Metasurance: A Blockchain-Based Insurance Management Framework for Metaverse

Aritra Bhaduri[1][a], Ayush Kumar Jain[1], Swagatika Sahoo[1,2][b], Raju Halder[1][c]
and Chandra Mohan Kumar[1]
[1] *Indian Institute of Technology Patna, India*
[2] *Kalinga Institute of Industrial Technology, India*

Keywords: Metaverse, Virtual Assets, Insurance, Blockchain, Hyperledger Fabric.

Abstract: The worlds of commerce, business, entertainment, education, and healthcare are set for a transition into the Metaverse, enabling people to socialize, shop, invest, manufacture, buy, and sell in the virtual world. This paradigm shift introduces a myriad of risks and threats to the virtual assets, unveiling new avenues for the insurance marketplace to thrive. This paper presents Metasurance, a blockchain-based decentralized platform that enables insurance organizations in crafting and administering tailored insurance products for various virtual assets across different Metaverse platforms. Our solution supports automated management of the complete life cycle, starting from insurance shopping and purchase, premium payments, maturity and claim settlement without any hassle by establishing an interoperability among different Metaverse ecosystems. Moreover, we leverage dynamic price prediction through federated learning, enabling insurance companies to optimize premiums effectively. We present our working prototype developed based on the Hyperledger Fabric blockchain platform, supported by empirical evidence from system benchmarks and load testing, demonstrating enhanced transaction throughput. To the best of our knowledge, this is the first proposal for an insurance solution within the Metaverse ecosystem.

1 INTRODUCTION

In the ever-evolving landscape of technology, the Metaverse's emergence as a symbol of digital innovation highlights its transformative potential spanning over various sectors, including banking, education, e-commerce, entertainment, business, and many more (Wang et al., 2023). In essence, the Metaverse represents a digital universe where individuals immerse themselves in virtual experiences, social interactions, and economic activities. It serves as a bridge between the physical and digital worlds, offering endless possibilities for creativity, collaboration, and innovation. As per the report (met,), Metaverse market is projected to reach a value of US$74.4 billion in 2024 and is expected to grow at an annual growth rate (Compound Annual Growth Rate (CAGR) 2024-2030) of 37.73%, resulting in a projected market volume of US$507.8 billion by 2030. Moreover, by 2024, it is anticipated that there will be over 34 million virtual reality (VR) headset installations worldwide, with 1.7 billion mobile augmented reality (AR) users globally. As a result, such dynamic nature of this environment may introduce various risks and threats where accidents or losses loom. Few examples include cyber-attack, failure due to technical glitches, server downtime, physical and mental health (both real body and virtual avatar), and unintentional infringement of real-world rights (Di Pietro and Cresci, 2021). These emphasize the importance for Metaverse residents to safeguard their virtual assets from unforeseen events, highlighting the need for tailored insurance coverage.

Furthermore, the inherent complexities of the Metaverse demand a blockchain-based (Nakamoto, 2019) insurance system which offers unparalleled security and efficiency through the utilization of smart contracts. These self-executing contracts not only automate claims processing but also ensure a level of transparency and trust that traditional insurance mechanisms often struggle to attain. As we navigate this epoch of the Metaverse, it becomes evident that insurance, augmented by blockchain technology, is

[a] https://orcid.org/0009-0000-8352-9994
[b] https://orcid.org/0000-0002-8572-9348
[c] https://orcid.org/0000-0002-8873-8258

Bhaduri, A., Jain, A., Sahoo, S., Halder, R. and Kumar, C.
Metasurance: A Blockchain-Based Insurance Management Framework for Metaverse.
DOI: 10.5220/0012722100003687

In *Proceedings of the 19th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2024)*, pages 190-201
ISBN: 978-989-758-696-5; ISSN: 2184-4895

poised to play a pivotal role in shaping the future of risk management in the digital realm.

1.1 Motivation and Contributions

Even though there have been a number of blockchain-based insurance systems (Amponsah et al., 2021; Brophy, 2020; Kar and Navin, 2021; Popovic et al., 2020; Raikwar et al., 2018; Kalsgonda and Kulkarni, 2022; Hassan et al., 2021; Loukil et al., 2021) in the literature, they have not addressed the unique challenges of the Metaverse, including: (1) Metaverse is very new - lack of understanding among insurers, insured, and products to cover in the metaverse, (2) highly dynamic pricing behaviour of the metaverse assets, (3) interoperability - capability to connect with multiple metaverse platforms, (4) requirement of universal ID for metaverse objects across the platforms, (5) identity management for the participants and information flow, (6) achieving scalability through a careful design of the platform (on-chain vs. off-chain components), and (7) secure payment system.

Now, let us explore potential failure scenarios within the Metaverse, prompting users to seek insurance coverage for potential losses:

- **Virtual Asset Loss.** Users may face loss or theft of virtual assets like in-game items or digital currencies due to hacking or unauthorized access. Insurance policies such as virtual asset insurance offer financial recovery by compensating for the lost items.
- **Decentralized Finance (DeFi) Risks.** Participants in DeFi activities within the Metaverse are exposed to risks like smart contract exploits or liquidity pool failures. Insurance solutions like DeFi Risk Insurance provide a safety net, ensuring financial security in such scenarios.
- **Network Downtime or Technical Failures.** Technical glitches or network downtime can disrupt user experiences in the Metaverse. Insurance options like Network Downtime Insurance offer financial recovery, allowing users to navigate disruptions confidently.
- **Virtual Property Damage.** Events causing virtual property destruction or damage can lead to financial threats. Insurance options such as Virtual Property Insurance compensate users for the loss, empowering them to innovate within the virtual realm.
- **Marketplace Fraud.** Virtual marketplaces may involve fraudulent activities risking financial losses. Insurance solutions like Marketplace Fraud Insurance provide coverage, fostering trust and security among users.
- **Identity Theft in the Metaverse.** Instances of identity theft within the Metaverse can compromise user accounts and assets. Identity theft protection in insurance policies, like Identity Theft Insurance, ensures a secure Metaverse experience.
- **Virtual Events and Experiences Cancellations.** Unforeseen cancellations of virtual events or experiences can lead to financial losses. Insurance policies offer coverage for such cancellations, encouraging users to explore the Metaverse confidently.
- **Cross-Metaverse Transactions.** Users engaging in transactions across different Metaverse platforms may encounter complexities and risks. Insurance coverage provides assurance, allowing users to navigate cross-Metaverse transactions confidently.
- **Regulatory Changes and Compliance Risks.** Evolving regulations may pose legal risks to Metaverse activities. Insurance policies act as legal allies, offering protection and support amidst changing regulatory environments.

