HSF

HEP Software Foundation

Metadata

Dr Paul Laycock

UNIVERSITÉ DE GENÈVE

FACULTÉ DES SCIENCES

Département d'astronomie

GWSC GRAVITATIONAL WAVE SCIENCE CENTER

Overview

- **The HEP Software Foundation (HSF) and Metadata**
  - Metadata definitions and ownership
  - The Data Analysis working group
  - Paper, review and conclusions
- **HSF and Conditions Data**
  - The 2017 Community White Paper
  - Today, the HSF Conditions Data activity
  - HSF-style conditions databases is the next talk
- **Configuration**
  - DAGs and workflow-friendly code

Metadata, what is it and who owns it?

Metadata can be classified as **“non-event data”**. That’s about as fine-grained as the category of **“non-oxygen elements”**

Metadata covers a huge range of use cases

Metadata often starts with a well-defined use case, with well-defined *stake-holders, workflows and tools*

Metadata becomes problematic when someone downstream realises they also have a use case for some of that metadata that was produced upstream

Analysis lives downstream and tends to be impacted most by this problem, hence the HSF Data Analysis WG’s concern with the topic

Metadata, what is it and who owns it?

Metadata can be classified as **“non-event data”**. That’s about as fine-grained as the category of **“non-oxygen elements”**

Metadata covers a huge range of use cases

Metadata often starts with a well-defined use case, with well-defined *stake-holders, workflows and tools*

Metadata becomes problematic when someone downstream realises they also have a use case for some of that metadata that was produced upstream

Analysis lives downstream and tends to be impacted most by this problem, hence the HSF Data Analysis WG’s concern with the topic. ***End to end lifecycle management of Metadata is a topic that needs to be addressed.***

Metadata from the ~~salmon~~ analysis perspective

Navigating upstream tends to be difficult

Dealing with what washes downstream is often the only option

It comes in a variety of forms and may be of dubious provenance and quality

End to end lifecycle management of Metadata is a topic that needs to be addressed

HSF, Data Analysis, and Metadata

The **HSF Data Analysis working group (DAWG)** spent quite a lot of effort surveying metadata use cases, particularly in the context of data analysis:

May 19th 2019: *Workshop on Analysis Metadata:* https://indico.cern.ch/event/813208/

Jan 19th 2021: *DAWG Metadata discussions:* https://indico.cern.ch/event/993412/

Feb 2nd 2021: *DAWG Metadata discussions:* https://indico.cern.ch/event/993423/

Feb 16th 2021: *DAWG Metadata discussions:* https://indico.cern.ch/event/993424/

It received a lot of attention at both of the *"HSF Analysis Ecosystem workshops"*, particularly the second edition:

https://indico.cern.ch/event/613842/ (1st edition) and https://indico.cern.ch/event/1125222/

The HSF Metadata paper

The **HSF Data Analysis WG** published a paper based on their findings: [Constraints on Future Analysis Metadata Systems in High Energy Physics]

Essentially a wishlist covering the many use cases:

- *Dataset provenance, and user-defined dataset annotations*
- *Book-keeping, e.g. cutflows*
- *Data quality assessment*
- *Calibrations (fudge factors!)*
- *Monte Carlo annotations, e.g. cross-sections, k-factors, more fudge factors*

and how data access solutions should ideally look and feel (executive summary: “simple”)

The HSF Metadata paper review

The **HSF Data Analysis WG** paper: Constraints on Future Analysis Metadata Systems in High Energy Physics was reviewed by a team of experts who made their findings available for the ***second Analysis Ecosystem workshop***: https://arxiv.org/abs/2203.00463

