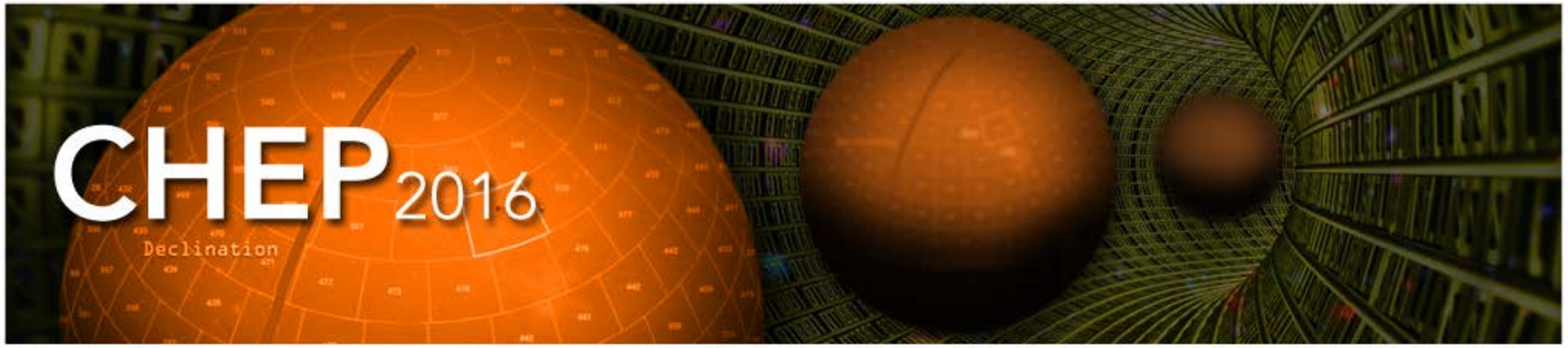
ATLAS EventIndex General Dataflow and Monitoring Infrastructure

D. Barberis, A. Favareto, **Álvaro Fernández Casaní**,

C. García Montoro, S. González de la Hoz, J. Hřivnáč,

F. Prokoshin, J. Salt Cairols, J. Sánchez, R. Töebbicke, R.J.Yuan

on behalf of the ATLAS Collaboration.

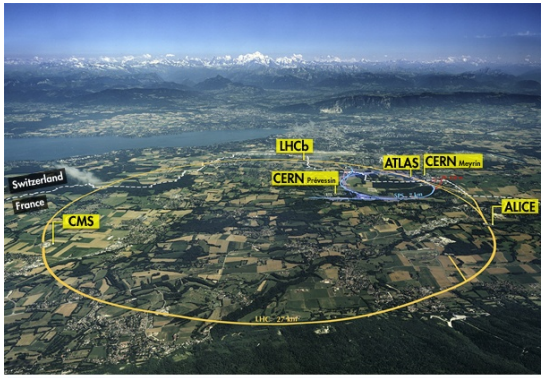


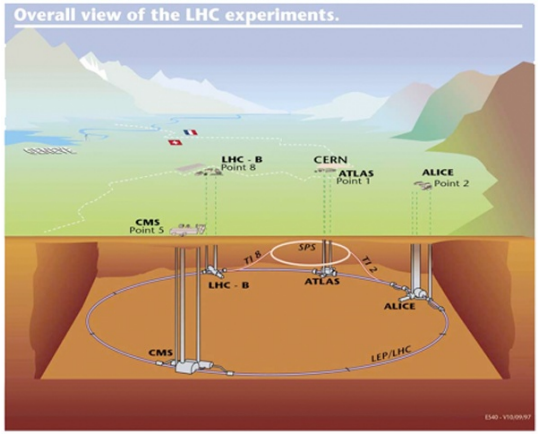
22nd International Conference on Computing in High Energy and Nuclear Physics, Hosted by SLAC and LBNL, Fall 2016

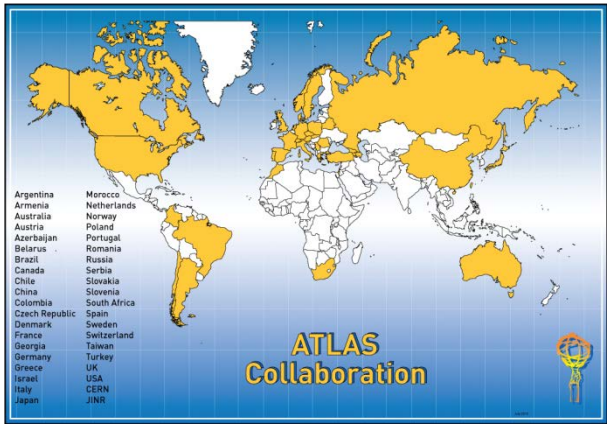
Outline

- ATLAS experiment at CERN
- ATLAS EventIndex
  - Use cases
  - General architecture
- Distributed Data Collection Architecture
  - Current Production Version (Messaging based)
  - Data Collection Monitoring
  - New Development Prototype (Object Store based)
- Summary

