OpenCHAMI for AI Systems: Reflections on the first year

OpenCHAMI

No Bundled Software. No Problem

DGX

- Bundled and Integrated
- Higher Cost

HGX

- Self Integrated
- Lower Cost

OpenCHAMI

OpenCHAMI

OpenCHAMI is a consortium-driven set of composable software to enable sysadmins to securely and quickly provision and manage HPC/AI systems at any scale.

HPSF EMERGING PROJECT

Los Alamos NATIONAL LABORATORY

NERSC

Hewlett Packard Enterprise

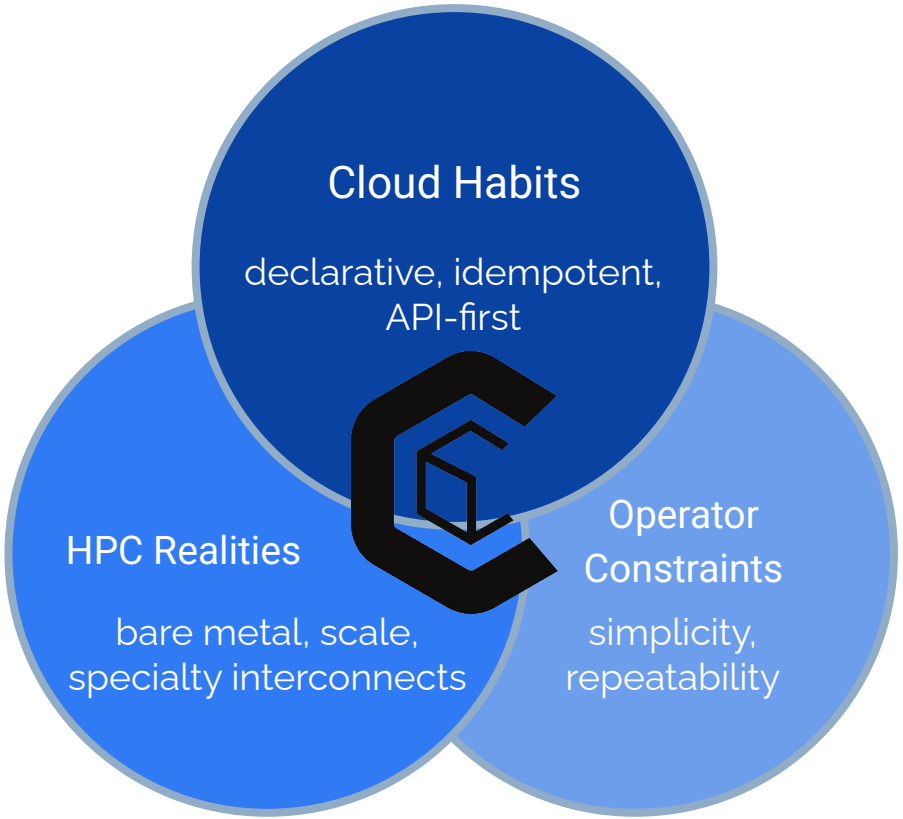
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

DELL Technologies

University of BRISTOL

BriCS

HPC Operations is becoming Hybrid Cloud Ops

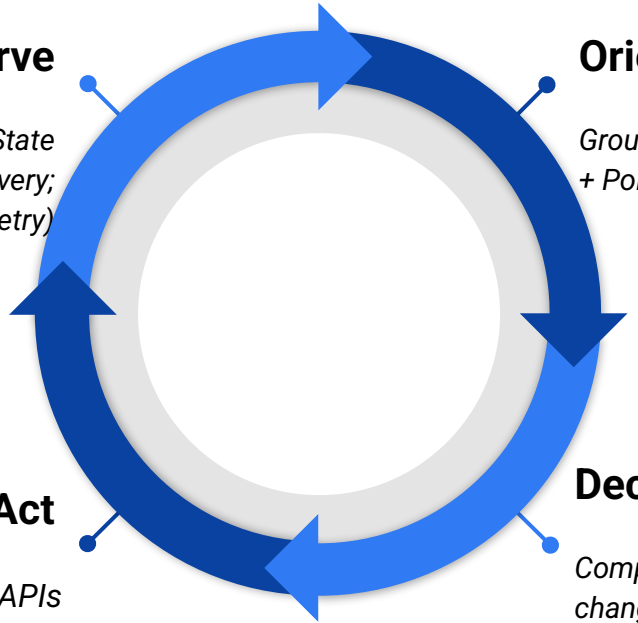


- HPC Operators have different needs than cloud engineers
- Fewer single-use clusters
- More adaptive workloads

