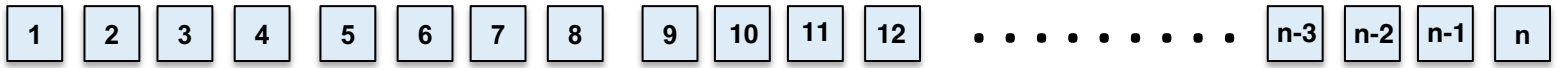
DUNE pre-collaboration meeting

9 May 2022

Processing complications (1)

- The collider physics approach to data processing

  Existing computing frameworks process data in rigid ways

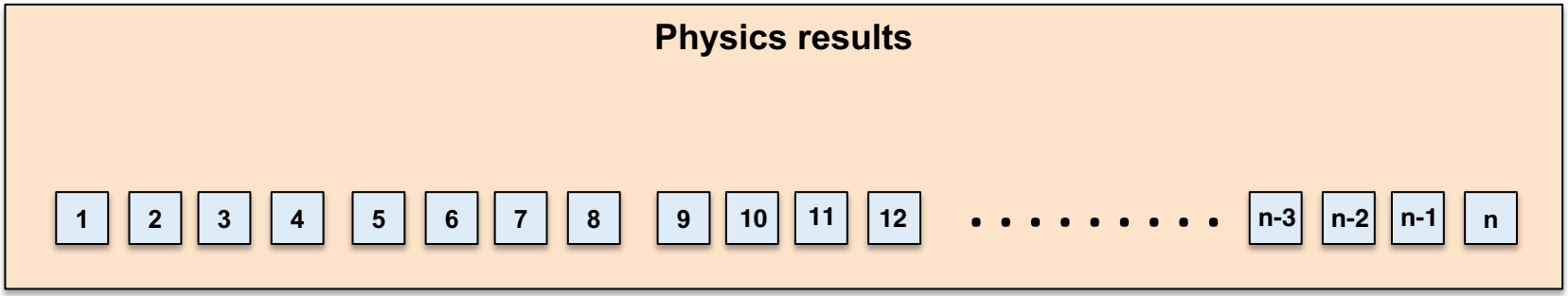


Fermilab

Processing complications (1)

- The collider physics approach to data processing

Existing computing frameworks process data in rigid ways



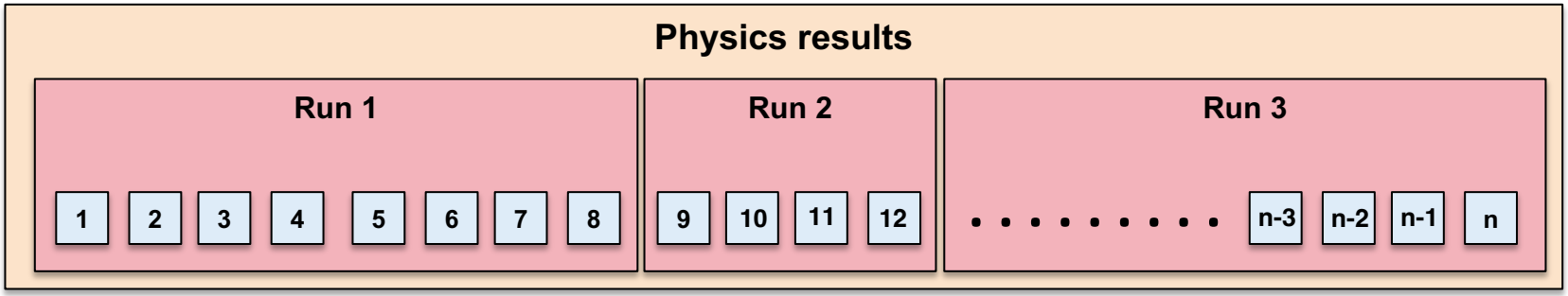
Physics results are obtained by analyzing the data as a whole

Fermilab

Processing complications (1)

