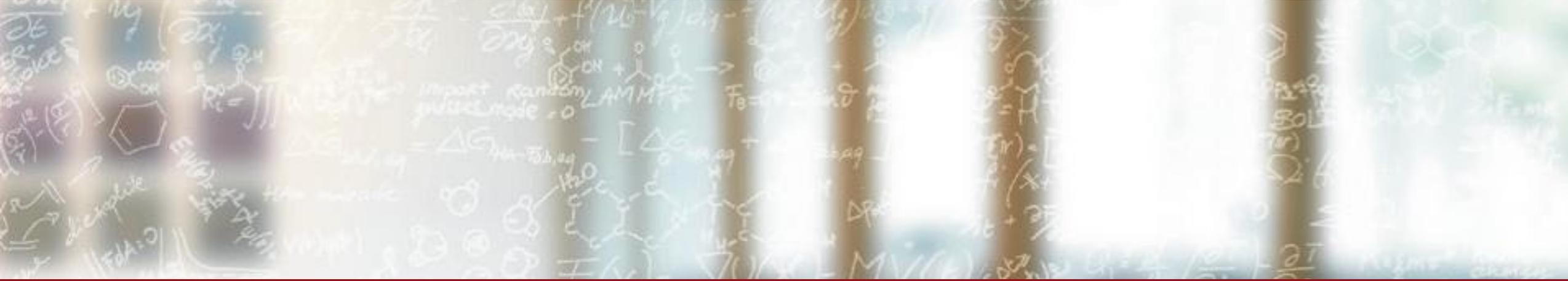




**CSCS**

Centro Svizzero di Calcolo Scientifico  
Swiss National Supercomputing Centre

**ETH** zürich



# Evolving Sarus to augment Podman for HPC on Cray EX

CUG 2025

Alberto Madonna, Gwangmu Lee, Felipe A. Cruz, ETH Zurich / CSCS

May 7th, 2025

# Table of Contents

1. Progression of HPC container engines at CSCS
2. Why are we doing this?
3. Tracing a path forward
4. Early tests
5. Conclusions

# Sarus (2019)

- Combines container portability with native HPC performance
- Supports container specialization and HPC features through OCI hooks (modular plug-in components)
- Adopts OCI-compliant standards and technologies:
  - Runtime (runc)
  - Hooks (e.g. NVIDIA Container Toolkit)
  - Image conversion tools (Skopeo, umoci)



# Container Engine (CE) (2023)

- Forward-looking solution implementing “containers as first-class elements”
  - Intended to enable containerized user environments on Alps
  - Based on significantly modified versions of NVIDIA Enroot and Pyxis software
  - Built on knowledge and insights from Sarus
- 
- Paper at CUG 2024 discussing principles, architecture, implementation, use cases.

## Containers-first user environments on HPE Cray EX

Felipe A. Cruz  
Swiss National Supercomputing Centre  
Lugano, Switzerland  
felipe.cruz@cscs.ch

Alberto Madonna  
Swiss National Supercomputing Centre  
Lugano, Switzerland  
alberto.madonna@cscs.ch

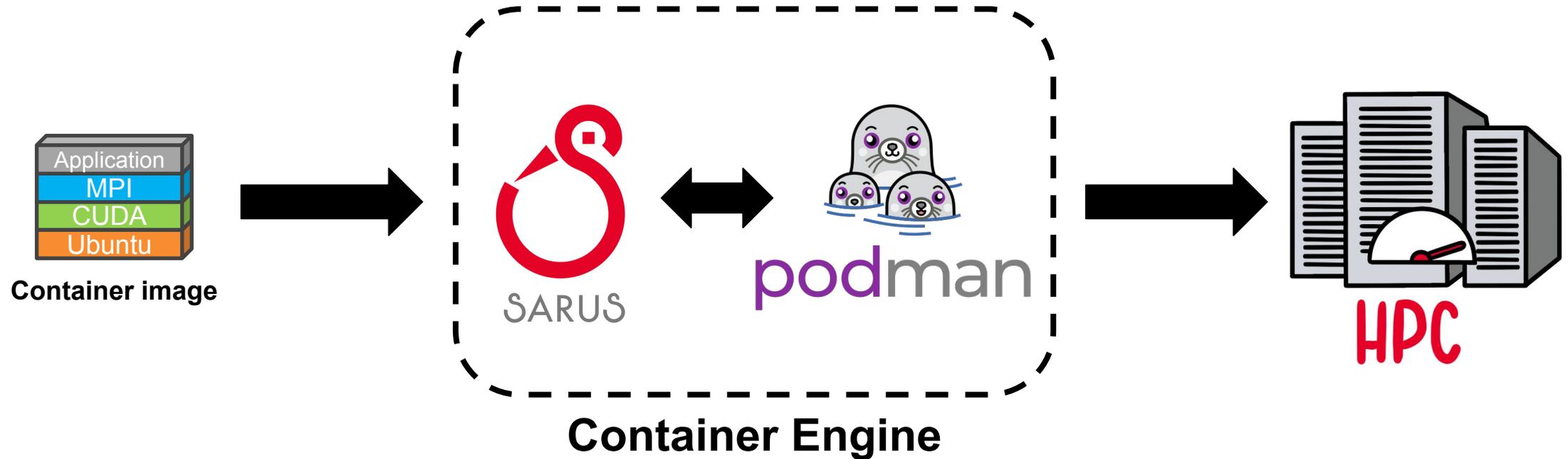
*Abstract*—In High-Performance Computing (HPC), managing the user environment is a critical and complex task. It involves composing a mix of software that includes compilers, libraries, tools, environment settings, and their respective versions, all of which depend on each other in intricate ways. Traditional approaches to managing user environments often struggle with finding a balance between stability and flexibility, especially in large systems serving diverse user needs.

