Trigger information data flow for the ATLAS EventIndex

M.Mineev[1], F.Prokoshin[2], A.Yakovlev[1]

(on behalf of the ATLAS Collaboration)

[1]Joint Institute for Nuclear Research, Dubna, Russia.
[2]Centro Científico Tecnológico de Valparaíso-CCTVal, Universidad Técnica Federico Santa María, Valparaiso, Chile.

Outline

- Introduction (EventIndex & Trigger Database libray)
- Trigger Database library (EITrigDB)
- EITrigDB library update
  - Monte Carlo data support
    - SMK-Trigger Chains table (MC Trigger DB)
    - R-tag – SMK table (AMI)
  - Changes in the EITrigDB library
- Conclusion
- Questions

EventIndex and EITrigDB library

EventIndex (EI) is a catalog for ATLAS experiment of data of all events in all processing stages needed to meet multiple use cases and search criteria.

ATLAS Events are stored in files (identified by GUID)

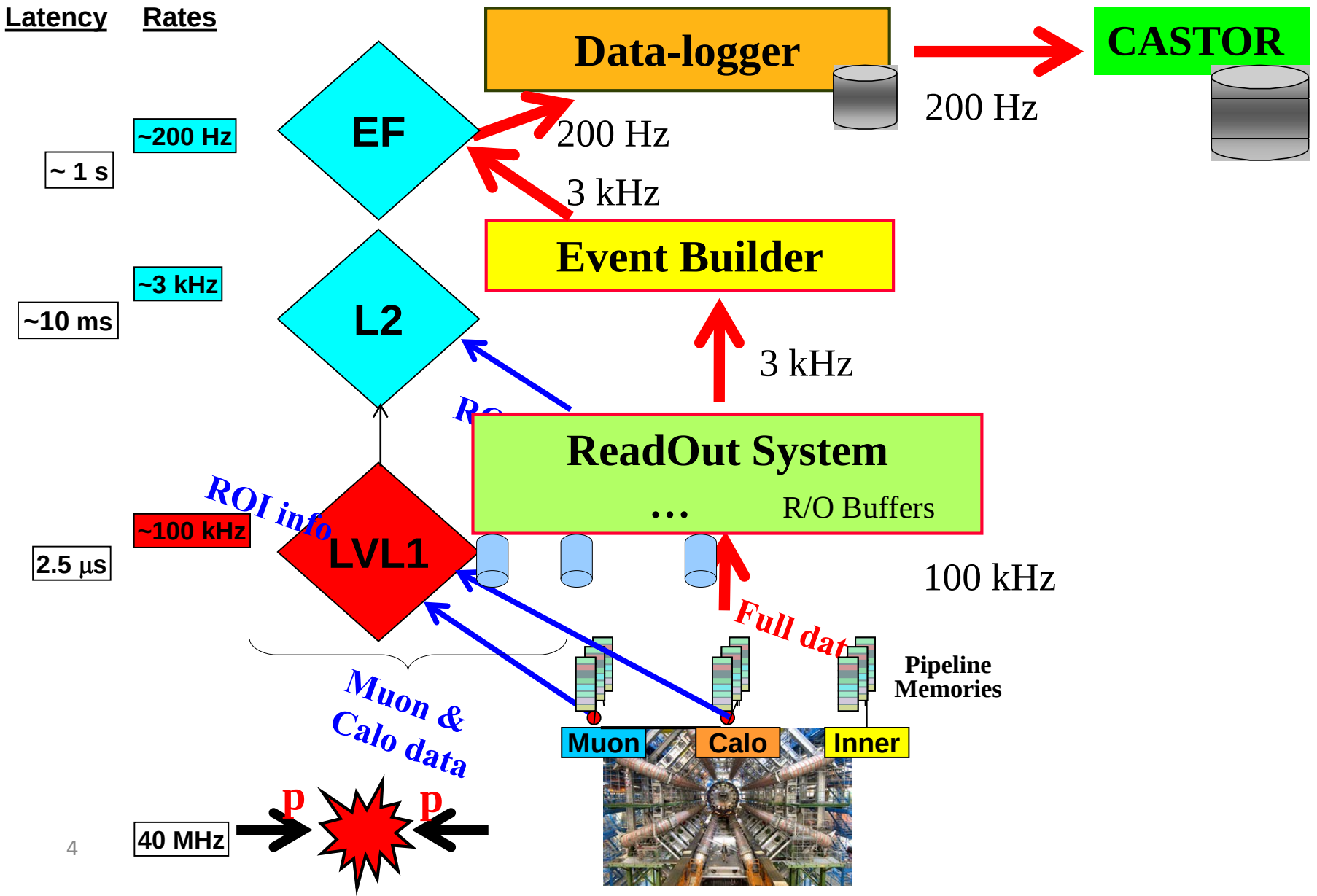
Files are grouped into DATASETS

Event Index information:

