Hewlett Packard
Enterprise

# Software-defined Multi-tenancy on HPE Cray Ex Supercomputers

Jeremy Duckworth, Vinay Gavirangaswamy, David Gloe, Brad Klein

May 2023

## Background

Multi-tenancy and related concepts

Multi-tenancy applied to HPC

## CSM's Multi-tenancy Solution

Programmable infrastructure

## Future Work

Exploring PaaS vs. IaaS

Exploring predicate-based tenant scheduling

Exploring HPC storage multi-tenancy

# Background

Multi-tenancy and related concepts, applications in HPC

> **“Multi-tenancy is a property of a system where multiple customers, so-called tenants, transparently share the system’s resources, such as services, applications, databases, or hardware, with the aim of lowering costs, while still being able to exclusively configure the system to the needs of the tenant.”**

KABBEDIJK, JAAP,ET AL."DEFININGMULTI-TENANCY:ASYSTEMATIC MAPPING STUDY ON THE ACADEMIC AND THE INDUSTRIAL PERSPECTIVE." JOURNAL OF SYSTEMS AND SOFTWARE 100 (2015): 139- 148.

# Related Terms and Concepts

**Multi-instance**

**Elasticity**

**“Soft” Multi-tenancy**

**“Hard” Multi-tenancy**

- A variation of multi-tenancy where tenants use exclusive, not shared, resources

# Related Terms and Concepts

Multi-instance

**Elasticity**

"Soft" Multi-tenancy

"Hard" Multi-tenancy

- A system property where resources can be rapidly added or removed from a tenant's resource pool to address changes in workload demand and manage costs

# Related Terms and Concepts

**Multi-instance**

**Elasticity**

**“Soft” Multi-tenancy**

**“Hard” Multi-tenancy**

- Multi-tenancy implementations that are optimized towards accident prevention and agility versus strict resource isolation (canonically cybersecurity focused).

# Related Terms and Concepts

**Multi-instance**

**Elasticity**

**“Soft” Multi-tenancy**

**“Hard” Multi-tenancy**

- Multi-tenant implementations that are optimized towards strict isolation (canonically cybersecurity focused) versus just accident prevention and agility
  - For example, to meet multi-level security requirements or performance SLAs

# Multi-tenancy: Potential Strengths and Weaknesses

| Strengths | | Weaknesses | |
|---|---|---|---|
| **General** | **HPC** | **General** | **HPC** |
| Cost Savings (provider, consumer) via shared hardware, economies of scale, and energy efficiency<br>Elasticity<br>Accelerated access to the latest technology<br>Support for multiple security trust domains per system to meet multi-level security (MLS) requirements | Rapid, software-based reconfigurability, at scale<br>Logical Test and Development Systems, support for blue/green deployments | System Complexity<br>Limiting change impact across tenants | Requires "advanced" security in design for MLS OR for delivering solutions at the lowest levels of the aaS model (i.e., IaaS) |

# As a Service: Example layers, top to bottom

| Layer | Description |
|---|---|
| Workflow as a Service (WaaS) | Hosted workflows, like those that can be modeled as Directed Acyclic Graphs (DAGs) |
| Function as a Service (FaaS) | Hosted functions (e.g., serverless computing) |
| Database as a Service (DBaaS) | Hosted databases |
| Software as a Service (SaaS) | Hosted applications |
| Platform as a Service (PaaS) | Hosted operating systems |
| Infrastructure as a Service (IaaS) | Hosted virtual machine or bare metal servers |

# As a Service: Shared Responsibility Models

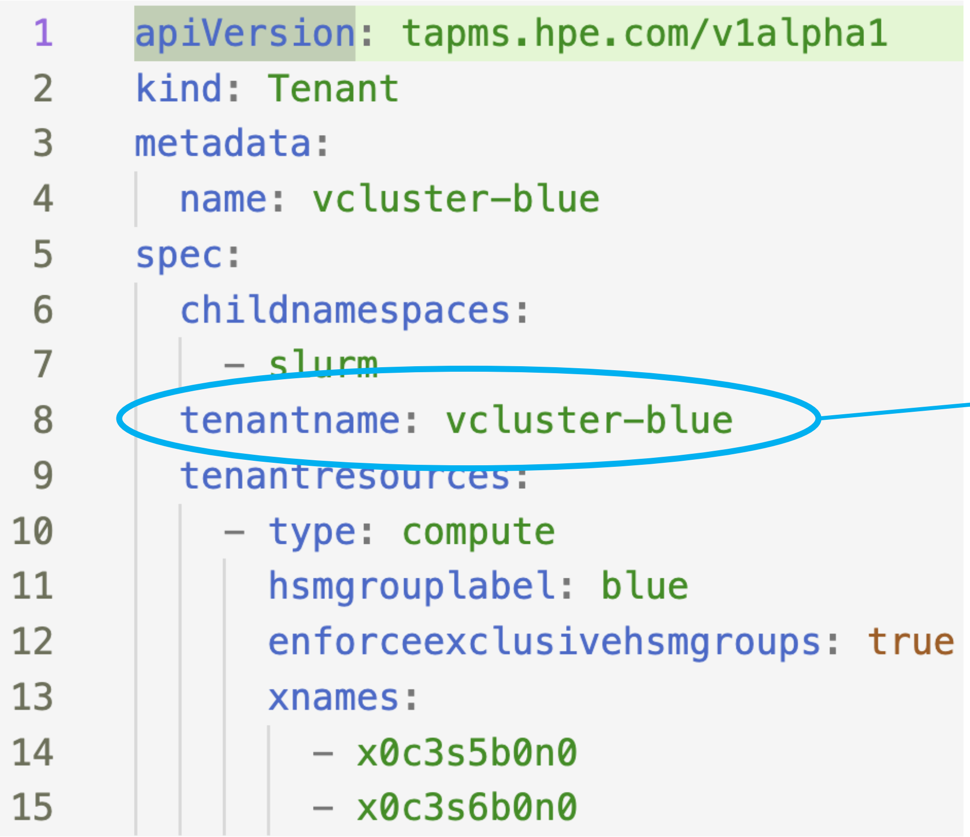
- Operational risks are shared amongst all parties involved in delivery or consumption of the service.
- Parties typically require contractual agreements to protect shared interests and limit liability should events like a security breach or data loss occur. These agreements are often based on a shared responsibility model.

