Toward a Decentralized, Trust-less Marketplace for Brokered IoT Data Trading using Blockchain

Shaimaa Bajoudah[1,2], Changyu Dong[1] and Paolo Missier[1]

Abstract— **As data marketplaces are becoming ubiquitous, it is also becoming clear that data streams generated from Internet of Things (IoT) devices hold value for potential third party consumers. We envision a marketplace for IoT data streams that can unlock such potential value in a scalable way, by enabling any pairs of data providers and consumers to engage in data exchange transactions without any prior assumption of mutual trust. We present a marketplace model and architecture to support trading of streaming data, from the advertising of data assets to the stipulation of legally binding trading agreements, to their fulfilment and payment settlement. We show that by using blockchain technology and Smart Contracts in particular, we can offer participants a trade-off between the cost of transactional data exchange, and the risk of data loss when trading with untrusted third parties. We experimentally assess such trade-offs on a testbed using Ethereum Smart Contracts.**

I. Introduction

Data streams that originate from Internet of Things (IoT) devices are increasingly viewed as tradeable assets with value not only to the device owners, but also with resell value, i.e., to third party buyers. New forms of dedicated data marketplaces are emerging to help unlock such value [1], but these are comparatively less mature than more traditional data marketplaces for static data, cf. eg [2], [3], [4] for surveys on these. Unlike static data, IoT data streams tend to lose their value if they are not consumed in near-real time, and data transmission and delivery may be unreliable. On the other hand, data exchange architectures based on message brokers systems such as MQTT allow a single data stream to be delivered to multiple parties, potentially enabling large-scale open marketplaces where data owners may resell their streams in real-time multiple times. While the IoT network and message-passing infrastructure can support a scalable marketplace, this inevitably leads to issues of mutual trust amongst participants, especially when those have no prior reputation within the marketplace. Also, the short-lived nature of streams requires efficient, automated mechanisms to create legally binding trade agreements, including payment arrangements, and to enforce such agreements throughout data transmission.

New generation blockchain technology that supports Smart Contracts is a natural choice to address all of these requirements, as Smart Contracts can act as a trusted intermediary within an untrusted community of marketplace participants, by adding transactionality to each of their interactions: before, during, and after data exchange. An example of such approach is Datum [5] (`datum.org`), based on the Ethereum network, which however is designed to let anyone *store* structured data on the blockchain. In contrast, we envision a decentralised marketplace for real-time IoT data, i.e., without any storage, that is scalable in the number of participants and does not require prior trust amongst them, while at the same time providing simple guarantees regarding data and monetary loss in case of participant's fraud. The marketplace should be able to flexibly accept new participants (either individuals, institutions or business organizations), be resilient to leaving participants, and accommodate unanticipated business relationships amongst those participants. Thus, anyone who controls IoT devices and generates IoT data streams should be able to monetize it and use it as tradeable assets in the marketplace. Additionally, in contrast to existing proposals, e.g. [6], we aim to define a marketplace that does not require a centralized trust component, such as a brokerage platform with trusted ownership, but relies instead on collective verification mechanisms, such as blockchain, to enforce its own governance rules.

Our approach involves using Ethereum Smart Contracts to support each phase of the interaction amongst a data provider and a consumer. It separates the data exchange interaction, which occurs on the IoT network and core cloud network, from transaction-based interactions aimed at enforcing non-repudiability of participant's actions and resolving their disputes, which occurs on the blockchain network.

A. Contributions

This work follows on from our earlier proposal for a IoT data marketplace, where we suggested that Ethereum is capable of supporting a fully decentralised marketplace without any assumption of mutual trust [7]. The approach suggested in [7] is based on the idea that each participant would periodically report to a Smart Contract on the data sent to and received from other participants, and the Contract would then be able to use such reports to settle any disputes.

In contrast, here we begin by proposing a different and much simpler protocol involving data providers, consumers, and a Smart Contract, based on the notion of periodic checkpoints during data exchange, supported by blockchain transactions to ensure limited scope for fraud on either side.

We then use our own prototype implementation of the marketplace model on a private Ethereum network, to experimentally evaluate the cost/risk trade-offs that are available

[1] School of Computing, Newcastle University, Newcastle Upon Tyne, UK. Email: s.bajoudah1@ncl.ac.uk, Changyu.Dong@ncl.ac.uk and Paolo.Missier@ncl.ac.uk

[2] College of Computer and Information Systems, Umm AlQura University, Makkah, KSA. Email: sbajoudah@uqu.edu.sa

by setting checkpoint frequency, also taking advantage of *potential* external mechanisms for establishing trust amongst participants, *if they are* available.

B. Related Work

The monetization of the huge amount of available IoT data is a challenging task with respect to automation and scalability. Many marketplaces exist that are designed to deal with IoT data using either centralized or decentralized architectures, for instance Microsoft Azure, BDEX (`bdex.com`), and Big IoT Marketplace (`http://big-iot.eu/`), a European project to enable IoT Ecosystems where IoT data producers can sell their data. These are all examples of centralised solutions where a central authority controls and manages the trades between data provider and data buyer.

