2018 IEEE 4th International Conference on Collaboration and Internet Computing

A Decentralized Marketplace Application on The Ethereum Blockchain

R. Vishnu Prasad
Department of Computer Science
University of North Texas
Denton, USA
vishnupr4296@gmail.com

Ram Dantu
Department of Computer Science
University of North Texas
Denton, USA
ram.dantu@unt.edu

Aditya Paul
Department of Computer Science
University of North Texas
Denton, USA
aditya.paul@my.unt.edu

Paula Mears
Department of Computer Science
University of North Texas
Denton, USA
paula.mears@untsystem.edu

Kirill Morozov
Department of Computer Science
University of North Texas
Denton, USA
kirill.morozov@unt.edu

***Abstract*—Modern centralized online marketplaces such as eBay offer an alternative option for consumers to both sell and purchase goods with relative ease. However, drawbacks to these marketplaces include the platform's ability to block merchants at their own whim, the fees paid to the platform when listing a product and when selling a product, and the lack of privacy of users' data. In this paper, we propose an application that remedies all three of these drawbacks through use of the Ethereum blockchain platform. The application was developed using the Truffle development framework. The application's functions were contained within an Ethereum smart contract, which was then migrated to the Ethereum network. The user's input was read through a web interface and sent to the Ethereum network via the web3.js API. Statistics about the application were gathered on the Rinkeby test network. The application was shown to have an average transaction runtime of 3.8 seconds, and an average gas consumption of 4.6 wei. Contract creation times for the application were shown to be less than a second. A cost analysis of the application was then conducted. The gas consumption of the transactions needed to both buy and sell a product was converted into US dollars, and the gas cost of the application was then compared to the cost to use an online auction marketplace such as eBay as well as an in-person auction house such as Sotheby's. The results showed that selling on the application is cheaper than existing online options as well as existing in-person options. These tests showed that our application was successful in addressing the drawbacks of current auction marketplaces.**

***Keywords*—blockchain, Ethereum, marketplace, smart contract, decentralized, cryptocurrency, e-commerce**

I. Introduction

The potential that blockchain technology has to disrupt existing markets has been gaining recognition in the last few years [1], fueled by the entrance of Bitcoin [2] into the mainstream consumer space. Aside from its cryptocurrency applications, blockchains can also be used as a platform to build truly decentralized applications with no central point of failure and no hierarchical ownership of user data. This has significant advantages for data security, privacy, and ownership, as well as the potential to dramatically reduce middle-man costs. The issue of user trust in decentralized applications that run on a blockchain platform is one that has been studied [3] and developed on recently, and it is now possible to build application that the user can trust with their money.

These developments have potentially significant applications in the field of online marketplaces. Traditional marketplaces charge a fee for their services which hurts both buyers and sellers. A decentralized application on a blockchain would reduce costs associated with using the platform while giving users more control of their data and information. As centralized marketplaces have the ability to compromise and sell user data, blockchain is a suitable platform to create such a marketplace on. The desire for a system with no central point of failure can be seen in the eBay data breach in 2014, in which 145 million users were at risk of their data being stolen [19]. However, in a decentralized system, it is possible to have each user safeguard their own data.

The ideal platform to develop such an application is on the Ethereum blockchain, as it has been designed from the ground-up to provide developers with tools to build decentralized applications on its blockchain [4]. Ethereum's easy-to-use framework and its use of ether as a standard currency that can be used in applications makes it relatively intuitive to create marketplace and e-commerce solutions.

Ethereum is the second largest cryptocurrency by market-cap and has extensive documentation and an active developer community. A number of decentralized applications have already been built on the network, with the popular CryptoKitties application showing the potential for mainstream success [5].

978-1-5386-9502-9/18/$31.00 ©2018 IEEE
DOI 10.1109/CIC.2018.00023

Ethereum provides developers the ability to use "smart contracts" which are code that executes when an event is triggered. These events are then interpreted by the Ethereum web3.js API and used to manipulate a webpage. Users interact with the Ethereum network through Metamask, a Chrome extension that is able to interface with an Ethereum wallet. A user with a cryptocurrency wallet and the Metamask extension on their browser can interact with the app and buy or sell items. This makes programming on the blockchain a realistic possibility. As decentralized apps enter the mainstream, browsers can be expected to provide additional support by default to these apps, further increasing their visibility to the general population.