- Event identifiers (run / event numbers, trigger stream, luminosity block)
- Online trigger pattern (L1 and HLT)
- References (pointers) to the events at each processing step (RAW, ESD, AOD, DAOD) in all permanent files on storage

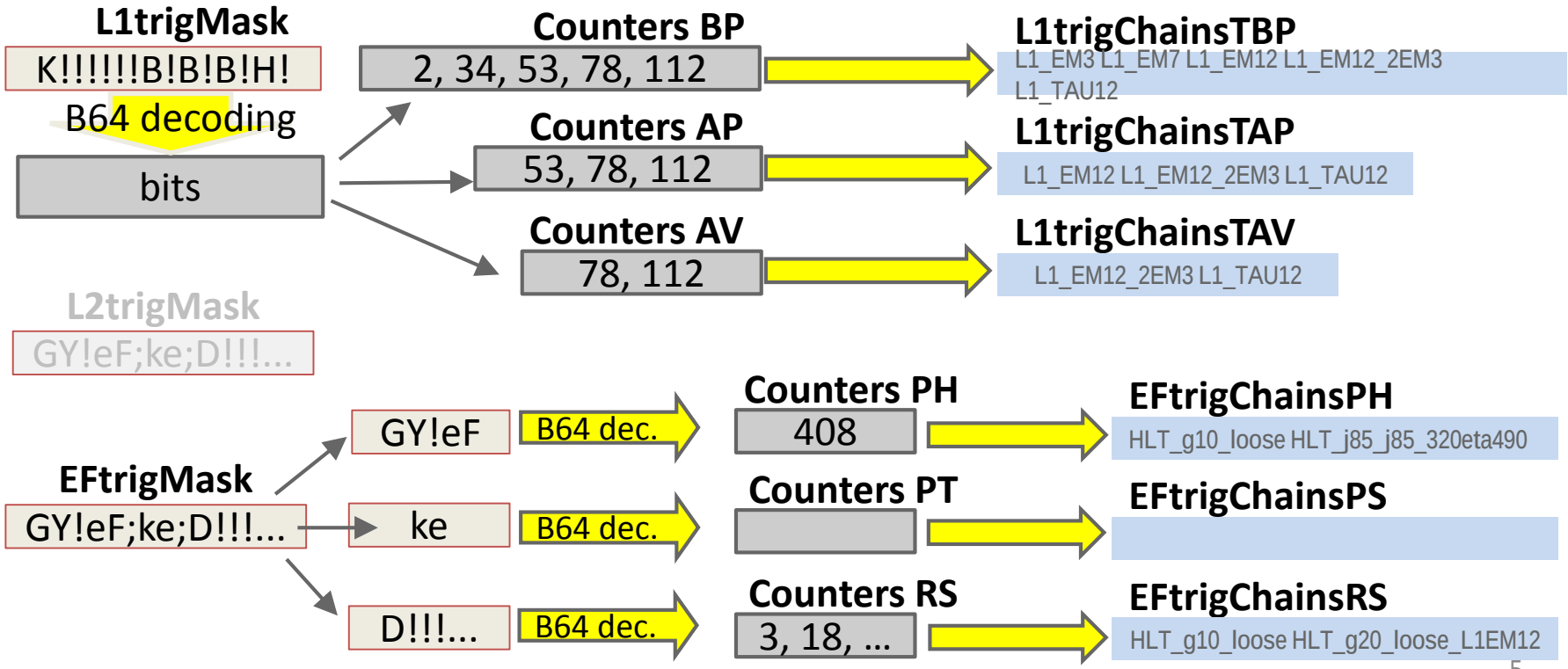
The **Trigger Database library** (**EITrigDB**) is used for trigger information getting and decoding in the EventIndex. This library is written in Java. Trigger information is stored in HBase. Apache HBase is an open-source, distributed, versioned, non-relational database. HBase provides Bigtable-like capabilities on top of Hadoop and HDFS.