A number of blockchain networks have been used to support IoT data exchange. Some, like Hyperledger (`hyperledger.org`),Quorum (`jpmorgan.com/global/Quorum`) and Corda (`marketplace.r3.com`) are private or permissioned. Hyperledger shows low latency requirements for consensus but does not fully satisfy decentralization goals, while both J.P.Morgan's Quorum and Corda target the financial sector using different approach, whereby IoT data are stored off chain and the consensus function is designed to ensure agreements among trade participants. The Ethereum blockchain [8], used as a testbed for this work, provides a public platform and automated agreements among interacting parties in the form of smart contracts and supports the development of DApps, making it one of the blockchin-based platforms of choice.

Some decentralized IoT marketplaces also exist. IDMoB [9] is designed to trade non real-time and not critical IoT data between IoT data producers and consumers. It runs on Ethereum and uses Smart Contracts to manage and control the market and to interact with the Raiden micropayment network.

The same as Databroker DOA (`databrokerdao.com`) which is a peer to peer marketplace for local IoT sensor data. Based on their white paper [10], the sensor owners place their data generated by their sensors up for sale. They believe their marketplace will have be the online retailers for sensor data.

Suliman A. et al [11] propose a marketplace to monetize IoT data using smart contract in the blockchain. Similar to our model, their approach involves sending IoT data through MQTT broker and using smart contracts to manage and settle payments. The main difference with our approach is that a deposit is required before subscription to a topic may take place. This conflicts with our no-trust assumption, as leaving a deposit ahead of receiving goods is likely to be viewed as risky by the buyer.

Huang Z. et al.'s decentralized platform for IoT data exchange [12] comes close to addressing issues of mistrust amongst participants, and similar to our approach, data is exchanged off-chain and made available to buyers once the contract is in place. However, the data to be purchased is stored, making this solution unsuitable for streaming. Furthermore, no guarantees are offered to ensure that the data is genuine, so advance payment i.e. to get access to data download is risky.

Another effort has emerged in IoT marketplace in the area of data source verification. Datapace (`datapace.io`) is a distributed and decentralised system based on blockchain with technical and policy-based data verification. It is a marketplace for IoT sensor data where the IoT sensors are connected the IoT platform Mainflux which is integrated into Datapace system part called Datapace IoT platform. The difference between this model and our model that this model provide data source verification by their own sensing equipments. While our model assume data source producers' honesty and no special verification hardware.

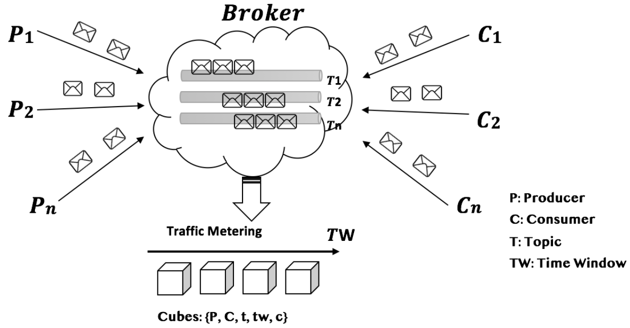
Similarly, AnyLedger (`anyledger.io`) is an embedded wallet for the IoT devices which connect the physical world to the blockchain. Each IoT device will be able to execute transaction to the blockchain. It is the first IoT-Blockchain application enablement platform starting from hardware device and the embedded software and finally end with the remote device management and blockchain nodes. Based on their white paper [13], AnyLedger blockchain solutions allow seamless deployment of tamper proof sensors, which is remotely controlled. It uses IPFS technology as decentralised storage end point for secure storage for data monetization.

Finally, a recently proposed alternative blockchain provider, IOTA (`iota.org`), announced their support for decentralized marketplaces at the end of 2017, with the goal of "enabling a truly decentralized data marketplace to open up the data silos that currently keep data limited to the control of a few entities". One distinguishing feature of this solution is that, unlike others cited above, here the IoT data is actually stored in the blockchain (or IOTA's version of it, called the Tangle [14]. To the best of our knowledge this solution has not yet been released.

As opposed to IOTA, Streamer (`streamr.com`) felt that there is no need to develop a completely new blockchain and instead, saving resources by using the existing Ethereum blockchain. It is a real time data streams exchange platform. It creates an ecosystem for data producers to sell their data to consumers. As explained in their white paper, a data producer creates a data streams for their data and push it to brokers nodes which is responsible to deliver it to its data consumer who purchase the desired data by the interaction with the Ethereum smart contract for management, data permission and payment.

Trust and reputation management is not directly addressed in this paper, however a trust management model should also be established as part of the marketplace. Existing trust frameworks can be used on top of our infrastructure. Yan et al. [15], for instance, explore the notion of trust across the IoT platform layers (physical sensing, network, and application layers), with the focus on a wide range of properties from security to goodness, strength, reliability, availability, ability of data. However, their survey largely overlooks issues of trust amongst participants in a data marketplace, i.e., in the context of data exchange transactions. More directly useful

Fig. 1: Centralized Brokered IoT Data Marketplace Architecture



in our setting, is Roman and Gatti's study of trust in data marketplaces [16], based on *credit scoring*, where a direct connection is made to the use of blockchain technology with data trading.

