

Application: Inclusive and Accessible Scientific Computing in the Jupyter Ecosystem

Tania Allard - taniar.allard@gmail.com

EOSS4: Essential Open Source Software for Science (Cycle 4)

Summary

ID: EOSS4-0000000164

Last submitted: May 18 2021 04:48 AM (PDT)

Applicant Details

Completed - Mar 29 2021

Applicant Details

Please complete the following information for the Applicant (required):

The information entered should be for the individual submitting the application who will act as the main person responsible for the application and as its point of contact. Information about other Key Personnel on the proposal should be entered where requested in the Software Project Details part of the application. **Complete all fields in this task; all fields are required.**

To edit your name or email, please do so in your account information by clicking your name in the upper right corner and clicking My Account in the dropdown menu.

1. Name: Tania Allard

2. Email: taniar.allard@gmail.com

3. Degrees (check all that apply)

Responses Selected:

Bachelor's degree or equivalent

Master's degree or equivalent

PhD or equivalent

Proposal Details

Completed - Mar 29 2021

Proposal Details

Please complete the following proposal information. **All sections are required.**

1. Proposal Title: Inclusive and Accessible Scientific Computing in the Jupyter Ecosystem

To edit your proposal title, navigate to the main page; click on the three dots to the right of the application title (next to the Preview link); and select Rename from the dropdown menu. Proposal title is limited to a maximum of 75 characters, including spaces.

2. Did you previously apply for funding for this or a related proposal under the CZI EOSS program?

Yes- please specify application number, e.g. EOSS-0000005345, EOSS2-0000005145: EOSS-0000000142 , EOSS2-0000000084

3. Have you previously received funding for this proposal under the CZI EOSS program?

No

4. Proposal Purpose:

Limit to one sentence (maximum of 255 characters, including spaces)

Bring the traditionally marginalized voices of disabled scientists into scientific computing communities via building and applying accessibility tools, standards, and community contribution practices in JupyterLab and the larger Jupyter ecosystem.

5. Amount Requested:

Enter total budget amount requested in USD, including indirect costs; this number should be between \$100k and \$400k total costs and match total described in the Budget Description. Enter whole numbers only (no dollar signs, commas, or cents)

400000

6. Proposal Summary/Scope of Work (required):

Provide a short summary of the application (maximum of 500 words)

For open-source projects such as JupyterLab to live up to their values and maximize their impact across society, they must ensure that the whole experience -both inside and outside the software itself- is accessible to all users. However, while a large community uses JupyterLab, it severely lacks in implementing accessibility standards, making it unusable for many differently-abled users. This proposal aims to improve JupyterLab to build tools and standards that make JupyterLab and Jupyter interfaces accessible to a broader audience. This work will provide a foundation that boosts accessibility in the Jupyter community and the broader PyData ecosystem. In doing so, Project Jupyter will set an example for accessible open-source communities in scientific computing.

We aim to achieve this in three ways.

First, we will focus our primary development efforts on manual and automated accessibility audits. These efforts will ensure that accessibility standards are consistently met (and problems identified) across many Jupyter tools. While various accessibility auditing and integration tests exist, these tests do not adequately account for the complex interactions involved in interactive computing software. Hence, Jupyter needs a unique testing solution based on combined existing accessibility tests and manual tester knowledge. This work will help us mitigate accessibility regressions and ensure new contributions to the project take accessibility into account.

Second, we will address Web Content Accessibility Guidelines violations in JupyterLab and other core Jupyter ecosystem projects it relies on (such as the infrastructural Lumino or the documentation theme PyData Sphinx) as well as derived projects such as JupyterBook. Web Content Accessibility Guidelines are the international standard and foundation for accessibility legislation in many countries. Gaining

compliance with these standards encompasses the development work needed to ensure that disabled scientists can use JupyterLab, related projects, and documentation.

Third, we will develop new documentation on best practices for development, documentation, design, and community guidelines and procedures to include accessibility as a critical part of the contribution process moving forward. We will also work on a set of guides focused on accessibility audits and the usage of Continuous Integration frameworks for accessibility testing. These additions will start with JupyterLab and other Jupyter projects but can serve as a framework for bringing accessible practices to the PyData and scientific computing ecosystems at large.

7. Landscape Analysis (required):

Briefly describe the other software tools (either proprietary or open source) that the audience for this proposal is primarily using. How do the software projects in this proposal compare to these other tools in terms of user base size, usage, and maturity? How do existing tools and the project(s) in this proposal interact? (maximum of 250 words)

We are not aware of any web-based browser or desktop interactive scientific computing interfaces that disabled scientists can successfully use with default accessibility settings. From experience, they are at best able to use combinations of tools to express themselves in code; at worst, they cannot access the tools they need and are, therefore, barred from opportunities in education and industry. To our knowledge, many educational institutions have considered using Jupyter for teaching and the democratisation of scientific computing. However, they cannot accomplish this mission due to the current state of inaccessibility in Jupyter. Finally, many industry projects and bioinformatics teams have drifted away from tools in the Jupyter ecosystem and opted for other technologies with better in-built accessibility standards and capabilities (e.g. VSCode).

Even as it neglects the estimated 20% of the population that is disabled, Jupyter is extraordinarily successful and has a significant influence on the pathways available to people in data science. Nevertheless, as Jupyter interfaces are inaccessible, this creates a technical barrier to data science success and a source of inequity in the ecosystem, which runs directly counter to the project's mission.

