The use of Blockchain technology to achieve web scale agent-based simulations: A Literature Review

Conor Deegan[1][0000−0001−7835−2696]

School of Computer Science, University College Dublin, Dublin, Ireland
conor.deegan@ucdconnect.ie

1 Introduction

Agent-Based Modelling (ABM) is a bottom-up approach to studying the behaviour of complex systems [42]. Systems are modelled as a collection of individual agents where each agent encapsulates private state and can make decisions based on a set of rules [6]. The behaviour of the system emerges through the decisions made by these agents and their interactions with one another. ABM has been applied in many domains including logistics [20], biology, ecology, and social science [39].

One of the key challenges in ABM is the ability to model systems of increasing complexity [14]. Work has shown that more complex ABM cannot be captured by a single model on a single machine [33]. Current solutions to this scaling problem include the use of Hybrid Simulation (HS) which combines multiple interconnected sub-simulations where each simulation may be implemented using different modelling techniques [22]. The work in [14] builds upon this further by exploring the use of loosely-coupled microservices to implement HS at scale.

In recent years, there has been increasing concerns around centralised web services in areas such as privacy, governance, surveillance, and security. This has triggered the emergence of new distributed technologies such as blockchain technology (BCT) [51]. Blockchains, such as the Ethereum blockchain, come equipped with a built in Turing-complete programming language which can be used to create decentralised applications which are executed publicly on a global network of decentralised nodes.

The goal of this work is to explore the use of BCT to achieve web scale agent-based simulations. This paper explores existing literature and provides an overview of ABM; issues with, and current solutions for, scaling ABMs; an overview of BCT; and existing work utilising BCT for scaling ABMs.

2 Agent Based Simulations

2.1 Introduction

The history of ABM can be traced back to the work of John von Neumann in the 1960's and his theory of self-reproducing automata, self operating machines used to synthesize natural systems [26].

In ABM, a system is modeled as a collection of autonomous decision-making entities called agents. Each agent individually assesses its situation and makes decisions on the basis of a set of rules [6]. The behaviour of the system emerges through the decisions made by these agents and their interactions. ABMs are used in many scientific domains including biology, ecology and social science [39].

Overview

Agents At it's core, ABM is about creating a set of agents that are situated in, and interact with, an environment according to a set of rules in order to realise some global system behaviour. There is no universal agreement amongst researches on the exact definition of an agent [16]. However, a popular definition was proposed by Wooldridge and Jennings in their paper "Intelligent agents: theory and practice". They characterise an agent as an encapsulated computer system, situated in some environment, and capable of flexible autonomous action in that environment in order to meet its design objectives [55]. Some features that are common to agents are discussed below [55] [23] [16]:

- Autonomy: agents are autonomous units, capable of processing information and exchanging this information with other agents in order to make independent decisions.
- Heterogeneity: agents permit the development of autonomous individuals. Although groups of agents can exist, they are amalgamations of similar autonomous individuals.
- Active: agents are active because they exert influence within a simulation.

Within an application, some of these features of an agent will be more important than others [55]. It can also be the case that there are many different types of agents within one simulation [16].

Rules and Behaviour Agents typically poses rules which govern their behaviour and interaction with other agents and their surrounding environment. Agent Based Models typically simulate the behaviour of agents over discrete time points whereby the execution of the model is synchronized at each point in time. This means that each agent can be thought of as some form of Finite State Automata with a state and the ability to transition this state at each time step.

Environment The environment is the shared context for the execution of agent behaviours. The environment is responsible for scoping the agent behaviours, mediating access to resources, and mediating interactions between other agents.

Advantages of Agent-Based Modelling In their work, Crooks and Heppenstall [16] propose that an agent based approach has three main advantages over traditional modelling techniques. They state that the agent based approach captures emergent phenomena, provides a natural environment for the study of

certain systems, and is flexible. Other works [4] [24] describe the pragmatic advantages of agent-based models as being more akin to reality when compared to other modelling techniques, defining them as a suitable method for simulating systems composed of real-world entities.

