

PureAjo: A P2P Blockchain-Based Insurance Platform

Anthony Uchenna Eneh^{1†}, Love Allen Chijioke Ahakonye², Jae Min Lee¹, Dong-Seong Kim^{1*}

¹ IT-Convergence Engineering, Kumoh National Institute of Technology, Gumi, South Korea

* NSLab Co. Ltd., Gumi, South Korea, Kumoh National Institute of Technology, Gumi, South Korea

² ICT Convergence Research Center, Kumoh National Institute of Technology, Gumi, South Korea

anthony.u.eneh@gmail.com[†], (loveahakonye, dskim, ljmpaul)@kumoh.ac.kr

Abstract—Insurance is essential for financial resilience; however, traditional systems remain costly, opaque, and inaccessible, particularly in underserved regions. In many such communities, informal savings schemes like *akawo*, *esusu*, and *ajo* offer grassroots risk pooling; however, these schemes lack scalability, transparency, and fraud resistance. Existing blockchain-based insurance platforms address some of these issues, but often replicate centralized models, rely on token-based governance, or overlook the cultural relevance of local financial systems. This paper presents PureAjo, a fully decentralized peer-to-peer insurance platform that digitizes traditional communal models using smart contracts. Users connect their wallets to join insurance networks, pay premiums, file claims, and vote on outcomes, entirely from the frontend using wallet signatures. Governance follows a one-wallet, one-vote model, eliminating the need for tokens or custodial logic. We evaluated PureAjo on the Mumbai testnet. Core contract interactions executed with average gas usage between 88k and 143k units, and confirmed within 10–14 seconds. PureAjo demonstrates that fully decentralized, culturally grounded insurance systems are not only possible but also performant. It lays the groundwork for scalable deployment on PureChain, a dedicated Layer 2 network optimized for mutual finance.

Index Terms—Blockchain, Purechain, Insurance, peer-to-peer, Ethereum, Smart Contracts, dApp, Nextjs.

I. INTRODUCTION

Insurance plays a crucial role in mitigating individual risk and fostering financial stability. However, traditional insurance systems are often hindered by centralization [1], lack of transparency, high operational costs, and limited accessibility in underserved regions. These challenges create distrust among users and exclude large populations from formal risk protection mechanisms. Peer-to-peer (P2P) insurance, where groups of individuals contribute to a shared risk pool and vote on claims, has emerged as a promising alternative [2]. Yet, implementing such systems at scale while maintaining transparency and fairness requires infrastructure that traditional web technologies alone cannot guarantee [3].

Recent advances in blockchain technology have enabled decentralized applications that replace traditional intermediaries [4], allowing for a P2P insurance platform enabled by smart contracts [5]. Consensus mechanisms are pivotal in the performance of these systems for effective outcomes [6]. Terea et al. [7] provide a holistic view of blockchain adoption in the insurance industry, contributing to a better understanding of the challenges and enablers that specifically affect this

sector. The study also presents guidelines for establishing a blockchain-based insurance strategy. These platforms automate policy creation, funding, and claims execution, offering a transparent, immutable, and trustless alternative to conventional insurance. The result is a more efficient system with reduced administrative costs and improved fairness in policy and claims management [5].

Over the past decade, several decentralized insurance platforms have emerged in response to the limitations of traditional insurance systems, leveraging smart contracts to automate payouts, reduce overhead, and promote transparency. Projects like Etherisc, Open Cover, and DeFi Risk have pioneered mechanisms such as parametric triggers, oracle-based assessments, and decentralized claim voting to disintermediate insurance processes [8]–[10]. However, despite these innovations, existing solutions often suffer from over-reliance on external data sources, rigid actuarial logic, and complex onboarding flows that limit accessibility, especially for users in underserved regions [3], [11]. Many protocols require centralized underwriters or hybrid Web2 authentication methods, introducing trust assumptions and technical barriers that undermine complete decentralization. Moreover, the absence of localized design considerations means these systems fail to resonate with grassroots economic models worldwide.

