Applications of Distributed Ledger Technologies in Robotics*

Garry Clawson[1,2]

Abstract— **Increasing convergence of virtual space and physical space driven by society 5.0 has propelled research in distributed ledgers to identify how trust in autonomous systems and robotics can be enabled through decentralised frameworks for decision making and consensus building. Industrial, multi-robot systems, embedded systems, and more have seen growth in applications using distributed ledgers. Although progressing, implementations are currently ad-hoc with middleware systems, such as ROS, not yet adopting a standard to permit easy integration of these technologies. Nor is the technology readiness level of distributed ledgers easily identifiable. This creates challenges for adoption and system integration.**

I. INTRODUCTION

As the convergence between the virtual space and physical space accelerates [1], interest in the application of Distributed Ledger Technology (DLT) within the robotics domain from industrial, social, agri-tech, transport and health care has been seen [2]. Each of these systems requires resilience and twenty four seven availability. A principal requirement for the deployment of robotic systems in the public domain is to have trust among robots as well as between robots and any human facing environment these systems are operated within [3]. Across swarm robots, multi-robot systems, healthcare or even industrial robots, trust and integrity is a critical factor [4]. However, robots that operate in remote and hostile environments and are often open to the outside world, enabling attackers an opportunity to take control of a robot with the potential to disrupt the operations of a fleet or an entire system [5]. A compromised robot may behave in byzantine manner which can present danger or harmful situations to operators or users alike[6]. Blockchain technology has the potential to play an increasingly important role in providing levels of trust, security, privacy as well as auditability across a robotic system not previously available without centralised co-ordination [7].

Towards this aim, this review explores current applications of DLT's in robotics, current challenges and future directions. Section II reviews the preliminaries of blockchain, section III current applications, section IV integration of DLTs into robotic middleware, section V and VI conclude with a discussion and conclusion respectively.

II. PRELIMINARIES OF DLT

This section introduces the component technologies of blockchain. It begins by briefly describing blockchain architecture, consensus algorithms, smart contracts, and key functional properties of blockchain.

A. Architecture

A distributed ledger or blockchain can be considered as a time stamped cryptography connected linked list of immutable blocks [8]. When a block is generated, it is appended to its parent block and linked via a cryptographic hash pointer which consist of the parent block itself [9]. Each block, which consists of transactions as well as header information, then acts to create a continually updating ledger [10]. as shown in Fig. 1. Each block in the chain therefore implicitly verifies the integrity of the full chain preceding it. This distributed ledger or blockchain, is held and validated by all participants in the network [11].

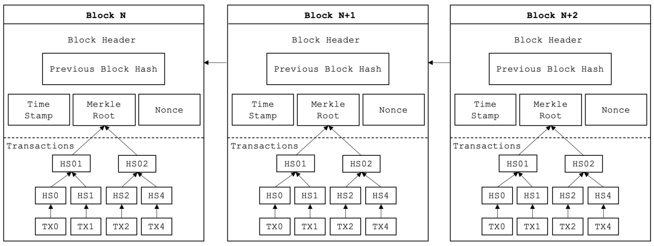

Fig. 1. A blockchain is organized into blocks. Each block consists of two components - a header and body. The header contains the metadata of the previous block which creates a chain in the form of a linked list. The body contains the transaction data that forms part of the block header [12].

B. Consensus Algorithms

A peer to peer distributed blockchain system requires a consensus mechanism to ensure that one conical blockchain state across the peer to peer network is maintained [13]. A key feature of the consensus mechanism is to provide a practical solution to the Byzantine Generals Problem [14]. Proof of Work [15], Proof of Stake [16], Delegated Proof of Stake [17] and Proof of Elapsed Time [18] are all highly popular consensus mechanisms. A review of these mechanisms, and others, has previously been covered by [19] and [20].

C. Smart Contracts

A Smart Contract is a *self-executing* contract where the conditions of the contract between the interested parties are enforced by consensus protocols [21]. Contracts are written in the form of a computer program that exists across a

---

*This work was supported by the Engineering and Physical Sciences Research Council [EP/S023917/1].

[1] Garry Clawson is with the School of Computer Science, University of Lincoln, Lincoln, UK. `18685030@students.lincoln.ac.uk`

distributed, decentralized blockchain network. The contracts can encode any pre-defined rules and execute the corresponding operations when defined conditions are satisfied [22]. Smart contracts allow transactions to be conducted between anonymous or untrusted parties without the need for a central authority.

D. *Key Properties of Blockchain*

Evaluation of distributed ledgers is a critical component when choosing where to build distributed infrastructure. Accessibility, throughput, scaleablity and capacity are useful metrics. Within a middleware robotics messaging system, such as the Robotic Operating System (ROS) [23], millions of real time or queued messages can be available to be published or subscribed to at any one time. This provides key constraints.

