Limits to Orchestration in Decentralized Platforms: Exploring the Effect of Ethereum's Transaction Verification Mechanism on dApp Heterogeneity

Hanna Halaburda
NYU Stern
hhalaburda@gmail.com

Daniel Obermeier
NYU Stern
do2169@stern.nyu.edu

Feb 15, 2023

Abstract

Blockchain technology disintermediates digital platforms by substituting a centralized authority with a market mechanism that ensures automated enforcement of transactions following pre-defined rules. Although this substitution is hailed as facilitating Web 3.0, a new era of the internet that promises more inclusive and democratic digital platforms, it also limits the platform provider's toolset to orchestrate a healthy and appealing ecosystem of platform complements. Based on a sample of 1,560 decentralized applications (dApps) on the Ethereum blockchain, we show that Ethereum's gas fee mechanism favors finance dApps at the cost of crowding out dApps from other categories and reducing the heterogeneity of complements offered on the platform. This finding highlights that blockchain platforms that rely on a similar transaction verification mechanism will struggle to become the general-purpose platforms necessary to realize the promises of Web 3.0.

Keywords: Blockchain, Platform Economics, Within Platform Competition, Decentralization, DApps

Limits to Orchestration in Decentralized Platforms: Exploring the Effect of Ethereum's Transaction Verification Mechanism on dApp Heterogeneity

1. Introduction

Blockchain technology disintermediates digital platforms by substituting a centralized authority with a market mechanism that ensures automated enforcement of transactions following pre-defined rules (Nakamoto 2008). According to proponents of blockchain technology, this disintermediation limits a platform provider's ability to modify the platform rules or exclude complementors unilaterally and allows platform architects to design platforms where the created value is distributed more evenly among all participating parties (Catalini and Tucker 2018, Vergne 2020). Based on these promises, Gavin Wood, one of Ethereum's founding fathers, envisioned that blockchain technology will enable what he refers to as Web3.0, a new form of the World Wide Web that is more fair, democratic, and free from powerful platform intermediaries that exploit their users' data (Wood 2014a). With this vision, he spurred a whole new industry that aims to disrupt prevailing digital platforms across industries such as finance, gaming, insurance, and health.

However tempting this vision might be, it is also important to consider that disintermediation is no panacea free from limitations. For example, it is commonly known that blockchain platforms bear higher coordination costs as protocol changes require a consensus by the community and higher storage costs as the same data is replicated across different nodes (Pereira et al. 2019). In this study, we take a platform orchestration perspective and focus on another important limitation recently receiving burgeoning interest. This limitation is that blockchain platforms truncate the platform provider's strategic tools to prioritize some transactions over others to orchestrate an appealing set of third-party applications (platform complements) and steer the direction of innovation when necessary (Leiponen et al. 2021).

One of platform providers' most powerful strategic tools is their ability to set prices and engage in price discrimination to enhance the quality of services offered on the platform (e.g., Lin 2020, Liu and Serfes 2013, Wang and Wright 2017). Blockchain platforms eliminate this tool, as no entity has the power to set prices for transacting on the platform unilaterally. On blockchain platforms, the transaction price and how it is set are inherent parts of the reward mechanism necessary to incentivize nodes to exert the effort to maintain the network. Although the platform providers can initially design the overall reward and transaction fee mechanism, they cannot interfere with how the price of a specific transaction is set after the system is launched.

Currently, most blockchain platforms like Bitcoin and Ethereum rely on a market mechanism that sets the price for transacting on the platform (Buterin 2014, Nakamoto 2008). However, for this market mechanism to work, they also restrict the supply of transactions.[1,2] The limited supply of transactions in combination with a market mechanism has led to skyrocketing transaction fees in the past. As a result, some dApp providers saw a decline in their dApp usage and decided to leave the platform. Most prominently, Dapper Labs (dapperlabs.com), the developer of the CryptoKitties collectibles game, left and developed their own blockchain platform called "Flow," which is specifically tailored to the needs of NFT collectible games. At the time of writing, Flow hosts 427 dApps.[3] Most of these dApps are gaming and NFTs collectibles dApps and are exclusive

---

[1] Restricting the supply is also necessary to maintain a sufficient level of decentralization. Not limiting the supply would favor nodes with more computational power that can compute and verify more transactions and thus exclude smaller nodes with less computational power.
[2] While changing the supply will in general impact the price, it does not allow for price discrimination. Further, changing the supply also requires a consensus, e.g., vote by all miners on the Bitcoin network and a EIP (Ethereum Improvement Proposal) or community vote on Ethereum.
[3] https://www.flowverse.co/, accessed on 02/01/2023.

to the Flow blockchain. The most famous example is NBA Top Shots (nbatopshot.com) which has already attracted more than $1.1b in sales until January 2023.[4] The emergence of the Flow blockchain can be seen as a direct consequence of Ethereum's inability to regulate the market mechanism and stay attractive for all types of complements. To understand if the market exit of dApp providers like Dapper Labs underlies a systematic pattern, our research aims to investigate the consequences of using a market mechanism to determine transaction fees from a platform orchestration perspective. Specifically, we study whether blockchain platforms—which remove the platform provider's ability to set prices and substitute it with a market mechanism—are a viable blueprint for platforms that aspire to host a variety of different applications and become general-purpose platforms.

We argue that such a market mechanism prioritizes complements only based on the transaction fee sensitivity of their users. Whereas this leads to an efficient allocation for homogenous transactions (e.g., like transactions on the Bitcoin network), it can lead to long-run inefficiencies in the case of heterogenous complements as it favors some types of complements over others based on their current user's transaction fee sensitivity but not on the value that the complement might provide in the future. It does so by adding an additional externality in the form of congestion costs to the already existing competition between complements of the same category: if one complement attracts more users and thus increases the demand for transactions, the transaction fees for all other complements—irrespective of the service they offer—rise as well, as they all compete for the same supply of transactions. This is problematic because, as we show later (Section 4), there are several characteristics other than the quality of a complement that determines its users' sensitivity toward transaction fees. Especially in times of congestion and high transaction fees, some complements

[4] https://www.flowverse.co/applications/nba-top-shot, accessed on 02/01/2023.

will be used less, and as the platform provider has no tools to protect them, if necessary, they have to abandon the platform even if it would be overall beneficial for the platform if the complement would stay in the long run.

Such an unregulated reduction of complement heterogeneity is not desirable, as we know from the literature on platform competition that users value the diversity of platform complements. Thus, an unsolicited reduction of complements can hamper a platform's potential to leverage same-side and cross-side network effects (see Rietveld and Schilling 2020). Further, it also questions the neutrality of the blockchain and raises concerns about how this mechanism influences investment incentives for complementors and platform providers that are similar to the discussion around net neutrality (Choi and Kim 2010). Finally, it also questions whether blockchain platforms that rely on a market mechanism to enforce the correct execution of transactions will be a viable option for Web 3.0, where all web applications have to run fully decentralized.

Despite the important implications and the potentially detrimental effects of using a market mechanism to ensure the automated verification of transactions, there is scant research investigating how such a mechanism and the lack of strategic tools to protect complements, if necessary, impacts the heterogeneity of complements offered on a platform. To fill this void, we ask the following research questions: how does a market mechanism for the decentralized verification of transactions affect the usage of platform complements? What complements will be offered on blockchain platforms in the long run?

To answer these questions, we use the context of the Ethereum blockchain. Ethereum provides a unique opportunity to study our research questions for three reasons. First, Ethereum was the first platform to enable smart contracts, which are computer scripts that enable complementors to offer web applications (Buterin 2014). As these applications run on top of a blockchain, they are also referred to as decentralized applications or dApps (Wu et al. 2021). Accordingly, Ethereum qualifies as a multi-sided platform where complementors can offer arbitrary services to platform users. Ethereum is also the most popular platform for dApps, offering services in categories such as finance, gaming, social, insurance, and health. Second, Ethereum uses a market mechanism to allocate the limited supply of transactions among transaction senders (i.e., users of a dApp). This market mechanism resembles a first-price auction where users must bid on how much they are willing to pay for the computational effort required by their transaction (Roughgarden 2020). Third, Ethereum served as the blueprint for many other blockchain platforms that now use a similar mechanism to allocate transactions and thus enhances the generalizability of our results.

For our empirical strategy, we use daily transaction data from a sample of 1,590 dApps on Ethereum and estimate different demand curves for different groups of dApps. To address the endogeneity issues arising from the simultaneous determination of transaction fees by demand and supply, we introduce Ethereum's difficulty bomb as a novel demand-side instrument that has led to exogenous variation in the supply of transactions.

