Project Motivation and Impact

1.1 Project outline

The proposed work targets an unmet cyberinfrastructure need in the direct-detection dark matter and computational astrophysics fields: easy, extensible access to binary data from multiple, volatile sources. This work is aligned with the MPS/PHY and MPS/AST divisions and the CISE directorate.

Understanding dark matter has been identified as a programmatic priority by both the NSF and DOE. The next generation of dark matter experiments must understand their backgrounds to an unprecedented level to be able to discover dark matter. Combining data from detectors that are co-located can provide much-needed insight on the identity of possible dark matter signals. This is currently impossible because the data formats and analysis software are incompatible. The proposed infrastructure will enable this type of analysis, and the PICO and IceCube collaborations have committed to pilot analyses during the grant period. See Section 1.2.1 for further discussion of this project's impact on the dark matter field.

Astrophysics seeks to understand the matter that makes up our universe, and how that impacts observable quantities. With the advent of multi-messenger astrophysics, it is critical that astrophysical simulations and experimental data be accessible for easy cross-analysis. `yt` has made strides toward this and the proposed work would lower the barrier to including new data sources, increase the maintainability of the `yt` codebase, and increase the discoverability and accessibility of data sets across the field. See Section 1.2.2 for further discussion of why the proposed infrastructure is critical to computational astrophysics.

We propose to design, implement and deploy a data-delivery service that explicitly tuned to the data challenges of existing dark matter and astrophysics experiments while requiring no changes to existing data formats. This non-invasive, no-changes-necessary support for arbitrary and custom file formats provides opportunities to expand beyond our two identified use cases into many other data-driven science domains that may utilize custom file formats. As part of our deployment and development, we will reach out to additional communities throughout the grant period. PI Roberts is ideally suited to reach out to the broader dark matter community and Co-PI Turk has recently been funded specifically to expand the use of `yt` in data-intensive fields. See Section 1.2.3 for further details.

This proposed work delivers an infrastructure that seamlessly delivers data in a well-supported format (such as Parquet) from multiple sources. We call this infrastructure PONDD (Personal Data Delivery). To successfully deliver cross-experiment data to end users, we bring together ongoing projects from High Energy Physics and the broader NSF community; while this project will involve development of software products (as detailed below) it will also include synthesis of *existing* investments in cyberinfrastructure and efforts to improve their long-term sustainability. Figure 1 shows the envisioned architecture and Table 1 outlines the necessary projects and who will provide support for integrating these projects into PONDD.

We envision PONDD as a user-interface for accessing and acting on data, enabling science-focused, results-driven inquiry backed by straightforward, maintainable data format definitions.

1.2 Science Drivers

1.2.1 Direct-detection dark matter searches: For the next generation of dark matter experiments to be able to discover dark matter, experiments have to understand their backgrounds to an unprecedented level [2]. Combining data from detectors that are located in the same place can provide much-needed insight on the identity of possible dark matter signals, especially when these detectors use complementary technology. But this is currently impossible because the data formats and analysis software are incompatible. Combining data from multiple experiments could solve decades-long mysteries in the field, help answer new questions, and enable conclusive discovery of new phenomena. PICO [3] and SuperCDMS [4] have both committed to working together on a pilot analysis. These two dark matter experiments are located in the

Table 1: This project combines multiple, ongoing projects to deliver a novel data-delivery service.

PONDD component	Funded integration lead	Additional support
Network-accessible data storage	UC Denver Postdoc	Open Storage Network, Jetstream
Dataset identifier service (Rucio)	UC Denver Postdoc	IRIS-HEP Sustainable Systems Lab, Jetstream,
Data filtering and transformation (ServiceX)	UC Denver Postdoc	IRIS-HEP Sustainable Systems Lab, Jetstream, Ben Galewsky
Format conversion code (Kaitai Struct)	Ben Galewsky	Kaitai Struct, Amy Roberts
Volumetric analysis software (`yt`)	Kacper Kowalik	`yt`, Matthew Turk
End-user analysis environment (JupyterLab)	UC Denver Postdoc	XSEDE Extended Collaboration Support Services, Jetstream, Amy Roberts
Community training	Funded fellowships	Community Engagement Board, Amy Roberts, Matthew Turk
Documentation stress-testing	UC Denver undergraduates	Amy Roberts

SNOLAB underground science facility. PICO is already installed at SNOLAB and SuperCDMS-SNOLAB is currently being built in a nearby drift; the first data will be available at the beginning of this grant period.

In addition to constraining backgrounds, combining experimental data has the potential to address additional science needs of the dark matter community:

Annual variation in signal is a common method used for dark matter searches. But these analyses are extremely sensitive to any variations in the environment or detector setup and the positive signal seen by DAMA [5] has never been replicated. Using timestamps to combine data from multiple detectors to identify correlated variations has the potential to resolve decades-long tension between experiment results and provide a path forward for new types of analysis.

Connecting data from experiments around the world will create a global telescope of detectors that are designed to be sensitive to the smallest particle signals ever detected. Such a network offers unprecedented ways to search for new phenomena. IceCube, a high-energy neutrino experiment operated at the North Pole, has committed to participating in a pilot analysis using the proposed infrastructure. Overlapping data with other experiments at the SNOLAB science facility could reveal planetary-scale cosmic ray showers and could provide new types of constrains on annual modulation searches.

1.2.2 Computational Astrophysics: `yt` [6] is an open-source, community-driven platform for analysis and visualization of volumetric data. Originally created to analyze the outputs from astrophysical simulations, with the support of the NSF (through the CSSI and SI2 programs) it has expanded in recent years to include support for geodynamics, weather, nuclear engineering, and molecular dynamics datasets. The core philosophy of yt is that the mechanisms by which researchers interact with data should be independent of the representation of that data, both on-disk and in-memory. Practically, this means that `yt` provides a three-tiered data analysis platform; the lowest tier is that of data ingestion, the second is regularization of data, and the third (and most user-facing) data visualization and application of domain-specific models. In essence, `yt` applies a "grammar" of analysis to volumetric data, which can be applied to different discretizations (such as Lagrangian, Eulerian, finite element, or hybrid methods) and which abstracts the specifics of the file formats from this grammar. Rather than specifying the process of indexing and loading subsets of data into memory, conducting reductions or visualizations and then discretizing these into pixel buffers, `yt`

