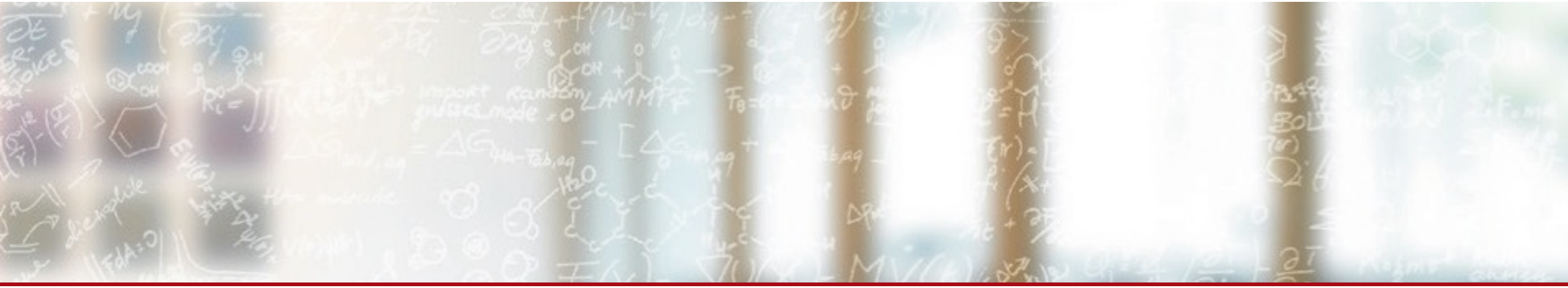
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



CSCS Service Catalog on Alps: Compute & Data

HPC-CH

Maria Grazia Giuffreda, Miguel Gila, Luca Marsella

2023-10-05

Alps - HPE Cray EX

Compute

1024 AMD Rome 7742 nodes 256/512GB
144 Nvidia A100 GPU nodes
  32 AMD MI250x GPU nodes
  Some thousands of GraceHopper modules
Slingshot network (200 Gbps injection)
100% liquid cooled
100+10 PiB HDD
5+1 PiB SSD (RAID10)

Data

100s of PiB tape library

Water cooled blades

Grace-Hopper superchip (GH200)

