EPiC Series in Computing

Volume 109, 2026, Pages 96–107

Proceedings of EUNIS 2026 Annual Congress

UniGPT Revisited: From a Simple Chatbot to an API-First AI Platform — Two Years of On-Premises LLM Operations

Jonathan Radas[1], Benjamin Risse[1], and Raimund Vogl[1]

University of Münster, Münster, Germany
{jonathan.radas, b.risse, rvogl}@uni-muenster.de

Abstract

In 2024, we introduced uniGPT, an on-premises Kubernetes-based LLM platform at a major German university designed for GDPR compliance, digital sovereignty, and avoiding vendor lock-in. This paper evaluates nearly two years of operation (May 2024 – February 2026), tracing its evolution from a simple chatbot into a multi-modal, API-first AI infrastructure. Using the TOE framework, we analyze this progression as an iterative design cycle triggered by technological, organisational, or environmental factors. We detail 8 key iterations – including frontend and inference engine swaps, adding an OpenAI-compatible API layer, multi-modal services, and RAG pipelines. Notably, we find that >99% of usage now occurs via API rather than the chat frontend. Finally, we offer generalizable lessons for institutions building sustainable on-premises AI infrastructure in higher education.

Keywords: Large Language Models, on-premises AI, Higher Education, Design Science Research, Kubernetes, GDPR, Experience Report, Digital Sovereignty

1 Introduction

The landscape of artificial intelligence in higher education has shifted dramatically since 2024. What began in many institutions as a response to ChatGPT, deploying a simple, GDPR-compliant chat interface, has rapidly evolved into a comprehensive AI infrastructure spanning text generation, image synthesis, translation, transcription, and retrieval-augmented generation (RAG). At the University of Münster, the uniGPT platform exemplifies this transformation: launched in May 2024 as a chatbot service, it now operates as a multi-modal, API-first platform where over 99% of all usage is programmatic, primarily driven by other researchers and embedded AI.

In our previous work [22], we presented the initial design and deployment of uniGPT, motivated by four concerns: (1) data privacy and copyright restrictions that precluded using cloud-based commercial services for sensitive university data; (2) the rapidly escalating cost of software-as-a-service LLM subscriptions at institutional scale; (3) the risk of vendor lock-in; and (4) the need for LLMs as a shared resource for future research. The solution was an on-premises deployment of open-weight models (initially Llama 2 and Mixtral) on a Kubernetes

cluster with NVIDIA GPU nodes. This initial focus has since shifted: crucially, the platform's greatest leverage no longer stems from the services we operate directly, but from empowering researchers and departments to build their own AI-driven applications atop this shared, managed infrastructure.

The following three key contributions form the core of our study:

1. **A structured analysis of 8 major design iterations**, each evaluated through the Technology–Organization–Environment (TOE) framework [28] to characterize the technical, institutional, and external drivers of the platform's evolution. (Section 4)
2. **An analysis of quantitative adoption data** illustrating two years of user growth and a significant paradigm shift from frontend-centric interaction to API-driven usage. (Section 5)
3. **Generalizable lessons for higher education institutions**, providing guidance for building and sustaining on-premises AI infrastructure while balancing digital sovereignty with operational scalability. (Section 6)

2 Background and Related Work

2.1 On-premises LLMs in Higher Education

uniGPT launched at the University of Münster in May 2024, making this university one of the first universities to offer locally hosted LLMs to its members. Other German universities have built comparable infrastructure but chose to run their services on HPC clusters [8] or on bare-metal servers [31], in contrast to the Kubernetes-based deployments favored by OpenAI [18, 16] and Google [11, 10]. Similar efforts have emerged internationally, for instance the University of Michigan's closed AI tool suite [15] or UT Austin's AI-tutor platform UT Sage [26]. Beyond universities, commercial vendors have increasingly targeted higher education, most notably through OpenAI's ChatGPT Edu subscription [19].

Several surveys capture the current state of AI readiness in higher education. The EDUCAUSE AI Landscape Study summarizes strategic planning across more than 900 US institutions [24]; Jin et al. examined adoption policies at 60 universities across six global regions [14]; and Belkina et al. systematically reviewed empirical case studies of GenAI integration in teaching and learning [4].

A common thread across these publications is their focus on *initial* deployments: architectural descriptions of newly launched services [8, 31, 15], or cross-sectional surveys taken at a single point in time [24, 14]. Long-term reports that trace the *evolution* of a university AI deployment remain rare. This paper addresses that gap by reporting how uniGPT's architecture and user patterns changed over two years, structured through the TOE framework and iterations inspired by Design Science Research [13].