This work introduces a containers-first approach for HPC, enhancing stability and flexibility in user environments by seamlessly integrating container technologies on an HPE Cray EX system. This approach evolves the user environment management and delivery, enabling customized, fast, and transparent deploy-

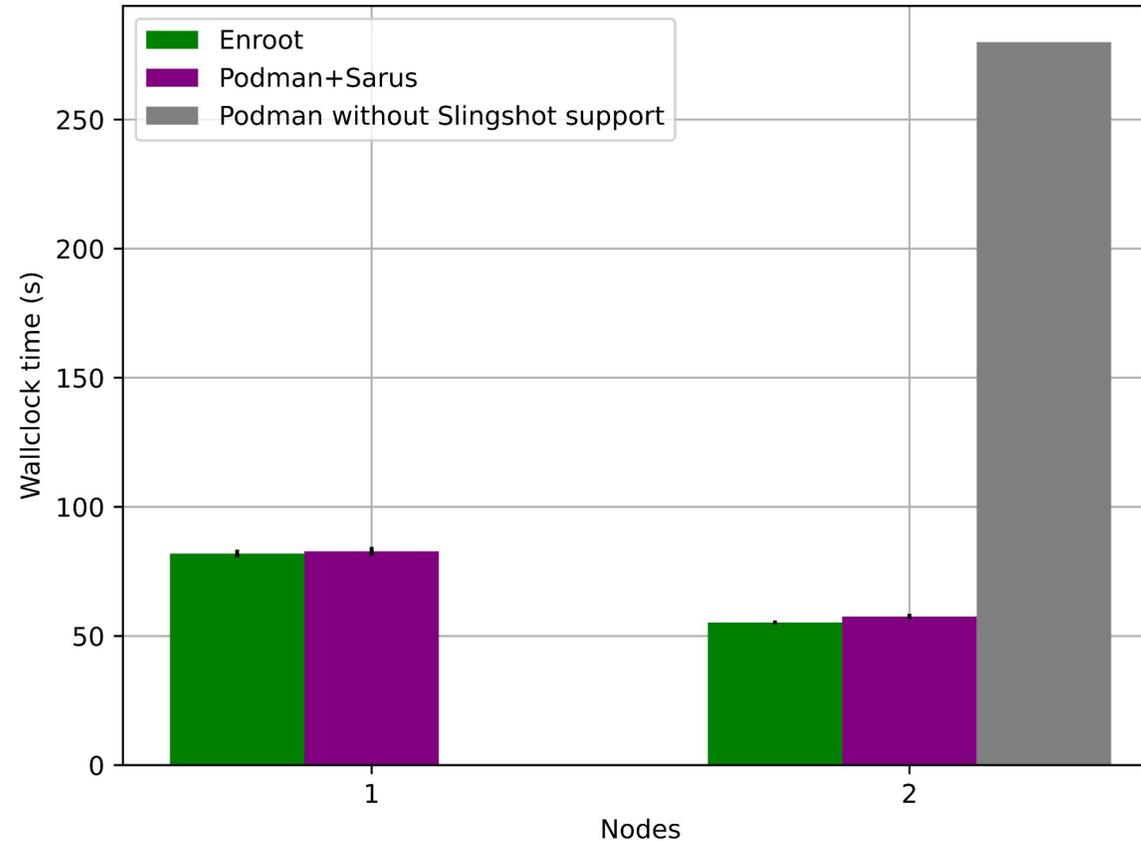
An extensive usage of containers can bring major benefits to HPC users and providers; and could address relevant problems encountered by HPC users and providers alike:

- Decouple user environment from system components: more freedom and flexibility in planning and executing system maintenances, improved robustness and consistency of the user environment as they do not change across maintenance, more informative and meaningful analysis of regressions as user-side components do not change and are reproducible.

