AIRAVAT

Lakshya Rawat, *Project Lead & Blockchain Developer* Ayush Bansal, *Zero Knowledge Expert and Cyber Security Develope* Abhinav Singh, *Agentic AI Developer* and Akhil Murarka, *AI Developer*

***Abstract*—In the rapidly evolving landscape of financial technology, the responsible sharing of sensitive customer data has become both a critical necessity and a significant challenge. Airavat is a comprehensive, proof-of-concept platform designed to address this concern. Developed during the SuRaksha Hackathon under the theme "Securing Sensitive Customer Data in Fintech Ecosystems with Responsible Data Sharing and Privacy Protection," Airavat leverages advanced technologies such as AI agents, zero-trust architecture, blockchain auditability, and privacy-enhancing frameworks. The system empowers users with granular consent control and provides third parties with verifiable, secure, and policy-compliant access to data, ensuring robust protection, transparency, and adherence to data privacy regulations like India's DPDP Act and the EU's GDPR.**

***Index Terms*—IEEE, IEEEtran**

I. Introduction

In the rapidly evolving fintech ecosystem, the protection of sensitive customer data has become a pressing challenge. With the rise of digital banking, online transactions, and third-party financial services, user data such as bank account details, transaction histories, and personal identifiers are increasingly being shared across platforms. This raises critical concerns around data privacy, misuse, unauthorized access, and regulatory non-compliance. The need for secure and responsible data-sharing mechanisms has never been more vital—particularly in light of global regulations such as the General Data Protection Regulation (GDPR) and India's recently enacted Digital Personal Data Protection (DPDP) Act.

To address this challenge, we developed ***Airavat***, a proof-of-concept application for the SuRaksha hackathon under the theme "Securing Sensitive Customer Data in Fintech Ecosystems with Responsible Data Sharing and Privacy Protection." Airavat reimagines data privacy using a combination of AI-powered agents, zero-trust architecture, blockchain auditability, and privacy-enhancing technologies to offer a secure, transparent, and user-consent-driven data-sharing framework.

At its core, Airavat is designed to give users granular control over how their financial data is shared, for how long, and for what purpose—ensuring that every transaction is both ethically compliant and technically secure. Through its modular multi-agent system, Airavat not only enforces policy and access boundaries but also brings transparency to the data-sharing lifecycle by maintaining an immutable record on the blockchain and demonstrating the use of zero-knowledge proofs to validate data without revealing it.

II. Objectives

The primary goals of **Airavat** are:

- **Enable privacy-preserving and regulation-compliant data exchange** between users and third-party entities in fintech environments.
- **Provide users with fine-grained control** over their personal and financial data, including configurable access levels, use-case permissions, and time-bound sharing constraints.
- **Ensure complete transparency through blockchain-based logging**, creating immutable, tamper-proof audit trails for every data request and transaction.
- **Leverage AI agents in a zero-trust architecture** to dynamically assess risk, validate transactions, and prevent unauthorized access without relying on implicit trust.

III. System Overview

A. User Dashboard Overview

Airavat is designed on a modern web architecture that ensures users experience fast, secure, and responsive dashboards. The backend infrastructure supports seamless interaction between AI agents, dashboards, and privacy modules, enabling dynamic consent handling and real-time security validation.

B. Security Architecture

The platform adopts a zero-trust security model, emphasizing verification over implicit trust. User credentials are securely hashed using bcrypt, and JWTs are stored in HTTP-only cookies to prevent XSS attacks. Middleware functions protect backend routes by validating tokens and enforcing session checks before allowing access to sensitive operations. Logout endpoints automatically invalidate the token, ensuring that expired or stolen tokens cannot be reused. The backend infrastructure leverages MongoDB with Mongoose ORM for structured and reliable storage, and all authentication operations are protected through industry-standard encryption and hashing mechanisms.

C. Consent Management System

A central feature of Airavat is its consent management system, which empowers users to define, review, and modify how their data is accessed and used. Consent preferences include data sharing levels (Minimal, Moderate, Full) across key categories like Transaction Data, Account Details, and Personal Information. Additionally, users can enable or disable sharing for specific purposes such as loan processing, fraud detection, credit scoring, and marketing. A time-bound access control setting (ranging from 1 to 365 days) allows users to specify how long data may be shared, and an optional notes section offers space for context or restrictions. These preferences are securely stored and dynamically enforced during every third-party request.