II. Problem Definition

There are several problems with existing centralized marketplaces:

1) The sellers are at the mercy of the company: due to their central position in the application infrastructure, the company can decide to block the merchant from transacting on their platform at their own discretion any time which decreases job security.
2) Sellers pay a fee to list their product and also pay commission on sales. This reduces profit margins or leads to increased prices for the buyer.
3) Platform users do not own any of their data. Reviews, purchase history and personal information are all owned by these companies. If a merchant wanted to move their operation to another provider, it may not be possible, as the platform may still own the previous user's data.

Because all of these problems include a centralized service that manages data as well as transactions and has power over both buyers and sellers, a decentralized system like blockchain can replace this middleman and fully automate this service, and in turn, drive down prices and protect user data.

The goal of this application is to build a decentralized marketplace infrastructure using smart contracts on the Ethereum blockchain that would solve all these problems. The application will allow a seller to list an item, multiple users to bid within a given time and the highest bidder to get the item. Once the buyer is satisfied, the bid amount is transferred to the seller's digital wallet. This framework can be specialized in the future for a specific market. The advantage over the traditional model is the lack of a middle man making profits and central points of failure. For example, a middle-man cannot block merchants from selling goods on the platform, and each user keeps their data to themselves, as ensured by the Ethereum framework.

III. Literature and Competition Survey

A. Literature Survey

Trust is a key component in the success of online marketplaces. Buyers are wary of what they perceive to be opportunism by marketplace intermediaries [22]. Risk perception was found to be an equally important consideration by buyers in the online marketplace [23].

Bitcoin was proposed as a system for conducting transactions without the need for trust. Nakamoto describes the underlying technology of bitcoin as a network of peer-to-peer nodes building the chain of transparent transactions through cryptographic hashes incorporating the previous block on the chain [2]. Bitcoin has ushered in the adoption of the distributed general ledger for a variety of applications. Ethereum was developed to take advantage of the Bitcoin framework for promoting trust-based transactions between end users. Gavin Wood, the founder of Ethereum and Ethcore, states that the purpose of the Ethereum project was to facilitate transactions between individual with no prior trust mechanisms. His goal was to provide a system that provides interaction of absolute confidence in the outcomes [4].

Pongnumkul et al. compared the performance of Ethereum with Hyperledger, an open-source blockchain framework hosted by the Linux Foundation. The performance parameters they measured included: I) Transaction deployment time, II) Transaction completion time, III) Execution Time IV) Latency V) Throughput and VI) Maximum Concurrent Transactions. The findings indicated that Hyperledger had a higher throughput but Ethereum could handle more concurrent transactions. This makes Ethereum the more suitable platform for developing scalable applications. Ethereum's use of its own cryptocurrency in applications using its framework enables developers to create e-commerce applications using the Ethereum cryptocurrency, ether [6].

Toneli et al. studied the characteristics of over 12,000 smart contracts to identify software metrics to find that generally smart contracts metrics have ranges more restricted than the corresponding metrics in traditional software systems. Software metrics they studied include: Number of events, mapping, modifier, contracts, functions, lines, comments, bytecode and cyclomatic complexity. These metrics are common to all software and can be measured for non-decentralized applications as well, but those metrics unique to smart contracts include: calls to or from other addresses, intra-smart contract calls, gas consumption, cryptocurrency exchanges, and bytecode/ABI metrics [7].

Researchers at Vanderbilt University addressing a healthcare specific application, list more qualitative measures such as: structural interoperability, support of Turing-complete operations, support for user

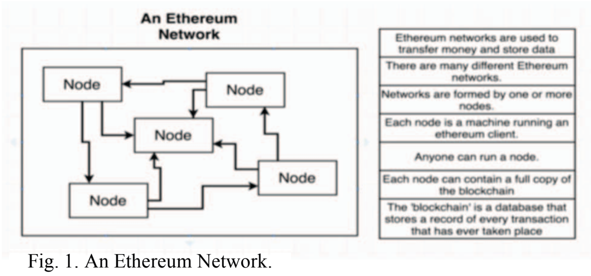


Fig. 1. An Ethereum Network.

