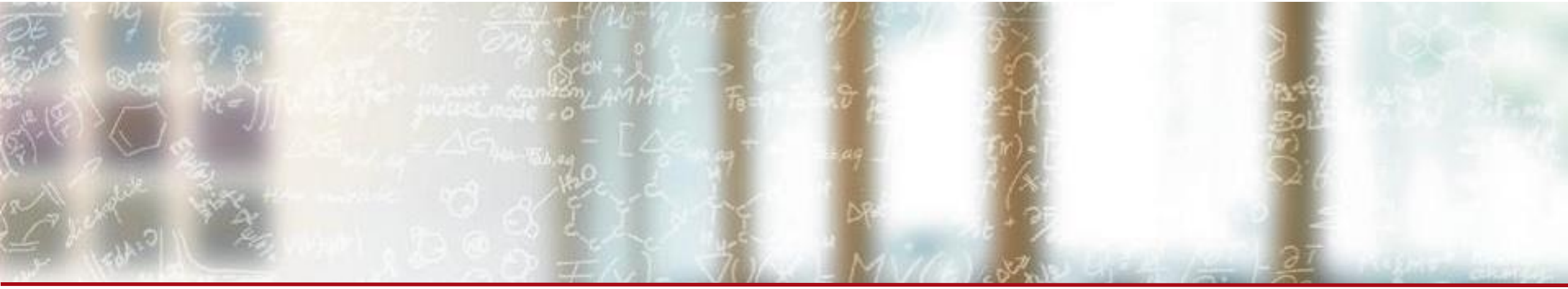
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



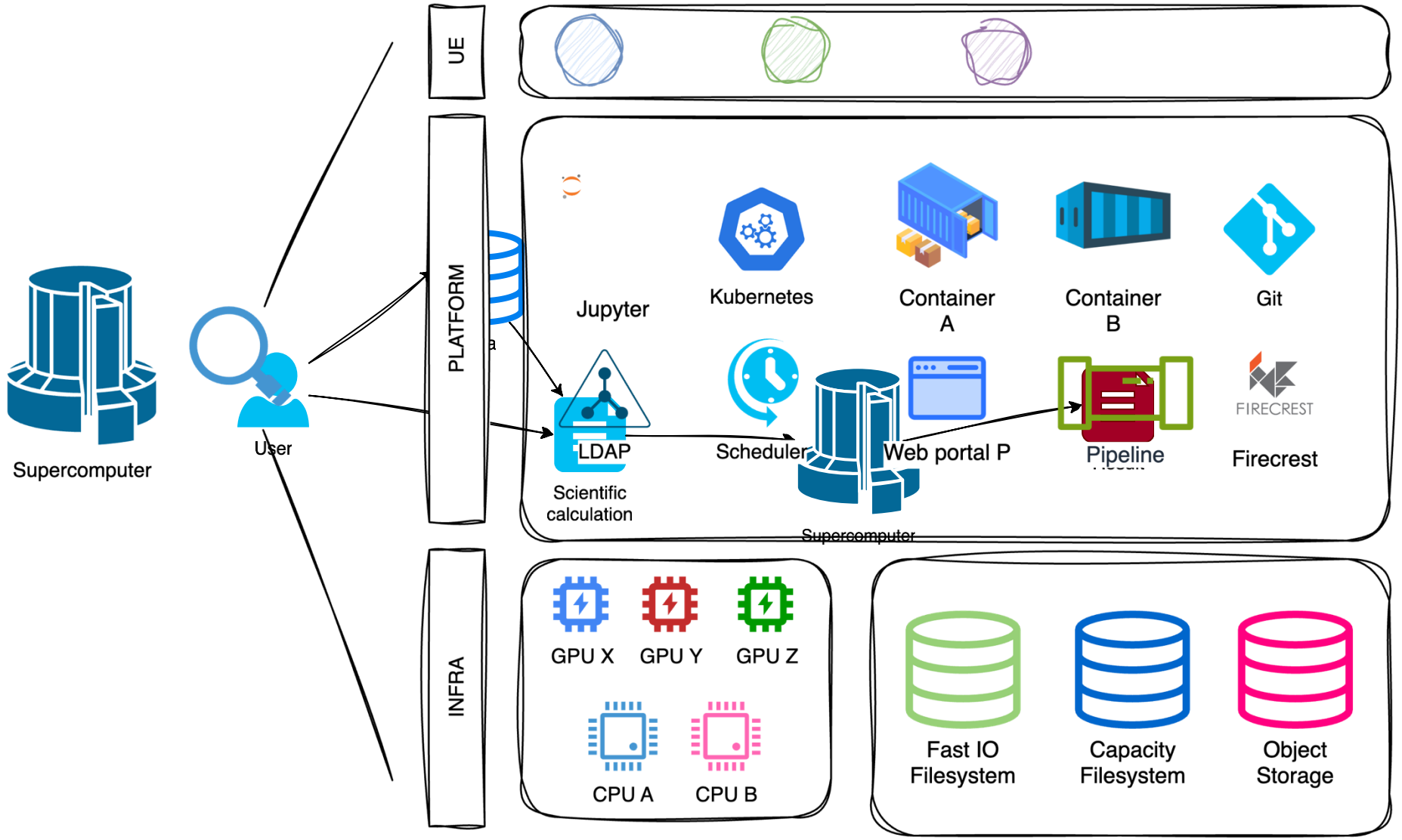
Service Management and the vCluster Technology

