Smart Health 36 (2025) 100577

Contents lists available at ScienceDirect

Smart Health

journal homepage: www.elsevier.com/locate/smhl

Dynamic fog computing for enhanced LLM execution in medical applications

Philipp Zagar [a,b,*], Vishnu Ravi [a], Lauren Aalami [a], Stephan Krusche [b], Oliver Aalami [a], Paul Schmiedmayer [a]

[a] Stanford Mussallem Center for Biodesign, 318 Campus Drive, Stanford, 94305, CA, United States
[b] Technical University of Munich, Boltzmannstraße 3, Garching bei München, 85748, Bavaria, Germany

ARTICLE INFO

Dataset link: https://github.com/StanfordSpezi, https://github.com/StanfordBDHG

Keywords:
Open-source software
Large language models
Fog computing
Decentralized infrastructure
Mobile app development
iOS

ABSTRACT

The ability of large language models (LLMs) to process, interpret, and comprehend vast amounts of heterogeneous data presents a significant opportunity to enhance data-driven care delivery. However, the sensitive nature of protected health information (PHI) raises concerns about data privacy and trust in remote LLM platforms. Additionally, the cost of cloud-based artificial intelligence (AI) services remains a barrier to widespread adoption. To address these challenges, we propose shifting the LLM execution environment from centralized, opaque cloud providers to a decentralized and dynamic fog computing architecture. By running open-weight LLMs in more trusted environments, such as a user's edge device or a fog layer within a local network, we aim to mitigate the privacy, trust, and financial concerns associated with cloud-based LLMs. We introduce *SpeziLLM*, an open-source framework designed to streamline LLM execution across multiple layers, facilitating seamless integration into digital health applications. To demonstrate its versatility, we showcase *SpeziLLM* across six digital health applications, highlighting its broad applicability in various healthcare settings.

1. Introduction

The convergence of digital technology and healthcare has revolutionized medical monitoring and intervention, generating vast amounts of data through electronic health records (EHRs), wearable devices, and digital health applications. When used responsibly, this data can significantly enhance healthcare delivery, patient engagement, and clinical outcomes (Tapuria et al., 2021). Yet, efficiently interpreting and utilizing this heterogeneous data remains a challenge.

LLMs can bridge the gap between complex raw data and meaningful, human-readable insights by answering questions, summarizing, and interpreting structured or unstructured textual data, sometimes outperforming human experts (Thirunavukarasu et al., 2023). Applications such as the open-source *LLMonFHIR* system demonstrate how LLMs can directly interact with Fast Healthcare Interoperability Resources (FHIR)-formatted health data, enabling users to effectively query their information (Schmiedmayer et al., 2024). However, deploying LLMs in healthcare presents challenges related to data privacy, trust, transparency, regulatory compliance, and the high costs associated with centralized cloud computing environments (Weidinger et al., 2021; Yuan, Tang, Jiang, & Hu, 2023).

To address these limitations, we propose shifting LLM execution from centralized cloud environments to a decentralized, dynamic fog computing architecture (Bonomi, Milito, Zhu, & Addepalli, 2012). **Fog computing**, an extension of *edge computing* (Cao,

* Corresponding author.
E-mail addresses: zagar@stanford.edu (P. Zagar), vishnur@stanford.edu (V. Ravi), laalami@stanford.edu (L. Aalami), krusche@tum.de (S. Krusche), aalami@stanford.edu (O. Aalami), schmiedmayer@stanford.edu (P. Schmiedmayer).

https://doi.org/10.1016/j.smhl.2025.100577
Received 19 March 2025; Accepted 24 March 2025
Available online 2 April 2025

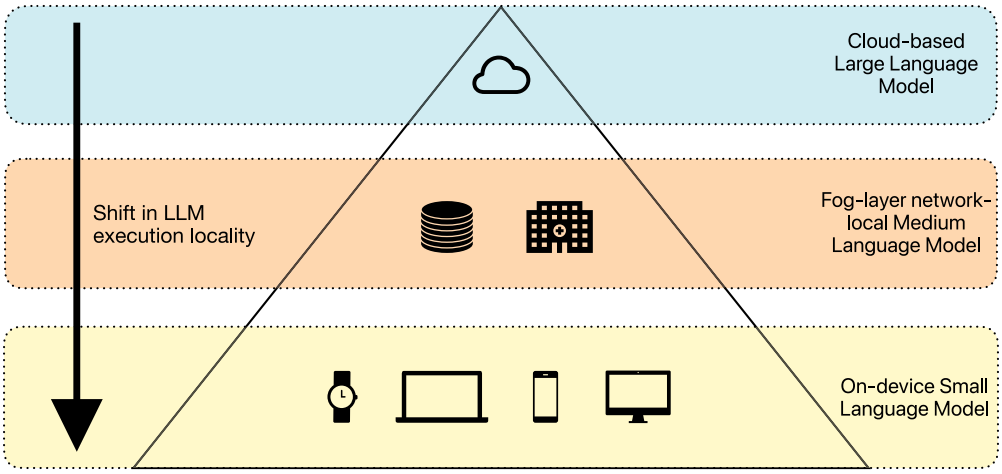


Fig. 1. Transition of LLM inference from cloud-based platforms to user-proximate environments.

