Decentralization and the Law of the Jungle: An Empirical Investigation of Ethereum's Market Mechanism*

Hanna Halaburda† Daniel Obermeier‡

April 10, 2025

Abstract

Blockchain technology aims to disintermediate traditional platforms by replacing centralized governance with decentralized market mechanisms. However, this shift introduces a tradeoff: while decentralization removes the platform's ability to capture value, it also eliminates platform-wide mechanisms—such as subsidies, curation, or pricing strategies—that can support long-term platform performance. Using Ethereum as a case study, this paper examines how its market-based transaction validation system affects the allocation of transaction capacity and shapes platform dynamics. Specifically, we estimate demand elasticities across more than 1,500 decentralized applications (dApps) and evaluate the effects of transaction fees on various application categories. To address endogeneity concerns, we leverage Ethereum's "difficulty bomb," a protocol feature that periodically reduces transaction throughput, as an instrumental variable. This method provides exogenous variation in transaction costs, allowing us to identify differences in demand elasticities across dApps. Our analysis shows that during periods of high fees, low-elasticity transactions—such as those associated with exploitative miner extractable value (MEV) activities—tend to dominate the network. This crowding out effect disproportionately impacts fee-sensitive categories of applications—such as gaming, social, and utility dApps—reducing their viability and shifting transaction capacity toward applications that prioritize short-term profitability over long-term platform performance. This dynamic helps explain why, despite repeated promises to disintermediate platforms like Uber, decentralized platforms may struggle to support fee-sensitive but socially valuable use cases.

Keywords: blockchain, decentralization, platform governance, market design, decentralized applications, Ethereum

JEL Codes: C26, D47, D62, L11, L14, L86, O33

*We thank Yannis Bakos, Eric Budish, Chris Conlon, Neil Gandal, Guillaume Haeringer, Joachim Henkel, Aija Leiponen, Natalia Levina, Raveesh Mayya, Foster Provost, Marc Rysman, Dan Waldinger and participants of the NYU Blockchain research group for their valuable discussions and comments. Further, we thank all participants of the TUM Blockchain Workshop 2021, CIST 2022, WISE 2022, NBER Digitization Workshop 2023, and the EU-DPRN 2023 for their feedback.This research was supported by a 2022 NBER Economics of Digitization Small Grant.

†NYU Stern School of Business; 44 W 4th Street, New York, New York 10012; email: hhalaburda@gmail.com

‡Nova School of Business and Economics; daniel.obermeier@novasbe.pt

1 Introduction

Over the past decade, blockchain technology has been heralded as a transformative innovation with the potential to disintermediate traditional digital platforms. The blockchain ecosystem has grown rapidly, with the global market size for blockchain technology reaching $28 billion in 2022 and projections of $825 billion by 2032.[1] Platforms like Ethereum enable decentralized applications (dApps) that operate without centralized oversight. For instance, decentralized finance (DeFi) applications like Uniswap, Aave, and Curve allow users to borrow, lend, or trade assets directly with peers, eliminating the need for intermediaries such as banks. Similarly, Ethereum Name Service offers decentralized identity and naming infrastructure, without centralized registrars like ICANN or DNS providers. And game CryptKitties demonstrated that user-owned digital assets and gameplay could bypass traditional gaming platforms. Advocates envision extending this model even further — a common example is a decentralized alternative to Uber, where drivers and riders connect directly, avoid 30% platform fees, and collectively govern the system. By replacing centralized governance with decentralized market mechanisms, blockchain platforms promise transparency, inclusivity, and openness. While there are several reasons why a decentralized application may fail — it may be technically flawed or conceptually incoherent — we focus on how decentralized market mechanisms can undermine the viability of applications that could work in principle, thereby endangering the long-term performance of blockchain platforms.[2]

This risk arises because, unlike centralized platforms such as Apple's iOS, which actively gatekeep, curate and subsidize access to support wide range of user activity to drive engagement and complementarity, decentralized platforms rely on market-driven allocation

[1]https://www.fortunebusinessinsights.com/industry-reports/blockchain-market-100072

[2]We define a decentralized application as "working" if it satisfies three conditions: (1) it is technically implementable via smart contracts, (2) it is incentive-compatible without centralized enforcement, and (3) it is usable at scale under real-world blockchain conditions. Many dApps (including the examples in this paragraph) are demonstrably viable in principle: they meet conditions (1) and (2) and generate meaningful user value. This paper focuses on condition (3) and how the allocation of transaction capacity affects the viability of such applications, and long-term performance of the platform.

through transaction fees. Moreover, as we argue, credible blockchain decentralization requires throughput constraints enforced by the protocol. This raises an economic tradeoff between decentralization and platform governance: while decentralization removes the platform's ability to capture value, it may also disproportionally favor financially extractive activities — such as arbitrage and miner extractable value (MEV) transactions — over applications that generate broader, long-term value. In the case of the hypothetical decentralized alternative to Uber, a largre proportion of everyday ride requests could be driven out by automated agents or bots that extract profit by preempting, reselling, or manipulating ride allocations. In such a setting, the absence of central control lowers rents but may also weaken the platform's ability to ensure availability, affordability, and diversity of service. This dynamic illustrates a broader paradox: the very mechanisms that enable decentralization can also create inefficiencies and distortions — undermining not just individual applications, but the platform's resilience and long-term performance. Can decentralized platforms function as viable alternatives to traditional intermediaries, sustaining a diverse set of transactions, including fee-sensitive but socially beneficial ones? Or does their reliance on market-based allocation inherently favor short-term profitability at the expense of long-term platform performance? Understanding whether these platforms can serve as a foundation for the next generation of digital markets is critical for assessing their economic significance.

This paper explores an inherent paradox in blockchain platforms: while disintermediation ostensibly limits the monopolistic power of platform providers, it also removes governance tools that are critical for orchestrating a diverse and healthy ecosystem of complements. A critical feature of blockchain protocols is their deliberate limitation of transaction supply to maintain decentralization. Without such limits, validators with the most powerful computational resources could dominate by increasing the throughput of validated, executed, and stored transactions to levels that exceed the capacity of less powerful machines, effectively centralizing control over the blockchain's core operations. Relying solely on market mechanisms to allocate a limited transaction supply may foster inefficiencies, such as crowding out entire categories of applications and reducing platform diversity. The price-driven crowding-out of certain applications provides insight into the role that governance mechanisms—absent in decentralized platforms but present in centralized ones—might play in shaping application variety. These dynamics raise fundamental questions about whether decentralized platforms can function as general-purpose infrastructure while maintaining the inclusivity and variety envisioned by Web 3.0.

We use Ethereum—the most prominent blockchain platform for decentralized applications (dApps)—as a case study. A comprehensive dataset of daily transaction-level and application-level data from Ethereum, covering over 1,500 dApps across various categories, enables us to estimate demand curves for transactions, measure their sensitivity to rising transaction fees, and analyze how variations in transaction fees impact the long-term viability of different applications. To address endogeneity concerns, we employ Ethereum's difficulty bomb, a protocol feature that increases mining difficulty over time, as a novel supply-side instrument. The difficulty bomb's activation is predetermined within Ethereum's protocol and unrelated to immediate market conditions. Importantly, it was reset three times arbitrarily and without prior announcements, creating unexpected exogenous changes in transaction supply. By requiring greater computational effort during activation periods, the difficulty bomb reduces transaction capacity, thereby creating exogenous variations in transaction fees. This ensures that its effects on transaction costs are not influenced by changes in user behavior or dApp-specific factors, satisfying the exclusion restriction necessary for valid instrumental variable analysis.

The observed variations in demand elasticities highlight significant disparities in how different dApp categories compete for transaction capacity. Applications with higher elasticity—often fee-sensitive but socially valuable—are crowded out by low-elasticity transactions, including those resulting from miner extractable value (MEV) activities. MEV refers to transactions that exploit the transparent nature of blockchain systems, allowing validators or bots to reorder, insert, or exclude transactions for profit. For example, front-running in blockchain systems mirrors practices in traditional financial markets, where brokers or traders use advanced knowledge of pending orders to execute their own transactions first, capturing profits at the expense of others. These transactions, often involving front-running or sandwich attacks, drive up fees by bidding aggressively. Their high willingness to pay stems from their exploitative nature, as they capture value at the expense of other users. This aggressive bidding dominates the limited transaction capacity, effectively crowding out fee-sensitive but socially valuable applications. This crowding-out effect reduces the variety of dApps and prioritizes short-term revenue extraction over long-term platform viability, underscoring a critical inefficiency in Ethereum's market-based design. Under centralized governance, platforms can internalize these externalities by directly curating or subsidizing applications to sustain a broader ecosystem.

Our contributions are threefold. First, we use Ethereum's difficulty bomb as a novel instrument to address endogeneity concerns, enabling us to estimate demand curves for different dApp categories and reveal their varying elasticities. Second, we empirically demonstrate that the combination of limited capacity and the market mechanism for transaction validation on Ethereum creates a trade-off between allocative efficiency and platform diversity. Specifically, the allocation of transaction capacity systematically favors low-elasticity transactions that prioritize immediate profitability. As a result, fee-sensitive but potentially high-value applications are crowded out, reducing diversity and innovation on the platform. Third, we examine the broader implications of this dynamic, including the dominance of exploitative behaviors such as miner extractable value (MEV) activities. MEV undermines the inclusivity and fairness promised by decentralized platforms, further exacerbating the inefficiencies of Ethereum's market-based design.

This study advances the economic literature on platform governance and market-based resource allocation. By seeing blockchain platforms as a natural experiment in decentralized market design, we provide new insights into how market mechanisms influence platform ecosystems, challenging prevailing assumptions about the self-regulating nature of decentralized markets.

While Ethereum serves as the context for our analysis, its transaction validation system is emblematic of decentralized blockchain platforms more broadly. Like Ethereum, these platforms enforce supply constraints to maintain decentralization, frequently using auction-based fee structures to allocate transaction capacity based on willingness to pay. These mechanisms ensure decentralization but also create systemic inefficiencies and equity challenges. Our findings about efficiency trade-offs and platform dynamics thus apply broadly across decentralized blockchain ecosystems, highlighting the foundational role of capacity constraints and fee structures in shaping both their benefits and limitations. By shedding light on the systemic challenges posed by market-based transaction allocation, we aim to inform both academic discourse and policy debates about the design and regulation of emerging decentralized economies.

The remainder of this paper is structured in the following way: Section 2 explains how we relate and contribute to the existing literature. Section 3 introduces the context of our study, describes all necessary details to understand the process of transacting with an application on the Ethereum blockchain, and conceptualizes Ethereum as a market for transactions. Section 4 describes our conceptual framework. Section 5 summarizes our data. Section 6 discusses the empirical strategy to identify the demand curves for different types of applications. Section 7 reports the results of our analysis. Finally, Section 8 discusses our study's implications and limitations.

2 Related literature

Decentralized digital platforms existed already before the advent of blockchain technology. Therefore, to understand how blockchain platforms provide a novel way to decentralized digital platforms, we briefly discuss how they compare to centralized and existing decentralized platforms before we move on to discussing our work's theoretical foundations.

Blockchain platforms are transaction-based platforms that substitute a central authority with a network of peers who collectively validate, enforce, and record transactions (Halaburda et al. 2022). However, besides the collective validation, enforcement, and recording of transactions, blockchain platforms also devolve all other platform governance decisions to the community. For instance, while on centralized platforms the provider decides who is allowed to offer complements (Wessel et al. 2017), moderate content (Zeng and Kaye 2022), sets transaction fees (Wang and Wright 2017), or modifies the platform's underlying technology and infrastructure (Ondrus et al. 2015), blockchain platforms either remove these decisions or devolve them to the community.[3] Archetypal blockchain platforms are typically free from censorship and thus neither limit access to the platform nor moderate content offered on the platform. As blockchain platforms explicate all platform rules in a collectively maintained protocol, changes to these rules require an ex-ante community consensus. Similar to blockchain platforms, established decentralized platforms, like open-source software (OSS) or Wikipedia, also devolve some of these governance decisions to the community. However, as the pertinent literature has shown, they still rely on some form of central authority (e.g., a core developer team or arbitrators in the case of Wikipedia) to settle disputes, moderate content, or steer development efforts (Puranam et al. 2014). Further, the platform provider often still owns the intellectual property underlying the technology and decides how to license it. Therefore, the platform provider can decide to change the technology unilaterally as long as all platform participants adopt it ex-post. Accordingly, although the community contributes most of the work on these platforms, the platform provider often maintains important governance rights.

Table 1 compares the different platform types regarding how the most important governance decisions are made.

——— insert Table 1 about here ———

[3]Although some centralized platforms also devolve some of these decisions to their users (see Boudreau 2010, Eisenmann and Parker, Goeffrey, Van Alstyne, Marshall 2009), it still is the platform provider's decision to what extent they include the community.

To contextualize our analysis, we first examine the foundational perspectives that have shaped the vision for decentralized platforms before turning to research on governance mechanisms and transaction fee dynamics.

2.1 Foundational Perspectives and the Vision for Decentralized Platforms

In addition to the contemporary strands of literature on platform governance and blockchain fee dynamics, a set of foundational works has shaped the way we understand the promise and challenges of decentralized platforms. Seminal contributions such as Nakamoto (2008) introduced the idea of Bitcoin as a peer-to-peer electronic cash system, laying the groundwork for a radical rethinking of centralized intermediaries. Building on these ideas, Catalini and Tucker (2018) and Vergne (2020) have argued that blockchain technology could enable a fairer distribution of value among network participants, challenging the traditional concentration of power in digital platforms.

Equally influential is Wood's (2014a) articulation of the Web 3.0 vision—a decentralized internet where transparency, inclusivity, and the democratization of control replace the conventional, centrally managed ecosystems. This vision has not only inspired technological innovation but has also framed the normative debate over how digital economies should be structured.

However, the early promise of disintermediation has been met with practical challenges. Research by Pereira et al. (2019) draws attention to the coordination and storage costs that arise when data is replicated across a decentralized network, highlighting an inherent trade-off between decentralization and operational efficiency. Similarly, Staub et al. (2022) emphasize that the removal of centralized governance tools—tools that have traditionally been used to curate and sustain diverse ecosystems—can create significant challenges for maintaining a healthy platform complement environment. In line with these concerns, Halaburda et al. (2020) offer a rigorous microeconomic analysis that underscores how decentralized protocols, while promising democratized value distribution, inevitably confront significant coordination and governance challenges in practice.

While later studies (e.g., Buterin 2014, Wu et al. 2021, Roughgarden 2020, Park 2023) delve into the empirical and design challenges of Ethereum's market-based fee mechanism and its susceptibility to front-running or other inefficiencies, they do so against a backdrop set by these earlier visionary narratives. This interplay between lofty aspirations and the economic realities of market-based transaction allocation provides a rich context for our investigation.

By revisiting these foundational perspectives alongside recent empirical evidence, this paper aims to bridge the gap between blockchain's original promise of a democratized digital infrastructure and the emerging evidence on its practical limitations. In doing so, we contribute not only to the literature on platform governance and fee dynamics but also to a broader understanding of how the initial ideals of blockchain technology confront—and sometimes conflict with—real-world market outcomes.

2.2 Research on platform governance and ecosystem orchestration

Although the vision for decentralized platforms emphasizes openness and transparency, research on platform governance reveals the structural challenges that arise when centralized coordination is replaced by market-driven mechanisms. The traditional platform governance literature highlights how centralized platforms actively manage their ecosystems through a set of governance tools, including pricing, curation, and selective incentives to foster innovation and maintain diversity. Traditional platforms do not rely solely on market forces; instead, they use strategic price discrimination and participation incentives to sustain a balanced complement ecosystem (e.g., Church and Gandal 1992, Shapiro and Varian 2010, Brynjolfsson and Kemerer 1996, Katz and Shapiro 1985, Farrell and Saloner 1986, Choi 1994).[4]

[4]For an extensive overview, see (Rietveld and Schilling 2020)

One key mechanism is selective pricing to attract and retain high-value participants. Google's search ad auctions do not allocate placements based purely on the highest bid but instead factor in quality scores preventing spam-like content from dominating search results, balancing short-term revenue generation with long-term ecosystem health (Varian 2007, Edelman et al. 2007).

In both digital and physical marketplaces, platforms use targeted pricing strategies to shape their ecosystems and encourage participation from key players. Visa and Mastercard, for instance, employ variable interchange fees to attract high-value merchants, such as supermarkets, by offering them lower fees, recognizing their role in driving high transaction volumes (Rochet and Tirole 2003, Prager et al. 2009). Similarly, shopping malls subsidize key tenants by providing preferential rents to anchor stores, ensuring their presence increases overall foot traffic and benefits smaller retailers (Caillaud and Julien 2003, Rysman 2009).

At a broader level, research emphasizes that platforms employ governance mechanisms to carefully orchestrate their ecosystems, shaping the composition and incentives of their participants. Rosaia's (2024) spatial equilibrium model and Marra's (2024) structural auction analysis both underscore how fee mechanisms and complementor heterogeneity shape platform ecosystems, introducing inefficiencies that centralized governance might mitigate. These studies reinforce the idea that governance mechanisms are essential for sustaining a diverse and innovative platform ecosystem, ensuring long-term sustainability and complement diversity (e.g., Cennamo and Santaló 2019, Tiwana 2015, Tudón 2022, Boudreau 2012, Casadesus-Masanell and Hałaburda 2014, Parker and van Alstyne 2018). Building on these insights, Gutierrez (2021) highlights how platforms such as Amazon Prime enhance welfare by fostering a diverse complementor ecosystem, while Brynjolfsson et al. (2003) demonstrate that consumer surplus from variety is a critical component of platform value creation. Similarly, platforms like Uber balance immediate profits with long-term ecosystem health by prioritizing variety and network diversity, even at the cost of short-term revenue (Sullivan 2022, Castillo 2023).

Our research adds to this stream by using blockchain platforms as a natural experiment to examine how replacing centralized governance with decentralized market mechanisms impacts ecosystem composition and performance. Decentralized platforms, by design, lack governance tools such as differential pricing, quality control, and targeted subsidies, exposing participants to externalities like fee-driven competition, congestion costs, and short-term profit extraction. Without these governance mechanisms, Ethereum's one-size-fits-all pricing model fails to differentiate between high value applications and extractive activities, disproportionately disadvantaging fee-sensitive but socially valuable applications while allowing price-insensitive, often extractive behaviors, such as maximal extractable value (MEV) transactions, to thrive. This dynamic challenges the assumption that network effects alone will sustain a robust and diverse complement ecosystem, reinforcing the need for governance mechanisms tailored to decentralized contexts.

2.3 Research on transaction fees on blockchain platforms

Beyond governance challenges, transaction fees play a fundamental role in how decentralized platforms allocate resources and shape participation. The foundational vision of blockchain as a decentralized marketplace (as discussed in Section 2.1) is operationalized through these fee structures, which determine transaction costs and influence platform behavior. Scholars have characterized blockchains as marketplaces for transaction validation services, where miners offer computational resources and users compete for blockspace.

