PROCEEDINGS OF SCIENCE

The Helmholtz Model Zoo: Data Sovereignty Through On-Premise AI Inference Infrastructure

Hans Werners,[a,*] **Engin Eren,**[a] **Patrick Fuhrmann**[a] **and Philipp Heuser**[a]

[a] *Helmholtz Imaging, Deutsches Elektronensynchrotron – DESY, Notkestraße 85, 22607 Hamburg, Germany*

E-mail: hans.werners@desy.de, engin.eren@desy.de, patrick.fuhrmann@desy.de, philipp.heuser@desy.de

For research institutions that handle sensitive or regulated data, the location of AI inference is not architecturally neutral. Running models on commercial cloud platforms exposes research data to external processors and conflicts directly with data sovereignty requirements, national regulations, and the data governance policies that publicly funded research organizations are bound by. The Helmholtz Model Zoo (HMZ) is a fully on-premise AI inference platform built on institutional infrastructure within the Helmholtz Association. The platform runs on dedicated GPU nodes at DESY's Maxwell HPC cluster, stores all data in dCache under the user's institutional account, and authenticates users via Helmholtz ID. Domain scientists submit data to pre-deployed models through a web interface without programming expertise, while model providers integrate their trained models through standardized implementation templates. Virtual organizations provide fine-grained access control that extends to external collaborators without data leaving institutional infrastructure. We describe the platform's design, its integration with Helmholtz services, and operational considerations for sovereign on-premise AI inference infrastructure.

International Symposium on Grids and Clouds (ISGC2026)
15-20, March, 2026
Academia Sinica, Taipei, Taiwan

*Speaker

© Copyright owned by the author(s) under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (CC BY-NC-ND 4.0) All rights for text and data mining, AI training, and similar technologies for commercial purposes, are reserved. ISSN 1824-8039 . Published by SISSA Medialab.

PoS(ISGC2026)027

1. Introduction

For publicly funded research organizations, running AI inference on external platforms means relying on infrastructure the institution neither operates nor controls. Data sovereignty, the principle that data must remain on infrastructure the institution itself operates, is the central requirement in this context. A model that is accessible via a well-documented interface but can only be executed on a commercial cloud platform may not satisfy the requirements of researchers who are not permitted to transfer their data to external services. Beyond data transfer restrictions, the continued availability of such a model remains outside institutional control.

The Helmholtz Association, Germany's largest research organization with 18 centers and approximately 47,500 employees, faces these challenges across six research fields. The association maintains shared digital infrastructure through the Helmholtz Federated IT Services (HIFIS) [1], the Helmholtz Cloud [2], and the Helmholtz Authentication and Authorisation Infrastructure (Helmholtz ID) [3]. This infrastructure provides the building blocks for AI inference that operates entirely within the association's own infrastructure, hosted at a single Helmholtz center but governed and accessible across the federation.

In this work, we present the Helmholtz Model Zoo (HMZ) as a reference implementation of sovereign on-premise AI inference infrastructure. We describe the platform's architecture, its authentication and access control mechanisms, and the user-facing design that allows domain scientists without deep computational expertise to use leading trained deep learning models on institutional GPU hardware. We also discuss operational considerations and the transferability of the approach to other research organizations.

2. Platform Overview

The Helmholtz Model Zoo (HMZ) is an AI inference platform that enables researchers across the Helmholtz Association to share and run trained deep learning models through a web frontend and REST APIs. The platform is designed to lower the barrier for two groups with different needs and backgrounds. Users, typically domain scientists, access pre-deployed models through the web frontend and submit data for inference without programming expertise. All compute runs on dedicated GPU nodes at DESY's Maxwell HPC cluster. Model providers, the AI specialists and researchers who develop and train models, integrate their pre-trained models into the platform through a standardized submission workflow in which automated pipelines validate and deploy them. Throughout this paper, we use "model" for the trained deep learning network and its inference logic, "model weights" for the parameter files that encode it, and "model container" for the OCI image that bundles the inference code, runtime, and its specific orchestration.

Model providers select one of three implementation templates, all of which execute GPU workloads on NVIDIA Triton Inference Server [4]. The Python Backend template provides a native Python environment with minimal implementation effort and no throughput optimizations. The ASAP (Asynchronous Serving And Processing) template structures execution into three asynchronous stages (preprocessing, inference, postprocessing) and improves throughput at the cost of stricter model decomposition. The FLOW (Flow-based Logical Orchestration Wrapper) template imposes no fixed execution structure and is suited to models that require data chunking or com-

PoS(ISGC2026)027

plex orchestration, but achieves competitive throughput only when the provider invests in custom orchestration.