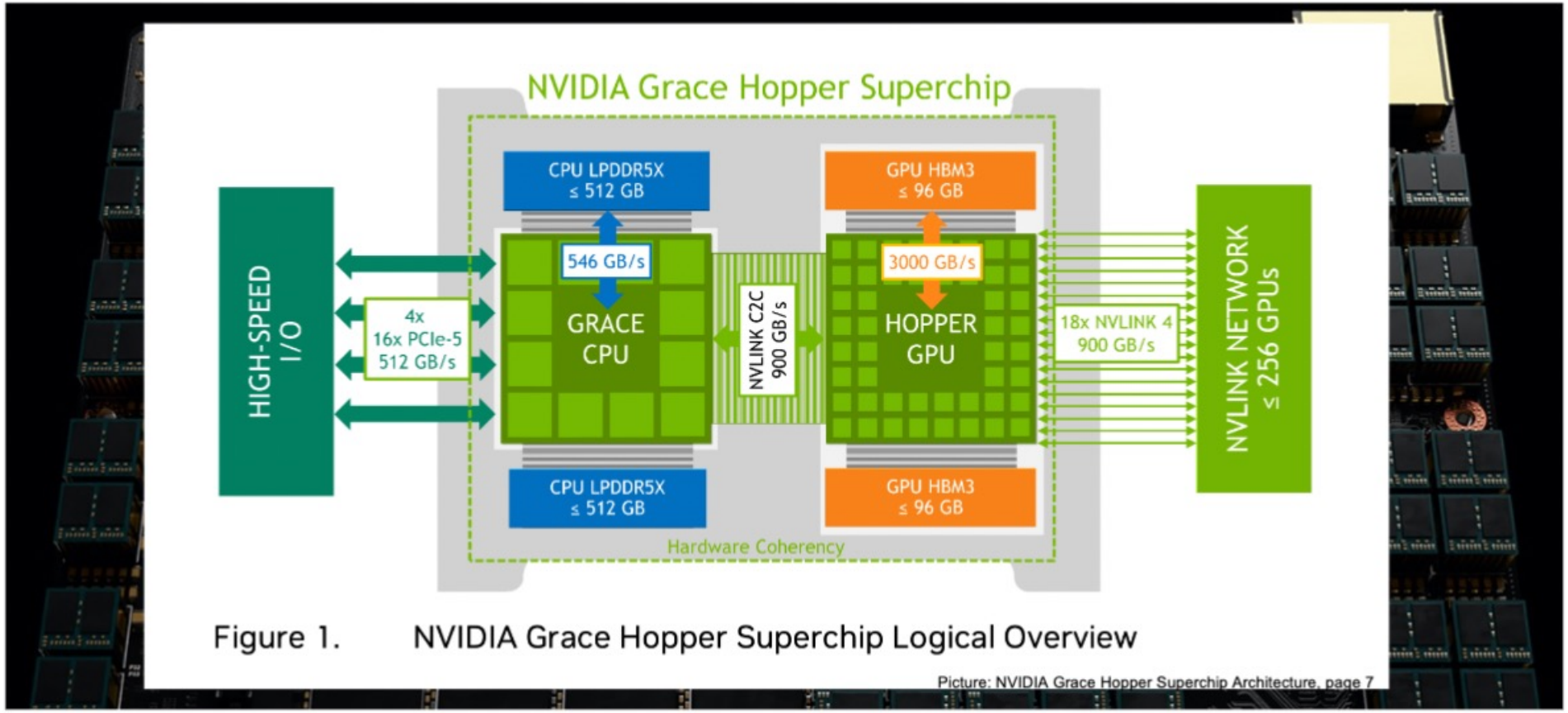


Figure 1. NVIDIA Grace Hopper Superchip Logical Overview

Picture: NVIDIA Grace Hopper Superchip Architecture, page 7

HPC & Cloud Convergence

HPC & Cloud Convergence

Science and engineering requires more and more computer assisted experiments

- Simulations of physical phenomena
- Digital Twins
- Design engineering products
- AI/ML statistic solutions

HPC & Cloud Convergence

HPC offers high-performance compute and data access

- Improves Time-to-Solution
- Managed efficiently data to compute
- Bare-metal performance, fixed amount of resources

Cloud offers high flexibility for business needs

- XaaS – business logic as a service
- Economy of scale – oversubscription of resources
- Virtualized resources, scalable to the infinite (and beyond)

