Oak Ridge National Laboratory

January 8th, 2026 | Los Alamos, NM

From Kernels to Standards: How Kokkos and Modern C++ Future-Proof Critical Simulation Codes

Damien Lebrun-Grandié

Computational Sciences and Engineering Division

U.S. DEPARTMENT of ENERGY

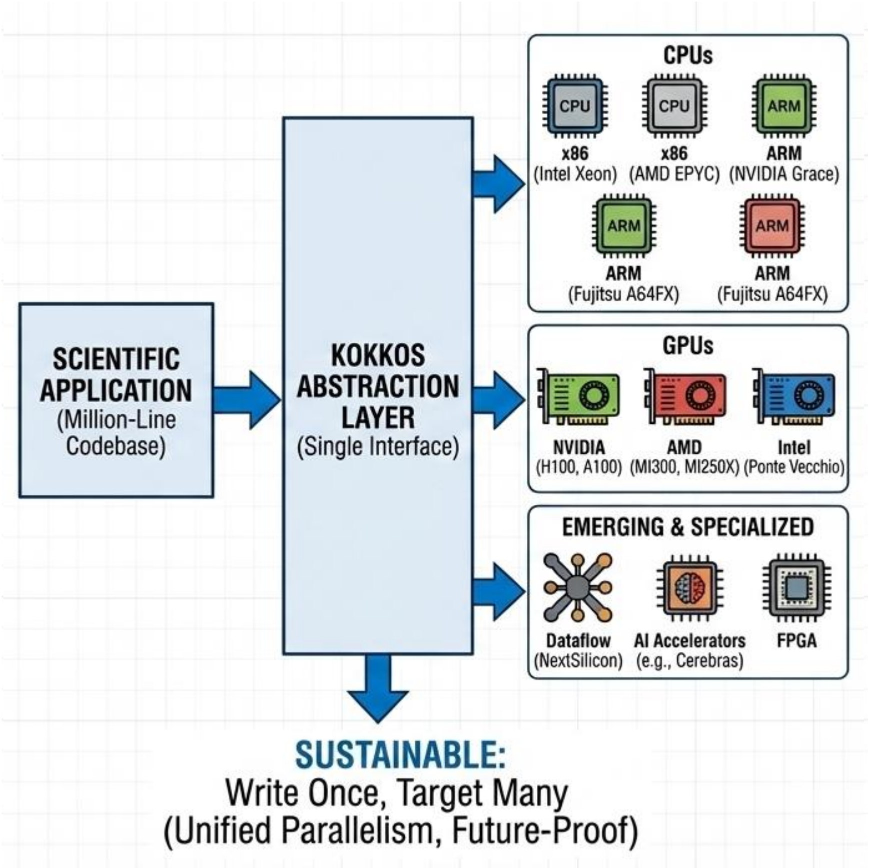
ORNL IS MANAGED BY UT-BATTELLE LLC
FOR THE US DEPARTMENT OF ENERGY

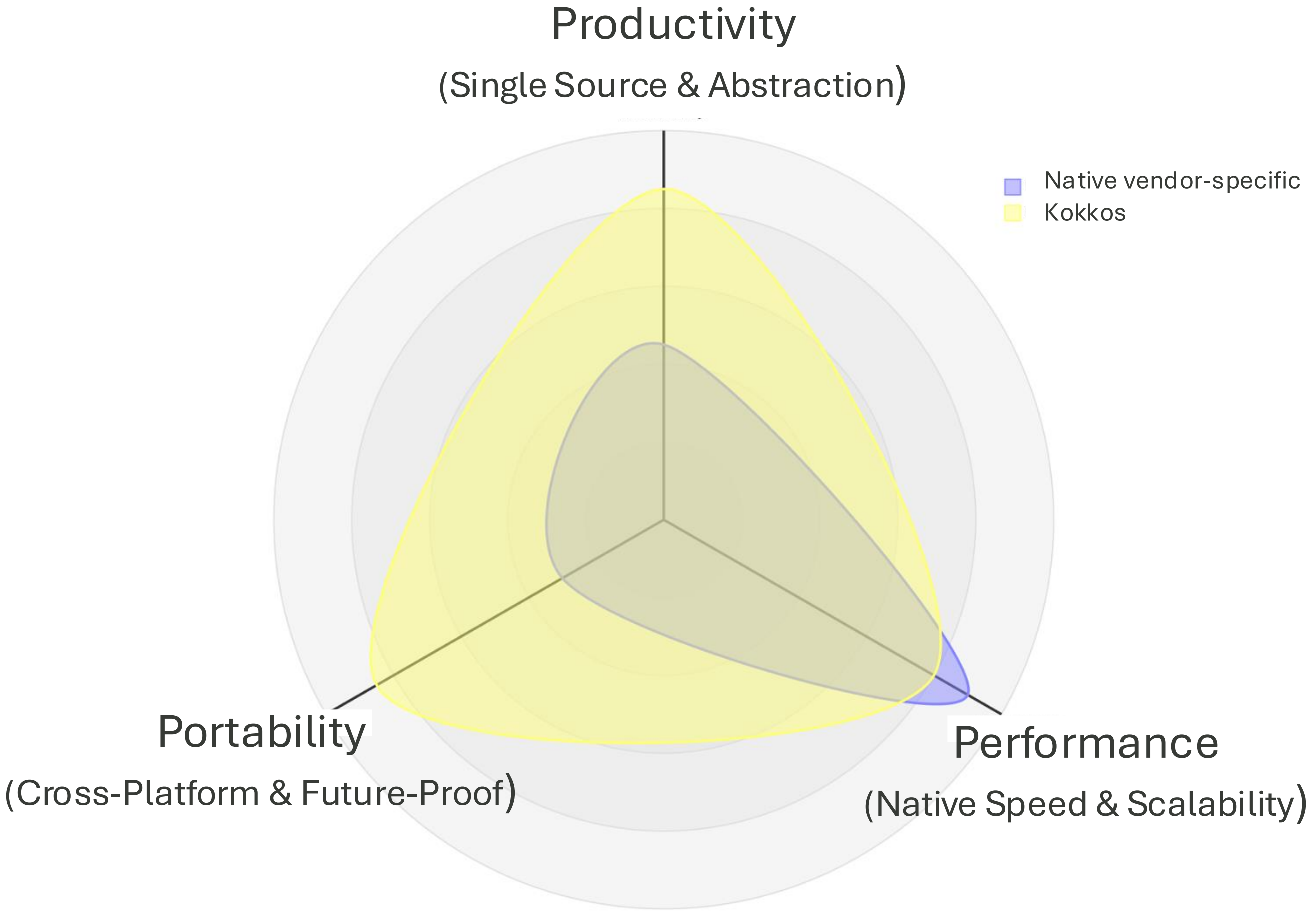
Acknowledgments

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 Exascale Era Challenge





The “P3” Challenge

Three Pillars to Tackle the Exascale Challenge

Kokkos

Standard C++

HPSF

OAK RIDGE
National Laboratory

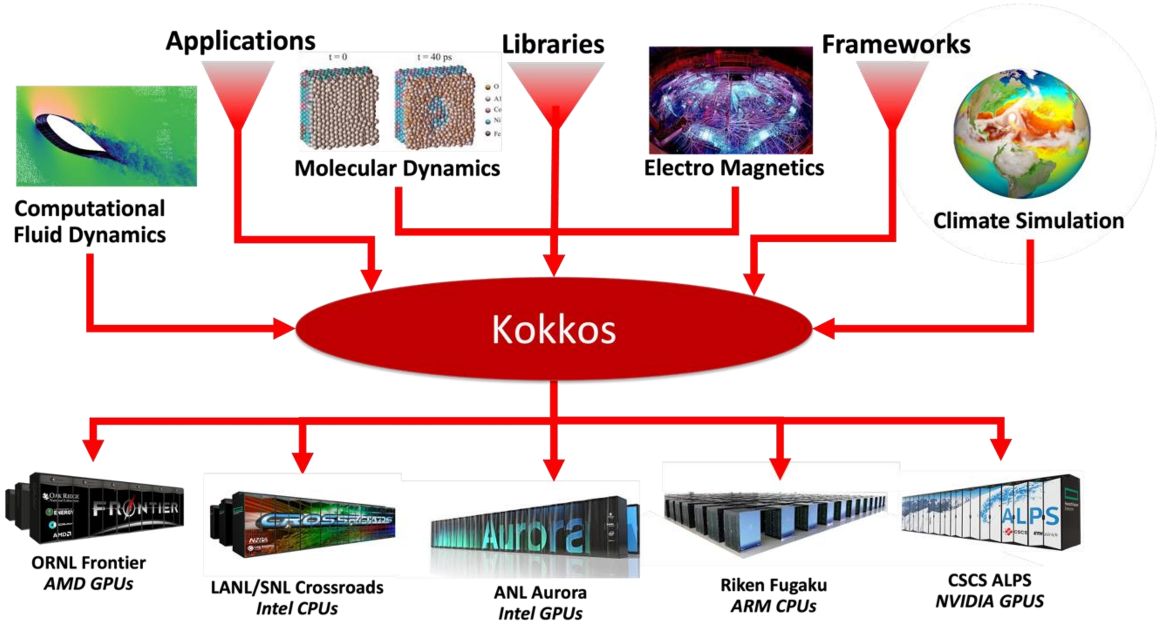
The "Kernels" – Kokkos and Performance Portability