# Outlook: Augmenting Podman for the next generation Container Engine



# PyFR (Flux Reconstruction CFD)



**Container:** PyFR 2.1, OpenMPI 5.0.7, CUDA 12.8, Ubuntu 24.04

**Test case:** 3D Taylor-Green vortex (4 ranks per node, 20 repetitions)

**System:** Alps Infrastructure - Development vCluster (2 x AMD EPYC 7713, 4 x NVIDIA A100, Slingshot 11 200Gb)



**CSCS**

Centro Svizzero di Calcolo Scientifico  
Swiss National Supercomputing Centre

**ETH** zürich

# Why are we doing this?

---

# Cloud-native landscape continues to evolve

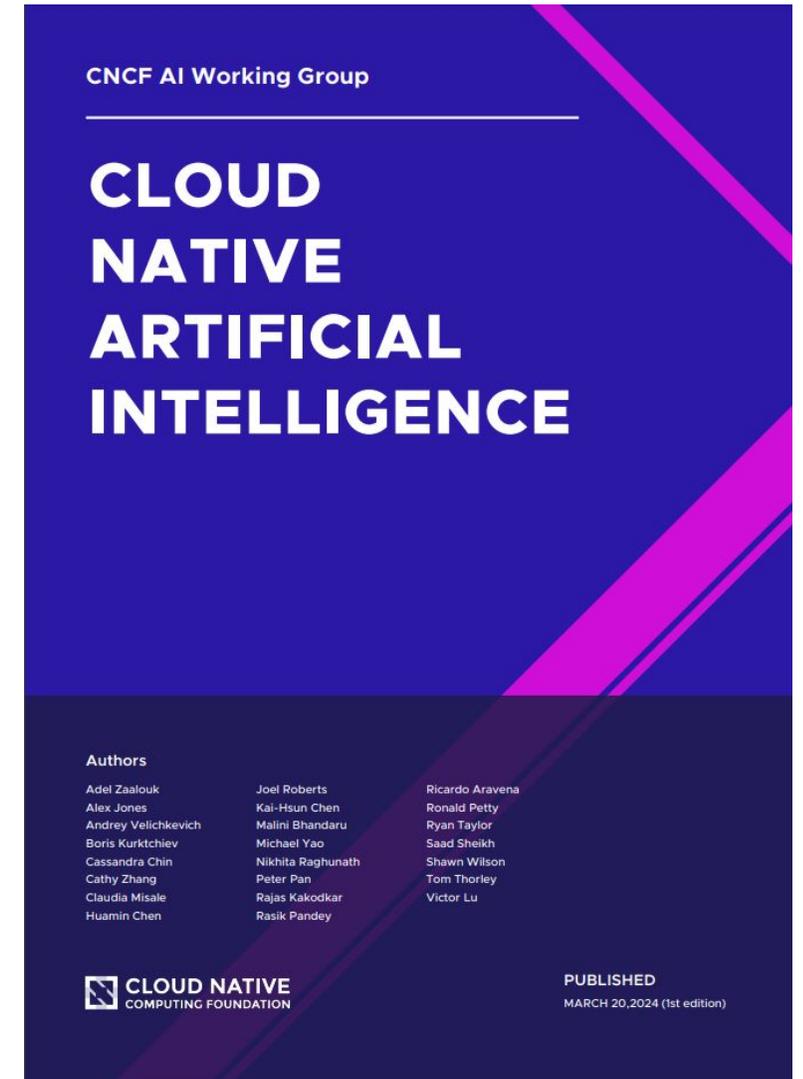
The image displays a vast collection of logos for cloud-native technologies, organized into several functional categories. Each logo is accompanied by its name and a status indicator (e.g., 'CNCF GRADUATED', 'CNCF INCUBATING').

- Application Definition & Image Build:** Includes logos for HELM, Backstage, Buildpacks.io, dapr, KubeVeto, KubeVirt, OPERATOR FRAMEWORK, CHEF, and others.
- Database:** Features logos for TiKV, Vitess, and various other database solutions.
- Continuous Integration & Delivery:** Lists tools like Argo, Flux, Keptn, and OpenTribes.
- Streaming & Messaging:** Includes CloudEvents, STRIMZI, and other messaging solutions.
- Scheduling & Orchestration:** Shows logos for KEDA, Kubernetes, Concourse, KARMADA, Knative, and VOLCANO.
- Service Mesh:** Includes Istio and LINKERD.
- Remote Procedure Call:** Features gRPC.
- Service Proxy:** Lists Envoy and Contour.
- API Gateway:** Includes Embray Ingress and others.
- Coordination & Service Discovery:** Shows CoreDNS and etcd.
- Cloud Native Storage:** Includes Rook, CubeFS, LONGHORN, and others.
- Cloud Native Network:** Features Cilium and CNI.
- Container Runtime:** Lists cri-o and Lima.
- Security & Compliance:** Includes Falco, In-toto, Keycloak, and Kyverno.
- Automation & Configuration:** Shows KubeEdge, Chef, and others.
- Container Registry:** Includes Harbor and Dragonfly.
- Key Management:** Lists Spiffe and SPIRE.
- Observability:** Features Fluentd, Prometheus, Cortex, and Thanos.
- Continuous Optimization:** Includes OpenCost and Denisy.
- Chaos Engineering:** Shows Chaos Mesh and Litmus.
- Feature Flagging:** Lists Open Feature and others.

