

EIC Software Infrastructure Review

Code Repository and Continuous Integration

Wouter Deconinck

*On behalf of the EPIC
Collaboration*

EIC Software Statement of Principles: Community

4. We will aim for user-centered design.

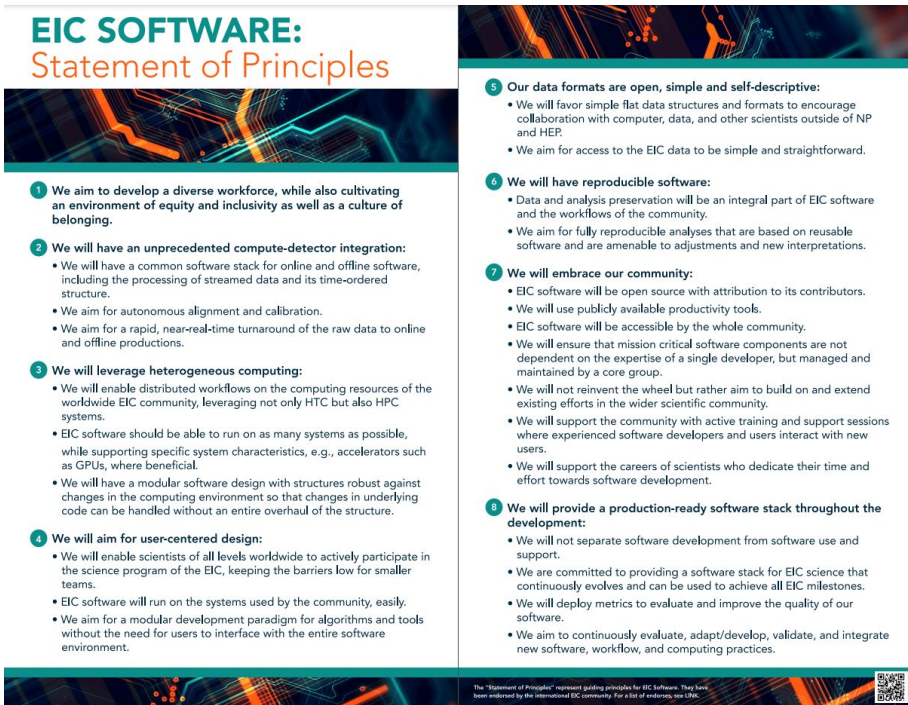
- “[...] keeping the barriers low [...]”
- “We aim for a modular development paradigm [...]”

6. We will have reproducible software.

- “Data and analysis preservation will be an integral part of EIC software [...]”

7. We will embrace our community.


- “We will use publicly available productivity tools.”
- “EIC software will be accessible to the whole community.”
- “We will support the careers [...]”



