Available online at www.sciencedirect.com

ScienceDirect

Advances in Space Research xxx (xxxx) xxx

ADVANCES IN SPACE RESEARCH
(a COSPAR publication)

www.elsevier.com/locate/asr

The DIARieS ecosystem – A software ecosystem to simplify discovery, implementation, analysis, reproducibility, and sharing of scientific results and environments in Heliophysics

Rebecca Ringuette [a,b,*], Alec Engell [c], Oliver Gerland [d], Ryan M. McGranaghan [b,e], Barbara Thompson [b]

[a] *ADNET Systems Inc, 6720B Rockledge Dr., Suite 504, Bethesda, MD 20817, USA*
[b] *NASA Goddard Space Flight Center, Greenbelt, MD 20769, USA*
[c] *NextGen Federal Systems LLC, 10010 Junction Dr Suite 206-N, Annapolis Junction, MD 20701, USA*
[d] *Ensemble Consultancy, 80 M Street, SE Suite 100, Washington D.C. 20003, USA*
[e] *Atmosphere and Space Technology Research Associates (ASTRA), 282 Century Place, Suite 1000, Louisville, CO 80027, USA*

Received 16 August 2021; received in revised form 2 May 2022; accepted 7 May 2022

Abstract

The infrastructure of the Heliophysics discipline has promising components but with several missing gaps, drastically reducing research and development efficiency. Developing an online discovery and analysis software ecosystem will close several of these gaps. The five main focuses on this ecosystem should be Discovery, Implementation, Analysis, Reproducibility, and Sharing of results (DIARieS). In this paper, we give a detailed description of how the proposed software ecosystem should operate, and point out the large range of possible applications to benefit many disparate groups, such as researchers, operational staff, decision-makers, and educators. The infrastructure components and technological capabilities necessary for its completion are either currently available or in development, making such an ecosystem possible for the first time. One main focus of current infrastructure investments must be to adapt and connect these pieces together into a cohesive whole to increase our research and development efficiency.

Keywords: Infrastructure; Discovery; Implementation; Analysis; Reproducibility; Sharing

1. Introduction

Heliophysics is a relatively new field, established as an integrated discipline less than two decades ago. This new conglomeration of communities still retains some disconnections originating from the initially disjointed fields, impairing the wide range of research desired by the community members. There are tremendous benefits to bridging these gaps, which is now made possible by recent advancements in technology. We propose and describe an online software 'ecosystem', where redundant efforts are minimized, knowledge and capabilities are cumulatively shared, and results are more robust and easily validated. This online ecosystem will support the entire Heliophysics community collectively, with extensions to other communities also possible.

In its current state, the Heliophysics community does not provide sufficient infrastructure to researchers. Data, catalogs, models, software, and hardware remain difficult to discover and utilize in a workflow; especially as a collective capability. Poor citation of these research components

* Corresponding author.
E-mail address: rebecca.ringuette@nasa.gov (R. Ringuette).

https://doi.org/10.1016/j.asr.2022.05.012
0273-1177/Published by Elsevier B.V. on behalf of COSPAR.

Please cite this article as: R. Ringuette, A. Engell, O. Gerland et al., The DIARieS ecosystem – A software ecosystem to simplify discovery, implementation, analysis, reproducibility, and sharing of scientific results and environments in Heliophysics, Advances in Space Research, https://doi.org/10.1016/j.asr.2022.05.012

plagues most journals, negatively impacting careers and scientific progress. Publication results are notoriously difficult to reproduce. Research involving multiple disciplines is difficult at best and generally unattainable for most. Finally, analysis of large, disparate and distributed datasets is typically impossible for smaller institutions. The community needs an online discovery and analysis software ecosystem to address these barriers to progress. The concept we describe, called DIARieS, will form a framework to establish such an ecosystem.

Various online discovery and analysis ecosystems have existed in other fields for more than a decade. For example, the Bloomberg Terminal (Bloomberg Finance, 2021) has effectively addressed similar infrastructure needs in financial markets since 1981. The company website advertises use by approximately 325,000 ‘influential decision makers’, indicating such an ecosystem is a well-accepted solution. Yet another popular example of a similar ecosystem in industry is provided by Palantir Technologies, which advertises custom software stacks to, “integrate [an organization’s] data, … decisions, and … operations into one platform” (Palantir Technologies, 2021). To a lesser extent, the Pangeo software stack addresses a reasonable sample of similar barriers in the geosciences with a free, open-source approach (https://pangeo.io/). The current infrastructure needs in Heliophysics are similar to those addressed by these platforms. Thus it directly follows that the Heliophysics community will similarly benefit from the development of a similar ecosystem capability.

The DIARieS software ecosystem should be freely available to the community online, generally based on open-source software, and, by definition, customizable by any user. The five pillars of the software ecosystem are to simplify Discovery, Implementation, Analysis, Reproducibility, and Sharing of results (DIARieS). Many of the required technological capabilities already exist, including containerization, online programming notebooks, widgets (code-free interactive interface components), and automated software management. A simplistic definition of containerization is the packaging of a piece of software together with all its dependencies, including the operating system, to preserve its functionality (see Merkel, 2014; Kurtzer et al., 2017). We also specifically emphasize widgets here as a way to decrease the user programming language knowledge required to access and operate the more universal aspects of the proposed ecosystem (e.g. software installation, data access and utilization, and common plot formats). Applying this and other technologies to our research workflows will drastically broaden the likely user population and increase research and development efficiency, as opposed to the currently predominant command line syntax and programming language knowledge requirements. In addition to such technologies, a large portion of the infrastructure needed for such an ecosystem also already exists or is in development, such as the Heliophysics Data Portal (HDP) and its services, the Community Coordinated Modeling Center (CCMC) and its software and services, the Python in Heliophysics Community (PyHC) software group, and various analysis software packages.

Despite the existence of all these components, scientists and developers consistently spend a large portion of their time (and thus also funding) searching for available research components (e.g. data, models, and software) or reproducing effort already achieved by others. What remains to be done is to mature and connect these components together into a more useful and collective architecture and in a predominantly language-free environment. This paper proposes a map of how these components can be efficiently matured and connected together to build a user-friendly ecosystem that is useful across occupational boundaries. As a community-developed capability, the ecosystem will connect and unify with other development efforts, providing a stronger infrastructure to the entire Heliophysics community. Strengthening the infrastructure in this way will increase our efficiency, meaning the funded projects will take less time and return more complete and transparent results.

The goal of the current paper and the associated online library resource paper (Ringuette & McGranaghan, 2022) is not to isolate various infrastructure resources for criticism, or ignore those not mentioned, but rather to propose a vision of what these resources could unite to provide the community. These two papers attempt to answer the question: What infrastructure would a researcher or student new to Heliophysics, and possibly also new to programming, want to more easily brainstorm, network, and begin a new research project, even involving multiple disciplines? The vision presented here should be considered one of many possibilities of what the next generation workflow could look like. Indeed, we can envision many variations of the idea proposed, which will appeal to differing sections of the community in different ways. We encourage such creative variations and even entirely different ideas, but also encourage the community to coordinate the development of these new applications and technologies to increase our efficiency. By working together as a community, we can discuss the best way to mature our infrastructure, apply new technologies to our research and development, and more closely coordinate our efforts on these tasks.

Contributions to the development of the ecosystem can come from several portions of the community. Data and model providers, archive staff, researchers and software developers can work together to build and verify quick-look graphics capabilities, clickable data downloads, and sample scripts to read and plot the data, which can also be converted into importable widgets by developers. (See Ringuette & McGranaghan, 2021 for other needed capabilities.) Scientists, software developers and software engineers can work together to build containerized example analysis notebooks to be used as use cases (e.g. executable papers). All of these elements will be imperative for development engineers to build the top-level interface discussed in this paper, which the entire community should give feedback on. Additional ways to contribute are discussed in the summary.