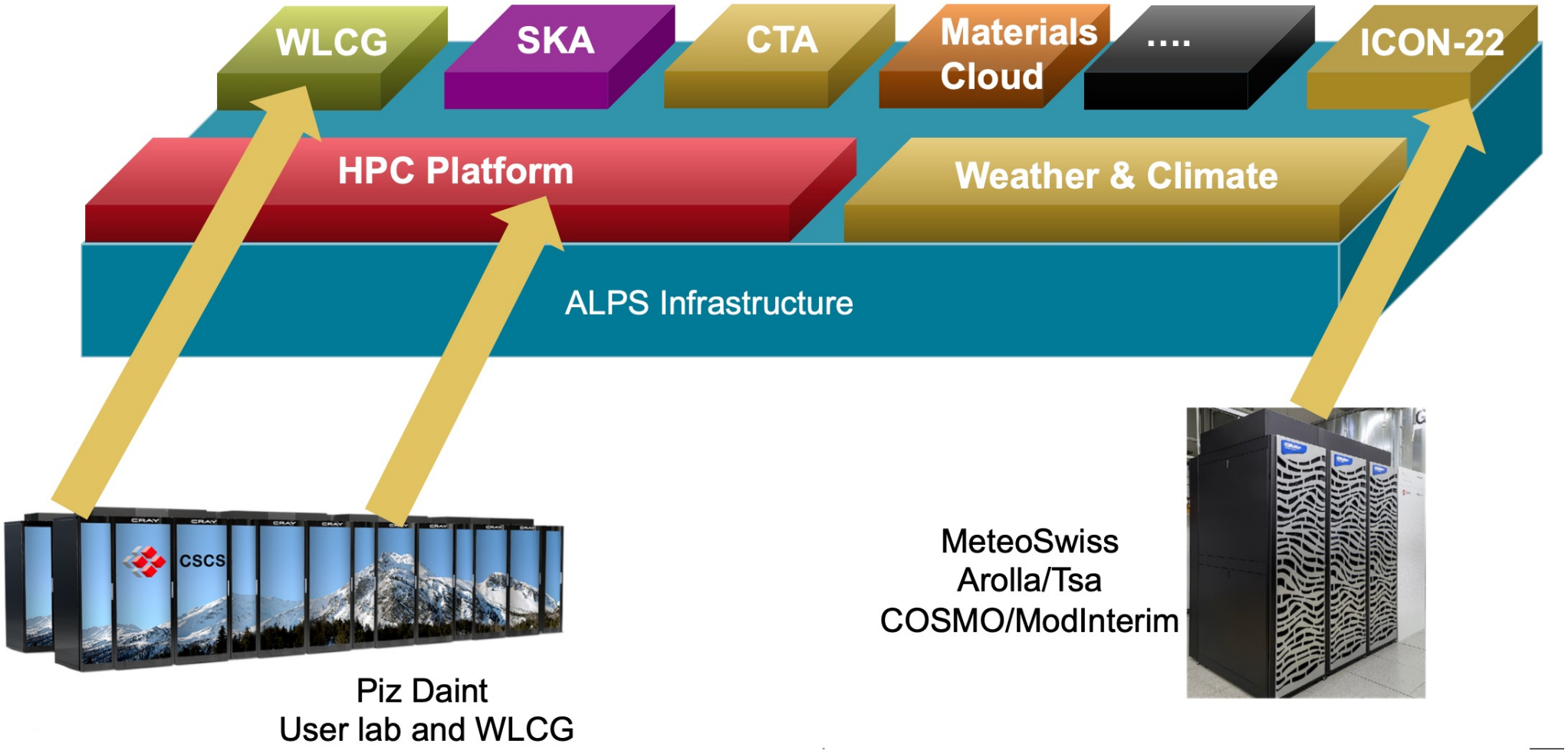
HPC & Cloud Convergence

Versatile software-defined cluster (vCluster)

HPC: High performance → vertically integrated stack→ limited set of services
Cloud: Virtualization at scale → high flexibility → limited performance

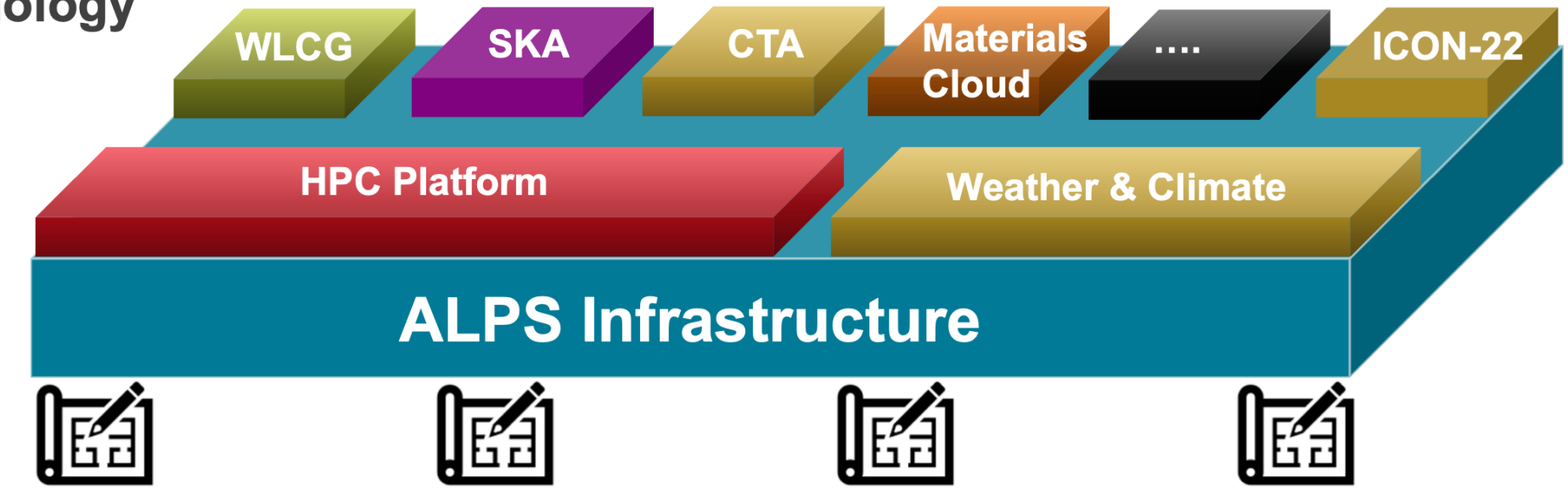
vCluster is a set of services to ensure flexibility on top of HPC

