Shipping HPC Ecosystems Across Platforms

Portable and Composable HPC Clusters as Code

Authors: German Giraldo[1], Théo Grivel[1], George Ioannidis[1], Edita Kizinevic[1], Carolina Lindqvist[1], Nicolas Litchinko[1], Pablo Llopis[1], Antonio J. Russo[1], Gilles Fourestey[1]

Contact: scitas@epfl.ch

[1] École Polytechnique Fédérale de Lausanne, Rte Cantonale, 1015 Lausanne, Switzerland

Scan the QR for video narration and code

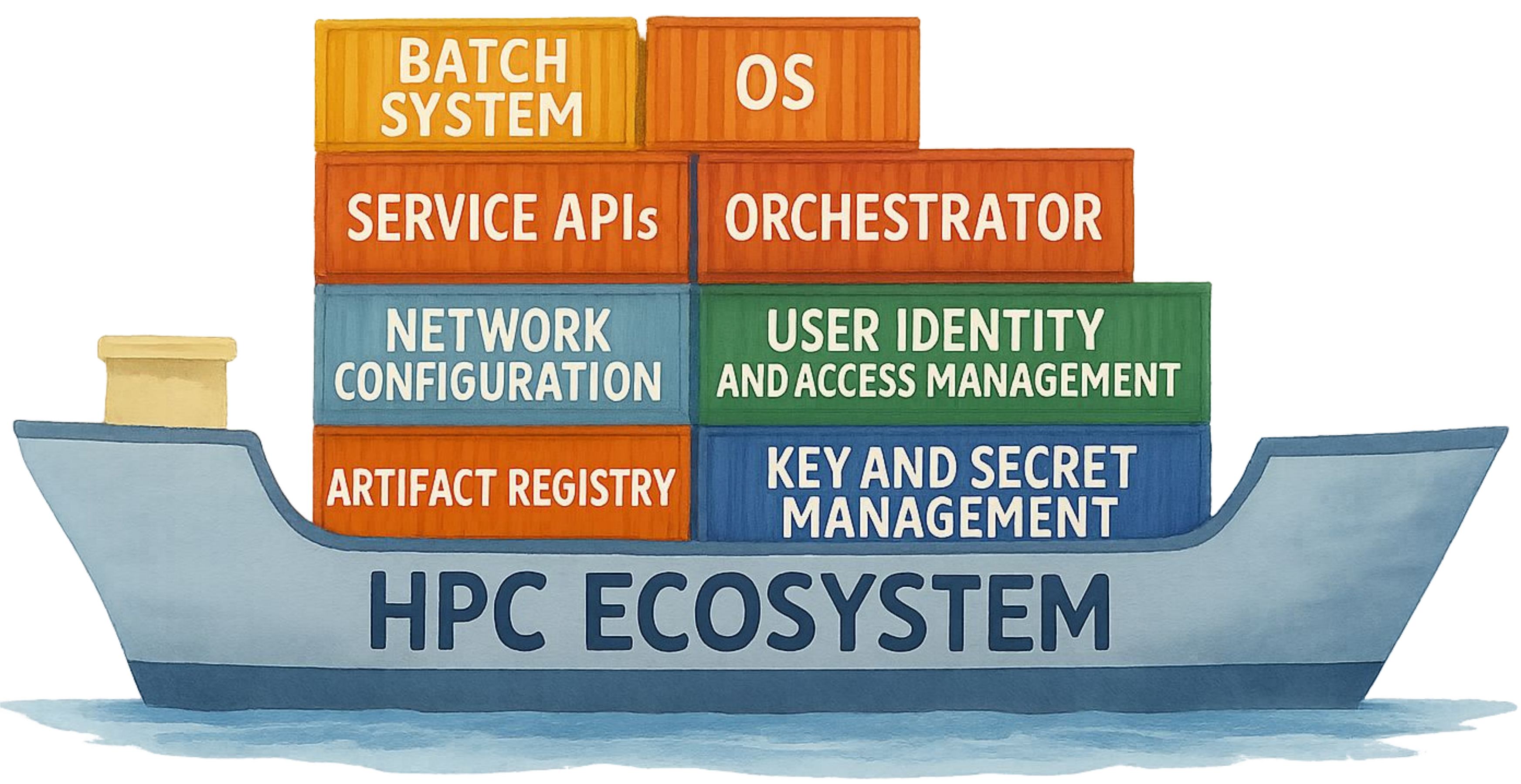
PROBLEM STATEMENT

- HPC infrastructures typically require intricate and platform-specific vertically integrated configurations. This poses a challenge for the portability and reproducibility of HPC system stacks as a whole.
- We leverage and extend versatile software-defined clusters (**vClusters**)[1], pioneered by CSCS, which enable modular Infrastructure-as-Code DevOps practices for their supercomputing ecosystem, Alps[2].