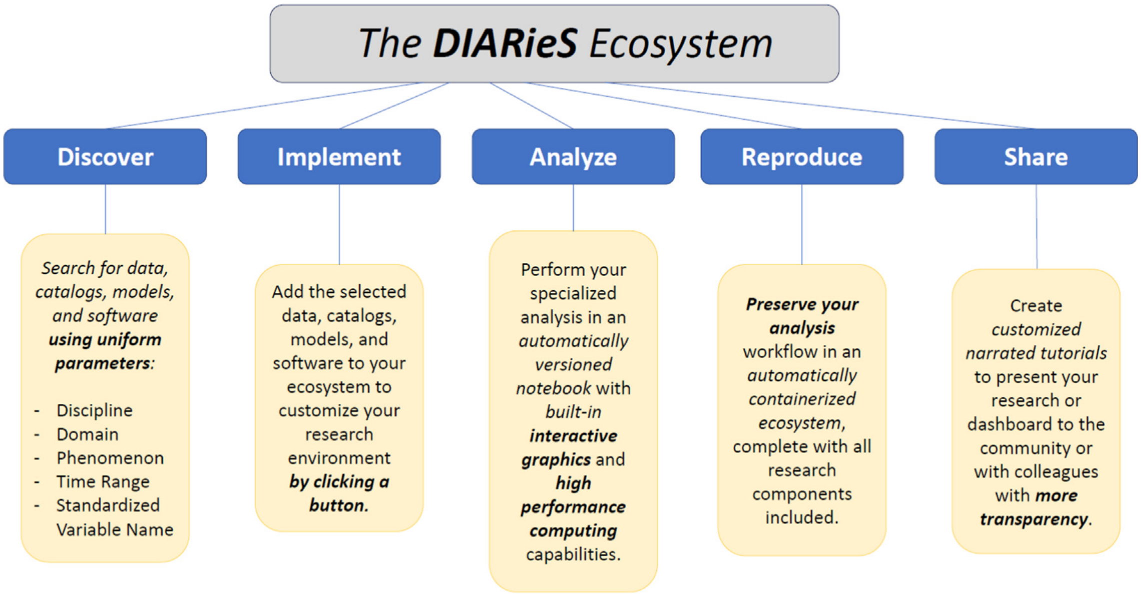
The Heliophysics community faces several problems that existing capabilities cannot solve. Today's relevant data, models, catalogs, and software are difficult to discover, handicapping new researchers and accomplished researchers from other disciplines. Large, disparate, and distributed datasets are challenging to interrogate and analyze, requiring long download times, sometimes daunting hardware infrastructure, and complex data fusion processes, thus restricting researchers at smaller institutions. Proper citation of the research components (data, models, catalogs, software, and hardware) used in a publication is often incomplete thereby inhibiting scientific progress and the career prospects of many. This is particularly problematic for software developers, who leave our field almost as fast as they are trained. Additionally, analysis methods used to produce published results are difficult and often impossible to reproduce, and are often not transparent, lending doubt to many important results. Finally, high performance computing capabilities remain difficult to access, putting smaller groups and minority institutions at a disadvantage. We must work towards reducing these barriers not only to increase our own efficiency, but also to research to attract and retain the next generation of researchers and developers.

2. The main goals of the software ecosystem

These barriers and challenges can be resolved by developing an online discovery and analysis ecosystem called DIARieS (Discovery, Implementation, Analysis, Reproducibility, and Sharing of results) (Fig. 1). The main goals of DIARieS are to provide the following basic capabilities:

- Discover data, models, software, catalogs, and hardware (collectively called research components) easily in one location. This feature will be intrinsically based on a new infrastructure component, an online community library resource, which is described in Ringuette & McGranaghan (2022).
- Easily Implement research components into a customizable analysis ecosystem.
  - o For data, this means adding the data to the user's ecosystem in a commonly used form, such as a NumPy array, XArray object, or Pandas dataframe, by simply clicking a button.
  - o For models, users will be able to calculate values based on point, line, trajectory, and volumetric objects, such as flying a satellite through a model's output, and in turn include the resulting model data into the user's ecosystem, all without writing custom code.
  - o Software will be installed by clicking a button, and all software installation conflicts will be resolved in the background.
  - o Catalogs will be interactive and adjustable with the underlying selected data exportable to the environment for further analysis.
  - o Never miss a citation again with automatically generated citations based on the research components selected.

- Analyze and interact with the research components using services such as Jupyter notebooks and code-free generation of complex interactive plots and media, including a built-in form to request high performance



Search for data, catalogs, models, and software ***using uniform parameters****:*
- Discipline
- Domain
- Phenomenon
- Time Range
- Standardized Variable Name

Add the selected data, catalogs, models, and software to your ecosystem to customize your research environment ***by clicking a button.***

Perform your specialized analysis in an *automatically versioned notebook* with *built-in* ***interactive graphics*** and ***high performance computing*** capabilities.

Preserve your analysis workflow in an *automatically containerized ecosystem*, complete with all research components included.

Create *customized narrated tutorials* to present your research or dashboard to the community or with colleagues with ***more transparency****.*

Fig. 1. Depiction of the five pillars of the DIARieS Ecosystem. Pillars are shown in blue with short descriptions in yellow.

computation time. Previously developed widget-based visualization tools will be available for users to customize their plots without customizing someone else's code. Advanced users can also build dashboards and deploy them to the web with the underlying code abstracted away for community consumption.

- Ensure every step of your process is Reproducible and transparent in an automatically version-controlled, containerized, and customizable environment online.
- Share your research for virtual collaboration and presentations with a 'replay' button, which replays every analysis step saved in the environment, from research component discovery and commented analysis code to final results and plot generation, with optional narrations by the author. We note the sharing feature options should be able to support open-source or private code development by simply changing who has access to the container and their privileges.

These capabilities directly address multiple research tasks, phrased as questions, for which infrastructure is lacking in Heliophysics, and the related identified gaps discussed at the Heliophysics Infrastructure Workshop (2021). Specifically:

- Question: How do I access the data/model/documents I'm interested in?
- Question: What is the appropriate way to process these data to be consistent with a previous publication/science result?
- Question: Where are the data which have properties/p arameters/quantities 'Y'?
- Gap: Citable data and code
- Gap: Discoverability and usability of resources
- Gap: Access to resources
- Gap: Transparency/Reproducibility (e.g. of data products, models, and science results)
- Gap: Linked datasets/research artifacts

In the following section, we outline the vision for this system in more detail. We then compare the capabilities of currently existing ecosystems to this vision. We will find that the majority of the necessary technology exists or is already in development, and only needs to be matured and integrated for a more complete solution.

3. Envisioned capabilities

Heliophysics researchers use a large selection of programming languages, and not all are eager to learn Python despite its current popularity and free availability. Although the online ecosystem we propose will be developed in Python, we prefer a combination of a web-based and widget-based environment sections to increase the accessibility of the ecosystem to a larger audience by reducing the amount of Python knowledge required to operate it. We note that it is typically impossible to perform research and development without some knowledge of a programming language, but also that the current amount needed for a variety of workflows can be greatly reduced by this approach. By web-based section, we mean a section that presents an external webpage to the user, but has additional capabilities made available through the ecosystem. Similarly, a widget-based section is a section that offers users a code-free interface to the desired functionality (e.g. the plotting menus in Microsoft Excel) through widgets typically written in Python. We defend this choice of language by pointing to several inter-language pathways to Python that already exist (e.g. f2py for calling fortran codes from Python, pybind11/cppimport for calling C++ from Python, Python Bridge for calling IDL from Python, and CPython for calling C from Python) (Peterson, 2009, Jakob et al., 2017, Harris Geospatial Solutions Inc., 2020, Behnel et al., 2011), and to the growing popularity of Python in the upcoming generation of researchers who will eventually lead the field (Bobra et al., 2020). Notably, calling other languages from Python is also the solution chosen by popular packages such as SciPy (Virtanen et al., 2020). Choosing Python does not render useless the immense amount of valuable code in other languages, but rather enables the community to continue to build upon a larger portion of it and simultaneously capitalize on the talents of the newer members of the Heliophysics community.

We envision this online ecosystem to include several web-based and widget-based sections with complementary capabilities, building upon already existing systems. The first two sections of the ecosystem will be web-based sections: a data, model, and catalog discovery section and a software discovery section, which will both interlink with the online library resource referred to earlier (yet to be developed) for easier discoverability of any desired research component. The third section will be a notebook-style interface in Python for custom analysis procedures using the selected research components, with access to high performance capabilities. The fourth section will be an interactive plotting widget to enable quick visualization of any variable in the analysis section with the user-selected visualization software. The proper citation list for all components will be automatically logged, based on a flexible metadata system (described in Ringuette & McGranaghan, 2022), with the ability to be exported for publication and reproducibility. The entire environment of each workflow will be automatically containerized for reproducibility, research transparency, and easy deployment to IT infrastructures. Each section is described separately below.

