Yotta: A Large-Scale Trustless Data Trading Scheme for Blockchain System

XIANG LIU*, National University of Singapore, Singapore
ZHANPENG GUO*, Xidian University, China
LIANGXI LIU, Northeastern University, USA
MENGYAO ZHENG, Harvard University, United States
YIMING QIU, University of Michigan, , United States
LINSHAN JIANG†, National University of Singapore, Singapore

Data trading is one of the key focuses of Web 3.0. However, all the current methods that rely on blockchain-based smart contracts for data exchange cannot support large-scale data trading while ensuring data security, which falls short of fulfilling the spirit of Web 3.0. Even worse, there is currently a lack of discussion on the essential properties that large-scale data trading should satisfy. In this work, we are the first to formalize the property requirements for enabling data trading in Web 3.0. Based on these requirements, we are the first to propose **Yotta** — a complete batch data trading scheme for blockchain — which features a data trading design that leverages our innovative cryptographic workflow with IPFS and zk-SNARK. Our simulation results demonstrate that Yotta outperforms baseline approaches up to 130 times and exhibits excellent scalability to satisfy all the properties.

CCS Concepts: • **Security and privacy → Distributed systems security**; **Information accountability and usage control**; • **Networks → Peer-to-peer protocols**.

Additional Key Words and Phrases: Blockchain, Data Trading, Web 3.0

ACM Reference Format:
Xiang Liu, Zhanpeng Guo, Liangxi Liu, Mengyao Zheng, Yiming Qiu, and Linshan Jiang. 2018. Yotta: A Large-Scale Trustless Data Trading Scheme for Blockchain System. In *Proceedings of Yotta: A Large-Scale Trustless Data Trading Scheme for Blockchain System (Nanyang Blockchain Conference 2025).* ACM, New York, NY, USA, 9 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

We are transitioning from the Web 2.0 era to the Web 3.0 era. In Web 3.0, AI has become a rapidly developing focus. For AI, data and computing power have, in fact, become the most valuable resources, with data serving as the foundation. As a result, data trading has emerged as one of the key focuses of Web 3.0 [12].

However, the current methods for data trading on Web 3.0 data markets remain limited. For instance, trading AI datasets or medical data at a large scale often requires going through official authorities, which is complex and

*Both authors contributed equally to this research. The order of the two authors was chosen randomly.
†Corresponding Author.

Authors' Contact Information: Xiang Liu, liuxiang@comp.nus.edu.sg, National University of Singapore, Singapore, Singapore; Zhanpeng Guo, zhanp.guo@gmail.com, Xidian University, Xi'an, China; Liangxi Liu, liu.liangx@northeastern.edu, Northeastern University, Boston, USA; Mengyao Zheng, mengyaozheng@alumni.harvard.edu, Harvard University, Boston, United States; Yiming Qiu, yimingq@umich.edu, University of Michigan, Ann Arbor,, United States; Linshan Jiang, linshan@nus.edu.sg, National University of Singapore, Singapore, Singapore.

Manuscript submitted to ACM

cumbersome. Although such processes can mitigate the risks of deception by untrusted parties, they contradict the core principles of Web 3.0 — decentralization [19], User ownership/sovereignty over data [15], trustless systems [22], and permissionless innovation [9] — and ultimately hinder the development of Web 3.0.

On the contrary, in Web 3.0, everyone can freely contribute and trade data from themselves, and the volume of data has already reached the zettabyte scale [11]. Although using Ethereum smart contract [24] on Web 3.0 can ensure the security of data, it still fails to meet practical demands. First, it cannot support large-scale data purchasing, such as many-to-many transactions in real-world scenarios. Second, the traded data is trustless, making it difficult to assess the credibility of the entities or organizations involved, which can potentially result in the exchange of garbage data and expose participants to significant financial loss risks.

