A User-Centric, Privacy-Preserving, and Verifiable Ecosystem for Personal Data Management and Utilization

Osama Zafar[1 [0009−0008−9621−6899]], Mina Namazi[1 [0000−0002−8878−9362]], Yuqiao Xu[1 [0009−0009−3552−2136]], Youngjin Yoo[2 [0000−0001−8548−3475]], and Erman Ayday[1 [0000−0003−3383−1081]]

[1] Case Western Reserve University, Cleveland, OH 44106, USA
{oxz23,mxn559,yxx914,exa208}@case.edu
http://www.springer.com/gp/computer-science/lncs
[2] The London School of Economics and Political Science (LSE)
y.yoo@lse.ac.uk

Abstract. In the current paradigm of digital personalized services, the centralized management of personal data raises significant privacy concerns, security vulnerabilities, and diminished individual autonomy over sensitive information. Despite their efficiency, traditional centralized architectures frequently fail to satisfy rigorous privacy requirements and expose users to data breaches and unauthorized access risks. This pressing challenge calls for a fundamental paradigm shift in methodologies for collecting, storing, and utilizing personal data across diverse sectors, including education, healthcare, and finance.

This paper introduces a novel decentralized, privacy-preserving architecture that handles heterogeneous personal information, ranging from educational credentials to health records and financial data. Unlike traditional models, our system grants users complete data ownership and control, allowing them to selectively share information without compromising privacy. The architecture's foundation comprises advanced privacy-enhancing technologies, including secure enclaves and federated learning, enabling secure computation, verification, and data sharing. The system supports diverse functionalities, including local computation, model training, and privacy-preserving data sharing, while ensuring data credibility and robust user privacy.

Keywords: Privacy-Enhancing Technologies · Decentralized Data Management · Verifiable Computation.

1 Introduction

Managing personal data has become a defining challenge in modern digital systems. As organizations increasingly collect and store vast amounts of sensitive information in cloud-based centralized architectures, fundamental concerns about privacy, security, and ownership over data arise. Centralized systems, while efficient, pose vulnerabilities such as single points of failure and limited user control

over data, necessitating a thorough reevaluation of data governance and management.

Traditional protocols often fall short in terms of privacy, exposing users to breaches and unauthorized access. Centralized data systems are prime targets for cyberattacks, sparking concerns among users about data security and sovereignty, particularly in sectors handling extremely sensitive personal information. Healthcare organizations manage electronic health records and data from smart devices. Educational institutions handle student records and learning analytics. Financial institutions deal with confidential transactions and personal financial information. The rise of IoT devices and increased connectivity has further accelerated the exponential growth of personal data, amplifying the need for stricter privacy regulations and innovative approaches to data management.

Existing centralized platforms face three fundamental limitations. First, they create security vulnerabilities through data concentration, making them targets for cyberattacks [7]. Second, they disenfranchise individuals from data ownership and portability, allowing service providers to manage personal data without user control [9]. Third, despite their centralized nature, these systems result in fragmented data silos that collect data within their respective domains. On the other hand, current decentralized solutions address privacy concerns [4, 10]; however, significant gaps remain in developing comprehensive functionalities required for modern digital services. They struggle to perform computations on private data, facilitate controlled sharing between service providers, and maintain data utility while preserving privacy [15, 21].

To address these limitations, this work proposes a privacy-preserving, decentralized, AI-enabled data ecosystem. At its core are user-controlled decentralized entities called 'data agents' that serve as secure vaults for personal data storage and computation. These data agents provide users with privacy by design, returning control while enabling complex data utilization. The proposed architecture functions similarly to a conventional model by employing a consent mechanism through access control management. This enables users to set the access permissions of service providers to their data while keeping the raw data within the user's data agent. The proposed architecture incorporates several key technical innovations. First, it implements privacy-preserving computation capabilities, allowing service providers to run analyses without accessing underlying raw data. Second, it integrates federated learning methodologies, enabling collaborative model training across distributed data sources. Additionally, it utilizes secure enclaves, enabling computations in a secure and verifiable manner protected by robust cryptographic signatures. Building on these capabilities, our architecture delivers privacy-preserving analytics by managing and leveraging personal data responsibly. It meets the dual demands of privacy and functionality, fostering trust and innovation in an increasingly data-driven world. This positions our system as a robust and future-ready platform for addressing the complex needs of modern industry applications.

From a technical implementation perspective, the system is platform-agnostic and can be easily deployed to different cloud computing platforms. We utilize

AWS services as a case study. The proposed system leverages AWS Nitro enclaves [6] for secure computation and verifiability, ensuring that even when processing user data, proprietary models and algorithms from service providers remain confidential. Our architecture includes novel data plugs that enable secure data collection from various sources while maintaining user control through comprehensive access management tools. Integrating DIDComm [3] for secure communication ensures end-to-end privacy in all data exchanges. Robust security is a cornerstone of our architecture, implemented through a multilayered approach. At its foundation, data agents implement strict access controls and secure storage using AWS security features, including multi-factor authentication. All data collection includes digital signatures for a verifiable chain of possession.

A distinctive advantage of our proposed architecture is its ability to consolidate data from various aspects of a user's life, including finance, healthcare, education, social media, entertainment activity, GPS, driving, history, and more, all while retaining ownership with the user. This holistic data integration from the user's entire life allows service providers to develop a more comprehensive profile of the user's personality and deliver enhanced personalized services without compromising the privacy of the users. For instance, content platforms like Netflix collect data on users' preferences and recommend content to watch. However, it operates within restricted visibility of user preferences due to limited user activity on the platform. This limits their recommendation capabilities to activities within their specific service. In contrast, services like YouTube and Spotify collect similar data to recommend content, benefiting from a higher activity level due to their shorter content duration. Our proposed system enables sophisticated personalization by allowing service providers to analyze patterns across platforms and contexts, all while preserving privacy and user control. In this instance, Netflix can leverage consolidated user activity data to enhance its recommendation algorithms significantly. Similarly, other service providers can deliver highly personalized content that resonates with individual tastes by analyzing user interactions across different platforms, including social media activity and cross-platform engagement.