To achieve comprehensive coverage, our research advocates for tailored insurance products for Metaverse assets. Motivated by the identified risks, our research aims to leverage blockchain technology for the following desired goals:

- **Enhanced Security and Transparency.** Implement a blockchain-based insurance system to ensure secure and transparent record-keeping, reduce fraud, and enhance trust in the Metaverse insurance ecosystem.
- **Efficient Claims Processing.** Utilize blockchain's decentralized nature for efficient and tamper-resistant claim processing, ensuring a streamlined and trustworthy mechanism for users to access insurance benefits.
- **Flexibility and Adaptability.** Leverage blockchain's flexibility to design insurance policies that can adapt to the evolving risks and complexities of the Metaverse, offering users tailored and up-to-date coverage.
- **Smart Contract Automation.** Employ smart contracts to automate insurance processes, enhancing efficiency and reducing the likelihood of errors in policy execution by eliminating untrusted third party.
- **Seamless Interactions among Various Metaverse Platforms.** Design a blockchain-based insurance system that is compatible across various

Metaverse platforms, providing users with seamless coverage in a multi-platform environment.

To summarize, this paper makes the following contributions:

1. We propose Metasurance, a novel approach which provides insurance solutions to adeptly address and mitigate the emerging risks of Metaverse by leveraging the power of blockchain technology. By introducing this system into the dynamic Metaverse landscape, users gain assurance regarding the security of their digital assets. Insurance companies and third-party verifiers collaborate to assess and compensate for losses incurred in the Metaverse, mirroring real-world insurance mechanisms.
2. Our proposed system captures the entire spectrum of activities which encompasses registration, policies marketplace, policy purchase, paying premiums, claiming policies, claim verification, claim settlement, and interoperability using tokens. This ensures interactive experiences for customers and facilitates streamlined claim processing for insurers.
3. We present our working prototype using Hyperledger Fabric and NodeJS. The empirical evidence acquired from system benchmarks and load testings is encouraging, and shows us the effectiveness of such a framework in the Metaverse setting.

The structure of the paper is organized as follows: The related work are discussed in Section 2. The detailed descriptions of our proposed approach are presented in Section 3. Section 4 provides discussion on dynamic price prediction using Federated Learning (FL), highlighting its significance in our framework. The communication process with various Metaverse platforms for asset verification is discussed in Section 5. The security threats and their possible countermeasures are discussed in Section 6. We present the proof-of-concept and its detailed experimental evaluation in Sections 7 and 8. Finally, Section 9 concludes our work.

2 RELATED WORK

There have been a number of proposals (Amponsah et al., 2021; Brophy, 2020; Kar and Navin, 2021; Popovic et al., 2020; Raikwar et al., 2018; Kalsgonda and Kulkarni, 2022; Hassan et al., 2021; Loukil et al., 2021) which explored the potential of blockchain to revolutionize trusted insurance frameworks. Anokye et al. in (Amponsah et al., 2021) conducted a comprehensive analysis of blockchain's implications for the insurance sector, examining both its advantages and potential threats. In (Brophy, 2020), Brophy explored blockchain's role in insurance from commercial and regulatory perspectives. Kar et al. in (Kar and Navin, 2021) discussed blockchain's pivotal role in addressing scalability and adoption challenges in the insurance sector. Popovic et al. in (Popovic et al., 2020) provided guidance on blockchain for actuaries, risk professionals, and insurance companies, detailing its use cases. In (Raikwar et al., 2018), Raikwar et al. designed a blockchain-enabled platform for processing insurance transactions with an experimental prototype on Hyperledger Fabric. Kalsgonda et al. in (Kalsgonda and Kulkarni, 2022) proposed a research framework and overviewed Hyperledger Fabric's use cases in insurance. The authors in (Hassan et al., 2021) introduced a framework leveraging smart contracts on a private Ethereum network for insurance contracts. Loukil et al. in (Loukil et al., 2021) presented CioSy, a collaborative blockchain-based insurance system for monitoring and processing transactions.

There are some proposals (Sedkaoui and Chicha, 2021; Demir et al., 2019; Liu et al., 2021; Nizamuddin and Abugabah, 2021; Bader et al., 2018; Roriz and Pereira, 2019; Pagano et al., 2019; Sharifinejad et al., 2020), designed for specifically for providing insurance for certain application domains, such as flight (Sedkaoui and Chicha, 2021), automobile (Vo et al., 2017; Demir et al., 2019; Liu et al., 2021; Nizamuddin and Abugabah, 2021; Bader et al., 2018; Roriz and Pereira, 2019) and more (Pagano et al., 2019; Sharifinejad et al., 2020). Sedkaoui et al. in (Sedkaoui and Chicha, 2021) introduced Axa's Fizzy platform for blockchain-based travel insurance. In (Vo et al., 2017), authors proposed a blockchain solution for managing data in pay-as-you-go car insurance systems. Demir et al. in (Demir et al., 2019) proposed a tamper-free ledger for motor vehicle insurance records. Liu et al. in (Liu et al., 2021) proposed a blockchain-based auto insurance data-sharing scheme. In (Nizamuddin and Abugabah, 2021), Nishara et al. developed a decentralized framework for regulating automobile insurance claims. Bader et al. in (Bader et al., 2018) presented a smart contract-based platform for car insurance. The authors in (Roriz and Pereira, 2019) addressed fraud prevention in vehicle insurance using Ethereum. Pagano et al. in (Pagano et al., 2019) outlined a methodology for blockchain-based digital insurance contracts against natural hazards. Sharifinejad et al. in (Sharifinejad et al., 2020) demonstrated blockchain's applicability in smart city insurance, showcasing reduced delays compared to conventional methods.