2.2 Scaling Agent Based Models

Overview A key challenge in the area of ABM is the ability to model systems of increasing complexity [14]. Although there are a range of approaches to implementing ABMs [1], they have been traditionally viewed as a desktop computer style of exercise where simulations are executed on a single machine [14]. However, work has shown that desktop agent based models do not scale to what is required for extremely large applications in the study of realistic complex systems [2] [33].

Components of ABMs that contribute to slower execution times and memory issues when scaling to larger simulations include; a larger number of agents, model design, and computational complexity of the agents behaviour [13]. This can lead to model simplifications until the execution time is acceptable on a single machine [50].

ABMs which push the limits of resources due to their large numbers of agents or their complexity may be referred to as 'Massively Multi-agent Systems (MMAS)' or 'Massive Agent-based Systems (MABS)' [31] [29].

Current Solutions The research in [41] suggests the following techniques to handle ABM at scale:

- Reduce the number of agents, or level of agent complexity, in order for model to run on existing hardware.
- Revert to a population based modelling approach.
- Invest in a larger or faster serial machine.
- Run the model on a vector computer.
- Invest in a large scale computer network and reprogram the model in parallel.

Of these suggestions, the first two propose decreasing the complexity of the simulation in order for it to run effectively on a single machine. However, this would limit the effectiveness of ABM in the study of realistic complex systems. The third and fourth suggestions are not truly scalable. A larger single machine may be suitable for a current complex ABM, but as these systems grow in further complexity larger machines will be required. As a result, this is not a cost-effective and scalable solution. The final suggestion, to run such models in parallel, was also researched in [13]. However, several key challenges arise when implementing an agent model in parallel. These include load balancing, synchronising events to ensure causality, monitoring of the distributed simulation, and managing communication between nodes [52]. Each of these may negatively affect the increase in performance achieved.

Another proposed solution to handle this greater complexity is the use of Hybrid Simulation [22]. Hybrid Simulation combines multiple interconnected sub-simulations where each simulation may be implemented using different modelling techniques [37]. Within the simulation community, two main approaches have emerged; Distributed Simulation and Cloud Based Simulation.

The Distributed Simulation (DS) approach focuses on the development and deployment on high-performance computing clusters, leading to tools such has RePAST HPC [12]. However, these implementations are typically bespoke and tailored to specific tasks [49]. Thus, a DS approach does not solve issues such as interoperability and model reuse that are required for large-scale ABM.

Another criticism of DS is the cost and availability of computing clusters. This has led to the proposal of Cloud Based Simulation (CBS) [49], which focuses on the deployment of simulations in the cloud using microservices [50].

However, research has shown that there is a lack of suitable tools and frameworks for integrating ABMs with other technologies [42]. Interoperability is a particular issue leading to many existing Hybrid Simulations being built using a single tool [22].

The research in [14] argues that Hybrid Simulations should not be built on monolithic architectures but instead implemented as loosely-coupled microservices in a manner that ensures scalability. They highlight that the use of microservices to achieve greater scale in ABM allows for reusable components, interoperability, polyglot development, and deployment at scale. They propose that each sub-simulation is encapsulated as a microservice that uses REST for communication and can be integrated into larger simulations that are not agent-based. This has been demonstrated using Multi-Agent Microservices (MAMS) in the ASTRA agent programming language [53] [40] [15] [17]. Of the proposed solutions to best handle scaling ABMs, this research seems to be the most viable as it is in line with current software architecture best-practises [19] and benefits directly from the advantages of a microservice architecture [27].

3 Blockchain

3.1 Introduction

History Although blockchains were first conceived in the early 1990's [25], the technology began to receive widespread interest with the release of the Bitcoin protocol by Satoshi Nakamoto in 2008 [38]. The Bitcoin protocol allows for peer-to-peer transactions in a fully trust-less environment. It does this through the use of digital signatures, proof-of-work [21], and a distributed ledger to maintain the state of the network. Crucially, the Bitcoin protocol solves the double-spending problem [38] - a potential flaw in a digital cash scheme in which the same single digital token can be spent more than once.

