Argonne NATIONAL LABORATORY

ATLAS Metadata Infrastructure Evolution for Run 2 and Beyond

Dr. Jack CRANSHAW

Dr. David MALON

- Dr. Peter VAN GEMMEREN

Dr. Alexandre VANIACHINE

On behalf of the ATLAS Collaboration

U.S. DEPARTMENT OF ENERGY

Outline

- Metadata use-cases
- The Run 1, incident-driven metadata infrastructure
- The metadata infrastructure inside the multi-processing framework
- Fine-grained event processing framework and metadata
- Metadata infrastructure and ATLAS future frameworks

Introduction

- Metadata are essential to event data processing, in a variety of roles
- Metadata handling and metadata flow, though, differ in significant ways from event loop management and execution control
  - Handling may be asynchronous to control framework state machine transitions, object lifetime management is different, scheduling, processing, and propagation may be different, …
- Evolution of event processing frameworks (to multiprocessing and multithreaded models, to scatter-gather architectures, to operability on heterogeneous and high-performance platforms) and the need for metadata access in analyses downstream of experiments' frameworks, require concomitant evolution of metadata handling and its supporting infrastructure
- This presentation describes some of the ways in which ATLAS metadata infrastructure is evolving to address these needs

Metadata use-cases

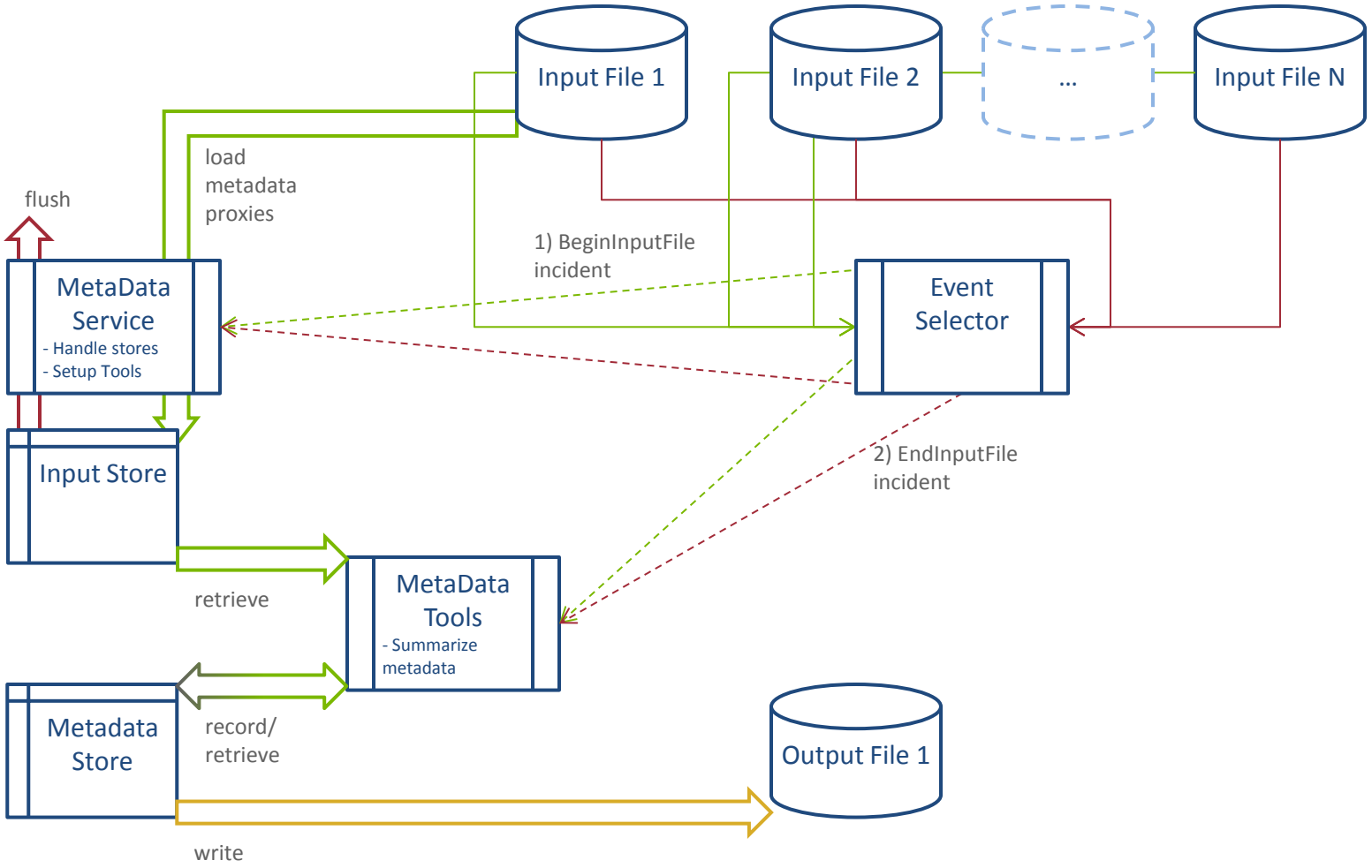
- Data Integrity (Framework)
  - There are certain metadata that identify the file and its contents and how it connects to larger data groupings such as datasets.
- Provenance (Framework)
  - As ATLAS has a long chain of data products, it is useful for a file to know from which data products it was derived.
- Auto-configuration (User/Framework)
  - Job-configuration depends on data that is stored as in-file metadata. This can include cached conditions data using an interval-of-validity structure.
- Bookkeeping (User/Framework)
  - A lot of things happen to the data during processing including filtering. The metadata tracks counts needed for efficiency and luminosity calculations.
    - This can also be needed by the grid data processing to make sure files are merged into data blocks of related data, e.g. luminosity blocks, physics streams, etc.
- Analysis Caching (User)
  - At the analysis level the metadata can be used to help physicists avoid database or release accesses by caching data needed within the file. At this point the data and tools also need to be in a form that can be used outside Athena, the ATLAS control framework.