The data, model data, and catalog discovery section and the software discovery section are the two sections the researcher will engage with first. Fig. 2 shows how these two sections will interact with the corresponding library discovery resources, which are each layered on top of already existing services. We leave discussion of the library discovery resource to Ringuette & McGranaghan (2022) and refer the reader to Fig. 1 for a simple depiction of the immediately relevant portions. A sample of a large group of resources, including international options, are included in Fig. 1 for demonstration purposes. (See Fung et al. 2022 for more information on these and other Heliophysics resources.) Each of these two discovery sections in the DIARieS ecosystem will have three tabs, similar in format to tabs of a web page, with complimentary functions. Using the first tab of these sections, researchers will be able to discover and implement any data, model output, catalog, and software components to their environment without any manual installations, understanding custom syntax associated with different archives, or waiting for large data downloads, regardless of whether the component comes from industry, academia, or a government agency. This capability will be possible through a specially designed interface with the online library resource referenced earlier, which will enable users to not only easily discover the desired research components in archive centers across the world, but also implement them in the proposed ecosystem simply by clicking a button and naming the component (the equivalent of 'import software as S' statements in Python and naming a given array or dataframe, but without the syntax).

For large datasets, users will be able to request a containerized version of the dataset for online analysis, if one doesn't already exist, without downloading the data, as is done with EarthCube's "SciUnit" (Valentine et al., 2021). Instead of adding the large dataset to the ecosystem's container, the user will communicate with the dataset's container. In this case, the reproducibility of the workflow will be dependent upon the availability of the version of the dataset available at the later request date. Smaller datasets can be included directly in the container depending on the storage and execution resources allowed for the container. Any software package dependency conflicts will be handled automatically with a conda-like implementation using a tree of containers, possibly a Kubernetes Cluster. Additionally, every data, model, catalog, and software selection will automatically add the relevant citation information to the container which will be available in popular citation formats with a click of a button. For help on each component added, a second tab in each of the two sections will provide links to documentation and examples for each data, model data, catalog, and software component available in the respective discovery tab, with a third tab for easy access to the information on each component already included in the environment. An option to remove the selected component from the container will be available on this third tab via a two-step confirmation process. Completing this process will automatically update the citation information of the container and save a new version of the container.

Fig. 3 demonstrates the envisioned analysis workflow after the research elements are discovered. Once the desired research components are implemented in the ecosystem, users will then analyze the selected data with the selected software in the third section. The first tab in this section

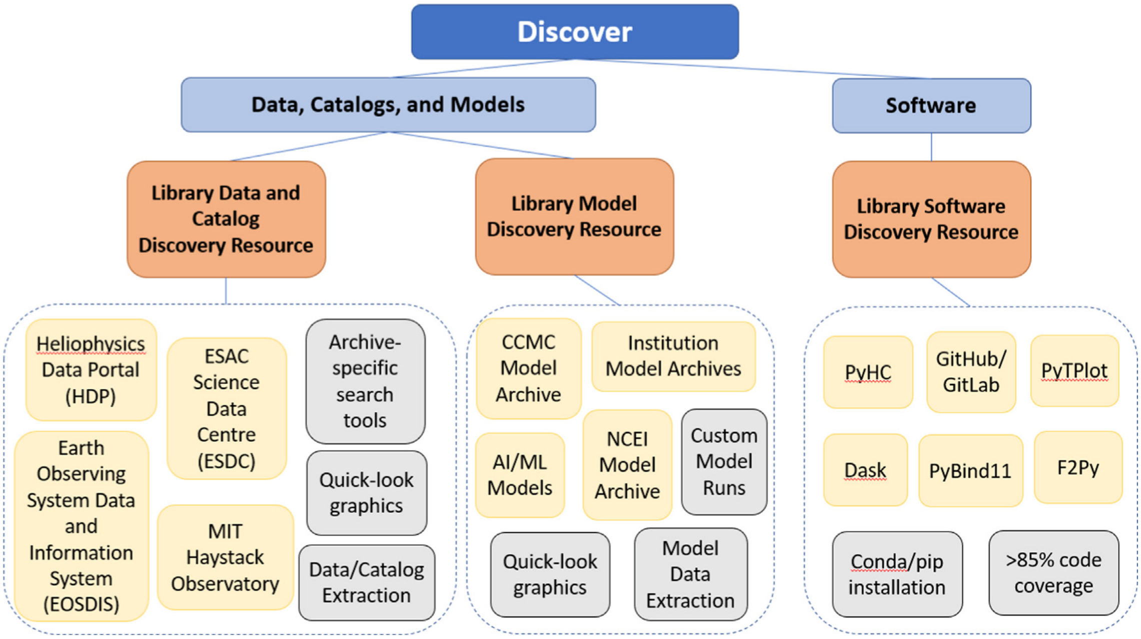

Discover
Data, Catalogs, and Models
Software
Library Data and Catalog Discovery Resource
Library Model Discovery Resource
Library Software Discovery Resource
Heliophysics Data Portal (HDP)
ESAC Science Data Centre (ESDC)
Archive-specific search tools
Earth Observing System Data and Information System (EOSDIS)
MIT Haystack Observatory
Quick-look graphics
Data/Catalog Extraction
CCMC Model Archive
Institution Model Archives
AI/ML Models
NCEI Model Archive
Custom Model Runs
Quick-look graphics
Model Data Extraction
PyHC
GitHub/ GitLab
PyTPlot
Dask
PyBind11
F2Py
Conda/pip installation
>85% code coverage

Fig. 2. Depiction of the operational structure of the two discovery sections of the DIARieS ecosystem. Light blue boxes represent the two discovery sections discussed in the text, orange boxes represent the library discovery resources accessible in each section, which are each a layer on top of the elements in the dashed boxes. Yellow boxes represent a sample of the archives and software elements in each category, while grey boxes are examples of necessary microservices or capabilities provided by each element in the same category. (Repeated from Ringuette & McGranaghan, 2022.).

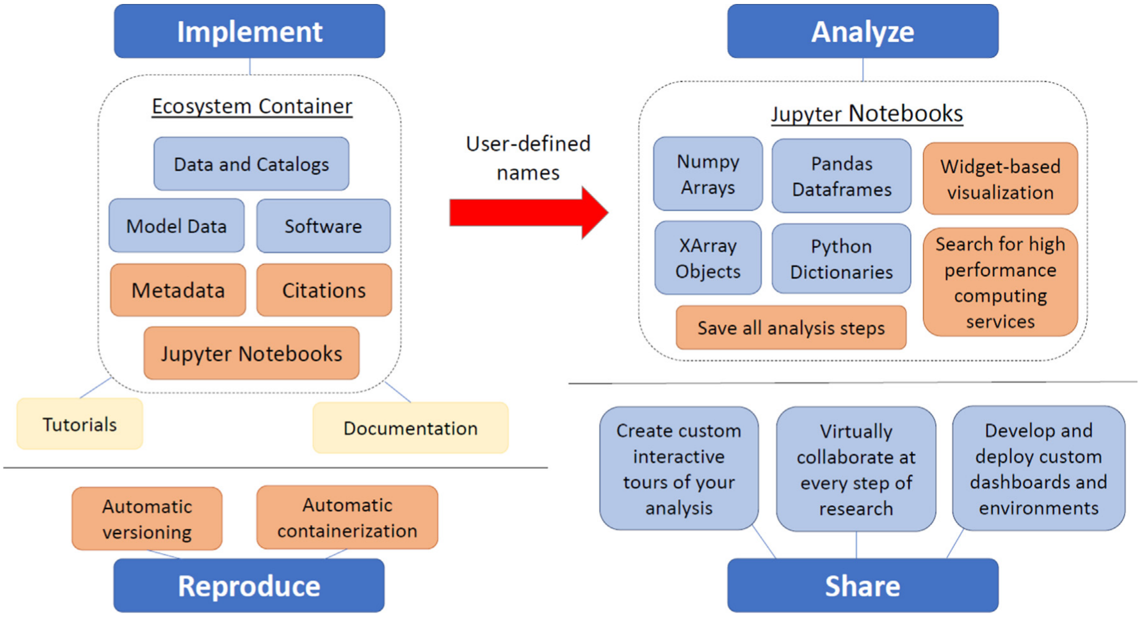


Fig. 3. Diagram of the components and built-in capabilities of the Implement, Analyze, Reproduce, and Share pillars of the DIARieS ecosystem. Light blue boxes represent discovered research components implemented in the ecosystem through the discovery sections, with the exception of the Share pillar. Instead, blue boxes connected to the Share pillar indicate capabilities of the ecosystem. The dashed lines separate elements included in the container, which includes the notebooks, from linked resources accessible from but not included in the container (yellow boxes). Orange boxes show automatically included elements based on current or future capabilities of the underlying ecosystem infrastructure (e.g. various metadata and Jupyter Notebooks).

will be a notebook with all selected research components already included. Four drop-down lists will be located at the top of the notebook, with the included data, model data, software, and catalog names listed in their respective tabs. Clicking on any name in these dropdown lists open the third tab of the respective discovery section for more information on the included object. The second tab of the analysis section will be an interface for the user to search for and request HPC time (high performance computing time) for any computation exceeding the built-in capability of the ecosystem as the users' funding allows. Researchers will be able to seamlessly use packages with conflicting dependencies to perform the desired analysis steps as a result of the software installation tree. Alternatively, users can design their own widget-based analysis environments, or dashboards, for operational or educational applications by choosing from a selection of predesigned widgets, or by requesting the desired functionality. More advanced users can contribute to the community by converting their own discrete analysis steps into widgets available on the ecosystem for possible adoption by others in the community. All analysis steps and changes will be saved automatically to the environment container, which can be version-controlled to preserve previously tested workflows while testing continues and to support virtual collaboration.

