RULE 110 – “NET” TOKEN WHITE PAPER

Version: 1.0
Date: January 31, 2025

---

1. General Information About the Issuer

Name of Issuer: Rule 110, Inc. (“Rule 110” or the “Company”)
Registered Address: 131 CONTINENTAL DRIVE SUITE 305, NEWARK, New Castle, DE 19713
Jurisdiction of Incorporation: Delaware, U.S.
Website: https://realitynet.xyz
Contact: wyatt@realitynet.xyz

1.1 Corporate Purpose

Rule 110, Inc. (“Rule 110” or “we” or “us”) is dedicated to the research, development, and provision of blockchain-based software solutions, including the **Reality Network** (“Network”). The Network is designed to enable participants to operate nodes, host decentralized Reality Applications (“rApps”), and use and earn the Network’s utility token, **$NET** (the “Token,” or “NET Token”).

1.2 Management & Governance

The Company’s leadership comprises individuals with experience in distributed systems, cryptography, and decentralized governance. **Founder & CEO**, **Wyatt Meldman-Floch**, oversees strategic direction and product development, drawing upon experience in blockchain architecture and cryptographic systems. He has a strong background in software engineering, focusing on machine learning, distributed systems, and functional programming. Prior to founding Reality Network, Wyatt was the Chief Technology Officer at Constellation Network, Inc., where he led the development of scalable blockchain solutions. He holds a Bachelor of Science degree in Physics and Mathematics and has contributed to research in numerical modeling and unsupervised learning. Wyatt's expertise in decentralized technologies and his leadership in innovative projects underscore his commitment to advancing blockchain infrastructure. **Keenan Thompson** serves as the **Co-Founder** of Reality Network. He brings a wealth of experience in blockchain technology and decentralized systems. His strategic vision and leadership have been instrumental in guiding the company's mission to create accessible and efficient blockchain solutions. Keenan's background includes significant roles in technology development and project management within the blockchain industry, contributing to the advancement of decentralized applications and platforms. His formal background is in Machine

Learning, receiving a masters from National Taiwan Normal University and undergraduate degree from The College Of William And Mary.

1.3 Regulatory Status

Rule 110 operates on the basis that the NET Token is intended as a **pure utility token** that grants access to, and governance participation in, certain functionalities of the Reality Network. While Rule 110 closely monitors the evolving crypto-asset regulatory frameworks (including MiCA in the EU), we do **not** guarantee that the NET Token will be deemed outside the scope of local securities or other financial regulations in all jurisdictions. You are encouraged to seek independent legal advice on any applicable laws to which you may be subject.

1.4 Financial Standing

- **Funding**:
    - In the prior year, Rule 110 successfully closed an **angel round** of financing for approximately **USD 2 million**.
    - The Company is currently engaged in a **pre-seed round** targeting up to **USD 3 million**, primarily via a SAFE note plus token warrant structure.
- **Audits**: Rule 110 has **not** undertaken any formal financial audits to date.
- **Vesting & Lockups**: Capital raised through equity or token warrants is subject to **standard vesting periods** and lockups of up to four years, with typical cliffs and monthly vesting schedules.

No further financial disclosures are available at this time. This White Paper does not constitute financial or investment advice, and prospective participants should conduct their own due diligence.

---

2. Project Description: Reality Network & the 2MEME Protocol

2.1 Overview of the Network

The **Reality Network** is an experimental blockchain-based platform under development by Rule 110. It leverages the **2MEME consensus protocol** featuring “Proof of Useful Work” (PoUW). This approach aims to minimize the barriers typically associated with traditional Proof of Work or Proof of Stake networks, making node operation more accessible for consumer-grade hardware.

Key Features:

- **Node Operation on Consumer Devices**: Encourages broader participation by allowing individuals to run network nodes on standard hardware.
- **Hosting of rApps (Decentralized Applications)**: Enables nodes to host user-developed rApps and receive usage-based incentives.
- **Staking of $NET**: Nodes that stake $NET increase their likelihood of being selected for consensus snapshots, directly contributing to network throughput and finality.

2.2 2MEME Consensus Protocol

The **2MEME** algorithm implements a **multi-layered consensus architecture**, comprising:

- **L1 (Data Dependency Graph)**: Where block proposals are formed through a "triangulation" approach involving an owner and two facilitators.
- **L0 (Snapshot Layer)**: Periodically collects L1 blocks into "Snapshots" for final confirmation.

2MEME specifically calculates **entropy rates** among nodes, incentivizing them to maintain consistent, correct, and "non-sybil" behavior. By **staking** $NET, nodes can be selected more frequently as **active L0 peers**, receiving potential rewards in $NET for accurate consensus participation.

Further details on the protocol's structure, mathematics, and empirical validation are outlined in Section 7 and the "2MEME White Paper" portion of this document.

2.3 Utility of the NET Token

The NET Token serves multiple functions within the Reality Network:

1. **Access to Node Operation**: Staking $NET allows consumer-grade devices to become "active L0 peers," earning additional token rewards when they produce accurate snapshots.
2. **Resource Allocation for rApps**: Nodes hosting resource-intensive rApps can "stake" more NET Tokens to signal greater capacity, improving throughput for those applications.
3. **Governance Participation**: NET holders may participate in community voting for protocol upgrades, parameter tuning, or feature requests, subject to final on-chain or off-chain processes.

---

3. Legal Classification & Rights Conferred

Nature of Token: The NET Token is a **utility token** designed to facilitate and reward participation in the Reality Network's decentralized infrastructure.

- **No Ownership or Equity Interest**: NET Tokens do **not** represent any equity, shares, partnership interests, or similar ownership rights in Rule 110 or any affiliated entity.
- **No Profit-Sharing**: NET Token holders are **not** entitled to dividends, profit shares, or other financial returns from Rule 110.
- **Functional & Governance Utility**: Token holders may use $NET to stake nodes, pay for certain Network services, and participate in governance proposals (if and when available).
- **Transferability**: NET Tokens are generally transferable on supported wallets and exchanges, subject to compliance with local regulations, Terms of Use, and any lock-up or vesting conditions if applicable.
- **No Redemption Right**: The issuance of NET Tokens does **not** create any redemption or "cash out" obligation on the part of Rule 110.

3.1 Tokenomics & Issuance Schedule

- **Total Token Supply**: A **fixed supply** of 1,000,000,000 (1 billion) NET Tokens.
- **Token Issuance Model**:
    - **7% of total supply** allocated to a block rewards schedule inspired by "Bitcoin halvenings" (gradual reduction in block rewards over time).
    - The remaining supply is reserved for team vesting, community incentives, ecosystem growth, and operational budgets.
- **Vesting & Lockup Periods**:
    - Core team and other early participants are subject to a **4-year** vesting schedule with a typical cliff and monthly vesting.
    - Private sale and pre-seed token allocations may also include lockups, as disclosed in applicable sale agreements.
- **No Burning or Minting Mechanism**: There is **no** plan for token burning or further minting mechanisms beyond the initial supply.

---

4. Risk Factors

Investing in or using the NET Token and Reality Network involves significant **technological, regulatory, and financial** risks. Potential token holders and users should carefully consider the following non-exhaustive list:

4.1 Technological Risks

- **Experimental Consensus**: 2MEME and PoUW are still experimental. There may be bugs or unforeseen vulnerabilities.
- **Cybersecurity Threats**: Malicious actors could exploit smart contract vulnerabilities, conduct Sybil or Eclipse attacks, or compromise node software.

- **Network Forks**: Unforeseen technical disagreements or updates may lead to network forks, splitting token holders across different chains.

4.2 Regulatory & Legal Risks

- **Changing Regulations**: Crypto-asset regulations, including MiCA, may evolve and impose additional registration or licensing obligations on issuers or token holders.
- **Potential Classification**: NET Tokens, while intended as utility tokens, may be deemed securities or e-money in some jurisdictions, subjecting holders and the issuer to additional requirements.
- **Jurisdictional Restrictions**: Some countries may ban or restrict crypto-assets, preventing local residents from participating.

4.3 Market & Financial Risks

- **Price Volatility**: Digital assets are speculative and may experience extreme price fluctuations.
- **Liquidity**: Tokens may be illiquid, with no guarantee of resale.
- **Competition**: The blockchain sector is highly competitive; emergent protocols could supplant Reality Network.

4.4 Node Operator & Staking Risks

- **Staking Lock-up**: Staked tokens may be locked for certain intervals, creating an opportunity cost.
- **Slashing**: Nodes involved in malicious or incorrect consensus activity risk losing part or all of staked tokens (“slashing”).
- **Operational & Hardware Risks**: Running a node on consumer hardware could cause hardware failures or performance bottlenecks.

4.5 Experimental Nature

- **Testnet Disclaimer**: The Reality Network may operate on a testnet basis initially. Testnet $NET may not be redeemable for mainnet tokens at any specified ratio or timeframe.
- **No Guaranteed Mainnet**: Launch of a fully operational mainnet is not assured and remains at the discretion of Rule 110.

You are advised to thoroughly evaluate your financial circumstances, consult professional advisors, and exercise caution before participating in or purchasing NET Tokens.

---

5. Use of Proceeds, Financial Planning & Development Roadmap

5.1 Use of Proceeds & Financial Planning Structure

Rule 110 intends to allocate any funds raised (if any) through the offering or sales of NET Tokens according to an approximate structure as follows:

1. **Research & Development (40% of proceeds)**
   This includes core protocol improvements (2MEME, PoUW, node software) at approximately 15%, security audits and testing at 10%, technical documentation at 5%, and infrastructure development at 10%.

2. **Ecosystem Growth (25% of proceeds)**
   Developer grants may represent around 10%, strategic partnerships 8%, and community-building initiatives 7%.

3. **Operations & Compliance (20% of proceeds)**
   Team salaries and benefits will comprise roughly 10%, with legal and compliance costs at 5%, and general administrative expenses at 5%.

4. **Reserves & Contingencies (15% of proceeds)**
   An emergency fund of around 10% and strategic opportunities fund of 5% to handle unforeseen expenses or pivot needs.

5.2 Business Model & Revenue Structure

Rule 110's **primary revenue streams** may include:

1. **Transaction Fees**: Applied to network usage (estimated 0.1–0.5% per transaction).
2. **Swap Fees**: For cross-chain swaps (estimated 1% per swap).
3. **rApp Token Rewards**: Revenues from partner or third-party tokens generated within the ecosystem.

Cost Structure

- **Fixed Costs**:
    - Core team salaries and benefits
    - Infrastructure maintenance (server costs, hosting fees)
    - Legal and compliance overhead
    - Ongoing research and development

- **Variable Costs**:
    - Network scaling expenses based on growing transaction volume
    - Marketing and community initiatives
    - Developer incentives (bounties, grants)
    - Expanded support and maintenance needs

Financial Projections (5-Year Outlook)

- **Year 1 (illustrative)**:
    - Projected network transactions: ~500,000
    - Estimated revenue: ~USD 1 million
    - Operating costs: ~USD 600,000
- **Years 2–5**:
    - Anticipated annual network growth rate: ~30–50%
    - Aim to diversify revenue streams beyond initial transaction fees
    - Continued scaling of infrastructure and team

All figures above are preliminary estimates and subject to change based on actual network adoption, market conditions, and regulatory developments.

5.3 Development Roadmap (Indicative)

- **Q1 2025**: Finalize testnet updates; optimize node operation for consumer hardware. Expand testnet to public participants; launch initial on-chain governance modules.
- **Q2 2025**: Launch mainnet (subject to testnet stability); introduce $NET Token bridging solutions.
- **Q3 2025**: Integrate third-party rApps; research advanced privacy features & post-quantum cryptography migration.
- **Q4 2025 & Beyond**: Pursue community-driven proposals, ecosystem expansions, and iterative network upgrades.

All roadmap items are estimates and may change in response to technological, market, or regulatory factors.

---

6. Distribution, Exchanges & Marketing

6.1 Exchange Listings

Rule 110 aspires to list the NET Token on reputable crypto-asset exchanges, including Kraken. **No guarantee** is made that any listing will be achieved or maintained. Exchanges typically perform their own due diligence and compliance checks, and have sole discretion in continuing or delisting trading pairs.