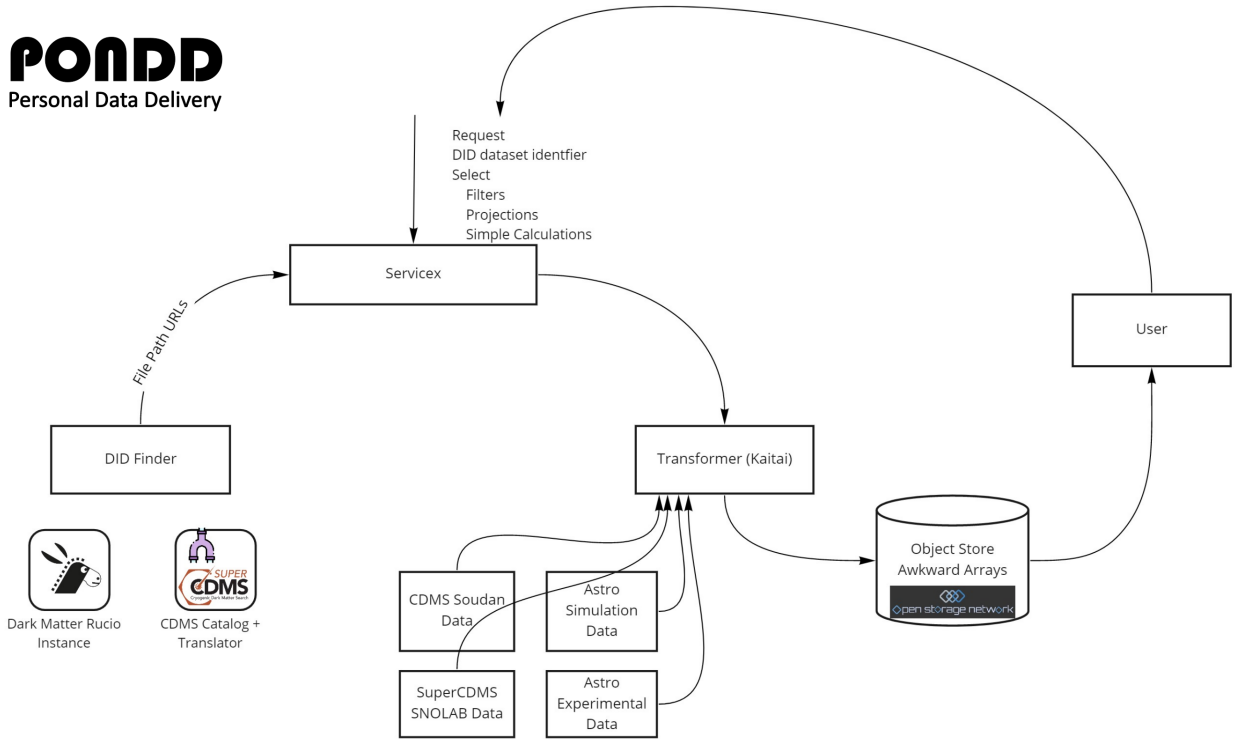

Figure 1: ServiceX[1] developed as part of the IRIS-HEP Software Institute, will accept requests for data and handle reliable orchestration of parallel transformation data files into supported data structures. Kaitai is a data-description language that is well-suited for scientific data and its compiler will operate as a transformer for ServiceX. The Open Storage Network will store both the static, custom format data and also the supported-format data that is requested during analysis. We will provide user analysis environments through a JupyterHub instance on XSEDE and deploy analysis software through CVMFS to make it easy to support analysis sites on other computing clusters. Finally, users will need to have their data registered in a ServiceX-compatible catalog. For users with no existing data catalog infrastructure, a Rucio instance will allow users to specify data sets to ServiceX that are needed for analysis. For users with existing catalogs, we will develop a ServiceX "plugin" that sits on top of the existing catalog and provides compatibility with ServiceX. Some of the initial users of this infrastructure will be SuperCDMS and PICO (co-located dark matter experiments that use different detector technologies) and IceCube (a neutrino observatory installed at the South Pole).

abstracts this process behind an API focused on physically-meaningful operations and semantics.

In order to enable these operations in an efficient, memory-saving fashion, `yt` conducts indexing at both the coarse and fine levels; for gridded (finite volume) datasets, this is accomplished through traditional octree or R-tree based indexing. For discretely and irregularly sampled datasets, often representing particles, `yt` utilizes a hierarchical (compressed) bitmap indexing scheme to minimize unnecessary data reads. The second-level selection process is where fine-grained cuts are applied, so that data selections only include those data points actually necessary for the calculation. By minimizing memory-consumption and prevent overzealous allocations, `yt` ensures that extremely large datasets are accessible with even modest resources. `yt`-centric widgets enable interactive access to large-scale datasets in Jupyter, as well as both software- and hardware-based volume rendering [7]–[10].

Combining this abstraction of data with multi-level indexing greatly increases the efficiency and depth with which researchers can analyze data; unfortunately, it also requires a fairly structured approach to data ingestion that often necessitates careful management of memory and IO. At present, `yt` is able to read data in several dozen different output formats, in some instances including multiple "boutique" formats customized

by individual research groups. In general, data formats in computational astrophysics take on only a handful of forms, although the details of each format vary considerably. Some simulation codes such as Enzo [11], [12], FLASH [13] and GAMER [14] organize their data using HDF5 [15], which abstracts the representation of integer, floating point and string values so that they can be accessed in a uniform manner. Other simulation codes, such as ART [16], RAMSES [17], 2HOT [18] and some deployments of Gadget [19] utilize binary formats that more directly reflect the internal memory organization of the running calculation. Some, such as Boxlib-based codes (such as, e.g., Orion [20]) provide a mixture of type information with binary floating point representation. Developing systems for indexing-in-advance, and thus mapping selections in coordinate-space to file operations, requires ingestion and indexing routines be written for each data format to be read. At small scales these routines and operations are straightforward to develop; managing access to a handful of data file formats is straightforward and tractable. However, long-term sustainability of the overall platform requires that these formats occasionally be updated to better match filesystem performance needs, to address scalability issues on next-generation supercomputers, and to update the file formats to new revisions while retaining backwards-compatible support for older data files. When taking into account the burgeoning landscape of numerical array backends (such as numpy, xarray, dask, RAPIDS, jax, xnd [21]–[25] and the requirements of backward-compatibility of existing scientific workflows, this task increases in both scope and magnitude.

Providing access to these datasets is not simply of academic interest; dark matter simulations [18], [19] provide calibrations for large-scale telescope surveys [26], [27], while simulations of galaxy formation and provide context and deeper insight into observations [27], [28] and simulations of compact objects can help to understand observations from sources such as the Event Horizon Telescope [29]. By ensuring that `yt` is able to provide access to legacy, modern and *emerging* classes of simulations, we ensure that the mechanisms and methods by which simulations are accessed and processed to provide insight remain stable and usable while the underlying data formats may shift and change. Providing access to the "dark data" of *observational* astrophysics is an ongoing concern [30] being addressed even within the CSSI community. `yt` is usable to access the "dark data" of *computational* astrophysics, and in this project we will enable it to be more flexible with respect to data format specifics, thus reducing the barrier to entry for individual researchers, and to present a wider range of access through utilization of Rucio and ServiceX.

1.2.3 Data-intensive science domains: The need to analyze data from multiple sources is not unique to the dark matter or astrophysics communities. Co-PI Turk was recently awarded a grant to expand the use of the `yt` package to Meteorology, Seismology, Nuclear Engineering, Plasma Simulations, Observational Astronomy, and Hydrology. PI Roberts maintains strong ties to the experimental nuclear physics community, where the need to combine data from separate detector systems is growing as the Facility for Rare Isotope Beams comes online. And the need for analysis across experiments defines the multi-messenger astronomy field, which has recently created a backbone organization to address the challenges of this type of analysis [31].

The proposed work fills an infrastructure need that affects nearly all scientific communities whose data needs are growing into the gigabyte-to-terabyte range. High-energy physics has successfully organized and led software efforts to plan for their upcoming influx of data. This project leverages services built by the HEP community and applies them to problems in our fields of expertise. We will actively recruit members from other scientific communities facing similar data challenges.

1.3 Innovation

The proposed work innovates by combining existing, well-supported projects to provide a service that meets the needs of the dark matter and astrophysics communities: enabling analyzers to combine data from multiple experiments with no significant infrastructure changes required by those experiments.

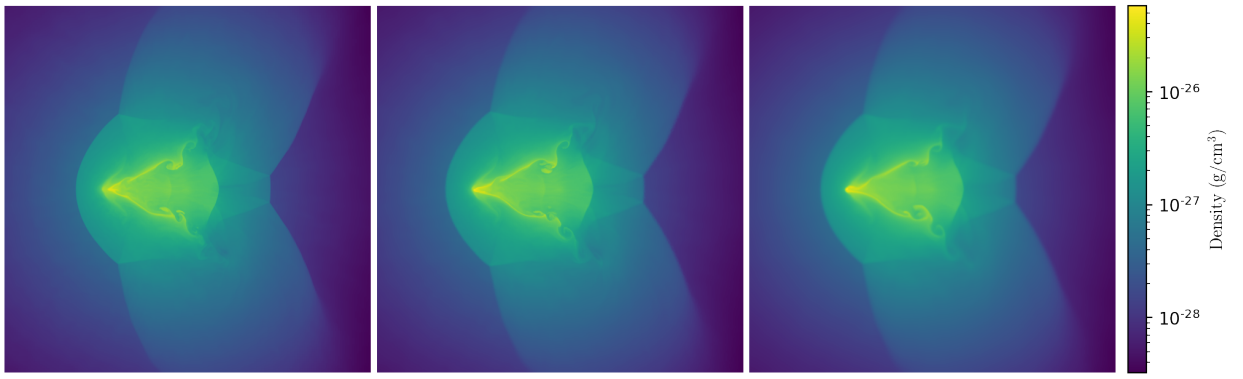

Figure 2: `yt` can be used for both in-depth analysis of computational astrophysics simulation results as well as direct comparisons across different simulation methodologies. In this figure, (courtesy of John ZuHone), we demonstrate `yt`'s ability to directly compare across codes. From left to right, the simulation platforms represented are GAMER-2 (grid-based), AREPO (moving mesh adaptive Lagrangian-Eulerian) and GIZMO (meshless particle-based). `yt` is able to process each of these different simulation outputs utilizing the same commands for each, enabling direct comparisons of quantities of interest and analysis that is independent of the simulation data format. The images are slices through the density field of a galaxy cluster simulation, 3 Mpc on a side, where each simulation type requires a different method of discretization to construct pixel buffers, as well as data access.

