Karlsruhe Institute of Technology

Institute for Beam Physics and Technology

Self-Hosting Research Data Infrastructure with Kadi4Mat: A Practical Use Case for Managing Physics Data at IBPT, KIT

Saeid Masoumi, J. Gethmann, W. Mexner, M. Schuh, R. Ruprecht, A.-S. Müller

Motivation

- ✓ Parameter space at KARA & FLUTE is too large → scans are impractical
- ✓ A metadata database (RDM system) is essential
- ✓ Enables combination of datasets, reuse of past experiments, and better planning
- ✓ Ensures efficient use of costly beam time
- ✓ Accelerates transition from setup to actual measurements
- ✓ Goal: Self-hosted, scalable, FAIR-compliant research data infrastructure for physics data

How Kadi tested

- ➢ **Test Environments**: Virtual Machine & Dedicated Server
- ➢ **Storage**: Local Storage, S3 (Object Storage), LSDF
- ➢ **Deployment Approaches**: Monolithic Application, Dockerized Setup
- ➢ **Upload Methods**: WebUI, Kadi-apy (CLI)
- ➢ **Mounting File Systems**: CIFS, davfs, sshfs
- ➢ **Data Chunk Sizes**: 10 MB, 64 MB, 128 MB, 256 MB, 512 MB, 1 GB

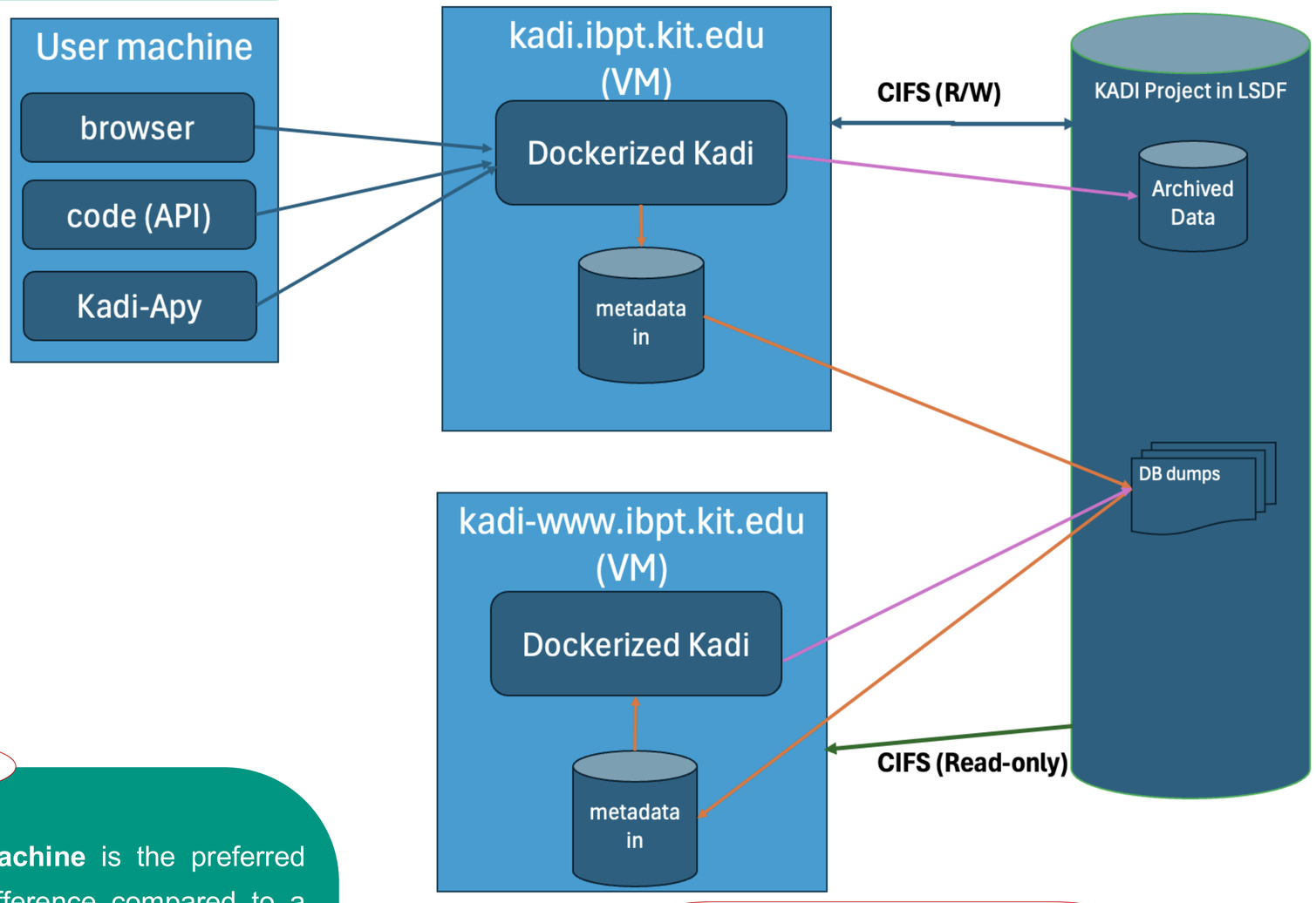
Goal: speed up big data uploads & reduce number of requests