The Run 1, incident-driven metadata infrastructure

- This infrastructure profits from a rich feature set provided by the ATLAS execution control framework including:

1. Standardized interfaces and invocation mechanisms for tools and services
   - It uses a MetaDataSvc to handle StoreGate instances for MetaData and relies on MetaDataTools to summarize and propagate metadata from input files.
2. Segregation of transient data stores with concomitant object lifetime management
   - An input metadata store is used for reading and mirrors the lifetime of the input file (i.e.: is flushed on input file transitions). Metadata objects are written from a separate metadata store.
3. Mechanisms for handling occurrences asynchronous to the control framework's state machine transitions
   - MataDataSvc and MetaDataTools are invoked via handling FileIncidents that are fired on file boundaries.

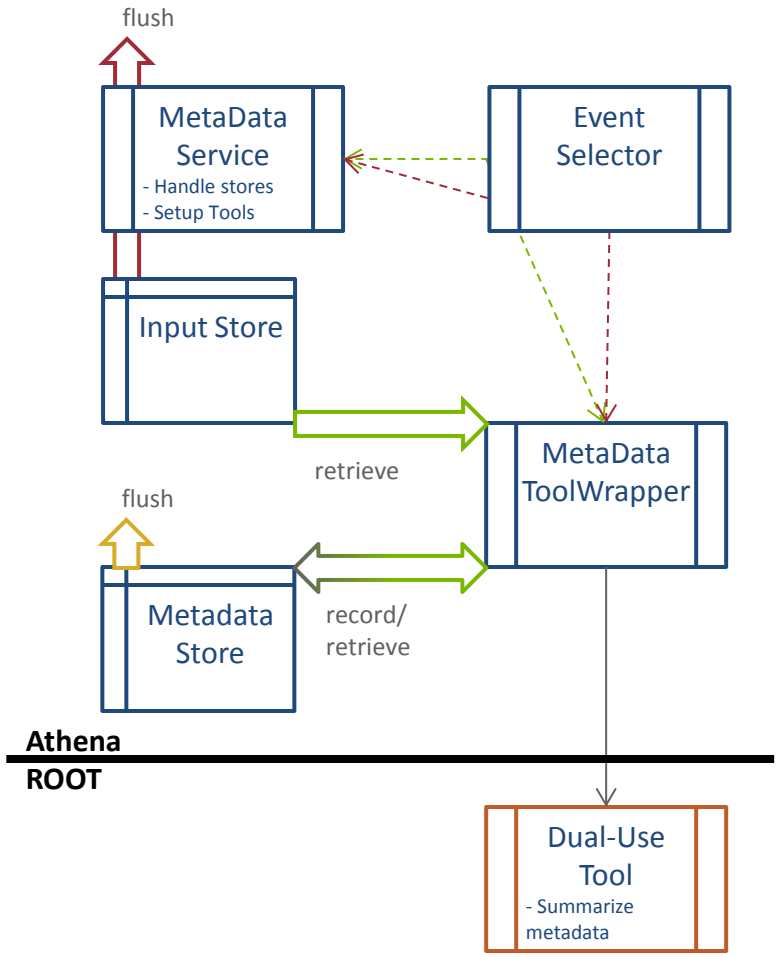
Incident-driven metadata infrastructure



The metadata infrastructure reuse in downstream physics analyses on xAOD

see <Scott Snyder: Implementation of the ATLAS Run 2 event data model>

- ATLAS has changed its event data model to unify Athena and ROOT analyses (xAOD).
- To allow the reuse of metadata components in downstream analyses that are not utilizing the ATLAS control framework, Dual-Use Tools are being developed to summarize metadata records.
  - Transfer some functionality from the framework MetaDataTools to Dual-Use Tools
  - Provide generic MetaDataTool-Wrapper to allow framework integration
    - Listen to incidents
    - Interact with stores

