GEKO: GPUs energieeffizient für KI-Inferenz orchestrieren

Project proposal (VHB) for software campus 2026

Natalie Carl

October 17, 2025

Applicant	Technische Universität Berlin
Project lead	Natalie Carl Technische Universität Berlin FG Skalierbare Softwaresysteme (E-N 17) Einsteinufer 17 10587 Berlin E-Mail: nc@3s.tu-berlin.de
Academic supervision	Prof. Dr.-Ing. David Bermbach Technische Universität Berlin FG Skalierbare Softwaresysteme (E-N 17) Einsteinufer 17 10587 Berlin E-Mail: db@3s.tu-berlin.de
Industry partner	Dr. Sripriya Srikant Adhatarao Huawei Technologies Duesseldorf GmbH Advanced Wireless Technologies Lab Riesstraße 25 80992 München E-Mail: sripriya.srikant.adhatarao@huawei.com
Project start **Project duration**	March 1, 2026 24 Months

Contents

1 Task Definition and Motivation

Function-as-a-Service (FaaS) is a serverless computing model in which developers write small, stateless functions that are invoked by a cloud platform in response to external requests. In this model, the cloud platform manages nearly all aspects of execution, including resource allocation, auto-scaling, and the runtime environment. Especially in edge environments, where resources are scarce, FaaS has proven to be a suitable paradigm for sharing hardware between applications and allocating resources only when they are actually needed. At the same time, the rapid growth of cloud platforms and data centers has made their energy consumption a critical concern. Over the past decades, global data center electricity use has steadily increased without any noticible signs of slowing down. In 2023, data centers in the United States alone consumed $176\,\mathrm{TWh}$, accounting for $4.4\,\%$ of the country's total electricity use, with projections estimating a rise to $6.7\,\%$ to $12\,\%$ by 2028 [She+24]. Despite significant improvements in Power Usage Effectiveness (PUE), rising demand consistently outpaces efficiency gains, and data centers already account for roughly $1\,\%$ of global electricity consumption [Gan+23; Mas+20; Sha24]. This trajectory directly conflicts with the Paris Agreement's target of limiting global warming to well below $2\,^{\circ}\mathrm{C}$, which requires rapid and substantial reductions in greenhouse gas emissions. These developments create a responsibility for both developers and cloud platforms alike to consciously consider and continuously improve the environmental footprint of their operation [Chi21].

The energy demand of artificial intelligence workloads is a particularly pressing issue. Modern inference tasks are heavily GPU-bound, and while GPUs provide the necessary computational performance, they are also associated with high energy costs. This creates a fundamental tension between society's growing reliance on AI-powered services and the urgent need to reduce the carbon footprint of digital infrastructures. To address this challenge, we focus on FaaS as the underlying paradigm. The serverless model provides fine-grained elasticity, centralized infrastructure management, resource sharing across applications, and a large degree of control to the cloud platform, which is particularly valuable both in large-scale cloud data centers and in resource-constrained edge environments. These properties make FaaS a natural foundation for implementing energy-aware orchestration strategies that can adaptively control resource usage. At the same time, a lot of open challenges need to be addressed in order to improve the currently poor energy efficiency of contemporary FaaS platforms [Sha23]. The scientific question that motivates this project is therefore: How can GPU resources for serverless inference be orchestrated in an adaptive and energy-efficient manner, without compromising performance?

The relevance of this question lies in the sustainability of digital infrastructures. As AI models become important components of everyday applications, from medical diagnostics to large language models, operators must reconcile latency and throughput demands with climate responsibility. Practically, today's serverless platforms provide limited support for GPU execution, in general, and fine-grained GPU management, in particular: In most deployments and when available, GPUs remain powered even during idle times, wasting significant amounts of energy. Addressing this inefficiency not only has the potential to greatly affect environmental impact of serverless infrastructure but also lowers operational costs in cloud-scale environments. This issue is even more pressing for Germany, as the country already experiences substatially increased levels of warming compared to global trends and currently even the most pessimistic RPC8.5 scenario [DWD25].

Practical application examples can be found in inference-as-a-service offerings. Applications such as real-time medical image analysis or conversational AI systems must respond elastically to fluctuating demand. A dynamic orchestration approach will ensure that GPUs are powered on only when required, while predictive scheduling mechanisms mitigate cold-start overheads. This will enable cloud providers and companies using private clouds to deliver sustainable, latency-sensitive services at scale without sacrificing user experience. The goal of the **GEKO** ("**G**PUs **e**nergieeffizient für **K**I-Inferenz **o**rchestrieren") project is therefore to develop a serverless platform architecture and programming abstractions that will enable deploying, managing, and using serverless functions with GPU support for AI applications in an energy-efficient manner. The platform and abstractions will be made open source, to facilitate scientific and industrial R&D efforts.

1.1 Focus and objectives

FaaS has emerged as a promising abstraction for building scalable and elastic applications. However, despite its advantages, current FaaS platforms remain highly energy-inefficient [Sha23]. This inefficiency stems from two key factors: the strong variance in request loads, which often leads to overprovisioning or idle resources, and the expensive software-level isolation required to execute short-lived functions securely [GF20; Sch+23]. As a result, serverless applications today are far from exploiting their potential for sustainable operation.

At the same time, FaaS has the ideal prerequisites to serve as the core programming model for energy-efficient AI inference [Pat+21]. Its fine-grained elasticity, centralized control of resources, and abstraction from application logic make it a natural fit for orchestrating energy-aware scheduling and adaptive GPU usage. Yet, realizing this potential requires foundational research, since existing platforms offer only limited support in this direction. Notably, most public FaaS services do not provide GPU support at all, leaving no basis for exploring efficient orchestration of inference workloads.