# CSM’s Multi-tenancy Solution

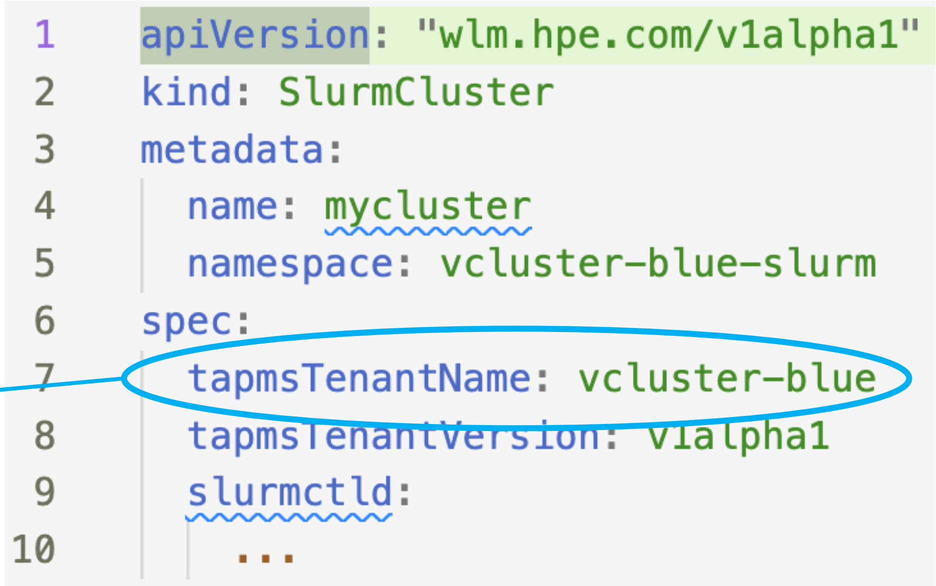
Programmable infrastructure through declarative configuration

# Cooperative Operators for Tenant Provisioning

Purpose built, composable K8s operators that provide a multi-tenancy substrate



```yaml
apiVersion: tapms.hpe.com/v1alpha1
kind: Tenant
metadata:
  name: vcluster-blue
spec:
  childnamespaces:
    - slurm
  tenantname: vcluster-blue
  tenantresources:
    - type: compute
      hsmgrouplabel: blue
      enforceexclusivehsmgroups: true
      xnames:
        - x0c3s5b0n0
        - x0c3s6b0n0
```