Liu, Meng, & Sun, 2020), strategically positions substantial computing resources closer to data sources through intermediary fog nodes (Bonomi et al., 2012). This approach reduces latency, optimizes bandwidth, and enhances privacy and trust by leveraging computational power in localized, trusted environments. Fog architectures typically consist of three layers (Iorga, Feldman, Barton, Martin, Goren, & Mahmoudi, 2018):

- **Edge Layer:** Internet of Things (IoT) and end-user devices at the network edge, where data is generated and utilized.
- **Fog Layer:** Positioned between the cloud and the edge, fog nodes with substantial computational power process data closer to the source, offering a more trusted compute environment.
- **Cloud Layer:** Centralized units with vast computational resources, but raising concerns regarding privacy, trust, and cost.

However, fog computing introduces significant complexities, especially for resource-intensive LLMs. Managing limited resources, dynamic availability, and network constraints across decentralized nodes makes it challenging to achieve a viable fog architecture while balance latency, cost, and privacy—key factors in digital health applications relying on LLM interactions.

Research Question

What software mechanisms are necessary to enable a dynamic fog computing architecture to distribute LLM inference tasks across decentralized edge, fog, and cloud environments with the goal to enhancing privacy, trust, and cost efficiency in digital health applications?

We introduce *SpeziLLM* as an open-source, LLM-agnostic framework designed to investigate these software mechanisms across different LLM inference environments. SpeziLLM aims to transparently integrate various LLMs across diverse execution environments, while abstracting technical complexities of fog orchestration. We demonstrate *SpeziLLM*'s versatility through six digital health applications across different execution modalities. We discuss the insights, challenges, and opportunities of a transparent LLM execution across different processing layers and their applicability in different digital health applications.

2. Architecture: Fog computing for LLMs

We propose a software architecture that dynamically shifts LLM inference closer to user devices based on trust, availability, and computational resources (Fig. 1). *Edge computing* enhances trust, security, and privacy by processing data locally (Cao et al., 2020), but adapting compute-intensive LLMs remains challenging due to the limited computational power of edge devices. Efforts to address these constraints focus on model compression techniques such as quantization (Frantar, Ashkboos, Hoefler, & Alistarh, 2023; Gunter et al., 2024). *Fog computing* extends cloud capabilities to the network's edge, reducing latency and optimizing resources through dynamic task dispatch and agile resource allocation (Bonomi et al., 2012; Iorga et al., 2018). At the core of this approach are fog nodes—heterogeneous devices near the edge that enhance processing efficiency, provide a uniform interface for interacting with computational infrastructure, regardless of execution locality. By processing data closer to the source, they help mitigate security and privacy concerns, offering a more controlled environment for handling sensitive medical data in compliance with HIPAA/GDPR regulations before engaging cloud-based LLMs.

We introduce a simplified mental model to standardize LLM interactions within a fog computing framework, comprising four core components (Fig. 2). A **Schema** defines model configurations (e.g., model type, parameters) and remains immutable once initialized. Schemas are passed to a centralized **Runner**, which delegates inference tasks to local, fog, or remote **Platforms**. Each platform manages its LLM execution environment, translating schemas into active **Sessions** that execute inference tasks while maintaining in-flight context and state.

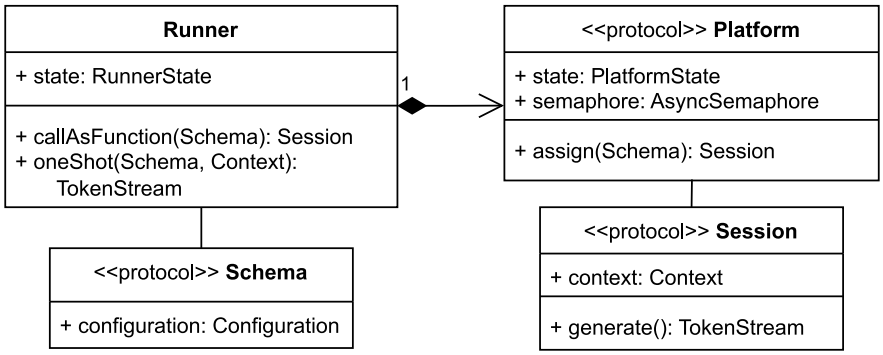


Fig. 2. Mental model of LLM interactions as a UML class diagram.

