

# Basic Requirements for the protoDUNE Raw Data Management System and its Possible Configuration

M. Potekhin<sup>a</sup> and B. Viren<sup>a</sup>

<sup>a</sup>*Brookhaven National Laboratory, Upton NY*

August 10, 2016

## **Abstract**

The protoDUNE detectors (dual-phase NA02 and single-phase NA04) require a number of systems in order to marshal raw data from their respective DAQ to prompt processing and mass storage. This document provides a high-level description of essential functionality and configuration of these systems, and initial notes on their design.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	The role of the Raw Data Management System . . . . .	3
1.2	Overview of the Data Flow . . . . .	3
1.3	Hardware Systems . . . . .	4
1.4	Software Systems . . . . .	4
<b>2</b>	<b>Hardware Systems</b>	<b>4</b>
2.1	Buffer Farms . . . . .	4
2.2	EOS and CASTOR . . . . .	4
2.3	FNAL Ingest and Enstore . . . . .	5
<b>3</b>	<b>Software Systems</b>	<b>5</b>
3.1	Transfer of data from DAQ to the Buffer Farm . . . . .	5
3.2	RDMS as a state machine . . . . .	5
3.3	Database . . . . .	6
3.4	Agent Interface . . . . .	6
3.5	Agents . . . . .	6
3.5.1	EOS Transfer Agent . . . . .	6
3.5.2	CASTOR Transfer Agent . . . . .	7
3.5.3	Purge Agent . . . . .	7
3.5.4	FNAL Transfer Agent . . . . .	7
<b>4</b>	<b>Summary and Future</b>	<b>8</b>
<b>5</b>	<b>References</b>	<b>8</b>

# 1 Introduction

## 1.1 The role of the Raw Data Management System

This document sets forth a few requirements for a protoDUNE Raw Data Management System (which will be referred to as **RDMS** for brevity in this text). The system will receive the raw data from two sources: separate Data Acquisition Systems for the NP02 Dual-Phase (DP) and NP04 Single-Phase (SP) DUNE prototype detectors. It will provide support for prompt (“express streams” and other) processing. It will marshal the data to mass storage, first to to the offline disk and then to the archive tape system at CERN. It will also transfer a copy to FNAL and potentially to a few other sites in the US and Europe for production and analysis.

## 1.2 Overview of the Data Flow

Fig.1 shows an overview of the systems that will be required.