3 METASURANCE: PROPOSED BLOCKCHAIN BASED INSURANCE MANAGEMENT FRAMEWORK

This section elucidates our proposed blockchain-based insurance management framework, called Metasurance, for various Metaverse assets including Virtual Lands, NFTs, Gadgets, and Avatars. Metasurance involves a number of stakeholders, such as users, insurers, and third-party claim verifiers, and it hosts a set of smart contracts offering services such as registration, policy initiation, purchase, claim, and verification. Figure 1 depicts the overall system components of the proposed Metasurance, which comprises the following phases: (1) Stakeholder registration, (2) Adding assets/creation of policies, (3) Purchasing policies, (4) Paying premiums, (5) Claim request, (6) Request verification, and (7) Claim approval.

We use a number of smart contracts in the proposed framework, as follows: (1) `UserSc`: This smart contract us used to register a user who wishes to get insured for his assets, (2) `InsurerSc`: This smart contract us used to register a insurer who wishes to provide insurance services, (3) `VerifierSc`: This smart contract us used to register a insurer who wishes to verify various claim requests, (4) `PolicySc`: This smart contract is used by Insurers to create their policy schemes and by users to view all available policies, (5) `AssetSc`: This smart contract is used by users to register their blockchain assets, (6) `PolicyUserMappingSc`: This smart contract is used by users to register their asset with a policy, and pay premiums of the policy, (7) `ClaimSc`: This contract is used by users to claim a policy in case of any damage of the asset that is covered in the insurance, (8) `TokenSc`: This smart contract serves as the currency and carries out the transactions between the different parties.

Let us now provide a detailed description of each of above-mentioned phases.

3.1 Registration

In order to access the system, stakeholders initiate the registration process through their respective contracts `UserSc`, `InsurerSc`, `VerifierSc`. This step enables the involved parties (User, Insurer and Verifier) to formally register themselves, acquiring the necessary credentials for subsequent authentication procedures. Once registered, the users, verifiers, and insurers gain extended access to the system's functionalities, allowing them to register assets and create policies, respectively. Note that each peer node is equipped with a unique cryptographic key pair via trusted authority (say, certificate authority), ensuring the security of all transactions within the system. Let us now describe the registration phase for stakeholders, policies and virtual assets.

3.1.1 Stakeholder Registration

The stakeholder registration consists of three phases, namely set up, key generation and authentication, for user $\mathcal{U}$, verifier $\mathcal{V}$, and insurer I. Let us now discuss each of these steps in detail.

- **Setup.** The setup algorithm works in a manner similar to (Goyal et al., 2006). This phase is initiated by trusted authority $\mathcal{TA}$ (e.g. Certificate Authority). Initially, $\mathcal{TA}$ selects a security parameter λ'. It then selects two multiplicative cyclic groups G_1, G_2 of prime order p, where $p \geq 2^{\lambda'}$. Let $e : G_1 \times G_1 \to G_2$ be a bilinear map and g be the generator of G_1.
- **Keygen.** $\mathcal{TA}$ runs keygen algorithm (Shamir, 1985) to generate secret keys for insurer I, verifier $\mathcal{V}$, and user $\mathcal{U}$. It chooses a random unique numbers (i.e. $id \in Z_p$) for each of the insurers, verifiers, and users as their unique identities (BLAKLEY, 1979). Next, $\mathcal{TA}$ randomly selects $r, y \in Z_p$ and computes $KU_0 = g^{y+r}(id)^r, KU_1 = g^r$. Finally, it returns secret key $SK = \{KU_0, KU_1\}$ to I, $\mathcal{V}$, and $\mathcal{U}$. Then, for public key computation, $\mathcal{TA}$ computes $PK = e(g,g)^y$.
- **Verify and Signing Phase.** After verification of the identity of the stakeholders through proper KYC documents, the $\mathcal{TA}$ signs the public key PK of the corresponding I, $\mathcal{V}$, and $\mathcal{U}$, and publishes it on the ledger. The secret key SK is then sent to the respective I, $\mathcal{V}$, and $\mathcal{U}$ through a secret channel.

3.1.2 Policy Registration

In this phase, the registered insurers can use their login credentials to log back into their respective accounts and finally, based on their organizational principles, they can register various policies related to the virtual assets. There are several field values which are necessary and need to be provided by the insurers based on which policies can only be listed for the further processes. These field values include: (1) unique

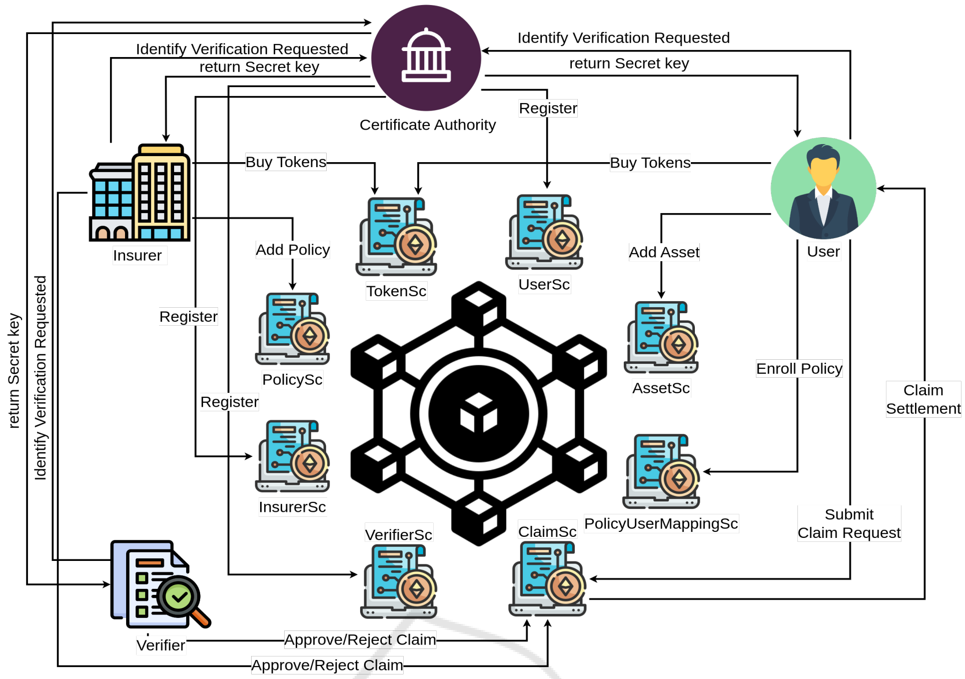

Figure 1: Overview of the proposed system.