Even worse, beyond just security and large-scale data trading, there is currently no formalized summary of the properties that an ideal scheme for large-scale data trading in Web 3.0 should satisfy. A natural question then to ask is "What should an ideal scheme look like?" To address this gap, we are the first to propose the "**SQUATS**" principle that an ideal data trading scheme should fulfill:

- **Scalability**: The scheme should not only support atomic transactions but also enable large-scale batch data trading as well as many-to-many transactions.
- **Quality**: The scheme should allow verification of data quality, as high-quality data is fundamental for data mining and machine learning. It should ensure that the seller's data meets the buyer's specified requirements.
- **Usability**: The scheme should be easy to use, user-friendly, and extensible for broader adoption.
- **Autonomy**: The scheme should enable automated trading, integrating seamlessly with existing popular smart contracts without excessive manual intervention or delays.
- **Transparency**: The platform should be transparent, eliminating the need for a trusted third party. Buyers and sellers should not have to rely on intermediaries to escrow data or payments.
- **Security**: The scheme should guarantee data security, preventing malicious third parties from tampering with or decrypting the data. It should also ensure that sellers provide the correct decryption keys, and that only authorized buyers can access the purchased data.

Building on the above principle, we are the first to integrate the spirits of Web 3.0 and provide a complete batch data trading scheme based on blockchain, named **Yotta**. Yotta leverages Inter-Planetary File System (IPFS) [2], zero-knowledge succinct non-interactive argument of knowledge (zk-SNARK) [5, 10], and novel cryptographic algorithms and workflow designs to realize an innovative encrypted data trading scheme for blockchain. The key of Yotta lies in secure and reliable large-scale batch data trading and satisfies the aforementioned properties. Our method demonstrates excellent scalability, paving the way for handling future yottabyte scale data volumes. Furthermore, Yotta is designed to be easily horizontally extensible, allowing seamless integration with future methods for enhanced performance. Our contributions are summarized as follows:

- We are the first to systematically propose the essential properties that a large-scale data trading scheme in Web 3.0 should satisfy, and we summarize them as the **SQUATS** principles.
- Based on these principles, we propose **Yotta**, a blockchain-based data trading scheme that satisfies all the identified properties. Yotta is also designed to be extensible and can be easily integrated with existing systems.
- We build our Yotta prototype and conduct simulation experiments, demonstrating that our approach achieves up to 130× acceleration compared to baseline methods.

2 Related Works

2.1 Key Elements in Web 3.0

2.1.1 Ethereum. The Ethereum protocol was the first to implement practical smart contracts, extending the Bitcoin system [20] and becoming one of the cornerstones of blockchain technology. Due to Ethereum's scalability and versatility, its role has evolved beyond merely supporting digital currencies, such as Decentralized Finance (DeFi) and Non-Fungible Tokens (NFTs), to enabling decentralized applications (DApps). Ethereum allows a large number of entities to participate in the execution of smart contracts, with each maintaining a copy of the blockchain ledger. This decentralized nature ensures transparency and immutability, enabling the realization of atomic transactions. Our Yotta prototype is also built upon improvements to Ethereum smart contracts.

2.1.2 zk-SNARK. Zero-knowledge proofs (ZKPs), especially zero-knowledge succinct non-interactive arguments of knowledge (zk-SNARK), have been widely adopted in the Web 3.0 ecosystem—not only in cryptocurrencies [4], but also in blockchain rollups [17], AI regulations [18], and anonymous credentials [21]. ZKPs offer strong privacy guarantees, support scalability, and eliminate the need for trusted third parties, aligning with the principle of minimal disclosure. zk-SNARKs, in particular, are attractive due to their succinct constant-sized proofs and verification complexity that remains independent of the data size, making them highly applicable in real-world systems. In fact, any problem with complexity below or equivalent to NP-complete can be efficiently reduced to a zk-SNARK-friendly format. As a result, Yotta integrates zk-SNARK to enable verifiable data quality and reduce the overall verification complexity.

2.1.3 IPFS. Libp2p allows users to discover peers and establish seamless communication through a Distributed Hash Table (DHT), while also supporting decentralized content distribution. As a foundational technology within the Web 3.0 ecosystem, IPFS integrates libp2p to build its peer-to-peer network. IPFS is a distributed, decentralized file sharing system, where file contents are located and verified using cryptographic hashes to ensure data integrity. Consequently, a large number of users in Web 3.0 and cloud services leverage IPFS to share their content. In our Yotta prototype, we also adapt IPFS as the underlying system for data storage.