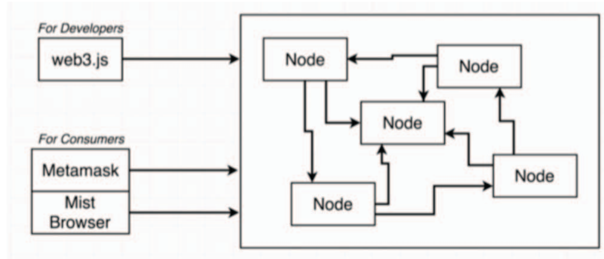


Fig. 2. Interfacing with an Ethereum Network.

identification and authentication, cost-effectiveness, and scalability across large populations [8].

Others suggested security patterns for newly built smart contracts comparing measures including: checks-effects-interaction, emergency stop, speed bump rate, limit mutex, and balance limit [9].

B. *Existing Market Analysis*

Ten of the most popular startups in the field were analyzed. The idea of a decentralized marketplace founded in a blockchain (which could potentially disrupt large corporations) has become popular only very recently: over the last 6-8 months. While all of them have landing pages and whitepapers, the majority do not have a usable product, with all 10 being in very early stages of development (7 out of 10 started work in 2017).

This is currently the cutting edge of blockchain development, and there is no clear leader in the market. The idea for this project is to build a general marketplace.

Metrics are ranked qualitatively by: Very Bad, Bad, Average, Good, Very Good and Unknown The analysis of existing decentralized marketplaces:

- OpenBazaar: Oldest software in the field (started 2015), it is an open source electron app that stores data on IPFS (a protocol to store files across a network). Not truly a blockchain platform aside from accepting cryptocurrencies and not based on the rapidly expanding Ethereum. Metrics: Speed(Bad), Cost(Average), Security(Average), Reliability(Bad), Ease of Use(Good), Traffic Load Capacity(Unknown)
- KioskProtocol: In development since October 2017, white paper out but no product. Metrics: None
- UHUB.IO: Live ICO, no working product (expected later in 2018). Metrics: None
- BitBoost(TheBlock): In development, no final product yet (expected Q3 2018). Metrics: None
- Hamster: Working application that is B2C. Not peer-to-peer, hence has centralization and related issues. Metrics: Speed(Good), Cost(Bad), Security(Unknown), Reliability(Bad), Ease of Use(Bad), Traffic Load Capacity(Unknown)
- Syscoin: Working application but uses custom currency that is not linked to any major cryptocurrency, hence volatile and risky for the near future. Metrics: Speed(Good), Cost (Very Bad),
  Security (Very Bad), Reliability(Bad), Ease of Use(Good), Traffic Load Capacity(Unknown)
- Storiqa: In progress. Only landing page deployed. Based on Ethereum smart contracts. Already raised $25 million with product supposed to come later this year. Metrics: None
- $hop: At ICO stage. Metrics: None
- Blocklancer: In alpha stage. Metrics: None
- Coinlancer: UI released only few weeks back; very early stage of development. Metrics: None

IV. ARCHITECTURE

The generalized structure of an Ethereum network is provided in Fig. 1. An interface diagram is represented in Fig. 2. A general marketplace is presented in Fig. 3. and an architecture diagram specific to the application being developed is represented in Fig. 4.

An Ethereum network is composed of a set of nodes running an Ethereum client (Fig. 1). Each of these nodes have a copy of the blockchain, which contains a list of all operations performed on the network. This enables nodes to prevent fraudulent activities, such as counterfeiting and duplicating cryptocurrencies, as well as containing an auditable record of all transactions performed on the network. Due to the decentralized nature of the network, the Ethereum framework preserves user pseudonymity (a weaker form of anonymity), as each user's identity is given by a public key. This enables users on the platform to perform functions such as transferring money, buying and selling, and much more.

Ethereum provides an interface with the Ethereum network for developers in the web3.js API (Fig. 2). This allows web apps to interpret events sent by the Ethereum network and submit transactions to the network. Ethereum also provides an interface for consumers through blockchain-enabled extensions and browsers such as Metamask for Google Chrome and the Mist browser. These interfaces provide a consumer access to