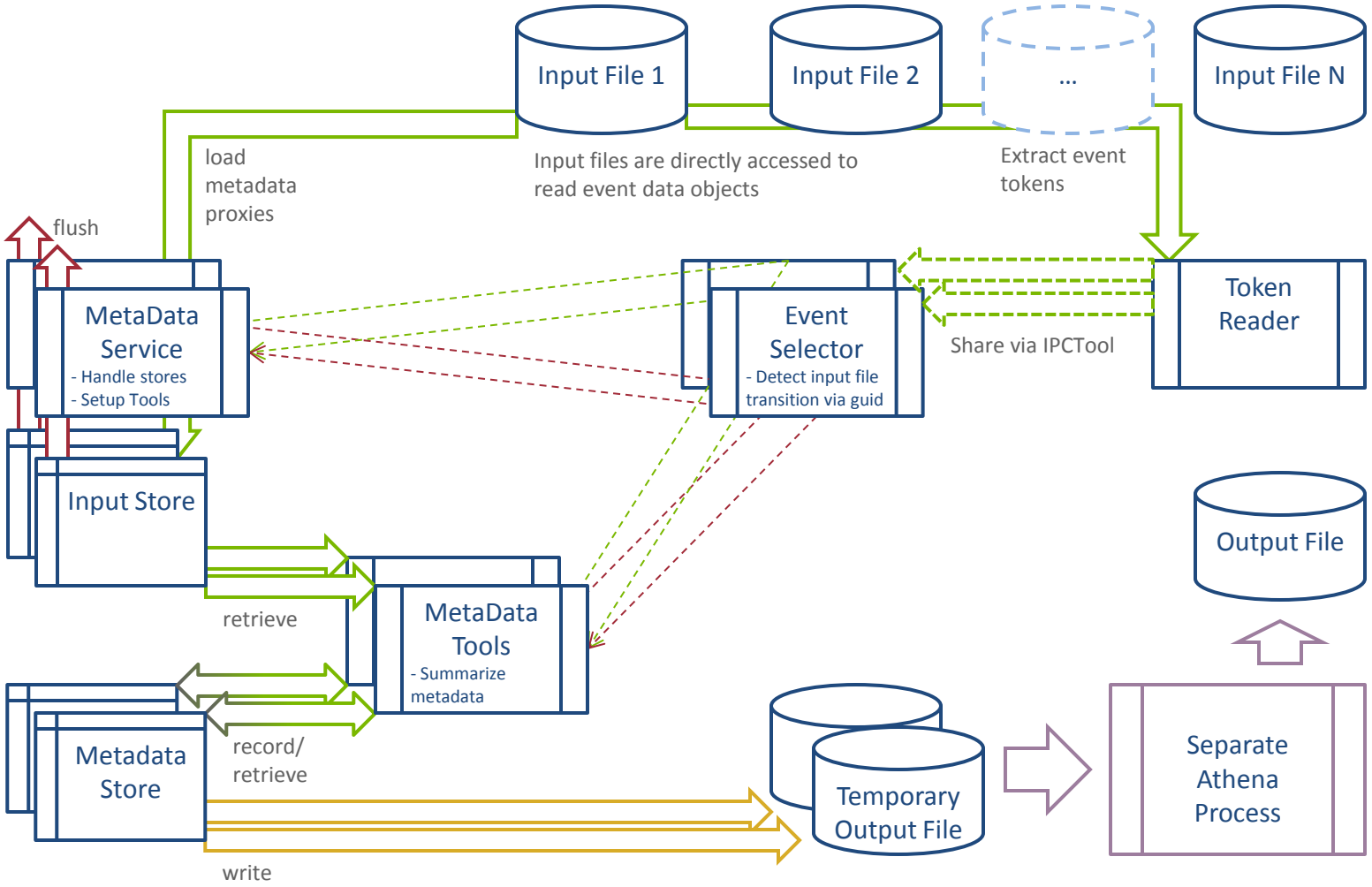


The metadata infrastructure inside AthenaMP

see <Vakho Tsulaia: Running ATLAS workloads within massively parallel distributed applications using Athena Multi-Process framework (AthenaMP)>

- AthenaMP, the multi-processing version of the ATLAS control framework, starts out as a single process and after initialization forks of several worker processes to process the events.
  1. Each worker has an EventSelector that reads their own sub-sample of events directly from the input files
  2. Incident firing is the same as for serial execution and each worker has access to all the metadata.
     - If a worker does not process any events in an input file then that file is skipped for metadata processing as well.
  3. Each worker produces an output file containing events and metadata. These output files have to be merged after completion of the AthenaMP job.
     - In Run 1, metadata merging required execution of the full Athena framework, whereas event data can be appended with more light-weight tools.
- Optionally, a shared TokenReader can be used to iterate over the input files and dispatch event processing by sending Token (event references) to the worker.
  - The worker will still access the file directly for event and metadata.