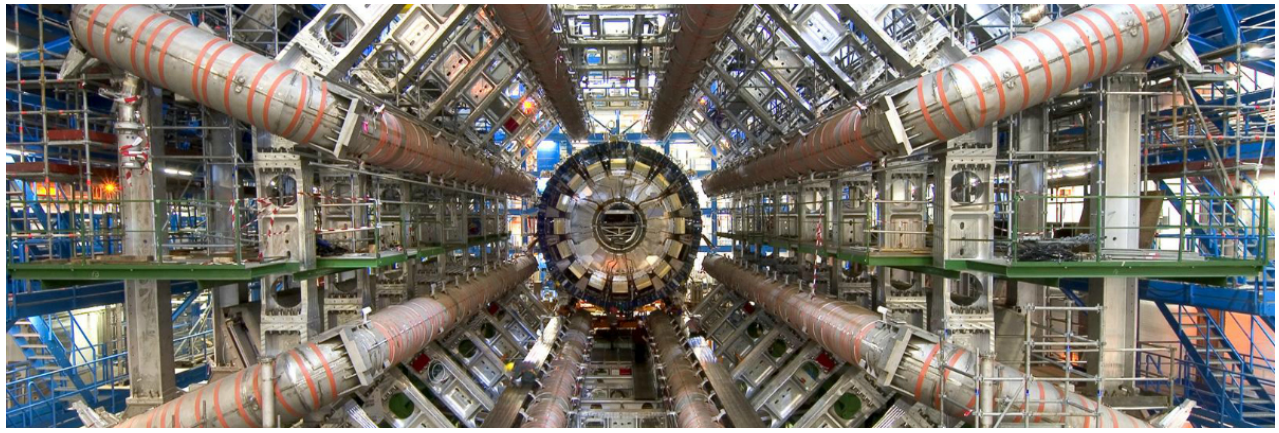
ATLAS experiment at CERN



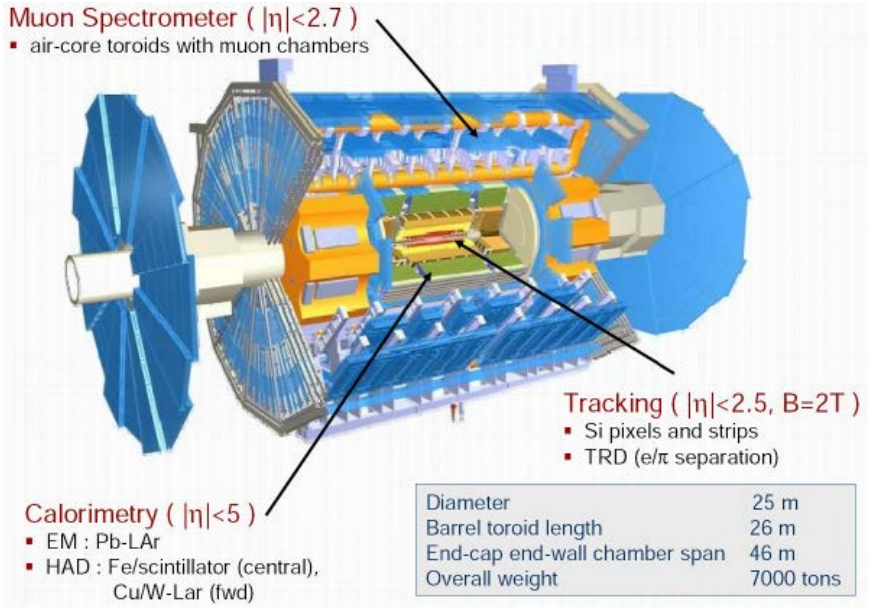


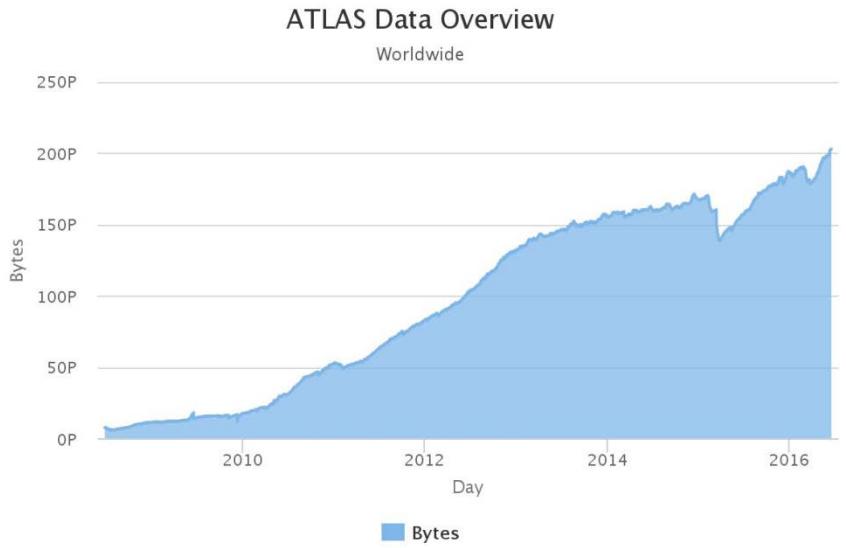


- **Large Hadron Collider (LHC)** is a particle accelerator located at CERN in the border of Switzerland and France.
- The circular tunnel has a length of 27 km, and is 175 meters below ground.
- **ATLAS** is one the 4 big detectors, devoted to test the predictions of the Standard Model, that lead to the discovery Higgs boson in 2012, and to physics beyond the Standard Model and the development of new theories to better describe our universe.
- ATLAS experiment is a **collaboration** of 5000 scientists from about 180 institutions around the world, representing 38 countries.



ATLAS EventIndex Challenges





2016 milestone (run 2): 200 Petabytes

- The **ATLAS experiment** at CERN has produced, just last year 2015, **12 Billion real events** in different processing versions in **1 million files**, and **5 Billion simulated events in 8 million files**.