EIC SOFTWARE: Statement of Principles

- 1 We aim to develop a diverse workforce, while also cultivating an environment of equity and inclusivity as well as a culture of belonging.**
- 2 We will have an unprecedented compute-detector integration:**
 - We will have a common software stack for online and offline software, including the processing of streamed data and its time-ordered structure.
 - We aim for autonomous alignment and calibration.
 - We aim for a rapid, near-real-time turnaround of the raw data to online and offline productions.
- 3 We will leverage heterogeneous computing:**
 - We will enable distributed workflows on the computing resources of the worldwide EIC community, leveraging not only HTC but also HPC systems.
 - EIC software should be able to run on as many systems as possible, while supporting specific system characteristics, e.g., accelerators such as GPUs, where beneficial.
 - We will have a modular software design with structures robust against changes in the computing environment so that changes in underlying code can be handled without an entire overhaul of the structure.
- 4 We will aim for user-centered design:**
 - We will enable scientists of all levels worldwide to actively participate in the science program of the EIC, keeping the barriers low for smaller teams.
 - EIC software will run on the systems used by the community, easily.
 - We aim for a modular development paradigm for algorithms and tools without the need for users to interface with the entire software environment.
- 5 Our data formats are open, simple and self-descriptive:**
 - We will favor simple flat data structures and formats to encourage collaboration with computer, data, and other scientists outside of NP and HEP.
 - We aim for access to the EIC data to be simple and straightforward.
- 6 We will have reproducible software:**
 - Data and analysis preservation will be an integral part of EIC software and the workflows of the community.
 - We aim for fully reproducible analyses that are based on reusable software and are amenable to adjustments and new interpretations.
- 7 We will embrace our community:**
 - EIC software will be open source with attribution to its contributors.
 - We will use publicly available productivity tools.
 - EIC software will be accessible by the whole community.
 - We will ensure that mission critical software components are not dependent on the expertise of a single developer, but managed and maintained by a core group.
 - We will not reinvent the wheel but rather aim to build on and extend existing efforts in the wider scientific community.
 - We will support the community with active training and support sessions where experienced software developers and users interact with new users.
 - We will support the careers of scientists who dedicate their time and effort towards software development.
- 8 We will provide a production-ready software stack throughout the development:**
 - We will not separate software development from software use and support.
 - We are committed to providing a software stack for EIC science that continuously evolves and can be used to achieve all EIC milestones.
 - We will deploy metrics to evaluate and improve the quality of our software.
 - We aim to continuously evaluate, adapt/develop, validate, and integrate new software, workflow, and computing practices.

The "Statement of Principles" represent guiding principles for EIC Software. They have been endorsed by the international EIC community for a list of members, see EIC.



Code Repositories and Continuous Integration???

Code Repository:

A central location for collaborative development of all software components, and for preservation of a full record of the development activity.

Several widely used options based on git:

- GitHub (github.com or enterprise instance)
- GitLab (gitlab.com or self-hosted instance)
- Others...

Or some more esoteric non-git options...

Milestones and versioning, reproducibility, preservation, collaboration, code review

Continuous Integration/Deployment (CI/CD):

A strategy of automatic evaluation of software components, and of automatic deployment into testing and production environments.

Tightly integrated with repositories:

- GitHub
- GitLab

Or as a separate service:

- Jenkins, Travis, CircleCI

Automation, quality control, workflows, deployment into production environments

Why is this important?

A community of 1200+ EIC researchers contributes to or relies upon software developed simultaneously by 100s of people.

We must be able to provide a **validated reference implementation** in a **fast-moving development environment** during the phase towards CD2/3a.

We must maintain insight in the software versions (for all components) that were used to reach design decision (~ Data Analysis & Preservation).

Note: 100+ developers and 1000+ users makes a pay-per-user service infeasible. At \$10s per user per year (or per month!) this is not a sustainable cost model. Self-hosted services operated by laboratories can fit under operating expenses.

Our Requirements for Code Repository/Integration

- Service should **not require a paid account per user**
- Service should be **accessible from anywhere in the world**, without requirements for a specific DOE laboratory account
- Repositories should allow for **configurable access policies**, ranging from world-readable to private (with access only for select users)
- Supports **continuous integration** into production environments
- **Non-restrictive limits** on our level of interaction (for public projects):
 - ≥ 1000 repositories (sum of public and private),
 - ≥ 10 GB repository size (with ability to increase as needed without significant cost),
 - ≥ 1 TB / mo bandwidth (with ability to increase as needed without significant cost).

