Karlsruhe Institute of Technology

Institute for Beam Physics and Technology

Research Data Management at KIT's Accelerator Facilities

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)
- ✓ **File Systems**: CIFS, davfs, sshfs
- ✓ **Data Chunk Sizes**: 10 MB, 64 MB, 128 MB, 256 MB, 512 MB, 1 GB

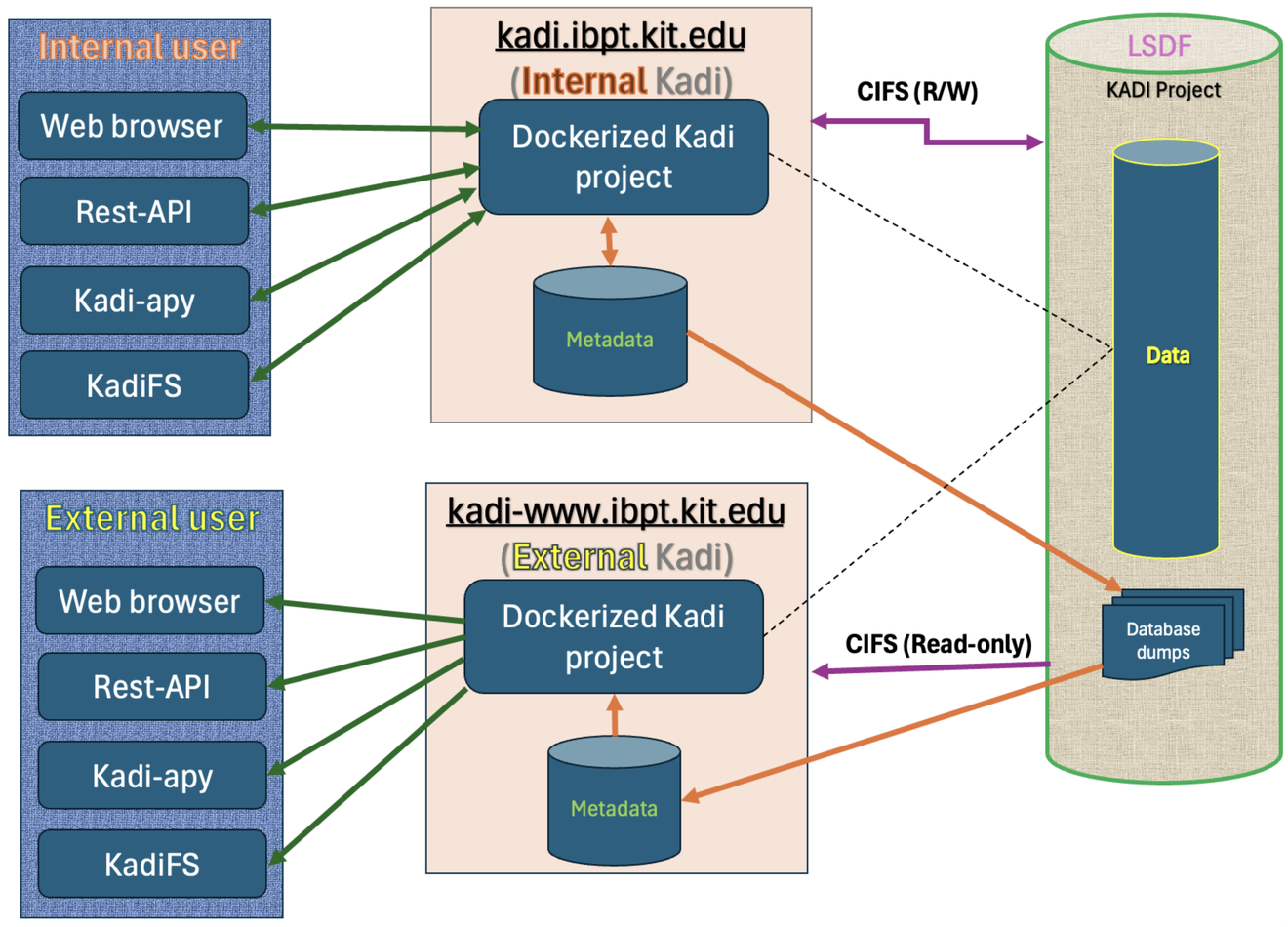
Goal: speed up big data uploads and reduce number of requests