<https://landscape.cncf.io/>

# Cloud-native evolving to enable AI

- AI is rising as a dominant cloud workload, despite remaining gaps
- Containers are essential building blocks of cloud native, and many container projects are involved
- Aligning and adapting to these innovations can open significant opportunities



[CNAI Whitepaper, March 2024](#)

# Our current software not in a position to keep up

- Sarus
  - CSCS-developed HPC container engine, in production since 2019
  - Designed primarily to deploy user-defined software stacks and run traditional HPC apps
  - Missing many interactivity and manageability features due to focus on HPC integration
- Container Engine (CE)
  - Pathfinder to demonstrate the “containers as first-class elements” approach
  - Based on NVIDIA’s Enroot + Pyxis software
    - Not officially supported products
    - No significant feature development ongoing or expected
    - No adoption of OCI standards
- Expanding scope of requirements is very demanding in terms of developers time and expertise
- We also want to consolidate work from the two efforts



**CSCS**

Centro Svizzero di Calcolo Scientifico  
Swiss National Supercomputing Centre

**ETH** zürich

# Tracing a path forward

---

# Core concepts

- Container portability + native HPC performance
- Container specialization and HPC features through modular components
- Adopt OCI and cloud-native standards and technologies
- Usability and seamless integration with WLM
- Advanced runtime
- Very actively developed and large user community
- Early access to innovative technologies and bleeding-edge developments



SARUS  
(2019)



CE  
(2023)



podman

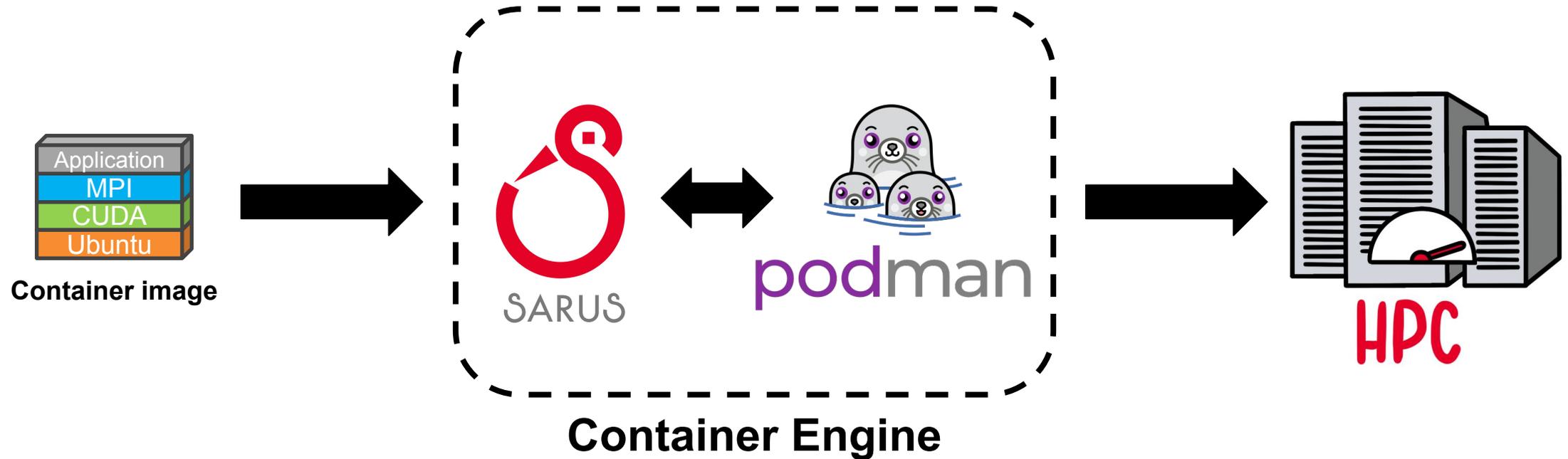


**Sarus is evolving into a collection of software to unlock the power of HPC for community tools based on open standards.**