1) Throughput: is the number of transactions that can be processed by the chain during every block period [24]. Blocks times are poisson distributed and can range from a few seconds to a few hours, depending on the blockchain.

2) Latency: is the amount of time it takes for a transaction, from being first issued to being confirmed in a block and achieve a consensus [25]. Latency can be measured by the Time To Finality (TTF) of which the transaction is deemed to be practically irreversible. This metric varies per blockchain.

3) Fault Tolerance: specifies the number of dishonest or faulty nodes that can be tolerated by the blockchain at any one time [26]. A blockchain network or a individual node should be able to sustain a considerable quantity of dishonest nodes and still achieve a network consensus.

4) Energy Consumption: is a primary concern due to the potential environmental impact of poorly sourced energy [27]. Proof of Stake blockchains can consume over 75% less energy than Proof of Work, but at a cost of possible equitability [28].

5) Scaleability: is the ability of the distributed ledger to function without any degradation as the chain grows and more users take part in the network. Scaleability is a combination of throughput, storage, and chain growth [25].

6) Cost: of transacting on distributed ledgers is derived by two main types of transaction - deployment and execution [7]. Deployment is the cost of deploying a contract to the blockchain network. Execution is the act of calling that contract and interacting with it, either though reading or writing. Any state change that takes place on a blockchain has an associated execution cost. The larger the state change made, the higher the associated cost.

III. APPLICATIONS OF DLT

The requirement for robots to have robust and secure and communication and the ability to build an agreed distributed consensus provides new opportunities for DLT applications. This section outlines applications of DLT within the robtics domain across industrial robotics, swarm robotics, multi-robot systems and embedded systems.

A. *Industrial Robots*

Smart factories are increasingly being deployed with a reduced workforce powered by smart IoT to cut costs and improve margins [29]. Automatic logging, quality control, data collection and associated utilisation data are now forming part of the new digital economy. As well as improving operational costs, the digitisation of industrial systems may provide a new potential data stream that can be commercialised [30].

Ferandes et al., [31] built on previous work by Lopes et al., [32] on detecting anomalies among robotic events. They present an approach utilising the Tezos blockchain to demonstrate the reduction of bottleneck points in an assembly line by reducing the miss or unintentional behavior of robots. This extends upon the work of Lopes by creating a separate chain, RobotChain, that registers events in closed environments such as factories which they argue is not currently possible in an immutable way. They propose smart contracts as a calling agent to recover a next command once a defined series of actions has been completed. If this sequence does not occur within a permitted time (1 second) it is deemed as failed. A shortcoming of this work is that it is not immediately clear how utilising a smart contact in this way will reduce a bottleneck point rather than just highlight it for attention. Further work including process load balancing may help resolve this.

To address the above challenge, a concept of *robonomics* is introduced by Kapitonov et al., [33]. Their work describes the organisation of a smart factory using multiple autonomous agents. Here, they build on Ethereum and smart contracts to present a robot economy by which agents choose their own behavior and build consensus with one another. They note a key principle of robonomics is the decentralized nature, multiplicity and heterogeneity of an autonomous agent system. They deploy a market token system using messaging exchange over the Interplanetary File System (IPFS) to organise topics for publisher and subscriber. All network agents are automatically subscribed to a *aira_market* topic from which all other topics are then created depending on the needs of the smart factory. An *aira_ros_bridge* was created to perform low level interaction between ROS and the smart contract. Their results show that a successful market place was launched and organised over four factories successfully performing 36 transactions. One potential shortcoming of the work is how the smart contract is organised for scaleablity. Depending on the number of transactions made, message ordering may get misaligned causing the market process to become out of sync.

B. *Swarm Robotics*

Applications of swarm robotics have seen use in military applications [34], search and rescue operations [35], logistics [36], and monitoring and mapping within precision farming [37]. These environments are challenging and are prone to a variety of attacks. Blockchain as a decentralised platform can help alleviate these concerns by validating, using various smart contracts, the automation and data sharing between authorised agents over openly available infrastructure.

Strobel et al., [12] demonstrated a proof of concept to address the challenge of collective decision making among a swarm of byzantine robots. They note that robotic swarm systems often claim high fault tolerance in lab conditions only, and note that real world scenarios are virtually ignored in the presence of byzantine robots, that they define as robots with arbitrarily faulty or malicious behavior. They note that a single byzantine robot could be sufficient to let current swarm coordination mechanisms fail with volatile or detrimental consequences.