OpenCHAMI

Hybrid Cloud Ergonomics for Bare Metal

The cloud difference:
you can re-run it safely until reality matches intent.



OpenCHAMI

On Complexity

A complex system that works is invariably found to have evolved from a simple system that works. The inverse proposition also appears to be true: A complex system designed from scratch never works and cannot be made to work.

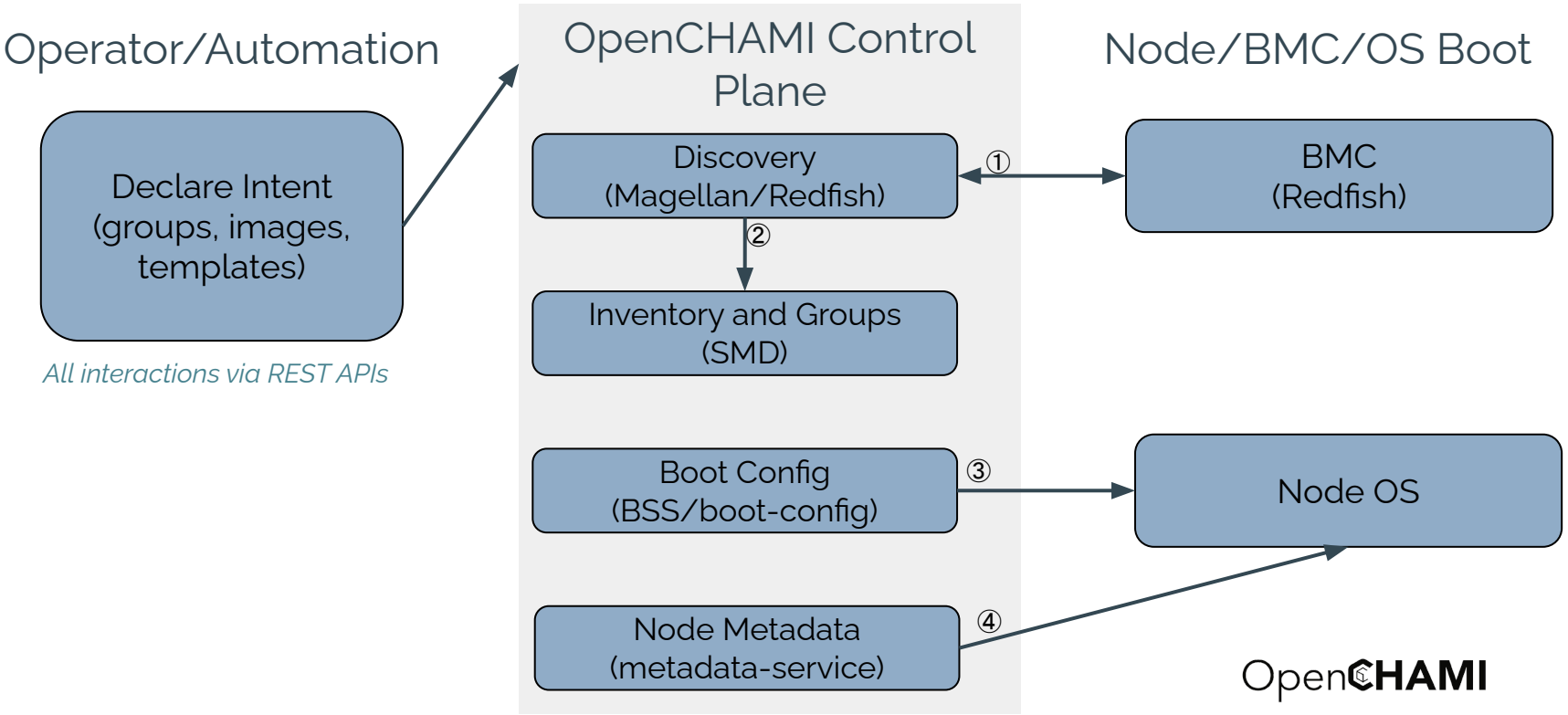
- John Gall, Systemantics (1975)