Jupyter is not only a standard for scientific computing; its ecosystem supports and influences established projects like VSCode, nteract, and Google Colab. Abiding by Web Content Accessibility Guidelines in Jupyter will transform the many projects that Jupyter is a part of and spring more comprehensive accessibility changes. Leading to a more inclusive scientific computing ecosystem and increased access to computational literacy to more prospective scientists.

8. Value to Biomedical Users:

Briefly described the expected value the proposed scope of work will deliver to the biomedical research community (maximum of 250 words)

The Jupyter ecosystem is a critical cornerstone of technology for biomedical science education, research, and manufacturing. Jupyter represents an ecosystem of open source tools that enable researchers to work with code and data in a single interface. Jupyter's growth in the classroom and laboratory speak to the value of these technologies in science. The current state of these tools, however, make them usable only by abled individuals.

Disability affects roughly 20% of the population, meaning that for every million Jupyter users, we exclude ~200,000 prospective scientists from participating. Also, software that lacks accessibility features can have significant ethical and legal implications in various contexts.

Our proposed work aims to make JupyterLab's computing interface, its documentation, and associated documentation generation tools inclusive to disabled scientists. We will align JupyterLab, and its documentation with Web Content Accessibility Guidelines that set the standards for interfacing with assistive technology. A successful effort will open up biomedical computational literacy for scientists with visual, auditory, physical, speech, cognitive, language, learning, and neurological disabilities. This accessibility effort will also help bring traditionally excluded voices and users into the Jupyter ecosystem, diversify it, and contribute to its long-term sustainability.

9. Open Source Software Projects

How many software projects are involved in your proposal (maximum of five)?

5

Complete the table for each software project. You may need to use the scroll bar at the bottom of the table to scroll right to view and to complete all fields. Alternatively, you can tab to move through and complete the fields. If multiple software projects are involved, details must be entered for all of them. All fields are

required. All URLs should be in the format <https://example.com> and **only one primary link should be provided where requested**:

	Software project name	Homepage URL	Hosting platform	Main code repository	Description of software project
1	JupyterLab	https://jupyterlab.readthedocs.io/en/latest/	GitHub	https://github.com/jupyterlab/jupyterlab	JupyterLab is the next-generation web-based user interface for Project Jupyter.
2	Project Jupyter	https://jupyter.org/	GitHub	https://github.com/jupyter	Project Jupyter exists to develop open-source software, open-standards, and services for interactive computing across dozens of programming languages.
3	pa11y-ci	https://pa11y.org	GitHub	https://github.com/pa11y/pa11y-ci	A command-line tool which iterates over a list of web pages and highlights accessibility issues. This is a CLI that's more geared towards use in CI.
					Jupyter Book is an open-source

4	Jupyterbook	https://jupyterbook.org/intro.html	GitHub	https://github.com/executablebooks/jupyter-book	tool for building publication-quality books and documents from computational material.
5	Robot framework	https://robotframework-jupyterlibrary.readthedocs.io/en/latest/ https://robotframework.org/	GitHub	https://github.com/robotframework-jupyter/robotframework-jupyterlibrary	Robot Framework is a generic open source automation framework. It can be used for test automation and robotic process automation (RPA). A Robot Framework library for automating (testing of) Jupyter end-user applications and extensions

Applicant Details

Completed - May 16 2021

Applicant Details

origins in or who maintain an affiliation to the Middle East or North African regions)	(No response)
Native Hawaiian or Other Pacific Islander (a person having origins in any of the original peoples of Hawaii, Guam, Samoa, or other Pacific Islands)	(No response)
American Indian or Alaska Native (a person having origins in North and South America, including Central America, and who maintain tribal affiliation or community attachment)	(No response)
Prefer not to say	(No response)
Prefer to describe	(No response)

Proposal Details

Completed - May 18 2021

Proposal Details

Please complete the following proposal information. **All sections are required.**

1. Proposal Title: Inclusive and Accessible Scientific Computing in the Jupyter Ecosystem

To edit your proposal title, navigate to the main page; click on the three dots to the right of the application title (next to the Preview link); and select Rename from the dropdown menu. Proposal title is limited to a maximum of 75 characters, including spaces.

2. Did you previously apply for funding for this or a related proposal under the CZI EOSS program?

No

3. Have you previously received funding for this proposal under the CZI EOSS program?

No

4. Proposal Purpose (required):

Limit to one sentence (maximum of 255 characters, including spaces)

Bring systematically marginalized voices of disabled scientists into scientific computing communities via building and applying accessibility tools, standards, and community contribution practices in JupyterLab and the larger Jupyter ecosystem.

5. Amount Requested (required):

Enter the amount requested per year (including indirect costs), as well the total budget requested for all years in USD (this number should be between \$100k and \$400k total, between \$50k and \$200k per year; auto-filled from LOI), including indirect costs; these numbers should match those described in the Budget Description. Enter whole numbers only (no dollar signs, commas, or cents)

Year One	199859
Year Two	199589
Total All Years	399448

6. Proposal Summary (required):

Provide a short summary of the application (maximum of 500 words) (auto-filled from LOI; update if needed)

For open-source projects such as JupyterLab to live up to their values and maximize their impact across society, they must ensure that the whole experience -both inside and outside the software itself- is accessible to all users. However, while a large community uses JupyterLab, it severely lacks in

implementing web content accessibility standards, making it unusable for many disabled users. This proposal aims to improve JupyterLab to build tools and standards that make JupyterLab and Jupyter interfaces accessible to a broader audience of users requiring assistive technologies. This work will provide a foundation that boosts accessibility to Jupyter's community and the broader PyData ecosystem. In doing so, Project Jupyter will set an example for accessible open-source communities in scientific computing.