- **EventIndex** is in production since mid-2015, **reliably collecting information worldwide about all produced events** and storing them using a central Hadoop infrastructure at CERN. A subset of this information is copied to an Oracle relational database for some use-cases.
- **This talk describes the general architecture of the project and the data flow and operation issues, which are being addressed by recent developments to improve the throughput of the overall system**.

Use Cases

1) Event picking: users able to select single events depending on constraints. Order of hundreds of concurrent users, with requests ranging from 1 event (common case) to 30k events (occasional).

2) Production consistency checks

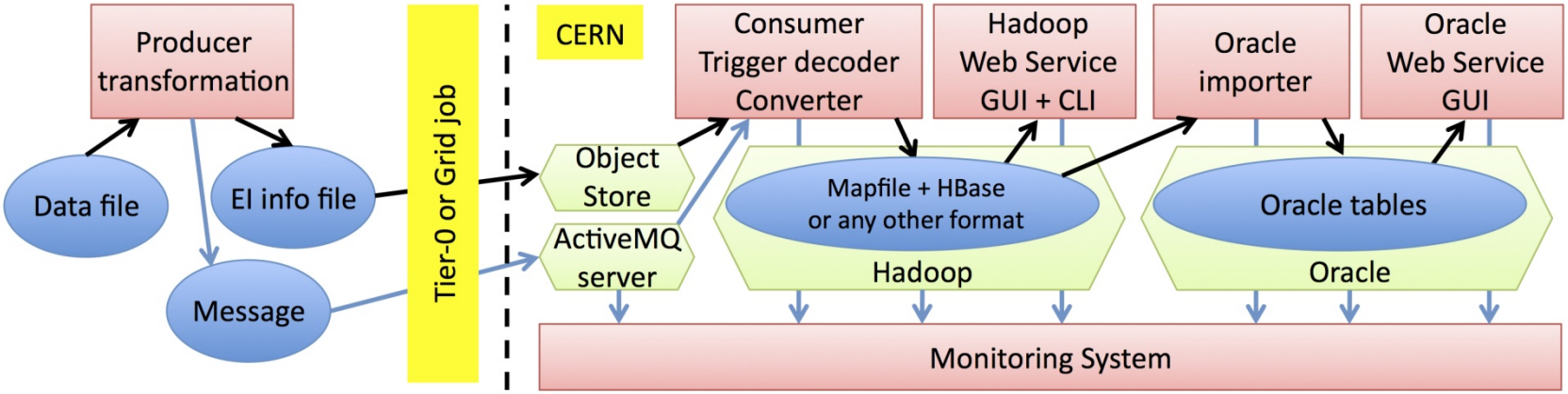
- **Duplicate event checkings**: events with same Id appearing in same or different files/datasets.
- **Overlap detection** in derivation framework: construct the overlap matrix identifying common events across the different files.

