

1 Mu2e Offline Computing Model

2 **Mu2e-doc-48677**

3 **6/30/2024**

4 Robert Bernstein³, Richard Bonventre¹, David Brown¹, Ray Culbertson³, Stefano Di
5 Falco⁴, Bertrand Echenard², Andrew Edmonds⁵, Ralf Ehrlich⁶, Robert Kutschke³, Sophie
6 Middleton², and Stefano Miscetti⁷

7 ¹Lawrence Berkeley National Laboratory

8 ²California Institute of Technology

9 ³Fermi National Accelerator Laboratory

10 ⁴INFN Pisa

11 ⁵York College of the City University of New York

12 ⁶University of Virginia

13 ⁷Laboratori Nazionali di Frascati dell'INFN

14 **Abstract.** Mu2e is a discovery experiment searching for charged lepton flavor violation, lepton number violation, and other evidence of physics beyond
15 the Standard Model detectable through muon decays. The signature measurement is a search for muon-to-electron conversion, a process that previous experiments have determined occurs fewer than approximately once every 10^{12}
16 stopped muons. In this document, we present the Mu2e offline computing model developed to store, process, simulate, and analyze the data collected by
17 Mu2e. Since the experiment is designed to search for rare events, the focus is on supporting the precision analysis. We describe the overall plan to realize the
18 computing model and to support the commissioning, operation, and analysis of the first data. This document does not describe the online computing model for
19 triggering and recording raw data, except as that affects offline workflows or
20 makes use of offline algorithms.
21
22
23
24
25
26

1 Contents

2	1 Introduction	4
3	1.1 The Mu2e Experiment	4
4	1.2 Beam structure and run types	6
5	1.3 Schedule and Data Taking Plan	7
6	1.3.1 Cosmic ray commissioning run	8
7	1.3.2 Beam commissioning	8
8	1.3.3 Physics data taking	9
9	1.3.4 Magnetic Field Mapping	9
10	1.4 Offline Computing challenges	9
11	2 Mu2e computing organization	11
12	2.1 Internal organization	12
13	3 Offline Computing Model Overview	13
14	4 Trigger and Data Acquisition System	16
15	4.1 DAQ system overview	16
16	4.2 Data streams	18
17	4.3 Trigger algorithms	19
18	5 Data Handling	20
19	5.1 System components	20
20	5.2 Storage	20
21	5.3 Job production manager	21
22	5.4 File family plan	21
23	6 Data Processing	24
24	6.1 Pass 1	24
25	6.2 Calibration	25
26	6.3 Pass 2	25
27	6.4 Reduced data sets	26
28	7 Simulations	27
29	7.1 Simulation workflow	27
30	7.2 Event generators	28
31	7.3 Pileup simulation	28
32	7.4 Geant4 Physics processes	31
33	7.5 Geometry description	31
34	7.6 Magnetic field maps	32
35	7.7 Detector response	32
36	7.8 Simulation campaigns	33
37	7.9 Ensembles	34
38	8 Event Reconstruction	35
39	8.1 Tracker Hit Reconstruction	35
40	8.2 Signal Track Reconstruction	37
41	8.3 Straight Track Reconstruction	39
42	8.4 Calorimeter Reconstruction	39
43	8.5 CRV Reconstruction	39

1	8.6 STM Reconstruction	40
2	8.7 Extinction Monitor Reconstruction	40
3	9 Detector Calibration and Alignment	41
4	9.1 Tracker calibration	41
5	9.2 Calorimeter calibration	42
6	9.3 CRV calibration	43
7	9.4 STM calibration	43
8	10 Physics Analysis	44
9	10.1 Analysis Frameworks and Interfaces	44
10	10.2 Reference Analysis	45
11	11 Visualization and Event Display	47
12	11.1 REveMu2e	47
13	12 Databases	49
14	12.1 Run Conditions database	50
15	12.2 EPICS database	50
16	12.3 Conditions database	51
17	12.4 DQM database	51
18	12.5 Good-run database	52
19	12.6 Luminosity database	52
20	12.7 Data-handling databases	52
21	13 Software Management	53
22	13.1 Offline repositories	53
23	13.2 Code development	53
24	14 Data management and Preservation	55
25	15 Offline Data Quality Monitoring	56
26	16 Communication Tools	57
27	17 Documentation and training	58
28	17.1 Documentation	58
29	17.2 Training	58

1 Introduction

The success of the Mu2e experiment at FNAL critically depends on the ability to store, process, simulate, and analyze data in an efficient and timely way. This document presents the Offline Computing Model (OCM) developed by Mu2e to accomplish these goals. It describes the software, infrastructure, tools, and workflows to perform the various computing tasks throughout the life of the experiment. In the following, Offline covers all computing activities besides the data acquisition system and the trigger software.

The Computing Model strives to use efficiently all available resources while minimizing personnel and hardware needs. As HEP computing evolves rapidly, this document focuses on tools and systems rather than specific algorithms and products. This strategy enables the development of flexible solutions that can be quickly adapted to an ever-changing computing landscape, taking full advantage of new technologies or facilities as they emerge. This document begins by a brief introduction of the Mu2e experiment and the challenges faced by Offline computing, followed by an overview of the Computing Model and a detailed description of each of its components.

1.1 The Mu2e Experiment

The Mu2e experiment [1] aims to search for neutrino-less conversion of a negative muon into an electron in the field of a nucleus, a charged lepton flavor violating (CLFV) process. Unlike flavor violation in the quark and neutral lepton sectors, CLFV reactions are extremely suppressed in the Standard Mode (SM) [2–6], and an observation would be a clear sign of New Physics (see for example Ref. [7]). More broadly, CLFV processes are linked to the questions of flavor, families, neutrino masses, and even the matter excess of the Universe, should it arise from leptogenesis [8, 9].

Muon-to-electron conversion is one of the most sensitive CLFV probes in the muon sector. The best experimental limit on the process rate

$$R_{\mu e} = \frac{\Gamma(\mu^- + N(A, Z) \rightarrow e^- + N(A, Z))}{\Gamma(\mu^- + N(A, Z) \rightarrow \nu_\mu + N(A, Z - 1))} < 7 \times 10^{-13} \text{ (90%CL)}$$

has been set by the SINDRUM-II experiment [10]. Mu2e plans to increase the sensitivity by four orders of magnitude using a concept developed to overcome the limitations faced by the previous generation of experiments [11], as illustrated in Figure 1. A primary 8 GeV proton beam is extracted from the Fermilab Delivery Ring via slow resonant extraction [12] with a pulsed timing structure of 250 ns-wide proton pulses separated by 1695 ns. The beam collides with a production target positioned at the center of the production solenoid (PS), and the pions produced in these interactions are directed towards the transport solenoid (TS) by the graded PS magnetic field. Muons are mostly formed in $\pi \rightarrow \mu\nu$ decays in the PS and TS. A target made of Al annular foils is positioned in the detector solenoid (DS), stopping about 3×10^{10} muons/s. In addition to pions, the primary proton beam also produces a large number of electrons, resulting in a flash of low-momentum particles reaching the detector after 150–200 ns of the primary proton interaction. Muons trapped in the field of the target nucleus quickly cascade down to the muonic 1s bound state before undergoing one of the following processes: neutrinoless conversion $\mu^- + N(A, Z) \rightarrow e^- + N(A, Z)$, a coherent process resulting in the emission of a mono-energetic electron; decay in orbit (DIO) $\mu^- \rightarrow e^- \bar{\nu}_e \nu_\mu$, producing electrons with a momentum spectrum extending up to the conversion energy; or muon capture $\mu^- + N(A, Z) \rightarrow$ all captures, producing a wide range of final states. The conversion energy and the relative capture and decay rates depend on the nucleus. In aluminum, the capture (decay) rate is 61% (39%). The conversion energy is given by $E_{\mu e} = m_\mu - E_{binding} - E_{recoil} =$

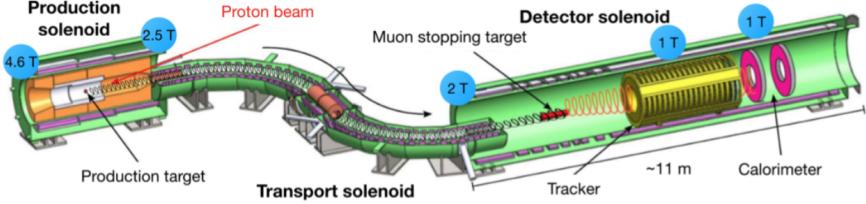


Figure 1. Schematic view of the Mu2e experiment.

1 104.97 MeV, where m_μ is the muon rest mass, $E_{binding}$ denotes the binding energy in the 1s
 2 state, and E_{recoil} is the nuclear recoil energy [13, 14]. The time distribution of $\mu - e$ conversion
 3 is a function of the muonic atom lifetime (τ_N), which is also nucleus-dependent. For instance,
 4 $\tau_{Al} = 864$ ns and $\tau_{Au} = 74$ ns.

5 Background can arise from several processes, scaling with the number of protons on
 6 target or exposure. Decays in orbit (DIO) produce a momentum spectrum extending up to
 7 the conversion energy with a rapid fall near the spectrum endpoint. Radiative pion captures
 8 $\pi^- N \rightarrow \gamma N'$ (RPC), followed by internal or external photon conversion, can also produce
 9 electrons having energies compatible with a conversion signal. This background is rejected
 10 by taking advantage of the pulsed proton beam. Since the pion lifetime (26 ns) is much
 11 shorter than the Al muonic atom lifetime (864 ns), restricting the conversion search to a time
 12 window greater than 700 ns reduces the RPC background to acceptable levels, as schemati-
 13 cally shown in Figure 2. This technique requires a beam extinction between proton pulses at
 14 the level of 10^{-10} to ensure that no significant amount of delayed pions are produced. Cos-
 15 mic rays interacting or decaying in the detector volume are another source of background
 16 electrons, as well as antiprotons annihilating in the stopping target or the TS, beam electrons
 17 with momentum ~ 100 MeV, or decays in flight of negative muons and pions in the DS.
 The detector is optimized to search for the conversion process while maintaining an expected

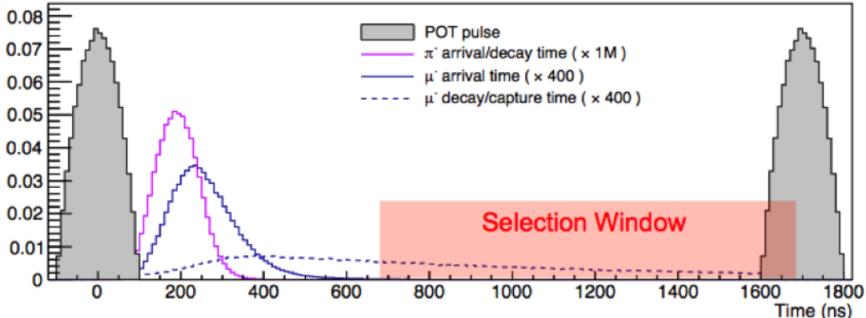


Figure 2. Proton pulses hit the production target 1695 ns apart. A delayed livegate selection window
 (indicated in red) is used to remove prompt backgrounds.

18
 19 background level below a single event. An annular tracker of approximately 3 m long con-

taining 18 tracking stations composed of 5 mm diameter straw tubes (for a total of 20736 straws) located downstream of the stopping target. Each straw is read out from both ends, providing an estimate of the location of each hit along the wire by comparing the signal arrival times. The intrinsic momentum resolution is at the level of $\Delta p < 300$ keV for 100 MeV electrons. An electromagnetic calorimeter comprising two annular disks containing 674 undoped CsI crystals each is positioned right after the tracker. Each disk contains 674 undoped CsI crystals read out by two silicon photo-multipliers. Test of an early prototype indicates an energy resolution $\Delta E/E \sim 16\%$ at 100 MeV and a time resolution of about 100 ps [15]. The inner regions of both detectors are left uninstrumented, blinding the apparatus to nearly all low-momentum muonic atom backgrounds and remnant beam particles. Combined measurements in the tracker and calorimeter provide particle identification sufficient to reject the background from muons misidentified as electrons. Cosmic rays interacting with the detector could produce signal-like electrons in the detector, and a Cosmic Ray Veto (CRV) system surrounding the DS and part of the TS is employed to reject this background with an efficiency of 99.99%. It consists of modules of four layers of extruded plastic scintillation counters outfitted with wavelength-shifting fibers and read out by SiPMs. The proton beam extinction is monitored with a magnetic spectrometer built from Si pixel sensors designed by ATLAS, located downstream of the proton beam. The stopped muon rate is estimated by measuring photons emitted during the capture process with high-purity Ge and LaBr₃ detectors positioned in a shielding house ~ 34 m downstream of the stopping target foils. The data collected by each sub-system are digitized by the front-end electronics and recorded by the data acquisition system. The detector readout starts only after the flux of beam flash particles has subsided (about 500 ns after the primary proton beam hits the production target) to limit the readout rate.

The conversion signal is identified using information from the tracker and the calorimeter. The track reconstruction algorithm is factorized into three main parts. Hits along the tracker wires are first reconstructed from the digitized signals, and a multivariate classifier is used to reject hits produced by low-momentum Compton electrons. Pattern recognition algorithms are then used to select groups of hits compatible with helical trajectories. Two separate methods are used: a calorimeter-seeded and a tracker-seeded algorithm. Finally, a Kalman fit is performed to increase the accuracy of the reconstructed track and further reject spurious candidates. The resulting efficiency for conversion electrons is around 98% for a nominal proton bunch intensity with a background rejection factor greater than 1000. Applications of AI/ML algorithms are investigated to further improve the performance of reconstruction algorithms.

The Mu2e data-taking plan assumes two running periods (run 1 and run 2) separated by a shutdown to upgrade the accelerator complex. The experiment will start taking data using a low-intensity proton beam to facilitate the commissioning phase before switching to a high-intensity mode. A detailed study of the various background sources has concluded that the expected run 1 5σ discovery sensitivity is $R_{\mu e} = 1.2 \times 10^{-15}$, with a total expected background of 0.11 ± 0.03 events [16]. Mu2e can also provide complementary information regarding the nature of neutrinos by searching for $\mu^- + N(A, Z) \rightarrow e^+ + N(A, Z - 2)$, and search a light neutral invisible particle in $\mu^+ \rightarrow e^+ + X$ (e.g. an axion-like particle (ALP) or a familon with lepton-flavor violating couplings).

1.2 Beam structure and run types

The Fermilab accelerator complex will deliver the proton beam to Mu2e. As this system provides beam for other experiments as well, only a fraction of the running time will be allocated to Mu2e. Protons are first accelerated to 8 GeV and compressed into batches in

1 the Booster Synchrotron ring. The booster can deliver proton batches every 67 ms, a time
 2 interval called a "tick". In low-intensity mode (aka 1 booster batch mode), one batch is sent
 3 to the recycler ring and re-bunched into four bunches (or spills) of 10^{12} protons each. The
 4 spills are then transferred to the delivery ring every 112.3 ms. The total time to perform these
 5 operations corresponds to 8 ticks. The next 12 ticks are used for accelerating beam with
 6 the main injector (MI) and delivering protons to other experiments. This 20 ticks cycle, the
 7 main injector cycle, repeats without a break for about a minute, followed by a brief pause to
 8 execute other cycles (e.g. delivering protons to the Fermilab test beam areas). This whole
 9 process is called a supercycle and is repeated uninterrupted during continuous operation.

10 Each spill in the deliver ring is resonantly extracted to the Mu2e experiment. Pulses of
 11 1.6×10^7 protons (on average) are sent to the Mu2e primary target every 1695 ns during 107.3
 12 ms, followed by a 5 ms reset period. This corresponds to 63298 proton pulses per spill, with
 13 4 spills per Main Injector cycle. The proton pulses are about 250 ns wide, and the number
 14 of protons between pulses is required to be less than 10^{-10} [17] to reduce backgrounds from
 15 late interactions. In high-intensity mode (aka 2 booster batches mode), the main injector
 16 cycles are re-arranged into 21 ticks with two booster batches used to create 8 spills for Mu2e.
 17 Each spill is extracted in the delivery ring during 48.1 ms with 3.9×10^7 protons / pulse.
 18 Including reset time, beam is sent to Mu2e for 380 ms every 1.4 seconds. The low- and
 19 high-intensity beam patterns are illustrated in Figure 3. The period during which beam is
 20 delivered to Mu2e is traditionally called "on-spill", by contrast to the "off-spill" time during
 21 which no protons are delivered. The proton bunch intensity (PBI), defined as the number of
 22 protons per pulse, can vary substantially within a spill. Slow-extraction fluctuations impart an
 23 approximately log-normal distribution on the proton pulse intensity, varying on a timescale
 24 of milliseconds. The beam intensity distribution is characterized by the Spill Duty Factor
 25 (SDF), which measures the relative spread of the proton intensity distribution. Maintaining
 26 high DAQ and reconstruction efficiency requires an SDF of at least 60% [17].

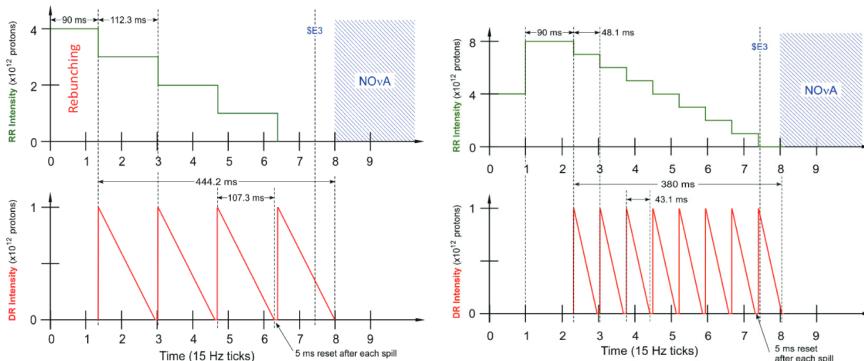


Figure 3. The beam structure as a function of time for the (left) low-intensity and (right) high-intensity modes. The recycler ring and delivery ring beam intensities are shown in green and red, respectively.

27 1.3 Schedule and Data Taking Plan

28 The Mu2e schedule derived from the Project WBS and the Ops WBS (June 2024) is displayed
 29 in Figure 4. The major schedule milestones relevant to Mu2e Offline Computing are discussed

below. A more detailed development and operations plan for offline computing is being developed and maintained separately from this document.

Fiscal year	FY24				FY25				FY26				FY27				FY28				FY29			
	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1
Calendar year	2024																							
Solenoids	1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 4 5 6 7 8 9 10 11 12																							
Inner Detectors	Construction, Installation, check out																							
Shielding	Construction																							
CRV	Installation																							
Beamlines	External																							
Beam Delivery	Insertion																							
Data taking	Hatch																							
	ARR																							
	Rampup to full intensity																							
	First run																							
	Long Shutdown																							

Figure 4. The Mu2e Schedule (June 2024)

2

3 1.3.1 Cosmic ray commissioning run

4 The cosmic ray commissioning period will start in May 2025 and end with the achievement
5 of the detector KPP, as described in Ref [18]. Immediately following the KPP, another cosmic
6 ray run will start and continue until the detector is inserted into the magnet, currently sched-
7 uled for May 2026. During these periods, cosmic ray data will be recorded by the tracker,
8 calorimeter, and a subset of the cosmic ray veto modules. The cosmic ray run will resume
9 after the insertion of the tracker and calorimeter into the magnet, being the first operations
10 in vacuum and in a magnetic field. The cosmic ray veto modules will also be installed at
11 this time, adding information to the data stream. The data rates and associated storage needs
12 should be relatively modest.

13 In addition to supporting detector commissioning activities, offline computing will ex-
14 ercise the full processing chain: transferring data from the DAQ disks, performing the first
15 reconstruction pass to derive calibration constants and producing data quality metrics, repro-
16 cessing data with updated calibration constants, and producing reduced datasets for analysis.
17 To facilitate detector debugging efforts, Offline will follow the beam data processing plan
18 documented in section 6, with a nominal turnaround of 2-3 hours to reconstruct the data,
19 subject to resource availability.

20 1.3.2 Beam commissioning

21 The beam commissioning phase is expected to start in December 2026 with the full detector,
22 including the stopping target and extinction monitors, at low beam intensity. Prior to the start
23 of beam commissioning, the offline calibration and alignment procedures will be exercised
24 using simulated events. Stress tests of the data logging and data processing with synthetic
25 loads up to levels expected during run 1 beam data taking will be conducted, as well as stress
26 tests of condition information delivery. Analysis tools and event displays will be further
27 developed. Routine operation of many elements of offline computing will be demonstrated.

28 During early beam running, the Offline computing priority will be to support detector sub-
29 system teams working debug the detector. In addition, offline will support detector groups
30 to adapt algorithms to observed conditions, in particular calibration and alignment proce-
31 dures, and continue the development of computing infrastructure tools. Fast processing of

1 raw detector data will be critical to provide detailed data quality monitoring and feedback
2 to the detector sub-systems in a timely manner, and offline computing will use this phase
3 to improve workflows and procedures to make sure the experiment is ready for physics data
4 taking.