ATLAS Trigger-DAQ levels



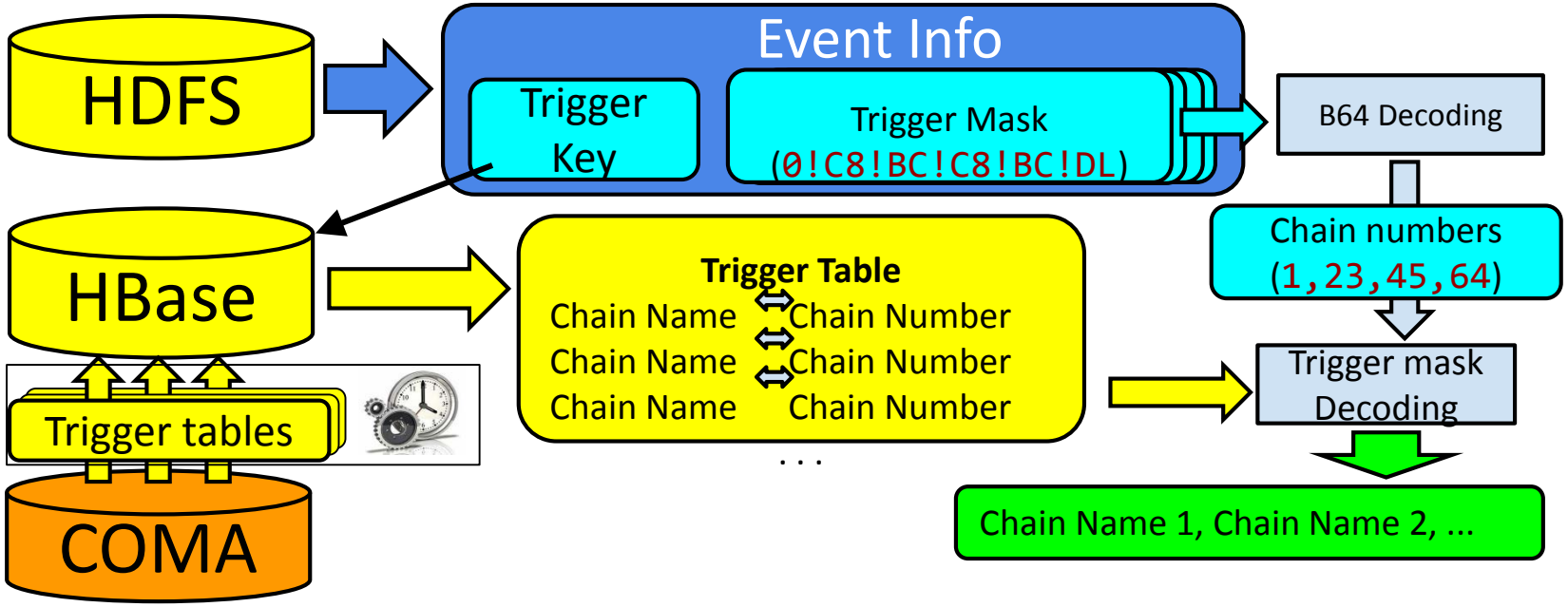
Trigger Data in the EventIndex

- Trigger decision is stored in data files as trigger masks.
- If a bit of the mask is set, this means that one of the sets of trigger conditions (Trigger chains) has been passed in this event. A position of this bit in mask (chain_counter) indicates which chains have passed.
- In Run 1 there were 3 trigger masks for 3 trigger levels
- In Run 2 L2 and EF were combined into HLT



Decoding trigger information

- Trigger tables, with information on relation between chain names and chain counters for specific interval are replicated from ATLAS COMA (COnditions/Configuration MetadatA) database to HBase.
- Trigger masks from event record are being decoded, and chain counters are converted to chain names
- Trigger masks depend on the Trigger Key (or SMK – Super Master Key)
- Initially this decoding was done on-the-fly when processing user requests
- Now trigger masks are converted to lists of trigger chain once, normally during import