A key reason why FaaS is particularly well suited for sustainable computing lies in its platform-centric model. Instead of requiring every developer to solve sustainability challenges individually, the serverless abstraction concentrates responsibility for energy-efficiency at the platform level. This enables resource sharing, workload consolidation, and energy-aware scheduling to be implemented once and leveraged by all applications running on the platform. In principle, this makes FaaS one of the strongest candidates for aligning large-scale digital infrastructures with sustainability goals, provided that the necessary system mechanisms exist.

The focus of this project is therefore to establish the groundwork for sustainable, scalable GPU-based inference in serverless environments. We aim to provide the missing system-level mechanisms that allow GPUs to be integrated into FaaS platforms and managed adaptively with respect to workload demands. In doing so, the GEKO project seeks to bridge the gap between today's energy-inefficient serverless platforms and a future in which FaaS is the foundation of sustainable AI infrastructure.

1.2 Scientific and/or technical objectives of the project

Building on the motivation outlined above, this project aims to develop the foundations for sustainable and scalable GPU-based inference in serverless platforms. The overarching goal is to transform Function-as-a-Service from an energy-inefficient abstraction into a viable basis for sustainable AI inference infrastructure. To achieve this, we focus on system-level mechanisms that enable adaptive GPU orchestration, efficient resource sharing, and transparent integration of energy-aware scheduling policies into the serverless execution model.

The concrete objectives of this project are threefold. First, we seek to design and implement the missing platform mechanisms that allow GPUs to be exposed as first-class resources in FaaS environments for AI workloads. Second, we aim to develop orchestration strategies that adapt GPU allocation dynamically to workload fluctuations, minimizing idle energy costs while preserving performance. Third, we plan to evaluate the effectiveness of these strategies across a diverse set of inference workloads and deployment settings, thereby quantifying their impact on both energy efficiency and quality of service.

From these objectives, the following technical and research questions emerge:

(1) How can GPUs be exposed and managed as first-class resources in FaaS environments for AI workloads, given the short-lived and highly dynamic nature of serverless functions?

(2) What orchestration strategies can dynamically adapt GPU allocation to workload fluctuations, ensuring high utilization while minimizing idle energy costs?

(3) How do the proposed mechanisms perform across diverse AI inference workloads, and what trade-offs emerge between energy efficiency, performance, and scalability?

To address them, the project will create an open-source prototype of a serverless platform that integrates GPU support and implements energy-aware orchestration mechanisms. This prototype will be designed to be usable and extensible by both the research community and industry practitioners, providing a practical foundation for future work on sustainable AI infrastructures.

1.3 Relation of the project to funding policy objectives/funding program

The project GEKO is closely aligned with the strategic goals of the BMFTR and the objectives of its "Hightech Agenda Deutschland". In particular, it addresses two central themes emphasized in the funding policy: advancing artificial intelligence as a key enabling technology and promoting sustainable digital infrastructures in line with Germany's climate commitments.

First, GEKO contributes to strengthening Germany's technological leadership in AI by addressing the high energy demand of inference workloads. The project develops innovative system-level mechanisms for energy-efficient orchestration of GPU resources in serverless environments. Consequently, it complements existing AI research that predominantly focuses on model efficiency and instead tackles the infrastructure and platform perspective, which are equally important for the practical use of AI technology. This is directly in line with the Hightech Agenda's objective to expand AI research across the full technology stack and secure digital sovereignty in Europe. Second, GEKO has a strong relation to the BMFTR's climate and sustainability goals. The project's central objective, i.e., reducing the carbon footprint of AI workloads in cloud platforms, directly supports Germany's contribution to achieving the climate targets of the Paris Agreement. By focusing on platform-level orchestration and resource sharing, GEKO demonstrates how sustainability can be built into digital infrastructures rather than being left to individual developers or applications. This systemic approach has the potential to deliver significant energy savings at scale, thereby making a measurable contribution to sustainable digitalization.

Finally, the project strengthens education and innovation transfer. Embedded in the Software Campus program, GEKO provides graduate students with the opportunity to develop practical expertise in systems research, cloud platforms, and sustainable computing. In parallel, the project fosters leadership and soft skills that are crucial for future roles in academia and industry. The planned open-source prototype of a serverless GPU platform ensures that the results are not only of academic value but also accessible to the wider research community and industrial stakeholders, thereby accelerating innovation transfer and supporting Germany's role as a leading hub for sustainable AI. In this way, GEKO not only advances fundamental research but also facilitates direct knowledge transfer from academia to industry, ensuring that insights into sustainable AI infrastructures quickly translate into practical innovations in the German and European technology sector.

2 State of the Art in Science and Technology

Function-as-a-Service is a serverless programming model in which developers express applications as small, stateless functions that are invoked on demand by the cloud platform. This abstraction frees developers from concerns about scalability and infrastructure management, while enabling large-scale resource sharing across many tenants. Combined with a pay-per-use billing model, FaaS has quickly become a central paradigm in both industry and research for building elastic applications [Jon+19]. Importantly, this model also has untapped potential as a cornerstone for sustainable computing: If orchestrated carefully, shared resources can be provisioned more efficiently at platform level than if each developer had to optimize for sustainability individually. The GEKO project builds on a solid foundation of scientific groundwork laid by a small but active research community in the field of sustainable serverless computing. This includes early explorations into how serverless abstractions could become enablers for energy-aware resource management and carbon accounting. A cornerstone in this emerging area is the articulation of a broader vision for sustainable serverless computing, which frames FaaS as a potential driver of greener cloud services and highlights key research gaps that remain open [Sha23].