SOS27

Miguel Gila, CSCS

March 18, 2025

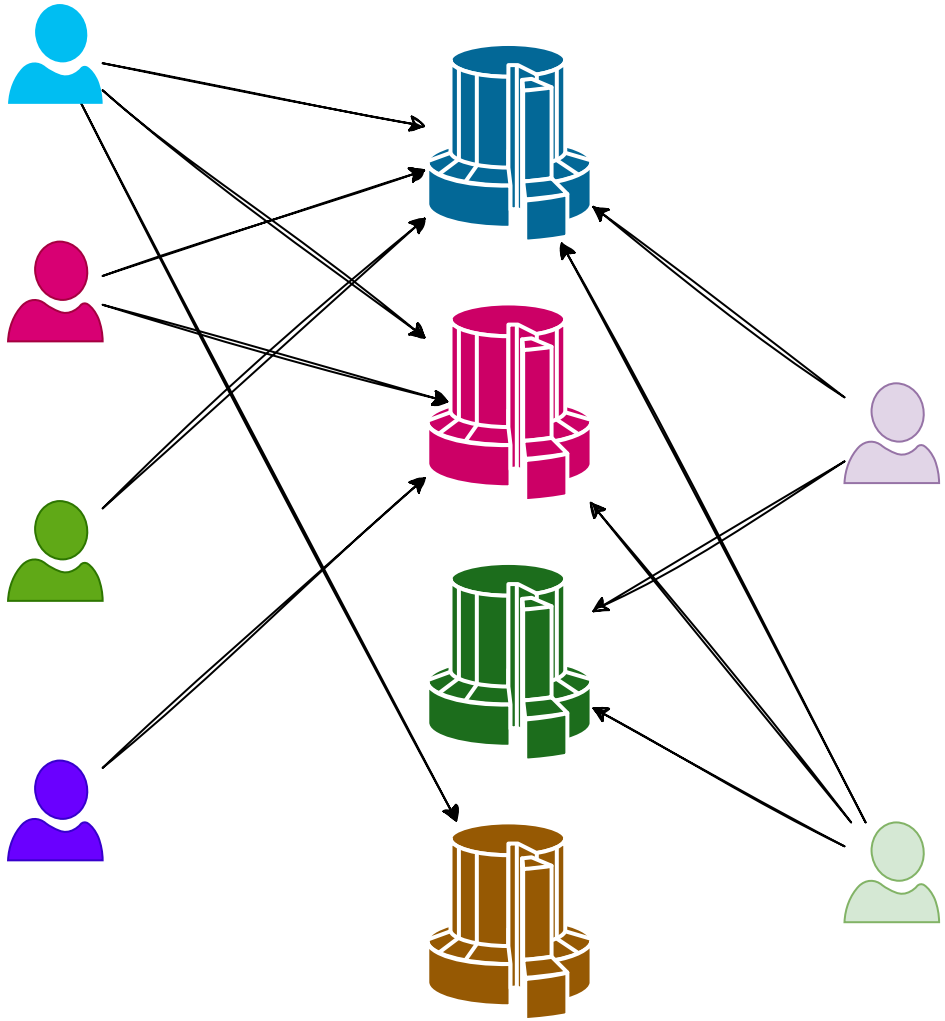
Running jobs, right?



Adding dimensions complicates things even more

- Multiple user **communities**
- **Different requirements** per community, and of course, even within a given community
- Multiple **sites**
- Potentially several infrastructure **vendors**
- Different **configurations** of hardware and software

- Data management
- Service management
- User management
- …



vCluster Technology

HPC and Cloud convergence

- Science and engineering requires more and more computer assisted experiments
  - Simulation of physical phenomena and behaviours
  - Digital Twins
  - Design engineering products
  - AI/ML solutions

- HPC offers high-performance compute and data access
  - Improves time to solution
  - Managed efficiently data to compute

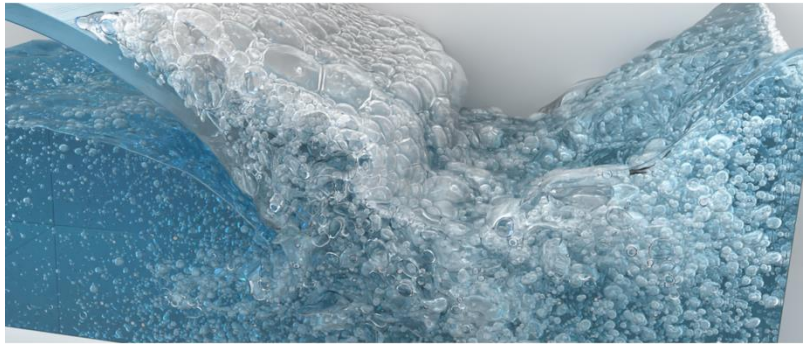
- Cloud offers high flexibility for business needs
  - XaaS – business logic as a service
  - Economy of scale – oversubscription of resources

How to combine the best of the two worlds?

HPC and Cloud convergence

- Cloud providers have invested heavily in abstractions from the hardware (IaC, APIs, virtualization, etc.)
- **Cloud** design principle – towards enterprise
- Economy of scale – oversubscription of resources

