Kamino: A Scalable Architecture to Support Medical AI Research Using Large Real World Data

Fongci Lin
Section of Biomedical Informatics and Data Science School of Medicine
Yale University
New Haven, CT, USA
fongci.lin@yale.edu

Patrick Young
Department of Laboratory Medicine
School of Medicine
Yale University
New Haven, CT, USA
patrick.young@yale.edu

Huan He
Section of Biomedical Informatics and Data Science School of Medicine
Yale University
New Haven, CT, USA
huan.he@yale.edu

Jimin Huang
Section of Biomedical Informatics and Data Science School of Medicine
Yale University
New Haven, CT, USA
jimin.huang@yale.edu

Roger Gagne
Digital & Technology Solutions
Yale New Haven Hospital
New Haven, CT, USA
roger.gagne@yale.edu

Daniel Rice
Digital & Technology Solutions
Yale New Haven Hospital
New Haven, CT, USA
daniel.rice@ynhh.org

Nathan Price
Digital & Technology Solutions
Yale New Haven Hospital
New Haven, CT, USA
nathan.price@yale.edu

Will Byron
Digital & Technology Solutions
Yale New Haven Hospital
New Haven, CT, USA
william.byron@yale.edu

Yan Hu
Mcwilliam School of Biomedical Informatics University of Texas Health Science at Houston
Houston, TX, USA
yan.hu@uth.tmc.edu

Donn Felker
Digital & Technology Solutions
Yale New Haven Hospital
New Haven, CT, USA
dfelker@agilevent.com

Will Button
Digital & Technology Solutions
Yale New Haven Hospital
New Haven, CT, USA
will@willbutton.co

Deniella Meeker
Section of Biomedical Informatics and Data Science School of Medicine
Yale University
New Haven, CT, USA
daniella.meeker@yale.edu

Allen Hsiao
Pediatrics and of Emergency Medicine at Yale School of Medicine
Yale University
New Haven, CT, USA
allen.hsiao@yale.edu

Hua Xu
Section of Biomedical Informatics and Data Science School of Medicine
Yale University
New Haven, CT, USA
hua.xu@yale.edu

Charles Torre Jr
Digital & Technology Solutions
Yale New Haven Hospital
New Haven, CT, USA
charles.torrejr@ynhh.org

Wade Schulz*
Department of Laboratory Medicine
School of Medicine
Yale University
New Haven, CT, USA
wade.schulz@yale.edu

***Abstract*— Electronic Health Records (EHRs) represent a crucial data source for real-world evidence generation. To facilitate biomedical studies using EHRs, standard data models like the OMOP CDM have been developed. Nevertheless, recent advancements in biomedical AI research that leverage EHRs have introduced new challenges, encompassing security considerations, large-scale data retrieval, and computational resource management, including GPUs. This paper introduces Kamino, an innovative architectural solution tailored to support biomedical AI research using EHR data. Kamino offers a user-friendly interface with features designed for efficient team access management in accordance with regulatory requirements. It facilitates direct data retrieval from an OMOP CDM instance and includes a resource allocation system based on Kubernetes orchestration. Here, we demonstrate the practical application and utility of Kamino through a clinical natural language processing task. We firmly believe that such a tool will significantly expedite AI research conducted with EHR data within academic institutions.**

Keywords—Health Informatics, Observational Medical Outcomes Partnership, Common Data Model, Data Governance, System Design

I. INTRODUCTION

In the era of Artificial Intelligence (AI) transformation, the use of computational platforms and large clinical data sets has become a pivotal avenue for groundbreaking discoveries and innovations in biomedicine [1]. The integration of large, real-world healthcare data (e.g., electronic health records – EHRs) into AI research tools holds the potential to unlock unprecedented insights into complex medical phenomena and enhance clinical care [2]. However, the integration of healthcare data into AI research architectures introduces a multifaceted set of challenges [3]. These challenges include not only the need for diverse and secure analysis environments, but also systems that multiple users can leverage for AI-driven research endeavors. As AI research progresses, the need for new and more advanced computational resources grows. The scalability of these resources is pivotal to accommodate the increasing data volume and computational demands of sophisticated AI workflow.[4] This study aims to design a computational health platform for AI research, called Kamino, with an

* Corresponding author

emphasis on seamless data retrieval and optimized resource utilization within a secure environment.

