CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

Alps, a versatile research infrastructure

CUG 2025

Maxime Martinasso, CSCS

Everyone at CSCS and partners -- now and before!

May 2025

Motivation: HPC limitations

Source: Ben Cumming!

- Vertically integrated stack
  - Updates are bundling many components together
  - Hard to identify the source of new problems
  - High cost in re-building and re-validating scientific code
- Monolithic service offering
  - The stack includes services like Slurm
  - Vendor support for those services, no flexibility in changing it
  - Adding new services is a complex endeavours
- Operational constraints
  - Release of fixes into bundles, delays
  - Using workaround, keep track of update fix
  - Downtime of the systems

Alps objectives

- Provide data and compute capability for science (new flagship capability)
- Develop versatile and composable platforms to answer science needs
- Manage custom service offerings together with the scientific communities
- Use HPC+Cloud technology to manage composability and service updates

Alps, a versatile research infrastructure

Alps technology

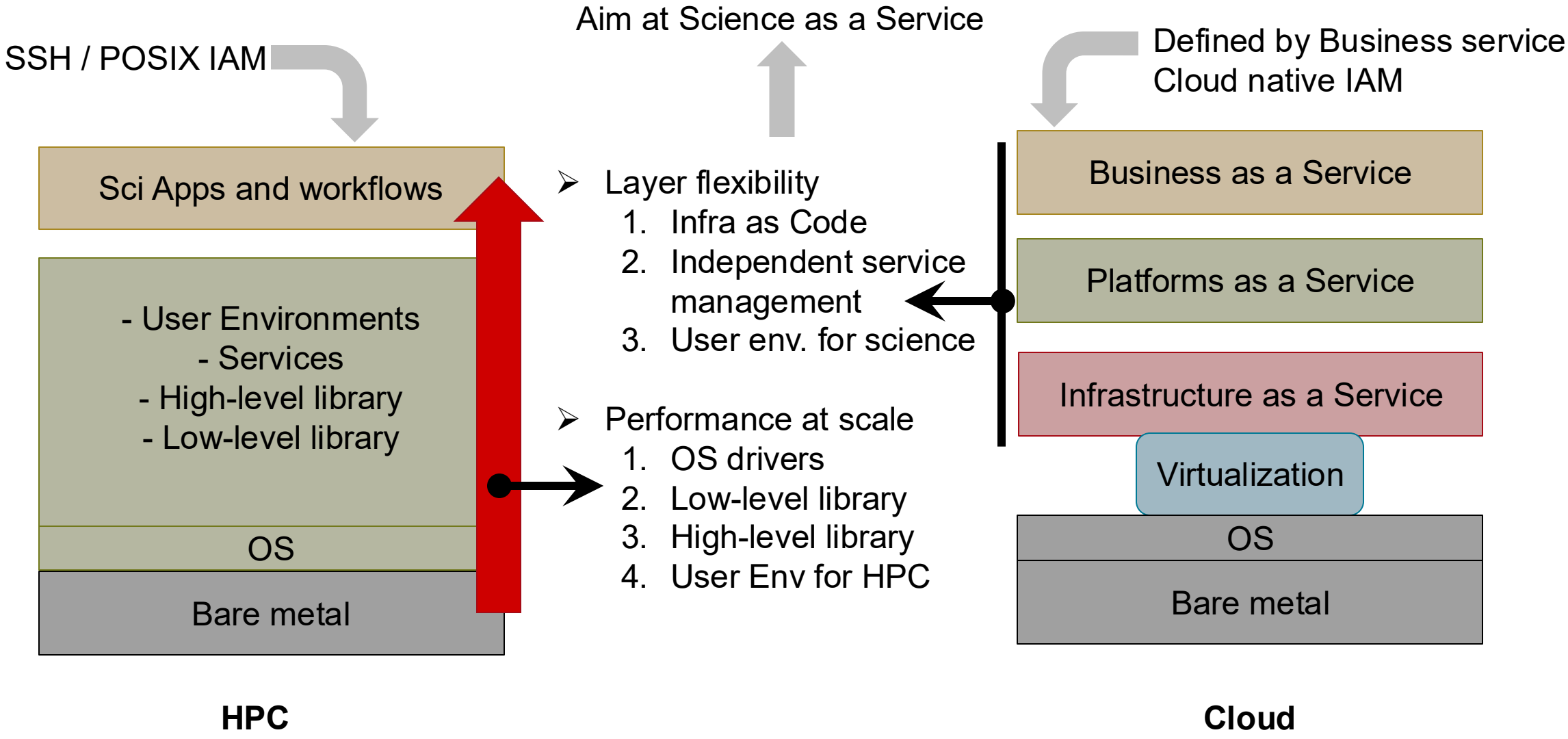
Alps heterogeneous hardware

- Alps is an HPE Cray EX supercomputer being our new flagship infrastructure

  - 1024 AMD Rome-7742 nodes 256/512GB
  - 144 Nvidia A100 GPU nodes
  - 24 AMD MI250x GPU nodes (LUMI1 type)
  - **128 AMD MI300A GPU nodes**
  - ***2688 Grace-Hopper nodes, 10752 GH200, ~430 PF***
  - Slingshot network (200 Gbps injection)
  - Two availability zones (HA, non-HA)
  - 100% liquid cooled
  - 100+10 PiB HDD
  - 5+1 PiB SSD (RAID10)
  - 100s of PiB tape library
  - ~10 MW (envelope for power and cooling)

