contact@rhinohealth.com
22 Boston Wharf Rd., 7th Fl
Boston, MA 02210
+1 (857) 205-7433

Rhino Federated Computing Platform (Rhino FCP): Activating the World's Data through Privacy Preserving Artificial Intelligence and Data Analytics on the Edge

June 2024

Rhino FCP Overview White Paper

contact@rhinohealth.com
22 Boston Wharf Rd., 7th Fl
Boston, MA 02210
+1 (857) 205-7433

1. Introduction

In today's rapidly evolving technological landscape, industries across sectors increasingly rely on Artificial Intelligence (AI) and advanced analytics to drive innovation and efficiency. However, this reliance brings a significant challenge: balancing robust data access with stringent data privacy protection. Traditional data centralization approaches often compromise data security and privacy, making adopting AI and analytics solutions difficult, especially in healthcare, life sciences (biopharma), medtech, financial services, manufacturing, and public sectors.

The need for secure, scalable data computation without exposing sensitive information is more critical than ever. Federated Learning (FL) and Edge Computing technologies offer a promising solution to this problem by enabling decentralized data processing. This approach maintains data privacy by keeping data within its local environment while allowing collaborative analytics and AI model training across multiple sites.

These challenges are addressed in the Rhino Federated Computing Platform (Rhino FCP), which leverages FL and Edge Computing to provide a secure and scalable framework for data computation. This platform simplifies the orchestration of customized workflows without centralizing data, thus reducing barriers to AI adoption across various industries. Rhino FCP allows users globally to integrate data discovery, harmonization, AI model training, deployment, and statistical analysis seamlessly—without intricate collaboration agreements or data transfers. All computation occurs on local IT infrastructure that complies with regional Data Privacy and IT Security standards, ensuring robust data protection while facilitating advanced analytics.

2. Federated Computing Fundamentals

The term "federated" refers to a collaborative structure where multiple entities contribute resources or capabilities while maintaining some level of autonomy. In computing, **Federated Computing** enables computations on **decentralized data**, meaning the data resides on

contact@rhinohealth.com
22 Boston Wharf Rd., 7th Fl
Boston, MA 02210
+1 (857) 205-7433

individual devices or servers, (referred to in this context as **Rhino Clients**) rather than a central location.

Federated Learning:

Federated Learning (FL), introduced by Google researchers in 2016[1], trains machine learning models on decentralized data across multiple nodes without exchanging raw data samples, thus safeguarding privacy and enhancing model robustness through diverse data inputs. This technique is particularly advantageous in healthcare and financial services, where data confidentiality is critical. For instance, FL facilitated the development of predictive models for COVID-19 patient outcomes using data from various healthcare providers while maintaining strict data privacy, as demonstrated in a landmark publication by Rhino Health's co-founder and CEO, Dr. Ittai Dayan, (Dayan, I. et al., 2021[2]).

FL allows machine learning models to be trained on decentralized data while preserving privacy. The two main paradigms of FL are Horizontal Federated Learning (HFL) and Vertical Federated Learning (VFL).

- **Horizontal Federated Learning (HFL):** HFL, or Sample-based FL, involves participants with datasets with similar features but different samples. This allows for collaborative model training without sharing raw data. In the healthcare sector, multiple hospitals can train a disease diagnosis model using their patient data without sharing actual patient records. HFL is particularly advantageous for its ability to enhance model robustness through diverse data inputs while maintaining data privacy.[3]
- **Vertical Federated Learning (VFL):** VFL, or Feature-based FL, occurs when participants have datasets with the same samples but different features. This enables a more comprehensive model by combining different attributes while maintaining data privacy. In the healthcare sector, a hospital and genomic research institute can jointly train a model on the same patients without directly sharing data, leveraging diverse

[1] McMahan, H. B., Moore, E., Ramage, D., et. al. (2017). Communication-efficient learning of deep networks from decentralized data. I*n Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)* (pp. 1273-1282). PMLR.

[2] Dayan, I., Roth, H. R., Zhong, A., et al. (2021). Federated Learning for Predicting Clinical Outcomes in Patients with COVID-19. *Nature Medicine*. Available at: https://www.nature.com/articles/s41591-021-01506-3.