In today's practice, however, serverless computing is far from energy efficient. Current FaaS deployments follow two dominant paths. On the one hand, organizations operate open-source serverless platforms on top of Kubernetes or similar orchestration systems. While this approach offers flexibility, it suffers from heavy reliance on virtualization and container isolation, which introduces substantial overheads; these, in turn, increase per-request energy consumption between 15× and 30×, depending on the virtualization technique used [Sha23]. Recent studies show that software-based isolation layers such as gVisor can significantly degrade efficiency, limiting the potential for sustainable operation [You+19]. On the other hand, large public cloud providers offer commercial FaaS services, which remain largely opaque to researchers and offer only limited resource control. In particular, GPU support is mostly absent from these platforms, making it difficult to realize AI workloads in a serverless fashion. Despite these limitations, the research community has begun to address sustainability in serverless computing. Early work has explored quantifying the energy overheads of function isolation and assigning carbon intensity metrics to individual function invocations [SF24]. Other studies have examined spatio-temporal scheduling, where functions are steered to data centers with lower grid carbon intensity. However, these approaches often conflict with the latency requirements of serverless workloads and cannot fully exploit hardware-level optimizations [Suk+24]. The majority of existing work still focuses on performance-oriented goals, such as reducing cold start latencies, rather than systematically reducing the energy footprint of the platform. In the context of artificial intelligence, the gap is even more pronounced. Modern inference workloads are increasingly GPU-bound, but the lack of GPU integration in public serverless platforms is a contributor to preventing FaaS from being used for scalable AI inference. While research in AI sustainability has made progress on model-level and hardware-level optimizations, the platform dimension, i.e., deciding how, when, and which hardware is activated for inference, remains largely unexplored. As a result, serverless computing today does not yet realize its potential as a key abstraction for sustainable AI inference workloads.

Against this background, the GEKO project addresses a clear research gap. Unlike public FaaS platforms, GEKO builds on an open-source foundation that allows transparent investigation of GPU integration and orchestration. In contrast to existing open-source solutions, it focuses not only on enabling GPU support but also on making the serverless platform itself responsible for hardware usage decisions. This system-level focus avoids the mismatch between application-level assumptions and platform-level realities, ensuring that resources are shared and utilized as efficiently as possible. As a result, GEKO directly advances the state of the art in sustainable FaaS and establishes the groundwork for scalable, energy-efficient AI inference.

3 Partners and Previous Work

The project described is integrated into the Software Campus funding program with Huawei Technologies Deutschland GmbH as the industry partner and Technische Universität Berlin as the academic partner. The industry partner is not funded by the project but offers parallel mentoring and valuable perspectives from the industry.

3.1 Research partner – Technische Universität Berlin

Technische Universität Berlin is one of the most renowned technical universities in Germany. With a wide range of degree programs and a strong focus on research and innovation, Technische Universität Berlin is one of Europe's leading institutions for technical education and science.

Technische Universität Berlin is represented in the project by the Scalable Software Systems chair. Under the direction of Prof. Dr.-Ing. Bermbach, the department researches the software design and experiment-driven evaluation of distributed IT systems in the context of modern application domains. The current focus is on data management systems and application architectures in cloud, edge, and fog computing, particularly with regard to issues of data and application component placement.

The Scalable Software Systems group has already contributed numerous results in the areas of serverless computing, performance engineering, and edge computing. Previously, the group successfully completed three other Software Campus projects (SPENCER, CODES, EMPIRIS), each addressing different aspects ranging from performance benchmarking to serverless architectures and applications in LEO satellite networks. In addition, the chair has contributed expertise in networking as part of the BMBF-funded project 6G NeXt and is currently leading the DFG-funded project OptiFaaS, which, too, focuses on serverless computing. This strong track record of projects and expertise directly underpins the objectives of the proposed GEKO project, providing an excellent foundation for advancing energy-efficient serverless platforms for AI inference.

3.2 Industry partner – Huawei Technologies Deutschland

Huawei Technologies is a leading global provider of information and communications technology with a presence in over 170 countries and a comprehensive portfolio of telecommunications products and solutions. Huawei has been active in Germany since 2001 and has had a significant impact on the country's economy and growth, generating gross value of nearly €2.3 billion and employment effects for tens of thousands of people. Huawei plays a crucial role in digitalization and the introduction of technologies such as 5G/6G, which form the basis for smart cities, Industry 4.0, and sustainable mobility.

For the GEKO project, Huawei Technologies Deutschland contributes extensive experience in the research and development of cloud and AI technologies and provides valuable industrial perspectives on building sustainable IT infrastructures. In particular, the Advanced Wireless Technologies Lab has significant expertise in serverless computing, cloud and edge infrastructures, resource-efficient platform architectures, and sustainability backed by its ongoing SNS-JU project titled EXIGENCE whose main focus is on achieving sustainability in the end-to-end service offerrings of telecommunication domain. This expertise complements the scientific work of the GEKO project partners and supports the transfer of research results into practical, industry-relevant solutions.

3.3 Relationship between research partner and industry partner

Technische Universität Berlin and Huawei Technologies Deutschland already maintain a close and productive cooperation. In 2025, both partners jointly established the Huawei Graduate School, which fosters long-term collaboration in research and education across cloud, edge, and AI technologies. In addition, TU Berlin and Huawei have worked together within the Software Campus program on the project SPENCER, which investigated novel abstractions for edge computing in satellite networks.

These collaborations provide a strong foundation for extending the partnership within the GEKO project.

The GEKO project will involve close cooperation between Technische Universität Berlin and Huawei Technologies Germany, particularly between the Scalable Software Systems chair (TUB) and the Advanced Wireless Technologies Lab (Huawei Technologies Germany GmbH). This cooperation involves primarily a bilateral scientific and technical exchange throughout the entire duration of the project.