Cloud flexibility with HPC simplicity

- Modular Services
- Stateless Deployment Options
- Opt-In Complexity
- Containerized Deployment Optional (Kubernetes, Docker, etc…)
- Works as a small, single-purpose cluster; scales up only when you need it

OpenCHAMI

Architecture Overview



Meet Selene and Helios

Selene: 4 Nodes w/8 GPU each

Helios: 10 Nodes 2/8 GPU each

Delivered by DELL

OpenCHAMI

Before the Hardware Arrived

- iDRAC testing
- System Image Development
- Runbooks

OpenCHAMI

Los Alamos Supercomputer Institute

“Make OpenCHAMI useful for undergraduates with no HPC experience”

- Four teams of all skill levels
- Eleven weeks of intense technical mentorship
- Interns of all skill levels and backgrounds
- Three Week build-a-cluster Boot Camp phase
- Nine Week Project phase to use the cluster
- Supervised by LANL sysadmins and researcher mentors

The Clock Starts

- Schedules compress
- Teams aren't available when you need them
- Cabling Issues are common

Two weeks integrating Selene

OpenCHAMI Installation → Troubleshoot Cabling → OpenCHAMI Provisioning

Hardware Checkout

Kickstart Provisioning

OpenCHAMI Installation

```
You are in a maze of twisty little passages, all alike.
e

You are in a maze of twisty little passages, all alike.
s

You are in a maze of twisty little passages, all alike.
s

You are in a maze of twisty little passages, all alike.
s

You are in a maze of twisty little passages, all alike.
n

You are in a maze of twisty little passages, all alike.
```

Run Magellan Discovery

Configure Groups

Test image with GPU drivers

Test Infiniband

Programming Environment Testing

OpenCHAMI

Three days integrating Helios

Hardware Checkout

Kickstart Provisioning

OpenCHAMI Installation

Run Magellan Discovery

Configure Groups

Test image with GPU drivers

Test Infiniband

Programming Environment Testing

The Team

- Two Hardware Techs
- Three Students
- One Sysadmin

Reflections: What Changed?

- The power of pipelines
- The importance of training and documentation
- Adapting tooling to improve user experience

Co-Scheduling

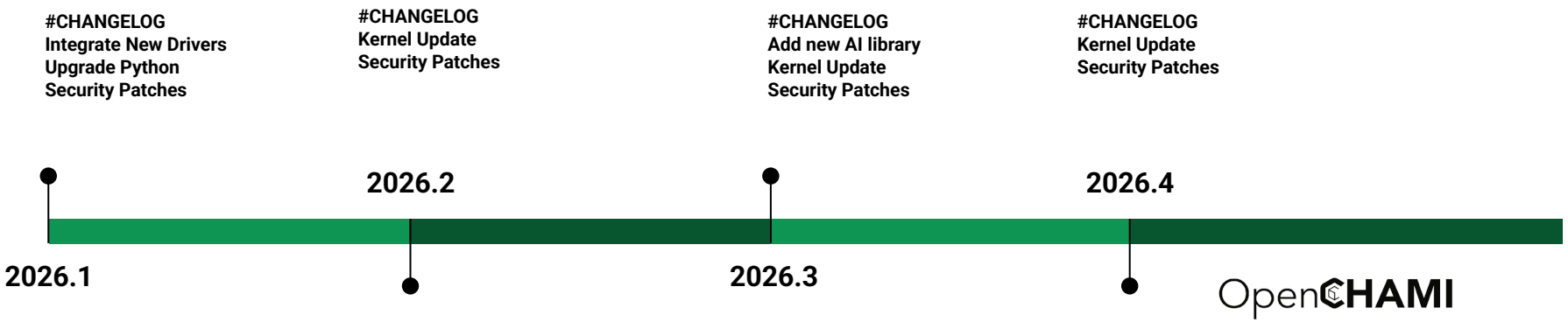
- Switching schedulers can mean switching kernels
- Users will want different things from their cluster at different times
- The pace of change in AI is only increasing

The Release Train

Changes to intent can happen at any time.