[3] Yang, Q., Liu, Y., Tong, Y. et. al. (2019). Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2), pp.1-10. https://dl.acm.org/doi/10.1145/3298981.

perspectives for improved accuracy. VFL is beneficial for combining heterogeneous data sources to develop more accurate models while preserving data confidentiality.[4]

Edge Computing:

Edge Computing[5] represents a significant shift in data processing, driven by the proliferation of Internet of Things (IoT) devices. This paradigm decentralizes data processing by moving it closer to where data is generated, enhancing performance by reducing latency and bandwidth demands on central servers. Critical for real-time applications, Edge Computing enables local data processing and storage, providing faster insights and improved response times. In Rhino FCP, Edge Computing is vital for executing code on local data, aligning seamlessly with the Federated Computing model to ensure efficient, secure, and privacy-preserving operations.

Rhino FCP's Federated Computing approach offers several advantages:

- **Enhanced Data Privacy and Security:** Rhino FCP minimizes the risk of data breaches and unauthorized access by keeping data on Rhino Clients for processing purposes only. This local data storage supports privacy and security, reducing the potential for data compromise.
- **Cost Efficiency and Resource Optimization:** By processing data locally, edge computing significantly reduces the need to transmit large amounts of data, lowering data transfer expenses. Additionally, this approach avoids duplicate storage, further optimizing resource utilization and cost.
- **Improved Scalability:** Rhino FCP's architecture allows for easy integration of new Rhino Clients and handles increasing data volumes efficiently without requiring significant infrastructure changes.
- **Reduced Latency:** Processing data locally on Rhino Clients minimizes latency compared to scenarios where data needs to be sent to a central location for processing.

[4] Liu, Y., Kang, Y., Zou, T., et al. (2022). Vertical Federated Learning: Concepts, Advances and Challenges. *Cornell University*. https://arxiv.org/abs/2211.12814.

[5] Brecko, A., Kajati, E., Koziorek, J., et al., (2022). Federated Learning for Edge Computing: A Survey. *Applied Sciences*. 12(8). https://www.mdpi.com/2076-3417/12/18/9124.

contact@rhinohealth.com
22 Boston Wharf Rd., 7th Fl
Boston, MA 02210
+1 (857) 205-7433

- **Speed:** Faster processing and response times due to proximity to data sources.

3. Rhino FCP: Architecture and Capabilities

3.1. Platform Architecture and Interfaces

Rhino FCP consists of two main components:

1. **Rhino Client:** Software installed on a virtual machine or physical server behind the data custodian's firewall, whether on-prem in a data center or on a Virtual Private Cloud (VPC). The Rhino Client can securely access local datasets and has access to GPUs/CPUs in order to securely run computational workloads (e.g., AI-model training) without moving sensitive data outside the firewall.
2. **Rhino Cloud:** Software that orchestrates tasks across Rhino Clients (e.g., FL), and is the access point for all user interactions with Rhino FCP. It is hosted on AWS. Patient data is not transferred to the Rhino Cloud, thus enabling collaborations with data custodians who are restrictive in their data-sharing requirements.

The Rhino Cloud exposes three types of interfaces to users:

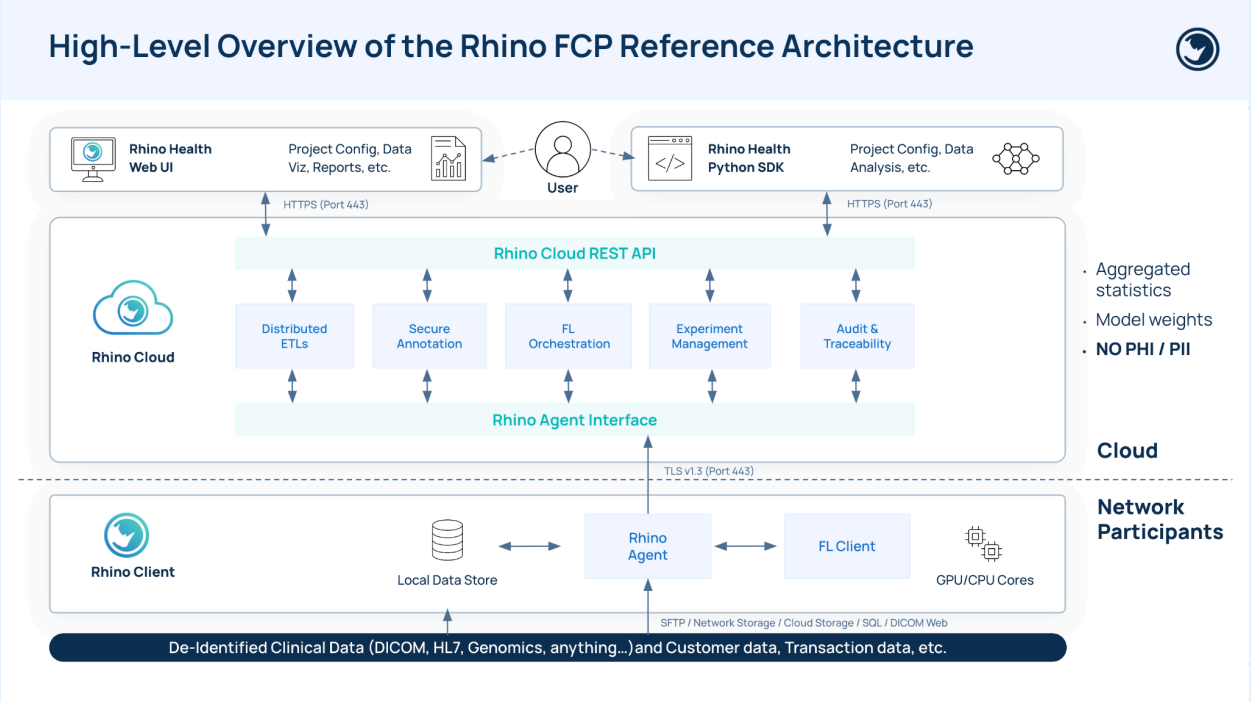
1. **Web UI:** A graphical web-based user interface (GUI) accessible via browsers at https://dashboard.rhinohealth.com.
2. **Python SDK:** A Python library accessible via browsers at https://pypi.org/project/rhino-health/ for interacting with Rhino FCP from Jupyter Notebooks or any other Python-based system.
3. **REST API:** An HTTPS-based REST API that interacts with Rhino FCP. Both the Web UI and the Python SDK use this REST API under the hood.

The approach is "centralized control with decentralized execution," allowing users to manage federated AI projects using cloud interfaces to oversee compute processes run at the edge. Rhino FCP manages the interactions between the Rhino Clients and Rhino Cloud so that the data on the Rhino Clients never leaves the data custodian's network. This is accomplished via a

secure interface between the Rhino Clients and the Rhino Cloud that only supports specific actions, not allowing patient data to be transferred from the Rhino Client.

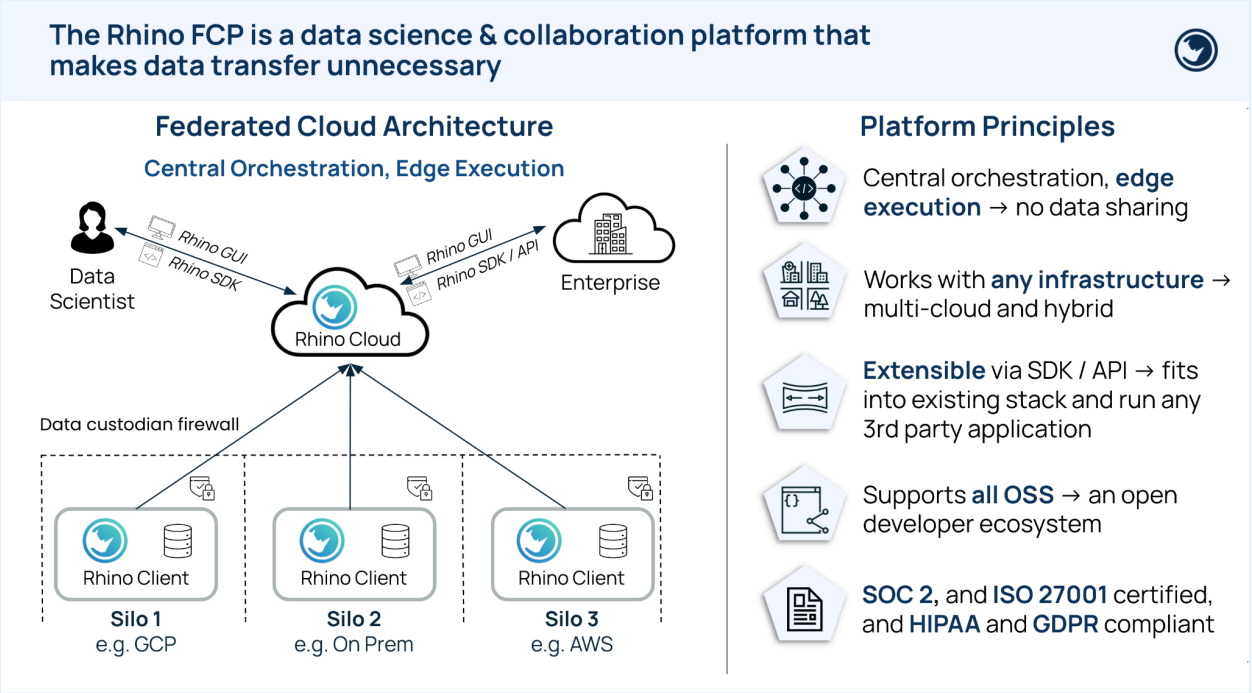
Rhino FCP is HIPAA and GDPR-compliant and is ISO 27001 and SOC 2 Type II certified. Additional information regarding Rhino FCP's different security features is outside the scope of this white paper, and can be provided on a case-by-case basis.