6.2 Marketing & Outreach

Marketing efforts include:

- Publishing open-source code repositories and technical documentation;
- Running developer education campaigns and node-operator tutorials;
- Engaging with communities (online and in-person) to encourage rApp creation.

All promotional materials will refrain from guaranteeing returns or offering financial advice. **Compliance** with MiCA and other relevant regulations is paramount.

6.3 Territorial Scope

Rule 110 does not actively market or offer NET Tokens in jurisdictions where crypto-assets are **prohibited** or **restricted** (e.g., certain embargoed countries). Users bear the responsibility of ensuring compliance with local laws before acquiring or using NET Tokens.

6.4 Distribution Mechanism

- **Private Sale**: Ongoing and/or completed offerings pursuant to SAFE notes plus token warrants (or equivalent contractual structures), featuring standard lock-up and vesting schedules.
- **Public Sale & TGE**: Planned for a future date (Q2 2025), contingent on market conditions and regulatory compliance. The **Token Generation Event (TGE)** is projected to launch the token at a price of **$0.125**, implying a fully diluted valuation of **$125 million**. This figure draws upon the market capitalization of comparable projects and the unique advantages of Reality Network technology.
- **Exchange Listings**: After the TGE, $NET may be available on third-party exchanges and potentially on our own network-based DEX or swap functionality.
- **Airdrops**: No formal airdrop plan exists at present.
- **Testnet $NET Redemption**: Participants who earn “testnet $NET” by running nodes may be eligible to redeem them for mainnet $NET after launch. The exact ratio and process remain undecided and will likely be determined through governance.

---

7. Technical Appendix: The 2MEME Protocol

Below is a concise summary of the “2MEME White Paper,” which provides the scientific and technical foundation for the Reality Network’s consensus.

7.1 Multi-Layered Architecture

- **L1 (Data Dependency Graph)**: Blocks are produced via a triangulation approach (owner plus two facilitators) and reference two prior “Tips” to form a DAG structure. Owner-Facilitator proposals unify states across nodes through partial merges.
- **L0 (Snapshot Layer)**: Periodically aggregates L1 blocks into “Snapshots” approved by a majority of active L0 peers. Active L0 peers are chosen partly by stake weighting and partly by random selection.
- **LΩ (Meta-Layer)**: Maintains a peer list, processes join requests, and acts as a block explorer. Malicious or diverging peers risk slashing or removal.

7.2 2MEME: Minimizing Entropy & Self-Avoiding Walk

- **Entropy Rate**: Each node’s data is modeled as a stochastic process; lower entropy indicates alignment with the network’s majority state, reducing the risk of forks and sybil attacks.
- **Self-Avoiding Random Walk**: Employs Monte Carlo simulations to detect anomalies or sybil nodes, removing or penalizing them.

7.3 Incentive Mechanism (Proof of Useful Work)

- **Rewards**: Nodes staking NET and proposing valid blocks or snapshots receive additional NET from an issuance reserve.
- **Slashing**: Nodes proposing invalid data or double spends are penalized through token reduction.
- **Usefulness**: “Useful Work” is linked to a node’s consistency and alignment with the global network state, thereby reducing overall entropy.

For a complete explanation (including formal proofs, pseudocode, and simulations), please refer to the extended “2MEME White Paper” included in this document’s annex.

---

8. Disclaimers

8.1 Experimental Technology Disclaimer

The Reality Network, 2MEME Protocol, and NET Token are highly experimental. By participating in the Network, you acknowledge the possibility of complete or partial loss, including (but not limited to) the loss of staked tokens or total token value.

8.2 Regulatory Disclaimer

This document does **not** constitute an offer of securities or any other regulated financial instrument in any jurisdiction. Rule 110 does not provide **investment advice, brokerage, or banking** services. Compliance with local laws regarding crypto-assets remains solely the user's responsibility.

8.3 Forward-Looking Statements

Any statements about the future (e.g., development roadmap, technical upgrades, and protocol improvements) are **aspirational** and subject to change without prior notice. Actual outcomes may differ materially from current expectations.

8.4 No Warranties

The NET Token, the Reality Network, and all related software are provided "AS IS" and "AS AVAILABLE." **No warranties** of any kind—express or implied—are offered, including those regarding fitness for a particular purpose or non-infringement.

8.5 Limitation of Liability

To the **maximum extent permitted by law**, Rule 110 (including its officers, directors, employees, or agents) shall **not** be liable for any indirect, special, incidental, or consequential damages, including lost tokens, lost profits, or business interruption.

8.6 Terms of Use

Participation in the Network is governed by our [Terms of Use] and **Privacy Notice**, which are **incorporated by reference** into this White Paper. In the event of any conflict between this White Paper and the ToU, the ToU shall prevail concerning usage policies, disclaimers, and liability limitations.

---

9. Governance & Decision-Making

9.1 On-Chain Voting

Certain aspects of the Reality Network's governance mechanisms remain under development. Potential or proposed governance features include:

- **Protocol Upgrades**: Voting on major changes (e.g., transitioning from ECDSA to post-quantum cryptography).

- **Parameter Adjustments**: Modifying block intervals, snapshot frequencies, or reward schedules.
- **Community Grants**: Allocating resources for open-source rApps or other ecosystem expansions.

9.2 Role of the Company

Rule 110 may initially retain administrative controls to update the protocol for security, regulatory, or user-protection reasons. As the Network evolves, **further decentralization** and robust on-chain governance remain key objectives.

---

10. AML, KYC, and Compliance

10.1 Anti-Money Laundering (AML) & Counter-Financing of Terrorism (CFT)

Where applicable, Rule 110 may require prospective NET Token purchasers to complete Know-Your-Customer (KYC) verifications and AML/CFT screenings under relevant legal frameworks.

10.2 Sanctions Compliance

NET Token users must confirm they are **not** located in embargoed or sanctioned regions and are **not** listed on prohibited-party registers. Participation in violation of sanctions law is strictly prohibited.

---

11. Liability & Indemnification

By using the Reality Network, operating a node, or acquiring NET Tokens, you agree (except as prohibited by law) to **indemnify and hold harmless** Rule 110 from any claims, liabilities, losses, or damages resulting from your use or misuse of the Network or breach of this White Paper, the Terms of Use, or the Privacy Notice.

---

12. Environmental Impact Assessment

12.1 Energy Consumption Analysis

The Reality Network's **Proof of Useful Work (PoUW)** consensus mechanism is designed with an emphasis on energy efficiency, distinguishing it from conventional Proof of Work systems:

- **Estimated Power Consumption**
  - **Individual Node (consumer hardware)**: ~50–100 W/hour
  - **Enterprise-Grade Node**: ~200–400 W/hour
  - **Projected Network at Scale**: ~1.2 GWh/year
  - **Comparative Efficiency**: Approximately 99.95% more efficient than traditional PoW networks

12.2 Carbon Footprint

- **Per-Transaction Footprint**: ~2.5 gCO2e
- **Projected Annual Network Footprint**: ~500 metric tons CO2e
- **Comparison to Traditional Financial Systems**: Estimated at ~95% lower carbon intensity than many legacy payment networks

12.3 Sustainability Measures

The Reality Network actively pursues sustainability through:

- **Optimized Node Software**: Reduces computational overhead for standard hardware.
- **Green Hosting Partnerships**: Collaborations with renewable energy providers.
- **Carbon Offset Program**: Offsetting operational emissions via reputable environmental initiatives.
- **Energy Efficiency Rewards**: Encouraging best practices in node operation through potential token incentives.

12.4 Environmental Mitigation Strategies

- **Technical Optimizations**: Intelligent workload distribution, dynamic power scaling, and efficient data storage.
- **Regular Impact Audits**: Periodic assessments of the Network's environmental footprint.
- **Renewable Energy Certifications**: Future plans to introduce official recognition for nodes running on verified green power.
- **Community Sustainability Initiatives**: Encouraging ecosystem participants to propose and vote on improvements that reduce ecological impact.

---

13. Technical Specifications & Security

13.1 Smart Contract Security

- **Audit Schedule**
    - **Q1 2025**: Initial security audit by [Audit Firm]
    - **Q2 2025**: Follow-up audit by [Audit Firm 2]
    - **Q3 2025**: Ongoing monitoring and penetration testing
- **Security Framework**
    - Comprehensive threat modeling to identify possible exploit vectors
    - Regular vulnerability assessments and code reviews
    - 24/7 monitoring for anomalous network activity
    - Formal incident response protocols for critical breaches
- **Bug Bounty Program**
    - **Critical Vulnerabilities**: Up to $250,000
    - **High Severity**: Up to $100,000
    - **Medium Severity**: Up to $25,000
    - **Low Severity**: Up to $5,000

13.2 Node Requirements

- **Minimum Hardware**
    - CPU: 4 cores, 2.5GHz+
    - RAM: 8GB
    - Storage: 256GB SSD
    - Network: 10Mbps stable connection
- **Recommended Hardware**
    - CPU: 8 cores, 3.5GHz+
    - RAM: 16GB
    - Storage: 512GB NVMe SSD
    - Network: 100Mbps stable connection

13.3 Security Measures

- **Access Control**
    - Multi-signature requirements for critical operations
    - Role-based permissions
    - Hardware Security Module (HSM) integration
    - Cold storage for reserve tokens
- **Network Security**
    - DDoS mitigation strategies

    - Node authentication through unique POH technology
    - Encrypted communication channels
    - Automated threat detection with alert systems
- **Emergency Procedures**
    - Network pause mechanism for critical vulnerabilities
    - Automated backup solutions for ledger and node data
    - Comprehensive incident response plan

13.4 Testing Procedures

- **Quality Assurance**
    - Unit test coverage target above 95%
    - Automated integration testing across development phases
    - Load and concurrency testing under varied conditions
    - Security penetration testing (white-box and black-box)
- **Performance Benchmarking**
    - Transaction throughput and latency metrics
    - Resource utilization under stress
    - Chaos engineering experiments to test fault tolerance

---

14. White Paper Updates & Revisions

Rule 110 reserves the right to **amend or update** this White Paper if material changes occur in the:

- **Product Strategy or Protocol Design**
- **Regulatory Developments** affecting the classification or distribution of NET Tokens
- **Token Supply or Distribution Plan**

Updates will be posted on the official website (https://realitynet.xyz) and/or announced via public channels customarily used by the Reality Network community. It is **your responsibility** to periodically review the latest version of the White Paper to remain informed about any revisions.

---

15. Conclusion & Acknowledgment

The **NET Token** and the **Reality Network** represent an ambitious effort to establish **accessible, decentralized, and efficient** blockchain infrastructure. By utilizing the 2MEME consensus

algorithm, we aim to incentivize consistent, low-entropy participation among a diverse array of nodes, mitigating sybil and malicious conduct while offering high-throughput for decentralized applications.

Before obtaining or using NET Tokens, you should:

1. Thoroughly read and understand **this White Paper**, the **Terms of Use**, and any supplementary policies referenced herein.
2. Consider the **risks** outlined in Section 4.
3. Seek **independent advice** regarding your legal, financial, or tax obligations.
4. Recognize that **no guarantees** of value, liquidity, or future performance are provided.

If you do **not** agree with or comprehend these terms, disclaimers, and requirements, you should **refrain** from acquiring, holding, staking, or using NET Tokens.

Signatures & Declarations

Rule 110, Inc.
By: **Wyatt Meldman-Floch**
Title: **Founder & CEO**
Date: **January 31, 2025**

> *"I, the undersigned, declare that, to the best of my knowledge, the information contained in this White Paper is in accordance with the facts and contains no omission likely to affect its import."*

_____

Annex: Full 2MEME White Paper Text

(For brevity, only the key sections are reproduced here. The complete text as provided in the attached material is incorporated by reference.)

> **See the attached "2MEME White Paper" (November 27, 2023 edition by Wyatt L. Meldman-Floch)** for formal algorithmic descriptions, proofs, simulations, and references concerning the underlying consensus engine.

Disclaimer

This **White Paper** is drafted in good faith to meet the **MiCA** requirement for crypto-assets offered within the European Union. **No part of this document** should be construed as investment advice or a recommendation to purchase or sell any securities. **Always consult** independent legal, financial, and tax professionals before engaging in crypto-related transactions.