II. Marketplace Model

A. Brokered IoT Data Exchange

We assume, following standard IoT data streaming practices, that the exchange of streaming data between any pair of participants, i.e., a data Provider P and a Consumer C, is mediated by some transaction-agnostic broker infrastructure, such as the one shown in Fig. 1. In this data transfer model, the stream is broken down into discrete message batches. Providers tag their messages with *topics* that uniquely identify that Provider's stream. A Consumer is allowed to subscribe to a topic subject to the conditions set in a Trade Agreement, as described below.

In our previous work [7] we assumed initially a network architecture where the broker is a trusted component that can be relied upon to generate truthful data exchange reports, which in turn can be used to settle disputes between producers and consumers (the "cubes" in Fig. 1). In such a scenario, the Smart Contract is simply in charge of settlement given the reports. In the same paper, we then proposed a more ambitious trust-less model where the task of generating reports is left to each participant. In this case the Smart Contract has a difficult task because the report themselves cannot be trusted, and disputes cannot be settled by ascribing certain responsibility to either participant.

B. Model Elements

In this work we work around these difficulties, as we do not require the broker or the participants to generate any report at all. Instead, the broker is simply a network element. The goal of the marketplace is twofold. Firstly, to enable trading of streaming data through the broker while offering guarantees, i.e., regarding the max loss incurred by either of them in case of adversarial behaviour. And secondly, to resolve disputes about the amount of data exchanged. To achieve this, we augment the data exchange with the exchange of *data receipts* between C and P, which occurs at regular intervals and throughout the duration of the data stream. Such receipts are exchanged as part of transactions that are mediated by a smart contract, denoted SC, on the blockchain. The length of the exchange interval, denoted as Batch Size or BS, is set at the time of trading agreement negotiation. As we will see, this parameter enables P to control the level of risk they are prepared to tolerate given limited trust in C.

The model consists of the following elements:

1) The description of data offered by a Producer;
2) A trade agreement, which includes details of the data to be exchanged and the exchange protocol, the corresponding market value, and additional parameters such as BS mentioned above;
3) A protocol for the exchange of data receipts, which includes both parties in addition to a neutral smart contract;
4) A reputation model, which allows a reputation score to be assigned to every pair P and C of participants at the end of each transaction they are involved in. Participants may use reputation scores to assess the risk of entering into an agreement with an untrusted participant.

In this paper we are concerned primarily with (1-3), which are described in detail below. Regarding (4), we are going to assume that a reputation model is in place and that a up-to-date score is associated with each participant, without concern for how it works. The design of a customised reputation model is the object of our ongoing work, and it is beyond the scope of this paper. Proposals on how to achieve such a model exist, however, see eg. [17].

The smart contract is responsible for each transaction associated with (1-3), and specifically for recording (i) the specification of the data offering, (ii) the trade agreement, and (iii) each data receipt.

C. Data Offering

The first function of the smart contract is to let data Providers publish their data offerings on the blockchain, where they can be then discovered by prospective Consumers. As mentioned, a data stream consists of a sequence of messages uniquely identified by a provider's topic, and a data offering describes the type of stream and specifies how to subscribe to the stream. Specifically, a data offering $DO = \langle T, TI, MR, UP \rangle$ includes, in addition to the topic T, a specification of (i) the time interval TI during which the offer is valid, (ii) the expected streaming message rate MR, eg. in messages/time, (iii) the unit cost UP of each message in the stream. Note that here we are only concerned with the overhead cost of trading, while the pricing of the data itself is not a concern in this work. Interested readers may find recent proposals on data pricing relevant [18], [19], [20], [21].

D. Trade Agreement

The trade agreement is a legally binding contract (we use the term "agreement" to avoid confusion with smart contracts) between a producer and a consumer, which defines the terms of the data exchange. An agreement comes into force when (i) it is signed by both parties using their blockchain account keys (Ethereum in our implementation), and (ii) a smart contract transaction containing the agreement is committed to the blockchain, at which point it can no longer be amended. The agreement contains (i) a specific data offering DO and (ii) a time interval $TATI$, contained within the time interval TI, during which the agreement is in force. For instance, C may want to subscribe to a portion of an event that is offered over a long period of time. We denote the total price as $TP = UP \cdot TATI$ and the *estimated* total number of messages in the agreement as $ETM = MR \cdot TATI$. The latter is an estimate, rather than a set value, because the total number of messages that can be sent within interval $TATI$ is affected by the time required to carry out the Data Receipt protocol, as explained next.

E. Data Receipt protocol

Once the trade agreement is in force, C is allowed to subscribe to P's stream. Under normal circumstances and when both parties comply with the agreement, and data transfer takes place as expected, at the end of the $TATI$ interval C informs SC that the agreement has been fulfilled, and SC proceeds to settle the payment as per the agreement. Suppose however that C fails to inform SC. This may happen because C actually failed to receive some of the data in the stream, or because it fraudulently *claims* not to have received the data. In our model we assume that SC is unable to distinguish between these two events, because there is no requirement for the data broker to keep a (verifiably truthful) log of its message delivery. In this situation, the only possible course of action for SC is to believe C's claim, and to withhold P's payment as a consequence. Thus, assuming minimal accountability on the broker and no trust amongst participants, P may become the victim of C's fraud.