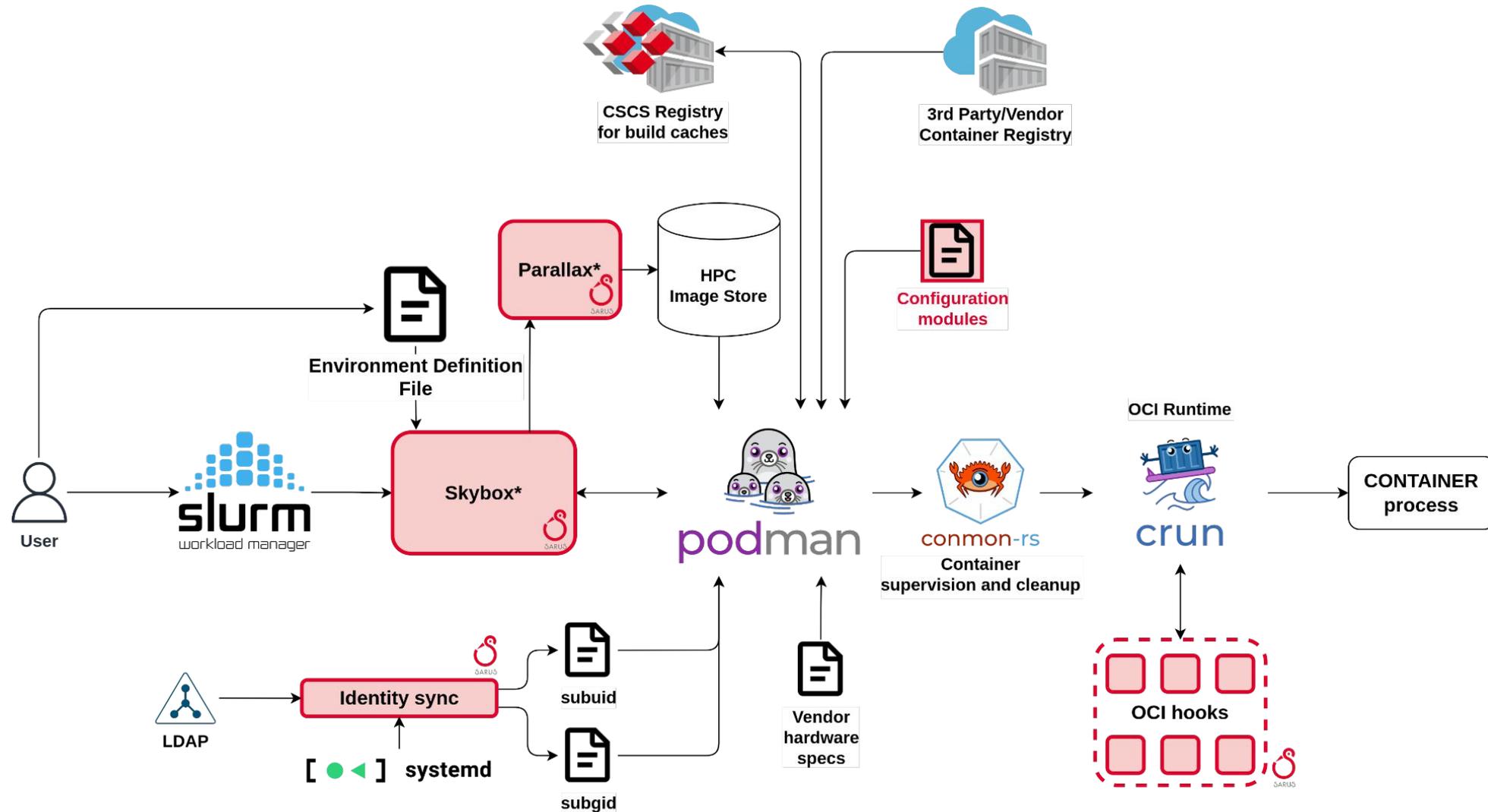
- **Sarus Hooks:** OCI hooks for HPC use cases
- **Parallax\*:** Support SquashFS image storage on parallel filesystems
- **Skybox\*:** Slurm SPANK plugin for transparent deployment of containers as job environments
- **libsarus:** Foundational C++ library for container tools and OCI hooks
- **rootless-subid-sync:** System-level utility to synchronize Linux subordinate IDs across computing clusters