The proposed solution protects personal and sensitive information in an open ecosystem, providing a multidirectional data flow that allows individuals and small entities to co-create meaningful value from their data. Furthermore, it presents an economic incentive for organizations to maintain and update decentralized datasets, making the overall open data ecosystem more sustainable.

Our system has been designed to address the unique challenges of key domains such as education, healthcare, and finance, making it highly practical and relevant to these critical sectors. The architecture's potential impact extends beyond immediate applications, offering a foundation for future privacy-preserving digital services across diverse industries.

We summarize our main **contributions** as follows.

- Integration of secure and privacy-preserving computation within a decentralized framework, enabling data processing without centralization;

- Support for federated learning across distributed data agents while maintaining privacy;
- Comprehensive data agent model that combines secure storage, computation, and sharing capabilities;
- Practical prototype implementation focused on specific applications such as healthcare, education, and the finance industry requirements.

The rest of the paper is organized as follows. We summarize the related work in Section 2. The proposed architecture is introduced in Section 3, and its security and privacy analyses are discussed in Section 4. We evaluate the feasibility of our proposed scheme in Section 5 and discuss its application in Section 6. Finally, we discuss extending our framework to include more functionalities and support more robust security definitions in Section **??**, and conclude our research in Section 7.

2 Related Work

Public awareness of data rights and privacy has grown significantly in recent years. Regulations such as the General Data Protection Regulation (GDPR) [22] in the European Union and the California Consumer Privacy Act (CCPA) in the United States underscore the global shift toward improved data protection standards. These legal frameworks impose stricter data management protocols and place the responsibility on organizations to ensure robust consumer data privacy.

In response to these evolving requirements, several platforms have emerged to address the need for privacy-preserving data management. Digi.me [12] offers a personal data platform that allows users to aggregate and control their data from various sources. While providing users a centralized view of their data, it lacks the advanced computational capabilities and privacy-preserving features such as decentralized model training and secure computation environments.

MIT's Solid (Social Linked Data) project [20] and OpenPDS (Personal Data Store) [17] present decentralized data storage systems that enable individuals to collect, store, and provide fine-grained access to their data. Both systems offer a solution to store and manage personal data with complete access control. However, they do not guarantee complete data ownership, as shared data with third parties becomes part of their centralized system and can not easily be recalled.

Dataswift's PDA (Personal Data Account) [11] provides an infrastructure for individual data ownership and control. It enables individuals to collect, store, process, and use their own personal data in the cloud. It allows users to permit third-party applications to read and write data. However, it lacks control over the subsequent storage, processing, and use of users' data by the authorized third parties. .

PersonalData.IO [18] is a platform designed to empower individuals by providing them with tools and resources to control their personal data. It focuses on creating transparency and accountability in how companies collect, store, and

use personal data. By leveraging GDPR compliance and other privacy regulations, PersonalData.IO allows users to understand what personal information companies hold about them, request data access or deletion, and maintain their privacy rights.

Meeco [14] offers a user-centric data management platform that bridges the gap between personal data ownership and ethical data usage. Focusing on secure data sharing and privacy, Meeco bridges the gap between personal data ownership and ethical data usage. Its user-centric approach ensures that individuals can actively manage who has access to their information while maintaining transparency and accountability in how their data is used.

The Databox architecture [16] offers a privacy-centric approach through a local data collection and processing system that empowers individuals to control their data while enabling secure third-party sharing. Databox focuses on the local processing of IoT data and eliminates the need to send sensitive information to third-party services. It shifts data control from centralized cloud providers to users through a hybrid system with a local physical device and cloud-hosted services that work together to manage personal data collection, storage, and processing. While being a compelling privacy-centric framework for personal data management, it has certain limitations. It primarily focus on IoT data collection and processing, which restricts its ability to aggregate and leverage multi-modal data from diverse third-party services. Although the Databox allows the execution of third-party applications in isolated environments, it lack advanced computational and model training capabilities required by complex service models like recommendation systems. It also lack robust verifiability guarantees against data tampering for the results generated by third-party applications.

Data Bank model [13] is a privacy-preserving architecture for cloud-IoT platforms designed to protect users' sensitive data by giving them control over what data their devices transmit and providing tools to manage privacy-utility tradeoffs. It incorporates a category-based data access (CBDA) model for managing privacy policies, allowing data owners to define access permissions based on the data category.

P-PDS (Privacy-Aware Personal Data Storage) [19] is a user-centric system designed to automate privacy decisions for third-party access requests based on user preferences. PDS specifically offers individuals the capability to keep their data in unique logical data stores that can be connected and used by proper analytical tools or shared with third parties under the control of end users.

Current privacy-preserving data platforms generally fall into one of three categories: (i) solutions that focus primarily on decentralizing data access and ownership (e.g., Digi.me and MIT's Solid), (ii) platforms that emphasize control over data sharing mechanisms (e.g., OpenPDS and Meeco), and (iii) architectures that focus on local storage and processing capabilities (e.g., Data Bank and Databox). However, these approaches are constrained by their limited computational capabilities, which restrict their ability to perform complex analysis while maintaining privacy. Our proposed architecture distinguishes itself by enabling decentralized data control while seamlessly integrating federated learning

and secure computation capabilities without centralizing or exposing personal user data. This enables scalable data analysis without centralizing or exposing personal information. Unlike existing systems that either sacrifice privacy for functionality or rely on limited local processing, our approach allows privacy-preserving data sharing and analysis. Our distinctive design safeguards user privacy and empowers individuals to retain data ownership while contributing to valuable insights and analytics. The key innovation is that users only share the computation results rather than raw data, significantly enhancing privacy and security without compromising the data utility. This approach addresses a significant limitation in existing solutions, which often sacrifice privacy for functionality and fail to deliver actionable insights without risking data exposure.