Our approach to mitigate this circumstance is to introduce *checkpoints* throughout the duration of data delivery. The number of messages between two checkpoints is the batch size BS, which P can configure as part of the agreement negotiation with C. At each checkpoint, C is expected to send a *data receipt* to SC as part of a blockchain transaction, which acknowledges receipt of one batch of data from P. When the transaction is confirmed, SC records the receipt and then informs P. Meanwhile, at the end of each batch P will have suspended its streaming to C until it receives the acknowledgment from SC. If P does not receive a message within a certain time limit, it times out and terminates the trade agreement in order to cut its losses (in practice, C's subscription to the stream is cancelled). Thus, the data exchange protocol and data receipt protocols are interconnected as shown in Fig. 2.

The timeout is a configurable parameter that reflects the expected time required for a receipt transaction to be confirmed on the blockchain. In our experiments we model this time as a random variable, denoted RT (for Receipt Time), with an experimentally determined distribution (see Sec. IV). P may configure the timeout RT_{max} to be more or less tolerant of the variance in confirmation times, however longer RT_{max} intervals translate into fewer effective messages delivered to C, as in our model the latency RT_{max} counts as part of the total agreement interval, $TATI$, as explained next.

F. Cost / risk / time trade-offs

In the model just illustrated, P and C agree on a total duration for the data streaming, $TATI$, and a streaming rate, MR. We have also assumed that P has a way to assess the risk of *data loss* when C is not trustworthy, i.e., by accessing C's reputation score (the details of which we have omitted). In this setting, the term data loss refers to the number of messages that P will have sent to C, that C will not acknowledge and therefore will not pay for.

In order to minimise its risk, P is motivated to choose frequent checkpoints, that is, by setting BS to a small value. This, however, has a cost impact, because checkpoints are smart contract transactions and as such, in blockchain models like Ethereum, each of them incurs a fee. There is therefore a trade-off between the risk of losing data and the cost of engaging in a long-running trade with many checkpoints along the way.

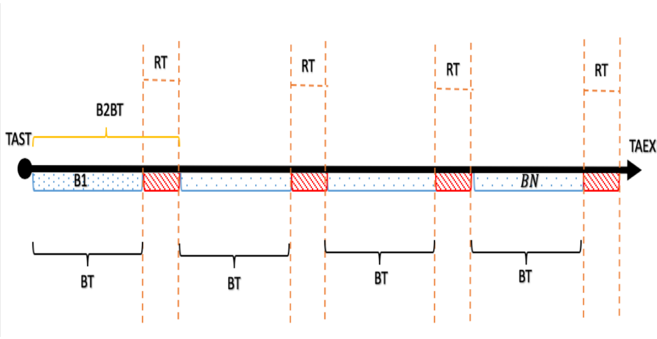
Now suppose that (i) the transaction fees for data receipts are charged to C, and (ii) the latency RT due to each data receipt transaction is detracted from the total contract time, $TATI$.

These conditions potentially create a tension between P and C, as P is interested in low risk, while C is interested in low transaction cost and minimal reduction in effective contract time. Such tension is embodied by C's reputation score, $C_{rep} \in [0, 1]$. The ideal win-win scenario occurs when C is fully trusted, that is, $C_{rep} = 1$, as in this case there is no need for checkpoints, i.e., $BS = TATI \cdot MR$. When $C_{rep} < 1$, P will set $BS < TATI \cdot MR$, and C will experience higher cost and fewer total messages delivered within $TATI$.

Thus, it makes sense to assume that BS is a function of C_{rep}: $BS = f(C_{rep})$, where $f()$ is a parameter in the model and can be chosen to either amplify or reduce the effect of reputation. In our experiments we have used a logarithmic function: $f(C_{rep}) = \ln(C_{rep} + 1)/k$. The analysis in Sec. IV shows that by choosing a suitable value for the constant k, we can control the min number of receipts required. For instance, setting $k = 2.77$ has the effect to produce a minimum of 4 data receipts. Many other functions can be chosen to map the reputation score to the batch size, and thus indirectly to the number of receipts.

It is straightforward to see that, in this model, P's maximum data loss is simply one batch of messages. As mentioned, RT_{max} denotes P's estimate of RT, which will be used as timeout. RT_{max} is determined from RT's empirical distribution, observed from the network behaviour. In practice, the expected confirmation time for a Smart

Fig. 2: Data receipt protocol Interactions between C, P and SC



Contract transaction on Ethereum is largely determined by the user's choice of gas price.

While this setting has no direct effect on data loss, it does affect C's cost and the actual number of messages received. Because of our assumption (ii), a large value for RT_{max} results in fewer messages sent to C. To make this observation precise, consider $TATI$, MR and BS as constants from the agreement. The time required to send a batch of data is

$$BT = \frac{BS}{MR}$$

and the number of batches sent during $TATI$ is

$$BN = \frac{TATI}{BT + RT_{max}}$$

because of assumption (ii) above. Equivalently,

$$BN = \frac{TATI \cdot MR}{BS + RT_{max} \cdot MR}$$

Assuming the Ethereum cost model with gas unit price GUP and gas consumption per transaction GT, the total cost RC due to the receipt transactions is:

$$RC = BN \cdot GT \cdot GUP$$

As expected, the cost is inversely proportional to BS.