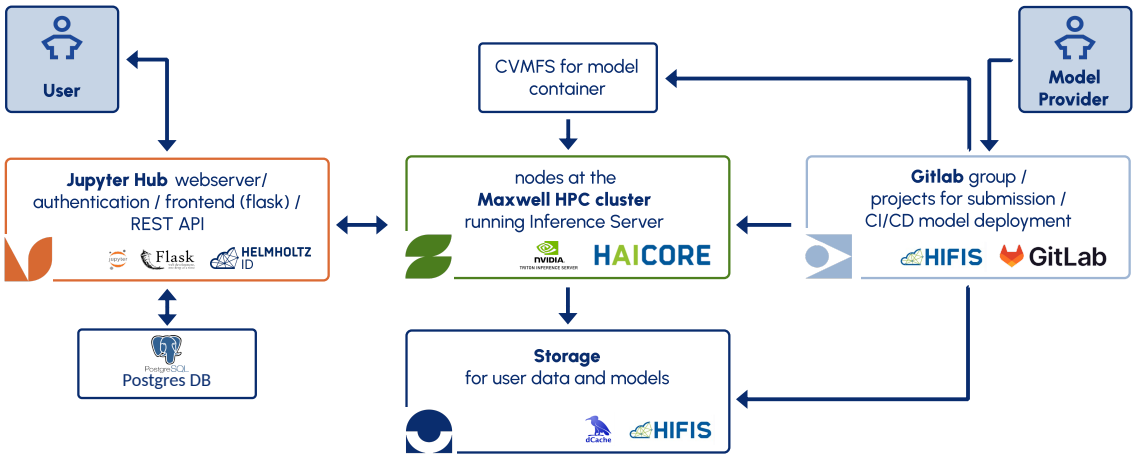

Figure 1: Architecture of the Helmholtz Model Zoo. All components operate within institutional infrastructure. Users authenticate via Helmholtz ID. Data is stored in dCache under the user's own account and compute runs on DESY's Maxwell HPC cluster.

Figure 1 shows the platform's two-sided design. On the user side, domain scientists reach the platform through a JupyterHub-hosted Flask web frontend or its REST API, both served from front-end servers that are the only hosts able to reach the GPU nodes. Both entry points submit inference jobs to Slurm, which schedules them on the dedicated GPU nodes, where Triton Inference Server orchestrates execution and reads model weights from dCache and container images from CVMFS. On the provider side, model authors maintain a GitLab repository that holds the inference code, the implementation template configuration, and references to the trained model weights. CI/CD pipelines push the model weights to dCache, build the container image and publish it to CVMFS, and test the model end-to-end on a GPU runner whose Triton server matches the production configuration. Every component runs on infrastructure operated by DESY or the Helmholtz Association, a deliberate response to the data sovereignty requirements of publicly funded research.

Each model is described by a `meta.json` record covering name, description, scientific domain, input and output formats, licensing, and version. These records populate the per-model pages on the web frontend, where all available models are listed.

The platform accepts models from multiple deep learning frameworks, including PyTorch, TensorFlow, and ONNX [5], through Triton's multi-backend architecture, which avoids framework lock-in for model providers. Pre-trained model artifacts are hosted on DESY's institutional GitLab instance. CI/CD pipelines validate that each model integrates correctly with the platform's execution environment before deployment. Model weights are stored persistently in dCache, and versioned updates are supported through the same submission workflow.

PoS(ISGC2026)027

3. On-Premise Infrastructure

The Helmholtz Cloud [2] is the federated cloud platform of the Helmholtz Association, developed and maintained by HIFIS [1] and delivered through services hosted at participating Helmholtz centers. The HMZ is deployed within this environment, on infrastructure operated by DESY rather than on rented hardware or shared data centers outside institutional control.

The compute infrastructure consists of seven dedicated nodes in DESY's Maxwell HPC cluster, comprising six nodes with four NVIDIA L40S GPUs each and one node with two NVIDIA H100 GPUs. These nodes are funded by the HAICORE initiative [6] and reserved for HMZ workloads. On each node, NVIDIA Triton Inference Server runs as a persistent service, managed through Puppet [7] as part of the node configuration. Inference requests reach the platform as jobs submitted to the Simple Linux Utility for Resource Management (Slurm) [8], an open-source cluster workload manager, which launches the corresponding model container on an allocated node. The model container implements the preprocessing, inference, and postprocessing stages defined by the chosen implementation template (cf. Section 2) and issues inference calls over gRPC to the local Triton instance. Depending on the template, CPU-bound stages execute either directly in the model container or as additional steps within a Triton ensemble, so Triton orchestrates both GPU and CPU work inside the same execution graph. Each job runs in the name of the user who submitted it, and the model container, started with Apptainer [9], runs under the same identity. The NVIDIA Triton Inference Server runs under a functional account of the platform, manages the GPU workload of the node, and loads models on demand. Which models a job may call is decided beforehand at the web frontend and the REST API through the virtual organization check described in Section 4.