policy ID p_{id}, (2) policy name p_{name}, (3) policy type p_{type}, (4) insurer $p_{insurer}$, (5) insurance coverage $\mathcal{A}_{inc}$, (6) premium amount $\mathcal{A}_p$, (7) claims per year $\mathcal{T}$, and (8) policy owner p_{own} (which is set 'NULL' initially). Each policy has a unique policy ID (p_{id}) that is securely generated using randomized UUID generators (Leach et al., 2005). The algorithm for policy registration in PolicySc smart contract is detailed in Algorithm 1. The policy can encompass additional parameters such as custom terms and conditions mandated by the company, coverage limits, exclusions, cancellation provisions, and endorsements. These elements are intentionally excluded from the current solution for the sake of simplicity. However, they can be seamlessly incorporated if the specific use case necessitates their inclusion.

Algorithm 1: PolicyRegistration.

```
Data: $p_{name}, p_{type}, \mathcal{A}_{inc}, \mathcal{A}_p, \mathcal{T}, I$
Result: Policy ID $p_{id}$
$r = \text{random}();$
$p_{id} = \text{IdGen}(r);$
$p_{own} = \text{NULL};$
$p_{insurer} = I;$
$\text{createPolicy}(p_{id}, p_{name}, p_{type}, p_{insurer}, \mathcal{A}_{inc}, \mathcal{A}_p, \mathcal{T}, p_{own});$
return $p_{id}$;
```

3.1.3 Asset Registration

In Metaverse, users can own multiple digital assets such as virtual lands, avatars, NFTs, Gadgets, etc. While these assets have different roles in the virtual ecosystem, they are prone to various kinds of risks and threats, such as cyberattack, failure due to technical glitches, server downtime, physical and mental health (both real body and virtual avatar), and unintentional infringement of real-world rights. Keeping these facts in mind, Metaverse residents therefore need to protect their digital assets with appropriate insurance products. In order to avail this service, the users first need to register their virtual assets through AssetSc smart contract for which they intend to purchase insurance products. This requires the following details: (1) asset name a_{name}, (2) asset owner a_{own}, (3) asset type a_{type}, (4) asset value a_{val}, and (5) asset-age a_{age}. Once the process is done, asset gets registered and is ready for further process. The asset registration algorithm is similar to policy registration and is depicted in Algorithm 2.

Algorithm 2: AssetRegistration.

```
Data: $a_{name}, a_{own}, a_{type}, a_{val}, a_{age}, \mathcal{U}$
Result: Asset ID $a_{id}$
$r = \text{random}();$
$a_{id} = \text{IdGen}(r);$
$a_{own} = \mathcal{U};$
$\text{createAsset}(a_{id}, a_{name}, a_{own}, a_{type}, a_{val}, a_{age});$
return $a_{id}$;
```

3.2 Policies Marketplace

In this phase, users can easily check out different insurance policies from various insurers. Our interface allows for easy filtering based on the supported asset types and the specific insurers providing appropriate policy. Registering for a policy is contingent on its alignment with the asset type, and successful registration necessitates the payment of a purchase fee. The querying policies based on different filters is shown in Algorithm 3. The details of all policies are stored in the levelDB database which is refered to as $\mathcal{DB}$ in algorithm.

Algorithm 3: SearchPolicy.

```
Data: Search Keywords: Asset name $x$, Policy type $y$, Insurer-ID $I$
Result: List of policies found
$i\_list = \mathcal{DB}$.find({$insurer = I$});
$p\_list = i\_list$.filter({$type$ = y});
return $p\_list$;
```

3.3 Policy Purchase

As mentioned earlier, the smart contract `PolicySc` houses a comprehensive list of insurance policies related to the Metaverse world. To sign up for a policy, user utilizes the function AssignPolicy in the smart contract `PolicyUserMappingSc` by providing the *policyID* (p_{id}) and *assetID* (a_{id}). The user is then required to pay the initial amount which will be the first premium for the insurance, which is a mandatory amount. This action internally generates a *policyMap* structure instance with unique ID m_{id}, indicating the association of the new policy p_{id} with the user's asset a_{id}, along with additional attributes 'premium paid' (p_paid), 'claim count' (c_count), and 'claim amount' (c_amt) initialized to 0. The resulting structure is then stored in an array linked to the user's $\mathcal{U}$. The policy purchase algorithm is depicted in Algorithm 4.

Algorithm 4: PolicyPurchase.

```
Data: $a_{id}$, $p_{id}$, $\mathcal{U}$
Result: Policy-Asset Map ID $m_{id}$
asset = getAsset($a_{id}$);
policy = getPolicy($p_{id}$);
if ($asset.a_{own}$ != $\mathcal{U}$)
    exit;
$p\_paid = 0, c\_count = 0, c\_amt = 0$;
success = PayAmount(asset, policy, amount);
if (!success)
    exit;
$r$ = random();
$m_{id}$ = IdGen($r$);
AssignPolicy($m_{id}, p_{id}, a_{id}, p\_paid++, c\_count, c\_amt$);
return $m_{id}$;
```

3.4 Paying Premiums

Upon acquiring a policy for an asset, users engage in a streamlined premium payment process. This approach incorporates token utilization at the time of payment, enhancing security and efficiency. Users are presented with flexible premium payment options, ranging from installment plans to convenient one-time payments, all tailored to the insurer's proposed model. This novel premium payment system mirrors traditional insurance frameworks while incorporating advanced features for an enhanced user experience. Within the comprehensive *user* data model, meticulous tracking of premium payments is ensured through a dedicated '*premium paid*' counter. This counter serves as a reliable indicator, keeping users informed about the number of premiums already settled. The algorithmic steps are depicted in Algorithm 5.