Kamino has three main features. Firstly, it has a team-based workspace that aligns with IRB compliance, enhancing secure and efficient interdisciplinary collaboration. Secondly, an automated ETL (Extract, Transform, Load) pipeline, that streamlines data extraction from the OMOP (Observational Medical Outcomes Partnership) CDM (Common Data Model). Lastly, the environment supports customizable computing configurations to meet the demands of AI research and real-world evidence (RWE) generation. These advancements collectively establish our platform for healthcare AI research, poised to adapt to the sector's evolving needs.

Despite the utility of tools like ATLAS [5] and the i2b2 Query & Analysis Tool [6] project request plugin in their fields, they fall short in meeting the unique requirements of AI research. These tools lack the necessary scalability, dynamic resource allocation, and customization needed to efficiently manage. Our proposed architecture addresses these shortcomings by offering a solution that is both scalable and adaptable, tailored specifically for AI research. It ensures adjustment of resources to meet project demands and provides customizable environments, enabling researchers to optimize their setups for specific tasks. This adaptability is crucial for enhancing the processing and exploration of large data sets, thereby accelerating innovation in AI research.

II. Task analysis and design requirements

A. Construction of a Team-Based Collaborative Workspace

A key requirement is to establish a collaborative workspace that is team-oriented with role-based access control. This workspace should facilitate seamless communication and cooperation among researchers and data analysts. It must support various roles and permissions, enabling team members to effectively share resources and collaborate within a study team.

B. Automated ETL Pipeline from Standard Data Repository

Efficient and standardized data retrieval is a critical step in medical AI research. The system should allow researchers to submit data requests, verify its compliance with IRB specifications, and upon verification, automatically execute an ETL pipeline to extract relevant data from EHR-derived research databases in a standard data model, such as the OMOP CDM [7]. This process involves extracting the requested data, transforming it to ensure compatibility with the research requirements, and loading it with appropriate permissions into the researcher's computing environment. This automation not only streamlines the data retrieval process, but also ensures compliance with ethical standards and data privacy regulations.

C. Customizable Computing Environments

Researchers should have the flexibility to choose their computing environment, tailored to their specific AI research computing needs. This includes the selection of hardware resources, such as CPUs, GPUs, and memory, as well as a base operating environment through a containerized image. The system must be capable of automatically scheduling and allocating these resources based on the researcher's selection and available capacity. This feature is critical to accommodate diverse research needs and computational demands in a multi-user environment, ensuring efficient and effective data analysis.

III. System Overview

The architecture we proposed is accompanied by a seven-step workflow process, as depicted in Figure 1. The sequence of operations within the workflows is outlined as follows: (1) An initial user action commences from the web application, which entails generating a new team request. This request is subsequently transmitted to a message queue. (2) The team request action is relayed to a task runner, which then initiates the creating of a distinct namespace and assigns a resource claim within a Kubernetes environment for the respective team. (3) Users are afforded the capability to submit a multitude of personalized data requests. These may encompass a cohort and data element query utilizing the OMOP CDM format. (4) Incoming data requests are temporally stored in the message queue. Subsequent processing by a task runner initiates an ETL pipeline, culminating in the storage of data within a high-speed NVMe storage array. (5) In a parallel fashion, the web application also handles requests for specific compute environment configurations. (6) Environment requests are queued within the message queue system. After this, the task runner facilitates the deployment of the containerized images and mounted file system containing preloaded user-requested data. This is provisioned with the required hardware resources via the Kubernetes resource manager. (7) Upon the readiness of the computational environment, the task runner aids in establishing a private endpoint through Kubernetes. This endpoint may integrate services such as interactive notebook and terminal, which are pivotal for the processing of data on a substantial scale with tools familiar to end users.

We structured our system into five primary layers, each contributing critical functions to the system's overall performance. In the following sections, we will highlight the functions inherent to each layer, demonstrating how they collectively establish a harmonious and efficient workflow.