They compared two scenarios, a classical scenario and a blockchain scenario where robots moved around a black and white checkerboard to determine the most frequent colour in the environment. In the classical approach the behaviours is determined by a probabilistic finite state machine with explorations states E_i and dissemination D_i states. A random walk is used to explore the board. They use a voting mechanism in the last 30 ticks of the state at timestep i, when in the dissemination state to receive opinions from other robots on the most frequent colour on the board. The blockchain scenario replicates the classical scenario, but upon decision making each robot issues its own state to be mined through a PoW action. When two robots are near one another, their own copy of the chains combine and the longest chain will be seen as the truth. Their results show that in the presence of byzantine robots the classical approach will converge to the wrong colour if the simulation is not stopped at a sub-swarm consensus. However, they demonstrate that during the blockchain approach, consensus is achieved and the right colour chosen, in a fully decentralised way. They note a shortcoming of the work is that PoW will only secure the system if no individual robot achieves a majority hash rate. A recommendation to mitigate this, is Proof of *Physical* Work.

C. Multi Robot Systems

Multi robot systems are similar to swarm systems but lack a decentralised system at the operational level [38]. This allows them to have a broader scope than swarms as one or several robots can complete all of the tasks required. This provides them different modes of flexibility as they can contain heterogeneous robots. Today, there are a multitude of these systems in operation ranging from underwater multi-robot system [39] to wind turbine maintenance [40].

Basegio et al., [41] proposed an communication architecture to enable a dynamic and decentralised task allocation system. They argue that the use of a centralised planning system has a single point of failure that real world multi complex task systems cannot tolerate. They devise an architecture to overcome this constraint using a Belief-Desire-Intention model [42]. They utilise a blockchiain controller to allow communication between agents that provides a set of actions that each agent can be allocated depending upon its capability. Results are presented for a flooding case study where task allocation is based on a bidding process using a Monte Carlo method. Once a bid was won, the task was sent to the robot as a header file within a block. Shortcomings of this work is that it is a very high level theoretical framework, with little detail on the implementation. This makes it almost impossible to repeat or verify any of the results presented.

Gou et al., [43] present an improved implementation using a micro-blockchain and a LoRa [44] communication protocol. The micro-blockchain consisted of seven parts, which are composed of ID (Robot Number), INDEX (block number), TIME, CONTENT (sharing data), SELFHASH (block hash), PREHASH (front block hash), CRC16 (16-bit CRC check), respectively. The chain structure was formed by HASH value and stored on each robot node. A novel heartbeat function was used to control block generation. The objective of the system is to assign tasks via a voting method that takes place through mining. They show that a simple two robot experimental system can come to an agreement on task allocation within 11 blocks generated over 3 seconds by 3 nodes. A shortcoming of this work is scaleability. They concede that system stability is unknown and requires formal verification to fully validate.

Queralta and Westerlund [45], propose a proof of work system to enable an online estimation of the available computational resources at different robots within a multi swarm system. They then build upon this to define smart contracts that integrate information about the environment from different robots in order to evaluate and rank the quality and accuracy of each of the robots' sensor data. They utilise 3D lidar data from a 32-channel Velodyne laser scanner, to obtain environmental feedback. They then push this through an Ethereum smart contract as a data payload. It is not fully clear what the benefit of this system is, however the authors did note that many of the aspects required were first stage research and not yet available in current blockchains and so remains theoretical work.

D. Embedded Systems

Systems on a Chip (SoC) and Single Board Computers (SBC) make a up a large amount of embedded systems. We see embedded systems around us everyday with the growing proliferation of IoT and rich connectedness. These twenty four seven online systems are direct lines of sight for attack and threat vectors. Accountability is a desirable property of distributed systems that enables the detection, identification, and removal of faulty or malicious behavior [46]. Without a decentralised network, a robot will primarily depend upon external computational devices for completing specific tasks, leading to security, safety and response delay issues.

To address the above challenges Falcone et al., [47] present a blockchain based mapping protocol for distributed robotic systems running on embedded hardware. The protocol was based on a robotic system designed to move on a lattice structure framework for space applications. A novel consensus mechanism, Proof of Validity (PoV), was introduced to allow the effort of mining blocks to correlate with the desired tasks the robotic system was designed for. Similar to Gou, robot communication was achieved using peer to peer LoRa radio. Their results show that PoV is a means of constructing a distributed mapping framework. The consensus algorithm allowed the robots to traverse the full surface without collision with enough exploration to enable the map to be incorporated into the PoV blockchain. Several trade offs were noted, such as blockchain forks, hardware requirements for low power embedded systems, block processing requirements and any form of long term memory which is critical for chain validation.

IV. DLT INTEGRATION INTO ROBOTIC MIDDLEWARE

Suitable interfacing to robot middleware is required to enable DLT applications to provide higher level functionality - such as smart contracting and consensus building - that extend little attention has be applied to what these interfaces. This section outlines the current s of DLT across robotic middleware, specifically the Robot Operating System, Lightweight Communications and Marshalling, ZeroMQ and Open Robot Control Software.