The actual number of messages ATM delivered at the end of $TATI$ is

$$ATM = (TATI - BN \cdot RT_{max}) \cdot MR$$

which decreases as BS and RT_{max} increase, as expected.

As assumed above (i), cost RC is charged to C. The total cost associated with a trade agreement also includes one-off transaction fees, which are split between P and C, as follows. Firstly, in order to participate in the marketplace each participant, in either a consumer or producer role, must register itself with the network. This incurs a one-off *registration cost* to execute the smart contract user registration function. Secondly, the deployment of a trade offer to the network is also implemented as a smart contract function, which again incurs a fee. This is a provider-only cost. Thirdly, a smart contract fee is paid when a new trade agreement is recorded on the blockchain. This cost is split between P and C.

G. *End of trade*

Marketplace practice suggests that C should pay a deposit at the start of the trade, as a guarantee that sufficient funds are available to settle the agreement at the end of it. The funds are held by the SC and used against the final payment, or they are returned to C in case the trade is terminated early, i.e., if P times out on a data receipt. Importantly, however, this deposit cannot be used as leverage to ensure C's honesty, because we have assumed that SC cannot distinguish between fraud and genuine data loss, i.e., in the brokered network. Thus, deposit details do not add to the specific model we are proposing, and we are not going to elaborate further.

Finally, we have overlooked details of the reputation model, as it is beyond our current scope. It is important to note, however, that the reputation manager should be able to update participants' reputations at the end of each trade, i.e., based on the outcome of the trade. This is not straightforward, because a trade terminating early does not automatically apportion blame to either C or P. The design of a dynamic reputation model that can deal with this situation is the focus of our current research.

III. System view and marketplace interactions

The system consists of a data transfer layer, where IoT data transfer is mediated by brokers, and a blockchain layer, where all trade-related transactions occur.

In data transfer layer, the actual data is transferring from producers to consumers off-chain (in broker level) in different batch sizes as stated in the trade agreement in blockchain layer.

The blockchain layer consists of a collection of smart contracts SC written in Solidity, Ethereum's smart contract language, and executed on the Ethereum Virtual Machine, EVM).

As shown in Figure 3, initially a new participant must register itself in the blockchain, by calling the register function of SC. Data providers P publish their data offers , or post updates to current offers, again using SC so that the offers are stored in the blockchain and are publicly visible. At the same time, consumers C can inspect offers, and then make a request to the SC including the reference to and the required time interval. This causes a new Trade Agreement to be created on the blockchain, possibly encrypted by the consumer for privacy purposes.

The offer's provider is then involved in the definition of the agreement, which includes setting parameters BS and RT_{max} based on the consumer's current reputation score. The negotiation phase occurs out of band and is not part of our implementation. Once C and P sign the agreement, this is posted on the blockchain through a SC call.

IV. Implementation and evaluation

Our testbed consists of a set of smart contracts for trade management and monetary settlement, deployed on a private Ethereum test network. We used Ethereum's web-based IDE Remix (`remix.ethereum.org`) to write, deploy and connect to the private chain through Remote Producer Calls. We used fake accounts with balances provided by Remix as trades participants.

Here we experimentally determine the costs associated with each phase of the $P - C$ interaction through SC. To recall, SC and thus gas fees are involved in registering new participants, to deploy new offers, and to create a new trade agreement. Once the agreement is in place, C provides a deposit TP based on ETM, as suggested earlier (Sec. II-G). Importantly, we assess the cost RC due to the data receipt protocol.

Table I shows the costs broken down per phase, incurred by P, C, or both. We have measured the gas consumption using the Remix debugger, which provides consumed gas for every transaction. This can also be obtained by monitoring the balances of participants and check the differences before and after invoking the smart contract method.

TABLE I: shows transactions cost in each cost category (in Gas)

Cost Category	**Operation**	**Producer Gas Consumption**	**Consumer Gas Consumption**
Registration Cost	- Register in the network	204739 gas	199093 gas
Offering Cost	- Deploy an offer	491862 gas	-
Setup Cost	- Make an order and create a new TA	-	620865 gas
	- Set Batch size and sign off the TA	82063 gas	-
Receipt Cost	- Send a receipt	-	144367 gas

The settlement is done by the settlement smart contract when the trade ends.

For evaluation purposes, we have defined a family of functions $BS = f(C_{rep}) = \ln(C_{rep}+1)/k$ where parameter k is set by constraining the minimum number of receipts when $TATI$ is set to one day (24 hours) and $MR = 100 msgs/s$. The three columns in Table II show the effect of setting $k = 1.38, 2.10, 2.77$, with corresponding min receipts 2,3, and 4, across the range of reputation scores.