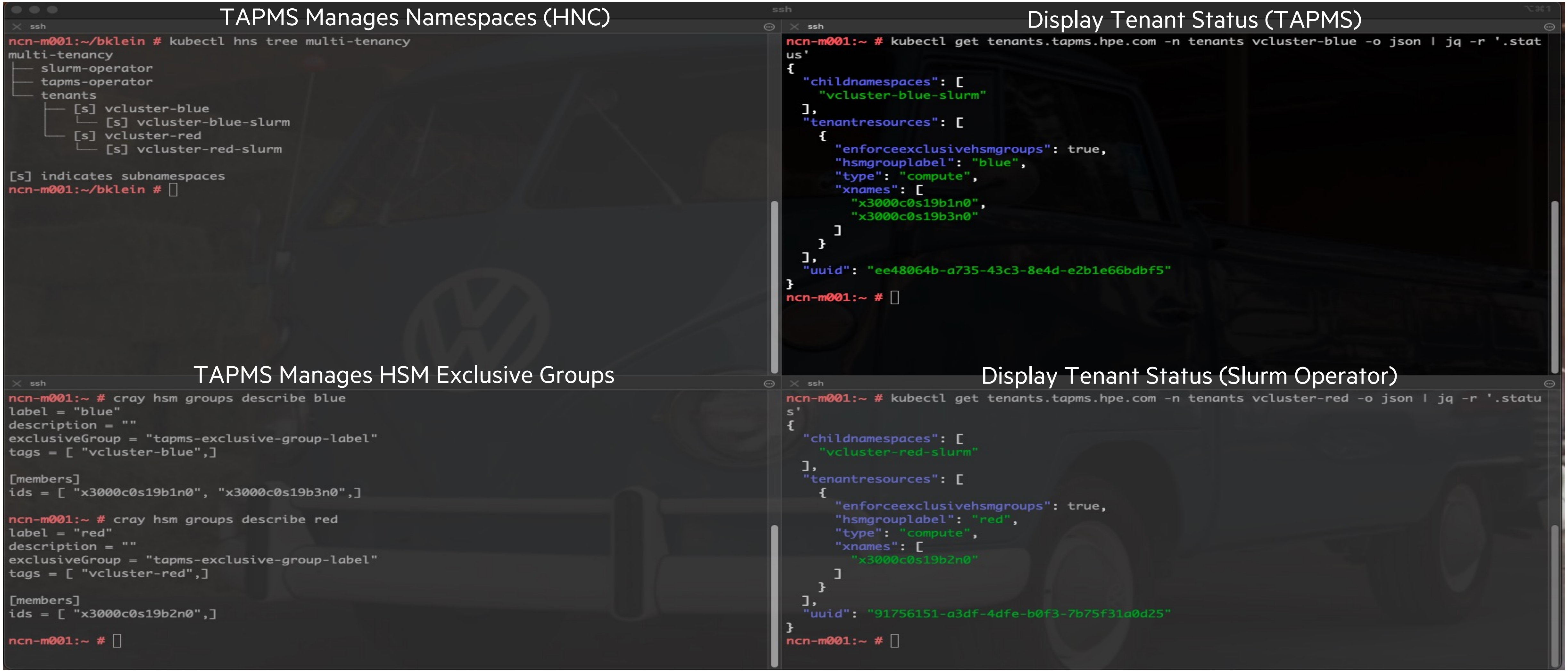


```yaml
apiVersion: "wlm.hpe.com/v1alpha1"
kind: SlurmCluster
metadata:
  name: mycluster
  namespace: vcluster-blue-slurm
spec:
  tapmsTenantName: vcluster-blue
  tapmsTenantVersion: v1alpha1
  slurmctld:
    ...
```

https://kubernetes.io/docs/concepts/extend-kubernetes/operator/
https://github.com/Cray-HPE/cray-tapms-operator
https://github.com/Cray-HPE/docs-csm/tree/release/1.3/operations/multi_tenancy

# Cooperative Operators for Tenant Provisioning

## Screenshot illustrating multiple provisioned tenants



TAPMS Manages Namespaces (HNC)

```
ncn-m001:~/bklein # kubectl hns tree multi-tenancy
multi-tenancy
├── slurm-operator
├── tapms-operator
└── tenants
    ├── [s] vcluster-blue
    │   └── [s] vcluster-blue-slurm
    └── [s] vcluster-red
        └── [s] vcluster-red-slurm

[s] indicates subnamespaces
ncn-m001:~/bklein # 
```

Display Tenant Status (TAPMS)

```
ncn-m001:~ # kubectl get tenants.tapms.hpe.com -n tenants vcluster-blue -o json | jq -r '.stat
us'
{
  "childnamespaces": [
    "vcluster-blue-slurm"
  ],
  "tenantresources": [
    {
      "enforceexclusivehsmgroups": true,
      "hsmgrouplabel": "blue",
      "type": "compute",
      "xnames": [
        "x3000c0s19b1n0",
        "x3000c0s19b3n0"
      ]
    }
  ],
  "uuid": "ee48064b-a735-43c3-8e4d-e2b1e66bdbf5"
}
ncn-m001:~ # 
```