2.2 Data Trading Schemes

Early data trading schemes primarily focused on privacy. Zyskind et al. [28] first propose a decentralized approach for personal privacy protection. Yue et al. [25] focus on security to explore blockchain-based models for big data sharing. Zheng et al. [27] further provide a comprehensive review of blockchain applications in large-scale data sharing. Dong et al. [8] leverage secure multi-party computation and differential privacy to enhance data protection, enabling potential blockchain applications. However, these approaches lack discussions with practically supporting large-scale data trading.

Although Jung et al. [13, 14] support large-scale data markets, their schemes assume that agents are trusted, which can lead to information leakage. Dai et al. [7] propose a traceable data trading protocol, but it does not guarantee data quality. Taleb et al. [23] provide a quality evaluation framework but without sufficient implementation details. Zheng et al. [26] are the first to explore large-scale data trading on the blockchain; however, their prototype remains inefficient and cannot effectively support one-to-many transactions with a trusted key center. Some researchers have further explored the use of techniques such as Divisible Computation Diffie–Hellman (DCDH) as their decentralized method to exchange data in a manner similar to key exchange without a trusted third party to fulfill the requirements from buyer

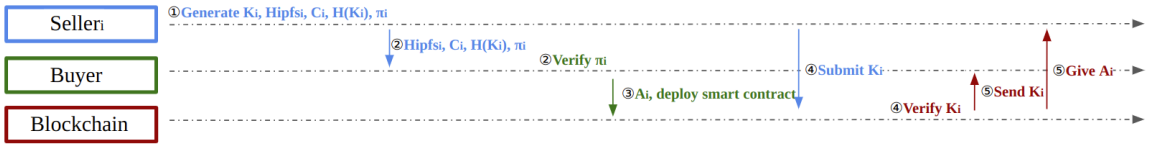

Fig. 1. Overview of our Yotta scheme, which consists of five steps.

and seller in Web 3.0. Nevertheless, these approaches still face similar limitations: they struggle to achieve scalability when integrated with blockchain and smart contracts for large-scale data trading.

We are the first to ensure privacy while supporting large-scale data exchange, and simultaneously satisfying all properties defined in our SQUATS principles. In the next section, we will describe our scheme in detail, starting with an introduction to how Yotta achieves one-to-many transactions as a foundation for realizing many-to-many transactions.

3 Our Methods

3.1 System Model

Our Yotta system model mainly consists of three entities: Buyer, Seller and Blockchain system.

- Buyer: Intends to purchase data from one or more sellers.
- Sellers: Multiple parties (sellers) who own and wish to sell their respective data, which is stored on IPFS.
- Blockchain: Responsible for running smart contracts and verifying and executing the payment transactions.

3.2 Preliminaries

This subsection introduces the mathematical notations used in this paper. The following examples use IPFS for illustration purposes; however, any data platform could be used in practice. Our current prototype is implemented on IPFS.

- For the data trading, we have dataset $(Dataset, D)$. The dataset owned by the n sellers consists of multiple secret data items $[S_1, S_2, \cdots S_i \cdots S_n]$, where each S_i represents the data belonging to seller i and all the data is stored on IPFS.
- IPFS Content Hash $(IPFSHash, Hipfs)$: Used to uniquely identify data stored on IPFS.
- zk-SNARK Proof $(Proof, \pi)$: Each seller must generate a proof demonstrating that they indeed possess each secret item in the dataset, and that these items match the data referenced by their corresponding IPFS addresses.
- Encryption and Decryption Function (either symmetric or asymmetric) $(Encrypt, Decrypt)$: Used to encrypt and decrypt each address for trading data.
- The seller discloses a subset of data samples along with their labels. The buyer examines the samples to verify that they meet the intended requirements, and publicly specifies an evaluation function $F()$.

3.3 Yotta Scheme

Our Yotta Scheme is primarily composed of five steps, as shown in Figure 1. Then, we will discuss the five steps in detail in this subsection.