Consolidation of platform



Platform: a set of vClusters that answer a business/scientific need

Technology



User environments management	Build programming environments
Platforms and services management	Deploy platform services
Infrastructure as Code	Provision infrastructures with specific base images

- **Separation of concerns with layers**

Versatile software-defined cluster

  - Platforms
    - Provisioning of services with Nomad and/or Kubernetes
    - Container as an abstraction layer for compute nodes

  - Infrastructure as code
    - APIs and configuration management
    - Multi-tenancy: exclusive compute, network and storage segregation

Preparing for the future

Preparing for the future

Performance and Flexibility

Use container as a virtualization layers with OCI hooks

- Keep OS near bare metal – Accelerators and High-speed network drivers
- Bring low-level libraries in the container with OCI Hooks

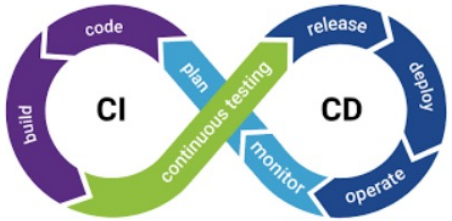
- Bring your own User Environment
  - Decouple HPC programming environments from underlying layers
  - UE can potentially become “just” an artifact mounted in the container

- **HPC business logic**
  - Web-facing API to access HPC resources (submit jobs, move data)
  - Web gateway

Preparing for the future

AiiDA: a workflow engine for Materials science

- Provenance, Plug-in, HPC interface
- Porting the engine to W&C workflows



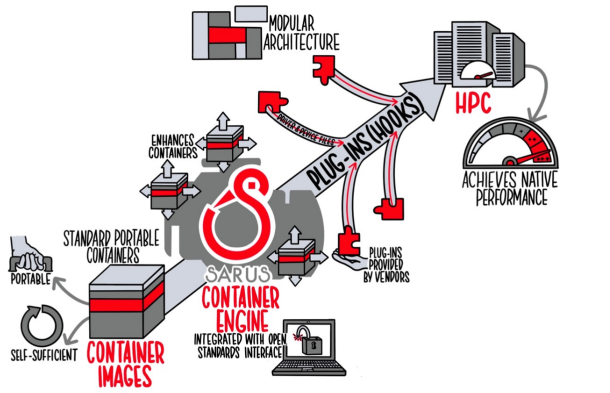
Provide a web-facing CI/CD services

- Use FirecREST to spawn jobs from the web
- Enable federation of identities

Testing framework with focus on performance

- Integrate into the CI/CD test steps with FirecREST

SARUS: Container Runtime Environment



SARUS

Sarus runs Linux containers on HPC environments:

- Developed following the specific requirements of HPC systems
- Spawns isolated containers built by users for a specific application
- Extensible runtime by means of OCI hooks for custom hardware

Compatible with the Open Container Initiative (OCI) standards:

- Pulls from registries with OCI Distribution Specification or Docker
- Imports and convert images with the OCI Image Format (e.g. Docker)
- Uses an OCI-compliant runtime to spawn the container process

More information: https://user.cscs.ch/tools/containers/sarus

User Environment

Alps User Environment (uenv mount with compiler, MPI library,...)