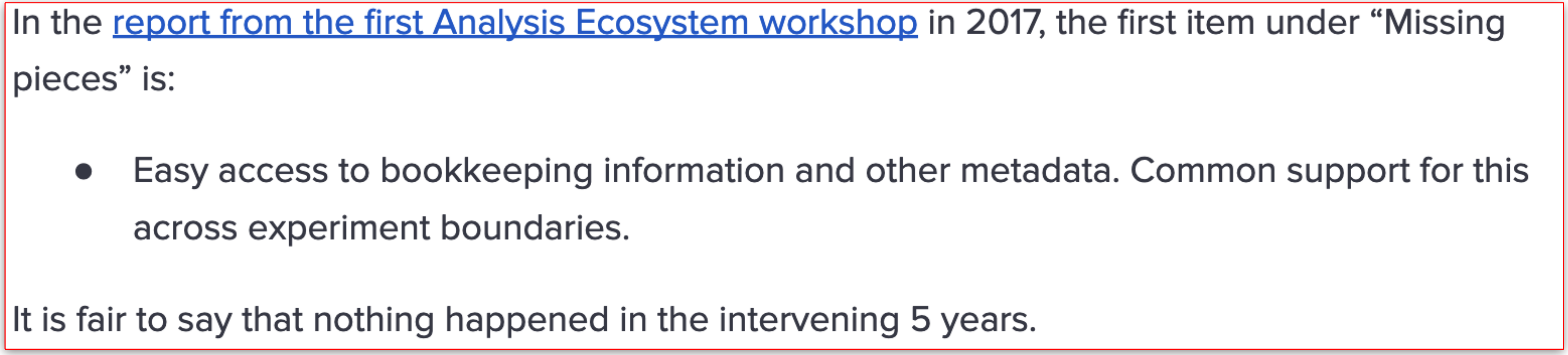
The review pointed out that more work was needed, in addition to the *end to end lifecycle management problem*, e.g. the definition of **metadata scopes** in that paper provoked several questions that need answers:

- *Metadata is data about data. Which data?*
- *On what does the metadata depend?*
- *Is the metadata known at the time of data production?*
- *How is the metadata produced and by whom?*
- *How is the metadata used and by whom?*
- *Who is the owner of the metadata, and are there transfers of ownership?*
- *What relations between different levels of metadata exist and how are they handled?*

The HSF Metadata status?

More work is needed, it’s not obvious that attacking “Metadata” as one big problem is the right approach, but it’s equally clear that we need solutions

Metadata was identified as a challenge in the first Analysis Ecosystem workshop and to quote the report from the second Analysis Ecosystem workshop report: https://doi.org/10.5281/zenodo.7003963



In the report from the first Analysis Ecosystem workshop in 2017, the first item under “Missing pieces” is:

- Easy access to bookkeeping information and other metadata. Common support for this across experiment boundaries.

It is fair to say that nothing happened in the intervening 5 years.

HSF and Conditions Data

- “Conditions Data” was one of the working groups of the HSF Community White Paper: *A Roadmap for HEP Software and Computing R&D for the 2020s, The HEP Software Foundation, Albrecht, J. et al., Comput Softw Big Sci (2019) 3, 7* https://doi.org/10.1007/s41781-018-0018-8

- The Conditions Data working group wrote a fuller report: https://arxiv.org/abs/1901.05429

- There was a good level of consensus across experiments about best practices, including:
  - ***Loose coupling between client and server using RESTful interfaces***
  - ***The ability to cache queries as well as payloads***
  - ***Separation of payload queries from metadata queries***