2.1. Context-aware generation

Limited context windows present challenges when integrating LLMs with extensive data, particularly on resource-constrained devices. Injecting large datasets is not only cost-prohibitive due to per-token inference pricing but also constrained by strict context length limitations. We address this using an industry-standard *tool calling* mechanism, a specific instantiation of Retrieval-Augmented Generation (RAG), enabling structured external data interaction while remaining transparent to the LLM execution layer.

We implement context injection through a declarative, LLM-agnostic domain-specific language (DSL) that abstracts technical complexities and state management across all execution modalities (Fig. 3). Requests trigger inference on the **LLM Service**, which may return tool invocations (functions and parameters). The **Runner** executes these tools concurrently and seamlessly reintegrates the results into the context.

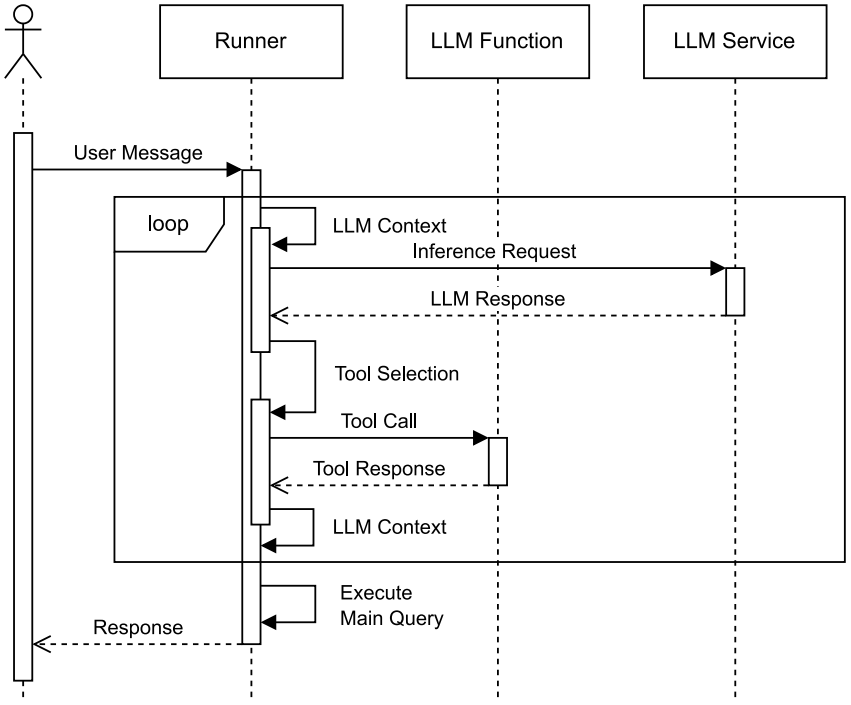


Fig. 3. UML sequence diagram illustrating LLM tool calling within our architecture.

2.2. Dynamic task dispatching

While fog computing provides decentralized computing resources near user devices, these resources may not always be available, requiring a transparent and dynamic task dispatch mechanism across the distributed architecture. Our software architecture is designed to allocate inference tasks to optimal fog nodes based on proximity and capability.