During the analysis process, a widget-based plotting window will support generating plots from any variable in the analysis notebook or respective dashboard using their selected plotting software. The plotting window will be hosted in the first tab of this section. Choosing a widget-based plotting interface inherently encourages visualization software developers to develop more intuitive interfaces for their impressive visualization capabilities for use by a larger population (even high school students). For the more seasoned researcher, or those more comfortable with coding, a second tab in the same section provides a notebook-style interface for building upon the software selected, which will also be saved with the container.

Finally, a tutorial section will provide the researcher with documentation specific to the ecosystem software. The first tab of this section will include options to start an interactive tutorial session for a selected example analysis ecosystem or watch a video of the same tutorial e.g. similar to https://helioviewer.org/). On a second tab in the same section, the user will be able to create their own interactive tutorials and videos to display their research process and result. This feature gently encourages users to explain to themselves and others why they made various decisions and assumptions in the research presented, what problems they faced in the analysis and how they overcame it, what important lines of code accomplish and how, and other intricacies of the research or development they consider important. With so many resisting efforts to increase documentation in analysis code, this feature provides an easier way to increase transparency for their future selves and for others trying to understand the result. We envision this to also be useful not only for conference presentations and interactive posters, but also for education, communication with business personnel and politicians, and research efforts involving multiple disciplines. It will also be of use to paper referees and other researchers looking to understand and build upon the result of the analysis.

Supporting project features will be included via clickable buttons along the top of the interface. Each feature offered in the development mode is listed below by the button title followed by a short description of the feature.

- Share: Email a link for their customized ecosystem, with either view, comment, or edit permissions (as with a file on Google drive).
- Publish: Save the current version to a permanent repository of the user's choice and request a DOI for the current version of the user's ecosystem.
- Demo Mode: Turn off all editing permissions and hide any section the user chooses so the user can test the features of the developed environment or dashboard for use by others in operations, education, and communication.
- Citation List: Generate a list of citations in several popular formats to include the proper research component citations in any publication based on the research or development performed in the ecosystem. The user will choose the desired format, then be able to copy the text by clicking a button (similar to that offered by various journals).
- Add Author: Add their contributor's information to the ecosystem.

The published version of the ecosystem will have a complementary set of features. Of the features described, only the 'Publish' and 'Add Author' buttons will not be available in the published version. In addition to these features, the published version will also have a 'Cite Me' button to allow other users to more easily cite the developed ecosystem, and a 'Copy' button for others to copy and then adapt the developed ecosystem for their own purposes in the spirit of open-source development. These complementary capabilities provide functionality similar to what users have come to expect of other technology resources, and simplify citation, transparency, and reproducibility issues.

This unique approach addresses several gaps in the current research workflow. Data, models, software, catalogs, and hardware become more easily discoverable and usable, leading to more impact per component developed, thus increasing the return on those investments. All the research components of a given project will be collected into one place, a customizable, containerized environment. This containerized environment can be preserved upon request with a persistent DOI for future replication and comparison, then linked to the related published paper for simple accessibility. The containerized online workflow inherent to the recommended ecosystem simplifies virtual collaboration, even across domains and disciplines (e.g. Heliophysics, earth science, and data science). Access to high performance computing services will be more equitable, enabling smaller, less-connected research groups to accomplish larger projects. Finally, the automatic inclusion of the proper citations of data, models and software will encourage proper attribution of credit, which is imperative for a vibrant research community and attracting and retaining talent.

One example of how a group of scientists would use the DIARieS ecosystem to host an entire workflow is to assess the accuracy of a variable's forecast performed with a machine learning code trained on ensemble modeling output. The scientist familiar with space weather observational data would use the data discovery interface to find, implement, and filter the desired time intervals of data in the ecosystem. The various model data outputs would be acquired in a similar way through the interface to the model data repositories, linking to data on the cloud if possible. Then, a scientist familiar with machine learning and Kalman filtering would use the software discovery interface to add machine learning tools and Kalman filtering software to the ecosystem, along with any desired plotting software. Once all the research components are gathered, the scientists would work together in Python and any linked languages to perform the Kalman filtering on the model data, form the model ensemble, design and train the machine learning model on the ensemble of model data, and use the trained model to predict the chosen variable's behavior into the future in a shorter time than possible if the models themselves were run. Finally, the scientists would perform the appropriate accuracy analysis and easily create publication quality visualizations with the software installed on the user's ecosystem. At any step in the process, the scientists will be able to virtually collaborate with each other, such as on the particulars of Kalman filtering and the specifics of interpolating model data on various grids, without the other researchers having to repeat the steps on their own computers. At the end of the analysis process, the scientists can use the automatically generated citation list to properly reference all of the research components used, and can use the Demo Mode to form an interactive poster to guide others through the work done, including referees of the corresponding published paper. This level of collaboration is currently difficult at best due to the lack of an online discovery and analysis ecosystem, such as the one described in this work.

The benefits of this ecosystem warrant significant work towards its development in the Heliophysics community. Researchers will use the proposed ecosystem to more easily collaborate on their research with their colleagues worldwide without requiring duplicate time invested into component implementation. Scientists will be able to easily share their completed research with others in the community in a completely transparent way, which will also be automatically reproducible in many cases (e.g. traditional research, data calibration methods, and even software documentation examples). Other scientists will be able to more easily create tagged datasets and catalogs and make them publicly accessible, develop machine learning models, and perform data-to-model comparison science, including model validation. Developers will use this to more efficiently create dashboards for use by operations, decreasing the lag between research and operations. Educators will use this new capability to develop interactive learning environments based on dashboards for students and researchers from other domains. Communicators will use this ecosystem to develop interactive "What If" environments for students and decision-makers to understand how various phenomena impact us on Earth and elsewhere. As a bonus, we expect competing versions of this ecosystem to iteratively increase the capabilities of the software, with various branching versions serving different disciplines and connecting between them.

4. Currently existing or lacking capabilities and technologies

Notably, this ecosystem can be established based on some systems that are currently being developed. Several efforts are underway to lower the barriers to data access and utilization, such as Pysat's data-agnostic interface to a wide range of datasets, HAPI's development of command-line access to observational data, and Kamodo's model-agnostic interface to an expanding sample of Heliophysics models (Stoneback et al., 2018, Weigel et al., 2021, and Pembroke et al., 2022). (See https://heliopython.org/ and Burrell et al., 2018 for more examples.) We envision the discovery of high-performance computing services to be discoverable using the same library resource referenced previously for other discovery applications, but universal accessibility to some services may need to be negotiated, even if cost is passed on to the user. Data and model archives are already working towards using more intuitive search forms and quick-look graphics (e.g. the Heliophysics Data Portal), but this effort needs to be extended to the software and HPC communities, especially the AI/ML software community.