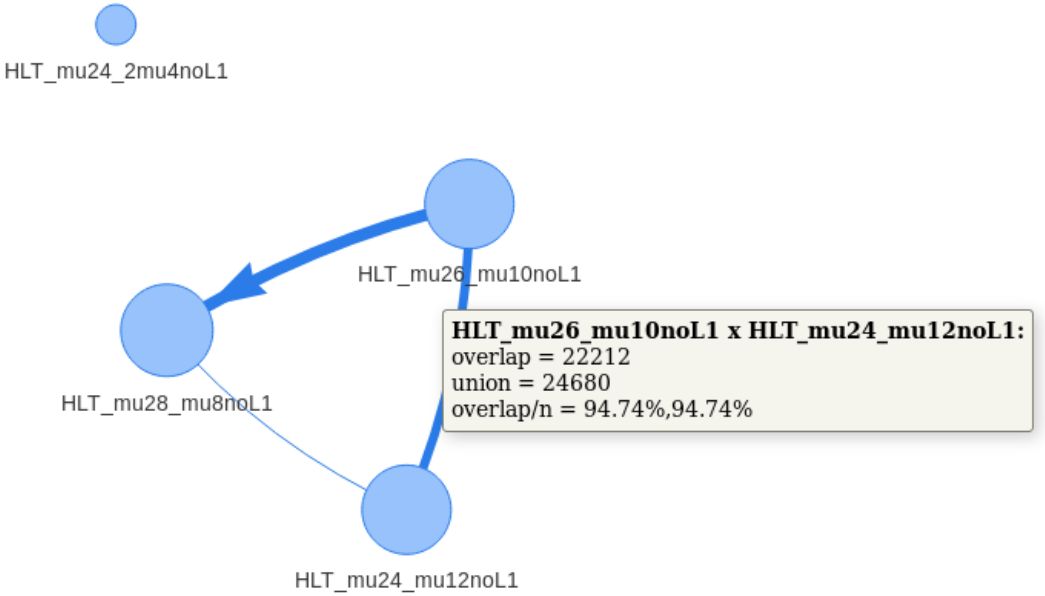
Hadoop Data Browser: Dataset Trigger Overlaps

https://atlas-event-index.cern.ch/EIHadoop

data18_13TeV.00356259.physics_Main.merge.AOD.f956_m2004 (based on 1000 events sample) evt=40069000,t3=39948793	**HLT_mu24_2mu4noL1** **240414** (0.60%)	**HLT_mu24_mu12noL1** **681173** (1.70%)	**HLT_mu26_mu10noL1** **801380** (2.00%)	**HLT_mu28_mu8noL1** **761311** (1.90%)
HLT_mu24_2mu4noL1 **240414** (0.60%)	**240414** (0.60%) (240414) 100.00%,100.00%	**40069** (0.10%) (881518) 16.67%,5.88%	**120207** (0.30%) (921587) 50.00%,15.00%	**120207** (0.30%) (881518) 50.00%,15.79%
HLT_mu24_mu12noL1 **681173** (1.70%)	**40069** (0.10%) (881518) 5.88%,16.67%	**681173** (1.70%) (681173) 100.00%,100.00%	**601035** (1.50%) (881518) 88.24%,75.00%	**560966** (1.40%) (881518) 82.35%,73.68%
HLT_mu26_mu10noL1 **801380** (2.00%)	**120207** (0.30%) (921587) 15.00%,50.00%	**601035** (1.50%) (881518) 75.00%,88.24%	**801380** (2.00%) (801380) 100.00%,100.00%	**721242** (1.80%) (841449) 90.00%,94.74%
HLT_mu28_mu8noL1 **761311** (1.90%)	**120207** (0.30%) (881518) 15.79%,50.00%	**560966** (1.40%) (881518) 73.68%,82.35%	**721242** (1.80%) (841449) 94.74%,90.00%	**761311** (1.90%) (761311) 100.00%,100.00%

Relational Graph

Triggers can be selected to get overlap matrix.
Relational graph is also available.



EITrigDB Library

EITrigDB library is written in Java.

The main EITrigDB library functionalities:

- decoding data stored in the trigger masks,
- getting trigger information from the databases (COMA, MC Trigger DB, AMI),
- data replication in HBase,
- reading the data from HBase.

EITrigDB core class included in the ATLAS EventIndex core package (Atlas-Event-Index-Core):

https://twiki.cern.ch/twiki/bin/viewauth/AtlasComputing/EventIndex

EITrigDB Library Update.