TABLE II: Minimum number of receipts with three different constant values in $f(C_{rep})$

Reputation	**Number of Receipt**		
	$\ln(C_{rep}+1)/1.38$	$\ln(C_{rep}+1)/2.10$	$\ln(C_{rep}+1)/2.77$
0.1	15	21	29
0.2	8	11	16
0.3	6	7	11
0.4	5	6	9
0.5	4	5	7
0.6	3	4	6
0.7	3	3	6
0.8	3	3	5
0.9	3	3	5
1	**2**	**3**	**4**

Setting $k = 1.38$ produces the minimum number of receipts, 2, for $C_{rep} = 1$ and increases to 3 for $0.5 \leq C_{rep} \leq 0.9$ In contrast, $k = 2.10$ produces the min number of receipts within reputation range $C_{rep} \geq 0.7$, and for $k = 2.77$, the min occurs for $C_{rep} = 1$, while the number of receipts are fairly evenly distributed in the range $0.6 \leq C_{rep} \leq 0.7$ with 6 receipts and for $0.8 \leq C_{rep} \leq 0.9$ with 5 receipts.

Because the focus of our experiment is the trade-off between the cost and the consumer reputation, based on these experiments we settled for $k = 2.77$ for our cost evaluation, as this provides as good segregation of cost relative to reputation while limiting producer loss.

The increase in a number of batches received means that a consumer will incur more gas. If we assume that a producer has no incentive not to send the data as agreed in the agreement, a trade fails when the consumer fails to send a receipt or was not honest in reporting the exact number of messages received. The data receipt protocol is designed to make the marketplace sustainable by providing incentives to parties to increase their reputation.

Because the unit gas price GUP largely determines the duration of the transactions in the blockchain to be confirmed by miners, a consumer has the option to increase the GUP in order to process their transaction faster and therefore he will have more timem out of the total $TATI$, to receive more batches. To clarify, in the Ethereum cost and POW model, higher GUP gives the SC transaction priority, as miners who will validate the transactions usually follow the strategy of picking the transactions with higher GUP to be included in the next block. Thus, the increase in GUP contributes to decreasing RT and therefore provides a larger ATM (actual total messages) within the $TATI$ interval.

The minimum and the maximum GUP in the network can be found using the *ETH Gas Station* (`ethgasstation.info`). This is a tool to understand the conditions of the current gas market and current policies of network miners. Based on the current condition of the network at the moment of writing, the recommended gas prices from Gas Station is shown in Table III. The Table shows the maximum time taken by miners to confirm the transaction for each GUP. In addition, the Gas Station provides the median time of transaction confirmation for each GUP.

For the purpose of evaluation, we have used the three different GUP for different consumer reputations to calculate

Fig. 3: System Sequence Diagram

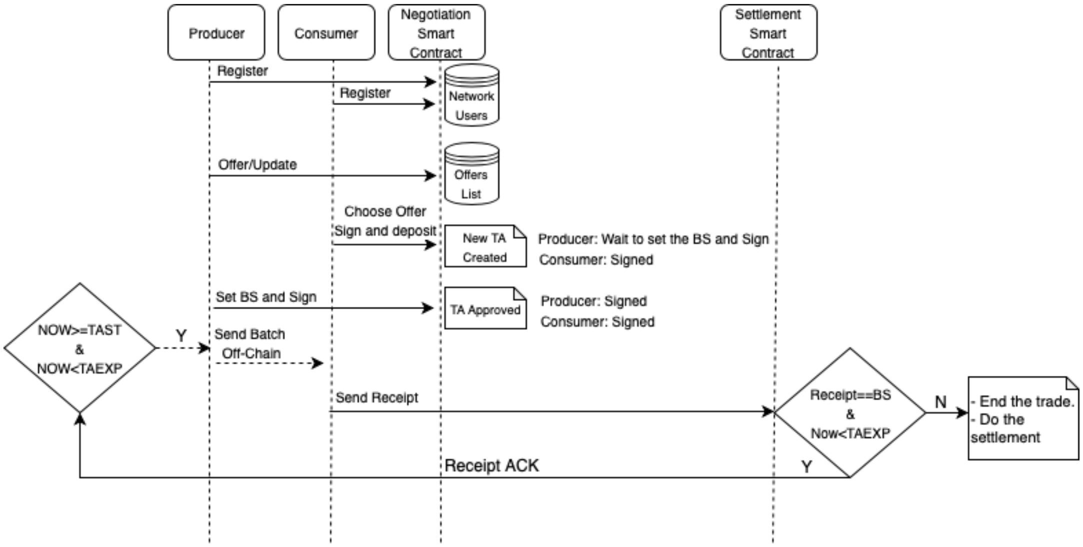


TABLE III: Gas Prices and Speeds

Gas Price (Gwei)	Speed	Median Speed
1.6	SafeLow (<30m)	4.3m
4.2	Standard (<5m)	2.5m
7.2	Fast (<2m)	0.8m

the RC, as explained in Sec. II-F.