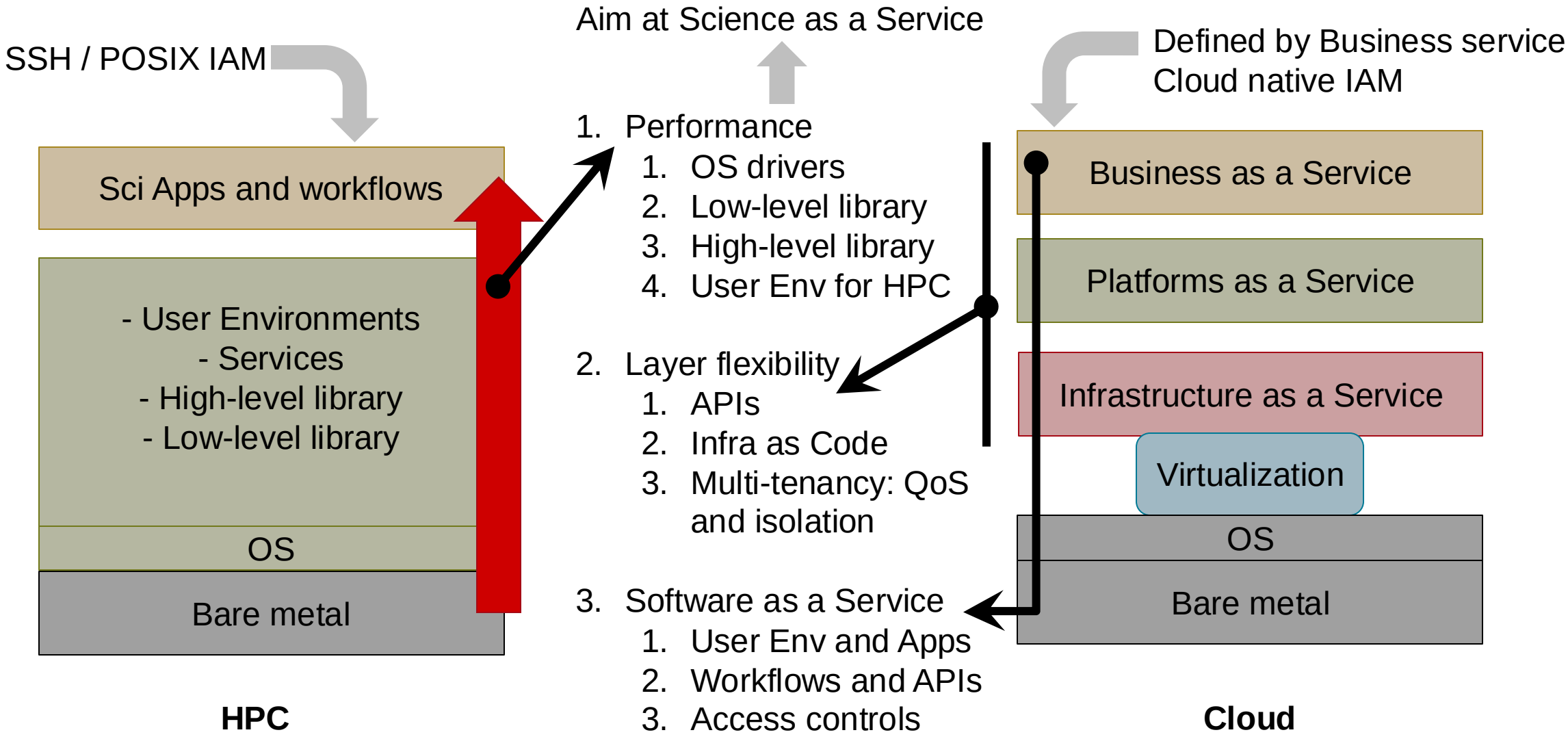
Virtualization at scale → High **flexibility** → Limited performance



- **HPC** design principle – towards science
- Traditionally, HPC has invested heavily in vertically integrated environments
- Improves Time to Solution

High performance → **Limited** set of services → Fully integrated stack

HPC and Cloud concepts to enable Science



How to achieve HPC and Cloud convergence?

- Performance and Flexibility
  - Use container as an abstraction 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 UE technology
    - Decouple HPC programming environments from underlying layers
    - UE as an artifact mounted in the container
- Separation of concerns with layers
  - Platforms
    - Provisioning of services with orchestrators
    - Container as an abstraction layer for compute nodes
  - Infrastructure as code
    - APIs and configuration management
    - Multi-tenancy: exclusive compute, network and storage segregation
- HPC business logic
  - Web-facing API to access HPC resources (submit jobs, move data)
  - Web gateway

Versatile software-defined cluster

vCluster Technology

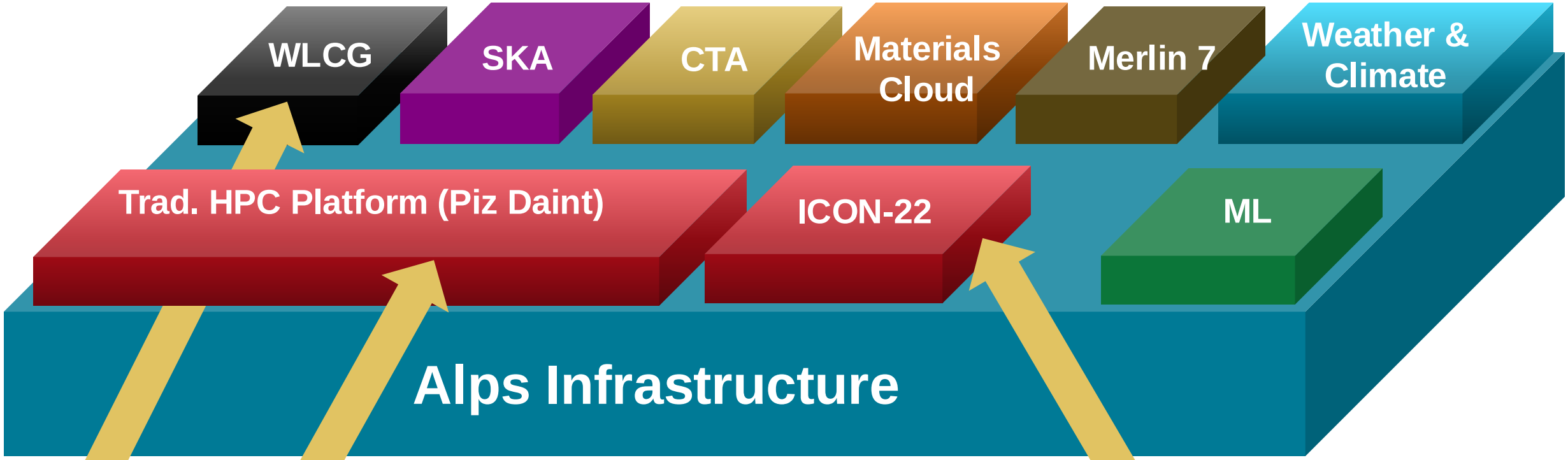
- The vCluster Technology is a set of components and practices that allows CSCS to “partition” one or more computing systems into logical units called Platforms and vClusters

- A Platform is a collection of 1 or more vClusters

- A vCluster is a collection of resources and services grouped for a common set of use-cases (e.g., ML community, HPC community, WLCG, etc.)