D. Multi-Agent AI System

Airavat is architected around a modular multi-agent system comprising six specialized AI agents that collaboratively secure data flow and enforce policy decisions. These include:

- **VRA (Vigilant Risk Analyzer):** Assesses the risk level of incoming data access requests based on context and user-defined limits.
- **RBA (Request Brainiac Agent):** Parses and interprets incoming access requests to determine their purpose and scope.
- **TMA (Task Maestro Agent):** Coordinates task execution among other agents to ensure smooth, rule-compliant data sharing.
- **BBA (Blockchain Builder Ace):** Logs each validated transaction onto a blockchain for transparent, tamper-proof auditing.
- **ZKBA (Zero-Knowledge Builder Ace):** Demonstrates data validity without revealing actual user data using zero-knowledge proofs.
- **OCA (Orchestration Control Agent):** Oversees all agent interactions and ensures real-time enforcement of user consent preferences.

IV. System Architecture

The architecture of Airavat is designed to ensure privacy-preserving, secure, and transparent data exchange in fintech ecosystems. It is built around a modular, multi-layered approach that incorporates modern web frameworks, decentralized logging, and AI-powered agents operating within a zero-trust environment. The entire system is orchestrated to ensure robust user privacy, consent enforcement, and secure communication between stakeholders.

A. Platform Stack

At its core, Airavat is built using **Next.js 15** for the frontend and API layers, offering a fast and scalable framework with server-side rendering and routing capabilities. **MongoDB**, accessed via **Mongoose**, serves as the backend database, storing user credentials, consent preferences, and audit logs. Authentication is handled through **JWT (JSON Web Tokens)**, with secure, HTTP-only cookie storage and route-level middleware validation.

Security is enhanced using **bcryptjs** for password hashing, **Radix UI** and **Lucide Icons** for modern UI components, and **Tailwind CSS** for consistent styling and theming. This setup ensures a responsive, secure, and developer-friendly environment for building sensitive fintech applications.

B. Component Layers

The system is divided into the following functional layers:

- **Presentation Layer:** Includes user interfaces for both bank and customer dashboards. It allows users to register, log in, configure consent preferences, view logs, and control data access.
- **API Layer:** Exposes secured endpoints for authentication (*/register, /login, /logout*) and consent management (*/consent-preferences*) with input validation and access control.
- **Data Layer:** Handles storage of user data, hashed passwords, and consent preferences. MongoDB collections are organized to ensure minimal privilege access and logical separation of concerns.
- **Security Layer:** Implements JWT-based authentication, cookie based session protection, and dynamic route guards to prevent unauthorized access.
- **Blockchain Logging Layer:** All consent changes and third-party data accesses are recorded using blockchain-inspired logging, enabling tamper-proof audit trails.
- **AI Agent Layer:** Six autonomous agents operate asynchronously to analyze, validate, orchestrate, and log activities in the system based on their respective domains.

C. Zero Trust Security Model

Airavat's architecture is centered around the Zero Trust model, which operates on the principle of "never trust, always verify." This means no internal component or user is implicitly trusted—authentication, authorization, and context verification are continuously enforced. Multi-agent validation, JWT validation, and consent compliance checks occur at every stage of interaction.

D. Data Flow Summary

- A user registers and logs in using the dashboard. JWT tokens are issued and securely stored in HTTP-only cookies.
- Upon login, default consent preferences are created in the database.
- The user can modify their preferences, which are then stored and versioned.
- When a third-party data request is made (e.g., loan processing), agents evaluate risk, verify consent, and either allow or deny access.
- All decisions and actions are logged on a blockchain-inspired system for transparency.

V. Key Features

A. Fine-Grained Consent Management

Airavat empowers users to maintain full control over their personal and financial data through a granular consent management system. Users can define **who** can access their data, **for what purpose**, and **for how long**, with options to set:

- Purpose-bound sharing constraints
- Time-limited access periods
- Revocable consent at any point

This ensures that data sharing is always transparent, intentional, and in alignment with privacy regulations like GDPR or India's DPDP Act.