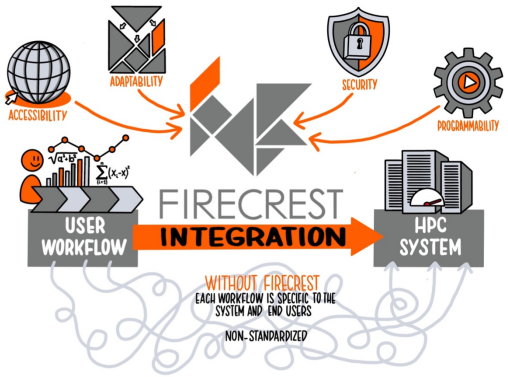
- AI/ML: PyTorch,TensorFlow,...
- Materials Science: CP2K, QuantumESPRESSO, VASP,...
- MD Simulations: GROMACS, LAMMPS, NAMD,...
- Visualization: Paraview, Visit, VMD,...

Cray Programming Environment:

- programming environment on Cray systems with performance tools

More information:
https://confluence.cscs.ch/display/KB/User+Environments+on+Alps

FirecREST



FirecREST

FirecREST is a RESTful API for managing HPC resources at CSCS

- Integrate FirecREST into web-enabled portals and applications
- Securely access CSCS services (job submissions, data transfer,...)

Users can make HTTP requests to perform the following operations:

- Basic system utilities like ls, mkdir, mv, chmod, chown,...
- Actions with the Slurm workload manager (submit, query, cancel jobs)
- Data transfer: internal (between CSCS systems) and external

More information: https://user.cscs.ch/tools/firecrest

Regression testing with ReFrame

ReFrame is a framework for regression tests on HPC systems

- Abstract away the complexity of the interactions with the system
- Separate the logic of a regression test from the low-level details
- Users can write portable regression tests, focusing on functionality

CSCS provides ReFrame pre-configured for its systems:

- ReFrame can be accessed with **module load reframe-cscs-tests**
- The module provides the configuration as well as regression tests
- Users can run the tests already provided by CSCS staff

More information: https://user.cscs.ch/tools/reframe

Interactive computing with JupyterLab

JupyterLab is the web-based user interface for Project Jupyter

- Create and share documents with live code, equations, visualization, ...
- Same notebook document format as the classic Jupyter Notebook
- Ability to work with multiple documents using tabs or splitters side by side

CSCS JupyterLab powered by JupyterHub with a ready-made Python kernel

- You can add your own kernels based on your own virtual environments
- Multi-user hub spawning multiple instances of single-user Jupyter server
- Interactive execution of JupyterLab over single and multiple nodes