Our analysis yields several important findings. First, by finding a downward-sloping demand curve, we can confirm that the law of demand also applies to transactions on Ethereum. While this finding seems theoretically trivial, the ongoing debate on the prevalence of speculation activity, extreme volatility, and illicit transaction conduct questions whether blockchain platforms are subject to standard supply and demand dynamics comparable to other financial markets (Foley et al. 2019, Li et al. 2018) and thus calls for empirical clarity before scholars can move on with further empirical inquiries. Second, we find that different groups of dApps significantly vary regarding their sensitivity towards transaction fees and that, in times of congestion, finance applications crowd out transactions to other applications by increasing the market price for transacting on the network. Third, our results suggest that building network effects and bundling transactions more efficiently are the only options a dApp has to influence its sensitivity towards transaction fees.

With our research, we contribute threefold. First, we contribute to the platform literature by exploring how competition induced by a market mechanism for the allocation of transactions affects the heterogeneity of complements offered on such platforms in the long run. These insights are not only important as they allow us to gauge how competitive such decentralized platforms are in comparison to their centralized counterparts but also because they help us to understand that it is currently unlikely that one of the existing blockchain platforms will be able to cater towards the need of all types of dApps and dominate all other platforms. Second, we also contribute more specifically to the literature on platform orchestration by extending it to the realm of decentralized platforms and discussing which orchestration tools might still work when the ability of the platform provider to steer transaction activity is limited. Finally, by proving that the basic law of demand also applies to transactions on blockchain platforms that offer smart contract-based applications and providing a novel instrument that helps to overcome endogeneity problems, we pave the way for future scholars that want to leverage the rich data a blockchain platform provides to investigate the economic dynamics on blockchain platforms further.

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 provides a conceptual framework that is the basis for our empirical analysis. 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 concludes with implications for platform providers, complementors, and policymakers and discusses avenues for further research.

2. Related Literature

For the theoretical foundation of our work, we draw on two streams of prior research.

Research on platform competition and platform orchestration

The first stream of literature is the literature on platform competition. This stream stems from early work in the domain of standard setting and standards battles (e.g., Church and Gandal 1992, Cusumano et al. 1992, Shapiro and Varian 2010) and the seminal work by Katz and Shapiro (1985) and Farrell and Saloner (1986) on network effects. Prior work typically focuses on how platform providers can use strategic tools such as setting prices (e.g., Brynjolfsson and Kemerer 1996, Gandal 1994), investment in quality (e.g., Choi 1994), or subsidizing complements (e.g., Riggins et al. 1994) to their competitive advantage in setting with strong network effects.

Building on this stream of research, platform governance, and orchestration have emerged as strong research themes in the platform competition literature (Rietveld and Schilling 2020). This research takes a macro perspective and investigates how the rules of a platform are set and enforced, how these rules influence the behavior of platform participants, and how the participants' consequential behavior impacts the overall outcome of the platform. Therefore, when we refer to a platform orchestration perspective in our context, we refer to this stream of literature and seek to examine how the rules of a platform attract complements to the ecosystem or provoke their exit from the platform.

It should be noted that already early research that examined platforms' decisions regarding pricing (e.g., Brynjolfsson and Kemerer 1996), openness/control (e.g., Ghazawneh and Henfridsson 2013, Parker and van Alstyne 2018), or platform evolution (e.g., Tiwana et al. 2010) and platform ecosystems (e.g., Parker and van Alstyne 2017) more generally implicitly addressed governance and orchestration questions. Yet, only recently has research started to investigate more explicitly how platform providers' governance and orchestration strategies influence the ecosystem of complementors and the platform's overall performance. For instance, Tudón (2022) investigates the platform providers' trade-off between fostering entry of new complements and preventing congestion of the platform and finds that consumer welfare would drop significantly without prioritization on the supply side. Similarly, Panico and Cennamo (2020) investigate the effect of too many complements on the quality of the ecosystem depending on the nature of increasing returns of the complementors and find that if network effects of complementors diminish with their network size, a larger network of complementors will dilute the average complement quality. With this finding, both studies question the often-oversimplified tenet of the network affects literature that a greater breadth and depth of the network is typically considered attractive to consumers. This idea is also echoed by other scholars who suggest that too many complements may result in coordination problems, increase coordination costs, and decrease consumers' value (e.g., Boudreau 2012, Casadesus-Masanell and Hałaburda 2014, Markovich and Moenius 2009).

Regarding the governance of the platform, O'Mahony and Karp (2020) investigate how the decentralization of decision rights on a platform influences participation on a platform. Based on an in-depth case study, they find that although the benefits depend on the platform's products, participants, and markets, for most of the participants in their sample, participation increase with the platform's transition towards decentralized leadership. Related but in the context of blockchain-based platforms, Chen et al. (2021) find an inverted-u-shaped relationship between the decentralization of blockchain platforms and developer activity. Taken together, this literature emphasizes that platform providers must make careful strategic decisions about how many and what type of complements they want to attract to join the platform.

We add to this stream of literature by investigating blockchain platforms from a platform orchestration perspective. Blockchain platforms are an interesting novel phenomenon as they provide an alternative blueprint for established centralized platforms. They substitute a centralized authority with a market mechanism that ensures the correct and automated enforcement of transactions. By doing so, they truncate the platform provider's strategic tools to attract or exclude complements by setting prices, offering subsidies, or limiting entry. Hence, blockchain platforms limit the power of a strong "visible" hand by shifting the agency towards the "invisible" hand of a decentralized market. Although platform providers can define the initial rules of this market, they cannot interfere with them afterward. Due to the importance of a healthy ecosystem of complements for a platform's success, it is paramount to understand how the market mechanism used on blockchain platforms to verify transactions influences what types of complements will be offered on such platforms.

Research on transaction fees on blockchain platforms

The first stream is the nascent literature that studies transaction fee mechanisms on blockchain platforms. Within this literature, scholars have already started to characterize blockchains as marketplaces where miners offer their services to transaction senders and study the dynamics of these marketplaces with different theoretical perspectives. For instance, Basu et al. (2019) and Easley et al. (2019) build game theoretic models to analyze how Bitcoin's fee mechanism causes high variability in transaction fees and thus might deter miners (Basu et al. 2019) and users (Easley et al. 2019). Other scholars like Huberman et al. (2017) and Donmez and Karaivanov (2021) use queuing theory to investigate the implications of transaction fee mechanisms on blockchains.

Huberman et al. (2017) use this theoretical lens to study the entry and exit of miners and find that Bitcoin's transaction fee mechanism protects users from monopoly pricing. Donmez and Karaivanov (2021) use queuing theory to investigate the determinants of transaction fees and reveal that changes in transaction demand and the type of transactions are important factors associated with higher fees. The third stream of researchers builds on auction theory (e.g., Lavi et al. 2017). Most notably, Ilk et al. (2021) take a supply and demand perspective on Bitcoin's transaction fee mechanism and show that the basic forces of demand and supply determine the price of transactions on the Bitcoin platform. They also find that due to a relatively inelastic demand curve and a comparatively elastic supply curve, Bitcoin's current transaction fee mechanism can efficiently self-regulate transaction fees as increasing fees stimulate mining by a larger magnitude than dampening demand. In addition to the literature specifically dedicated to transaction fees on blockchain platforms, there is ample more general research on the economics of cryptocurrencies and on blockchain mining, some of which also address transaction fees and their implications as a peripheral topic. Regarding the general microeconomic forces, Halaburda et al. (2020) provide a general review. Regarding mining, for instance, Houy (2016) and Cong et al. (2021) provide a general analysis of Bitcoin's mining game and miners' behavior. Kroll et al. (2013) scrutinize the security of Bitcoin's mining mechanism and conclude that transaction fees only have limited importance. Arnosti and Weinberg (2018) develop a model that considers heterogenous cost structures among miners and explains how this heterogeneity fosters the concentration of mining power. Finally, Sapirshtein et al. (2016) study the equilibrium between miners and conclude that a proper design of the transaction fee mechanism only produces a reliable system in equilibrium if miners are sufficiently small.

Although all these prior accounts either implicitly or explicitly focus on the implications of the mining process and develop suggestions on how to improve the protocol, they only focus on the implications of the transaction fees mechanism for the miner or the users (i.e., transaction senders). Despite their importance for the long-run success of second-generation platforms, like Ethereum, that enable third parties to offer additional services in the form of dApps, the consequences of a transaction fee mechanism for these complementors are currently neglected. To fill this void, our research adds to this stream of literature by being the first that investigate the implications of the transaction fee mechanism on platform complements (i.e., dApps). Arguably for complementors, the transaction allocation mechanism can have severe implications if skyrocketing transaction fees prevent users from sending transactions to the dApp.