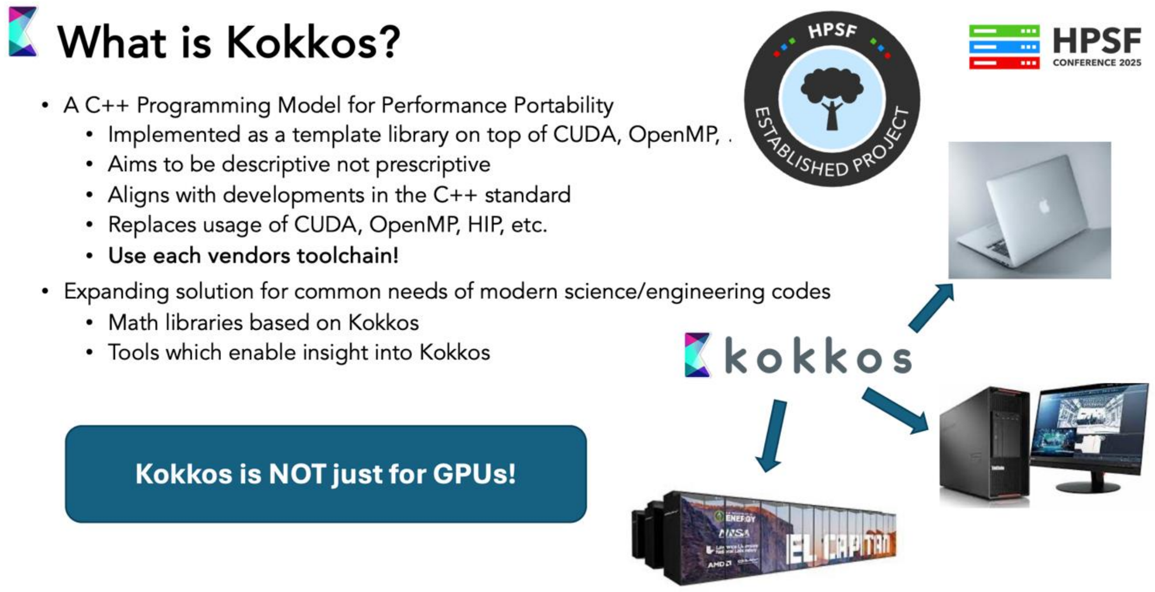
kokkos

U.S. DEPARTMENT of ENERGY

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

The Kokkos Programming Model



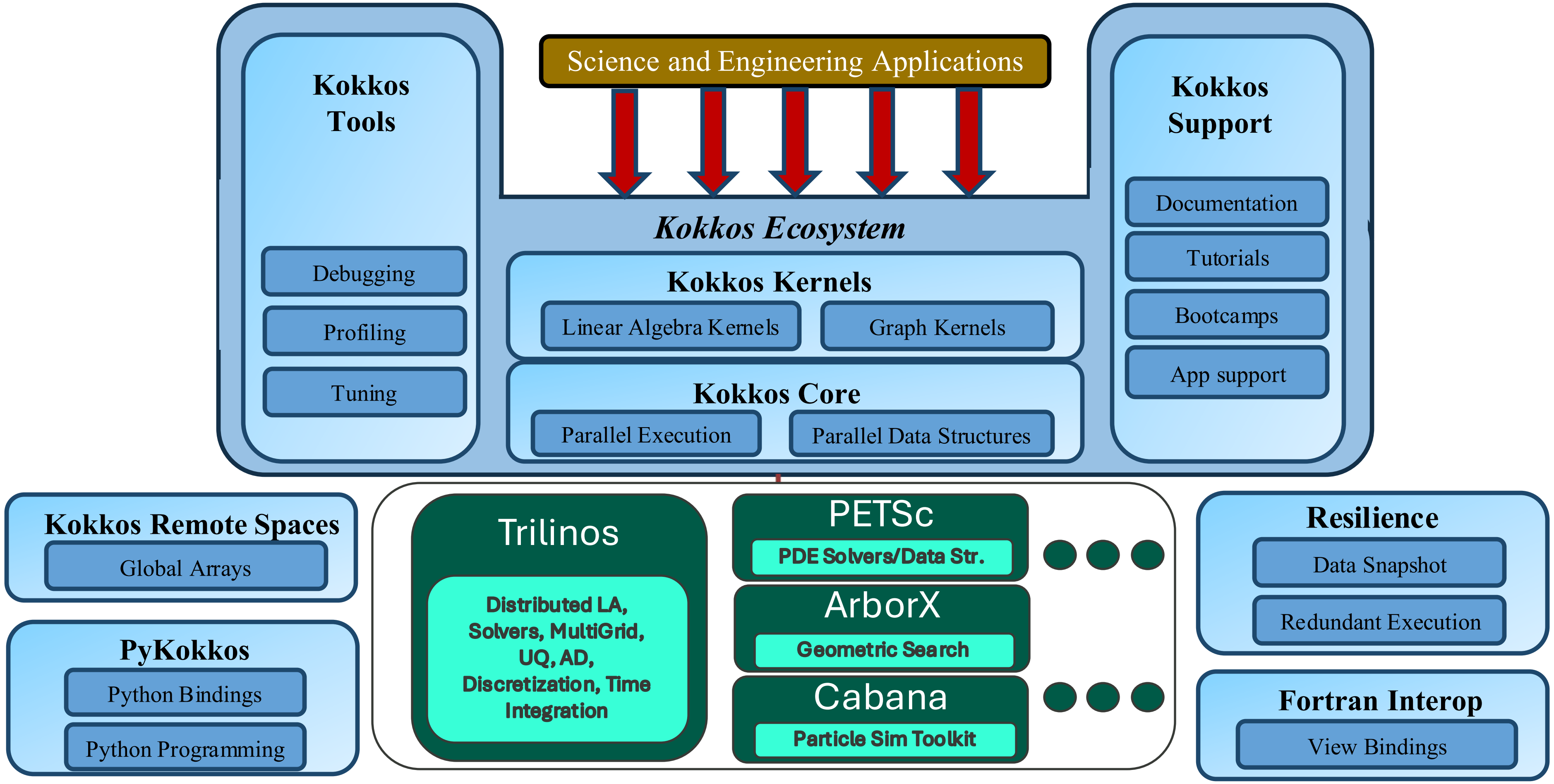


What is Kokkos?

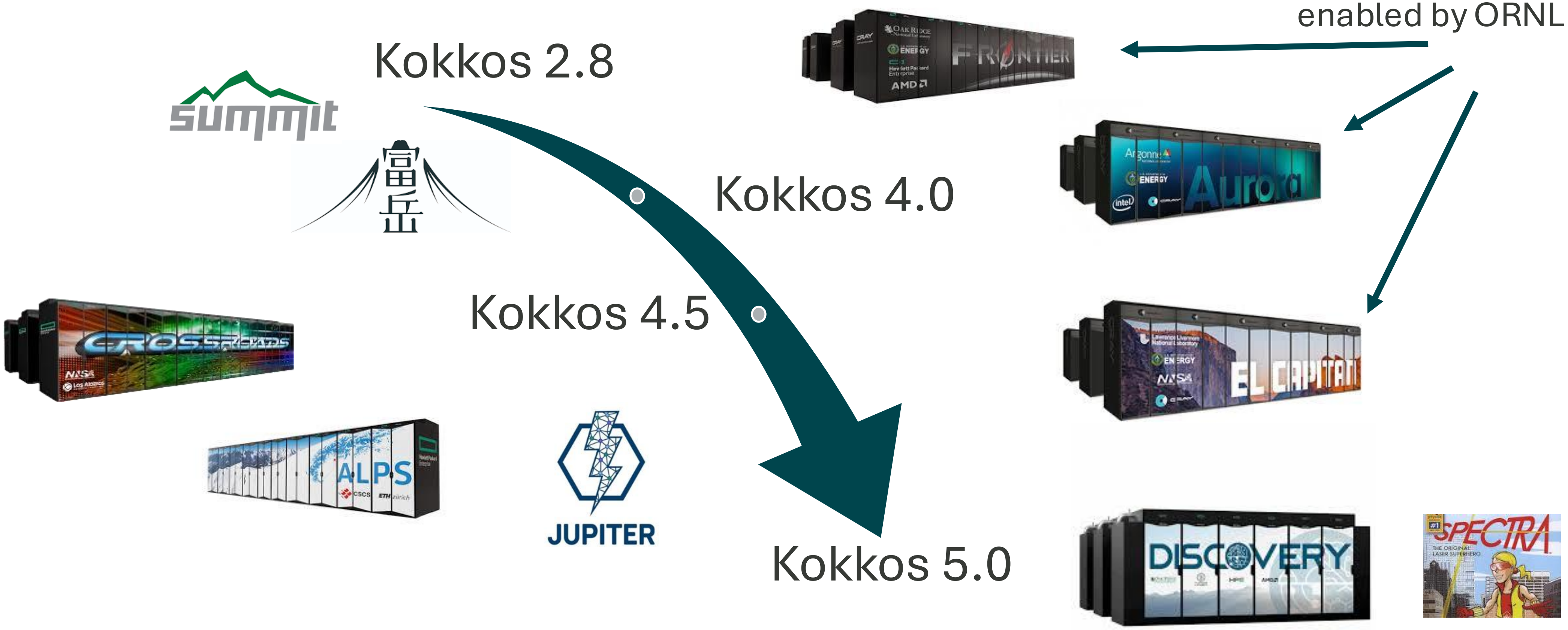
- A C++ Programming Model for Performance Portability
  - Implemented as a template library on top of CUDA, OpenMP, .
  - Aims to be descriptive not prescriptive
  - Aligns with developments in the C++ standard
  - Replaces usage of CUDA, OpenMP, HIP, etc.
  - **Use each vendors toolchain!**
- Expanding solution for common needs of modern science/engineering codes
  - Math libraries based on Kokkos
  - Tools which enable insight into Kokkos

Kokkos is NOT just for GPUs!