End of Document

RULE 110 PRIVACY NOTICE

Last Updated: 31 January 2025

1. INTRODUCTION

Welcome to **Rule 110, Inc.** ("**Company**," "**we**," "**us**," or "**our**"). We respect your privacy and are committed to protecting your personal data. This **Privacy Notice** ("**Notice**") explains how we collect, use, share, and protect your personal information when you:

- Visit or interact with our websites (including https://realitynet.xyz/ and any subdomains) (the "**Site**");
- Use our Portal Software, including any downloadable or hosted versions;
- Interact with the **Reality Network** ("**Network**"), whether by running nodes, deploying/using "rApps," or participating in testnets;
- Communicate with us in any way.

By accessing or using our Site, Software, or other related services (collectively, the "**Services**"), you acknowledge that you have read, understood and agree to this Notice. If you do **not** agree, please **do not** access or use the Services.

We may update this Notice from time to time. Any terms not defined herein have the definitions provided in our Terms of Use (the "**Terms**"), which is hereby incorporated by reference. The "Last Updated" date above reflects the most recent revision. Your continued use of the Services after any changes are posted constitutes acceptance of the revised Notice.

---

2. SCOPE OF THIS NOTICE

This Notice applies to personal information ("**Personal Data**" or "**personal information**") that we process in connection with:

- The Site and any official **Portal Software** we provide,
- Any rApps or content that the Company **directly** manages (if applicable),
- Communications with us (e.g., email, customer support),
- Our testnet and related blockchain-based features.

It does **not** apply to third-party rApps or any node operators **not** directly controlled by us, as those parties may collect or process your Personal Data under their own policies and licenses. Please carefully review each rApp developer's privacy policy or license terms.

---

3. INFORMATION WE COLLECT

3.1 Information You Provide Voluntarily

- **Account or Profile Data**: If you create an account with us (e.g., via a web portal), we may collect your name, email address, or other details.
- **Communications**: When you contact us (e.g., support tickets, email inquiries), we collect the information you provide, including your name, email, the content of the message, and any attachments.
- **KYC/AML Data** (If Applicable): If required by law or specific Services, we may request additional identity verification data (e.g., government ID).
- **Feedback or Surveys**: If you participate in optional surveys or provide feedback, we collect the information you submit.

3.2 Information Collected Automatically

When you use our Site or certain parts of our Services:

- **Device & Usage Data**: We collect IP address, browser type, operating system, device IDs, pages visited, time spent on pages, and crash/diagnostic logs.
- **Cookies & Similar Technologies**: We use cookies, beacons, and similar tools to track preferences and analyze trends. See Section 6 for more details.
- **Node Logs** (If Operating Our Nodes): If you connect to nodes we directly operate, we may log connection info (e.g., IP address, timestamps) for debugging and security.

3.3 Blockchain & Testnet Data

- **Blockchain Immutability**: Certain interactions (e.g., transactions, rApp deployments) may be recorded on the Reality Network, a **decentralized** ledger that we do not control. **Once on-chain, data is typically immutable** and publicly accessible.
- **Testnet Environment**: The Reality Network may operate in a “testnet” mode. While we may periodically reset or modify the testnet, data or transactions logged on distributed nodes might still persist in archives or logs. **We do not** guarantee complete erasure of testnet data.

3.4 Children’s Information

Our Services are **not** directed at children under the age of 13 (or 16 in certain jurisdictions). We do not knowingly collect personal data from them. If you believe a child has provided personal information, please contact us at `[Contact Email]`, and we will take steps to delete it as required by law.

---

4. HOW WE USE YOUR INFORMATION

We use the personal information we collect for the following purposes:

1. **Provide and Maintain Services**: To enable node operations, rApp deployment, user accounts, community forums, or other features you request.
2. **Customer Support**: To respond to inquiries, troubleshoot technical issues, and handle customer service requests.
3. **Site and Service Improvements**: To analyze usage, track trends, and enhance or debug our Site, Software, or node services.
4. **Security & Compliance**: To detect, investigate, and prevent fraudulent or unauthorized activities, comply with legal obligations, or enforce our Terms of Use.
5. **Communications**: To send you updates, administrative messages, or marketing (where permitted by law and subject to your opt-out rights).
6. **Legal Obligations**: We may process personal data where necessary to comply with applicable laws (e.g., KYC/AML), respond to lawful requests, or protect our legal rights.

---

5. LEGAL BASES FOR PROCESSING (EU/EEA/UK USERS)

Where the **General Data Protection Regulation (GDPR)** or similar laws apply, we rely on these legal bases:

- **Contractual Necessity**: Processing is required to provide the Services you have requested (e.g., operating your account, enabling node interactions).
- **Legitimate Interests**: Protecting the security of the Network, analyzing usage to improve our Services, preventing fraud, or enforcing our Terms.
- **Consent**: Where required (e.g., certain marketing communications, optional cookies). You may withdraw consent at any time without affecting the lawfulness of processing before withdrawal.
- **Legal Obligations**: Where processing is necessary for us to comply with a legal requirement (e.g., AML regulations).

---

6. COOKIES AND TRACKING TECHNOLOGIES

We (and our third-party analytics/service providers) use cookies, web beacons, and similar technologies to:

- Remember user preferences (like language selection),

- Analyze usage patterns to improve user experience,
- Monitor performance and detect security issues.

You can typically configure your browser to refuse cookies or alert you when cookies are being sent. However, some features of our Site may not function correctly if cookies are disabled.

---

7. SHARING AND DISCLOSURE OF INFORMATION

1. **Third-Party Providers**: We may share limited personal data with vendors or service providers who perform functions on our behalf (e.g., hosting, analytics, payment processing). They are contractually bound to process such data only under our instructions and in compliance with this Notice.
2. **rApps**: Independent developers build rApps on the Network. If you choose to use a third-party rApp, your personal data may be collected, processed, or shared under that rApp developer's own privacy policies. **We do not control** or take responsibility for how rApps handle personal information.
3. **Node Operators**: Node operators in the decentralized Network act independently. They may log connection or transaction metadata for their own purposes. We **cannot** oversee each node operator's data practices.
4. **Business Transfers**: In the event of a merger, acquisition, asset sale, or bankruptcy, personal data may be transferred as part of that transaction, subject to the acquirer honoring existing privacy commitments.
5. **Legal & Regulatory Compliance**: We may disclose personal data if required by law or lawful requests (e.g., subpoenas, court orders), or if we believe in good faith that disclosure is necessary to protect our rights, investigate fraud, or ensure safety.

---

8. DECENTRALIZED NATURE & BLOCKCHAIN IMMUTABILITY

1. **Public Ledger**: **All** transactions, rApp code deployments, and certain user interactions on the Reality Network may be publicly viewable on a decentralized ledger. Once recorded, the data is **immutable** and may be accessible to anyone.
2. **Erasure Limitations**: If you or a rApp developer embeds personal data on-chain, we cannot alter or delete it. This may limit our ability to fulfill data subject rights (like erasure) under GDPR or other privacy laws.
3. **Testnet**: Our Network may run in a testnet environment subject to resets, forks, or modifications. Although we may attempt to wipe or reset testnet data, some node operators or external archives may retain historical data. We make **no guarantee** that testnet data will be fully removed from all decentralized locations.

---

9. INTERNATIONAL DATA TRANSFERS

Your personal data may be stored and processed in countries outside your own, including in jurisdictions that may have data protection laws different from (and potentially less protective than) your home jurisdiction. If we transfer personal data from the EU/EEA or UK to a country that lacks an adequate data protection standard, we implement appropriate safeguards (such as Standard Contractual Clauses) to ensure your data remains protected.

---

10. DATA SECURITY

We implement reasonable administrative, technical, and physical security measures to protect personal data. However, **no method** of transmission over the internet or electronic storage is completely secure. We cannot guarantee absolute security. You remain responsible for safeguarding your own devices and credentials, especially if you store private keys or seed phrases containing personal data.

---

11. DATA RETENTION

We retain personal data as long as is **necessary** for the purposes described in this Notice (e.g., to fulfill Services, comply with legal obligations, enforce agreements). When we no longer have a legitimate purpose for retaining data, we will delete or anonymize it. However, **data written to the blockchain** (on-chain data) cannot be altered or erased due to the technology's immutable nature.

---

12. YOUR RIGHTS (GDPR/STATE PRIVACY LAWS)

Subject to local laws, you may have the right to:

- **Access**: Obtain confirmation about whether we process your personal data and access a copy.
- **Rectification**: Request that we correct any inaccurate or incomplete personal data.
- **Erasure**: Ask us to delete or remove personal data; however, this may **not** extend to data on the decentralized ledger we do not control.
- **Restriction**: Request limited processing of your personal data.

- **Objection**: Object to processing of your personal data if we rely on legitimate interests.
- **Data Portability**: Receive your data in a structured, commonly used format.
- **Withdraw Consent**: If processing is based on consent, you can withdraw at any time.

To exercise these rights, contact us at `[Contact Email]`. We will respond in accordance with applicable laws. We may require you to verify your identity before handling certain requests.

---

13. NODE OPERATORS & rAPP DEVELOPERS

1. **Independent Data Controllers**: Node operators and rApp developers may collect, log, or process personal data independently of Rule 110. They may be considered separate data controllers (or co-controllers) under EU/EEA law. We **do not** control their data practices.
2. **License & Data**: rApp developers are responsible for implementing their own licenses or end-user terms regarding data usage or collection. If you operate a node or use an rApp that processes personal data, you must comply with any applicable data protection laws in your jurisdiction.

---

14. TESTNET-SPECIFIC DATA

1. **Redeemability & Data Logs**: You may earn or interact with testnet $NET tokens, and transactional or wallet data may be logged on-chain. While we may attempt to reset or modify testnet states, certain logs (including personal data) could persist on distributed nodes or in external archives.
2. **No Guaranteed Removal**: We make **no** representation that testnet transaction data will be fully erased or removed from all repositories. Users should exercise caution in submitting personal data on testnet.

---

15. COMPLIANCE WITH LAWFUL REQUESTS & ENFORCEMENT

We may disclose personal data to public authorities if we believe, in good faith, that such disclosure is required by law or regulation (e.g., subpoenas, court orders) or is necessary to investigate illegal or suspicious activities, protect our rights or those of others, or enforce our Terms of Use.

---

16. THIRD-PARTY LINKS & EXTERNAL SITES

Our Site or Services may contain links to third-party websites or rApps not controlled by us. We do **not** endorse and are not responsible for their privacy practices. Please review each third party's privacy policies before providing personal information.

---

17. UPDATES TO THIS NOTICE

We may revise this Notice from time to time. If we make material changes, we will notify you by posting the updated version on our Site and updating the "Last Updated" date. Where required by law, we will seek your consent or give you the opportunity to opt out.

---

18. CONTACT US

If you have any questions, comments, or concerns about this Privacy Notice or our data practices, please contact us at:

Rule 110, Inc.
Attn: Privacy Team
[Insert Mailing Address]
Email: [privacy@realityfoundation.io] (example)

We will make every reasonable effort to address and resolve your inquiries.

Last Updated: 31 January 2025

End of Privacy Notice

RULE 110 TERMS OF USE & DISCLAIMERS

Last Updated: 31 January 2025

1. INTRODUCTION

Rule 110, Inc. ("**Company**," "**we**," "**us**," or "**our**") designs, develops, and maintains the **Reality Network** (the "**Network**")—an experimental blockchain-based system using the **2MEME** consensus protocol featuring "Proof of Useful Work" ("**PoUW**"). Through our Portal Software, websites, and other interfaces (collectively, the "**Services**"), you can:

- Operate Network nodes using consumer devices;
- Develop and host decentralized applications ("**rApps**");
- Participate in governance and earn/use the Network's utility token ("**$NET**") or other digital assets;
- Access additional features, tools, and documentation related to the Network.

These **Terms of Use** ("**Terms**"), together with our **Privacy Notice** and any rules, notices, or policies referenced on our website (https://realitynet.xyz/), form a legally binding agreement (the "**Agreement**") between **you** ("**User**" or "**you**") and the Company. By accessing or using any part of the Services, you **accept** these Terms. If you do **not** agree, you must **not** use the Services.

