AutomataDAO: A Blockchain-Based Data Marketplace for Interactive Robot and IoT Data Exchanges Using Ethermint and State Channels

Irvin Steve Cardenas, John Brian May, and Jong-Hoon Kim

Abstract We present AutomataDAO, a decentralized data marketplace for robot and IoT cyberphysical systems. Unlike present literature on blockchain-based data marketplaces, AutomataDAO is implemented as a decentralized autonomous organization (DAO) through a set of governance smart contracts that account for both human and synthetic agents that may take part in the DAO. Concerns over transactional costs and throughput are resolved using state channel technology, and a preliminary decentralized ID (DID) solution is leveraged to establish a reputation system that is considered within the cryptoeconomics of the system; the latter is also accounted for in the market's data pricing model.

Keywords Data marketplace · Mobile robots · IoT · Blockchain · Smart contracts · Cosmos network · Ethermint

1 Introduction

Mobile robots present ample opportunities for data collection. Unlike, stationary robotic systems (e.g. industrial manipulators) or traditional Internet of Things (IoT) devices (e.g. motes)—mobile robots can navigate remote environments or inspect points of interest be it through direct teleportation, semi-autonomously or fully autonomously. Whether it be an unmanned ground vehicle (UGV) or an unmanned aerial vehicle (UAV)—data collection performed by mobile robots can be more thorough, accurate and reliable than data sourced from static sensors or crowd-sourced from humans. Data collection can be more thorough and accurate due to a mobile robot's ability to navigate and cover unsurveyed areas. It can be more reliable due to the ability of networked robots to collaborate and re-assign sensing tasks during

I. S. Cardenas · J. B. May · J.-H. Kim (✉)
Advanced Telerobotics Research Lab, Department of Computer Science,
Kent State University, Kent Ohio, USA
e-mail: jkim72@kent.edu
URL: http://www.atr.cs.kent.edu

S.-W. Lee et al. (eds.), *Blockchain Technology for IoT Applications*,
Blockchain Technologies, https://doi.org/10.1007/978-981-33-4122-7_2

failures. Overall real-time data collected, at the extent that robots are able to, can be of great value to both businesses and the commons.

The present most efficient means to acquire large amounts of data is through centralized data marketplaces. These marketplaces purchase data, perform due diligence on data providers, and directly re-sell or broker data through dedicated online portals. Overall, these services are necessary due to the lack of trust between the originating data providers and buyers. This lack of trust is due to various reasons such as concerns over: (1) the quality or integrity of datasets, (2) data collection practices or (3) long-term reliability of the data provider. Similar to other traditional centralized marketplaces, data marketplaces tackle two-sided market problems such as buyer–seller discovery and supply–demand generation through proprietary services (e.g. analytics, marketing, customer outreach). However, even though centralized marketplaces serve a purpose, they have various limitations. From a socio-economic perspective, exorbitant dataset prices present a barrier-to-entry for data enthusiasts or early-stage entrepreneurs. From an organizational perspective, existing centralized data marketplaces focus on offering static datasets and their data offerings lack diversity—they focus on specific categories of data (e.g. financial, agricultural, geospatial). Most relevant to our work, existing centralized marketplaces do not focus on acquiring real-time data collected by robots or IoT devices. The closest type of data marketplaces related to our work are those that focus on selling aerial survey data collected by UAVs. From a general perspective, we can consider that centralized data marketplaces create data silos.

Although various "decentralized" data marketplaces, that leverage blockchain technology, have emerged, none has yet reached liquidity, and are predominately reliant on a centralized infrastructure that enables storage and streaming of data. The "decentralized" nature of these marketplaces is primarily attributed to the use of smart contracts running on a specific blockchain, and the use of a utility-specific token—e.g. a token for purchasing or storing data. Furthermore, governance of the data acquisition protocols is often left up to developers of the platform. Similarly, data pricing models are initially defined by the developers of the platform or are broadly based on open-market dynamics. With the increasing deployment of robots in "the wild", we imagine a near future where robots may not only populate the workplace, but also homes, and open environments. This would lead to an increasing production and collection of data that may be leveraged for various applications and purposes. Such data could generate monetary revenue for the respective robot owners or be offered freely. More interestingly, such revenue could go directly to independent robots that own cryptocurrency (crypto) wallets. This crypto revenue could be used to interact with decentralized applications (DApps) and to engage in peer-to-peer transactions with other robots. We further elaborate on this in [3], where we discuss human–robot interactions and robot-to-robot interactions that take place when robots can interact with smart contracts and use "money" without third-party mediation.

This paper presents AutomataDAO, a self-governed decentralized data marketplace for robot and IoT data where both human and robot agents can perform direct peer-to-peer data transactions. Robot agents or owners of robotic systems can register

to the DApp marketplace sharing details of the available data streams and schemas. A buyer can subscribe to a data stream through a DApp and engage in interactive control of the robot. If a semi-autonomous feature is available, the buyer can identify points of interest and set navigation waypoints. At each navigation waypoint, the buyer could validate the data through our DApp dashboard and continue to pay for the stream. Rather than performing continuous contract calls, we leverage state channels to accumulate the state of the interaction, accumulating payments at each waypoint upon verification of the data. To our knowledge, this application of state channels in the context of robotics and IoT systems is novel and it is a more optimal solution in comparison to work presented by existing literature related to IoT-blockchain-based data marketplaces. Existing literature proposes systems that continuously interact with the blockchain—making the entire system unsustainable and costly. The latter could be attributed to the fact that most literature leverages private blockchains for proof-of-concepts, rather than existing public blockchain, disregarding transaction fees, and throughput/latency limitations. In our system implementation, the contracts are deployed on Ethermint, which makes the execution of our smart contracts more economical and faster than on the Ethereum network. The latter further creates opportunities to interact with other existing blockchains in the Cosmos ecosystem that may provide additional decentralized data processing capabilities.

The paper is organized as follows: Sect. 2 reviews existing literature on data IoT marketplaces and decentralized data marketplaces. Section 3 discusses our decentalized autonomous organization (DAO) smart contracts. Sections 4 and 5 briefly present the pricing models explored in this work and selected use cases for the system. Section 6 presents a preliminary discussion of our system architecture. Section 7 presents our architecture. Section 8 provides the implementation details of our system. Section 9 discusses the limitation and future work of our research. Section 10 concludes our paper.