Algorithm 5: PremiumPayment.

```
Data: $m_{id}$, $\mathcal{U}$
Result: Confirmation of payment
$\langle p_{id}, a_{id} \rangle$ = GetState($m_{id}$, $\mathcal{U}$);
asset = getAsset($a_{id}$);
policy = getPolicy($p_{id}$);
success = PayAmount(asset, policy, amount);
if (!success)
    exit;
$m_{id}.p\_paid++$;
return success;
```

3.5 Policy Claim, Verification, and Settlement

Upon purchasing a policy and fulfilling of premiums according to policy agreement, users are able to initiate a claim. The process involves furnishing necessary details and securely storing documents through the InterPlanetary File System (IPFS) [1], as depicted in Figure 3. The details required for initiating a claim request include (1) issuance id m_{id}, (2) user ID $\mathcal{U}$, (3) issuing insurer ID I, and (4) claim evidence CE. Initiating a claim prompts the request to be sent to I and subsequently to an independent verifier $\mathcal{V}$, as depicted in Algorithms 6 and 7. These entities autonomously assess the claim, verify claim evidences and uploads verification-reports on IPFS, deciding to accept or reject. Accepted claims proceed to the settlement phase, where the issuing insurer determines and automatically adds the settlement amount to the user's account using backend chaincode logic. The main steps in claim settlement, depicted in Figure 2,

[1] https://ipfs.tech/

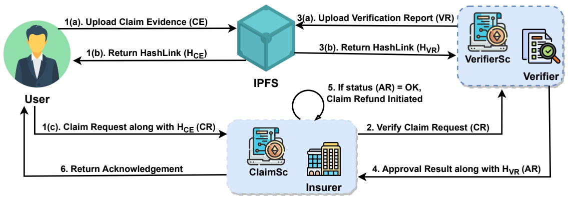
1(a). Upload Claim Evidence (CE)
1(b). Return HashLink (H_{CE})
User
IPFS
3(a). Upload Verification Report (VR)
3(b). Return HashLink (H_{VR})
VerifierSc
Verifier
5. If status (AR) = OK, Claim Refund Initiated
1(c). Claim Request along with H_{CE} (CR)
2. Verify Claim Request (CR)
6. Return Acknowledgement
4. Approval Result along with H_{VR} (AR)
ClaimSc
Insurer

Figure 2: Flow diagram of policy claim, verification, and settlement.

are: (1) User uploads claim evidence (CE) to IPFS and receives the hash link, (2) User calls the `ClaimSc` contract to submit the claim, providing all the necessary details given above with the hash link as the claim evidence, (3) The contract stores the details as a submitted claim request, and waits for a verifier to verify, (4) Verifier verifies the claim request off chain and uploads a Verification Report (VR) to IPFS, receives the hash link and sends the approve/decline result along with the report link, and (5) Insurer, on approval of claim request, processes the transaction and sends the claimed amount decided by the verifier in its report to insured user with a claim acknowledgement.

Algorithm 6: ClaimPolicy.

```
Data: $m_{id}$, $\mathcal{U}$, $I$, $CE$
Result: Claim Request
$\langle p_{id}, a_{id} \rangle$ = getState($m_{id}$, $\mathcal{U}$, $I$);
asset = getAsset($a_{id}$);
policy = getPolicy($p_{id}$);
$H_{CE}$ = IPFSupload($CE$);
$CR$ = generateClaim(asset, policy, $\mathcal{U}$, $I$, $H_{CE}$);
Call ProcessClaim($CR$, $\mathcal{U}$, $I$);
```

Algorithm 7: ProcessClaim.

```
Data: $CR$, $\mathcal{U}$, $I$
Result: Claim settlement
$\mathcal{V}$ = selectVerifier($CR$);
$AR$ = verifyClaimRequest($CR$, $\mathcal{V}$)                ;
    /* Verificationperformed by $\mathcal{V}$ using VerifierSc smart
    contract */
if status($AR$) == OK and $CR.claimed$ == false
    SendAmount($I$, $\mathcal{U}$, $CR.Claim\_Amount$)               ;
    /* Send amount $CR.Claim\_Amount$ from $I$ to $\mathcal{U}$.  */
    $CR.claimed$ = true;
```

4 FL-DRIVEN DYNAMIC PRICING PREDICTION FOR INSURANCE PRODUCTS IN THE METAVERSE

Navigating the complexities of pricing in the Metaverse is akin to charting unexplored territory. Unlike physical products, whose values tend to remain relatively stable, Metaverse assets (such as virtual land, cryptocurrencies, etc.) are prone to rapid and unpredictable fluctuations. This presents a formidable challenge for insurance companies striving to set consistent premiums and claims costs to offer reliable services to users.

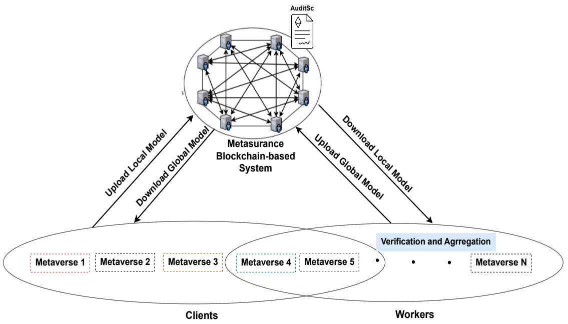

Figure 3: FL framework and global model training process for dynamic pricing prediction.

Our system addresses this challenge by establishing a collaborative framework wherein insurance firms collaborate with various Metaverse platforms to tailor premiums and claims costs. Leveraging federated learning (FL), our system facilitates seamless collaboration and information exchange across diverse Metaverse platforms while safeguarding data confidentiality and privacy.

Within our architecture, as depicted in Figure 3, multiple Metaverse platforms operate independent blockchain networks and train local neural networks. Acting as an intermediary, our blockchain-based platform audits and supervises the interaction process (parameter exchange) between clients and workers, ensuring transparency and accountability in information exchange. Clients, which include Metaverse platforms, upload their local model parameters to the Metasurance platform, while workers, also comprising Metaverse platforms, download these local model parameters for processing. On the worker side, after aggregating and verifying the local model parameters, workers upload the resulting global model parameters back to the Metasurance platform. Subsequently, clients download these global model parameters from the Metasurance to inform their decision-making processes.