We aim to achieve this in three ways.

First, we will focus our primary development efforts on manual and automated accessibility audits. These efforts will ensure that accessibility standards are consistently met (and problems identified) across many Jupyter tools. While various accessibility auditing and integration tests exist, these tests do not adequately account for the rich interactions involved in interactive computing software. Hence, Jupyter needs a complex testing solution based on existing accessibility tests and manual tester knowledge. This work will help us mitigate accessibility regressions and ensure new contributions to the project take accessibility into account.

Second, we will address Web Content Accessibility Guidelines violations in JupyterLab and other core Jupyter ecosystem projects it relies on (such as the infrastructural Lumino or the documentation theme PyData Sphinx) as well as derived projects such as JupyterBook. Web Content Accessibility Guidelines 2.1 are the current international standard and foundation for accessibility legislation in many countries. Gaining compliance with these standards encompasses the development work needed to ensure that disabled scientists can use JupyterLab, related projects, and documentation.

Third, we will develop new documentation on best practices for development, documentation, design, and community guidelines and procedures to include accessibility as a critical part of the contribution process moving forward. We will also work on a set of guides focused on accessibility audits and the usage of Continuous Integration frameworks for accessibility testing. These additions will start with JupyterLab and other Jupyter projects but can serve as a framework for bringing accessible practices to the PyData and scientific computing ecosystems at large.

7. Work Plan (required):

A description of the proposed work the applicants are requesting funding for, including resources the applicants will provide that are not part of the requested funding. For software development related work

(e.g., engineering, product design, user research), specify how the work fits into the existing software project roadmap. For community outreach related activities (e.g., sprints, training), specify how these activities will be organized, the target audience, and expected outcomes (maximum of 750 words)

This proposal aims to improve the accessibility of several tools in the Jupyter ecosystem and develop documentation and best practices around accessibility for the larger scientific computing ecosystem. We aim to achieve this through:

Adopting a robust accessibility testing framework for projects in the Jupyter ecosystem and their documentation tools.

Addressing critical technical issues in JupyterLab, Pydata Sphinx theme and JupyterBook, which currently impact their usability and accessibility.

Developing documentation focused on best practices for accessibility compliance and practical steps for maintainers and contributors to the PyData ecosystem to make accessibility a priority in the development workflow.

Comprehensive accessibility testing

Project Jupyter was established around 2014 to support interactive scientific computing and democratising data science. Since then, and until recently, there had been minimal to non-existent efforts to make the tools in the Jupyter ecosystem compliant with international accessibility standards. This lack of accessibility of the Jupyter ecosystem has resulted in educators, researchers, and whole corporations drifting away from the project. Also, it has increased the technical barriers to disability inclusion in the scientific computing ecosystem, further contributing to systemic inequalities and marginalisation of disabled scientists.

To address some of these issues, a group of volunteers in the community formed a Jupyter accessibility working group. This group has been responsible for performing essential accessibility audits and identifying pathways to improve the overall accessibility of the Jupyter ecosystem (<https://github.com/jupyter/accessibility>). This proposal builds on the efforts of this group. To enable the continuation of this work, we propose the following actions:

Adopting a multi-approach testing framework (e.g. Compatibility testing, Unit Testing, Acceptance Testing) to audit conformance Jupyter tools to the Web Content Accessibility Guidelines 2.1 (WCAG). This framework will include:

Using existing testing tools like pa1ly and axe integrations in Jest to audit the complex needs for

inclusive, interactive computing.

Extending the Jupyter's continuous integration tests, based on Galata, a Jest interface for JupyterLab, and the projects its experience is built upon, from UI component libraries to documentation themes.

Performing manual usability and accessibility tests.

Developing a public roadmap to address critical accessibility issues identified by adopting the described framework and the outcomes of previous audits carried out by the accessibility working group.

Writing reports on the findings and actions taken to address accessibility issues in the Jupyter ecosystem.

Work will be performed by Isabela Presedo-Floyd (15 hours/week) and Tania Allard (10 hours/week), using the standard collaboration methods of the project (i.e. GitHub, Discourse) and the organisation of community accessibility calls. The testing and continuous integration improvements will be carried out by the testing and software engineers (to be hired at 20hrs/week) and delivered via pull requests on GitHub and documented accordingly.

We aim for this work to be participatory, actively and meaningfully involving people with disability. To accomplish this, we might need to include testing services like Fable, which connects projects with disabled testers.

Address critical accessibility issues

We will use the generated reports and roadmaps to drive improvements to the JupyterLab codebase. In the first instance, we will focus on WCAG 2.1 guidelines and success criteria while setting the basis for compliance with the forthcoming WCAG 2.2 and WCAG 3 criteria.

To accomplish this, we propose the following:

A set of research-based recommendations to achieve WCAG compliance or equivalent.

Critical fixes and enhancements to the codebase as needed (based on the advice and roadmaps).

The software engineer will carry this work with the Jupyter accessibility working group and deliver it through pull requests on GitHub.

Documentation and outreach efforts

Since disability inclusion is still widespread, much more evidence is available about the impact of excluding people with disabilities than guides on accessible design. This lack of evidence makes it harder for developers and open source contributors to consider accessibility a core part of the development

workflow. We expect this accessibility work to serve as a framework for disability inclusion in the broader scientific computing ecosystems by:

Developing new documentation on best practices for disability inclusion and accessibility for the scientific computing communities (i.e. accessibility auditing, accessible documentation, inclusive design and participatory community practices).