On the empirical side, only a few accounts estimate the impact of transaction fees on the usage of blockchain platforms, and most are focused on the Bitcoin blockchain (e.g., Easley et al. 2019, Ilk et al. 2021). For example, Ilk et al. (2021) provide empirical evidence that the basic economic theory (i.e., the law of demand) also holds for transactions on blockchains by finding a downwards-sloping demand and an upwards sloping supply curve for transactions on the Bitcoin blockchain. For Ethereum, this evidence is still lacking. Although few accounts investigate the relationship between network congestion and gas prices (Donmez and Karaivanov 2021) or gas prices and throughput (Azevedo Sousa et al. 2021, Spain et al. 2020), or how high gas fees antagonize Ethereum's goal of inclusion and democratization by excluding users who cannot afford the increasing gas fees (Cong et al. 2022), there is a paucity of research that analyzes supply and demand dynamics on Ethereum and in particular how these impact the usage of dApps. We, however, argue that the possibility to offer dApps distinguishes the potential of Ethereum and demarcates the potential of similar decentralized platforms to compete with established centralized platforms like Apple's iOS or Google's Android. If we want to understand if decentralized platforms can deplete the prevalence of established centralized platforms, one important step is to understand under what conditions platform complements must work on such decentralized platforms. To this end, our work also adds to the related literature by presenting empirical evidence for the impact of a decentralized transaction verification mechanism on the usage of dApps. As an aside, we also provide initial empirical evidence that basic economic theory applies to transactions on Ethereum and thus pave to way for further economic inquiries.

3. Background

Ethereum is the second-largest blockchain platform, with a market capitalization of 300 billion USD and over 1.2 million daily transactions.[5] It is the context of our study as it 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 blockchain-based apps running on top of the blockchain (Buterin 2014). As transactions differ regarding complexity and thus require differing computational effort to be executed by miners, Ethereum introduced a new market mechanism that incentivizes miners to compute more computationally expensive transactions. 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).

[5] https://etherscan.io/ (retrieved on March 30rd, 2022).

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 pre-defined 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 pre-defined 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 or 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 but 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).

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

[6] Obermeier and Henkel (2022) discuss that smart contracts only remove the necessity of trust if the users have read and completely understood its source code. In practice, due 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.

arbitrary service. According to DappRadar, Ethereum currently hosts more than 3,600 dApps across categories such as finance, games, gambling, insurance, social media, property, and digital identity. 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, 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.[7]

Ethereum's market for transactions

To verify and enforce transactions users send to dApps, Ethereum uses a decentralized transaction verification and enforcement 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, Easley 2019) and a market for data space more specifically (Ilk 2020). We also characterize Ethereum's transaction verification 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 verified 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). This puzzle requires miners to brute-force numerous hashes until they find a hash that satisfies the conditions imposed by the protocol. Only

---

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

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 compute the transaction and check it against a list of rules only if the transaction fulfills these rules can the miner add it to the block. If only one transaction in a block did not fulfill the requirements, all other miners would reject the whole block. Second, the miner needs to solve the proof of stake puzzle by computing numerous hashes until one miner finds a block hash that fulfills the requirements for a new block. 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 solving a computationally expensive puzzle to decide who gets to write the next block but randomly assigns the privilege to write new blocks to miners according to their stakes tokens) will drastically decrease the computational efforts miners have to invest in finding a new block, it will not impact the effort miners have to invest in verifying every individual transaction. In essence, the update to PoS will even increase the relative importance of the effort required to verify 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 maintain decentralization by ensuring that miners with less powerful machines can also participate in mining transactions, the maximum gas of a block is limited (*block gas limit*). 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. To allocate the limited gas supply, Ethereum uses a market mechanism that we conceptualize as a market for transactions or, more specifically, a market for the verification and enforcement service of transactions.

The commodity sold on this market is the gas required to verify a transaction.[8] 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.[9]

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

[8] 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.
[9] See Appendix A for the formular used to compute the mining difficulty.

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 are the product of gas used and the gas price the user is willing to pay 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

In this section, we discuss the intuition that underlies our empirical analysis. 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 following theoretical arguments also apply to the period when Ethereum updated to PoS.[10] 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 verifying 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 verification

[10] Also our arguments should 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.

comparable to the one discussed above (e.g., Aztec Network, Binance Smart Chain, Optimism, Polygon).

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. 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, 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 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 are 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., get data from an oracle) or to purchase goods (e.g., buying NFTs). Considering that some NFTs are sold for well above $100,000,[11] 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 fees for a transaction with the dApp. Again, especially for dApps that rely on network effects, the number of other users of a dApp should increase the utility of transacting with this dApp.

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

Implications for the platform provider

The heterogeneity of complements is a decisive factor in a platform's success (Rietveld and Schilling 2020). Therefore, platform providers need to strategize on how many and what types of complements it wants to attract to join the platform to orchestrate an ecosystem of complements that create the most value for their users. However, in the case of blockchain platforms, the strategic toolset of platform providers to steer the transaction activity on the platform to attract or deter dApps is limited, as the "invisible" hand of the market determines the prioritization of transactions. Therefore, platform providers must understand and consider possible market dynamics already during the design of the market mechanism and carefully align it with the platform strategy. If not designed carefully, a market mechanism can lead to the discrimination of a certain type of complements, provoke them to leave the platform, and jeopardize the platform's long-term goals.

As we have elaborated above, some dApps might be more sensitive to changes in gas prices. Especially in times of high gas prices, for these dApps, it should be more difficult to attract users. If this decline in usage sustains for longer, the dApp might have to terminate its business and leave the platform. Consequently, characteristics associated with a higher sensitivity towards the gas price should also be associated with a higher likelihood of an exit and a lower likelihood of entry, particularly in times of high gas prices (also see supplementary survival analysis in Appendix C).

Although the exit of an unsuccessful dApp might be desirable for the platform provider and platform users if the exit is due to the bad quality of the dApp (e.g., it relies on inefficient smart contracts that require more gas than the smart contract of a competitor), it might be less desirable if the exit is due to the fact that the transaction verification mechanism discriminates against other characteristics of a dApp (e.g., type of service offered, or value carried by a transaction with a dApp) —especially if the platform aspires to become a general-purpose platform. To understand if there is undesirable discrimination in a market for transactions, next, we empirically investigate the drivers of a dApp's sensitivity we hypothesized above and discuss their implications.

5. Data and sample construction

We combine 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.

Data collection procedure and sample

We obtained our data from four different sources. First, we use the Ethereum ETL[12] 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).[13] 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 verification 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 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 consist

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

[13] 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).

of 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[14] 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 the *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.[15] 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.[16]

-------- insert Table 1 about here --------

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*. It refers to the daily amount of computational verification effort demanded by all transactions with a dApp. It is measured in Giga gas units. This variable operationalizes the goods supplied by the miners and demanded by the transaction senders.

---

[14] https://etherscan.io/apis
[15] https://www.coingecko.com/en/api/
[16] To mitigate multicollinearity issues arising from similar transaction patterns across similar categories, we aggregated the 17 categories into 5 groups that resemble in the type of service they offer. We obtained the groups by applying a cluster analysis to variables like daily transaction count and transaction value.

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 *market gas price* 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 blocks in whose verification miners circumvent the first-price auction mechanism by adding their own transactions with a gas price close to zero or even zero. Accordingly, using the marginal 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* 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:

$$difficulty\ bomb_d = (network\ hash\ rate_d\ x\ target\ blocktime) - difficulty_{\ observed,d}$$

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.

To account for the degree to which miners fill the blocks on a given day, we measure the *network utilization* 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* of a transaction with a dApp. To reflect the requirements of a transaction with a dApp, we measure the *average value of Ether or tokens* 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*, *average the number of unique externally owned accounts (EOA) that transactions with a dApp (i.e., our proxy for users)*,[17] the *average gas price users pay* for a transaction with a dApp, the *average number of transactions per externally owned account* on a given day, and the *surplus gas price* transaction senders pay beyond the market gas price on a given day.

We also control for the following network-level variables: *Ether price* measures the price of one Ether in USD on the day the transaction was executed; *Ether volatility* measures the daily change in the exchange rate of one Ether; *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* and *year* dummy variables, and a *trend*. Table 2 provides descriptive statistics and correlation scores for all variables in our data set.

---

[17] 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.

----- insert Table 2 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.

Baseline specification

The specification for our dApp-level analysis is:

$$\log(Gas\ used_{dt}) = \alpha_0 + \alpha_1 \log(Market\ gas\ price_t) + \alpha_2 Network\ utilization_t + \alpha_3 Network\ utilization_t^2 + \alpha_4 \log(Ether\ price_t) + \alpha_5 \log(Ether\ volatility_t) + \alpha_6 \log(Gas\ limit_t) + age_{dt} + \mu_d + \mu_{dayofweek} + \mu_{year} + u_t,$$

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.[18] In addition to these variables, we also control for the intrinsic growth of the dApp by adding age_{dt} as the number of days since the dApp entered the platform and specify μ_d as dApp fixed effects, $\mu_{dayofweek}$ as a day of week fixed effects, μ_{year} as a year fixed effects, and u_t as the error term.

---

[18] 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.

Validity of the instrument

In this model, log(*Gas used*$_t$) and log(*Market gas price*$_t$) are the endogenous variables, as both are jointly determined in equilibrium. To address this simultaneity issue, we use the *difficulty bomb* 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(*Market gas price*$_t$). In the second stage, we estimate the specification above by replacing the log(*Market gas price*$_t$) 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, changes to the difficulty 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 above), 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.