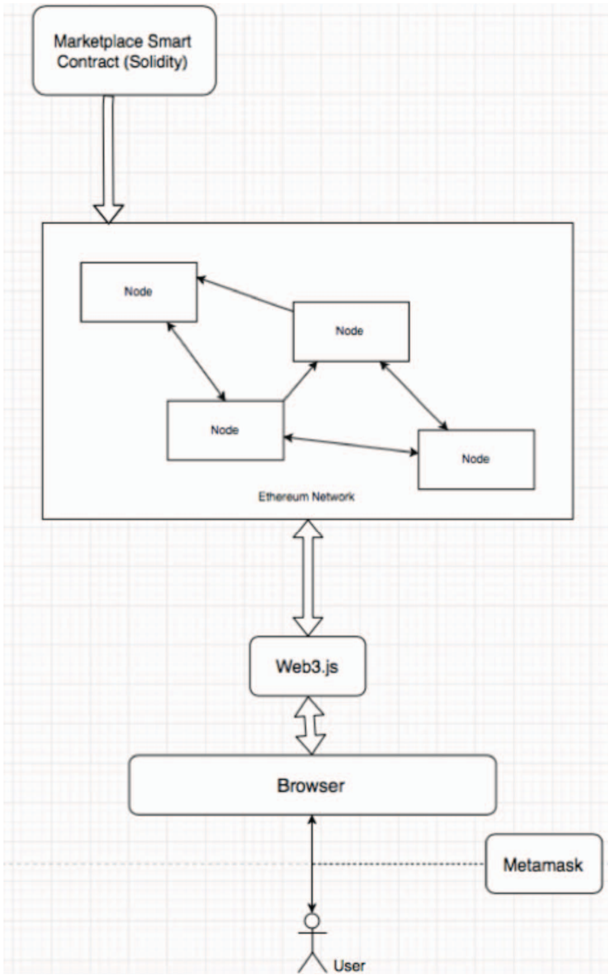


Fig. 3. Marketplace Infrastructure.

their ether wallets, enabling them to use Ethereum to pay for goods online.

The application also uses MongoDB (a JSON database that connects to applications using NodeJS) for optimization, as well as IPFS in order to keep item data stored securely in a decentralized fashion. A point of note here is that in the tested application, MongoDB was not used in the tested application, and the design was revised to eliminate MongoDB as a potential central point of failure, and adapted to utilize a decentralized database service such as BigChainDB instead. Due to its use of authentication tokens to preserve privacy, user data remains private on the blockchain, and cannot be viewed by malicious third parties. Only references and metadata about consumer information will be stored on the Ethereum blockchain; this way, the platform remains private. However, the users bids will still be shown on the blockchain. Providing confidentiality for user bids is out of the scope of this work. However, it is possible to

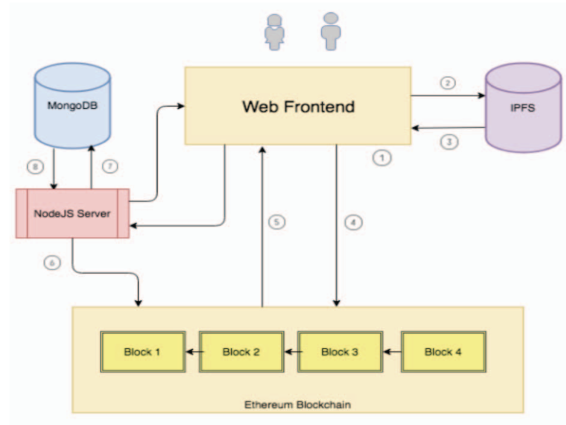


Fig. 4. Original Architecture of Proposed Application. In the revised design, MongoDB is replaced with a decentralized service like BigChainDB.

do this by using cryptographic methods described in [24].

In order for a transaction to be completed, gas, a fixed amount of ether proportional to the computational complexity of a transaction, is transferred to the node processing the transaction to incentivize its completion. Due to this incentivization, contract creation time depends on the complexity of the contract and system load. Similarly, transaction time depends on number of concurrent transactions occurring over entire system and complexity of application logic. The application logic resides in the app.js file. In this application, two separate contracts are used: E-commerce and Escrow.

In the application, multiple users bid using test ether. The winner of the auction gets the product after both buyer and seller agree or one of the two and the escrow party agree. After that, the ether is transferred to the agreed party.

The decentralized application has the following application flow (Fig. 3):

1) The web frontend contains HTML form where user enters the product details (name, start price, image, description etc.) and hits save.
2) The web frontend uploads the product image and product description to the IPFS and gets back links of those uploaded assets.
3) The web frontend then invokes the contract to store the product information and IPFS links on to the blockchain. An event is fired by the contract on successfully adding the product to the blockchain. The event contains all the product information.
4) The backend server is set up to listen to these events and when an event is fired by the contract, the server reads the contents of the event and inserts the product.