2.2 Analysis Frameworks

Design Science Research (DSR) treats information systems as purposefully created artifacts that are iteratively built and evaluated to address real-world problems [13].

The Technology–Organization–Environment (TOE) framework explains technology adoption through three contexts [28, 3]: *technological* (internal and external capabilities), *organisational* (internal resources and structures), and *environmental* (external pressures and regulations). While originally designed for initial adoption, we apply TOE to explain *post-adoption* design iterations, following Zhu et al [32].

3 Methodology

We follow a mixed-methods approach that integrates qualitative operational themes with quantitative usage trends. By synthesizing these two strands, we derive meta-inferences [29] in Section 6—higher-order lessons learned concerning the sustainable operation of on-premises LLM infrastructure in higher education.

The quantitative strand (Section 5) draws on anonymized system telemetry, such as the volume of inference requests and active users. In the qualitative strand (Section 4), we retrospectively analyze uniGPT as a design artifact [13] from May 2024 to February 2026. Specifically, we use Design Science Research cycles to frame the artifact's evolution and the TOE framework to categorize the drivers of each design iteration.

4 System Evolution

4.1 Design Principle Zero: The Existing Infrastructure Foundation

Before detailing the individual design iterations, we highlight the foundational architectural decision: deploying on an existing Kubernetes cluster to reuse available GPU resources from previous institutional projects [30, 5]. This pre-existing infrastructure provided the technological baseline for all subsequent evolution. uniGPT operates within this Kubernetes cluster, built on OpenStack and other open-source cloud technologies, leveraging a mix of NVIDIA GPUs. The cluster was extended sequentially over the two year period, incorporating additional NVIDIA A100 and L40S nodes. All deployments adhere to a GitOps workflow using Helm, Kustomize, and ArgoCD, ensuring rapid, version-controlled, and rollback-capable updates. This pre-existing, container-native GPU infrastructure with Kubernetes orchestration, GPU scheduling, and deployment tooling was already in place. Each subsequent platform change, from model updates to entirely new multi-modal services, could be deployed within days by adding or modifying Helm charts and, in several cases, by using upstream Docker images without modification.

4.2 Design Iterations

Table 1 summarizes the 8 major design cycles. Each cycle was triggered by a combination of technological (T), organisational (O), and environmental (E) factors, consistent with the TOE framework [28]. Each cycle is described in more detail below.

Cycle 1: Initial Chatbot. Driven by faculty demand for LLM integration in programming education, the project addressed the regulatory and administrative barriers of commercial services (specifically GDPR non-compliance and account mandates) by deploying open-weight models and a custom frontend on institutional Kubernetes infrastructure. This iteration was previously presented in [22].

Cycle 2: API layer — Adding LiteLLM. This iteration was triggered by an *organisational shift*: While uniGPT was initially conceived as an open alternative to ChatGPT, institutional demand pivoted toward programmatic access to the hosted models. On the *technical side*, our infrastructure had some headroom, particularly prior to the widespread adoption of reasoning (CoT) models; chat-based workloads are governed by human input latencies, leaving GPU resources underutilized. Thus, we decided to open up our API to researchers at our university.