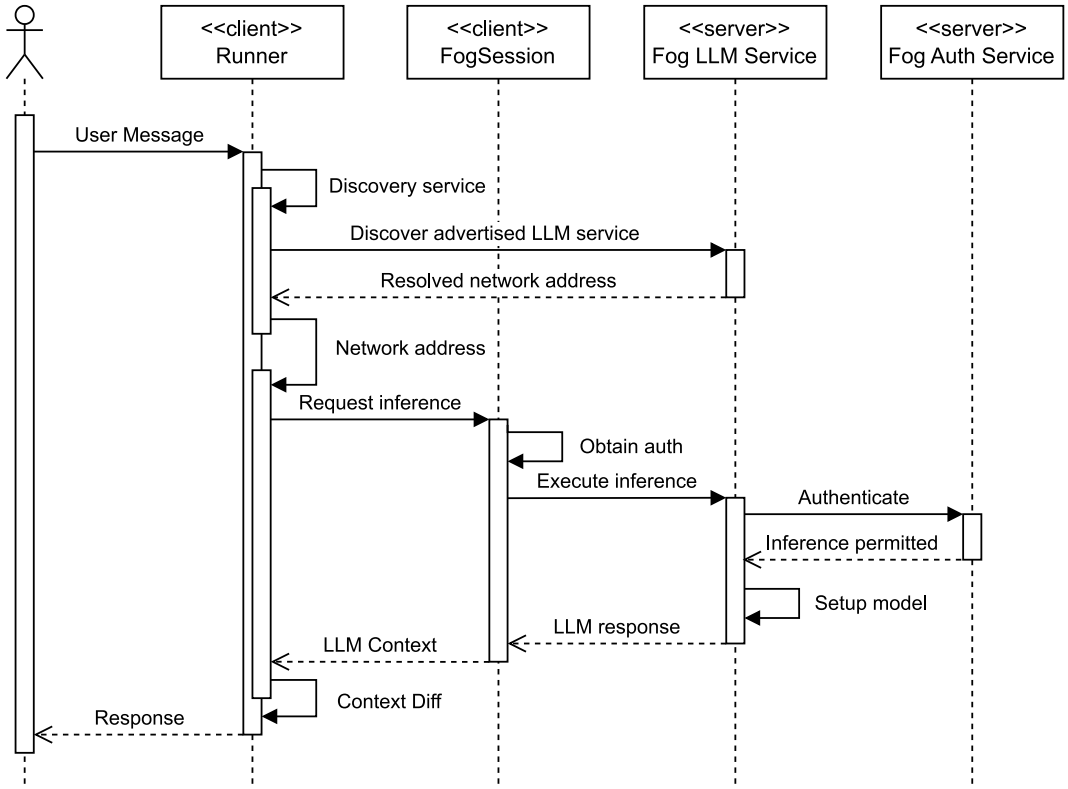


Fig. 4. UML sequence diagram depicting LLM inference execution within the fog computing layer.

Fig. 4 illustrates inference task allocation. Upon receiving a user message, the **Runner** discovers available **Fog LLM Services** in the local network. The selected **FogSession** securely dispatches the inference task with authorization credentials. If approved by the **Fog Auth Service**, the fog node executes the model inference and streams the results directly back to the user's edge device.

3. Reference implementation: SpeziLLM

Based on the fog computing LLM architecture described in Section 2, we developed the *SpeziLLM*[1] reference implementation—an open-source, MIT-licensed Swift framework that simplifies modular and flexible LLM integration into digital health applications. SpeziLLM supports Apple's major operating systems (iOS, macOS, visionOS), leveraging Apple's software ecosystem and optimizing performance on Apple Silicon. It is embedded within the *Stanford Spezi*[2] ecosystem of modules, enabling easy integration into digital health applications and seamless access to health data in Spezi-based applications.

SpeziLLM provides modular and composable LLM components, including schemas, session state management, and streamlined error handling, along with prebuilt, customizable UI elements that accelerate development, while aligning with the mental model in Section 2. The reference implementation ensures uniformity and consistency across different LLM execution layers.

3.1. Cloud layer

SpeziLLM integrates with cloud-based LLM services using the widely adopted OpenAI-compatible application programming interface (API), ensuring compatibility with major providers such as OpenAI, Anthropic and Gemini. By leveraging OpenAPI schema-based code generation, SpeziLLM remains future-proof and provides mechanisms to easily replicate similar setups for other cloud-based API integrations. The framework abstracts tool-calling integration through a declarative DSL, simplifying structured interactions with external data sources. Comprehensive examples and documentation are available in the open-source documentation[3].

[1] https://github.com/StanfordSpezi/SpeziLLM/
[2] https://github.com/StanfordSpezi
[3] https://swiftpackageindex.com/stanfordspezi/spezillm/documentation/spezillmopenai/functioncalling

Table 1
A list of six health applications built between 2023 and 2024 using SpeziLLM.

Application	Description	Developers	Models	Status
1. LLMonFHIR	Explains and provides helpful context for FHIR-formatted patient data using LLMs. Aims to enhance patient health literacy.	Stanford Biodesign Digital Health	Cloud, Fog, Edge (Llama 3.1 8B)	Study enrollment started
2. OwnYourData	Increases diversity in cancer clinical trials through LLM- and FHIR EHR-based patient/study matching.	Stanford Biodesign Digital Health and OwnYourData LLC & Startup LLC	Cloud (OpenAI GPT-4o)	In development
3. HealthGPT	Enables users to query and interact with their Apple Health data using natural language.	Stanford Biodesign Digital Health	Cloud, Fog, Edge (Llama 3.1 8B)	minimum viable product (MVP) built
4. Nourish	Meal tracking app designed to support outpatients with Avoidant/Restrictive Food Intake Disorder (ARFID).	Stanford Biodesign Digital Health & Lucile Packard Children's Hospital Stanford	Cloud (OpenAI GPT-4o)	MVP built; study planned
5. Stronger	Tracks protein intake and resistance exercise training in postmenopausal research participants.	Stanford Biodesign Digital Health & Stanford Medicine	Cloud (OpenAI GPT-4o)	MVP built; study planned
6. Intake	Pre-populates medical intake forms based on FHIR records via interactive LLMs.	Stanford Biodesign Digital Health	Cloud (OpenAI GPT-4o)	MVP built

3.2. Fog layer

SpeziLLM's fog computing integration utilizes dynamically discoverable network-local LLM inference services, aligning with the OpenAI API for seamless interchangeability. Fog nodes are containerized using Docker for rapid deployment and leverage the open-source inference tool Ollama[4]. SpeziLLM employs standardized service discovery protocols like Multicast Domain Name Service (mDNS) and Apple's Bonjour, ensuring broad compatibility across Linux and Apple ecosystems. Inference requests are securely authorized and dispatched using SSL and JWT-based authentication, preserving privacy and trust within the execution environment.

