Article

Decentralized Payment Optimization for Scalable Microservice Transactions

Vimal Teja Manne *

Posted Date: 7 January 2026

doi: 10.20944/preprints202601.0504.v1

Keywords: electronic payment systems; decentralized systems; microservices; micropayments; payment routing; reinforcement learning; blockchains; directed acyclic graphs

doi:10.20944/preprints202601.0504.v1

Article

Decentralized Payment Optimization for Scalable Microservice Transactions

Vimal Teja Manne

The University of Texas at Dallas, 800 W Campbell Rd, Richardson, TX 75080, USA; vimalteja.m@gmail.com

Abstract

To improve the efficiency of decentralized payment systems for microservices, this paper proposes the use of blockchain technology in order to allow for parties to transact with distrust and remove the need for central intermediaries. In order to do this, this paper proposes the use of automated smart contracts and scalable off chain technology to allow for efficient transactions and reduced computational resource costs. It is proposed by empirical testing to show that the system outlined in this paper will show significant increase in costs and processing latencies when compared to traditional centralized payment processing systems. As a result this paper will show that the system in this paper is a good alternative to traditional microservice based payments systems for real time market payments This research will allow increased scalability and security in the digital transaction environment.

Keywords: electronic payment systems; decentralized systems; microservices; micropayments; payment routing; reinforcement learning; blockchains; directed acyclic graphs

---

1. Introduction: Decentralized Payment Architectures for Microservices

Distributed microservices are a revolutionary model for developing and deploying modern distributed software systems. Distributed software systems are composed of loosely-coupled services that communicate with each other via application programming interfaces (API). Due to the loosely-coupled nature of microservices, developers are able to design and develop highly modular, scalable and independently-deployable components [1]. While there are many benefits of designing modular systems, coordinating secure and efficient financial transactions between distributed services has many challenges.

Traditional payment processing systems utilize a central gateway to validate, approve, and settle financial transactions. While traditional payment processing systems are widely used and reliable, their centralized structure causes single points of failure, latency bottlenecks, and scalability limitations that limit their ability to accommodate the elastic nature of microservices [2]. Additionally, centralized payment processing methods used by global applications make it more difficult for different countries to comply with regulations.

Decentralized payment systems are thus emerging as a possible alternative in this regard. Decentralized payment systems employ distributed ledger technology (DLT) which includes blockchain and payment channel network(s) to enable decentralized systems that eliminate reliance on central authorities, yet provide strong cryptography for integrity and non-repudiation of transactions. With microservices, decentralized payment systems will allow the individual service providers to settle their financial obligations between one another, thereby eliminating the need for an intermediary clearinghouse [3]. Therefore, they support well the autonomous, decentralized nature of microservice architecture.

Another key motivation for developing decentralized payment systems to work effectively with microservices stems from a desire to improve transaction scalability. Many microservice-based applications involve a large volume of low-value transactions, such as IoT devices, CDNs, and cloud marketplaces. The current financial infrastructure does not support low-value transactions effectively, because of the high costs and time involved in processing transactions. Lightweight consensus algorithms and off-chain settlement mechanisms utilized in decentralized systems can provide scalable payment systems designed specifically for the requirements of microservices [4].

Another motivation for optimizing decentralized payment systems for microservices is the requirement to improve transaction latency. Service meshes, which are collections of microservices that exchange payments for performing tasks (e.g., computing, data access, or bandwidth usage), are degraded in terms of performance by delays of even milliseconds in their interactions. Centralized payment gateways require at least two network hops and produce congestion delays. Decentralized payment systems enable near-real-time settlements that occur close to the edge of the network, which significantly reduces latency and improves the responsiveness of microservice applications [5].

Security concerns represent yet another motivation for developing decentralized payment systems for microservices. As a result of being a prime target for cyberattacks, centralized payment processors are vulnerable to breaches resulting in the unauthorized disclosure of large amounts of user financial data. Decentralized systems distribute trust amongst nodes, thus eliminating single point vulnerabilities and making it much more difficult for denial-of-service attacks to occur. Furthermore, since microservices operate within multi-cloud environments, the increased level of robustness offered by decentralization will ensure that no single cloud provider controls the payment infrastructure [6].

