

ProtoDUNE II Data Management Milestones

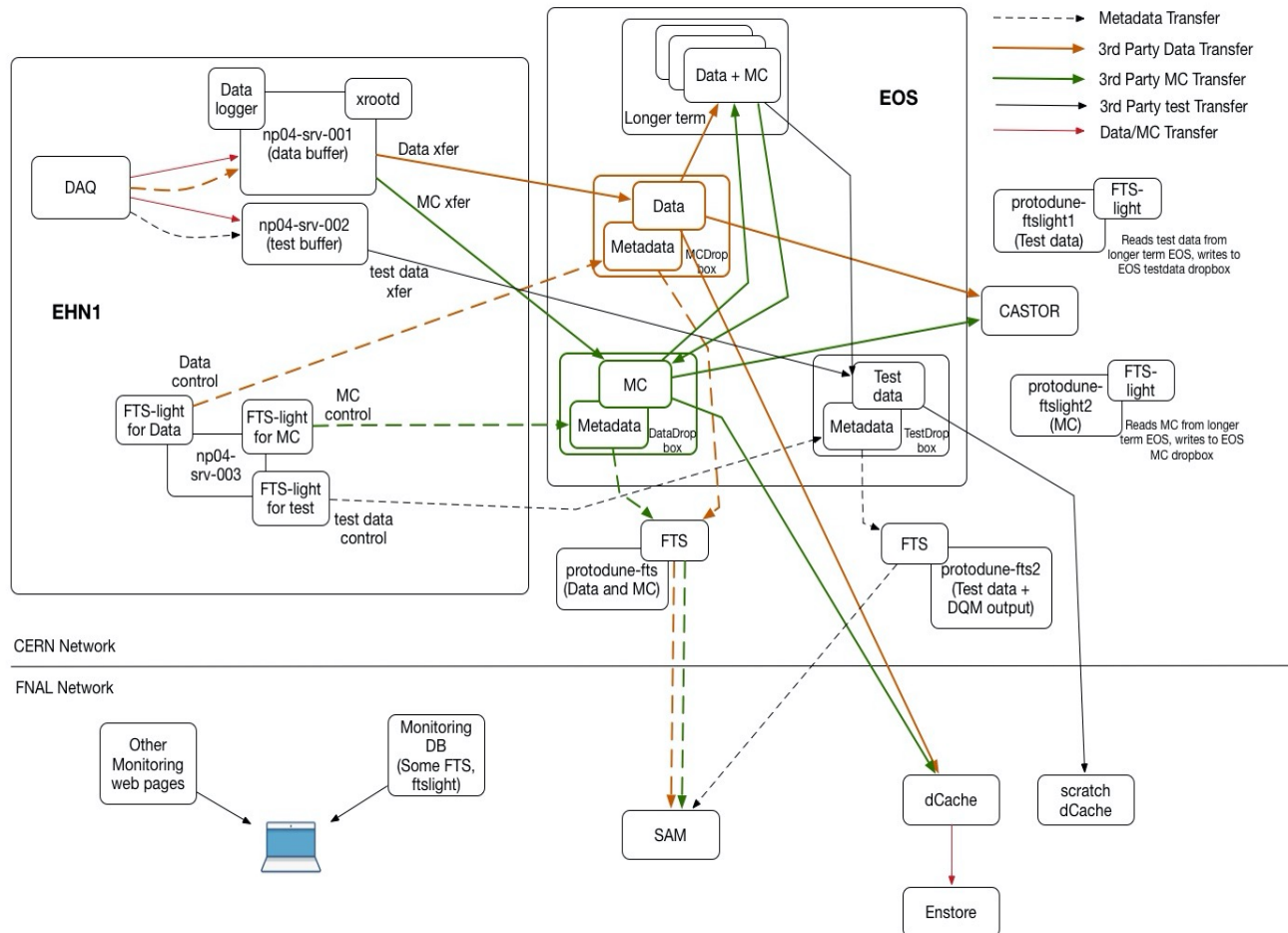
Steven Timm

Dec 14 2021

ProtoDUNE II Big picture

- Both coldboxes taking data right now
- Horizontal Drift (NP04) – APA's in coldbox with cold N2
 - Plus independent electronics test stand
- Vertical Drift (NP02), CRP in coldbox with LAr
 - Top Electronics
 - Bottom Electronics
 - Photon Detectors
- NP02 Cryostat HV testing
 - Legacy phototube readout
 - Arapucas and SiPM
- All ProtoDUNE raw data 2 tape copies, 1 @Cern CTA, 1 @FNAL

