

archi current deployments

Juan Pablo Salas (UW Madison, USCMS) on behalf of archi team

Hardware-Aware AI: Next-Generation Data Quality Monitoring
September 30, 2026



Introduction

Archi is an open-source framework that organizes heterogeneous information sources and deploys agents that retrieve and reason over them.

We want Archi to serve multiple experiments and subgroups, so it has to be *configurable*, *portable*, and easy to install.

This talk compares two production deployments to show what that looks like in practice.

This talk

01

CompOps deployment

03

Side by side comparison

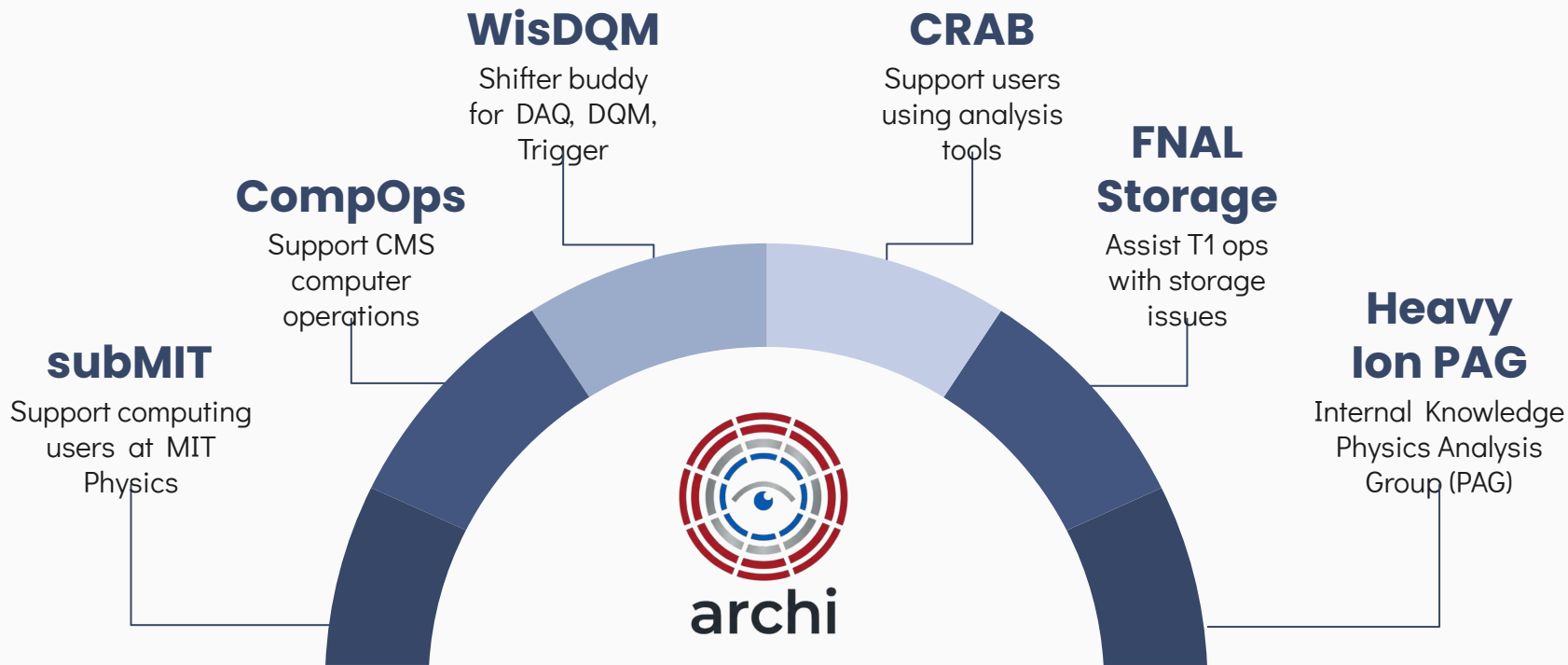
02

WisDQM deployment

04

Next steps

Current HEP related archi deployments



CMS CompOps

Knowledge sources

- Public documentation: FTS3, Rucio, PdmV
- Private documentation: CMS Monit, CMS Web
- Git based documentation: WMCore, Tier0, DM Ops, Pnr
- Jira tickets (hourly)

Dynamic sources

- OpenSearch API tool
- DeepWiki MCP
- Rucio MCP
- DBS MCP

CMS CompOps

Deployment details

- VM on CERN OpenStack (flavor m2.2xlarge: 16 vCPU, 30 GB RAM, 160 GB disk, RHEL 8)
- Three containers: chatbot, data manager, PostgreSQL
- Models served through CERN's LiteLLM gateway: GPT-5, Mistral Large (latest), gpt-oss-120b, Qwen3-32B

CMS CompOps

How are upgrades handled?

- Managed manually by a single admin
- Back up the PostgreSQL database
- Merge upstream changes and apply local configuration
- Rebuild and restart the containers with `archi restart`

WisDQM

The DQM shift assistant supports CMS shifters at the LHC Physics Center at Fermilab.