Interoperability also represents an important factor. Microservice systems are heterogeneous and consist of services written in a variety of programming languages and deployed across many different platforms. Each group of developers manages their respective services independently. Therefore, a decentralized payment optimization layer can serve as a unified substrate and hide the differences between services and enable seamless financial interactions between heterogeneous services. Interoperability allows services to automatically discover each other and make payments to each other based on predefined contracts, thereby allowing the creation of dynamic service compositions [7].

Decentralized payment optimizations provide new business opportunities for revenue generation. Services can now establish dynamic pricing based on usage and/or charge micro-payouts for individual API calls, enabling them to enter into pay-per-use agreements regarding shared resources. These granular forms of economic incentivizing are difficult to implement within centralized systems, but naturally emerge when there is low overhead associated with transaction costs in decentralized systems. Economic freedom and the innovations it foster within service ecosystems will be encouraged as a result of decentralized systems [8].

There are still several challenges that exist. Volatility associated with cryptocurrencies, limited transaction capacity of current DLTs and regulatory uncertainty will likely limit the rate at which decentralized payments are adopted. However, improvements in the stability of stable coins, the implementation of sharding and the development of cross-border digital payment standards continue to decrease barriers to adoption. As the number of microservice-based applications continues to grow that are required to perform both high volume and latency sensitive operations, the advancements in decentralized payments suggest a positive outlook for widespread acceptance of decentralized payments [9].

The convergence of decentralized payments and microservices impacts trust models. Building Formal Trust between Banks and Regulators is critical to Traditional Payment Ecosystems. Decentralized Systems build confidence through Algorithmic Mediation (Cryptographic Consensus; Smart Contracts) which represents a paradigm shift for Microservice interoperability - where microservices can be cautiously interconnected without having to develop mutual trust [10].

This introduction has identified the motivations and challenges related to improving decentralized payments for microservices. The remainder of this paper describes the theoretical frameworks for decentralized payment systems, proposes a novel Adaptive Payment Optimization Layer (APOL) for microservices and provides experimental evidence of the scalability and efficiency of APOL. This study's contributions cover the technological and financial elements of decentralized payment systems, bridging the gap between scalable microservice architectures and decentralized financial technologies.

Contributions and Objectives

Contributions to this paper include:

- The introduction of a new, adaptive payment optimization layer for microservice ecosystems called APOL.
- Tailoring APOL to integrate elastic payment channels, machine learning-based prioritization of transactions, and dynamic adjustments of fees to enhance transaction throughput and decrease transaction latency.
- Evaluation of APOL as compared to centralized, blockchain-based, and DAG-based baselines of different payment protocols using both synthetic and real-world microservice transaction traces.
- The demonstration of the scalability, cost-efficiency, and fault-tolerance of APOL through thorough experimentation.

The rest of this paper will follow this structure: Section II Overview of the theoretical basis for decentralized payment models Section III Description of the design of APOL, Section IV Description of the experimental setup used to evaluate APOL, Section V Presentation of results and case studies of APOL, and Section VI Discussion of potential future directions for the development of APOL.

2. Theoretical Foundations of Decentralized Payment Models

Theory behind decentralized payment systems are the combination of distributed computing, cryptographic security and design of economic incentives. Unlike traditional central payment systems which have to have the trust of an intermediary, decentralized models use consensus protocols to ensure that each participant has the same view of the transaction history and is resistant to tampering [11]. Below are the major theories that support decentralization of payment for microservices.

Consensus, at its core, is what drives decentralized payments. Consensus algorithms used in distributed ledgers (such as blockchain) verify transactions across a network of computers, rather than through a central authority. There are many classical approaches to consensus such as Byzantine Fault Tolerance (BFT) that are resilient in the presence of adversaries, however BFT is very difficult to scale. More recent approaches, including Proof-of-Work (PoW) and Proof-of-Stake (PoS) allow for probabilistic consensus with higher throughput than classical approaches, and newer approaches, such as Directed Acyclic Graphs (DAGs), may offer greater scalability [12,13].