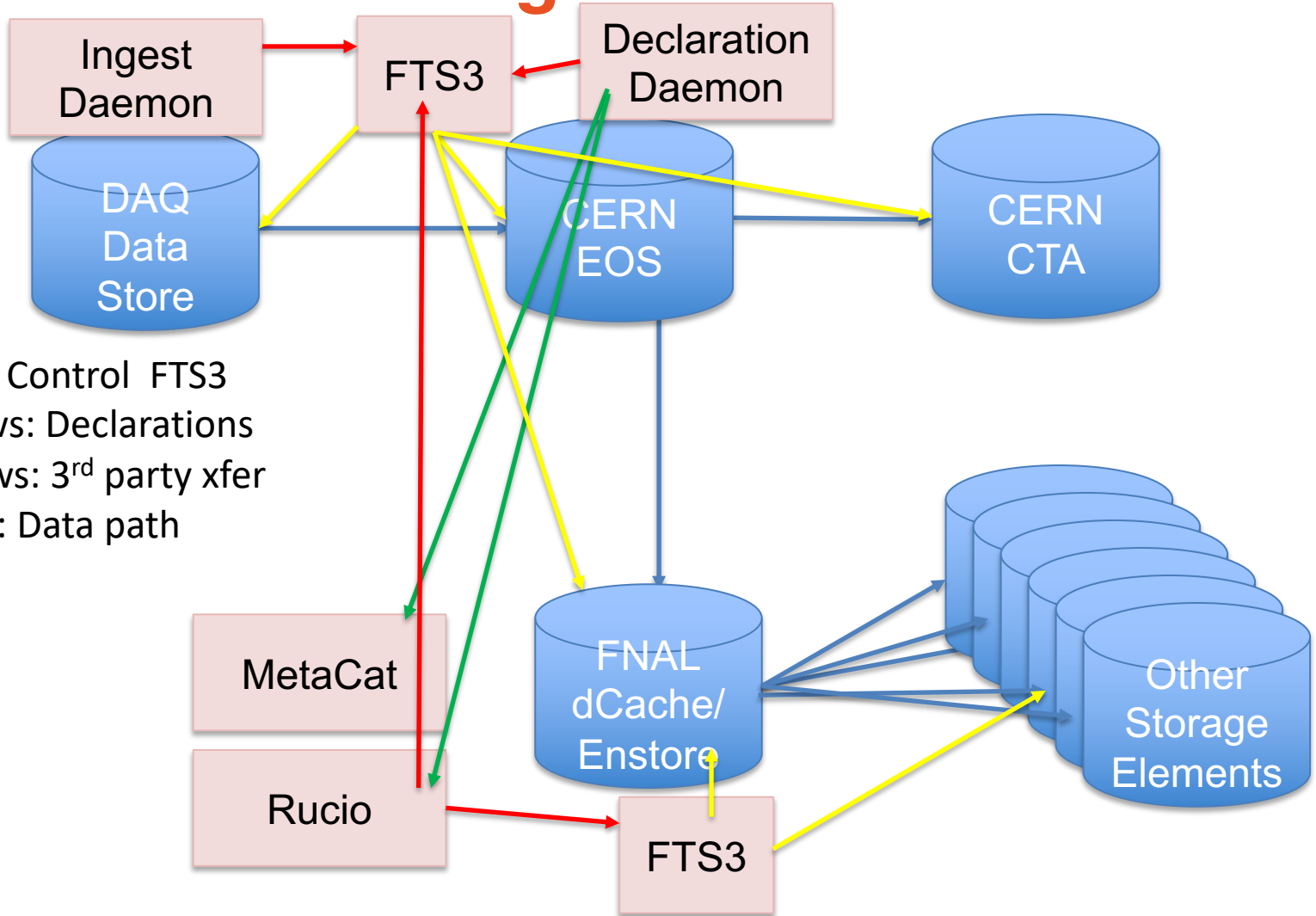
Data Flow Diagram PD-I



Goals by beam time

- Unified file transport for NP02 / NP04 in EHN1->EOSPUBLIC
 - Use FTS3 for both (like NP02 does now)
- Unified file auth for NP02/NP04
- Rucio / FTS3 to do the CERN->FNAL transfer
- Two stages:
 - Ingest daemon EHN1-> EOSPUBLIC (current Fermi-FTS-Light)
 - (Does not require live network connection to FNAL)
 - Declaration daemon EOSPUBLIC -> Metacat and Rucio
 - Rucio handles Transfer via FTS3.
 - Metacat, Rucio, and Data Dispatcher in production

ProtoDUNE-II Configuration



Rucio Ingest Daemon: Requirements

- At Detector Hall in EHN-1
 - Detect New Files on DAQ data store
 - NP02 and NP04 do this differently try to make it the same next time around.
 - Extract metadata, add extra fields if necessary
 - Calculate checksum
 - Monitoring and retry logic
 - Initiate FTS3 3rd party transfer to first SE (EOS Public)
 - Right now we think this is *not* a Rucio upload
 - Right now we think Rucio does *not* manage the DAQ data store.
- Rough replacement for FTS-Light functionality

Rucio Declaration Daemon

- Rough replacement for Fermi FTS
- Runs in computing center (currently in the CERN cloud)
- Declare to Rucio and MetaCat
- Make rules to send to CERN CTA and Fermilab Enstore
- Delete file from initial DAQ data store
- Monitoring and retry logic.
- Replacing functions of current Fermi FTS
 - (Fermi FTS code can already use FTS3 as a transport)
- Almost all of these functions also needed at Fermilab for files that come out of reconstruction/MC.

Rucio Policy: Ingest (1)