Containerization technology, such as Docker (Merkel, 2014) and Singularity (https://sylabs.io/guides/3.5/user-guide/introduction.html), is gaining traction in the Heliophysics community, as is notebook-style analysis. Interestingly, the combination of the two theoretically enables the ecosystem to be used with any programming language supported by the notebook software (e.g. Jupyter) (Kluyver et al., 2016), not just Python. We leave the development of a similar ecosystem in other languages to those proficient in those languages. In industry, a trend towards widget-based environments is also gaining popularity, which increases accessibility to researchers trained in programming languages other than Python and those new to programming in general (see https://ccmc.gsfc.nasa.gov/iswa/ for an example). Multiple data archives are also working towards containerizing selected large datasets, which is necessary for streamlining their analysis. The building blocks of the envisioned method to handle installation of software with conflicting dependencies also already exists. Anaconda already automatically handles installation of a given software's dependencies through conda based on the dependencies preprogrammed by the software developers (Anaconda Software Distribution, 2020). What remains to be applied is a more advanced version of conda, such as Kubernetes (https://kubernetes.io/), that is able to automatically handle unresolvable conflicts (e.g. package A needs version 1.2 of a given package but package B in the same environment needs version 2.2 of same package) via creation of a cluster of containerized packages. The practice of versioning containers is already used on Jupyter Hub (https://jupyter.org/hub) and DeepNote (https://deepnote.com/), and that technology can be applied to this ecosystem, simplifying virtual collaboration between researchers across the world, including different domains and disparate disciplines.

The development of SPRINTS (Engell et al., 2017), which is akin to Pangeo (Abernathey et al., 2017) but for Heliophysics, is a case in point. At its heart, SPRINTS is built in Python as an online, dashboard-based environment, currently customized in a variety of ways for specific applications in space weather research-to-operations (see Figs. 4 and 5 for examples). Despite this, the components of the system are strikingly similar to many of the features of the online ecosystem proposed. The requested data is already curated and added to the interface, data investigation and some basic analysis can be done on the fly, machine learning software is installed and operating behind the scenes to generate forecasts, and plots of certain variables are either automatically generated or intuitively available through drop-down menus and clickable buttons. What remains to be done to build upon this software is to generalize the development process into a widget-based ecosystem for users to easily customize the resulting interface for their own research, as outlined here in detail. Consequently, this also holds great promise for the research-to-operations and operations-to-research, as multiple variations of operations interfaces can be built by researchers and tested against each other by forecasters, as done in terrestrial weather operations development (e.g. the NOAA Hazardous Weather Testbed yearly workshop https://hwt.nssl.noaa.gov/).

In addition to SPRINTS, the Kamodo software suite and its associated API have validated the use of containerization and functionalization techniques to streamline model deployments as a method to easily visualize and manipulate space weather model output. The Kamodo software suite was first developed at the Community Coordinated Modeling Center (CCMC) to simplify the analysis, visualization, and tooling of space weather models hosted at the CCMC. Kamodo provides a functional programming solution that can not only mitigate current data integration and collaboration challenges within the space weather community, but also enables scientific workflows not possible with any single standard. Kamodo's ability to functionalize existing space weather resources is critical because many analyses and visualization challenges, such

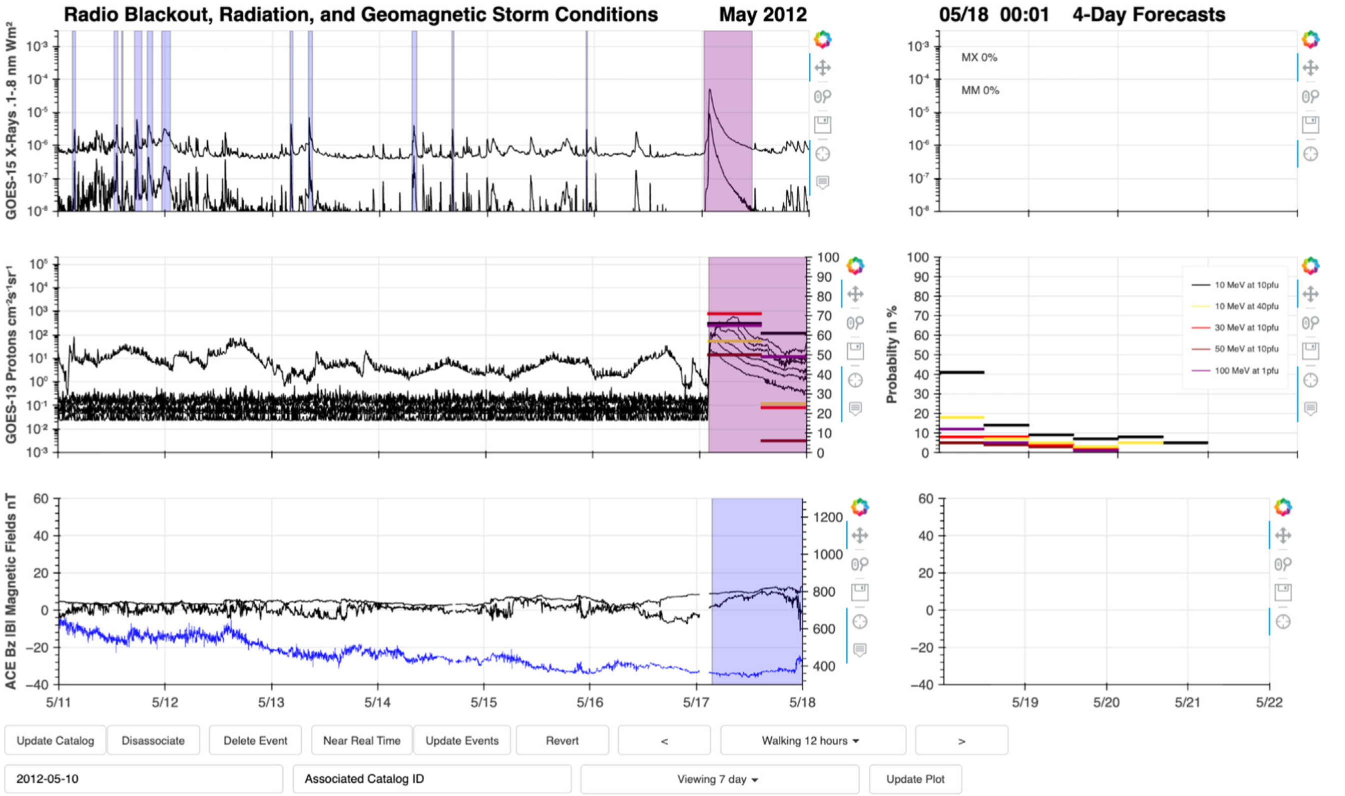


Fig. 4. A screenshot of a real-time SPRINTS dashboard from May 2012. *Left column:* The last seven days of GOES X-ray, GOES proton, and solar wind data. The red highlighted X-Ray flare and SEP (solar energetic proton) events (top and middle plots) show that they are associated with each other in the SPRINTS relational time-series database. The blue highlighted data (top and bottom plots) show flare and solar wind "events" that are automatically detected. *Right column:* The top and bottom panels are built to show SWPC (NOAA/NWS Space Weather Prediction Center) forecast data which was not available at the time in the SPRINTS database for this period. The middle panel shows the post-eruptive forecast based on the SPRINTS machine-learned model for the five Air Force Research Laboratory SEP forecast requirements. *Bottom:* Various buttons enable the user to explore the entire data holdings (e.g., back to 1986 for GOES) and update the event times and associations in the database. The SPRINTS dashboard is highly customizable and the supporting time-series database and APIs can easily be extended to other data.

as data-model comparisons and coordinate transformations, may be framed in terms of function composition. In its current state, Kamodo is delivered via an open-source Python package, a REST API, and a drag-and-drop widget dashboard interface that hosts an increasing number of space weather model output and data resources including Pysat and CORHEL (see Figs. 6 and 7 for examples) (Stoneback et al., 2018, Riley et al., 2012). Each dashboard is a python functional interface that provides interpolated observational data or model output on request. Each isolated docker container hosts the data to be served to the dashboard and handles its own dependencies. Two containers host the 'Kamodofied' scientific resources while one container orchestrates the assimilation of the models.

Other analysis platforms are also in development by various members of the community. We note the efforts of the Frontier Development Lab in developing SpaceML (spaceml.org) and the development of Pangeo (https://pangeo.io/) as partial variative implementations of the idea proposed. These and others are generally based on Jupyter notebooks with various supporting technologies, such as built-in high-performance computing services, pre-built software stacks, and pre-customized dashboards. However, none of those known to us provide all the capabilities described here. Nonetheless, we encourage continued development in this area by the various groups, and recommend these groups coordinate their efforts to focus on differing, complementary capabilities with the goal of piecing these capabilities together into a cohesive ecosystem such as the one described here.