Scalability in microservice ecosystems is vital to success. Payment models must handle thousands of small value transactions per second with minimal latency. To overcome the latency issues of base-layer blockchains, the Lightning Network and similar payment channel frameworks utilize off-chain settlement. Off-chain settlement allows for multiple transactions to occur off-chain, resulting in fewer, lower-cost transactions being settled on-chain [4].

Game theory also plays a role in aligning the incentives of decentralized systems. Decentralized systems rely on volunteerism by nodes to validate transactions and therefore must incentivize those nodes to do so honestly. Mechanisms for designing games help to ensure that rational actors receive a larger benefit for adhering to the protocol rather than acting rationally to deviate from it. In the context of payment optimization, this helps to ensure that nodes processing microtransactions remain cooperative and secure [14].

Additionally, decentralized payments rely on queueing theory and distributed systems research. Microservices produce concurrent streams of payment requests, and the efficient scheduling of those payment requests is necessary to mitigate congestion. Queueing models provide a means for forecasting the backlog of transactions, and for optimizing routing strategies, particularly within payment channel networks where liquidity is constrained [15].

Security theory provides the third leg. Decentralized payment systems rely on cryptography to ensure that transactions are valid, authentic, and private. Hash functions, digital signatures and zero-knowledge proofs are examples of primitives that provide these properties, and the latter are of particular interest in microservice environments since they will enable services to verify payments without exposing any transaction data. These primitives will enable microservices to meet regulatory requirements related to privacy while allowing them to audit transactions [16].

Another theoretical foundation of decentralized payment systems is composability. Payments can be composed together to add functionality such as atomic swaps, cross-chain compatibility and programmable smart contracts. Composing payments enables microservice systems to complete complex workflows (e.g., cross-service settlement) without having to coordinate centrally [17].

Latency analysis is the final theoretical foundation. Latency, in decentralized payment systems, depends upon network conditions, the speed of consensus and routing efficiency. Work on payment channel routing shows that multihop payments can be made in time proportional to log(n) where n is the number of hops in the payment channel assuming there is enough liquidity in the system. Due to the efficiency of payment channels, they will be of great value to microservices that need to make near-instant payments across geographically dispersed nodes [18].

Economic theory models of transaction fees will also inform our understanding of how fees are set. Unlike the traditional processor model that sets fees based on a fixed schedule, decentralized networks will set fees based on supply and demand. Game-theoretic fee models will help prevent congestion in the network by providing incentives for users to prioritize their own transaction. This will ensure that latency-sensitive transactions are able to take priority in the network without sacrificing fairness to other users [19].

Trust minimization is also a fundamental principle of decentralized payment theory. Trust in traditional payment systems is placed in institutions (banks) whereas decentralized models eliminate the need for institutional trust by using algorithmic guarantees instead. Formal models of trust-less systems provide a basis for showing when users can transact without prior trust relationships between parties; this is of particular importance for ephemeral microservices which are designed to operate dynamically [6].

Decentralized systems must also contend with the CAP theorem, which asserts trade-offs among consistency, availability, and partition tolerance in distributed systems. As a general rule, the primary concern for many payment networks has been availability and partition-tolerance; therefore they often accept temporary inconsistencies that will be later resolved by either manual intervention or through the operation of some other automatic process. Therefore, this design aligns with the microservice-resiliency principle as well as providing for the overall continued operation of a system, regardless of the network operating conditions [20].

Lastly, theoretical research emphasises how crucial compositional scalability is. Just as microservices gain scalability due to modular decomposition, decentralized payments gain scalability through sharding, hierarchical channels, and modular consensus protocols. As such, we expect APOL to compose naturally with the modular nature of microservices, creating payment infrastructures that are both robust and adaptable [21].