More information: https://user.cscs.ch/tools/interactive/jupyterlab

Storage & Data Strategy

Alps era storage specs and pictures

In the process of being accepted

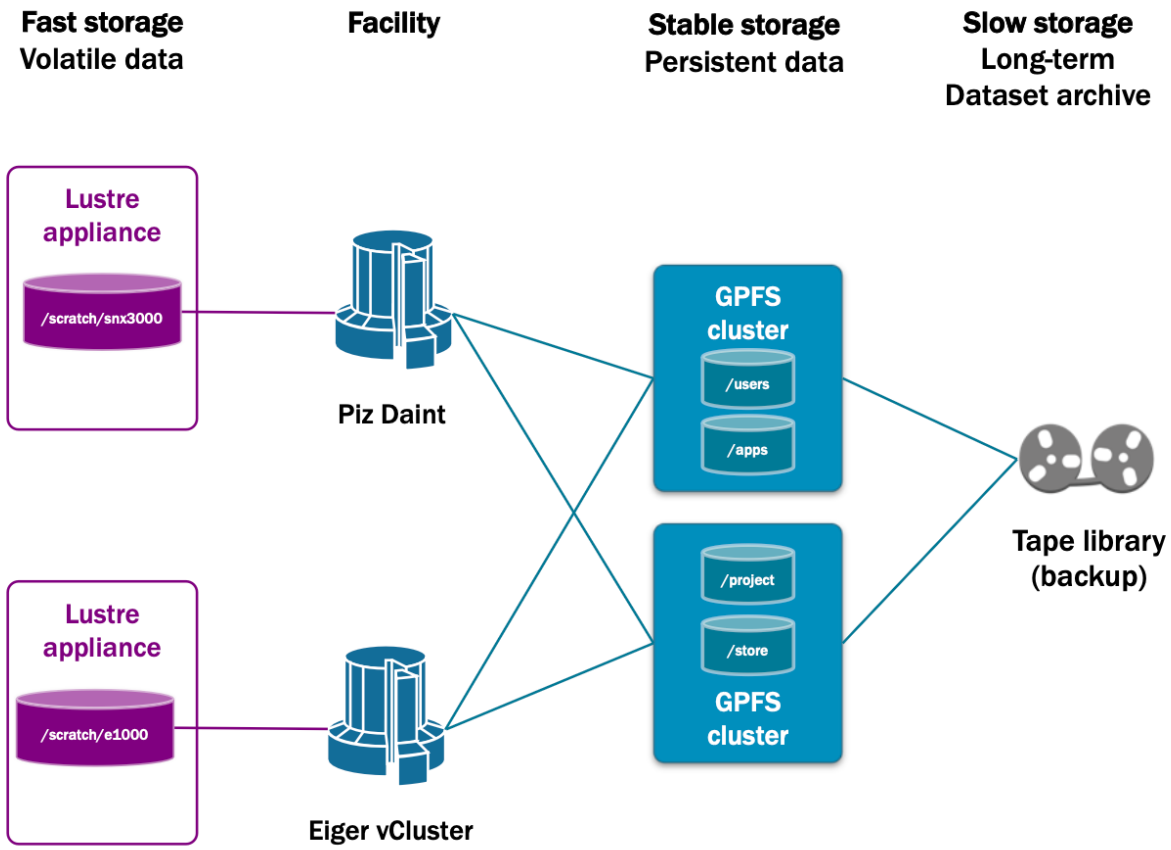
- Capstor
  - 100 PiB HPE ClusterStor E1000D
  - Spinning disks
  - Raw performance:
    - **~1TB/s**
    - 300k Write iops | 1.5M Read iops
  - 8480 spinning disks (16 TB each)
  - 6 metadata servers
  - 11 full racks
  - Slingshot 11

Accepted, rolling to users soon

- iopsstor
  - 3.2 PiB HPE ClusterStor E1000F
  - All flash
  - Raw performance:
    - 240GB/s Write | 600 GB/s Read
    - **13.5M Write iops | 18.4M read iops**
  - 240 NVMe devices (30TB each)
  - 2 metadata servers
  - 1 rack
  - Slingshot 11