3.3.1 *Step 1.*

- (1) Each seller i selects their secret trading data S_i;

- (2) Each seller i independently generates an encryption key K_i for their secret data address;
- (3) Each seller i uploads their data to IPFS and obtains the corresponding IPFS content hash $Hipfs_i$ provided by IPFS;
- (4) Each seller i computes the encrypted address information $C_i = Encrypt(K_i, address_i)$;
- (5) Each seller computes the hash of their encryption key $H(K_i)$;
- (6) Each seller uses zk-SNARK to generate a proof π_i demonstrating that they possess both S_i and K_i, and each $H(K_i)$ matches the given hash value, and S_i satisfies $F()$; and each C_i is the encryption of the address using K_i, and the associated data matches the contents referenced by the IPFS address.

3.3.2 *Step 2.*

- (1) Each seller presents to the buyer their corresponding $H(K_i), C_i$, IPFS content hash $Hipfs_i$ and proof π_i;
- (2) The buyer verifies the validity of each π_i to ensure that the seller indeed possesses the dataset S_i and the corresponding encryption key K_i; and S_i satisfies $F()$; and C_i is the encryption of the address using K_i; and the data is indeed stored at the corresponding IPFS address.

3.3.3 *Step 3.*

- (1) The buyer deploys a smart contract on the blockchain, depositing the payment amount A_i for each S_i and specifies the payment condition: each seller must provide a decryption key K_i such that C_i can be decrypted to reveal the data address, and the result matches the corresponding hash preimage;
- (2) The contract includes the following logic: i. If the provided K_i successfully decrypts C_i and the resulting address satisfies the condition that $H(K_i)$ matches the expected value, the payment A_i is released to the seller i; ii. Otherwise, after a predefined timeout, the payment A_i is refunded to the buyer.

3.3.4 *Step 4.*

- The sellers submit the set of decryption keys $[K_1, K_2, \cdots, K_i, \cdots K_n]$ to the smart contract;
- The blockchain network verifies whether each K_i can correctly decrypt the corresponding C_i obtain the address, and whether $H(K_i)$ matches the expected hash. If all verifications pass, the smart contract releases the payment A_i to the respective seller i.

3.3.5 *Step 5.*

- If the key set passes verification, the seller i receives the payment amount A_i and the buyer decrypt the $address_i$ using $Decrypt(K_i, C_i)$;
- If the verification fails or the seller fails to submit the key K_i within the specified time, the payment A_i is refunded to the buyer.

In general, only one of the key or the data is exposed, and the data address is privately shared. Privacy refers to the data.

3.4 Properties of Our Yotta Scheme

In the previous discussion, we presented the first scheme for large-scale one-to-many batch data trading. In practice, however, there may be multiple buyers, each independently publishing their own evaluation function F, and sellers are free to sell their data to multiple buyers. Therefore, our Yotta scheme essentially supports many-to-many large-scale

data trading on blockchain systems. In the following, we revisit our proposed scheme to assess its compliance with the properties outlined in the SQUATS principles for Web 3.0.

- **Scalability:** In traditional data trading models, transactions between buyers and sellers are typically limited to atomic exchanges. In contrast, our approach enables many-to-many large-scale data trading and is even scalable to future yottabyte-scale scenarios, as our protocol could enable users to sell their data in batches. Furthermore, by leveraging recursive SNARKs [3], our protocol supports the aggregation of proofs from multiple sellers, allowing the buyer to verify all proofs in a single step, thereby significantly improving trading efficiency.
- **Quality:** In IPFS, each file is assigned a unique Content Identifier (CID), which is a cryptographic hash computed from the file's content. This content-based addressing ensures that identical files produce the same CID. In our protocol, we leverage zk-SNARKs to generate proofs that bind the CID to the actual data, demonstrating that the data stored in IPFS is exactly what passes the evaluation function. This not only guarantees data integrity but also enables easy verification of consistency and correctness during the data trading process. Thus, by leveraging zk-SNARKs, the proof π can be used to verify whether the data satisfies the specified evaluation function $F()$, thereby enabling verification of data quality.
- **Usability:** Our scheme also supports convenient horizontal enhancements and extensions. It can provide user-friendly APIs to facilitate ease of use.
- **Autonomy:** Our approach integrates seamlessly with blockchain smart contracts, requiring minimal manual labor or intervention.
- **Transparency:** The protocol records all transactions on the blockchain, making the entire trading process fully transparent and auditable [6]. All data hashes and zero-knowledge proofs are stored on-chain, ensuring that no party can tamper with the transaction history, thereby improving the trustworthiness and security of data trading for our Yotta.
- **Security:** In our scheme, we do not have an untrusted third party to access all the data. In our design, only one of the keys or the data is publicly revealed, while the data address is privately shared. Privacy here refers to the data itself; only the encryption key K_i is visible on the blockchain, preventing any third party from accessing the actual data content.