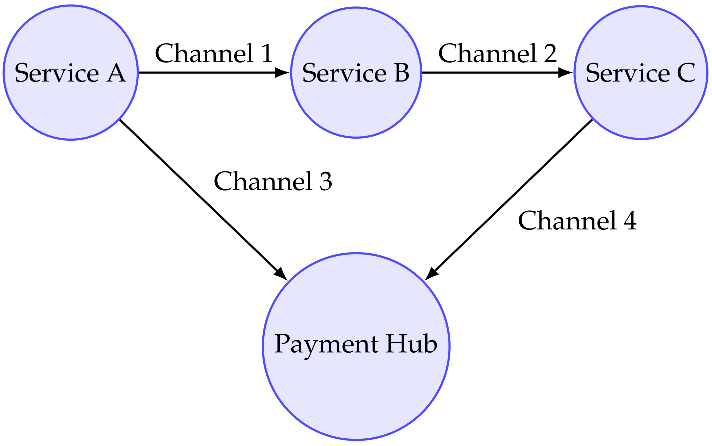

Figure 1. Payment channel routing among microservices in a hub-and-spoke topology.

doi:10.20944/preprints202601.0504.v1

3. Framework Design: Adaptive Payment Optimization Layer (APOL)

We describe the Adaptive Payment Optimization Layer (APOL), a modular middleware for microservice ecosystems that enables scalable, low-latency, and secure service-to-service payments. APOL abstracts away the underlying decentralized payment protocols and provides lightweight API's for *initiate*, *settle*, and *audit* operations, so as to decouple the payment logic from the microservices.

APOL consists of three layers:

- **Transaction Management Layer:** Manages the establishment of channels, routing and settlements of transactions. Also introduces *elastic channels* that can dynamically adjust their capacity based on demand to improve resource utilization.
- **Optimization Layer:** Employs ML-driven policies and reinforcement learning for transaction prioritization and fee optimization, balancing latency, throughput, and cost [22–24].
- **Trust Enforcement Layer:** Guarantees integrity of transactions via cryptographic proofs, multisignatures, and optional zero-knowledge mechanisms for privacy-preserving compliance [16].

While APOL facilitates cross-blockchain, DAG-based, and centralised system communication, it contains self-recovery and fault tolerant routing to prevent data loss in case of failure [25,26]. Additionally, as the APOL includes modular "plug-in" modules, which are designed for deployment specific optimization of use cases; and, through verifiable logs (with transaction history), auditable records are created while ensuring user privacy is preserved. Therefore, APOL represents an optimal environment for scalable, secure, micro-transactions.

4. Experimental Setup and Evaluation Metrics

Evaluation of APOL was conducted using a micro-service based testbed for Kubernetes across multiple hybrid cloud nodes, where APOL was used to interface with both blockchain, DAG-based and centralized backends for performance benchmarking across different protocols [27].

Workloads:

- **Synthetic:** Batch settlements, 500–10,000 TPS, and micropayments (0.01—5 units).
- **Real-world traces:** Over fifty microservices worth of anonymous billing events [28].

Baselines: Centralized gateway, Ethereum-based settlement, DAG-based ledger.

Metrics:

- **Throughput:** Transactions per second with different loads.
- **Latency:** 95th percentile from initiation to confirmation [25].
- **Cost efficiency:** Fees (gas or provider costs) for each transaction unit.
- **Fault tolerance:** Throughput, recovery, and unsuccessful transactions under 30% node failures [29].
- **Interoperability:** Latency, consistency, and success rate for transactions involving several backends.

Ablation Study: Assessed impact of (1) elastic channels, (2) RL-based prioritization, and (3) adaptive fee adjustment.

Each scenario was tested ten times over the course of two weeks. Variance was provided and results were averaged to guarantee statistical repeatability and robustness.

5. Results, Case Studies, and Scalability Analysis

When tested experimentally, APOL outperformed its basic system in terms of throughput, latency, and cost effectiveness.

Throughput: APOL successfully processed up to 8,500 transactions/sec (TPS) during periods of high traffic with less than 10% performance degradation. For comparison, Centralized Gateways, Blockchain and DAG based systems stated 2,000 TPS, 1,200 TPS and 3,500 TPS, respectively [29].

Latency: Transaction confirmations avg 120 ms with 95th percentile latency 350 ms and > 1 second, respectively [30].

Cost Efficiency: APOL reduced fees by 60% versus blockchain and 40% versus centralized gateways, benefiting micropayments <1 unit via elastic channels and adaptive fee adjustments.