- Our purpose is to develop a **portability model** for HPC ecosystems that goes beyond container-based and cloud native solutions by enabling mobility at the infrastructure level.
- We evaluate our approach with a portable implementation of the Alps supercomputer vCluster ecosystem that runs the production Swiss national weather forecast (MeteoSwiss[3]) HPC workflows. Our work demonstrates that one can employ portable vCluster definitions to bring up the same HPC system on a different platform. We showcase the portability of the Alps-based Weather and Climate HPC vCluster on GCP[4].

GOALS

Portability: The model must enable deploying vClusters both on-premises and on a completely different platform, such as the public cloud.

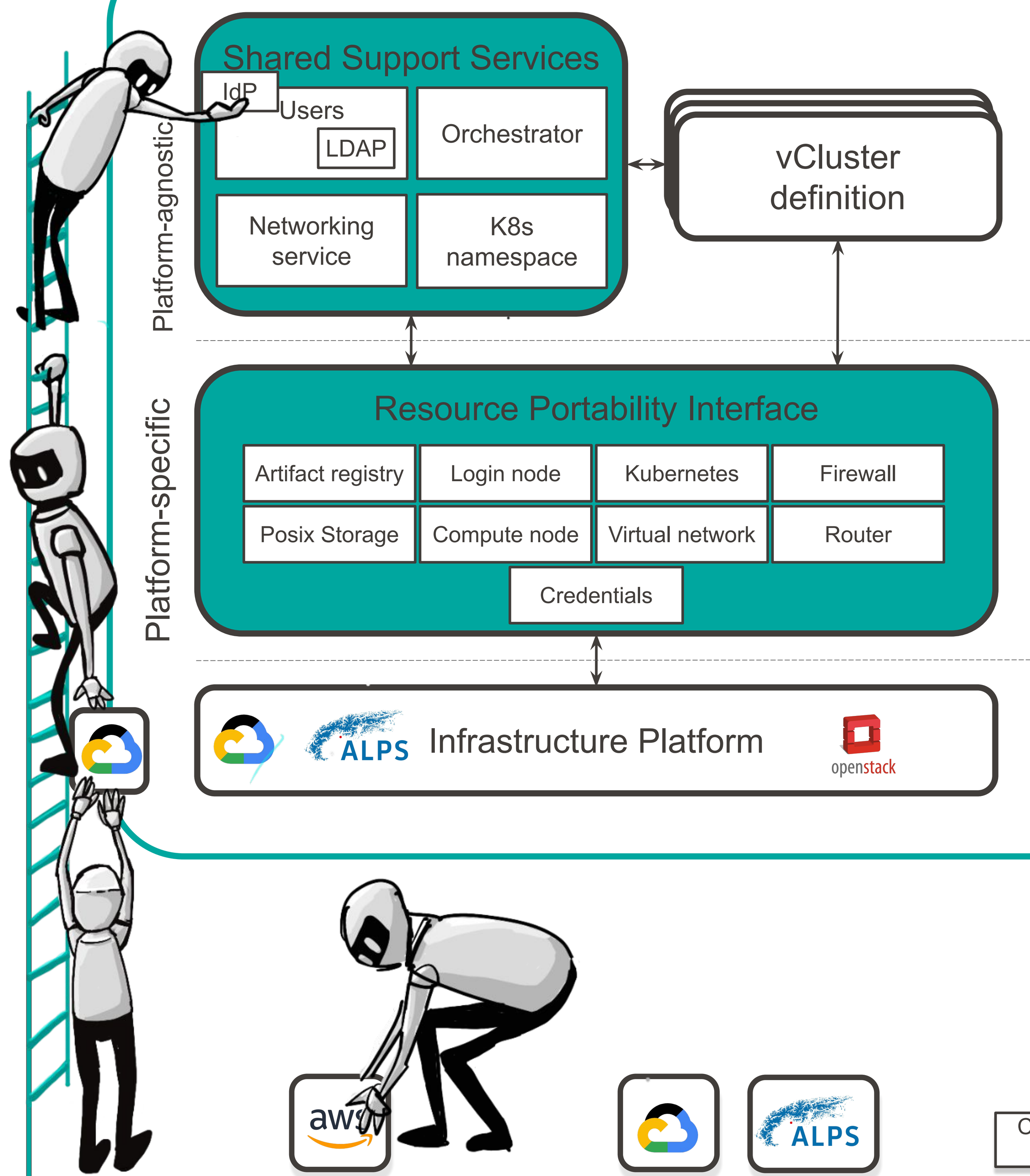
Flexibility: The model should enable users to configure their HPC infrastructure according to the requirements of their use case.