The high level of expertise of both partners ensures the project's chances of success: As the mentor and contact person at Huawei, Dr. Sripriya Srikant Adhatarao, Senior Researcher at the Advanced Wireless Technologies Lab at Huawei Technologies in Munich, will actively participate in the requirements analysis, research, and development of the project and provide valuable feedback for continuous quality assurance. As head of the Scalable Software Systems department and technical mentor to the micro-project manager, Prof. Dr.-Ing. David Bermbach will be responsible for scientific quality assurance and will support the publication of findings.

4 Detailed Description of the Work Plan

This section contains detailled descriptions of the project's work packages (WPs) and milestones (section 4.1), time and ressource planning (section 4.2), and financial plans (section 4.3).

4.1 Work packages and milestones

The project comprises four following work packages: analysis, requirements, and system design (WP 1), prototype development (WP 2), prototype evaluation (WP 3), and reporting and publishing (WP 4). Figure 4.3.3 contains a timeline and overview of the entire project duration.

4.1.1 WP 1 – analysis, requirements, and system design

During the analysis and conception phase, the focus is on a comprehensive review of current publications, patents, and research reports, with particular emphasis on energy-efficient function execution, GPU orchestration strategies, and sustainable serverless platform design. This systematic review is intended to yield sound findings that will serve as the basis for the further development of the project.

Close cooperation with the industry partner makes it possible to define the exact scope, fundamentals, and quality dimensions of the project. Both functional and non-functional requriements are identified and recorded in a precise catalog, which includes not only technical aspects but also sustainability metrics such as energy efficiency and carbon impact. This catalog serves as a guideline for the subsequent development and evaluation phases of the project.

Particular attention is paid to applications that can benefit from energy-efficient serverless computing and GPU-accelerated AI inference. Identifying and considering these fields of application helps anticipate future demands on serverless platforms and ensures that the developed concepts and prototypes address both scalability and sustainability requirements in practice. The results of this analysis directly inform the architecture of the open-source prototype to be developed in WP 2.

4.1.2 WP 2 – prototype development

During the prototype development phase, the work is structured into three main parts: system design (WP 2.A), prototype implementation (WP 2.B), and the development of exemplary applications (WP 2.C). Together, these activities ensure that the conceptual requirements defined in WP 1 are translated into a working open-source prototype that demonstrates the practical feasibility and impact of the project.

System design (WP 2.A). The first step is a detailed technical design of the system architecture. Building on the catalog of requirements from WP 1, this design phase translates functional and non-functional objectives into a concrete architectural design, from which we later develop a software blueprint. Particular emphasis is placed on mechanisms for adaptive GPU orchestration, efficient function isolation, and energy-aware scheduling. The design process follows an iterative approach, where early prototypes are continuously validated against the requirements to ensure that every requirement is directly tied to measurable evaluation criteria such as energy efficiency, elasticity, and performance.

Prototype implementation (WP 2.B). Based on the finalized design, the open-source prototype is implemented in iterative development cycles. The prototype builds upon established open-source technologies but introduces novel components for sustainable serverless execution, with a particular focus on GPU management and energy-aware orchestration. Four main subsystems form the backbone of the prototype: (1) a modular serverless runtime that supports multiple programming languages and provides custom GPU abstractions through dedicated libraries; (2) a platform layer capable of handling GPU-intensive workloads and orchestrating them for maximum energy efficiency; (3) mechanisms for software- and hardware-level isolation, enabling systematic comparisons of the effects of different design choices on performance and energy trade-offs; and (4) a modular architecture that allows individual components to be swapped and extended. Continuous integration and testing pipelines are established to guarantee that the prototype converges toward a stable, reproducible, and extensible system.

Exemplary applications (WP 2.C). To demonstrate the applicability and evaluate the practicality of the developed system, a set of exemplary applications is implemented. These applications are drawn from domains where serverless AI inference has high relevance, such as computer vision workloads or natural language processing services. Each application will later serve as a benchmark scenario to validate whether the platform fulfills the functional and non-functional requirements specified earlier. By running these applications in controlled experiments, we can systematically assess the platform's energy efficiency, scalability, and other metrics under realistic workloads. In addition, these applications provide reference use cases for the wider community.

4.1.3 WP 3 – empirical evaluation

The empirical evaluation phase focuses on assessing the prototype developed in WP 2 against the functional and non-functional requirements defined in WP 1. Evaluation activities are conducted iteratively throughout the development process to account for different stages of the prototype and provide continuous feedback for refinement. The evaluation framework includes benchmarks that primarily target metrics of energy efficiency, such as GPU utilization, power consumption, and approximations overall carbon impact. In addition, performance metrics such as latency, throughput, and scalability are measured to ensure that the system meets the expected service levels.

WP 3 also systematically compares different strategies for hardware orchestration, resource management, and isolation mechanisms at both the software and hardware levels. This enables identification of trade-offs between energy efficiency, performance, and system overhead. By running controlled experiments with representative AI inference workloads, the evaluation phase verifies whether the platform achieves its goals of sustainable and scalable serverless execution. The results of WP 3 not only provide quantitative evidence of the effectiveness of the developed system but also guide potential adjustments to the prototype and inform best practices for energy-efficient serverless platform design.

4.1.4 WP 4 – publishing and reporting

In the final work package, we focus on two main activities. First, the developed software artifacts will be released as open source. This ensures that other researchers, developers, and organizations can benefit from the findings and the sustainable serverless platform developed within the GEKO project. The open-source release provides full access to the source code, documentation, and relevant resources, promoting transparency, reproducibility, and enabling further development and innovation within the broader community. Preparing the software for release involves ensuring high code quality, comprehensive documentation, and proper formatting, and runs continuously alongside the ongoing development and extension of the prototype.