- Relies on DevOps concepts, mixing ideas from both the cloud world, and the traditional HPC environment

Consolidation of platforms

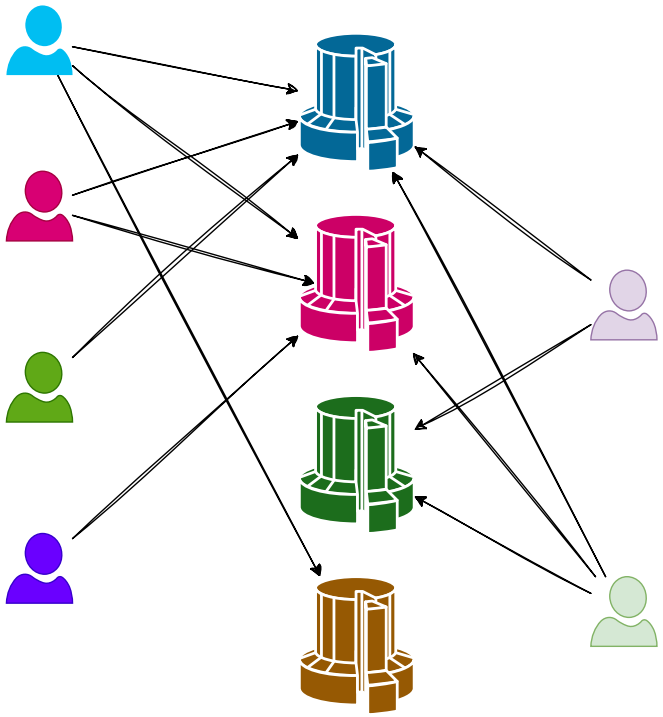


Piz Daint
User lab and WLCG

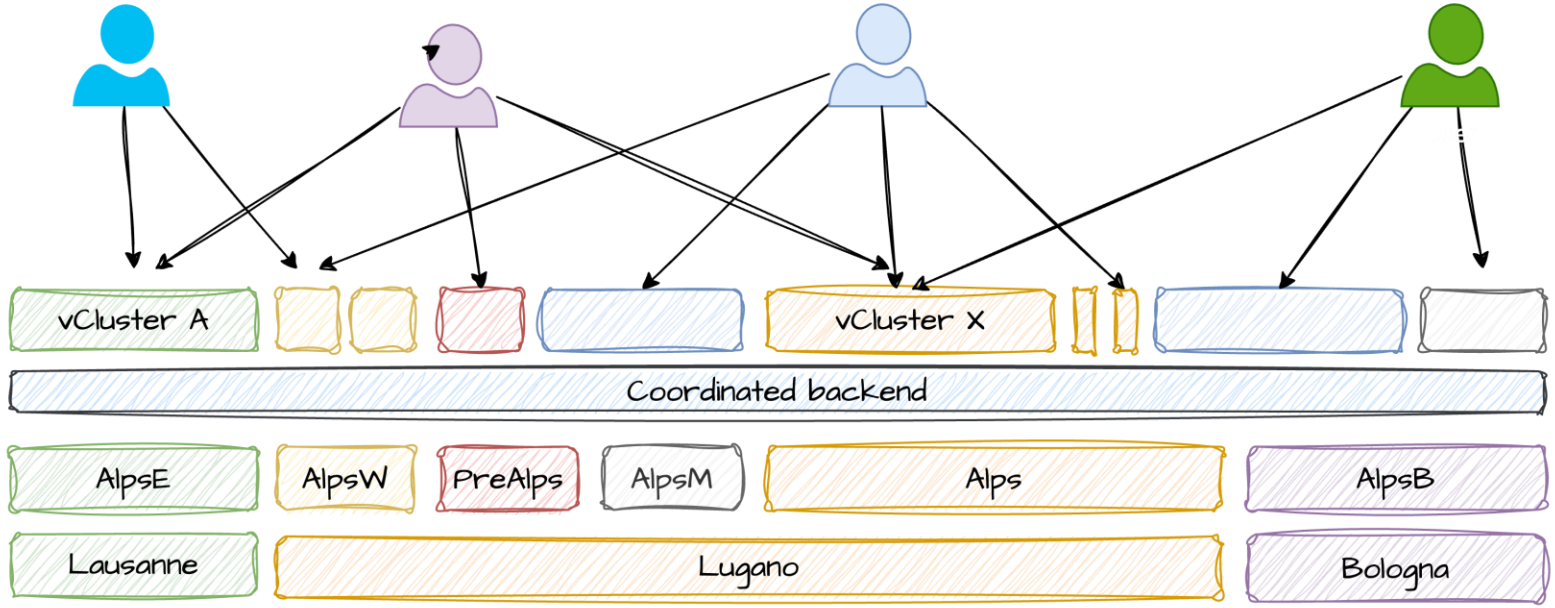
MeteoSwiss
Arolla/Tsa
COSMO/ModInterim

vClusters in context

- Instead of having discrete HPC systems, one per community group:

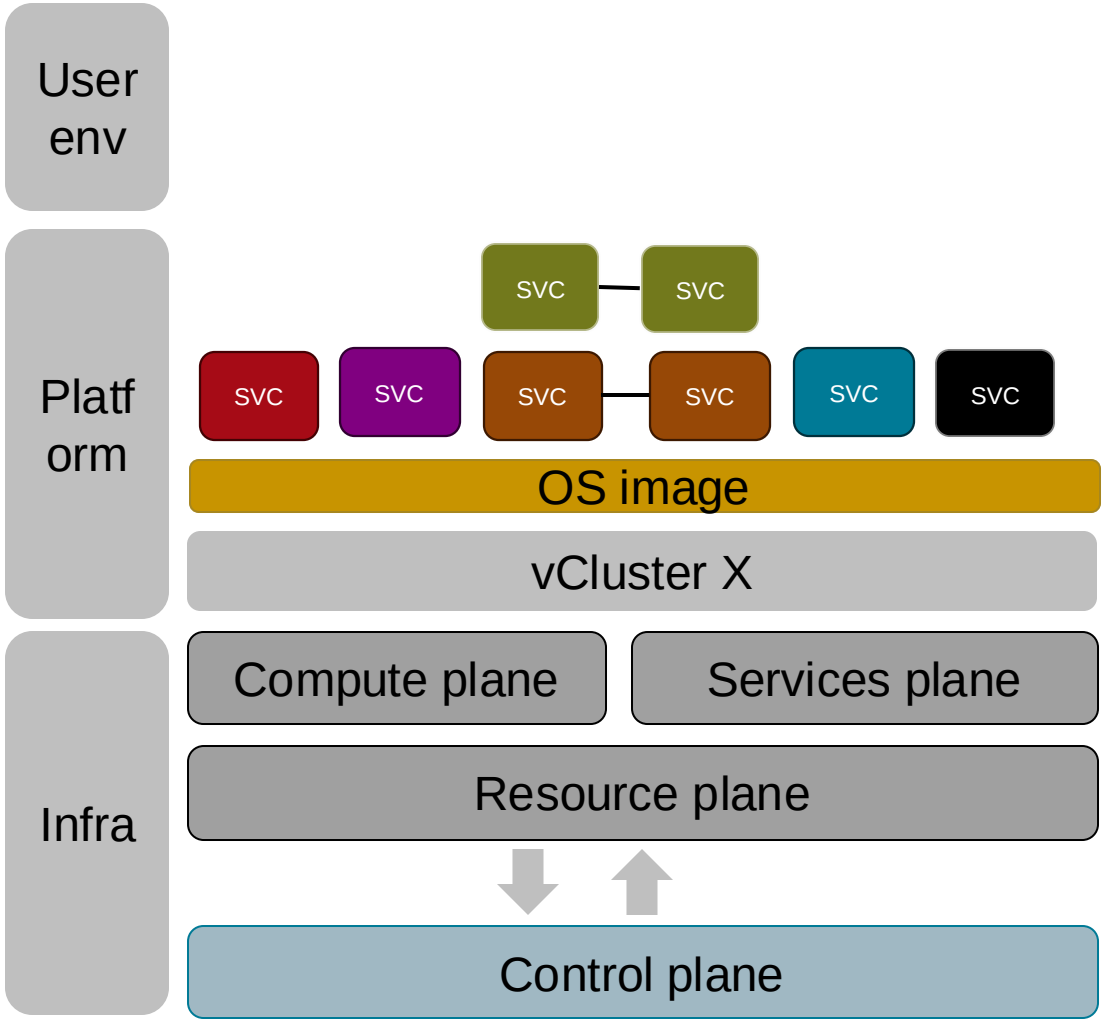


- To a distributed environment with multiple infrastructures acting as a coordinated backend to run multiple vClusters