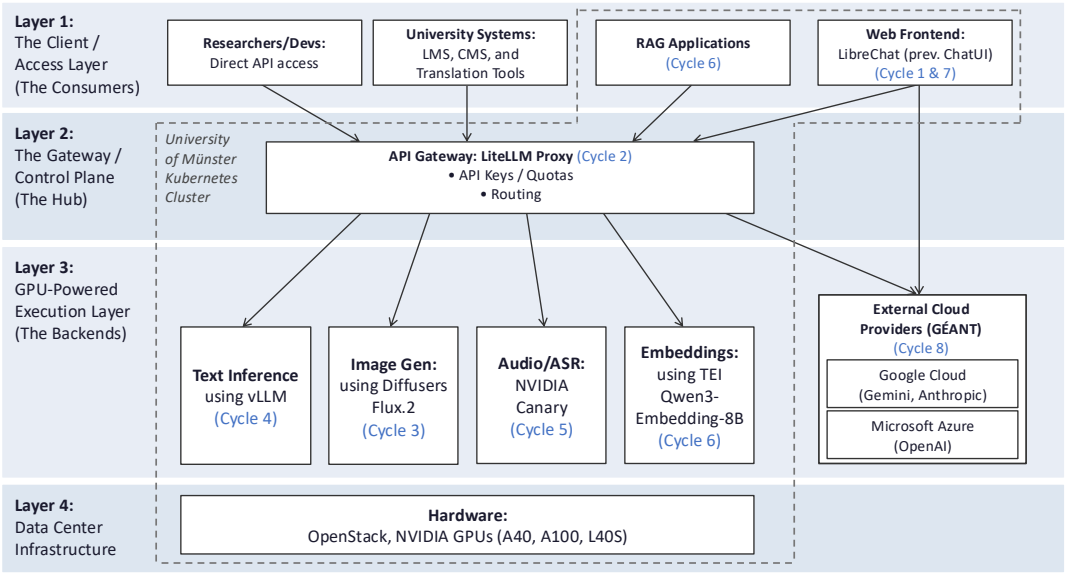

Figure 1: uniGPT architecture overview. The Kubernetes cluster with GPU nodes serves as the foundation for all services.

As the TGI backend lacks a native OpenAI-compatible interface, we required a middleware solution to (1) ensure API compatibility and (2) manage authentication, quotas, and keys. We selected LiteLLM as the proxy layer due to its maturity and comprehensive open-source feature set. It has wide support not only for proxying but also for API key management. Other popular frameworks like Portkey offer only an open-sourced proxy but reserve administrative features for proprietary, closed-source tiers.

This architectural pivot redefined uniGPT from a standalone chatbot into a foundational AI infrastructure. Surprisingly, the researchers using AI came not only from computer science and related domains but also from the humanities (e.g., "Document-level analysis of handwritten notes"), law (e.g., "GDPR compliance of all websites in Germany"), biology (e.g., "Annotation of cell gene signatures") and psychology (e.g., "Modulating personality expressions") [21].

Cycle 3: Image Generation. From an *environmental* perspective, user expectations evolved; AI is no longer perceived as a text-only utility but as a multi-modal ecosystem. This shift was mirrored by *organisational* demand, particularly from public relations and communications departments seeking local image generation. On the *technical* side, fast advancement in research and release of competitive open-weight models like Stable Diffusion [25] made on-premises image generation possible.

Following the emergence of tool-calling capabilities in LLMs, we integrated image generation directly into the chat interface. We deployed Flux.1-dev, which represented the state-of-the-art in open-source synthesis at the time of implementation. Since we already used other parts from the Hugging Face ecosystem (TGI, ChatUI), we used the reference implementation available in their Diffusers library. Therefore, we could integrate the model in a couple of days, using the exact same infrastructure and even physically the same GPUs as for our LLM offering. Ultimately, adding new models or modalities became as routine as deploying another container.

Table 1: 8 DSR design iterations and their TOE drivers.

#	Cycle	T (Technology)	O (Organization)	E (Environment)
1	Initial chatbot	Llama 2 offered sufficient performance for many use cases	Lecturers demanded GDPR-compliant access to LLMs	Students started using LLMs to study
2	API layer: LiteLLM added	LiteLLM offered OpenAI-compatible API proxy	Researchers and developers demanded programmatic access	Industry-wide shift: AI consumed via API, not chat
3	Image generation	Stable Diffusion and Flux reached competitive quality	User requests for image generation	Societal expectation: AI is more than text generation
4	Inference engine swap: TGI → vLLM	vLLM released with superior throughput	OOM errors under growing concurrent load	New model releases required compatible runtime
5	Automatic speech recognition (ASR)	Whisper surpassed SOTA of proprietary models	Researchers' need for GDPR-compliant transcription	Societal expectation: Voice assistants made audio input ubiquitous
6	RAG pipeline	Vector DBs and retrieval frameworks matured	Need for answers grounded in internal university documents	Enterprise RAG became industry standard
7	Frontend swap: ChatUI → LibreChat	LibreChat matured as OSS; multi-model UI support	Users reported ChatUI limitations; need for model switching	Paradigm shift toward reasoning & agentic workflows required tool-calling UIs
8	Multi-provider integration & hybrid cloud	Frontend and API proxy natively supported multiple providers	Users demanded single-interface access instead of separate subscriptions	LLM landscape diversified; Anthropic and Google matched or exceeded GPT-4

Cycle 4: Inference engine swap — TGI to vLLM. The transition from Hugging Face's Text Generation Inference (TGI) to vLLM [17] was driven by *technical* and *environmental* factors. Technologically, TGI suffered from severe VRAM exhaustion and KV-cache fragmentation during peak usage; while Kubernetes orchestration mitigated total service failure, persistent Out-of-Memory (OOM) errors compromised throughput and high availability. Furthermore, TGI lacked robust support for advanced features like native function calling.