Solutions Considered

GitHub Organization (free tier)

- GitHub is a closed-source commercial product with a free tier for public projects
- Powerful cloud backends (Azure) with administrative limits to encourage upgrades to enterprise tier
- Premier platform for open source projects
- Imposing of future restrictions is unlikely

Experience in the EIC community:

- Existing EIC organization on GitHub
- Many proposal-stage projects on GitHub

High accessibility, standard platform, but limits on continuous integration opportunities

Self-Hosted GitLab Server (e.g. ANL eicweb)

- GitLab Server is an open-source product (with features as commercial add-ons)
- Self-hosting allows **scaling with demand**, but requires **dedicated personnel** and an institutional **commitment to continuity**

Experience in EIC community:

- Existing EIC on commercial GitLab
- Many yellow report projects on GitLab
- Self-hosted GitLab server at ANL used by ATHENA collaboration

High customizability, dedicated processing power, excels at continuous integration

Decision Consensus: Hybrid Solution

- “We will implement a **hybrid solution** that uses **GitHub as the primary code repository**, while using the **eicweb GitLab instance for CI/CD.**”
- “An ad hoc committee of eicweb experts will investigate the best option for leveraging CI/CD at ANL using GitHub (e.g. GitHub runners, mirrors, webhooks, etc...)”
- “The existing EIC organization at GitHub established by the EIC User Group Software Working Group will be used.
 - Some admin privileges will need to be shared with the EPIC Working Group conveners.”
- “The best practices model for the repository will include:
 - Repositories will be open and public unless there is a specific reason to make them private
 - External packages will not be forked/cloned to the eic organization and modified unless under extremely exceptional circumstances.”

Hybrid solution **meets all requirements** for code repository and continuous integration.

Examples of comparable hybrid solutions in other large projects: HPC-oriented package manager spack uses GitHub as its community front-end with DOE-hosted exascale resources with GitLab instances as backends.

GitHub Organization Management



Electron-Ion Collider (EIC) Software

Electron-Ion Collider (EIC) software, documentation and resources

<https://eic.github.io> [✉ eicug-software-conveners@eicug.org](mailto:eicug-software-conveners@eicug.org)

README.md 

This organization collects all Electron-Ion Collider (EIC) software, repositories, documentation and resources. It is maintained by the EIC Software Group and the EPIC Collaboration Working Groups.

How to join?

All EIC users may request to become part of this organization. Simply email the [EIC User Group Software Working Group conveners](#) from your institutional email address with your GitHub account and whether you or your sponsor/advisor is a member of the EIC User Group listed on the [Phone Book](#). This will give you read access to all public repositories.

You may also wish to join teams such as [EPIC devs](#) to gain write access to select repositories.

GitHub Organization Management

Any EIC collaborator can join this community and use its resources.

Administered by the EICUG Software Working Group conveners.

Sub-teams for subprojects:

- [@eic/epic-devs](#): write permission to EPIC-related repositories (branches)
- [@eic/epic-admins](#): admin permission to EPIC-related repositories (settings)

Tools for collaborative development:

- GitHub Discussions
- GitHub Projects

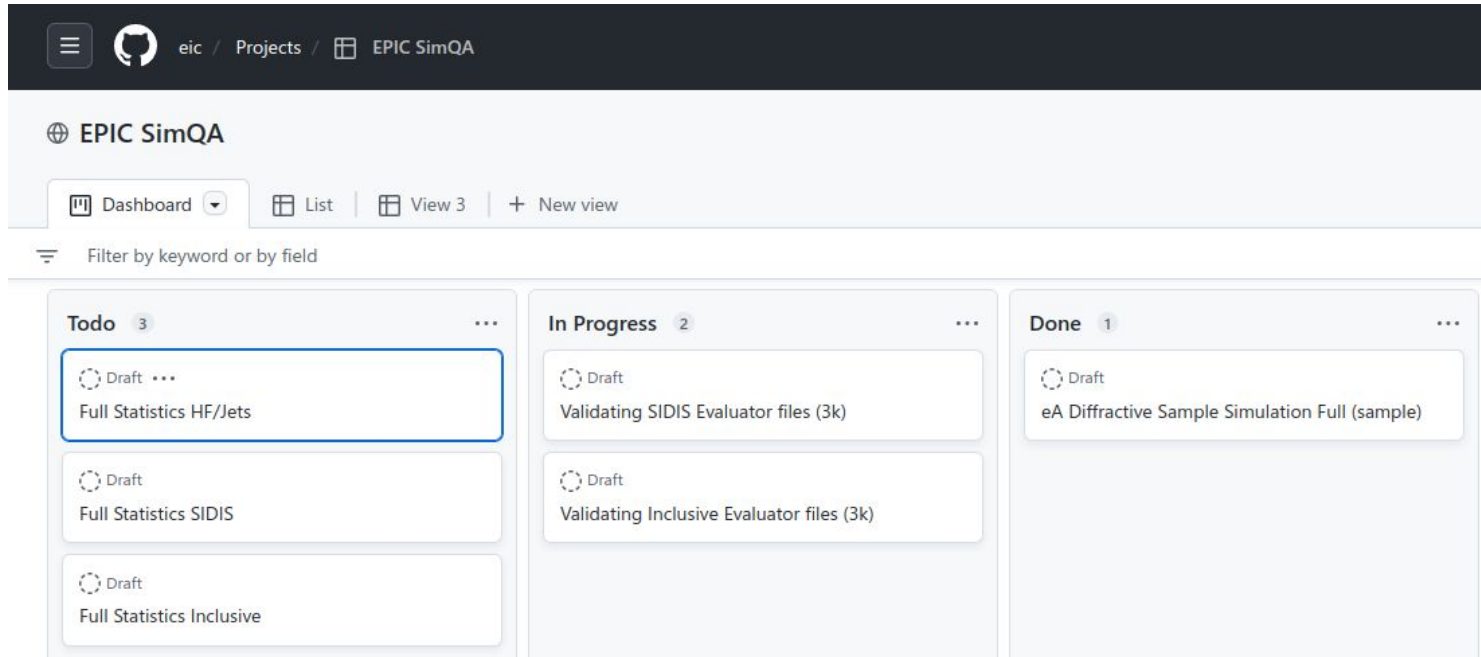
GitHub Discussions

- EIC-wide question and answer, discussion forum, etc...

The screenshot displays the GitHub Discussions interface for the 'Electron-Ion Collider (EIC) Software' repository. At the top, there are navigation tabs: Overview, Repositories, Discussions (selected), Projects, Packages, Teams, People, and Settings. Below the tabs is a blue header banner with a grid of discussion icons and a 'New discussion' button. The main content area shows a search bar and a list of discussions. The first discussion is a Q&A post by user 'wdconinc' titled 'What is the difference between `GeneratedParticles` and `ReconstructedParticles`?'. The interface also includes a sidebar with categories like 'View all discussions', 'Announcements', 'General', 'Ideas', 'Polls', 'Q&A', and 'Show and tell'. On the right, there is a 'Most helpful' section showing the user 'wdconinc' and links to 'Community guidelines' and 'github.com/orgs/eic/discussions'.

GitHub Projects

- Kanban task tracking boards, e.g. [EPIC SimQA](#) working group



The screenshot shows a GitHub Project Kanban board for the 'EPIC SimQA' project. The interface includes a navigation bar with the GitHub logo, the user 'eic', and the project name 'EPIC SimQA'. Below the navigation bar, the project name 'EPIC SimQA' is displayed with a globe icon. A view selector shows 'Dashboard' selected, with options for 'List', 'View 3', and '+ New view'. A filter bar allows filtering by keyword or by field. The Kanban board is divided into three columns: 'Todo' (3 items), 'In Progress' (2 items), and 'Done' (1 item). Each item is a 'Draft' card with a title and a description.

Column	Item Count	Item Title	Item Description
Todo	3	Draft	Full Statistics HF/Jets
Todo	3	Draft	Full Statistics SIDIS
Todo	3	Draft	Full Statistics Inclusive
In Progress	2	Draft	Validating SIDIS Evaluator files (3k)
In Progress	2	Draft	Validating Inclusive Evaluator files (3k)
Done	1	Draft	eA Diffractive Sample Simulation Full (sample)

User Incentives for Collaboration on GitHub

GitHub functions as an **extension of the resume** for students and researchers interested in research software engineering.