A. The ROS operating System

The Robot Operating System (ROS) [23] is set of software libraries and tools used to build robotics systems. Its purpose is to provide a structured communication layer on top of the host operating system of a heterogeneous compute cluster [48]. The philosophical goals of ROS are: 1. *peer to peer*, 2. *tools based*, 3. *multi-lingual*, 4. *thin* and 5. *free and open source*. In many respects, these philosophical aspirations are shared within the blockchain community. ROS utilises a nodes based structure which communicate via messages. Messages are sent by publishing to a relevant topic. Any node that is interested in these messages then subscribes to the topic, as shown in Fig. 2. There is no limit to the number of topics that can be subscribed to (barring bandwidth) or messages sent. Systems can be packed to enable an out of the box plug and play type scenario. Although increasing, a limited number of applications of ROS and DLT have been found including a ROS-Ethereum covenant bridging tool [49], ROS-Ethereum bridge for smart factories [33], a ROS2-Hyperledger Fabric application [50], and a Multyi-UAV system using ROS1 [51].

B. Lightweight Communications and Marshalling LCM

The Lightweight Communications and Marshalling (LCM) [52] library is a message passing and data marshalling protocol. The objective of LCM is to simplify the development of low latency message passing systems, with a focus on real time robotics applications. Messages can be transmitted between different processes using LCM's publisher and subscriber message passing system, similar to ROS. A platform and language independent type specification language separates message description from implementation [53]. LCM has achieved popularity by providing a real time deep message inspection tool that can decode and display message traffic with little user effort and minimal system resource requirements. LCM has been used across a range of domains including land, underwater and aerial robots, including forklift trucks [54] and indoor robots. To the best of the authors knowledge, no applications have yet been developed utilising blockchain technology with LCM.

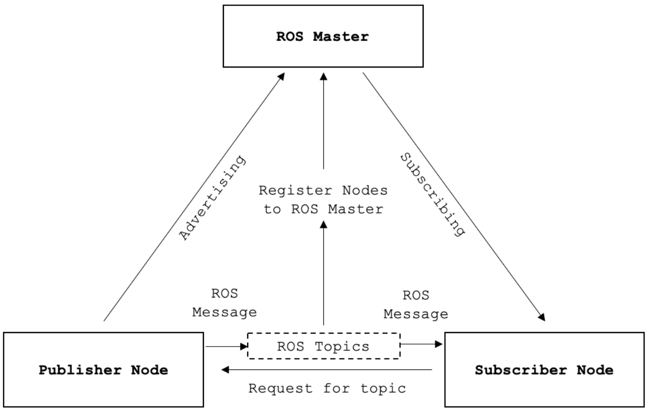

Fig. 2. ROS is a nodes based structure which communicates via messages and managed via a ROS Master node. Messages are sent by publishing to a relevant topic. Any node that is interested in a message then subscribes to the topic. New topics can be created which can then be subscribed to.

C. ZeroMQ

ZeroMQ [55] is an open-source networking library originally developed by iMatix under the LGPLv3 license. Although not broadly used as ROS, ZeroMQ has been used widely in robotics ranging from social robots [56] to cloud enabled humanoid robots [57]. The ZeroMQ library provides a compact and simple socket API that can be used to establish in-process, inter-process or inter-host using TCP or multicast communication. It supports various types of paradigms from request/reply to publisher/subscriber models, task distribution and fan-out using the ZMTP protocol (ZeroMQ Message Transfer Protocol) [58]. Smart use of message batching, asynchronous communication and support for zero copy makes it one of the most efficient libraries for creating distributed applications. ZeroMQ core is written in C/C++ but it has bindings and native ports for modern languages and operating systems. Very limited implementations of blockchain technology utilising ZeroMQ within a robotics domain have been demonstrated, however a Blockchain-based Decentralized Forward-Trading Energy Exchange for Transactive Microgrids [59], was an interesting use case but outside of the scope of robotic systems.

D. Open Robot Control Software

The Open Robot Control Software (OROCOS) [60], is an open source, general purpose robot control software package. The main objective of OROCOS is be a framework for real-time safe interaction between individual software components in a robotics system, and to establish a library of ready-to-use components. As a software project its objectives are to be modular, flexible, high quality, independent of commercial robot manufacturers and straightforward to use for all robotic and computer platforms. The OROCOS Real-Time Toolkit (RTT) and Component Library (OCL) are actively developed and used widely by research groups and companies. Components of the OROCOS ecosystem include the Bayesian Filtering Library (BFL) and the Kinematics and Dynamics Library (KDL). OROCOS has integrations with ROS1 and ROS2, however is not as popular as the ROS platform. To the best of the authors knowledge, no applications have yet been developed utilising blockchain technology with OROCOS.