3.1.1. Rhino FCP Reference Architecture



Rhino FCP consists of four secure layers. Users interact via the Rhino Web UI, Python SDK, or API. The Rhino Cloud environment, accessed through HTTPS, manages components like distributed ETLs, secure annotation, and FL orchestration without transferring row-level data. The Rhino Client, operating within the data custodian's network, handles storage and computing, communicating securely with the Cloud. This architecture ensures row-level data remains within the data custodian's network, enabling secure collaboration for model training and privacy-preserving analysis, with encryption at rest, in transit, and during processing.

3.1.2. Federated Cloud Networks



Rhino FCP leverages a "Central Orchestration, Edge Execution" model to create a Federated Cloud Network. This network consists of interconnected yet autonomous Rhino Clients operating across various environments, including on-premises data centers, virtual private clouds (VPCs), and public cloud services. Rhino FCP enables secure, efficient, and scalable data collaboration across multiple sites without transferring sensitive data outside local environments by federating these clients.

Rhino FCP is a comprehensive data science and collaboration platform that eliminates the need for data transfer. It employs a "Central Orchestration, Edge Execution" model, where users interact with the Rhino Cloud through the Rhino GUI and Python SDK. At the same time, enterprises connect via the Rhino Cloud GUI and SDK/API. This setup supports multi-cloud and hybrid environments, is extensible via SDK/API, and is compatible with all open-source software, fitting seamlessly into existing software stacks.

contact@rhinohealth.com
22 Boston Wharf Rd., 7th Fl
Boston, MA 02210
+1 (857) 205-7433

3.2. Data Ingestion and ePHI / PII Processing

The Rhino Client can ingest data from external sources for processing. Rhino FCP is data agnostic, meaning all data types and formats are supported, including structured data (e.g., from EHR), textual data (e.g., from clinical notes), imaging data, videos, genomic data, and any other data type or formats. Data can be made accessible to the Rhino Client directly from transaction data or data lake/data warehousing systems (e.g., via a DICOM interface or from a database or data warehouse) or by providing access to data that was extracted by a local team member at the site (e.g., as files that are made accessible to the Rhino Client via network file storage).

The site can de-identify data before making it accessible to Rhino FCP. Rhino FCP can also work with electronic Protected Health Information (ePHI) and electronic Personally Identifiable Information (ePII). The data can be made accessible to the Rhino Client, where de-identification or data masking can create a de-identified copy or obscure data for processing, ensuring privacy through k-anonymity. This unlocks new use cases, such as working with pseudonymized data—data modified to remove direct identifiers but can still be linked back to individuals by authorized personnel, often called data custodians or stewards. For example, a third-party application could identify patients needing intervention and notify the hospital without transferring electronic Protected Health Information (ePHI) or de-identified ePHI. This ensures that all processing occurs at the edge, maintaining strict data privacy.