Water cooled blades

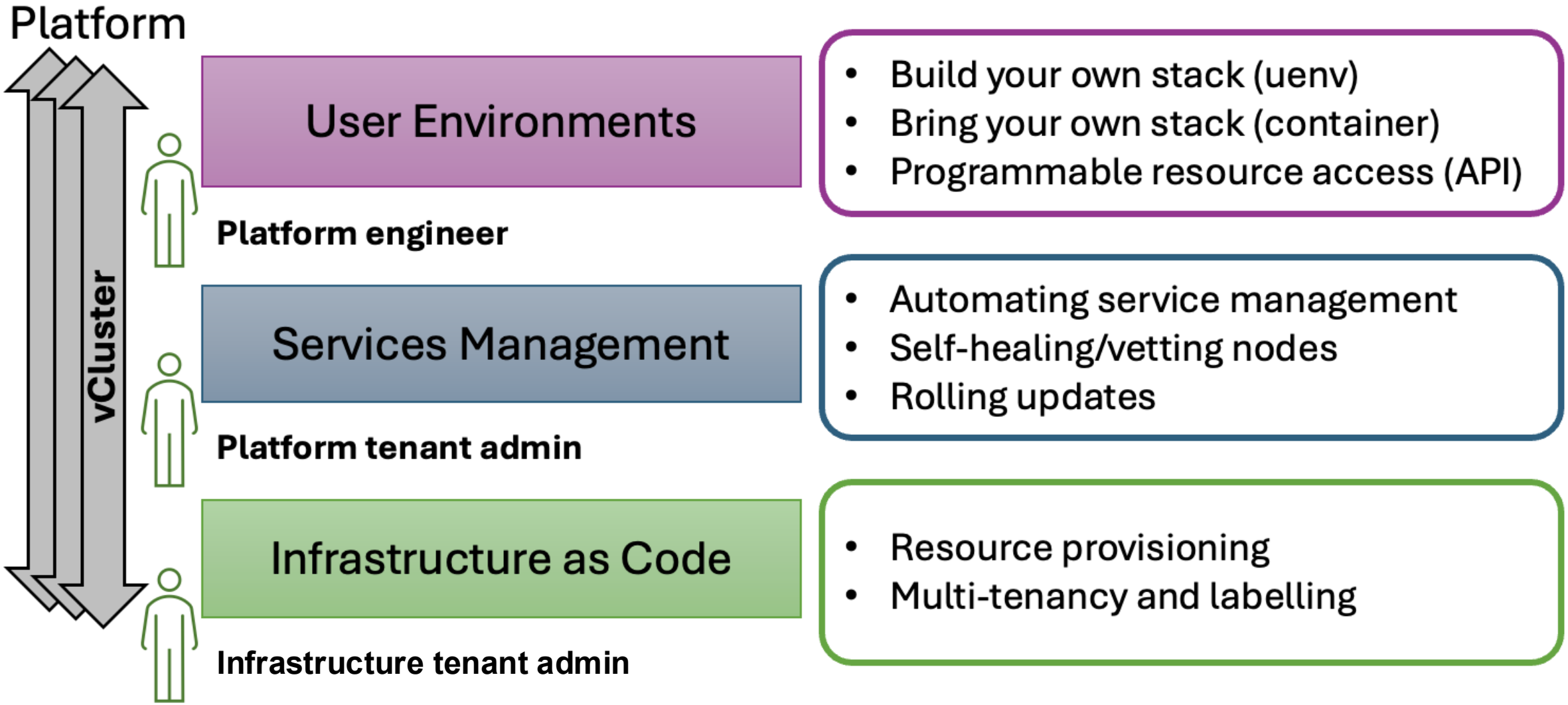
HPC and cloud convergence



vCluster concept

- Versatile software defined cluster (vCluster)
  - Three layers separation of concerns: infrastructure, service mgmt., user environments
  - Resources, service sets and configurations define in files
  - Pipelines create, test and deploy vClusters and the services (vServices)
  - Automate and facilitate the life cycle of those services using a cloud orchestrator
- A scientific platforms is a set of vClusters

Three layers **concept**



Infrastructure as code

- Multi-tenancy, resource grouping and provisioning
  - Compute node labelling
  - Base image with minimal customisation
  - Storage is not multitenant, directory mount on Lustre, or dedicated systems
  - Network segregation when needed using VLAN