They are validated through continuous testing

The most recent fully tested configuration is released on a fixed schedule



OpenCHAMI

Kexec based reboots

- BSS always knows the intended boot parameters for each node
- SystemD has a target called kexec
- We simply connect the two and reboot in seconds

```
# bss-kexec-load.sh – runs before kexec executes
BSS_URL="http://openchami.cluster:27778/boot/v1/bootparams"
JSON=$(curl -sf "$BSS_URL")

KERNEL=$(echo "$JSON" | jq -r '.[0].kernel.path')
INITRD=$(echo "$JSON"  | jq -r '.[0].initrd[0]')
CMDLINE=$(echo "$JSON" | jq -r '.[0].params')

curl -so /var/lib/kexec/kernel "$KERNEL"
curl -so /var/lib/kexec/initrd "$INITRD"

kexec --load /var/lib/kexec/kernel \
      --initrd=/var/lib/kexec/initrd \
      --command-line="$CMDLINE"
```

What's real today?

- **Today:** Stable Core Workflows, Systems in Production, Active Community
  - Deployed on five supported platforms at LANL
  - ~12 active committers across five sites
- **In Transition:** Replacing legacy services while maintaining stable APIs
  - Fabrica based boot-service, metadata-service, fru-tracking, boot-orchestration
- **Next:** Broader scaling options and more extensible control
  - Stateless horizontal scaling -> Stateful cell-based architecture

Open Project: Decision Making and Contributions

How Decisions Happen

- Roadmap Issues are Public
- RFD = Discussion
- ADR = Decision

How to Contribute

- Browse PRs and comment
- Pick an issue to implement
- Join the discussions in RFDs

How to Join the Community

- Weekly Office Hours on Public Calendar
- Join the Slack Channel and Mailing List
- Review the weekly Github updates

HIGH PERFORMANCE SOFTWARE FOUNDATION

THE LINUX FOUNDATION

github.com/OpenCHAMI

openchami.org

OpenCHAMI

OpenCHAMI

Join us today!

openchami.org/docs/getting-started

Appendix

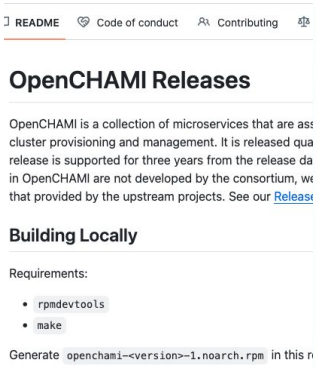
OpenCHAMI

Why Microservices?

- Swap implementations behind stable APIs
- Deploy only what you need
- Scale pieces independently
- Auditable Security Boundaries

Repository Map

Release	SMD	BSS	metadata-service	Roadmap
RPM for installing the way we do at LANL	Main Inventory Microservice	Boot Script Microservice	Cloudinit metadata microservice	Discussions about priorities


README Code of conduct Contributing

OpenCHAMI Releases

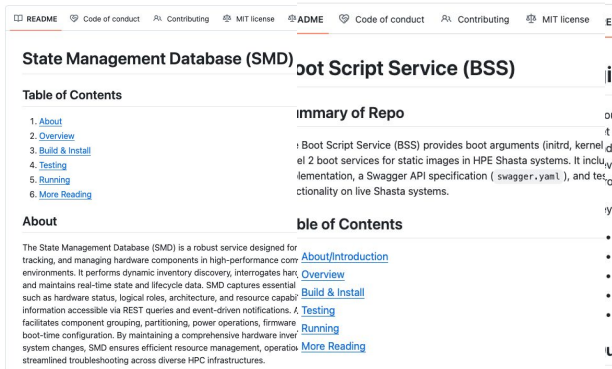
OpenCHAMI is a collection of microservices that are ass
cluster provisioning and management. It is released qua
release is supported for three years from the release da
in OpenCHAMI are not developed by the consortium, we
that provided by the upstream projects. See our Release

