nature reviews electrical engineering

https://doi.org/10.1038/s44287-024-00034-9

Perspective

Check for updates

Blockchain technology for mobile multi-robot systems

Marco Dorigo ✉, Alexandre Pacheco, Andreagiovanni Reina & Volker Strobel ✉

Abstract

Blockchain technology generates and maintains an immutable digital ledger that records transactions between agents interacting in a peer-to-peer network. Initially developed for financial transactions between human agents, the technology could also be used across a broader spectrum of applications, providing transparency, security and trust without the need for a central authority. In this Perspective, we discuss how blockchain technology can enhance mobile multi-robot systems. This enhancement includes ensuring that autonomous robotic agents adhere to applicable laws, are identifiable and accountable for their behaviour, are capable of identifying and neutralizing malfunctioning robots and can actively participate in economic transactions for the exchange of goods and services. Discussing the first applications, we highlight the open challenges and describe the research directions that could reshape the mobile multi-robot research field in the coming decades.

Sections

IRIDIA, Université Libre de Bruxelles, Brussels, Belgium. ✉e-mail: mdorigo@ulb.ac.be; volker.strobel@ulb.be

Key points

- Blockchain technology and smart contracts are a novel way to program distributed systems and can provide multi-robot systems with properties that will be fundamental for their real-world deployment.
- There are different possible ways to integrate blockchain technology into a mobile multi-robot system: the blockchain can be hosted by the robots or it can be hosted externally; hybrid solutions are also conceivable.
- Smart contracts can assist multi-robot systems by providing supervision, synchronized data storage capabilities and system-wide rules.
- The behaviour of robots can be recorded in the blockchain, which is a tamper-proof database that allows for online fault detection and offline auditing.
- Even though initial results are promising, the usage of blockchain technology in multi-robot systems needs substantial research before it can be successfully deployed.

Introduction

A mobile multi-robot system (Box 1) consists of a group of mobile autonomous robots that work together to solve a problem or perform a task[1]. Such multi-robot systems might be more efficient, robust and flexible than a single robot[2]. For example, mobile multi-robot systems can simultaneously cover and sense a large area[3], can prevent a single point of failure[4], can replace broken robots[5] and can reconfigure their shape in accordance with a required task[6]. Even though most examples of mobile multi-robot systems are still demonstrated in research environments[7,8] — with the notable exception of a few real-world implementations in warehouse automation[9] — it is believed that once hardware and control limitations, as well as economic constraints, are overcome, their deployment in the real world will become widespread[8,10,11]. However, there are security aspects that are often overlooked but that will be of paramount importance for successful deployment in the real world: we need to equip these systems with properties such as accountability of behaviour[12], mitigation of Byzantine faults[13] (Box 1), confidentiality about the mission[14] and compliance with the law[15] in order to protect the robots, their owners, the environment in which they operate and the humans with whom they interact[16].

Blockchain technology has been used to start addressing these issues[13,17–19]. Developed in 2008 to store transactions of the digital currency Bitcoin[20], blockchain technology (Box 2) secures consensus on a decentralized ledger without the need for a trusted third party, such as a bank. A decentralized blockchain network maintains information that can be trusted even if the participating agents do not trust each other. Following the release of Bitcoin, the blockchain framework Ethereum was developed to support smart contracts (Box 2), which are tamper-proof and Turing-complete programs that are executed by each node of the blockchain network[21]. Even though blockchain technology was originally designed for establishing digital currencies, it holds great potential for integration into other systems, thanks to its decentralized character and fault tolerance and the availability of smart contracts.

In this Perspective, we first summarize the state of the art of blockchain-based mobile multi-robot systems[22–24]. As of the beginning of 2024, only preliminary proofs of concept for coordinating and securing multi-robot systems via smart contracts have been reported. We highlight the main challenges that need to be overcome before blockchain-based multi-robot systems can be used in real-world situations. We then discuss how blockchain not only provides new opportunities for the solution of the above-mentioned security-related issues but could also enable novel ways to organize the activities of the robots.

Blockchain-based mobile multi-robot systems

Blockchain technology can be integrated into mobile multi-robot systems using different architectures that let the robots manage the blockchain activities with various degrees of autonomy (Fig. 1). The maximum degree of autonomy is provided by Architecture 1: the blockchain is maintained exclusively by the multi-robot system. Each robot produces, broadcasts and validates the contents of new blocks, and maintains a local copy of the blockchain. This architecture is suitable for fully autonomous mobile multi-robot systems that do not require or permit any external interaction after deployment. Architecture 2 provides the minimum degree of autonomy: the mobile multi-robot system interacts with an external blockchain, for example, a publicly hosted blockchain. This architecture could be chosen if a reliable connection between the robots and an external infrastructure is available, but independence from a single controlling authority is desired. In between these two ends of the spectrum, there are potential hybrid architectures; for example, Architecture 3, a hybrid architecture where the multi-robot system hosts and maintains an internal blockchain (often called a sidechain[25]) which synchronizes relevant information with an externally hosted blockchain (mainchain) when possible. Architecture 3 could be chosen if only intermittent connections to an external infrastructure are possible.

In the following, we limit the discussion of the state of the art to Architecture 1 and Architecture 2 as there is currently no research that discusses Architecture 3.

Architecture 1

Some of the first attempts to use blockchain technology in mobile multi-robot systems have studied the architecture in which the blockchain was maintained by the robots themselves (Fig. 1, Architecture 1). The initial goal was to demonstrate that security issues in multi-robot systems (in particular, in robot swarms with only local communication capabilities) could be handled using smart contracts. Securing robots in a robot swarm via a smart contract was first experimentally demonstrated in 2018 with a simulated robot swarm that maintained an internal Ethereum blockchain, where each robot was a blockchain node[13]. A smart contract enabled the robot swarm to detect inconsistencies in robots' behaviours (some of the robots were Byzantine), demonstrating how blockchain technology could add a security layer on top of existing swarm robotics algorithms in a binary decision-making scenario. The work was then extended to demonstrate the collective estimation of an environmental feature: a reputation management system was implemented in a smart contract by assigning a trust value to the robots in the swarm, over time neutralizing the misleading estimates of Byzantine robots[26]. A comparison between a robot swarm controlled by a smart contract and one using consensus protocols available in the literature showed that smart contracts could identify and exclude Byzantine robots, whereas the existing consensus protocols failed[27]. Importantly, the robot swarms controlled by the smart contracts were also resilient to Sybil attacks[27] — attacks in which a small minority of robots forge many fake identities to try and gain control over the robot swarm.

The first implementations on real robots demonstrated the practicality of operating blockchain software within a swarm of physical robots (considering their computing power, random-access memory and storage capabilities)[17,28]. Exploiting the tamper-proof crypto tokens offered by blockchain technology enabled a blockchain-based token economy in the robot swarms. Robots could earn crypto tokens by sending information that was judged useful by the smart contract, and would lose crypto tokens otherwise. This design ensured that the number of crypto tokens owned by Byzantine robots would decrease over time, making it impossible for them to continue participating in the token economy[17].

Besides securing robot swarms, smart contracts can also coordinate and supervise the actions of individual robots in multi-robot systems[29], aggregating information gathered by the individual robots, and then performing group-level decisions that improve the performance and efficiency of the entire system.

Architecture 2

Architecture 2 is employed when mobile multi-robot systems exploit a blockchain that is hosted by an external infrastructure (Fig. 1, Architecture 2). This is an interesting approach when a stable connection between the robots and the external blockchain nodes can be established or when the data stored on the blockchain need to be accessed by other parties during the robots' operation.

Similar to Architecture 1, Architecture 2 can be employed to increase the security of a mobile multi-robot system. For example, Byzantine robots, when used in a leader–follower formation, can temporarily misguide their peers. This misguidance, however, can eventually be detected and undone by analysing the transaction history on the external blockchain[30]. An external blockchain can also be employed to improve data integrity and protection against malicious attacks[31].

Smart contracts residing on an external blockchain have also been used for path planning in multi-robot systems: when all robots store their planned paths on the blockchain, they can detect whether their path would lead to collisions with other robots and adapt the path accordingly. This path planning is enabled by the blockchain's shared data storage. Performing the collision detection can be done using an off-chain planner[32] or entirely on a smart contract[18]. In 2021, a first proof of concept demonstrated the use of a reward system, regulated by a smart contract, to incentivize surveying points of interest by multiple unmanned aerial vehicles. A distributed ledger, originally developed for the Internet of Things, that employs a directed acyclic graph (DAG) in its architecture was used[18]. In a similar example, smart contracts were employed to assign different roles to robots (for example, the role of worker or the role of distributing crypto tokens as a reward) to incentivize the completion of a collaborative task[19]. This work was then extended by showing how robots in a heterogeneous multi-robot system can allocate tasks (such as object retrieval or object transportation) in a warehouse application using smart contracts[33,34].