Through this seamless exchange facilitated by our Metasurance platform, our system ensures the efficient aggregation and verification of local model parameters, enabling accurate predictive modeling. This process empowers insurance companies to make informed decisions on premiums and claims costs, instilling confidence in the Metaverse economy while paving the way for sustainable growth and innovation within the digital landscape.

5 INTEROPERABILITY AMONG METAVERSE PLATFORMS AND METASURANCE

In the realm of verifying user claims concerning Metaverse assets, establishing connections with the respective Metaverse platforms is crucial for validation. Our system must seamlessly integrate with these platforms, which calls for the adoption of interoperability protocol solutions. The hash-locking protocol (Dai et al., 2020) emerges as a promising solution for this task.

Through the hash-locking protocol, tokens representing ownership or attributes of Metaverse assets can be securely exchanged between our framework and the Metaverse platforms. When a verification request is initiated by an insurance company, our system can generate a unique hash value based on the relevant asset information. This hash value is then locked within a token along with additional metadata, ensuring the integrity and authenticity of the verification process.

Upon receiving the token from our framework, the Metaverse platform verifies the hash value to ensure its consistency with the asset information stored on the platform. Once validated, the token is unlocked, granting access to the requested asset details or confirming its ownership. This mechanism not only ensures the security of the verification process but also provides a tamper-proof method for validating Metaverse assets.

So, this protocol offers a robust solution for achieving interoperability using tokens and validating through tokens in the context of Metaverse asset verification. By leveraging cryptographic hashing and secure token exchange mechanisms, our framework can establish a reliable and tamper-proof method for verifying Metaverse assets, enhancing the integrity and trustworthiness of insurance claims within the Metaverse ecosystem.

6 SECURITY ANALYSIS

This section highlights the core security and privacy features of our Metasurance framework. Designed to resist potential threats, our approach ensures that only authorized users can access and communicate within the system securely. In the dynamic Metaverse landscape, attackers may exploit vulnerabilities, making a robust security infrastructure crucial.

The following discussions delve into specific attacks and the robust solutions implemented to counteract them effectively.

- Eavesdropping Attack: To counter eavesdropping attacks, the Metasurance framework implements end-to-end encryption. This ensures that sensitive information, such as user credentials and policy details, remains confidential during transmission. The use of cryptographic protocols protects against unauthorized interception of data, providing a secure communication channel.
- Data Manipulation Attacks: The Metasurance framework safeguards against data manipulation attacks through the use of blockchain technology. Immutable and transparent ledger records ensure that once data is added to the blockchain, it cannot be altered without consensus. This feature enhances the integrity of critical information, such as policies, claims, and transactions.
- Token System Vulnerabilities: The token system embedded in the `TokenSc` contract of the framework prioritizes security as its core design principle. Employing advanced cryptographic techniques, it safeguards both the generation and validation processes of tokens. Periodic security assessments are diligently carried out to pinpoint and rectify any potential vulnerabilities.

By integrating these security measures, the Metasurance framework establishes a robust defense against a spectrum of potential threats, ensuring the safety and confidentiality of user interactions and data within the dynamic Metaverse environment.

7 PROOF OF CONCEPT

In this section, we provide an in-depth overview of the prototype implementation for our platform. The prototype comprises three key components: (1) the blockchain network, (2) the Fabric backend, and (3) the client application. For the implementation of our blockchain solution, we opted for Hyperledger Fabric v1.4, with GoLang serving as the programming language for our smart contract applications. In the backend, the client application utilizes the Hyperledger Fabric, NodeJS SDK, and communication between the backend and the client interaction occurs through REST APIs, employing the HTTP protocol. We have used the IPFS to store the documents uploaded by users for making policy claims. Figure 4 depicts a concise representation of the system architecture. The organizations we have created are (1) User, (2) Insurer and (3) Verifier. For every organization, we provide a single peer and certificate authority (CA) node. Each organization also has a couchDB instance running as

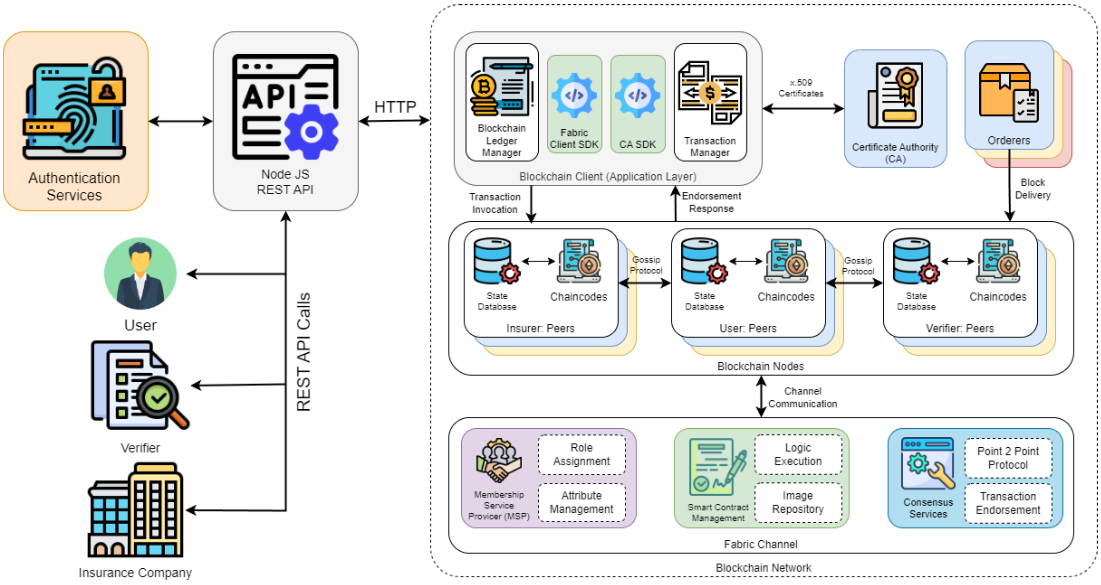

Figure 4: Proof-of-Concept System Architecture.

the state database. The blockchain has a orderer organization with the raft ordering service. Here, we consider a single instance of the peer node and a single channel through which the peers interact. The authentication module is there for insurers and users and uses cookie-based authentication. The cookie can be generated using any algorithm like JSON Web Tokens (JWT), and here we used a randomized salt-based hashing algorithm to generate session tokens from usernames that can be stored in any database like MySQL, PostgreSQL, or MongoDB throughout the session. The same can even be done with JWT.