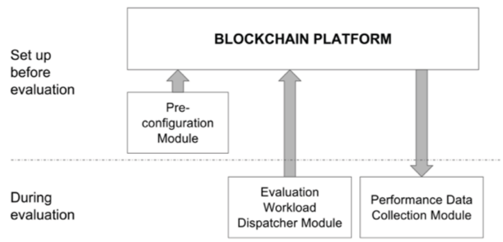


Fig. 5. A possible model for evaluating performance of a blockchain

The decentralized application has the following internal structure (Fig. 4):

1) Frontend: combination of HTML, CSS and JavaScript, with heavy use of web3js. Users will interact with the blockchain, IPFS and the NodeJS server through this frontend application.
2) Blockchain: This is the part of the application where all the code and transactions are stored. All the products in the store, user bids, and escrow are stored on the blockchain.
3) Backend: Although the products are stored on the blockchain, it is not efficient to query the blockchain to display products and apply various filters, thus we propose to use BigChainDB/NodeJS backend.
4) IPFS: When a user lists an item in the store, the frontend will upload the product files and description to the IPFS and store the hash of the uploaded file on to the blockchain as it is not efficient to store all images on the blockchain.

The following steps are involved in the implementation of the decentralized application:

1) Implement the contracts in Solidity using the Truffle framework, deploy it to Ganache and interact with the contract through the Truffle console.
2) Install IPFS and interact with it through the command line.
3) With backend implementation complete, the web frontend is built to interact with the contract and the functionality in the frontend.
4) Backend server installed data structure to store the product designed.
5) Once the database is up and running, implement the NodeJS server side code which listens to the contract events and logs the request to the console. Then implement the code to insert products on to the database.
6) Update frontend to lookup products from the database instead of the blockchain.
7) Implement the Escrow contract and the corresponding frontend where participants can release or refunds money to the buyer/seller.

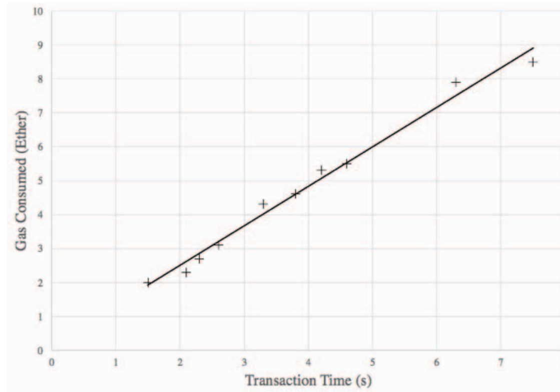


Fig. 6. Gas Consumption in comparison to Transaction Time.

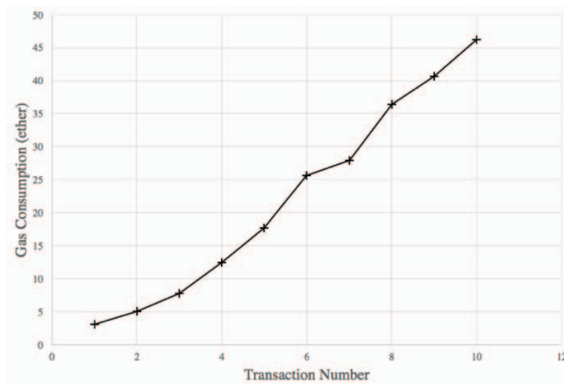


Fig. 7. The application's cumulative gas consumption.

V. Results and Discussion

Metrics were collected on the Rinkeby Test Network. The Rinkeby Test Network provides ether to test applications, as well as in-depth statistics that can be used to track the amount of time a transaction takes to execute, as well as the amount of gas consumed by the transaction [20]. Ganache was used to generate multiple users to simulate real world transactions [21].