\* : provisional development codenames

# Augmenting Podman for the next generation Container Engine



# Podman+Sarus Container Engine architecture





**CSCS**

Centro Svizzero di Calcolo Scientifico  
Swiss National Supercomputing Centre

**ETH** zürich

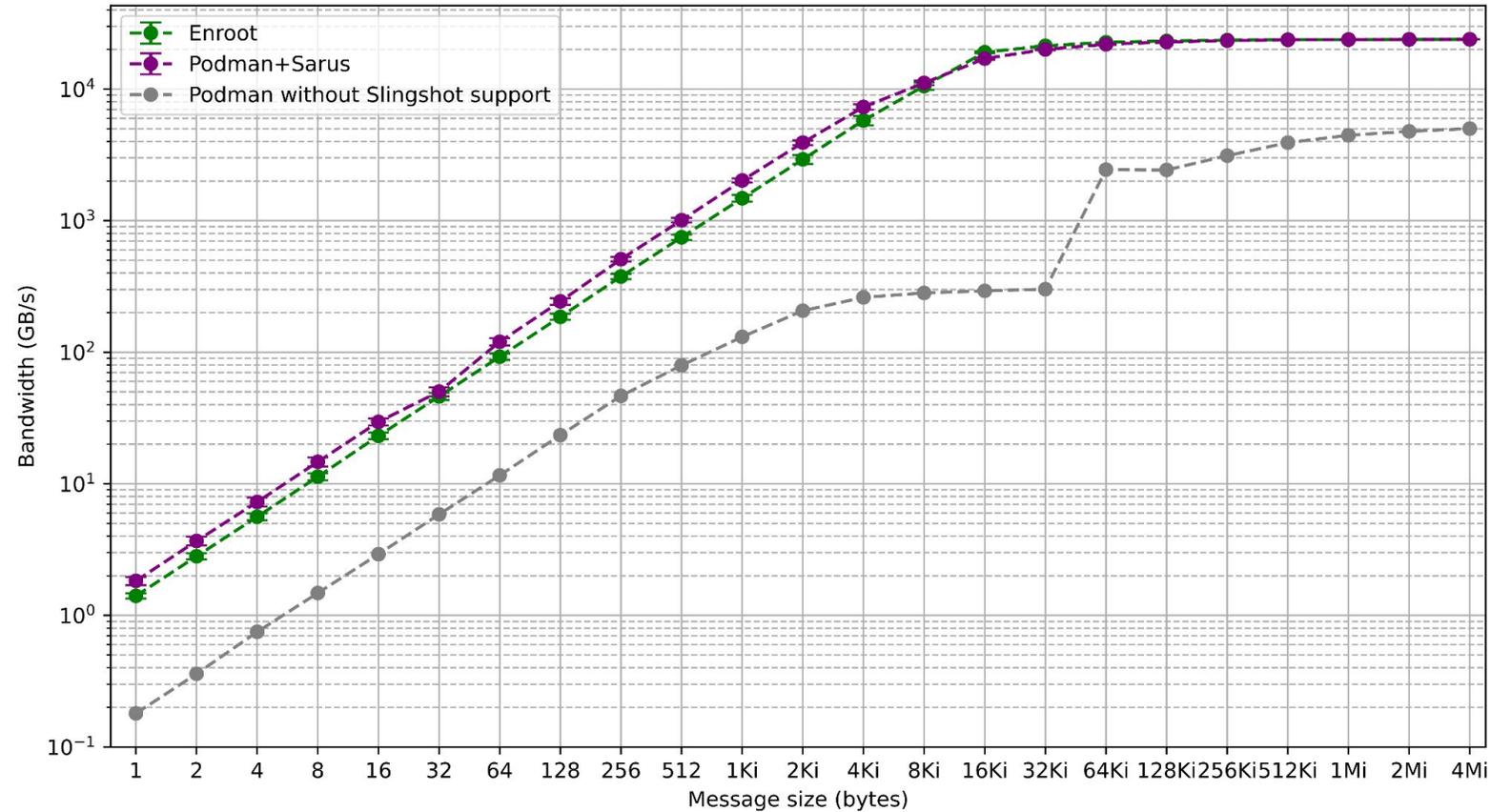
# Early tests

---

# Test setup

- System: Test & development vCluster within the Alps infrastructure
  - Node architecture: 2x AMD EPYC 7713 (64c) + 4x NVIDIA A100
  - Slingshot 11 200Gb interconnect
- Comparing Enroot (currently used in production) and Podman as internal runtimes within the CE toolset
- HPC feature support for Podman
  - Common “HPC mode” settings: dedicated Podman config module
  - CXI/Slingshot: CSCS-written CDI + Sarus MPI hook
  - GPU: NVIDIA CDI
  - Squashfs images migration and mounting: Parallax
- Using customized Pyxis as SPANK plugin
  - Skybox development not started yet