HSF Recommendations for Conditions Data access

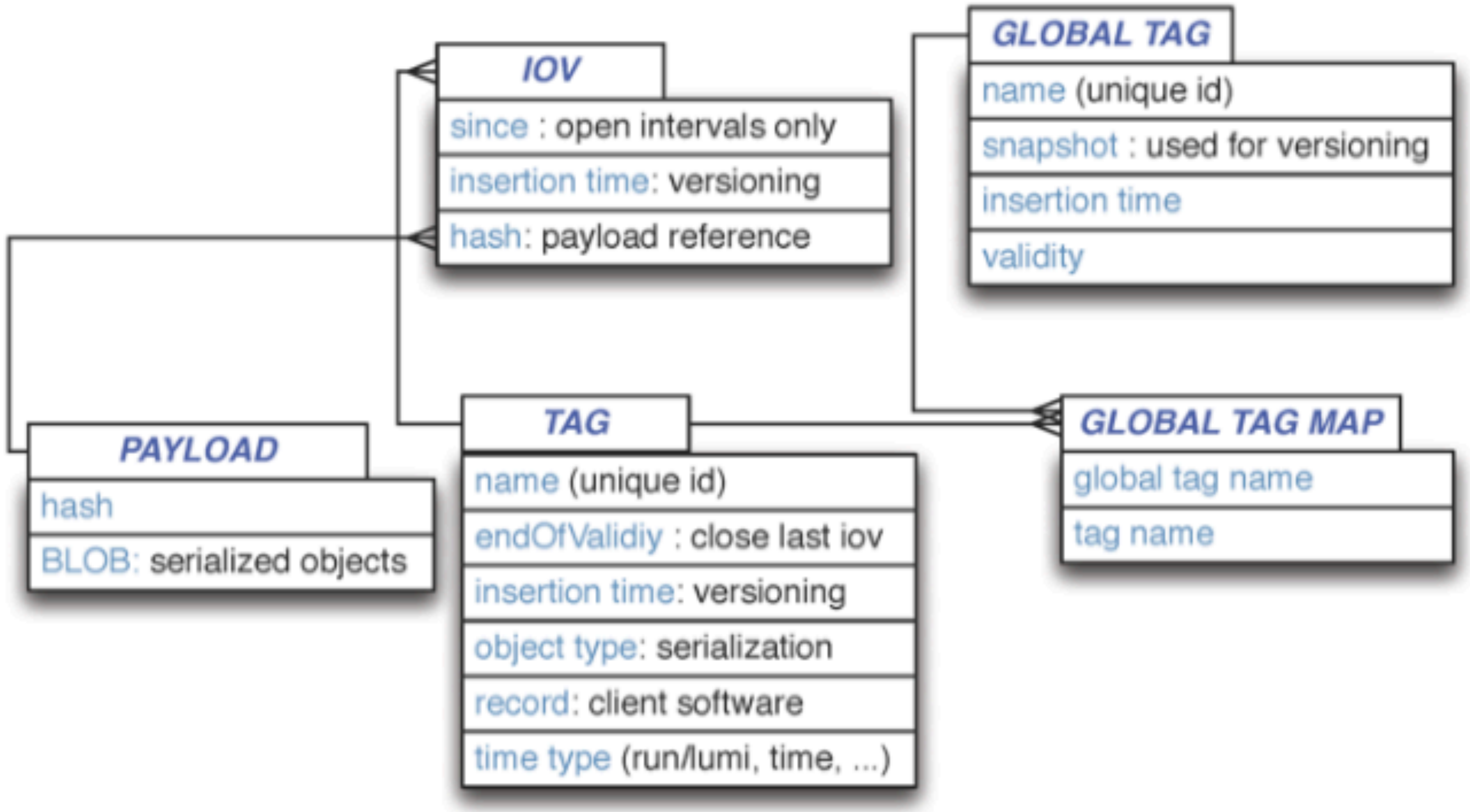
- From the HSF Conditions Data activity:

  https://hepsoftwarefoundation.org/activities/conditionsdb.html
- Key recommendations for conditions data handling
  - *Separation of payload queries from metadata queries*
  - *Schema below to enable appropriate configuration*

Both **CMS** and **Belle II** use similar (though more complex) schema today

HEP Software Foundation

HEP Software Foundation
Community White Paper Working Group – Conditions Data



IOV
since : open intervals only
insertion time: versioning
hash: payload reference