A. Web Application

The Web Application layers the primary interface for user interaction with the system, containing the following functionality:

1) Team-Based Workspace: The workspace is architected around IRB application requirements. It ensures that the composition of team members and data access control adhere strictly to the approved protocol including security plans and data specificatons, establishing a secure and compliant environment for collaborative research.

2) Data Request: Users are empowered to define cohorts and conduct queries for specific data elements within the OMOP CDM as part of their data requests. Elasticsearch [8] is employed as a search engine for the OMOP query. This integration allows for swift, efficient, and precise searches, streamlining the data retrieval process and aiding users in design their data request [9].

3) Environment Configuration: The web application provides users with the capability to tailor their computational environment according to their project needs. Users can specify resource requirements, including CPU, GPU, and memory allocations, as well as designate specific container configurations. The Spark framework is also

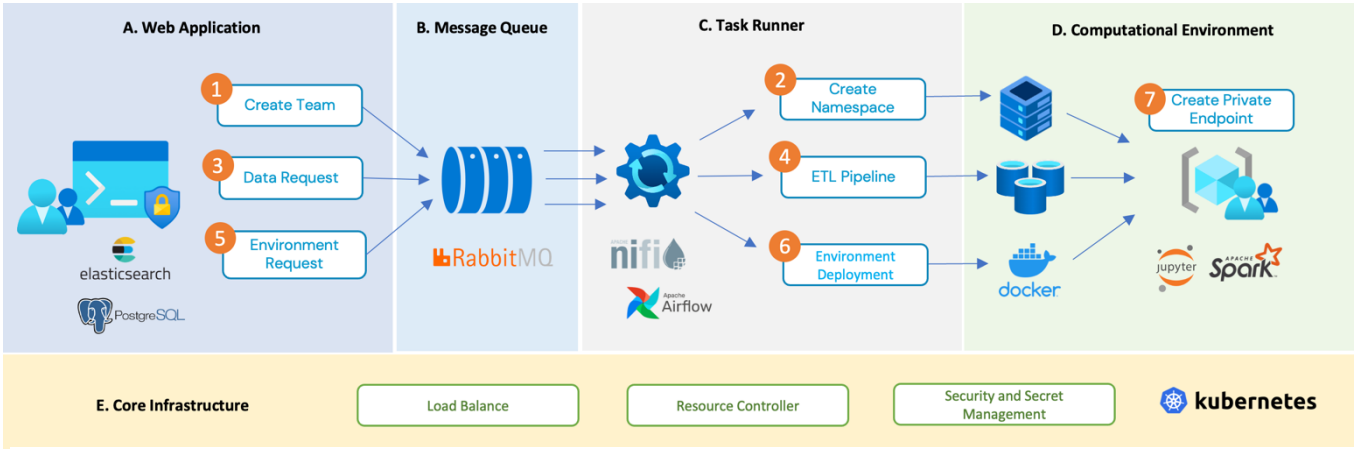

Figure 1. Overview of system architecture

included within the environment to allow for at-scale distributed compute. These personalized settings ensure that the computational environment is optimally configured for the user demand.

Each of these functionalities—team-based workspace requests, data requests, and environment configurations—are encapsulated into messages. These messages are then seamlessly transmitted to the subsequent Message Queue layer.

B. *Message Queue*

The Message Queue layer powered by RabbitMQ, which ensures that message delivery is reliable and scalable. This layer is vital for the system's asynchronous processing capabilities and allows for component decoupling, which enhances the scalability and resilience in a multi-user system. It handles the routing of messages to the Task Runner, which maintains the integrity and order of the message flow.

C. *Task Runner*

The Task Runner layer is responsible for executing tasks such as ETL processes, managing interactions with data resources, and deploying computational environments. It uses Apache NiFi [10] for data flow automation and Apache Airflow [11] or orchestrating complex computational workflows.

D. *Computational Environment*

The computing environment consist of containerized images, which ensures consistency and ease of deployment. Data sets are mounted to these containers to facilitate access and manipulation. The analytics engine is powered by Apache Spark, which provides a fast and general-purpose cluster-computing framework. Additional, validated tools and libraries can be installed within the user space to support the specific development needs of end users.