TAPMS Manages HSM Exclusive Groups

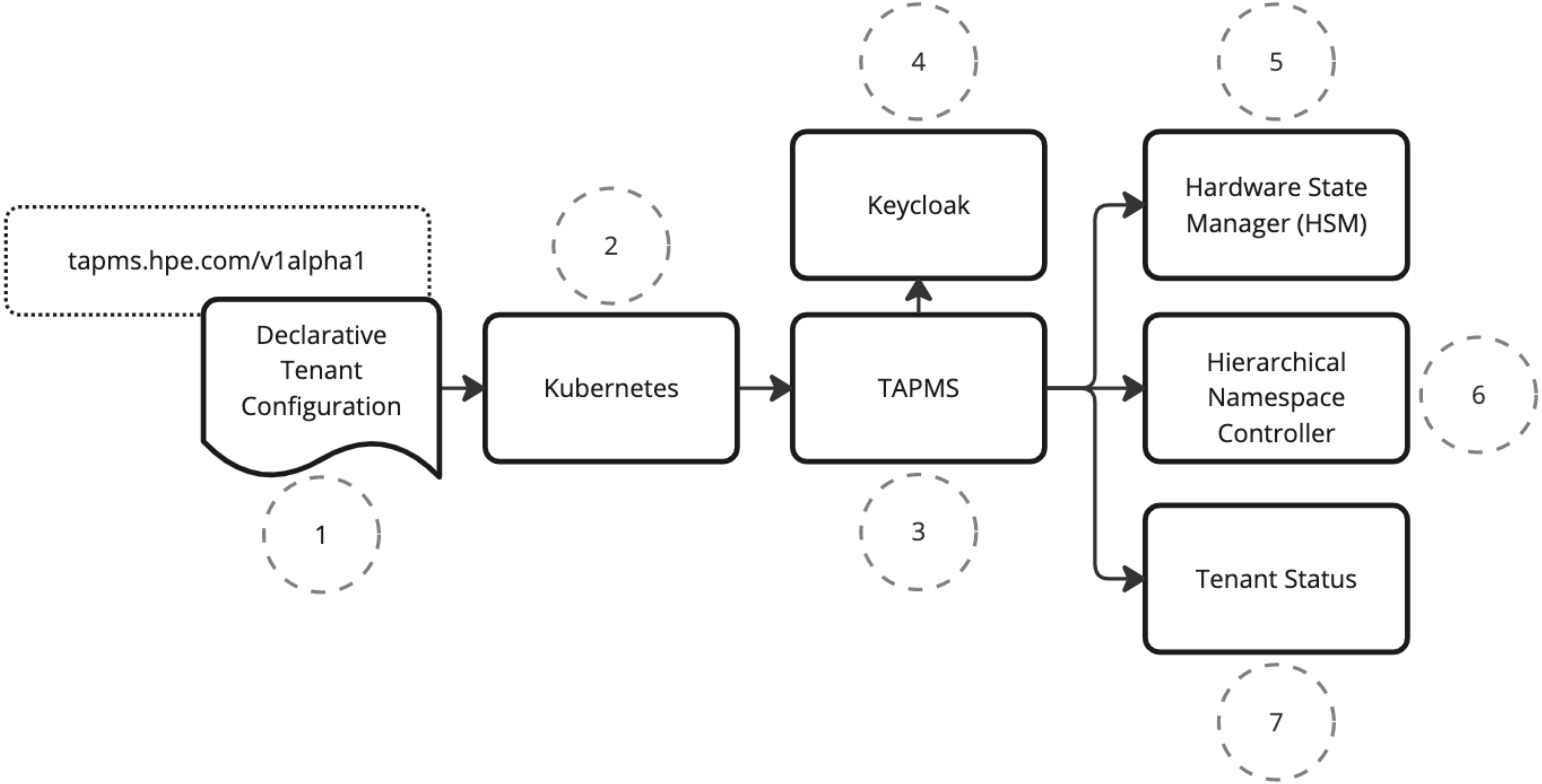
```
ncn-m001:~ # cray hsm groups describe blue
label = "blue"
description = ""
exclusiveGroup = "tapms-exclusive-group-label"
tags = [ "vcluster-blue",]

[members]
ids = [ "x3000c0s19b1n0", "x3000c0s19b3n0",]

ncn-m001:~ # cray hsm groups describe red
label = "red"
description = ""
exclusiveGroup = "tapms-exclusive-group-label"
tags = [ "vcluster-red",]

[members]
ids = [ "x3000c0s19b2n0",]

ncn-m001:~ # 
```

Display Tenant Status (Slurm Operator)

```
ncn-m001:~ # kubectl get tenants.tapms.hpe.com -n tenants vcluster-red -o json | jq -r '.statu
s'
{
  "childnamespaces": [
    "vcluster-red-slurm"
  ],
  "tenantresources": [
    {
      "enforceexclusivehsmgroups": true,
      "hsmgrouplabel": "red",
      "type": "compute",
      "xnames": [
        "x3000c0s19b2n0"
      ]
    }
  ],
  "uuid": "91756151-a3df-4dfe-b0f3-7b75f31a0d25"
}
ncn-m001:~ # 
```