Some individual collaborations are in the process of building similar data-delivery systems, but no service exists that can take in different types of data and provide common-format files for analysis. There are two projects that have aims similar to PONDD: Frictionless Data and Intake.

Frictionless Data [32] has a nearly identical mission to the proposed work - allow people to work easily with datasets from multiple sources. However, Frictionless Data has focused on fixed-column text data and has not yet tried to support binary data and does not have the personnel to do so at this time.

Intake [33] is a promising new project supported by Anaconda that also aims to deliver data from multiple sources to end users. Intake supports a wide array of data formats, but currently supports none of the highly-custom formats of the SuperCDMS collaboration. This is not surprising; formats used by scientific collaborations are usually understood by a handful of people. Intake provides an extensible way for developers to add support for new file formats. But this still poses a significant barrier to using this software for a scientist as writing a parser for binary data requires significant file i/o expertise. In addition, Intake does not provide the ability to pre-process or filter data. This is a critical ability for experiments whose data sets are growing into the hundreds of GB range.

The PONDD framework will give end users access to data from multiple experiments that has been converted into a standard format (Parquet files) and that can be pre-filtered. The type of data (custom-format binary), the size of the data (GB to TB), and the ability to query across multiple, custom-formatted, binary data sources make the proposed work unique.

Broader Impacts

There is a clear need for broadening participation in research at the undergraduate level, as well as increasing data and computational literacy for STEM majors [34]–[36]. Accessible tools are a recommended practice for broadening participation in data science [35]; these web-accessible tools lower the barrier to entry for all students. Accessible entry to research is particularly critical for students from underrepresented groups—groups that are often less likely to use university resources [37]—and the existing analysis workflows in both dark matter and computational astrophysics require students to be willing to demand significant amounts of an experts' time.

PI Roberts has led a transition to web-based analysis environments within the Cryogenic Dark Matter Search collaboration, doubling their undergraduate participation in science analysis. In addition, both graduate students and postdoctoral researchers have embraced the new system as providing a quicker onboarding experience: previously, every user had to compile their own analysis tools. But accessing data stored across many different systems and accessing data stored in old formats still hinders analysis.

The fundamental goal of the proposed work is to make data easily accessible to individuals who want to answer science questions. The immediate target audience for this work are active scientists, early-career researchers, and undergraduate students who are participating in research that require access of custom-format binary data from multiple experiments. While this infrastructure is designed to make it possible to do cross-experiment analysis, it also has the potential to lower barriers for analysis within individual experiments. Any experiment that can deploy or connect to PONDD will be able to access their data with well-supported analysis libraries. This reduces the need to use specialized software to analyze experiment data and reduces the risk of unintended gatekeeping.

The proposed work will create the kernel of an ecosystem that is usable, well-documented for both novices and experts, and enjoys the support of a community that is accessible online. All of these features increase equity in science access by reducing "secret knowledege" within collaborations. To maximize the broader impact, we will focus development efforts on easy deployment of the PONDD service. PI Roberts will also work with experiments using PONDD to track project descriptions and duration for undergraduate students during the period of the grant.

Cyberinfrastructure Plans

3.1 Building on existing, recognized capabilities

PONDD combines existing, well-supported projects and services to enable analyzers to easily combine data from multiple experiments. It is possible to build such an ambitious service because of these already-existing building blocks and because, in many cases, their parent organizations are funded to support their adoption. `yt` is an NSF-supported project, as is IRIS-HEP (ServiceX, Rucio), the Open Storage Network (object store), and Jetstream (JupyterHub). The Science Community Gateways Institute is a "glue" organization that will provide advising on our infrastructure security and sustainability. Below we describe the role of each project within PONDD.

3.1.1 The role of ServiceX: ServiceX [1] provides users a way to seamlessly request data within their analysis scripts. It provides "transformers" that can perform simple format changes or more complex filtering requests and thus provides both a way to turn data from a custom format into a common, well-supported format and to easily reduce the size of data.

The pluggable ServiceX "transformer" is the critical piece that allows end users to request data from different experiments and get data in a consistent, supported format despite the custom format of the original data.

3.1.2 The role of Rucio: Rucio [38] is a dataset identification program that allows experiments to specify data locations across an astonishing array of storage types. However, Rucio delivers data as-is and has no ability to transform it into a common format. A community Rucio instance would allow data access across experiments but this data would be unusable to analyzers because of the different formats. Rucio (and other dataset-identifier services) must be paired with a program like ServiceX that transforms the data into something usable by common data-access libraries.

A community Rucio instance allows experiments without existing data catalogs to participate. We chose Rucio because it is one of the few dataset identifier services that is supported by an active community and IRIS-HEP has extensive experience with it.

3.1.3 The role of Kaitai Struct: Several tools exist for data-description; we identify three that are directly of relevance to this project. The first is the Python-native tool Construct. Construct presents a semi-declarative approach to data format definitions; the format definitions are defined in python code which can consist of immutable or runtime-determind characteristics. While this does provide flexibility, it also intertwines the "compiler" with the "definition" in a way that restricts extraction. Additionally, the development of Construct has undergone several periods of sustained inactivity, and it does not currently have feature parity with Kaitai. The second alternative software package is the (incubating) Apache project Daffodil, based on the DFDL (data format description language) specification. While this project provides an extensive mechanism for specifying data formats and translating those formats into in-memory representation, the relatively heavy dependencies of the software stack and the complex specification posed difficulties for researchers whose needs may be much simpler than those supported by a comprehensive solution such as DFDL. The final option we identified is that of either developing a small domain-specific language for data formats of our own, or one that is based on the numpy data type specification. This is, in many ways, the *status quo* for researchers; in fact, it is represented in `yt` in several data ingestion formats. Where Kaitai Struct distinguishes itself is in three primary areas: 1) it provides multiple control flow options including iteration, stream-length and termination criteria; 2) it allows for a composable types that can be parameterized; 3) the format-description definition is independent of the target language.

3.1.4 The role of `yt`: As described in §1.2.2, `yt` provides access to many different forms of volumetric data, including but not limited to computational astrophysics. There are other tools that provide access to volumetric data, such as traditional scientific visualization software such as VisIT, ParaView and MayaVi [39]–[41]; one particularly challenging aspect of utilizing the first of these remains the binary data format problem that we intend to address in this work. Fortunately, the data format descriptors we intend to develop will remain "platform neutral" in that they will not be tied explicitly to `yt`. Because they will be accessible independently, and as a byproduct of the polyglot nature of Kaitai, we hope to enable the work done here to be usable in those alternate platforms, and to provide multiple methods of access to computational astrophysics data outside of `yt`. From the perspective of computational astrophysics, `yt` serves a plurality of the community; it is widely-cited with a thriving, active community, and it is the obvious choice for an implementation platform in this project. But, as noted above, the development here will not be limited to just `yt` users; in principle, individual researchers seeking alternate methods of analysis and visualization will be able to utilize the data format descriptors here in their own projects. This directly fosters both *reproducibility* and *replicability*, as we are ensuring the long-term sustainability of both the methods and implementations of analysis of computational astrophysics results. Other projects exist that provide access to computational astrophysical data; in fact, several of these are embedded within the Open Storage Network itself. These include established, generic platforms such as SciServer, as well as custom systems such as the Illustris public data platform [42], [43]. We do not see the integrated stack we are proposing as in competition to these other projects, but rather complementary, as it can provide additional methods for data retrieval and access for those platforms.

3.1.5 The role of the Open Storage Network: The Open Storage Network provides well-supported object storage to the national computing infrastructure. This storage complements the computational power available through XSEDE.