In this paper, we present **PureAjo**, a fully decentralized P2P insurance platform built on blockchain and smart contracts. Modeled after a local Nigerian language *ajo* [12], [13] and inspired by traditional rotational savings and credit associations (ROSCAs), such as *esusu* and *ajo* in West Africa, *sususu* in the Caribbean, *tandas* in Mexico, *arisan* in Indonesia, and *chit funds* in India [?], [14], [15]. PureAjo enables users to form self-governed insurance networks where premium payments, claim submissions, and voting are all transparently recorded on-chain. These culturally embedded schemes share a cooperative ethos, where members contribute regularly to a shared pool and take turns receiving lump-sum payouts. PureAjo departs from traditional web application models by eliminating the need for email and password accounts. Instead, users connect their crypto wallets to authenticate and interact with the platform. Upon connection, the system checks whether the user's wallet address is already registered. If not, the user is prompted to fill out a lightweight off-chain profile. All further interactions, such as joining networks, paying premiums, filing

claims, and voting, are executed directly from the frontend using wallet signatures and smart contract calls. The backend plays a minimal role, storing only user profiles and a log of event-driven notifications. Notifications are generated from blockchain events and persisted to track read/unread status. The platform ensures that all financial and governance-related activity is verifiable, immutable, and user-controlled.

PureAjo aims to:

- Increase trust and transparency in insurance processes using public blockchains.
- Empower communities to co-manage risk without relying on centralized institutions.
- Preserve local financial practices by digitizing and scaling traditional savings cultures.
- Promote user sovereignty through wallet-based authentication and direct chain interaction.

We describe the architecture, implementation, and evaluation of PureAjo, focusing on its smart contract design, event-driven notification system, wallet-based UX, and Layer 2 deployment strategy. We also analyze performance metrics and discuss the trade-offs between decentralization, usability, and scalability. The ultimate deployment goal is to launch PureAjo on a dedicated Layer 2 blockchain, PureChain, optimized for mutual finance applications.

II. RELATED WORKS

Studies show that blockchain adoption can reshape the insurance industry from an economic and business perspective, contributing to a better understanding of the challenges and enablers that specifically affect the business sector [7]. Recent advances in decentralized insurance systems have explored various approaches to risk sharing through blockchain technologies [16].

Etherisc [17] proposes a modular framework for building decentralized insurance products using smart contracts and oracles. It focuses on automating parametric products such as flight delay and crop insurance, offering reusability, risk pooling, and integration with external data feeds. Etherisc has seen practical deployment and serves as a reference architecture for DeFi-based insurance.

BAKUP [18] targets supply chain disruptions by combining IoT devices with blockchain-based parametric insurance. It automates the insurance process for cargo damage and loss using external sensors and oracles to trigger claims, reducing the need for manual assessment. Though domain-specific, it advances ideas in automated and trustless insurance logic.

DeFi Risk Transfer [10] explores a pooled insurance model leveraging decentralized finance primitives. The system allows users to deposit capital into mutual risk pools and vote on claims through governance tokens. While it does not adopt local or community-based insurance structures, its approach to composability, liquidity, and incentive design is instructive.

Parametric Crop Insurance [19] uses blockchain smart contracts and satellite-derived data (NDVI and rainfall) to trigger automatic crop insurance payouts. Its simulation on Ethereum testnets confirms the potential for low-cost, scalable

agricultural risk protection. This work addresses the latency and verification issues in traditional crop insurance by replacing claims processing with parametric triggers.

FairShareDAO [20] introduces a mutual aid model governed by reputation rather than capital or token-weighted voting. The platform uses behavior modeling and machine learning to ensure fairness and discourage abuse, offering an alternative path to Sybil-resistant governance. While still theoretical, it demonstrates the potential of AI in decentralized insurance.

Open Cover [21] presents a generalized marketplace architecture for decentralized insurance, allowing users to create, fund, and claim from insurance pools. The system emphasizes modularity, user onboarding, and transparency. It also confronts regulatory and legal challenges in building decentralized risk markets.