Mainstreaming disability in the Jupyter ecosystem by considering disability inclusion and accessibility in the contribution and development workflows moving forward.

Creating documentation on the usage of Continuous Integration frameworks for accessibility testing.

Presenting talks on accessibility and disability inclusion and the project's outcomes at relevant conferences and spaces (i.e. Maintainer Summit at PyCon and SciPy, Read the Docs).

The whole team will work on these documentation deliverables in conjunction with the Jupyter accessibility working group through GitHub.

8. Milestones and Deliverables (required):

List expected milestones and deliverables, and their expected timeline. Be specific and include (where possible) any goals for metrics the software project(s) are expected to reach upon completion of the grant (maximum of 500 words)

We anticipate the work supported by this grant to contribute to Jupyterlab becoming a usable and welcoming project for disabled scientists, thus improving the overall community.

Our main priority will be Jupyterlab accessibility improvements to leverage the existing community efforts during the first year.

First 6 months: we will focus on laying the foundations for a sustainable accessibility effort in JupyterLab. The expected deliverables are:

Develop the first iteration of automated tests for JupyterLab based on the existing accessibility by integrating jest-axe into JupyterLab's Galata UI testing framework.

Produce automated reports on Jupyterlab's compliance to WCAG standards using Galata and pa11y.

Generate a public roadmap for accessibility improvements to Jupyterlab (available at [jupyter/accessibility](https://jupyter.org/accessibility)). This roadmap will also help establish pathways for participatory community

engagement and contributions.

Work towards JupyterLab compliance with WCAG 2.1 requirements focused on low-vision and ambulatory users. In scope are:

Jupyter should be navigable by keyboard alone.

Support for 400% zoom.

Adequate colour contrast for interactive regions.

At 12 months: we will have made significant contributions towards refining, documenting, and stabilising JupyterLab-specific improvements. Specific deliverables are:

Align testing, compliance, and roadmap efforts with the upcoming WCAG 2.2 release (scheduled for late 2021).

Automated accessibility tests merged into Jupyterlab's documentation theme and PyData Sphinx (built on pally and Lighthouse).

Work towards JupyterLab compliance with WCAG 2.1/2.2 guidelines, specifically focused on accessibility for blind users.

Produce documentation on accessibility features new to JupyterLab.

Improve Jupyterlab's contributor documentation to account for accessibility testing and best practices as part of the workflow.

Achieve full compliance with WCAG 2.1/2.2 requirements in the PyData Sphinx theme.

Achieve full compliance with WCAG 2.1/2.2 requirements in the JupyterLab documentation.

In the second year, we plan to extend the work carried during the first year into an accessibility framework. We expect this framework to support other Jupyter-related projects and the broader scientific computing ecosystem to adopt accessibility and universal design best practices.

At 18 months: we will work on extending our work within the Jupyter ecosystem. Specific deliverables are:

Identify and reach out to other projects in the Jupyter ecosystem needing to incorporate accessibility tools and/or practices into their project-specific workflow.

Automated tests contributed to at least one other project in the Jupyter ecosystem.

Contribute improvements to the automated tests and contribution practices developed in year one, based on input from other Jupyter projects.

Produce documentation on best practices for considering accessibility in writing documentation, code contribution, design, community outreach and engagement.

At 24 months: we will ensure that our work is recorded and can be used to support future accessibility efforts in the PyData ecosystem. The key deliverables are:

Contribute accessibility best practices documentation to at minimum one other project in the Jupyter ecosystem.

Produce documentation on our process for auditing interactive computing software.

Present the work and its outcome at scientific computing, developer and documentation focused conferences.

9. Existing Support (required):

List active and recent (previous two calendar years) financial or in-kind support for the software project(s), including duration, amount in USD, and source of funding. Include in this section any previous funding for these software projects received from CZI (maximum of 250 words)

The Jupyter Project encompasses a large number of subprojects, developed by many teams around the world, here is a list of grants that the Jupyter Project received. None were for direct development of the project described in this proposal.

- 2014-2020: Helmsley Charitable Trust, Moore Foundation, Sloan Foundation – Original Jupyter grant - \$6,000,000 - Grant received by UC Berkeley and Cal Poly

- 2018-2021: Sloan foundation – Enabling Safe Access to Sensitive Data - \$1,680,000 - Grant received by UC Berkeley and Cal Poly

- 2018-2020: Schmidt Futures – Integrated Digital Information System Research– \$2,200,000 – Grant received by Cal Poly

- 2019-2021: Helmsley Charitable Trust – Jupyter Sustainability – \$1,000,000 – grant Received by UC Berkeley and Cal/Poly

- 2019-2022: National Science Foundation – Jupyter meets the Earth: Enabling discovery in geoscience through interactive computing at scale – \$1,700,000 – Grant received by UC Berkeley

- 2019-2020: Chan Zuckerberg Initiative – JupyterHub Contributor in Residence – \$141,000 – Grant Received by NumFocus

- 2020-2021: Chan Zuckerberg Initiative – Real-Time Collaboration in Jupyter – \$241,000 – Grant received by QuanSight

- 2015–Present: Bloomberg, Two Sigma, JP Morgan, Amazon – Collaboration with employees. In-kind.