Partnership with the Open Storage Network ensures that experiments who wish to participate in this pilot project but whose data is not network-accessible have a place to copy data. It is still common for small-scale experiments to store data on disks that are inaccessible through https or s3 protocols.

The Open Storage Network serves a second purpose in this project. The transformed data that ServiceX produces must be stored somewhere, and the Open Storage Network provides ammple space for this on-demand data.

On its own, the Open Storage Network offers only file storage. It must be paired with services to query and locate the data (Rucio) and transform data into a common format (ServiceX) to provide an accessible analysis environment to end users.

3.1.6 The role of JupyterHub: Many in the scientific community have mixed feelings about Jupyter's web-based environment, but within the SuperCDMS collaboration the web-accessible python environment, text editor, and terminal have been an unmistakable step forward in productivity for undergraduate students, graduate students, and even postdocs. The Jetstream-hosted JupyterHub server provides anyone who logs in with an instant IDE that is already set up for analysis of SuperCDMS data sets.

We choose the Jetstream-hosted JupyterHub analysis environment as the first target for PONDD integration for several reasons. First, the Jetstream resources provide extraordinary flexibility and control and we anticipate having all the necessary privileges to successfully integrate PONDD. Second, JupyterHub (deployed to access software distributed through CVMFS) is the officially-supported analysis environment of the SuperCDMS collaboration. If PONDD is successful, this service could become part of our permanent analysis infrastructure. Finally, integrating PONDD with a Jetstream-hosted JupyterHub server is one of the most portable ways to deploy the service: any experiment interested in using the service can request an allocation to also spin up a JupyterHub server on Jetstream.

3.2 Project plans, and system and process architecture

3.2.1 Timeline: By the end of the grant period, we will deliver software and services that work together to provide data delivery to any experiment in the dark matter and computational astrophysics communities. Milestones for this work are shown in Figure 3.

In the first two years of the grant, the UIUC team will deliver two pieces of software: (1) `yt` will be refactored to support Awkward1.0 and (2) Kaitai will be extended to include Awkward1.0 as a target. Awkward1.0 provides an in-memory representation optimized for scientific data and is compatible with a wide array of common analysis libraries such as Arrow, Pandas, and numpy. Implicit in these efforts will be conversion of the extant data formats in `yt` to the Kaitai data format description. These software improvements alone provide value to the scientific community and together they make the data service usable by any collaboration able to describe their data.

Testing the service infrastructure will proceed in parallel with these software efforts; this effort will be led by the UC Denver team and supported by IRIS-HEP. The first year will focus on deploying services and ensuring that they work together; a community Rucio instance, ServiceX instance, OSN storage, and JupyterHub server will be available and working together to deliver data to analyzers. Efforts after this first proof-of-concept will focus on making these services sustainable by creating ready-to-deploy packages on the XSEDE Jetstream system and working with additional experiments to stress-test the documentation.

Several proof-of-concept analyses will be produced throughout the grant period. The first will be completed during the infrastructure testing in the first year, using a data format already supported by ServiceX. The second will be completed during the `yt` and Kaitai software efforts, shortly after the first year. This analysis will use `yt` and Kaitai to do a simple analysis on data in a custom format. Finally, an analysis testing the full capability of the system will be done in the second year. This analysis will combine data from multiple experiments - exercising the full capabilities of PONDD - and will be performed by the core team supported by this grant. The grant also provides funds to train analyzers from additional experiments to use PONDD, with the goal of additional analyses and additional feedback. Experiments will be welcome to participate in the community through informal methods like Slack and github issues. They will also be able to participate more formally by joining the Community Engagement board.

3.2.2 Engineering processes for the design, development, and release of the products: The proposed work makes extensive use of development infrastructure and methods already in place through IRIS-HEP

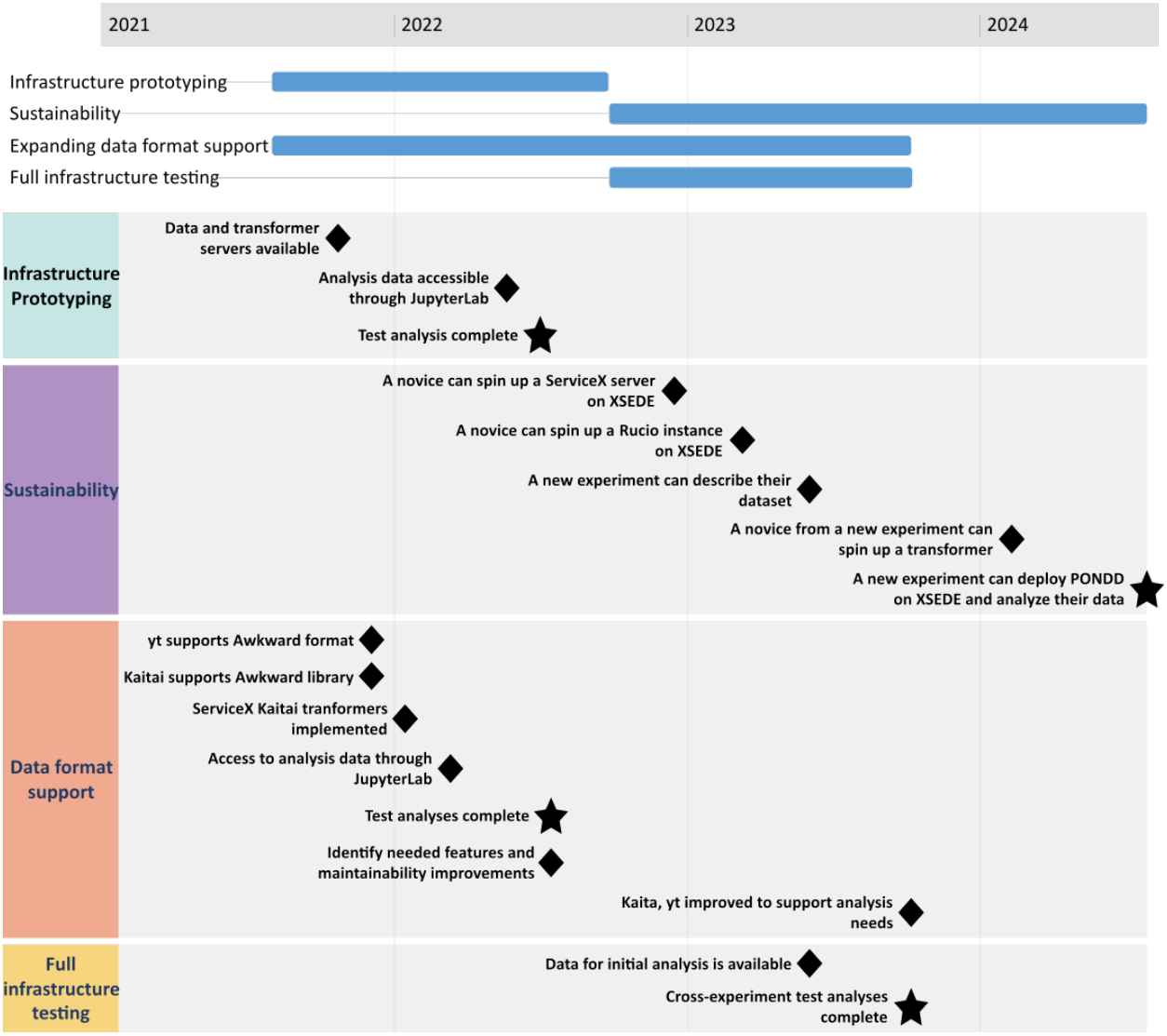

Figure 3: The first year will focus on deploying services and ensuring that they work together; a community Rucio instance, ServiceX instance, OSN storage, and JupyterHub server will be available and working together to deliver data to analyzers. Efforts after this first proof-of-concept will focus on making these services sustainable by creating ready-to-deploy packages on the XSEDE Jetstream system and working with additional experiments to stress-test the documentation. The UIUC team will work in parallel for the first two years to deliver two pieces of software: (1) `yt` will be refactored to support Awkward1.0 and (2) Kaitai will be extended to include Awkward1.0 as a target. Analysis efforts are integrated throughout the entire development and will involve data from the SuperCDMS, PICO, and IceCube collaborations. These are marked with stars.

and SGCI. Ben Galewsky (UIUC) is a developer with IRIS-HEP and brings strong leadership on ServiceX development and integration. PIs Roberts and Turk are active in research in their scientific disciplines and bring knowledge of the data needs and science needs of their communities. This team is ideally suited to combine HEP software infrstructure to meet the needs of the dark matter and astrophysics communities.