# OSU Bandwidth

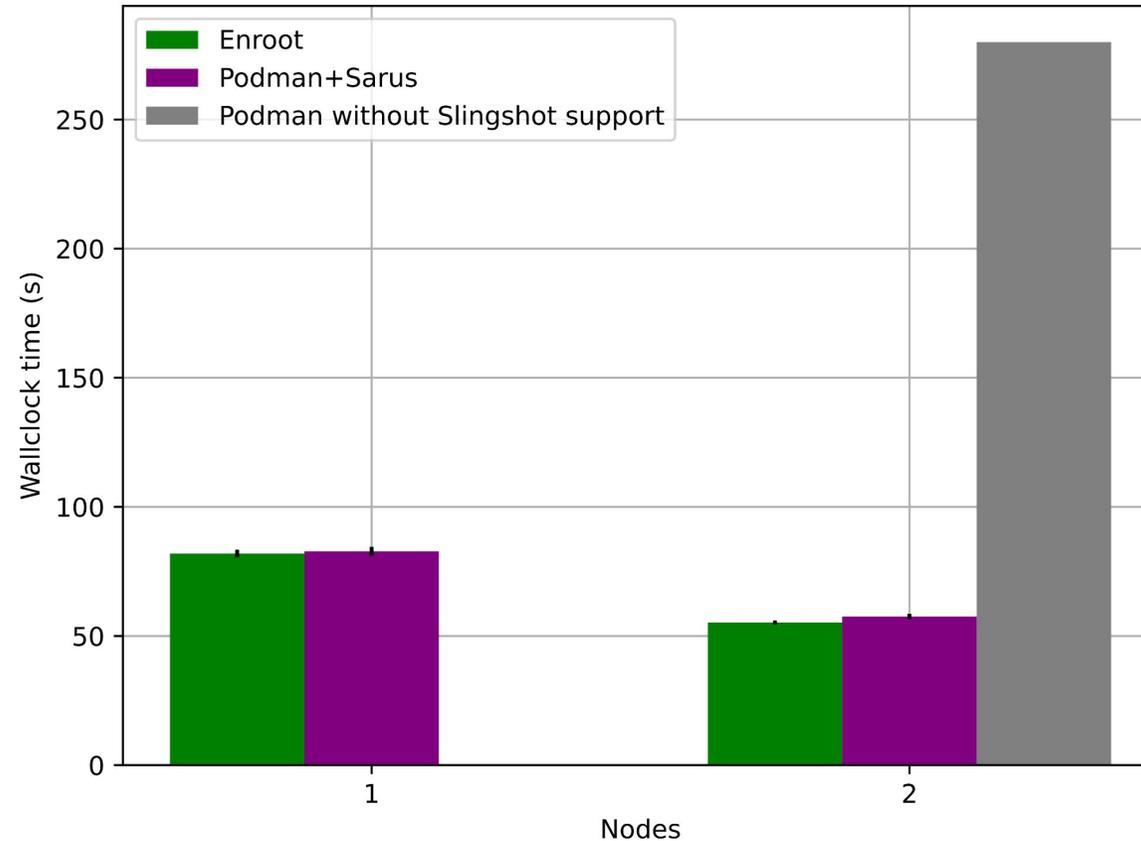


**Container:** OSU Micro-Benchmarks 7.5, OpenMPI 5.0.7, Ubuntu 24.04

**Test case:** osu\_bw benchmark (2 physical nodes, 20 repetitions)

**System:** Alps Infrastructure - Development vCluster (2 x AMD EPYC 7713, 4 x NVIDIA A100, Slingshot 11 200Gb)

# PyFR (Flux Reconstruction CFD)

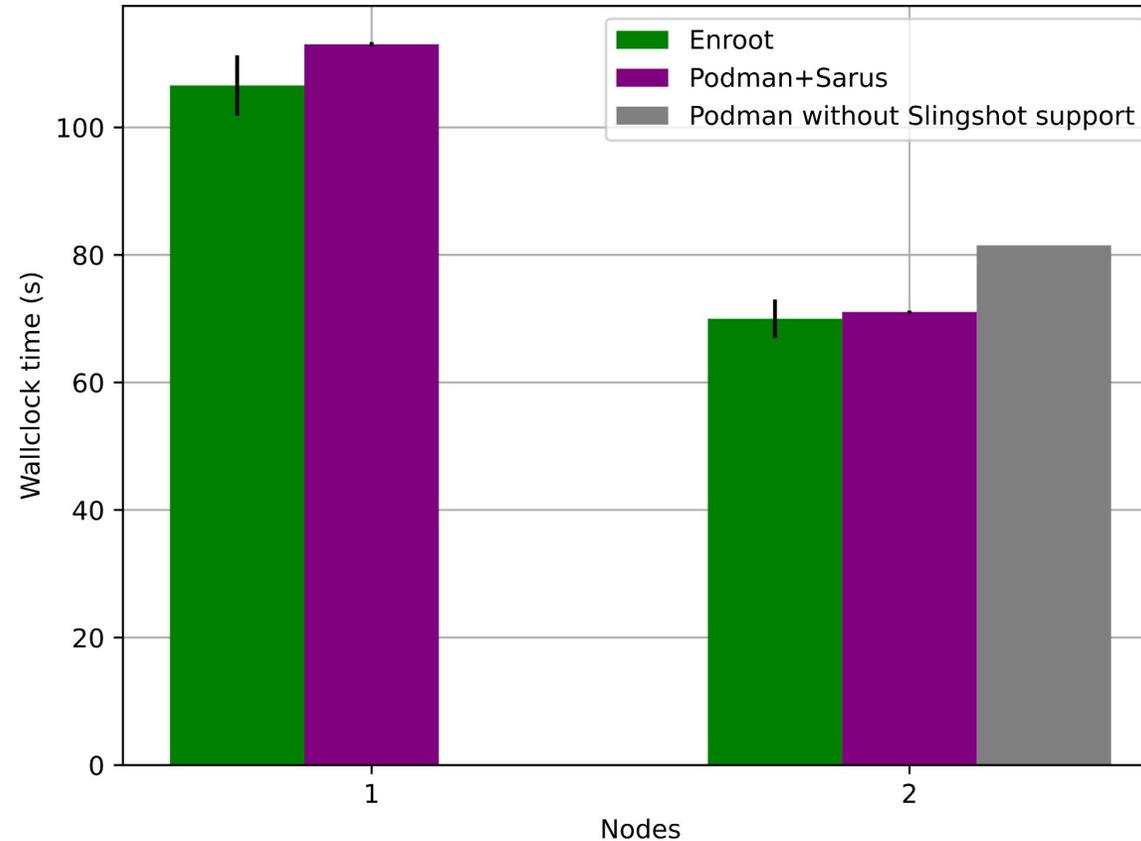


**Container:** PyFR 2.1, OpenMPI 5.0.7, CUDA 12.8, Ubuntu 24.04

**Test case:** 3D Taylor-Green vortex (4 ranks per node, 20 repetitions)

**System:** Alps Infrastructure - Development vCluster (2 x AMD EPYC 7713, 4 x NVIDIA A100, Slingshot 11 200Gb)

# SPH-EXA (Smoothed Particle Hydrodynamics)



**Container:** SPH-EXA v0.93.1, OpenMPI 5.0.7, CUDA 12.8, Ubuntu 24.04

**Test case:** Sedov spherical blast wave (4 ranks per node, 20 repetitions)

**System:** Alps Infrastructure - Development vCluster (2 x AMD EPYC 7713, 4 x NVIDIA A100, Slingshot 11 200Gb)

# Runtime startup comparison

	Enroot	Podman
Start function in SPANK plugin	3.7523 ± 0.052 s	1.2821 ± 0.054 s