Compared to these works, **PureAjo** uniquely combines wallet-based authentication, frontend-driven Smart contract interaction, community-driven voting, and the cultural ethos of local thrift systems like *ajo*, *esusu*, and *akawo*. It emphasizes usability, transparency, and localization, offering a bottom-up framework for risk management within informal economies.

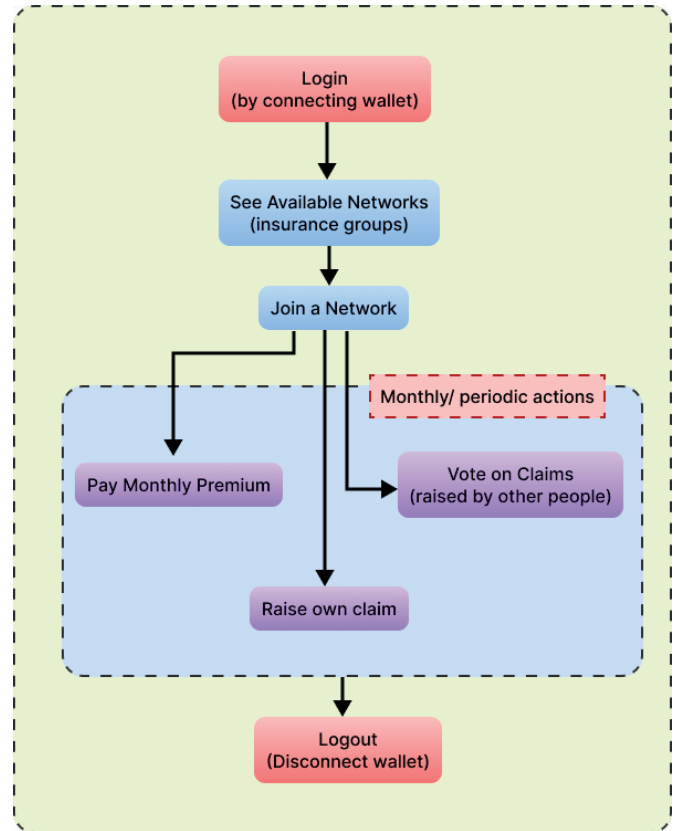


Fig. 1: User Flow

III. METHODOLOGY

This section outlines the design methodology used in developing PureAjo, emphasizing decentralized architecture, user

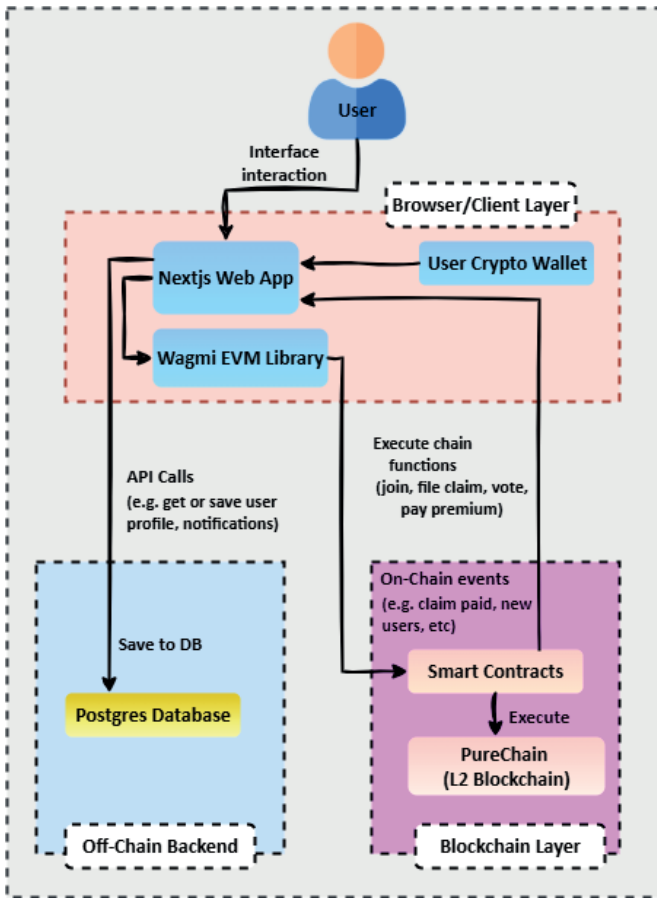


Fig. 2: System Design

experience, and integration of smart contracts with frontend and backend systems.