Design of the software and service deployment will be done in consultation with the current developers. All design will begin with a requirements document to ensure that the final product is able to meet the science needs. In all cases, design decisions will be implemented in the smallest increments that allow meaningful testing. Where needed, we will seek expertise from IRIS-HEP and other outside experts. In particular, we plan to work with SGCI on security and authentication design.

Development of the software and services will follow well-established CI practices. The development team will version control all software with git. All the software and services that are part of this grant are currently developed in the open on github and these practices will continue for this grant work. The development will have weekly working meetings to check in on progress and ensure that questions on integration are quickly addressed.

Documentation of the software and services will follow the standards already in place for each package. All of the packages (`yt`, Kaitai, ServiceX, and Rucio) have existing documentation that includes instructions for end users and for contributors. This documentation is versioned in git and automatically deployed on static sites. We anticipated contributing to this documentation throughout the grant period and will do so according to the contribution instructions already established by each project.

PONDD is a collection of these services and we will develop documentation for the deployment and integration of these packages. We will use mkdocs [44] to build the documentation and host the documentation on github pages, a free service [45]. The mkdocs package uses a lightweight markdown syntax and compiles the documentation into a searchable website. The participation of multiple experiments and dedicated funding for undergraduate students to stress test the deployment documentation will help ensure complete, functioning documentation.

Testing and validation of all software will follow the standards already in place for each package. Kaitai has extensive tests that validate the ability of the generated software to read different binary formats. `yt` requires that all contributions pass unit tests against multiple python versions and against multiple operating systems. `yt` also enforces code style with a linter service. ServiceX has extensive unit testing in the system and measures the percentage of lines of code that are covered by the tests. The main ServiceX application is currently at 90% code coverage and code submissions must at least maintain this coverage.

Release and deployment All software packages use semantic versioning [46] for their release labels and we will conform to this practice. Semantic versioning allows users and developers to identify breaking changes and makes the decision to update easier. Some packages (`yt`) are deployed via python package managers automatically as part of the release process. The latest release - or a user-specified release - of all the other packages can be downloaded via the github API.

PONDD is a collection of these services, wired to work together to deliver data to a user. PONDD will also use semantic versioning for its releases. The initial goal for deployment will be that a novice can deploy an updated version of PONDD in less than one day. During the sustainability efforts in the second two years of the grant, the goal will be to automatically deploy an updated version of PONDD on XSEDE whenever the release version on the repository is updated.

Acceptance and evaluation by the end users is integrated into every stage of the development process. Dedicated test analyses are the final milestone of the infrastructure prototyping and data format support efforts. A cross-experiment analysis is the final milestone of the combination of these two systems, and

the final milestone of the sustainability efforts is a test analysis done by a different experiment. The SuperCDMS, PICO, and IceCube data will be available for analysis milestones. See Figure 3 to see the timing of each of these analysis challenges; analysis milestones are marked with the "star" symbol.

Security: To make user access as seamless as possible, we will implement cilogon for authentication. We will work with SGCI technical consulting, trustedci.org, and IRIS-HEP to identify issues and solutions during the design phase. Co-PI Turk has led multiple projects that require authentication management and his existing working relationships with cybersecurity experts will be helpful during this implementation.

This project will ultimately deploy services on the XSEDE cloud provider Jetstream. We will work with Extended Collaboration Support Services and SGCI to determine appropriate measures to protect against the misuse of these services.

Integrity and Provenance: The proposed work leverages existing projects wherever possible. This creates a more sustainable project and also a more secure and reliable project. The Open Storage Network and the Jetstream object storage will provide the storage for initial, "custom format" data and also for data transformed by ServiceX that is ready for analysis. Both systems provide user authentication systems and disk quality control. In addition, the services we use to deliver the data, Rucio and ServiceX, have the ability to verify the checksums of transferred data. We will take advantage of these pre-existing capabilities and add end-to-end tests to our system that verifies the output given a known data input.

3.3 Close collaborations among stakeholders

This project builds on, extends or deeply utilizes several community-driven open source projects. As such, it requires a delicate balance between funded development driving enhancements and the potentially unfunded individuals in the community that utilize the software projects. To mitigate potential friction, we have developd a comprehensive strategy of community engagement in an effort to balance these two competing concerns and ensure that our project can effectively make progress while ensuring that the development is accepted by the community.

Of the software stack we intend to utilize and develop, the Kaitai Struct and `yt` projects will require the most engagement during the development process. `yt` is developed by a community of researchers and developers, with clearly defined standards for code review, discussion of breaking and incompatible changes (the YTEP, or "`yt` enhancement proposal" process), and an steering committee (which includes PI Turk). During the conceptualization and design phase of the project, we will develop a YTEP in collaboration with the `yt` community that describes the changes to be made to `yt` to support Kaitai Struct (and, more broadly, declarative file formats) as an ingestion mechanism. We anticipate this to be developed as a "living document" that grows and changes as the implementation itself is developed; this matches previous development processes for potentially-invasive changes in `yt`. This project and `yt` share deep personnel ties, and while we fully-intend to follow established community standards, we note that the standards were developed with the intention of enabling work such as this to proceed.

The Kaitai Struct project is composed of several components, including (but not limited to) a reference compiler implementation, runtime libraries for each of the supported languages, a web-based interactive development environment (IDE), a substantial number of existing data format definitions, and a comprehensive continuous integration suite. Kaitai Struct is developed in the open on GitHub, but without as clear of a method for proposing changes to the underlying code base as `yt`. The process of extending or adding a feature to the existing data format definition requires both community-acceptance of that new feature, as well as enabling it in at least the reference compiler and one or more of the language-specific runtime libraries. In our proposed utilization of Kaitai Struct, we have attempted to *minimize* any invasive changes to the language; in particular, while our proposed work may require that some aspects be extended, we have done so in such a way that provides clear, staged and incremental changes that can be evaluated individu-

ally, to minimize the risk of the upstream developers declining them. More importantly, however, is that we intend to foster a collaborative, *mutually-beneficial* relationship with the Kaitai Struct community, and to participate as peers, contributing time and development energy to the project. We have connected with the principle developer of Kaitai Struct Mikhail Yakshin and intend to ensure that any work we do is done in a way that respects community standards and norms, and provides value to the project.

Measurable Outcomes

4.1 Deliverables

The primary deliverable of the proposed work is a production quality hosted service that accepts requests specifying: (1) dataset identifiers, (2) a file format mapping specification, and (3) filtering and projection criteria. We will integrate the service with JupyterHub (deployed on Jetstream) to provide a complete analysis environment. The full list of products, along with their sustainability plan after the grant period, are below.

- A ServiceX instance and transformers
  Sustainability plan: The ServiceX instance will initially be managed by IRIS-HEP. Over the course of the grant PI Roberts' group will increasingly manage this instance and by the end of the grant it will be deployed on Jetstream. Delivery to the community will consist of (1) a package of the server that allows users to easily deploy the service through the Jetstream interface, (2) documentation that explains the requirements and procedures of deployment for advanced users. After the grant period, maintenance and continued allocation requests will pass to PI Roberts and the experiments who use the service.
- Kaitai compiler that converts a data description into a library that outputs data into a standard format
  Sustainability plan: this is software that will be complete by the end of the grant. Maintenance of the Kaitai software will pass to PI Amy Roberts. A documentation goal of the project is to ensure that contribution and testing instructions for this software are usable by entry-level scientists.
- Object store
  Sustainability plan: Object storage resources are currently available through the Open Storage Network and Jetstream. Instructions for requesting allocations and instructions for connecting object store into a working PONDD deployment will be hosted on the PONDD documentation site. Responsibility for requesting allocations will pass to experiments who wish to continue using the service.
- Community Rucio instance
  Sustainability plan: The Rucio instance will initially be managed by IRIS-HEP. Over the course of the grant PI Roberts' group will increasingly manage this instance and by the end of the grant it will be deployed on Jetstream. Delivery to the community will consist of (1) a package of the server that allows users to easily deploy the service through the Jetstream interface and (2) documentation that explains the requirements and procedures of deployment for advanced users. After the grant period, maintenance and continued allocation requests will pass to PI Roberts and the experiments who use the service.
- ServiceX-compatible router code
  Sustainability plan: this is software that will be complete by the end of the grant. Maintenance of this code will pass to the collaborations who use it.
- End-user analysis environments, running on XSEDE Jetstream
  Sustainability plan: JupyterHub instances connected to PONDD will require continued allocation requests. Responsibility for submitting allocation requests will be the responsibility of participating experiments both during and after the grant period. To ensure that this knowledge remains accessible, documenting the process (with links to XSEDE instructions wherever possible) will be high priority.
- Kaitai-compatible `yt` modules and data format library
  Sustainability plan: `yt` has reached a state of peer-production; this particular deliverable will reduce the

overall maintenance burden and the surface area of interaction for new researchers. We anticipate that this will in itself improve the sustainability of the project.