Storage is provided by dCache [10], a high-performance distributed storage system developed at DESY in collaboration with Fermilab and NeIC, hosted at DESY and provided through the Helmholtz Cloud [2] as the dCache InfiniteSpace service [1]. dCache supports multi-protocol access (WebDAV, rclone) and stores both user data and model weights. All dCache storage pools used by the HMZ are located on hardware at DESY, with no replication to external systems. Model container images are published to the CernVM File System (CVMFS) [11], a content-addressable read-only file system developed at CERN for distributing software across grid and HPC sites, from institutionally operated repositories hosted at DESY, where they are stored in unpacked, deduplicated form rather than as layered image archives. The CVMFS client mounted on every GPU node maintains a local cache, so when Slurm launches a container from its CVMFS path, only files not already cached on the node are fetched from the institutional CVMFS infrastructure. Storing images unpacked avoids the pull-and-unpack step of a conventional container runtime and considerably speeds up container startup. No image data is pulled from external registries.

The model integration pipeline runs on DESY's institutional GitLab instance. Model source code, CI/CD pipeline definitions, and the central model registry all reside within this self-hosted GitLab. Container images are built by GitLab CI runners on institutional hardware and published to CVMFS. The container images on CVMFS are readable by any host that mounts the institutional repository.

Node configuration is managed through Puppet, which defines the GPU node software stack, Triton configuration, filesystem mounts, and system dependencies as version-controlled code. This approach enables the operations team to maintain reproducible configurations and to scale the node

PoS(ISGC2026)027

pool by applying the same Puppet manifests to new hardware.

The software stack is dominated by open-source components (Slurm, JupyterHub, Flask, PostgreSQL, Puppet, CVMFS) and institutionally operated services (dCache, Helmholtz ID). Triton Inference Server is available either as the upstream open-source build (BSD-3-Clause) [4, 12] or as a prebuilt NVIDIA container that adds proprietary components under the NVIDIA Deep Learning Container License [13]. Operators select whichever path fits their constraints without changes to the rest of the architecture. No component depends on a remotely hosted commercial cloud service, which eliminates external vendor lock-in and keeps the full stack under institutional control.

4. Data Sovereignty and Access Control

The infrastructure described in Section 3 also defines how user data is governed. Uploaded inputs and generated outputs remain in the user's own dCache area, from which they can be downloaded at any time through the web interface, the REST API, WebDAV, or rclone, and are deleted automatically after a configurable retention period.

This architecture provides data residency by construction rather than by contract. Since the institution that operates the hardware also writes the data policy, residency is enforced by the physical location of the machines, not by obligations placed on a third party. The absence of any external processor simplifies compliance with EU regulations such as the General Data Protection Regulation (GDPR) [14] and the EU AI Act [15], and the institution retains direct access to model provenance and execution logs.

Access to the HMZ is governed by Helmholtz ID, the association's federated single sign-on (SSO) system. All members of the Helmholtz Association and their registered collaborators authenticate using their institutional credentials, so the HMZ maintains no user database of its own and may transitively federate to other identity providers depending on the user's home institution.

Authorization operates through virtual organizations (VOs) managed within the Helmholtz ID infrastructure. VOs represent research collaborations, projects, or organizational units. By default, a deployed model is discoverable and executable by any authenticated Helmholtz user. A model provider can instead restrict access to one or more VOs, in which case the model is visible only to members of those groups and only those members can submit jobs against it or access the model weights stored in dCache. Execution of the model containers accessible through CVMFS is confined to the HMZ's GPU nodes, which accept jobs only through the platform's web interface or REST API, so the same access restrictions apply by this design. The web interface enforces this restriction by reading the VO claims in the user's token and building the model list from the metadata store accordingly. Slurm schedules each job onto a GPU node that it shares with other jobs, and isolation between the jobs of different users and VOs comes from the job identities described in Section 3. User inputs and outputs are stricter still. Each user's data lives in their own dCache area and is visible only to that user, not to other VO members. This is particularly relevant for models trained on restricted datasets or developed within collaborations that require controlled sharing.