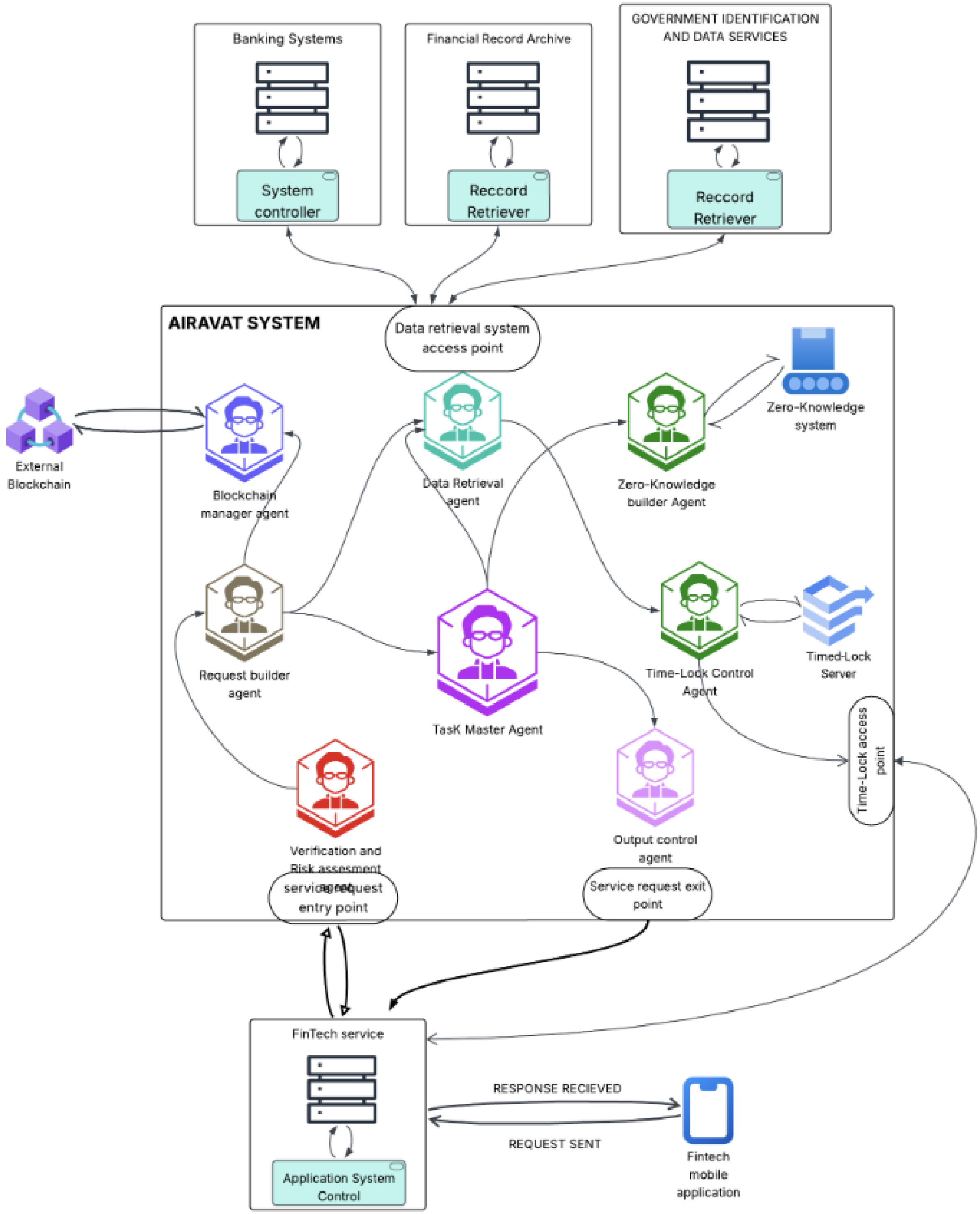

Fig. 1. System Architecture of Airavat

B. Blockchain-Based Audit Trail

To guarantee transparency and prevent tampering, every data request, consent grant, and transaction is **hashed and recorded on a private blockchain ledger**. This immutable log allows users, auditors, and service providers to verify:

- When data access occurred
- Which entity accessed the data
- Whether access complied with granted permissions

Such transparency builds trust and adds a verifiable accountability layer for all parties involved.