TABLE IV: Shows the RC in USD and Data Percentage Delivered for consumer reputations for 1 day trade with three GUP values, MR=100 msg/sec. (1 Eth $\approx$ 202.70 USD)

Consumer Reputation	GUP= 1.6 Gwei		GUP= 4.2 Gwei		GUP= 7.2 Gwei	
	Cost in USD	Data Percentage	Cost in USD	Data Percentage	Cost in USD	Data Percentage
0.1	1.465\$	92.236%	3.970\$	95.313%	7.016\$	98.444%
0.2	0.903\$	95.81%	2.372\$	97.57%	4.277\$	99.17%
0.3	0.716\$	97.01%	1.881\$	98.26%	3.224\$	99.44%
0.4	0.623\$	97.61%	1.635\$	98.61%	2.802\$	99.55%
0.5	0.529\$	98.20%	1.389\$	98.95%	2.381\$	99.66%
0.6	0.482\$	98.51%	1.266\$	99.13%	2.170\$	99.72%
0.7	0.482\$	98.51%	1.266\$	99.13%	2.170\$	99.72%
0.8	0.435\$	98.81%	1.143\$	99.30%	1.960\$	99.78%
0.9	0.435\$	98.81%	1.143\$	99.31%	1.960\$	99.78%
1.0	0.388\$	99.10%	1.020\$	99.48%	1.749\$	99.83%

Fig. 4(a) shows the number of smart contract invocations, that is, the number of receipts, vs consumer reputation. The cost of these invocations is depicted in Fig. 4 (b).

Note that the maximum data delivery when $C_{rep} = 1$ for $GUP = 1.6\ Gwei$, $4.2 Gwei$ and $7.2\ Gwei$ are 99.10%, 99.48%, and 99.83%, respectively. If we assume that the minimum time for a transaction to be confirmed is about 1 second, the maximum number of messages could be delivered to a consumer is $\leq (ETM - MR * BN)$. Recall that the overhead due to processing the receipts, represented as RT, is included in $TATI$, which reduces ETM by $RT \cdot BN$.

Although the number of receipts as shown in Fig. 4 (a) are nearly the same, which lead to similar costs for receipts as shown in Fig. 4 (b), the fraction of data received, ATM, is higher for a higher reputation consumer as expected, as shown in Fig. 4 (c).

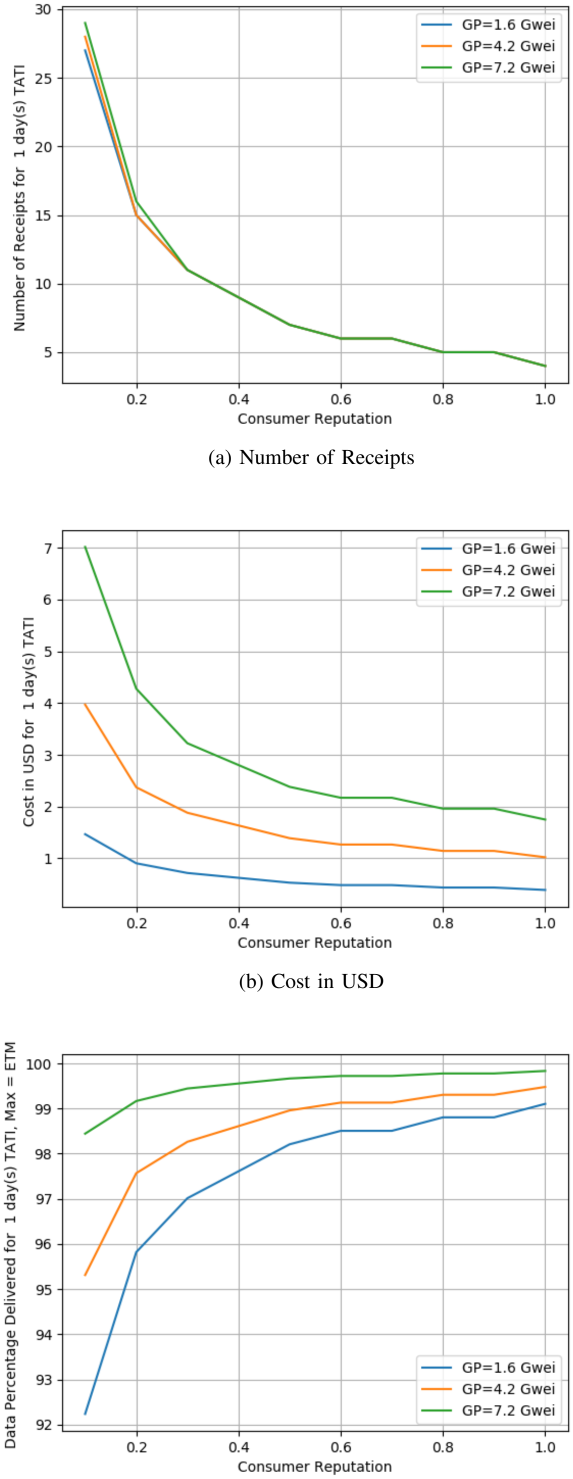
V. CONCLUSIONS

In this paper we have proposed a decentralized marketplace for trading brokered IoT data under assumptions of limited trust amongst participants. Smart Contracts on the Ethereum public network are used to mediate all interactions amongst data producers and consumers, in order to achieve non-repudiability and transparency. The model separates the exchange of streaming data, which is supported by message brokers off-blockchain, from transactions that occur at regular checkpoints during data transfer.

The model makes the trade-offs between risk of data loss, cost, and total number of messages exchanged explicit, empowering the participants to negotiate a balance between those elements, based on their current reputation. We have shown experimentally that, if a reputation score can be obtained, i.e., from a third party service that is currently beyond the scope of this paper, then the trade-offs can be easily quantified, making trading risk becomes manageable.