The VO mechanism extends to external collaborators without compromising data residency. Research groups can register guest users through Helmholtz ID, granting them membership in specific VOs and thereby access to selected models under the same residency guarantees as internal users. Sharing decisions remain with the model provider rather than the platform operator, which

PoS(ISGC2026)027

preserves the autonomy of each research group over both their models and the conditions under which others may use them.

5. Deployment and Operations

The HMZ's infrastructure is physically located at DESY in Hamburg, but is shared across all 18 research centers of the Helmholtz Association. Any researcher with a Helmholtz ID can use the platform immediately through the public web interface, without per-center provisioning. Data transfer between the user's institution and DESY occurs over the German national research network (DFN) [16], which connects all major research institutions with high-bandwidth, low-latency links. By avoiding per-center inference deployments, the platform also avoids duplicated effort and fragmented model collections, since models from any center can be made available to researchers across the association, subject to access controls defined by the model providers.

The web interface and the REST API are served over HTTPS, and direct transfers to dCache through WebDAV or rclone run over HTTPS as well, so every path between a user's institution and DESY is encrypted with TLS. The browser session is authenticated through the Helmholtz ID single sign-on, the REST API through a token that JupyterHub issues after that login, and dCache through its own mechanisms [10].

The trust model underlying this arrangement is that of shared infrastructure. DESY operates the platform on behalf of the association, funded through the HAICORE initiative. For large datasets, the rclone integration with dCache allows users to stage their data into dCache ahead of submitting an inference job. Submitting a job then incurs only the inference time itself, since the inputs already reside on institutional storage. This is particularly relevant for users at centers far from DESY, where transfer time over the DFN would otherwise be added to every job.

The fixed GPU allocation requires capacity planning based on anticipated demand. The HMZ's current seven-node allocation has been sufficient for the beta phase, but growing adoption will require monitoring usage patterns to determine when additional hardware is needed. Slurm's job queue absorbs demand peaks by introducing waiting times rather than rejecting requests, which is acceptable for batch inference workloads but may require attention for latency-sensitive interactive use cases. Additional GPU nodes can be integrated into the Maxwell cluster and configured through the existing Puppet infrastructure.

Operating the full stack, from GPU node configuration to web service deployment, places significant responsibility on a small development team. The HMZ mitigates this by building on services already provided through the Helmholtz Cloud (dCache InfiniteSpace, Helmholtz ID) and on DESY institutional services (CVMFS, GitLab), so that the HMZ team focuses on the application layer rather than infrastructure operations. Puppet-based configuration management further reduces the operational surface by making node setup reproducible and scalable.

A benefit of institutional operation is environment stability. The inference environment is version-controlled and does not change unless the operations team explicitly updates it. This property supports reproducibility. The Triton version, GPU drivers, and system libraries remain fixed until a deliberate upgrade, so an experiment run today can be repeated the following year on the same software stack. Scientific workloads often come with specific, sometimes unusual requirements. Because every layer of the execution environment is under direct institutional control,

PoS(ISGC2026)027

the platform can accommodate whatever each project demands, a level of flexibility that third-party infrastructure rarely offers.

The platform's design is transferable to other research organizations with comparable infrastructure. The minimum requirements are (1) an HPC cluster or dedicated server with GPU nodes and a job scheduler, (2) a high-performance storage system accessible from the compute nodes, (3) a federated identity provider compatible with OpenID Connect [17], and (4) a self-hosted GitLab instance or equivalent CI/CD system. Organizations within the European Open Science Cloud (EOSC) [18] or similar federated research infrastructures could substitute their own implementations for each component. For example, EGI Check-in could replace Helmholtz ID as the identity federation, and any POSIX-compatible high-performance storage could replace dCache. The remaining software components, including Slurm, JupyterHub, and CVMFS, are open-source and not specific to the Helmholtz ecosystem. For Triton, operators can choose either the open-source build (BSD-3-Clause) or the NVIDIA-distributed container, as discussed in Section 3. The HMZ platform software is itself available on request to research organizations interested in deploying a comparable service.

6. Conclusion

In summary, we presented the Helmholtz Model Zoo as a sovereign, on-premise AI inference platform that provides association-wide model sharing without requiring research data to leave institutional infrastructure. Building exclusively on institutional components provides data sovereignty by construction. For publicly funded research organizations that handle sensitive or regulated data, compliance with data protection requirements is enforced by the infrastructure rather than relying on external guarantees. Because no component depends on proprietary cloud services, the architecture transfers to other research organizations with comparable infrastructure, particularly those operating within federated environments such as the European Open Science Cloud.