5 **1.3.3 Physics data taking**

6 Once stable beam conditions have been achieved, the DAQ configuration will be optimized
7 to record potential conversion electrons. Some of the trigger bandwidth will be allocated
8 to dedicated calibration channels, and specific calibration runs will also be performed. Data
9 will be routinely processed by the first pass of offline reconstruction as soon as they have been
10 transferred from the DAQ buffer disk, and data quality metrics will be available within a few
11 hours of data collection. Once calibration has been performed, the data will be reprocessed
12 with the updated calibration constants. Calibration data taken with specific configurations
13 will require separate processing. This document assumes this configuration is used for the
14 large majority of the data recorded and processed in the computing model.

15 **1.3.4 Magnetic Field Mapping**

16 Precise reconstruction of tracks in Mu2e requires a detailed model of the DS magnetic field.
17 A dedicated Mu2e operations team will oversee a detailed map of the Mu2e DS field using a
18 custom device, and the conversion of those measurements into an Offline field model. Studies
19 using a range of simulated coil displacements showed that a field model based on a calculation
20 assuming the nominal DS coil positions meets the requirements of the Mu2e trigger [19].
21 Consequently, if the Mu2e operations schedule requires, the field map may be postponed
22 until after run 1 . Final quality track-based physics results will only be possible after the
23 field model is complete and the data are reprocessed.

24 **1.4 Offline Computing challenges**

25 Mu2e offline computing faces several challenges, some of which are specific to Mu2e while
26 others are shared by the HEP community.

27 **Simulation** - Mu2e is a precision experiment aiming to maintain a background level for
28 the conversion search well below a single event. Given the expectation of $> 10^{18}$ muons
29 delivered, large samples of simulated events are needed to understand such low background
30 levels. Directly simulating beam protons through the full detector response is prohibitively
31 resource intensive in such quantities, so the simulation process is split into discrete stages,
32 and resampling is used to reduce the computational cost. Managing simulation produc-
33 tion and keeping track of information between these stages present non-trivial challenges.
34 Integration of high-performance computing to speed up simulation campaigns introduces
35 additional complications as well. These aspects are discussed in sections 5, 6, and 7.

36 **Reconstruction and calibration** - To efficiently reject the DIO background, tracks must
37 be reconstructed with a high degree of precision, and a high suppression of non-Gaussian
38 tails. To support track reconstruction and to suppress cosmic ray particle induced physics
39 signal backgrounds, the calorimeter energy scale must be known to 1%, and the timing
40 relative to the tracker to less than 1 ns. To veto the majority of the cosmic rays that can
41 fake a conversion signal, the CRV relative timing must be known to < 1 ns, and the bar effi-
42 ciencies to much better than 1%. Achieving these requirements requires precise calibration
43 procedures that can be executed on data recorded in-situ by the detector, using particles

1 both from the beam and from cosmic rays. To accurately reconstruction track momenta, a
2 precise map of the detector solenoid magnetic field is required.

3 The unusual geometry of the Mu2e detector (un-instrumented inner region) presents unique
4 challenges, particularly in track finding. The signal particles in Mu2e are low energy com-
5 pared to most HEP experiments, and are unusual in that they loop inside the detector.
6 These features require dedicated reconstruction algorithms to identify and reconstruct sig-
7 nal particles, and particles useful for in-situ calibration. Reconstruction and calibration are
8 reviewed in sections 8 and 9.

9 **Analysis** - Tools and reduced data formats need to be developed to enable efficient analysis
10 of the data. Extraction of parameters of interest in the presence of many nuisance pa-
11 rameters remains a computationally intensive task and high-performance computing might
12 prove beneficial to produce final results on time. Integration of machine learning techniques
13 into a complex software ecosystem also requires substantial efforts to train models and sup-
14 port special data formats. Event displays also provide critical input to design analysis and
15 extract physics results, and more generally operate the detector and develop algorithms.
16 Sections 10 and 11 focus on these topics.

17 **System infrastructure** - A large suite of activities, common to many experiments, must be
18 performed to ensure smooth and efficient computing operation. These include database de-
19 sign and operation, code development and management, documentation, and user support.
20 In particular, the continuous technological progress and evolution of security requirements
21 need frequent modifications to the overall computing system and associated workflows.
22 Similarly, proper code management and documentation are essential to ensure the long-
23 term success of the experiment. To reduce costs and the global environmental footprint,
24 the computing architecture must be designed to efficiently use storage and CPU resources.
25 These issues are examined in sections 12, 13 and 14.

26 **Documentation and training** - Besides optimizing workflows and algorithms, documen-
27 tation and user training are essential components to improve coding practices and reduce
28 nonessential data (re-)processing. Data quality monitoring also plays an important role in
29 identifying and correcting issues as early as possible. Efficient on-boarding of new per-
30 sonnel also requires good training materials. Efforts in these domains are discussed in
31 sections 15, 17, and 16.

2 Mu2e computing organization

Mu2e is organized into two major divisions under the direct supervision of the experiment leadership: operations and analysis. Computing is distributed across these two groups. Offline computing is the responsibility of Analysis, while Online computing is part of Operations. The trigger software overlaps both sides; real-time issues are managed by the Operations group, but the algorithmic content and physics selections made by the trigger are managed by the Analysis group. The event processing framework (art), conditions data infrastructure, event display, and some data monitoring tools are also shared between Online and Offline. The formal transition between Online and Offline data processing occurs once raw data art files have been written to the data logging disk buffers by the DAQ system.

Mu2e Collaboration Organization

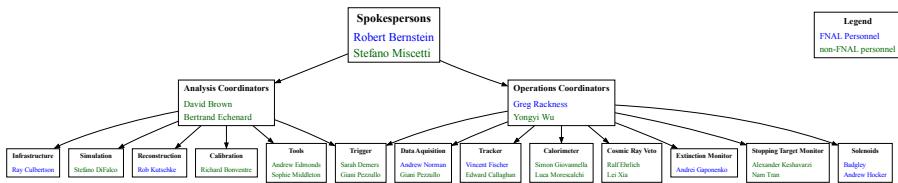


Figure 5. Mu2e experiment collaboration chart, June 2024.

Mu2e Collaboration Organization

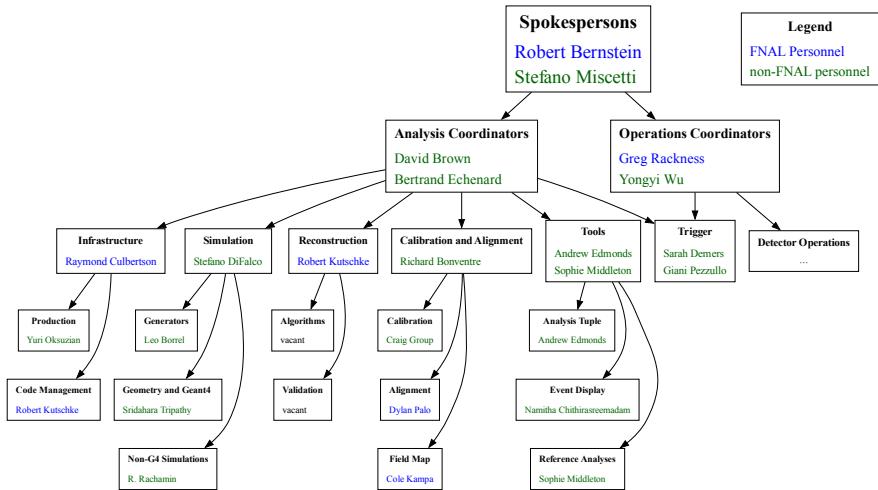


Figure 6. Mu2e Analysis Organization as of June 2024.

1 **2.1 Internal organization**

2 The Mu2e organization chart is shown in Figure 6. Under the Analysis branch, Offline com-
3 puting is divided among several Core Groups responsible for providing tools and expertise
4 for the collaboration, ensuring experiment-wide coherence, and minimizing duplication of
5 effort. The five core groups, and a summary of their charges from [20], are given below:

6 **calibration & alignment** - supervises the development of Offline tools and procedures to
7 align and calibrate the various Mu2e detector systems and coordinate the data processing
8 required to derive calibration and alignment values. This group is also responsible for
9 estimating the statistical and systematic uncertainties on the alignment and calibration pa-
10 rameters, integrating constraints on the overall momentum scale coming from dedicated
11 measurements, determining the calibration and alignment parameters used in simulation,
12 and defining the quantities required to monitor the detector response and validate the agree-
13 ment of detector simulations with the data.

14 **simulation** - develops and validates the Monte Carlo generators used to model signal and
15 background components relevant to Mu2e analyses, based on the best available theoretical
16 models and experimental data. This group is also responsible for maintaining the descrip-
17 tion of the detector geometry and materials in the simulation, and developing Mu2e detec-
18 tor simulations in different Monte Carlo frameworks (e.g. FLUKA) to establish systematic
19 uncertainties inherent to the Geant4 simulation.

20 **reconstruction** - develops and characterizes the high-level algorithms used to create
21 physics objects (e.g. background hit removal, track reconstruction, or calorimeter clus-
22 tering) used in calibration, analysis, and trigger selections. This group also ensures that the
23 reconstruction algorithms used in the trigger meet DAQ requirements, and it defines the
24 reconstruction outputs to be monitored in the validation and Continuous Integration (CI)
25 systems.

26 **infrastructure and production** - manages the offline computing infrastructure and hard-
27 ware resources to ensure event data, simulated event data, and conditions data are deliv-
28 ered for analysis in a timely and accessible way. This group maintains tools to perform
29 data transfer between the online DAQ storage and offline storage, build and distribute the
30 Mu2e software, access data sets catalogs, access and export Mu2e data offsite, maintain
31 offline conditions databases, and perform offline data quality monitoring and continuous
32 integration testing of the Mu2e Offline code base. It is also responsible for collaborating
33 with CSAID and other resource providers to ensure access to the computational resources
34 and technical expertise required by Mu2e.

35 **analysis tools** - develops the tools required to perform the various physics analyses within
36 the collaboration, including the definition of an analysis framework for the experiment with
37 reduced data formats (aka ntuples), event display(s), and reference analyses to evaluate the
38 impact of reconstruction code developments on the experimental sensitivity.

39 Additional responsibilities are shared by all groups, such as documenting tools and proce-
40 dures, or ensuring that the codes meet Mu2e coding standards and performance requirements.
41 A complete description of the core groups is available in Ref. [21].

42 Interfaces between Offline computing and the Mu2e experiment are managed at several
43 levels. Offline computing is represented by the Analysis coordinators during weekly meet-
44 ings with the experiment leadership and Operation coordinators. The analysis coordinators
45 (or their designees) also attend relevant meetings, including DAQ, online computing, and
46 operation coordination. In addition, core group conveners are meeting regularly to discuss
47 issues pertaining to Offline computing. Finally, working groups within each core group are in
48 contact with detector and DAQ experts to coordinate computing and software development.

1 3 Offline Computing Model Overview

2 The Mu2e Offline Computing Model (OCM) describes the software, infrastructure, tools, and
3 workflows to store, process, simulate, and analyze data. This definition covers all computing
4 activities downstream of the data acquisition process. The DAQ system and Online software
5 are not included in this scope, but a summary is provided in section 4 to introduce the concepts
6 and terminology used in the rest of this document. The formal interface between Online and
7 Offline lies at the DAQ buffer disks in the Mu2e building. Online is responsible for writing
8 data on these disks, and Offline is in charge of transferring the files for further processing and
9 safely deleting them on the buffer disk.

10 The OCM is devised to support Mu2e physics goals by designing systems minimizing op-
11 erations effort, addressing anticipated obstacles, and aligning with the host laboratory tools,
12 support, and strategy. This system is based on the framework and tools provided by the FNAL
13 Computing Science and Artificial Intelligence Division (CSAID) and the Fabric for Intensity
14 Frontier Experiments (FIFE) group for small- and mid-scale intensity frontier experiments.
15 The computing resources are mostly provided by the host laboratory in the form of conven-
16 tional UNIX batch systems accessible via the Open Science Grid (OSG), supplemented by
17 external resources whenever available (e.g. high-performance computing centers). However,
18 several hurdles limit the use of external resources. They are usually not controlled by Mu2e,
19 or even HEP, and might be subject to annual applications and restrictions. Some centers may
20 also have limited network connectivity, making it difficult to move large amounts of data in
21 a reliable manner. In these instances, performing computing tasks with large CPU/IO ratios
22 (e.g. simulation) may be more advantageous. Finally, High Performance Computing Cen-
23 ters (HPCs) are very diverse in the hardware offered and might require significant software
24 development effort to use specific resources efficiently. The overall Mu2e computing strat-
25 egy is, therefore, to store all raw, reconstructed, and simulated data at the host laboratory
26 to perform most of the simulation, reconstruction, and calibration tasks. External resources
27 are exploited whenever available to perform non-critical tasks, such as simulation or data re-
28 processing, privileging CPU-intensive operations. Reduced data sets of modest size are also
29 produced to enable data analysis at local institutions.

30 One of the main components of the OCM, the event processing framework, is built on the
31 *art* framework developed by SCD, supplemented by Mu2e-specific algorithms, a condition
32 system, a geometry description, and the definitions of data products. This code base is usually
33 referred to as Mu2e Offline (or just Offline). This model has been in development for more
34 than a decade, from characterizing the detector performance in the TDR and CD-3 phases
35 to the most recent estimate of the experiment sensitivity [16]. At each stage, the model has
36 evolved to manage the growing processing requirements and the increasing complexity of the
37 simulation and reconstruction tasks. This model will continue to mature to take advantage
38 of future computing technologies and infrastructures as well as advances in computational
39 techniques over the coming decade.

40 The workflow designed for the OCM starts with the data written by the DAQ system on
41 the buffer disk. At the beginning of each run, the DAQ system copies all information used to
42 configure the detector and DAQ into the online run conditions database. The online content
43 is periodically streamed to an offline database instance. As new runs appear in the online
44 database, the raw data are transferred from the DAQ storage disk to a disk visible to Offline
45 worker nodes. There are about 15 independent data streams but these may be packaged into
46 fewer files due to constraints from the TDAQ system and data movement capabilities. A first
47 pass (Pass 1) of Offline reconstruction is triggered with minimal latency. This Pass 1 uses
48 the reconstruction results to produce updated calibration constants and offline data quality
49 metrics, persisted in the corresponding databases. The data are then reprocessed (Pass 2) with

1 the updated calibration conditions. Reconstructed data objects from both passes are stored to
2 tape and registered in a file catalog. A blinding scheme will be developed and applied to data
3 released to collaborators to avoid any implicit experimental bias. In specific cases, e.g. if
4 substantial improvements to either the reconstruction codes or calibration values, a complete
5 data reprocessing of all data will be done. Finally, reduced data sets (aka skims and ntuples)
6 are produced for further analysis. This workflow assumes 8/5 support for offline databases,
7 with best effort outside standard hours.

8 Mu2e events are selected by trigger algorithms based mostly on information from the
9 tracker and calorimeter. The low-level digital data coming from each subsystem are first con-
10 verted into physical objects using calibration data (e.g. objects providing a physical time,
11 position, energy, etc.). A set of algorithms then filters and aggregates these hits into increas-
12 ingly complete physical objects. A two-step clustering algorithm is used to form calorimeter
13 clusters from crystal hits, including a procedure to recover split-off clusters. The track-finding
14 algorithm starts by filtering hits produced by low-energy electrons with dedicated algorithms,
15 then identifies clusters of hits with a short time window (typically \sim 50 ns). These hits are
16 passed to helix finding algorithms to extract an approximate helix parameterization. One al-
17 gorithm uses the position of high-energy calorimeter clusters and the stopping target to seed
18 the helix search, while the second is purely based on tracker information. The calorimeter-
19 seeded algorithm tends to be more robust against higher levels of background but has inher-
20 ently lower efficiency. Reconstructed helices are finally passed to a Kalman filter track fitter.
21 Two configurations have been developed for the Mu2e KinKal fit, one optimized for use in
22 the online trigger, and a second for offline analysis. Physics and calibration data are selected
23 by a set of trigger filters based on the reconstructed tracks and calorimeter clusters, with ad-
24 justable prescale factors to tune the total trigger rate. Data from the extinction monitor and
25 stopping target monitor are reconstructed in dedicated applications. Their final outputs are
26 associated with the corresponding event data by their DAQ timestamp.

27 Complete end-to-end simulations of Mu2e datasets are prohibitively expensive to pro-
28 duce directly given the huge number of protons on target. Current samples were produced
29 by splitting the processing into pileup, signals, and physics backgrounds. Geant4 is used
30 to model the Mu2e experiment and particle interactions. Pileup is simulated starting from
31 protons hitting the production target, recording the proton daughter particles as they exit
32 the Transport Solenoid. Daughter particles are re-simulated (resampled) many times to in-
33 crease the effective statistics, limited by the eventual repetition of the same daughter particles
34 (oversampling). Current pileup samples were produced with a \sim 10M core hour campaign,
35 producing roughly 1 second of pileup at the expected nominal Mu2e intensity. Muon-based
36 signals and physics backgrounds are simulated starting from stopped muons recorded during
37 the pileup simulation, which are resampled many times, forcing them to decay via a desired
38 physics process. Stopped muons are plentiful in the pileup sample, and existing muon-based
39 signal and physics background samples are many times larger than what Mu2e might observe
40 in the region of interest. Cosmic ray backgrounds are simulated starting with standard gen-
41 erators (CRY, CORSIKA), stopping and resampling the particles entering the CRV. Current
42 samples correspond to roughly 3 times the expected run 1 sky live-time. The net output of
43 the pileup, signal, and physics background simulations are energy deposits in the Mu2e de-
44 tector. Simulated samples of raw on-spill data are created by mixing samples of pile-up and
45 signal/physics background scaled to the expected beam intensity average and fluctuations.
46 The different sources are finally mixed to produce a realistic sample, and processed with the
47 same reconstruction code used to produce the data.

48 Tools and policies are in place for code management, code review, building code, code
49 validation, release management, release distribution, setup of the development environment,
50 and setup of the runtime environment. For example, the code base is maintained in Git

1 repositories, and the GitHub pull request system is used to control and review contributions
2 to the code base. The SCD-supplied CMSBOT software is also used for the launch tests that
3 are executed on the SCD Jenkins system. These tools will continue to evolve to integrate
4 changes in the supported product stack by SCD and FIFE. Calibration data are stored in
5 condition databases, indexed by (fraction of) runs, and grouped into intervals of validity.
6 The metadata needed to access and process data (DQM, MetaCat, Rucio, luminosity,...) are
7 stored in several dedicated databases. All databases are supported by the database group of
8 the FNAL IT division.

9 The analysis model is designed to be lightweight and flexible. Information about Mu2e
10 events will be provided to users in the form of ROOT ntuples containing a simplified
11 output of the full reconstruction algorithms (while we expect the vast majority of analysts will
12 use reduced data sets, analyses could still be conducted within the full framework). These
13 reduced datasets require much less storage, facilitating data analysis on local resources and
14 reducing the development time of analysis pipelines. A Python interface and a common
15 analysis environment will also be provided to facilitate the inclusion of external tools (e.g.
16 machine learning or statistical tools). In addition, several event displays are available to
17 visually inspect individual events and help design analysis codes.

18 To this day, generators for all of the relevant signal and background processes exist and
19 have been exercised in Mock Data campaigns. Reconstruction algorithms exist for all of the
20 Mu2e detectors, with the majority of detector response functions tuned to measured data.
21 Most of these algorithms are highly advanced and in nearly final form, where further optimiza-
22 tion requires actual data. Many detector calibration algorithms have been demonstrated
23 using bench test data and simulations. In many cases, these are fully developed, while in
24 others they constitute only proof of principle. Some algorithms still require further develop-
25 ment to meet the performance requirements. However, none of the missing calibrations are
26 critical to triggering, recording, or evaluating the quality of Mu2e data, but will be required
27 for precision physics analyses.

28 Procedures to automatically process raw data as they appear from the online system exist
29 and have been demonstrated to meet requirements, using data from bench tests and sim-
30 ulations. A conditions database and file catalog system based on the central tools currently
31 supported by CSAID are in active, daily use, and have been exercised at scale in simulation
32 campaigns. Operation of Geant4 in multithreaded mode on HPC resources has been demon-
33 strated. The performance of the Offline processing has been benchmarked using preliminary
34 estimates of the data volumes the Online system will produce, and shown to fit within the
35 processing and storage envelope agreed with FNAL central computing. Configurations of
36 the Offline algorithms run as part of the Online software-based event selection process (trig-
37 ger) are in an advanced development stage and have been shown to meet requirements for
38 run 1 operations.

39 Code development adheres to industry standards and best practices, following a continu-
40 ous integration (CI) model in which changes are frequently integrated into the source code.
41 All changes are done via pull requests reviewed by experts and validated with a series of
42 quality checks before integration. Database development is fully integrated with this work-
43 flow and synchronized with software releases. Prototype analysis interfaces to the Mu2e
44 data have been developed and are in active daily use for developing analyses and calibration
45 algorithms.

4 Trigger and Data Acquisition System

This section provides a brief summary of the Mu2e Trigger and Data Acquisition (DAQ) subsystem to guide the reader through the rest of this document; more details can be found in Refs. [22, 23].