PORTABILITY MODEL

- We propose a model that aims to enable portable **vClusters** by establishing a clear interface between platform-specific and platform-agnostic code.
- **vServices** are modular components that run individual HPC Infrastructure services, such as *Slurm*, or *NVIDIA/enroot*. vService deployment to vCluster worker nodes is orchestrated via **Nomad**[1].
- Our objective is to isolate and reduce any platform-specific definitions to facilitate mobility across platforms.
- A **vCluster** is an Infrastructure-as-Code-based declaration of an HPC cluster. They are comprised of **infrastructure resources**, and **vServices**.
- The portability model needs to provide all necessary resources and services for the deployment and operation of vClusters.
- The resulting code must be compatible with the declarative IaC design of vClusters.

DETAILED DESCRIPTION



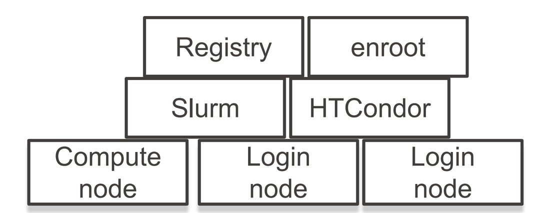
A number of **supporting services** that are shared among **vClusters** are deployed as part of our portable ecosystem approach.

These services enable the bootstrapping and management of vClusters, network security and user identities.

A **resource portability interface** provides a library of infrastructure components as instantiable resources such as *networks*, *compute nodes*, or *firewall rules*.

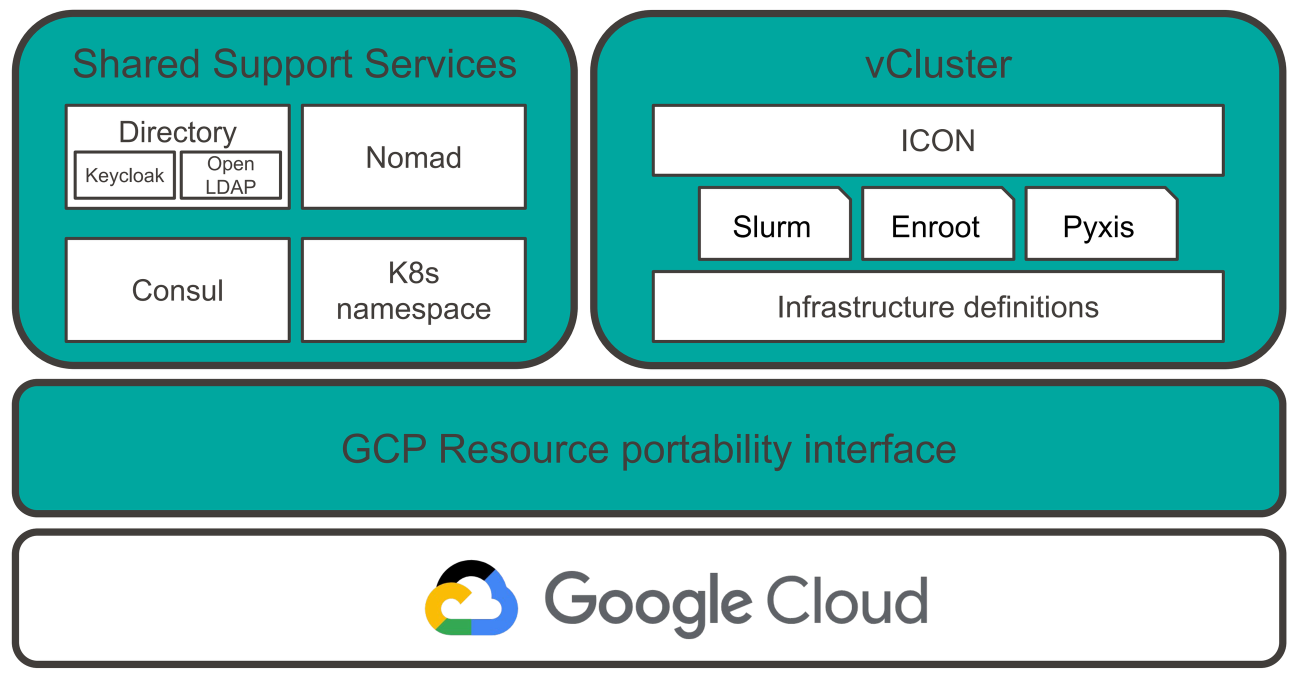
Along with vServices, these components enable **composability** and allow for **expressive vCluster definitions**.