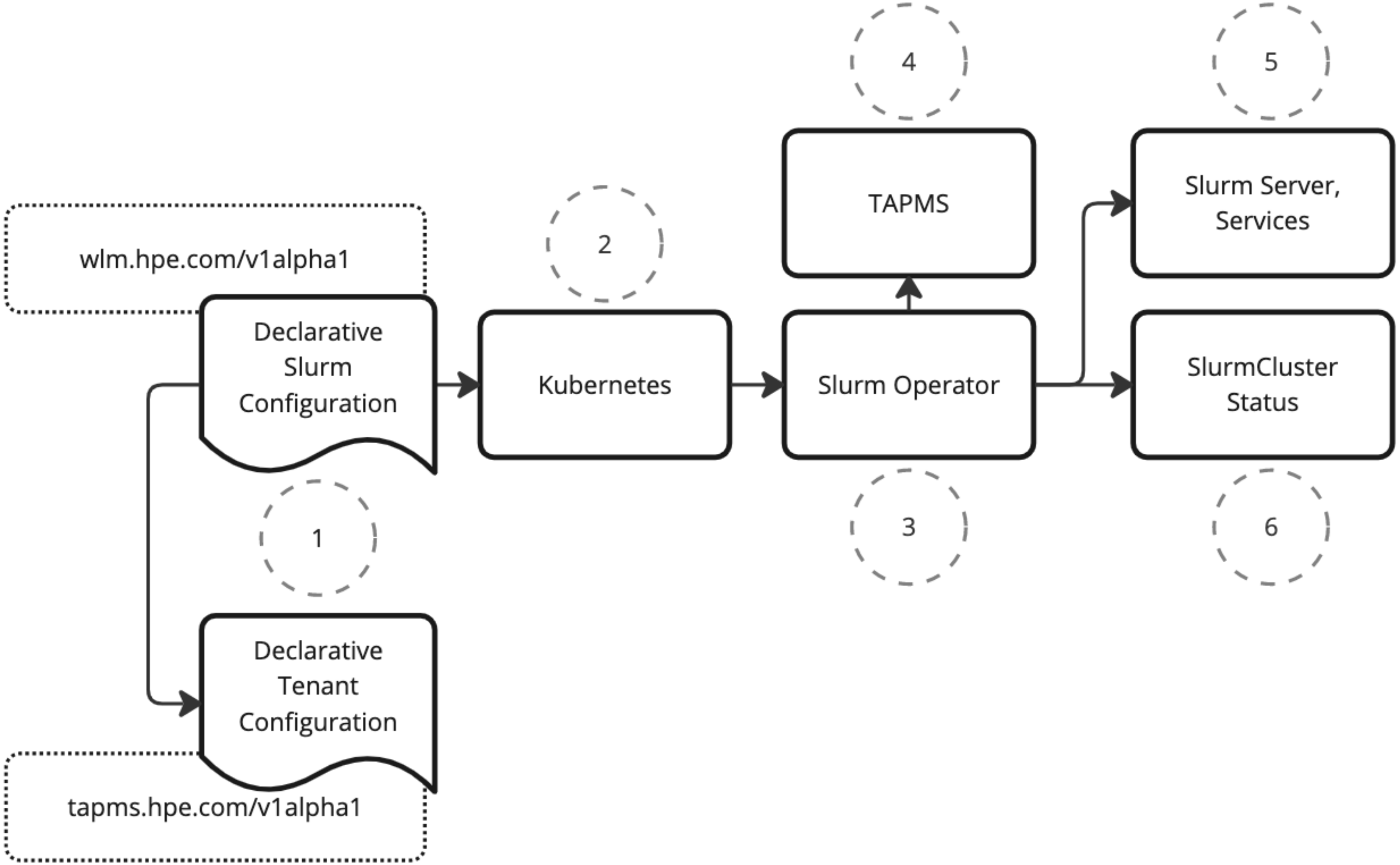
# Cooperative Operators for Tenant Provisioning

Provisioning Sequencing: Tenant and Partition Management System (TAPMS)


4
5
Keycloak
Hardware State Manager (HSM)
tapms.hpe.com/v1alpha1
2
Declarative Tenant Configuration
Kubernetes
TAPMS
Hierarchical Namespace Controller
6
1
3
Tenant Status
7

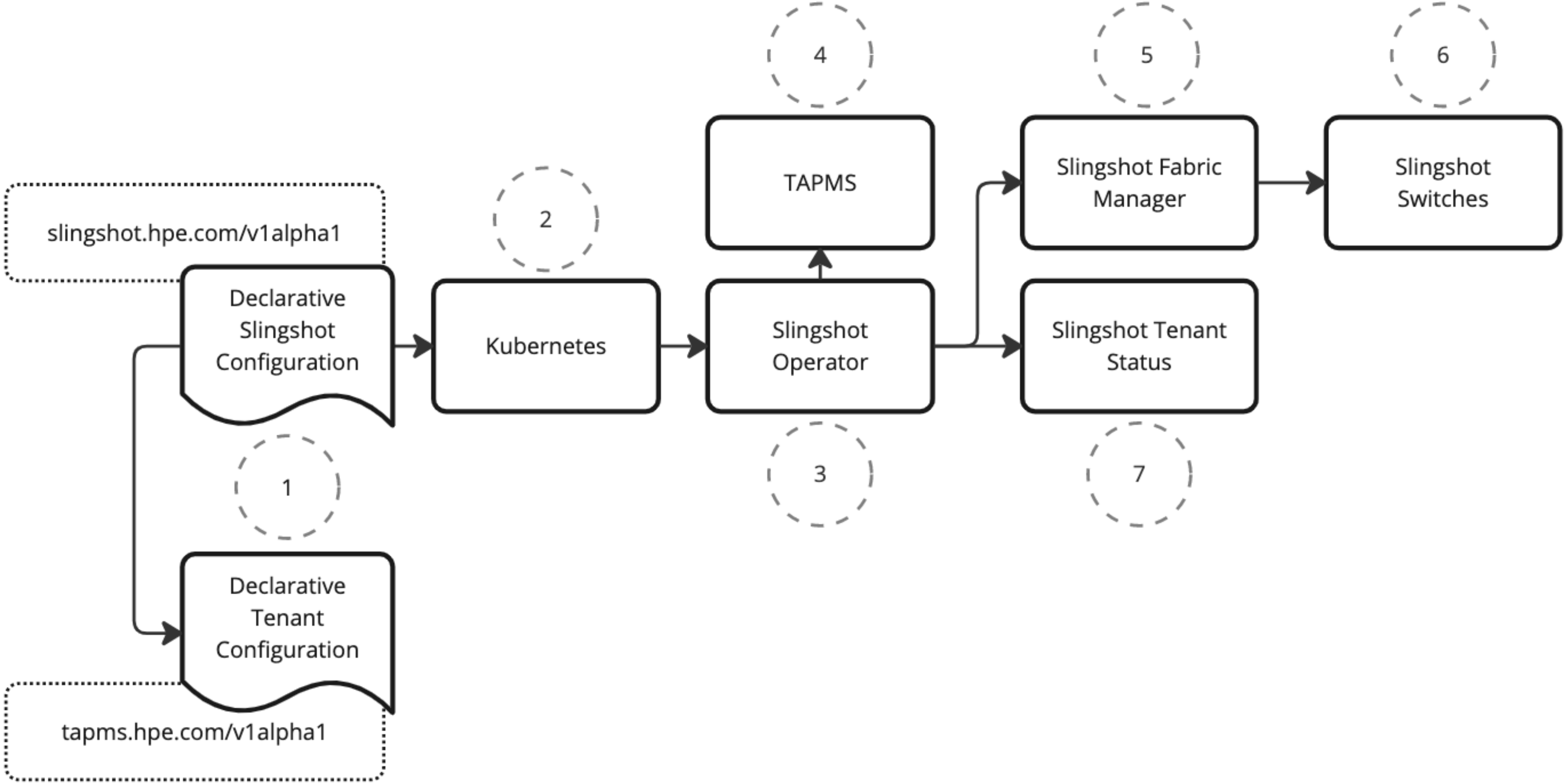
# Cooperative Operators for Tenant Provisioning