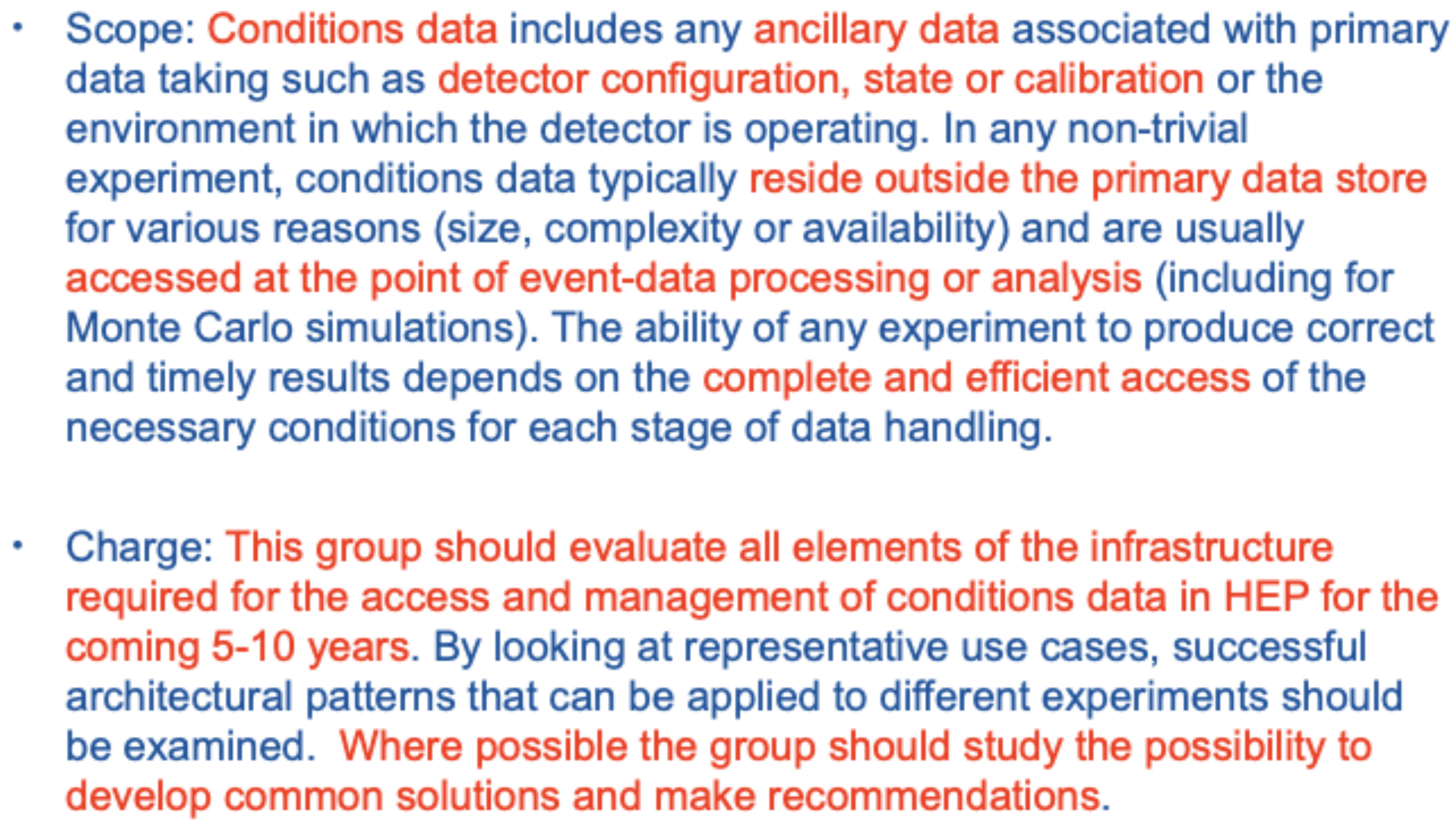
GLOBAL TAG
name (unique id)
snapshot : used for versioning
insertion time
validity

PAYLOAD
hash
BLOB: serialized objects

TAG
name (unique id)
endOfValidiy : close last iov
insertion time: versioning
object type: serialization
record: client software
time type (run/lumi, time, ...)