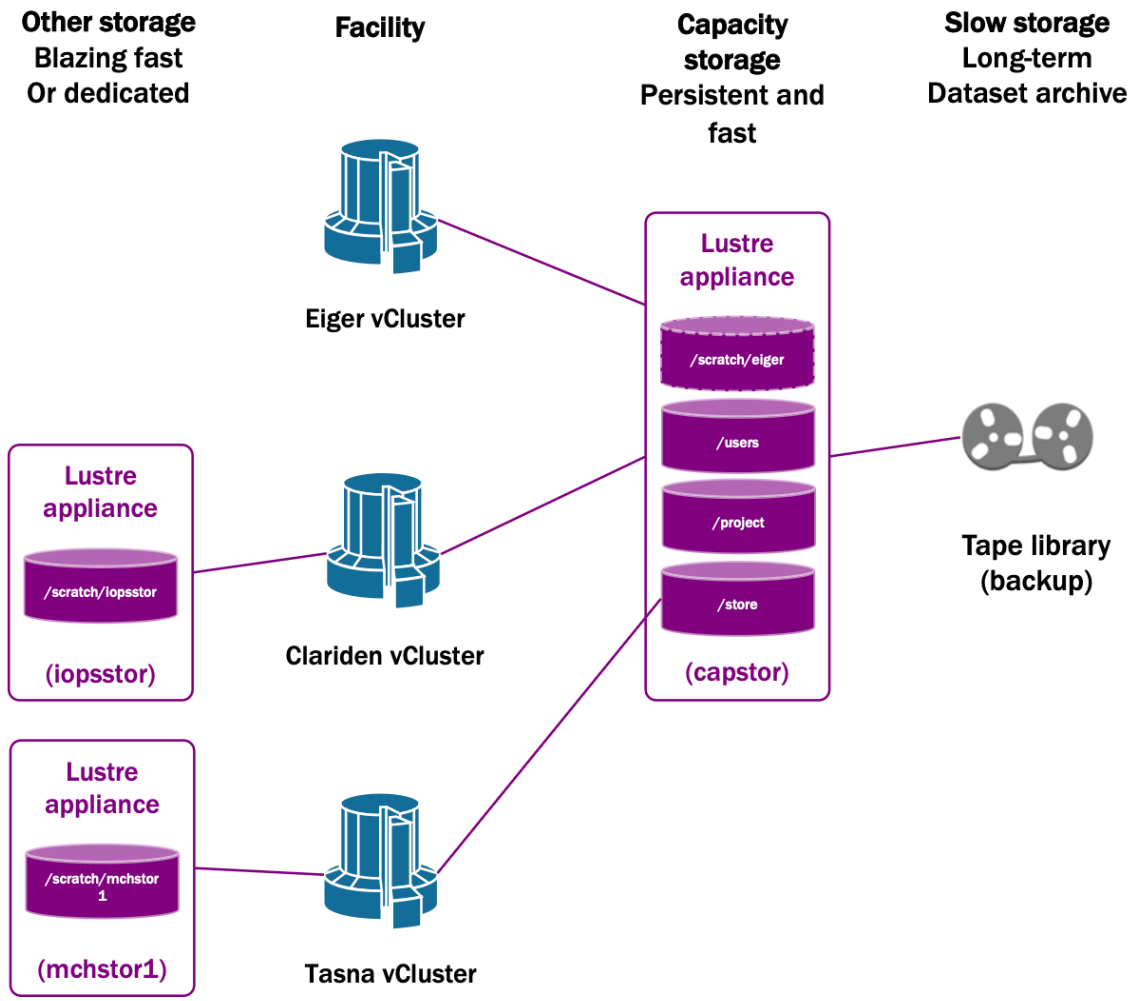
Storage in the Piz Daint era

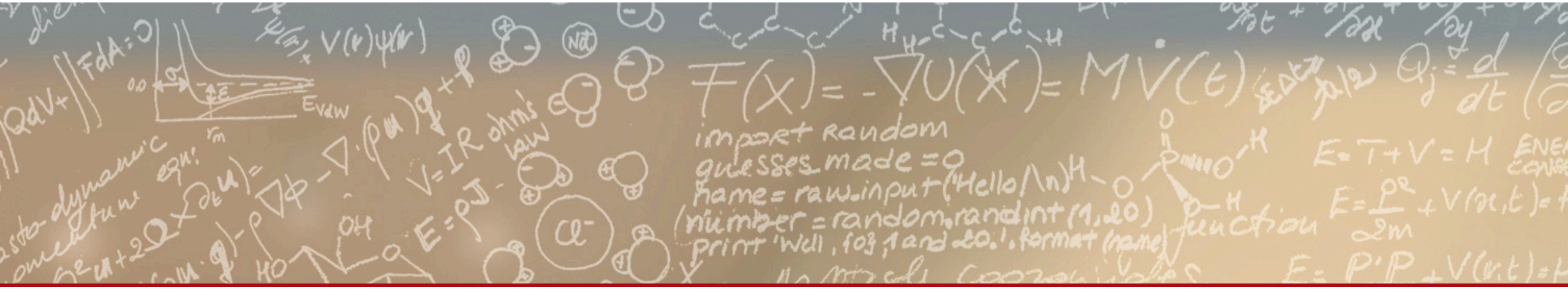
- Different storage backends depending on
  - Use-case
  - Performance envelope
  - Technology constraints
  - Historical reasons
- Persistent data and tape access stored on IBM Spectrum Scale (GPFS) clusters and a myriad of backends (SAS or FC disks, SAN, JBODs, etc.)
- Volatile data with fast access patterns on Lustre appliances



Storage in the Alps era

- Consolidate backends and technologies
- Most storage areas will be based on Lustre, which is a much more mature product now
- Filesystems/spaces available:
  - Capstor
  - Iopsstor
  - Purpose, dedicated filesystem
- Metadata ops for the different areas hitting different servers





Thank you for your attention