3.3. Federated Computing Platform - Core Capabilities

Rhino FCP supports federated projects flexibly, allowing users to employ their preferred tools or frameworks.

The following sections explain some of the platform capabilities that make federated projects a reality:

3.3.1. Project and Permissions Management

Project Management: On Rhino FCP, users can define and manage distributed projects. Collaborators can be invited with specific role-based permissions, allowing multi-site and

contact@rhinohealth.com
22 Boston Wharf Rd., 7th Fl
Boston, MA 02210
+1 (857) 205-7433

single-site project management. The platform provides a unified view of projects, enhancing collaboration and efficiency.

Role-Based Access Control (RBAC): Rhino FCP supports fine-grained permissions based on user roles, enabling diverse collaboration models. This flexibility ranges from fully collaborative projects with visible data and code access to projects where only specific roles can access code and results.

Audit Log: A comprehensive audit log captures all significant actions within a project, providing traceability and accountability. Users can view and export this log for detailed project analysis.

3.3.2. Data Management

Data Sources: Rhino FCP handles various data sources seamlessly, ensuring efficient and secure management across distributed environments. The platform supports multiple data input methods, accommodating diverse infrastructures and facilitating comprehensive data integration for FL and analytics.

Supported Data Sources:

- **Cloud Storage:** The platform integrates with cloud storage systems such as AWS S3 and Google Cloud Storage, enabling the handling of large unstructured datasets like images, videos, and documents.
- **Network Storage:** The platform can access data in network-attached storage (NAS) systems or network shares like SMB, benefiting organizations with existing network storage infrastructure and ensuring easy integration without significant changes.
- **Secure File Transfer Protocol (SFTP):** Rhino FCP supports SFTP for securely transferring files, allowing users to upload data files directly to the platform. This is particularly useful for securely transferring bulk data from remote locations.
- **SQL Databases:** Rhino FCP connects to SQL databases, allowing users to query structured data directly.

contact@rhinohealth.com
22 Boston Wharf Rd., 7th Fl
Boston, MA 02210
+1 (857) 205-7433

Data Ingestion and Validation: The platform can verify schema compliance with predefined data models upon ingestion. This step helps maintain data integrity and compatibility with FL algorithms deployed on the platform. Users can validate new datasets against predefined schemas, ensuring compliance with code specifications and reducing the need for debugging downstream.

Data Segregation, Versioning, and Auditability: Data within Rhino FCP is partitioned by project, with strict access control based on project-specific permissions. Each dataset action is logged to maintain a comprehensive version history, facilitating auditability and traceability. This ensures that data usage permissions are controlled by project settings, governing individual access to specific data.

Privacy-Preserving Data Management: Rhino FCP emphasizes privacy preservation by ensuring that sensitive data never leaves the local environment. Data is always encrypted at rest and in transit, with additional measures like differential privacy and k-anonymization applied to protect individual data points during aggregation and analysis.

Rhino FCP provides a flexible and secure environment for collaborative FL and data analytics by supporting various data sources and implementing robust data management practices. This ensures organizations can use their data assets while maintaining strict privacy and security standards.

3.3.3. Running Any Code at the Edge

Containerized Code: Securely run any general-purpose computational code (in any programming language or framework with any 3rd party libraries) on distributed data with maximum flexibility by simply importing the user-created code into a secure container registry for subsequent execution on the edge. Common use cases include data preprocessing, filtering, batch inference, and federated queries. The absence of network access and granular permissions management have all been put in place to ensure that containerized code can be run securely. Sites can also select to assess the contents of any container before granting permission to execute that container at their site.

Python Code: Users can simply provide their Python code, including third-party libraries, and build customized containers. Rhino FCP will automatically build a container image from this code so that users don't need to learn how to work with Docker containers.

contact@rhinohealth.com
22 Boston Wharf Rd., 7th Fl
Boston, MA 02210
+1 (857) 205-7433

Data Preprocessing and Harmonization Pipelines: Build your code/ETL pipeline step-by-step and easily connect those steps into a single repeatable pipeline from ingestion to reporting while using multiple user-defined data schemas to control data quality and drive interoperability across the network.