In the final months of the project, a comprehensive project report will be produced. This report documents the entire project lifecycle, providing detailed insights into the analysis, development, and evaluation phases. It offers a thorough assessment of the milestones achieved, discusses challenges encountered and the approaches taken to resolve them, and provides recommendations for potential future extensions of the platform. The report compiles technical details, methodological approaches, results, and conclusions to give a holistic overview of the project outcomes and their significance for sustainable serverless AI computing.

Milestones. At the end of each work package, a clearly defined milestone marks the completion of the corresponding project phase. Together, these milestones provide measurable checkpoints to ensure that the project progresses according to plan and achieves its objectives.

M1 WP 1 concludes with a written and elaborated catalog of requirements that serves as the foundation for all subsequent development.

M2 WP 2 is finalized with a working open-source prototype, demonstrating the feasibility of sustainable serverless GPU orchestration.

M3 WP 3 culminates in a set of realistic use cases and benchmarks that enable systematic comparison of different serverless platforms, in general, and orchestration and isolation strategies, in particular.

M4 Finally, WP 4 delivers a comprehensive project report documenting the entire project lifecycle, including technical results, evaluation findings, and recommendations for future work.

4.2 Time and ressource planning

For our calculation, we define one person-month (PM) to mean a student assistant working 40 hours per month. The work packages partly overlap to ensure smooth transitions between project stages.

WP	timeline	duration	PM
1	03/26–09/26	6 months	27
2	09/26–11/27	14 months	33
3	01/27–11/27	10 months	15
4	01/27–03/28	14 months	33

Figure 4.1: Overview of person-months and work packages. The overview does not include the project lead, as it is not financed through Software Campus.

In total, the project requires 108 PM, which corresponds to 3 student assistants working 40 hours per month and one student assistant working 60 hours per month over the duration of the entire project. Subtask within individual work packages are distributed equally among student assistants; hence, we calculate the PM based on the four students assistants working eqally on the current tasks, according to their monthly hours. A more detailed timeline, which includes milestones and subcomponents of the work packages, can be found in 4.3.3.

4.3 Financial plan and preliminary calculations

Based on the work and resource planning in the previous section, we now present the financial planning for the GEKOproject. This represents a preliminary calculation to the best of our knowledge and belief, based on the principle of economical and appropriate use of funds. Overall, a total amount of 114,964.00 € is requested, with financial items for staff (student assistants), business trips (Software Campus trainings/summit), and hardware (GPUs). An overview of all expenses can be found in figure 4.3.

4.3.1 Staff

As described in section 4.2, three student assistants working 40 hours per month and one student assistant working 60 hours per month are required. They are hired based on the TV Stud III (Tarifvertrages für studentische Beschäftigte III). TU Berlin calculates with 17 € per hour worked by a student assistant, which covers their wage, taxes, and insurance costs. Due to the shortage of qualified applicants in the highly competitive field of computer science, it is necessary to incentivize applications by highly qualified applicants by paying a benefit ("Zulage") on top of the wage set by TV Stud III. According to the establishment agreement for bonuses ("Dienstverenbarung Zulage"[1]),

[1] https://www.static.tu.berlin/fileadmin/www/10004224/Antraege/Personaldeckungszulage_sigd___.pdf

the bonus is set to 50 %. Recent hirings by the research group have shown that it is impossible to fill positions without doing so.

The technical setup for the student assistants is provided by regular funds from the research group.

4.3.2 Business trips

As part of the project, the project lead will attend Software Campus trainings and two Software Campus summits. Additional travel to disseminate the results of the research conducted in this project are paid through the research group's regular funds. Table 4.2 shows an overview of all travel related to the project and their planned cost. As all planned communication between the project lead and industry partner will be virtual, no additional travel costs emerge.

For project managers of microprojects within the Software Campus, participation in one summit per year and a total of four training sessions is required. These events take place at various locations in Germany, which are not yet known, and will each require two overnight stays. For project managers participating in these events, a total of 100.00 € is estimated for travel to and from the venue, for example by train. In addition, based on the recommendations and rules of TU Berlin, a flat rate of 80.00 € per night is calculated for accommodation expenses (e.g., in hotels).

Event	**Dest.**	**Travel**	**Hotel**	**Per diem**	**Duration**	**Total**
Software Campus summit '26	DE	100.00 €	80.00 €	28.00 €	3 days	344.00 €
Software Campus summit '27	DE	100.00 €	80.00 €	28.00 €	3 days	344.00 €
SWC Training 1	DE	100.00 €	80.00 €	28.00 €	3 days	344.00 €
SWC Training 2	DE	100.00 €	80.00 €	28.00 €	3 days	344.00 €
SWC Training 3	DE	100.00 €	80.00 €	28.00 €	3 days	344.00 €
SWC Training 4	DE	100.00 €	80.00 €	28.00 €	3 days	344.00 €
Σ total expenses						2,064.00 €

Figure 4.2: Overview of planned trips and respective expenses.

4.3.3 Hardware

In order to successfully complete the project, additional hardware is required. In particular, four graphics processing units will be used to sufficiently mimic real-world systems and to guarantee realistic results during the empirical evaluation phase. The detailled hardware requirements and planned expenses are listed in table 4.3. In the appendix, we show comparisons of current prices of the GPUs.