Overview

Distributed ledger A distributed ledger is a database spread over several nodes on a peer-to-peer network, where each node replicates and maintains an identical copy of the ledger. When a new transaction occurs each node constructs the new transaction and the nodes vote by consensus algorithm on which copy is correct. Once a consensus has been determined, all of the other nodes update themselves with the new, correct copy of the ledger.

Wallets To interact with a blockchain, users must hold a digital wallet. A digital wallet is designed to store digital currencies and, importantly, has an associated public and private key pair. The public key is used as an address to publicly identify the wallet and to receive digital currencies. The private key is used to initiate transactions and prove ownership of the wallet.

Transactions An simple transaction is shown in Figure 1 on page 6. If Alice wishes to transfer some digital currency to Bob from her wallet, she must first obtain Bob's wallet address. Alice can now create a transaction containing the amount of the digital currency she wishes to send, the network fee she is willing to pay, and Bob's public wallet address. Before broadcasting this transaction to the blockchain, Alice must sign this transaction with her wallet's private key. This provides cryptographic proof that Alice owns her wallet. Once the transaction is submitted to the blockchain, the transaction is verified by miners. Once verified, the block containing the transaction is added to the blockchain and Bob receives the digital currency sent by Alice.

Consensus Decentralised networks such as Bitcoin and Ethereum use proof-of-work as a foundation for consensus. The proof-of-work algorithm is a form of cryptographic proof in which one party proves to others that a certain amount of non-trivial computational effort has been expended. Other parties can subsequently confirm this expenditure trivially [30].

The following explanation is based on information provided in the Ethereum Whitepaper [7]. When a transaction is broadcast to the network, it is combined with other unconfirmed transactions into a block. This block contains the list of unconfirmed transactions, a link to the previous block, and a random nonce. Miners race to find this random nonce such that the computed hash of the block and the nonce begins with a specified number of zeros. Finding this nonce is computationally expensive and ensures that a certain amount of compute power is required to validate a block. Once a valid nonce has been found, it is trivial for other miners to verify that this nonce meets the specified requirements and that the block is valid. In return for their computational effort, the miner who finds the correct nonce receives the block reward - some digital currency native to the blockchain e.g Ether.

3.2 Ethereum

Overview Ethereum was created as an alternative protocol that allows for building decentralised applications [7]. One of the core differences between Ethereum

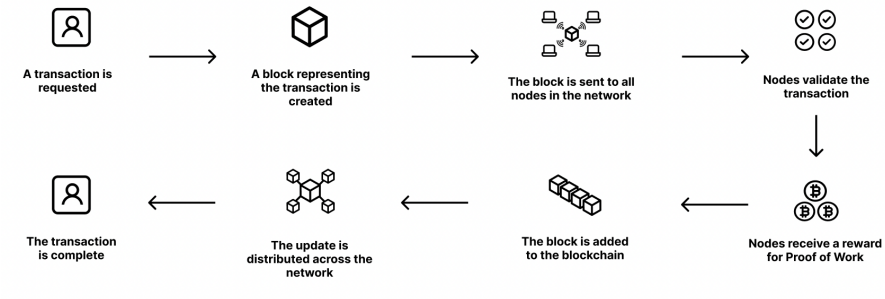

Fig. 1. A blockchain transaction

and other protocols, such as Bitcoin, is that Ethereum provides a blockchain with a built in Turing-complete programming language that can be used to create "smart contracts".

Smart Contracts Smart contracts are the building blocks of Ethereum applications. They are computer programs that are stored and run on the Ethereum blockchain. Fundamentally, a smart contract is a collection of logical functions and state that resides at a specific address on the network. Smart contracts are a type of Ethereum account, meaning they have an Ethereum balance and can send transactions over the network. User accounts can interact with a smart contract by submitting transactions that execute a function defined on the contract [28].