More Than Just a Programming Model: A Comprehensive Ecosystem



Powering the DOE Exascale Trilogy



Kokkos 2.8

Kokkos 4.0

Kokkos 4.5

Kokkos 5.0

AMD and Intel GPUs support enabled by ORNL

Kokkos at ORNL: Driving Mission-Critical Science

Enabling Libraries

- Geometric Search (ArborX)
- Particle-based Simulations (Cabana)
- Particle-In-Cell (Picasso)
- Mesh Generation (Halyard)
- Parallel I/O (ADIOS)
- Performance Portability and Fortran Code Porting (YAKL)

Core Developer Team

5 Dedicated Dual-Role Experts (30%-60% Effort on Kokkos)

#1 contributor rank (last 4y)

Lead CI and Build & Packaging WGs

OLCF support

Post-ECP software stewardship funding

Powering Applications

- Fusion (VERTEX)
- Additive Manufacturing (Toucan, ExaCA, PicassoMPM)
- Fracture Mechanics (CabanaPD)
- Enrichment (Spinnaker)
- Combustion (Quilt, Grit)
- Quantum Materials (DMRG++)

We don’t just develop Kokkos; we ensure it works for ORNL’s mission-critical science

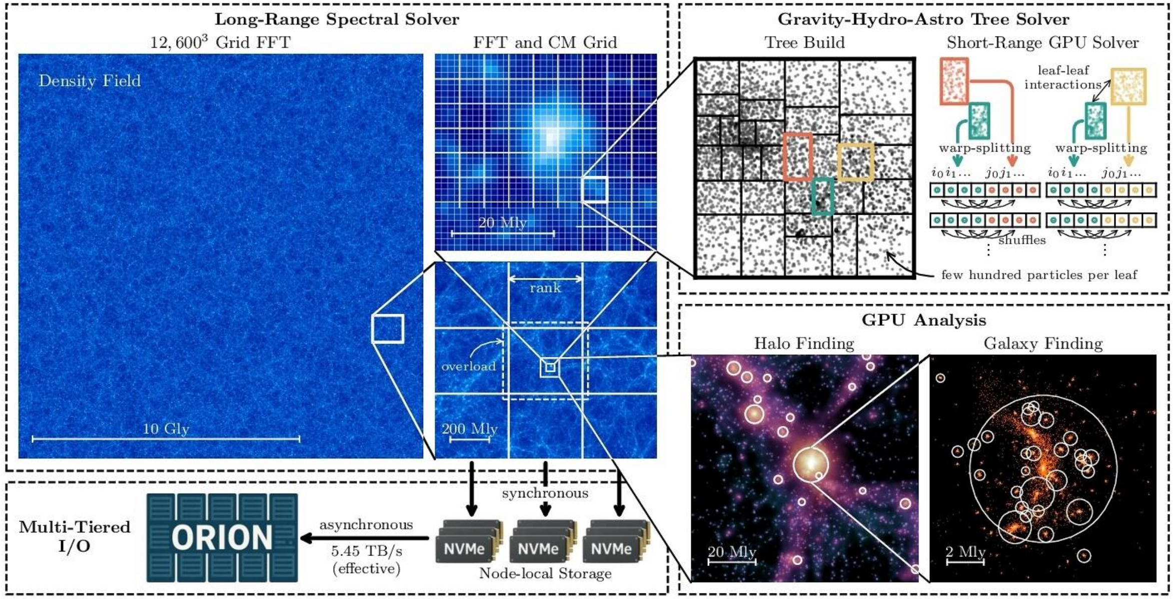
Impact Case Study: ArborX

Spatial Indexing and Clustering Library

- Provides data structures to accelerate searching for objects in space
- Includes support for k-NN, ray tracing, collision detection, particle neighbor lists , interpolation, etc.
- Co-designed experimental feature to lower the GPU occupancy

Featured in **2025 Gordon Bell Prize Finalist**
“Cosmological Hydrodynamics at Exascale: A Trillion-Particle Leap in Capability”

- Largest ever cosmological hydrodynamics full-sky simulation (> 10x state-of-art)
- Required 9000 nodes for one week on Frontier
- Achieved 513 PFLOPs peak performance
- Produced 100 PB of cosmological data



https://www.olcf.ornl.gov/2025/11/14/largest-ever-universe-simulation-up-for-supercomputings-highest-prize/
https://doi.org/10.1145/3712285.3771786

Building a Community

Developer Team

Primary Teams

Sandia National Laboratories

OAK RIDGE National Laboratory

cea

Support Efforts

Argonne NATIONAL LABORATORY

CSCS

BERKELEY LAB

Los Alamos NATIONAL LABORATORY

LIÈGE université

TEXAS The University of Texas at Austin

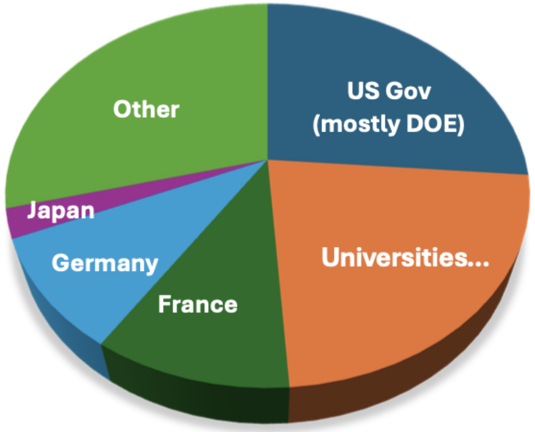
Helped us get there: ECP EXASCALE COMPUTING PROJECT