Based on the above discussion, we conclude that Yotta effectively satisfies all the essential properties required for large-scale data trading in the Web 3.0 era.

3.5 Discussions of Yotta

3.5.1 Complexity. We then discuss the complexity of our Yotta scheme.

Time Complexity: Our scheme improves trading efficiency in the following ways.

- Our approach ensures transparency and tamper-resistance by combining complete IPFS content addressing with zk-SNARK-based proofs, which feature constant proof size and computation cost that does not grow with the data volume.
- Proofs on the seller side: All required zero-knowledge proofs can be generated by the seller, including (1) Proof that the data passes the buyer's evaluation function. (2) Proof that the seller truly possesses the secret information in the dataset and that this information matches the content stored at the corresponding IPFS address. (3) Proof that the provided decryption key is indeed the correct key for decrypting the data address. This design shifts the computational burden away from the buyer, significantly reducing their time cost.

- Recursive SNARK Verification on the buyer side: We leverage recursive SNARKs to batch-verify proofs from multiple sellers, substantially lowering the computational resource consumption. This approach not only accelerates verification but also reduces its computational cost.

Space Complexity: Our scheme highlights the following methods to reduce communication overhead.

- Proof size: The transmitted proof is succinct, requiring only a small and constant amount of data regardless of the input size.
- Encrypted Data Storage: Sellers encrypt their data and store them on IPFS. Only the IPFS address and the corresponding decryption key need to be transmitted on blockchain systems, which significantly reduces the overhead of transferring large datasets.

Thus, the Yotta scheme holds significant promise for practical deployment, meeting all previously defined principles while maintaining both time and space efficiency.

3.5.2 Extensibility of Yotta. **Application-level**: Sellers can freely sell their secret information to multiple buyers in batches or sell multiple pieces of secret data to a single buyer as long as the seller properly defines the granularity of each secret item. **Technical-level**: Our scheme can be easily integrated with other emerging techniques, such as evaluation functions based on Shapley value for more accurate data quality assessment, more efficient decentralized storage systems beyond IPFS, or improved ZKP mechanisms.

4 Simulation Experiment

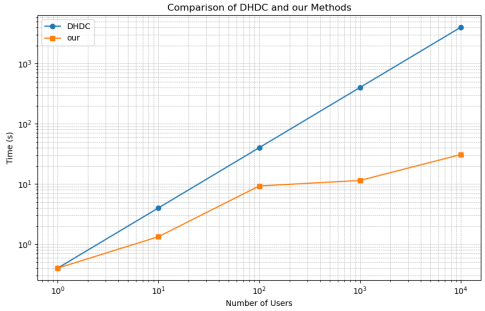

Fig. 2. Our simulation experiments with Yotta scheme prototype and DHDC.

As shown in Figure 2, we implemented a prototype of our method, Yotta, and compared it with the most practical existing data-secure approach—DCDH. In our experiments, we have one buyer and multiple sellers. We gradually increased the number of participating sellers (users), with the y-axis plotted on a logarithmic scale.

It can be observed that with 10 users, our method achieves approximately 3× the performance of DCDH. With 100 users, the improvement reaches around 12×, and with 1,000 users, about 35×. Notably, at 10,000 users, our method outperforms DCDH by approximately 130×. These results strongly demonstrate the effectiveness of our approach compared to existing solutions. Our method demonstrates increasing performance gains as the number of users scales up, which aligns with realistic scenarios where thousands or even tens of thousands of sellers may participate.