4.1 DAQ system overview

The DAQ system collects, filters, and monitors the digitized data from the different sub-detectors; delivers the data to online and offline processing for analysis; and synchronizes and controls the detector readout. The DAQ system is based on a “streaming” readout in which all detector data are digitized and zero-suppressed in detector front-end electronics before being transmitted to the DAQ system. While this approach results in a larger data rate, it provides a simpler architecture with greater flexibility in data filtering and monitoring. The DAQ system uses otsdaq [24] as a solution, together with the artdaq [22] and art [25] framework to process and filter events.

An overview of the full system is shown in Figure 7. The major components perform the following functions:

- The Readout Controllers (ROC) are reading out, digitizing, and streaming the data from the readout electronics to the DTC. There are 216 ROCs for the tracker, 140 ROCs for the calorimeter, 16 ROCS for the CRV, 1 ROC for the ExtMon, and 1 ROC for the STM. The data, timing signal, and detector control signals are transmitted over a single link from the DTC to tracker/calorimeter ROCs. The timing signal is transported in a separate link for the CRV, while the data are transmitted via ethernet for the monitors.
- The Data Transfer Controller (DTC) receives the system clock and event readout request from the CFO and timing fanout module, and reads out data from the ROC. They are installed in the DAQ servers (up to 2 DTCs per server) and are connected to a maximum of 6 ROCs. Each tracker/calorimeter DTC receives the data of a fraction of the detector, and the data are shuffled among the DTCs via Event Building Switches to form a complete event. The data from the CRV are only pulled for requested (triggered) events.
- The Command fanout module (CFO) is responsible for generating and synchronizing readout requests. It receives the accelerator zero-crossing marker, supplies a continuous clock to the DTCs, and sends readout request control packets for each system clock. A timing fan-out module is used to broadcast the system clock to drive the transmit link in the chain of DTCs to prevent jitter accumulation
- The DAQ servers contain the DTCs and run the online trigger software. The DTCs in each server are connected to two different event-building networks. Triggered events are sent to the data logger node, and a dispatcher forwards in a non-blocking way a subset of events to the data quality monitoring nodes. The data from the monitors (STM and ExtMon) are acquired and persisted by separate processes. Slow control monitoring (EPICS) runs on the EPICS host. Communication is performed over two networks: the DCS & management network and the control & data storage network. Communication to the lab network is managed by gateway nodes.

The data flow starts with the tracker and calorimeter DTCs forwarding readout requests from the CFO to their ROCs. The ROCs read out the detector data in response and the DTC collects ROC data for a given Event Window Tag (EWT), a unique tag sent by the CFO to define an event/. Each DTC stores a fraction of the detector data in its memory. The DTCs are grouped into two sets, connected among themselves via an event-building network (in a given

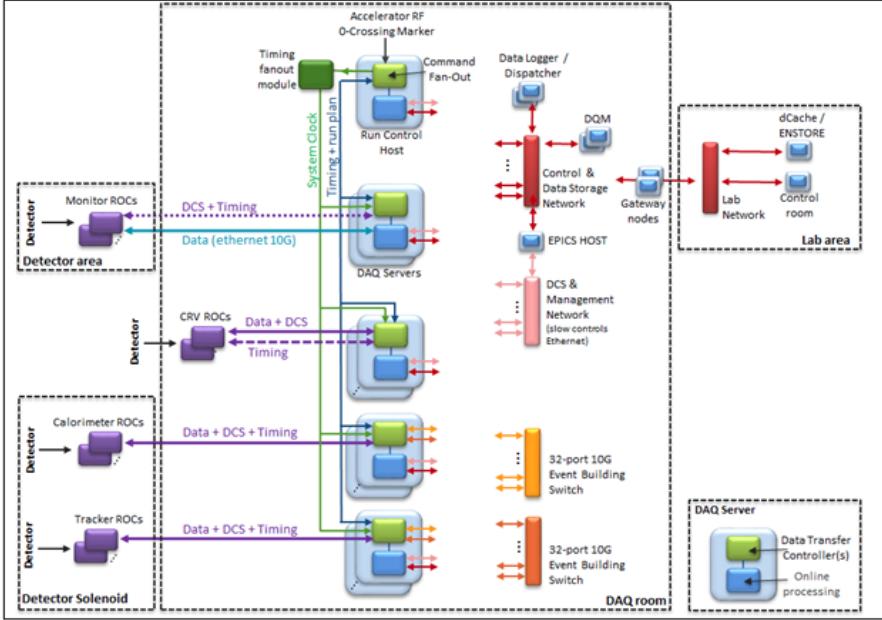


Figure 7. Overview of the Mu2e DAQ system

server, each DTC is connected to a different network). In each group of DTCs, one DTC acts as an Event Builder for a given EWT, and the remaining DTCs send their data to that DTC via the event-building network. The system is set such that the two halves of an event are built by two DTCs in the same server. The two halves of an event are transferred from the DTCs to the main memory by an ArtdaqBoardReader process, which creates fragments and sends them to the ArtdaqEventBuilder process. The ArtdaqEventBuilder runs the online trigger software on the tracker and calorimeter data. In addition, information from the intensity streams is sent to a data logger. Triggered events are forwarded to a secondary ArtdaqEventBuilder. Information from the CRV is requested via an ArtdaqBoardReader process. The BoardReader instructs the DTC to fetch the corresponding EWT data from the CRV ROCs. A fully assembled event is finally forwarded to a data logger node to be persisted to disk. A dispatcher sends a subset of events in a non-blocking way to the data quality monitoring (DQM) nodes. The dispatcher cannot interrupt data logging; events are simply not forwarded if the DQM nodes are all busy. The DQM metrics are periodically forwarded to the online monitoring nodes. Data acquisition from the monitors follows a similar logic and proceeds independently of the tracker, calorimeter, and CRV. The data are read with their ROCs and forwarded to the corresponding DTCs. Dedicated algorithms are processing and persisting the data to separate files.

The data will be labeled by a three-part identifier: run, subrun, and event. The DAQ system has been designed to provide deadtimeless transitions between subruns, while transitions between runs incur a deadtime of a few minutes to allow restarting processes and reloading the firmware. The subrun duration is chosen so that conditions are constant throughout the subrun, it spans an integer of MI cycles, and it starts near the end of the $\sim 1\text{s}$ off-spill period. Subruns will therefore contain both on-spill and off-spill data. The subrun and run duration will be chosen based on operational experience, balancing the size to keep a total number

1 of files manageable while ensuring fast file transfer. At present, the subrun (run) duration is
2 likely to be at the level of a few minutes (6 to 12 hours). For some types of interruptions, a
3 new run might be started when data-taking resumes.

4 Information about the DAQ/trigger configuration and summary information (e.g. duration
5 of subruns, total number of events processed,...) for each run will be stored in an online
6 run conditions database. Slow-control and other monitoring data are persisted in the EPICS
7 database, and run quality information is similarly recorded. Curated information from online
8 content will be periodically streamed to offline database instances to configure data processing
9 tasks and provide calibration constant required by reconstruction algorithms. This information
10 needs to be available promptly in the offline environment to minimize downstream
11 processing delays.

12 **4.2 Data streams**

13 The data collected by the DAQ will be written in about 15 different file streams (or streams)
14 with information from different sub-systems. These are summarized in Table 1 and discussed
15 below. A guiding principle is that offline reconstruction avoids reading events from two
16 different streams at the same time. All events from one subrun will be contained within a
17 single physical file to avoid memory bloat when reading sparse skims. However, a single file
18 may contain events from multiple subruns, and two file streams may have different groupings
19 of subruns into files.

Data Type	Beam mode
Trk+Cal+CRV triggered events	On-Spill
Intensity Stream	On-Spill
ExtMon data	On-Spill
Trk+Cal+CRV triggered events	Off-Spill
CRV zero bias triggers	Off-Spill
Calorimeter Pulser Events	Off-Spill
STM data	On-Spill + Off-Spill
Error/Debug Stream	On-Spill + Off-Spill
Online DQM output	On-Spill + Off-Spill
Log files	On-Spill + Off-Spill

Table 1. Files streams produced by the Mu2e DAQ.

20 The largest stream, triggered on-spill events, contains events selected by the main physics
21 and calibration triggers. The same trigger lines will be run during off-spill periods to record
22 cosmic ray daughters which induce signal-like particles. Additional trigger lines sensitive
23 to through-going cosmic rays passing through the tracker, the calorimeter, or both, will be
24 enabled off-spill. The Intensity stream includes data from proxies to follow fluctuations in
25 the Proton Pulse Intensity for every on-spill event. The Extinction Monitor views a small
26 fraction of the phase space of the scattered proton beam with a pixel telescope and a thick
27 scintillator read out by a PMT (called the Accelerator Fast Feedback). The associated stream
28 produces summary information from these two systems, as well as raw and intermediate data
29 for the pixel tracker. The packaging of summary information remains to be defined but a
30 leading candidate is to collect information for each spill and add a data product summarizing
31 the data in the *art* SubRun object. During off-spill data taking, TDAQ plans to record samples
32 of non-zero-suppressed data from a few sectors of the CRV to study pedestals and events in

1 which the calibration signals are injected into the calorimeter and measured. These events
2 will be identified by a bit in the event heartbeat packet and will be written to their own files.
3 The STM will record pulse summary data for every measured pulse and the full waveform for
4 pulses with an energy close to that of the X-ray and gamma-ray lines of interest. Prescaled
5 raw data and pre-processed data will also be written. The STM operates throughout all of
6 the on-spill and off-spill periods in the supercycle. Moreover, the trigger processes will flag
7 events that exhibit unusual behavior and write them into a dedicated Error/Debug stream. For
8 example, selected algorithms might run in a separate thread with a timeout enabled. If the
9 timeout triggers before the algorithm completes, the event will be written to the Debug/Error
10 stream. Finally, the DAQ system will produce online Data Quality Monitoring data and log
11 files, both of which will be saved in their respective streams.

12 **4.3 Trigger algorithms**

13 Physics and calibration data are selected by a set of filtering algorithms implemented as mul-
14 tiple independent reconstruction paths, each path running one or several reconstruction al-
15 gorithms. The main physics trigger uses the offline reconstruction algorithms with settings
16 optimizing the timing performance (see section 8). The artdaq architecture can reuse the
17 output of a given algorithm if it appears in multiple paths, thus minimizing the running time
18 of the online processing. Prescaled factors can be adjusted for each path independently to
19 tune the total trigger rate. Similarly, paths can be enabled or disabled depending on the beam
20 conditions (on-spill or off-spill).

21 The main physics trigger is based on information from reconstructed tracks and, to some
22 extent, reconstructed calorimeter clusters. The track reconstruction algorithm is factorized
23 into three main parts:

- 24 • Hits reconstruction from the digitized signals recorded by the tracker. Each signal is trans-
25 formed into a hit along the tracker wires. A multivariate classifier is used to reject hits
26 produced by a low-momentum Compton electron.
- 27 • Pattern-recognition to select groups of hits compatible with helicoidal trajectories. Two
28 separate methods are used: a calorimeter-seeded and a tracker-seeded algorithm.
- 29 • Track fit to increase the accuracy of the reconstructed track and improve the background
30 rejection. A simplified Kalman fit is finally performed to improve the accuracy of the
31 reconstructed track parameters and further reject spurious candidates.

32 The resulting efficiency for conversion electrons generating 15 or more hits in the tracker
33 is around 98% for the nominal proton bunch intensity, with a background rejection factor
34 greater than 1000, and the efficiency only decreases by a few percent for proton bunch inten-
35 sity up to three times the nominal value. Other trigger streams include for example $\mu^- \rightarrow e^+$
36 conversion electrons, high-momentum decay-in-orbit electrons, high energy photons pro-
37 duced by radiative pion or muon capture, cosmic rays, or electrons from muons stopping
38 in the internal proton absorber.

1 5 Data Handling

2 The data logging system brings data from the DAQ disks to the archival storage facility and
3 distributes it to the various computing elements. The components include a file transfer agent
4 to transfer data from the DAQ system, a replica manager orchestrating transfers between
5 storage elements, a file metadata catalog, and interfaces to deliver the appropriate files to
6 workflow management systems and interactive users. The existing Mu2e data catalog, Se-
7 quential Access via Metadata (SAM), was designed for the previous generation of collider
8 experiment at FNAL, then adopted by many intensity frontier experiments. This infrastruc-
9 ture is currently being replaced by a new system based on Rucio, MetaCat, Data Dispatcher,
10 and DeclaD.

11 5.1 System components

12 MetaCat [26] is a general-purpose metadata catalog storing permanent file description infor-
13 mation, with location and delivery handled by Rucio. MetaCat provides four major functions:
14 store metadata associated with a file; provide a mechanism to retrieve these metadata; effi-
15 ciently query the metadata database to find entries matching a list of predicates; and provide a
16 mechanism to integrate metadata stored into external sources to query the database. The last
17 functionality allows, for example, to seamlessly find a set of files matching criteria stored in
18 a condition or run database. The metadata representation is flexible enough to accommodate
19 a wide range of types and complex structures, and the catalog can scale to several hundred
20 million entries.

21 Rucio [27] is a replica manager system designed to centrally manage large volumes of
22 data backed by many heterogeneous storage backends. This system was originally developed
23 by the ATLAS experiment and has now been deployed to several other HEP and astronomy
24 experiments. In a distributed system where data are physically stored over a multitude of stor-
25 age servers, potentially each relying on different storage technologies (SSD/Disk/Tape/Object
26 storage), Rucio provides an interface enabling users to interact with the storage backends in
27 a unified way. The data can be accessed interactively or in batch jobs, and the closest file
28 replica to the running job is delivered via streaming. Rucio is a rule-based system, allowing
29 users to define high-level rules such as "keep 2 copies on 2 different sites". If one copy is lost,
30 the system will automatically copy the data from a different storage site to maintain the rule.

31 Data Dispatcher interacts with user and production systems to provide file location infor-
32 mation from Rucio to clients. More specifically, it delivers file handles to consumers, keeps
33 track of consumer status and files consumed, provides monitoring and control, and coordi-
34 nates data processing among data consumer processes.

35 DeclaD, the Declaration Daemon, is a file transfer agent used to drain files from a set of
36 directories and store the corresponding information in Rucio and MetaCat. DeclaD is used
37 to transfer files from the Mu2e DAQ disks to dCache. The automated follow-on processing
38 (Pass 1) is discussed in the next section.

39 5.2 Storage

40 Data processing will primarily utilize two types of storage: dCache and tape. dCache is
41 a distributed, multi-petabyte scalable disk storage system with a single rooted filesystem
42 providing location-independent file access. The dCache storage is readable and writable from
43 grid machines via xrootd and transfer mechanisms such as Intensity Frontier Data Handling
44 (ifdh). It is expected to be the main storage element for the output of distributed production
45 jobs. The dCache scratch area is used mainly for output produced from grid jobs and files are

1 automatically removed based on a policy with a typical lifetime of one month. The dCache
2 persistent volume is an area for persistent storage of user files (e.g. files must be removed
3 by their owner). The dCache tape-backed volume is a disk cache sitting in front of the tape
4 system to prestage files written to tape. The tape storage infrastructure provides long-term
5 storage. The current tape storage software, Enstore, provides access to data over the wide area
6 network through the dCache disk caching system. Fermilab is actively pursuing a transition
7 to using the CERN Tape Archive (CTA) software.

8 The files produced by the DAQ system will be written to persistent dCache, and most of
9 them will be immediately copied to tape-backed dCache. The copy in persistent dCache is
10 needed to ensure that the file is on disk when Pass 1 reconstruction starts (see Section 6). This
11 copy will be deleted once processing is complete. Several data streams contain pedestal data
12 from the CRV or pre-scaled raw and intermediate data from the ExtMon and STM. Depending
13 on the DOE data retention policy and the need of the sub-systems, these data might only need
14 to reside in persistent dCache for a short period of time, without being written on tape. The
15 final strategy will be elaborated closer to data taking. Other streams might also produce
16 small data files (e.g. off-spill triggered events or the Error/Debug Stream). These files will be
17 held in persistent dCache for some time, perhaps a few days, and concatenated before being
18 written to tape to make efficient use of this media. We plan to concatenate the Error/Debug
19 stream into a single art file and to make compressed tar files of the DQM output and log files.
20 Once transferred, the persistent dCache copy will be given an expiry date. Finally, a large
21 dCache tape-backed volume may be requested to speed up pre-staging if a large amount of
22 data needs to be read from the tape. This might be the case, for example, if a significant
23 fraction of the data are reprocessed at a later time.

24 **5.3 Job production manager**

25 The production system is used to streamline data processing. It automatically detects new
26 entries in the file catalog and launches the corresponding processing tasks. A prototype of the
27 data production system has been developed with previous generation technologies: SAM (file
28 catalog), FTS (file transfer agent), and POMS (Production Operations Management Service)
29 supporting automated multi-stage workflows and job management. The production system
30 has been in operation for about a year to record and process data from the cosmic ray veto test
31 stands. POMS has also been used extensively in simulation campaigns. The existing system
32 is adequate to support upcoming sub-detector Vertical Slice Tests, cosmic commissioning,
33 beam commissioning, and run 1 processing, within the context of the previous generation of
34 file tools.

35 The transition to the new data handling tools is in progress. Preparations for using Rucio
36 and MetaCat are nearly complete. The new data handling tools however are not supported
37 within POMS, and there is currently no promise by the POMS developers to provide that.
38 Several possibilities are currently being considered to replace POMS by a new job manage-
39 ment system. Requirements on this system have been defined, emphasizing robustness and
40 accessibility to ensure that data production can be easily managed by non-experts. Compati-
41 bility with existing, fully developed workflows is also a priority. A decision on the choice of
42 technology is expected in the near future, with the goal of producing a first prototype well in
43 advance of the cosmic ray data taking.

44 **5.4 File family plan**

45 Mu2e faces two types of file management challenges: efficiently pre-staging large data sets
46 from tape, and managing data sets containing many small files. The first challenge will

1 be faced only starting at the beginning of data taking with beam. To mitigate this issue,
2 large data sets will be segregated into their own file families, and pre-staging will be tested
3 using the new data handling tools and tape system (RUCIO, dCache, and CTA). This system
4 should offer much better performance than the current architecture based on SAM, dCache,
5 and ENSTORE. Tests could be conducted soon after CTA is available (scheduled for early
6 CY25), giving enough time to work with the CSAID Storage Group to tune the process. The
7 fallback position is to use migration mode, similar to the g-2 experiment.

8 The handling of small files presents another challenge as *small file aggregation* (SFA)
9 packs cannot be written with CTA anymore. Potential strategies to concatenate small files
10 will be tested once CTA is available. Mu2e is also following the recent work performed by
11 the CSAID storage team regarding the impact of file size on CTA performance and lessons
12 learned will be included in the overall management strategy.

13 Table 2 gives a preliminary list of tape file families Mu2e will use. Some file families will
14 be primarily archival and only rarely accessed (e.g. log files or DQM data). As such, they
15 do not need to be finely divided. Two copies of the files in the Raw Data file families will be
16 kept on tape at physically separate locations, and a single copy will be stored for all other file
17 families.

18 As Mu2e gains experience and codes improve, data will be reprocessed and re-simulated,
19 creating multiple generations of data. New file families will be created for each new generation
20 of files for large data-volume outputs produced by reconstruction and simulation. A single family
21 will be used for smaller data sets for the duration of the experiment. Beam data
22 will be split into separate families for run 1 and run 2 .

23 Additional file families for Pass 1 will be defined for the CRV, STM, and ExtMon
24 sub-systems once their data processing workflows are finalized. The "Normalization data
25 streams" contain files from the Intensity Stream, the STM pulse summary data, and the summary
26 data from the ExtMon (both Pixel and AFF). These files are written to a single file
27 family since they will be analyzed together, and files close in time should be written close
28 together on tape.

29 For the simulation campaigns, all events will be recorded as analysis format data and a
30 subset of events will be recorded with the full simulation and reconstruction outputs. Files
31 from user simulation should be small enough to fit on a single tape volume, as user simulation
32 campaign producing significantly higher volumes would be promoted to a collaboration
33 simulation campaign. The size of Analysis Format data is expected to be small enough to fit a
34 generation on less than a single tape volume. We expect this plan to evolve as the workflows
35 are developed in more detail.

Raw Data (2 copies)	Trk+Cal+CRV triggered events on-spill Trk+Cal+CRV triggered events off-spill Normalization stream Calorimeter pulser events CRV zero bias triggered data ExtMon prescaled R&I data (Pixels & AFF) STM waveforms for selected pulses STM prescaled R&I data Other raw data
Pass1_N	Full reconstructed output on-spill Full reconstructed output off-spill Physics stream on-spill Random on-spill trigger stream Tracker calibration on- and off-spill Calorimeter calibration on- and off-spill
Pass2_N	Full reconstructed output on-spill Full reconstructed output off-spill Physics stream on-spill
Reco_N	Other reconstructed data
Analysis_N	Reconstruction format skims (per channel) Analysis format skims Display format skims
Simulation_N	Physics and background primaries Muon, Electron, and Neutral Beam Stopped Muons Digitized simulated events Reconstructed simulated events Analysis format simulated events
User Files	Data skims, reconstruction format Data skims, analysis format Simulation skims, reconstruction format Simulation skims, analysis format
Other	Archived Online and Nearline DQM data Archived Online and Nearline logs Archived Offline DQM data Archived Offline logs Archived simulation production logs

Table 2. The Mu2e plan for tape file families. A unique file family for each data processing cycle will be used for streams labeled "_N".