For instance, Basu et al. (2019) and Easley et al. (2019) use game-theoretic models to show how Bitcoin's transaction fees fluctuate due to competitive bidding, potentially discouraging both miners and users. Huberman et al. (2021) show that Bitcoin's transaction fee mechanism protects users from monopoly pricing. While these studies focus primarily on Bitcoin's validation process and miner incentives, recent models have expanded this analysis to consider how fee structures impact transaction pricing dynamics, including base fees, parallel execution, and differences in fee policies across blockchain networks (Ndiaye 2024b,a, 2025).

Ilk et al. (2021) build on earlier theoretical work by providing empirical evidence on Bitcoin's transaction fee mechanism, showing that due to a relatively inelastic demand curve and a comparatively elastic supply curve, Bitcoin's fee structure can self-regulate. However, since their study focuses on Bitcoin, it only examines miners and transaction fee stability. In contrast, smart contract platforms like Ethereum require an assessment of how fee mechanisms impact platform participants such as dApp providers.

Empirical research on Ethereum's fee dynamics remains limited. Some studies examine network congestion and gas prices (Donmez and Karaivanov 2021) or gas price effects on throughput (Azevedo Sousa et al. 2021, Spain et al.), while others highlight how high fees contradict Ethereum's goal of financial inclusion by disproportionately excluding low-income users (Cong et al. 2022).

While research in this area focuses on how blockchain fee mechanisms affect validators and users, these studies primarily treat transaction pricing as a market efficiency problem, overlooking its broader impact on platform complementors and ecosystem diversity—a key focus of our research. Notably, there is limited work on how Ethereum's fee structure shapes dApp usage across different categories and influences overall platform composition. Our research addresses this gap by focusing on dApp providers as a critical component of blockchain platforms, analyzing how transaction fees influence their participation and the overall composition of these systems. Understanding their role is crucial, as they enable blockchain platforms to support a more diverse range of services, positioning them as potential alternatives to centralized platforms like Apple's iOS or Google's Android.

3 Background on Ethereum

Ethereum is the second-largest blockchain platform, with a market capitalization of 236 billion USD and over 1.2 million daily transactions.[5] It is the context of our study as it

[5]https://etherscan.io/ (retrieved on March 21rd, 2025).

was the first blockchain platform to introduce smart contracts, which enable more complex transactions than simple money transfers and thus allow complementors to develop their own dApps—smart contract-based apps running on top of the blockchain (Buterin 2014). As transactions differ depending on the complexity and thus require differing computational efforts to be executed by miners, Ethereum introduced a new market mechanism that incentivizes miners to compute transactions independently of their computational intensity. This market mechanism served as a blueprint for many other blockchain platforms that enable smart contracts and thus is seminal for the whole industry. In the following, we briefly review the core features of Ethereum's market for transactions and particularly focus on the economic aspect relevant to our paper. For a more technical review, we refer to Antonopoulos and Wood (2019) and Wood (2014b).

3.1 Smart contracts and dApps

Smart contracts are immutable and automatically enforced computer programs running on top of a blockchain (Fröwis and Böhme 2017). They allow developers to specify arbitrary agreements between two parties in the form of predefined obligations and rules written in computer code. If triggered by receiving a transaction, a smart contract is automatically enforced by the decentralized network according to the predefined rules, making it impossible for parties to unilaterally alter or renegotiate the transaction's outcome with a smart contract (Halaburda et al. 2019).

As smart contracts enable arbitrary programs, they can be used to develop so-called decentralized applications (dApps) (Wu et al. 2021). DApps are blockchain-based apps that resemble normal web applications regarding their user interface but differ from normal web applications as they run their business logic as a smart contract on a decentralized blockchain platform. Due to the immutability and automated enforcement of the underlying smart contract, users of a dApp do not have to trust the dApp provider or rely on third-party institutions to fulfill its obligations. Instead, they can read the smart contract and ascertain that the promised outcome will be delivered.[6] Therefore, the promise of dApps is to solve problems of centralized control, limited access, downtime, censorship resistance, and trust issues arising from weak institutions (Leiponen et al. 2021). For example, in the case of a collectibles game, the ownership of the collectible is not managed by the game provider but by a smart contract. Therefore, the provider cannot duplicate collectibles or change the ownership unilaterally and even in the event that the provider's servers are shut down, the owner of the collectible will not lose access to it.

DApps are the complements of interest for our study as they extend the functionality of the Ethereum network. Without dApps, Ethereum users could use the network only to send Ether (i.e., Ethereum's native cryptocurrency) to each other. With dApps, complementors can offer any arbitrary service. Currently, Ethereum hosts more than 4,900 dApps across categories such as finance, games, gambling, insurance, social media, property, and digital identity.[7] It is Ethereum's vision to grow further the number and diversity of dApps offered on the platform and ultimately pave the way for Web3.0, a more inclusive and democratic version of the Internet, where apps are available to everyone without any downtime, censorship, entry restrictions, and central control of the data.[8]

3.2 Ethereum's market for transactions

To validate, enforce, and record transactions users send to dApps, Ethereum uses a decentralized transaction mechanism that relies on cryptography, a decentralized consensus mechanism, and economic incentives to substitute a centralized intermediary. Prior scholars have already characterized Bitcoin mining, which uses a similar mechanism, as a two-sided market (e.g., Basu et al. 2019) and a market for data space more specifically Ilk et al. (2021).

[6] Obermeier and Henkel (2022) discusses that smart contracts only remove the necessity of trust if the users have read and completely understood its source code. In practice, due to the time and effort it takes to read a smart contract, this is rather unlikely. Still, they also argue that smart contract enables a new form of trust that is based on the possibility of reading the source code. This form of trust differs from trust in the dApp provider as it is based on logically provable facts (i.e., what is written in the source code) rather than on inference about latent characteristics of the dApp provider.

[7] https://dappradar.com/, last checked 03/21/2025

[8] https://ethereum.org/en/upgrades/vision/

We also characterize Ethereum's transaction validation and execution process as a market but highlight some important differences due to Ethereum's capability to run smart contracts and offer dApps.

Like on the Bitcoin network, transactions on Ethereum are not instantly effective but have to be validated by special users called miners. At regular intervals, these miners select transactions from the pool of pending transactions, verify their validity according to rules specified in Ethereum's protocol, bundle the transactions together, and participate in a computationally demanding puzzle known as "proof-of-work" (PoW). Only the winners of this puzzle get to write their block onto the blockchain and receive the block reward in addition to all transaction fees paid by the transaction senders. It is important to note that the mining of transactions comprises two tasks. First, the miner needs to solve the proof of work puzzle by computing numerous hashes (i.e., a string of character that results from transforming data a fixed-length string) until one miner finds a block hash that fulfills the requirements for a new block. Second, the miner needs to compute the transaction and check it against a list of rules. Only if the transaction fulfills these rules the miner can add it to the block. If even one transaction in a block would not fulfill the requirements, the whole block would be rejected by other miners. Both tasks require computational effort. Although the update from PoW to Proof-of-Stake (PoS; i.e., an alternative consensus mechanism that does not require to solve a computationally expensive puzzle to decide who gets to write the next block but randomly assigns the privilege to write a new blocks to miners according to their stakes tokens) drastically decreased the computational efforts miners have to invest in finding a new block, it does not impact the effort miners have to invest to validate every transaction. In essence, the update to PoS will even increase the relative importance of the effort it requires to validate a transaction.

In contrast to Bitcoin and to facilitate dApps and arbitrary transactions, Ethereum does not charge a fee per transaction but a fee for the computational effort a transaction requires. A transaction's computational effort is measured in *units of gas* according to a list that indicates a fixed gas requirement for every atomic computation.

To preserve decentralization, Ethereum intentionally limits the supply of transactions by limiting the maximum gas available in each block (*block gas limit*). Limiting the supply is necessary to allow as many validators as possible to join and help maintain a decentralized and secure network. In addition to limiting the total gas a block can use, the Ethereum protocol also tries to keep the average time it takes to find a new block (*average block time*) within a 12 to 14 seconds interval (Wood 2014b). These two limitations imply that the total amount of available gas has an upper limit. Not limiting the supply of transactions would favor validators with the most powerful machines as they could increase the transaction throughput to a point exceeding the capacity of the less powerful machines, preventing them from contributing new blocks and fostering the network's re-centralization.[9] Further, it prevents the network from getting trapped in an infinite loop of transactions. To allocate the limited transaction supply, most blockchain platforms like Bitcoin and Ethereum rely on a market mechanism to determine the price for transacting on the platform (Buterin 2014, Nakamoto 2008). We conceptualize this market mechanism as a market for transactions or, more specifically, a market for the validation and execution service of transactions. The limited supply of transactions in combination with this market mechanism has led to skyrocketing transaction fees in the past as even if more validators join the network the total supply of transactions does not increase. This design has real-world implications; for instance, high transaction fees have contributed to cases where dApp providers, such as Dapper Labs (dapperlabs.com), the developer of the CryptoKitties, a highly popular collectibles

[9]Even with a limited supply of transactions, Ethereum has experienced a significant increase in hardware requirements for running a full node. In the early days of Ethereum, when the first decentralized applications (dApps) launched, a system with 8 GB of RAM, approximately 50 GB of SSD storage, and a moderately fast internet connection was sufficient to stay synchronized with the blockchain. Today, running a full node requires at least 16 GB of RAM, 2 TB of SSD storage, and a high-speed internet connection to validate transactions (https://geth.ethereum.org/docs/getting-started/hardware-requirements). It is important to note that these specifications apply solely to validating and storing transactions and do not account for the hardware requirements of mining under the Proof of Work (PoW) consensus mechanism. Consequently, these requirements have persisted following the transition to Proof of Stake (PoS). Without limiting the gas supply (which limits the transaction supply), even more powerful machines would be necessary, potentially excluding a greater number of validators.

game, exited Ethereum to launch alternative platforms like Flow.

The commodity sold on Ethereum's market for transactions is the gas required to validate a transaction.[10] Accordingly, users (transaction initiators) are the buyers, whereas miners are the sellers of this commodity. On the supply side, the supply of gas on each day is fixed due to the block gas limit and the limited average block time. Although miners can decide to what extent they use this limit, they cannot change it individually. Changing this limit requires successful voting by all miners and a protocol update. Also, suppose more miners join the network and participate in the race to solve the mining puzzle. In that case, the network will increase the mining difficulty (i.e., the number of hashes it takes on average to find a new block) to keep the average block time within the target window of 12 to 14 seconds and keep the supply of gas fixed.[11]

To incentivize miners to provide their computation service, they are rewarded with a mining reward for every block they find. This reward consists of a static block reward (at the time of writing, 2 Ether) for finding a new block plus the sum of all gas fees (usually measured in $GWei$; 1 Ether = 10^9 GWei) paid by all transactions t which a miner includes in this block.

On the demand side, users cast transactions to other externally owned accounts (i.e., simple Ether transfers to other users or wallets controlled by computers) or smart contracts. To initiate a transaction, users must indicate a *transaction gas limit* (i.e., the maximum amount of gas a miner is allowed to use to compute the transaction) and a *gas price*(e.g., the price the user is willing to pay for each unit of gas). If the gas limit is reached before the transaction is fully computed, the transaction will be aborted and not included in the block. Users only pay for the used gas if the computation is finished before reaching the limit. Also, only the actually used gas is considered for the block gas limit. Accordingly, the fees a user has to pay is the product of gas used and the gas price the users is willing to pay

[10] It is important to note that the transaction initiator only has to pay the gas fees for the computation of the transaction but not for the computational effort the miner has to invest solving the PoW puzzle that is required to find a new block.

[11] See Appendix A for the formula used to compute the mining difficulty.

for every unit of gas.

As the supply of gas is limited, transaction senders compete with other senders by choosing a gas price that is high enough that miners pick their transactions from the pool of pending transactions. Typically, miners engage in profit maximization (Basu et al. 2019). Hence, they sort transactions by the indicated gas price and requirement and fill up the block until its gas limit is reached. Especially in times of congestion, offering too low a gas price means that a transaction will not be picked up by any miner and ultimately be deleted from the pool of pending transactions. Although, in theory, it is possible for transaction initiators to observe the gas price bids by other initiators and adjust their bids in response, we follow Roughgarden (2020) and see this price mechanism as a first-price, sealed-bid auction. Our reasoning for this type of auction in threefold. First, even though the pool of pending transactions is openly available, the peer-to-peer nature of the pool implies that not every participant sees every transaction simultaneously. Thus, it is difficult for initiators to determine what transactions were available to the miner when they assembled the block. Second, although a block is found on average every 12-14 seconds, the exact timing of a block's discovery cannot be predicted. Therefore, initiators do not know when they need to be among the highest bidders. Third, some wallets already offer gas price suggestions that help to gauge a price that has a high likelihood of leading to the inclusion of the transaction in one of the next blocks. However, these tools are only backward-looking. They suggest a gas price by extrapolating the gas prices that have led to the inclusion of the transaction on one of the last blocks. If initiators want to ensure that their transaction is processed with certainty, they still need to exceed this suggestion and account for the possibility that other initiators will do so, too. This gas price mechanism has led to considerable fluctuations in the amount of gas used, and the price users have paid for a unit of gas. For illustration, Figure 1 depicts the daily gas usage on the left and the daily average gas price on the right.

——— insert Figure 1 about here ———

In the next section, we develop a conceptual framework that explains the intuition underlying our empirical analysis. As our study focuses on the implications of Ethereum's market for transactions on the heterogeneity of complements offered on the network, the framework mainly focuses on the implications of gas fees on the usage of dApps. For an analysis of how gas fees impact the user (i.e., transaction senders) and miners in the network, we refer to Cong et al. (2022) and Basu et al. (2019).

4 Conceptual framework

The driving force behind our framework is that the usage of a dApp—hence its success—on Ethereum depends on the usage of the platform, which in turn again depends on the usage of other dApps.[12] However, due to two countervailing forces, it is unclear if increasing the user base and dApp base benefits all dApp providers. On the one hand, entering dApps attract new users to the platform, which fosters the platform's adoption, and enlarges the number of possible users of the focal dApp. On the other hand, the limited supply of transactions in combination with the first-price auction that allocates this limited supply aggravates the direct competition among dApps by introducing a negative externality: new dApps and users increase demand and intensify the competition for the limited supply of gas. The increasing demand and competition lead to increasing congestion costs and higher gas prices. Because transaction initiators need to pay transaction fees to interact with every dApp, increasing gas prices lessen the overall utility and, thus, the usage of dApps. Accordingly, the relative magnitude of these countervailing effects will determine the effect of Ethereum's market for

[12] It is important to note that although our empirical analysis is—due to the selection of our instrumental variable—limited to a period when Ethereum relied on PoW as a consensus mechanism, our theoretical arguments also apply to the period when Ethereum updated to PoS. Our arguments also apply to the period after EIP1559 (Ethereum Improvement Proposal). Although EIP1559 introduced a more flexible block gas limit and introduced an upper limit to the amount fees users can pay miners to incentivize them to process their transaction fast, it neither changed the fact that the supply of gas is still fixed and that users can outbid others by paying higher fees. The update to PoS only removed the computationally expensive puzzle of finding a new block but did not change the fact that users still need to compensate miners for validating and enforcing their transactions by paying fees for the gas used by their transactions. In a similar vein, our arguments should also apply to other smart contract-enabling platforms that rely on an auction-based transaction validation comparable to the one discussed above (e.g., Aztec Network, Binance Smart Chain, Optimism, Polygon,).

transactions on the success of the platform complements.

Although the net impact of increasing gas prices as a response to more platform usage is theoretically undetermined—due to the countervailing forces described above—we can analyze which characteristics of a dApp expose it more to changes in the gas price. Understanding this is not only useful for the complementors' decision to enter such a market but also for the platform provider, as it might have important implications for the heterogeneity of complements offered on the platforms. We hypothesize that depending on four characteristics, dApps are more or less sensitive to changes in the gas price and, therefore, better or worse equipped to compete in a market for transactions.

First, we expect that the type of service a dApp offers influences its sensitivity towards changes in the gas price. This intuition becomes clear when considering that some dApps provide social and entertainment services while others provide financial or security-related services. Although finance dApps do not necessarily provide more utility to users than leisure-related dApps, it is easier to compute the expected utility of a finance transaction. Therefore, it should be easier for users to evaluate if they still want to send a transaction whereas for other dApps the uncertainty and cognitive effort to gauge the expected utility will deter them from sending a transaction. Further, finance-related transactions are often more time-sensitive, and as Donmez and Karaivanov (2021) show, users on Ethereum are more willing to pay higher gas fees for timely transactions. Another reason why types of services might differ regarding their gas price elasticity of demand might be the frequency of required interactions. For instance, property and identity-related dApps typically require only infrequent interaction, whereas gaming or finance dApps require regular interactions. Through frequent interactions, gas fees can quickly accumulate and deter usage.

Second, even within the same type of service, dApps can substantially differ regarding the requirements of the transaction. For example, dApps can differ in the complexity of the underlying transaction and hence the gas required for the computation of it. On the one hand, the gas requirement correlates with the complexity of the underlying functionality. On the other hand, it is also driven by the efficiency of the code itself. Particularly within the same type of service, where the functionality and complexity of transactions with dApps is similar, the code's efficiency should be the main determining factor for the gas requirement. Especially in times of high gas prices, we expect users to be more sensitive to such differences and use dApps that require less gas for the same functionality. Another factor determining a dApp's gas price sensitivity should be the value transferred in a transaction with a dApp. For example, finance dApps carry value to transfer money to other accounts or to invest it (e.g., provide money to a liquidity pool). Other dApps require users to pay for their services (e.g., getting external data from an off-chain data sources called oracle) or to purchase goods (e.g., buying NFTs). Considering that some NFTs are sold for well above $100,000,[13] it becomes evident that even gas fees of a few dollars are negligible. Therefore, we expect that depending on the average transaction value that a dApp usually carries, the dApp should be more or less sensitive to changes in the gas price.

Third, dApps also differ in the overall quality of their services or their usability and hence in the value they create for their users. Accordingly, some dApps are more appealing to users than others. These dApps should not only perform better at baseline but are also more likely to benefit from the entry of other dApps. Consider, for example, that numerous new dApps enter Ethereum. This should attract additional users since users appreciate product variety. But once the users join, they will disproportionately choose the dApp offering more utility. This effect can be exacerbated if the dApp itself benefits from network effects, which should be the case for dApps such as currency exchanges, marketplaces, or social messengers. For such dApps, the increasing utility due to the larger network could counterbalance the additional fees resulting from the intensified competition for gas among dApp users.

Fourth, the current performance of a dApp should influence users' willingness to pay for a transaction with the dApp. Again, especially for dApps that rely on network effects, the

[13]For example, see CryptoPunk which are sold for as much as 8,000 Ether:
https://opensea.io/collection/cryptopunks

number of other users of a dApp should increase the value of transacting with this dApp.

To understand how Ethereum's market for transactions influences dApp usage, we next empirically investigate the drivers of dApps' transaction fee sensitivity as hypothesized above.

5 Data and sample construction

5.1 Research context and data

We combine daily block and transaction-level data publicly stored on the Ethereum blockchain with three different data sources that provide supplementary off-chain data, such as the category of the dApp or the exchange rate for one Ether or other tokens. Below we explain the data sources and the resulting sample and then discuss the variables in our data set.