External blockchains have also been proposed as a means for data-sharing between humans, robots and organizations, for example, to store personal data of patients and protect their privacy when a robot needs access to the data[35]. Motivated by the COVID-19 crisis, a framework based on the combination of blockchain and multi-robot systems was proposed for battling pandemics through spotting lockdown violations or delivering medication[36].

Smart contracts residing on an external blockchain can interface with robots, building the basis for human to robot economic transactions and robots-as-a-service applications[37]. The Autonomous Intelligent Robot Agent (AIRA) project proposes proof-of-concept software for such applications, exploiting smart contracts on an external Ethereum blockchain to hire multi-robot systems composed of unmanned aerial vehicles[38].

Box 1

Mobile multi-robot systems, robot swarms and Byzantine robots

A mobile multi-robot system is a robotic system composed of two or more mobile robots that communicate and coordinate to perform a task[1,2]. When the number of robots is high and emphasis is put on distributed control and/or self-organization, the mobile multi-robot system is typically called a robot swarm[81,82]. Communication between the robots can take different forms — such as wireless (for example, using Wi-Fi), visual (for example, by flashing LEDs or performing specific movements) or situated (for example, using range and bearing)[83]. Coordination can be achieved by exploiting a central control unit that has knowledge about all the robots and tells each of them how to behave, or, as is typically done in robot swarms, by exploiting self-organization where each robot directly interacts with its neighbours and no particular robot in the swarm is in charge to control the system[2].

It is anticipated that, in the near future, mobile multi-robot systems will be more and more present in our lives to support people and industries in their daily activities[10,11]. In particular, mobile multi-robot systems might enable the efficient execution of activities — such as environmental monitoring, waste collection (including the cleaning of oceans), underwater exploration and farming — that could help in the necessary transition towards a sustainable and liveable future. Mobile multi-robot systems might also provide important support to mitigate black swan events such as nuclear disasters, earthquakes and terrorist attacks: for example, by performing search and rescue missions or by measuring the amount of toxins in the air[8,84]. In particular, we envision that mobile multi-robot systems will surpass human performance in some of these activities, and therefore perform them better and more efficiently while ensuring human safety.

However, it is expected that robots will not always behave as expected in such real-world deployments. The term Byzantine robot[13], inspired by the Byzantine generals problem[85], refers to a robot that shows a discrepancy between its specified behaviour and its actual behaviour. The discrepancy, also called a Byzantine fault, can be due to issues such as programming errors, failed hardware components and malicious attacks[86,87]. In the absence of adequate protocols, the behaviour of a single Byzantine robot can negatively influence other robots, leading to a complete failure of the multi-robot system[27].

Box 2

Blockchain technology and smart contracts

Blockchain foundations

Blockchain technology enables agents to agree on who owns crypto tokens in decentralized peer-to-peer networks, where crypto tokens are digital tokens that can represent values or ownership rights and are often used as a form of digital currency (cryptocurrency)[20]. Before the blockchain innovation, digital tokens could be easily copied, and thus spent more than once in multiple transactions (called double-spending attack): without a trusted third party, it is difficult to tell in such a case which transaction came first and is therefore valid[88]. Satoshi Nakamoto was the first to solve this problem by engineering a mechanism (now called Nakamoto consensus) that employs a decentralized data structure — a blockchain — to act as a digital ledger to record transactions of a cryptocurrency called Bitcoin. From a technical point of view, a blockchain consists of data blocks, each containing a list of transactions and a pointer to the previous block. The pointer is a cryptographic hash of the previous block; therefore, any attempt to change information in a block *i* invalidates any block *j* with $j > i$, thus breaking the blockchain. This feature leads to blockchain data integrity.

How blockchains work

The functioning of the Bitcoin blockchain — and its underlying Nakamoto consensus mechanism — can be summarized in three steps (see the figure):

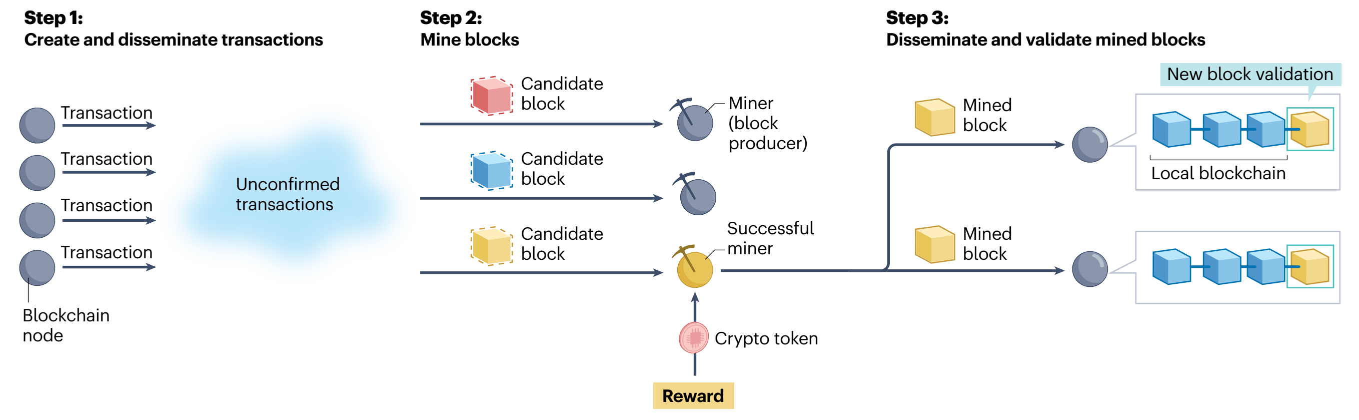
- Step 1: when agents participating in the blockchain network, called blockchain nodes, intend to transfer units of a cryptocurrency, they create transactions and disseminate them in the peer-to-peer network. Other blockchain nodes then keep these transactions in their pool of unconfirmed transactions.
- Step 2: to be confirmed, and thus made permanent, these transactions need to be included in a block of the blockchain. New blocks are generated by a certain type of blockchain nodes, called miners, that select a subset of the unconfirmed transactions and spend computational power to ‘mine’ a block — in practice, miners use their computational power to solve a puzzle that can only be solved by brute computational force.
- Step 3: as soon as a miner solves the puzzle, it disseminates the block in the network. To motivate miners to spend computational power, those miners that succeed in adding a new block to the blockchain receive a reward in the form of crypto tokens. Blockchain nodes receiving the new block validate it and append it to their local blockchain. In case of conflicting blockchain versions, Nakamoto proposed a consensus mechanism, based on proof of work[89], that allows blockchain nodes to agree on a particular blockchain version: the blockchain version with the highest amount of accumulated computational work is considered the correct version. This mechanism protects the blockchain from double-spending attacks, as an attacker would need to acquire more than 50% of the computational power available in the network to create an alternative chain.

As proof of work is computationally very demanding, alternative consensus algorithms that implement a similar logic were developed. For example, proof of stake (PoS consensus mechanism), the protocol currently used by Ethereum, selects block producers based on the amount of crypto tokens they own, and proof of authority (PoA consensus mechanism) selects the block producers based on their identity in a round-robin fashion.

Smart contracts

A smart contract is programming code that resides on the blockchain and that is executed by all the participants of the blockchain network, and therefore cannot be stopped or manipulated.

Originally, a transaction was used to transfer units of the cryptocurrency Bitcoin. However, it was soon realized that transactions could be used to store and transfer other kinds of data, such as Internet domain names, documents or images. In particular, it was found that transactions can be used for storing programming code and for sending arguments to functions, in this way enabling Turing-complete smart contracts (first implemented in the Ethereum framework[21]). By using smart contracts, the blockchain nodes can reach an agreement on the precise program outputs based on given inputs. Once the nodes agree on the execution of the program, it cannot be refuted or reverted.