Supports us in our journey to the next level: HPSF

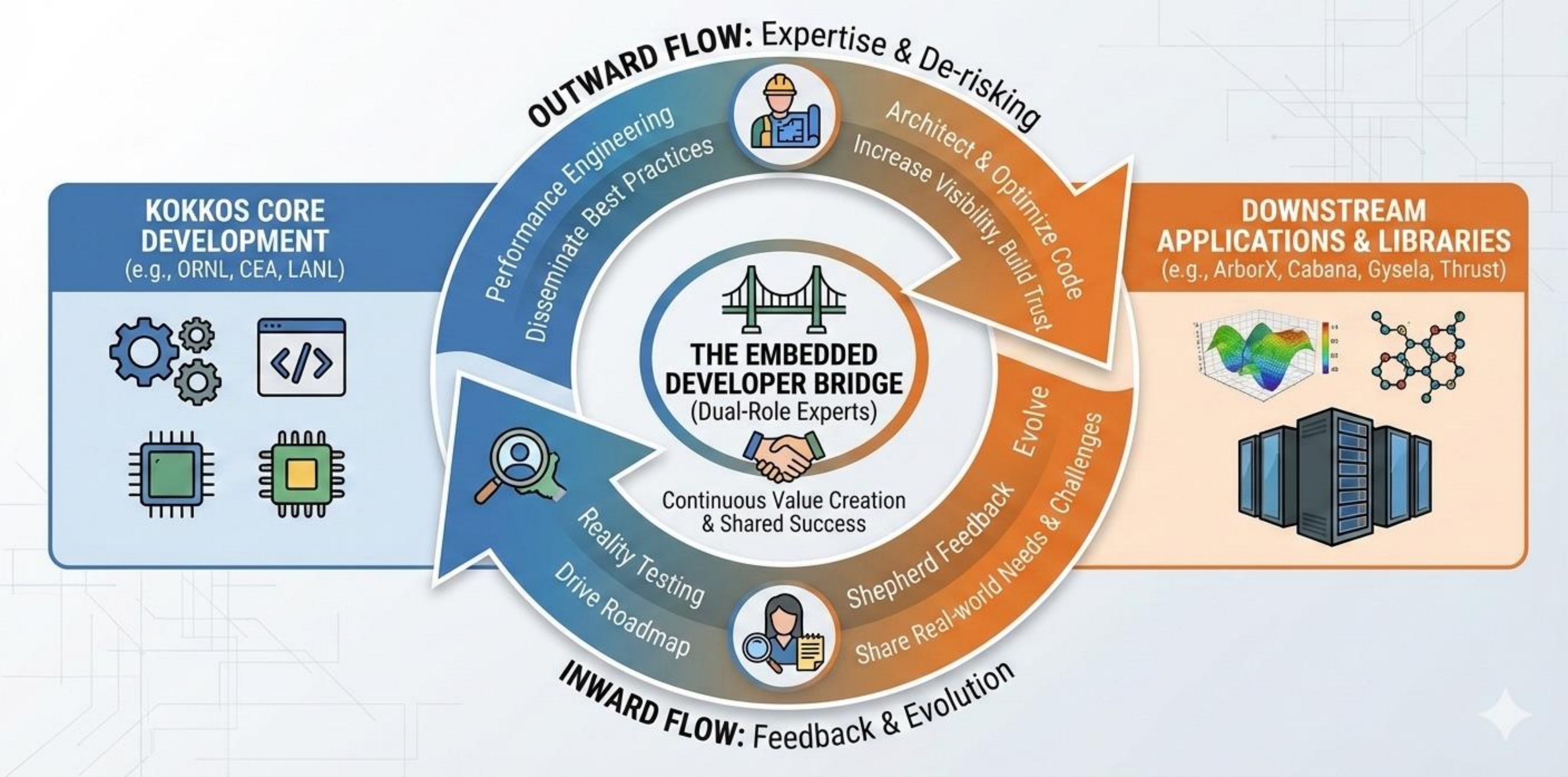
Users Community

- Thousands of users (2500 registered on Slack)
- Hundreds of institutions and projects

Slack User Origin



The Embedded Developer Model
Maximizing Kokkos Return On Investment


OUTWARD FLOW: Expertise & De-risking
KOKKOS CORE DEVELOPMENT
(e.g., ORNL, CEA, LANL)
Performance Engineering
Disseminate Best Practices
Architect & Optimize Code
Increase Visibility, Build Trust
DOWNSTREAM APPLICATIONS & LIBRARIES
(e.g., ArborX, Cabana, Gysela, Thrust)
THE EMBEDDED DEVELOPER BRIDGE
(Dual-Role Experts)
Continuous Value Creation
& Shared Success
Evolve
Shepherd Feedback
Share Real-world Needs & Challenges
Reality Testing
Drive Roadmap
INWARD FLOW: Feedback & Evolution

Hardening the Ecosystem: Production-Grade Reliability

Structural Governance: Investing in the Future

- **Dedicated Working Groups**: Established “CI” and “Build & Packaging” WGs to move beyond "ad-hoc" maintenance and toward institutionalized quality standards.
- **Reinvestment Rule:** Dedicating 10% of core effort to managing technical debt and paying down the principal