2 Background Work

The confluence of robotics, IoT and blockchain has recently become popular in academic research and in commercial applications. While numerous companies are attempting to build blockchain-based data marketplaces, a truly efficient decentralized solution remains elusive. Most projects are either built with inherently centralized infrastructure or are not scalable for real-world use. These issues have recently crossed into the realm of academic research. A number of academic papers have been written proposing systems that rely on the IOTA platform, which is inherently centralized due to its coordinator-based architecture [17], but, promotes itself as a decentralized IoT blockchain [35]. In other literature,“enterprise blockchains” or “consortium distributed ledgers” such as Hyperledger [19] and Quorum [8] are often used. Contrary to public blockchains, these are explicitly closed invite-only systems. Hyperledger and Quorum were, respectively, founded by IBM and JP Morgan, and later open-sourced. Similarly, a vast literature has proposed IoT systems and data

marketplaces that use public Ethereum—but, the end-work presented in the literature uses private network instances of Ethereum to implement proof-of-concepts. Such work does not account for Ethereum's scalability issues and expensive transaction fees, which have recently skyrocketed since the time of the published literature. On the other hand, work bridging the world of robotics and blockchain is still in its infancy. Some work simply proposes using a blockchain for decision-making or consensus in distributed robotics, others discuss the well-known benefits of blockchain and IoT, but in the context of robotic systems. Some recent work explores the application of blockchains, smart contracts and cryptocurrencies as it relates to human–robot interaction (HRI). We further elaborate in the following sections.

2.1 On Centralized and Decentralized Data Marketplaces

A wide range of traditional data marketplaces presently exist. These can be seen as industry-specific data silos, where data from industries such as health care, automotive, agriculture, finance or retail might end up. References [34, 37, 40] provide an academic overview on these type of data marketplaces, [9] provides an industry perspective on the opportunities for holistic data marketplaces. Centralized data marketplaces, like other centralized systems, have the benefits of seamlessly coordinating multiple bespoke parties through direct oversight and strict governance set by the owners of the marketplace.

The organizational structure and business mechanics vary across different centralized data marketplaces. But, in summary, these function as follows: (a) From a data provider's perspective: the provider contacts the marketplace operator(s) to inquire about selling data. The marketplace engages with the provider, collecting details about the provider, details about the data and evaluating the data. Then, the provider is allowed to set the terms and conditions for the use of such data. The marketplace then either collects/stores the dataset(s) in its infrastructure, or allows the provider to list the data on the marketplace's portal—simply acting as a broker. In some data marketplaces, part of the benefit is the operational and marketing services offered—e.g. identifying possible buyers, customer outreach and closing deals. (b) From the buyer's perspective, the buyer enrolls and may undergo due diligence, gains access to the marketplace portal, browses through available datasets and can often make a direct purchase. In some cases, e.g. for sensitive data, the buyer may need to contact the marketplace and undergo further due diligence.

An example beyond simple data brokerage is presented by AWS Data Exchange [38]. Direct infrastructure integration and access to millions of AWS customers are key advantages that allow the AWS Data Exchange to streamline data acquisition and delivery, and to minimize the buyer–seller discovery process. Similar to other data marketplaces, the AWS Data Exchange only allows "qualified" data providers to sell data, and provides flexibility over the terms and conditions of a transaction. Providers are able to publish free or paid products under specific terms, or even issue private offers and custom terms and conditions to specific AWS customers. Additionally,

similar to other traditional data marketplaces, providers can opt to approve each subscription based on the intended use case or regulatory compliance (e.g. GDPR) of the data subscriber.

Over the recent years, there has been an emergence of "decentralized" data marketplaces that aim to democratize data and allow monetization of data by the masses. These include projects with multi-million dollar marketc.aps such as Streamr [41], Ocean Protocol [15] and IOTA [20]—presently with a billion dollar marketc.ap, as well as early blockchain-based data marketplace platforms such as Datum [16] whose marketc.ap has dropped below the $1 million and development has begun to stagnate, as noted by GitHub commits. There are also various academic works that explore data marketplaces in the context of IoT, cf. e.g. [1, 31, 36, 43, 45]. Such academic literature and other present proof-of-concept systems that leverage either private network instances of Ethereum or consortium-distributed ledgers, and in particular propose the use of cryptographic receipts that represent data transactions.

For the most part, in publicly funded data marketplace projects, the basis of "decentralizatio" is solely based on their application of smart contracts. For example, Streamr relies on a set of Ethereum smart contracts for payments and data permission. Additionally, an ERC-20 standard token, is used for settlement and to incentivize a data transport network to provide enough bandwidth. The underlying means to transfer or stream the data are still structurally centralized, as nodes within the network are incentivized to run on cloud service providers such as AWS. On the other hand, Ocean Protocol runs a proof-of-authority network where their smart contracts are deployed. Unlike other decentralized blockchain data projects, Ocean Protocol provides a generalized and modular smart contract framework, and has implemented a DID smart contract framework called Keeper contracts. Datum simply leverages smart contracts to process the purchase of data—but, future road map milestones discuss the development of the Datum blockchain to overcome Ethereum's drawback, and the development of a storage network. Beyond projects focused on becoming data marketplaces, other multi-million marketc.ap IoT-focused blockchain projects, such as IoTeX [42], propose the implementation of data marketplaces that integrate into their blockchain.

2.2 IoT and Blockchain Technology

At its core, the concept of the Internet of Things (IoT) is about interconnecting everyday physical objects and allowing them to access internet services [30]. Since the early 1990s, there has been a growth in smart devices and networking technologies aimed at turning the vision of IoT into reality. But, although a lot of progress has been made, even to this day the Internet of Things is faced with various technical and social questions. Some of these include technical questions over secure communication between IoT devices, protocol standardization and requirements over standard identity and authentication solutions [18, 24]. There are also various questions related to IoT data. These include questions over data collection practices employed

by devices, the integrity of data as it is shared between devices and services, and hard guarantees over the provenance of data [12]. One approach towards improving data provenance is to allow an IoT device to have a decentralized identity (DID) and allowing it to sign its own transactions, as discussed in [23] and as discussed in the context of robotics in [3].

In [7], the authors outline five categories of use cases for IoT and blockchain: (1) data storage management, (2) trade of goods and data, (3) identity management, (4) rating systems, (5) other. Part of these use cases can be seen addressed in [33] that presents the design and implementation of an IoT network leveraging LoRA gateways and the concept of "smart proxies" or relay servers that communicate with the blockchain. Similar to other work, the latter uses a private Ethereum network for its proof-of-concept. For the most part, existing literature on IoT and blockchain remains pigeonholed on the application of private, permissioned or consortium-distributed ledgers, instead of the use of public blockchains. In general, the core reason behind this is the computational requirements of running full nodes or light clients on lo-power wide area (LPWA) networks that allow low-power IoT devices to connect and communicate efficiently with minimal power costs. Additionally, IoT-blockchain literature makes the same case against public blockchains, as other existing literature—throughput/latency limitations, and volatile transaction costs. In this work, we address this by leveraging state channel technology and minimizing the number of on-chain calls.