Baseline dApp-level results

Following our baseline specification, Table 3 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)).

----- insert Table 3 about here -----

Confirming our theoretical 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,[19] 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 (121.39) 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 (Stock and Yogo 2005) test for weak instruments, which shows that the Cragg-Donald-Wald F Statistic (2542.47) exceeds the predetermined critical value (16.38). Further, we compute the Kleibergen-Paap LM Statistic (70.04) 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.

To interpret the magnitude of the effect of the gas price (log(*gas price*)) on the demand of gas log(Gas used), the coefficient of -0.64 implies that a 1% increase in the market price of a unit of gas decreases the amount of gas demanded by 0.64%. 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 roughly 1,703 smart contract transactions per day or 14,923 Ether transfers, which require 21,000 units of gas. Considering that the median dApp only receives

[19] 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.

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 verification 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 (Katz and Shapiro 1985), 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 verification process dampens the effectiveness of same-side network effects.

Differing Demand Curves per Group

Column 3 in Table 3 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. 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 upwards sloping. An explanation for this upwards-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 affects these finance-related dApps could 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 downwards-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.

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. 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 4, 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 4 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 gas price and the average Ether value (0.14) and token value (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(Market gas price) x Average value or token value X group*), we only find that one out of 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 5 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 5 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 6 about here -----

To further investigate network effects, we analyze the impact of dynamic usage indicators that vary for each dApp over time. Table 6 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.

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 7 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 7 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. Conclusion

Decentralized blockchain platforms like Ethereum have been hailed for challenging the dominance of centralized digital platforms that currently prevail in the digital economy (Murray et al. 2019, Vergne 2020). Yet, little is known about how the decentralized transaction verification mechanism, which distinguishes blockchain platforms from their centralized counterparts, impacts the platform's performance by determining its usage and complements. To investigate this question, we study Ethereum's transaction verification mechanism as a market for transactions and use a panel data set of 1,590 dApps together with a novel supply-side instrument to estimate different price elasticities of the demand for transactions with dApps. We find strong evidence that Ethereum's gas price mechanism leads to negative network effects (i.e., a growth of the transaction demand makes transacting more expensive) that counteract the positive network effects usually present on multi-sided platforms. Further, we find that the relative magnitude of these effects depends on characteristics of a dApp that are mostly predetermined. Particularly, the type and complexity of the service a dApp offers are decisive factors. For instance, across the board, the demand for transactions with finance or exchange dApps seems to be less impacted by changes in the gas price than dApps that offer games, gambling, social, or media-related services. This is especially problematic as the transaction verification mechanism adds a new externality to the existing competition on such platforms: all dApps—no matter what service they offer—must compete for the limited gas supply. Hence, it favors some dApps over others and finally forces disadvantaged dApps to leave the platform leading to a decrease in the heterogeneity of dApps offered on Ethereum and a reduced value for platform users who joined because of the variety of complements offered on the platform.

The main contribution of this work is to unpack the consequences of using a market mechanism instead of a central authority to allocate transactions for the dApps offered on a blockchain platform. Our results have several important implications for the platform provider, the complementors, and policymakers. First, regarding platform providers: as we find that the type of service and its complexity determine a dApps sensitivity towards gas prices and thus their likelihood of entry or exit, platform providers have to consider these discriminatory effects when designing the transaction verification mechanism. Especially because the decentralized nature of blockchain platforms limits the strategic toolset that can be used to orchestrate complements, such as entry restrictions or other means of prioritization, a careful design of the transaction verification mechanism is warranted and has to align with the platform strategy. Carelessly expanding the complementor side on such a platform in the hope that it naturally benefits the platform's performance might be detrimental to the long-run goals of the platform. Our analysis provides a case in point, as it shows that the current version of Ethereum's gas price mechanism favors finance and exchange dApps over other dApps and thus contradicts Ethereum's vision of becoming a general-purpose platform that caters to all sorts of dApps. Further, it questions whether platforms with similar transaction verification mechanisms are viable options for Web 3.0.

Second, regarding complementors: in a market for transactions, platform complementors not only need to pay attention to their direct competition but also need to carefully analyze the current and future congestion of the network and consider their own sensitivity towards gas fees in comparison to all other dApps offered on the platform. Further, as our analysis shows that the gas requirement of a transaction with a dApp is another important determinant for its gas price elasticity, dApp providers need to consider how to bundle or split interactions with the dApp into one or multiple transactions.

Finally, regarding policymakers: policymakers and regulators are frequently concerned about antitrust competition between platforms. From this perspective, reduced heterogeneity of complements on one platform might be desirable as it gives rise to other platforms that are more closely tailored to the need of the complements and thus reduces the likelihood of one platform that dominates the whole industry. Although addressing the general impact of the transaction verification mechanism on the market via the creation of multiple other platforms is beyond the scope of this paper, our results need to be considered in the regulatory process. In particular, a transaction verification mechanism like in the case of Ethereum could be a self-regulation tool that mitigates the "winner-takes-it-all" typically associated with digital platforms that strongly rely on network effects.

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 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. 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 that are only accessible through Chinese 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 will be Ethereum’s long-announced update from PoW to PoS. Given that we predict that this update will only get rid of the computationally expensive puzzle of finding a hash that fulfills some properties required by the protocol but not the computation and verification of the transaction, the gas price mechanism should even become more important as it remains the most important driver of the costs of verifying transactions. Therefore, it would be interesting to see how validators prioritize transactions and influence the usage of dApps after the PoS update.

Figures

Figure 1: Daily gas used and gas price

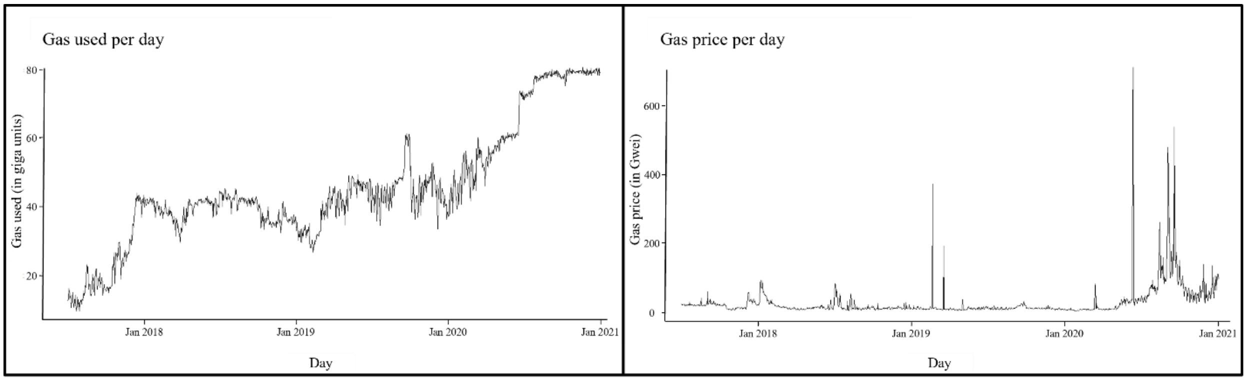
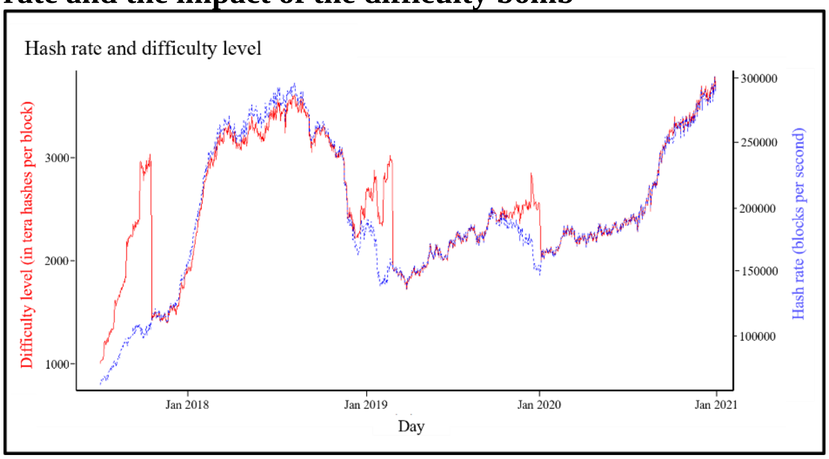


Figure 2: Hash rate and the impact of the difficulty bomb



Tables

Table 1: Groups of dApps

Groups	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	Dovu, Aaragon, CryptoTunes, XCloud	177

Table 2: Summary statistics and correlations (dApp level)