- **Monte Carlo** (MC) datasets support was added.
- Development challenges:
  - Absence of the documentation
  - Trigger data stored in different databases

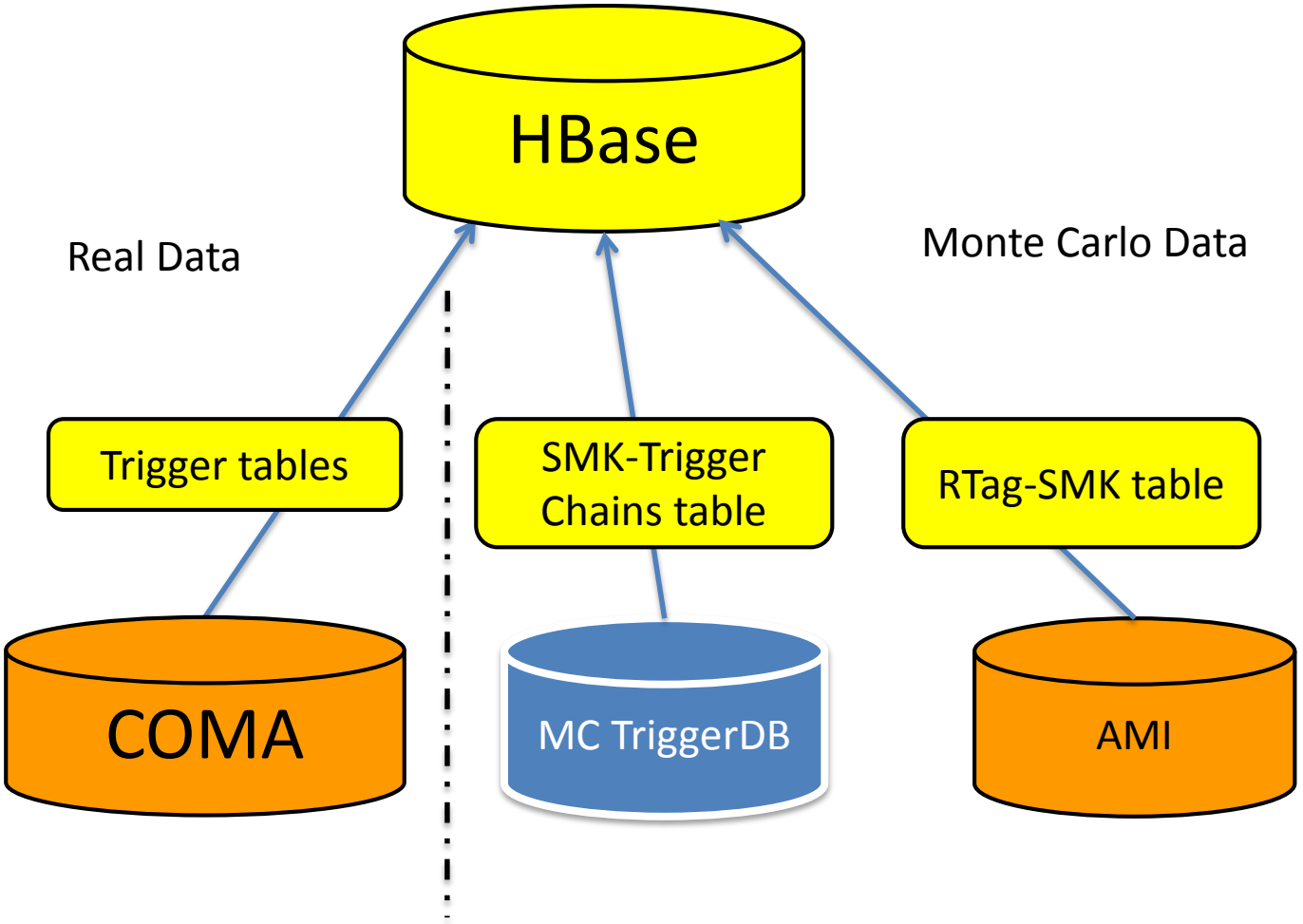
  (**MC TriggerDB** and **AMI** (ATLAS Metadata Interface))
- The two cases for SMK number:
  - SMK number is present in the EventIndex data,
  - SMK number is absent. In this case it is taken from r-tag (see below).

Trigger Chains for SMK

- MC Trigger information stored in the MC Trigger DB (TRIGGERDBMC).