## Provisioning Sequencing: Slurm Operator



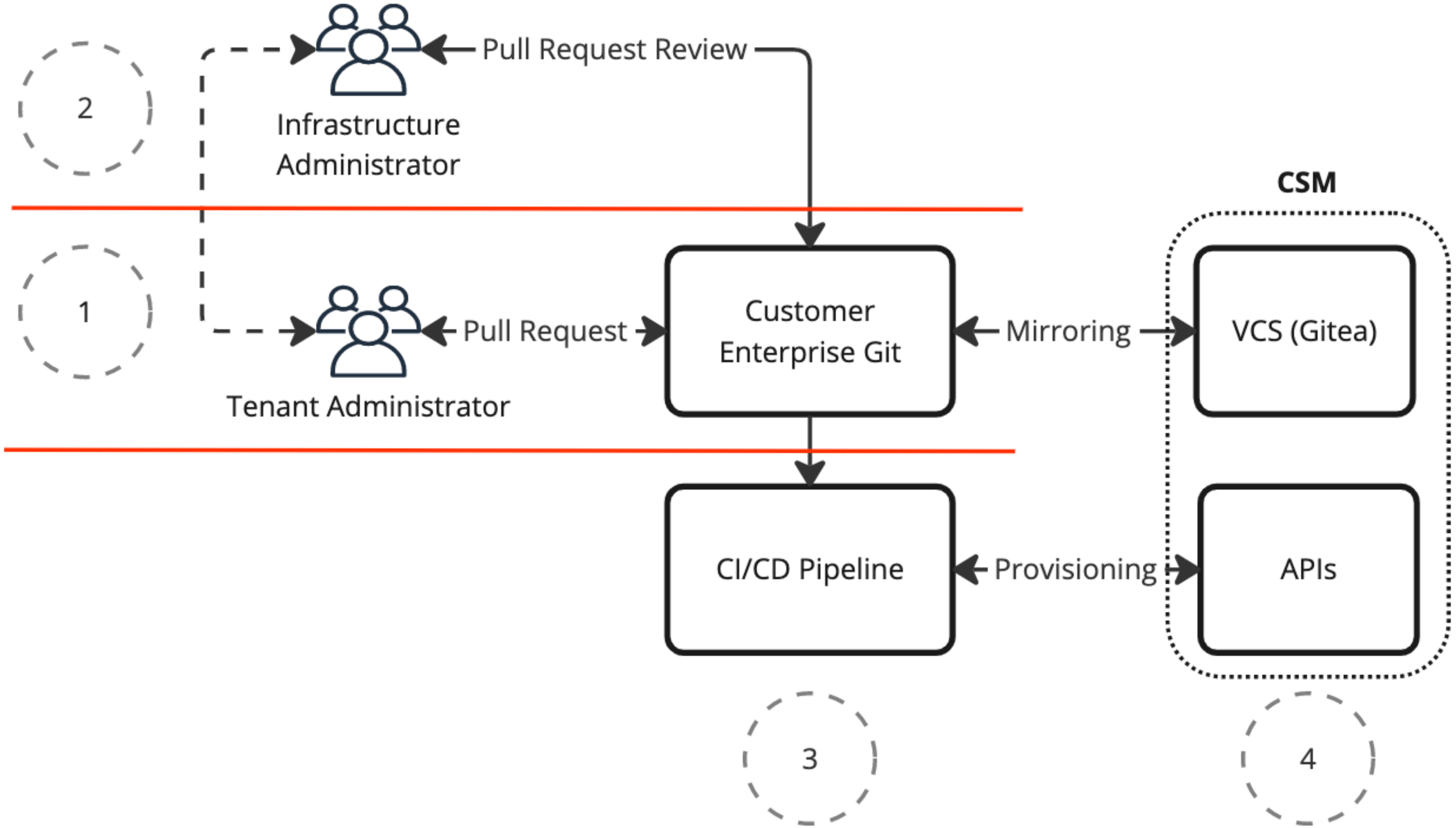
# Cooperative Operators for Tenant Provisioning