In order to test the computational complexity of the application's transactions, the gas consumption was evaluated. The gas consumed, shown in Fig. 6, as expected, is proportional to the transaction time, and is also computed by the Ethereum network to be proportional to the computational complexity of a transaction [4]. This is due to the propensity of nodes to prefer processing less computationally-heavy operations, and in turn, slow down transactions that require a large amount of gas to execute. Average transaction time is 3.8 seconds with 4.6 gwei consumed, as shown in Fig. 8. Contract creation time is within 1 second. This shows that the test network is quite responsive to the application's transaction requests, and because of this, is relatively fast. These gas metrics are a

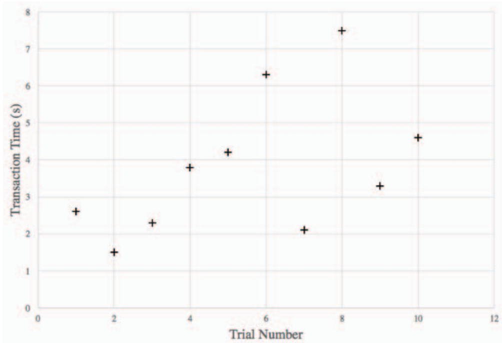
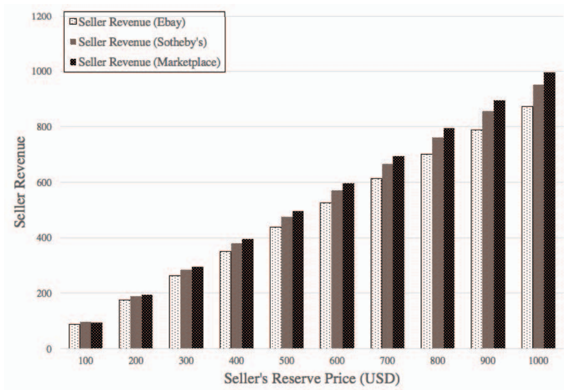


Fig. 8. The application's transaction runtimes. The variance in the transaction runtimes are attributed to test network metrics at the time.

Transaction gas costs

Transaction Type	Average Gas Cost (Truffle), USD	Average Gas Cost (MyEtherWallet), USD	Average Gas Cost (Mainnet), USD
Deployment	219.2399874	87.69599496	17.539198992
Add Product	17.420524478	6.9682097912	1.39364195824
Bid	4.9315752234	1.97263008936	0.394526017872
Reveal Bid	3.1529705452	1.26118821808	0.252237643616
Finalize Auction	46.742116505	18.696846602	3.7393693204
Release Funds	2.687711152	1.0750844608	0.21501689216
Call for Refund	1.903661802	0.7614647208	0.15229294416

Fig. 9. The gas costs for each transaction.



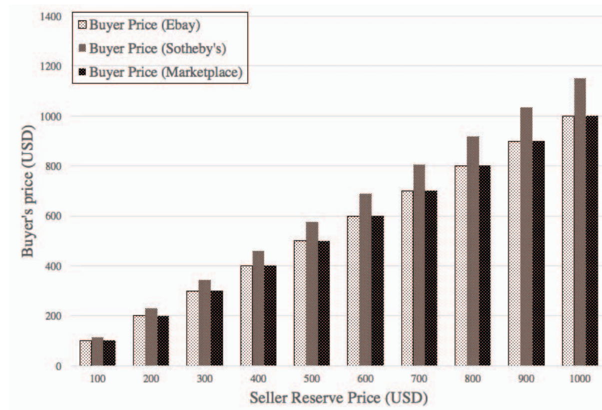


Fig. 10a and 10b. A comparison of reserve price and price to use the platform. In 10b, eBay's data is so close to that of the Marketplace that eBay's data nearly coincides with the Marketplace's data.

reflection of application complexity. Variations reflect the change in the load of the test network.

The cumulative gas, shown in Fig. 7, spent over the transactions when run on the Rinkeby network came to 46.2 gwei after 10 transactions were run, which is an acceptable amount for an Ethereum application. Later optimization may lead to a reduced cumulative gas consumption. We also examine deployment on the Ethereum Main network, which yields a lower total gas cost for the user.

Each of these transactions were run 10 times, leading to an average transaction time of 3.8 seconds, represented in Fig. 8. The variance in these results may have been caused by various test-network metrics at the time, such as network latency and throughput. The amount of gas consumed by each of these transactions also was a factor in these runtime evaluations. However, each transaction executes well enough for a buyer to bid for a product effectively.