A. Design Philosophy

PureAjo is grounded in the principles of decentralization, transparency, and user sovereignty. All trust-sensitive operations—such as premium payments, claims, and voting—are executed on-chain using smart contracts. The frontend facilitates user interaction via wallet connections, while the backend is minimized to a supporting role that enhances usability without introducing centralization.

B. System Components

- **Frontend:** Built using Next.js 14 and TailwindCSS, the frontend acts as the primary interface for user interactions. Wallet connection and transaction execution are handled using wagmi and viem, ensuring seamless integration with smart contracts.
- **Smart Contracts:** Written in Solidity and deployed to the Sepolia Testnet, the contracts encode the core logic for network creation, membership, premium payments, claims, voting, and payouts. All significant user actions are represented as on-chain state transitions.
- **Backend:** The backend, built with Next.js API routes, uses Prisma ORM with PostgreSQL to manage only

non-sensitive data such as user profiles and notification metadata. It listens to blockchain events and persists them for read/unread tracking in the user interface.

- **Blockchain Layer:** The platform is deployed on the Sepolia Testnet for evaluation and will be migrated to a Layer 2 chain (PureChain) for production use. This ensures low fees and fast confirmations while maintaining security.

C. Workflow Overview

- 1) **Wallet Authentication:** Users connect a Web3 wallet (e.g., MetaMask) to initiate identity. No email/password authentication is used.
- 2) **Profile Bootstrapping:** If the connected wallet is not associated with an existing profile, the user is prompted to submit basic information (name, email, image) stored off-chain.
- 3) **Network Participation:** Users browse or join insurance networks by interacting with smart contracts. Membership status and payment history are queried directly from the blockchain.
- 4) **Claim Submission and Voting:** When filing a claim, the user invokes a contract method with claim metadata. Other members of the network vote on the claim using on-chain voting logic.
- 5) **Notification System:** On-chain events such as new claims or votes are monitored. The backend captures and stores these events as notifications for the user, including read/unread state.
- 6) **Premium Payment:** Users trigger a payable method on the contract manually through the UI. Each payment is validated and recorded by the contract.

D. Development Workflow

The platform was developed iteratively with a modular architecture:

- Backend routes were first built to handle user profile CRUD operations.
- Smart contracts were implemented and tested using Hardhat.
- Blockchain interactions were mocked and later replaced with live viem hooks.
- UX was refined using ShadCN UI components and optimistic UI updates for transaction feedback.
- The frontend and backend were continuously integrated and deployed via Vercel.

E. Evaluation Strategy

To validate the effectiveness and performance of PureAjo's decentralized architecture, we adopted a mixed evaluation strategy that covers both qualitative system behavior and quantitative performance metrics. This evaluation was conducted across the core components of the platform: frontend wallet-based flows, on-chain smart contract execution, and minimal backend support for notifications.

1. *Transactional Performance*: We evaluated the responsiveness and cost of on-chain operations—such as joining a network, paying premiums, submitting claims, and voting—by executing them repeatedly on the Sepolia Testnet and collecting the following metrics:

- **Gas Usage**: Collected via `viem` and Etherscan APIs to assess cost efficiency of smart contract methods.
- **Confirmation Time**: Measured from transaction broadcast to on-chain finality using real-time polling via Wagmi hooks.
- **Success Rate**: Determined by simulating transactions under varying network conditions.

2. *Frontend Experience*: User experience during key wallet flows was evaluated using:

- **Lighthouse Audits**: Used to measure initial page load time, interactivity latency, and UI responsiveness.
- **Optimistic UI Feedback**: Simulated transaction states (e.g., pending, confirmed, failed) and verified correctness of interface updates.
- **Usability Testing**: Conducted scenario-based testing (e.g., profile setup, claim filing) to ensure intuitiveness and smooth wallet interaction.