Item	Number	Individual item	Expenses
Staff	0822	Student assistant (40 hours per month)	24,480.00 €
	0822	Student assistant (40 hours per month)	24,480.00 €
	0822	Student assistant (40 hours per month)	24,480.00 €
	0822	Student assistant (60 hours per month)	36,720.00 €
Σ personell expenses			**110,160.00 €**
Items less than 800 €	0831	AMD Radeon RX 9070 XT	635.55 €
	0831	AMD Radeon RX 9070 XT	635.55 €
	0831	AMD Radeon RX 9070 XT	635.55 €
	0831	AMD Radeon RX 9070 XT	635.55 €
Σ hardware expenses			**2,542.2 €**
Travel	0846	Software Campus	2,064.00 €
Σ travel expenses			**2,064.00 €**
Σ total expenses			**114,766.20 €**

Figure 4.3: Overview of the financial plan.

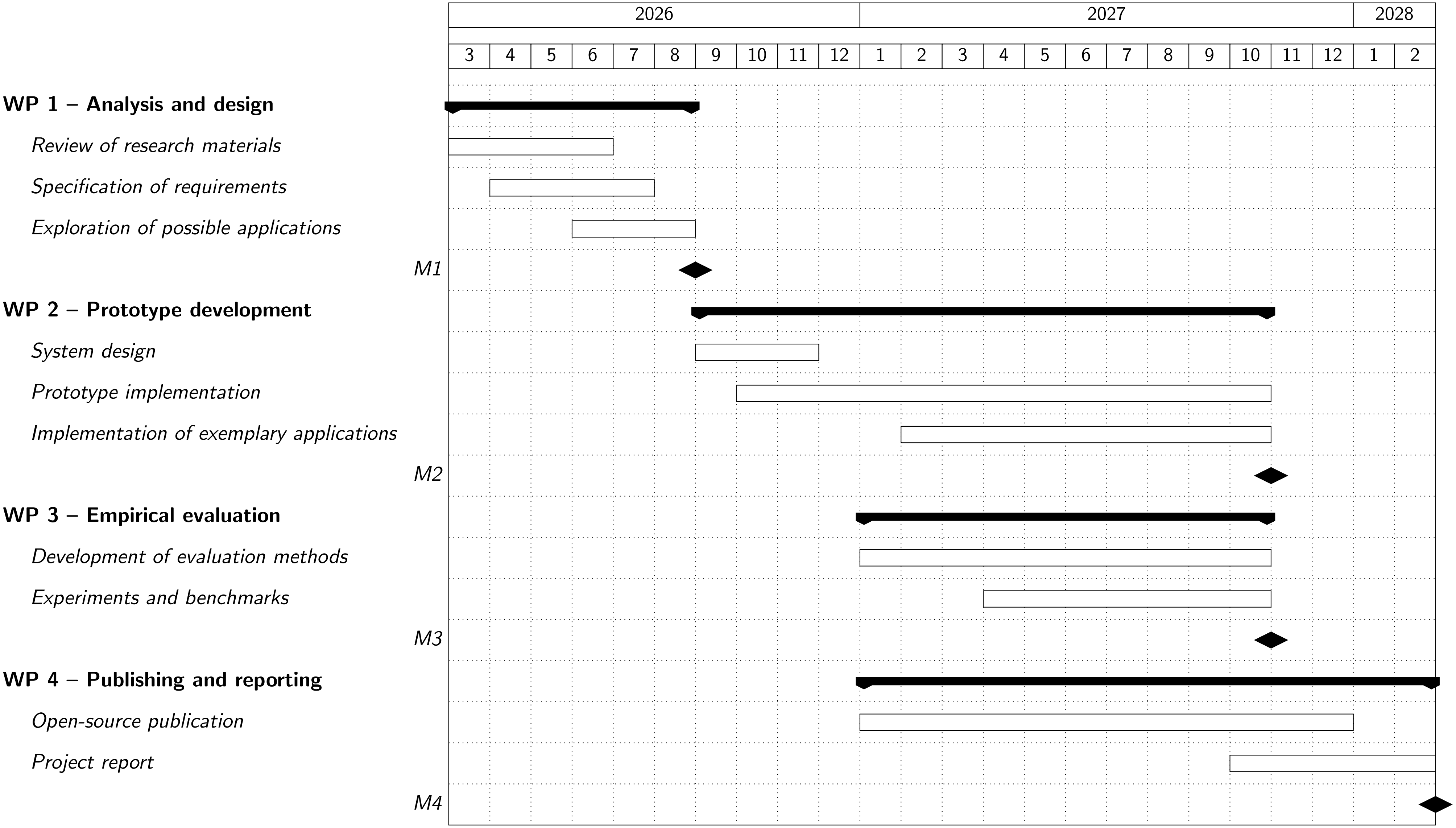


Figure 4.4: Timeline of the GEKO project.

5 Utilization Plan

This section outlines how the results of the GEKO project will be utilized, both in terms of industrial application and research impact, highlighting potential economic benefits, knowledge transfer, contributions to scientific advancement, and opportunities for follow-up research and innovation.

5.1 Economic prospects of success

Although the primary focus of the GEKO project is research rather than immediate commercialization, the outcomes nonetheless provide valuable opportunities for industrial utilization. All software artifacts developed during the project will be released as open-source under licensing models that permit both commercial and non-commercial use. This ensures that the broader community can not only directly adopt the developed serverless platform but also extend it further for domain-specific applications, thereby fostering innovation beyond the project itself.

The industry partner Huawei Technologies Deutschland GmbH can integrate the project results into its own products and services, particularly in the areas of sustainable cloud and edge computing infrastructures. In addition, Huawei benefits from the direct knowledge transfer enabled by the collaboration with the Scalable Software Systems group at Technische Universität Berlin, for instance through joint development of GPU orchestration strategies and evaluation methods. This exchange strengthens Huawei's capacity to bring more energy-efficient AI inference services into production environments while contributing to the competitiveness of the European and global ICT markets.