6 Data Processing

This section covers the workflow to process the data produced by the DAQ system, illustrated in Fig. 8. The files written on the DAQ buffer disk are automatically transferred to offline storage, and the first reconstruction pass is run with the same conditions used by the trigger algorithms. Once completed, the calibrations constants are re-evaluated and updated in the conditions database. The main physics streams are then reprocessed using the latest conditions, and reduced datasets are finally produced for further analysis. These steps proceed in parallel: newer data are processed through the first reconstruction pass while older data are simultaneously reconstructed with the second reconstruction pass. The workflow is optimized to minimize the need to pre-stage files from tape and maximize the data throughput. Each stage is discussed in greater detail in the remainder of this section.

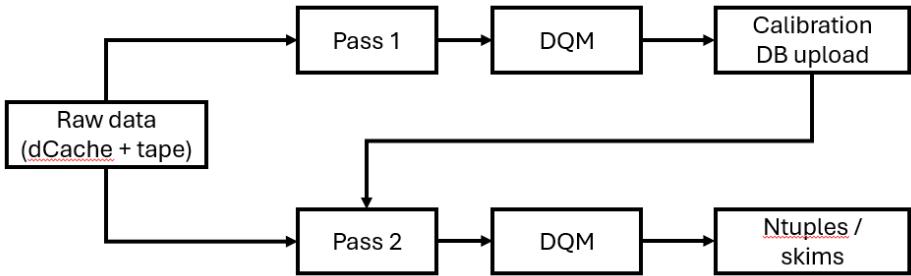


Figure 8. Schematic of the Mu2e data processing workflow

6.1 Pass 1

Pass 1 is the first stage, starting almost immediately after the data have been transferred from the DAQ buffer disks to dCache. It will process the totality of triggered events through the full set of offline reconstruction algorithms using the same conditions information used by the trigger algorithms. The following outputs will be produced:

1. The main physics output file, which will contain events with signal-like tracks, for both $\mu^- \rightarrow e^-$ and $\mu^- \rightarrow e^+$, plus events of interest for background studies.
2. Calibration files with events needed for the various calibration algorithms.
3. Offline DQM information (see Section 15) with higher statistics and more complete coverage than that provided by the online DQM system.
4. Optionally, a file containing all data for debugging and validation purposes

When necessary, the workflow will include stages to concatenate small output files into a single large file, e.g., off-spill events (there is a competing demand to produce prompt offline DQM information, and the optimal concatenation policy will be determined by the operational constraints).

Once a new run is started, the Pass 1 workflow will start as soon as selected information from the online database is available offline. As the data logging system delivers files, the production system will monitor the file catalog and launch grid jobs to process new files (either a single or many files in one job). Under normal conditions, the jobs should be launched

1 10 minutes or less after the files have been cataloged and complete within 2 to 3 hours af-
2 ter submission. However, this time distribution is expected to have (long) tails during peak
3 usage periods or if the system requires maintenance. Offline DQM information will be sum-
4 marized in both timelines and metrics, and out-of-tolerance data will generate an alarm. The
5 histograms and TTrees from which these metrics are derived will be retained on disk for a
6 certain time and persisted on tape. Files in the Error/Debug stream will also be copied to
7 tape, concatenated if necessary, and retained on disk for inspection by experts for a limited
8 amount of time.

9 The normalization file stream includes the Intensity stream and the summary data from
10 the ExtMon and the STM. These streams contain information to evaluate the proton bunch
11 intensity variations over the course of each spill. These data are needed to assess the recon-
12 struction efficiency accurately and properly normalize the results. These streams have sparse
13 per-event information (i.e. not all information is available for every event). In addition, pack-
14 aging small data products as objects in an art : :Event is inefficient in both disk/tape space
15 and access time. One possibility might be to organize the data around spills, and the resulting
16 information could be persisted in a database or included in each art SubRun object. Final
17 decisions will be taken as the design matures. For these files, Pass 1 will package the infor-
18 mation in a way that is convenient for downstream processing, and concatenate input files if
19 necessary.

20 The first class of low-level calibration streams includes the prescaled raw and interme-
21 diate data from the monitors and the calorimeter pulser events. The corresponding Pass 1
22 workflow will be defined by the detector teams, but the output should include DQM and up-
23 dated condition information that will be uploaded to the condition database. The input files
24 might only be stored in dCache for a limited amount of time. The second class of low-level
25 calibration streams contains STM waveform data for pulses with energies near the important
26 lines. The STM team plans to periodically refit these pulses with updated conditions infor-
27 mation to improve the estimate of the stopped muon yield. This data will be written to tape,
28 with file concatenation when necessary.

29 **6.2 Calibration**

30 The data workflow continues with the calibration step, using the data directly produced in
31 Pass 1 to update the detector calibration constants. A detailed description of calibration pro-
32 cedures is given in Section 9. Most of the time, the output of Pass 1 should be sufficient
33 to perform this work. However, it may be necessary to re-run a subset of Pass 1 using the
34 conditions information determined by the previous pass, and repeat the procedure until the
35 calibration information has converged. This might be required, for example, during com-
36 missioning and early data taking. When necessary, the subsystem teams will be able to run
37 repeated Pass 1 jobs using the data production system, or by manual invocation of particular
38 components of it. Once calibrations are certified, the offline condition database is updated.
39 The online condition database will be periodically synchronized to the offline database.

40 **6.3 Pass 2**

41 Once calibration constants for a time interval are certified, the main physics streams (on-spill
42 and off-spill triggered events) are reprocessed using updated conditions information. This
43 operation is called Pass 2 and will produce a new set of DQM information and output data
44 files. During commissioning and early data taking, Pass 2 might perform the full reconstruc-
45 tion from the raw data. As we gain experience, it might become possible to only refit existing
46 tracks and recalculate the parameters of existing calorimeter clusters and CRV stubs. In that

1 case, the raw data in the output of Pass 1 could be filtered to only include the necessary
2 information.

3 Based on experience with previous experiments, Mu2e expects the following three-week
4 cycle at the start of the experiment: during week N, take data for week N, run calibration jobs
5 for data taken during week N-1, and run Pass 2 for week N-2 data. As data taking becomes
6 stable, it may be possible to compress this time scale. During a year-long run, we also expect
7 continuous improvements in calibrations and reconstruction algorithms. When integrated
8 improvements warrant it, we will re-run Pass 2 on all recorded data. In this situation, a
9 large dCache tape-backed volume and additional drives may be requested to speed up file
10 pre-staging.

11 **6.4 Reduced data sets**

12 The data workflow ends with the production of reduced data sets. The format and use cases
13 are discussed in Section 10. There may be several types of datasets, each targeted to a particu-
14 lar analysis task. These data sets will be directly produced from the output of Pass 2, running
15 additional reconstruction algorithms if needed. They will be produced for every iteration of
16 Pass 2. As analysis algorithms evolve, reduced data sets will be reproduced to reflect these
17 improvements. The output of Pass 2 and reduced data sets should be small enough that this
18 step will not require significant resources.

19 Additionally, skim data sets could also be produced from the output of Pass 2. Each skim
20 data set contain a fraction of the original events, selected by a dedicated filter and preserving
21 or augmenting content of the original event. These data sets would reduce computing needs
22 and processing times during the analysis phase.

1 7 Simulations

2 Prior to the start of operations, the main purpose of Mu2e simulation was to verify the ex-
3 pected performance of the experiment and to allow optimization of the detector design. That
4 required an accurate, detailed, and flexible model of the experiment. In the analysis phase, the
5 simulation will be used to optimize alignment, calibration, and reconstruction algorithms, to
6 estimate the expected detector acceptance and resolution, and to develop analysis algorithms.
7 The Mu2e simulation was designed to satisfy both of these objectives.

8 The Mu2e simulation is based on the Geant4 framework [28–30], including a description
9 of the *as built* geometry and the material composition for both the sub-system parts and the
10 experimental hall; the temporal and spatial structure of the proton beam; magnetic field maps;
11 the implementation of the detector response as measured at beam tests and with cosmic rays;
12 and a complete interface with the calibration database. A set of custom event generators
13 dedicated to the conversion electron channel and the main sources of background are also
14 used, and full events are simulated with overlapping beam background particles. Additional
15 studies have been performed using other Monte Carlo codes (MARS[31], MCNP[31, 32],
16 FLUKA[33], PHITS[34]) to validate Geant4 predictions and to estimate their uncertainty.

17 The following sections describe the simulation workflow, the detector modeling, the
18 physics generators, and the production campaigns in more detail.

19 7.1 Simulation workflow

20 The simulation of the large number of events expected in Mu2e requires some optimization
21 of the processing time. This is achieved with a staged simulation approach together with
22 resampling techniques. Given the statistical nature of the particle interactions in the material
23 crossed before reaching the detectors, it's possible, where appropriate, to reuse many times
24 the particles stored at the end of one stage (*resampling*) increasing the available statistics at a
25 reduced cost of CPU time. The absence of statistical biases (*oversampling*) must be checked
26 case by case. The staged approach helps to save additional CPU time when no changes occur
27 in the first stages and it's sufficient to rerun only the last stages. In addition, different particle
28 range cuts can be used in each stage to avoid unnecessary particle tracing.

29 The main beam simulation is divided into four stages:

- 30 1. from the 8 GeV proton beam on the Production Target (PT) to the entrance of the
31 Detector Solenoid (DS);
- 32 2. from DS entrance to the stopping target;
- 33 3. from the stopping target to the exit from the DS volume (tracker, calorimeter, and
34 cosmic ray veto hits are stored);
- 35 4. detector response and digitization.

36 The initial beam protons are given the nominal beam energy and direction, distributed
37 across the PS entrance as a symmetric Gaussian with the width expected from beam simula-
38 tions. The time of each proton is sampled from a distribution generated by detailed simula-
39 tions of the slow extraction and active extinction system. To save processing time, charged
40 particles that exit the PS volume forward are killed.

41 Special workflows have been developed to optimize the production of dedicated samples
42 as well. Pion decay can be disabled to increase the number of stopped pions in the stopping
43 target, and the stopped pion probability is then corrected for the pion survival probability
44 depending on its lifetime. The first stage of the antiproton simulation has also been divided

1 into four phases to use a customized differential cross section for antiproton production and
2 optimize the absorber window located at the center of TS. The cosmic ray simulation re-
3 quires a dedicated workflow as well. In the first stage secondaries produced by the CRY
4 generator [35] according to the flux at sea level are traced from the top of the Mu2e building
5 to the DS. Events are divided into three categories according to the energy deposited in the
6 Cosmic Ray Veto (CRV): high ($E > 16$ MeV), low ($E < 16$ MeV), or null (neutrons or parti-
7 cles passing through the TS hole). In the following stage, particles are resampled and traced
8 through the detector. Finally, digitization is performed. In all cases, only the events with a
9 minimum energy deposit in the tracker, the calorimeter, or the cosmic ray veto are passed to
10 the digitization stage.

11 **7.2 Event generators**

12 A set of generators has been realized to have a more accurate simulation of the particle yield
13 from the stopping target. Starting from the position and time of the stopped muons or pions,
14 the following single particles can be generated with isotropic direction: conversion elec-
15 tron (Leading Log spectrum [13, 14]), conversion positron (Leading Log spectrum [13, 14]
16 rescaled to the expected peak momentum of 92.3 MeV/c), DIO (Leading Log spectrum [36]),
17 muon capture secondaries (protons and deuterons spectrum adapted from Ref. [37, 38], neu-
18 tron spectrum adapted from Ca measurements [39] and normalized according to Ref. [40],
19 x-rays and gamma rays used to study the Stopping Monitor response are generated as sin-
20 gles lines), radiative muon capture secondaries (photon spectrum from the fit of experimental
21 data [41] with the closure approximation [42]), radiative pion capture secondaries (photon
22 spectrum adapted from Mg data [43]). Cosmic ray studies are performed with the CRY gen-
23 erator to evaluate the cosmic background and the CORSIKA generator [44] to estimate the
24 uncertainty on flux normalization. A custom generator to produce antiprotons from protons
25 interaction in the production target has been developed fitting the existing data and extrapo-
26 lating them to the 8 GeV Mu2e beam energy [16].

27 **7.3 Pileup simulation**

28 In addition to simulating particles of interest for physics studies and calibration, Mu2e must
29 model the pileup of the many low energy particles produced by beam particles and de-
30 cay/capture products of the vast majority of stopped muons which do not convert to elec-
31 trons, or produce a particle which could be a signal background. Pileup is not itself a source
32 of background for physics analyses, but the detector signals it generates can significantly
33 affect the reconstruction efficiency and/or resolution of signal-like particles.

34 Mu2e simulates pileup using Geant-4, starting with protons hitting the production target
35 (POT), as described above. Charged particles traveling backward and entering the TS are
36 check-pointed as they enter the DS. Neutral particles that exit the beamline are check-pointed
37 as they exit the TS or PS volume.

38 In subsequent simulation jobs, each check-pointed particle’s passage through the DS is re-
39 sampled 1000 to 10000 times, depending on the sample. Particles producing energy deposits
40 in the detector sensitive volumes are saved, along with their genealogy and energy deposits.
41 To allow correct normalization when used downstream, the fraction of saved events is calcu-
42 lated automatically during the resampling, relative to the original number of simulated POT,
43 and stored in the conditions database.

44 Resampled muons stopping in the stopping target are check-pointed before they decay and
45 then resampled 10000 times in turn to model pileup from muon capture and decay daughters.
46 The muon lifetime is randomly sampled for each resampling, and custom codes are used to

randomly select a decay mode. These same check-pointed target-stopped muons are also used as starting space points in the signal generators described above. The efficiency of resampled stopped muons producing energy deposits in sensitive volumes is computed and recorded in the database. The breakdown of signals expected from simulated pileup by time and stream in individual Mu2e detector sensor elements (straws for tracker, etc.) is shown in Fig. 9.

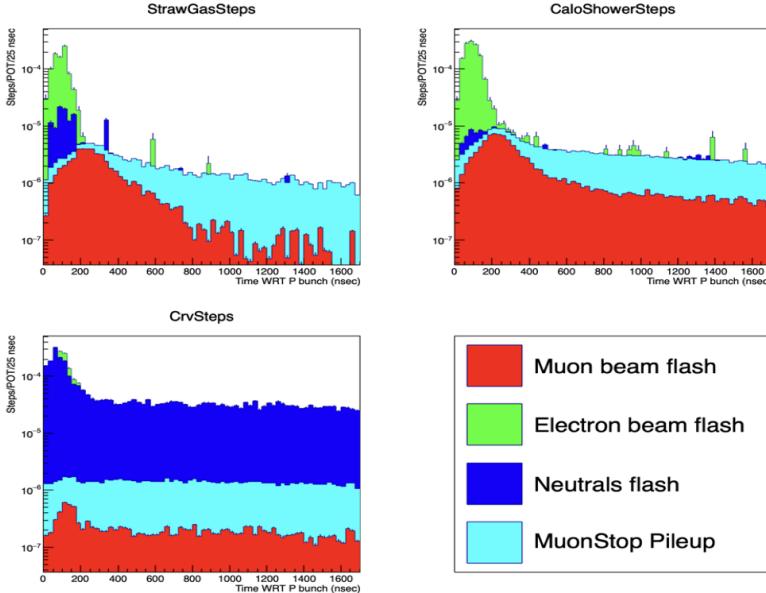


Figure 9. Simulated signal counts in individual detector elements (steps) per POT relative to the arrival time of protons on the production target, wrapped to the beam pulse period, for the four pileup streams described in the text.

Pileup in the CRV is dominated by the neutral particles (mostly neutrons) leaving the PS. As the cross-section for generating signals in the CRV is small, these can be resampled many times, currently 10000. A physics list including detailed neutron interactions is used for this stage. Neutral CRV pileup resampling dominates the resources needed for pileup simulation, as the required statistics are large and the processes involved are time-consuming to model.

Pileup is overlaid on signal events by adding the sensitive volume energy deposits from an appropriate number of pileup daughters to the energy deposits generated by the signal process particle. Pileup data are read through art secondary input streams, in sequential mode, starting from a random starting event. The number of pileup daughters needed is computed for each event by sampling the expected POT distribution, scaling that by the products of the relevant resampling efficiencies read from the database, and then applying Poisson fluctuations. The resulting summed energy deposits are then passed through the detector response simulation. Since the proton pulse intensity is expected to be relatively constant over millisecond timescales, and the pulse trains are long, the time of each pileup particle energy deposit is wrapped around the 1695 ns bunch period when summing.

Most Mu2e pileup simulations use a POT intensity distribution modeled by a log-normal distribution with Spill Duty Factor (SDF) = 60% [16], scaled to the average beam intensity expected for operations with either 1 or 2 booster batches. A time sequence of POT com-

puted using a detailed beam extraction simulation can also be used. The workflow for pileup simulation from POT and subsequent mixing onto simulated signals is shown in Fig. 10.

Even when leveraged by resampling, simulating pileup from POT is very inefficient. Using $\sim 7M$ core-hours of processing time, the most recent simulation campaign produced roughly 5 seconds (1.3 seconds) of experiment running time (beam live time). These samples also show statistical artifacts due to over-sampling of some particles. To avoid these limitations, Mu2e is developing code to overlay pileup extracted from beam data on simulated signals. By triggering randomly, Mu2e can record essentially unlimited pileup events with essentially zero resource cost that will naturally track the real POT intensity, and follow changing detector conditions. Unlike simulated signals and pileup, which are merged prior to digitization, pileup from beam data will be already digitized. Digital pileup that don't overlap with simulated signal particle energy deposits in time or channel can be merged trivially. Digital pileup that overlaps in both time and channel with simulated signals will be handled by dedicated codes. For the tracker, at the expected nominal beam intensity with 1 booster batch, 98% of pileup signals are non-overlapping.

The event identity of mixed (digital) pileup plus simulated (signal) events is given by the simulated event, to allow digitized pileup frames to be reused, and to distinguish them as Monte Carlo. Most conditions data used in simulating the signal and reconstructing the combined event are keyed to the pileup. This includes dead or noisy channels, sensor and electronics response, and resolutions. For effects which are not simulated, such as individual straw miss-alignments, conditions data are taken from the simulation. Preliminary studies overlaying digitized simulated pileup on simulated conversion electrons, using a very naive overlap resolution algorithm, shows nearly identical results post reconstruction as energy deposit pileup overlay. Digital pileup overlay is currently only implemented for the tracker. Extension of the digital pileup overlay algorithms to include the calorimeter and CRV, and development of more sophisticated algorithms for treating overlapping signals, as planned for the immediate future. The workflow for mixing pileup from random triggers onto simulated signals is shown in Fig. 11.

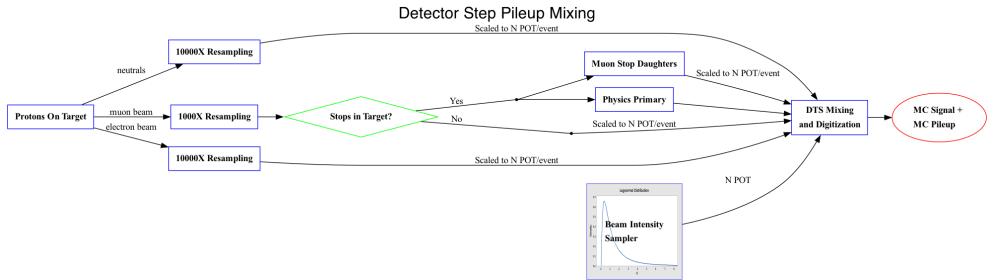


Figure 10. Workflow for producing simulated signal events with overlaid simulated beam pileup.

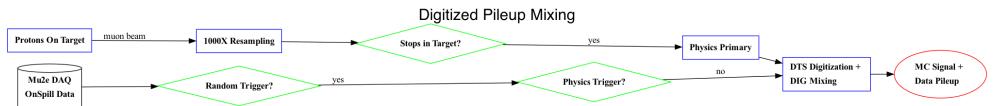


Figure 11. Workflow for producing simulated signal events with overlaid beam pileup extracted from data.

1 **7.4 Geant4 Physics processes**

2 The main physics interactions are simulated using a modification of the GEANT4 "Shielding"
3 physics list. The main modification consists of increasing the threshold used to pass from
4 the Bertini Cascade (BERT) model for low energy hadron-nucleus interactions to the Fritiof
5 (FTF) model from 4-5 GeV to 9.5-9.9 GeV to better reproduce the experimental data on pion
6 production. Nuclear de-excitations and radioactive decay of long-lived isotopes are included.
7 Hyperon and anti-baryon production is obtained from the chiral invariant phase space model
8 (CHIPS). Neutron simulation uses the high precision model (HP) up to 20 MeV. This is
9 particularly relevant for the prediction of the thermal neutron background.

10 Geant4 is used only to produce the electrons and the photons from the atomic cascade
11 to the ground state for muon capture in the aluminum stopping target. The nuclear muon
12 capture, the muon decay in orbit, the radiative muon capture, and the muon conversion are
13 simulated using custom generators (see below). The same holds for the radiative pion capture
14 in the stopping target. Simulation of muon and pion captures outside of the stopping target is
15 generally managed by Geant4, but custom generators have been created for particular studies
16 related to detector calibration.