Metadata infrastructure inside AthenaMP with token reader

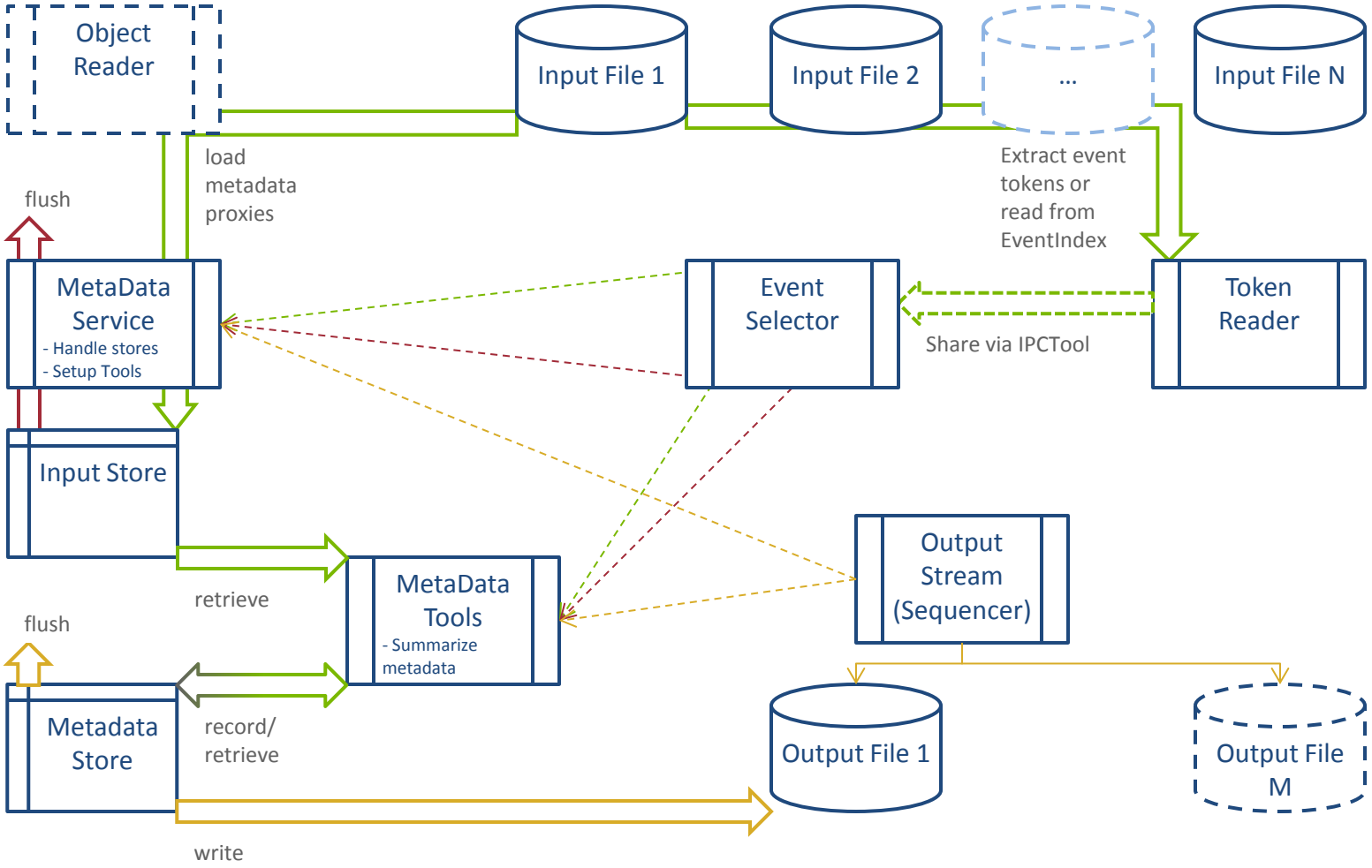


Event service framework and metadata

see <Torre Wenaus: The ATLAS Event Service: A New Approach to Event Processing>

- Fine-grained approach to event processing to exploit opportunistic, potentially short-lived resources (such as HPC, Amazon spot market or volunteer computing).
  - Decouple processing from the chunkiness of files, from data locality considerations, from WAN access latencies
- Stream output events away quickly to minimize losses if the worker vanishes and small local storage demands.
  - Jobs cannot rely on finalization before producing output files. Instead a new incident is used to trigger output file transitions.
  - Straight forward implementation for event data, but metadata clients need to be able to produce records describing the sub-sample of events in the output file.
- For the Event Service, AthenaMP manages distribution of events to parallel workers via TokenReader
- Workers retrieve event data using the token either by directly accessing the file or using a shared ObjectReader (in development).

Metadata infrastructure with output file sequencing



Metadata infrastructure and ATLAS future frameworks

- Requirements to ensure that all events in semantically meaningful units are processed, and to maintain semantic integrity in data organization, are independent of framework and processing architecture
  - But may need to be supported differently, and with greater generality
- ATLAS future framework requirements foresee decreased reliance upon incidents
  - Due to potential blocking and other issues in multithreaded deployment
  - Framework retains as necessary for some purposes the notion of "schedulable incidents," but proposes that most operations be done under the control of a scheduler
  - "Asynchronous to Gaudi state machine transitions" is not the same as "unschedulable"
- It is not difficult to define a whiteboard architecture in which some components are listening/waiting, not for event data, but for metadata objects
  - Not unschedulable
  - Such a strategy may require framework evolution to support heterogeneity in the "type" of the next datum to be processed: it may not always be the "next event"
- Must be possible to propagate semantic context from input to output, with context accessible to components that need it

Outlook

- During Run 1, a robust and versatile metadata infrastructure has proven essential for ATLAS.
  - Job configuration relies on in-file metadata
  - Event filtering requires sufficient bookkeeping metadata
- Run 2 conditions further emphasize the importance of metadata for distributed data processing and analysis
  - Increased data rates will cause Luminosity Blocks to no longer be constrained to a single file and their accounting becomes more complex.
  - A common Event Data Model for Athena and ROOT analyses requires to share metadata and tools handling metadata.
- At the same time, the move to new computing architectures requires extensions to the metadata infrastructure.