3) Trigger checks and event skimming: Count or give an event list based on trigger selection.

- **Trigger Overlap detection:** number of events in a real data Run/Stream satisfying trigger X which also satisfies trigger Y.

- Current cross-checkings helped detecting ATLAS produced data problems:
  - ie: Detected incorrect RAW provenance values
- Duplication checkings and overlaps are used to check data on a common basis.

EventIndex Architecture



1) Distributed Data Collection of all the produced events, from CERN and hundreds of Grid sites worldwide, in an a reliable and efficient manner. Design adoptions:

- A Producer/Consumer architecture. Producers run at distributed sites, and Consumers at Cern.
- Messaging infrastructure to convey EventIndex information(Json encoded)
- Consumers write information into Backend Storage

2) Backend storage and processing, using **Hadoop** technologies for efficiently storing all indexed data. A subset is copied into **Oracle** for accessing. (see contribution 'An Oracle-based EventIndex for ATLAS' in this conference)

3) Query services for final users.

4) Functional Monitoring infrastructure

Message Flow

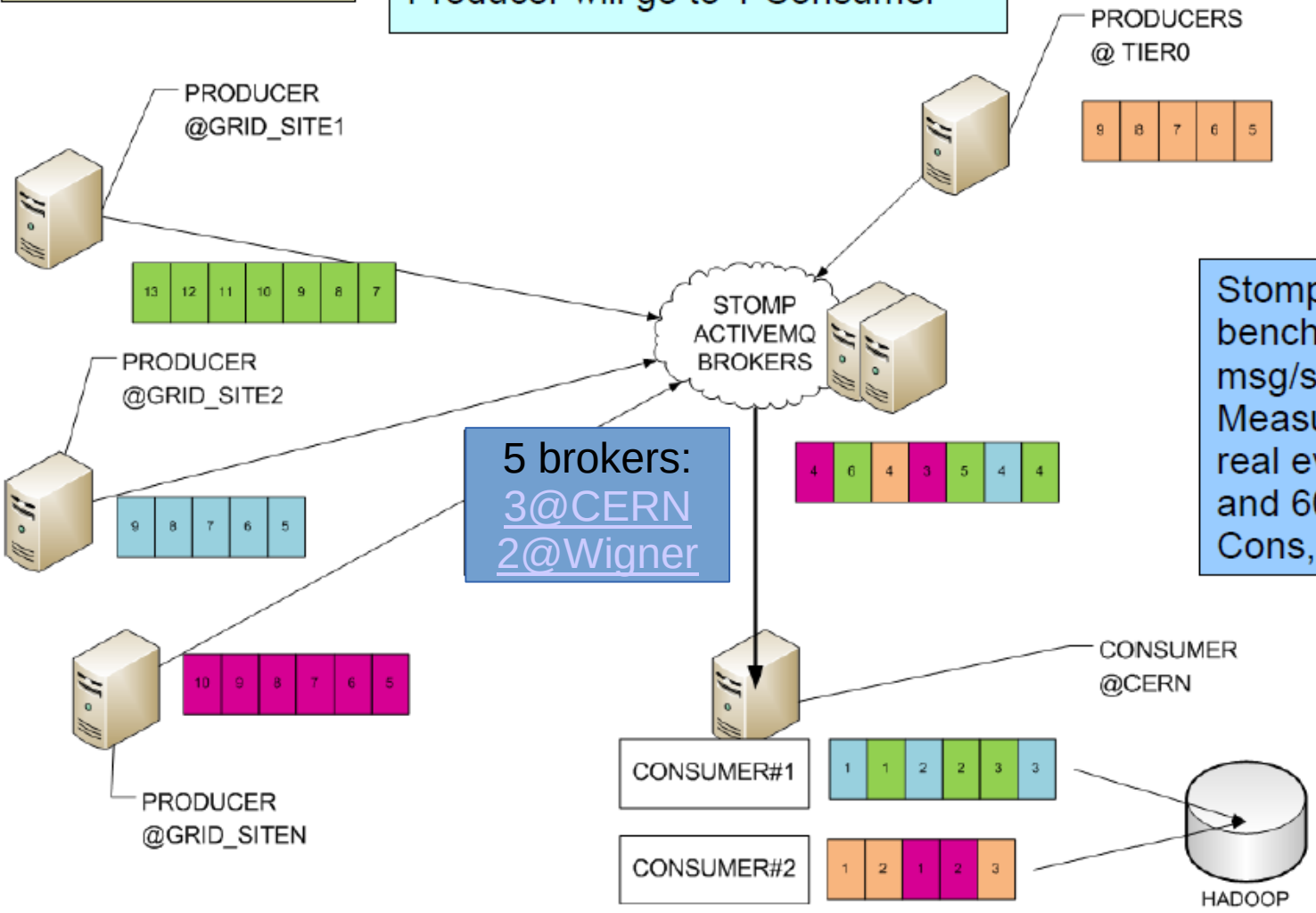
Message size is set small 1-10kB to keep broker queues agile.