From an *environmental* perspective, vLLM emerged as the community standard, often providing day-0 support for new model architectures through direct contributions from model authors such as Mistral or DeepSeek [12]. Relying on smaller inference engines would have necessitated manual implementations or delayed releases. Leveraging our Kubernetes-based infrastructure and organisational readiness, the migration was executed in less than a week. We adopted a phased approach rather than a "big bang" release, deploying all new models via vLLM while gradually deprecating legacy models as part of the standard lifecycle.

Cycle 5: Automatic Speech Recognition. From a *technological* perspective, the release of Whisper [23] established a new baseline for open-source ASR, reaching near-human transcription accuracy. Compared to LLMs, ASR deployment is less demanding: the models are

smaller and operate statelessly, utilizing only temporary storage for audio processing. To maximize institutional utility, we implemented a dual-access layer consisting of a general-purpose frontend and an API for developers and researchers.

Following a comparative multi-language benchmark conducted with our stakeholders, we transitioned to NVIDIA's Canary model [27], which outperformed both Parakeet [27] and Whisper [23] in our specific use cases. As with image generation, the ASR services utilize the same existing shared hardware pool. This architectural synergy allowed us to move from concept to a deployed prototype, including the user interface, within a single day.

Cycle 6: RAG pipeline and Modular Implementation. Addressing the prevalent issue of LLM hallucinations, we established a custom RAG platform to ground model responses in the university's vast collection of internal documents. Recognizing the high effort required for document curation, we focused on providing the computation-heavy components, specifically the LLM and embedding models, as a service, while leveraging our Kubernetes infrastructure to host department-specific middleware and vector databases. This modular "separation of concerns" allows users to deploy custom logic (e.g., using LangChain) tailored to their specific project requirements. For text vectorization, we deployed Qwen3-Embedding-8B, a current leader in multilingual embedding benchmarks [9].

Several projects are using our RAG offering: (1) TutorAI, RAG system for lectures (2) Quin, a chatbot for IT customer support and (3) a RAG system for publication analysis [7] (4) Quin-Max, a bigger RAG system using internal knowledge bases. In a related work at our university, Al Thaher [2] built and evaluated a RAG-based IT support chatbot on top of uniGPT's API and Kubernetes infrastructure. A mixed-methods evaluation found that 24 of 25 users rated the chatbot as providing a correct answer — outperforming web search, generic LLMs, and the IT support website — while 80% of IT staff agreed it could reduce repetitive requests but not fully substitute the support hotline. Crucially, retrieval quality depended on document curation: the chatbot answered five of eight top FAQs fully correctly but failed when the web crawler had not reached relevant subpages.

Cycle 7: Frontend Swap — ChatUI to LibreChat. The transition was driven by several factors. The primary technological trigger occurred in July 2025, when the maintenance of the ChatUI project became uncertain following the temporary suspension of the HuggingChat service. Even though they later restarted HuggingChat and the underlying open-source project, we proceeded with the migration to leverage the advanced capabilities of modern alternatives.

The most significant *environmental* driver was the shift in the LLM landscape from basic chat interfaces toward reasoning-based and agentic tool-calling architectures. OpenAI's o1 model family introduced reasoning models [20], a paradigm subsequently adopted by open-weight models such as DeepSeek-R1 [12]. Concurrently, the broader field moved toward agentic AI systems characterised by proactive planning, tool use (e.g., web search), and autonomous action [1]. Our existing ChatUI frontend was lacking support for these new interaction patterns.

Therefore, we looked for an alternative open-source front end. Since the first introduction of uniGPT in 2024, two open-source frontends emerged as industry standards: LibreChat and OpenWebUI. Both surpass ChatUI in functional maturity. We selected LibreChat over OpenWebUI due to organisational and legal considerations regarding OpenWebUI's non-standard license, which imposed restrictive terms on UI customization.

Initial user feedback on the frontend was highly positive, with many users praising both the design and functionality of the new frontend.