We may update or modify these Terms at any time. Changes take effect upon posting unless otherwise stated. Your continued use after changes are posted indicates acceptance of the revised Terms.

---

2. ELIGIBILITY & SCOPE

1. **Legal Capacity**. You represent you have the legal capacity to enter into this Agreement. If you are under the age of majority in your jurisdiction, a parent or legal guardian must review and accept these Terms on your behalf.
2. **Entity Representation**. If you use the Services on behalf of an organization, you warrant you have the authority to bind that entity.
3. **Regulatory Compliance**. You are not located in an embargoed or sanctioned region and are not on any prohibited-party lists. You will comply with all applicable laws (including sanctions, export controls, AML/CFT, consumer protection).

4. **Experimental Nature**. You understand the Network relies on experimental blockchain technology, posing elevated risks of bugs, forks, volatility, or potential total loss of digital assets.

---

3. SERVICES & MODIFICATIONS

1. **Definition**. “Services” include our Portal Software, websites, user interfaces, APIs, node operations we control, rApp hosting frameworks, and all features for node operation, governance, or digital asset usage.
2. **Modifications & Forks**. We reserve the right to modify, suspend, or discontinue any part of the Services at any time, with or without notice. This includes protocol changes, updates, forks, or changes to consensus rules. Such modifications may create incompatibilities or require you to upgrade software.
3. **No Liability for Changes**. We are not liable for any losses, disruptions, or incompatibilities arising from modifications, forks, or discontinuation of any feature.
4. **Updates**. We may provide patches or updates; you are responsible for installing them promptly. Using outdated software could risk security or performance.

---

4. PRIVATE KEYS & SECURITY

1. **User Responsibility**. You alone generate, store, and secure your private keys, seed phrases, passwords, or other credentials required to access the Network or Services.
2. **Irreversible Transactions**. Blockchain transactions are typically final once broadcast. If your private keys are compromised, unauthorized parties may irreversibly transfer or destroy digital assets.
3. **No Key Custody**. The Company does not store or manage your credentials and cannot recover lost or stolen private keys.
4. **Security Best Practices**. You agree to implement strong security (e.g., secure backups, hardware wallets, patched systems) and promptly address vulnerabilities.
5. **Disclaimer**. We disclaim all liability for any unauthorized access or credential compromise arising from your failure to maintain proper security measures.

---

5. NODE OPERATOR RESPONSIBILITIES

1. **License to Run rApps**. You acknowledge that rApp developers grant node operators a non-exclusive license to store, reproduce, and execute their rApps for normal operation.

You agree not to exceed that license scope or tamper with rApp code for unauthorized ends.

2. **Prohibited Misuse**. You will not:
    - Alter or reverse-engineer rApp code to exploit, steal data, or break functionalities;
    - Inject malicious code or deliberately corrupt the rApp or node environment;
    - Use your node to circumvent or compromise security features in the rApp or the Network.
3. **User Data**. To the extent you gain access to rApp user data through node operations, you must comply with applicable privacy laws and refrain from any unauthorized use or disclosure.

---

6. PROHIBITED ACTIVITIES

You agree **not** to:

1. Violate any applicable law (e.g., AML, sanctions, export controls) or engage in illegal, deceptive, or fraudulent activities;
2. Host, upload, or distribute content that infringes third-party IP or contains malware, viruses, or harmful code;
3. Manipulate network consensus, disrupt node operations, or otherwise sabotage the Network;
4. Bypass or disable security measures, rate limits, or authentication methods in the Services;
5. Impersonate Company personnel, misrepresent your affiliation with us, or cause confusion about official Services;
6. Engage in any conduct that may harm, disable, or overload the Network, rApps, or other users' access.

We may immediately suspend or terminate your access, and, if necessary, report violations to relevant authorities.

---

7. INTELLECTUAL PROPERTY & LICENSES

7.1 Portal Software & MIT License

Unless explicitly stated otherwise, the source code for the Portal Software is provided under the [**MIT License**]. You may use, copy, modify, publish, and distribute it subject to the MIT conditions. Some components may be licensed differently; where noted, those terms govern exclusively for that component.

7.2 Trademarks

"**Rule 110**," "**Reality Network**," and related names or logos ("**Trademarks**") belong to us. The MIT License does **not** grant you any right to use our Trademarks beyond factual reference to the Software's origin or compatibility. Any other trademark usage requires our prior written consent.

If you develop or deploy a rApp, you grant us a limited, non-exclusive license to use your rApp name, logo, and related marks solely to identify, list, or describe your rApp within our Software, Services and Network. This license is restricted to accurate factual references and does not imply our endorsement of your rApp. We will respect your trademark guidelines if provided, and you may revoke this license for trademark misuse. All goodwill from our use of your marks inures to your benefit.

7.3 rApps & Third-Party Licenses

1. **Deployment License**. By deploying or hosting a rApp on the Network, you (the rApp developer) grant the Company and all Network participants (including node operators) a non-exclusive, worldwide, royalty-free license to store, reproduce, host, and execute the rApp's code and content as required for normal operation and access by end users.
2. **Your rApp Terms**. You may choose any license model (open source or proprietary) for your rApp code, provided it does not conflict with the license in Section 7.3(1) or these Terms. You are solely responsible for providing end users with the rApp's own license terms.
3. **No Endorsement**. We do not endorse, audit, or certify rApp code. We disclaim liability for any legal or regulatory issues arising from a rApp's licensing or content.
4. **IP Warranties**. You represent and warrant that your rApp does not infringe any third-party IP or other rights and that you have all necessary permissions for any included third-party libraries or assets.

7.4 Contributions to Company Repositories

1. **License Grant**. By submitting code, documentation, or other materials ("**Contributions**") to any official Rule 110 repository (e.g., on GitHub), you grant the Company and recipients of the Portal Software an irrevocable, perpetual, worldwide, royalty-free license to use, reproduce, modify, adapt, distribute, display, and create derivative works from those Contributions under the MIT License (or similarly permissive open-source terms).
2. **Representations**. You represent and warrant that you have all necessary rights to submit Contributions and that doing so does not infringe or violate any third-party intellectual property.
3. **Moral Rights Waiver**. To the fullest extent permitted by law, you waive any moral rights in your Contributions.

7.5 User-Generated Content (Non-Code)

Any non-code content (text, images, etc.) you submit remains yours, but you grant us a non-exclusive, royalty-free license to use, display, reproduce, and distribute it as needed to operate or improve the Services. We may remove content that we believe violates these Terms or the law.

---

8. TESTNET DISCLAIMER & REDEMPTION

1. **Ongoing Testnet**. You acknowledge that the Network may currently be operating on a "testnet" basis for experimentation and testing. We may reset, modify, or discontinue the testnet at any time without notice.
2. **$NET Rewards**. You may earn or receive $NET tokens in the testnet environment; **however**, the specific redemption rate or method of converting testnet $NET to any mainnet or production $NET is **not yet determined**.
3. **Redemption is Undecided**. We do not guarantee any particular **value**, **ratio**, or **mechanism** for converting testnet $NET to mainnet $NET (if mainnet launches). We may decide, at our sole discretion, whether or how to allow redemption or migration of testnet $NET.
4. **No Liability**. We disclaim liability for any assumptions regarding the convertibility, ratio, or potential value of testnet $NET. Testnet balances may be wiped, invalidated, or modified, and you assume all risk relating to testnet participation.
5. **No Obligation**. Nothing herein obligates us to launch a mainnet, maintain any specific token economics, or honor any future redemption. **All** decisions on testnet or mainnet issuance remain at our sole discretion.

---

9. DISCLAIMERS & ASSUMPTION OF RISK

1. **Experimental Technology**. The Network, Services, and Portal Software are highly experimental. Bugs, consensus failures, or security exploits may occur. **You accept these risks** and acknowledge the possibility of irreversible loss.
2. **No Warranties**. TO THE FULLEST EXTENT PERMITTED BY LAW, THE SERVICES, PORTAL SOFTWARE, AND NETWORK ARE PROVIDED "AS IS" AND "AS AVAILABLE," WITHOUT WARRANTIES OF ANY KIND—EXPRESS, IMPLIED, OR STATUTORY (INCLUDING MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND NON-INFRINGEMENT) AND ANY OTHER WARRANTIES ARISING BY COURSE OF DEALING OR USAGE OF TRADE. **NO ADVICE OR INFORMATION, WHETHER ORAL OR WRITTEN, OBTAINED BY YOU FROM THE**

COMPANY OR THROUGH THE SERVICES CREATES ANY WARRANTY. NOTHING HEREIN LIMITS ANY WARRANTY THAT CANNOT BE LAWFULLY WAIVED.

3. **Token & Financial Disclaimers**. We do not guarantee the value, stability, or liquidity of $NET or any other digital asset. **We are not** your broker, dealer, or financial advisor. You alone bear responsibility for any investment or tax implications.
4. **No Money Transmitter or Banking Services**. We are **not** a bank, deposit institution, money transmitter, broker-dealer, exchange, or other regulated financial institution under applicable law. The Software, Services and Network do **not** involve exchanging real currencies, e-money, or deposit-taking services. You are solely responsible for determining whether your activities (e.g., running a node, exchanging digital assets) require any license or registration under local laws.
5. **No Fiduciary or Advisory Relationship**. You acknowledge that we do **not** act as your agent, fiduciary, trustee, or advisor. We owe **no** fiduciary or similar duty to you in connection with your use of the Services or any digital assets, and we do **not** provide personalized financial, tax, or legal advice.
6. **Securities Law Disclaimers**. Digital assets, including $NET, may be subject to **complex and evolving securities laws** in the U.S. and other jurisdictions. **We make no guarantees** about the legal classification of tokens. You **alone** are responsible for determining and satisfying any securities or financial regulatory requirements. We may modify or restrict access to the Services at any time to comply with regulatory requirements. You alone are responsible for understanding and complying with any laws or regulations that apply to your use of the Services or any tokens.
7. **Third-Party rApps & Content**. The Company disclaims liability for the security, compliance, or reliability of any third-party rApp, code, or content accessible via the Network. Your interactions with third-party offerings are at your own risk.
8. **Forward-Looking Statements**. Any roadmap or future feature statements are **aspirational** and may change without notice. We have no obligation to deliver specific features by a set timeline.
9. **Release and Waiver of Claims**. **To the maximum extent permitted by law**, you hereby **irrevocably release, waive, and discharge** the Company (including its officers, directors, employees, agents, shareholders, and affiliates) from any and all claims, demands, actions, causes of action, losses, damages, costs, expenses, and liabilities of every kind or nature ("**Claims**"), whether known or unknown, suspected or unsuspected, matured or unmatured, that arise from or relate to your use of or inability to use the Software, Services, or the Network. You **expressly waive** any rights granted under **California Civil Code Section 1542**, or any similar statute, which states: "A general release does not extend to claims that the creditor or releasing party does not know or suspect to exist in his or her favor at the time of executing the release and that, if known by him or her, would have materially affected his or her settlement with the debtor or released party." By accepting these Terms, you acknowledge that you have been advised of the existence and contents of Section 1542 (and any equivalent law in another jurisdiction) and **explicitly waive** any and all rights granted thereunder. **Nothing** in this subsection or elsewhere in these Terms **limits or excludes** any liability that

cannot be limited or excluded **under applicable law**, including liability for our own fraud, willful misconduct, or gross negligence where such liability cannot be waived.

10. **DAO Interactions** (If Applicable). Certain features of the Network may allow participation in or interaction with decentralized autonomous organizations ("DAOs"). Such DAOs operate independently of the Company. We disclaim any liability for decisions, votes, or actions taken by a DAO or its members. You acknowledge that any proposals or governance actions are conducted by the DAO and not by the Company, and you assume all risks associated with such participation.
11. **Governance Participation**. Any governance mechanisms (e.g., on-chain voting for protocol upgrades) are experimental and may result in forks, changes to consensus rules, or other network-altering outcomes. We do not guarantee that governance decisions will be conflict-free or beneficial. You assume all risk arising from or related to governance participation, including any disputes, forks, or incompatible updates.
12. **Bridging & Cross-Chain Interactions**. You may use or rely on third-party bridging technologies or cross-chain tools to transfer assets between the Reality Network and external blockchains. These third-party services are **not** controlled by us, and we disclaim liability for any losses, delays, hacks, or protocol failures associated with them. You assume all risk for bridging or cross-chain transactions.