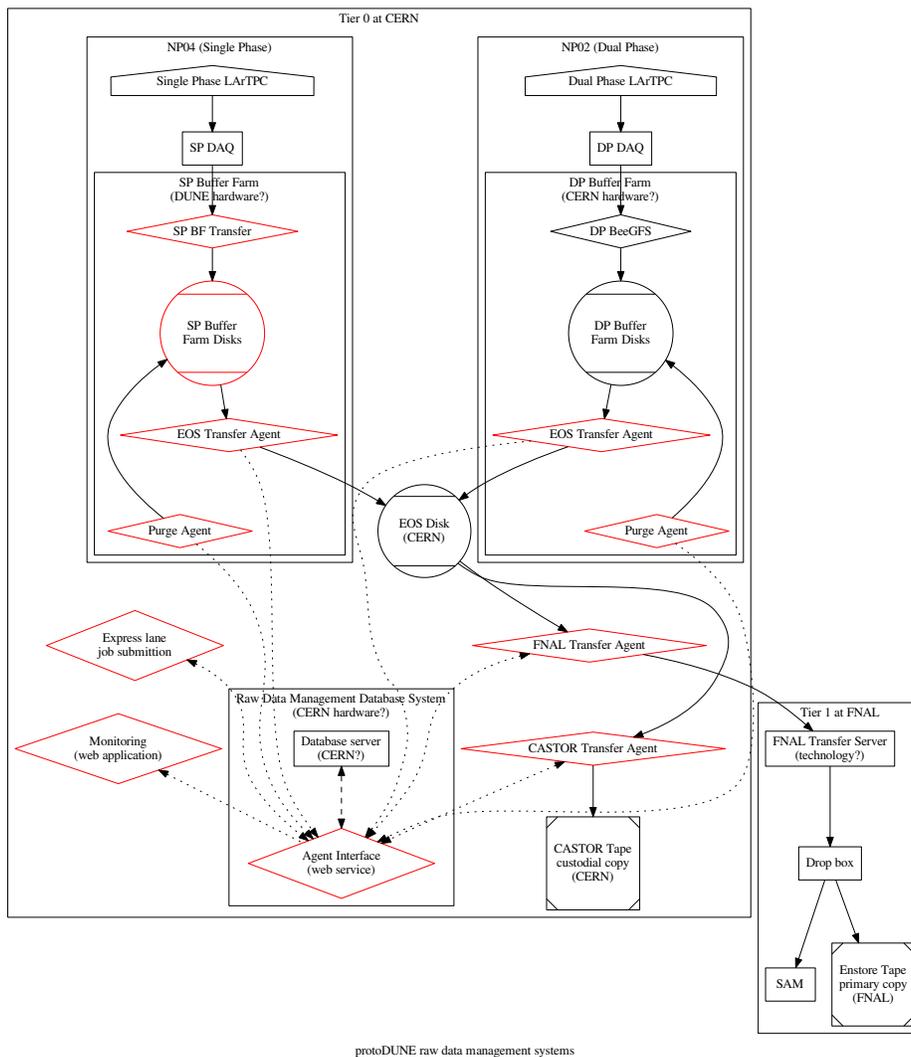


Figure 1: Systems required for protoDUNE raw data management.

In this diagram, red indicates systems where new development is needed. Lined circles indicate disk and lined boxes indicate tape storage hardware systems. Diamonds indicate software systems. Solid arrows indicate flow of detector data, dotted arrows indicate flow of metadata and dashed indicates database queries.

### 1.3 Hardware Systems

**Buffer Farms** each detector is expected to have a nearby and dedicated disk buffer farm

**RDMS Servers** hardware to run the RDMS databases and its Agent Interface.

**EOS & CASTOR** CERN's disk and tape systems

**Enstore** FNAL tape system

### 1.4 Software Systems

**DB and its interfaces** The RDMS contains a central catalog for holding the state of all raw data files, which must be equipped with a proper interface.

**agents** Various programs responsible for performing some action on a set of raw data files, potentially based on the state registered in the DB and recording the outcome to the DB.

Details of these systems are given in the following sections.

## 2 Hardware Systems

### 2.1 Buffer Farms

Initial design for both NP02 and NP04 includes a nearby and dedicated *buffer farm* cluster storage system. They are necessary to meet a number of requirements, such as

- Isolate online systems from offline ones and provide adequate local buffer space to enable data taking even in the event of an outage in the network connection to the CERN central storage or mass storage itself.
- Provide necessary bandwidth for recording data at a very high rate which cannot be achieved by a single disk drive.
- In case of NP02, serve as a large staging area for the processing of the data that is planned to take place locally on site of the experiment. This is not planned for NP04.

The two buffer farms are not expected to be similar or identical in terms of hardware characteristics, configuration or its software because the two DAQs differ.

### 2.2 EOS and CASTOR

The EOS disk and CASTOR tape systems are expected to be provided by CERN. DUNE must estimate the required volume and bandwidth into each of these systems (and out of EOS) for each detector individually and should then negotiate with CERN to understand what is needed for them to provide that. DUNE should formulate an internal policy and mechanisms to assure proper storage management and sharing so that data from both prototypes can be accommodated.

## 2.3 FNAL Ingest and Enstore

The protoDUNE raw data management system will assure the raw data files successfully reach the FNAL transfer server. Another system is expected to then assure the data is properly cataloged and archived to FNAL Enstore tape. It is expected the same systems used for DUNE 35t and other Fermilab-based experiments will be used. Specifically the transfer server will deposit the accepted raw data files into a “drop box”. From there any metadata about the file will be inserted into the SAM databases and the file itself archived to Enstore.

## 3 Software Systems

### 3.1 Transfer of data from DAQ to the Buffer Farm

Raw data must be transferred from each detector DAQ to its respective buffer farm. The general requirements for such transfer are:

- must scale to provide required level of throughput from DAQ to disk.
- integrating into developed DAQ software must not require substantial new effort.
- the DAQ must not block while a transfer is ongoing such that it can not accept new data from readout electronics.

The transfer systems for the two detectors may differ. Current understanding and recommendations are summarized:

**dual-phase** NP02 has determined that XRootD [1] and GPFS are the two candidates for buffer storage that are most likely to satisfy their needs. More R&D is under way in this area.

**single-phase** NP04 will begin to evaluate XRootD [1].

In either case, RDMS will be responsible for transferring the raw data files out of the buffer farm, according to the data flow pattern in Fig.1 or a similar design.