- 2015-Present: DE Shaw, Goldman Sachs, and other companies – Commercial funding for development

10. Landscape Analysis (required):

Briefly describe the other software tools (either proprietary or open source) that the audience for this proposal is primarily using. How do the software projects in this proposal compare to these other tools in terms of size of user base, usage, and maturity? How do existing tools and the project(s) in this proposal interact? (maximum of 250 words) (auto-filled from LOI; update if needed)

We are not aware of any web-based browser or desktop interactive scientific computing interfaces that disabled scientists can successfully use with default accessibility settings. From experience, they are at best able to use combinations of tools to express themselves in code; at worst, they cannot access the tools they need and are, therefore, barred from opportunities in education and industry. To our knowledge, many educational institutions have considered using Jupyter for teaching and the democratisation of scientific computing. However, they cannot accomplish this mission due to the current state of inaccessibility in Jupyter. Finally, many industry projects and bioinformatics teams have drifted away from tools in the Jupyter ecosystem and opted for other technologies with better in-built accessibility standards and capabilities (e.g. VSCode).

Even as it neglects the estimated 20% of the population that is disabled, Jupyter is extraordinarily successful and has a significant influence on the pathways available to people in data science. Nevertheless, as Jupyter interfaces are inaccessible, this creates a technical barrier to data science success and a source of inequity in the ecosystem, which runs directly counter to the project's mission.

Jupyter is not only a standard for scientific computing; its ecosystem supports and influences established projects like VSCode, nteract, and Google Colab. Abiding by Web Content Accessibility Guidelines in Jupyter will transform the many projects that Jupyter is a part of and spring more comprehensive accessibility changes. Leading to a more inclusive scientific computing ecosystem and increased access to computational literacy to more prospective scientists.

11. Value to Biomedical Users (required):

Briefly describe the expected value the proposed scope of work will deliver to the biomedical research community (maximum of 250 words) (auto-filled from LOI; update if needed)

The Jupyter ecosystem is a critical cornerstone of technology for biomedical science education, research, and manufacturing. Jupyter represents an ecosystem of open source tools that enable researchers to work with code and data in a single interface. Jupyter's growth in the classroom and laboratory speak to the value of these technologies in science. The current state of these tools, however, make them usable only by abled individuals.

Disability affects roughly 20% of the population, meaning that for every million Jupyter users, we exclude ~200,000 prospective scientists from participating. Also, software that lacks accessibility features can have significant ethical and legal implications in various contexts.

Our proposed work aims to make JupyterLab's computing interface, its documentation, and associated documentation generation tools inclusive to disabled scientists. We will align JupyterLab, and its documentation with Web Content Accessibility Guidelines that set the standards for interfacing with assistive technology. A successful effort will open up biomedical computational literacy for scientists with visual, auditory, physical, speech, cognitive, language, learning, and neurological disabilities. This accessibility effort will also help bring traditionally excluded voices and users into the Jupyter ecosystem, diversify it, and contribute to its long-term sustainability.

12. Diversity, Equity, and Inclusion Statement (required):

Advancing DEI is a [core value](#) for CZI, and we are requesting information on your efforts in this area. Describe any efforts the software project(s) named in this proposal have undertaken to increase diversity, equity, and inclusion with respect to their contributors and audience. Please see [examples](#) from applications funded in previous cycles (maximum of 250 words)

Project Jupyter is committed to creating an inclusive and equitable environment for community members from all walks of life. It has prioritised inclusion by creating an inviting, welcoming, and supporting community culture. So far, efforts to introduce and support marginalised groups have focused on community events and sub-projects such as organising regular community meetings across several time zones. Also, there have been considerable efforts to onboard new contributors through programmes like Outreachy, the Jupyter Contributor in Residence initiative and with support from Cal Poly San Luis Obispo.

However, there is still work to be done- particularly in the area of disability inclusion. Lack of accessibility and standards in the design and development of digital technologies and infrastructure can prevent a substantial number of people with disabilities from achieving social inclusion and digital literacy (Rimmerman, 2013, pp. 3, 76). Jupyter current state is inaccessible to many individuals and organisations, making it hard to live up to its core principles.

It was not until recently that JupyterLab community members launched an initiative to address this issue. The proposed work builds on universal design principles: equitable, flexible, intuitive and straightforward use and low physical effort (DESA, 2013, p. 18). Adopting such principles will allow disabled and historically underrepresented groups to participate in open source. We commit to taking concrete steps towards inclusion, such as ensuring that communication, organisational and attitudinal barriers to inclusion are identified and addressed promptly in the project and being transparent about this project's outcomes and impact in the community.

Optional Attachments

Completed - May 18 2021

Upload in PDF format; attachments should be uploaded in a combined single PDF. Include up to three pages of additional information. This section can include figures, charts and tables, references for the proposal, or any additional material in support of the proposal (maximum of three pages). Uploading any additional information is completely optional and not required.

scopus_jupyter_citations

Filename: scopus_jupyter_citations.pdf **Size:** 45.3 kB

CV of Applicant

Completed - May 16 2021

Upload in PDF format; include current and recent employment, education history, and references to any major publications, software contributions, or other relevant outputs (maximum of two pages)

TaniaAllard_CV

Filename: TaniaAllard_CV.pdf **Size:** 52.4 kB

Budget Description

Completed - May 18 2021

Upload in PDF format; budgets can be uploaded in a combined single PDF or one PDF for each software project (one page per software project maximum)