5.2 Data collection procedure and sample

We obtained our data from four different sources. First, we use the Ethereum ETL[14] to download all block-level and transaction-level data publicly stored on the Ethereum blockchain for our study period (July 1st, 2017, until December 31st, 2020).[15] The block-level data include a unique identifier (i.e., block hash), a timestamp, the difficulty of the block, the gas limit, which indicates the maximum of gas miners are allowed to use in this block, and the gas used, which is the sum of computational effort the validation of all transactions in this block required. The transaction-level data contain the block hash, a sender and recipient address, the gas used by this transaction, and the gas price the sender has paid for one unit of gas in GWei (1 GWei = 10^{-9} Ether). Second, we use two websites that provide a curated list of dApps (stateofthedapps.com and defillama.com) to identify dApps that are running on Ethereum, the addresses of their associated smart contracts, and the category of the

[14] https://ethereum-etl.readthedocs.io/en/latest/

[15] We chose this study period as it allows us to observe three periods where the additional difficulty induced by the difficulty bomb caused a shortage in gas supply (see Figure 2 and Ethereum Improvement Proposal (EIP) 649, 1234, and 2384).

application. This step allows us to map the pseudonymous smart contract addresses on the blockchain to their respective dApp and is necessary because a dApp can rely on multiple smart contracts. Overall, we identified 1,590 dApps with 4,680 associated smart contracts active in our study period. As neither stateofthedapp.com nor defilama.com provides an exhaustive list of all smart contracts associated with a dApp, we further collect a list of all verified smart contracts from the Etherscan API[16] and manually match 1,316 additional smart contracts to the dApps in our sample. Through the address of the smart contracts, we can link transactions with their associated dApps. We also use the Etherscan API to collect further daily network-level data, such as network utilization, which measures the extent to which the block gas limit has been used. Finally, we retrieve the daily prices for one Ether and other tokens associated with the dApps in our sample from the CoinGecko API.[17] To ensure that all variables are on the same level and to mitigate high-frequency variation in the data, we first merge the block-level and transaction-level data by using the block hash reported for every transaction and then aggregate the resulting data at the daily level. Our consolidated dataset covers 1,279 days. Table 1 provides an overview of the number of dApps per group of categories. We obtained the groups by a cluster analysis based on variables describing the dApps' usage pattern (e.g., daily transaction count, transaction value, average gas requirement) .

——— insert Table 2 about here ———

5.3 Data sets, variables, and measurement

Besides the daily aggregation, we further aggregate transactions on the level of a dApp.

Our main variable of interest is the quantity of gas used ($gasUsed_t$). It refers to the daily amount of computational validation effort demanded by all transactions with a dApp. It is measured in Giga gas units. This variable operationalizes the goods supplied by the miners

[16]https://etherscan.io/apis
[17]https://www.coingecko.com/en/api/

and demanded by the transaction senders.

The gas price is the price (in GWei) transaction initiators must pay for each gas unit. As the gas price an initiator pays varies according to the outcome of a first-price auction, we define the gas price in times of the $marketGasPrice_t$ a sender would have had to pay for their transaction to just make it into one of the blocks on a given day. We proxy this market price with the daily average of the bottom fifth percentile gas price recorded on each block on that day in GWei. We use this proxy because there are some blocks where miners circumvent the market mechanism and add their own transactions at a gas price close to zero or even zero. Accordingly, using the minimum gas price (i.e., the lowest gas price on a day at which a transaction is just included in a block) would not correctly reflect the market mechanism. We also run several robustness checks with alternative gas price variables (e.g., different percentiles of the gas price in USD).

We define the variable difficulty bomb ($difficultyBomb_t$) as the average additional difficulty induced by Ethereum's difficulty bomb on a given day. Next to the automated adjustment of the mining difficulty, the difficulty bomb is the second mechanism encoded in Ethereum's protocol that influences the total network difficulty (i.e., the average number of hashes it takes to find a block). The goal of the difficulty bomb is to force miners to switch from PoW to PoS once the PoS update is available. To this end, the difficulty bomb exponentially increases the mining difficulty until it is almost impossible to find new blocks by solving the PoW puzzle. As Ethereum planned right from its start to switch to PoS at some point, the difficulty bomb was always part of the protocol. However, because the update to PoS was delayed several times, the difficulty bomb increased the difficulty too fast, resulting in a disproportionate increase that was not reflected by the network hash rate and the discovery of significantly fewer blocks per day. Because the resulting shortage in gas was not intentional (the plan was that PoS-blocks would grow at the same rate as the PoS-blocks would decline), the Ethereum community issued a protocol update that turned back the additional difficulty. Over our study period, this pattern occurred three times and is reflected in three protocol updates (EIP649, EIP1234, and EIP2384). As the difficulty induced by the difficulty bomb is not reported in any database, we leverage the fact that Ethereum's protocol continuously tried to keep the block time within the target window of 12-14 seconds and constructed the variable as follows. The difficulty induced by the difficulty bomb on a day d is the difference between the total observed difficulty and the theoretical difficulty required to reach the target block time, given the current hash rate in the network. Accordingly, the difficulty bomb on a day d is:

$$difficultybomb_d = difficulty_{observed,d} - (networkhashrate_d \times targetblocktime)$$

The unit of this variable is the number of Tera hashes it requires on average to find a new block. Due to the exponential growth and the fluctuation of the network difficulty within the target window, especially at the beginning of the activity of the difficulty bomb, the added difficulty is not always distinguishable from zero. To account for this fact, although the difficulty bomb is always active, we only assign a positive value to the difficulty bomb if the block time is noticeably above the target window (> 14s). According to this conservative approach, we only observe on 16% (182 days) of all days in our sample a difficulty bomb above zero. To establish robustness, we also use different cutoffs and approaches to measure the activity of the difficulty bomb. We will discuss our instrument's relevance and exogeneity later in the empirical strategy and results section. Figure 2 overlays the network hash rate with the observed total mining difficulty. Gaps between both curves indicate excessive difficulty added by the difficulty bomb.

——— insert Figure 2 about here ———

To account for the degree to which miners fill the blocks on a given day, we measure the network utilization (*networkUtilization*) as the fraction of total available gas (sum of the gas limit of all blocks) on a day that is used by all transactions on that day in percent. It captures the platform's usage level and has been used by prior researchers as a measurement for congestion (Donmez and Karaivanov 2021).

In addition to these variables, we compute several measures that allow us to study the transaction requirements of each dApp or their usage patterns. To reflect the complexity of an interaction with a dApp, we measure the average gas requirement (*avgGasRequirement*) of a transaction with a dApp. To reflect the requirements of a transaction with a dApp, we measure the average value of Ether (*avgValue*) or (*avgTokens*) a dApp receives as a proxy for how much value transactions with the dApp usually carry. In addition, we measure the following performance indicators for every dApp: average daily transaction activity (*avgDailyTxn*), average number of unique externally owned accounts (*avgDailyEOA*) that transactions with a dApp (i.e., our proxy for users),[18] the average gas price users pay for a transaction with a dApp (*avgGasPricePaid*), the average number of transactions per externally owned account on a given day (*avgTxnPerEOA*), and the surplus gas price the transaction senders paid beyond the market gas price on a given day *surplusGasPrice*.

We also control for the following network-level variables: Ether price (*EtherPrice*) is the exchange rate of one Ether in USD on the day the transaction was executed; Ether volatility *EtherVolatility* measures the daily change in the exchange rate of one Ether; gas limit *gas Limit* measures the sum of all block gas limits on a day and accounts for the fact that over our sample period, the total units of gas that can be used in a block has been increased several times; and finally day of the week (*weekday*) and year (*year2017-2020*) dummy variables, and a trend (*trend*).

Based on this data, we created two data sets. The first data set is aggregated on the network level and has one time series for all transactions on Ethereum (including Ether transfers between wallets), one for all dApp transaction in our sample, and one for every group. This data set allows us to estimate a demand curve for each group of dApps in our sample and compare it to the demand curve of all transactions. The second data set

[18] Technically it is possible to differentiate between smart contract addresses and wallet addresses, but not if a wallet address is controlled by a bot. To account for this fact, we refrain from calling wallet addresses "users" and call them instead "externally owned accounts" to emphasize that they do not necessarily correspond to human users. Therefore, this variable is only a proxy.

is a panel data set on the dApp-day level. It only comprises transaction to dApps in our sample. It allows us to control for dApp level fixed effects and to conduct further moderation analyses.

Table 3 provides descriptive statistics and correlation scores for all variables in our network-level data set. Table 11 in Appendix 9 show descriptive statistics and correlations for our dApp-level data set.

—— insert Table 3 about here ——

6 Estimation strategy

In this section, we discuss our baseline specification and the instrumental variable (IV) we use to address the endogeneity of the gas price.

6.1 Baseline specification

The specification for our dApp-level analysis is:

$$\begin{aligned}\log(gasUsed_t) = \alpha_0 + \alpha_1 \log(marketGasPrice_t) + \alpha_2 networkUtilization_t + \\ \alpha_3 networkUtilization_t^2 + \alpha_4 \log(EtherPrice_t) + \alpha_5 \log(EtherVolatility_t) + \\ \alpha_6 \log(gasLimit_t) + \mu_{day\ of\ week} + \mu_{year} + trend + u_t\end{aligned}$$

where gas used is the equilibrium gas demand aggregated over all executed transactions on the network or per group of dApps in the period t (day), $\mu_{\text{dayofweek}}$ denotes the day of week effects, μ_{year} the year effects, and u_{t} is the error term. We chose a log-log specification for gas used and market gas price to be able to interpret α_1 as the price elasticity of the demand. Due to the skewed distributions of Ether price, Ether volatility, and the gas limit, we use log-transformed versions of these variables in our specification. In addition, we also control for the level of network utilization. This allows us to control for the degree to which miners use the available block gas limit on a given day and has been used by prior scholars as a measure of network congestion (Donmez and Karaivanov 2021). We also add a quadratic term to account for the nonlinear relationship between gas price and network utilization.[19] Our specification for the dApp-level data set resembles the equation above but adds a dApp-level fixed effect (see Appendix 9)

6.2 Validity of the instrument

In this model, $\log(gasUsed_t)$ and $\log(marketGasPrice_t)$ are the endogenous variables, as both are jointly determined in equilibrium. To address this simultaneity issue, we use the *difficultyBomb* as an instrumental variable in a two-stage least squares approach (2SLS). In the first stage, we use the difficulty and all other control variables listed above to predict the $\log(marketGasPrice_t)$. In the second stage, we estimate the specification above by replacing the $\log(marketGasPrice_t)$ with its predicted value. The economic intuition underlying our approach is that we leverage the difficulty bomb as an exogenous supply shifter. Due to the consistent adjustment of the network difficulty and the resulting constant block time, the gas supply curve resembles a fixed vertical line. When the difficulty bomb is active, the added difficulty increases the block time and thus decreases the number of blocks on a given day. As the maximum gas a block can contain is limited, fewer blocks lead to a decrease in the gas supply and hence a horizontal shift of the supply curve to the left. We exploit this supply shift to identify the demand curve.

We argue that the difficulty bomb is exogenous and influences the gas demand only through the increased gas price for three reasons. First, it is programmed into the Ethereum protocol, and changing it requires a successful protocol update (called Ethereum Improvement Proposal or EIP) which is only possible after a majority vote and hence unlikely to be a response to a short-term market situation. Therefore, the difficulty bomb and its resets

[19] We also compute the same model with a threshold specification where we added only the linear term and dummy variable that takes on the value one if the utilization level exceeds 90%. The were qualitatively the same regarding the magnitude and significance of the coefficients we obtained.

bomb can be seen as exogenous policy interventions. Second, as the difficulty level is not reported in wallet applications or by an API and has to be manually calculated (see Section 5.3), it is plausible to assume that ordinary Ethereum users were not aware of the existence of the difficulty bomb. Third, even if users were aware of the existence of the difficulty bomb, it is difficult for them to comprehend its exponential growth and differentiate its impact—at least in the initial phase—from normal fluctuations due to the exit and entry of miners. Further, it would also be difficult for users to predict the mining power and cost structure of every single miner and to evaluate when they cannot keep up with the difficulty level.

7 Results

In this section, we report and discuss four sets of results. First, we report the results of our baseline estimation and our finding of a downward-sloping demand curve. Second, we report our results regarding different gas price elasticities for each group of dApps. Third, we present our analysis regarding further characteristics of a dApp that determine its sensitivity towards changes in the gas price. Finally, we discuss the additional checks we conduct to establish the robustness of our results.

7.1 Baseline Network-level results

Following the network-level specification, Table 4 reports the results of our 2SLS demand curve estimation. Column 1 presents the first stage results, where we predict the gas price (*log(Market gas price)*) with our IV (*difficulty bomb*). Column 2 presents the second stage results, where we use the predicted gas price to estimate the price elasticity of the gas demand (*log(Gas used)*). Finally, column 3 provides an OLS model for comparison.

—– insert Table 4 about here —–

Consistent with our prediction, Columns 2 and 3 suggest a downwards-sloping demand curve for gas on Ethereum. The first stage reported in Column 1 shows that an increase in additional difficulty due to the difficulty bomb is significantly associated with increased gas prices. This is in line with our explanation that the added difficulty reduces the supplied gas—by reducing the number of blocks explored per day—and thus intensifies price competition among transaction senders. The coefficient of the difficulty bomb is highly significant even though we control for network utilization (i.e., the degree to which miners use the available block space), network utilization squared,[20] the exchange rate of Ether to USD, the daily fluctuation of this exchange rate, the block gas limit, as well as day of the week and year dummies and a common trend.

Regarding the validity of our instrument, by comparing the first-stage with and without the instrument, we obtain an incremental F (305.20) that is well beyond the suggested cut-off of 10 (Stock and Yogo 2005) and thus suggests that our instrument strongly correlates with the endogenous gas price. To test the relevance of our instruments, we compute the Stock-Yogo test for weak instruments, which shows that the Cragg-Donald-Wald F Statistic (85.6) exceeds the predetermined critical value (16.38). Further, we compute the Kleibergen-Paap LM Statistic (4.18) for under-identification, which is highly significant. These tests suggest that our instrument is both strong and relevant. Regarding the exogeneity of our instrument, we have already explained above that the difficulty bomb does not impact the gas demand through means other than an increase in gas price as the mining difficulty simply is a "production factor" for miners that is unlikely to be tracked by the casual Ethereum user.

When comparing the 2SLS estimate with the OLS results, we find that although it is still negative and significant, the OLS estimmate is smaller in size (2SLS: -0.69 vs -0.04). To understand this underestimation of the true effect of the gas price on the demand, consider a positive but unobserved shock in demand. This shock shifts the demand curve upwards.

[20]The inclusion of the quadratic term is suggested by a scatterplot that shows a highly nonlinear relationship between the network utilization and the gas price. Especially, when the network utilization exceeds 90% the gas price increases dramatically. We also performed a robustness check using a threshold effect at 90% network utilization in form of a binary variable that is equal to 1 if the utilization is above 90% and 0 otherwise which we then interact with the linear term. This finding is similar to Donmez and Karaivanov (2021) who test the impact of congestion on the gas price for a shorter observation period.

Given the fixed supply of gas this demand shift leads to a higher intersection of the demand curve with the supply curve and an increase in the equilibrium price. As a result, the unobserved error and the gas price are positively correlated which leads to a downward bias when not controlling for the endogeneity of the gas price. Accordingly, without our instrument the true effect of the gas price on the demand for gas would be underestimated.

To interpret the magnitude of the effect of the gas price ($\log(marketGasPrice_t)$) on the demand of gas log(Gas used), the coefficient of -0.69 implies that a 1% increase in the market price of a unit of gas decreases the amount of gas demanded by 0.69%. Considering that the average transaction on Ethereum consumes 184,000 units of gas (which corresponds to a normal smart contract interaction), this equals a decrease of approximately 1,750 smart contract transactions per day or 15,000 Ether transfers which require 21,000 units of gas. Considering that the median dApp only receives eight transactions per day, the order of magnitude of this effect can have significant economic implications.

In sum, this analysis provides first empirical evidence that the well-established "law of demand" (Gale 1955) also applies to the validation service of transactions on Ethereum. It also provides evidence that Ethereum's gas price mechanism introduces a form of price competition among transaction senders that counteract the main prediction of the two-sided market literature (Rochet and Tirole 2006), i.e., that, due to the same-side network effect, an increase in the demand side draws even more consumers into the market and leads to subsequent increases in demand. On Ethereum, an increase in transaction senders increases not only the utility of transacting on Ethereum but also price competition. However, as the demand for gas is negatively associated with its price, the market mechanism underlying Ethereum's transaction validation process dampens the effectiveness of same-side network effects.

7.2 Differing demand curves per group

In addition to estimating a demand curve for all transactions on Ethereum, we also estimate a specific demand curve for every group of dApps along with their confidence intervals. Table 5 reports the second stage result of this estimation. Each of these models uses the aggregated daily gas used by all dApps within the respective group as the dependent variable. Columns 2-6 depict that the coefficients of log(*Market gas price*) significantly vary between the groups of dApps and thus signal that the groups differ regarding their sensitivity to changes in the gas price.

—– insert Table 5 about here —–

To compare the different gas price elasticities, we also compute their 95 percent confidence intervals. Figure 3 depicts these intervals and shows that not all elasticities can be distinguished with enough confidence, but some significant differences are still noticeable. Especially games and marketplaces (group 3) seem to be far more sensitive to changes in gas prices than dApps in group 1 and group 2. Considering that group 3 mainly comprises collectible games, such as crypto kitties, where the timing of the transaction does not matter as much as, for example, finance or cryptocurrency exchange dApps, where the timing often matters due to swift changes in prices of cryptocurrencies, this result seems plausible. Further, the one-time nature and relatively high transaction values in group 2 (identity and property dApps) can explain why users are relatively insensitive to changes in the gas price.

—– insert Figure 3 about here —–

7.3 Heterogeneous effect of Ethereum gas price mechanism

Beyond the category of a dApp, we use our rich data to explore further the characteristics of dApps that impact their sensitivity toward the gas price. For this analysis, we use our dApp-level data set. Appendix 9 shows the baseline results for this data set. The first set of characteristics pertains to the formal requirements of a transaction with a dApp. These characteristics are the amount of gas a transaction with a dApp requires and the value of Ether and tokens a transaction with a dApp usually carries. To analyze these characteristics, we computed the total average for all these variables over all transactions a dApp has received. Because this average is time-invariant, we interacted these variables with the gas price and group in different models: In Table 6, Columns 1 and 4 show the two-way and three-way interaction models regarding the average gas requirement; Columns 2 and 5 show the interaction models with the average Ether value sent; and Columns 3 and 6 the models with the average token value sent.