C. AI-Driven Zero Trust Security

Instead of relying on traditional perimeter-based security models, Airavat integrates a **Zero Trust Architecture** backed by AI agents. These intelligent agents dynamically:

- Validate session authenticity
- Assess behavioral anomalies
- Prevent unauthorized access in real-time

This model eliminates implicit trust, ensuring each transaction is continuously verified before approval.

D. Role-Based Dashboards for Users and Banks

Airavat provides separate, context-aware dashboards:

- **User Dashboard**: Enables individuals to view data requests, manage consents, monitor activity logs, and revoke access easily.
- **Bank/Third-Party Dashboard**: Lets financial institutions request data with specified purposes, track request statuses, and stay compliant with access constraints.

These dashboards are designed with usability and privacy awareness in mind, allowing both transparency and ease of use.

E. Zero-Knowledge Proof Demo

To showcase potential privacy-enhancing technologies (PETs), Airavat includes a demo implementation of **Zero-Knowledge Proofs (ZKPs)**. In this proof-of-concept, users can validate ownership of sensitive data **without revealing the data itself**, offering a glimpse into future-ready fintech authentication mechanisms.

VI. Methodology

Airavat employs a **modular, agent-based methodology** inspired by the *zero-trust architecture* to secure sensitive customer data in fintech systems. The entire workflow is governed by autonomous agents, each executing a specific task in the data access lifecycle—from request creation and policy enforcement to logging and delivery. These agents work in tandem to ensure **privacy**, **purpose-bound access**, **traceability**, and **compliance**.

A. Request Initialization and Validation

Verification and Risk Assessment Agent (VRAA)

The process begins when a third-party application initiates a request to access user data. The **VRAA** authenticates the requesting entity using:

- JWT verification,
- Integrity checks, and
- Evaluation of prior behavior history.

It calculates a *risk score* for the requester based on static (e.g., IP, headers) and dynamic signals (e.g., request frequency). Requests exceeding the trust threshold are passed to the next stage; others are blocked or flagged.

B. Request Structuring and Delegation

Request Builder Agent (RBA)

The **RBA** constructs a structured request payload, specifying:

- Which data attributes are needed (e.g., PAN, Salary Slip),
- Purpose of use (e.g., credit assessment),
- Validity duration, and
- Required security constraints.

It ensures the request matches the predefined schemas and includes proper metadata for downstream validation.

Task Master Agent (TMA)

The **TMA** acts as an orchestration agent that:

- Coordinates all downstream agents,
- Sets deadlines,
- Defines the execution path, and
- Logs intermediate status for traceability.

It ensures that no unauthorized or unverified access path is allowed by enforcing strict stage-by-stage transitions.

C. Data Collection and Privacy Protection

Data Retrieval Agent (DRA)

The **DRA** interfaces with the MongoDB database or other approved sources to fetch only the *allowed data fields* as per user consent. It strictly follows:

- Field-level masking,
- Minimum disclosure principle, and
- Role-based access control (RBAC) checks.

Zero-Knowledge Builder Agent (ZKBA)

For sensitive scenarios (e.g., verifying salary $>$ Rs. 50K without disclosing exact value), the **ZKBA** constructs *zero-knowledge proofs*. These proofs allow verification without revealing the actual data. ZKBA currently simulates basic ZK logic, with a placeholder for pluggable zk-SNARK integrations in future iterations.

D. Compliance, Logging, and Auditing

Blockchain Builder Agent (BBA)

The **BBA** hashes and immutably logs the entire transaction flow to a *local blockchain ledger*. Logged fields include:

- Timestamp,
- Request metadata,
- Agent decisions, and
- Outcome (approved/denied).

This guarantees **non-repudiation**, **traceability**, and **audit readiness** under frameworks like RBI-DPS and the DPDP Act.