Therefore, we present a practical, sublinear solution for large-scale batch data trading. Our method shows significant promise for future real-world applications.

5 Application Discussions

Our Yotta scheme can be deployed in various Web 3.0 applications:

In supply chain management, different participants (e.g., suppliers, manufacturers, retailers) can encrypt their data and store it in IPFS, enabling data exchange via blockchain. Each participant can verify the integrity and correctness of the data, ensuring trust and transparency across the supply chain. Specifically, (1) manufacturers can provide production data, which retailers and consumers can verify through the blockchain to ensure product quality and authenticity [16]; (2) logistics providers encrypt and upload transport data to IPFS. All supply chain nodes can verify this data to ensure a transparent and reliable delivery process. Our Yotta could easily meet the requirements of supply chain management.

In the data market, participants can publish data requirements along with corresponding evaluation methods. Individuals or organizations can encrypt compliant datasets and upload them to IPFS, enabling secure data sharing and trading via blockchain. Buyers can verify the integrity and correctness of the data before receiving the decryption key, ensuring data trustworthiness. In particular, (1) financial institutions can sell encrypted market data, customer behavior data, etc. Buyers verify the data's integrity through proofs before obtaining the decryption keys; (2) hospitals and health research institutes can sell encrypted patient data. Researchers can purchase and verify the data before decrypting and using it [1]. Thus, our Yotta scheme is the ideal method that could fulfill all the demands of the data market.

6 Conclusion

Large-scale data trading is increasingly becoming a practical demand for participants in the Web 3.0 ecosystem. However, there is currently no practical solution available. In this work, we are the first to formalize the essential properties that a practical data trading scheme should satisfy, which we define as the SQUATS principles. Based on these principles, we design Yotta, a many-to-many large-scale data trading scheme. We provide a detailed explanation of Yotta's design, analyze its features and extensibility, and implement a prototype system. Our evaluation shows that Yotta outperforms existing methods by up to 130×, demonstrating its effectiveness and practical potential.

References

[1] Baoyi An, Mingjun Xiao, An Liu, Yun Xu, Xiangliang Zhang, and Qing Li. 2021. Secure crowdsensed data trading based on blockchain. *IEEE Transactions on Mobile Computing* 22, 3 (2021), 1763–1778.

[2] Juan Benet. 2014. Ipfs-content addressed, versioned, p2p file system. *arXiv preprint arXiv:1407.3561* (2014).

[3] Dan Boneh, Justin Drake, Ben Fisch, and Ariel Gabizon. 2020. Halo infinite: Recursive zk-SNARKs from any additive polynomial commitment scheme. *Cryptology ePrint Archive* (2020).

[4] Sean Bowe, Alessandro Chiesa, Matthew Green, Ian Miers, Pratyush Mishra, and Howard Wu. 2020. Zexe: Enabling decentralized private computation. In *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 947–964.

[5] Binyi Chen, Benedikt Bünz, Dan Boneh, and Zhenfei Zhang. 2023. Hyperplonk: Plonk with linear-time prover and high-degree custom gates. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 499–530.

[6] Fei Chen, Jiahao Wang, Changkun Jiang, Tao Xiang, and Yuanyuan Yang. 2022. Blockchain based non-repudiable iot data trading: Simpler, faster, and cheaper. In *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 1958–1967.

[7] Weiqi Dai, Chunkai Dai, Kim-Kwang Raymond Choo, Changze Cui, Deiqing Zou, and Hai Jin. 2019. SDTE: A secure blockchain-based data trading ecosystem. *IEEE Transactions on Information Forensics and Security* 15 (2019), 725–737.

[8] Xiangqian Dong, Bing Guo, Yan Shen, Xuliang Duan, YC Shen, and H Zhang. 2018. An efficient and secure decentralizing data sharing model. *Chinese Journal of Computers* 41, 5 (2018), 1021–1036.