—— insert Table 6 about here ——

Regarding the gas requirement of a transaction with a dApp, we do not find a significant two-way interaction effect between the gas price and the average gas requirement (Column 1), but we find significant three-way interactions between gas price, gas requirement and group two, three, and four (Column 4). These interactions indicate that for some groups of dApps, the two-way interaction significantly differs from the reference category (group 1). For instance, for gambling dApps, the negative coefficient of the three-way interaction (-0.24) implies that the negative impact of the gas price on the gas demand is even stronger if the gambling dApp demands a high amount of gas for a transaction. On the other hand, for dApps in group 2, the coefficient of the three-way interaction is positive (0.58). This implies that, in comparison to the dApps in group 1, for identity and property dApps, a high gas requirement counteracts the downward slope of the demand curve to some extent, leading to a decrease in the sensitivity towards changes in the gas price. One possible explanation for this finding could be the required frequency of interaction with a dApp. In contrast to gambling and finance applications, where users obtain utility from regularly interacting with dApps, identity and property dApps only require sporadic transactions. If a property dApp bundles more functionality into one transaction, not only the gas requirement but also the utility of the transaction increase. Accordingly, the user might be willing to accept high gas prices for this transaction as the additional gas fees become less relevant in relation to the one-time transaction effort. For gambling and finance applications, however, users generate utility through more frequent interactions. Here, more functionality in a single transaction might increase the utility but, in the long-run, also pile up more transaction fees. Thus, users might be less inclined to higher gas requirements as they prefer less complex but dedicated functions realized through singular transactions. Another explanation could be that due to the frequent interaction gambling dApps require, there is more pressure for such dApps to improve the efficiency of their smart contracts in terms of gas requirement.

Regarding the average value (in Ether or other tokens) sent with a transaction to a dApp, we find a positive moderation of the negative demand curve (Columns 2 and 3). The positive interaction coefficients between the log(*marketGasPrice* and the log(*avgValue*) (0.14) and log(*avgTokenValue*) (0.31), in combination with the negative linear coefficient of the gas price (-0.64 and -0.74) are an indicator that the gas price elasticity of dApps decreases with a higher average transaction value. This finding is in line with prior studies that find users' fee sensitivity declines with the transaction value (e.g., Wang and Wright 2017).

Regarding the three-way interactions (log(*marketGasPrice*) × log(*avgValue*) × *group2-5* and log(*marketGasPrice*) × log(*avgTokenValue*) × *group2-5*), we only find that one out of the eight coefficients is significant. This indicates that, apart from group 5, the positive and significant interaction of the transaction value with the gas price does not differ across the groups of dApps and suggests that dApps that receive a higher average transaction value exhibit a less elastic demand curve.

Next to the requirements of a transaction with a dApp, we also computed average performance indicators for each dApp. Table 7 reports the interaction result regarding the average daily number of transactions, the average daily number of externally owned accounts (EOA), and the average daily transactions per EOA.

— insert Table 7 about here —

For the average daily transactions and average daily EOA, we find a positive and significant two-way interaction with the gas price. This suggests that the demand for gas for transactions with dApps with a high average of daily transactions and users is less impacted by changes in the gas price. However, by adding the group dummies to these two-way interactions, we find that this interaction significantly differs between dApps in group one and all other groups. Whereas dApps in group 1 still seem to benefit from more transactions and EOAs—as indicated by the positive and significant two-way interactions between the gas price and the average number of transactions (Column 4, 0.39) and the average number of daily EOA (Column 5, 0.39)—the three-way interactions with all other groups are highly significant and negative. This indicates that for dApps in these groups, the effect of receiving, on average, more transactions or having more unique EOAs transacting with them is less prevalent or even makes them more sensitive to changes in the gas price. Again, network effects could be a plausible explanation for this observation. Particularly, finance dApps and cryptocurrency exchange dApps should highly benefit from network effects. A gas price increase caused by an influx of additional users could be compensated by the additional utility the growing number of users provides to finance and exchange dApps. Simultaneously, because dApps from other groups benefit less from network effects, they cannot compensate for the additional gas fees their users would have to pay to transact with them. Especially, for dApps that already have a high average number of users but fail to benefit from network effects, this effect can lead to an increase in the sensitivity towards the gas price and a decline in demand for transactions with these dApps—especially in times when there is less supply of gas and fierce price competition. For the average number of transactions per EOA (Columns 3 and 6), we only obtain a few significant results that do not allow us to infer systematic patterns.

—– insert Table 8 about here —–

To further investigate network effects, we analyze the impact of dynamic usage indicators that vary for each dApp over time. Table 8 reports the interaction results of the daily ratio of transactions per EOA and the average price users were willing to pay above market gas price. Regarding the number of transactions per EOA, we find a positive interaction (0.08, Column 2) between the number of transactions per EOA and the gas price (*log(Market gas price)*). According to the three-way interactions, except for group 5, this moderation does not significantly differ between the different groups of dApps. Because for dApps in group five, the interaction is even stronger than for all other dApps, attracting heavy users might be a valid strategy for these dApps to survive the competition in a market for transactions. Considering that group 5 comprises dApps such as storage or energy services and given the strong lock-in effects these services typically exhibit, also these findings seem plausible.

Finally, regarding the average surplus gas price transaction senders are willing to pay on a given day for transacting with a dApp, we also observe a positive interaction with the gas price (0.16, Column 5). Again, except for group 5, this moderation approximately remains its direction and magnitude across the different groups. Only for group 5, the three-way interaction has a negative sign. This implies that, in comparison to dApps in group 1, dApps in group 5 are more sensitive to changes in the gas price in periods where their users overpay the market gas price. Such periods could be periods with high fluctuations in the gas price that expose users to high uncertainty regarding the gas price and forces them to overpay for a certain inclusion of their transaction. Therefore, a possible explanation for the negative three-way interaction could be that users of dApps in this group are more sensitive to this form of uncertainty related to overpaying and thus react by becoming more price sensitive.

7.4 Additional robustness checks

To assess the robustness of our analysis, we tested them against several alternative measures and samples. For example, we used the transaction count instead of gas used, applied different levels of winsorization to restrict the impact of possible outliers, used different percentile and levels of winsorization for the market gas price together with the average gas price, and also a different measurement of the difficulty bomb where we subtracted the observed number of blocks from the target number of blocks given the targeted block time. Further, we also conducted our analysis only for the periods where the difficulty bomb was active. Table 13 reports the coefficients we obtain through the robustness tests. Overall, we find the results to be consistent with the results of our baseline specification.

——— insert Table 9 about here ———

Moreover, we further report two additional analyses that corroborate our results in the appendix. The first analysis replicates parts of our analysis on the network level. For this analysis, we aggregated all transactions on the network and group level instead of the dApp level. Rather than using the group as an interaction term, this allows us to estimate a dedicated demand curve for each group of dApps. The results we obtain are qualitatively the same, except that we do not observe an upwards-sloping demand curve for group 1 (finance dApps) but a slightly downwards-sloping demand curve. The second analysis is a survival analysis that shows that dApps from different groups are subject to different hazard rates.

8 Discussion

8.1 Interpretation of results

Blockchain technology allows substitution a centalized platform intermediary with a decentralized market mechanism and thus has induced a paradigm shift in how we think of platform designs. Although there is a burgeoning stream of theoretical (Easley et al. 2019, Basu et al. 2019, e.g.,) and empirical (Ilk et al. 2021, Donmez and Karaivanov 2021, Cong et al. 2022, e.g.,) research that has started investigating the consequences of replacing a platform intermediary with a market mechanism for the validators and users, there is a lack of research focusing on the consequences of such a mechanism for the complements offered on blockchain platforms. Addressing this gap is important for three reasons. First, it provides an interesting opportunity study to what extent the invisible hand of the market can orchestrate an appealing ecosystem of platform complements. Second, it allows studying a new source of negative network effect induced by the competition of complements for the same resource. Finally, as we know from the platform literature (Rietveld and Schilling 2020) how crucial complements are for the success of a platform, addressing this gap also helps us understanding if and how blockchain platforms will be able to compete with their centralized counter parts. To address this gap, our goal was to test our hypothesis that a market mechanism that forces all sorts of complements to compete for the same resources (i.e., the verification of transactions) and allocates these resources only based on the users' willingness to pay might be efficient in the short run but lead to undesirable long term consequences.

Our empirical analysis has provides three main insights into this hypothesis. First, using a novel instrumental variable (i.e., Ethereum's difficulty bomb), we address the simultaneity issues of demand and supply and estimate a demand curve for transactions on a blockchain platform that offers third-party complements. Although the downward-sloping demand curve we find aligns with basic economic theory and might seem trivial, this finding is significant because scholars previously questioned the applicability of economic theory to transactions on blockchain platforms due to the prevalence of malicious and erroneous transaction behavior (Donmez and Karaivanov 2021). Our results confirm that economic theory can indeed be applied to transactions on blockchain platforms, providing a necessary foundation for further analysis of blockchain fee markets. Furthermore, this result is crucial as it reveals the magnitude of the effect of changes in the gas price on transaction demand. According to our estimation, a 1% increase in the gas market price reduces demand by 1,703 smart contract transactions per day or 14,923 Ether transfers. Our instrumental variable also indicates that without addressing the endogeneity issue surrounding the gas price, we would underestimate the price sensitivity of demand and the detrimental effect of increasing gas prices. Given that the majority of dApps have fewer than a few hundred daily transactions, this finding emphasizes that even small price increases can significantly reduce transaction volumes for many dApps. Consequently, stabilizing gas price fluctuations should be a top priority for platform providers.

Second, as predicted by our hypothesis, we confirm that different types of dApps exhibit varying sensitivities to changes in the gas price. While this finding may not be surprising, the magnitude of the effect is noteworthy. On average, users on Ethereum demonstrate relatively inelastic demand ($\alpha_{1,\text{ all dApps}}$ =-0.45) However, the elasticities of demand for different types of dApps vary significantly. For instance, with a coefficient of -2.09, the demand for transactions with gaming dApps is highly elastic. A 1% increase in the gas market price is expected to decrease daily gas demand by 17.2 million units of gas or 172 transactions, assuming a simple transaction with a gaming dApp requires 100,000 units of gas. Considering there are, on average, 18,985 daily transactions with gaming dApps during our sample period, a 10% increase in the daily gas market price (e.g., from an average of 14.11 to 15.52 GWEI) will result in a 9% decrease in transactions with gaming dApps, potentially causing many gaming dApps to receive no transactions at all. Although our estimates indicate less elastic demand curves for groups 2, 4, and 5, Figure 5 empirically shows that transactions to dApps in these groups are also crowded out by finance dApp transactions when gas prices rise. This analysis reveals that, contrary to the claims of Ethereum's market fee mechanism proponents, dApps with higher sensitivity do not merely experience longer wait times for transaction processing. Instead, sustained periods of high gas fees imply that their transactions will never be validated and ultimately dropped from the pool of pending transactions.

——— insert Figure 5 about here ———

Third, according to our moderation analysis, dApp providers have almost no strategic tools to influence their price sensitivity and mitigate being crowded out by finance transactions. One valid attempt would be to optimize the dApp's smart contract gas requirements. However, depending on the smart contract's intended functionality, the optimization potential might be limited [21] and also available to finance dApps and hence will not help to counteract their users' comparatively lower willingness to pay. Another approach revealed by our analysis is to increase the transaction value, as transactions with higher value are less price sensitive. While this approach might be feasible for identity and property dApps (group 2) and could explain why they are less sensitive to changes in the market gas price, other dApps, such as gaming (group 3) or social messenger dApps (group 4), require transactions with little to no transaction value. Finally, dApps could also try to create network effects. However, building artificial network effects is more difficult for most applications compared to finance dApps, particularly DeFi dApps, which naturally benefit from network effects. Given this limited toolset to counteract the implications of Ethereum's market mechanism and the limited strategic tools available to platform providers to protect disadvantaged complements, especially applications with a high sensitivity to changes in the gas price, might not be viable on such platforms in the long run.

8.2 Unintended consequences of Ethereum's market mechanism

Our empirical results provoke an important discussion about efficiency versus fairness on blockchain platforms that use a market mechanism to allocate the limited supply of transactions and ensure decentralization. On the one hand, allocating transaction supply to the parties with the highest willingness to pay is efficient, as it leads to a Pareto-optimal allocation and maximizes returns for validators. Additionally, it provides an objective basis for the optimal allocation decision, which can be automatically enforced by a decentralized protocol. On the other hand, assessing the fairness of a purely market-based transaction supply allocation is more challenging, as it depends on the platform's goals and definition of equitable benefits and burdens. Blockchain platforms typically follow a utilitarian approach and claim decentralization is "fair" because they replace a centralized, profit-maximizing platform provider with a market mechanism. However, from a more socially focused per-

[21] https://www.vibraniumaudits.com/post/gas-optimization-in-ethereum-smart-contracts-10-best-practices

spective on fairness, one can argue that a market mechanism is not fair in terms of social equity, as it does not ensure a basic level of service to all parties. For instance, Cong et al. (2022) show that despite claims that blockchain helps to bank the unbanked, platforms like Ethereum exclude poorer individuals from transacting on the platform.

Our goal is not to resolve this debate. Instead, we aim to highlight the potentially unintended consequences of a purely market-based transaction allocation mechanism and to shed light on the mismatches between the economic reality of blockchain platforms using such mechanisms and the Web 3.0 rhetoric they employ to advertise the value they create. The first unintended consequence is a long-run loss in complement heterogeneity. As some types of dApps are more sensitive to increasing gas fees than others, rising gas prices—driven by the entry of less price-sensitive dApps—imply that more price-sensitive dApps will receive fewer transactions. Our analysis shows that even small to moderate gas price increases can cause some dApps to stop receiving transactions and leave the market. This mechanism is problematic because factors other than the quality of the dApp (e.g., the type of application or transaction value) determine its sensitivity to changes in the gas price. Finance applications, in particular, exhibit lower gas price sensitivity and may crowd out other types of applications. Figure 6 illustrates this dynamic. It depicts the number of active finance (red line) and non-finance (blue line) dApps together with the gas market price in GWEI.number[0] We can see that in 2020, the gas price significantly increased as more finance dApps entered the market. At the same time, the high transaction fees prevented non-finance dApps from receiving transactions, forcing them to exit the platform. This loss of complement diversity is problematic because platform users value the diversity of complements offered on a platform (Rietveld and Schilling 2020). Furthermore, it contradicts the Web 3.0 goal of enabling a wide range of applications on decentralized platforms.

———— insert Figure 6 about here ————

The second unintended consequence is closely related to the first and is a loss of the

[0] A supplementary survival analysis in Appendix9 confirms this crowding-out effect.

platform's experimentation and innovation capability. As the market mechanism prioritizes current willingness to pay over the future potential of a dApp, high gas fees prevent new and small dApps from entering the platform, as they will not receive any transactions. Without transaction activity, these dApps cannot validate their product-market fit, especially when their product depends on network effects. Consequently, a market mechanism that allocates the supply of transactions might prevent promising new applications from growing and reaching a critical mass of users that would justify paying higher transaction fees, even if the dApp would be beneficial for the platform in the long run. Figure 7 provides evidence for this argument. It splits the gas price history depicted in Figure 6 into three regimes and plots the number of dApps over their age when they exit the platform. In the first two periods, there is a lot of experimentation, with many young dApps entering and leaving the market. In the third period, when gas prices are high, this experimentation vanishes. This loss of experimentation capability is especially problematic as blockchain technology is still searching for its "killer application" that will bring the technology into the mainstream. It is questionable whether we would have seen innovations like NFTs (non-fungible tokens) if they had been introduced after the DeFi hype. Furthermore, with the current price regime, it is unlikely that we will see promising new applications outside of DeFi. In terms of Web 3.0 rhetoric, the fact that this mechanism helps incumbent dApps protect their market share and concentrates transaction traffic on a few powerful dApps contradicts the idea of a decentralized market.

——— insert Figure 7 about here ———

The third and most concerning consequence is a phenomenon called *miner–* or *maximal extractable value* (MEV). MEV refers to the value miners or validators can extract directly from smart contracts due to their control over the ordering of transactions (Daian et al. 2020). As pending transactions are typically observable in the pool of pending transactions, parties with access to these pools can exploit the market mechanism and front-run transactions by paying higher fees to get their transaction executed before the targeted transaction. Since the market mechanism prioritizes transactions based solely on the gas price the sender is willing to pay, it explicitly enables this type of value extracting transactions. MEV transactions do not enhance the overall welfare. They do not create value but rather extract value created by others, clearly contradicting most notions of fairness. There is ample research attempting to measure the extent of MEV activity (Qin et al. 2021, Park et al. 2024, Daian et al. 2020). Although it is difficult to measure the exact extent of MEV transactions, evidence suggests that these transactions constitute a considerable share of all transactions on Ethereum. [22] MEV parties, often trading bots, compete with other parties for the MEV, engaging in bid wars. These bid wars further inflate gas prices and crowd out normal transactions.

MEV activity is particularly prevalent in decentralized finance (DeFi) applications, especially automated market makers (AMMs), which are among the most popular decentralized finance platforms due to their ability to facilitate swift transactions that dominate limit order markets under certain conditions (Capponi and Jia 2021, Lehar and Parlour 2021, Hasbrouck et al. 2022). However, conceptual flaws in their design make AMMs especially susceptible to sandwich attacks—a specific MEV strategy that combines front-running with at least one back-running transaction (Park 2023). These attacks significantly exacerbate network congestion by tripling the number of transactions required for a single operation. Therefore, beyond exploiting individual users, our empirical findings reveal that this increase in high-paying transactions imposes broader negative externalities on all platform participants as all participants compete for the same supply of gas by offering higher fees. Ironically, the very mechanism that creates this externality is what facilitates this form of MEV.

To explore the demand dynamics of MEV transactions, we use a sample of 5.5 million MEV transactions provided by Park et al. (2024) and estimate a demand curve for these transactions. As Table 10 suggests, the demand curve for MEV transactions is upward sloping, providing further evidence for bid wars and the gas price inflating effect of MEV transactions.

[22] https://studio.glassnode.com/metrics?a=ETHm=transactions.TxTypesBreakdownRelative

———— insert Table 10 about here ————

8.3 Potential solutions and future research

There are three potential ways to mitigate the problem of high transaction fees and their consequences on blockchain platforms. Each method carries its own trade-offs, and further research is necessary to evaluate their costs and benefits.

The first approach is to rely on the invisible hand of the market and wait for it to self-correct by creating new platforms that cater to the needs of various non-finance dApps. However, this approach might be problematic because permissionless platforms do not control platform entry. Consequently, they cannot prevent finance dApps from entering the platform and restarting the cycle unless the platform provider relinquishes full decentralization and assumes control over who is allowed to enter the platform.

The second approach is to directly address the problem of limited supply by scaling the platform and increasing its throughput. This method is currently most actively pursued by the Ethereum community.[23] To increase throughput, either the data to be processed needs to be reduced or compressed to fit more transactions into a block (e.g., Bitcoin's SegWit update), or the block size needs to be increased. As most platforms already strive to be as efficient in their data usage as possible, and further updates would require significant alterations to their protocols, changing the block size is often considered easier. However, increasing the block size creates a trade-off with decentralization. Validators with less powerful machines may not be able to stay synchronized with the blockchain, thus excluding them from participating in the consensus. Another approach to scaling throughput involves layer two (L2) scaling solutions. These solutions take transactions off the main blockchain, process them on a separate platform, and only post the result of the transaction back on the main blockchain(Cong et al. 2023). Often, these platforms rely on only one or a few validators. While this approach significantly increases throughput and reduces transaction

[23] https://ethereum.org/en/roadmap/scaling/

costs, it compromises decentralization. Therefore, none of the approaches in this category fully resolves the inherent problem that decentralization relies on creating redundancies, which leads to more capacity constraints compared to centralized platforms. These solutions merely postpone the issue. Future research is necessary to identify which types of transactions might benefit from different trade-offs between decentralization and scalability.