- *Description of the costs to be funded by this grant at a high level and in narrative or tabular form, outlining costs for personnel (including names, if known), supplies, equipment, travel, meetings/hackathons/sprints, subcontracts, other costs, and up to 15% indirect costs (excluding equipment and subcontracts).*
- *Indirect costs are limited to up to 15% of direct costs and are included within the annual budget total. Indirect costs may not be assessed on capital equipment or subcontracts, but subcontractors may include up to 15% indirect costs of their direct costs.*
- *Budget should be requested in US dollars.*
- *International grantees must use all grant funds exclusively for activities conducted outside the United States of America. Travel expenses to the United States (including round-trip tickets) should not be covered from the requested grant funds.*
- *Application budgets must reflect the actual needs of the proposal. The Chan Zuckerberg Initiative will work closely with successful applicants to arrive at a mutually acceptable budget after review.*

Jupyter_EOSS_budget

Filename: Jupyter_EOSS_budget.pdf **Size:** 31.0 kB

Number of Open Source Software Project(s)

Completed - May 16 2021

Number of Open Source Software Projects

How many software projects are involved in this proposal that will be supported by this grant?

Each proposal can request funding for work that includes up to five open source software projects. If multiple software projects are involved, details must be entered for all of them.

If your proposal involves more than one software project, mark this step as complete so you can fill in details for additional projects. You may need to scroll down using the scroll bar in the task menu on the left side of the page to see these tasks. To change the number of software projects indicated after marking as complete, click the three dots in the upper right of the task and select edit, which will re-open the task.

3

Open Source Software Project #1: Details

Completed - May 18 2021

Software Project #1: Details

Provide details and metrics for each open source software project that will be supported by the grant to help us assess its impact and quality.

Software Project Details:

Complete the following table for Open Source Software Project #1 of your proposal. All URLs should be in the format <https://example.com> and only one primary link should be provided.

1. Software Project name (required)	JupyterLab
2. Homepage URL (required)	https://jupyterlab.readthedocs.io/en/latest/
3. Hosting platform (required)	GitHub
4. Main code repository (e.g. GitHub URL) (required)	https://github.com/jupyterlab/jupyterlab
5. DOI of major publication(s) describing software project (if applicable)	(No response)
6. Social media handles (if applicable)	@projectjupyter
7. Do you or software project key personnel have commit rights to the code repositories for this software project? (required)	No
8. Short description of software project (200 words maximum) (required)	JupyterLab is the next-generation web-based user interface for Project Jupyter and a key player in interactive computing across domains.

List of Known Key Personnel:

Key personnel are people involved in the software project who will be supported by the grant if the application is successful.

Complete the following for the key personnel on the open source software project listed above (up to 5) (required); **enter n/a if any field is not applicable**. Personnel to be hired that have not been identified at this time can be listed in the budget section. You may need to use the scroll bar at the bottom of the table to scroll right to view and to complete all fields. Alternatively, you can tab to move through and complete the fields. **To add another person/row (up to five), click the box at the end of the row.**

	First name	Last name	Email address	Current employer /Affiliation	Job title	Developer username if applicable (e.g., GitHub handle)	Country of Residence	Add another person/row
1	Tania	Allard	tallard@quansight.com	Quansight	Co-director	trallard	United Kingdom	✓
2	Isabela	Presedo-Floyd	ipresedo@quansight.com	Quansight	UX/UI designer	isabela-pf	United States	✕

Software Project Metrics: Quality (required):

Complete for the open source software project listed above.

1. What is the software project license?

Permissive license (e.g. BSD 3-Clause, MIT, Apache 2.0)

2. What is the main programming language?

Other (please specify): Typescript

3. Does the software project have a code of conduct?

Yes

3. Link (optional; format <https://example.com>):

https://github.com/jupyterlab/.github/blob/master/CODE_OF_CONDUCT.md

4. Does the software project have end-user documentation?

Yes

4. Link (optional; format <https://example.com>):

<https://jupyterlab.readthedocs.io/en/stable/>

5. Does the software project have an issue tracker?

Yes

5. Link (optional; format <https://example.com>):

<https://github.com/jupyterlab/jupyterlab/issues>

6. Does the software project have a community engagement / Q&A forum (self-hosted, on Stack Exchange etc.)?

Yes

6. Link (optional; format <https://example.com>):

<https://discourse.jupyter.org/>

7. Does the software project have contribution / coding guidelines?

Yes

7. Link (optional; format <https://example.com>):

<https://jupyterlab.readthedocs.io/en/latest/developer/contributing.html>

8. Are there examples or demo notebooks, scripts, and datasets?

Yes

8. Link (optional; format <https://example.com>):

<https://mybinder.org/v2/gh/jupyterlab/jupyterlab-demo/5a5eb6b?urlpath=lab/tree/demo>

9. Is there a corresponding package available in a package manager (PyPi, CRAN, etc.)?

Yes

9. Link (optional; format <https://example.com>):

<https://pypi.org/project/jupyterlab/>

<https://anaconda.org/conda-forge/jupyterlab>

10. Does the software project support continuous integration for testing?

Yes

10. Comment (optional):

GitHub actions <https://github.com/jupyterlab/jupyterlab/tree/master/.github/workflows>

Software Project Metrics: Impact (optional):

Complete the following for the open source software project listed above. **Providing metrics is optional and metrics can be approximate.** For each metric, please provide a source, clarify how the metric was computed, and/or provide any other comments. For monthly metrics, please provide data from the most recent month for which the corresponding metric is available.