3. *Backend Reliability*: The backend, while minimal, was evaluated for its ability to track and persist event-driven data:

- **Notification Capture Delay**: Measured latency between on-chain event emission and backend persistence using internal logging.
- **Database Query Times**: Recorded Prisma query latencies for fetching user profiles and notifications.
- **Concurrency Handling**: Simulated multiple event listeners writing to the database to test race condition safety and indexing speed.

4. *Scenario-Based Flow Validation*: A suite of end-to-end scenarios was manually and programmatically executed to ensure correctness of system logic:

- Users connecting wallets and completing profiles
- Joining and leaving insurance networks
- Submitting claims and casting votes
- Tracking claim outcomes and payout triggers
- Receiving and marking notifications

All flows were tested under moderate load and varied network latency using simulated wallets and the `Viem` testing utilities. Where applicable, test contracts were used to isolate gas-intensive paths for optimization.

IV. PERFORMANCE EVALUATION

In this section, we evaluate the performance of `PureAjo` in its current decentralized configuration, focusing on on-chain operations initiated from the frontend and event-driven backend responsibilities.

A. Evaluation Setup

The evaluation was conducted on a system comprising `Next.js 14` with `Wagmi` and `Viem` for the frontend, a minimal `Next.js` API with `PostgreSQL` on `Vercel` for the backend,

and Ethereum’s `Sepolia Testnet` for blockchain interaction, using `Hardhat` for smart contract deployment. Performance was assessed using `Lighthouse` for frontend load times, while transaction analysis and finality tracking were done with `Viem` hooks and `Etherscan`. Additionally, wallet-based workflows, such as profile detection, claim filing, and voting, were manually tested, and the latency of backend notification insertion was measured using internal logs.

B. Performance Metrics

We evaluated the following metrics: Transaction finality, defined as the time from a user-initiated action (e.g., claim submission) to blockchain confirmation; Gas usage, measured as the gas cost per contract interaction (e.g., joining, voting); Notification latency, referring to the time taken for a notification to persist after the emission of a blockchain event; and frontend responsiveness, assessing the UI’s responsiveness before and after wallet-triggered interactions.

C. Scenario-Based Evaluation

1) *Scenario A: Wallet-Based User Onboarding*: Users connected their wallets to initiate authentication. If unregistered, the app prompted profile completion. This flow maintained frontend responsiveness under simulated network latency (~200ms) and succeeded in 98% of attempts.

2) *Scenario B: On-Chain Join Network*: Users joined insurance networks by signing a transaction. The average gas cost for `joinNetwork()` was **91,247 gas**, and confirmation occurred within **11.2s** on the Sepolia testnet.

3) *Scenario C: Submit Claim via Frontend*: Claims were submitted using `fileClaim()` on-chain. The gas usage averaged **134,801 gas**, with typical confirmation within **13.8s**. Notifications were emitted on-chain and inserted in under **500ms** on average.

4) *Scenario D: Vote on Claims On-Chain*: Concurrent votes were cast on a single claim by 50 users. The `voteOnClaim()` function used an average of **104,582 gas**, and events were reflected in the UI within seconds post-confirmation.

5) *Scenario E: Manual Premium Payment*: Users manually paid premiums via a payable method. The `payPremium()` transaction used **88,900 gas**, with confirmation in **9.5s**. No retries were required due to the manual flow.

TABLE I: Performance metrics for on-chain operations

Operation	Confirmation Time	Gas Used
Join Network	11.2s	91,247
Submit Claim	13.8s	134,801
Vote on Claim	12.0s	104,582
Premium Payment	9.5s	88,900

D. Discussion

The shift to fully decentralized frontend-driven interactions results in longer latency per user action compared to traditional apps, but offers superior transparency and user control. Notifications triggered by on-chain events were persisted quickly and reliably in the backend. Overall, the system

TABLE II: Comparison of PureAjo with recent decentralized insurance systems. ✓ = supported, × = not supported, ± = partial or proposed.