3.3. Edge layer

SpeziLLM's edge execution is powered by *MLX*[5], an open-source inference library optimized for Apple's hardware ecosystem. Local inference is managed by SpeziLLM, which serializes inference tasks and automates efficient resource handling, including downloading and persistent storage of LLM model files. SpeziLLM's MLX inference infrastructure natively supports popular open-weight model families such as Llama, Gemma, Phi, and DeepSeek. The MLX integration allows SpeziLLM to seamlessly incorporate additional open-weight models, enabling developers to download, store, and manage local LLM models within Spezi-based applications.

A comprehensive technical analysis and performance evaluation of SpeziLLM is provided by Nissen et al. (2025) demonstrating the variety of supported models and their computational efficiency as well as performance in answering medical-related questions.

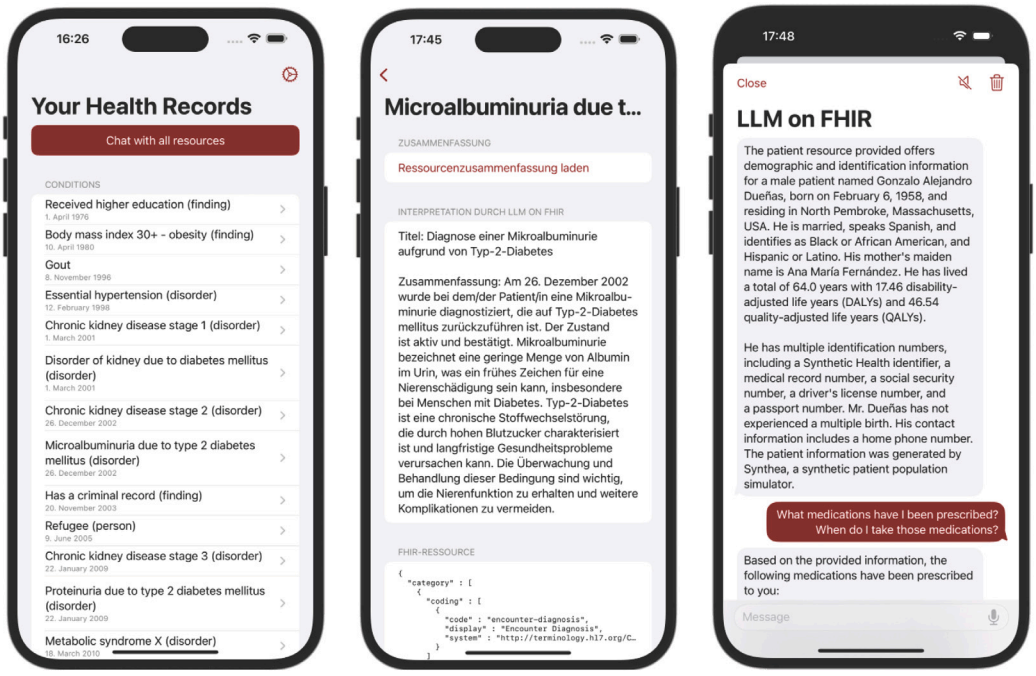
4. Methods

We conducted six case studies to demonstrate SpeziLLM's applicability across diverse mobile health platforms (Table 1). For both the edge and fog layers of SpeziLLM, we employed the *Llama 3.1 Instruct* model (8B variant[6]) in its 4-bit quantized version (Frantar et al., 2023), a common quantization approach for mobile LLM deployments to optimize inference speed. Edge inference was executed locally on an iPhone 16 Pro (A18 Pro System on a Chip (SoC), 8 GB RAM), while fog inference was carried out on a fog node hosted on a MacBook Pro 16" (M4 Pro, 48 GB RAM) running within a Docker container. For cloud inference, we utilized OpenAI's GPT-4o (`gpt-4o-2024-08-06`).

As the LLM landscape evolves rapidly, our methods do not aim to provide an exhaustive performance benchmark or evaluation of model output quality as demonstrated by Nissen et al. (2025). Instead, our focus is on the case studies conducted with SpeziLLM and the architectural decisions underlying its deployment across local, fog, and cloud environments. By analyzing these case studies, we offer comparative insights into inference speed, providing a practical reference for expected performance across different execution layers based on model size.