Knowledge sources

- Twiki documentation: DQM, Offline Tracker, Trigger
- Git based documentation: PPD, L1T, TKDQM

Dynamic sources

- OMS MCP (in development)
- Run Coordination Report tool

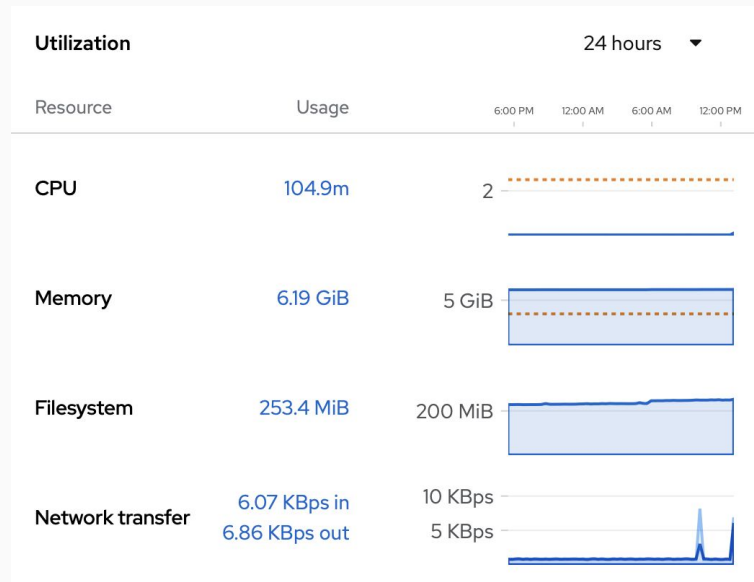
WisDQM

Deployment

- OKD (community OpenShift) cluster at Fermilab
- Same three components: chatbot, data manager, PostgreSQL + Grafana
- `archi install` generates a complete Helm chart from the deployment config: Chart.yaml, values.yaml, Kubernetes manifest templates, Secrets, and PersistentVolumeClaims
- Models fetched via Fermilab's Ollama (qwen3.6:35b, qwen2.5:7b) previously vLLM (gpt-oss-120b)

Example: WisDQM test instance

Name ↑	Status ↑	Ready ↑	Restarts ↑	Owner ↑	Memory ↑	CPU ↑	Created ↑
 wisdqm-test-chatbot-d6799b6f-pp89q	 Running	1/1	0	 wisdqm-test-chatbot-d6799b6f	509.6 MiB	0.001 cores	 Aug 19, 2026 at 5:53 PM
 wisdqm-test-data-manager-6c5544444-7-8wp7s	 Running	2/2	0	 wisdqm-test-data-manager-6c5544444	1,038.4 MiB	0.004 cores	 Aug 19, 2026 at 5:53 PM
 wisdqm-test-postgres-55c8f88b68-4grb5	 Running	1/1	0	 wisdqm-test-postgres-55c8f88b68	112.4 MiB	0.001 cores	 Aug 17, 2026 at 12:23 PM



Currently running at <https://wisdqm-test.fnal.gov> (under CERN SSO) for test instances.

WisDQM

How are upgrades handled?

- Upstream CI/CD builds and publishes the base container images
- Pushing config changes to GitLab triggers a pipeline that regenerates the Helm chart
- Local patches layered on top of upstream container images for fast iteration
- Chart is currently applied to OKD manually with helm upgrade (full automation pending)

Comparison

	VM	Kubernetes
<i>Upgrade process</i>	Manual, by one admin	GitLab pipeline (collaborative) + manual helm upgrade
<i>Speed of changes</i>	Fast, direct edits	Slower, goes through CI + patches
<i>Traceability</i>	Limited	Every change tracked in Git
<i>Rollback</i>	Restore from DB backup	Helm rollback to a previous release
<i>Scaling</i>	Fixed VM size	Stateless services can scale out
<i>Network access</i>	Restricted to CERN network	Managed through OKD Routes/Ingress

Which deployment should you choose?

- **VM:** quick to set up, good for prototyping or a small team with one maintainer
- **Kubernetes:** recommended for production, shared maintenance, and anything that needs traceable, repeatable upgrades

Future development ideas

Continuous
deployment

adopt Argo CD (GitOps) to deploy automatically, with separate dev and prod instances

Operational
Knowledge Graph
(OKG)

as the knowledge base structure changes, the Helm charts will need to evolve with it

Central MCP
hub

host our self-hosted MCP (Model Context Protocol) servers in one place, with access controlled through an SSO gateway

Thanks!



Do you have any questions?

Team

Christoph Paus, Tim Kraska, Pietro Lugato, Luca Lavezzo, Jason Mohoney, Hasan Ozturk, Dmytro Kovalskyi, Hassan Ahmed, Viphaba Ohm, Juan Pablo Salas, Gabriele Benelli, Aron Soha, Samantha Sunnarborg, Krittin Phornsiricharoenphant, Antonio Battaglia, Austin Swinney, Manasvita Joshi, Warren Nam, and many more!

Repo: <https://github.com/archi-physics/archi>

Docs: <https://archi-physics.github.io/archi/>

Contact: a2rchi@mit.edu