Service management

Composability and modularity

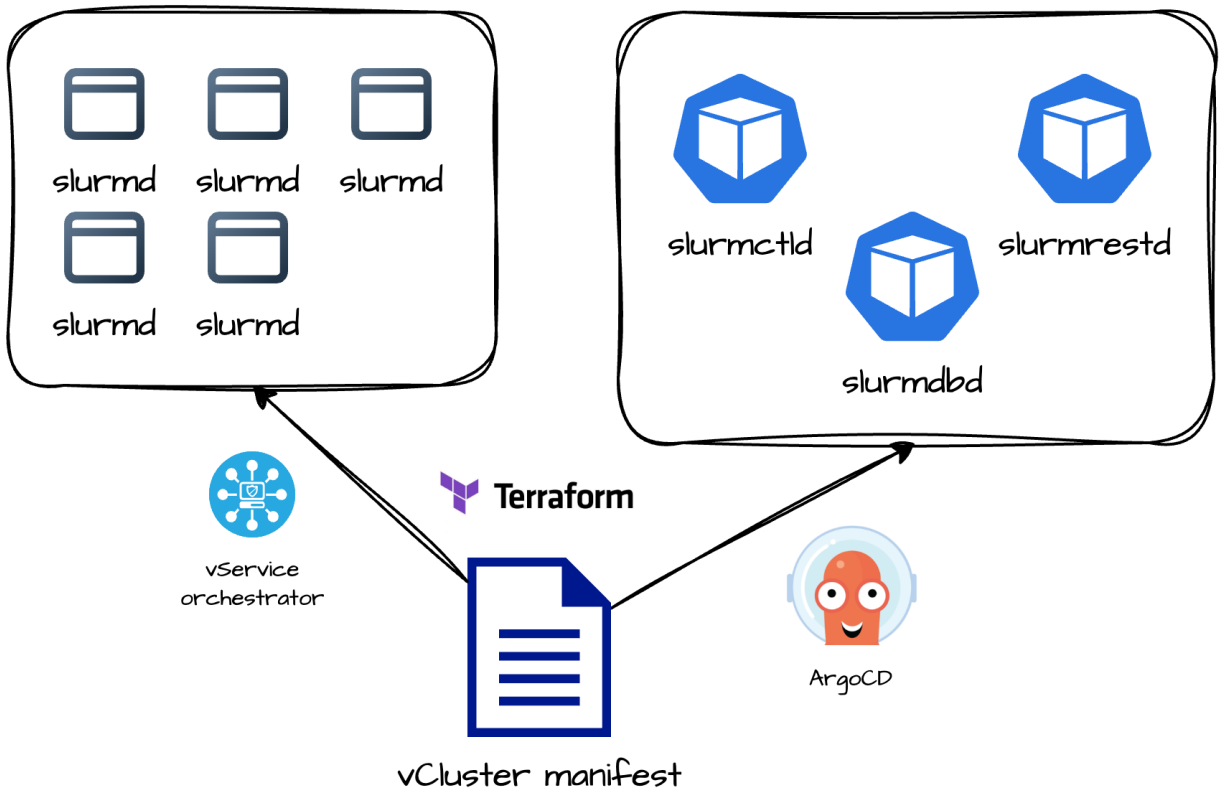


- A vCluster is a collection of resources on top of hardware
- On those resources, we run an OS image validated by infra team, and collection of services, named vServices
- Those vServices can span across multiple planes, e.g.,:
  - Compute plane
  - Services plane
- An example:
  - Slurmd runs on the compute plane
  - Slurmctld (+ SlurmDBD + SlumRESTd) run on the services plane

Example: Slurm vService

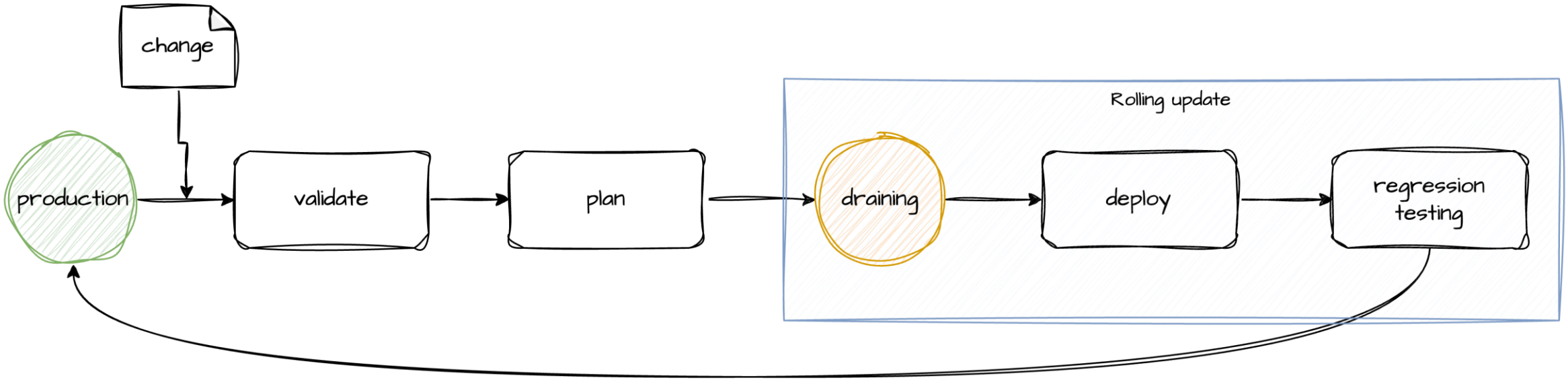
- Services on both planes are declared in a single manifest, and provisioned **as a single entity**:

- On the compute plane, our vService orchestrator ensures that the relevant Slurm artifacts for a given release (packages, config files, etc.) are deployed and running.
- On the services plane, ArgoCD deploys the necessary helm charts, pods (potentially even VMs), for the selected product release.



What makes this special?

- Simplified view of the configuration of a vCluster
- vServices and vClusters are products that follow common development, testing and validation processes
- This enables automated integration testing and rolling updates(*) with minimal human intervention