We integrated SpeziLLM into all six case studies (Table 1), each using the cloud execution layer. Two applications also leveraged fog and edge layers with minimal code modifications, enabled by SpeziLLM's uniform interface.

[4] https://github.com/ollama/ollama
[5] https://github.com/ml-explore/mlx
[6] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct



(a) Overview of available records. (b) Resource interpretation. (c) Interactive health chat.

Fig. 5. Screenshots from LLMonFHIR iOS Application.

5. Results

The applications summarized in Table 1 demonstrate SpeziLLM's adaptability across various digital health scenarios.

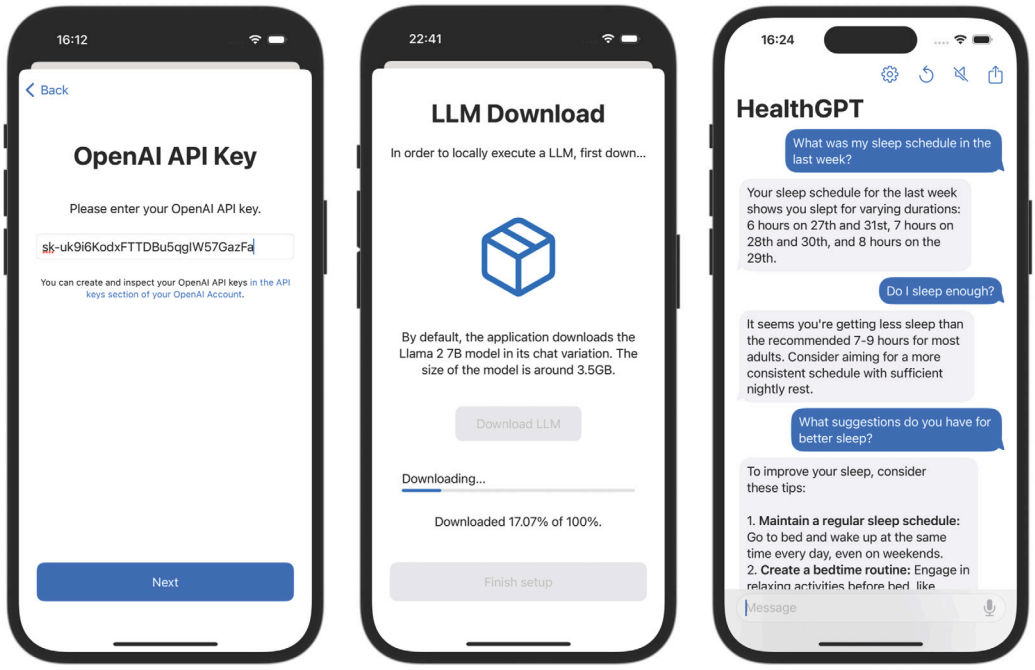
LLMonFHIR (Fig. 5) (Schmiedmayer et al., 2024) dynamically selects the optimal LLM inference environment by combining multiple LLM layers. Simple tasks such as summarization, transformation, or interpretation of individual FHIR resources (Fig. 5(b)) are handled by local or fog-based models with higher trust levels (Section 2), minimizing exposure of patient identifiers to remote cloud providers. These outputs support more complex tasks, such as the interactive chat view (Fig. 5(c)), where a cloud-based OpenAI LLM interprets locally generated summaries within user dialogues without accessing raw FHIR resources. LLMonFHIR particularly benefited from SpeziLLM's prebuilt UI elements and declarative DSL for OpenAI's tool calling (Fig. 3), reducing development complexity and enhancing parallel processing efficiency.

SpeziLLM's uniform interface and cross-layer dynamic dispatch are integral to the rearchitected *HealthGPT*[7] (3) application (Fig. 6), which utilizes multiple LLM execution environments. This open-source Stanford Spezi-based tool enables users to query Apple Health data via natural language, retrieving metrics such as sleep, step count, exercise minutes, body mass, and heart rate through a bidirectional speech-to-text chat interface (Fig. 6(c)). In addition to OpenAI cloud LLM inference (Fig. 6(a)), HealthGPT supports local and fog-layer models (Fig. 6(b)), ensuring sensitive health data and user interactions remain in trusted environments.

In general, local and fog models produced comprehensible outputs, though they were typically more verbose and less detailed than GPT-4o. Fig. 7 shows that local inference experienced delays due to hardware constraints, such as limited main memory in mobile devices. To improve response time, we leveraged an observed inverse correlation between context size and inference speed, reducing the local LLM health data context from 14 days to five.