Challenges for blockchain-based mobile multi-robot systems

Blockchains were originally created to operate in networks of computers. Their use in mobile multi-robot systems requires tackling challenges caused by the mobility of the robots and by hardware constraints. In the following, we identify and discuss four main challenges that should be overcome to enable the deployment of blockchain-based mobile multi-robot systems.

Computation, storage and communication requirements

The computing power, storage and communication capabilities of robots used in mobile multi-robot systems are typically more limited than in the dedicated stationary computers employed in traditional blockchain networks – making it challenging to use computing-intensive consensus protocols such as proof of work (Box 2). Alternative consensus mechanisms have therefore been explored. An example is given by Raft[39] (available in the Hyperledger Fabric blockchain framework[40]), which is a low-cost consensus algorithm that, however, only works under the assumption that the nodes are non-Byzantine[41]. Alternatively, permissioned consensus protocols, such as proof of authority, can be employed in Ethereum networks, as demonstrated on a group of 10 physical robots[28] – later extended to 24 physical and 120 simulated robots[17] – with limited hardware capabilities.

Although permissioned protocols enable secure and cost-effective consensus, the delays introduced by the blockchain processes could still be too high to be useful for typical applications in mobile multi-robot systems. Even though the block period (that is, the time between two consecutive blockchain blocks) can be shortened, its reduction increases the costs in terms of data storage and bandwidth, as blocks are generated more frequently and the blockchain synchronization process becomes more demanding[29]. Therefore, a trade-off exists between costs and delays, leading us to expect that most blockchain applications in multi-robot systems will be reserved for high-level decision-making and for security-critical applications rather than for lower-level control of individual robots.

The implementation of social consensus mechanisms based on trust graphs (see, for example, the Stellar Consensus Protocol), where the robots that are most trustworthy and active in message exchanges are trusted to produce blocks, is an alternative to the introduction of permissioned consensus protocols. The Decentralized Blocklist Protocol[42] establishes a system in which robots can levy accusations of misconduct against their peers, potentially leading to diminished trust and loss of privileges for the accused robots (in the study, the simulations were performed with up to 100 robots). Another, as yet unexplored, alternative could be the extension of the concept of proof of useful work[43] to a proof-of-physical-work consensus protocol[13] where robots can add information to a blockchain only if they first perform some physical work, such as transporting an object – thus linking the physical world with blockchain mechanisms.

In terms of storage, every blockchain node (Box 2) needs to keep a copy of the blockchain. Although this might not be a problem for Architecture 2 that uses an external blockchain, robots in Architecture 1 and Architecture 3 need to consider storage limitations. Having intermittent access to external infrastructure would enable uploading old blocks to an external infrastructure and keeping a trimmed blockchain with the most recent blocks. Storage and communication requirements can also be reduced by storing the hash values of large data files (for example, videos or maps) on the blockchain. These hash values can be used as unique identifiers, and the larger files can be shared on request through off-chain exchanges.

Regarding communication requirements, network topologies in mobile multi-robot systems can be much more dynamic than those in networks composed of non-mobile nodes; the potentially local communication capabilities of the robots, together with the relatively high likelihood of broken robots, might lead to periods of disconnection. Such disconnections might affect the workings of the blockchain protocol – in particular of block production – as existing blockchain consensus protocols were not designed with these issues in mind. For example, because of potentially high partitioning in a mobile multi-robot system, proof-of-work consensus protocols might become susceptible to local majority attacks, where the largest partition creates the longest blockchain, and can therefore influence the sequence of blocks. Proof-of-authority systems might halt the production of blocks when none of the authorized block producers is reachable.

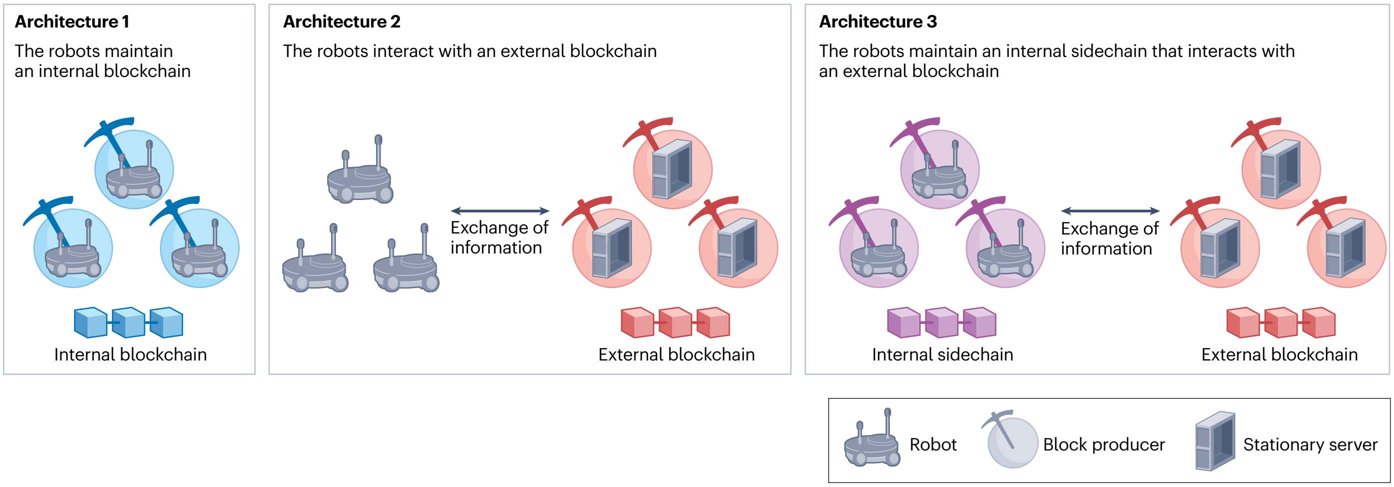


Fig. 1 | Architectures of blockchain-based multi-robot systems. Architecture 1: the blockchain is maintained exclusively by the multi-robot system. Architecture 2: the multi-robot system interacts with an external blockchain, for example, a publicly hosted blockchain. Architecture 3: the multi-robot system hosts an internal sidechain that is maintained by the robots and synchronizes relevant information with an externally hosted blockchain when possible.

To conclude, there is a need for dedicated blockchain frameworks – or at least for making appropriate design choices in existing ones, for example, in terms of consensus protocol, block period or block size – considering the specificities of mobile multi-robot systems. The SwarmDAG protocol, for example, organizes information using a partition-tolerant distributed database based on a DAG instead of a linear blockchain[44]. In this way, when the multi-robot system is split into disconnected subsystems, robots can first reach a local consensus in a subnetwork partition and then a global consensus when they reunite, leading to eventual data consistency.

The implementation of DAG-based ledgers is still in an early stage and tests on real robots are limited. One advantage of DAG-based ledgers lies in their ability to append transactions in different partitions, which might improve the scalability of the system as the number of robots increases. However, this same feature makes the execution of smart contracts more challenging because DAG-based ledgers lack the inherent transaction ordering and immutability that a linear blockchain provides. The DAG-based distributed ledger framework IOTA[45,46] tackles this challenge by executing smart contracts on separate blockchain layers maintained by subnetworks of nodes. However, IOTA requires a centralized coordinator on the DAG layer, thus contradicting the intended decentralized nature of the system. It remains unclear whether the added value provided by the higher partition tolerance compensates for the associated costs caused by the increased data structure complexity and for the extra challenges in achieving a consistent data state.

Blockchain architecture

The scalability issues and design trade-offs presented above bring us to the question of where to host the blockchain, that is, which architecture to choose (Fig. 1).

In Architecture 1, the blockchain network is hosted by the robots themselves that act as full blockchain nodes without any external interactions. Such an architecture is particularly interesting for the classical application domains of swarm robotics, where the robots are supposed to act fully autonomously and where a connection to the Internet or other external infrastructure is not warranted (for example, underwater or underground). In some cases – such as in heterogeneous robot swarms composed of robots with different degrees of computational capabilities – it might be possible to use the more capable robots as full nodes (they store and propagate the blockchain) and let the less capable robots act as light nodes (they interact with the blockchain through the full nodes without either storing or propagating the blockchain).