Producers tag messages by group (JMSXGroupID)

Ensures that all messages from 1 Producer will go to 1 Consumer

Atomic transactions on Producers:
if connections breaks no partial processing occurs.

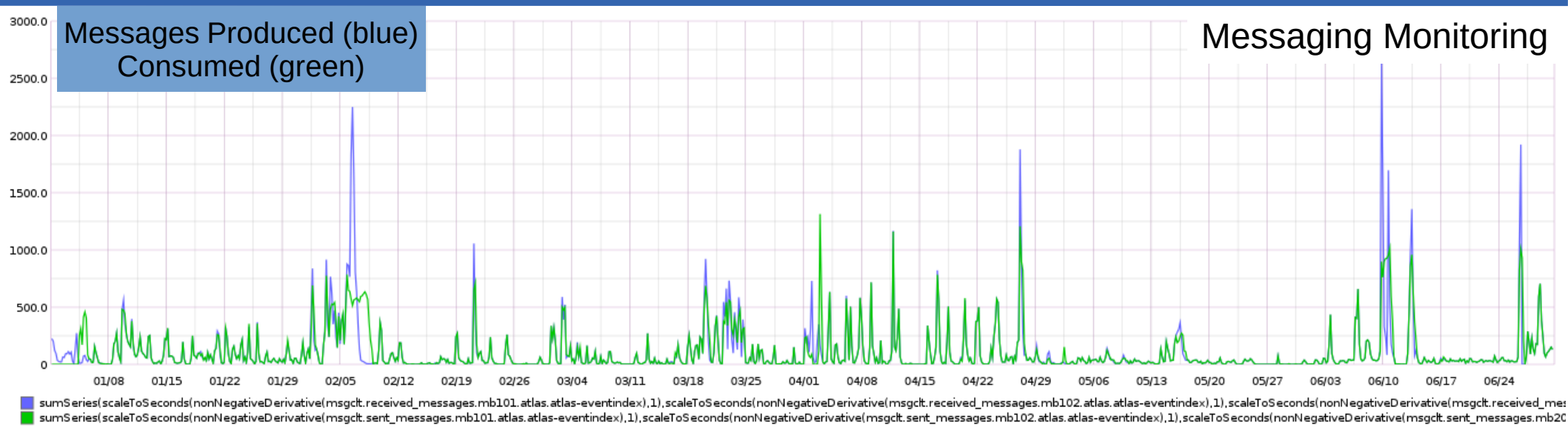
Status messages are sent from producers and consumers to an alternate queue.



Stomp performance in our benchmarks reaches over 350 msg/s and 10Mb/s per Producer. Measured performance for sending real events reached 200K event/s and 60Mb/s (1Broker, 6 Prod/s, 4 Cons, 50K events/job)