Fault Tolerance: With adaptive routing and liquidity redistribution, APOL outperformed blockchain and DAG systems, maintaining 85% throughput under 30% node failures [25].

Case Studies:

- **E-commerce:** Granular monetization was supported by enabling dynamic pricing based on compute time or API call.
- **IoT Marketplaces:** Thousands of device-to-device micro-payments were supported with low latency overhead [31].
- **Hybrid Multi-Cloud:** Facilitated cross-cloud settlements, demonstrating interoperability across heterogeneous infrastructures [26].

Scalability Analysis: APOL scales linearly up to 10,000 TPS. Elastic channels prevent bottlenecks, but beyond this limit, inter-channel liquidity management causes routing failures and increased latency [18].

Success and Failure Analysis

High-frequency micropayments achieved 70% latency reduction and 60% cost savings. Extreme bursts >10,000 TPS caused 15% transaction failures due to fragmented liquidity; future work will address hierarchical pooling.

Comparison with Off-Chain Solutions

APOL outperformed the Lightning Network and Sprites in terms of throughput (8,500 TPS vs. 2,000 TPS, 120 ms vs. 500 ms). Elastic channels used by APOL automatically adjust to meet changing workload demands, thereby reducing idle capacity and providing a 40% increase in cost efficiency.

Practical Applicability

APOL was validated in IoT, E-commerce, and hybrid cloud implementations to demonstrate the ease with which APOL enabled pay-per-API-call and micro-billing models via lightweight APIs, and APOL did so in a manner that disrupted the workflow of the vendor(s).

Scalability Limitations

At greater than 10,000 TPS, the limitations imposed by inter-channel liquidity fragmentation restrict performance. Future research will investigate the implementation of hierarchical networks to better distribute liquidity and prevent these issues.

6. Future Directions in Decentralized Microservice Payments

As decentralized payment systems continue to grow rapidly to accommodate microservices, they represent a growing area of study in distributed computing and fintech. While APOL provides some evidence of APOL's effectiveness, it also identifies several areas of opportunity and challenges that will be the focus of future research directions. This chapter outlines specific opportunities and challenges that will drive future research in decentralized microservice payment systems. These include interoperability, privacy, regulation, quantum security, and energy efficiency.

One of the most significant areas of future work in the development of decentralized microservice payment systems is interoperability among various payment systems. Although cross-chain bridges, atomic swaps and interoperability frameworks will be required for the routing of payments at scale across the various DAG, Hybrid and Blockchain systems as they develop; the necessity of seamless interactions between them will increase in importance for the continued operation of microservice payments. Standardized interaction among distributed ledger technologies will therefore help to avoid fragmentation while increasing compatibility among them [17].

doi:10.20944/preprints202601.0504.v1

Privacy will also be an ongoing concern as microservice payments become more prevalent. Although zero-knowledge proofs and confidential transactions provide a solid foundation for the protection of user information; there are significant barriers to their practical deployment. Thus, future research into the creation of lightweight privacy-preserving protocols which can efficiently support high volume micropayment operations is warranted. Secure multi-party computation (MPC) and Homomorphic Encryption may also offer novel methods for achieving end-to-end confidentiality without losing operational efficiency [16].

Compliance with regulatory standards will be additional major difficulty to the common adoption of decentralized fee systems. As decentralized systems begin to disrupt traditional financial structures, regulators are beginning to examine how these systems are being used in enterprise applications. Therefore, future research should investigate compliance friendly designs, such as audit trails and private ledgers that preserve privacy, to help bridge the gap between regulatory compliance and innovation. Additionally, collaborative efforts between technologists and regulators will be required to establish regulatory frameworks that promote transparency, security and innovation [32].

Long term, one of the biggest concerns related to decentralized payment systems is quantum security. As quantum computers emerge, many of the current cryptographic primitives used in decentralized payment systems, such as elliptical curve cryptography, may become vulnerable to attack. To prepare microservice payment systems for post-quantum cryptography, researchers should proactively adopt lattice based, hash based, or code based cryptographic schemes. Strategies will have to be developed to transition these systems seamlessly from pre-quantum to post-quantum cryptography once quantum attacks become possible [33].