17 Additional Monte Carlo codes are used to validate and evaluate systematic uncertainties
18 in the predictions of critical quantities obtained with Geant4. MARS has been extensively
19 used to study the radiation levels in the Mu2e hall, the effectiveness of the concrete shielding
20 and of the Heat and Radiation Shield, and the dose and neutron fluence in the tracker and
21 calorimeter electronics. The effect of concrete shielding on neutron and kaon radiation has
22 also been studied with FLUKA and MCNP. Pion production in the PT has been investigated
23 with MARS, MCNP, FLUKA, and PHITS, while Geant4 expectation for antiproton produc-
24 tion in the production target has been corrected given its disagreement with MARS, MCNP,
25 and FLUKA [16].

26 **7.5 Geometry description**

27 The experimental hall as designed for the first run of the experiment is shown in Fig. 12.
28 The main information obtained from the simulation of the passive parts is the effectiveness
29 of the shielding against the radiation produced by beam interactions and cosmic rays. The
30 simulation includes the dirt surrounding the detector hall, the building walls, the concrete
31 shielding blocks, the solenoids warm and cold mass, the radiation shield, the mechanical
32 structure of the tracker and calorimeter, the beam dump downstream of the proton beam with
33 the extinction monitor, ... The amount, type, and location of shielding have been decided as
34 a compromise between budget considerations and the amount of radiation acceptable for the
35 different sub-systems. It will be reviewed for Run II according to the result of Run I.

36 The geometrical description of each sub-system is maintained by the corresponding work-
37 ing groups. The level of detail of the geometry description is a compromise between the time
38 needed to simulate the events and the agreement between data and Monte Carlo simulation.
39 The main parameters (e.g. material composition, number of elements, single element dimen-
40 sion, and location) are included in the geometry database. The dimensions and positions of
41 the active parts (tracker straw tubes, Ecal crystals, ...) are the nominal ones: mechanical
42 tolerances, gravitational sags, or misalignments are introduced at the digitization level.

43 A separate geometry package has been realized for the detector commissioning with cos-
44 mic rays. The simulation of the experimental setup in this *extracted position* includes the
45 tracker and calorimeter out of the Detector Solenoid and a few modules of the Cosmic Ray
46 Veto on top of them (Figure 12(right)). Alternative geometrical descriptions for the detec-
47 tor have been realized using MARS and FLUKA to determine the uncertainty of simulation
48 estimates of the radiation levels and the stopped muon yield.

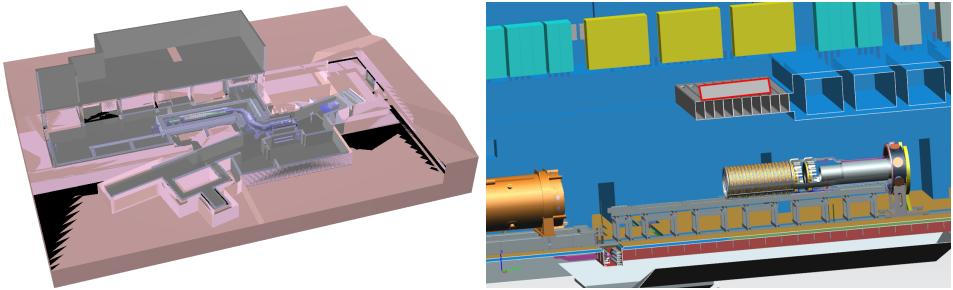


Figure 12. Left: the Mu2e experiment and building in the Geant4 Offline simulation. Right: Mu2e detectors in the extracted position.

1 7.6 Magnetic field maps

2 Detailed magnetic field maps are produced using the OPERA 3D [45] and HELICALC [46]
 3 software packages using (warm) as-measured conductor geometries and coil positions pro-
 4 vided by the Mu2e project solenoid construction team. The field is stored as a discrete vector
 5 map over a Cartesian grid, separately for the Production Solenoid (PS), Transport Solenoid
 6 (TS), Detector Solenoid (DS), and the region outside all the solenoids. The effect of re-bar
 7 in the concrete shielding has been calculated but not yet included. Field maps for special run
 8 conditions used for calibration purposes have also been generated. The field map is accessed
 9 through a software interface that dynamically interpolates the field vectors from the 9 grid
 10 points nearest the requested reference point.

11 The Mu2e operations team will determine the in-situ DS magnetic field using a custom
 12 survey device that will measure the magnetic field components at fixed locations within a
 13 cylindrical coordinate grid. A fit to the measured data using a combination of a multi-pole
 14 expansion and relaxation ANN constrained to Maxwell's equations will be used to interpolate
 15 and average the measurements into the Cartesian grid format. This map will be corrected for
 16 the estimated effect of re-bar in shielding materials that can't be in place during the field
 17 survey.

18 Publication quality simulations and reconstruction will use the field map determined from
 19 the survey. Changes in physics objects reconstructed using alternative maps spanning the
 20 measurement and shielding magnetization uncertainties will be used to estimate the system-
 21 atic errors on physics measurements due to residual field map uncertainties.

22 7.7 Detector response

23 Each sub-system has a dedicated simulation of its digitization. This includes a parameter-
 24 ization of the physics processes bringing the signal from the position where the energy is
 25 deposited to the readout electronics and also the simulation of the electronics response. Sim-
 26 ulated digitization uses the timing information from the (simulated) DAQ clock system used
 27 to define the start and stop of each digitization period, including the 5-fold repeating pattern
 28 of different event start times and lengths caused by the period mis-match of the booster and
 29 the delivery ring clocks.

- 30 • Tracker - energy deposits in the straw tubes from different particles are collated and pro-
 31 cessed by applying a parameterized simulation of the drift time, gas amplification, signal
 32 transit time along the wire, electronic amplification, and shaping. Electronic signals at the

1 straw ends are superimposed and the resultant waveform is digitized. Only digitized hits
2 passing the discrimination threshold at both ends are saved.

- 3 • Calorimeter - energy deposits in each crystal are grouped in time. The number of photons
4 produced by scintillation takes into account the average light response uniformity measured
5 during crystal qualification. The number of photo-electrons obtained using the average
6 PDE of the SiPMs is smeared with a Poisson distribution. The digitizer waveforms corre-
7 sponding to energy deposits in the same readout unit are superimposed. Radiation-induced
8 noise and electronic noise are added. Zero suppression is reproduced.
- 9 • Cosmic Ray Veto - energy deposits in each module are grouped in time. The bunch time
10 structure is applied (only for the on-spill simulation). The time distribution of the charge
11 collected by the SiPMs located at the edge of each module is calculated using the scin-
12 tillator light yield, the WLS collection efficiency, the light attenuation along the fiber, the
13 reflection probability on the walls, and the SiPM photon detection efficiency. The charge
14 collected by the SiPMs is transformed into signal waveforms after applying a random time
15 jitter and photo-electron fluctuation. ADC and TDC counts are obtained by applying wave-
16 form sampling and discrimination.
- 17 • Stopping Target Monitor - energy deposited by each photon in the HPGe detector is used
18 to generate a waveform according to the deposited ionizing energy, measured gain, and
19 collection efficiency. Waveforms from different photons are superimposed using the ex-
20 pected time and energy distribution. Electronic noise from the pre-amplifier and amplifier
21 is added. The resulting waveforms are digitized using the 16bit ADC, and are used together
22 to form STMWaveformDigis. A similar technique will be used for LaBr response.
- 23 • Extinction Monitor - energy deposits are converted into electron-hole pairs. The number
24 of charge carriers is fluctuated using the silicon Fano factor of 0.1. The deposited charge is
25 split into a number of clusters uniformly distributed along the particle path inside the sen-
26 sor. Each cluster is drifted individually. The sum of the charge reaching the pre-amplifier
27 with its time structure is passed to a discriminator that simulates the time and time over
28 threshold response of the readout chip. At the final stage, the digitization algorithm adds
29 random hits to the output [47].

30 **7.8 Simulation campaigns**

31 Several simulation campaigns have been conducted since the inception of the experiment to
32 guide the design and evaluate the corresponding physics performance. The SU2020 cam-
33 paign was based on workflows developed in 2018, and the output was used to update the
34 sensitivity estimate of the experiment for expected run 1 operations. That work pulled in ef-
35 fort from a number of younger collaborators, and was published [16]. The most recent cam-
36 paign, MDC2020, used updated geometries, detector response simulations, and a streamlined
37 workflow to generate large samples of beam, pileup, signals, and physics backgrounds. The
38 principal output of MDC2020 was a large sample of conversion signals with pileup, used
39 in algorithm development, plus pure pileup events, for trigger studies. MDC2020 used the
40 Production Operations Management Service (POMS) to run complex grid campaigns, and
41 GitHub to store scripts and configuration files. The whole effort was driven by the Mu2e
42 Production Manager. Different sets of calibration and alignment conditions were simulated
43 to evaluate their impact on the reconstruction efficiency. Most of the conditions were also
44 directly extracted directly from the database, providing a test of the condition system in a pro-
45 duction setting. These data have proved essential to improve and evaluate the performance
46 of reconstruction algorithms, including those used in the trigger, and perform the SU2020
47 sensitivity study [16].

1 **7.9 Ensembles**

2 While useful for software development and some physics studies, individual background
3 streams produced in MDC2020 are not representative of the mixed-signal samples that will
4 be used for analysis. Mock Data samples were introduced to provide a more accurate model
5 of what Mu2e will record during data-taking phases. These samples are produced by merg-
6 ing individual background sources (Pile-up, cosmic rays, DIO, RPC, ...) into a data collection
7 with mixed content, according to event event fractions expected in real data. The combined
8 sample is then passed through our digitization and reconstruction framework to create "data-
9 like" samples. Ensemble creation uses a dedicated art input module that knows how to sam-
10 ple multiple input files. In addition to the development of calibration procedures and analysis
11 frameworks, the ensemble data sets will also enable test of the computing infrastructure in
12 more realistic conditions. Production of large ensemble data sets for dry-run tests of the of-
13 fline calibration and processing workflow at scale is foreseen before the start of Mu2e beam
14 running.

1 **8 Event Reconstruction**

2 Algorithms for converting Mu2e raw data into physics objects (i.e. reconstruction) have
3 been in development for over 14 years. Operating on the output of the detailed simulations
4 described in section 7, these algorithms have been used to quantify the expected performance
5 of the detector [1] and the physics reach of the experiment [16]. The output of reconstruction
6 is used to perform the high-level calibration and alignment discussed in section 9, and as the
7 basis of the physics analysis processing described in section 10.

8 Reconstruction of the Mu2e primary event data stream proceeds in several stages. First,
9 the low-level digital data (digis) coming from each Mu2e subsystem are converted into hit
10 objects, which present the equivalent information in physical units, using calibration objects
11 from the conditions service (see section 12). A sequence of algorithms then aggregates and
12 filters these hits into increasingly complete and accurate representations of physical particle
13 candidates. The final versions of these particle candidates, with their ancillary information,
14 are stored for downstream analysis. The reconstruction sequence relies on AI/ML at several
15 stages, pointed out in the text.

16 Data from the extinction monitor and stopping target monitor are reconstructed in dedi-
17 cated applications. Their final outputs will be recorded in the conditions database, associated
18 with the corresponding event data by their intervals of validity. Reconstruction algorithms
19 are currently under development.

20 All Mu2e reconstruction is implemented within the art framework [25]. Raw data from
21 the DAQ system are stored as compressed digitizations with associated headers, called frag-
22 ments, which are reformatted into Offline digi collections before reconstruction begins. Sim-
23 ultated data are produced directly as Offline digi collections. Individual reconstruction stages
24 are implemented as art producer or filter modules. Wherever possible and applicable, stan-
25 dard utilities from stl, root GPL, and other public sources are used. Specialty codes, in
26 particular the final Kalman filter track fit, are linked as external utilities.

27 **8.1 Tracker Hit Reconstruction**

28 The tracker digi data consists of separate TDC and Time Over Threshold (TOT) measure-
29 ments from both ends of the straw and a digitized ADC waveform from the analog sum of
30 the signals from the two straw ends. Raw data are zero-suppressed by requiring threshold
31 crossings on both ends before readout. The TDC LSB and bin size calibration are defined in
32 the digitization FPGA firmware. Each TDC values are separately converted to physical time
33 using a linear function obtained from the conditions service, keyed on event ID and channel.

34 TOT is recorded online in units of readout clock ticks, counted from when the analog
35 signal rises past threshold to when it falls below it. It is converted into an estimate of the drift
36 time using a non-linear function obtained from the conditions service.

37 ADC waveforms are converted into an estimate of the deposited ionization energy from
38 the particle crossing the straw. Several algorithms have been explored for this, including
39 multi-peak fits to the waveform. By default, and in the trigger, the ionization energy is es-
40 timated by scaling the difference between the waveform maximum and the average of the
41 waveform pre-samples (before the TDC threshold is crossed), which has been shown to be
42 accurate in over 99% of cases. Calibrations for electronic and physical gains are obtained
43 from the conditions service, keyed by event ID and channel.

44 The reconstructed times, TOT, and ionization energy from each straw digitization are
45 saved as straw hit objects. A downstream module averages hits in adjacent straws and close
46 in time into a combined panel hit object. Panel hits provide a more accurate longitudinal
47 position, and reduce the combinatorics of downstream algorithms. A subsequent module

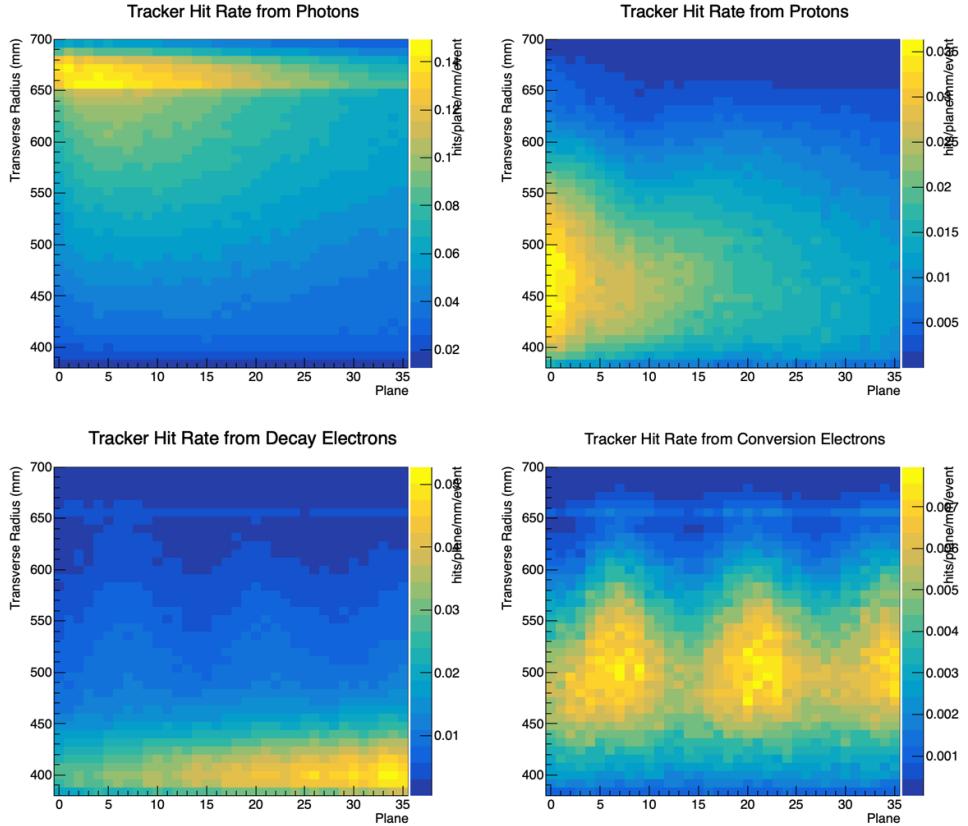


Figure 13. Track hit rate *vs* plane number and reconstructed transverse radius from different sources in simulated conversion electron events with beam pileup overlay.

1 combines nearby panel hits in a tracker station into stereo hits, with more accurate position
 2 and some directional information. Straw, Panels, and Stereo hits all are represented by a
 3 single class so that downstream pattern recognition algorithms can pick the most appropriate
 4 to use without recoding.

5 The majority (>95%) of reconstructed tracker hits come from beam pileup; isolated hits
 6 from photons, clusters of hits from spiraling δ -rays and Compton electrons, and hits from
 7 Decay In Orbit (DIO) electrons and protons coming from stopped muons.

8 A 2-D image of reconstructed hit positions from a typical Ce signal plus pileup event
 9 is shown in Fig. 13. The majority of these hits must be filtered out before tracks can be
 10 successfully found.

11 Hits from protons are removed using their large energy deposit. Hits from DIO are sup-
 12 pressed by requiring a minimum transverse radius, while hits originating from neutron cap-
 13 ture in the DS are suppressed by requiring a maximum transverse radius. Hits from low-
 14 energy electrons are filtered using a dedicated algorithm that first clusters the hits in time
 15 and transverse position and then classifies the clusters using an ANN trained to separate low-
 16 energy from high-energy electrons by their geometric properties. The filtered hit collection

1 passed to track pattern recognition has a signal purity of roughly 50% in signal plus pileup
2 simulation, inside the time window defined by the conversion electron hits.

3 **8.2 Signal Track Reconstruction**

4 The high-momentum track candidate search starts by looking for clusters of filtered hits with
5 times within a sliding ~ 50 ns window. The hit time resolution is improved by roughly 30%
6 by subtracting the TOT-based drift time estimate. The hit straw Z position is used to correct
7 the particle propagation time, dependent on the assumed or measured particle velocity.

8 Hits in a time cluster are passed to one of several helix finding algorithms, which extract
9 rough helix parameters from the hit 3-D positions. Helix reconstruction is split into transverse
10 (circle) and longitudinal phases, which are iterated between until convergence. One algorithm
11 uses the position of high-energy (>50 MeV) calorimeter clusters and the stopping target to
12 seed the helix search, resulting in higher purity, but lower efficiency. The others use purely
13 tracker information. Helices found by different algorithms that share the majority of their hits
14 are merged. Helices from all algorithms use the same output data class. The performance of
15 the different finding algorithms are similar, but not identical: in particular, the calorimeter-
16 seeded algorithm is more robust against very high levels of pileup. As these algorithms
17 are still being developed and tuned, and as they are easy to swap in and out and combine,
18 Mu2e has not yet made a final decision on which ones will be used in the final trigger and
19 reconstruction sequence.

20 Reconstructed helices are passed to a Kalman filter track fit to make final selections. The
21 Mu2e fit uses the KinKal [48] package, which is designed for precision kinematic recon-
22 struction of low-momentum tracks in graded magnetic fields. The KinKal fit implements
23 configurable simulated annealing, which allows for iterative pattern recognition and calibra-
24 tion refinement during the fit, and supports the integration of AI/ML pattern recognition tools
25 as part of the fit. Within the KinKal framework, Mu2e has implemented a specialized track
26 object, created fit constraint objects for both tracker hits and calorimeter clusters, and several
27 ANN-based hit classification algorithms, which are applied as part of the annealing schedule.
28 Associated calorimeter clusters are included as constraints on the fit time, allowing a more
29 precise interpretation of the tracker hit drift time as a position constraint. The ANN classi-
30 fiers remove residual background hits, refine the drift calibration, and help assign left-right
31 ambiguities to tracker hits.

32 Two configurations have been developed for the Mu2e KinKal fit, one optimized for use
33 in the online trigger, and the other for offline analysis. The "trigger configuration" uses a
34 minimal annealing schedule, a 1% precision magnetic field correction, does not use track
35 hit drift information, and does not add missing hits to the track. It provides a fit in roughly
36 5 msec/track processing time, with very high efficiency. The reconstructed momentum and
37 the intrinsic momentum resolution using the trigger fit configuration for simulated conver-
38 sion electron plus beam pileup events are shown in Fig. 14. Interpreting the full width
39 half maximum (FWHM) of the momentum distribution as an effective core resolution us-
40 ing $\sigma = \text{FWHM}/2.35$, the trigger fit achieves an absolute resolution of roughly 340 keV
41 down to the current trigger threshold of 80 MeV, which meets the Mu2e track trigger re-
42 quirements [49].