Ethereum Virtual Machine The Ethereum Virtual Machine (EVM) is the runtime environment for transaction execution in Ethereum. Ethereum's state is a large data structure which holds accounts, balances, and a machine state which can execute machine code. The EVM is responsible for defining the rules for computing new valid network states from block to block. This ensures that the Ethereum network has only one valid state at any point in time [54].

Scalability Peer-to-peer (P2P) networks have received an increase in attention in recent years due to emergent blockchain technologies (BCT) [38] [7]. P2P systems are scalable by design. In a P2P network, each node acts as a server which allows the distributed system to grow linearly with the number of nodes available [43]. In a typical P2P network, this is augmented by the fact that no single node must maintain a global knowledge of the system, allowing each node to act on a smaller subsection of the system [32].

The Ethereum network is a P2P network in which [35] estimated there to be approximately 300,000 nodes on the network at the time of their research. Each of these nodes acts as a server running the EVM. This allows for cost-effective scale without the need for application developers to manage a large

cluster of servers. This provides a good argument for leveraging existing open P2P protocols for efficiently running applications at scale.

As it stands, the Ethereum network does not implement the partitioning of network data across different nodes. This means that each node is required to maintain a global state of the network at any point in time. Given this, along with the POW consensus algorithm used, the Ethereum network is currently limited to 15 transactions per second (TPS). A comparison that can be made is to Visa, who as of 2018, was processing roughly 24,000 transactions per second [5]. Application layer solutions, knows as Layer-2 solutions, are being developed to allow for greater throughput [45]. On top of this, the Ethereum foundation is actively updating the network architecture and consensus algorithm to achieve closer to 100,000 transactions per second on the Layer-1 network [8] [44].

4 Blockchain and Agent-Based Simulations

There exists a body of research concerned with leveraging decentralised peer-to-peer networks, such as blockchain technology (BCT), to enhance elements of centralised systems [34] [48].

Some of this work is concerned with utilising BCT with ABM and Multi-Agent Systems (MAS) [36] [9] [46]. Generally, there exists two approaches when applying BCT to ABM and MAS. The first approach uses traditional centralised agents running on a centralised machine where the blockchain is used as an immutable ledger and trusted data store. The second approach is more integrated, the agents directly interact with, or are part of, the blockchain technology. [11] describe these two approaches as "agent-vs-blockchain" - where the agent and blockchain run side by side, allowing the agents to exploit blockchain services when needed and "agent-to-blockchain" - where the effort is focused on incorporating agent-oriented models and technologies directly into the blockchain.

Limited research could be found exploring the use of blockchain technology to achieve scale in agent-based simulations. Most studies were concerned with other affordances offered by BCT such as trust [10] [3] [18] [51], data immutability [18] [3] [51], mitigating single-points of failure [3], cost [3], and security [47] [10].

Blockchains can be typically classified in two main categories. Permissionless public blockchains allow the addition of blocks to the chain by any process provided a cryptographic puzzle is solved, examples of permissionless blockchains include the Ethereum and Bitcoin networks. Permissioned blockchains are run by a preselected closed group or centralised entity, examples of permissioned blockchains include Hyperledger Fabric (HF) [9]. One of the core benefits of permissioned blockchains, such as HF, is that they can handle a greater number of transactions per second due to their centralised nature.

In their work [9] used the HF blockchain to create an agent based autonomous intersection management system for smart cities. They created agent-based smart contracts which could handle one transaction per second on this network. [46] used the public Ethereum blockchain along with Software Defined Networking (SDN) to create a blockchain based hybrid network architecture for

smart cities. With this architecture they achieved a median latency of 3.9 seconds per transaction. They also tested the latency directly on the Ethereum network without the addition SDN. This resulted in a median latency of 21 seconds per transaction. From these works it is clear that permissioned blockchains such as HF and permissionless blockchains with some centralised system components achieve greater transactions per second compared to architectures running solely on permissionless blockchains such as the Ethereum network.