The third approach to addressing the problem of high transaction fees and their consequences is to accept the limited supply and focus on allocating it differently. This approach is currently underexplored and stands to benefit the most from future research. One potential solution within this framework is the use of subsidies. For example, a recent Ethereum update proposes account abstraction[24], which allows dApp providers or third parties to pay the fees instead of users. While this might help new dApps overcome the critical mass problem by using venture capital to subsidize early users, the costs will ultimately be passed on to the users in the long run. As this update is relatively new, we do not yet know how these dynamics will play out, and more research is required to investigate under what conditions this method of subsidizing transactions is a useful tool. Alternatively, instead of subsidizing transactions through account abstraction, dApps could run their own nodes and include transactions with their dApps at lower fees. However, this approach incurs opportunity costs, and its feasibility depends on the dApp's block production costs (e.g., electricity and hardware costs in PoW or cost of capital in PoS). Another alternative is to experiment with different transaction allocation mechanisms. Random allocation, charging fixed costs, or rationing come to mind as possible alternatives to allocating based on the highest willingness to pay. Random allocation or "first-come-first-served" could be alternative approaches that do not require active monitoring or interference. However, these are technically difficult to implement due to the pseudonymous nature of transactions and unlimited entry, which would allow more affluent parties to create multiple accounts and transactions. Charging fixed costs would mitigate bidding wars and price hikes but would

[24]https://ethereum.org/en/roadmap/account-abstraction/text

not allow users to express preferences regarding their transaction confirmation time. Moreover, fixed prices might become outdated and require updating, which is challenging as it necessitates knowledge about demand and supply. Rationing could be implemented by assigning a fixed amount of supply to different types of transactions. Ethereum's introduction of blob space and danksharding[25] can be seen as such an approach. Solana, another popular blockchain platform, also experiments with a form of rationing in the form of neighborhood fees. These fees increase only for similar dApps if their demand for transactions increases disproportionately compared to other types of dApps. While this approach might mitigate a crowding-out effect, it also limits the potential of "superstar" complements, which we know are crucial for the success of the entire platform(Rietveld and Schilling 2020). Furthermore, there is a small stream of literature proposing multidimensional blockchain fees(Diamandis et al. 2023, Angeris et al. 2024) that consider not only block space but also other dimensions such as bandwidth. Although this research shows that multidimensional fees can enhance welfare and improve network performance, more research is required to investigate if they also help preserve a diverse dApp ecosystem.

In summary, this discussion demonstrates that all solutions involve trade-offs. Therefore, decentralized platforms must carefully consider these trade-offs and ideally vote on them. Future research could support this process by developing various measures of economic and social welfare specific to decentralized platforms and evaluating different transaction allocation approaches based on these measures.

8.4 Limitations

This paper has some limitations that open opportunities for further research. One limitation is that we only observe one platform. Even though our analysis suggests that the gas price mechanism on Ethereum might lead complementors to leave the network and join other platforms, this paper abstains from addressing cross-platform competition and substitution

[25] https://ethereum.org/en/roadmap/danksharding/text

patterns. A natural extension of our work is to extend our analysis to other blockchain platforms that offer dApps, and study platform complements' switching and multi-homing behavior. One particularly interesting platform is Solana which relies on a different approach to prioritize transactions and thus might provide a promising case to investigate if their approach is able to strike a better balance between prioritizing high in demand dApp and protecting innovation on the edges. Another limitation is our sample of dApps and their associated smart contracts. Although we tried to include as many dApps as possible in our analysis and even manually matched smart contracts to these dApps, more dApps are running on Ethereum than our sample reflects. Particularly, dApps only accessible through Chinese or Russian websites might have slipped our attention and are not represented in our sample. Therefore, and although in some periods, our sample accounts for as much as 85% of all Ethereum transactions, our results should be seen as initial empirical evidence and would profit from replications that incorporate a different set of dApps or take a more fine-grained perspective on the rich available data. Particularly, zooming in on single days and following the bidding behavior of individual users or studying the usage pattern of a single dApp in light of changing gas prices could be promising. Finally, due to the infancy of and the rapid development in this field, our results should be treated as preliminary and could be reevaluated after major protocol updates. One such change was Ethereum's long-announced update from PoW to PoS. As this update only removed the computationally expensive puzzle of finding a hash that fulfills some properties required by the protocol but not the computation, validation, and recording of the transaction, the gas price mechanism should be even more important as now it is the most important driver of the costs of validating transactions. As we discussed above, we have good reasons to believe that the main mechanism behind our results is not affected by the switch to PoS. Nonetheless, it would be interesting to see empirical evidence on how validators prioritize transactions and influence the usage of dApps under PoS and on platforms that use similar transaction verification mechanisms.

9 Conclusions

Blockchain platforms aspire to create an equitable, decentralized economy by replacing a rent-extracting platform intermediary with a decentralized network of participants and an auction-based market mechanism for transaction allocation. We show that this approach introduces a tradeoff: while decentralization eliminates platform-level rent extraction, it creates an auction-based environment that disproportionately favors financially extractive activities — such as arbitrage and maximal extractable value (MEV) transactions — over applications that generate broader, long-term value.

MEV refers to a class of trading strategies in which market participants—such as miners, validators, or automated trading bots—strategically reorder, insert, or exclude transactions within a blockchain's validation process to extract profit. These strategies often rely on front-running and sandwich attacks (a combination of front- and back-running), which exploit the transparency of blockchain transaction queues. Because MEV extractors are willing to pay high transaction fees to secure priority, they can outbid other applications for transaction capacity.

On decentralized blockchains, transaction capacity is inherently limited to reduce the risk of centralization. Unlike centralized platforms that can scale computational resources as needed, most blockchain networks impose limits on throughput to avoid allowing only the most well-resourced parties to participate in transaction validation. These constraints mean that transactions must compete for scarce block space, with priority determined by users' willingness to pay fees. This process influences the allocation of transaction resources on the platform, shifting capacity toward applications that prioritize short-term revenue extraction over other uses.

Using Ethereum as a case study, we analyze transaction data from over 1,500 decentralized applications (dApps) to examine this dynamics. DApps are software applications that operate on a blockchain rather than being hosted on centralized servers. They are used for a range of purposes, including financial services, gaming, marketplaces, and social networking. To estimate how different categories of dApps respond to transaction fees, we leverage Ethereum's difficulty bomb, a protocol feature designed to gradually make adding blocks more computationally expensive over time. This feature was introduced as part of Ethereum's long-term plan to transition from Proof-of-Work (PoW)—where miners validate transactions by solving complex computational puzzles—to Proof-of-Stake (PoS), where transaction validation is based on capital commitments rather than computational effort. The difficulty bomb artificially slowed down block production, raising transaction fees. However, because the Ethereum's core developers arbitrarily reset the difficulty bomb multiple times through protocol updates, the resulting changes to transaction fees were not driven by market forces or user demand. This exogenous variation in fees allows us to estimate demand curves in different application categories.

Our findings show substantial differences in price sensitivity between dApp categories. When transaction fees increase, demand for transactions declines most sharply in applications that rely on frequent, low-cost interactions, such as gaming, social, and marketplace applications. In contrast, financial services applications, particularly those related to decentralized finance (DeFi)—which include arbitrage and MEV transactions mentioned earlier—exhibit lower price elasticity and maintain activity even as fees increase. The asymmetric effects of transaction fees create a feedback loop: as financial applications sustain activity in high-fee environments, they contribute to congestion, further limiting participation by more fee-sensitive applications.

Although our findings are derived from the Ethereum platform during its PoW era, the underlying mechanisms we document—capacity constraints and auction-based transaction allocation—are present across almost all decentralized blockchain platforms, including those using PoS. Unlike centralized platforms, which can curate applications, subsidize usage, or differentially price transactions to maintain ecosystem balance, decentralized platforms lack governance tools to correct congestion-driven exclusion that prioritizes short-term revenue extraction. As a result, there are no straightforward solutions to mitigating the distortions we identify. This raises concerns about whether decentralized platforms can fulfill their envisioned role as inclusive and diversity-enhancing general-purpose infrastructure for Web 3.0 and become a viable alternative to currently dominant centralized platforms.

However, our study provides insights into how alternative transaction allocation mechanisms could better balance efficiency with platform inclusivity while mitigating inefficiencies without relying on centralized coordination. Several approaches—layer-2 scaling solutions, differential pricing models, or alternative auction mechanisms—could help address fee-driven exclusion, but each involves tradeoffs. Scaling solutions may increase throughput but risk weakening decentralization, while rationing or subsidies require governance mechanisms that may not be feasible within existing blockchain structures. Further research on the economic design of transaction pricing and resource allocation in decentralized markets could inform both platform development and regulatory considerations.

References

Angeris G, Diamandis T, Moallemi C (2024) Multidimensional blockchain fees are (essentially) optimal. *arXiv preprint arXiv:2402.08661* .

Antonopoulos A, Wood G (2019) *Mastering Ethereum: Building smart contracts and DApps*. EBL-Schweitzer (Beijing, Boston, Farnham, Sebastopol, Tokyo: O'Reilly), 1st ed. edition, ISBN 9781491971918, URL `https://ebookcentral.proquest.com/lib/gbv/detail.action?docID=5594049`.

Azevedo Sousa J, Oliveira V, Valadares J, Dias Gonçalves G, Moraes Villela S, Soares Bernardino H, Borges Vieira A (2021) An analysis of the fees and pending time correlation in ethereum. *International Journal of Network Management* 31(3), ISSN 1055-7148, URL `http://dx.doi.org/10.1002/nem.2113`.

Basu S, Easley D, O'Hara M, Sirer E (2019) Towards a functional fee market for cryptocurrencies URL `http://arxiv.org/pdf/1901.06830v1`.

Boudreau K (2010) Open platform strategies and innovation: Granting access vs. devolving control. *Management Science* 56(10):1849–1872, ISSN 0025-1909, URL `http://dx.doi.org/10.1287/mnsc.1100.1215`.

Boudreau K (2012) Let a thousand flowers bloom? an early look at large numbers of software app developers and patterns of innovation. *Organization Science* 23(5):1409–1427, ISSN 10477039, URL `http://dx.doi.org/10.1287/orsc.1110.0678`.

Brynjolfsson E, Hu Y, Smith MD (2003) Consumer surplus in the digital economy: Estimating the value of increased product variety at online booksellers. *Management science* 49(11):1580–1596.

Brynjolfsson E, Kemerer C (1996) Network externalities in microcomputer software: An econometric analysis of the spreadsheet market. *Management Science* 42(12):1627–1647, ISSN 0025-1909, URL `http://dx.doi.org/10.1287/mnsc.42.12.1627`.

Buterin V (2014) Ethereum whitepaper. URL `https://ethereum.org/en/whitepaper/`.

Caillaud B, Julien B (2003) Competing in network industries: Divide and conquer. *RAND J. Econ* 34:309–328.

Capponi A, Jia R (2021) The adoption of blockchain-based decentralized exchanges. *arXiv preprint arXiv:2103.08842* .

Casadesus-Masanell R, Hałaburda H (2014) When does a platform create value by limiting choice? *Journal of Economics & Management Strategy* 23(2):259–293, ISSN 10586407, URL `http://dx.doi.org/10.1111/jems.12052`.

Castillo JC (2023) Who benefits from surge pricing? *Available at SSRN 3245533* .

Catalini C, Tucker C (2018) Antitrust and costless verification: An optimistic and a pessimistic view of the implications of blockchain technology. *SSRN Electronic Journal* 1–14, URL `http://dx.doi.org/10.2139/ssrn.3199453`.

Cennamo C, Santaló J (2019) Generativity tension and value creation in platform ecosystems. *Organization Science* 30(3):617–641, ISSN 10477039, URL `http://dx.doi.org/10.1287/orsc.2018.1270`.

Choi J (1994) Network externality, compatibility choice, and planned obsolescence. *The Journal of Industrial Economics* 42(2):167, URL `http://dx.doi.org/10.2307/2950488`.

Church J, Gandal N (1992) Network effects, software provision, and standardization. *The Journal of Industrial Economics* 40(1):85, URL `http://dx.doi.org/10.2307/2950628`.

Cong L, Tang K, Wang Y, Zhao X (2022) Inclusion and democratization through web3 and defi? initial evidence from the ethereum ecosystem. *SSRN Electronic Journal* URL `http://dx.doi.org/10.2139/ssrn.4229308`.

Cong LW, Hui X, Tucker C, Zhou L (2023) Scaling smart contracts via layer-2 technologies: Some empirical evidence. *Management Science* 69(12):7306–7316.

Daian P, Goldfeder S, Kell T, Li Y, Zhao X, Bentov I, Breidenbach L, Juels A (2020) Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability. *2020 IEEE symposium on security and privacy (SP)*, 910–927 (IEEE).

Diamandis T, Evans A, Chitra T, Angeris G (2023) Designing multidimensional blockchain fee markets. *5th Conference on Advances in Financial Technologies (AFT 2023)* (Schloss Dagstuhl-Leibniz-Zentrum für Informatik).

Donmez A, Karaivanov A (2021) Transaction fee economics in the ethereum blockchain. *Economic Inquiry* 60:265–292, URL `https://onlinelibrary.wiley.com/doi/pdfdirect/10.1111/ecin.13025?download=true`.

Easley D, O'Hara M, Basu S (2019) From mining to markets: The evolution of bitcoin transaction fees. *Journal of Financial Economics* 134(1):91–109, ISSN 0304405X, URL `http://dx.doi.org/10.1016/j.jfineco.2019.03.004`.

Economides N, Himmelberg C (1995) critical mass and network size with application to the us fax market. *Mimeo, Stern School of Business at New York University)* .

Edelman B, Ostrovsky M, Schwarz M (2007) Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *American economic review* 97(1):242–259.

Eisenmann T, Parker, Goeffrey, Van Alstyne, Marshall (2009) Opening platforms: How, when and why. *Platforms, Markets, and Innovations* (UK: Edward Elgar Publishing).

Farrell J, Saloner G (1986) Installed base and compatibility: Innovation, product preannouncements, and predation. *The American Economic Review* 76(5):940–955.

Fröwis M, Böhme R (2017) In code we trust? measuring the control flow immutability of all smart contracts deployed on ethereum. Garcia-Alfaro J, Navarro-Arribas G, Hartenstein H, Herrera-Joancomartí J, eds., *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, volume 10436 of *Lecture Notes in Computer Science*, 357–372 (Cham: Springer International Publishing), ISBN 978-3-319-67815-3.

Gale D (1955) The law of supply and demand, , vol. 3. *Mathematica Scandinavica* 3(1):155–169.

Gutierrez G (2021) The welfare consequences of regulating amazon. *Available at SSRN 3965566* .

Halaburda H, Haeringer G, Gans JS, Gandal N (2020) The microeconomics of cryptocurrencies. *National Bureau of Economic Research Working Paper Series* No. 27477, URL `http://dx.doi.org/10.3386/w27477`.

Halaburda H, Levina N, Min S (2019) Understanding smart contracts as a new option in transaction cost economics. *Proceedings of the 40th International Conference on Information Systems, Munich, Germany* .

Halaburda H, Sarvary M, Haeringer G (2022) *Beyond bitcoin* (Springer).

Hasbrouck J, Rivera TJ, Saleh F (2022) The need for fees at a dex: How increases in fees can increase dex trading volume. *Available at SSRN 4192925* .

Huberman G, Leshno J, Moallemi C (2021) Monopoly without a monopolist: An economic analysis of the bitcoin payment system. *The Review of Economic Studies* 88(6):3011–3304, URL `http://dx.doi.org/10.1093/restud/rdab014`.

Ilk N, Shang G, Fan S, Zhao J (2021) Stability of transaction fees in bitcoin: A supply and demand perspective. *MIS Quarterly* 45(2):563–692, ISSN 02767783, URL `http://dx.doi.org/10.25300/MISQ/2021/15718`.

Katz M, Shapiro C (1985) Network externalities, competition, and compatibility. *The American Economic Review* (75):424–440.

Lehar A, Parlour C (2021) Decentralized exchange: The uniswap automated market maker. *The Journal of Finance* .

Leiponen A, Thomas L, Wang Q (2021) The dapp economy: a new platform for distributed innovation? *Innovation* 1–19, ISSN 1447-9338, URL `http://dx.doi.org/10.1080/14479338.2021.1965887`.

Marra M (2024) Estimating an auction platform game with two-sided entry. *Available at SSRN 4879222* .

Nakamoto S (2008) Bitcoin: A peer-to-peer electronic cash system. URL `https://bitcoin.org/bitcoin.pdf`.

Ndiaye A (2024a) Blockchain price vs quantity controls. *Available at SSRN 4538155* .

Ndiaye A (2024b) Why bitcoin and ethereum differ in transaction costs: A theory of blockchain fee policies. *Available at SSRN 4991772* .

Ndiaye A (2025) Parallel execution fee mechanisms. *Working paper* (arXiv:2410.09555), URL `http://dx.doi.org/10.48550/arXiv.2410.09555`.

Obermeier D, Henkel J (2022) Smart contracts on a blockchain: Transaction governance with the potential of deductive certainty. *Working paper* .

Ondrus J, Gannamaneni A, Lyytinen K (2015) The impact of openness on the market potential of multi-sided platforms: A case study of mobile payment platforms. *Journal of Information Technology* (30):260–275.

Park A (2023) The conceptual flaws of decentralized automated market making. *Management Science* 69(11):6731–6751.

Park S, Jeong W, Lee Y, Son B, Jang H, Lee J (2024) Unraveling the mev enigma: Abi-free detection model using graph neural networks. *Future Generation Computer Systems* 153:70–83.

Parker G, van Alstyne M (2018) Innovation, openness, and platform control. *Management Science* 64(7):3015–3032, ISSN 0025-1909, URL `http://dx.doi.org/10.1287/mnsc.2017.2757`.

Pereira J, Tavalaei M, Ozalp H (2019) Blockchain-based platforms: Decentralized infrastructures and its boundary conditions. *Technological Forecasting and Social Change* 146:94–102, ISSN 00401625, URL `http://dx.doi.org/10.1016/j.techfore.2019.04.030`.

Prager RA, Manuszak MD, Kiser EK, Borzekowski R (2009) Interchange fees and payment card networks: Economics, industry developments, and policy issues .

Puranam P, Alexy O, Reitzig M (2014) What's new about new forms of organizing? *The Academy of Management Review* (39):162–180, ISSN 03637425.

Qin K, Zhou L, Gervais A (2021) *Quantifying Blockchain Extractable Value: How dark is the forest?*

Rietveld J, Schilling M (2020) Platform competition: A systematic and interdisciplinary review of the literature. *Journal of Management* (in press).

Rochet JC, Tirole J (2003) Platform competition in two-sided markets. *Journal of the European Economic Association* (1):990–1029.

Rochet JC, Tirole J (2006) Two-sided markets: a progress report. *The RAND journal of economics* 37(3):645–667.

Rosaia N (2024) Competing platforms and transport equilibrium. *Working Paper* .

Roughgarden T (2020) Transaction fee mechanism design for the ethereum blockchain: An economic analysis of eip-1559 URL `http://arxiv.org/pdf/2012.00854v1`.

Rysman M (2009) The economics of two-sided markets. *Journal of economic perspectives* 23(3):125–143.

