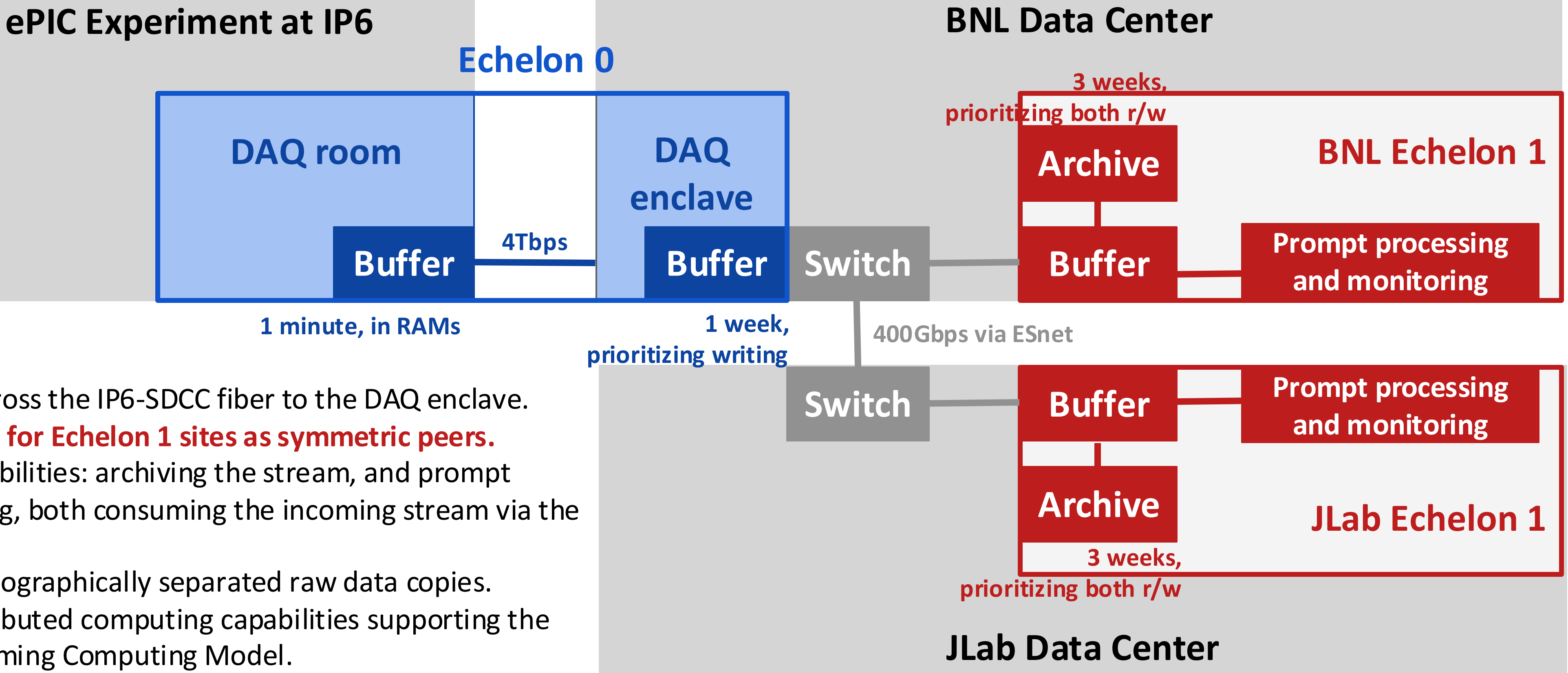
Data Transfer Between Echelon 0 and Echelon 1

Buffers in DAQ and **Echelon 1** sites ensure latency tolerance to avoid deadtime, smooth streaming operation and robustness against data flow interruptions.



- Echelon 0 extends across the IP6-SDCC fiber to the DAQ enclave.
- **Deliver the capability for Echelon 1 sites as symmetric peers.**
- Foremost E1 responsibilities: archiving the stream, and prompt processing/monitoring, both consuming the incoming stream via the buffer:
  - Delivers two geographically separated raw data copies.
  - Uses ePIC distributed computing capabilities supporting the E0/1/2/3 Streaming Computing Model.
  - Will be up to the ePIC collaboration together with sites to determine the E1 roles in detail.

What Data is Being Transferred Between Echelon 0 and 1?

- The **definition of raw data** is up to ePIC and to-be-defined in the ePIC Data Management Plan.
- It is the **data that flows**, during data taking, **out of the Streaming DAQ** (Echelon 0) to the switch and is distributed to the two Echelon 1 sites:
  - When ePIC starts, and for at least *n* years, the data stream arriving from the Streaming DAQ will be archived in full, untouched.
  - In year *n+1* ePIC may be confident enough to do immediate processing at the two Echelon 1 sites to reduce the data before archiving, i.e. archive only events of interest.
  - Regardless, our responsibility is to deliver a system designed to archive 100% of the Echelon 0 stream.
- There will be other data in the stream with continuous relevance during accelerator and detector operations that will stream continuously but is not part of the raw data, e.g. slow control monitoring or collider- experiment feedback that will be archived in databases.

- **Data reduction in the Streaming DAQ will be strictly limited**:
  - From the beginning, irreversible data reductions will be recorded for event subsamples, to develop and debug.

- The event data stream is in the form of **timeframes of 0.6ms** (defined by 2^{16} cycles of the EIC Clock):
  - DAQ inserts file and run markers into the stream.
  - It is files that hit the switch, the full dataset delivered identically to the two Echelon 1 sites.
  - Time frames will be aggregated into **supertimeframes**, **more details on next slide.**

Supertimeframe (STF) Concept (Details from our Meeting Notes)

- **STFs aggregate time-ordered timeframes (TFs)** such that processing can be based on non-overlapping STFs.
  - STFs should be long enough for losses at the edges to be negligible, e.g. ~1000 TFs, corresponding to .6sec, ~2GB.
  - There is no compelling reason to require time-ordering of STFs in a file. So we do not impose this requirement. Relaxing the ordering gives more flexibility for distributed processing.
  - We also allow TF transfer as well as STF, to support low-latency transfer of a partial raw data stream for monitoring/calibration.
- Files should be identical at the two E1s (as in the scheme discussed up to now, with DAQ inserting file markers),
- STFs are the bookkeeping unit. An STF-level bookkeeping/metadata database will be vital.

- **Intend to manage data at the STF level rather than the file level, with an object store (OS) based interface to E1 storage.**
- Allows us to work at the level we bookkeep at and care about, the STFs, without worrying about files
- Allows each facility to optimize how it maps OS content to mass storage, respecting the directives/hints that we provide (e.g. keep data associated with a run together).
- If we did pursue an OS based scheme, a file based fallback should also be supported.
- STF will have a metadata header for convenient access to info needed for STF processing.
  - Header info in the object, or external? E.g.. for Rucio, can assign user defined metadata to a Rucio object. Rucio records checksum in its system-level metadata.
- **STF as primary data object can map onto the Rucio entity. Rucio support seems good fit, and scalable.**