Cycle 8: Multi-Provider Integration and Hybrid Cloud. The *environmental* context was characterized by the rapid diversification of the LLM landscape; by mid-2025, providers such as Anthropic and Google reached or exceeded GPT-4 performance levels [6]. *Organisationally*, users requested access to these models through a single interface rather than managing and paying for separate subscriptions. From a *technology* perspective, we were well prepared for the launch. Both our old frontend and the new frontend (Cycle 7) natively supported multi-provider backend integration.

Operationalization was facilitated by the GÉANT OCRE framework, which provided pre-negotiated cloud contracts for European research institutions. This allowed for simpler on-boarding of Google Cloud (Gemini, Anthropic) and Microsoft Azure (OpenAI) alongside our on-premises models. By integrating these external providers into uniGPT's API-first infrastructure, we maintained a centralized governance and security layer while offering researchers the broadest possible spectrum of state-of-the-art models.

Smaller iteration cycles Besides these larger iterations, there were also smaller iteration cycles updating the models and expanding the hardware. We started with Mistral and Llama 2 and later added various open-weight models of different generations (gpt-oss, Gemma, Qwen). The hardware grew from a single A40 node (4 GPUs) to 2 A100 nodes, 2 A40 nodes and 2 L40 nodes. Besides that, through the API uniGPT was embedded into several university systems, including the Learning-Management-System (LMS), the Content-Management-System (CMS) and embedded into a translation tool utilizing the official university glossary.

5 Adoption and Usage

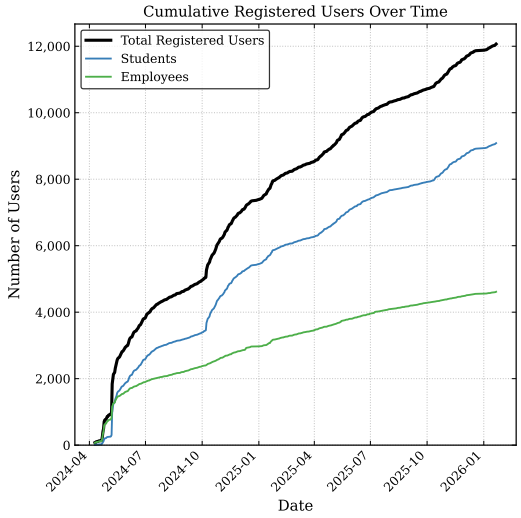

Figure 2: Total users of uniGPT (May 2024 – February 2026). In addition to the continuous growth in student enrollment, the number of employees has also risen, now exceeding 50% of the total staff.

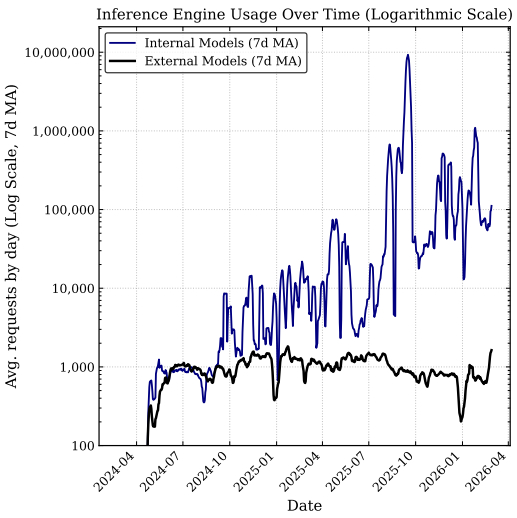

Figure 3: Daily request volume (7-day moving average, log scale) for internally hosted open-weight models versus externally proxied models Both categories started at comparable levels, but internal model usage grew roughly 100x after the introduction of the API layer (Cycle 2).