In order to evaluate the cost-effectiveness of the application, we utilize an approach similar to that of Toyoda et al. [16]. The amount of gas used in each transaction was recalculated and converted to a dollar amount, which was then used to calculate the price of selling and buying an item with the application, and then compared with the amounts that both a buyer and seller must pay for use of services such as auction houses and eBay.

Each of the transactions were converted into US dollars by multiplying the amount of gas consumed with the default gas price of Truffle (1 gas = 100 gwei) [18], the recommended gas price of MyEtherWallet (1 gas = 40 gwei) [15], and the Main Ethereum Network gas price (1 gas = 8 gwei). This gas cost was determined to yield a mean transaction time of 16 seconds on the Main Ethereum Network as of Aug. 13, 2018, enabling buyers to bid in real time, regardless of network traffic. This value was determined using the Gas Tracker on etherscan.io, a website that the Ethereum project uses to track network statistics [17]. These gas costs are seen in Fig. 9.

The gas costs calculated with the Main Ethereum Network gas price were then used to calculate a possible seller revenue and buyer's price, both of which were compared with those of auction houses such as Sotheby's [14] and online platforms like EBay [12], represented in Fig. 10a and 10b.

The results of this comparison demonstrate that the application maximizes seller revenue when compared to that of EBay and auction houses like Sotheby's, an important step up from the traditional auction system. This increased seller revenue leads to decreased prices for buyers as well. The buyer's price is slightly higher than that of EBay by about USD $0.86 for all prices; however, at higher prices, this price difference becomes negligible. The slim difference in prices can be seen in Fig. 10b, in which the eBay data is not visible, due to this small difference. This flat per-purchase rate to use the platform is advantageous to the buyer, as the buyer does not have to pay a premium as a percentage of the reserve price. Due to the lowered cost to use the application, items sold on the website can become much more affordable for both buyer and seller.

VI. CONCLUSIONS

There is no mature application for a decentralized marketplace yet; This is a field that will become very important over the next 1-2 years as a number of startups are in the early stages of building similar products. At this stage, it is impossible to predict which decentralized marketplace will be the first to reach mass adoption, but creating a competing Dapp on the promising Ethereum platform with a simple UI is promising both for the experience gained in the technology and the potential it has to compete in a newly emerging field.

However, with regards to the problem definition stated earlier in the paper, our application can be considered successful as:

1) Through the use of smart contracts, sellers cannot be discriminated against by the system, as the rules for the system are set in stone by the smart contract.
2) As an added benefit, we have publicly verifiable transparency of the system rules. Moreover, the fact that the transaction data are stored on the blockchain simplifies the audit procedure.
3) It is shown in the cost analysis of the system that it is not expensive for a seller to list an item or sell the item, thereby driving prices for buyers down.
4) The Ethereum framework preserves the security of user data, ensuring that no users can acquire another user's information.

VII. FUTURE WORK

Future work in this field includes integrating a decentralized database system such as BigChainDB in order to enable a decentralized scheme of handling product information, and further optimizing transaction requests to deal solely with references to the product data in BigChainDB rather than with the data itself.

Another potential place for improvement in the application is the utilization of a consensus mechanism among the users to approve products being sold on the market. Due to lack of a centralized authority, sellers will not be at the mercy of a "company". Instead, our design harnesses the potential of decentralized computing, and uses consensus mechanisms to agree if the item should be sold or should be considered contraband.

Providing confidentiality of users' bids is a natural extension of this work. A variety of cryptographic tools can be applied to solve this problem; as specific solutions we can consider [24] (and the references therein).

VIII. REFERENCES

[1] Dejan Vujičić, Dijana Jagodić, Siniša Ranđić, "Blockchain Technology, Bitcoin, and Ethereum: A Brief Overview" 17th International Symposium INFOTEH-JAHORINA, 21-23 March 2018

[2] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system", [Online]. Available: https://bitcoin.org/bitcoin.pdf.

[3] Vanessa Bracamonte, Hitoshi Okada, Information and Society Research Division, National Institute of Informatics, Tokyo, Japan, "The Issue of User Trust in Decentralized Applications Running on Blockchain Platforms", 2017 IEEE International Symposium on Technology in Society (ISTAS) Proceedings.

[4] G. Wood, "Ethereum: a secure decentralised generalised transaction ledger", Ethereum Project Yellow Paper, vol. 151, pp. 1-32, 2014.

