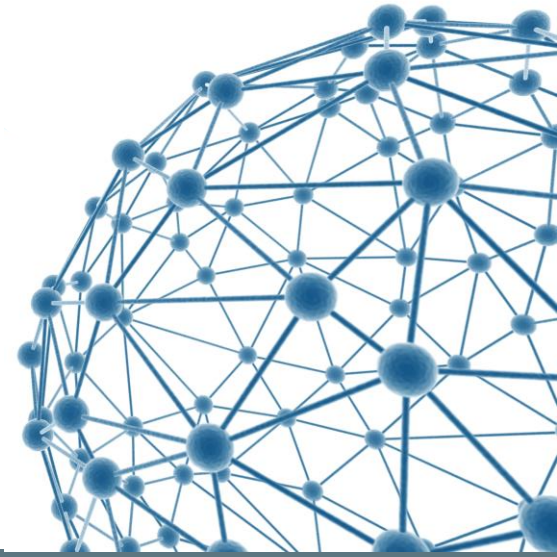
HL-LHC computing model
(just some brainstorming ideas)

Simone Campana, Torre Wenaus

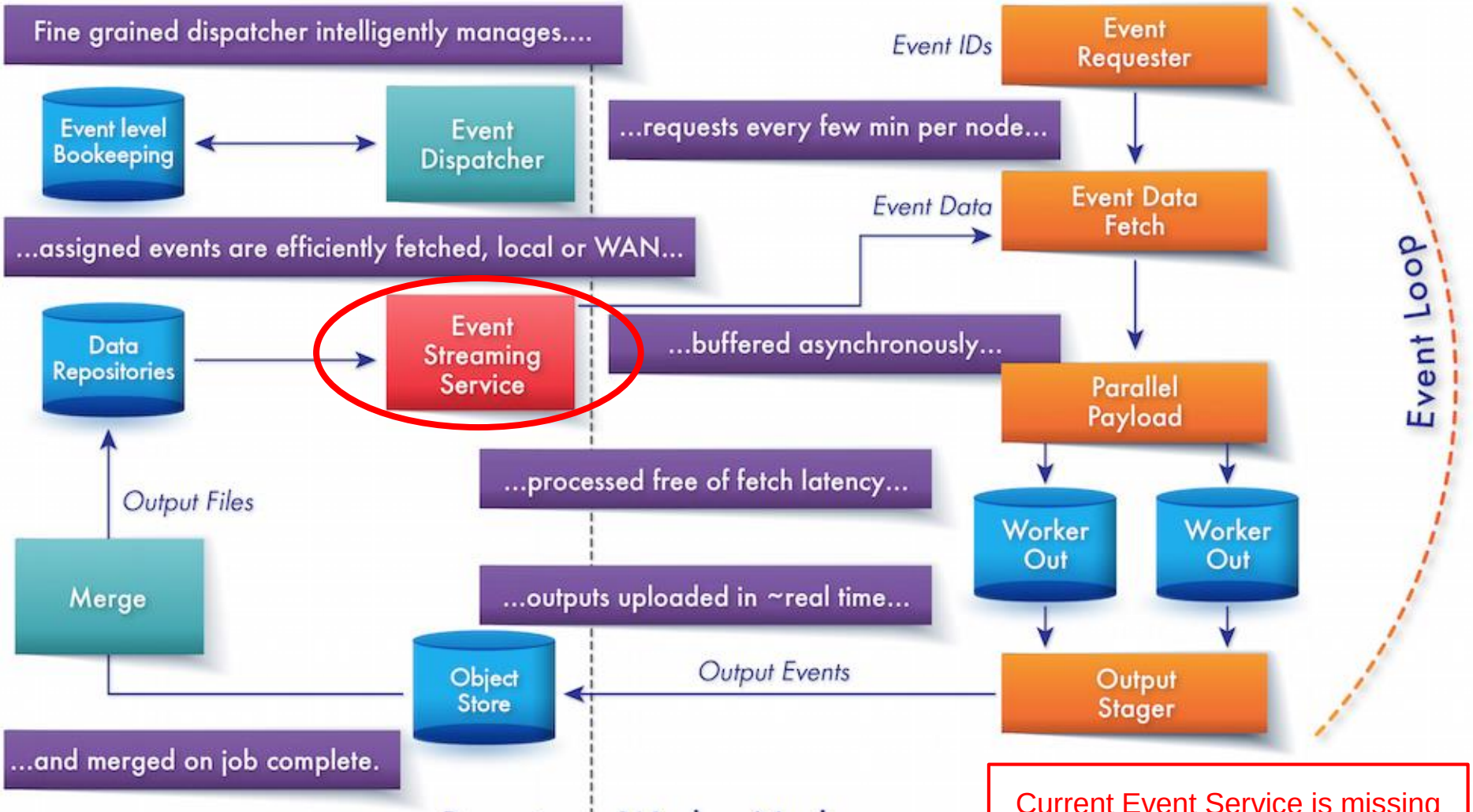


Observations

- Our computing model is still 20 years old
  - Lots of changes, like flattening hierarchy, abandoning the monarch model …
  - But we still have hierarchical storage, a lot of deterministic data placement, etc..
- Our use case is data driven. The challengs is the data, how you organize, it, store it, serve it to a CPU core
- Our data is generally “cold”
  - We store 100PB, we access 1EB
  - Accessing 10 times each file in a collaboration of 4k people is not a lot
- Our quantum is the event
  - We organize events in file because file systems talk “files”
  - We organize files in datasets for scalability
  - But in many respects you have advantages if you can manage the lowest possible granularity

Event Service Schematic

Schematic is in the abstract; implementations realizing the architecture are in green



PanDA/JEDI, Object Stores

PanDA pilot, Yoda, AthenaMP payload

Current Event Service is missing its data flow component, the Event Streaming Service