Interacting with an external blockchain (Architecture 2) can be interesting when the robots in the mobile multi-robot system can reliably connect to it. For example, if the external blockchain is a public blockchain, this architecture can enable the implementation of business models, as the crypto tokens of a public blockchain usually have a real-world monetary value. In addition, an established public blockchain provides all the necessary security features without adding considerable computational overhead to the robots. Although an external blockchain might, at first sight, resemble control via a central server, there are important differences. An external blockchain is a decentralized system and, therefore, does not exhibit a single point of failure. Additionally, any node of the external peer-to-peer network that maintains the blockchain is a possible entry point to interact with the blockchain and the mobile multi-robot system. Even if some of the blockchain nodes become unavailable, both the external blockchain and the mobile multi-robot system continue to work. The use of external blockchains also has some limitations. First, blockchains can usually be more easily accessed by outsiders than central servers, and therefore additional privacy features should be implemented. Second, blockchains can introduce longer delays in updating information than central servers, and therefore the right value for relevant parameters – such as the block period and block size – should be selected. However, parameter personalization is not always possible as external blockchains often are not customizable. Third, storing information in external blockchains can potentially be costly, and therefore it might be necessary to choose which information should be processed on-chain and which information should be processed off-chain.

We believe that a hybrid sidechain architecture (Architecture 3), which combines sidechains hosted by the robots with a mainchain hosted by an external blockchain network, could be an efficient and scalable choice. Even though the sidechains are hosted by the robots themselves, it is possible for the robots to transfer information and crypto tokens from a mainchain to the sidechains, without overloading the mainchain and without being restricted by the properties of the mainchain (such as transaction costs and long block periods). Similar to Architecture 1, in such a hybrid approach an important design choice is which robots should maintain the sidechain.

Authenticating real-world information

Whereas blockchains guarantee the integrity of the information stored in the smart contracts and the validity of their deterministic output, they cannot guarantee the validity of off-chain (real-world) inputs[47,48]. However, sensing, agreeing and acting on real-world states (for example, environmental conditions) and events (for example, completion of a task) is important for many mobile multi-robot system activities. The bridge between off-chain and on-chain information can be established by entities, often called oracles[47], which provide external data to smart contracts.

An important design choice is whether to rely on centralized oracles (trusted third parties that feed the smart contract with the required data) or on decentralized oracles (a group of contributors that do not necessarily trust each other but try to reach a consensus on the required data). A centralized oracle can be a reasonable design choice for externally hosted blockchains. However, such a solution is not suitable for fully decentralized blockchain-based multi-robot systems, where instead decentralized oracles could be used: first, they do not exhibit a single point of failure (making them more robust than a centralized one); and, second, the contributors to decentralized oracles could be the robots of the multi-robot system themselves[49]. Certain robotic tasks, such as collective sensing[50], can be regarded as decentralized oracle problems when the robots aggregate the individually collected information on the blockchain. Trust in and integrity of the information obtained via a decentralized oracle could be achieved through mechanisms based on cryptography[49], which protect the systems from external attacks, and economics[51], which regulate the exchange of crypto tokens among contributors.

Transparency, confidentiality and privacy

Although it is sometimes assumed that all blockchains provide anonymity[52], in general, transactions on blockchains are publicly accessible. The purported anonymity is, in fact, pseudonymity and stems from the difficulty of matching a public key used in the blockchain to a real-world identity[53]. Once the real identity of a public key is exposed, however, many blockchain frameworks provide an easy-to-follow trace of the actions of this specific identity. Because transactions are publicly accessible and smart contracts are created by sending transactions, in frameworks supporting smart contracts, the compiled code is also public. Consequently, the source code can be retrieved either because the creator of the smart contract made it publicly accessible or by reverse-engineering from the compiled code.

On the one hand, such transparent open-source code can give the community the chance to detect vulnerabilities and improve the system so that it is secure and trustworthy[54]. Security patches can be applied either by designing smart contracts that can be upgraded at run time[55] or by letting the multi-robot system use a new smart contract each time a new vulnerability is detected. On the other hand, transparency also facilitates exploits, as for example happened with an exploit in an Ethereum smart contract where an attacker was able to steal 3.6 million ether tokens (worth over US$50 million at the time) due to a vulnerability in the code[56]. In addition, generally, the transactions that serve as input to the functions of a smart contract are publicly accessible, making the system vulnerable to data leakages. It will be particularly important to determine how to prevent such exploits as blockchain-based multi-robot systems, due to their physicality, can pose an immediate danger to life and the environment.

Privacy-enhancing technologies can be employed to conceal the trail of transactions that is usually exposed in blockchain frameworks. To this purpose, Monero conceals the trail of transactions by employing a set of privacy-enabling features, such as ring signatures. However, Monero does not support smart contracts and in blockchain frameworks that do so, protecting privacy becomes more challenging.

There are proposals for secret smart contracts, as exemplified by the Oasis Network or by the Secret Network. These protocols, however, can require the use of specialized hardware (such as trusted execution environments[57]) and are, therefore, not suited for all robotic applications. In addition, current frameworks do not offer full privacy and it is still possible to read both the input and the output, that is, the transactions and the results of the execution of the smart contracts.

Another possibility is to temporarily conceal the input to smart contracts by using commitment schemes based on hashing[58,59]. For example, in certain applications, a smart contract might reward robots based on the quality of the information that they send (see also 'Authenticating real-world information'). In these applications, it is important that each oracle data point is obtained independently: the robots should not copy the data from other robots without performing any sensing or action themselves. By first sending only the hash of the actual data point as a commitment, the data can initially be concealed. With a smart contract, it is possible to wait for a pre-specified number of oracle data points, and afterwards to ask the robots to reveal the actual data. Data integrity could be ensured by comparing the actual data with the previously sent hash value.

Opportunities for blockchain-based mobile multi-robot systems

Enhancing multi-robot systems through blockchain technology enables a set of opportunities that future research could exploit (see Fig. 2). Note that none of these opportunities concern the low-level control of individual robots, that likely needs to be executed using traditional off-chain control. Therefore, the designer will need, first, to understand whether a robot activity should be implemented using on-chain smart contracts, and then whether it is possible to combine the off-chain and on-chain controls using hybrid control methods.

Self-governance

Blockchain technology offers the opportunity for robots to self-bootstrap and self-govern multi-robot systems of which they are part, through decentralized autonomous organizations (DAOs). A DAO is an organization whose policy and interactions among its members are managed digitally by smart contracts[60,61]. Even though members of DAOs are typically humans, it is possible to conceive DAOs governed by robots or a mix of robots and humans[62] (Fig. 2a) that could acquire and use voting shares in the form of DAO tokens (crypto tokens issued by the respective DAO). The governance could comprise system activities and decisions about, for example, who are the leaders for specific missions, what are the internal rules and regulations, whether the robots' software should be updated and who are the owners of the robots.

Such governance could be particularly relevant for open multi-robot systems, which are composed of robots belonging to different stakeholders[27,63] (such as individuals, companies or governmental bodies). These robots participate in a collective application but have potentially conflicting individual goals, such as maximizing individual, rather than collective, profit. We refer to such dynamic multi-robot systems as 'open' because robots from different parties can join and leave the systems. In an open multi-robot system, a DAO could establish an access control layer that dictates which robots can join or need to leave the system. In this case, it will be important to determine how to specify the rules for joining (for example, by voting or by DAO token acquisition) and leaving (for example, voluntary leave of the robot or forced leave due to Byzantine behaviour) the multi-robot system.

Compliance and accountability

Once mobile multi-robot systems are deployed in the real world, they should exhibit compliant behaviour: they should work as intended, comply with applicable laws and not cause any harm to humans, to the environment or to other robots[15,16]. Robots that do not comply with the intended behaviour – also called Byzantine robots (Box 1) – can adversely affect group performance, and even cause full system failure[27].

Non-compliant robot behaviour could be detected within the multi-robot system without external interaction through the integration of a blockchain (Fig. 2b), for example, by comparing the robots' behaviours with each other or with the agreed-upon protocol stored in a smart contract. Smart contracts can also implement anomaly detection methods to exclude Byzantine robots from participating in the multi-robot system activities and can be used to store robots' tamper-proof reputation values. Robots that contribute valuable information for a given task could increase their reputation, whereas Byzantine robots that send misleading information (determined by an anomaly detection method) would be assigned a lower reputation. Such reputation values could then be used for weighted information aggregation so that the information contributed by robots with a higher reputation has a larger weight than information from robots with a lower reputation – preventing misbehaving robots from harming collective success.

Because a blockchain can securely exchange and store robot to robot and human to robot messages, it could also be used to analyse and monitor robots' and people's behaviour by external observers – both during live operation and after the completion of a mission[64]. In addition, the blockchain could also be used to establish accountability, and the chain of actions recorded in it could be used in law enforcement when the system did not act in compliance with applicable laws.