3 Proposed Framework

We present our proposed comprehensive, decentralized, privacy-preserving, AI-driven architecture for personal data management. This section details the system model and setting, threat model, and technical components that enable secure and private data processing while maintaining utility for service providers. The used notation is introduced in Table 1.

Table 1. Used Notation

Notation	Description
$U = \{u_1, \ldots, u_n\}$	Set of users in the system
$SP = \{sp_1, \ldots, sp_n\}$	Set of service providers
$DS = \{d_1, \ldots, d_n\}$	Set of data sources
$RE = \{re_1, \ldots, re_q\}$	Set of computation requests
P	Data plug component
DA	Data agent component
UC	User controller component
MG	Model aggregator component
AC	Access control system
CP	Computation policies defining allowed computations
OP	Set of operations
$Auth$, $Cred$	Authentication system, Credential
M, M^{up}	Model, Updated model

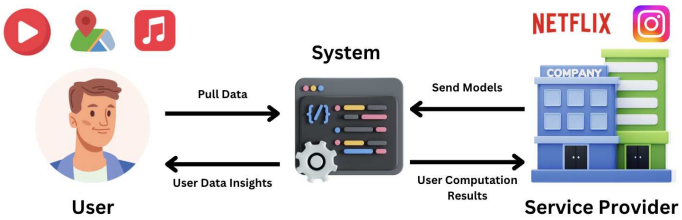

Fig. 1. Proposed decentralized, privacy-preserving, AI-driven architecture for personal data management.

3.1 System Model and Settings

The proposed architecture comprises two key actors: the set of users U and the set of service providers SP. Each party has a component, denoted by *data agent* DA, to securely convey, integrate, manage, and aggregate data from different platforms. The user's data agent consists of *data plugs* (P), and *user controllers* (UC). The SP's data agent consists of the service provider controller similar to the UC and a *model aggregator* (MG).

Data plugs P are the data collection components that pull user data through API integration with different raw data sources $DS = \{d_1, d_2, \ldots, d_n\}$ such as medical records, fitness activity records, financial records, and entertainment application activities. UC enforces the access control AC settings for all requests from the SP. We implemented a decentralized communication and verification mechanism called DIDComm [3] to establish the connection by exchanging decentralized identifiers called DIDs [2]. The data agent is the central part of the proposed architecture. They are self-identifiers that enable secure communication using verifiable digital identities. MG is a component that the service provider uses to manage its machine learning models and aggregate the users' computation results.

First, data plugs P establish secure connections to various service providers and data sources, pulling data into the system. When a SP intends to utilize this data (for generating personalized recommendations, training machine learning models, or performing analytics), they must submit a request to the user. Every request received is checked by the UC for permission in the AC setting to ensure the user allows the sender (service provider) access to perform a requested action. If the request is approved, UC runs the computation, producing a verifiable result, including a cryptographic attestation σ ensuring computational integrity.

Multiple data agents can participate in collaborative scenarios, such as federated learning, while preserving an individual's privacy. Each agent performs local computations (model training) on their data, and the updated trained models are aggregated through secure protocols without exposing raw data. The general framework of the proposed architecture is represented in Figure 1.

3.2 Threat Model

We define the security of our proposed decentralized, privacy-preserving system against external adversaries and potentially curious or malicious internal parties, including service providers.

In our framework, the users are the data owners who fully trust their data agents. Service providers are considered honest-but-curious, meaning they follow protocol instructions, but might be curious to learn additional unauthorized information. Data sources are trusted to provide accurate data, but may be compromised. Secure enclaves are trusted for secure computation, and DIDComm is trusted to establish secure communications. External adversaries are considered to have complete network control and can attempt to compromise any participant except the secure enclaves. We acknowledge that securing against side-channel attacks inside secure enclaves is not our concern.

An adversary might attempt to intercept, modify, or inject communications between system components. Our framework prevents these attacks through the DIDComm protocol, which establishes authenticated and encrypted channels between parties using decentralized identifiers (DIDs). Each data source digitally signs its data, creating a verifiable chain of possession. When data agents communicate with service providers or other data agents, they use DIDComm's cryptographic protocols to verify the authenticity of each message.

Malicious actors may manipulate model training results to corrupt the system's output or gain insight into user data through modified computations. Our framework prevents this through a comprehensive verification system. Every computation executed in a secure enclave produces an attestation that cryptographically proves the calculation was performed correctly on legitimate data. For federated learning scenarios, the model aggregator verifies each contribution's attestation before incorporating it into the global model, ensuring that only legitimate, correctly computed updates are included.

Adversaries might attempt to bypass access control mechanisms to gain unauthorized access to user data or computational resources. Each request must satisfy both the permission policy and the computation policy. The system validates all credentials cryptographically and enforces fine-grained permissions through AWS attribute-based access control. Even if an attacker obtains valid credentials, they cannot exceed the explicitly granted permissions, as the user controller component validates each request against the stored access policies before allowing any data access or computation.

The system's decentralized nature, with data stored in individual data agents, eliminates vulnerabilities associated with centralized points of failure. This distributed architecture enhances the system's overall resilience and protects against large-scale data breaches that centralized systems are vulnerable to.

We define the security model of the proposed framework and formally prove it in Section 4.

3.3 Overview

Our framework introduces an end-to-end solution that enables service providers to derive valuable insights while allowing users to maintain control over their data. The data plug component aggregates data from external sources (such as Google Maps, Spotify, and YouTube) and stores it in the user's data agent. The user's data agent enables service providers to execute computation functions on the user's data and train machine learning models in a secure and trusted manner. The system utilizes secure enclaves on the user data agents to ensure the verifiability of computations and protect the confidentiality of functions owned by service providers.

Figure 2 illustrates the general data flow within the proposed architecture, wherein the user retains control over their data through the user's data agent, while the service provider submits requests to perform computations or train machine learning models via the service provider's data agent.