- Interface to system management
  - Manta is a vendor-agnostic interface to system management with RBAC
    - Interfacing CSM and OpenCHAMI
    - RBAC controls actions on specific labelled resources
- Immutable infrastructure
    - Actions on the system are done “exclusively” with pipelines
    - Pipelines use Manta, GitOps process

Services management

- vCluster manifest
  - List of vServices
  - ~10 vServices per vCluster
  - vServices:
    - SSH/LDAP
    - Storage
    - Uenv
    - Container engine
    - Slurm
    - Node health
    - …

vService definition in the manifest

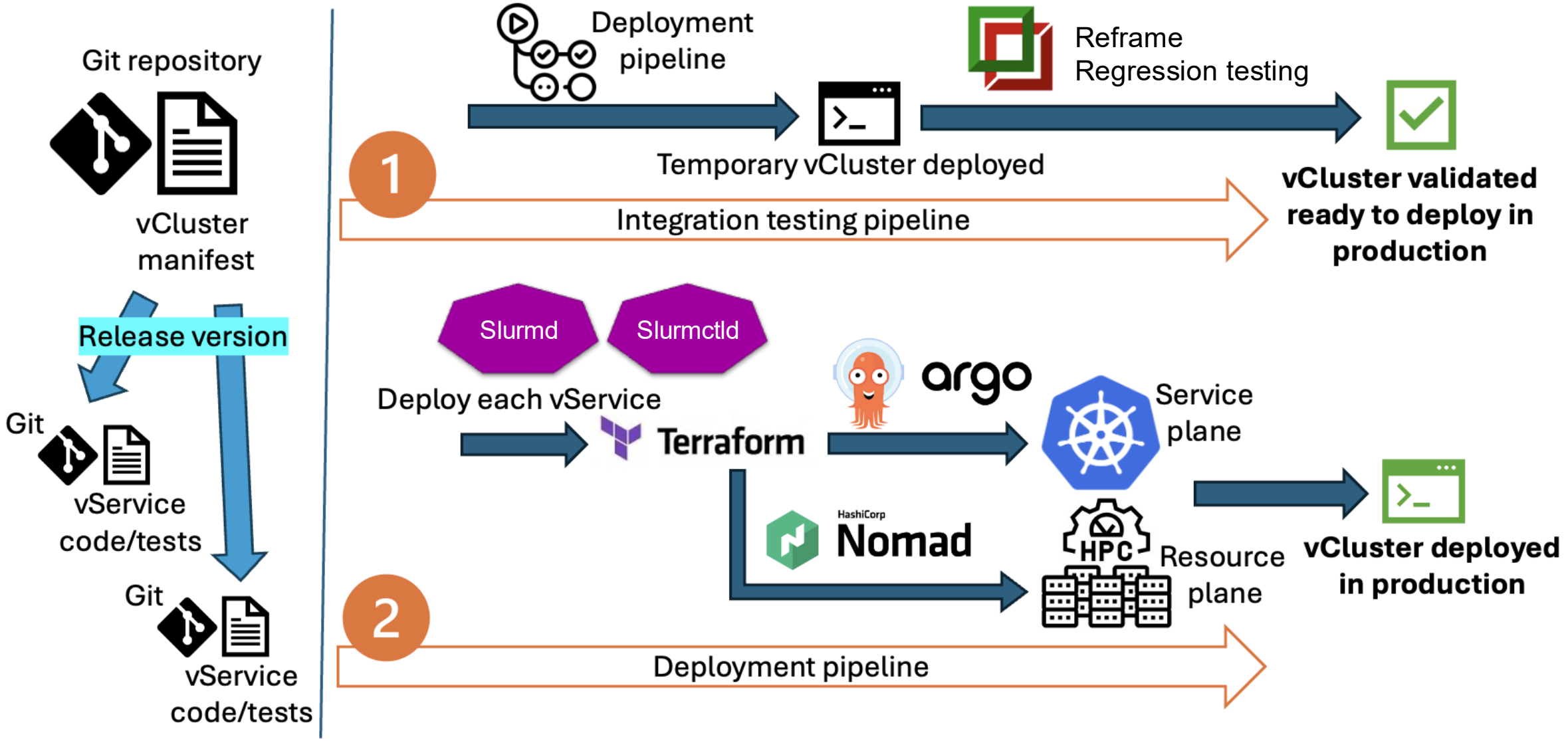
Resource label to identify compute nodes

```
34  module "cscs-config" {
35    source       = "git@git.cscs.ch:alps-platforms/vservices/vs-cscs-config.git?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  }
```