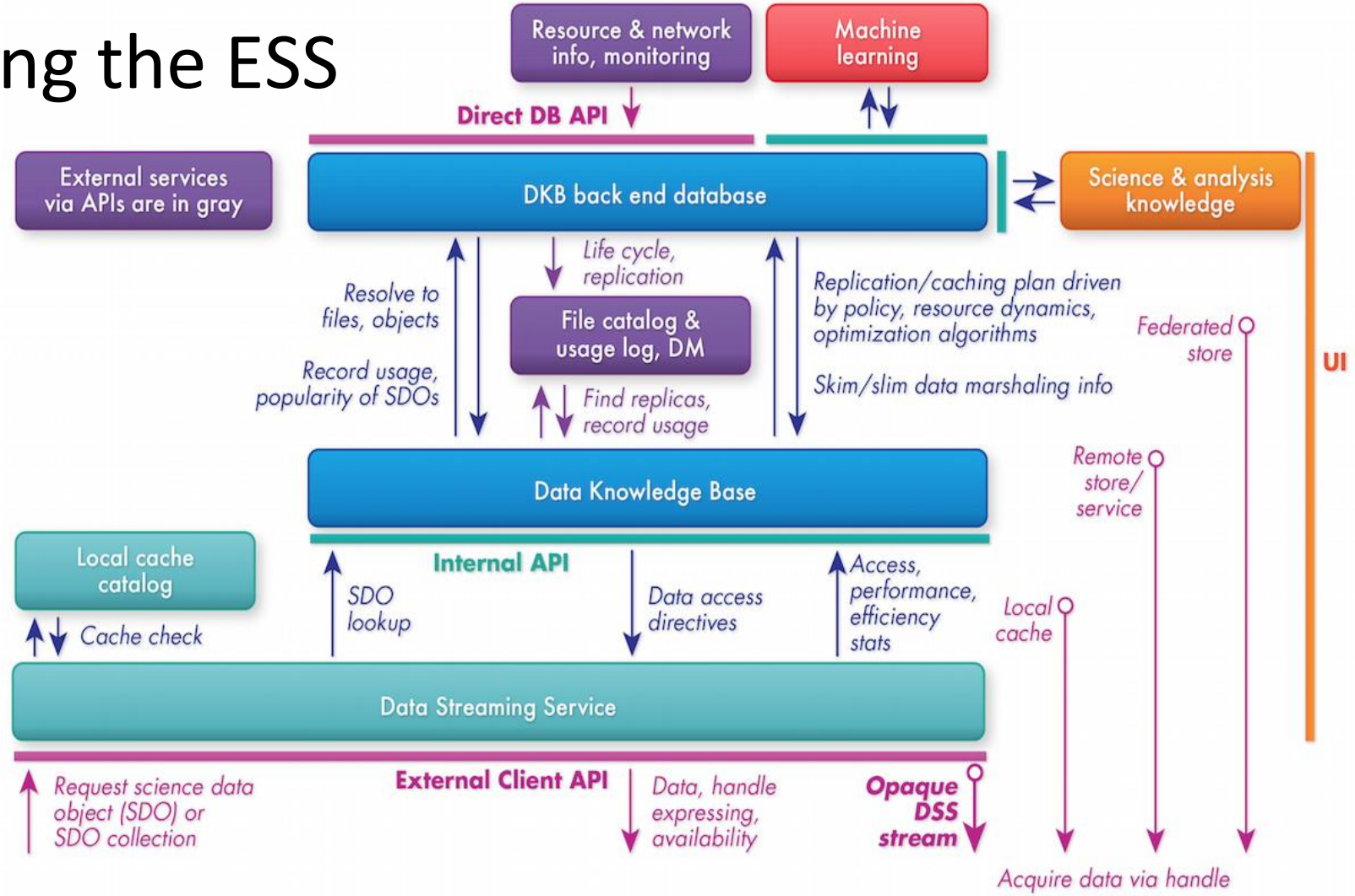
BROOKHAVEN NATIONAL LABORATORY

BERKELEY LAB

Event Streaming Service (ESS)

- Event Service today reads input data at the worker level
- Can be local or remote WAN
- If it's remote WAN the processing suffers the WAN latency
- TTreeCache asynchronous prefetch can ameliorate this
  - If it works well (we've been trying it for a long time)
  - But even if it does, it couples WAN remote read and its many potential failure modes to the application
  - Making problem isolation and debugging more difficult
  - It also takes away our ability to pre-fetch how we want
- The ESS is an in-house means of implementing asynchronous prefetch
- Analogous to how outputs are handled, the pilot runs a data prefetcher process asynchronous from the payload
- As the pilot receives event range assignments, the prefetcher performs WAN (or local) reads to fetch the data, creating small input files corresponding to event ranges
- The input files are consumed by the payload

Building the ESS



Two primary components: Data Streaming Service

- CDN-like intelligence in efficient data delivery
- With minimal replication
- Data marshaling
- Smart local caching