- **Long-term Stability:** Ensuring the core remains agile to be ready for what’s coming next

Rigorous Testing: Integration at Scale

- **Ecosystem Integration:** We don't test in a vacuum; we run continuous integration with downstream partners (e.g. ArborX ,Trilinos, PETSc, LAMMPS) to catch regressions before they impact the science.
- **Universal Coverage:** Continuous testing across the full landscape:
  - On-Prem: Sandia and ORNL internal resources
  - Leadership Facilities: OLCF, ALCF, and NERSC systems

Advanced Tooling: Detecting the Invisible

- **Static Analysis:** Developing Clang-Tidy extensions for invalid code detection and automated migration.
- **Dynamic Safety:** Kokkos Tools recently resolved critical "View of Views" hangs in production code.
- **The Next Frontier:** Increasing investment in tooling to move from bug detection to automated prevention.

OAK RIDGE
National Laboratory

The "Standards" – Bridging to ISO C++

The Strategy of Language Evolution

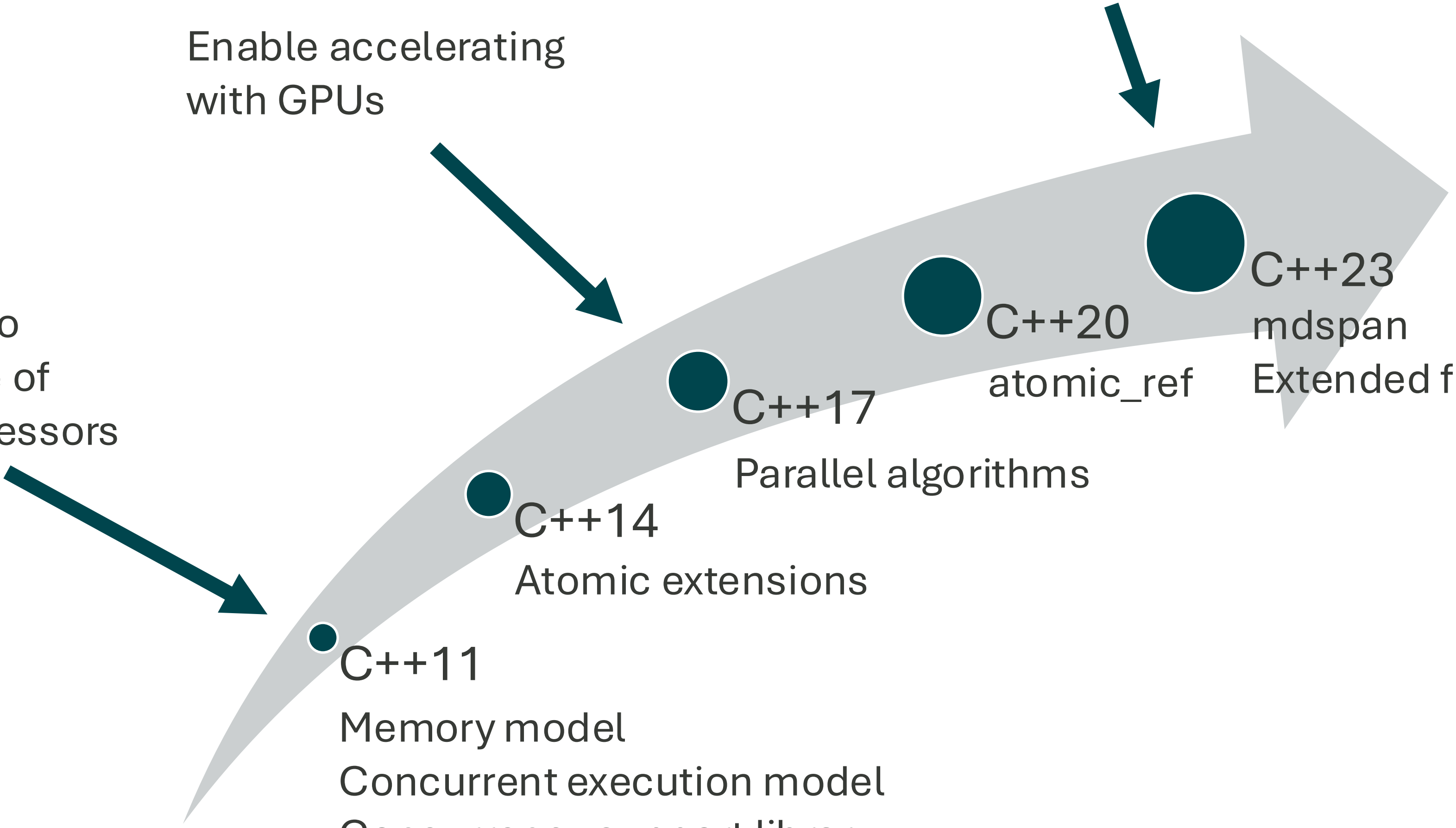
The Problem with Libraries: Even the best libraries represent a dependency. Long-lived codes (30+ years) require foundational stability.