vService code and configuration

vService version

vCluster deployments

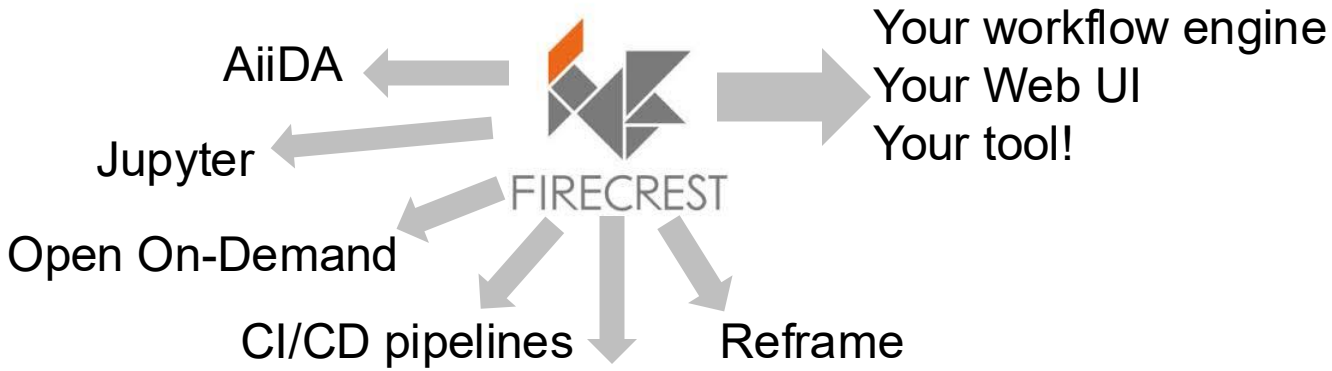


Service management

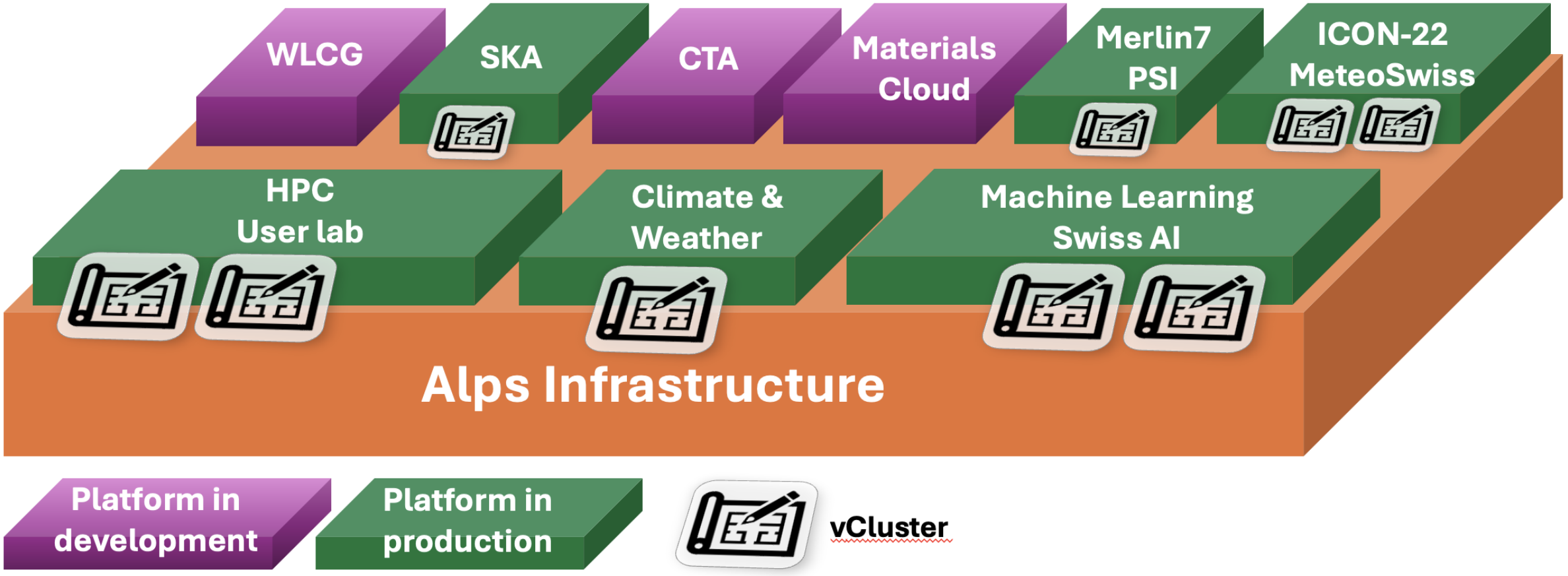
1. Automating service managements
   - Manifest, code, tests, configuration in versioned in Git
   - Composability, select your services and versions and do a merge request
   - Pipelines deploy updates triggered from merges
2. Self healing/vetting nodes
   - vService node health which monitor at job starts and ends
   - Pipeline actions in case of issues, use Manta to reboot nodes
   - If not recovered, remove node from scheduler and service orchestrator
3. Rolling updates
   - Adding, updating version or removing a service is done by the service orchestrator
   - Enable frequent adaptation without user disruption
   - Extra care for certain vServices like Slurm (in the code of the vService)
   - Not at the infrastructure level, network or system mgmt updates are not rolling!

User environments

- Build your own stack
  - Provide **uenv** tool for communities to manage their own PE and applications
  - CI external from GitHub
- Bring your own stack
  - Use **container** to enable external stack with access to HPC hardware with OCI hooks
- Programmatic access
  - Web-facing RESTful **API** to move data and submit jobs