Code Debugging: View live or historical logs from your remote code runs and halt your code run if it isn't performing the right actions.

Federated Learning: Rhino FCP implementation of Federated Learning (FL) is designed to provide unmatched privacy, scalability, and efficiency in distributed machine learning. The key advantages of this approach are highlighted below:

- **Privacy Preservation:** The FL implementation ensures that sensitive data never leaves the local environment. By sharing only model parameters and not raw data, the risk of data breaches is minimized, complying with stringent data protection regulations.
- **Scalable Collaboration:** Secure, scalable collaboration across institutions is enabled. Multiple organizations can participate in training and refining machine learning models without exposing their proprietary or sensitive data.
- **Enhanced Model Robustness:** Utilizing the diverse data from various nodes enhances the robustness and generalizability of the models. This diversity helps in developing more accurate and reliable models.
- **Efficient Knowledge Sharing:** Rhino FCP supports rapid model iteration and refinement through efficient parameter exchanges and comprehensive visualizations. Tools like TensorBoard are integrated to facilitate real-time monitoring and adjustments.
- **Regulatory Compliance and Security:** Rhino FCP is designed to comply with all major data privacy laws, including HIPAA and GDPR. Additional security measures, such as differential privacy and k-anonymization, are integrated to protect data integrity.

Rhino FCP's FL implementation streamlines the entire process, from data harmonization to model training and validation, ensuring that organizations can leverage their data assets while maintaining the highest privacy and security standards.

contact@rhinohealth.com
22 Boston Wharf Rd., 7th Fl
Boston, MA 02210
+1 (857) 205-7433

Interactive Containers: Interactive Containers facilitate the deployment and operation of any Linux-based containerized application in an interactive session. This is particularly beneficial for use cases that require real-time interaction with data, such as visualization, annotation, or interactive data analysis. By enabling users to connect to the UI of these applications remotely, Rhino FCP provides a robust and flexible environment for users to perform their tasks efficiently and securely.

The key features of Interactive Containers are:

- **Direct Data Interaction:** Interactive Containers allow users to connect to applications running at the edge, akin to having a remote desktop session. This enables direct interaction with the data, making it easier to visualize, analyze, and annotate datasets directly within the secure confines of the local environment.
- **Versatile Application Support:** Rhino FCP supports various applications, from custom data visualization and annotation tools (e.g., 3D Slicer) to interactive data analysis environments like Jupyter Notebooks. This flexibility ensures users can use their preferred tools and workflows without significant modifications.
- **Secure and Compliant:** All interactions within Interactive Containers are secure, with robust encryption ensuring that data remains protected. The containers operate within the local Rhino Client, ensuring that sensitive data does not leave the secure environment. This approach aligns with stringent data privacy and security standards, such as differential privacy and k-anonymization.
- **Scalable and Extensible:** Rhino FCP is designed to scale, supporting the deployment of multiple Interactive Containers across various edge locations. This scalability ensures that organizations can manage large-scale FL projects efficiently. Additionally, the platform's integration with container orchestration tools allows for easy extension and customization of the interactive environments.

Interactive Containers can be used in a variety of cases:

- **Data Annotation Projects:** Researchers can use Interactive Containers to deploy annotation tools that enable remote teams to annotate and label data in real-time, enhancing the quality and accuracy of the datasets used for model training.

contact@rhinohealth.com
22 Boston Wharf Rd., 7th Fl
Boston, MA 02210
+1 (857) 205-7433

- **Interactive Data Analysis:** Users can use Jupyter Notebooks within Interactive Containers to perform exploratory data analysis, develop and test machine learning models, and visualize results, all while ensuring data compliance.
- **Custom Visualization Tools:** Rhino FCP supports deploying custom visualization tools. These tools allow users to interact with complex datasets visually and intuitively, facilitating better insights and decision-making.

3.3.4. Applications and Use Cases

Harmonization Copilot

Most data science and AI development require data harmonization, which involves transforming data from a source format (e.g., a hospital's custom data structure) to a target schema (e.g., OMOP or FHIR). Rhino FCP offers the Harmonization Copilot application, which leverages cutting-edge Generative AI capabilities and Federated Learning (FL) to streamline this transformation. The Harmonization Copilot application simplifies and accelerates data harmonization by standardizing clinical data to fit Common Data Models.