- Self-hosted GitLab repositories leave this information hidden from potential employers.

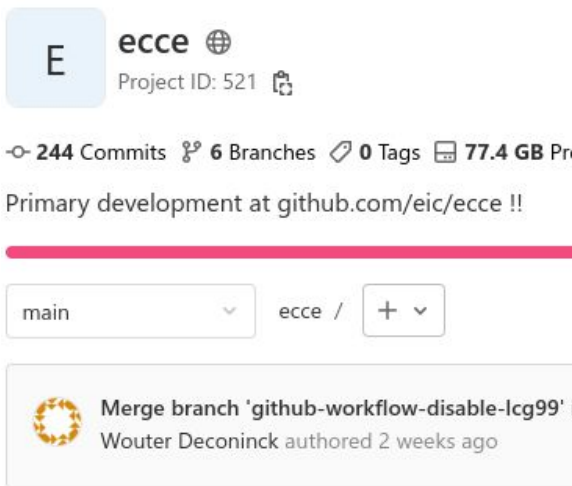
Visibility of existing software in a single organization accessible to everyone opens up to contributions from outsiders who wish to build on our projects.

The screenshot shows the GitHub profile of Christopher Dilks. At the top, there are navigation tabs for Overview, Repositories (61), Projects, Packages, and Stars (21). The profile picture is a circular geometric pattern. Below it, the name 'Christopher Dilks' and handle 'c-dilks' are shown, along with a 'Follow' button. His bio is 'High Energy Physics Postdoc' with 4 followers and 5 following. He is affiliated with 'Duke University and @JeffersonLab'. The 'Achievements' section shows '3x' for 'YoLo' and 'Beta' for 'Send feedback'. The 'Organizations' section lists 'JeffersonLab'. The 'Pinned' section features four repositories: 'dispin' (C++, 1 fork), 'diagrams' (HTML), 'clasqa' (Groovy, 1 fork), and 'sidis-eic' (C++, 3 forks). The '1,234 contributions in the last year' section includes a calendar heatmap and a radar chart. The radar chart shows 76% for Commits, 1% for Code review, and 9% for Issues.