E. *Core Infrastructure*

At the base of the architecture is the Core Infrastructure layer, which provides foundational support across the other layers. Through the use of Kubernetes, the infrastructure includes a load balancer to distribute incoming traffic and ensure high availability, a resource controller to manage the allocation and deallocation of system resources dynamically, and the orchestration of containerized applications [12]. Security and secret management are also centralized in this layer, ensuring that all aspects of the system maintain integrity and confidentiality.

The above architecture offers a comprehensive and scalable solution tailored to support AI research with large, real-world data sets, and emphasizes scalable and dynamic resource allocation alongside customizable computing environments.

IV. A Practical Example of AI Research

To elucidate the practical application of our proposed architecture, we focus on a scenario to support a Natural Language Processing (NLP) task. For NLP tasks, leveraging data parallelism involves distributing large datasets across multiple GPUs to enhance batch processing capabilities [13]. We designed an experiment focused on Named Entity Recognition (NER) [14] with the LLAMA2-7b-chat model [15] that involves identifying and classifying key entities in text into predefined categories such as the names of disease entities, treatments from clinical note. Speed tests in NER tasks are crucial to evaluate the efficiency of NER systems, especially in time-sensitive applications. This example uses the metric of Tokens Per Minute (TPM) to quantify the speed of an NER system, providing insights into its performance and scalability.

We compiled below three cases for distributed data parallel using NCCL (NVIDIA Collective Communications Library) configurations.

1) Base Case: Using a single GPU, serving as the control setup to benchmark the improvements offered by parallel processing techniques.

2) Data Paralle with 4 GPUs: Leveraging four GPUs to implement data parallelism with NCCL, aiming to calcuate TPM by distributing the workload across multiple GPUs.

3) Data Parallel with 8 GPUs: Employing eight GPUs with the NCCL to facilitate efficient distributed data parallelism, potentially offering further reductions in processing times due to enhanced inter-GPU communication and synchronization.

To build this computational environment, we structured our architecture's workflow to support NLP research. (1) Initiation of Team Request: The procedure is initiated through a user-initiated action within the web interface, where a request for a new team is formulated. This request is then systematically transmitted to a message queue for subsequent processing. (2) Create Namespace: The Task Runner handles this request to communicate with Kubernetes for creating a namespace and applying maximum compute quotas for CPU, GPU, and memory. (3) Submit Data Request: The team

submits a request for retrieval of clinical notes in the OMOP CDM format. (4) ETL Pipeline: Upon receipt of the data request, the task runner initiates an automated data processing workflow, which extracts relevant information from the data lake. To ensure data integrity, the extracted data is permissioned as read-only. Additionally, read access is restricted to the team generating the request, thereby ensuring data privacy. The processed data is then loaded into a high-speed Network File System (NFS), making it accessible to the project's computational tasks. (5) Environment Request: A request is formulated for environmental configuration, selecting a container image equipped with CUDA (Compute Unified Device Architecture) version 12 or higher, which supports deep learning frameworks such as PyTorch. Additionally, the request includes the selection of eight NVIDIA A100 GPU cards, and other hardware requirements for the computational tasks. (6) Environment Deployment: Based on the environment configuration request, the task runner communicates with Kubernetes to deploy the computational environment, incorporating the specified container image and mounting the associated data from NFS. This step ensures the provision of a tailored computational environment conducive to the project's requirements. (7) Create Private Endpoint: Following a successful health check of the computational environment, the system establishes a private endpoint within the web application. This endpoint serves as a secure access point for the team.

The results are delineated below *Table 1*, across the three distinct experimental setups. The experiment demonstrated significant enhancements in NER task performance through data parallelism. Starting with a baseline of 1,740 TPM on a single GPU, the system's performance increased to 6,420 TPM with four GPUs—a 3.69-fold improvement. Expanding to eight GPUs further amplified the system's throughput to 12,360 TPM, marking a 7.10-fold increase from the baseline.

In this example, we have illustrated the scalable capabilities of Kamino as a computational platform designed to support advanced AI research. Moreover, the architectural workflow of Kamino, which encompasses team initiation, namespace creation, data retrieval, ETL pipeline processing, environment configuration, and deployment, further exemplifies the platform's comprehensive support for AI and NLP research. The seamless integration of Kubernetes, CUDA-enabled container images, and GPUs distributed across multiple devices within this architecture illustrates a sophisticated infrastructure that is both flexible and robust, capable of catering to the demanding requirements of AI research projects.