SpeziLLM also facilitated application development for the digital health projects *Nourish* (4), *Stronger*[8] (5), and *Intake*[9] (6), all built during the 10-week *CS342 Building for Digital Health*[10] course at Stanford University. Students integrated SpeziLLM into their apps, enabling chat interfaces and tool-calling mechanisms for domain logic interactions. Through iterative student feedback, we validated the framework's abstraction level, API/DSL design, and functionality. Notably, *Intake* leverages LLMs to automate medical form pre-population using patient FHIR records, effectively transforming raw EHR data into structured insights—a key capability that makes LLMs particularly well-suited for digital health applications (Section 1).

[7] https://github.com/StanfordBDHG/HealthGPT
[8] https://github.com/CS342/2024-Stronger
[9] https://github.com/CS342/2024-Intake
[10] https://cs342.stanford.edu



(a) Collect OpenAI API key during onboarding.

(b) Download the selected local LLM.

(c) Chat with Apple Health records.

Fig. 6. Screenshots of the HealthGPT iOS application.

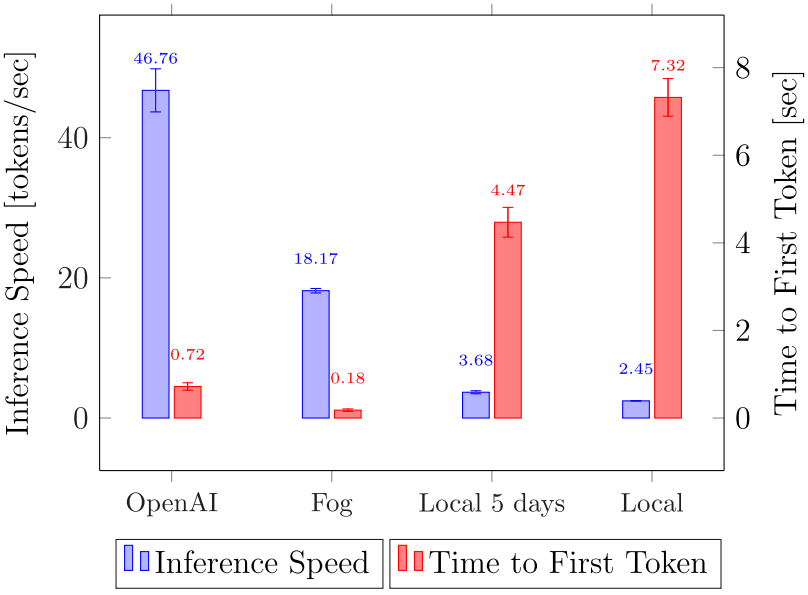


Fig. 7. Comparison of inference speed and time to first token for different LLMs in HealthGPT. Each measurement represents the mean of five quantifications, with standard deviation in brackets. Measurements were taken in response to the question, "*How much did I sleep last week, and how can I improve my sleep?*".

6. Discussion

Our results demonstrate the feasibility of using fog nodes in Docker containers for easy deployment, though they faced performance constraints due to limited hardware-backed LLM inference acceleration. Interestingly, Fig. 7 shows that despite these constraints, the fog layer achieved a low time-to-first-token latency, outperforming cloud-based inference and local edge execution.

Even with Docker-imposed limitations, the fog layer exceeded natural conversational speed (200–300 words per minute or five to seven tokens per second) while running on consumer hardware. Still, we acknowledge that the high cost of modern, optimized hardware remains a barrier to widespread adoption.

Our case studies, including HealthGPT and LLMonFHIR, show that edge-based LLMs handled simple tasks effectively but struggled with deeper contextual reasoning, often producing verbose and less precise outputs than cloud-based GPT models. Sensitive data can be preprocessed in more trusted local or fog layers before potentially interacting with more powerful but less trusted cloud resources. Techniques such as few-shot prompting may help mitigate these limitations by better leveraging small model strengths.

Advancements in mobile inference frameworks like Android's AICore and Apple's MLX promise improved local execution and memory efficiency (Gunter et al., 2024; Nissen et al., 2025). SpeziLLM's adaptable architecture and unified interface are well-positioned to integrate these developments, enabling secure and efficient LLM adoption in digital health.

7. Conclusion

Our fog-computing-based architecture (Section 2) and SpeziLLM reference implementation (Section 3) mitigate concerns related to privacy, trust, and cost when using LLMs in medical applications. These challenges stem from sensitive data transmission to opaque cloud services and resource-intensive computation. We leverage fog computing by dynamically assigning LLM inference tasks to edge, fog, or cloud layers based on task complexity and data sensitivity, all managed through a uniform interface.

SpeziLLM's multi-layered approach enables efficient mobile inference, decentralized fog node discovery and dynamic dispatch, and seamless cloud integration via an OpenAI-compatible API and declarative DSL. Evaluations across six digital health applications validated its versatility and ease of integration.

