



Dec 19, 2025 | PBK Graduation

Introducing the Kokkos Community Playbook

Damien Lebrun-Grandie (ORNL)
Luc Berger-Vergiat (SNL)



U.S. DEPARTMENT
of **ENERGY**

ORNL IS MANAGED BY UT-BATTELLE LLC
FOR THE US DEPARTMENT OF ENERGY

FRONTIER



Acknowledgments

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Next-Generation Scientific Software Technologies program, under contract number DE-AC05-00OR22725.

This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.



The purpose

Goal: To provide a comprehensive handbook for developers and streamline the onboarding process for aspiring new contributors

Use case summary

This playbook serves as the "human infrastructure" guide for a project that has scaled from a localized team to a global effort across US DOE labs, and collaborators in Europe.

Audience

We expect this playbook to be used by aspiring new contributors looking to make their first commit, as well as existing developers across our partner institutions (SNL, ORNL, CEA, ANL, CSCS, LANL, LBNL, ULiege, UTexas) to ensure alignment

Challenges I hope my playbook will help solve:

- **Norm setting:** Establishing consistent coding and communication standards across different institutional cultures.
- **Scaling:** Managing the growth from a single-site team to an international collaboration
- **Transparency:** Making decision-making and contribution pathways visible to everyone, regardless of their location.
- **Smooth collaboration across structural barriers:** Navigating the different workflows of US DOE labs and European research centers
- **Advocating for the community:** Documenting successes via blog posts and the application registry to showcase impact



The format and structure

Where the playbook will be hosted

- Publicly on kokkos.org with the source code managed on GitHub.
- Documentation utilizes Sphinx for automatic deployment upon merge via GitHub Pages.
- Includes a "development" repository for internal agendas and working group notes.

How the playbook is structured

- **Section 1: Community Foundations** (Code of Conduct, Team Directory)
- **Section 2: Development Workflow** (Coding Standards, PR & Code Review Practices)
- **Section 3: Project Operations** (Release Process and Support Policy)
- **Section 4: Governance** (Working Group Agendas and Notes)
- **Section 5: Impact** (Downstream Applications Registry)

Screenshot or content audit

- Screenshots of kokkos.org, Core and Kernels documentations

A screenshot of the "Kokkos Kernels: Portable Math Kernels" documentation page. The page title is "Kokkos Kernels: Portable Math Kernels". Below the title, there is a section for "C++ Performance Portable Math Library" which states: "Kokkos Kernels provides math kernels for dense, batched and sparse linear algebra as well as graph computations. Our interfaces and implementations are portable to major HPC architectures and based on the Kokkos programming model. The library supports vendor implementation when available and more performant than our native algorithms." Below this, there is a table titled "The Kokkos Ecosystem includes:" with columns for "Name" and "Info".

Name	Info
kokkos	Programming Model - Parallel Execution and Memory Abstraction GitHub link
kokkos-kernels	(this library) Sparse, dense, batched math kernels GitHub link
kokkos-tools	Profiling and debugging tools GitHub link
pykokkos	Provides Python bindings to the Kokkos performance portable parallel programming. GitHub link
kokkos-remote-spaces	Shared memory semantics across multiple processes GitHub link
kokkos-resilience	Resilience and Checkpointing Extensions for GitHub link

A screenshot of the left sidebar of the Kokkos documentation website. It contains a search bar and a list of navigation links: "Get Started" (with a sub-menu for Quick Start, Requirements, Integrating Kokkos into Your Project, Building From Source, Configuration Guide, Package Managers), "Programming Guide", "API References", "Tutorials and Examples", "Contributing" (with sub-menu for Documentation Templates, Developer Guides, Kokkos Planning and Testing), and "GitHub Repo".