This work is still in progress and, in the long run, aims to deliver a customised reputation model where reputation changes dynamically as a function of the history of past trades in the marketplace. Our challenge is then to show that (i) the marketplace encourages honest behaviour, i.e., participants have an incentive to increase their reputation over time, and (ii) it is economically viable, vis a vis the trading costs.

Also, we work - as an future work and work in progress - on the marketplace scalability with the growth in participants number and the transaction confirmation time in the blockchain network, and how well-designed reputation model guarantee less time for higher reputation participants.



(a) Number of Receipts

(b) Cost in USD

(c) Data Percentage Received

Fig. 4: The cost, the number of receipts and the percentage of the data received for three different gas prices

ACKNOWLEDGMENT

This research has been funded by Umm AlQura University, Makkah, The Kingdom of Saudi Arabia.

REFERENCES

[1] Kresimir Misura and Mario Zagar. Internet of things cloud mediator platform. In *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on*, pages 1052–1056. IEEE, 2014.

[2] Charith Perera, Chi Harold Liu, and Srimal Jayawardena. The Emerging Internet of Things Marketplace From an Industrial Perspective: A Survey. *IEEE Transactions on Emerging Topics in Computing*, 3(4):585–598, dec 2015.

[3] Fabian Schomm, Florian Stahl, and Gottfried Vossen. Marketplaces for Data: An Initial Survey. *SIGMOD Rec.*, 42(1):15–26, 2013.

[4] Florian Stahl, Fabian Schomm, Gottfried Vossen, and Lara Vomfell. A classification framework for data marketplaces. *Vietnam Journal of Computer Science*, 3(3):137–143, aug 2016.

[5] Roger Haenni. Datum Network - The decentralised data marketplace. *White paper*, 2017.

[6] Tien-Dung Cao, Tran-Vu Pham, Quang-Hieu Vu, Hong-Linh Truong, Duc-Hung Le, and Schahram Dustdar. MARSA: A Marketplace for Realtime Human Sensing Data. *ACM Trans. Internet Technol.*, 16(3):16:1—-16:21, may 2016.

[7] Paolo Missier, Shaimaa Bajoudah, Angelo Capossele, Andrea Gaglione, and Michele Nati. Mind my value: A decentralized infrastructure for fair and trusted iot data trading. In *Proceedings of the Seventh International Conference on the Internet of Things*, IoT '17, pages 15:1–15:8, New York, NY, USA, 2017. ACM.

[8] Vitalik Buterin. A next-generation smart contract and decentralized application platform. 2014.

[9] Kazım Rıfat Özyılmaz, Mehmet Doğan, and Arda Yurdakul. Idmob: Iot data marketplace on blockchain. *arXiv preprint arXiv:1810.00349*, 2018.

[10] Roderik van der Veer Matthew Van Niekerk. Databroker doa global marketplace for local data.

[11] Ahmed Suliman, Zainab Husain, Menatallah Abououf, Mansoor Alblooshi, and Khaled Salah. Monetization of iot data using smart contracts. *IET Networks*, 2018.

[12] Zhiqing Huang, Xiongye Su, Yanxin Zhang, Changxue Shi, Hanchen Zhang, and Luyang Xie. A decentralized solution for iot data trusted exchange based-on blockchain. In *Computer and Communications (ICCC), 2017 3rd IEEE International Conference on*, pages 1180–1184. IEEE, 2017.

[13] Lorenzo Pieri Bogdan Djukic. Anyledger: Embedded wallet for decentralized iot.

[14] Serguei Popov. The Tangle. 2016.

[15] Zheng Yan, Peng Zhang, and Athanasios V. Vasilakos. A survey on trust management for Internet of Things. *Journal of Network and Computer Applications*, 42:120–134, jun 2014.

[16] Dumitru Roman and Gatti Stefano. Towards a Reference Architecture for Trusted Data Marketplaces: The Credit Scoring Perspective. In *2016 2nd International Conference on Open and Big Data (OBD)*, pages 95–101. IEEE, aug 2016.

[17] Alexander Schaub, Rémi Bazin, Omar Hasan, and Lionel Brunie. A trustless privacy-preserving reputation system. In *IFIP International Information Security and Privacy Conference*, pages 398–411. Springer, 2016.

[18] Chao Li, Daniel Yang Li, Gerome Miklau, and Dan Suciu. A Theory of Pricing Private Data. *ACM Trans. Database Syst.*, 39(4):34:1—-34:28, dec 2014.

[19] Soumya Sen, Carlee Joe-Wong, Sangtae Ha, and Mung Chiang. Smart Data Pricing: Using Economics to Manage Network Congestion. *Commun. ACM*, 58(12):86–93, nov 2015.

[20] Dusit Niyato, Dinh Thai Hoang, Nguyen Cong Luong, Ping Wang, Dong In Kim, and Zhu Han. Smart data pricing models for the internet of things: a bundling strategy approach. *IEEE Network*, 30(2):18–25, mar 2016.

[21] Dusit Niyato, Xiao Lu, Ping Wang, Dong In Kim, and Zhu Han. Economics of Internet of Things: an information market approach. *IEEE Wireless Communications*, 23(4):136–145, aug 2016.