	N	Mean	SD	Min	Max	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)
(1) Gas used	370,748	180,178	1,288,645	21	85,346,148	1																		
(2) Transaction activity	370,748	893	9,213	1	518,357	0.89	1																	
(3) EOA	370,748	288	3,143	1	168,900	0.82	0.97	1																
(4) Average transaction gas price	370,748	28	44	0.00	6,250	0.05	0.05	0.05	1															
(5) Market gas price	370,748	8	14	1	54	0.06	0.06	0.06	0.71	1														
(6) Difficulty bomb	370,748	65	210	0.00	1,610	-0.01	-0.01	-0.01	-0.09	-0.13	1													
(7) Network utilization	370,748	301	195	84	1,385	0.02	0.02	0.02	0.25	0.23	-0.20	1												
(8) Network utilization^2	370,748	0.24	20	-228	153	0.01	0.01	0.01	0.01	0.06	0.001	0.06	1											
(9) log(Ether price)	370,748	0.85	0.10	0.30	0.98	0.03	0.04	0.04	0.37	0.54	-0.12	0.34	0.04	1										
(10) log(Ether volatility)	370,748	0.73	0.17	0.09	0.97	0.04	0.04	0.04	0.40	0.57	-0.12	0.36	0.04	1.00	1									
(11) Gas limit	370,748	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	370,748	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)Average gas requirement	370,748	322	478	21	9,900	0.04	-0.02	-0.02	-0.05	0.01	-0.02	-0.08	-0.001	0.01	0.01	0.05	-0.10	1						
(14) Average value sent USD	370,748	366	3,656	0.00	99,002	0.01	-0.001	-0.0002	0.01	0.01	-0.001	0.02	0.001	0.01	0.01	0.01	0.01	0.01	1					
(15) Average token value sent USD	370,748	2,781	13,909	0.00	185,968	0.05	0.04	0.03	0.02	0.01	0.01	0.02	0.003	-0.002	0.0004	0.001	0.07	-0.03	0.01	1				
(16) Average daily transactions	370,748	893	5,695	1.00	71,089	0.53	0.62	0.61	0.02	0.01	-0.004	0.002	-0.0004	0.01	0.01	0.01	0.05	-0.04	-0.002	0.06	1			
(17) Average daily EOA	370,748	288	1,954	1.00	24,975	0.50	0.61	0.62	0.02	0.01	-0.01	0.002	-0.001	0.01	0.01	0.01	0.05	-0.04	-0.0002	0.05	0.99	1		
(18) Average transactions per EOA	370,748	6	20	1.00	354	0.03	0.01	-0.01	0.01	0.04	-0.005	-0.02	0.01	0.02	0.02	0.05	-0.04	0.09	-0.01	-0.02	0.01	-0.02	1	
(19) Transactions per EOA	370,748	6	44	1.00	4,488	0.07	0.02	-0.01	0.002	0.01	0.01	-0.01	0.001	-0.01	-0.01	0.003	-0.04	0.04	-0.01	-0.01	0.01	-0.01	0.46	1
(20) Surplus gas price paid	370,748	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	0.002	-0.001

Table 3: Demand curve estimation – baseline model (dApp level)

	(1)	(2)	(3)
	log(Market gas price)	log(Gas used)	log(Gas used)
Difficulty bomb	0.20^{***} (0.0000)		
log(Market gas price)		-0.64^{***} (0.21)	0.27^{***} (0.05)
log(Ether price)	-0.0004 (0.01)	0.15^{***} (0.04)	0.18^{***} (0.04)
log(Ether volatility)	-0.01^{***} (0.0004)	0.01^{**} (0.004)	0.02^{***} (0.003)
Network utilization	-2.36^{***} (0.06)	-1.20^{**} (0.47)	0.30^{***} (0.11)
Network utilization^2	16.30^{***} (0.37)	8.59^{***} (3.29)	-1.89^{***} (0.68)
log(Gas limit)	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)
Year^{2018}	-0.82^{***} (0.02)	-0.68^{***} (0.22)	-0.09 (0.15)
Year^{2019}	-1.09^{***} (0.02)	-0.66^{***} (0.25)	0.07 (0.15)
Year^{2020}	-0.95^{***} (0.02)	-0.28 (0.24)	0.36^{**} (0.16)
weekday^{Thursday}	-0.02^{***} (0.001)	-0.03^{***} (0.01)	-0.01^{*} (0.01)
weekdays^{Friday}	0.02^{***} (0.001)	-0.02^{**} (0.01)	-0.03^{***} (0.01)
weekdays^{Wednesday}	-0.005^{***} (0.001)	-0.001 (0.01)	0.002 (0.01)
weekdays^{Monday}	-0.02^{***} (0.001)	-0.03^{***} (0.01)	-0.02^{**} (0.01)
weekdays^{Saturday}	0.01^{***} (0.002)	-0.07^{***} (0.01)	-0.08^{***} (0.01)
weekdays^{Sunday}	0.01^{***} (0.002)	-0.08^{***} (0.01)	-0.09^{***} (0.01)
log(Market gas price) × group 2			-0.43^{***} (0.15)
log(Market gas price) × group 3			-0.64^{***} (0.12)
log(Market gas price) × group 4			-0.49^{***} (0.10)
log(Market gas price) × group 5			-0.28^{***} (0.09)
Observations	370,392	370,392	370,392
R^2	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***

Note: Heteroskedastic and autocorrelation consistent (HAC) standard errors are shown in parentheses. Interacted and squared variables are centered before interacting or squaring them.

$^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$

Table 4: Interactions with transcation requirements (dApp level)

	(1)	(2)	(3)	(4)	(5)	(6)
	log(Gas used)	log(Gas used)	log(Gas used)	log(Gas used)	log(Gas used)	log(Gas used)
log(Market gas price)	-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(Market gas price) × log(Average gas requirement)	-0.06 (0.04)			0.02 (0.05)		
log(Market gas price) × log(Average value sent USD)		0.14*** (0.04)			0.15** (0.06)	
log(Market gas price) × log(Average token value sent USD)			0.31*** (0.04)			0.40*** (0.09)
log(Market gas price) × group 2				-0.17 (0.17)	-0.08 (0.18)	0.10 (0.20)
log(Market gas price) ×group 3				-0.28* (0.15)	-0.24 (0.15)	0.03 (0.16)
log(Market gas price) × group 4				-0.17 (0.14)	-0.15 (0.14)	0.09 (0.16)
log(Market gas price) × group 5				0.04 (0.14)	0.09 (0.14)	0.23 (0.16)
log(Market gas price) × log(Average gas requirement) × group 2				-0.58*** (0.16)		
log(Market gas price) × log(Average gas requirement) × group 3				-0.24** (0.11)		
log(Market gas price) × log(Average gas requirement) × group 4				-0.19* (0.10)		
log(Market gas price) × log(Average gas requirement) × group 5				-0.003 (0.08)		
log(Market gas price) × log(Average value sent USD) × group 2					-0.25 (0.17)	
log(Market gas price) × log(Average value sent USD) × group 3					0.18 (0.16)	
log(Market gas price) × log(Average value sent USD) × group 4					-0.02 (0.08)	
log(Market gas price) × log(Average value sent USD) × group 5					-0.10 (0.11)	
log(Market gas price) × log(Average token value sent USD) × group 2						-0.19 (0.15)
log(Market gas price) × log(Average token value sent USD) × group 3						-0.05 (0.13)
log(Market gas price) × log(Average token value sent USD) × group 4						-0.21 (0.14)
log(Market gas price) × log(Average token value sent USD) × group 5						-0.24** (0.11)
Controls	YES	YES	YES	YES	YES	YES
log(Ether volatility)	0.01* (0.004)	0.01** (0.004)	0.01* (0.004)	0.01* (0.004)	0.01* (0.004)	0.01* (0.004)
Network utilization	-1.24*** (0.47)	-1.18** (0.47)	-1.31*** (0.48)	-1.27*** (0.48)	-1.25*** (0.48)	-1.34*** (0.49)
Network utilization^2	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(Gas limit)	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

Note: Heteroskedastic and autocorrelation consistent (HAC) standard errors are shown in parentheses. Interacted and squared variables are centered before interacting or squaring them.