The Helmholtz Model Zoo is available at `https://hmz-hub.desy.de`. Support and feedback can be directed to `support@helmholtz-imaging.de` [19].

Acknowledgements

We acknowledge the contributions of DESY IT, Helmholtz Federated IT Services (HIFIS), and Helmholtz AI teams whose expertise was instrumental in developing the platform. Special thanks to the HAICORE initiative for financing the GPU nodes powering the HMZ. This work utilized Maxwell computational resources at DESY, Hamburg, Germany.

References

[1] Uwe Konrad, Tobias Huste, Uwe Jandt, and Tobias Schlauch. Helmholtz Federated IT Services: Innovative Cloud and Software Services for Science. In *Software Engineering 2022 (SE 2022), rSE22 track*, 2022. https://doi.org/10.5281/zenodo.15235978.

[2] Helmholtz Association. Helmholtz Cloud. https://helmholtz.cloud/, 2025.

PoS(ISGC2026)027

[3] Helmholtz Association. Helmholtz AAI. https://hifis.net/aai/, 2025.

[4] NVIDIA. NVIDIA Triton inference server. https://github.com/triton-inference-server/server, 2025.

[5] ONNX Community. Open neural network exchange (ONNX). https://github.com/onnx/onnx, 2025.

[6] Helmholtz Association. HAICORE – Helmholtz AI Compute Resources. https://www.helmholtz.ai/you-helmholtz-ai/computing-resources/, 2025.

[7] Luke Kanies. Puppet: Next-generation configuration management. *;login: The USENIX Magazine*, 31(1):19–25, 2006.

[8] Andy B. Yoo, Morris A. Jette, and Mark Grondona. SLURM: Simple Linux Utility for Resource Management. In *Job Scheduling Strategies for Parallel Processing*, volume 2862 of *Lecture Notes in Computer Science*, pages 44–60, Berlin, 2003. Springer.

[9] Gregory M. Kurtzer, Vanessa Sochat, and Michael W. Bauer. Singularity: Scientific containers for mobility of compute. *PLOS ONE*, 12(5):e0177459, 2017. Singularity has since been renamed to Apptainer.

[10] Tigran Mkrtchyan, Krishnaveni Chitrapu, Vincent Garonne, Dmitry Litvintsev, Svenja Meyer, Paul Millar, Lea Morschel, Albert Rossi, and Marina Sahakyan. dCache: Inter-disciplinary storage system. *EPJ Web of Conferences*, 251:02010, 2021.

[11] Jakob Blomer, Brian Bockelman, Predrag Buncic, Ben Couturier, Daniel-Florin Dosaru, Dave Dykstra, Gerardo Ganis, Manuel Giffels, Hardi Nikola, Nick Hazekamp, Seppo Heikkila, Shahnur Isgandarli, René Meusel, Simone Mosciatti, Radu Popescu, Joshua Randall, Tim Shaffer, Samuel Teuber, Douglas Thain, Ben Tovar, Steve Traylen, and Derek Weitzel. The CernVM file system: v2.7.5, 2020.

[12] NVIDIA. Triton inference server – BSD 3-clause license. https://github.com/triton-inference-server/server/blob/main/LICENSE, 2025.

[13] NVIDIA. NVIDIA deep learning container license. https://developer.download.nvidia.com/licenses/NVIDIA_Deep_Learning_Container_License.pdf, 2021.

[14] European Parliament and Council of the European Union. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). Official Journal of the European Union, L 119, pp. 1–88, https://eur-lex.europa.eu/eli/reg/2016/679/oj, 2016.

[15] European Parliament and Council of the European Union. Regulation (EU) 2024/1689 of the European Parliament and of the Council laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). Official Journal of the European Union, L, 2024/1689, http://data.europa.eu/eli/reg/2024/1689/oj, 2024.

PoS(ISGC2026)027

[16] DFN-Verein. DFN-Verein Mitgliedseinrichtungen. https://dfn.de/wp-content/uploads/2022/06/Mitgliederliste_alphabetisch.pdf, 2022. Member list of the German National Research and Education Network.

[17] Nat Sakimura, John Bradley, Michael B. Jones, Breno de Medeiros, and Chuck Mortimore. OpenID Connect Core 1.0. https://openid.net/specs/openid-connect-core-1_0.html, 2014.

[18] European Commission. European Open Science Cloud (EOSC). https://eosc.eu/, 2025.

[19] Helmholtz Imaging. Helmholtz Imaging. https://helmholtz-imaging.de/, 2025.

PoS(ISGC2026)027