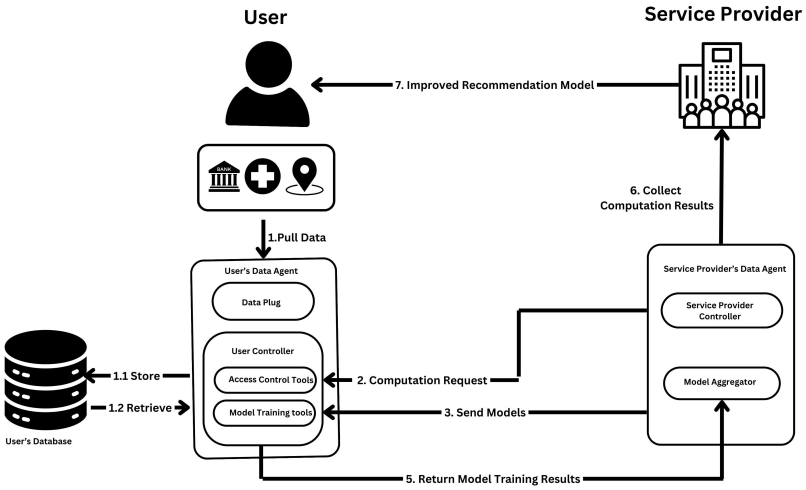

Fig. 2. General user flow diagram.

3.4 Technical Details

The technical details of the proposed framework build upon the formal definitions of data agents and their interactions with service providers through secure protocols and computation mechanisms.

User's Data Agent Components

Data plug component implements a secure data collection and integration mechanism hosted with AWS Elastic Container Service (ECS) [1]. The data plug component regularly collects the data from third-party sources (such as Google Maps, Spotify, and YouTube) via their APIs and sends it to the user controller component, which pre-processes and stores it in a database for later use. The user can configure and connect the data plug (via an app) to each new data source as it requires i) proof of the user's identity, such as username and password, and ii) the access control settings for the newly added data source. This configuration process for a new data source requires a user's credentials, such as username and password *cred*, and an access control setting for the data source AC.

User controller component is the core of the entire architecture. The main functionalities of this component are secure communication and computation. The user controller enforces access control settings for all incoming service provider requests. They check each received request to satisfy the permission in the access control setting, ensuring the user granted the sender (service provider) access to perform a requested action (building a model, computation, and sharing) using the requested piece of the user's data. The user controls the processes of connecting to the data sources and managing them through the access control settings. These settings provide the user with an interface accessible through a web browser or an app to view and manage permission levels for different types of data access. Access control can be configured for each data source, deciding which service provider can access the users' particular data and the operation

types they can run. User can update these permissions according to their preferences. To protect stored data, AWS provides a robust security mechanism with two-factor authentication that requires a username, password, and a randomly generated one-time password shared via a previously registered mode of communication like email or phone text message. Furthermore, standardized protocols, such as OpenID Connect (OIDC) and OAuth, can be adopted to provide security beyond usernames and passwords.

The access control system manages data access permissions for the U through a formal specification $AC = (U, SP, DS, CP, OP, Auth)$.

The permissions are granted for each data source DS if $Perm(DS) = (SP, OP) \rightarrow 1$.

The access control function evaluates requests RE. It allows access if all the computation policies in user-defined access control settings are satisfied, namely if the $Valid$ function outputs 1 on the inputs, $Valid(RE, CP) \rightarrow 1$, and allows the process, $Allow(RE) = Valid(AC) \wedge Valid(RE, CP)$.

Service Provider's Data Agent

Model aggregator component, MG, is operated by service providers to manage machine learning models and aggregate trained model results from user data agents. When a service provider initiates a model-related request, MG first distributes the computation task to eligible users' data agents. Each user data agent performs local computations (model training) on their private data and returns results with an attestation of the correctness of the result to the MG. Then, depending on the setting, MG implements an aggregation protocol and combines individual results. The MG maintains a set of models and manages their updates based on the aggregated results. This process is advantageous for service providers to utilize distributed computations across multiple data agents, while ensuring that individual user data is protected within each data agent. As a result, the raw data of the users is not exposed to the service provider or any other participants.

Service Provider Controller is similar to the user controller, as it sends requests from the service provider side to perform tasks on the user's data. Hence, we denoted it using the same notion UC.

We deploy a DIDComm Agent [3], which is a communication middleware to establish the decentralized communication connection between parties (between user data agents or between a user data agent and service provider) via the exchange of decentralized identifiers called DIDs [2]. DIDs are self-identifiers that enable verifiable digital identities for secure communication. They are designed to be secure and privacy-preserving using cryptographic methods to demonstrate control and ensure trust in the interactions associated with them.

The DIDComm agent sends messages from the user controller to the communication agent on the service provider side, sharing them with the controller. These agents ensure secure and protected communication of information between users and service providers. Users and service providers will have interactive web interfaces connected to their respective controller components. Similarly, using the DIDComm agents, user controllers can send messages to each other

to execute distributed computation. The web interface enables users and service providers to perform all operations efficiently and view responses easily.

System Functions

Compute function enables SP to analyze user data using a privacy-preserving method. It can perform custom functions on the data agent to compute and return derived values without directly exposing personal user data. When a service provider submits a computation request, the UC validates it against access control settings AC. Upon approval, the secure computation is executed (within an isolated enclave environment), preventing direct access to raw data. The enclave performs the specified analysis on the authorized subset of data. The framework utilizes compute functionality to run ML models locally and compute particular functions. The outcome of such computations, i.e., data products, can then be used to provide information to the user or the service provider (e.g., for dashboard analytics). An advantage of this approach is that a service provider can compute its functions using a wide variety of data that may be generated by other service providers and stored in users' data agents. When a SP submits a request with permitted operations OP, the compute function performs as $Cmp(SP, RE, DS) \rightarrow (r_{Cmp}, \sigma)$, if the operations are among the permitted operations.