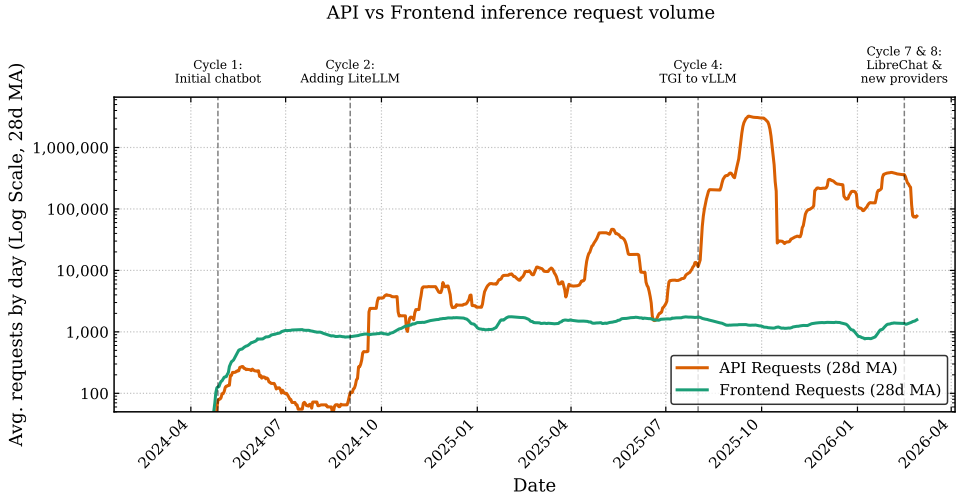

Figure 4: Daily request volume (28-day moving average, log scale) for frontend requests versus direct API requests. The API became visible to researchers in September 2024, quickly surpassing frontend usage (which remains stable between 1,000 and 2,000 daily requests). API requests now exceed frontend requests by a factor of 100, clearly illustrating the platform's transformation from a chatbot to an API infrastructure.

This section presents the quantitative evaluation of the uniGPT artifact based on anonymized system telemetry collected over almost two years. The most significant shift in the platform's usage pattern is the transition from frontend-dominated to API-dominated consumption (Figure 4). After the introduction of LiteLLM (Cycle 2, Section 4), API usage grew rapidly and now accounts for over 99% of all interactions, with peaks reaching 1.66 million local inference requests per day. Cycles 7 and 8 further accelerated demand, particularly for the newly integrated external models (Figure 3).

This pattern is driven by the environmental context identified in our TOE analysis: as developers, researchers, and even administrative workflows increasingly consume AI capabilities programmatically. User input is limited by how much a user types and interactively uses a chat, while API usage does not follow the limits; some researchers might analyze thousands or even millions of documents.

6 Discussion and Lessons Learned

Following the convergent mixed-methods approach described in Section 3, we now merge the quantitative findings from Section 5 with the qualitative insights from Section 4 in a joint display (Table 2). From this integration, we derive 4 lessons learned as meta-inferences.

L1: An OpenAI-compatible API transforms the platform's identity. The quick adoption of the API with >99% API usage was the pivotal moment. This mirrors the industry-wide trend. Universities that only offer a chat UI will underserve their user base. Thinking API-first is thus crucial.

Table 2: Joint display merging quantitative usage data (QUAN) with qualitative operational insights (QUAL) to derive meta-inferences (lessons learned).

QUAN Finding	**QUAL Explanation**	**Meta-Inference**
API usage overtook frontend within days; now >99% of all requests	Researchers adopted API access immediately, scaling far beyond interactive chat	**L1:** An OpenAI-compatible API transforms the platform's identity
Model and provider changes caused no user disruption but increased usage	Decoupling UI from inference backend enabled seamless model and provider changes	**L2:** One unified frontend allows rapid model and provider changes
Embedded AI services adopted by administrative staff and non-technical faculties	Departments with no prior chatbot interest engaged through domain-specific services	**L3:** Multi-modal services attract user groups a text-only chatbot never reaches
All 8 iterations deployed within days or weeks	Kubernetes absorbed every evolution without architectural redesign	**L4:** Container-native GPU infrastructure is the precondition for keeping pace

L2: One unified frontend allows for rapid model and provider changes. From an operational perspective, decoupling the user interface from the underlying inference engine has two advantages. First, as infrastructure providers, we can seamlessly integrate new open-weight models or external commercial models without disrupting the user experience. From a user perspective, a unified interface empowers researchers to easily switch between models — even mid-conversation and directly compare their outputs side-by-side. This flexibility is especially crucial for academic use cases, where evaluating model performance and identifying the best tool for specific research tasks are common requirements.

L3: Multi-modal and embedded AI services attract user groups a text-only chatbot never reaches. Translation, transcription and other tools attracted administrative staff and faculties that had less interest in a text chatbot. This diversification of embedded AI strengthens the political case for continued infrastructure investment by broadening the user base.

L4: Container-native GPU infrastructure is the key enabler. Flexible, yet standardized software-defined infrastructure is the precondition for keeping pace. The most significant finding across all 8 iterations is that the Kubernetes infrastructure absorbed every evolution without architectural redesign. This is not merely an operational convenience — it is what allows the platform to keep pace with the rapidly evolving open-source ecosystem. Users actively track model releases and expect prompt access. The initial investment in flexible, container-native GPU infrastructure was the enabling condition for everything that followed. Other architecture choices likely would have required more time and additional personnel for each cycle.