5 Conclusion

This review explored existing literature and provides an overview of ABM; issues with, and current solutions for, scaling ABMs; an overview of BCT; and existing work utilising BCT for scaling ABMs. It is clear that in order to model systems of increasing complexity an interoperable, reusable, and scalable solution is required. Current state-of-the-art solutions to this scaling problem propose the use of Hybrid Simulation in a microservice environment. Given the ability to execute arbitrary functions across a global decentralised P2P network, BCT offers another possible solution to this scaling problem. No existing literature could be found which looked specifically at the use of BCT to achieve web scale agent-based simulations. Given the Ethereum layer-1 network is currently limited in terms of transactions per second, current research into utilising BCT to run ABMs is forced to use either centralised components or private blockchains in order to achieve throughput similar to production grade centralised systems. However, new layer-2 solutions and upcoming upgrades to the layer-1 protocol will allow the public Ethereum network to scale it's transactions per second beyond current centralised systems. As a consequence, the remaining research will focus on creating a proof of concept, scalable, ABM using BCT and smart contracts.

References

[1] Sameera Abar et al. "Agent Based Modelling and Simulation tools: A review of the state-of-art software". In: *Computer Science Review* 24 (2017), pp. 13–33.

[2] Robert John Allan et al. *Survey of agent based modelling and simulation tools*. Science & Technology Facilities Council New York, 2010.

[3] Hany F Atlam et al. "A Review of Blockchain in Internet of Things and AI". In: *Big Data and Cognitive Computing* 4.4 (2020), p. 28.

[4] Robert Axelrod. "The complexity of cooperation". In: *The Complexity of Cooperation*. Princeton university press, 1997.

[5] Leo Maxim Bach, Branko Mihaljevic, and Mario Zagar. "Comparative analysis of blockchain consensus algorithms". In: *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. Ieee. 2018, pp. 1545–1550.

[6] Eric Bonabeau. "Agent-based modeling: Methods and techniques for simulating human systems". In: *Proceedings of the national academy of sciences* 99.suppl 3 (2002), pp. 7280–7287.

[7] Vitalik Buterin. *Ethereum: A next-generation smart contract and decentralized Application Platform.* 2014. URL: `https://ethereum.org/en/whitepaper/`.

[8] Vitalik Buterin. *The merge.* URL: `https://ethereum.org/en/upgrades/merge/`.

[9] Alina Buzachis et al. "A multi-agent autonomous intersection management (MA-AIM) system for smart cities leveraging edge-of-things and Blockchain". In: *Information Sciences* 522 (2020), pp. 148–163.

[10] Davide Calvaresi et al. "Explainable multi-agent systems through blockchain technology". In: *International Workshop on Explainable, Transparent Autonomous Agents and Multi-Agent Systems*. Springer. 2019, pp. 41–58.

[11] Giovanni Ciatto et al. "From agents to blockchain: Stairway to integration". In: *Applied Sciences* 10.21 (2020), p. 7460.

[12] Nicholson Collier and Michael North. "Repast HPC: A platform for large-scale agent-based modeling". In: *Large-scale computing techniques for complex system simulations* (2011), pp. 81–110.

[13] Nicholson Collier, Jonathan Ozik, and Charles M Macal. "Large-scale agent-based modeling with repast HPC: A case study in parallelizing an agent-based model". In: *European Conference on Parallel Processing*. Springer. 2015, pp. 454–465.

[14] Rem Collier, Seán Russell, and Fatemeh Golpayegani. "Harnessing Hypermedia MAS and Microservices to Deliver Web Scale Agent-based Simulations". In: (2021).

[15] Rem W Collier, Seán Russell, and David Lillis. "Reflecting on agent programming with AgentSpeak (L)". In: *International Conference on Principles and Practice of Multi-Agent Systems*. Springer. 2015, pp. 351–366.

[16] Andrew T Crooks and Alison J Heppenstall. "Introduction to agent-based modelling". In: *Agent-based models of geographical systems*. Springer, 2012, pp. 85–105.

[17] Akshat Dhaon and Rem W Collier. "Multiple inheritance in AgentSpeak (L)-style programming languages". In: *Proceedings of the 4th International Workshop on Programming based on Actors Agents & Decentralized Control*. 2014, pp. 109–120.