## Provisioning Sequencing: Slingshot Operator (FUTURE)



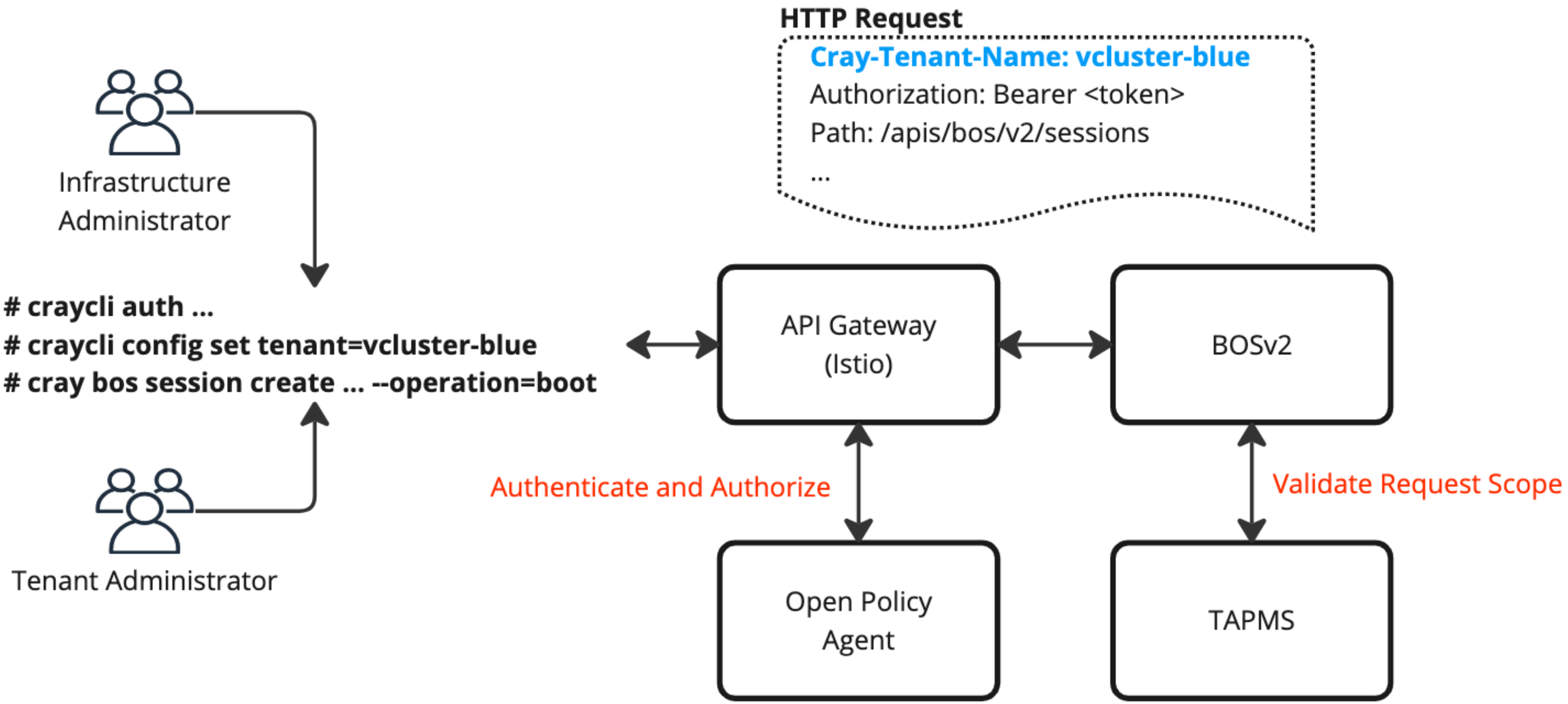
# Transitioning to Multi-tenant Aware, Shared Management API Services

## Change Management: Following a time-tested DevOps Pattern (FUTURE)



# Transitioning to Multi-tenant Aware, Shared Management API Services

Establish a blueprint for incremental refactoring by functional grouping (FUTURE)



**HTTP Request**

```
Cray-Tenant-Name: vcluster-blue
Authorization: Bearer <token>
Path: /apis/bos/v2/sessions
...
```

Infrastructure Administrator

```
# craycli auth ...
# craycli config set tenant=vcluster-blue
# cray bos session create ... --operation=boot
```

Tenant Administrator

API Gateway (Istio)