5.2 Scientific and technical prospects of success

The primary focus of the GEKO project is the scientific utilization of its results. On the one hand, the project offers excellent opportunities for publishing its findings in internationally recognized scientific venues. The project leadership's extensive track record of high-quality publications in the field of distributed and serverless systems underscores the potential for impactful dissemination. Within the scope of GEKO, two publications are planned, for which the project lead will assume first authorship. The specific conferences or journals for these publications are to be determined. Additionally, the project results will be documented in technical reports and shared through open-source repositories, further contributing to the research community and enabling follow-up work, in addition to the final report.

In addition, the GEKO project offers opportunities for students to gain qualifications. As outlined in above, the student assistants are to remain involved in follow-up projects with the Scalable Software Systems group even after the project has ended and, depending on their qualifications, will have the opportunity to pursue a doctorate at Technische Universität Berlin, for example. The fundamentals of scientific work at PhD level are already being taught as part of the GEKO project. In addition, the project can also provide a framework for qualification work by students at Technische Universität Berlin at bachelor's and master's level, for example the supervision of final theses, as well as project work on topics whose results can be used scientifically.

5.3 Scientific and economic connectivity

The GEKO project establishes a strong link between scientific research and industrial application by combining the Scalable Software Systems group's expertise in serverless computing with Huawei's experience in cloud and edge infrastructures. Through the development of an open-source, energy-efficient serverless platform, knowledge and technical innovations are disseminated to both the research community and industry partners, enabling adoption, adaptation, and further experimentation. The project results will be shared through publications, technical reports, and open-source releases, fostering collaboration and follow-up research in sustainable AI inference. Looking forward, GEKO provides a foundation for future work, including extending the platform to additional AI workloads, exploring cross-cloud and hybrid deployments, investigating predictive scaling strategies, and evaluating long-term sustainability impacts. These directions open opportunities for new research initiatives, collaborative projects, and potential commercial applications, ensuring that the scientific and economic benefits of the project continue to grow beyond its immediate duration.

6 Bibliography

[Chi21] Andrew A. Chien. “Driving the cloud to true zero carbon”. en. In: *Communications of the ACM* 64.2 (Jan. 2021), pp. 5–5. ISSN: 0001-0782, 1557-7317. DOI: 10.1145/3445037.

[DWD25] DWD, Geschäftsbereich Klima und Umwelt. *Klimastatusbericht Deutschland Jahr 2024*. Offenbach, 2025.

[GF20] Samuel Ginzburg and Michael J. Freedman. “Serverless Isn’t Server-Less: Measuring and Exploiting Resource Variability on Cloud FaaS Platforms”. en. In: *Proceedings of the 2020 Sixth International Workshop on Serverless Computing*. Delft Netherlands: ACM, Dec. 2020, pp. 43–48. ISBN: 978-1-4503-8204-5. DOI: 10.1145/3429880.3430099.

[Gan+23] Anshul Gandhi, Dongyoon Lee, Zhenhua Liu, Shuai Mu, Erez Zadok, Kanad Ghose, Kartik Gopalan, Yu David Liu, Syed Rafiul Hussain, and Patrick Mcdaniel. “Metrics for Sustainability in Data Centers”. en. In: *ACM SIGEnergy Energy Informatics Review* 3.3 (Oct. 2023), pp. 40–46. ISSN: 2770-5331. DOI: 10.1145/3630614.3630622.

[Jon+19] Eric Jonas, Johann Schleier-Smith, Vikram Sreekanti, Chia-Che Tsai, Anurag Khandelwal, Qifan Pu, Vaishaal Shankar, Joao Carreira, Karl Krauth, Neeraja Yadwadkar, Joseph E. Gonzalez, Raluca Ada Popa, Ion Stoica, and David A. Patterson. *Cloud Programming Simplified: A Berkeley View on Serverless Computing*. en. arXiv:1902.03383 [cs]. Feb. 2019. DOI: 10.48550/arXiv.1902.03383.

[Mas+20] Eric Masanet, Arman Shehabi, Nuoa Lei, Sarah Smith, and Jonathan Koomey. “Recalibrating global data center energy-use estimates”. en. In: *Science* 367.6481 (Feb. 2020), pp. 984–986. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.aba3758.

[Pat+21] Panos Patros, Josef Spillner, Alessandro V. Papadopoulos, Blesson Varghese, Omer Rana, and Schahram Dustdar. “Toward Sustainable Serverless Computing”. In: *IEEE Internet Computing* 25.6 (Nov. 2021), 42–50. ISSN: 1941-0131. DOI: 10.1109/mic.2021.3093105.

[SF24] Prateek Sharma and Alexander Fuerst. “Accountable Carbon Footprints and Energy Profiling For Serverless Functions”. en. In: *Proceedings of the ACM Symposium on Cloud Computing*. Redmond WA USA: ACM, Nov. 2024, pp. 522–541. ISBN: 979-8-4007-1286-9. DOI: 10.1145/3698038.3698531.

[Sch+23] Trever Schirmer, Nils Japke, Sofia Greten, Tobias Pfandzelter, and David Bermbach. “The Night Shift: Understanding Performance Variability of Cloud Serverless Platforms”. In: *Proceedings of the 1st Workshop on SErverless Systems, Applications and MEthodologies*. SESAME ’23. New York, NY, USA: Association for Computing Machinery, May 2023. DOI: 10.1145/3592533.3592808.

[Sha23] Prateek Sharma. “Challenges and Opportunities in Sustainable Serverless Computing”. en. In: *ACM SIGEnergy Energy Informatics Review* 3.3 (Oct. 2023), pp. 53–58. ISSN: 2770-5331. DOI: 10.1145/3630614.3630624.