2.3 Decentralized IoT Data Marketplaces

An extension of the latter ideas is to allow data produced by IoT devices to be sold in an open "decentralized" marketplace. Existing literature such as [31, 45] highlight the application of smart contracts for access management and micropayments. Similar to the literature mentioned prior, the actual systems have been implemented using private networks. This is once again, is due to some of the constraints of existing public blockchains such as (1) transaction throughput, (2) block finality and (3) transaction costs.

To elaborate, as of this writing, the Ethereum blockchain supports 15 transactions per second (TPS) and its proof-of-work consensus leads to probabilistic finality—on average it is recommended to use 20–25 block confirmations to prevent a double-spend attack. Additionally, transaction fees (gas) have increased from 9.898 to 480.10 Gwei, or approximately 48.50 times, from January 16 to September 2. Although other public blockchains exist, our work leverages Etheremint, an Ethereum Virtual Machine implementation built with Tendermint consensus [25]. The use of Ethermint allows us to seamlessly deploy Solidity smart contracts and leverage existing Ethereum tooling. Prior work explored the sole use of the public Ethereum network as it is one of the most established public blockchains, and it's underlying technology more decentralized than others, e.g. EOS [44].

Other work such as [1] not only focuses on recording payment transactions on the blockchain, but also on using the blockchain to store data offerings through a smart contract. This approach is unsustainable given the volatility of transaction fees, and the requirement to store data offerings on-chain—considering that offering information is subject to change according to the growth of data sources. Other efforts such as [32] present data marketplaces that focus on on-demand crowdsensing—whereby a buyer can make on-demand requests for a specific type of data and the "crowds" (equipped with IoT/smart devices) could collaboratively respond to data requests. In comparison to other academic literature, the latter proposes running a network of market operators that maintain the notion of a marketplace and interact with data consumers—no further implementation details are provided. Work such as [31] and [45] develop data marketplace on IOTA and present the same ideas from IoT-blockchain work discussed prior.

2.4 Robots and Blockchain Technology

As part of the Internet of Things, mobile robotic systems present ample opportunities for the collection and application of data. Current literature explores the application of blockchain technology across different verticals in robotics. In [3], the authors present a holistic view into using smart contracts to create cypherphysical collaborative games between humans and robots, and to assess the novel interactions afforded when a robot can make unmediated physical payments and enter into agreements with humans. Overall, this human–robot interaction perspective proposes studying the social and psychological impact of robot–human peer-to-peer financial transactions and agreements. Other work, such as [5] proposes the use of a blockchain as a means to coordinate a swarm of robots through decision-making/voting taking place on the blockchain. Similarly, [29] propose a similar concept except that it focuses on storing robot events on a Tezos-based consortium blockchain. Other work such as [39] and [22] proposes a utility-token-based protocol to manage robot activity. This type of work on "coordinating" various agents and managing the state of an "interaction" can be seen as similar to previous IoT-related work, except that rather than coordinating thousands of devices the authors make the premise of using a blockchain to coordinate a swarm of robots. In the case of managing interactions, the authors' goals are to record data transactions or record control of the robot on a blockchain. The latter can be compared to record the interactions that take place between IoT devices and internet services.

Overall, the use of a fully fledged proof-of-work blockchain, or even a private blockchain, can be seen as an overkill for such a task of coordinate multiple agents. A simpler approach could be taken by solely leveraging modular consensus protocols such as Tendermint [25], which is the core consensus protocol in Cosmos [26]. Another argument against the plain and simple use of a blockchain for direct robot coordination and communication is that continuous communication via smart contract calls is often unnecessary and costly if ran on a public ledger. A different

and more efficient approach would be to use state channels, as presented in this paper. Other ongoing works such as [13] can be seen as also applying the lessons and concepts from IoT-blockchain research. In particular, the paper uses a blockchain to generate and store cryptographic receipts of data transactions—but, instead of the receipts being linked to IoT data access, reads and writes it focus on linking the cryptographic receipts to data collected/stored during robot therapy—hence achieving the properties of security and auditabilty.

2.5 State Channels and Micropayments

A state channel is a blockchain scaling solution and generalization of what is known as a payment channel, this concept was presented in [21]. It came about due to the lack of network throughput and high transactional fees. The fundamental idea is that multiple payments between two users can occur sequentially outside of a blockchain ledger (off-chain) until a given deadline when the resulting balance updates are committed to a digital ledger. These individual payments, often called micropayments, are not recorded in the given blockchain; instead, only the final balance update is recorded in the blockchain at the end of the interaction. In essence, the concept of a state channel allows for a single transaction to process a whole series of payments, and for these individual micropayments to occur nearly instantaneously—given the underlying communication scheme later discussed in Sects. 7.1 and 8.1.

A payment channel between two parties is initialized by a deposit of funds from at least one party into a smart contract to be held in escrow for a given time period. This initial balance can be sent from one party to another. Balance updates are determined by the digital signatures of the two parties involved. The party making the payments provides the receiving party a series of signed messages, one for each incremental balance update between them. After a given amount of time, the channel will end and the receiving party will submit the latest signed balance from the sending party along with their own signature of the final balance. This will trigger a payment to the second party from a smart contract that had been holding funds in escrow. If consensus between parties is not reached by the deadline the original escrow is returned.

The efficiency of state channels can be expanded by the use of network routing and smart contracts to create what are known as payment networks. Payments can be routed through intermediary connections so that two users who have not established a state channel connection can still send offline payments to each other. This significantly reduces network fees and increases transaction speeds. Payments can be secured through the use of a series of hash lock time contracts [21] between parties which ensure intermediaries are incentivized to relay funds to the end user by locking fees until the transfer is finished. Another routing method known as virtual channels has been proposed by [11], which claims to increase network efficiency by eliminating the need for direct payment relays.

This basic methodology has been advanced to applications more complex than simply payments. Rather than merely updating payment balances users can make agreements on the state of "generalized smart contracts". These smart contracts essentially function outside of the primary blockchain the state channel is built for. Implementations of generalized state channels are currently being developed by Celer Network [14] and state channels [6].

2.6 Decentralized Autonomous Organizations (DAO)

The concept of a Decentralized Autonomous Organization (DAO) is presented as a superset of the concept of a Decentralized Autonomous Corporation (DAC) in [2]. A DAC, originally being a term coined by Larimer in 2013 [28]. They have traditionally been used to allocate funding by democratic consensus through a series of proposals and voting rounds. Voting is restricted to DAO participants. Token ownership and reputation scores are the two primary mechanisms through which voting access is mediated. Overall, a DAO is implemented as a set of smart contracts that hold the governance logic and token distribution information.