Despite all the exciting technological advances, there are many in the research community that resist adopting them for various reasons. We note the opposite trend in younger scientists, and expect the overall trend to change as their careers progress. Additionally, many contributors continue to neglect documentation of their own research or development steps, costing them and others significant time and effort. Therefore, we recommend the community form groups to research two topics: best practices for documentation of analysis steps, and training development for early-, mid-, late-, and transitioning career researchers and developers. We also point to the described interactive touring capability of the DIARieS software ecosystem as a promising solution to this problem. By researching these topics, we will be prepared to guide the community in the

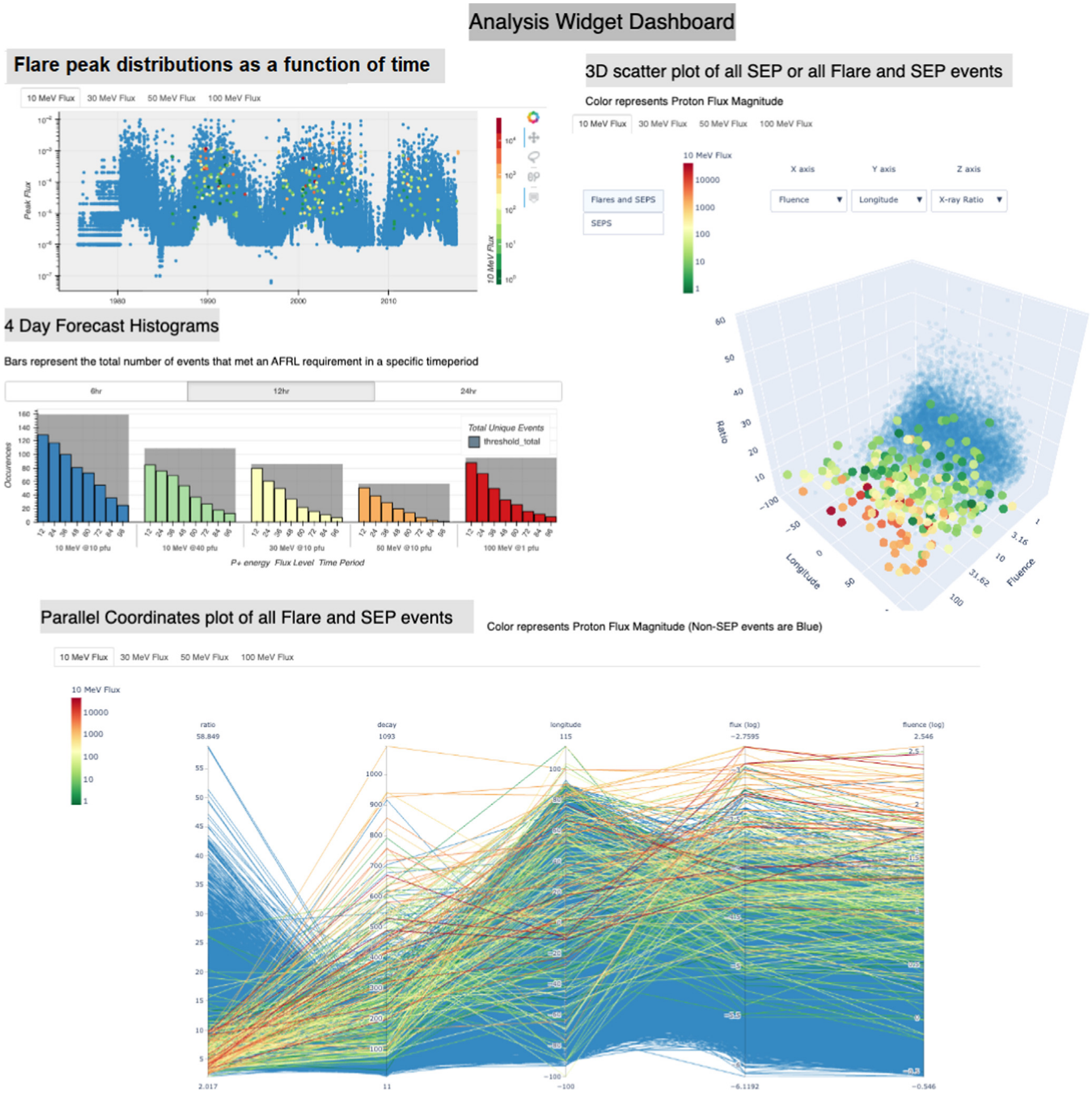

Analysis Widget Dashboard
Flare peak distributions as a function of time
10 MeV Flux
30 MeV Flux
50 MeV Flux
100 MeV Flux
Peak Flux
10 MeV Flux
1980
1990
2000
2010
3D scatter plot of all SEP or all Flare and SEP events
Color represents Proton Flux Magnitude
Flares and SEPS
SEPS
X axis
Y axis
Z axis
Fluence
Longitude
X-ray Ratio
10000
1000
100
10
1
Ratio
Longitude
Fluence
4 Day Forecast Histograms
Bars represent the total number of events that met an AFRL requirement in a specific timeperiod
6hr
12hr
24hr
Occurences
Total Unique Events
threshold_total
10 MeV @10 pfu
10 MeV @40 pfu
30 MeV @10 pfu
50 MeV @10 pfu
100 MeV @1 pfu
P+ energy Flux Level Time Period
Parallel Coordinates plot of all Flare and SEP events
Color represents Proton Flux Magnitude (Non-SEP events are Blue)
ratio
decay
longitude
flux (log)
fluence (log)
58.849
1093
115
−2.7595
2.546
2.017
11
−100
−6.1192
−0.546

Fig. 5. Screenshot of one dashboard variation developed using SPRINTS. This dashboard, developed to support operational space weather forecasting and model development and improve event and associated event catalogs (e.g., SEPs and CMEs associated with solar flares). It incorporates a variety of datasets with python and machine learning functionality to produce the interactive plots and forecasts. Clicking on the widgets and the plots' interactive capabilities gives the user access to a broader range of information to further inform their decisions without any necessary programming.

best way to adapt these modern technologies, including the DIARieS ecosystem proposed, and to improve communication and transparency with others in the community.

5. Summary

The infrastructure available to the Heliophysics research community is currently lacking in several ways. Several important datasets, expensive model results, developed catalogs, and interesting software packages and codes remain unused, simply due to poor discoverability, accessibility, and utility. Poor citation of these research components continues to impair the careers of those who developed or created them, putting them at a disadvantage compared to scientists with more traditional results. Minority and smaller institutions remain disadvantaged in 'big data' and collaborative projects due to lack of infrastructure and accessibility.

These handicaps demand the development of a primarily widget-based online discovery and analysis software