- The collider physics approach to data processing

  Existing computing frameworks process data in rigid ways



  Physics results are obtained by analyzing the data as a whole

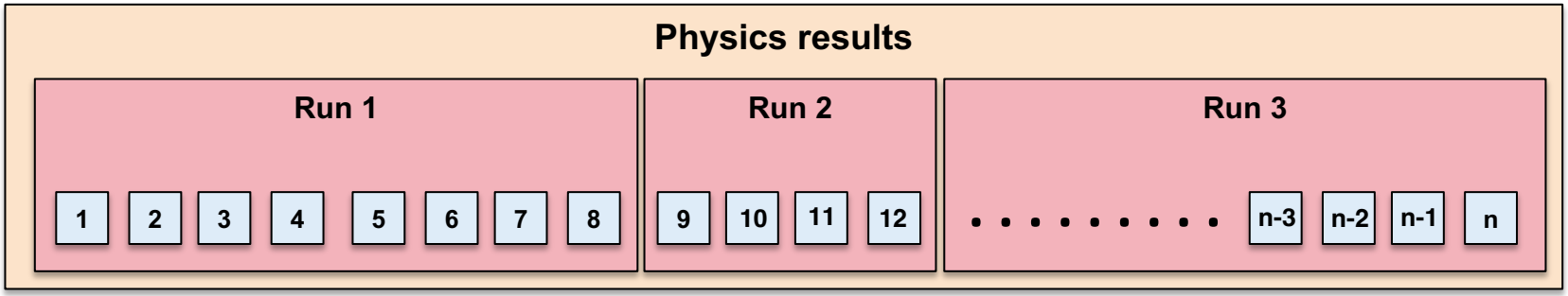
  Data is grouped into runs to reflect changes in (e.g.) beam or detector conditions

Fermilab

Processing complications (1)

- The collider physics approach to data processing

Existing computing frameworks process data in rigid ways



Physics results are obtained by analyzing the data as a whole

Data is grouped into runs to reflect changes in (e.g.) beam or detector conditions

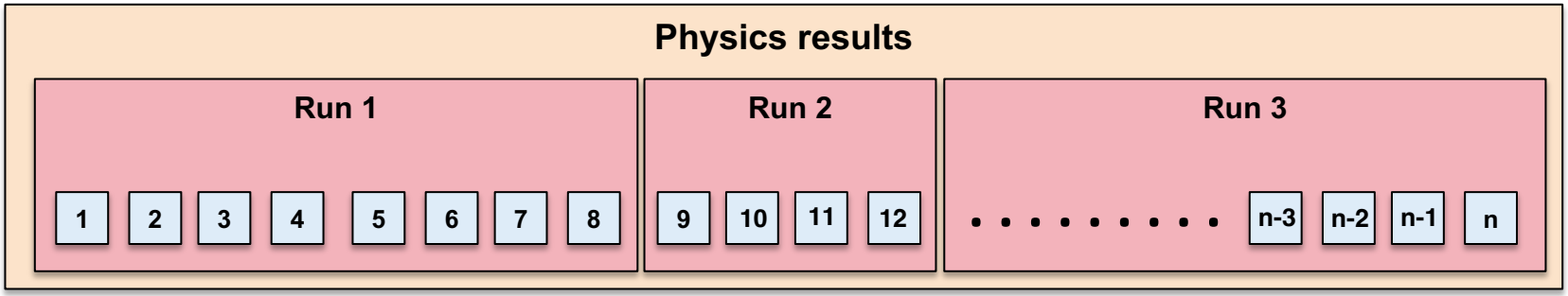
Data-processing frameworks set these hierarchies “in stone”

Fermilab

Processing complications (1)

- The collider physics approach to data processing

  Existing computing frameworks process data in rigid ways



  Physics results are obtained by analyzing the data as a whole

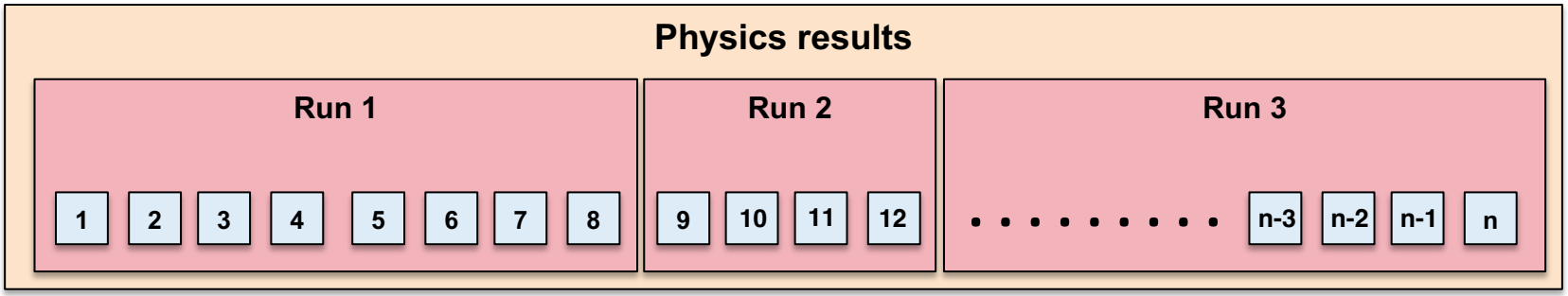
  Data is grouped into runs to reflect changes in (e.g.) beam or detector conditions