### 3.2 RDMS as a state machine

The protoDUNE RDMS is based on the idea of each file progressing through a *state machine* of well defined states and explicit transitions between those states. State transitions are enacted by a set of software agents. The agent is required to perform an action based on the current state of a file as set in the central RDMS database and record the outcome of the transition in said database.

Most agents will be written to perform a number of state transitions. Some may inject initial state based on external information (eg, a file appearing on a buffer farm). Other agents may merely monitor state changes. Figure 1 describes one possible set of agents. As we learn more the set may grow with new functionality or splitting large agents into smaller ones. These agents are described in section 3.5.

### 3.3 Database

The database is required to:

**performance** accept new records at a rate on order of up to 10Hz and read-only queries an order of magnitude higher.

**schema** full history of inserts (no update nor delete of a previously inserted row), generate a unique identifier for each managed raw data file, relational associations to that ID.

The database is not directly exposed to agents but instead an Agent Interface (RDMS AI) is provided as an HTTP “web service” API, described next.

### 3.4 Agent Interface

The only direct access to the database should be through the Agent Interface. It is required to:

- present all database operations through a HTTP “web service” API.
- expose functionality required by all expected agents as described below.
- support and require authentication/authorization where needed.

All clients of the database shall use this interface, be they actually effecting state changes or be they merely monitoring the state of the database.

### 3.5 Agents

Agents are expected to be command line programs or in some cases web applications. In general, they are required to:

- access and authenticate to the RDMS AI
- respect the allowed state transitions
- report back on the outcome

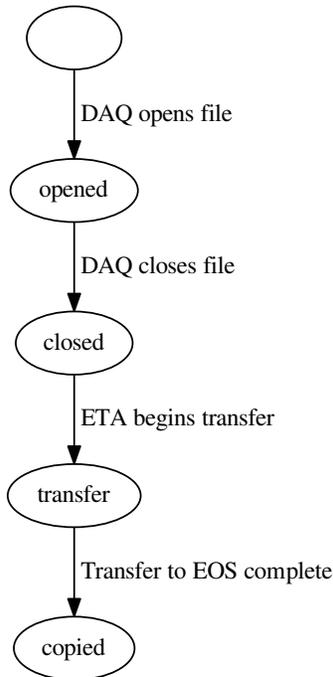
#### 3.5.1 EOS Transfer Agent

The EOS Transfer Agent (ETA) is responsible for transferring files from the buffer farms to EOS and recording the results. Some detail will be given here as a prototype for what must be given for all agents.

Figure 2 gives the state machine implemented by the ETA. This agent progresses sequentially through its possible states. It begins by watching its BF for the appearance of a new file and when one appears it registers the file to the RDMS DB via the AI that the file is in state **opened**. When it determines the DAQ has finished writing the file this is registered as **closed** and a transfer to EOS is begun with the equivalent **transfer** state recorded. When the transfer successfully concludes the **copied** state is entered.

In addition to that behavior the ETA must

- be able to perform read and delete file operations on raw data files from both detectors buffer farm disk storage
- honor detector-specific policy regarding the timing of these file operations.
- able to produce copies of raw files on EOS



Example file state machine for EOS Transfer Agent

Figure 2: EOS Transfer Agent states and transitions.

### 3.5.2 CASTOR Transfer Agent

The CASTOR Transfer Agent (CTA) archives a file from EOS to CASTOR tape. It is required that the CTA:

- only archive a file that has reached the `copied` state
- can read EOS file system and write to CASTOR

### 3.5.3 Purge Agent

The Purge Agent is responsible for deleting files when certain conditions are met, depending on the context of the file to purge. For example, the files residing in the Buffer Farm might only be purged after copied to EOS or they may be delayed until archived to CASTOR. A differently configured Purge Agent may operate on files in EOS and purge based the file reaching some appropriate state in the RDMS DB.

### 3.5.4 FNAL Transfer Agent

The FNAL Transfer Agent (FTA) is responsible for transferring files that have been copied to EOS to Fermilab via a to be determined transfer server. It is required to read EOS disk and connect to this server.

Once the transfer successfully complete, the fate of the raw file is no longer the concern of the protoDUNE raw data management system. Its continued management is taken over by systems at Fermilab.

## 4 Summary and Future

This report gives an outline for a design for a Raw Data Management System for DUNE single-phase and dual-phase prototype detectors. It is based on agents enacting state changes on the files orchestrated by a central database. There are still many unknowns and gaps in the design and it is expected that this document will be updated and new ones created as the design matures.

## 5 References

### References

- [1] XRootD, high performance, scalable fault tolerant access to data repositories. <http://xrootd.org/>.