Robot economy

Even though blockchain technology is now used in a wide variety of applications, such as supply chain management and voting[65], it was originally developed for the maintenance of decentralized digital

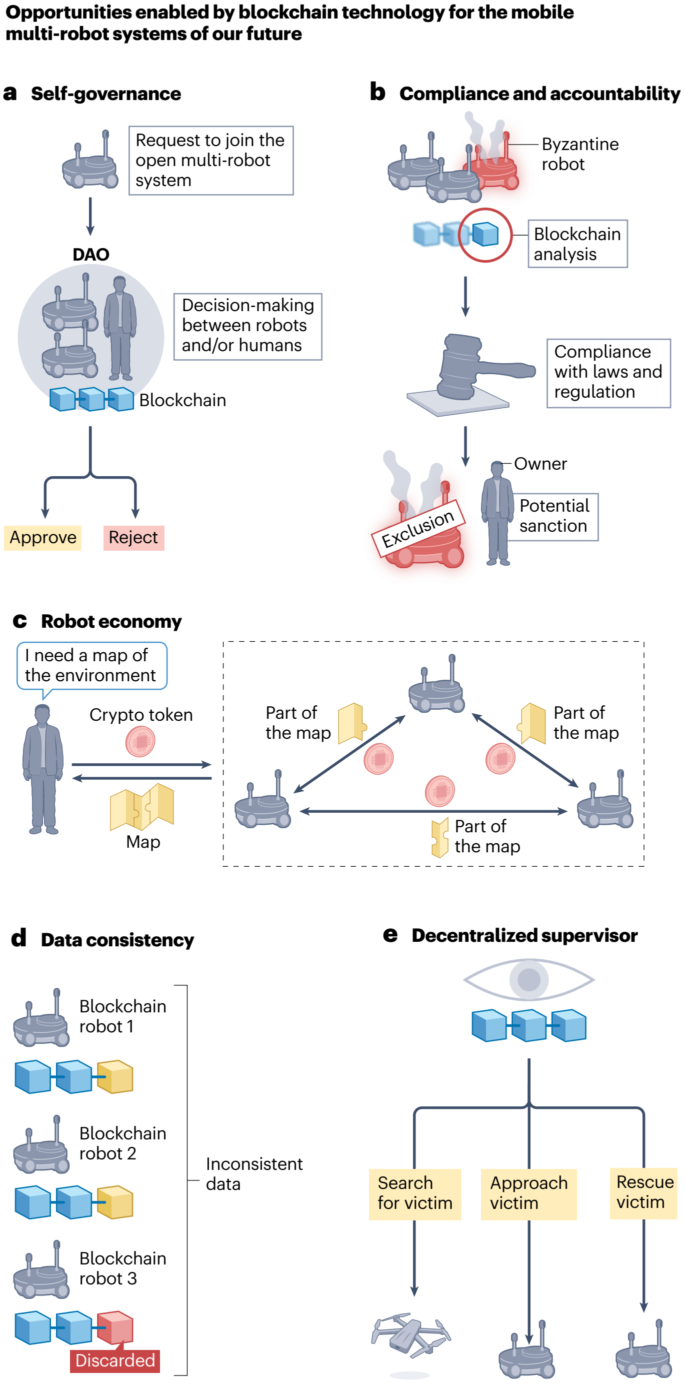


Fig. 2 | Opportunities enabled by blockchain technology for the mobile multi-robot systems of our future. a, By regulating decentralized autonomous organizations (DAOs) through smart contracts on the blockchain, robots and humans could make joint decisions on how to self-govern the mobile multi-robot system, for example, which robots can join or need to leave the system. **b**, Compliance with regulations and accountability could be achieved through online or offline analysis of the tamper-proof blockchain, where actions and decisions made by each accountable robot are logged. In this way, it is possible to exclude Byzantine robots and, potentially, sanction their owners. **c**, Robots trading crypto tokens as economic agents could lead to the creation of a decentralized interface between humans and multi-robot systems. Trading crypto tokens could also enable cooperation between robots as part of a robot to robot economy. **d**, Consistent shared data can facilitate the execution of collaborative tasks, such as simultaneous localization and mapping (SLAM) and federated learning, and can also prevent Sybil attacks. **e**, Smart contracts can act as a decentralized supervisor implementing system-level action policies, for example, for assigning tasks to robots according to their capabilities.

currencies (cryptocurrencies) to execute global financial transactions (see also Box 2). Digital currencies not only enable humans to exchange tokens of value but could also enable robots to take part in economies[66]. Therefore, we envision robot to human economies, in which humans could hire mobile multi-robot systems for certain applications (such as mapping an unknown environment) and pay by sending crypto tokens to the systems' accounts, establishing robots-as-a-service applications[14] (Fig. 2c). Mobile multi-robot systems could also pay humans for certain tasks, such as maintenance or battery recharges.

Blockchain technology could also enable multi-robot systems to establish an economy among robots – within the multi-robot system – without the need for a centralized management of the economy. Robots that belong to different stakeholders could trade goods and services among themselves in exchange for crypto tokens. For example, flying robots could sell locations of resources, maps of the environment and navigation advice to ground robots, whereas ground robots could sell physical help in object transportation. In open multi-robot systems, such economic exchanges could enable cooperative behaviour in groups of self-interested robots. Additionally, by granting economic rewards to robots based on their performance, it could be possible to measure the reputation of a robot through the number of crypto tokens it possesses.

Data consistency

Many activities performed by mobile multi-robot systems require an agreement on the data that are shared among the robots (Fig. 2d). Synchronizing consistent and conflict-free data across the robots in a mobile multi-robot system is a challenging problem due to the potential occurrence of failed components, communication delays and attacks[67]. For example, a severe problem in decentralized systems is double counting, where the sender of a possibly erroneous message receives back its own message after it was disseminated in the network and treats it as a new message[68]. When the messages are stored and aggregated on a blockchain, such a situation can be avoided, as each message is uniquely identifiable. Whereas double counting might be unintentional, decentralized systems can also be vulnerable to intentional Sybil attacks[69]. During a Sybil attack, a small minority of nodes (robots in multi-robot systems) forge many fake identities to gain control over the system or interfere with its operations. A blockchain can prevent this situation by assigning unique identifiers to each robot in closed mobile multi-robot systems[17] or, alternatively, by introducing scarcity into the system, for example, by charging a fee in crypto tokens for each sent message in open mobile multi-robot systems (the limiting factor is therefore the number of crypto tokens and not the number of identities)[27].

Collaborative simultaneous localization and mapping (SLAM) is an example task that needs shared data in mobile multi-robot systems. For this task, a database is needed to store and aggregate the map. Although this topic is currently being discussed in the literature[70,71], most of the mobile multi-robot SLAM research addresses the issue by employing centralized map aggregation. Blockchain technology has the potential to provide the necessary infrastructure for mobile multi-robot systems to achieve this in a conflict-free and distributed manner, protecting from double counting and Sybil attacks.

Data synchronization is also important for collective learning in mobile multi-robot systems. Traditional machine learning requires the transmission of all data from the robots to a server to train a machine-learning model, which can have huge memory and communication costs. Federated learning is a technique where the devices (robots in this case) train a model locally and then exchange only the learned parameters instead of the raw data[72]. This technique is advantageous as it increases privacy, reduces data exchange and allows for parallelization of computing resources. To aggregate the locally trained parameters, existing work mostly uses centralized servers. A blockchain maintained by the mobile multi-robot system can potentially serve as a decentralized and secure data structure that is able to execute federated learning algorithms.

Decentralized supervisor

A smart contract can run algorithms that act as a decentralized supervisor of the mobile multi-robot system, enhancing its collective performance and decision-making abilities. Such a decentralized supervisor gathers individual robot inputs and generates system-level action policies that the individual robots can implement considering their capabilities[29] (Fig. 2e). This form of hierarchical control could allocate group-level decisions to the smart contract, while simultaneously allowing for fast actions to be made by the individual robots, in a way that preserves local robustness and responsiveness.

Decentralized supervisors could also have an important role in situations where groups of robots are required to make safety-critical group decisions, particularly in emergency scenarios where it is imperative to respond in an effective manner. In such scenarios, a centralized controller becomes a potential single point of failure and risks being overloaded by the increased communication demands that occur during an emergency. Conversely, letting robots exchange votes locally can be time-consuming and might not ultimately lead to a consensus on a system-level action plan[73], thus resulting in a weak or inadequate response to the emergency. A decentralized supervisor could be used to yield a swift and coordinated response from all of the robots.