V. CHALLENGES OF DLT WITHIN ROBOTICS

As the technology used within blockchains continues to mature several high level challenges are still present. Performance due to throughput constraints, scaleability due to chain growth, and appropriate blockchain selection are a few of the most pressing. This section briefly presents a high level overview of these challenges and presents some initial considerations on how these may resolved.

A. *Performance*

Network performance depends upon factors such as throughput and latency. Middleware systems such as ROS that utilises a publisher-subscriber model requires higher messaging throughput than most current blockchains can provide block capacity for. Although within ROS there are queuing mechanism, this would further add complexity in ordering the messages within a block itself which may impact order execution of smart contracts. This may be mitigated as new technologies extend upon current blockchain architecture with new layer two blockchain protocols [61].

B. *Scaleablity*

Blockchain storage requirements grow as more participants, such as robots or associated middleware nodes utilise the network. This increasing demand is placed upon the network operators to ensure that they are keeping up with chain growth. For example, validating the most recent block - or tip - of the chain, and ensuring that the storage of the chain is available for constant validation and interrogation by smart contracts is one such requirement. Additional to extending a blockchain into further layers, developing new techniques to parallelise transactions via techniques such as sharding [62] - splitting the storage of the ledger and/or data used to recreate the ledger across many shards - or utilising a sidechain [63] - a secondary blockchain connected to the main blockchain via a pegging mechanism - can support scaleablity requirements. However, potential tradeoffs could be reduced security or decentralisation of the network [2].

C. *Selection*

As the blockchain ecosystem has evolved, the requirement to standardise a method to identify blockchain technology maturity has grown. The Technology Readiness Level (TRL) [64] is one example of a standard to assess technology maturity and is used across many domains such as aerospace [65], defence [66], and machine learning systems [67]. Although work has taken place to begin to categories current blockchain applications into the TRL scale [68], to the best of the authors knowledge no TRL or equivalent methodology has been used to directly assess blockchain maturity. A mechanism to support blockchain selection using a formal process such as a TRL would benefit DTL development across the Robotics and middleware ecosystems.

VI. DISCUSSION

Distributed ledger technology and its applications within robotics is a developing field of research across a broad range of robotic and system domains. Approaches from industrial robots to swarm robotics have illustrated the diverse challenges DLTs can be applied towards. Advances in security through the application of smart contracts as well as group consensus and decision making have extended on the state-of-the-art through a decentralised lens on what has classically been a centralised problem. Such applications require considered thought towards the choice of blockchain a robotic system will utilise. The choice of blockchain has mainly been driven by research funding, such as RobotChain, supported by Tezoz Foundation, or ad-hoc choice based on developer activity or available documentation to create a proof of concept. This could prove to be a substantial bottleneck for future development.

The integration layer of DLT's across a variety of middleware applications for robotics and systems control is evident. Work has trended towards larger and more supported middleware systems such as ROS. Very few integrations were found on older or less supported networking and messaging protocols such as ZeroMQ, LCM or OROCOS. A majority of the works surveyed that had any DLT integration into a middleware package, were project specific with little opportunity for integration. This makes it extremely difficult to build on past projects, particular as available solutions tended to be purpose built for a specific need. The lack of a standard ROS bridging package is a potential challenge towards further adoption of DLTs within robotic operating systems.

VII. CONCLUSION

In this review, we examined DLT's and their applications within robotics and associated middleware tools. We found a broad scope of DLT implementations ranging from swarm robotics to industrial robot applications. The Robot Operating System had the most usage for building applications with DLT's. This review showed that although DLT research is progressing within a robotic domain, there are still several challenges; particularly when assessing DLT maturity for development upon. The application of DLT's towards robotics would benefit from a standardised and coherent readiness level for DLT's similar to a Technology Readiness Level. Additionally, future work towards building a standardised bridge between the most popular robotic operating systems, such as ROS, using DLTs with high readiness levels would provide value to the community by enabling reuse of modules and packages between related projects.

ACKNOWLEDGMENT

This work was supported by the Engineering and Physical Sciences Research Council [EP/S023917/1].

REFERENCES

[1] M. M. Nair, A. K. Tyagi, and N. Sreenath, "The future with industry 4.0 at the core of society 5.0: open issues, future opportunities and challenges," in *2021 International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1–7, IEEE, 2021.

[2] J. Al-Jaroodi and N. Mohamed, "Blockchain in industries: A survey," *IEEE Access*, vol. 7, pp. 36500–36515, 2019.