Benefits of Standardization:

- Vendor Neutrality: Compilers (GCC, Clang, MSVC, Intel, NVIDIA) must support the standard
- Interoperability: Different libraries (e.g., solvers, I/O, viz) can communicate via standard types rather than custom wrappers.
- Reduced Technical Debt: Lowering the barrier to entry for new developers by using standard C++ features taught in universities.

Role as DOE lab representative: Ensuring that the unique constraints of HPC are core to C++'s evolution

HPC Features by Standards



Standard way to take advantage of multicore processors

Enable accelerating with GPUs

Multidimensional arrays

C++11

Memory model

Concurrent execution model

Concurrency support library

C++14

Atomic extensions

C++17

Parallel algorithms

C++20

atomic_ref

C++23

mdspan

Extended floating-point types

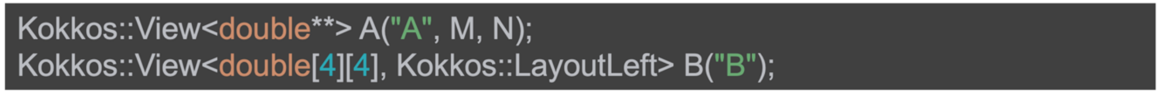
C++26

Basic linear algebra algos

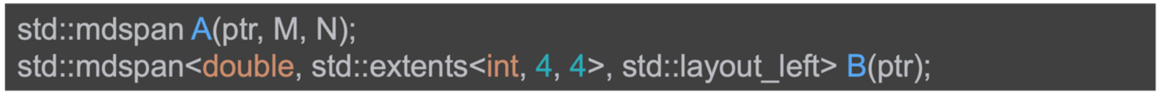
Data-parallel types (SIMD)

Execution control library

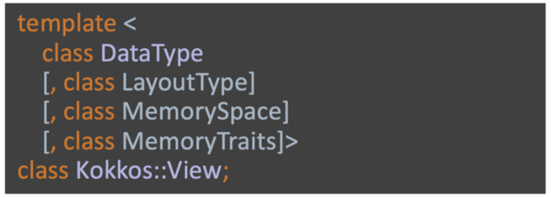
Success Story: `mdspan` (Multi-dimensional arrays in C++23)



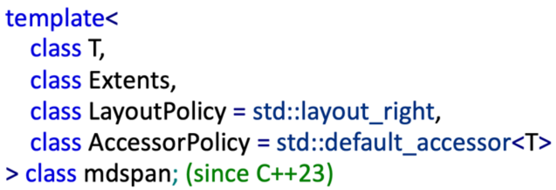
```
Kokkos::View<double**> A("A", M, N);
Kokkos::View<double[4][4], Kokkos::LayoutLeft> B("B");
```



```
std::mdspan A(ptr, M, N);
std::mdspan<double, std::extents<int, 4, 4>, std::layout_left> B(ptr);
```



```
template <
   class DataType
   [, class LayoutType]
   [, class MemorySpace]
   [, class MemoryTraits]>
class Kokkos::View;
```



```
template<
   class T,
   class Extents,
   class LayoutPolicy = std::layout_right,
   class AccessorPolicy = std::default_accessor<T>
> class mdspan; (since C++23)
```

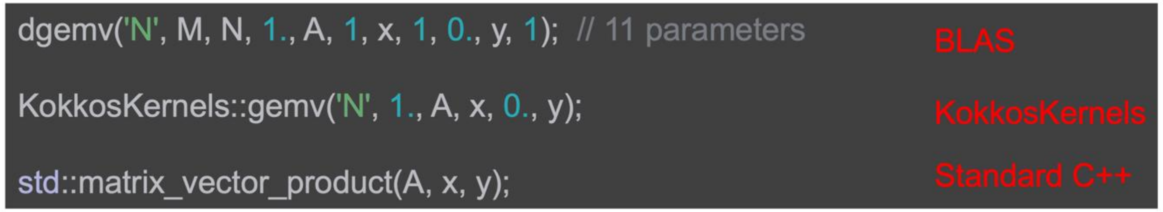
Challenge: C++ lacked a native way to represent multidimensional arrays

P0009 `mdspan`: A Non-Owning Multidimensional Array reference

Impact: Kokkos-inspired feature now allows scientific codes to share data across libraries seamlessly

kokkos/mdspan backport to C++17 deployed in NVHPC
Contributed an implementation to libc++ (LLVM)
`Kokkos::View` refactored in terms of mdspan in release 5.0

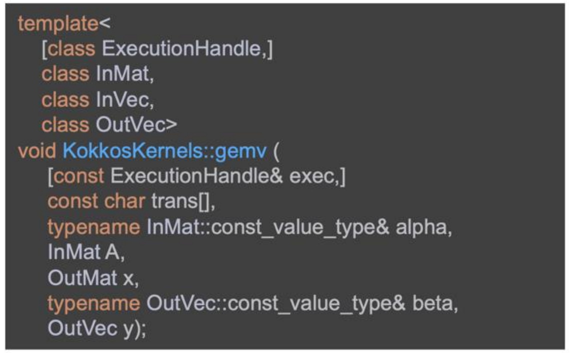
The Next Frontier: Standard Linear Algebra (C++26)