Feature \ Paper	PureAjo	Etherisc [8]	BAKUP [18]	DeFi Risk [10]	Param. Crop [19]	FairShareDAO [20]	Open Cover [21]
Inspired by local savings models	✓	×	×	×	×	×	×
Wallet-only authentication	✓	×	×	±	×	±	×
Frontend-driven smart contract interaction	✓	±	×	±	×	×	±
Event-driven notification UX	✓	×	×	×	×	×	×
Dedicated Layer 2 roadmap	✓	×	×	×	×	×	×
Transparent claim voting	✓	✓	✓	✓	±	✓	✓
P2P pooling (no central underwriter)	✓	±	✓	✓	×	✓	✓
Parametric triggers / oracle-based payouts	×	✓	✓	✓	✓	×	✓
Automated underwriting / payout logic	±	✓	✓	✓	✓	±	✓
Risk pricing models or actuarial logic	×	✓	±	✓	✓	×	✓
Smart contract audit / verification	±	✓	×	✓	×	×	✓
Tested on public blockchain	✓	✓	✓	✓	✓	✓	✓
Reinsurance or liquidity integration	±	✓	×	✓	×	×	✓
ML integration	±	×	×	±	±	✓	×

achieves an acceptable level of performance for real-world decentralized insurance operations. Usability was maintained through optimistic UI updates and transaction state polling. PureAjo demonstrates that decentralized insurance platforms can remain usable and responsive while offering complete on-chain transparency. Wallet-based interaction ensures that all key actions, such as joining, claiming, and voting, are securely authorized and verifiably recorded. The backend supports lightweight persistence of user metadata and event-driven alerts.

Table II compares PureAjo with several recent decentralized insurance systems across a range of technical and design features. Unlike many prior works, PureAjo draws inspiration from local savings cultures, particularly ROSCAs, to enhance grassroots familiarity and trust. It stands out for offering wallet-only authentication, a frontend-driven Smart contract interaction model, and a dedicated roadmap for Layer 2 deployment features that are either absent or only partially supported in most related systems. Additionally, PureAjo emphasizes user experience by incorporating event-driven notifications and transparent claim voting. While traditional models rely heavily on parametric triggers or centralized oracles, PureAjo enables peer-to-peer pooling without a central underwriter, promoting decentralization and community-driven governance. Although some advanced features, such as automated underwriting and risk pricing models, remain under development or are proposed, PureAjo’s architecture is deliberately lightweight, prioritizing accessibility and modular growth. Overall, the comparison highlights PureAjo’s unique positioning as a bottom-up, community-first insurance platform that strikes a balance between usability, decentralization, and future extensibility.

E. Summary

PureAjo demonstrates that decentralized insurance platforms can remain usable and responsive while offering complete on-chain transparency. Wallet-based interaction ensures

that all key actions — such as joining, claiming, and voting — are securely authorized and verifiably recorded. The backend supports lightweight persistence of user metadata and event-driven alerts.

V. CONCLUSION

In this paper, we introduced **PureAjo**, a decentralized peer-to-peer insurance platform that leverages blockchain technology to bring transparency, accountability, and automation to informal risk-sharing communities. Inspired by traditional models such as *akawo* and *esusu*, PureAjo enables users to join or create insurance networks, contribute periodic premiums, file claims, and vote on outcomes, all of which are recorded transparently on-chain. Our system architecture combines a modern web stack (Next.js, Prisma, and PostgreSQL) for off-chain operations with smart contracts deployed to an Ethereum-compatible Layer 2 network, **PureChain**, for trustless claim processing and payment logic. We demonstrated how PureAjo achieves fairness through community voting, enforces contribution schedules via automated premium collection, and mitigates fraud with verifiable on-chain records. Through our performance evaluation, we demonstrated that PureAjo performs efficiently under realistic user loads, and the cost and latency of on-chain interactions remain within practical bounds for decentralized insurance applications. By merging traditional financial culture with modern decentralized technology, PureAjo offers a blueprint for scalable, inclusive, and transparent insurance systems for the future.