8 EXPERIMENTAL RESULTS

Let us now manifest the experiments we conducted to quantify and evaluate the performance of the prototype implementation of our proposal. In order to evaluate, we cover a range of experiments where we measure the read-write output of various operations. All the experiments were performed on a laptop with AMD Ryzen 5 7530U processor, 8 GB RAM, and Ubuntu 23.10. We perform these tests using the Hyperledger Caliper benchmarking tool[2]. We use the following performance metrics[3] in our benchmarking process, defined below:

- Send rate $r_s = \frac{\tau_{sent}}{t}$, where τ_{sent} is the number of transactions sent and t is the time in which all of them were submitted to the blockchain.
- Transaction throughput $\eta = \frac{C_{commit}(t)}{t}$, where $C_{commit}(t)$ is simply the number of transactions committed to blockchain at time t.
- Transaction latency $\lambda = t_{cnf} - t_{sub}$, where t_{cnf} is the confirmation time of a transaction and t_{sub} is the submit time of a transaction.

We evaluate the transaction latency and throughput of our system consisting of various transactions such as readUser, getPolicies, getAssets, viewIssuedPolicies, and payPremium. Let us discuss the experimental findings for both Read and Write operations pertaining to these transactions. Figure 5 illustrates our assessment of the system's throughput and latency during the readUser transaction, focusing on reading user profiles under high send rates. We observe that a linear increase in both the send rate and throughput over time. However, at higher send rates (above 300), they stabilize, indicating minimal change despite further increases in send rates. Notably, the peak throughput for reading user profiles occurs at a rate close to 300 transactions per second (TPS).

In Figure 6, we evaluate the system's performance while continuously increasing the rate of requests from 1 TPS to up to 300 TPS during getPolicies transaction. We observe that the throughput initially increases, but then slows the rate of increase after getting to 300 TPS. Still, the rate keeps increasing slightly but mostly remains constant with minimal fluctuations.

In Figure 7, we evaluate the system's performance while continuously increasing the rate of requests

[2] https://hyperledger.github.io/caliper

[3] https://www.hyperledger.org/learn/publications/blockchain-performance-metrics

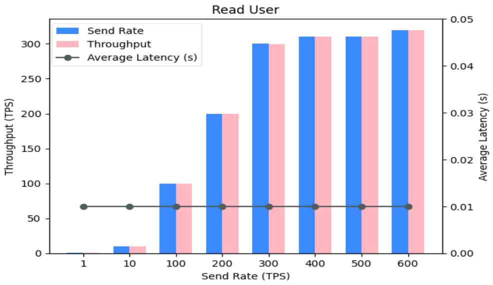

Figure 5: readUser transaction performance.

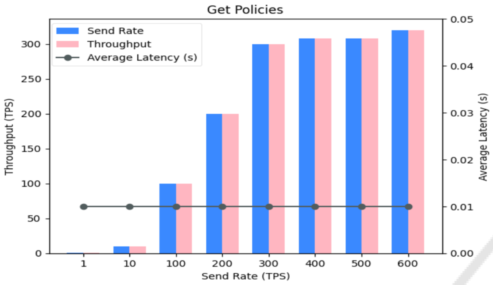

Figure 6: getPolicies transaction performance.

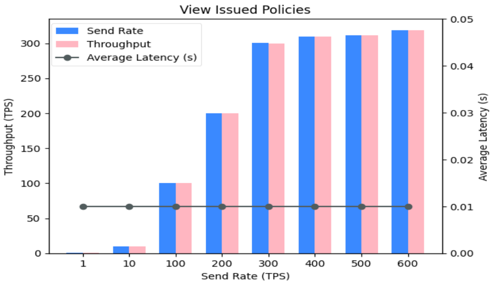

Figure 8: viewIssuedPolicies transaction performance.

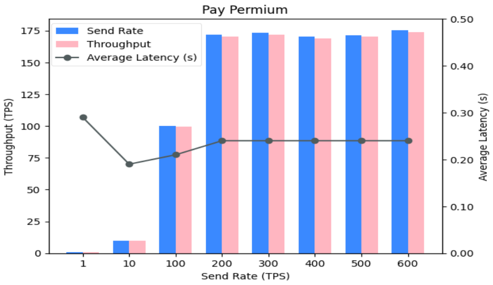

Figure 9: payPremium transaction performance.

from 1 TPS to upto 300 TPS during getAssets transactions. We observe that the throughput initially increases, but it begins to slow down once it reaches 300 TPS. Despite some fluctuations, the rate mostly stabilizes with minimal changes, albeit slightly higher than the throughput depicted in Figure 6.

Figure 8 shows a similar performance evaluation while we are increasing send TPS from 1 to 300 for viewIssuedPolicies transaction. Here also, the send rate and throughput coincide with each other and become almost flat after 300 TPS.

Let us delve into Write operations. In Figure 9, we scrutinize the payPremium performance, which assesses how users pay premiums for their policies. Initially, both the send rate and throughput escalate steadily up to 100 TPS. However, beyond this threshold, although the throughput continues to climb, it does so at a slower pace until it reaches around 175 TPS. Here, the network reaches its maximum capacity, resulting in a stabilized throughput thereafter. Despite a slight dip in throughput at the 400 TPS send rate, the overall change is minimal when increasing the send rates beyond 200 TPS. This decline in throughput compared to read operations can be attributed to the need for exclusive locks on shared states during write operations, slowing down transaction processing in the blockchain. Additionally, examining the latency plots, we observe that while Figures 5, 6, 7, and 8 maintain relatively constant latency, Figure 9 initially experiences a latency dip as the send rate increases, eventually stabilizing to an almost horizontal level.