Shapiro C, Varian H (2010) *Information rules: A strategic guide to the network economy* (Boston, MA: Harvard Business School Press), [nachdr.] edition, ISBN 087584863X.

Spain M, Foley S, Gramoli V (????) The impact of ethereum throughput and fees on transaction latency during icos. URL http://dx.doi.org/10.4230/OASICS.TOKENOMICS.2019.9.

Staub N, Haki K, Aier S, Winter R (2022) Governance mechanisms in digital platform ecosystems: Addressing the generativity-control tension. *Communications of the Association for Information Systems* 51:906–939, URL http://dx.doi.org/10.17705/1CAIS.05137.

Stock J, Yogo M (2005) Testing for weak instruments in linear iv regression. Stock JH, Andrews DWK, eds., *Identification and inference for econometric models*, 80–108 (New York: Cambridge university press), ISBN 9780521844413, URL http://dx.doi.org/10.1017/CBO9780511614491.006.

Sullivan M (2022) Price controls in a multi-sided market. Technical report, Working paper.

Tiwana A (2015) Evolutionary competition in platform ecosystems. *Information Systems Research* 26(2):266–281, ISSN 1047-7047, URL http://dx.doi.org/10.1287/isre.2015.0573.

Tudón J (2022) Prioritization vs. congestion on platforms: evidence from amazon's twitch.tv. *The RAND Journal of Economics* 53(2):328–355, ISSN 0741-6261, URL http://dx.doi.org/10.1111/1756-2171.12409.

Varian HR (2007) Position auctions. *international Journal of industrial Organization* 25(6):1163–1178.

Vergne JP (2020) Decentralized vs. distributed organization: Blockchain, machine learning and the future of the digital platform. *Organization Theory* 1(4):1–26, ISSN 2631-7877, URL http://dx.doi.org/10.1177/2631787720977052.

Wang Z, Wright J (2017) Ad valorem platform fees, indirect taxes, and efficient price discrimination. *The RAND Journal of Economics* 48(2):467–484, ISSN 0741-6261, URL http://dx.doi.org/10.1111/1756-2171.12183.

Wessel M, Thies F, Benlian A (2017) Opening the floodgates: The implications of increasing platform openness in crowdfunding. *Journal of Information Technology* 32(4):344–360, ISSN 0268-3962, URL http://dx.doi.org/10.1057/s41265-017-0040-z.

Wood G (2014a) apps: What web 3.0 looks like. URL http://gavwood.com/dappsweb3.html.

Wood G (2014b) *Ethereum: A secure decentralised generalised transaction ledger*. URL https://files.gitter.im/ethereum/yellowpaper/viyt/paper.pdf.

Wu K, Ma Y, Huang G, Liu X (2021) A first look at blockchain–based decentralized applications. *Software: Practice and Experience* 51(10):2033–2050, ISSN 0038-0644, URL http://dx.doi.org/10.1002/spe.2751.

Zeng J, Kaye DBV (2022) From content moderation to visibility moderation : A case study of platform governance on tiktok. *Policy & Internet* 14(1):79–95, ISSN 1944-2866, URL http://dx.doi.org/10.1002/poi3.287.

Tables and Figures

Figures

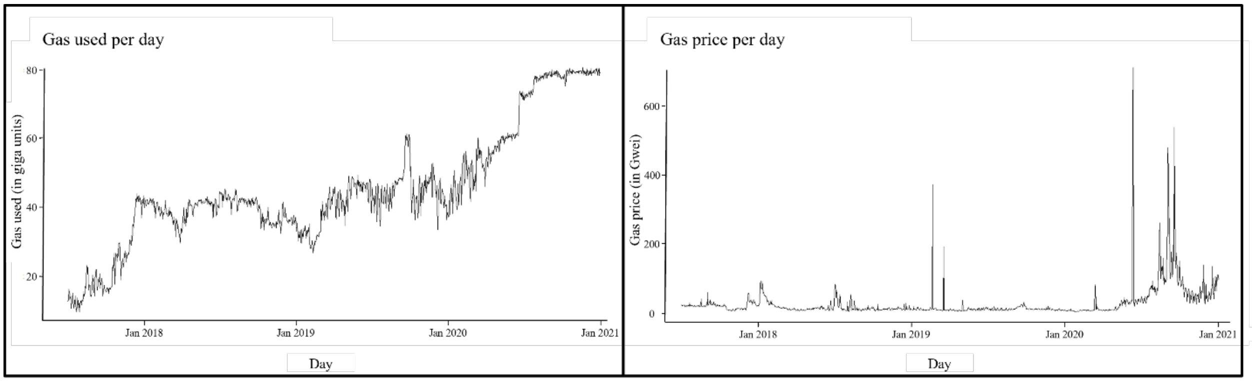

Figure 1: Daily gas used and gas price

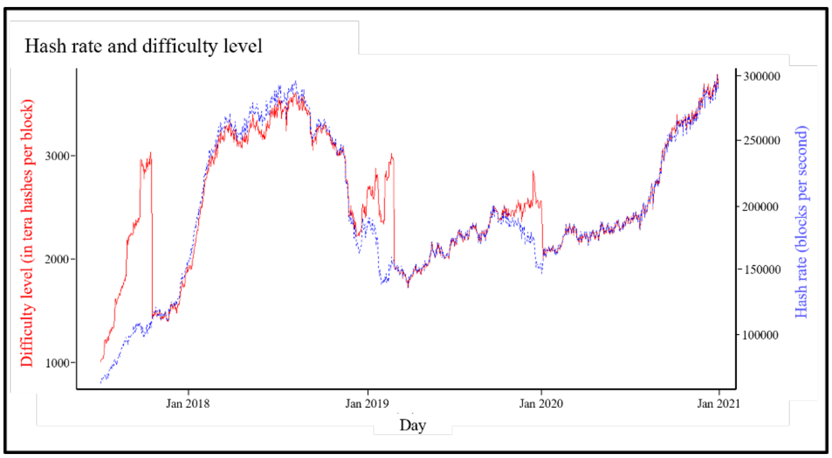

Figure 2: Hash rate and the impact of the difficulty bomb

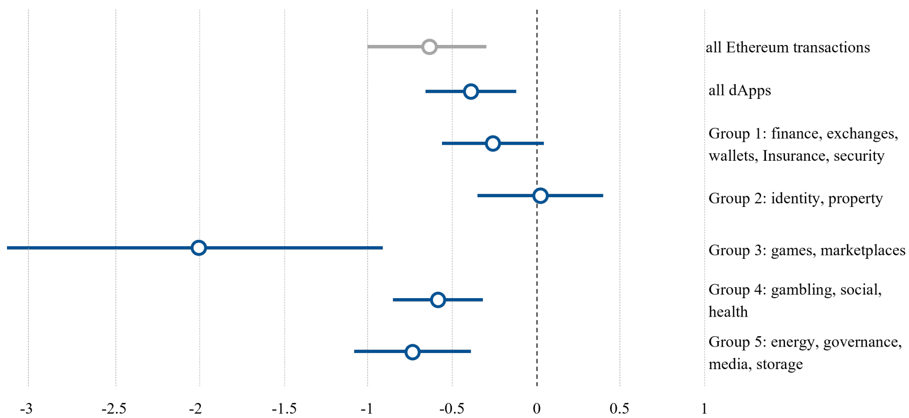

Figure 3: Price elasticities of demand per group of dApps

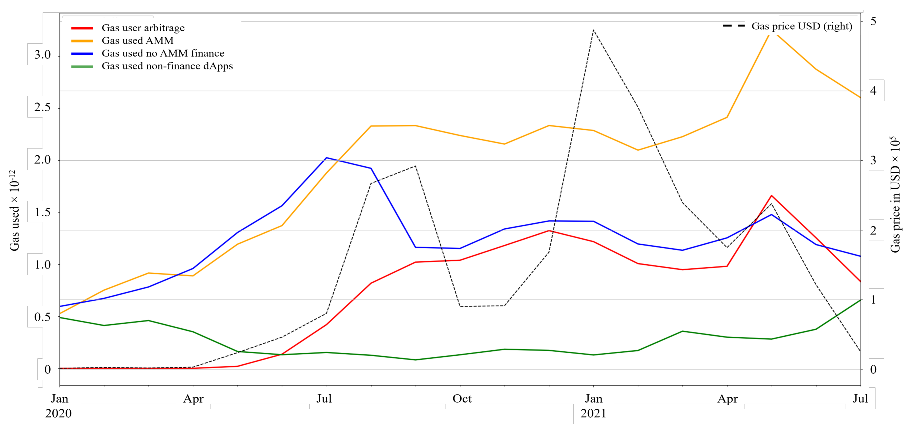

Figure 4: Gas used by MEV vs Defi vs. non-finance dApps

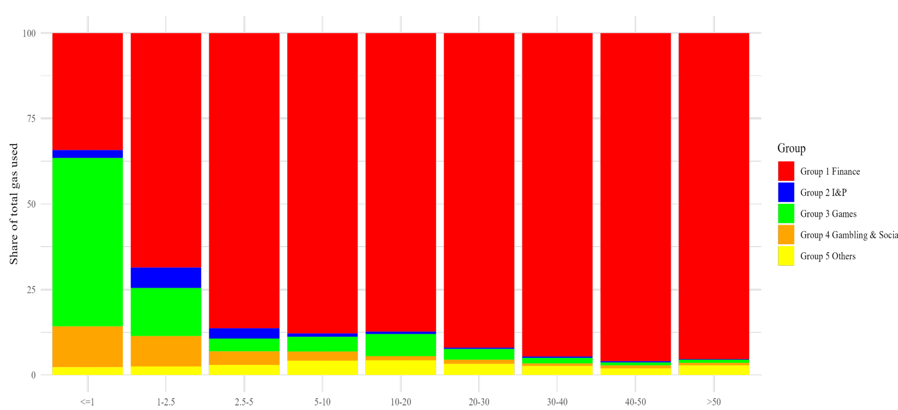

Figure 5: Gas used by gas price bucket and group

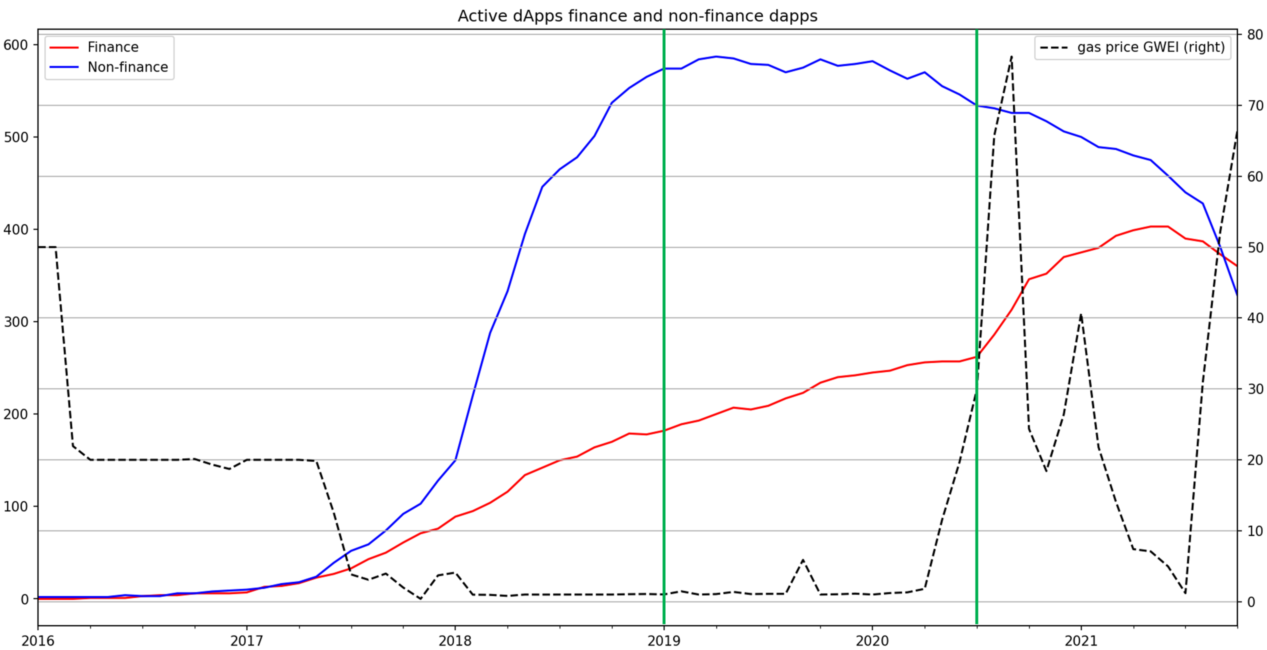

Figure 6: Active dApps finance vs. non-finance

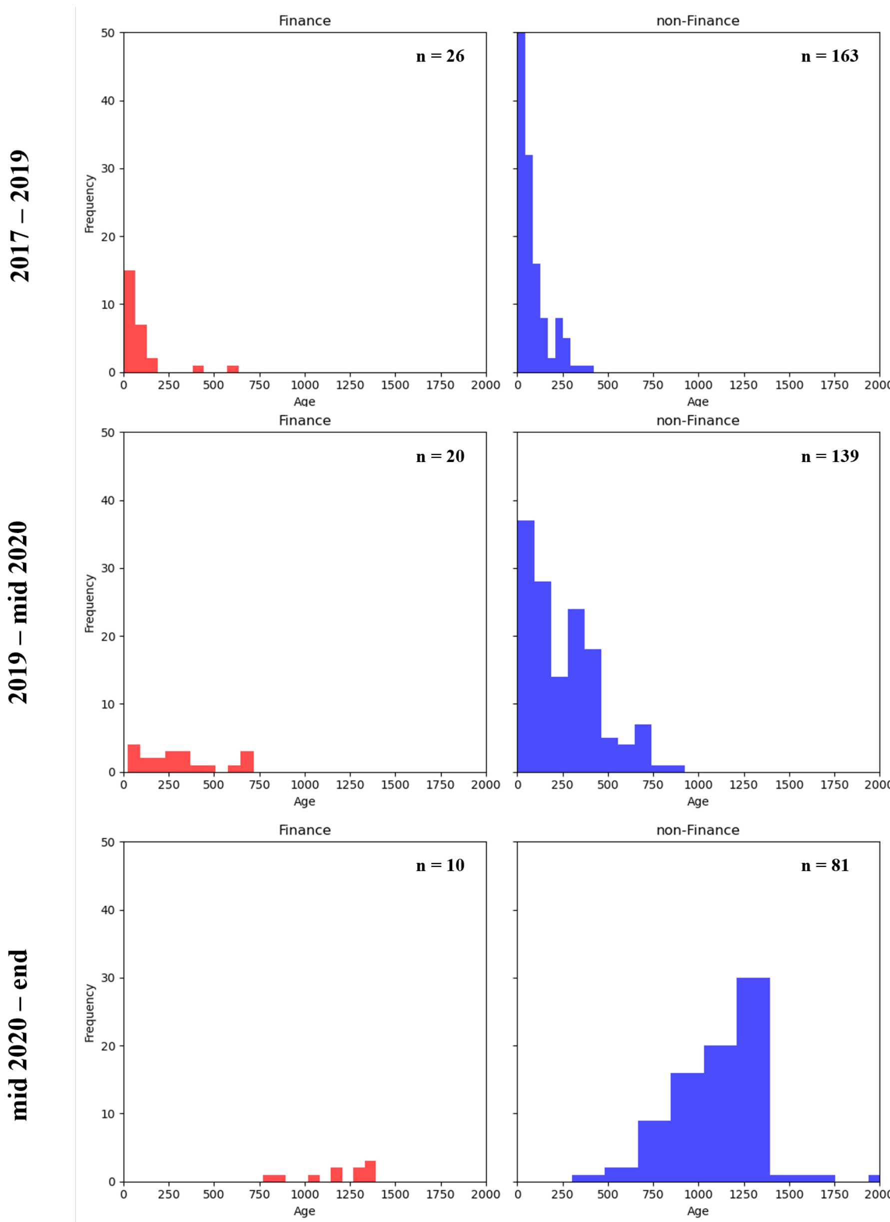

Figure 7: Histogram of dApp age at exit (finance vs. non-finance)

Tables

Table 1: Governance decisions on centralized and decentralized platforms

Governance tools	centralized platforms (e.g., iOS, Android, Amazon, Youtube)	established decentralized platforms (e.g., Wikipedia, OSS)	Blockchain platforms
Transaction validation	platform provider	community	peer-to-peer network
Platform access	platform provider	unlimited	unlimited
Content moderation	platform provider	moderators and arbitrators	none
Setting transaction fees	platform provider	no fees	market mechanism
Changes to the infrastructure	platform provider	ex-post community consent required	ex-ante community consent required

Table 2: Groups of dApps

	dApp categories	examples	dApps
Group 1	finance, exchanges, wallets, insurance, security	Sushi swap, OmiseGo, Status, Nexus Mutual, Chainlink	507
Group 2	identity, property	ENS Manager, Decentraland	45
Group 3	games, marketplaces	Axie Infinity, Cryptokitties	464
Group 4	gambling, social, health	FunFair, Minds, BEAT	397
Group 5	energy, governance, media, storage	Dovui, Aaragon, CryptoTunes, XCloud	177

Table 3: Descriptive statistics and correlations (network level)

Variables	N	Mean	S.D.	1	2	3	4	5	6	7	8	9	10	11	12
1. gasUsed	1,280	45.42	17.15	1											
2. gasUsed group 1	1,280	18.96	18.65	0.88	1										
3. gasUsed group 2	1,280	0.39	0.66	-0.5	-0.28	1									
4. gasUsed group 3	1,280	2.43	1.77	-0.04	-0.25	-0.23	1								
5. gasUsed group 4	1,280	0.86	0.61	-0.09	-0.27	-0.12	0.46	1							
6. gasUsed group 5	1,280	0.56	0.53	-0.21	-0.2	0.09	-0.14	-0.42	1						
7. marketGasPrice	1,280	6.75	12.29	0.73	0.86	-0.16	-0.33	-0.33	-0.15	1					
8. difficultyBomb	1,280	1.08	2.92	-0.48	-0.23	0.25	-0.25	-0.06	-0.05	-0.12	1				
9. networkUtilization	1,280	0.83	0.13	0.73	0.53	-0.6	0.01	-0.2	0.03	0.45	-0.18	1			
10. EtherPrice	1,280	327.48	218.96	0.1	0.11	-0.04	-0.19	-0.62	0.64	0.13	-0.16	0.27	1		
11. EtherVolatility	1,280	0.36	23.46	0.03	0.05	-0.01	0.04	-0.01	0.04	0.05	0.01	0.03	0.07	1	
12. gasLimit	1,280	0.01	0.002	0.93	0.9	-0.41	-0.08	-0.02	-0.29	0.75	-0.31	0.53	0.001	0.03	1

Table 4: 2SLS model with 1st and 2nd stage and OLS benchmark (network level)

	(1) 2SLS 1st stage log($marketGasPrice$)	(2) 2SLS 2nd stage log($gasUsed$)	(3) OLS log($gasUsed$)
difficultyBomb	0.10*** (0.02)		
log($marketGasPrice$)		-0.69*** (0.16)	-0.04** (0.02)
networkUtilization	-3.03*** (0.35)	-1.58*** (0.43)	0.20 (0.19)
networkUtilization2	17.51*** (1.85)	10.38*** (2.60)	-0.33 (0.87)
log($EtherPrice$)	0.09 (0.13)	0.06 (0.08)	0.12** (0.05)
log($EtherVolatility$	-0.02 (0.02)	-0.01 (0.01)	0.001 (0.003)
log($GasLimit$)	3.08*** (1.11)	3.02*** (0.99)	0.53* (0.28)
DThursday	-0.04 (0.03)	-0.03 (0.02)	-0.001 (0.002)
DFriday	0.01 (0.03)	0.005 (0.02)	-0.001 (0.003)
DWednesday	-0.02 (0.02)	-0.01 (0.02)	0.0002 (0.002)
DMonday	-0.05 (0.03)	-0.03 (0.02)	-0.00004
DSaturday	-0.02 (0.04)	-0.01 (0.02)	-0.01 (0.01)
DSunday	-0.03 (0.04)	-0.02 (0.02)	-0.01 (0.01)
D2018	-1.21*** (0.20)	-0.85*** (0.26)	0.13 (0.19)
D2019	-1.61*** (0.29)	-1.11*** (0.30)	-0.005 (0.24)
D2020	-1.30** (0.62)	-0.90** (0.40)	-0.03 (0.27)
Trend	0.001 (0.001)	0.001* (0.0005)	0.001*** (0.0003)
Constant	-13.30 (18.66)	-2.97 (12.00)	-7.81 (6.25)
Observations	1,279	1,279	1,279
R2	0.79		0.94
F Statistic (df = 16; 1262)	305.20***		1,220.08***
C-D Wald F Stat.		85.06	
Stock-Yogo Critical Value		16.38	
Kleibergen-Paap LM Stat.		4.18**	

Note: Heteroskedastic and autocorrelation consistent (HAC) standard errors are shown in parentheses, where the optimal bandwidth (23) is calculated following Newey and West (1987).