Files stored on Mapfile format usable by Hadoop Core Services

Data Collection Monitoring

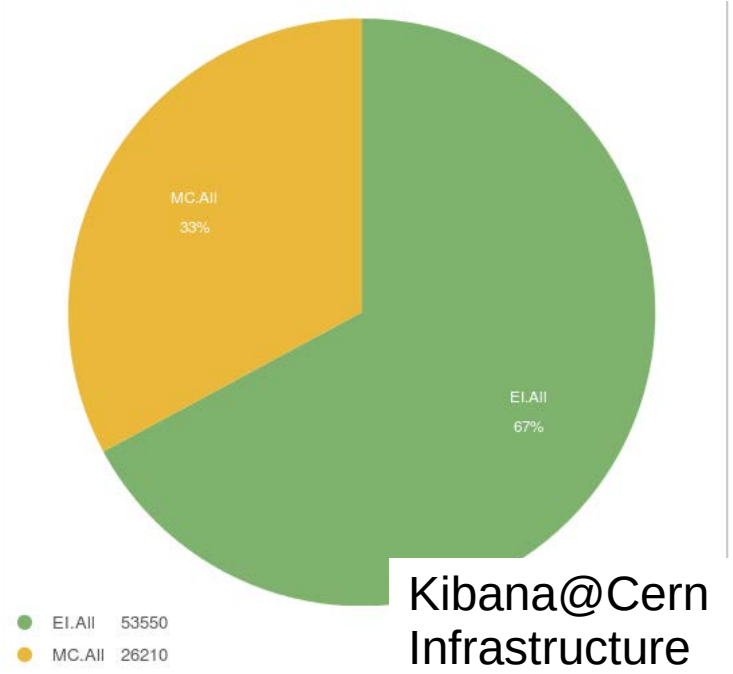


- Messaging Statistics

- 5 Brokers. 3 long-lived Consumers per broker. Varying number of simultaneous short-lived Producers.
- More than 10^9 messages handled.
- Usual rate of 100 msg/s produced. Peaks of >3500 msg/s, not consumed instantaneously (possible future congestion problems)

- Current EventIndex Data in Hadoop:

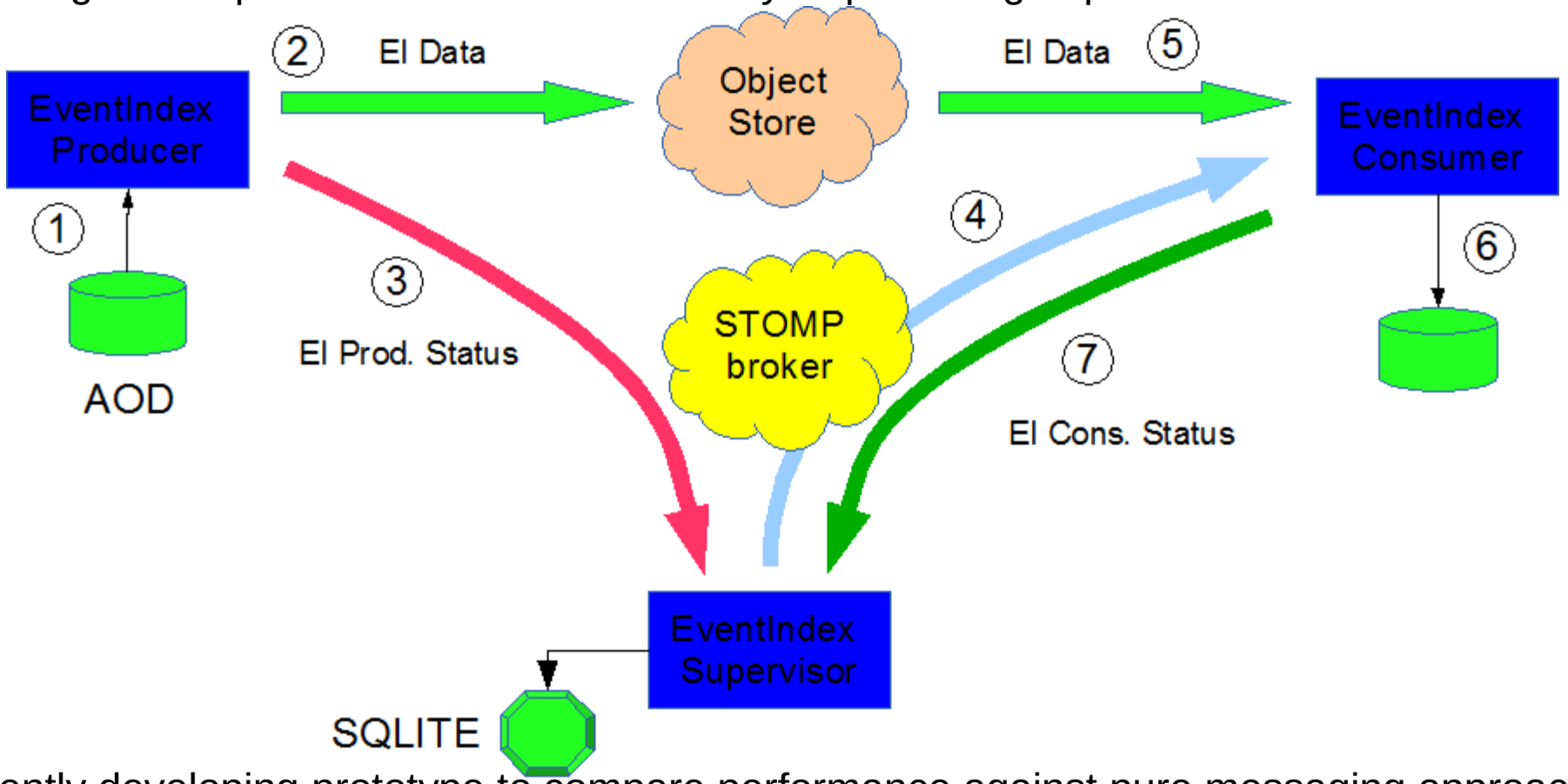
- 81 TB of events data
  - 50 TB real data (1/3 2016, 1/3 2015, 1/3 previous years)
  - 31 TB MonteCarlo simulated data (2015 mainly)
- Plus archive and backup off all data