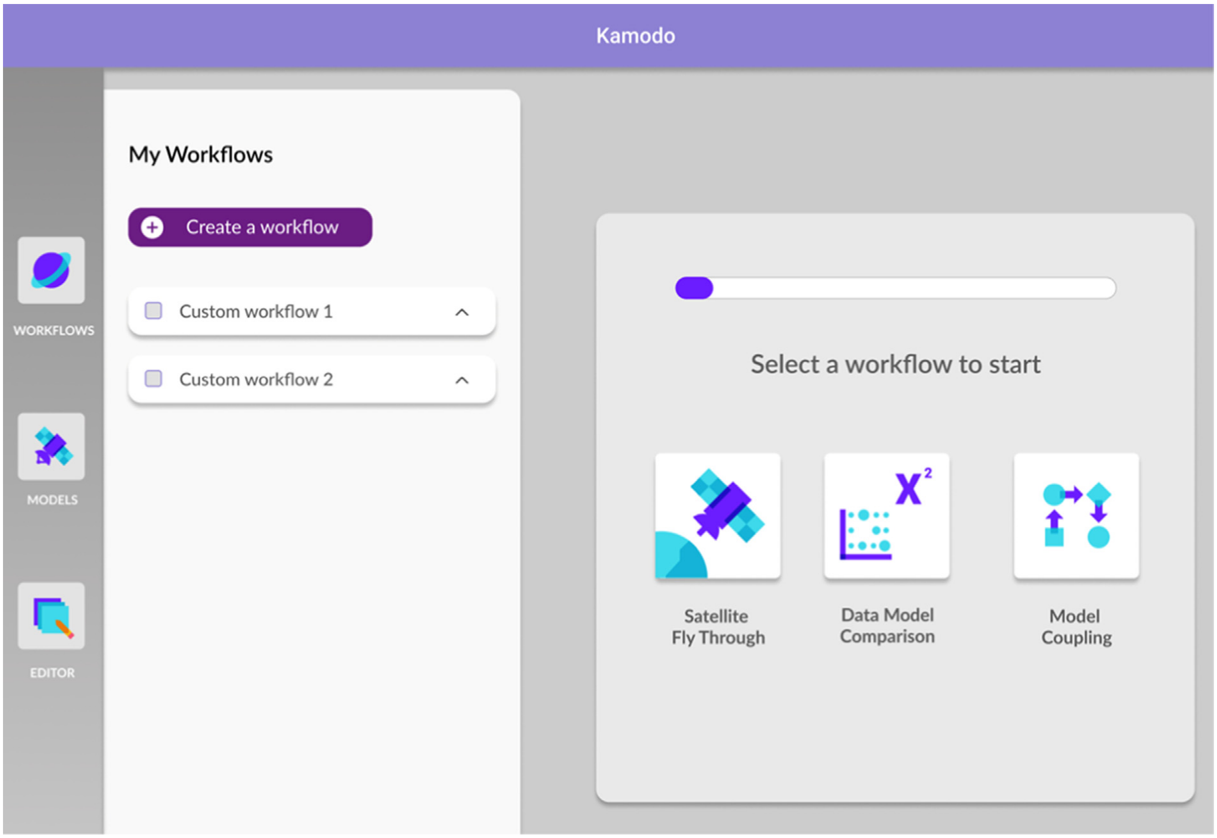


Fig. 6. Screenshot of one of Kamodo's code-free interfaces in development. Users can build their own custom workflows implementing a variety of features available through Kamodo (seen on the far right) and interact with a growing list of models. These basic functionalities are available without any programming needed, but users also have the option to perform additional analysis via the editor option (far left column).

ecosystem. This ecosystem will be built around five sections of complementary capabilities: data, model, catalog, and software discovery and implementation, software discovery and installation, analysis reproducibility and transparency with access to high performance computing services, customizable interactive visualizations, and custom tutorial creation. Other important capabilities include easier dashboard development, automated registration of and generation of a citation list, association of the customized ecosystem with a DOI, and virtual collaboration with other researchers or students, all with the click of a button for most features. Necessarily, this ecosystem will depend on and interconnect with other developing infrastructure components in the field, some of which are already in use. We expect competition between different versions of the ecosystem to spur on development, improving the workflow of all researchers in the field, and likely also in other disciplines. As a result, the next generation of researchers will have a faster, more shareable workflow, yielding greater return on each successful research project.

Some aspects of the envisioned ecosystem will likely take several years of development to achieve, particularly the discovery capability, automatic citations, and other similar features (see Ringuette and McGranaghan, 2022 for more details). Also, the infrastructure challenges associated with big data capabilities are a crux that does not appear to have a near-term solution without significant funding. However, many of the remaining capabilities are becoming possible even now, as described in Section 4, and can be accelerated by community collaboration.

There are several tasks that can be begun immediately. File readers and other similar capabilities should be made readily available for users to easily include the data into their environment with one simple command (or as few as possible). Without these capabilities, all archive users are left to write their own scripts to load and interact with the data, unnecessarily repeating the work of others and particularly increasing the difficulty of interacting with data in unfamiliar fields. At the recent CEDAR 2021 conference, participants expressed their frustration at the large amount of time wasted duplicating these efforts.

Part of the underlying cause of this frustration is a discoverability gap between data and software. This can be alleviated in the immediate future by simply including associations between software packages and datasets (observational and modeled), such as including the satellite and model names as keywords in the software packages the associated data are accessible from (e.g. 'MMS' as a keyword in the SpacePy package metadata and 'CTIPe' as a keyword in Kamodo's metadata) (MMS: Burch et al., 2016, SpacePy: Morley et al., 2011, CTIPe (Coupled Thermosphere Ionosphere Plasmasphere Electrodynamics

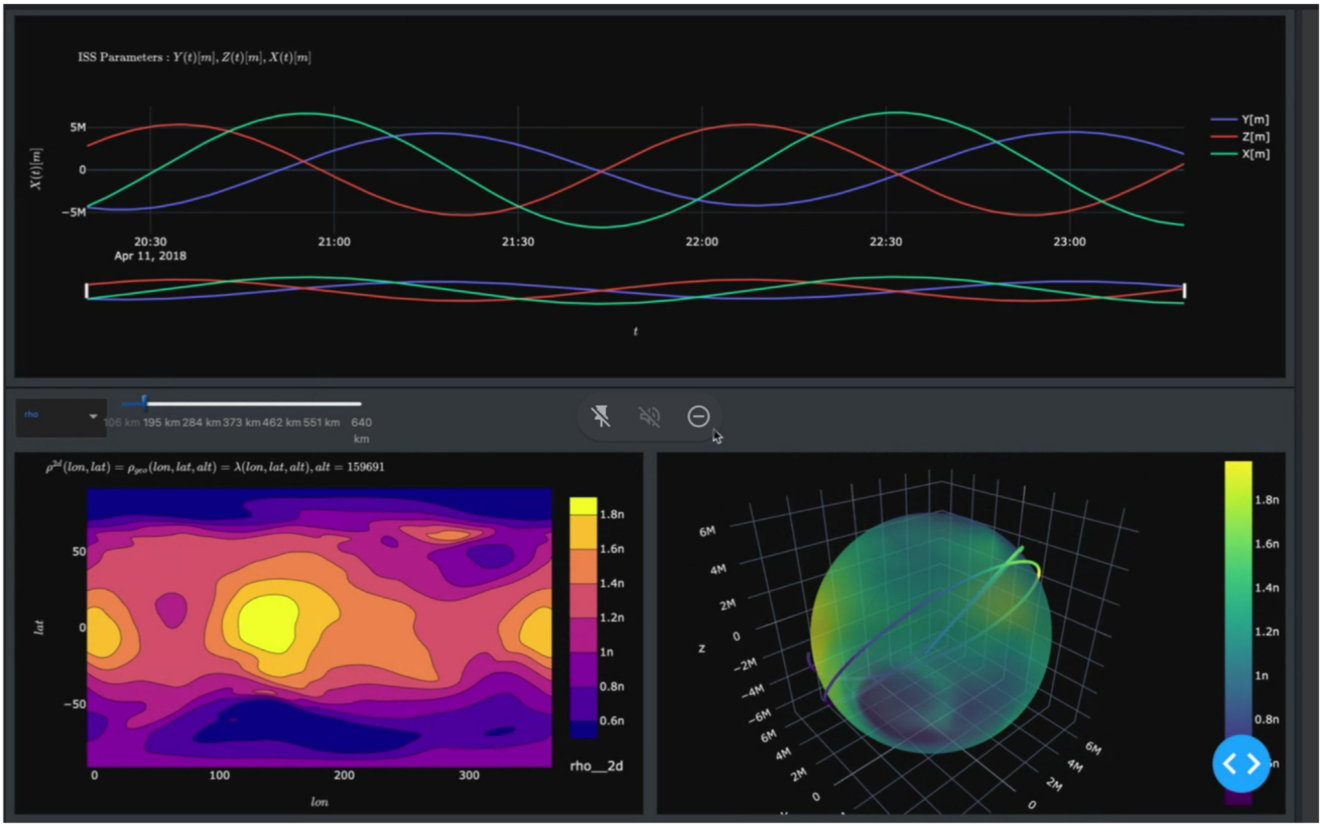


Fig. 7. Example of one of Kamodo's satellite flythrough simulation dashboards in development. Users can select the desired time range, variable, and other parameters using the interactive plot of the satellite trajectory (top panel) and the drop down menu and slider (bottom left). The bottom panels provide interactive displays of the model data chosen by the user.

model): Codrescu et al., 2008, Kamodo: Ringuette et al., 2022). (See the PyHC projects page for additional packages: https://heliopython.org/projects/.) In parallel, data providers should work with currently existing software packages to incorporate their data plotting, calibration, filtering, and other utilization scripts into those packages, thus increasing the long-term sustainability of those scripts and increasing the community's ease of use for those data. In many cases, such collaborations will also decrease the effort required by the data providers as the Heliophysics community's movement towards open data becomes more formalized. As noted earlier, scripts written in other languages can be included in Python software packages by using cross-language interfaces (e.g. f2py for calling fortran codes from Python, pybind11/cppimport for calling C++ from Python, Python Bridge for calling IDL scripts from Python, and CPython for calling C from Python) (Peterson, 2009, Jakob et al., 2017, Harris Geospatial Solutions Inc., 2020, Behnel et al., 2011).