- Files can move into Rucio by three ways:
 - Rucio Upload:
 - Copy the file into the storage Element by xrdcp and declare it.
 - File is owned by the user proxy that uploaded it.
 - Declare DID/replica for files already in the storage element
 - This is how we do it now
 - Files end up with the right user ID.
 - Moving from site to site via Rucio requesting the transfer via FTS3
 - All Rucio transfers (even those in user namespaces) use a production proxy and are owned by dunepro
 - File ownership questions get even harder when tokens replace proxies.

Rucio Policy: Ingest (2)

- Rucio has an optional hook to say that every file declared to Rucio MUST also have metadata in an external Metadata Catalog
 - We are going to deploy this hook.
- At Fermilab at least, normal users eventually will be able to write only to Scratch dCache
 - For production, declaration of metadata and of rucio replicas will be done by the new ingest scripts that replace Fermi FTS
 - Anything that gets to tape-backed storage MUST have metadata declared to be there.
 - That metadata must include the expected file retention lifetime.
 - Will need the equivalent of Sam4Users to do this for user files.
 - We do not expect the end user to have to learn and use Rucio commands to do their work.

Milestones Calendar Year 2022

- Rucio database fully loaded
- Metacat/Rucio become master
 - Implies we have all the clients to talk to them
- Data dispatcher runs in place of SAM
- Ingest/Declaration daemons ready
- EHN1 DAQ systems reconfigured to use them
- Data Challenge
- Beam
- Fondue and chocolate.