[5] "Tencent's latest game merges CryptoKitties and Pokémon Go", April 24 2018 Quartz Magazine https://qz.com/1260192/tencents-blockchain-game-merges-cryptokitties-and-pokemon-go/

[6] Suporn Pongnumkul, Chaiyaphum Siripanpornchana, Suttipong Thajchayapong, "Performance Analysis of Private Blockchain", 26th International Conference on Computer Communication and Networks (ICCCN), 2017.

[7] R. Tonelli, G. Destefanis, M. Marchesi and M. Ortu, "Smart Contracts Software Metrics: a First Study" arXiv:1802.01517v2, 7th Feb 2018.

[8] Peng Zhang, Michael Walker, Jules White, Douglas C. Schmidt, Vanderbilt University "Metrics for Assessing Blockchain-based Healthcare Decentralized Apps", IEEE 2018

[9] Maximilian Wöhrer and Uwe Zdun, University of Vienna "Smart Contracts: Security Patterns in the Ethereum Ecosystem and Solidity"

[10] Christian Decker, Roger Wattenhofer, "Information Propagation in the Bitcoin Network", 13-th IEEE International Conference on Peer-to-Peer Computing

[11] Edward Lehner, Dylan Hunzeker, John R. Ziegler, "Funding Science with Science: Cryptocurrency and Independent Academic Research Funding", ISSN 2379-5980 (online) DOI 10.5915/LEDGER.2017.108

[12] "Selling fees," *eBay Customer Service*. [Online]. Available: https://www.ebay.com/help/selling/fees-credits-invoices/selling-fees?id=4364. [Accessed: 11-Jun-2018].

[13] Koc et al. "Towards Secure E-Voting Using Ethereum Blockchain", 2018 IEEE 6th International Symposium on Digital Forensic and Security

[14] "Update Regarding Sotheby's Buyer's Premium on Sotheby's Blog," *Update Regarding Sotheby's Buyer's Premium*. [Online]. Available: http://www.sothebys.com/en/news-video/blogs/all-blogs/sotheby-s-at-large/2018/05/sothebys-buyers-premium-update.html. [Accessed: 13-Jun-2018].

[15] MyEtherWallet, "What is Gas?," *What is Gas? · Gas & Transaction Fees | MyEtherWallet Help & Support*, 26-Sep-2017. [Online]. Available: https://kb.myetherwallet.com/gas/what-is-gas-ethereum.html. [Accessed: 12-Jun-2018].

[16] Kentarō Toyoda, P. Takis Mathiopoulos, Iwao Sasase, Tomoaki Ohtsuki, Keio University, Kanagawa, Japan, "A Novel Blockchain-Based Product Ownership Management System for Anti-Counterfeits in the Post Supply Chain",

[17] etherscan.io, "Ethereum GAS Tracker," *Ethereum Transactions Information*. [Online]. Available: https://etherscan.io/gastracker. [Accessed: 13-Aug-2018].

[18] Truffle Suite, "Configuration," *Truffle Suite*. [Online]. Available: https://truffleframework.com/docs/advanced/configuration. [Accessed: 20-Jun-2018].

[19] A. Peterson, "eBay asks 145 million users to change passwords after data breach," *The Washington Post*, 21-May-2014. [Online]. Available: https://www.washingtonpost.com/news/the-switch/wp/2014/05/21/ebay-asks-145-million-users-to-change-passwords-after-data-breach/?noredirect=on&utm_term=.b3766d2ec50b.

[20] "Rinkeby: Ethereum Testnet," *Rinkeby: Ethereum Testnet*. [Online]. Available: https://www.rinkeby.io/#stats.

[21] Truffle Suite, "Ganache," *Truffle Suite*. [Online]. Available: https://truffleframework.com/ganache/.

[22] Pavlou, P. A., & Gefen, D. "Building effective online marketplaces with institution-based trust." Information Systems Research, 15(1), 37-59, 2004.

[23] Gimun Kim, Hoonyoung Koo, "The causal relationship between risk and trust in the online marketplace: A bidirectional perspective" Computers in Human Behavior Volume 55, Part B, February 2016, pp. 1020-1029.

[24] Erik-Oliver Blass, Florian Kerschbaum, "Strain: A Secure Auction for Blockchains", European Symposium on Research in Computer Security (ESORICS) 2018, pp. 87-110.