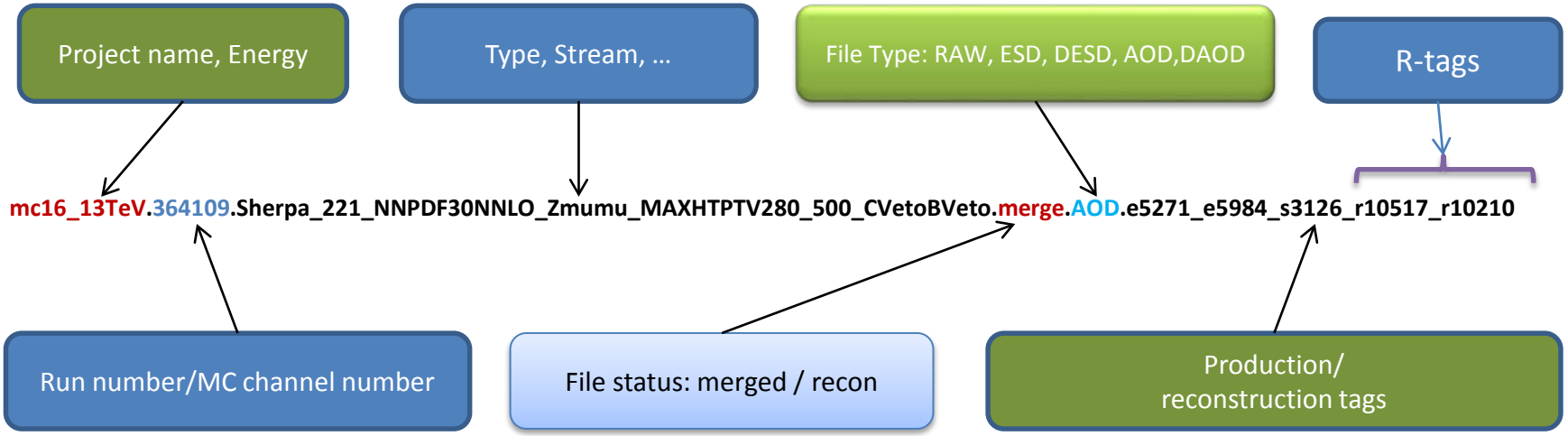
- ATLAS Offline Release contains the Python package TrigConfigSvc:
  https://twiki.cern.ch/twiki/bin/viewauth/Atlas/TrigConfigSvc
  - SQL query and DB parameters were used in the EITrigDB Java code to read the trigger chains.

Data Source for Monte Carlo



Dataset Name Structure

When the SMK info is absent it can be found from the dataset name



If SMK is absent it can be found using r-tag.

Real data:

run number -> SMK

Monte Carlo data:

reconstruction tag (r-tag) -> SMK

(the trigger information appears in the reconstruction stage)

AMI:

r-tag	productionStep	SMK
r10517	recon	2203
r10210	merge	-

R-tag & SMK on AMI Web-page

EITrigDB library update

The new classes added for Monte Carlo data:

EITrigDB -> EITrigDBMC (HBase)
EITrigDBig -> EITrigDBigMC (MC TriggerDB, AMI)

The methods have the same names in both cases:

getRunSMKTableOracle()
getSMKTableOracle()
getChainsOracle(long trig_key) …

New methods added in the EITrigDBigMC

(for example to get the SMK for a MC dataset name: getSMKForDataset(String name))

R-tags are stored in the HBase database as run numbers without preffix "r", to save the same DB structure in HBase.

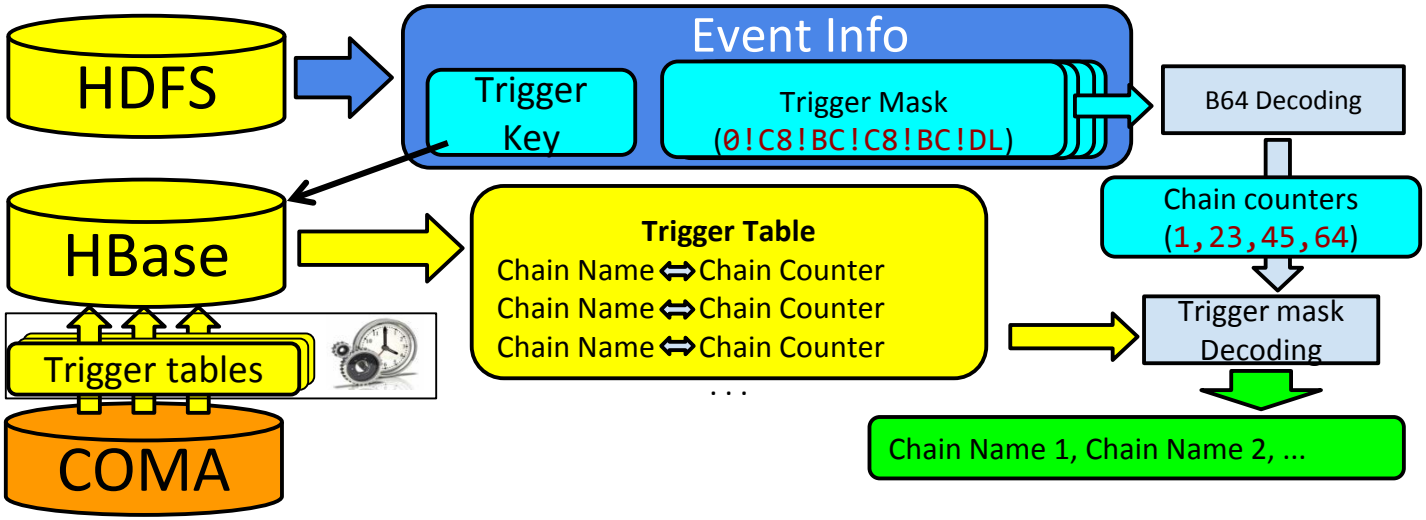
Current version uses external library written in Python (pyAMI) to get the data from AMI.