Economic and strategic viability. The total hardware investment amounted to a few hundred thousand euros, and the whole Kubernetes platform is operated by a small cloud team with only one FTE dedicated to AI services described in this paper. This is modest compared to commercial per-seat subscriptions at approximately $20 per user per month, which would cost over 1 million euros annually even for a fraction of the university's more than 50,000 members.

Beyond cost, the unified frontend and API proxy allow us to add, replace, or remove models and providers within hours, route sensitive data exclusively to on-premises models, and avoid committing to any single vendor — preserving digital sovereignty not as a static policy but as an operational capability.

Limitations While this experience report focused on a single-case study at one large research university in Germany, it might not represent other types, e.g., smaller universities. We also rely on system-level data only. Future work may target a more user-centric view with user surveys or interviews.

7 Conclusion

This paper evaluated nearly two years of continuous operation of uniGPT, an on-premises AI platform at the University of Münster. Using Design Science Research cycles to frame the artifact's evolution and the TOE framework to categorize the drivers of each design iteration, we showed that the platform underwent 8 major changes — from frontend and inference engine swaps to the addition of an API layer, multi-modal services, and a RAG pipeline. The quantitative analysis of system telemetry revealed a paradigm shift: over 99% of usage is now programmatic, with daily peaks reaching 1.66 million local inference requests per day, marking the transition of generative AI from a conversational tool to a foundational institutional infrastructure. Notably, API-driven research adoption spans well beyond computer science into disciplines such as humanities, law, biology, and psychology.

Our core lessons are that combining an OpenAI-compatible API layer with a unified, provider-agnostic frontend enables digital sovereignty — GDPR compliance via on-premise models and freedom from vendor lock-in — while keeping pace with the diversifying model landscape. Furthermore, multi-modal services and embedded AI broaden adoption well beyond technically oriented users. Underpinning all eight iterations, flexible container-native GPU infrastructure on Kubernetes proved to be the foundational enabler that allowed each evolution, from model swaps to entirely new modalities, to be deployed within days rather than months. Economically, a few hundred thousand euros in hardware and a single dedicated FTE compare favourably to per-seat subscriptions, which would exceed one million euros annually for a fraction of the university's 50,000+ members.

Future work will focus on operationalizing agentic workflows, expanding the RAG knowledge base, and embedding AI into more workflows. We will also aim to foster inter-university collaborations to establish a shared, sovereign AI ecosystem.

References

[1] Mohamad Abou Ali and Fadi Dornaika. Agentic AI: A comprehensive survey of architectures, applications, and future directions. *Artificial Intelligence Review*, 2025.

[2] Josef Al Thaher. Exploring retrieval-augmented generation for large language models: Enhancing university it support with a rag-based chatbot. Master's thesis, University of Münster, April 2025.

[3] Jeff Baker. The technology–organization–environment framework. In Yogesh K. Dwivedi, Michael R. Wade, and Scott L. Schneberger, editors, *Information Systems Theory: Explaining and Predicting Our Digital Society, Vol. 1*, pages 231–245. Springer, 2011.

[4] Marina Belkina, Scott Daniel, Sasha Nikolic, Rezwanul Haque, Sarah Lyden, Peter Neal, Sarah Grundy, and Ghulam M. Hassan. Implementing generative AI (GenAI) in higher education: A

systematic review of case studies. *Computers and Education: Artificial Intelligence*, 8:100407, 2025.

[5] Markus Blank-Burian, Jürgen Hölters, and Raimund Vogl. Jupyterhub on an on-premises cloud-a special focus on gpu accelerated machine learning and 3d visualization. In *EUNIS*, pages 69–76, 2021.

[6] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025.

[7] Luis Filipe de Araujo Pessoa and Raimund Vogl. Applications of llm and nlp in the retrieval and analysis of institutional publications. In *Proceedings of EUNIS*, volume 107, pages 60–69, 2025.

[8] Ali Doosthosseini, Jonathan Decker, Hendrik Nolte, and Julian M. Kunkel. Chat ai: A seamless slurm-native solution for hpc-based services, 2024.

[9] Kenneth Enevoldsen, Isaac Chung, Imene Kerboua, Márton Kardos, Ashwin Mathur, David Stap, Jay Gala, Wissam Siblini, Dominik Krzemiński, Genta Indra Winata, and ... Mmteb: Massive multilingual text embedding benchmark, 2025.