Scientific platforms



Operation and organization

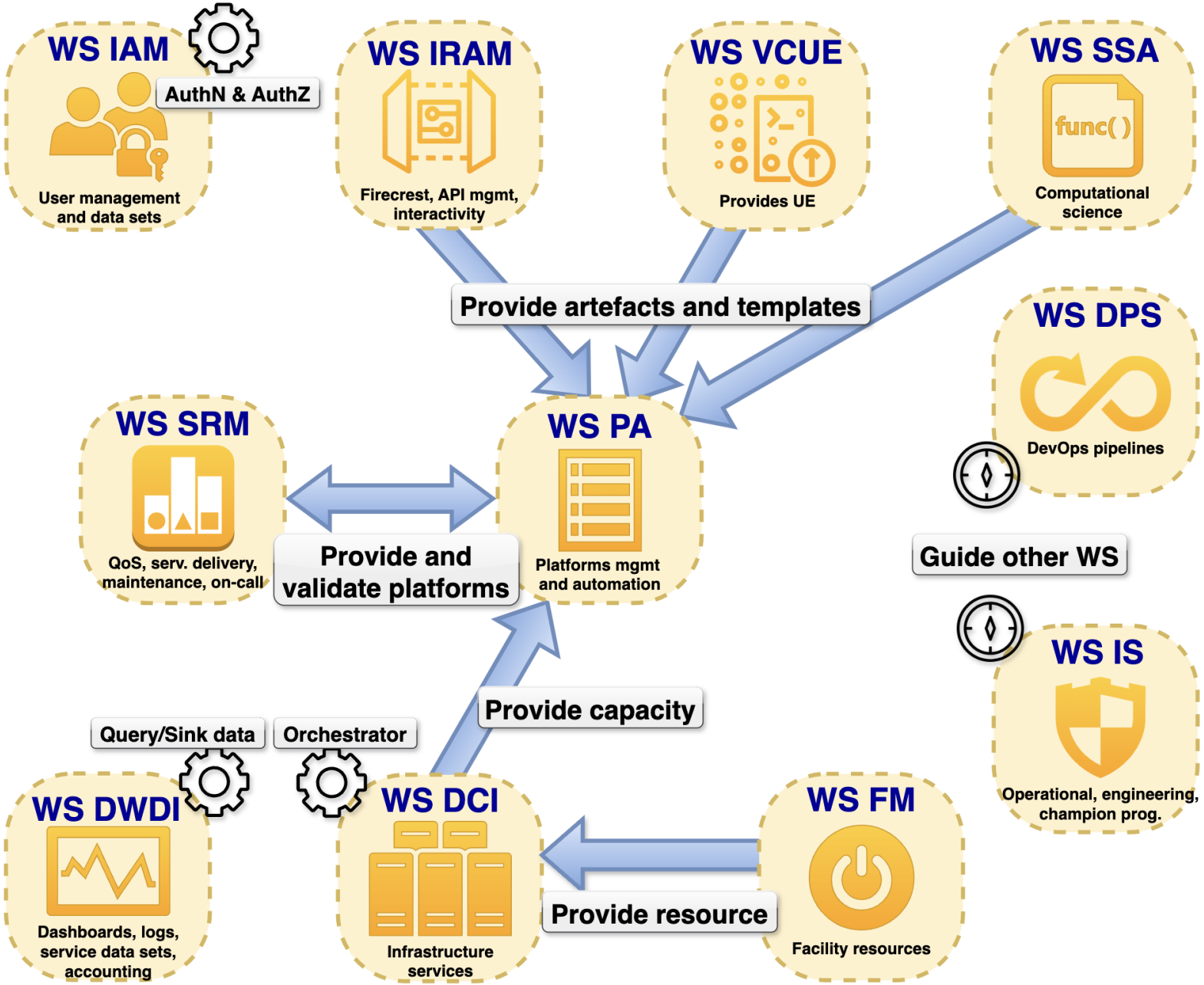
New team organization

Home of 100 engineers

Working Structure (WS)