The same vCluster can then be easily **spun up on a different platform** that is supported by the platform-specific infrastructure modules.



PROOF OF CONCEPT

- To evaluate the model's feasibility, we use MeteoSwiss HPC workflows, which utilize their weather forecasting numerical model, ICON[6]. The MeteoSwiss production runs on an Alps-based vCluster at CSCS, and we enable the automatic creation of an equivalent vCluster on GCP. The figure below depicts this specific implementation.



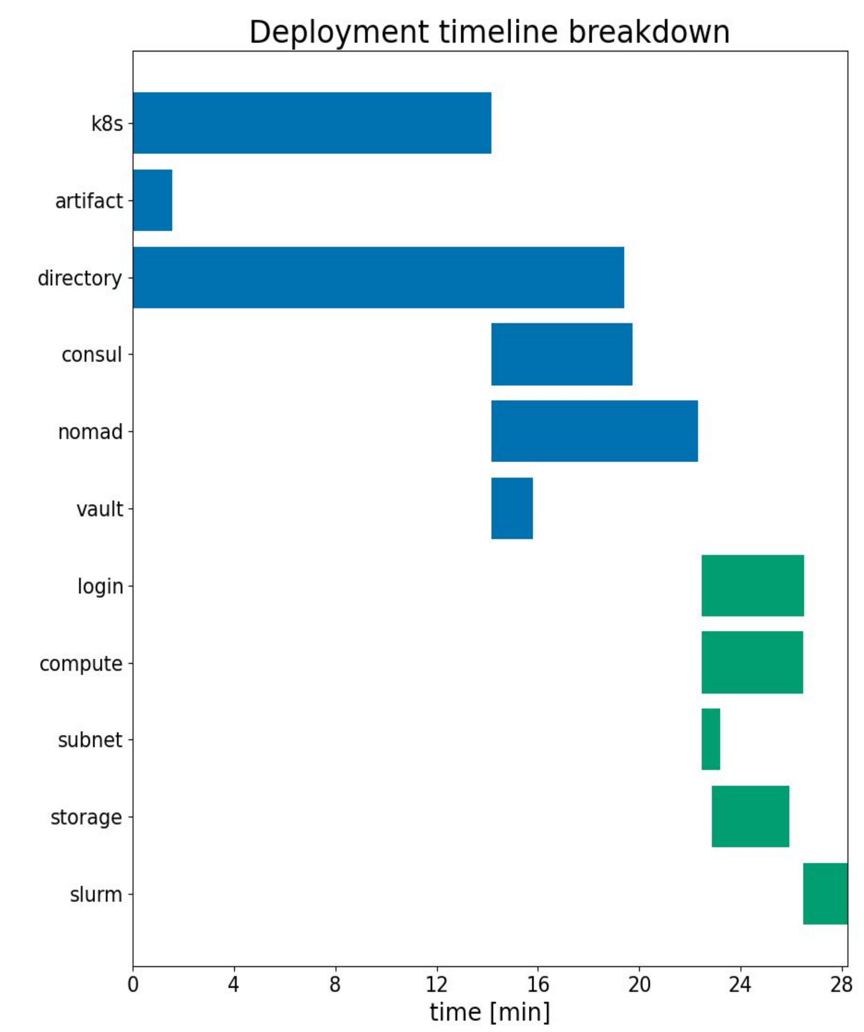
- The system is deployed in two stages. Initially, we deploy the shared support services, establishing the infrastructure to build vClusters. In the second phase, we deploy the vCluster according to the requirements of the workflow.