4.2 Sustainability

The sustainability of software projects is an active area of both research and experimentation in the scientific community, and Co-PI Turk has been involved in a number of the efforts to both understand and foster sustainability; this includes understanding the impact of collaboration on software development and sustainability [47], [48], prescriptive methods for software engineering to encourage sustainable development [49], [50], and collaborative knowledge sharing through workshops [51]–[55]. SGCI also offers workshops focused on developing sustainability plans for science gateways ("Focus Week"). The PIs intend to make full use of this resource and will be attending this year's workshop.

We have identified **three** aspects of sustainability, each of which we have identified plans to address. Firstly, the sustainability of the developed systems (Kaitai Struct, `yt`) in this project; secondly, the sustainability of the actual user-facing deployments of the services developed in this project; and finally, the sustainability of the scientific community using PONDD.

Sustainability of our Developments This proposal combines multiple software packages into a framework that can deliver data from any experiment to analyzers. An advantage of this approach is that these software packages already have strong community support. This project will further increase their user base and will focus on contributing to both end-user and developer documentation to further strengthen these communities.

The ServiceX and Awkward1.0 packages are all officially supported by the recently-funded IRIS-HEP grant, which aims to develop the cyberinfrastructure needed for handling the PB-scale flood of data needed for the next generation of high-energy physics discovery. Likewise, Rucio enjoys significant support throughout the high-energy physics community and is part of the IRIS-HEP Scalable Systems Laboratory, which is funded to develop scalable, reusable, and reproducible systems for managing large-scale data.

The other two software projects, `yt` and Kaitai Struct, both have active user and developer communities. `yt` has recently received funding to expand its user base to Meteorology, Seismology, Nuclear Engineering, Plasma Simulations, Observational Astronomy, and Hydrology. Using `yt` in PONDD will increase its already-large user base and contribute an easily-deployed data delivery system for scientists who need to combine data from multiple sources. Kaitai Struct is a GPL-licensed, open source project that has sustained an active community for more than five years. We will work alongside both the `yt` and Kaitai Struct communities extensively throughout the first two years of the grant period.

Sustainability of our Services IRIS-HEP will support the development and initial deployment of the services needed for the PONDD infrastructure under their Sustainable Systems Lab banner. Because the Sustainable Systems Lab focuses on prototyping but not on delivering production services, a UC Denver postdoc will help with these deployments and learn about these services. Once the system meets its requirements, the UC Denver postdoc will work to deploy these services on XSEDE Jetsream cloud resources.

PI Roberts has experience deploying prototype and production services on Jetstream. She has led the SuperCDMS outreach and analysis JupyterHub deployment on XSEDE. Although Jetstream cloud orchestration is still early in its development, the Extended Collaboration Support Services (ECSS) provided by XSEDE have resulted in a successful deployment that, while originally designed as a full-featured outreach platform, are increasingly used for full SuperCDMS analyses.

The XSEDE allocations needed for deploying PONDD will be requested during the second year of the grant period. XSEDE will provide feedback and suggestions as we develop these allocation requests, improve our probability of success. We will also request ECSS staff time via this mechanism. In addition, Jetstream

commits to working with our team to develop a containerized packaging of the software to make it available through the Jetstream packages interface. This will ensure that collaborations interested in using this service can deploy them with minimal external support. PONDD documentation will clearly identify what allocations are necessary for collaborations wishing to set up this infrastructure and we will work with XSEDE to test the deployment documentation.

Finally, if the infrastructure proves useful for analysis within the SuperCDMS collaboration, PI Roberts will assume responsibility for maintaining a PONDD service for collaboration use. If the larger dark matter community has a desire for this service to continue, we will consider a frameworks grant that requests dedicated space for some of the PONDD resource needs such as an Open Storage Network pod.

Self-sustaining analysis communities Building self-sustaining analysis communities will take efforts beyond the scope of this grant. However, four aspects of the proposed work will support their growth:

Experiment engagement fellowships will be available to undergraduate students, graduate students, and postdocs. Funding is available in the second two years of the grant to help early-career scientists learn about the PONDD infrastructure, with the intent of preparing them to perform an analysis of their experiment's data using the system. We will continue to provide support to these fellows after their training period.

A Community Engagement Board will provide a more formal method for experiments to shape the requirements of PONDD. While there are many informal methods for interested scientists and experiments to become involved in the project, experiments interested in meeting some or all of their ongoing analysis needs with PONDD, or who are interested in becoming major contributors to parts of the system, can participate more formally through blueprint meetings.

Transparency and discoverability of meetings, priorities, and development efforts will ensure that community members who wish to become involved can find clear paths to do so. We will follow the combined examples of IRIS-HEP and `yt` here; IRIS-HEP's web pages include easy-to-find contact information and a public calendar of meetings. `yt` web pages provide a clear list of different ways to get in contact with the project and their software documentation includes clear developer instructions.

Clear community standards will be set through a code of conduct and enforced by the PIs and senior personnel. Co-PI Turk has built `yt` into a supportive and welcoming community and has experience setting a precedent for positive, professional interactions.

4.3 Metrics

We have identified several concrete metrics that we will use to measure the impact of the project; while these are imperfect, we have attempted to select those that are the most direct proxies and which provide the most insight useful for planning and strategic direction during the course of the project. We have broken these up into those metrics that reflect the impact of the software, the impact of the service through its delivered data, and the scientific results that are produced from the delivered data.

Software Impact Studying the impact of software on the research process is not only under active investigation by researchers in socio-technical studies, but is in fact an oft-featured topic at meetings for CSSI Principal Investigators. There are numerous methods for measuring the imapct of software [56], [57, e.g.] through formal methods. We will utilize measures such as citation to released software products, but we will also provide indirect utilization metrics. These will include commonly-used metrics such as the number of "stars" on GitHub, the number of issues and pull requests issued from funded and non-funded individuals, engagement on relevant mailing lists and discussion forums (such as Slack, Gitter, and so forth as appropriate), and the number of downloads of our produced software. We also note that these metrics, especially downloads, may be inflated by rapid burts of discussion or downloads to continuous integration services, and we will attempt to mitigate this in our reporting. In addition to these metrics, we will also provide utiliza-

tion metrics for the documentation, as measured (in accordance with the privacy policies of our respective universities) by visits to our documentation, tutorials, and training materials.

Service Impact Measuring the impact of the deployed service is somewhat more straightforward, as every individual that utilizes the service interacts with it directly to do so. We will measure the network traffic, both egress and ingress, to our platform. Additionally, in accordance with the privacy policies established by the platform we utilize, we will identify the means of access, and use that to understand the impact of the capabilities developed in this project.

Scientific Impact Measurement of the impact of our development on the final output science products will be provided in two distinct forms. The first will be through *direct* citations of our work, both in acknowledgments and in the list of citations proper. This will provide a measurement of the first-order impact of the work, where it was instrumental to the process of discovery. We will additionally provide "second-order" utilization of our work; this metric, while considerably less direct and prone to inflation, will help to quantify the impact of the discoveries *enabled* by our work. This so-called "second-order" utilization will be a collection of citations *to* papers that directly utilize our work. This will help us to quantify the longer-term impact, as we anticipate that the network effect of our project will result in greater connections between different research groups.