Informed by the **Data Knowledge Base** providing the intelligence on

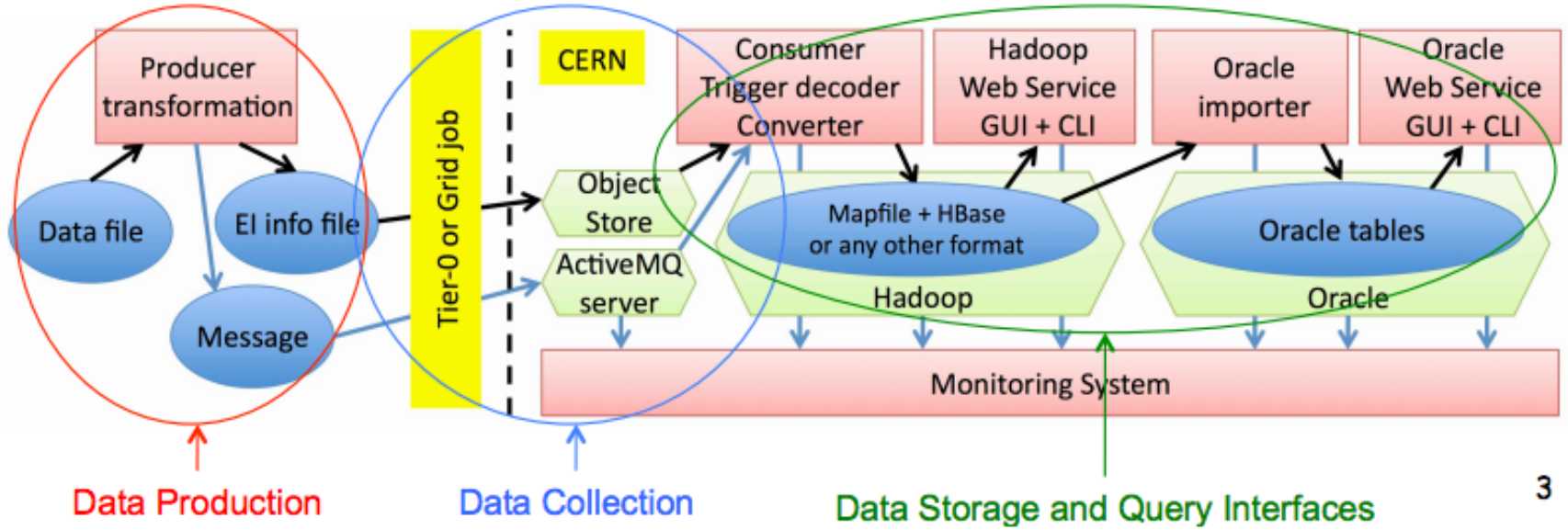
- Dynamic resource landscape
- Science data object (SDO) knowledge
- Analysis processes & priorities
- ML-derived predictive knowledge

EventIndex Use Cases

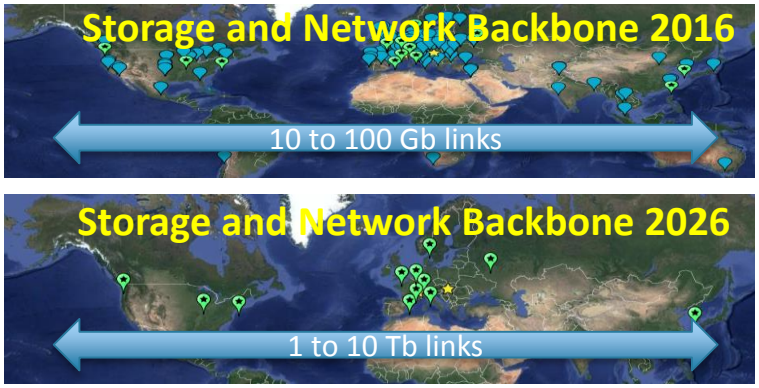
- A system designed to be a complete catalogue of ATLAS events, real and simulated data
- Main use cases:
  - Event picking (give me this event in that format and processing version)
  - **Count and select events based on Trigger decisions**
  - Production completeness and consistency checks (data corruption, missing and/or duplicated events)
  - **Trigger chain overlap counting**
  - Derivation overlap counting
- Summary of Use Cases here:
  https://twiki.cern.ch/twiki/bin/view/AtlasComputing/EventIndexUseCases
- Contents:
  - Event identifiers (run and event numbers, trigger stream, luminosity block, BCID)
  - **Trigger decisions**
  - References (GUID plus internal pointer) to the events at each processing stage in all permanent files generated by central productions (for event picking)
    - [RAW], [ESD], AOD, (DAOD) for real events
    - EVNT, [RDO], [ESD], AOD, (DAOD) for simulated events