Outlook

In the past few years, we have seen conceptual papers presenting blockchain-based mobile multi-robot systems[16,36,74] and first proofs of concept with simulated and real robots[13,17–19,27,28,30]. However, the integration of blockchains into multi-robot systems requires addressing several technological challenges. Some challenges, such as the management of transparency, confidentiality and privacy, are common to both the domain of mobile multi-robot systems and stationary computer networks, whereas others require robotics-specific solutions. For example, to address scalability one should explicitly consider computing, storage and communication limitations of hardware-constrained robots, as well as frequent network partitioning caused by the mobility of the robots. The techniques to address scalability which have found success in the domain of traditional blockchain networks, such as sidechain architectures[25] (similar to Architecture 3) and permissioned consensus protocols[39], require further investigation within the domain of robotics before scalable and secure deployment is possible, particularly when the mobile multi-robot system is composed of a very large number of robots.

Another critical challenge to deploy blockchain-based multi-robot systems is the design of oracles that inject trustworthy real-world information into smart contracts. We believe that further research in decentralized oracles, in which the participants – robots or humans – act as oracle contributors, can lead to solutions that avoid centralization and a potential single point of failure. Game theory and mechanism design[75] could be applied to implement economic mechanisms that reward useful information. In this way, oracle contributors would be motivated to adhere to the specified protocol and make useful contributions to the decentralized oracle to maximize their reward.

Once challenges inherent to blockchain technology are solved, the implementation of smart contracts can enable several of the opportunities presented in this Perspective (Fig. 2). Blockchain technology could enable open mobile multi-robot systems having different stakeholders. Although such open multi-robot systems could be highly flexible, they might prove challenging to coordinate. DAO-based secure self-governance is a promising solution, but requires further research to become a reality. Deploying autonomous robots that are compliant with regulations and accountable for their actions could be key in favouring the rise of mobile multi-robot systems and their acceptance by the public because they can be trusted. Additionally, if some of the robots are destroyed or lost, it would be possible to recover a tamper-proof record of the events of a mission (similarly to black boxes used in the aviation industry)[76–78] and allow investigations to assign liability.

Once we have ensured the compliant behaviour of mobile multi-robot systems, we can start deploying them in real-world applications and let them interact with our economy or even build their own economies. Building an economy among robots that aim at maximizing their reward could lead to efficient self-organization and task allocation where robots perform the tasks for which their skills bring the best contribution. For achieving such self-organized behaviour, it will be important to define the appropriate economic incentives in the smart contracts regulating the robot economy[59].

Data consistency in mobile multi-robot systems is of paramount importance in many collective activities such as, for example, monitoring an environment with robots submitting independent evaluations[79] or selection of the shortest available path during collective motion in a cluttered environment. Further research will need to investigate which activities would benefit from activity-specific data synchronization and how the capabilities of the robots in terms of storage and communication bandwidth, as well as communication delays caused by partitions in the mobile multi-robot system communication network, affect data synchronization.

Finally, how to supervise the activities of a mobile multi-robot system without relying on centralized control is a general unsolved problem. This problem has been addressed by trying to introduce elements of centralized control in an otherwise self-organized system so that the typical desired properties of self-organization (such as scalability, robustness and flexibility) are preserved[4,80]. The use of blockchain technology could allow an alternative implementation of such centralized components: smart contracts can play the role of central controllers while benefitting from the fully distributed nature of the blockchain. To exploit the opportunity to implement decentralized supervisors, several scientific questions need to be addressed, including how to integrate the commands from decentralized supervisors with a robot's local control software, understanding which are the applications where such an approach is the most desirable and how to preserve system scalability when the number of robots and tasks is large.