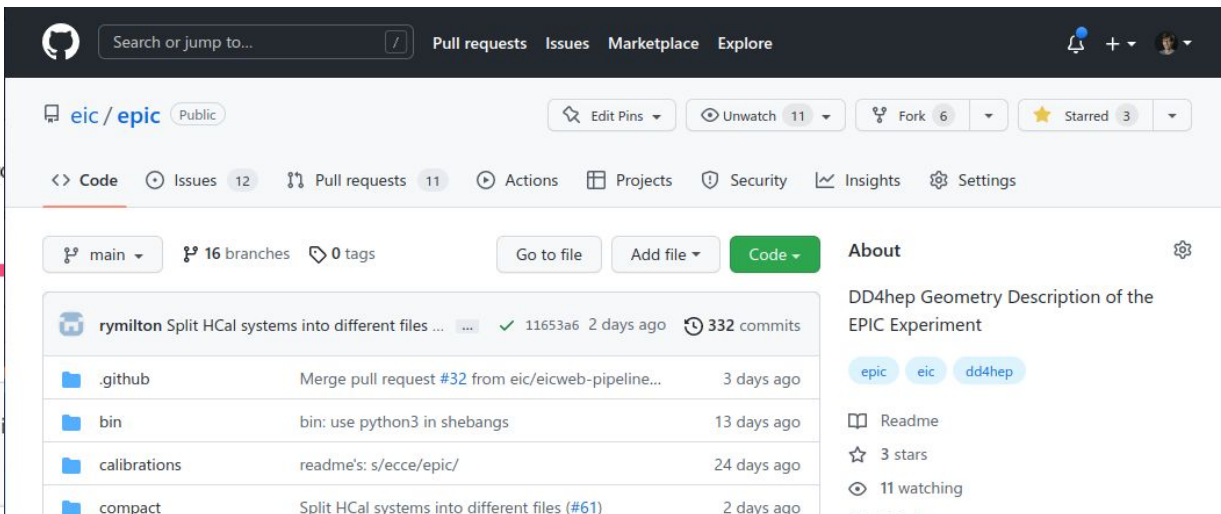
GitHub Migration: Geometry Repositories

Move completed from eicweb to [eic/epic](#) and [eic/ip6](#) on GitHub

- Primary geometry development now happening completely on GitHub
- Regular weekly developer meetings: review of issues, PRs, and WIPs
- All merge requests require passing both GitHub and eicweb CI pipelines



The screenshot shows the GitHub repository page for 'eic/ece'. The repository is public and has 244 commits, 6 branches, 0 tags, and a size of 77.4 GB. The primary development is at [github.com/eic/ece](#). The current branch is 'main'. A merge request is visible: 'Merge branch 'github-workflow-disable-lcg99'' by Wouter Deconinck, authored 2 weeks ago.



The screenshot shows the GitHub repository page for 'eic/epic'. The repository is public and has 12 issues, 11 pull requests, and 332 commits. The current branch is 'main'. The repository description is 'DD4hep Geometry Description of the EPIC Experiment'. The repository has 3 stars and 11 watchers. A commit by rymilton is highlighted: 'Split HCal systems into different files ...' with commit hash 11653a6, made 2 days ago.

File	Commit Message	Time
.github	Merge pull request #32 from eic/eicweb-pipeline...	3 days ago
bin	bin: use python3 in shebangs	13 days ago
calibrations	readme's: s/eic/epic/	24 days ago
compact	Split HCal systems into different files (#61)	2 days ago

GitHub Pipelines

- EPIC: ~50+ jobs on GitHub
- IP6: ~10+ jobs on GitHub
- Downstream detector, reconstruction, and physics benchmarks on eicweb
- CI must respond promptly
- Community-supported GitHub Actions infrastructure:
 - [eic/trigger-gitlab-ci](#)
 - [eic/run-cvmfs-osg-eic-shell](#)
 - [AIDAsoft/run-lcg-view](#)
 - [cvmfs-contrib/github-action-cvmfs](#)

Triggered via push 3 days ago	Status	Total duration	Artifacts
veprbl pushed 2325756 main	Success	54m 19s	18

Triggered via push 16 days ago	Status	Total duration	Artifacts
wdconinc pushed 4ed9578 master	Success	1h 28m 40s	5

linux-eic-shell.yml
on: push

Add more commits by pushing to the `dr1ch-opt1cs-7-27` branch on `eic/epic`.

All checks have passed
52 successful checks [Hide all checks](#)

linux-eic-shell / dawn-view-slices (view15, 1100) (pull_request)	Successful in 2m	Details
linux-eic-shell / dawn-view-slices (view15, 1300) (pull_request)	Successful in 2m	Details
linux-eic-shell / dawn-view-slices (view15, 1500) (pull_request)	Successful in 3m	Details
linux-eic-shell / dawn-view-slices (view15, 1700) (pull_request)	Successful in 2m	Details
linux-eic-shell / dawn-view-slices (view15, 1900) (pull_request)	Successful in 2m	Details
eicweb/detector_benchmarks — The detector benchmarks succeeded!		Details

Summary

The software infrastructure will use a **hybrid solution** that combines the benefits of public and accessible **code repositories on GitHub** with powerful and scalable backends with **self-hosted GitLab servers for continuous integration**.

The transition of repositories and software projects from GitLab servers used in the past is well underway, and proceeding on schedule towards completion by October.

Implementation of integration of GitHub continuous integration with self-hosted GitLab servers is proceeding ahead of schedule.

Self-hosted GitLab server eicweb can be operated under MOU with EIC project, and EIC host laboratories could contribute similar infrastructure in the future.