This application assists data teams by reducing the time required for data harmonization projects from months to days.It uses a human-in-the-loop workflow that improves with usage—locally and across the network. Importantly, it maintains patient data privacy by keeping data local at each site and providing state-of-the-art tools that operate at the edge without using Public APIs.

The Harmonization Copilot ensures robust data lineage by preserving the raw data and tracking each transformation step. This approach guarantees the accuracy and reliability of the harmonized data, providing a solid foundation for high-quality analytics pipelines and AI models.`

Federated Datasets

Performing feasibility studies or simply searching for sites that have the right data for a project can be time-consuming and expensive. Rhino FCP allows data custodians to create Federated Datasets, selecting a specific dataset and making it (or parts of it) explorable by others via

contact@rhinohealth.com
22 Boston Wharf Rd., 7th Fl
Boston, MA 02210
+1 (857) 205-7433

Rhino FCP, all while abiding by strict privacy constraints, e.g., k-anonymization, differential privacy, Role-Based Access Control (RBAC), and more.

Users who want to find the right data for their projects or select the right sites for their study can browse the different Federated Datasets and explore aggregated statistics of sub-groups and filtered subsets of different datasets. Throughout this time, the underlying patient data is kept local at each site - only aggregated statistics are shared with researchers. Once they have identified a sub-group (from a set of sites) that meets their needs, the data custodians can review and approve access. Then, the researcher can begin a Federated Computing project with the selected dataset sub-group, including appending new fields to the dataset as the Federated Datasets maintain data lineage

Remote Data Viewers and Annotation

Zero-footprint Access to Authorized Data: The zero-footprint visualization technology enables providing and receiving permissions to view specific data points (e.g., tabular data, imaging studies) in a dynamic and audited way while enforcing that data never persists outside its local environment (i.e., the local Rhino Client). This capability allows users to:

- Review their own data within the platform, including a secure in-platform image viewer, in an integrated and auditable way while using all the tooling on Rhino FCP.
- Ascertain quality, annotate, and add feedback to remote data (i.e., other Rhino Clients), enabling a wide range of remote annotation projects with any custom annotation tool.
- Provide collaborators temporary access to specified data points for similar purposes (i.e., access to specific data on your Rhino Client).

Integrated Viewers: Rhino FCP has a fully integrated tabular data viewer, textual log viewer, image viewer (e.g., .png, .jpg), and DICOM viewer.

Custom Viewers and Annotation Tools: Besides the integrated viewers mentioned above, Rhino FCP supports running any Linux-based containerized application remotely and connecting to its UI in an interactive session via Interactive Containers (see above). This capability is like having a remote desktop to a container running on the Rhino Client.

Federated Training and Validation

contact@rhinohealth.com
22 Boston Wharf Rd., 7th Fl
Boston, MA 02210
+1 (857) 205-7433

Training and Validation of Any Model Architecture: Rhino FCP provides a robust end-to-end platform for users to get up and running with FL quickly. Users can train their FL models with Rhino FCP using NVIDIA's FLARE SDK, whether the code is based on TensorFlow, PyTorch, Scikit-learn or other frameworks. NVIDIA maintains NVIDIA FLARE, which is continuously assessed and pressure-tested from a security perspective. Rhino FCP's integration with FLARE is updated with the latest features, making running, training, and validating of federated models streamlined and automated.

Federated TensorBoard: Use TensorBoard to see real-time results from model runs, halt model runs that aren't converging, and compare model runs across different experiments. All this while preserving the TensorBoard logs at the edge.

Experiment Management: With Rhino FCP, you can run multiple experiments with different model versions, perform hyper-parameter tuning, store multiple checkpoints/params, halt model runs, or run federated inference using different model params and different validation datasets at different sites.

Federated Statistical Methods and Analytics

Rhino FCP includes a large set of privacy-preserving federated metrics, including Basic Statistics (Mean, Standard deviation, Median, and Percentiles), Common Statistical Methods [t-test, ANOVA (Analysis of Variance), Chi-square test, Correlation analysis (e.g., Pearson, Spearman), Regression analysis (e.g., linear regression, logistic regression)], and Epidemiology Metrics [Risk (e.g., relative risk, odds ratio), Prevalence, Incidence, Kaplan-Meier survival analysis, Cox proportional hazards (PH) model]. Only aggregate data are shared, and a privacy filter is applied to ensure that data extracted via these metrics cannot be used for re-identification, including k-anonymization, differential privacy, and field-level permissions.