Signif. Codes: **: 0.01, **: 0.05, *: 0.1

Table 5: 2SLS model with 1st and 2nd stage and OLS benchmark (network level)

	(1) 2SLS 2nd stage log($gasUsedbyalldApps$)	(2) 2SLS 2nd stage log($gasUsedbygroup1$)	(3) 2SLS 2nd stage log($gasUsedbygroup2$)	(4) 2SLS 2nd stage log($gasUsedbygroup3$)	(5) 2SLS 2nd stage log($gasUsedbygroup4$)	(6) 2SLS 2nd stage log($gasUsedbygroup5$)
log($marketGasPrice$)	-0.45*** (0.14)	-0.0464	0.09 (0.19)	-2.09*** (0.63)	-0.59*** (0.13)	-0.48*** (0.17)
networkUtilization	-1.04*** (0.36)	-0.27 (0.41)	-0.84 (0.61)	-2.37 (1.67)	-0.4368	-1.05** (0.51)
networkUtilization2	6.61*** (2.25)	2.51 (2.58)	2.89 (3.60)	17.04* (10.24)	5.44* (2.81)	7.20** (3.04)
log($EtherPrice$)	0.20** (0.08)	0.39*** (0.08)	0.03 (0.09)	-0.02 (0.23)	-0.93*** (0.09)	0.37*** (0.10)
log($EtherVolatility$)	-0.0000 (0.01)	0.01 (0.01)	-0.02 (0.02)	-0.005 (0.03)	0.02 (0.02)	-0.02 (0.01)
log($gasLimit$)	2.49*** (0.92)	1.56 (1.05)	-0.75 (1.07)	7.61*** (2.28)	1.88** (0.86)	2.68*** (0.91)
DThursday	-0.03 (0.02)	-0.02 (0.02)	0.02 (0.04)	-0.12 (0.08)	-0.0015	-0.09** (0.04)
DFriday	0.01 (0.02)	0.01 (0.02)	-0.04 (0.04)	0.03 (0.07)	-0.02 (0.03)	-0.13*** (0.04)
DWednesday	-0.002 (0.02)	0.004 (0.01)	-0.02 (0.03)	-0.06 (0.05)	-0.03 (0.02)	-0.0024
DMonday	-0.02 (0.02)	-0.01 (0.02)	-0.03 (0.04)	-0.10 (0.07)	-0.06** (0.03)	-0.12*** (0.03)
DSaturday	-0.04 (0.03)	-0.07*** (0.03)	-0.09** (0.04)	0.13* (0.07)	-0.0018	-0.13*** (0.05)
DSunday	-0.04 (0.02)	-0.08*** (0.02)	-0.004	0.14* (0.07)	-0.07** (0.03)	-0.13*** (0.05)
D2018	-1.25*** (0.28)	-1.36*** (0.35)	-0.26 (0.31)	-1.29 (1.15)	-0.66** (0.28)	-0.23 (0.30)
D2019	-1.53*** (0.32)	-1.80*** (0.40)	-0.23 (0.38)	-1.69 (1.43)	-0.41 (0.35)	0.22 (0.38)
D2020	-1.35*** (0.38)	-1.61*** (0.42)	-0.29 (0.44)	-1.90 (1.35)	-0.34 (0.40)	1.37*** (0.42)
Trend	0.002*** (0.0004)	0.003*** (0.0005)	-0.001** (0.001)	0.002 (0.001)	0.0004 (0.001)	-0.003*** (0.001)
Constant	-0.03 (10.36)	-18.54 (12.14)	35.66** (14.67)	16.61 (30.89)	24.97* (13.23)	83.40*** (12.13)
Observations				1,279		
C-D Wald F Stat.				85.06		
Stock-Yogo Critical Value				16.38		
Kleibergen-Paap LM Stat.				4.19**		

Heteroskedastic and autocorrelation consistent (HAC) standard errors are shown in parentheses,
where the optimal bandwidth (23) is calculated following Newey and West (1987).
All models use the first-stage regression reported in Table 4.

Signif. Codes: ***: 0.01, **: 0.05, *: 0.1

Table 6: Interactions with transaction requirements (dApp level)

	(1) log($gasUsed$)	(2) log($gasUsed$)	(3) log($gasUsed$)	(4) log($gasUsed$)	(5) log($gasUsed$)	(6) log($gasUsed$)
log($marketGasPrice$)	-0.66*** (0.21)	-0.64*** (0.21)	-0.73*** (0.21)	-0.59** (0.26)	-0.62** (0.27)	-0.82*** (0.30)
log($marketGasPrice$) × log($avgGas$ Requirement)	-0.06 (0.04)			0.02 (0.05)		
log($marketGasPrice$) × log($avgValue$)		0.14*** (0.04)			0.15** (0.06)	
log($marketGasPrice$) × log($avgToken$)			0.31*** (0.04)			0.40*** (0.09)
log($marketGasPrice$) × $group2$				-0.17 (0.17)	-0.08 (0.18)	0.10 (0.20)
log($marketGasPrice$) × $group3$				-0.042	-0.24 (0.15)	0.03 (0.16)
log($marketGasPrice$) × $group4$				-0.17 (0.14)	-0.15 (0.14)	0.09 (0.16)
log($marketGasPrice$) × $group5$				0.04 (0.14)	0.09 (0.14)	0.23 (0.16)
log($marketGasPrice$) × log($avgGas$ Requirement) × group 2				-0.58*** (0.16)		
log($marketGasPrice$) × log($avgGas$ Requirement) × group 3				-0.24** (0.11)		
log($marketGasPrice$) × log($avgGas$ Requirement) × group 4				-0.019		
log($marketGasPrice$) × log($avgGas$ Requirement) × group 5				-0.003 (0.08)		
log($marketGasPrice$) × log($avgValue$)$group2$					-0.25 (0.17)	
log($marketGasPrice$) × log($avgValue$)$group3$					0.18 (0.16)	
log($marketGasPrice$) × log($avgValue$)$group4$					-0.02 (0.08)	
log($marketGasPrice$) × log($avgValue$)$group5$					-0.10 (0.11)	
log($marketGasPrice$) × log($avgTokens$)$group2$						-0.19 (0.15)
log($marketGasPrice$) × log($avgTokens$)$group3$						-0.05 (0.13)
log($marketGasPrice$) × log($avgTokens$)$group4$						-0.21 (0.14)
log($marketGasPrice$) × log($avgTokens$)$group5$						-0.24** (0.11)
log($EtherVolatility$)	0.01* (0.004)	0.01** (0.004)	0.01* (0.004)	0.01* (0.004)	0.01* (0.004)	0.01* (0.004)
networkUtilization	-1.24*** (0.47)	-1.18** (0.47)	-1.31*** (0.48)	-1.27*** (0.48)	-1.25*** (0.48)	-1.34*** (0.49)
networkUtilization2	8.87*** (3.30)	8.48** (3.30)	9.37*** (3.36)	9.06*** (3.32)	8.96*** (3.36)	9.54*** (3.40)
log($gasLimit$)	1.94*** (0.53)	1.88*** (0.53)	1.95*** (0.54)	1.95*** (0.54)	1.94*** (0.54)	1.99*** (0.55)
Age	-0.002*** (-0.0003)	-0.002*** (-0.0003)	-0.002*** (-0.0003)	-0.002*** (-0.0003)	-0.002*** (-0.0003)	-0.002*** (-0.0003)
Year dummies	YES	YES	YES	YES	YES	YES
Weekday dummies	YES	YES	YES	YES	YES	YES

HAC standard-errors in parentheses.
*Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

Table 7: Interactions with average performance indicators (dApp level)

	(1) log($Gasused$)	(2) log($Gasused$)	(3) log($gasUsed$)	(4) log($gasUsed$)	(5) log($gasUsed$)	(6) log($gasUsed$)
log($marketGasPrice$)	-0.67*** (0.21)	-0.68*** (0.21)	-0.64*** (0.21)	-0.81*** (0.29)	-0.81*** (0.29)	-0.59** (0.26)
log($marketGasPrice$)) × log($avgDailyTxn$)	0.16*** (0.06)			0.39*** (0.08)		
log($marketGasPrice$)) × log($avgDailyEOA$)		0.21*** (0.06)			0.39*** (0.07)	
log($marketGasPrice$)) × log($avgTxnPerEOA$)			-0.03 (0.04)			0.02 (0.06)
log($marketGasPrice$) × $group2$				0.08 (0.19)	0.06 (0.19)	-0.02 (0.15)
log($marketGasPrice$) × $group3$				-0.12 (0.15)	-0.13 (0.15)	-0.33** (0.15)
log($marketGasPrice$) × $group4$				0.01 (0.16)	0.02 (0.16)	-0.16 (0.14)
log($marketGasPrice$) × $group5$				0.22 (0.15)	0.22 (0.15)	0.06 (0.14)
log($marketGasPrice$) × log($avgDailyTxn$) × $group2$				-0.51*** (0.17)		
log($marketGasPrice$) × log($avgDailyTxn$) × $group3$				-0.64*** (0.14)		
log($marketGasPrice$) × log($avgDailyTxn$) × $group4$				-0.47*** (0.13)		
log($marketGasPrice$) × log($avgDailyTxn$) × $group5$				-0.45*** (0.11)		
log($marketGasPrice$) × log($avgDailyEOA$) × $group2$					-0.0448	
log($marketGasPrice$) × log($avgDailyEOA$) × $group3$					-0.55*** (0.13)	
log($marketGasPrice$) × log($avgDailyEOA$) × $group4$					-0.38** (0.15)	
log($marketGasPrice$) × log($avgDailyEOA$) × $group5$					-0.46*** (0.11)	
log($marketGasPrice$) × log($avgTxnPerEOA$) × $group2$						-0.46*** (0.10)
log($marketGasPrice$) × log($avgTxnPerEOA$) × $group3$						-0.28** (0.12)
log($marketGasPrice$) × log($avgTxnPerEOA$) × $group4$						-0.12 (0.08)
log($marketGasPrice$) × log($avgTxnPerEOA$) × $group5$						0.03 (0.10)
log($EtherPrice$)	0.15*** (0.04)	0.15*** (0.04)	0.15*** (0.04)	0.14*** (0.04)	0.15*** (0.04)	0.15*** (0.04)
log($EtherVolatility$)	0.01** (0.004)	0.01* (0.004)	0.01** (0.004)	0.01 (0.004)	0.01 (0.004)	0.01* (0.004)
networkUtilization	-1.22** (0.48)	-1.25*** (0.48)	-1.21** (0.47)	-1.40*** (0.50)	-1.42*** (0.50)	-1.28*** (0.48)
networkUtilization2	8.73*** (3.34)	8.92*** (3.36)	8.66*** (3.29)	10.02*** (3.48)	10.16*** (3.51)	9.12*** (3.33)
log($gasLimit$)	1.88*** (0.53)	1.89*** (0.53)	1.90*** (0.53)	2.08*** (0.55)	2.10*** (0.56)	1.95*** (0.54)
Age	-0.002*** (0.0003)	-0.002*** (0.0003)	-0.002*** (0.0003)	-0.002*** (0.0003)	-0.002*** (0.0003)	-0.002*** (0.0003)
Year dummies	YES	YES	YES	YES	YES	YES
Weekday dummies	YES	YES	YES	YES	YES	YES

HAC standard-errors in parentheses.
*Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

Table 8: Interactions with usage indicators (dApp level)

	(1) log($gasUsed$)	(2) log($gasUsed$)	(3) log($gasUsed$)	(4) log($gasUsed$)	(5) log($gasUsed$)	(6) log($gasUsed$)
log($marketGasPrice$)	-0.44** (0.17)	-0.43** (0.17)	-0.0836	-0.66*** (0.21)	-0.71*** (0.22)	-0.71** (0.29)
log($txnPerEOA$)	1.27*** (0.03)	1.28*** (0.04)	1.17*** (0.06)			
log($marketGasPrice$) $\times$ log($txnPerEOA$)		0.08*** (0.03)	0.08** (0.04)			
log($marketGasPrice$) $\times$ $group2$			-0.15 (0.17)			-0.002 (0.19)
log($marketGasPrice$) $\times$ $group3$			-0.28** (0.13)			-0.15 (0.16)
log($marketGasPrice$) $\times$ $group4$			-0.0252			-0.02 (0.16)
log($marketGasPrice$) $\times$ $group5$			0.05 (0.12)			0.12 (0.15)
log($txnPerEOA$) $\times$ $group2$			-0.03 (0.15)			
log($txnPerEOA$) $\times$ $group3$			0.35*** (0.08)			
log($txnPerEOA$) $\times$ $group4$			0.01 (0.09)			
log($txnPerEOA$) $\times$ $group5$			0.17 (0.11)			
log($marketGasPrice$)$\times$ log($txnPerEOA$) $\times$ $group2$			-0.13 (0.15)			
log($marketGasPrice$)$\times$ log($txnPerEOA$) $\times$ $group3$			-0.001 (0.07)			
log($marketGasPrice$)$\times$ log($txnPerEOA$) $\times$ $group4$			-0.02 (0.05)			
log($marketGasPrice$)$\times$ log($txnPerEOA$) $\times$ $group5$			0.16*** (0.06)			
log($surplusGasPrice$)				0.08*** (0.03)	-0.14*** (0.04)	-0.07 (0.07)
log($surplusGasPrice$)$\times$ log($marketGasPrice$)					0.16*** (0.02)	0.16*** (0.03)
log($surplusGasPrice$)$\times$ group 2						-0.39*** (0.11)
log($surplusGasPrice$)$\times$ group 3						-0.35*** (0.11)
log($surplusGasPrice$)$\times$ group 4						-0.18 (0.11)
log($surplusGasPrice$)$\times$ group 5						0.14 (0.11)
log($marketGasPrice$)$\times$ log($surplusGasPrice$) $\times$ $group2$						0.11** (0.05)
log($marketGasPrice$)$\times$ log($surplusGasPrice$) $\times$ $group3$						0.09* (0.05)
log($marketGasPrice$)$\times$ log($surplusGasPrice$) $\times$ $group4$						-0.05 (0.05)
log($marketGasPrice$)$\times$ log($surplusGasPrice$) $\times$ $group5$						-0.18*** (0.05)
log($EtherPrice$)	0.15*** (0.04)	0.15*** (0.04)	0.15*** (0.04)	0.14*** (0.04)	0.15*** (0.04)	0.15*** (0.04)
log($EtherVolatility$)	0.01** (0.004)	0.01* (0.004)	0.01** (0.004)	0.01 (0.004)	0.01 (0.004)	0.01* (0.004)
networkUtilization	-1.22** (0.48)	-1.25*** (0.48)	-1.21** (0.47)	-1.40*** (0.50)	-1.42*** (0.50)	-1.28*** (0.48)
networkUtilization2	8.73*** (3.34)	8.92*** (3.36)	8.66*** (3.29)	10.02*** (3.48)	10.16*** (3.51)	9.12*** (3.33)
log($gasLimit$)	1.88*** (0.53)	1.89*** (0.53)	1.90*** (0.53)	2.08*** (0.55)	2.10*** (0.56)	1.95*** (0.54)
Age	-0.002*** (0.0003)	-0.002*** (0.0003)	-0.002*** (0.0003)	-0.002*** (0.0003)	-0.002*** (0.0003)	-0.002*** (0.0003)
Year dummies	YES	YES	YES	YES	YES	YES
Weekday dummies	YES	YES	YES	YES	YES	YES

HAC standard-errors in parentheses.
*Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

Table 9: Robustness checks

	(1) Baseline	(2) Alternative Dependent variable	(3) Alternative market gas price (25th percentile)	(4) Alternative market gas price (average gas price)	(5) Alternative market gas price (normalized by ETH supply)	(6) Alternative instrument (block difference)	(7) Outliers (5th-95th percentile gas used)	(8) Subsample (specific difficulty bomb period)
	log($gasUsed$)	log($txnCount$)	log($gasUsed$)	log($gasUsed$)	log($gasUsed$)	log($gasUsed$)	log($gasUsed$)	log($gasUsed$)
log($marketGasPrice$)	-0.69*** (0.16)	-0.63*** (0.15)	-0.80*** (0.20)	-1.83** (0.61)	-0.57** (0.24)	- -0.75** (0.24)	-0.69** (0.19)	-2.70 (2.85)
Observations	1,279	1,279	1,279	1,279	1,279	1,279	1,279	101

HAC standard-errors in parentheses.
*Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