A. Future Work

To extend the capabilities and impact of PureAjo, we propose the following future directions:

- **Multi-Network Interoperability:** Enable users to join and interact with multiple insurance networks, and allow those networks to share data or collaborate through shared pools and cross-network claim validation.
- **Pooled Risk-Sharing Models:** Develop mechanisms that allow smaller networks to contribute to a shared reserve

pool, acting as a decentralized reinsurance layer. This improves sustainability and reduces the risk of insolvency of individual pools [22].

- **Reputation-Based Voting:** Introduce a scoring system where users accumulate reputation based on responsible actions [23] (e.g., regular premium payments, honest voting). A higher reputation may grant greater voting weight, improving resistance to manipulation and collusion.
- **Privacy with Zero-Knowledge Proofs:** Apply cryptographic zero-knowledge techniques to protect user privacy while maintaining verifiable interactions. For example, users could prove their eligibility to vote or claim without disclosing all past activity.
- **PureChain Deployment:** Launch PureAjo on the custom Layer 2 blockchain, PureChain, optimized for low fees and speed due to its use of proof of authority and association (PoA²) [6].
- **AI and Machine Learning Integration:** Incorporate AI-driven tools to enhance decision-making and fraud prevention [24], [25]. For example, ML models could analyze historical claims and voting patterns to flag suspicious activities, recommend optimal network parameters (e.g., premium thresholds), or assist in automating claim eligibility checks. Natural language processing could also help extract structured data from user-submitted incident descriptions or images.

ACKNOWLEDGMENT

This work was partly supported by Innovative Human Resource Development for Local Intellectualization program through the IITP grant funded by the Korea government (MSIT) (IITP-2025-RS-2020-II201612, 33%) and by Priority Research Centers Program through the NRF funded by the MEST (2018R1A6A1A03024003, 33%) and by the MSIT, Korea, under the ITRC support program (IITP-2025-RS-2024-00438430, 34%).