Results from prior NSF support

5.1 Amy Roberts (UC Denver PI)

NSF 1809769: *Collaborative Research: The SuperCDMS SNOLAB Experiment.* **Amount:** $340,000 (UC Denver), $12,000 (Co-PI Roberts). **Period:** 8/15/2018–7/31/2021. **Products:** There are not yet any related publications. However, the PI's group has made substantial contributions to the analysis software. An undergraduate from the PI's lab has worked closely with SLAC computing division to successfully deploy the SuperCDMS analysis environment via a web interface. This provides an unprecedented ease of access to collaboration data and has made it possible for test facilities working on crucial R&D and calibration efforts to efficiently analyze their data. **Intellectual Merit:** This grant supports students and scientists working on an experiment addressing one of the most fundamental problems of modern science: the nature of dark matter. The SuperCDMS-SNOLAB experiment will achieve world-leading sensitivity for dark matter searches in the 1–10 GeV/c2 mass range. **Broader Impacts:** The SuperCDMS experimental and R&D efforts advance phonon-mediated detectors, which have already found many applications in cosmology, astronomy, and industry. Co-PI Roberts supports these efforts through experimental and software expertise.

5.2 Matthew Turk (UIUC PI)

NSF 1663914: *SI2-SSI: Inquiry-Focused Volumetric Data Analysis Across Scientific Domains: Sustaining and Expanding the yt Community*, **Amount:** $1,061,721 **Period:** 10/1/2017-9/30/2022, **Products:** `yt` source code **Intellectual Merit:** The method paper for `yt` has been cited over 800 times and the source code has been used to dramatically advance the speed and detail of research in astrophysical sciences; as part of this grant, `yt` has been expanded to other domains, specifically geodynamics and weather, as well as enhanced to include more advanced analysis mechanisms. **Broader Impacts:** The developments from this grant have materially contributed to analysis results and visualizations used in numerous astrophysical papers, as well as those in the geosciences and weather sciences. `yt` has been used in the development of planetarium shows for education and public outreach.

Readability Note: References for NSF proposals within particle physics are allowed an exceptions: longer author lists may be shortened with 'et al. This is due to the author lists being 15-30 pages and containing thousands of names. Otherwise, the reference list would be unreadable and would be hundreds of pages.

References

[1] URL: https://github.com/ssl-hep/ServiceX.

[2] P. Cushman et al. *Snowmass CF1 Summary: WIMP Dark Matter Direct Detection*. https://arxiv.org/abs/1310.8327.pdf. 2013. arXiv: 1310.8327 [hep-ex].

[3] C. Amole, M. Ardid, I. J. Arnquist, et al. "Dark matter search results from the complete exposure of the PICO-60 C_3F_8 bubble chamber". In: *Phys. Rev. D* 100 (2 July 2019), p. 022001. DOI: 10.1103/PhysRevD.100.022001. URL: https://link.aps.org/doi/10.1103/PhysRevD.100.022001.

[4] R. Agnese et al. "Projected sensitivity of the SuperCDMS SNOLAB experiment". In: *Physical Review D* 95.8 (2017), p. 082002.

[5] Davide Castelvecchi. "Beguiling dark-matter signal persists 20 years on". en. In: *Nature* 556.7699 (Apr. 2018), pp. 13–14. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/d41586-018-03991-y.

[6] M. J. Turk, B. D. Smith, J. S. Oishi, et al. "yt: A Multi-code Analysis Toolkit for Astrophysical Simulation Data". In: *ApJS* 192, 9 (Jan. 2011), p. 9. DOI: 10.1088/0067-0049/192/1/9. arXiv: 1011.3514 [astro-ph.IM].

[7] Fernando Pérez and Brian E. Granger. "IPython: a System for Interactive Scientific Computing". In: *Computing in Science and Engineering* 9.3 (May 2007), pp. 21–29. ISSN: 1521-9615. DOI: 10.1109/MCSE.2007.53. URL: https://ipython.org.

[8] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, et al. "Jupyter Notebooks-a publishing format for reproducible computational workflows." In: *ELPUB*. 2016, pp. 87–90.

[9] Madicken Munk and Matthew J. Turk. "widgyts: Custom Jupyter Widgets for Interactive Data Exploration with yt". In: *Journal of Open Source Software* 5.45 (2020), p. 1774. DOI: 10.21105/joss.01774. URL: https://doi.org/10.21105/joss.01774.

[10] Christopher Havlin, Holtzman Benjamin, Kacper Kowalik, et al. *3D volume rendering of geophysical data using the yt platform*. May 2020. DOI: 10.5072/zenodo.538947. URL: https://doi.org/10.5072/zenodo.538947.

[11] Corey Brummel-Smith, Greg Bryan, Iryna Butsky, et al. "ENZO: An Adaptive Mesh Refinement Code for Astrophysics (Version 2.6)". In: *The Journal of Open Source Software* 4 (Oct. 2019).

[12] G L Bryan, M L Norman, B W O'Shea, et al. "Enzo: An adaptive mesh refinement code for astrophysics". In: *Astrophys. J.* (2014).

[13] B. Fryxell, K. Olson, P. Ricker, et al. "FLASH: An Adaptive Mesh Hydrodynamics Code for Modeling Astrophysical Thermonuclear Flashes". In: *ApJS* 131.1 (Nov. 2000), pp. 273–334. DOI: 10.1086/317361.

[14] H Y Schive, J A ZuHone, N J Goldbaum, et al. "gamer-2: a GPU-accelerated adaptive mesh refinement code – accuracy, performance, and scalability". In: *Mon. Not. R. Astron. Soc.* (2018).

[15] Quincey Koziol and Dana Robinson. *HDF5*. en. 2018. DOI: 10.11578/DC.20180330.1. URL: https://www.osti.gov/doecode/biblio/9801.

[16] Andrey V. Kravtsov, Anatoly A. Klypin, and Alexei M. Khokhlov. "Adaptive Refinement Tree: A New High-Resolution N-Body Code for Cosmological Simulations". In: *ApJS* 111.1 (July 1997), pp. 73–94. DOI: 10.1086/313015. arXiv: astro-ph/9701195 [astro-ph].

[17] R. Teyssier. "Cosmological hydrodynamics with adaptive mesh refinement. A new high resolution code called RAMSES". In: *A&A* 385 (Apr. 2002), pp. 337–364. DOI: 10.1051/0004-6361:20011817. arXiv: astro-ph/0111367 [astro-ph].

[18] Samuel W. Skillman, Michael S. Warren, Matthew J. Turk, et al. "Dark Sky Simulations: Early Data Release". In: *arXiv e-prints*, arXiv:1407.2600 (July 2014), arXiv:1407.2600. arXiv: 1407.2600 [astro-ph.CO].

[19] Volker Springel. "The cosmological simulation code GADGET-2". In: *MNRAS* 364.4 (Dec. 2005), pp. 1105–1134. DOI: 10.1111/j.1365-2966.2005.09655.x. arXiv: astro-ph/0505010 [astro-ph].

[20] Mark R. Krumholz, Richard I. Klein, Christopher F. McKee, et al. "Equations and Algorithms for Mixed-frame Flux-limited Diffusion Radiation Hydrodynamics". In: *ApJ* 667.1 (Sept. 2007), pp. 626–643. DOI: 10.1086/520791. arXiv: astro-ph/0611003 [astro-ph].

[21] Charles R Harris, K Jarrod Millman, Stéfan J van der Walt, et al. "Array programming with NumPy". In: *Nature* 585.7825 (Sept. 2020), pp. 357–362.

[22] S. Hoyer and J. Hamman. "xarray: N-D labeled arrays and datasets in Python". In: *Journal of Open Research Software* 5.1 (2017). DOI: 10.5334/jors.148. URL: http://doi.org/10.5334/jors.148.

[23] Dask Development Team. *Dask: Library for dynamic task scheduling*. 2016. URL: https://dask.org.

[24] RAPIDS Development Team. *RAPIDS: Collection of Libraries for End to End GPU Data Science*. 2018. URL: https://rapids.ai.