- Dynamic teams deliver technical solution of one domain
- Designed to change and evolve

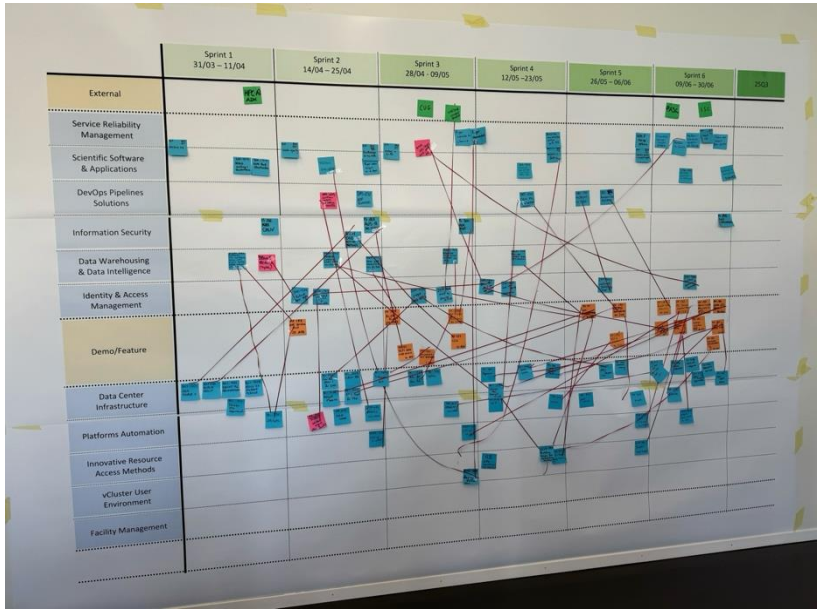
End-to-end responsibility
- Solution development
- Service ownership