Conclusion

- EITrigDB package is working now as a stand-alone version.
- Trigger tables for 218 trigger keys (SMKs) were found in the TRIGGERDBMC during last tests.
- Integration in to the existing code is in the progress.
- Possible changes:
  - Data could be retrieved from the AMI by a Java client instead of pyAMI.

Backup Slides

- Trigger decision is stored in data files as trigger bit masks.
  - If a bit of the mask is set, this means that event passed one of the trigger chains.
  - A position of this bit in mask (chain counter) indicates which chains have passed.
- Trigger masks from event record are being decoded, and chain counters are converted to chain names using trigger tables replicated from COMA .



- A Trigger counting and overlap tool was developed
  - Count how many events passed each trigger chain
  - Counts overlaps between trigger chains
  - User can specify a list of trigger chains to search for overlaps, or get the list of all overlapped chains with corresponding trigger counts for them

EITrigDB Trigger data package

- Methods for manipulating trigger tables, importing them from COMA and decoding trigger decision information from the event records
- Support Run I and Run II data
- SVN repository: https://svnweb.cern.ch/cern/wsvn/atlas-atlevind/EITrigDB
- Core class Included in a TagConverter package

Methods that can be used from mr jobs

- chainFired(String chainName) - checks if event passed chain "chainName"
- trigFired(String chainName) – the same with on-the-fly decoding
- nChainsFired(long level, long type) – checks, how many chains passed an event
- nTrigsFired(long level, long type) - the same as previous, but using the trigger masks information.

level :	type:	
1 for L1 **2** for L2 **3** for EF **4** for HL	**(for L1)**	**(for L2 and EF)**
	0 for TAV **1** for TAP **2** for TBP	**0** for PH **1** for PT **2** for RS

- showChainsFired(long level, long type) - show chains that were passed in current event
- showChainsFiredDec(long level, long type) - the same as previous, but using the trigger masks information.
- showTrigMask(long level) - show trig mask for the current event
- showChainsFired(), showChainsFiredDec(), showTrigMask() - the same for all levels

Example:

```
ei -query 'regex path:EI15.1/data15_13TeV.*.physics_Main.merge.DAO*‘ –mr
'chainFired("EF_mu18_MG_medium")'
```

Trigger operations in TagConverter

Auxiliary Trigger transformation in TagConverter

- Decoding of trigger data during import
- For each Trigger level it creates three fields with lists of names of chains that passed this event
- Trigger transformation can be run again if necessary on already imported data, as it will be done with a data with xTrigDecision object (see later)

Trigger statistics and verification using Map Reduce

Usage: `ei -query path:<path> -mr 'true' -aux "net.hep.atlas.Database.EIHadoop.Accessor.Aux.TrigStat;-l <level>, -t <type> [-m]"`

Example: ei -query 'path:EI12.1/data12_8TeV.00209776.physics_Muons.merge.AOD.f475_m1218' -mr 'true' -aux "net.hep.atlas.Database.EIHadoop.Accessor.Aux.TrigStat;-l 1 -t 0 -m"