The key contribution of SpeziLLM is its unified, LLM-agnostic interface, which transparently allocates inference tasks across edge, fog, and cloud environments based on complexity and data sensitivity. Lightweight, specialized models can handle simple tasks locally, generating privacy-filtered outputs for fog or cloud-based processing of more complex tasks. Ongoing advancements in local inference efficiency, particularly in memory-constrained environments, will further enhance SpeziLLM's performance and applicability.

CRediT authorship contribution statement

Philipp Zagar: Conceptualization, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **Vishnu Ravi:** Software, Validation, Writing – review & editing. **Lauren Aalami:** Validation, Writing – original draft, Writing – review & editing. **Stephan Krusche:** Writing – review & editing. **Oliver Aalami:** Conceptualization, Supervision, Validation, Writing – review & editing. **Paul Schmiedmayer:** Conceptualization, Methodology, Software, Supervision, Validation, Writing – original draft, Writing – review & editing.

Acknowledgments

We thank the *Stanford Mussallem Center for Biodesign* for supporting this project and our digital health research.

Funding

This work was supported by the *German Academic Exchange Service* (DAAD) "Internationale Forschungsaufenthalte für Informatikerinnen & Informatiker" scholarship [grant number 91886835] and the *Bavaria California Technology Center (BaCaTec)* project fund [grant number 10 2023-2].

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. The SpeziLLM framework and the larger Stanford Spezi ecosystem are open-source and licensed using the MIT license. Stanford Spezi is being used to foster a digital health ecosystem and teach the next generation of digital health leaders.

Data availability

We have included links to all the open-source software used in this manuscript. Further information and an overview of our open-source tools are available at https://github.com/StanfordSpezi and https://github.com/StanfordBDHG.

References

Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on mobile cloud computing* (pp. 13–16). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/2342509.2342513.

Cao, K., Liu, Y., Meng, G., & Sun, Q. (2020). An overview on edge computing research. *IEEE Access*, *8*, 85714–85728. http://dx.doi.org/10.1109/ACCESS.2020.2991734, URL https://ieeexplore.ieee.org/document/9083958.

Frantar, E., Ashkboos, S., Hoefler, T., & Alistarh, D. (2023). GPTQ: Accurate post-training quantization for generative pre-trained transformers. http://dx.doi.org/10.48550/arXiv.2210.17323, arXiv:2210.17323, URL https://arxiv.org/abs/2210.17323.

Gunter, T., Wang, Z., Wang, C., Pang, R., Narayanan, A., Zhang, A., et al. (2024). Apple intelligence foundation language models. arXiv:2407.21075, URL https://arxiv.org/abs/2407.21075.

Iorga, M., Feldman, L., Barton, R., Martin, M., Goren, N., & Mahmoudi, C. (2018). *Fog computing conceptual model*. Gaithersburg, MD: Special Publication (NIST SP), National Institute of Standards and Technology, http://dx.doi.org/10.6028/NIST.SP.500-325.

Nissen, L., Zagar, P., Ravi, V., Zahedivash, A., Reimer, L. M., Jonas, S., et al. (2025). Medicine on the edge: Comparative performance analysis of on-device LLMs for clinical reasoning. arXiv:2502.08954, URL https://arxiv.org/abs/2502.08954.

Schmiedmayer, P., Rao, A., Zagar, P., Ravi, V., Zahedivash, A., Fereydooni, A., et al. (2024). LLM on FHIR – demystifying health records. http://dx.doi.org/10.48550/arXiv.2402.01711.

Tapuria, A., Porat, T., Kalra, D., Dsouza, G., Xiaohui, S., & and, V. C. (2021). Impact of patient access to their electronic health record: systematic review. *Informatics for Health and Social Care*, *46*(2), 194–206. http://dx.doi.org/10.1080/17538157.2021.1879810, PMID: 33840342, arXiv:https://doi.org/10.1080/17538157.2021.1879810.

Thirunavukarasu, A. J., Ting, D. S. J., Elangovan, K., Gutierrez, L., Tan, T. F., & Ting, D. S. W. (2023). Large language models in medicine. *Nature Medicine*, *29*(8), 1930–1940. http://dx.doi.org/10.1038/s41591-023-02448-8.

Weidinger, L., Mellor, J., Rauh, M., Griffin, C., Uesato, J., Huang, P.-S., et al. (2021). Ethical and social risks of harm from language models. arXiv:2112.04359, URL https://arxiv.org/abs/2112.04359.

Yuan, J., Tang, R., Jiang, X., & Hu, X. (2023). Large language models for healthcare data augmentation: An example on patient-trial matching. arXiv:2303.16756, URL https://arxiv.org/abs/2303.16756.