*p<0.1; **p<0.05; ***p<0.01

Table 5: Interactions with average performance indicators (dApp level)

	(1)	(2)	(3)	(4)	(5)	(6)
	log(Gas used)	log(Gas used)	log(Gas used)	log(Gas used)	log(Gas used)	log(Gas used)
log(Market gas price)	-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(Market gas price) × log(Average daily transactions)	0.16*** (0.06)			0.39*** (0.08)		
log(Market gas price) × log(Average daily EOA)		0.21*** (0.06)			0.39*** (0.07)	
log(Market gas price) × log(Average transactions per EOA)			-0.03 (0.04)			0.02 (0.06)
log(Market gas price) × group 2				0.08 (0.19)	0.06 (0.19)	-0.02 (0.15)
log(Market gas price) × group 3				-0.12 (0.15)	-0.13 (0.15)	-0.33** (0.15)
log(Market gas price) × group 4				0.01 (0.16)	0.02 (0.16)	-0.16 (0.14)
log(Market gas price) × group 5				0.22 (0.15)	0.22 (0.15)	0.06 (0.14)
log(Market gas price) × log(Average daily transactions) × group 2				-0.51*** (0.17)		
log(Market gas price) × log(Average daily transactions) × group 3				-0.64*** (0.14)		
log(Market gas price) × log(Average daily transactions) × group 4				-0.47*** (0.13)		
log(Market gas price) × log(Average daily transactions) × group 5				-0.45*** (0.11)		
log(Market gas price) × log(Average daily EOA) × group 2					-0.28* (0.16)	
log(Market gas price) × log(Average daily EOA) × group 3					-0.55*** (0.13)	
log(Market gas price) × log(Average daily EOA) × group 4					-0.38** (0.15)	
log(Market gas price) × log(Average daily EOA) × group 5					-0.46*** (0.11)	
log(Market gas price) × log(Average transactions per EOA) × group 2						-0.46*** (0.10)
log(Market gas price) × log(Average transactions per EOA) × group 3						-0.28** (0.12)
log(Market gas price) X log(Average transactions per EOA) × group 4						-0.12 (0.08)
log(Market gas price) × log(Average transactions per EOA) × group 5						0.03 (0.10)
log(Ether price)	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(Ether volatility)	0.01** (0.004)	0.01* (0.004)	0.01** (0.004)	0.01 (0.004)	0.01 (0.004)	0.01* (0.004)
Network utilization	-1.22** (0.48)	-1.25*** (0.48)	-1.21** (0.47)	-1.40*** (0.50)	-1.42*** (0.50)	-1.28*** (0.48)
Network utilization^2	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(Gas limit)	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

Note: Heteroskedastic and autocorrelation consistent (HAC) standard errors are shown in parentheses. Interacted and squared variables are centered before interacting or squaring them.

*p<0.1; **p<0.05; ***p<0.01

Table 6: Interactions with usage indicators (dApp level)

	(1) log(Gas used)	(2) log(Gas used)	(3) log(Gas used)	(4) log(Gas used)	(5) log(Gas used)	(6) log(Gas used)
log(Market gas price)	-0.44** (0.17)	-0.43** (0.17)	-0.38* (0.22)	-0.66*** (0.21)	-0.71*** (0.22)	-0.71** (0.29)
log(Transactions per EOA)	1.27*** (0.03)	1.28*** (0.04)	1.17*** (0.06)			
log(Market gas price) × log(Transactions per EOA)		0.08*** (0.03)	0.08** (0.04)			
log(Market gas price) × group 2			-0.15 (0.17)			-0.002 (0.19)
log(Market gas price) × group 3			-0.28** (0.13)			-0.15 (0.16)
log(Market gas price) × group 4			-0.21* (0.12)			-0.02 (0.16)
log(Market gas price) × group 5			0.05 (0.12)			0.12 (0.15)
log(Transactions per EOA) × group 2			-0.03 (0.15)			
log(Transactions per EOA) × group 3			0.35*** (0.08)			
log(Transactions per EOA) × group 4			0.01 (0.09)			
log(Transactions per EOA) × group 5			0.17 (0.11)			
log(Market gas price) × log(Transactions per EOA) × group 2			-0.13 (0.15)			
log(Market gas price) × log(Transactions per EOA) × group 3			-0.001 (0.07)			
log(Market gas price) X log(Transactions per EOA) × group 4			-0.02 (0.05)			
log(Market gas price) × log(Transactions per EOA) × group 5			0.16*** (0.06)			
log(Surplus gas price paid)				0.08*** (0.03)	-0.14*** (0.04)	-0.07 (0.07)
log(Surplus gas price paid) × log(Market gas price)					0.16*** (0.02)	0.16*** (0.03)
log(Surplus gas price paid) × group 2						-0.39*** (0.11)
log(Surplus gas price paid) × group 3						-0.35*** (0.11)
log(Surplus gas price paid) × group 4						-0.18 (0.11)
log(Surplus gas price paid) × group 5						0.14 (0.11)
log(Market gas price) × log(Surplus gas price paid) × group 2						0.11** (0.05)
log(Market gas price) × log(Surplus gas price paid) × group 3						0.09* (0.05)
log(Market gas price) × log(Surplus gas price paid) × group 4						-0.05 (0.05)
log(Market gas price) × log(Surplus gas price paid) × group 5						-0.18*** (0.05)
log(Ether price)	0.18*** (0.03)	0.18*** (0.03)	0.18*** (0.03)	0.14*** (0.04)	0.14*** (0.04)	0.13*** (0.04)
log(Ether volatility)	0.005 (0.003)	0.005 (0.003)	0.004 (0.003)	0.004 (0.005)	0.0003 (0.01)	-0.001 (0.01)
Network utilization	-0.73* (0.38)	-0.72* (0.38)	-0.82** (0.39)	-1.15** (0.46)	-1.41*** (0.48)	-1.48*** (0.49)
Network utilization^2	5.45** (2.66)	5.38** (2.67)	6.07** (2.72)	8.23** (3.20)	10.20*** (3.38)	10.67*** (3.44)
log(Gas limit)	1.39*** (0.45)	1.39*** (0.45)	1.47*** (0.46)	1.71*** (0.48)	1.85*** (0.49)	1.88*** (0.50)
Age	-0.001*** (0.0002)	-0.001*** (0.0002)	-0.001*** (0.0002)	-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

Note: Heteroskedastic and autocorrelation consistent (HAC) standard errors are shown in parentheses. Interacted and squared variables are centered.

*p<0.1; **p<0.05; ***p<0.01

Table 7: Robustness checks

	(1)	(2)	(3)	(4)	(5)	(6)	(7)
	Baseline	Alternative Dependent variable	Alternative market gas price (25th percentile)	Alternative market gas price (average gas price)	Alternative instrument (block difference)	Outliers (5th-95th percentile gas used)	Subsample (specific difficulty bomb period)
	log(Gas used)	log(Transaction count)	log(Gas used)	log(Gas used)	log(Gas used)	log(Gas used)	log(Gas used)
log(Market gas price)	-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

Note: Heteroskedastic and autocorrelation consistent (HAC) standard errors are shown in parentheses.

*p<0.1; **p<0.05; ***p<0.01

References

Agarwal R, Gort M (2002) Firm and Product Life Cycles and Firm Survival. *AER*. 92(2):184–190.

Antonopoulos AM, Wood GA (2019) *Mastering Ethereum. Building smart contracts and DApps*. EBL-Schweitzer (O'Reilly, Beijing, Boston, Farnham, Sebastopol, Tokyo).

Arnosti N, Weinberg SM (2018) Bitcoin: A Natural Oligopoly. Forthcoming.

Azevedo Sousa JE, 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. *Int J Network Mgmt*. 31(3).

Basu S, Easley D, O'Hara M, Sirer EG (2019) Towards a Functional Fee Market for Cryptocurrencies. Forthcoming.

Boudreau KJ (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.

Brynjolfsson E, Kemerer CF (1996) Network Externalities in Microcomputer Software: An Econometric Analysis of the Spreadsheet Market. *Management Science*. 42(12):1627–1647.

Buterin V (2014) Ethereum Whitepaper, https://ethereum.org/en/whitepaper/.

Casadesus-Masanell R, Hałaburda H (2014) When Does a Platform Create Value by Limiting Choice? *Journal of Economics & Management Strategy*. 23(2):259–293.

Catalini C, Tucker CE (2018) Antitrust and Costless Verification: An Optimistic and a Pessimistic View of the Implications of Blockchain Technology. *SSRN*:1–14.

Chen Y, Richter JI, Patel PC (2021) Decentralized Governance of Digital Platforms. *Journal of Management*. 47(5):1305–1337.

Choi JP (1994) Network Externality, Compatibility Choice, and Planned Obsolescence. *The Journal of Industrial Economics*. 42(2):167.

Choi JP, Kim B-C (2010) Net Neutrality and Investment Incentives. *RAND Journal of Economics*. 41(3):446–471.

Church J, Gandal N (1992) Network Effects, Software Provision, and Standardization. *The Journal of Industrial Economics*. 40(1):85.

Cong L, Tang K, Wang Y, Zhao X (2022) Inclusion and Democratization Through Web3 and DeFi? Initial Evidence from the Ethereum Ecosystem. *SSRN Journal*. Forthcoming.

Cong LW, He Z, Li J (2021) Decentralized Mining in Centralized Pools. *The Review of Financial Studies*. 34(3):1191–1235.

Cox DR (1972) Regression Models and Life-Tables. *Journal of the Royal Statistical Society Series B (Methodological)*. 34(2):187–220.

Cusumano MA, Mylonadis Y, Rosenbloom RS (1992) Strategic Maneuvering and Mass-Market Dynamics: The Triumph of VHS over Beta. *Bus. Hist. Rev.* 66(1):51–94.

Donmez A, Karaivanov A (2021) Transaction fee economics in the Ethereum blockchain. *Economic Inquiry*. 60:265–292.