Build is available to SP (by running their model aggregator) to perform federated learning and train new machine learning (ML) models. Figure 3 illustrates the data flow of the build functionality. The UC are distributed devices that train models using personal data without directly sharing such data with service providers. The MG and data agent communication is established via DIDComm [3]. Hence, the parties can engage in trusted communication without revealing unnecessary personal details of any participating user. Upon establishing the communication, the MG and UC must present one or more verifiable data products (e.g., attributes or credentials) to establish trust. Verifiable data products are attributes or credentials that an issuer digitally signs to enable the authenticity and validity of the records.

Once a UC receives a model from the MG, verifies the AC settings, trains the model using its data, and sends the trained model's result back to the SP. Formally, after initiating the communication between SP and MG and verification of each other's credentials, consequentially, and upon receiving the request, the build function outputs M^{up} as the locally trained model update by aggregating the results running $Build((r_{cmp_1}, \sigma_1), ..., (r_{cmp_n}, \sigma_n)) \rightarrow M^{up}$.

Verify functionality validates i) the correctness (or legitimacy) of user inputs to the functions provided by service providers and ii) the correctness of computation for these functions. It runs cryptographic verifications against the data products of the input user data. Therefore, the system ensures the SP's confidence in the user's data and computation results provided by the UC through secure enclaves. Hence if the signatures on the dataset and computation results are valid, the verify function is satisfied, $Vrf(SP, UC, DS, (r_{cmp}, \sigma)) \rightarrow 1$.

We leverage the AWS Nitro enclave [8], a secure virtual environment that AWS provides for verifiable and trustworthy computations. They allow users to

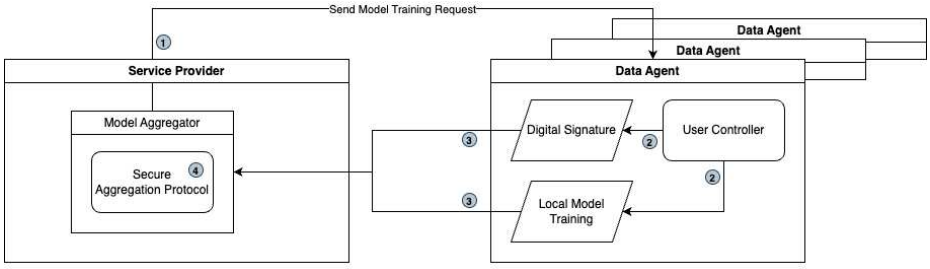

Fig. 3. Data flow of the build functionality.

establish isolated computing environments, enhancing the protection and secure processing of sensitive data. It gives service providers trust and confidence in the computation output by the user controller.

4 Security and Privacy Analyses

This Section provides the security theorem of the proposed scheme π in Section 3 based on the threat model we provided in Section 3.2.

We demonstrate user data privacy by showing that service providers learn nothing beyond the computation results and proving that responses to multiple requests from service providers leak no unauthorized information. We define the real and ideal worlds of the data agent and service provider and construct a simulator that can generate views indistinguishable from real protocol executions without access to the actual user data. We show that distinguishing between real and simulated views requires breaking the security of the underlying enclave or the attestation mechanism. The privacy analyses is guarantees under standard computational assumptions and not under information-theoretic (statistical) privacy.

Theorem 1. *For any probabilistic polynomial time (PPT) adversary $\mathcal{A}$ corrupting service provider SP, in the proposed architecture π in Section 3, there exists a PPT simulator $\mathcal{S}_1$ such that for all user data DS and computation requests $re_i \in RE$, where $i = \{1, \ldots, q\}$, the views of a $\mathcal{A}$ are computationally indistinguishable with the following security properties.*

- *Privacy: For any datasets DS_0, DS_1:*

$$\{\mathsf{VIEW}^{\pi}_{\mathcal{A}}(DS_0, \{re_i\}^q_{i=1})\} \approx_c \{\mathsf{VIEW}^{\pi}_{\mathcal{A}}(DS_1, \{re_i\}^q_{i=1})\}.$$

- *Computation Correctness: For a valid (r_{cmp}, σ):*

$$\Pr[Vrf(SP, UC, DS, (r_{cmp}, \sigma)) = 1 \wedge r_{cmp} \neq Cmp(SP, R, DS)] \leq \mathsf{negl}(k).$$

- *Access Control: For any unauthorized request R':*

$$\Pr[Allow(RE') = 1] \leq \mathsf{negl}(k)$$

- *Model Integrity: For a valid build request:*

$$\Pr[Bld(\{(r_{cmp_i}, \sigma_i)\}^q_{i=1}) \neq Agg(\{r_{cmp_i}\}^q_{i=1})] \leq \mathsf{negl}(k).$$

We analyze the privacy proof comprehensively, and the other protocol properties follow similar arguments. We prove that for any two datasets, the views of a polynomial-time adversary corrupting a service provider are computationally indistinguishable. No polynomial-time adversary can extract additional information about the underlying user data. We guarantee that privacy holds even under multiple executions of the protocol.

Proof. We construct the proof by designing a simulator $\mathcal{S}_1(1^k, re_i, r_{cmp_i})$ for a $re_i \in RE$, where q is sequence of computation requests. The $\mathcal{S}_1$ can generate an indistinguishable view from the real protocol without access to the dataset. The $\mathcal{S}_1$ inputs the re_i, the security parameter 1^k, and a protocol's honest result $\{r_{cmp_i}\} \leftarrow Cmp(SP, RE, DS)$ at each request. Then it generates the simulated signature $\tilde{\sigma}_i$, and adds $(re_i, \tilde{r_{cmp_i}}, \tilde{\sigma}_i)$ to the simulated view which is $\mathsf{VIEW}_{S_1} = \{re_i, \tilde{\sigma}_i, r_{cmp_i}\}_{i=1}^{q}$.