---

10. LIMITATION OF LIABILITY

1. **Aggregate Liability Cap**. TO THE MAXIMUM EXTENT ALLOWED BY LAW, OUR TOTAL LIABILITY FOR ANY CLAIM ARISING UNDER OR RELATED TO THESE TERMS OR THE SERVICES SHALL NOT EXCEED THE GREATER OF (A) US$100 OR (B) ANY AMOUNT YOU ACTUALLY PAID US (IF ANY) IN THE SIX (6) MONTHS PRECEDING THE EVENT GIVING RISE TO LIABILITY.
2. **Exclusion of Certain Damages**. UNDER NO CIRCUMSTANCES WILL THE COMPANY OR ITS OFFICERS, DIRECTORS, EMPLOYEES, OR AGENTS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL, OR PUNITIVE DAMAGES, INCLUDING LOST PROFITS, LOST DATA, LOST TOKENS, OR LOSS OF GOODWILL, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. NOTHING IN THESE TERMS EXCLUDES OR LIMITS LIABILITY FOR OUR OWN WILLFUL MISCONDUCT, FRAUD, OR GROSS NEGLIGENCE WHERE SUCH LIABILITY CANNOT BE EXCLUDED OR LIMITED UNDER APPLICABLE LAW.
3. **Specific Disclaimers**. We are not liable for security breaches, lost private keys, unauthorized transactions, node malfunctions, or regulatory actions affecting tokens.
4. **Certain Jurisdictions**. Some places do not allow certain liability limitations. In those areas, our liability is limited to the fullest extent permitted by law.

---

11. INDEMNIFICATION

Except to the extent that any such claims or damages arise from our own fraud, willful misconduct, or gross negligence, you agree to **defend, indemnify, and hold harmless** the Company (including its officers, directors, employees, and agents) from any claims, damages, losses, costs, or expenses (including reasonable attorneys' fees) arising out of or related to:

- Your use or misuse of the Services, Network, or rApps;
- Your violation of these Terms or of any applicable law or regulation;
- Any claim that your rApp or content infringes or misappropriates third-party IP;
- Any negligence or misconduct regarding your private keys, node operation, or security measures;
- Disputes between you and any third party concerning tokens, rApps, or user data.

---

12. DISPUTE RESOLUTION & ARBITRATION

1. **Agreement to Arbitrate**. Except where prohibited by local law, any dispute arising from or relating to these Terms or the Services will be **resolved by binding arbitration** administered by the American Arbitration Association under its Commercial Arbitration Rules. Judgment on the award may be entered in any court of competent jurisdiction.
2. **Class Action Waiver**. TO THE FULLEST EXTENT PERMITTED BY LAW, YOU WAIVE ANY RIGHT TO PARTICIPATE IN A CLASS, CONSOLIDATED, OR REPRESENTATIVE ACTION. **YOU ALSO WAIVE** ANY RIGHT TO A JURY TRIAL.
3. **Opt-Out**. You may opt out of arbitration by sending written notice to `[Legal Contact]` within **30 days** of first agreeing to these Terms.
4. **Governing Law & Venue**. These Terms are governed by **Delaware** law, without regard to conflict-of-law provisions. Arbitration will occur in **Deer Lake, Michigan**, unless local law mandates otherwise. Court-based disputes (for non-arbitrable claims) must be brought in Delaware's state or federal courts, unless local consumer laws require a different venue.
5. **One-Year Limitation**. Any claim must be filed within **one (1) year** of the date it first could be brought; otherwise, it is permanently barred.

---

13. MISCELLANEOUS

1. **Force Majeure**. We are not liable for delays or failures caused by events beyond our reasonable control, such as natural disasters, cyberattacks, or government actions.
2. **Assignment**. You may not assign or transfer these Terms without our prior written consent. We may assign or transfer our rights/obligations to an affiliate or successor in a merger, acquisition, or asset sale.

3. **No Agency**. These Terms do not create a partnership, joint venture, fiduciary, or agency relationship.
4. **Notices**. We may provide notices by posting on our website or sending email to your address on file. Notices are effective upon posting or sending. You must keep your contact details accurate.
5. **Waiver & Severability**. A failure to enforce any provision is not a waiver of future enforcement. If any provision is invalid, the remaining provisions remain valid and enforceable.
6. **Entire Agreement**. These Terms (including referenced policies, such as our Privacy Notice) constitute the entire agreement between you and us and supersede prior communications on the subject matter.
7. **Survival**. Terms that by their nature should survive termination (e.g., disclaimers, limitations of liability, indemnities, dispute resolution) will survive.

---

14. ACKNOWLEDGMENT & ACCEPTANCE

By accessing or using the Services, you:

- **Represent** that you have read, understood, and agree to these Terms;
- **Acknowledge** and **accept** all associated risks, disclaimers, and liability limitations;
- **Agree** to the dispute resolution terms, including binding arbitration and class action waiver (unless you opt out under Section 12.3);
- **Understand** that any $NET tokens on testnet may be redeemable in the future at a ratio or mechanism yet to be determined, and we have no obligation to provide specific redemption terms.

If you do not agree to the entirety of these Terms, you must **not** use or continue to use the Services.

Last Updated: 31 January 2025

Contact: [Insert Contact Email or Address]

[ ] I AGREE AND CONSENT TO THESE TERMS

2MEME

Wyatt L. Meldman-Floch

11/27/2023

Abstract

2MEME, a reputation model for p2p systems based on peer metadata and cryptographically secure hashes, is presented. It is dynamic, not relying on a set of trusted peers, and rotates the most accessible peers as leaders: those with the minimal entropy rate relative to all peers, or peers producing the most correct information. The core application and focus is optimizing consistency of a multi-layered consensus protocol, achieving Asynchronous Byzantine Fault Tolerance.

Contents

Forward

The following is a solution of a long standing goal in the crypto ecosystem: creating a consensus process that rewards good behavior and mitigates bad behavior, directly. A fundamental problem with existing blockchain technology is that while it aims to achieve decentralization, both the logic and the algorithmic complexity of existing consensus protocols limit the group of individuals who can mine to a small few. A core requirement to make mining accessible to the average person i.e. run on consumer hardware, is to remove the barrier to entry caused by proof or work which requires expensive hardware or proof of stake which requires substantial financial capacity. A consensus protocol that's accessible to the average consumer will make mining a viable alternative to application hosting and monetization of the web as a whole; as well as create new systems of governance that can engage the average person to participate in directly. It turns out that the solution is to provide incentives to nodes for acting in a way that optimizes their consistency (the C in the CAP theorem) and penalizes for network partitions.

2MEME is partially named because it incorporates two memetic concepts into a sybil attack resistance model: relative entropy or information gain, defined by the commonality of each node's state and node influence, which calculates how adherent each node's behavior is relative to the total set of nodes' behavior. The adherence to a performance 'meme' is how rewards are generated, essentially paying nodes for being the most consistent (common state with the whole) and the adherence to an influence 'meme' is how sybil behavior is determined; nodes that are not sybil follow a noisy distribution, while sybil nodes are identified by a strong signal. It's also sort of the sequal to an older approach to solving the above problem, which was called MEME. MEME differs from 2MEME in that it didn't use information gain as the feature space, opting to try and define a classifier and passing in a covariance matrix into the self-avoiding walk below, then normalizing using entropy of the walk output. That older version, which became Constellation's PRO, relied on a set of pre-trusted peers, which is a similar approach applied by most DAG protocols, but essentially became a federated proof of authority. The improvement, which became 2MEME, stems from application of a previous work, still in preprint, called the Generative Calculus, and the algorithm/results here will be used to expand upon generative calculus in that paper. 2MEME's implementation within the reality protocol is completely permission-less, but it could be re-written for a permissionned environment which would result in a potential improvement on fault tolerance (at least algorithmically, as it removes a serial state transition).

Note: This work is in pre-print and as of now a first-draft. The pseudocode sections aren't exactly the syntax I wanted but the pseudocode library is getting the job done. Also, I'm still working on formatting the surface plots in the results section. They're actually rendered in 3d, so if you run the code yourself[1] you can re-orient to get different views of the surfaces.

Introduction

For a distributed system to maintain consistency, it needs to optimize information gain and minimize discrepancies. One set of approaches rely on trust or reputation models to mitigate potentially conflicting updates from untrusted peers. There are many approaches to solving reputation problems in p2p networks. The most famous is Eigentrust, and there are many expansions upon the base framework, such as Honestpeer and Powertrust. Due to the curse of dimensionality, they all employ some type of random walk to explore the search space of transitive trust between nodes, calculating a probability distribution such as via Monte-Carlo integration to generate probabilistic trust scores of all peers. These expansions typically focus on finding new features or representations of trust, such as in Deepwalk or Node2vec, which create an embedding of

[1] https://github.com/reality-foundation/reality/blob/enable_model/simulation_data/surface_plot.py

social data to normalize the edge weights of the peer graph.

This paper follows a similar approach using entropy or disorder across peer behavior and is specifically applied to a dag-based multi-layered consensus protocol. Whereas many approaches such as Eigentrust require a seed or whitelist of authority nodes to base trust upon, this is insufficient when requiring decentralization such as for distributed consensus networks like cryptocurrencies. 2MEME circumvents this by determining correctness without pre-trusted peers, allowing nodes to join and leave and preventing centralized control over consensus.

The core of the algorithm extends from the principle of maximal entropy, however applied in reverse. The maximum entropy principle states that new information added to a system increases disorder relative to the previous state of that system, proportional to novel information added. However, in the system architecture described below, the data structures themselves optimize the spread of incoming data via rumor based gossip such that discrepancies between peer state form cliques representing potential network partitions.

Periodically, a self avoiding random walk is performed on a graph of nodes representing peers, with the edges representing a vector of the entropy rate calculated between data processed by each peer and the total set of data processed. The model chooses correct nodes according to a 'node influence metric' based on 'availability', which is defined as the most strongly linked nodes; ranking the peers/nodes by how similar their behavior is relative to the total set of nodes' behavior.

Another goal of 2MEME is to improve upon PRO models by handling node 'churn' in a permissionless environment, achieving 'elasticity' comparable to elastic infrastructure like Elasticsearch and Elastic Map Reduce, while still maintaining objective decentralization by operating without potentially fraudulent human input.

System Architecture

The system considered here consists of a two layer consensus protocol, with two separate consensus processes, L1 and L0, directly influencing each other. Future work incorporating more consensus layers can be formulated using the Poincare Protocol and Protocol Topology specified in Blockchain Cohomology[2].

L1 peers perform a federated consensus of $O(\sum peers/3) \equiv O(n)$ complexity, converging on the state of each peer's state cache. The contents of each state

[2]Meldman-Floch, Wyatt, "Blockchain Cohomology: Sec 23' http://ceur-ws.org/Vol-2478/paper2.pdf

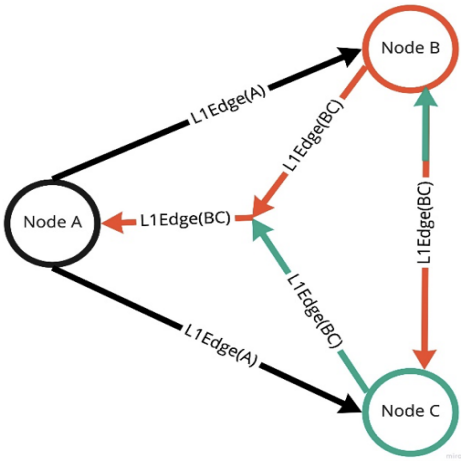


Figure 1: L1 consensus

cache is a ledger of Addresses and collection of Transactions: data structures performing the transfer of a numerical amount (tokens) from one Address to another Address. Each Address has an associated linked list formed out of Transactions sent from this Address. The links are recursive cryptographic signature hashes between each sequential transaction at a discrete Ordinal.