  Data-processing frameworks set these hierarchies “in stone”

- **This approach does not work well for DUNE.**

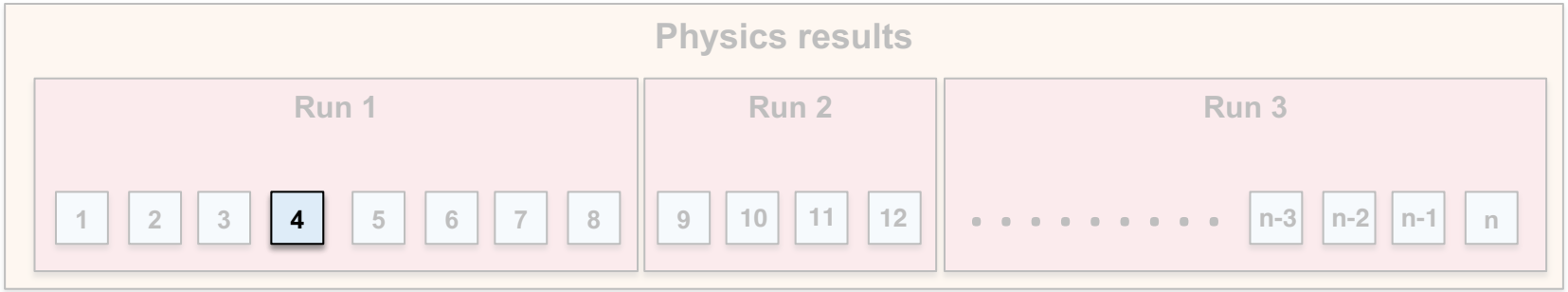
Fermilab

Processing complications (2)