[Sha24] Prateek Sharma. “The Jevons Paradox In Cloud Computing: A Thermodynamics Perspective”. In: (2024). arXiv: 2411.11540 [cs.DC].

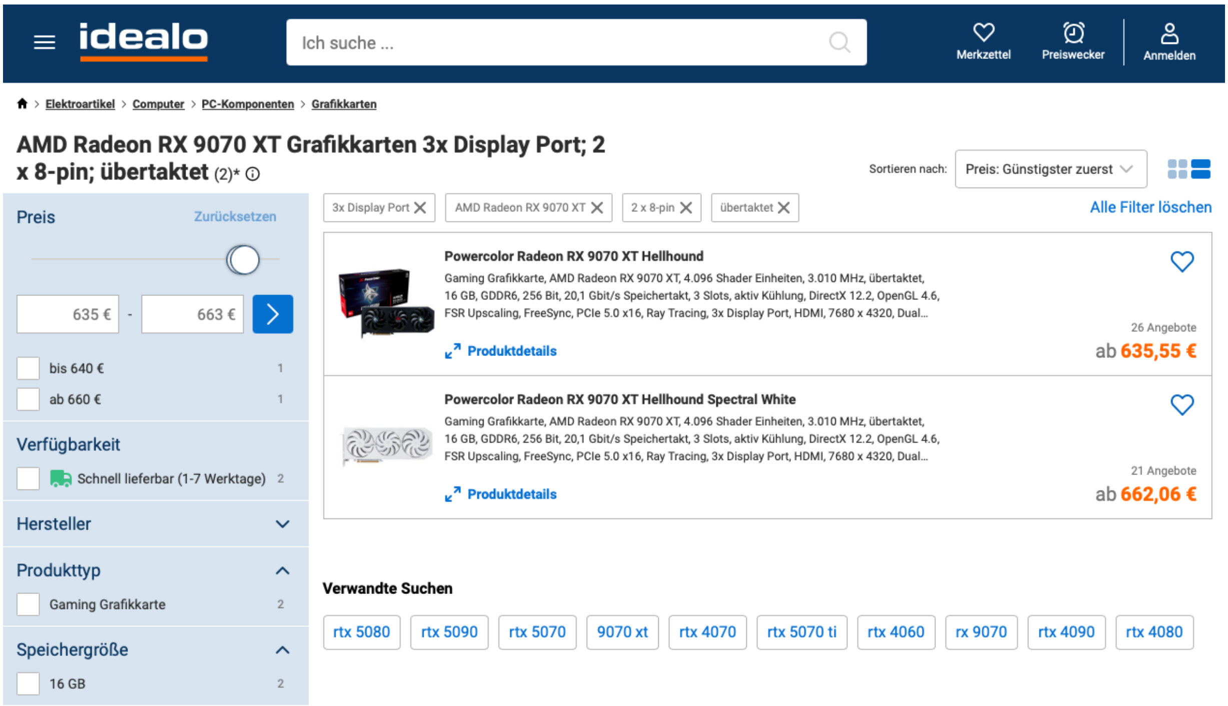
[She+24] Arman Shehabi, Sarah Smith, Dale Sartor, Richard Brown, Magnus Herrlin, Jonathan Koomey, Eric Masanet, Nathaniel Horner, Inês Azevedo, and William Lintner. “United States Data Center Energy Usage Report”. en. Tech. rep. LBNL–1005775, 1372902. June 2024, LBNL–1005775, 1372902. DOI: 10.2172/1372902.

[Suk+24] Thanathorn Sukprasert, Abel Souza, Noman Bashir, David Irwin, and Prashant Shenoy. “On the Limitations of Carbon-Aware Temporal and Spatial Workload Shifting in the Cloud”. en. In: *Proceedings of the Nineteenth European Conference on Computer Systems*. Athens Greece: ACM, Apr. 2024, pp. 924–941. ISBN: 979-8-4007-0437-6. DOI: 10.1145/3627703.3650079.

[You+19] Ethan G Young, Pengfei Zhu, Tyler Caraza-Harter, Andrea C Arpaci-Dusseau, and Remzi H Arpaci-Dusseau. "The True Cost of Containing: A gVisor Case Study". In: *11th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 19)*. 2019.

A Appendix

A.1 Hardware



AMD Radeon RX 9070 XT Grafikkarten 3x Display Port; 2 x 8-pin; übertaktet (2)*

Powercolor Radeon RX 9070 XT Hellhound

Gaming Grafikkarte, AMD Radeon RX 9070 XT, 4.096 Shader Einheiten, 3.010 MHz, übertaktet, 16 GB, GDDR6, 256 Bit, 20,1 Gbit/s Speichertakt, 3 Slots, aktiv Kühlung, DirectX 12.2, OpenGL 4.6, FSR Upscaling, FreeSync, PCIe 5.0 x16, Ray Tracing, 3x Display Port, HDMI, 7680 x 4320, Dual...

Produktdetails

26 Angebote

ab 635,55 €

Powercolor Radeon RX 9070 XT Hellhound Spectral White

Gaming Grafikkarte, AMD Radeon RX 9070 XT, 4.096 Shader Einheiten, 3.010 MHz, übertaktet, 16 GB, GDDR6, 256 Bit, 20,1 Gbit/s Speichertakt, 3 Slots, aktiv Kühlung, DirectX 12.2, OpenGL 4.6, FSR Upscaling, FreeSync, PCIe 5.0 x16, Ray Tracing, 3x Display Port, HDMI, 7680 x 4320, Dual...

Produktdetails

21 Angebote

ab 662,06 €

Figure A.1: Comparison of current offers for AMD Radeon 9070 XT. Last visited on October 17th, 2025.