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.

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)*. Forthcoming.

Farrell J, Saloner G (1986) Installed Base and Compatibility: Innovation, Product Preannouncements, and Predation. *The American Economic Review*. 76(5):940–955.

Foley S, Karlsen JR, Putniņš TJ (2019) Sex, Drugs, and Bitcoin: How Much Illegal Activity Is Financed through Cryptocurrencies? *The Review of Financial Studies*. 32(5):1798–1853.

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*, Lecture Notes in Computer Science (Springer International Publishing, Cham), 357–372.

Gale D (1955) The Law of Supply and Demand,, Vol. 3. *Mathematica Scandinavica*. 3(1):155–169.

Gandal N (1994) Hedonic Price Indexes for Spreadsheets and an Empirical Test for Network Externalities. *The RAND J of Economics*. 25(1):160.

Ghazawneh A, Henfridsson O (2013) Balancing platform control and external contribution in third-party development: the boundary resources model. *Info Systems J*. 23(2):173–192.

Halaburda H, Haeringer G, Gans JS, Gandal N (2020) The Microeconomics of Cryptocurrencies. *NBER Working Paper*. No. 27477.

Halaburda H, Levina N, Min S (2019) Understanding Smart Contracts as a New Option in Transaction Cost Economics. *ICIS*. Forthcoming.

Houy N (2016) The Bitcoin Mining Game. *ledger*. 1:53–68.

Huang P, Ceccagnoli M, Forman C, Wu DJ (2013) Appropriability Mechanisms and the Platform Partnership Decision: Evidence from Enterprise Software. *Management Science*. 59(1):102–121.

Huberman G, Leshno J, Moallemi CC (2017) Monopoly Without a Monopolist: An Economic Analysis of the Bitcoin Payment System. *SSRN Journal*. Forthcoming.

Ilk N, Shang G, Fan S, Zhao JL (2021) Stability of Transaction Fees in Bitcoin: A Supply and Demand Perspective. *MISQ*. 45(2):563–692.

Katz ML, Shapiro C (1985) Network Externalities, Competition, and Compatibility. *AER*. (75):424–440.

Kroll J, Davey I, Felten Edward (2013) The economics of bitcoin mining, or bitcoin in the. *The Twelfth Workshop on the Economics of Information Security (WEIS 2013) Washington, DC*:1–21.

Lavi R, Sattath O, Zohar A (2017) Redesigning Bitcoin's fee market. Forthcoming.

Leiponen A, Thomas LDW, Wang Q (2021) The dApp economy: a new platform for distributed innovation? *Innovation*:1–19.

Li T, Shin D, Wang B (2018) Cryptocurrency Pump-and-Dump Schemes. *SSRN Electronic Journal*. Forthcoming.

Lin S (2020) Two-Sided Price Discrimination by Media Platforms. *Marketing Science*. 39(2):317–338.

Liu Q, Serfes K (2013) Price Discrimination in Two-Sided Markets. *JEMS*. 22(4):768–786.

Markovich S, Moenius J (2009) Winning while losing: Competition dynamics in the presence of indirect network effects. *International Journal of Industrial Organization*. 27(3):346–357.

Murray A, Kuban S, Josefy M, Anderson J (2019) Contracting in the Smart Era: The Implications of Blockchain and Decentralized Autonomous Organizations for Contracting and Corporate Governance. *AMP*. Forthcoming.

Nakamoto S (2008) Bitcoin: A Peer-to-Peer Electronic Cash System, https://bitcoin.org/bitcoin.pdf.

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

O'Mahony S, Karp R (2020) From proprietary to collective governance: How do platform participation strategies evolve? *SMJ*. 43(3):530–562.

Panico C, Cennamo C (2020) User preferences and strategic interactions in platform ecosystems. *Strategic Management Journal*. 43(3):507–529.

Parker G, van Alstyne M (2017) Platform Ecosystems: How Developers Invert the Firm. *MIS Quarterly*. 41(1):255–266.

Parker G, van Alstyne M (2018) Innovation, Openness, and Platform Control. *Management Sci.* 64(7):3015–3032.

Pereira J, Tavalaei MM, Ozalp H (2019) Blockchain-based platforms: Decentralized infrastructures and its boundary conditions. *Technological Forecasting and Social Change*. 146:94–102.

Rietveld J, Schilling M (2020) Platform Competition: A Systematic and Interdisciplinary Review of the Literature. *JOM*:(in press).

Riggins FJ, Kriebel CH, Mukhopadhyay T (1994) The Growth of Interorganizational Systems in the Presence of Network Externalities. *Management Science*. 40(8):984–998.

Roughgarden T (2020) Transaction Fee Mechanism Design for the Ethereum Blockchain: An Economic Analysis of EIP-1559. Forthcoming.

Sapirshtein A, Sompolinsky Y, Zohar A (2016) Optimal Selfish Mining Strategies in Bitcoin. Forthcoming.

Shapiro C, Varian HR (2010) *Information rules. A strategic guide to the network economy* (Harvard Business School Press, Boston, Mass.).

Spain M, Foley S, Gramoli V (2020) The Impact of Ethereum Throughput and Fees on Transaction Latency During ICOs.

Stock JH, Yogo M (2005) Testing for Weak Instruments in Linear IV Regression. Stock JH, Andrews DWK, eds. *Identification and inference for econometric models. Essays in honor of Thomas Rothenberg* (Cambridge university press, New York), 80–108.

Tiwana A, Konsynski B, Bush AA (2010) Research Commentary —Platform Evolution: Coevolution of Platform Architecture, Governance, and Environmental Dynamics. *Information Systems Research*. 21(4):675–687.

Tudón J (2022) Prioritization vs. congestion on platforms: evidence from Amazon's Twitch.tv. *The RAND J of Economics*. 53(2):328–355.

Vergne JP (2020) Decentralized vs. Distributed Organization: Blockchain, Machine Learning and the Future of the Digital Platform. *Organ. Theory*. 1(4):1-26.

Wang Z, Wright J (2017) Ad valorem platform fees, indirect taxes, and efficient price discrimination. *The RAND J of Economics*. 48(2):467–484.

Wood GA (2014a) ÐApps: What Web 3.0 Looks Like, http://gavwood.com/dappsweb3.html.

Wood GA (2014b) *Ethereum: A secure decentralised generalised transaction ledger*.

Wu K, Ma Y, Huang G, Liu X (2021) A first look at blockchain-based decentralized applications. *Softw: Pract Exper*. 51(10):2033–2050.

Appendix A – Additional formulas

Block time

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

$$block\ time_b = \frac{mining\ difficulty_b}{network\ hash\ rate_{b-1}}$$

Where $mining\ difficulty_b$ is the average number of hashes it requires to find a new block and $network\ hash\ rate_{b-1}$ 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:

$$mining\ reward_b = 2 + \sum_{\forall t \epsilon b} \frac{gas\ price_t \times gas\ used_t}{10^9}$$

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:

$$transaction\ fees_t = \frac{gas\ price_t\ x\ gas\ used_t}{10^9}$$

Appendix B – Network-level analysis

In addition to the dApp-level analysis, we created a second data set that aggregates all network transactions. This additional analysis allows us to estimate network-level demand curves (i.e., one demand curve for all transactions), compare the demand curves between Ether transfers between two externally owned accounts and dApp transactions, and estimate a separate demand curve for every group of dApps by filtering only transactions to a specific group of dApps. Further, it ensures comparability with other studies that conduct their analysis only on the network level (e.g., Donmez and Karaivanov 2021, Ilk et al. 2021). The variables used in this analysis are analogous to the dApp level data set. Table 8 depicts summary statistics and correlations of this dataset.

Table 8: Summary statistics and correlations (network level)

Variables	N	Mean	S.D.	1	2	3	4	5	6	7	8	9	10	11	12
1. Gas used	1,280	45.42	17.15	1											
2. Gas used group 1	1,280	18.96	18.65	0.88	1										
3. Gas used group 2	1,280	0.39	0.66	-0.50	-0.28	1									
4. Gas used group 3	1,280	2.43	1.77	-0.04	-0.25	-0.23	1								
5. Gas used group 4	1,280	0.86	0.61	-0.09	-0.27	-0.12	0.46	1							
6. Gas used group 5	1,280	0.56	0.53	-0.21	-0.20	0.09	-0.14	-0.42	1						
7. Market gas price	1,280	6.75	12.29	0.73	0.86	-0.16	-0.33	-0.33	-0.15	1					
8. Difficulty bomb	1,280	1.08	2.92	-0.48	-0.23	0.25	-0.25	-0.06	-0.05	-0.12	1				
9. Network utilization	1,280	0.83	0.13	0.73	0.53	-0.60	0.01	-0.20	0.03	0.45	-0.18	1			
10. Ether price	1,280	327.48	218.96	0.10	0.11	-0.04	-0.19	-0.62	0.64	0.13	-0.16	0.27	1		
11. Ether volatility	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. Gas limit	1,280	0.01	0.002	0.93	0.90	-0.41	-0.08	-0.02	-0.29	0.75	-0.31	0.53	0.001	0.03	1

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

$$\log(Gas\ used_t) = \alpha_0 + \alpha_1 \log(Market\ gas\ price_t) + \alpha_2 Network\ utilization_t + \alpha_3 Network\ utilization_t^2 + \alpha_4 \log(Ether\ price_t) + \alpha_5 \log(Ether\ volatility_t) + \alpha_6 \log(Gas\ limit_t) + \mu_{dayofweek} + \mu_{year} + trend + u_t,$$

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), $\mu_{dayofweek}$ denotes the day of week fixed effects,

μ_{year} the year fixed 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.[20]

Baseline network-level results

Following the network-level specification, Table 9 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.