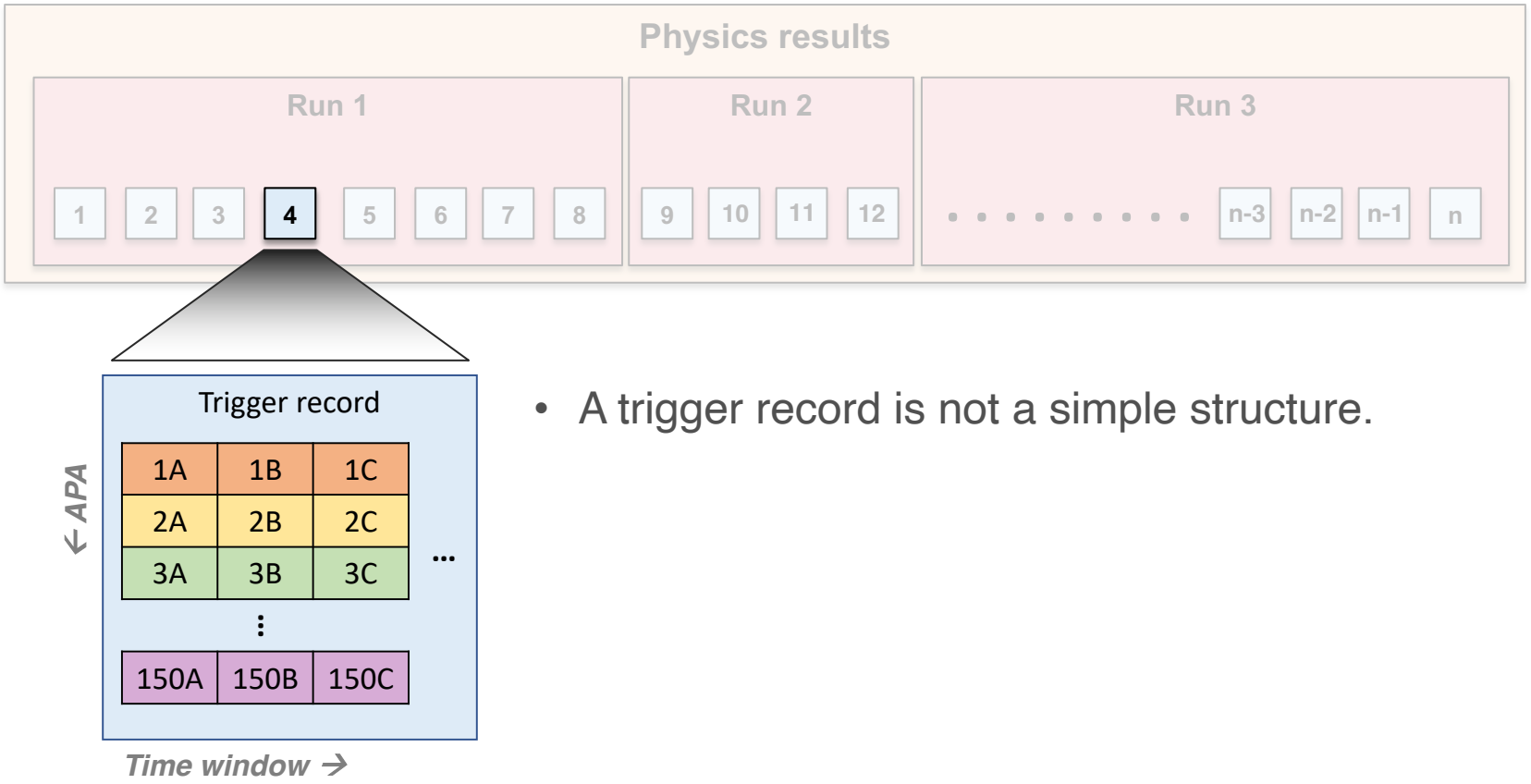
Fermilab

Processing complications (2)



Fermilab

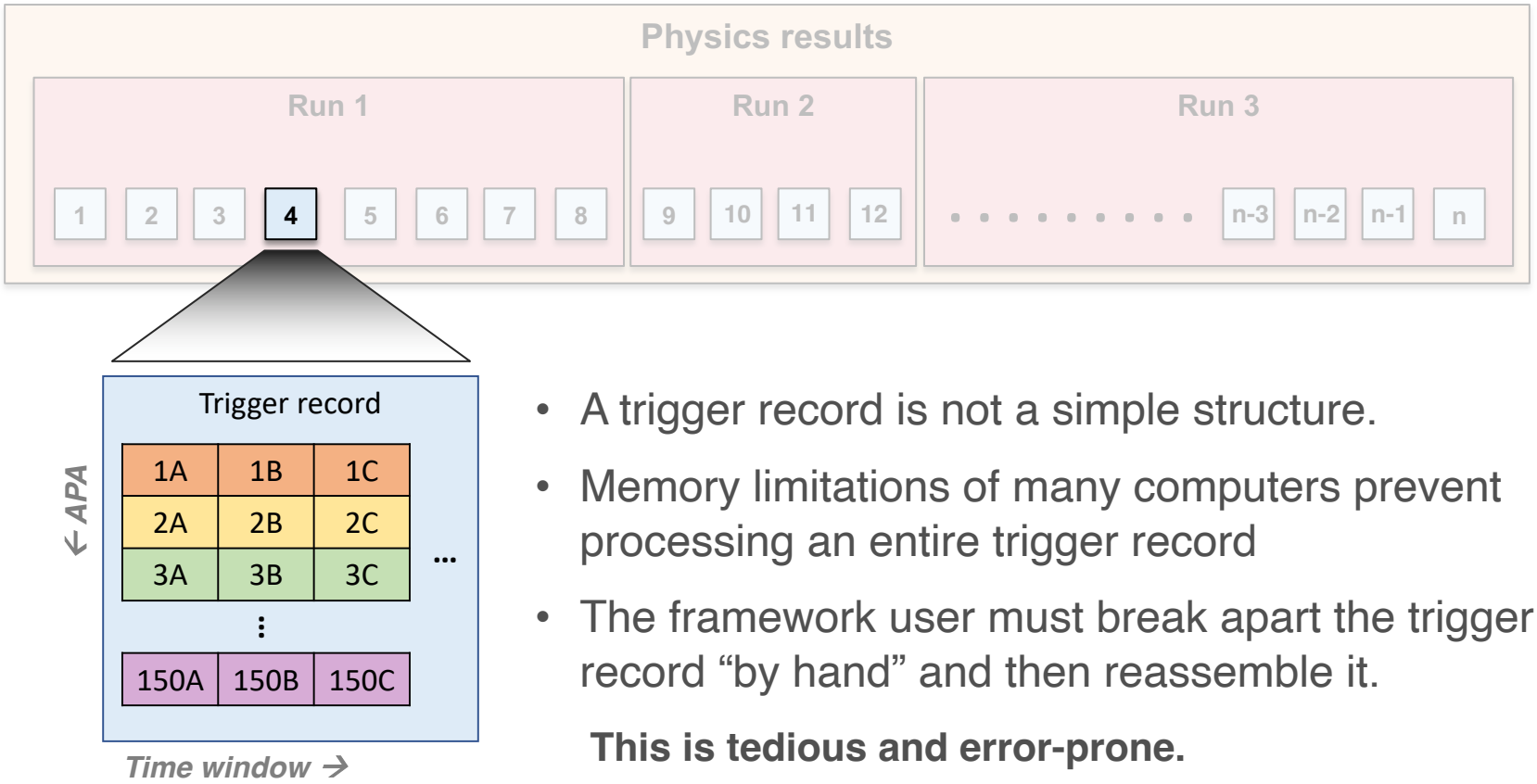
Processing complications (2)