Periodically, as each L1 node reaches the limit of its mempool or in response to a timed trigger, they initiate a consensus process, acting as an 'owner' peer, which selects two 'facilitator' peers to share its state cache with. The two facilitator peers also share their state cache with each other, then back to the owner, and then to all other peers. The output of this process is a data structure, signed by the owner and facilitators, called a 'block' which consists of each peer's state cache data and two 'parent edges' called Tips, which are hashes of previous blocks; the end result is a triangle, where two corners are tips and the third is the new block. This can be conceptualized as a 'triangulation of state' which forms a forward arrow of time out of parallel-process state transition data. This is realized as a data structure called the 'Data Dependency Graph' which is a directed acyclic graph of blocks with two parent edges, and three dimensions; namely, height, width and depth.

These blocks constitute a directed acyclic graph called the Data Dependency Graph and their contents are validated by the L0, which updates the ledger of Addresses with valid Transactions. Both of these have a poset topology, from which the forward arrow of time can be constructed out of parallel events. Each block is then spread across all L1 and L0 peers via rumor based gossip, so that they can be used as 'Tips' to form edges between old and new blocks.

Tips and Facilitators are selected using a pure function that seeks to maxi-

Algorithm 1 L1 consensus algorithm

$n \leftarrow N \in \mathbb{N}$ ▷ these are L1 nodes
$n.mempool \equiv \{\text{Transactions}\}$
$n.mempool.size, n.mempool.triggersize \in \mathbb{N}$
$n.mempool.size, n.mempool.triggersize \leftarrow n$
$receivedOwnerProposal() : bool$
$recievedFacilitatorProposal() : bool$
$isOwnerPeer(n) : bool$
while True **do**
  if $n.mempool.size \geq n.mempool.t$ **then**
    Create owner proposal
    Send to 2 L1 peers
    Validate responses
    Form block and gossip to all L1 and L0 nodes
  else if $receivedOwnerProposal()$ **then**
    Sign and send proposal made from current node's mempool to other Facilitator peer
  else if $receivedFacilitatorProposal()$ **then**
    if $isOwnerPeer$ **then**
      Ensure both Facilitator proposals are equal
      sign and send block (made of all 3 mempools) to other all L1 and L0 nodes
    else
      Send Facilitator proposal to Owner
    end if
  end if
end while

mize the area (in terms of dimensions: height, width and depth) thus increasing the potential parallelism by enforcing consistency across all peers. L1 nodes are pruned according to an entropy calculation by L0 nodes on the blocks they create; each block is created deterministically according to the total set of tips and peers, deviations from this result low rewards and in low trust. L1 trust scores are then normalized between 1 and -1 with low trust outliers being removed and addresses temporarily blacklisted.

Data contained in the L0 peer's mempool, consisting of L1 blocks, is spread amongst L0 peers via rumor based gossip with $O(log(\sum peers))$ complexity, converging the state of each peer to the union of all peers'. Periodically, as the Data Dependency Graph reaches new heights or on a timed trigger in case of low network traffic, each L0 peer proposes a Snapshot to its L0 peers using rumor based gossip. The contents of these Snapshots are L1 blocks, and a parent reference, formed out of recursive cryptographic signatures, to the peer's previous Snapshot proposal. This forms a sequential/poset topology for L0 peer proposals like for Addresses. At each 'height' step, one proposal is selected from the set of all proposals, deemed the Majority Snapshot. The Majority Snapshot is chosen as the count of occurrences within the set of proposals, multiplied by the total node influence or cumulative sum of trust scores of each peer that proposed it.

Periodically, at an interval multiple of snapshot height called the 'height-diff' interval, the L0 nodes cycle between 'active' and 'passive' states. 'Active' nodes actively perform consensus, 'passive' nodes gather final snapshots and index into a database (block explorer). The total number of 'active' L0 nodes is equal to $\sqrt{\sum peers}$, thus the total complexity for ledger state convergence (known as finality time) is $O(\sqrt{\sum peers}^2)+O(log(n) = O(n)+O(log(n))$. Peers are cycled at each interval according to a deterministic locality sensitive hash function applied to the last snapshot hash and all the L0 peer addresses that have not been identified as Sybil; a node's probability of selection is also influenced by the amount of tokens in the address of the node, however the amount does not affect the outcome of consensus (which constitutes proof of stake). Rewards for each round are proportional to the cumulative sum of information gain reported by L0's peers and low trust outliers are blacklisted temporarily.

In both L0 and L1, a 'double spend' or submission of invalid data as valid, with the attempted result being the attribution of new tokens to some Address, the node Address's balance is reduced to 0 in a process known as 'slashing'. It is also worth noting that, as shown via recursive tip structure, there is no fault tolerance due to latency, thus it's fault tolerance is asynchronous and achieves Asynchronous Byzantine Fault Tolerance.

Finally, there is a meta-process running on all L0 nodes (both active and passive), which acts as a block explorer in addition to validating L0 snapshots and curating the list of all peers, namely the LΩ. All active L0 peers need to

Algorithm 2 $L\Omega$

$h, i \in \mathbb{N}$
$H \ggg i$
$h \leftarrow H$
$n \leftarrow N$ ▷ these are $L\Omega$ nodes
$n.mempool \equiv$ {Snapshots}
$n.mempool.size, n.mempool.triggersize, n.peers.active, n.peers.passive \in \mathbb{N}$
$n.mempool.size, n.mempool.triggersize, n.peers.active, n.peers.passive \leftarrow n$
$h.signatureChain \leq n.peers.active$
$isActive(n) : bool$
$invalidSnapshot(h) : bool$
$notUnique(h) : bool$
for $h \leftarrow H$ **do**
  if $invalidSnapshot(h)$ **then**
    Slash address value for all signatories on h
    Remove invalid signatories from n.peers.active
    if $n.peers.active - invalidsignatories == 0$ **then**
      Recalculate active peers from previous snapshot and previous set of inactive peers
      Gossip new active state to all subscribers (including L1 and client apps)
    end if
  else if $notUnique(h)$ **then**
    if $h.signatureChain < n.peers.active$ **then**
      Initiate health check/removal on missing peer ids
    end if
    Choose h with highest $\sum_n score(n)$
  else if $0 \equiv h \bmod i == 0$ **then**
    Calculate new active peers
    if $isActive(n)$ **then**
      Initiate active L0 process with other active peers
    end if
  else if $0 \equiv h \bmod i - 1 == 0$ **then**
    Send
  else if $\not{i}sActive(n)\& \not{i}nvalidSnapshot(h)$ **then**
    Gossip join requests, Snapshots
    Submit data to Downloading nodes
    Process ledger state requests
  end if
end for

sign the result of consensus (form a signature chain), which enforces convergence before sending to the larger $L\Omega$ network, but also creates a record of divergence (either malicious or not) and the nodes that diverged. If this happens, a health check is performed, essentially asking for the missing data; if it is not received, then the unresponsive peers are removed and consensus continues. In order to maintain a consistent peer list, new peer requests are gossiped across the active and passive nodes via $L\Omega$ and included in Snapshots.

Active Peer Selection

Active peer selection is performed by a deterministic locality sensitive hash function which operates on a list of peers and chooses equal amounts randomly between two partitions of active nodes; one partition A making up a minimum of 50% of all staked tokens, and the second partition B containing all remaining nodes. It's clear to see that as the network grows it implies that $B \ggg A$ due to token scarcity. While this does provide an advantage for early users/token holders, it mitigates the security risk inherent in horizontally scalable permissionless networks (which get faster as the network grows) while still providing ample possibility for new users (with lower stake) to participate.

Entropy Rate

Information gain can be formulated as the reduction of entropy or disorder in a dynamical system and depending on the characteristics of the system, it is calculated in one of many ways. For the purposes of 2MEME, which is formulated for application to consensus networks, it is calculated as a stochastic process. A stochastic process is an indexed sequence of random variables that do not need to be independent or identically distributed. In a consensus network, each peer continuously proposes variable state data, converging on an accepted state according to the rules of the consensus algorithm. This state data, in our case called blocks, can be independent or dependent on each other; and the amount of blocks as well as the specific blocks proposed can fluctuate or differ completely. Each block has an indexed order, or in the case of the system architecture above, a poset topology; meaning that they are strictly ordered. Thus these distributed systems fulfill the requirements of a stochastic process and can be modeled as such. While it is possible to apply 2MEME to linear blockchain protocols, it was formulated specifically for use in the system architecture above, with three indices: height, width and depth. The following formulas are specific to this poset topology. Consider a set of peers $N = (n_0 \dots n_i)$, acting as random variables which produce outputs $O = (o_{0,0,0} \dots o_{h,w,d})$, such that $h < w$ and $w > d$, the system has a strict order given by poset topology. These indices can be reduced to discrete indexes, such that each index, there is a binary value representing

each node proposing a specific block or not. This is calculated using following formula[3] for joint entropy

$$H(N) = -\sum_{n_1 \in N_1} \cdots \sum_{x_n \in N_n} P(n_1, \ldots, n_N) log_2[P(n_1, \ldots, x_n)] \tag{1}$$

the limit of which as h, w, d approaches infinity gives the entropy rate. If a block index contains each node, then the specific block has entropy of 0. If it contains $|n| < N$, the entropy is > 0. Conversely, if nodes propose blocks such that their indices conflict with other proposals, they contribute to the overall disorder in the system. Proposed blocks with valid yet duplicated data, contribute to overall disorder as well. Thus the state with the minimal entropy can be considered the greatest common subset of all proposed blocks, and entropy calculated as deviation from the greatest common subset. Note that in the case of graph partitions that are not within this subset, yet still contain valid data, the data should still be contained within the overall state transition, however the node that only processed it's lone subset can be considered faulty (potentially Sybil) in terms of consistency and partition tolerance (CAP). In order to promote consistency, a rumor based gossip algorithm propagates blocks, calculating signatures upon them, which then can be used as Tips, to optimize for a maximal subgraph of peers to accept the block and propose it within its Snapshot. This impedes several sybil attacks such as lie and wait and ddos, by attempting to create the longest signature chain as possible, i.e. the largest common subset; nodes attempting these types of attacks are identified via independent subsets and/or invalid blocks.

Finally, the inner product of entropy rates are used to construct a feature space F before passing into the random walk below (first part of 2MEME)

$$F = \sum_{N} \lambda_N \left|N_n\right\rangle \left\langle N_n\right| \tag{2}$$

where λ is a normalization function that maps the sum to between -1 and 1.

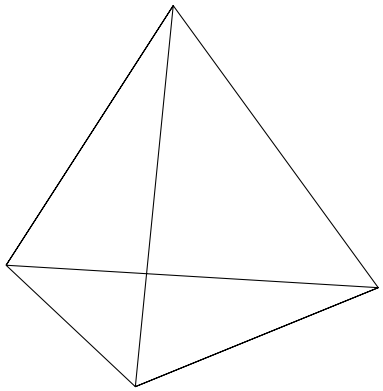
Maximization vs Minimization

While from the L0 perspective and optimal system is defined in terms of a minimal entropy rate, the converse is true for the L1 perspective. An optimal L1 maximizes the inforcmation added within blocks to the data dependency graph. From this perspective, information gain can be described in terms of a Lebesgue measure, which is a geometric realization of an information space; we

[3] thm 2.4 'https://math.nd.edu/assets/275279/leblanc_thesis.pdf'

can calculate it as the volume of a tetrahedron where the length of each side is equal to the entropy or disorder between two tips.

Calculation of this platonic solid is an computationally hard problem and equivalent to a proof of work step in POW blockchains, however the calculation of these polyhedra actually optimizes the data stored in the data dependency graph data structure, both enforcing a work step (preventing ddos and lie in wait type attacks) but in a useful way. Hence "Useful Proof of Work".



L0 consensus: Permissionless vs Permissioned approaches

Two algorithms for gathering entropy data are presented below. The key algorithmic difference is the rate and logic for making proposals.

The first enforces a service level agreement requiring each peer to train its model at the same rate, achieving greater determinism and enabling a token reward model for an open network and preventing nodes' ability to forge scores to manipulate the selection algorithm; by first committing an encrypted hash of their scores to the ledger, then sending the unencrypted scores so peers can calculate the Majority Snapshot. In the batch model, at every snapshot height-diff interval, each peer proposes a new predicted trust vector (scores) within their proposals. They are then used to weight peer proposals for the Majority Snapshot calculation.