A screenshot of the "Kokkos: The Programming Model" page. The title is "Kokkos: The Programming Model". Below the title, there is a section for "C++ Performance Portability Programming Model" which states: "Kokkos Core implements a programming model in C++ for writing performance portable applications targeting all major HPC platforms. For that purpose it provides abstractions for both parallel execution of code and data management. Kokkos is designed to target complex node architectures with N-level memory hierarchies and multiple types of execution resources. It currently can use CUDA, HIP, SYCL, HPX, OpenMP and C++ threads as backend programming models with several other backends development." Below this, there is a table titled "The Kokkos Ecosystem includes:" with columns for "Name" and "Info".

Name	Info
kokkos	(this library) Programming Model - Parallel Execution and Memory Abstraction GitHub link
kokkos-kernels	Sparse, dense, batched math kernels GitHub link
kokkos-tools	Profiling and debugging tools GitHub link
pykokkos	Provides Python bindings to the Kokkos performance portable parallel programming. GitHub link
kokkos-remote-spaces	Shared memory semantics across multiple processes GitHub link
kokkos-resilience	Resilience and Checkpointing Extensions for Kokkos GitHub link

A screenshot of the kokkos.org homepage. The header includes the "kokkos" logo, a search bar, and navigation links: "About", "Repositories", "Documentation", "Blog", and "Community". A banner for an "Upcoming Event: KUG returns to Chicago March 19-20, 2026. Submit to speak now!" is visible. Below the banner, there is a "view" section with a sub-menu for "Kernels", "Tools", and "Tutorials". The main content area features an article titled "Overview" with a sub-header "kokkos C++ Performance Portability Ecosystem is a production level solution for writing modern C++ applications in a hardware agnostic way. It is a Linux Foundation project and part of the High Performance Software Foundation (HPSF). Kokkos originated in the US Department of Energy National Labs, and was part of the Exascale Siting Project – at its time the leading effort in the US to prepare the HPC community for the next generation of computing platforms." Below the article, there is a list of organizations supporting Kokkos development: "the principal organizations supporting Kokkos development are Sandia National Laboratories, Oak Ridge National Laboratory and the French Alternative Energies and Atomic Energy Commission. The Ecosystem consists of the libraries addressing the primary concerns for developing and maintaining applications in a portable way. The main components are the Kokkos Core Programming Model, the Kokkos Kernels Math Libraries and the Kokkos Profiling and Debugging Tools."

A screenshot of the "Community" dropdown menu from the kokkos.org website. The menu items are: "Code of Conduct", "Chat", "Events", "Mailing Lists", "CEXa Tea Time", "Release Briefings", "Applications", "Team", "Developer Meetings", and "Working Groups".



Maintenance plan

Sections that will need to be updated

- **Release Process & Support Policy:** These require the most active maintenance to reflect current project cycles
- **Working Group Agendas:** Needs frequent updates to reflect active research thrusts.
- **Team Members Directory:** This will require active maintenance as developers join or move between institutions
- **Applications Registry:** Periodic review and curation of user-submitted applications to ensure the registry remains current and high-quality

How to keep things updated

- **Frequency:** A formal review will be scheduled **every 4 months** to align with the Kokkos release cadence.
- **Responsibility:** The entire development team shares responsibility, maintaining it as a "living document" similar to the API reference.



Socialization plan

Who do you need to share the playbook with?

- The **Extended Leadership Team** for high-level buy-in
- Individual **Developer Teams and Working Groups** across the DOE labs, CSCS, and CEA
- The **Community** of users and contributors to address their questions and help onboarding

What do you need from them and how will you incentivize them?

- We will ask team leads to integrate the guide into their local onboarding and use it as the "source of truth" during reviews
- We will host a dedicated **developer session** at the upcoming **Kokkos User Group meeting** to gather feedback and encourage contributions to the guide itself
- **What do you need from them?**
 - **From Developers:** We need everyone to contribute their details to the directory to improve team visibility
 - **From Users:** At the **Kokkos User Group meeting**, we will actively invite users to submit their projects to the new registry
- **Incentives:**
 - **Visibility:** Inclusion in the registry provides downstream projects with visibility on kokkos.org, potentially leading to better support and collaboration opportunities.