43 The "analysis configuration" uses a more gradual annealing schedule, a 10^{-4} precision
44 magnetic field correction, full track hit drift information, and adds missing hits. This configu-
45 ration requires roughly 50 msec/track, achieves essentially the same efficiency, and improved
46 resolution. The reconstructed momentum and the intrinsic momentum resolution using the
47 analysis fit configuration, sampled at the tracker entrance, for simulated conversion elec-

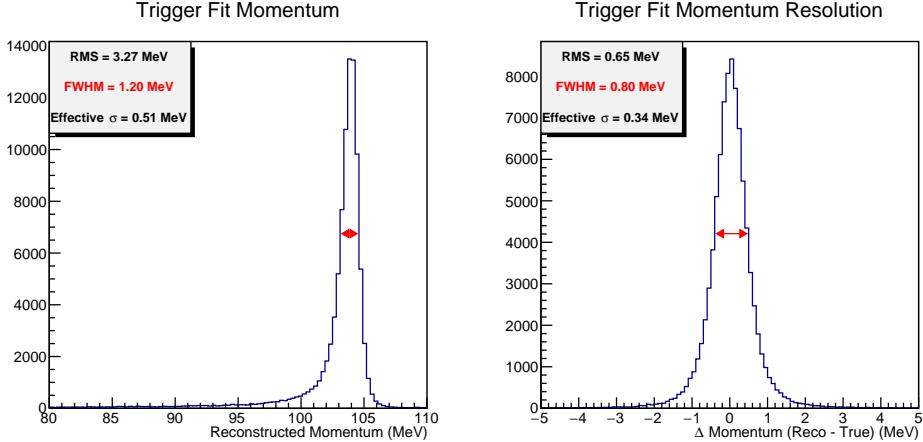


Figure 14. Track momentum (left) and momentum resolution (right) in simulated conversion events with beam pileup overlay, reconstructed using the trigger KinKal configuration.

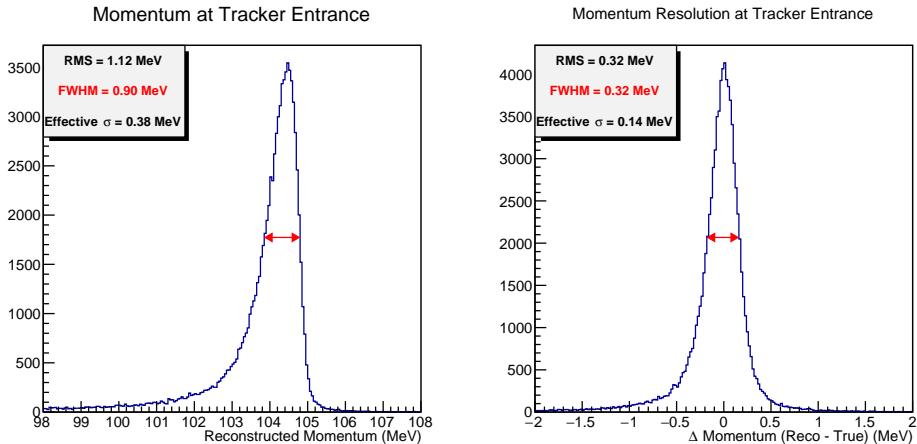


Figure 15. Track momentum (left) and momentum resolution (right) sampled at the entrance to the tracker in simulated conversion events with beam pileup overlay, reconstructed using the analysis KinKal configuration (see text for definition).

tron plus beam pileup events, are shown in Fig. 15. The fit achieves an intrinsic resolution $\sigma = \text{FWHM}/2.35 = 140 \text{ keV}$, exceeding the tracker requirement of $\sigma < 180 \text{ keV}$ [50].

As indicated by Fig. 15, the absolute momentum resolution of Mu2e tracks is dominated by energy loss straggling in the stopping target and IPA. Evaluating the track momentum after extrapolation to the stopping target can potentially improve the absolute momentum resolution by accounting for estimated energy loss track-by-track instead of in aggregate. Algorithms to extrapolate reconstructed KinKal fit tracks beyond the fit region are under development. The core fit engine support for extrapolation is fully implemented and extrapol-

1 lation in material-free regions has been demonstrated. The remaining task is the development
2 of a KinKal model of the stopping target and IPA material and its integration with the fit.

3 **8.3 Straight Track Reconstruction**

4 The straight track reconstruction is used for cosmic ray tracks in the field-off configuration.
5 As in the signal-track reconstruction, it begins by looking for clusters of filtered hits in time,
6 but without attempting to correct for the particle propagation time. The straight line track
7 seed also uses a simpler brute force algorithm. For each pair of straw hits, a straight line
8 between the reconstructed hit positions is drawn and tested for intersections with the other hit
9 straws, and the track with the highest number of successful intersections is chosen.

10 Two possible algorithms are used for the final reconstruction. First is a Kalman filter fit
11 based on KinKal. Here the particle momentum is an input parameter that is held fixed during
12 the fit. Similar to the signal fit, this fit can be configured to use a minimal annealing schedule
13 and no drift information. In addition to making it faster, this configuration also makes it more
14 robust against miscalibrations.

15 The second algorithm is a maximum likelihood fit that uses the ROOT Minuit2 minimizer.
16 The likelihood for each straw hit includes both drift and longitudinal information, with the
17 drift time likelihood for a given drift radius given by an exponentially modified Gaussian
18 distribution that models the effect of both the random variations of the drift times for a single
19 cluster and the exponential distribution of the effective drift distance given the discrete cluster
20 statistics. The track is parameterized as a perfect straight line so no scattering is included. As
21 a global fit, the results from this algorithm can easily be used with the Millepede-II alignment
22 algorithm.

23 **8.4 Calorimeter Reconstruction**

24 The data written by the calorimeter front-end electronics contains zero-suppressed waveforms
25 recorded by the photo-sensors at the back of each crystal. These waveforms could be pro-
26 duced by a single or several particles depositing energy in a crystal during a short amount of
27 time. In the latter case, the recorded waveform contains overlapping peaks, exhibiting mul-
28 tiple local maxima. The signal extraction procedure starts by fitting the waveform with an
29 iterative procedure to extract the time and amplitude of each peak. The fit model is revised by
30 adding or removing peaks until the chi-squared of the fit falls below a specified threshold (up
31 to nine peaks can be extracted in a single waveform). The leading edge of the first peak is also
32 refit to improve the timing resolution. The signal amplitudes are converted into energy de-
33 posits by using photosensor-specific calibration constants. The outputs of two photo-sensors
34 within a time window of 4 ns are then merged to form crystal hits.

35 Calorimeter clusters are finally formed by combining crystal hits. The most energetic hit
36 is taken as a cluster seed, and all simply connected hits compatible with the seed time are
37 added to the cluster content. These hits are then removed from the pool of available hits,
38 and the procedure is repeated until all hits have been assigned to a cluster. A fraction of
39 the particles interacting with the calorimeter produces one or more low-energy clusters near
40 a more energetic one. A procedure is applied to combine these split-off clusters with the
41 main cluster and improve the energy resolution. The energy, time, and other properties of the
42 clusters are finally determined.

43 **8.5 CRV Reconstruction**

44 The CRV front-end boards record waveforms of each channel of the CRV detector. This is
45 done in a non-zero-suppressed mode and a zero-suppressed mode, where the zero suppres-

1 sion threshold will be at around 5.5 photo-electrons. Non-zero-suppressed data is used for
2 calibration purposes, and to validate the zero-suppression algorithm.

3 The zero-suppressed waveform data is used to reconstruct candidates of cosmic rays go-
4 ing through the CRV. First, individual waveforms are searched for pulses (local maxima),
5 extracting the pulse time and pulse area. The pulse area is normalized by a channel-specific
6 calibration constant (extracted from the conditions database) to obtain an estimate of the
7 number of photo-electrons in this pulse. Based on the time, number of photo-electrons, and
8 location of recorded CRV pulses, groups of pulses are associated with coincidence clusters.
9 The current coincidence criterion requires pulses in 3 out of 4 CRV layers, with at least 10
10 photo-electrons, within 20 ns. Conversion electron candidates consistent in time and space
11 with originating from a CRV coincidence cluster will be rejected (vetoed) in analysis. The
12 CRV coincidence criteria and track consistency algorithms remain to be optimized for maxi-
13 mal acceptance while keeping the expected number of cosmic background tracks well below
14 one.

15 **8.6 STM Reconstruction**

16 The STM data written by the DAQ consists of prescaled unsuppressed and zero-suppressed
17 digitized waveforms as well the outputs of the online pulse processing algorithm, which con-
18 sist of the uncalibrated energy and time of each hit. The unsuppressed and zero-suppressed
19 waveforms are used to validate the online algorithms, and the outputs of the pulse-processing
20 algorithms are used for the analysis. The pulse-processing algorithms have been tested at
21 high rates using data collected in a test beam setup.

22 The outputs of the pulse-processing algorithms are calibrated in the Offline framework
23 with calibration constants stored in the conditions service. We plot the energy spectrum of
24 the calibrated hits and fit Gaussian functions to the peaks of interest in order to estimate the
25 number of stopped muons. This method achieves a 10% statistical uncertainty on the number
26 of stopped muons with ~ 10 minutes of data in simulation. These ~ 10 -minute counts and
27 associated uncertainties will be stored in a database. Event-by-event information such as the
28 number of tracker hits or the total energy deposit in the calorimeter, can be used to interpolate
29 between STM measurements to estimate the number of stopped muons per event.

30 **8.7 Extinction Monitor Reconstruction**

31 The reconstruction starts by grouping adjacent pixel hits from the same clock tick into clus-
32 ters. The coordinates of a cluster are computed as the center of gravity of its component
33 hits. The next step is pattern recognition. Straight line tracklets are formed from clusters in
34 the upstream and, independently, downstream sensor stacks. A tracklet is constructed from
35 3 clusters; all in the same clock tick but in different planes. The position cuts are loose, al-
36 lowing for a pixel-size misalignment error and a multiple scattering angle of up to 5 mrad in
37 a plane. Upstream and downstream tracklets are matched into candidate tracks if they are in
38 the same clock tick and if the angle difference in the non-bend projection for the tracklets is
39 within 0.005. A χ^2 fit is performed on candidate tracks to determine 5 parameters: the starting
40 position (x, y) and direction at the detector entrance, and the bend radius in the magnet.

1 **9 Detector Calibration and Alignment**

2 The calibration and alignment parameters needed by the reconstruction algorithms reside in
3 the conditions database described below, along with the metadata allowing for versioning and
4 defining the run/subrun range over which any set of calibration constants are valid. Several
5 concurrent sets of calibrations will cover different use cases, for example, an initial set of
6 values used during reconstruction for the trigger. Nominally, a given calibration set can be
7 extended with values covering a new set of runs, allowing calibrated reconstruction to be
8 performed on those runs. When calibration algorithms are updated or inputs are changed, a
9 new version of the calibration set can be defined as well, possibly requiring reprocessing of
10 relevant runs. The calibrations can also be overridden with text files, so iterating calibration
11 procedures can easily include rerunning reconstruction using the previous step's results.

12 A large part of the calibration and alignment will be performed in situ, using cosmic ray or
13 beam data collected during normal (including extracted position) Mu2e running conditions.
14 There are also several sources of calibration data that will need dedicated setup and runs,
15 including tracker charge injection pulsers, calorimeter laser system and DT source, Y-88 and
16 Eu-152 sources for the stopping target monitor, etc. Most of these will also require specialized
17 data processing. The various detector alignment procedures will also make use of metrology
18 measurements made during both construction and after installation.

19 **9.1 Tracker calibration**

20 The majority of the tracker calibrations will be performed iteratively using data from recon-
21 structed extracted or off-spill cosmic tracks. For extracted cosmics, the initial triggering and
22 event selection is based on a simple time clustering of hits in the tracker. Each cluster is then
23 reconstructed using both a KinKal Kalman filter track fit as well as a maximum likelihood-
24 based straight-line track fit. Both track fit results are processed with Mu2e Offline modules
25 to produce simplified output for calibration.

26 Information from the KinKal fit is used to determine tracker calibration parameters from
27 the TrkAna trees, including channel time offsets, time-to-distance relationships, and the TOT-
28 to-drift time. Mean residuals for each channel are used to calculate corrections to the cali-
29 brations. Collecting a few thousand hits per channel requires around one day of extracted
30 cosmic data.

31 The likelihood fit results are used to align the tracker modules with the Millepede-II algo-
32 rithm [51, 52]. Given a set of track fits Millepede-II performs a simultaneous least squares fit
33 to all track parameters and alignment parameters, including correlations. It can be run inter-
34 actively, taking only a few minutes to process millions of tracks. The Millepede fit can also
35 include results from metrology measurements as Gaussian constraints on any linear combi-
36 nation of the alignment parameters. Separate 6-DOF alignment parameters are included for
37 different levels of tracker modules, including each panel, plane, and the entire tracker. Redun-
38 dant degrees of freedom are removed by applying fixed constraints within Millepede. We plan
39 to eventually use KinKal fits of off-spill cosmic tracks for alignment as well, which should
40 provide better constraints on systematic shifts and rotations. A track refit based on General
41 Broken Lines [53] would allow us to reformulate the KinKal result and use Millepede again.

42 Both the alignment and calibration are performed iteratively. Each iteration requires re-
43 performing the hit and track reconstruction on the uncalibrated digi and producing a new set
44 of data, but we expect the number of iterations to be small, even for initial bootstrapping.
45 After convergence, the updated calibration and alignment parameters are uploaded to the
46 condition database, and the calibration set can be extended to include new values to allow for
47 pass-2 reconstruction.

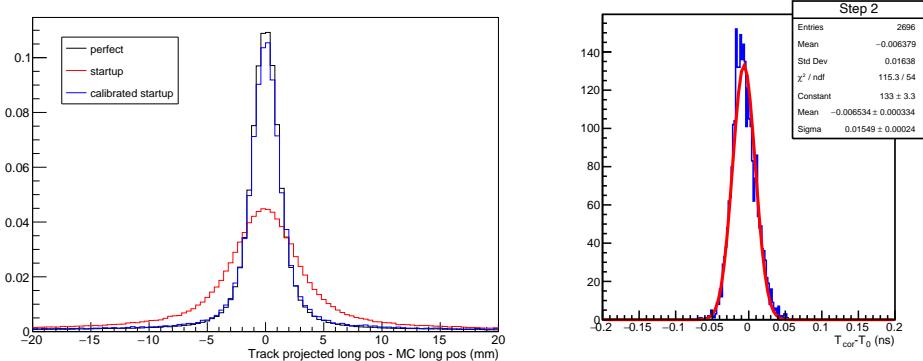


Figure 16. Left: Track fit longitudinal resolution in uncalibrated and calibrated simulated startup condition data as well as perfect simulated data. Right: Results of the calorimeter energy calibration procedure based on a simulated sample of cosmic rays equivalent to a data-taking period of 10 hours.

1 The ability to bootstrap the calibration and alignment using the first cosmic data was
 2 demonstrated using simulations. In the simulations, calibration and alignment parameters are
 3 set to randomly distributed values to model uncalibrated reconstruction. Additionally, param-
 4 eters for the detector configuration like thresholds and channel statuses are varied. Simula-
 5 tions of cosmic data in the extracted position were produced using distributions of calibration
 6 parameters and states that model the expected condition of the detector for the initial data
 7 taking. The KinKal track fit with a reduced set of annealing steps is used for the first iteration
 8 of the calibration, after which the likelihood fit and full KinKal fit can be used for further
 9 steps. The performance of the calibrated track reconstruction is shown in Figure 16.

10 **9.2 Calorimeter calibration**

11 The calorimeter makes use of three different calibration methods, based on the laser system,
 12 the source calibration systems, and cosmic ray events.

13 Cosmics calibration requires data samples in extracted position or off-spill during physics
 14 runs. The energy calibration is performed with minimum ionizing particles, identified as
 15 events with clusters from straight tracks with a minimum energy deposit. A $\sim 0.5\%$ spread is
 16 obtained from 10-hour simulated cosmic ray events in extracted position. The time calibration
 17 is performed in two steps. First, reconstructed hits produced by the laser are used to evaluate
 18 time offsets due to the electronics. The second step performs a least squares fit to clusters
 19 produced by minimum ionizing particles to calculate a correction to the time offsets for each
 20 channel. Simulation studies show that 15 ps time alignment is obtained. The energy and time
 21 calibration procedures have been validated using data collected by a large-scale calorimeter
 22 prototype containing 52 crystals. Results are consistent with MC expectations, demonstrating
 23 the robustness of the procedures.

24 The calorimeter has a radioactive source-based calibration system that allows for an ab-
 25 solute energy calibration of each crystal using a 6.13 MeV photon line. The distribution of
 26 reconstructed energy for each crystal is fit with 3 crystal ball functions, one for the main peak
 27 and two escape peaks, over a logistic background. This calibration has been tested using sim-
 28 ulations that include crystal non-uniformity, electronics noise, and different photo-statistics,
 29 demonstrating consistency to less than one percent with data from a ten-minute long run. The

1 laser system distributes green light to each calorimeter SiPM to perform a precise gain deter-
2 mination, with a dedicated run of a few minutes, and monitor the stability of the response by
3 continuously firing a few pulses in the off-spill data collection period. Source and laser runs
4 will be performed on a weekly basis.

5 The results from each calibration source are stored in "archive" tables in the condition
6 database (see Section 12), and combined to produce the final calibration constants stored in
7 the condition database.

8 The calorimeter alignment is determined during construction using a laser tracker survey
9 with a $O(100 \mu\text{m})$ accuracy on crystal position, enough for calorimeter reconstruction pur-
10 poses. The geometrical alignment of the calorimeter will be completed once on the Mu2e
11 detector train, through three aiming points placed on the calorimeter external structure. Cos-
12 mic ray events will then be used to determine the position and time alignment with respect to
13 the tracker and the CRV.

14 **9.3 CRV calibration**

15 The CRV uses a special set of non-zero-suppressed waveforms for calibration, collected and
16 digitized alongside normal data taking. The first stage processes the raw CRV digis and
17 histograms the waveform values to calculate per channel pedestal values. Once pedestals
18 have been calibrated, pulse reconstruction can be run, and an Offline module aggregates pulse
19 heights and areas for dark rate pulses. A standalone script then fits for the single PE value for
20 each channel.

21 This pulse calibration includes temperature corrections and requires data read out by the
22 slow controls, both in the calibration process and in the calibrated pulse reconstruction. Ini-
23 tially, these temperatures are stored in a separate slow control database and are then imported
24 into the reconstruction's conditions database at regular intervals (once every hour, depend-
25 ing on the temperature fluctuations in the Mu2e hall). Less than one second of non-zero-
26 suppressed data is required for the full CRV pulse calibration.

27 Reconstruction and calibration of dark pulses from the non-zero-suppressed waveforms
28 takes around 0.5 seconds per microbunch, or about 7 hours for the full pulse calibration
29 dataset. These procedures have been demonstrated on data from several test stands so far.

30 **9.4 STM calibration**

31 The location of the STM and its collimation system is initially performed by optical survey.
32 The relative alignment of the upstream and downstream structures will be then monitored by
33 a laser alignment system to ensure that there are no large changes in the alignment that need
34 to be corrected.

35 Acceptance and energy scale will be calibrated using Eu-152 and Y-88 sources. During
36 data taking, the energy scale is expected to be stable over a few hours. Beam data will be
37 used to monitor energy scale drift using various spectra lines (x-ray, gamma ray, 511 keV).
38 Calibration constants will be stored in the conditions database. The time delay between the
39 STM detectors and the other detectors can be measured using a scintillator paddle with a
40 known cable length.

10 Physics Analysis

2 Data analysis broadly refers to the ensemble of tools and techniques used to extract constraints on the physical properties of the muon. Data analysis tasks include, for example, measurements of muon branching fractions and capture rates, comparisons between data and Monte Carlo simulations, extraction of calibration and detector performance parameters, or search for new phenomena. For these tasks, the software framework used for large-scale data processing might not be the most appropriate solution. This difference may not necessarily arise from missing functionalities or technological limitations in the large-scale production framework but might reflect the preference of analysts to trade capabilities in favor of flexibility and simplicity. The usage of external tools, such as machine learning or fit algorithms, might also be easier within a custom framework. Similarly, reduced data analysis samples (aka ntuples) might be more convenient to analyze than data containing the full event information. These samples could be structured differently from the data recorded by the detector (e.g. columnar structure) to improve the processing performance as well. This section describes the analysis frameworks and reduced data samples used by the experiment, together with a generic reference analysis based on these tools. While the development of a blinding strategy is of high importance for the collaboration, it is not expected to have a significant impact on computing and won't be discussed any further in this document.

10.1 Analysis Frameworks and Interfaces

20 Offline analysis can be facilitated by using reduced data sets ("ntuples") consisting of a smaller set of physics variables deemed most useful for analyses. Information about low-level (hits) and high-level (tracks, clusters) products may need to be stored during commissioning and early data-taking periods. As the experiment matures, reduced ntuple scheme where only information from high-level data products is stored may become the default. Analysis groups will access these ntuples with either ROOT- or Python-based analysis scripts to create plots and perform statistical inferences to derive results.

27 To ensure operability on both FNAL and local resources, the ntuples data are stored as fundamental types instead of containing Mu2e-specific data products, removing the dependency on the full Mu2e software environment. Using the same assumptions and mock data samples as for the reconstruction output, the size of a ntuple dataset is estimated to be of the order of 100 GB (1 TB) if hit-level information is discarded (stored), small enough to be stored on persistent dCache (to avoid pre-staging the files from tape) or even locally. The simpler structure also translates into a shallower learning curve for new users since it can be easily accessed with ROOT and/or Python, the latter of which in particular is a common language used by early-career researchers.