[18] André Diogo et al. "A Multi-Agent System Blockchain for a Smart City". In: *The Third International Conference on Cyber-Technologies and Cyber-Systems (CYBER 2018), no. Figure*. Vol. 1. 2018, pp. 68–73.

[19] Nicola Dragoni et al. "Microservices: yesterday, today, and tomorrow". In: *Present and ulterior software engineering* (2017), pp. 195–216.

[20] Juan Du et al. "An ontology and multi-agent based decision support framework for prefabricated component supply chain". In: *Information Systems Frontiers* 22.6 (2020), pp. 1467–1485.

[21] Cynthia Dwork and Moni Naor. "Pricing via processing or combatting junk mail". In: *Annual international cryptology conference*. Springer. 1992, pp. 139–147.

[22] Tillal Eldabi et al. "Hybrid simulation challenges and opportunities: a life-cycle approach". In: *2018 winter simulation conference (WSC)*. IEEE. 2018, pp. 1500–1514.

[23] Stan Franklin and Art Graesser. "Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents". In: *International workshop on agent theories, architectures, and languages*. Springer. 1996, pp. 21–35.

[24] Nigel Gilbert and Pietro Terna. "How to build and use agent-based models in social science". In: *Mind & Society* 1.1 (2000), pp. 57–72.

[25] Stuart Haber and W Scott Stornetta. "How to time-stamp a digital document". In: *Conference on the Theory and Application of Cryptography*. Springer. 1990, pp. 437–455.

[26] Brian L Heath and Ray R Hill. "The early history of agent-based modeling". In: *IIE Annual Conference. Proceedings*. Institute of Industrial and Systems Engineers (IISE). 2008, p. 971.

[27] Kasun Indrasiri and Prabath Siriwardena. "The case for microservices". In: *Microservices for the Enterprise*. Springer, 2018, pp. 1–18.

[28] *Introduction to smart contracts*. 2022. URL: `https://ethereum.org/en/developers/docs/smart-contracts/`.

[29] Toru Ishida, Les Gasser, and Hideyuki Nakashima. *Massively Multi-Agent Systems I: First International Workshop, MMAS 2004, Kyoto, Japan, December 10-11, 2004, Revised Selected and Invited Papers*. Vol. 3446. Springer, 2005.

[30] Markus Jakobsson and Ari Juels. "Proofs of work and bread pudding protocols". In: *Secure information networks*. Springer, 1999, pp. 258–272.

[31] Nadeem Jamali, Paul Scerri, and Toshiharu Sugawara. *Massively Multi-Agent Technology: AAMAS Workshops, MMAS 2006, LSMAS 2006, and CCMMS 2007 Hakodate, Japan, May 9, 2006 Honolulu, HI, USA, May 15, 2007, Selected and Revised Papers*. Vol. 5043. Springer, 2008.

[32] Anne-Marie Kermarrec and Francois Taiani. "Want to scale in centralized systems? Think P2P". In: *Journal of Internet Services and Applications* 6.1 (2015), pp. 1–12.

[33] Olga Victorovna Kitova et al. "Hybrid intelligent system of forecasting of the socio-economic development of the country". In: *International Journal of Applied Business and Economic Research* 14.9 (2016), pp. 5755–5766.

[34] Zhi Li, Ali Vatankhah Barenji, and George Q Huang. "Toward a blockchain cloud manufacturing system as a peer to peer distributed network platform". In: *Robotics and computer-integrated manufacturing* 54 (2018), pp. 133–144.

[35] Soo Hoon Maeng, Meryam Essaid, and Hong Taek Ju. "Analysis of ethereum network properties and behavior of influential nodes". In: *2020 21st Asia-Pacific network operations and management symposium (APNOMS)*. IEEE. 2020, pp. 203–207.

[36] Luana Marrocco et al. “Basic: Towards a blockchained agent-based simulator for cities”. In: *International Workshop on Massively Multiagent Systems*. Springer. 2018, pp. 144–162.

