

Co-developing OpenCHAMI Services with Fabrica

Problem Statement

As the HPC community adopts OpenCHAMI, the risk of API fragmentation increases. With multiple stakeholders (vendors, sites, and open-source contributors) building independent microservices, reliance on PDF specifications creates "interpretation drift" (the subtle inconsistencies that arise when different developers implement the same written standard). Additionally, the operational overhead of manually implementing boilerplate code, such as Kafka buses, CloudEvents, and Authentication/Authorization, slows down the delivery of features and introduces inconsistency.

Framework-Driven Governance

We propose a shift from "compliance by inspection" to "compliance by generation." We will introduce Fabrica, a toolchain used by the OpenCHAMI project that treats the API specification as the source of truth.

We will cover:

- How Fabrica enforces the OpenCHAMI API Working Group standards automatically, ensuring that decisions, such as using Kubernetes-style envelopes versus flat models, are applied consistently across services.
- How removing infrastructure boilerplate allowed the development team to focus purely on the application logic of FRU tracking, significantly increasing velocity.
- How updates to the standard can be propagated to existing services via code regeneration, future-proofing the stack against architectural pivots without requiring manual rewrites.

Case Study: The FRU Inventory Service (HPE & LANL)

We will present a retrospective on the co-development of the FRU Inventory Service to demonstrate this model in action.

We will cover:

- How LANL provided operational requirements while HPE provided the implementation framework, allowing for rapid iteration.

- The resulting production-ready service, which was delivered with significantly reduced lead time compared to legacy development cycles.
- A brief walkthrough of the service architecture and the code generation workflow to demonstrate the reduction in developer friction.

Key Takeaways for Attendees

- Understand how OpenCHAMI guarantees interoperability between different tools through adherence to generated contracts.
- Learn a strategy for speeding up service development using schema-driven tools that minimize or eliminate the need to write repetitive and error-prone boilerplate code.
- See a proven model for effective vendor-site open-source collaboration that bridges the gap between product roadmaps and site-specific needs.