36 Mu2e currently has two ntuple frameworks: TrkAna and Stntuple. Both output ROOT-based structures with data organized into event rows and containing reconstructed information extracted from Mu2e data products associated with the tracker, calorimeter, and CRV, as well as Monte Carlo truth information. TrkAna provides a simple ROOT TTree structure that can be accessed with ROOT or Python. Stntuple provides a separate lightweight interactive ntuple-based analysis framework based on that used by the CDF experiment at Fermilab. The Stntuple framework was ported to Mu2e and has evolved to meet the experiment's needs over several years. It supports multiple job configurations, interfaces with the data handling system, and has a built-in 2D event display. Stntuple was used for the run 1 sensitivity estimate [16] and TrkAna is used for current mock data analysis efforts and to study cosmic ray alignment in the extracted position.

1 The framework is expected to evolve significantly in the near future with the definition of
2 a single ntuple format satisfying most analysis needs and the development of interfaces with
3 modern analysis tools. This ntuple will necessarily have a "jagged" structure. For example,
4 there will be different numbers and types (e.g. electron, muon) of reconstructed tracks in
5 an event, and each track will include fit results from the Kalman fitter at different segments
6 along the track (e.g. entrance of the tracker). The framework will provide simple interfaces
7 to analyzers to manage this dimensional complexity and reduce the learning curve.

8 Analyzers will use either ROOT or Python to perform their analyses. ROOT-based analy-
9 ses will use C/C++ macros, and Python-based analyses will use Python macros and/or Python
10 notebooks. The ntuple is stored as a ROOT TTree so ROOT-based analyses will either loop
11 through the TTree directly or use RDataFrames. RDataFrames have the advantage of per-
12 forming "lazy" histogramming (i.e. define all cuts and histograms before looping through
13 the data) and offering a transparent use of multi-threading capabilities. However, this would
14 require code development to write RNtuples instead of TTrees, and we plan to evaluate the
15 cost-benefit of this work during the long shutdown after run 1 . Python-based analyses can
16 access the data using ROOT via PyROOT or via the Python packages uproot and awkward
17 array. Python-based analyzers can also use a wide range of available Python packages (e.g.
18 numpy, scipy, hist) to perform their analyses. A common analysis environment is required
19 to aid interoperability between groups and to ensure reproducible results. For ROOT-based
20 analyses, a standard ROOT installation is maintained in Mu2e Offline, while a virtual en-
21 vironment will be provided for Python that can be easily recreated on local machines with
22 either the venv or conda package managers. Tutorials and documentation of both Python-
23 and ROOT C++-style analyses will be maintained throughout the experiment's lifetime.

24 Machine learning (ML) algorithms are expected to play an important role in analysis-
25 related tasks. Track quality and particle identification neural networks have already been
26 trained in Python (with Tensorflow) and then evaluated in Mu2e Offline during the ntuple stage.
27 For these, the ROOT TMVA::SOFIE interface is used to produce the C++ code to
28 perform the calculation. New algorithms developed within the analysis groups will either be
29 converted using TMVA::SOFIE and evaluated in the ntuple stage, or a common interface will
30 be developed so that these tools are accessible to all analysis groups. ML algorithm training
31 and other computing-intensive tasks (e.g. statistical inference with a large number of nuisance
32 parameters) will leverage hardware accelerators and high-performance computing architec-
33 tures to improve performance. The Elastic Analysis Facility (EAF) at FNAL could also be
34 used to scale up data analysis: analysts can prototype their workflow on a subset of the data
35 before seamlessly processing the full dataset. In any case, the computing needs for analysis
36 are expected to be significantly lower than those needed for simulation and reconstruction.

37 **10.2 Reference Analysis**

38 The Reference Analysis is developed to quantify the impact of software and algorithmic
39 changes on the physics performance of the experiment; to prototype analysis tool; and to
40 provide a cross-check to analysis groups. It takes input from the standard analysis framework
41 in the form of reconstructed Mock Data developed using the output of large-scale ensemble
42 production (see Section 7.9). In ensemnbles, conversion signals are combined with DIO tail
43 backgrounds, cosmic backgrounds, radiative pion and muon capture backgrounds, and pile-
44 up. The pile-up model contains contributions from neutral particles, beam electrons, and
45 other muon processes such as decay in flight. The simulated data are then passed through the
46 digitization and reconstruction framework to create samples similar to the data that will be
47 collected by the DAQ system.

1 Datasets equivalent to a year of data taking are created for two different $R_{\mu e}$, plus a no-
2 signal scenario. A corresponding set of samples for the $\mu^- N \rightarrow e^+ N'$ (including leading log
3 corrections) is also produced. Digitization and reconstruction are simulated under two sets of
4 calibration conditions: perfect and best. The "perfect" scenario describes a situation where
5 the detector and digitization parameters behave in an ideal way, while the "best" scenario
6 assumes more realistic conditions. As calibration algorithms evolve, more diverse sets of
7 conditions could be simulated.

8 The current Reference Analysis contains signal extraction derived from two complemen-
9 tary and independent strategies: a simple counting experiment and an unbinned maximum
10 likelihood fit. A few sources of systematic uncertainties will be added as nuisance parame-
11 ters into the fit. Work to integrate the reference analysis into the validation workflow to track
12 the evolution of reconstruction and analysis codes has begun.

1 11 Visualization and Event Display

2 Visualization of raw and reconstructed data provides essential information to assist experi-
3 menters in multiple tasks:

- 4 • **Algorithm development** - tracks or clusters from alternative algorithms can be visualized
5 within the geometries and alongside Monte Carlo truth information, allowing easy com-
6 parisons and aiding debugging
- 7 • **Simulation studies** - Monte Carlo information can be visualized, allowing studies of par-
8 ticle interactions in instrumented or uninstrumented detector regions
- 9 • **Detector commissioning** - during initial commissioning sub-systems will be tested inde-
10 pendently, sometimes in incomplete configurations. Visualizing of through-going cosmic
11 rays can provide useful information to detector teams
- 12 • **Physics analysis** - visualization of data within the detector systems can provide invaluable
13 information about the nature of an event. In the event of a signal-like track, the collab-
14 oration we will want to explore in great detail the properties of this event, and the event
15 display will be a key tool.

16 Three event displays are currently available. The first is a custom ROOT-based display
17 showing geometries and data products as TGeo objects. The second is TEveMu2e, exploit-
18 ing ROOT's existing Event Visualization (EVE) framework, TEve. This framework allows
19 visualization of all detectors via the ROOT OpenGL interface. The full Mu2e geometry
20 can be imported into TEve via a simple GDML file. In addition to the main 3D display,
21 users can browser-specific 2D views of each detector system via tabs in the browser. A third
22 display, REveMu2e, is based on ROOT's REve class [54]. REve is an evolution of TEve
23 for the ROOT-7 era, designed upon information collected by the HEP Software Foundation
24 Community White Paper on Visualization [55]. It is developed by members of the CMS col-
25 laboration and will be maintained throughout the coming decade and beyond. REve utilizes
26 the OpenUi5 (<https://openui5.org/>) framework to build a customizable web-based GUI, pro-
27 viding more modern infrastructure. A REve-based display can be set up remotely via a web
28 browser and an SSH tunnel, allowing fast and secure operation of the viewer. REveMu2e is
29 the default event viewer within the Mu2e experiment, and it has been integrated into the DAQ
30 system to serve as an online event monitor during data taking.

31 11.1 REveMu2e

32 REveMu2e displays the entire Mu2e geometry (read from a gdml file) and all reconstructed
33 and truth-level data products in the tracker, calorimeter, and CRV. The detector display in the
34 "extracted position" during the commissioning phase is also supported. The user can navigate
35 to a chosen event (this is directional as backward navigation is prohibited in art) and rotate,
36 zoom, and pan the view.

37 The code is structured in sets of "scenes" containing different types of information. The
38 geometry scenes include individual detector geometry objects, such as the tracker or CRV
39 scenes, allowing users to add or remove sub-systems from the display interactively. The
40 event "scene" contains event-by-event information on physical objects: REveLines for tra-
41 jectories and REvePointSets for hits or clusters. Within this scene, objects of interest can be
42 highlighted. For example, the color of the crystal and its length is proportional to the energy
43 deposited in that crystal. The display can also show GEANT4 trajectories, allowing offline
44 analysts to compare their reconstruction with Monte Carlo truth information. Information
45 associated with a given data product can be obtained by simply scrolling over the product.

- Interactive labels will then pop up with various data, such as the algorithm used to reconstruct the object, its position, time, energy, etc. The GUI also provides options to print further details to the terminal. In addition, a set of 2d views is provided by default: an XY and YZ view of the tracker with XY views of each calorimeter disk. Additional views can easily be added if necessary. Illustrations of the capabilities of REveMu2e are shown in Fig. 17 and 18.

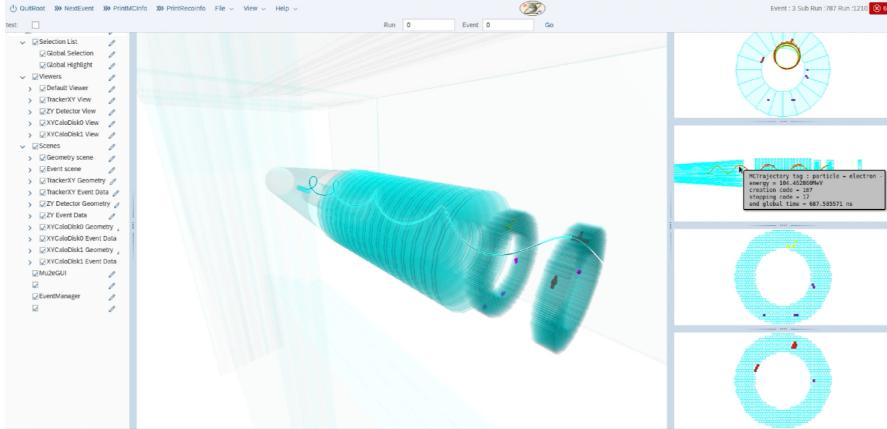


Figure 17. User Experience: An example REveMu2e displaying a conversion entering the detector region and producing a helical track in the tracker and calorimeter. The MC truth trajectory is highlighted and a label is visible.

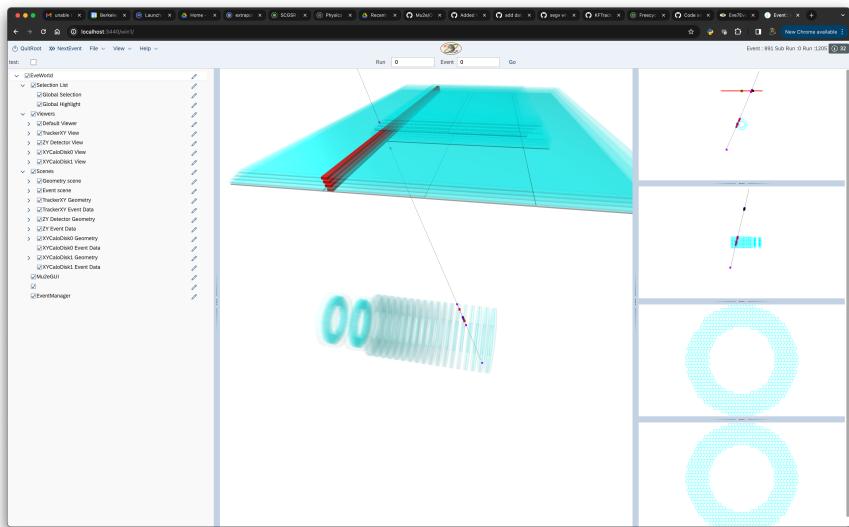


Figure 18. Extracted Cosmic View: An example REve displays of the showing a cosmic muon entering the detector region in extracted (field off) configuration

12 Databases

2 The Mu2e database structure comprises several databases containing the metadata required
3 to collect, store, process, and analyze data. An overview of information flow through Mu2e
4 databases is shown in Figure 19. The subset of metadata needed to understand the data
5 collected by the Mu2e detector is often referred to as condition data (e.g., detector alignment
6 parameters or calibration constants). These data are indexed by (fraction of) runs, grouped
7 into intervals of validity. Conditions data sampled at a lower frequency will be interpolated to
8 match the physics requirements of the experiment. While conditions metadata will generally
9 be stored in appropriate databases, they could be inserted in the raw detector data stream in
10 some instances. APIs providing access to condition metadata independent of the underlying
11 technical details will be provided to users.

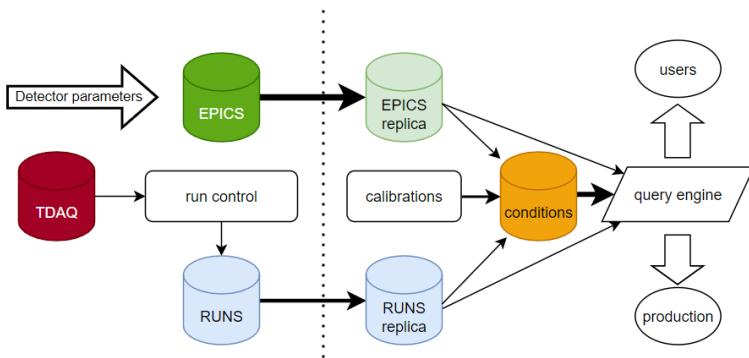


Figure 19. An overview of information flow through Mu2e databases.

12 Reliable access to metadata over the full lifetime of the experiment is a crucial require-
13 ment of the database system. To mitigate the risk that the underlying technology becomes
14 obsolete during this period, Mu2e is following the general FNAL strategy of adopting open-
15 source and widely supported products. The Mu2e DAQ system is based on MongoDB tech-
16 nology, while all other databases are using the PostgreSQL database management system.
17 These databases are supported by the database group of the FNAL IT division, providing the
18 software, the platform, and the servers (except the DAQ MongoDB database, hosted on the
19 DAQ server to ensure continued access in case of network outage). All offline database ac-
20 counts are authenticated by kerberos principles and are created by the database support group
21 via a service desk ticket. The support group also provides a streaming service for copying
22 from the online to the offline instance. The Mu2e workflow assumes 8/5 support for offline
23 database with best effort outside standard hours.

24 All offline database content is written via SQL commands generated by Mu2e custom
25 tools in c++ and python. Database queries are performed through the Query Engine, a lab-
26 supported product providing an HTTP interface for all simple, high-rate queries. Two types
27 of queries are available: reads from the web server cache (the cache refreshed with a pro-
28 grammable lifetime), or directly from the database via the web server. This system can be
29 scaled to provide conditions content to jobs distributed across large numbers of grid-based or
30 high-performance computing (HPC) systems. The service also provides valuable real-time
31 monitoring of system response times and timelines. In addition, (partial) database content
32 can be dumped into a file on /cvmfs and used for subsequent reconstruction or simulation

1 tasks. This option could be useful in cases where the network bandwidth becomes limited,
2 for example running a very large number of simulation jobs at a high-performance computing
3 center.

4 Database development is fully integrated into the Mu2e code development workflow and
5 performed in collaboration with sub-system experts. A core principle is to maintain an au-
6 tomated robust audit trail of the processing history of each file, from TDAQ to final analy-
7 sis datasets. This includes recording version information for code, databases and workflow
8 configuration, and parent-child information among files. Mu2e has already recorded this in-
9 formation during the various simulation campaigns and we are investigating improved imple-
10 mentations. Among the desired features is the ability to specify that a file must be reprocessed
11 using the same database versions that were used in the trigger.

12 The list of Mu2e databases is presented in Table 3, and each instance is described in more
13 detail below.

label	schema	description
DAQ	lab/Mu2e	DAQ configuration
EPICS	EPICS	detector metrics
run conditions	Mu2e	run metadata
offline EPICS	Mu2e	streamed from online
offline run conditions	Mu2e	streamed from online
conditions	Mu2e	calibrations
DQM	Mu2e	offline data quality
good run	Mu2e	good run list
luminosity	Mu2e	runs luminosity
SAM	lab	file metadata
MetaCat	lab	file metadata
Rucio	lab	file locations

Table 3. Listing of Mu2e operations databases. All are installed and managed by lab professionals.
All are postgres except DAQ, which is MongoDB.

14 **12.1 Run Conditions database**

15 The DAQ database is managed by the online operations group. At the beginning of each
16 run, the DAQ system copies all information used to configure the detector and DAQ from the
17 MongoDB DAQ database into the online run conditions database. Each run is divided into
18 subruns, and the system records the precise times of the subruns in both wall clock time and
19 accelerator ticks. At the end of the run, summary information (e.g. total number of processed
20 events or run time) is added to the record.

21 The online content is periodically streamed to an offline database instance. As a new run
22 appears, offline data reconstruction is triggered and the job configuration is retrieved from the
23 database. A web page with interactive search for data discovery also provides information
24 about recent runs. Finally, tools are developed to extract the trigger configuration so that it
25 can be reproduced for simulation or performance studies.

26 **12.2 EPICS database**

27 EPICS is a system designed to monitor, display, and alarm arbitrary quantities, such as volt-
28 ages, status, and environmental conditions. This stream of monitoring data is archived in

1 an online postgres database using standard EPICS tools and then streamed to an offline in-
2 stance. Users can access content through a simple custom tool (epicsTool), available both at
3 the command line and in c++.

4 Environmental conditions required for detector response reconstruction is automatically
5 extracted from the EPICS database offline instance and re-packaged in the conditions system
6 to ensure reproducibility.

7 **12.3 Conditions database**

8 The conditions database tracks the detector conditions (e.g., pedestals, gains, bad channels,...)
9 and their run dependence using a custom schema. The user specifies a purpose (such as
10 "Pass 1", "Sim2020", or "CaloCalibration") together with a version number to define which
11 "conditions set" the reconstruction job will access. The version number has three fields: the
12 major version, used by experts to alert the user of an important change in conditions philos-
13 ophy; the minor number, indicating changes to content; and the final number, specifying the
14 extension listing the intervals of validity. Reconstruction jobs with the same major and minor
15 numbers will produce identical results, as a design requirement. The schema implements a
16 closed interval of validity (with an option to relax if needed). When a repair is needed, or
17 new types of tables need to be added, a new minor version is created.

18 Each detector sub-system defines the content of the tables added to the schema. In ad-
19 dition, "archive" tables may be created to store intermediate results which are not used in
20 reconstruction, but should still be saved. The system also provides "ad-hoc" tables for sec-
21 onary needs like record-keeping. A development database is available to verify the SQL
22 commands and compatibility with code before creating the same table in the production en-
23 vironment. The content and intervals of validity provided by each sub-system are grouped
24 into a condition set.

25 A user interface dubbed "Proditions" provides access to the database in the reconstruction
26 code. This layer can perform intermediate calculations based on the database contents and re-
27 organize the data into different data structures or subsets of table data. The run-dependence of
28 this content is automatically tracked based on the database table run dependence. Proditions
29 has also an onboard cache to prevent thrashing.

30 Other implementation features include a text table option for overriding database content
31 for testing purposes. Should the database access be unavailable, the text option can provide
32 the entire conditions set content. A programmable onboard cache saves tables so switching
33 back and forth between tables does not generate new fetches. All external operations are
34 timed to evaluate the performance. Multiple layers of verbosity are provided to help debug-
35 ging. The code is designed to be thread-safe as well.

36 All tools needed for database upload, queries, and repair have been developed. The con-
37 ditions database system has already been extensively tested, and the measured performance
38 will greatly exceed requirements for Pass 1, the most time-sensitive operation.

39 **12.4 DQM database**

40 Data quality information may be produced at any stage of processing. A set of analysis mod-
41 ules histogram critical quantities and the histogram files are preserved. A series of tags label
42 the files according to source, version, time, and run. A process then converts the histograms,
43 and other information, into a series of metrics which are inserted in the DQM database. These
44 metrics can be displayed as a function of run number or time or extracted as CSV tables for
45 further analysis. A system of limits provides alarms when metrics are out of tolerance.

1 **12.5 Good-run database**

2 The definition of "good runs" will depend on the context and might involve input from the
3 different detector sub-systems and experts. The good-run database collects information rela-
4 tive to run quality (typically an int) for a given interval of validity. Users can retrieve and edit
5 this information to create a custom selection, which is then persistent in the database. The
6 data can be accessed either by dedicated tools or within the reconstruction software.

7 **12.6 Luminosity database**

8 The number of stopped muons must be known to properly normalize the experimental results.
9 To this end, it may be useful to track the number of protons on the primary target and other
10 experimental quantities (e.g. the number of protons in the tracker), possibly as frequently as
11 every micro-bunch. This large amount of data would be preserved in a database and accessed
12 using custom tools or within the reconstruction software.

13 **12.7 Data-handling databases**

14 The data-handling database is fully designed and managed by the lab and contains standard
15 schema. All writes are by standard tools with standard authentication systems. All reads are
16 via load-balanced http services designed and maintained by the lab. The SAM database is a
17 legacy system and will be deprecated before full operations.

1 13 Software Management

2 The Mu2e code base is managed using public Git repositories, with the official copy stored
3 on GitHub under the Mu2e Organization. Code development adheres to industry standards
4 and best practices to ensure high-quality software is continually produced. The development
5 workflow follows a continuous integration (CI) model in which changes are frequently in-
6 tegrated into the source code to ensure that the code base remains in a functional state. All
7 changes are done via pull requests reviewed by experts and validated with a series of quality
8 checks before integration. High-statistics validation tests are also performed to assess the
9 impact of code changes on the reconstruction algorithms and overall physics performance of
10 the experiment. These practices ensure that production quality code can be quickly released
11 at all times.