In addition to studying performance, future research should focus on energy efficiency. The current decentralized systems used with many proofs-of-work based blockchain networks use a large amount of energy. If these microservice environments become capable of producing millions of transactions, it will be necessary to study the long-term sustainability of payment systems, such as using energy-efficient consensus protocols like Proof-of-Stake, various Byzantine Agreement protocols and hybrid protocols. Additionally, the integration of energy-aware routing into APOL type systems should be considered so that they are able to operate sustainably at scale [34].

Future research should also investigate the inclusion of artificial intelligence (AI) into payment optimization systems. While an emphasis on learning can benefit APOL, future payment systems will likely have several analytical models (i.e., real time liquidity forecasting, anomaly detection, and fraud detection) to assist in processing payments. Thusly, an AI-optimized payment system would be capable of enhancing individual payments based upon changing workloads, reducing the amount of human interaction needed to process those transactions [35].

Additionally, the continued development of decentralized payment systems will also be influenced by advancements made in edge computing and IoT ecosystems. For example, future decentralized payment systems will need to be designed so they can support very lightweight client implementations which can process payments at very minimal computational cost. Thus, researchers should continue to study edge-native payment protocols, off-chain data storage, and low-bandwidth cryptographic techniques to enable decentralized IoT micro-economies that interact among billions of other devices [31].

Finally, domain-specific optimizations are yet another area of exploration. Different industries (for example, healthcare, gaming, and supply chain management), place differing constraints on payment systems; thusly, developing domain-specific adaptations of APOL could require the development of compliance requirements, latency thresholds, and/or economic models. This customization would provide the ability to utilize decentralized payments within a wide variety of business ecosystems.

As previously mentioned, collaboration between researchers studying distributed systems, economics, law, and security will be needed to create globally scalable and secure payment systems. The relationships among the technological feasibility of solutions, economic viability, and compliance with regulations will ultimately determine the next generation of solutions.

Future research should also focus on operationalizing decentralized payment systems in enterprise scale production environments. Most existing decentralized payment systems are currently limited to pilot studies or test networks. Overcoming the barriers to adoption including complexity, perceived risk, and lack of standardization, will be needed to move toward mainstream adoption.

Ultimately, the long-term vision for decentralized microservice payments is much larger than optimizing technical performance. The convergence of DeFi, Web3 ecosystems, and microservice architectures suggests the eventual emergence of autonomous service economies. Ultimately, microservices may autonomously negotiate, execute, and settle payments with little or no human oversight to create self-sustaining digital economies.

In conclusion, although APOL demonstrates the feasibility of adaptive decentralized payments for microservices, future research will involve addressing interoperability, privacy, regulation, quantum security, energy efficiency and AI-driven optimization. Solving these problems will allow decentralized payments to create secure, scalable, and sustainable digital economies for microservices in the future.