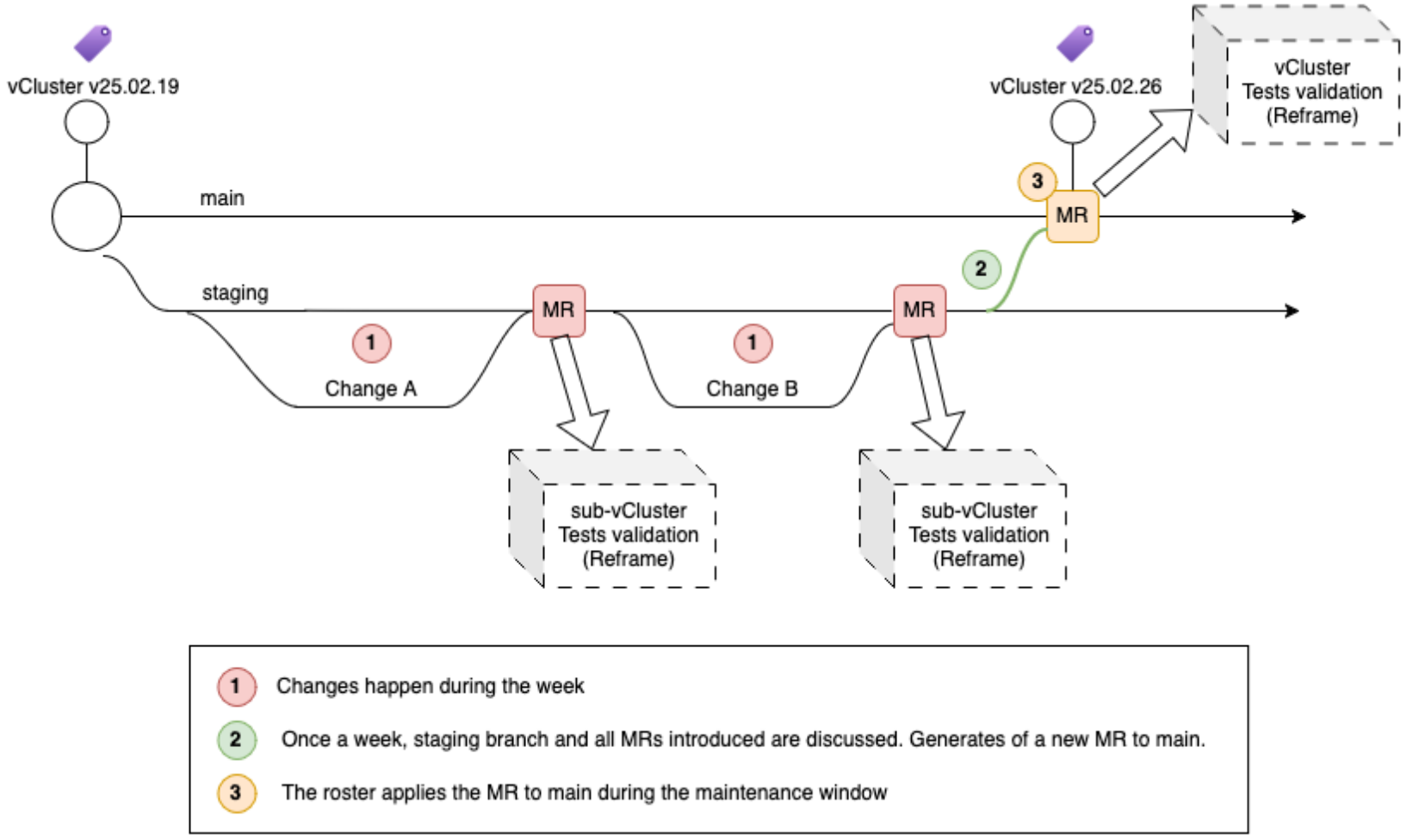
(*) work in progress

Simplified manifest

```
34  module "cscs-config" {
35    source       = "git@git.cscs.ch:alps-platforms/vservices/vs-cscs-config.gi?ref=v1.0.1"
36    deploy       = true
37    vcluster     = module.vcluster
38    hsm_groups   = "daint"
39    ansible_vars = file("platform/config/cscs-config/vars.yml")
40    ccm_version  = "cscs-24.8.0"
41    playbook     = "site.yml"
42  }
43
44  module "storage" {
45    source            = "git@git.cscs.ch:alps-platforms/vservices/vs-storage.git?ref=v1.0.6"
46    deploy            = true
47    vcluster          = module.vcluster
48    ansible_vars      = file("platform/config/storage/vars.yml")
49    node_dependencies = ["cscs-config"]
50
51    capstor_scratch_cscs_state  = "mounted"
52    capstor_store_cscs_state    = "mounted"
53    capstor_users_cscs_state    = "mounted"
54    iopsstor_scratch_cscs_state = "mounted"
55    iopsstor_store_cscs_state   = "mounted"
56    vast_users_cscs_state       = "mounted"
57  }
58
```

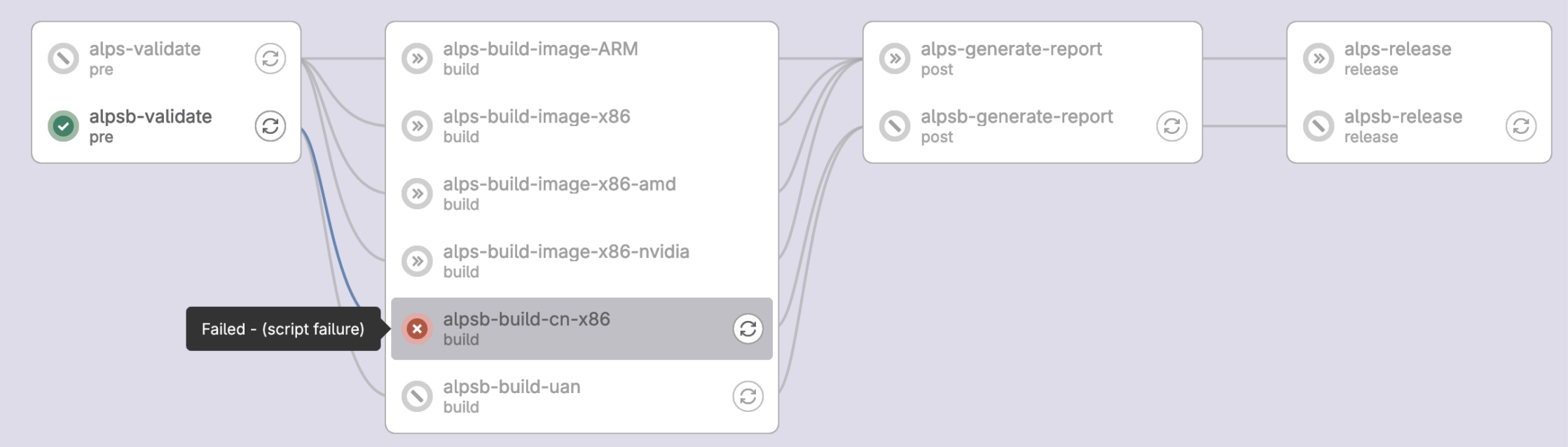
Introducing changes with full test coverage

- GitFlow
- Using and abusing Git and pipelines for **any** change
- Tests happen with each MR
- Service changes are orchestrated in a rolling fashion
- Minimal human intervention