The real world's executed result follows the steps of the protocol, and the real view is $\mathsf{VIEW}_{\mathcal{A}}^{\pi} = \{re_i, \sigma_i, r_{cmp_i}\}_{i=1}^{q}$.

We prove in the following that the simulated view is computationally indistinguishable from the real view of the protocol.

Assume, there exists a PPT distinguisher $\mathcal{S}_2$ that can differentiate between the simulated protocol view and the real one on dataset DS with non-negligible probability ϵ:

$$\left|\Pr[\mathcal{S}_2(\text{VIEW}_A^{\pi}(DS, \{re_i\}_{i=1}^{q})) = 1] - \Pr[\mathcal{S}_2(S_1(1^k, \{re_i, r_{cmp_i}\}_{i=1}^{q} = 1]\right| \geq \epsilon$$

We can construct an adversary $\mathcal{B}$ against the enclave security using this distinguisher. $\mathcal{B}$ receives the security parameter 1^k and has access to the enclave's oracle $\mathcal{O}$. It has the honest computation results r_{cmp_i}. For each r_{cmp_i}, the $\mathcal{B}$ queries enclaves oracle $\mathcal{O}(r_{cmp_i})$. It receives θ_i that equals to a real output $\sigma_i = Sig(r_{cmp_i})$, or the simulated results $\tilde{\sigma}_i$, depending on the oracle's mode. $\mathcal{B}$ constructs its view as $V = \{re_i, r_{cmp_i}, \theta_i\}_{i=1}^{q}$. It uses $\mathcal{S}_2$ to run the view and outputs 1 if and only if $\mathcal{S}_2$'s output is 1.

The $\mathcal{B}$ simulates the real view when the oracle provides the real enclave signature. Otherwise, when oracle provides simulated enclaves signature, $\mathcal{B}$ simulates the $\mathcal{S}_1's$ view. Therefore, $\mathcal{B}$'s advantage in distinguishing between the two worlds is non-negligible, and it can break the security of the enclaves. This contradicts the security assumption and completes the proof.

The proofs for the other defined security properties, including computational correctness, access control, and model integrity, follow similar reduction arguments.

5 Evaluation

This section presents a practical implementation and evaluation of our privacy-preserving architecture. Through a functional prototype, we assess the proposed architecture's feasibility throughout the complete lifecycle of collecting, storing, and processing individuals' sensitive information while maintaining privacy guarantees.

To demonstrate the feasibility of our architecture, we have developed and deployed [3] a prototype incorporating the core components described in Section 3.4. The details of implementation, deployment, and testing use cases are as follows.

The data plug component is the primary mechanism for securely retrieving information from third-party service providers. We implement specialized data plugs for multiple platforms, including: (i) a Reddit API integration that collects social media activity data (posts, likes, dislikes), (ii) a Spotify API connection that retrieves user profile and music preference data and (iii) a direct upload functionality compatible with Google Takeout exports. The modular approach allows users to connect their accounts, provide necessary authentication credentials, and designate specific service providers as authorized data sources. Data plug implementation can be readily extended to incorporate diverse source APIs for platforms such as Facebook, LinkedIn, Google Maps, or any educational information systems (as discussed in our application scenarios in Section 6). It makes the proposed system compatible with the most popular service providers for its use in various application areas.

We have developed a web-based interface for comprehensive access control settings using AWS Attribute-Based Access Control [5] as the underlying mechanism. This implementation gives users fine-grained permission control over their data and its usage contexts, ensuring that service providers can only access information explicitly authorized by the users.

The user controller component forms the operational core of our implementation. All incoming requests from service providers are processed through an auditing system that verifies permissions against the established access control settings. Only requests with valid permission from the user are processed and executed, and the results are shared. The system supports both computation requests and model training operations across multiple data sources.

We evaluated our system's computation capabilities using data collected from the Reddit API (saved, liked, and disliked posts). Using data collected from the Reddit API, we apply a Natural Language Processing (NLP) technique called Name Entity Recognition (NER) to identify music artists mentioned in posts data collected from the Reddit platform. These extracted preferences were then shared with a music service provider (Spotify) to enhance recommendation relevance without exposing the user's raw data. Secondly, we implemented sentiment analysis computation on YouTube interaction data, calculating average sentiment scores across user comments on watched videos. This provided service providers with valuable engagement metrics while preserving user privacy.

Our implementation allows service providers to select from various computational approaches, including linear regression, statistical aggregations, and custom functions. They can also specify which user data portions should be included in the analysis, enhancing flexibility while maintaining privacy boundaries.

To evaluate the framework's usability, we conducted a comparative analysis of the computation runtime. Specifically, we compare the runtime of computation in a centralized setting (baseline) with that observed in our decentralized frame-

[3] All implementation codes are available on GitHub and will be provided upon request.

work, considering scenarios both with and without the use of a secure enclave (for evaluation system information, see Appendix B). We also assess systems' scalability by incrementally increasing the data size. Figure 4 shows that the runtime rises linearly with the volume of processed data (number of posts), with our system demonstrating comparable performance to the centralized baseline despite its enhanced privacy protections. The runtime for computations executed within the Nitro Enclave does exceed that of the decentralized framework. This disparity can be attributed to the inherent overhead associated with loading and executing computations within the isolated environment of the secure enclave, coupled with the differential in computational resources at the time of testing. The Nitro Enclave, while providing enhanced privacy guarantees, comes with a tradeoff between efficiency and elevated security features (privacy of the SP's computation functions and verifiability of the results). As the Nitro Enclave is hosted on AWS Elastic Compute Cloud (EC2), its computational capacity (CPU and memory allocation) can be increased to potentially mitigate runtime and enhance overall efficiency. This indicates that our privacy-preserving architecture maintains efficiency without significant computational overhead.