- A trigger record is not a simple structure.

Fermilab

Processing complications (2)

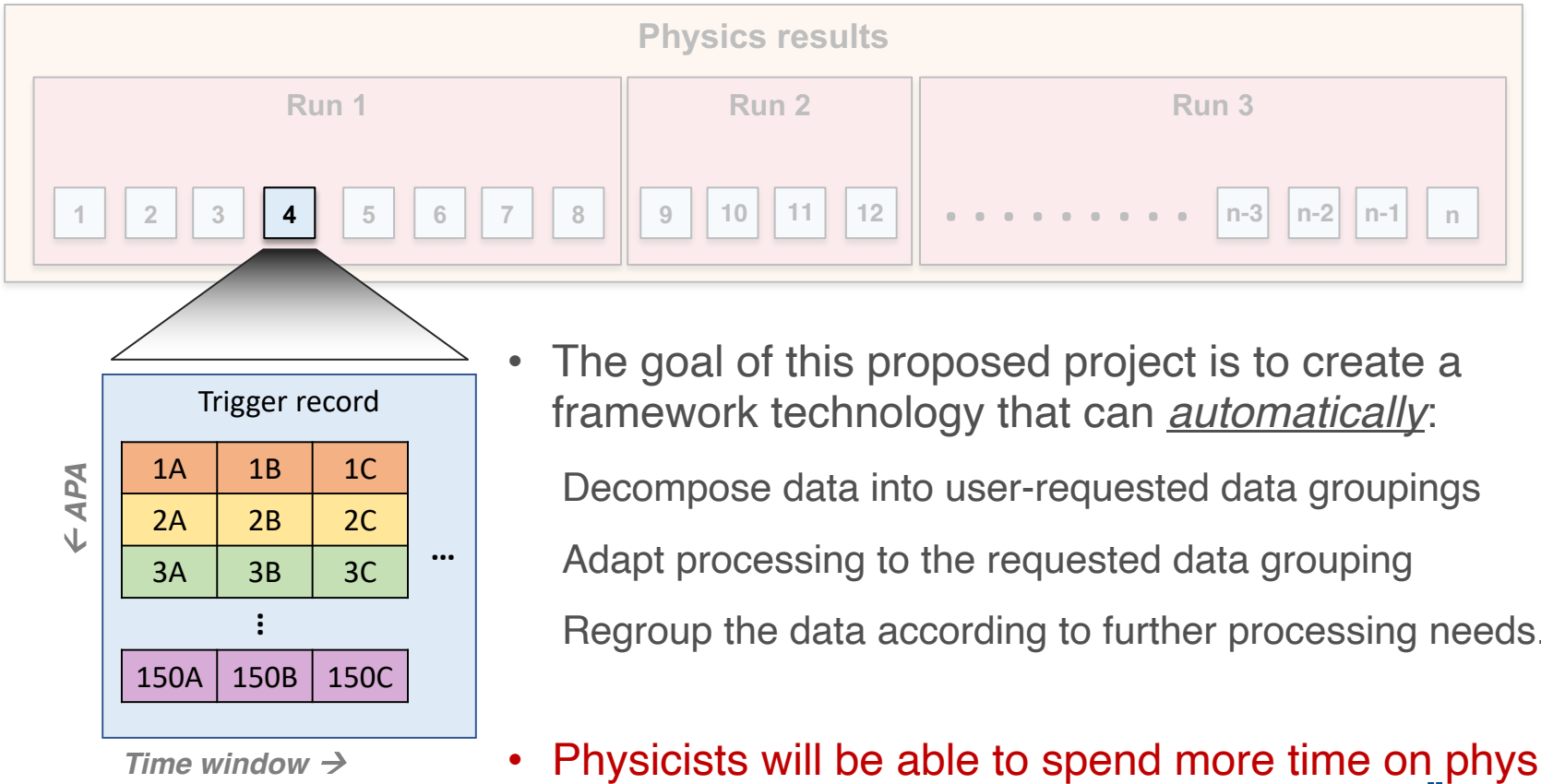


- A trigger record is not a simple structure.
- Memory limitations of many computers prevent processing an entire trigger record
- The framework user must break apart the trigger record “by hand” and then reassemble it.