3 Marketplace DAOs and Collective Robot Ownership

We propose the implementation of a marketplace for robot data as a decentralized autonomous organization (DAO), where both robot and human agents can participate in governance. Nearly, all aspects of robot operations can be determined by group consensus. Proposals and votes by DAO participants can determine where a robot operates, which services it provides, and the fees it charges to data consumers. Total marketplace revenue for any number of robots can be split among participants allowing for collective ownership and investment. Our marketplace smart contracts allow for DAOs to be integrated in a modular manner for any robot provider. This permits unique governance structures for each robot or group of robots. This architecture is explained in Sect. 7, however, the implementation details of the possible DAO smart contracts are beyond the scope of this paper, more details can be found in [27].

4 Pricing Model

We focus on two specific types of pricing models: (1) time-dependent pricing and (2) volume-dependent pricing. Time-dependent pricing is compatible with our smart contract/state channel framework as it assumes that a real-time or close to real-time data stream is more valuable if it is continuous. For example, if a mobile robot is surveying foot traffic at a shopping centre having gaps in the data stream would

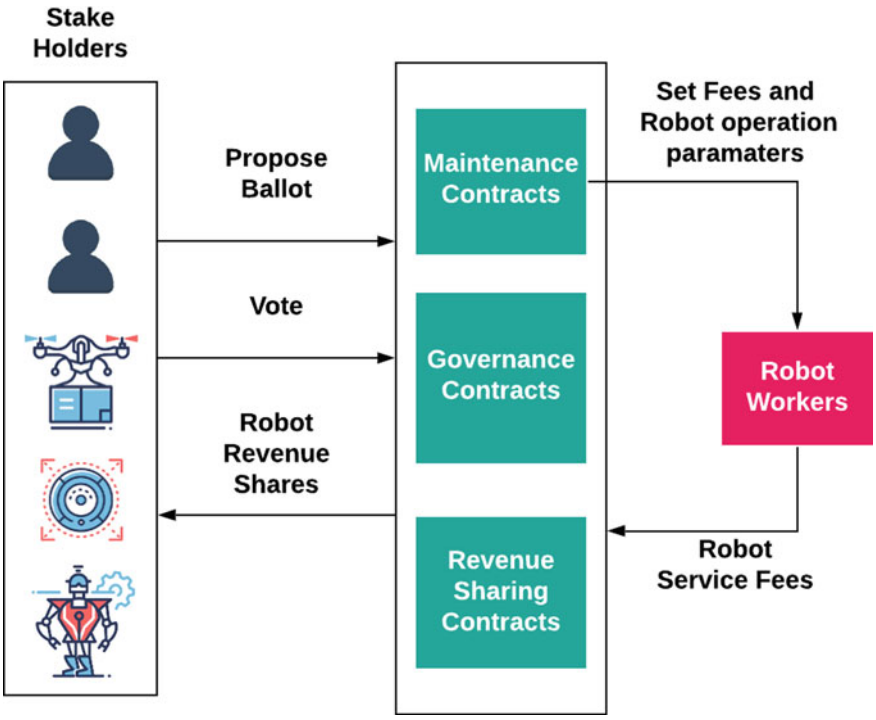

Fig. 1 DAO interaction overview

be undesirable. On the contrary, volume-dependent pricing assumes that the buyer is comfortable with gaps of missing data during the stream and overall pays for a specific quantity of ingested data.

5 Interaction Use Cases

Below we discuss a set of use cases supported by our DAO and data marketplace smart contracts. In particular, we consider two types of control/ownership models for robotic systems. The first one is an independent control/independent ownership model. The second one is a master–slave control/collective ownership model. In the former, we assume that robots are independent self-sufficient agents with a high degree of autonomy. In this configuration, a robot possesses a wallet, has sufficient computational and storage capacity and can interact with the digital world on its own. The latter model assumes that multiple robots are controlled or owned by a single entity. This is similar to a master–slave model or a Robot-as-a-Service (RaaS) business model, where a service provider may hire multiple independent robots for data collection tasks. In this configuration, the service provider (master) coordinates a set

of robots and signs any blockchain transactions related to payments or governance of the DAO. In a real-world application, we expect that the second configuration would be most suitable since the service provider can scale the underlying infrastructure to achieve desirable system properties (e.g. low-latency, high-throughput data streaming).

5.1 Offloading Data Collection Capacity

Similar to traffic offloading in network routing, independent robots may need to offload the performance of a data collection service to other robots. This could be due to various reasons such as form factor constraints that may limit the robot to adequately navigate a complex environment, or perhaps an anticipated component failure (e.g. battery failure). In such a scenario, a robot can transfer the `performance liability` to another robot that is capable of carrying out the data collection service. In practice, this use case can be carried by networked robots that can share navigation plans, and would require a robot to share state representation of the service being carried out. In future scenarios, we can imagine the implementation of `performance bonds`, either as a collateral deposit made by the robot or central service provider, or a smart-contract-based surety bond.

5.2 Proxied Data Collection Using Master–Slave Control Model

In this interaction, a central entity (either human or robotic) acts as a service provider that handles all data and payment transactions on behalf of the actual robot performing the data collection tasks. This sort of interaction could take place when we consider robotic systems that are constrained to local networks, or robots that are present on a rental basis—i.e. Robot-as-a-Service. From an implementation perspective, the service provider relays all data transactions and interacts directly with a state channel to receive payment—this is contrary to a robot directly streaming the data to a consumer and directly receiving payments. In a Robot-as-a-Service model where the service provider rents or hires independent robots, payments are distributed dynamically through the smart contracts.

5.3 Collaborative Data Collection by Independent Agents

In a more interesting scenario, a data buyer may purchase a data stream from a single robot and the robot may opt to collaborate with other robots and share the

revenue from selling such data stream. This is applicable to scenarios where a single drone may be tasked with surveying an environment and may request other drones to collaborate in the data collection task.

6 System Preliminaries

Figure 2 presents a high-level architecture of our system. The data and communication pipeline is based on two assumptions: (1) Robots may be resource constrained. Meaning that not all robots may support high bandwidth data transactions or have extensive computational power available. (2) Robots may be constrained to local networks. This may be due to security requirements or limitations on software architecture.