Why Kadi4mat

- ✓ Self-hosted & Open Source
  - ❖ deployable on institutional infrastructure with Docker
- ✓ Flexible Metadata
  - ❖ supports multiple schemas, templates, and semantic dataset linking
- ✓ Data Lifecycle Tracking
  - ❖ from data creation to publication, with provenance & versioning
- ✓ Integration & Workflows
  - ❖ Nice supporting tools (Kadi-APY, KadiFS, Kadistudio)
- ✓ Scalable & FAIR
  - ❖ suited for large-scale datasets and compliant with FAIR principles
- ✓ Community-driven
  - ❖ open development, extensible, and widely adopted

Our implementation

- Two Kadi Instances
  - ✓ Internal: Archiving research data
  - ✓ External: FAIR data publication (open access)
- Access Control
  - ✓ Internal instance: only inside institute
  - ✓ External instance: accessible from internet
- Authentication: KIT OpenID Federation
- Data Handling
  - ✓ Internal Kadi: read/write
  - ✓ External Kadi: read-only
- Database & Backup
  - ✓ Shared database with daily backup
  - ✓ Internal DB dump → restored on external instance



Lessens learned

In our evaluations, we observed that using a **virtual machine** is the preferred choice due to easier management, and performance (difference compared to a dedicated server was negligible). For storage, the **LSDF** infrastructure proved most suitable, as it is specifically designed for large-scale scientific data.

To increase security and maintain modularity, we opted for a **Dockerized setup** instead of a monolithic application. For handling large datasets, the **Kadi-apy** CLI tool turned out to be more efficient than the WebUI, offering significantly faster uploads. Regarding storage protocols, **CIFS** was chosen as the most stable option for mounting LSDF within the Kadi VM.

Furthermore, by testing different data chunk sizes, we identified **1 GB** as the optimal size for balancing upload speed and system stability. To ensure the fairness of our experiments, memory caches were cleared before each upload.

Feedback & future work

Our implementation of Kadi has proven to be stable, scalable, and user-friendly, with positive test results confirming its readiness for broader deployment. Building on this foundation, the next steps include the expansion of the system to other accelerator facilities, introducing data immutability within the Kadi framework, and exploring the integration of large language models (LLMs) to enrich metadata management.

Contact: **saeid.masoumi@kit.edu**

EURO-LABS
EUROPEAN LABORATORIES FOR ACCELERATOR BASED SCIENCES

This project has received funding from the European Union's Horizon Europe Research and Innovation programme under Grant Agreement No 101057511.