Published online: 25 March 2024

References

1. Dudek, G., Jenkin, M. R., Milios, E. & Wilkes, D. A taxonomy for multi-agent robotics. *Autonomous Robot.* **3**, 375–397 (1996).
2. Parker, L. E. Multiple mobile robot systems. In *Springer Handbook of Robotics* 921–941 (Springer, 2008).
This paper presents an accessible introduction to the foundations and early successes of mobile multi-robot systems.
3. Zhang, L., Zhang, Z., Siegwart, R. & Chung, J. J. Distributed PDOP coverage control: providing large-scale positioning service using a multi-robot system. *IEEE Robot. Autom. Lett.* **6**, 2217–2224 (2021).
4. Mathews, N., Christensen, A. L., O'Grady, R., Mondada, F. & Dorigo, M. Mergeable nervous systems for robots. *Nat. Commun.* **8**, 439 (2017).
5. Timmis, J., Ismail, A. R., Bjerknes, J. D. & Winfield, A. F. T. An immune-inspired swarm aggregation algorithm for self-healing swarm robotic systems. *Biosystems* **146**, 60–76 (2016).
6. Mathews, N., Christensen, A. L., Stranieri, A., Scheidler, A. & Dorigo, M. Supervised morphogenesis: exploiting morphological flexibility of self-assembling multirobot systems through cooperation with aerial robots. *Robot. Autonomous Syst.* **112**, 154–167 (2019).
7. Rizk, Y., Awad, M. & Tunstel, E. W. Cooperative heterogeneous multi-robot systems: a survey. *ACM Comput. Surv*. **52**, https://doi.org/10.1145/3303848 (2019).
This paper presents an overview of recent research achievements as well as open challenges in multi-robot systems.
8. Dorigo, M., Theraulaz, G. & Trianni, V. Swarm robotics: past, present and future. *Proc. IEEE* **109**, 1152–1165 (2021).
This review surveys the past, present and future of swarm robotics, discussing open challenges and research directions.
9. Wurman, P., D'Andrea, R. & Mountz, M. Coordinating hundreds of cooperative, autonomous vehicles in warehouses. *AI Mag.* **29**, 9–20 (2008).
10. Yang, G.-Z. et al. The grand challenges of *Science Robotics*. *Sci. Robot.* **3**, eaar7650 (2018).
11. Dorigo, M., Theraulaz, G. & Trianni, V. Reflections on the future of swarm robotics. *Sci. Robot.* **5**, abe4385 (2020).
12. Wilson, J. et al. Trustworthy swarms. In *Proc. First Int. Symp. Trustworthy Autonomous Systems* https://doi.org/10.1145/3597512.3599705 (ACM, 2023).
13. Strobel, V., Castelló Ferrer, E. & Dorigo, M. Managing Byzantine robots via blockchain technology in a swarm robotics collective decision making scenario. In *Proc. 17th Int. Conf. Autonomous Agents and Multiagent Systems (AAMAS 2018)* 541–549 (IFAAMAS, 2018).
14. Castelló Ferrer, E., Hardjono, T., Pentland, A. & Dorigo, M. Secure and secret cooperation in robot swarms. *Sci. Robot.* **6**, abf1538 (2021).
15. Hunt, E. R. & Hauert, S. A checklist for safe robot swarms. *Nat. Mach. Intell.* **2**, 420–422 (2020).
This paper presents a ten-item checklist to determine whether a robot swarm is safe.
16. Castelló Ferrer, E. If blockchain is the solution, robot security is the problem. *Front. Blockchain* **6**, 1181820 (2023).
17. Strobel, V., Pacheco, A. & Dorigo, M. Robot swarms neutralize harmful Byzantine robots using a blockchain-based token economy. *Sci. Robot.* **8**, eabm4636 (2023).
This paper presents the first large-scale proof of concept of how to integrate blockchain technology into decentralized multi-robot systems.
18. Santos De Campos, M. G., Chanel, C. P., Chauffaut, C. & Lacan, J. Towards a blockchain-based multi-UAV surveillance system. *Front. Robot. AI* **8**, 557692 (2021).
19. Grey, J., Godage, I. & Seneviratne, O. Swarm contracts: Smart contracts in robotic swarms with varying agent behavior. In *Proc. 2020 IEEE Int. Conf. Blockchain (Blockchain 2020)* 265–272 (IEEE, 2020).
20. Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/en/bitcoin-paper (2008).
This article describes the first practical implementation of a consensus-based decentralized digital currency that overcomes the Byzantine generals problem and introduces blockchain technology as a ledger for storing transactions of the cryptocurrency Bitcoin.
21. Buterin, V. A next-generation smart contract and decentralized application platform. Ethereum Project white paper. *Ethereum* https://ethereum.org/en/whitepaper/ (2014).
In this work the blockchain framework Ethereum generalizes the idea behind a blockchain from a store of value to a decentralized computing system, enabling smart contracts.
22. Peña Queralta, J. et al. Blockchain and emerging distributed ledger technologies for decentralized multi-robot systems. *Curr. Robot. Rep.* **4**, 43–54 (2023).
23. Aditya, S., Singh, R., Singh, P. K. & Kalla, A. A survey on blockchain in robotics: issues, opportunities, challenges and future directions. *J. Netw. Computer Appl.* **196**, 103245 (2021).
24. Peña Queralta, J. & Westerlund, T. Blockchain for mobile edge computing: Consensus mechanisms and scalability. In *Mobile Edge Computing* 333–357 (Springer, 2021).
25. Singh, A. et al. Sidechain technologies in blockchain networks: an examination and state-of-the-art review. *J. Netw. Comput. Appl.* **149**, 102471 (2020).
26. Strobel, V. & Dorigo, M. Blockchain technology for robot swarms: A shared knowledge and reputation management system for collective estimation. In *Swarm Intelligence—Proc. ANTS 2018—11th Int. Conf.* 425–426 (Springer, 2018). [Lecture Notes in Computer Science 11172].
27. Strobel, V., Castelló Ferrer, E. & Dorigo, M. Blockchain technology secures robot swarms: a comparison of consensus protocols and their resilience to Byzantine robots. *Front. Robot. AI* **7**, 54 (2020).
28. Pacheco, A., Strobel, V. & Dorigo, M. A blockchain-controlled physical robot swarm communicating via an ad-hoc network. In *Swarm Intelligence—Proc. ANTS 2020—12th Int. Conf*. 3–15 (Springer, 2020). [Lecture Notes in Computer Science 12421].
29. Pacheco, A., Strobel, V., Reina, A. & Dorigo, M. Real-time coordination of a foraging robot swarm using blockchain smart contracts. In *Swarm Intelligence—Proc. ANTS 2022—13th Int. Conf*. 196–208 (Springer, 2022). [Lecture Notes in Computer Science 13491].
30. Castelló Ferrer, E., Jiménez, E., Lopez-Presa, J. L. & Martín-Rueda, J. Following leaders in Byzantine multirobot systems by using blockchain technology. *IEEE Trans. Robot.* **38**, 1101–1117 (2021).
31. Alsamhi, S. H. et al. Blockchain-empowered security and energy efficiency of drone swarm consensus for environment exploration. *IEEE Trans. Green. Commun. Netw.* **7**, 328–338 (2023).
32. Mokhtar, A., Murphy, N. & Bruton, J. Blockchain-based multi-robot path planning. In *Proc. 5th IEEE World Forum on Internet of Things (WF-IoT 2019)* 584–589 (IEEE, 2019).
33. Grey, J., Seneviratne, O. & Godage, I. Blockchain-based mechanism for robotic cooperation through incentives: Prototype application in warehouse automation. In *Proc. 2021 IEEE Int. Conf. Blockchain (Blockchain 2021)* 597–604 (IEEE, 2021).
34. Mallikarachchi, S., Dai, C., Seneviratne, O. & Godage, I. Managing collaborative tasks within heterogeneous robotic swarms using swarm contracts. In *Proc. 4th IEEE Int. Conf. Decentralized Applications and Infrastructures (DAPPS 2022)* 48–55 (IEEE, 2022).
35. Castelló Ferrer, E., Rudovic, O., Hardjono, T. & Pentland, A. Robochain: A secure data-sharing framework for human-robot interaction. In *Proc. 10th Int. Conf. Health, Telemedicine, and Social Medicine (eTELEMED 2018)* 124–130 (IARIA, 2018).
36. Alsamhi, S. H. & Lee, B. Blockchain-empowered multi-robot collaboration to fight COVID-19 and future pandemics. *IEEE Access.* **9**, 44173–44197 (2021).
37. Kapitonov, A., Lonshakov, S., Bulatov, V., Montazam, B. K. & White, J. Robot-as-a-service: from cloud to peering technologies. *Front. Robot. AI* **8**, 560829 (2021).
38. Kapitonov, A., Lonshakov, S., Krupenkin, A. & Berman, I. 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)* 84–89 (IEEE, 2017).
39. Ongaro, D. & Ousterhout, J. In search of an understandable consensus algorithm. In *2014 USENIX Annu. Technical Conf. (USENIX ATC 14)* 305–319 (2014).
40. Androulaki, E. et al. Hyperledger Fabric: A distributed operating system for permissioned blockchains. In *Proc.* 13th *EuroSys Conf*. 1–15 (ACM, 2018).
41. Salimi, S., Peña Queralta, J. & Westerlund, T. Hyperledger Fabric blockchain and ROS 2 integration for autonomous mobile robots. In *2023 IEEE/SICE Int. Symp. System Integration* 1–8 (IEEE, 2023).
42. Wardega, K., von Hippel, M., Tron, R., Nita-Rotaru, C. & Li, W. Byzantine resilience at swarm scale: A Decentralized Blocklist Protocol from inter-robot accusations. In *Proc. 2023 Int. Conf. Autonomous Agents and Multiagent Systems (AAMAS '23)* 1430–1438 (IFAAMAS, 2023).
43. Hoffmann, F. Challenges of proof-of-useful-work (PoUW). In *Proc. IEEE 1st Global Emerging Technology Blockchain Forum: Blockchain & Beyond (iGETblockchain 2022)* https://doi.org/10.1109/iGETblockchain56591.2022.10087185 (IEEE, 2022).
44. Tran, J. A. et al. SwarmDAG: a partition tolerant distributed ledger protocol for swarm robotics. *Ledger* **4**, https://doi.org/10.5195/ledger.2019.174 (2019).
45. Keramat, F., Peña Queralta, J. & Westerlund, T. Partition-tolerant and Byzantine-tolerant decision making for distributed robotic systems with IOTA and ROS2. *IEEE Internet Things J.* **10**, 12985–12998 (2023).
46. Salimpour, S., Keramat, F., Peña Queralta, J. & Westerlund, T. Decentralized vision-based Byzantine agent detection in multi-robot systems with IOTA smart contracts. In *Foundations and Practice of Security: 15th Int. Symp., FPS 2022, Revised Selected Papers* 322–337 (Springer, 2023).
47. Al-Breiki, H., Rehman, M. H. U., Salah, K. & Svetinovic, D. Trustworthy blockchain oracles: review, comparison, and open research challenges. *IEEE Access.* **8**, 85675–85685 (2020).
48. Mühlberger, R. et al. Foundational oracle patterns: Connecting blockchain to the off-chain world. In *Business Process Management: Blockchain and Robotic Process Automation Forum* 35–51 (Springer, 2020).
49. Zhao, H. et al. A generic framework for Byzantine-tolerant consensus achievement in robot swarms. In *IEEE/RSJ Int. Conf. Intelligent Robots and Systems—IROS 2023* 8839–8846 (IEEE, 2023).
50. Valentini, G., Brambilla, D., Hamann, H. & Dorigo, M. Collective perception of environmental features in a robot swarm. In *Swarm Intelligence—Proc. ANTS 2016—10th Int. Conf*. 65–76 (Springer, 2016). [Lecture Notes in Computer Science 9882].
51. Brekke, J. K. & Alsindi, W. Z. Cryptoeconomics. *Internet Policy Rev.* **10**, https://doi.org/10.14763/2021.2.1553 (2021).
52. Andola, N., Raghav, Yadav, V. K., Venkatesan, S. & Verma, S. Anonymity on blockchain based e-cash protocols—a survey. *Computer Sci. Rev.* **40**, 100394 (2021).
53. Conoscenti, M., Vetrò, A. & De Martin, J. C. Blockchain for the Internet of Things: A systematic literature review. In *Proc. 13th IEEE/ACS Int. Conf. Computer Systems and Applications (AICCSA 2016)* 1–6 (2016).
54. Raymond, E. S. *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary* (O'Reilly Media, 1999).