6.1 Robot Constraints

The first assumption is similar to that in the IoT world. In robots, the form factor may constrain introducing additional hardware. Additionally, a robot's internal computational or network requirements may limit the robot from sharing such resources to process or stream data. An example of this is a drone (UAV) equipped with a set of cameras performing Visual Simultaneous Localization and Mapping (SLAM). The goal of SLAM is to continuously estimate the position of the robot in 3D space while constructing a 3D map during navigation. There are many ways to perform SLAM, but in Visual SLAM the robot relies solely on cameras. Although, this is the preferred method for UAVs, processing camera feeds requires extensive computations which in turn increase the amount of power needed. In turn, performing additional computations can increase power demands, which would then require additional batteries that add weight to the vehicle and offset its performance. Hence, UAVs and other similar robots are resource constrained. A similar constrained can be considered for affordable commercial robots that are not equipped with high-end processors, but which instead may run computations on an edge computing device. The second assumption considers that present commercial robotic systems do not always connect to the Internet. This could due to limited functionality or privacy and security measurements. Instead, robots are connected to local networks—some leveraging a popular open-source framework and middleware called Robot Operating System (ROS). It is worthy to note that even when some robots are solely connected to a local network, they may be subject to attacks—as discussed in [10]. This is often due to misconfigurations in the network and or known issues within the middleware leveraged by the robots.

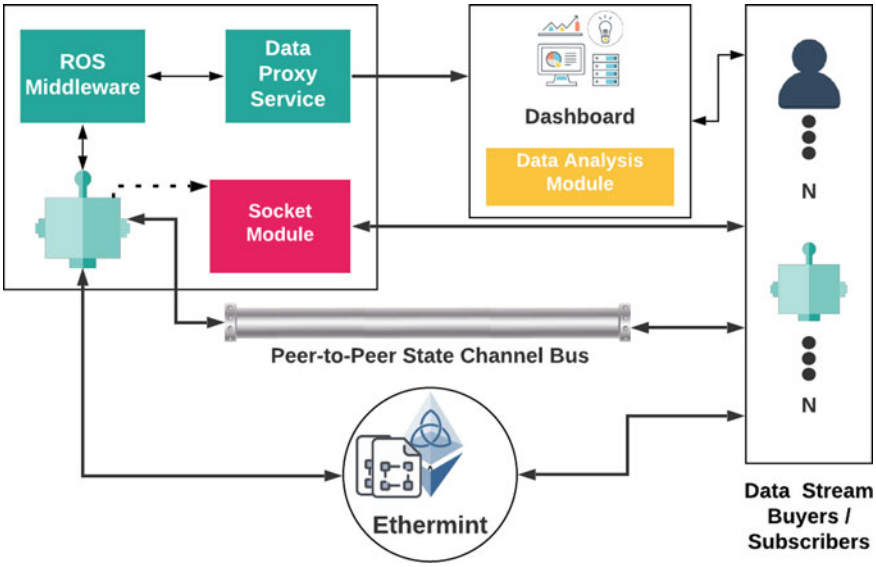

Fig. 2 High-level system architecture

6.2 Data Schemas, Evolution and Compatability

From the data buyer side, we assume that the buyer has agreed to a given data schema, which can be used as part of data validation checkpoints when the buyer receives the streaming data. This process is similar to the use of data schemas (e.g. Avro, ORC) in a traditional data pipeline of a distributed/big data system. In our present implementation, we simply define schemas in JSON format and share the schema prior. To elaborate, we do not make use of a centralized schema registry. Instead, the schema is initially shared between the data provider and the buyer prior to the purchase. During the state channel initialization step, a hash of the schema is recorded on-chain which serves as an identifier of the data and can also help in future work related to dispute resolution. In future work, we plan to integrate Avro or Protocol Buffers to serialize and deserialize the data being streamed.

7 Architecture

Figure 2 presents a high-level architecture of our system.

7.1 Data Marketplace Smart Contracts and State Channels

Robot owners register by providing a unique DID identifier, an address for state channel signatures and an address to receive payments. Each provider has a mapping that stores state channel initialization for each unique user address by using the user address and a unique agreement id as keys. This allows users to register for multiple services concurrently. Each service is represented as a struct with a bytes32 representing a hash of the service agreement details, a nonce which is necessary for state channel update security, a timeout for when the channel will close which is formatted in Unix time and a value which is the maximum payout that can be paid to the service provider.

The ERC-20 token standard will be utilized for payments. A token deposit must be made by the consumer before registration. This is done by calling the user deposit function. This transfers tokens into the marketplace smart contract and adds the equivalent value to a global spending balances mapping. This balance can be spent by the user on any service. Users can register for services with Providers by signing an agreement that initializes a state channel for service payment. The is done in the `registerForService` function which takes signatures from the provider signer address and the user, along with the initial service details. It verifies the signatures match the hash of the service details provided then commits the service to memory. The value associated with any registered service is locked and deducted from the global spending balance.

After a given series of transactions, a service state channel can be updated to send payment to the provider through the `commitServiceUpdate` function. This function takes signatures from the provider and user, and checks when they are valid signatures of the hash of a bytes32 `update_state` concatenated with a withdraw value. If this condition is valid payment is withdrawn and sent to the Provider receiver address. If a consensus is not reached, the user can wait until the timeout date is reached and withdraw their deposit by calling the `withdrawServiceDeposit` function passing the correct bytes32 identifier for that service agreement.

7.2 Data Streaming

Data management workflow for real-time/streaming data are based around the notion of data contracts, whereby one party can agree to the shape or structure of the data and other parties can development systems that ingest, process or analyse such data based on such structure. On the other hand, big data systems like Data Lakes are on ingesting both structure and unstructured data and applying the respective schema on-read. Prior to making a purchase, a buyer requests a data stream's schema. This is requested through the Data Proxy Service which queries a ROS topic that contains the data stream that will be sold. If the buyer approves the data schema and a purchase is made, the robot will be notified through a smart contract event. In the case of a robot