The second, online algorithm, reduces the in-memory cost of running each peer's model on a deterministic schedule albeit at the loss of determinism that would allow fair selection of validators. It is more suited perhaps to applications that can relax determinism required by an open network to focus on elasticity. The online algorithm periodically gossips predicted trust vectors to

Algorithm 3 Batch Algorithm (permissionless): Entropy Rate (Optimal for node rewards/cycling between active and passive nodes)

$h, i \in \mathbb{N}$
$H \ggg i$
$h \leftarrow H$
$n \leftarrow N$ ▷ these are L0 nodes
$n.mempool \equiv \{\text{Blocks}\}$
$n.mempool.size, n.mempool.triggersize \in \mathbb{N}$
$n.mempool.size, n.mempool.triggersize \leftarrow n$
for $h \leftarrow H$ **do**
  if $0 \equiv h \bmod i - 1 == 0 \& n.mempool.size \geq n.mempool.t$ **then**
    $stateSpace = entropy(N)$ ▷ Entropy rate relative to GCS
    $probabilitySpace = selfAvoidingWalk(statespace)$
    Send encrypted *probabilitySpace* to peers within Snapshot
    Choose proposed snapshots weighted by cumulative sum of scores
  else if $0 \equiv h \bmod i == 0 \& n.mempool.size \geq n.mempool.t$ **then**
    Send unencrypted *probabilitySpace* within snapshot to peers
    if $valid(n), \forall n \in N$ **then**
      for $n \leftarrow N$ **do**
        Update scores for peers
        Choose Majority Snapshot
        Update active and passive peerlist
      end for
    end if
  else if $n.mempool.size \geq n.mempool.t$ **then**
    Send snapshot to peers
    Choose proposed snapshots weighted by cumulative sum of scores
  end if
end for

peers over the Peer api endpoint (“ /trust”), which are then cached and fed into the TrustManager on a time based periodic interval.

Algorithm 4 Online Algorithm (permisioned): Approximate Entropy Rate (current implementation, optimal for minimal resource usage, training model over shorter periods should help output, spamming results/sybil collusion should be detected by model, good test)

$h, i \in \mathbb{N}$
$H \ggg i$
$h \leftarrow H$
$n \leftarrow N$ ▷ these are nodes
$n.mempool \equiv \{\text{Blocks}\}$
$n.mempool.size, n.mempool.triggersize \in \mathbb{N}$
$n.mempool.size, n.mempool.triggersize \leftarrow n$
$e = rand() : bool$ ▷ This is a timed based trigger
while True **do**
  if $n.mempool.size \geq n.mempool.t \& e == True$ **then**
    $statespace = entropy(N)$ ▷ Entropy rate relative to GCS
    $probabilitySpace = selfAvoidingWalk(statespace)$
    Send unencrypted scores to peers in Snapshot
    Update active and passive peerlist
  else$n.mempool.size \geq n.mempool.t$
    Send snapshot to peers
    Choose Majority Snapshot
  end if
end while

Monte Carlo simulation: estimation via self avoiding random walk

Next a self-avoiding walk is employed to perform community detection, the output of which can be used to calculate availability and node influence[4]. Note, to avoid confusion between a consensus network and graph, nodes are called servers below.

The output of the entropy rate calculation is a graph, with a server's peers corresponding to nodes and edges as the relative joint entropy between the server and its peers. The edges are a ‘view’ of the performance of each peer relative to itself. This matrix (namely $F = \sum_N \lambda_N |N_n\rangle \langle N_n|$) is passed to the TrustManager, a background process that performs a self-avoiding random walk across the graph of nodes connected by relative joint entropy and outputs

[4]https://www.sciencedirect.com/science/article/pii/S0378437118304242

a vector containing a trust score for each node relative to the server hosting running the process (predicted trust). Pseudocode for this and the following methods are omitted due to length, but can be found in the Reality Protocol's codebase[5].

The self avoiding walk is performed by the method runwalkfeedbacksinglenode, which performs a series of feedback rounds, walking on the input graph and adjusting the edge weights between nodes for each successive round. The total number of feedback cycles are configurable and in general the larger number of cycles has a more accurate output, albeit at the cost of increasing resource intensity. The configurations are batchIterationSize and maxIterations. On each batch iteration a random path length from a random number generator (between 1 and total nodes) is chosen and then passed to the walk. The walk goes through and only walks on positive edges (relative entropy scores are normalized between -1 and 1), keeps track of nodes visited so far, then the sampling function determines the next neighbor to walk on. This is determined according to the normalized probability (via method normalizedPositiveEdges), such that the positive edge subset sum up to one.

As this iterates, transitive trust scores are added, because products of trust are quite small. However, over many iterations they sum, to large numbers which is better for differentiating between scores. The main walk function walk() gets invoked by walkFromOrigin() inside runWalkRaw() which iterates over numIterations, adding up the scores into val walkScores for each node, removing the server's own. After that function is called, one "batch" has been created.

Finally there is batch convergence in runWalkBatchesFeedback. This converges when a delta variable, which is just root mean squared error, becomes less than or equal to an epsilon variable, where epsilon is set to 1e-6; i.e the function terminates when scores don't change between batches by one part in a million. The output of the walk, batchScores, is not normalized, so they are renormalized by the Normalize function between each iteration until convergence.

One difference from similar models is the incorporation of negative scores. After the batches are performed, it explores the negative scores (val negativeScores) of nodes that it trusts (positive outputs of the walk). On the first cycle, the model only reaches nodes with positive transitive trust, which can be considered the most influential servers. These most influential servers are relative to the host server, and the servers they distrust have their scores down weighted. The positive scores and negative scores are added, then weighted by how influential the server proposing these scores is. They are weighted such that its negative edge trust quantity*(influential node's score/numNegEdges), after that they are all normalized via renormalizeAfterNegative. Output, P_i^h is defined as follows.

[5]https://github.com/reality-foundation/reality/tree/enable_model/modules/sdk/src/main/scala/org/tessellation/sdk/infrastructure/trust

Given a source node i, suppose it is possible to reach $N_i(h)$ different nodes performing walks of length h, departing from i. Then we say that i has N_i neighbors at a distance h. Each neighbor is reached with a different probability, which is represented by the vector $P_i^h = \{P_1^{(h)} \dots P_{N_i(h)}^{(h)}\}$ Given P_i^h,the accessibility $k_i(h)$, defined below is

$$k_(h) = \exp\left(-\sum_j p_j^h log p_j^h\right) \tag{3}$$

Classification logic

Now that we have a manifold of node influence, we can look for signals between sybil and non-sybil nodes and build a classifier. Pseudocode for the classifier is omitted due to length, but can be found in the Reality Protocol's codebase[6].

The classifier defined as follows simply looks at the first three principal eigenvectors (those with the largest three eigenvalues), to identify sybil nodes. Its quite possible to improve upon the experimental results using a differential equation solver as provided in Tensorflow [7] or Pytorch[8], which would be able at least incorporate higher order terms and improve distribution fit, if not catch hidden nonlinear signals. As you'll see, the classifier follows steps similar to those in analytically solving systems of differential equations.

First a new vector consisting of the successive diffs, normalized between 0 and 1, between all elements in P_i^h is calculated, namely d_i^h which has $dim(i, h-1)$ then generate a manifold from direct sum with itself:

$$d_i^h \oplus d_i^h \tag{4}$$

calculate the principal components (eigenvectors sorted by highest eigenvalue), $\lambda_1, \lambda_2, \lambda_3$ as well the 'max plane' p_i, similar to the 'top hat' in signal/-fourier analysis which is an index range in d_i^h with flat values (not perfectly flat, all diffs are within a threshold of the mean μ or a minimum of 0.1), the 'right bias' b_r and 'left bias' b_l which are the ration of total sum of values to the left of p_i divided by total scores and total sum of values to the right of b divided by total scores respectively. Finally consider 'population diffs'

[6] https://github.com/reality-foundation/reality/blob/enable_model/modules/sdk/src/main/scala/org/tessellation/sdk/infrastructure/trust/TrustModel.scala

[7] https://www.tensorflow.org/probability/api_docs/python/tfp/math/ode

[8] https://github.com/rtqichen/torchdiffeq

$$\lambda_{pop} = 1 - (\lambda_{i-left}/\lambda_{i-right}) \tag{5}$$

where $1 - (\lambda_{i-left}/\lambda_{i-right})$ are the population or number of nodes contained within biases calculated from each of the principal eigenvalues.

Now, there are five scenarios to determine non-sybil nodes: first if the max plane makes up over 20% of the distribution and either biases are less than 90% of the distribution, return the biases as non-sybil nodes. Second, if either the absolute value of the ratio of mean to population variance is less than $1x10^{-17}$ or the difference between the absolute value of the first principal component and the absolute value of the second principal component is greater than the population standard deviation by 30%, return the entire set of P_i. Third, if the difference between the principal and second principal eigenvector is greater than the population standard deviation by 20% or the minimum population diff (of all 3) is the population diff of the maximum principal component, then if the difference between the absolute values of the population biases of the principal eigenvalue $|\lambda_{left}| - |\lambda_{right}|$ is greater than the population standard deviation, return the right bias of the max principal component; if only the first condition is true, return the left bias of the max principal component. Next he same logic is then applied for the second and third principal components.

Finally, if the total non-sybil nodes to be returned are less than 10% of the population (90% identified as sybil), return the entire set of nodes.

Experimental results

The following results were generated from a simulation involving 100 node instances. Several simulations were run for various types of attacks, which will be outlined below, and with varying percentages of sybil nodes all acting in unison from 0% to 100% in 20 percent intervals. Each type of attack can be reduced down to sybil nodes reporting scores of themselves or non-sybil nodes according to a distribution i,e low scores for non-sybil or high scores for sybil nodes. In the following visualizations, each attack type occurs on the y axis, the sybil percentage is given by the x axis and the z axis is the performance, either F-score of the classifier or a gain in selection rate.

F-score, specifically F1-score was chosen as that performance metric because it's application to classifiers, specifically binary classifiers, is widely accepted. The F-score is the geometric mean of precision and recall, which means that it measures not just the ability of the classifier to filter out sybil nodes, but also to not remove non-sybil nodes. The choice of using F1 in the first plot came from the desire to show fairness as opposed to just the removal of sybil nodes (i.e. include false negatives). The second performance metric chosen was the ratio of false positives vs true positives, which is an accuracy metric showing

the percentage of sybil nodes in the output of the classifier; at most an average of 3.5% of the sybil group were not removed.

Note that many different scenarios are tested within unit tests and the simulation framework, notably analysis of sybil score cliques which submitted scores according to non-uniform distributions (i.e. linear), however the model was equally as good at identifying them as for uniform scores (1.0, -1.0, 0.5 etc. except for 0.0), but the test data is of different dimensionality and the code will need to be refactored before expanding the visualization. It's also worth noting the as the clique size increases to the size of the total set of nodes, we see performance mirroring that of the perfect non-sybil case, which is expected. Also sybil nodes reporting forged scores for non-sybil nodes was equally identified as self-reports.

The tests used in visualization are as follows:

organicDistroDir: no attack, organic behavior from each node

symmetricSybilOne: all sybil nodes give each other the max value in their proposal.

symmetricNegOne: all sybil nodes give each other the min value in their proposal.

symmetricZeroSybil: all sybil nodes choose 0.0 as their proposal for each other.

symetricLinearDistro: all sybil nodes propose increasing scores for each other in a linear distribution.

Here we have a surface plot of F-score according to attack and sybil threshold. As we can see, the most effective attack is for sybil nodes to self report scores of 0.0, yielding a minimum F-score of 0.73, the reduction here coming from false negatives. This is still a solid score for any classifier and compared to existing consensus models (i.e. Bitcoin) which has an F-score of 0 for sybil percentages above 50 percent (due to the fact that one outcome of a 51% attack is removal of non-sybil nodes), is remarkably successful and provides a huge comparative advantage.

Here we have a plot of false positives / true positives. As mentioned above, F-scores drop primarily due to false negatives. However for our model which is most focused on removing sybil nodes, we can see that very few are not removed; specifically the small bump in symmetricZeroSybil has a maximum of 0.035, meaning only 3.5% of nodes selected are from the sybil group. The spike at the end is the 100% sybil case, which is where the model breaks down (note the actual was 1/0, so a value of 1.0 was imputed.)