[10] Google Cloud. How GKE powers AI innovation. Google Cloud Blog, April 2025. Accessed: 2026-02-24.

[11] Google Cloud. How GKE inference gateway improved latency for Vertex AI. Google Cloud Blog, February 2026. Accessed: 2026-02-24.

[12] Daya Guo, DeepSeek-AI Yang, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *Nature*, 645:633–638, 2025.

[13] Alan R. Hevner, Salvatore T. March, Jinsoo Park, and Sudha Ram. Design science in information systems research. *MIS Quarterly*, 28(1):75–105, 2004.

[14] Yueqiao Jin, Lixiang Yan, Vanessa Echeverría, Dragan Gasevic, and Roberto Martínez Maldonado. Generative AI in higher education: A global perspective of institutional adoption policies and guidelines. *Computers and Education: Artificial Intelligence*, 7:100307, 2024.

[15] Ravi Pendse Jones. How (and why) the University of Michigan built its own closed generative AI tools. *EDUCAUSE Review*, February 2024.

[16] Kubernetes. OpenAI case study. Kubernetes Case Studies, 2018. Featuring Christopher Berner, Head of Infrastructure at OpenAI. Accessed: 2026-02-24.

[17] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention, 2023.

[18] OpenAI. Scaling Kubernetes to 7,500 nodes. OpenAI Blog, January 2021. Accessed: 2026-02-24.

[19] OpenAI. Introducing ChatGPT Edu. https://openai.com/index/introducing-chatgpt-edu/, May 2024. Accessed: 2026-02-24.

[20] OpenAI. OpenAI o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

[21] Leon Pielage, Ole Hätscher, Mitja Back, Bernhard Marschall, and Benjamin Risse. Dynamic personality adaptation in large language models via state machines, 2026.

[22] Jonathan Radas, Benjamin Risse, and Raimund Vogl. Building unigpt: A customizable on-premise llm-solution for universities. In Raimund Vogl, Laurence Desnos, Jean-François Desnos, Spiros Bolis, Lazaros Merakos, Gill Ferrell, Effie Tsili, and Manos Roumeliotis, editors, *Proceedings of EUNIS 2024 annual congress in Athens*, volume 105 of *EPiC Series in Computing*, pages 108–116. EasyChair, 2025.

[23] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning*, pages 28492–28518. PMLR, 2023.

[24] Jenay Robert and Mark McCormack. 2025 EDUCAUSE AI landscape study: Into the digital AI divide. Technical report, EDUCAUSE, Boulder, CO, February 2025.

[25] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022.

[26] Julie Schell, Kasey Ford, and Arthur B. Markman. Building responsible AI chatbot platforms in higher education: An evidence-based framework from design to implementation. *Frontiers in Education*, 10:1604934, 2025.

[27] Monica Sekoyan, Nithin Rao Koluguri, Nune Tadevosyan, Piotr Zelasko, Travis Bartley, Nikolay Karpov, Jagadeesh Balam, and Boris Ginsburg. Canary-1b-v2 and parakeet-tdt-0.6b-v3: Efficient and high-performance models for multilingual asr and ast, 2025.

[28] Louis G. Tornatzky and Mitchell Fleischer. *The Processes of Technological Innovation*. Lexington Books, Lexington, MA, 1990.

[29] Viswanath Venkatesh, Susan A. Brown, and Hillol Bala. Bridging the qualitative–quantitative divide: Guidelines for conducting mixed methods research in information systems. *MIS Quarterly*, 37(1):21–54, 2013.

[30] Raimund Vogl and Markus Blank-Burian. An update on the münster university cloud-technical architecture and user adoption. In *Proc. EPiC Ser. Comput.*, volume 105, pages 126–135, 2025.

[31] Torsten Zesch, Michael Hanses, Niels Seidel, Piush Aggarwal, Dirk Veiel, and Claudia de Witt. Flexible llm experimental infrastructure (flexi) – enabling experimentation and innovation in higher education through access to open llms. *2024 21st International Conference on Information Technology Based Higher Education and Training (ITHET)*, pages 1–8, 2024.

[32] Kevin Zhu and Kenneth L Kraemer. Post-adoption variations in usage and value of e-business by organizations: cross-country evidence from the retail industry. *Information systems research*, 16(1):61–84, 2005.