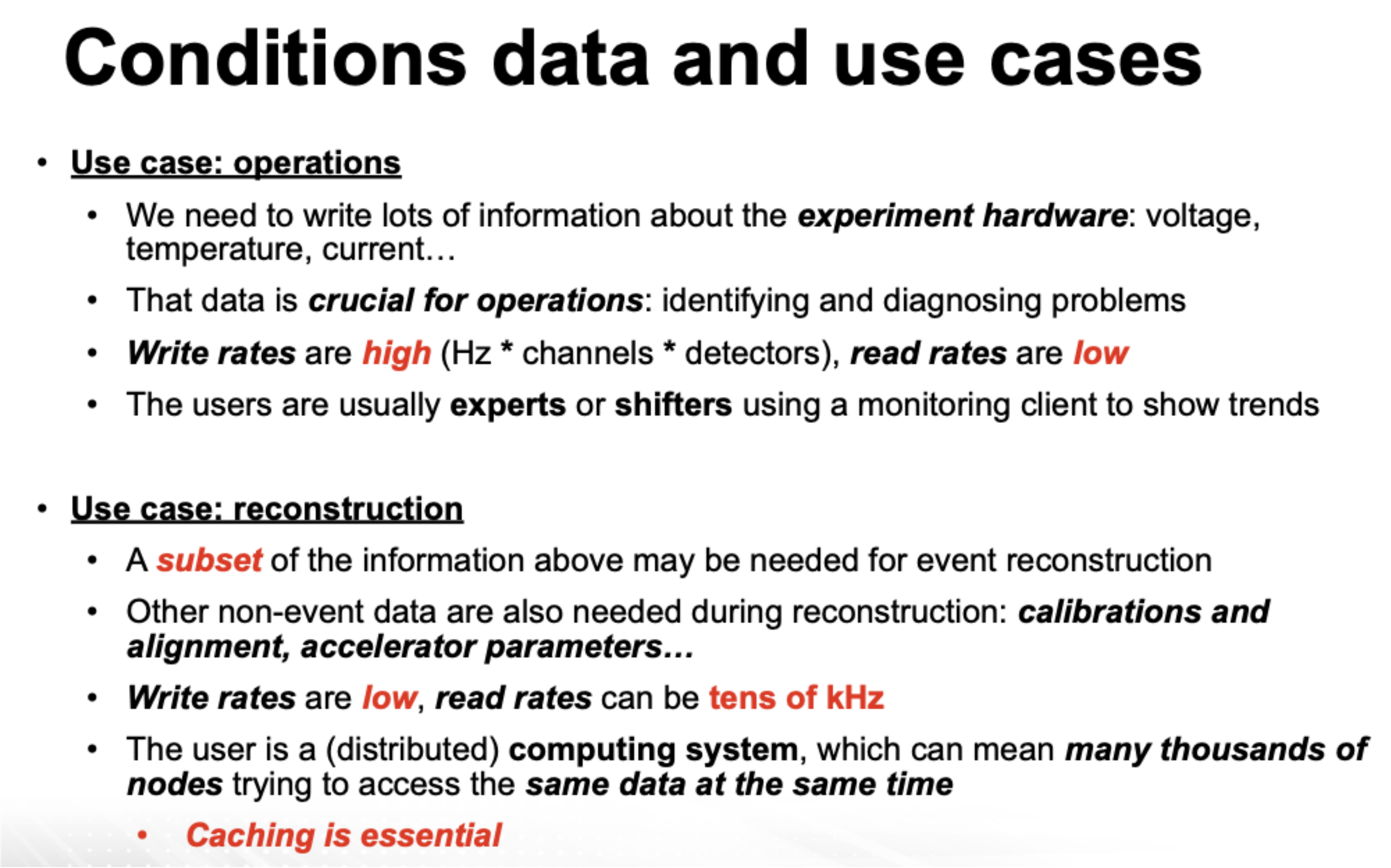
GLOBAL TAG MAP
global tag name
tag name

Conditions data according to HSF



- Scope: Conditions data includes any ancillary data associated with primary data taking such as detector configuration, state or calibration or the environment in which the detector is operating. In any non-trivial experiment, conditions data typically reside outside the primary data store for various reasons (size, complexity or availability) and are usually accessed at the point of event-data processing or analysis (including for Monte Carlo simulations). The ability of any experiment to produce correct and timely results depends on the complete and efficient access of the necessary conditions for each stage of data handling.

- Charge: This group should evaluate all elements of the infrastructure required for the access and management of conditions data in HEP for the coming 5-10 years. By looking at representative use cases, successful architectural patterns that can be applied to different experiments should be examined. Where possible the group should study the possibility to develop common solutions and make recommendations.