Kibana@Cern Infrastructure

New development: ObjectStore Dataflow

- Performance of data collection is satisfactory, but in the future there might be peaks of production that we want to digest better.
- Exploring alternatives for the data flow: conveying EventIndex data with a **ObjectStore.**
- Redesign other parts of the flow that currently require a high operation effort.

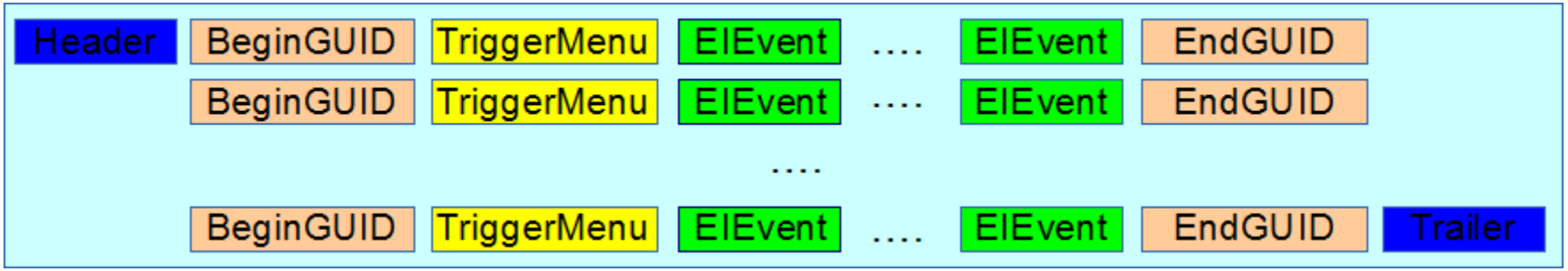


Currently developing prototype to compare performance against pure messaging approach:

- The object store is used to transport EI data (Substitutes messaging to convey data)
- Communication between Producer-Supervisor-Consumer uses the STOMP broker