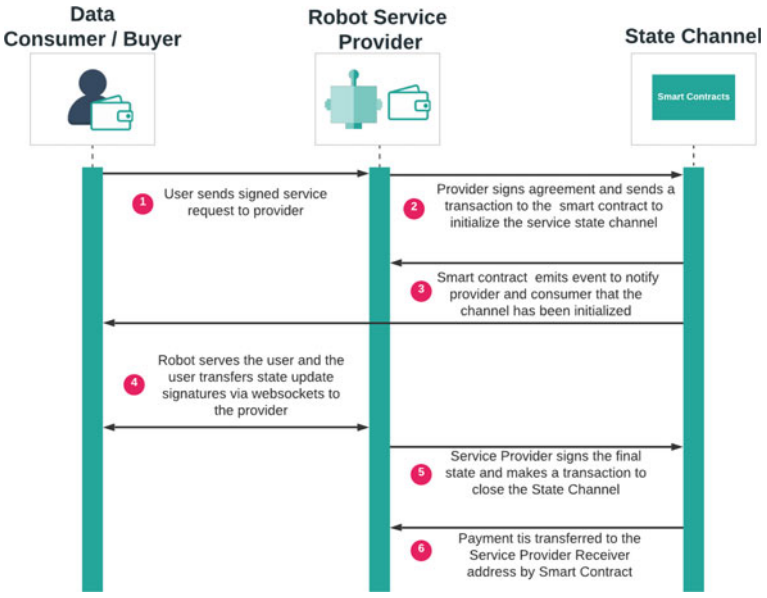

Fig. 3 Transaction sequence diagram

that has enough capacity to stream data, the data will be shared through a direct socket connection to the buyer. As the data flows through, the buyer uses a Data Analysis Module to verify the integrity of the data—this is a type of checkpoint performed at different times based on whether the buyer is purchasing a time-dependent data stream or a volume-dependent data stream. In either way, this resembles the same concept used in a big data pipeline where the producer and consumer of data can check the data against the schema both parties have agreed on. Section 8.3 elaborates on the implementation details. Listing 1.1 presents a sample data schema.

8 System Implementation

8.1 Smart Contracts and State Channels

The smart contracts are written in Solidity and deployed an Ethermint. Ethermint is an implementation of the Ethereum Virtual Machine (EVM) built on top of the Tendermint consensus engine. Multiple Ethermint nodes comprise an Ethermint Cosmos Zone—a proof-of-stake blockchain. Ethermint allows Solidity smart contracts to be deployed to this Zone and enables the use of Ethereum development tools. Appendix listing 1.2 presents an overview of the smart contract marketplace. The base marketplace contracts offer significant flexibility in terms of governance. The behaviour

of each robot marketplace provider can be governed by a separate DAO contract as the receiver address in the provider struct can be a contract. This allows a group of individuals to hold and distribute the robot service fees collectively. The DAO contract could also govern and change the signer address used to initialize service agreements. This allows for granular control over the possible services offered as all the service details are determined by the signer. In addition to lower transaction costs, the state channel implementation offers an important safeguard against fraud which can be prevalent in trust-less systems. By splitting services into micropayments, the potential loss for a provider given a non-paying user is only the amount of one sub-payment. After not receiving that payment the provider will cease the service to not lose anymore. Thus the smaller the individual sub-payments the less risk there is to the provider. This is true for consumers too. They are not obligated to pay until they received accurate data as they always have the option to wait until the state channel agreement times out. The state channel signature transfer is implemented using secure WebSockets over OpenSSL.

8.2 Robot and Smart Contract Communication

ROS middleware and packages were used internally to operate a Turtlebot 2 mobile robot, and a set of Parrot AR drones each equipped with a Raspberry PI that runs a ROS node with visual SLAM packages. The Turtlebot was used to simulate direct teleoperation and represented an independent robot. The computer operating the Turtlebot had an Intel Core i7 processor, with 16GB Memory, an NVIDIA GeForce RTX 2060 GPU, and 1TB SSD. This allowed the robot to: (1) support a light client, (2) run a web3 event listener that monitor smart contract calls and (3) stream data directly to a consumer. Details regarding data streaming, caching and the service proxy are discussed next. The drones were used to simulate a collective control model, where drones receive specific control commands from an edge computing device—a computer with the same specs as the Turtlebot's computer.

8.3 Data Streaming and Processing

Two system configurations were discussed at the beginning of Sect. 5: (1) independent control/ownership, and (2) collective control/ownership. In the first configuration, a robot with self-ownership or unilateral control should have the computational and storage capacity to process data. The second configuration mirrors that of a consumer-facing service provider that manages a single or multiple robot agents. As a service provider, this entity can deploy or control multiple robots to perform data collection/sensing tasks—the service provider holds the `performance liability` of the data collection tasks. There are two more granular interactions that can take place within this configuration: (a) independent robots can be employed by the consumer-

facing service provider, or (b) the service provider owns the robots. In the case of (a), the service provider is in charge of paying the robot(s) involved in a task, according. The distribution of the payments could be automated via the smart contracts where the payment is unlocked once the robot has completed a service task. In the case of (a), the service provider is the sole entity that receives the payment.

In the first configuration, we developed a `ROS Kafka adpater` and hosted a single broker Kafka cluster. This allows the robot to collect ROS messages in Kafka, which can then be sent to the data buyer through Kafka Proxy Service hosted on the robot itself. We use Kafka for three reason: (1) it is a lightweight messaging queue that could be used as temporary storage, (2) publish-subscribe communication protocol is similar to that of ROS' publish-subscribe node communication, (3) because it is distributed we can simulate a scenario of a two or three broker cluster where each broker is hosted on separate computers within the robot. Multi-computer robot architectures are often used in computational intensive robots, e.g. self-driving vehicles and humanoids. Furthermore, the use of Kafka allows us to perform schema validation prior to the Kafka producer publishing data. One key challenge which we explore in our future work is decentralization of a schema registry rather than relying on a centralized repository or instead of having the robot first send the schema to the data buyer prior to a purchase.

The data streaming implementation for the second configuration (collective/control ownership) is an extension of the implementation described for the first configuration. In essence, the service provider could host an edge computing device, where a single or multiple robots would send their data to. The edge computing device would then proxy the data to the data buyer. Since the service provider either manages a single or multiple robots they can all be connected to a local network and simply leverage ROS to publish the data collected as individual topic-based messages, which then the edge computing device would subscribe to and send to the data buyer. We provide further details on a robot data lake implementation in [4].

9 Limitations and Future Work

In the present implementation, a data stream's schema is shared by the robot prior to the purchase of a data stream. A different approach that was tested was the implementation of a schema registry. Having the robot share the schema adds an additional step to the interaction. On the other hand, schema registries are centralized. One way that a schema registry can be decentralized is by storing the schemas and metadata in a decentralized storage solution.

Presently, we use Ethermint for its value proposition against Ethereum. But, it is still under development and various features are still missing, e.g. it presently only supports an RPC endpoint and not WebSocket connection hence we are unable to directly subscribe to events. Another thing to consider is that, as a proof-of-stake blockchain, Ethermint requires active validators—presently Ethermint has not beeen deployed as a zone.

Another item to consider is that this work does not address open-market pricing models or considers an extensive reputation system. Future works explore such reputation system(s) and presents dispute resolution and/or arbitration mechanisms—i.e. when the buyer believes the integrity of the data does not meet certain requirements outside of the scope of the schema. Overall, open-market pricing models are complicated topics. Existing smart data pricing (SDP) literature makes assumptions about the data providers and consumers' identity, or assume that they are driven by monetary incentives. Literature is yet to address scenarios where fully anonymous or pseudo-anonymous data providers and data consumers are involved.