Requirements for ICON workflow:

- Minimum 2 Compute nodes
- 4x NVIDIA A100 GPUs per node
- Slurm, Pyxis and Enroot vServices

RESULTS

- We are able to automatically deploy all the supporting infrastructure and shared services necessary to deploy vClusters on GCP.
- We can set up a platform-agnostic vCluster and provision the necessary infrastructure and vServices. Then, we are able to execute the ICON workflow on the vCluster successfully.
- The model enables a prompt decommissioning of the infrastructure after the HPC workflow finishes. This feature avoids wasting resources, which is of particular importance when using public cloud platforms such as GCP.



Example of Minimalistic vCluster definition with vService.

```
module "vcluster" {
  source = "git@gitlab.epfl.ch:swisstwins/vs-vcluster.git..."
  k8s_namespace_control_enable = true
  namespace                    = "${terraform.workspace}-mvp"
  owner                        = "hpc-systems@groupes.epfl.ch"
}

module "subnet" {
  source = "git@gitlab.epfl.ch:swisstwins/vc-modules-gcp..."
  network_id        = var.infra_info.network_id
  subnet_cidr_range = "10.10.0.0/16"
  subnet_region     = "europe-west1"
  vcluster_name     = module.vcluster.namespace
}

module "storage_posix" {
  source = "git@gitlab.epfl.ch:swisstwins/vc-modules-gcp..."
  capacity_gb           = 1024
  compute_subnetwork_id = module.subnet.subnet_id
  location              = "europe-west1-b"
  mount_point_directory = "/scratch"
}

module "compute_instances" {
  source = "git@gitlab.epfl.ch:swisstwins/vc-modules-gcp..."
  disk_size_gb   = 30
  image          = "rhel/rhel-9"
  instance_count = 1
  instance_type  = "n1-standard-2"
  nomad_info     = var.infra_info.nomad_info
  storage_info   = module.storage_posix
  subnetwork_id  = module.subnet.subnet_id
  zone           = "europe-west1-c"
}

module "example_vservice" {
  source = "git@gitlab.epfl.ch:swisstwins/vs-example.gitf=1dbac..."
  vcluster_name = module.vcluster.namespace
  secret        = data.vault_generic_secret.my-secret.data["value"]
}
```

CONCLUSIONS

Collaboration
This project shows how working together across institutions helps improve cloud-based HPC solutions, making them more flexible and scalable.

Mobility
This approach enables mobility at the infrastructure layer, going beyond existing solutions that focus on OS or Application layer portability

Flexibility
The solution enables novel HPC solutions in areas like urgent computing and business continuity.

Modularity
The modular infrastructure and service definitions enable vClusters to be adapted to a wide range of use cases.

REFERENCES

[1] Sadaf R Alam, Miguel Gila, Mark Klein, Maxime Martinasso, and Thomas C Schulthess. 2023. Versatile software-defined HPC and cloud clusters on Alps supercomputer for diverse workflows. The International Journal of High Performance Computing Applications 37, 3-4 (2023), 288–305.arXiv:https://doi.org/10.1177/10943420231167811
[2] Swiss National Supercomputing Center. 2024. Alps. Retrieved August 14, 2025 from https://www.cscs.ch/computers/alps
[3] MeteoSwiss. 2025. About Us. Retrieved August 14, 2025 from https://www.meteoswiss.admin.ch/about-us.html
[4] Google LLC. 2025. Google Cloud Documentation. Retrieved August 14, 2025 from https://cloud.google.com/docs
[5] Terraform by HashiCorp. HashiCorp, https://www.terraform.io/. Accessed 1 May. 2025.70.8
[6] MeteoSwiss. 2025. ICON Forecasting Systems. Retrieved August 14, 2025 from https://www.meteoswiss.admin.ch/weather/warning-and-forecasting-systems/icon-forecasting-systems.html

* Hand-drawn illustration enhanced using AI (flux-kontext)