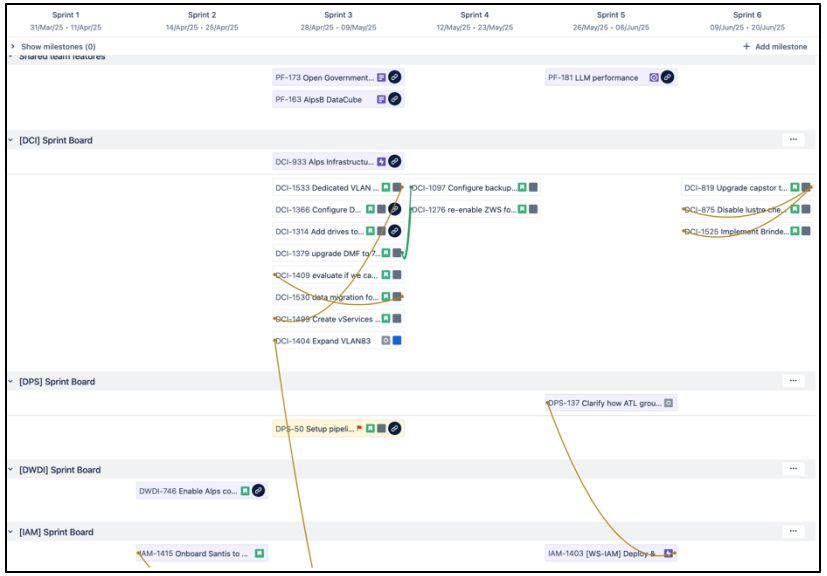
They are responsible on how to build and the quality of what they build



Agile process

- Follows the Scaled Agile framework
- Quarterly planning process
- Every quarter all engineers and key partners meet for 2 days to plan the activities
- Priorities of the quarter are detailed and explained before the meeting
- Risk assessment and confidence vote of the planning
- During the quarter, checkpoints with demos are scheduled to ensure progress and adaptability
- Work with synchronised sprint of 2 weeks





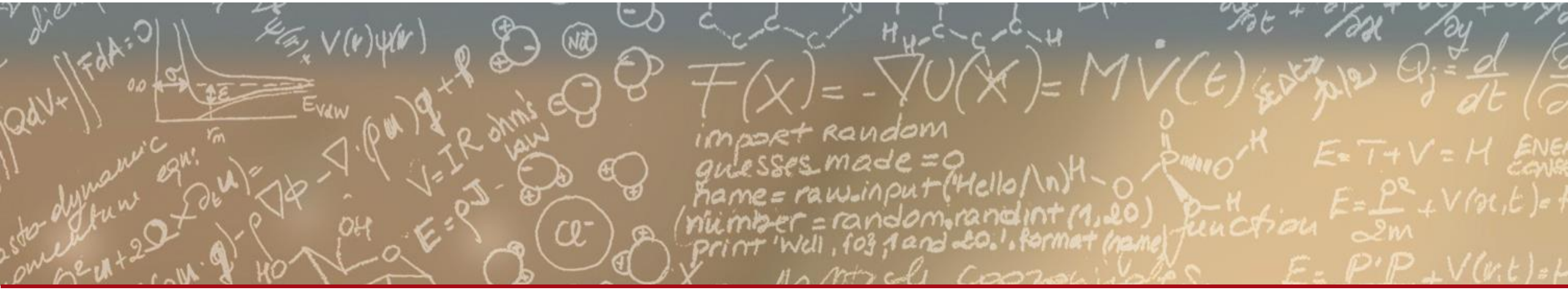
Discussion

- Versatility vs complexity
  - Build a new technology, a new organisation and pioneering new hardware at scale
  - More platforms, more services, more teams, more coordination increased complexity
  - Need engagement to understand versatility and collaboration to operate it
- Operational experience
  - Cultural shift to get to immutability and GitOps process
  - Life cycle management instead of operations
  - Software practices – testing and debugging
- Automation: promise vs. practice
  - Automation brings many benefits, version control, reproducibility, release notes
  - Rolling updates of services, learning and collaboration with partners and vendors
  - Complex debugging, need to understand all the elements and gather the logs
- Sustainability of the vCluster technology
  - Good software practices, avoid building a technology debt
  - Will be made Opensource and transferable to other sites

Future work

- Harden and opensource the vCluster technology
- Introduce new platforms and vClusters
- Elastic resource exchange among vCluster (IPDPS - JSSP Milan, June 3rd)
- Multitenant storage management
- Multi sites infrastructure management
- Move system management to OpenCHAMI for Alps

Continue this fantastic journey!



Thank you for your attention.