Further work on our state channel implementation is also required. We look towards Celer Network's [14] implementation and may leverage it in our future research. Lastly, research on decentralized governance models that allow both humans and robotic systems to partake in is necessary. As well as research on the tokenization of robotic systems, essentially turning them into fungible assets whose price is equivalent to its value production. Further references to this work can be found in [3].

10 Conclusion

We demonstrated how a truly trust-less marketplace can operate for robots within the limitations of current blockchain technology. We showed how state channels can be used to both minimize trust and transaction fees. We presented a simple modular smart contract framework as the backbone of this system allowing for the implementation of a number of different governance systems in robot ownership. We illustrated a few possibilities based on the architectures of previous decentralized autonomous systems.

The implementation is based on Solidity smart contracts deployed on an Ethermint network, this allows for higher transactional throughput and reduced transactional fees, and creates further opportunities to integrate into the Cosmos Network ecosystem and interact with other Zones (sidechains) that can provide additional services such as decentralized private compute. Streaming of data is performed peer-to-peer removing the need for a centralized broker, and two purchasing models are implemented. The first purchasing model is time dependent, whereby the data buyer/consumer only pays for the time it has consumed a data stream. This is implemented by validating the data at different temporal checkpoints and aggregating payments through a state channel. If the data is invalid, no payment is made to the robot/service provider after the invalid data checkpoint. This "pay-as-you-go" model safeguards the buyer against data streams, whose integrity has decreased purposely by a malicious actor (robot) or unintentionally—if a robot failure takes place. This is also useful when the continuity of a data stream is a core requirement The second purchasing model is volume based, whereby the data buyer purchases a stream and pays for a given total size of ingested data. This is also implemented using state channels, whereby the robot reports how much data is has produced and the buyer reports how

much it has consumed. Our smart contract framework also accounts for scenarios, where the robot may need to offload the service task to other robots, hence a payment for the data collection service may need to be split accordingly.

An initial implementation of AutomataDAO was presented at the 2020 ETH Denver conference and was awarded first prize in the Cosmos Network development track. Further implementation details, in particular, related to the DAO are discussed in [27] and in future papers.

Acknowledgements The authors thank Antoine De Vuyst and Brandon West for their contributions during the development of the initial implementation of AutomataDAO. As well as thank Cosmos Network for their award at the 2020 ETH Denver conference.

Appendix: Sample Data Schema

```
{
  "type": "record",
  "name": "weather",
  "fields": [
      {"name": "name",  "type": "string" }
    , {"name": "temperature", "type": "float" }
    , {"name": "timestamp", "type": "string", "default": null}
  ]
}
```

Listing 2.1 Sample Data Schema

Appendix: Smart Contract Overview

```
pragma solidity >=0.4.22 <0.7.0;

/**
 * @title Storage
 * @dev Store & retreive value in a variable
 */
contract Storage {
    mapping (uint => Provider) public Providers;
    public uint totalProvider;

    struct service {
        bytes32 state;
        uint nonce;
        uint timeout;
        uint value;
    };
    struct Provider {
        bytes32 DID;
        address receiver;
        address signer;
        mapping(address=>mapping(bytes32=>service)) ServiceStates;
    };
    /**
     * @dev Store value in variable
     * @param num value to store
     */
    function addProvider(
```

```
        bytes32 _DID
      , address _receiver
      , address _signer) { ... }

    /**

     */
    function registerForService(
        bytes32 service_sig0
      , bytes32 service_sig1
      , bytes32 intially_state
      , address _user
      , bytes32 provider
      , uint timeout
      , uint value ) { ... }

    /**

     */
    function commitServiceUpdate(
        bytes32 state_sig0
      , bytes32 state_sig1
      , bytes32 update_state
      , address _user
      , bytes32 provider
      , bytes32 service
      , uint  value) { ... }

    /**

     */
    function withdrawDeposit(
      bytes32 service ) { ... }
}
```

Listing 2.2 Smart Contract Interface

References

1. S. Bajoudah, C. Dong, and P. Missier. Toward a decentralized, trust-less marketplace for brokered iot data trading using blockchain. In *2019 IEEE International Conference on Blockchain (Blockchain)*, pages 339–346, July 2019
2. Vitalik Buterin. DAOs, DACs, DAs and More: An Incomplete Terminology Guide. https://blog.ethereum.org/2014/05/06/daos-dacs-das-and-more-an-incomplete-terminology-guide/
3. Irvin Steve Cardenas and Jong-Hoon Kim. Robonomics: The Study of Robot-Human Peer-to-Peer Financial Transactions and Agreements. In *Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '20, page 8–15, New York, NY, USA, 2020. Association for Computing Machinery
4. Irvin Steve Cardenas, Pradeep Kumar Paladugula, and Jong-Hoon Kim. Large Scale Distributed Data Processing for a Network of Humanoid Telepresence Robots. In *The 2020 IEEE International IOT, Electronics and Mechatronics Conference*. IEEE, 2020
5. Eduardo Castelló Ferrer. The blockchain: A new framework for robotic swarm systems. In Kohei Arai, Rahul Bhatia, and Supriya Kapoor, editors, *Proceedings of the Future Technologies Conference (FTC) 2018*, pages 1037–1058, Cham, 2019. Springer International Publishing
6. State Channels. Statechannels docs. https://docs.statechannels.org/
7. M. Conoscenti, A. Vetrò, and J. C. De Martin. Blockchain for the internet of things: A systematic literature review. In *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*, pages 1–6, Nov 2016