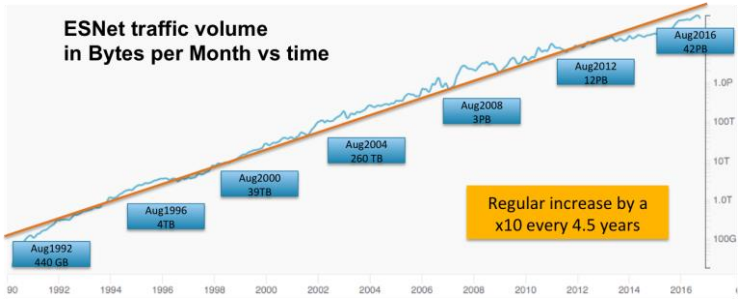
EventIndex Architecture

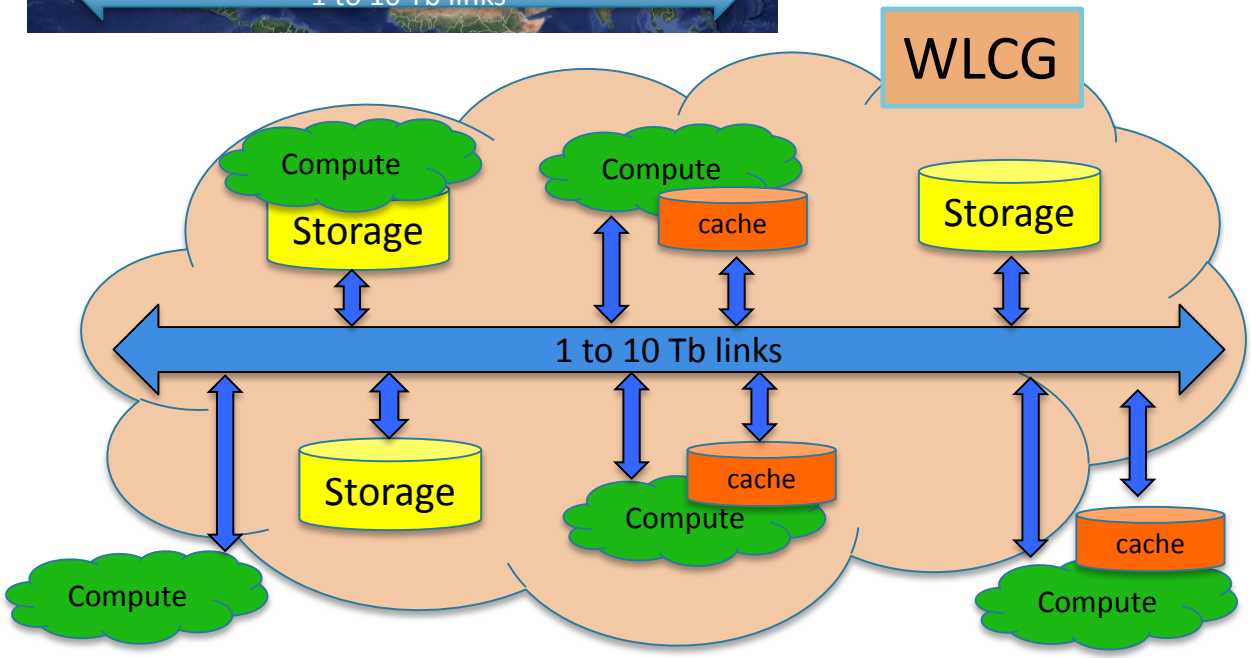
- Partitioned architecture, following the data flow
- Data Production: extract event metadata from files produced at Tier-0 or on the Grid
- Data Collection: transfer EventIndex information from jobs to the central servers at CERN
- Data Storage: provide permanent storage for EventIndex data and fast access for the most common queries + finite-time response for complex queries
  - Full info in Hadoop; reduced info (only real data, no trigger) in Oracle for faster queries
- Monitoring: keep track of the health of servers and the data flow



Computing infrastructure in HL-LHC







A data cloud for science

Storage and Compute loosely coupled but connected through a fast network

Heterogeneous Computing facilities (Grid/Cloud/HPC/ …) both in and outside the cloud

Different centers with different capabilities, fo different use cases

Industry?

- HEP does not drive industry
- It will be vital to understand which directions industry and commercial companies will be taking
- For example, may be tape is dead, but people will still need to archive ZettaBytes of stuff
- Same is true for many other aspects
- Openlab seems to me the right initiative: discussion with industry, early adoption of solutions, prototyping