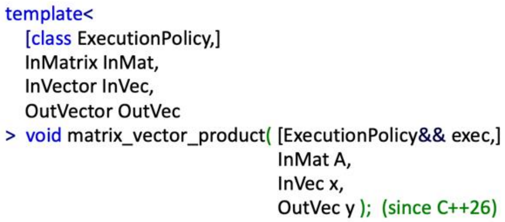
```
dgemv('N', M, N, 1., A, 1, x, 1, 0., y, 1);  // 11 parameters

KokkosKernels::gemv('N', 1., A, x, 0., y);

std::matrix_vector_product(A, x, y);
```



```
template<
   [class ExecutionHandle,]
   class InMat,
   class InVec,
   class OutVec>
void KokkosKernels::gemv (
    [const ExecutionHandle& exec,]
    const char trans[],
    typename InMat::const_value_type& alpha,
    InMat A,
    OutMat x,
    typename OutVec::const_value_type& beta,
    OutVec y);
```



```
template<
   [class ExecutionPolicy,]
   InMatrix InMat,
   InVector InVec,
   OutVector OutVec
> void matrix_vector_product( [ExecutionPolicy&& exec,]
                              InMat A,
                              InVec x,
                              OutVec y );  (since C++26)
```

Scientific computing relies on BLAS/LAPACK, but calling these often requires archaic Fortran interfaces or vendor-specific wrappers

P1673: A C++ Standard Library Interface for Parallel Linear Algebra

What it Delivers:

- A modern, type-safe interface for vector and matrix operations
- Support for execution policies (running on GPUs/Multi-core via std::execution) and mixed precision
- A common language for developers to express mathematical intent that vendors can optimize under the hood

Strategic Value: Future-proofs computationally expensive parts of simulation codes by making linear algebra a first-class citizen of the C++ language

Leading the HPC Voice in the C++ Community

Serving as "HPC Filter" in WG21: Review incoming proposals to ensure they don't inadvertently harm performance or ignore the needs of large-scale distributed systems

Community Liaison: Bridging the gap between lab scientists and the C++ community at large

- Scientific Computing Track at CppCon
- Disseminate best practices and share lessons learned from other domains (Embedded, Game dev, Trading, etc.)

Goal: Make C++ the most productive language for AI and post-exascale science without losing the "zero-overhead" performance guarantee

OAK RIDGE
National Laboratory

Strategic Future-Proofing – HPSF & Leadership

U.S. DEPARTMENT of ENERGY

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

HPSF

HIGH PERFORMANCE
SOFTWARE FOUNDATION

Beyond the Code: The High Performance Software Foundation

The Vision:

Creating a neutral, vendor-agnostic home under the Linux Foundation to host and sustain the essential software stack for global HPC.

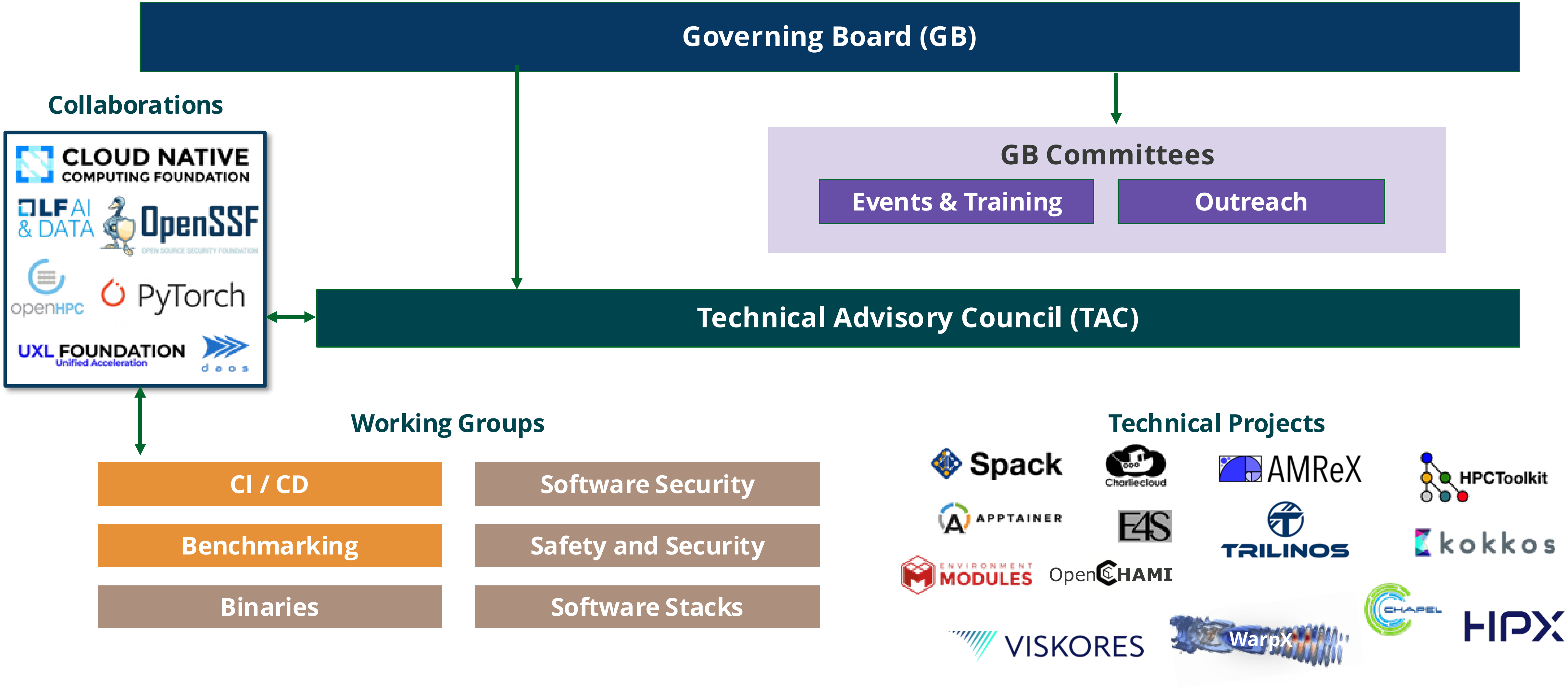
Value to the Labs:

- Ensures that critical tools have a life beyond a single funding cycle.
- Fosters a "neutral ground" for DOE, industry (AMD, NVIDIA, Intel), and international partners to collaborate.

Leadership Role:

- **Co-Founder:** Collaborated with industry and lab partners to define the foundation's core mission and governance.
- **Governing Board Member:** Driving strategic decisions on budget, policy, and cross-foundation resource allocation.
- **Technical Advisory Council (TAC):** Providing technical oversight and ensuring interoperability across flagship projects like Kokkos, Spack, Trilinos, Viskores, HPCToolkit, etc.

HPSF Charter



Governing Board (GB)

GB Committees

Events & Training

Outreach

Technical Advisory Council (TAC)

Collaborations

Working Groups

CI / CD

Software Security

Benchmarking

Safety and Security

Binaries

Software Stacks

Technical Projects

HPSF Members

Premier

aws
Lawrence Livermore National Laboratory
Hewlett Packard Enterprise
Sandia National Laboratories
Microsoft

General

NVIDIA
OAK RIDGE National Laboratory
cea
AMD
kitware Delivering Innovation
Los Alamos NATIONAL LABORATORY EST. 1943
intel
RIKEN
R-CCS
Argonne NATIONAL LABORATORY
arm

Associate

BERKELEY LAB
JÜLICH Forschungszentrum
UCL
Tennessee TECH
सी-डैक CDAC
ETH zürich
UNIVERSITY OF MARYLAND 18 56
UNIVERSITY OF OREGON
UTokyo
Universität der Bundeswehr München

What's in it for Projects?