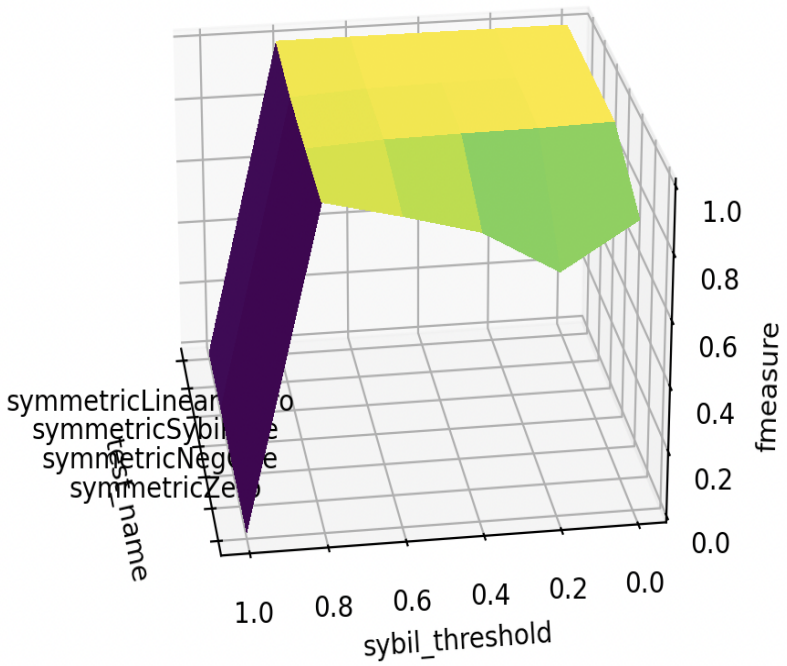


Figure 2: F-scores by each attack type and sybil threshold

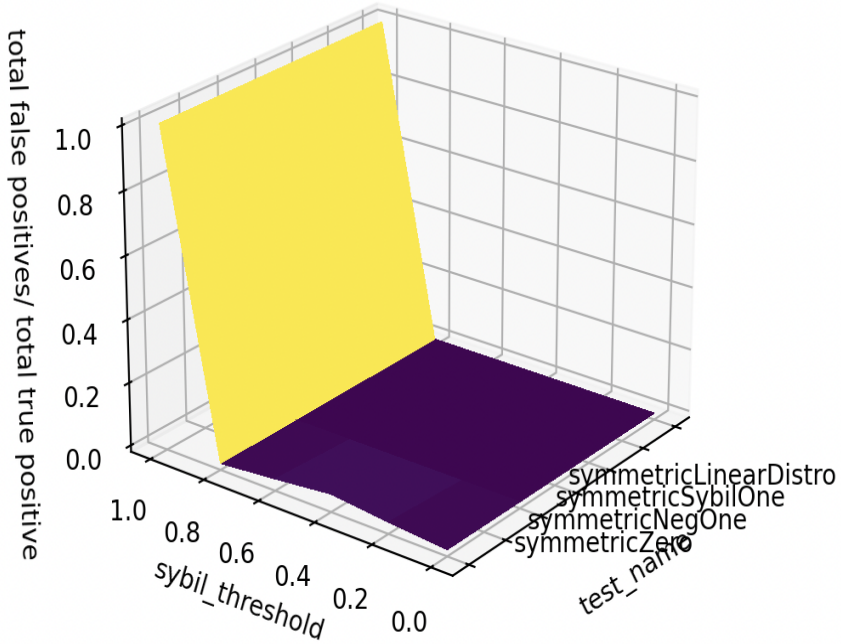


Figure 3: False positive / true positive by each attack type and sybil threshold

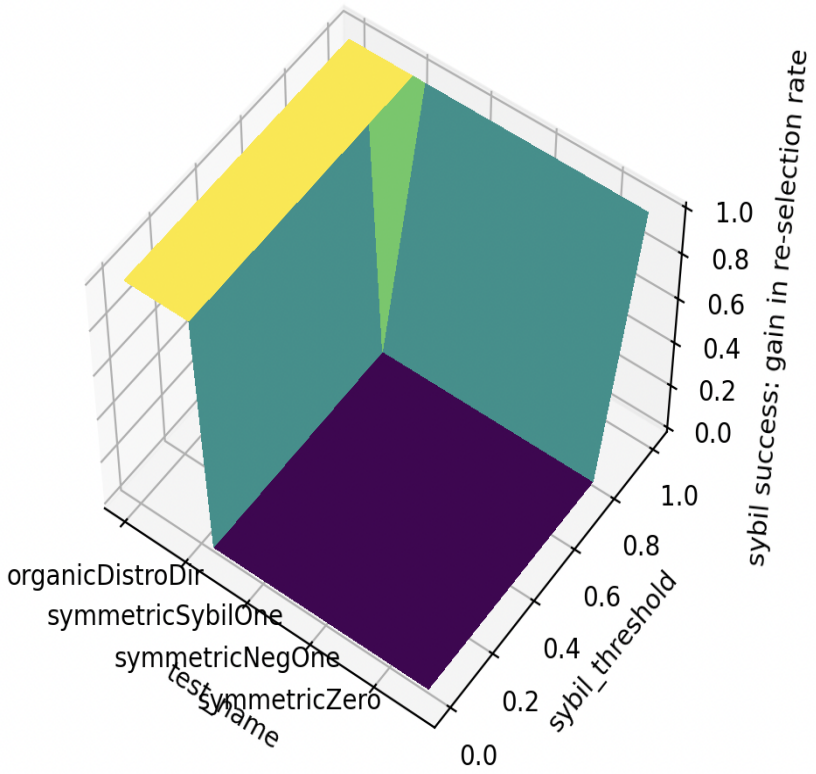


Figure 4: Sybil node gain

Here we have a plot of the difference between nodes cheating and not cheating. The vertical axis is the percent increase in sybil nodes (forging scores) that are selected for the next round of consensus vs if they actually just submitted scores based upon their true entropy rate (not cheating). The ledge on the top left represents the perfect non-sybil case (0%) and the top right shows the 100% sybil case; both of these shows 100% of nodes selected for next round (as expected). Most importantly, the flat plane shows that the only sybil nodes selected would have been selected anyway, based upon their immutable entropy rate; this means that the classifier removes gains from cheating.

Sybil resistance

Open networks pose inherent security risks deriving from Sybil attacks; where a group of individuals attempt to forge network or ledger state. The attack surface grows as the cost for participation, either through hash power or stake, is reduced; essentially reducing the cost of a sybil attack by minimizing the cost to run sybil nodes. This has the transitive effect of also reducing the efficacy of horizontal scaling techniques, which increase speed as a function of the number of nodes. 2MEME mitigates these attacks in such a way that promotes participation (low fees and high transactions per second), both as a miner and a user.

Sybil attack

As shown in the false positive vs true positive plot, the classifier is able to remove the vast majority of sybil nodes (at most a 3.5% sybil rate) and only fails with close to 100% sybil rate. That being said, given that the size of the active peers is much less than the passive, it would be feasible, albeit of a low probability, that entire committees of active nodes could be sybil. However, our approach is still able to increase the cost required for a 100% successful sybil attack beyond the typical 51%.

Active L0 nodes are chosen from two buckets: the smallest set of nodes who's stake adds up to a minimum of 50% and the rest of the nodes. Because of this, the cost to achieve a sybil attack with 100% guarantee (i.e. all active nodes are sybil) would require over 50% of the total stake in the network and over 50% of the total hash power (number of nodes); most permissionless networks require only one of these to be over 50% to fail whereas this system requires both.

Consider the case where 10% of all L0 nodes contain over 50% of staked tokens and the other 90% makes up less than 50%, that would mean a guaranteed attack requires over 50% of staked tokens and up to 100% of the hash power.

This is considerably more costly than pure POW or POS which fail above 50% controlling hash power or stake, respectively.

Lie and wait attack

In a reputation system, a lie in wait attack occurs when nodes increase their reputation so that it can then use the high score later for abuse. In 2MEME, reputation is calculated based on an immutable log of information gain for each system, however the specific reputation score only has relevance within the context of the information gain of all other nodes and is ephemeral. Thus due to the nature of the model, which doesn't carry over reputation values between the active/passive cycle, it's not possible to 'save up reputation points' and spend them on an attack.

The closest one could come to this attack is performing well by producing minimal entropy in every snapshot throughout the entire consensus round, and then forging their score proposals, however this is the focus of the various attacks in the experimental analysis section and the model has shown to be successful at identifying sybil nodes regardless of their scores, or the distribution of scores across all sybil nodes.

Eclipse attack

As shown in the $L\Omega$ definition, the result of a 51% attack is a network partition where the offending nodes would only have an invalid chain state. This is essentially how one would perform an Eclipse attack, where offending nodes send incorrect peer data or chain data to a newly joining node. In this case, honest nodes would not be able to join the invalid partition (validation of the download step from a sybil node would not pass) and honest subscribers such as client utilities or L2 applications would be able to identify a sybil partition via invalid published snapshots. The end result of a wider 51% attack would be an increase in latency (as for an ordinary network partition) as opposed to an invalid chain state.

Ghost Chain Attack

As in most Proof of Work protocols, like Bitcoin, the probability of an attacker generating an alternative chain which is longer than the accurate chain in an attempt to change one of its own transactions to take back money spent is isomorphic to the Gambler's Ruin problem[9]. For a probability of attack success

[9]Sec 11, https://bitcoin.org/bitcoin.pdf

less that 0.1% with sybil attack probability (1− percentage of sybil nodes needed to perform a sybil attack), number of parent blocks that need to be accepted is given by the poisson distribution[10]. Given our sybil threshold caps neatly at 80 (2MEME can handle 80% sybil nodes) we can achieve 99.9% probability of thwarting this attack with $z = 11$ or rather only selecting a snapshot in finality which has 11 accepted parents.

Further investigation

Modifications to the self-avoiding walk implementation could yield positive effects. As in many Monte Carlo integrations, the direction chosen at each step could be chosen according to a distance metric as opposed to randomly. Albeit, at the computational cost of increasing the dimensionality of the trust graph's edges as well as in-memory expense of the metric calculation. Notably, this approach was employed by N. Koroviako[11] who used Jaccard similarity to define trust out of relationships between review texts; as 2MEME's approach focuses on information gain, it would follow to select the next path at each step based on minimizing entropy rate of second order proposals (each edge would contain raw proposals, and choose the next node that has the minimal entropy rate compared to the current's proposals.)

As our SAW has non-linear (fractal) dimensionality[12], another possible improvement could come from using Hausdorff clustering[13] to identify embedded hierarchies (perfom classification in terms of the surface of clusters); this could have applications in further anomaly detection providing higher dimensions of entropy, which would be useful for improving the classifier with a differential equation solver or perhaps a graph embedding/deep learning approach. Note that the node influence metric calculated by the SAW is not strictly defined (it does not strictly obey the triangle equality, thus it's a measure not a metric) so it is actually compatible for the Hausdorff measure.

The cryptographic signatures described in the system architecture use ECDSA. However, with the dawn of quantum computers and the fact that elliptic curve cryptography is vulnerable Shor's algorithm[14] new 'post-quantum' cryptography (PQC) has been introduced which would allow 2MEME to remain effective in a post-quantum world. Fortunately, due to the structure of the key utilities in our codebase, we can easily swap out ECDSA for these newer PQC implementations.

[10] pg 7, https://bitcoin.org/bitcoin.pdf
[11] https://www.sciencedirect.com/science/article/pii/S1877050912003936?ref=pdf_download&fr=RR-2&rr=8306a2ef2866cee9
[12] http://www.math.uchicago.edu/ lawler/miami1.pdf
[13] https://arxiv.org/pdf/0801.0748.pdf
[14] https://arxiv.org/pdf/1804.00200.pdf

Conclusion

It's been shown that 2MEME incentivizes consistency by rewarding nodes according to their information gain. Also by showing a lack of gain for sybil nodes while still minimizing removal of non-sybil nodes (as evidenced by the F-score distributions,) 2MEME is able to provide a fair playing field for honest nodes.