55. Rodler, M., Li, W., Karame, G. O. & Davi, L. EVMPatch: Timely and automated patching of Ethereum smart contracts. In *Proc. 30th USENIX Security Symposium (USENIX Security 21)* 1289–1306 (USENIX Association, 2021).
56. DuPont, Q. Experiments in algorithmic governance: A history and ethnography of 'The DAO,' a failed decentralized autonomous organization. In *Bitcoin and Beyond: Cryptocurrencies, Blockchains, and Global Governance* 157–177 (Routledge, 2017).
57. Sabt, M., Achemlal, M., & Bouabdallah, A. Trusted Execution Environment: What it is, and what it is not. In *Proc. 14th IEEE Int. Conf. Trust, Security and Privacy in Computing and Communications* 57–64 (IEEE Press, 2015).
58. Wöhrer, M. & Zdun, U. Design patterns for smart contracts in the Ethereum ecosystem. In *Proc. IEEE 2018 Int. Congress on Cybermatics* 1513–1520 (IEEE, 2018).
59. Van Calck, L., Pacheco, A., Strobel, V., Dorigo, M. & Reina, A. A blockchain-based information market to incentivise cooperation in swarms of self-interested robots. *Sci. Rep.* **13**, 20417 (2023).
60. Hassan, S. & De Filippi, P. Decentralized autonomous organization. *Internet Policy Rev*. **10**, https://doi.org/10.14763/2021.2.1556 (2021).
61. Wang, S. et al. Decentralized autonomous organizations: concept, model, and applications. *IEEE Trans. Computational Soc. Syst.* **6**, 870–878 (2019).
62. Cardenas, I. S., May, J. B. & Kim, J.-H. AutomataDAO: A blockchain-based data marketplace for interactive robot and IoT data exchanges using Ethermint and state channels. In *Blockchain Technology for IoT Applications* 17–38 (Springer, 2021).
63. Reina, A. Robot teams stay safe with blockchains. *Nat. Mach. Intell.* **2**, 240–241 (2020).
64. Danilov, K., Rezin, R., Afanasyev, I. & Kolotov, A. Towards blockchain-based Robonomics: Autonomous agents behavior validation. In *Proc 9th IEEE Int Conf Intelligent Systems (IS 2018)* 222–227 (IEEE, 2018).
65. Abou Jaoude, J. & Saade, R. G. Blockchain applications—usage in different domains. *IEEE Access.* **7**, 45360–45381 (2019).
66. Castelló Ferrer, E. et al. Gaka-chu: A self-employed autonomous robot artist. In *Proc. 2023 IEEE Int. Conf. Robotics and Automation (ICRA 2023)* 11583–11589 (IEEE, 2023).
67. Lajoie, P.-Y., Ramtoula, B., Wu, F. & Beltrame, G. Towards collaborative simultaneous localization and mapping: a survey of the current research landscape. *Field Robotics* **2**, 971–1000 (2022).
68. Chong, C.-Y., Chang, K.-C. & Mori, S. A review of forty years of distributed estimation. In *Proc. 21st Int. Conf. Information Fusion (Fusion 2018)* 1–8 (IEEE, 2018).
69. Douceur, J. R. The Sybil attack. In *1st International Workshop on Peer-to-Peer Systems* 251–260 (Springer, 2002). [Lecture Notes in Computer Science 2429].
70. Saeedi, S., Trentini, M., Seto, M. & Li, H. Multiple-robot simultaneous localization and mapping: a review. *J. Field Robot.* **33**, 3–46 (2016).
71. Kegeleirs, M., Grisetti, G. & Birattari, M. Swarm SLAM: challenges and perspectives. *Front. Robot. AI* **8**, 618268 (2021).
72. Majcherczyk, N., Srishankar, N. & Pinciroli, C. Flow-FL: Data-driven federated learning for spatio-temporal predictions in multi-robot systems. In *Proc. 2021 IEEE Int. Conf. Robotics and Automation (ICRA 2021)* 8836–8842 (IEEE, 2021).
73. Zakir, R., Dorigo, M. & Reina, A. Robot swarms break decision deadlocks in collective perception through cross-inhibition. In *Swarm Intelligence—Proc. ANTS 2022—3th Int. Conf*. 209–221 (Springer, 2022). [Lecture Notes in Computer Science 13491].
74. Castelló Ferrer, E. The blockchain: A new framework for robotic swarm systems. In *Proc. Future Technol. Conf. (FTC 2018)* Vol. 881 1037–1058 (Springer, 2018).
75. Maskin, E. Introduction to mechanism design and implementation. *Transnatl. Corporations Rev.* **11**, 1–6 (2019).
76. White, R., Caiazza, G., Cortesi, A., Cho, Y. & Christensen, H. Black block recorder: immutable black box logging for robots via blockchain. *IEEE J. Robot. Autom.* **4**, 3812–3819 (2019).
77. Lopes, V. & Alexandre, L. A. Detecting robotic anomalies using Robotchain. In *IEEE Int. Conf. Autonomous Robot Systems and Competitions (ICARSC 2019)* 174–179 (IEEE, 2019).
78. Lopes, V., Pereira, N., Fernandes, M. & Alexandre, L. A. A time-segmented consortium blockchain for robotic event registration. In *Proc. 3rd Int. Conf. Blockchain Technology (ICBCT 2021)* 117–122 (ACM, 2021).
79. Talamali, M. S., Saha, A., Marshall, J. A. R. & Reina, A. When less is more: robot swarms adapt better to changes with constrained communication. *Sci. Robot.* **6**, eabf1416 (2021).
80. Zhu W. et al. Self-organizing nervous systems for robot swarms. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2401.13103 (2024).
81. Dorigo, M., Birattari, M. & Brambilla, M. Swarm robotics. *Scholarpedia* **9**, 1463 (2014).
82. Hamann, H. *Swarm Robotics: A Formal Approach* (Springer, 2018).
83. Gielis, J., Shankar, A. & Prorok, A. A critical review of communications in multi-robot systems. *Curr. Robot. Rep.* **3**, 213–225 (2022).
84. Demir, K. A., Döven, G. & Sezen, B. Industry 5.0 and human-robot co-working. *Procedia Comput. Sci.* **158**, 688–695 (2019).
85. Lamport, L., Shostak, R. & Pease, M. The Byzantine generals problem. *ACM Trans. Program. Lang. Syst.* **4**, 382–401 (1982).
    This foundational paper introduces the Byzantine generals problem — a thought experiment that highlights the challenges of achieving a consensus in distributed networks where the agents (the 'Byzantine generals') are not necessarily reliable.
86. Castro, M. & Liskov, B. Practical Byzantine fault tolerance and proactive recovery. *ACM Trans. Computer Syst.* **20**, 398–461 (2002).
87. Dwork, C., Lynch, N. & Stockmeyer, L. Consensus in the presence of partial synchrony. *J. ACM* **35**, 288–323 (1988).
88. Chaum, D., Fiat, A. & Naor, M. Untraceable electronic cash. In *Advances in Cryptology—Crypto '88* (Springer, 1990). [Lecture Notes in Computer Science 403].
89. Dwork, C. & Naor, M. Pricing via processing or combatting junk mail. In *Proc. Annu. Int. Cryptology Conf.—Advances in Cryptology (Crypto' 92)* 139–147 (Springer, 1992). [Lecture Notes in Computer Science 740].

Acknowledgements

This work was supported by the Program of Concerted Research Actions (ARC) of the Université Libre de Bruxelles and by the Service Public de Wallonie Recherche under grant n° 2010235 — ARIAC by DigitalWallonia4.ai. M.D., A.R. and V.S. acknowledge support from the Belgian F.R.S.-FNRS. A.R. also acknowledges support by the Deutsche Forschungsgemeinschaft (DFG; German Research Foundation) under Germany's Excellence Strategy — EXC 2117-422037984.

Author contributions

M.D. and V.S. wrote the manuscript. All authors discussed the scope of the article, contributed with ideas and revised the manuscript.

Competing interests

The authors declare that they do not have any competing interests.

Additional information

Peer review information *Nature Reviews Electrical Engineering* thanks Erol Sahin and the other anonymous reviewer(s) for their contribution to the peer review of this work.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Related links

Hyperledger: https://www.hyperledger.org
IOTA: https://www.iota.org/
Monero: https://www.getmonero.org/
Oasis Network: https://oasisprotocol.org/
PoA consensus mechanism: http://eips.ethereum.org/EIPS/eip-225
PoS consensus mechanism: http://ethereum.org/en/developers/docs/consensus-mechanisms/pos
Secret Network: https://scrt.network/
Stellar Consensus Protocol: https://stellar.org/learn/stellar-consensus-protocol