[20] 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.

Table 9: 2SLS model with 1st and 2nd stage and OLS benchmark (network level)

	(1)	(2)	(3)
	2SLS *1st stage*	2SLS *2nd stage*	OLS
	log(Gas price)	log(Gas used)	log(Gas used)
Difficulty bomb	0.10^{***} (0.02)		
log(Market gas price)		-0.69^{***} (0.16)	-0.04^{**} (0.02)
Network utilization	-3.03^{***} (0.35)	-1.58^{***} (0.43)	0.20 (0.19)
Network utilization^2	17.51^{***} (1.85)	10.38^{***} (2.60)	-0.33 (0.87)
log(Ether price)	0.09 (0.13)	0.06 (0.08)	0.12^{**} (0.05)
log(Ether volatility	-0.02 (0.02)	-0.01 (0.01)	0.001 (0.003)
log(Gas limit)	3.08^{***} (1.11)	3.02^{***} (0.99)	0.53^{*} (0.28)
$D^{Thursday}$	-0.04 (0.03)	-0.03 (0.02)	-0.001 (0.002)
D^{Friday}	0.01 (0.03)	0.005 (0.02)	-0.001 (0.003)
$D^{Wednesday}$	-0.02 (0.02)	-0.01 (0.02)	0.0002 (0.002)
D^{Monday}	-0.05 (0.03)	-0.03 (0.02)	-0.01^{*} (0.004)
$D^{Saturday}$	-0.02 (0.04)	-0.01 (0.02)	-0.01 (0.01)
D^{Sunday}	-0.03 (0.04)	-0.02 (0.02)	-0.01 (0.01)
D^{2018}	-1.21^{***} (0.20)	-0.85^{***} (0.26)	0.13 (0.19)
D^{2019}	-1.61^{***} (0.29)	-1.11^{***} (0.30)	-0.005 (0.24)
D^{2020}	-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
R^2	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). $^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$

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 10 reports the results of these robustness checks.

Table 10: Robustness checks

	(1)	(2)	(3)	(4)	(5)	(5)	(6)	(7)
	Baseline	Alternative Dependent variable	Alternative market gas price (25th percentile)	Alternative market gas price (average gas price)	Alternative market gas price (normalized by ETH supply)	Alternative instrument (block difference)	Subsample (5th-95th percentile gas used)	Subsample (specific difficulty bomb period)
	log(Gas used)	log(Transaction count)	log(Gas used)	log(Gas used)	log(Gas used)	log(Gas used)	log(Gas used)	log(Gas used)
log(Market gas price)	-0.69*** (0.16)	-0.63*** (0.15)	-0.80*** (0.20)	-1.83** (0.61)	-0.57 (0.14)	-0.75** (0.24)	-0.69** (0.19)	-2.70 (2.95)
Observations	1,279	1,279	1,279	1,279	1,279	1,279	1,279	101

Note: Heteroskedastic and autocorrelation consistent (HAC) standard errors are shown in parentheses. *p<0.1; **p<0.05; ***p<0.01

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 11 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.

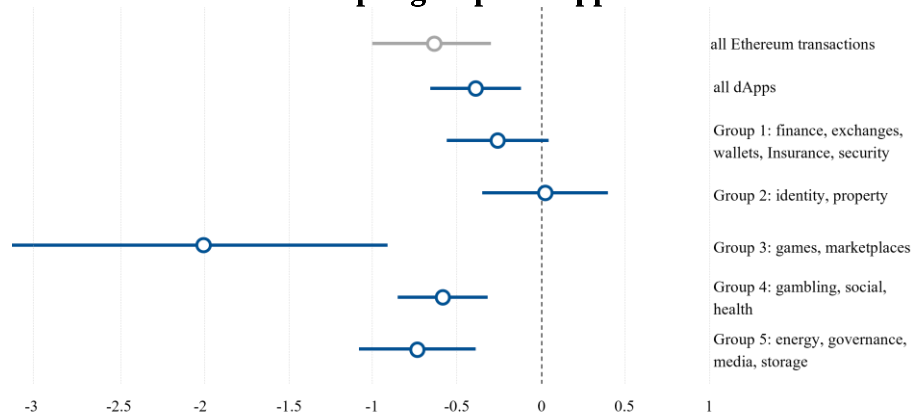
Table 11: 2SLS models by group

	(1)	(2)	(3)	(4)	(5)	(6)
	2SLS 2nd stage	2SLS *2nd stage*	2SLS *2nd stage*	2SLS *2nd stage*	2SLS *2nd stage*	2SLS *2nd stage*
	log(Gas used by all dApps)	log(Gas used by group 1)	log(Gas used by group 2)	log(Gas used by group 3)	log(Gas used by group 4)	log(Gas used by group 5)
log(Market gas price)	-0.45*** (0.14)	-0.29* (0.16)	0.09 (0.19)	-2.09*** (0.63)	-0.59*** (0.13)	-0.48*** (0.17)
Network utilization	-1.04*** (0.36)	-0.27 (0.41)	-0.84 (0.61)	-2.37 (1.67)	-0.91* (0.48)	-1.05** (0.51)
Network utilization^2	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(Ether price)	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(Ether volatility)	-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(Gas limit)	2.49*** (0.92)	1.56 (1.05)	-0.75 (1.07)	7.61*** (2.28)	1.88** (0.86)	2.68*** (0.91)
$D^{Thursday}$	-0.03 (0.02)	-0.02 (0.02)	0.02 (0.04)	-0.12 (0.08)	-0.05* (0.03)	-0.09** (0.04)
D^{Friday}	0.01 (0.02)	0.01 (0.02)	-0.04 (0.04)	0.03 (0.07)	-0.02 (0.03)	-0.13*** (0.04)
$D^{Wednesday}$	-0.002 (0.02)	0.004 (0.01)	-0.02 (0.03)	-0.06 (0.05)	-0.03 (0.02)	-0.06* (0.04)
D^{Monday}	-0.02 (0.02)	-0.01 (0.02)	-0.03 (0.04)	-0.10 (0.07)	-0.06** (0.03)	-0.12*** (0.03)
$D^{Saturday}$	-0.04 (0.03)	-0.07*** (0.03)	-0.09** (0.04)	0.13* (0.07)	-0.06* (0.03)	-0.13*** (0.05)
D^{Sunday}	-0.04 (0.02)	-0.08*** (0.02)	-0.08* (0.05)	0.14* (0.07)	-0.07** (0.03)	-0.13*** (0.05)
D^{2018}	-1.25*** (0.28)	-1.36*** (0.35)	-0.26 (0.31)	-1.29 (1.15)	-0.66** (0.28)	-0.23 (0.30)
D^{2019}	-1.53*** (0.32)	-1.80*** (0.40)	-0.23 (0.38)	-1.69 (1.43)	-0.41 (0.35)	0.22 (0.38)
D^{2020}	-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	1,279	1,279	1,279	1,279	1,279
C-D Wald F Stat.			85.06			
Stock-Yogo Critical Value			16.38			
Kleibergen-Paap LM Stat.			4.19**			

Note: 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. *p<0.1; **p<0.05; ***p<0.01

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 (identify and property dApps) can explain why users are relatively insensitive to changes in the gas price.

Figure 3: Price elasticities of demand per group of dApps



Appendix C – Supplementary survival analysis

To investigate the impact of Ethereum's transaction verification 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 12 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.

Table 12: Survival models

	(1) all dApps stratified by group	(2) all dApps stratified by group
log(Market gas price)	0.02 (0.09)	-1.7* (0.11)
log(Market gas price) × group 2		0.49** (0.23)
log(Market gas price) × group 3		0.15 (0.10)
log(Market gas price) × group 4		0.21** (0.09)
log(Market gas price) × group 5		0.22* (0.12)
Network utilization	-6.68 (8.24)	-6.89 (8.18)
Network utilization^2	4.01 (5.32)	4.15 (5.28)
log(Ether price)	-0.04 (0.14)	-0.02 (0.14)
log(Ether volatility)	0.01 (0.04)	0.01 (0.04)
log(Gas limit)	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	-2,088.394	-2,083.793

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.

$^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$

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 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.