8. Consensys. Quorum. https://consensys.net/quorum/
9. Johannes Deichmann, Kersten Heineke, Thomas Reinbacher, and Dominik Wee. Creating a successful Internet of Things data marketplace. https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/creating-a-successful-internet-of-things-data-marketplace, Oct. 2016
10. N. DeMarinis, S. Tellex, V. P. Kemerlis, G. Konidaris, and R. Fonseca. Scanning the internet for ros: A view of security in robotics research. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8514–8521, 2019
11. Stefan Dziembowski, Lisa Eckey, Sebastian Faust, and Daniel Malinowski. Perun: Virtual payment hubs over cryptocurrencies. Cryptology ePrint Archive, Report 2017/635, 2017. https://eprint.iacr.org/2017/635
12. D. Fakhri and K. Mutijarsa. Secure iot communication using blockchain technology. In *2018 International Symposium on Electronics and Smart Devices (ISESD)*, pages 1–6, 2018
13. E. Ferrer, Ognjen Rudovic, T. Hardjono, and A. Pentland. Robochain: A secure data-sharing framework for human-robot interaction. *ArXiv*, abs/1802.04480, 2018
14. ScaleSphere Foundation Ltd. ("Foundation"). Celer Network: Bring Internet Scale to Every Blockchain. https://www.celer.network/doc/CelerNetwork-Whitepaper.pdf, June 2018
15. BigchainDB GmbH and Newton Circus (DEX Pte. Ltd.). Ocean Protocol: A Decentralized Substrate for AI Data & Services Technical Whitepaper. https://oceanprotocol.com/tech-whitepaper.pdf, Aril 2019
16. Roger Haenni. Datum Network: The decentralized data marketplace. https://datum.org/assets/Datum-WhitePaper.pdf, June 2017
17. Heilman Ethan, Narula Neha, Tanzer Garrett, Lovejoy James, Colavita Michael, Virza Madars, Dryja Tadge (2019) Cryptanalysis of Curl-P and Other Attacks on the IOTA Cryptocurrency. IACR Cryptol. ePrint Arch. 2019:344
18. Steve Huckle, Rituparna Bhattacharya, Martin White, and Natalia Beloff. Internet of things, blockchain and shared economy applications. *Procedia Computer Science*, 98:461 – 466, 2016. The 7th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2016)/The 6th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH-2016)/Affiliated Workshops
19. Hyperledger. Hyperledger Farbric. https://github.com/hyperledger/fabric/. Accessed: 2019-02-12
20. David Sønstebø (IOTA). IOTA Data Marketplace. https://blog.iota.org/iota-data-marketplace-cb6be463ac7f, November 2017
21. Thaddeus Dryja Joseph Poon. The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments . https://lightning.network/lightning-network-paper.pdf, January 2016
22. A. Kapitonov, S. Lonshakov, A. Krupenkin, and I. Berman. Blockchain-based protocol of autonomous business activity for multi-agent systems consisting of uavs. In *2017 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, pages 84–89, 2017
23. Yki Kortesniemi, Dmitrij Lagutin, Tommi Elo, Nikos Fotiou, and Roberto Nardone. Improving the privacy of iot with decentralised identifiers (dids). *J. Comput. Netw. Commun.*, 2019, January 2019
24. Kshetri N (2017) Can Blockchain Strengthen the Internet of Things? IEEE IT Professional 19(4):68–72
25. Jae Kwon. Tendermint: Consensus without Mining. https://tendermint.com/static/docs/tendermint.pdf, 2014
26. Jae Kwon and Ethan Buchnan. Cosmos: A Network of Distributed Ledgers. https://cosmos.network/cosmos-whitepaper.pdf, 2016
27. The Advanced Telerobotics Research Lab. AutomataDAO: A Decentralized Robot Data Marketplace. http://www.atr.cs.kent.edu/our-research/automatadao/
28. Dan Larimer. Bitshares as a bank—the origin of the dac. https://bytemaster.github.io/article/2014/12/20/BitShares-as-a-Bank/

29. V. Lopes, N. Pereira, and L. A. Alexandre. Robot workspace monitoring using a blockchain-based 3d vision approach. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2812–2820, 2019
30. Friedemann Mattern and Christian Floerkemeier. *From the Internet of Computers to the Internet of Things*, pages 242–259. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010
31. S. Musso, G. Perboli, M. Rosano, and A. Manfredi. A Decentralized Marketplace for M2M Economy for Smart Cities. In *2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pages 27–30, 2019
32. D. Nguyen and M. I. Ali. Enabling On-Demand Decentralized IoT Collectability Marketplace using Blockchain and Crowdsensing. In *2019 Global IoT Summit (GIoTS)*, pages 1–6, June 2019
33. Ozyilmaz KR, Yurdakul A (2019) Designing a blockchain-based iot with ethereum, swarm, and lora: The software solution to create high availability with minimal security risks. IEEE Consumer Electronics Magazine 8(2):28–34
34. Perera C, Liu CH, Jayawardena S (2015) The emerging internet of things marketplace from an industrial perspective: A survey. IEEE Transactions on Emerging Topics in Computing 3(4):585–598
35. Serguei Popov. The Tangle. https://assets.ctfassets.net/r1dr6vzfxhev/2t4uxvsIqk0EUau6g2sw0g/45eae33637ca92f85dd9f4a3a218e1ec/iota1_4_3.pdf, April 2018
36. G. S. Ramachandran, R. Radhakrishnan, and B. Krishnamachari. Towards a decentralized data marketplace for smart cities. In *2018 IEEE International Smart Cities Conference (ISC2)*, pages 1–8, 2018
37. Schomm Fabian, Stahl Florian, Vossen Gottfried (2013) Marketplaces for data: An initial survey. SIGMOD Rec. 42(1):15–26
38. Amazon Web Services. Introducing AWS Data Exchange. https://aws.amazon.com/about-aws/whats-new/2019/11/introducing-aws-data-exchange/, November 2019
39. Alexander Smirnov and Nikolay Teslya. Robot interaction through smart contract for blockchain-based coalition formation. In Paolo Chiabert, Abdelaziz Bouras, Frédéric Noël, and José Ríos, editors, *Product Lifecycle Management to Support Industry 4.0*, pages 611–620, Cham, 2018. Springer International Publishing
40. Stahl Florian, Schomm Fabian, Vossen Gottfried, Vomfell Lara (2016) A classification framework for data marketplaces. Vietnam J. of Computer Science 3(3):137–143
41. Streamr. Unstoppable Data for Unstoppable Apps: DATAcoin by Streamr. https://s3.amazonaws.com/streamr-public/streamr-datacoin-whitepaper-2017-07-25-v1_1.pdf, July 2017
42. IoTeX Team. IoTeX A Decentralized Network for Internet of Things Powered by a Privacy-Centric Blockchain. https://s3.amazonaws.com/web-iotex-static/home/IoTeX_Whitepaper_1.5_EN.pdf, July 2018
43. H. T. T. Truong, M. Almeida, G. Karame, and C. Soriente. Towards secure and decentralized sharing of iot data. In *2019 IEEE International Conference on Blockchain (Blockchain)*, pages 176–183, 2019
44. Brent Xu, Dhruv Luthra, Zak Cole, and Nate Blakely. EOS: An Architectural, Performance, and Economic Analysis. https://www.whiteblock.io/wp-content/uploads/2019/07/eos-test-report.pdf, July 2019
45. K. R. Özyilmaz, M. Doğan, and A. Yurdakul. IDMoB: IoT Data Marketplace on Blockchain. In *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, pages 11–19, 2018