Conditions data and use cases

- <u>**Use case: operations**</u>
  - We need to write lots of information about the ***experiment hardware***: voltage, temperature, current…
  - That data is ***crucial for operations***: identifying and diagnosing problems
  - ***Write rates*** are ***high*** (Hz * channels * detectors), ***read rates*** are ***low***
  - The users are usually **experts** or **shifters** using a monitoring client to show trends

- <u>**Use case: reconstruction**</u>
  - A ***subset*** of the information above may be needed for event reconstruction
  - Other non-event data are also needed during reconstruction: ***calibrations and alignment, accelerator parameters…***
  - ***Write rates*** are ***low***, ***read rates*** can be **tens of kHz**
  - The user is a (distributed) **computing system**, which can mean ***many thousands of nodes*** trying to access the ***same data at the same time***
    - ***Caching is essential***

Offline generally has very little influence on the online DB design

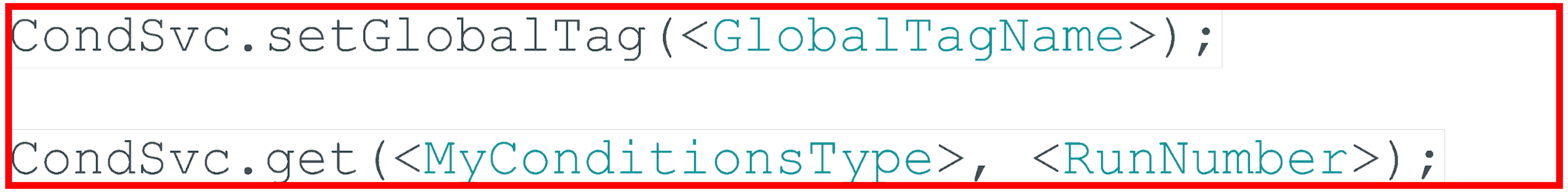
Conditions data access from software frameworks

Most people don’t have experience with, and therefore are uncomfortable working with, databases. Those creative spirits will go to great lengths to avoid interacting with a database. See e.g. the HSF Data Analysis WG’s paper on Metadata !

We want an API for frameworks that hides the database access. Just configure it, and then request your data using the minimal information possible

Intuition leads us to think of e.g. time-stamp access to conditions data, but the main “user” (at least the one that we need to worry about most) is a production system with thousands of cores running ***jobs***

- *Get your conditions data from ONE DB interface (not an ad hoc DB interface for each source)*
- *Access your data by Run number (and support sub-run granularity)*