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) & seamless integration with GitLab CI/CD pipelines
- Scalable & FAIR
  - ❖ suited for large-scale datasets and compliant with FAIR principles
- Community-driven
  - ❖ open development, extensible, and widely adopted

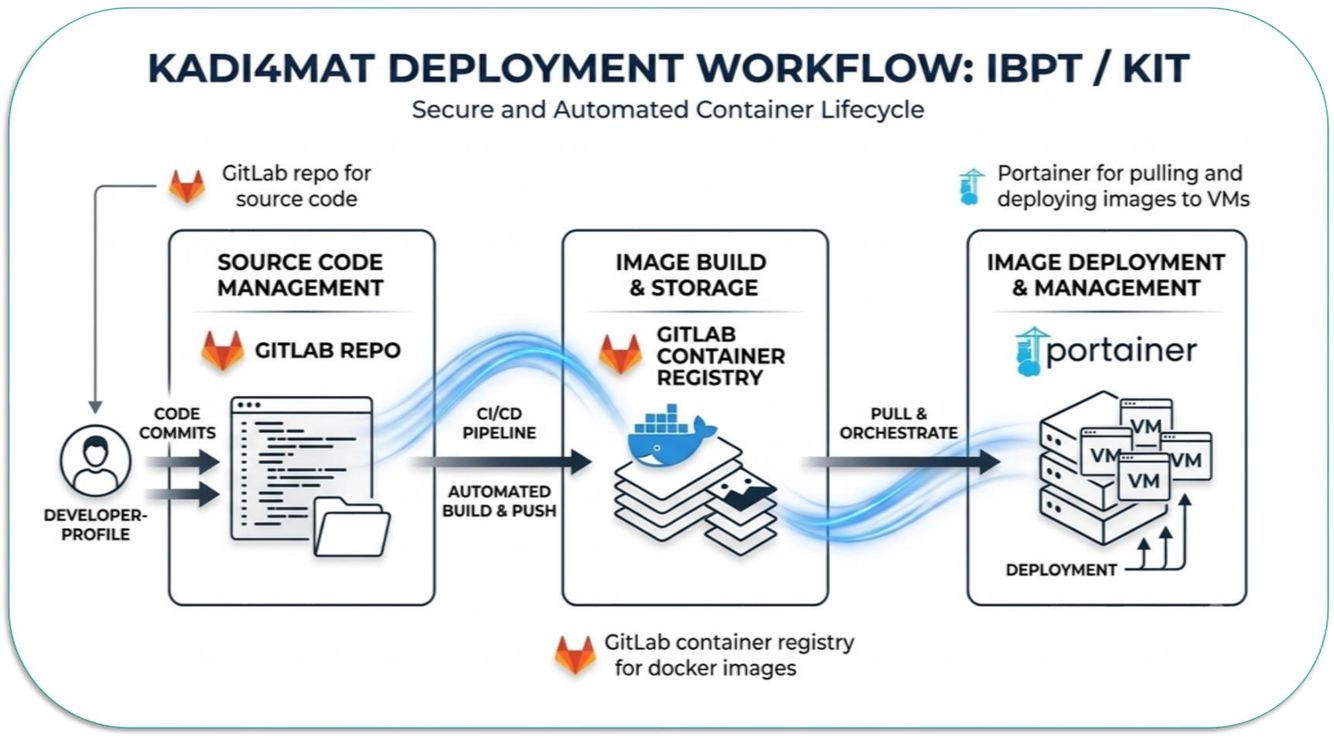
Our implementation

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



Conclusion

- ✓ **System Readiness**: Our implementation of Kadi proves to be stable, scalable, and user-friendly, with positive test results confirming its readiness for broader deployment.
- ✓ **Upload Efficiency**: For handling large datasets, the **Kadi-apy** CLI tool is significantly more efficient than the WebUI, offering faster upload speeds.
- ✓ **Storage Optimization**: CIFS was selected as the most stable protocol for mounting the LSDF within the Kadi VM.
- ✓ **Architecture & Security**: To enhance security and maintain modularity, we opted for a **Dockerized setup** over a monolithic application.
- ✓ **Optimal Chunk Size**: Through rigorous testing of various data chunk sizes, **1 GB** was identified as the optimal size for balancing upload speed and system stability.



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

KIT – The University in the Helmholtz Association

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

www.kit.edu