E. Output Control and Conditional Delivery

Output Control Agent (OCA)

The **OCA** is responsible for *final packaging and conditional release* of the data. It ensures:

- All agents approved the flow,
- Masking and format policies are correctly applied,
- Tamper checksums match.

It also applies purpose-specific formatting (e.g., PDF, JSON) and generates secure download links or API responses.

F. Timed Access Control

Time-Lock Server & Time-Lock Server Agent (TLS & TLSA)

Together, these enforce *time-bound access validity*. The **TLSA** embeds time constraints into the access token/response. The **Time-Lock Server (TLS)** monitors token lifetimes and revokes expired access links. It also blocks re-use of expired credentials, protecting against *replay attacks* and *access misuse*.

Summary

This layered, multi-agent methodology ensures:

- Secure and consent-bound data sharing,
- End-to-end traceability via blockchain logging,
- Risk-based request filtration,
- Autonomous privacy enforcement using zero-knowledge proof mechanisms, and
- Time-bound and purpose-limited data access.

Airavat's modular agent flow is designed to be **composable**, **extensible**, and **compliant** with emerging data privacy laws.

Test Scenario	Result	Status
Consent-compliant data request	Data shared	Passed
Consent-violating data request	Access denied	Passed
Unauthorized role access	Blocked	Passed
Expired JWT reuse attempt	Invalid token	Passed

TABLE I
FUNCTIONAL TESTING RESULTS

VII. Evaluation and Results

To evaluate the effectiveness and robustness of **Airavat**, we focused on functional validation, security behavior, and user interaction across various test scenarios that mimic real-world fintech operations. While the system was developed as a proof-of-concept during the hackathon, key components were tested against predefined benchmarks.

A. Functional Testing

We validated the end-to-end flow from data request to secure delivery, including:

- **Consent Enforcement**: Verified that data requests failing to match purpose or time constraints were correctly denied.
- **Role-Based Access**: Different user roles (Admin, Agent, Viewer) could access only their authorized data.
- **Session Expiry and Token Invalidity**: JWT tokens were tested to auto-expire after the configured time window with proper logout behavior.

B. Security Evaluation

We performed simulation tests for common attack vectors:

- **Replay Attack Prevention**: Each request was assigned a unique token and timestamp using the Time-Lock Agent.
- **Tamper Detection**: All access events were hashed and logged on-chain using the Blockchain Builder Agent.
- **Password Security**: `bcryptjs` ensured all passwords were hashed with salt, preventing plaintext exposure.

C. Performance and Responsiveness

- The frontend, built on Next.js 15 with Tailwind CSS and Radix UI, achieved sub-1s load times for all core pages.
- Backend operations like consent validation and blockchain logging executed in under 300ms on average.

D. User Experience

Participants rated the platform's usability highly during demo feedback, especially appreciating:

- The clear, modular dashboard.
- Real-time visibility into who accessed what data and when.
- The ability to revoke consent mid-session.

E. Technical Milestones Achieved

- JWT-based auth with cookie-based session management.
- Blockchain-based immutable logs with live transaction simulation.
- ZKP (Zero Knowledge Proof) demo integration for proof of consent.
- Fine-grained access control enforced via multi-agent logic.

VIII. Conclusion

Airavat presents a robust and forward-thinking approach to securing sensitive customer data within the fintech ecosystem. By integrating zero-trust architecture, AI-driven agents, blockchain audit trails, and privacy-preserving mechanisms like zero-knowledge proofs, the system ensures responsible data sharing while maintaining user autonomy and regulatory compliance. The modular, agent-based design of Airavat allows for flexible integration with existing infrastructures and easy scalability across various fintech applications.

Our architecture emphasizes transparency, accountability, and consent-centric access, empowering users with control over their personal data. With promising initial results and a practical implementation within a limited hackathon timeframe, Airavat demonstrates the feasibility of building real-world, privacy-first fintech systems that uphold the highest standards of trust, security, and compliance.

Appendix A
Proof of the First Zonklar Equation

Appendix one text goes here.

Acknowledgment

The authors would like to thank...

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.