Finally, we assess the performance of the claimPolicy by gradually increasing the send rate. Notably, the network's throughput shows significant fluctuations throughout this evaluation. Initially, it ascends steadily until reaching 100 TPS, after which the rate of increase diminishes slightly but continues up to 175

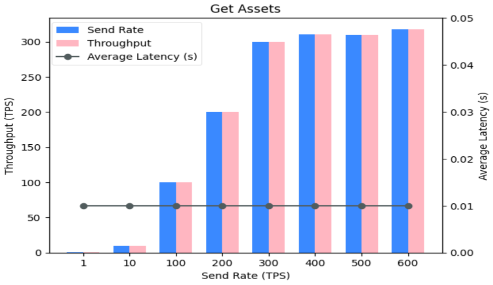

Figure 7: getAssets transaction performance.

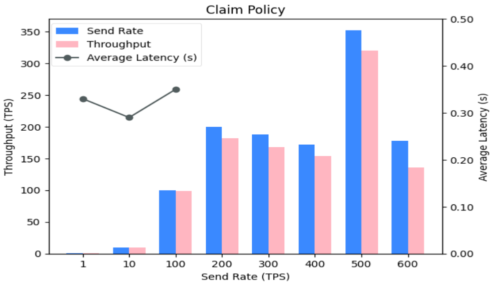

Figure 10: claimPolicy Benchmark.

TPS. Subsequently, network congestion leads to a decline in both the send rate and throughput, even as the send rate reaches 400 TPS. However, as the network congestion eases, the throughput begins to rise again before eventually declining once more after reaching its peak. This observed trend is depicted in Figure 10.

9 CONCLUSION

In this paper, we introduce Metasurance, a blockchain-driven decentralized platform designed to empower insurance organizations for Metaverse ecosystem. This platform facilitates the creation and management of insurance products specifically tailored for Metaverse assets across diverse platforms. We demonstrate a proof-of-concept based on Hyperledger Fabric, by systematically designing and implementing various smart contracts. Additionally, we undertake experiments utilizing Hyperledger Caliper, a performance benchmark framework, to meticulously assess the performance of our system. Our findings conclusively demonstrate the feasibility, efficiency, and cost-effectiveness of our proposed system. While our current implementation does not encompass the FL and interoperability components, we view them as integral parts of our forthcoming roadmap.

ACKNOWLEDGEMENT

This research is partially supported by the research grant provided by IIT Bhilai Innovation and Technology Foundation (IBITF).

REFERENCES

Metaverse - worldwide [online]. Available: https://www.statista.com /outlook/amo/metaverse/worldwide.

Amponsah, A. et al. (2021). Blockchain in insurance: Exploratory analysis of prospects and threats. *International Journal of Advanced Computer Science and Applications*.

Bader, L. et al. (2018). Smart contract-based car insurance policies.

BLAKLEY, G. R. (1979). Safeguarding cryptographic keys. In *1979 International Workshop on Managing Requirements Knowledge (MARK)*, pages 313–318.

Brophy, R. (2020). Blockchain and insurance: a review for operations and regulation. *Journal of financial regulation and compliance*, 28(2):215–234.

Dai, B. et al. (2020). Research and implementation of cross-chain transaction model based on improved hash-locking. In *Blockchain and Trustworthy Systems*, pages 218–230. Springer.

Demir, M. et al. (2019). Blockchain Based Transparent Vehicle Insurance Management. In *2019 Sixth International Conference on Software Defined Systems (SDS)*, pages 213–220.

Di Pietro, R. and Cresci, S. (2021). Metaverse: Security and privacy issues. In *2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, pages 281–288.

Goyal, V. et al. (2006). Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 89–98. Acm.

Hassan, A., Ali, M. I., Ahammed, R., Khan, M. M., Alsufyani, N., and Alsufyani, A. (2021). Secured insurance framework using blockchain and smart contract. *Scientific Programming*, 2021:6787406.

Kalsgonda, V. and Kulkarni, R. (2022). Role of blockchain smart contract in insurance industry. *Available at SSRN 4023268*.

Kar, A. K. and Navin, L. (2021). Diffusion of blockchain in insurance industry: An analysis through the review of academic and trade literature. *Telematics and Informatics*, 58:101532.

Leach, P. J. et al. (2005). A Universally Unique IDentifier (UUID) URN Namespace. RFC 4122.

Liu, X. et al. (2021). A blockchain-based auto insurance data sharing scheme. *Wireless Communications and Mobile Computing*, 2021:1–11.

Loukil, F. et al. (2021). Ciosy: A collaborative blockchain-based insurance system. *Electronics*, 10(11).

Nakamoto, S. (2019). Bitcoin: A peer-to-peer electronic cash system. Technical report, Manubot.

Nizamuddin, N. and Abugabah, A. (2021). Blockchain for automotive: An insight towards the ipfs blockchain-based auto insurance sector. *International Journal of Electrical and Computer Engineering*, 11:2443–2456.

Pagano, A. J. et al. (2019). Implementation of blockchain technology in insurance contracts against natural hazards: A methodological multi-disciplinary approach. *Environmental and Climate Technologies*, 23.

Popovic, D. et al. (2020). Understanding blockchain for insurance use cases. *British Actuarial Journal*, 25:e12.

Raikwar, M., Mazumdar, S., Ruj, S., Sen Gupta, S., Chattopadhyay, A., and Lam, K.-Y. (2018). A Blockchain Framework for Insurance Processes. In *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–4.

Roriz, R. and Pereira, J. L. (2019). Avoiding insurance fraud: A blockchain-based solution for the vehicle sector. *Procedia Computer Science*, 164:211–218.

Sedkaoui, S. and Chicha, N. (2021). Blockchain-based smart contract technology application in the insurance industry: The case of "Fizzy".

Shamir, A. (1985). Identity-based cryptosystems and signature schemes. In Blakley, G. R. and Chaum, D., editors, *Advances in Cryptology*, pages 47–53, Berlin, Heidelberg. Springer Berlin Heidelberg.

Sharifinejad, M. et al. (2020). BIS- A blockchain-based solution for the insurance industry in smart cities. *CoRR*, abs/2001.05273.

Vo, H. T. et al. (2017). Blockchain-based data management and analytics for micro-insurance applications. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2539–2542.

Wang, H. et al. (2023). A survey on the metaverse: The state-of-the-art, technologies, applications, and challenges. *IEEE Internet of Things Journal*, 10(16):14671–14688.