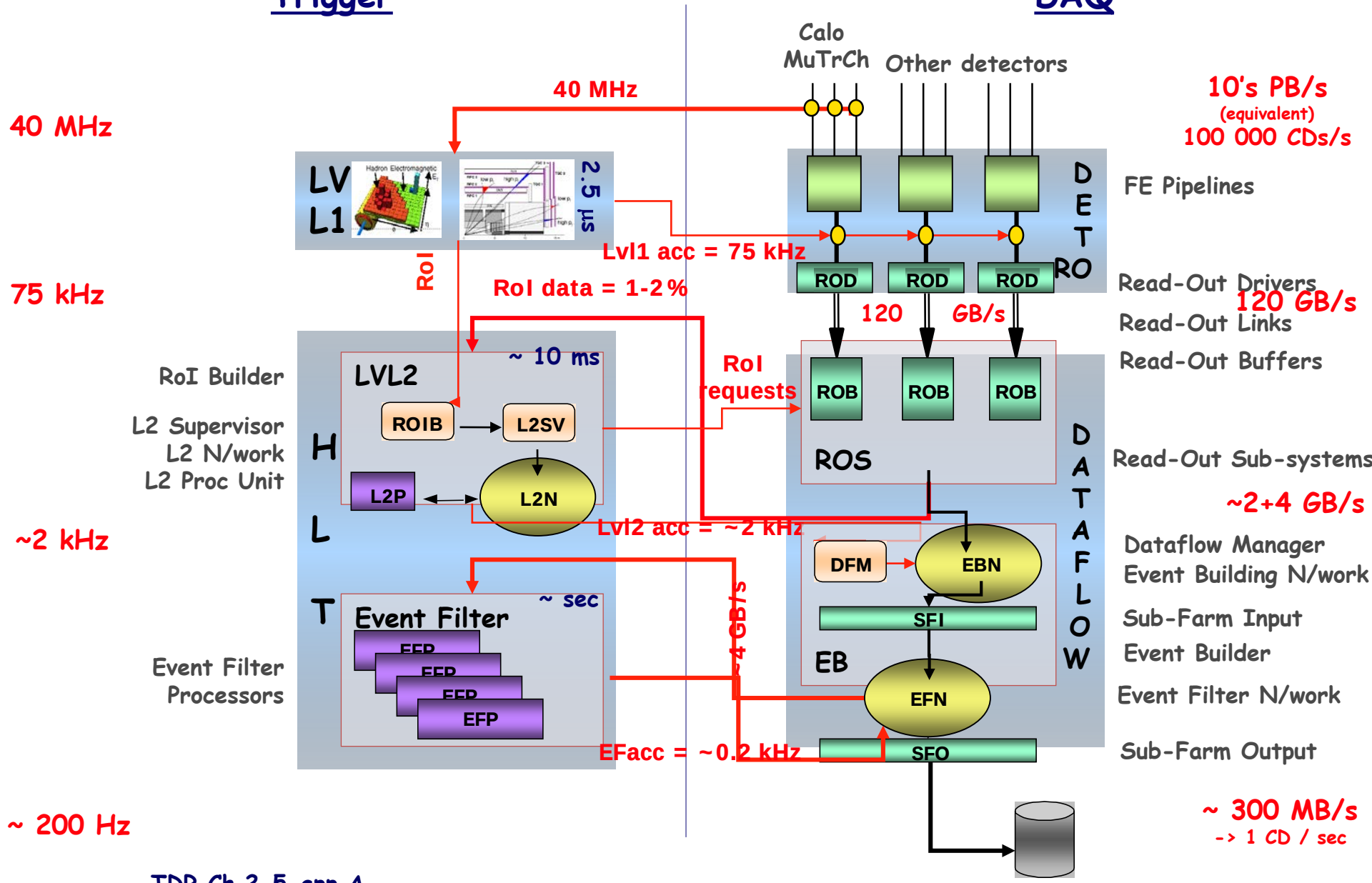
Keys:

--level, -l : Trigger level (L1, L2, EF)
--type, -t : Trigger type (TAV, PH, etc...)
-m : Do trigger mask decoding, otherwise use chain lists
--unknown, -u : check if trigger mask format is decodable

The output of the job is stored in the hdfs cahe, for example: EICache/prof/2014.11.17@16.12.52.514
it can be displayed or converted to human readable format using inspector

ARCHITECTURE

(Functional elements and their connections)



TDR Ch.2-5-app.A

References

AMI - ATLAS Metadata Interface
https://ami.in2p3.fr/

ATLAS COMA Database - COndition MetadatA
(Conditions/Configuration Metadata for ATLAS)
https://cds.cern.ch/record/1450065/files/ATL-SOFT-PROC-2012-040.pdf
http://iopscience.iop.org/article/10.1088/1742-6596/396/5/052033/meta

HBase
http://hbase.apache.org