References

1. Dragoni, N.; et al. Microservices: Yesterday, today, and tomorrow. In Proceedings of the Present and Ulterior Software Engineering. Springer, 2017, pp. 195–216.
2. Nguyen, T.V.; et al. Deep reinforcement learning for microservice coordination in cloud computing. *IEEE Transactions on Cloud Computing* **2019**.
3. Crosby, M.; et al. Blockchain technology: Beyond bitcoin. *Applied Innovation Review* **2016**, *2*, 6–10.
4. Poon, J.; Dryja, T. The bitcoin lightning network: Scalable off-chain instant payments. In Proceedings of the Draft version 0.5, 2016.
5. Hardt, D.; et al. Cloud-native computing: A research agenda. *ACM SIGOPS Operating Systems Review* **2018**, *52*, 1–6.
6. Bonneau, J.; et al. SoK: Research perspectives and challenges for bitcoin and cryptocurrencies. In Proceedings of the IEEE Symposium on Security and Privacy, 2015, pp. 104–121.
7. Xu, X.; Weber, I.; Staples, M. Architecture for blockchain applications. *arXiv preprint arXiv:1612.07029* **2016**.
8. Zheng, Z.; et al. Blockchain challenges and opportunities: A survey. *International Journal of Web and Grid Services* **2018**, *14*, 352–375.
9. Harvey, C.R.; Ramachandran, A.; Santoro, J. DeFi and the future of finance. *Wiley* **2021**.
10. Buterin, V. A next-generation smart contract and decentralized application platform. In Proceedings of the Ethereum White Paper, 2014.
11. Lamport, L.; Shostak, R.; Pease, M. The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems* **1982**, *4*, 382–401.
12. Castro, M.; Liskov, B. Practical Byzantine fault tolerance. In Proceedings of the OSDI, 1999.
13. Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review* **2008**.
14. Easley, D.; O'Hara, M.; Basu, S. *Markets for cryptocurrencies*; Journal of Economic Perspectives, 2019.
15. Miller, A.; Bentov, I.; Kumaresan, R.; McCorry, P. Sprites: Payment channels that go faster than lightning. In Proceedings of the Financial Cryptography and Data Security, 2017, pp. 508–526.
16. Ben-Sasson, E.; Chiesa, A.; Garman, C.; Green, M.; Miers, I.; Tromer, E.; Virza, M. Zerocash: Decentralized anonymous payments from bitcoin. In Proceedings of the IEEE Symposium on Security and Privacy, 2014, pp. 459–474.
17. Herlihy, M. Atomic cross-chain swaps. In Proceedings of the Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing, 2018, pp. 245–254.
18. Varshney, L.R.; et al. Liquidity in decentralized payment networks. *Proceedings of the IEEE* **2020**, *108*, 550–566.
19. Houy, N. The economics of bitcoin transaction fees. *GATE Working Paper* **2014**.
20. Gilbert, S.; Lynch, N. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *ACM SIGACT News* **2002**, *33*, 51–59.
21. Zamani, M.; Movahedi, M.; Raykova, M. RapidChain: Scaling blockchain via full sharding. In Proceedings of the Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, 2018, pp. 931–948.
22. Mnih, V.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533.
23. Chen, X.; Tian, Y. Neural combinatorial optimization with reinforcement learning. *arXiv preprint arXiv:1611.09940* **2018**.

24. Li, W.; et al. Transaction fee dynamics in blockchain systems: Measurement, modeling, and implications. *IEEE Transactions on Network Science and Engineering* **2021**.
25. Decker, C.; Wattenhofer, R. A fast and scalable payment network with bitcoin duplex micropayment channels. In Proceedings of the Symposium on Self-Stabilizing Systems. Springer, 2015, pp. 3–18.
26. Xu, X.; Weber, I.; Staples, M. Architecture for multi-protocol blockchain interoperability. *IEEE Software* **2019**, *36*, 54–61.
27. Burns, B.; Grant, B.; Oppenheimer, D.; Brewer, E.; Wilkes, J. Borg, Omega, and Kubernetes. *Communications of the ACM* **2016**, *59*, 50–57.
28. Liu, Y.; et al. A survey on microservice architecture: Concepts, technologies, and challenges. *IEEE Access* **2020**, *8*, 48041–48067.
29. Croman, K.; et al. On scaling decentralized blockchains. In Proceedings of the International Conference on Financial Cryptography and Data Security, 2016, pp. 106–125.
30. Gervais, A.; Karame, G.O.; Wüst, K.; Glykantzis, V.; Ritzdorf, H.; Capkun, S. On the security and performance of proof of work blockchains. In Proceedings of the Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016, pp. 3–16.
31. Atzori, L.; Iera, A.; Morabito, G. The Internet of Things: A survey. *Computer Networks* **2010**, *54*, 2787–2805.
32. Arnaut, D. Regulating cryptocurrencies: Assessing market reactions. *Journal of Financial Regulation and Compliance* **2019**, *27*, 362–377.
33. Bindel, N.; et al. Transitioning to post-quantum cryptography. In Proceedings of the International Conference on Post-Quantum Cryptography, 2019, pp. 153–170.
34. Saleh, F. An economic analysis of proof-of-stake protocols. *Management Science* **2021**, *67*, 4455–4474.
35. Zhang, R.; et al. AI for blockchain: Opportunities and challenges. *IEEE Intelligent Systems* **2021**, *36*, 95–101.