



# Enabling Xarray extensions with new labeled array-like data structures

A proposal to develop new labelled array data structures for the scientific Python ecosystem

## Key Personnel:

This proposal was written in collaboration with the Xarray [developer team](#). Each of the following individuals contributed directly to this proposal.

Name	Affiliation	GitHub handle
Joe Hamman	CarbonPlan, NCAR	<a href="#">jhamman</a>
Anderson Banihirwe	NCAR	<a href="#">andersey005</a>
Aureliana Barghini	B-Open	<a href="#">aughs</a>
Alessandro Amici	B-Open	<a href="#">alexamici</a>
Stephan Hoyer	Google	<a href="#">shoyer</a>
Deepak Cherian	NCAR	<a href="#">dcherian</a>

# Contents

Proposal Summary:	3
Work plans	4
Milestones and Deliverables:	6
Existing Support	8
Landscape Analysis	8
Diversity, Equity, and Inclusion (DEI) Statement	9
Budget	10

# Proposal Summary

Xarray is an open source scientific Python project that provides a data model and toolkit for multidimensional labeled arrays and datasets. Originally developed for analyzing climate data, Xarray is now used across many domains.

Xarray provides two data structures: “DataArray”, which adds dimension names, coordinate labels, and arbitrary metadata to an underlying array object; and “Dataset”, a collection of “DataArray”s. Building on progress supported by NSF, NASA, Nvidia, and CZI, we propose to extend Xarray to encompass two major use-cases for “labeled multi-dimensional arrays.”

First, we propose a new tree-like data structure for hierarchical datasets represented by “groups” in HDF5 and Zarr, essentially a collection of “Datasets”. This structure has wide application in bioimaging, condensed matter physics, remote sensing, image pyramids, and bayesian inference.

Second, there is a need for a lightweight array structure with named dimensions for convenient indexing and broadcasting. Xarray includes such a structure internally (“Variable”). We will factor out “Variable” into a standalone package with minimal dependencies for integration with libraries like scikit-learn. “Variable” will follow established array protocols and the new [data-apis](#) standard. It will be capable of wrapping multiple array-like objects (e.g. NumPy, Dask, Sparse, Pint, CuPy, Pytorch). While “DataArray” fits some of these requirements, it offers a more complex data model than is desired for many applications and depends on Pandas.

We will also use new funding to support regular maintenance, feature development beyond the scope of typical volunteer contributions, and community activities.

# Work plans

Our proposed work plan includes initiatives that broaden Xarray's selection of data structures and provide focused user and developer support to ensure the project continues to grow sustainably. The three main initiatives are:

1. TreeDataset: New data structure supporting hierarchical datasets ([#4118](#))
2. Xvariable: New lightweight Variable API supporting labeled arrays without coordinates ([#3981](#))
3. User and developer support

The first development objective aligns with the broader aim to make Xarray useful for a larger user-base by supporting more complex data-types. The second development aims at making available a generic and mature piece of functionality to other open source projects that require simpler data-structures than the "DataArray".

## TreeDataset: New tree-like data structure supporting hierarchical datasets

We plan to have an extended design and prototype phase during the first year involving stakeholders from downstream projects. Our initial focus will be on developing functionality to support two primary use cases: non-flat namespaces (e.g. arViz, [#1092](#)) and image pyramids / multi-scale data (e.g. [zarr-developers/zarr-specs#50](#)). In the second year, we will turn our focus toward an API consolidation and code stabilisation phase. We have identified the following key tasks and deliverables:

- a. Interview prospective users and identify key use-cases and features. For the two primary use cases of non-flat namespaces and image pyramids, we expect to gather information to guide feature development in areas such as serialization to/from netCDF and Zarr, simultaneous operations across multiple hierarchy levels, and support for domain-specific extensions.
- b. Write a design document describing the initial approach, building on the model that sub-groups are not required to have fully aligned dimensions/indexes.
- c. Initial implementation, in collaboration with downstream users and developers in multiple projects (e.g., arViz, Zarr, Napari, Pangeo, Open Microscopy).
- d. API stabilization and code consolidation.

## Xvariable: New lightweight Variable API (labeled arrays without coordinates)

This work area is about cleaning up Xarray's internals, and allowing our low-level "Variable" data structure to be usable by other projects in the scientific Python ecosystem. We have identified the following key tasks and deliverables:

- a. Formalize Xarray's contract for valid data inside "Variable", and remove/replace some legacy features that would be hard to justify for a generic library:
  - i. Move the "encoding" attribute from "Variable" onto a new "duck array" class that can be used inside a "Variable" ([#5082](#))
  - ii. Expose Xarray's internal model for "explicit array indexing" as a public API. Xarray uses this feature because supporting the full complexity of NumPy's full indexing API is hard for many array implementations.
- b. Separate out and possibly rename/re-brand to Xvariable parts of Xarray internals into new projects. This will increase their visibility, find new users for these tools and improve the maintainability of Xarray itself.
  - i. The "Variable" class will move into a separate project that only depends upon NumPy, and that Xarray in-turn will depend upon. We hope this project will be of interest to users interested in simpler tools than Xarray ([#3981](#)).
  - ii. The new package will support indexing and a limited series of other operations lazily on arrays loaded from disk or remote storage, without loading the entire array into memory. This project is of interest to other projects such as Napari ([#5081](#)).
- c. API stabilization, code consolidation and maintenance of the external project.