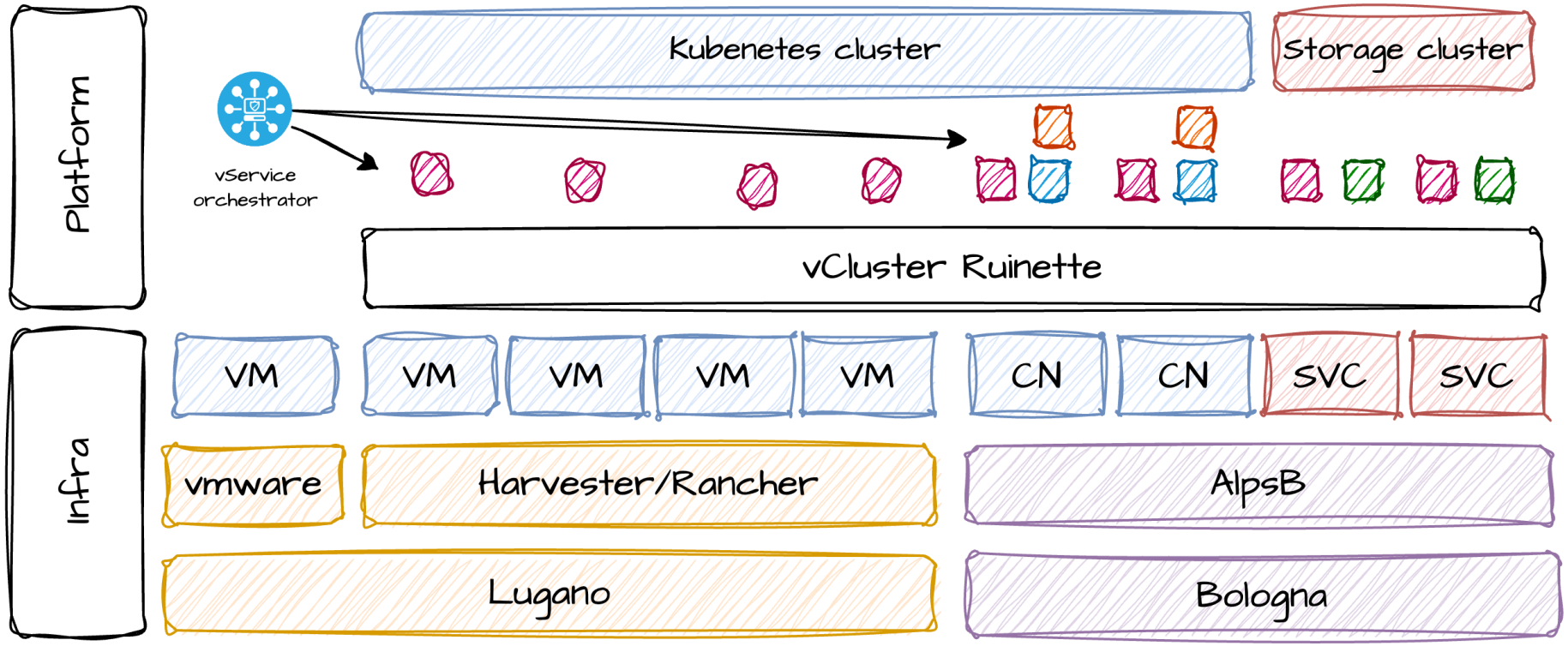
Example: building a vCluster image

- vCluster images are generic: one per infra and hardware type
- Built automatically by a pipeline using **manta**
- Ready for OpenCHAMI



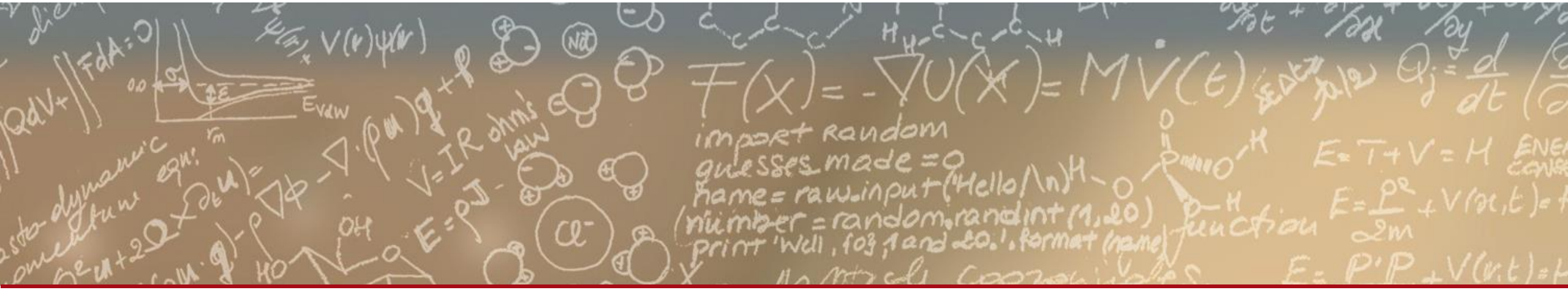
Example: vCluster Ruinette

- Best example of a vCluster: some components live in Lugano, others in Bologna, all are orchestrated with the vCluster technology



vClusters and vServices in numbers

- Currently have around 20 vServices in active development (and usage)
  - Core: IAM, Storage, Container Engine(s), **Network**, Slurm, **Security**
  - Uenv, Firecrest, Observability, Validation, Health Engine etc.
- ~900 MRs in the last two years
- A given vCluster operates with a manifest of about 100 lines of code
- 16 vClusters in operation with this model
- About 8 more that will transition to it in 2025 Q2
- After this, Alps alone will have about 50k vService instances running



Thank you for your attention.