Table 1. Performance of NER Task Execution Across Different GPU Configurations

Experimental Setup	TPM	Speed Up
Base case with 1 GPU	1,740	1
Data Parallel with 4 GPUs	6,420	3.69
Data Parallel with 8 GPUs	12,360	7.10

V. Discussion

The deployment of the inaugural version of Kamino within Yale School of Medicine and (YSM) Yale New Haven Health System (YNHHS) signifies a pivotal advancement in the integration of technology and healthcare. This platform not only serves as a cornerstone for our current operations but also paves the way for future innovations in healthcare data management and analysis. Through this framework, we have successfully supported the use of a real-world data repository, which has enhanced our ability to store, process, and analyze vast amounts of healthcare data.

A noteworthy achievement in our endeavor has been to leverage the standardized OMOP CDM. YNHHS' OMOP CDM has accumulated over 12 billion records and is instrumental in facilitating comprehensive research studies, enabling advanced analytics, and supporting decision-making processes in healthcare. The OMOP CDM is a widely used data model that has been adopted by hundreds of healthcare systems, indicating our architecture can be readily applied to many other medical centers, with the potential to enable scalable and interoperable studies cross sites.

Moreover, our platform can provide the computational scale needed to provide daily updates from the EHR, augmented with real-time data from our middleware systems and patient monitoring devices, which has generated over 700 million data points. This real-time capability is crucial for AI and analytics tasks such as risk prediction, treatment recommendations, and biospecimen identification [16][17]. The stability and reliability of our service in managing such a significant amount of data daily underscores the technical prowess and the advanced architecture of our computational healthcare platform.

The platform has successfully supported a variety of biomedical AI and real-world evidence generation studies, including the assessment of next-generation phenotypes [18], COVID-19 outcomes research and AI model development [17], [19], and real-world evidence generation for biomedical devices [20] and blood products [21].

This study demonstrates the integration of multiple layers into an orchestrated platform, using Kubernetes to provide load balancing, high availability, and scalability. The computational environments are secure and isolated, enhancing data integrity and privacy. This approach marks an advancement in healthcare informatics, offering a robust, scalable, and secure infrastructure for computational healthcare platform.

VI. Conclusion and Further works

The Kamino platform developed at YSM/YNHHS enhances biomedical AI research through various features, including collaborative workspaces, data analysis, and customizable environments. It adheres to technical and ethical standards, offering the flexibility to meet the evolving demands of healthcare research. Future development will emphasize the diversification of computational environments, targeting specific applications like end-to-end machine learning and LLMOps (Large Language Model Operations) [22].

We are also expanding our real-world data integration to include diverse data sets, focusing on medical imaging and genomics. This will improve diagnostic and predictive capabilities to enable personalized medicine through enhanced healthcare AI research.

As the system evolves, it will be crucial to continually assess and adapt these components to meet emerging challenges. This ongoing evolution will ensure that the system remains at the forefront of healthcare research and data analysis.

References

[1] "Evolution of Artificial Intelligence-Powered Technologies in Biomedical Research and Healthcare | SpringerLink." Accessed: Jan. 30, 2024. [Online]. Available: https://link.springer.com/chapter/10.1007/10_2021_189

[2] Z. Ahmed, K. Mohamed, S. Zeeshan, and X. Dong, "Artificial intelligence with multi-functional machine learning platform development for better healthcare and precision medicine," *Database*, vol. 2020, p. baaa010, Jan. 2020, doi: 10.1093/database/baaa010.

[3] "Federated learning-based AI approaches in smart healthcare: concepts, taxonomies, challenges and open issues | Cluster Computing." Accessed: Jan. 30, 2024. [Online]. Available: https://link.springer.com/article/10.1007/s10586-022-03658-4

[4] J. Ejarque *et al.*, "Enabling dynamic and intelligent workflows for HPC, data analytics, and AI convergence," *Future Gener. Comput. Syst.*, vol. 134, pp. 414–429, Sep. 2022, doi: 10.1016/j.future.2022.04.014.