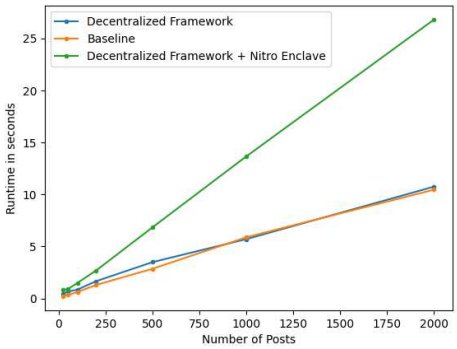

Fig. 4. Runtime of name entity recognition on Reddit data.

Similarly, we successfully implemented and tested a neural network model trained on data from multiple users using federated learning. Using data collected from the Reddit API, a classification model was trained to predict user preference, specifically whether a given title would elicit positive engagement (i.e., "liked") from users. Each user's data agent receives and executes the training request independently according to the permission of its access control setting. The data agents send their respective model weight updates to the service provider agent via secure DIDComm messages.

On the service provider side, the "service provider controller" component aggregates these weights and updates the model (detailed further in Section 3.4), then coordinates subsequent training rounds. This implementation offers service providers considerable flexibility in specifying data features for training, selecting from standard machine learning models or implementing custom architectures, and configuring various hyperparameters such as the number of training rounds.

Our implementation demonstrates that robust privacy protection and valuable data utilization can coexist within a properly designed architecture, addressing the fundamental limitations of existing approaches.

6 Applications

Our proposed decentralized privacy-preserving architecture finds multifaceted applications across several industries. In healthcare, technology can aid in pandemic prevention and monitoring by leveraging wearable devices and self-reported formats. Workforce development and enhancement can benefit through personalized data collection relating to an employee's work skills and well-being metrics, ensuring a healthier and more efficient workforce. Students can centralize all their academic and skill-related achievements within education and training, making it easier for prospective employers or educational institutions to evaluate their capabilities. In transportation, our architecture can help rental companies provide a seamless, secure, and customized experience for their customers by storing essential driving-related data. For more extensive details on additional applications, see Appendix A.

7 Conclusion

We introduced a groundbreaking architecture that fundamentally re-imagined personal data management, enabling users to control their sensitive information while allowing service providers to derive valuable insights through privacy-preserving computations. We proposed a decentralized, privacy-preserving architecture that addresses the inherent shortcomings of traditional centralized systems regarding privacy and security. The proposed framework mitigates data abuse or loss of privacy by allowing users to retain complete control and ownership over their sensitive data. By integrating advanced privacy-enhancing technologies, such as secure enclaves, verifiable computation, and federated learning, the framework allows service providers to execute secure and verifiable computations and model training while ensuring that sensitive information remains protected. This architecture offers significant advantages regarding user privacy, data security, and transparency, providing a scalable and trustworthy solution for the future of data-driven services across sectors such as education, healthcare, and finance.

In future work, we will extend the framework to support more complex interactions between data agents, enabling broad collaborative computations while maintaining strong privacy guarantees. The architecture will incorporate zero-knowledge proofs, secure multi-party computations, or differential privacy to support data analysis and sharing capabilities across domains, ensuring long-term sustainability and widespread adoption of privacy-preserving data management solutions.

8 Acknowledgments

This research was supported in part by the National Science Foundation (NSF) under awards OAC-2112606 and 2427505, and partially by the Walmart.

References

1. Amazon elastic container service, https://aws.amazon.com/ecs/
2. Decentralized identifiers (dids), https://www.w3.org/TR/did-core/
3. Didcomm messaging, https://identity.foundation/didcomm-messaging/spec/v2.1/
4. Alazab, M., Gadekallu, T.R., Pham, Q.V., Maddikunta, P.K.R., Bhattacharya, S., Piran, M.J., Hossain, M.S.: Privacy and security in distributed learning: A review of challenges, solutions, and open research issues. IEEE Transactions on Industrial Informatics (2024). https://doi.org/10.1109/TII.2024.3021234, https://ieeexplore.ieee.org/document/10278413
5. Amazon-AWS: Attribute-based access control (abac) for aws, https://aws.amazon.com/identity/attribute-based-access-control
6. Amazon.com: Aws nitro enclaves: Create additional isolation to further protect highly sensitive data within ec2 instances, https://aws.amazon.com/ec2/nitro/nitro-enclaves/
7. Aslan, O., Aktuğ, S.S., Ozkan-Okay, M., Yilmaz, A.A., Akin, E.: A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions. Electronics **12**(6) (2023). https://doi.org/10.3390/electronics12061333, https://www.mdpi.com/2079-9292/12/6/1333
8. AWS, A.: What is aws nitro enclaves?, https://docs.aws.amazon.com/enclaves/latest/user/nitro-enclave.html
9. Battiston, I., Boncz, P.: Improving data minimization through decentralized data architectures. arXiv **2312.12923** (2023), https://arxiv.org/abs/2312.12923
10. Bernal Bernabe, J., Canovas, J.L., Hernandez-Ramos, J.L., Torres Moreno, A., Skarmeta, A.: Privacy-preserving solutions for blockchain: Review and challenges. IEEE Access **7**, 164908–164940 (2019). https://doi.org/10.1109/ACCESS.2019.2950872, https://ieeexplore.ieee.org/document/8888155
11. Dataswift: Dataswift - personal data accounts (2023), https://dataswift.io/
12. digi.me: digi.me - your data, your way (2023), https://digi.me/
13. Fernández, M., Franch Tapia, A., Jaimunk, J., Martinez Chamorro, M., Thuraisingham, B.: A data access model for privacy-preserving cloud-iot architectures. In: Proceedings of the 25th ACM Symposium on Access Control Models and Technologies. p. 191–202. SACMAT '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3381991.3395610, https://doi.org/10.1145/3381991.3395610
14. Meeco: Meeco: Enterprise infrastructure for the personal data economy. https://www.meeco.me/, accessed: 2024-12-03
15. Mireshghallah, F., Lundgren, M., Asghari, P., Ren, S., Kuzmanovic, A., Nilizadeh, S.: Privacy-preserving machine learning: Methods, challenges, and directions. arXiv preprint arXiv:2108.04417 (2021)
16. Mortier, R., Zhao, J., Crowcroft, J., Wang, L., Li, Q., Haddadi, H., Amar, Y., Crabtree, A., Colley, J., Lodge, T., Brown, T., McAuley, D., Greenhalgh, C.: Personal data management with the databox: What's inside the box? In: Proceedings of the 2016 ACM Workshop on Cloud-Assisted Networking. p. 49–54. CAN '16, Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/3010079.3010082, https://doi.org/10.1145/3010079.3010082
17. openpds: Openpds, personal data with privacy, https://openpds.media.mit.edu