Time spent in the runtime itself	3.7521 ± 0.052 s	1.2360 ± 0.052 s

Maximum runtime startup times averaged across 20 repetitions of PyFR runs

Container: PyFR 2.1, OpenMPI 5.0.7, CUDA 12.8, Ubuntu 24.04

Including support for CXI/Slingshot, NVIDIA GPUs, AWS NCCL plugin

# Conclusions

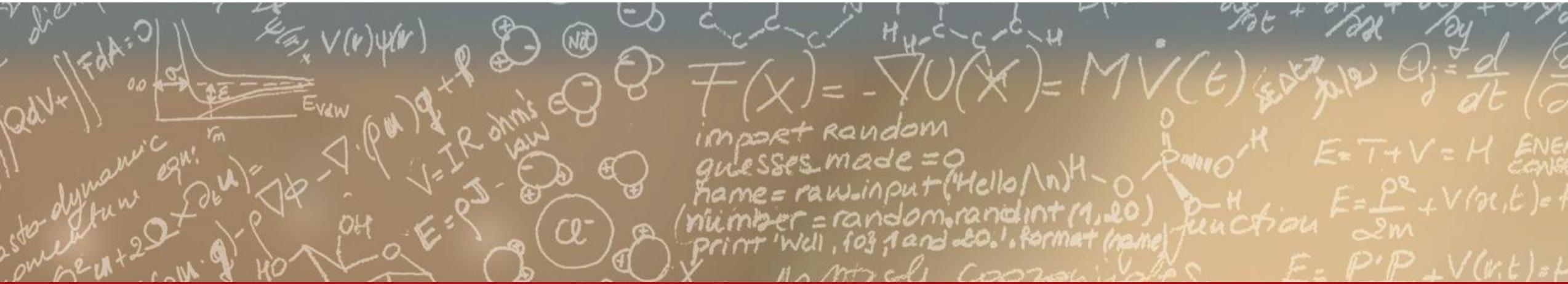
- Sarus evolving from an engine to a suite of tools
  - Modular, open-source collection of reusable components
  - Continuing to combine HPC with cloud-native flexibility and standards
- Enabling first-class container integration with HPC
  - Containers are transparent, manageable, and fully integrated into HPC
- Strategic shift towards performance-enhanced Podman as runtime
  - CSCS will use Sarus components to enable profitable usage of Podman in HPC
- Forward-looking and OSS-oriented
  - Towards a container stack for adaptability and sustainability
  - Smoother adoption path to other cloud native technologies



CSCS

Centro Svizzero di Calcolo Scientifico  
Swiss National Supercomputing Centre

ETH zürich



**Thank you for your attention.**