Table 10: Demand curve comparison MEV vs. dApp groups - OLS only

	(1) OLS all dApps log($gasUsed$)	(2) OLS MEV only log($gasUsed$)	(3) OLS Group 1 log($gasUsed$)	(4) OLS Group 2 log($gasUsed$)	(5) OLS Group 3 log($gasUsed$)	(6) OLS Group 4 log($gasUsed$)	(7) OLS Group 5 log($gasUsed$)
log($marketGasPrice$)	0.06^{***} (0.02)	0.26^{***} (0.09)	0.08^{***} (0.02)	-0.10 (0.08)	-0.64^{***} (0.09)	-0.07 (0.08)	0.05 (0.08)
networkUtilization2	1.71^{***} (0.63)	12.88^{***} (4.50)	1.68^{**} (0.69)	-0.24 (2.18)	4.98^{**} (2.03)	-4.95^{*} (2.71)	-6.58^{***} (2.48)
log($EtherPrice$)	0.13^{**} (0.06)	-0.43 (0.29)	0.13^{**} (0.06)	0.70^{**} (0.35)	-0.95^{***} (0.36)	0.36 (0.33)	0.57^{*} (0.31)
log($EtherVolatility$)	-0.003 (0.004)	-0.01 (0.02)	-0.003 (0.004)	-0.0001 (0.02)	-0.02 (0.02)	-0.02 (0.02)	-0.06^{***} (0.02)
log($gasLimit$)	1.75^{***} (0.21)	10.32^{***} (1.26)	1.83^{***} (0.23)	0.78 (1.01)	-0.96 (0.68)	1.13 (0.92)	-1.70^{*} (0.90)
DThursday	-0.01 (0.01)	0.03 (0.05)	-0.01 (0.01)	-0.04 (0.06)	-0.08 (0.05)	0.01 (0.04)	0.002 (0.08)
DFriday	-0.01 (0.01)	0.04 (0.05)	-0.01 (0.01)	0.004 (0.06)	-0.01 (0.05)	-0.01 (0.04)	-0.07 (0.06)
DWednesday	-0.004 (0.01)	0.01 (0.04)	-0.003 (0.01)	-0.05 (0.04)	-0.05 (0.04)	0.01 (0.03)	-0.06 (0.06)
DMonday	-0.004 (0.01)	0.08^{*} (0.04)	-0.001 (0.01)	-0.03 (0.05)	-0.01 (0.06)	0.003 (0.03)	-0.14^{***} (0.05)
DSaturday	-0.01 (0.01)	0.11^{*} (0.06)	-0.02 (0.01)	0.07 (0.08)	0.07 (0.05)	-0.002 (0.04)	-0.08 (0.06)
DSunday	-0.03^{**} (0.01)	0.16^{**} (0.06)	-0.03^{**} (0.01)	0.06 (0.07)	-0.01 (0.05)	0.03 (0.04)	-0.04 (0.07)
Trend	-0.0003 (0.0004)	0.005^{**} (0.002)	-0.0004 (0.0004)	-0.002 (0.002)	0.01^{***} (0.002)	-0.01^{**} (0.002)	-0.001 (0.002)
Constant	35.92^{***} (7.00)	-30.03 (42.79)	37.57^{***} (7.55)	59.20 (37.81)	-185.65^{***} (40.70)	127.24^{***} (44.48)	34.23 (38.74)
Observations	242	242	242	242	242	242	242
R^2	0.93	0.95	0.93	0.18	0.82	0.62	0.36
F Statistic (df = 12; 229)	257.53^{***}	382.87^{***}	250.92^{***}	4.18^{***}	86.84^{***}	31.46^{***}	10.54^{***}

Heteroskedastic and autocorrelation consistent (HAC) standard errors are shown in parentheses, where the optimal bandwidth (23) is calculated following Newey and West (1987).
*Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

Appendix A – Additional formulas

Block time

Ethereum adjusts the mining difficulty for every new block according to the following function:

$$blockTime_b = \frac{miningDifficulty_b}{networkHashRate_{b-1}}$$

Where $miningdifficulty_b$ is the average number of hashes it requires to find a new block and $networkhashrate_{b-1}$ is the number of hashes computed per second by all miners while searching for the previous block.

Mining reward

To incentivize miners to provide their computation service, they are rewarded with a mining reward for every block they find. This reward consists of a static block reward (at the time of writing, 2 Ether) for finding a new block plus the sum of all gas fees (usually measured in $GWei$; 1 Ether = 10^9 GWei) paid by all transactions t which a miner includes in this block. Hence, the mining reward for every block b is:

$$miningReward_b = 2 + \sum_{\forall t \epsilon b} \frac{gasPrice_t \times gasUsed_t}{10^9}$$

Transaction fees

On Ethereum, users only pay for the used gas if the computation is finished before reaching the limit. Also, only the actually used gas is considered for the block gas limit. Accordingly, the fees a user has to pay for a transaction t are computed as follows:

$$transactionFees_t = \frac{gasPrice_t \times gasUsed_t}{10^9}$$

Appendix B – dApp-level analysis

Table 11: Descriptive Statistics

	dapps*days (N = 379,748)	Mean	SD	Min	Max	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)
(1)	gasUsed	180,178	1,288,645	21	85,346,148	1																		
(2)	transactionActivity	893	9,213	1	518,357	0.89	1																	
(3)	EOA	288	3,143	1	168,900	0.82	0.97	1																
(4)	avgGasPricePaid	28	44	0	6,250	0.05	0.05	0.05	1															
(5)	marketGasPrice	14	1	54	0.06	0.06	0.06	0.71	1															
(6)	difficultyBomb	65	210	0	1,610	-0.01	-0.01	-0.01	-0.09	-0.13	1													
(7)	networkUtilization	301	195	84	1,385	0.02	0.02	0.02	0.25	0.23	-0.2	1												
(8)	networkUtilization2	0.24	20	-228	153	0.01	0.01	0.01	0.01	0.06	0.001	0.06	1											
(9)	log($EtherPrice$)	0.85	0.1	0.3	0.98	0.03	0.04	0.04	0.37	0.54	-0.12	0.34	0.04	1										
(10)	log($EtherVolatility$)	0.73	0.17	0.09	0.97	0.04	0.04	0.04	0.4	0.57	-0.12	0.36	0.04	1	1									
(11)	gasLimit	9,278	1,739	6,704	12,485	0.06	0.06	0.06	0.49	0.77	-0.21	0.12	0.06	0.51	0.54	1								
(12)	Age	415	322	1	1,280	0.05	0.08	0.08	0.23	0.33	-0.11	-0.15	0.03	0.24	0.24	0.49	1							
(13)	avgGasRequirement	322	478	21	9,900	0.04	-0.02	-0.02	-0.05	0.01	-0.02	-0.08	0.00	0.01	0.01	0.05	-0.1	1						
(14)	avgValue	366	3,656	0	99,002	0.01	0.00	0.00	0.01	0.01	0.00	0.02	0.01	0.01	0.01	0.01	0.01	0.01	1					
(15)	avgTokens	2,781	13,909	0	185,968	0.05	0.04	0.03	0.02	0.01	0.01	0.02	0.01	0.00	0.00	0.01	0.07	-0.03	0.01	1				
(16)	avgDailyTxn	893	5,695	1	71,089	0.53	0.62	0.61	0.02	0.01	0.00	0.00	0.00	0.01	0.01	0.01	0.05	-0.04	-0.01	0.06	1			
(17)	avgDailyEOA	288	1,954	1	24,975	0.5	0.61	0.62	0.02	0.01	-0.01	0.00	0.00	0.01	0.01	0.01	0.05	-0.04	-0.01	0.05	0.99	1		
(18)	avgTxnPerEOA	6	20	1	354	0.03	0.01	-0.01	0.01	0.04	-0.01	-0.02	0.01	0.02	0.02	0.05	-0.04	0.09	-0.01	-0.02	0.01	-0.02	1	
(19)	txnPerEOA	6	44	1	4,488	0.07	0.02	-0.01	0.00	0.01	0.01	-0.01	0.001	-0.01	-0.01	0.01	-0.04	0.04	-0.01	-0.01	0.01	-0.01	0.46	1
(20)	surplusGasPrice	19	30	-129	6,249	0.04	0.05	0.04	0.93	0.44	-0.06	0.23	-0.01	0.25	0.27	0.29	0.15	-0.08	0.01	0.03	0.02	0.02	0.00	-0.01

The baseline specification for our network level is analogous to our dApp level specification but without dApp-level fixed effects:

$$\begin{aligned}\log(gasUsed_{td}) = \alpha_0 + \alpha_1 \log(marketGasPrice_t) + \alpha_2 networkUtilization_t + \\ \alpha_3 networkUtilization_t^2 + \alpha_4 \log(EtherPrice_t) + \alpha_5 \log(EtherVolatility_t) + \\ \alpha_6 \log(gasLimit_t) + \mu_{day\ of\ week} + \mu_{year} + \mu_d + trend + u_t\end{aligned}$$

where gas used is the equilibrium gas demand for each dApp d in the period t (day). We chose a log-log specification for gas used and market gas price to be able to interpret α_1 as the price elasticity of the demand. Due to the skewed distributions of Ether price, Ether volatility, and the gas limit, we use log-transformed versions of these variables in our specification. The network utilization allows us to control for the degree to which miners use the available block gas limit on a given day and has been used by prior scholars as a measure of network congestion (Donmez and Karaivanov 2021). We also add a quadratic term to account for the nonlinear relationship between gas price and network utilization.[26] In addition to these variables, we also control for the intrinsic growth of the dApp by adding $\mathrm{age}_{\mathrm{dt}}$ as the number of days since the dApp entered the platform and specify μ_{d} as dApp fixed effects, $\mu_{\mathrm{dayofweek}}$ as a day of week fixed effects, μ_{year} as a year fixed effects, and u_{t} as the error term.

Baseline dApp-level results

Following our baseline specification, Table 12 reports the results of our 2SLS demand curve estimation. Column 1 presents the first stage results, where we predict the gas price (log(Market gas price)) with our IV (difficulty bomb). Column 2 presents the second stage results, where we use the predicted gas price to estimate the price elasticity of the gas demand (log(Gas used)).

To establish robustness, we ran a series of alternative models of the network-level analysis similar to the robustness checks reported in the main paper. Table 13 reports the results of these robustness checks.

Differing Demand Curves per Group

Column 3 in Table 12 reports the different demand curves for each group of dApps. We obtain these demand curves by interacting the instrumented market gas price with the group of a dApp.

[26] We also compute the same model with a threshold specification where we added only the linear term and dummy variable that takes on the value one if the utilization level exceeds 90%. They were qualitatively the same regarding the magnitude and significance of the coefficients we obtained.

With a positive and significant coefficient (0.27) for our reference group (finance dApps), our results suggest that the demand curve for these dApps in upward-sloping. An explanation for this upward-sloping demand curve could be that the entry of additional finance-related dApps has caused an influx of high willingness-to-pay customers and that the network effects these finance-related dApps realize compensated for the higher transaction fees these transaction senders had to pay. This explanation is in line with prior research that describes networked goods (e.g., financial services) by irregularities such as an upward-sloping demand curve for low quantity levels (Economides and Himmelberg 1995). Particularly, if a service relies on strong network effects, no one will pay for the product if no one else uses it. Although the entry of high willingness-to-pay users is typically beneficial for a platform, the fact that we observe downward-sloping demand curves in the form of negative moderations of all other groups poses a danger that, particularly in times of high transaction fees, dApps from other groups are not used anymore and finally have to leave the platform. This reduction of complement heterogeneity can ultimately harm the long-term attractiveness of Ethereum, especially as a general-purpose platform.

Table 12: Demand curve estimation – baseline model (dApp level)

	(1) $\log(marketGasPrice)$	(2) $\log(gasUsed)$	(3) $\log(gasUsed)$
difficultyBomb	0.20*** (0.0000)		
$\log(marketGasPrice)$		-0.64*** (0.21)	0.27*** (0.05)
$\log(EtherPrice)$	-0.0004 (0.01)	0.15*** (0.04)	0.18*** (0.04)
$\log(EtherVolatility)$	-0.01*** (0.0004)	0.01** (0.004)	0.02*** (0.003)
networkUtilization	-2.36*** (0.06)	-1.20** (0.47)	0.30*** (0.11)
networkUtilization2	16.30*** (0.37)	8.59*** (3.29)	-1.89*** (0.68)
$\log(gasLimit)$	2.40*** (0.03)	1.89*** (0.53)	0.13 (0.20)
Age	0.001*** (0.0000)	-0.002*** (0.0003)	-0.002*** (0.0002)
Year2018	-0.82*** (0.02)	-0.68*** (0.22)	-0.09 (0.15)
Year2019	-1.09*** (0.02)	-0.66*** (0.25)	0.07 (0.15)
Year2020	-0.95*** (0.02)	-0.28 (0.24)	0.36** (0.16)
weekdayThursday	-0.02*** (0.001)	-0.03*** (0.01)	-0.0001
weekdaysFriday	0.02*** (0.001)	-0.02** (0.01)	-0.03*** (0.01)
weekdaysWednesday	-0.005*** (0.001)	-0.001 (0.01)	0.002 (0.01)
weekdaysMonday	-0.02*** (0.001)	-0.03*** (0.01)	-0.02** (0.01)
weekdaysSaturday	0.01*** (0.002)	-0.07*** (0.01)	-0.08*** (0.01)
weekdaysSunday	0.01*** (0.002)	-0.08*** (0.01)	-0.09*** (0.01)
$\log(marketGasPrice)group2$			-0.43*** (0.15)
$\log(marketGasPrice)group3$			-0.64*** (0.12)
$\log(marketGasPrice)group4$			-0.49*** (0.10)
$\log(marketGasPrice)group5$			-0.28*** (0.09)
Observations	370,392	370,392	370,392
R2	0.78	0.11	
Incremental F	121.39		
C-D Wald F Stat.	2542.47	118.07	
Stock-Yogo Critical Value	16.38	26.87	
Kleibergen-Paap LM Stat.	70.04***	25.16***	

HAC standard-errors in parentheses.
*Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

Table 13: Robustness checks

	(1) Baseline	(2) Alternative Dependent variable	(3) Alternative market gas price (25th percentile)	(4) Alternative market gas price (average gas price)	(5) Alternative instrument (block difference)	(6) Outliers (5th-95th percentile gas used)	(7) Subsample (specific difficulty bomb period)
	$\log(gasUsed)$	$\log(txnCount)$	$\log(gasUsed)$	$\log(gasUsed)$	$\log(gasUsed)$	$\log(gasUsed)$	$\log(gasUsed)$
$\log(marketGasPrice)$	-0.64*** (0.21)	-0.42** (0.19)	-0.57*** (0.18)	-0.82*** (0.26)	-1.03** (0.45)	-0.58*** (0.20)	-1.48* (0.87)
Observations	370,392	370,392	370,392	370,392	370,392	370,392	35,756

HAC standard-errors in parentheses.
*Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

Appendix C – dApp age at exit - additional plots

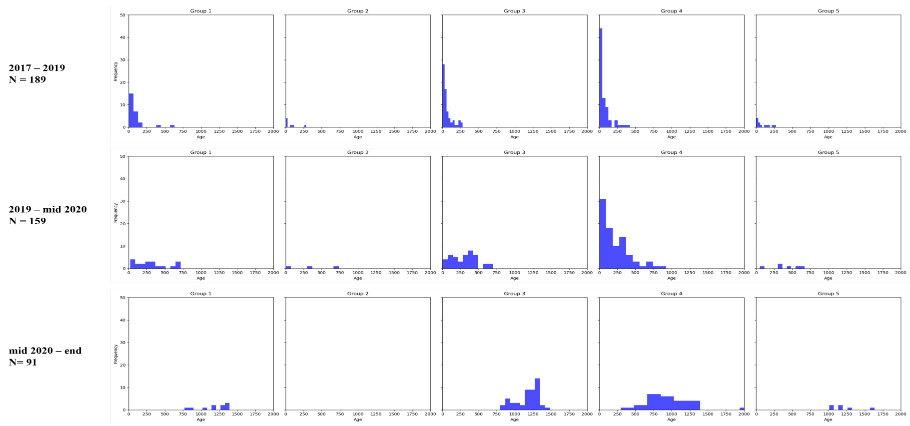

Figure 8: Histogram of dApp age at exit - all groups

Appendix D – Supplementary survival analysis

To investigate the impact of Ethereum's transaction validation mechanism on platform complements' heterogeneity, we examine our explanatory variables' simultaneous effect on the overall hazard-rate function by using the semi-parametric Cox proportional-hazards regression analysis (Cox 1972). Previous scholars have used Cox-proportional hazard models to study market exit or entry (e.g., Agarwal and Gort 2002, Huang et al. 2013). In our benchmark specification, we estimate the hazard of dApp d leaving the market on day t as:

$$h_{dt} = h_o(t) exp\{\beta_x' x_t\}$$

Where h_0(t) is the baseline hazard, $x_t$is a vector of explanatory and control variables pertaining to time t. With this model, we are not interested in predicting the exit time but the effect of gas price as a time-dependent covariate. For the analysis, we cluster the standard errors on the dApp level to control for heteroskedasticity and nonindependence of observations. Further, we stratify our observations by the group of the dApp. This allows us to account for different baseline hazard rates between the groups of dApps. To measure market exit, we leverage the fact that stateofthedapps.com reports the status of dApps and classifies discontinued dApps as "abandoned." For the exact timing of the market exit, we take the date of the last transaction a dApp has received. Table 14 reports the results of our analysis. Column 1 shows our benchmark specification. Column 2 depicts the gas price interacted with the group of the dApp.

Our benchmark specification shows no significant impact of the gas price on the survival of a dApp. However, after interacting the gas price with the group of a dApp (Column 2), we find that a 10% increase in the Market price (∼0.095 increase in log(Market price) is associated with a reduction of the hazard rate (β = -1.7; hazard rate = exp(0.095×-1.7) = 0.851) by around 16.9% for our base category (group 1, finance dApps). The positive and (except for group 3) significant interactions indicate that all other groups of dApps profit less from a higher gas price and face a higher likelihood of market exit. For instance, for group 2, the hazard rate decrease only equals 10.9% (exp((-1.7 + 0.49) ×0.095)=0.891).

The results of our hazard model suggest that an increase in the market gas price reduces the likelihood of a market exit on a given day, but groups differ significantly regarding this effect. Especially when considering that the gas price fluctuates quickly and sometimes doubles or even triples within a month (e.g., January 2018, June 2020 at the start of the Defi hype), these results can be of economic significance. Further, the result seems plausible as an increase in the gas price is typically the consequence of increased demand for gas caused by more transaction activity with dApps. Again, however, we can see that dApps from group

Table 14: 2SLS model with 1st and 2nd stage and OLS benchmark (network level)

	(1) all dApps stratified by group	(2) all dApps stratified by group
log($MarketgasPrice$)	0.02 (0.09)	-0.187
log($MarketgasPrice$) × $group2$	0.49** (0.23)	
log($MarketgasPrice$) × $group3$	0.15 (0.10)	
log($MarketgasPrice$) × $group4$	0.21** (0.09)	
log($MarketgasPrice$) × $group5$	0.22* (0.12)	
networkUtilization	-6.68 (8.24)	-6.89 (8.18)
networkUtilization2	4.01 (5.32)	4.15 (5.28)
log($EtherPrice$)	-0.04 (0.14)	-0.02 (0.14)
log($EtherVolatility$)	0.01 (0.04)	0.01 (0.04)
log($gasLimit$)	1.07 (0.71)	1.11 (0.71)
Year of entry dummies	YES	YES
Observations	783,619	783,619
Market exit events	399	3991
Log-likelihood	-2,088.39	-2,083.79

Note: Robust standard errors are clustered at the group level and reported in parentheses. Hazard ratios can be calculated by exponentiating the coefficients reported for each variable. Signif. Codes: ***: 0.01, **: 0.05, *: 0.1

one benefit more from this effect than other dApps and thus have an overall higher likelihood of staying in this market. This differentiating effect is problematic as it corroborates our main argument by showing that a market for transactions disproportionately favors a specific type of dApps and thus leads to a long-run reduction of the heterogeneity of dApps offered on the Ethereum platform.