## User / developer support

As Xarray's user base has grown, we've seen a steady increase in support requests on Xarray's issue tracker, Stack Overflow, and other community forums. Consequently, the volume of new feature contributions, bug fixes, and documentation enhancements from contributors outside the core developer team is outpacing the bandwidth of the current team. Thus, a key part of our proposal is the allocation of funds to facilitate dedicated community support activities.

We've recently had the opportunity to prototype such a role on our team using a smaller one-time grant from NVIDIA. Using these funds, we've been able to support general project maintenance and focus on prompt user support replies and code review on Pull Requests. Here we plan to follow a similar pattern – funding a part-time community manager role focused on managing Xarray's user support channels and coordinating developer issues, reviews, etc.

We also have reserved a small amount of funding for an annual user survey and general project communications.

# Milestones and Deliverables

Our work plan describes three high-level development objectives, each of which has two to four milestones and deliverables. We list these below, noting the primary developer and their expected completion time (from the start of the funding period).

Milestone	Description	Completion time
1.1	Hierarchical data structure (TreeDataset) requirements and design document + TreeDataset structure in Xarray (pre-alpha)	Month 12
1.2	TreeDataset structure in Xarray (alpha)	Month 12
1.3	TreeDataset structure in Xarray (beta, software and documentation)	Month 18
1.4	TreeDataset structure in Xarray (community handover, software and documentation)	Month 24
2.1	Lightweight data structure (Xvariable) requirements and design document + XArray Variable class refactor	Month 12
2.2	Xvariable split to an external repo in pydata (alpha)	Month 12
2.3	Xvariable project (beta, software and documentation)	Month 18
2.4	Xvariable project (community hand-over, software and documentation)	Month 24
3.1	User / Developer Support	Month 24
3.2	User Survey / Project Communications	Month 24

Our primary goal is to make Xarray more useful to scientists across a broad range of scientific domains. With this in mind, we will continue our annual “user survey”, which we will use to gain insights that will help us tune our development work and the evaluation metrics while we complete the project. We will also track progress in our milestones using a range of traditional metrics such as user visits to our online documentation and closure of GitHub issues. Those metrics are listed below for each high-level development objective:

1. Tree Dataset
  - a. Dataset groups mapped from Zarr or NetCDF ([#1092](#))
  - b. Multiscale arrays / image pyramids ([zarr-developers/zarr-specs#50](#))
2. Xvariable
  - a. Public Variable API with minimal dependencies ([#3981](#))
  - b. Public array indexing interface ([#5081](#))
3. Community engagement metrics

- a. The number of open issues and pull requests over a period of time
- b. The number of new contributors and unique contributors over a period of time
- c. The number of unique users as measured by visits to Xarray's online documentation
- d. Number of domain-specific examples in the documentation
- e. Number of GitHub stars and dependents ("Used By")
- f. Citation of Xarray paper in scientific literature

# Existing Support

In the past two years, Xarray has been fortunate to have received funding to support targeted feature development (from CZI) as well as unrestricted developer time (from NVIDIA). Here we provide a brief list summarizing Xarray's most recent funding:

- Xarray: Multidimensional Labeled Arrays and Datasets in Python, CZI EOSS (Funding Cycle 2), \$150,000 to NumFOCUS, 2020-07-01 – 2021-06-30
- Donation, NVIDIA, \$50,000 to NumFOCUS, 2020-11-05 –

# Landscape Analysis

Today, most of the audience for Xarray uses either lower level data structures (i.e. unlabeled multi-dimensional arrays), tabular data structures (e.g., dataframes) or custom domain-specific tooling. Examples of domain-specific tools include Iris, which is quite similar to Xarray but scoped exclusively for climate data, Scikit-Bio, a library for working with biological data, and Scipp, a label-aware ND-array package for working with neutron-scattering data.

Xarray's advantage is that it solves a very general need in scientific data analysis — by building on other libraries like Dask, NumPy, Pandas, and Zarr — without becoming bloated with domain-specific functionality. This is evidenced by the recent development of domain specific tools that leverage Xarray:

- Arviz builds on Xarray to provide a toolkit for exploratory data analysis and visualization of Bayesian models,
- Napari, a fast, interactive, multi-dimensional image viewer with limited support for visualizing Xarray objects, and
- The Allen SDK which provides tooling for reading and processing Allen Institute for Brain Science data.



# Diversity, Equity, and Inclusion (DEI) Statement

Xarray has adopted a Code of Conduct derived from the popular [Contributor Covenant](#). In this document, which is included in the Xarray source code repository, we make this pledge:

“In the interest of fostering an open and welcoming environment, we as contributors and maintainers pledge to making participation in our project and our community a harassment-free experience for everyone, regardless of age, body size, disability, ethnicity, gender identity and expression, level of experience, nationality, personal appearance, race, religion, or sexual identity and orientation.”

The Xarray development team has, like many open source software projects, evolved organically. As such, it suffers from many of the diversity challenges that are well known in the open source software community. We are committed to doing our best to reverse this counterproductive pattern. In particular, we have engaged in the following activities:

- Worked with NumFOCUS to provide recognition to new contributors each year,
- Provided direct mentoring to individuals from underrepresented groups to help participation with the project, and coordinated tutorials and sprints at events like SciPy and OceanHackWeek for beginners.

# Budget

We have scoped a budget for this full proposal of approximately 400k USD. Because each deliverable could be supported separately, the per deliverable budget is shown below:

1. TreeDataset: 133,400 USD
2. Xvariable: 133,400 USD
3. User and developer support: 126,500 USD