[37] Navonil Mustafee et al. “Purpose and benefits of hybrid simulation: contributing to the convergence of its definition”. In: *2017 Winter Simulation Conference (WSC)*. IEEE. 2017, pp. 1631–1645.

[38] Satoshi Nakamoto. “Bitcoin: A peer-to-peer electronic cash system”. In: *Decentralized Business Review* (2008), p. 21260.

[39] Muaz Niazi and Amir Hussain. “Agent-based computing from multi-agent systems to agent-based models: a visual survey”. In: *Scientometrics* 89.2 (2011), pp. 479–499.

[40] Eoin O’Neill et al. “Explicit modelling of resources for multi-agent microservices using the cartago framework”. In: *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*. 2020, pp. 1957–1959.

[41] Hazel R Parry and Mike Bithell. “Large scale agent-based modelling: A review and guidelines for model scaling”. In: *Agent-based models of geographical systems* (2012), pp. 271–308.

[42] J Gareth Polhill et al. “Crossing the chasm: a ‘tube-map’for agent-based social simulation of policy scenarios in spatially-distributed systems”. In: *GeoInformatica* 23.2 (2019), pp. 169–199.

[43] Rüdiger Schollmeier. “A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications”. In: *Proceedings First International Conference on Peer-to-Peer Computing*. IEEE. 2001, pp. 101–102.

[44] Daniel Sel, Kaiwen Zhang, and Hans-Arno Jacobsen. “Towards solving the data availability problem for sharded ethereum”. In: *Proceedings of the 2Nd Workshop on Scalable and Resilient Infrastructures for Distributed Ledgers*. 2018, pp. 25–30.

[45] Cosimo Sguanci, Roberto Spatafora, and Andrea Mario Vergani. “Layer 2 blockchain scaling: A survey”. In: *arXiv preprint arXiv:2107.10881* (2021).

[46] Pradip Kumar Sharma and Jong Hyuk Park. “Blockchain based hybrid network architecture for the smart city”. In: *Future Generation Computer Systems* 86 (2018), pp. 650–655.

[47] Anshu Shukla, Swarup Kumar Mohalik, and Ramamurthy Badrinath. “Smart contracts for multiagent plan execution in untrusted cyber-physical systems”. In: *2018 IEEE 25th International Conference on High Performance Computing Workshops (HiPCW)*. IEEE. 2018, pp. 86–94.

[48] Yun Tang et al. “Deploying P2P networks for large-scale live video-streaming service [Peer-to-peer multimedia streaming]”. In: *IEEE Communications Magazine* 45.6 (2007), pp. 100–106.

[49] Simon JE Taylor. “Distributed simulation: State-of-the-art and potential for operational research”. In: *European Journal of Operational Research* 273.1 (2019), pp. 1–19.

[50] Simon JE Taylor et al. "Innovations in simulation: experiences with cloud-based simulation experimentation". In: *2020 Winter Simulation Conference (WSC)*. IEEE. 2020, pp. 3164–3175.

[51] Antonio Tenorio-Fornés, Samer Hassan, and Juan Pavón. "Open peer-to-peer systems over blockchain and ipfs: An agent oriented framework". In: *Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems*. 2018, pp. 19–24.

[52] Ingo J Timm and Dirk Pawlaszczyk. "Large scale multiagent simulation on the grid". In: *CCGrid 2005. IEEE International Symposium on Cluster Computing and the Grid, 2005.* Vol. 1. IEEE. 2005, pp. 334–341.

[53] Rem W. Collier et al. "MAMS: Multi-Agent MicroServices". In: *Companion Proceedings of The 2019 World Wide Web Conference*. 2019, pp. 655–662.

[54] Gavin Wood et al. "Ethereum: A secure decentralised generalised transaction ledger". In: *Ethereum project yellow paper* 151.2014 (2014), pp. 1–32.

[55] Michael Wooldridge and Nicholas R Jennings. "Intelligent agents: Theory and practice". In: *The knowledge engineering review* 10.2 (1995), pp. 115–152.