[3] Z. Dobra and K. S. Dhir, "Technology jump in the industry: Human–robot cooperation in production," *Industrial Robot: the international journal of robotics research and application*, 2020.

[4] B. C. Kok and H. Soh, "Trust in robots: Challenges and opportunities," *Current Robotics Reports*, vol. 1, no. 4, pp. 297–309, 2020.

[5] P. Bhoyar, P. Sahare, S. B. Dhok, and R. B. Deshmukh, "Communication technologies and security challenges for internet of things: A comprehensive review," *AEU-International Journal of Electronics and Communications*, vol. 99, pp. 81–99, 2019.

[6] G. Deng, Y. Zhou, Y. Xu, T. Zhang, and Y. Liu, "An investigation of byzantine threats in multi-robot systems," in *24th International Symposium on Research in Attacks, Intrusions and Defenses*, pp. 17–32, 2021.

[7] U. S. Aditya, R. Singh, P. K. Singh, and A. Kalla, "A survey on blockchain in robotics: Issues, opportunities, challenges and future directions," *Journal of Network and Computer Applications*, vol. 196, p. 103245, 2021.

[8] S. Gupta and M. Sadoghi, "Blockchain transaction processing.," 2019.

[9] S. Bano, M. Al-Bassam, and G. Danezis, "The road to scalable blockchain designs," *USENIX; login: magazine*, 2017.

[10] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *International Journal of Web and Grid Services*, vol. 14, no. 4, pp. 352–375, 2018.

[11] S. Pearson, D. May, G. Leontidis, M. Swainson, S. Brewer, L. Bidaut, J. G. Frey, G. Parr, R. Maull, and A. Zisman, "Are distributed ledger technologies the panacea for food traceability?," *Global food security*, vol. 20, pp. 145–149, 2019.

[12] V. Strobel, E. Castelló Ferrer, and M. Dorigo, "Managing byzantine robots via blockchain technology in a swarm robotics collective decision making scenario," 2018.

[13] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, "A survey on consensus mechanisms and mining strategy management in blockchain networks," *IEEE Access*, vol. 7, pp. 22328–22370, 2019.

[14] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," in *Concurrency: the Works of Leslie Lamport*, pp. 203–226, 2019.

[15] N. Stifter, A. Judmayer, P. Schindler, A. Zamyatin, and E. Weippl, "Agreement with satoshi–on the formalization of nakamoto consensus," 2018.

[16] C. T. Nguyen, D. T. Hoang, D. N. Nguyen, D. Niyato, H. T. Nguyen, and E. Dutkiewicz, "Proof-of-stake consensus mechanisms for future blockchain networks: fundamentals, applications and opportunities," *IEEE Access*, vol. 7, pp. 85727–85745, 2019.

[17] D. Larimer, "Delegated proof-of-stake (dpos)," *Bitshare whitepaper*, 2014.

[18] H. sawtooth, *PoET 1.0 Specification*, 2017 (accessed August 19, 2020).

[19] L. Bach, B. Mihaljevic, and M. Zagar, "Comparative analysis of blockchain consensus algorithms," in *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 1545–1550, IEEE, 2018.

[20] N. Chaudhry and M. M. Yousaf, "Consensus algorithms in blockchain: comparative analysis, challenges and opportunities," in *2018 12th International Conference on Open Source Systems and Technologies (ICOSST)*, pp. 54–63, IEEE, 2018.

[21] S. Wang, Y. Yuan, X. Wang, J. Li, R. Qin, and F.-Y. Wang, "An overview of smart contract: architecture, applications, and future trends," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 108–113, IEEE, 2018.

[22] N. Szabo, "Formalizing and securing relationships on public networks," *First Monday*, 1997.

[23] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, *et al.*, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, p. 5, Kobe, Japan, 2009.

[24] M. Kuzlu, M. Pipattanasomporn, L. Gurses, and S. Rahman, "Performance analysis of a hyperledger fabric blockchain framework: throughput, latency and scalability," in *2019 IEEE international conference on blockchain (Blockchain)*, pp. 536–540, IEEE, 2019.

[25] Q. Zhou, H. Huang, Z. Zheng, and J. Bian, "Solutions to scalability of blockchain: A survey," *Ieee Access*, vol. 8, pp. 16440–16455, 2020.

[26] H. Sukhwani, J. M. Martínez, X. Chang, K. S. Trivedi, and A. Rindos, "Performance modeling of pbft consensus process for permissioned blockchain network (hyperledger fabric)," in *2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS)*, pp. 253–255, IEEE, 2017.