BOSv2

Authenticate and Authorize

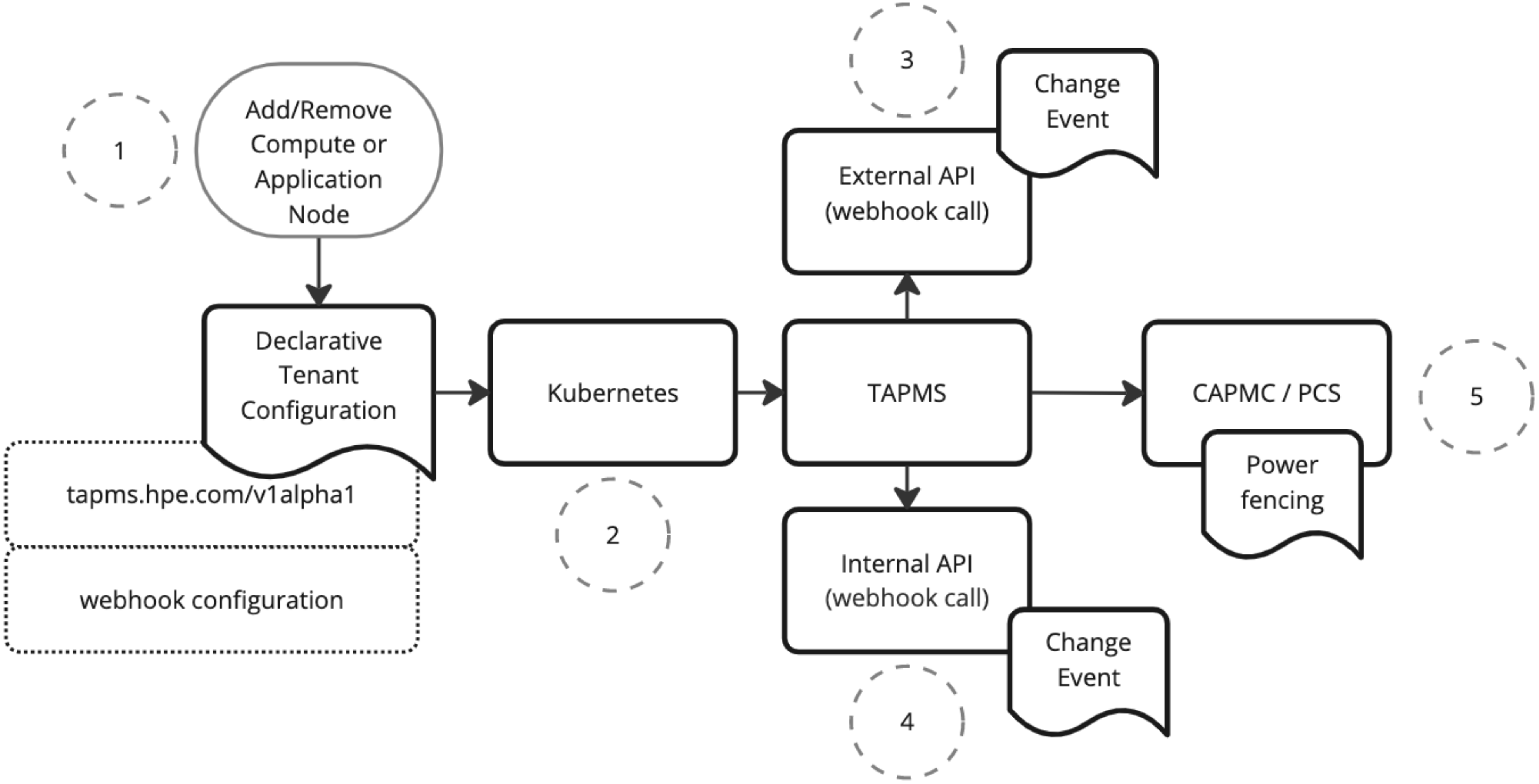
Validate Request Scope

Open Policy Agent

TAPMS

# Transitioning to Multi-tenant Aware, Shared Management API Services

TAPMS webhook, application node resource groups, power fencing (FUTURE)


1
Add/Remove Compute or Application Node
Declarative Tenant Configuration
tapms.hpe.com/v1alpha1
webhook configuration
Kubernetes
2
TAPMS
3
External API (webhook call)
Change Event
Internal API (webhook call)
Change Event
4
CAPMC / PCS
Power fencing
5

# Conclusion and Future Work

Exploring PaaS vs. IaaS, predicate-based tenant scheduling, and HPC storage multi-tenancy

# Conclusion and Future Work

## Conclusion

- Multi-tenancy, the aaS business model, and heterogenous HPC workflows are catalyzing the state of the art for programmable infrastructure and cybersecurity in HPC.
- HPE Cray, in partnership with CSCS and the HPC community, is engaged and actively influencing these trends towards improved outcomes in scientific computing.
- We are excited about the future of the technology and applications, and opportunities for collaborative development with the HPC community.

# Conclusion and Future Work

## Future Work

- Our immediate focus is on helping CSM users, like CSCS, to operationalize the phase one multi-tenancy capabilities, and likewise for phase two to meet production goals.
- Next, as the demarcation point between PaaS and IaaS may benefit from added clarity, we are exploring use case alignment, alongside technologies, designs, and trade-offs that could bring true IaaS multi-tenancy to large scale HPC
- Finally, while multi-tenancy represents a very broad and diverse set of architectural concerns, we would like to explore predicate-based scheduling in TAPMS (e.g., implicit tenant resource selection by hardware properties versus explicit geolocation identifiers), and the state of HPC storage multi-tenancy from a systems architecture perspective.

# Questions?