[5] "ATLAS: Home." Accessed: Feb. 06, 2024. [Online]. Available: https://atlas-demo.ohdsi.org/#/home

[6] "i2b2 Web Client." Accessed: Feb. 06, 2024. [Online]. Available: https://www.i2b2.org/webclient/

[7] G. Hripcsak *et al.*, "Observational Health Data Sciences and Informatics (OHDSI): Opportunities for Observational Researchers," *Stud. Health Technol. Inform.*, vol. 216, pp. 574–578, 2015.

[8] "Elasticsearch: The Official Distributed Search & Analytics Engine," Elastic. Accessed: Feb. 06, 2024. [Online]. Available: https://www.elastic.co/elasticsearch

[9] S. Liu *et al.*, "Implementation of a Cohort Retrieval System for Clinical Data Repositories Using the Observational Medical Outcomes Partnership Common Data Model: Proof-of-Concept System Validation," *JMIR Med. Inform.*, vol. 8, no. 10, p. e17376, Oct. 2020, doi: 10.2196/17376.

[10] "Apache NiFi," Apache NiFi. Accessed: Feb. 06, 2024. [Online]. Available: https://nifi.apache.org/

[11] "Home," Apache Airflow. Accessed: Feb. 06, 2024. [Online]. Available: https://airflow.apache.org/

[12] E. Kim, K. Lee, and C. Yoo, "On the Resource Management of Kubernetes," in *2021 International Conference on Information Networking (ICOIN)*, Jan. 2021, pp. 154–158. doi: 10.1109/ICOIN50884.2021.9333977.

[13] S. Li *et al.*, "PyTorch Distributed: Experiences on Accelerating Data Parallel Training." arXiv, Jun. 28, 2020. Accessed: Jan. 31, 2024. [Online]. Available: http://arxiv.org/abs/2006.15704

[14] "A Survey on Deep Learning for Named Entity Recognition | IEEE Journals & Magazine | IEEE Xplore." Accessed: Feb. 01, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9039685

[15] H. Touvron *et al.*, "Llama 2: Open Foundation and Fine-Tuned Chat Models." arXiv, Jul. 19, 2023. doi: 10.48550/arXiv.2307.09288.

[16] T. J. S. Durant, G. Gong, N. Price, and W. L. Schulz, "Bridging the Collaboration Gap: Real-time Identification of Clinical Specimens for Biomedical Research," *J. Pathol. Inform.*, vol. 11, p. 14, 2020, doi: 10.4103/jpi.jpi_15_20.

[17] A. D. Haimovich *et al.*, "Development and Validation of the Quick COVID-19 Severity Index: A Prognostic Tool for Early Clinical Decompensation," *Ann. Emerg. Med.*, vol. 76, no. 4, pp. 442–453, Oct. 2020, doi: 10.1016/j.annemergmed.2020.07.022.

[18] "Temporal relationship of computed and structured diagnoses in electronic health record data | BMC Medical Informatics and Decision Making | Full Text." Accessed: Feb. 06, 2024. [Online]. Available: https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-021-01416-x

[19] J. McPadden *et al.*, "Clinical characteristics and outcomes for 7,995 patients with SARS-CoV-2 infection," *PLOS ONE*, vol. 16, no. 3, p. e0243291, Mar. 2021, doi: 10.1371/journal.pone.0243291.

[20] G. Jiang *et al.*, "Feasibility of capturing real-world data from health information technology systems at multiple centers to assess cardiac ablation device outcomes: A fit-for-purpose informatics analysis report," *J. Am. Med. Inform. Assoc.*, vol. 28, no. 10, pp. 2241–2250, Oct. 2021, doi: 10.1093/jamia/ocab117.

[21] W. L. Schulz *et al.*, "Blood Utilization and Transfusion Reactions in Pediatric Patients Transfused with Conventional or Pathogen Reduced Platelets," *J. Pediatr.*, vol. 209, pp. 220–225, Jun. 2019, doi: 10.1016/j.jpeds.2019.01.046.

[22] "LLMs for Enterprise and LLMOps | SpringerLink." Accessed: Feb. 01, 2024. [Online]. Available: https://link.springer.com/chapter/10.1007/978-1-4842-9994-4_7