[27] W. J. Sutherland, P. W. Atkinson, S. Broad, S. Brown, M. Clout, M. P. Dias, L. V. Dicks, H. Doran, E. Fleishman, E. L. Garratt, *et al.*, "A 2021 horizon scan of emerging global biological conservation issues," *Trends in Ecology & Evolution*, vol. 36, no. 1, pp. 87–97, 2021.

[28] R. Zhang and W. K. V. Chan, "Evaluation of energy consumption in block-chains with proof of work and proof of stake," in *Journal of Physics: Conference Series*, vol. 1584, p. 012023, IOP Publishing, 2020.

[29] L. D. Evjemo, T. Gjerstad, E. I. Grøtli, and G. Sziebig, "Trends in smart manufacturing: Role of humans and industrial robots in smart factories," *Current Robotics Reports*, vol. 1, no. 2, pp. 35–41, 2020.

[30] W. Mao, Z. Zheng, and F. Wu, "Pricing for revenue maximization in iot data markets: An information design perspective," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pp. 1837–1845, IEEE, 2019.

[31] M. Fernandes and L. A. Alexandre, "Robotchain: Using tezos technology for robot event management," *Ledger*, 2019.

[32] V. Lopes and L. A. Alexandre, "Detecting robotic anomalies using robotchain," in *2019 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pp. 1–6, IEEE, 2019.

[33] A. Kapitonov, I. Berman, V. Bulatov, S. Lonshakov, and A. Krupenkin, "Robonomics based on blockchain as a principle of creating smart factories," in *2018 Fifth International Conference on Internet of Things: Systems, Management and Security*, pp. 78–85, IEEE, 2018.

[34] H. Zhao, H. Liu, Y.-W. Leung, and X. Chu, "Self-adaptive collective motion of swarm robots," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 4, pp. 1533–1545, 2018.

[35] M. Bakhshipour, M. J. Ghadi, and F. Namdari, "Swarm robotics search & rescue: A novel artificial intelligence-inspired optimization approach," *Applied Soft Computing*, vol. 57, pp. 708–726, 2017.

[36] Y. Liu, L. Wang, H. Huang, M. Liu, and C.-z. Xu, "A novel swarm robot simulation platform for warehousing logistics," in *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 2669–2674, IEEE, 2017.

[37] D. Albani, J. IJsselmuiden, R. Haken, and V. Trianni, "Monitoring and mapping with robot swarms for agricultural applications," in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–6, IEEE, 2017.

[38] M. Chamanbaz, D. Mateo, B. M. Zoss, G. Tokić, E. Wilhelm, R. Bouffanais, and D. K. Yue, "Swarm-enabling technology for multi-robot systems," *Frontiers in Robotics and AI*, vol. 4, p. 12, 2017.

[39] Z. Zhou, J. Liu, and J. Yu, "A survey of underwater multi-robot systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 1, pp. 1–18, 2021.

[40] S. Manjanna, A. Q. Li, R. N. Smith, I. Rekleitis, and G. Dudek, "Heterogeneous multi-robot system for exploration and strategic water sampling," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4873–4880, IEEE, 2018.

[41] T. L. Basegio, R. A. Michelin, A. F. Zorzo, and R. H. Bordini, "A decentralised approach to task allocation using blockchain," in *International Workshop on Engineering Multi-Agent Systems*, pp. 75–91, Springer, 2017.

[42] M. Georgeff, B. Pell, M. Pollack, M. Tambe, and M. Wooldridge, "The belief-desire-intention model of agency," in *International workshop on agent theories, architectures, and languages*, pp. 1–10, Springer, 1998.

[43] S. Guo, S. Cao, and J. Guo, "Study on collaborative algorithm for a spherical multi-robot system based on micro-blockchain," in *2019 IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 1465–1470, IEEE, 2019.

[44] M. Bor, J. E. Vidler, and U. Roedig, "Lora for the internet of things," 2016.

[45] J. P. Queralta and T. Westerlund, "Blockchain-powered collaboration in heterogeneous swarms of robots," *arXiv preprint arXiv:1912.01711*, 2019.

[46] M. Chiu, A. Goldsmith, and U. Kalabić, "Blockchain for embedded system accountability," in *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pp. 1–5, IEEE, 2021.

[47] S. Falcone, J. Zhang, A. Cameron, and A. Abdel-Rahman, "Blockchain design for an embedded system," *Ledger*, 2019.

[48] M. Cashmore, M. Fox, D. Long, D. Magazzeni, B. Ridder, A. Carrera, N. Palomeras, N. Hurtos, and M. Carreras, "Rosplan: Planning in the robot operating system," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 25, pp. 333–341, 2015.

[49] S. Zhang, M. Tang, X. Li, B. Liu, B. Zhang, F. Hu, S. Ni, and J. Cheng, "Ros-ethereum: A convenient tool to bridge ros and blockchain (ethereum)," *Security and Communication Networks*, vol. 2022, 2022.