Building Locally

Requirements:

- `rpmdevtools`
- `make`

Generate `openchami-<version>-1.noarch.rpm` in this r


README Code of conduct Contributing MIT license

State Management Database (SMD)

Table of Contents

1. About
2. Overview
3. Build & Install
4. Testing
5. Running
6. More Reading

About

The State Management Database (SMD) is a robust service designed for
tracking, and managing hardware components in high-performance com
environments. It performs dynamic inventory discovery, interrogates har
and maintains real-time state and lifecycle data. SMD captures essential
such as hardware status, logical roles, architecture, and resource capabi
information accessible via REST queries and event-driven notifications.
facilitates component grouping, partitioning, power operations, firmware
boot-time configuration. By maintaining a comprehensive hardware inver
system changes, SMD ensures efficient resource management, operatio
streamlined troubleshooting across diverse HPC infrastructures.

ADME Code of conduct Contributing MIT license

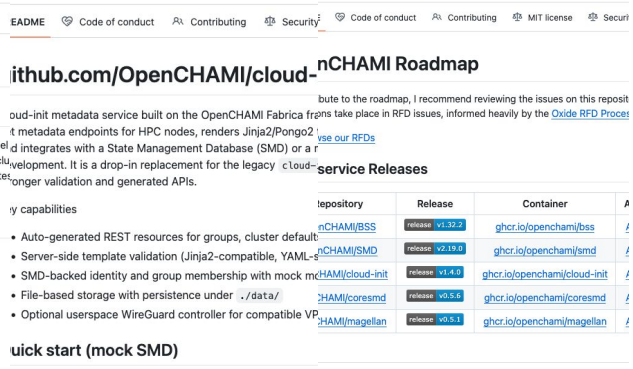
oot Script Service (BSS)

mmary of Repo

Boot Script Service (BSS) provides boot arguments (initrd, kernel
el 2 boot services for static images in HPE Shasta systems. It inclu
lementation, a Swagger API specification (`swagger.yaml`), and tes
ctionality on live Shasta systems.

ble of Contents

About/Introduction
Overview
Build & Install
Testing
Running
More Reading


EADME Code of conduct Contributing Security

ithub.com/OpenCHAMI/cloud-

oud-init metadata service built on the OpenCHAMI Fabrica fr
t metadata endpoints for HPC nodes, renders Jinja2/Pongo2
d integrates with a State Management Database (SMD) or a r
velopment. It is a drop-in replacement for the legacy `cloud-`
onger validation and generated APIs.

y capabilities

- Auto-generated REST resources for groups, cluster default
- Server-side template validation (Jinja2-compatible, YAML-s
- SMD-backed identity and group membership with mock m
- File-based storage with persistence under `./data/`
- Optional userspace WireGuard controller for compatible VP

uick start (mock SMD)

Code of conduct Contributing MIT license Securi

nCHAMI Roadmap

bute to the roadmap, I recommend reviewing the issues on this reposit
ns take place in RFD issues, informed heavily by the Oxide RFD Proces

vse our RFDs

service Releases

epository	Release	Container
nCHAMI/BSS	release v1.32.2	ghcr.io/openchami/bss
nCHAMI/SMD	release v2.19.0	ghcr.io/openchami/smd
HAMI/cloud-init	release v1.4.0	ghcr.io/openchami/cloud-init
HAMI/coresmd	release v0.5.6	ghcr.io/openchami/coresmd
HAMI/magellan	release v0.5.1	ghcr.io/openchami/magellan

OpenCHAMI

Composable. Secure. Practical.

OpenCHAMI

Modernization: Generated Services

Scaling OpenCHAMI