Data, catalog, and model archives need to develop and implement a syntax-free API for all downloadable data, where users can click a button to download the requested data without user log-ins or burdensome syntax. A syntax-free API (or nearly syntax-free), such as the one described here for data access, greatly reduces the complications currently inherent with data implementation. This capability is described in more detail in the associated online library resource paper (Ringuette & McGranaghan, 2022). Some partial examples of this capability exist already and should be improved upon and imitated elsewhere (e.g. the Heliophysics Data Portal: https://heliophysicsdata.gsfc.nasa.gov/websearch/dispatcher). Large software packages can also progress towards the interoperability required in DIARieS by providing containerized versions of their packages online for users to easily include in their work without installing the software themselves. Additional software contributions would include efforts to convert commonly used plotting and analysis scripts into widgets, such as offering pyTplot as a widget-based application as is being done for other packages (see Figs. 4-7). Finally, the software and research communities also need to collaborate to develop containerized example workflows to be available online, which can then be used by ecosystem developers as example use cases in their development efforts.

The online discovery and analysis ecosystem described here, called DIARieS, is one possible workflow the next-generation researcher will use. We anticipate the development of this ecosystem and its components will advance the Heliophysics discipline, in both the private and public sectors. We emphasize that 1) the components of DIARieS can and are being implemented incrementally, and 2) the components are usable on their own, so the full ecosystem does not have to be actualized before it is valuable to the community. With careful guidance and generous encouragement and funding, these advancements can result in closer connections between industry and research, software developers and scientists, and infrastructure developers and users, resulting in more effective use of funding and better preservation and dissemination of the results. Now is the time to develop this next generation research workflow to accelerate Heliophysics understanding, increase reproducibility and transparency, and close the operations-to-research and operation-to-research gaps. Without a community-wide effort, the current possible solutions to the identified gaps in the Heliophysics infrastructure will be difficult to make compatible with each other, delaying research progress and preventing more efficient use of research funding. We call on the community to work together on this goal to launch our field into the future.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

R. R. acknowledges the fall AGU 2020 conference, the 43rd COSPAR Scientific Assembly (2021), the Heliophysics Infrastructure Workshop (2021), and CEDAR 2021 for enlightening interactions and thought-provoking presentations that led to this paper. R. R. also acknowledges helpful conversations with Brian Thomas, Maria Kuznetsova, and Asher Pembroke, and some helpful comments on the manuscript from Lutz Rastaetter. This work was supported by NASA Center for HelioAnalytics funded by NASA ISFM.

References

Abernathey, R., Paul, K., Hamman, J., Rocklin, M., Lepore, C., et al., 2017. Pangeo NSF Earthcube Proposal. https://doi.org/10.6084/m9.figshare.5361094.v1.

Anaconda Software Distribution, 2020. Anaconda Documentation. Anaconda Inc. Retrieved from https://docs.anaconda.com/.

Behnel, S., Bradshaw, R., Citro, C., Dalcin, L., Seljebotn, D.S., Smith, K., 2011. Cython: The best of both worlds. Comput. Sci. Eng. 13 (2), 31–39. https://doi.org/10.1109/MCSE.2010.118.

Bloomberg Finance, L.P., 2021. The Terminal: Bloomberg Professional Services. https://www.bloomberg.com/professional/solution/bloomberg-terminal/. Website accessed Aug 6, 2021.

Bobra, M.G., Mumford, S.J., Hewett, R.J., Christie, S.D., Reardon, K., et al., 2020. A survey of computational tools in solar physics. Sol. Phys. 295, 57. https://doi.org/10.1007/s11207-020-01622-2.

Burch, J.L., Moore, T.E., Torbert, R.B., Giles, B.L., 2016. Magnetospheric Multiscale Overview and Science Objectives. Space Sci. Rev. 199, 5–21. https://doi.org/10.1007/s11214-015-0164-9.

Burrell, A.G., Halford, A., Klenzing, J., Stoneback, R.A., Morley, S.K., et al., 2018. Snakes on a spaceship—An overview of Python in heliophysics. J. Geophys. Res.: Space Phys., 123, 10384– 10402. https://doi.org/10.1029/2018JA025877.

Codrescu, M.V., Fuller-Rowell, T.J., Munteanu, V., Minter, C.F., Millward, G.H., 2008. Validation of the coupled thermosphere ionosphere plasmasphere electrodynamics model: CTIPe-Mass Spectrometer Incoherent Scatter temperature comparison. Space Weather 6, S09005. https://doi.org/10.1029/2007SW000364.

Engell, A.J., Falconer, D.A., Schuh, M., Loomis, J., Bissett, D., 2017. SPRINTS: A framework for solar-driven event forecasting and research. Space Weather 15, 1321–1346. https://doi.org/10.1002/2017SW001660.

Fung, S., Andre, N., Bergatze, L.F., Bouchemit, M., Candey, R.M., et al., 2022. SPASE Metadata as a Heliophysics Science-Enabling Tool. Submitted to Adv. Space Res.

Harris Geospatial Solutions, Inc, 2020. The Python Bridge. https://www.l3harrisgeospatial.com/docs/Python.html. Website accessed Aug 9, 2021.

Jakob, W., Rhinelander, J., Moldovan, D., 2017. pybind11 — Seamless operability between C++11 and Python. Presented at the EuroPython 2017 conference July 9-16, 2017, Rimini, Italy. https://doi.org/10.5446/33723.

Kluyver, T., Ragan-Kelley, B., Perez, F., Granger, B., Bussonnier, M., et al., 2016. Jupyter Notebooks - a publishing format for reproducible computational workflows. Positioning and Power in Academic Publishing: Players, Agents and Agendas, 87-90, IOS Press eBooks, Clifton, VA, USA. http://dx.doi.org/10.3233/978-1-61499-649-1-87.

Kurtzer, G.M., Sochat, V., Bauer, M.W., 2017. Singularity: Scientific containers for mobility of compute. PLoS ONE 12 (5). https://doi.org/10.1371/journal.pone.0177459 e0177459.

Merkel, D., 2014. Docker: lightweight Linux containers for consistent development and deployment. Linux J., 239, 2. https://www.linuxjournal.com/content/docker-lightweight-linux-containers-consistent-development-and-deployment.

Morley, S.K., Welling, D.T., Koller, J., Larsen, B.A., Henderson, M.G., et al., 2011. SpacePy - A Python-based Library of Tools for the Space Sciences. https://doi.org/10.25080/Majora-92bf1922-00c.

Palantir Technologies, 2021. Palantir. https://www.palantir.com/. Website accessed Aug 6, 2021.

Pembroke, A., De Zeeuw, D., Rastaetter, L., Ringuette, R., Gerland, O., et al., 2022. Kamodo: A functional api for space weather models and data. JOSS, submitted.

Peterson, P., 2009. F2PY: a tool for connecting Fortran and Python programs. Int. J. Comput. Sci. Eng. 4 (4), 296–305. https://doi.org/10.1504/IJCSE.2009.029165.

Riley, P., Linker, J.A., Lionello, R., Mikić, Z., 2012. Corotating interaction regions during the recent solar minimum: the power and limitations of global MHD modeling. J. Atmos. Sol.-Terr. Phys. 83, 1–10. https://doi.org/10.1016/j.jastp.2011.12.013.

Ringuette, R., McGranaghan, R.M., 2022. The LIKED Resource - A LIbrary KnowledgE and Discovery online resource for discovering and implementing knowledge, data, and infrastructure resources. Submitted to Adv. in Space Res.

Ringuette, R., Rastaetter, L., De Zeeuw, D., Pembroke, A., 2022. Simplifying Model Data Access and Utilization, Advances in Space Research, submitted.

Stoneback, R., Burrell, A.G., Klenzing, J., Depew, M.D., 2018. PYSAT: Python Satellite Data Analysis Toolkit. JGR Space Phys. 123, 5271–5283. https://doi.org/10.1029/2018JA025297.

Valentine, D., Zaslavsky, I., Richard, S., Meier, O., Hudman, G., et al., 2021. EarthCube Data Discovery Studio: A gateway into geoscience data discovery and exploration with Jupyter notebooks. Concurrency and Computation: Practice and Experience 33. https://doi.org/10.1002/cpe.6086 e6086.

Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods, 17, 3, 261-272. https://doi.org/10.1038/s41592-019-0686-2.

Weigel, R.S., Vandegriff, J., Faden, J., King, T., Roberts, D.A., et al., 2021. HAPI: An API Standard for Accessing Heliophysics Time Series Data. JGR Space Phys. 126, 12. https://doi.org/10.1029/2021JA029534.