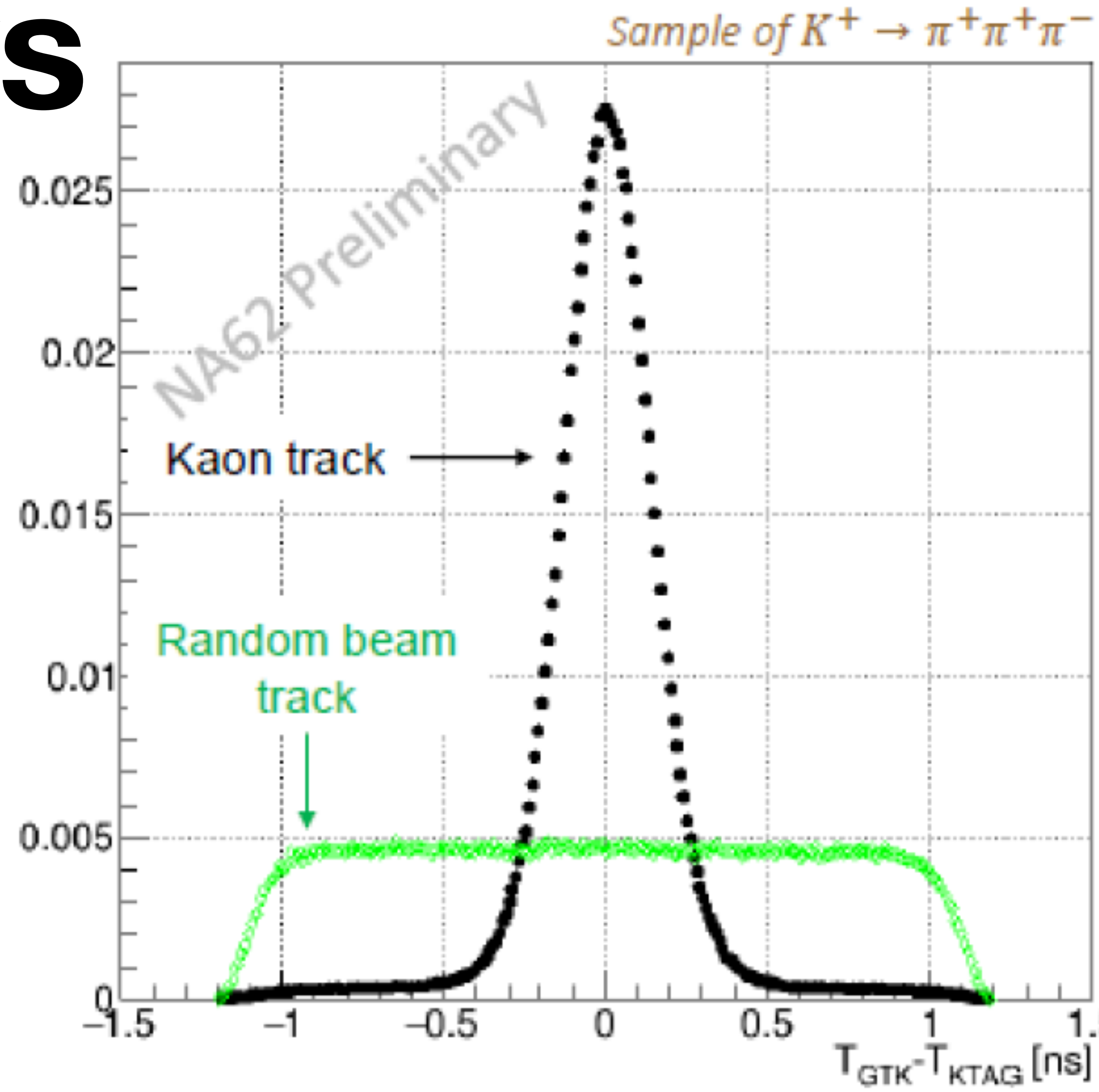
```
CondSvc.setGlobalTag(<GlobalTagName>);

CondSvc.get(<MyConditionsType>, <RunNumber>);
```

Once per job

In each module

NA62: Data processing workflows

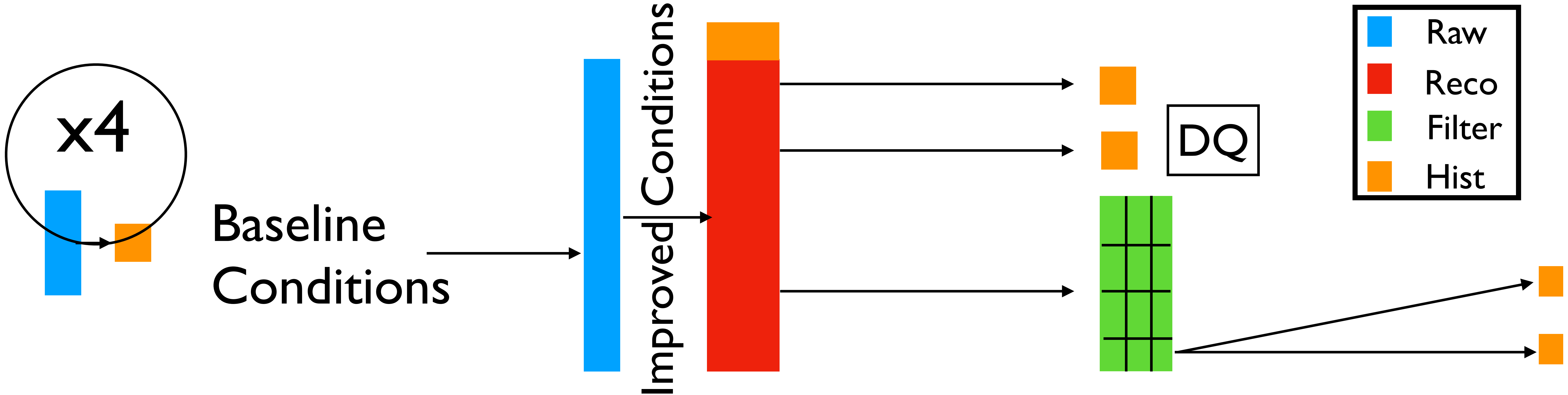


Key requirement:

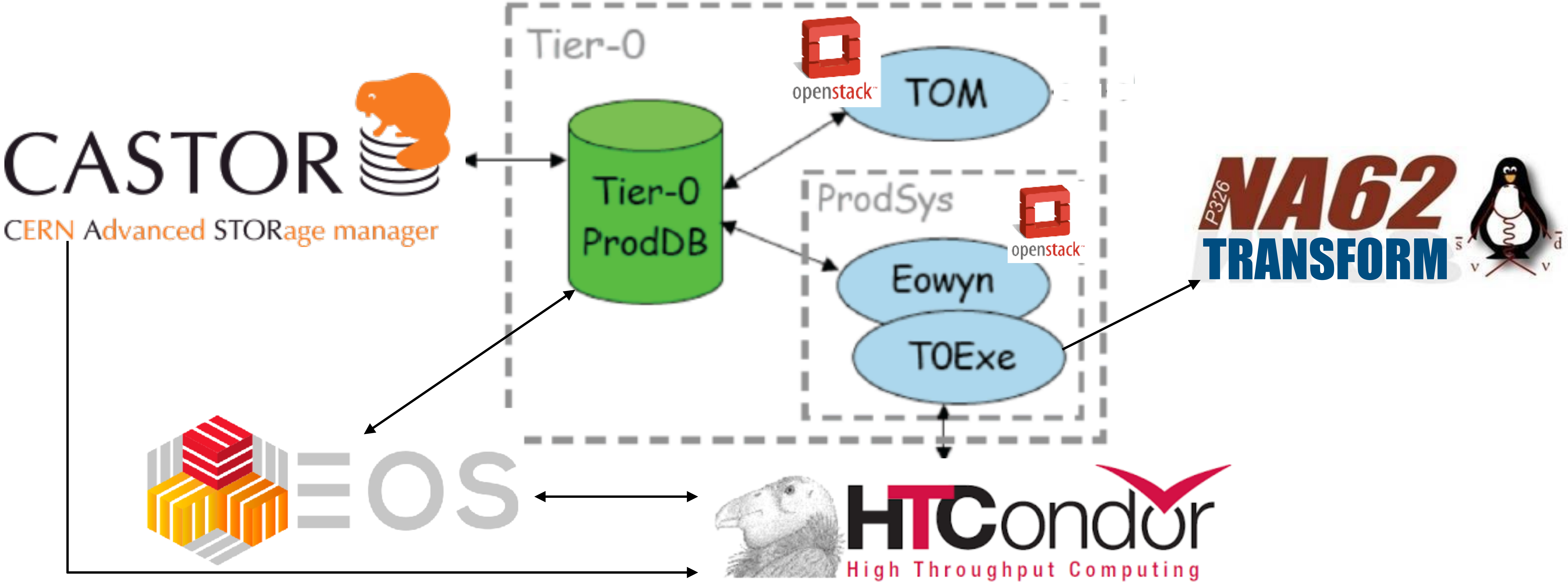
- *Average t0s aligned at the 10ps level*

Workflow:

- **Iterative calibration procedure performs coarse and then fine t0 calibration using 100 bursts, 1 DAQ run ~= 1500 bursts**
- **Burst by burst calibration for high precision detectors during full event reconstruction to achieve 10ps alignment**
- **Post-production steps for filters and data quality**



NA62 Production system



- A “Transform” knows how to turn inputs into outputs (c.f. DAGs and modern workflow languages)
    - For each step, needs to configure the software framework, including providing the conditions GlobalTag parameter (one single parameter to define all of the conditions)

Conclusions

Metadata is a complex topic covering a large range of use cases

End to end lifecycle management is a problem that needs to be solved!

Analysis tends to suffer the most

The HSF provides a very useful forum to discuss these topics

The problem itself still requires work, not least to properly define it

In the case of **Conditions Data**, there is progress toward solutions - see next talk !

Factorising configuration and conditions from code is very powerful !

#Sustainable~~Salmon~~Metadata