This is tedious and error-prone.

Fermilab

Processing complications (2)



- The goal of this proposed project is to create a framework technology that can *automatically*:
  - Decompose data into user-requested data groupings
  - Adapt processing to the requested data grouping
  - Regroup the data according to further processing needs.
- Physicists will be able to spend more time on physics.

Fermilab

Establishing expectations

- **Goal of LDRD is to explore solutions for DUNE's framework needs**

  A monthly "small" meeting with DUNE has been established to ensure that LDRD directions are headed in the right direction.

Establishing expectations

- **Goal of LDRD is to explore solutions for DUNE's framework needs**

  A monthly "small" meeting with DUNE has been established to ensure that LDRD directions are headed in the right direction.

- ***However…***

  The scope is broader than just DUNE.

  I request DUNE's ideas/guidance, but the project is still self-directed.

  The deliverable is to inform how to adjust existing frameworks, ***not*** a new framework.

Establishing expectations

- **Goal of LDRD is to explore solutions for DUNE's framework needs**

  A monthly "small" meeting with DUNE has been established to ensure that LDRD directions are headed in the right direction.

- ***However…***

  The scope is broader than just DUNE.

  I request DUNE's ideas/guidance, but the project is still self-directed.

  The deliverable is to inform how to adjust existing frameworks, ***not*** a new framework.

- Prototype package https://github.com/knoepfel/meld

  It will change ***a lot*** in the months ahead.

Meld

External software	Provided by Spack or system
Building and testing	CMake
Configuration	CLI11 + Boost JSON (probably Jsonnet later; not FHiCL or Python)
Plugin handling	Boost DLL
Task scheduling	Intel oneTBB (yes, MPI is on the table)

Meld

External software	Provided by Spack or system
Building and testing	CMake
Configuration	CLI11 + Boost JSON (probably Jsonnet later; not FHiCL or Python)
Plugin handling	Boost DLL
Task scheduling	Intel oneTBB (yes, MPI is on the table)

Meld already supports:

- Experiment-defined data-processing levels
- Parallelism across levels

  Uses TBB flow graph

Meld

External software	Provided by Spack or system
Building and testing	CMake
Configuration	CLI11 + Boost JSON (probably Jsonnet later; not FHiCL or Python)
Plugin handling	Boost DLL
Task scheduling	Intel oneTBB (yes, MPI is on the table)

Meld already supports:

- Experiment-defined data-processing levels
- Parallelism across levels

  Uses TBB flow graph

Meld does not yet support:

- Bookkeeping/metadata
- I/O system
- Serializing for a specific resource

The basic idea

- The framework should support experiment-defined processing levels.

  Inheritance-based modules don't necessarily work well.

- `meld` uses template metaprogramming to process data of the "right type".

  This may change.

The basic idea

- The framework should support experiment-defined processing levels.

  Inheritance-based modules don't necessarily work well.

- `meld` uses template metaprogramming to process data of the "right type".

  This may change.

```
#include "meld/core/module.hpp"
#include "meld/utilities/debug.hpp"
#include "test/data_levels.hpp"

#include <chrono>
#include <iostream>
#include <thread>
#include <typeinfo>

namespace meld::test {
  struct run_only_module {
    void
    process(run const& r, concurrency::unlimited) const
    {
      using namespace std::chrono_literals;
      std::this_thread::sleep_for(1s);
      debug("Processing ", r.name(), ' ', r, " in run-only module.");
    }
  };

  MELD_REGISTER_MODULE(run_only_module, run)
}
```

Issues to address

- Looking through DUNE's workflow
- **Finding: It's not clear yet what the data flow should look like.**

  Does the concept of a "path" still make sense? *Probably*

  What's the best scheduler for the various use cases? *Under exploration*

  What does it look like in code to "reassemble" a trigger record without blowing the memory budget? *Don't know yet*

- I am working through these types of issues.