[25] James Bradbury, Roy Frostig, Peter Hawkins, et al. *JAX: composable transformations of Python+NumPy programs*. Version 0.1.55. 2018. URL: http://github.com/google/jax.

[26] Ž. Ivezić, S. M. Kahn, J. A. Tyson, et al. "LSST: From Science Drivers to Reference Design and Anticipated Data Products". In: *ApJ* 873, 111 (Mar. 2019), p. 111. DOI: 10.3847/1538-4357/ab042c. arXiv: 0805.2366.

[27] Peter Behroozi, Risa H Wechsler, Andrew P Hearin, et al. "UniverseMachine: The correlation between galaxy growth and dark matter halo assembly from z = 0-10". In: *Monthly Notices of the Royal Astronomical Society* 488.3 (May 2019), pp. 3143–3194. ISSN: 0035-8711. DOI: 10.1093/mnras/stz1182. eprint: https://academic.oup.com/mnras/article-pdf/488/3/3143/29016136/stz1182.pdf. URL: https://doi.org/10.1093/mnras/stz1182.

[28] Clayton Strawn, Santi Roca-Fàbrega, Nir Mandelker, et al. "OVI Traces Photoionized Streams With Collisionally Ionized Boundaries in Cosmological Simulations of $z \sim 1$ Massive Galaxies". In: *arXiv e-prints*, arXiv:2008.11863 (Aug. 2020), arXiv:2008.11863. arXiv: 2008.11863 [astro-ph.GA].

[29] Event Horizon Telescope Collaboration, Kazunori Akiyama, Antxon Alberdi, et al. "First M87 Event Horizon Telescope Results. V. Physical Origin of the Asymmetric Ring". In: *ApJL* 875.1, L5 (Apr. 2019), p. L5. DOI: 10.3847/2041-8213/ab0f43. arXiv: 1906.11242 [astro-ph.GA].

[30] P. Bryan Heidorn, Gretchen R. Stahlman, and Julie Steffen. "Astrolabe: Curating, Linking, and Computing Astronomy's Dark Data". In: *ApJS* 236.1, 3 (May 2018), p. 3. DOI: 10.3847/1538-4365/aab77e. arXiv: 1802.03629 [astro-ph.IM].

[31] A. Brazier, W. Anderson, P. Brady, et al. "SCIMMA: A Framework for Data-Intensive Discovery in Multimessenger Astrophysics". In: *American Astronomical Society Meeting Abstracts #235*. Vol. 235. American Astronomical Society Meeting Abstracts. Jan. 2020, p. 107.03.

[32] URL: https://frictionlessdata.io/.

[33] URL: https://github.com/intake/intake.

[34] *Vision and Change in Undergradaute Biology Education: A Call to Action*. http://www.visionandchange.org/. Washington, DC, 2011.

[35] National Academies of Sciences, Engineering, and Medicine. *Data Science for Undergraduates: Opportunities and Options*. Washington, DC: The National Academies Press, 2018. ISBN: 978-0-309-47559-4. DOI: 10.17226/25104. URL: https://www.nap.edu/catalog/25104/data-science-for-undergraduates-opportunities-and-options.

[36] Joint Task Force on Undergraduate Physics Programs. *Phys21: Preparing Physics Students for 21st-Century Careers*. https://www.compadre.org/jtupp/report.cfm. Washington, DC, 2016.

[37] S. Hurtado, M. K. Eagan, M. C. Tran, et al. ""We Do Science Here": Underrepresented Students' Interactions with Faculty in Different College Contexts". In: *The Journal of social issues* 67 (3 2011), pp. 553–579. DOI: doi:10.1111/j.1540-4560.2011.01714.x.

[38] Martin Barisits, Thomas Beermann, Frank Berghaus, et al. "Rucio: Scientific Data Management". In: *Computing and Software for Big Science* 3.1 (Aug. 2019), p. 11. ISSN: 2510-2044. DOI: 10.1007/s41781-019-0026-3.

[39] Hank Childs, Eric Brugger, Brad Whitlock, et al. "VisIt: An End-User Tool For Visualizing and Analyzing Very Large Data". In: *High Performance Visualization–Enabling Extreme-Scale Scientific Insight*. Oct. 2012, pp. 357–372.

[40] Utkarsh Ayachit. *The ParaView Guide: A Parallel Visualization Application*. USA: Kitware, Inc., 2015. ISBN: 9781930934306.

[41] P. Ramachandran and G. Varoquaux. "Mayavi: 3D Visualization of Scientific Data". In: *Computing in Science & Engineering* 13.2 (2011), pp. 40–51. ISSN: 1521-9615.

[42] D. Nelson, A. Pillepich, S. Genel, et al. "The illustris simulation: Public data release". In: *Astronomy and Computing* 13 (Nov. 2015), pp. 12–37. DOI: 10.1016/j.ascom.2015.09.003. arXiv: 1504.00362 [astro-ph.CO].

[43] M. Taghizadeh-Popp, J. W. Kim, G. Lemson, et al. "SciServer: A science platform for astronomy and beyond". In: *Astronomy and Computing* 33, 100412 (Oct. 2020), p. 100412. DOI: 10.1016/j.ascom.2020.100412. arXiv: 2001.08619 [astro-ph.IM].

[44] URL: https://www.mkdocs.org/.

[45] URL: https://pages.github.com/.

[46] URL: https://semver.org/.

[47] Matthew J Turk. "Scaling a code in the human dimension". In: *Proceedings of the Conference on Extreme Science and Engineering Discovery Environment: Gateway to Discovery*. ACM. 2013, p. 69.

[48] Matthew Turk. "Fostering Collaborative Computational Science". In: *Computing in Science & Engineering* 16.2 (2014), pp. 68–71.

[49] Alice Allen, Cecilia Aragon, Christoph Becker, et al. "Engineering Academic Software (Dagstuhl Perspectives Workshop 16252)". In: *Dagstuhl Manifestos* 6.1 (2017). Ed. by Alice Allen et al., pp. 1–20. ISSN: 2193-2433. DOI: 10.4230/DagMan.6.1.1. URL: http://drops.dagstuhl.de/opus/volltexte/2017/7146.

[50] A Dubey, MJ Turk, and BW O'Shea. "The impact of community software in astrophysics". In: *Joint 11th World Congress on Computational Mechanics, WCCM 2014, the 5th European Conference on Computational Mechanics, ECCM 2014 and the 6th European Conference on Computational Fluid Dynamics, ECFD 2014*. International Center for Numerical Methods in Engineering. 2014, pp. 1813–1820.

[51] Frank Timmes, RENCI Stan Ahalt, NCSA Matthew Turk, et al. "2015 Software Infrastructure for Sustained Innovation (SI 2) Principal Investigators Workshop". In: (2015).

[52] Frank Timmes, Matthew Turk, Stan Ahalt, et al. *2016 Software Infrastructure for Sustained Innovation (SI2) PI Workshop*. Tech. rep. USA, 2016.

[53] Daniel S Katz, Sou-Cheng T Choi, Hilmar Lapp, et al. "Summary of the first workshop on sustainable software for science: Practice and experiences (WSSSPE1)". In: *arXiv preprint arXiv:1404.7414* (2014).

[54] Daniel S Katz, Gabrielle Allen, Neil Chue Hong, et al. "Second workshop on sustainable software for science: Practice and experiences (wssspe2): Submission, peer-review and sorting process, and results". In: *arXiv preprint arXiv:1411.3464* (2014).

[55] Frank Timmes, Rich Townsend, and Lars Bildsten. “Digital Infrastructure in Astrophysics”. In: (Jan. 2020). arXiv: 2001.02559 [astro-ph.IM].

[56] Douglas Thain, Todd Tannenbaum, and Miron Livny. “How to measure a large open-source distributed system”. In: *Concurrency and Computation: Practice and Experience* 18.15 (2006), pp. 1989–2019. DOI: 10.1002/cpe.1041. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpe.1041. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.1041.

[57] D. S. Katz, N. P. Chue Hong, T. Clark, et al. “Software and Data Citation”. In: *Computing in Science Engineering* 22.2 (2020), pp. 4–7. DOI: 10.1109/MCSE.2020.2969730.