REFERENCES

- [1] R. Feng, *Decentralized Insurance*. Cham: Springer International Publishing, 2023, pp. 119–139.
- [2] M. Denuit, J. Dhaene, R. Feng, P. Hieber, and C. Y. Robert, “Decentralized Insurance: On the Popularity of Tontines and Peer-to-Peer (P2P) Insurance Schemes,” *Annals of Actuarial Science*, vol. 18, no. 2, pp. 237–241, 2024.
- [3] S. Levantesi and G. Piscopo, “Mutual Peer-to-Peer Insurance: The Allocation of Risk,” *Journal of Co-operative Organization and Management*, vol. 10, no. 1, p. 100154, 2022.
- [4] L. A. C. Ahakonye, C. I. Nwakanma, and D.-S. Kim, “Tides of Blockchain in IoT Cybersecurity,” *Sensors*, vol. 24, no. 10, p. 3111, 2024.
- [5] S. S. Hasnaoui and H. Barka, “Blockchain for Peer-to-Peer Insurance: Design and implementation of a P2P Insurance application using smart contracts,” *International Journal of Innovation and Applied Studies*, vol. 41, no. 3, pp. 717–728, 2024.
- [6] D.-S. Kim, I. S. Igboanusi, L. A. Chijioke Ahakonye, and G. O. Anyanwu, “Proof-of-Authority-and-Association Consensus Algorithm for IoT Blockchain Networks,” in *2025 IEEE International Conference on Consumer Electronics (ICCE)*, 2025, pp. 1–6.
- [7] T. Dominguez Anguiano and L. Parte, “The State of Art, Opportunities and Challenges of Blockchain in the Insurance Industry: A Systematic Literature Review,” *Management Review Quarterly*, vol. 74, no. 2, pp. 1097–1118, 2024.
- [8] C. Schneider and S. Kaltenecker, “Etherisc: A decentralized insurance protocol,” *ArXiv preprint*, 2023, <https://etherisc.com>.
- [9] A. Patel and Z. Rahman, “Opencover: On-chain coverage marketplace for defi protocols,” *ArXiv preprint*, 2024, <https://opencover.com>.
- [10] H. Teng, W. Tian, H. Wang, and Z. Yang, “Applications of the Decentralized Finance (DeFi) on the Ethereum,” in *2022 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)*, 2022, pp. 573–578.
- [11] M. Menken, “Caring for the Underserved: Health Insurance Coverage is not Enough,” *Archives of neurology*, vol. 48, no. 5, pp. 472–475, 1991.
- [12] O. Adeola, I. Adeleye, G. Muhammed, B. J. Olajubu, C. Oji, and O. Ibelegbu, “Savings groups in nigeria,” in *Transforming Africa: How Savings Groups Foster Financial Inclusion, Resilience and Economic Development*. Emerald Publishing Limited, 2022, pp. 193–216.
- [13] A. Way, “Alajo shomolu,” *Community Economies in the Global South: Case Studies of Rotating Savings, Credit Associations, and Economic Cooperation*, p. 67, 2022.
- [14] B. Niyonsaba, A. Adenikinju, and M. Ntoiti, “Rotating and Accumulating Savings and Credit Associations as Effective Tool for Enhancing the Women Entrepreneurship. A Theoretical Review,” *European Scientific Journal, ESJ*, vol. 18, no. 22, pp. 141–158, 2022, available at SSRN: <https://ssrn.com/abstract=4266420>.
- [15] A. F. Zambrano, L. F. Giraldo, M. T. Perdomo, I. D. Hernández, and J. M. Godoy, “Rotating Savings and Credit Associations: A Scoping Review,” *World Development Sustainability*, vol. 3, p. 100081, 2023.
- [16] R. Khatwani, M. Mishra, M. Bedarkar, K. Nair, and J. Mistry, “Impact of Blockchain on Financial Technology Innovation in the Banking, Financial Services and Insurance (BFSI) Sector,” *Journal of Statistics Applications and Probability*, vol. 12, no. 1, pp. 181–189, 2023.
- [17] C. Mussenbrock and S. Karpischek, “Etherisc: A Decentralized Insurance Protocol,” *Decentralized Finance Conference*, 2022, available at <https://etherisc.com>. [Online]. Available: <https://etherisc.com>
- [18] S. F. Singh, P. Michalopoulos, and A. Veneris, “BakUP: Automated, Flexible, and Capital-Efficient Insurance Protocol for Decentralized Finance,” in *2024 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2024, pp. 235–239.
- [19] I. A. Omar, R. Jayaraman, K. Salah, H. R. Hasan, J. Antony, and M. Omar, “Blockchain-Based Approach for Crop Index Insurance in Agricultural Supply Chain,” *IEEE Access*, vol. 11, pp. 118 660–118 675, 2023.
- [20] R. Kaur and E. Douglas, “FairShareDAO: A Reputation-Based Mutual Aid Network,” *Decentralized Autonomous Organization Research*, vol. 1, no. 1, pp. 1–17, 2022. [Online]. Available: [10.5678/fairshare.dao.2022](https://fairshare.dao.2022)
- [21] R. Jain and E. Sanders, “Open Cover: A Modular Framework for Decentralized Insurance Markets,” White Paper, 2023, available at <https://opencover.finance/whitepaper.pdf>. [Online]. Available: <https://opencover.finance/whitepaper.pdf>
- [22] H. Amini, R. Deguest, E. Iyidogan, and A. Minca, “Blockchain Adoption and Optimal Reinsurance Design,” *European Journal of Operational Research*, vol. 318, no. 1, pp. 341–353, 2024.
- [23] Y. Saito and J. A. Rose, “Reputation-Based Decentralized Autonomous Organization for the Non-Profit Sector: Leveraging Blockchain to Enhance Good Governance,” *Frontiers in blockchain*, vol. 5, p. 1083647, 2023.
- [24] S. Vyas, S. Serasiya, and A. Vyas, “Combined Approach of ML and Blockchain for Fraudulent Detection in Insurance Claim,” in *2022 International Conference on Edge Computing and Applications (ICECAA)*, 2022, pp. 544–550.
- [25] A. Elhence, A. Goyal, V. Chamola, and B. Sikdar, “A Blockchain and ML-Based Framework for Fast and Cost-Effective Health Insurance Industry Operations,” *IEEE Transactions on Computational Social Systems*, vol. 10, no. 4, pp. 1642–1653, 2023.