[50] J. P. Queralta, T. Westerlund, *et al.*, "Towards managing industrial robot fleets with hyperledger fabric blockchain and ros˜ 2," *arXiv preprint arXiv:2203.03426*, 2022.

[51] M. G. S. De Campos, C. P. Chanel, C. Chauffaut, and J. Lacan, "Towards a blockchain-based multi-uav surveillance system," *Frontiers in Robotics and AI*, vol. 8, 2021.

[52] A. S. Huang, E. Olson, and D. C. Moore, "Lcm: Lightweight communications and marshalling," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4057–4062, IEEE, 2010.

[53] D. Moore, E. Olson, and A. Huang, "Lightweight communications and marshalling for low-latency interprocess communication," 2009.

[54] S. Teller, M. R. Walter, M. Antone, A. Correa, R. Davis, L. Fletcher, E. Frazzoli, J. Glass, J. P. How, A. S. Huang, *et al.*, "A voice-commandable robotic forklift working alongside humans in minimally-prepared outdoor environments," in *2010 IEEE International Conference on Robotics and Automation*, pp. 526–533, IEEE, 2010.

[55] J. Lauener, W. Sliwinski, and G. CERN, "How to design & implement a modern communication middleware based on zeromq," in *Proc of ICALEPCS*, vol. 17, pp. 45–51, 2017.

[56] E. Coronado, F. Mastrogiovanni, and G. Venture, "Development of intelligent behaviors for social robots via user-friendly and modular programming tools," in *2018 IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO)*, pp. 62–68, IEEE, 2018.

[57] D. M. Lofaro, A. Asokan, and E. M. Roderick, "Feasibility of cloud enabled humanoid robots: Development of low latency geographically adjacent real-time cloud control," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pp. 519–526, IEEE, 2015.

[58] J. Moore, A. Arcia-Moret, P. Yadav, R. Mortier, A. Brown, D. McAuley, A. Crabtree, C. Greenhalgh, H. Haddadi, and Y. Amar, "Zest: Rest over zeromq," in *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pp. 1015–1019, IEEE, 2019.

[59] A. Laszka, S. Eisele, A. Dubey, G. Karsai, and K. Kvaternik, "Transax: A blockchain-based decentralized forward-trading energy exchanged for transactive microgrids," in *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 918–927, IEEE, 2018.

[60] H. Bruyninckx, "Open robot control software: the orocos project," in *Proceedings 2001 ICRA. IEEE international conference on robotics and automation (Cat. No. 01CH37164)*, vol. 3, pp. 2523–2528, IEEE, 2001.

[61] L. Gudgeon, P. Moreno-Sanchez, S. Roos, P. McCorry, and A. Gervais, "Sok: Layer-two blockchain protocols," in *International Conference on Financial Cryptography and Data Security*, pp. 201–226, Springer, 2020.

[62] H. Dang, T. T. A. Dinh, D. Loghin, E.-C. Chang, Q. Lin, and B. C. Ooi, "Towards scaling blockchain systems via sharding," in *Proceedings of the 2019 international conference on management of data*, pp. 123–140, 2019.

[63] A. Singh, K. Click, R. M. Parizi, Q. Zhang, A. Dehghantanha, and K.-K. R. Choo, "Sidechain technologies in blockchain networks: An examination and state-of-the-art review," *Journal of Network and Computer Applications*, vol. 149, p. 102471, 2020.

[64] J. Straub, "In search of technology readiness level (trl) 10," *Aerospace Science and Technology*, vol. 46, pp. 312–320, 2015.

[65] M. J. Ward, S. T. Halliday, and J. Foden, "A readiness level approach to manufacturing technology development in the aerospace sector: an industrial approach," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 226, no. 3, pp. 547–552, 2012.

[66] B. Sauser, J. E. Ramirez-Marquez, R. Magnaye, and W. Tan, "A systems approach to expanding the technology readiness level within defense acquisition," tech. rep., STEVENS INST OF TECH HOBOKEN NJ SCHOOL OF SYSTEMS AND ENTERPRISES, 2009.

[67] A. Lavin, C. M. Gilligan-Lee, A. Visnjic, S. Ganju, D. Newman, S. Ganguly, D. Lange, A. G. Baydin, A. Sharma, A. Gibson, *et al.*, "Technology readiness levels for machine learning systems," *Nature Communications*, vol. 13, no. 1, pp. 1–19, 2022.

[68] K. Holm and R. C. Goduscheit, "Assessing the technology readiness level of current blockchain use cases," in *2020 IEEE Technology & Engineering Management Conference (TEMSCON)*, pp. 1–6, IEEE, 2020.