Rhino FCP also offers a set of federated AI model evaluation metrics that allow assessing model performance in a privacy-preserving manner such as Classification Metrics [ROC (Receiver Operating Characteristic) curve and AUC (Area Under the Curve)[6],[7], Precision, Recall, F1 score]; Ranking Metrics [FROC (Free-response Receiver Operating Characteristic)

[6] Soltani, B., Zhou, Y., Haghighi, V., et al. (2023). A Survey of Federated Evaluation in Federated Learning. *Cornell University*. Available at: https://arxiv.org/abs/2305.08070 (Accessed 31 May 2024).

[7] Flower. Flower Framework: Federated evaluation. Available at: https://flower.ai/docs/framework/explanation-federated-evaluation.html (Accessed 31 May 2024).

contact@rhinohealth.com
22 Boston Wharf Rd., 7th Fl
Boston, MA 02210
+1 (857) 205-7433

curve[4], Average precision]; Regression Metrics (Mean squared error, Mean absolute error). These metrics can be computed locally using each Rhino Client's private data. Then, the individual metric values can be aggregated across Rhino Clients using secure aggregation protocols like federated analytics or multi-party computation[4,5]. This allows for the evaluation of the global model performance without directly sharing raw Rhino Client data. Additionally, subgroup analysis (e.g., by demographics) can help assess model fairness and identify potential biases in a privacy-preserving way[8]. Evaluation metrics are computed locally and only the aggregated values are shared, providing a privacy-preserving way to benchmark the use of the AI model.

Additional metrics beyond those already implemented in the Rhino FCP can be added via Rhino FCP core capabilities (e.g., Python Code, Containerized Code, or Interactive Containers).

Create custom visualizations using any preferred library (e.g., Matplotlib, Seaborn, D3.js, or any other visualization library), produce reports, and share them through the web UI.

3.4. Integrations and Platform Extensibility

Extend Rhino FCP using the Rhino SDK and REST API—integrate third-party tools (e.g., experiment management systems, visualization, and BI tools), automate your distributed AI workflows (e.g., reusable ETLs that function as code objects within Rhino FCP, including templated ETLs from Harmonization Copilot), and integrate Rhino FCP into existing AI tools and workflows.

3.5. Security and Privacy

Rhino FCP employs robust security measures, including data encryption at rest and in transit, role-based access control, and comprehensive audit logs.

Check the Trust Center for compliance with stringent data privacy and security standards, including HIPAA, GDPR, ISO 27001, and SOC 2 Type II.

[8] Divi, S., Lin, Yi-S., Farrukh, H., et al. (2021). New Metrics to Evaluate the Performance and Fairness of Personalized Federated Learning. *Cornell University*. Available at: https://arxiv.org/abs/2107.13173 (Accessed 31 May 2024).

contact@rhinohealth.com
22 Boston Wharf Rd., 7th Fl
Boston, MA 02210
+1 (857) 205-7433

4. Implementation and Deployment

Implementing Rhino FCP involves a streamlined hardware provisioning process, software installation, and security configuration. Rhino FCP supports network participant-managed and Rhino-managed deployments, offering flexibility and scalability. Deployment typically takes 2-4 weeks, with most of the timeline driven by information security review and hardware provisioning, with less than 10 minutes of installation. Detailed documentation on installation requirements and best practices is available upon request to ensure a smooth and efficient deployment process.

5. Conclusions

The Rhino Federated Computing Platform (Rhino FCP) unlocks data silos across clouds, data centers, geographies, and organizations. Rhino FCP allows enterprises to set up computation pipelines on distributed data sources in days instead of months while maintaining stringent confidentiality and privacy. By empowering organizations to harness the power of AI, Federated Learning, and Edge Computing, Rhino FCP drives innovation and efficiency across industries. It enables secure, scalable data collaboration, making advanced analytics accessible and compliant globally. Detailed documentation and support resources are available to assist with your implementation and maximize the benefits of Rhino FCP.