18. PersonalData.IO: Home - personaldata.io. https://personaldata.io/en/home-en/, accessed: 2024-12-03
19. Singh, B.C., Carminati, B., Ferrari, E.: Privacy-aware personal data storage (p-pds): Learning how to protect user privacy from external applications. IEEE Transactions on Dependable and Secure Computing **18**(2), 889–903 (2021). https://doi.org/10.1109/TDSC.2019.2903802
20. Solid: Solid: Your data, your choice, https://solidproject.org
21. Unknown, A.: Synergizing privacy and utility in data analytics through advanced techniques. arXiv preprint arXiv:2404.16241 (2024)
22. Wikipedia contributors: General data protection regulation (2021), accessed: Mar 30, 2025

A Extended Applications

Healthcare data collected from the wearable devices and apps is stored in the personal data agents rather than shared directly with service providers. Collected data is then used for i) executing local AI models to provide insights to the users; ii) developing new AI models via federated learning; and iii) privacy-preserving data sharing with the medical providers.

Similarly, users can aggregate data from fitness trackers, such as Google Fitbit, which monitor heart rate, sleep patterns, and physical activity levels, along with data from health apps, nutrition tracking apps, and other related applications. Healthcare providers can access insights derived from this data while users maintain privacy and control, enabling personalized treatment recommendations without exposing raw data.

Another use case applies to collecting employee health data to promote workforce efficiency and wellness. This system: i) aggregates stress-related metrics from various sources and provides personalized recommendations to employees; ii) develops improved stress-level inference models and facilitates team-building through privacy-preserving compatibility assessments.

Furthermore, healthcare professionals can connect productivity tracking applications to generate a comprehensive profile when seeking employment opportunities. Employers can receive cryptographic proofs of the employee's qualifications and skills, ensuring authenticity while respecting privacy.

Education and training includes transcript and skill management for students. Collection of all skills a student has gained (including transcripts, training, reference letters/endorsements, books, or videos) in the student's user agent. Processing such material to extract the skills and providing a set of particular skills (in a verifiable way) to a potential future employer or school is an application of our proposed scheme. The system can also provide privacy-preserving dashboarding. It can also create AI models using the data stored in students' user agents.

Usage-Based Auto Insurance (UBI) determines auto insurance premiums based on individual driving behavior, mileage, and data collected from vehicles through a telematics device. This device records crucial metrics like speed and

braking patterns. First, it sends the data to the driver's Data Agent, which processes it locally to generate a driving score without revealing raw data.

The Data Agent employs verifiable computation to protect privacy, creating a cryptographic signature for the driving score that reflects the driver's behavior. Insurance providers can assess risk and calculate premiums based on this score while maintaining data confidentiality and privacy. Drivers can selectively disclose their driving data, and as the Data Agent updates the score over time, it can adjust premiums accordingly.

This integration enhances privacy, enables accurate information sharing, and builds trust between drivers and insurers. It also allows storing driving credentials in a digital wallet linked to the driver's decentralized identity, facilitating potential sharing with other insurers and promoting fairness in premium calculation.

Personalized Treatment Planning with a Digital Twin replicas of physical entities like organs, systems, or entire patients are increasingly being used to simulate and analyze various medical conditions and treatment options. Data agents are crucial in managing and securing vast amounts of sensitive data in creating and utilizing digital twins. For example, consider a patient named Bob undergoing treatment for a complex cardiac condition. His healthcare provider creates a digital twin of his heart. A detailed virtual model that simulates how his heart functions under different conditions and treatment scenarios. Cardiologists use this digital twin to plan and optimize Bob's treatment.

Throughout this process, data agents are essential in securely managing the collection and integration of data from various sources, including Bob's medical records, imaging data, genetic information, and real-time data from wearable devices. The data agent ensures that all of Bob's sensitive health data is decentralized, giving him complete control over who can access it. During the simulation phase, the data agent facilitates the secure sharing of necessary data with simulation tools, ensuring that only authorized parties can contribute to the analysis without directly accessing Bob's raw data.

Sports Field Fan Engagement employs Decentralized Identifiers (DID) to verify credentials, thereby enhancing security, compliance, and fan engagement. The system stores digital credentials on a decentralized network, which enables tamper-proof validation at venue entry points. The system facilitates age verification at point-of-sale terminals without compromising personal information. Furthermore, this credential infrastructure supports an integrated loyalty program, allowing fans to redeem their credentials for various rewards. The system also has potential applications for regulatory compliance in sports betting platforms.

Government Services and Administration: Our system can enhance government operations by improving service delivery while protecting citizen privacy. It allows eligibility verification for social benefits without disclosing detailed income, enables secure inter-agency information sharing, and facilitates anonymous census data collection. Companies can prove contract compliance without revealing proprietary information, and a digital identity system can ensure se-

cure e-government access. Additionally, it supports verifiable participation in public consultations and allows travelers to prove visa compliance without sharing travel history.

B Evaluation Setup

For evaluation, the system is deployed locally on a Windows machine equipped with an Intel(R) Core(TM) i7-10750H CPU and 16 GB of RAM. For a secure enclave, we deploy an AWS Nitro Enclave inside Amazon Elastic Container Service, utilizing 2 m5xlarge CPUs and 4 GB of RAM.