[9] Gregory Entin. 2023. Web3 Basics: The Future of a Decentralized Web. https://www.linkedin.com/pulse/web3-basics-future-decentralized-web-gregory-entin-vsvuc/ Accessed: 2025-04-28.

[10] Ariel Gabizon, Zachary J Williamson, and Oana Ciobotaru. 2019. Plonk: Permutations over lagrange-bases for oecumenical noninteractive arguments of knowledge. *Cryptology ePrint Archive* (2019).

[11] IDC. 2018. The Digitization of the World: From Edge to Core. https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf Accessed: 2025-04-28.

[12] Ali Jadbabaie. 2014. IEEE Transactions on Network Science and Engineering. *IEEE Transactions on Network Science and Engineering* 1, 01 (2014), 2–9.

[13] Taeho Jung, Xiang-Yang Li, Wenchao Huang, Jianwei Qian, Linlin Chen, Junze Han, Jiahui Hou, and Cheng Su. 2017. Accounttrade: Accountable protocols for big data trading against dishonest consumers. In *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 1–9.

[14] Taeho Jung, Xiang-Yang Li, Wenchao Huang, Zhongying Qiao, Jianwei Qian, Linlin Chen, Junze Han, and Jiahui Hou. 2018. Accounttrade: Accountability against dishonest big data buyers and sellers. *IEEE Transactions on Information Forensics and Security* 14, 1 (2018), 223–234.

[15] David Krause. 2024. Web3 and the Decentralized Future: Exploring Data Ownership, Privacy, and Blockchain Infrastructure. *Privacy, and Blockchain Infrastructure (December 19, 2024)* (2024).

[16] Chunlin Li, SongYu Liang, Jing Zhang, Qiao-e Wang, and Youlong Luo. 2022. Blockchain-based data trading in edge-cloud computing environment. *Information Processing & Management* 59, 1 (2022), 102786.

[17] Tianyi Liu, Tiancheng Xie, Jiaheng Zhang, Dawn Song, and Yupeng Zhang. 2024. Pianist: Scalable zkrollups via fully distributed zero-knowledge proofs. In *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1777–1793.

[18] Tianyi Liu, Xiang Xie, and Yupeng Zhang. 2021. zkCNN: Zero knowledge proofs for convolutional neural network predictions and accuracy. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 2968–2985.

[19] Mozilla Foundation. 2024. Internet Health Report. https://foundation.mozilla.org/en/insights/internet-health-report Accessed: 2025-04-28.

[20] Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. (2008).

[21] Michael Rosenberg, Jacob White, Christina Garman, and Ian Miers. 2023. zk-creds: Flexible anonymous credentials from zksnarks and existing identity infrastructure. In *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 790–808.

[22] Unknown Speaker. 2025. Web3 & Decentralization. Talk at Bratislava OpenCamp 2025. https://pretalx.opencamp.sk/bratislava-opencamp-2025/talk/A8EJ8V/ Accessed: 2025-04-28.

[23] Ikbal Taleb, Mohamed Adel Serhani, and Rachida Dssouli. 2018. Big data quality assessment model for unstructured data. In *2018 International Conference on Innovations in Information Technology (IIT)*. IEEE, 69–74.

[24] Gavin Wood et al. 2014. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper* 151, 2014 (2014), 1–32.

[25] Li Yue, Huang Junqin, Qin Shengzhi, and Wang Ruijin. 2017. Big data model of security sharing based on blockchain. In *2017 3rd International Conference on Big Data Computing and Communications (BIGCOM)*. IEEE, 117–121.

[26] Shuli Zheng, Lixuan Pan, Donghui Hu, Meng Li, and Yuqi Fan. 2020. A blockchain-based trading platform for big data. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 991–996.

[27] Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang. 2017. An overview of blockchain technology: Architecture, consensus, and future trends. In *2017 IEEE international congress on big data (BigData congress)*. Ieee, 557–564.

[28] Guy Zyskind, Oz Nathan, et al. 2015. Decentralizing privacy: Using blockchain to protect personal data. In *2015 IEEE security and privacy workshops*. IEEE, 180–184.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009