HPSF

Vendor-Neutral Home & Stability

Ensures project independence and long-term sustainability under the Linux Foundation.

Growth, Branding & Focus

HPSF handles legal, marketing (websites, new logos), and community growth, letting your team focus on the code.

Infrastructure & Automated Testing

Access dedicated hardware and CI/CD Working Group support for robust automated testing and benchmarking.

Collaboration & Events

Host user groups at the annual HPSF Conference and facilitate multi-institutional collaboration and workforce development.

HPSF Impact: Professionalizing the Kokkos Ecosystem

Governance & Operational Maturity

- Formalized project governance and started developing a community playbook.
- Core leadership graduated from the CSCCE Training for Open-Source Leaders (birdaro 2025 program).
- Committed to the OpenSSF Best Practices Program for security and sustainability.

Engineering & Infrastructure

- Secured early access to hardware for pre-release testing, such as NVIDIA Blackwell.
- Implemented LFX Tools and Slack Pro to manage and analyze a community of 2,500+ users.
- Utilizing Sustainable Research Pathways to diversify and grow the developer pipeline.

Transparency & Engagement

- Established regular Release Briefings and recorded technical seminars.
- Institutionalized Kokkos User Group (KUG) meetings and the Tea-Time Seminar Series.

International Strategic Partnerships

Aligning Global Roadmaps: Actively bridging the gap between US, European, and Japanese HPC efforts.

DOE/MEXT (US-Japan): Leading working groups on Programming Languages and Scientific Libraries to ensure our exascale stacks remain compatible and state-of-the-art.

ADAC: Driving international standards for "Portability, Sustainability, and Integration" in large-scale codes.

Strategic Outcome: Reducing redundant development by leveraging global expertise and shared software components.

U.S. DEPARTMENT of ENERGY

文部科学省
MEXT
MINISTRY OF EDUCATION,
CULTURE, SPORTS,
SCIENCE AND TECHNOLOGY-JAPAN

EuroHPC
Joint Undertaking

Cultivating the HPC Community

Expanding the C++ Ecosystem:

- Founded and Chaired the Scientific Computing Track at CppCon
- Created a dedicated space for scientists within the world's largest C++ conference to influence the "mainstream" developer community.

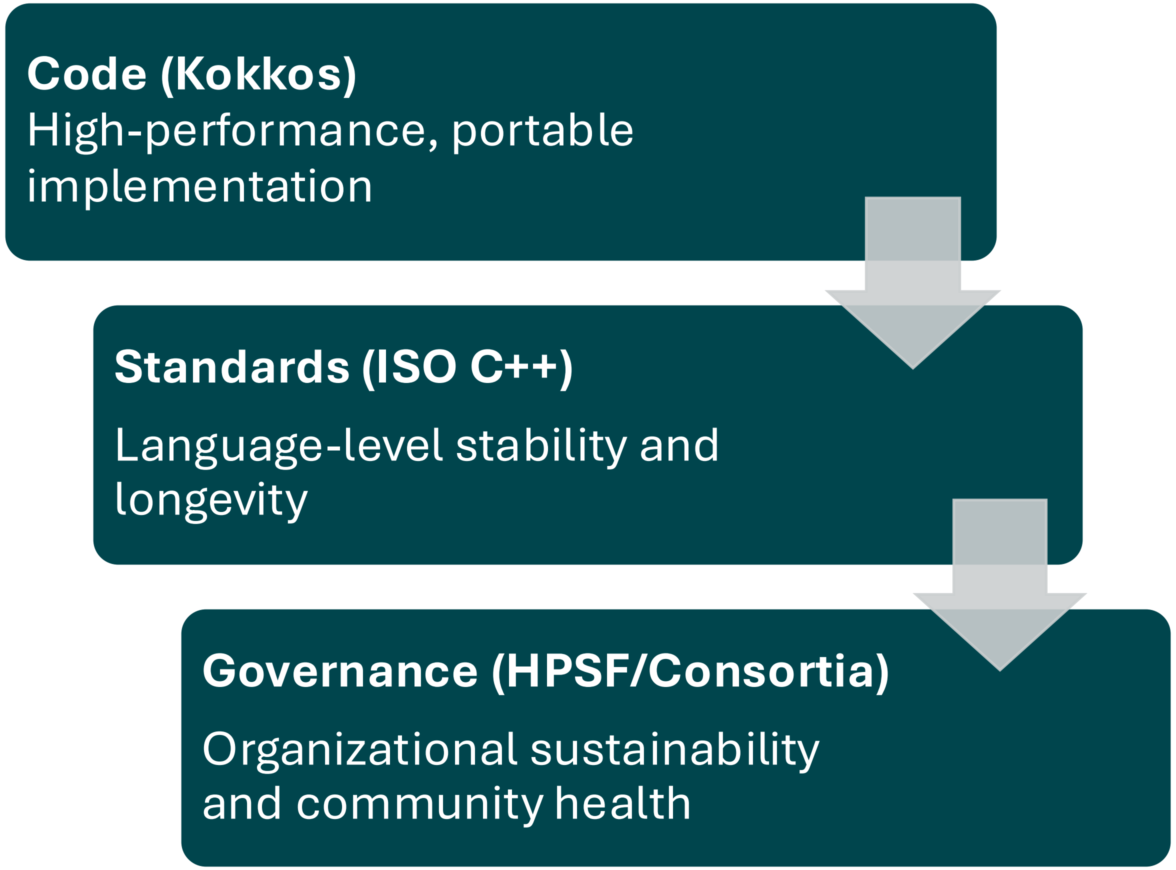
Organizational Service:

- 2nd edition of the HPSF Conference in Chicago March 19-20, 2026
- ISO C++ panel to discuss the latest development of the language and tutorials to teach how to leverage standard parallelism on heterogeneous systems (SC, ISC, HPC Asia)

Mentorship & Workforce Development:

- Engaging with application and other software teams
- Active mentorship of post-docs and junior staff
- Hosting student interns and collaborating with faculties
- Ensuring the "Performance Portability" mindset is passed down to the next generation of computational scientists

Conclusion



Code (Kokkos)
High-performance, portable implementation

Standards (ISO C++)

Language-level stability and longevity

Governance (HPSF/Consortia)

Organizational sustainability and community health

Looking Forward:

- Beyond exascale: AI-integrated HPC and specialized hardware.
- Our commitment: Continuing to bridge the gap between low-level hardware kernels and the high-level needs of the scientific mission.

Thank you – Questions?

lebrungrandt@ornl.gov