1. Complete the following table. List the number and explanation for each, if needed:

	Number	Comment
Scholarly paper(s) (including preprints) citing or mentioning the software project		Not specific for Jupyterlab but a plot for Jupyter citations in pdf attachment added
Monthly users, if applicable (based on one or more of the following: monthly downloads from websites, monthly downloads from package managers, monthly unique requests for updates, etc.)	14582478	Downloads from https://pepy.tech/project/jupyterlab
Software projects that depend on the project (if applicable)		
Monthly visitors to project's website, discussion forum (e.g. Stack Overflow), or similar	7000000	Yearly visits to man oage

2. Size of the largest potential user base:

	Number	Comment
Estimate the potential number of unique users who could adopt this project in the relevant field/discipline. Use as guidance the number of users of comparable projects, the number of papers published in the domain to which the project is applicable, number of labs able to adopt the project, etc.	over 100,000	

3. List of upstream, downstream, or related software projects that the team is contributing to or receiving contributions from:

The broader Jupyter ecosystem

4. Additional metrics from project code repositories and package managers:

Provide a short description of any considerations or caveats we should be aware of when computing metrics (e.g. a recent change in the name or hosting of the repository), or any additional information you would like to share about the project's impact and quality. (maximum of 500 words)

(No response)

Open Source Software Project #2: Details

Completed - May 18 2021

Software Project #2: Details

Provide details and metrics for each open source software project that will be supported by the grant to help us assess its impact and quality.

Software Project Details:

Complete the following table for Open Source Software Project #2 of your proposal. All URLs should be in the format <https://example.com> and only one primary link should be provided.

1. Software Project name (required)	PyData Sphinx
2. Homepage URL (required)	https://pydata-sphinx-theme.readthedocs.io
3. Hosting platform (required)	GitHub
4. Main code repository (e.g. GitHub URL) (required)	https://github.com/pydata/pydata-sphinx-theme
5. DOI of major publication(s) describing software project (if applicable)	(No response)
6. Social media handles (if applicable)	pandas_dev
7. Do you or software project key personnel have commit rights to the code repositories for this software project? (required)	No
8. Short description of software project (200 words maximum) (required)	PyData Sphinx is a Sphinx theme commonly used for documentation across the PyData ecosystem. Major projects that use this theme include Pandas, NumPy, Bokeh, JupyterHub and Binder, and Fairlearn.

List of Known Key Personnel:

Key personnel are people involved in the software project who will be supported by the grant if the application is successful.

Complete the following for the key personnel on the open source software project listed above (up to 5) (required); enter n/a if any field is not applicable. Personnel to be hired that have not been identified at this time can be listed in the budget section. You may need to use the scroll bar at the bottom of the table to scroll right to view and to complete all fields. Alternatively, you can tab to move through and complete the fields. **To add another person/row (up to five), click the box at the end of the row.**

	First name	Last name	Email address	Current employer /Affiliation	Job title	Developer username if applicable (e.g., GitHub handle)	Country of Residence	Add another person/row
1	Tania	Allard	tallard@quansight.com	Quansight	Co-director	trallard	United Kingdom	✓
2	Isabela	Presedo-Floyd	ipresedo@quansight.com	Quansight	UX/UI designer	isabela-pf	United States	✕

Software Project Metrics: Quality (required):

Complete for the open source software project listed above.

1. What is the software project license?

Permissive license (e.g. BSD 3-Clause, MIT, Apache 2.0)

2. What is the main programming language?

Python

3. Does the software project have a code of conduct?

No

3. Link (optional; format https://example.com):

(No response)

4. Does the software project have end-user documentation?

Yes

4. Link (optional; format https://example.com):

<https://pydata-sphinx-theme.readthedocs.io/en/latest/index.html>

5. Does the software project have an issue tracker?

Yes

5. Link (optional; format <https://example.com>):

<https://github.com/pydata/pydata-sphinx-theme/issues>

6. Does the software project have a community engagement / Q&A forum (self-hosted, on Stack Exchange etc.)?

Yes

6. Link (optional; format <https://example.com>):

<https://groups.google.com/g/pydata>

7. Does the software project have contribution / coding guidelines?

Yes

7. Link (optional; format <https://example.com>):

<https://pydata-sphinx-theme.readthedocs.io/en/latest/contributing.html>

8. Are there examples or demo notebooks, scripts, and datasets?

Yes

8. Link (optional; format <https://example.com>):

<https://pydata-sphinx-theme.readthedocs.io/en/latest/>

9. Is there a corresponding package available in a package manager (PyPi, CRAN, etc.)?

Yes

9. Link (optional; format <https://example.com/>):

<https://pypi.org/project/pydata-sphinx-theme/>

10. Does the software project support continuous integration for testing?

Yes

10. Comment (optional):

(No response)

Software Project Metrics: Impact (optional):

Complete the following for the open source software project listed above. **Providing metrics is optional and metrics can be approximate.** For each metric, please provide a source, clarify how the metric was computed, and/or provide any other comments. For monthly metrics, please provide data from the most recent month for which the corresponding metric is available.