12 13.1 Offline repositories

13 The major repositories relevant to offline computing are:

- 14 1. Offline: the main Mu2e repository, containing the core infrastructure, the core algo-
15 rithms for simulation and reconstruction, the data product definitions, and utility rou-
16 tines. It also contains the standard configurations for all *art* modules and services.
- 17 2. Production: contains the definitions for complete *art* jobs for non-trigger purposes,
18 derived from the component definitions found in Offline. It also includes the configu-
19 ration information for POMS campaigns and the corresponding scripts.
- 20 3. mu2e_trig_config: contains the definitions for complete *art* jobs that will run in the
21 trigger, derived from the component definitions found in Offline. Once the experiment
22 starts taking data, the json file content of the repository will be superseded by tables
23 created by the DAQ run management software.
- 24 4. TrkAna and Stntuple: contains the code to produce the analysis format data sets
- 25 5. Mu2eREve: contains the code for the main event display
- 26 6. Tutorial: source code and documentation for the tutorial examples
- 27 7. TrackerAlignment, CaloCalibration: calibration and alignment algorithms for the
28 tracker and calorimeter
- 29 8. CRVteststand: calibration and alignment algorithms for the CRV test stand

30 Additional repositories support TDAQ, elements of the Mu2e Construction Project, data
31 handling, and the systems discussed below. The core repositories comprise Offline, produc-
32 tion, and mu2e_trig_config, and the remaining ones are built on top of these. Repositories
33 are tagged using semantic versioning.

34 13.2 Code development

35 The Mu2e development and run-time environments are managed by a thin bookkeeping layer
36 called Mu2e Software Environment (muse), and the code is built using scons [56]. Muse
37 supports adding pre-built binary distributions of the core repositories to the development en-
38 vironment to reduce the time needed to build development code. It also includes a feature

1 to produce a tarball of the current development area usable by grid jobs. Software for dis-
2 tribution is built using projects on the CSAID-supplied Jenkins platform to produce tarballs
3 that are unwound and published on /cvmfs. Production grid jobs always run using software
4 distributions published on /cvmfs, and interactive user jobs that do not require code develop-
5 ment also run this way. External products, such as compilers, Geant, root, and the *art* suite
6 are supplied as pre-built binaries supplied by the CSAID Scisoft team. These products are
7 delivered as spack packages on the /cvmfs file system.

8 Mu2e uses a Continuous Integration (CI) system in which selected activities on GitHub
9 trigger a GitHub webhook. The webhook sends a message to the CSAID Jenkins platform,
10 which is forwarded to the Mu2e Jenkins projects after authentication. These projects build the
11 requested code and run a suite of 16 tests. Most tests verify that an *art* job will successfully
12 run a few events; together these have coverage for most of the simulation and reconstruction
13 codes. The tests also include validating the Geant geometry and checking for compliance
14 with coding standards and practices using clangtidy. All changes to the offline repositories
15 are done by submitting a pull request (PR), which also triggers the CI activities just described.
16 The Mu2e computing management organizes a review of each PR and merges the code once
17 all reviewers have given their approval. When needed, the CI can be re-triggered by the author
18 of the PR or repository administrators. We have also implemented commit hooks to check for
19 common formatting errors (line-end whitespace, excessive blank lines, blank lines at EOF,
20 etc.), blocking the commit if they fail. We run clang-tidy on every PR, and have started to
21 require users to address the errors, and examine the warnings, before the PR is merged.

22 Mu2e runs a validation suite every night, checking for successful builds of all of the
23 major repositories. Similarly to the CI, the validation checks that several jobs running a few
24 events each complete successfully. The validation system also launches 9 grid jobs running
25 a total of 140 grid processes, each of about 30 minutes duration, covering all of the major
26 Mu2e workflows. The validation system collates the outputs and compares them to the most
27 recent reference output using Offline DQM tools described in Section 15. The validation
28 system has automated recovery procedures to rerun grid jobs that suffered transient failures
29 of infrastructure. The code driving the nightly validation is maintained in repositories in the
30 Mu2e GitHub organization.

31 The code is extensively documented on the Mu2e wiki pages, including tutorials, instruc-
32 tions to build and run the software, advice for developers,... Version-dependent documenta-
33 tion is maintained on GitHub. Additional details are given in Section 17.

14 Data management and Preservation

2 Mu2e will follow the procedures for the management of experimental data outlined in the
3 Mu2e Data Management Plan[57]. This plan is reviewed every two years.

4 Raw data will be archived at Fermilab for several years after the end of the Mu2e data-
5 taking phase. Two copies will be stored in physically separate locations. Over the lifetime of
6 the experiment, many generations of derived datasets will be produced and Mu2e is develop-
7 ing a plan for life-cycle dataset management. Reduced datasets for analysis will have a much
8 smaller size and those used for publication will also be retained for several years after the
9 end of data taking. For all other datasets, generation N-2 will be deleted once generation N
10 becomes inactive to free media space. This policy applies to both experimental and simulated
11 data.

12 The Mu2e software is stored on GitHub and most repositories are publicly readable. As
13 discussed in the Mu2e Software Policy [58], repositories that hold proprietary analysis code
14 are private until publication of the relevant paper. Once published, the repository is made
15 publicly readable.

16 Prior to the data-taking phase, Mu2e will consult with CSAID to determine which support
17 is best suited for making data publicly available, including support to preserve data at the end
18 of the experiment.

15 Offline Data Quality Monitoring

2 Offline data quality monitoring (DQM) provides rapid feedback to the data-taking process.
3 Offline computing should be aware of incoming data via the database stream within min-
utes of the start of a new run, triggering the first reconstruction pass (Pass 1, see section 3).
5 DQM code will run immediately afterward with access to all data streams to inspect tracks,
6 calorimeter clusters, and all components of the system. The results should appear within a
7 few hours of the start of a run, and experiment operations will have immediate access. DQM
8 could be run at any other stage, including on raw data before Pass 1, during the calibration
9 process, or on the final dataset production.

10 The first component of the system is histogramming. For simplicity and robustness, the
11 system starts with histogramming the content of reconstruction output, but anything that can
12 be computed in an art job can be reported. The output is a histogram file, labeled by the
13 process which produced it, the run and subrun involved, and the degree of concatenation.
14 Some quantities will require very little data to be useful, while some might need a full run or
15 even many. An arbitrary level of concatenation is allowed. These histogram files are saved to
16 disk for several weeks, then archived to tape.

17 The second step in the DQM process is to derive metrics from the histogram files, such
18 as the mean or RMS of a histogram, the number of hits on a track, or normalized quantities
19 (e.g. the number of CRV hits per time). These metrics are then labeled to keep track of the
20 provenance, and inserted in the DQM database (see section 12). The DQM core system (a
21 nominal set of histograms and metrics) has been running for over a year as part of nightly
22 validation.

23 The final part of DQM processing is displaying the results. Mu2e supports two display
24 styles. The first is an overlay of histograms, for example, comparing the current run to the
25 previous one or to a standard set of plots. Several tests of comparison are provided, including
26 a systematic allowance. The second display style shows the metrics as a timeline, allowing
27 for some selection on the display (see Figure 20). The system uses the plofty package to
28 allow interactive use.

29 Several standard sets of plots will be provided to the shift crew and experts, together with
30 tolerances for histogram comparisons or acceptable ranges for the metrics. An alarm will be
31 sent to experts when automatic comparisons are out of tolerance.

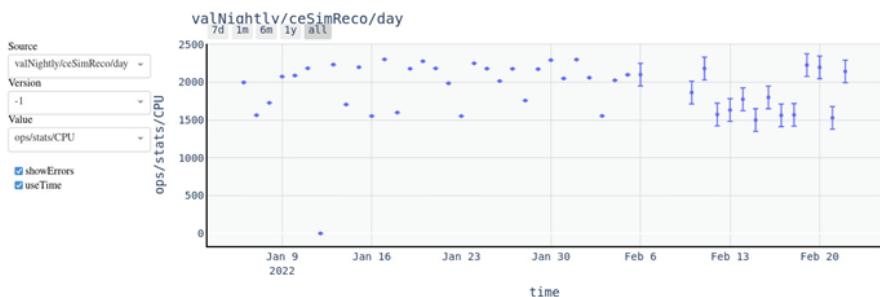


Figure 20. An example display of a metric timeline

16 Communication Tools

2 Lines of communication need to flow among developers, among analyzers, and between de-
3 velopers and analyzers. Mu2e uses the following tools to communicate along all lines: email
4 lists, Slack, doc-db, and GitHub.

5 Slack is the major communication mode used for quick-response and day-to-day commu-
6 nication such as general announcements, questions, and debugging of active issues. It is also
7 to communicate with users to help debug. Mu2e has a premium plan such that all conversa-
8 tions are backed up and stored and Slack can be queried to allow users to look up previous
9 conversations in all public channels. It is also used by developers to announce important
10 hardware/software updates to analyzers that will affect their workflows. Communication for
11 Analysis Reviews and internal Analysis Group discussions was recently migrated to dedi-
12 cated slack channels, replacing the previous use of hypernews for this purpose.

13 Email lists are used mostly to announce meetings and important events of broad general
14 interest to the entire collaboration. The most important email lists are linked directly to Slack
15 channels to allow maximum reach to the intended audience. Any major software updates or
16 infrastructure changes are also announced to relevant listservs.

17 Doc-db is used to store meeting agendas, presentations, and reports. Private groups can
18 be created within doc-db, such that only members of a given Analysis Group can see these
19 documents. This allows for independent analyses to be conducted privately while retaining
20 all relevant documents in a common repository.

21 Github is used to communicate between software developers. Analyzers can post tech-
22 nical issues and feature requests. Developers review pull requests. Issues can be searched
23 to find open projects in search of new personnel. A system for labeling issues consistently
24 across the Mu2e repositories is under development.

17 Documentation and training

2 The Mu2e collaborations includes more than 200 members, with new collaborators regularly
3 joining the experiment. To ensure the long term success of the experiment, a harmonized
4 documentation effort, coupled with consistent training for both newcomers and existing users,
5 is paramount. The documentation related to the computing aspects is accessible on a variety
6 of platforms, each with specific goals and access policies.

7 17.1 Documentation

8 The existing Mu2e wiki provides the starting point for the computing documentation. The
9 wiki pages are divided into a public side containing general information about Mu2e com-
10 puting, and a private side storing sensitive information. Users need a FNAL Single Sign-On
11 service domain account to access the private side. The public Mu2e wiki is divided into three
12 section: the Mu2e experiment, Mu2e computing, and practical information. The computing
13 section includes six topics:

- 14** • Computing tutorial : provides an overview of the computing, guided tutorials, and intro-
15 ductory references
- 16** • Accounts, Authentication, and Infrastructure : describes the various types of account and
17 authentication procedures and the general computing infrastructure
- 18** • Code : contains information about the Offline code base, the development tools, the coding
19 standards, the fcl language, the Mu2e data products, and various utilities
- 20** • Grids, Workflows, and Data Handling; reviews procedures to prepare, submit and manage
21 large jobs, upload files, dCache, tape,....
- 22** • Getting help : lists the various resources to get help and additional information
- 23** • Management : includes information about the FNAL and Mu2e computing management
24 structure and related tasks

25 The wiki is maintained by the collaboration at large, and each collaborator is requested to
26 document their work. Mu2e plans to conduct an annual "wiki review" to systematically
27 identify outdated material, add missing documentation, and review actual content.

28 The Mu2e software repositories are hosted on the GitHub platform, which offers a ticket-
29 ing system to facilitate debugging, revisions and updates from the community of users. This
30 features has been extensively used by the collaboration, and continue to be the main system
31 to track code development tasks. Version-dependent documentation is also hosted on GitHub.

32 17.2 Training

33 Mu2e training is primarily aimed at new users, particularly younger early career users (such
34 as new graduate students and post-docs). The primary training for new Mu2e members is
35 done twice a year during dedicated sessions, usually a day workshop following a collabora-
36 tion meeting, to ensure both new collaborators and experts are present in person. A zoom
37 component is also offered for those who cannot travel. Training material is maintained
38 throughout the year to provide "at home" learning as well. The training follows a hands-
39 on example-based task format, starting with basic examples and moving gradually towards
40 more advanced material.

41 The tutorial material is divided into three sections: General Offline Mu2e computing,
42 TrkAna/analysis, and event visualization. Within the GitHub repositories of the TrkAna and
43 REveMu2e packages are a set of tutorials with gradually increasing difficulties. TrkAna,

1 tutorials exist for both Python and ROOT C++-based analyses. The basic tutorial shows
2 users how to import the information to their analysis platform, manipulate it, add selection
3 cuts, and make basic histograms. The more advanced tutorials provide users with a real
4 analysis problem and demonstrate the method to perform maximum likelihood fits to a Mock
5 Data sample to provide point estimates of the signal and background yields. The REveMu2e
6 tutorial starts by showing the user how to visualize tracks, hits, and clusters in the three main
7 detectors and demonstrates the interactive features and event navigation. Training for more
8 general Mu2e offline software development is also offered to users wishing to contribute to the
9 source code. This tutorial includes setting up and building the Mu2e software environment
10 and running a few simple tasks to modify the C++ source code. As the software evolves
11 rapidly, the training material is updated frequently to ensure it remains always functional.

1 References

- [1] L. Bartoszek et al. (Mu2e Collaboration) (2014), [1501.05241](#) 1.1, 8
- [2] S. Petcov, Sov. J. Nucl. Phys. **25**, 340 (1976), [Erratum: Sov. J. Nucl. Phys. 25, 698 (1977), Erratum: Yad. Fiz. 25, 1336 (1977)] 1.1
- [3] W. Marciano, A. Sanda, Phys. Lett. B **67**, 303 (1977)
- [4] B.W. Lee, S. Pakvasa, R.E. Shrock, H. Sugawara, Phys. Rev. Lett. **38**, 937 (1977), [Erratum: Phys. Rev. Lett. 38, 1230 (1977)]
- [5] B.W. Lee, R.E. Shrock, Phys. Rev. D **16**, 1444 (1977)
- [6] P. Blackstone, M. Fael, E. Passemar, Eur. Phys. J. C **80**, 506 (2020), [1912.09862](#) 1.1
- [7] L. Calibbi, G. Signorelli, Riv. Nuovo Cim. **41**, 71 (2018), [1709.00294](#) 1.1
- [8] M. Fukugita, T. Yanagida, Phys. Lett. B **174**, 45 (1986) 1.1
- [9] S. Davidson, E. Nardi, Y. Nir, Phys. Rept. **466**, 105 (2008), [0802.2962](#) 1.1
- [10] W.H. Bertl et al. (SINDRUM II), Eur. Phys. J. C **47**, 337 (2006) 1.1
- [11] R.M. Dzhilkibaev, V.M. Lobashev, Sov. J. Nucl. Phys. **49**, 384 (1989) 1.1
- [12] V. Nagaslaev, K.A. Brown, M. Tomizawa, Phys. Rev. Accel. Beams **22**, 043501 (2019) 1.1
- [13] A. Czarnecki, X. Garcia i Tormo, W.J. Marciano, Phys. Rev. D **84**, 013006 (2011), [1106.4756](#) 1.1, 7.2
- [14] R. Szafron, Acta Phys. Polon. B **48**, 2183 (2017) 1.1, 7.2
- [15] O. Atanova et al., JINST **12**, P05007 (2017), [1702.03720](#) 1.1
- [16] F. Abdi et al. (Mu2e Collaboration), Universe **9**, 54 (2023), [2210.11380](#) 1.1, 3, 7.2, 7.3, 7.4, 7.8, 8, 10.1
- [17] R. Bernstein, S. Werkema (Mu2e Collaboration), *Proton Beam Requirements Document* (2016), mu2e-doc-1105 1.2
- [18] R. Ray (Mu2e Collaboration), *Key performance parameters* (2016), mu2e doc-db 4665 1.3.1
- [19] D. Brown et al. (Mu2e collaboration), *B field task force report* (2023), mu2e doc-db 46856 1.3.4
- [20] D. Brown, B. Echenard (Mu2e Collaboration), *Analysis Core Group Charges* (2024), mu2e-doc-48639 2.1
- [21] D. Brown, B. Echenard (Mu2e collaboration), *Analysis core group charges* (2024), mu2e doc-db 48639 2.1
- [22] A. Gioiosa et al., *Trigger-DAQ and Slow Controls Systems in the Mu2e Experiment*, in *22nd IEEE Real Time Conference* (2020), [2010.16208](#) 4, 4.1
- [23] A. Gioiosa et al., PoS **EPS-HEP2021**, 823 (2022) 4
- [24] K. Biery, E. Flumerfelt, A. Lyon, R. Rechenmacher, R. Rivera, M. Rominsky, L. Upfinger, M. Votava, *The Fermilab Test Beam Facility Data Acquisition System Based on otsdaq*, in *21st IEEE Real Time Conference* (2018), [1806.07240](#) 4.1
- [25] C. Green, J. Kowalkowski, M. Paterno, M. Fischler, L. Garren, Q. Lu, J. Phys. Conf. Ser. **396**, 022020 (2012) 4.1, 8
- [26] I. Mandrichenko (DUNE), EPJ Web Conf. **251**, 02048 (2021) 5.1
- [27] M. Barisits et al., Comput. Softw. Big Sci. **3**, 11 (2019), [1902.09857](#) 5.1
- [28] A. Agostinelli et al. (Geant4 collaboration), Nucl. Instrum. Methods Phys. Res. Sect. A **506**, 250 (2003) 7
- [29] J. Allison et al., IEEE Trans. Nucl. Sci. **53**, 270 (2006)
- [30] J. Allison et al., Nucl. Instrum. Methods Phys. Res. Sect. A **835**, 186 (2016) 7

- 1 [31] S. Mashnik et al., Nucl. Instrum. Methods Phys. Res. Sect. Beam Interact. Mater. Atoms
 2 **267**, 3426– (2009) 7
 3 [32] T. Goorley et al., Nucl. Technol. **180**, 298– (2012) 7
 4 [33] G. Battistoni et al., J. Phys. Conf. Ser. **408**, 012051 (2013) 7
 5 [34] T. Sato et al., J. Nucl Sci Technol. **55(6)**, 684 (2018) 7
 6 [35] C. Hangman et al., Proceedings of the 2007 IEEE Nuclear Science Symposium Conference Record, Honolulu, HI, USA **2**, 1143 (2007) 7.1
 7 [36] R. Szafron et al., Phys. Rev. D **94** (2016) 7.2
 8 [37] A. Gaponenko et al., Phys. Rev. C **101**, 035502 (2020) 7.2
 9 [38] A. Edmonds et al., Phys. Rev. C **105**, 035501 (2022) 7.2
 10 [39] T. Kozlowski et al., Nucl. Phys. A **305**, 368– (1978) 7.2
 11 [40] B. Macdonald et al., Phys. Rev. B **139**, 1253 (1965) 7.2
 12 [41] P.C. Bergbusch et al., Phys. Rev. C **59**, 2853 (2022) 7.2
 13 [42] P. Christillin et al., Nucl. Phys. A **345**, 331 (1980) 7.2
 14 [43] J. Bistirlich et al., Phys. Rev. C **5**, 1867 (1972) 7.2
 15 [44] D. Heck et al., Karlsruhe Forschungszentrum Technical Report **6019** (1998) 7.2
 16 [45] Opera (Opera Collaboration), *Opera. electromagnetic and electromechanical simulation*, <https://www.3ds.com/products/simulia/opera/> 7.6
 17 [46] H. Glass et al., *The helicalc integration package* (2022), mu2e doc-db 41682 7.6
 18 [47] A. Gaponenko (Mu2e Collaboration), *A silicon pixel based extinction monitor detector* (2014), mu2e doc-db 2480 7.7
 19 [48] D. Brown, EPJ Web Conf. **295**, 06001 (2024) 8.2
 20 [49] R. Tscherhart et al. (Mu2e Collaboration), *Trigger Requirements Document* (2015), mu2e-doc-1150 8.2
 21 [50] R. Bernstein, A. Mukherjee (Mu2e Collaboration), *Tracker Requirements Document* (2014), mu2e doc-db 732 8.2
 22 [51] V. Blobel, C. Kleinwort, *A new method for the high-precision alignment of track detectors* (2002), [hep-ex/0208021](https://arxiv.org/abs/hep-ex/0208021) 9.1
 23 [52] V. Blobel, *Millepede ii draft manual* (2007), https://www.desy.de/~kleinwrt/MP2/doc/html/draftman_page.html 9.1
 24 [53] C. Kleinwort, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **673**, 107 (2012) 9.1
 25 [54] A.M. Tadel, M. Tadel, A. Yagil, D. Kovalskyi, S. Linev (CMS), EPJ Web Conf. **245**, 08027 (2020) 11
 26 [55] M. Bellis et al. (2018), [1811.10309](https://arxiv.org/abs/1811.10309) 11
 27 [56] *SCons: A software construction tool*, <https://scons.org/> 13.2
 28 [57] Mu2e (Mu2e Collaboration), *Mu2e Data Management Plan*, <https://publicdocs.fnal.gov/cgi-bin>ShowDocument?docid=515> 14
 29 [58] D. Brown et al. (Mu2e Collaboration), *Mu2e Software Policy* (2019), mu2e-doc-26349
 30 14