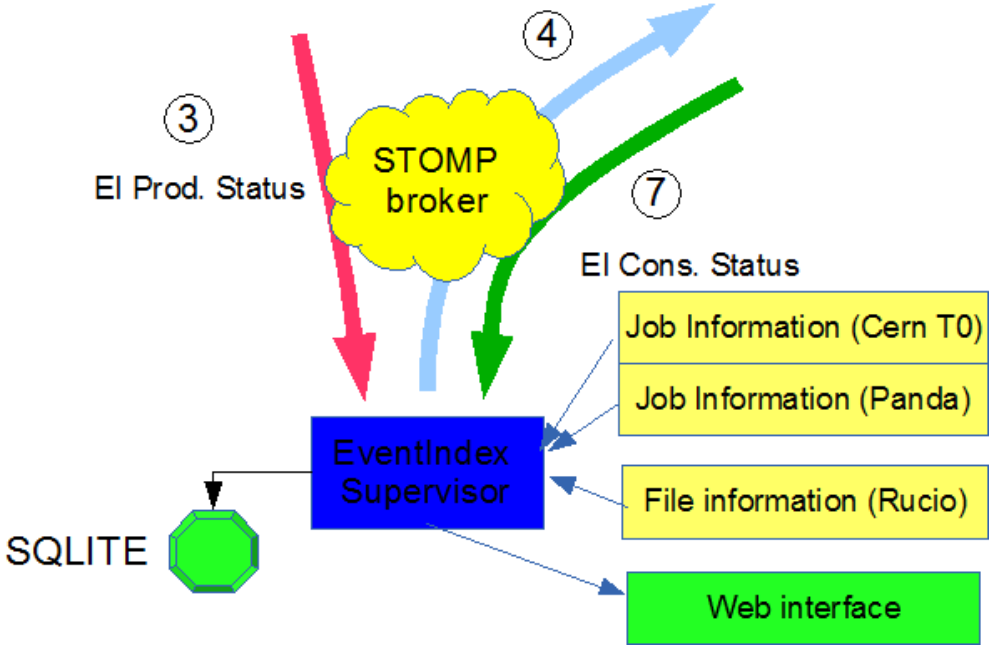
EventIndex Producer

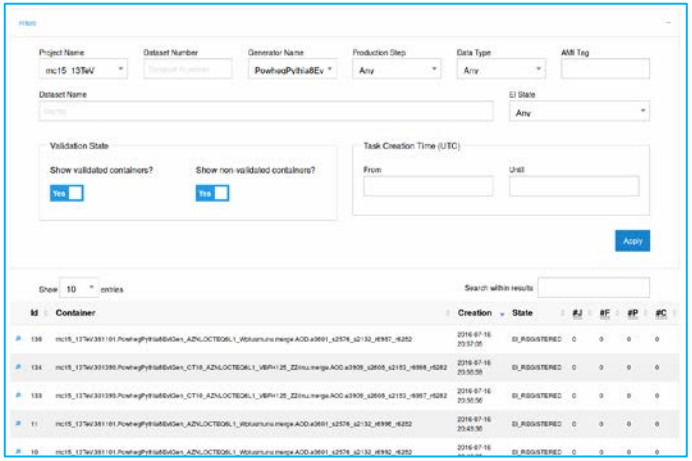
- In charge of extracting the required information for the indexed (AOD) files.
- Runs at Tier0 at Cern, and wordlwide grid jobs.
- Encode the information with new Google Protobuf format. (Previous json encoding available for compatibility).
- Stores the information at Cern ObjectStore (Ceph), and submits a control message to be received by the Eventindex Supervisor



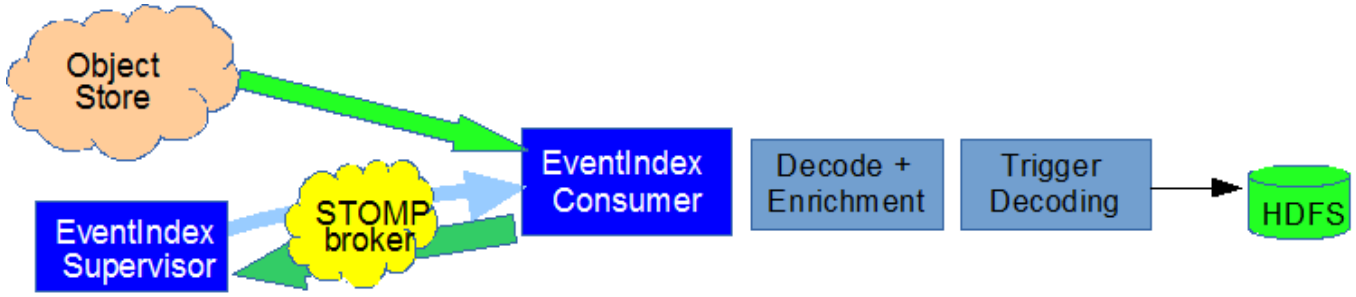
EventIndex Supervisor

- Supervises the EventIndex data collection at all stages.
- Receives status information about what EventIndex data has been produced (3).
- Consults external sources about jobs running at Tier0 and worldwide (Panda manage system). Also about the files that belong to a dataset (Rucio).
- A dataset comprises several produced files. When a dataset can be validated submits this validation information to Consumer (4). Later Consumer will confirm that dataset has been processed(7)
- The information of all the stages and process can be followed via the Web Interface .





EventIndex Consumer



- Consumer receives validation messages from the EventIndex Validator.
- **Retrieves EventIndex information from ObjectStore** .
  - Moves from previous Push to Pull model (only retrieving validated information from ObjectStore)
  - Using amazonaws s3 java packages
  - Streams data directly, and decompressed on the fly.

- **Data decoding and enrichment chain**
  - Decoding of all new Protocol buffer format provides more information.
  - Data enrichment: simplifies cataloguing by Hadoop Core task.
  - Trigger decoding: enriched protobuf information able to decode trigger. (In development).
- **Stores information in Hadoop/HDFS**
  - Effectively reduces number of hdfs files from previous model (1 per GUID) to 1 per validated dataset

Summary

- Current **EventIndex** dataflow based on a messaging system **runs in production since mid-2015**, successfully collecting millions of messages and thousands of datasets.
- **Monitoring is a key operation.** Current functional operation of the production servers is based on CERN Kibana infrastructure. Operational monitoring and **corrective actions are done by developers and experts**.
- **Next generation prototype based on a transient Object Store is on development**, envisaged to cope with future production rates, and to improve and automate operations.