1. Complete the following table. List the number and explanation for each, if needed:

	Number	Comment
Scholarly paper(s) (including preprints) citing or mentioning the software project		
Monthly users, if applicable (based on one or more of the following: monthly downloads from websites, monthly downloads from package managers, monthly unique requests for updates, etc.)	84653	Data from https://pepy.tech/project/pydata-sphinx-theme/
Software projects that depend on the project (if applicable)		
Monthly visitors to project's website, discussion forum (e.g. Stack Overflow), or similar		

2. Size of the largest potential user base:

	Number	Comment
Estimate the potential number of unique users who could adopt this project in the relevant field/discipline. Use as guidance the number of users of comparable projects, the number of papers published in the domain to which the project is applicable, number of labs able to adopt the project, etc.	over 100,000	

3. List of upstream, downstream, or related software projects that the team is contributing to or receiving contributions from:

Sphinx

Most of the PyData ecosystem uses now this theme

4. Additional metrics from project code repositories and package managers:

Provide a short description of any considerations or caveats we should be aware of when computing metrics (e.g. a recent change in the name or hosting of the repository), or any additional information you would like to share about the project's impact and quality. (maximum of 500 words)

(No response)

Open Source Software Project #3: Details

Completed - May 18 2021

Software Project #3: Details

Provide details and metrics for each open source software project that will be supported by the grant to help us assess its impact and quality.

Software Project Details:

Complete the following table for Open Source Software Project #3 of your proposal. All URLs should be in the format <https://example.com> and only one primary link should be provided.

1. Software Project name (required)	Galata
2. Homepage URL (required)	https://github.com/jupyterlab/galata
3. Hosting platform (required)	GitHub
4. Main code repository (e.g. GitHub URL) (required)	https://github.com/jupyterlab/galata
5. DOI of major publication(s) describing software project (if applicable)	(No response)
6. Social media handles (if applicable)	(No response)
7. Do you or software project key personnel have commit rights to the code repositories for this software project? (required)	No
8. Short description of software project (200 words maximum) (required)	Galata is a UI testing framework designed specifically for JupyterLab. It is optimized for handling and reporting on the rich interactions that are core to properly testing interactive computing workflows.

List of Known Key Personnel:

Key personnel are people involved in the software project who will be supported by the grant if the application is successful.

Complete the following for the key personnel on the open source software project listed above (up to 5) (required); enter n/a if any field is not applicable. Personnel to be hired that have not been identified at this time can be listed in the budget section. You may need to use the scroll bar at the bottom of the table to scroll right to view and to complete all fields. Alternatively, you can tab to move through and complete the fields. **To add another person/row (up to five), click the box at the end of the row.**

	First name	Last name	Email address	Current employer /Affiliation	Job title	Developer username if applicable (e.g., GitHub handle)	Country of Residence	Add another person/row
1	Tania	Allard	tallard@quansight.com	Quansight	Co-director	trallard	United Kingdom	✓
2	Isabela	Presedo-Floyd	ipresedo@quansight.com	Quansight	UX/UI designer	isabela-pf	United States	✕

Software Project Metrics: Quality (required):

Complete for the open source software project listed above.

1. What is the software project license?

Permissive license (e.g. BSD 3-Clause, MIT, Apache 2.0)

2. What is the main programming language?

Javascript

3. Does the software project have a code of conduct?

Yes

3. Link (optional; format https://example.com):

https://github.com/jupyter/governance/blob/master/conduct/code_of_conduct.md

4. Does the software project have end-user documentation?

No

4. Link (optional; format https://example.com):

(No response)

5. Does the software project have an issue tracker?

Yes

5. Link (optional; format https://example.com):

<https://github.com/jupyterlab/galata/issues>

6. Does the software project have a community engagement / Q&A forum (self-hosted, on Stack Exchange etc.)?

Yes

6. Link (optional; format <https://example.com>):

<https://discourse.jupyter.org/>

7. Does the software project have contribution / coding guidelines?

Yes

7. Link (optional; format <https://example.com>):

<https://jupyterlab.readthedocs.io/en/latest/developer/contributing.html>

8. Are there examples or demo notebooks, scripts, and datasets?

Yes

8. Link (optional; format <https://example.com>):

<https://github.com/jupyterlab/galata/tree/main/packages/galata/tests>

9. Is there a corresponding package available in a package manager (PyPi, CRAN, etc.)?

Yes

9. Link (optional; format <https://example.com>):

<https://www.npmjs.com/package/@jupyterlab/galata>

10. Does the software project support continuous integration for testing?

Yes

10. Comment (optional):

(No response)

Software Project Metrics: Impact (optional):

Complete the following for the open source software project listed above. **Providing metrics is optional and metrics can be approximate.** For each metric, please provide a source, clarify how the metric was computed, and/or provide any other comments. For monthly metrics, please provide data from the most recent month for which the corresponding metric is available.

1. Complete the following table. List the number and explanation for each, if needed:

	Number	Comment
Scholarly paper(s) (including preprints) citing or mentioning the software project		
Monthly users, if applicable (based on one or more of the following: monthly downloads from websites, monthly downloads from package managers, monthly unique requests for updates, etc.)		
Software projects that depend on the project (if applicable)		
Monthly visitors to project's website, discussion forum (e.g. Stack Overflow), or similar		

2. Size of the largest potential user base:

	Number	Comment
Estimate the potential number of unique users who could adopt this project in the relevant field/discipline. Use as guidance the number of users of comparable projects, the number of papers published in the domain to which the project is applicable, number of labs able to adopt the project, etc.		

3. List of upstream, downstream, or related software projects that the team is contributing to or receiving contributions from:

(No response)

4. Additional metrics from project code repositories and package managers:

Provide a short description of any considerations or caveats we should be aware of when computing metrics (e.g. a recent change in the name or hosting of the repository), or any additional information you would like to share about the project's impact and quality. (maximum of 500 words)

(No response)